Trần Ngọc Công - 20176707

**Bài tập chương 2:**

2.1. 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.

**Bài làm:**

Câu lệnh thứ nhất gán giá trị vào 2 byte cao của thanh ghi => giá trị $s0 là 0x40500000

Câu lệnh thứ hai thực hiện phép logic or => giá trị thanh ghi $s0 là0x405000FB

Câu lệnh thứ ba dịch giá trị lưu trong $s0 sang trái 4 bit rồi lưu vào $s1 => Giá trị thanh ghi $s1 là 0x05000FB0

Câu lệnh thứ tư cộng giá trị trong $s0 và $s1 lưu vào $s2 => Giá trị thanh ghi $s2 là 0x455010AB

Câu lệnh thứ tư cộng giá trị trong $s1 và 0x0FE7 lưu vào $s2 => Giá trị thanh ghi $s3 là 0x00000FA0

2.2. 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.

1. f = - g - A[4]
2. B[8] = A[i-j]

**Bài làm:**

a)

.data

A: .word 0,1,2,3,4

B: .word 0,1,2,3,4,5,6,7,8

.text

addi $s1, $zero, 5 # g =5

la $s6, A # Lưu địa chỉ của A vào $s6

lw $t0, 16($s6) # $t0 = A[4]

sub $t1, $zero, $s1 # $t1 = -g

sub $s0, $t1, $t0 # f = $s0 = -g - A[4]

b)

.data

A: .word 0,1,2,3,4

B: .word 0,1,2,3,4,5,6,7,8

.text

addi $s3, $zero, 3 # i = 3

add $s4, $zero, 1 # j = 1

la $s6, A # Lưu địa chỉ của A vào $s6

la $s7, B # Lưu địa chỉ của B vào $s7

sub $t2, $s3, $s4 # $t2 = i - j

sll $t2, $t2, 2 # 4\*(i-j)

add $t3, $s6, $t2 # $t3 = Địa chỉ của A[i-j]

lw $t4, 0($t3) # $t4 = A[i-j]

sw $t4, 32($s7) # B[8] = A[i-j]

2.3. 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)

**Bài làm:**

1. sll $s2, $s4, 1 #dịch trái 1 bit => $s2 =$s4\*2

add $s0, $s2, $s3 # $s0 =$s2 + $s3

add $s0, $s0, $s1 # $s0 = $s0 +$s1

* Lệnh trong C:

h = 2 \* j;

f = h + i;

f += g

b) sll $t0, $s0, 2 #dịch trái 1 bit => $t0 =$s0 \* 2

add $t0, $s6, $t0 #$t0 = $t0 + $s6

sll $t1, $s1, 2 #dịch trái 2 bit => $t1 =$s1 \*4

add $t1, $s7, $t1 #$t1 =$t1 + $s7

lw $t3, 0($t0) #$t3 = giá trị lưu ở địa chỉ trong $t0

addi $t2, $t0, 4 # $t2 =$t0 + 4

lw $t4, 0($t2) #$t4 = giá trị lưu ở địa chỉ trong $t2

add $t5, $t3, $t4 #$t5 =$t3 + $t4

sw $t5, 0($t1) # lưu giá trị $t5 vào địa chỉ trong $t1

* Lệnh trong C: B[g] = A[f]+A[f+1]

2.4. 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:

a. Tính số lệnh được thực hiện khi chạy đoạn chương trình trên.

b. Xác định giá trị thanh ghi $s2 sau khi thực hiện đoạn chương trình trên.

**Bài làm:**

1. Trước vòng lặp thực hiện 2 câu lệnh

Khi $t1 >= 0 thực hiện vòng lặp => lặp 8 vòng, mỗi vòng có 6 câu lệnh =>48 câu lệnh

Khi $t1 = -1 vẫn thực hiện 2 câu lệnh kiểm tra trước khi kết thúc chương trình => 2 câu lệnh

Tổng: 2 + 48 + 2 =52 câu lệnh

1. Giá trị ban đầu của $s2 là 0

Các câu lệnh trong vòng lặp thực hiện thay đổi với $s2

addi $s2, $s2, 3 # $s2 = $s2 +3

sll $s2, $s2, 1 # $s2 = s2\*2

Sau 8 vòng lặp giá trị trong $s2 là 1530(dec)

2.5. 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:

1. 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.
2. 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?

**Bài làm:**

a) $s2 = 2\*18 = 36 (ELSE thực hiện 18 lần)

b) Với $t1 = N thì ELSE sẽ được gọi N lần

Tổng số lệnh là: 3N + 2(N+1) + 1 = 5N + 3

2.6. 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*).

**Bài làm:**

.data

M: .word # khai báo mảng M

.text

lui $at, 0x1001 # Set 16 bit đầu của $at bởi 16 bit của hằng số

ori $s2, $at, 0x0000 # $s2 lưu giá trị địa chỉ của đầu mảng M

addi $s0, $zero, 0 # i = 0

addi $s1, $zero, 1 # a = 1

while:

slti $t0, $s0, 10 # i < 10 ? $t0 = 0 : $t0 = 1

beq $t0, $zero, end\_while # không phải i < 10 -> end\_while

sll $t1, $s0, 2 # $t1 = 4i

add $t3, $s2, $t1 # $t3 = Address M[i]

add $t2, $s1, $s0 # $t2 = a + i

sw $t2, 0($t3) # M[i] = a + i

addi $s0, $s0, 1 # i++

j while

end\_while:

2.7. 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];

}

.data

M: .word 1,2,3,4,5 # Khai báo mảng M

.text

lui $at, 0x1001 # Set 16 bit đầu của $at bởi 16 bit của hằng số

ori $s1, $at, 0x0000 # $s1 lưu giá trị địa chỉ của đầu mảng M

addi $s2, $zero, 5 # $s2 = n = 5 = số lượng phần tử của M

lw $t0, 0($s1) # Max = $t0 = M[0]

add $t1, $zero, 1 # i = $t1 = 1

loop:

slt $t2, $t1, $s2 # Nếu i < n thì $t2 = 1

beq $t2, $zero, end\_loop # Không phải i < n thì kết thúc chương trình

sll $t3, $t1, 2 # $t3 = 4i

add $t3, $s1, $t3 # $t3 = Địa chỉ M[i]

lw $t5, 0($t3) # $t5 = M[i]

slt $t4, $t5, $t0 # if M[i] > Max thì $t4 = 1

bne $t4, $zero, if\_end # M[i] < Max thì nhảy đến nhãn if\_end

lw $t0, 0($t3) # Max = M[i]

if\_end:

addi $t1, $t1, 1 # i++

j loop

end\_loop: