

::: Architecture :::

semaine n°7
Port série

- Le port série ou RS232 est un système en voie de disparition sur nos PC mais toujours bien présent en micro-électronique.
- La communication série se distingue par le fait que les données sont transmises bit à bit contrairement à une communication parallèle où elle le sont octet par octet par exemple.
- La communication série est donc moins rapide à fréquence égale mais elle nécessite moins de fils (seulement 3) permettant des communications à plus longue distance (prix du câble)

- La communication série RS232 est asynchrone : il n'y a pas de signal d'horloge transmis avec le signal. L'émetteur et le récepteur ont leur propre horloge qu'ils doivent synchroniser avec les données reçues.
- Des bits supplémentaires sont envoyés pour aider à cette synchronisation. BIT START, BIT STOP ; ils encadrent chaque octet.
- La communication peut être half-duplex : bidirectionnelle mais il n'est pas possible d'émettre et recevoir en même temps. Elle peut être full-duplex : bidirectionnelle sans restriction.
- La communication peut être simple : unidirectionnelle.

- Les successeurs de la RS232, communiquant en mode série sont l'USB, le Firewire ...
- Les périphériques qui utilisaient de mode de communication parallèles (disque IDE) se mettent à la communication série (Serial ATA) => Les fréquences élevées de communication compensent largement la perte due à la sérialisation / désérialisation des données.

- Fonctionnement de l'interface série :
 - Pour communiquer en série, puisqu'il n'y a pas d'horloge externe pour donner le rythme, il faut se caler sur un rythme pré-établi. Les débits RS232 sont :
75, 150, 300, 600, 1200, 2400, 4800, 9600 bits/s
19200, 38400 ... 115K bits/s
 - Le MCS251 supporte des débits jusqu'à 9600 bits/s

Pour choisir ce débit, nous allons utiliser un temporisateur configuré en mode 2 (8bits à rechargement automatique). Ce mode sera une garantie de stabilité du rythme de transmission. Sans cette garantie, l'émetteur et le récepteur se désynchroniseraient.

- Fonctionnement de l'interface série :
 - La communication sur le fil se fait bit à bit, mais pour le programmeur, les données sont au minimum des octets. La transmission d'un octet, bit à bit est donc assurée directement par le matériel.
 - En émission, le programme écrit l'octet à envoyer dans le registre SBUF ; le matériel détecte cette écriture et envoie les données bit à bit. Le matériel va ensuite informer le programme que l'émission est terminée : il positionne le bit TI à 1 et, si elle est activée, déclenche une interruption.
 - En réception, l'arrivée d'un octet déclenche une interruption ou positionne le bit RI à 1. Le programme pourra lire l'octet dans le registre SBUF.

- Fonctionnement de l'interface série :
 - L'interface série peut être configurée en mode 9 bits, dans ce cas, le 9ème bit est utilisé pour indiquer une parité. La parité est un bit qui indique si dans les 8 autres le nombre de bit à 0 ou 1 est pair ou impair.

Lorsqu'une communication est parasitée, le plus souvent, 1 seul bit est modifiée (inversé). Dans ce cas, la parité calculée à la réception diverge de celle reçue. Cette méthode permet de détecter des erreurs de transmission. N'étant pas totalement fiable, elle est généralement complétée par un système de CRC ou checksum.

::: Architecture :::

semaine n°7

Architecture des processeurs actuels

Qu'est – ce qui fait la performance d'un ordinateur ?

Qu'est – ce qui fait la performance d'un ordinateur ?

- => Sa fréquence de fonctionnement
- => Le nombre de processeurs à l'intérieur
- => Le nombre d'unités de traitement par processeur

- => Le débit entre la mémoire et le processeur
- => La taille des caches

- => La performance des périphériques
- => La performance des liaisons entre les périphériques

- => ...

Mais d'abord ... Qu'est-ce que la performance ???

- => Le nom du composant ?
- => Est-ce le nombre de Mhz ?
- => Est-ce la taille des cache ?
- => Est-ce le nombre d'unités de traitement ?

Comment la mesurer ?

Mais d'abord ... Qu'est-ce que la performance ???

=> Le nom du composant ?

- Athlon Xp 2400 + est-il vraiment aussi rapide qu'un P4 à 2,4GHz ?

=> Est-ce le nombre de Mhz ?

- Les Mhz donnent le temps d'un cycle, pas ce qui est fait en 1 cycle (1,2,3,4 instructions ? $1/2$, $1/4$ d'instruction ?)

Comment la mesurer ?

=> MIPS : Million d'instruction par seconde

=> FLOPS : Floating point opérations par seconde

=> SPEC : des tests dédiés à des applications particulières (bureautique, calcul mathématiques, java ...)

La performance n'est pas le résultat de données théoriques et commerciales (fréquence, mémoire ...). C'est un résultat plutôt empirique et contextuel issu de l'architecture, c'est à dire du système informatique pris dans sa globalité.

La performance se compare plus qu'elle ne se mesure et il est important de se méfier des mesures car bon nombre de systèmes sont conçu pour n'être performance QUE pour les tests de référence.

Pour dire : « je suis le plus performant » ou « je grille tel processeur »

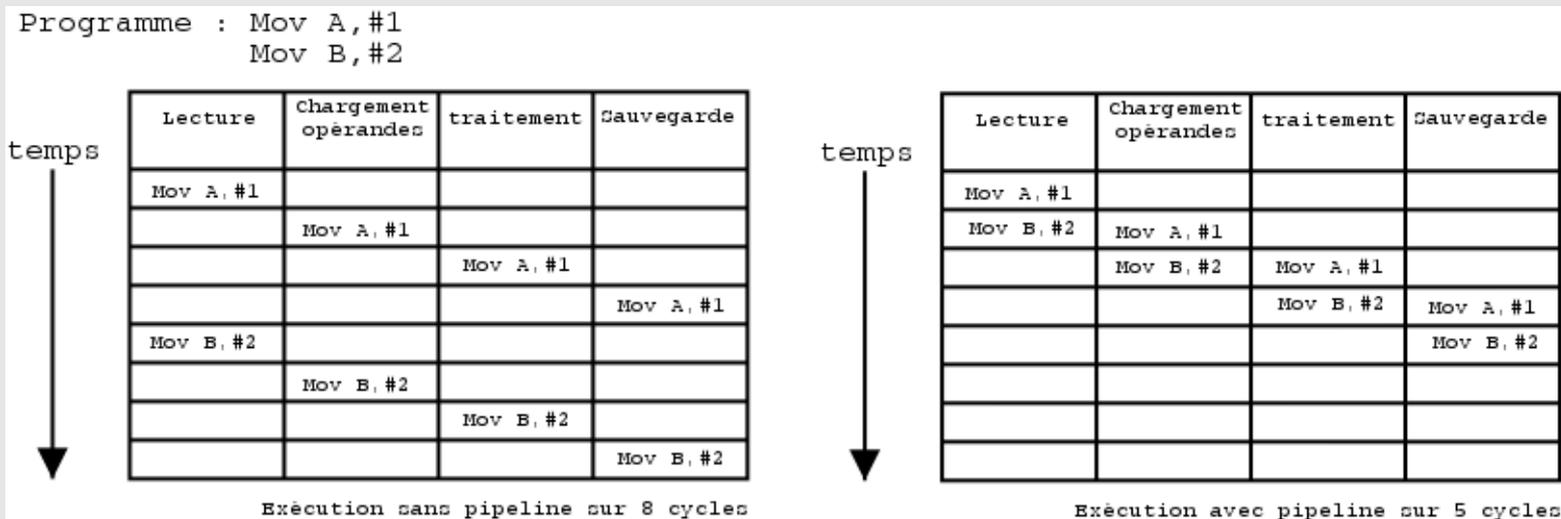
D'une façon général, voyons une approche de la performance par les MIPS : capacité à traiter un certain nombre d'instructions par seconde.

=> Cas du MCS251, une instruction dure plusieurs cycle entre 1 et 20 ; la moyenne étant autour de 3 cycles, pour une fréquence de 8Mhz la puissance sera de seulement 2,6 MIPS environ.

=> Cas d'un processeur RISC où en général, les instructions ne durent qu'un cycle pour 8MHz, le système délivre 8MIPS.

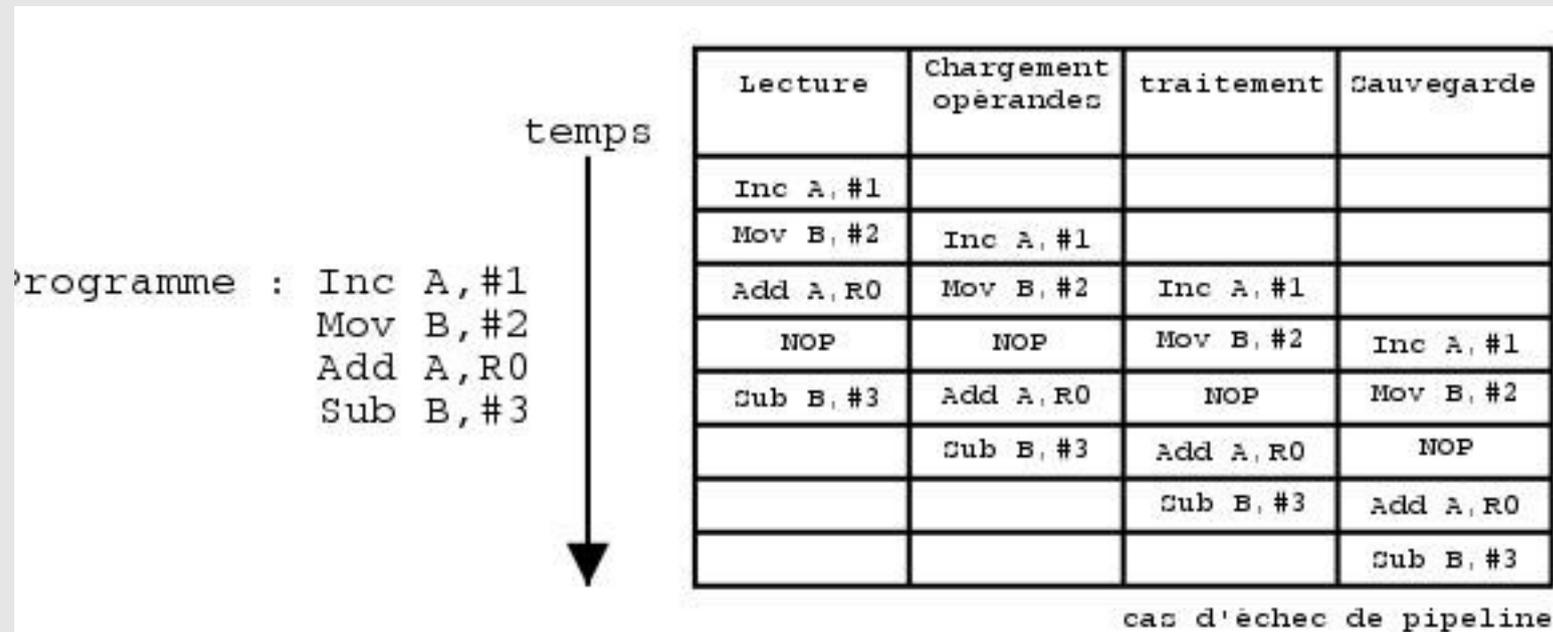
=> Cas d'un processeur INTEL/AMD.. du monde PC, bien que les instructions soient du monde CISC et qu'elles nécessitent plusieurs cycles, le nombre de MIPS proche de la fréquence !

=> Ces processeurs utilisent une architecture avec PIPELINE



Le PIPELINE permet de sortir 1 instruction à chaque cycle

=> Toutefois, ça ne marche pas à tous les coups



Lorsque l'opérateur d'une instruction est remplacé par un NOP, l'opérateur de l'instruction précédente dont le résultat est en cours de calcul une bulle (un NOP) doit être inséré.

LE RÔLE DU COMPILATEUR EST PRIMORDIAL

Comment faire mieux que la fréquence ? avoir plus de MIPS que de MHZ ?

- => Il faut exécuter plusieurs instructions en même temps
 - => Avec plusieurs processeurs ou core (mais dans ce cas on exécute des processus différents)
 - => Avec plusieurs unités de traitement dans le processeur on parle d'architecture SUPERSCALAIRE.
- Cas du PC.

Programme :
 Mov A , #1
 Mov B , #2
 Mov R0 , #3
 Mov R1 , #4

Cycle	Chargement	Exécution	Sauvegarde	
1	Mov A , #1			unité A
	Mov B , #2			unité B
2	Mov R0 , #3	Mov A , #1		unité A
	Mov R1 , #4	Mov B , #2		unité B
3		Mov R0 , #3	Mov A , #1	unité A
		Mov R1 , #4	Mov B , #2	unité B
4			Mov R0 , #3	unité A
			Mov R1 , #4	unité B
5				unité A
				unité B

Utilisation d'une architecture superscalaire avec pipeline

Les architectures superscalaires souffrent du même problème que pour le pipeline : en cas de conflits des NOP doivent être insérés ; et les raisons sont très nombreuses...

Ainsi malgré son architecture pipeline + superscalaire, le processeur des PC délivre à peine plus de MIPS que leur fréquence alors qu'en théorie il pourrait délivrer 2 instructions par cycles.

D'où l'idée d'Intel : l'**HYPER-THREADING** – Profiter de tous ces temps morts où l'on insère des NOP pour exécuter d'autres processus, le temps que les conflits soient résolus. Le gain est de l'ordre de 20 à 30%

En pratique exécution d'un même programme :

386_25Mhz=> 291s = 7275s pour 1Mhz

486_33Mhz=> 161s = 5313s pour 1Mhz (1/2 pipeline)

586_60Mhz=> 31s = 1860s pour 1Mhz (pipeline / sscal)

P2_450Mhz=> 1,68s = 756s pour 1Mhz

P3_800Mhz=> 0,87s = 696s pour 1 Mhz

Ath_1,2Ghz=> 0,43s = 516s pour 1 Mhz

P4_2,2Ghz => 0,41s = 836s pour 1 Mhz

AthXp_2,5Ghz=> 0,28s = 701s pour 1 Mhz

Après une phase de très forts gains issus de l'architecture, la performance est plutôt en régression actuellement ... **Pourquoi ??**

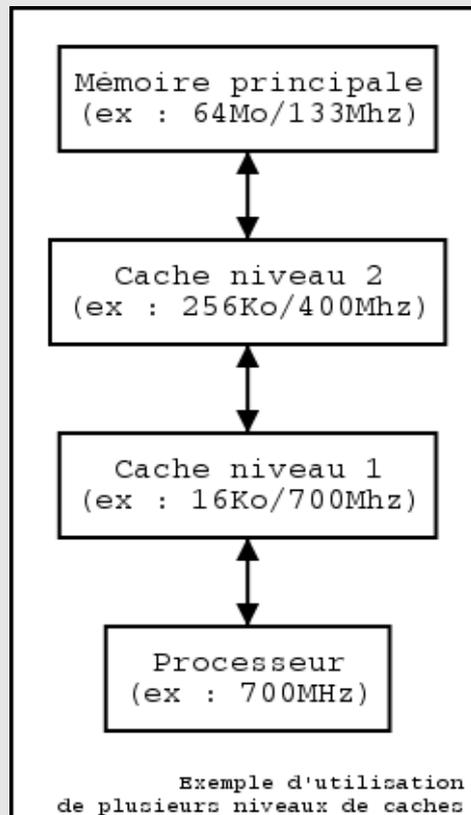
Principal problème : la mémoire

=> Notre processeur, fonctionne à 4 Ghz et peut consommer pas loin de 5 milliards d'instructions par seconde ...

=> mais notre mémoire, elle, fonctionne à 200 Mhz ou 2x200Mhz en DDR (un peu plus maintenant avec la DDR2) ne peut pas en fournir plus de 400 millions d'instructions à la seconde...

Soit 10 fois moins ... Notre processeur, bien que très rapide passe son temps à attendre des données !

=> D'où l'importance des CACHES : avoir de la mémoire, même en petite quantité mais qui aille à la vitesse du processeur !



Lorsque le processeur accède au cache il n'attend pas et peut donc profiter au maximum de sa fréquence. Mais dès lors qu'il veut accéder à des données qui n'y sont pas ... il attend !

Plus l'écart entre vitesse du processeur et vitesse de FSB est grande ... plus il attend !

La mémoire cache est chère ... donc on en met peu.

Le code doit être optimisé en fonction de la taille du Cache.

C'est comme ça que l'on gagne un bench ... AMD machins+

Les copro-dédiés :

=> Les traitements câblés sont beaucoup plus rapides que d'enchaîner des instructions pour les traitements complexes (multimédia, 3D...)

=> On leur dédie des co-processeurs : MMX, 3DNow ... et des cartes (composante 3D des cartes graphiques)

=> Les premiers co-processeurs étaient arithmétiques, pour les calculs en virgule flottante.

=> Souvent, lorsqu'il y a plusieurs unités de traitements, voire plusieurs core, il n'y a pas autant de co-processeurs que d'unités.

Les architectures 16 – 32 – 64bits

=> Est-ce LA solution à la performance ?

=> Le type d'architecture définit grosso-modo la taille de l'élément de base manipulé dans le processeur ; concrètement, la taille du « int » en C.

=> Cela veut dire :

-> Que les calculs sont effectués en hard sur des nombre de cette taille. Les nombres 16bits ne sont pas très usuels, trop petits pour compter le temps, l'argent et dénombrer bon nombre de choses ; du coup le passage a 32 bit a été très intéressant pour optimiser les calculs !

-> Que les adresses pourront être sur 64bits et nous allons donc pouvoir mettre plus de mémoire dans les machines. Toutefois, les familles P4 & co pouvait déjà accéder à 36bits de mémoire soit 64 Go. Ce n'est donc pas une fin en soit et ne concerne vraiment que les serveurs.

=> Cela veut dire :

-> Que les données seront lues par blocs de 64 bits et non de 32 et là c'est un vrai gain, car pour la même fréquence d'interface avec la mémoire, on multiplie par deux le débit des données : le processeur passe moins de temps à ne rien faire ! (DDR2)

-> C'est l'occasion de repenser complètement l'architecture du processeur et réalisant une vraie rupture. Les PII, PIII et PIV n'ont rien de très très différents les uns par rapport aux autres...

-> Par ailleurs, le marketing a évolué, la course au Ghz fait place au MIPS/Watt, au dB dégagés, et aux technologies multi-coeur.

-> cela fait aussi 2-3 ans qu'un utilisateur lambda n'obtient plus rien en changeant de PC...