

INF1500 :

Logique des systèmes numériques

- Cours 2: Table de vérité, simplification des expressions booléennes, logique mixte

Table de vérité

Une *table de vérité* nous fait connaître la réaction d'un circuit logique (sa valeur de sortie) aux diverses combinaisons de niveaux logiques appliqués aux entrées (2^n).

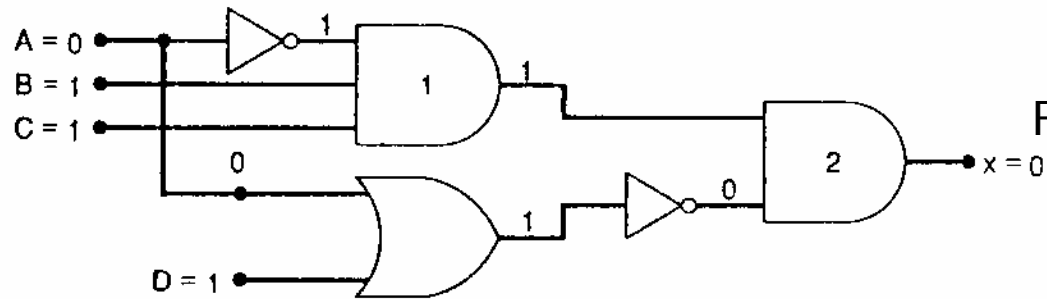
Table 4-4
General truth table structure for a 3-variable logic function, $F(X,Y,Z)$.

Row	X	Y	Z	F
0	0	0	0	$F(0,0,0)$
1	0	0	1	$F(0,0,1)$
2	0	1	0	$F(0,1,0)$
3	0	1	1	$F(0,1,1)$
4	1	0	0	$F(1,0,0)$
5	1	0	1	$F(1,0,1)$
6	1	1	0	$F(1,1,0)$
7	1	1	1	$F(1,1,1)$

Table 4-5
Truth table for a particular 3-variable logic function, $F(X,Y,Z)$.

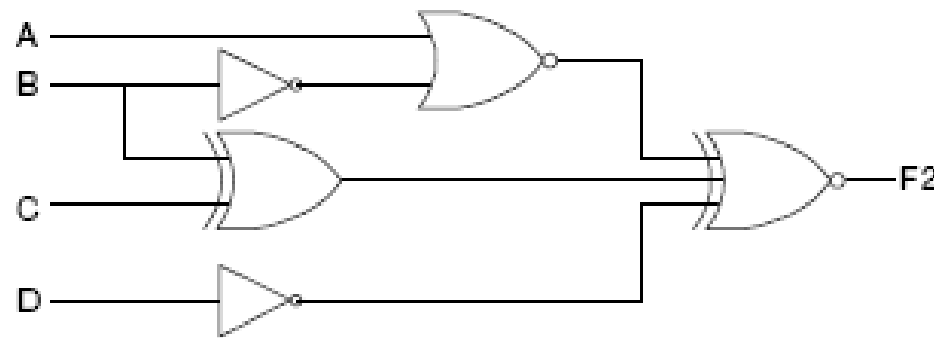
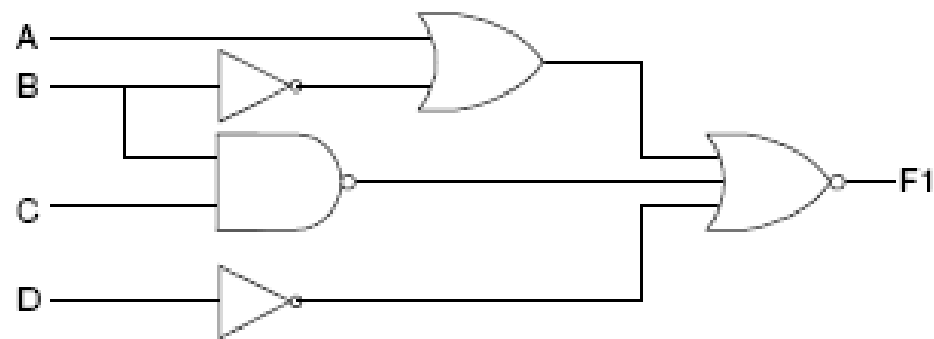
Row	X	Y	Z	F
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

Table de vérité d'un circuit.

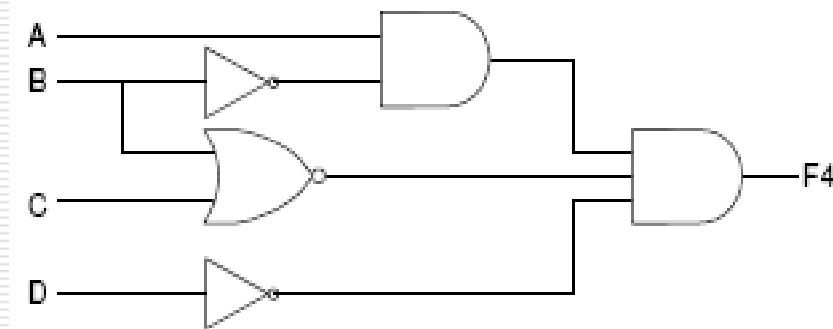
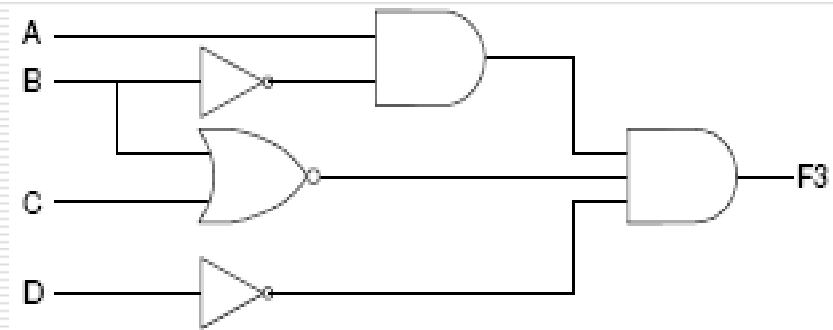


A	B	C	F
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

EXERCICES



EXERCICES



Formes canoniques et expansions en mintermes et maxtermes

□ *formes canoniques*

- À partir d'une table de vérité, on peut facilement donner la valeur de la fonction logique correspondante par l'une des deux formes canoniques : une somme de produits ou un produit de sommes.
- Pour obtenir la somme de produits, on énumère les termes de la fonction qui correspondent à une valeur de 1 de celle-ci. Chaque terme est composé d'un produit (ET logique) de chaque variable de la fonction. Une variable ayant la valeur 0 dans la rangée correspondante est complémentée.
- Pour obtenir le produit des sommes, on énumère les termes de la fonction qui correspondent à une valeur de 0 de celle-ci. Chaque terme est composé d'une somme (OU logique) de chaque variable de la fonction. Une variable ayant la valeur 1 dans la rangée correspondante est complémentée.

Exemples

#	<i>A</i>	<i>B</i>	<i>C</i>	<i>F</i>
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	0

somme de produits : $F = A'B'C + A'BC' + A'BC + ABC'$

produit de sommes : $F = (A + B + C)(A' + B + C)(A' + B + C')(A' + B' + C')$

Exemples

#	<i>A</i>	<i>B</i>	<i>C</i>	<i>F</i>
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	0

somme de produits : $F = A'B'C + A'BC' + A'BC + ABC'$

produit de sommes : $F = (A + B + C)(A' + B + C)(A' + B + C')(A' + B' + C')$

Exemples

#	<i>A</i>	<i>B</i>	<i>C</i>	<i>F</i>
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	0

Somme de produits : $F = A'B'C + A'BC' + A'BC + ABC'$

Produit de sommes : $F = (A + B + C)(A' + B + C)(A' + B + C')(A' + B' + C')$

Le tableau suivant donne l'expression des mintermes et maxtermes pour une fonction à trois variables

Minterme - Maxterme

#	A	B	C	m_i	M_i
0	0	0	0	$m_0 = A' B' C'$	$M_0 = A + B + C$
1	0	0	1	$m_1 = A' B' C$	$M_1 = A + B + C'$
2	0	1	0	$m_2 = A' B C'$	$M_2 = A + B' + C$
3	0	1	1	$m_3 = A' B C$	$M_3 = A + B' + C'$
4	1	0	0	$m_4 = A B' C'$	$M_4 = A' + B + C$
5	1	0	1	$m_5 = A B' C$	$M_5 = A' + B + C'$
6	1	1	0	$m_6 = A B C'$	$M_6 = A' + B' + C$
7	1	1	1	$m_7 = A B C$	$M_7 = A' + B' + C'$

Pour l'exemple précédent, on a alors :

$$\begin{aligned}
 F &= m_1 + m_2 + m_3 + m_6 = \sum (m_1, m_2, m_3, m_6) \\
 &= M_0 M_4 M_5 M_7 = \prod (M_0, M_4, M_5, M_7)
 \end{aligned}$$

Minterme - Maxterme

Table 4-6
Minterms and maxterms
for a 3-variable logic
function, $F(X, Y, Z)$.

<i>Row</i>	<i>X</i>	<i>Y</i>	<i>Z</i>	<i>F</i>	<i>Minterm</i>	<i>Maxterm</i>
0	0	0	0	F(0,0,0)	$X' \cdot Y' \cdot Z'$	$X + Y + Z$
1	0	0	1	F(0,0,1)	$X' \cdot Y' \cdot Z$	$X + Y + Z'$
2	0	1	0	F(0,1,0)	$X' \cdot Y \cdot Z'$	$X + Y' + Z$
3	0	1	1	F(0,1,1)	$X' \cdot Y \cdot Z$	$X + Y' + Z'$
4	1	0	0	F(1,0,0)	$X \cdot Y' \cdot Z'$	$X' + Y + Z$
5	1	0	1	F(1,0,1)	$X \cdot Y' \cdot Z$	$X' + Y + Z'$
6	1	1	0	F(1,1,0)	$X \cdot Y \cdot Z'$	$X' + Y' + Z$
7	1	1	1	F(1,1,1)	$X \cdot Y \cdot Z$	$X' + Y' + Z'$

Observations

- On observe que $m_i = M_i'$.

- Résumé :
 - Pour obtenir une fonction, on prend la somme des mintermes où la fonction vaut 1, ou le produit des maxtermes où la fonction vaut 0.

 - Pour obtenir l'inverse d'une fonction, on prend la somme des mintermes où la fonction vaut 0, ou le produit des maxtermes où la fonction vaut 1.

Simplification des expressions booléennes avec les théorèmes de Boole

Table 4-1
Switching-algebra theorems with one variable.

(T1)	$X + 0 = X$	(T1')	$X \cdot 1 = X$	(Identities)
(T2)	$X + 1 = 1$	(T2')	$X \cdot 0 = 0$	(Null elements)
(T3)	$X + X = X$	(T3')	$X \cdot X = X$	(Idempotency)
(T4)	$(X')' = X$			(Involution)
(T5)	$X + X' = 1$	(T5')	$X \cdot X' = 0$	(Complements)

$X + X' \cdot Y = X + Y$
Ou exclusif = $X \cdot Y' + X' \cdot Y$
Non-Ou ex = $X \cdot Y + X' \cdot Y'$

Table 4-2 Switching-algebra theorems with two or three variables.

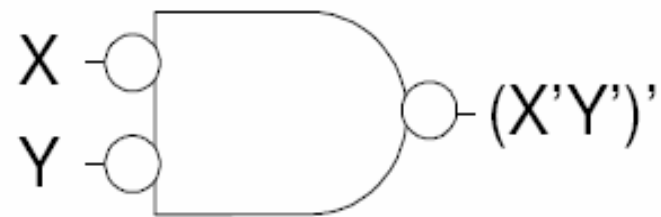
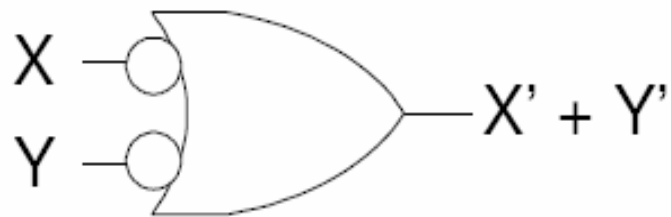
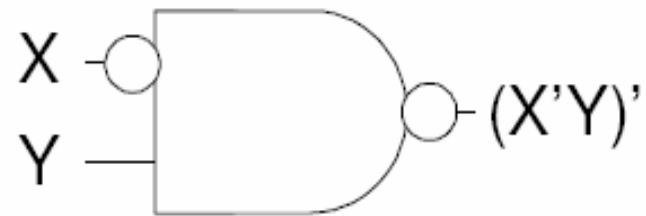
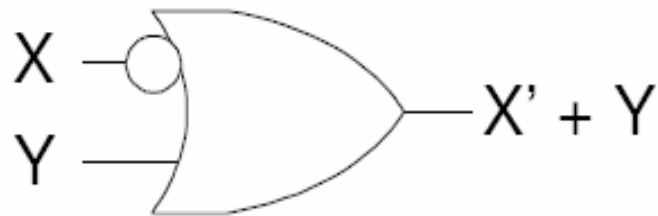
(T6)	$X + Y = Y + X$	(T6')	$X \cdot Y = Y \cdot X$	(Commutativity)
(T7)	$(X + Y) + Z = X + (Y + Z)$	(T7')	$(X \cdot Y) \cdot Z = X \cdot (Y \cdot Z)$	(Associativity)
(T8)	$X \cdot Y + X \cdot Z = X \cdot (Y + Z)$	(T8')	$(X + Y) \cdot (X + Z) = X + Y \cdot Z$	(Distributivity)
(T9)	$X + X \cdot Y = X$	(T9')	$X \cdot (X + Y) = X$	(Covering)
(T10)	$X \cdot Y + X \cdot Y' = X$	(T10')	$(X + Y) \cdot (X + Y') = X$	(Combining)
(T11)	$X \cdot Y + X' \cdot Z + Y \cdot Z = X \cdot Y + X' \cdot Z$			(Consensus)
(T11')	$(X + Y) \cdot (X' + Z) \cdot (Y + Z) = (X + Y) \cdot (X' + Z)$			

Table 4-3 Switching-algebra theorems with n variables.

(T12)	$X + X + \dots + X = X$	(Generalized idempotency)
(T12')	$X \cdot X \cdot \dots \cdot X = X$	
(T13)	$(X_1 \cdot X_2 \cdot \dots \cdot X_n)' = X_1' + X_2' + \dots + X_n'$	(DeMorgan's theorems)
(T13')	$(X_1 + X_2 + \dots + X_n)' = X_1' \cdot X_2' \cdot \dots \cdot X_n'$	
(T14)	$[F(X_1, X_2, \dots, X_n, +, \cdot)]' = F(X_1', X_2', \dots, X_n', \cdot, +)$	(Generalized DeMorgan's theorem)
(T15)	$F(X_1, X_2, \dots, X_n) = X_1 \cdot F(1, X_2, \dots, X_n) + X_1' \cdot F(0, X_2, \dots, X_n)$	(Shannon's expansion theorems)
(T15')	$F(X_1, X_2, \dots, X_n) = [X_1 + F(0, X_2, \dots, X_n)] \cdot [X_1' + F(1, X_2, \dots, X_n)]$	

Fonctions logiques et portes équivalentes

Bien que ce soit moins utilisé, on peut aussi placer le symbole d'inversion à l'une ou aux deux entrées de la porte logique (ainsi qu'à sa sortie). On obtient alors plusieurs combinaisons possibles, dont en voici quatre exemples. On verra plus tard les équivalences résultantes.



Fonctions logiques et portes équivalentes

Conséquence immédiate des théorèmes de De Morgan - Ces transformations sont à la base de la logique mixte

Copyright © 2000 by Prentice Hall, Inc.
Digital Design Principles and Practices, 3/e

(Fonction ET)

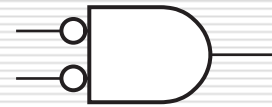
- a) 7408 porte AND
- b) 7400 porte NAND
- c) 7402 porte NOR
- d) 7432 porte OR



(a)



(b)



(c)



(d)

Copyright © 2000 by Prentice Hall, Inc.
Digital Design Principles and Practices, 3/e

(Fonction OU)

- a) 7432 porte OR
- b) 7402 porte NOR
- c) 7400 porte NAND
- d) 7408 porte AND



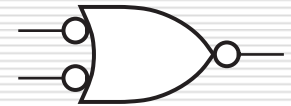
(a)



(b)



(c)

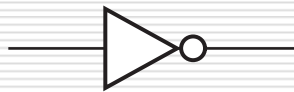


(d)

Copyright © 2000 by Prentice Hall, Inc.
Digital Design Principles and Practices, 3/e

(Fonction INV)

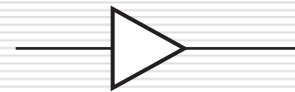
- a) b) Inverseur
- c) d) Suiveur (buffer)



(a)



(b)

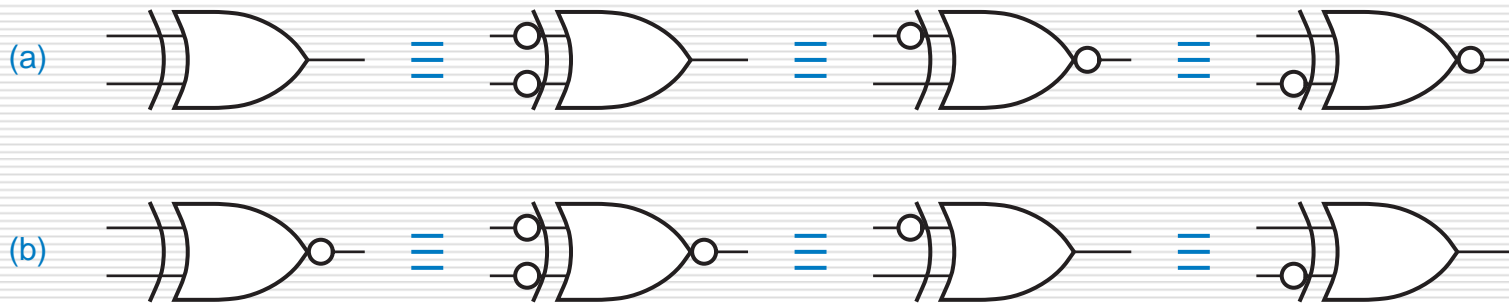


(c)



(d)

Fonctions logiques et portes équivalentes - Suite



Copyright © 2000 by Prentice Hall, Inc.
Digital Design Principles and Practices, 3/e

Figure a (Fonction XOR)

74x86 porte XOR

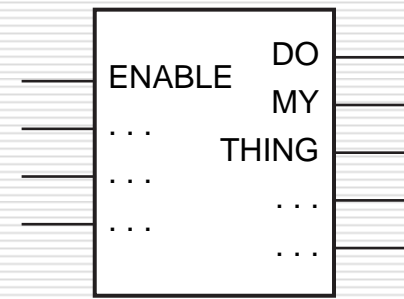
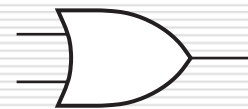
Figure b (Fonction XNOR)

74x266 porte XOR

Logique Mixte - Niveaux logiques

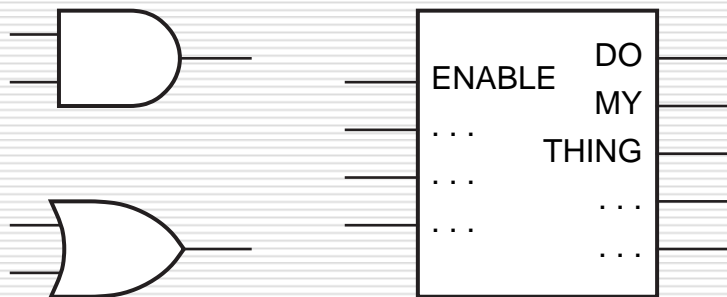
Table 5-1
Each line shows a different naming convention for active levels.

<i>Active Low</i>	<i>Active High</i>
READY-	READY+
ERROR.L	ERROR.H
ADDR15(L)	ADDR15(H)
RESET*	RESET
ENABLE~	ENABLE
~GO	GO
/RECEIVE	RECEIVE
TRANSMIT_L	TRANSMIT

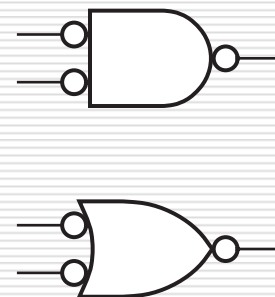


(a)

Copyright © 2000 by Prentice Hall
Digital Design Principles and Practices



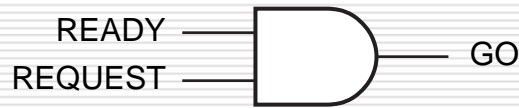
(a)



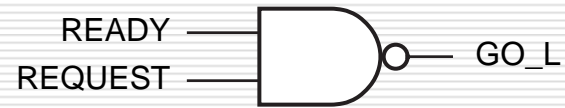
(b)

Copyright © 2000 by Prentice Hall, Inc.
Digital Design Principles and Practices, 3/e

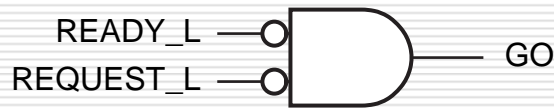
Logique Mixte: Exemples de choix de signaux



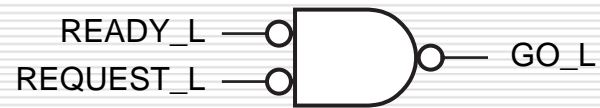
(a)



(b)



(c)

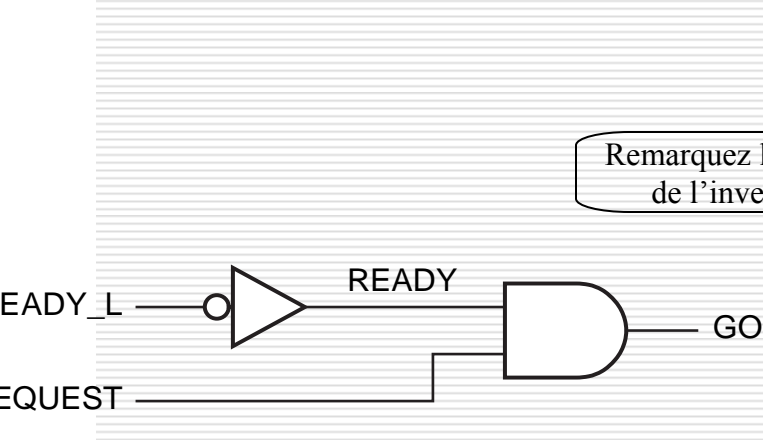


(d)

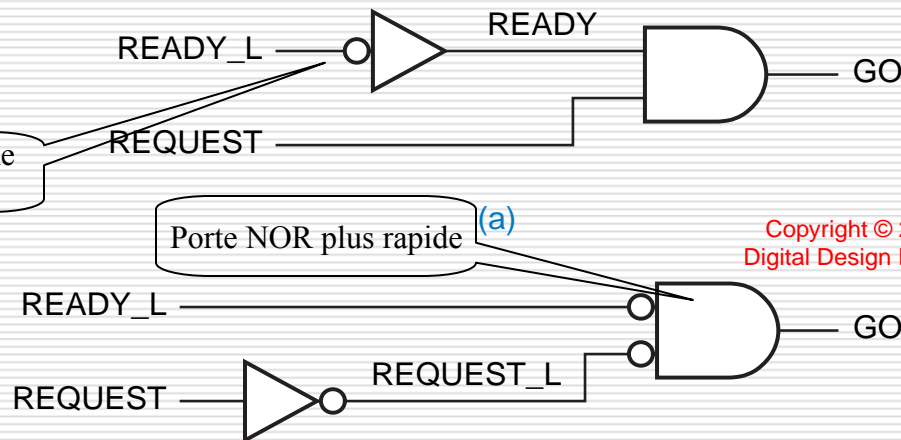
Copyright © 2000 by Prentice Hall, Inc.
Digital Design Principles and Practices, 3/e

Remarquez la forme de l'inverseur

Porte NOR plus rapide (a)



(a)

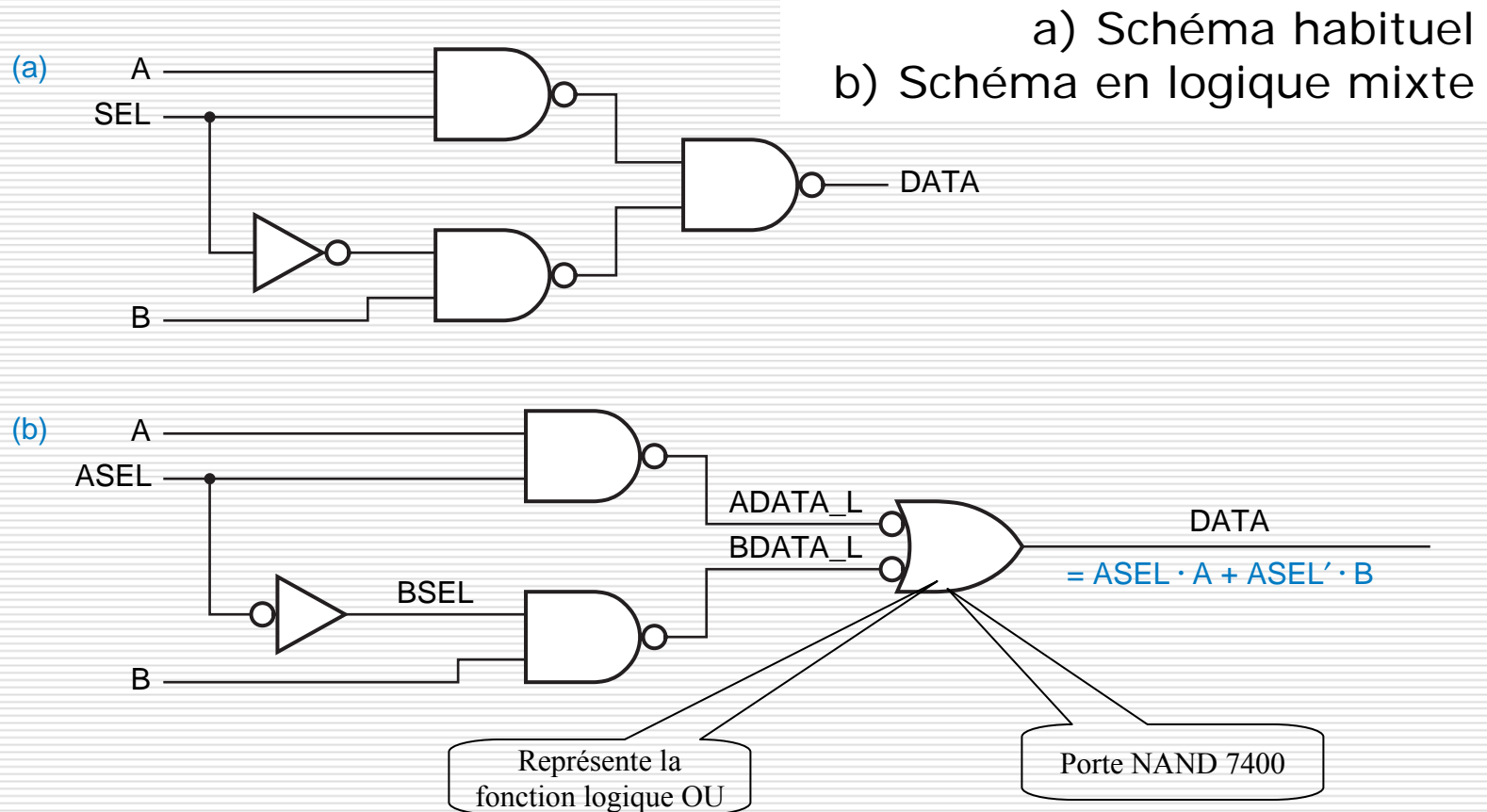


(b)

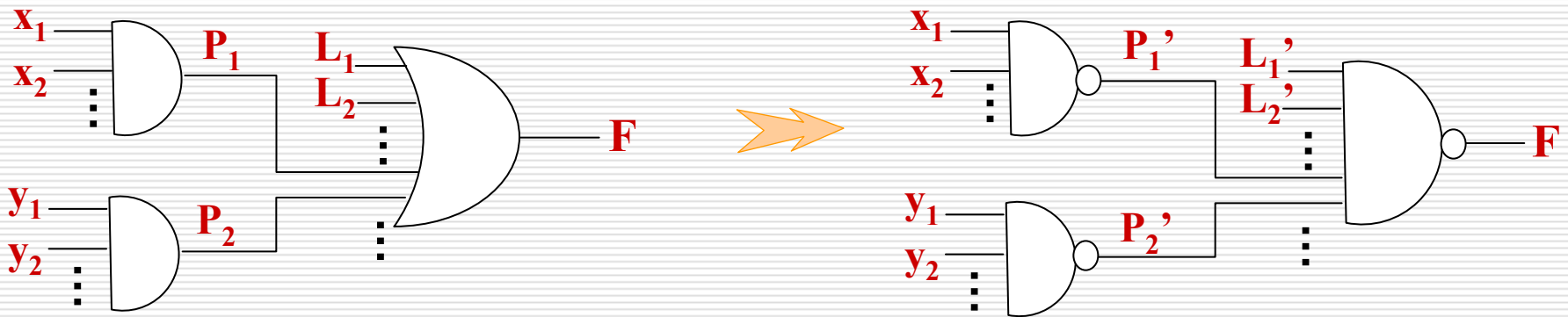
Copyright © 2000 by Prentice Hall, Inc.
Digital Design Principles and Practices, 3/e

Copyright © 2000 by Prentice Hall, Inc.
Digital Design Principles and Practices, 3/e

Logique Mixte: Exemples de circuits



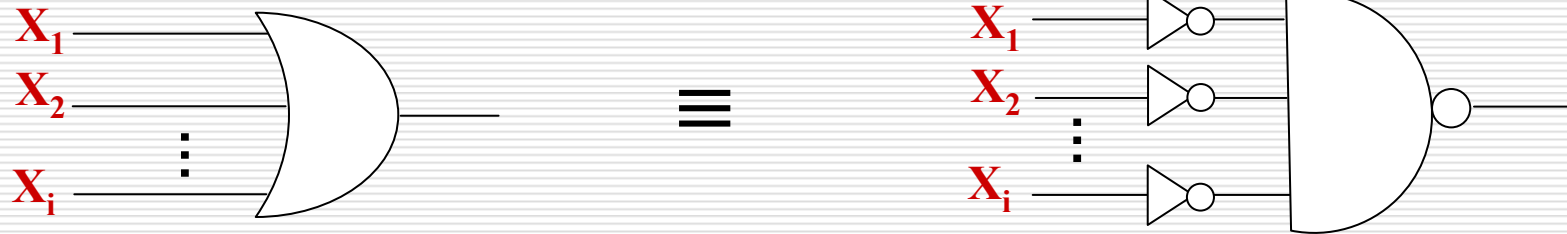
Autre exemple de transformation



Avant transformation

Après transformation

Car:



$$(X_1 + X_2 + \dots + X_n)'' = (X_1' X_2' \dots X_n')'$$

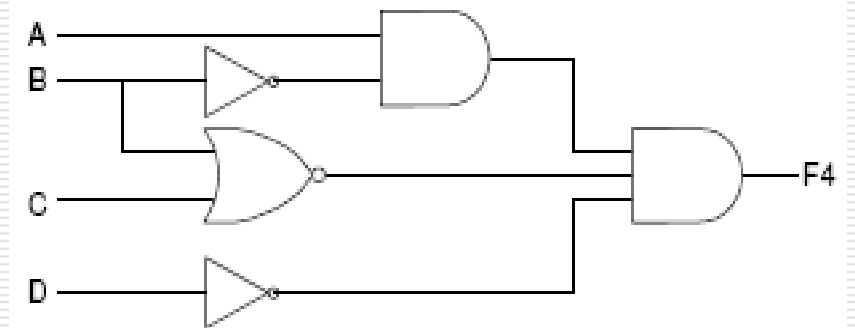
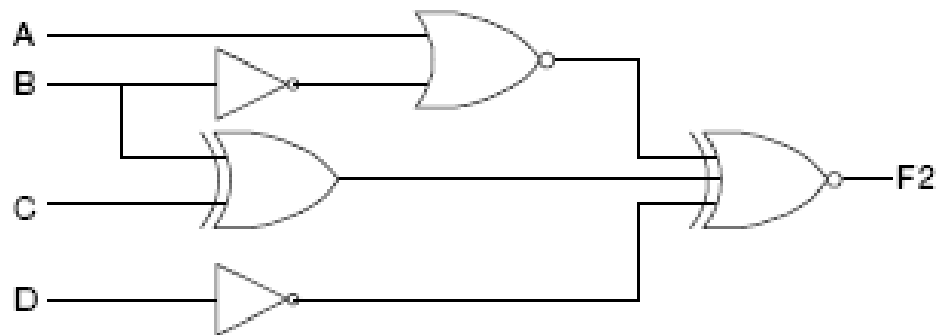
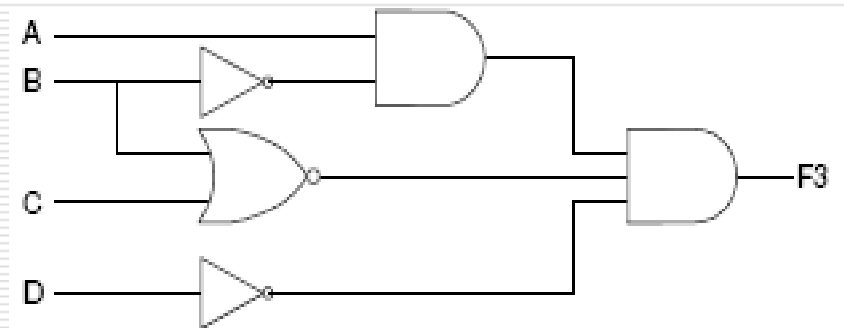
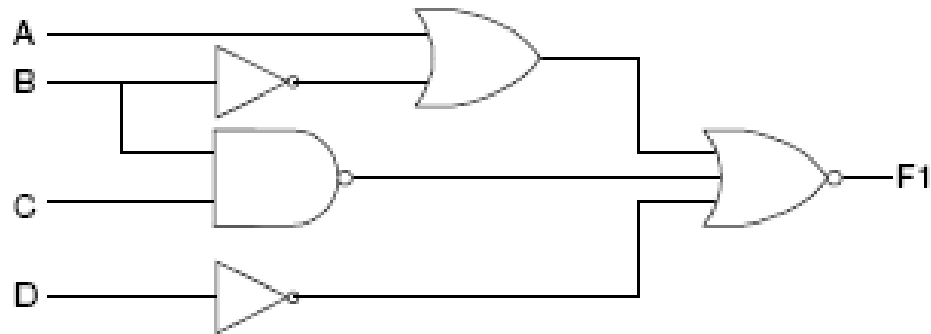
Exercices

Donner la table de vérité ainsi que le diagramme de portes logiques pour les équations booléennes suivantes:

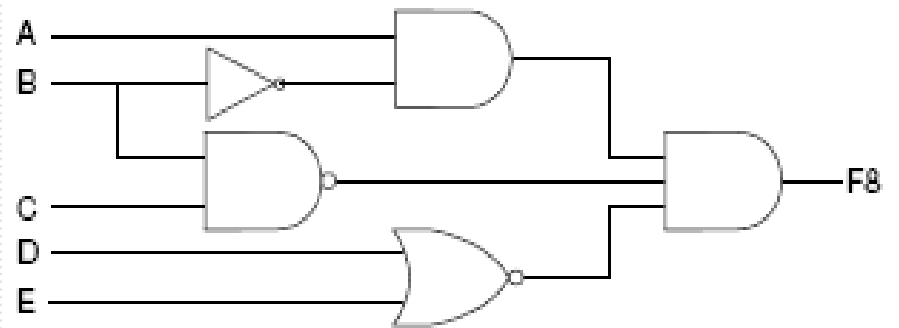
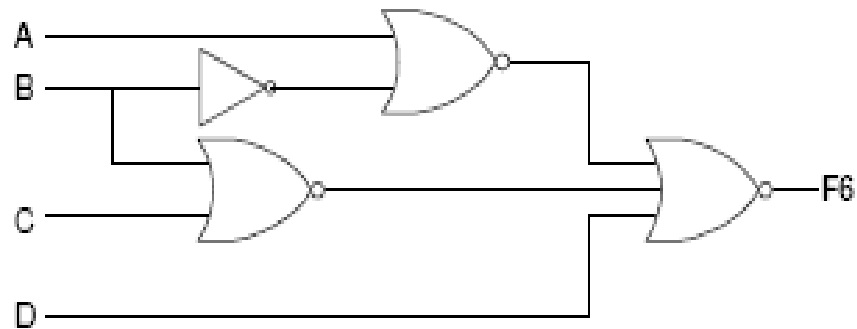
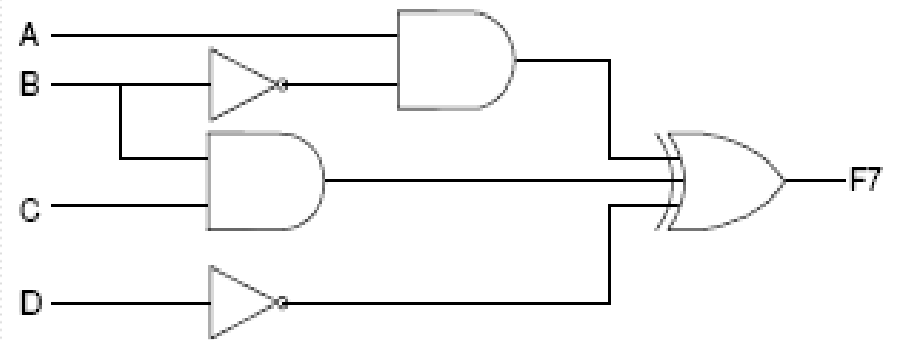
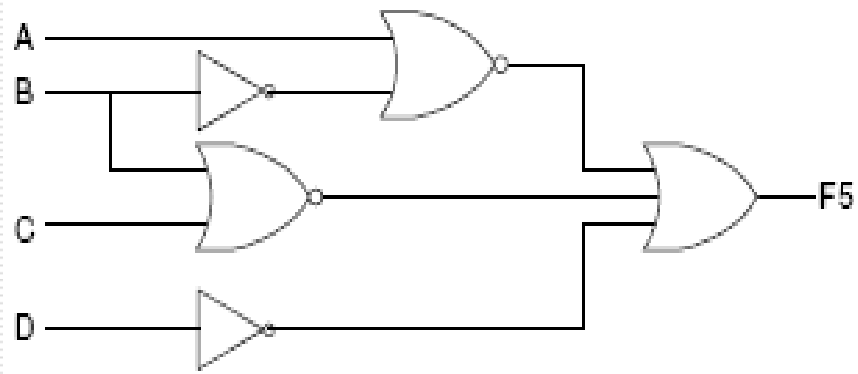
- ◆ $F = A + B'$
- ◆ $F = A'B$
- ◆ $F = A(B + C)$
- ◆ $F = AB + C$
- ◆ $F = A + B'C$
- ◆ $F = AB + B'C + ABC + C$

Exercices

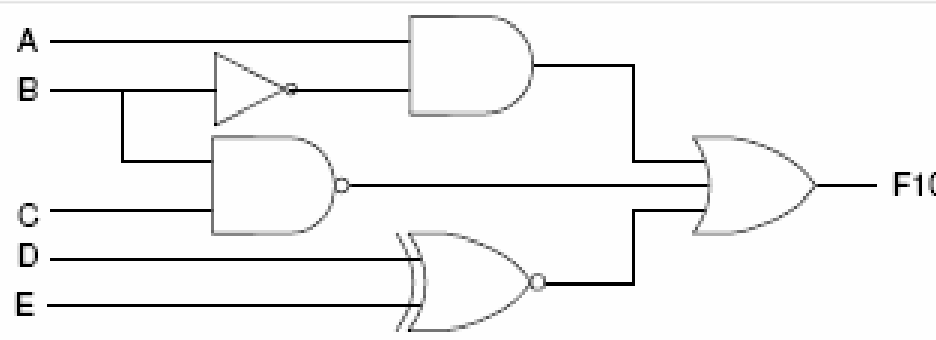
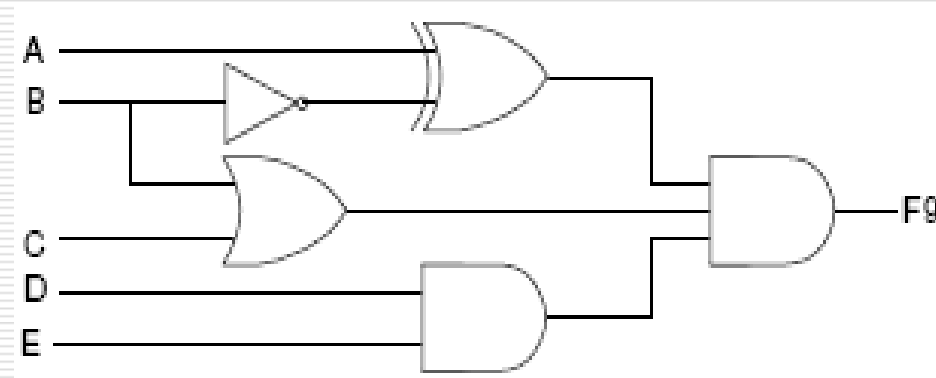
Pour chacun des circuits, donner l'expression logique correspondant à la fonction logique F.



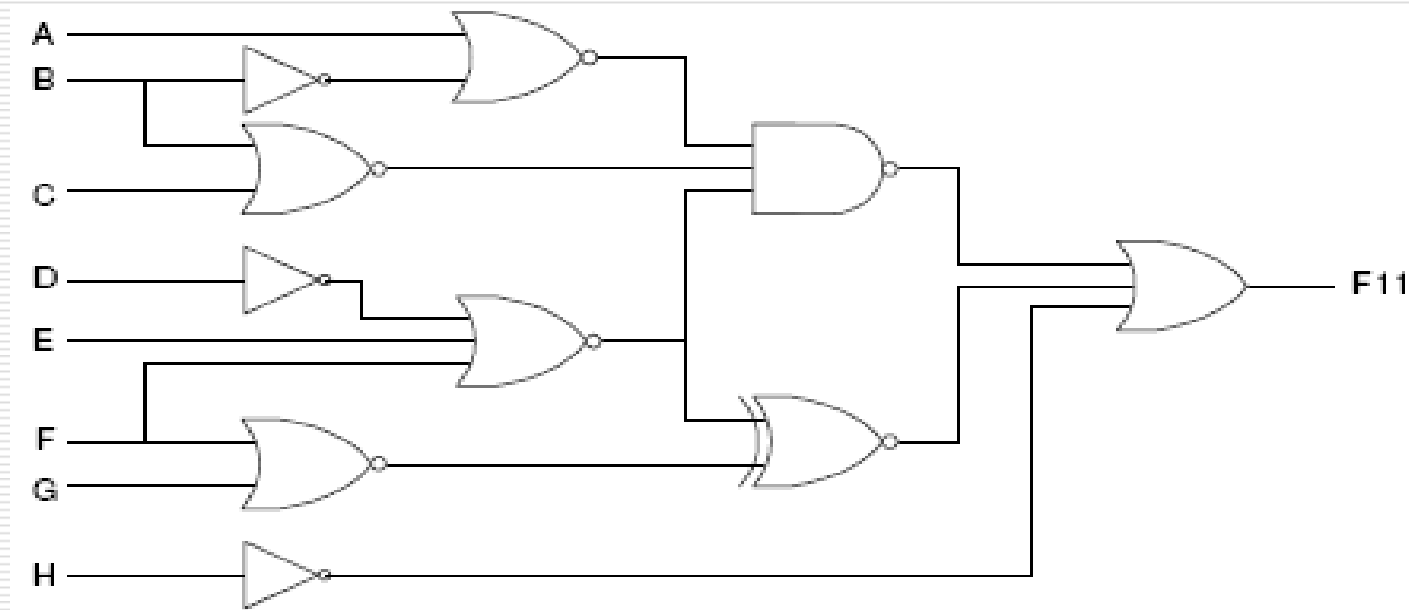
Exercices



Exercices



Exercices



Exercices

Dans chaque cas, donner :
le produit de sommes de la fonction F
la somme de produits de la fonction F
le produit de sommes de la fonction F'
la somme de produits de la fonction F'

#	A	B	C	$F1$
0	0	0	0	1
1	0	0	1	0
2	0	1	0	1
3	0	1	1	0
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

#	A	B	C	$F2$
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	0
6	1	1	0	0
7	1	1	1	1

Exercices

#	<i>A</i>	<i>B</i>	<i>C</i>	<i>F3</i>
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	0
4	1	0	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	1	0

#	<i>A</i>	<i>B</i>	<i>F4</i>
0	0	0	1
1	0	1	0
2	1	0	1
3	1	1	1

#	<i>A</i>	<i>B</i>	<i>F5</i>
0	0	0	1
1	0	1	1
2	1	0	1
3	1	1	1

Exercices

#	A	B	C	D	F6
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	1
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	0

Exercices

Prouver les théorèmes 1, 2, 3, 4 et 5.

1. $X + 0 = X$

1D. $X \cdot 1 = X$

2. $X + 1 = 1$

2D. $X \cdot 0 = 0$

3. $X + X = X$

3D. $X \cdot X = X$

4. $(X')' = X$

5. $X + X' = 1$

5D. $X \cdot X' = 0$

Exercices

□ Commutativité, associativité et distributivité

□ **6** : (Commutativité) $X + Y = Y + X$

□ **6D** : (Commutativité) $XY = YX$

#	X	Y	X + Y	Y + X
0	0	0		
1	0	1		
2	1	0		
3	1	1		

#	X	Y	XY	YX
0	0	0		
1	0	1		
2	1	0		
3	1	1		

Exercices

7 : (Associativité) $(X + Y) + Z = X + (Y + Z)$

7 D: (Associativité) $(XY)Z = X(YZ)$

#	X	Y	Z	$X+Y$	$(X+Y)+Z$	$Y+Z$	$X+(Y+Z)$
0	0	0	0				
1	0	0	1				
2	0	1	0				
3	0	1	1				
4	1	0	0				
5	1	0	1				
6	1	1	0				
7	1	1	1				

#	X	Y	Z	XY	$(XY)Z$	YZ	$X(YZ)$
0	0	0	0				
1	0	0	1				
2	0	1	0				
3	0	1	1				
4	1	0	0				
5	1	0	1				
6	1	1	0				
7	1	1	1				

Exercices

8 : (Distributivité) $X(Y + Z) = XY + XZ$

#	X	Y	Z	$Y + Z$	$X(Y + Z)$	XY	XZ	$XY + XZ$
0	0	0	0					
1	0	0	1					
2	0	1	0					
3	0	1	1					
4	1	0	0					
5	1	0	1					
6	1	1	0					
7	1	1	1					

8D : (Distributivité) $X + YZ = (X + Y)(X + Z)$

#	X	Y	Z	YZ	$X + (YZ)$	$X + Y$	$X + Z$	$(X+Y)(X+Z)$
0	0	0	0					
1	0	0	1					
2	0	1	0					
3	0	1	1					
4	1	0	0					
5	1	0	1					
6	1	1	0					
7	1	1	1					

Exercices

Prouver les théorèmes 9, 10, 11, 14 et 15 (théorème du consensus).

9. $XY + XY' = X$

9D. $(X + Y)(X + Y') = X$

10. $X + XY = X$

10D. $X(X + Y) = X$

11. $(X + Y)Y = XY$

11D. $XY' + Y = X + Y$

14. $(X + Y)(X' + Z) = XZ + X'Y$

14D. $XY + X'Z = (X + Z)(X' + Y)$

15. $XY + YZ + X'Z = XY + X'Z$

15D. $(X + Y)(Y + Z)(X' + Z) = (X + Y)(X' + Z)$

Inversion et lois de De Morgan

12. $(X + Y + Z + \dots)' = X'Y'Z' \dots$

12D. $(XYZ \dots)' = X' + Y' + Z' + \dots$

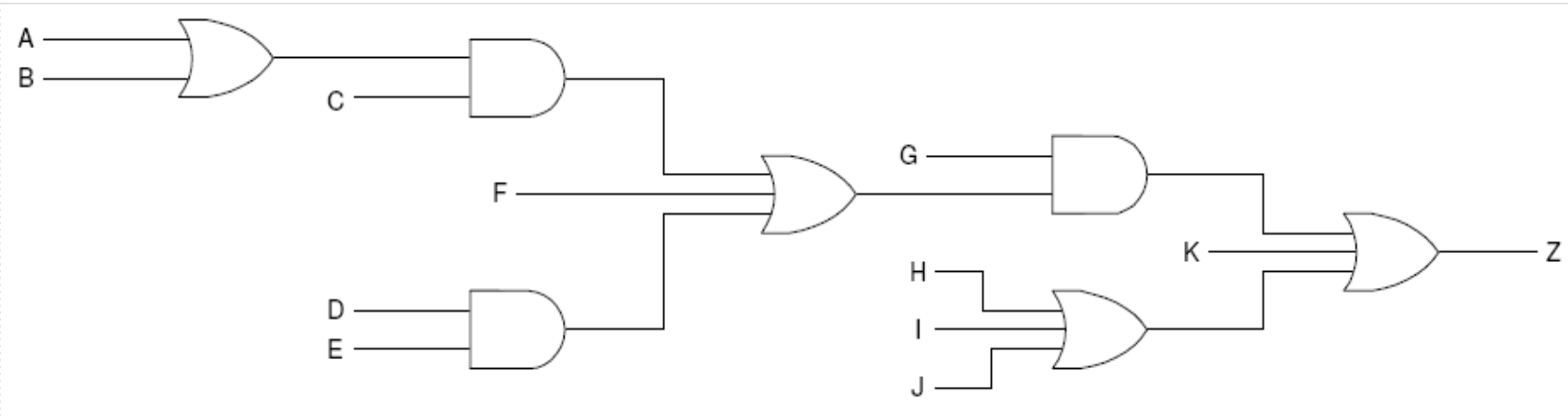
La forme inverse d'une expression booléenne est obtenue en faisant les substitutions suivantes:

chaque variable \rightarrow *son complément*, $ET \rightarrow OU$, $OU \rightarrow ET$, $0 \rightarrow 1$, $1 \rightarrow 0$

Prouver le théorème 12 pour deux variables.

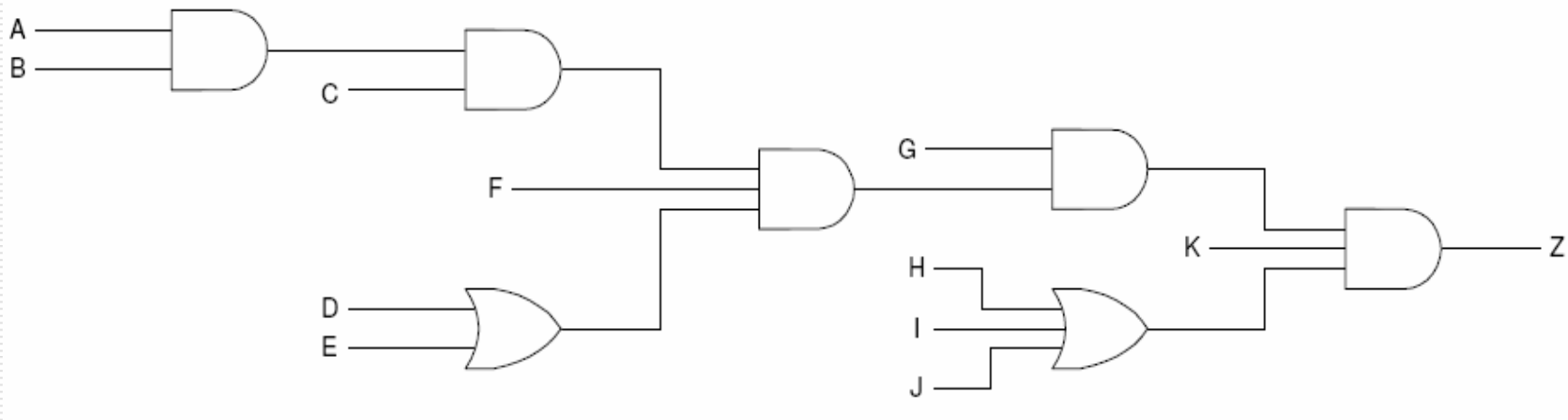
Exercices

Convertir chaque circuit pour n'utiliser que des portes NON-ET ou bien NON-OU.



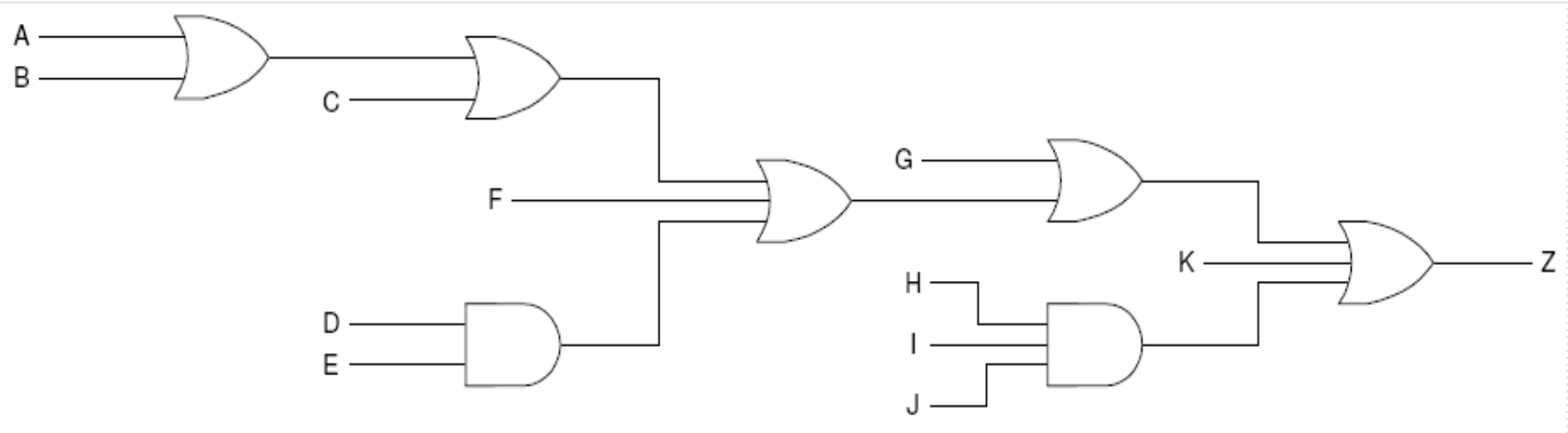
Exercices

Convertir chaque circuit pour n'utiliser que des portes NON-ET ou bien NON-OU.



Exercices

Convertir chaque circuit pour n'utiliser que des portes NON-ET ou bien NON-OU.



Exercices

Convertir chaque circuit pour n'utiliser que des portes NON-ET ou bien NON-OU.

