Laboratory 3

Load/ Store , Jump & Branch instructions

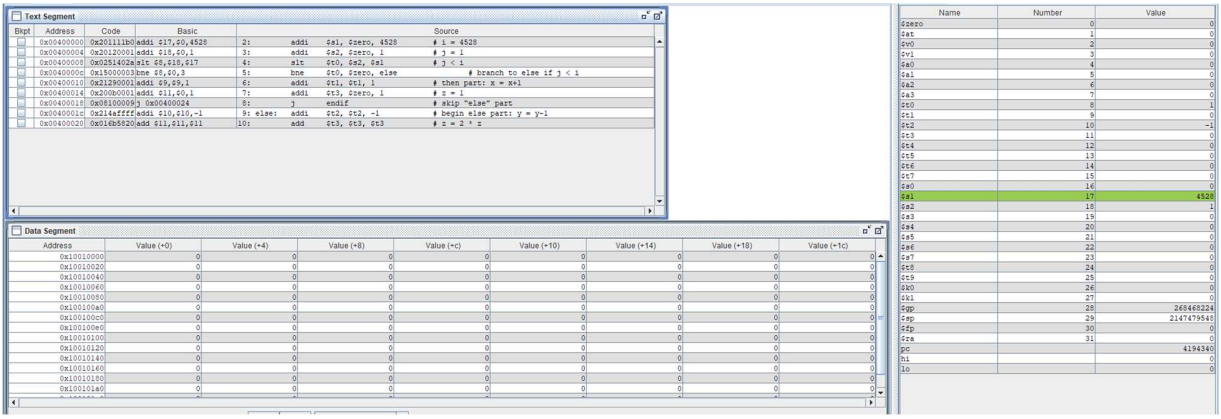
Đỗ Hải Dương - 20194528

Assignment 1

i = 4528 ; j =1

Assignment 1.asm

```
1   start:
2           addi    $s1, $zero, 4528        # i = 4528
3           addi    $s2, $zero, 1           # j = 1
4           slt     $t0, $s2, $s1           # j < i
5           bne     $t0, $zero, else                # branch to else if j < i
6           addi    $t1, $t1, 1             # then part: x = x+1
7           addi    $t3, $zero, 1           # z = 1
8           j       endif                   # skip "else" part
9   else:   addi    $t2, $t2, -1            # begin else part: y = y-1
10          add     $t3, $t3, $t3           # z = 2 * z
11  endif:
```



- Step 1: khởi tạo i = 4528

- Step 2: khởi tạo j = 1

- Step 3: so sánh, nếu giá trị của $s2 < $s1 thì $t0 = 1, còn $s2 >= $s1 thì $t0 = 0

- Step 4: Nếu $t0 != 0 thì rẽ vào nhánh “else”

- Nếu $t0 = 0 thì : Step 5: tăng $t1 thêm 1

Step 6: cho $t3 = 1

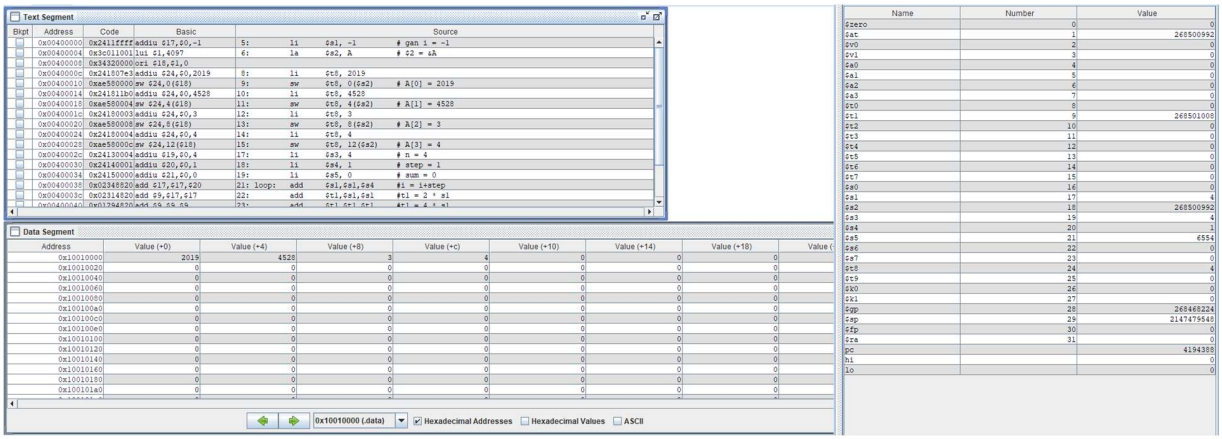
- Step 7: bước nhảy endif

- Nếu $t0 != 0 thì: Step 8: giảm $t2 đi 1

Step 9: nhân đôi $t3

Assignment 2

```
1   .data
2   A: .space 16 # Khai bao mang A co 4 phan tu
3
4   .text
5           li      $s1, -1         # gan i = -1
6           la      $s2, A          # $2 = &A
7
8           li      $t8, 2019
9           sw      $t8, 0($s2)     # A[0] = 2019
10          li      $t8, 4528
11          sw      $t8, 4($s2)     # A[1] = 4528
12          li      $t8, 3
13          sw      $t8, 8($s2)     # A[2] = 3
14          li      $t8, 4
15          sw      $t8, 12($s2)    # A[3] = 4
16
17          li      $s3, 4          # n = 4
18          li      $s4, 1          # step = 1
19          li      $s5, 0          # sum = 0
20
21  loop:   add     $s1,$s1,$s4     #i = i+step
22          add     $t1,$s1,$s1     #t1 = 2 * s1
23          add     $t1,$t1,$t1     #t1 = 4 * s1
24          add     $t1,$t1,$s2     #t1 store the address of A[i]
25          lw      $t0,0($t1)      #load value of A[i] in $t0
26          add     $s5,$s5,$t0     #sum = sum + A[i]
27          bne     $s1,$s3,loop    #if i != n, goto loop
28
```

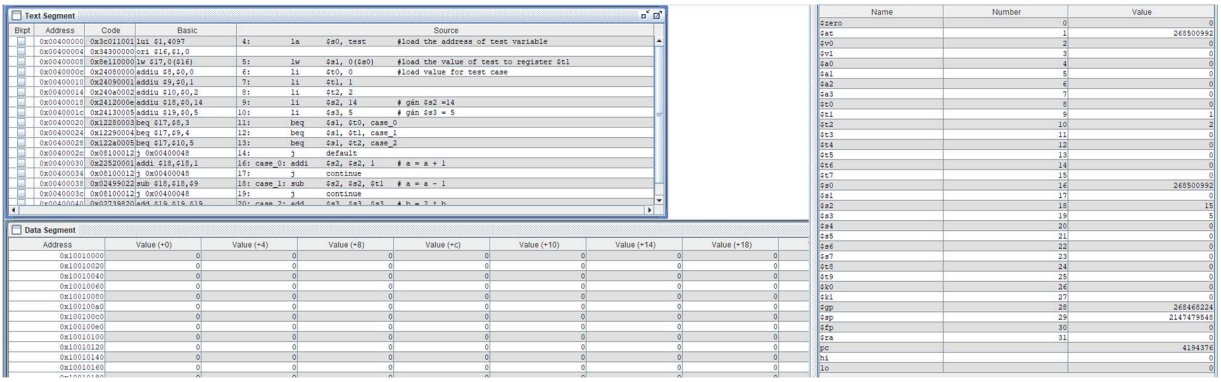


Nhận xét:

- Khi khởi tạo giá trị của các biến ban đầu là I = 0, A[] = [2019, 4528, 3, 4], n = 4, step = 1 và sum = 0. Chương trình chạy qua các phần tử của mảng A[1], A[2], A[3] và đọc giá trị vào thanh gih $t0 rồi cộng tổng vào thanh ghi chứa sum là $s5.
- Giá trị thanh ghi lưu địa chỉ của phần tử A[i] cũng thay đổi theo từng step.
- Sau khi chạy xong, kết quả của thanh ghi $s5 = 6554 đúng như kết quả mong đợi.

Assignment 3

```
1   .data
2   test: .word 0
3   .text
4           la      $s0, test       #load the address of test variable
5           lw      $s1, 0($s0)     #load the value of test to register $t1
6           li      $t0, 0          #load value for test case
7           li      $t1, 1
8           li      $t2, 2
9           li      $s2, 14         # gán $s2 =14
10          li      $s3, 5          # gán $s3 = 5
11          beq     $s1, $t0, case_0
12          beq     $s1, $t1, case_1
13          beq     $s1, $t2, case_2
14          j       default
15
16  case_0: addi    $s2, $s2, 1     # a = a + 1
17          j       continue
18  case_1: sub     $s2, $s2, $t1   # a = a - 1
19          j       continue
20  case_2: add     $s3, $s3, $s3   # b = 2 * b
21          j       continue
22  default:
23  continue:
24
```



Nhận xét:

- Dòng 4: $s0 lưu địa chỉ biến test
- Dòng 5: $s1 = 1 (Giá trị biến test)
- Dòng 6, 7, 8: Gán 1, 2, 3 vào các thanh ghi $t0, $t1, $t2
- Dòng 9, 10: Gán giá trị cho hai thanh ghi $s2, $s3
- Dòng 11, 12, 13: Các câu lệnh so sánh để rẽ nhánh

- $s2 = 14, $s3 = 5
  - ✓ test = 0 => case_0: giá trị lưu ở thanh ghi $s2 sẽ tăng lên 1 và bằng 15

$s2	18	15
$s3	19	5

  - ✓ test = 1 => case_1: giá trị lưu ở thanh ghi $s2 sẽ giảm đi 1 và bằng 13

$s2	18	13
$s3	19	5

  - ✓ test = 2 => case_2: giá trị lưu ở thanh ghi $s3 sẽ tăng lên 2 lần và bằng 10

$s2	18	14
$s3	19	10

Assignment 4

a. i < j

```
1  start:
2          addi $s1, $zero, 3       # i = 3
3          addi $s2, $zero, 4       # j = 4
4          slt $t0, $s1, $s2        # i < j
5          bne $t0, $zero, else     # branch to else if j < i
6          addi $t1, $t1, 1         #then part: x = x + 1
7          addi $t3, $zero, 1       # z = 1
8          j endif                  # skip "else" part
9  else:   addi $t2, $t2, -1        #begin else part: y = y + 1
10         add $t3, $t3, $t3        # z = 2 * z
11 endif:
```

$zero	0	0
$at	1	0
$v0	2	0
$v1	3	0
$a0	4	0
$a1	5	0
$a2	6	0
$a3	7	0
$t0	8	1
$t1	9	0
$t2	10	-1
$t3	11	0
$t4	12	0
$t5	13	0
$t6	14	0
$t7	15	0
$s0	16	0
$s1	17	3
$s2	18	4
$s3	19	0
$s4	20	0
$s5	21	0
$s6	22	0
$s7	23	0
$t8	24	0
$t9	25	0
$k0	26	0

b. i > = j

```
1  start:
2          addi    $s1, $zero, 3          # i = 3
3          addi    $s2, $zero, 2          # j = 2
4          slt     $t0, $s1, $s2          # i >= j
5          beq     $t0, $zero, else       # branch to else if i >= j
6          addi    $t1, $t1, 1            # then part: x = x + 1
7          addi    $t3, $zero, 1          # z = 1
8          j       endif                  # skip "else" part
9  else:   addi    $t2, $t2, -1           #begin else part: y = y + 1
10         add     $t3, $t3, $t3          # z = 2 * z
11 endif:
12
```

$zero	0	0
$at	1	0
$v0	2	0
$v1	3	0
$a0	4	0
$a1	5	0
$a2	6	0
$a3	7	0
$t0	8	0
$t1	9	0
$t2	10	0
$t3	11	0
$t4	12	0
$t5	13	0
$t6	14	0
$t7	15	0
$s0	16	0
$s1	17	0
$s2	18	0
$s3	19	0
$s4	20	0
$s5	21	0
$s6	22	0
$s7	23	0
$t8	24	0
$t9	25	0

c. i + j <= 0

```
1  start:
2          addi    $s1, $zero, -3         # i = -3
3          addi    $s2, $zero, 2          # j = 2
4          add     $s3, $s1, $s2          # s3 = i + j
5          slt     $t0, $zero, $s3        # i + j <= 0
6          bne     $t0, $zero, else       # branch to else if i + j > 0
7          addi    $t1, $t1, 1            # then part: x = x + 1
8          addi    $t3, $zero, 1          # z = 1
9          j       endif                  # skip "else" part
10 else:   addi    $t2, $t2, -1           #begin else part: y = y + 1
11         add     $t3, $t3, $t3          # z = 2 * z
12 endif:
13
```

⇨ **Kết quả:**

Registers | Coproc 1 | Coproc 0

Name	Number	Value
$zero	0	0
$at	1	0
$v0	2	0
$v1	3	0
$a0	4	0
$a1	5	0
$a2	6	0
$a3	7	0
$t0	8	0
$t1	9	1
$t2	10	0
$t3	11	1
$t4	12	0
$t5	13	0
$t6	14	0
$t7	15	0
$s0	16	0
$s1	17	-3
$s2	18	2
$s3	19	-1
$s4	20	0
$s5	21	0
$s6	22	0
$s7	23	0
$t8	24	0
$t9	25	0
$k0	26	0
$k1	27	0
$gp	28	268468224
$sp	29	2147479548
$fp	30	0
$ra	31	0
pc		4194344
hi		0
lo		0

d. i + j > m + n

```
1   start:
2           addi    $s1, $zero, 3           # i = 3
3           addi    $s2, $zero, 2           # j = 2
4           add     $s3, $s1, $s2           # s3 = i + j
5           addi    $s4, $zero, 1           # i = 1
6           addi    $s5, $zero, 2           # j = 2
7           add     $s6, $s4, $s5           # s6 = m + n
8           slt     $t0, $zero, $s3
9           bne     $t0, $zero, else
10          addi    $t1, $t1, 1             # then part: x = x + 1
11          addi    $t3, $zero, 1           # z = 1
12          j       endif                   # skip "else" part
13  else:   addi    $t2, $t2, -1            #begin else part: y = y + 1
14          add     $t3, $t3, $t3           # z = 2 * z
15  endif:
```

➪ **Kết quả:**

Registers | Coproc 1 | Coproc 0

Name	Number	Value
$zero	0	0
$at	1	0
$v0	2	0
$v1	3	0
$a0	4	0
$a1	5	0
$a2	6	0
$a3	7	0
$t0	8	1
$t1	9	0
$t2	10	-1
$t3	11	0
$t4	12	0
$t5	13	0
$t6	14	0
$t7	15	0
$s0	16	0
$s1	17	3
$s2	18	2
$s3	19	5
$s4	20	1
$s5	21	2
$s6	22	3
$s7	23	0
$t8	24	0
$t9	25	0
$k0	26	0
$k1	27	0
$gp	28	268468224
$sp	29	2147479548
$fp	30	0
$ra	31	0
pc		4194356
hi		0
lo		0

Assignment 5

a. i < n

```
1   .text
2
3   loop:   add     $s1, $s1, $s4   #i=i+step
4           add     $t1, $s1, $s1   #t1=2*s1
5           add     $t1, $t1, $t1   #t1=4*s1
6           add     $t1, $t1, $s2   #t1 store the address of A[i]
7           lw      $t0, 0($t1)     #load value of A[i] in $t0
8           add     $s5, $s5, $t0   #sum=sum+A[i]
9           slt     $t8, $s1, $s3   # if i < n
10          bne     $t8, $zero, loop        #if i != n, goto loop
```

+ Step 1: khởi tạo i = 0

+ Step 2: lưu địa chỉ của mảng

+ Step 3: lưu giá trị của n

+ Step 4: lưu bước nhảy của i

+ Step 5: khởi tạo sum = A[0]

+ Step 6: tăng i

+ step 7, 8: t1 = 4s1

+ Step 9: t1 lưu địa chỉ của $s2

+ Step 10: load dữ liệu của A[i] vào t0

+ Step 11: s5 = s5 + t0

+ Step 12: So sánh nếu i < n thì t8 = 1, ngược lại bằng 0

+ Step 13: Nếu t8 != 0 thì quay lại vòng lặp

b. i <= n

```
1   .text
2           li      $s1, 0          # gán i = 0
3
4   loop:   add     $s1, $s1, $s4   #i=i+step
5           add     $t1, $s1, $s1   #t1=2*s1
6           add     $t1, $t1, $t1   #t1=4*s1
7           add     $t1, $t1, $s2   #t1 store the address of A[i]
8           lw      $t0, 0($t1)     #load value of A[i] in $t0
9           add     $s5, $s5, $t0   #sum=sum+A[i]
10          slt     $t8, $s3, $s1   # if i <= n
11          beq     $t8, $zero, loop        #if i != n, goto loop
```

+ Step 1: khởi tạo i = 0

+ Step 2: lưu địa chỉ của mảng

+ Step 3: lưu giá trị của n

+ Step 4: lưu bước nhảy của i

+ Step 5: khởi tạo sum = A[0]

+ Step 6: tăng i

+ step 7, 8: t1 = 4s1

+ Step 9: t1 lưu địa chỉ của $s2

+ Step 10: load dữ liệu của A[i] vào t0

+ Step 11: s5 = s5 + t0

+ Step 12: So sánh nếu i <= n thì t8 = 1, ngược lại bằng 0

+ Step 13: Nếu t8 != 0 thì quay lại vòng lặp

c. sum >= 0

```
1   .data
2   A: .space 16    # Khai báo mảng A có 4 phần tử
3
4   .text
5           li      $s1, -1                 # gán i = -1
6           li      $s5, 0                  # sum = 0
7
8   loop:   add     $s1, $s1, $s4   #i=i+step
9           add     $t1, $s1, $s1   #t1=2*s1
10          add     $t1, $t1, $t1   #t1=4*s1
11          add     $t1, $t1, $s2   #t1 store the address of A[i]
12          lw      $t0, 0($t1)     #load value of A[i] in $t0
13          add     $s5, $s5, $t0   #sum=sum+A[i]
14          sge     $t8, $s5, $zero # if 0 <= sum
15          bne     $t8, $zero, loop        #if i != n, goto loop
```

+ Step 1: khởi tạo i = -1
+ Step 2: lưu địa chỉ của mảng
+ Step 3: lưu giá trị của n
+ Step 4: lưu bước nhảy của i
+ Step 5: khởi tạo sum = A[0]
+ Step 6: tăng i
+ step 7, 14: t1 = 4s1
+ Step 8: t1 lưu địa chỉ của $s2
+ Step 9: load dữ liệu của A[i] vào t0
+ Step 10: s5 = s5 + t0
+ Step 11: So sánh nếu sum >= 0 thì t8 = 1, ngược lại bằng 0
+ Step 12: Nếu t8 != 0 thì quay lại vòng lặp

d. A[i] == 0

```
4   .text
5           li      $s1, -1                 # gán i = -1
6           li      $s5, 0                  # sum = 0
7           li      $s4, 1                  # step = 1
8
9   loop:   add     $s1, $s1, $s4   #i=i+step
10          add     $t1, $s1, $s1   #t1=2*s1
11          add     $t1, $t1, $t1   #t1=4*s1
12          add     $t1, $t1, $s2   #t1 store the address of A[i]
13          lw      $t0, 0($t1)     #load value of A[i] in $t0
14          add     $s5, $s5, $t0   #sum=sum+A[i]
15          seq     $t8, $t0, $zero # if A[i] == 0
16          bne     $t8, $zero, loop        #if i != n, goto loop
```

+ Step 1: khởi tạo i = -1
+ Step 2: lưu địa chỉ của mảng
+ Step 3: lưu giá trị của n
+ Step 4: lưu bước nhảy của i
+ Step 5: khởi tạo sum = 0
+ Step 6: tăng i
+ step 7: t1 = 4s1
+ Step 8: t1 lưu địa chỉ của $s2
+ Step 9: load dữ liệu của A[i] vào t0
+ Step 10: s5 = s5 + t0
+ Step 11: Nếu t0 == 0 thì quay lại vòng lặp

Assignment 6

```
.data
A: .word -5, -4, 4528, 15        #Khai bao mang A co 4 phan tu

.text
        li      $s1, -1          #gan i = -1
        la      $s2, A           # $2 = &A
        li      $s3, 4           # n = 4
        li      $s4, 1           # step  = 1
        li      $s5, 0           # max = 0

loop:
        add     $s1,$s1,$s4      #i=i+step
        add     $t1,$s1,$s1      #t1=2*s1
        add     $t1,$t1,$t1      #t1=4*s1
        add     $t1,$t1,$s2      #t1 store the address of A[i]
```

```
          add     $s1,$s1,$s4      #i=i+step
          add     $t1,$s1,$s1      #t1=2*s1
          add     $t1,$t1,$t1      #t1=4*s1
          add     $t1,$t1,$s2      #t1 store the address of A[i]
          lw      $t0,0($t1)       #load value of A[i] in$t0
          abs     $t8,$t0
          sgt     $t7,$t8,$s5      # if max > A[i]
          bne     $t7,$zero,else   #lap lai
          j       endif

else :    add     $s5,$zero,$t8

endif :   bne     $s1,$s3,loop
```

+ Step 1: khởi tạo i = -1

+ Step 2: lưu địa chỉ của mảng vào $s2

+ Step 3: lưu giá trị của n =4

+ Step 4: lưu bước nhảy của i

+ Step 5: khởi tạo max = 0

+ Step 6: tăng i

+ step 7: t1 = 4s1

+ Step 6: t1 lưu địa chỉ của $s2

+ Step 9: load dữ liệu của A[i] vào t0

+ Step 10: Lưu giá trị tuyệt đối của t0 vào t8

+ Step 11: Nếu t7 > s5 thì t8 = 1

+ Step 12: Nếu t7 != 0 thì rẽ nhánh đến else

+ Step 13: Khi rẽ đến else thì gán t8 cho t5

+ step 14: đóng if

+Step 15:Khi i!= n thì thực hiện tiếp tục vòng lặp

⇨ **Kết quả:** $s5 =4528

Registers | Coproc 1 | Coproc 0

Name	Number	Value
$zero	0	0
$at	1	0
$v0	2	0
$v1	3	0
$a0	4	0
$a1	5	0
$a2	6	0
$a3	7	0
$t0	8	0
$t1	9	268501008
$t2	10	0
$t3	11	0
$t4	12	0
$t5	13	0
$t6	14	0
$t7	15	0
$s0	16	0
$s1	17	4
$s2	18	268500992
$s3	19	4
$s4	20	1
$s5	21	4528
$s6	22	0
$s7	23	0
$t8	24	0
$t9	25	0
$k0	26	0
$k1	27	0
$gp	28	268468224
$sp	29	2147479548
$fp	30	0
$ra	31	0
pc		4194380
hi		0
lo		0