SUPPORT DE COURS B1-DEV

Introduction à la programmation

Généralités

Date	Révision
16/09/2018	Publication initiale BTS SIO
11/08/2019	Corrections et améliorations (exemple + simplifications)
03/10/2019	Remise au standard DR2019
03/08/2021	Amélioration des titres et ajout d'explications sur compilation versus interprétation

TABLE DES MATIÈRES

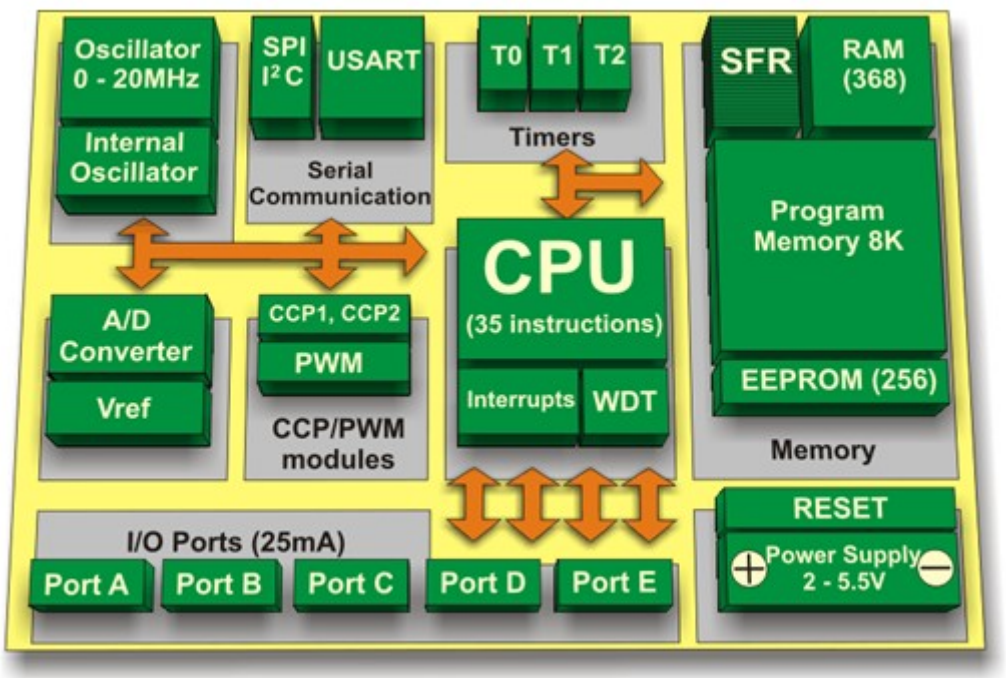
Icon made by Freepik from www.flaticon.com

1 ARCHITECTURE MATÉRIELLE

Pour pouvoir fonctionner, un ordinateur nécessite au minimum 4 composants :

- Une **mémoire** pour contenir le programme, les données et d'autres informations
- Un **processeur** capable de lire les informations dans la mémoire et les traiter
- Une **horloge** pour faire avancer le programme : sans horloge, le processeur resterait sur la même instruction, indéfiniment
- Un **bus** (et même plusieurs bus) pour permettre à tous les éléments et périphériques de communiquer ensemble. Il s'agit d'un ensemble de pistes (de fils électriques) qui relient la mémoire et le processeur (bus principal) ou d'autres bus pour le clavier, les cartes vidéos, etc.

Il faut également ajouter des **ports** d'entrées et sorties (ils fonctionnent de manière similaire aux ports marins) pour relier l'écran, le clavier et les autres périphériques.



Architecture d'un contrôleur PIC

Pour résumer, l'horloge cadence les vitesses du CPU, de la mémoire et des ports de communications. Elle permet d'incrémenter la lecture des instructions pour passer aux suivantes.

A noter : on différencie les architectures RISC (Reducted Instruction Set Computing) et CISC (Complex Instruction Set Computing).

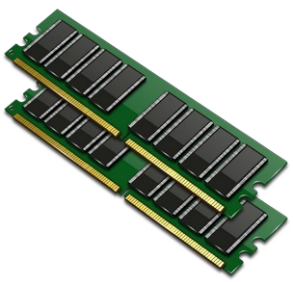
Dans l'architecture RISC, le processeur exécute une instruction par cycle d'horloge. Dans l'architecture CISC, certaines instructions sont plus longues et le processeur bloque l'avancement du programme pour les terminer.

1.1 La mémoire

Elle est généralement représentée sous la forme d'un tableau linéaire de case :

Adr-0000	*Adr-0004*	*Adr-0008*	*Adr-000B*						*Adr-FFFE*
data-1	data-2	data-3							data-X

Chaque case représente un octet et actuellement, les ordinateurs bas de gamme disposent de 4 gigaoctets de mémoire : 4 x 10^9 octets.

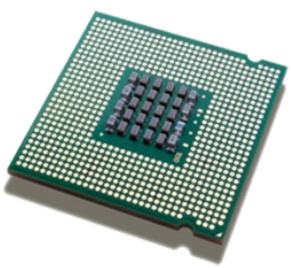


Cette mémoire est modifiable à volonté et très rapidement. Par contre, elle s'efface immédiatement lors d'une coupure d'électricité.

Toutes les instructions de vos programmes seront contenues dans la mémoire, pendant leur exécution.

1.2 Le processeur

C'est le composant qui lit dans la mémoire, les instructions de vos programmes : il peut effectuer des calculs (addition, multiplication…) mais aussi des tests et des comparaisons. Enfin, il peut écrire des informations dans la mémoire, qu'il pourra traiter plus tard.



Toutefois, il ne connaît qu'un seul langage, le sien : on l'appelle le langage machine, car il n'est constitué que de 0 et de 1.

Les processeurs 64 bits ont un bus de 64 pistes, soit 64 fils...

1.3 L'horloge

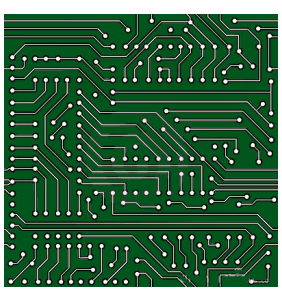
C'est un composant original qui utilise les propriétés du quartz et de l'électricité. En faisant passer un courant dans le quartz, celui-ci se met à vibrer et émettre des impulsions électriques. E

n utilisant un compteur, il devient possible de générer des tensions électriques alternatives régulières très précises.

C'est ce qui permet au processeur de passer d'une instruction à une autre : sans horloge, le processeur lirait toujours la même zone mémoire.

1.4 Les bus

Ce ne sont pas des composants mais les fils qui relient les composants entre eux. On distingue les bus internes (PCI, IDE, SATA) et les bus externes (USB, série, HDMI). Pour ce cours, on se rappellera surtout qu'il y a un bus pour les données et un bus pour les adresses : le processeur indique une adresse sur le bus d'adresse et récupère les données qui s'y trouve.



On trouve les architectures 32 bits et 64 bits : cela représente la taille du bus.

2 FONCTIONNEMENT DYNAMIQUE

Le fonctionnement d'un ordinateur implique que les instructions changent dans le temps (horloge), que chaque donnée soit accessible dans une zone (emplacement mémoire), puisse circuler depuis et vers le CPU (bus) pour que l'on puisse y appliquer un traitement (processeur).

Pour cela, les premiers programmes ont été écrits en langage machine, c'est-à-dire, dans un langage compréhensible par la machine (binaire) mais facilement traduisible en langage humain.

2.1 EXEMPLE DE L'ASSEMBLEUR 68000

L'assembleur est un langage humain dans lequel, chaque mot correspond à une instruction du microprocesseur.

Ce langage est rarement utilisé de nos jours, car on utilise des programmes appelés "compilateurs" pour traduire des instructions en de multiples instructions qui s'enchaînent.

Le microprocesseur Motorola 68000 est un composant simple et facile à comprendre, voilà pourquoi je l'utilise dans ce cours.

2.1.1 FORMAT D'UNE LIGNE D'INSTRUCTION

La plupart du temps, les instructions comportent 3 champs :

Mnémonique	Source	Destination
MOVE.B	#$65	D0

L'exemple ci-dessus, déplace le nombre hexadécimal 65 dans le registre du processeur D zéro.

Dans le détail, le .B signifie que la taille de la donnée à déplacer est un octet (8 bits). Attention, le registre contient 32 bits (4 octets) dont ici, seul le dernier octet est modifié, pour déplacer la valeur hexadécimale 65 en écrasant tous les bits du registre il faut écrire :

MOVE.L #$65, D0

Dans ce langage, **#** signifie valeur (si on ne précise pas le #, le processeur ira lire la mémoire de l'ordinateur à l'adresse indiquée), **$** indique qu'il s'agit du codage en hexadécimal.

2.1.2 SOUSTRACTION EN ASSEMBLEUR 68000

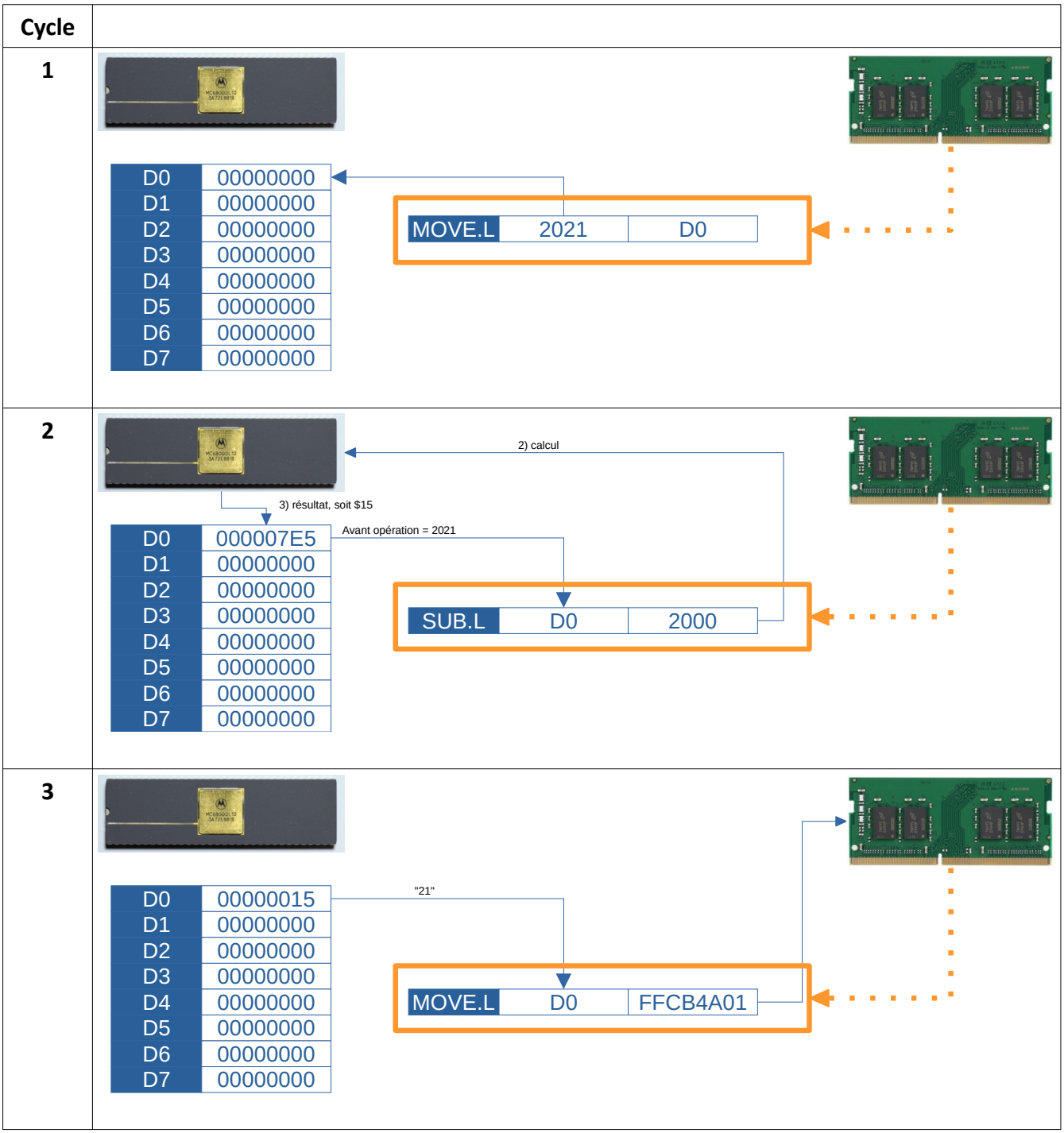
Ainsi, pour connaître votre âge, on pourrait écrire le programme suivant :

```
MOVE.L #2021, D0
SUB.L D0, 2000
MOVE.L D0, $FFCB4A01
```

Ce programme se lit "mettre 2021 dans D0, soustraire 2000 à D0 (et ranger le résultat dans D0) et placer le contenu de D0 dans la mémoire, à l'adresse hexadécimale FFCB4A01.

2.2 Représentation graphique du programme

Voici maintenant le fonctionnement du programme (stocké en mémoire vive) :



Chaque temps étant un cycle d'horloge, on peut considérer qu'un processeur exécute plusieurs milliards d'opérations par secondes (même si dans la réalité, certaines opérations prennent plusieurs cycles d'horloge).