

# ARCHITECTURE DES ORDINATEURS

PARTIEL Octobre 2009

Tous documents autorisés – 2H

**Pour toutes les questions, on utilise le jeu d'instructions NIOS-II. Les différentes parties sont indépendantes**

## **PARTIE 1 : Implantation mémoire**

Soit la déclaration de variables suivante pour un programme C

```
char toto[11];
```

```
int X,Y,Z;
```

```
short A,B,C;
```

```
double D[10],E;
```

```
float F,G,H;
```

**Q 1) Donner le déplacement à partir de l'adresse toto[0] de X, A, D[0], et H. En supposant que toto[0] soit à l'adresse F000 0000<sub>H</sub>, donner les adresses (hexadécimales) de X, A, D[0], et H. Quelle est l'occupation mémoire en octets ?**

**Q 2) Proposer une déclaration qui minimise l'occupation mémoire et donnez le nombre d'octets correspondant à cette nouvelle déclaration.**

## **PARTIE 2 : Exécution d'instructions**

Les registres du processeur NIOS contiennent les valeurs suivantes en hexadécimal.

R0	0000 0000
R1	89AB 1234
R2	5432 9876
R3	FFFF FFFF
R4	ABEF CD01

**Q 3) Donner le contenu des registres R5 à R10 (sous forme de huit chiffres hexadécimaux) après exécution des instructions suivantes.**

- a) ADD R5, R2, R1
- b) SUB R6, R1, R2
- c) SLLI R7,R1, 4
- d) SRAI R8, R1,8
- e) SRLI R9, R1, 1
- f) MUL R10, R1,R3

**Q 4) Donner le contenu des registres R11 à R15 (sous forme de huit chiffres hexadécimaux) après exécution des instructions suivantes.**

- a) OR R11, R1, R2
- b) AND R12, R1, R2
- c) XOR R13, R1, R2
- d) CMPGE R14, R0,R1
- e) CMPLT R15,R3,R4

**Q 5) Donner le contenu (sous forme de huit chiffres hexadécimaux) du registre CP après exécution des instructions suivantes en supposant à chaque fois que l'adresse de l'instruction est : 1000 0000<sub>H</sub>.**

- a) BEQ R0,R0, +8
- b) BNE R0,R0, +4

**PARTIE 3 : Désassemblage**

Soit le programme assembleur P1 qui travaille sur un tableau d'entiers T[N] dont l'adresse de début est contenue dans le registre R1 et range le résultat dans la case mémoire d'adresse 00007000<sub>H</sub>

	ADDI R2,R0, N // N <2 <sup>15</sup>
	SLLI R2,R2,2
	ADD R2,R1,R2
	ADD R3,R0,R0
Boucle	LDW R4, 0(R1)
	ADDI R1,R1,4
	BNE R4,R0, Suite
	ADDI R3,R3,1
Suite	BNE R1, R2, Boucle
	STW R3, 7000 <sub>H</sub> (R0)

**Q 6) Que fait le programme P1 ?**

**Q 7) Donner le programme C correspondant au programme P1**

**PARTIE 4 : Programmation assembleur**

**Q 8) Ecrire un programme en assembleur NIOS qui a dans R4 (partie basse) et R5 (partie haute) un entier non signé sur 64 bits d'une part, dans R6 (partie basse) et R7 (partie haute) un second entier non signé de 64 bits et renvoie le plus grand de ces deux entiers non signés dans R2 (partie basse) et R3 (partie haute).**

**Q 9) Indiquer les instructions qu'il faut ajouter au programme de la question Q8 pour le transformer en une procédure MAXU64**

**Q 10) Ecrire une procédure NIOSII MAXS64 qui reçoit dans R4 (partie basse) et R5 (partie haute) un entier signé sur 64 bits d'une part, dans R6 (partie basse) et R7 (partie haute) un second entier signé de 64 bits et renvoie le plus grand de ces deux entiers signés dans R2 (partie basse) et R3 (partie haute)**