

GIF-17455 – Architecture des microprocesseurs

Examen Final

Mardi le 13 décembre 2005

Durée: 13h30-15h30

Vous devez montrer vos calculs ou justifier vos réponses. Une réponse, même bonne, mais sans autre information, ne vaut aucun point.

Question 1 (20 points)

Le pipeline d'un processeur à ordonnancement statique est

IF1 – IF2 – IF3 – ID – EX – M1 – M2 – M3 - WB

Dans ce pipeline :

- les instructions sont disponibles à la fin du cycle IF3 si ces instructions sont en cache.
- les données lues en mémoire sont disponibles à la fin du cycle M3 si ces données sont en cache.
- les données à écrire en mémoire doivent être disponibles au cycle M1
- Les opérations logiques et arithmétiques entières prennent 1 cycle en EX, incluant le calcul des adresses pour les lectures et écritures.
- Les additions en point flottant prennent 3 cycles en EX.
- Les multiplications en point flottant prennent 7 cycles en EX.
- Les unités d'exécution sont toutes réalisées en pipeline (donc 3 étages pour une addition point flottant, 8 étages pour une multiplication point flottant).
- L'envoi des données (data forwarding) est utilisé pour réduire les latences.

En faisant l'hypothèse que les accès aux caches sont réussis, donnez les latences de ce processeur pour les quatre cas suivants:

	<i>Instruction qui génère la donnée</i>	<i>Instruction qui utilise la donnée</i>
1.	L.D	ADD.D ou MUL.D
2.	MUL.D	ADD.D ou MUL.D
3.	ADD.D	S.D
4.	DADDIU (entier)	DADDIU (entier)

La latence est définie comme étant le nombre de cycles qui doivent s'écouler entre une instruction qui produit un résultat et une instruction qui utilise ce résultat.

Note : ADD.D, MUL.D : addition et multiplication en point flottant

Question 2 (35 points)

Un processeur avec pipeline statique a un seul niveau de cache. Il y a un cache pour les données et un cache pour les instructions. Les deux caches sont à correspondance directe, ont 8 KO chacun, les blocs ont 32 octets, et le temps d'accès réussi est de 1 cycle. Le processeur ne supporte pas la « lecture du mot critique d'abord ».

Le système a les caractéristiques suivantes:

- Mémoire : latence d'accès de 30 cycles (incluant temps de transfert de l'adresse de la cache vers la mémoire).
- Bus entre mémoire et cache : le bus a 8 octets de large, et il faut deux cycles pour transférer 8 octets.
- Cache des données
 - Cache à réécriture (write-back) et écriture allouée (write allocate).
 - En tout temps, 30% des blocs du cache de données sont modifiés (sales – dirty).
 - Il n'y a pas de tampon d'écriture – lorsqu'il faut écrire en mémoire, les données doivent donc se rendre en mémoire.
 - 6% des accès en cache sont des échecs
- Cache des instructions
 - 3% des accès en cache sont des échecs
- 20% des instructions sont des lectures de données, 15% des instructions sont des écritures de données.
- Chaque lecture ou écriture de données se fait sur 8 octets.
- Les instructions ont 4 octets.

a) Si le CPI du processeur avec caches parfaites (aucun échec lors des accès aux caches) est de 1.5, quel est le CPI du système décrit ci-haut ?

b) Si on augmente la fréquence du processeur, tout en gardant le reste du système identique (donc même puces de mémoire, même bus entre le processeur et la mémoire, etc...), quels sont les paramètres donnés ci-haut qui ne sont plus valides ? Comment changeraient ces paramètres ? Comment changerait le CPI ? Comment changerait le temps d'exécution ? Justifiez bien vos réponses.

Question 3 (25 points)

Soit la boucle suivante pour laquelle les suspensions de pipeline dues aux dépendances de données sont indiquées (R1 est initialisé à 0, F0 contient une constante) :

```
A :  L.D      F2, 800(R1)
      L.D      F4, 1600(R1)
      suspension
      MUL.D    F6, F2, F4
      suspension
      suspension
      suspension
      ADD.D    F6, F6, F0
      suspension
      S.D.     F6, 2400(R1)
      DADDIU   R1, R1, #8
      DSGTIU   R2, R1, #800      (R2 = 1 si R1 > 800)
      suspension
      BEQZ     R2, A
```

Déroulez la boucle le nombre minimum de fois pour que le code ordonnancé n'ait pas de suspension de pipeline. Donnez le code ordonnancé. Faites attention aux registres utilisés et aux adresses des lectures et écritures.

Question 4 (20 points)

a) Dans le protocole d'espionnage (snooping) décrit à la section 6.3 du volume et que nous avons vu en classe, on ne fait pas de différence entre un accès réussi en écriture et un échec en écriture lorsque le bloc est dans l'état « partagé ». Expliquez quel serait l'avantage de pouvoir différencier les deux cas, et comment il faudrait modifier le protocole pour différencier les deux cas.

b) Le même protocole de la figure 6.3 ne différencie pas un bloc non-modifié (propre/clean) qui est partagé ou qui n'est pas partagé. Dans les deux cas, ce bloc est représenté par l'état « partagé ». Lors d'un accès réussi en écriture, quel est l'avantage de différencier les deux cas en ajoutant un état « exclusif – non-modifié » ?

Question 5 – BONUS 10 points

On ajoute à un processeur à lancement statique le support matériel pour pouvoir exécuter des instructions prédiquées sur un registre (prédication). Pour pouvoir améliorer le compilateur, on doit déterminer quand l'utilisation de la prédication est préférable à celle des branchements.

Le pseudo-code suivant

```
if ( R1 == 0 ) {  
    Bloc de  $N_A$  instructions }  
else {  
    Bloc de  $N_B$  instructions }
```

peut donc être exécuté sans ou avec prédication :

Code sans prédication

```
BNEZ  R1, L1  
instruction 1 }  
...          } bloc de  $N_A$  instructions  
instruction  $N_A$  }  
JUMPL2  
L1: instruction 1 }  
...             } bloc de  $N_B$  instructions  
instruction  $N_B$  }  
L2:  /* suite du programme */
```

Code avec prédication

```
CMP.EQ pT, pF = R1, R0  
instruction 1 }  
...          } bloc de  $N_A$  instructions  
instruction  $N_A$  }  
instruction 1 }  
...          } bloc de  $N_B$  instructions  
instruction  $N_B$  }
```

Comment pourrait-on déterminer quand il est préférable d'utiliser le code avec prédication par rapport au code sans prédication ? N'oubliez pas de prendre en compte que les branchements causent des suspensions de pipeline.