







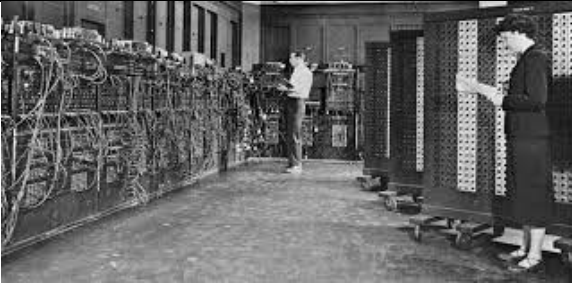
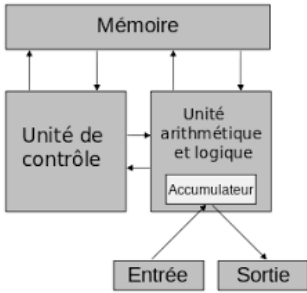
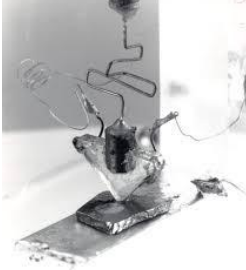
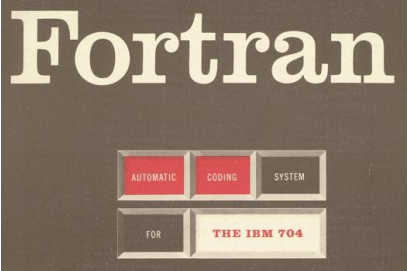










Architecture Ressource



1 Histoire de l'informatique

1er siècle av J.-C	La machine d' Anticythère est une calculatrice mécanique antique permettant de calculer des positions astronomiques	
1623	Wilhelm Schickard (1592-1635) invente pour Kepler une « horloge calculante » destinée à calculer les éphémérides.	
1644	Blaise Pascal (1623-1662) invente une machine à calculer (« <u>la Pascaline</u> ») capable d'additionner et de soustraire des nombres de huit chiffres.	
1823	Charles Babbage (1791-1871) commence à construire sa machine à différences. (Le but de la machine est de calculer les polynômes)	
1854	George Boole (1815-1864) publie un article sur la logique binaire. Les travaux de Boole, s'ils sont théoriques, n'en trouveront pas moins des applications primordiales dans les systèmes informatiques.	
1936	Alan Turing (1912-1954) publie un ouvrage qui définit les limites théoriques de l'ordinateur. Il présente le modèle des machines de Turing et construit (mathématiquement) la première machine universelle. Il casse le code Enigma en 1942.	

1943	<p>L'ENIAC (Electronic Numerical Integrator And Computer) est créé par John W. Mauchly (1907-1980) et John Presper Eckert (1919-1995). Il sera opérationnel en 1946. Son poids est de 30 tonnes pour des dimensions de 2,4 x 0,9 x 30,5 mètres occupant une surface de 67 mètres carrés. Il fut utilisé pour des calculs ayant servi à mettre au point la bombe H.</p>	 <p>Son principal inconvénient était sa programmation : l'ENIAC était en effet uniquement programmable manuellement avec des commutateurs et des câbles à enficher.</p>
1944	<p>John Von Neumann (né Neumann János, 1903-1957) a donné son nom à l'architecture de von Neumann utilisée dans la quasi- totalité des ordinateurs modernes</p>	
1947	<p>Invention du transistor par les américains John Bardeen, William Shockley et Walter Brattain.</p>	
1954	<p>Création du Fortran, premier langage de programmation à être implémenté sur un ordinateur.</p>	
1969	<p>Début du réseau Arpanet, renommé plus tard Internet.</p>	

1973	Commercialisation du Micral, le premier micro-ordinateur. C'est François Gernelle qui en est l'inventeur. Cet ordinateur ne possédait ni clavier ni écran et était commandé par des interrupteurs,	
1976	Apparition du premier supercalculateur : le Cray one.	
1978	Intel 8086 est un microprocesseur 16 bits fabriqué par Intel. C'est le premier processeur de la famille x86, qui est devenu l'architecture de processeur la plus répandue dans le monde des ordinateurs personnels (29 000 transistors gravés en 3 µm)	
1985	Apparition du CD-ROM	
1989	Tim Berners-Lee invente le World Wide Web (WWW) pour que les chercheurs puissent partager les informations au sein du CERN	
1997	Deep Blue bat Gary Kasparov. Deep Blue est un superordinateur spécialisé dans le jeu d'échecs, développé par IBM. C'est la première fois qu'un ordinateur bat un champion du monde d'échecs	
2008	Core i7 (1,60 GHz à 5 GHz) 731,000,000 transistors 45nm https://en.wikipedia.org/wiki/Transistor_count	

2 Différence entre compilateur et interpréteur

Nous écrivons généralement un programme informatique utilisant un langage haut niveau (ou langage évolué) compréhensible par l'homme. Un langage haut niveau (code source) contient des mots et des phrases de la langue anglaise. Un ordinateur quant à lui, ne comprend que les programmes écrits en binaire 0 et 1 appelé le code machine.

Les compilateurs et les interpréteurs convertissent le code source en code machine mais pas de la même manière.



Un compilateur prend tout le programme et le convertit en code objet qui est généralement stocké dans un fichier. Le code objet est également référencé en tant que code binaire et peut être exécuté directement par la machine après la liaison.

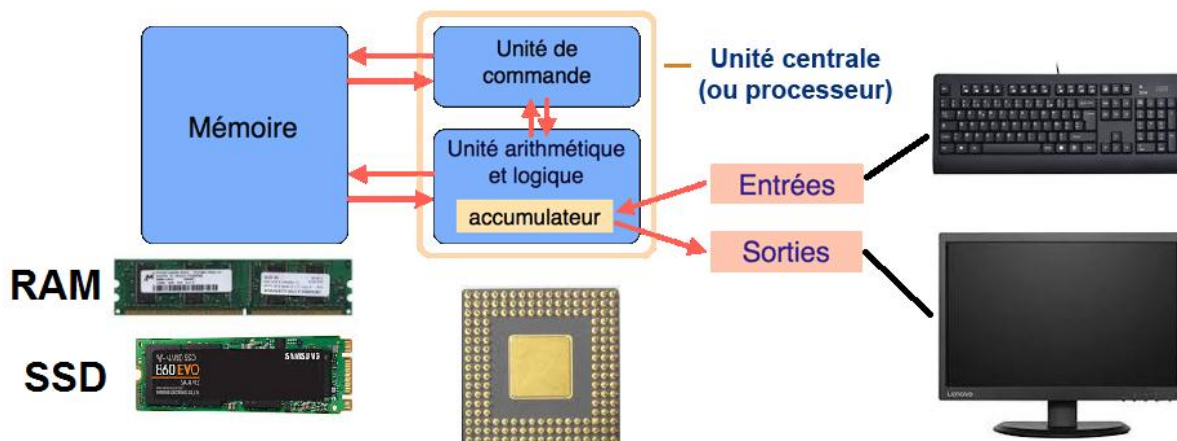


Un interpréteur exécute directement des instructions écrites dans un langage de programmation ou de script sans les convertir en un code objet ou un code machine.

Interpréteur	Compilateur
Convertit le programme en prenant une seule ligne à la fois	Analyse l'ensemble du programme et le traduit dans son ensemble en code machine
L'analyse du code source prend moins de temps, mais le temps d'exécution global est plus lent.	L'analyse du code source prend beaucoup de temps, mais le temps d'exécution global est bien plus rapide. (10x)
Aucun code d'objet intermédiaire n'est généré, la mémoire est donc efficace	Génère du code d'objet intermédiaire qui nécessite en outre une liaison, nécessite donc davantage de mémoire.
Continue de traduire le programme jusqu'à ce que la première erreur soit rencontrée. Par conséquent, le débogage est facile	Il génère le message d'erreur uniquement après avoir analysé l'ensemble du programme. Par conséquent, le débogage est relativement difficile
Langage de programmation comme Python, Ruby utilisent des interpréteurs	Langage de programmation comme C, C++ utilisent des compilateurs

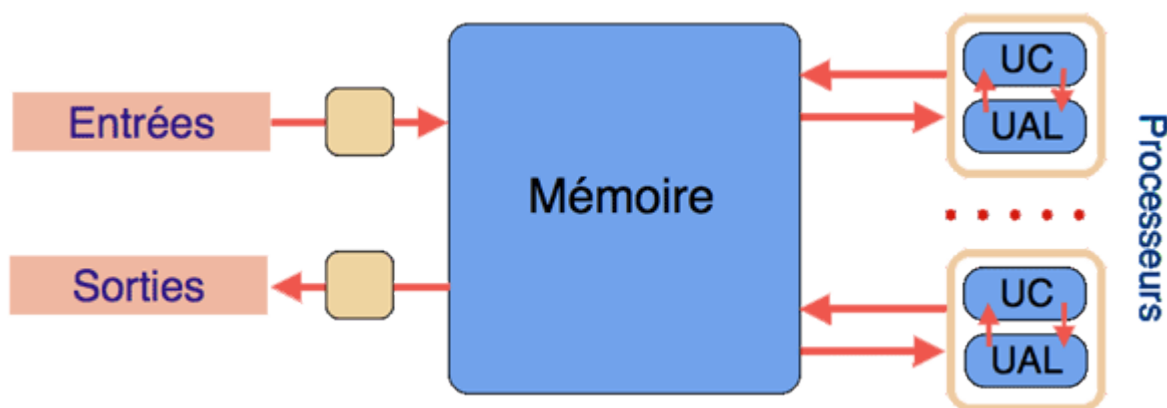
3 Le modèle d'architecture de von Neumann

Schéma :



- 1 **L'unité arithmétique et logique (UAL)** ou unité de traitement, qui effectue les opérations de base
- 2 **L'unité de commande**, qui est chargée du séquençage des opérations ;
- 3 **La mémoire**, qui contient à la fois les données et le programme qui indique à l'unité de contrôle quels calculs faire sur ces données. La mémoire se divise en mémoire vive (programmes et données en cours de fonctionnement : RAM) et mémoire de masse (programmes et données de base de la machine : SSD) ;
- 4 **Les dispositifs d'entrées-sorties**, qui permettent de communiquer avec le monde extérieur.

Le modèle de von Neumann, aujourd'hui.



Les entrées-sorties, initialement commandées par l'unité centrale, sont depuis le début des années 1960 sous le contrôle de processeurs autonomes. Associée à la multiprogrammation (partage de la mémoire entre plusieurs programmes), cette organisation a notamment permis le développement des systèmes en temps partagé.

Les ordinateurs comportent maintenant des processeurs multiples, qu'il s'agisse d'unités séparées ou de « cœurs » multiples à l'intérieur d'une même puce. Cette organisation permet d'atteindre une puissance globale de calcul élevée sans augmenter la vitesse des processeurs individuels, limitée par les capacités d'évacuation de la chaleur dans des circuits de plus en plus denses.

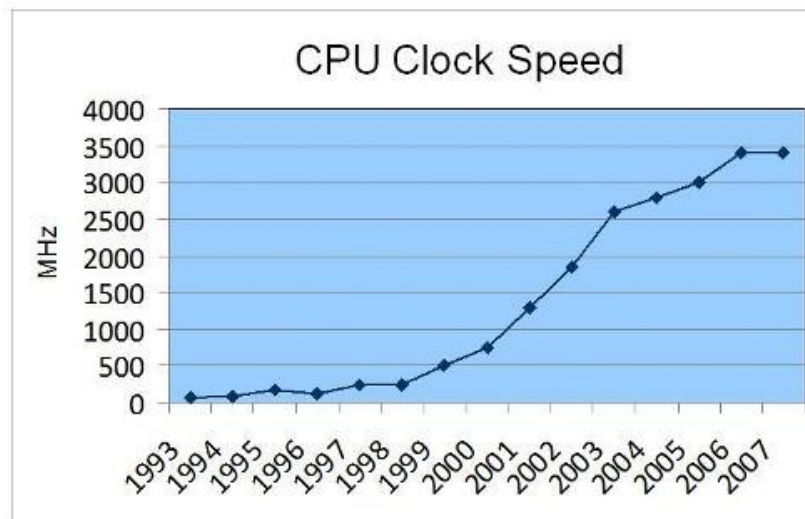
Ces deux évolutions ont pour conséquence de mettre la mémoire, plutôt que l'unité centrale, au centre de l'ordinateur, et d'augmenter le degré de parallélisme dans le traitement et la circulation de l'information. Mais elles ne remettent pas en cause les principes de base que sont la séparation entre traitement et commande et la notion de programme enregistré.

Par ailleurs, ces dernières années, la vitesse des processeurs s'est considérablement accrue. Les améliorations de la mémoire ont, quant à elles, surtout touché la densité, c'est-à-dire la capacité à stocker davantage de données dans un espace plus restreint, plutôt que le débit de transfert. Plus les vitesses augmentaient, plus le processeur restait inactif dans l'attente des données en provenance de la mémoire. Même un processeur qui fonctionne très bien à haute vitesse reste limité par le débit dans le goulot d'étranglement. Souvent, un processeur plus rapide n'a pour effet que d'augmenter le temps d'inactivité.

Il existe plusieurs moyens de remédier au goulot d'étranglement de von Neumann :

- Mise en cache : stockage des données d'usage fréquent dans une zone spéciale, généralement la mémoire vive (RAM), afin de les rendre plus facilement accessibles que si elles résidaient dans la mémoire principale.
- Prélecture : déplacement de certaines données en cache avant même qu'elles soient demandées, de façon à en accélérer l'accès le moment voulu.
- Multithreading : gestion simultanée de plusieurs demandes dans des threads distincts.
- Nouveaux types de mémoire vive (RAM, Random Access Memory) : par exemple la DDR SDRAM, qui permet le transfert des données sur les deux fronts, montant et descendant, de l'horloge système et pas uniquement sur le front montant, pour potentiellement doubler le débit.

Pendant des années, pour augmenter les performances des ordinateurs, les constructeurs augmentaient la fréquence d'horloge des microprocesseurs : la fréquence d'horloge d'un microprocesseur est liée à sa capacité à exécuter un nombre plus ou moins important d'instructions machines par seconde. Plus la fréquence d'horloge du CPU est élevée, plus ce CPU est capable d'exécuter un grand nombre d'instructions machines par seconde.



Comme on peut le remarquer sur le graphique ci-dessus, à partir de 2006 environ, la fréquence d'horloge a cessé d'augmenter, pourquoi ? À cause d'une contrainte physique : en effet plus on augmente la fréquence d'horloge d'un CPU, plus ce dernier chauffe. Il devenait difficile de refroidir le CPU et les constructeurs de microprocesseurs (principalement Intel et AMD) ont décidé d'arrêter la course à l'augmentation de la fréquence d'horloge.

On considère souvent le goulot d'étranglement de von Neumann comme un problème ne pouvant être résolu qu'en apportant des modifications significatives aux architectures des ordinateurs ou des processeurs.

On pourrait se dire que l'augmentation du nombre de cœurs entraîne obligatoirement une augmentation des performances du CPU, en fait, c'est plus complexe que cela : pour une application qui n'aura pas été conçue pour fonctionner avec un microprocesseur multicœur, le gain de performance sera très faible, voire même nul. En effet, la conception d'applications capables de tirer profit d'un CPU multicœur demande la mise en place de certaines techniques de programmation (techniques de programmation qui ne seront pas abordées ici). Il faut aussi avoir conscience que les différents cœurs d'un CPU doivent se "partager" l'accès à la mémoire vive : quand un cœur travaille sur une certaine zone de la RAM, cette même zone n'est pas accessible aux autres cœurs, ce qui, bien évidemment va brider les performances. De plus, on trouve à l'intérieur des microprocesseurs de la mémoire "ultrarapide" appelée mémoire cache (il ne faut pas confondre mémoire cache et registres). Le CPU peut stocker certaines données dans cette mémoire cache afin de pouvoir y accéder très rapidement dans le futur, en effet, l'accès à la mémoire cache est beaucoup plus rapide que l'accès à la RAM. La mémoire cache ayant un coût assez important, la quantité présente au sein d'un CPU est assez limitée, les différents cœurs vont donc devoir se partager cette mémoire cache, ce qui peut aussi provoquer des ralentissements (en fait il existe plusieurs types de mémoire cache appelés L1, L2 et L3, chaque cœur possède son propre cache L1, alors que les caches L2 et L3 sont partagés par les différents cœurs).

4 A l'intérieur du processeur INTEL core i7

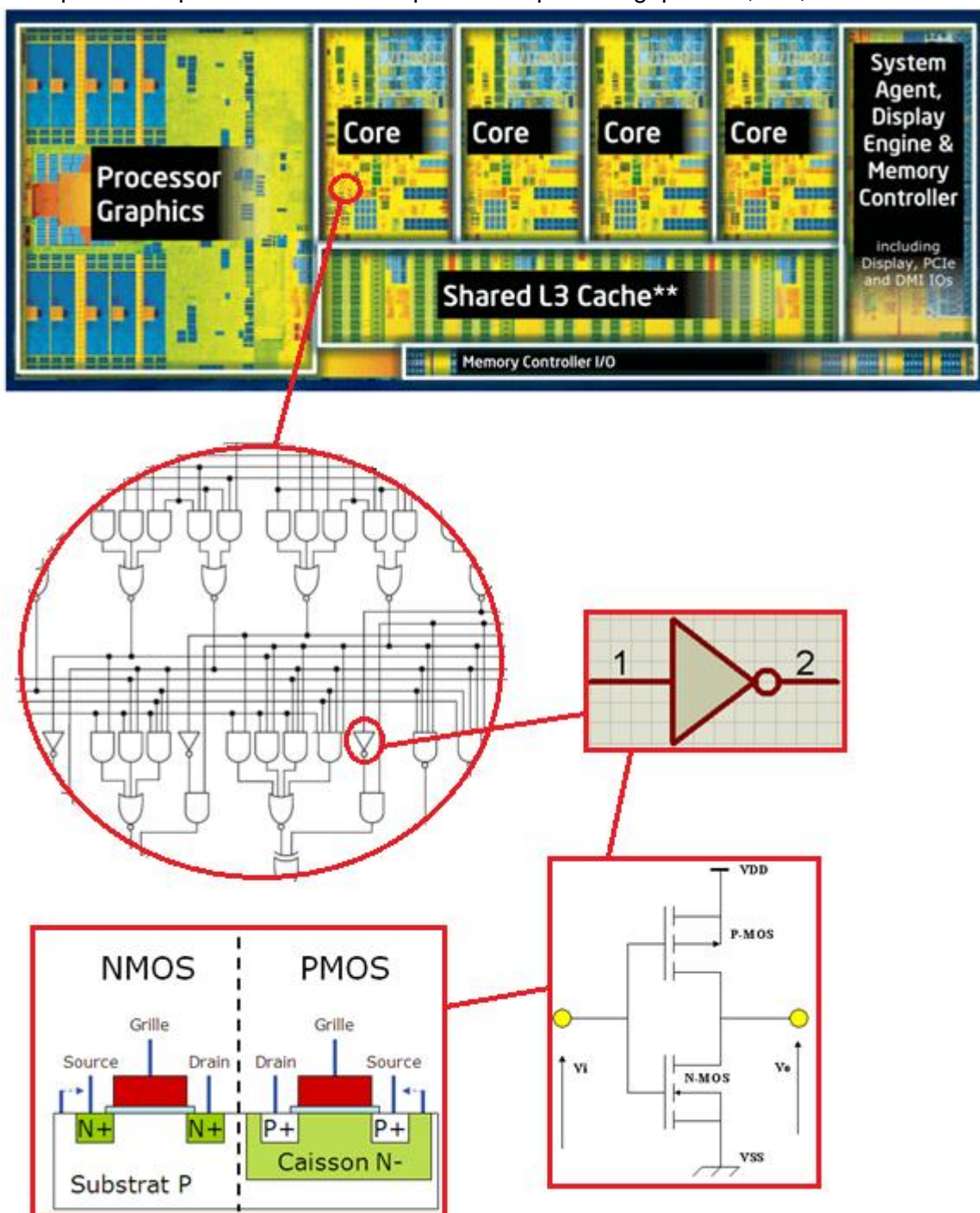
Production	Depuis 2008
Fabricant	Intel
Fréquence	1,60 GHz à 5 GHz
Fréquence du FSB	2,5 GT/s à 6,4 GT/s
Procédé de fabrication	14 nm à 45 nm
Architecture	x86, x86-64, MMX, SSE, SSE2, SSE3, SSSE3, SSE4
Micro-architecture	Nehalem, Sandy Bridge, Haswell
Socket(s)	LGA1366, LGA1156, LGA1155, LGA2011, LGA1150, LGA1151
Exemple Socket LGA1151 :	
1155 Contacts.	



L'hyper-threading consiste à créer deux processeurs logiques sur une seule puce, chacun doté de ses propres registres de données et de contrôle. Ces deux unités partagent les éléments du cœur de processeur, le cache et le bus système. Ainsi, deux processus peuvent être traités simultanément par le même processeur. Cette technique multitâche permet d'utiliser au mieux les ressources du processeur en garantissant que des données lui sont envoyées en masse.

Une mémoire cache ou antémémoire est, en informatique, une mémoire qui enregistre temporairement des copies de données provenant d'une source, afin de diminuer le temps d'un accès ultérieur à ces données.

Les différentes parties du processeur sont composées de portes logiques ET, OU, NON

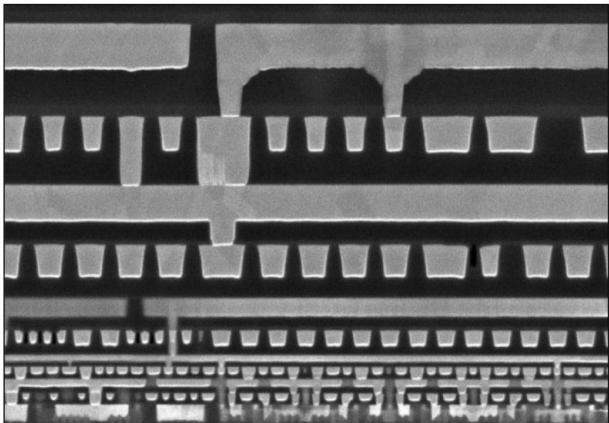
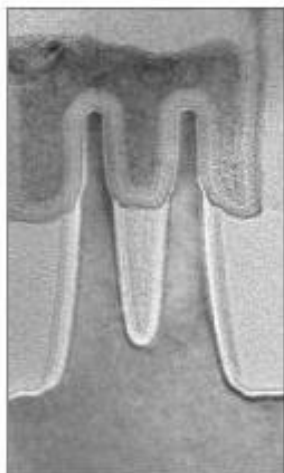


Les portes logiques élémentaires sont fabriquées à partir de transistor MOS (Métal Oxide Semiconducteur) canal P ou canal N. Un transistor est similaire à un interrupteur commandé.

	<p>Interrupteur fermé, le courant passe entre le Drain et la Source.</p>
	<p>Interrupteur ouvert, le courant ne passe pas entre le Drain et la Source</p>

Un processeur est composé de millions de transistors. La taille d'un transistor dépend du procédé de fabrication exprimée en **nano mètres** (nm).

Le procédé de fabrication est toujours amélioré.

Interconnexion des transistors dans le circuit intégré	Transistor
<p style="text-align: center;">14 nm Process</p>  <p style="text-align: center;">52 nm (0.65x) minimum pitch</p>	 <p style="text-align: center;">14 nm 2nd Generation Tri-gate Transistor</p>

MOSFET scaling (process nodes)

- 10 μm – 1971
- 6 μm – 1974
- 3 μm – 1977
- 1.5 μm – 1981
- 1 μm – 1984
- 800 nm – 1987
- 600 nm – 1990
- 350 nm – 1993
- 250 nm – 1996
- 180 nm – 1999
- 130 nm – 2001
- 90 nm – 2003
- 65 nm – 2005
- 45 nm – 2007
- 32 nm – 2009
- 22 nm – 2012
- 14 nm – 2014
- 10 nm – 2016
- 7 nm – 2018
- 5 nm – 2019
- Future**
- 3 nm – ~2021

Vidéo de la fabrication d'un processeur 14nm

<https://youtu.be/d9SWNLZvA8g>

Année	Procédé de fabrication	Exemple de processeur
2012	22 nm	Core i5 Core i7
2014	14 nm	Core i9
2016	10 nm	Apple A11 (4 300 000 000 de transistors)
2018	7nm	Apple A12 Bionic

Regarder la vidéo de « C'est pas sorcier ici » : <https://www.youtube.com/watch?v=oq6OI550K3w>

5 L'unité Arithmétique et logique

Source : <https://fr.wikipedia.org/>

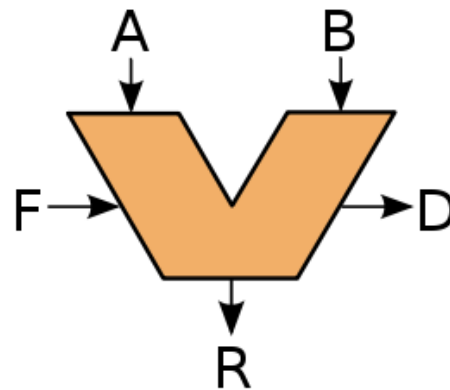
L'unité arithmétique et logique (UAL, en anglais arithmetic–logic unit, ALU), est l'organe de l'ordinateur chargé d'effectuer les calculs.

Les UAL élémentaires calculent sur des nombres entiers, et peuvent effectuer les opérations communes, que l'on peut séparer en quatre groupes :

- Les opérations arithmétiques : addition, soustraction, changement de signe etc..
- Les opérations logiques : compléments à un, à deux, et, ou, ou-exclusif, non, non-et, etc.
- Les comparaisons : test d'égalité, supérieur, inférieur
- Les décalages et rotations.

La figure ci-contre représente un schéma classique d'UAL. Celle-ci possède deux entrées A et B sur lesquelles on présente les données à traiter. L'entrée F désigne l'opération à effectuer. (Addition, soustraction etc...)

Enfin, l'UAL possède deux sorties, R qui est le résultat de l'opération, et D les éventuels drapeaux.



Il existe différents drapeaux qui définissent une erreur dans le résultat (division par zéro, dépassement de mémoire...), ou alors des codes conditions (supérieur, inférieur, égal à zéro...).

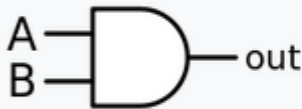
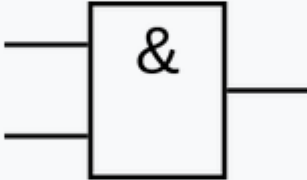

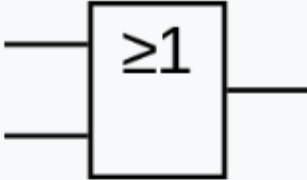
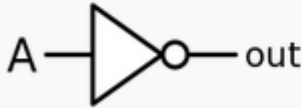

La plupart des UAL possèdent des drapeaux qui ont différentes significations, dont voici quelques exemples :

- Zéro, indique que tous les bits du Résultat R sont égaux à zéro.
- Négatif, indique que le résultat de l'opération arithmétique est négatif.
- Débordement (en anglais : overflow), indique que le résultat d'une opération arithmétique a excédé la capacité numérique de R.

6 Logique combinatoire

Regarder la vidéo de « C'est pas sorcier ici » : <https://www.youtube.com/watch?v=c96KP5jZVYk>

Symbole et table de vérité des portes logiques élémentaires :

Type	Symbole américain	Symbole européen	Opération booléenne	Table de vérité																		
ET			$A \cdot B$	<table><tr><th colspan="2">Entrée</th><th>Sortie</th></tr><tr><th>A</th><th>B</th><th>A ET B</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	Entrée		Sortie	A	B	A ET B	0	0	0	0	1	0	1	0	0	1	1	1
Entrée		Sortie																				
A	B	A ET B																				
0	0	0																				
0	1	0																				
1	0	0																				
1	1	1																				
OU			$A + B$	<table><tr><th colspan="2">Entrée</th><th>Sortie</th></tr><tr><th>A</th><th>B</th><th>A OU B</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	Entrée		Sortie	A	B	A OU B	0	0	0	0	1	1	1	0	1	1	1	1
Entrée		Sortie																				
A	B	A OU B																				
0	0	0																				
0	1	1																				
1	0	1																				
1	1	1																				
NON			\overline{A}	<table><tr><th>Entrée</th><th>Sortie</th></tr><tr><th>A</th><th>NON A</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	Entrée	Sortie	A	NON A	0	1	1	0										
Entrée	Sortie																					
A	NON A																					
0	1																					
1	0																					