

Laboratory Exercise 10.1 – Report:

Nguyễn Hải Dương- 20194530

1. Assignment 1

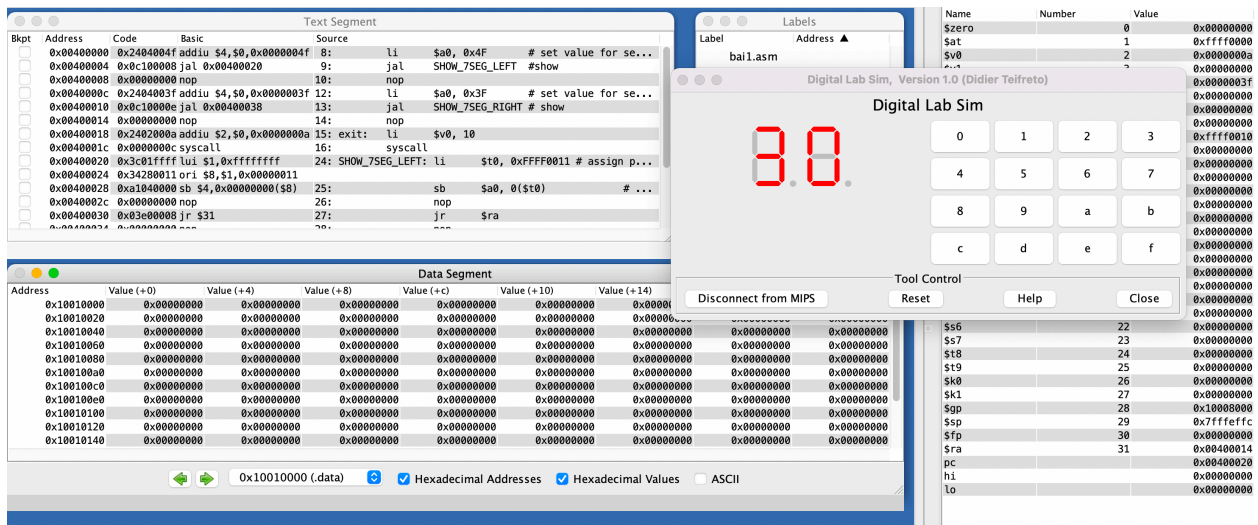
- Mã nguồn:

```

1  .eqv SEVENSEG_LEFT 0xFFFF0011          # Dia chi cua den led 7 doan trai.
2                                          # Bit 0 = doan a;
3                                          # Bit 1 = doan b; ...
4                                          # Bit 7 = dau .
5  .eqv SEVENSEG_RIGHT 0xFFFF0010        # Dia chi cua den led 7 doan phai
6  .text
7  main:
8      li      $a0, 0x4F                  # set value for segments
9      jal     SHOW_7SEG_LEFT             #show
10     nop
11
12     li      $a0, 0x3F                  # set value for segments
13     jal     SHOW_7SEG_RIGHT            # show
14     nop
15 exit:  li      $v0, 10
16     syscall
17 endmain:
18
19 #-----
20 # Function SHOW_7SEG_LEFT
21 # param(in)  $a0 value to shown
22 # remark    $t0 changed
23 #-----
24 SHOW_7SEG_LEFT: li      $t0, SEVENSEG_LEFT    # assign port's address
25                 sb      $a0, 0($t0)          # assign new value
26                 nop
27                 jr      $ra
28                 nop
29
30 #-----
31 # Function SHOW_7SEG_RIGHT:    turn on/off the 7seg
32 # param(in) $a0                value to shown
33 # remark    $t0                changed
34 #-----
35 SHOW_7SEG_RIGHT: li      $t0, SEVENSEG_RIGHT    # assign port's adress
36                 sb      $a0, 0($t0)          # assign new value
37                 nop
38                 jr      $ra
39                 nop
40

```

- Kết quả chạy mô phỏng:

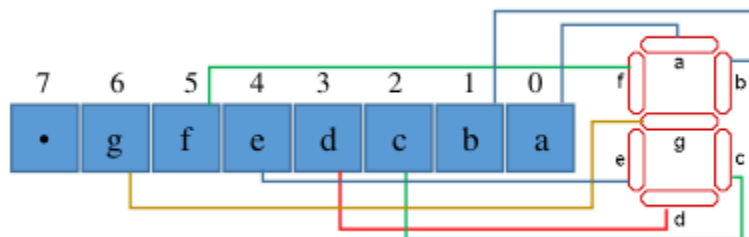


- Giải thích:

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:



Ví dụ, nếu muốn hiển thị số 3: thì những thanh ghi được bật sẽ là a, b, c, d, g và vì không hiển thị dấu chấm nên bit thứ 7 sẽ tắt

-> mã nhị phân: 01001111 = 0x4F

2. Assignment 2

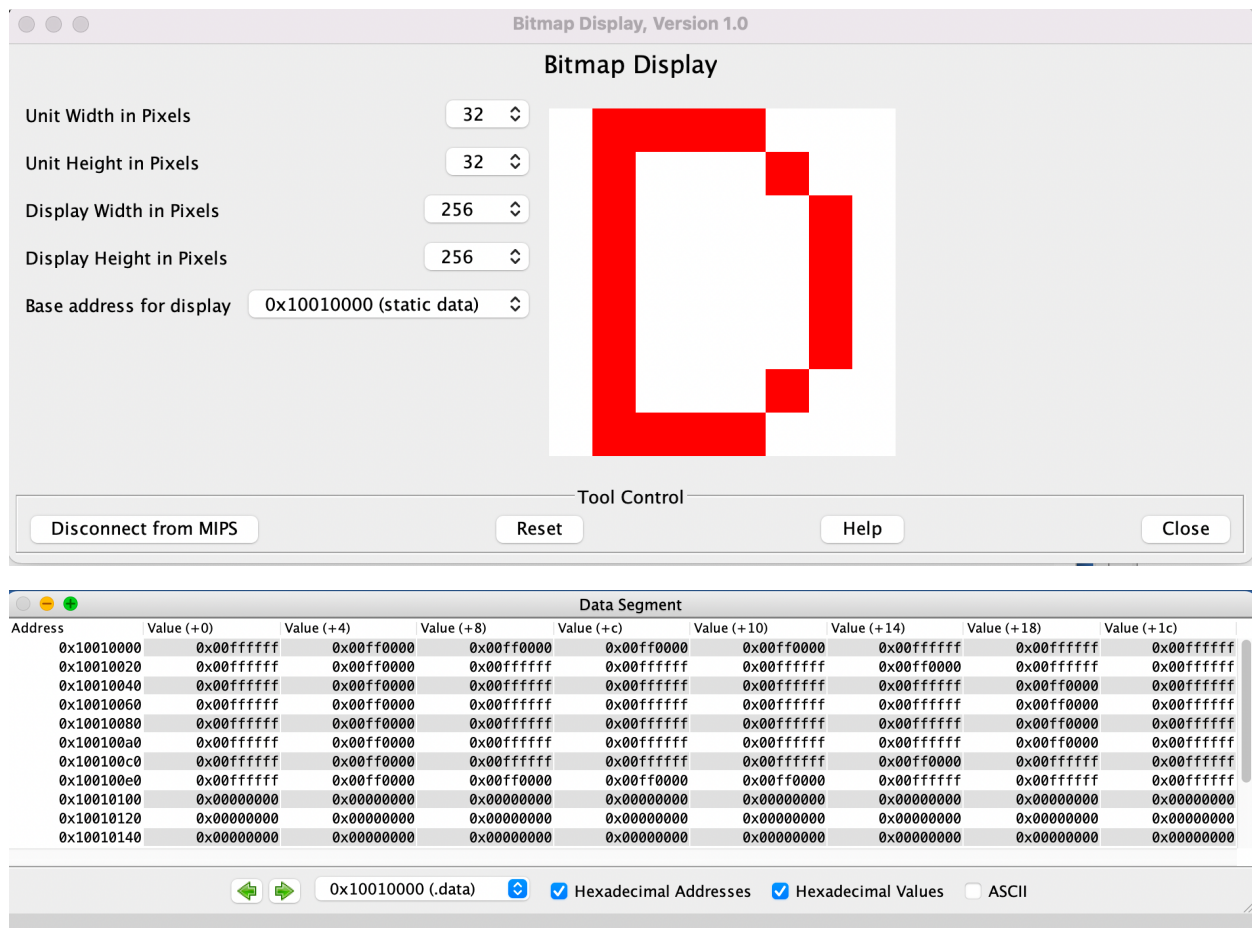
- Mã nguồn:

```

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

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.

