Nom : Sujet : Jury : Page : /

23.- Fonctions et circuits booléens en architecture des ordinateurs

Niveau : L2

Prérequis : Notions d'architecture + représentation des nombres en binaire

Motivation : Comprendre la théorie derrière les circuits électroniques permettant de créer des composants essentiels à la conception des ordinateurs.

I] Fonctions booléennes :

1.) Algèbre de Boole

Définition 1 : Une algèbre de Boole est un quadruplet $(B, +, \cdot, \overline{\cdot})$ vérifiant les propriétés suivantes :

- B est un ensemble avec, au moins, deux éléments distincts, usuellement notés 0 et 1
- $\overline{\cdot}$ est une loi unaire de B, et $+$ et $\cdot$ sont des lois binaires de $B \times B$ dans B.
- 0 est l'élément neutre pour $+$: $\forall x \in B,\ 0 + x = x + 0 = x$
- 1 est l'élément neutre pour $\cdot$: $\forall x \in B,\ 1 \cdot x = x \cdot 1 = x$
- $+$ et $\cdot$ sont commutatives l'une par rapport à l'autre.
- $\forall x \in B,\ x + \overline{x} = 1$ et $x \cdot \overline{x} = 0$.

Propriété 2 : Soit $(B, +, \cdot, \overline{\cdot})$ une algèbre de Boole. Alors, $+$ et $\cdot$ sont associatives. De plus, 0 est absorbant pour $\cdot$, et 1 pour $+$.

Propriété 3 : Soit $(B, +, \cdot, \overline{\cdot})$ une algèbre de Boole. Alors, on a :

- $\forall (x_1, \dots, x_n) \in B^n,\ \overline{x_1 + \dots + x_n} = \overline{x_1} \cdot \overline{x_2} \cdot \dots \cdot \overline{x_n}$
- $\forall (x_1, \dots, x_n) \in B^n,\ \overline{x_1 \cdot \dots \cdot x_n} = \overline{x_1} + \dots + \overline{x_n}$

Ces égalités sont appelées les lois de De Morgan.

Remarque 4 : Il existe plusieurs algèbres de Boole, mais on travaille usuellement avec l'algèbre à deux éléments, où $\mathbb{B} = \{0,1\} = \{\text{Vrai}, \text{Faux}\} = \{\top, \bot\}$ et où + représente le "ou" logique, · le "et" et $\bar{\cdot}$ le "non".

Définition 5 : On appelle fonction booléenne à n variable toute application de $\mathbb{B}^n$ dans $\mathbb{B}$.

Exemple 6 : $\text{xor} : \mathbb{B}^2 \longrightarrow \mathbb{B}$
$(x,y) \longmapsto x \cdot \bar{y} + \bar{x} \cdot y$

$\text{nand} : \mathbb{B}^2 \longrightarrow \mathbb{B}$
$(x,y) \longmapsto \overline{x+y}$

2.) Représentations

Proposition 7 : On peut résumer l'ensemble des valeurs de vérité prises par une fonction booléenne dans une table. On appelle alors cette table une table de vérité.

Exemples 8 :

x	y	$\text{xor}(x,y)$
0	0	0
0	1	1
1	0	1
1	1	0

Table de vérité du xor

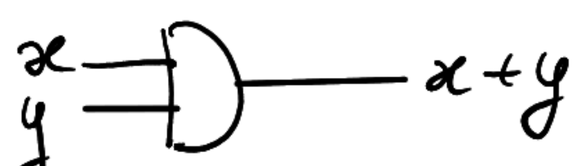
x	y	nand
0	0	1
0	1	1
1	0	1
1	1	0

Table de vérité du nand

Proposition 9 : On peut représenter les fonctions booléennes usuelles par des symboles conventionnels sur des diagrammes électriques.

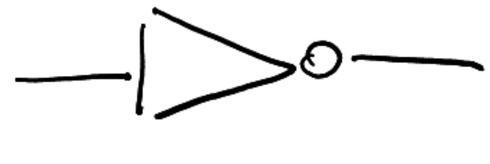
Definition 10: On appelle porte logique le symbole d'une fonction booléenne, et circuit logique un diagramme composé de plusieurs portes logiques.

Exemples 11:



Porte logique ET

Porte logique OU



Porte logique NON

Exercice 12: Donner la table de vérité et un circuit logique pour la fonction booléenne $f(x,y,z) = x\cdot\bar{y}\cdot z + y\cdot\bar{z}$.

3.) Groupes logiques complets

Définition 13: Un groupe logique complet est un ensemble de fonctions booléennes avec lesquelles on peut exprimer toute fonction booléenne, peu importe le nombre de variables.

Propriété 15: {et, ou, non} est un groupe logique complet.

Exercice 16: Montrer que {nand} est un groupe logique complet.

II) Circuits logiques en architecture

Definition 14: On distingue deux types de circuits logiques:

- les circuits combinatoires dont la sortie ne dépend que de l'état actuel des entrées
- les circuits séquentiels (ou basculs) dont la sortie dépend des entrées et des données traitées précédemment.

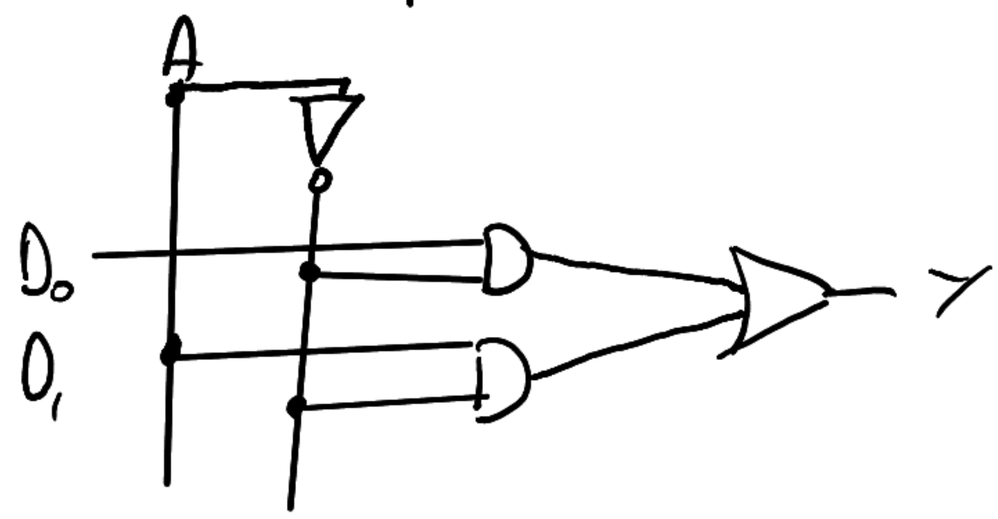
1.) Circuits combinatoires

Remarque 15 : De nombreux circuits combinatoires sont présent au sein d'un ordinateur. On verra ici les exemples du multiplexeur et de l'additionneur complet.

a.- Multiplexeurs

Definition 16 : Un multiplexeur est un circuit logique combinatoire avec plusieurs entrées et une sortie tel que la sortie est égale à l'une des entrées choisie par un signal de contrôle.

Schéma 17 : Multiplexeur à 2 entrées



Pour D = 10, on a :

A	Y
0	1
1	0

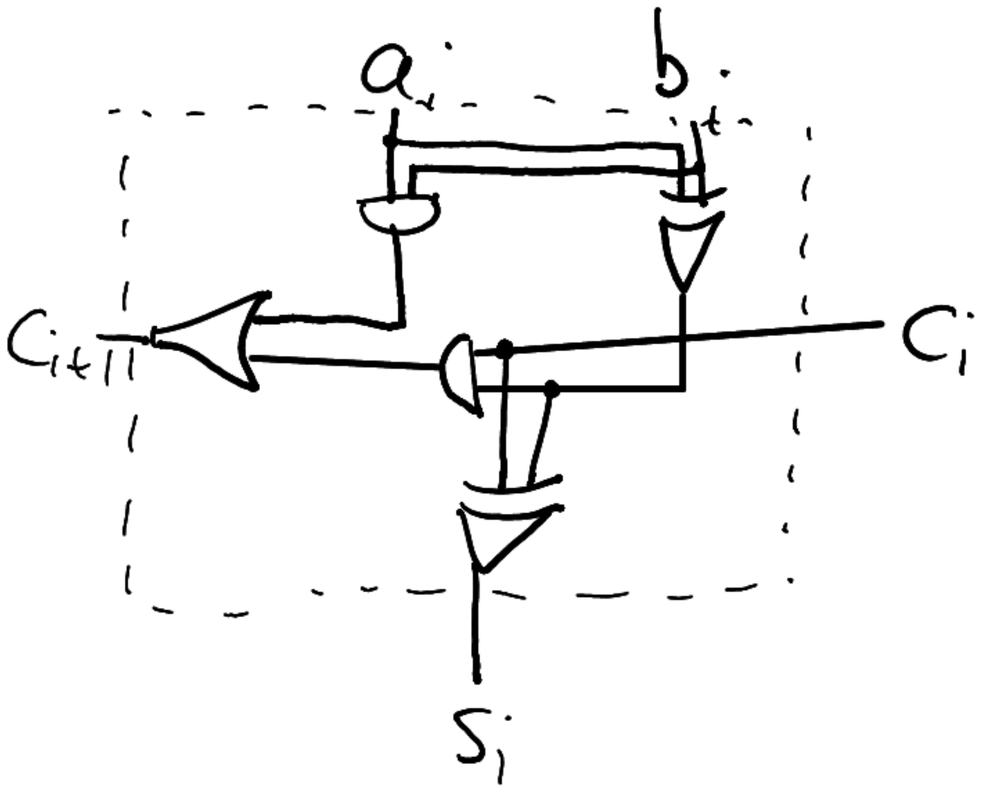
} On retrouve D

Proposition 18 : On peut généraliser les multiplexeurs à plusieurs entrées de plusieurs bits et une sortie du même nombre de bits.

b.- Additionneurs

Definitions 19 : Un additionneur simple est un circuit logique pouvant additionner deux bits en entrée et ayant pour sortie la somme et la retenue. Un additionneur complet est un circuit logique pouvant additionner trois bits en entrée et ayant deux sorties : la somme et la retenue.

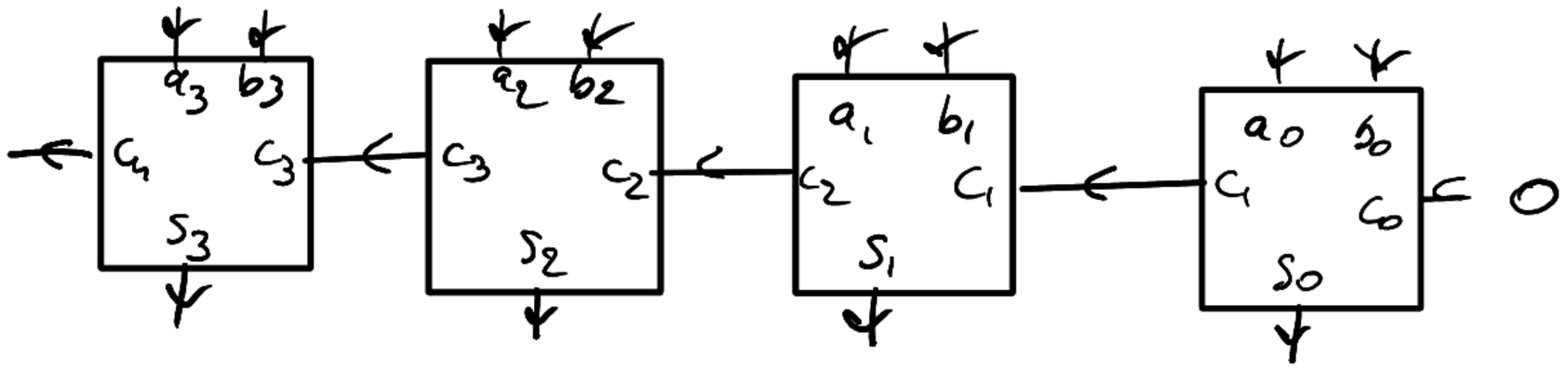
Schéma 20 : Additionneur complet



$$\begin{cases} S_i = A_i \oplus B_i \oplus C_i \\ C_{i+1} = A_i B_i + (A_i \oplus B_i) \cdot C_i \end{cases}$$

A	B	C_i	S_i	C_{i+1}
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

Schéma 21 : Additionneur sur plusieurs bits



Définition 22 : Le temps de propagation d'un circuit logique correspond au temps nécessaire à la stabilisation des sorties après modification des entrées.

Propriété 23 : Pour cet additionneur binaire, le temps de propagation est linéaire en fonction du nombre de bits des entrées.

DEV : Additionneur CLA

2.) Circuits séquentiels

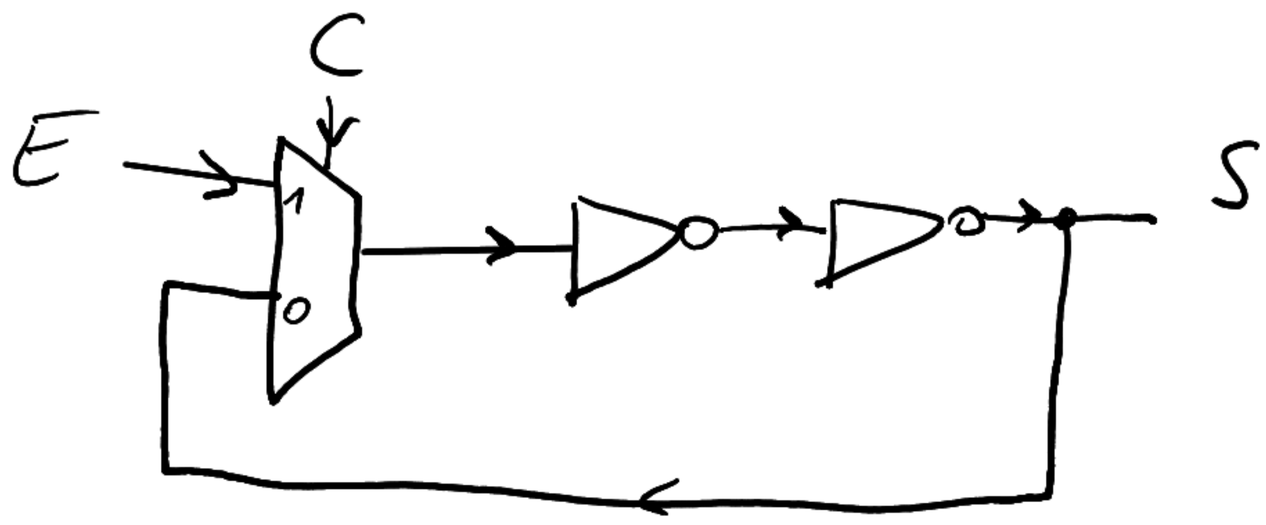
Proposition 24 : Le fait d'autoriser des cycles dans des circuits logiques permet de créer des circuits séquentiels.

Définitions 25 : On distingue deux types de circuits logiques séquentiels :
- les circuits séquentiels asynchrones dont les sorties peuvent changer à tout moment quand les entrées changent
- les circuits séquentiels synchrones dont les sorties ne peuvent changer qu'à un signal d'horloge.

Proposition 26 : Les circuits séquentiels synchrones peuvent réagir à un niveau d'horloge ou à un front d'horloge.

a.- Mémorisation élémentaire

Schéma 27 : Élément de mémorisation élémentaire



Proposition 28 : Dans ce circuit séquentiel, le multiplexeur permet de modifier l'information stockée (S) :
- si C = 0, alors le cycle conserve son état
- si C = 1, alors S va prendre la valeur de E

b.- Basculer et verrous

DEV : Basculb D