

Pipeline

ARO
8 - Pipeline

Résumé du document
TBD

Table des matières

- 1. Type de parallélisme 2**
 - 1.1. Parallélisme Bit-level 2
 - 1.2. Parallélisme d'instructions 2
 - 1.3. Parallélisme de donnée 2
 - 1.4. Parallélisme de tâches (threads) 2
- 2. Taxonomie de Flynn 3**
 - 2.1. SISD 3
 - 2.2. SIMD 3
 - 2.3. MISD 3
 - 2.4. MIMD 3
- 3. Le pipeline 4**
 - 3.1. Formule séquentielle vs pipeline 4
 - 3.2. Formule pipeline 4
 - 3.3. Niveau de pipeline 4
 - 3.4. Calcul des performances 5
 - 3.4.1. Latence 5
 - 3.4.2. Débit 5
 - 3.4.3. Accélération 5

1. Type de parallélisme

1.1. Parallélisme Bit-level

- Augmenter la taille des mots
- 64 bits (mais DDR2 de 256 bits)

1.2. Parallélisme d'instructions

- **Pipeline**
- Superscalaire : Le matériel est responsable à l'exécution du lancement parallèle des instructions
- VLIW (very long instruction word): Le compilateur est responsable de présenter au matériel des instructions exécutables en parallèle

1.3. Parallélisme de donnée

- Traitement vectoriel, images, matrices

1.4. Parallélisme de tâches (threads)

- Multithreading simultané

2. 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

2.1. SISD

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

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

2.2. 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).

2.3. MISD

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

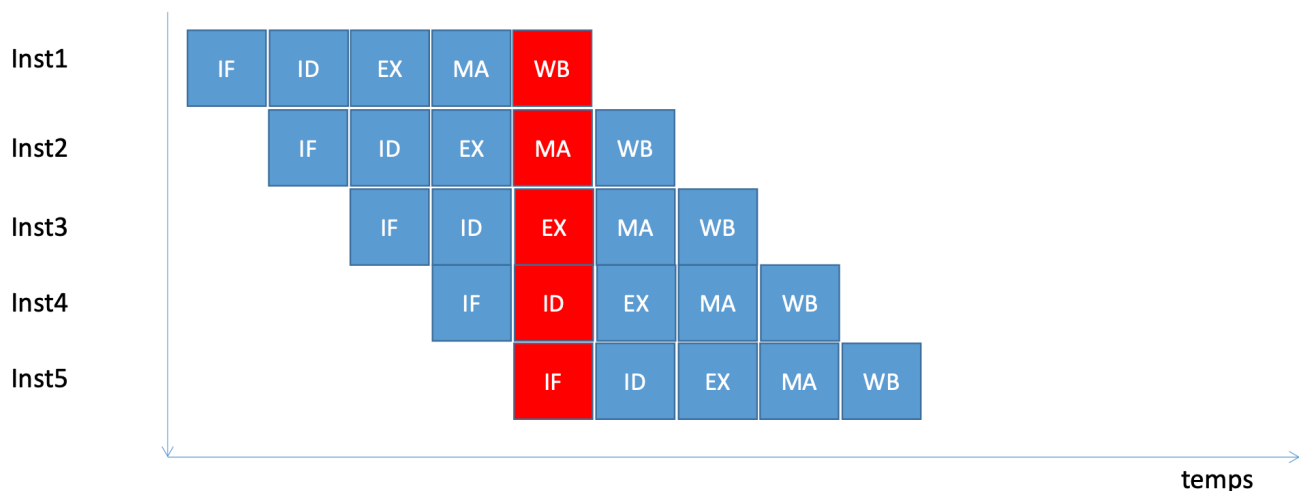
2.4. MIMD

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.

3. Le pipeline

- Permet d'accroître la vitesse d'exécution des instructions dans un micro-processeur.
- Découpe une tâche en plusieurs morceaux de manière à augmenter le débit.
- Les données progressent les unes derrière les autres.
- Permet de commencer une instruction avant la fin de la précédente.



3.1. 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

$$T_t = n * m * T_e$$

3.2. Formule pipeline

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

3.3. Niveau de pipeline

- Pipeline à 3 niveaux : Fetch, Decode, Execute
- Pipeline à 5 niveaux : Fetch, Decode, Execute, Memory, WriteBack

La principale raison d'augmenter le nombre de niveaux de pipeline est d'augmenter le débit d'instructions (IPC, Instructions Per Cycle) et donc d'améliorer les performances globales du processeur. En théorie, plus il y a d'étapes dans le pipeline, plus le processeur peut travailler sur plusieurs instructions en parallèle, ce qui permet d'augmenter le débit.

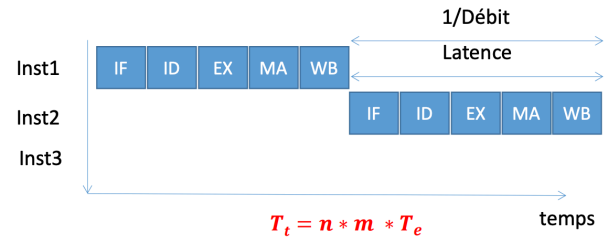
Un pipeline plus profond peut entraîner des retards supplémentaires dus à des bulles de pipeline (lorsque le pipeline doit attendre qu'une instruction précédente se termine avant de pouvoir continuer). De plus, un pipeline plus profond peut rendre plus difficile la gestion des dépendances de données, des sauts conditionnels, et des erreurs de prédiction de branchement.

Intel Pentium 4 Prescott	31
Intel Pentium 4	20
AMD K10	16
IBM POWER5	16
IBM PowerPC 970	16
Intel Core2 Duo	14
Intel Pentium II	14
Sun UltraSPARCIV	14
Sun UltraSPARCIi	14
AMD Opteron1xx	12
AMD Athlon	12
IBM POWER4	12
Intel Pentium III	10
Intel Itanium	10
MIPS R4400	8
Motorola PowerPC G4	7
ARM9	5
MIPS	5

3.4. Calcul des performances

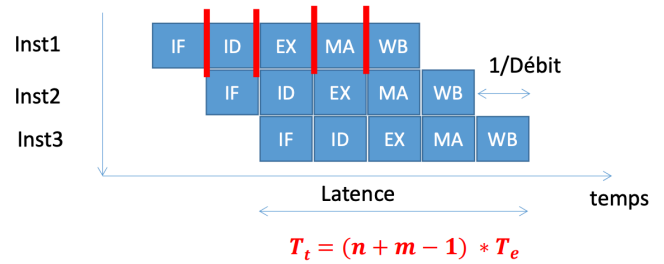
3.4.1. Latence

Latence : temps écoulé entre le début et la fin d'exécution de la tâche (instruction).



3.4.2. Débit

Débit: nombre d'opérations (tâches, instructions) exécutées par unités de temps.



3.4.3. Accélération

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

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