Ex1:

*- Chương trình:*

Graphical user interface, application

Description automatically generated

Graphical user interface, table

Description automatically generated

*- Chạy từng dòng lệnh:*

Step 1:

Table

Description automatically generated

$s0: 0x00000000

pc: 0x00400000

Step 2:

Table

Description automatically generated

*- Sự thay đổi giá trị của các thanh ghi:*

+ Sự thay đổi giá trị của thanh ghi $s0 khi chạy từng lệnh :

0x00000000 → 0x00003007 → 0x00000000

+ Sự thay đổi giá trị của thanh ghi $pc khi chạy từng lệnh:

0x00400000 → 0x00400004 → 0x00400008

*- So sánh mã máy của các lệnh:*

addi $s0, $zero, 0x3007

op: 8

rs: $zero

rt: $s0

imm: 0x3007

0010 0000 0001 0000 0011 0000 0000 0111 => 0x20103007

add $s0, $zero, $0

op:0

rd: $s0

rs: $zero

rt: $0

shamt: 0

funct:32

0000 0000 0000 0000 1000 0000 0010 0000 => 0x00008020

*- Sửa lại lệnh:*

Graphical user interface, application

Description automatically generated

Graphical user interface, table

Description automatically generated

Step 1:

Table

Description automatically generatedStep 2:

Table

Description automatically generatedStep 3:

Table

Description automatically generatedStep 4:

Table

Description automatically generated

Lệnh addi: 0x211003d được chia thành 0x00002110 và 0x0000003d (Vì hằng số ở đây là 32 bit, máy phải tách ra thành 2 lệnh lui và ori để lần lượt chạy).

Ex2.

*- Chương trình:*

Graphical user interface

Description automatically generated

Graphical user interface, table

Description automatically generated

*- Chạy từng dòng lệnh:*

Step 1:

Table

Description automatically generated

Step 2:

Table

Description automatically generated

*- Sự thay đổi của giá trị của các thanh ghi:*

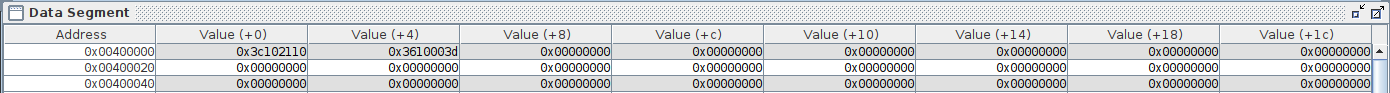
+ Sự thay đổi giá trị của thanh ghi $s0 khi chạy từng lệnh :

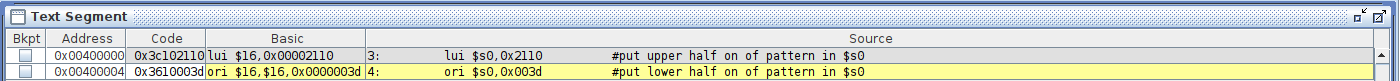
0x00000000 => 0x21100000 => 0x2110003d

+ Sự thay đổi giá trị của thanh ghi $pc khi chạy từng lệnh :

0x00400000 => 0x00400004 => 0x00400008

+ Ở cửa sổ Data Segment của byte đầu tiên trùng với cột code trong Text Segment.





Ex3.

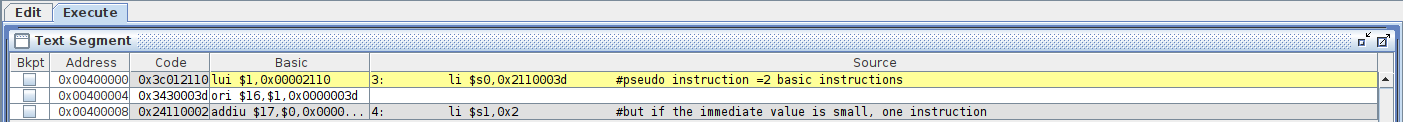
*- Chương trình:*

Graphical user interface

Description automatically generated

Graphical user interface, table

Description automatically generated



*- Giải thích:*

+ Lệnh li là giả lệnh, được phân tách thành 2 lệnh là lui và ori.

+ Vì giá trị 0x2110003d là hằng số có giá trị 32 bit nên phải tách thành 2 phần 16 bit và dùng lệnh lui và ori để lấy địa chỉ.

+ lui $1, 0x00002110: thực hiện gán địa chỉ 0x00002110 vào thanh ghi $1.

+ ori $16, $1, 0x0000003d: thực hiện gán địa chỉ 0x0000003d vào thanh ghi $16.

+ Lệnh addiu $17,$0,0x00000002 gán giá trị vào $17.

Ex4:

*- Chương trình:*

Graphical user interface

Description automatically generated

Graphical user interface, table

Description automatically generated

*- Chạy từng dòng lệnh:*

Step 1:

Table

Description automatically generated

Step 2:

Table

Description automatically generated

Step 3:

Table

Description automatically generated

Step 4:

Table

Description automatically generated

Kết quả bằng 9 => chạy đúng kết quả.

- Sự thay đổi của các thanh ghi:

$t1: 0x00000000 => 0x00000005

$t2: 0x00000000 => 0xffffffff

$s0: 0x00000000 => 0x0000000a => 0x00000009

- Kiểm nghiệm với khuôn mẫu của kiểu lệnh I:

addi $9, $0, 0x00000005

op: 8

rs: $0

rt: $9

imm: 0x00000005

0010 0000 0000 1001 0000 0000 0000 0101 => 0x20090005

addi $10, $0, 0xffffffff

op: 8

rs: $0

rt: $10

imm: 0xffffffff

0010 0000 0000 1010 1111 1111 1111 1111 => 0x200affff

- Kiểm nghiệm với khuôn mẫu của kiểu lệnh R:

add $16, $9, $9

op: 0

rs: $9

rt: $9

rd: $16

sh: 0

fn: 32

0000 0001 0010 1001 1000 0000 0010 0000 => 0x01298020

add $16, $16, $10

op: 0

rs: $16

rt: $10

rd: $16

sh: 0

fn: 32

0000 0010 0000 1010 1000 0000 0010 0000 => 0x020a8020

Ex5:

*- Chương trình:*

Graphical user interface, table

Description automatically generated

Graphical user interface, table

Description automatically generated

*- Giải thích điều bất thường:*

+ Lệnh mul đầu tiên thực hiện bình thường do là lệnh mul basic ( nhân hai biến).

+ Lệnh mul thứ 2 không phải là lệnh basic ( vì nhân hằng với biến), nên thực hiện biến đổi thành hai lệnh ( gán hằng số cho thanh ghi tạm rồi mới thực hiện phép nhân).

*- Chạy từng dòng lệnh:*

Step 1:

Table

Description automatically generated

Step 2:

Table

Description automatically generated

Step 3:

Table

Description automatically generated

Step 4:

Table

Description automatically generated

Step 5:

Table

Description automatically generated

Step 6:

Table

Description automatically generated

- Sự thay đổi của các thanh ghi:

$t1 : 0x00000000 => 0x00000004

$t2: 0x00000000 => 0x00000005

$lo: 0x00000000 => 0x00000014 => 0x0000003c

$at: 0x00000000 => 0x00000003

$s0: 0x00000000 => 0x0000003c

$s1: 0x00000000 => 0x0000003c

Thanh ghi hi không thay đổi giá trị vì kết quả dưới 32 bit, kết quả ở đây được ghi vào thanh ghi lo.

Ex6:

- *Chương trình:*

Graphical user interface, text

Description automatically generated

Table

Description automatically generated

- Lệnh la được biên dịch bằng cách tách thành 2 lệnh basic là lui và ori.

- Ở cửa sổ Data Segment :

Địa chỉ của X = $t8

Địa chỉ của Y = $t9

Địa chỉ của Z = $t7

Tương ứng với hằng số khi biên dịch lệnh la thành mã máy.

Step 1:

Graphical user interface, application, table

Description automatically generated

Step 2:

Graphical user interface, application, table

Description automatically generated

Step 3:

Graphical user interface, application, table

Description automatically generatedGraphical user interface, application, table

Description automatically generated

Step 4:

Table

Description automatically generated

Step 5:

Graphical user interface, application, table

Description automatically generated

Step 6:

Table

Description automatically generated

Step 7:

Table

Description automatically generated

Step 8:

Table

Description automatically generated

Step 9:

Table

Description automatically generated

Step 10:

Table

Description automatically generated

Step 11:

Table

Description automatically generated

- Sự thay đổi giá trị các thanh ghi :

$at : 0x00000000 => 0x10010000

$t8 : 0x00000000 => 0x10010000

$t9 : 0x00000000 => 0x10010004

$t1 : 0x00000000 => 0x00000005

$t2 : 0x00000000 => 0xffffffff

$s0 : 0x00000000 => 0x0000000a => 0x00000009

$t7 : 0x00000000 => 0x10010008

- Vai trò của lệnh lw và sw :

+ lw $rt, imm($rs) : gán giá trị của thanh ghi $rs vào thanh ghi $rt ( $rt = M[$rs+imm])

+ Ở đây lw gán $t1 = $t8 và $t2 = $t9

+ sw $rt, imm($rs) : gán giá trị của thanh ghi $rt vào thanh ghi $rs (M[$rs+imm] = $rt)

- Các lệnh lb, sb:

+ lb : chép 1 byte tại vị trí trong bộ nhớ RAM vào byte thấp của thanh ghi.

+ sb : lưu một byte thấp trong thanh ghi vào vị trí trong bộ nhớ RAM.