Simplification des fonctions logiques

- Table de Karnaugh
- Fonctions complètement définies
- Fonctions incomplètement définies

andre.stauffer@epfl.ch

Simplification

Les diverses expressions algébriques qui représentent une même fonction sont dites équivalentes ou égales

Une <u>transformation</u> d'une expression algébrique est le passage de cette expression à une expression équivalente

On appelle <u>simplification</u> la transformation qui correspond au passage de la forme canonique à un <u>polynôme contenant le nombre minimal de lettres</u>

Ce polynôme minimal conduit à des réalisations matérielles qui réduisent le nombre de portes logiques du circuit

Table de Karnaugh

La table de Karnaugh est un mode de représentation des fonctions logiques qui permet d'effectuer des simplifications graphiques
La table de Karnaugh à quatre variables présente deux variantes

D

A

B

C

DC

BA	00	01	11	10
00				
01				
11				
10				

Table de Karnaugh

Dans les cases des deux variantes de la table, on a respectivement représenté le numéro décimal de l'état D,C,B,A et son équivalent binaire

D

0	4	12	8
1	5	13	9
3	7	15	11
2	6	14	10

A

B

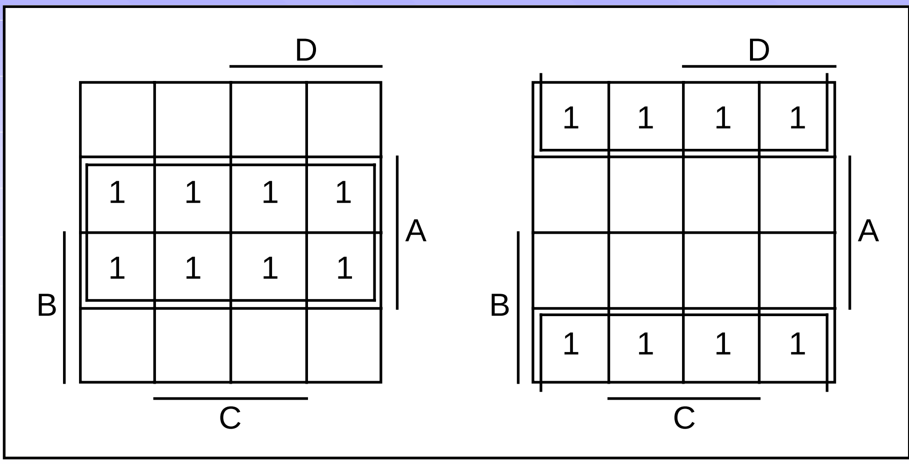
C

DC

BA	00	01	11	10
00	0000	0100	1100	1000
01	0001	0101	1101	1001
11	0011	0111	1111	1011
10	0010	0110	1110	1010

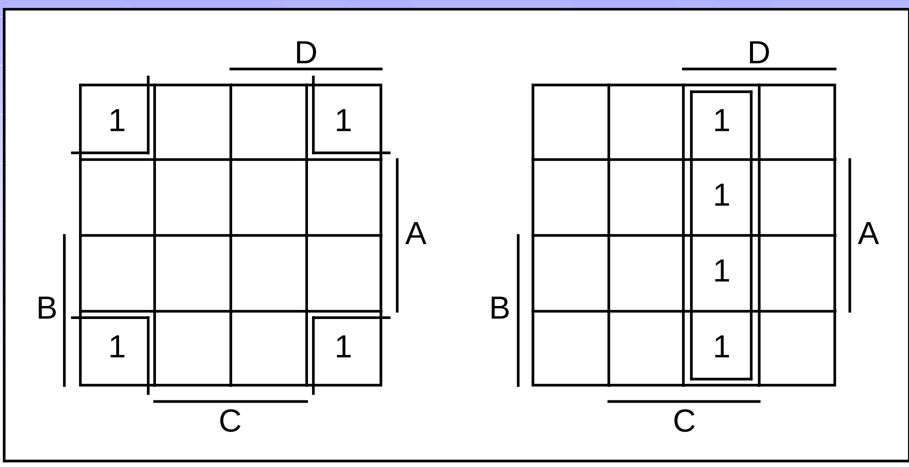
Représentation des variables

La représentation d’une <u>variable</u> correspond à un <u>bloc de 8 cases</u>
On a ainsi représenté A et A’ dans les tables ci-dessous



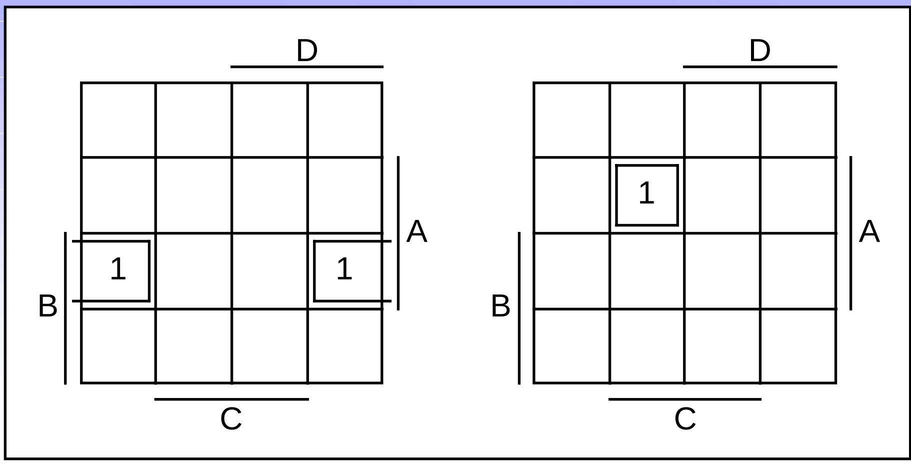
Représentation des monômes

Les <u>produits de deux variables</u> correspondent à des <u>blocs de quatre cases</u> ainsi que C’A’ et DC ci-dessous



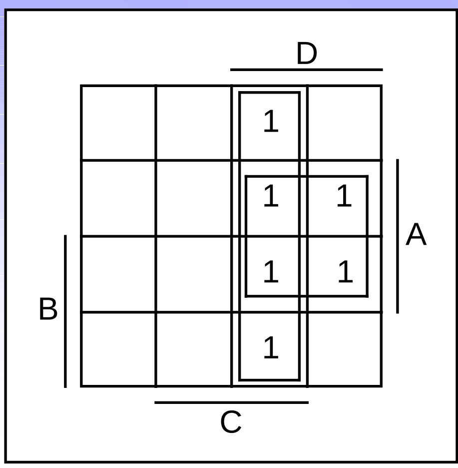
Représentation des monômes

Les <u>produits de 3 (resp. 4) variables</u> correspondent à des <u>blocs de 2 (resp. 1) cases</u> ainsi que C'BA et D'CB'A ci-dessous



Représentation des polynômes

Tout <u>polynôme</u>, tel que Z = DA + DC, est représenté par la <u>réunion des blocs</u> qui décrivent ses monômes

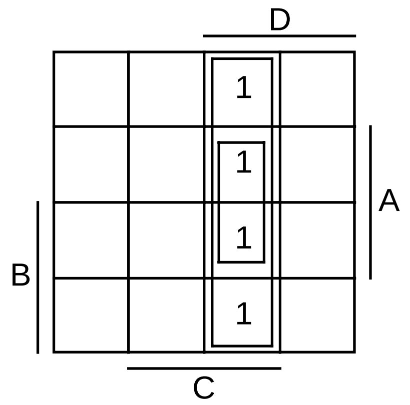


Impliquant premier

On appelle <u>impliquant premier</u> d'un polynôme tout monôme qui <u>n'est pas totalement inclus</u> dans un monôme plus grand

Z = DC + DCA
Z = DC (1 + A)
Z = DC . 1
Z = DC

Le monôme DCA qui est totalement inclus dans le monôme DC peut être supprimé



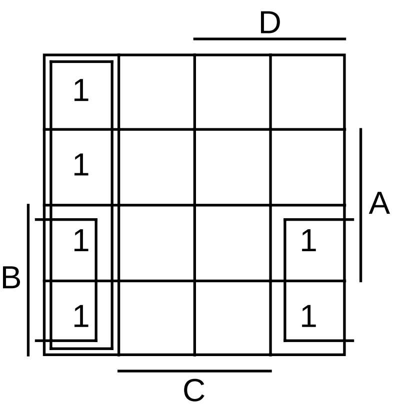
Exemple de simplification

On cherche à simplifier la fonction Z(D,C,B,A)=Σ0,1,2,3,10,11
On trouve:

0 bloc de 8 cases
2 blocs de 4 cases
7 blocs de 2 cases
6 blocs de 1 case

Seuls les blocs de 2 cases sont des impliquants premiers et ils conduisent au polynôme:

z = D'C' + C'B



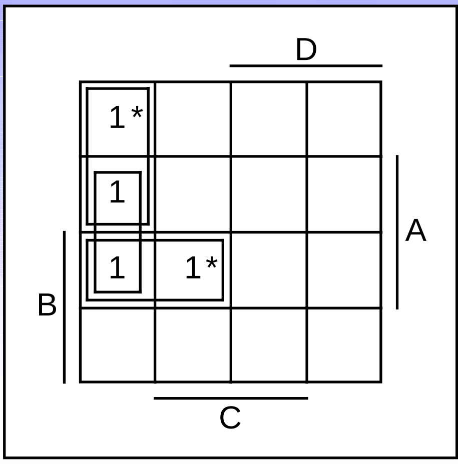
Exemple de simplification

On cherche à simplifier la fonction Z(D,C,B,A)=Σ0,1,3,7
On trouve 3 blocs de 2 cases:

D'C'B', D'C'A, D'BA

Ce sont tous des impliquants premiers mais D'C'A qui est contenu dans la réunion des deux autres peut être supprimé

z = D'C'B' + D'BA



Impliquant premier essentiel

On appelle <u>impliquant premier essentiel</u> un impliquant qui, dans la table de Karnaugh, remplit une case au moins qui n'est pas incluse dans un autre impliquant

On marque d'un astérisque (*) ce type de case

Les impliquants premiers essentiels sont des impliquants qui sont inclus dans toutes les solutions minimales résultant de la simplification

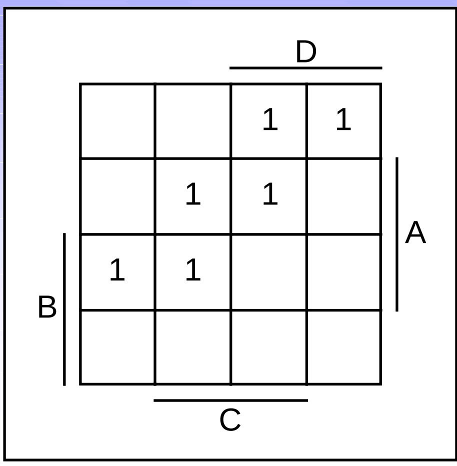
Méthode de simplification

La méthode de simplification des fonctions logiques s'effectue en quatre étapes:

1) Introduire la fonction dans la table de Karnaugh
2) Trouver tous les blocs de 1 qui correspondent à des impliquants premiers de la fonction
3) Marquer d'un astérisque (*) les impliquants premiers essentiels
4) Déterminer le polynôme minimal qui se compose de tous les impliquants premiers essentiels et d'un ensemble minimal d'impliquants premiers non essentiels destinés à couvrir les 1 de la fonction qui ne sont pas couvert par les impliquants premiers essentiels

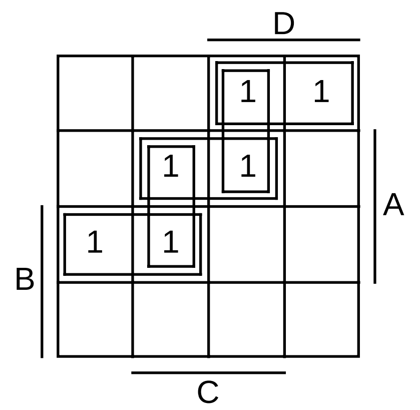
Méthode de simplification

1) Introduction de la fonction Z(D,C,B,A)=Σ3,5,7,8,12,13 dans la table de Karnaugh



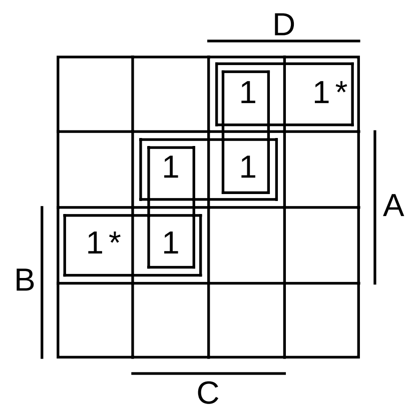
Méthode de simplification

2) 5 impliquants premiers de la fonction (blocs de deux cases): D’CA, D’BA, DCB’, DB’A’ et CB’A



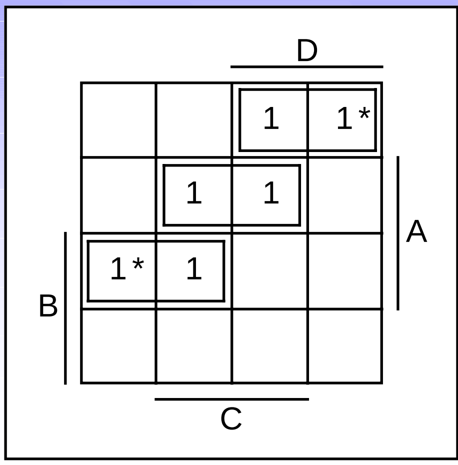
Méthode de simplification

3) 2 impliquants premiers essentiels (*): D’BA et DB’A’



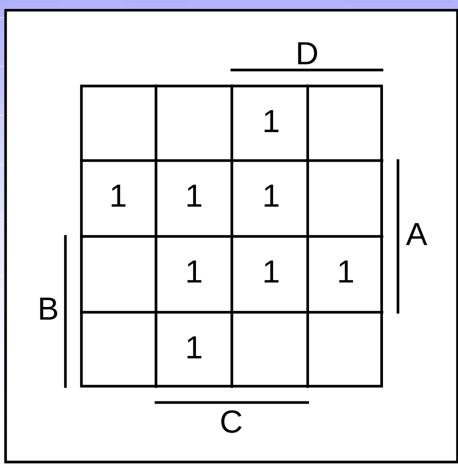
Méthode de simplification

4) 1 solution minimale: Z = D'BA + DB'A' + CB'A



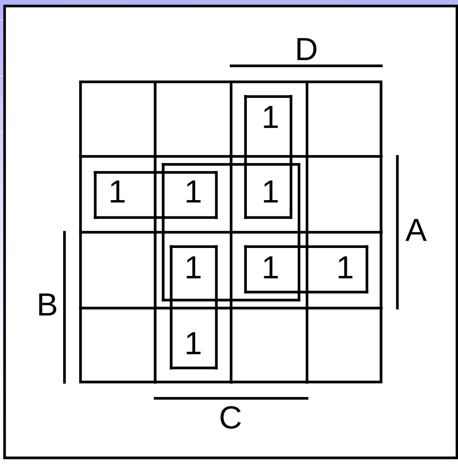
Méthode de simplification

1) Introduction de la fonction Z(D,C,B,A)=Σ1,5,6,7,11,12,13,15 dans la table de Karnaugh



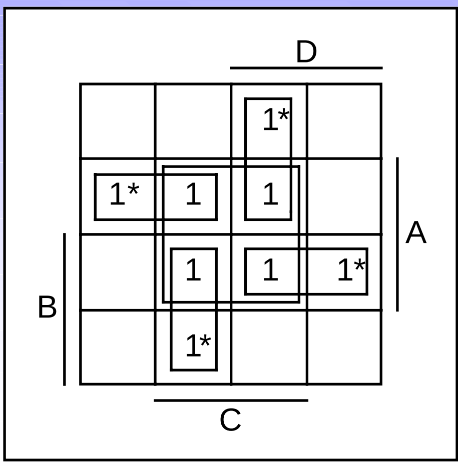
Méthode de simplification

2) 5 impliquants premiers de la fonction (1 bloc de 4 cases et 4 blocs de 2 cases): CA, D’CB, DCB’, D’B’A et DBA



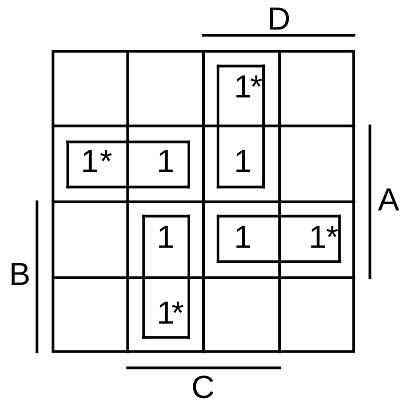
Méthode de simplification

3) 4 impliquants premiers essentiels (*): D’CB, DCB’,D’B’A et DBA



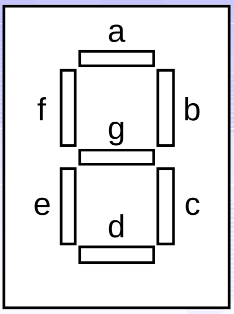
Méthode de simplification

4) 1 solution minimale: Z = D’CB + DCB’ + D’B’A + DBA



Application

On dispose de l’affichage à 7 segments représenté ci-dessous
Il s’agit de calculer les fonctions des segments a, b, … , g de manière à représenter le nombre binaire D,C,B,A sous forme hexadécimale



Application

On commence par établir une table de Karnaugh générale dans laquelle on représente les états de l'affichage

D

0	4	C	8
1	5	d	9
3	7	F	b
2	6	E	A

A

B

C

Application

Simplification du segment supérieur a (allumé=1):
1) Introduction de la fonction logique dans la table

D

1		1	1
	1		1
1	1	1	
1	1	1	1

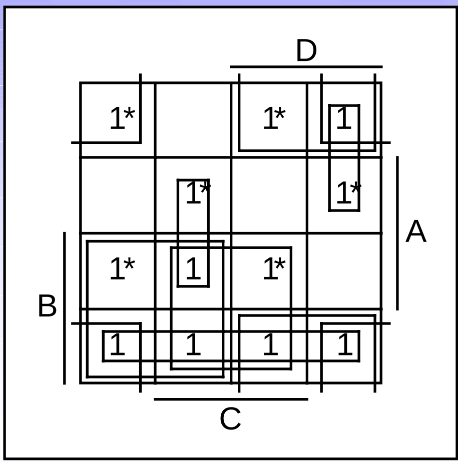
A

B

C

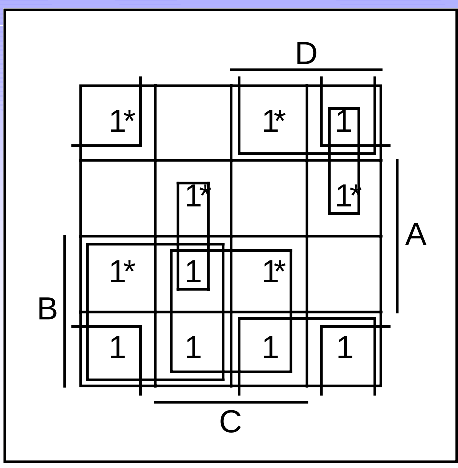
Application

2) 7 impliquants premiers
3) 6 impliquants premiers essentiels (*)



Application

4) 1 solution minimale:
a = D’B + DA’ + C’A’ + CB + D’CA + DC’B’



Application

Simplification du segment supérieur droite b:
1) Introduction de la fonction logique dans la table

			D	D
	1	1		1
A	1		1	1
A, B	1	1		
B	1			1
		C	C	

Application

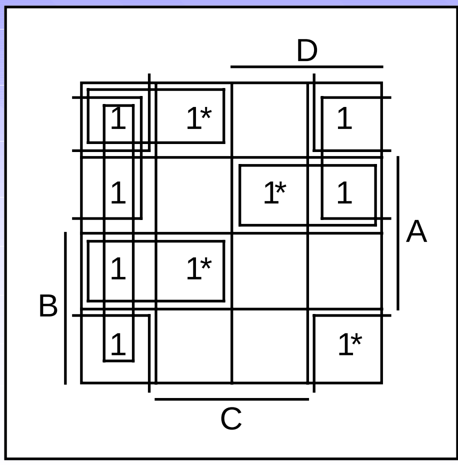
2) **6 impliquants premiers**
3) **4 impliquants premiers essentiels (*)**

			D	D
	1	1*		1
A	1		1*	1
A, B	1	1*		
B	1			1*
		C	C	

Application

4) 2 solutions minimales:

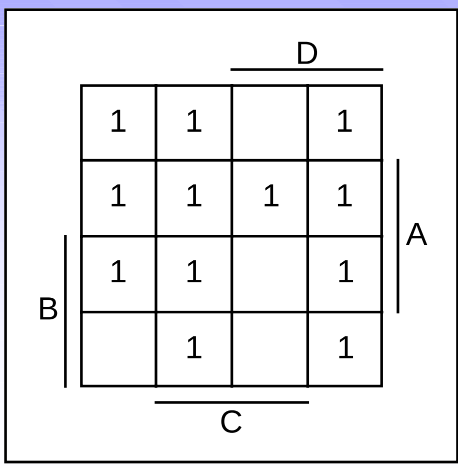
b = C’A’ + D’B’A’ + D’BA + DB’A + (D’C’ ou C’B’)



Application

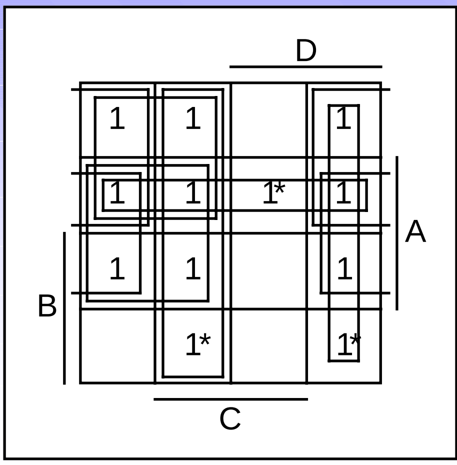
Simplification du segment inférieur droite c:

1) Introduction de la fonction logique dans la table



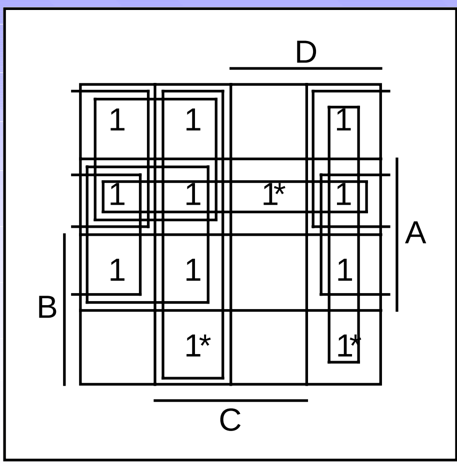
Application

2) 7 impliquants premiers
3) 3 impliquants premiers essentiels (*)



Application

4) 4 solutions minimales:
c = D’C + DC’ + B’A + (D’B’ ou C’B’) + (D’A ou C’A)



Réalisation NAND

Il s'agit de déterminer le nombre minimal de porte NAND pour implémenter les fonctions a, b et c de l'affichage:

a = D'B + DA' + <u>C'A'</u> + CB + D'CA + DC'B'
b = <u>C'A'</u> + D'B'A' + D'BA + DB'A + (D'C' ou <u>C'B'</u>)
c = D'C + DC' + B'A + (D'B' ou <u>C'B'</u>) + (D'A ou C'A)

On a souligné les termes qui se retrouvent dans les expressions
Il convient d'y recourir car on ne les implémente qu'une fois

Bilan matériel:

- **9 portes NAND à 2 entrées**
- **5 portes NAND à 3 entrées**
- **2 portes NAND à 5 entrées**
- **1 porte NAND à 6 entrées**

Réalisation circuits intégrés

Sachant qu'on dispose des circuits intégrés suivants:

- **circuit intégré 7430 (1 porte NAND à 8 entrées)**
- **circuit intégré 7420 (2 portes NAND à 4 entrées)**
- **circuit intégré 7400 (4 portes NAND à 2 entrées)**

Pour implémenter:

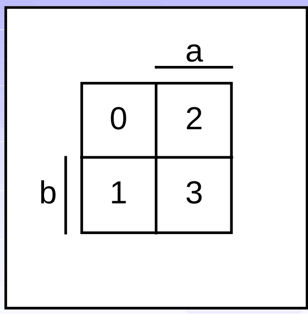
- **9 portes NAND à 2 entrées**
- **5 portes NAND à 3 entrées**
- **2 portes NAND à 5 entrées**
- **1 porte NAND à 6 entrées**

On aura recours à:

- **3 circuits 7430 (1 NAND à 6, 2 NAND à 5)**
- **3 circuits 7420 (5 NAND à 3, 1 NAND à 2)**
- **2 circuits 7400 (8 NAND à 2)**

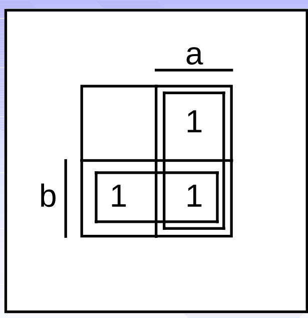
Fonctions de deux variables

La table de Karnaugh d'une fonction de deux variables z(a,b) comporte quatre cases numérotées de 0 à 3



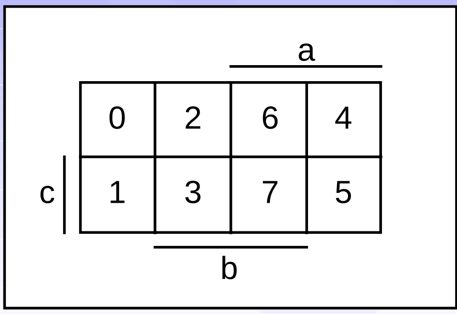
Fonctions de deux variables

La simplification de la fonction OU z(a,b)=Σ1,2,3 fait apparaître deux impliquants premiers essentiels a et b
On retrouve ainsi la relation z=a+b



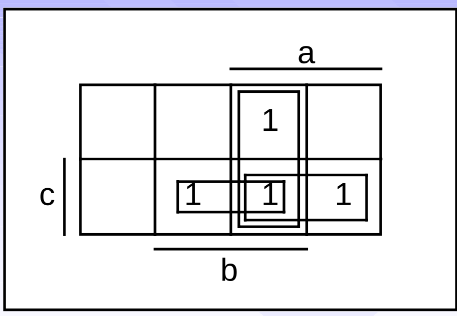
Fonctions de trois variables

La table de Karnaugh d’une fonction de trois variables z(a,b,c) comporte huit cases numérotées de 0 à 7



Fonctions de trois variables

La simplification de la fonction MAJ(a,b,c)=Σ3,5,6,7 fait apparaître trois impliquants premiers essentiels ab, ac et bc
On retrouve ainsi la relation MAJ(a,b,c)=ab+ac+bc



Fonctions de cinq variables

La table de Karnaugh d'une fonction de cinq variables Z(E,D,C,B,A) comporte 32 cases numérotées de 0 à 31

E=0 (D sur les deux colonnes de droite, C sur les deux colonnes du milieu, B sur les deux lignes du bas, A sur les deux lignes du milieu)

0	4	12	8
1	5	13	9
3	7	15	11
2	6	14	10

E

16	20	28	24
17	21	29	25
19	23	31	27
18	22	30	26

On trouve alors des blocs qui appartiennent à:

- **la table E=0 uniquement tel que E'BA'**
- **la table E=1 uniquement tel que EC'B**
- **la table E=0 et la table E=1 simultanément tel que D'C'**

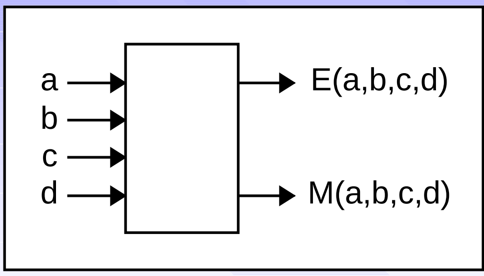
1			
1			
1			
1	1	1	1

E

1			
1			
1			1
1			1

Fonctions égalité et majorité

Considérons les fonctions égalité E et majorité M de quatre variables a, b, c et d



Condition indifférente

En cas d'égalité E=1, la fonction majorité peut prendre indifféremment la valeur 0 ou 1
Il s'agit d'une <u>condition indifférente</u> que nous symboliserons par la lettre Φ
On est ainsi conduit à la notion de <u>fonction incomplètement définie</u> (on dit aussi <u>incomplètement spécifiée</u>)

	E	M
Majorité de 0	0	0
Majorité de 1	0	1
Egalité	1	Φ

No	a	b	c	d	E(a,b,c,d)	M(a,b,c,d)
0	0	0	0	0	0	0
1	0	0	0	1	0	0
2	0	0	1	0	0	0
3	0	0	1	1	1	Φ
4	0	1	0	0	0	0
5	0	1	0	1	1	Φ
6	0	1	1	0	1	Φ
7	0	1	1	1	0	1
8	1	0	0	0	0	0
9	1	0	0	1	1	Φ
10	1	0	1	0	1	Φ
11	1	0	1	1	0	1
12	1	1	0	0	1	Φ
13	1	1	0	1	0	1
14	1	1	1	0	0	1
15	1	1	1	1	0	1

Condition indifférente

Les formes canoniques décimales des fonctions égalité E et majorité M des quatre variables a, b, c et d s'écrivent:

$$E(a,b,c,d) = \Sigma\ 3,5,6,9,10,12$$

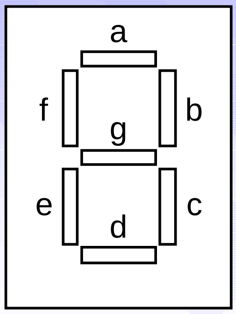
$$M(a,b,c,d) = \Sigma\ 7,11,13,14,15 + \Phi\ 3,5,6,9,10,12$$

Comme chacune des six conditions indifférentes de M peut prendre la valeur 0 ou 1, il y a 2^6=64 solutions possibles pour cette fonction

En particulier, la fonction majorité de trois variables MAJ(a,b,c) est une solution pour la fonction majorité de quatre variables M(a,b,c,d)

Application

On dispose de l'affichage à 7 segments représenté ci-dessous Il s'agit de calculer les fonctions des segments a, b, … , g de manière à représenter sous forme décimale les nombres binaires D,C,B,A compris entre 0000 et 1001 (0 et 9)



Application

La table de Karnaugh générale comporte des conditions indifférentes pour tous les états de D,C,B,A compris entre 1010 et 1111 (10 et 15)

			D	D	
	0	4	Φ	8	
	1	5	Φ	9	A
B	3	7	Φ	Φ	A
B	2	6	Φ	Φ	
		C	C		

Application

Le segment inférieur gauche e(D,C,B,A)=Σ0,2,6,8+Φ10,…,15 correspond ainsi à la table de Karnaugh individuelle ci-dessous

			D	D	
	1		Φ	1	
			Φ		A
B			Φ	Φ	A
B	1	1	Φ	Φ	
		C	C		

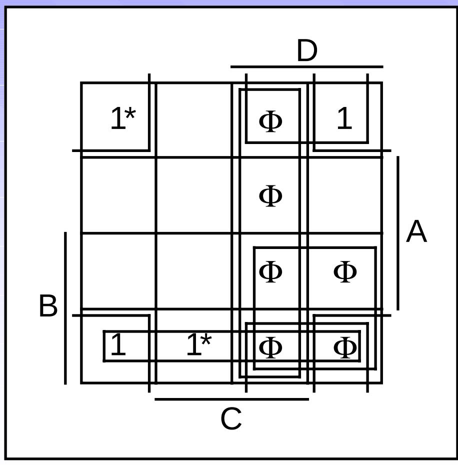
Application

Pour trouver les plus grands blocs possibles, on recherche d'abord les impliquants premiers de la fonction en considérant Φ=1

			D	D	
	1		Φ	1	
			Φ		A
B			Φ	Φ	A
B	1	1	Φ	Φ	
		C	C		

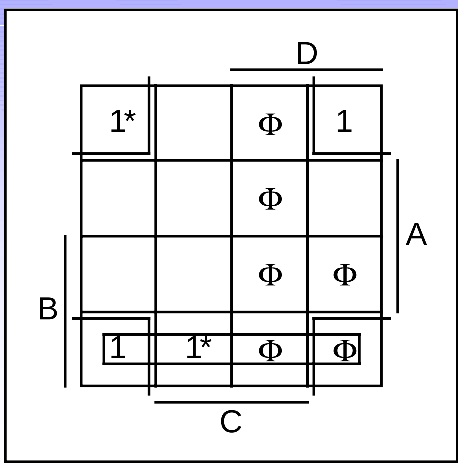
Application

On détermine ensuite parmi ces impliquants ceux qui sont essentiels (*) en considérant cette fois Φ=0



Application

Les deux impliquants premiers essentiels (*) suffisent à couvrir tous les 1 de la fonction: e = C’A’ + BA’



Méthode de simplification

La méthode de simplification des fonctions incomplètement définies s’effectue en quatre étapes:

1) **Introduire la fonction dans la table de Karnaugh**
2) **Trouver tous les blocs de 1 qui correspondent à des impliquants premiers de la fonction (Φ=1)**
3) **Marquer d’un astérisque (*) les impliquants premiers essentiels (Φ=0)**
4) **Déterminer le polynôme minimal qui se compose de tous les impliquants premiers essentiels et d’un ensemble minimal d’impliquants premiers non essentiels destinés à couvrir les 1 de la fonction qui ne sont pas couvert par les impliquants premiers essentiels**

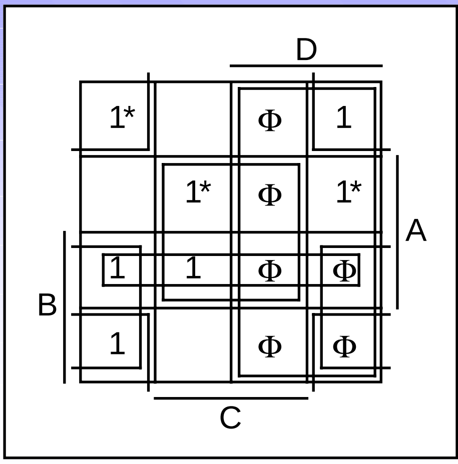
Application

Simplification du segment supérieur a:
1) Introduction de la fonction logique dans la table

			D	D	
	1		Φ	1	
		1	Φ	1	A
B	1	1	Φ	Φ	A
B	1		Φ	Φ	
		C	C		

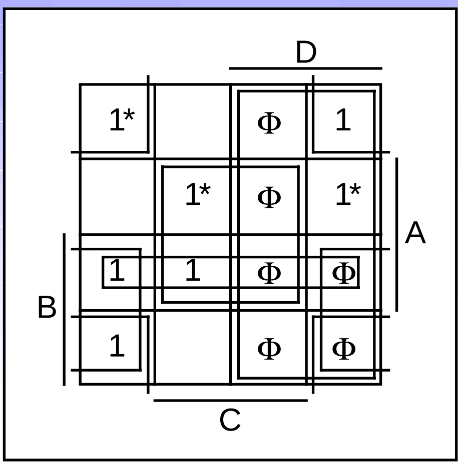
Application

2) 5 impliquants premiers (Φ=1)
3) 3 impliquants premiers essentiels (*, Φ=0)



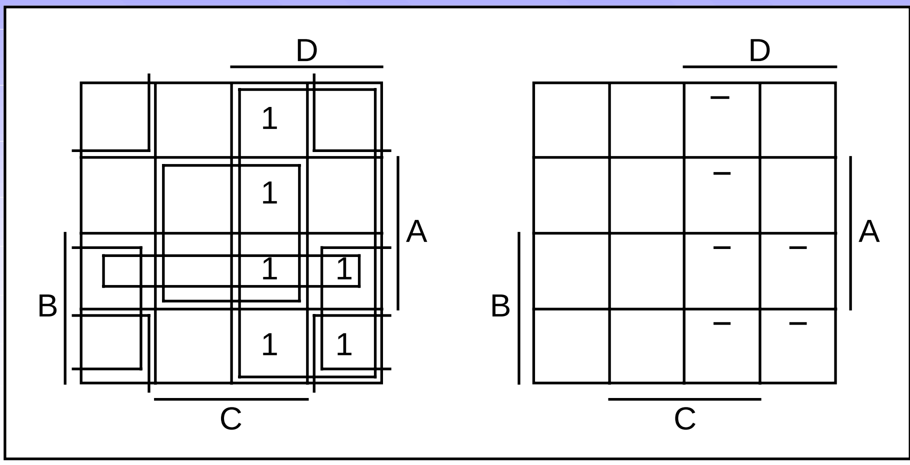
Application

4) 2 solutions minimales:
a = D + C’A’ + CA + (C’B ou BA)



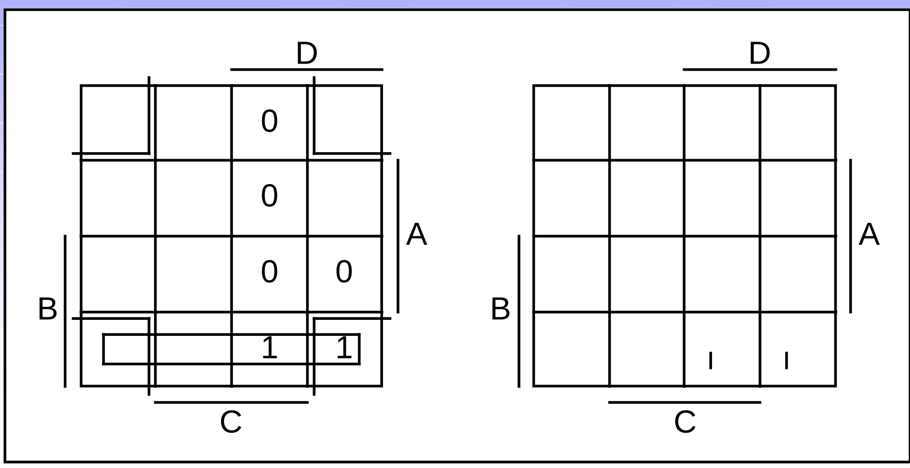
Application

L'analyse des six états non définis du segment a montre que pour les deux solutions minimales ils correspondent tous à des 1



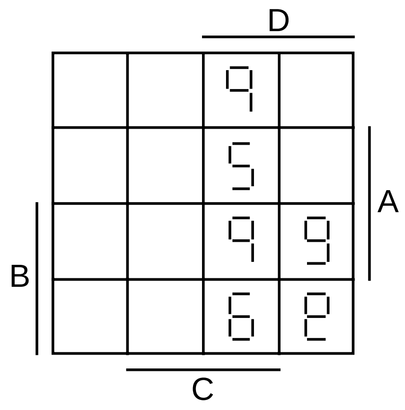
Application

L'analyse des six états non définis du segment e montre que seul deux d'entre eux correspondent à des 1



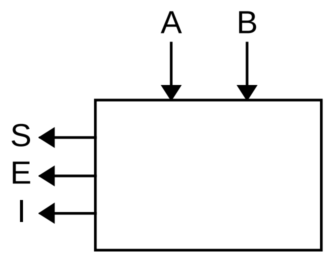
Application

Pour l’ensemble des segments, l’analyse des six états non définis conduit à l’affichage représenté ci-dessous



Comparateur

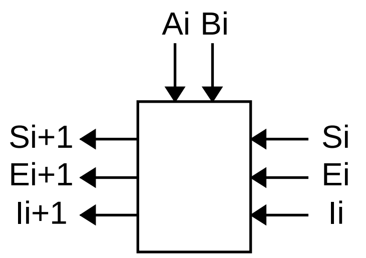
On se propose de calculer les fonctions supériorité S, égalité E et infériorité I résultant de la comparaison de deux nombres binaires A et B de n bits chacun



A > B : (S,E,I) = 100
A = B : (S,E,I) = 010
A < B : (S,E,I) = 001

Comparateur

Pour le faire, on va concevoir un module qui compare les bits de rang i des deux nombres A et B
Ce module fournit les comparaisons S, E et I pour le rang i+1



(Si+1,Ei+1,Ii+1) = 100
(Si+1,Ei+1,Ii+1) = 010
(Si+1,Ei+1,Ii+1) = 001

Si	Ei	Ii	Ai	Bi	Si+1	Ei+1	Ii+1
0	0	0	Φ	Φ		-	
0	0	1	0	0	0	0	1
			0	1	0	0	1
			1	0	1	0	0
			1	1	0	0	1
0	1	0	0	0	0	1	0
			0	1	0	0	1
			1	0	1	0	0
			1	1	0	1	0
0	1	1	Φ	Φ		-	
1	0	0	0	0	1	0	0
			0	1	0	0	1
			1	0	1	0	0
			1	1	1	0	0
1	0	1	Φ	Φ		-	
1	1	0	Φ	Φ		-	
1	1	1	Φ	Φ		-	

Comparateur

La simplification de la fonction Si+1 conduit à la relation:

Si+1 = SiAi + SiBi’ + AiBi’

Ei (left map); Si, Ei, Bi (right map); Ai; Ii

-	0	-	0
-	0	-	0
-	0	-	0
-	1	-	1

1	-	-	-
0	-	-	-
1	-	-	-
1	-	-	-

Comparateur

La simplification de la fonction Ii+1 conduit à la relation:

Ii+1 = IiAi’ + IiBi + Ai’Bi

Ei (left map); Si, Ei, Bi (right map); Ai; Ii

-	1	-	0
-	1	-	1
-	1	-	0
-	0	-	0

0	-	-	-
1	-	-	-
0	-	-	-
0	-	-	-

Comparateur

La simplification de la fonction Ei+1 conduit à la relation:

Ei+1 = EiAi’Bi’ + EiAiBi

Ei

-	0	-	1
-	0	-	0
-	0	-	1
-	0	-	0

Ai

Ii

Si

Ei

0	-	-	-
0	-	-	-
0	-	-	-
0	-	-	-

Bi

Ii

Comparateur

Les équations du module de comparaison vérifient ainsi les trois relations:

Si+1 = SiAi + SiBi’ + AiBi’

Ii+1 = IiAi’ + IiBi + Ai’Bi

Ei+1 = EiAi’Bi’ + EiAiBi

Du fait du codage 1 parmi 3 utilisé, il est cependant possible de remplacer n’importe laquelle de ces relations par une combinaison des deux autres:

Si+1 = Ii+1’Ei+1’

Ii+1 = Si+1’Ei+1’

Ei+1 = Si+1’Ii+1’

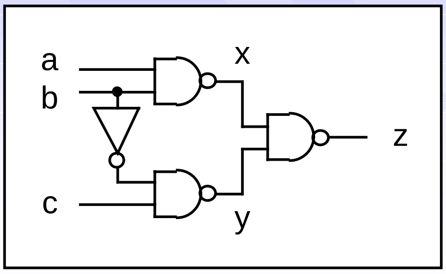
Caractéristiques temporelles

Tout circuit logique qui réalise une somme de produits peut engendrer un <u>aléa statique</u> (en anglais: static hazard) à sa sortie si son schéma comporte trois niveaux de portes

Le circuit ci-dessous qui implémente la fonction logique

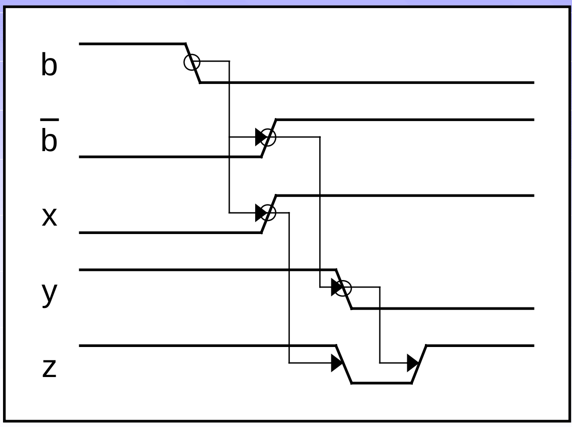
z = ab + b'c

présente un aléa statique lorsqu'on passe de l'état d'entrée a,b,c=111 à l'état a,b,c=101



Caractéristiques temporelles

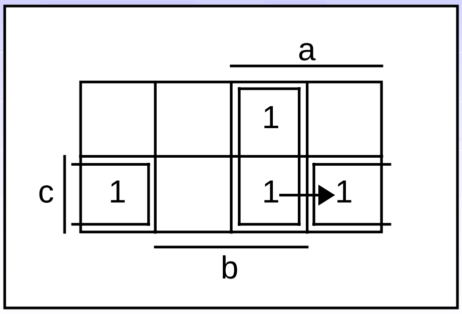
L'aléa statique engendré par le déclenchement de l'entrée b se traduit par une <u>impulsion indésirable</u> (en anglais: glitch) 1->0->1 à la sortie z



Caractéristiques temporelles

Dans la table de Karnaugh, où la fonction z correspond aux deux impliquants premiers ab et b’c, l’aléa statique produit par le déclenchement de b résulte du passage de l’un à l’autre de ces impliquants

Pour le supprimer, il convient d’ajouter l’impliquant ac, qui couvre la transition, dans l’équation de z



Caractéristiques temporelles

On appelle aléas dynamiques (en anglais: dynamic hazards) les séquences non désirables

z = 1 -> 0 -> 1 -> 0

ou

z = 0 -> 1 -> 0 -> 1

qui résultent de l’enclenchement ou du déclenchement d’une variable d’entrée

Les aléas dynamiques peuvent survenir dans les circuits dont le nombre de niveaux de portes logiques est supérieur à trois