

Circuits combinatoires et logique booléenne

Thèmes architectures matérielles et types de données de base

Première NSI Lycée du Parc

Table des matières

Crédits	1
Préambule	1
1 Portes logiques	2
1.1 Le transistor porte logique de base	2
1.2 D'autres portes logiques	4
1.2.1 Transistors en série ou en parallèle	4
1.2.2 Portes logiques et fonctions logiques élémentaires	5
2 Fonctions booléennes	8
2.1 Fonctions booléennes	8
2.2 QCM types E3C	10
2.3 Pour aller plus loin (hors programme de première NSI)	12
2.3.1 Dresser la table de vérité d'une fonction booléenne	12
2.3.2 Exprimer une fonction booléenne à partir de sa table de vérité	13
3 Circuits combinatoires	13
3.1 Définition	13
3.2 Décodeur avec 2 bits d'entrées	14
3.3 Demi-additionneur et additionneur 1 bit	15
3.4 Simuler le hasard	17
4 Opérations bit à bit en Python (hors programme de première NSI)	18

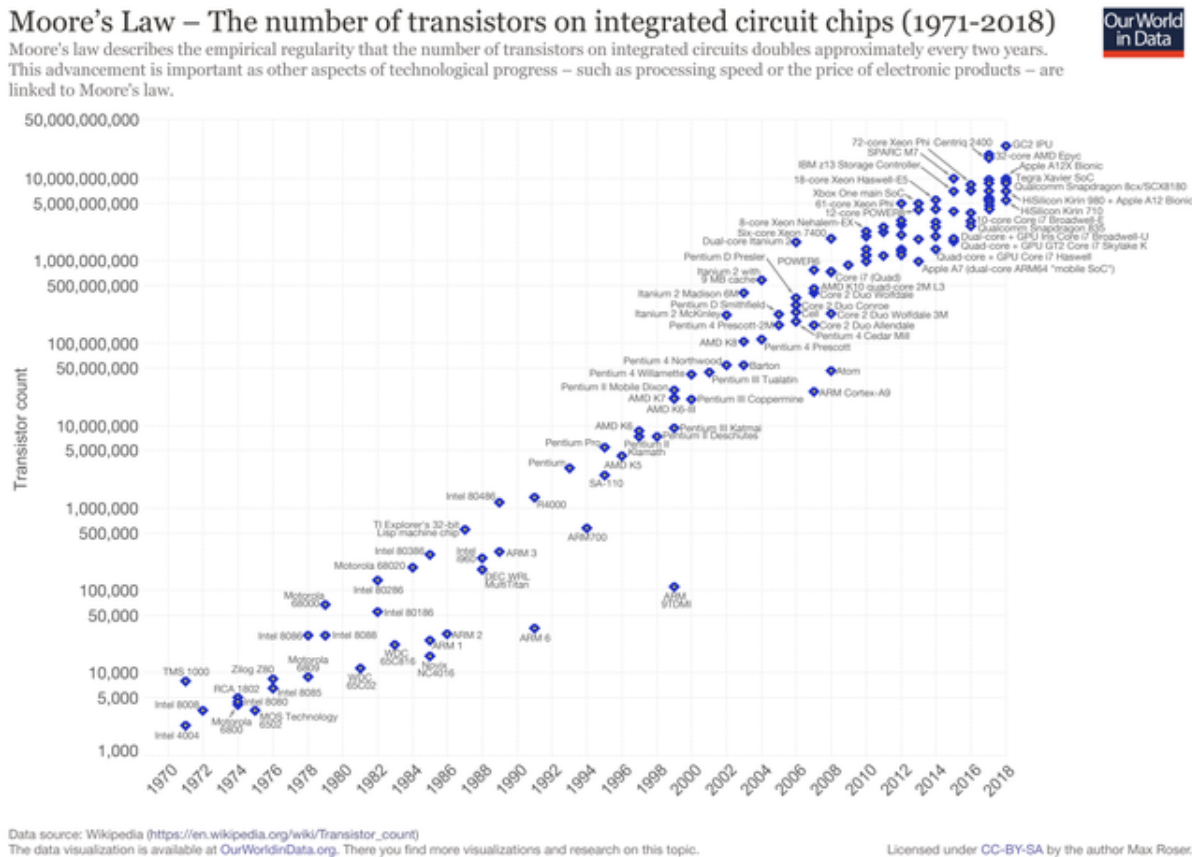
Crédits

Ce cours est largement inspiré du chapitre 22 du manuel NSI de la collection Tortue chez Ellipsen auteurs : Ballabonski, Conchon, Filliatre, N'Guyen.

Préambule

Les circuits d'un ordinateur manipulent uniquement des 0 ou des 1 représentés en interne par des tensions hautes ou basses. Les premiers ordinateurs construits dans la période 1945-1950 sont basés sur

une technologie de tube à vide ou tube électrique. En 1947, aux laboratoires Bell, [Shockley](#), [Bardeen](#) et [Brattain](#) inventent le **transistor** au *germanium* un petit composant électronique qui se comporte comme un interrupteur. Les transistors, plus petits et dissipant moins de chaleur, vont supplanter les tubes électriques : en 1954 le *germanium* est remplacé par le *silicium*, en 1955 apparaissent les premiers ordinateurs entièrement transistorisés, en 1960 le transistor à effet de champ permet l'intégration de dizaines de composants dans un centimètre carré. Les transistors sont ensuite directement gravés dans une plaque de *silicium* constituant un **circuit intégré**. En 1965 Gordon Moore futur directeur d'Intel énonce la **loi empirique** portant son nom qui fixe une feuille de route à l'industrie des microprocesseurs : le doublement de la densité d'intégration des transistors tous les deux ans. Cette loi s'est vérifiée jusqu'à présent avec une finesse de gravure d'environ 5 nanomètres en 2020. Le [graphique](#) ci-dessous représente l'évolution du nombre de transistors par circuit intégré.



1 Portes logiques

1.1 Le transistor porte logique de base



Définition 1

Un **transistor** possède trois broches : la grille, la sortie (ou drain) et la source soumis à des états de tension haute ou basse qu'on peut assimiler aux valeurs binaires 1 et 0 d'un **bit**. Si la tension appliquée

sur la grille est haute (bit à 1) alors le transistor laisse passer le courant entre la source d'énergie et la sortie et cette dernière passe à l'état de tension basse (bit à 0), sinon la sortie reste en tension haute (bit 1).

Une **fonction logique** prend un ou plusieurs bits en entrée et retourne un ou plusieurs bits en sortie. Une **porte logique** est un circuit électronique représentant une **fonction logique**.

Une **table logique** représente les sorties produites par une fonction logique pour toutes les entrées possibles.

Un transistor représente une fonction logique dont le bit d'entrée est l'état de tension de la grille et le bit de sortie, l'état de tension de la sortie. La **table logique** (table 1) associée est celle du **NON logique** ou **Inverseur**.

Fichier de test [Logisim](#) : [transistor.circ](#).

Table 1: **Table logique d'une porte NON**

A	B = NON(A)
0	1
1	0

Il existe deux conventions de représentation symbolique des portes logiques, une européenne et une américaine.

Tutoriel video [Logisim](#) : le transistor

1
1
1
0

0
0
0
1

porte NON
à partir d'une
porte NOR

1
0

porte ET
à partir de 3
portes NOR

1000

porte NOT a partir
d'une porte NAND

1
0

porte OR
a partir de 3
portes NAND

Loi de Morgan.
 $\text{not}(\text{not}(A) \text{ and } \text{not}(B))$

0
1
1
1

$$= (\text{not}(\text{not}(A))) \text{ or } (\text{not}(\text{not}(B)))$$
$$= A \text{ or } B$$

[1,0]
[1,0]

a b

