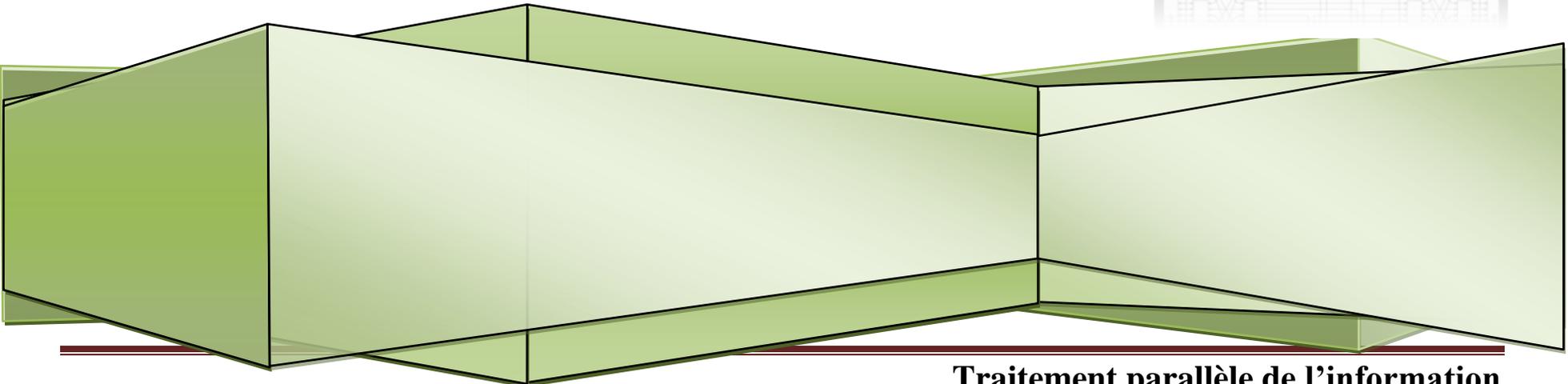
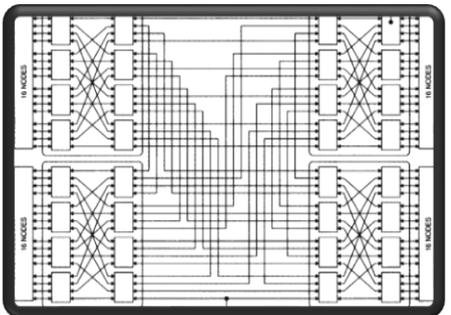


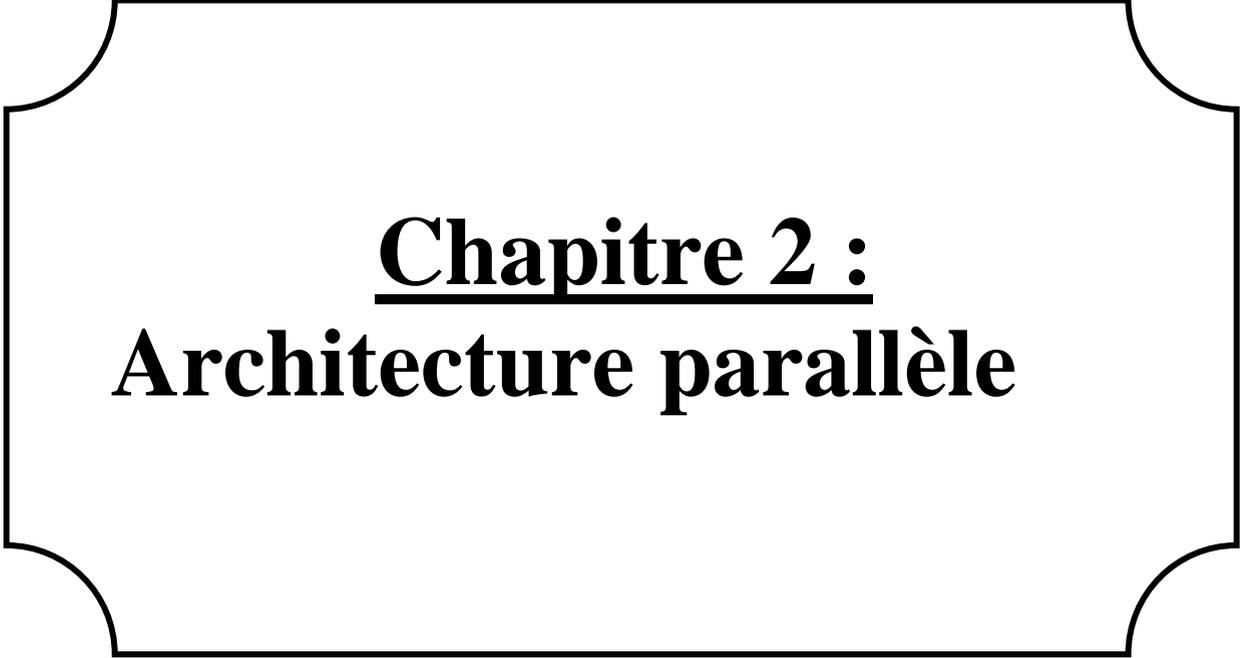
Traitement parallèle de l'information

Dr Maziz Asma

Cours

Master I : Vision Artificielle





Chapitre 2 :
Architecture parallèle

Plan

Chapitre 2 : Architecture parallèle
2.1. Types d'architectures parallèles.....
2.1.1. Les machines à mémoire partagée.....
2.1.2. Les machines à mémoire distribuée.....
2.2. Classification de Flynn
2.2.1. Les machines SISD
2.2.2. Les machines SIMD
2.2.3. Les machines MISD
2.2.4. Les machines MIMD
2.3. Classification de REINA

2.1. Types d'architectures parallèles

On peut classer les machines parallèles suivant leur efficacité, leur nombres de processeurs, le type des processeurs/réseaux/mémoires...etc.

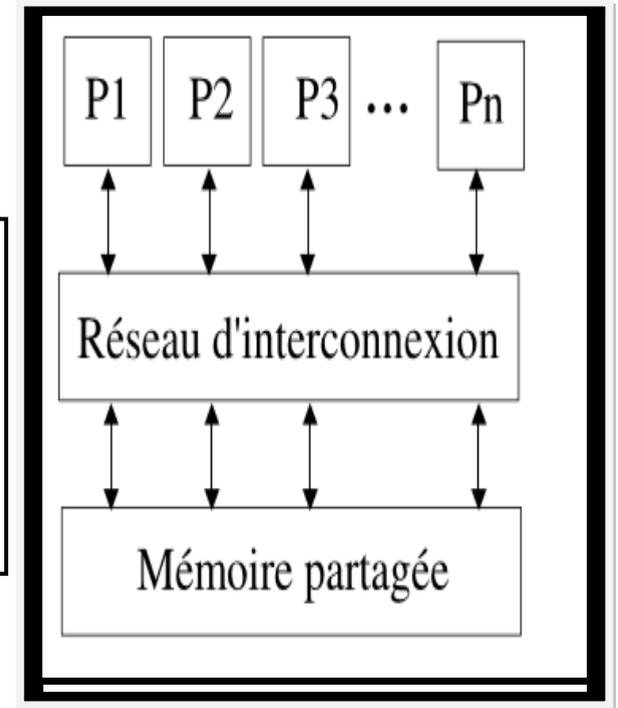
Il existe deux types de machine parallèle :

- ❖ Les machines parallèles à mémoire partagée
- ❖ Les machines parallèles à mémoire distribuée

2.1.1. Les machines parallèles à mémoire partagée

a) Caractéristiques :

- ✓ Plusieurs processeurs avec des horloges indépendantes
- ✓ Une seule mémoire commune à tous les processeurs
- ✓ Programmable par le standard portable OpenMP



Chapitre 2 : Architecture parallèle

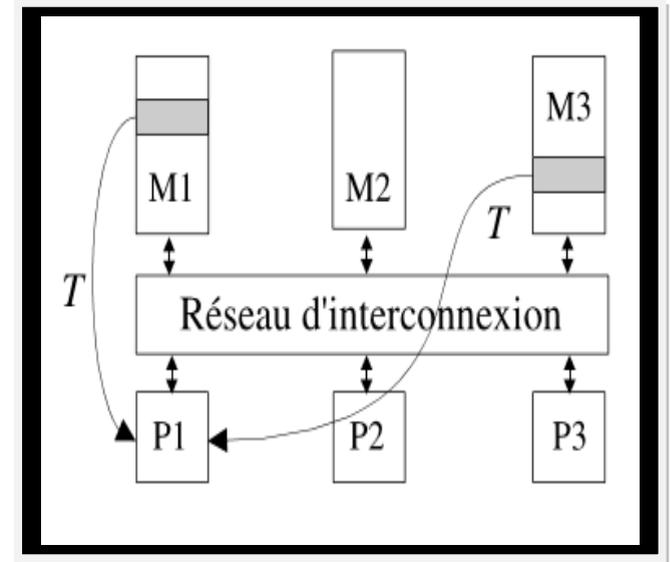
Dans les machines parallèles à mémoire partagée, le programmeur n'a pas besoin de spécifier la distribution des données sur chaque processeur. Il définit seulement la partie du programme qui doit être parallélisée (directives) et doit gérer les synchronisations.

Chapitre 2 : Architecture parallèle

Il existe plusieurs **organisations** possibles des machines à mémoire partagée :

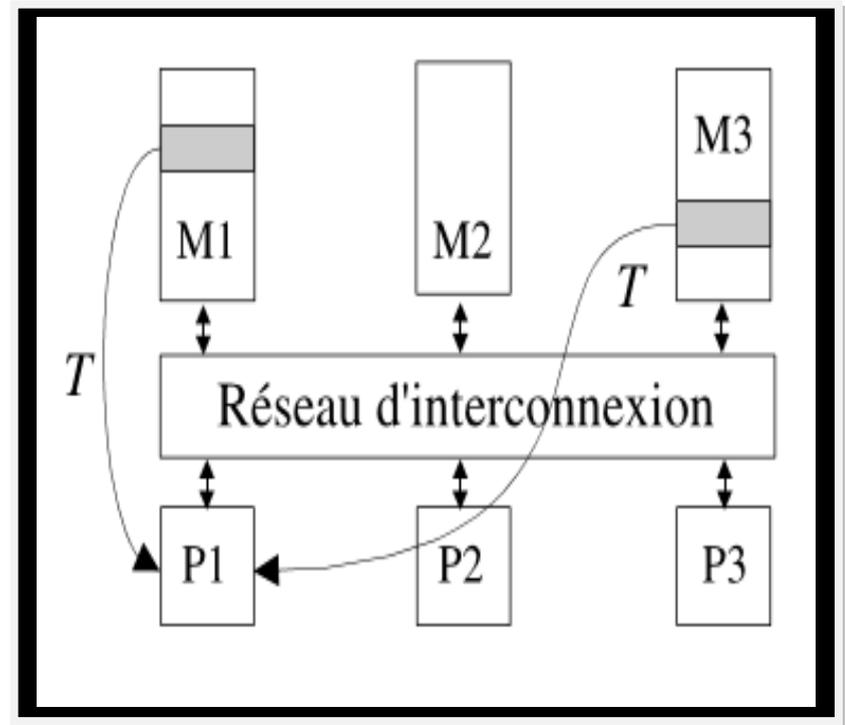
- **NORMA (No Remote Memory Access):**

- Il n'y a pas de moyen d'accès aux données distantes.
- La communication inter-processeurs se fait par passage de messages.



- **UMA (Uniform Memory Access) :**

- Les processeurs accèdent à la mémoire de manière symétrique.
- Le coût d'accès à la mémoire est identique pour tous les processeurs.



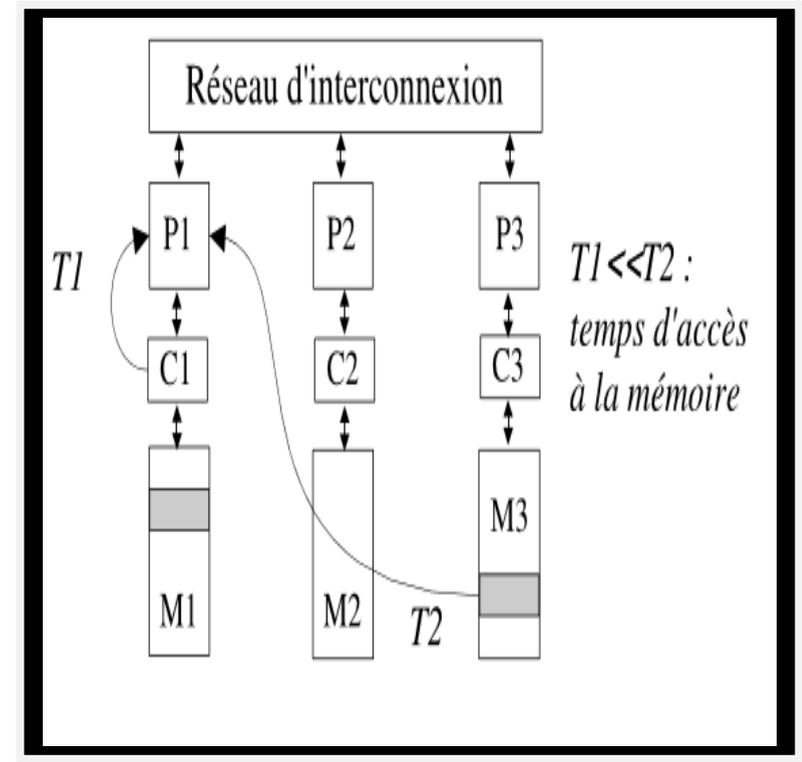
Chapitre 2 : Architecture parallèle

- **NUMA (Non Uniform Memory Access) :**

- Les performances d'accès dépendent de la localisation des données.
- La majorité des machines parallèles appartiennent à cette classe car l'accès en temps constant à la mémoire est une contrainte technologique difficile à réaliser.

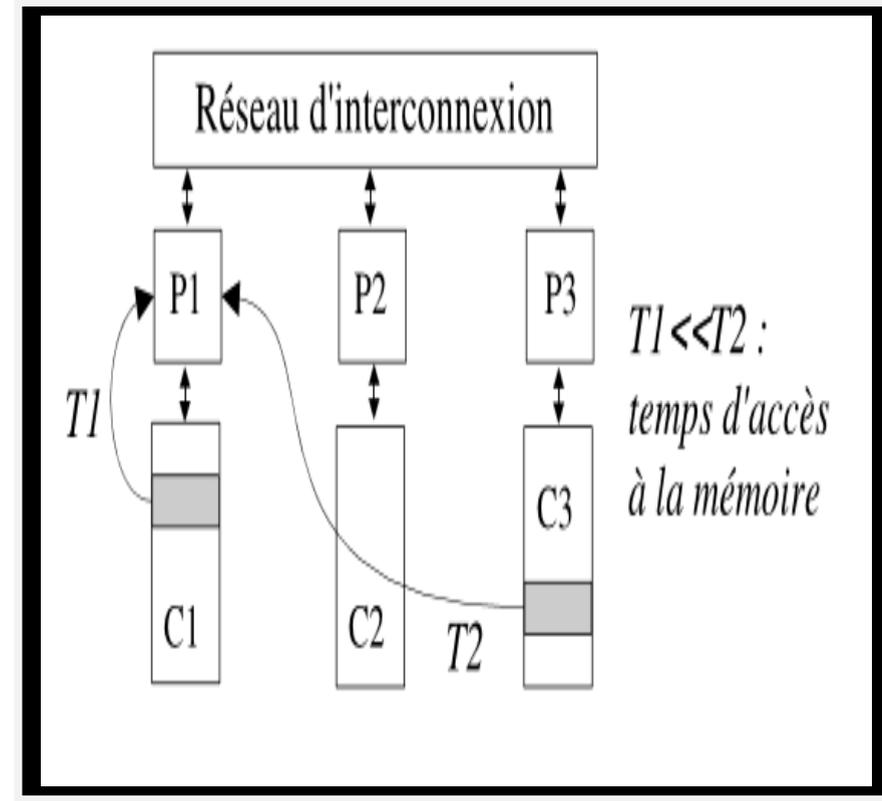
- **CC-NUMA (Cache Coherent – NUMA) :**

- Type d'architecture NUMA intégrant des caches.



- **COMA(Cache Only Memory Architecture):**

- Les mémoires localise comportent comme des caches, de telle sorte qu'une donnée n'a pas de processeur propriétaire ni d'emplacement déterminé en mémoire.



b) **Avantages :**

- Simplicité d'utilisation
- Parallélisation de haut niveau
- Portabilité au SMP (architecture homogène à mémoire partagée)
- Scalabilité

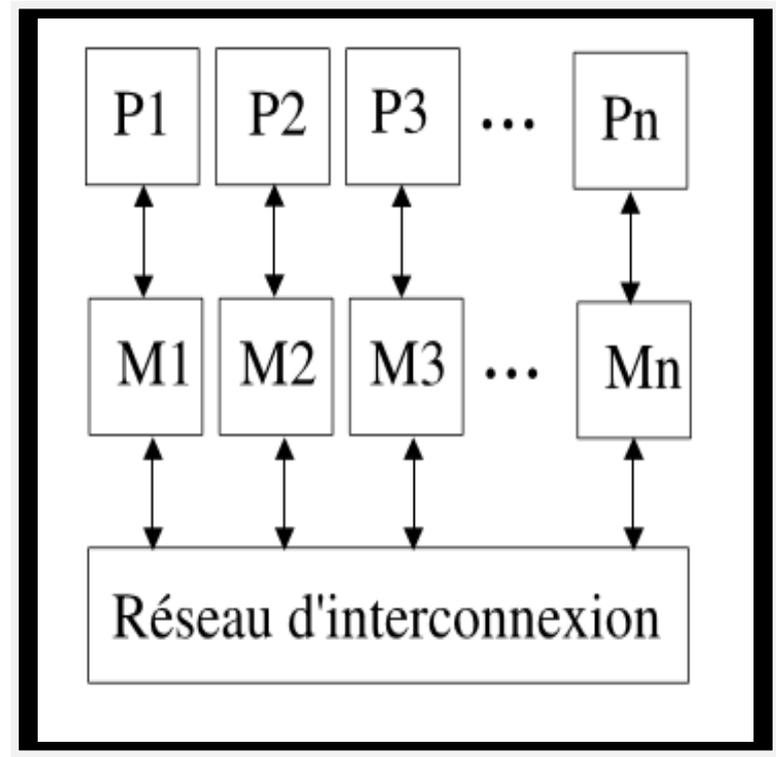
c) **Inconvénients :**

- Coût
- Limitation du nombre de processeurs (conflit d'accès au niveau matériel)
- La bande passante du réseau est le facteur limitant de ces architectures

2.1.2. les machines parallèles à mémoire distribuée

a) Caractéristiques :

- Interconnexion (réseau local rapide) de systèmes indépendants (noeuds).
- mémoire propre locale à chaque processeur.
- chaque processeur exécute des instructions identiques ou non, sur des données identiques ou non.
- Parallélisme par échange de messages.



- ❑ Les communications entre processeurs sont réalisées par l'appel à des fonctions de bibliothèque standard :
 - PVM (Parallel Virtual Machine) ou
 - MPI (Message Passing Interface).

b) **Avantages :**

- ✓ Non limité théoriquement en nombre de processeurs
- => Grande puissance de calcul
- ✓ Réalisation aisée par la conception de cluster mise en réseau d'un certain nombre de machines
- ✓ Partage transparent des ressources
- ✓ Portabilité des applications MPI

c) **Inconvénients :**

- ✓ Plus difficile à programmer que les machines à mémoire partagée
- ✓ Efficacité limitée.....

2.2. Classification de Flynn

Il existe plusieurs classifications des ordinateurs parallèles, les plus connus : Flynn, Kuck, Treleaven, Gajski, Feng, Handler,

- ❑ La classification la plus largement utilisée, depuis 1966, est la taxonomie de Flynn, ce qui nous intéresse dans ce chapitre.
- ❑ La classification de Flynn est une classification proposée par Michael J. Flynn en 1966, spécifique aux architectures informatiques parallèles,
- ❑ Ce type de classifications est basé sur le nombre d'instructions simultanées (OPCODE : addition, soustraction, multiplication ...) et de flux de données disponibles dans l'architecture (les variables),
- ❑ La séquence d'instructions lues en mémoire est appelée un flux d'instructions (Instruction Stream) et les opérations effectuées sur les données dans le processeur sont appelées flux de données (Data Stream),

Chapitre 2 : Architecture parallèle

- ❑ Donc, on peut aussi classer les machines parallèles suivant les programmes qui y sont utilisés (classification de Flynn).
- ❑ Classification de Flynn basée sur le type d'organisation du flux de données et du flux d'instructions.

Pour cela, il est distingué les types suivants :

	Single Instruction	Multiple Instructions
Single Data	SISD	MISD
Multiple Data	SIMD	MIMD

2.2.1. Les machines SISD

“Single Instruction, Single Data”

Cette classe est en série et ne peut pas être dite parallèle car on n'a pas de plusieurs processeurs. Il s'agit d'un ordinateur séquentiel qui n'exploite aucun parallélisme, tant au niveau des instructions qu'au niveau de la mémoire. C'est la machine la plus conventionnelle avec un processeur contenant une seule unité arithmétique - logique. L'ordinateur SISD et les ordinateurs séquentiels sont identiques

Donc, chaque machine SISD est un :

- Un ordinateur séquentiel qui n'exploite aucun parallélisme,
- Modèle de Von Neuman (1945),
- Chaque opération s'effectue sur une donnée à la fois.

Chapitre 2 : Architecture parallèle

UC = Unité de Contrôle

(Chargée du séquençage des instructions)

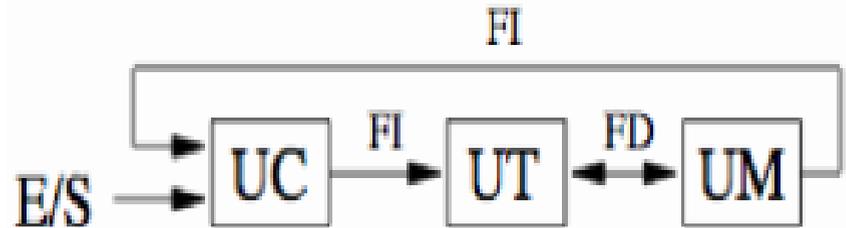
UT = Unité de Traitement

(Effectue les opérations)

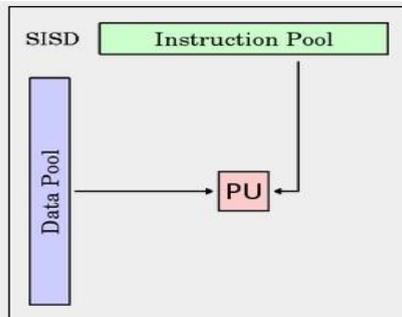
FI = Flot d'Instructions

UM = Unité Mémoire (contient les instructions et les données)

FD = Flot de Données



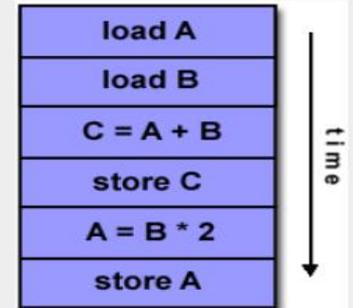
Exemples des machines SISD



PU : unité de calcul

Instruction Pool : flux d'instructions

Data Pool : flux de données



UNIVAC1



IBM 360



CRAY1



CDC 7600



PDP1



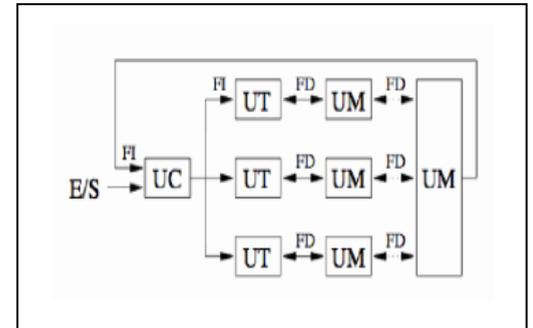
Dell Laptop

2.2.2. Les machines SIMD

“Single Instruction, Multiple Data”

Sont des machines **synchrones** :

- Très grand nombre d'éléments de calcul (4096 à 65536) de faible puissance avec une toute petite mémoire locale.
- Machine multiprocesseur capable d'exécuter la même instruction sur toutes les CPU mais fonctionnant sur des flux de données différents.
- En général, la façon de mettre en œuvre des architectes SIMD est d'avoir un processeur principal qui envoie des instructions à exécuter simultanément un ensemble d'éléments de traitement qui assurent les exécuter.



Chapitre 2 : Architecture parallèle

- ✓ Modèle SIMD convient particulièrement bien aux traitements dont la structure est très régulière, comme c'est le cas pour le calcul matriciel.
- ✓ Les applications qui profitent des architectures SIMD sont celles qui utilisent beaucoup de tableaux, de matrices, ou de structures de données du même genre.
- ✓ On peut notamment citer les applications scientifiques, ou de traitement du signal.
- ✓ Machine SIMD très courantes parmi les premières machines parallèles mais de moins en moins usité de nos jours.

Motivations initiales pour machines SIMD:

- ✓ Obtenir une machine parallèle de cout faible en dupliquant uniquement les unités d'exécution sans dupliquer l'unité de contrôle.
- ✓ Minimiser l'espace mémoire requis en ayant une seule copie de code.
- ✓ Le masquage des temps d'attente de la mémoire.
- ✓ Une programmation compacte qui reste lisible.

Chapitre 2 : Architecture parallèle

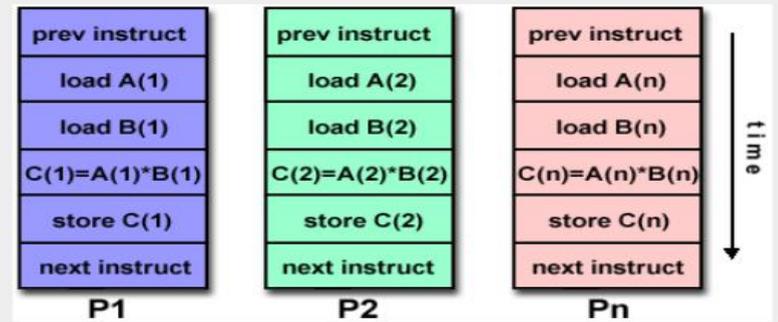
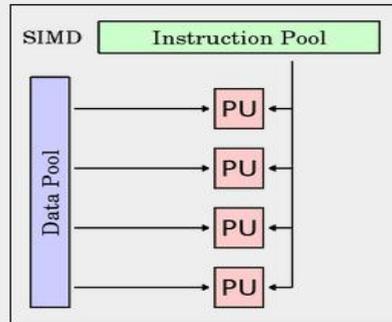
Les opérations peuvent être :

- Des opérations bit à bit (*et, ou, not ...*)
- Des additions, des soustractions, des multiplications, éventuellement des divisions.
- Des opérations mathématiques plus complexes.

Toutes ces instructions SIMD travaillent sur un ensemble de données de même taille et de même type.

Ces données sont rassemblées dans des espèces de blocs de données, d'une taille fixe, qu'on appelle un vecteur. Ces vecteurs contiennent plusieurs nombres entiers ou nombres flottants placés les uns à côté des autres.

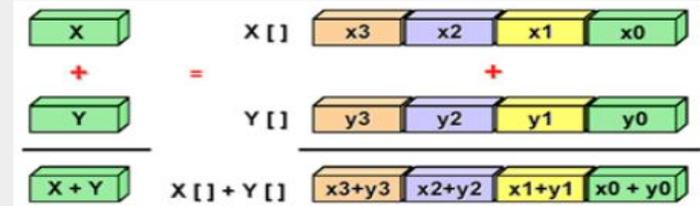
Exemples des machines SIMD



ILLIAC IV



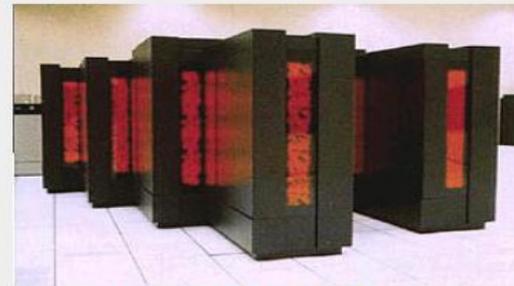
MasPar



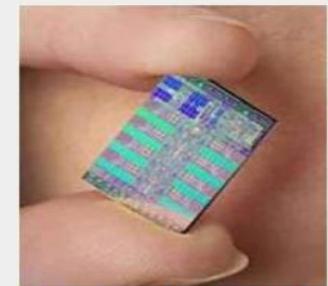
Cray X-MP



Cray Y-MP



Thinking Machines CM-2



Cell Processor (GPU)

Exemples de machines SIMD

✓ Illiac IV (1972):

Universitaire, 64 processeurs de 64 bits, 16 ko par processeur, en grille.

✓ STARAN de Goodyear (1972):

256 processeurs, 32 octets par processeur,

✓ DAP d'ICL (Distributed Array Processor) (1980):

4k processeurs à un bit, 512 octets par processeur,

✓ MPP de Goodyear (Massively Parallel Processor) (1982):

16k processeurs à un bit 128 octets par processeur, réseau à deux dimensions.

✓ MP-1 de Maspar (1989):

16k processeurs de 4 bits, 16 à 64 ko par processeur,

✓ MP-2 de Maspar:

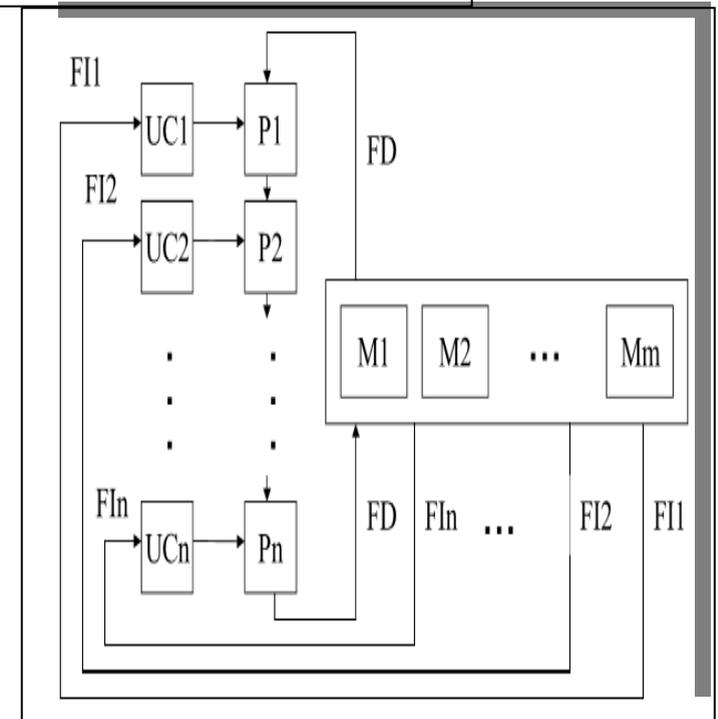
16k processeurs de 32 bits, 64ko par processeur.

2.2.3. Les machines MISD

“Multiple Instructions, Single Data”

Les machines vectorielles multi-processeurs :

- ✓ Peut exécuter plusieurs instructions en même temps sur la même donnée (processeurs vectoriels et architectures pipelines).
- ✓ Faible nombre de processeurs puissants (1 à 16)
- ✓ Mémoire partagée.
- ✓ Limite atteinte, coût important.



Chapitre 2 : Architecture parallèle

- ❑ Dans une architecture MISD, chaque processeur exécute une séquence différente d'instructions sur les mêmes données utilisées par les autres processeurs.
- ❑ Dans une architecture MISD, plusieurs flux d'instructions sont appliqués sur un seul flux de données, et par conséquent une seule séquence de données est transmise à un ensemble de processeurs.
- ❑ Théoriquement, certains auteurs considèrent que cette classe n'est pas commercialisée et d'autres auteurs considèrent le pipeline comme un schéma MISD.
- ❑ Par conséquent, l'architecture MISD reste un modèle théorique car il n'existe aucune machine construite sur ce modèle.

2.2.4. Les machines MIMD

“Multiple Instructions, multiple data”

-Désigne les machines multiprocesseurs où chaque processeur exécute son code de manière asynchrone et indépendante.

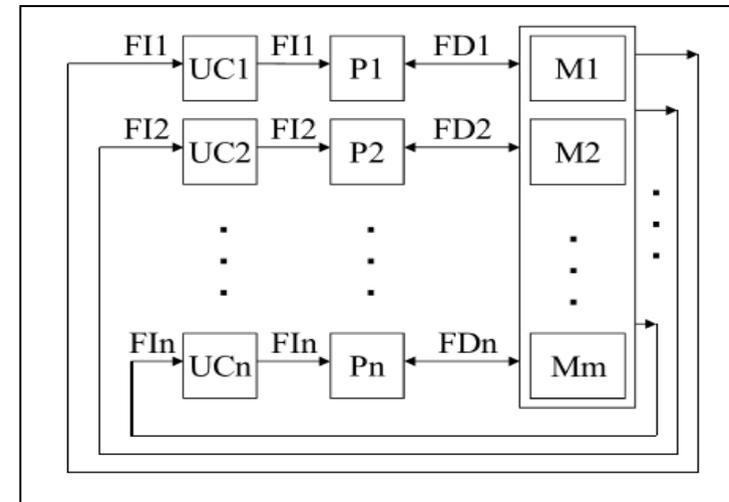
-Plusieurs processeurs traitent des données différentes,

-Chaque processeur possède une mémoire distincte.

-L'architecture parallèle la plus utilisée.

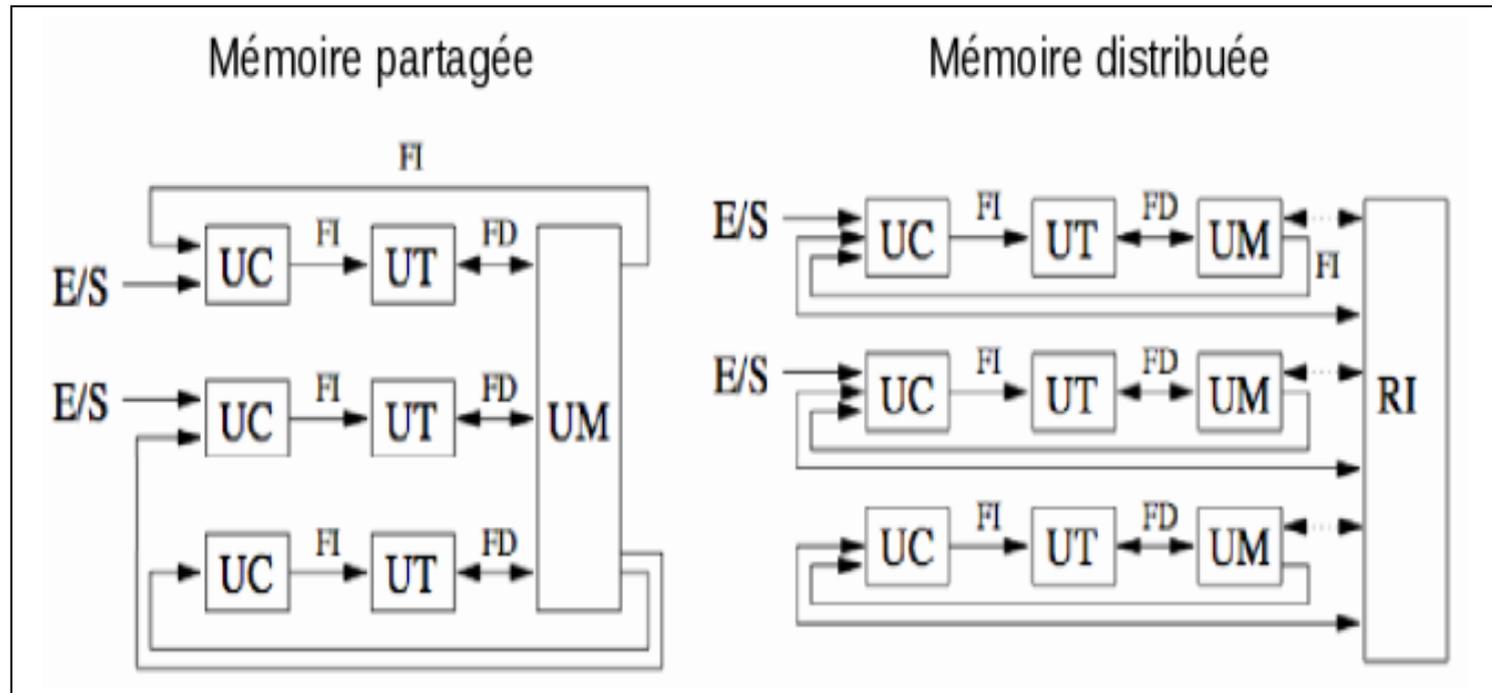
-Pour assurer la cohérence des données, il est souvent nécessaire de synchroniser les processeurs entre eux,

-Les techniques de synchronisation dépendent de l'organisation de la mémoire.



Chapitre 2 : Architecture parallèle

Deux sous classes :

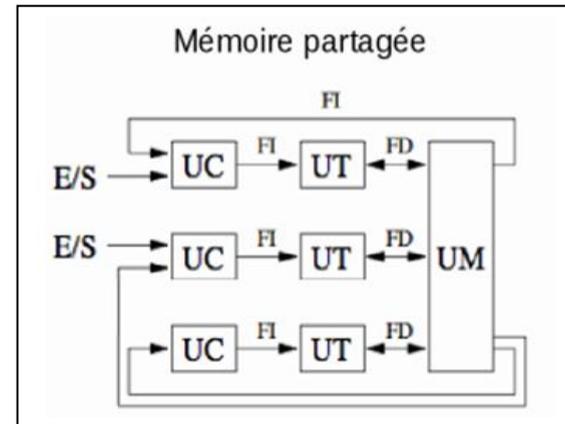


Chapitre 2 : Architecture parallèle

1) MIMD à mémoire partagée:

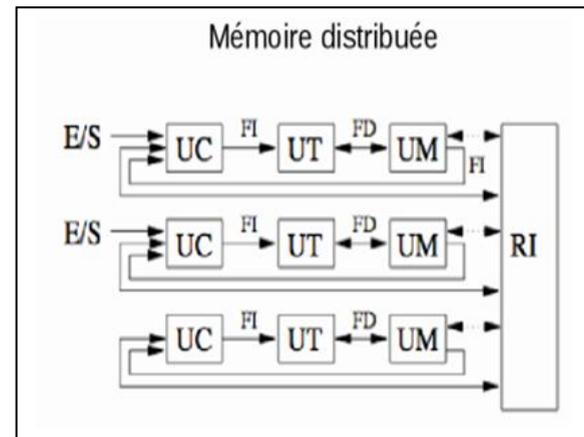
Les processeurs accèdent à une mémoire commune : la synchronisation peut se faire au moyen de:

- ❖ Sémaphores
- ❖ Verous, ou Mutex (exclusion mutuelle)
- ❖ Barrières de synchronisation

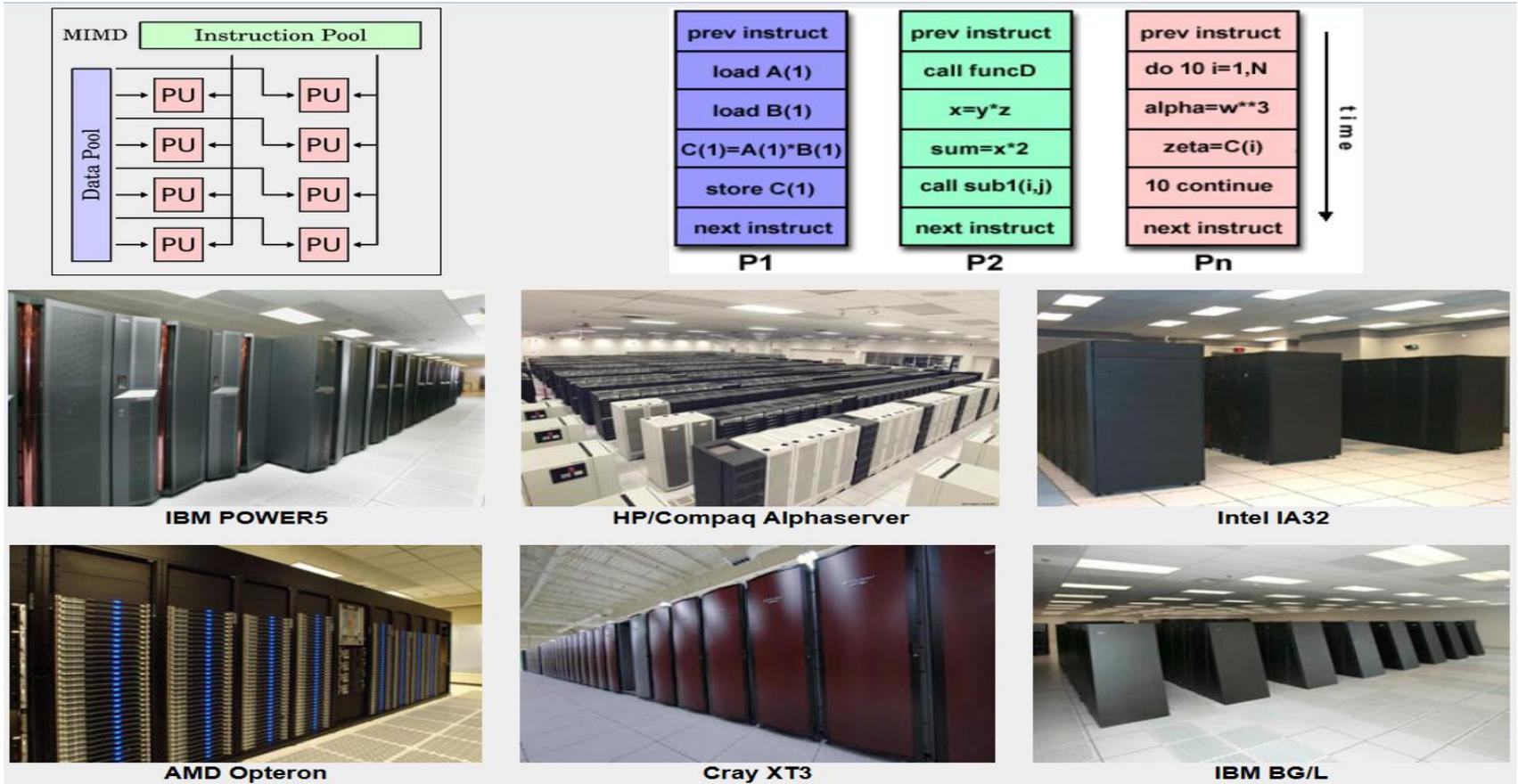


2) MIMD à mémoire distribuée:

Chaque processeur dispose de sa propre mémoire, et n'y a pas d'accès à celle des autres processeurs.



Exemples des machines MIMD :



2.3. Classification de REINA

- **Prise en compte de l'espace d'adressage**

2.3.1. SASM (Single Address space, Shared Memory):

- ✓ mémoire partagée.
- ✓ Les processeurs ont accès à la mémoire comme un espace d'adressage global.
- ✓ Tout changement dans une case mémoire est vu par les autres processeurs.
- ✓ La communication et la synchronisation inter-processeurs sont effectués via la mémoire globale (variables partagées).

2.3.2. DADM (Distributed Address space, Distributed Memory):

- Chaque processeur a sa propre mémoire et son propre système d'exploitation.
- Mémoire distribuée, sans accès aux données distantes,
- L'échange de données entre processeurs s'effectue nécessairement par passage de messages, moyen d'un réseau de communication.

2.3.3. SADM (Single Address space, Distributed Memory):

- ✓ Mémoire distribuée, avec espace d'adressage global,
- ✓ Autorisant éventuellement l'accès aux données situées sur d'autres processeurs.
- ✓ C'est l'architecture la plus utilisée par les superordinateurs.
- ✓ Elle combine les caractéristiques des deux classes précédentes.

Chapitre 2 : Architecture parallèle

