Contrôle écrit IF4 – ARCH

Le 2 décembre 2010

Durée 2H

M. AKIL

Documents autorisés : polycopiés et notes du cours

I. Analyse des dépendances. Soit la boucle ci-dessous :

```
for     (i=1 ; i<=100 ; i=i+1 {
        y[i] = x[i]/c          /* Instruction I1
        x[i] = x[i] + c        /* Instruction I2
        z[i] = y[i] + c        /* Instruction I3
        y[i] = c - y[i]        /* Instruction I4
        }
```

I.1 Donner tous les types de dépendances (vrai dépendance, dépendance de sortie, anti-dépendance) entre les différentes instructions I1, I2, I3 et I4 de la boucle ci-dessus.

I.2 Donner la version de la boucle qui élimine les dépendances de sortie et les anti-dépendances. Expliquer la technique utilisée pour éliminer ces types de dépendance.

II. Processeur RISC DLX. Soit le programme DLX ci-dessous :

```
loop :  LD      F5,0(R1)    1
        LD      F6,0(R2)    2
        MULTD   F7,F5,F6    4
        ADDD    F4,F4,F7    2     =>
        ADDI    R1,R1,#8    1
        ADDI    R2,R2,#8    1
        SUBI    R3,R3,#1    1
        BNEZ    R3,loop     1
        nop                 1
```

On suppose que le nombre cycles du multiplieur flottant est de 4 cycles, celui de l'additionneur flottant est de 2 cycles. La durée des opérations sur des entiers est de 1 cycle.

La latence d'un chargement (load) est de 2 cycles. On a un branchement retardé d'un délai (branchement retardé d'un cycle) et il n'y pas de délai entre une opération sur les entiers et une instruction de branchement conditionnel qui en dépend.

Les registres R1, R2 contiennent respectivement : les adresses de 2 vecteurs v1 et v2, et le registre R3 contient le nombre d'éléments de chaque vecteur.

II.1 Sans réorganisation du code ci-dessus, donner le nombre de cycles par itération.

II.2 Donner une version optimisée de ce code avec ordonnancement, ainsi que le nombre de cycles par itération. Motivez votre réponse : expliquer la version proposée ainsi que les optimisations que vous proposez.

II.3 Proposer une version déroulée une fois de cette boucle. Donner le temps d'exécution de la boucle déroulée en nombre de cycles.

II.4 On suppose les latences de pipeline ci-dessous :

Instruction produisant le résultat	Instruction utilisant le résultat	latence
Op UAL flottante	Une autre Op UAL flottante	3
Op UAL flottante	Rangement double	2
Chargement double	Op UAL flottante	1
Chargement double	Rangement double	0

Dérouler la boucle suivante autant de fois que nécessaire pour l'ordonnancer sans aucune suppression et en diminuant les instructions de gestion de boucle. On suppose un branchement retardé d'un cycle. Donner le temps d'exécution de la boucle déroulée et ordonnancée. Motivez et justifiez votre réponse. Quel est le calcul effectué par cette boucle.

```
Boucle:   LD      F0,0(R1)
          MULTD   F0,F0,F2
          LD      F4,0(R2)
          ADDD    F0,F0,F4
          SD      0(R2),F0
          SUBI    R1,R1,#8
          SUBI    R2,R2,#8
          BNEZ    Boucle
```

III. Mémoire cache

II.1 On suppose que l'on dispose d'une mémoire principale de 64 méga octets et d'une mémoire cache de 128 kilo octets et la taille de chaque bloc est de 16 octets.

Donner :

- les différents champs de l'adresse de la mémoire principale
- la signification et la taille en bits de chacun de ces champs

Pour chacun des cas suivants :

1. mémoire cache à correspondance directe
2. mémoire cache associative par ensembles de 8 éléments.

II.2. soit la section de code ci-dessous :

```
for (j=0 ; j<100 ; j = j+1)

  for (i =0 ; i<5000 ; i = i+1)

      x[i][j] = a*x[i][j] ;
```

Proposer une version de code de cette section permettant de réduire les échecs dans le cache.

Quel est le type de localité que vous avez amélioré ? Expliquez votre réponse ainsi que le code proposé.