

ARCHITECTURE DES ORDINATEURS
Examen Juin 2010
3H – Tous documents autorisés
Les questions sont indépendantes

On utilise le jeu d'instructions NIOS II.

PROGRAMMATION ASSEMBLEUR

Q 1) Ecrire la suite des instructions qui exécute le programme C ci-dessous. On supposera que les variables g , h , k sont aux adresses hexadécimales 20, 24 et 28.

```
int g, h, k
if (g>h) k=g;
    else k=h;
```

Q 2) Ecrire la suite d'instructions qui calcule $y = 2^x$ (2 puissance x). La valeur x , inférieure ou égale à 30, est initialement dans le registre R2. La résultat sera dans le registre R1. On souhaite obtenir la version la plus rapide.

Q 3) Ecrire en assembleur NIOS un programme calculant le nombre d'entiers 32 bits négatifs compris dans les cases d'adresses F000 0000 à F000 FFFC. Le résultat sera rangé dans R3.

Q 4) Proposer une version de la question Q3 pour laquelle on n'utilise qu'un seul branchement par itération : le branchement de boucle.

CACHES.

On suppose que le processeur utilisé a un cache données de 1 Ko à correspondance directe, avec des blocs de 32 octets.

Soit le code C

```
int SIZE, STRIDE;
int A[SIZE];
// Initialisation de SIZE et STRIDE
for (j=0; j<N; j++)
    for (i=0; i<SIZE; i=i+STRIDE)
        X+=A[i];
```

On suppose que N est très grand. On suppose également que toutes les valeurs exceptées celles du tableau A sont dans des registres.

Q 5) Quel est le taux d'échec (nombre d'échecs/nombre d'accès) lorsque $STRIDE = 2$ et $SIZE = 256$? Donner successivement

- le nombre d'accès pour la boucle interne
- le nombre d'échecs cache pour le premier accès à la boucle interne
- le nombre d'échecs cache pour les accès suivant à la boucle interne

- le taux d'échecs en fonction de N
- le taux d'échecs lorsque N tend vers l'infini.

Q 6) Donner le taux d'échec pour les différentes configurations du tableau ci-dessous (pour N très grand)

	STRIDE = 2	STRIDE = 4	STRIDE = 8
SIZE = 256			
SIZE = 512			

EXECUTION DE BOUCLES

On ajoute au jeu d'instructions NIOS II des instructions flottantes simple précision (32 bits) (Figure 1) 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 3 cycles pour les instructions flottantes.

Les branchements ne sont pas retardés.

LF	2	LF ft, déplac(rs)	ft ← MEM [rs +SIMM]
SF	1	SF ft, déplac(rs)	ft → MEM [rs + SIMM]
FADD	3	FADD fd, fs,ft	fd ← fs + ft (addition flottante simple précision)
FMUL	3	FMUL fd, fs,ft	fd ← fs * ft (multiplication flottante simple précision)
FSUB	3	FSUB fd, fs,ft	fd ← fs - ft (soustraction flottante simple précision)

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

Soit le programme assembleur ci-dessous qui travaille sur des tableaux de flottants simple précision (float) X[N] et Y[N] rangés successivement en mémoire, avec N = 256. L'adresse de X[0] est initialement contenue dans le registre R1. F0 contient une constante flottante A.

```
ADDI R2, R1,51210
```

```
Boucle :LF F1, 0(R1)
        LF F2, 1024(R1)
        FMUL F1,F1,F0
        FADD F2,F2,F1
        ADDI R1,R1,4
        SF F2, 1020 (R1)
        BLT R1,R2, Boucle
```

Q 7) Donner le code C correspondant au programme assembleur.

Q 8) En montrant l'exécution cycle par cycle du programme assembleur P3, donner le nombre de cycles par itération de la boucle du programme assembleur en supposant qu'il n'y a aucune pénalité de branchement. Quel est le temps d'exécution total en cycles d'horloge ?

Q 9) Quel est le nombre de cycles par itération de la boucle avec déroulage d'ordre 2, en supposant qu'il n'y a pas de pénalité de branchement ?

Q 10) En déroulant au maximum la boucle, quel nombre de cycles par itération obtient-on ?