

ARCHITECTURE PIPE-LINE

Principe

L'architecture Pipe-line généralise le concept de recouvrement entre instruction car l'autre architecture présente l'exécution des instructions avec un aspect séquentiel.

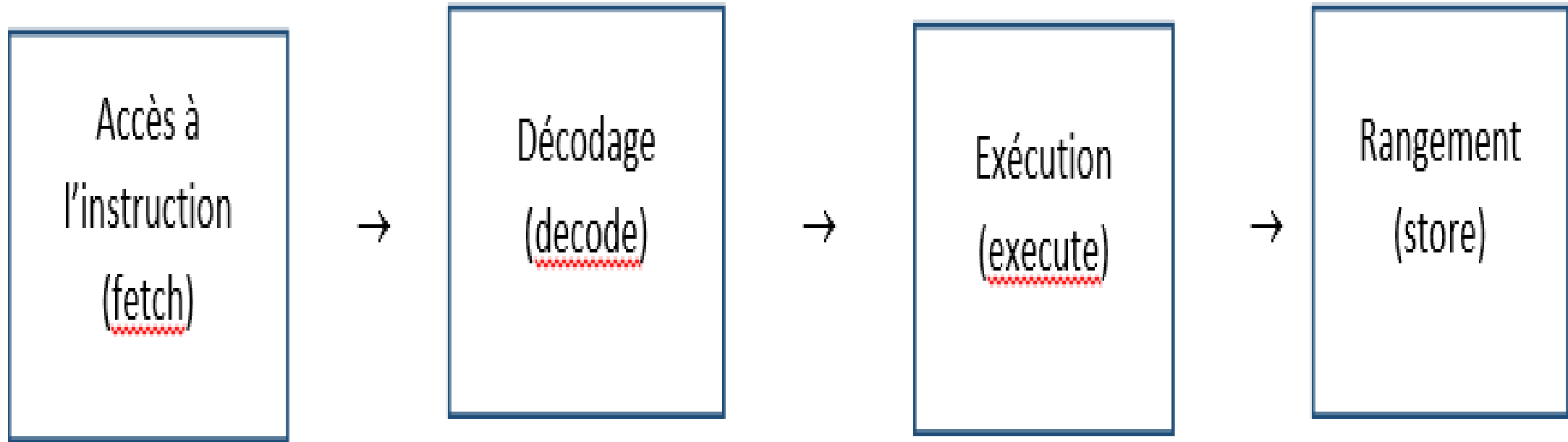
Les activités au sein du processeur sont organisées comme sur une chaîne de montage : on découpe l'unité de commande chargée de lecture des instructions, de leur décodage, de leur exécution, en plusieurs modules fonctionnellement « disjoints ».

Chacun de ces modules travaille en parallèle avec les autres. On commence ainsi le traitement d'une nouvelle instruction avant que la précédente ne soit terminée. Ce qui crée un flot continu en entrée de chacun des modules.

On considère la décomposition suivante, en quatre modules fonctionnellement indépendants :

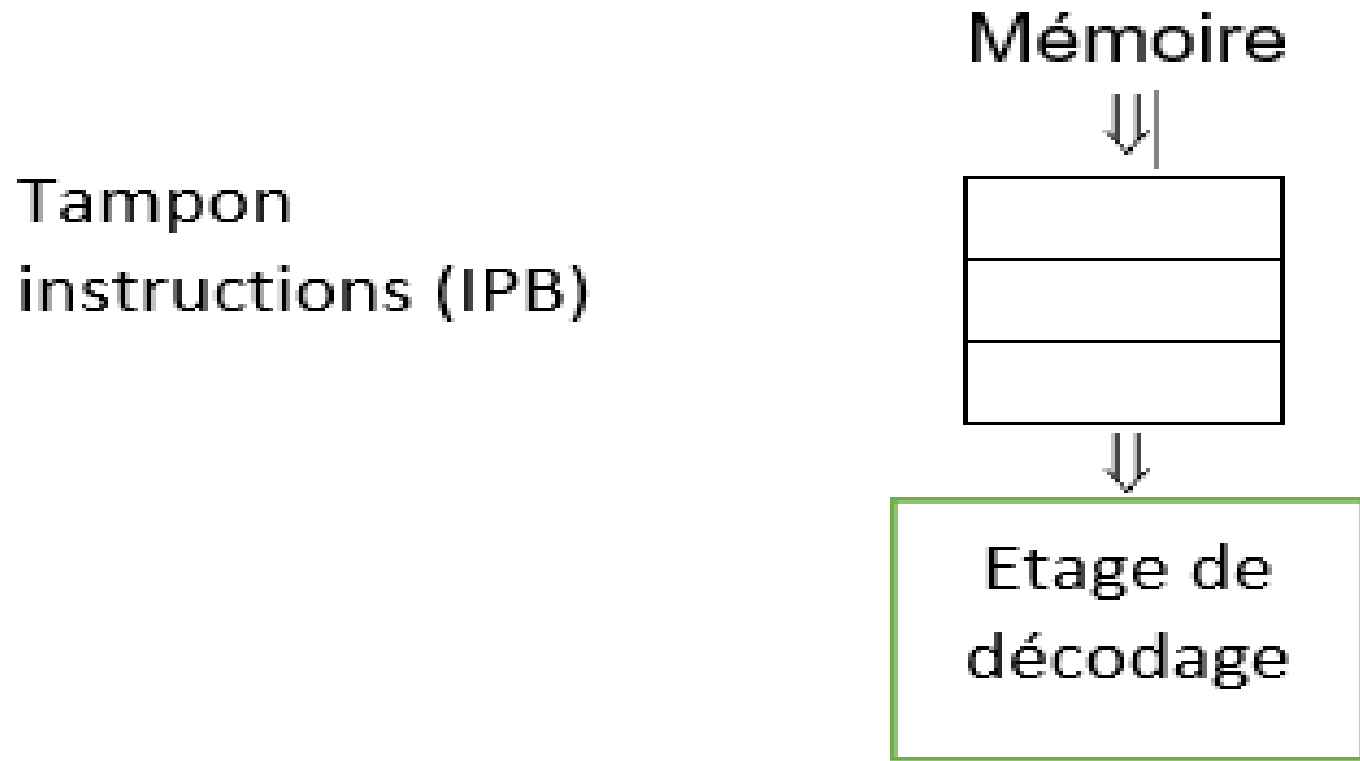
- Le module d'accès aux instructions (fetch) ;
- Le module de décodage (decode) ;
- Le module d'exécution (execute) ;
- Le module de rangement du résultat (Store).

Ainsi, le déroulement d'une instruction peut alors être schématisé :



Chacun des quatre modules est désigné par **étage**, et l'ensemble des étages par le terme **pipe-line**.

Le module d'accès aux instructions comporte le plus souvent un tampon auxiliaire (instruction prefetch buffer) destiné à recevoir les instructions.



Les instructions sont lues en avance dans la mémoire ou antémémoire. Ce tampon est géré comme une **pile FIFO** : lorsque l'une des instructions qui y sont présentes passe dans le module de décodage, l'entrée correspondante est libérée et une nouvelle instruction est lue en mémoire.

Définitions

Le cheminement des instructions I1, I2, I3, I4, I5, I6,... dans les quatre étages du pipeline se présente de la façon suivante :

Cycle	T0	T1	T2	T3	T4	T5	T6	T7	T8
<u>Fetch</u>	I1	I2	I3	I4	I5	I6			
<u>Decode</u>		I1	I2	I3	I4	I5	I6		
<u>Execute</u>			I1	I2	I3	I4	I5	I6	
<u>Store</u>				I1	I2	I3	I4	I5	I6
	Cycle majeur				Cycle mineur				

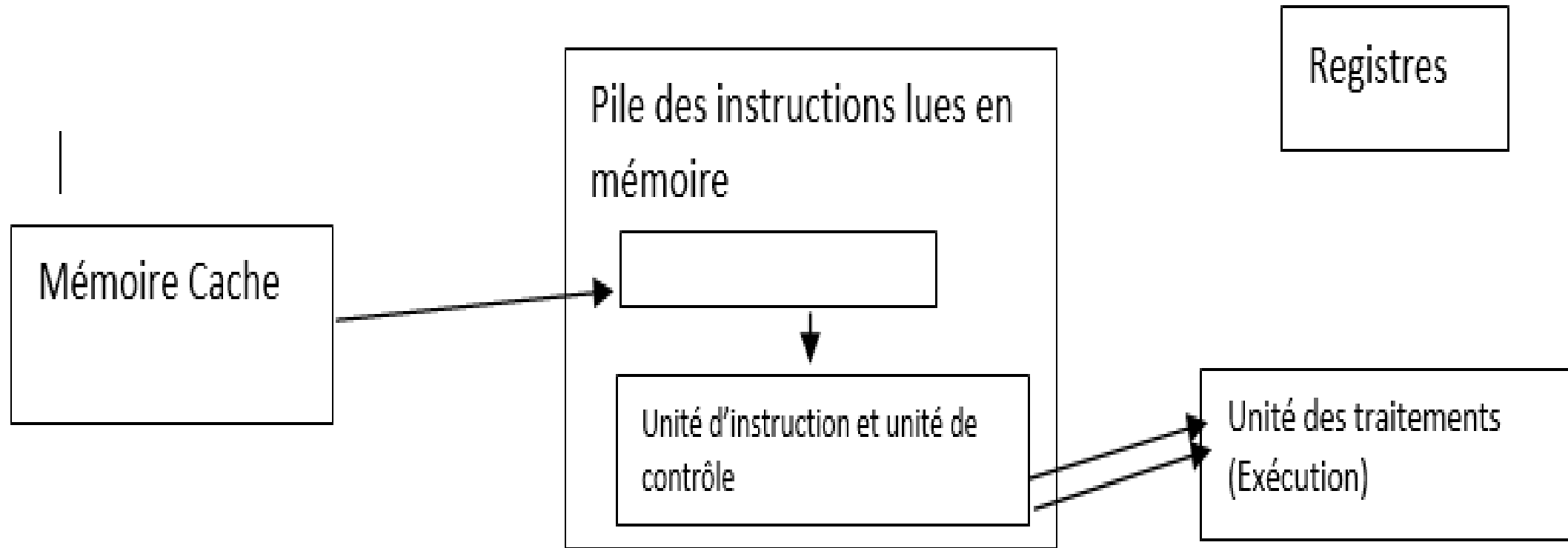
Le cycle mineur **C_m** est donné par la période de l'horloge de base.

Le cycle majeur **CM** est égal au produit du nombre d'étages **E** par le cycle mineur **CM=E*C_m**.

L'augmentation du nombre d'étages accroît le débit global au prix d'une plus grande complexité de la logique de contrôle.

Structure

Si l'on s'intéresse qu'à la partie instruction, la structure d'une machine à architecture pipe-line est la suivante :



Le processeur comporte une pile – le tampon d'instructions située entre la mémoire cache et d'unité d'instruction. Cette pile assure un débit régulier d'instruction en entrée du pipe-line.

L'unité d'exécution peut, elle aussi, être organisée sous forme de pipe-line.

Les modules pipe-line sont séparés par des registres tampons assurant le stockage des informations manipulées tout au long du traitement.

Les différents types de pipe-line

La technique pipe-line peut être étendu à tous les opérateurs présents :

- Pipe instruction, qui peut se décomposer comme suit : recherche instruction, décodage, recherche opérande, exécution, rangement du résultat ;
- Pipe scalaire, pour l'exécution des opérations sur les entiers ;
- Pipe flottant, pour l'exécution des opérations sur les nombres flottants ;
- Pipe vectoriel, pour l'exécution des opérations sur les opérandes de type vecteur.

Remarques :

- Quel que soit le type du pipe-line, la durée d'exécution d'une instruction d'une instruction ne change pas. L'amélioration du rendement de la machine en termes d'instructions traitées est uniquement dûe à un aménagement de son architecture.
- Il existe un parallélisme certain dans le pipe et même entre les pipes de différents niveaux, mais il n'y a toujours qu'un seul compteur ordinal, c'est-à-dire une seule unité de commande. Ce parallélisme ne doit pas être confondu avec celui d'une architecture multiprocesseur.

Les accès à la mémoire

Les problèmes posés par les accès à la mémoire cache dans une architecture pipe-line sont illustrées ici sur la machine vue précédemment.

La lecture en mémoire ou écriture en mémoire

On remarquera que la donnée lue en mémoire ne peut pas être utilisée par l'instruction qui suit, c'est ce qu'on appelle le problème de dépendance, alors le cycle fetch suivant est donc retardé. De même pour l'écriture.