

ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KỸ THUẬT MÁY TÍNH



BÁO CÁO BÀI THỰC HÀNH 1:
CE119 - THỰC HÀNH KIẾN TRÚC MÁY TÍNH

CHỦ ĐỀ
CE119-Lab01/IT012-Lab03

Giảng viên hướng dẫn: ThS. Trần Văn Quang

Sinh viên thực hiện:

Đỗ Minh Hội - 23520550

1. Lý thuyết

Giảng viên hướng dẫn sinh viên sử dụng phần mềm MARS dựa theo tài liệu:

MARS – chương trình mô phỏng hợp ngữ (assembly) MIPS

2. Thực hành

2.1 Sinh viên tìm hiểu tài liệu “Một số lệnh assembly MIPS cơ bản” và mô phỏng việc thực thi các lệnh và cho biết chức năng của các lệnh cơ bản sau:

add, addi, addu, addiu, sub, subu, and, andi, or, nor, lw, sw, slt, slti, sltu, sltiu, syscall

1.add: Cộng hai giá trị từ hai thanh ghi và lưu kết quả vào thanh ghi đích (có kiểm tra tràn).

- Cú pháp: add \$d, \$s, \$t
- Ví dụ: add \$t0, \$t1, \$t2 // $\$t0 = \$t1 + \$t2$

2.addi: Cộng giá trị hằng số với giá trị trong thanh ghi và lưu kết quả vào thanh ghi đích (có kiểm tra tràn).

- Cú pháp: addi \$d, \$s, imm
- Ví dụ: addi \$t0, \$t1, 10 // $\$t0 = \$t1 + 10$

3.addu: Cộng hai giá trị từ hai thanh ghi mà không kiểm tra tràn.

- Cú pháp: addu \$d, \$s, \$t
- Ví dụ: addu \$t0, \$t1, \$t2 // $\$t0 = \$t1 + \$t2$

11.addiu: Cộng giá trị hằng số với giá trị trong thanh ghi mà không kiểm tra tràn.

- Cú pháp: addiu \$d, \$s, imm
- Ví dụ: addiu \$t0, \$t1, 10 // $\$t0 = \$t1 + 10$

12.sub: Trừ giá trị trong thanh ghi thứ hai khỏi giá trị trong thanh ghi thứ nhất và lưu kết quả vào thanh ghi đích (có kiểm tra tràn).

- Cú pháp: sub \$d, \$s, \$t
- Ví dụ: sub \$t0, \$t1, \$t2 // $\$t0 = \$t1 - \$t2$

13.subu: Trừ giá trị trong thanh ghi thứ hai khỏi giá trị trong thanh ghi thứ nhất mà không kiểm tra tràn.

- Cú pháp: subu \$d, \$s, \$t
- Ví dụ: subu \$t0, \$t1, \$t2 // $\$t0 = \$t1 - \$t2$

14.and: Thực hiện phép AND bit giữa hai thanh ghi và lưu kết quả vào thanh ghi đích.

- Cú pháp: and \$d, \$s, \$t
- Ví dụ: and \$t0, \$t1, \$t2 // $\$t0 = \$t1 \& \$t2$

15.andi: Thực hiện phép AND bit giữa thanh ghi và giá trị hằng số, lưu kết quả vào thanh ghi đích.

- Cú pháp: andi \$d, \$s, imm
- Ví dụ: andi \$t0, \$t1, 10 // $\$t0 = \$t1 \& 10$

16.or: Thực hiện phép OR bit giữa hai thanh ghi và lưu kết quả vào thanh ghi đích.

- Cú pháp: or \$d, \$s, \$t
- Ví dụ: or \$t0, \$t1, \$t2 // $\$t0 = \$t1 | \$t2$

4.nor: Thực hiện phép NOR bit giữa hai thanh ghi và lưu kết quả vào thanh ghi đích.

- Cú pháp: nor \$d, \$s, \$t
- Ví dụ: nor \$t0, \$t1, \$t2 // $\$t0 = \sim(\$t1 | \$t2)$

5.lw: Load word. Tải một từ (word, 32-bit) từ bộ nhớ vào thanh ghi.

- Cú pháp: lw \$t, offset(\$s)
- Ví dụ: lw \$t0, 4(\$t1) // $\$t0 = \text{Mem}[\$t1 + 4]$

6.sw: Store word. Lưu một từ (word, 32-bit) từ thanh ghi vào bộ nhớ.

- Cú pháp: sw \$t, offset(\$s)
- Ví dụ: sw \$t0, 4(\$t1) // $\text{Mem}[\$t1 + 4] = \$t0$

7.slt: Set less than. So sánh hai thanh ghi, nếu $\$s < \t thì $\$d = 1$, ngược lại $\$d = 0$.

- Cú pháp: slt \$d, \$s, \$t

- Ví dụ: `slt $t0, $t1, $t2 // $t0 = ($t1 < $t2)`

8.slti: So sánh thanh ghi với giá trị hằng số, nếu $\$s < \text{imm}$ thì $\$d = 1$, ngược lại $\$d = 0$.

- Cú pháp: `slti $d, $s, imm`
- Ví dụ: `slti $t0, $t1, 10 // $t0 = ($t1 < 10)`

9.sltu: So sánh không dấu giữa hai thanh ghi, nếu $\$s < \t thì $\$d = 1$, ngược lại $\$d = 0$.

- Cú pháp: `sltu $d, $s, $t`
- Ví dụ: `sltu $t0, $t1, $t2 // $t0 = ($t1 < $t2) unsigned`

10.sltiu: So sánh không dấu giữa thanh ghi và giá trị hằng số, nếu $\$s < \text{imm}$ thì $\$d = 1$, ngược lại $\$d = 0$.

- Cú pháp: `sltiu $d, $s, imm`
- Ví dụ: `sltiu $t0, $t1, 10 // $t0 = ($t1 < 10) unsigned`

17.syscall: Gọi hệ thống. Được sử dụng để gọi các dịch vụ của hệ điều hành như nhập/xuất dữ liệu, thoát chương trình.

- Cú pháp: `syscall`
- Ví dụ: Dùng để in ra chuỗi hoặc số, đọc vào số, hoặc thoát chương trình.

2.2 Mô phỏng các chương trình bên dưới và có biết ý nghĩa của chương trình:

VD1:

```
.data
var1: .word 23

.text
start:
lw $t0, var1
li $t1, 5
sw $t1, var1
```

Ý nghĩa: chương trình thay đổi giá trị của `var1` từ 23 thành 5

VD2:

```

.data
array1: .space 12

.text
start:
la $t0, array1
li $t1, 5
sw $t1, 0($t0)
li $t1, 13
sw $t1, 4($t0)
li $t1, -7
sw $t1, 8($t0)

```

Ý nghĩa: chương trình tạo 1 mảng array1 gồm các giá trị [5,13,-7]

VD3:

```

li $v0, 5
syscall

```

Ý nghĩa: chương trình đọc 1 số nguyên từ đầu vào và lưu kết quả vào thanh ghi \$v0

VD4:

```

.data
string1: .ascii "Print this.\n"
.text
main:
li $v0, 4
la $a0, string1
syscall

```

Ý nghĩa: chương trình in ra chuỗi string1 “Print this.\n”

3. Bài tập

3.1 Nhập vào một chuỗi, xuất ra cửa sổ I/O của MARS theo từng yêu cầu sau:

a) Khai báo và xuất ra cửa sổ I/O 2 chuỗi có giá trị như sau:

- Chuỗi 1: Chao ban! Ban la sinh vien nam thu may?

- Chuỗi 2: Hihi, minh la sinh vien nam thu 1 ^-^

```
.data
    chuoi1: .ascii "Chao ban! Ban la sinh vien nam thu may?"
    chuoi2: .ascii "Hihi, minh la sinh vien nam thu 1 ^-^"
.text
    li $v0,4
    la $a0,chuoi1
    syscall

    #xuong dong
    li $v0, 11
    li $a0, 10
    syscall

    li $v0,4
    la $a0,chuoi2
    syscall
```

Chao ban! Ban la sinh vien nam thu may?

Hihi, minh la sinh vien nam thu 1 ^-^

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

b) Biểu diễn của 2 chuỗi trên dưới bộ nhớ là gì?

Data Segment								
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	o a h C	n a b	a B !	a l n	n i s	i v h	n n e	t m a
0x10010020	m u h	\0 ? y a	i h i H	i m ,	l h n	i s a	v h n	n e i
0x10010040	m a n	u h t	- ^ l	\0 \0 \0 ^	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010060	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010080	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x100100a0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x100100c0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0

c) Xuất ra lại đúng chuỗi đã nhập

Ví dụ:

Nhap: Truong Dai hoc Cong nghe Thong tin

Xuất: Truong Dai hoc Cong nghe Thong tin

```

.data
str: .space 100
input: .asciiz "Nhap vao chuoi: "
.text

li $v0,4
la $a0,input
syscall

li $v0,8
la $a0,str # $a0=address(str)
la $a1,100
syscall

li $v0,4
la $a0,str
syscall

Nhap vao chuoi: Truong Dai hoc Cong nghe Thong tin
Truong Dai hoc Cong nghe Thong tin

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

```

d) Nhập vào 2 số nguyên sau đó xuất tổng của 2 số nguyên này

```

.data
msg1: .asciiz "Nhap vao so nguyen thu nhat: "
msg2: .asciiz "Nhap vao so nguyen thu hai: "
msg3: .asciiz "Tong cua 2 so nguyen la: "

.text
li $v0,4
la $a0,msg1
syscall
# dùng syscall trước move khi nhập
li $v0,5
syscall
move $s0,$v0

li $v0,4
la $a0,msg2
syscall

li $v0,5
syscall
move $s1,$v0

```

```

add $s2,$s1,$s0

li $v0,4
la $a0,msg3
syscall

li $v0,1
move $a0,$s2    # Đưa kết quả (tổng) vào $a0 để in
syscall

```

```

Nhap vao so nguyen thu nhat: 17
Nhap vao so nguyen thu hai: 19
Tong cua 2 so nguyen la: 36
-- program is finished running (dropped off bottom) --

```