

2.5

Représentation des booléens

NSI 1ÈRE - JB DUTHOIT

Histoire

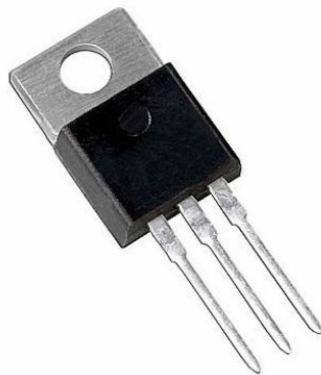
Georges Boole sort ' Mathematical Analysis of Logic' en 1847, qui est une structure algébrique à seulement deux états : 0 et 1. À cette époque, tout cela reste théorique mais sera grandement utilisé par la suite. Bien que le Binaire soit apparu en 3000 avant J.C. en Chine pour calculer des périodes religieuses, les Français ne l'utiliseront qu'en 1600 avec la table de Harriot (mathématicien et astronome anglais) et Leibniz y consacra un article important en 1703. Il fallut attendre 1930 pour que Claude Shannon démontre qu'avec des interrupteurs fermés pour 'vrai' et ouvert pour 'faux', on peut effectuer des opérations logiques en associant le nombre 1 pour vrai et 0 pour faux. C'est lui qui va populariser le mot 'bit' créé par John Tukey.

En 1937, Georges Stibitz et Samuel Williams fabriquent un calculateur avec des relais, basé sur le système binaire, capable de réaliser les 4 opérations avec des nombres décimaux en entrée en 1 minute ! Ils ont même établi un système de communication pour piloter une machine à des centaines de kilomètres. D'autres calculateurs avaient été fabriqués bien avant mais étaient mécaniques et utilisaient des engrenages. La Pascaline de Blaise Pascal, à 16 ans, est capable de soustraire et additionner. Gottfried Wilhelm Leibniz conçoit en 1694 une machine capable d'en plus multiplier et diviser. D'autres modèles mécaniques suivirent, le dernier fut une calculatrice mécanique de 1948 capable de faire une multitude de calculs mathématiques rapidement. Elle sera utilisée jusqu'en 1970, année de mise en marché de la calculatrice dite conventionnelle que l'on connaît aujourd'hui.

C'est l'arrivée du transistor en 1947 qui va bouleverser les travaux sur les calculateurs. Jusqu'alors on utilisait les relais ou les lampes. Les lampes étaient 1000 fois plus grandes et consommaient 1000 fois plus.

2.5.1 Qu'est ce qu'un transistor ?

Que fait un transistor ? Un transistor a trois broches : le collecteur, la base et l'émetteur. Le courant dans l'émetteur est égal à la somme des courants du collecteur et de la base. Si la base n'est pas alimentée, le courant ne passe pas et une faible tension dans la base laissera passer le courant... passe, passe pas,...1,0..



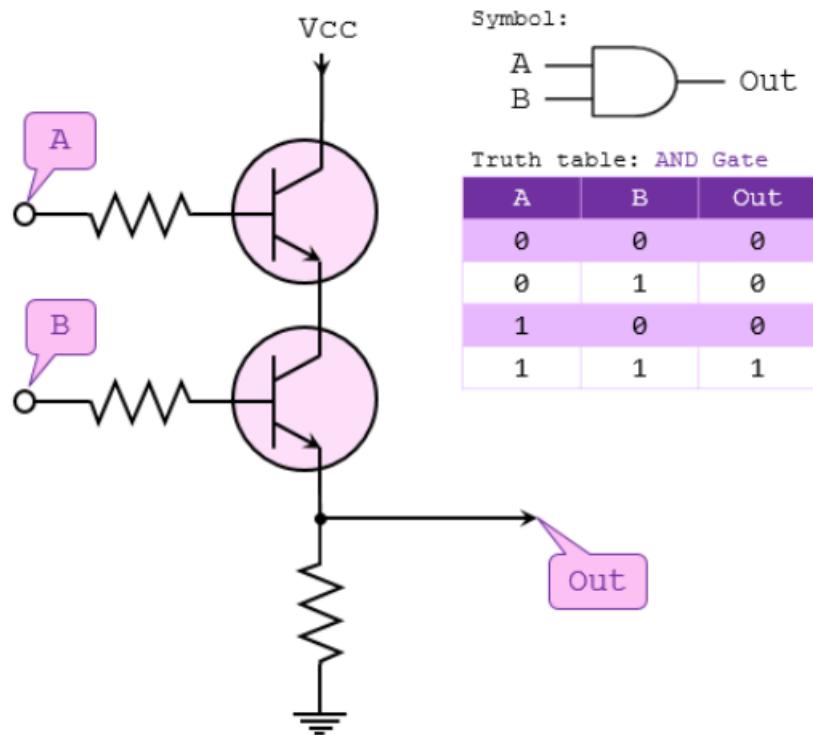
Transistor

2.5.2 Fonctionnement

On a un comportement très particulier :

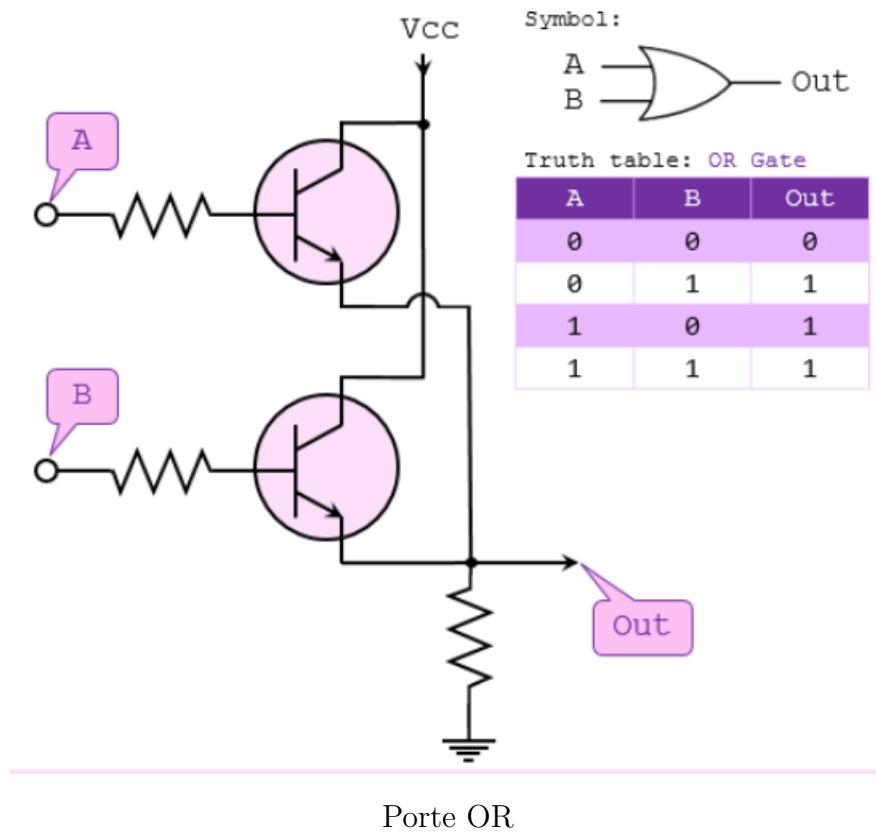
- en appliquant une petite tension à la base, le courant principal entre le collecteur et l'émetteur peut passer
- en supprimant cette tension à la base, le courant entre le collecteur et l'émetteur est coupé.

2.5.3 Porte AND

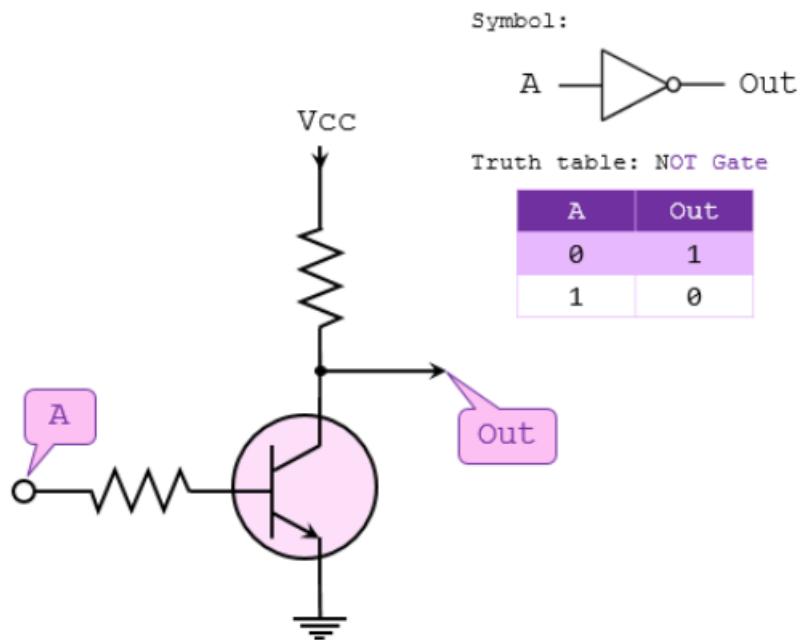


Porte AND

2.5.4 Porte OR

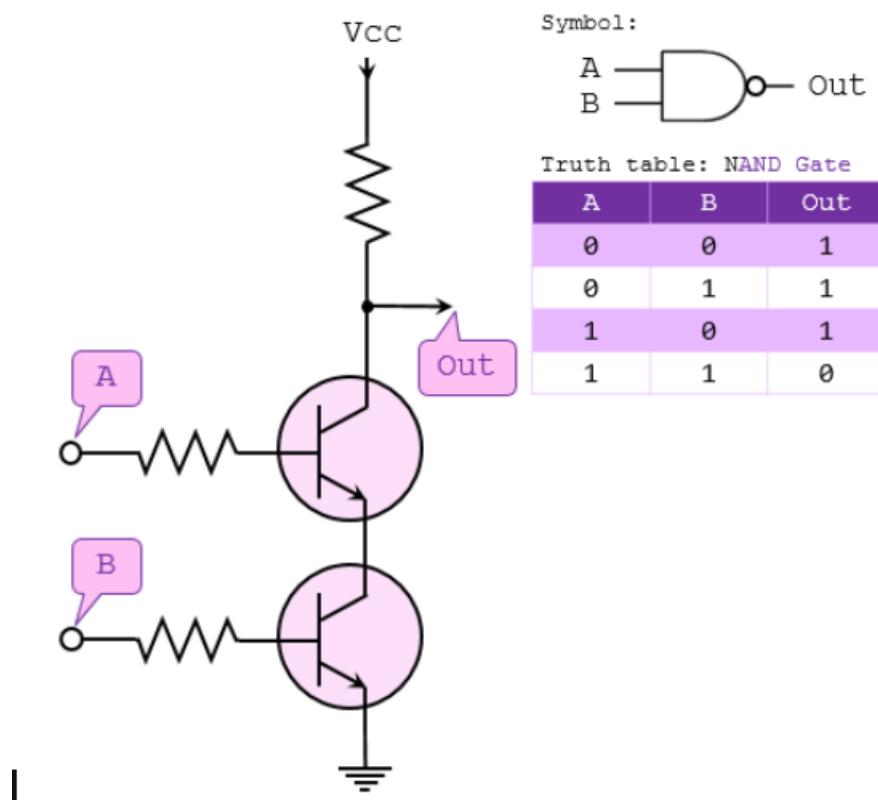


2.5.5 Porte NOT



Porte OR

2.5.6 Porte NOT AND



Porte NOT AND

2.5.7 Porte XOR

A	B	Out
0	0	0
0	1	1
1	0	1
1	1	0

2.5.8 Loi de Moore

La multiplication du nombre de transistors a suivi loi de Moore : "le nombre de transistors par circuit de même taille allait doubler, à prix constant, tous les ans. Il rectifia par la suite en portant à dix-huit mois le rythme de doublement. Il en déduisit que la puissance des ordinateurs allait croître de manière exponentielle, et ce pour des années." Depuis 2014 la loi de Moore n'est plus complètement vérifiée et subit un ralentissement.

Exercice 2.8

Faire un tableau à 5 lignes. Placer a et b dans les deux premières colonnes et y inscrire toutes les combinaisons possibles de 0 et 1 pour a et b dans les 4 lignes qui suivent. Ajouter des colonnes pour écrire not(a) and b, not(a) or b, not(a) or not(b), not(a and b). Comparer les deux dernières colonnes. Que concluez vous ?

Exercice 2.9

Faire de même avec (a or b) and (a and b), et (a or b) or (a and b). Comment peut-on écrire ces deux règles plus simplement ?

Exercice 2.10

Soit, liste1=[3,5,7] et liste2=[10,3,18,25].

Donner les valeurs de test1, test2 et test3 et test 4, avec :

test1= 3 in liste1 and 3 in liste2

test2= 10 in liste1 or 10 in liste2

test3= 10 in liste1 and 10 in liste2

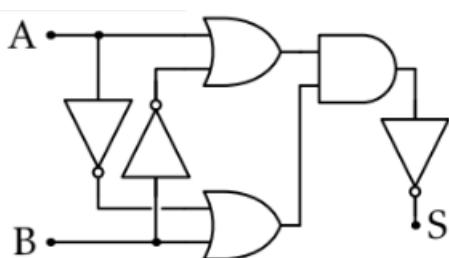
test4= 20 in liste1 and 20 in liste2

Exercice 2.11

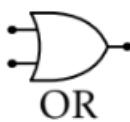
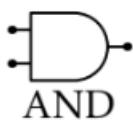
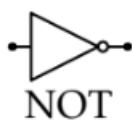
Faire un tableau pour (not(a) and b) or (a and not(b))

Exercice 2.12

Déterminer la table de vérité du circuit logique ci-dessous :

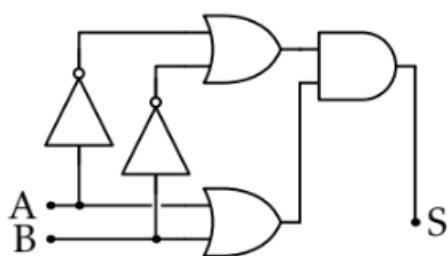


avec :

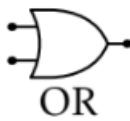
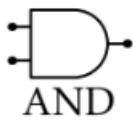
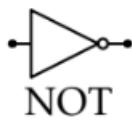


Exercice 2.13

Déterminer la table de vérité du circuit logique ci-dessous :



avec :



Exercice 2.14

En python, le "AND" se note "&", le "OR" se note "|", le "XOR" se note "^".

Réaliser ce genre de fonctions :

```
>>> tableET()
```

```
0 & 0 = 0
0 & 1 = 0
1 & 0 = 0
1 & 1 = 1
```

Table AND

Exercice 2.15

Cet exercice montre une des applications : le filtrage.

adresse IP :

11000000.10101000.00000001.00000010

masque de sous réseau :

11111111.11111111.11111111.00000000

L'adresse du réseau est obtenu en utilisant l'opérateur & pour chaque bit :

adresse du réseau :

11000000.10101000.00000001.00000000

Retrouver ce résultat avec Python, en utilisant les opérateur & et |.

```
Ob1100000010101000000000100000010
Ob1111111111111111111111111100000000
Ob10000000101010000000000100000000
```

Le résultat attendu

Exercice 2.16

Réaliser une fonction jeu() qui :

- Pose à l'utilisateur une question relative à tableau logique parmi AND, OR et XOR.
- L'utilisateur saisit sa réponse
- Un message est renvoyé pour dire à l'utilisateur si la réponse est correcte.

```
>>> jeu()
1 & 0 =
entrer votre réponse :0
bravo

>>> jeu()
0 ^ 1 =
entrer votre réponse :1
bravo

>>> jeu()
0 | 1 =
entrer votre réponse :1
bravo
```

Exemple

Exercice 2.17

Démontrer les lois de Morgan :

NON (a OR b) = (NON a) AND (NON b)

NON (a AND b) = (NON a) OR (NON b)

```
>>> not(True or False) == (not True) and (not False)
True
>>> not(1 or 0) == (not 1) and (not 0)
True
```

Loi de morgan en Python

Exercice 2.18

On reprend ici la fonction jeu() des exercices précédents.

Il s'agit ici de créer une partie composée de 10 questions. Après chaque réponse de l'utilisateur, un compteur indique le nombre de bonne réponse sur le total de questions répondues.

A la fin de la partie, on donne le score total.

```
>>> partie_simple()
1 ^ 1 =
entrer votre réponse :0
Bravo ! Votre score provisoire est 1 / 1
1 ^ 1 =
entrer votre réponse :0
Bravo ! Votre score provisoire est 2 / 2
1 & 0 =
entrer votre réponse :1
Perdu! Votre score provisoire est 2 / 3
1 | 0 =
entrer votre réponse :1
Bravo ! Votre score provisoire est 3 / 4
0 | 1 =
entrer votre réponse :1
Bravo ! Votre score provisoire est 4 / 5
0 | 0 =
entrer votre réponse :1
Perdu! Votre score provisoire est 4 / 6
0 ^ 1 =
entrer votre réponse :1
Bravo ! Votre score provisoire est 5 / 7
1 | 1 =
entrer votre réponse :1
Bravo ! Votre score provisoire est 6 / 8
1 | 0 =
entrer votre réponse :1
Bravo ! Votre score provisoire est 7 / 9
1 ^ 0 =
entrer votre réponse :1
Bravo ! Votre score provisoire est 8 / 10
jeu terminé ! Vous avez 8 /10
```

Exercice 2.19

On reprend ici la fonction jeu() des exercices précédents.

Il s'agit ici de créer une partie :

- Une partie est composée de 10 questions.
- Le joueur dispose de 30 secondes pour répondre aux questions.
- Pour chaque réponse, on affiche si c'est correct ou non, on affiche le score intermédiaire et le temps restant.
- La partie se termine :
 - Si le joueur a terminé les 10 questions. On affiche donc son score total sur 10.
 - Si le joueur dépasse les 30 secondes, la partie s'arrête. (Ne pas oublier de gérer le laps de temps entre la question posée et la réponse apportée). On affiche le score total sur 10.

```

>>> partie()
1 ^ 1 =
entrer votre réponse :0
Gagné ! score : 1 /10
Temps restant 28.3 secondes
0 ^ 1 =
entrer votre réponse :1
Gagné ! score : 2 /10
Temps restant 26.9 secondes
1 ^ 0 =
entrer votre réponse :1
Gagné ! score : 3 /10
Temps restant 24.7 secondes
1 ^ 0 =
entrer votre réponse :1
Gagné ! score : 4 /10
Temps restant 23.5 secondes
1 & 0 =
entrer votre réponse :0
Gagné ! score : 5 /10
Temps restant 21.6 secondes
1 | 1 =
entrer votre réponse :1
Gagné ! score : 6 /10
Temps restant 20.1 secondes
0 & 1 =
entrer votre réponse :0
Gagné ! score : 7 /10
Temps restant 18.5 secondes
1 & 0 =
entrer votre réponse :0
Gagné ! score : 8 /10
Temps restant 16.6 secondes
1 ^ 0 =
entrer votre réponse :1
Gagné ! score : 9 /10
Temps restant 15.2 secondes
0 & 0 =
entrer votre réponse :0
Gagné ! score : 10 /10
Temps restant 13.7 secondes
votre score finale est 10 /10

>>> partie()
1 | 0 =
entrer votre réponse :0
Raté ! score : 0 /10
Temps restant 27.5 secondes
0 & 0 =
entrer votre réponse :0
Gagné ! score : 1 /10
Temps restant 24.5 secondes
0 & 0 =
entrer votre réponse :0
Gagné ! score : 2 /10
Temps restant 14.9 secondes
0 & 1 =
entrer votre réponse :0
Gagné ! score : 3 /10
Temps restant 11.7 secondes
1 | 0 =
entrer votre réponse :1
Gagné ! score : 4 /10
Temps restant 9.5 secondes
1 | 0 =
entrer votre réponse :0
Raté ! score : 4 /10
Temps restant 6.7 secondes
1 ^ 1 =
entrer votre réponse :0
désolé, vous avez dépassé le temps
la dernière réponse ne sera pas comptée
Temps écoulé: votre score est 4 /10

```