

ARCHITECTURE DES ORDINATEURS
Examen Décembre 2008
3H - Tous documents autorisés
Les questions sont indépendantes

On utilise le jeu d'instructions NIOS II.

PROGRAMMATION ASSEMBLEUR

Q 1) Que fait le programme assembleur suivant ?

```
LDW R9,0(R5)
LDW R10, 0(R4)
STW R10, 0(R5)
STW R9, 0(R4)
```

Q 2) Que fait le programme ci-dessous, qui travaille sur un tableau T de 100 int, avec T[0] à l'adresse A000 0000H ? Donner le programme C correspondant.

On rappelle que dans le jeu d'instructions NIOS, MOV est une pseudo-instruction implantée à l'aide de l'instruction ADD. MOV R2,R1 = ADD R2,R1,R0

```
ORHI R2, A000H // Adresse T[0]
ADDI R3,R2, 40010 // Adresse T[99]+4
MOV R4,R2
Bouclei MOV R5,R4
ADDI R6,R4,4
Bouclek: LDW R7,0(R6)
LDW R8,0(R5)
BGE R7,R8, Suite
MOV R5,R6
Suite : ADDI R6,R6,4
BLT R6, R3, Bouclek
MOV R9,R8
LDW R10, 0(R4)
STW R10, 0(R5)
STW R9, 0(R4)
ADDI R4,R4,4
BLT R4,R3,Bouclei
.
```

PIPELINE

Soit un processeur qui a les pipelines suivants

Entiers
LI DI LR EX AC LD ER
Flottants

LI DI LRF EX1 EX2 EX3 EX4 RT ER

avec la signification suivante :

LI : lecture des instructions dans le cache instructions

DI : décodage des instructions

LR : lecture registres entiers

LRF : lecture des registres flottants

EX : exécution UAL pour les entiers, et calcul des adresses (mémoire et branchements)

EXi : phase d'une exécution flottante

AC : accès au cache. La donnée à écrire doit être disponible pour les instructions « store ».

LD : fin de l'accès cache (succès ou échec). Si succès, on dispose à la fin de LD de la donnée lue pour les instructions « load »

RT : Résolution des exceptions

ER : Écriture registres.

- Tous les "bypass" nécessaires existent.

Q 3) Donner les latences des instructions source (colonne de gauche) lorsqu'elles sont suivies par les instructions destination (colonne de droite) pour un processeur scalaire (une instruction par cycle)

NB : une latence de n cycles signifie que deux instructions peuvent se suivre aux cycles i et i+n.

Les instructions ci-dessous sont des instructions NIOS II avec l'ajout de la figure 1.

	Instruction source	Instruction consommateur	Latence
a	ADD R1,R2,R3	UALR4,R1,R2	
b	SUB R1,R2,R3	ST R1,0 (R5)	
c	AND R1,R2,R3	LD R6, 4(R1)	
d	LDW R1, 0(R7)	XOR R2, R1, R3	
e	OR R1, R2,R3	BGT R1,R9, déplacement	
f	FADD F1,F2,F3	FSUB F4,F1,F5	
g	LF F1,0(R2)	FMUL F5,F1,F2	
h	FSUB F1,F2,F0	SF F1,0 (R5)	

CACHES.

On suppose que le processeur utilisé a un cache données de 64 Ko, avec des blocs de 64 octets. Il utilise l'écriture simultanée (write through) non allouée. On rappelle qu'avec l'écriture non allouée, il n'y a pas de défauts de cache en écriture.

Soit le programme suivant P1, où abs calcule la valeur absolue de son argument.

```
#define N 256
int X[N], Y[N], S ;
for (i=0 ; i<N ; i++)
S+= abs(X[i]-Y[i]);
```

On suppose que les tableaux X[N], Y[N] sont rangés consécutivement à partir de l'adresse hexadécimale 1000 0000_H.

Q 4)

- a) Quelles sont les adresses de X[0] et Y[0], en notation hexadécimale) ?
- b) Quel est pour ce cache le nombre de bits pour l'adresse dans le bloc, le nombre de bits d'index et le nombre de bits d'étiquette dans les deux cas suivants : 1) correspondance directe, 2) associativité quatre voies (quatre blocs par ensemble).
- c) Quel est le nombre total de défauts de cache lors de l'exécution du programme P1 pour les deux cas suivants : a) correspondance directe, b) associativité deux voies (deux blocs par ensemble) ?

Q 5) Pour quelle valeur minimale de N étant une puissance de 2 aura-t-on deux défauts de cache par itération avec la correspondance directe ?

EXECUTION DE BOUCLES

On ajoute au jeu d'instructions NIOS II des instructions flottantes simple précision (32 bits) (Figure 2) et 32 registres flottants F0 à F31 (F0 est un registre normal).

Les additions, soustractions et multiplications flottantes sont pipelinées. Une nouvelle instruction peut démarrer à chaque cycle. Les latences sont de 2 cycles pour LF et de 4 cycles pour les instructions flottantes.

Les branchements ne sont pas retardés.

LF	2	LF ft, déplac(rs)	$ft \leftarrow MEM [rs + SIMM]$
SF	1	SF ft, déplac(rs)	$ft \rightarrow MEM [rs + SIMM]$
FADD	4	FADD fd, fs,ft	$fd \leftarrow fs + ft$ (addition flottante simple précision)
FMUL	4	FMUL fd, fs,ft	$fd \leftarrow fs * ft$ (multiplication flottante simple précision)
FSUB	4	FSUB fd, fs,ft	$fd \leftarrow fs - ft$ (soustraction flottante simple précision)
F2I	1	F2I rd, ft	$rd \leftarrow ft$ (copie d'un registre flottant dans un registre entier)

Figure 1 : Instructions flottantes ajoutées (Ce ne sont pas les instructions NIOS)

Soit le programme assembleur P1, qui travaille sur des tableaux de flottants simple précision (float) X[N] et Y[N] rangés successivement en mémoire, avec N = 100. L'adresse de X[0] est initialement contenue dans le registre R1. L'adresse de S est 100H.

```

FSUB F0,F0,F0
FSUB F3,F3,F3
ADDI R2, R0,40010
Boucle :LF F1, 0(R1)
        LF F2, 1024(R1)
        FSUB F1, F1,F2
        F2I R3,F1
        BGE R3,R0, suite
        FSUB F1,F0,F1
Suite : FADD F3,F3,F1
        ADDI R1,R1,4
        BLT R1,R2, Boucle
        SF F3, 100H(R0)

```

Q 6) Expliquer pourquoi on peut comparer le signe du résultat de la différence de deux nombres flottants en testant le bit 31 du résultat à l'aide des instructions F2I et BGE.

Q 7) Donner le code C correspondant au programme P1

Q 8) En montrant l'exécution cycle par cycle du programme assembleur P1, donner le nombre de cycles par itération de la boucle du programme assembleur P1 dans les deux cas (BGE pris et BGE non pris) en supposant qu'il n'y a aucune pénalité de branchement..

NB : cette question est faisable même sans avoir répondu aux questions Q6 et Q7

Q 9) Quel est le nombre de cycles par itération de la boucle après optimisation sans déroulage, en supposant qu'il n'y a pas de pénalité de branchement ?

Pour la question suivante, on suppose que l'on dispose d'une prédiction de branchement statique : les branchements avant sont prédits « non pris » et les branchements arrière sont prédits « pris ». Un branchement mal prédit a une pénalité de 3 cycles.

Q 10) En tenant compte des pénalités de branchement et des comportements possibles de BGE et BLT, donnez

- a) la valeur minimale du temps d'exécution du programme optimisé
- b) la valeur maximale de temps d'exécution du programme optimisé

Q 11) Peut on définir une instruction FABS (valeur absolue) sur un registre flottant 32 bits ? Que doit faire cette instruction ?

On suppose maintenant qu'il existe l'instruction FABS avec une latence de 1

Q 12) Quel est maintenant le nombre de cycles par exécution de la boucle optimisée ? Quel est le facteur de déroulage maximal que l'on peut utiliser et que devient le nombre de cycles par itération ?