CHƯƠNG 2. Instructions: Language of the Computer

BÀI TẬP

Câu 1. Sinh viên viết chương trình nhập vào hai số nguyên và hiện thị kết quả các phép toán +, - , *, /

Bài làm:

```
.data
myString1   :       .asciiz "BAI TAP VE NHA"
myString21  :       .asciiz "NHAP GIA TRI CUA A"
myString22  :       .asciiz "GIA TRI CUA A: "
myString31  :       .asciiz "NHAP GIA TRI CUA B"
myString32  :       .asciiz "GIA TRI CUA B: "
myString4: .asciiz "GIA TRI A-B: "
myString5: .asciiz "GIA TRI A+B: "
myString6: .asciiz "GIA TRI A*B: "
myString7: .asciiz "GIA TRI A%B: "
myString8: .asciiz "\n"
text.
Main:
```

1.	2.
li $v0, 4 la $a0, myString1 syscall li $v0, 4 la $a0, myString8 syscall li $v0, 4 la $a0, myString21 syscall li $v0, 5 syscall	syscall li $v0, 4 la $a0, myString31 syscall li $v0, 5 syscall move $s2, $v0 li $v0, 4 la $a0, myString32 syscall li $v0, 1

```
        move $s1, $v0
        li $v0, 4
        la $a0, myString22
        syscall
        li $v0, 1
        move $a0, $s1
        syscall
        li $v0, 4
        la $a0, myString8
```

```
        move $a0, $s2
        syscall
        li $v0, 4
        la $a0, myString8
```

3.

```
    syscall
        sub $s3, $s1, $s2
        li $v0, 4
        la $a0, myString4
        syscall
        li $v0, 1
        move $a0, $s3
        syscall
        li $v0, 4
        la $a0, myString8
        syscall
        add $s4, $s1, $s2
        li $v0, 4
        la $a0, myString5
        syscall
        li $v0, 1
        move $a0, $s4
        syscall
        li $v0, 4
        la $a0, myString8
```

4.

```
syscall
        mult $s1, $s2
        mflo $s5
        li $v0, 4
        la $a0, myString6
        syscall
        li $v0, 1
        move $a0, $s5
        syscall
        li $v0, 4
        la $a0, myString8
        syscall
        div $s1, $s2
        mflo $s6
        li $v0, 4
        la $a0, myString7
        syscall
        li $v0, 1
        move $a0, $s6
        syscall
        li $v0, 4
        la $a0, myString8
        syscall
```

Exit:

Kết quả:

```
BAI TAP VE NHA
NHAP GIA TRI CUA A 18
GIA TRI CUA A: 18
```

NHAP GIA TRI CUA B 9 GIA TRI CUA B: 9 GIA TRI A-B: 9 GIA TRI A+B: 27 GIA TRI A*B: 162 GIA TRI A%B: 2

-- program is finished running (dropped off bottom) --

Câu 2. Đối với câu lệnh C sau đây, mã hợp ngữ MIPS tương ứng là gì? Giả sử rằng các biến f, g, h và i đã cho và có thể được coi là số nguyên 32 bit như được khai báo trong chương trình C. Sử dụng số lượng tối thiểu các hướng dẫn lắp ráp MIPS.

f = g + (h − 5);

Bài làm

addi $S1, $S2, -5; //$S1=F, $S2=H, F=H-5

add $S1, $S3, $S1;//S3 = G, F=G+H-5

Câu 3. Đối với các hướng dẫn lắp ráp MIPS sau đây ở trên, một

câu lệnh C tương ứng?

add f, g, h (1)

add f, i, f (2)

Bài làm

Lệnh (1)	f = g + h
Lệnh (2)	f = i + f
Lệnh (1)(2)	f = i + g + h

Câu 4. Đối với câu lệnh C sau đây, câu lệnh tương ứng là gì?Mã lắp ráp MIPS? Giả sử rằng các biến f, g, h, i và j được gán lần lượt vào các thanh ghi $s0, $s1, $s2, $s3 và $s4. Giả sử rằng địa chỉ cơ sở của mảng A và B lần lượt nằm trong các thanh ghi $s6 và $s7.

B[8] = A[i−j];

Bài làm

Lệnh	Ghi chú
Sub $t0, $s3, $s4	$t0 = i - j

Sll $t0, $t0, 2	Dịch $t0 sang trái 4 byte
Add $t0, $t0, &s6	Lấy địa chỉ thực tế bằng cách lấy địa chỉ mảng cộng cho địa chỉ hiện tại &A[i-j]
Lw $t1, 0($t0)	Lấy giá của A[i-j] lưu vào $t1
Sw $t1, 8*4($s7)	Gán giá trị $t1 vào B[8]

Câu 5. Đối với các hướng dẫn lắp ráp MIPS bên dưới, câu lệnh C tương ứng là gì? Giả sử rằng các biến f, g, h, i và j được gán lần lượt vào các thanh ghi $s0, $s1, $s2, $s3 và $s4. Giả sử rằng địa chỉ cơ sở của mảng A và B lần lượt nằm trong các thanh ghi $s6 và $s7.

Bài làm

Lệnh	Giải thích	Kết quả
sll $t0, $s0, 2	Dịch trái f 4 byte lưu vào $t0	$t0 = f << 4 byte
add $t0, $s6, $t0	Lấy địa chỉ mảng A cộng cho địa chỉ $t0 ra địa chỉ cần tìm lưu vào $t0	$t0 = &A[f] + &t0
sll $t1, $s1, 2	Dịch trái g 4 byte lưu vào $t1	$t1 = g << 4 byte
add $t1, $s7, $t1	Lấy địa chỉ mảng B cộng cho địa chỉ $t1 ra địa chỉ cần tìm lưu vào $t1	$t1 = &B[g] + $t1
lw $s0, 0($t0)	Lưu giá trị của f vào A[f]	$s0 = A[f]
addi $t2, $t0, 4	Dịch phải 4 byte A[f]	$t2 = A[f+1]
lw $t0, 0($t2)	Lưu giá trị $t0 cho $t2	$t0 = A[f+1]
add $t0, $t0, $s0	Cộng $t0 với $s0 rồi lưu kết quả vào $t0	A[f+1] = A[f+1] + A[f]
sw $t0, 0($t1)	Gán kết quả $t0 cho $t1	B[g]= A[f+1] + A[f]

Câu 6. Hiển thị cách sắp xếp giá trị 0xabcdef12 trong bộ nhớ của máy little-endian và big-endian. Giả sử dữ liệu được lưu trữ bắt đầu từ địa chỉ 0.

0Xabcdef12

= 1010 1011 1100 1101 1110 1111 0001 0010

- little-endian: Max nằm bên trái:

1111 1110 1101 1100 1011 1010 0010 0001

- big-endian: Max nằm bên phải:

0001 0010 1010 1011 1100 1101 1110 1111

Câu 7. Dịch mã C sau sang MIPS. Giả sử rằng các biến f, g, h, i và j được gán lần lượt vào các thanh ghi $s0, $s1, $s2, $s3 và $s4. Giả sử rằng địa chỉ cơ sở của mảng A và B lần lượt nằm trong các thanh ghi $s6 và $s7. Giả sử rằng các phần tử của mảng A và B là các từ 4 byte:

B[8] = A[i] + A[j];

Bài làm

Lệnh	Ghi chú
Sll $s0, $s3, 2	Dịch trái i 4 byte rồi lưu vào f, f=i<<4
Add $s0, $s0, $s6	Lấy địa chỉ cần tìm &A[i] lưu vào f, f= &A[i]
Lw $s1, 0($s0)	Lưu giá trị của mảng tại ví trị i vào g, g = A[i]
Sll $s0, $s4, 2	Dịch trái j 4 byte rồi lưu vào f, f=j<<4
Add $s0, $s0, $s6	Lấy địa chỉ cần tìm &A[j] lưu vào f, f= &A[j]
Lw $s2, 0($s0)	Lưu giá trị của mảng tại ví trị j vào h, h = A[j]
Add $s0, $s1, $s2	f = A[i] +A[j]
Sw $s0, 8*4($s7)	Gán giá trị f cho mảng B tại vị trí 8 B[8] = A[i] +A[j]

Câu 8. Dịch mã MIPS sau sang C. Giả sử rằng các biến f, g, h, i và j được gán cho các thanh ghi $s0, $s1, $s2, $s3 và $s4, tương ứng. Giả sử rằng địa chỉ cơ sở của mảng A và B lần lượt nằm trong các thanh ghi $s6 và $s7.

Lệnh	Giải thích	Kết quả
addi $t0, $s6, 4	Dịch 4 byte của mảng A[0] lưu vào $t0	$t0 = &A[1]
add $t1, $s6, $0	Lấy chỉ mảng A[0] lưu vào $t1	$t1 = &A[0]
sw $t1, 0($t0)	Gián giá trị $t0 cho $t1	$t1 = &A[0] = A[1]
lw $t0, 0($t0)	Lưu giá trị $t0 vào $t0	$t0 = A[1]
add $s0, $t1, $t0	Lấy $t1 cộng $t0 lưu vào f	f = $A[0]+A[1]

Câu 9. Giả sử rằng các thanh ghi $s0 và $s1 giữ các giá trị tương ứng là 0x80000000 và 0xD0000000.

a. **Giá trị của $t0 cho mã hợp ngữ sau đây là bao nhiêu?**

add $t0, $s0, $s1

$t0 = $s0 + $s1

= 0x80000000 + 0xD0000000

= 0xF00000000

b. Đối với nội dung của các thanh ghi $s0 và $s1 như được chỉ định ở trên, giá trị của $t0 cho mã hợp ngữ sau đây là bao nhiêu?

sub $t0, $s0, $s1

$t0 = $s0 - $s1

= 0x80000000 - 0xD0000000

= 0xFB0000000

c. Đối với nội dung của các thanh ghi $s0 và $s1 như được chỉ định ở trên, giá trị của $t0 cho mã hợp ngữ sau đây là bao nhiêu?

add $t0, $s0, $s1

add $t0, $t0, $s0

$t0 = $s0 + $s1

= 0x80000000 + 0xD0000000

= 0xF00000000

$t0 = $t0 + $s0

= 0xF00000000 + 0x80000000

= 0xF80000000

Câu 10. Cung cấp hướng dẫn về loại và hợp ngữ cho giá trị nhị phân sau: 0000 0010 0001 0000 1000 0000 0010 0000two

000000 10000 10000 10000 00000 100000

Op : 0

Function : 20

⇨ Thanh ghi R lệnh add

Rs : 16

Rt : 16

Rd : 16

Add $s0, $s0, $s0

Câu 11. Cung cấp loại và biểu diễn thập lục phân của lệnh sau: sw $t1, 32($t2)

Lệnh thanh ghi I

Op(2b) : 101011

Rt($t2/10) : 01010

Rs($t1/9) : 01001

Constant of address(32) : 0000 0000 0010 0000

1010 1101 0100 1001 0000 0000 0010 0000

0XAD490020

Câu 12. Cung cấp loại, hướng dẫn hợp ngữ và biểu diễn nhị phân của hướng dẫn được mô tả bởi các trường MIPS sau:

op=0, rs=3, rt=2, rd=3, shamt=0, funct=34

op (0) : 000000

funct(22hec): 100010

➔ Loại thanh ghi R lệnh sub

Rs(3) : 00011/$v1

Rt(2) : 00010/$v0

Rd(3) : 00011/$v1

0000 0000 0110 0010 0001 1000 0010 0010

0X00621822

Sub $v1, $v1,$v0

Câu 13. Cung cấp loại, hướng dẫn hợp ngữ và biểu diễn nhị phân của hướng dẫn được mô tả bằng các trường MIPS sau:

op=0x23, rs=1, rt=2, const=0x4

⇨ Loại thanh ghi I lệnh Lw

Op(23hec) : 100011

Const : 0000 0000 0000 0010

Rs(1) : 00001

Rt(2) : 00010

1000 1100 0010 0010 0000 0000 0000 0010

0X8C220004

Lw $v0, 4($at)

Câu 14. 2.19 Giả sử nội dung thanh ghi sau:

$t0 = 0xAAAAAAAA, $t1 = 0x12345678

a. **Đối với các giá trị thanh ghi được hiển thị ở trên, giá trị của $t2 cho chuỗi lệnh sau là bao nhiêu?**

0XAAAAAAAA

= 1010 1010 1010 1010 1010 1010 1010 1010

0X12345678

= 0001 0010 0011 0100 0101 0110 0111 1000

sll $t2, $t0, 44

Dịch sang trái

$t2 = 0000 0000 0001 0101 0101 0101 0101 0101

or $t2, $t2, $t1

$t2 = 0000 0000 0001 0101 0101 0101 0101 0101

Hoặc

$t2 = 0001 0010 0011 0100 0101 0110 0111 1000

b. **Đối với các giá trị thanh ghi được hiển thị ở trên, giá trị của $t2 cho chuỗi lệnh sau là bao nhiêu?**

sll $t2, $t0, 4

addi $t2, $t2, −1

c. **Đối với các giá trị thanh ghi được hiển thị ở trên, giá trị của $t2 cho chuỗi lệnh sau là bao nhiêu?**

srl $t2, $t0, 3

addi $t2, $t2, 0xFFEF

d. **Giả sử $t0 giữ giá trị 0x00101000. Giá trị của $t2 sau các hướng dẫn sau là bao nhiêu?**

$t0 = 0000 0000 0001 0000 0001 0000 0000 0000

$t0 = 1 052 672

slt $t2, $0, $t0

Dịch 0 sang trái 2^(1 052 672) byte và lưu vào $t2

$t2 = 0000 0000 0000 0000 0000 0000 0000 0000

bne $t2, $0, ELSE

$t2 != 0 nếu:

- Đúng: nhảy tới else
- Sai tiếp túc cấu lệnh tiếp dưới.

$t2=0 ➔ đúng

j DONE

ELSE: addi $t2, $t2, 2

$t2 = $t2 + 2 =2

DONE:

Câu 15. Chuyển đổi đoạn mã MIPS sang ngôn ngữ C?

MIPS:

Bne $s3, $s3, Else

Add $s0, $s1, $s2

J Exit:

Else: Sub $s0, $s1, $s2

Exit:

Biết f,g,h,i và j là năm biến. Năm biến f đến j lần lượt tương ứng với 5 thanh ghi từ $s0 đến $s4.

Bne $s3, $s4, Else
If (i!=j) đúng
Add $s0, $s1, $s2
f = g + h
J Exit:
Thoát vòng if
Ngược lại
	Ngôn ngữ C
	If (i!=j)
	{ f = g + h; }
	Else f = g – h ;

Else: Sub $s0, $s1, $s2 f = g – h Thoát Exit:	

Câu 16. Dịch đoạn Code C sau sang mã MIPS

Code C:

While (save[i] == k)

{ i+=1; }

Biết biến i và k tương ứng với thanh ghi $s3 và $s5, địa chỉ cơ sở của mảng save lưu trong thanh ghi $s6.

Acssemly MIPS	Giải thích
Loop: Sll $t1,$s3,2	Dịch i sang trái 4 byte để lấy địa chỉ của i trong mảng save
Add $t0, $t0, $s6	Cọng địa chỉ của i với địa chỉ cơ sở của mảng save để có được địa chỉ cần tìm của Save[i]
Lw $t0, 0($t1)	Load dữ liệu tại địa chỉ Save[i] vào thanh ghi $t0
Bne $t0, $s5, Exit	So sánh xem $t0 != k không, nếu bằng tới exit, ngược lại tới câu lệnh tiếp theo
Addi $s3, $s3, 1 J Loop:	Tính i = i+1. Lặp
Exit	Thoát

Câu 17. Đỗi mã Asscemly MIPS sau sang mã máy

MIPS: Add $s1,$s2,$s3

Op : 00 0000

Rs : 1 0010

Rt : 1 0011

Rd : 1 0001

Shamt : 0 0000

Func : 10 0000

Mã máy: 0000 0010 0101 0011 1000 1000 0010 0000

⇨ **Ox02538820**

MIPS: Beq $s1, $s2, Address (Biết Beq có địa chỉ Ox00400010 và địa chỉ của Address Ox00400000)

if(R[rs]==R[rt]) PC=PC+4+BranchAddr

BranchAddr = { 14{immediate[15]}, immediate, 2'b0 }

Op : 00 0100

Rs : 1 0001

Rt : 1 0010

Constant :

Ox00400000 = Ox00400010 + 4 + { 14{immediate[15]}, immediate, 2'b0 }

⇨ { 14{immediate[15]}, immediate, 2'b0 } = 0000 0000 0100 0000 0000 0000 0000 0000

– 0000 0000 0100 0000 0000 0000 0001 0000 - 0000 0000 0000 0000 0000 0000 0000 0100

⇨ { 14{immediate[15]}, immediate, 2'b0 } = 1111 1111 1111 1111 1111 1111 1110 1100

⇨ **Immediate =** 1111 1111 1111 1111 1011

Mã máy: 0001 0010 0011 0010 1111 1111 1111 1011

⇨ **Ox1232FFFB**

MIPS: Slti $s1, $s2, -19

R[rt] = (R[rs] < SignExtImm)? 1 : 0

Op : 00 1010

Rs : 1 0001

Rt : 1 0010

Constant : 1111 1111 1110 1101

Mã máy: 0010 1010 0011 0010 1111 1111 1110 1101

⇨ **Ox2A32FFED**

Jal Label (Biết Label có địa Ox00400000 và Jal có địa chỉ Ox00400014)

PC=JumpAddr

JumpAddr = { PC+4[31:28], address, 2'b0 }

Op : 00 0011

Address :

Ox00400000 = { PC+4[31:28], address, 2'b0 }

⇨ Ox00400000 = {Ox00400014 + 4[31:28], address, 2'b0}
⇨ Ox00400000 = {Ox00400018[31:28], address, 2'b0}
⇨ Ox00400000 = {4'b0, address, 2'b0}
⇨ 0000 0000 0100 0000 0000 0000 0000 0000 = {4'b0, address, 2'b0}
⇨ Address = 0000 0100 0000 0000 0000 0000 00

Mã máy: 0000 1100 0001 0000 0000 0000 0000 0000

⇨ Ox0C100000