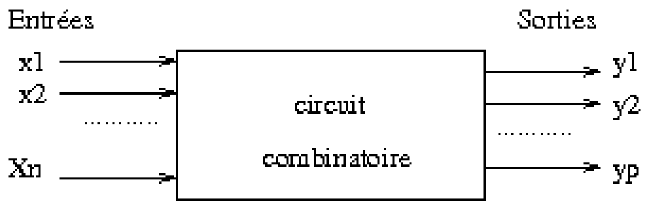
[Chapitre précédent](#)

Le calcul de circuits

Introduction

- Nous avons discrètement abordé ce point dans la partie précédente en montrant que le calcul de la retenue dans une addition de chiffres binaires avec retenue entrante ou le calcul de la retenue dans une soustraction étaient définissables sous forme d'une expression logique.

- Nous nous limiterons ici aux **circuits combinatoires** répondant au schéma général suivant :



Un tel circuit réalise donc le calcul de *p* fonctions f_1, ..., f_p et on a $\forall i \in \{1, ...p\}, y_i = f_i(x_1, ..., x_n)$
Les valeurs de sortie à un instant donné n'y dépendent que des valeurs courantes en entrée.

Il s'agit donc de dispositifs physiques

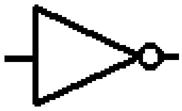
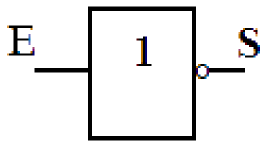
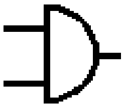
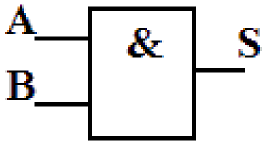
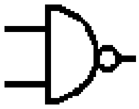
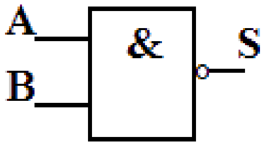
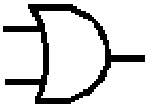
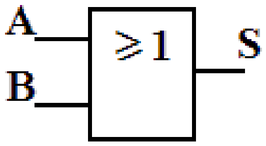
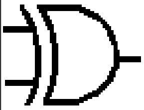
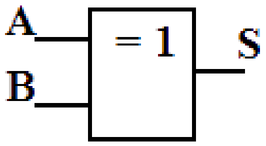
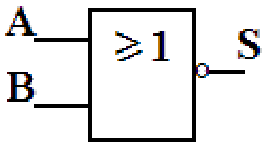
- recevant en entrée des informations binaires ayant seulement deux états stables notées 0 et 1. Chaque information est transmise sur une ligne et sa valeur est matérialisée par la présence ou l'absence de courant sur cette ligne.
  Ces circuits sont réalisés par combinaison de portes logiques réalisant électroniquement les connecteurs de la logique propositionnelle que nous venons d'étudier ou de circuits déjà construits (à partir de portes logiques) en utilisant des portes logiques pour les combiner.
  Ces composants élémentaires sont constitués de plusieurs transistors.
  Une porte possède d'une part une (dans le cas de la porte `NOT`) ou deux entrées et d'autre part une sortie.
- produisant en sortie un ensemble d'informations binaires : chaque information est transmise sur une ligne et sa valeur est matérialisée par la présence ou l'absence de courant sur la ligne correspondante.

: l'autre type de circuit, les **circuits séquentiels**, fait intervenir le temps : les valeurs de sortie dépendent non seulement des valeurs courantes en entrée mais aussi des données antérieures. Sur de tels circuits, certaines sorties peuvent être reliées à des entrées du circuit.

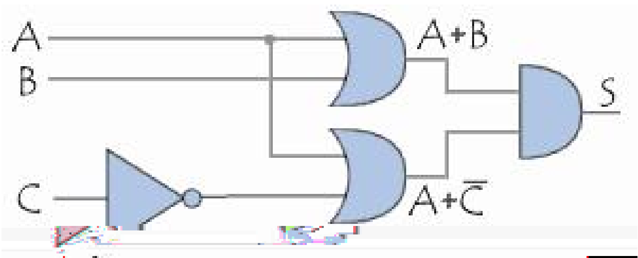
- **Représentation des portes**

A chaque type de porte sont associés un schéma universel et un schéma européen :

nom	schéma universel	schéma européen

négation NOT		
conjonction AND		
NAND		
disjonction OR		
disjonction exclusive XOR		
NOR		

Ainsi on associe à la formule $(A + B).(A + \overline{C})$ le schéma de circuit suivant



Simplifier une fonction

- Nous avons abordé la simplification algébrique des formules logiques en procédant de manière «empirique» en utilisant les propriétés fondamentales des connecteurs standard.

Si on considère la fonction logique de 4 variables associée à la table de vérité suivante (on a adopté comme dans la suite la notation 0/1 au lieu de la notation F/V car elle est plus proche de la vision binaire informatique des choses)

a	b	c	d	E
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0

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

on obtient la forme normale disjonctive suivante :

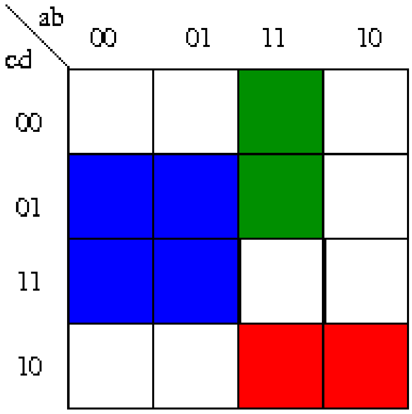
$$\overline{a}.\overline{b}.\overline{c}.d + \overline{a}.\overline{b}.c.d + \overline{a}.b.\overline{c}.d + \overline{a}.b.c.d + a.\overline{b}.c.\overline{d} + a.\overline{b}.c.d + a.b.\overline{c}.\overline{d}$$

Entreprendre de la simplifier par des transformations algébriques, sans méthode, relève de la gageure.

- Nous présentons rapidement ici la **méthode de Karnaugh**, méthode graphique qui permet de simplifier des formules construites avec un nombre restreint de variables, telle que l'expression précédente.

Les simplifications reposent sur le fait que $a.b + a.\overline{b}$ est équivalent à a. La méthode repose sur une représentation un peu différente de la table de vérité qui permet de repérer facilement les applications possibles de cett complémentée, les autres facteurs étant de la forme $x + \overline{x}$ et pouvant donc être éliminées. Le repérage d'un groupe de *n* variables permet l'élimination de *n* variables et donc la fabrication d'un terme de *m* - *n* variables.

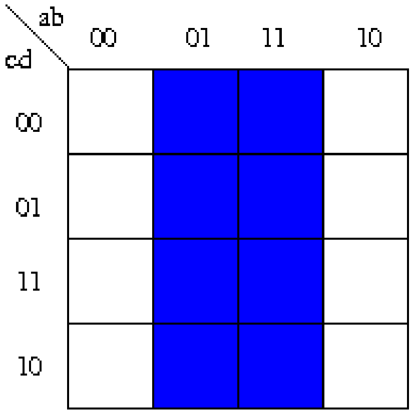
Des exemples de ces simplifications sont données ci-après :



– Le carré bleu conduit à la formule $\overline{a}.d$

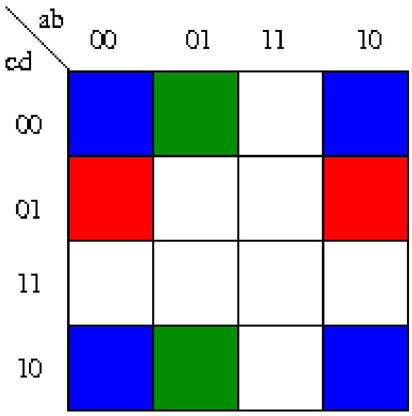
– Le rectangle vert conduit à la formule $a.b.\overline{c}$

– Le rectangle rouge conduit à la formule $a.c.\overline{d}$



Le rectangle bleu correspond à la formule b

La difficulté peut venir du repérage des groupes « périphériques » utilisant la structure circulaire et donc impliquant les bords et les coins



– Les cases bleues constituent un groupe et donnent la formule $\overline{b}.\overline{d}$

– Les cases rouges constituent un groupe et donnent la formule $\overline{b}.\overline{c}.d$

– Les cases vertes constituent un groupe et donnent la formule $\overline{a}.b.\overline{d}$

- L'exemple suivant illustre l'écriture de la table utilisée par la méthode de Karnaugh pour une **formule de trois variables** : on trouve à gauche l'écriture de la table de manière usuelle et à droite celle à deux dimensions utilisée (en gras la valeur de la fonction pour les valeurs des variables indexant les lignes et colonnes) :

a	b	c	f(a,b,c)
0	0	0	0
0	0	1	0
0	1	0	0

b c → a	0 0	0 1	1 1	1 0

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

↓				
0	0	0	1	0
1	0	1	1	1

b c / a	0 0	0 1	1 1	1 0
0	0	0	1	0
1	0	1	1	1

on procède aux regoupements des positions de 1 dans les plus grands rectangles possibles de manière à «couvrir» chacune d'elles au moins une fois :

- le rectangle vert : il correspond à la formule $a.b.c+\bar{a}.b.c$ ou encore $(a+\bar{a}).b.c$, c'est-à-dire $b.c$
- le rectangle rouge : de la même manière, il correspond à $a.c$
- le rectangle bleu : de la même manière, il correspond à $a.b$

Cela donne donc (à une commutation des termes près) l'expression « $a.b + a.c + b.c$ »

- Un point important que ne fait pas apparaître cet exemple, c'est l'aspect circulaire du tableau. Ainsi dans l'exemple suivant :

b c / a	0 0	0 1	1 1	1 0
0	1	1	0	1
1	1	0	0	1

- les quatre coins, qui correspondent à la valeur 1 de la fonction, constituent, dans une vision circulaire, un carré bleu (et donc un rectangle) à considérer dans la recehrche d'une expression réduite de la fonction. Cela correspond à la formule $\bar{a}.\bar{b}.\bar{c}+\bar{a}.b.\bar{c} + a.\bar{b}.\bar{c} + a.b.\bar{c}$
qui est équivalente à a
- le rectangle rouge conduit à $\bar{a}.\bar{b}$
- la fonction s'écrit donc $a.\bar{b} + a$

- Attaquons nous maintenant à l'exemple de **formule à quatre variables** donné en début de cette présentation.

Sa représentation en deux dimensions en vue de manipulation par la méthode de Karnaugh est :

c d a b	0 0	0 1	1 1	1 0
0 0	0	1	1	0
0 1	0	1	1	0
1 1	1	0	0	0
1 0	0	0	1	1

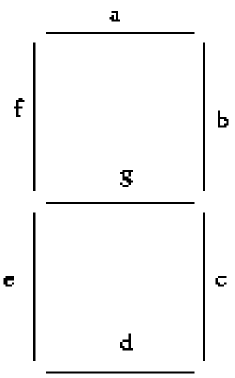
On peut y effectuer les regroupements suivants incluant les positions valant `1` :

c d a b	0 0	0 1	1 1	1 0
0 0	0	1	1	0
0 1	0	1	1	0
1 1	1	0	0	0
1 0	0	0	1	1

- pour le rectangle bleu on obtient : $\overline{a}.d$
- pour le rectangle rouge on obtient : $a.\overline{b}.c$
- pour le rectangle vert on obtient : $a.b.\overline{c}.\overline{d}$
- on obtient la forme suivante pour l'expression
  $\overline{a}.d + a.\overline{b}.c + a.b.\overline{c}.\overline{d}$

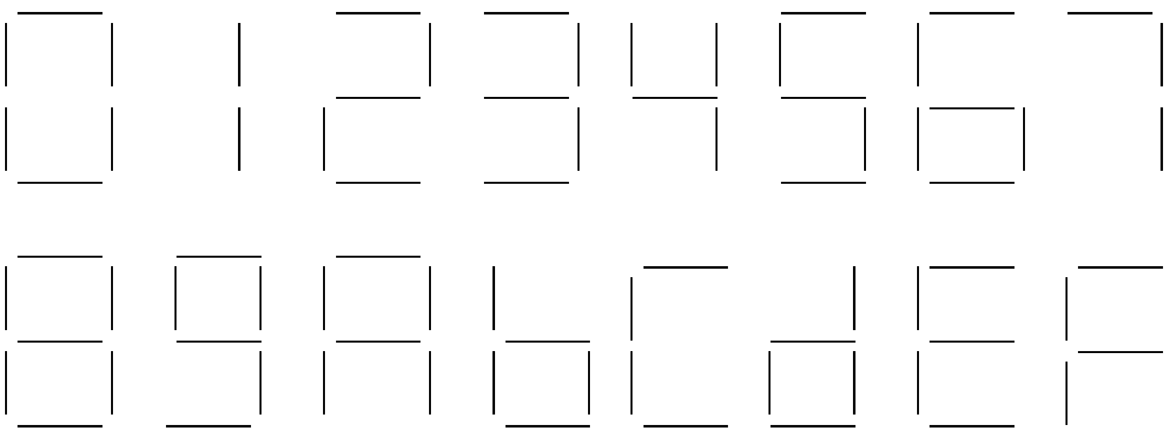
- **Exercice partiellement résolu**

On souhaite réaliser un afficheur 7 segments permettant de traduire un nombre binaire codé sur 4 bits en le caractère hexadécimal **`0 1 2 3 4 5 6 7 8 9 A B C D E F`**.
Chacun de ces seize caractères sera visualisé en utilisant 7 segments correspondant physiquement à des diodes electroluminescentes :



Les 7 diodes et leur nom

La figure suivante montre quelles diodes il faut éclairer (et donc auxquelles il faut fournir du courant) pour obtenir chacun des des caractères :



Les différentes diodes fonctionnent selon la table suivante selon les valeurs des 4 bits en entrée :

caractère	code 4 bits	a	b	c	d	e	f	g
0	0 0 0 0	1	1	1	1	1	1	0
1	0 0 0 1	0	1	1	0	0	0	0
2	0 0 1 0	1	1	0	1	1	0	1
3	0 0 1 1	1	1	1	1	0	0	1
4	0 1 0 0	0	1	1	0	0	1	1
5	0 1 0 1	1	0	1	1	0	1	1
6	0 1 1 0	1	0	1	1	1	1	1
7	0 1 1 1	1	1	1	0	0	0	0
8	1 0 0 0	1	1	1	1	1	1	1
9	1 0 0 1	1	1	1	1	0	1	1
A	1 0 1 0	1	1	1	0	1	1	1
B	1 0 1 1	0	0	1	1	1	1	1
C	1 1 0 0	1	0	0	1	1	1	0
D	1 1 0 1	0	1	1	1	1	0	1
E	1 1 1 0	1	0	0	1	1	1	1
F	1 1 1 1	1	0	0	0	1	1	1

Le diagramme de Karnaugh qui lui correspond est le suivant :

c d a b	0 0	0 1	1 1	1 0
0 0	1	0	1	1
0 1	0	1	1	1
1 1	1	0	1	1
1 0	1	1	0	1

Ce qui permet les six regroupements suivants :

c d / a b	0 0	0 1	1 1	1 0
0 0	1	0	1	1
0 1	0	1	1	1
1 1	1	0	1	1
1 0	1	1	0	1

d'où l'on déduit 6 termes :

- un carré rouge de 4 position utilisant la circularité de la table et qui donne le terme de 2 éléments « $a.\overline{d}$ »
- un carré bleu de 4 positions qui donne le terme de 2 éléments « $\overline{a}.c$ »
- un carré vert de 4 positions qui donne le terme de 2 éléments « **b.c** »
- un carré magenta de 4 positions (les 4 coins) utilisant la circularité de la table et qui donne le terme de 2 éléments « $\overline{b}.\overline{d}$ »
- un rectangle vert de 2 positions qui donne le terme de 3 éléments « $a.\overline{b}.\overline{c}$ »
- un rectangle bleu de 2 positions qui donne le terme de 3 éléments « $\overline{a}.b.d$ »

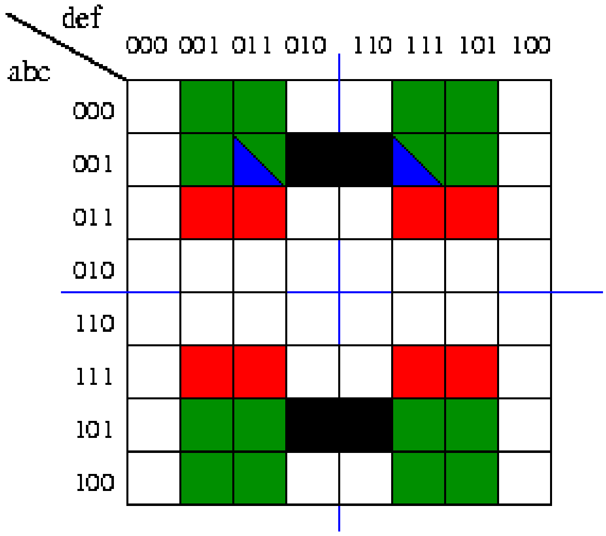
Le fonctionnement de la diode *a* est donc contrôlé par la fonction définie par la formule

$$a.\overline{d} + \overline{a}.c + b.c + \overline{b}.\overline{d} + a.\overline{b}.\overline{c} + \overline{a}.b.d$$

Exercice : chercher sur le même principe des expressions pour le fonctionnement des six autres diodes.

- **Et pour plus de quatre variables (5 ou 6)**

Le schéma suivant montre, sur un diagramme à 6 variables, comment on prend en compte deux axes de symétrie pour faire les regroupements :



– les cases bleues forment un groupe de deux cases

– les cases noires forment un groupe de quatre cases

– les cases rouges forment un groupe de huit cases

– les cases vertes forment un groupe de seize cases

Exercice : déterminer les formules associées aux différents groupes colorés dans le diagramme précédent.

Des exemples de circuits combinatoires

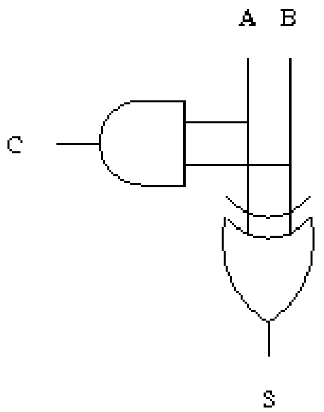
Histoires d'additions

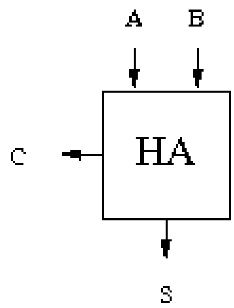
- Commençons par ce qu'on appelle un demi-additionneur (HA) qui permet de calculer la somme de deux bits *a* et *b* : il produit produit un chiffre *s* et une retenue *c* (*carry*).

La table de vérité des deux sorties *s* et une retenue *c* est :

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

On en déduit pour la valeur de **s** l'expression **a   b** et pour celle de **c** l'expression **a.b**
Cela conduit aux schémas suivants :





- Ce type de circuit ne permet pas l'addition de bits en cascade puisqu'il ne possède en entrée que deux bits : la réalisation d'une addition de nombres de plusieurs chiffres suppose de prendre en compte lors de l'addition de deux bits la valeur de la retenue provenant de l'addition des chiffres précédents. C'est ce que réalise un **additionneur complet**.

Question : construire un additionneur complet à partir de deux demi-additionneurs.

- Considérons les tables de vérité correspondant au calcul du chiffre s_i et de la retenue r_{i+1} correspondante à partir des deux chiffres a_i et b_i et de la retenue r_i.

a_i	b_i	r_i	r_{i+1}	s_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

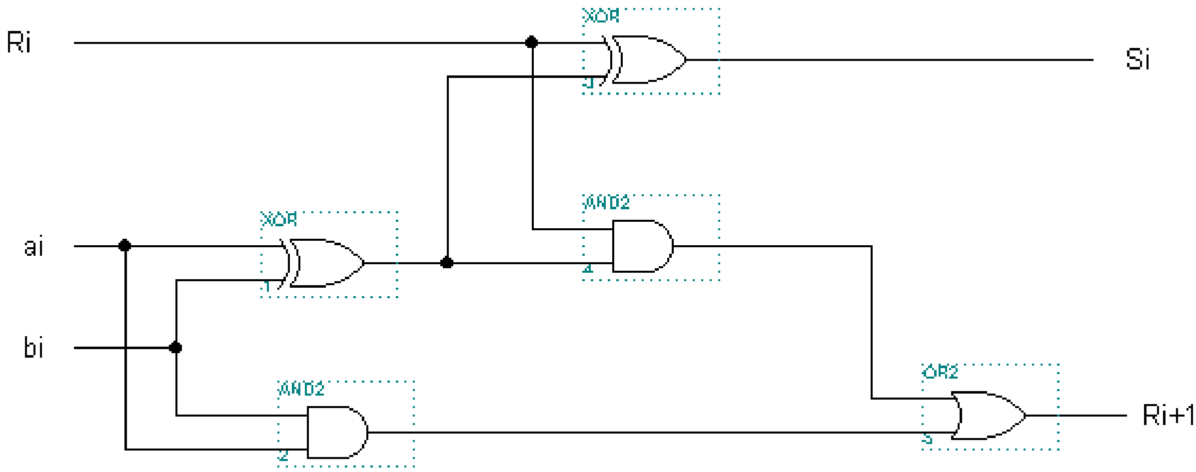
On en déduit pour s_i les formules équivalentes suivantes

`a.b + b.c + a.c` et `a.b + (a  b).c`

et pour r_{i+1} les formules

$\overline{a}.\overline{b}.c + \overline{a}.b.\overline{c} + a.b.c + a.\overline{b}.\overline{c}$ et `a  b &oplus c` (diagramme de Karnaugh en damier)

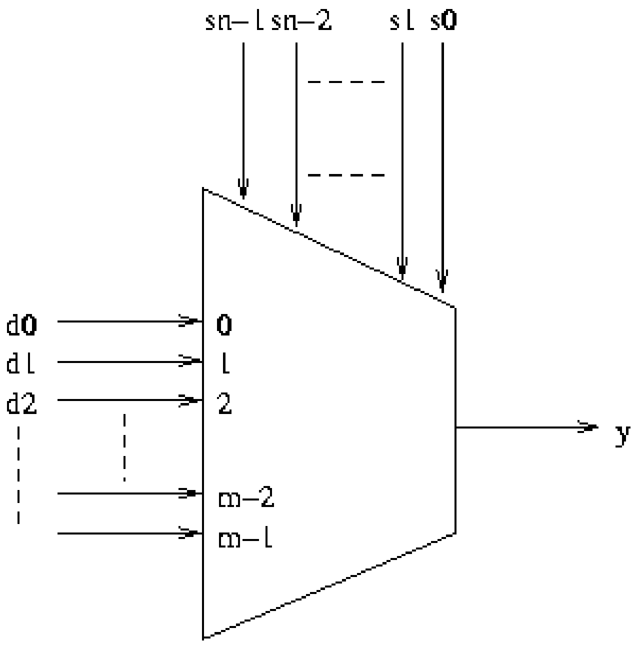
On en déduit le schéma suivant :



Les multiplexeurs

- Le rôle d'un multiplexeur est de sélectionner une entrée parmi l'ensemble de ses entrées et permet de réaliser des aiguillages.
Pour ce faire, il dispose de
  - de m lignes d'entrées : en général m est une puissance de deux et donc $m = 2^n$.
    Les entrées peuvent être numérotées de 0 à $m-1$
  - de n lignes de sélection : les n valeurs binaires permettent de désigner sous forme binaire une entrée (par son numéro codé en binaire).
  - une valeur de sortie

- Un tel multiplexeur correspond au schéma suivant :



la sortie y a comme valeur celle de l'entrée d $\sum_{i=0}^{n-1} s_i \times 2^i$

- Intéressons-nous au multiplexeur à deux entrées (et donc une seule voie de sélection) : il permet de sélectionner une des deux entrées (d_0 ou d_1 selon que la valeur de l'entrée de sélection s est `0` ou `1`).

Pour réaliser ce circuit, on peut procéder de deux manières

  - contruire la table de vérité et calculer une forme réduite

d_0	d_1	s	$f(d_0,d_1,s)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

s → d_0 d_1 ↓	0	1
0 0	**0**	**0**
0 1	**0**	**1**
1 1	**1**	**1**
1 0	**1**	**0**

Dans cette table, on réalise les regroupements suivants :

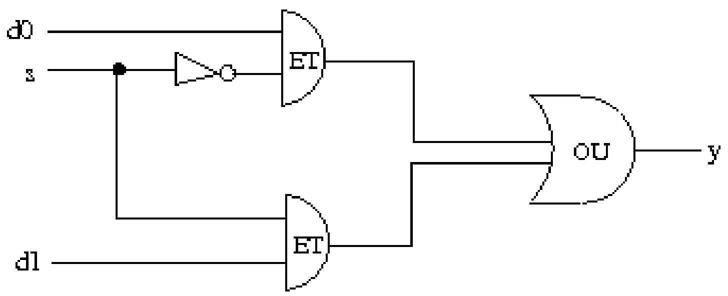
s / d0 d1	0	1
0 0	0	0
0 1	0	1
1 1	1	1
1 0	1	0

- le rectangle bleu : il correspond à la formule « $\overline{s}.d_0$ »,
- le rectangle rouge : il correspond à « $s.d_1$ »
- la formule associée est donc « $\overline{s}.d_0 + s.d_1$ »

- constater qu'un tel multiplexeur n'est rien d'autre que de l'opérateur « `si s alors` d_0 `sinon` d_1 »

  On obtient alors directement la forme recherchée « `si s alors` d_0 `sinon` d_1 ».

Le schéma suivant correspond au « montage électronique » du multiplexeur à deux entrées :



Question : construire un multiplexeur à quatre entrées uniquement par combinaison de multiplexeurs à deux entrées.

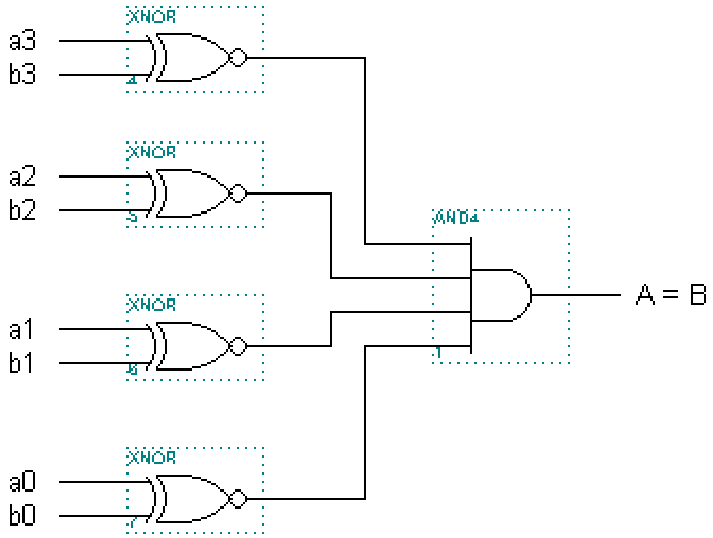
Histoires de comparaison

- Il s'agit tout d'abord de créer des circuits permettant de comparer deux bits. Parmi les questions qu'on peut se poser :
  - les deux bits sont-ils identiques?
  - les deux bits sont-ils complémentaires?

On peut exprimer chacune en termes de circuits élémentaires `ET`, `OU`, `XOR` et `NON` de la manière suivante :

- l'égalité des deux bits `a` et `b` correspond aux formules équivalentes
  $a.b + \overline{a}.\overline{b}$ et $\overline{a \quad b}$ (opérateur `XNOR`)
- le test de complémentarité des deux bits `a` et `b` correspond aux formules équivalentes
  $a.\overline{b} + \overline{a}.b$ et $a \quad b$

Pour comparer par exemple deux mots $a = a_3a_2a_1a_0$ et $b = b_3b_2b_1b_0$ de 4 bits chacun, on peut s'appuyer sur le comparateur de deux bits (circuit `XNOR`) pour réaliser un circuit (on utilise un circuit AND_4 pour faire le `ET` de 4 entrées) :



Les (en)codeurs décodeurs

- Un (en)codeur n bits est un circuit qui possède 2^n entrées exclusives (c'est-à-dire dont une et une seule est active et donc égale à 1 à un instant donné) et n sorties qui permettent de coder en binaire le numéro de l'entrée active.

Exercice : donner les relations entre les 8 entrées e_0, ..., e_7, et les 3 sorties s_0,s_1 et s_2 d'un codeur à 8 entrées et 3 sorties. En déduire un schéma pour le réaliser.

- Les décodeurs réalisent l'opération inverse de la précédente. Un décodeur dispose donc de n entrées et 2^n sorties. Son rôle est d'activer la sortie correspondant à l'entrée correspondant au numéro codé sur les n entrées.

Un tel circuit intervient par exemple dans les circuits mémoire pour le décodage des adresses et l'accès à la mémoire.

Exercice : donner un schéma pour réaliser un décodeur à 3 entrées et 8 sorties.

Chapitre précédent