

CHAPITRE	9	ARCHITECTURE DES ORDINATEURS	

Table des matières

1	Un ordinateur, quezaco ?	1
1.1	Un peu d'histoire...	2
1.2	Une couche matérielle...	4
1.3	Une couche logicielle...	5
2	Les systèmes d'exploitation	5
2.1	Une interface graphique révolutionnaire	6
2.2	Et en mode console ?	7
2.3	Découvrir linux	7
3	Comment la magie opère ?	8
3.1	Découvrir la logique combinatoire	8
3.2	Les portes logiques	8
3.2.1	La porte NON	8
3.2.2	La porte ET	8
3.2.3	La porte OU	9
3.2.4	Et ensuite... ?	9
3.3	Un demi additionneur !	9
3.4	Un vrai additionneur !	11
4	Architecture générale	12
5	En mode assemblage !	12

1 Un ordinateur, quezaco ?

À regarder la définition donnée dans Wikipédia, on se doute que la tâche n'est pas simple! La machine nous est maintenant familière, déclinée sous diverses formes (de bureau , portable, serveur,...) mais son invention est à l'échelle humaine très récente.

1.1 Un peu d'histoire...

En 1945, on voit apparaître le premier ordinateur entièrement électronique qui peut en principe, résoudre tous les problèmes de calculs numériques. C'est l'ENIAC qui pesait tout de même 27 tonnes!!!



Les premiers ordinateurs étaient des super calculateurs !

À partir de 1960, l' **architecture de von Neumann** est mise en œuvre et reste en vigueur encore de nos jours : l'usage de l'ordinateur reste professionnel ou industriel et se décline sous la forme de grosses machines (*mainframe*).

En 1964, la société **IBM** lance la série IBM/360 : son succès est fulgurant et va influencer toute l'industrie informatique.



FIGURE 1 – IBM 360

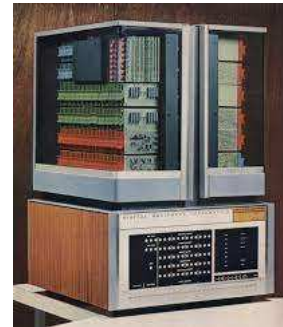


FIGURE 2 – Le PDP8

Puis un concurrent DEC(Digital Equipment Corporation) innove en créant des mini-ordinateurs pour conquérir de nouveaux marchés : le PDP-8 est vendu à 40000 exemplaires(18000 dollars tout de même...). Un événement majeur car il supporte **UNIX**.



La société **INTEL** est fondée en 1968 et crée en 1972 le premier microprocesseur Intel 4004(4bits) : suivront le 8088 en 1981 et le 386 en 1986.

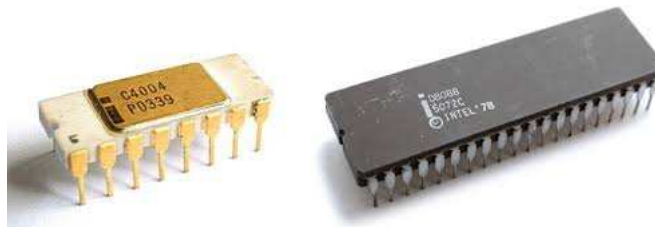


FIGURE 3 – Le 4004 et le 8088 de la société Intel

La décennie 70 marque l'avènement des ordinateurs personnels : le premier ordinateur en kit, l'**Altair 8800** coûte 400 dollars !



FIGURE 4 – L'altair 8800

Deux hommes proposent de développer un BASIC pour Altair : l'un deux est Bill Gates qui fonde la société **Microsoft**(1975).

Plusieurs sociétés se lancent alors dans la course : certaines éphémères et d'autres plus durables. En 1975 , Steve Wozniak et Steve Jobs créent la société **Apple** Computer.



L'Apple I sort en 1976 suivit de l'Apple II(couleur et son, 1200\$).



FIGURE 5 – Les premières versions des ordinateur de la société Apple

De nouvelles inventions comme la souris ou les interfaces graphiques améliorent encore l'ergonomie des ordinateurs : alors que le **Xerox Star** (1981) subit un échec commercial , l'**Apple Macintosh**(1984) connaît un incroyable succès.



FIGURE 6 – L'un est passé à la postérité, l'autre a disparu...

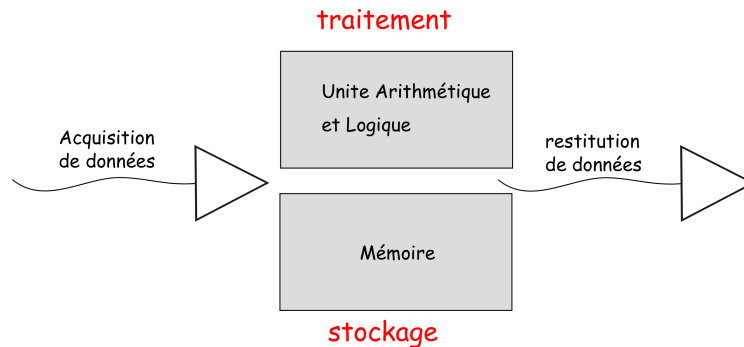
Son concurrent sort l'**IBM PC** en 1981 et fera la fortune de Microsoft par la multiplication ses clones(Compaq, Dell,...).

Au début des années 1990, l'arrivée du service Web va démocratiser l'usage des ordinateurs personnels dans la maison avec l'évolution que nous connaissons aujourd'hui : PC, portables, tablettes, smart-phone,...

1.2 Une couche matérielle...

Un ordinateur est une machine constituée d'éléments physiques comme la carte mère, des barrettes de mémoire, de disque, de fils et de pleins de composants électroniques¹.

Un ordinateur peut être décrit comme une machine de traitement de l'information obéissant à des programmes constitués de suites d'opérations arithmétiques et logiques : il est capable d'acquérir de l'information, de la stocker, de la transformer puis de la restituer sous une autre forme.



- l'**acquisition** se fait par l'intermédiaire de périphériques d'entrées comme le clavier, la souris, la caméra, le scanner,...
- le **stockage** se fait dans la mémoire (morte ou vive) ;
- le **traitement** est le rôle du processeur (UAL) ;
- la **restitution** se fait par les périphériques de sorties, l'écran ou l'imprimante en général mais pas que...

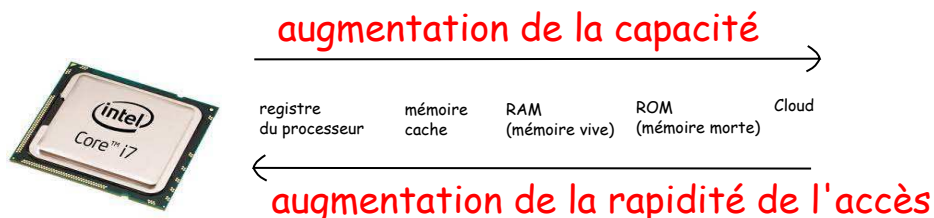


On appelle **Architecture de Von Neumann**, l'architecture classique des ordinateurs regroupant les composants cités ci-dessus.

Les données doivent circuler entre les différentes parties d'un ordinateur, notamment entre la mémoire vive et l'UAL : ces voies de transmission sont appelées les **bus**.

Dans un ordinateur, on trouve plusieurs mémoires qu'il est bon de connaître :

- les **registres** sont des mémoires internes au processeur(de 8 à 64 bits). Accès très rapide
- la **mémoire « cache »**, paramétrable, est un tampon entre les registres et la mémoire vive(quelques kilo octets). Accès rapide.
- la mémoire **vive** des barrettes (Giga Octet) ; Accès moins rapide.
- la mémoire **morte** comme le disque dur ou la clé USB (capacité quasi illimité...).Accès lent.



1. On verra plus tard comment l'assemblage d'éléments électroniques simples peut créer un additionneur !

1.3 Une couche logicielle...

Pour utiliser les capacités de la machine, les ressources du CPU, gérer les entrées/sorties et les utilisateurs, tout ordinateur a besoin d'un système d'exploitation. Ce point est particulièrement développé dans le prochain paragraphe de ce cours.

Puis l'installation de logiciels dédiés permet de répondre aux besoins les plus connus : divertissement, bureautique, DAO, CAO,...

2 Les systèmes d'exploitation

Une vidéo de présentation de mon prof préféré d'architecture de l'université de Bordeaux, Monsieur Raymond Namyst :

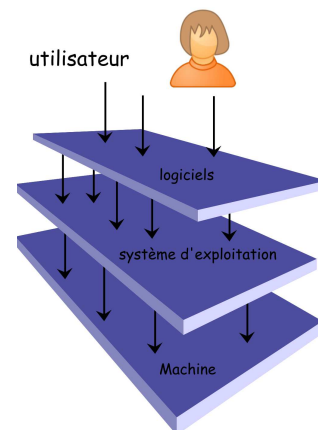
<https://www.youtube.com/watch?v=np5SDSeTdAY>



Le BIOS est le premier programme chargé en mémoire : il installe ensuite tous les éléments nécessaires au bon fonctionnement de l'ordinateur y compris le système d'exploitation.

Un système d'exploitation :

- gère et cache la complexité du matériel à l'utilisateur
- fournit une interface virtuelle de la machine (pas toujours graphique...)



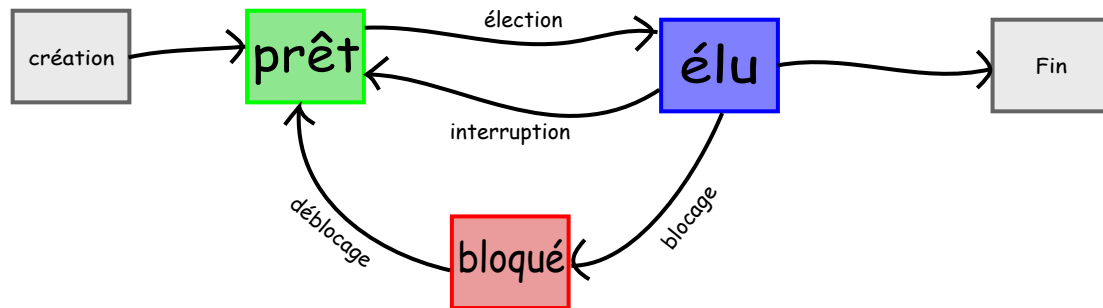
Il gère en outre le temps CPU (ordonnancement des tâches sur le processeur), l'allocation de la mémoire, les entrées et sorties, les utilisateurs, la consommation d'énergie...



FIGURE 7 – Principaux OS et leur place dans une architecture classique

Un système d'exploitation a pour fonction de charger les programmes depuis la mémoire de masse et de lancer leur exécution en créant des **processus** (process en anglais), de gérer l'ensemble des ressources, de traiter les interruptions ainsi que les entrées-sorties et enfin d'assurer la sécurité globale du système(notamment les utilisateurs...).

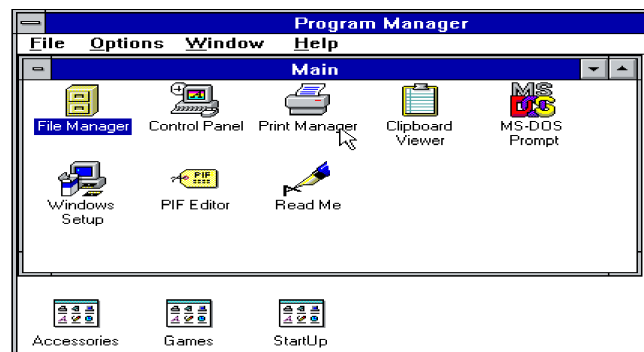
Tous les systèmes d'exploitation "modernes" (Linux, Windows, MacOS, Android, iOS...) sont capables de gérer l'exécution de plusieurs processus en même temps. Mais pour être précis, cela n'est pas un réel "en même temps", mais plutôt un "chacun son tour". Pour gérer ce "à chacun son tour", les systèmes d'exploitation attribuent des "états" au processus afin de les hiérarchiser.



2.1 Une interface graphique révolutionnaire

Ce qui a fait le succès de la société Microsoft, c'est certainement les interfaces graphiques des systèmes d'exploitation et des logiciels qui ont simplifié et démocratisé leur utilisation. Curseur, icône, fenêtre,... sont des éléments qui n'existaient pas dans les premiers ordinateurs. Les instructions étaient tapées dans **l'invite de commande**.

FIGURE 8 – Un des premiers OS : Windows 3.1



Quelques émulateurs des principaux OS à leur début. Vous noterez quelques petites différences...

Windows 1.0(1985) :<https://www.pcjs.org/software/pcx86/sys/windows/1.00/>

Windows 3.1(1992) :<https://www.pcjs.org/software/pcx86/sys/windows/3.10/>

Mac OS 1.0(1984) :<https://jamesfriend.com.au/pce-js/>

On peut toujours s'affranchir de l'interface graphique, en choisissant un démarrage du système en invite de commande seulement. Il faut savoir que les interfaces graphiques des systèmes d'exploitation consomment beaucoup de ressources et de temps processeur...

2.2 Et en mode console ?

Des machines équipées en linux (système d'exploitation libre) n'ont pas toujours un système d'exploitation en *mode interface graphique*. Ce choix peut être fait pour optimiser les performances de sa machine.

2.3 Découvrir linux

Vous utilisez tous les jours un système d'exploitation :

- ➔ votre téléphone Samsung utilise Android
- ➔ votre téléphone Apple iOS

Mais connaissez-vous linux ?

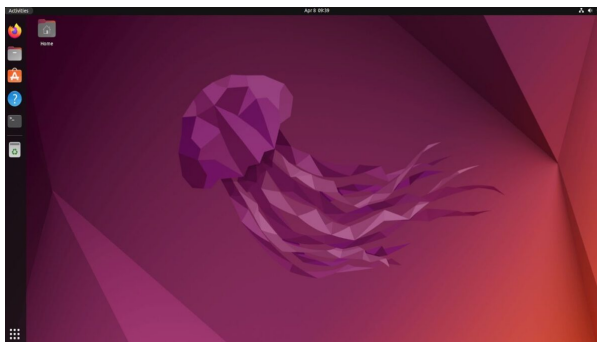


Linux est une famille de systèmes d'exploitation open source de type Unix fondés sur le noyau Linux créé en 1991 par Linus Torvalds.

Sauf choix personnel, Linux n'est pas installé sur les PC vendus dans le commerce. Il est pourtant totalement gratuit...

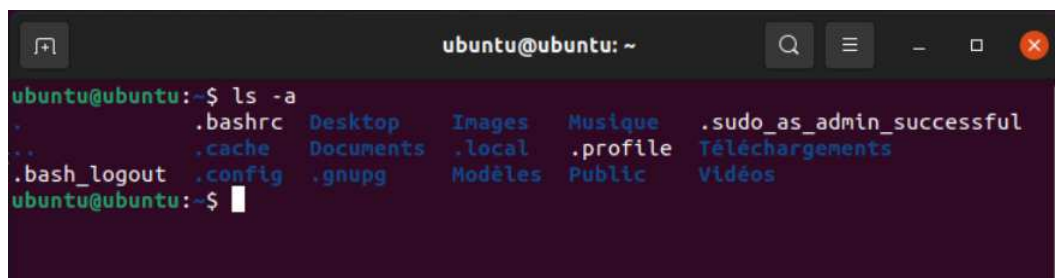
Il s'est spécialisé pour divers types de systèmes informatiques comme les **serveurs**, les **systèmes embarqués** ou encore les **téléphones portables**.

FIGURE 9 – Le bureau de Ubuntu présente peu de différences avec les SE microsoft...



Les utilisateurs confirmés de Linux utilise la ligne de commande pour interagir avec le système.

FIGURE 10 – Une console ou un terminal dans Ubuntu



3 Comment la magie opère ?

On a vu précédemment qu'un processeur traite des informations. Mais cela signifie quoi ? Quelles informations traite-t-il ? Il paraît que tout est binaire dans un ordinateur...

3.1 Découvrir la logique combinatoire

Les processeurs sont constitués d'une multitude de composants électroniques tellement petits qu'on ne peut même plus les visualiser !



Les transistors sont de minuscules composants électroniques qui agissent comme des interrupteurs de courants.

Un processeur Intel Core i7 peut contenir entre 730 et 2600 milliards de transistors, selon la génération et le modèle.

On n'insistera pas sur ce sujet pour ne s'intéresser qu'aux portes logiques qui permettent de définir toute une arithmétique à partir de signaux électriques !



Les portes logiques sont des composants électroniques constitués de transistors et de résistances.

3.2 Les portes logiques

Les portes logiques sont des composants qui ont une ou plusieurs entrées et qui produisent une sortie. Les entrées sont des tensions électriques :

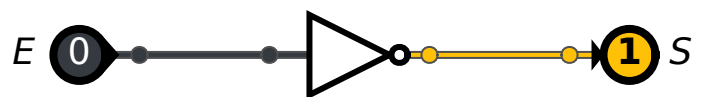
- ➔ une tension nulle (0V) à laquelle on associe la valeur binaire 0 ;
- ➔ une tension non nulle (5V par exemple) à laquelle on associe la valeur binaire 1.

La sortie, électrique aussi, a une valeur qui dépend de la porte considérée...

3.2.1 La porte NON

La sortie est toujours le contraire de l'entrée.

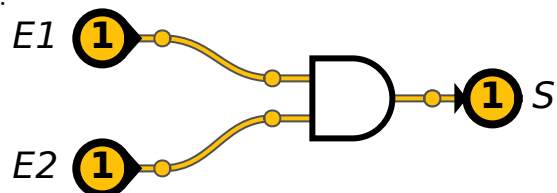
E	S
0	1
1	0



3.2.2 La porte ET

La sortie ne vaut 1 que si les deux entrées sont à 1.

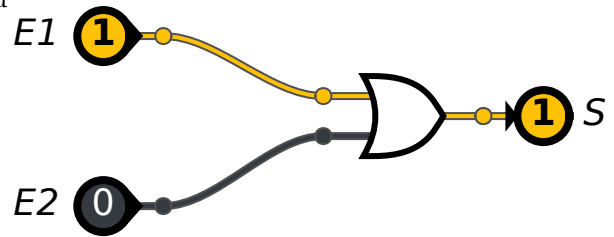
E1	E2	S
0	0	0
0	1	0
1	0	0
1	1	1



3.2.3 La porte OU

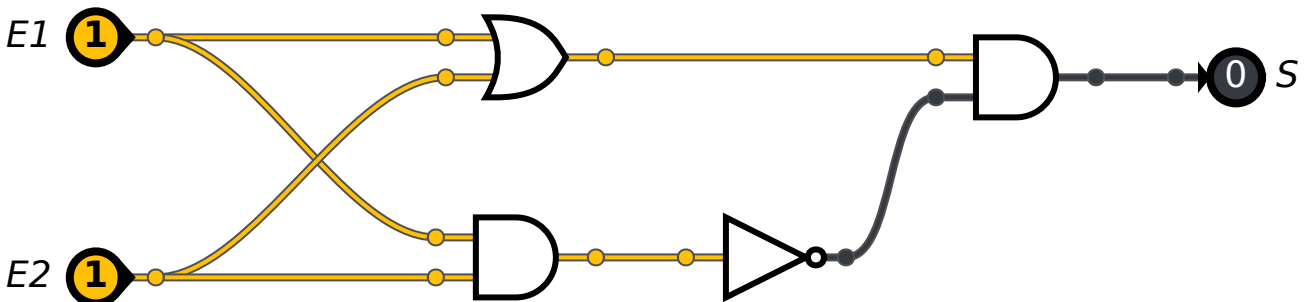
La sortie vaut 1 que si l'une des deux entrées au moins est à 1.

E1	E2	S
0	0	0
0	1	1
1	0	1
1	1	1



3.2.4 Et ensuite... ?

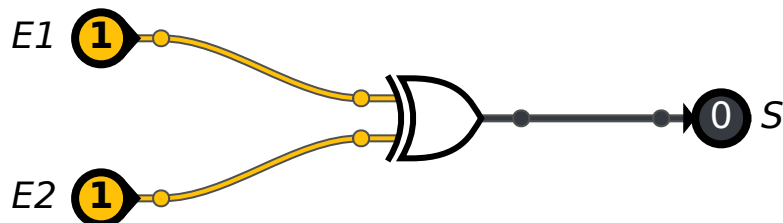
On peut alors définir de nouveaux circuits par la combinaison de ces portes :



Exercice n°1

Déterminer la table de vérité de ce circuit

Ce circuit définit une nouvelle porte logique appelée **X-OR** ou OU exclusif dont la représentation est donnée ci-dessous :



3.3 Un demi additionneur !

Le circuit précédent à livrer sa table de vérité! Avez-vous remarqué qu'il s'agit à la retenue près, d'une addition de deux bits ?



Quand on additionne deux bits il faut prendre en compte la retenue !

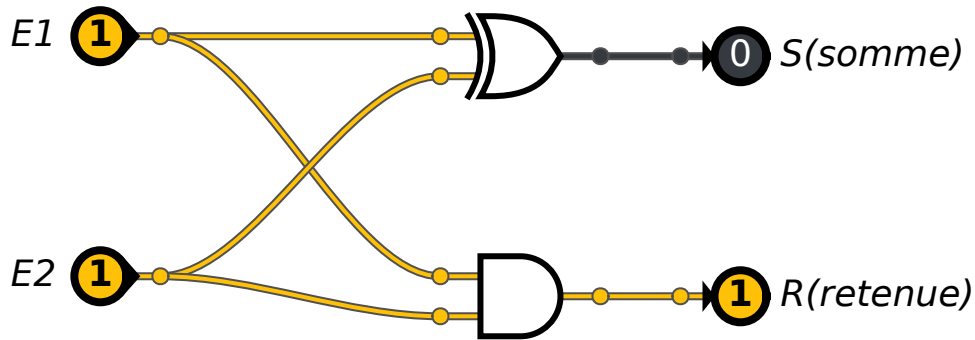
Ainsi, la table de vérité de l'addition de deux bits serait-elle :

E1	E2	S	R
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Exercice n°2

- 1 Quelle porte logique permet de déterminer R en fonction des entrées E1 et E2 ?
- 2 Quelle porte logique permet de déterminer S en fonction des entrées E1 et E2 ?

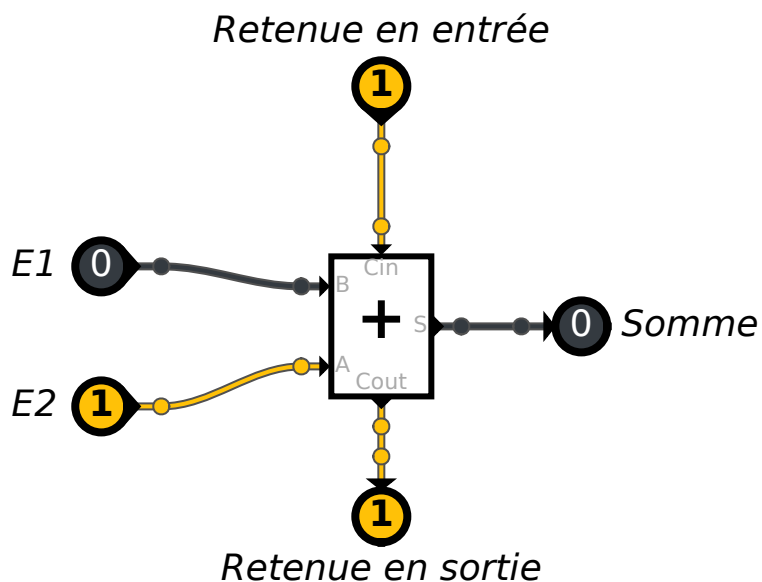
Donc la combinaison de deux portes logiques élémentaires permet l'addition de deux bits :



Mais dans la pratique, ce n'est pas deux bits que nous ajoutons mais plutôt trois : en effet en posant l'opération nécessaire à l'addition de deux octets, la retenue est prise en compte :

$$\begin{array}{r} 1\ 1\ 0\ 0\ 1\ 0\ 1\ 1\ (203) \\ +\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ (138) \\ \hline 1\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ \text{retenues} \\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ (341) \end{array}$$

C'est-à-dire qu'il faut trois bits en entrées pour deux sorties possibles ! Et l'additionneur complet se présente alors sous la forme :



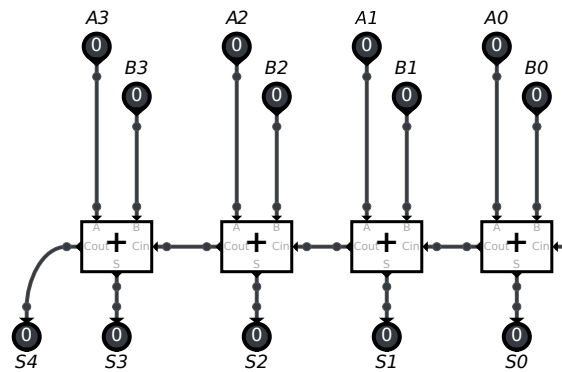
3.4 Un vrai additionneur !

Montrons sur un exemple, comment additionner deux nombres binaires de 4 bits.

Exercice n°3

On souhaite assembler des portes logiques qui vont permettre l'addition de deux demi-octets A et B. Les valeurs de chaque bits sont appelées A3,...,A0 pour le premier et B3,...,B0 pour le second.

- 1 Aller sur le site [ici](#)
- 2 Réaliser le circuit ci-dessous.
- 3 Que fait la somme 0111 + 1100 ?



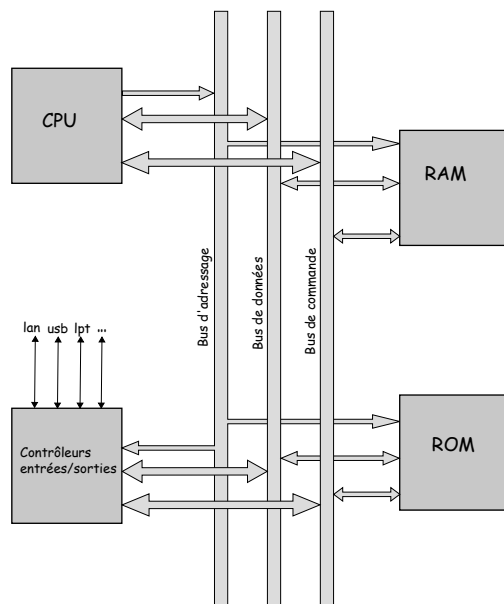
On pourrait compléter cette chaîne d'additionneurs pour des plus grands nombres, le principe restant le même.



La dernière retenue constitue un **overflow** qui n'est pas pris en compte dans l'architecture 4 bits proposée ci-dessus...

4 Architecture générale

L'architecture d'un ordinateur désigne l'agencement de ses différents composants et en particulier des éléments vitaux :



Vous découvrirez l'intérêt des trois **bus** dans le travail sur les microprocesseurs.

5 En mode assemblage !

Alors que nous, humain, programmons en Python ou en C, le processeur possède son propre langage, **totalemment incompréhensible** pour nous.



L'instruction 0010011010101110011010111 est constituée d'un champ **code** qui indique ce que doit faire le processeur sur les nombres contenus dans le champ **opérande** !

Un processeur donné est capable d'exécuter un certain nombre d'opérations de base, celles pour lesquelles il dispose d'un circuit électronique qui les réalise. L'ensemble des instructions exécutables directement par le microprocesseur (instructions machines) constitue ce que l'on appelle le **langage machine** du processeur. Chaque instruction machine correspond à une configuration électronique binaire composée principalement de 2 parties :

- ◇ le champ « code opération(opcode) » qui indique le type de traitement à réaliser ;
- ◇ le champ « opérande » indique la nature des données sur lesquelles l'opération précédente agit.



Les opérations fondamentales sont de trois sortes : les instructions arithmétiques (calculs sur des nombres, entiers ou flottants), les instructions de transfert (CPU vers mémoire et vice versa) et les instructions de saut(alternative ou boucle).

Programmer en langage machine est extrêmement difficile (très longue suite de 0 et de 1). Pour pallier à cette difficulté, les informaticiens ont remplacé les codes binaires abscons par des symboles mnémoniques (plus facile à retenir qu'une suite de 1 et de 0), cela donne **l'assembleur**.