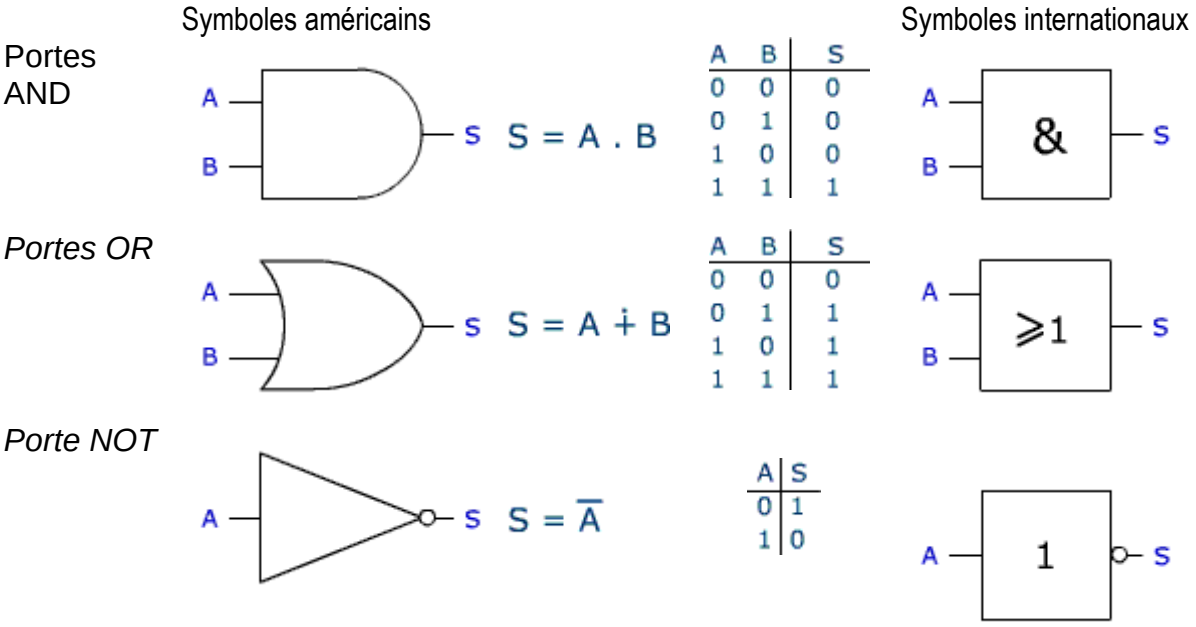
Les portes logiques

Nous avons jusqu'ici utilisé des boutons poussoirs et une lampe pour illustrer le fonctionnement des opérateurs logiques. En électronique digitale, les opérations logiques sont effectuées par des portes logiques. Ce sont des circuits qui combinent les signaux logiques présentés à leurs entrées sous forme de tensions. On aura par exemple 5V pour représenter l'état logique 1 et 0V pour représenter l'état 0.

Voici les symboles des trois fonctions de base.

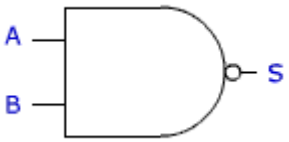


Le nombre d'entrées des fonctions AND et OR n'est pas limité. Voici par exemple une représentation de ces portes avec trois entrées :

Combinaisons de portes logiques.

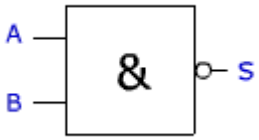
Ces trois fonctions logiques de base peuvent être combinées pour réaliser des opérations plus élaborées en interconnectant les entrées et les sorties des portes logiques.

La porte NAND (Non ET)



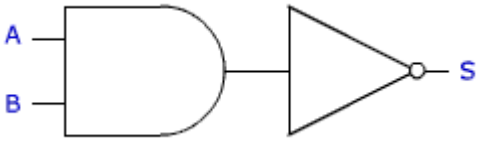
$S = \overline{A \cdot B}$

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

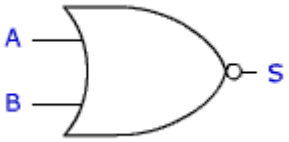


(Symbole américain) (Symbole international)

=NOT AND

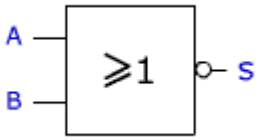


Porte NOR (Non OU)



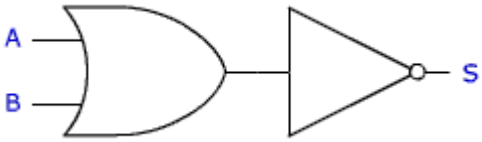
$S = \overline{A + B}$

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



(Symbole américain) (Symbole international)

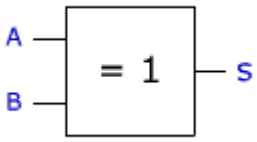
= NOT OR



Porte XOR

$S = A \oplus B$

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



(Symbole américain) (Symbole international)

Porte XOR à deux entrées

La fonction "OU Exclusif" est en principe d'une fonction de deux variables :

$$S = A \oplus B$$

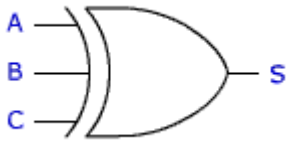
La sortie est à 1 si une seule des deux entrées vaut 1, d'où son appellation « Ou exclusif ».

Porte XOR à plusieurs entrées

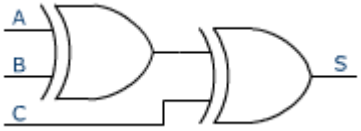
Pour calculer le résultat de $S = A \oplus B \oplus C$, on doit pouvoir faire d'abord l'opération entre deux termes, puis refaire un ou exclusif entre le résultat obtenu et le troisième terme.

Ce qui se traduit par $S = (A \oplus B) \oplus C$ ou par $S = A \oplus (B \oplus C)$

On constate que l'appellation "Ou exclusif" n'est plus aussi ben à propos puisque avec trois variables, le résultat vaut 1 si une seule entrée ou toutes les trois valent 1.



≡



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

Le résultat est en fin de compte un bit de parité. Il vaut 1 si le nombre d'entrées à 1 est impair.

L'inverse de la porte XOR à 2 entrées

Voyons ce que donne la table de vérité si on inverse la sortie d'une porte XOR :

A	B	$S = \overline{A \oplus B}$
0	0	1
0	1	0
1	0	0
1	1	1

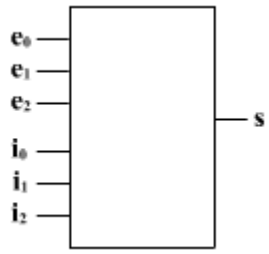
Le résultat est vaut 1 si les deux entrées sont identiques.

Cette porte teste donc l'équivalence des deux entrées. Certains appellent cette fonction logique, "fonction équivalence", d'autres l'appelle "XNOR"

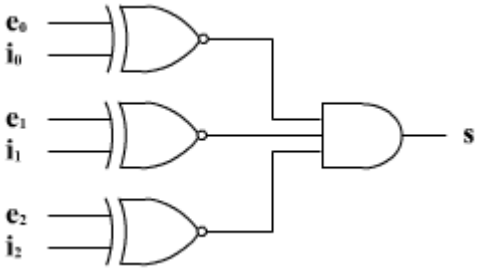
Circuits logiques qui jouent un rôle important dans le hardware

Comparateur

Le comparateur est un circuit qui compare deux mots de n bits. En sortie, un bit indique le résultat de la comparaison : 1 s'il y a égalité entre les deux codes présents à l'entrée, 0 si ces codes sont différents.



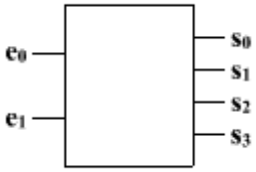
S = 1 si
$e_1=i_1$
et $e_2=i_2$
et $e_3=i_3$



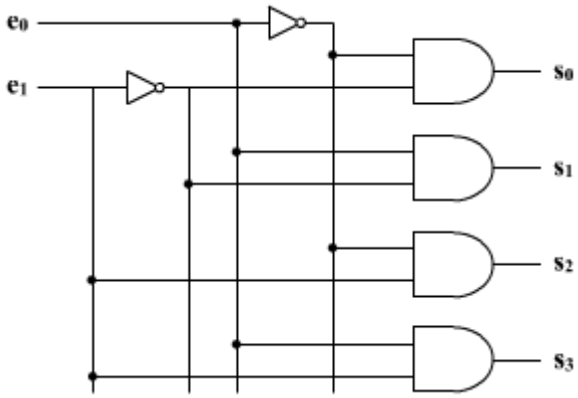
Décodeur

Le décodeur est un circuit qui possède n bits à d'entrées et au plus 2^n bits en sortie. Parmi toutes ces sorties une seule est active, son numéro est formé par les n bits en entrée.

Exemple : Décodeur "1 parmi 4"



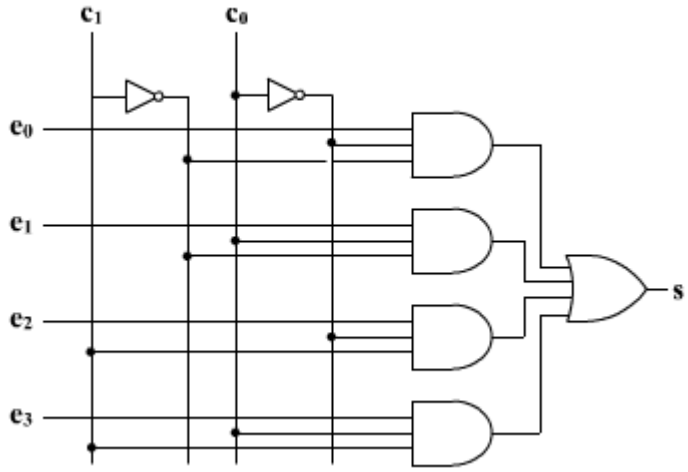
e_1	e_0	s_3	s_2	s_1	s_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0



Multiplexeur

Le multiplexage est une opération qui consiste à utiliser un équipement unique pour traiter plusieurs signaux. Exemple : une ligne de transmission pour transmettre plusieurs signaux. On parle alors de multiplexage temporel : De mêmes intervalles de temps sont accordés successivement pour chacun des signaux à transmettre.

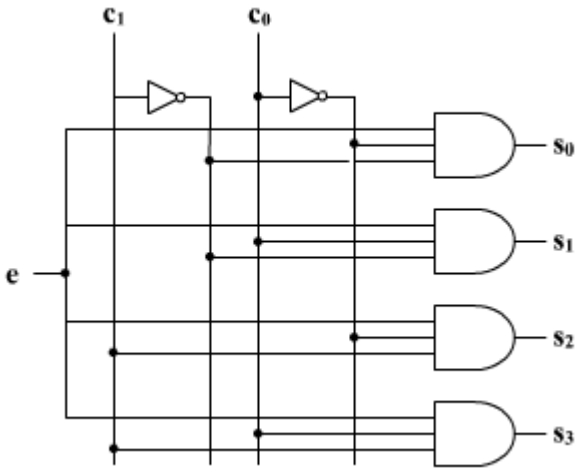
C_1	C_0	S
0	0	e_0
0	1	e_1
1	0	e_2
1	1	e_3



Le multiplexeur agit comme un "commutateur" qui transmet à la sortie le signal d'une entrée sélectionnée par un code binaire.

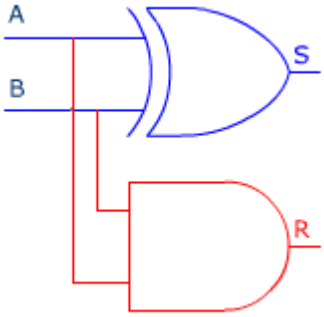
Démultiplexeur

Ce circuit réalise la fonction inverse du multiplexeur. Il possède plusieurs sorties (2^n) , un signal en entrée et n bits pour désigner la sortie vers laquelle sera aiguillé le signal d'entrée.



C_1	C_0	S_3	S_2	S_1	S_0
0	0	0	0	0	e
0	1	0	0	e	0
1	0	0	e	0	0
1	1	e	0	0	0

Le demi additionneur *half adder*→ circuit à 2 entrées : 1 bit + 1 bit



$S = A \oplus B$
est la somme

$R = A \cdot B$
est le report

A	B	S	R
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Le demi additionneur effectue la somme de deux bits. S est la somme et R le report. (*carry*)
Ce schéma ne convient cependant que pour additionner 2 nombres de 1 bit.

```
  0      1      0      1
 +0     +0     +1     +1
 --     --     --     --
 00     01     01     10
```

Le plein additionneur *full adder*

Pour additionner de deux nombres de plusieurs bits il faut mettre en cascade des additionneurs qui additionnent les bits correspondant des deux nombres plus les reports R_{i-1} issus des additions des bits précédents.

Exemple : Calculons 1 + 3
En binaire cela donne : 0001 + 0011 →

```
    0 1 1
  0 0 0 1
+ 0 0 1 1
  -------
  0 1 0 0
```

L'addition des bits de droite est une addition de deux bits,
elle peut être réalisée avec le demi additionneur
Il faut tenir compte d'un éventuel report pour les bits suivants.
Ainsi dès le deuxième bit de notre exemple (en comptant les bits de droite à gauche) il a fallu faire 2 additions (`1 + 0 + 1 = 10` ⇒ on pose `0` et on reporte `1`)

Table de vérité du circuit plein additionneur

Cette table de vérité comporte 3 entrées : R_{n-1} (le report de l'addition précédente), A et B

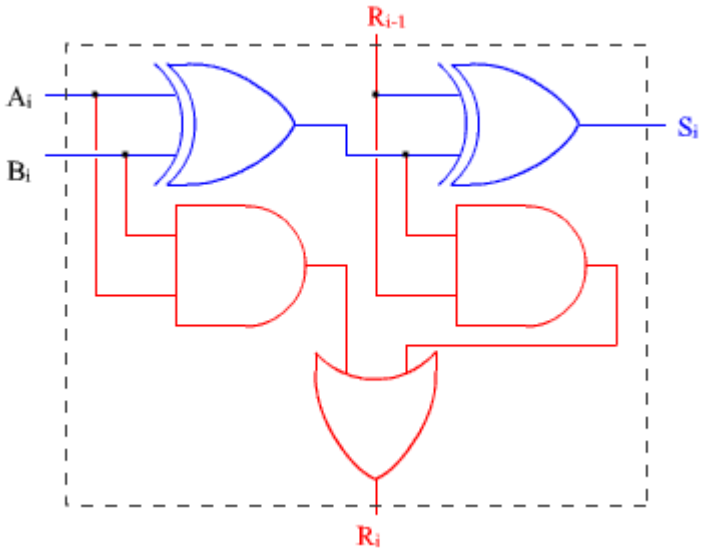
Il lui faut deux sorties : S = la somme de 3 bits $(A + B + R_{n-1})$
R = le nouveau report

R_{i-1}	A	B	S	R
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Equations du circuit

$$S_i = A_i \oplus B_i \oplus R_{i-1}$$
$$R_i = (A_i \cdot B_i) + R_{i-1} \cdot (A_i \oplus B_i)$$

Schéma du circuit plein additionneur



$$S_i = A_i \oplus B_i \oplus R_{i-1}$$
$$R_i = (A_i \,.\, B_i) + R_{i-1} \,.\, (A_i \oplus B_i)$$

Le plein additionneur est un circuit à 3 entrées. Il se compose de 2 demi additionneurs et d'une porte OU qui génère le report quand la somme vaut 2 ou 3

Addition de deux nombres de n bits

Exemple : Mise en cascade de 4 additionneurs pour l'addition de deux nombres de 4 bits
Le circuit peut tenir compte de l'éventuel report précédent R_{-1}
Le report $R_3 = 1$ dès que l'écriture de la somme nécessite plus de 4 bits