

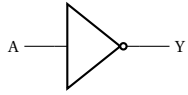
Architecture des ordinateurs  
IFT1227  
**Introduction**

Franz Girardin

18 janvier 2024

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

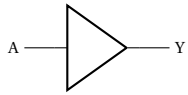
### Porte NON



A	Y
1	0
0	1

$$Y = \bar{A}$$

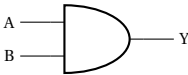
### Porte BUF



A	Y
0	0
1	1

$$Y = A$$

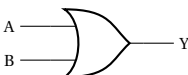
### Porte ET



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

$$Y = AB$$

### Porte OU



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

$$Y = A + B$$

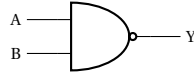
### Porte XOR



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

$$Y = A \oplus B$$

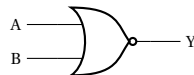
### Porte NAND



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

$$Y = \overline{AB}$$

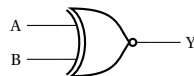
### Porte NOR



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

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

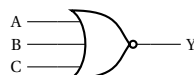
### Porte XNOR



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

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

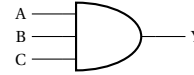
### Porte NOR3



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

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

### 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

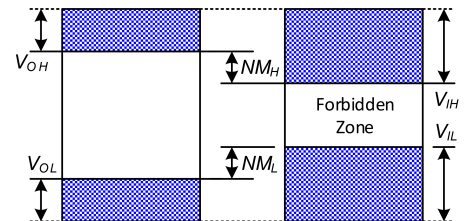
$$Y = ABC$$

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

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

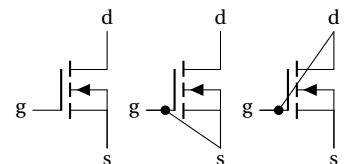
$$NM_H = V_{OH}(\text{ém.}) - V_{IH}(\text{s.r.c.})$$

$$NM_L = V_{IL}(\text{s.r.c.}) - V_{OL}(\text{ém.})$$



### 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  $\Rightarrow$  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
$B = 0 \text{ if } B \neq 1$	$B = 1 \text{ if } B \neq 0$	Binary 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
$0 \cdot 1 \cdot 0 = 0$	$1 + 0 = 1 + 1 = 1$	AND/OR

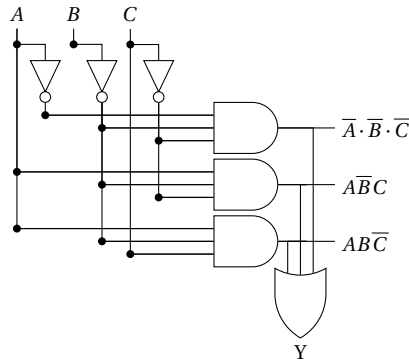
Théorème	Dual	Nom
$B \cdot 1 = B$	$B + 0 = B$	Identité
$B \cdot 0 = 0$	$B + 1 = 1$	Élément nul
$B \cdot B = B$	$B + B = B$	Idempotence
	$\overline{\overline{B}} = B$	Involution
$B \cdot \overline{B} = 0$	$B + \overline{B} = 1$	Complément

## De Morgan

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

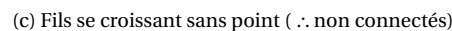
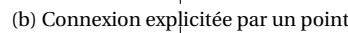
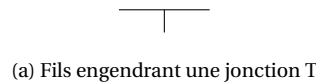
$$\neg(A \cdot B) = \neg A + \neg B$$

## Exemple simple de circuit combinatoire

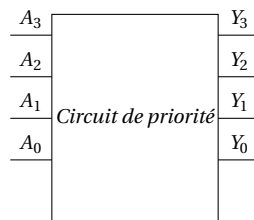


$$Y = \overline{A} \cdot \overline{B} \cdot \overline{C} + A\overline{B}C + AB\overline{C}$$

**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.

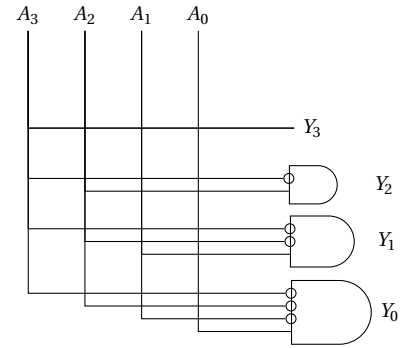


## Circuit de priorité



A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	Y <sub>3</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>
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

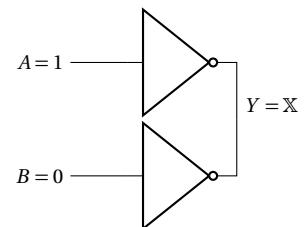
## Représentation d'un circuit de priorité



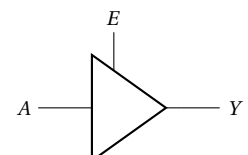
**Entrées don't care ou X** Ces entrées sont utilisées pour spécifier que la variables possédant le *don't care* n'affecte pas le résultat de la fonction logique. Une file de priorité peut être résumée par la table suivante.

A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	Y <sub>3</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>
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 Z** Circuit dans lequel une entrée *E* est connecté à une porte tampon et contrôle la **propagation du signal**. Lorsque l'entrée *E* est sous tension haute, la porte agit comme un tampon normal; lorsque l'entrée *E* est sous tension basse, la porte produit le signal *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

		AB			
		00	01	11	10
C	0	1	0	0	0
	1	1	0	0	0

		AB			
		00	01	11	10
C	0	$\overline{A} \cdot \overline{B} \cdot \overline{C}$	0	0	0
	1	$\overline{A} \cdot \overline{B} C$	0	0	0

## 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.