

TD3 – Architecture logicielle : Instructions arithmétiques et logiques

On considère le jeu d'instructions MIPS32.

1. Etude des instructions

Initialement, le registre r1 contient 0x95842103 et le registre r2 contient 0x00001000

En repartant à chaque fois de l'état initial, donner l'état des registres modifiés après exécution des instructions

- a) ADD r3,r1,r2
- b) ADDI r3, r1, -1
- c) SLL r2,r1,4
- d) SRL r2,r1,8
- e) SRA r2,r1,12
- f) LUI r2, 0xABCD
- g) ORI r2,r1,0xABCD
- h) SLT r3, r1, r2
- i) SLTU r3, r1, r2

Quelle est la différence entre ADD et ADDU ?

2. Format des instructions

Donner le codage hexadécimal des instructions

- a) ADD r3,r1,r2
- b) ADDI r2,r1, -1
- c) SLL r2,r1,4
- d) SRL r2,r1,12
- e) SRA r2,r1,8
- f) LUI r2, 0xABCD
- g) LW r1, 4(R3)

Quel est l'intervalle des entiers représentables par le champ immédiat en format R-I?

Pourquoi n'il y a-t-il pas d'instruction SUBI ?

3. Compilation des constantes

- a. Quel est le résultat des deux séquences d'instructions suivantes :

Séquence 1

LUI r1, 0x9876
ADDI r1,r1, 0x8432

Séquence 2

LUI r1, 0x9876
ORI r1,r1, 0x8432

- b. Ecrire une séquence d'instructions qui positionne le registre r2 à 0x00009876.
- c. Comment peut-on implémenter la pseudo instruction LI r2, 0x12348765 ?

4. Lecture de programme

Que fait le programme suivant (on imagine que le registre r1 est déjà initialisé) ?

```
LI r2, 0x80000000
AND r2, r2, r1
SRL r2, r2, 31
ADDU r1, r1, r2
```

5. Programmation

Exercice 1

- a. Ecrire un programme qui multiplie par 65 le contenu du registre r1 interprété en naturels, avec résultat dans r2, sans considérer le problème du dépassement.
- b. Ecrire un programme qui multiplie par 15 le contenu du registre r1 interprété en naturels, avec résultat dans r2, sans considérer le problème de la correction du résultat.

Exercice 2

- a. Ecrire un code qui effectue l'addition de deux entiers naturels sur 64 bits contenus dans (R1, R2) et (R3, R4) avec résultat dans (R5, R6). Les registres impairs contiennent le mot de poids fort.

Exercice 3

- a. Initialement, R10 contient 0x12345678. Que contient R3 à la fin de l'exécution du programme suivant ?

```
ROL $R1,$R10,8
LI R2,0X00FF00FF
AND R3,R1,R2
ROR R1,R10,8
NOT R2,R2
AND R1,R1,R2
OR R3,R3,R1
```
- b. A quoi peut servir ce programme ?