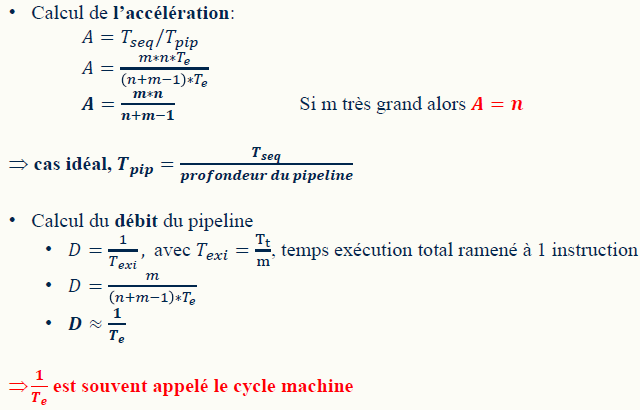
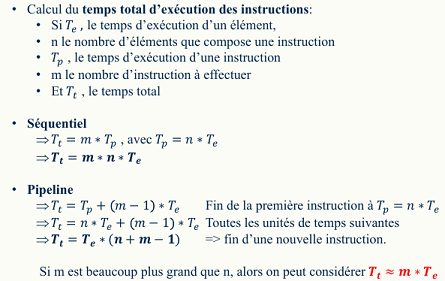
**Temps de cycle** : temps de traitement de chaque étage

**Latence** : temps d'attente du premier résultat



**Débit** : nombre d'instruction par seconde traité par le pipeline

Chaque registre est généralement isolé par un registre.

**Aléa structurel** : conflits d'accès aux ressources

**Aléa de donnée** : modifications de l'ordre d'accès aux ressources

**RAW** : lecture d'un registre où on récemment écrit

**WAR** : écriture d'un registre qu'une autre instruction utilise comme source

**WAW** : écriture dans un registre où on a récemment écrit

**Données**

**Méthode simple** : hardware : arrêter le pipeline, insérer des nops

**Forwarding ou bypassing** : Utiliser les résultats de l'alu sans attendre l'étape write back

Les opérandes sont stockés dans des registres Ces méthodes ne fonctionnent pas pour des pipelines à plus de 5 étages ou quand le temps de traitement varie selon l'instruction. Dès lors, on résout les aléas avec :

**Un ordonnancement dynamique** : On modifie l'ordre d'exécution pour éviter les aléas.

**Scoreboarding** : Les instruction sont lues dans l'ordre du programme mais exécutées selon une liste d'attente pour la disponibilité des opérandes. Bonne technique pour les RAW.

**Méthode Tomasulo** : Utilisation de registres temporaires, particulièrement bonne pour la résolution WAR et WAW

**Contrôle**

Arrêt de N instructions jusqu'à qu'on connaisse l'adresse de saut, changement de l'ordre des d'exécution des instructions, exécution des tests de branchement dans le decode

**Une prédiction dynamique** : Permet d'anticiper les branchements en utilisant les probabilités. L'adresse la plus probable est stockée dans la table. En cas de saut c'est cette adresse qui est chargée. La prédiction peut se faire sur un bit ou deux bits.