

SOMMAIRE

CHAPITRE I : LE PROCESSEUR.....	Page 2
CHAPITRE II : LA MÉMOIRE VIVE.....	Page 09

LE PROCESSEUR

Qu'est-ce qu'un processeur

Le processeur (CPU: Central Processing Unit) est un circuit électronique cadencée au rythme d'une horloge interne, c'est-à-dire un élément qui envoie des impulsions (que l'on appelle top). A chaque top d'horloge les éléments de l'ordinateur accomplissent une action. La vitesse de cette horloge (le nombre de battements par secondes) s'exprime en Mégahertz, ainsi un ordinateur à 200Mhz a donc une horloge envoyant 200,000,000 de battements par seconde (un cristal de quartz soumis à un courant électrique permet d'envoyer des impulsions à une fréquence précise).

A chaque top d'horloge (pour les instructions simples) le processeur :

lit l'instruction à exécuter en mémoire

effectue l'instruction

passé à l'instruction suivante

Le processeur est en fait constitué:

d'une unité de commande qui lit les instructions et les décode

d'une unité de traitement (UAL - unité arithmétique et logique) qui exécute les instructions.

Lorsque tous les éléments d'un processeur sont regroupés sur une même puce, on parle alors de microprocesseur.

A quoi ressemble une instruction?

Les instructions (opération que le processeur doit accomplir) sont stockées dans la mémoire principale. Une instruction est composée de deux champs:

le code opération: c'est l'action que le processeur doit accomplir

le code opérande: c'est les paramètres de l'action. Le code opérande dépend de l'opération, cela peut être une donnée ou bien une adresse d'un emplacement mémoire



Une instruction peut être codée sur un nombre d'octets variant de 1 à 4 suivant le type de données.

les registres

Lorsque le processeur traite des données (lorsqu'il exécute des instructions) le processeur stocke temporairement les données dans de petites mémoires de 8, 16 ou 32Ko (qui ont l'avantage d'être très rapides) que l'on appelle registres. Suivant le type de processeur le nombre de registres peut varier entre une dizaine et plusieurs centaines.

Les registres les plus importants sont:

le registre accumulateur: il permet de stocker les résultats des opérations arithmétiques et logiques

le registre tampon: il permet de stocker temporairement une des opérandes

le registre d'état: il permet de stocker les indicateurs

le registre instruction: il contient l'instruction en cours de traitement

le compteur ordinal: il contient l'adresse de la prochaine instruction à traiter

le registre tampon: il permet de stocker temporairement une donnée provenant de la mémoire

les signaux de commande

Les signaux de commande sont des signaux électriques qui permettent au processeur de communiquer avec le reste du système (le signal Read/Write - lecture/écriture - permet notamment de signaler à la mémoire qu'il désire lire ou écrire une information)

Le microprocesseur

Qu'est-ce qu'un microprocesseur ?

Le premier microprocesseur (Intel 4004) a été inventé en 1971. Depuis, la puissance des microprocesseurs augmente exponentiellement. Quels sont donc ces petits morceaux de silicium qui dirigent nos ordinateurs?



Le processeur (CPU) est le cerveau de l'ordinateur, c'est lui qui coordonne le reste des éléments, il se charge des calculs, bref il exécute les instructions qui ont été programmées. Toutes ces opérations permettent de manipuler des informations [numériques](#), c'est-à-dire des informations codées sous forme [binaire](#). Pour réaliser ces traitements, les microprocesseurs utilisent de "petits interrupteurs" utilisant l'effet transistor découvert en 1947 par *John Barden* et *Walter Brittan* qui reçurent le prix Nobel l'année suivante pour cette découverte. Il existe plusieurs millions de ces transistors sur un seul processeur !



Les éléments principaux d'un microprocesseur sont:

Une horloge qui rythme le processeur. Entre deux tops d'horloge le processeur effectue une action. Une instruction nécessite une ou plusieurs actions du processeur. Ainsi plus l'horloge a une fréquence élevée, plus le processeur effectue d'instructions par seconde (l'unité retenue pour caractériser le nombre d'instructions traitées par unité de temps est généralement le *MIPS, Millions d'instruction par seconde*). Par exemple un ordinateur ayant une fréquence de 100 Mhz effectue 100 000 000 d'instructions par seconde

Une unité de gestion des bus qui gère les flux d'informations entrant et sortant

Une unité d'instruction qui lit les données arrivant, les décode puis les envoie à l'unité d'exécution.

Une unité d'exécution qui accomplit les tâches que lui a donné l'unité d'instruction.



Le processeur travaille en fait grâce à un nombre très limité de fonctions (ET logique, Ou logique, addition ...), celles-ci sont directement câblées sur les circuits électroniques. Il est impossible de mettre toutes les instructions sur un processeur car celui-ci est limité par la taille de la gravure, ainsi pour mettre plus d'instructions il faudrait un processeur ayant une très grande surface, or le processeur est constitué de silicium et le silicium coûte cher, d'autre part il chauffe beaucoup. Le processeur traite donc les informations compliquées à l'aide d'instructions simples.

Le parallélisme

Le parallélisme consiste à exécuter simultanément sur des processeurs différents des instructions relatives à un même programme. Cela se traduit par le découpage d'un programme en plusieurs processus qui seront traités par des processeurs différents dans le but de gagner en temps d'exécution. Cela nécessite toutefois une communication entre les différents processus. C'est le même principe de fonctionnement que dans une entreprise: le travail est divisé en petits processus traités par des services différents et qui ne servent à rien si la communication entre les services ne fonctionne pas (ce qui est généralement le cas dans les entreprises...).

Le pipelining

Le *pipelining* est un principe simple à comprendre. Il permet de mettre à disposition du microprocesseur les instructions qu'il va devoir effectuer. Les instructions font la "file" (d'ougrave; le nom de "pipeline") dans la mémoire cache. Ainsi, pendant que le microprocesseur exécute une instruction, la suivante est mise à sa disposition.

Le pipelining permet donc en quelque sorte d'empiéter la fin d'une instruction sur le début de la suivante. En effet, une instruction se déroule selon trois phases :

Récupération de la donnée (notée *F* pour *Fetch*) : recherche en mémoire de l'instruction, mise à jour du compteur ordinal

Décodage (noté *D* pour *Decode*) : obtention des calculs à faire, des éléments de données concernés

Exécution (notée *E* pour *Execute*) : calcul à proprement dit

Dans une structure non pipelinée, il faut 9 temps pour faire 3 instructions : F1-D1-E1-F2-D2-E2-F3-D3-E3 (dans l'ordre chronologique)

Dans une structure pipelinée idéale, on réalise plusieurs phases en même temps, ceci étant possible en mettant les résultats des différentes phases dans des registres tampon :

F1 - D1+F2 - E1+D2+F3 - E2+D3 - E3 (dans l'ordre chronologique)

Il suffit ainsi de 5 temps uniquement. Ceci n'est cependant pas toujours possible, pour des questions de dépendance d'une instruction vis-à-vis du résultat de la précédente...

l'architecture CISC

L'architecture CISC (*Complex Instruction Set Computer*, ce qui signifie "ordinateur avec jeu d'instructions complexes") est utilisée par tous les processeurs de type x86, c'est-à-dire les processeurs fabriqués par Intel, AMD, Cyrix, ... Les processeurs basés sur l'architecture CISC peuvent traiter des instructions complexes, qui sont directement câblées sur leurs circuits électroniques, c'est-à-dire que certaines instructions difficiles à créer à partir des instructions de base sont directement imprimées sur le silicium de la puce afin de gagner en rapidité d'exécution sur ces commandes. L'inconvénient de ce type d'architecture provient justement du fait que des fonctions supplémentaires sont imprimées sur le silicium, d'où un coût élevé.

D'autre part, les instructions sont de longueurs variables et peuvent parfois prendre plus d'un cycle d'horloge ce qui les rend lentes à l'exécution étant donné qu'un processeur basé sur l'architecture CISC ne peut traiter qu'une instruction à la fois!

l'architecture RISC

Contrairement à l'architecture CISC, un processeur utilisant la technologie RISC (*Reduced Instruction Set Computer*, dont la traduction est "ordinateur à jeu d'instructions réduit") n'a pas de fonctions supplémentaires câblées. Cela impose donc des programmes ayant des instructions simples interprétables par le processeur. Cela se traduit par une programmation plus difficile et un compilateur plus puissant. Cependant vous vous dîtes qu'il peut exister des instructions qui ne peuvent pas être décrites à partir des instructions simples... En fait ces instructions sont tellement peu nombreuses qu'il est possible de les câbler directement sur le circuit imprimé sans alourdir de manière dramatique leur fabrication.

L'avantage d'une telle architecture est bien évidemment le coût réduit au niveau de la fabrication des processeurs l'utilisant. De plus, les instructions, étant simples, sont exécutées en un cycle d'horloge, ce qui rend l'exécution des programmes plus rapides qu'avec des processeurs basés sur une architecture CISC.

De plus, de tels processeurs sont capables de traiter plusieurs instructions simultanément en les traitant en parallèle.

CISC ou RISC

A comparer les spécificités des deux types d'architecture on pourrait conclure que les processeurs basé sur une architecture de type RISC sont les plus utilisés... Cela n'est malheureusement pas le cas... En effet les ordinateurs construits autour d'une architecture RISC nécessitent une quantité de mémoire plus importante que les ordinateurs de type CISC .

LES PROCESSEURS INTEL CORE

Les **processeurs Intel Core** ne date pas d'hier ! La **première génération** de ces processeurs est en effet apparue en 2008. Depuis, les Intel Core se sont considérablement améliorés : de nouvelles technologies, de meilleures performances, des graphismes largement optimisés... **Aujourd'hui en 2016, nous en sommes à la sixième génération Skylake.** Pour mieux comprendre les **caractéristiques des Intel Core** et leurs **évolutions** au fil des générations, ce cours vous propose une petite rétrospective autour des processeurs phares d'Intel : les **Core i3, i5 et i7.**

Intel Core de 1^{ère} génération « Nehalem » (2008)

Les puces Intel Core i3, i5 et i7 de **première génération** sont sorties de 2008 à 2010 et sont **gravés en 45 nm** (nanomètres).

Je comprends déjà rien ! Ça veut dire quoi « gravé en 45nm, 32nm ou 22nm » ?

Cela fait référence à la **finesse de gravure** et les différentes **générations de processeurs** n'ont pas forcément la même finesse de gravure (mesurée en micromètres (µm) ou nanomètres (nm)).

A quoi ça sert la finesse de gravure d'un processeur ?

Diminuer la finesse de gravure permet d'abord de **baisser les coûts de fabrication**, c'est donc utile pour le fabricant. Mais cela permet également de **diminuer la quantité de chaleur** produite par le processeur, de **diminuer sa consommation d'énergie** et de pouvoir le **monter plus haut en fréquence** (le rendre plus performant). Et ça, c'est bon pour nous, les consommateurs !

Les **Core de première génération** ont encore un processeur et une carte graphique séparée. Les Core des générations suivantes embarqueront un processeur et un circuit graphique au sein d'une même puce.

Intel Core de 2nd génération « Sandy Bridge » (2011)

En 2011, Intel lance la **seconde génération** des processeurs Core gravés en 32nm (vous savez ce que ça veut dire maintenant ! :D). Leur nom de code est **Sandy Bridge**.

Les **processeurs Core** de deuxième génération sont très différents de ceux de la génération précédente puisqu'ils intègrent **le processeur et la carte graphique au sein d'une même puce**, ces derniers étant habituellement séparés sur une carte mère. Cette intégration permet d'avoir des échanges beaucoup plus rapides entre le CPU et le contrôleur graphique. Les améliorations sont notamment visibles lors de la lecture de vidéos HD et dans les jeux vidéos. Globalement, les benchmarks ont montré que les **processeurs de 2nd génération** étaient 42% plus rapides que ceux de 1^{ère} génération.

Attention, ces **processeurs** ne sont pas compatibles avec les cartes mères qui accueilleraient les Core de première génération. Afin de s'assurer que les utilisateurs ne se trompent pas en essayant de mettre un **processeur Sandy Bridge** dans une carte mère de génération précédente, le **nombre de broches** sur les nouveaux processeurs Intel est passé à **1155**. Comme les **processeurs Sandy Bridge** ont 1155 broches, il est alors physiquement impossible d'en mettre un sur une carte mère supportant les **processeurs Core de première génération** (qui ont 1366 ou 1156 broches).

Comme toutes les anciennes cartes mères étaient incompatibles, Intel a sorti de nouvelles cartes mères basées sur les chipsets H61, H67, P67 et Z68 compatibles avec les **processeurs Sandy Bridge**.

Intel Core de 3^{ème} génération « Ivy Bridge » (2012)

Sortis en 2012, les **processeurs Intel Core de troisième génération** sont gravés en 22 nm. Leur nom de code est **Ivy Bridge**.

Combinés avec les nouveaux **chipsets Intel Série 7**, cette association permet de bénéficier du **support du PCIe 3.0** (une version plus rapide du connecteur qui sert à brancher des cartes d'extension sur sa carte mère) et de l'**USB 3.0**, et permettre à la mémoire vive de monter à une **vitesse de 1600 MHz** nativement.

La bonne nouvelle, c'est que les **processeurs Ivy Bridge** fonctionnent avec beaucoup

de **cartes mères au socket 1155** de génération précédente (même si une mise à jour du BIOS est généralement nécessaire). De plus, les **processeurs Sandy Bridge** de génération précédente sont également utilisables sur la plupart des nouveaux **chipsets Série 7**, ce qui permet à beaucoup d'utilisateurs de mettre à jour leur configuration facilement, que ce soit au niveau de la carte mère ou du cpu.

Revenons si vous le voulez bien à nos **processeurs Core Ivy Bridge**. Alors que la **gamme de fréquence** de ces **processeurs** n'est pas beaucoup plus élevée que celle de la génération précédente Sandy Bridge, les principales améliorations se trouvent dans la **carte graphique intégrée** (Intel HD 2500 et HD 4000) qui a été nettement améliorée et dans la **réduction de la production thermique** (grâce à une finesse de gravure réduite). Avec une consommation d'énergie significativement réduite, les **processeurs Ivy Bridge** sont principalement conçus pour les **Ultrabooks** – les ordinateurs ultra-portables – et les tablettes x86 car ils permettent d'allonger la durée de vie des batteries.

Pour utiliser tout le potentiel des nouveaux **processeurs Ivy Bridge**, Intel a sorti des cartes mères basées sur les chipsets Z68, Z77 et H77.

Intel Core de 4^{ème} génération « Haswell » (2013)

Les **processeurs Haswell** arrivent en 2013 avec une toute **nouvelle architecture** (tout en conservant le procédé de fabrication en 22nm) et un nouveau socket, le **LGA 1150**. L'objectif numéro 1 de ces processeurs de quatrième génération est de réduire leur **consommation énergétique**. Mission réussie, les processeurs Haswell se montrent en effet moins énergivore même si les gains de performance par rapport à la génération précédente sont timides et moins flagrants qu'autrefois.

Pour finir, les processeurs Haswell intègrent un nouveau processeur intégré, **Intel HD Graphics 4600**, plus performant que l'ancien Intel HD 4000.

Intel Core de 5^{ème} génération « Broadwell » (juin 2015)

Après de longs mois de retard, les **processeurs Broadwell** débarquent enfin pour les PC de bureau en juin 2015. Malheureusement, les **Core i7-5775C** et **Core i5-5675C** ne resteront pas dans les annales des utilisateurs de PC, non pas pour qu'ils sont décevants (au contraire) mais au cas d'un mauvais timing : les processeurs de la génération suivante (Skylake) arrivent en effet en août 2015 avec une toute nouvelle architecture. Les utilisateurs ont donc logiquement préféré attendre deux mois la sortie de ces **processeurs Skylake** pour bénéficier des dernières avancées technologiques.

Revenons aux **processeurs Broadwell**. Ce sont les premières puces de bureau en socket LGA 1150 gravées en 14 nm. Pour rappel, une finesse de gravure supérieure permet une consommation moindre, un échauffement mesuré et une capacité d'overclocking (une montée en fréquence) supérieure. La **consommation énergétique** se révèle impressionnante, le Core i7 5775C consomme en effet moins qu'un Core i3 Haswell ! Mais là où les processeurs Broadwell frappent (très) fort, c'est avec leur nouveau iGPU, le processeur graphique intégré.

Intel Core de 6^{ème} génération « Skylake» (août 2015)

Les **processeurs Skylake** sont sortis en août 2015, introduisant une nouvelle micro-architecture et un nouveau socket : le **LGA 1151**. La plate-forme Skylake introduit également la **mémoire DDR4** sur toutes les plates-formes grand public.

Que valent ces **processeurs de sixième génération** alors ? Eh bien leurs performances sont en hausse de 10% en moyenne par rapport à la génération précédente Broadwell et d'environ 23% par rapport aux processeurs Sandy Bridge. Du côté de la consommation, on peut dire que ce n'est pas leur point fort, ils se révèlent un peu plus gourmands que ceux de la génération précédente. Le passage au 14 nm n'a pas eu l'effet escompté de ce côté-là.

LA MEMOIRE VIVE

Rôle de la mémoire vive (RAM)

La **mémoire vive**, généralement appelée **RAM** (*Random Access Memory*, traduisez *mémoire à accès aléatoire*), est la mémoire principale du système, c'est-à-dire qu'il s'agit d'un espace permettant de stocker de manière temporaire des données lors de l'exécution d'un programme.

En effet le stockage de données dans la mémoire vive est temporaire, contrairement au stockage de données sur une mémoire de masse telle que le disque dur (mémoire avec laquelle les novices la confondent généralement), car elle permet uniquement de stocker des données tant qu'elle est alimentée électriquement. Ainsi, à chaque fois que l'ordinateur est éteint, toutes les données présentes en mémoire sont irrémédiablement effacées.

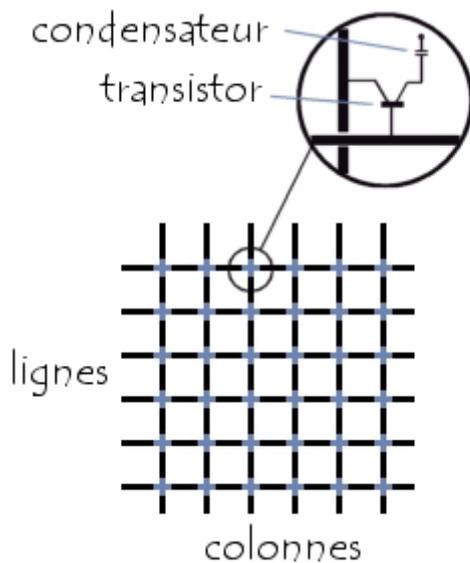
La **mémoire morte**, appelée **ROM** pour *Read Only Memory* (traduisez *mémoire en lecture seule*) est un type de mémoire permettant de conserver les informations qui y sont contenues même lorsque la mémoire n'est plus alimentée électriquement. A la base ce type de mémoire ne peut être accédée qu'en lecture. Toutefois il est désormais possible d'enregistrer des informations dans certaines mémoires de type *ROM*.

Fonctionnement de la mémoire vive

La mémoire vive est constituée de centaines de milliers de petits condensateurs emmagasinant des charges. Lorsqu'il est chargé, l'état logique du condensateur est égal à 1, dans le cas contraire il est à 0, ce qui signifie que chaque condensateur représente un [bit](#) de la mémoire.

Etant donné que les condensateurs se déchargent, il faut constamment les recharger (le terme exact est *rafraîchir*) à un intervalle de temps régulier appelé **cycle de rafraîchissement** (d'une durée d'environ 15 nanosecondes (ns) pour une mémoire DRAM).

Chaque condensateur est couplé à un transistor (de type *MOS*) permettant de "récupérer" ou de modifier l'état du condensateur. Ces transistors sont rangés sous forme de tableau (matrice), c'est-à-dire que l'on accède à une "case mémoire" (aussi appelée *point mémoire*) par une ligne et une colonne.



Chaque point mémoire est donc caractérisé par une adresse, correspondant à un numéro de ligne et un numéro de colonne. Or cet accès n'est pas instantané et s'effectue pendant un délai appelé **temps de latence**. Par conséquent l'accès à une donnée en mémoire dure un temps égal au temps de cycle auquel il faut ajouter le temps de latence.

Ainsi, pour une mémoire de type DRAM, le temps d'accès est de 60 nanosecondes (35ns de délai de cycle et 25ns de temps de latence). Sur un ordinateur, le temps de cycle correspond à l'inverse de la fréquence de l'horloge, par exemple pour un ordinateur cadencé à 200Mhz, le temps de cycle est de 5ns ($1/(200 \cdot 10^6)$).

Par conséquent un ordinateur ayant une fréquence élevée et utilisant des mémoires dont le temps d'accès est beaucoup plus long que le temps de cycle du processeur doit effectuer des **cycles d'attente** (en anglais *wait state*) pour accéder à la mémoire. Dans le cas d'un ordinateur cadencé à 200Mhz utilisant des mémoires de types DRAM (dont le temps d'accès est de 60ns), il y a 11 cycles d'attente pour un cycle de transfert. Les performances de l'ordinateur sont d'autant diminuées qu'il y a de cycles d'attentes, il est donc conseillé d'utiliser des mémoires plus rapides.

La correction d'erreurs

Certaines mémoires possèdent des mécanismes permettant de pallier les erreurs afin de garantir l'intégrité des données qu'elles contiennent. Ce type de mémoire est généralement utilisé sur des systèmes travaillant sur des données critiques, c'est la raison pour laquelle on trouve ce type de mémoire dans les serveurs.

Bit de parité

Les barrettes avec bit de parité permettent de s'assurer que les données contenues dans la mémoire sont bien celles que l'on désire. Pour ce faire, un des bits de chaque octet stocké en mémoire sert à conserver la somme des bits de données. Le bit de parité vaut 0 lorsque la somme des bits de données est impaire et 1 dans le cas contraire.

De cette façon les barrettes avec bit de parité permettent de vérifier l'intégrité des données mais ne permettent pas de corriger les erreurs. De plus pour 8 Mo de mémoire, seulement 7 serviront à stocker des données, dans la mesure où le dernier mégaoctet conservera les bits de parité.

Barrettes ECC

Les barrettes de mémoire ECC (*Error Correction Coding*) sont des mémoires possédant plusieurs bits dédiés à la correction d'erreur (on les appelle ainsi *bits de contrôle*). Ces barrettes, utilisées principalement dans les serveurs, permettent de détecter les erreurs et de les corriger.