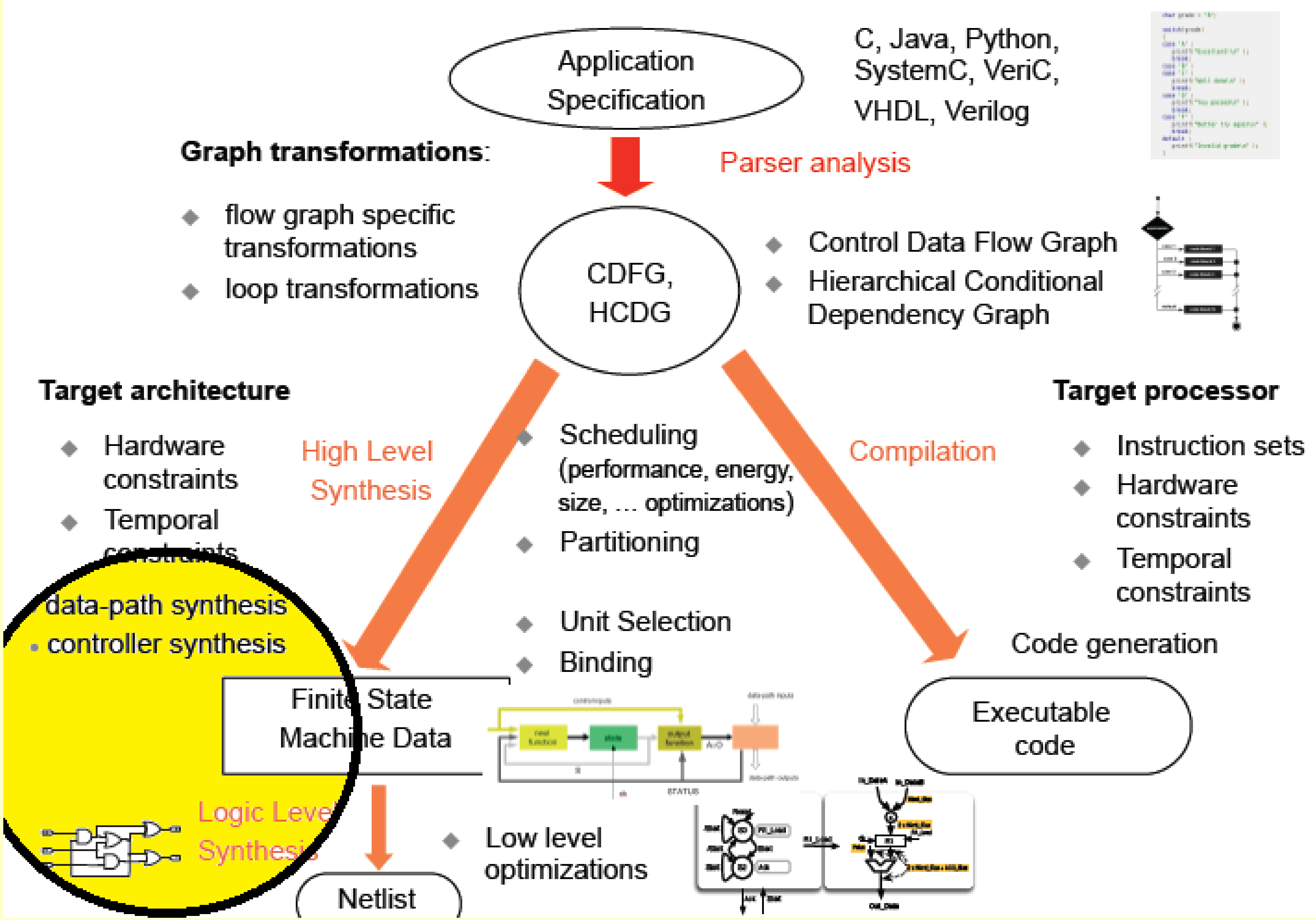
	CM	Ens	TD	Ens	TP	Ens
1	UT-UC- Langage, Graphe CDGF, C + DFG	C.W				
2	UT-UC- Langage, Graphe CDGF, C + DFG	C.W				
3	Algebre de BOOLE / FPGA	J.B	UT-UC- Langage, Graphe CDGF, C + DFG	C.W		
4	Représentation des valeurs	J.B	UT-UC- Langage, Graphe CDGF, C + DFG	J.B		
5	**CC - Modelisation du système UT-UC**	C.W	Algebre de BOOLE /représentations	J.B		
6	Circuits combinatoires / Mémoires	J.B	Algebre de BOOLE /représentations	J.B		
7	Circuits séquentiels	J.B	Circuits séquentiels	J.B	Apprentissage Max2Plus	P.H.
8	modelisation /Ut-UC/ réalisation	C.W	Circuits séquentiels	J.B	Circuits combinatoires	P.H.
			FSM relaisation / Examlpes : modelisation /Ut-UC/ réalisation	J.B	Circuits séquentiels	C.W
			FSM relaisation / Examlpes : modelisation /Ut-UC/ réalisation	J.B	UT-UC- Langage, Graphe CDGF, C + DFG	C.W
					Projet - Processeurs	C.W
					Projet - Processeurs	C.W
					Projet - Processeurs	C.W
					Soutenance Projet	C.W

Outline :



Application Specification
C, Java, Python, SystemC, VeriC, VHDL, Verilog
Graph transformations:
flow graph specific transformations
loop transformations
Parser analysis
CDFG, HCDG
Control Data Flow Graph
Hierarchical Conditional Dependency Graph
Target architecture
Hardware constraints
Temporal constraints
High Level Synthesis
Scheduling (performance, energy, size, … optimizations)
Partitioning
Unit Selection
Binding
Compilation
Target processor
Instruction sets
Hardware constraints
Temporal constraints
Code generation
data-path synthesis
controller synthesis
Finite State Machine Data
Executable code
Logic Level Synthesis
Low level optimizations
Netlist

ARC1 - Cours n°3

ALGEBRE DE BOOLE / FPGA

A. ALGEBRE DE BOOLE

1. Variables et fonctions binaires
2. Les 3 fonctions de base en algèbre de Boole
3. Autres fonctions courantes de 2 variables
4. Propriétés des opérateurs ET, OU, NON
5. Fonctions quelconques
6. Ecritures canoniques d’une fonction logique quelconque
7. Simplification de l’écriture des fonctions logiques

B. CIRCUITS LOGIQUES PROGRAMMABLES

1. Généralités sur les circuits programmables
2. Les FPGA (Field Programmable Gate Array)

A. ALGEBRE DE BOOLE

1. Variables et fonctions binaires

- Machines traitant l'information constituées de circuits ne possédant que **2 états** électriques **stables** (2 niveaux de tension possibles)

- Correspondance entre **niveaux de tension** et **niveaux logiques**

- Information (nombres, symboles, caractères..etc) représentée par une combinaison de ces deux états.

- Information représentée sous forme « **binaire** », par un ensemble de chiffres binaires (<u>bi</u>nary digi<u>ts</u> = **bits**) : **0 ou 1**

- Calcul de **fonctions de sortie binaires** à partir de variables d'entrée binaires
- Etude et conception de ces fonctions de variables binaires : **algèbre de Boole** (ou « booléenne »)

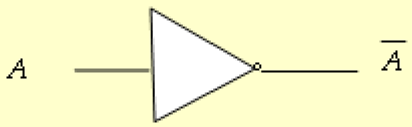
ECOLE SUPERIEURE D'INGENIEURS DE RENNES

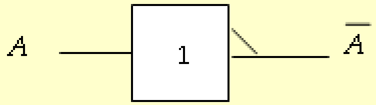
2. Les 3 fonctions de base en algèbre de Boole

2.1. La fonction NON

- Aussi appelée « Négation », « inversion » ou « complément »
- Notée : $f(x) = \bar{x}$ (x barre)
- Table de vérité (donne la valeur de la fonction, 0 u 1 pour l'ensemble des combinaisons des variables)

variable	Fonction NON
x	$\bar{x}$
0	**1**
1	**0**

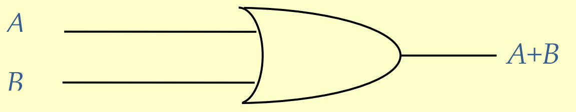


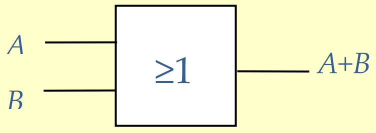


2.2. La fonction OU

- Fonction de 2 variables f(x,y)
- Aussi appelée **addition logique**, somme logique
- Notée avec le signe **+** de l'arithmétique traditionnelle
  $f(x, y) = x + y$ (on prononce x ou y)
- Autre notation parfois utilisée :
  $f(x, y) = x\ V\ y$
- Table de vérité et symboles :

Variables		Fonction OU
x	y	x+y
0	**0**	**0**
0	**1**	**1**
1	**0**	**1**
1	**1**	**1**

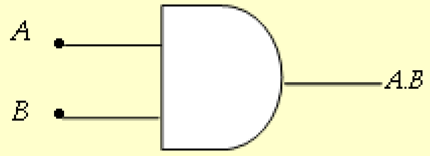


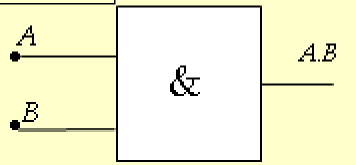


2.3. La fonction ET

- Fonction de 2 variables f(x,y)
- Aussi appelée multiplication logique, produit logique
- Notée avec le signe **.** de l'arithmétique traditionnelle
  $f(x,y) = x.y$ (on prononce xy) ou même sans le . entre les 2 variables s'il n'y a pas de doute possible
- Autre notation parfois utilisée :
  $f(x,y) = x \wedge y$
- Table de vérité et symboles :

Variables		Fonction ET
x	y	x.y
0	**0**	**0**
0	**1**	**0**
1	**0**	**0**
1	**1**	**1**





ESIR ECOLE SUPERIEURE D'INGENIEURS DE RENNES

2.4. Extension des fonctions ET et OU

- Les fonctions ET et OU s'étendent à plus de deux variables
- Table de vérité :

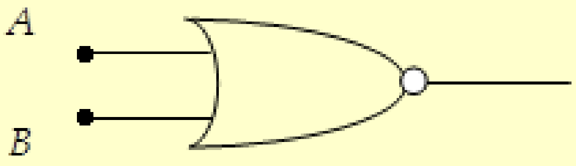
Variables			Fonction ET	Fonction OU
x	y	z	x.y.z	x+y+z
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

3. Autres fonctions courantes de 2 variables

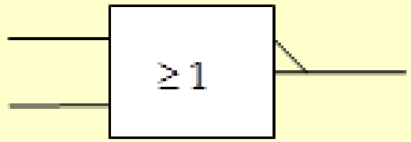
3.1. Fonction NON-OU (NI, NOR)

- Fonction OU inversée
- Notée : $f(x, y) = \overline{x + y}$ (x ou y barre)
- Table de vérité :

Variables		Fonction OU	Fonction NOR
x	y	x+y	$\overline{x + y}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0



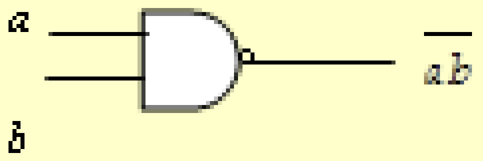
$x = \overline{A + B}$

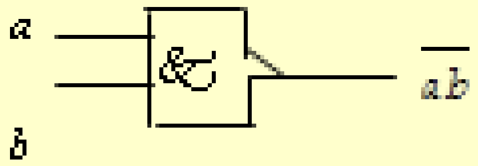


3.2. Fonction NON-ET (NAND)

- Fonction ET inversée
- Notée : $f(x, y) = \overline{x.y}$ (x et y barre)
- Table de vérité :

Variables		Fonction ET	Fonction NOR
x	y	x.y	$\overline{x.y}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

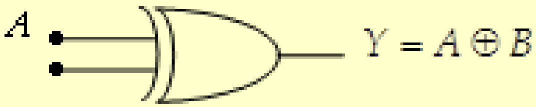


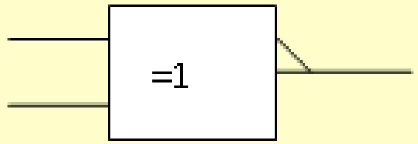


3.3. Fonction OU exclusif (XOR)

- Fonction de deux variables vaut 1 si et seulement si une seule des 2 variables vaut 1
- Notée : $f(x, y) = x \oplus y$ (x ou exclusif y)
- Table de vérité et symboles :

Variables		Fonction OU exclusif
x	y	x⊕y
0	0	0
0	1	1
1	0	1
1	1	0





- Propriétés du OU EXCLUSIF :

  $a \oplus b = \bar{a}.b + a.\bar{b}$ $\qquad a \oplus 0 = a$ $\qquad a \oplus 1 = \bar{a}$ $\qquad a \oplus a = 0$ $\qquad a \oplus \bar{a} = 1$
- OU EXCLUSIF à 3 variables

4. Propriétés des opérateurs ET, OU, NON

- Identités utiles dans l'étude de fonctions complexes
- Peuvent être démontrées à partir des tables de vérité des fonctions ET, OU, NON
- Parfois proches de l'algèbre classique…. : $x+0=x$; $x.0=0$
- Mais pas toujours : $x+1=1$; $x.x=x$; $x+x=x$

Fonctions	Propriétés	Observation
1 variable	$\bar{\bar{x}} = x$ $x + 0 = x$ et $x.0 = 0$ $x + 1 = 1$ et $x.1 = x$ $x + x = x$ et $x.x = x$ $x + \bar{x} = 1$ et $x.\bar{x} = 0$	
2 variables	$x + y = y + x$ $x.y = y.x$	Commutativité
3 variables	$x + y + z = (x + y) + z = x + (y + z)$ $x.y.z = x.(y.z) = (x.y).z$ $x.(y + z) = x.y + x.z$ $x + y.z = (x + y).(x + z)$	Associativité Distributivité
Nb quelconque de variables	$\overline{x + y + z + \cdots} = \bar{x}.\bar{y}.\bar{z}$ .... $\overline{x.y.z\ \ldots.} = \bar{x} + \bar{y} + \bar{z} + \cdots$	Théorème de **De Morgan**
Fonction quelconque	Elle peut s'exprimer avec les seuls opérateurs ET, OU, NON	

Exercice d'application

Réaliser les fonctions NON, ET, OU, avec des NAND uniquement et avec des NOR uniquement

a) Fonction NON avec des NAND

b) Fonction NON avec des NOR

c) Fonction ET (à deux entrées) avec des NAND

d) Fonction ET (à deux entrées) avec des NOR

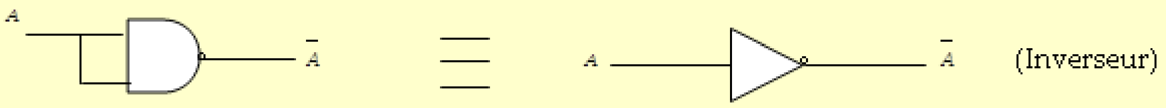
e) Fonction OU (à deux entrées) avec des NAND

f) Fonction OU (à deux entrées) avec des NOR

Correction

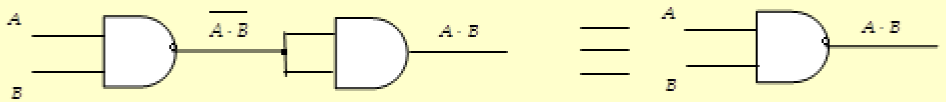
Réaliser les fonctions NON, ET, OU, avec des NAND et des NOR

a) Fonction NON avec des NAND



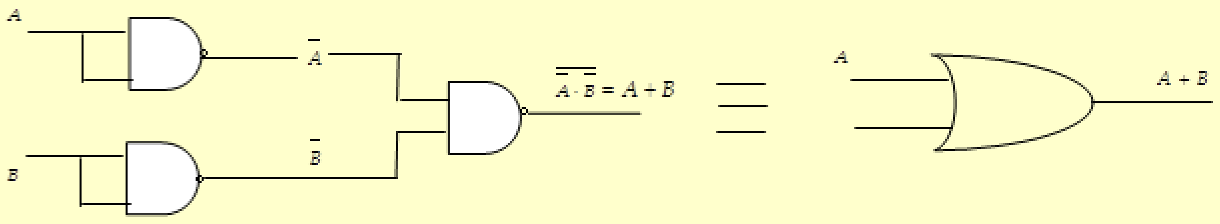
b) Fonction NON avec des NOR

c) Fonction ET (à deux entrées) avec des NAND



d) Fonction ET (à deux entrées) avec des NOR

e) Fonction OU (à deux entrées) avec des NAND



f) Fonction OU (à deux entrées) avec des NOR

5. Fonctions quelconques

5.1. Fonctions complètement définies

- Fonctions élémentaires ont des noms particuliers car elles sont très fréquemment utilisées
- Existent sous forme de circuits logiques dans les catalogues des constructeurs (NON, ET, OU, NOR, NAND…)
- Autres fonctions = fonctions quelconques, sans nom, définies par une table de vérité
- Exemple d'une table de vérité pour une fonction quelconque de 3 variables :

x	y	z	Fonction F(x,y,z)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

5.2. Fonctions incomplètement définies

- Pour l'instant : toutes les fonctions vues prenaient une valeur pour toutes les combinaisons des variables d'entrée
- Pour n variables, il y a 2^n combinaisons
- Pour certaines combinaisons :
  - ✓ **Combinaison impossible** (en fonction du sens physique des variables)
  - ✓ Fonction peut prendre la **valeur 0 ou 1 sans importance**
- Fonction f **incomplètement définie**
- Dans table de vérité : on note X (f peut prendre la valeur 1 ou 0), exemple :

x	y	z	Fonction F(x,y,z)
0	0	0	0
0	0	1	X
0	1	0	1
0	1	1	0
1	0	0	X
1	0	1	1
1	1	0	0
1	1	1	X

6. Ecritures canoniques d’une fonction logique

- Toute expression booléenne peut être convertie en l’une des deux formes standard :

  ✓ Somme de produits
  ✓ Produit de sommes

- Standardisation permet de :

  ✓ Evaluer
  ✓ Simplifier
  ✓ Mettre en œuvre

..de manière simple et méthodique (ex : circuits programmables)

6.1. Forme canonique Somme de produits

- Une forme canonique Somme de produits ne contient que des produits incluant **toutes les variables ou leur complément** (ex : pour une fonction à 3 variables, tous les produits de la somme comportent 3 variables représentées sous **forme normale** ou **complémentée**).

Ex : $f(a,b,c) = a\bar{b}c + \bar{a}\bar{b} + ab\bar{c}d$ peut s'écrire sous la forme canonique somme de produits :

$$a\bar{b}cd + abcd + \bar{a}\bar{b}cd + \bar{a}\bar{b}c\bar{d} + \bar{a}\bar{b}\bar{c}d + \mathrm{a}\bar{b}\bar{c}\bar{d} + ab\bar{c}d$$

- Chaque produit (appelé **minterme**) comporte les 4 variables d'entrées de la fonction (forme normale a, b, c ou d ou forme complémentée $\bar{a}$, $\bar{b}$, $\bar{c}$, $\bar{d}$)

- Construction de la forme canonique Somme de produits **à partir de la table de vérité** :
  - ✓ Nombre de produits = nombre de 1 dans la table de vérité de la fonction
  - ✓ Produit = produit des variables pour lesquelles la fonction vaut 1 (avec variable normale si elle vaut 1 et complémentée sinon)

Exercice d’application :

Donner la forme canonique somme de produits de la fonction dont la table de vérité est :

Entrées			Sortie
A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Correction

- Il y a 4 produits car la fonction vaut 1 pour 4 combinaisons des entrées.

- Les produits sont :

$$\bar{a}\bar{b}c, \bar{a}b\bar{c}, a\bar{b}\bar{c}, abc$$

- La fonction F est donc une somme de ces termes :

$$F = \bar{a}\bar{b}c + \bar{a}b\bar{c} + a\bar{b}\bar{c} + abc$$

- Construction de la forme canonique Somme de produits **à partir d’une expression quelconque** :

  ✓ Toute expression logique peut être convertie sous la forme canonique d’une somme de produits en appliquant les techniques de l’algèbre de Boole.

  ✓ Si un terme de l’expression ne contient pas toutes les variables, les variables manquantes sont introduites en utilisant la règle : $a + \bar{a} = 1$.

  ✓ Méthode : multiplier chaque terme de la forme non standard par la somme de la variable manquante et de son complément (cette somme valant 1, cela ne change rien).

  ✓ Exemple : Montrez par cette méthode que la fonction $f(a, b, c) = a\bar{b}c + \bar{a}\bar{b} + abc\bar{d}$ peut s’écrire sous la forme canonique somme de produits :

$$a\bar{b}cd + abcd + \bar{a}\bar{b}cd + \bar{a}\bar{b}c\bar{d} + \bar{a}\bar{b}\bar{c}d + a\bar{b}\bar{c}\bar{d} + abc\bar{d}$$

  Rm : on commencera par multiplier le premier terme par $(d + \overline{d})$ et ainsi de suite.

6.2. Forme canonique Produit de sommes

- Cette forme canonique est moins utilisée que la précédente
- Pour l'obtenir à partir d'une table de vérité :
  - ✓ Associer à chaque ligne de la table de vérité où la fonction vaut 0, une somme construite à partir des compléments des variables de la ligne (pour la ligne x=0 y=0 z=0, la somme est x+y+z)
  - ✓ Faire le produit de toutes ces sommes
- Exemple 1 : forme canonique somme de produits de la fonction dont la table de vérité est la suivante

Entrées			Sortie
A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

On trouve : $F = (A + B + C)(A + \bar{B} + \bar{C})(\bar{A} + B + \bar{C})(\bar{A} + \bar{B} + C)$

- Exemple 2 : forme canonique produit de sommes de la fonction dont la forme générale est la suivante :

$$\mathrm{f} = \mathrm{a} + \mathrm{bc}$$

✓ **Distributivité** : $f = a + bc = (a + b)(a + c)$

✓ Lorsqu'il manque une variable dans une somme : on l'ajoute en utilisant la **propriété** $\mathbf{x.\bar{x} = 0}$

$$\mathrm{f} = (\mathrm{a} + \mathrm{b})(\mathrm{a} + \mathrm{c}) = (\mathrm{a} + \mathrm{b} + \mathrm{c}\bar{\mathrm{c}})\left(\mathrm{a} + \mathrm{c} + \mathrm{b}\bar{\mathrm{b}}\right)$$

✓ **Distributivité** à nouveau :

$$f = (a + b + c\bar{c})\left(a + c + b\bar{b}\right) = (a + b + c)(a + b + \bar{c})(a + c + b)(a + c + \bar{b})$$

ECOLE SUPERIEURE D'INGENIEURS DE RENNES

7. Simplification de l'écriture de fonctions logiques

- Algèbre classique : simplifier une fonction = rechercher écriture plus condensée ou plus facile à calculer
- Exemple : $f(a,b) = \frac{a^2-b^2}{a+b}$ devient $f(a,b) = a - b \; (si \; a + b \neq 0)$

- Algèbre de Boole : simplifier = trouver une **forme plus condensée**, avec moins de symboles, dont la **réalisation matérielle est plus compacte**

- Méthodes de simplification basées essentiellement sur propriétés suivantes :

$$x + \bar{x} = 1, \quad x\bar{x} = 0, \;\; 1 + x = 1$$

- Principalement 2 méthodes :

  ✓ Simplification par manipulation algébrique
  ✓ Simplification par tableaux de Karnaugh

7.1. Simplification algébrique

- Simplification de la fonction définie par la table de vérité :

x	y	z	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

- Forme canonique somme de produits :

$$f = \bar{x}yz + x\bar{y}z + xy\bar{z} + xyz$$

- Nouvelle écriture (en utilisant la **propriété x + x = x**):

$$f = (\bar{x}yz + xyz) + (x\bar{y}z + xyz) + (xy\bar{z} + xyz)$$

- Simplification :

$$f = yz(x + \bar{x}) + xz(y + \bar{y}) + xy(z + \bar{z})$$

- Forme simplifiée :

$$f = yz + xz + xy$$

Exercice d'application :

Simplifier par la méthode algébrique la fonction suivante :

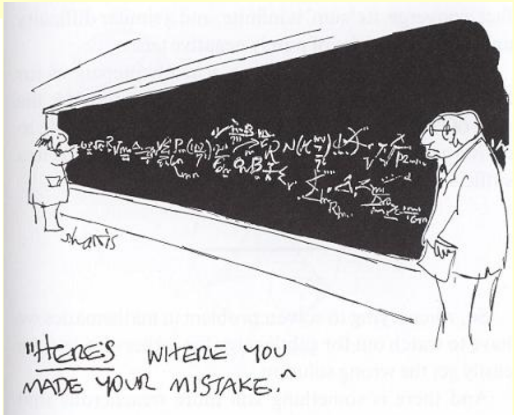
$$f = \bar{a}bcd + a\bar{b}\bar{c}d + abc\bar{d} + acd + \bar{a}b + c\bar{d} + \bar{b}c\bar{d} + a\bar{c} + \bar{a}b\bar{c}d + \bar{b}c + abc$$

Correction :

- $f = a + b + c$

- Repose beaucoup sur astuce et chance de voir les bons regroupements à faire

- Risque de s'y perdre !



- Au-delà de 3 variables, calculs difficiles, méthode assez longue à appliquer

7.2. Simplification par la méthode de Karnaugh

- Construction des tableaux
- Méthode utilise l'identité : $ax + a\bar{x} = a$
- Mettre en évidence graphiquement les termes d'une fonction qui ne diffèrent que par l'état d'une variable (ex : $x\bar{y}zt$ et $xyzt$). Le regroupement de ces deux termes permet de simplifier.
- N variables ⇒ 2^n produits de variables
- Chaque produit représenté par une case du tableau
- Le tableau est construit de telle façon que l'état d'une seule variable change entre une ligne et la suivante, idem en colonne.
- Exemples :

z \ xy	00	01	11	10
0	$\bar{x}\bar{y}\bar{z}$	$\bar{x}y\bar{z}$	$xy\bar{z}$	$x\bar{y}\bar{z}$
1	$\bar{x}\bar{y}z$	$\bar{x}yz$	xyz	$x\bar{y}z$

- Utilisation du code Gray pour remplir les entrées des tableaux

Le code GRAY

- Permet l'incrémentation ou la décrémentation d'une valeur par la modification d'un bit unique
- Permet donc de réaliser ces opérations très rapidement (et avec une logique très simple) car n'aura pour effet que de changer un bit (∀ longueur du code)
- Encore appelé code binaire réfléchi (construction)
- Exemple du code Gray à 3 variables :

Binaire pur

0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

1 changement
2 changements
1 changement
3 changements
1 changement
2 changements
1 changement

Gray - Binaire réfléchi

c a

0	0	0
0	0	1
0	1	1
0	1	0
1	1	0
1	1	1
1	0	1
1	0	0

1
b
2

d e

- Tableau de Karnaugh à 4 variables (x, y, z, t) :

xy / zt	00	01	11	10
00				
01				
11				
10				

- Tableau de Karnaugh à 5 variables (x, y, z, t, u)

u=0

xy / zt	00	01	11	10
00				
01				
11				
10				

u=1

xy / zt	00	01	11	10
00				
01				
11				
10				

- Tableau de Karnaugh à 5 variables (x, y, z, t, u) / Autre possibilité (utiliser le code Gray pour remplir les entrées de façon à garantir qu'une seule variable change):

ztu xy	000	001	011	010	110	111	101	100
00								
01								
11								
10								

- Cases adjacentes :

xy zt	00	01	11	10
00				
01			X	
11		X		X
10			X	

xy zt	00	01	11	10
00		X		X
01	X			
11				
10	X			

xy zt	00	01	11	10
00		X		X
01			X	
11				
10			X	

- Passage de la table de vérité au tableau de Karnaugh : écrire des 1 dans les cases correspondant aux combinaisons où la fonction vaut 1, et des zéros ailleurs

xyzt	F
0000	1
0001	0
0010	1
0011	0
0100	0
0101	1
0110	0
0111	0
1000	1
1001	1
1010	1
1011	1
1100	0
1101	1
1110	0

zt xy	00	01	11	10
00	1	0	0	1
01	0	1	0	0
11	0	1	0	0
10	1	1	1	1

1111	0

- Passage de l'expression de la fonction au tableau de Karnaugh : écrire des 1 dans les cases correspondant à chaque terme de la somme, et des zéros ailleurs :
- Exemple : $f = ab\bar{c} + \bar{a}b + cb$
  Tableau :

c / ab	0	1
00	0	0
01	1	1
11	1	1
10	0	0

- Simplification par la **méthode** de Karnaugh :

  ✓ Transposer la table de vérité (ou l'expression non simplifiée de la fonction) dans un tableau de Karnaugh
  ✓ Regrouper les cases comprenant des 1
  ✓ Les regroupements ont les tailles 2, 4, 8… (puissance de 2)

- ✓ On cherche à faire le moins de regroupements possible (donc chaque regroupement doit rassembler un maximum de cases)
- ✓ Si une case ne peut pas être regroupée, le terme correspondant apparaît tel quel dans l'expression
- ✓ Dans un groupement de 2 termes (2 cases), on élimine la variable qui change d'état, et on conserve le produit des variables (directes ou inverses) qui n'ont pas changé d'état dans le groupement (car $(ax + a\bar{x}) = a$)
- ✓ Dans un groupement de 4 termes, on élimine les deux variables qui changent d'état et l'on conserve le produit des variables directes ou inverses qui n'ont pas changé d'état
- ✓ L'expression logique finale (simplifiée) est un OU entre l'expression des groupements.

Exercice d'application n°1 :

Donner l'expression simplifiée des fonctions F, G, H de 4 variables (x, y, z, t), représentées par les tableaux de Karnaugh suivants :

F :

zt / xy	00	01	11	10
00	0	0	0	0
01	0	**1**	**1**	0
11	0	0	0	0
10	**1**	**1**	**1**	**1**

G :

zt / xy	00	01	11	10
00	**1**	0	0	**1**
01	0	**1**	**1**	**1**
11	0	**1**	**1**	0
10	**1**	0	0	**1**

H :

zt / xy	00	01	11	10
00	0	**1**	**1**	0
01	**1**	**1**	**1**	0
11	**1**	**1**	**1**	0
10	0	**1**	**1**	0

Correction :

$$F = x\bar{y} + \bar{x}yt$$