Simplification OU-exclusive

- Opérateur OU-exclusif
- Circuit itératif

andre.stauffer@epfl.ch

Opérateur OU-exclusif

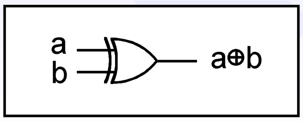
La transformation d'une forme canonique en une expression comportant des sommes OU-exclusives est motivée par les considérations suivantes:

- L'opérateur OU-exclusif est d'un emploi fréquent
- Les fonctions dont la simplification conduit à un polynôme compliqué peut généralement être réalisée par une expression algébrique simple comportant des sommes OU-exclusives
- La table de Karnaugh peut être utilisée pour déterminer des expressions comportant des sommes OU-exclusives

Opérateur OU-exclusif

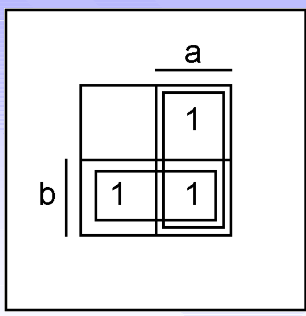
a	b	z6
0	0	0
0	1	1
1	0	1
1	1	0

La fonction XOR (en français: OU-exclusif) est définie algébriquement par la relation z6=a⊕b
Le symbole de la porte XOR s'apparente à celui de la porte OU



Opérateur OU-exclusif

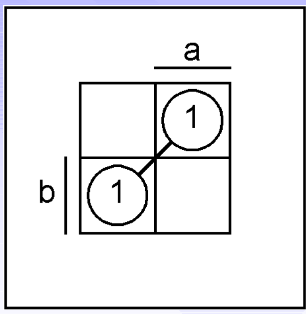
Représentation de la fonction OU dans la table de Karnaugh
z = a + b



Opérateur OU-exclusif

Représentation de la fonction XOR dans la table de Karnaugh

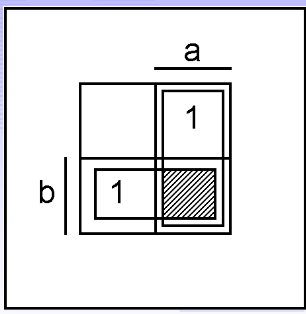
$z = a \oplus b = a'.b + a.b'$



Opérateur OU-exclusif

Représentation de la fonction XOR dans la table de Karnaugh

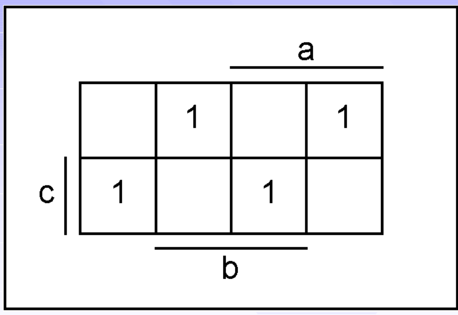
$z = a \oplus b = a'.b + a.b'$



Opérateur OU-exclusif

Exemple d’une fonction de trois variables

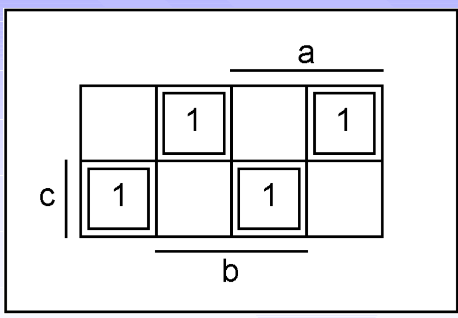
$z(a, b, c) = \Sigma$ 1, 2, 4, 7



Opérateur OU-exclusif

Simplification tabulaire classique

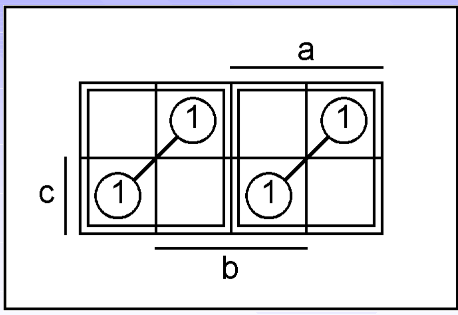
$z = a'.b'.c + a'.b.c' + a.b'.c' + a.b.c$



Opérateur OU-exclusif

Simplification en faisant usage de la <u>représentation en damier</u>

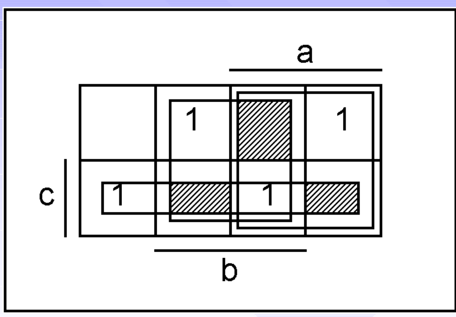
$z = a'.(b \oplus c) + a.(b \oplus c)' = a \oplus (b \oplus c) = a \oplus b \oplus c$



Opérateur OU-exclusif

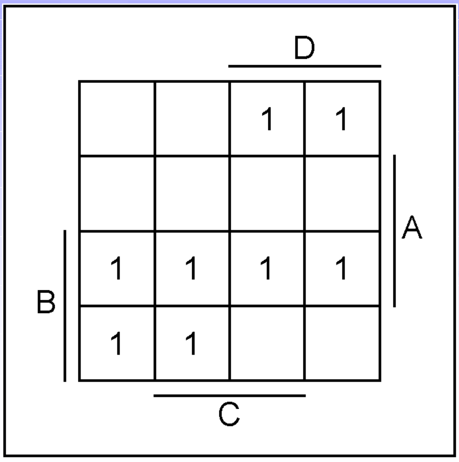
Simplification en faisant usage de la <u>soustraction logique</u>

$z = a \oplus b \oplus c$



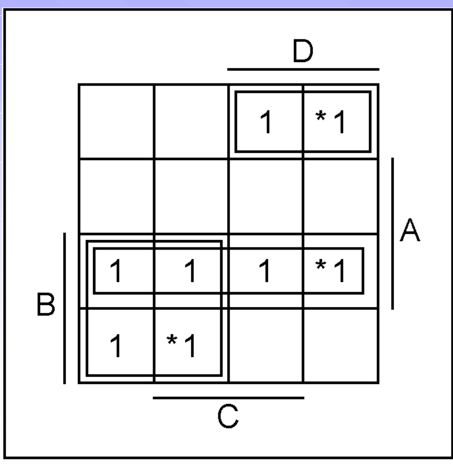
Opérateur OU-exclusif

Exemple d'une fonction de quatre variables
Z(D, C, B, A) = Σ 2, 3, 6, 7, 8, 11, 12, 15



Opérateur OU-exclusif

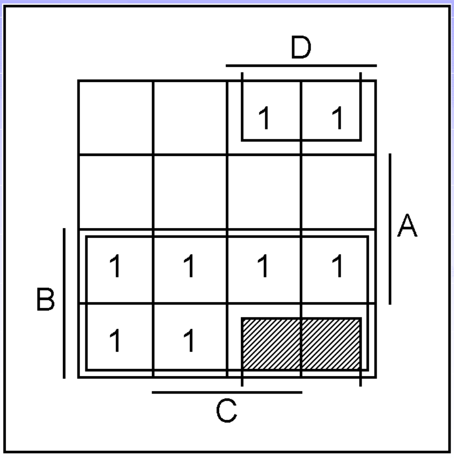
Simplification tabulaire classique
Z = D'.B + B.A + D.B'.A' (7 lettres)



Opérateur OU-exclusif

Simplification par soustraction logique

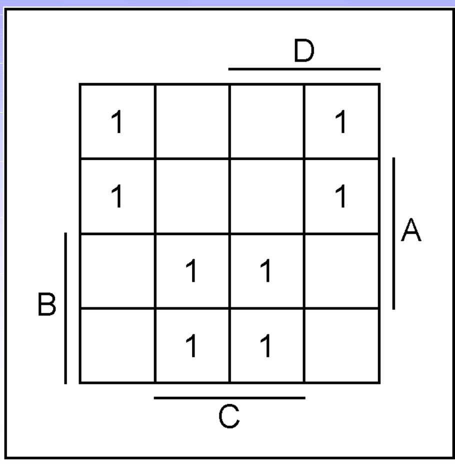
Z = B ⊕ D.A’ (3 lettres)



Opérateur OU-exclusif

Exemple d’une fonction de quatre variables

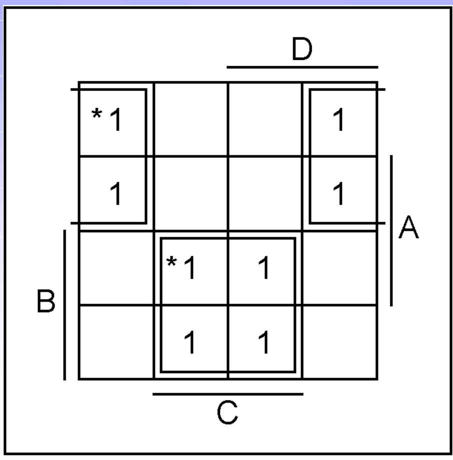
Z(D, C, B, A) = Σ 0, 1, 6, 7, 8, 9, 14, 15



Opérateur OU-exclusif

Simplification tabulaire classique

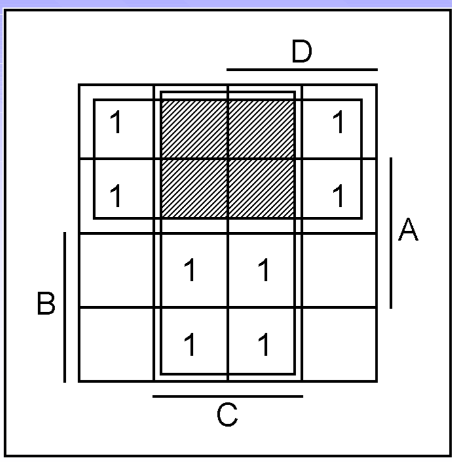
Z = C’.B’ + C.B (4 lettres)



Opérateur OU-exclusif

Simplification par soustraction logique

Z = C ⊕ B’ (2 lettres)



Opérateur OU-exclusif

Exemple d’une fonction de quatre variables
Z(D, C, B, A) = Σ 3, 5, 6, 9, 10, 12

D

		1	
	1		1
1			
	1		1

A

B

C

Opérateur OU-exclusif

Décomposition en deux fonctions de quatre variables
Z(D, C, B, A) = Z1(D, C, B, A) + Z2(D, C, B, A)

D

		1	
			1
1			
	1		

A

B

C

D

		1	
	1		
1			
			1

A

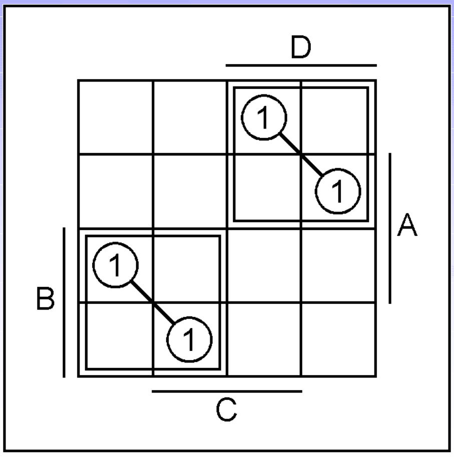
B

C

Opérateur OU-exclusif

Simplification par représentation en damier de Z1

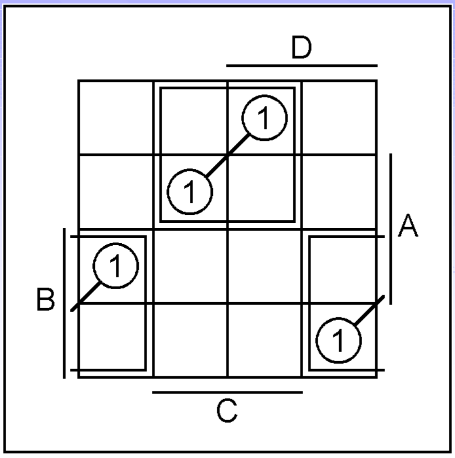
Z1 = D'B(C⊕A) + DB'(C⊕A) = (C⊕A)(D'B + DB') = (C⊕A)(D⊕B)



Opérateur OU-exclusif

Simplification par représentation en damier de Z2

Z2 = C'B(D⊕A) + CB'(D⊕A) = (D⊕A)(C'B + CB') = (D⊕A)(C⊕B)



Opérateur OU-exclusif

Simplifications OU-exclusives de Z1 et Z2:

$$Z1 = (C \oplus A)(D \oplus B)$$

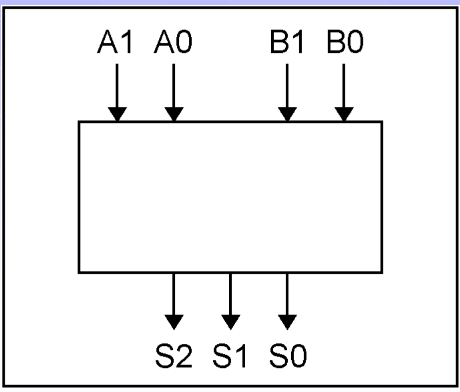
$$Z2 = (D \oplus A)(C \oplus B)$$

Simplification OU-exclusive de Z:

$$Z = Z1 + Z2 = (C \oplus A)(D \oplus B) + (D \oplus A)(C \oplus B) \quad \text{(8 lettres)}$$

Additionneur

On se propose d'effectuer la simplification OU-exclusive d'un additionneur calculant la somme S2:0 de deux nombre de deux bits A1:0 et B1:0



Additionneur

La représentation des nombres décimaux A et B à l'aide des bits A1, A0, B1 et B0 est représentée ci-dessous

A1 A0 B1 B0	A B
0 0	0
0 1	1
1 0	2
1 1	3

Additionneur

La table de Karnaugh principale donne les valeurs décimales de la somme S2:0 en fonction des bits A1, A0, B1 et B0

			A1	A1	
	0	1	3	2	
	1	2	4	3	B0
B1	3	4	6	5	B0
B1	2	3	5	4	
		A0	A0		

Additionneur

La table de Karnaugh générale donne les valeurs binaires de la somme S2:0 en fonction des bits A1, A0, B1 et B0

			A1	A1	
	000	001	011	010	
	001	010	100	011	B0
B1	011	100	110	101	B0
B1	010	011	101	100	
		A0	A0		

Additionneur

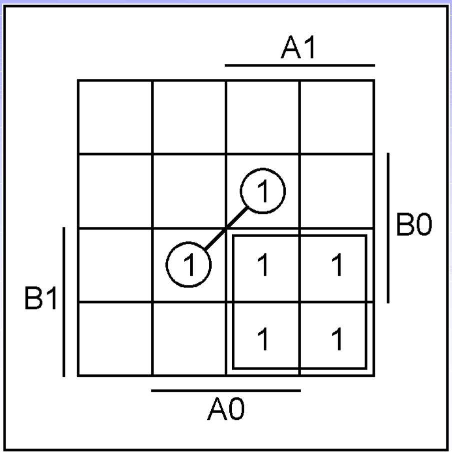
La table de Karnaugh individuelle donne les valeurs du bit S2 de la somme S2:0 en fonction des bits A1, A0, B1 et B0

			A1	A1	
			1		B0
B1		1	1	1	B0
B1			1	1	
		A0	A0		

Additionneur

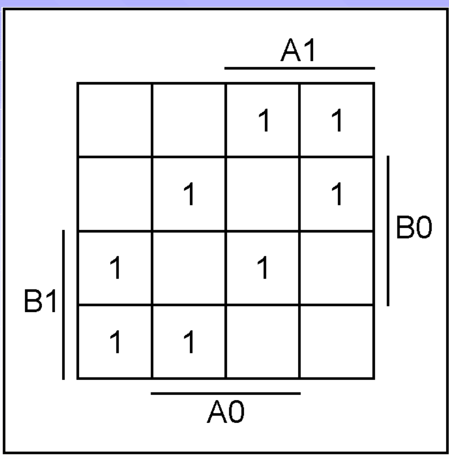
La simplification OU-exclusive de S2 conduit à la relation:

$S2 = A1B1 + A0B0.(A1 \oplus B1) = A1B1 \oplus A0B0.(A1 \oplus B1)$



Additionneur

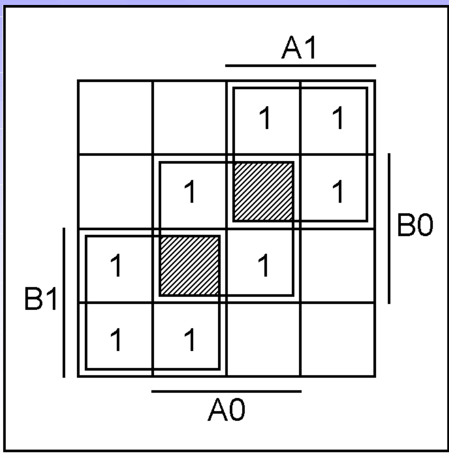
La table de Karnaugh individuelle représente les valeurs du bit S1 de la somme S2:0 en fonction des bits A1, A0, B1 et B0



Additionneur

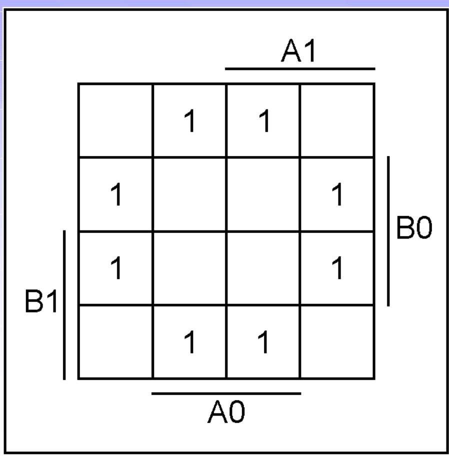
La simplification OU-exclusive de S1 conduit à la relation:

$S1 = A0B0 \oplus (A1'B1 + A1B1') = A0B0 \oplus A1 \oplus B1$



Additionneur

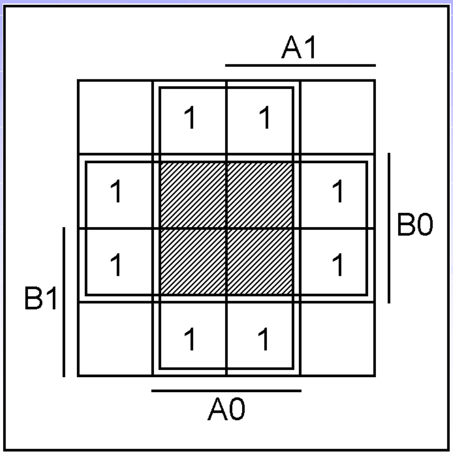
La table de Karnaugh individuelle représente les valeurs du bit S0 de la somme S2:0 en fonction des bits A1, A0, B1 et B0



Additionneur

La simplification OU-exclusive de S0 conduit à la relation:

$$S0 = A0 \oplus B0$$



Additionneur

L'addition de deux nombres de n bits An-1:0 et Bn-1:0 conduit à une somme de n+1 bits Sn:0

	An-1	...	A1	A0
+	Bn-1	...	B1	B0
Sn	Sn-1	...	S1	S0

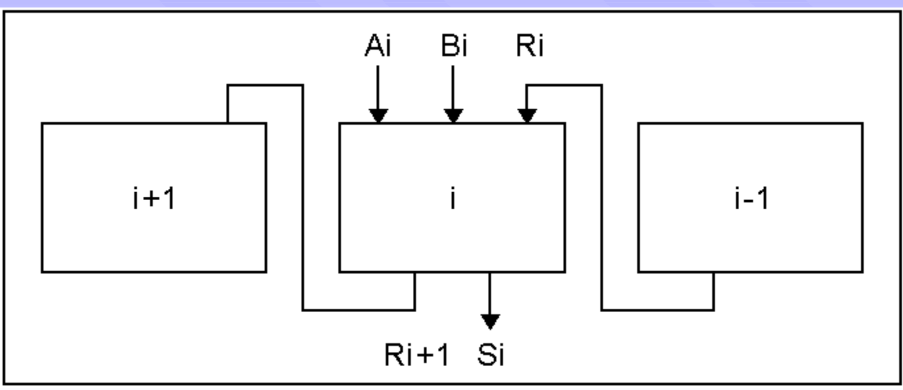
Additionneur

L'addition peut se faire comme dans le système décimal colonne par colonne en reportant la retenue R

R:	1	1	0	0	
A:		1	1	0	0
B:		0	1	1	0
S:	1	0	0	1	0

Circuit itératif

Cette addition s'effectue dans un circuit itératif dont le module de rang i calcule, à partir des bits Ai et Bi des deux nombres et du report Ri, le bit de somme Si et le report Ri+1



Circuit itératif

La table de Karnaugh principale donne les valeurs décimales de l’addition des trois bits Ai, Bi et Ri

		Ai	
0	1	2	1
1	2	3	2

Ri (second row) — Bi (columns 2–3)

Circuit itératif

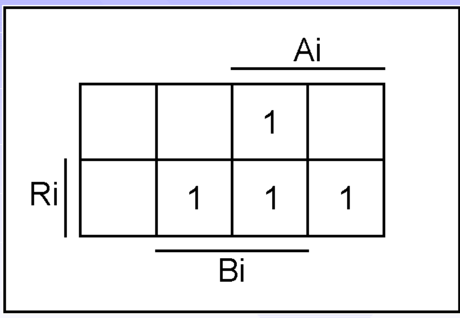
La table de Karnaugh générale donne les valeurs binaires du nombre de deux bits Ri+1Si résultant de l’addition

		Ai	
00	01	10	01
01	10	11	10

Ri (second row) — Bi (columns 2–3)

Circuit itératif

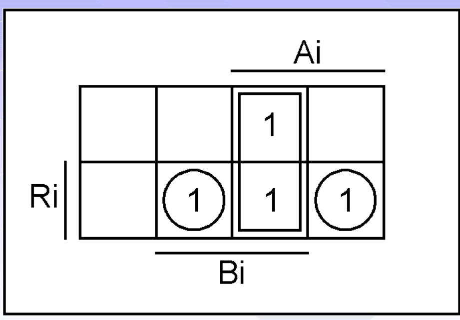
La table de Karnaugh individuelle donne les valeurs binaires du bit de report Ri+1 résultant de l'addition



Circuit itératif

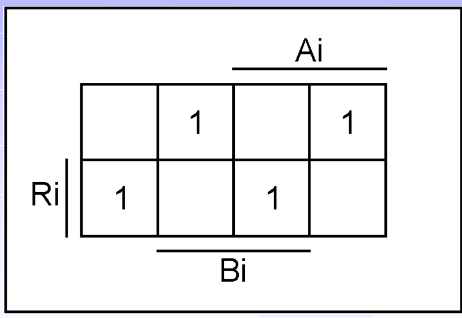
La simplification OU-exclusive de Ri+1 conduit à la relation:

$$R_{i+1} = A_iB_i + (A_i'B_i + A_iB_i').R_i = A_iB_i \oplus (A1 \oplus B1).R_i$$



Circuit itératif

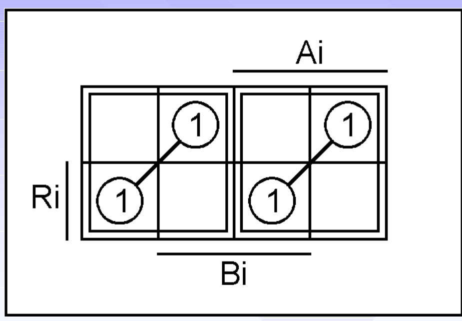
La table de Karnaugh individuelle donne les valeurs binaires du bit de somme Si résultant de l'addition



Circuit itératif

La simplification OU-exclusive de Si conduit à la relation:

$Si = Ai'.(Bi \oplus Ri) + Ai.(Bi \oplus Ri)' = Ai \oplus Bi \oplus Ri$



Réalisation circuits intégrés

Il s'agit de déterminer le nombre minimal de portes ET à Deux entrées et de portes OU-exclusives à deux entrées pour Implémenter le module de base de l'additionneur itératif:

$Ri+1 = AiBi \oplus (\underline{A1 \oplus B1}).Ri$

$Si = \underline{Ai \oplus Bi} \oplus Ri$

Bilan matériel:

- **2 portes AND à 2 entrées**
- **3 portes XOR à 2 entrées**

Réalisation circuits intégrés

Sachant qu'on dispose des circuits intégrés suivants:

- **circuit intégré 7408 (4 porte AND à 2 entrées)**
- **circuit intégré 7486 (4 portes XOR à 2 entrées)**

Pour implémenter:

- **2 portes AND à 2 entrées**
- **3 portes XOR à 2 entrées**

On aura recours à:

- **0.5 circuit 7408**
- **0.75 circuit 7486**