Báo cáo tuần 7

Bài 1 :

```
.text
main:
        li $a0, -440 #load input parameter
        jal abs #jump and link to abs procedure
        nop
        add $s0, $zero, $v0
        li $v0, 10 #terminate
        syscall
endmain:
#---------------------------------------------------------------
# function abs
# param[in] $a0 the interger need to be gained the absolute
value:
# return $v0 absolute value
#---------------------------------------------------------------
abs:
        sub $v0,$zero,$a0 #put -(a0) in v0; in case (a0)<0
        bltz $a0,done #if (a0)<0 then done
        nop
        add $v0,$a0,$zero #else put (a0) in v0

done:
      jr $ra
```

Run

Text Segment

Bkpt	Address	Code	Basic		Source
	0x00400000	0x2404fe48	addiu $4,$0,-440	3:	li $a0, -440 #load input parameter
	0x00400004	0x0c100006	jal 0x00400018	4:	jal abs #jump and link to abs procedure
	0x00400008	0x00000000	nop	5:	nop
	0x0040000c	0x00028020	add $16,$0,$2	6:	add $s0, $zero, $v0
	0x00400010	0x2402000a	addiu $2,$0,10	7:	li $v0, 10 #terminate
	0x00400014	0x0000000c	syscall	8:	syscall
	0x00400018	0x00041022	sub $2,$0,$4	17:	sub $v0,$zero,$a0 #put -(a0) in v0; in case (a
	0x0040001c	0x04800002	bltz $4,2	18:	bltz $a0,done #if (a0)<0 then done
	0x00400020	0x00000000	nop	19:	nop
	0x00400024	0x00801020	add $2,$4,$0	20:	add $v0,$a0,$zero #else put (a0) in v0
	0x00400028	0x03e00008	jr $31	23:	jr $ra

Giải thích code

Đầu tiên ta cần nạp giá trị vào biến a0

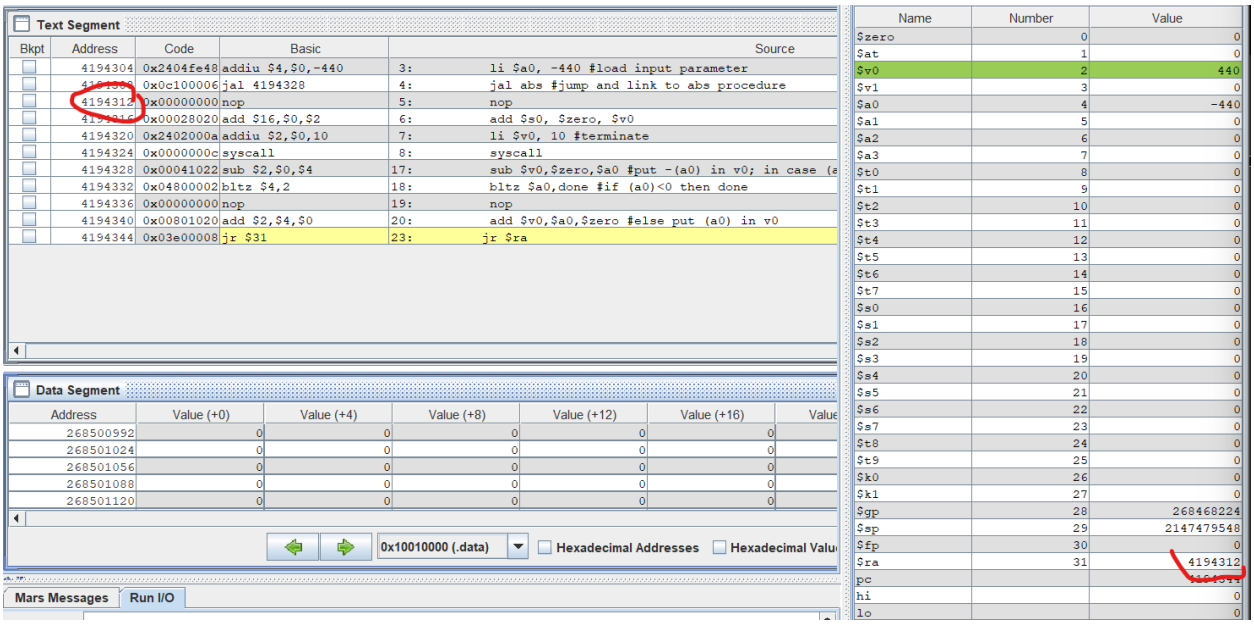
Tiếp đó ta nhảy tới abs

Ta lưu v0 = số đối của A

Ta kiểm tra xem a0 có < 0 hay không

Nếu như mà nhỏ hơn 0 thì nhảy tới done

Done nhảy tới địa chỉ của ra tức là địa chỉ của nop đầu tiên



Nếu như a0 < 0 vậy thì số đối của nó tức là v0 đang > 0 và ta thực hiện nhảy tới nop đầu tiên, nạp giá trị vào s0 và gọi lệnh syscall để kết thúc chương trình

Còn nếu như mà a0 > 0 tức là số đối v0 đang nhỏ hơn 0

Thi ta thực hiện lệnh v0 = a0 + 0 để v0 > 0 = a0

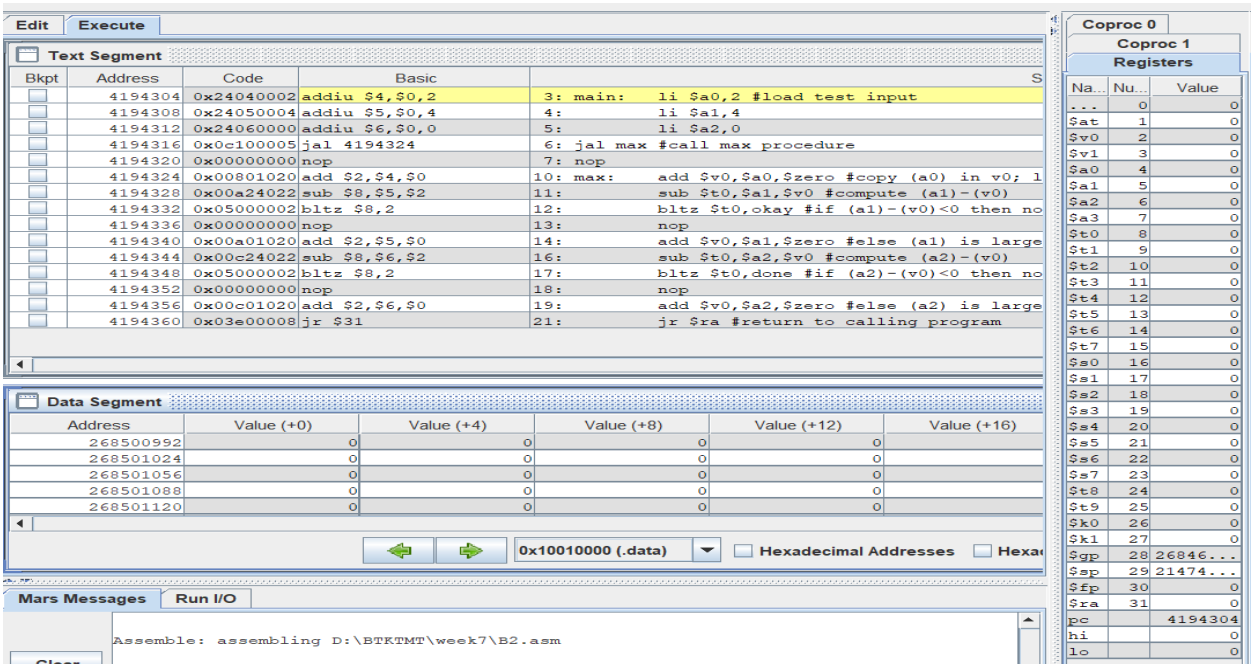
Sau đó nhảy tới nop đầu tiên và in ra lệnh như trên

Bài 2 :

Code

```
1
2   .text
3   main:   li $a0,2 #load test input
4           li $a1,4
5           li $a2,0
6   jal max #call max procedure
7   nop
8   j endmain
9
10  max:    add $v0,$a0,$zero #copy (a0) in v0; largest so far
11          sub $t0,$a1,$v0 #compute (a1)-(v0)
12          bltz $t0,okay #if (a1)-(v0)<0 then no change
13          nop
14          add $v0,$a1,$zero #else (a1) is largest thus far
15  okay:
16          sub $t0,$a2,$v0 #compute (a2)-(v0)
17          bltz $t0,done #if (a2)-(v0)<0 then no change
18          nop
19          add $v0,$a2,$zero #else (a2) is largest overall
20  done:
21          jr $ra #return to calling program
22
23  endmain:
```

Run



Giải thích code

Đầu tiên ta nhập các giá trị để tìm max vào lần lượt a0, a1, a2

Sau đó ta nhảy tới max

Ta lấy v0 = giá trị đầu tiên = a0

Ta thực hiện lệnh trừ t0 = a1 – v0 tức là a1 – a0

Vậy nếu như mà <0 thì nhảy tới okey (vì ở đây v0 được coi là max, và lúc này a0 đang lớn hơn a1)

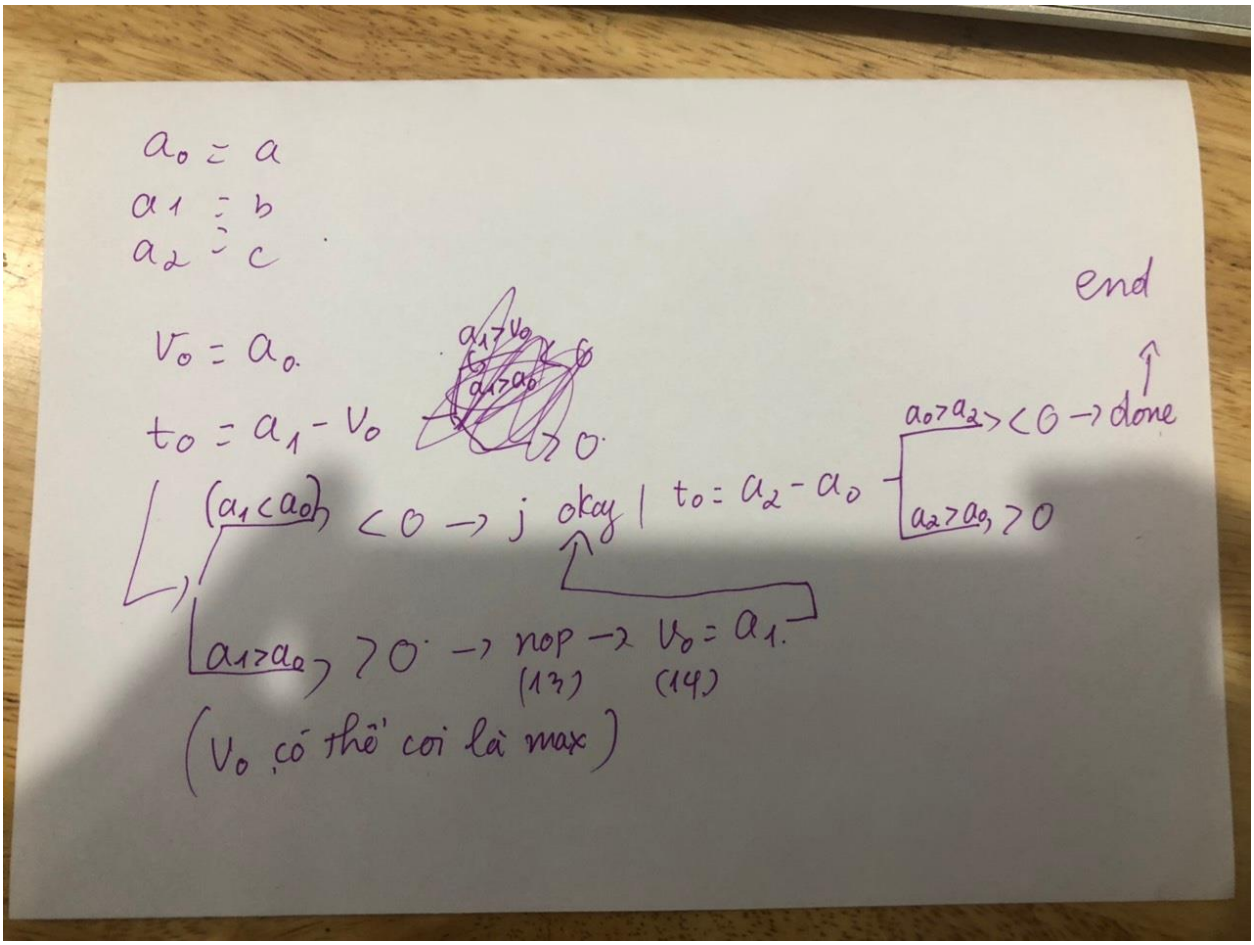
Ta thực hiện tiếp lệnh số 2 là mang a0 đi so sánh với a2

Tương tự như vậy ta nhảy tới done để nhảy tới nop kết thúc chương trình

Còn ở bước trên nếu như t0 > 0 tức là v0 = a1 vì lúc này a1 đang lớn hơn a0

Vậy nên ta có bước số 2 là mang v0 đi so sánh với a2 tức là mang a1 đi so sánh với a2

Để cho dễ hiểu ta có thể coi thuật toán mà em trình bày ở đây


$a_0 = a$
$a_1 = b$
$a_2 = c$
$V_0 = a_0$
$t_0 = a_1 - V_0$
$a_1 > V_0$
$a_1 > a_0$
> 0
$(a_1 < a_0) < 0 \rightarrow$ j okay
$a_1 > a_0 > 0 \rightarrow$ nop (13) $V_0 = a_1$ (14)
(V_0 có thể coi là max)
$t_0 = a_2 - a_0$
$a_0 > a_2 > < 0 \rightarrow$ done
$a_2 > a_0 > 0$
end

Bài 3