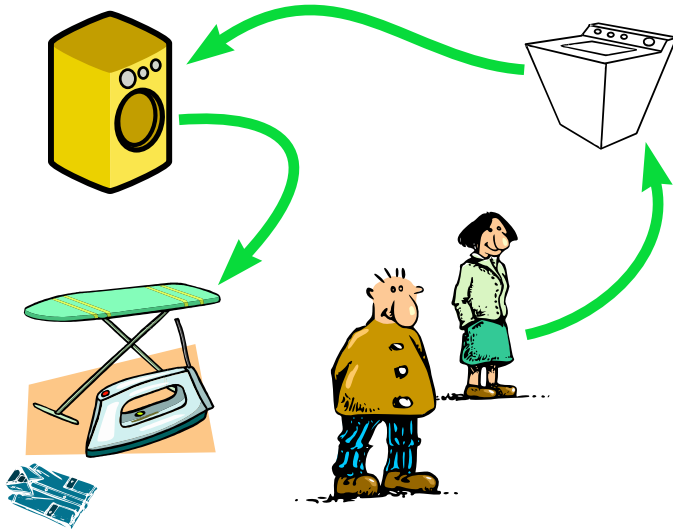


Architecture des ordinateurs (X31I050)

Frédéric Goualard

Laboratoire d'Informatique de Nantes-Atlantique, UMR CNRS 6241
Bureau 112, bât. 11
Frederic.Goualard@univ-nantes.fr

Pipelines



add \$r0, \$r1, \$r2
add \$r3, \$r4, \$r5

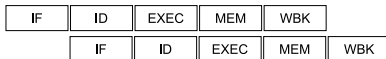
IF : Instruction Fetch
ID : Instruction Decode (+Registers Fetch)
EXEC : EXECution
MEM : MEMory Read/Write
WBK : Write Back



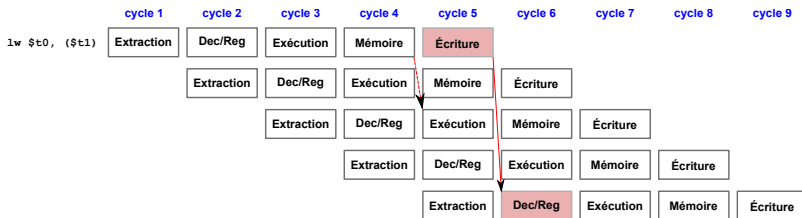
SANS PIPELINE



AVEC PIPELINE



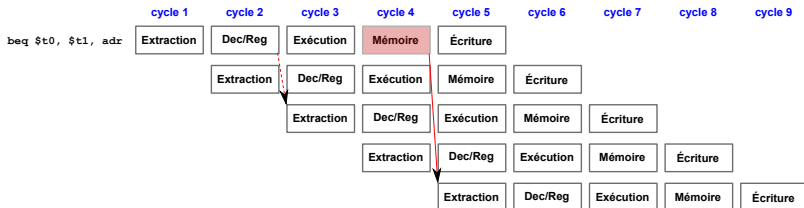
- ▶ Raccourcissement du temps de cycle
- ▶ Économe : unités fonctionnelles réutilisées
- ▶ Cinq unités indépendantes :
 - ▶ Extraction de l'instruction en mémoire (*Instruction Fetch*)
 - ▶ Décodage de l'instruction et lecture des registres (*Instruction Decode*)
 - ▶ Exécution de l'instruction par accès à l'UAL (*EXECution*)
 - ▶ Accès en lecture/écriture à la mémoire de données (*MEM*)
 - ▶ Accès aux registres en écriture (*Write Back*)



- ▶ Instruction de chargement extraite au cycle 1
- ▶ Valeur de `$t0` disponible seulement au cycle 6
 - ➡ Délai de trois instructions

Solutions possibles :

- ▶ Envoi de la donnée dès le cycle 4
- ▶ *Read after Write*



- ▶ Instruction de branchement extraite au cycle 1
- ▶ Adresse de destination écrite dans PC en fin de cycle 4
- ▶ Instruction de destination chargée au cycle 5
 - ➡ Délai de trois instructions

Solutions possibles :

- ▶ Envoi de la donnée dès le cycle 2 (ajout de matériel)
- ▶ Prédiction

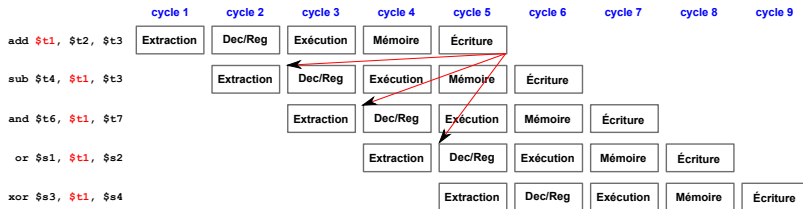
Aléas empêchent une instruction de s'exécuter dans son cycle

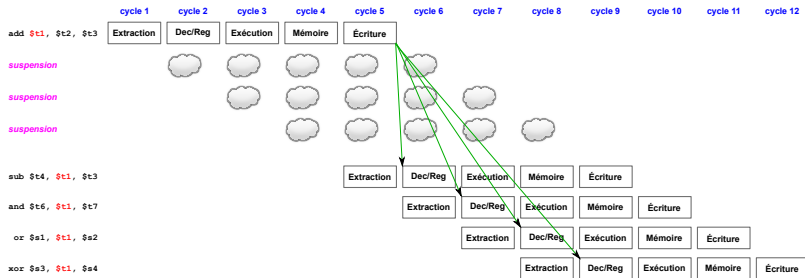
Aléas structurels. Le matériel ne supporte pas la combinaison d'accès (e.g., lecture et écriture en mémoire dans le même cycle)

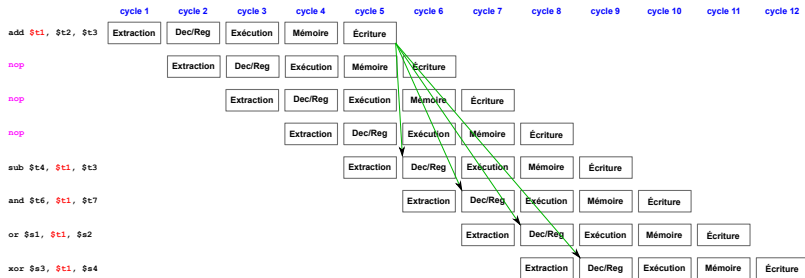
Aléas de données. Une instruction dépend du résultat d'une instruction dans le pipeline (e.g., cas du chargement)

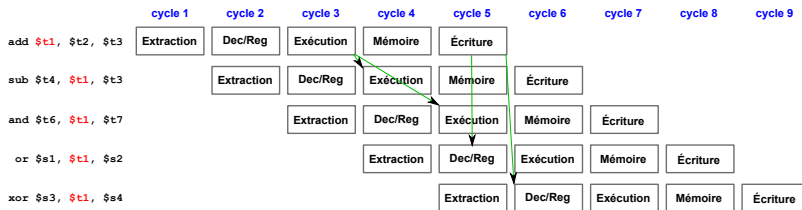
Aléas de contrôle. Des instructions suivant un branchement sont chargées sans que l'on en sache encore si elles doivent être exécutées

- ▶ Solution matérielle : bulles de suspensions
- ▶ Solution logicielle : ajout de nops
- ▶ Solution interne : envoi de données

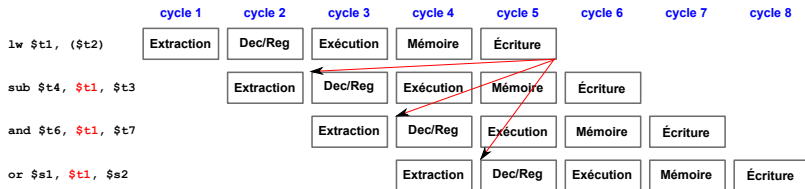


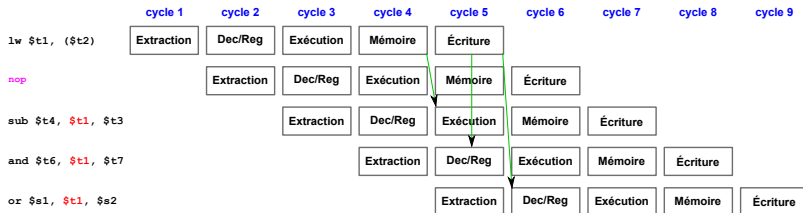






► *Read after Write*





- Envoi de donnée dès qu'elle est disponible dans le pipeline
- *Read-after-write*