Pentium

Parallélisme et multi-coeurs

Performances, comment les améliorer encore plus?

- Texe = N * CPI * 1/F
- Texe : Temps d'exécution d'un programme donné sur un processeur donné
- N : nombre d'instructions dans ce programme
- CPI : (cycles per instruction) nombre de cycles horloge nécessaire par instruction
- F : fréquence d'horloge

Optimisations diverses

- Texe = N * CPI * 1/F
- Objectif : exécuter plus vite, donc minimiser Texe
- Donc minimiser chaque composante du produit
- Minimiser N => optimiser les instructions (le noyau du processeur)
- Minimiser CPI => utiliser un cache
- Minimiser 1/F => **augmenter F** (fréquence d'horloge)

No Pentium 4 at 4 GHz

Craig Barret, (ex PDG d'Intel), s'excusant publiquement de ne pas respecter la promesse de cette compagnie de mettre sur le marché un Pentium 4 fonctionnant à 4Ghz!
La course à la vitesse est arrêtée...

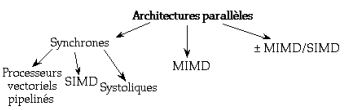
19 octobre 2004 - Orlando

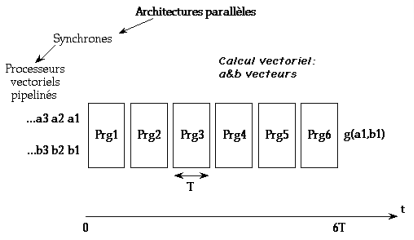
Architectures parallèles

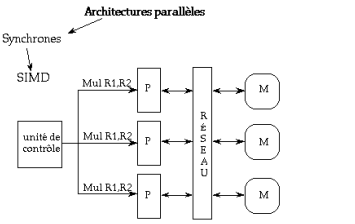
Classification de Flynn (années 70)

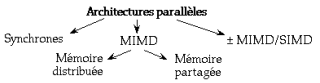
flots d'instructions et de données

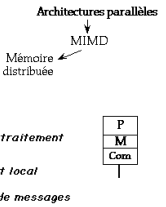
- . *SISD* *Single Instruction Single Data streams*
- . *SIMD* *Single Instruction Multiple Data streams*
- . *MIMD* *Multiple Instruction Multiple Data streams*
- . *MISD*

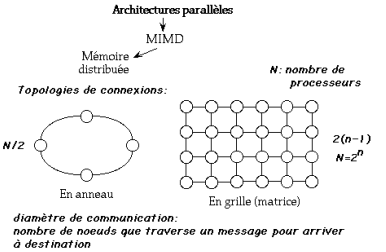


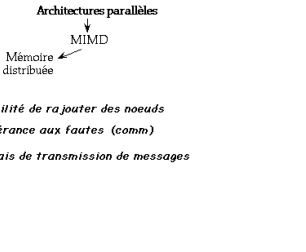

Architectures parallèles
Synchrones
Processeurs vectoriels pipelinés
Calcul vectoriel: a&b vecteurs
...a3 a2 a1
...b3 b2 b1
Prg1
Prg2
Prg3
Prg4
Prg5
Prg6
g(a1,b1)
T
t
0
6T

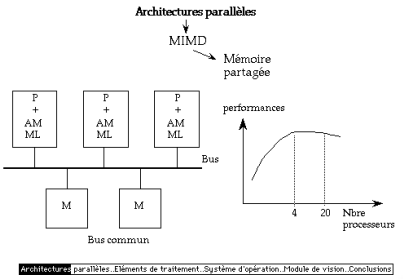

Architectures parallèles
Synchrones
SIMD
unité de contrôle
Mul R1,R2
Mul R1,R2
Mul R1,R2
P
P
P
RÉSEAU
M
M
M


Architectures parallèles
Synchrones
MIMD
± MIMD/SIMD
Mémoire distribuée
Mémoire partagée


Architectures parallèles
MIMD
Mémoire distribuée
• noeud de traitement
• traitement local
+
échange de messages
P
M
Com


Architectures parallèles
MIMD
Mémoire distribuée
Topologies de connexions:
N: nombre de processeurs
N/2
En anneau
2(n-1)
N=2^n
En grille (matrice)
diamètre de communication:
nombre de noeuds que traverse un message pour arriver à destination


Architectures parallèles
MIMD
Mémoire distribuée
+ facilité de rajouter des noeuds
+ tolérance aux fautes (comm)
– délais de transmission de messages



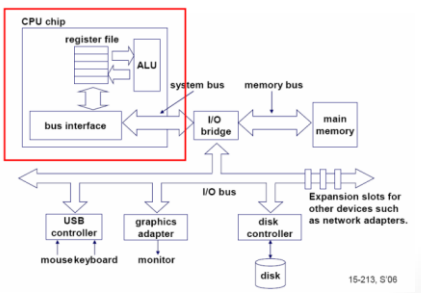
Intel : Plusieurs vues du parallélisme

- 2 processeurs côte à côte
  - Adapter le processeur pour une telle connexion
  - Bi processeurs, Quad processeurs
- Double cœur ou N-cœur à l'intérieur d'un même processeur
  - Deux noyaux très liés
  - Hyperthreading

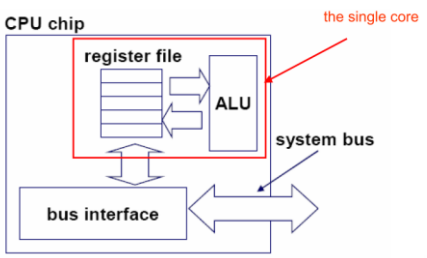
2 processeurs côte à côte

- Pentium version Itanimum (1 & 2)

mono-processeur

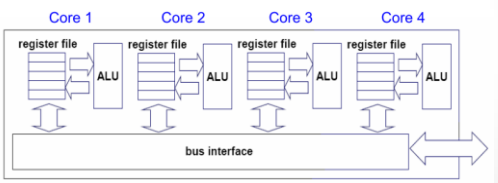


Simple coeur

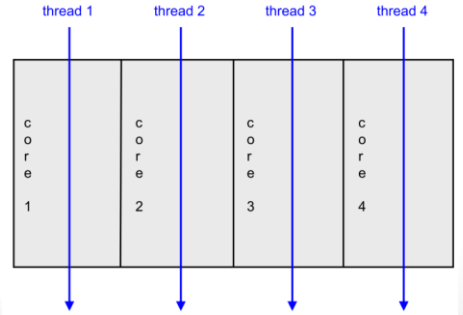


Multi-cœurs

- Répliquer le cœurs n-fois sur la même puce

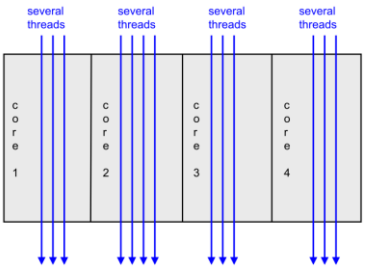


Les cœurs fonctionnent en parallèle



Hyper-threading

- Chaque cœur peut faire fonctionner en parallèle

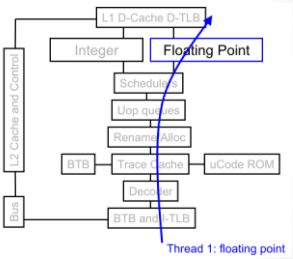


Parallélisme au niveau instruction machine

- Le processeur peut
  - ré-ordonnancer les instructions
  - Pipeliner les instructions
  - Décomposer les instructions en micro-instructions
  - Faire une prédiction de branchement poussée, etc.
- Le parallémisme au niveau instruction a propulsé les performances de processeurs sur les dernières 10 années

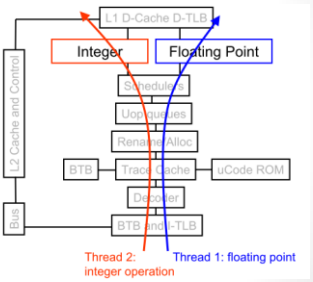
Core-Threading, le principe

- Une séquence unique de micro-instructions peut s'exécuter en même temps



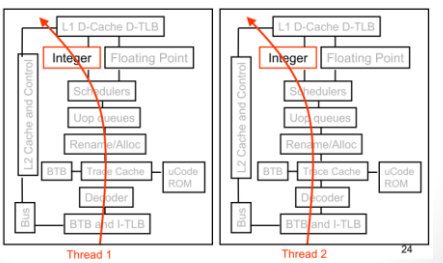
Simultaneous Multi-Threading

- Avec le SMT, 2 thread peut s'exécuter simultanément dans un cœur
- Condition :
  - Chacun sollicite une unité arithmétique différente
  - L'un Integer l'autre Floating point



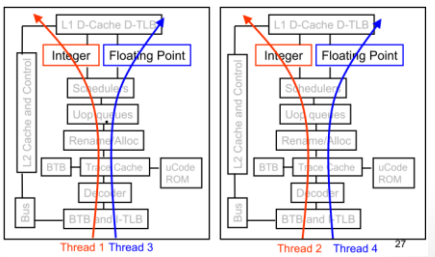
Multi-coeurs

- Deux cœurs peuvent exécuter des threads en parallèle



SMT et multi-coeurs

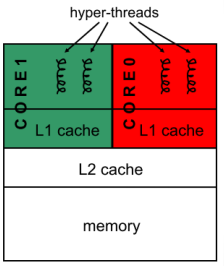
- En combinant SMT et multi-cœurs, les 4 threads peuvent s'exécuter en parallèle



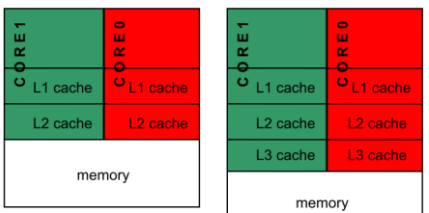
27

Difficultés hyperthreading

- Cohérence du cache
  - Caches L1 privés
  - Cache L2 partagé
  - Exemple Xeon
- Comment garantir que les variables partagées auraient les mêmes valeurs dans les caches privés ?



Conception avec cache L2 privé



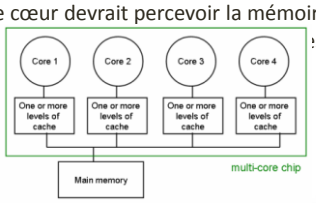
32

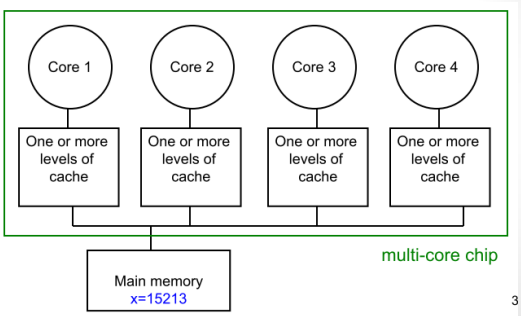
Cache privé/partagé, plus et moins

- Avantages du cache privé
  - Plus près du cœur, plus rapide d'accès
  - Diminue la contention
- Avantages du cache partagé
  - Les Threads sur des cœurs différents peuvent partager les mêmes données dans les caches
  - Plus d'espace cache est disponible si peu de threads tournent sur le système

Problème de cohérence du cache

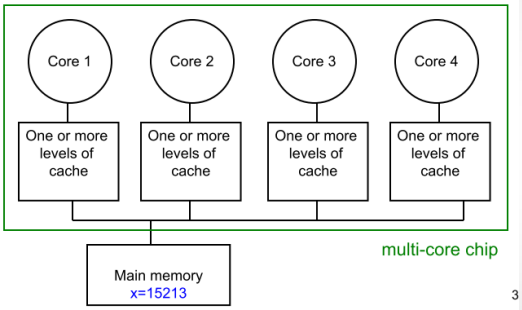
- Des copies des mêmes données résident dans des caches privés
- comment s'assurer que les données sont consistantes dans le différents caches ?
- Chaque cœur devrait percevoir la mémoire comme partagé





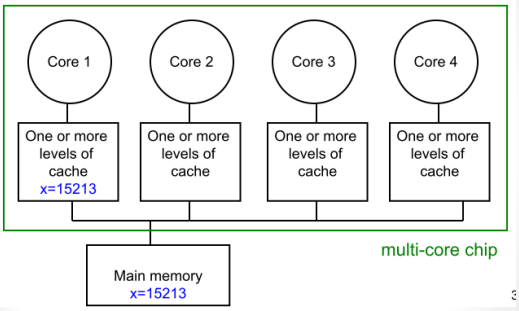
3

Suppose variable x initially contains 15213



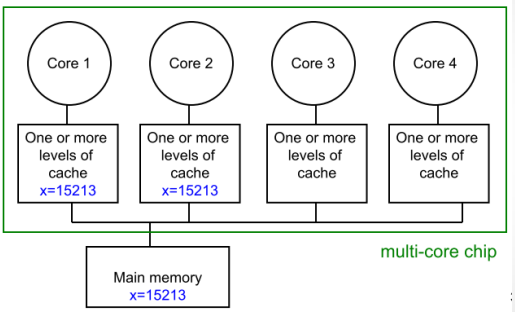
3

Core 1 reads x



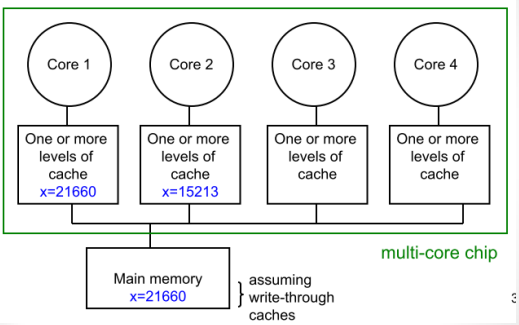
3

Core 2 reads x



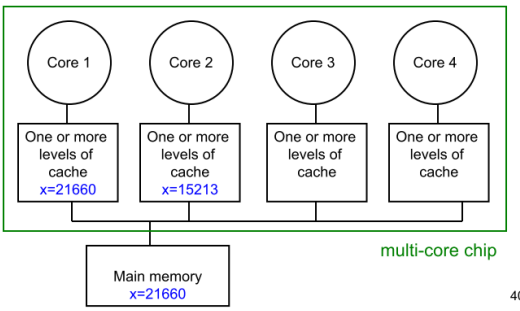
3

Core 1 writes to x, setting it to 21660



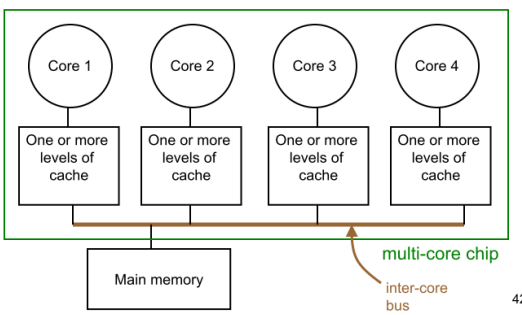
3

Core 2 attempts to read x… gets a stale copy



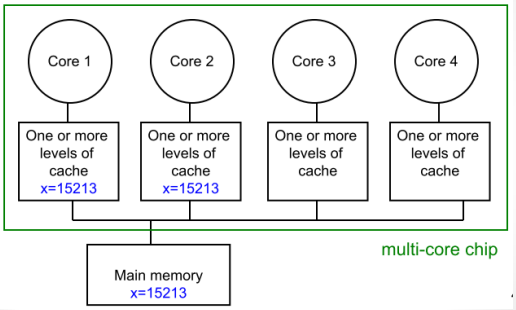
40

Inter-core bus

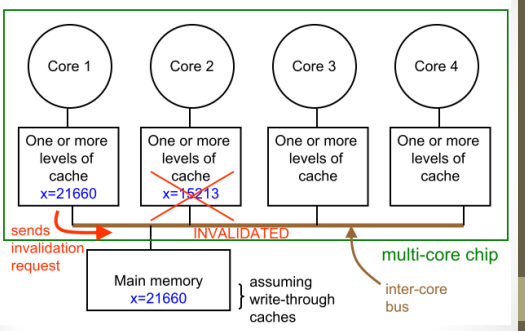


42

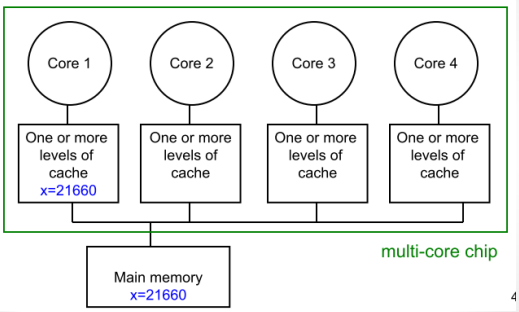
Revisited: Cores 1 and 2 have both read x



Core 1 writes to x, setting it to 21660



After invalidation:



4

Hyperpipeline et threading

- Noyau Prescott
  - Profondeur pipeline = 32 niveaux
  - Couplé à la station de réservation
  - Appellation « Threading »
    - Empruntée des systèmes d'exploitation multi-tâches
      - Ou multi « thread »

Interaction avec le système d'exploitation

- Le SE perçoit chaque cœur comme un processeur séparé
- L'ordonnanceur du processeur alloue des thread ou tâches aux différents cœurs
- La majorité des SE supporte le multicoeur aujourd'hui (Windows, Linux, Mac OS X, etc.)

Programmation en multi-core

- Le SE ne fait pas de parallélisation automatique du code
- Le SE alloue simplement un thread/process à un coeur
- Programmeurs doivent utiliser les process et threads
- Répartir la charge de calcul sur les cœurs
- Écrire des algorithmes parallèles

Sécurité du thread sous multi-coeurs

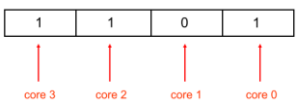
- Le changement de contexte pré-emptif peut arriver à n'importe quel moment
- Il s'agit d'une vraie concurrence et pas seulement du partage de temps sous mono-processeur
- La concurrence stigmatise plus rapidement les bogues

Allocation d'un thread à un coeur

- Chaque thread/process a un *masque d'affinité*
- Le masque d'affinité spécifie sur quels cœurs le thread est autorisé à tourner
- Les thread différent peuvent avoir des masques différents
- Les affinités sont héritées à travers la fonction fork()

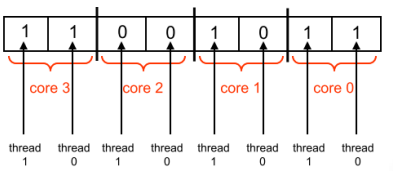
Structure des masques d'affinité

- Example : 4 cœurs, sans SMT
- Ce process/thread est autorisé à tourné sur les cœurs 0, 2 et 3, mais pas sur le cœur 1

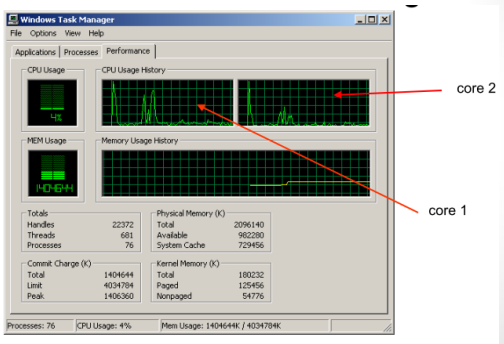


Masque d'affinité multi-cœur et SMT

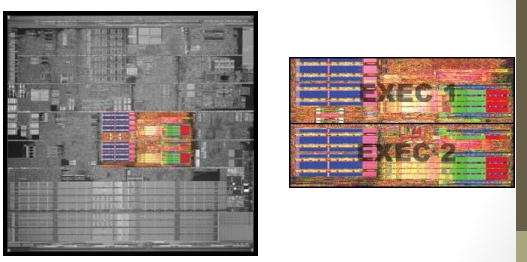
- Indicateur indépendant pour chaque thread
- Exemple : 4 cœurs, 2 threads par cœur



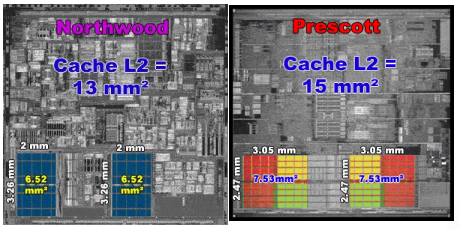
Gestionnaire de tâches windows



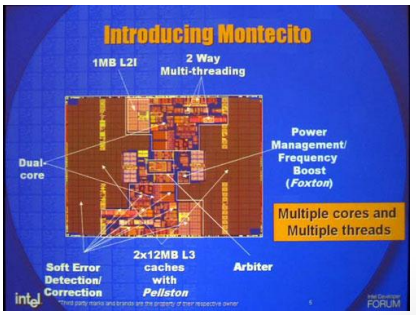
Pentium Prescott double core



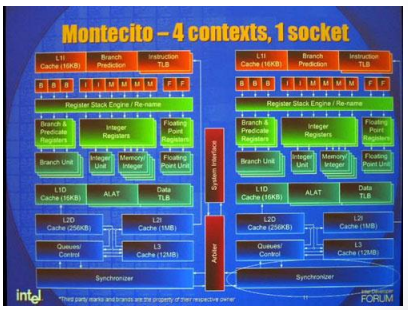
Northwood vs Prescott



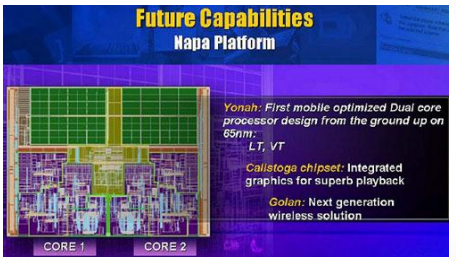
Pentium Montecito



Montecito double core



NAPA Double core + GP et Wireless

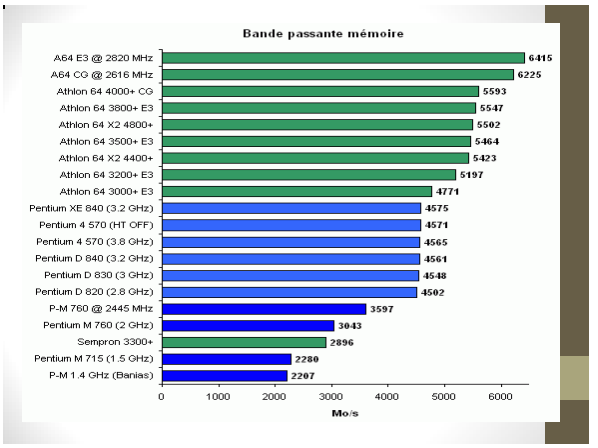


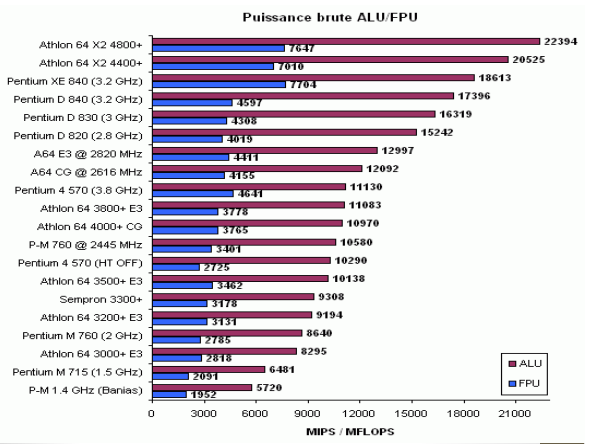
Spécifications des processeurs dual-core Intel

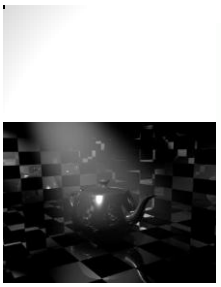
CPU	Pentium D 820	Pentium D 830	Pentium D 840	Pentium XE 840
Fréquence	2800 MHz	3000 MHz	3200 MHz	3200 MHz
Cache L1	2 x 28 Ko	2 x 28 Ko	2 x 28 Ko	2 x 28 Ko
Cache L2	2 x 1024 Ko	2 x 1024 Ko	2 x 1024 Ko	2 x 1024 Ko
FSB	800 MHz	800 MHz	800 MHz	800 MHz
Socket	LGA 775	LGA 775	LGA 775	LGA 775
Voltage	1,25 - 1,39 V	1,25 - 1,39 V	1,25 - 1,39 V	1,25 - 1,39 V
TDP	95 W	130 W	130 W	130 W
Nombre de transistors	230 millions	230 millions	230 millions	230 millions
Process	.09µ strained silicon	.09µ strained silicon	.09µ strained silicon	0.09µ strained silicon
Surface	206 mm²	206 mm²	206 mm²	206 mm²
Support du 64 bits	Oui	Oui	Oui	Oui
Support de l'EIST	Non : TM1 + TM2+ C0	Oui, TM1, TM2, C0	Oui, TM1, TM2, C0	Oui, TM1, TM2, C0
Support de l'HyperThreading	Non	Non	Non	Oui
Prix officiel	241 $	316 $	530 $	999 $

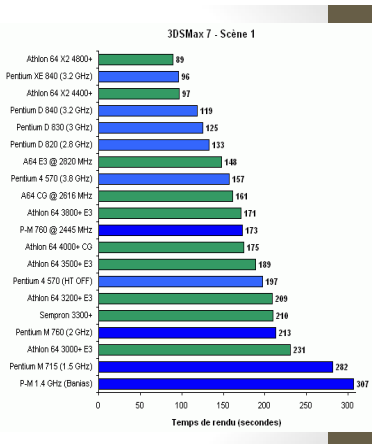
Spécifications des processeurs dual-core AMD

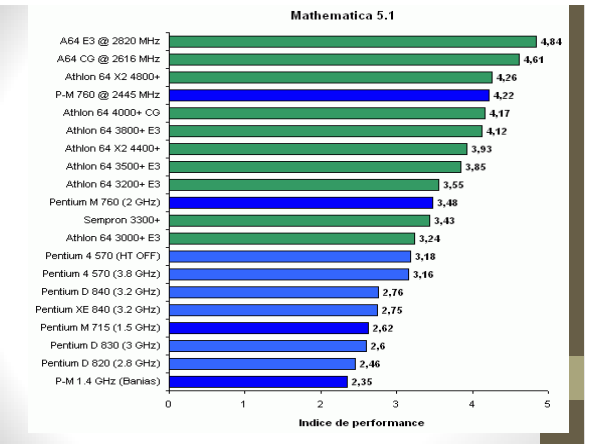
CPU	Athlon 64 X2 4200+	Athlon 64 X2 4400+	Athlon 64 X2 4600+	Athlon 64 X2 4800+
Fréquence	2200 MHz	2200 MHz	2400 MHz	2400 MHz
Cache L1	2 x 128 Ko	2 x 128 Ko	2 x 128 Ko	2 x 128 Ko
Cache L2	2 x 512 Ko	2 x 1024 Ko	2 x 512 Ko	2 x 1024 Ko
FSB	200 MHz	200 MHz	200 MHz	200 MHz
Socket	939	939	939	939
Voltage	1,35 -1,4 V	1,35 -1,4 V	1,35 -1,4 V	1,35 -1,4 V
TDP	110 W	110 W	110 W	110 W
Nombre de transistors	233 millions	233 millions	233 millions	233 millions
Process	.09µ SOI + DSL	.09µ SOI + DSL	.09µ SOI + DSL	.09µ SOI + DSL
Surface	199 mm²	199 mm²	199 mm²	199 mm²
Support du 64 bits	Oui	Oui	Oui	Oui
Support du Cool & Quiet	Oui	Oui	Oui	Oui
Support de l'HyperThreading	Non	Non	Non	Non
Prix officiel	537 $	581 $	803 $	1001 $

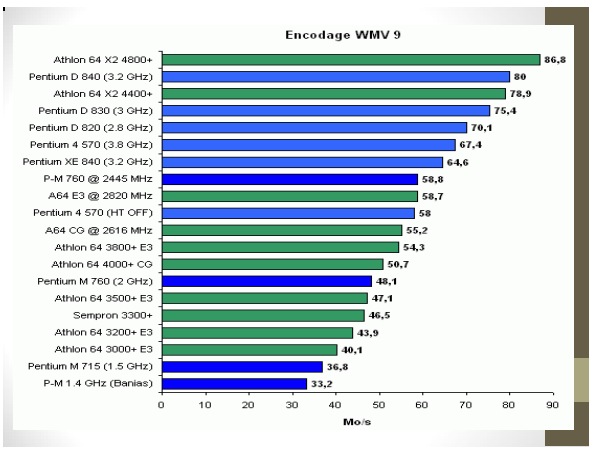










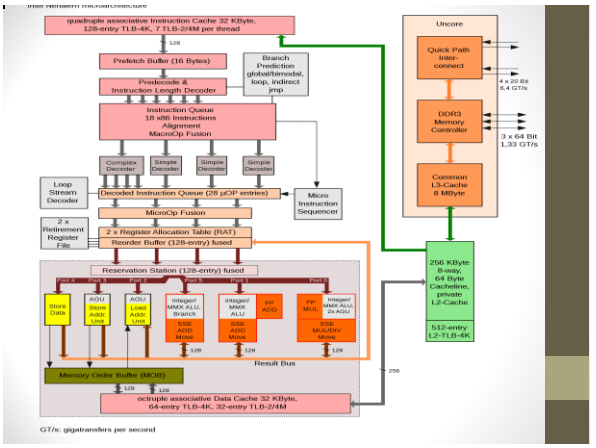


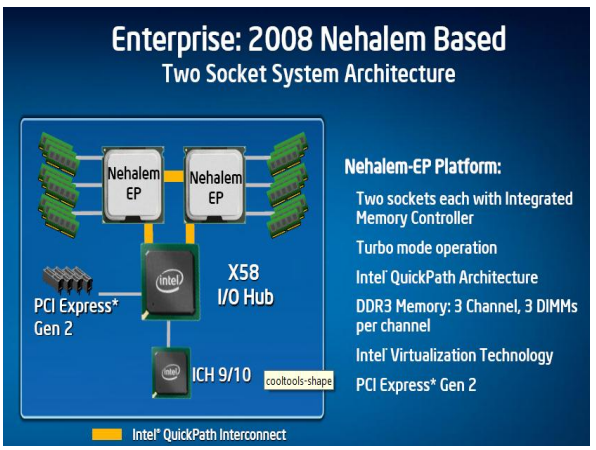
Inconvénients du multi core

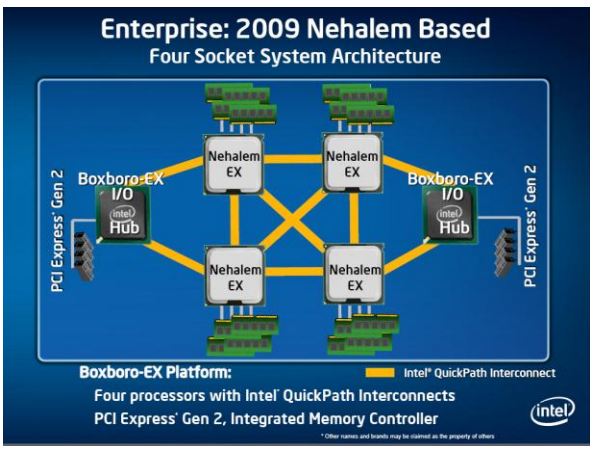
- Cohérence des caches et mémoires
  - Si 2 processeurs (ou plus) ont dans leur cache la copie de la même zone mémoire
  - si un processeur modifie son cache, refléter cette modification dans les caches de tous les autres processeurs
  - Et refléter ce changement dans la mémoire extérieure

Pentium i7 (Hyper Threading)

Numérotation	Mémoire cache	Fréquence d'horloge	Bus principal	Nb. de cœurs	Technologie Intel® VT²	Intel® 64²	Technologie Intel® TXT	Bit de verrouillage²
45 nm								
i7-950	8 MB	3,066 GHz	QPI 4,8 GT/s	4	✓	✓		✓
i7-940	8 MB	2,933 GHz	QPI 4,8 GT/s	4	✓	✓		✓
i7-920	8 MB	2,666 GHz	QPI 4,8 GT/s	4	✓	✓		✓
i7-870	8 MB	2,93 GHz	DMI 2,5 GT/s	4	✓	✓	✓	✓
i7-860	8 MB	2,8 GHz	DMI 2,5 GT/s	4	✓	✓	✓	✓

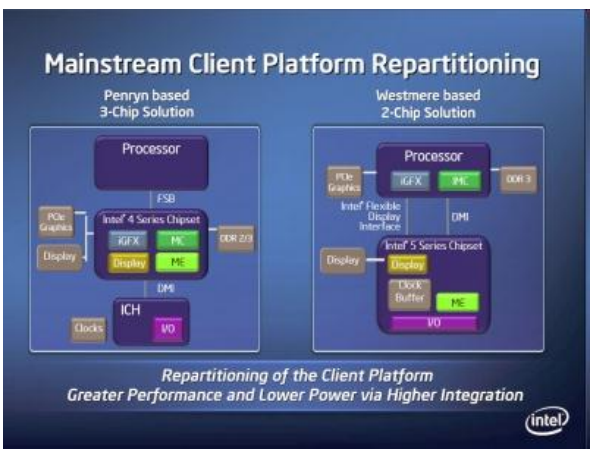


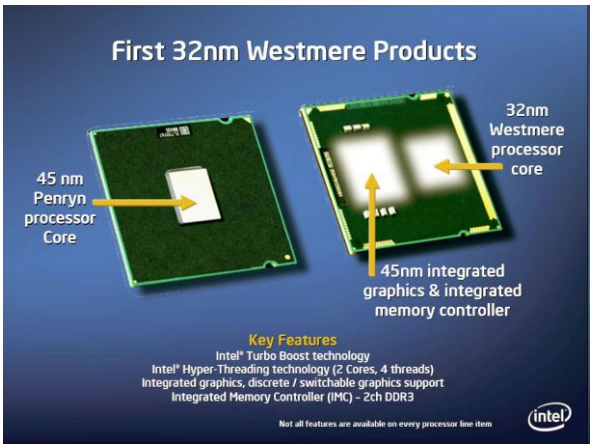




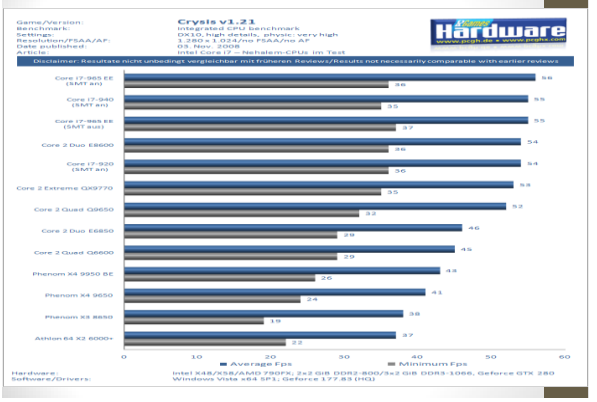
Core : 3 à 2 chip

- Nommé Westmere
- Fonctionalité GPU déplacée du pont nord vers CPU





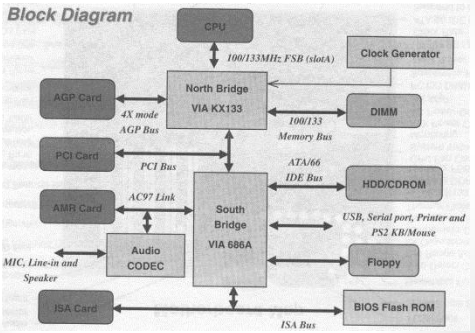
Other Fun Info



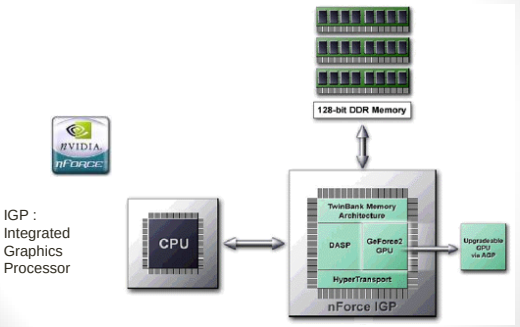
Pentium

Interface Graphique avancée

VIA Apollo KX133 et Apollo KT133 (Athlon et Duron)



AMD-760 (Athlon et Duron)



IGP : Integrated Graphics Processor

Le MCP