

A propos du cours de remise à niveau informatique

Objectif : acquérir le vocabulaire et une petite expérience de notions essentielles à la pratique de l'informatique - architecture des ordinateurs, systèmes d'exploitation, bases de données, algorithmes.

Info pratiques

mathieu.d-aquin@univ-lorraine.fr - Bureau 211 IDMC, B160
LORIA

4 séances, présentations, discussions, exercices pratiques :

- 1 introduction à l'architecture d'un ordinateur
- 2 systèmes d'exploitation et interfaces de ligne de commandes
- 3 bases de données
- 4 algorithmes et structures de données

Evaluation

Un petit quizz, sur Arche, à la fin de chaque séance.

(*) toutes les images proviennent de wikipedia

1- Introduction à l'architecture des ordinateurs

Remise à niveau informatique

Mathieu d'Aquin

septembre 2024

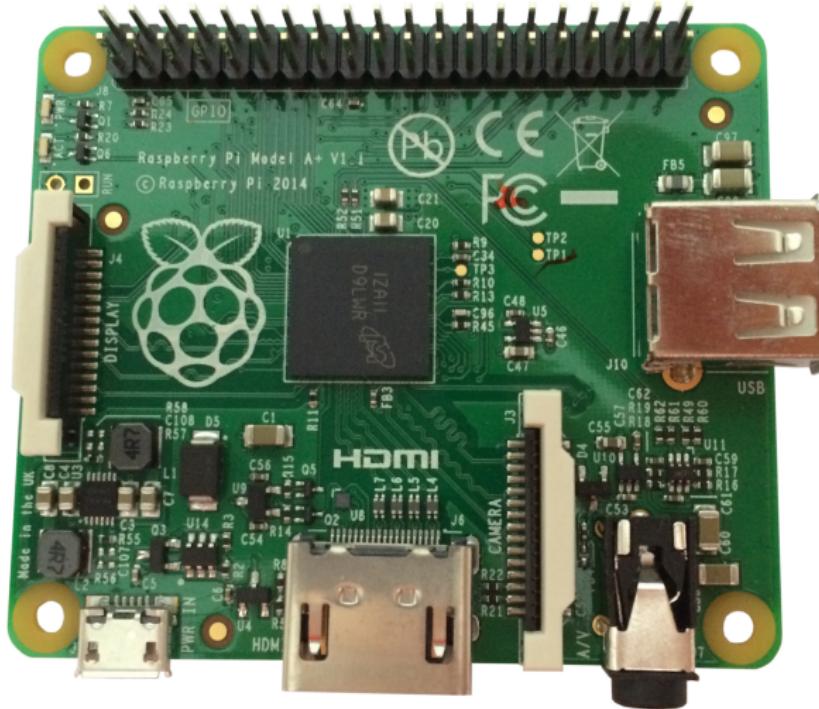
Un ordinateur



Un ordinateur aussi



Un ordinateur ?



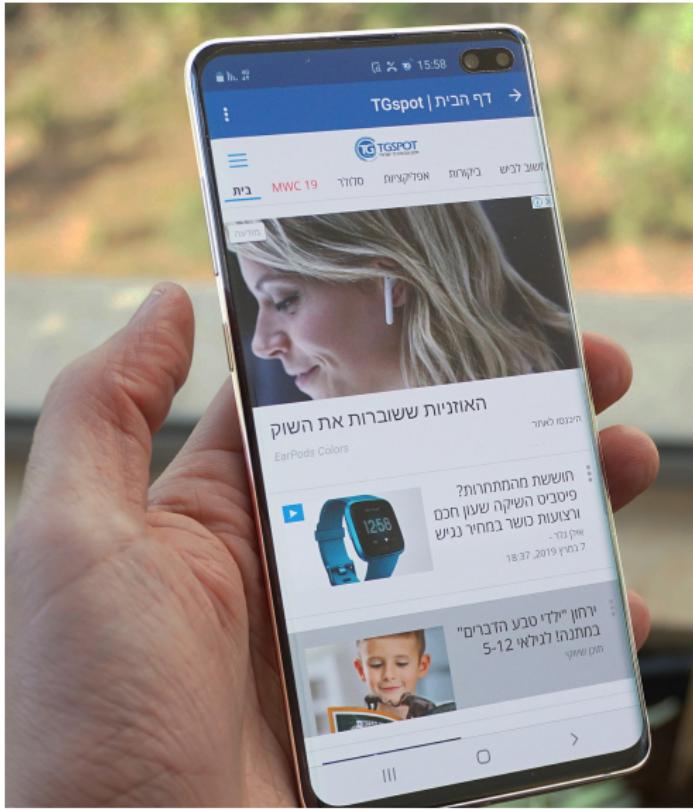
Qu'est ce qu'un ordinateur ?

Une machine capable de réaliser des calculs de façon automatisée.

L'élément principal, l'unité central, contient des composants pour :

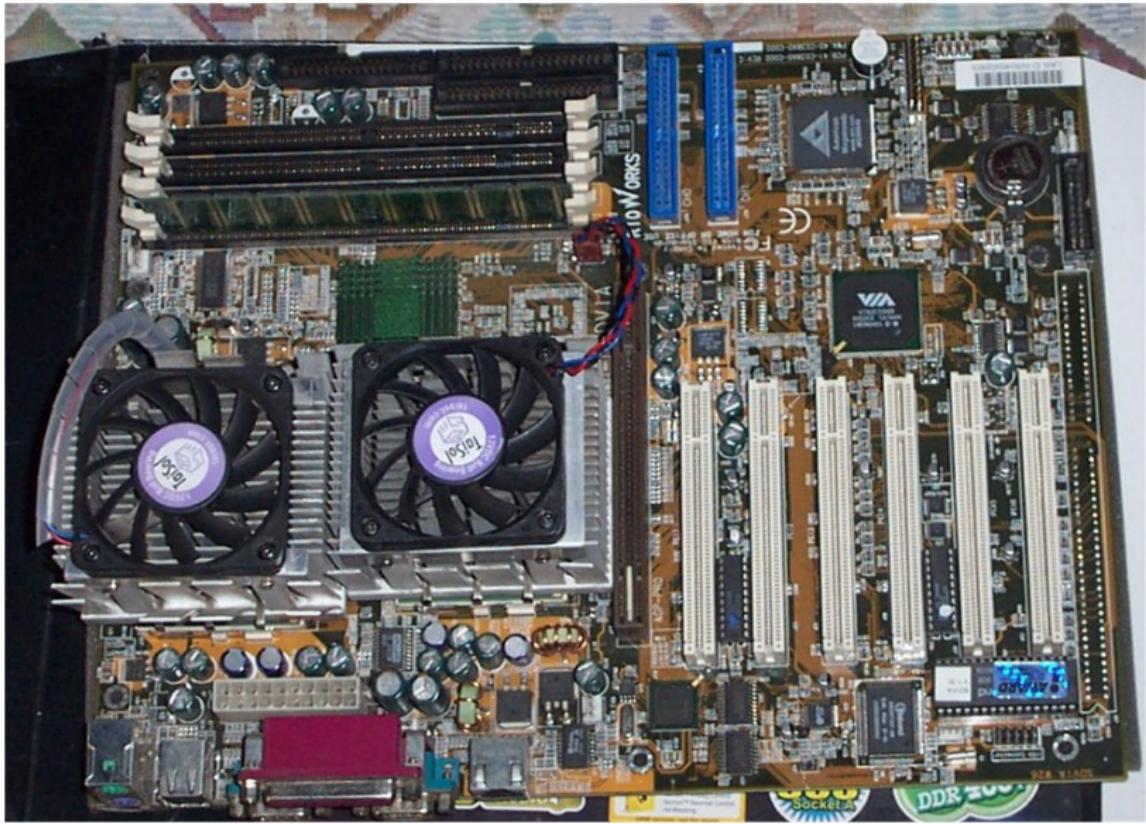
- Réaliser les calculs (un ou plusieurs processeurs)
- Garder les données et instructions courantes en mémoire (mémoire vive, RAM)
- Stocker de l'information (disque dur)
- Connecter des périphériques (clavier, souris, écrans, autres)

Un ordinateur ?



Composants d'un ordinateur

Vue de l'intérieur



Ce qu'on peut voir sur cette image



Le processeur (2)

Le processeur (CPU, *central processing unit*) est ce qui fait de l'ordinateur un ordinateur. C'est un composant électronique qui capable de réaliser des opérations arithmétiques, de suivre un flux d'instructions, et d'accéder à la mémoire vive.

Il existe plusieurs types de processeurs plus ou moins complexes. Une des indications de la puissance d'un processeur est sa vitesse, mesurée en Hertz (Hz), c'est-à-dire le nombre (de cycles) d'opérations qu'il peut réaliser par seconde.

Ce qu'on peut voir sur cette image



La mémoire vive

La mémoire vive (RAM, *random access memory*) est la mémoire de travail de l'ordinateur. C'est un composant électronique qui peut retenir et redonner des informations sous forme de nombres auxquels on accède par leur adresse mémoire. On peut le voir comme un ensemble de cases ou ranger des information. La mémoire vive est vidée lorsque l'ordinateur est éteint.

La caractéristique principale la mémoire vive est sa taille, c'est-à-dire la quantité d'information qu'elle peut contenir, exprimée en octets (o, Ko, Mo, Go, To, *byte* en anglais, B, KB, MB, GB, TB).

Ce qu'on **ne** peut **pas** voir sur cette image



Le disque dur

Le disque dur est un espace de stockage généralement plus grand que la mémoire vive, mais aussi plus lent, et qui est persistant : les informations stockées sur le disque dur ne disparaissent pas lorsqu'on éteint l'ordinateur.

Il existe différents types de disque dur (magnétiques, SSD) qui sont plus ou moins rapides. Comme pour la mémoire vive, la caractéristique principale est l'espace de stockage en octets.

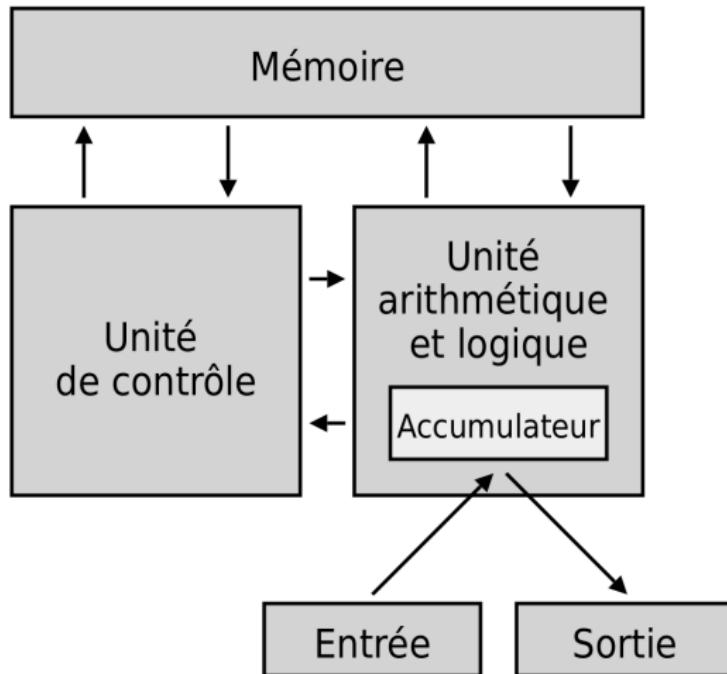
Ce qu'on **ne** peut **pas** voir sur cette image



Les autres périphériques

L'image représente la carte mère, c'est à dire le circuit qui permet de relier les différents composants. Celle-ci inclue aussi des connecteurs pour brancher d'autres périphériques comme des périphériques d'entrée (clavier, souris), de sortie (écran) ou autre.

De façon plus schématique



Architecture de von Neumann

Votre ordinateur

Question

Quelles sont les caractéristiques de votre ordinateurs ? (type et vitesse du processeur, taille mémoire, type et taille du disque dur)

Et de votre smartphone ?

Autres ordinateurs

Question

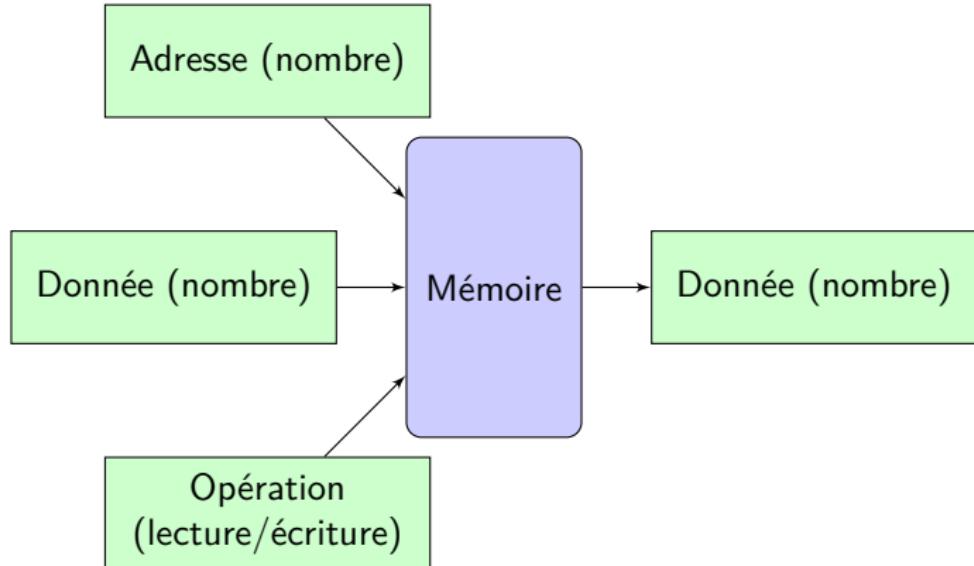
Quels autres appareils, qu'on n'appelle pas forcément des ordinateurs, sont aussi fondés sur cette architecture (et sont donc, finalement, des ordinateurs) ?

Retour sur la mémoire, et les octets

La mémoire vive



La mémoire vive



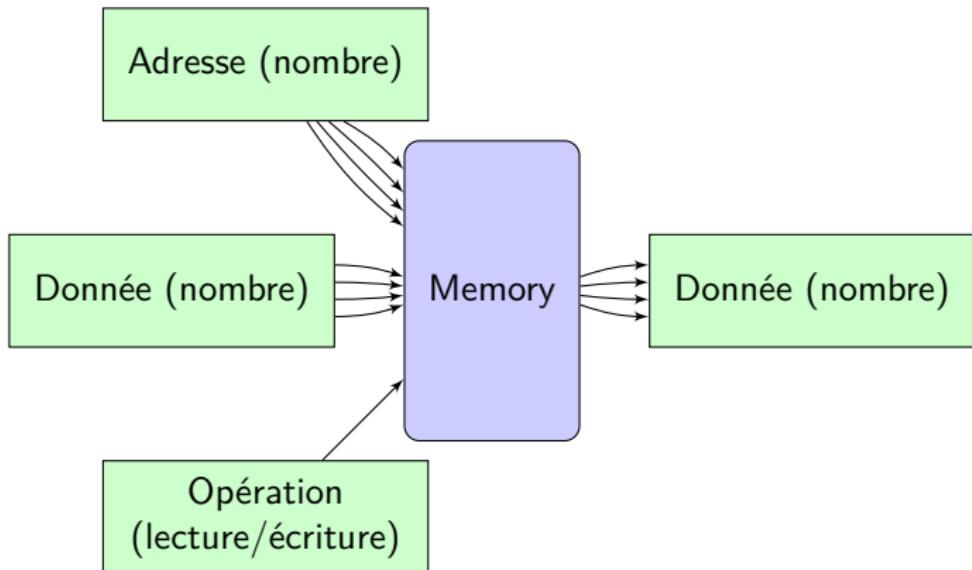
Pourquoi des octets et des bits

Un *bit* (*binary digit*) est un nombre qui peut prendre la valeur 0 ou 1.

Electroniquement, pour simplifier, cela correspond à un fil dans lequel soit il y a du courant, soit il n'y en a pas.

En combinant plusieurs bits, plusieurs fils, on peut représenter des nombres. Un octet est un ensemble de 8 bits.

La mémoire vive, dans un ordinateur représentant les nombres sur 4 bits



Représenter des nombres par des bits

Les nombres décimaux sont en base 10, c'est à dire que chaque chiffre correspond à une puissance de 10, par exemple :

$$1234 = 4 \times 10^0 + 3 \times 10^1 + 2 \times 10^2 + 1 \times 10^3$$

Les nombres binaires sont en base 2, c'est à dire que chaque chiffre (chaque bit) correspond à une puissance de 2, par exemple :

$$1011 = 1 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 1 \times 2^3 = 13$$

Représenter des nombres par des bits

Question

Combien de nombre peut on représenter avec 4 bits, 8 bits, 16 bits, 32 bits, 64 bits ?

Question

Que vaut 10111001 ?

Représenter des nombres par des bits

Pour les nombres négatifs

On réserve simplement un des bits, celui le plus à gauche, pour le signe.

Pour les nombres à virgule

On réserve certains bits pour la valeur (mantisse, dans un nombre à 32 bits, généralement 23 bits) et d'autres pour la position de la virgule (exposant, dans un nombre à 32 bits, généralement 8 bits) en plus du signe.

$$Nombre = -1^{Signe} \times Mantisse \times 10^{exposant-127}$$

$$-12,13 = -1^1 \times 1213 \times 10^{125-127}$$

1 0111 1101 000 0000 0000 0100 1101 0010

Représenter du texte par des bits

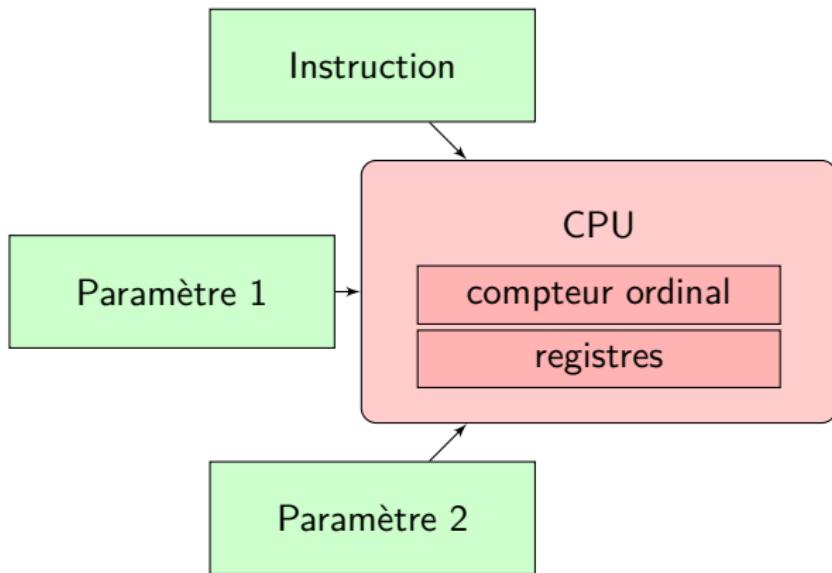
Pour représenter du texte, on utilise des standards d'encodage de caractères mettant en correspondance des nombres à des caractères. Le plus simple (7 bits) est ASCII (*american standard code for information interchange*, image ci-dessous), mais le plus courant maintenant est UTF8 (taille variable).

ASCII*Decimal*Hex*Binary Cross Reference															
ASC	Dec	Hex	Binary	ASC	Dec	Hex	Binary	ASC	Dec	Hex	Binary	ASC	Dec	Hex	Binary
!`	0	00	00000000	!`!	16	10	00001000	!`!	32	20	00100000	!`!	48	30	00110000
!`!	1	01	00000001	!`!	17	11	000010001	!`!	33	21	00100001	!`!	49	31	00110001
!`!	2	02	00000010	!`!	18	12	000010010	!`!	34	22	00100010	!`!	50	32	00110010
!`!	3	03	00000011	!`!	19	13	000010011	!`!	35	23	00100011	!`!	51	33	00110011
!`!	4	04	000000100	!`!	20	14	0000100100	!`!	36	24	001000100	!`!	52	34	00110100
!`!	5	05	000000101	!`!	21	15	0000100101	!`!	37	25	001000101	!`!	53	35	00110101
!`!	6	06	000000110	!`!	22	16	0000100110	!`!	38	26	001000110	!`!	54	36	00110110
!`!	7	07	000000111	!`!	23	17	0000100111	!`!	39	27	001000111	!`!	55	37	00110111
!`!	8	08	0000001000	!`!	24	18	0000100000	!`!	40	28	00101000	!`!	56	38	00111000
!`!	9	09	0000001001	!`!	25	19	0000100001	!`!	41	29	00101001	!`!	57	39	00111001
!`!	10	0A	0000001010	!`!	26	1a	0000100100	!`!	42	2a	00101010	!`!	58	3a	00111010
!`!	11	0B	0000001011	!`!	27	1b	0000100111	!`!	43	2b	00101011	!`!	59	3b	00111011
!`!	12	0C	0000001100	!`!	28	1c	0000110000	!`!	44	2c	00101100	!`!	60	3c	00111100
!`!	13	0D	0000001101	!`!	29	1d	0000110010	!`!	45	2d	00101101	!`!	61	3d	00111101
!`!	14	0E	0000001110	!`!	30	1e	0000111000	!`!	46	2e	00101110	!`!	62	3e	00111110
!`!	15	0F	0000001111	!`!	31	1f	0000111111	!`!	47	2f	00101111	!`!	63	3f	00111111
**--															
ASC	Dec	Hex	Binary	ASC	Dec	Hex	Binary	ASC	Dec	Hex	Binary	ASC	Dec	Hex	Binary
!`!	64	40	00000000	!`!	80	50	00100000	!`!	96	60	01100000	!`!	112	70	01110000
!`!	65	41	00000001	!`!	81	51	00100001	!`!	97	61	01100001	!`!	113	71	01110001
!`!	66	42	000000010	!`!	82	52	00100010	!`!	98	62	01100010	!`!	114	72	01110010
!`!	67	43	000000011	!`!	83	53	00100011	!`!	99	63	01100011	!`!	115	73	01110011
!`!	68	44	0000000100	!`!	84	54	001000100	!`!	100	64	011000100	!`!	116	74	01110100
!`!	69	45	0000000101	!`!	85	55	001000101	!`!	101	65	011000101	!`!	117	75	01110101
!`!	70	46	0000000110	!`!	86	56	001000110	!`!	102	66	011000110	!`!	118	76	01110110
!`!	71	47	0000000111	!`!	87	57	001000111	!`!	103	67	011000111	!`!	119	77	01110111
!`!	72	48	00000001000	!`!	88	58	0010000000	!`!	104	68	0110000000	!`!	120	78	01111000
!`!	73	49	00000001001	!`!	89	59	0010000001	!`!	105	69	0110000001	!`!	121	79	01111001
!`!	74	4a	00000001010	!`!	90	5a	0010001010	!`!	106	6a	0110001010	!`!	122	7a	01111010
!`!	75	4b	00000001011	!`!	91	5b	0010001011	!`!	107	6b	0110001011	!`!	123	7b	01111011
!`!	76	4c	00000001100	!`!	92	5c	0010001100	!`!	108	6c	0110001100	!`!	124	7c	01111100
!`!	77	4d	00000001101	!`!	93	5d	0010001101	!`!	109	6d	0110001101	!`!	125	7d	01111101
!`!	78	4e	00000001110	!`!	94	5e	0010001110	!`!	110	6e	0110001110	!`!	126	7e	01111110

Le processeur

À haut niveau

Le processeur est un composant électronique qui prend en entrée une commande (encodée par un nombre en binaire) et des paramètres, et qui réalise les commandes. Il inclut aussi un mécanisme de contrôle qui conserve l'adresse en mémoire de la prochaine instruction. Les résultats de ces opérations sont placés en mémoire ou dans des registres.



Les instructions d'un processeur

Les instructions qu'un processeur peut effectuer incluent des opérations arithmétiques, des opérations de contrôle (conditions, saut à une adresse mémoire pour la prochaine instruction, etc.), des opérations de déplacement de données, etc. Chaque type de processeur possède un jeu d'instruction (le langage machine).

MIPS32 Add Immediate Instruction

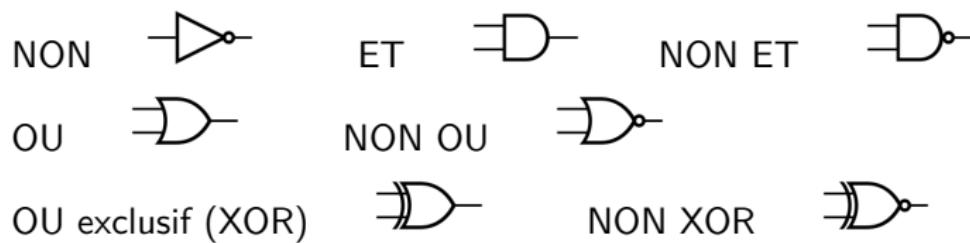
001000	00001	00010	0000000101011110
OP Code	Addr 1	Addr 2	Immediate value

Equivalent mnemonic: **addi \$r1, \$r2, 350**

Mais... comment ça marche ? (en simplifiant)

Portes logiques

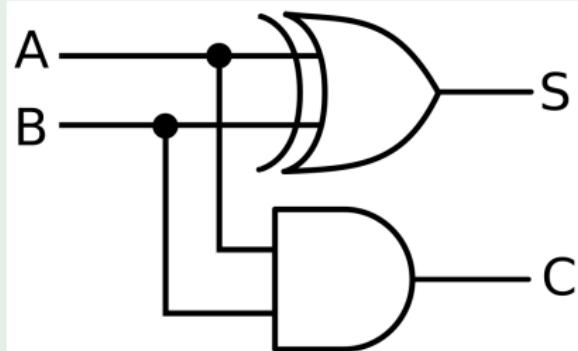
Les processeurs sont composés d'un ensemble de composants plus petits : des portes logiques. Celles-ci implémentent des opérations logiques élémentaires. En les combinant dans des circuits, on peut en faire des opérations plus complexes.



Circuit logique

Question

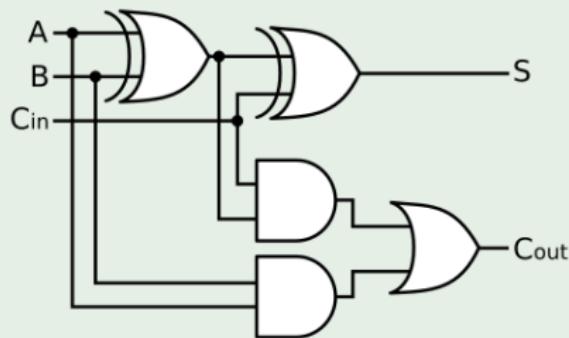
Que fait le circuit suivant ?



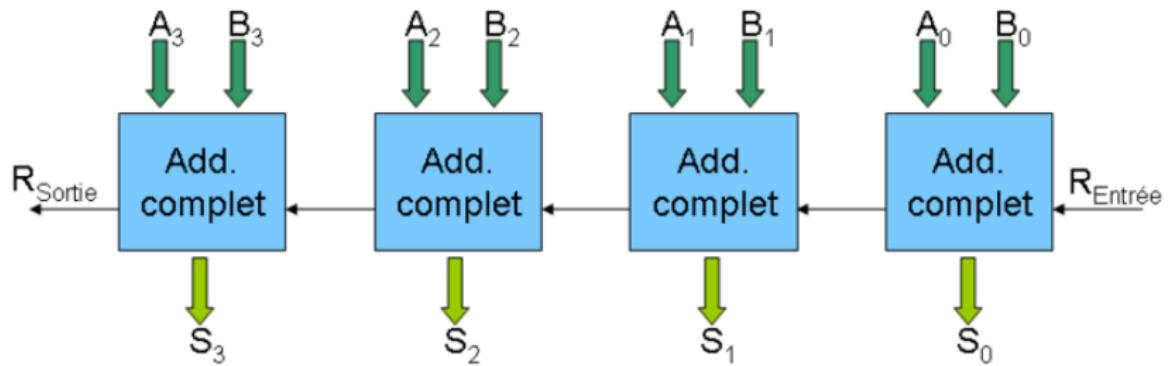
Circuit logique

Question

Que fait le circuit suivant ?



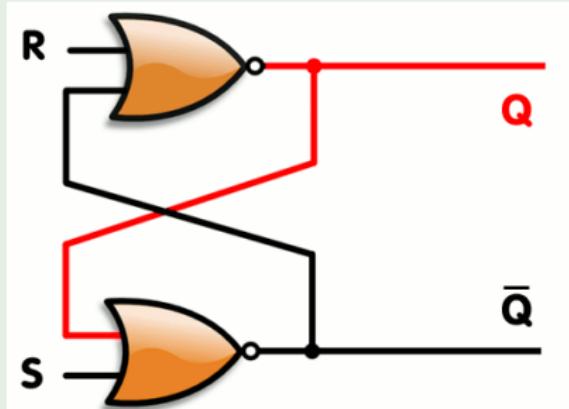
Additionner à 4 bits



Circuit logique

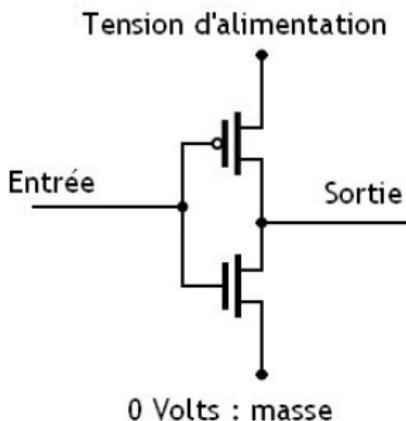
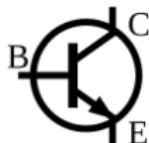
Question

Que fait le circuit suivant ?



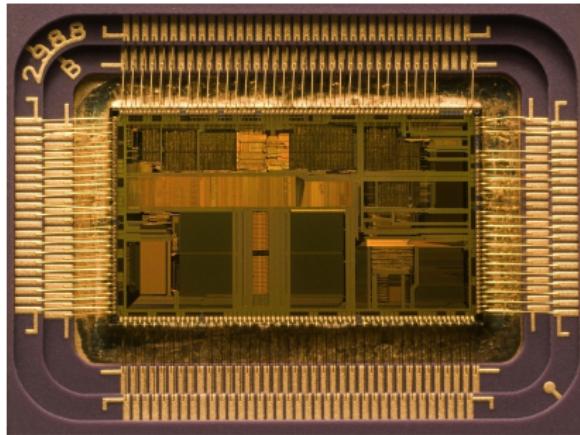
Mais vraiment, comment ça marche ?

Les portes logiques, au sein des processeurs, sont implémentées en utilisant des transistors. Un transistor, pour faire simple, se comporte comme un interrupteur électronique : sur la figure ci-contre, le courant passera entre C et E si il y a du courant sur B. Il est possible par exemple de construire une porte NON avec deux transistors.



Un processeur

Un processeur est donc composé d'un très grand nombre de transistors formant des portes logiques, elles même réalisant des circuits permettant de réaliser les opérations du jeu d'instructions. L'Intel 80486 ci-dessous en contenait 1 200 000 en 1989. Les processeurs d'aujourd'hui en contiennent plusieurs milliards.



Questions

Faire le quizz sur Arche.