Logique Combinatoire

Arthur Garnier

February 4, 2015

1 Introduction

Comment *traite*-on des mots binaires ?

X = $x_{n-1}, x_{n-2}...x_2, x_1, x_0 \in B^N$

$B = \{\emptyset; 1\}$

$B^2 = B * B \ni (00, 01, 11)$

B^N = B*B. . . *B = {emsemble de bus des mots binaires à N bits}

X –N–>[F]–M–>Y

$$\begin{cases} \text{–N–>} \\ \text{–P–>} \end{cases} \quad \text{[ ]–M–>}$$

2 Fonction 1 bit vers 1 bit

X–>[F]–>Y

Table de vérité :

X	Y
0	0
1	0

	X	0	1	
f0	Y	0	0	Source ∅
f1	Y	0	1	Identité
f2	Y	1	0	NOT
f3	Y	1	1	Source 1

3 Fonctions à deux bits vers 1 bit

$$\begin{cases} \text{A–>} \\ \text{B–>} \end{cases} \quad \text{[g]–>Y}$$

	AB	00	01	10	11
f0	Y	0	0	0	0
f1		0	0	0	1
f2		0	0	1	0
f7		0	1	1	1
f8		1	0	0	0
f14		1	1	1	0

Fonction OU

AB	0	1
0	0	1
1	1	1

Non-Ou

AB	0	1
0	1	0
1	0	0

4 Algèbre de Boole

- A+B = B+A
- A*B = B*A
- A+(B+C)=(A+B)+C
- A*(B*C)=(A*B)*C
- A*0=0
- A+0=A
- A*1=A
- A*(B+C)=A*B+A*C
- A*A=A
- A+A=A
- A+1 = 1
- A+(B*C) = (A+B)*(A+C)

4.1 Loi de Morgan

- $\overline{A+B} = \overline{A} * \overline{B}$
- $\overline{A*B} = \overline{A} + \overline{B}$

5 Fonctions de N bits de rentrée vers un bit de sortie

$\varphi(A, B, C) = A * B + A * C + B * C$

$\sigma(A, B, S) = \overline{S} * A + S * A$

5.1 Polynome booléen

Exemple :

$Y = A * \overline{B} + \overline{A} * B$

$Z = A * B * C + \overline{A} * BC = (A + \overline{A}) * BC = BC$

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

Table de Karnaugh :

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

6 Etude d'un opérateur combinatoire

6.1 Spécification

6.1.1 Interface

- Graphique
- Table des signaux

Shéma

Signaux	Largeur	Dir	Sélmemantique
A	4	entrée	opérande source
B	4		
P	2		Commande
R	4	Sortie	Résultat
C	1		Indication

6.1.2 Comportement

- Expression algébrique
- Table de vérité étendue
- Table de vérité

P	C	R
00		A
01		B
10	A_0	AΨB
11	A_{N-1}	AψB

6.2 Analyse

- Décomposition récursive de l'opérateur en opérateurs plus simples que l'on respécifie jusqu'à avoir des opérateurs terminaux à quelques bits d'entrée et de sortie
- Détermination de chaque bit de sortie de chacun des opérateurs terminaux au moyen d'un polynome booléen minimal

7 Synthèse

Indiquer comment réaliser les opérateurs terminaux avec des fonctions élémentaires disponibles

Note : On peut exprimer n'importe quelle fonction booléenne avec des ponts NAND et NOT ou NOR et NOT

$Y = A * \overline{B} + \overline{A} * C = \overline{\overline{A * \overline{B} + \overline{A} * C}} = \overline{\overline{(A * \overline{B})} * \overline{\overline{A} * C}} = \overline{(A\psi\overline{B}) * (\overline{A}\psi C)} = (A\psi B)\psi(\overline{A}\psi C)$

$\overline{U} = \overline{A} * B + A * C \rightarrow U = (A\psi\overline{B})\psi(\overline{A}\psi\overline{C}$

8 Etude d'un multiplexeur à 4 entrées de 3 bits

8.1 Spécification

8.1.1 Interface

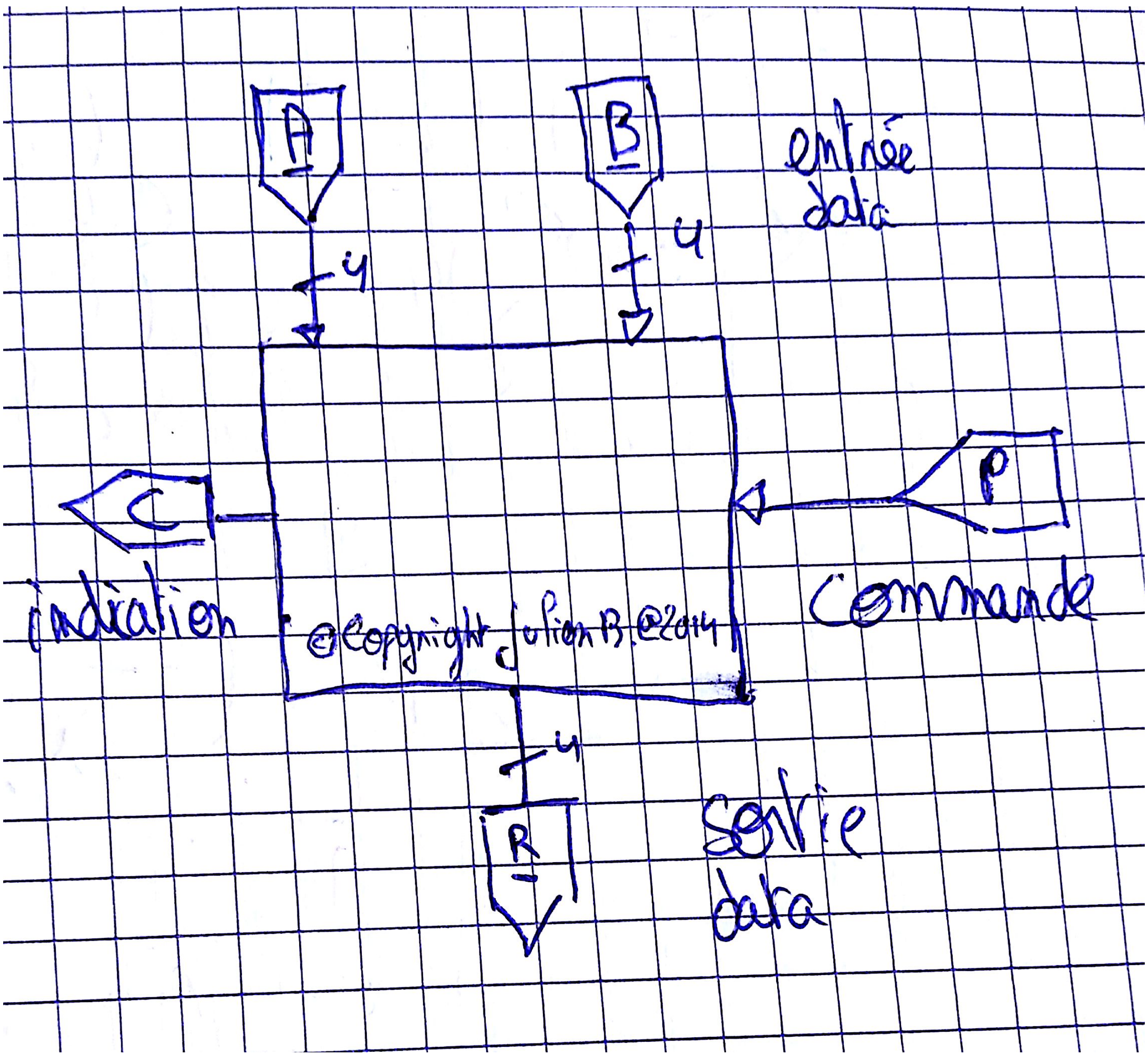

Figure 1: Interface

8.1.2 Comportement

Y=X[S]

Table de vérité étendue :

S	Y
00	X0
01	X1
10	X2
11	X3