

Architecture pipeline

M. Dubacq

S1D 2009

1 Processeur à huit étages

Objectif : *Comprendre le principe de l'architecture pipeline.*

1. On regarde le cas d'un processeur à huit étages. Quelle est le nombre maximal d'instructions qui peuvent s'exécuter simultanément dans ce processeur ? *Jusqu'à huit instructions simultanées*
2. Le temps de cycle de ce processeur est de 0,5 ns. Quel est le temps nécessaire pour exécuter une instruction ? *4 ns.*
3. Quelle est la cadence d'instructions maximale qui est possible sur ce processeur ? Quelle est l'accélération maximale ? *Jusqu'à une instruction toutes les 0,5 ns, soit 2 milliards d'instructions. L'accélération maximale est de 8.*
4. Les étapes 6 et 7 sont consacrées à la lecture et à l'écriture dans la mémoire cache de données (d'abord initialisation, puis consultation du bus de données). Quelle doit être la valeur maximale du temps d'accès au cache en cas de succès ? *1 ns puisque l'accès au cache doit se faire dans les étapes 6 et 7.*
5. Quel est le temps d'exécution des cinquante premières instructions ? *Cinquante instructions : 28,5 ns.*

2 Dépendance d'instructions et de données

Objectif : *Comprendre le principe de la dépendance d'instructions et de la dépendances de données.*

Soit le programme suivant (t0 vaut 2 au départ, et v0 vaut 10) :

```
boucle: addi $t0,$t0,-1
        add $t1,$t1,$t2
        sll $t2,$t2,1
        bne $t0,$zero,boucle
        syscall
```

On se situe dans le cas du processeur à cinq étages étudié en cours. Le jeu d'instruction a une syntaxe simple : \$ introduit l'utilisation d'un registre, le résultat des opérations est écrit dans le *premier registre*, `bne` compare ses deux arguments et va à l'étiquette mentionnée si et seulement si ils sont différents (et sinon, va à l'instruction suivante) (et à la limite, on se moque un peu de ce que font exactement les instructions pour les besoins de cet exercice).

1. Est-ce qu'il y a une dépendance de données entre le t2 de la ligne 2 et celui de la ligne 3 ? Pourquoi ? *Non, car le registre t2 n'est pas modifié dans la ligne 2.*
2. Est-ce qu'il y a une dépendance de données entre le t0 de la ligne 1 et celui de la ligne 4 ? Pourquoi ? *Oui. La valeur de t0 est modifié à la ligne 1, et enregistré quand cette instruction finit l'étape WB de la ligne 1. Or, la ligne 4 sera alors en train de finir l'étape 2, et lira donc la mauvaise valeur (l'ancienne valeur de t0, et non pas la nouvelle).*
3. Est-ce qu'il est possible de résoudre les problèmes mentionnés plus haut avec une instruction NOP ? Est-ce qu'il reste des dépendances de données ? *Oui, n'importe où. Dans ce cas, l'étape 5 de la ligne 1 sera fini avant que l'étape 2 ne commence pour la ligne 4. Il ne restera pas de dépendances de données, car on écrit sur le registre t1 avec quatre instructions de décalage, et pareil pour t2 (il faut bien le vérifier à cause des boucles).*
4. Comment peut-on faire pour résoudre la dépendance d'instructions de la ligne 4 ? *On pourrait soit mettre plusieurs NOP (au moins quatre) juste après le test. Une autre possibilité est de faire de l'anticipation de saut, et de permettre de faire une purge. Ce serait possible ici si on fait de l'anticipation de saut en supposant que le saut ne se fait pas, car le syscall est très facile à rattraper (on suppose qu'il agit au moment de l'étape 5, et il ne fait rien d'autre). Par contre, il faut insérer des NOP après le syscall, pour être sûr qu'aucun effet négatif ne peut arriver en lisant des instructions juste derrière.*
5. En supposant qu'on anticipe que le saut ne se fait pas, sachant que le temps de cycle est de 1 ns, quel est le temps entre l'entrée de la première instruction et la fin de traitement de l'instruction syscall ? *Je note A le NOP*

rajouté dans la boucle et BCDE les instructions derrière le syscall, voici la liste des instructions qui rentrent dans le pipeline :

1A234 5BCDE 1A234 5BCDE (l'instruction 5 sort alors du pipeline et le programme s'arrête). Donc, un temps total de 20 ns pour ces instructions.

6. Quelle est l'accélération effective ? *5 ns par instruction : 123412345, soit 45 ns. L'accélération est de 45/20, soit 2,25.*

3 Fonctionnement superscalaire

Objectif : Comprendre le principe de l'architecture superscalaire.

Les composants d'un processeur à cinq étages mettent un temps donné ci-dessous à se stabiliser :

1. Récupération d'instruction : 0,5 ns ;
 2. Décodage d'instruction : 0,7 ns ;
 3. Exécution : 0,9 ns ;
 4. Écriture : 0,5 ns ;
 5. Sauvegarde dans les registres, calcul d'adresse suivante : 0,3 ns ;
1. Dans une architecture classique, quel est le temps de cycle minimal pour ce processeur ? *Le processeur doit avoir le temps d'exécuter chacune des étapes, il a donc besoin de 2,9 ns.*
 2. Combien de temps est nécessaire pour faire une instruction élémentaire (c'est une architecture RISC) ? *Dans les architectures RISC, un seul cycle est nécessaire pour faire une instruction. Donc 2,9 ns.*
 3. Pour faire une architecture à plusieurs étages, il faut insérer des registres qui mémorisent ce qui se passe entre les étages. Du coup, le temps de stabilisation est prolongé de 0,1 ns pour chaque étage. Pour faire une architecture pipeline, quel est le temps de cycle qu'il faut choisir ? *Un cycle d'une nanoseconde est le minimum (le temps d'exécution est de 0,9 ns+0,1 ns).*
 4. Dans ce cas, quel est le nombre de cycles par instructions dans le cas de l'architecture pipeline ? Quel est le nombre d'instruction par cycle ? Quel est le débit d'instructions ? *Cinq cycles sont nécessaires pour une instruction, mais en fait jusqu'à 5 instructions peuvent être en cours, et à chaque cycle une instruction se termine (CPI=1). Il y a donc un nombre $IPC=1/1=1$. Le débit d'instructions est de $10^9 \times IPC = 10^9 = 1000MIPS$.*
 5. On envisage une architecture superscalaire où toutes les unités sont dédoublées. Peut-on changer le temps de cycle ? Quel est le nombre d'instructions par cycle et le débit d'instruction ? *Les unités fonctionnent à la même vitesse, on ne change donc pas le temps de cycle. À chaque cycle deux instruction se terminent (IPC=2). Le débit d'instructions est de $10^9 \times IPC = 2 \times 10^9 = 2000MIPS$.*
 6. On envisage une architecture superscalaire où seule l'unité d'exécution est dédoublée. Peut-on changer le temps de cycle ? Quel est le nombre d'instructions par cycle et le débit d'instruction ? *On peut utiliser en alternance chaque unité d'exécution. Le temps limitant devient celui de l'unité de décodage. On peut donc avoir un temps de cycle de 0,8 ns. À chaque cycle une instruction se termine (IPC=1). Le débit d'instructions est de $\frac{1}{0,8} \times 10^9 \times IPC = 1,25 \times 10^9 = 1250MIPS$.*
 7. On envisage une architecture superscalaire où l'unité d'exécution et de décodage est dédoublée. Peut-on changer le temps de cycle ? Quel est le nombre d'instructions par cycle et le débit d'instruction ? *On peut utiliser en alternance chaque unité d'exécution. Le temps limitant devient celui de l'unité de décodage. On peut donc avoir un temps de cycle de 0,6 ns. À chaque cycle une instruction se termine (IPC=1). Le débit d'instructions est de $\frac{1}{0,6} \times 10^9 \times IPC = 1,667 \times 10^9 = 1667MIPS$.*