Cours de Systèmes Logiques 1

Portes logiques

Etienne Messerli & Yann Thoma

Reconfigurable and Embedded Digital Systems Institute
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License

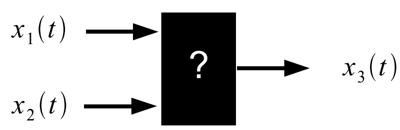
Septembre 2019

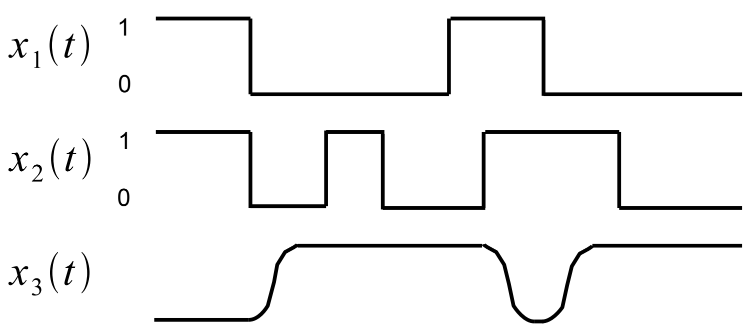
Polycopié: Electronique numérique

- Portes logiques et algèbre de Boole
  - chapitre 4, pages 35 à 54
- Circuits logiques combinatoires
  - Simplification, tables de Karnaugh
  - chapitres 5-1 à 5-6, pages 55 à 64
- Symboles utilisés
  - chapitre 4-9, pages 46 et 47

Expérience

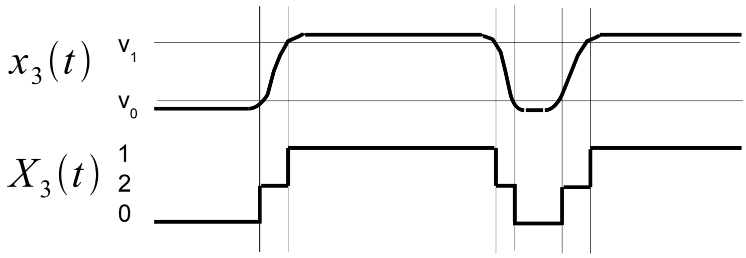
- Soient x_1, x_2 et x_3 des signaux électroniques





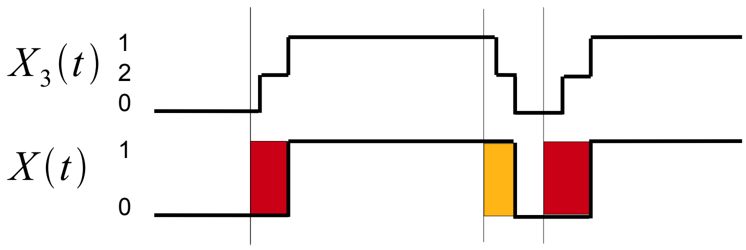
Hypothèse 1: Quantification

$$
\begin{aligned}
X_3(t) &= 0 \; si \; x_3(t) \leq v_0 \\
X_3(t) &= 1 \; si \; x_3(t) \geq v_1 \\
X_3(t) &= 2 \; si \; v_0 < x_3(t) < v_1
\end{aligned}
$$



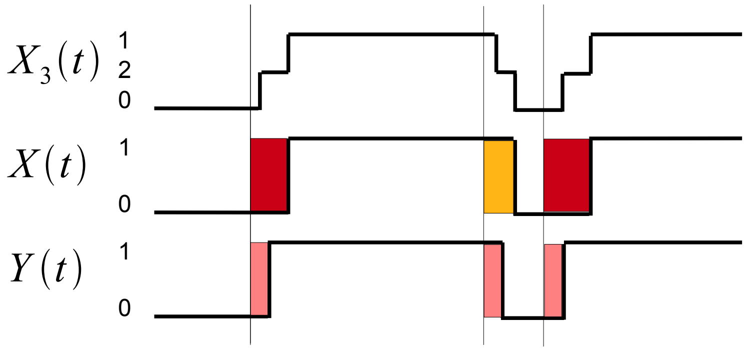
Hypothèse 2: Délais

- Hypothèses: Durée de l'état '2' faible
- Plus que 2 états ('0' et '1')
- Apparition d'un *délai*, ou *retard*, de durée différente...



Hypothèse 3: Délais constants

- Hypothèse : les délais sont tous de durées identiques. Cette durée est constante et indépendante du temps (ne fluctue pas).

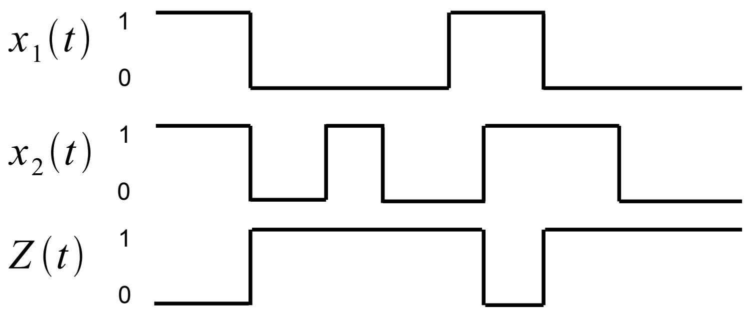


Modèle logique asynchrone

- En combinant les hypothèses
  - de quantification,
  - d'élimination des transitoires, et
  - d'égalisation des délais,
- nous obtenons le **modèle logique asynchrone**.

Modèle logique combinatoire

- En rajoutant que les délais sont nuls, nous obtenons le modèle logique combinatoire.

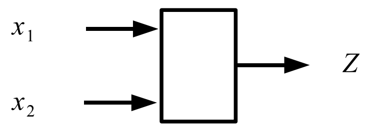


Etats

- Etats d'entrée
  - Chacune des $2^2 = 4$ combinaisons possibles des valeurs des deux entrées x_1 et x_2 est appelé état d'entrée.
  - Chacun de ces états sera symbolisé par (x_1, x_2).
  - De façon générale, chacune des 2^n combinaisons des valeurs de n entrées $x_1, x_2, \cdots, x_n$ est un état d'entrée $(x_1, x_2, \cdots, x_n)$.
- Etats de sortie
  - Par analogie, chacune des 2^r combinaisons des valeurs de r sorties est un état de sortie $(Z_1, Z_2, Z_3, \cdots, Z_r)$

Elément combinatoire

- Elément combinatoire : système idéal dont l'état de sortie Z est entièrement déterminé par son état d'entrée (x_1, x_2), en tout temps.



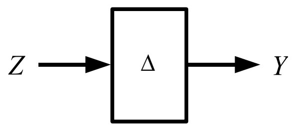
Fonction logique

- La fonction logique d'un élément combinatoire est l'opération réalisée par celui-ci. Elle peut être définie par des équations ou des tables de vérité.
- Ici, la table de vérité utilisée pour les précédents chronogrammes:

x_1	x_2	Z
0	0	1
0	1	1
1	0	1
1	1	0

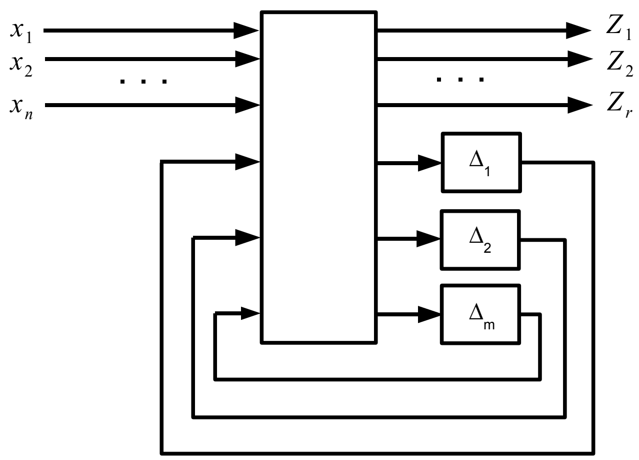
Elément de délai

- Par définition, il existe un élément idéal caractérisé par un délai ou retard constant : Δ.



Systèmes logiques asynchrones

- L'usage d'éléments de délai est traité dans le cadre des systèmes logiques asynchrones.

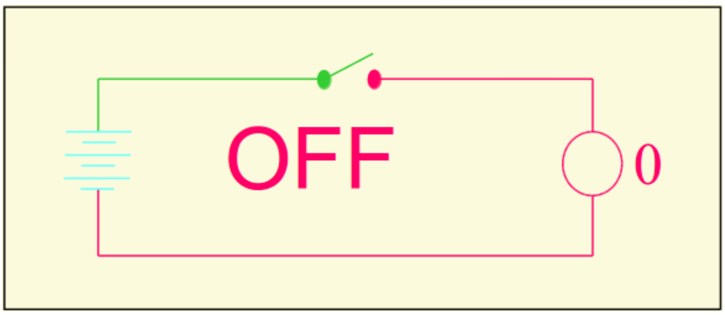


Systèmes logiques combinatoires

- Définition :
  - L'assemblage d'éléments combinatoires sans éléments de délai est appelé : *système logique combinatoire*.

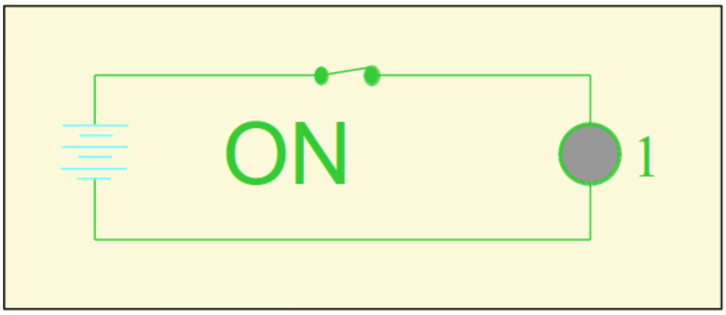
Principe de la logique (corollaire)

- Être logique, c'est avoir une réponse unique sans contradiction :
  - Pas d'Affirmation et de Négation en même temps !!!
  - Pas de Vrai et de Faux en même temps !!!
    - Une lampe ne peut jamais être Allumée (ON) et Eteinte (OFF) en même temps



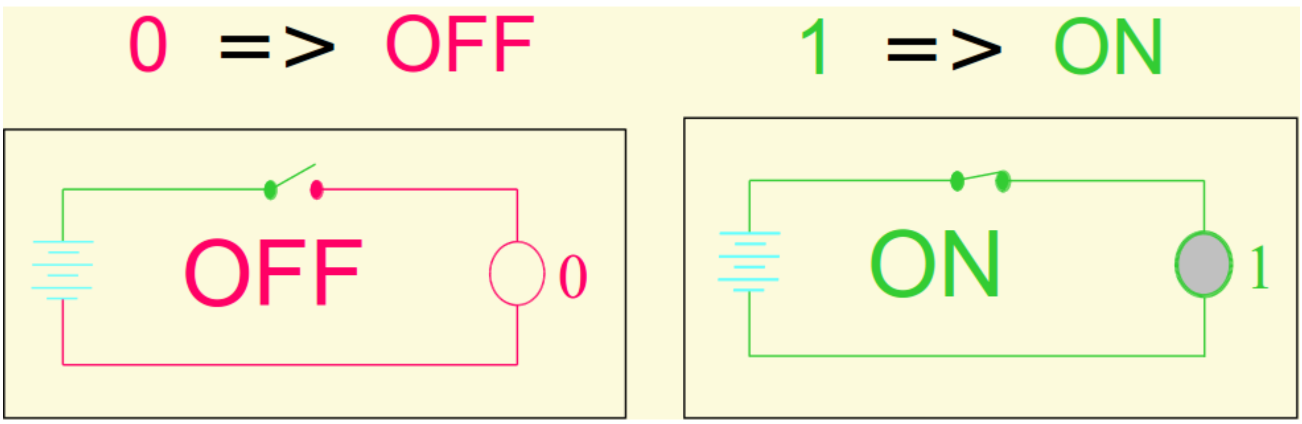
Principe de la logique (corollaire)

- Être logique, c'est avoir une réponse unique sans contradiction :
  - Pas d'Affirmation et de Négation en même temps !!!
  - Pas de Vrai et de Faux en même temps !!!
    - Une lampe ne peut jamais être Allumée (ON) et Eteinte (OFF) en même temps



Principe de la logique (corollaire)

- On voit clairement une *variable binaire*:



Système logique

- C'est un système qui traite l'information de façon logique
- Pour étudier un système logique, il faut connaître les fonctions de base (les composants) et le langage mathématique qui permet de décrire un comportement sous forme d'équations
- Pour un additionneur:

$$Z = f(X, Y)$$

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

Définitions

Etat Logique

Chacune des 2 valeurs que peut prendre une variable logique

Variable logique

Grandeur qui ne peut prendre que les 2 états logiques

Variable d'entrée (ou simplement entrée)

Information à 2 états reçue par un système logique

Variable de sortie (ou simplement sortie)

Information à 2 états générée par un système logique

Fonction logique

Relation logique entre une sortie et une ou plusieurs entrées

Types de systèmes logiques

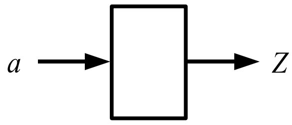
- Système combinatoire :
  - la valeur des sorties à un moment donné dépend uniquement des valeurs des entrées à cet instant
  - ⇒ Système univoque
  - le comportement est entièrement spécifié par une table, nommée table de vérité, qui pour chaque combinaison des entrées donne la valeur des sorties
  - pour n entrées, la table de vérité comporte 2^n lignes
  - la sortie est immédiate
- Système séquentiel :
  - la valeur des sorties dépend de la valeur des entrées au cours du temps: il faut une mémoire
  - ⇒ L'état du système dépend de l'historique
  - l'obtention d'un résultat peut demander plusieurs étapes, avec mémorisation de résultats intermédiaires

Les portes logiques de base

- Etude des fonctions d'une variable :
  - Nous verrons la porte logique de base :
    - NON
- Etude des fonctions de deux variables :
  - Nous verrons les portes logiques de base :
    - ET
    - OU
  - Nous verrons ensuite des combinaisons :
    - NON-ET
    - NON-OU
    - OU-Exclusif

Fonction d'une variable

- Système combinatoire le plus simple :
  1 entrée , 1 sortie.



- Enumération de toutes ses fonctions logiques possibles :

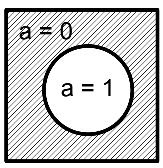
a	Z_0	Z_1	Z_2	Z_3
0	0	1	0	1
1	0	0	1	1

$$\begin{aligned} Z_0 &= 0 \text{ (constante logique)} \\ Z_3 &= 1 \text{ (constante logique)} \\ Z_1 &= a \text{ (identité ou forme vraie)} \\ Z_2 &= \overline{a} \text{ (complément, négation, inverse)} \end{aligned}$$

Fonction NON

- $Z_2 = \overline{a}$ (prononcer "a barre")

a	not a
0	1
1	0



Théorème d'involution

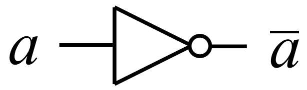
$$\overline{(\overline{a})} = \overline{\overline{a}} = a$$

$$\overline{\overline{\overline{a}}} = \overline{a}$$

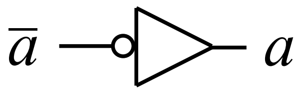
Définition

- On appelle : *porte NON* ou *opérateur NON* ou *inverseur* tout système combinatoire qui réalise la fonction NON.
- Tout système combinatoire réalisant une fonction x sera dorénavent appelé *porte x* ou *opérateur x*.

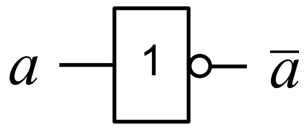
Logigramme



MIL (standard américain)



MIL



CEI (Commission Electrotechnique Internationale)

Fonctions de deux variables

- On dénombre $2^2 = 4$ états d'entrée (a, b) pour une fonction à deux variables.
- On peut alors définir 2^4 fonctions Z distinctes.
- La table de vérité de l'ensemble des fonctions de deux variables est:

a	b	Z_0	Z_1	Z_2	Z_3	Z_4	Z_5	Z_6	Z_7	Z_8	Z_9	Z_{10}	Z_{11}	Z_{12}	Z_{13}	Z_{14}	Z_{15}
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

- On peut noter que $Z_i = \overline{Z_{15-i}}$

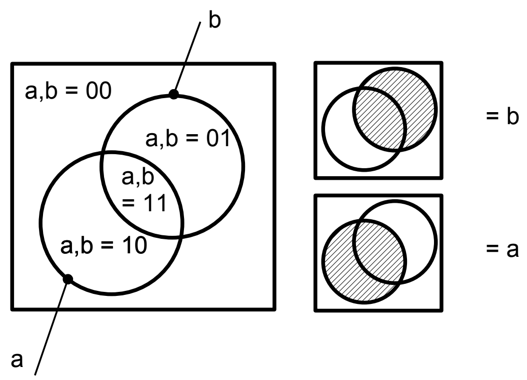
Commentaire

- Une fonction à n variables possède 2^n états d'entrées différents
- Et il est possible de définir 2^{2^n} fonctions à n variables.
- A titre indicatif, nous avons:

n	2^n	2^{2^n}
3	8	256
4	16	$65'536$
5	32	$4'294'967'296$

Diagramme de Venn

- Fonction à deux variables



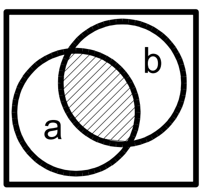
Fonction ET

$$Z_1 = a \cdot b = ab$$

- Z_1 est le *produit logique* ou *intersection* de a et b.
- La fonction Z_1 est appelée *fonction ET*, car Z_1 vaut 1 si a ET b valent **simultanément** 1.

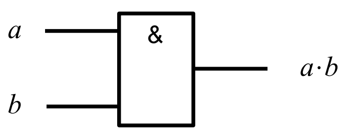
Fonction ET (2)

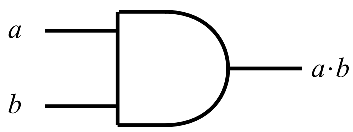
$$Z_1 = a \cdot b = ab$$



$Z_1 = a \cdot b$

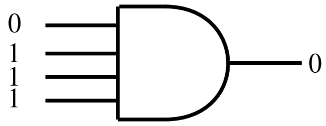
a	b	a et b
0	0	0
0	1	0
1	0	0
1	1	1

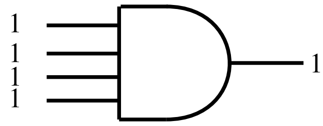




Fonction ET à n variables

- La définition peut être étendue à n variables:
  - La sortie d'une porte ET à n variables vaut 1 si TOUTES les entrées valent 1.





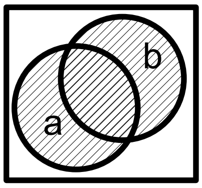
Fonction OU

$$Z_7 = a + b$$

- Z_7 est la *somme logique* ou *réunion* ou encore *union* de a et b.
- La fonction Z_7 est appelée *fonction OU*, car Z_7 vaut 1 si a OU b valent 1 (l'une ou l'autre des variables ou simultanément).
- Attention à ne pas confondre avec le *soit* de la langue française.

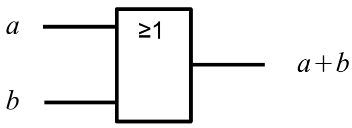
Fonction OU (2)

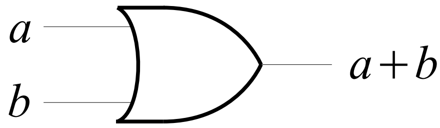
$$Z_7 = a + b$$



$Z_7 = a + b$

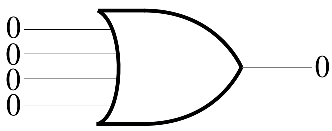
a	b	a ou b
0	0	0
0	1	1
1	0	1
1	1	1

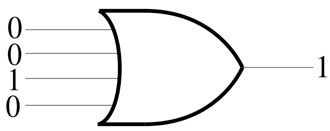




Fonction OU à n variables

- La définition peut être étendue à n variables:
  - La sortie d'une porte OU à n variables vaut 1 si AU MOINS UNE des entrées vaut 1.





Théorèmes de commutativité

$$
\begin{aligned}
a \cdot b &= b \cdot a \\
a + b &= b + a
\end{aligned}
$$

Théorèmes d'idempotence

$$
\begin{aligned}
a \cdot a &= a \\
a + a &= a
\end{aligned}
$$

Théorèmes d'idempotence (2)

- Forme généralisée

$$\begin{aligned} a \cdot a \cdot a \cdot \dots \cdot a &= a \\ a + a + a + \dots + a &= a \end{aligned}$$

Théorème des constantes

$$\begin{aligned} a \cdot 0 &= 0 \\ a \cdot 1 &= a \\ a + 0 &= a \\ a + 1 &= 1 \end{aligned}$$

- Démonstration tabulaire : on impose $b = 0$ ou $b = 1$ dans la table de Z_1 et Z_7

Théorèmes de complémentation

$$
\begin{aligned}
a \cdot \overline{a} &= 0 \\
a + \overline{a} &= 1
\end{aligned}
$$

Précédence des opérateurs

- Le ET a la précédence sur le OU

$$a \cdot b + b \cdot d = (a \cdot b) + (b \cdot d) \neq a \cdot (b + b) \cdot d$$

Théorème de distributivité

- On affirme que la fonction ET est distributive par rapport à la fonction OU.

$$a \cdot (b + c) = a \cdot b + a \cdot c$$

- De même, la fonction OU est distributive par rapport à la fonction ET.

$$a + b \cdot c = (a + b) \cdot (a + c)$$

- Démonstration tabulaire

Exercice

- Démontrer la relation suivante:

$$a + \overline{a} \cdot b = a + b$$

Théorèmes d'associativité

$$\begin{aligned} a\cdot(b\cdot c) &= (a\cdot b)\cdot c = a\cdot b\cdot c \\ a+(b+c) &= (a+b)+c = a+b+c \end{aligned}$$

- Démonstration tabulaire

Théorèmes du consensus

$$\begin{aligned} a\cdot x + b\cdot\overline{x} + a\cdot b &= a\cdot x + b\cdot\overline{x} \\ (a+x)\cdot(b+\overline{x})\cdot(a+b) &= (a+x)\cdot(b+\overline{x}) \end{aligned}$$

- Démonstration en séance d'exercices

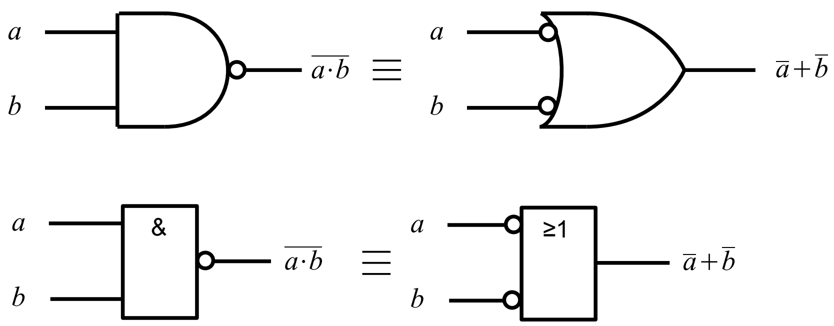
Fonction NON ET

$$Z_{14} = a \uparrow b = \overline{a \cdot b} = \overline{a} + \overline{b}$$

- La fonction Z_{14} est appelée *fonction NON ET*, NAND en anglais, car Z_{14} vaut 0 si a ET b valent **simultanément** 1.

Fonction NON ET (2)

$$Z_{14} = a \uparrow b = \overline{a \cdot b} = \overline{a} + \overline{b}$$



a	b	a nand b
0	0	1
0	1	1
1	0	1
1	1	0

Opérateur complet

- On appelle OPERATEUR COMPLET tout opérateur (porte) dont trois assemblages distincts permettent de réaliser les trois fonctions NON, ET et OU.
- Ainsi, suivant cette definition, la porte NAND est un OPERATEUR COMPLET.

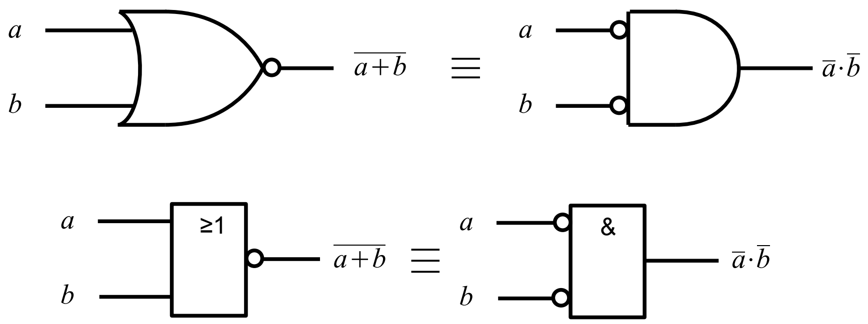
Fonction NON OU

$$Z_8 = a \downarrow b = \overline{a+b} = \overline{a} \cdot \overline{b}$$

- La fonction Z_8 est appelée *fonction NON OU*, NOR en anglais, car Z_8 vaut 0 si a ET b valent **simultanément** 0.

Fonction NON OU (2)

$$Z_8 = a \downarrow b = \overline{a + b} = \overline{a} \cdot \overline{b}$$



a	b	a nor b
0	0	1
0	1	0
1	0	0
1	1	0

Théorèmes de De Morgan

Théorèmes de De Morgan

$$\begin{aligned} \overline{a \cdot b} &= \overline{a} + \overline{b} \\ \overline{a + b} &= \overline{a} \cdot \overline{b} \end{aligned}$$

- Démonstration tabulaire

Théorèmes de De Morgan (Corollaires)

Théorèmes de De Morgan (Corollaires)

$$
\begin{aligned}
a \cdot b &= \overline{\overline{a} + \overline{b}} \\
a + b &= \overline{\overline{a} \cdot \overline{b}}
\end{aligned}
$$

- Démonstration algébrique

Définition

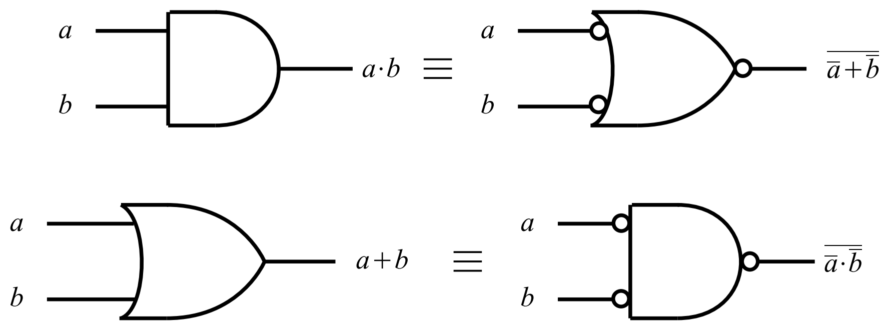
- Un produit de plusieurs variables apparaissant chacune sous la forme vraie ou inversée est un *monôme*.
- Une somme de produits (ou monômes) est un polynôme.
- Exemples de monômes: $a \cdot \overline{b}$, $a \cdot b \cdot c$
- Exemple de polynôme: $a \cdot \overline{b} + a \cdot b \cdot c$

Exemple

- Grâce à De Morgan, écrivez la fonction suivante sous la forme d'un polynôme:

$$Z = \overline{a + b + \overline{c} \cdot d}$$

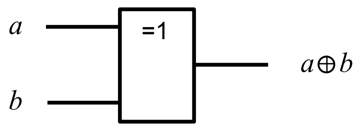
Transformation de logigrammes

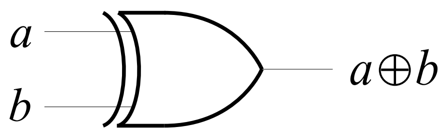


Fonction OU-exclusif

$$Z_6 = a \oplus b = \overline{a} \cdot b + a \cdot \overline{b}$$

- La fonction Z_6 est appelée *fonction OU-exclusif*, XOR en anglais, car Z_{14} vaut 0 si soit a soit b vaut 1.





a	b	a xor b	$\overline{a \text{ xor } b}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

Fonction OU-exclusif: théorèmes

$$\begin{aligned}
a \oplus b &= b \oplus a \\
a \oplus a &= 0 \\
a \oplus 0 &= a \\
a \oplus 1 &= \overline{a} \\
a \oplus \overline{a} &= 1 \\
\overline{a \oplus b} &= \overline{a} \cdot \overline{b} + a \cdot b = a \oplus \overline{b} = \overline{a} \oplus b
\end{aligned}$$