Thực hành Kiến trúc máy tính

Week 2

Vũ Ngọc Đức – 20225816

Assignment 1

Các lần chạy:

Lần 1:

$s0 => 0x00003007

pc => 0x00400004

Lần 2:

$s0 => 0x00000000

pc => 0x00400008

So sánh với mã máy:

Lệnh 1: addi $s0, $zero, 0x3007

op: 8 => 001000

rs: $0 => 00000

rt: $16 => 10000

imm: 0x3007 => 0011 0000 0000 0111

ans: 0010 0000 0001 0000 0011 0000 0000 0111 = 0x20103007

Lệnh 2: add $s0, $zero, $0

op: 0 => 000000

rs: 0 => 00000

rt: 0 => 00000

rd: 16 => 10000

shamt: 0 => 00000

func: 32 => 100000

ans: 0000 0000 0000 0000 1000 0000 0010 0000 = 0x00008020

Sửa lại lệnh addi: addi $s0, $zero, 0x2110003d

lui $1, 0x00002110 => lưu nửa đầu imm vào at = 0x21100000

ori $1, $1, 0x0000003d => cộng nửa sau imm vào at = 0x2110003d

add $16, $0, $1 => $s0 = at + 0 = 0x2110003d

add $16, $0, $0 => $s0 = 0 + 0 = 0

Assignment 2

Các lần chạy:

Lần 1:

$s0 => 0x21100000

pc => 0x00400004

Lần 2:

$s0 => 0x2110003d

pc => 0x00400008

Address 0x00400000

Value(+0) trùng cột code lệnh 1

Value(+4) trùng cột code lệnh 2

Assignment 3:

A screenshot of a computer

Description automatically generated

Dễ thấy lệnh li bị thay thế bằng câu lệnh khác 0x2110003d, do là 32 bit và là format I nên hệ thống phải đổi sang lui và ori để thực hiện câu lệnh, còn 0x2 là 16 bit nhưng do li không là lệnh thực hiện trực tiếp lên hệ thống phải đổi sang câu lệnh khác để có thể thực hiện.

Assignment 4:

A screenshot of a computer

Description automatically generated

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Trạng thái | $t1 | $t2 | $s0 | Pc |
| Ban đầu | 0x00000000 | 0x00000000 | 0x00000000 | 0x00400000 |
| Chạy addi 1 | 0x00000005 | 0x00000000 | 0x00000000 | 0x00400004 |
| Chạy addi 2 | 0x00000005 | 0xffffffff | 0x00000000 | 0x00400008 |
| Chạy add 1 | 0x00000005 | 0xffffffff | 0x0000000a | 0x0040000c |
| Chạy add 2 | 0x00000005 | 0xffffffff | 0x00000009 | 0x00400010 |

Phép tính được thực hiện: 2\*5 + (-1) = 9 đúng với kết quả

* Ở lệnh addi ta có dạng là addi $rt , $rs, immi tương ứng với I format
* Ta có 0x2 tương ứng với opcode là 2 là lệnh addi còn vị trí thứ 3 là 9 tương ứng với rt còn vị trí thứ 4 là 0 tương ứng với rs còn 4 bit cuối là 0005 tương ứng với immi
* Lệnh add thứ nhất: add $s0, $t1, $t1 ( add $16, $9, $9 )

op : 0 => 000000

rd : 16 => 10000

rs : 9 => 10001

rt : 9 => 10001

shamt : 0 => 00000

funt : 32 => 100000

ans: 0000 0001 0010 1001 1000 0000 0010 0000 = 0x01298020

* Lệnh add thứ hai: add $s0, $s0, $t2 ( add $16, $16, $10 )

op : 0 => 000000

rd : 16 => 10000

rs : 16 => 10000

rt : 10 => 01010

shamt : 0 => 00000

funt : 32 => 100000

ans: 0000 0010 0000 1010 1000 0000 0010 0000 = 0x020a8020

* Như vậy lệnh add phù hợp với R format

Assignment 5:

Ảnh có chứa văn bản, phần mềm, số, hàng

Mô tả được tạo tự động

* Lệnh mul với 1 số bị thay thế bằng 1 lệnh khác, do lệnh mul là R-format, ta lại truyền vào là I-format, do đó chương trình phải them 1 lệnh khác để chuyển 0x00000003 thành 1 beiens rồi mới thực hiện

Ảnh có chứa văn bản, ảnh chụp màn hình, số, hàng

Mô tả được tạo tự động

|  |  |  |
| --- | --- | --- |
| Trạng thái | hi | lo |
| Ban đầu | 0x00000000 | 0x00000000 |
| Chạy add 1 | 0x00000000 | 0x00000000 |
| Chạy add 2 | 0x00000000 | 0x00000000 |
| Chạy mul 1 | 0x00000000 | 0x00000014 |
| Chạy mul 2 | 0x00000000 | 0x0000003c |

* Kết quả cho ra đúng với phép toán: 3\*4\*5 = 60 (0x0000003c)

Assignment 6  
  
-Lệnh la dùng để lưu địa chỉ của biến 16 bit. Nhưng địa chỉ của biến trong bài là 32 bit nên sẽ thực hiện 2 lệnh lui và ori để lưu địa chỉ biến.

- Sau khi chạy lệnh la: X tại Value (+0), Y tại Value (+4), Z tại Value (+8)



- Sự thay đổi giá trị thanh ghi:

Chạy la 1: $t8 => 0x10010000

Chạy la 2: $t9 => 0x10010004

Chạy lw 1: $t1 => 0x00000005

Chạy lw 2: $t2 => 0xffffffff

Chạy add 1: $s0 => 0x0000000a

Chạy add 2: $s0 => 0x00000009

Chạy la: $t7 => 0x10010008

- Lệnh lw: dùng để lưu giá trị trên địa chỉ vào địa chỉ còn lại

- Lệnh sw: dùng để lưu giá trị trên thanh ghi vào 1 địa chỉ