

# BÀI TẬP CHƯƠNG 2

## Kiến trúc tập lệnh

---oOo---

Các bài tập chương này được trích dẫn và dịch lại từ:

*Computer Organization and Design: The Hardware/Software Interface*,  
Patterson, D. A., and J. L. Hennessy, Morgan Kaufman, **Third Edition**, 2011.

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.

#### Phần 1.

a)  $f = g + h + i + j$ ;

b)  $f = g + (h + 5)$ ;

$g, h, i, j$  là những số nguyên 32-bit

**1.1** Hãy tìm mã hợp ngữ MIPS tương đương các lệnh C trên.

a/ `add $f, $g, $h.`

`add $f, $f, $i.`

`add $f, $f, $j.`

b/ `addi $f, $h, 5.`

`add $f, $f, $g.`

**1.2** Có bao nhiêu lệnh MIPS để hiện thực các lệnh C trên.

- ở biểu thức a có 3 lệnh, b có 2 lệnh. Vậy tổng là 5 lệnh để thực hiện 2 biểu thức

**1.3** Nếu các biên  $f, g, h, i$  và  $j$  có giá trị tương ứng là 1, 2, 3, 4, 5 thì giá trị của  $f$  là bao nhiêu?

a/  $f = 2 + 3 + 4 + 5 = 14$

b/  $f = 10$

#### Phần 2.

a.

`add f, g, h`

b.

`addi f, f, 1`

`add f, g, h`

**1.3** Tìm lệnh C tương đương với các lệnh hợp ngữ MIPS trên.

$$a/f = g + h;$$

$$b/f = f + 1;$$

$$f = g + h;$$

**1.5** Nếu các giá trị  $f$ ,  $g$ ,  $h$  và  $i$  có giá trị tương ứng 1, 2, 3 và 4 thì giá trị cuối cùng của  $f$  là bao nhiêu?

$$a/f = 2 + 3 = 5$$

$$b/f = 2 + 3 = 5$$

## Bài 2.

### Phần 1.

$$a. f = g + h + B[4];$$

$$b. f = g - A[B[4]];$$

$f$ ,  $g$ ,  $h$ ,  $i$ ,  $j$  được lưu lần lượt ở các thanh ghi  $\$s0$ ,  $\$s1$ ,  $\$s2$ ,  $\$s3$ ,  $\$s4$

Địa chỉ cơ sở/nền (base address) của mảng A và B được lưu trong các thanh ghi  $\$s6$ ,  $\$s7$ .

Mỗi phần

**2.1** Hãy chuyển các câu lệnh C bên trên sang dạng hợp ngữ MIPS.

$$a/ lw \$s0, 16(\$s7)$$

$$add \$s0, \$s0, \$s1$$

$$add \$s0, \$s0, \$s2$$

$$b/lw \$t0, 16(\$s7) \quad \# \$t0=B[4]$$

$$sll \$t0, \$t0, 2 \quad \# \$t0=B[4] * 4$$

$$add \$t0, \$t0, \$s6 \quad \# \$t0=\&A[B[4]]$$

$$lw \$s0, 0(\$t0)$$

$$sub \$s0, \$s1, \$s0$$

**2.2** Cần bao nhiêu lệnh hợp ngữ MIPS để có chức năng tương đương với từng câu lệnh C?

$$a/3$$

$$b/5$$

**2.3** Có bao nhiêu thanh ghi khác nhau được dùng cho từng câu lệnh C bên trên.

$$a/4$$

$$b/5$$

### Phần 2.

a.

$$add \$s0, \$s0, \$s1$$

$$add \$s0, \$s0, \$s2$$

$$add \$s0, \$s0, \$s3$$

$$add \$s0, \$s0, \$s4$$

b.

$$lw \$s0, 4(\$s6)$$

**2.4** Hãy tìm câu lệnh C tương đương với các câu lệnh hợp ngữ MIPS bên trên.

$$a/add \$s0, \$s0, \$s1 \Rightarrow f=f+h$$

$$add \$s0, \$s0, \$s2 \Rightarrow f=f+h+g$$

$$add \$s0, \$s0, \$s3 \Rightarrow f=f+h+g+i$$

add \$s0, \$s0, \$s4 => f=f+h+g+i+j  
f=f+h+g+i+j

b/ f = A[1]

**2.5** Hãy thử rút gọn số lượng lệnh hợp ngữ MIPS trên nếu có thể.

a/Không thể

b/Không thể

**2.6** Có bao nhiêu thanh ghi được sử dụng trong đoạn hợp ngữ trên? Nếu ta có thể rút gọn số lệnh thì ta cần bao nhiêu thanh ghi?

a/5

b/2

=>Không thể rút gọn

### Bài 3.

a.  $f = -g + h + B[1]$ ;

b.  $f = A[B[g] + 1]$ ;

$f, g, h, i, j$  được lưu lần lượt tại các thanh ghi \$s0, \$s1, \$s2, \$s3, \$s4

Địa chỉ cơ sở của hai chuỗi A và B được lưu trong các thanh ghi \$s6, \$s7

**3.1** Hãy tìm các lệnh hợp ngữ MIPS tương đương với các câu lệnh C bên trên.

a/ lw \$s0, 4(\$s7)

sub \$s0, \$s0, \$s1

add \$s0, \$s0, \$s2

b/  $f = A[B[g] + 1]$ ;

# tìm B[g]

sll \$t0, \$s1, 2

add \$t0, \$t0, \$s7

lw \$t0, 0(\$t0) # \$t0 = B[g]

# tính B[g] + 1

addi \$t0, \$t0, 1 # \$t0 = B[g] + 1

# tìm A[B[g] + 1]

sll \$t0, \$t0, 2

add \$t0, \$t0, \$s6

lw \$s0, 0(\$t0)

**3.2** Cần bao nhiêu lệnh hợp ngữ MIPS để có chức năng tương đương với từng câu lệnh C?

a/ 3 lệnh

b/ 7 lệnh

**3.3** Có bao nhiêu thanh ghi khác nhau được dùng cho từng câu lệnh C bên trên.

a/ 4 thanh

b/ 4 thanh

### Bài 4.

Các câu hỏi dưới đây liên quan đến mở rộng dấu và tràn.

Thanh ghi  $\$s0$  và  $\$s1$  lưu các giá trị như bảng bên dưới. Hãy trả lời các câu hỏi liên quan đến lệnh hợp ngữ MIPS bên dưới và tính toán các kết quả.

a.	$\$s0 = 0x70000000; \$s1 = 0x0FFFFFFF$
b.	$\$s0 = 0x40000000; \$s1 = 0x40000000$

*Ghi chú: Khi 0x trước một giá trị thì giá trị đó đang biểu diễn trong hệ 16*

#### 4.1

Tính kết quả của  $\$t0$  sau khi thực hiện câu lệnh: `add $t0, $s0, $s1`

Kết quả trong thanh ghi  $\$t0$  đúng như mong muốn của phép toán chưa? Có xảy ra tràn không?

a/ $\$t0$  sau khi thực hiện lệnh =  $0x7FFFFFFF$

Kết quả như mong muốn, không tràn.

b// $\$t0$  sau khi thực hiện lệnh =  $0x80000000$

Phép toán add được thực hiện trên số có dấu (dùng bù hai). Phép cộng trên thực hiện cộng hai số dương, nhưng kết quả  $0x80000000$  rõ ràng là số âm => phép toán bị tràn

#### 4.2

Tính kết quả của  $\$t0$  sau khi thực hiện câu lệnh: `sub $t0, $s0, $s1`

Kết quả trong thanh ghi  $\$t0$  đúng như mong muốn của phép toán chưa? Có xảy ra tràn không?

a)  $\$t0 = 0x60000001$ , không tràn

b)  $\$t0 = 0$ , không tràn

#### 4.3

Tính kết quả của  $\$t0$  sau khi chạy chuỗi lệnh:

`add $t0, $s0, $s1`

`add $t0, $t0, $s0`

Kết quả trong thanh ghi  $\$t0$  đúng như mong muốn của phép toán chưa? Có xảy ra tràn không?

a)  $\$t0 = 0x70000000 + (0x70000000 + 0x0FFFFFFF) = 0xEEFFFFFFF$

Tràn, bởi vì cộng hai số dương nhưng bit dấu của kết quả lại là âm, không đúng.

b)  $\$t0 = 0x40000000 + (0x40000000 + 0x40000000) = 0xC0000000$

Tràn, bởi vì cộng hai số dương nhưng bit dấu của kết quả lại là âm, không đúng.

### Bài 5.

Chuyển các mã máy sau sang dạng hợp ngữ MIPS

a.	1010 1110 0000 1011 0000 0000 0000 0100 <sub>two</sub>
b.	1000 1101 0000 1000 0000 0000 0100 0000 <sub>two</sub>

**5.1 & 5.2.** Từ các giá trị binary ở bảng trên, hãy xác định chuỗi nhị phân thể hiện là lệnh gì?

Xác định các lệnh trên là thuộc kiểu lệnh gì (I-type, R-type, J-type).

<b>R</b>	opcode	rs	rt	rd	shamt	funct
	31 26 25	21 20	16 15	11 10	6 5	0
<b>I</b>	opcode	rs	rt	immediate		
	31 26 25	21 20	16 15	0		
<b>J</b>	opcode	address				
	31 26 25	0				

Chú ý: Tham khảo “MIPS reference data” (trang 2, sách tham khảo) để dò tìm opcode của các lệnh.

### 5.3

Nếu chuỗi nhị phân trên chỉ là dữ liệu đơn thuần. Hãy chuyển chúng sang dạng mã HEX.

### 5.4 & 5.5

Hãy dịch các lệnh sau sang dạng mã máy

a) add \$t0, \$t0, \$zero

b) lw \$t1, 4(\$s3)

5.6 Hãy trình bày dưới dạng mã HEX của các trường opcode, Rs và Rt của các lệnh trên.

Đối với các lệnh kiểu R, hãy trình bày dưới dạng mã HEX của các trường Rd và funct.

Đối với các lệnh kiểu I, hãy trình bày dưới dạng mã HEX trường trực tiếp (immediate field).

### Bài 6.

Cho giá trị của các thanh ghi sau:

<b>a.</b>	\$t0 = 0x55555555, \$t1 = 0x12345678
<b>b.</b>	\$t0 = 0xBEADFEED, \$t1 = 0xDEADFADE

6.1 Hãy cho biết giá trị của thanh ghi \$t2 sau khi chạy các lệnh sau:

*sll \$t2, \$t0, 4*

*or \$t2, \$t2, \$t1*

a) 0x57755778

b) 0xFEFFFEDE

6.2 Hãy cho biết giá trị của thanh ghi \$t2 sau khi chạy các lệnh sau:

*srl \$t2, \$t0, 3*

*andi \$t2, \$t2, 0xFFEF*

a) 0x0000AAAA

b) 0x0000BFCD

### Bài 7.

Giá trị của thanh ghi \$t0 được cho trong bảng bên dưới

<b>a.</b>	1010 1101 0001 0000 0000 0000 0000 0010 <sub>two</sub>
<b>b.</b>	1111 1111 1111 1111 1111 1111 1111 1111 <sub>two</sub>

**7.1**

Giả sử rằng thanh ghi \$t0 chứa giá trị ở bảng trên và \$t1 có giá trị

0011 1111 1111 1000 0000 0000 0000 0000<sub>two</sub>

Hãy cho biết giá trị của \$t2 sau khi chạy các lệnh dưới

```

        slt    $t2, $t0, $t1           # set on less than
        beq    $t2, $zero, ELSE        # go to ELSE if $t2=0
        j      DONE                   # go to DONE
ELSE:    addi   $t2, $zero, 2           # $t2 = 0+2
DONE:

```

**7.2**

Giả sử rằng thanh ghi \$t0 chứa giá trị trong bảng trên và được so sánh với giá trị X bằng lệnh MIPS bên dưới. Hãy chú ý cấu trúc của lệnh slti. Tìm giá trị của X (nếu có) để \$t2 có giá trị là 1.

slti \$t2, \$t0, X

**7.3**

Giả sử con trỏ PC đang có giá trị 0x0000 0020.

Ta có thể sử dụng lệnh nhảy trong hợp ngữ MIPS (lệnh j) để nhảy đến địa chỉ có giá trị như trong bảng trên hay không.

Tương tự, ta có thể sử dụng lệnh nhảy-nếu-bằng (lệnh beq) để nhảy đến địa chỉ có trong bảng trên hay không.

**Bài 8.** Tìm mã máy (biểu diễn hệ 16) cho các lệnh assembly của MIPS sau:

- and \$t3, \$s0, \$s2*
- sll \$t1, \$t5, 7*
- addi \$t0, \$s3, 25*
- addi \$t0, \$s3, -25*
- lw \$t0, 24(\$s0)*
- lw \$t0, -24(\$s0)*
- sw \$t2, 48(\$s0)*
- sw \$t2, -48(\$s0)*

**Bài 9.** Cho các mã máy như sau, hỏi tương ứng với từng mã máy là lệnh assembly gì của MIPS

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

**Bài 10.** Cho đoạn lệnh assembly sau, cho biết kết quả sau khi chạy

- 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 11.** 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)*