Partiel Architecture des Ordinateurs - 23 Octobre 2017

Tous documents autorisés. Calculatrices interdites.

Durée 2h

Les réponses seront fournies sur la **feuille jointe**. 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. Les questions sont indépendantes.

1. Représentation des entiers [3 pts]

Q1. Coder 45 et -192 en complément à 2 sur 8 bits (notation **hexadécimale**).

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).

0xA2 # 0x74 ; 0xA2 # 0xA2; 0xA2 # 0x24 ; 0x24 # 0x74

2. Jeu d'instructions MIPS [6 pts]

Pour Q3, Q4, Q5 et Q6, les résultats seront donnés en notation **hexadécimale**.

Q3. Donner le codage des instructions :

a) AND R6, R5, R4 b) SW R1, 8(R2)

Q4. L'état initial des registres est : R2 = 0xABCDEF12. Donner le contenu de R1 après l'exécution des instructions :

a) ADDI R1, R2, 4 b) ADDI R1, R2, 0xFFFF c) ORI R1, R2, 0xF0C0
c) SRA R1, R2, 16 d) SRL R1, R2, 8 e) SLL R1, R2, 4

Remarque : ces instructions ne modifient pas R2.

Q5. L'état initial des registres est R2 = 0xABCDEF12 et R1 = 0x00001000. La mémoire est organisée en Big Endian, avec le contenu de la table 1. Donner le contenu de R2 après l'exécution des instructions :

a) LW R2, 0(R1) b) LB R2, 3(R1) c) LH R2, 2(R1)

adresse	0x00001000	0x00001001	0x00001002	0x00001003
valeur	0x11	0x22	0x33	0xAA

Table 1 : plan mémoire

Q6. L'état initial des registres est : R2 = 0xABCDEF12 et R3 = 0x0000F001. Donner le contenu des registres R1 et R3 après l'exécution de la séquence de code

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

Q7. Combien d'instructions le format Registre-Immédiat permet-il de coder au maximum ?

3. Programmation [9 pts]

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

Q8. Donner la valeur (notation **décimale**) de R3 après l'exécution de *Prog1*, si la valeur initiale de R2 est 0x90000000

Q9. Le tableau d'entiers X est implanté à partir de l'adresse 0x10000000. Il contient {7,2,3,4,6,0,5 }. L'état initial est : R10 = 0x10000000. Pour cette question les entiers sont codés sur 32 bits.

- Donner la valeur (notation **décimale**) du registre R6 et l'état (notation **décimale**) du tableau X après l'exécution de la première itération de *Prog2*.
- Donner le contenu (notation **décimale**) du tableau X après l'exécution de *Prog2*.

<i>Prog1</i>	<i>Prog2</i>
ADDI R3, R0, 0 ADDI R4, R0, 31 Bcl: BGEZ R2, Suit ADDI R3, R3, 1 Suit: SLL R2, R2, 1 ADDI R4, R4, -1 BNEZ R4, Bcl	ADDI R11, R10, 24 Bcl : LW R4, 0(R10) LW R5, 0(R11) ADD R6, R4, R5 SW R6, 0(R10) ADDI R10, R10, 4 ADDI R11, R11, -4 BNE R10, R11, Bcl

Q10. Dans *Prog2*, on veut remplacer l'instruction ADD R6, R4, R5 par l'appel de la fonction foo :

```
int foo( int x, int y) {
    return (x+y)
}
```

- Donner le code de la fonction foo, sans sauvegarde de registres ni allocation de pile.
- Compléter *Prog2bis* pour qu'il soit équivalent à *Prog2*.

<i>Prog2bis</i>
ADDI R11, R10, 24 Bcl : LW R4, 0(R10) LW R5, 0(R11) <Inst 1> <Inst 2> ADDI R10, R10, 4 ADDI R11, R11, -4 BNE R10, R11, Bcl

NB On utilisera exclusivement la syntaxe des cours et des TD. En particulier, les registres sont notés R0-R31. Cet exercice est indépendant de la Q9.

4. Mémoire [2 pts]

Q11. On considère la déclaration C suivante :

```
char c[5];
int y[2];
char d;
```

```
double u[10];
```

Le placement est aligné et en big endian. Les variables sont allouées **dans l'ordre** à partir de l'adresse 0x00001000. Donner l'adresse (notation **hexadécimale**) de c[1], de d et de u[0].

Rappel : les tableaux C sont indexés à partir de 0, donc par exemple le dernier élément du tableau c est c[4].