

Họ tên: Nguyễn Văn Hiếu-20225717

Assignment 1: lệnh gán số 16 bits

-Ở mỗi lệnh, quan sát cửa sổ Registers ta thấy:

Sau lệnh 1:

\$s0	16	0x00003007
pc		0x00400004

Sau lệnh 2:

\$s0	16	0x00000000
pc		0x00400008

-So sánh:

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

I type:

Opcode: 8 => 001000

Rt: \$16=> 10000

Rs : \$0=> 00000

Imm: 0x00003007=> 0011 0000 0000 0111

0010 0000 0001 0000 0011 0000 0000 0111

0x20103007

Lệnh thứ hai: add \$16, \$0, \$0

I type:

Opcode: 0=> 000000

Rd: 16=> 10000

Rs: 0 => 00000

Rt: 0 => 00000

Sh: 0=> 00000

fn: 32=>100000

0000 0000 0000 0000 1000 0000 0001 0000

0x00008020

-Khi thay đổi câu lệnh:

Đầu tiên giá trị của thanh \$at thay đổi từ 0x00000000 => 0x021100000 => 0x2110003d. Sau đó thanh \$s0 thay đổi từ 0x00000000=>0x2110003d=>0x00000000.

Giải thích:

0x2110003d không thể trực tiếp ở 32 bit mà được tách làm 2 phần:

16-bit 2110 được thực hiện gán bởi lệnh lui \$1,0x0002110. Kết quả được lưu ở thanh ghi \$at: 0x21100000

16-bit 003d được thực hiện gán bởi lệnh ori \$1,\$1,0x000003d. Kết quả được cập nhật ở thanh ghi \$at: 0x2110003d

Sau đó kết quả được gán về \$0 từ \$at => \$s0: 0x2110003d

Kết quả sau khi thực hiện dòng lệnh add \$s0, \$zero, \$0 => \$s0: 0x00000000

Assignment 2: lệnh gán số 32 bits

Ở mỗi lệnh, quan sát cửa sổ Registers ta thấy:

Sau lệnh 1:

\$s0	16	0x21100000
pc		0x00400004

Sau lệnh 2:

\$s0	16	0x2110003d
pc		0x00400008

Quan sát các byte trong vùng lệnh .text: Các byte đầu tiên trùng với cột Code trong cửa sổ Text Segment.

Address	Value (+0)	Value (+4)
0x00400000	0x3c102110	0x3610003d

Code
0x3c102110
0x3610003d

Assignment 3: lệnh gán (giả lệnh)

Text Segment					
Bkpt	Address	Code	Basic	Source	
<input type="checkbox"/>	0x00400000	0x3c102110	lui \$1,0x00002110	2:	li \$s0, 0x2110003d
<input type="checkbox"/>	0x00400004	0x3430003d	ori \$16,\$1,0x0000003d		
<input type="checkbox"/>	0x00400008	0x24110002	addiu \$17,\$0,0x00000002	3:	li \$s1, 0x2

Khi quan sát các lệnh mã máy trong cửa sổ Text Segment: Lệnh li thứ nhất được tách thành 2 lệnh lui và ori do số 2110003d lớn hơn 16 bits, lệnh li thứ hai không tách do số 0x2 nhỏ hơn 16 bits.

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

Quan sát cửa sổ Registers ta thấy:

Sau lệnh 1: gán giá trị 5 vào thanh ghi \$t1

\$t1	9	0x00000005
pc		0x00400004

Sau lệnh 2: gán giá trị -1 vào thanh &t2 (trừ đi 1 chính là cộng số bù 2 của 1).

\$t2	10	0xffffffff
pc		0x00400008

Sau lệnh 3: thực hiện câu lệnh $s0 = t1 + t1 = X + X$.

\$s0	16	0x0000000a
pc		0x0040000c

Sau lệnh 4: thực hiện câu lệnh $s0 = s0 + t2 = 2X + Y$.

\$s0	16	0x00000009
pc		0x00400010

Sau khi kết chương trình, ta thu được kết quả trên thanh ghi s0, kết quả thu được là kết quả đúng.

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

Điểm tương đồng giữa hợp ngữ và mã máy:

Code	Basic
0x20090005	addi \$9,\$0,0x00000005
0x200affff	addi \$10,\$0,0xffffffff

Kiểm nghiệm lệnh addi với khuôn mẫu lệnh I:

Lệnh thứ nhất: addi \$9, \$0, 0x00000005

Opcode: 8 => 001000

rt: 9 => 01001

rs: 0 => 00000

imm: 0x00000005 => 0000 0000 0000 0101

=> 0010 0000 0000 1001 0000 0000 0000 0101

Lệnh thứ hai: addi \$10, \$0, 0xffffffff

Opcode: 8=> 001000

rt: 10 => 01010

rs: 0 => 00000

imm: 0xffffffff => 1111 1111 1111 1111

=> 0010 0000 0000 1001 1111 1111 1111 1111

Kiểm nghiệm lệnh add với khuôn mẫu R:

Lệnh thứ nhất: add \$16, \$9, \$9

Opcode: 0 => 000000

rd: 16 => 10000

rs: 9 => 01001

rt: 9 => 01001

sh: 0 => 00000

fn: 32 => 100000

=> 0001 0010 1001 1000 0000 0010 0000

Lệnh thứ hai: add \$16, \$16, \$10.

Opcode: 0 => 000000

rd: 16 => 10000

rs: 16 => 10000

rt: 10 => 01010

sh: 0 => 00000

fn: 32 => 100000

=> 0010 0000 1010 1000 0000 0010 0000

Assignmet 5: Phép nhân

Điều bất thường:

Text Segment				
Bkpt	Address	Code	Basic	Source
<input type="checkbox"/>	0x00400000	0x20090004	addi \$9,\$0,0x00000004	3: addi \$t1, \$zero, 4
<input type="checkbox"/>	0x00400004	0x200a0005	addi \$10,\$0,0x00000005	4: addi \$t2, \$zero, 5
<input type="checkbox"/>	0x00400008	0x712a8002	mul \$16,\$9,\$10	6: mul \$s0, \$t1, \$t2
<input type="checkbox"/>	0x0040000c	0x20010003	addi \$1,\$0,0x00000003	7: mul \$s0, \$s0, 3
<input type="checkbox"/>	0x00400010	0x72018002	mul \$16,\$16,\$1	
<input type="checkbox"/>	0x00400014	0x0008812	mflo \$17	9: mflo \$s1

Ta thấy rằng câu lệnh mul \$s0, \$s0, 3 được tách thành hai câu lệnh trong cửa sổ Text Segment. Điều đó xảy ra do câu lệnh mul chỉ phân hai ô nhớ với nhau nên thanh ghi \$at được dùng để lưu số 3.

Thay đổi giá trị thanh ghi:

Sau lệnh 1:

\$t1	9	0x00000004
pc		0x00400004

Sau lệnh 2:

\$t2	10	0x00000005
pc		0x00400008

Sau lệnh 3:

\$s0	16	0x00000014
pc		0x0040000c
lo		0x00000014

Sau lệnh 4:

\$at	1	0x00000003
pc		0x00400010

Sau lệnh 5:

\$s0	16	0x0000003c
pc		0x00400014
lo		0x0000003c

=> Chương trình cho ra kết quả đúng.

Assignment 6: Tạo biến và truy cập biến

Lệnh **la** được biên dịch thành hai lệnh **lui** và **ori**.

Ở cửa sổ Labels, địa chỉ của ba biến X,Y,Z giống với hằng số khi biên dịch lệnh **la** thành mã máy.

mips1.asm	
X	0x10010000
Y	0x10010004
Z	0x10010008

Sự thay đổi các thanh ghi:

Sau lệnh 1:

\$at	1	0x10010000
pc		0x00400004

