**Thực hành kiến trúc máy tính tuần 26**

**Full name: Lê Quang Khải**

**Student ID: 20225638**

Assignment 1

.text

addi $s0, $zero, 0x3007

add $s0, $zero, $0

**Sự thay đổi giá trị của thanh ghi $s0 và thanh ghi pc**

-Lần thứ nhất:

**A table with numbers and a number in it

Description automatically generated**

-Lần thứ hai: **A table with numbers and a green line

Description automatically generated**

-Lần thứ ba:

**A table with numbers and a green line

Description automatically generated**

**So sánh:**

**addi $s0, $zero, 0x3007**

addi $16, $0, 0x3007

opcode: 8 => 001000

rs: $0 => 00000

rt: $16 => 10000

imm: 0x3007: 0011 0000 0000 0111

Answer: 0010 0000 0001 0000 0011 0000 0000 0111

0x20103007

**add $s0, $zero, $0**

add $16, $0, $0

opcode: 0 => 000000

rs: $0 => 00000

rt: $0 => 00000

rd: $16 => 10000

sh: 00000

fn: 100000

Answer: 0000 0000 0000 0000 1000 0000 0010 0000

0x00008020

A screenshot of a computer

Description automatically generated

* Đúng như tập lệnh

**Sửa lại lệnh addi như bên dưới. Chuyện gì xảy ra sau đó. Hãy giải thích**

addi $s0, $zero, 0x2110003d

A screenshot of a computer

Description automatically generated

Mã code đã chuyển số 32 bit thành 2 nửa 16 bit và dùng lệnh lui và ori để gán nửa cao với nửa thấp sau đó cộng như bình thường.

Assignment 2

.text

lui $s0,0x2110

ori $s0,$s0,0x003d

**Các bước lần lượt như sau:**

**A table with numbers and a green border

Description automatically generatedA table with numbers and a number in it

Description automatically generatedA table with numbers and a green line

Description automatically generated**

Các byte đầu tiên ở vùng lệnh trùng với cột nào trong cửa sổ Text Segment: Trùng với cột “Code”

A screen shot of a computer code

Description automatically generatedA screenshot of a computer

Description automatically generated



Assignment 3

A screenshot of a computer

Description automatically generated

* Lệnh li là một lệnh mã giả được sử dụng để tải một giá trị ngay lập tức vào một thanh ghi. Lệnh này thực sự được thay thế bởi một hoặc nhiều lệnh thực tương đương.  
  Khi biên dịch và quan sát các lệnh mã máy trong cửa sổ Text Segment, điều bất thường có thể là sự khác biệt về số lượng lệnh tạo ra từ lệnh li tùy thuộc vào giá trị ngay lập tức cung cấp.
* Như ở trên ta thấy cùng 1 lệnh “li” nhưng 2 câu lệnh thực sự khác nhau là lui,ori và addiu

Assignment 4

**Sự thay đổi:**

**A table with numbers and a green line

Description automatically generated**

**A screenshot of a table

Description automatically generated**

****

**addi $t2, $zero, -1:**

0x200affff: 0010 0000 0000 1010 1111 1111 1111 1111

Opcode: 001000 => $8 => addi

rs: 00000 => $0 =>$zero

rt: 01010 => $10 => $t2

im: 1111 1111 1111 1111 => 0xffffffff => -1

Answer: addi $t2, $zero, -1:

**Đổi sang hệ 2:**

**add $s0, $t1, $t1 =** add $16, $9, $9

opcode: 0 => 000000

rs: $9 => 01001

rt: $9 => 01001

rd: $16 => 10000

sh: 00000

fn: 100000

Answer: 0000 0001 0010 1001 1000 0000 0010 0000

* Chính xác như khuôn mẫu của R-format

Assignment 5

Sự thay đổi:

A table with numbers and letters

Description automatically generatedA screenshot of a table

Description automatically generated A table with numbers and letters

Description automatically generated

**Sau khi chạy chương trình thì kết quả ra đúng.**

Điều bất thường ở đây là dòng lệnh mul $s0, $s0, 3 thành 2 lệnh addi và lệnh mul do lệnh mul không thể nhân 1 thanh ghi trực tiếp với 1 hằng số vì thế nó cần phải gán giá trị hằng số đó vào 1 thanh ghi tạm là thanh ghi $at rồi sau đó mới nhân 2 thanh ghi như bình thường.



Assignment 6

Cụ thể, lệnh la được biên dịch như sau:

Để tải địa chỉ của biến vào một thanh ghi, trình biên dịch MIPS sẽ sử dụng lệnh lui (load upper immediate) để tải 16 bit cao nhất của địa chỉ vào thanh ghi đích. Sau đó, trình biên dịch sẽ sử dụng lệnh ori (or immediate) để "hoàn thành" địa chỉ bằng cách thêm 16 bit thấp nhất vào thanh ghi đích. Kết quả là địa chỉ của biến được tải vào thanh ghi.

**So sánh địa chỉ của X, Y, Z với hằng số khi biên dịch lệnh la thành mã máy:** Chúng khác nhau bởi vì nó lệnh lui và or nó lưu địa chỉ còn hằng số là giá trị

**Sự thay đổi của các thanh ghi:**

Khi dùng lệnh la, lw

**A screenshot of a spreadsheet

Description automatically generated**

A screenshot of a computer

Description automatically generated

Sau khi cộng:

A screenshot of a table

Description automatically generated

A screenshot of a spreadsheet

Description automatically generated

Sau khi gán cho Z:

A table with numbers and a green line

Description automatically generated

**Xác định vai trò của lệnh lw và sw:**

**-** lw (Load Word): Lệnh lw được sử dụng để đọc một từ dữ liệu từ bộ nhớ vào một thanh ghi

- sw (Store Word): Lệnh sw được sử dụng để ghi một từ dữ liệu từ một thanh ghi vào bộ nhớ.

- **Lệnh lb và sb** cũng là hai lệnh cơ bản trong ngôn ngữ lập trình Assembly, nhưng chúng thực hiện truy cập dữ liệu ở mức byte thay vì từ như lw và sw. Đây là cách các lệnh này hoạt động:

lb (Load Byte): Lệnh lb được sử dụng để đọc một byte dữ liệu từ bộ nhớ vào một thanh ghi (register)

sb (Store Byte): Lệnh sb được sử dụng để ghi một byte dữ liệu từ một thanh ghi vào bộ nhớ. Lệnh này yêu cầu hai tham số: địa chỉ của byte dữ liệu trong bộ nhớ và thanh ghi chứa giá trị dữ liệu.