

Bài tập chương 3:

3.1. Cho đoạn chương trình hợp ngữ trên MIPS như sau:

```
lui    $s0, 0x4050
ori    $s0, $s0, 251
sll    $s1, $s0, 4
add    $s2, $s0, $s1
andi   $s3, $s1, 0x0FE7
```

Hãy xác định giá trị nội dung của các thanh ghi \$s0, \$s1, \$s2, \$s3 theo dạng Hexa sau khi thực hiện đoạn chương trình trên.

3.2. Chuyển các câu lệnh C sau đây về dạng hợp ngữ MIPS. Giả thiết rằng các biến f, g, h, i, j được gán tương ứng trong các thanh ghi \$s0, \$s1, \$s2, \$s2, \$s3, \$s4; A và B là hai mảng dữ liệu các phần tử số nguyên 32-bit có địa chỉ cơ sở tương ứng nằm trong các thanh ghi \$s6 và \$s7.

- a. $f = -g - A[4]$
- b. $B[8] = A[i-j]$
- c. $f = g - A[B[4]]$

3.3. Hãy viết lời giải thích cho các lệnh của các đoạn chương trình hợp ngữ MIPS dưới đây, rồi chuyển đoạn chương trình sang câu lệnh C tương ứng. Lời giải thích cho hợp ngữ cần sát với ý nghĩa của câu lệnh C. Giả thiết rằng các biến f, g, h, i, j được gán tương ứng trong các thanh ghi \$s0, \$s1, \$s2, \$s2, \$s3, \$s4; A và B là hai mảng dữ liệu các phần tử số nguyên 32-bit có địa chỉ cơ sở tương ứng nằm trong các thanh ghi \$s6 và \$s7.

a.

```
sll    $s2, $s4, 1
add    $s0, $s2, $s3
add    $s0, $s0, $s1
```

b.

```
sll    $t0, $s0, 2
add    $t0, $s6, $t0
sll    $t1, $s1, 2
add    $t1, $s7, $t1
lw     $t3, 0($t0)
addi   $t2, $t0, 4
lw     $t4, 0($t2)
add    $t5, $t3, $t4
sw     $t5, 0($t1)
```

3.4. Cho đoạn chương trình hợp ngữ sau:

```
lw    $t2, 32($0)
add   $t0, $s4, $s5
addi  $t0, $s3, -12
sub   $t0, $t3, $t5
```

Chuyển đổi các lệnh trên về dạng mã máy viết theo hệ Hexa

3.5. Chuyển đổi các lệnh mã máy sau đây về dạng lệnh hợp ngữ :

0x2237FFF1

0x02F34022

0x8E93FFE8

3.6. (2.19-COD) Assume the following register contents:

\$t0 = 0xAAAAAAAA, \$t1 = 0x12345678

1. For the register values shown above, what is the value of \$t2 for the following sequence of instructions?

```
sll   $t2, $t0, 4
or    $t2, $t2, $t1
```

2. For the register values shown above, what is the value of \$t2 for the following sequence of instructions?

```
sll   $t2, $t0, 4
andi  $t2, $t2, -1
```

3. For the register values shown above, what is the value of \$t2 for the following sequence of instructions?

```
srl   $t2, $t0, 3
andi  $t2, $t2, 0xFFEF
```

3.7. Cho đoạn chương trình vòng lặp viết bằng hợp ngữ của MIPS sau đây:

```
        addi $t1, $zero, 8
        add  $s2, $zero, $zero
LOOP:   slt  $t2, $zero, $t1
        beq  $t2, $zero, DONE
        addi $s2, $s2, 3
        sll  $s2, $s2, 1
        addi $t1, $t1, -1
        j    LOOP
DONE:
```

- Tính số lệnh được thực hiện khi chạy đoạn chương trình trên.
- Xác định giá trị thanh ghi \$s2 sau khi thực hiện đoạn chương trình trên.

3.8. Cho đoạn chương trình vòng lặp viết bằng hợp ngữ của MIPS sau đây:

```

LOOP:  slt   $t2, $zero, $t1
        bne  $t2, $zero, ELSE
        j    DONE
ELSE:   addi $s2, $s2, 2
        addi $t1, $t1, -1
        j    LOOP
DONE:

```

- Giả thiết các thanh ghi \$t1, \$s2 được khởi tạo các giá trị ban đầu là \$t1 = 18, \$s2 = 0, hãy xác định giá trị thanh ghi \$s2 sau khi thực hiện đoạn chương trình trên.
- Với vòng lặp hợp ngữ trên, giả sử thanh ghi \$t1 được khởi tạo giá trị bằng N (với N nguyên dương), hãy xác định khi thực hiện đoạn chương trình trên thì có bao nhiêu lệnh được thực hiện?

3.9. Cho đoạn mã C sau đây:

```

while (i < 10) {
    M[i] = a + i;
    i += 1;
}

```

Hãy dịch đoạn mã C ở trên thành đoạn mã hợp ngữ của MIPS với số lệnh tối thiểu (*yêu cầu không sử dụng lệnh giả*). Trong đó M là mảng dữ liệu các phần tử 32-bit. Giả thiết rằng các giá trị i, a nằm thanh ghi \$s0, \$s1 tương ứng, thanh ghi \$s2 chứa địa chỉ cơ sở của mảng M. (*Lưu ý: các lệnh của hợp ngữ cần viết lời giải thích*).

3.10. Cho M là mảng các phần tử số nguyên có dấu 32-bit, có địa chỉ cơ sở nằm trong thanh ghi \$s1 và số phần tử của mảng M được cho trong thanh ghi \$s2. Viết đoạn chương trình hợp ngữ MIPS (*yêu cầu không sử dụng lệnh giả*) để tìm phần tử lớn nhất của mảng M cất vào thanh ghi \$t0 theo thuật giải viết bằng C dưới đây. (*Lưu ý: lời giải thích cho các lệnh hợp ngữ cần mô tả sát với thuật giải*).

```

max = M[0];
for (i=1; i < n; i++) {
    if ( M[i] > max ) max = M[i];
}

```