Activité 2 : Circuits combinatoires

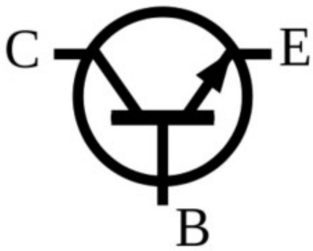
I. Portes logiques

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.
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.

1. Transistors

Les transistors, de l’anglais *transfer resistor* (résistance de transfert), que l’on trouve dans les circuits se comportent comme des interrupteurs. De nos jours, le transistor est tellement miniaturisé que votre ordinateur ou votre téléphone en contient plusieurs milliards sur une simple puce de silicium de la taille de votre ongle !

Symbole du transistor

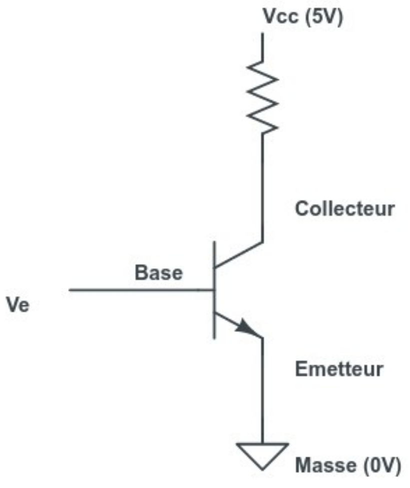


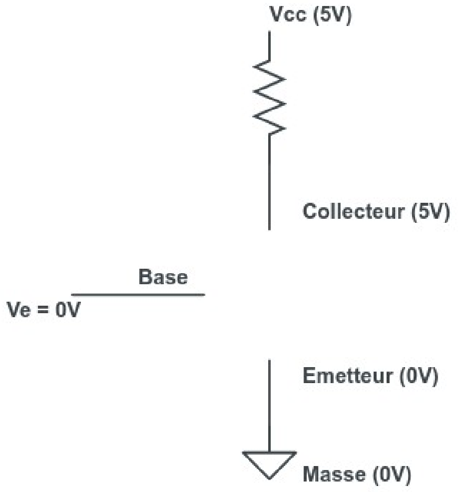
2. Fonctionnement

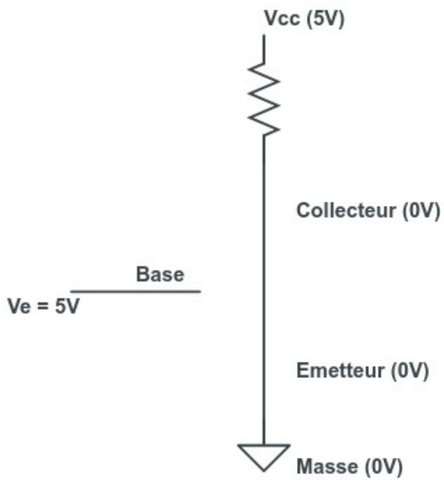
Le transistor (du type CMOS) comporte trois connexions externes :
— Le **collecteur (C)** est relié au fil d’où vient la tension Vcc (5 volts de courant continu) de l’alimentation.
— L’**émetteur (E)** est relié à la masse (0 volt).
— La **base (B)** constitue la connexion d’entrée Ve.

Tout dépend de la tension Ve qui lui est appliquée.

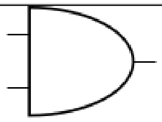
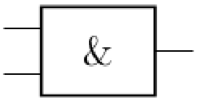
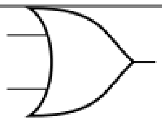
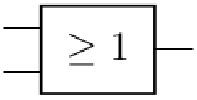
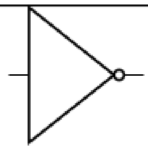
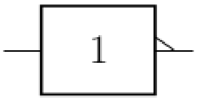
— Si l’on n’applique aucune tension à la base (Ve = 0), le transistor bloque le courant entre collecteur et émetteur.
— Si l’on met sur la base une tension de Ve = 5 volts en entrée, le courant passe entre le collecteur et l’émetteur.



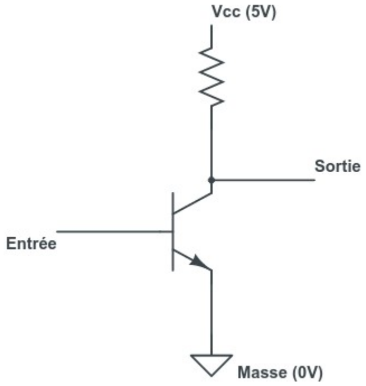




II. Portes logiques et fonctions associées

Porte logique	Symbole américain	Symbole européen
ET		
OU		
NON		

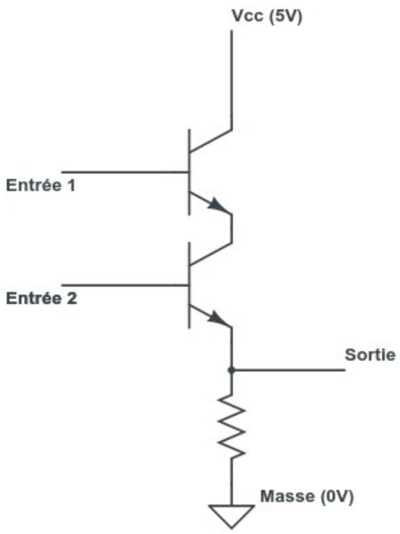
1. Porte ....



Symbole

Table de vérité

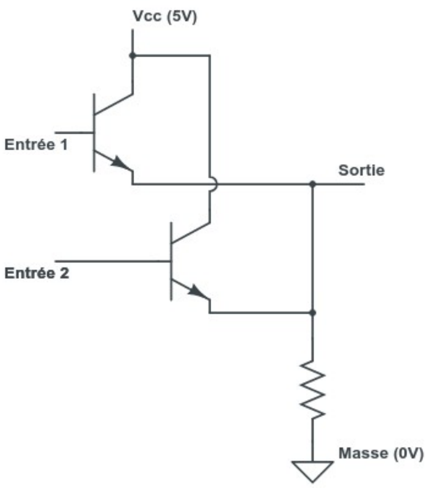
2. Porte ....



Symbole

Table de vérité

3. Porte ....



Symbole

Table de vérité

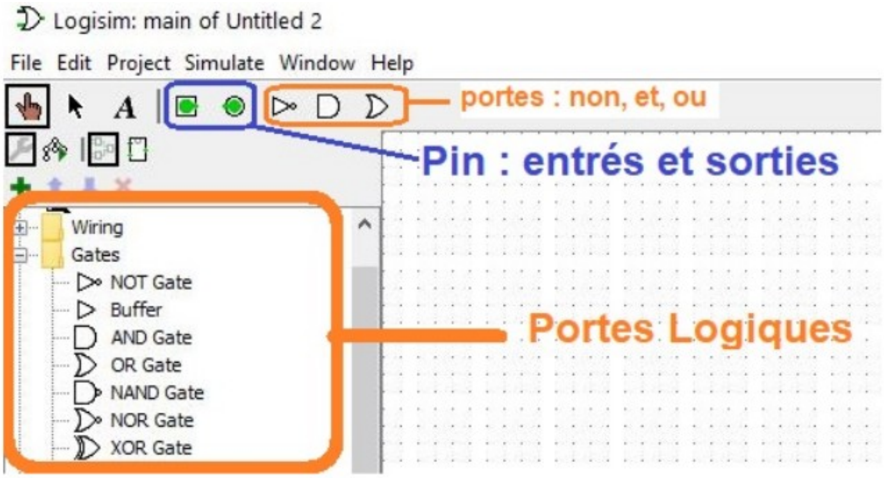
III. Circuits combinatoires

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.

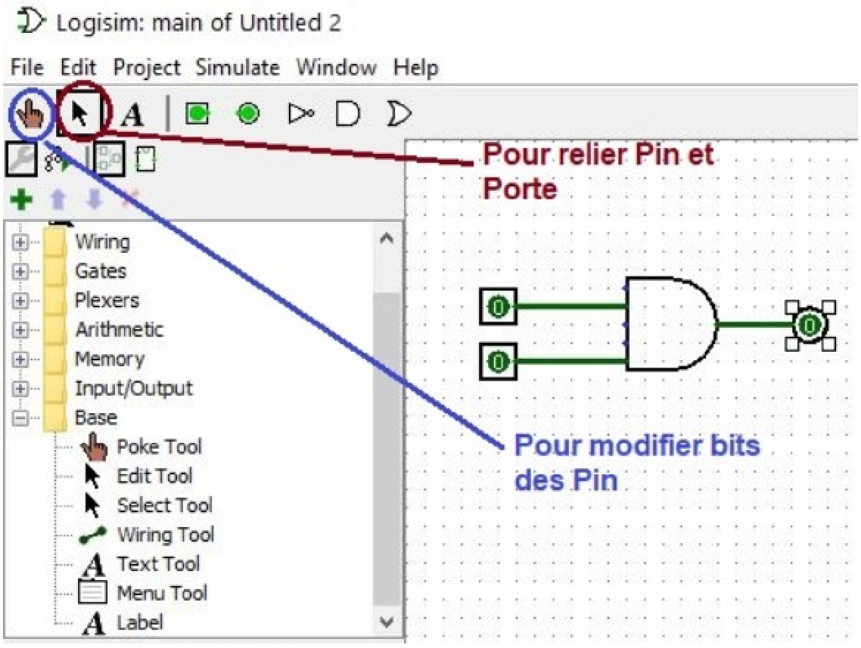
V. Les circuits logiques sur le logiciel Logisim

— Étape 1
Ouvrir le logiciel Logisim. C'est un simulateur booléen. Il ne cherche pas à faire intervenir des valeurs autres que 0 ou 1. Il s'agit d'un logiciel libre et gratuit.

— Étape 2
On découvre une feuille blanche sur laquelle on peut réaliser des circuits logiques ainsi qu'un menu qui propose plusieurs composants.
Dans le dossier Gates, trouvez une porte logique AND. Sélectionnez-la et dessinez-la sur la feuille blanche par simple clic.



— Étape 3
On utilise les "Pin" carrés verts pour les entrées et le "Pin" disque vert pour la sortie.
Relier les entrées et la sortie à la porte en cliquant sur la flèche.



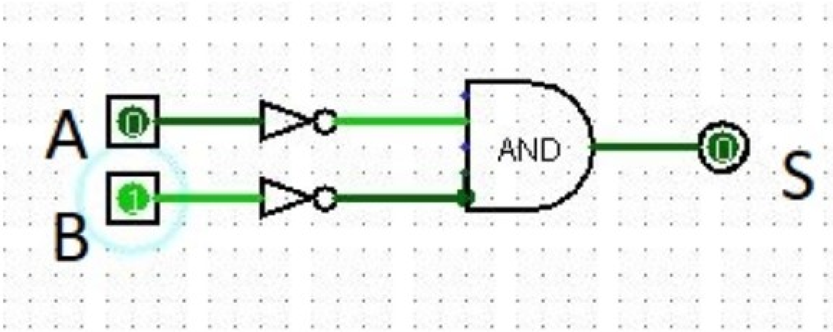
— Étape 4
On peut modifier les valeurs d'entrée en cliquant sur les "Pin" avec la "Main" du menu.
Sélectionner le mode Simulation (la main en haut à gauche). En cliquant sur les entrées, relever la valeur de sortie et vérifier que la table de vérité est correcte.
Aide : Menu Projet / Analyse circuit / Table ...

1. Pour bien débuter

Réaliser les circuits NON, ET et OU à l'aide du logiciel Logisim et vérifier les tables de vérité

Exercice 3.1 - Porte NOR

1. Réaliser le circuit suivant à l'aide du logiciel Logisim.



2. Donner l'expression booléenne de S en fonction des variables A et B.
3. Compléter la table de vérité ci-dessous.

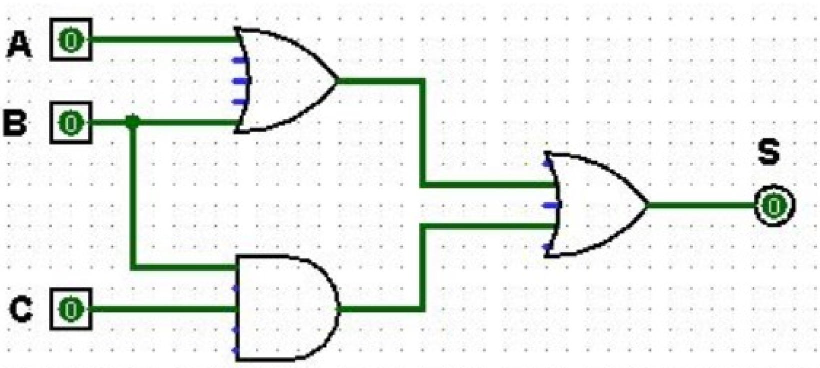
A	B	S
0	0	
0	1	
1	0	
1	1	

4. Par quel circuit comprenant seulement deux portes peut-on remplacer le circuit étudié ?
5. Ce circuit est en fait celui de la porte NOR. Vérifier cela en utilisant la porte NOR de logisim.

Porte NON-OU

Exercice 3.2

On considère le circuit logique ci-dessous.



1. Donner l'expression booléenne de S en fonction des variables A, B et C.
2. Compléter la table de vérité ci-dessous.

A	B	C	S
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

3. En déduire une formule pour S qui ne dépend que des variables A et B.

2. Porte XOR

Exercice 3.3 - Porte XOR : disjonction (OU) exclusive $\oplus$

La fonction logique OU eXclusif (XOR) est définie de la manière suivante :

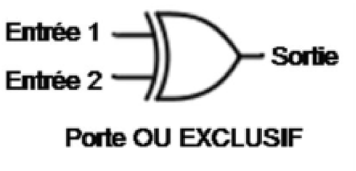
A `xor` B est Vrai

si et seulement si

A Vrai `or` B Vrai mais pas les deux.

La fonction booléenne de passage s'écrit :

$$\boxed{f(A,B) = A \bigoplus B = A \,\texttt{xor}\, B}$$



1. Écrire la table de vérité de la loi XOR.

Table de vérité : loi XOR

Entrées		Sortie
a	b	$a \bigoplus b = a \,\texttt{xor}\, b$
0	0	
0	1	
1	0	
1	1	

2. Écrire la table de vérité de la loi définie par :

$$f(A,B) = (\texttt{not}\ A\ \text{and}\ B)\ \texttt{or}\ (A\ \text{and}\ \texttt{not}\ B) = \left(\overline{A}\ \text{and}\ B\right)\ \texttt{or}\ \left(A\ \text{and}\ \overline{B}\right)$$

Table de vérité

Entrées		Sortie
A	B	$\left(\overline{A}\ \text{and}\ B\right)\ \texttt{or}\ \left(A\ \text{and}\ \overline{B}\right)$
0	0	
0	1	
1	0	
1	1	

3. Concevoir un circuit qui, étant données deux entrées A et B, donne en sortie la valeur $A \text{ xor } B$, en utilisant seulement des portes Non, Et, Ou.

Aide

La question précédente a montré que :

$$A \text{ xor } B = (\text{not } A \text{ and } B) \text{ or } (A \text{ and not } B) = A \oplus B = \left(\overline{A} \text{ and } B\right) \text{ or } \left(A \text{ and } \overline{B}\right)$$

4. Réaliser le circuit proposé à l'aide du logiciel Logisim puis directement avec la porte XOR.

Remarque

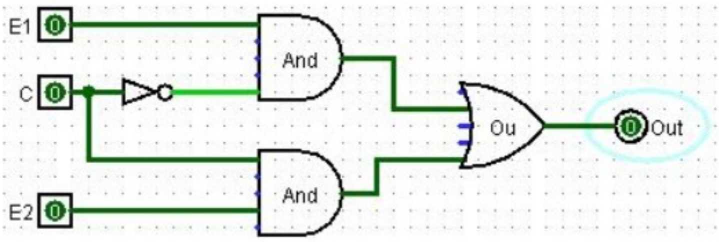
Application en électricité domestique.
Une application utilisée de l'opérateur logique XOR en électricité domestique est dans les salles où une ampoule peut être allumée ou éteinte par deux interrupteurs placés près de deux entrées. Chacun des deux interrupteurs peut soit allumer ou éteindre l'ampoule quelle que soit la position de l'autre interrupteur. Pour obtenir une telle fonctionnalité, on doit brancher les deux interrupteurs afin de former un opérateur XOR. C'est le montage dit « va-et-vient ».

3. Circuit multiplexeur

Un multiplexeur (abréviation : MUX) est un circuit permettant de concentrer sur une même voie de transmission différents types de liaisons (informatique, télécopie, téléphonie, télétex) en sélectionnant une entrée parmi N . Il possédera donc une sortie et N entrées, ainsi qu'une entrée de commande de plusieurs bits permettant de choisir quelle entrée sera sélectionnée. Il sert d'accès aux réseaux de transmission de données numériques ou analogiques.

Exercice 3.4 - Circuit MUX-2

On considère le circuit logique suivant.



1. Donner l'expression de Out en fonction de E_1, E_2 et C.
2. Compléter le tableau de vérité de ce circuit.

C	E_1	E_2	Out
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Remarque

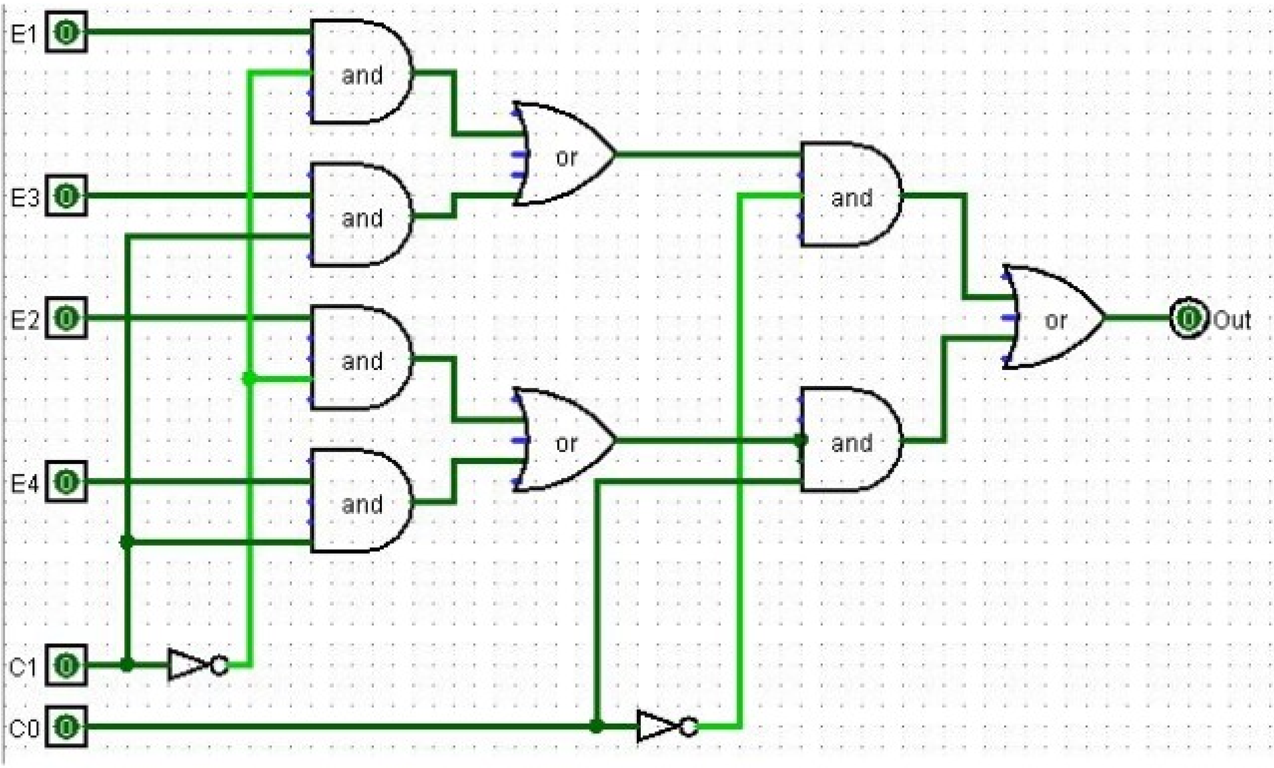
Le circuit étudié est appelé **multiplexeur à 2 entrées**.
Selon la valeur de la commande (C), il permet de reproduire en sortie (Out) :

— le signal E_1 si C est à 0.

— le signal E_2 si C est à 1.

Exercice 3.5 - Circuit MUX-4 (difficile *)

On considère un multiplexeur à 4 entrées, dont le circuit est représenté ci-dessous.



Aide

Pour cet exercice, vous pourrez vous aider du logiciel logisim qui vous fournit directement la table de vérité du circuit.
Menu Project / Analyse Circuit / Table (et imput pour régler l'ordre des variables).

1. Par analyse du circuit, déterminer l'expression booléenne de Out en fonction des entrées E_1, E_2, E_3, E_4 et des commandes C_0 et C_1.
On peut par exemple sur logisim vérifier quelle est la l'entrée reproduite en sortie selon les valeurs de C_0 et C_1.
2. Quelles sont les valeurs des commandes C_0 et C_1 qui permettent de sélectionner en sortie (Out) :
   — l'entrée E_1 ?
   — l'entrée E_2 ?
   — l'entrée E_3 ?
   — l'entrée E_4 ?

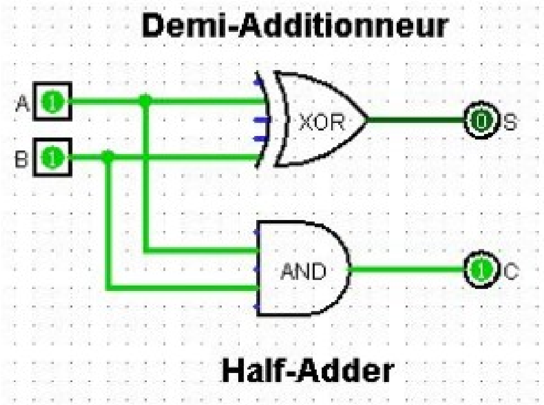
Remarque

Le circuit étudié est appelé **multiplexeur à 4 entrées**.
Selon la valeur des commandes C_0 et C_1, il permet de reproduire en sortie (Out) le signal E_1, E_2, E_3 ou E_4. Voir la dernière question de l'exercice.

4. Les additionneurs

Le circuit étudié, appelé demi-additionneur, permet d’additionner deux bits A et B.
Il comporte deux sorties C et S qui représentent deux expressions booléennes.

Exercice 3.6



1. Donner les expressions booléennes de C et S en fonction de A et B.
2. Compléter la table de vérité de C et S.

Table de vérité : S

Entrées		Sortie
a	b	S
0	0	
0	1	
1	0	
1	1	

Table de vérité de C

Entrées		Sortie
a	b	C
0	0	
0	1	
1	0	
1	1	

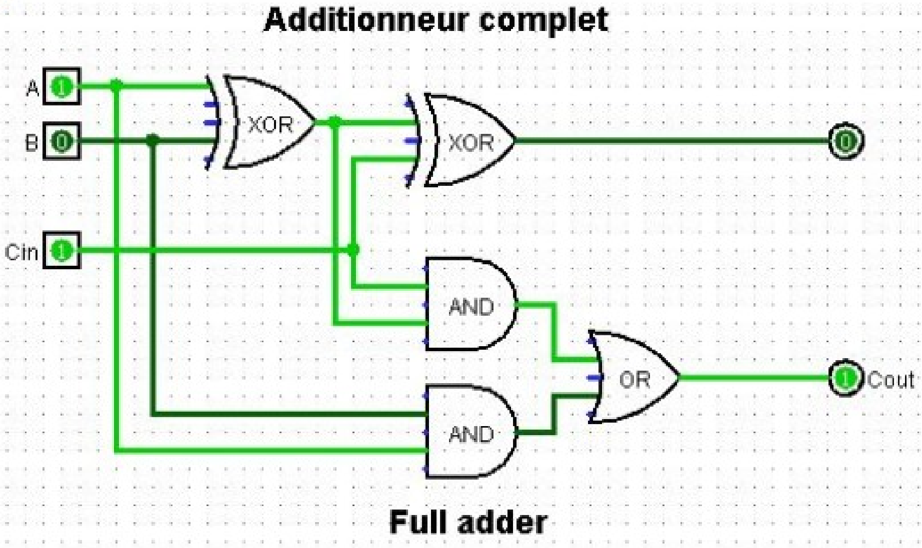
3. Quel est le rôle des sorties C et S dans la fonction du circuit ?

Remarque

Le choix de la lettre C vient du fait qu’en anglais, " **retenue** " se dit "**carry**" .

En pratique, une addition binaire est une suite d'additions sur 1 bit. Néanmoins, il faut connaître la retenue pour enchaîner ces additions. On réalise alors le circuit ci-dessous, appelé additionneur complet.
Il comporte deux sorties C_{out} , S et trois entrées, le bit A, le bit B et la retenue précédente C_{in}

Exercice 3.7

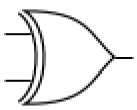
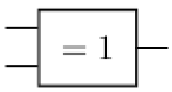


1. Réaliser ce circuit à l'aide du logiciel Logisim.
2. Compléter la table de vérité ci-dessous.

C_{in}	A	B	C_{out}	S
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

V. Exercices

Exercice 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.

<table>
<tr><th>Symbole américain</th><th>Symbole européen</th><th colspan="3">Table de vérité</th></tr>
<tr><td rowspan="5"></td><td rowspan="5"></td><td>A</td><td>B</td><td>$A \oplus B$</td></tr>
<tr><td>0</td><td>0</td><td></td></tr>
<tr><td>0</td><td>1</td><td></td></tr>
<tr><td>1</td><td>0</td><td></td></tr>
<tr><td>1</td><td>1</td><td></td></tr>
</table>

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.

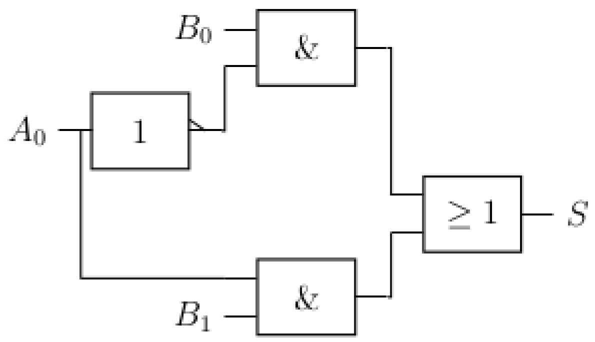
Exercice 2. [Le décodeur] Le décodeur est un circuit combinatoire à n entrées et 2^n sorties.
Il sélectionne une sortie en fonction de la valeur des entrées. Voici comment sont sélectionnées les sorties en fonction des entrées pour un décodeur de taille 2 :

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

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. [Le multiplexeur] Le multiplexeur est en quelque sorte l'inverse d'un décodeur.
Un multiplexeur k bits permet de sélectionner une entrée parmi 2k 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 4. [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 :

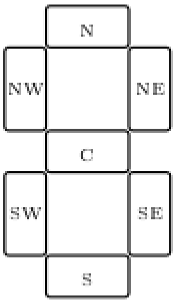
$$\begin{array}{lr} & 0110 \\ & +\ 0011 \\ \hline \text{retenues} & 11 \\ \hline & 1001 \end{array}$$

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

A	B	R	S

Exercice 5. [Le comparateur] Il s'agit un circuit servant à comparer deux mots $A_1 A_2 \ldots A_n$ et $B_1 B_2 \ldots B_n$ 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 6. [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 est :

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

3. Tracer le circuit combinatoire correspondant au segment SW.