



*Au commencement était le transistor,
puis nous créâmes les portes booléennes
et, à la fin de la journée, les ordinateurs.*

Dans ce chapitre, nous voyons de quoi sont faits les ordinateurs à l'échelle microscopique.

Nous partons du transistor et construisons successivement des circuits non et ou qui vont nous permettre ensuite de construire les circuits de toutes les fonctions booléennes.

Frances Allen (1932-) est une pionnière de la parallélisation automatique des programmes, c'est-à-dire de la transformation de programmes destinés à être exécutés sur un ordinateur séquentiel - contenant un unique processeur - en des programmes destinés à être utilisés sur un ordinateur parallèle - contenant plusieurs processeurs. Elle est aussi à l'origine de nouvelles méthodes, fondées sur la théorie des graphes, pour optimiser les programmes. Elle a reçu le prix Turing en 2006 pour ces travaux.

Un ordinateur est un assemblage de **transistors**.

Un **transistor** est un circuit électronique à trois fils appelés le **drain**, la **source** et la **grille**.

La résistance entre le drain et la source est ou bien très petite ou bien très grande en fonction de la tension appliquée entre la grille et la source.

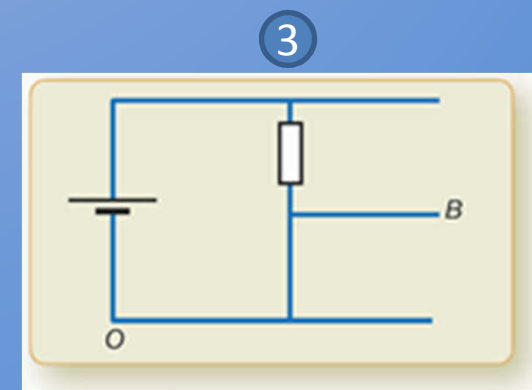
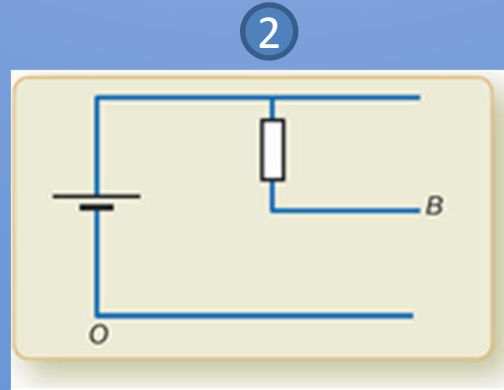
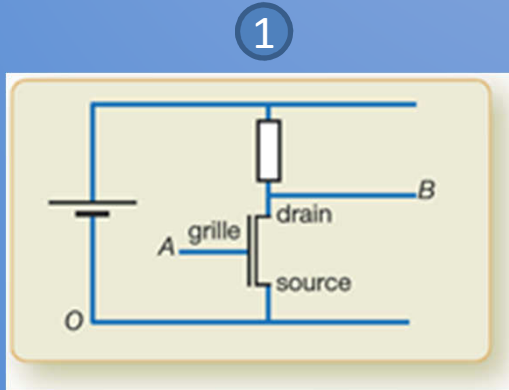
Quand cette tension est inférieure à un certain seuil, cette résistance est très grande, on dit que le transistor est **bloqué** ; quand la tension est supérieure à ce seuil, la résistance est très petite, on dit que le transistor est **passant**.

Le circuit non

Avec un transistor, une résistance et un générateur dont la tension est supérieure au seuil de basculement du transistor, on peut construire le circuit ①.

Si on applique entre le point A et le point O une tension inférieure au seuil de basculement du transistor, celui-ci est bloqué et le circuit est équivalent au circuit ②, si bien que la tension entre les points B et O est égale à la tension d'alimentation. Elle est donc supérieure au seuil de basculement.

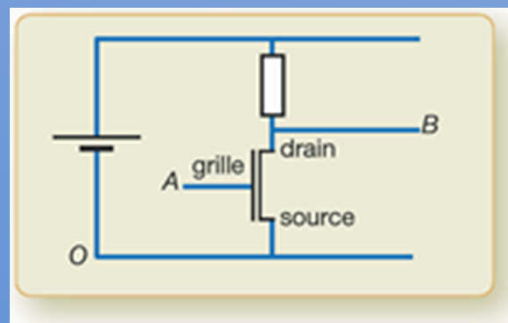
Si, en revanche, on applique entre les points A et O une tension supérieure au seuil de basculement du transistor, celui-ci est passant et le circuit est équivalent au circuit ③, si bien que la tension entre les points B et O est nulle. Elle est donc inférieure au seuil de basculement.



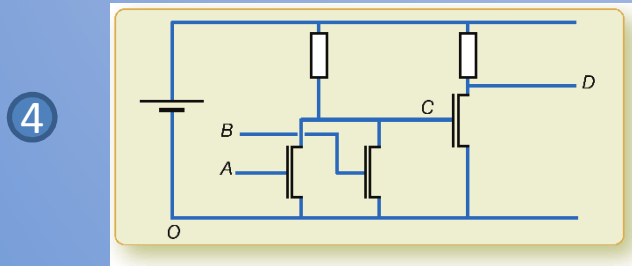
Si on décide qu'une tension inférieure au seuil de basculement représente le bit 0 et qu'une tension supérieure à ce seuil représente le bit 1, les deux remarques précédentes se reformulent ainsi :

- si on donne au circuit le bit 0 en A, il donne le bit 1 en B ;
- si on lui donne le bit 1 en A, il donne le bit 0 en B.

Autrement dit, ce circuit calcule une fonction booléenne : la **fonction non**.



Le circuit ④ est construit selon les mêmes principes, mais il a deux entrées A et B.



Si on donne aux deux entrées A et B le bit 0, les deux transistors dans la partie gauche du circuit sont bloqués, si bien que la tension entre les points C et O est égale à la tension d'alimentation, supérieure au seuil de basculement. Le transistor de droite est donc passant et la tension entre les points D et O est nulle ; autrement dit le point D est dans l'état 0.

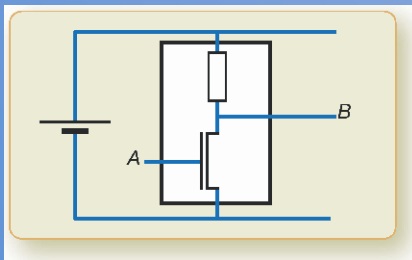
Si on donne à l'une ou l'autre des entrées A et B le bit 1, au moins l'un des deux transistors dans la partie gauche du circuit est passant, si bien que la tension entre les points C et O est nulle. Le transistor de droite est donc bloqué et la tension entre D et O est égale à la tension d'alimentation. Le point D est par conséquent dans l'état 1.

La table de ce circuit est donc la suivante (voir ci-contre) où l'on reconnaît la table de la fonction **ou**.

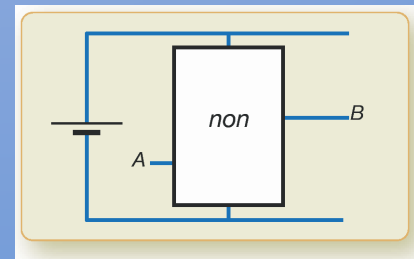
A	B	D
0	0	0
0	1	1
1	0	1
1	1	1

On peut schématiser ces circuits de manière plus succincte en remplaçant le morceau de dessin représentant le transistor et la résistance encadrés dans la figure ⑤ par un simple rectangle ⑥ et en remplaçant de même le morceau de dessin représentant les trois transistors et les deux résistances encadrés dans la figure ⑦ par un rectangle ⑧).

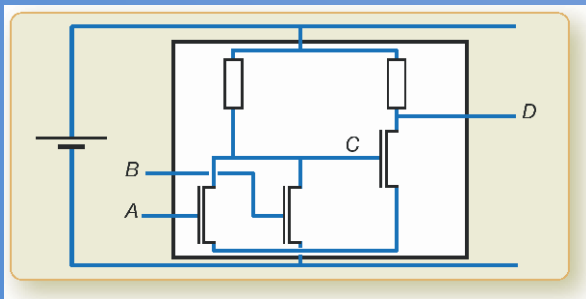
⑤



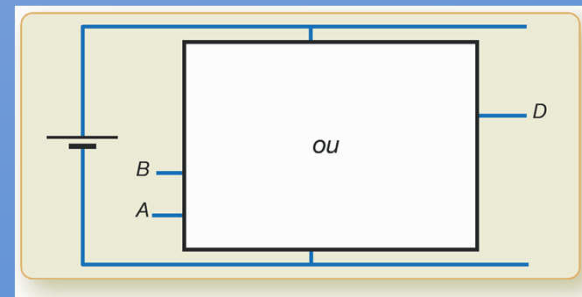
⑥



⑦



⑧

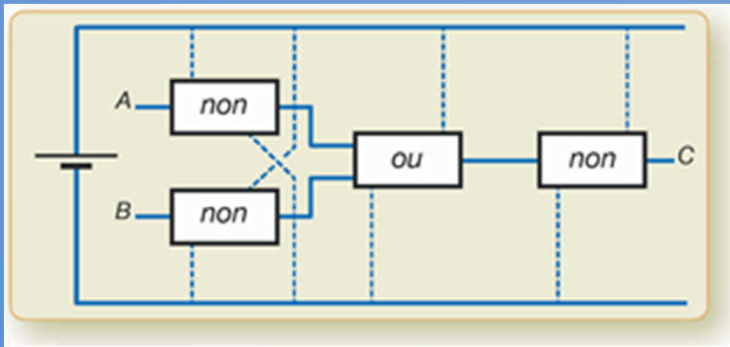


Les circuits non et ou s'appellent des portes booléennes ou parfois des portes logiques.

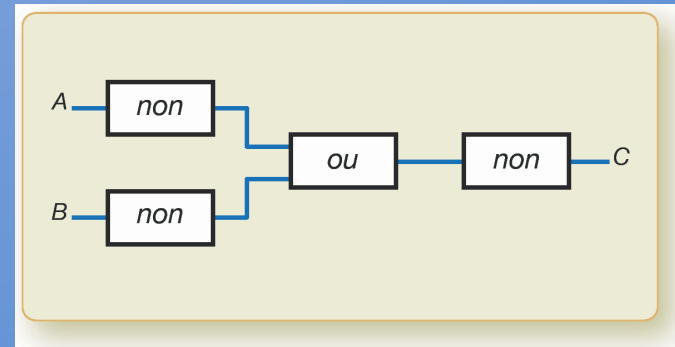
Dans ce chapitre et le suivant, on constitue petit à petit une boîte à outils de circuits réutilisables pour concevoir des circuits plus sophistiqués. Les portes non et ou sont les deux premiers éléments de cette boîte à outils.

Bien souvent, quand on représente un circuit, on ne dessine pas le générateur : il est implicite que chaque porte est alimentée. On obtient alors une troisième manière de représenter les circuits où le circuit 9 est représenté comme sur le schéma 10

9



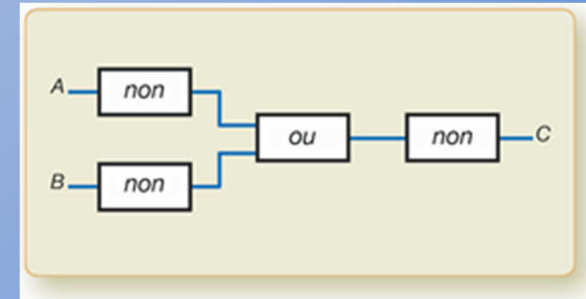
10



Exercice 1

Quelle est la table du circuit suivant ?

Est-ce la table d'une fonction booléenne connue ?



Correction exercice 1

La table de ce circuit est :

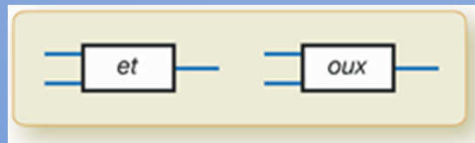
A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

C'est celle de la fonction booléenne et .

Exercice 2

Construire un circuit réalisant la fonction booléenne ou exclusif.

En plus des portes ou et non, on a désormais dans sa boîte à outils les portes **et** et **oux** :



B ———

Correction exercice 2

A	B	A ou B
0	0	0
0	1	1
1	0	1
1	1	0

La table de la fonction ou exclusif oux est :

Nous avons vu dans le chapitre les « fonctions booléennes » que :

$$A \text{ oux } B = (\text{non}(B) \text{ et } A) \text{ ou } (B \text{ et non}(A))$$

Un circuit possible utilisant les portes **non**, **ou** et **et** est donc :



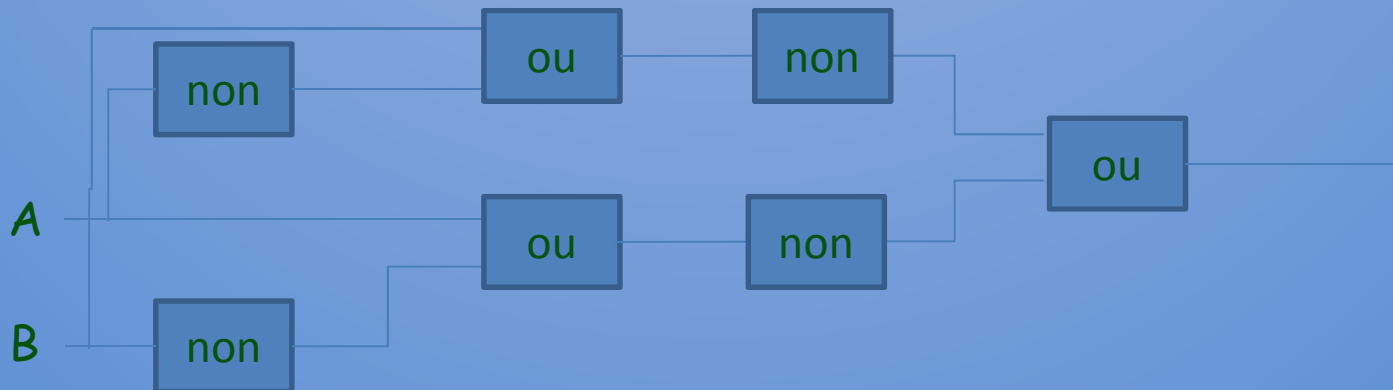
Correction exercice 2

Si on souhaite utiliser uniquement des portes **ou** et **non**,

On utilise A et $B = \text{non}(\text{non}(A) \text{ ou } \text{non}(B))$

D'où : $A \text{ ou } B = (\text{non}(B) \text{ et } A) \text{ ou } (B \text{ et } \text{non}(A)) = \text{non}(B \text{ ou } \text{non}(A)) \text{ ou } \text{non}(\text{non}(B) \text{ et } A)$

Un circuit possible utilisant les portes **non** et **ou** est donc :



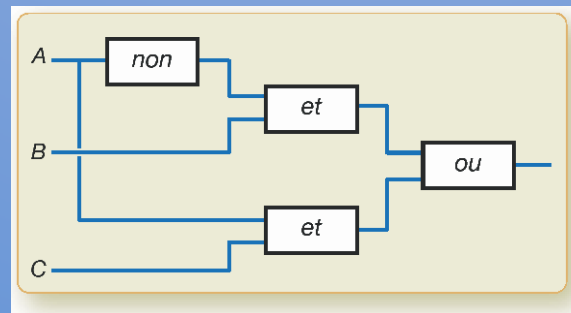
Exercice 3

Construire un circuit réalisant la fonction multiplexeur vue au chapitre « Les fonctions booléennes ».

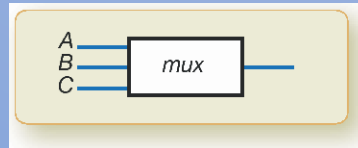
Correction exercice 3

La fonction multiplexeur peut se définir par $\text{mux}(A, B, C) = (\text{non}(A) \text{ et } B) \text{ ou } (A \text{ et } C)$.

Un circuit, parmi d'autres, réalisant cette fonction est donc :



On peut désormais utiliser directement le circuit suivant, dont l'unique sortie transmet la valeur de B ou de C selon la valeur de A :



Exercice 4

Construire le circuit réalisant le calcul de la fonction **Cout** définie par la table :

A	B	Cin	Cout
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Quelle est cette fonction ?

Correction exercice 4

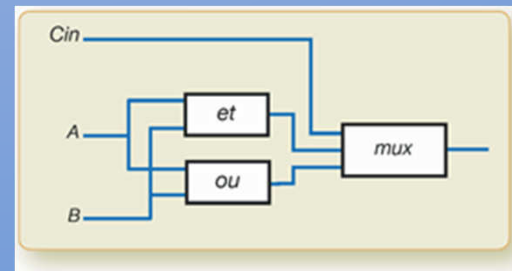
On utilise la méthode de décomposition par multiplexage (voir le chapitre « les fonctions booléennes »).

La fonction $Cout(Cin, A, B)$ s'écrit $mux(Cin, g(A, B), g'(A, B))$, avec $g(A, B)$ et $g'(A, B)$ définies par les tables ci-contre :

On reconnaît les tables des fonctions **et** et **ou**, respectivement.

Si bien que $Cout(A, B, Cin) = mux(Cin, A \text{ et } B, A \text{ ou } B)$.

Un circuit calculant cette fonction est donc :



A	B	g
0	0	0
0	1	0
1	0	0
1	1	1

A	B	g'
0	0	0
0	1	1
1	0	1
1	1	1

Cette fonction est la fonction chiffre des dizaines de $A + B + Cin$, qui sert au calcul de la retenue dans l'algorithme de l'addition de deux bits.

Ce circuit porte le nom de **Carry out** (retenue sortante).