

Evolution des performances

qkzk

Évolution des performances

Loi de Moore

Tous les 18 mois, le nombre de transistors par processeur double

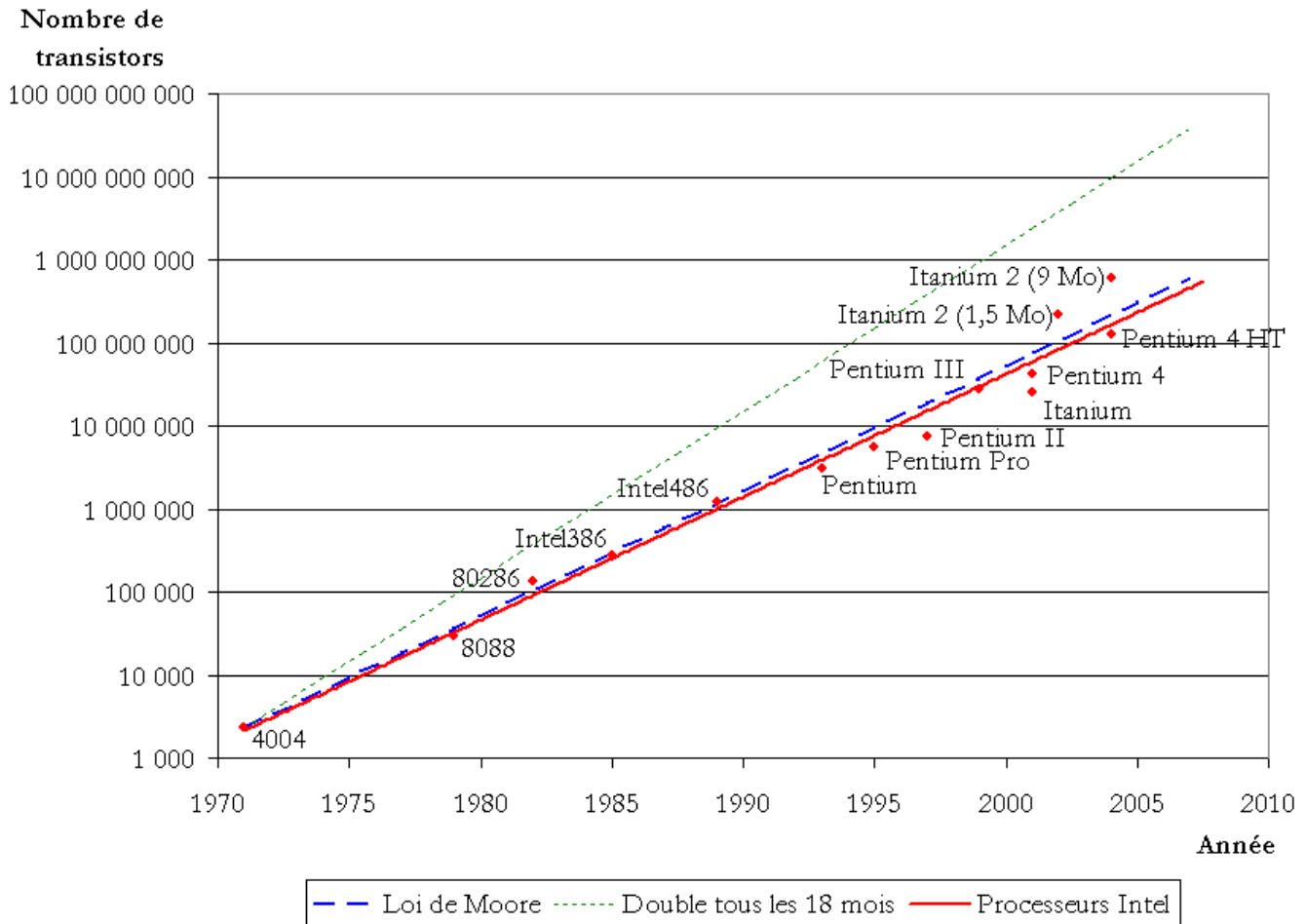


Figure 1: loi de moore

Loi de Moore

Tous les 18 mois, le nombre de transistors par processeur double

- Énoncée en 1965 par Gordon Moore (pdg d'Intel).
- Restée valable jusqu'en 2005 environ.

Applicable à d'autres contextes :

- énergie produite par le processeur,

- coût de développement etc.

Problème : la chaleur

La loi de Moore ne s'applique plus depuis ~15 ans.

Une contrainte physique : dissiper la chaleur

Plus on augmente la fréquence d'horloge d'un CPU, plus ce dernier chauffe.

Mais... si la surface diminue, dissiper la chaleur devient impossible.

Deux périodes dans l'histoire de l'informatique

Durant la loi de Moore

inutile d'optimiser (=accélérer) les programmes... il suffit d'attendre 2 ans / 4 ans et le matériel ira beaucoup plus vite...

Après la loi de Moore

Les améliorations matérielles sont marginales, il faut à nouveau améliorer les programmes...

Cela ouvre d'autres champs : recyclage, intégration de multiples composants sur une seule puce etc.

Évolutions récentes des processeurs : de multiples coeurs.

Comment contourner la loi de Moore ?

Une piste possible : utiliser plusieurs CPU

Mais qu'est qu'un coeur dans un microprocesseur ?

Un coeur est principalement composé :

- d'une UAL,
- de registres (R0, R1...)
- d'une unité de commande

Un coeur est donc capable d'exécuter des programmes de façon autonome.

Plusieurs coeurs sur une seule puce

Aujourd'hui (en 2019) on trouve sur le marché (=dans votre poche) des CPU possédant jusqu'à 18 coeurs !

Plusieurs coeurs = de meilleures performances ?

Pas forcément !

Tirer profit d'un CPU multicoeur demande certaines techniques de programmation...

Parallélisme

Les différents coeurs d'un CPU se partagent l'accès à la mémoire vive

L'accès aux ressources devient un enjeu. L'état des ressources aussi.

Parallélisme

- Mémoire

Adresse	Valeur
123	50
...	...

- coeur 1 : exécute les instructions :
 - lire la mémoire à l’adresse *123*
 - ajouter cette valeur et 1000
 - écrire le résultat à l’adresse *123*
- coeur 2 : exécute les instructions :
 - lire la mémoire à l’adresse *123*
 - si la valeur est > 100 , afficher “*bonjour*”
 - sinon, afficher “*au revoir*”

Parallélisme

- Si les instructions s’exécutent dans l’ordre :
C1, C2, C1, C2, C1, C2, etc
 on affiche : “*bonjour*”
- Si les instructions s’exécutent dans l’ordre :
C1, C1, C1, C2, C2, C2, etc.
 on affiche “*au revoir*”

Parallélisme

Comment contrôler l’ordre dans lequel les instructions sont réalisées sur plusieurs coeurs ?

Les processus communiquent entre eux par message...

(rassurez vous, c’est hors programme)