

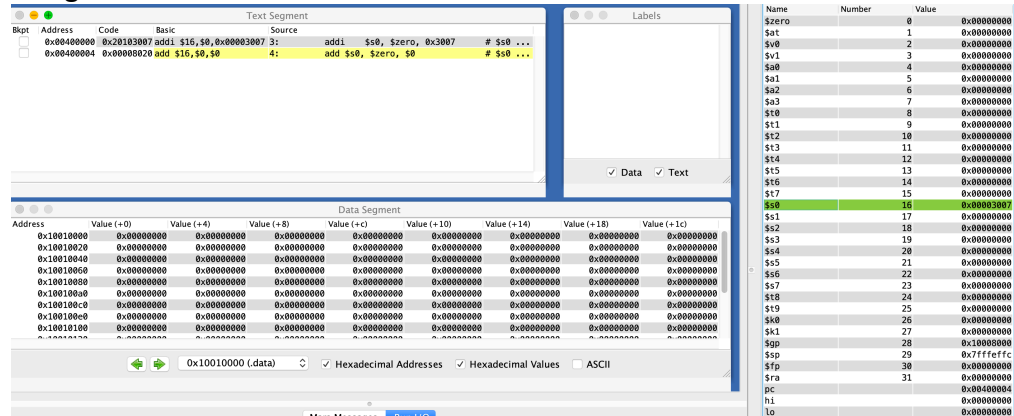


Báo Cáo Thực Hành

Môn: Thực Hành Kiến Trúc Máy Tính

Assignment 1:

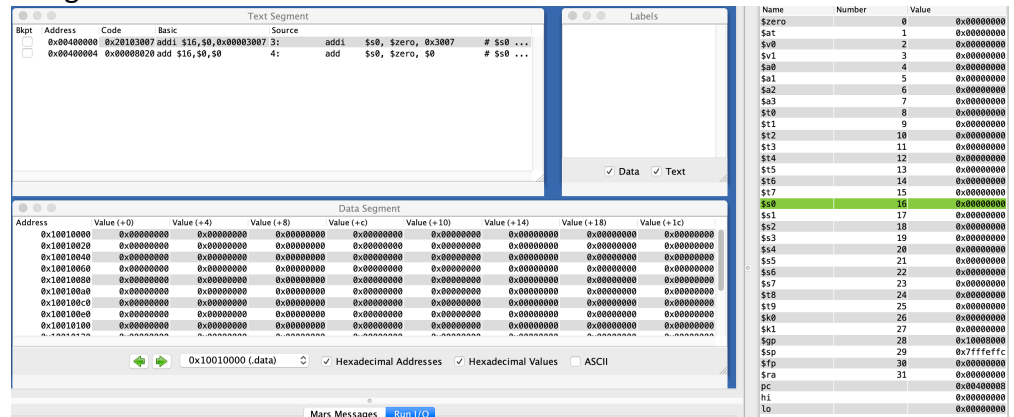
- Code ban đầu:
 - o Dòng 1:



Thanh ghi \$s0 thay đổi đúng bằng giá trị số đầu vào cộng với giá trị của thanh ghi \$0 bằng 0x3007

Thanh ghi \$pc chỉ đến địa chỉ của lệnh tiếp theo

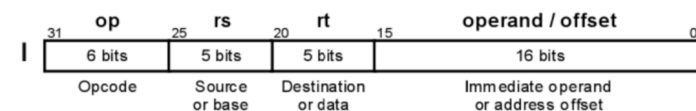
- o Dòng 2:



Thanh ghi \$s0 thay đổi đúng bằng giá trị thanh ghi \$0

Thanh ghi \$pc chỉ đến vùng nhớ tiếp theo (cộng thêm 4 byte)

- Khuôn dạng tập lệnh
 - o Lệnh addi \$16, \$0, 0x00003007:
 20103007(16)
 001000 00000 10000 0011000000000111 (2)
 Opcode = 8
 rs = 0
 rt = 16
 offset = 00003007(16)

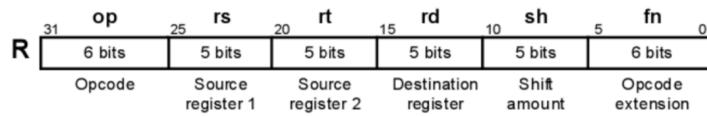


⇒ Chuẩn theo khuôn dạng mẫu I

- o Lệnh add \$16, \$0, \$0
 00008020(16)
 000000 00000 00000 10000 00000 100000 (2)
 Opcode = 0, fn = 32

rs = 0, rt = 0

rd = 16



⇒ Chuẩn khuôn dạng mẫu R

- Sửa thành `addi $s0, $zero, 0x2110003d`

Assembly code snippet:

```
0x00400000 0x3c102110 lui $1, 0x00002110
0x00400004 0x3421003d ori $1, $1, 0x0000003d
0x00400008 0x00018020 add $16, $0, $1
0x0040000c 0x00008020 add $16, $0, $0
```

Register File:

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x2110003d
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x2110003d
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$s8	24	0x00000000
\$s9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10000000
\$sp	29	0x7ffffcfc
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x0040000c
hi		0x00000000
lo		0x00000000

Câu lệnh `addi` sẽ được tách thành 3 câu lệnh nhỏ ở Basic

Do số đầu vào là 32bit(2) nên quy trình thực hiện sẽ là:

Lệnh lui nạp 16 bit trên gán vào \$1, sau đó lệnh ori nối 16 bit dưới với 16 bit trên vào \$1 và cuối cùng cho \$1 + \$0 gán vào \$16

Assignment 2:

- Dòng 1:

Assembly code snippet:

```
0x00400000 0x3c102110 lui $16, 0x00002110
0x00400004 0x3421003d ori $16, $16, 0x0000003d
```

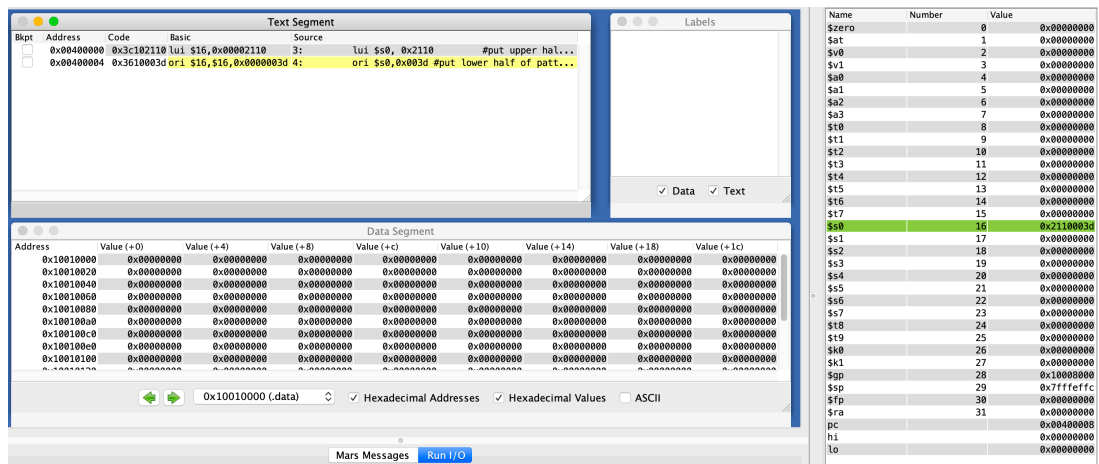
Register File:

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x00000000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x2110003d
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$s8	24	0x00000000
\$s9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10000000
\$sp	29	0x7ffffcfc
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x0040000c
hi		0x00000000
lo		0x00000000

Thanh ghi \$s0 được gán là nửa trên từ giá trị ban đầu 0x00002110

Thanh ghi \$pc chỉ đến địa chỉ của lệnh tiếp theo

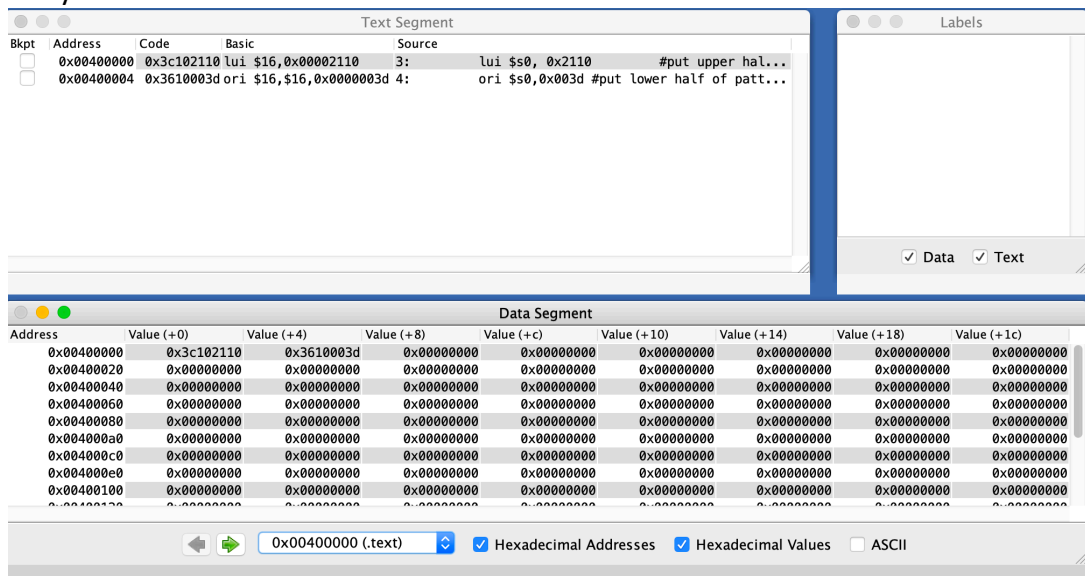
- Dòng 2:



Thanh ghi \$s0 được gán nối tiếp nửa sau từ giá trị 0x0000003d

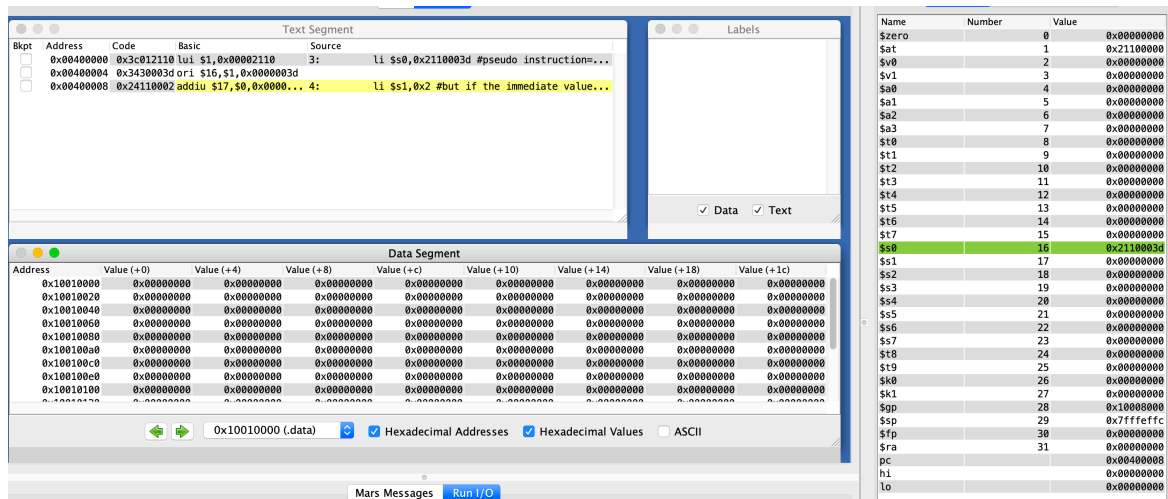
Thanh ghi \$pc chỉ đến vùng nhớ tiếp theo (cộng thêm 4 byte)

- Chuyển hết .text



Byte đầu tiên của vùng lệnh trùng với hàng đầu tiên của cột Address trong Text Segment

Assignment 3:



- Lệnh li \$s0, 0x2110003d được tách thành 2 lệnh:
 - o lui \$1, 0x00002110
 - o ori \$16, \$s1, 0x0000003d
- Vì lệnh I của MIPS chỉ có tham số 16 bits, nên được tách thành 2 lệnh.
- Lệnh li \$s1, 0x2 có tham số nhỏ hơn 16 bits nên không cần tách.

Assignment 4:

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x00000000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0x00000005
\$t2	10	0xffffffff
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000009
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10000000
\$sp	29	0x7fffffc0
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x00400010
hi		0x00000000
lo		0x00000000

- Sau lệnh 1, thanh \$t1 có giá trị 0x00000005
- Sau lệnh 2, thanh \$t2 có giá trị 0xffffffff
- Sau lệnh 3, thanh \$s0 có giá trị 0x0000000a
- Sau lệnh 4, thanh \$s0 có giá trị 0x00000009

=> Kết quả đúng

Sau kiểm nghiệm, lệnh addi phù hợp với khuôn mẫu lệnh I, lệnh add phù hợp với khuôn mẫu lệnh R

Assignment 5:

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x00000003
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0x00000004
\$t2	10	0x00000005
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x0000000c
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10000000
\$sp	29	0x7fffffc0
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x00400014
hi		0x00000000
lo		0x0000000c

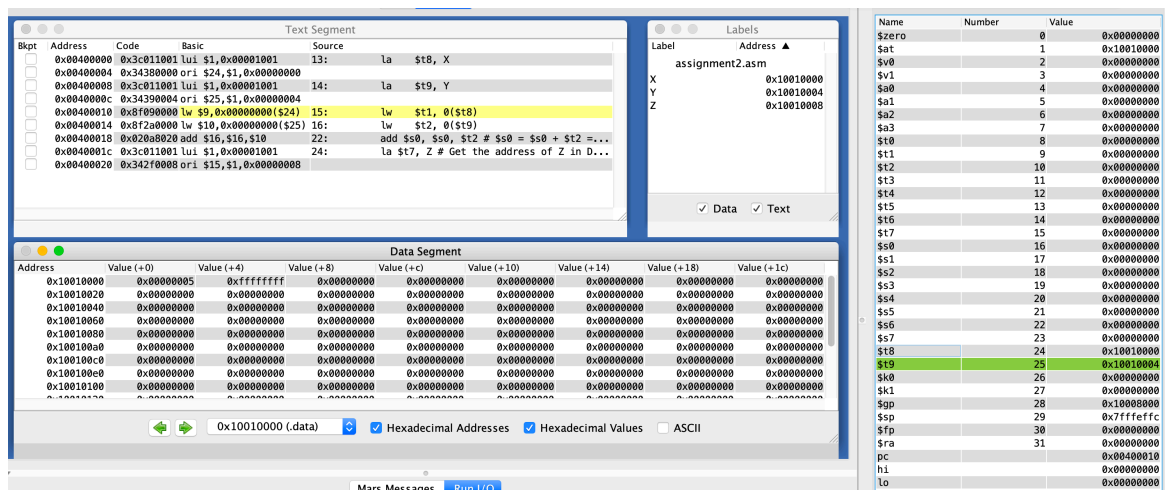
- Sau lệnh 1,2 \$t1 và \$t2 có giá trị lần lượt là 0x00000004 và 0x00000005

- Sau lệnh 3, \$s0 và lo có giá trị là 0x00000014
- Lệnh 4 được tách làm 2 lệnh mul và addi. Lệnh addi lưu 3 vào thanh ghi at. Sau đó lệnh mul mới tính tích của at và \$s0. Sau lệnh thanh lo, \$s0 có giá trị 0x0000003c
- Lệnh cuối cùng thay đổi giá trị \$s1 = \$s0

Assignment 6:

- Lệnh la được tách thành 2 lệnh lui và ori để load địa chỉ của biến x vì địa chỉ biến x là 32 bits.
- Địa chỉ của x, y, z được tách thành 2 nửa

Biến	Địa chỉ	Nửa trên	Nửa dưới
x	0x10010000	0x00001001	0x00000000
y	0x10010004	0x00001001	0x00000004
z	0x10010008	0x00001001	0x00000008



- Bảng label:

X	0x10010000
Y	0x10010004
Z	0x10010008

☒ Data ☒ Text

- Giá trị của x, y, z:

Biến	Giá trị
x	0x00000005
y	0xffffffff
z	0x00000000 (không khởi tạo giá trị)

- Lệnh lw: load giá trị kiểu word từ thanh ghi \$t8, \$t9 vào thanh ghi \$t1, \$t2.
- Lệnh sw: store giá trị kiểu word từ thanh ghi \$s0 vào thanh ghi \$t7.
- Lệnh lb: load giá trị kiểu byte vào một thanh ghi.
- Lệnh sb: store giá trị low-order 8 bits vào một thanh ghi.