

## Laboratory Exercise 2

### Instruction Set, Basic Instructions, Directives

Đỗ Hải Dương - 20194528

### Assignment 1: lệnh gán số 16-bit

```
1 #Laboratory Exercise 2, Assignment 1
2 .text
3 addi $s0, $zero, 0x3007 # $s0 = 0 + 0x3007 = 0x3007 ;I-type
4 add $s0, $zero, $0 # $s0 = 0 + 0 = 0 ;R-type
5
```

o Sự thay đổi giá trị của thanh ghi \$s0

- Ban đầu thanh ghi \$s0 có giá trị 0x00000000

\$s0	16	0x00000000
------	----	------------

- Sau khi thực hiện lệnh

```
3 addi $s0, $zero, 0x3007 # $s0 = 0 + 0x3007 = 0x3007 ;I-type
```

⇒ Cộng thanh ghi \$zero với 0x3007 và lưu vào \$s0

- Sau đó thực hiện lệnh

```
4 add $s0, $zero, $0 # $s0 = 0 + 0 = 0 ;R-type
```

⇒ Cộng thanh ghi \$zero với chính nó rồi lưu vào \$s0 nên thanh ghi quay trở lại giá trị 0x00000000.

o Sự thay đổi giá trị của thanh ghi \$pc

- Thanh ghi pc được tăng thêm 4 đơn vị sau mỗi lệnh được thực thi, trở tới lệnh tiếp theo sẽ được thực thi.

+) Ở cửa sổ Text Segment, hãy so sánh mã máy của các lệnh trên với khuôn dạng lệnh để chứng tỏ các lệnh đó đúng như tập lệnh đã qui định

Text Segment			
Bkpt	Address	Code	Basic
<input type="checkbox"/>	0x00400000	0x20103007	addi \$16,\$0,0x00003007
<input type="checkbox"/>	0x00400004	0x00008020	add \$16,\$0,\$0

- Lệnh addi là lệnh I có op = 8.
- Lệnh đầu có mã 0x20103007 ⇔ 001000|00000|10000|0011 0000 0000 01118(op)|0(rs)|16(rt)|3007(imm)
  - ⇒ là lệnh I có rt = 16 => \$s0
  - ⇒ rs = 0 => \$zero
  - ⇒ rt = rs + imm
  - ⇒ đúng với tập lệnh đã quy định
- Lệnh add là lệnh R có op=0, funct = 20
- Lệnh sau có mã 0x00008020
  - ⇔ 000000|00000|00000|10000|00000|1000000(op)

|0(rs)|0(rt)|16(rd)|0(shamt)|20(func)

⇒ đúng với tập lệnh đã quy định

+) Khi sửa lệnh lui thành

```
addi $s0, $zero, 0x2110003d
```

- Vì cộng với hằng số 32 bit nên 16 bit cao sẽ được nạp vào nửa cao của thanh ghi tạm thời \$at, 16 bit thấp sẽ nạp vào nửa thấp của \$at. sau đó dùng lệnh add để nạp \$at vào \$s0.

## Assignment 2: lệnh gán số 32-bit

```
1 #Laboratory Exercise 2, Assignment 2
2 .text
3 lui $s0,0x2110 #put upper half of pattern in $s0
4 ori $s0,$s0,0x003d #put lower half of pattern in $s0
5
```

o Sự thay đổi giá trị của thanh ghi \$s0

- Ban đầu thanh ghi \$s0 có giá trị 0x00000000

\$s0	16	0x00000000
------	----	------------

- Sau khi thực hiện lệnh

```
3 lui $s0,0x2110 #put upper half of pattern in $s0
```

Thanh ghi \$s0 có giá trị

\$s0	16	0x21100000
------	----	------------

- Tiếp tục thực hiện lệnh

```
4 ori $s0,$s0,0x003d #put lower half of pattern in $s0
```

Thanh ghi \$s0 có giá trị

\$s0	16	0x2110003d
------	----	------------

o Sự thay đổi giá trị của thanh ghi \$pc

- Thanh ghi pc được tăng thêm 4 đơn vị sau mỗi lệnh được thực thi, trở tới lệnh tiếp theo sẽ được thực thi.

## Assignment 3: lệnh gán (giả lệnh)

- ```
3 li $s0,0x2110003d #pseudo instruction=2 basic instructions
```

Gán \$s0 cho số 32 bit nên cần chia làm 2 phần, gán 16 bit cao vào biến \$at rồi dùng or để kết hợp với 16 bit thấp rồi gán cho %s0

- `4 li $s1, 0x2 #but if the immediate value is small, one ins`  
gán số 16 bit cho \$s1 nên dùng phép cộng với hằng số không dấu

#### Assignment 4: tính biểu thức $2x + y = ?$

|      |    |            |
|------|----|------------|
| \$s0 | 16 | 0x00000009 |
|------|----|------------|

Kết quả  $2 \times 5 - 1 = 9$

⇒ Đúng

- 16 bit cuối biểu thị số thực cộng thêm  
⇒ lệnh addi thành phần imm chứa 16 bit biểu thị số cộng thêm
- lệnh add giống với khuôn mẫu kiểu lệnh R

#### Assignment 5: Phép nhân

- Khi nhân 2 biến thì làm bình thường còn nhân 1 biến với hằng số thì cần gán hằng số cho thanh ghi tạm \$at rồi mới thực hiện phép nhân
- Kết quả đúng. thanh ghi hi không thay đổi do kết quả nhân dưới 32 bit. kết quả ghi vào thanh ghi lo.

#### Assignment 6: Tạo biến và truy cập biến

- gán 16 bit cao của địa chỉ biến X vào thanh ghi tạm \$at. dùng or kết hợp với 16 bit thấp rồi gán cho \$t8
- mã máy lệnh lui sẽ giống nhau do 16 bit cao giống nhau
- địa chỉ các biến liên tiếp tăng dần 4 đơn vị. imm của lệnh ori cũng như vậy
- lb, sb: tương tự nhưng chỉ ghi 8 bit thấp
- sw: gán giá trị của thanh ghi \$s0 vào biến mà \$t7 trỏ tới
- lw: lấy giá trị của biến mà địa chỉ của nó lưu trong thanh ghi \$t8