Architecture d'un ordinateur

Philippe Notez (philippe.notez@inmc.fr)

Sommaire

Introduction

L'électronique et l'informatique ont profondément modifié notre société. C'est certainement la révolution industrielle la plus rapide de l'histoire de l'humanité. Aujourd'hui, les systèmes embarqués sont omniprésents dans notre vie quotidienne et nous emmènent vers un monde de plus en plus connecté, avec ses avantages et ses inconvénients...

Ce document présente les composants principaux d'un ordinateur, en espérant toujours être le plus clair et précis possible. Malgré tout le soin apporté à la rédaction, l'auteur vous remercie de bien vouloir le contacter si vous relevez la moindre erreur ou omission, et vous souhaite une agréable lecture.

Les deux piliers

Nous avons vu comment :

- *calculer*, grâce à la logique combinatoire
- *mémoriser*, grâce à la logique séquentielle

Ces logiques constituent les piliers de l'électronique numérique et posent les bases de l'informatique moderne.

En effet, les deux composants les plus importants d'un ordinateur sont :

- le *processeur*, qui effectue les calculs
- la *mémoire*, qui contient les données

Modèle de von Neumann

L’informatique est la science du traitement de l’information, de manière automatique.
Un ordinateur est une machine qui gère ce traitement.

Pour traiter une information, il faut :

- un périphérique d’entrée (clavier, souris, capteur, etc…)
- un dispositif de mémorisation, puisque le périphérique d’entrée envoie les données au fil de l’eau (au fur et à mesure qu’elles arrivent)
- une unité de traitement, effectuant les calculs, les comparaisons, etc…
- un périphérique de sortie. Les premiers ordinateurs envoyaient le résultat du traitement vers une imprimante. Aujourd’hui, le résultat s’affiche généralement sur un écran

Ainsi, le modèle de von Neumann, élaboré en 1946 et toujours utilisé de nos jours, définit :

- une unité de traitement (*Unité Arithmétique et Logique*)
- une mémoire principale
- des Entrées/Sorties (clavier, souris, écran, etc…)
- une unité de contrôle, permettant la synchronisation des différentes unités

La mémoire principale étant composée :

- d’une mémoire morte, en lecture seule, contenant le programme (le traitement) à exécuter par l'UAL. Les données enregistrées dans cette mémoire sont conservées, même sans alimentation électrique
- d’une mémoire vive, en lecture et écriture, contenant les informations à traiter. Les données enregistrées dans cette mémoire sont perdues lors d'une coupure de courant

Les disques durs, DVD, etc… sont des *périphériques* de stockage et sont considérés comme des mémoires secondaires (mémoires de masse). Aujourd'hui, la mémoire morte n'est plus utilisée (sauf pour le BIOS), les programmes étant enregistrés en mémoire de masse et chargées en mémoire vive pour y être exécutés.

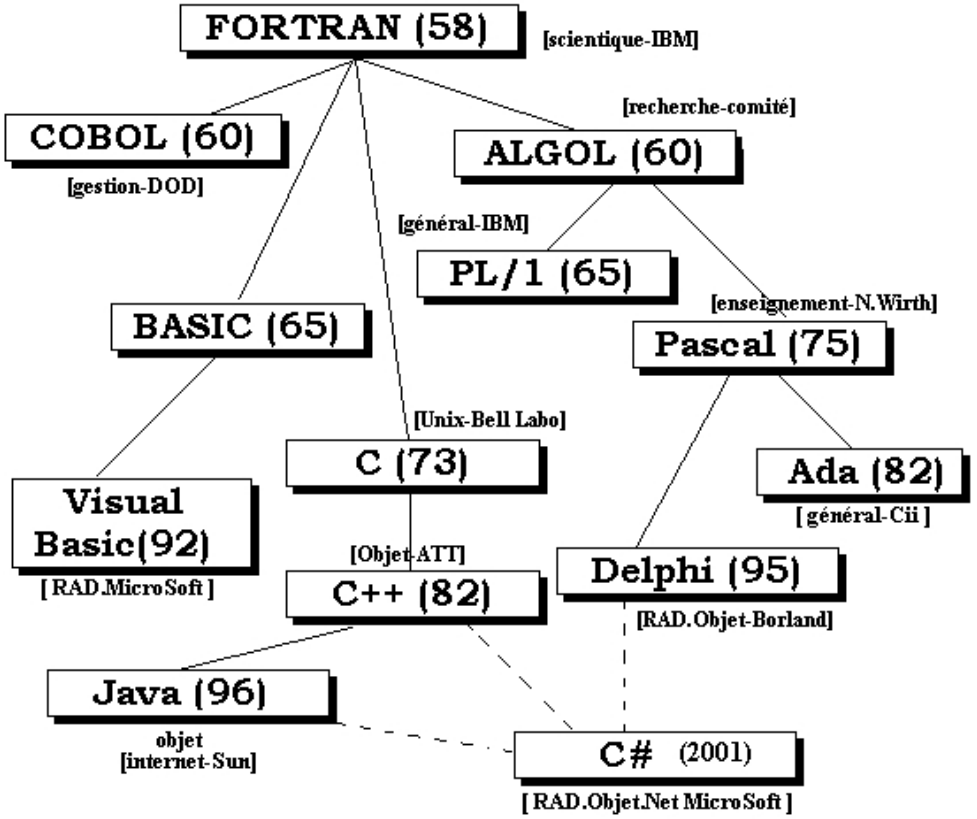
Les quatre générations

On distingue généralement quatre générations d’ordinateurs, en fonction de l’évolution du matériel, utilisant pour les traitements :

- des tubes à vide, entre 1945 et 1955 environ
- des transistors, entre 1955 et 1965 environ
- des circuits intégrés, entre 1965 et 1975 environ
- des microprocesseurs, à partir de 1975 environ

Nous ne parlerons que de la quatrième génération. Aujourd’hui, on utilise le terme *microprocesseur* car l'ensemble des composants nécessaires à son fonctionnement sont regroupés dans un seul circuit intégré.

D’autres classements existent, en fonction par exemple de l’évolution des langages de programmation :

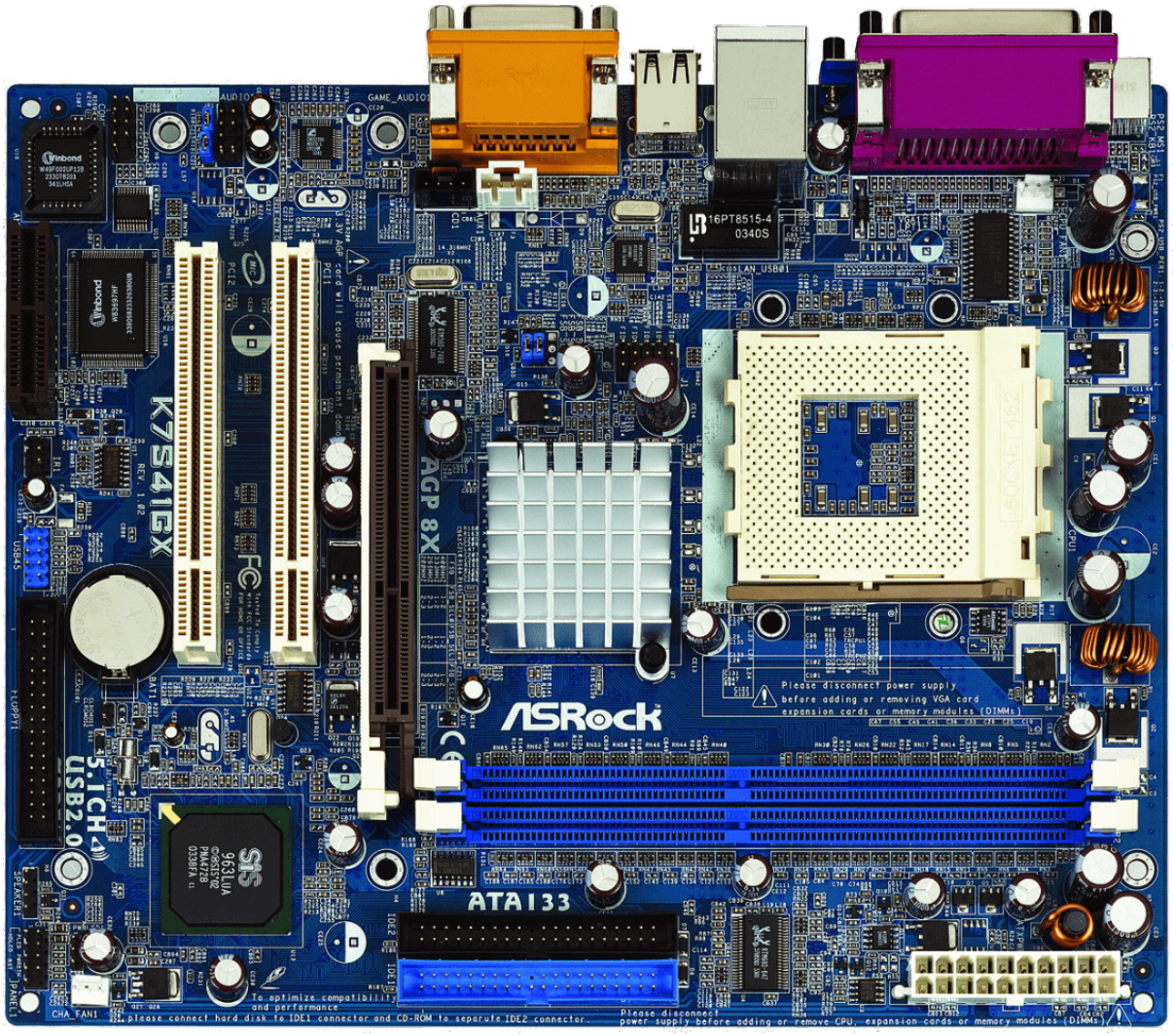


Composants et périphériques

Dans un ordinateur, on distingue les composants et les périphériques. Parmi les composants, on trouve la carte mère, le microprocesseur, la mémoire, la carte graphique, etc… Les composants sont liés physiquement (sans câble) à la carte mère, ou connectés via des bus d'extension, et sont choisis en fonction des caractéristiques de cette dernière (par exemple, une carte mère n'accepte que certains microprocesseurs en fonction de son socket). Les périphériques sont indépendants et peuvent donc (en théorie) être branchés sur tous les ordinateurs, grâce à des câbles.

Carte mère

La carte mère est la carte principale d'un ordinateur. Elle supporte le microprocesseur, la mémoire, ainsi que les différents bus d'extension, permettant par exemple la connexion d'une carte graphique. Elle gère la communication entre le microprocesseur et les autres composants / périphériques grâce à un chipset.



On peut voir le *socket* (grand connecteur blanc), en-dessous les deux connecteurs bleus pour la mémoire, à gauche du *socket* le *chipset* (ou plutôt son radiateur), et à gauche du *chipset* les connecteurs d'extension.

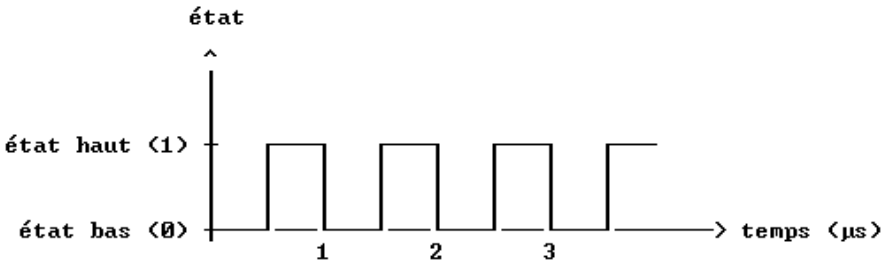
Microprocesseur : introduction

Le microprocesseur, appelé également CPU (*Central Processing Unit*) est le cerveau de l'ordinateur. Il contient l'UAL (*Unité Arithmétique et Logique*) ou ALU (*Arithmetic and Logic Unit*) en logique combinatoire. Depuis le premier microprocesseur, l'Intel 4004 (1971), les performances ont fortement évolué, grâce notamment à :

- la fréquence d'horloge, de quelques centaines de kHz à plusieurs GHz aujourd'hui
- la miniaturisation, permettant l'augmentation des capacités de traitement

Un microprocesseur est capable de faire des calculs, mais aussi des comparaisons (égal, inférieur, supérieur, etc...). Avec un jeu de quelques dizaines d'instructions, il est capable d'exécuter des programmes complexes.

Un microprocesseur utilise une horloge avec une fréquence définie (un nombre de cycles identiques par seconde, où l'horloge passe de l'état bas à l'état haut, et inversement). Avec une horloge dont la fréquence est de 1 MHz (10^6 cycles par seconde), la durée d'un cycle est de 1 µs ($1 / 10^6$ ou 1×10^{-6} ou 0,000001 seconde) :



Les instructions les plus simples nécessitent quelques cycles d'horloge pour s'exécuter (parfois même un seul cycle), alors que les plus complexes peuvent exiger plusieurs dizaines de cycles. La fréquence d'horloge est donc primordiale : plus la fréquence est élevée, plus le microprocesseur effectue les traitements rapidement. Mais l'augmentation de la fréquence pose un problème : plus le microprocesseur va vite, plus il chauffe ! On corrige ce problème en diminuant, par exemple, la tension d'alimentation. Néanmoins, nous atteignons aujourd'hui des limites physiques difficilement franchissables. On s'oriente donc depuis plusieurs années vers des microprocesseurs multi-cœurs, permettant d'augmenter encore les capacités de traitement. Des recherches sont également en cours pour utiliser des ordinateurs d'un autre type, comme par exemple l'ordinateur quantique. A suivre…

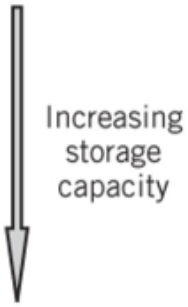
Microprocesseur : mémoire cache

La mémoire principale est constituée de mémoire vive dynamique nécessitant un rafraîchissement périodique pour conserver ses données. Ce rafraîchissement prend du temps, et pendant ce temps, la mémoire principale n'est pas disponible. Pour éviter au microprocesseur d'attendre la disponibilité de la mémoire principale, on va ajouter de la mémoire vive statique (*Static Random Access Memory*), que l'on appelle *mémoire cache*, entre le microprocesseur et la mémoire principale. La mémoire vive statique ne nécessite pas de rafraîchissement, elle est donc accessible plus rapidement que la mémoire principale, mais à taille égale nécessite plus d'espace à l'intérieur du circuit intégré car elle utilise plus de transistors.

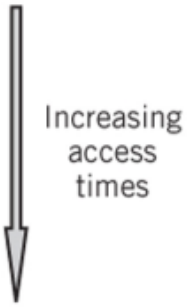
Le microprocesseur va vérifier si la donnée dont il a besoin se trouve en mémoire cache. Si ce n'est pas le cas, il va la récupérer en mémoire principale. Une fois le traitement effectué, il va l'enregistrer en mémoire cache. Ainsi, la prochaine fois qu'il en aura besoin, il va gagner du temps puisqu'elle se trouve désormais en mémoire cache. Il faut mettre en mémoire cache les données susceptibles d'être le plus souvent utilisées car elle est beaucoup plus petite que la mémoire principale. Il faut également utiliser des algorithmes de synchronisation permettant d'avoir les mêmes valeurs en mémoire cache et en mémoire principale, sinon on risque d'avoir des erreurs car seul le microprocesseur a accès à la mémoire cache.

Les microprocesseurs actuels intègrent plusieurs niveaux de mémoire cache. La mémoire cache (ou simplement *cache*) de niveau 1 est rapide, mais de taille limitée (quelques dizaines de kio*). Le cache de niveau 2 est moins rapide, mais de taille plus importante (quelques centaines de kio). On peut également trouver dans certains microprocesseurs un cache de niveau 3 (quelques Mio). Le microprocesseur va d'abord chercher la donnée dans le cache L1 (*Level* 1), puis, s'il ne la trouve pas, dans le cache L2, et ainsi de suite…

Quelques temps d'accès :



Device	*Typical access times*
CPU registers	0.25 nsec
Cache memory (SRAM)	1-10 nsec
Conventional memory (DRAM)	10-50 nsec
Flash memory	120 µsec
Magnetic disk drive	10-50 msec
Optical disk drive	100-500 msec
Magnetic tape	0.5 and up sec



Pour effectuer un traitement (calcul, comparaison, etc…), les instructions du microprocesseur ont accès uniquement aux registres. Ce sont les mémoires les plus rapides, mais au nombre très limité (quelques dizaines d'octets au plus).

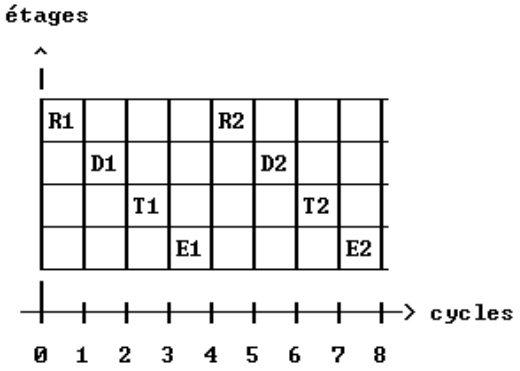
Microprocesseur : pipeline et architecture superscalaire

Pour exécuter une instruction, le microprocesseur passe par plusieurs étapes.

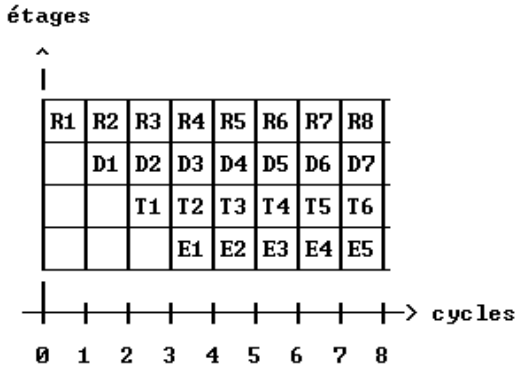
Considérons, par exemple, quatre étapes :

- la récupération de l'instruction en mémoire (étape R)
- le décodage de l'instruction, pour savoir si c'est une instruction de calcul, de comparaison, etc... (étape D)
- le traitement de l'instruction (étape T)
- l'enregistrement du résultat en mémoire (étape E)

Considérons également que chaque étape (ou étage) s'exécute en un cycle d'horloge, cela nous donne :



Au bout de huit cycles, nous avons exécuté deux instructions car l'instruction suivante ne peut commencer qu'une fois toutes les étapes de l'instruction précédente effectuées. Nous constatons beaucoup d'étapes inutilisées au niveau de chaque cycle. Et si on remplissait les cases vides ! Cela nous donne :



Nous exécutons toujours une instruction en quatre cycles d'horloge, mais au bout de huit cycles, nous avons exécuté cinq instructions au lieu de deux ! Bien-sûr, pour cela, le processeur doit pouvoir effectuer les quatre étapes simultanément. Au plus nous avons d'étapes (on parle de profondeur, la profondeur étant le nombre d'étapes), au plus le temps global d'exécution diminue... Enfin, jusqu'à une certaine limite. En effet, même si nous avons vu que le pipeline augmente les performances du microprocesseur, il pose problème dans certains cas : si, par exemple, l'instruction 5 a besoin du résultat de l'instruction 4, elle doit être arrêtée et attendre la fin de l'exécution de l'instruction 4. Pour éviter les erreurs, il faut utiliser des algorithmes qui gèrent ces dépendances, et on perds à nouveau du temps...

Une architecture superscalaire utilise plusieurs pipelines, plusieurs instructions peuvent donc être exécutées simultanément (traitement parallèle), mais là aussi avec des limitations liées aux dépendances.

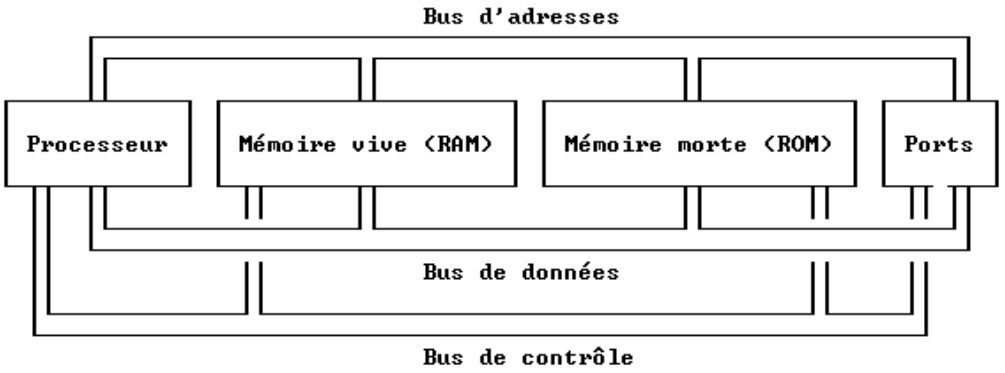
Microprocesseur : architectures CISC et RISC

L'architecture CISC (*Complex Instruction Set Computer*) utilise des instructions qui peuvent nécessiter plusieurs dizaines de cycles d'horloge et qui ne s'exécutent pas pendant le même nombre de cycles. Au contraire, l'architecture RISC (*Reduced Instruction Set Computer*) utilise des instructions qui ne nécessitent que quelques cycles d'horloge (voir même un seul cycle) et qui s'exécutent pendant le même nombre de cycles. Dans un pipeline, les instructions doivent s'exécuter pendant un même nombre de cycles, on ne peut donc pas utiliser de pipeline dans un microprocesseur CISC. Mais alors, dans ce cas, pourquoi avoir créé des microprocesseurs

CISC ? Car toute la complexité était gérée au niveau du microprocesseur. Pour le programmeur, le codage était donc beaucoup plus facile. Oui mais on perdait la rapidité (grâce aux pipelines) des microprocesseurs RISC...

Les deux architectures se sont ainsi opposées pendant des décennies, chacune vantant ses avantages par rapport à l'autre, dans les magazines de l'époque. Aujourd'hui, la paix est de retour (définitivement) car les microprocesseurs sont CISC... Et RISC en même temps ! Le ou les cœur(s) utilise(nt) une architecture RISC superscalaire (plus simple à mettre en œuvre qu'une architecture CISC, on peut donc augmenter le nombre de cœurs sans perdre trop de place), mais c'est une architecture CISC qui gère la partie «décodage des instructions» : les instructions CISC sont décomposées en instructions RISC, et sont ensuite envoyées au(x) cœur(s). Ainsi, le processseur est vu de l'extérieur comme un processseur CISC, le code reste simple et agréable à lire. Enfin, tout dépend du programmeur...

Bus de données, d'adresses et de contrôle



Pour communiquer avec les autres composants, un microprocesseur utilise trois bus : un bus de données, un bus d'adresses, et un bus de contrôle. Le bus de données permet la lecture ou l'écriture d'une valeur, c'est donc un bus bidirectionnel. Par contre, seul le microprocesseur définit l'adresse à lire ou à écrire, le bus d'adresses est donc un bus unidirectionnel. Le bus de contrôle est un bus bidirectionnel permettant la synchronisation des composants et la gestion des E/S, grâce au mécanisme des interruptions.

La largeur du bus de données détermine les nombres que le microprocesseur peut traiter. Ainsi, avec un bus de données sur 4 bits, le microprocesseur peut traiter des nombres allant de 0 à 15 (2^4-1) en binaire non signé, ou de -8 à 7 (-2^3 à 2^3-1) en binaire signé. En 1971, avec le premier microprocesseur, l'Intel 4004, on était donc un peu limité... Avec un bus de données sur 8 bits, le microprocesseur peut traiter des nombres allant de 0 à 255 (binaire non signé) ou de -128 à 127 (binaire signé), c'est un peu mieux. Avec un bus de données sur 16 bits, le microprocesseur peut traiter des nombres allant de 0 à 65535 (binaire non signé) ou de -32768 à 32767 (binaire signé), c'est beaucoup mieux. Avec un bus de données sur 32 bits, on dépasse 4 milliards en binaire non signé et 2 milliards en binaire signé, c'est suffisant dans la plupart des cas.

La largeur du bus d'adresses détermine la quantité de mémoire à laquelle le microprocesseur peut accéder. Avec un bus d'adresses sur 16 bits, le microprocesseur peut accéder à 65536 adresses (64 kio), c'est un peu juste. Avec un bus d'adresses sur 32 bits, le microprocesseur peut accéder à plus de 4 milliards d'adresses (4 Gio), c'est suffisant dans la plupart des cas, même si aujourd'hui les microprocesseurs utilisent des bus de données et d'adresses sur 64 bits.

Chipset

Le pont nord, *northbridge* ou GMCH (*Graphics and Memory Controller Hub*), gère le FSB (*Front Side Bus* ou bus système) reliant le microprocesseur, la mémoire et la carte graphique. Il nécessite donc une bande passante importante (bus rapide et large).

Le pont sud, *southbridge* ou ICH (*Input/output Controller Hub*), gère la communication avec les périphériques (clavier, souris, disque dur, carte réseau, ports USB, etc...). Il nécessite une bande passante plus faible.

Le composant appelé chipset est l'ensemble de ces deux ponts, dans un même circuit intégré.

Documentation complémentaire

- Architecture des ordinateurs
- 8086 & DOS

* 1 ki (kilo informatique) = 1024 et non pas 1000, donc 1 kio = 1024 octets et 16 kio = 16384 octets.