IUT de Colmar - Département GTR - 1ière année.

La Logique Combinatoire:

Laurent MURA.

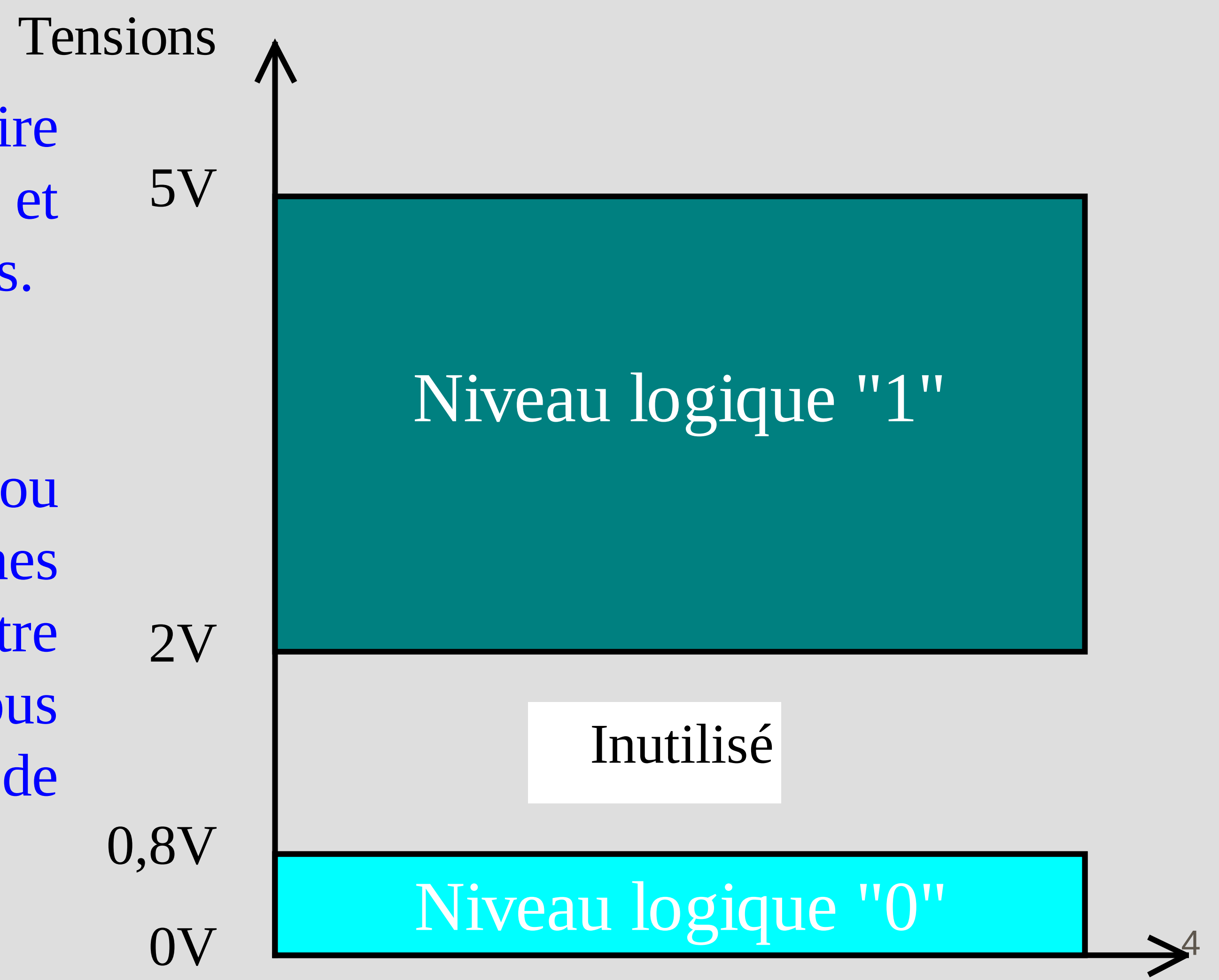
SOMMAIRE:

1. Introduction:

Le système binaire: Les constantes et variables booléennes:

•Le système binaire utilise les variables et constantes booléennes.

•Les variables ou constantes booléennes peuvent être représentées sous forme de plage de tensions:



Algèbre de Boole: Définition (1):

•L'algèbre de Boole ne concerne que des éléments (variables ou constantes booléennes) pouvant prendre les valeurs 0 et 1.

•Ces éléments servent souvent à représenter des tensions sur des fils (niveaux logiques) ou des conditions logiques (vrai ou faux):

Niveau logique 0	**Niveau logique 1**
Faux	Vrai
Arrêt	Marche
Bas	Haut
Non	Oui
Ouvert	Fermé

Algèbre de Boole: Définition (2):

- Il n'y a que deux valeurs possibles.
- En algèbre booléenne il n'y a:
  - ni fraction,
  - ni partie décimale,
  - ni nombre négatif,
  - ni racine carrée,
  - ni logarithme,
  - ni nombre complexe,
  - ni etc.…
- En fait, dans cette algèbre on ne retrouve que les trois opérations élémentaires recensées dans le tableau suivant:

L 'algèbre de Boole: Ces 3 opérations élémentaires:

Dénomination	**Opération**	**Symbole**
addition logique	OU	+
multiplication logique	ET	.
complémentation ou inversion logique	NON	-

La Logique Combinatoire: Définition:

•La logique combinatoire est la logique des systèmes numériques qui sont indépendants du temps.

•Les sorties de tels systèmes ne dépendent que de l'état des entrées.

•**Les tables de vérité:**

•Une table de vérité lie les entrées d'un système numérique à sa ou ses sorties.

•Elle représente les différentes combinaisons logiques du fonctionnement du système ou circuit.

Les tables de vérité: Exemples:

Entrées		Sortie
A	**B**	**X**
0	0	?
0	1	?
1	0	?
1	1	?

Entrées			Sortie
A	**B**	**C**	**X**
0	0	0	?
0	0	1	?
0	1	0	?
0	1	1	?
1	0	0	?
1	0	1	?
1	1	0	?
1	1	1	?

2. Les fonctions logiques élémentaires:

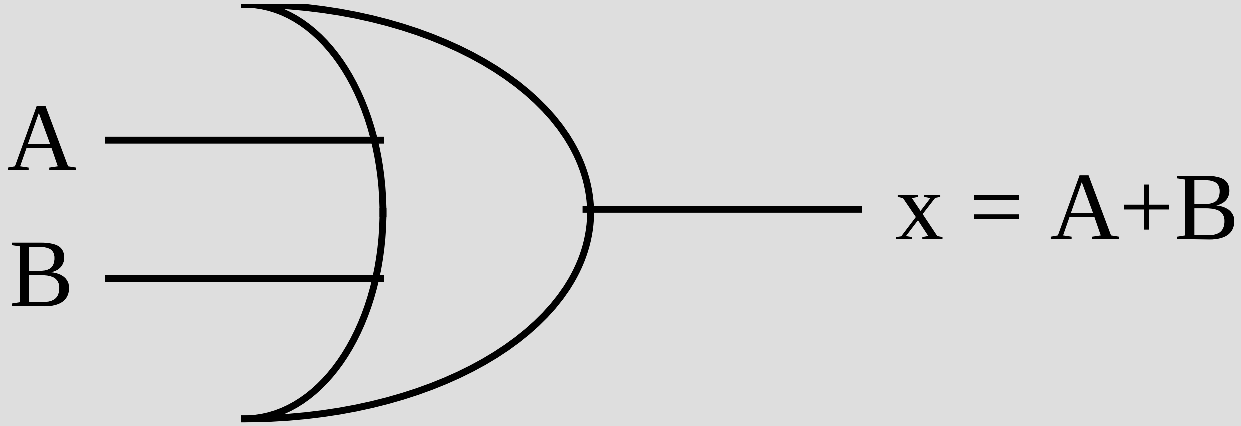
La fonction OU (+): Sa table de vérité:

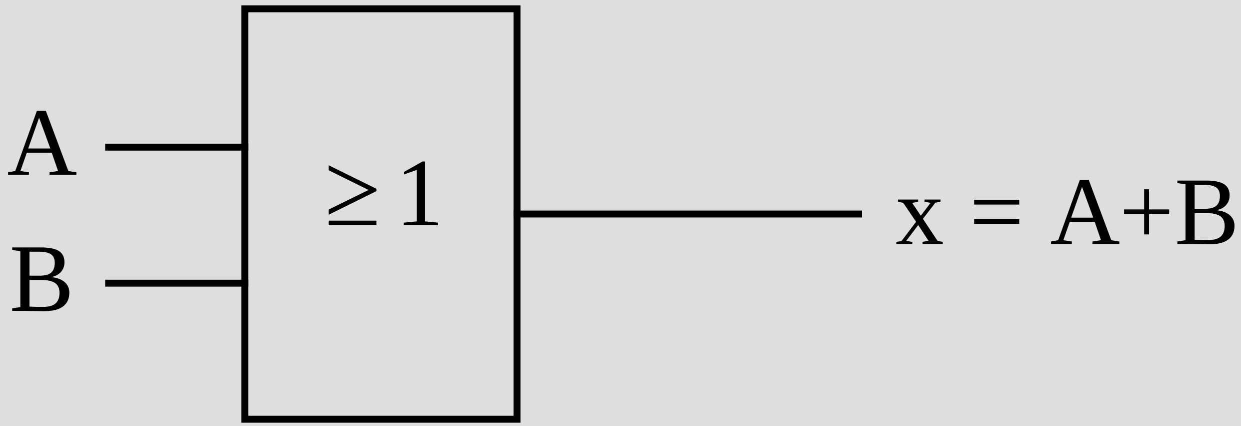
•La fonction OU réalise une ADDITION LOGIQUE:

A	B	X = A+B
0	0	0
0	1	1
1	0	1
1	1	1

•On peut conclure de cette table de vérité que l'opération OU donne un résultat vrai dès que l'une des composantes de l'opération est vraie.

La fonction OU (+): Sa porte logique:

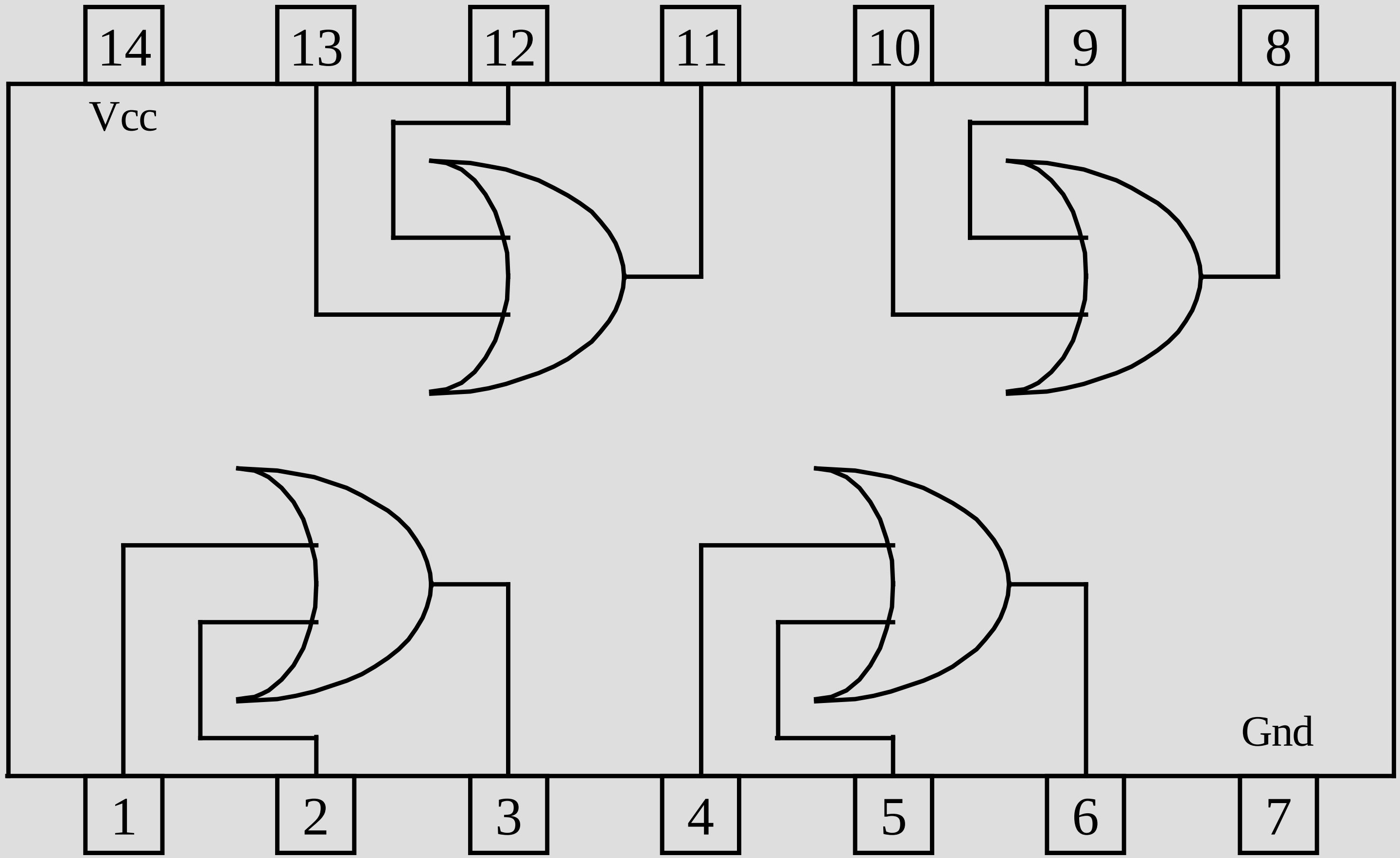




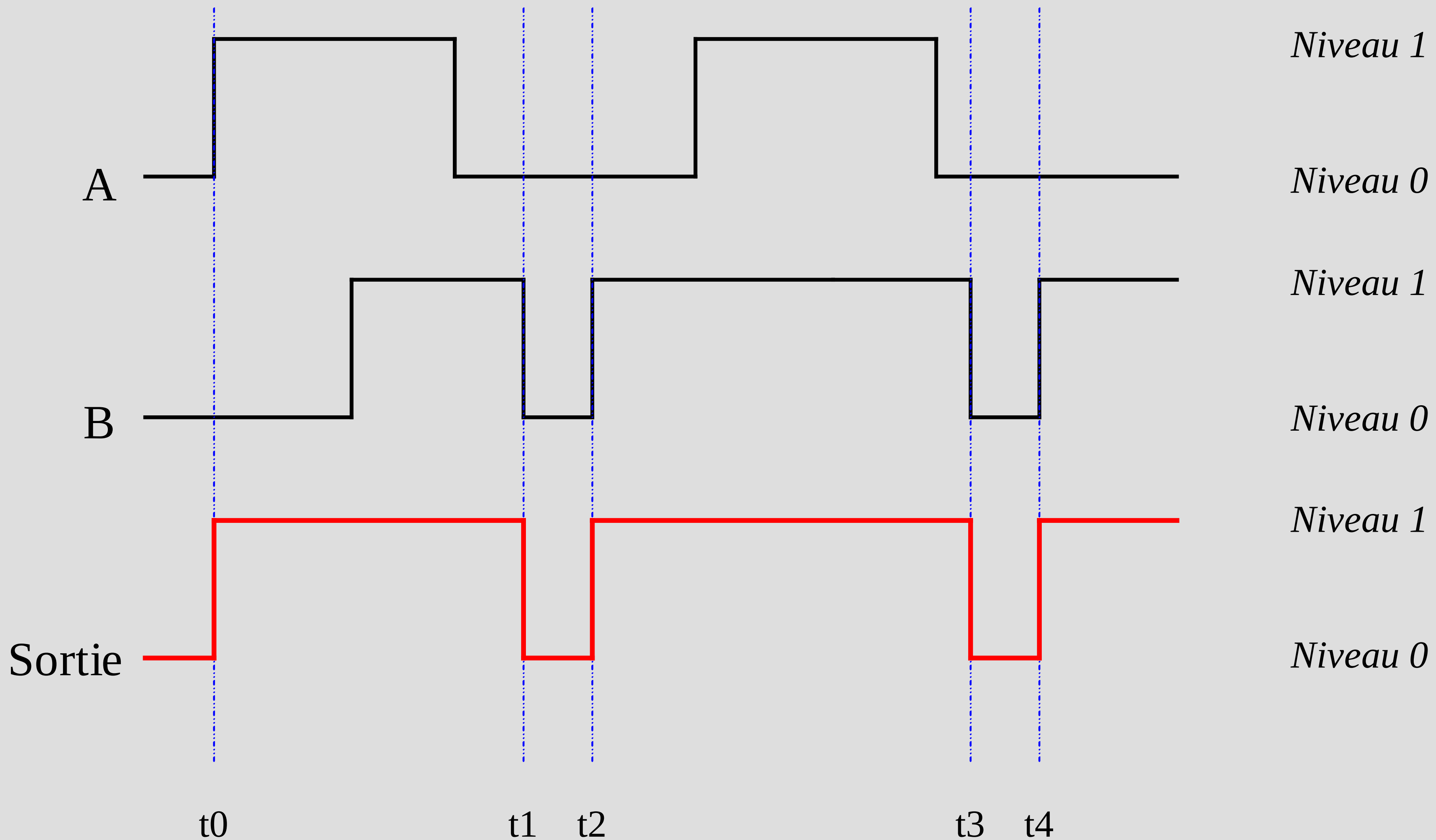
•La sortie d'une porte OU est à un niveau haut dès que l'une des entrées prend un niveau haut (quelque soit le nombre d'entrées).

La fonction OU (+): Composant intégrant 4 portes OU à 2 entrées:

7432



La fonction OU (+): Chronogramme:



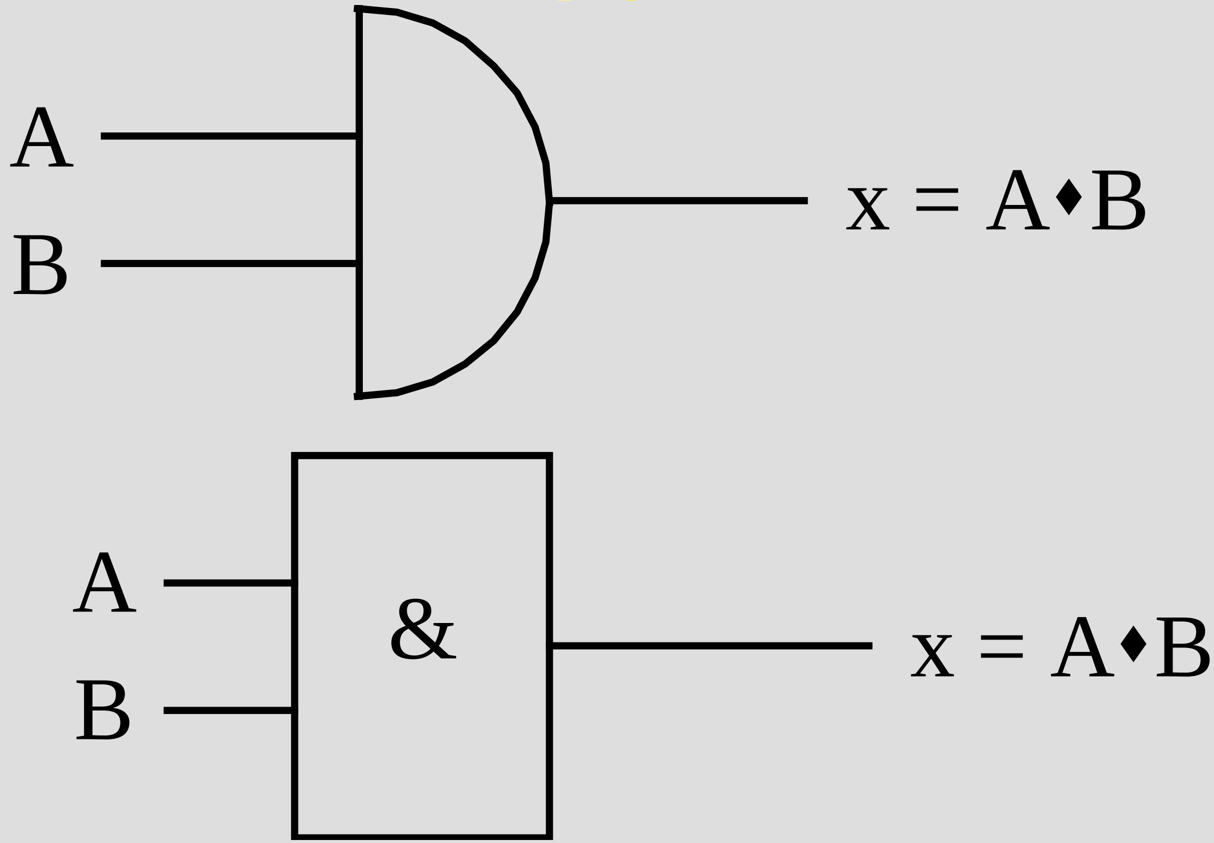
La fonction ET (.): Sa table de vérité:

•La fonction ET réalise une MULTIPLICATION LOGIQUE:

A	B	X = A.B
0	0	0
0	1	0
1	0	0
1	1	1

•On peut conclure de cette table de vérité que l'opération ET donne un résultat vrai si et seulement si toutes les composantes de l'opération sont vraies.

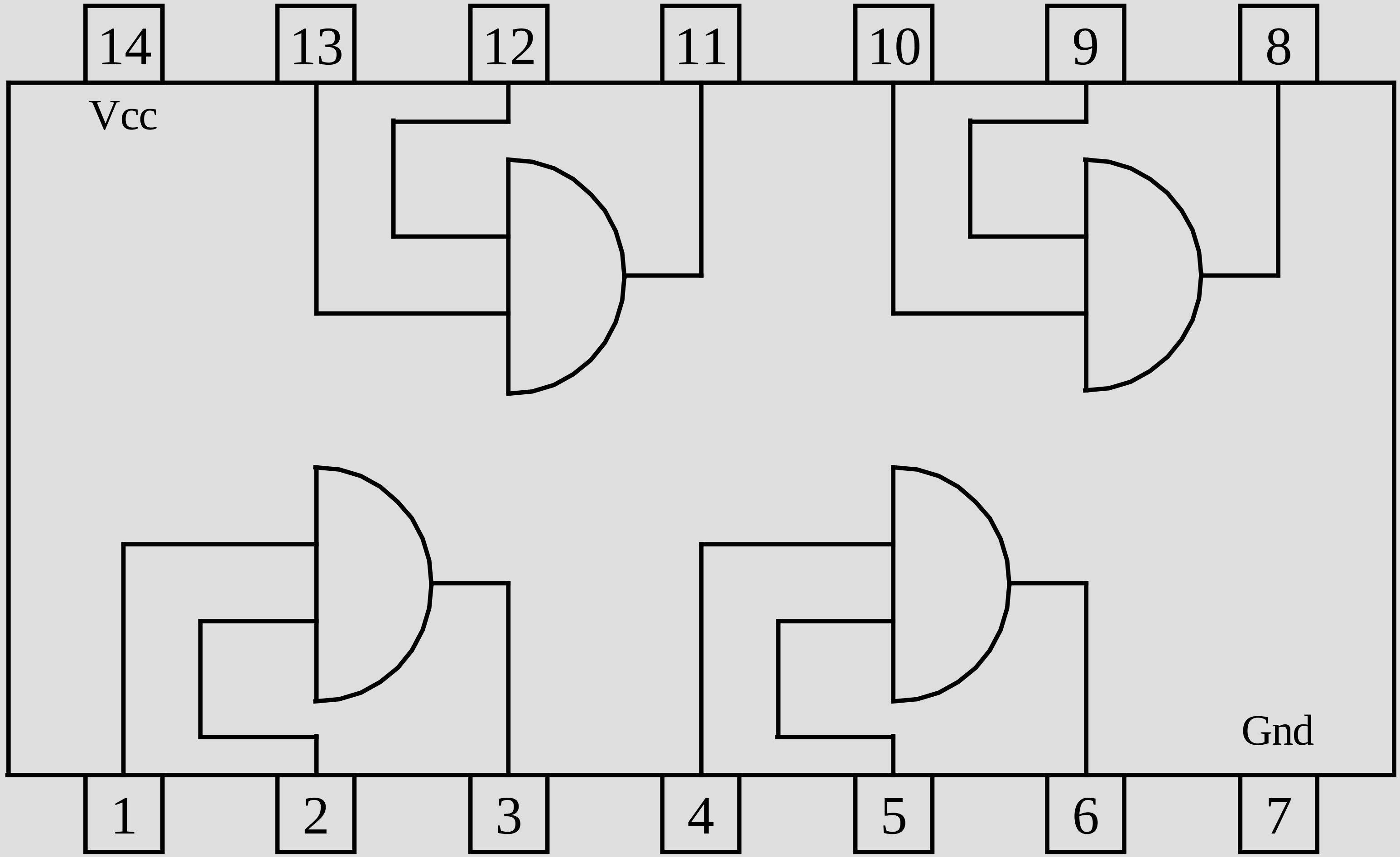
La fonction ET (.): Sa porte logique:



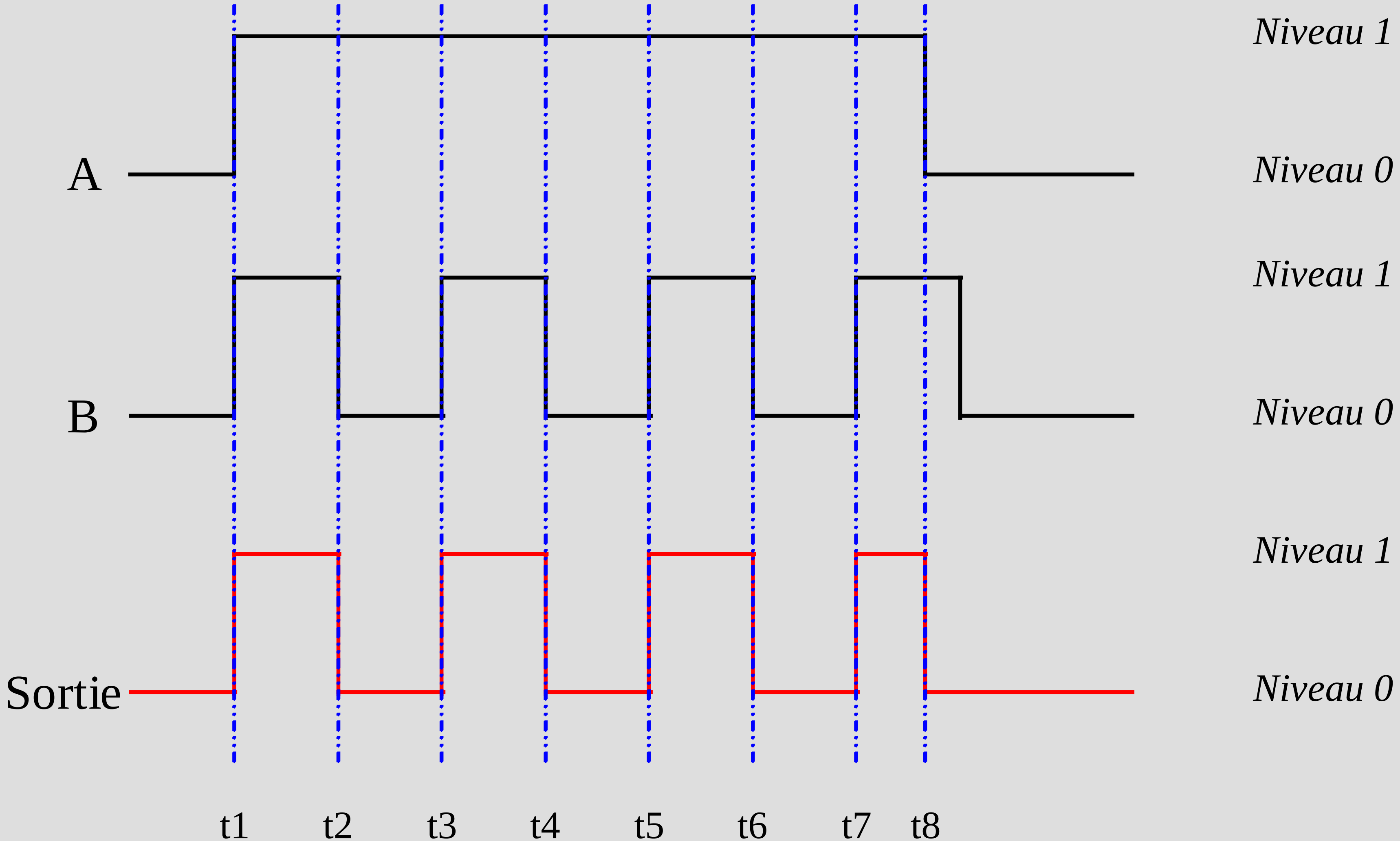
•La sortie d'une porte ET est à un niveau haut si et seulement si toutes les entrées sont à un niveau haut (quelque soit le nombre d'entrées).

La fonction ET (.): Composant intégrant 4 portes ET à 2 entrées:

7408



La fonction ET (.): Chronogramme:

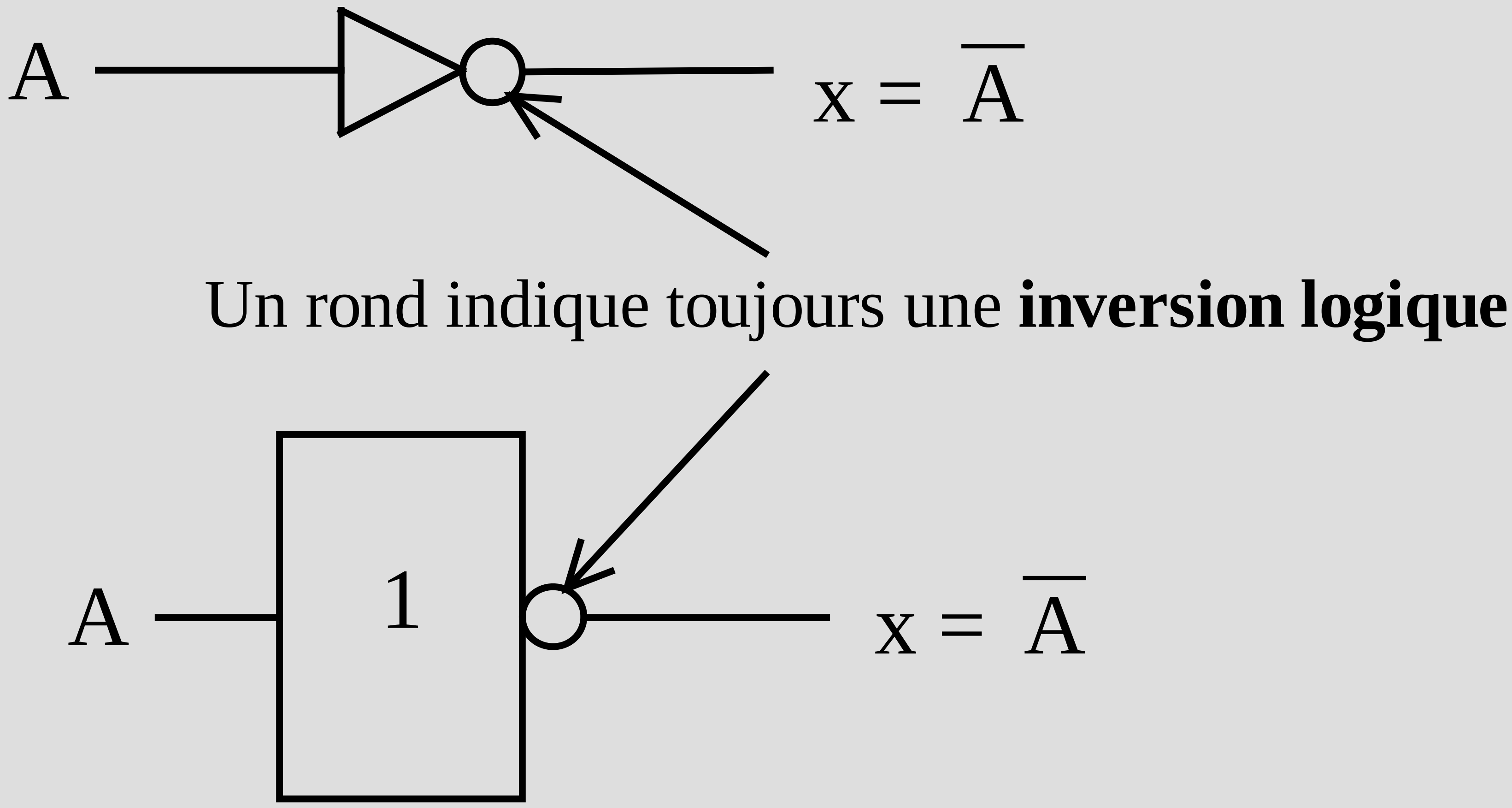


La fonction NON (¯): Sa table de vérité:

•La fonction NON réalise une INVERSION ou COMPLEMENTATION LOGIQUE:

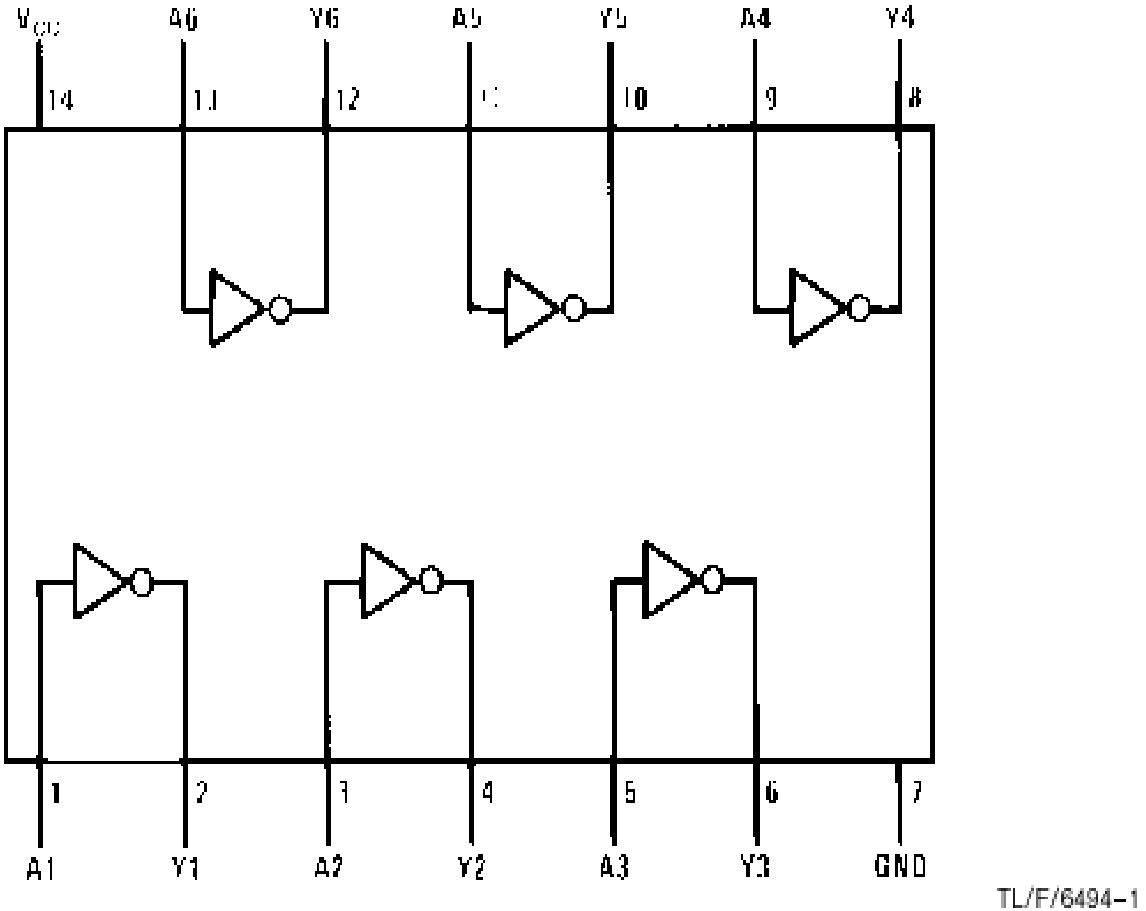
A	**X =** $\overline{A}$
0	1
1	0

La fonction NON (‾): Son symbole logique:



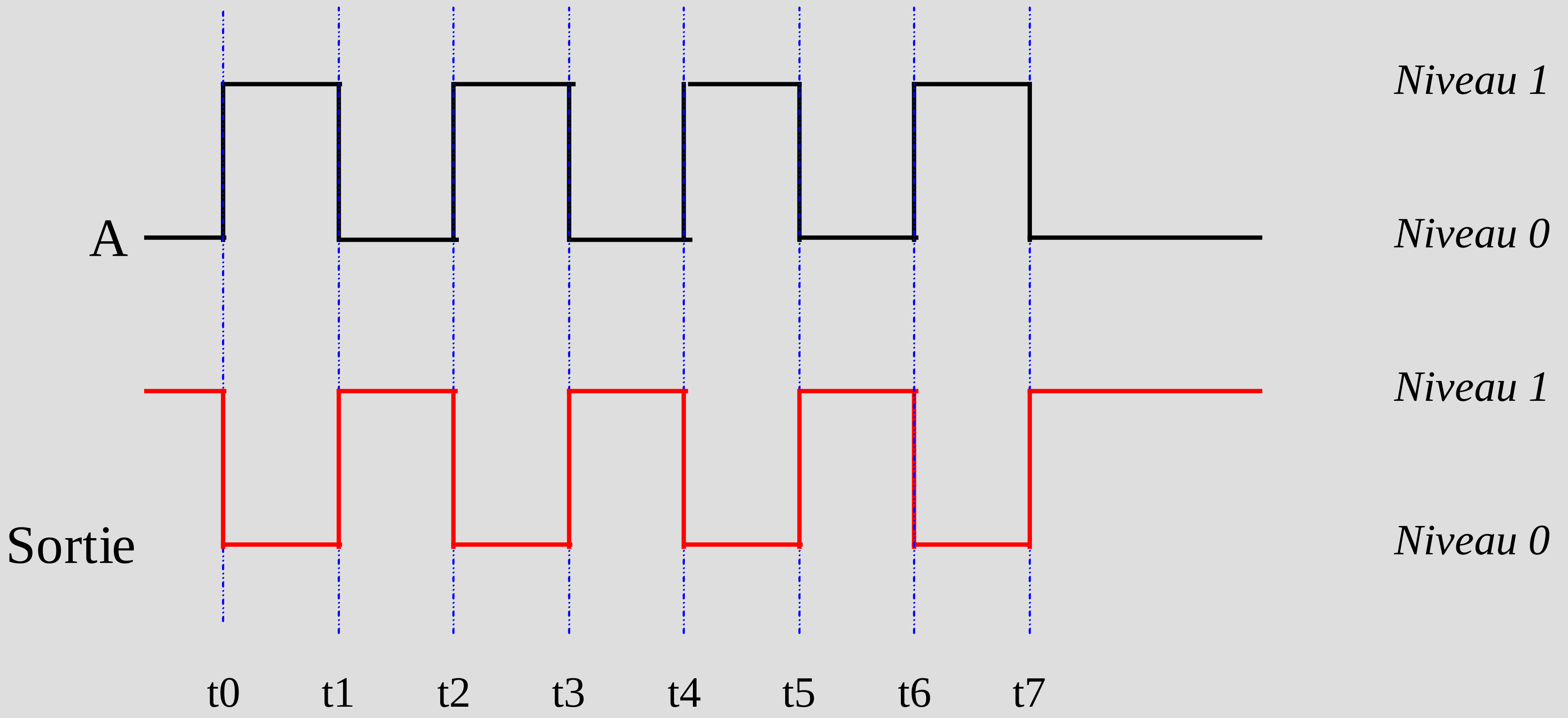
•La sortie d'un inverseur est à un niveau haut si l'entrée est à un niveau bas et vice-versa.

La fonction NON (¯): Composant intégrant 6 portes NON à 1 entrée:



Order Number 5404DMQB, 5404FMQB, DM5404J, DM5404W, DM7404M or DM7404N
See NS Package Number J14A, M14A, N14A or W14B

La fonction NON (‾): Chronogramme:



3. La forme algébrique:

Mise sous forme algébrique des circuits logiques: Définition et Priorité:

•Tout circuit logique, quelque soit sa complexité, peut être mis sous forme d'équations booléennes avec les fonctions de base:

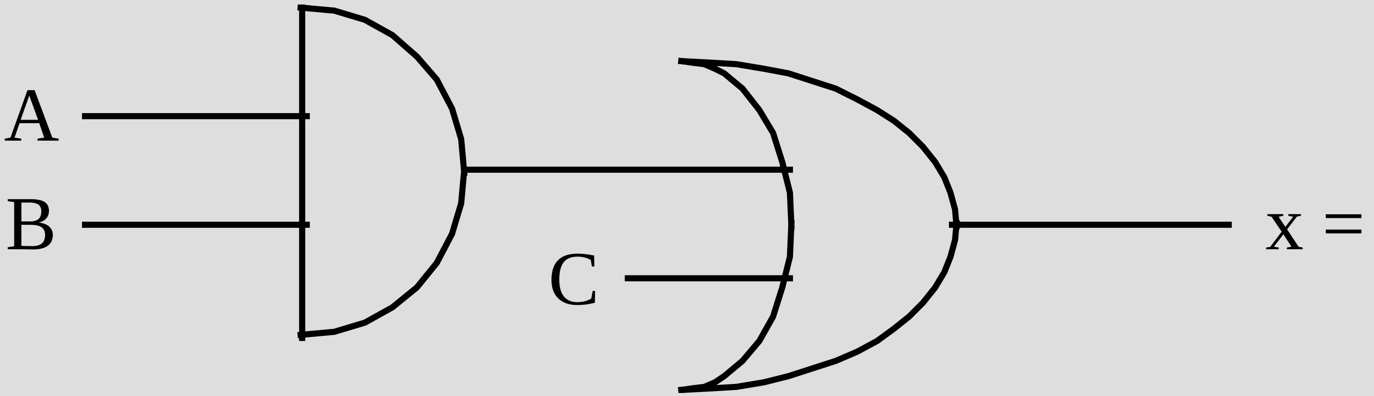
•OU (+),

•ET (.),

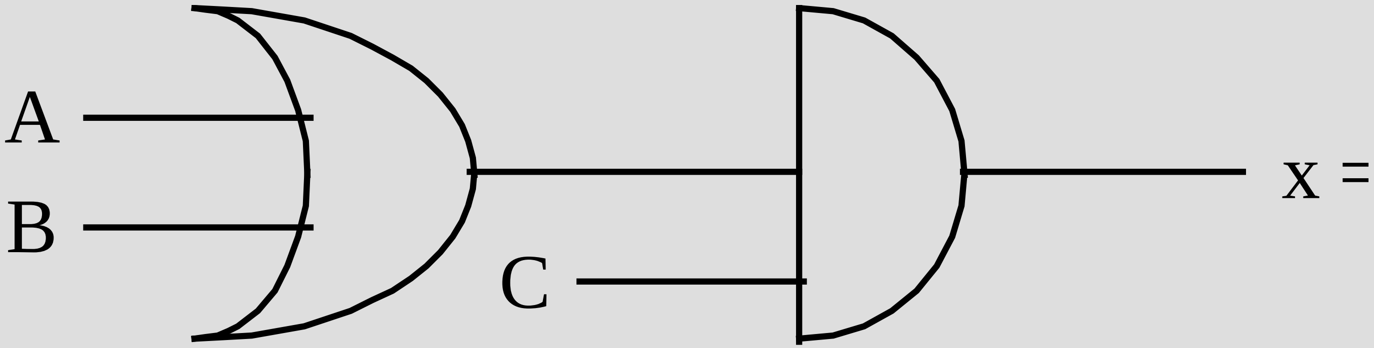
•NON ($\overline{\ }$).

•La fonction **ET** est prioritaire par rapport à la fonction **OU**.

Mise sous forme algébrique des circuits logiques: Exemples:



$X = (A.B) + C = A.B + C$

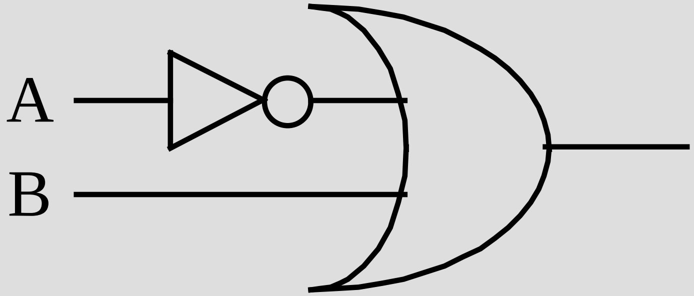


$X = (A+B).C$

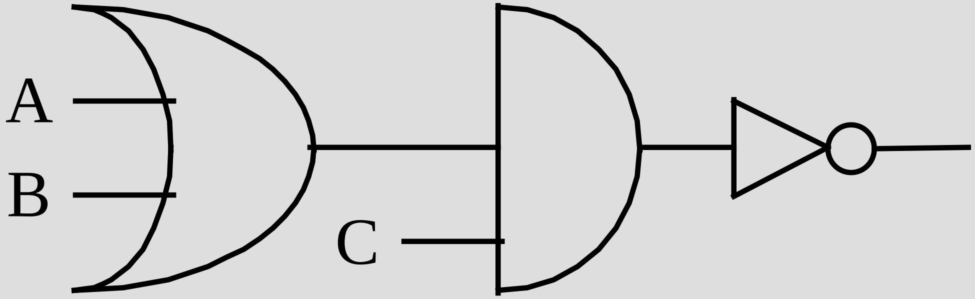
Priorité de la fonction Inversion :

•Chaque fois que se trouve un inverseur dans un circuit logique, son entrée est simplement surmontée d'un trait (complémentée).

•La complémentation est moins prioritaire que les fonctions ET et OU.

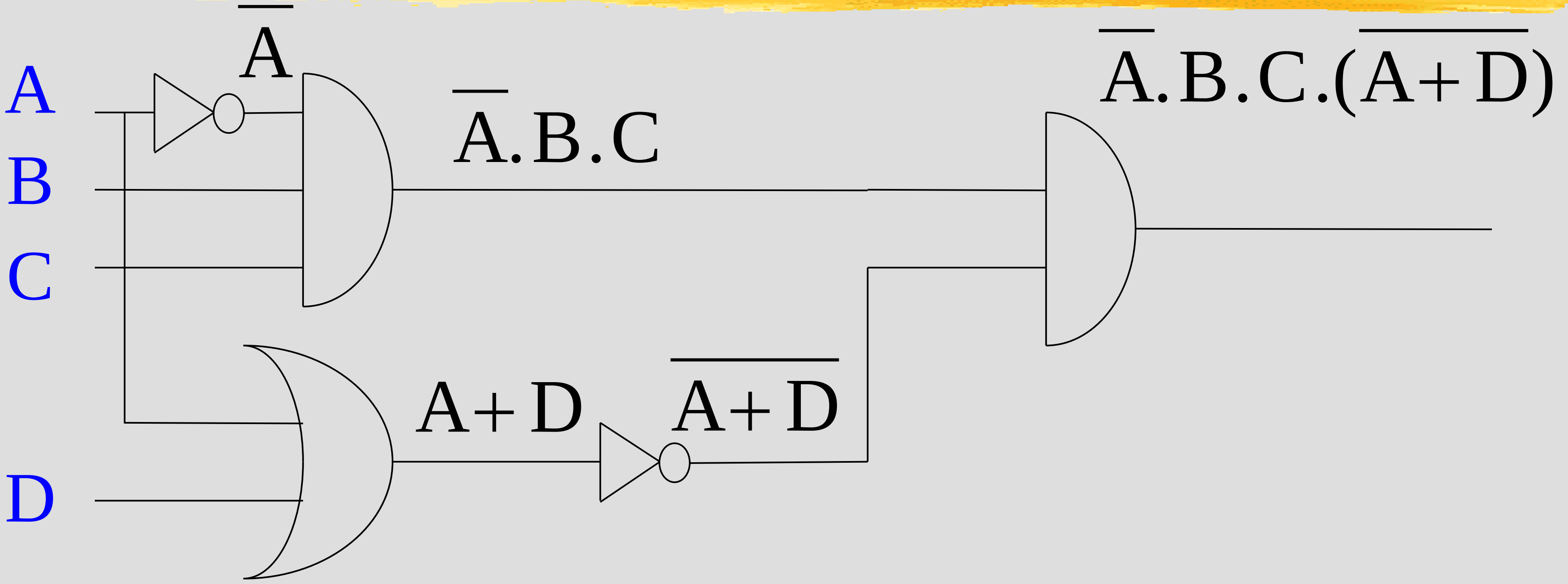


$$x = \overline{A} + B$$



$$x = \overline{(A + B) \bullet C}$$

Traduction circuits => équation (1):



•ATTENTION: $\overline{A+D} \neq \overline{A} + \overline{D}$

Traduction circuits => équation (2):

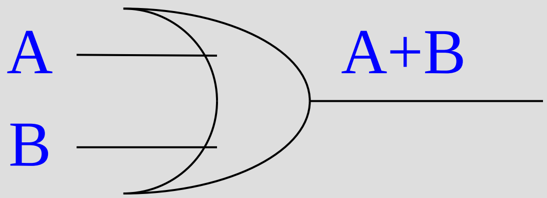
- On cherche x pour:
  - A=0,
  - B=1,
  - C=1,
  - D=1

$$
\begin{aligned}
x &= \overline{A} \bullet B \bullet C \bullet (\overline{A + D}) \\
&= \overline{0} \bullet 1 \bullet 1 \bullet (\overline{0 + 1}) \\
&= 1 \bullet 1 \bullet 1 \bullet (\overline{1}) \\
&= 1 \bullet 1 \bullet 1 \bullet 0 \\
&= 0
\end{aligned}
$$

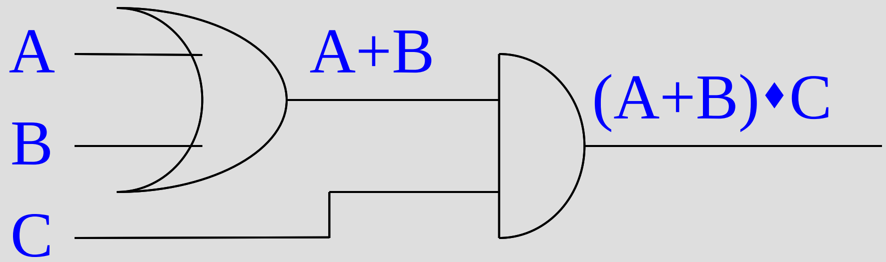
Traduction équation => circuits:

$$x = \overline{(A+B)\bullet C}$$

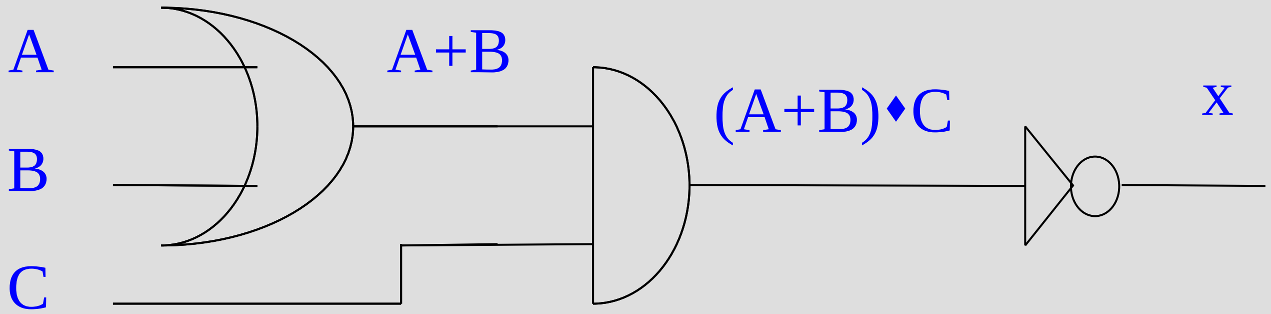
Etape 1



Etape 2



Etape 3



4. Fonctions logiques OU-NON et ET-NON:

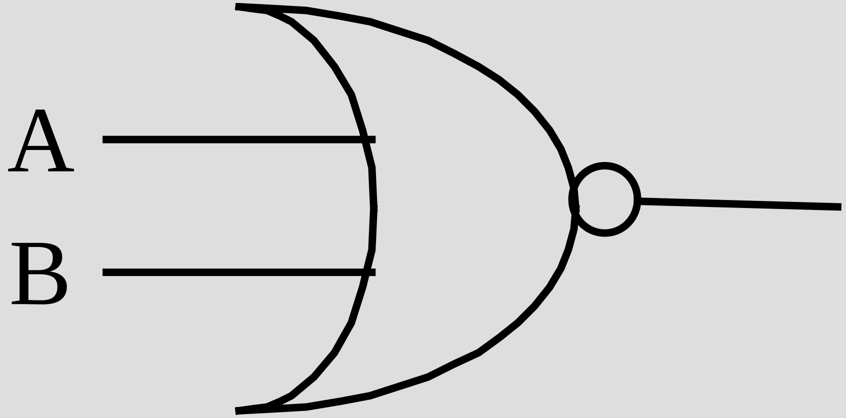
La fonction OU-NON (NOR): Sa table de vérité:

•La fonction NOR est la conjonction d'une fonction OU et d'une fonction NON:

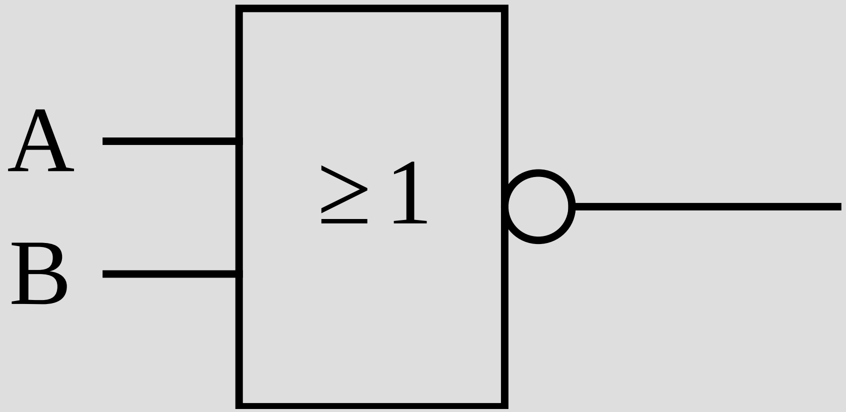
A	**B**	**X =** $\overline{A + B}$
0	0	1
0	1	0
1	0	0
1	1	0

•On peut conclure de cette table de vérité que la porte NOR donne un résultat vrai *si et seulement si* les deux entrées sont fausses.

La fonction NOR: Sa porte logique:



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

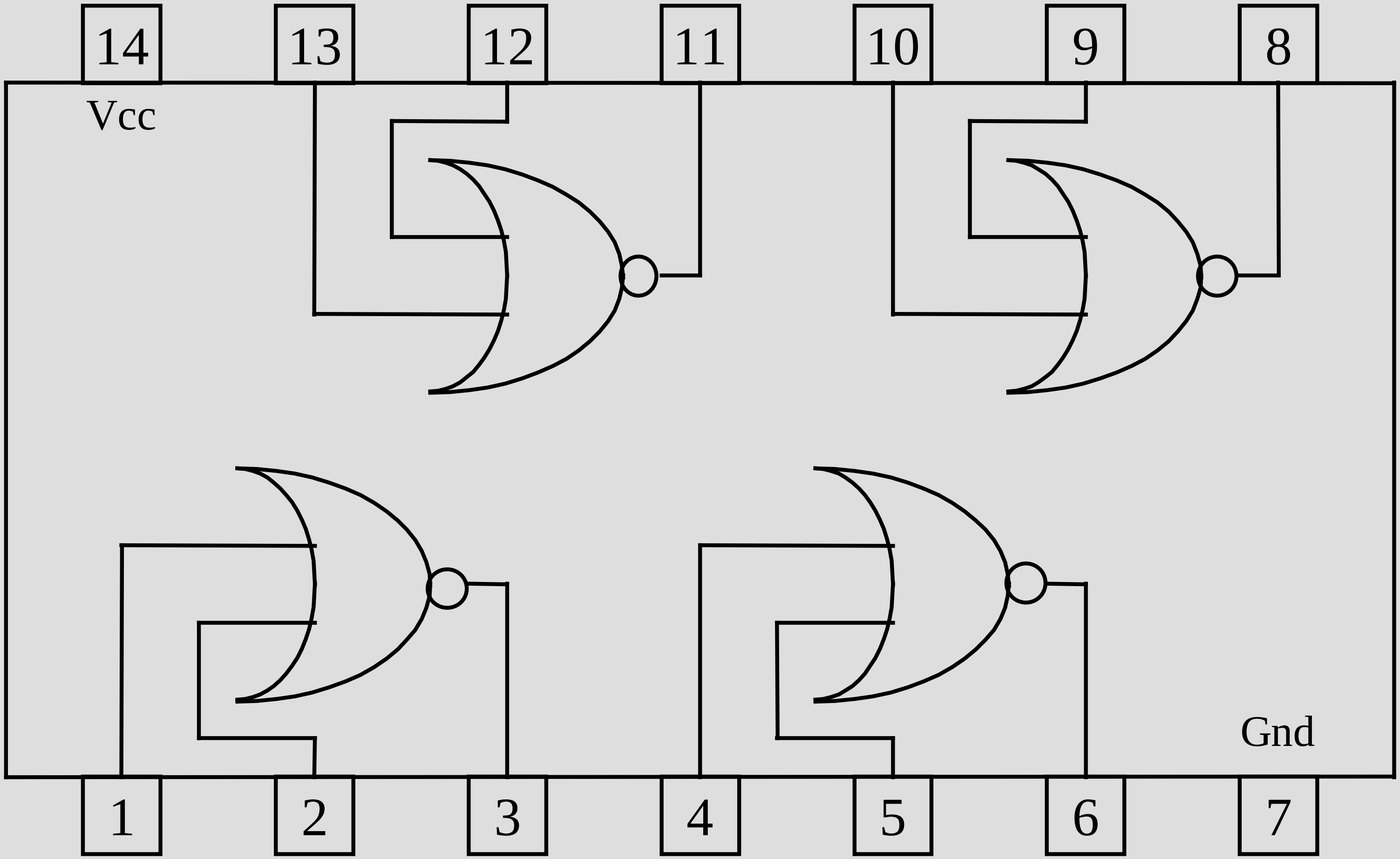


$$x = \overline{A + B}$$

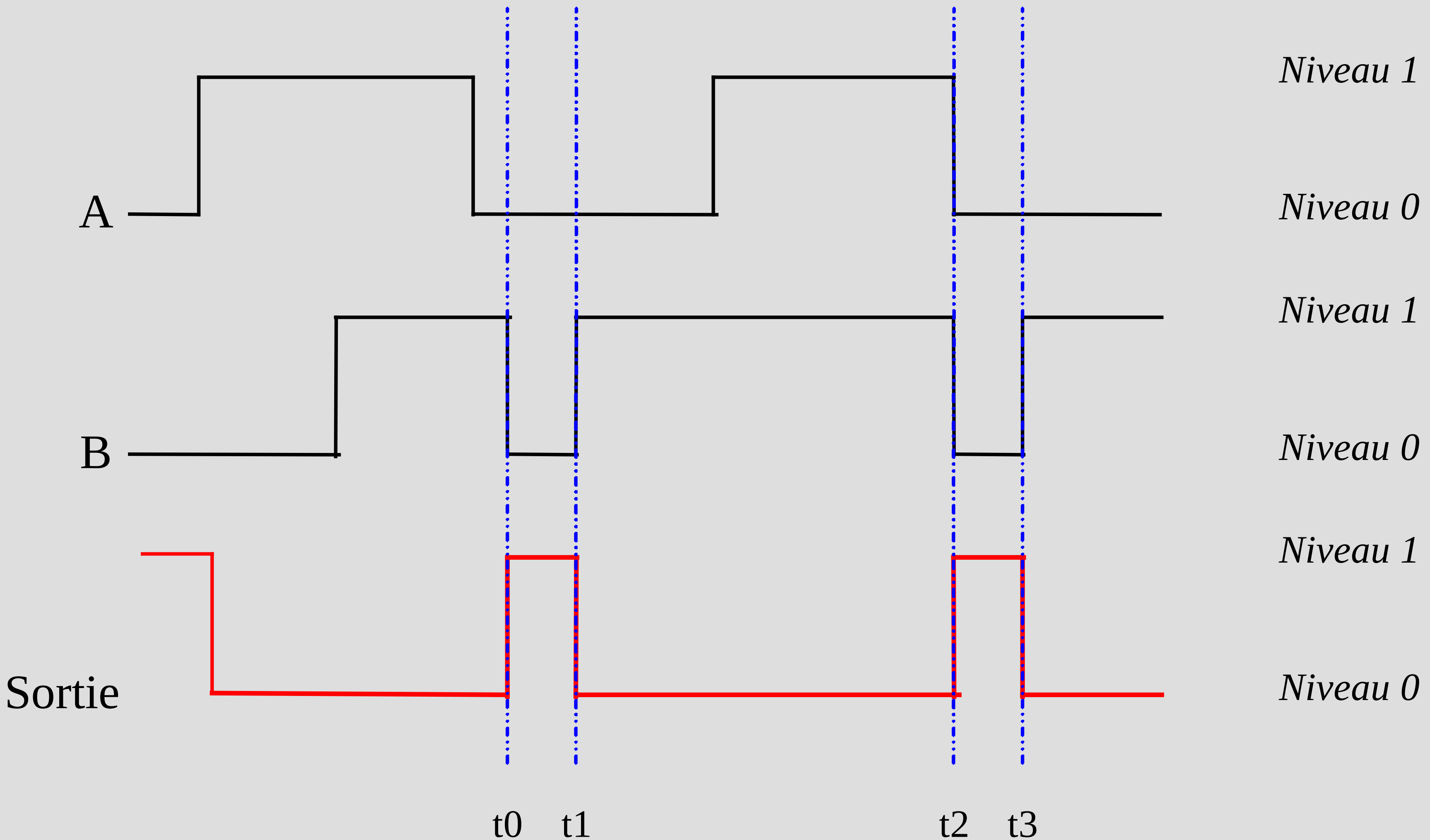
•La sortie d'une porte NOR est à un niveau haut seulement si les deux entrées ont un niveau bas (quelque soit le nombre d'entrées).

La fonction NOR: Composant intégrant 4 portes NOR à 2 entrées:

7436
ou
7402
(brochage
différent)



La fonction NOR: Chronogramme:



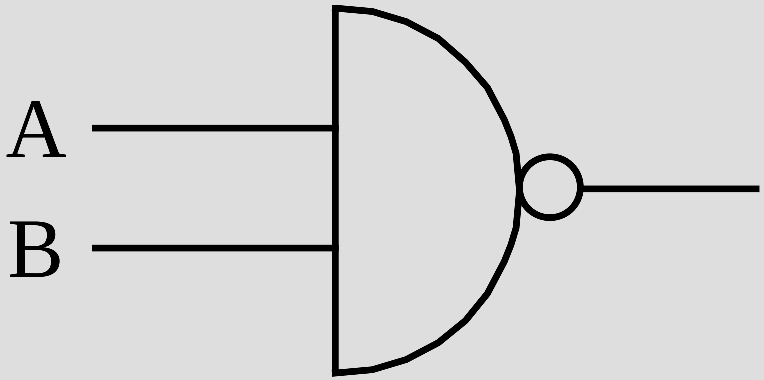
La fonction ET-NON (NAND): Sa table de vérité:

•La fonction NAND est la conjonction d'une fonction ET et d'une fonction NON:

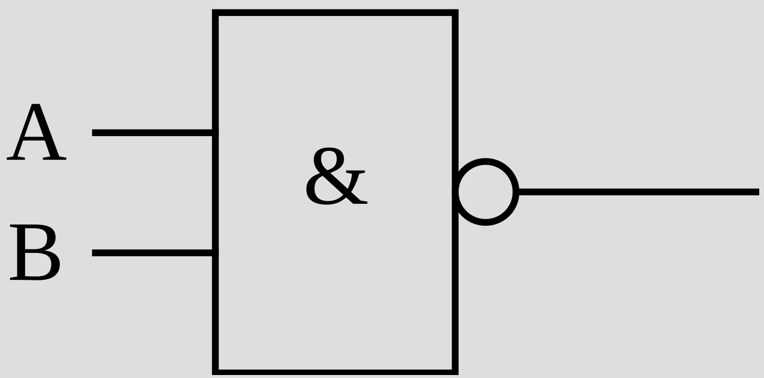
A	B	$X = \overline{A \bullet B}$
0	0	1
0	1	1
1	0	1
1	1	0

•On peut conclure de cette table de vérité que la porte NAND donne un résultat vrai dès que l'une des entrées est fausse. Le résultat devient faux uniquement si les deux entrées sont vraies.

La fonction NAND: Sa porte logique:



$$x = \overline{A \bullet B}$$

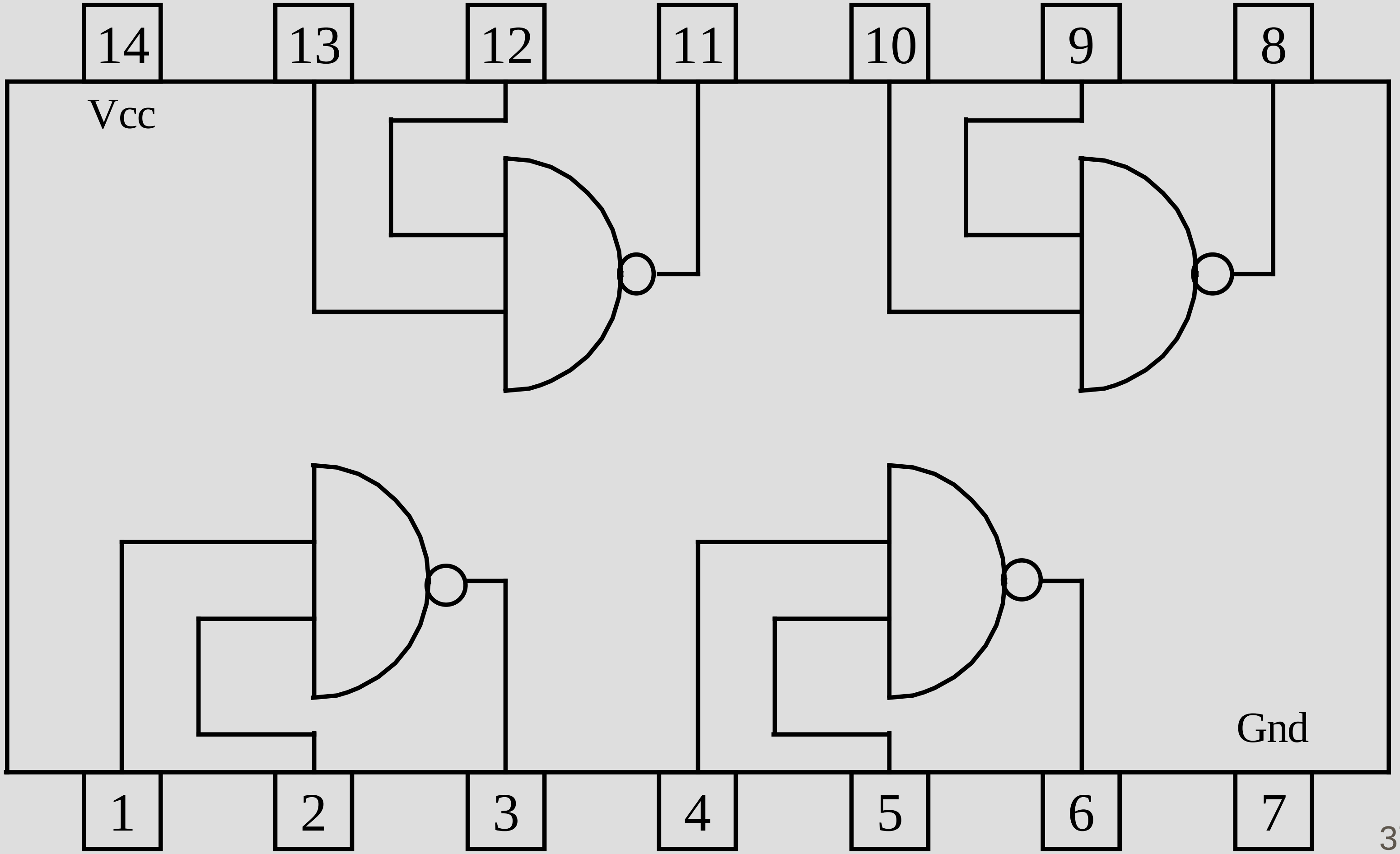


$$x = \overline{A \bullet B}$$

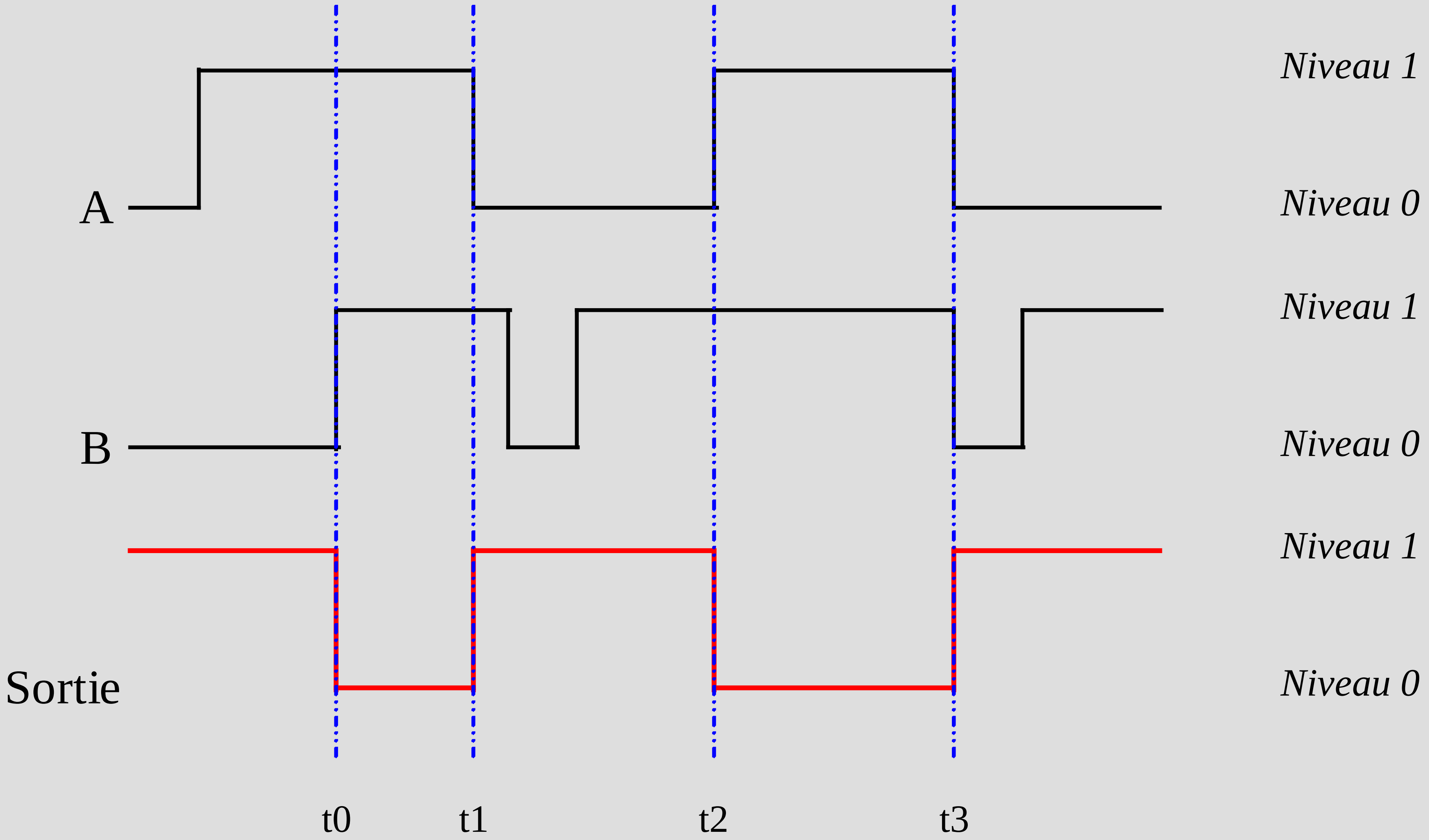
- La sortie d'une porte NAND est à un niveau haut dès que l'une deux entrées a un niveau bas (quel que soit le nombre d'entrées).
- Elle prend un niveau bas si et seulement si les deux entrées ont un niveau haut.

La fonction NAND: Composant intégrant 4 portes NAND à 2 entrées:

7400



La fonction NAND: Chronogramme:



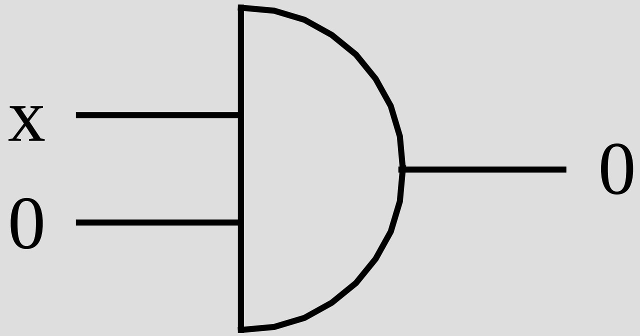
5. Les théorèmes de BOOLE et de DE MORGAN:

Les théorèmes de Boole: Les théorèmes pour 1 seule variable (1):

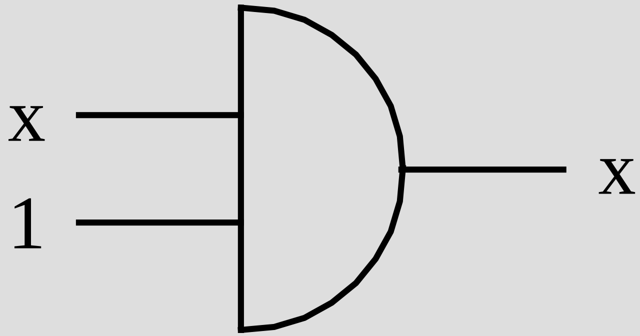
•Les théorèmes de Boole permettent de simplifier des expressions logiques.

•Les théorèmes pour 1 seule variable:

$$X \cdot 0 = 0$$

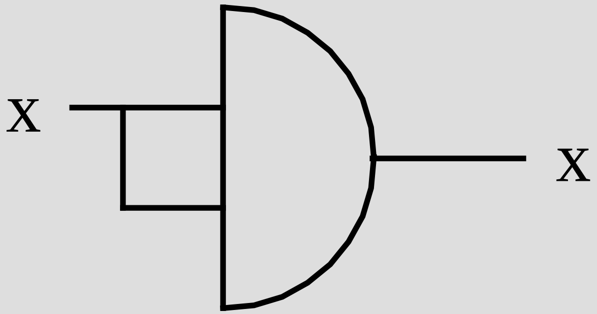


$$X \cdot 1 = X$$

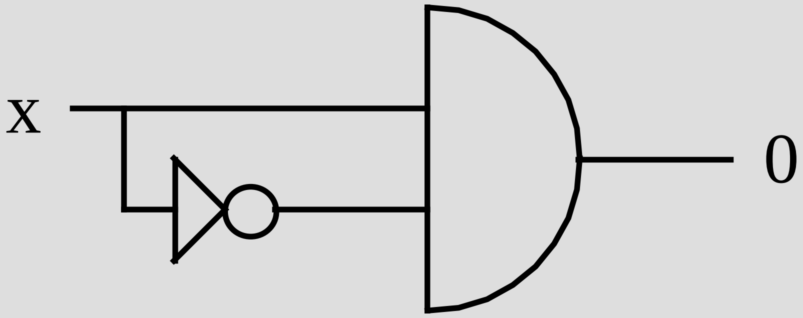


Les théorèmes pour 1 seule variable (2):

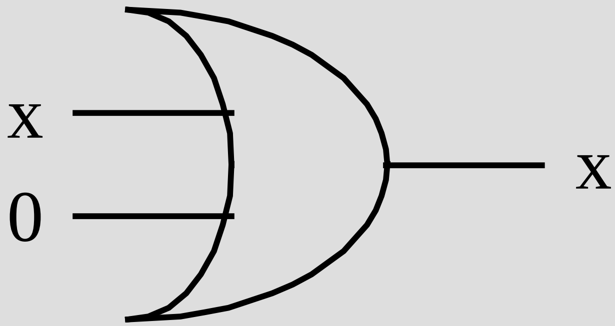
$$X \cdot X = X$$



$$X \cdot \overline{X} = 0$$

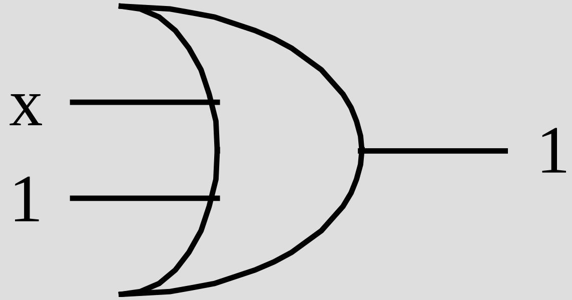


$$X + 0 = X$$

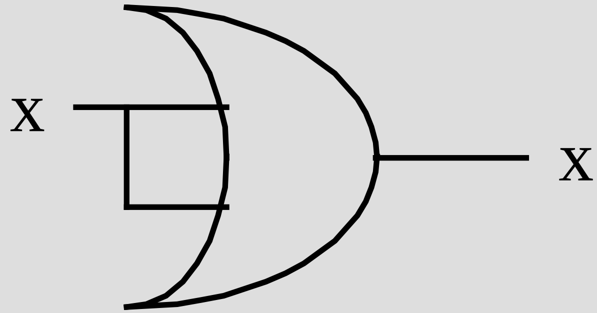


Les théorèmes pour 1 seule variable (3):

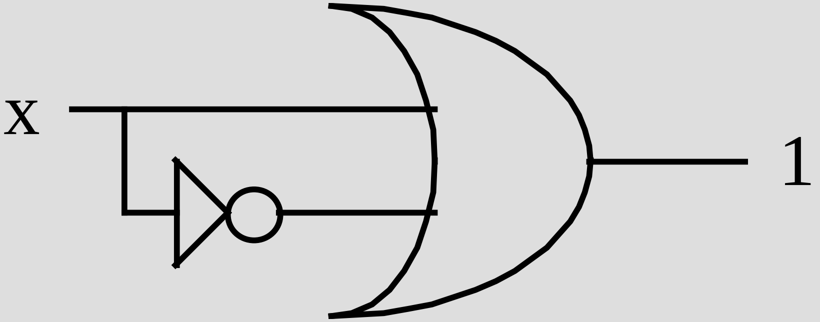
$$X + 1 = 1$$



$$X + X = X$$



$$X + \overline{X} = 1$$



Les théorèmes pour plusieurs variables (1):

- Commutativité de la fonction OU:
  - $x+y = y+x$
- Commutativité de la fonction ET:
  - $x.y = y.x$
- Associativité de la fonction OU:
  - $x+(y+z) = (x+y)+z = x+y+z$
- Associativité de la fonction ET:
  - $x(yz) = (xy)z = xyz$

Les théorèmes pour plusieurs variables (2):

- Distributivité de la multiplication par rapport à l'addition:
  - $x(y+z) = xy+xz$
  - $(w+x)(y+z) = wy+wz+xy+xz$
- Autres théorèmes:
  - $x+xy = x$
  - $x+\overline{x}y = x+y$

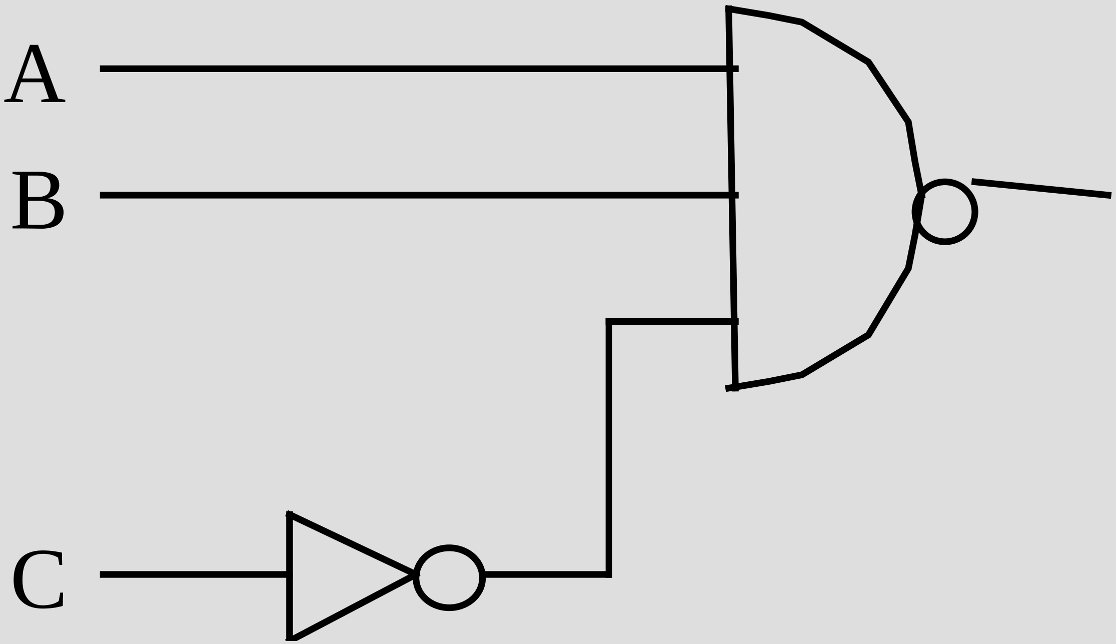
Les théorèmes de De Morgan:

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

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

Les théorèmes de De Morgan: Exemples:

$$z = \overline{A + \overline{B} \bullet C} = \overline{A} \bullet \left(\overline{\overline{B} \bullet C}\right) = \overline{A} \bullet \left(\overline{\overline{B}} + \overline{C}\right) = \overline{A} \bullet \left(B + \overline{C}\right)$$



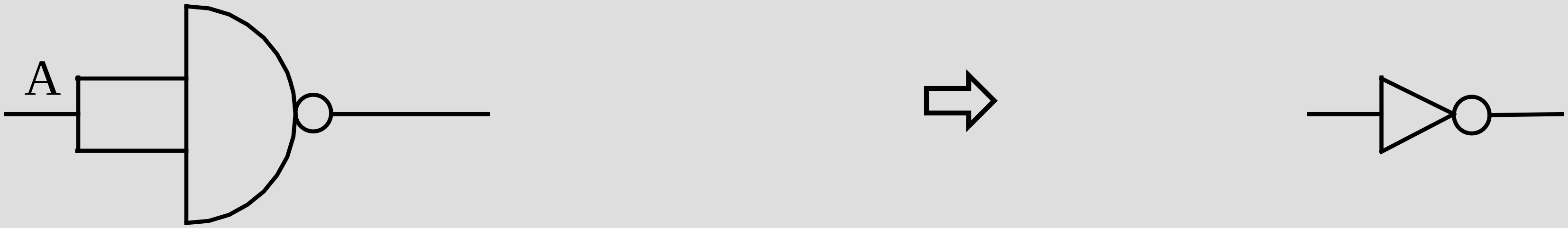
$$x = \overline{A \bullet B \bullet \overline{C}}$$

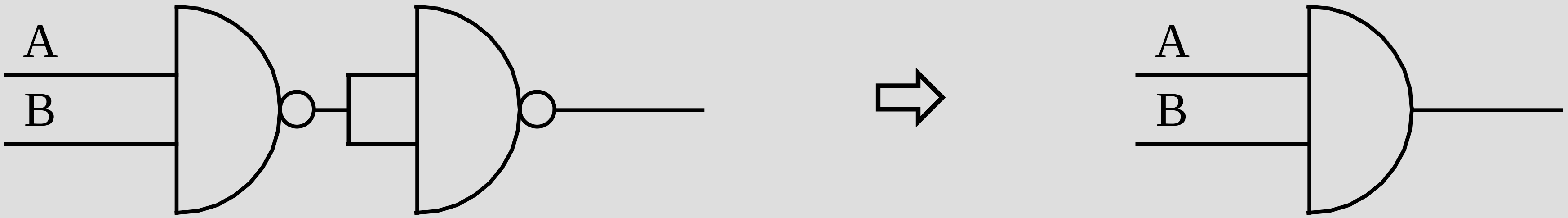
$$x = \overline{A} + \overline{B} + \overline{\overline{C}}$$

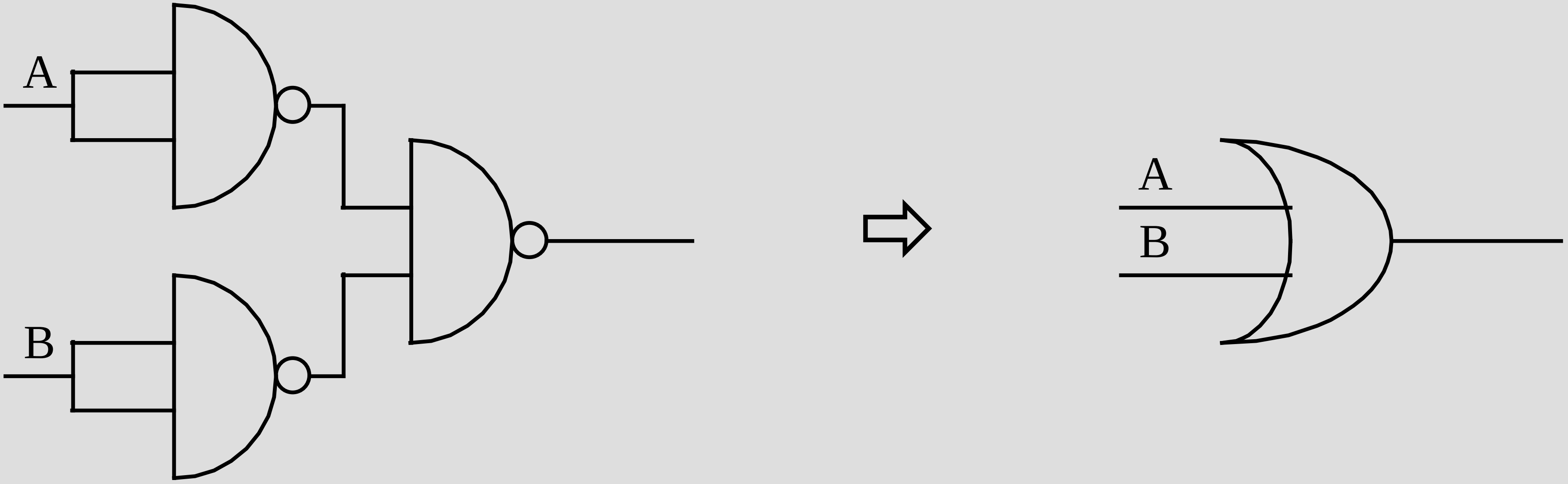
$$x = \overline{A} + \overline{B} + C$$

6. L 'utilisation des portes NOR et NAND:

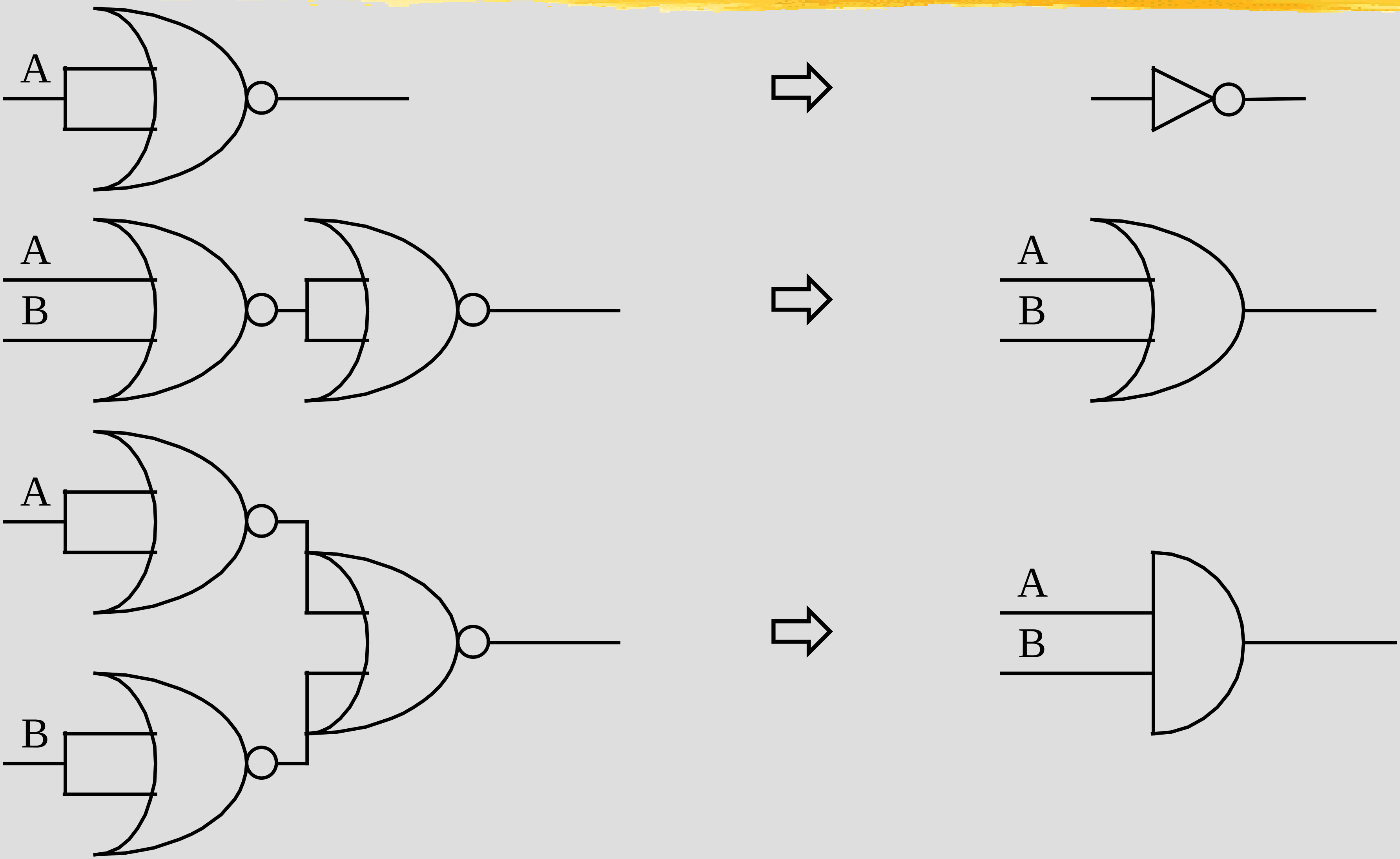
L 'universalité de la porte NAND:







L 'universalité de la porte NOR:



L 'utilisation des portes NAND et NOR (1) :

•RAPPEL: L 'universalité des portes NAND et NOR permet de créer toutes les fonctions logiques de base:

Les portes NAND et NOR offrent la possibilité de pouvoir réaliser n'importe quel circuit logique à l'aide d'un seul type de composant.

•Exemple

- •Soit à réaliser le circuit qui a pour expression de sortie:
  x = AB + CD, en utilisant le moins de CI possible.
- •Les CI à disposition sont des:
  - •7400 (NON-ET),
  - •7408 (ET),
  - •7432 (OU).
- •Chacun des CI comporte quatre portes identiques à deux entrées.

L 'utilisation des protes NAND et NOR (2):

•PREMIERE SOLUTION: (la plus simple, mais...)

- •2 portes ET,
- •1 porte OU.

•Soit au niveau des CI:

- •1*7408,
- •1*7432.

=> gaspillage de portes.

•DEUXIEME SOLUTION: (plus économique…)

Remplacer chaque porte ET et OU par son équivalent réalisé à l'aide de portes NAND. Il faut alors:

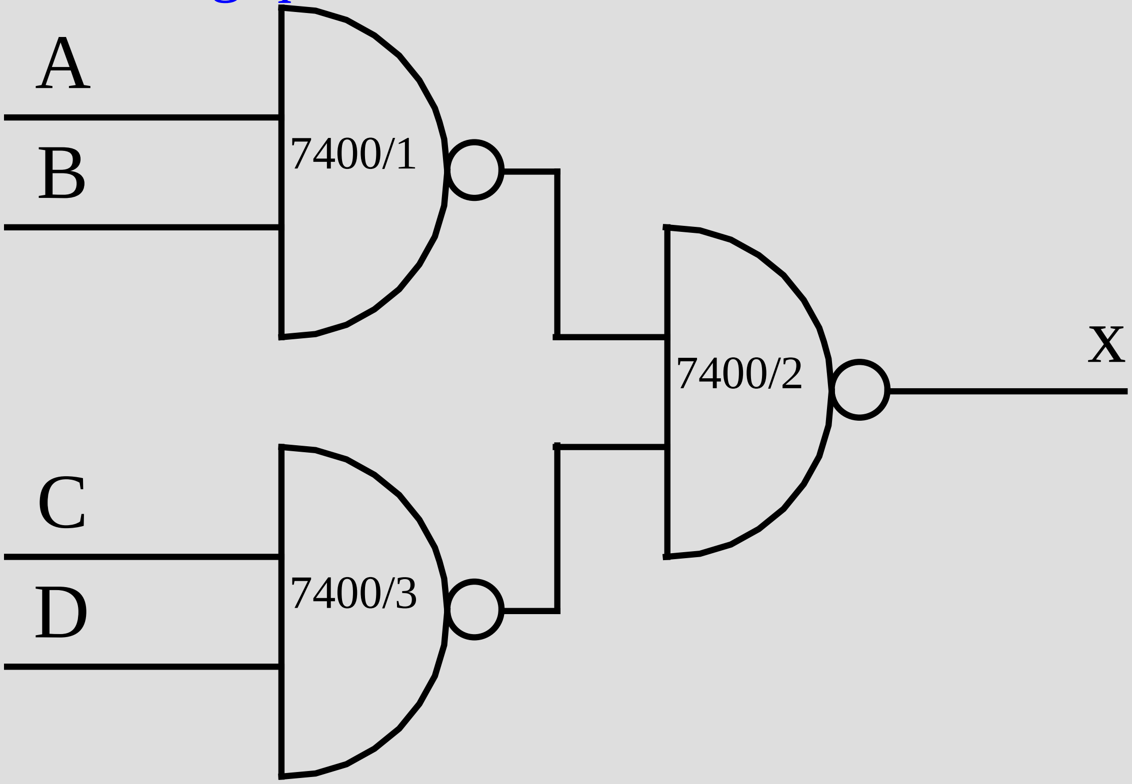
- • 7 portes NAND.

•Soit au niveau des CI:

- •1*7404.

L 'utilisation des portes NAND et NOR (3):

•Comme dans chaque branche il y a deux inverseurs, ceux-ci peuvent être supprimés et on se retrouve alors avec le schéma ci-dessous pour réaliser cette fonction logique:



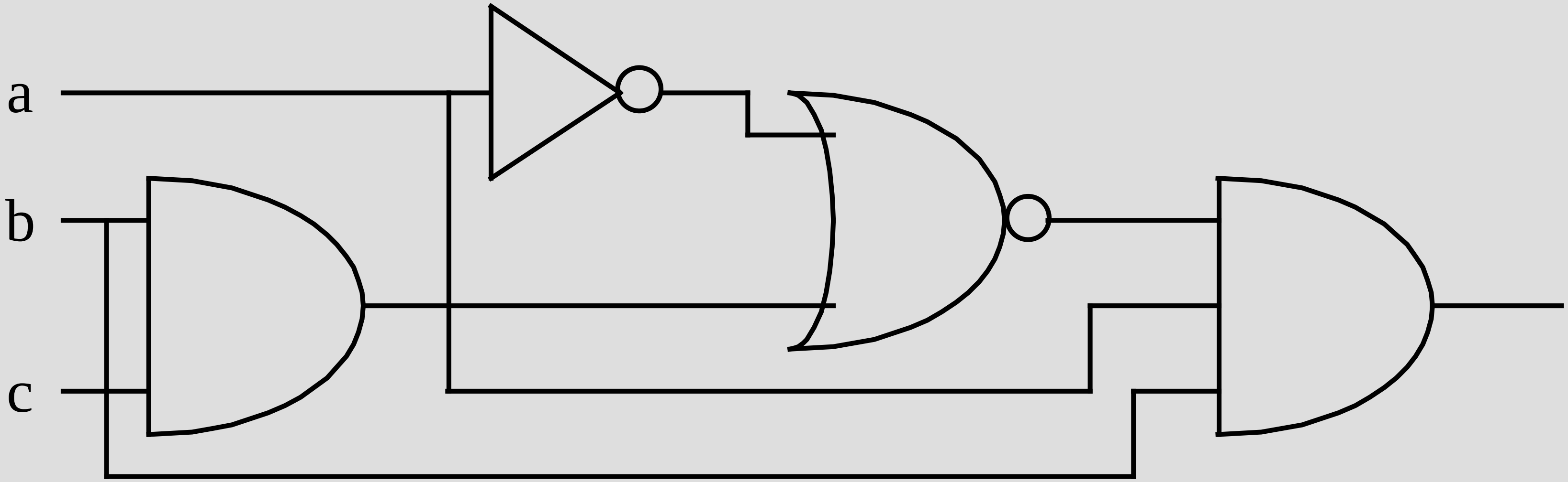
7. Simplification des circuits logiques:

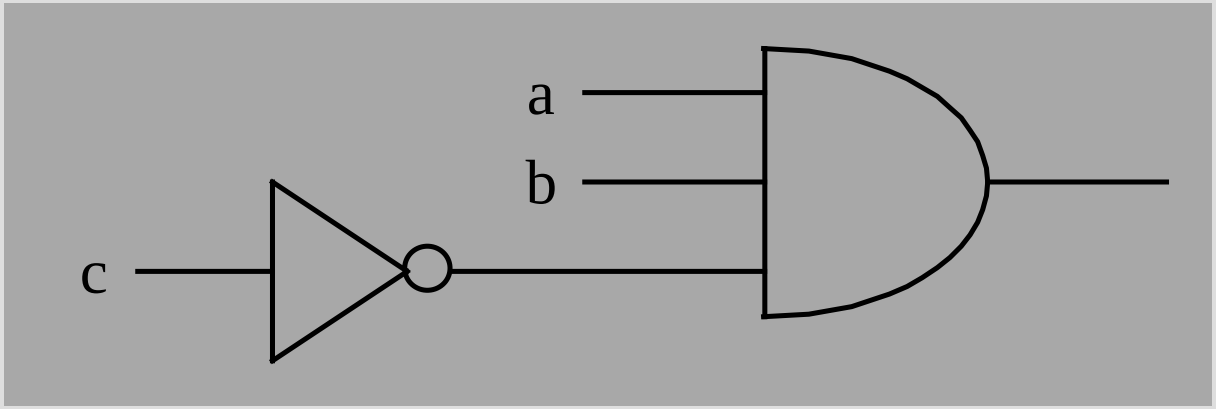
Simplification des circuits logiques: Introduction:

- La minimisation des circuits logiques a pour objectif de diminuer:
  - soit le nombre de termes,
  - soit le nombre de composants par terme.
- Ce qui conduit:
  - à utiliser moins de portes logiques,
  - à baisser le prix de revient.
  - De nombreux autres paramètres plaident en la faveur d'une simplification des circuits :

    - un nombre de connexions plus faible,
    - une consommation plus faible,
    - etc…

Simplification des circuits logiques: Exemple:





8. Simplification des expressions logiques:

Simplification des expressions logiques : Méthode:

•Simplification par approximations successives en utilisant les théorèmes énoncés auparavant.

•TECHNIQUE DE BASE : (similaire à l 'algèbre classique)

- utilisation des théorèmes de De Morgan
- développement => *sommes de produits*
- factorisation des variables communes pour éliminer plusieurs termes.

Simplification des expressions logiques: Exemple:

Simplification de l'équation:

$$X = ACB + A\overline{B}(\overline{\overline{A}.\overline{C}})$$

Décomposition au moyen du théorème de De Morgan

$$X = ABC + A\overline{B}\,(\overline{\overline{A}} + \overline{\overline{C}})$$

$X = ABC + A\overline{B}\,(A + C)$ *annulation de la double complémentation*

$X = ABC + A\overline{B}A + A\overline{B}C$ *multiplication*

$X = ABC + A\overline{B} + A\overline{B}C$ *A.A = A*

$X = AC\,(B + \overline{B}) + A\overline{B}$ *mise en facteur de (AC)*

$X = AC + A\overline{B}$ *car* $B + \overline{B} = 1$

$X = A\,(C + \overline{B})$ *mise en facteur de A*

9. Conception de circuits logiques complets:

Conception de circuits logiques complets: Méthode:

Cahier des charges

=>

Table de vérité

=>

Une équation(s) logique(s)

=>

Simplification

=>

Circuit physique.

Conception de circuits logiques complets: Exemple (1):

•Cahier des charges:

Création d'un circuit qui à sa sortie a un seulement si une majorité de ses trois entrées sont à 1.

•Table de vérité :

A	B	C	x
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Conception de circuits logiques complets: Exemple (2):

- Expression de la sortie x:

$$x = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

- Simplification de la sortie x:

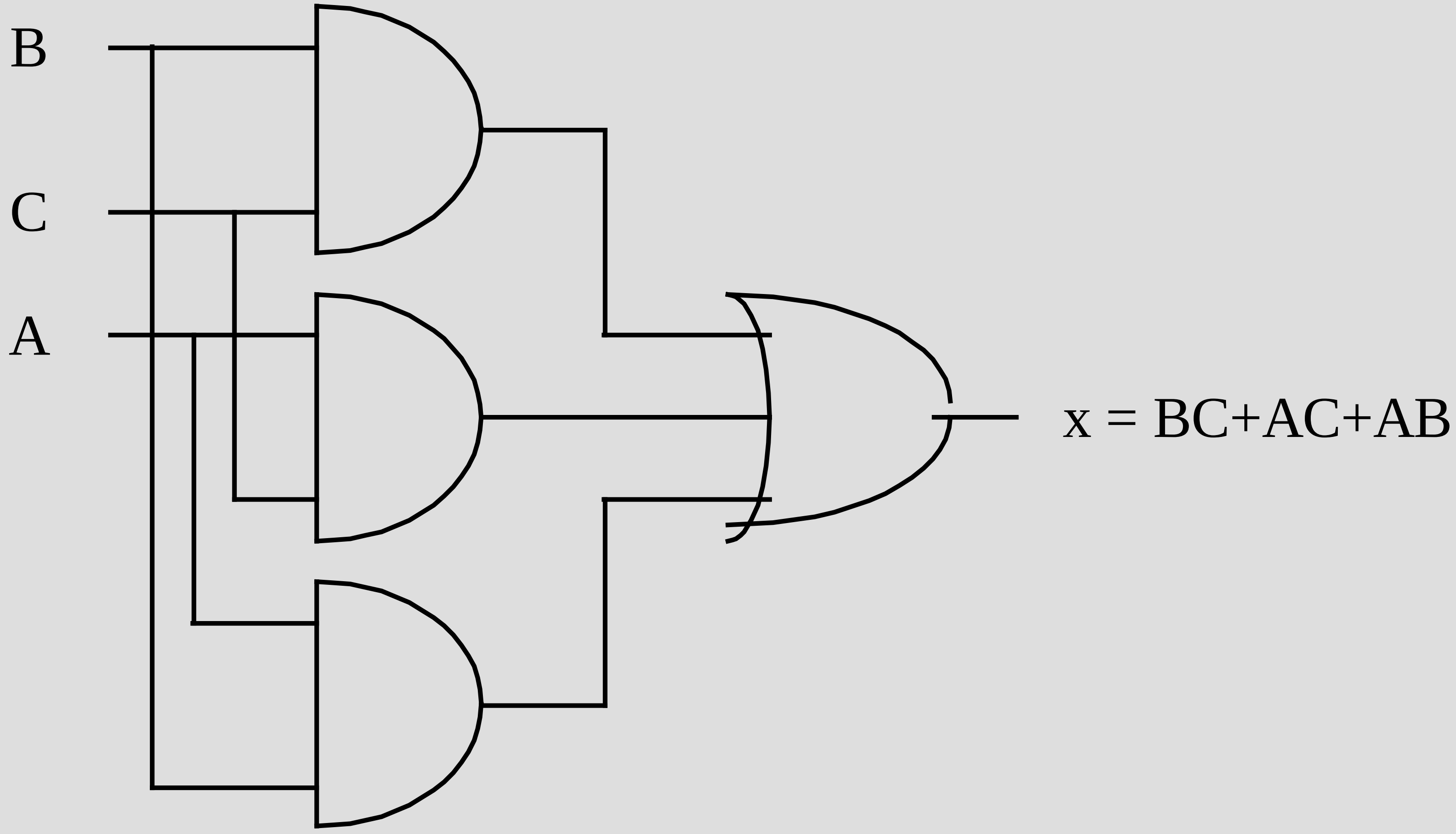
$$x = \overline{A}BC + ABC + A\overline{B}C + ABC + AB\overline{C} + ABC$$

$$= BC\left(\overline{A} + A\right) + AC\left(\overline{B} + B\right) + AB\left(\overline{C} + C\right)$$

$$= BC + AC + AB$$

Conception de circuits logiques complets: Exemple (3):

• Circuit physique:



10. Les diagrammes de KARNAUGH:

Les diagrammes de KARNAUGH: Définition:

•Un diagramme de Karnaugh, tout comme une table de vérité, met en évidence les relations qui existent entre les entrées et la sortie(s) de systèmes.

•Une présentation astucieuse des données permet alors d'utiliser cette forme de table de vérité pour effectuer des simplifications.

Les diagrammes de KARNAUGH: Exemple (1):

- Table de vérité:

A	B	X
0	0	1
0	1	0
1	0	0
1	1	1

- Tableau de KARNAUGH

	/B	B
/A	1	0
A	0	1

Les diagrammes de KARNAUGH: Exemples (2):

•Table de vérité:

A	B	C	X
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

•Tableau de KARNAUGH

	/C	C
/A/B	1	1
/AB	1	0
AB	1	0
A/B	0	0

Les diagrammes de KARNAUGH: Propriétés:

•La table de vérité donne la valeur de la sortie X pour chacune des combinaisons des valeurs d'entrée.

•Par contre, le diagramme de Karnaugh organise l'information de manière différente:

- •Chaque ligne de la table de vérité correspond à une case du diagramme de Karnaugh.
- •Utilisation du code GRAY => Le diagramme de Karnaugh est en fait un tableau circulaire. La première rangée du haut est en fait la suite de la dernière rangée du bas.

Simplification par diagramme de KARNAUGH: Définition:

•Il est possible de simplifier l'expression de la sortie X en combinant selon des règles précises les carrés du diagramme de Karnaugh qui contiennent des 1.

•On donne à ce processus de combinaisons le nom de *réunion*.

Réunion de doublet: Définiton et exemples (1):

•La réunion d'un doublet de 1 adjacents dans un diagramme de Karnaugh élimine la variable qui est à la fois complémentée et non complémentée.

•Exemples:

	/C	C
/A/B	0	0
/AB	1	0
AB	1	0
A/B	0	0

X = /AB/C+AB/C
= B/C

	/C	C
/A/B	0	0
/AB	1	1
AB	0	0
A/B	0	0

X = /AB/C+/ABC

= /AB

Réunion de doublet: Exemples (2):

	/C	C
/A/B	1	0
/AB	0	0
AB	0	0
A/B	1	0

X = /A/B/C+A/B/C

= /B/C

	/C/D	/CD	CD	C/D
/A/B	0	0	1	1
/AB	0	0	0	0
AB	0	0	0	0
A/B	1	0	0	1

X = /A/BCD + /A/BC/D + A/B/C/D + A/BC/D

= /A/BC + A/B/D

Réunion de quartet: Définition et exemples (1):

•La réunion d'un quartet de 1 adjacents dans un diagramme de Karnaugh élimine les deux variables qui sont à la fois complémentées et non complémentées.

•Exemples:

	/C	C
/A/B	0	1
/AB	0	1
AB	0	1
A/B	0	1

X = C

	/C/D	/CD	CD	C/D
/A/B	0	0	0	0
/AB	0	1	1	0
AB	0	1	1	0
A/B	0	0	0	0

X = BD

Réunion de quartet: Exemples (2):

	/C/D	/CD	CD	C/D
/A/B	0	0	0	0
/AB	0	0	0	0
AB	1	1	1	1
A/B	0	0	0	0

X = AB

	/C/D	/CD	CD	C/D
/A/B	0	0	0	0
/AB	0	0	0	0
AB	1	0	0	1
A/B	1	0	0	1

X = A/D

Réunion de quartet: Exemples (3):

	/C/D	/CD	CD	C/D
/A/B	1	0	0	1
/AB	0	0	0	0
AB	0	0	0	0
A/B	1	0	0	1

X = /B/D

Réunion d 'octet: Définition et exemples (1):

- La réunion d'octets de 1 adjacents dans un diagramme de Karnaugh élimine les trois variables qui sont à la fois complémentées et non complémentées.
- Exemples:

	/C/D	/CD	CD	C/D
/A/B	0	1	1	0
/AB	0	1	1	0
AB	0	1	1	0
A/B	0	1	1	0

	/C/D	/CD	CD	C/D
/A/B	1	0	0	1
/AB	1	0	0	1
AB	1	0	0	1
A/B	1	0	0	1

X = D

X = /D

Le processus de simplification au complet: Définition et exemples:

- Quand une variable se présente à la fois sous sa forme complémentée et non complémentée dans une réunion, cette variable est éliminée de l'expression.
- Seules apparaissent dans l'expression définitive les variables qui gardent la même forme dans tous les carrés d'une réunion.
- Exemples:

	/C/D	/CD	CD	C/D
/A/B	0	0	0	1
/AB	0	1	1	0
AB	0	1	1	0
A/B	0	0	1	0

X = /A/BC/D + ACD + BD

	/C/D	/CD	CD	C/D
/A/B	0	0	1	0
/AB	1	1	1	1
AB	1	1	0	0
A/B	0	0	0	0

X = /AB + B/C + /ACD

11. La fonction logique OU exclusif et son complément:

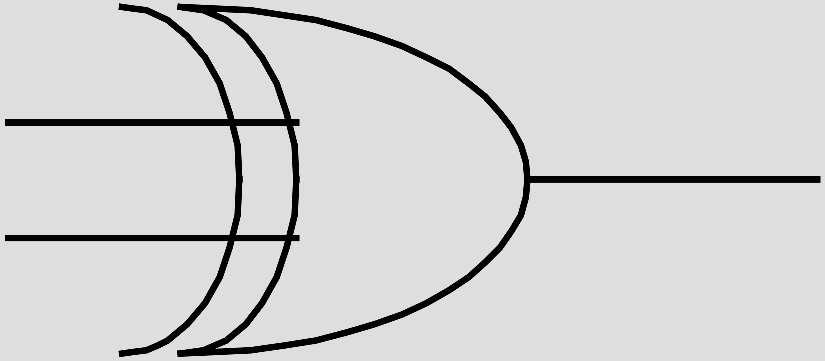
La fonction OU exclusif (X-OR): Sa table de vérité:

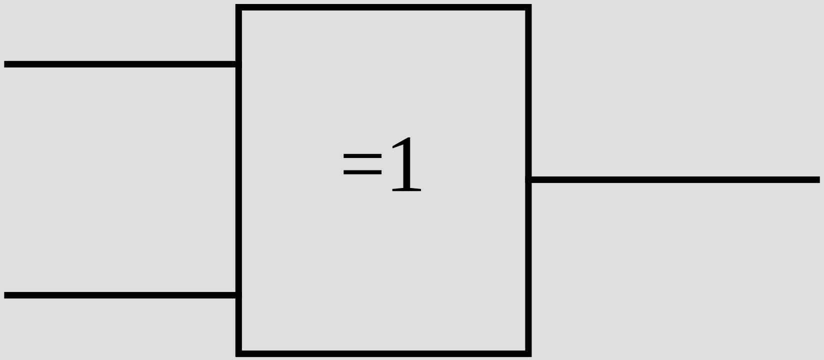
•L ’opérateur de la fonction OU exclusif est: $\oplus$; $X = A \oplus B$.

A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

•Cette table de vérité montre que la sortie n'est active que lorsque les signaux sur les deux entrées sont opposés.

La fonction OU exclusif (X-OR): Sa porte logique:



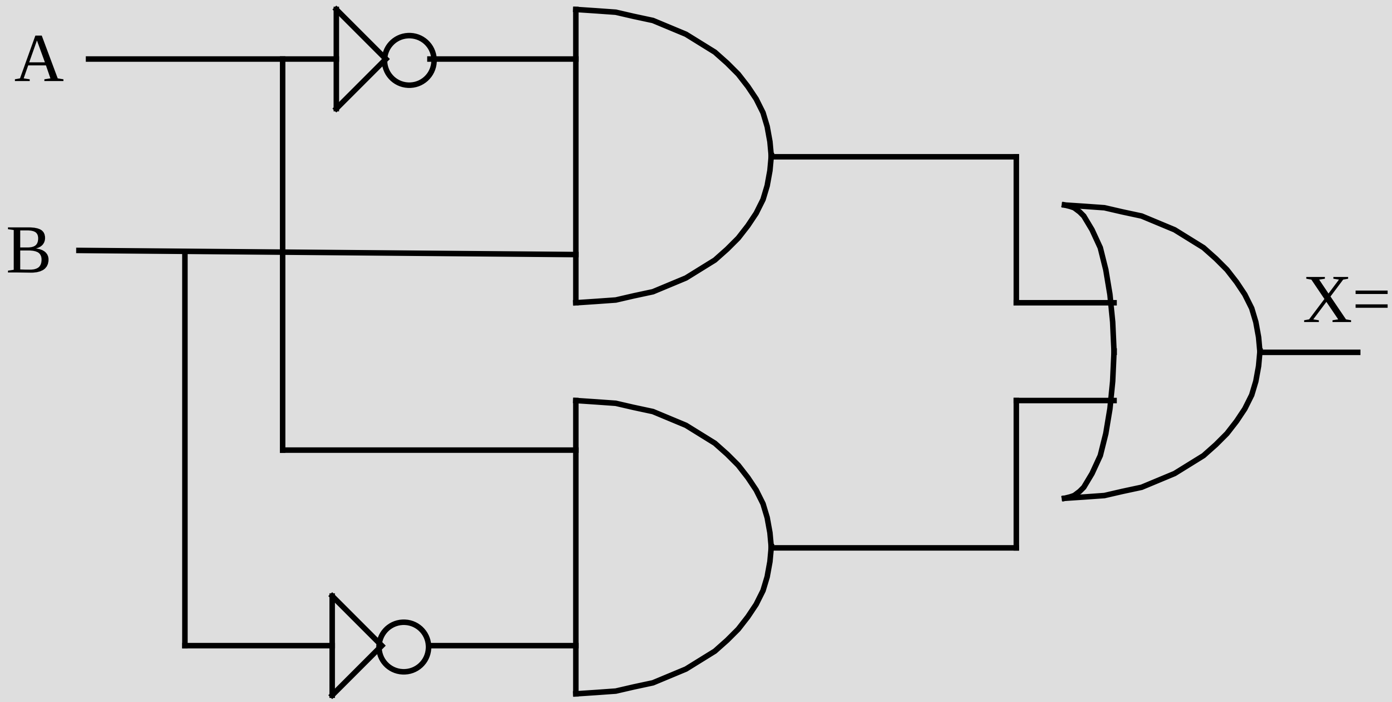


•Le circuit intégré qui contient des porte OU exclusif est le 74HC86.

•<u>Remarque:</u>

Une porte OU *exclusif* n'a toujours que deux entrées. Il n'existe pas de porte OU *exclusif* à trois ou quatre entrées.

La fonction X-OR: Structure interne:



$$X = \overline{A}.B + A.\overline{B}$$

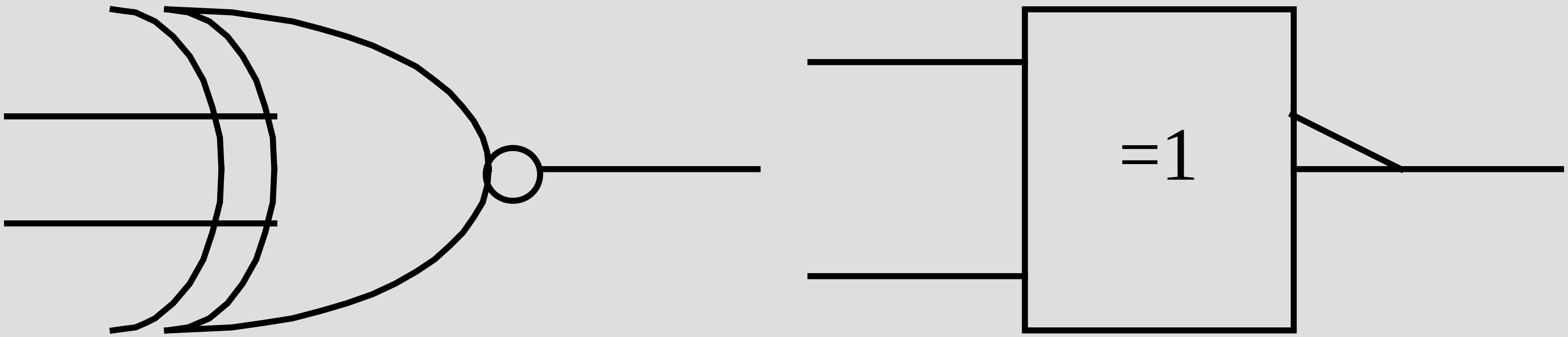
La fonction OU exclusif NON (X-NOR): Sa table de vérité:

•L 'équation est: $X = \overline{A \oplus B}$.

A	B	X
0	0	1
0	1	0
1	0	0
1	1	1

•Cette table de vérité montre que la sortie n'est active que lorsque les signaux sur les deux entrées sont identiques.

La fonction OU exclusif NON (X-NOR): Sa porte logique:

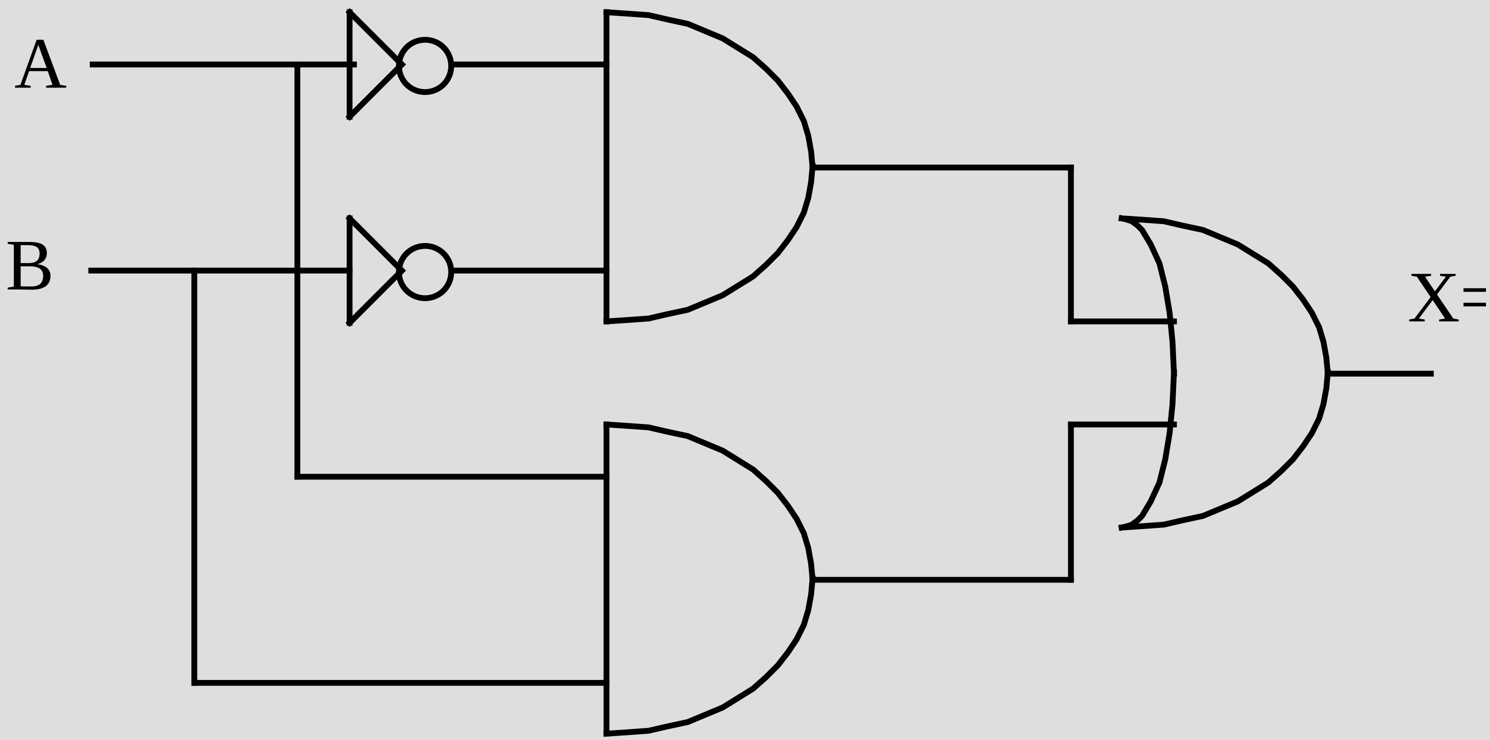


•Le circuit intégré qui contient des portes OU exclusif NON est le 74HC266.

•<u>Remarque:</u>

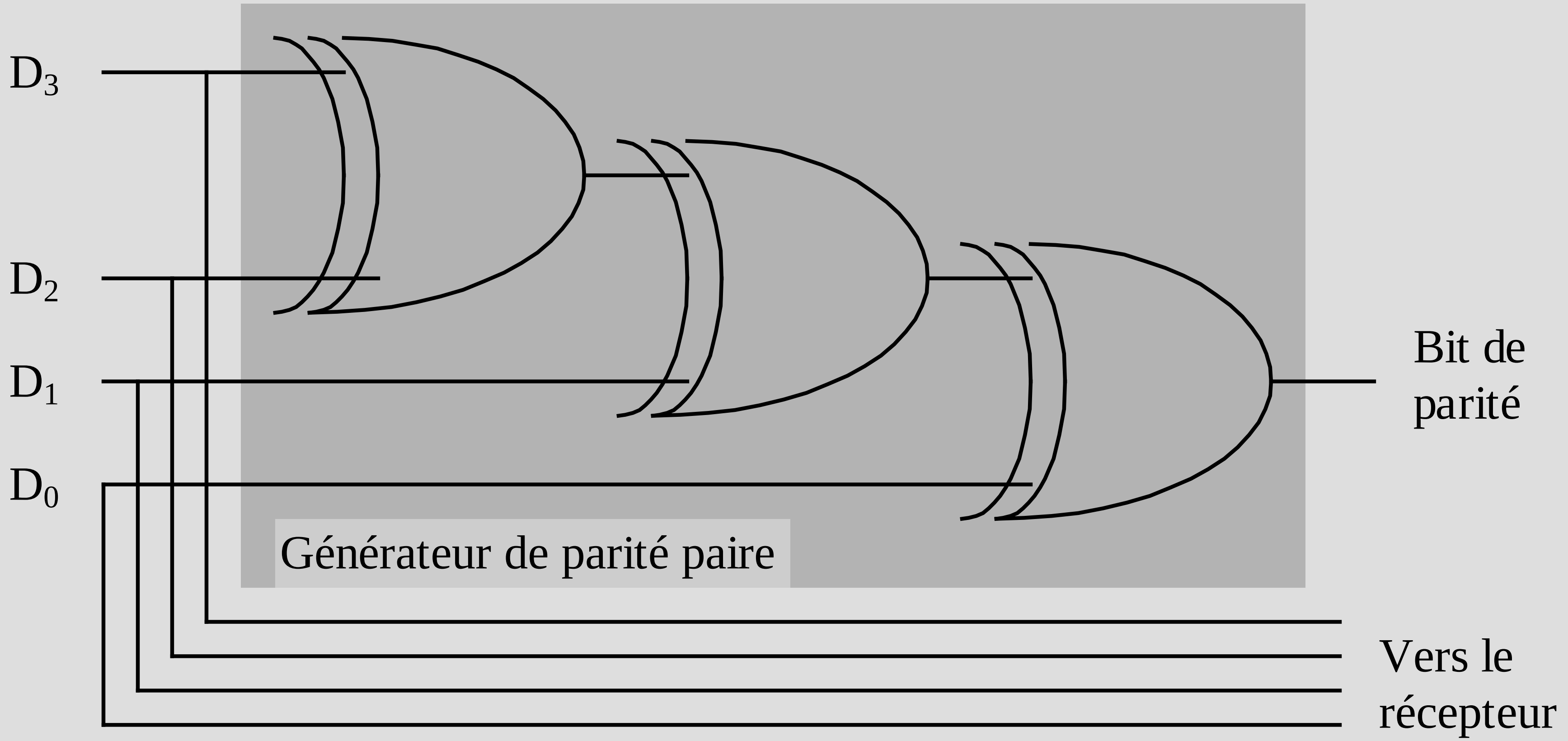
Une porte OU *exclusif* NON n'a toujours que deux entrées. Il n'existe pas de porte OU *exclusif* NON à trois ou quatre entrées.

La fonction X-NOR: Structure interne:



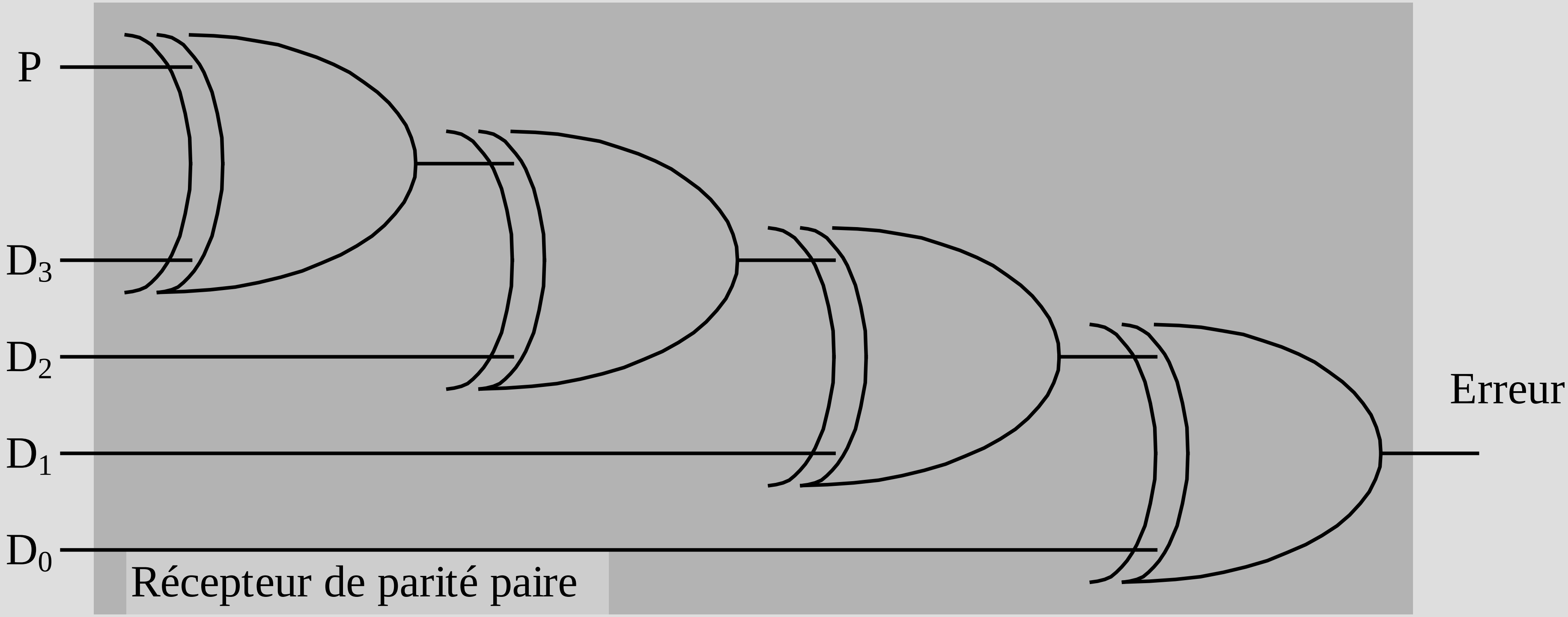
$$X = A.B + \overline{A}.\overline{B}$$

Exemple d 'application: Générateur de parité:



Générateur de parité paire à l'aide de portes X-OR

Exemple d 'application: Contrôleur de parité:



Récepteur de parité paire à l'aide de portes X-OR

CONCLUSION:

- **On a vu:**
  - **Les fonctions logiques élémentaires**
  - **La forme algébrique**
  - **Les théorèmes de BOOLE et de DE MORGAN**
  - **L 'utilisation des portes NOR et NAND**
  - **Simplification des circuits logiques et des expressions logiques**
  - **Les diagrammes de KARNAUGH**
- **Les connaissances de ce chapitre nous permettront de nous intéresser à la logique séquentielle.**
- **Nous étudierons les différentes technologies des circuits logiques.**