Pipeline instructions

Principe d'un pipeline

- Il s'agit de reprendre le principe du travail à la chaine
- On découpe une tâche T en un nombre e de taches T_1, T_2, ... T_e
- On va noter t la durée de T, t_1 la durée de T_1, t_2 la durée de T_2, ... te la durée de T_e
- On veut exécuter un grand nombre de tâches A_1, A_2... A_N avec N grand

Idée directrice

- On va commencer à exécuter la tâche A2 avant d'avoir fini A1
- Conséquences

  - il fait cadencer le pipeline à la vitesse de la tache la plus lente

  - un pipeline idéal vérifie $t_1=t_2=...=t_e=T/e$

- Il faut N+e-1 tops pour exécuter les N tâches

Conséquences

- Durée sans pipeline d=NxT
- Durée avec pipeline D=(N+e-1)T/e
- Accélération=d/D=Ne/(N+e-1)=e/(1+(e-1)/N)

  si N grand Accélération=e
- Avec un pipeline à e étages on peut aller e fois plus vite

Optimiser l'accélération

- Il faut découper la tâche initiale en le plus grand nombre d'étages
- Il faut toutefois que chaque tâche élementaire ait la même durée pour que le pipeline soit idéal !

Application

- La tâche de base est l'exécution d'un instruction en assembleur.
- Il y a un très grand nombre d'instructions en assembleur à réaliser.
- Exemple de découpage Chargement-Décodage-Exécution
  Pipeline à 3 étages ==> on peut aller 3 fois plus vite
- Dans la réalité le pipeline instruction fait une vingtaine d'étages.

Conclusion

- Le pipeline instruction permet d'accélérer considérablementl'exécution d'un programme
- Il complique énormément la structure du processeur car la mise au point du pipeline idéal est complexe.