

Partiel Architecture des Ordinateurs - 24 Octobre 2016

Tous documents autorisés. Calculatrices interdites.
Durée 2h

Les réponses seront fournies sur la feuille jointe. L'espace prévu pour les justifications est suffisant, mais vous pouvez utiliser les feuilles d'examen si vous le souhaitez vraiment. La notation demandée pour les valeurs numériques est impérative, toute autre réponse ne sera pas corrigée.

1. Représentation des entiers [3 pts]

Q1. Coder 74 et -127 en complément à 2 sur 8 bits.

Q2. On note # l'opération effectuée par un additionneur sur 8 bits. Pour les opérations suivantes, donner le résultat en notation **hexadécimale**, donner la retenue, et indiquer si le résultat est égal à celui de l'opération arithmétique d'addition lorsque les opérandes sont interprétés en naturels et en relatifs (complément à deux).

0x84 # 0x81 ; 0x2A # 0x39 ; 0x34 # 0x75 ; 0x84 # 0xFF

2. Jeu d'instructions MIPS [6 pts]

Pour Q3, Q5, Q6, Q7, les résultats seront donnés en notation **hexadécimale**.

Q3. Donner le codage des instructions :

a) ADDU R4, R5, R6 b) BLEZ R1, 128

Q4. Expliquer **en une phrase** à quoi sert le codop *SPECIAL* δ .

Q5. L'état initial des registres est : R2 = 0x9876A123. Donner le contenu du registre R1 après l'exécution des instructions :

a) ADDIU R1, R2, 0xF0C0 b) ANDI R1, R2, 0xF0C0
c) SRA R1, R2, 16 d) SRL R1, R2, 20

Remarque : ces instructions ne modifient pas R2.

Q6. L'état initial des registres est : R2 = 0x9876A123 et R3 = 0x0000F001. Donner le contenu des registres R1 et R3 après l'exécution de la séquence de code

```
SLT R1, R2, R3
MOVZ R3, R2, R1
```

Q7. L'état initial des registres est R1 = 0x10000008 et R2 = 0x9876A123. La mémoire est organisée en Big Endian. Initialement tous les octets de la mémoire sont à 0. Donner son contenu après exécution du code :

```
SW R2, -4(R1)
SB R2, -7(R1)
SH R2, -6(R1)
```

3. Programmation [9 pts]

On utilise le jeu d'instruction MIPS. On rappelle que le registre R0 est câblé à 0.

Q8. On considère le programme Prog1.

a) Donner la valeur (notation **décimale**) de R2 après l'exécution de Prog1, si la valeur initiale de R1 est 5

- b) Même question si la valeur initiale de R1 est 22. NB : 22 en notation décimale
- c) On veut que la valeur de R2 calculée par Prog1 soit représentable en naturels sur 8 bits. Quelle est la plus grande valeur initiale (notation **décimale**) possible pour R1 ?

Q9. Le tableau d'entiers X est implanté à partir de l'adresse 0x10000000. Il contient {7,2,3,4,6,0,1,8}. L'état initial est : R10 = 0x10000000. Pour cette question et la suivante, **les entiers sont codés sur 32 bits**.

- a) Donner le contenu (notation **décimale**) du tableau X après l'exécution de la première itération de Prog2.
- b) Donner le contenu (notation **décimale**) du tableau X après l'exécution de Prog2.

Q10. On considère les mêmes initialisations que dans Q9.

- a) Donner le contenu (notation **décimale**) du registre R10 après l'exécution de Prog3.
- b) Que calcule Prog3 ?

<i>Prog1</i>	<i>Prog2</i>	<i>Prog3</i>
SLL R2, R1, 2	ADDI R12, R10, 16	ADDI R12, R10, 32
ADDU R2, R2, R1	bcl : LW R1, 0(R10)	Deb : LW R1, 0(R10)
SLL R2, R2, 1	LW R2, 4(R10)	BEQZ R1, Suite
ADDIU R2, R2, 40	ADD R2, R2, R1	ADDI R10, R10, 4
	SW R2, 4(R10)	BNE R10, R12, Deb
	ADDI R10, R10, 8	Suite : ADD R10, R10, 32
	BNE R10, R12, bcl	SUB R10, R10, R12
		SRL R10, R10, 2

4. Mémoire [2 pts]

Q11. On considère la déclaration C suivante :

```
char c[3];
short x ;
int y[2] ;
double u[10] ;
```

Le placement est aligné et en big endian. Les variables sont allouées **dans l'ordre** à partir de l'adresse $b=0x00001000$. Donner les déplacements par rapport à b (notation **décimale**) de $c[2]$, de x et de $u[0]$.

Rappels :

- 1) le déplacement est la différence entre l'adresse de la variable et b ; mais il est inutile de calculer explicitement ces adresses : tout se passe comme si $b=0$.
- 2) Les tableaux C sont indexés à partir de 0, donc par exemple le dernier élément du tableau c est $c[2]$.