
ITI 1500

Hiver 2013

Systemes Numériques I

Cours

Lundi 11:30 - 13:00 SCS E217

Jeudi 13:00 - 14:30 SCS E217

TUTORIAL Mardi 8 :30-10:00 salle: LMX-221

Professeur : Dr A. Karmouch, Bureau: CBY-A508

Chapitre 3

Simplification des fonctions Logiques avec le Diagramme de Karnaugh

Diagramme de Karnaugh

- Une autre représentation des fonctions Booléennes
 - Utilisé pour simplifier les fonctions booléennes
 - Conversion Simple de la table de vérité vers le diagramme de Karnaugh
 - Facilite l'obtention de la fonction somme de produits.
 - Des étapes très simples permettent d'effectuer la simplification.
- Plus rapide et plus efficace que la simplification avec l'algèbre de Boole.

Le diagramme de Karnaugh

- Le diagramme est idéale pour 4 variable ou moins, mais non pratique pour 5 variables et plus
 - Chaque carrée représente un Minterm
 - *le diagramme est organisé de manière a ce que deux carrées adjacents différent seulement d'une variable (e.g. $ABC + ABC'$)*
 - *deux termes doivent être adjacent dans le diagramme*
 - un diagramme a n variables possède 2^n carrées
 - pour les fonctions booléennes les termes produits sont indiqués par des 1 et les termes somme sont indiqués par des 0 (représentés par des Minterms dans le diagramme).

Karnaugh avec 2 variables

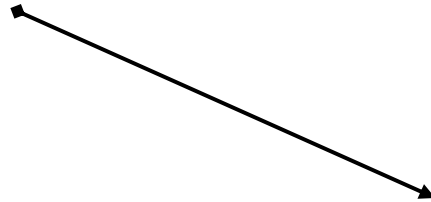
B \ A	A'	A
B'	A'B'	AB'
B	A'B	AB

B \ A	0	1
0	m ₀	m ₂
1	m ₁	m ₃

B \ A	0	1
0	00	10
1	01	11

3 variables

C	AB				
	A'B'	A'B	AB	AB'	
C'	A'B'C'	A'BC'	ABC'	AB'C'	
C	A'B'C	A'BC	ABC	AB'C	



C	AB				
	00	01	11	10	
0	0	2	6	4	
1	1	3	7	5	

4 variables

CD \ AB		A			
		00	01	11	10
C	00	0	4	12	8
	01	1	5	13	9
	11	3	7	15	11
	10	2	6	14	10

The table is annotated with curly braces: a vertical brace on the left labeled 'C' spans the four rows; a horizontal brace at the bottom labeled 'B' spans the four columns; and a vertical brace on the right labeled 'D' spans the four rows.

Karnaugh- les sorties et la valeur des carrées

Mettre les sorties de la fonction pour chaque Minterm dans le carrée correspondant

$$F(A,B,C,D) = A'B'C'D' + A'BC'D' + AB'C'D' + A'BC'D + ABC'D + ABCD' + AB'CD'$$

		AB			
		00	01	11	10
CD	00	1	1	0	1
	01	0	1	1	0
	11	0	0	0	0
	10	0	0	1	1

→ Pour simplifier une **somme de produits** on s'intéressera aux carrées à "1"

→ Pour simplifier **un produit de sommes** on s'intéressera aux carrées à "0"

Karnaugh- groupage des carrées a 1

- **des groupes de carrées sont formés selon les règles suivantes:**
 - chaque carrée qui contient 1 doit être considéré au moins une fois
 - un carrée a 1 peut être inclus dans différent groupes autant de fois désiré
 - Un groupe de carrée doit être le plus large possible
 - *le nombre de carrée dans un groupe doit être 2^n , i.e. 2,4,8,...*
- **L'expression simplifiée obtenu a partir du diagramme de K n'est pas unique, le groupage peut se faire de plusieurs manières**

2 variables- Karnaugh

	A	A'	A
B	B'	A'B'	AB'
	B	A'B	AB

→

	A	0	1
B	0	00	10
	1	01	11

$$F = AB' + AB$$

$$F = A'B' + A'B$$

	A	A
B		1
B		1

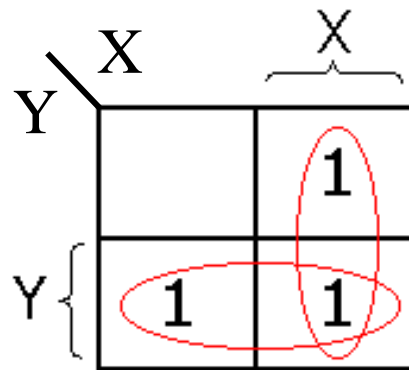
$$F = A$$

	A	
B	1	
B	1	

$$F = A'$$

2 variables -Examples

$$F = X'Y + XY + XY'$$



$$F = X + Y$$

3 Variables-Exemples

AB

C	A'B'C'	A'BC'	ABC'	AB'C'
	A'B'C	A'BC	ABC	AB'C

	AB	00	01	11	10
C	0	0	2	6	4
	1	1	3	7	5

B

$$F = XY'Z' + XYZ'$$

$$F = X'YZ' + XYZ + X'YZ$$

	YZ	00	01	11	10
X	0			1	1
	1			1	

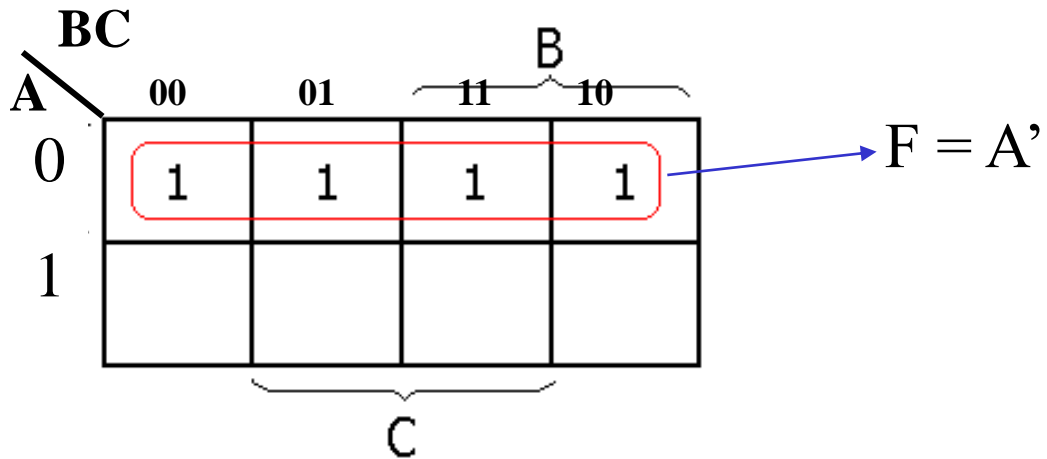
$$F = X'Y + YZ$$

	YZ	00	01	11	10
X	0				
	1	1			1

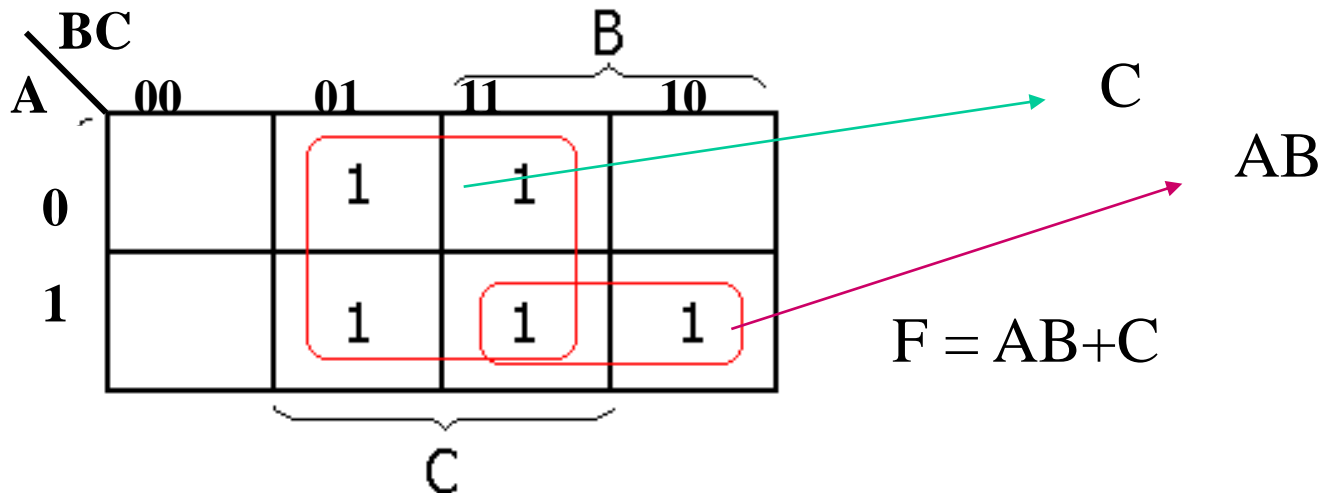
$$F = XZ'$$

3 variables - Examples

$$F(A,B,C) = A'BC' + A'B'C' + A'BC + A'B'C$$



$$F(A,B,C) = A'BC + A'B'C + AB'C + ABC + ABC'$$



4 variables-Exemple

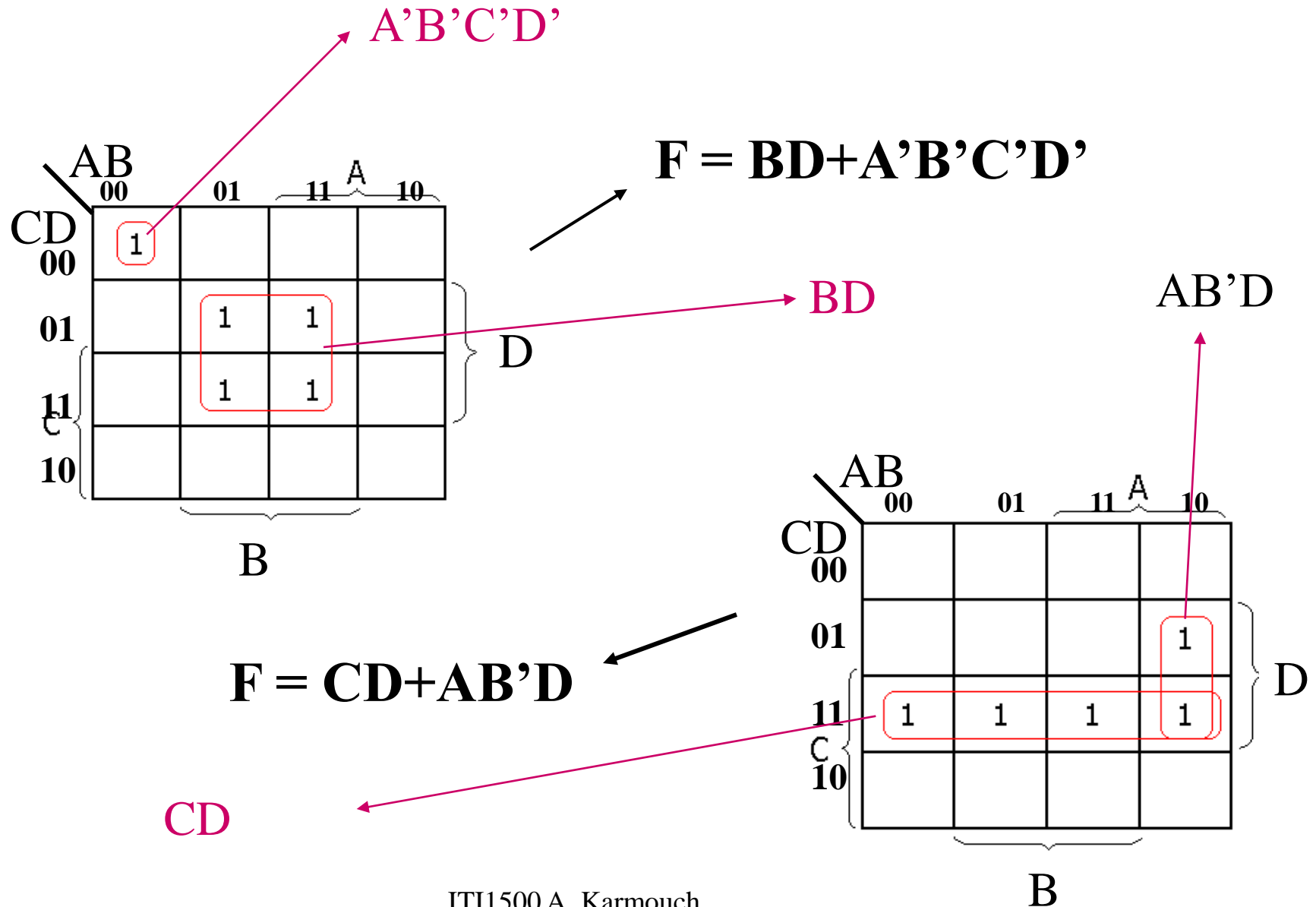
CD		AB		A	
		00	01	11	10
C	00	0	4	12	8
	01	1	5	13	9
	11	3	7	15	11
	10	2	6	14	10

B

D

Variables A,B,C,D

4 variables-Exemple



Fonction avec des sorties « indéterminée »

•Exemple

Un nombre en binaire pure est a convertir en un nombre de 5-4-2-1 DCB.
La table de vérité est la suivante.

$\Sigma d(10,11,12,13,14,15)$

Sorties indéterminées

Pour W, X, Y,Z

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

} Sorties indéterminées

Karnaugh- sorties « indéterminée »

- Les termes indéterminé sont dénotés par un « X »
- X est utilisée pendant le groupage pour augmenter soit le nombre de 1 soit le nombre de 0 dans un groupe.
- Exemple: $F(A,B,C,D) = \Sigma m(1,3,7,11,15)$ avec $\Sigma d(0,2,5)$

CD \ AB	00	01	11	10
00	X	1	1	X
01	0	X	1	0
11	0	0	1	0
10	0	0	1	0

$$F = A'D + CD$$

Solution du décodeur 5-4-2-1 BCD exemple

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

après simplifications on obtient:

$$W = A + BD + BC$$

$$X = BC'D' + AD$$

$$Y = CD + B'C + AD'$$

$$Z = AD' + A'B'D + BCD'$$

Sortie indéterminées

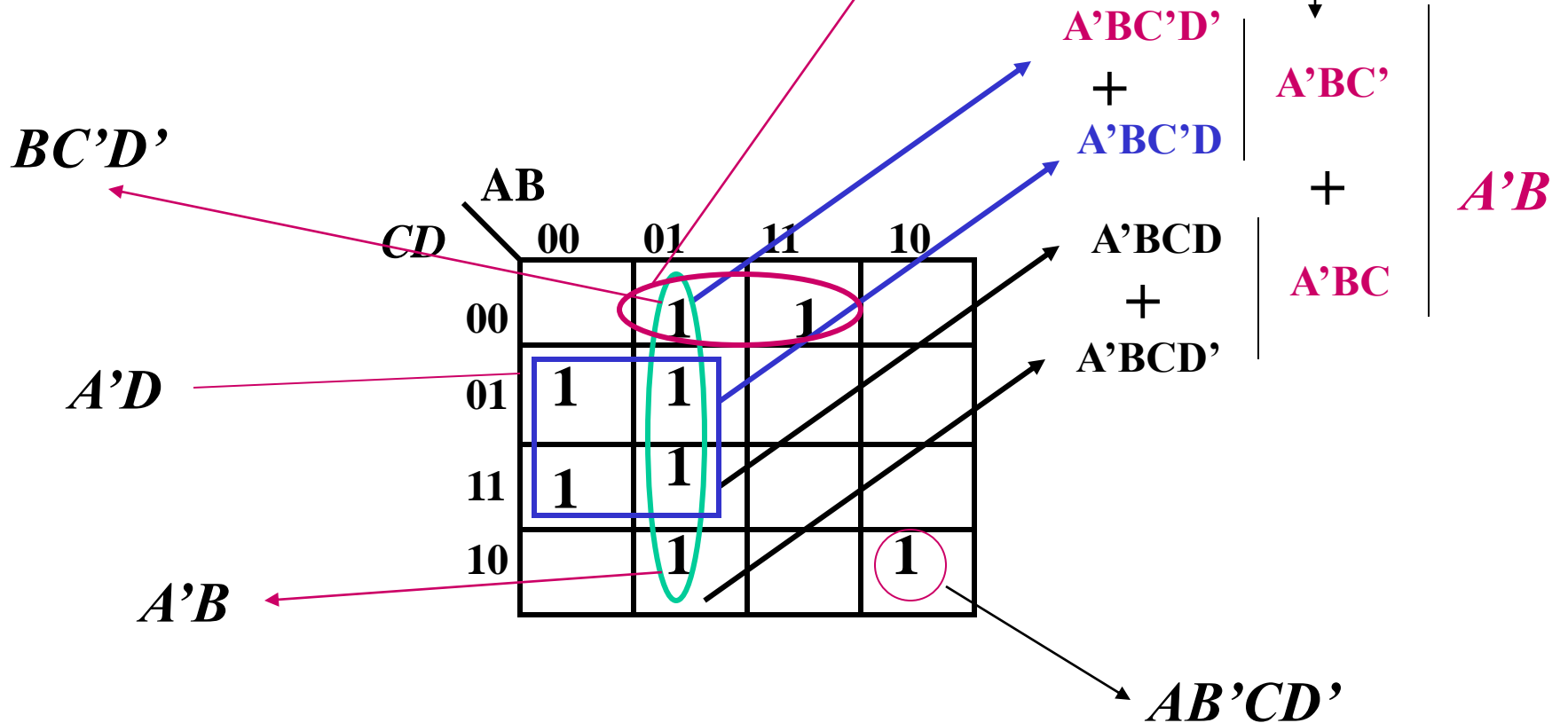
Diagramme de Karnaugh - Exemples

1- Simplifier la fonction suivante en utilisant le Diagramme de Karnaugh $F(A,B,C,D) = \sum m_i (1, 3, 4, 5, 6, 7, 10, 12)$

a) Construire le diagramme pour F

CD \ AB		A			
		00	01	11	10
C	00	0	4 1	12 1	8
	01	1 1	5 1	13	9
	11	1 3	7 1	15	11
	10	2	6 1	14	1 10
		B		D	

b) Formation de groupes de carrés $A'BC'(D+D') = A'BC'$



c) Écrire la fonction simplifiée

$$F(A,B,C,D) = A'B + A'D + BC'D' + AB'CD'$$

a) Construire le diagramme a partir de la table de Vérité

<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>F</i>
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	x
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

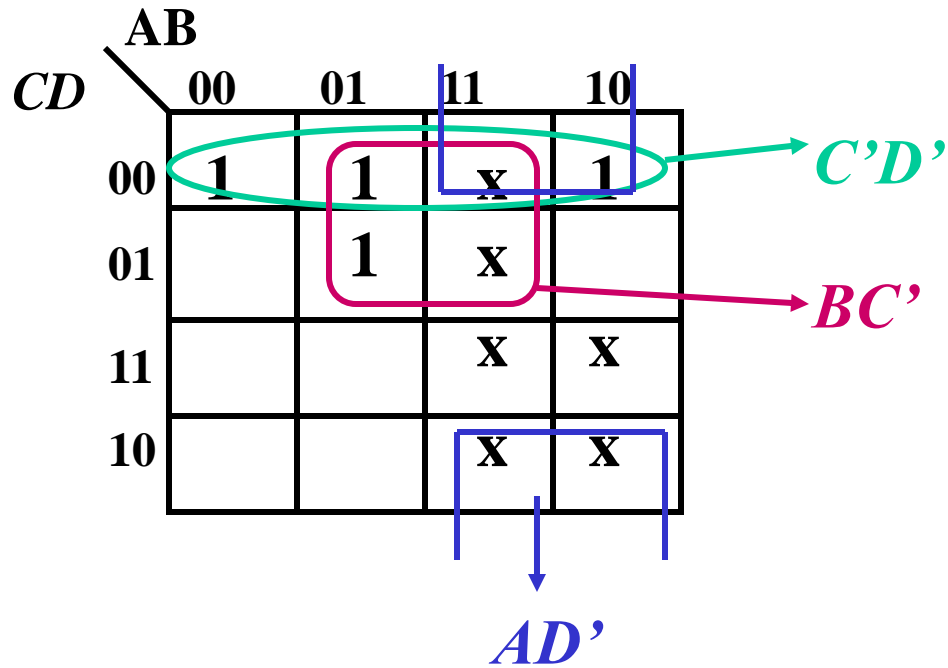
		<i>AB</i>			
		00	01	11	10
<i>CD</i>	00	1	1	x	1
	01	0	1	x	0
	11	0	0	x	x
	10	0	0	x	x

<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>F</i>
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	X
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

a) Construire le diagramme a partir de la table de Vérité

		AB			
		00	01	11	10
CD	00	1	1	X	1
	01	0	1	X	0
	11	0	0	X	X
	10	0	0	X	X

b) Obtenir la somme de produits pour F



$$F = BC' + C'D'$$

Impliquant premier

Lors du groupage faire attention:

- un groupe doit contenir le maximum de carrées adjacents
 - le groupe est appelée *Impliquant premier*
- chaque groupe représente un terme produit dans la fonction
- une fonction doit contenir un ensemble minimum de termes
- Lors de la composition de la fonction simplifiée :
 - Sélectionner les groupes qui contiennent au moins un carrée **non couvert** par d'autres groupes: *Impliquant Premier ESSENTIEL*
 - Ne pas sélectionner les groupes dont les carrées sont tous couverts par d'autres groupes: *Impliquant Premier Optionnel*

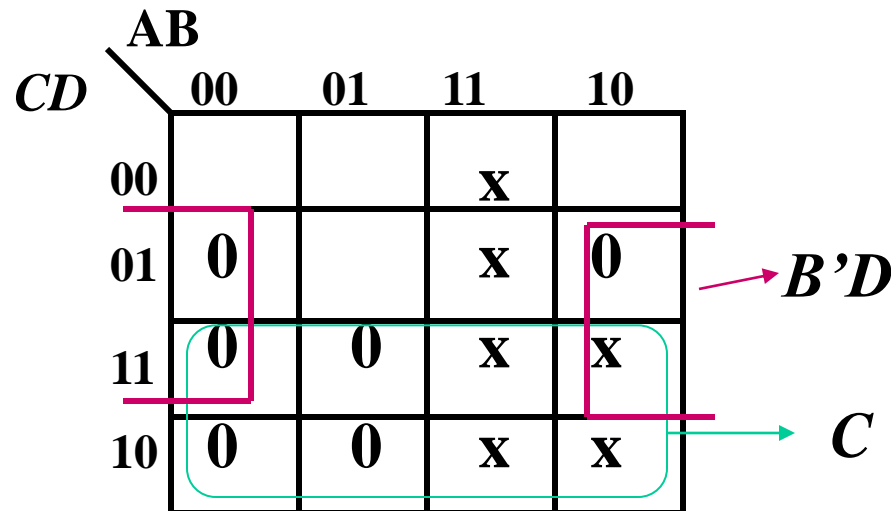
c) Obtenir le produits de sommes pour F : 2 ETAPES

1 – utiliser les Minterms pour simplifier F' et obtenir l'expression pour F' $F' = B'D + C$

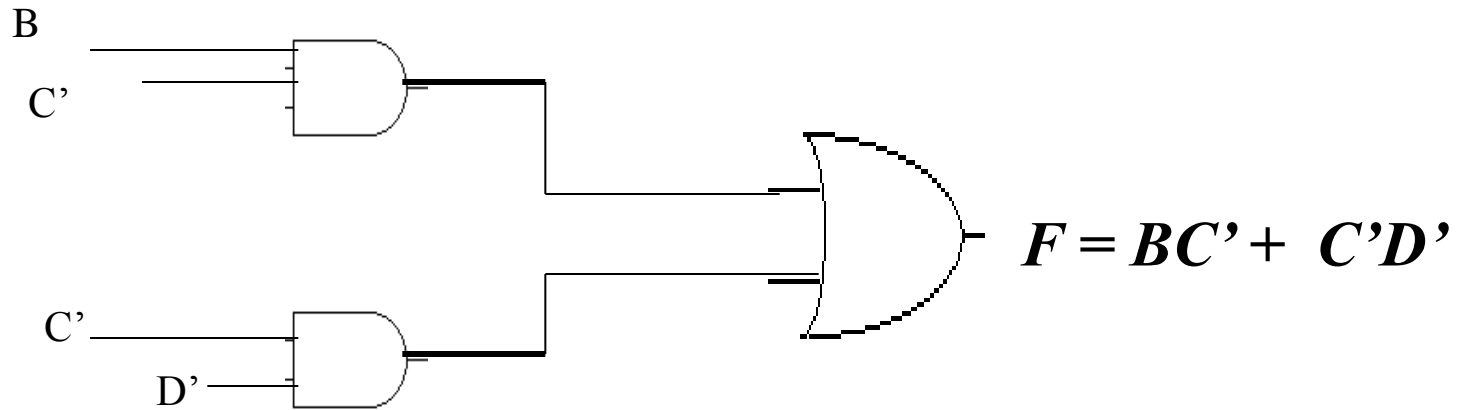
2 - complémenter F' pour obtenir la forme produit de sommes

$$F'' = (B'D + C)' = (B'' + D').C'$$

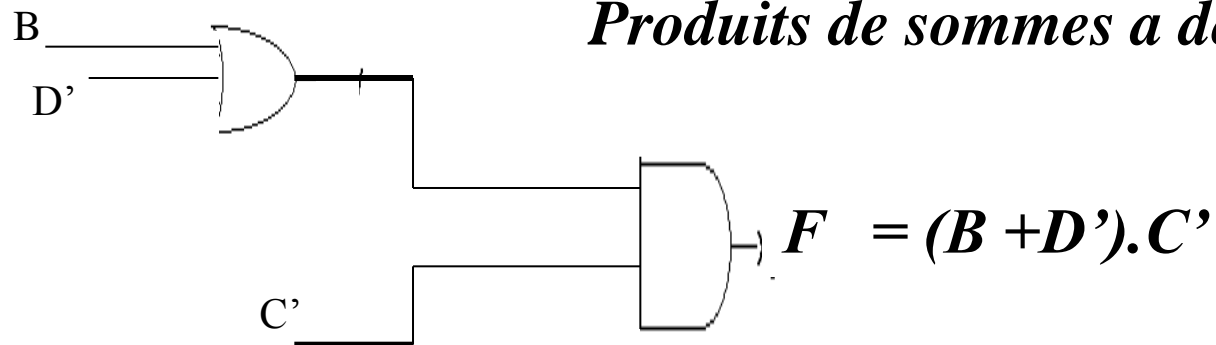
$$F = (B + D').C'$$



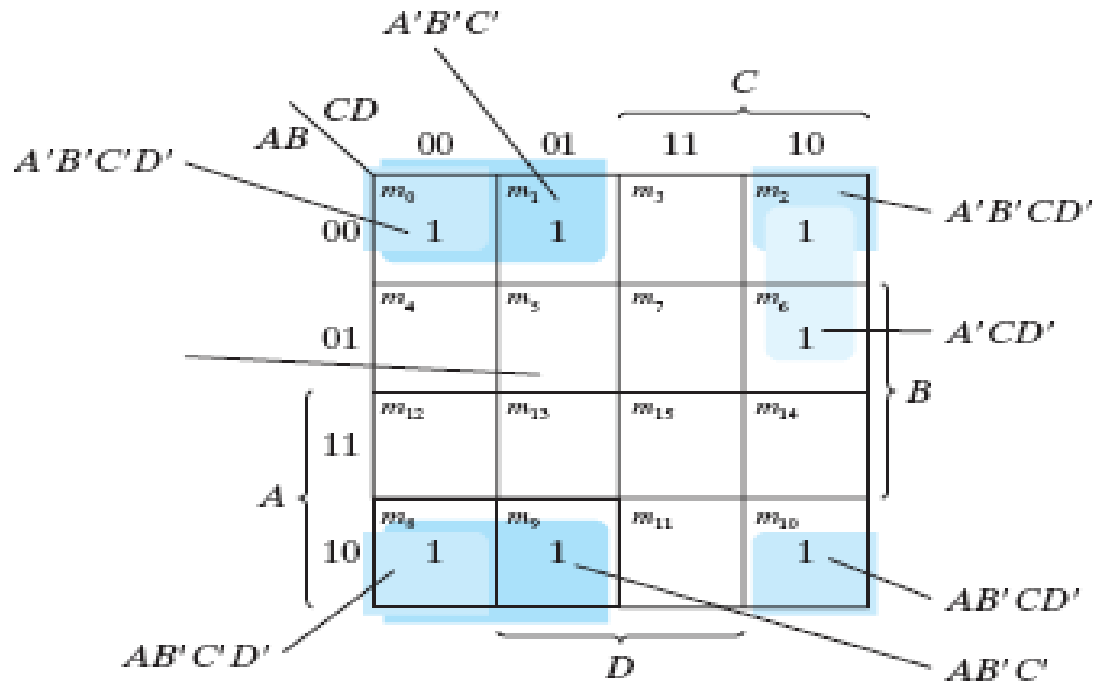
Somme de produits a deux niveaux



Produits de sommes a deux niveaux



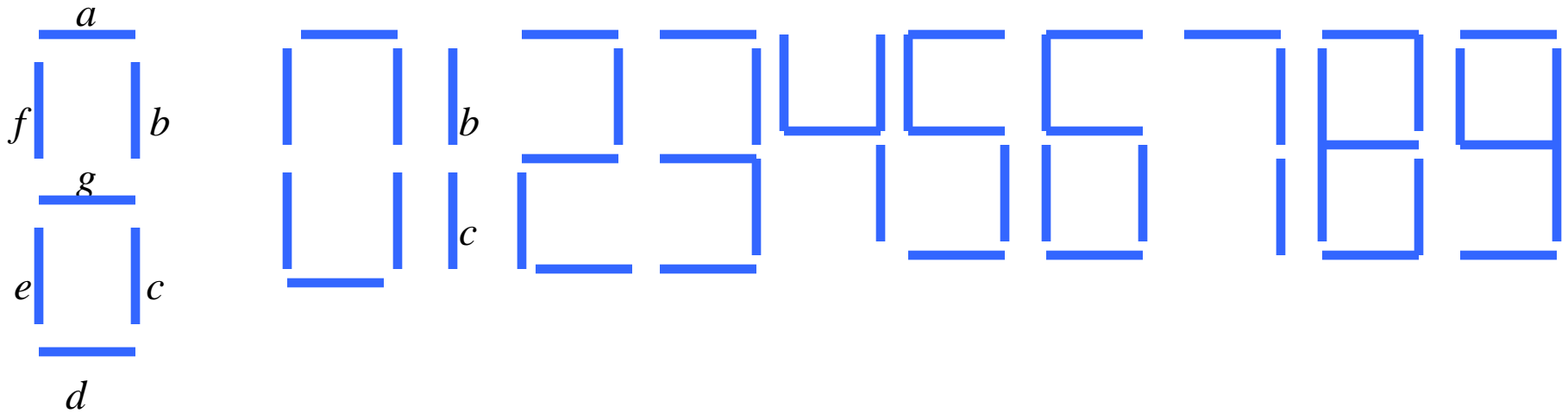
Attention a l'ordre des variables dans la table de K



Note: $A'B'C'D' + A'B'CD' = A'B'D'$
 $AB'C'D' + AB'CD' = AB'D'$
 $A'B'D' + AB'D' = B'D'$
 $A'B'C' + AB'C' = B'C'$

le Décodeur a 7 segments -Exemple

Un décodeur DCB vers sept Segments accepte en entrée une décimal codé binaire et le convertit en sa représentation en segment pour affichage en sortie

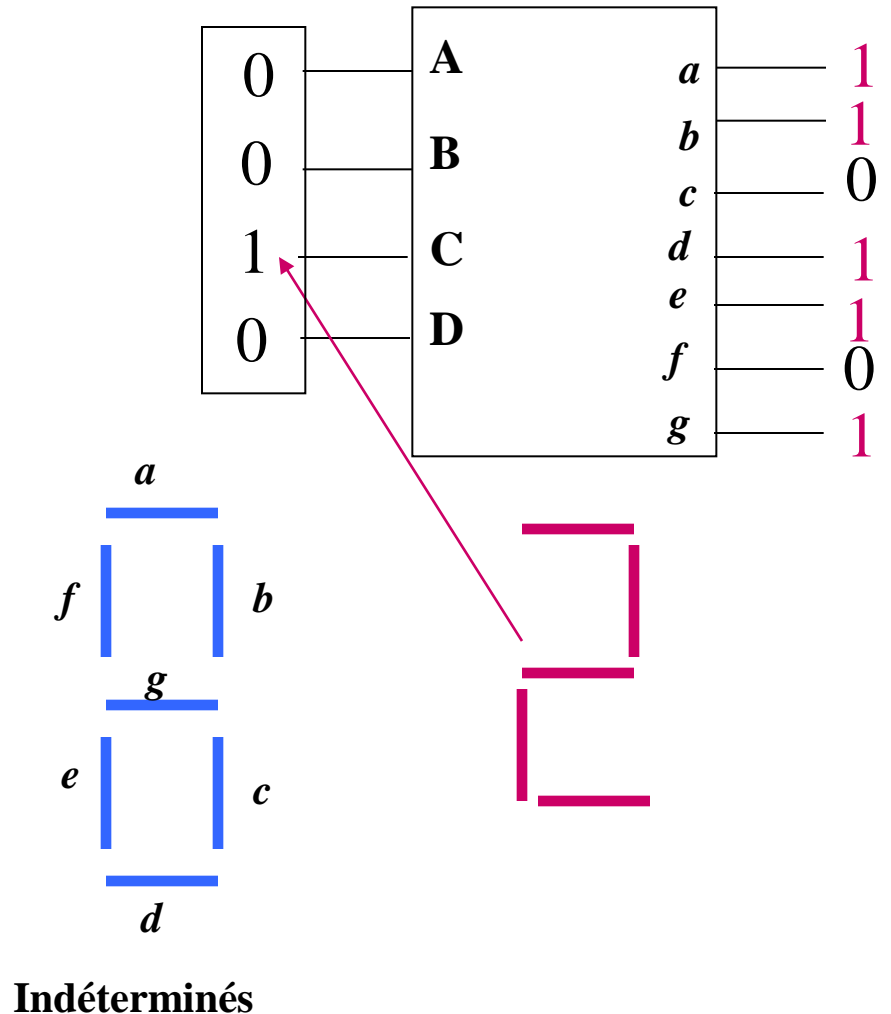


(a) Désignation du segment

(b) désignation numérique pour affichage

A- BCD to Seven Segment Decoder

A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
1	0	1	0	x	x	x	x	x	x	x
1	0	1	1	x	x	x	x	x	x	x
1	1	0	0	x	x	x	x	x	x	x
1	1	0	1	x	x	x	x	x	x	x
1	1	1	0	x	x	x	x	x	x	x
1	1	1	1	x	x	x	x	x	x	x



		CD			
		00	01	11	10
AB	00	1		1	1
	01		1	1	1
	11				
	10	1	1		

		CD			
		00	01	11	10
AB	00	1	1	1	1
	01	1		1	
	11				
	10	1	1		

$a =$

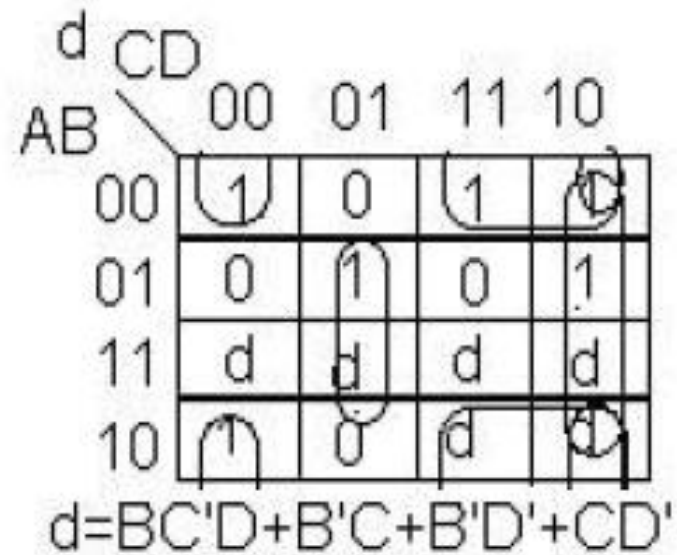
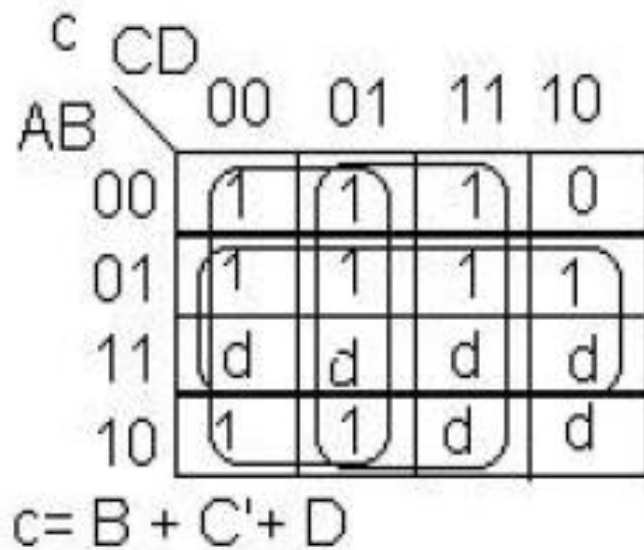
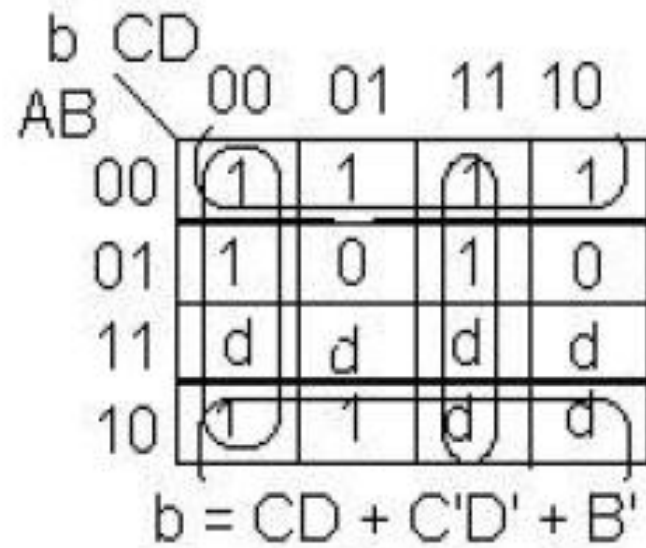
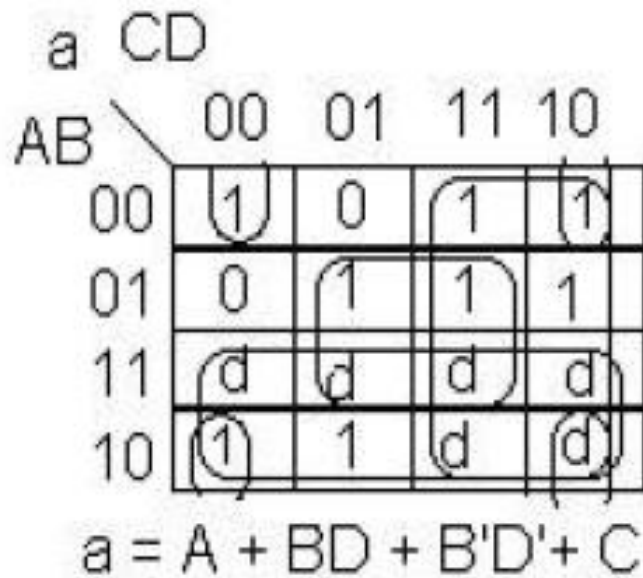
		CD			
		00	01	11	10
AB	00	1	1	1	
	01	1	1	1	1
	11				
	10	1	1		

$b =$

		CD			
		00	01	11	10
AB	00	1		1	1
	01		1		1
	11				
	10	1	1		

$c =$

$d =$



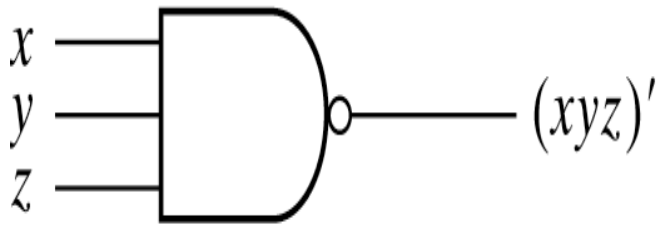
g		CD			
		00	01	11	10
AB	00	0	0	1	1
	01	1	1	0	1
	11	d	d	d	d
	10	1	1	d	d

$$g = BC' + B'C + CD' + A$$

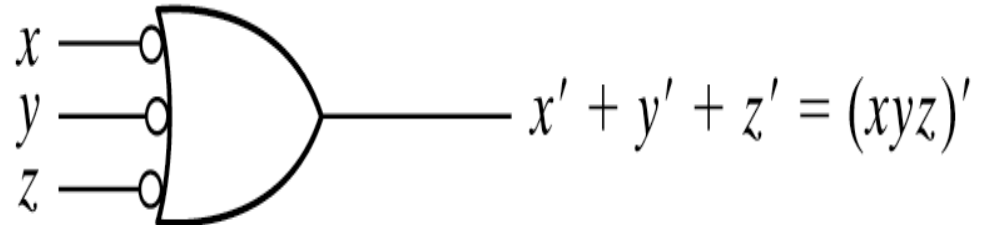
Example

- Construire un circuit qui permet de convertir les chiffres décimaux code suivant le code 8, 4, -2, -1 en code 8, 4, 2, 1
 - Construire la table de vérité
 - Simplifie la fonction somme de produits avec le diagramme de Karnaugh

Les Opérations Logiques avec NON-ET (NAND)



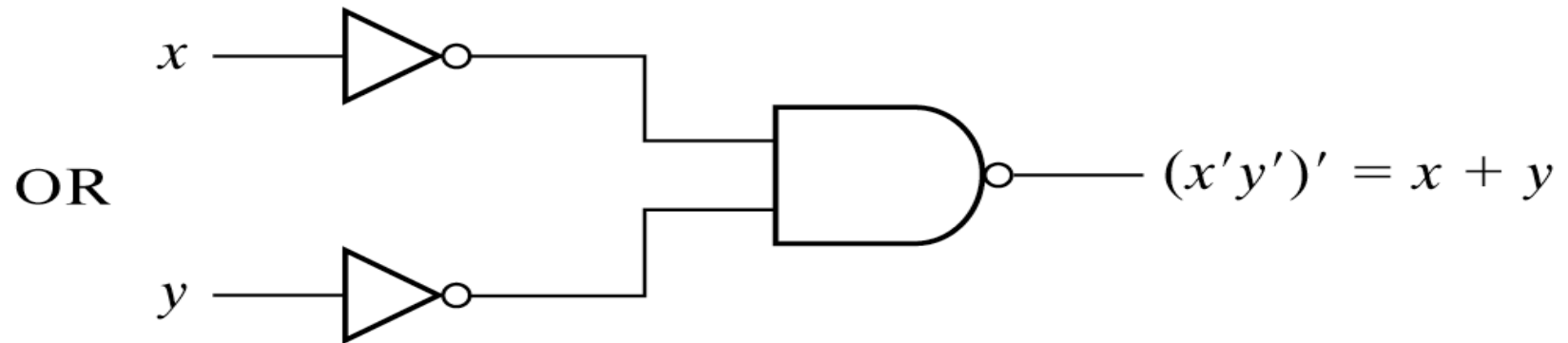
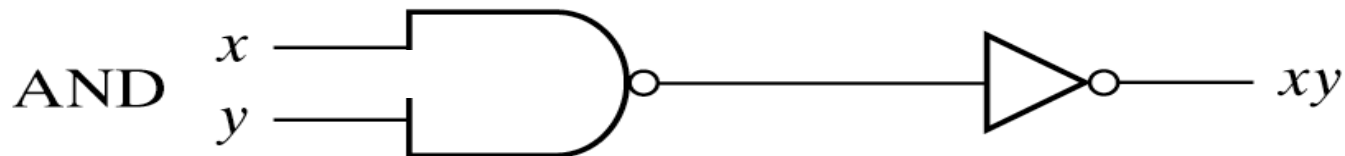
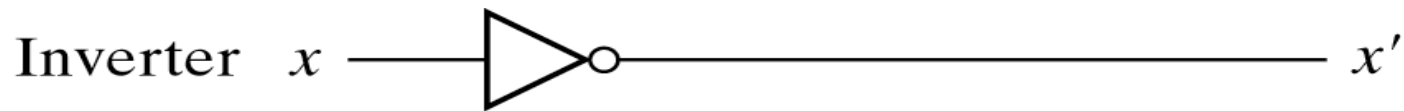
(a) AND-invert



(b) Invert-OR

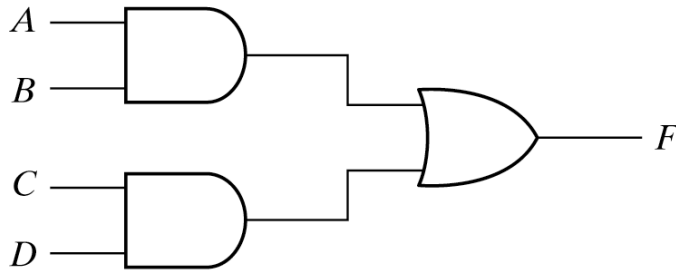
Deux symboles pour représenter NON-ET

Les Opérations Logiques avec NON-ET (NAND)

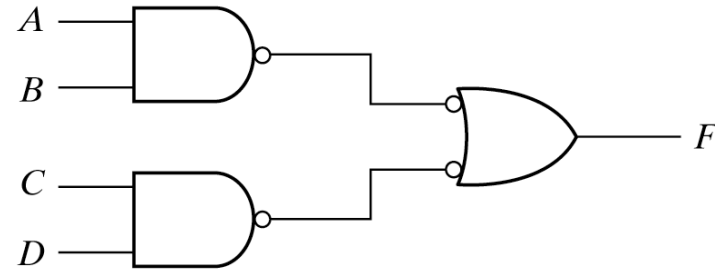


Les Opérations Logiques avec NON-ET (NAND)

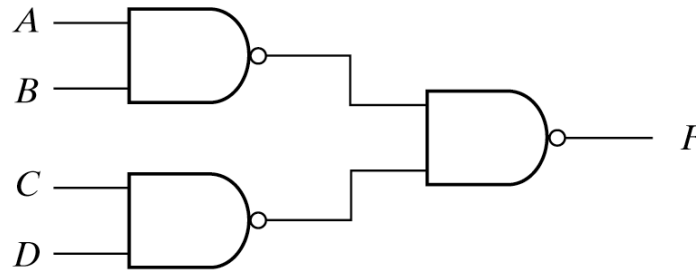
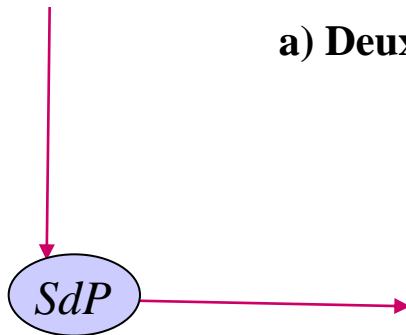
Réalisation avec des portes NON-ET



a) Deux niveaux avec ET-OU



b) Deux niveaux NON-ET & Inversion -OU



c) Deux niveaux avec NON-ET

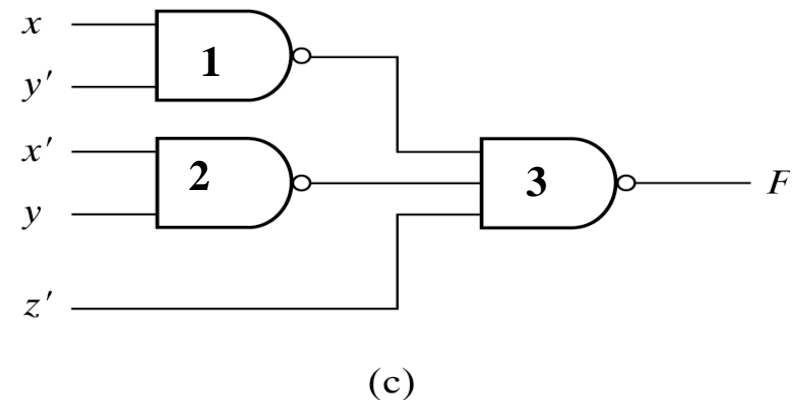
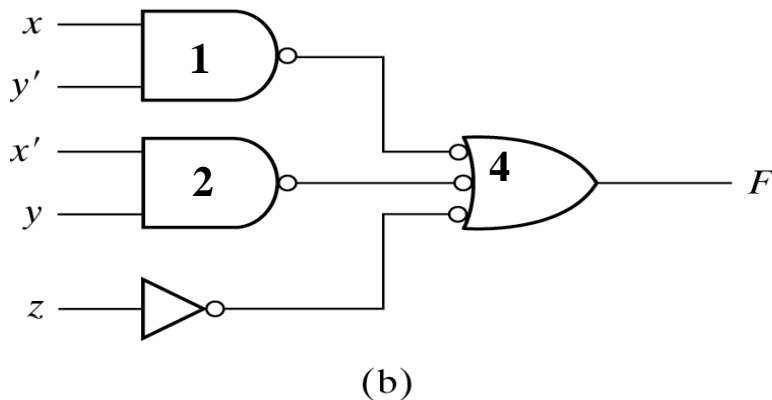
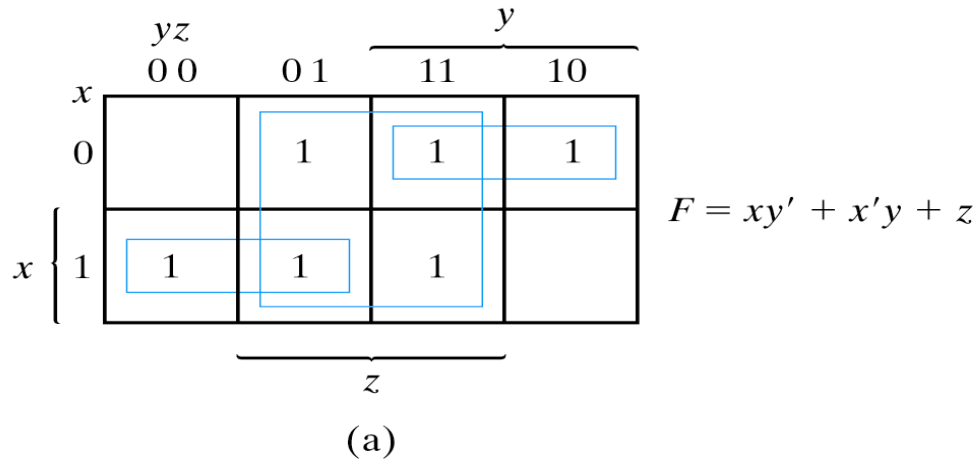
(utiliser ceci a l'exam)

3 manières de réaliser $F = AB + CD$

Réalisation avec des portes NON-ET -Exemples

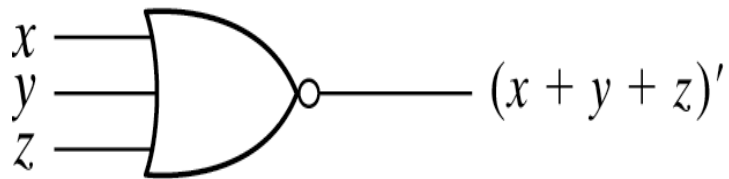
1: $(xy)'$ 2: $(x'y)'$ 4: $((xy)')' + ((x'y)')' + (z')' = xy' + x'y + z$

3: $((xy)')(x'y)(z')$ = $(xy)'' + (x'y)'' + (z')' = xy' + x'y + z$

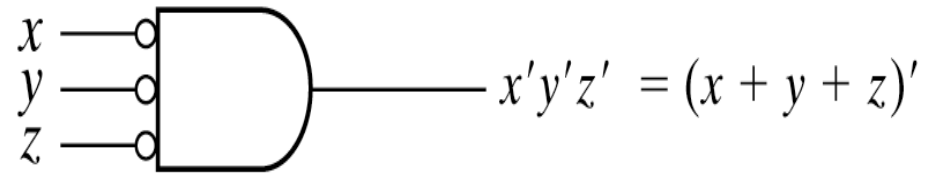


Using NAND gates to implement SOP

Les Opérations Logiques avec NON-OU (NOR)



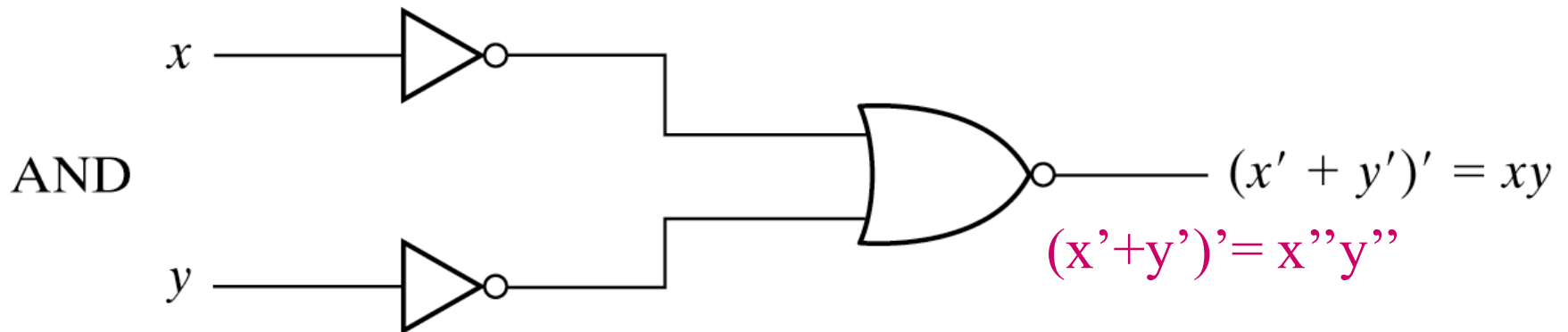
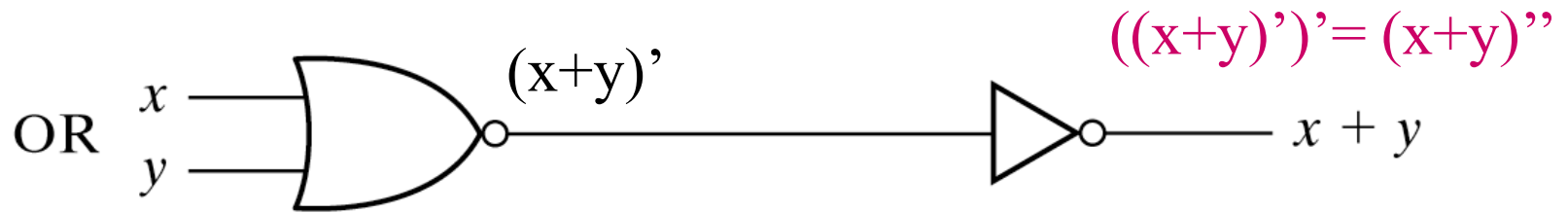
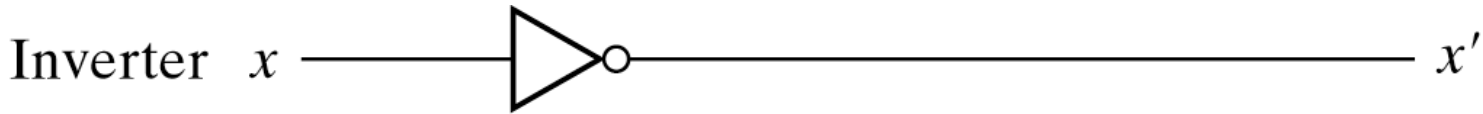
(a) OR-invert



(a) Invert-AND

Les Opérations Logiques avec NON-OU (NOR)

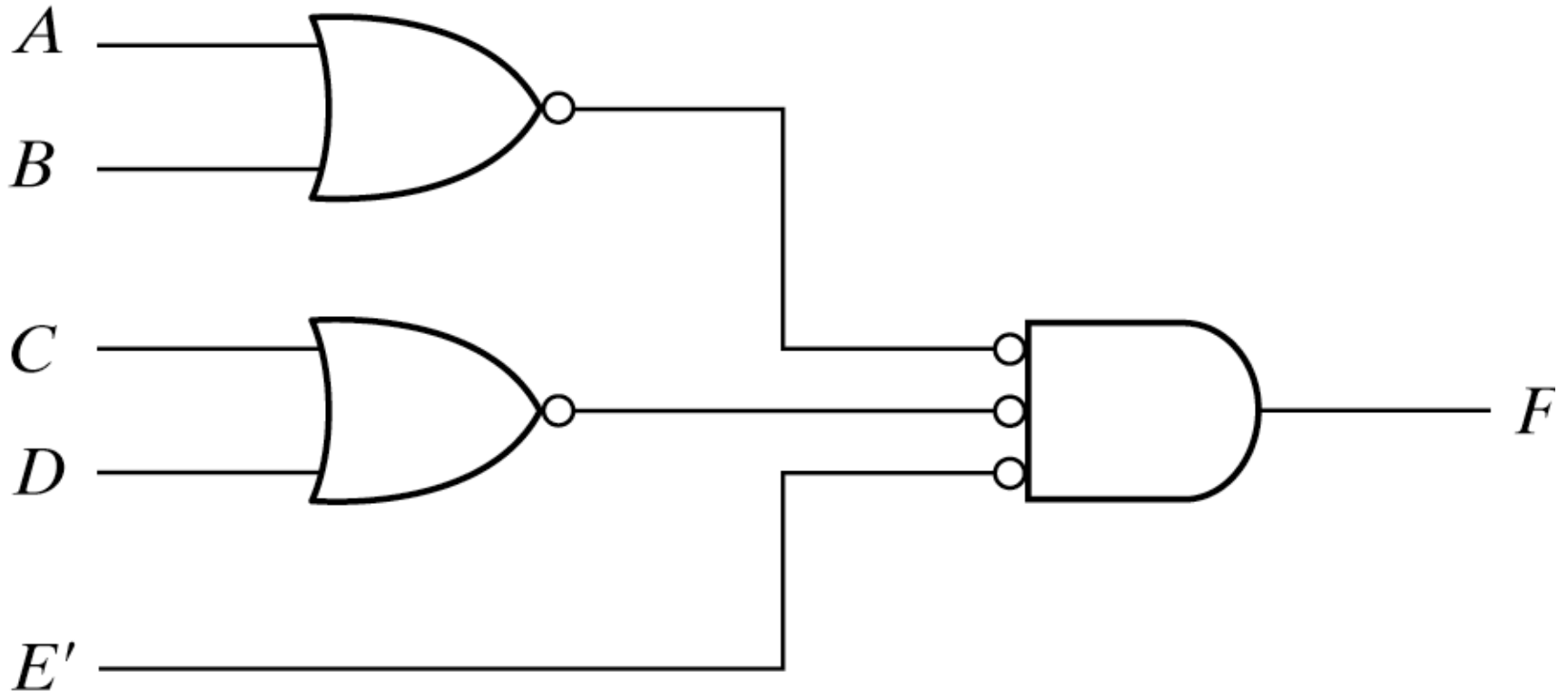
Les Opérations Logiques avec NON-OU (NOR)



Les Opérations Logiques avec NON-OU (NOR)

Réalisation avec des portes NON-ET- exemples

PdS avec des NON-OU



Réalisation de $F = (A + B)(C + D)E$

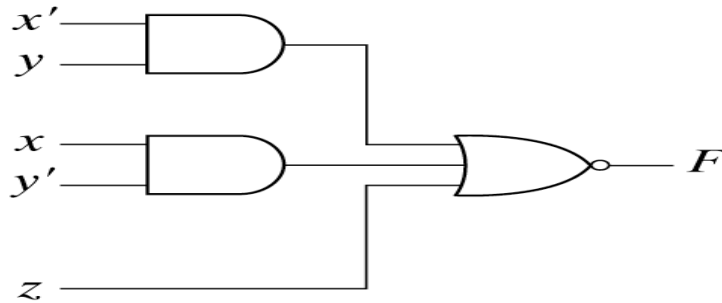
Autres réalisations a deux niveaux -Exemples

		yz		y	
		0 0	0 1	1 1	1 0
x	0	1	0	0	0
	1	0	0	0	1
		z			

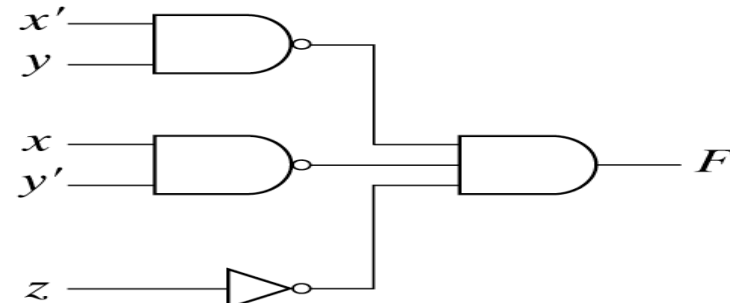
$$F = x'y'z' + xyz'$$

$$F' = x'y + xy' + z$$

(a) Map simplification in sum of products.

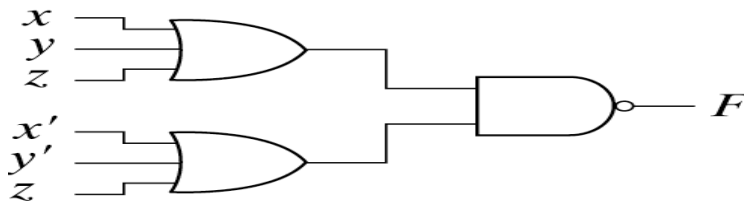


AND-NOR

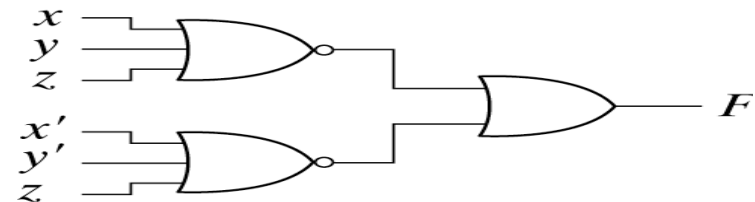


NAND-AND

(b) $F = (x'y + xy' + z)'$



OR-NAND



NOR-OR

(c) $F = [(x + y + z)(x' + y' + z)]'$

Other Two-level Implementations