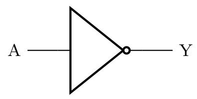
Architecture des ordinateurs
IFT1227
Introduction

Franz Girardin

22 janvier 2024

Couche logique numérique Constituées de *portes logiques* construite à partir de **transisteurs** qui prennent un **signal 0** ou **1** et calcule une **fonction logique** ET, OU et NON, etc.

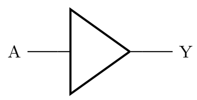
Porte NON



A	Y
1	0
0	1

$\boxed{Y = \overline{A}}$

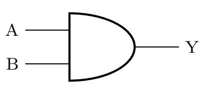
Porte BUF



A	Y
0	0
1	1

$\boxed{Y = A}$

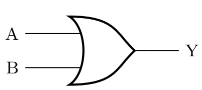
Porte ET



A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

$\boxed{Y = AB}$

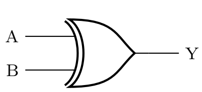
Porte OU



A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

$\boxed{Y = A + B}$

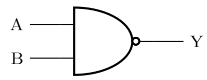
Porte XOR



A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

$\boxed{Y = A \oplus B}$

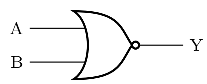
Porte NAND



A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

$\boxed{Y = \overline{AB}}$

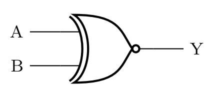
Porte NOR



A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

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

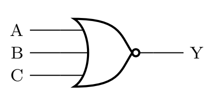
Porte XNOR



A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

$\boxed{Y = \overline{A \oplus B}}$

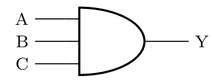
Porte NOR3



A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

$\boxed{Y = \overline{A + B + C}}$

Porte AND3



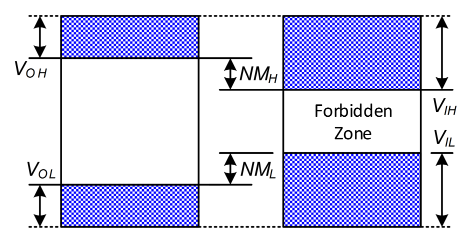
A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

$\boxed{Y = ABC}$

Définition de la marge de bruit Tolérance d'un circuit aux **perturbations** pouvant fausser l'interprétation du signal.

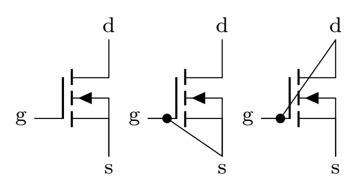
- V_{IH} : $\min(V | V_{in} ::= \mathbf{1})$
- V_{IL} : $\max(V | V_{in} ::= \mathbf{0})$
- Signal reçu **min** ou **max** est être interprété comme **1** ou **0**.
- V_{OH} : $\min(V | V_{out} ::= \mathbf{1})$
- V_{OL} : $\max(V | V_{out} ::= \mathbf{0})$
- Signal **min** ou **max** que l'émetteur s'engage à fournir pour être interprété comme **1** ou **0**.

$$NM_H = V_{OH}\ (\textit{ém.}) - V_{IH}\ (\textit{src.})$$
$$NM_L = V_{IL}(\textit{src.}) - V_{OL}(\textit{ém.})$$



Transistors

Éléments de base des circuits électroniques. Ils sont composés de trois broches ; le drain, la source, et la **grille** qui contrôle les deux autres comme un interrupteur. La figure suivante représente un transistor hors tension *off*, un transistor hors tension mais polarisé *off* et un transistor sous tension *on*.



Composition d'un circuit

Circuit::= E., S., spec. *fonct.*, spec. *temp.*

Propriétés d'un circ. combinatoire

- **Noeud** $\Longrightarrow$ *In* ou **connexion à** un *Out*.
- **Aucun** chemin cyclique.

Sommes de produits SOP En considérant les variables *In* d'une ligne de la table de vérité, il faut identifier la **conjonction** (produit) nécessaire pour engendrer un **1** logique. Un **minterm** est une représentation du produit engendrant un 1 logique.

A	B	Y	*minterm*
V	F	0	$\overline{A}\overline{A}$
0	1	1	$\overline{A}B$
1	0	0	$A\overline{B}$
1	1	1	AB

$$Y(A,B) = \overline{A}B + AB \leftrightarrow Y(A,B) = \sum(1,3)$$

Produits de sommes En considérant les variables *In* d'une ligne de la table de vérité, il faut identifier la **somme** nécessaire pour engendrer un **0** logique. Un **maxterm** est une représentation de la somme engendrant un 0 logique.

A	B	Y	*maxterm*
V	F	0	$A+B$
0	1	1	$A+\overline{B}$
1	0	0	$\overline{A}+B$
1	1	1	$\overline{A}+\overline{B}$

$$Y(A,B) = (A+B)(A+\overline{B}) = \prod(0,2)$$

Axiome	Dual	Nom
$B=0$ if $B \neq 1$	$B=1$ if $B \neq 0$	Bin. field
$\overline{0}=1$	$\overline{1}=0$	NOT
$0 \cdot 0 = 0$	$1+1=1$	AND/OR
$1 \cdot 1 = 1$	$0+0=0$	AND/OR
$0 \cdot 1 \cdot 0 = 0$	$1+0=1+1$	AND/OR

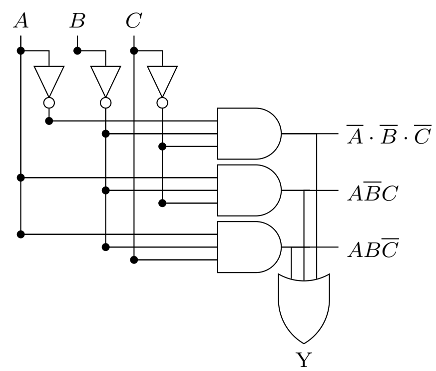
Théorème	Dual	Nom
$B \cdot 1 = B$	$B+0=B$	Identité
$B \cdot 0 = 0$	$B+1=1$	Élément nul
$B \cdot B = B$	$B+B=B$	Indépotence
	$\overline{\overline{B}}=B$	Involution
$B \cdot \overline{B} = 0$	$B+\overline{B}=1$	Complément

De Morgan

$$\neg(A+B) = \neg A + \neg B$$

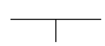
$$\neg(A \cdot B) = \neg A \cdot \neg B$$

Exemple simple de circuit combinatoire

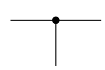


$$Y = \overline{A}\cdot\overline{B}\cdot\overline{C} + A\overline{B}C + AB\overline{C}$$

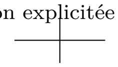
Simple règles de schématisation Les **E.** sont en haut à gauche et les **S.** sont en bas à droite ; on utilise des **fils droits**, préférablement.



(a) Fils engendrant une jonction T

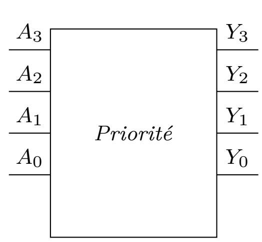


(b) Connexion explicitée par un point



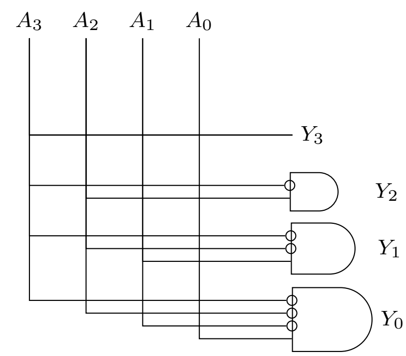
(c) Fils croisés sans point (∴ non connectés)

Circuit de priorité



A_3	A_2	A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	0
0	1	0	0	0	1	0	0
0	1	0	1	0	1	0	0
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	0
1	0	0	0	1	0	0	0
1	0	0	1	1	0	0	0
1	0	1	0	1	0	0	0
1	0	1	1	1	0	0	0
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	0
1	1	1	0	1	0	0	0
1	1	1	1	1	0	0	0

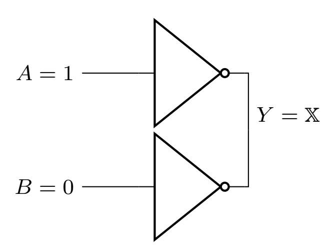
Représentation d'un circuit de priorité



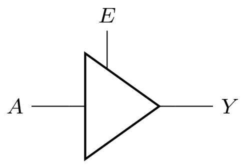
Entrée don't care ou $\mathbb{X}$ Ces entrés sont utilisées pour spécifier que la variables possédant le *don't care* n'affecte pas le résultat de la fonction logique. Une file de priorité peut être résumée par la table suivante.

A_3	A_2	A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	d	0	0	1	0
0	1	d	d	0	1	0	0
1	d	d	d	1	0	0	0

Contention : signal X Se produit lorsque les portes logiques et les entrées sont telles que la sortie à générer est **contradictoire**.



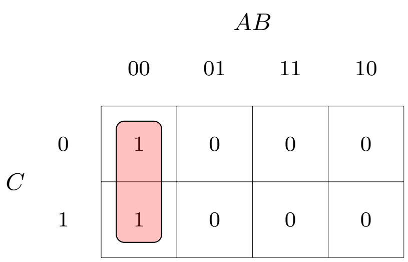
Tampon à trois états : signal $\mathbb{Z}$ Circuit dans lequel une entrée $\mathbb{E}$ est connecté à une porte tampon et contrôle la **propagation du signal**. Lorsque l'entrée $\mathbb{E}$ est sous tension haute, la porte agit comme un tampon normal ; lorsque l'entrée $\mathbb{E}$ est sous tension basse, la porte produit le signal $\mathbb{Z}$ qui indique que A est *contrôlé*.

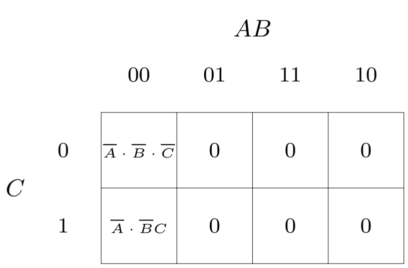


Méthodes de Karnaugh Méthode **graphique** permettant de simplifier les formules de circuits

- ▷ **Organiser** les éléments en grille de façon à ce que chaque cellule ne diffère d'une cellule voisine que par **1 bit**.
- ▷ Remplir la grille de façon à refléter le **tableau d'origine**.
- ▷ **Grouper** ou entourer les cellules adjacentes qui possèdent un **1**.

A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0



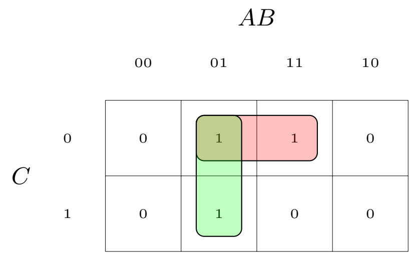


Solution : considérer les variables qui ne changent pas leurs valeurs entre les cases **groupés**. Ici, la variable C change de valeur entre le cellule 1 et la cellule 2 ; elle n'est donc pas considérée dans **l'équation simplifiée**. Nous avons alors :

$$Y = \overline{AB}$$

Autre exemple de Karnaugh-map de 3 entrées Parfois, il y a plusieurs **implicants**.

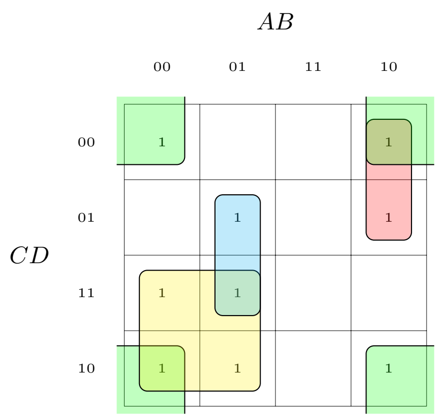
A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0



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

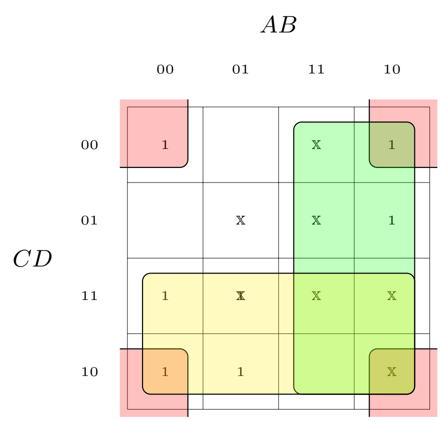
Autre exemple de Karnaugh-map de 3 entrées Pour les tables de Karnaugh à 4 entrées et plus, les implicants devinnent plus complexes.

A	B	C	D	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0



Karnaugh-map avec Don't Cares Les *don't care* peuvent complexifier la simplifiation de la fontion à cause du plus grand nombre de **cas possibles** lors du regroupement.

A	B	C	D	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	X
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X



$$Y = A + \overline{B} \cdot \overline{D} + C$$

Karnaugh-map de 5 entrées Il faut considérer **deux** K-map de **quatre** variables. Par convention, on peut omettre la première variable dans les deux K-map ; on considère que la dans la première K-map, la variable irgnorée a une valeur de **0** et, dans la 2e K-map, elle a une valeur de **1**. `Exemple sur Youtube` Soit la fonction et K-maps correspondantes :

$$f(A,B,C,D,E) = \sum(0,1,2,4,5,6,10,13,$$
$$1418,21,22,24,26,29,30)$$

K-map de $BCDE$ en considérant $A = 0$

DE \ BC	00	01	11	10
00	1_{0}	1_{1}	0_{3}	0_{2}
01	1_{4}	1_{5}	1_{7}	0_{6}
11	0_{12}	0_{13}	0_{15}	0_{14}
10	1_{8}	1_{9}	1_{11}	1_{10}

$$D\overline{E} + C\overline{D} \cdot E + \overline{A} \cdot \overline{B} \cdot \overline{D}$$

K-map de $BCDE$ en considérant $A = 1$

DE \ BC	00	01	11	10
00	0_{16}	0_{17}	0_{19}	0_{18}
01	0_{20}	1_{21}	1_{23}	0_{22}
11	0_{28}	0_{29}	0_{31}	0_{28}
10	1_{24}	1_{25}	1_{27}	1_{26}

Figure 1

$$D\overline{E} + C\overline{D} \cdot E + AB \cdot \overline{C}$$

Karnaugh-map de 6 entrées

Note :

On peut utiliser la même approches que la méthode pour entrée, cette fois en considérant **4** K-map de 4 variables superposées dans un cube tridimensionnel Exemple sur `Youtube`

Multiplexeur

- 2^n lignes d'entrées
- 2^n N lignes de sélections
- 2^n Une seule sorties Y

Possède deux implémentations secondaires, soit (1) **portes logiques** et (2) **tampons** à trois états.