

# **Travaux dirigés :**

## **calcul booléen et circuits combinatoires**

Année universitaire 2021–2022

### **1 Introduction au calcul booléen**

Vous étudiez actuellement la logique des propositions. Une approche algébrique de cette logique a été proposée au 19ème siècle par le mathématicien britannique George Boole. On parle d'algèbre de Boole, ou de calcul booléen. Cette approche est aujourd'hui largement utilisée en informatique et dans la conception des circuits électroniques.

En calcul booléen :

- L'opérateur de disjonction est noté '+.'

- L'opérateur de conjonction est noté '.' et il peut être omis. Ainsi,  $a.b$  peut également se noter  $ab$ . Il est prioritaire sur l'opérateur de disjonction.
- L'opérateur de négation est noté en ajoutant une barre au dessus de l'expression sur laquelle il porte. On différenciera ainsi  $\bar{x}.\bar{y}$  et  $\overline{x.y}$ .
- On se limite à ces opérateurs.

1. Donner des expressions du calcul booléen pour exprimer :

(a) l'opérateur XOR (ou exclusif).

(b) l'opérateur NAND (non et).

(c) l'opérateur NOR (non ou).

2. Exprimer les deux lois de De Morgan en utilisant le calcul booléen.

3. Donner une expression du calcul booléen équivalente à la formule  $(\neg p \wedge q) \rightarrow \neg(q \leftrightarrow \neg r)$

## **2 Universalité de la porte NAND**

Nous avons vu en cours que la porte NAND est assez simple à réaliser en technologie CMOS à l'aide de deux transistors P en parallèles et deux transistors N en série. Elle est peu coûteuse en nombre de transistors et possède de façon générale de bonnes propriétés. Elle est donc utilisée autant que possible dans la conception de circuit. C'est d'autant plus facile qu'il s'agit d'une porte dite universelle, ce qui veut dire qu'on peut réaliser n'importe quelle fonction logique en utilisant uniquement des portes NAND.

1. Rappeler la table de vérité de la porte NAND et son symbole.
2. Expliquer comment réaliser une porte NOT à partir de portes NAND.
3. Expliquer comment réaliser une porte AND à partir de portes NAND.

4. Expliquer comment réaliser une porte OR à partir de portes NAND.
5. Soit l'expression du calcul booléen  $(w.x) + (y.z)$ . Proposer une réalisation sous forme d'un circuit n'utilisant que des portes NAND.

### **3 Un additionneur 4 bits efficace**

Dans cet exercice, on cherche à définir un circuit pour calculer l'addition de deux entiers naturels codés en binaire.

1. Dans un premier temps, on s'intéresse à un circuit capable d'additionner des opérandes sur 1 bit. Identifier les entrées et sorties du circuit et donner sa table de vérité.
2. Donner une formule booléenne correspondant à cette table et dessiner le circuit correspondant.
3. Supposons que l'on veuille maintenant constru-

ire un additionneur 4 bits en utilisant quatre additionneurs 1 bit.

- Quelle(s) modification(s) doit(vent) être apportée(s) au circuit proposé ci-dessus ?
- Identifier les entrées et sortie de ce nouveau circuit, donner sa table de vérité, les fonctions booléennes calculant les sorties à partir des entrées, et dessiner le circuit correspondant. Le circuit conçu aux questions précédentes est habituellement appelé un demi-additionneur, tandis que le circuit conçu à cette question est appelé additionneur complet ou tout simplement additionneur.

4. Donner le circuit d'un additionneur 4 bits en utilisant les circuits conçus aux questions précédentes.
5. En considérant un additionneur  $n$  bits conçus sur le même principe, calculer est la longueur du plus long chemin à parcourir pour réaliser

une addition avec ce circuit (on exprimera cette longueur en nombre de porte logique à traverser séquentiellement). Qu'en pensez-vous ?

6. Pour gagner en efficacité, une solution consiste à exploiter le parallélisme permis par les circuits, quitte à utiliser davantage de portes. Pour cela, on va ajouter deux signaux de sortie à l'additionneur 1 bit :

- $p$  : vaut 1 lorsque les valeurs des entrées  $a$  et  $b$  sont telles qu'une retenue entrante sera propagée en sortie, 0 sinon ;
- $g$  : vaut 1 lorsque les valeurs des entrées  $a$  et  $b$  sont telles qu'une retenue sera engendrée en sortie et ce quelle que soit la valeur de la retenue entrante, et 0 sinon.

En contrepartie, on peut supprimer la sortie correspondant à la retenue sortante. Donner les tables de vérité et les fonctions booléennes associées à  $p$  et  $g$  puis dessiner le circuit.

7. Donner une fonction booléenne qui calcule  $c_{out}$  (la retenue sortante) à partir de  $p$ ,  $g$ , et  $c_{in}$  (la retenue entrante).
8. On considère un additionneur 4 bits utilisant les signaux  $p$  et  $g$ . Donner une fonction booléenne qui calcule  $c_{out,4}$  (la retenue sortante) à partir des signaux  $p_i$ ,  $g_i$  et  $c_{in,1}$  (la retenue entrante) pour  $i \in [1, 4]$ . Quelle est la longueur du chemin de calcul de  $c_{out,4}$  ?
9. En pratique, certaines portes logiques sont plus simple à fabriquer que d'autres. Ainsi, en technologie CMOS, on privilégie autant que possible les portes NAND et NOT.
- Donner une expression de  $c_{out,4}$  utilisant uniquement des portes NAND et donner la longueur du chemin critique correspondant.

## 4 Multiplexeur

Un multiplexeur est un circuit qui possède des entrées de donnée, des entrées de sélection, et une sortie.

La fonction réalisée par ce circuit est la suivante : la sortie recopie l'entrée de données dont le numéro est codée (en binaire) par les signaux reçus sur les entrées de sélection.

1. Si un multiplexeur possède  $n$  entrée de données, combien doit-il posséder d'entrée de sélection ?
2. Donner la table de vérité d'un multiplexeur possédant 2 entrées de données.
3. Donner une expression du calcul booléen pour à cette table et dessiner le circuit correspondant.
4. Donner une expression du calcul booléen pour un multiplexeur à 4 entrées de données.
5. Proposer un circuit démultiplexeur qui réalise la



fonction inverse.