Tutorat logique : TD2

Université François Rabelais

`Département informatique de Blois`

Logique pour l'informatique

*
* *

Problème 1

L'opérateur *Nand* noté $\uparrow$ est un opérateur très utilisé en électronique et dans la réalisation des microprocesseurs car il forme un système complet de connecteurs à lui seul.
On rappelle que sa table de vérité est telle que :

x	y	$x \uparrow y$
0	0	1
0	1	1
1	0	1
1	1	0

Son symbole associé en électronique est :

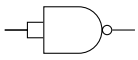

FIGURE 1 : Porte logique de l'opérateur $\uparrow$ NAND

1. Exprimer l'opérateur "ou exclusif" noté $\oplus$ à l'aide des opérateurs classiques puis uniquement en utilisant Nand.

   $x \oplus y = (\neg x \wedge y) \vee (x \wedge \neg y) = (x \vee y) \wedge (\neg x \vee \neg y)$

   On sait que l'on peut exprimer les opérateurs classiques uniquement à l'aide Nand étant donné qu'il forme un système complet de connecteurs. Dès lors :

   - $\neg x = x \uparrow x$
   - $x \vee y = (x \uparrow x) \uparrow (y \uparrow y)$
   - $x \wedge y = (x \uparrow y) \uparrow (x \uparrow y)$

   Ainsi, il vient que en simplifiant au maximum :

$$x \oplus y = [(x \uparrow y) \uparrow x] \uparrow [y \uparrow (x \uparrow y)]$$

2. On considère la modélisation de l'opérateur $\oplus$ suivante :

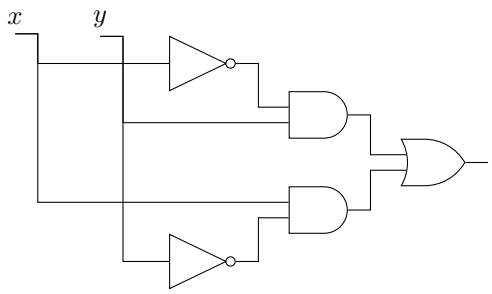

FIGURE 2 : Circuit logique de l'opérateur $\oplus$ OU EXCLUSIF

Expliquer pourquoi cette solution n'est pas satisfaisante. Proposer un circuit logique à l'aide de Nand. Pourquoi cette modélisation est meilleure ?

Cette solution n'est pas satisfaisante car l'on n'a pas essayé de minimiser le nombre de portes logiques pour la réalisation du circuit, de plus, on utilise trois sortes de portes ce qui oblige à réaliser différentes commandes de matériel. Une modélisation à l'aide de l'opérateur Nand est préférable car on utilise uniquement un type de porte ce qui est préférable et on a seulement quatre composants.

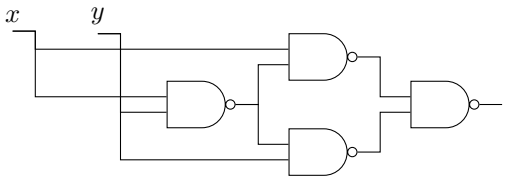

FIGURE 3 : Circuit logique de l'opérateur $\oplus$ OU EXCLUSIF à partir de l'opérateur $\uparrow$ NAND

Problème 2

Soit un chiffre $x \in [\![0, 9]\!]$.

1. Donner la représentation binaire de x.

$0 =< 0 >_2$

$1 =< 1 >_2$

$2 =< 10 >_2$

$3 =< 11 >_2$

$4 =< 100 >_2$

$5 =< 101 >_2$

$6 =< 110 >_2$

$7 =< 111 >_2$

$8 =< 1000 >_2$

$9 =< 1001 >_2$

2. On considère un vecteur booléen (A, B, C, D) permettant de représenter x en binaire. On souhaite réaliser un affichage de calculatrice tel que :

$$(A, B, C, D) \longrightarrow \Phi \longrightarrow \begin{vmatrix} \overline{\ } \\ \_ \\ \_ \end{vmatrix}$$

Donner la fonction associée à chaque segment de l'affichage puis l'écriture de la fonction Φ.

On donne la fonction Θ représentant la condition d'activation de la barre horizontale du milieu de la figure.

A	B	C	D	Θ
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	φ
1	0	1	1	φ
1	1	0	0	φ
1	1	0	1	φ
1	1	1	0	φ
1	1	1	1	φ

En effet, on remarque que certaines combinaisons sont interdites, cependant, ce n'est pas une raison pour les bannir. On remplace ces combinaisons par une variables $\varphi \in \{0, 1\}$ qui va nous servir lors de la simplification par tableau de Karnaugh, on pourra ainsi la considérer comme à

1 pour réaliser des simplification plus avantageuses.

ab \ cd	00	01	11	10
00	0	0	1	1
01	1	1	0	1
11	φ	φ	φ	φ
10	1	1	φ	φ

Il vient alors que $\Theta = A \vee (B \wedge \neg C) \vee (C \wedge D) \vee (\neg B \wedge C)$

Problème 3

Simplifier algébriquement les expressions suivantes :

1. $(A \wedge \neg B \wedge C) \vee (\neg A \wedge C) \vee (A \wedge \neg B) \vee (\neg A \wedge B \wedge C)$

   En vertu des *lois l'absorption*, et de la *commutativité* de $\vee$ et $\wedge$ il vient que $(\neg A \wedge C) \vee (\neg A \wedge B \wedge C) = \neg A \wedge C$ et $(A \wedge \neg B \wedge C) \vee (A \wedge \neg B) = A \wedge \neg B$

   On obtient le résultat simplifié :

   $$(A \wedge \neg B) \vee (\neg A \wedge C)$$

2. $(A \wedge B \wedge \neg C \wedge \neg D) \vee (A \wedge B \wedge D) \vee (A \wedge C) \vee (B \wedge C) \vee (\neg A \wedge C)$

   On applique la *factorisation* aux termes.

   $(A \wedge B \wedge \neg C \wedge D) \vee (A \wedge B \wedge D) = (A \wedge B) \wedge [(\neg C \wedge \neg D) \vee D]$

   De plus par *absorption* (i.e. *développement - tier-exlus*), $(\neg C \wedge \neg D) \vee D = \neg C \vee D$

   Par *factorisation* et par le *tier-exclus*, $(A \wedge C) \vee (\neg A \wedge C) = C \vee (A \wedge \neg A) = C$

   De là, il vient : $(A \wedge B \wedge \neg C) \vee (A \wedge B \wedge D) \vee C \vee (B \wedge C)$

   On utilise deux fois l'*absorbance* :

   $(A \wedge B \wedge \neg C) \vee (A \wedge B \wedge D) \vee C = (A \wedge B) \vee (A \wedge B \wedge D) \vee C$

   On utilise le théorème d'*absorption* une dernière fois sur la conjonction $A \wedge B$. Enfin, on obtient le résultat :

   $$(A \wedge B) \vee C$$

3. $A \vee (\neg B \wedge C) \vee (A \wedge D \neg E) \vee (A \wedge B \wedge C \wedge D \wedge E) \vee (B \wedge \neg C \wedge D \neg E) \vee (\neg A \wedge C \wedge D)$

   Par *théorème de l'inclusion*, $A \wedge D \wedge \neg E \subseteq A$. On a :

   $A \vee (\neg B \wedge C) \vee (B \wedge \neg C \wedge D \neg E) \vee (\neg A \wedge C \wedge D)$

Par *absorption* de A et $\neg A \wedge C \wedge D$, on a :

$A \vee (\neg B \wedge C) \vee (B \wedge \neg C \wedge D \neg E) \vee (C \wedge D)$

Par *factorisation* :

$A \vee (\neg B \wedge C) \vee [(B \wedge \neg C \wedge \neg E) \vee C] \wedge D$

Par *absorption* de $\neg C$:

$A \vee (\neg B \wedge C) \vee [(B \wedge \neg E) \vee C] \wedge D$

Soit le résultat simplifié :

$$A \vee (\neg B \wedge C) \vee (D \wedge B \wedge \neg E) \vee (D \wedge C)$$

Problème 4

Soient a, b, c et d quatre variables booléennes. On considère les formules logiques Φ et Ψ définies telles que :

- $\Phi = 1$ si et seulement si $a + b \leq c + d$. Avec "+" représentant l'addition usuelle
- $\Psi = 1$ si et seulement si l'entier dont l'écriture en base 2 de $abcd$ est strictement inférieur à 10.

À l'aide des tableaux de Karnaugh, donner l'expression la plus simple de Φ et Ψ.

Les tableaux de Karnaugh pour les formules Φ et Ψ sont respectivement :

ab \ cd	00	01	11	10
00	1	1	1	1
01	0	1	1	1
11	0	0	1	1
10	0	0	0	1

ab \ cd	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	0	0	0	0
10	1	1	0	0

On obtient les formules $\Phi = (\neg a \wedge \neg b) \vee (c \wedge \neg d) \vee (\neg a \wedge d) \vee (b \wedge c)$ et $\Psi = \neg a \vee (\neg b \wedge c)$

Problème 5

On considère la formule logique suivante :

$$\varphi = [(\neg a \vee b) \wedge c] \Leftrightarrow [a \oplus c]$$

1. Exprimer φ sous forme normale disjonctive puis sous forme normale conjonctive.

   On dresse la table de vérité de φ :

a	b	c	$\neg a \vee b$	$(\neg a \vee b) \wedge c$	$a \oplus c$	φ
0	0	0	1	0	0	1
0	0	1	1	1	1	1
0	1	0	1	0	0	1
0	1	1	1	1	1	1
1	0	0	0	0	1	0
1	0	1	0	0	0	1
1	1	0	1	0	1	0
1	1	1	1	1	0	0

Sous forme normale disjonctive, on a $\varphi = (\neg a \wedge \neg b \wedge \neg c) \vee (\neg a \wedge \neg b \wedge c) \vee (\neg a \wedge b \wedge \neg c) \vee (\neg a \wedge b \wedge c) \vee (a \wedge \neg b \wedge c)$.

De même, $\neg\varphi = (a \wedge \neg b \wedge \neg c) \vee (a \wedge b \wedge \neg c) \vee (a \wedge b \wedge c)$, donc sous forme normale conjonctive, on a $\varphi = (\neg a \vee b \vee c) \wedge (\neg a \vee \neg b \vee c) \wedge (\neg a \vee \neg b \vee \neg c)$.

2. À l'aide des tableaux de Karnaugh, simplifier les expressions obtenues.

On forme le tableau de Karnaugh suivant :

a \ bc	00	01	11	10
0	1	1	1	1
1	0	1	0	0

Il vient que $\varphi = \neg a \vee (\neg b \wedge c)$. Si on préfère une conjonction, on a $\neg\varphi = (a \wedge b) \vee (a \wedge \neg b \wedge \neg c)$, soit $\varphi = (\neg a \vee \neg b) \wedge (\neg a \vee b \vee c)$.

Problème 6

On travaille avec un langage de programmation qui ne dispose que d'une unique instruction :

$$a \leftarrow b \oplus c$$

où a, b, c sont des variables pouvant contenir des entiers codés sur n bits.
L'instruction range dans la variable a l'entier dont la représentation binaire $a_{n-1}...a_1a_0$ est définie telle que :

$$\forall k \in [\![0, n-1]\!], a_k = b_k \oplus c_k$$

Soient u et v deux variables. Donner une suite d'instructions à exécuter pour échanger les valeurs contenues dans les variables u et v sans utiliser d'autre variable.

La loi $\oplus$ étant associative dans $\mathbb{Z}/2\mathbb{Z}$, nous avons :

$$a \oplus (a \oplus b) = b \quad \text{et} \quad [a \oplus (a \oplus b)] \oplus (a \oplus b) = a$$

Considérons alors la séquence d'instructions suivante :

1. $v \leftarrow u \oplus v$
2. $u \leftarrow u \oplus v$
3. $v \leftarrow u \oplus v$

Si au départ u contient l'entier a et que v contient l'entier b, alors :

- Après la première instruction, u contient a et v contient $a \oplus b$,
- après la deuxième instruction u contient $a \oplus (a \oplus b) = b$ et v contient $a \oplus b$,
- après la troisième instruction u contient b et v contient $b \oplus (a \oplus b) = a$.

Les deux références ont vu leur contenus échangés.