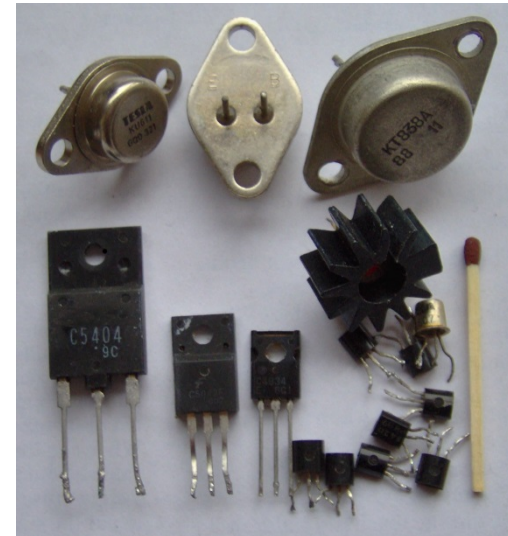


## True/False



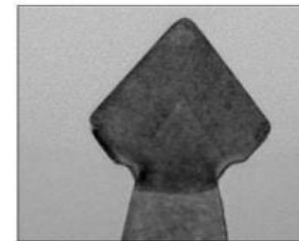
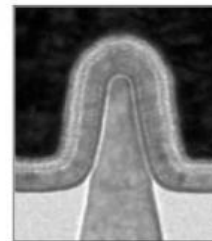
Une réplique du premier transistor(1947)



Quelques modèles de transistors



Un processeur récent  
(plus d'1 milliard de transistors)



Ordre de grandeur : 22nm

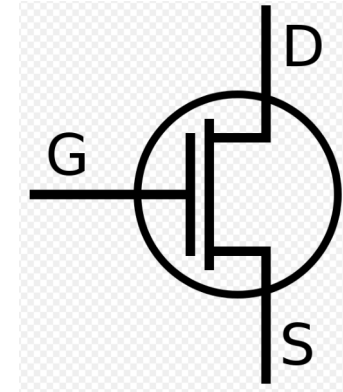
$$1\text{ nm} = 10^{-9}\text{ m}$$

## 1) Principe de fonctionnement d'un transistor

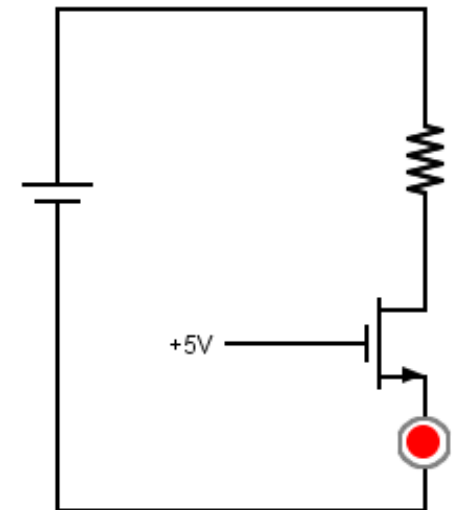
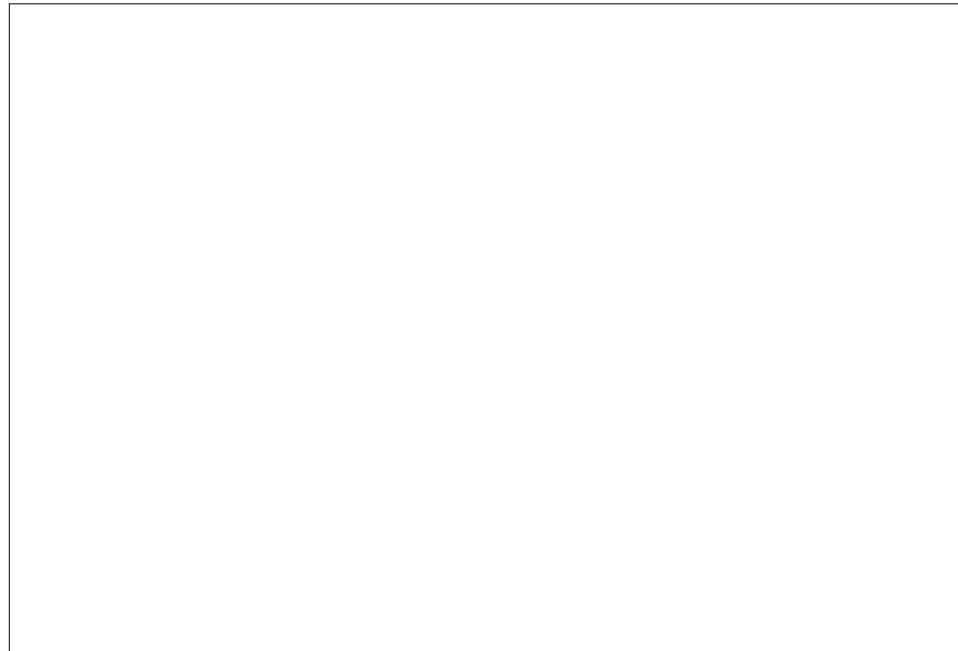
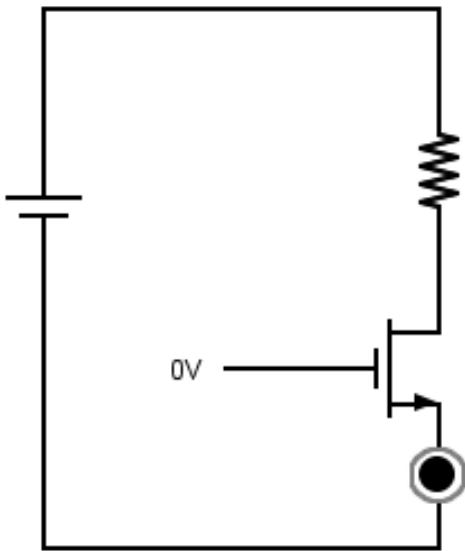
Un transistor est un composant électronique à trois branches.

La résistance entre le drain ( D ) et la source ( S ) varie en fonction de la tension appliquée entre la grille ( G ) et ( S ).

Quand cette tension est inférieure (**supérieure**) à un certain seuil, cette résistance est très grande (**petite**), le transistor est dit bloqué (**passant**).

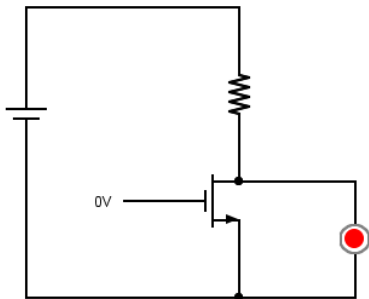


2) Avec l'application "circuit simulator", importer le fichier "transistor". Déterminer la tension à partir de laquelle le transistor est passant



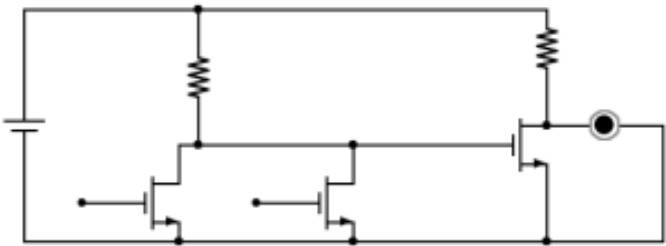
3) A l'aide de la même application, compléter les table correspondantes aux deux circuits suivants :

Circuit 2



Entrée	Sortie

Circuit 3

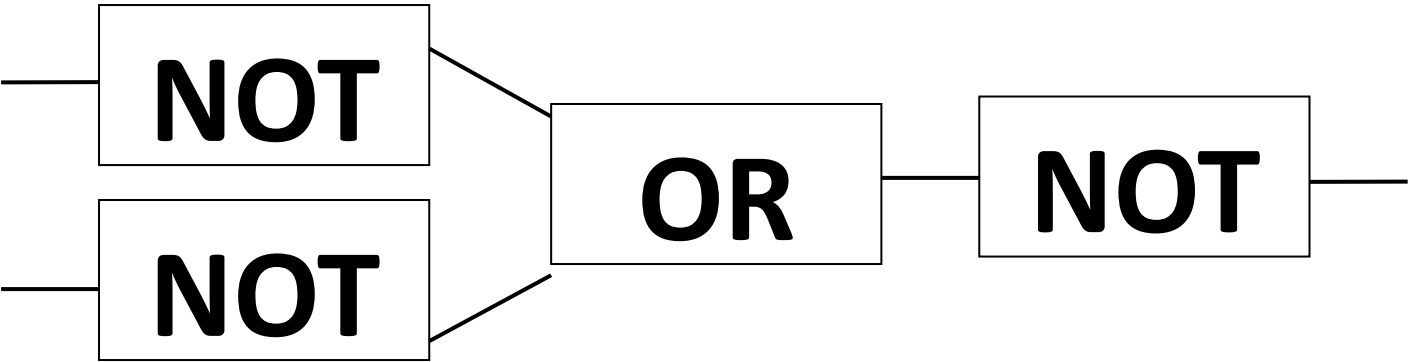


Entrée1	Entrée2	Sortie

Un autre niveau de représentation : Les portes booléennes(ou portes logiques)



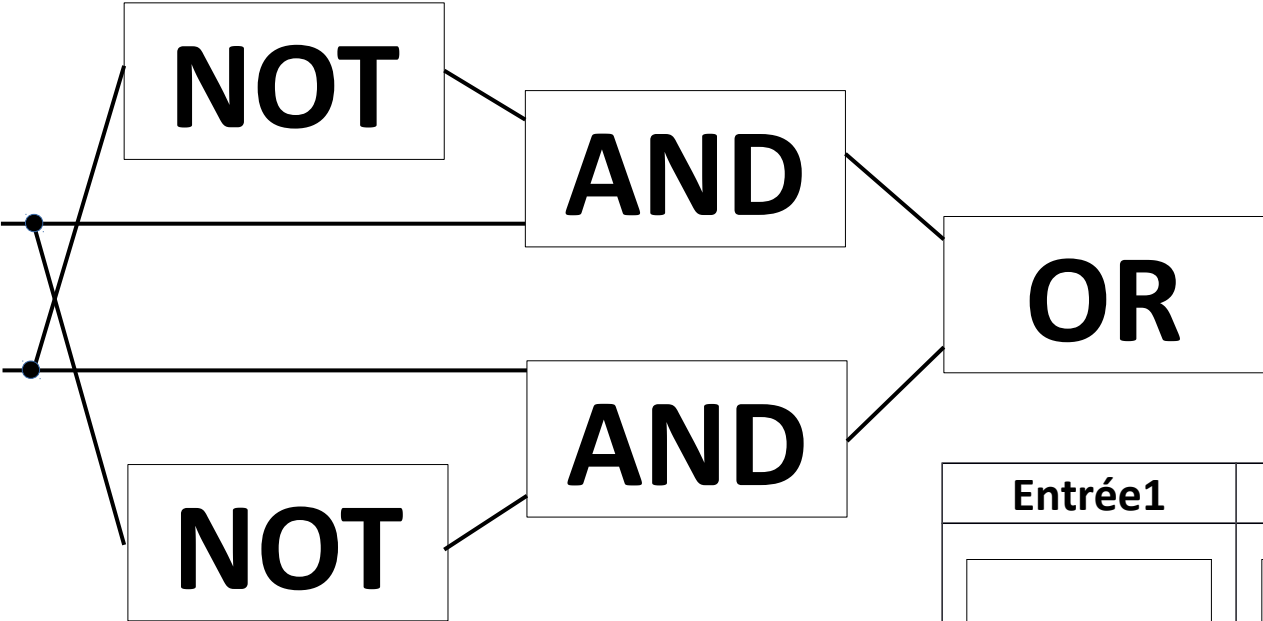
4) Compléter la table d'expression booléenne du circuit suivant



Entrée1	Entrée2	Sortie
<div></div>	<div></div>	<div></div>
<div></div>	<div></div>	<div></div>
<div></div>	<div></div>	<div></div>
<div></div>	<div></div>	<div></div>



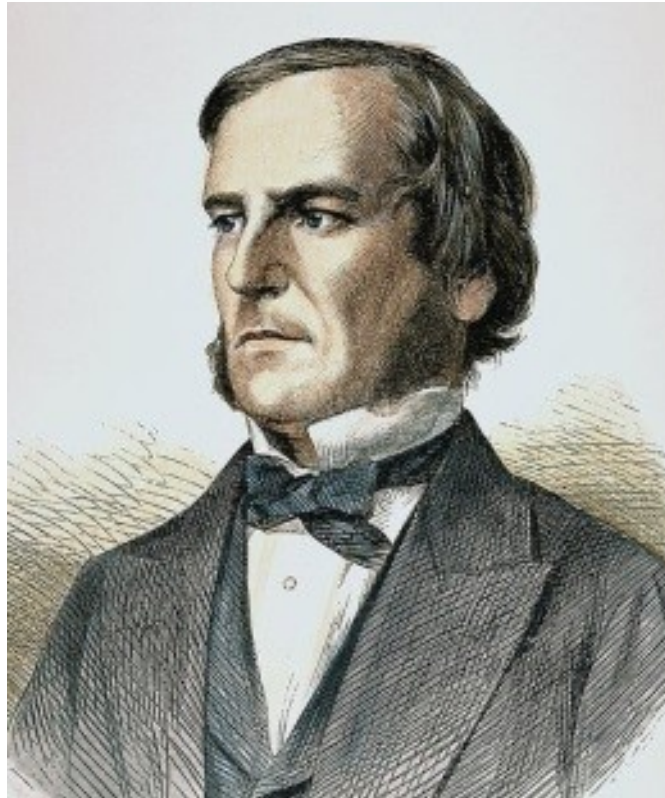
5) Compléter la table d'expression booléenne du circuit suivant :



XOR : ou exclusif  
Une application directe : l'addition binaire  
0+1=1  
1+0=1  
0+0=0  
1+1=0 et on retient 1

Entrée1	Entrée2	Sortie
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>

6) Qui est-ce ?



## 7/Synthèse

8/Dresser la table d’une expression booléenne

Quelque soit l’expression booléenne considérée, on peut dresser sa table de vérité en listant toutes les valeurs possibles en entrée et calculant pour chaque cas la valeur de sortie.

Exercice 1 :

Dresser la table de vérité de l’expression  
l’expression (A OR B ) AND C :

A	B	C	Sortie
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Exercice 2 :

Comparer les tables de vérité de ces 2 expressions  
NOT(A OR B)                      NOT(A) AND NOT(B)

A	B	NOT(A OR B)	NOT(A) AND NOT(B)
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<div></div>			



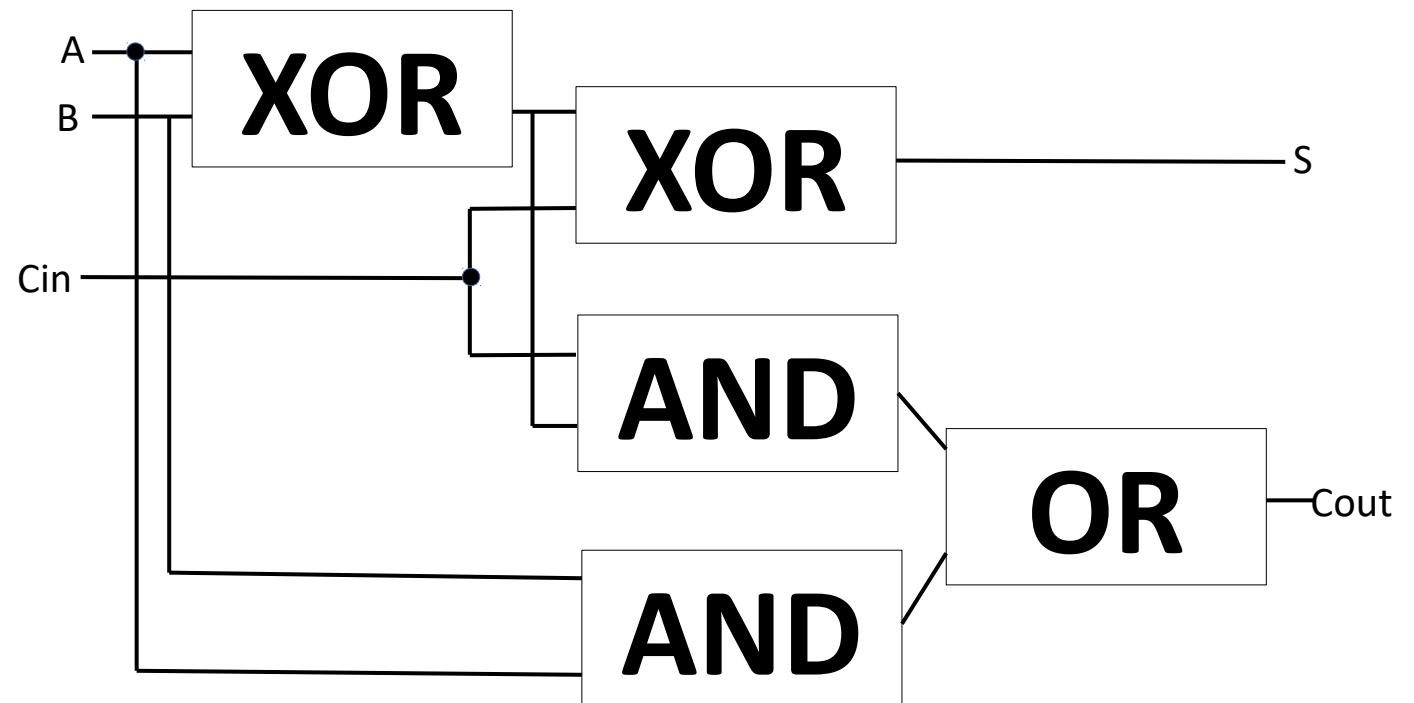
## 9/ Principe de l'addition binaire

En combinant les portes logiques, on obtient des circuits plus complexes. Par exemple en combinant 2 portes "XOR ", 2 portes "AND" et une porte "OR" on obtient un additionneur :

Comme son nom l'indique, l'additionneur permet d'additionner 2 bits (A et B) en tenant compte de la retenue entrante ("Cin" "carry in" en anglais). En sortie on obtient le résultat de l'addition (S) et la retenue sortante ("Cout" « Carry out »).

**Exercice : Compléter cette table**

A	B	Cin	Sortie	Cout
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>



En assemblant plusieurs fois ce circuit, on peut obtenir un additionneur sur plusieurs bits.

## 10/ Retour sur python

Exercice :

Les opérateurs and or et not existent sur python, mais pas l'opérateur xor.  
Ecrire la fonction `xor(a,b)` qui prend en paramètres les booléens a et b et qui renvoie le résultat de a xor b .  
Vérifier ensuite la table de vérité du ou exclusif avec python

Exercice :

x est une variable de type float. Donner une expression python signifiant :

1/ Que x est compris entre 0 et 10.

2/ Que x est compris strictement entre -3 et -1  
ou strictement entre 4 et 5

Exercice :

Tester les instructions suivantes :

`a=0`

`a!=0 and 0<1/a`

`0<1/a and a!=0`

Que se passe-t-il ? Pourquoi ?

11/ Vrai ou faux ?

`(True and False) or (not(True or False) and False) or (True and not False)`

12/ Encore un circuit

Entrée1	Entrée2	Sortie
<div></div>	<div></div>	<div></div>
<div></div>	<div></div>	<div></div>
<div></div>	<div></div>	<div></div>
<div></div>	<div></div>	<div></div>

