

ARCHITECTURE DES ORDINATEURS
Corrigé Examen Juin 2011
3H - Tous documents autorisés
Les questions sont indépendantes

On utilise le jeu d'instructions NIOS II.

PROGRAMMATION ASSEMBLEUR

1) Soit le code C

```
int a, b;
unsigned char c;
char d; // Les adresses de a, b, c, d sont initialement dans R1, R2,
        //R3, R4

a= (int) c;
b= (int) d
```

Question 1) Ecrire le code assembleur NIOS correspondant au code C

```
LBU R5, 0(R3)
STW R5, 0(R1)
LB R5, 0(R4)
STW R5, 0(R2)
```

2) Soit une fonction Prime(x dans R4) qui renvoie 1 dans R2 si x est premier et 0 sinon
R5 contient l'adresse du premier élément d'un tableau d'entiers T, et R6 contient le nombre N
d'éléments du tableau.

Soit le code assembleur NIOS II ci-dessous

```
        ADDI R4,R0,0
Test :BLE R6,R0, FIN
Boucle : ADDI R4,R4,1
        Call Prime
        BEQ R2,R0, Boucle
        STW R4, (R5)
        ADDI R5,R5,4
        ADDI R6,R6,-1
        BEQ R0,R0,Test
FIN :
```

Question 2) Donner le code C correspondant au code assembleur

```
X=0 ; i=0;
While (N>0)
{
Do {
```

```
    X++;  
While (Prime (X)==0 ;  
T[i] = X ;  
i++;  
N-- ;  
}
```

Question 3) Que fait le programme ?

Etablit le tableau des N premiers « nombres premiers »

CACHES

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

Le cache utilise la réécriture avec écriture allouée (il y a des défauts de cache en écriture)

Le processeur a des adresses sur 32 bits.

L'adresse de X[0] est F000 0000_H

L'adresse de Y[0] est FF00 0000_H

Question 4) 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 : a) correspondance directe, b) associativité quatre voies (quatre blocs par ensemble).

Correspondance directe

- Adresse dans bloc : 5 bits
- Index : 10 bits
- Etiquette : 17 bits

Associativité 4 voies

- Adresse dans le bloc : 5 bits
- Index : 8 bits
- Etiquette : 15 bits

Question 5) Dans quels blocs vont les flottants X[0] et Y[0] correspondance directe ?

X[0] et Y[0] vont dans le bloc 0 du cache en correspondance directe

Question 6) Quel est le nombre total de défauts de caches lors de l'exécution des boucles pour les deux cas suivants : a) correspondance directe, b) associativité quatre voies pour les corps de boucle suivant

```
1) for (i=0 ; i<2048 ; i++)  
    : S+=X[i] ;
```

Un bloc contient 8 floats. Il y a un défaut de cache toutes les 8 itérations, pour CD et 4 voies

soit un total de $2048/8 = 256$

CD = 256

4 voies = 256

```
2) for (i=0 ; i<2048 ; i++)  
    : S+=X[i] + Y[i];
```

En CD, il y a 2 défauts de cache par itération, soit un total de 4096

En 4 voies, il y a 2 défauts de cache toutes les 8 itérations, soit un total de 512

Float X[2048], Y[2048];

```
3) for (i=0 ; i<2040 ; i++)
    : S+=X[i] + X[i+8];
```

Il n'y a jamais de conflits

Itérations 0 à 7 : 2 défauts

Itérations 8 à 2040 : 1 défaut toutes les 8 itérations soit 254 défauts.

En CD et 4 voies, il y a donc un total de 256 défauts.

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)

Question 7) Quel est le temps d'exécution (en cycles par itération) de la boucle de la table 1. Optimiser la boucle et donner le nouveau temps d'exécution.

Table 1 : Programme C et programme assembleur

float X[100], Y[100], S; int i; for (i=0; i<100; i++) S+=X[i]*Y[i];	ADDI R5, R3, 400 FSUB F0, F0, F0 Boucle :LF F1,(R3) LF F2, (R4) FMUL F1,F1,F2 FADD F0,F0,F1 ADDI R3,R3,4 ADDI R4,R4,4 BNEQ R3,R5, Boucle SF F0, (adresse de S)
--	---

Boucle 1	:LF F1,(R3)
2	LF F2, (R4)
3	
4	FMUL F1,F1,F2
5	
6	

7	FADD F0,F0,F1
8	ADDI R3,R3,4
9	ADDI R4,R4,4
10	BNEQ R3,R5, Boucle

10 cycles

Version optimisée

Boucle 1	LF F1,(R3)
2	LF F2, (R4)
3	
4	FMUL F1,F1,F2
5	ADDI R3,R3,4
6	ADDI R4,R4,4
7	FADD F0,F0,F1
8	BNEQ R3,R5, Boucle

8 cycles

Question 8) Quel serait le temps d'exécution (en nombre de cycles de la boucle initiale) avec un déroulage de boucle d'ordre 4 ?

Version déroulée

Boucle 1	LF F1,(R3)
2	LF F2, (R4)
3	LF F3,4(R3)
4	LF F4,4 (R4)
5	LF F5,8(R3)
6	LF F6,8 (R4)
7	LF F7,12(R3)
8	LF F8,12 (R4)
9	FMUL F1,F1,F2
10	FMUL F3,F3,F4
11	FMUL F5,F5,F6
12	FMUL F7,F7,F8
13	FADD F0,F0,F1
14	FADD F10,F10,F3
15	FADD F11,F11,F5
16	FADD F1,F12,F7
17	ADDI R3,R3,16
18	ADDI R4,R4,16
19	BNEQ R3,R5, Boucle

19 cycles/4 = 4,75 cycles/itération

PREDICTION DE BRANCHEMENT

Un branchement a le comportement suivant, où "P signifie pris et N signifie non pris.

PPNPPPPNPPPPNPPPPNPPPPNPPPPNPPPPN....

Question 9 : après la phase d'initialisation, quelle est le pourcentage de bonnes prédictions

- a) avec un prédicteur 1 bit
- b) avec un prédicteur 2 bits.

Prédicteur 1 bit

N PPPPN

X NPPPP

3 bonnes prédictions sur 5 soit 60%

Prédicteur 2 bits

N PPPPN

P pPPPP

4 bonnes prédictions sur 5 soit 80%