Architecture des systèmes numériques et informatiques
TD 3 : Tableaux de Karnaugh

Halim Djerroud

5 novembre 2025

Exercice 1 : Fonction à 2 variables

Soit la fonction booléenne $F(A,B)$ définie par sa table de vérité :

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

1. Tracer le tableau de Karnaugh.
2. Identifier les groupements possibles.
3. Donner l'expression simplifiée de F.
4. Dessiner le logigramme correspondant.

Solution : K-map :

	$B=0$	$B=1$
$A=0$	1	0
$A=1$	1	1

Groupements :
— Groupe vertical (2 cases) : $A \cdot \overline{B}$ et $A \cdot B \rightarrow A$
— Reste : $\overline{A} \cdot \overline{B}$
Expression simplifiée : $F = A + \overline{A} \cdot \overline{B} = A + \overline{B}$
Alternative : $F = \overline{B} + A$

Exercice 2 : Fonction à 3 variables

Soit $F(A,B,C) = \sum m(0,2,5,7)$ où $m(i)$ représente le minterme i.

1. Écrire la table de vérité complète.
2. Construire le tableau de Karnaugh (2×4).
3. Effectuer les regroupements optimaux.
4. Donner l'expression simplifiée en somme de produits.

Solution : Table de vérité :

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

K-map (BC en colonnes avec code Gray) :

	00	01	11	10
$A = 0$	1	0	0	1
$A = 1$	0	1	1	0

Groupements :
— Groupe horizontal haut : $\overline{A} \cdot \overline{C}$
— Groupe horizontal bas : $A \cdot C$
Expression simplifiée : $F = \overline{A} \cdot \overline{C} + A \cdot C = A \odot C$ (XNOR)

Exercice 3 : Fonction à 4 variables

Soit $F(A, B, C, D) = \sum m(0, 1, 2, 5, 8, 9, 10)$.

1. Construire le tableau de Karnaugh 4×4.
2. Identifier tous les groupements de 1, 2 ou 4 cases possibles.
3. Choisir le regroupement optimal (minimum de termes).
4. Donner l'expression booléenne simplifiée.

Solution : K-map (AB en lignes, CD en colonnes, code Gray) :

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

Groupements optimaux :
— Groupe de 4 (colonnes 00 et 10) : $\overline{B} \cdot \overline{D}$
— Groupe de 2 (colonne 01, lignes 00 et 10) : $\overline{B} \cdot C \cdot D$
— Case isolée (01, ligne 01) : incluse avec la colonne 01 si possible
Meilleure solution : $F = \overline{B} \cdot \overline{D} + \overline{A} \cdot C \cdot D$
Vérification : couvre m(0,2,8,10) + m(1,5)

Exercice 4 : Conditions indifférentes (don't care)

Soit une fonction $F(A, B, C, D)$ où :
— $F = 1$ pour les mintermes : 1, 3, 7, 11, 15
— $F = X$ (indifférent) pour : 0, 2, 5
— $F = 0$ pour les autres

1. Construire le K-map en notant X les cases indifférentes.
2. Utiliser les X pour former des groupements plus grands.
3. Donner l'expression simplifiée optimale.
4. Comparer avec la simplification sans utiliser les X.

Solution : K-map :

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

Avec les X :
— Groupe de 4 (colonne 11) : $C \cdot D$
— Groupe de 4 (ligne 00, en incluant X) : $\overline{A} \cdot \overline{B}$
Expression optimale : $F = C \cdot D + \overline{A} \cdot \overline{B}$
Sans les X, il faudrait plus de termes.

Exercice 5 : Simplification en produit de sommes

Soit $F(A,B,C) = \sum m(1,3,6,7)$.

1. Simplifier F en somme de produits (SOP).
2. Identifier les cases à 0 dans le K-map.
3. Simplifier $\overline{F}$ en SOP.
4. En déduire F en produit de sommes (POS) via $F = \overline{\overline{F}}$.
5. Comparer les deux formes.

Solution : K-map :

A\BC	00	01	11	10
0	0	1	1	0
1	0	0	1	1

SOP de F : $F = \overline{A} \cdot C + A \cdot B$
Pour POS, regrouper les 0 : $\overline{F} = \overline{C} + \overline{A} \cdot \overline{B}$
Par De Morgan : $F = C \cdot (A + B)$
POS : $F = C \cdot (A + B)$

Exercice 6 : Décodeur 7 segments

Un afficheur 7 segments affiche les chiffres 0-9. Chaque segment (a-g) est une fonction booléenne des 4 bits d'entrée $ABCD$ (codage BCD).

Pour le segment a (haut) :
- $a = 1$ pour les chiffres : 0, 2, 3, 5, 6, 7, 8, 9
- $a = 0$ pour : 1, 4
- $a = X$ pour : 10-15 (codes invalides en BCD)

1. Construire le K-map pour a.
2. Simplifier en utilisant les conditions indifférentes.
3. Donner l'expression logique du segment a.

Solution : K-map pour segment a :

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

Groupements avec X :
- Grand groupe incluant la plupart des 1 et les X utiles

Expression simplifiée : $a = A + C + \overline{B} \cdot D + B \cdot \overline{D}$
Ou encore : $a = A + C + B \oplus D$

Exercice 7 : Comparateur 2 bits

Concevoir un circuit qui compare deux nombres de 2 bits : $X = X_1X_0$ et $Y = Y_1Y_0$.
Les sorties sont :
- E (égal) : $E = 1$ si $X = Y$
- G (plus grand) : $G = 1$ si $X > Y$
- S (plus petit) : $S = 1$ si $X < Y$

1. Établir les tables de vérité pour E, G et S.
2. Simplifier chaque fonction avec un K-map.
3. Donner les expressions booléennes.
4. (Bonus) Dessiner le circuit logique complet.

Solution partielle : Pour E : $E = (X_1 \odot Y_1) \cdot (X_0 \odot Y_0) = \overline{X_1 \oplus Y_1} \cdot \overline{X_0 \oplus Y_0}$
Pour G : Analyser bit par bit $G = X_1 \cdot \overline{Y_1} + (X_1 \odot Y_1) \cdot X_0 \cdot \overline{Y_0}$
Pour S : $S = \overline{Y_1} \cdot X_1 + (X_1 \odot Y_1) \cdot \overline{Y_0} \cdot X_0$
Les K-maps permettent de vérifier ces expressions.

Exercice 8 : Additionneur 1 bit complet

Un additionneur complet 1 bit calcule $S = A \oplus B \oplus C_{in}$ et C_{out}.

1. Établir la table de vérité complète (3 entrées : A, B, C_{in}).
2. Construire les K-maps pour S (somme) et C_{out} (retenue).
3. Simplifier les deux fonctions.
4. Vérifier que $C_{out} = AB + AC_{in} + BC_{in}$ (fonction majorité).

Solution : Table de vérité :

A	B	C_{in}	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Pour C_{out} (mintermes 3,5,6,7) : K-map donne : $C_{out} = A \cdot B + A \cdot C_{in} + B \cdot C_{in}$
Pour S (mintermes 1,2,4,7) : $S = A \oplus B \oplus C_{in}$ (pas vraiment simplifiable)

Exercice 9 : Système d'alarme

Un système d'alarme dispose de 4 capteurs : A (porte), B (fenêtre), C (mouvement), D (fumée).
L'alarme se déclenche ($F = 1$) si :
— Le détecteur de fumée est activé (priorité absolue)
— OU la porte ET la fenêtre sont ouvertes simultanément
— OU un mouvement est détecté ET la porte est ouverte

1. Écrire l'expression logique non simplifiée de F.
2. Construire le K-map correspondant.
3. Simplifier l'expression.
4. Combien de portes logiques sont nécessaires ?

Solution : Expression initiale : $F = D + A \cdot B + A \cdot C$
Développement : $F = D + A \cdot (B + C)$
Le K-map confirme cette simplification.
Circuit : nécessite 1 porte OR, 1 porte AND