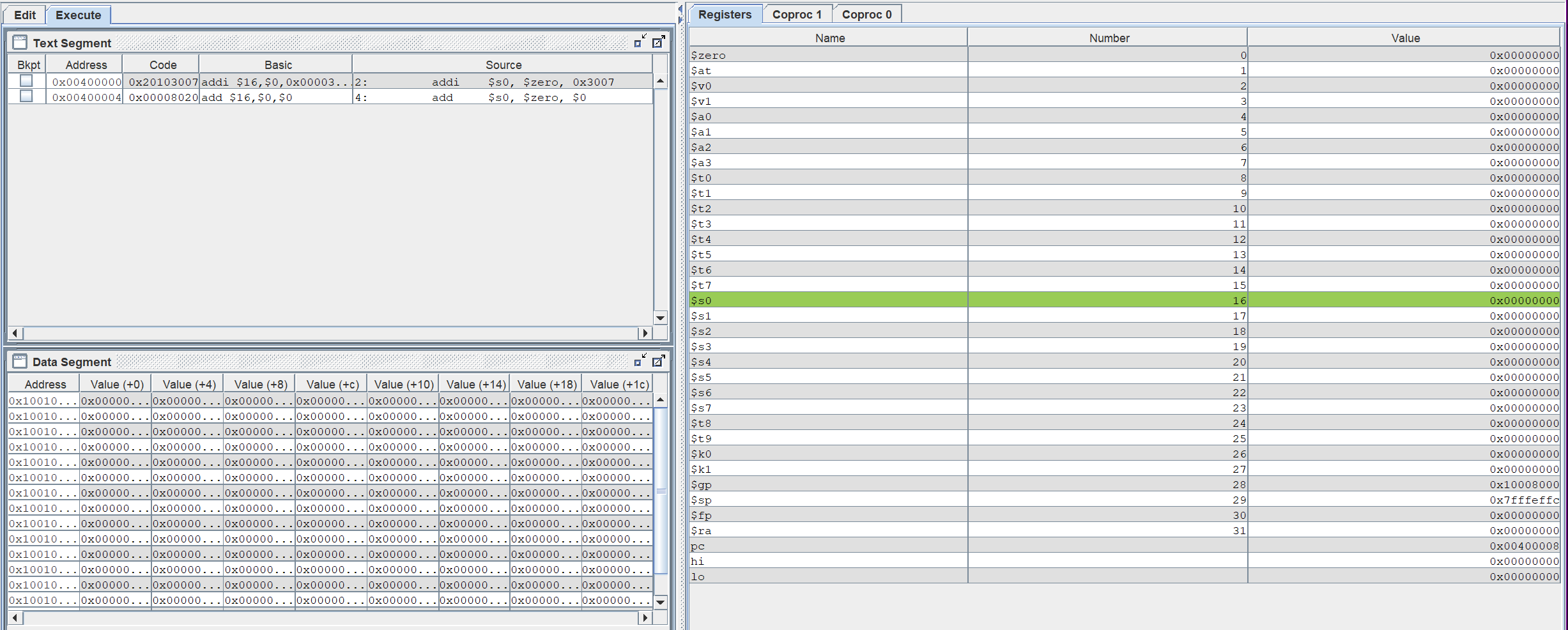
Báo cáo thực hành KTMT Tuần 2

Phạm Thành Lập 20215076

**Assigment 1**

****

|  |  |  |
| --- | --- | --- |
| Trạng thái | $s0 | pc |
| Ban đầu | 0x00000000 | 0x00400000 |
| Chạy addi | 0x00003007 | 0x00400004 |
| Chạy add | 0x00000000 | 0x00400008 |

* Lệnh addi $s0, $zero, 0x3007 ( addi $16, $0, 0x3007)

Opcode : 8 => 001000

Rs : 0 => 00000

Rt : 16 => 10000

Immi : 0x3007 => 0110 0000 0000 0111

Code: 0010 0000 0001 0000 0110 0000 0000 0111

Mã hex: 0x20103007

* Lệnh add $s0, $zero, $0 ( add $16, $0, $0)

Opcode : 0 => 000000

Rs : 0 => 00000

Rt : 0 => 00000

Rd : 16 => 01000

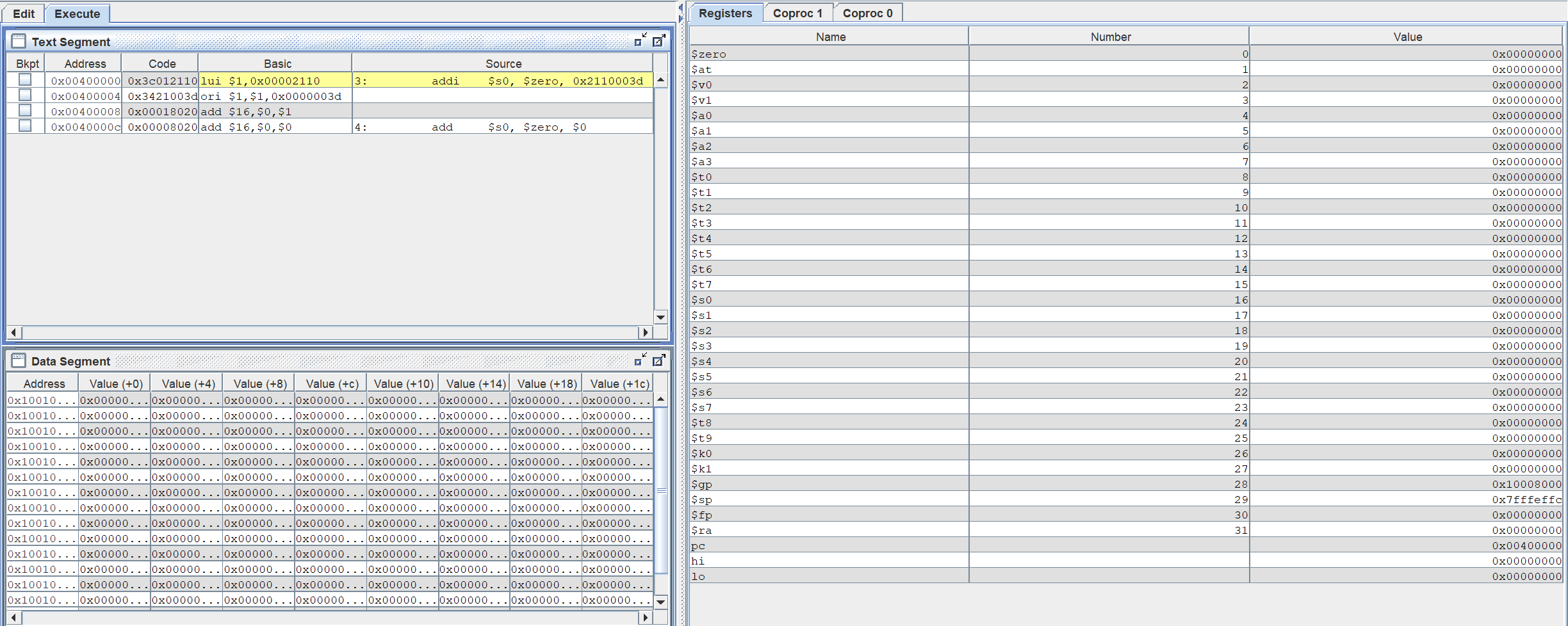
Shamt : 0 => 00000

Funt : 32 => 100000

Code : 0000 0000 0000 0000 1000 0000 0010 0000

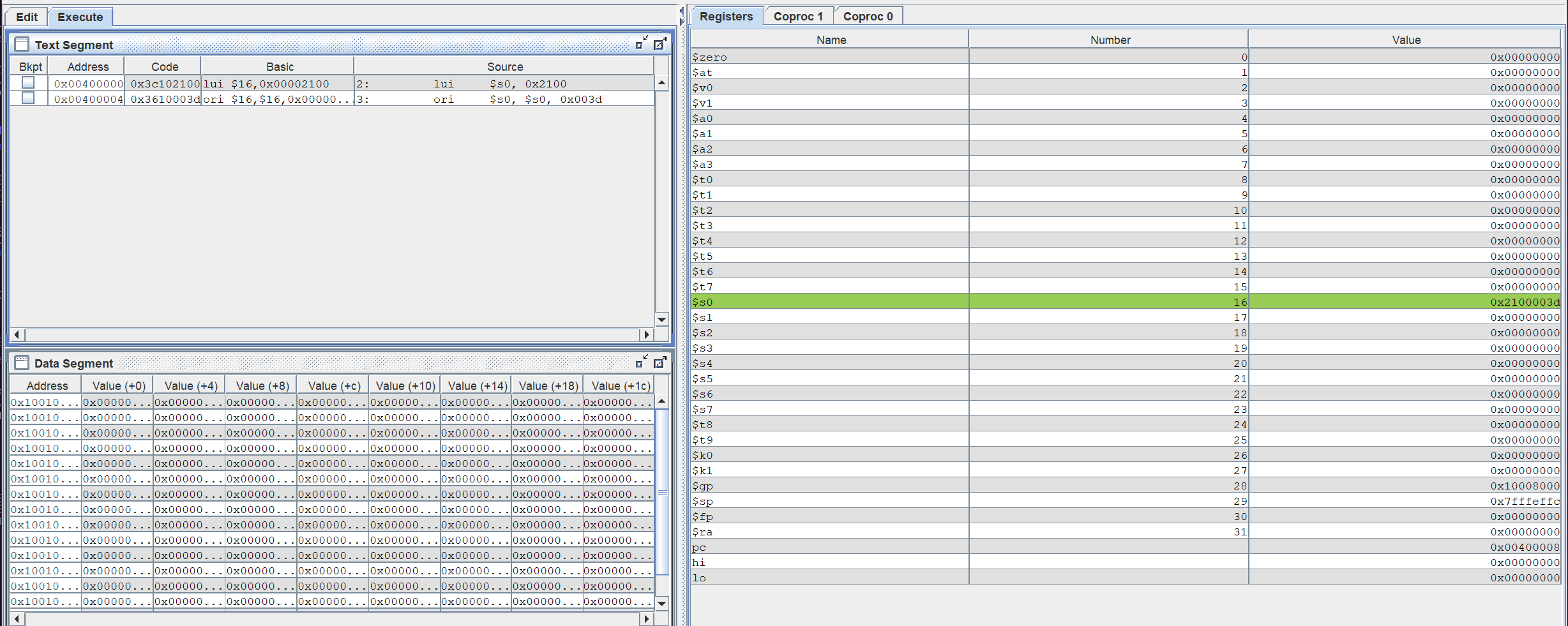
Mã hex: 0x00008020

* Cửa sổ textsegment trùng với mã máy

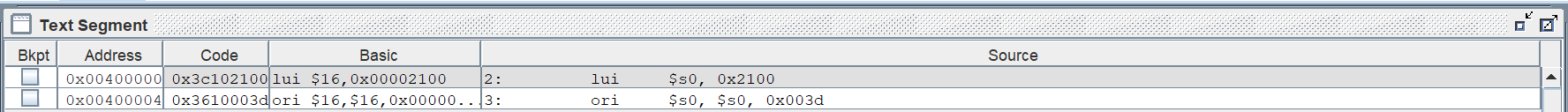


Chương trình không biên dịch lệnh addi mà thay vào đó là 2 lệnh lui và ori do 0x2110003d là 1 số 32 bit mà đây là format I lên chỉ truyền vào được 16 bit lên chương trình phải truyền 8 bit cao vào lui và 8 bit thấp vào ori để có thể chạy chương trình

**Assigment 2**

****

|  |  |  |
| --- | --- | --- |
| Trạng thái | $s0 | pc |
| Ban đầu | 0x00000000 | 0x00400000 |
| Chạy lui | 0x21000000 | 0x00400004 |
| Chạy ori | 0x2100003d | 0x00400008 |

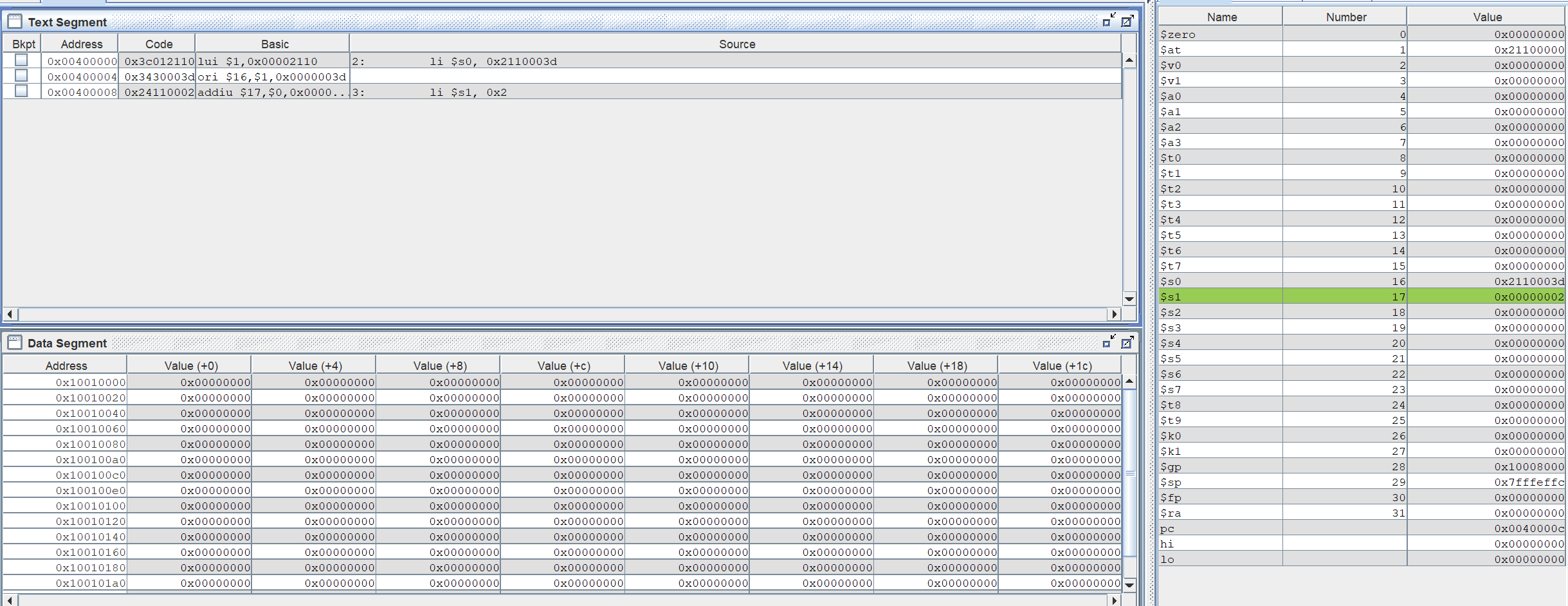


Ảnh có chứa văn bản

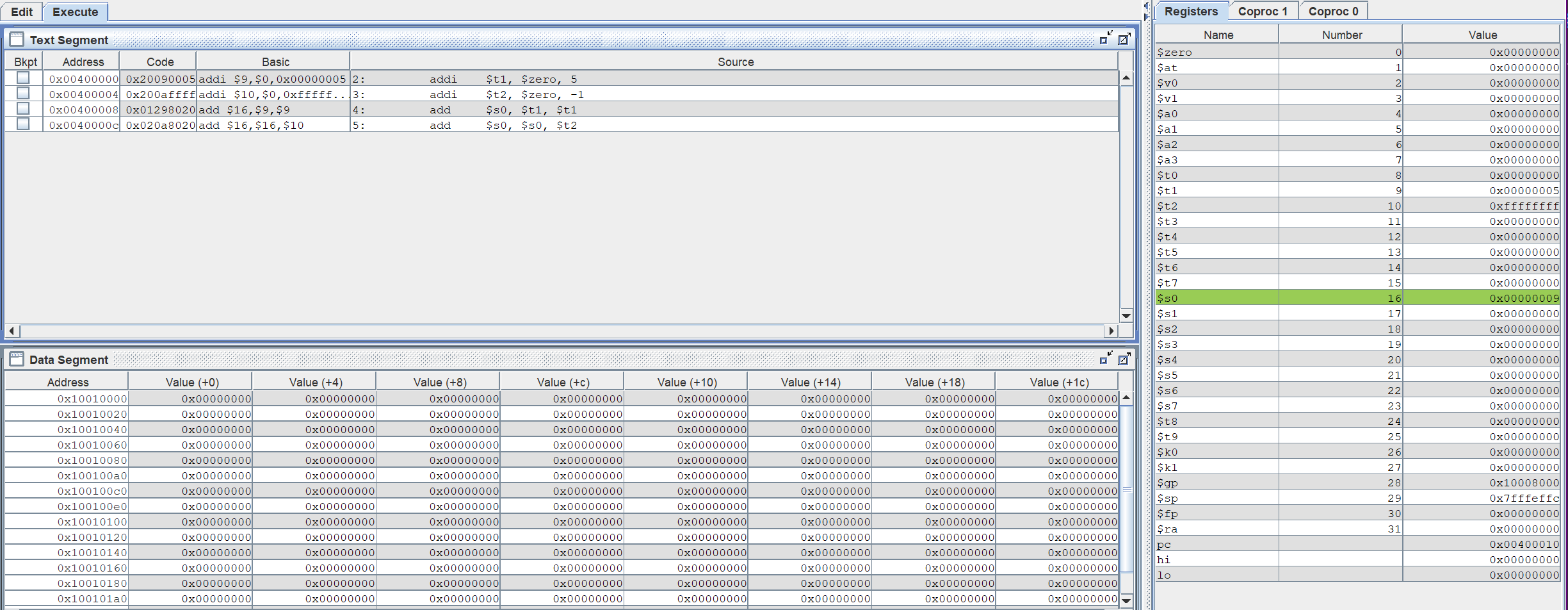
Mô tả được tạo tự động

Các byte đầu tiên của dòng lệnh ở cửa sổ datasegmnet trùng với Address và Code ở textsegment

**Assigment 3**



Dễ thấy lệnh li bị thay thế bằng câu lệnh khác 0x2110003d do là 32 bit nhưng do là format I lên 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.

**Assigment 4**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 |

Như vậy ta có kết quả là 9 đúng với phép tính 2\*5-1 = 9

* 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 )

Opcode : 0 => 000000

Rd : 16 => 10000

Rs : 9 => 10001

Rt : 9 => 10001

Shamt : 0 => 00000

Funt : 32 => 100000

Code: 0000 0001 0010 1001 1000 0000 0010 0000

Mã hex: 0x01298020

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

Opcode : 0 => 000000

Rd : 16 => 10000

Rs : 16 => 10000

Rt : 10 => 01010

Shamt : 0 => 00000

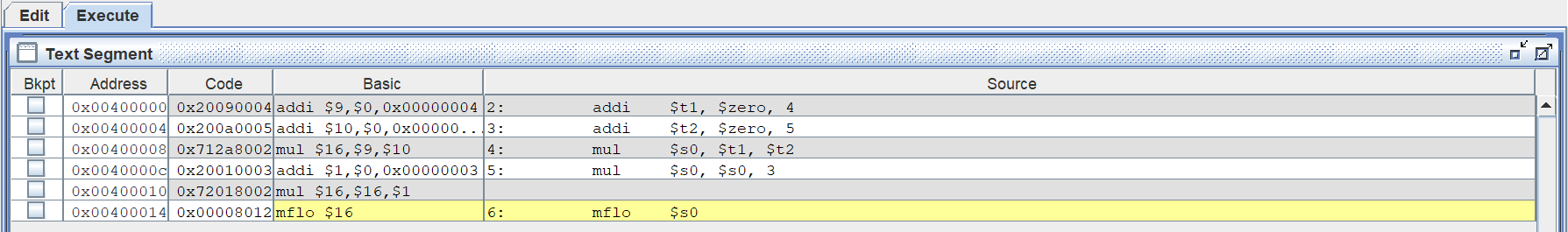
Funt : 32 => 100000

Code: 0000 0010 0000 1010 1000 0000 0010 0000

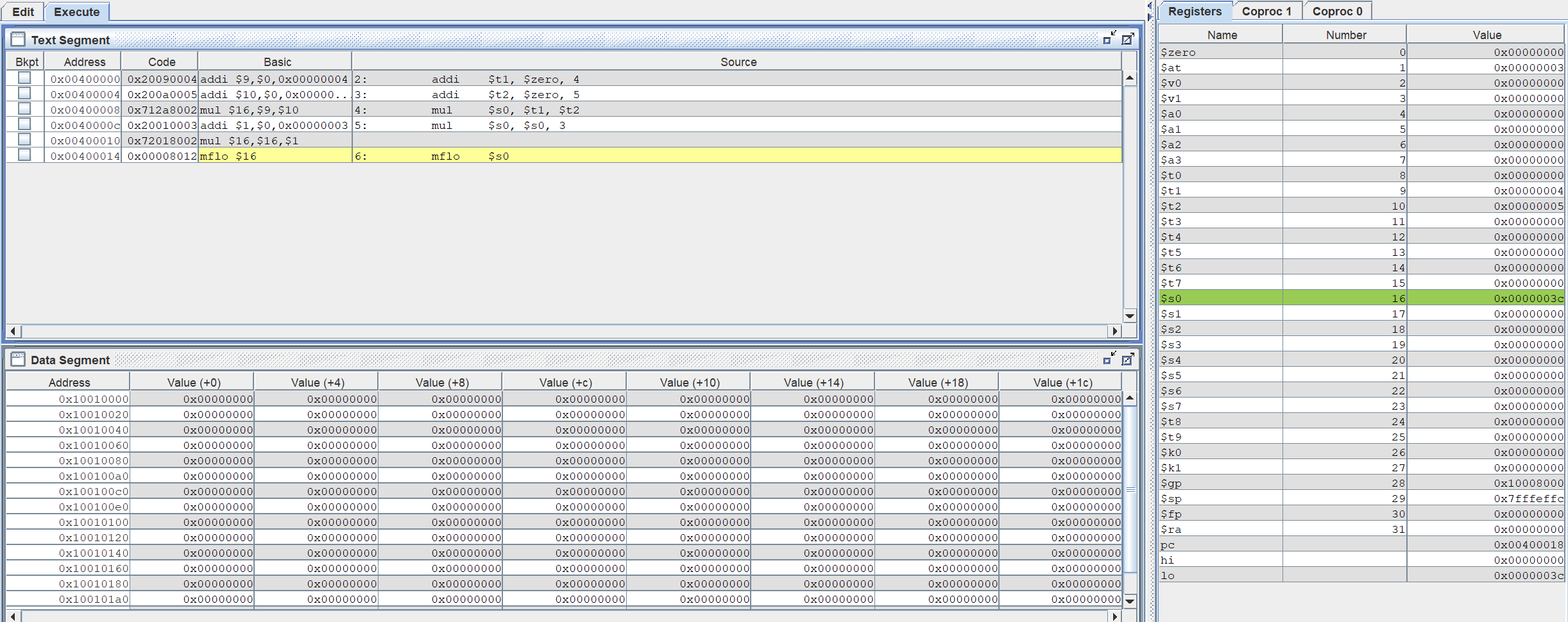
Mã hex: 0x020a8020

* Như vậy lệnh add phù hợp với R format

**Assigment 5**

****

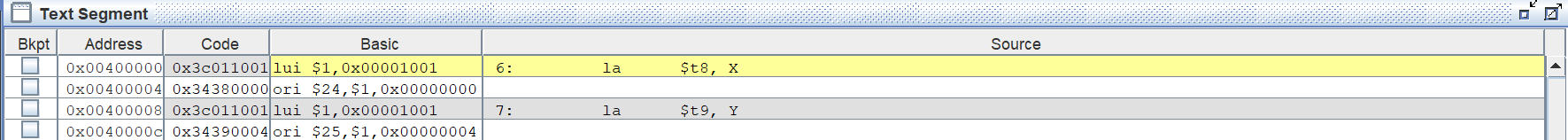
Dễ thấy lệnh mul với 1 số bị thay thế bằng 1 lệnh khác do lệnh mul là R format mà khi gõ ta gõ là I format lên chương trình phải thêm 1 lệnh khác để chuyển 0x3 thành 1 biến rồi mới thực hiện



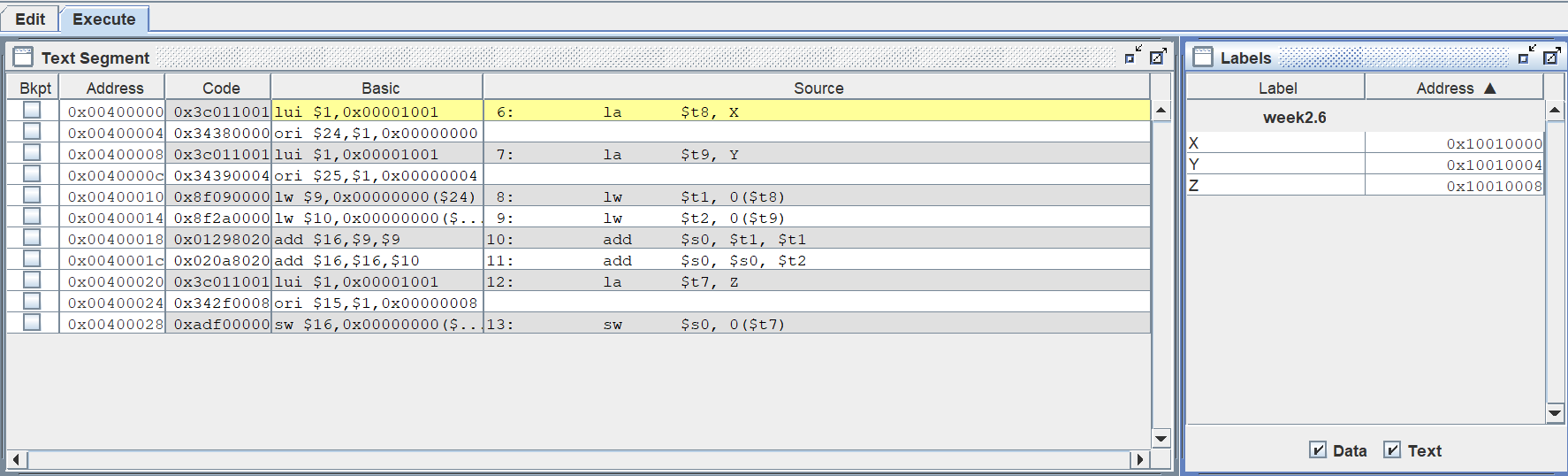
|  |  |  |
| --- | --- | --- |
| Trạng thái | hi | lo |
| Ban đầu | 0x00000000 | 0x00000000 |
| Chạy add 1 | 0x00000000 | 0x00000000 |
| Chạy add 2 | 0x00000000 | 0x00000000 |
| Chạy mul 1 | 0x00000000 | 0x00000014 |
| Chạy mul 2 | 0x00000000 | 0x0000003c |

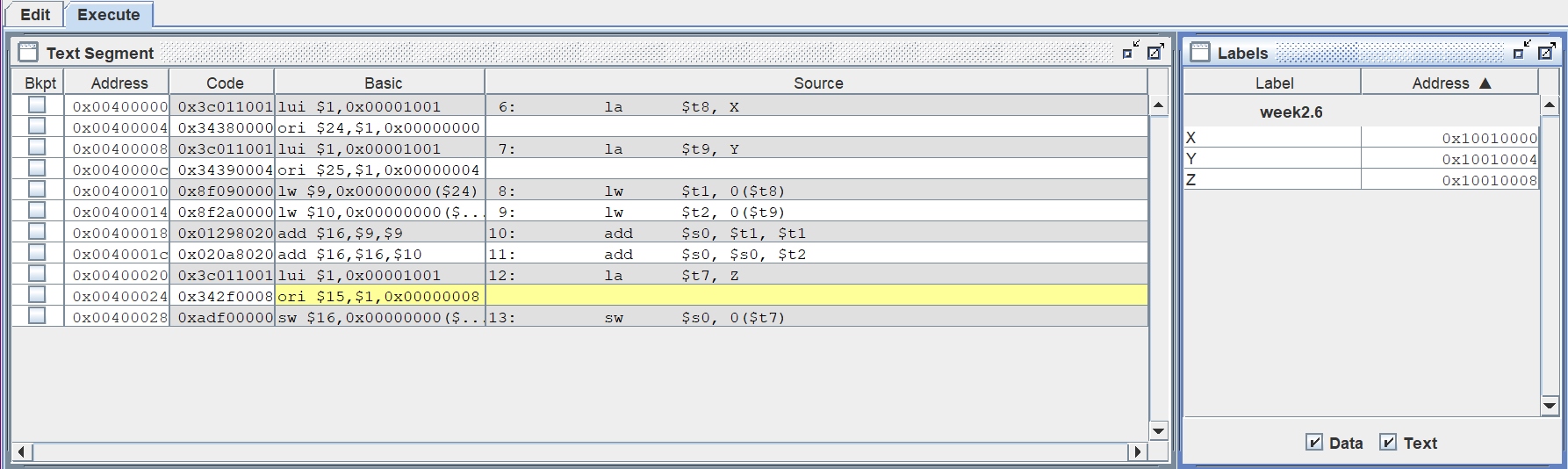
Như vậy kết quả cho ra đúng với phép toán 3\*4\*5 = 60 (3c)

**Assigment 6**

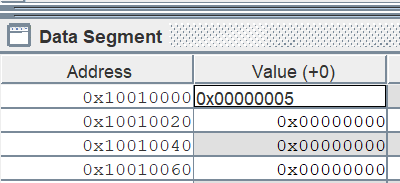


Lệnh la do chỉ lưu được địa chỉ 16 bit mà địa chỉ của biến là 32 lên la sẽ thực hiện 2 câu lệnh lui và ori để lưu địa chỉ của biến



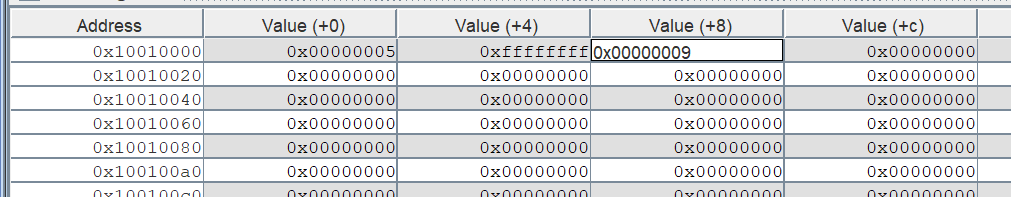


* Sau 3 lần chạy la thì ta thấy địa chỉ X,Y,Z bằng hằng số

X : 

Y: Ảnh có chứa bàn

Mô tả được tạo tự động

Z: 

Sự thay đổi giá trị của thanh ghi:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Trạng thái | $s0 | $t1 | $t2 | $t7 | $t8 | $t9 |
| Ban đầu | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| Chạy la 1 |  |  |  |  | 0x10010000 |  |
| Chạy la 2 |  |  |  |  |  | 0x10010004 |
| Chạy lw 1 |  | 0x00000005 |  |  |  |  |
| Chạy lw 2 |  |  | 0xffffffff |  |  |  |
| Chạy add 1 | 0x0000000a |  |  |  |  |  |
| Chạy add 2 | 0x00000009 |  |  |  |  |  |
| Chạy la |  |  |  | 0x10010008 |  |  |
| Chạy sw |  |  |  |  |  |  |

* lw: dùng để lưu giá trị trên địa chỉ vào địa chỉ còn lại
* sw: dùng để lưu giá trị trên thanh ghi vào 1 địa chỉ