

# Circuits et logique booléenne

## 1 Introduction

### 1.1 Tube à vide, transistor et circuit intégré

#### Exercice

Compléter le texte suivant :

À la base de la plupart des composants d'un ordinateur, on retrouve le ..... Ce composant électronique a été inventé fin .... par les Américains John ....., William ..... et Walter ..... L'invention du transistor a été un immense progrès, mais les premiers ordinateurs sont antérieurs à cette invention. En effet, ces premiers ordinateurs, par exemple le Colossus qui date de ...., étaient conçus à base de tubes électroniques (on parle aussi de tubes à vide) qui, bien que beaucoup plus gros et beaucoup moins fiables que les transistors, fonctionnent sur le même principe que ce dernier.

Autre aspect historique qu'il est important de préciser : on ne trouve plus, depuis quelque temps déjà, de transistors en tant que composant électronique discret. Dans un ordinateur, les transistors sont regroupés au sein de ce que l'on appelle des circuits ..... Dans un circuit ....., les transistors sont gravés sur des plaques de silicium, les connexions entre les millions de transistors qui composent un circuit ..... sont, elles aussi, gravées directement dans le silicium.

### 1.2 Circuits logiques

Le transistor est l'élément de base des circuits logiques. Un circuit logique permet de réaliser une opération booléenne. Ces opérations booléennes sont directement liées à l'algèbre de Boole. Un circuit logique prend en entrée un ou des signaux électriques (chaque entrée est dans un état « haut » (symbolisé par un « 1 ») ou à un état « bas » (symbolisé par un « 0 »)) et donne en sortie un ou des signaux électriques (chaque sortie est aussi dans un état « haut » ou à un état « bas »). Il existe deux catégories de circuits logiques :

- les circuits combinatoires (les états en sortie dépendent uniquement des états en entrée) ;
- les circuits séquentiels (les états en sortie dépendent des états en entrée ainsi que du temps et des états antérieurs).

Dans la suite nous nous intéresserons principalement aux circuits combinatoires.

## 2 Portes logiques

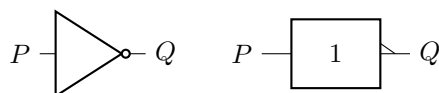
### 2.1 La porte NOT

Un transistor permet par exemple de réaliser une opération très simple appelée porte logique NON (NOT en anglais). Une porte logique est une fonction qui prend un ou plusieurs bits en entrée, et qui produit un bit en sortie.

La porte logique NON est une des portes les plus simples : elle n'a qu'un seul bit  $P$  en entrée et la sortie  $Q$  vaut 0 lorsque  $P$  vaut 1, et inversement la sortie vaut 1 lorsque  $P$  vaut 0. Pour représenter le calcul réalisé par une porte logique, on utilise une table logique. Celle de la porte NOT est donnée ci-dessous :

$P$	$Q$
0	1
1	0

Graphiquement, on représente la porte NOT comme dans le schéma ci-dessous, avec à gauche la notation américaine et à droite la notation européenne :



## Exercice

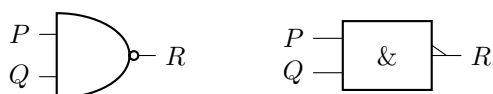
Compléter les tables logiques qui suivent.

### 2.2 La porte NAND

On peut fabriquer d'autres portes logiques en combinant plusieurs transistors, par exemple la porte NON ET (NAND en anglais) dont voici la table logique :

$P$	$Q$	$R$
0	0	..
0	1	..
1	0	..
1	1	..

Graphiquement, on représente la porte NAND comme dans le schéma ci-dessous (avec la notation américaine à gauche et la notation européenne à droite) :



### 2.3 La porte NOR

La porte NON OU (NOR en anglais) est représentée graphiquement ci-dessous :

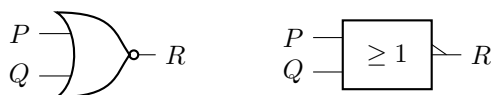


Table logique de la porte NOR

$P$	$Q$	$R$
0	0	..
0	1	..
1	0	..
1	1	..

### 2.4 La porte AND

La porte ET (AND en anglais) est représentée ci-dessous :

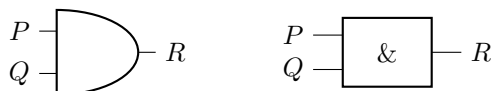


Table logique de la porte AND

$P$	$Q$	$R$
0	0	..
0	1	..
1	0	..
1	1	..

### 2.5 La porte OR

La porte OU (OR en anglais) est représentée ci-dessous :



Table logique de la porte OR

$P$	$Q$	$R$
0	0	..
0	1	..
1	0	..
1	1	..

### 2.6 La porte XOR

La porte OU EXCLUSIF (XOR en anglais) est représentée ci-dessous :

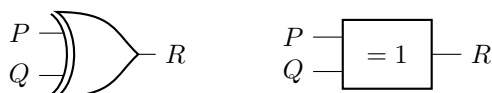


Table logique de la porte XOR

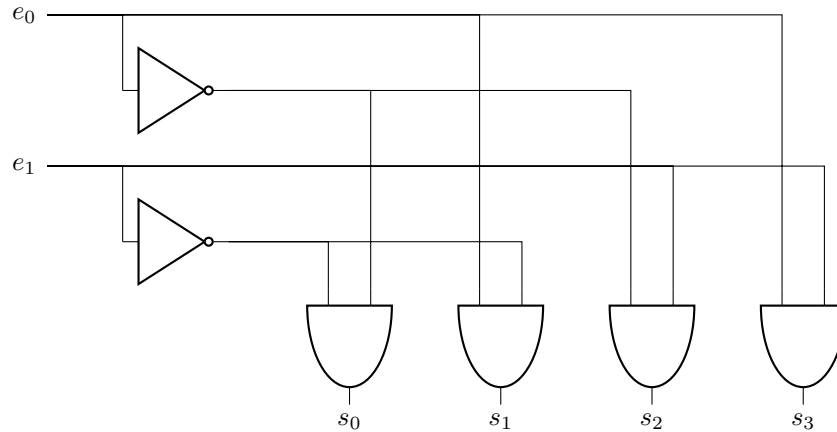
$P$	$Q$	$R$
0	0	..
0	1	..
1	0	..
1	1	..

### 3 Circuits combinatoires

En combinant plusieurs portes logiques, on obtient des circuits combinatoires qui peuvent posséder plusieurs entrées et plusieurs sorties.

#### 3.1 3.1 Décodeur $n$ bits

Le circuit suivant est appelé décodeur à 2 bits :



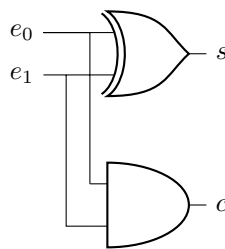
Dans le cas général, un décodeur  $n$  bits possède  $n$  entrées et  $2^n$  sorties. Les  $n$  bits en entrée sont utilisés pour mettre à 1 la sortie dont le numéro est égal au nombre codé (en base 2) par les entrées et mettre les autres sorties à 0.

#### Exercice

Compléter la table logique ci-dessous puis vérifier la remarque ci-dessus.

entrées		sorties			
$e_1$	$e_0$	$s_0$	$s_1$	$s_2$	$s_3$
0	0				

#### 3.2 3.2 Demi-additionneur 1 bit



Un demi-additionneur 1 bit prend en entrées deux bits  $e_0$  et  $e_1$  et il envoie sur une sortie  $s$  la somme  $e_0 + e_1$ .

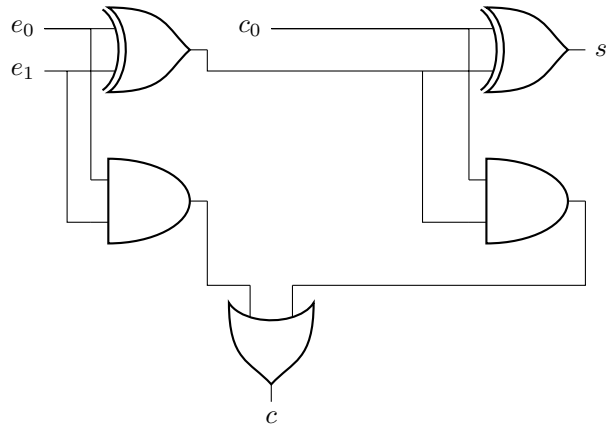
#### Exercice

Compléter la table logique ci-dessous :

entrées		sorties	
$e_0$	$e_1$	$s$	$c$
0	0		

Quelle est l'utilité de ce circuit ?

### 3.3 Additionneur 1 bit



#### Exercice

Compléter la table logique ci-dessous :

entrées			sorties	
$e_0$	$e_1$	$c_0$	$s$	$c$
0	0	0		

Quelle est l'utilité de ce circuit ?

## 4 À retenir

Les circuits d'un ordinateur sont fabriqués à partir de **portes logiques** élémentaires, elles-mêmes construites à partir de **transistors** où les valeurs booléennes 0 et 1 sont matérialisées par des courants électriques. Les portes logiques permettent de construire des **fonctions booléennes** arbitraires, telles que des décodeurs ou des additionneurs.