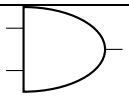
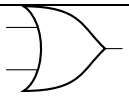
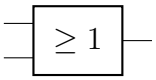
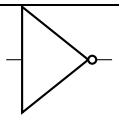
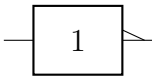
Chapitre 3

Circuits combinatoires

3.1 Portes logiques

Les portes logiques sont des composants électroniques permettant de réaliser physiquement les opérations de l'algèbre de Boole. Elles ont deux entrées et une sortie, chacune de celles-ci ont soit du courant (1 ou Vrai), soit n'en ont pas (0 ou Faux). Elles sont symbolisées comme ci-dessous.

Porte logique	Symbole américain	Symbole européen
ET		
OU		
NON		

Les portes logiques peuvent être fabriquées à l'aide de diodes, de lampes et de transistors. Ce sont ces derniers qui sont actuellement utilisés et les progrès technologiques font qu'ils ont tendance à approcher une taille moléculaire voire atomique.

3.2 Circuits combinatoires

Comme les fonctions booléennes peuvent être décomposées à l'aide des fonctions élémentaires ET, OU et NON, en combinant les trois portes logiques correspondantes, on peut réaliser physiquement des fonctions booléennes complexes. On appelle circuit combinatoire (ou circuit logique) un ensemble de portes logiques reliées entre elles pour répondre à une fonction booléenne.

3.3 Attendus et savoir-faire

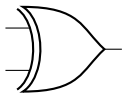
— Connaître les symboles des trois portes logiques élémentaires.

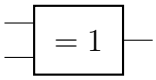
— Être capable traduire une fonction booléenne en un circuit combinatoire et réciproquement.

3.4 Exercices

Exercice 3.1. [La porte OU exclusif : XOR] En combinant les portes logiques ET, OU et NON, il est possible de fabriquer la porte XOR. Celle-ci est symbolisée comme suit.

Symbole américain	Symbole européen	Table de vérité

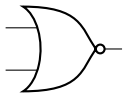
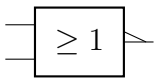
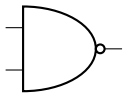




A	B	$A \oplus B$
0	0	
0	1	
1	0	
1	1	

1. Rappeler la décomposition à l'aide de ET, OU et NON de XOR puis compléter la table de vérité ci-dessus.
2. Dessiner le circuit combinatoire permettant de fabriquer une porte XOR à l'aide des portes ET, OU et NON.

Exercice 3.2. [Les portes NON-OU et NON-ET] Les portes NON-OU et NON-ET sont des portes universelles, chacune d'elles permet de reconstituer toutes les autres portes logiques, notamment ET, OU et NON ; elles sont donc à la base des composants des circuits logiques, avant même les portes précédemment citées. Elles sont symbolisées comme suit.

Porte	Symbole américain	Symbole européen	Table de vérité
NON-OU			
NON-ET			

A	B	$\neg(A \vee B)$
0	0	
0	1	
1	0	
1	1	

A	B	$\neg(A \wedge B)$
0	0	
0	1	
1	0	
1	1	

1. Compléter les tables de vérité ci-dessus.
2. Dessiner les circuits combinatoires permettant de fabriquer les portes NON, ET et OU à partir de la porte NON-OU. *Indication :* la porte NON-OU comporte deux entrées, il est possible de lui donner deux entrées identiques en dédoublant le câblage avant celle-ci.
3. Même question avec la porte NON-ET.

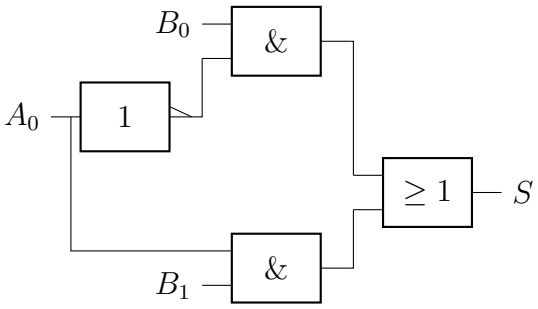
Exercice 3.3. [Le décodeur] Le décodeur est un circuit combinatoire à n entrées et 2^n sorties. Il sélectionne une sortie en fonction de la valeur des entrées. Voici comment sont sélectionnées les sorties en fonction des entrées pour un décodeur de taille 2 :

- $S_1 = (\neg A) \wedge (\neg B)$;
- $S_2 = (\neg A) \wedge B$;
- $S_3 = A \wedge (\neg B)$;
- $S_4 = A \wedge B$.

A	B	S_1	S_2	S_3	S_4
0	0				
0	1				
1	0				
1	1				

1. Compléter la table de vérité du décodeur ci-dessus.
2. Dessiner le circuit logique du décodeur de taille 2.
3. Mêmes questions avec un décodeur de taille 3.

Exercice 3.4. [Le multiplexeur] Le multiplexeur est en quelque sorte l'inverse d'un décodeur. Un multiplexeur k bits permet de sélectionner une entrée parmi 2^k disponibles. Il a $k + 2^k$ entrées et une seule sortie. Les k premières entrées A_1,...,A_k sont appelées bits d'adresses car elles donnent le numéro de l'entrée à sélectionner parmi les entrées B_1,...,B_{2^k}. La sortie S est alors égale à cette entrée sélectionnée. Le circuit ci-dessous représente un multiplexeur 1.



1. Donner la fonction booléenne associée à ce circuit et sa table de vérité.
2. Donner la fonction booléenne et le circuit logique d'un multiplexeur 2.
3. Que pensez-vous de l'affirmation « le multiplexeur contient un décodeur » ?

Exercice 3.5. [Le comparateur] Il s'agit un circuit servant à comparer deux mots $A_1A_2...A_n$ et $B_1B_2...B_n$ de n bits chacun. La sortie vaut 1 si les mots sont identiques et 0 sinon.

1. Dresser la table de vérité des mots A et B de 1 bit.
2. Tracer le circuit combinatoire d'un comparateur 1 bit.
3. En déduire le circuit combinatoire d'un comparateur 4 bits.

Exercice 3.6. [L'additionneur] Un additionneur est un circuit logique permettant de réaliser une addition. C'est une opération très courante dans un microprocesseur. Outre dans l'unité arithmétique, elle sert pour incrémenter le compteur de programme et pour les calculs d'adresses.

Dans l'exemple ci-dessous, on calcule en binaire 1010 + 0011 :

$$\begin{array}{r} 1010 \\ +\ 0011 \\ \hline 1101 \end{array}$$

Comme dans toutes les additions, il faut penser à utiliser des retenues. En effet, lorsqu'on a 1+1, le somme est 10 : le résultat (S) 0 et une retenue (R) 1. Voici un nouveau calcul avec les retenues indiquées :

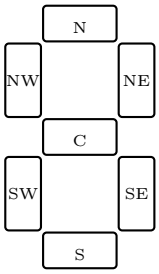
$$\begin{array}{rr} & 0110 \\ & +\ 0011 \\ \hline \text{retenues} & 11 \\ \hline & 1001 \end{array}$$

1. Compléter la table de vérité de l'additionneur 1 bit (les opérandes et le résultats sont sur 1 bit).

A	B	R	S

2. Vérifier que la fonction booléenne $((\neg A) \wedge B) \vee (A \wedge (\neg B))$ donne la même table de vérité que celle de S. De quelle fonction s'agit-il ?
3. Quelle est la fonction booléenne associée à R ?
4. En considérant que vous disposez de la porte logique XOR, tracer le circuit combinatoire d'un additionneur 1 bit.
5. En déduire le circuit combinatoire d'un additionneur 2 bits puis 4 bits.
6. En langage Python, écrire une fonction `add4b(a,b)` où `a` et `b` sont des nombres binaires codés sur 4 bits et qui renvoie la somme `a+b` ou « out of range » si la somme ne tient pas sur 4 bits.

Exercice 3.7. [L'afficheur numérique 7 segments] C'est un type d'afficheur très présent sur les calculatrices et les montres à affichage numérique : les chiffres s'écrivent en allumant ou en éteignant des segments, au nombre de sept. Quand les 7 segments sont allumés, on obtient le chiffre 8.



1. Compléter cette table de vérité pour chaque segment de l'afficheur.

Nombre	p	q	r	s	N	C	S	NW	SW	NE	SE
0	0	0	0	0							
1	0	0	0	1							
2											
3											
4											
5											
6											
7											
8											
9											

2. **[**]** Montrer que la fonction booléenne correspondant au segment SW :

$$(\neg s) \wedge ([(\neg q) \wedge (\neg r)] \vee [(\neg p) \wedge r]).$$

3. Tracer le circuit combinatoire correspondant au segment SW.