

Laboratory Exercise 10.1 – Report:

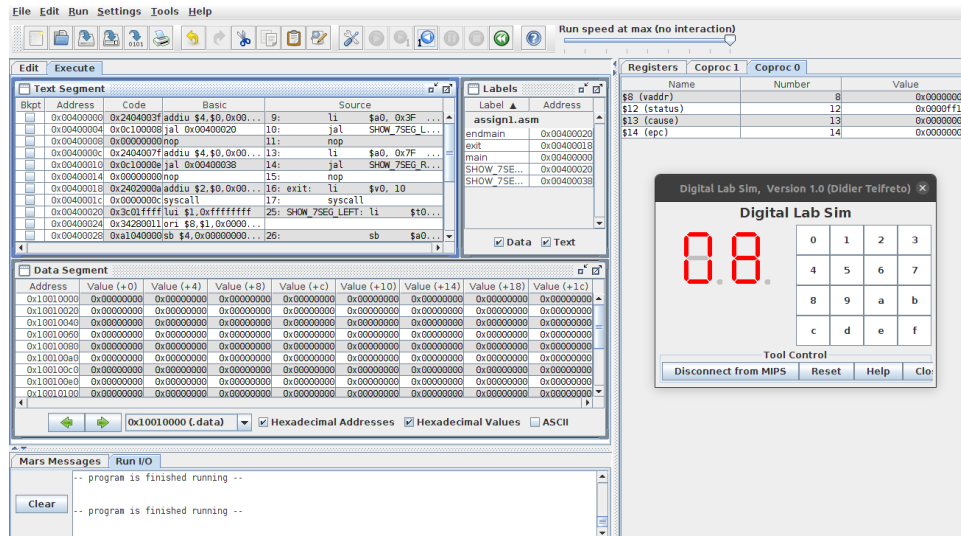
Lê Văn Duẩn - 20194508

1. Assignment 1

- Mã nguồn:

```
assign1.asm  assign2.asm
1  #Laboratory Exercise 10 Home Assignment 1
2  .eqv SEVENSEG_LEFT 0xFFFF0011      # Địa chỉ của đèn led 7 đoạn trái.
3                                     # Bit 0 = đoạn a;
4                                     # Bit 1 = đoạn b; ...
5                                     # Bit 7 = dấu .
6  .eqv SEVENSEG_RIGHT 0xFFFF0010    # Địa chỉ của đèn led 7 đoạn phải
7  .text
8  main:
9      li    $a0, 0x3F                # set value for segments
10     jal   SHOW_7SEG_LEFT           # show
11     nop
12
13     li    $a0, 0x7F                # set value for segments
14     jal   SHOW_7SEG_RIGHT          # show
15     nop
16 exit:  li    $v0, 10
17     syscall
18 endmain:
19
20 #-----
21 # Function SHOW_7SEG_LEFT
22 # param(in)  $a0 value to shown
23 # remark     $t0 changed
24 #-----
25 SHOW_7SEG_LEFT: li    $t0, SEVENSEG_LEFT    # assign port's address
26                 sb    $a0, 0($t0)           # assign new value
27                 nop
28                 jr    $ra
29                 nop
30
31 #-----
32 # Function SHOW_7SEG_RIGHT: turn on/off the 7seg
33 # param(in)  $a0 value to shown
34 # remark     $t0 changed
35 #-----
36 SHOW_7SEG_RIGHT: li    $t0, SEVENSEG_RIGHT  # assign port's adress
37                 sb    $a0, 0($t0)           # assign new value
38                 nop
39                 jr    $ra
```

- Kết quả chạy mô phỏng:

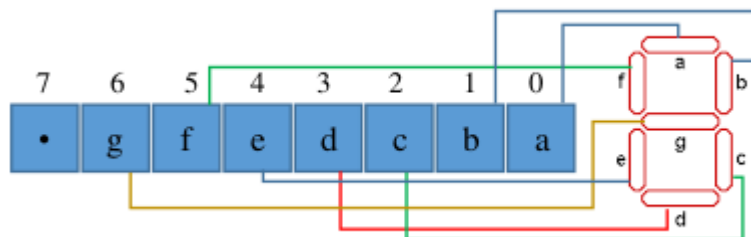


- Giải thích:

SEVENSEG_LEFT và SEVENSEG_RIGHT đều đã có địa chỉ đặt sẵn tùy theo nhà sản xuất chip

SHOW_7SEG_LEFT và SHOW_7SEG_RIGHT là các hàm hiển thị số ra theo hiển thị 7 thanh.

\$a0 là mã nhị phân của số mình muốn hiển thị 7 thanh ví dụ:



Như trong ví dụ hiển thị số 0: thì những thanh ghi được bật sẽ là a, b, c, d, e, f và vì không hiển thị dấu chấm nên bit thứ 7 và bit g(bit thứ 6) sẽ tắt

-> mã nhị phân: 00111111 = 0x3F

2. Assignment 2

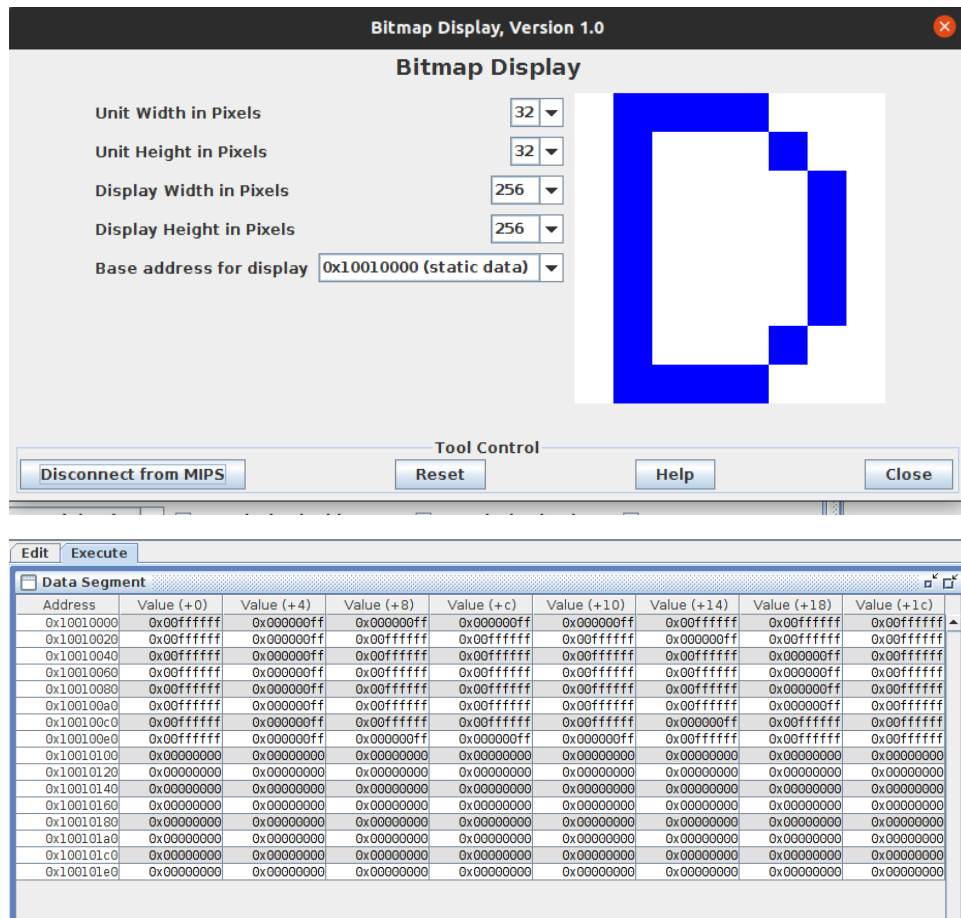
- Mã nguồn:

```

assign1.asm  assign2.asm
1  #Laboratory Exercise 10 Home Assignment 2
2  .eqv MONITOR_SCREEN 0x10010000 # địa chỉ bắt đầu của bộ nhớ màn hình
3  .eqv RED 0x00FF0000 # địa chỉ bắt đầu của màu đỏ
4  .eqv GREEN 0x0000FF00 # địa chỉ bắt đầu của màu xanh
5  .eqv BLUE 0x000000FF # địa chỉ bắt đầu của màu xanh
6  .eqv WHITE 0x00FFFFFF
7  .eqv YELLOW 0x00FFFF00
8
9  .data
10 arr1 : .word 0x00FFFFFF, 0x000000FF, 0x000000FF, 0x000000FF, 0x000000FF, 0x00FFFFFF, 0x00FFFFFF, 0x00FFFFFF
11 arr2 : .word 0x00FFFFFF, 0x000000FF, 0x00FFFFFF, 0x00FFFFFF, 0x00FFFFFF, 0x000000FF, 0x00FFFFFF, 0x00FFFFFF
12 arr3 : .word 0x00FFFFFF, 0x000000FF, 0x00FFFFFF, 0x00FFFFFF, 0x00FFFFFF, 0x00FFFFFF, 0x000000FF, 0x00FFFFFF
13 arr4 : .word 0x00FFFFFF, 0x000000FF, 0x00FFFFFF, 0x00FFFFFF, 0x00FFFFFF, 0x00FFFFFF, 0x000000FF, 0x00FFFFFF
14 arr5 : .word 0x00FFFFFF, 0x000000FF, 0x00FFFFFF, 0x00FFFFFF, 0x00FFFFFF, 0x00FFFFFF, 0x000000FF, 0x00FFFFFF
15 arr6 : .word 0x00FFFFFF, 0x000000FF, 0x00FFFFFF, 0x00FFFFFF, 0x00FFFFFF, 0x00FFFFFF, 0x000000FF, 0x00FFFFFF
16 arr7 : .word 0x00FFFFFF, 0x000000FF, 0x00FFFFFF, 0x00FFFFFF, 0x00FFFFFF, 0x000000FF, 0x00FFFFFF, 0x00FFFFFF
17 arr8 : .word 0x00FFFFFF, 0x000000FF, 0x00FFFFFF, 0x00FFFFFF, 0x00FFFFFF, 0x000000FF, 0x00FFFFFF, 0x00FFFFFF
18
19 .text
20 main: li $k0, MONITOR_SCREEN # nạp địa chỉ bắt đầu của màn hình
21       li $t2, 0
22       la $a0, arr1
23       jal print_line
24       nop
25
26       la $a0, arr2
27       jal print_line
28       nop
29
30       la $a0, arr3
31       jal print_line
32       nop
33
34       la $a0, arr4
35       jal print_line
36       nop
37
38       la $a0, arr5
39       jal print_line
40       nop
41
42       la $a0, arr6
43       jal print_line
44       nop
45
46       la $a0, arr7
47       jal print_line
48       nop
49
50       la $a0, arr8
51       jal print_line
52       nop
53
54 end_main: li $v0, 10
55           syscall
56
57
58 print_line: li $t1, 0
59 loop: beq $t1, 32, end
60
61       add $t3, $k0, $t2 # $k0 + t2*4 = $t3
62       add $t4, $a0, $t1 # $a0 + t1*4 = $t4
63
64       lw $t0, 0($t4)
65
66       sw $t0, 0($t3)
67       nop
68
69       addi $t1, $t1, 4
70       addi $t2, $t2, 4
71       j loop
72 end: jr $ra
73

```

- Kết quả chạy:



- Giải thích:

Hiển thị theo pixel trên màn hình 256x256 và mỗi pixel là 32x32 nên ta sẽ có 8 hàng và 8 cột

Xây dựng 8 mảng, mỗi mảng 8 phần tử tương ứng để hiển thị lần lượt theo hàng và mỗi phần tử của mảng chính là địa chỉ của màu để in. Ở ví dụ này ta dùng nền trắng 0x00FFFFFF và chữ màu xanh 0x000000FF

Hàm print_line dùng để duyệt mảng lưu ở \$a0 và hiển thị lần lượt ra màn hình thông qua \$k0 bằng các sw địa chỉ lần lượt.

