

Pipeline

Découpage des instructions

- Découpage d’une instruction
- **Fetch** : Recherche d’instruction
  - **Decode** : Décodage instruction & lecture registres opérandes
  - **Execute** : Exécution de l’opération / calcul adresse mémoire
  - **Memory** : Accès mémoire / écriture PC de l’adresse de saut
  - **Write back** : Écriture dans un registre du résultat de l’opération

Calcul de métriques

Formule séquentielle vs pipeline

$T_e$  = temps de passage à chaque étape

$T_p = n * T_e$  = temps de passage pour n étapes

$T_t = m * T_p$  = temps total pour m personnes

$T_t = n * m * T_e$

Formule pipeline

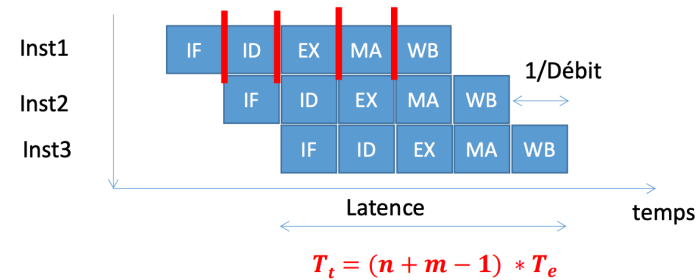
$T_t = n * T_e + (m - 1) * T_e$

Nombre de cycles d’horloges

$\frac{T_t}{T_e} = n + m - 1$

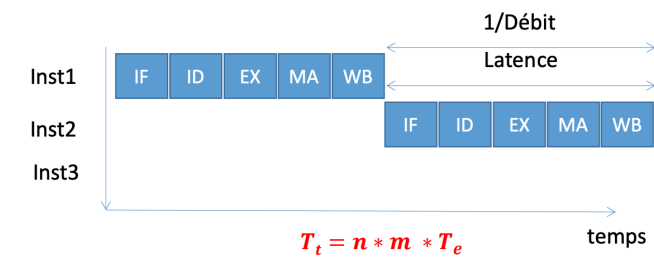
Débit

Nombre d’opérations (tâches, instructions) exécutées par unités de temps.



Latence

Temps écoulé entre le début et la fin d’exécution de la tâche (instruction).



IPC

Instructions Per Cycle : nombre d’instructions exécutées par cycle d’horloge.

$$IPC = \frac{1}{\text{ratio instruction sans arret} * 1 + \text{ratio instruction avec arret} * (\text{nb opération})}$$

Accélération

Accélération: nombre de fois plus vite qu’en séquentiel.

$$A = \frac{T_s}{T_p} = \frac{m * n * T_e}{(n + m - 1) * T_e} = \frac{m * n}{n + m - 1} \sim n \text{ (pour m très grand)}$$

- $T_t$  : temps total
- $m$  : nombre d’instructions fournies au pipeline
- $n$  : nombre d’étages du pipeline (MIPS, ARM = 5)
- $T_e$  : temps de cycle d’horloge ( $= \frac{1}{\text{Fréquence horloge}}$ )

Sans forwarding						
	1	2	3	4	5	6
ADD R1, R1, 1	IF	ID	EX	MA	WB	
ADD R2, R1, R0		IF	ID	EX	MA	WB
ADD R2, R1, R0		IF				ID
Avec Forwarding						
	1	2	3	4	5	6
ADD R1, R1, 1	IF	ID	EX	MA	WB	
ADD R2, R1, R0		IF	ID	EX	MA	WB

Pipelining aléas

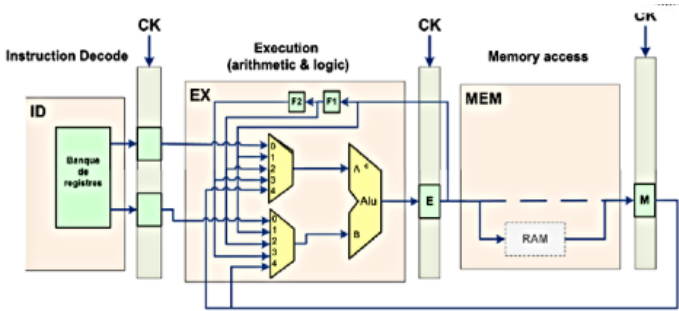
Résolution d’aléas

Arrêt de pipeline (hardware/software) **Hardware** : Arrêter le pipeline (stall/break).  
**Software** : Insérer des NOPs (no operation).

Sans forwarding

- Les règles de gestion des aléas sont les suivantes :
- Si l’instruction dépend d’une **valeur calculée par une instruction précédente** (RAW) nous devons attendre que l’opération **write-back** soit terminée.
  - Dans le cas où nous avons des dépendances de données (WAW ou WAR) cela n’impacte pas le pipeline.
- Attention** dans le cas d’aléas structurels, nous ne pouvons pas faire d’opération **Memory Access (M)** et **Fetch (F)** en même temps.

Forwarding



- Les règles de gestion des aléas sont les suivantes :
- Si l’instruction suivante dépend d’une valeur calculée par une instruction précédente, la valeur sera directement disponible dans le bloc **Execute**.

Taxonomie de Flynn

Classification basée sur les notions de flot de contrôle

- 2 premières lettres pour les instructions, I voulant dire Instruction, S - Single et M - Multiple
- 2 dernières lettres pour le flot de données avec D voulant dire Data, S – Single et M - Multiple

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

SISD

Machine SISD (Single Instruction Single Data) = Machine de «Von Neuman».

- Une seule instruction exécutée et une seule donnée (simple, non-structurée) traitée.

SIMD

Plusieurs types de machine SIMD : parallèle ou systolique.

- En général l’exécution en parallèle de la même instruction se fait en même temps sur des processeurs différents (parallélisme de donnée synchrone).

MISD

Exécution de plusieurs instructions en même temps sur la même donnée.

MIMD

Machines multi-processeurs où chaque processeur exécute son code de manière asynchrone et indépendante.

- S’assurer la cohérence des données,
  - Synchroniser les processeurs entre eux, les techniques de synchronisation dépendent de l’organisation de la mémoire.