

Bài Tập Chương 2 (Thêm)

---oOo---

Lưu ý: Các mảng sử dụng trong chương này đều là mảng mà mỗi phần tử chứa 1 word/từ nhớ, mỗi từ nhớ chứa 4 bytes.

Bài 1. Tìm mã máy (biểu diễn hệ 16) cho các lệnh assembly của MIPS sau:

- a. *and \$t3, \$s0, \$s2*
- b. *sll \$t1, \$t5, 7*
- c. *addi \$t0, \$s3, 25*
- d. *addi \$t0, \$s3, -25*
- e. *lw \$t0, 24(\$s0)*
- f. *lw \$t0, -24(\$s0)*
- g. *sw \$t2, 48(\$s0)*
- h. *sw \$t2, -48(\$s0)*

Bài 2. Cho các mã máy như sau, hỏi tương ứng với từng mã máy là lệnh assembly gì của MIPS

- a. *0x01304024*
- b. *0x2128fff3 (0x2128FFF3) (Trong hệ 16, các chữ từ a tới f có thể viết thường hoặc hoa đều được)*
- c. *0xad28fffc*

Bài 3. Cho đoạn lệnh assembly sau, cho biết kết quả sau khi chạy

- a. *and \$t0, \$s0, \$s1*
or \$t1, \$s0, \$s1
nor \$t0, \$t0, \$t1
sll \$t0, \$t0, 3

Biết trước khi chạy: *\$s0 = 0x12345678; \$s1 = 0x00000007*

Hỏi sau khi chạy xong đoạn lệnh trên, *\$s0, \$s1, \$t0, \$t1* bằng bao nhiêu

- b. *andi \$t0, \$s0, 12*
nor \$t0, \$t0, \$zero
ori \$t0, \$t0, 3
srl \$t0, \$t0, 2

Biết trước khi chạy: $\$s0 = 0x0000000f$

Hỏi sau khi chạy xong đoạn lệnh trên, $\$s0$, $\$t0$ bằng bao nhiêu

c.

```

    slt    $t2, $t0, $t1
    beq    $t2, $zero, ELSE
    add    $t2, $t2, $t0
    j      DONE
ELSE: add  $t2, $t2, $t1
DONE:

```

Biết trước khi chạy: $\$t0 = 0x00000008f$; $\$t1 = 0x00000009f$

Hỏi sau khi chạy xong đoạn lệnh trên, $\$t2$ bằng bao nhiêu

d.

```

    addi   $s0, $zero, 2
    addi   $t1, $zero, 6
loop: beq  $t1, $zero, end
    sll    $s0, $s0, 1
    addi   $t1, $t1, -1
    j      loop
end:  addi  $s1, $s0, 2

```

Sau đoạn chương trình này thì giá trị trong thanh ghi $\$s0$ là bao nhiêu?

Bài 4. Chuyển các đoạn lệnh C sau sang assembly của MIPS.

Biết i và j tương ứng với các thanh ghi $\$s0$ và $\$s1$. Mảng A là mảng mà các phần tử là số nguyên, mỗi phần tử chiếm 1 từ nhớ (4 bytes) và địa chỉ nền của mảng A lưu trong thanh ghi $\$s3$

a.

```

if (i < j) {
    A[i] = A[i] + 1;
    A[i+1] = 5;
}else{
    A[i] = A[i] - 1;
    A[i+1] = 10;
}
i++;

```

b.

```
if (i <= j && j > 0)
    A[j] = A[i] + A[i+1];
else
    A[j] = A[i] - A[i+1];
}
i++;
```

c.

```
while (i > 0){
A[i+1] = A[i]*8;
i--;
}
A[0] = 5;
```

d.

```
j = value;
for(i = 1; i < j; i++)
    A[i] = B[i];
j = 0;
(Với địa chỉ nền mảng B đang lưu trong thanh ghi $s4 và biến value tương ứng thanh ghi $s5)
```

e.

```
j = value;
max = 0;
for(i = 0; i < j; i++)
    if(A[i] > max) max = A[i];
j = 0;
(Với biến max tương ứng với thanh ghi $s4)
```