**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

**\*\*\*\*\*\*\*\***

A picture containing icon

Description automatically generated

**BÁO CÁO**

**Bài tập thực hành tuần 2**

**Học phần: Thực hành kiến trúc máy tính**

**Giảng viên hướng dẫn:** Lê Bá Vui

**Sinh viên thực hiện:** Phạm Huy Cảnh - 20194490

**Mã lớp:** 130938

**Hà Nội, tháng 4 năm 2022**

# **1. Assignment 1: Lệnh gán số 16-bit**

* ***Sau khi chạy dòng lệnh thứ nhất:*** addi $s0, $zero, 0x3007

Graphical user interface, application, Excel

Description automatically generated

Ta thấy:

+ Thanh ghi $s0 thay đổi giá trị bằng 0x00003007 ($zero + 0x3007)

+ Thanh ghi $pc = 0x00400004 (Đúng bằng địa chỉ của câu lệnh thứ hai)

=> Thanh ghi $pc tự động tăng thêm 4 để chỉ đến địa chỉ của câu lệnh tiếp theo

* ***Sau khi chạy dòng lệnh thứ hai:*** add $s0, $zero, $0

Graphical user interface, text, application

Description automatically generated

Ta thấy:

+ Thanh ghi $s0 thay đổi giá trị bằng 0x00000000 ($zero + $0)

+ Thanh ghi $pc = 0x00400008, Thanh ghi $pc tiếp tục tăng thêm 4 để chỉ đến địa chỉ đến vùng nh tiếp theo

* ***Khuôn dạng lệnh:***

+ Lệnh thứ nhất: addi $16, $0, 0x00003007

Mã lệnh = 0x20103007 = 0010 0000 0001 0000 0011 0000 0000 0111

|  |  |  |  |
| --- | --- | --- | --- |
| **opcode** | **rs** | **rt** | **immediate** |
| 6 bits | 5 bits | 5 bits | 16 bits |
| 001000 | 00000 | 10000 | 0011000000000111 |
| 8 | 0 | 16 | 0x3007 |

=> Câu lệnh thứ nhất đã chuẩn so với khuôn dạng lệnh I

+ Lệnh thứ nhất: add $16, $0, $0

Mã lệnh = 0x00008020 = 0000 0000 0000 0000 1000 0000 0010 0000

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **opcode** | **rs** | **rt** | **rd** | **shamt** | **funct** |
| 6 bits | 5 bits | 5 bits | 5 bits | 5 bits | 6 bits |
| 000000 | 00000 | 00000 | 10000 | 00000 | 100000 |
| 0 | 0 | 0 | 16 | 0 | 32 |

=> Câu lệnh thứ hai đã chuẩn so với khuôn dạng lệnh R

* ***Sửa câu lệnh thành*** addi $s0, $zero, 0x2110003d

Graphical user interface, application, table, Excel

Description automatically generated

+ Hiện tượng xảy ra: Câu lệnh addi $s0, $zero, 0x2110003d được thực hiện bằng 3 câu lệnh:

lui $1, 0x00002110

ori $1, $1, 0x0000003d

add $16, $0, $1

+ Giải thích: Vì câu lệnh một câu lệnh có độ dài 32 bits, trong khi đó ở câu lệnh trên phần immediate lại chiếm hết 32 bits của câu lệnh, nên để tính toán được nó chia phần immediate thành 2 phần là: 16 bits cao và 16 bits thấp. Sau đó ghi từng phần vào một thanh ghi 32 bits (ở đây là thanh ghi $1). Lệnh lui sẽ ghi 2110(16) vào 16 bits cao của thanh ghi $1, Lệnh ori sẽ ghi 003d(16) vào 16 bits thấp của thanh ghi $s1. Khi này thanh ghi $1 = 0x2110003d, cuối cùng thực hiện lệnh add hai thanh ghi $1 và $0 kết quả được ghi vào thanh ghi $16 ($s0)

**2. Assignment 2: Lệnh gán số 32-bit**

* ***Sau khi chạy dòng lệnh thứ nhất:*** lui $s0, 0x2110

Graphical user interface, application

Description automatically generated

Ta thấy:

+ Thanh ghi $s0 thay đổi giá trị bằng 0x21100000 (Ghi giá trị 2110(16) vào 2 bytes cao của thanh ghi $s0)

+ Thanh ghi $pc = 0x00400004 (Đúng bằng địa chỉ của câu lệnh thứ hai)

* ***Sau khi chạy dòng lệnh thứ hai:*** ori $s0, 0x003d

*Graphical user interface, application, Word

Description automatically generated*

Ta thấy:

+ Thanh ghi $s0 thay đổi giá trị bằng 0x2110003d (Ghi giá trị 003d(16) vào 2 bytes thấp của thanh ghi $s0)

+ Thanh ghi $pc = 0x00400008, Thanh ghi $pc tiếp tục tăng thêm 4 để chỉ đến vùng nhớ tiếp theo

* ***Quan sát các byte trong vùng lệnh .text:***

***Graphical user interface, application, table, Excel

Description automatically generated***

Nhận xét: Byte đầu tiên ở vùng lệnh trùng với cột Address trong cửa số Text Segment

**3. Assignment 3: Lệnh gán (giả lệnh)**

***Sau khi biến dịch:***

**Graphical user interface, application, table, Excel

Description automatically generated**

* ***Dòng lệnh thứ nhất:*** li $s0,0x2110003d

+ Câu lệnh này được thực hiện bởi hai câu lệnh:

lui $1, 0x00002110

ori $16, $1, 0x0000003d

+ Giải thích: Phần immediate đã vượt quá mức 16 bits nên phải tách thành hai câu lệnh như trên để ghi lần lượt 16 bits cao và 16 bits thấp vào thanh ghi $s0. Bằng cách đầu tiên sử dụng lệnh lui $1, 0x00002110 để ghi 2110 (16) vào 16 bits cao của thanh ghi $1, khi đó thanh ghi $1 = 0x21100000; sau đó sử dụng lệnh ori $16, $1, 0x0000003d để OR thanh ghi $16 với thanh ghi $1, khi đó thanh ghi $16 = 0x2110003d. Vậy kết quả ta gán được giá trị 0x2110003d vào thanh ghi $s0

* ***Dòng lệnh thứ hai:*** li $s1,0x2

+ Câu lệnh này được thực hiện bởi một câu lệnh:

addi $17, $0, 0x00000002

+ Giải thích: Phần immediate nằm trong mức cho phép nên câu lệnh trên chỉ cần thực hiện bằng một câu lệnh addi $17, $0, 0x00000002, khi đó ta được kết quả thanh ghi $s1 được gán bằng 0x2

**4. Assignment 4: Tính biểu thức 2x+y=?**

* ***Sau khi chạy dòng lệnh thứ nhất:*** addi $t1, $zero, 5

***Graphical user interface, application, Word

Description automatically generated***

Ta thấy: Thanh ghi $t1 được gán giá trị bằng 0x00000005 (5(10))

* ***Sau khi chạy dòng lệnh thứ hai:*** addi $t2, $zero, -1

***Graphical user interface, application, Word

Description automatically generated***

Ta thấy: Thanh ghi $t2 được gán giá trị bằng 0xffffffff (-1(10))

* ***Sau khi chạy dòng lệnh thứ ba:*** add $s0, $t1, $t1

***Graphical user interface, text, application

Description automatically generated***

Ta thấy: Thanh ghi $s0 có giá trị bằng 0x0000000a (10(10))

* ***Sau khi chạy dòng lệnh thứ tư:*** add $s0, $s0, $t2

**Graphical user interface, application, Excel

Description automatically generated**

Ta thấy: Thanh ghi $s0 có giá trị bằng 0x00000009 (9(10))

**=> Kết quả đúng**

* ***Kiểm nghiệm khuôn dạng lệnh:***

+ Lệnh addi

Mã máy: 0x20090005 = 0010 0000 0000 1001 0000 0000 0000 0101

Hợp ngữ: addi $9, $0, 0x00000005

|  |  |  |  |
| --- | --- | --- | --- |
| **opcode** | **rs** | **rt** | **immediate** |
| 6 bits | 5 bits | 5 bits | 16 bits |
| 001000 | 00000 | 01001 | 0000000000000101 |
| 8 | 0 | 9 | 5 |

Mã máy: 0x200affff = 0010 0000 0000 1010 1111 1111 1111 1111

Hợp ngữ: addi $10, $0, 0xffffffff

|  |  |  |  |
| --- | --- | --- | --- |
| **opcode** | **rs** | **rt** | **immediate** |
| 6 bits | 5 bits | 5 bits | 16 bits |
| 001000 | 00000 | 01010 | 1111111111111111 |
| 8 | 0 | 10 | -1 |

=> Cả hai câu lệnh addi đều đúng với khuôn dạng lệnh I

+ Lệnh add

Mã máy: 0x01298020 = 0000 0001 0010 1001 1000 0000 0010 0000

Hợp ngữ: add $16, $9, $9

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **opcode** | **rs** | **rt** | **rd** | **shamt** | **funct** |
| 6 bits | 5 bits | 5 bits | 5 bits | 5 bits | 6 bits |
| 000000 | 01001 | 01001 | 10000 | 00000 | 100000 |
| 0 | 9 | 9 | 16 | 0 | 32 |

Mã máy: 0x020a8020 = 0000 0010 0000 1010 1000 0000 0010 0000

Hợp ngữ: add $16, $16, $10

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **opcode** | **rs** | **rt** | **rd** | **shamt** | **funct** |
| 6 bits | 5 bits | 5 bits | 5 bits | 5 bits | 6 bits |
| 000000 | 10000 | 01010 | 10000 | 00000 | 100000 |
| 0 | 16 | 10 | 16 | 0 | 32 |

=> Cả hai câu lệnh add đều đúng với khuôn dạng lệnh R

**5. Assignment 5: Phép nhân**

* ***Biên dịch và quan sát các lệnh mã máy trong cửa sổ Text Segment:***

**Graphical user interface, application, Word

Description automatically generated**

**Nhận xét:** Sự khác thường khi thực hiện hai câu lệnh mul

+ Khi thực hiện câu lệnh mul $s0, $t1, $t2 thì máy thực hiện luôn việc nhân giữa hai thanh ghi $t1 và $t2, kết quả được ghi vào thanh ghi $s0

+ Khi thực hiện câu lệnh mul $s0, $s0, 3. Vì không có thanh ghi nào lưu trữ riêng giá trị 3 nên để thực hiện được câu lệnh này thì được máy thực hiện bằng hai câu lệnh. Đầu tiên là câu lệnh addi $1, $0, 0x00000003, bước này là để gán giá trị 3 cho thanh ghi $1. Sau đó thực hiện tiếp lệnh nhân mul $16, $16, $1

* ***Sau khi chạy dòng lệnh:*** addi $t1, $zero, 4

Graphical user interface, application, Word

Description automatically generated

Ta thấy: Thanh ghi $t1 = 0x00000004 (4(10))

* ***Sau khi chạy dòng lệnh:*** addi $t2, $zero, 5

Graphical user interface, application, Word

Description automatically generated

Ta thấy: Thanh ghi $t2 = 0x00000005 (5(10))

* ***Sau khi chạy dòng lệnh:*** mul $s0, $t1, $t2

Graphical user interface, application, Word

Description automatically generated

Ta thấy: + Thanh ghi $s0 = 0x00000014 (20(10))

+ Thanh ghi lo = 0x00000014 (20(10))

* ***Sau khi chạy dòng lệnh:*** mul $s0, $s0, 3

Graphical user interface, application, Word

Description automatically generated

Ta thấy: + Đầu tiên thanh ghi $1 được gán bằng 0x00000003 (3(10))

+ Sau lệnh nhân ta có $s0 = 0x0000003c (60(10)) và

lo = 0x0000003c (60(10))

* ***Sau khi chạy dòng lệnh:*** mflo $s1

**Graphical user interface, application, Word

Description automatically generated**

Ta thấy: Giá trị của thanh ghi lo được ghi vào trong thanh ghi

$s1 = 0x0000003c (60(10))

**=> Kết quả đúng**

**6. Assignment 6: Tạo biến và truy cập biến**

* ***Biên dịch và quan sát các lệnh mã máy trong cửa sổ Text Segment:***

***Graphical user interface, application, email

Description automatically generated***

Ta thấy: Lệnh la được biên dịch bằng hai lệnh lui và ori. Vì địa chỉ của X, Y, Z được khai báo với chỉ thị là .word nên địa chỉ của X, Y, Z là số 32 bits nên cần thực hiện bằng việc thực hiện hai câu lệnh lui và ori

* ***Ở cửa sổ Label và quan sát địa chỉ của X, Y, Z:***

***Graphical user interface, application, table

Description automatically generated***

Ta thấy: Địa chỉ của X, Y, Z được chia làm hai phần: 2 bytes cao và 2 bytes thấp. Tương ứng với mỗi lệnh la thì lệnh lui sẽ ghi 2 bytes cao của địa chỉ vào thanh ghi $1 và lệnh ori sẽ OR thanh ghi $1 với 2 bytes thấp của địa chỉ.

Table

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated

* ***Sau khi chạy lần lượt các dòng lệnh:***

Ta thấy: + Sau khi lấy địa chỉ của các biến X, Y, Z và ghi địa chỉ đó vào các thanh ghi $t8, $t9, $t7 bằng lệnh la

la $t8, X

la $t9, Y

+ Tiếp tục lấy ra giá trị 32 bit (word) nằm trong các ô nhớ có địa chỉ vừa ghi được vào thanh ghi $t8, $t9 bằng lệnh lw vào các thanh ghi $t1, $t2

lw $t1, 0($t8)

lw $t2, 0($t9)

+ Sau khi thực hiện các câu lệnh add, kết quả đang được ghi trong trong thanh ghi $s0. Để ghi giá trị 32 bit (word) trong thanh ghi $s0 vào ô nhớ có địa chỉ được ghi trong thanh ghi $t7 thì ta sử dụng lệnh sw

sw $s0, 0($t7)

* Lệnh lb, sb

+ lb - Để lấy ra dữ liệu kiểu byte (8 bit) tại ô nhớ thông qua địa chỉ trỏ đến ô nhớ đó ra thanh ghi

+ sb - Để ghi dữ liệu kiểu byte (8 bit, 8 bit này được lưu vào 8 bit thấp của ô nhớ) vào ô nhớ thông qua địa chỉ trỏ đến ô nhớ đó