

Họ và tên: Hà Trung Chiến

MSSV: 20225794

### Assignment 1:

*Mã nguồn:*

```
.data
    one_to_nine: .word 0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F,
0x6F
    .eqv SEVENSEG_LEFT 0xFFFF0011 # Địa chỉ của đèn led 7 đoạn trái.
# Bit 0 = đoạn a;
# Bit 1 = đoạn b; ...
# Bit 7 = dấu .
    .eqv SEVENSEG_RIGHT 0xFFFF0010 # Địa chỉ của đèn led 7 đoạn phải
.text
    la $t1, one_to_nine
main:
    li $a2, 0
count_0_to_9:
    sll $a1, $a2, 2
    move $t6, $t1
    add $t6, $t6, $a1
    lw $a0, 0($t6)
    jal SHOW_7SEG_RIGHT
    addi $a2, $a2, 1
    ble $a2, 9, count_0_to_9
    subi $a2, $a2, 2
count_9_to_0:
    sll $a1, $a2, 2
    move $t6, $t1
    add $t6, $t6, $a1
    lw $a0, 0($t6)
    jal SHOW_7SEG_RIGHT
    subi $a2, $a2, 1
    bge $a2, 1, count_9_to_0
    j count_0_to_9

endmain:
#-----
# Function SHOW_7SEG_LEFT : turn on/off the 7seg
# param[in] $a0 value to shown
# remark $t0 changed
```

#-----

SHOW\_7SEG\_LEFT:

```
li $t0, SEVENSEG_LEFT # assign port's address
sb $a0, 0($t0) # assign new value
jr $ra
```

#-----

# Function SHOW\_7SEG\_RIGHT : turn on/off the 7seg

# param[in] \$a0 value to shown

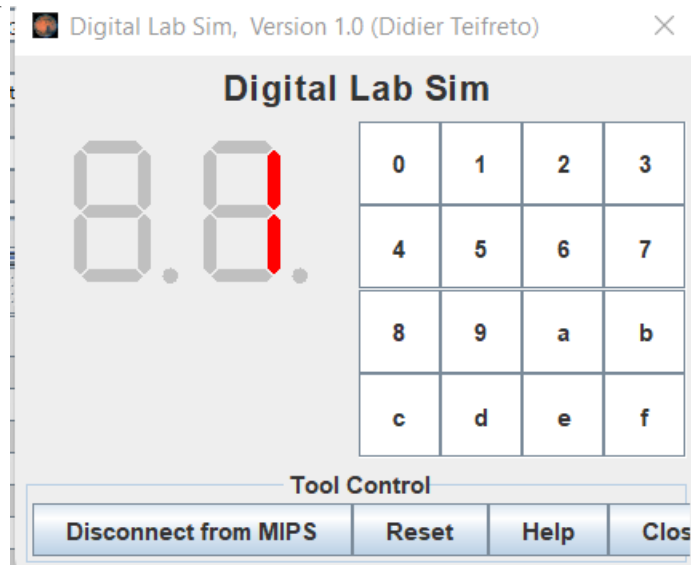
# remark \$t0 changed

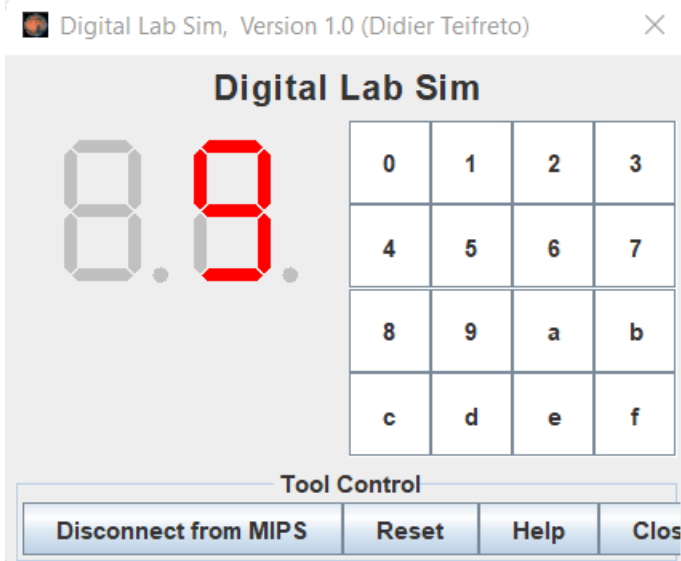
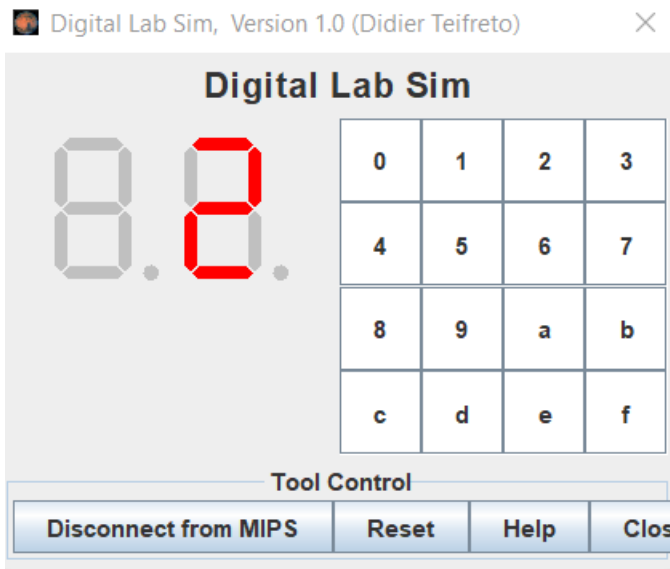
#-----

SHOW\_7SEG\_RIGHT:

```
li $t0, SEVENSEG_RIGHT # assign port's address
sb $a0, 0($t0) # assign new value
jr $ra
```

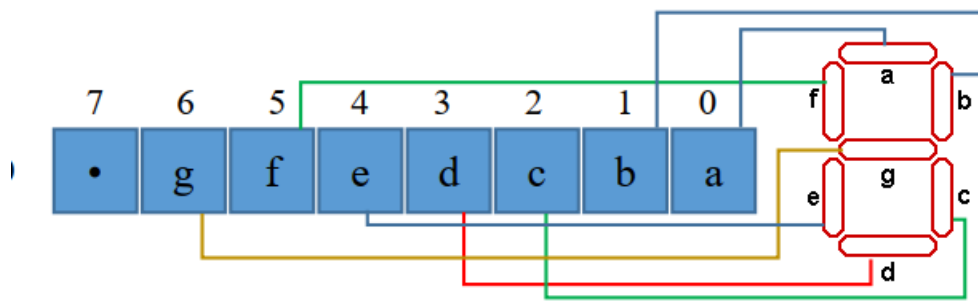
**Kết quả:**





**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 lưu giá trị mã nhị phân của số mình muốn hiển thị trên LED 7 thanh



- Ví dụ: nếu muốn 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 sẽ tắt  
→ giá trị thanh ghi \$a0: 00111111 = 0x3F

Ta có bảng giá trị mã nhị phân của các số 0 – 9:

Digit	.	g	f	e	d	c	b	a	Hex
0	0	0	1	1	1	1	1	1	0x3F
1	0	0	0	0	0	1	1	0	0X06
2	0	1	0	1	1	0	1	1	0X5B
3	0	1	0	0	1	1	1	1	0X4F
4	0	1	1	0	0	1	1	0	0X66
5	0	1	0	1	1	0	1	1	0X6D
6	0	0	1	1	1	1	0	1	0X7D
7	0	1	0	0	0	1	1	1	0X07
8	0	1	1	1	1	1	1	1	0X7F
9	0	1	1	0	0	1	1	1	0X6F

## Assignment 2:

*Mã nguồn:*

```
.eqv MONITOR_SCREEN, 0x10010000 #Địa chỉ bắt đầu của bộ nhớ màn hình
```

```
.eqv RED, 0x00FF0000 #Các giá trị màu thường sử dụng
```

```
.eqv BLACK, 0x00000000
```

```
.text
```

```
li $k0, MONITOR_SCREEN #Nạp địa chỉ bắt đầu của màn hình
```

```
li $a1, 7
```

```
move_square:
```

```
blt $a1, $zero, exit #so sánh nếu đã ở vị trí ngoài cùng bên phải thì dừng lại
```

```
sll $t1, $a1, 2 #
```

```
move $t2, $k0
```

```
add $t2, $t2, $t1 #lấy địa chỉ của vị trí ô vuông hiện tại
```

```
jal print_red
```

```

        jal delete_red
        subi $a1, $a1, 1
        j move_square
print_red:                #in ô đỏ
        li $t0, RED
        sw $t0, 0($t2)
        nop
        jr $ra
        nop
delete_red:               #xóa ô đỏ
        li $t0, BLACK
        sw $t0, 0($t2)
        nop
        jr $ra
        nop
exit:
        li $v0, 10
        syscall

```

### Giải thích:

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

ta sử dụng 2 hàm print\_square và delete\_square kết hợp với nhau để di chuyển ô vuông từ phải sang trái. Khởi đầu từ vị trí ngoài cũng ở hàng thứ nhất với \$a1 = 7 sau đó giảm dần \$a1 đi 1 tương đương với việc dịch ô vuông màu đỏ sang phải

### Kết quả:

