

Architecture : portes logiques

Objectifs :

- Introduction à l'architecture d'un ordinateur : des transistors aux portes logiques..
- Les portes logiques

I Un peu d'histoire

Le premier ordinateur moderne qui repose sur le modèle de Von Neumann est l'ENIAC en 1945. L'architecture qui a gardé son nom est basé sur l'idée que le programme est enregistré dans la mémoire. La mémoire de l'ordinateur sert à stocker les données mais aussi les programmes. Il n'utilisait pas de transistors mais des tubes électriques. Il pesait 30 tonnes et était un million de fois plus lent qu'un ordinateur des années 2000. Les tubes chauffaient trop ou tombaient en panne à cause d'insectes qui grillaient dessus. Il est resté l'expression « bug ».

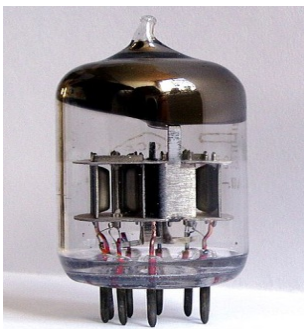


Figure 2: Un tube électronique

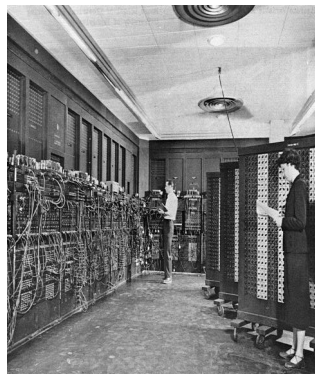


Figure 1: l'ENIAC



Figure 3: Quelques transistors

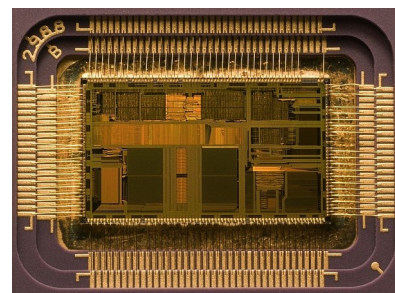
Le *transistor* inventé en 1947 devient très fiable et bon marché à partir de 1954 : il remplace avantageusement les tubes à vide notamment en raison de sa petite taille et de sa faible consommation.

Maintenant, les transistors sont directement gravés sur du silicium pour former des circuits imprimés. La miniaturisation de ces composants a permis l'essor de l'informatique grand public : En 1971, le processeur Intel 4004 contenait 2300 transistors, en 2019 le Threadripper 2990WX en contient 19,2 milliards.

Le premier circuit imprimé :

<http://www.espace-turing.fr/Demonstration-du-premier-circuit.html>

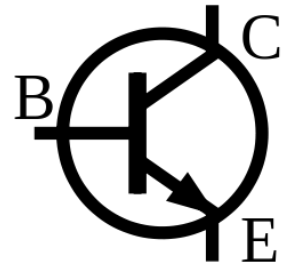
L'intérieur du processeur Intel 80486DX2 produit de 1989 à 2007, un peu plus d'un million de transistors de taille de l'ordre de 1000 μm .



II Qu'est-ce qu'un transistor ?

Pour nous un transistor est essentiellement un interrupteur.

Schéma d'un transistor : La tension de référence est reliée au collecteur C.. Si une tension suffisante est appliquée à l'émetteur B, le transistor laisse passer le courant et la tension de référence se retrouve au point E. Dans le cas contraire, le point E reste au potentiel de la *masse*, soit 0V.



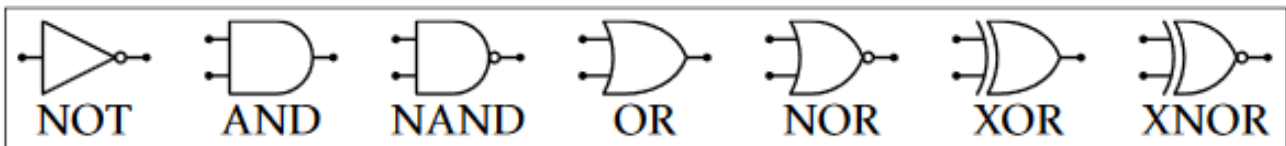
Une tension basse peut représenter 0 et une tension haute 1.

Les circuits d'un ordinateur moderne (mémoire, CPU) manipulent uniquement des tensions hautes ou basses. Les *circuits électroniques* se décomposent en de nombreux éléments dont les plus petits sont les *transistors*.

Associer des transistors permet de créer les portes logiques (or, and, etc.) ce que nous allons voir. Assembler des portes logiques permet de créer l'ensemble des parties logiques (addition, comparaison, branchement, etc.) ainsi que les mémoires d'un ordinateur. Ce que nous verrons prochainement.

III Les portes logiques

Les principales portes logiques :



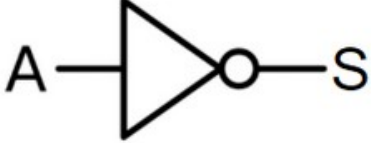

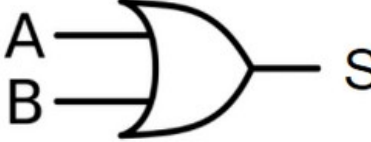

Une porte logique peut avoir plusieurs entrées qui sont à l'état haut ou à l'état bas et comporte une ou des sorties qui sont aussi soit à l'état haut soit à l'état bas. Ces portes réalisent des opérations booléennes. Chaque porte logique est caractérisée par sa table de vérité qui est un tableau dont les premières colonnes sont les états d'entrées possibles et les dernières les états de sorties.

L'état haut est codé par 1 et l'état bas est codé par 0.


Les portes logiques les plus utilisées sont non (not), et (and), ou (or) et ou exclusif (xor).

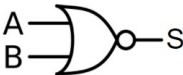
Dans l'expression « fromage ou dessert », qui veut dire que c'est soit désert soit fromage, mais pas les deux. Il s'agit du ou exclusif car les deux possibilités en même temps est exclu.

Dans l'expression « Voulez-vous du lait ou du sucre, dans votre café ? » La personne peut demander du lait, du sucre ou les deux. Il s'agit du ou sous entendu inclusif.

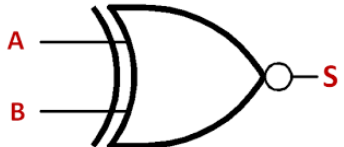
Nom	Symbole (ANSI)	Table de vérité
NON		
ET AND		
OU OR		
OU exclusif XOR		

Voici les tables de vérité des portes NAND (Not AND) et NOR (Not OR). :

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

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

Enfin la table de vérité de la porte XNOR :

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

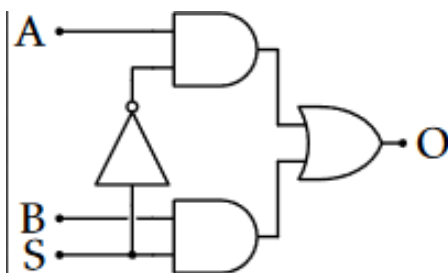
En fait, on peut démontrer que l'on peut construire toutes les portes logiques uniquement en utilisant des portes NOR ou uniquement des portes NAND.

Par exemple, le module de guidage d'Appollo sur la Lune était un ordinateur dont le processeur contenait un peu plus de 5000 portes NOR.

Pour en savoir plus: <http://www.righto.com/2019/09/a-computer-built-from-nor-gates-inside.html>

En assemblant différentes portes logiques, on obtient des circuits qui permettent des opérations plus complexes.

Le diagramme ci-dessous correspond à celui d'un **muxer**. Si $S = 0$ c'est A qui ressort en O, sinon c'est B. Ce que l'on constate dans sa table de vérité.



Addition sur 2 bits.

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

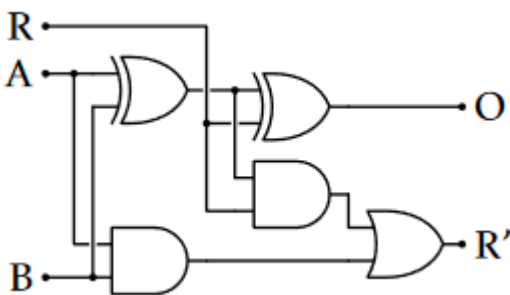
Pour faire une addition sur 2 bits sans tenir compte d'une éventuelle retenue, on utilise un xor et un and pour déterminer si il y a une retenue. On appelle cela un **demi-additionneur**.

Voici le schéma d'une demi-additionneur et sa table de vérité.



A	B	O	R
0	0		
0	1		
1	0		
1	1		

Pour faire un **additionneur** qui prend en argument une éventuelle retenue, le schéma est bien plus complexe. Voici le schéma et sa table de vérité :



A	B	R	O	R'
0	0	0		
0	1	0		
1	0	0		
1	1	0		
0	0	1		
0	1	1		
1	0	1		
1	1	1		