# Lab 8

Đinh Ngọc Khánh Huyền – 20225726

## Assigment 1

**Code**

```
.eqv SEVENSEG_LEFT 0xFFFF0011 # Address of the left 7-segment display
.eqv SEVENSEG_RIGHT 0xFFFF0010 # Address of the right 7-segment display

.text
main:
    # Display 0-9, A-F on the left 7-segment display
    li $a0, 0x3F # Display 0
    jal SHOW_7SEG_LEFT
    nop

    li $a0, 0x06 # Display 1
    jal SHOW_7SEG_LEFT
    nop

    li $a0, 0x5B # Display 2
    jal SHOW_7SEG_LEFT
    nop

    li $a0, 0x4F # Display 3
    jal SHOW_7SEG_LEFT
    nop
```

```
li $a0, 0x66 # Display 4
jal SHOW_7SEG_LEFT
nop

li $a0, 0x6D # Display 5
jal SHOW_7SEG_LEFT
nop

li $a0, 0x7D # Display 6
jal SHOW_7SEG_LEFT
nop

li $a0, 0x07 # Display 7
jal SHOW_7SEG_LEFT
nop

li $a0, 0x7F # Display 8
jal SHOW_7SEG_LEFT
nop

li $a0, 0x6F # Display 9
jal SHOW_7SEG_LEFT
nop

li $a0, 0x77 # Display A
```

```
jal SHOW_7SEG_LEFT
nop

li $a0, 0x7C # Display B
jal SHOW_7SEG_LEFT
nop

li $a0, 0x39 # Display C
jal SHOW_7SEG_LEFT
nop

li $a0, 0x5E # Display D
jal SHOW_7SEG_LEFT
nop

li $a0, 0x79 # Display E
jal SHOW_7SEG_LEFT
nop

li $a0, 0x71 # Display F
jal SHOW_7SEG_LEFT
nop

# Display 0-9, A-F on the right 7-segment display
li $a0, 0x3F # Display 0
jal SHOW_7SEG_RIGHT
```

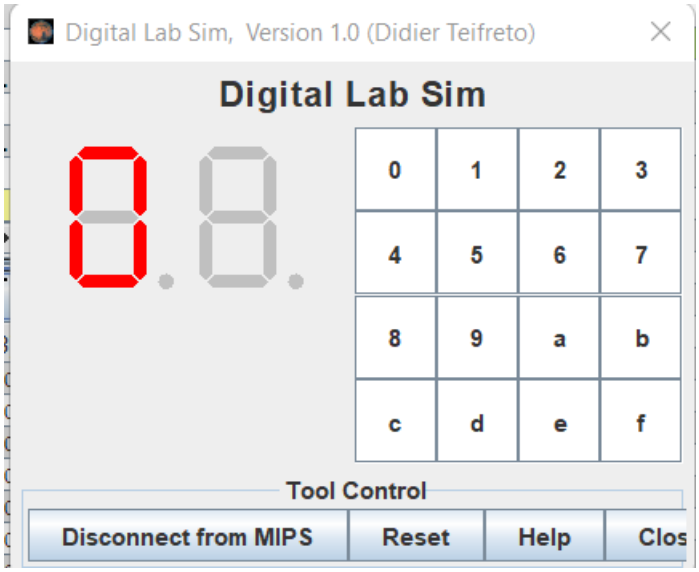
```
nop

li $a0, 0x06 # Display 1
jal SHOW_7SEG_RIGHT
nop

li $a0, 0x5B # Display 2
jal SHOW_7SEG_RIGHT
nop

li $a0, 0x4F # Display 3
jal SHOW_7SEG_RIGHT
nop

li $a0, 0x66 # Display 4
jal SHOW_7SEG_RIGHT
nop

li $a0, 0x6D # Display 5
jal SHOW_7SEG_RIGHT
nop

li $a0, 0x7D # Display 6
jal SHOW_7SEG_RIGHT
nop
```

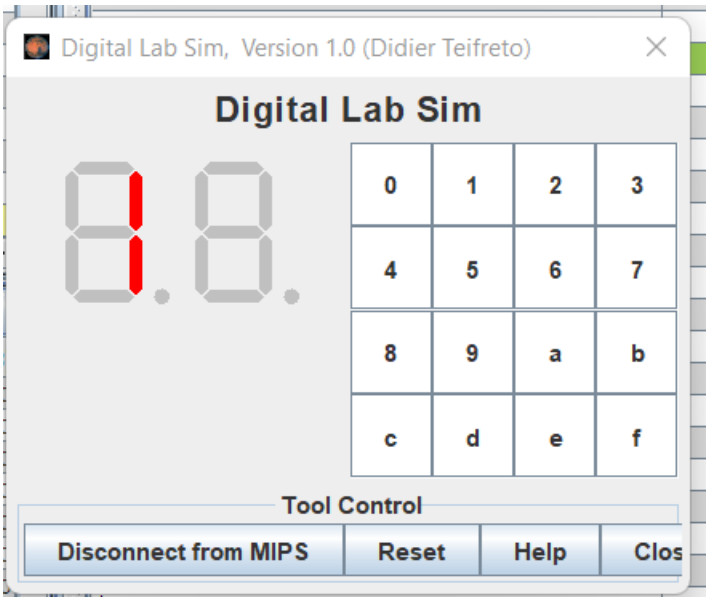
```
li $a0, 0x07 # Display 7
jal SHOW_7SEG_RIGHT
nop

li $a0, 0x7F # Display 8
jal SHOW_7SEG_RIGHT
nop

li $a0, 0x6F # Display 9
jal SHOW_7SEG_RIGHT
nop

li $a0, 0x77 # Display A
jal SHOW_7SEG_RIGHT
nop

li $a0, 0x7C # Display B
jal SHOW_7SEG_RIGHT
nop

li $a0, 0x39 # Display C
jal SHOW_7SEG_RIGHT
nop

li $a0, 0x5E # Display D
jal SHOW_7SEG_RIGHT
```

```
    nop

    li $a0, 0x79 # Display E
    jal SHOW_7SEG_RIGHT
    nop

    li $a0, 0x71 # Display F
    jal SHOW_7SEG_RIGHT
    nop

exit:
    li $v0, 10
    syscall
endmain:

# Function SHOW_7SEG_LEFT: Turn on/off the segments of the left 7-segment display
# param[in] $a0: Value to be shown
# remark $t0 changed
SHOW_7SEG_LEFT:
    li $t0, SEVENSEG_LEFT # Load the address of port
    sb $a0, 0($t0) # Set the value of the LED segments
    nop
    jr $ra
    nop
```

```
# Function SHOW_7SEG_RIGHT: Turn on/off the segments of the right 7-segment display
# param[in] $a0: Value to be shown
# remark $t0 changed
SHOW_7SEG_RIGHT:
    li $t0, SEVENSEG_RIGHT # Load the address of port
    sb $a0, 0($t0) # Set the value of the LED segments
    nop
    jr $ra
    nop
```

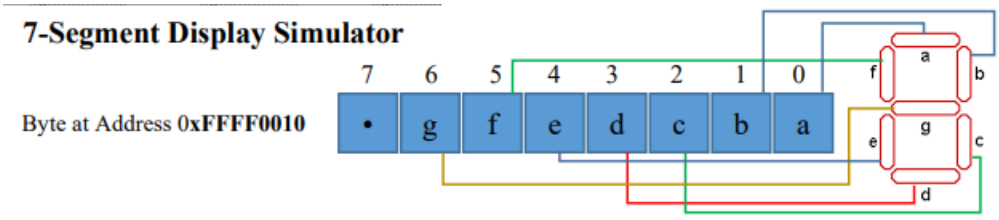
**Result**






Tương tự với các giá trị còn lại.

**Explain:**




Giả sử muốn hiển thị giá trị 0,

Bit 6: Đoạn a (bật)

Bit 5: Đoạn b (bật)

Bit 4: Đoạn c (bật)

Bit 3: Đoạn d (bật)

Bit 2: Đoạn e (bật)

Bit 1: Đoạn f (bật)

Bit 0: Đoạn g (tắt)

$0111111_2 = 0x3F$

⇨ Vì vậy, với giá trị 0x3F, các đoạn a, b, c, d, e và f sẽ được bật, còn đoạn g sẽ được tắt, tạo ra chữ số 0 trên màn hình LED 7 thanh.

## Assigment 2

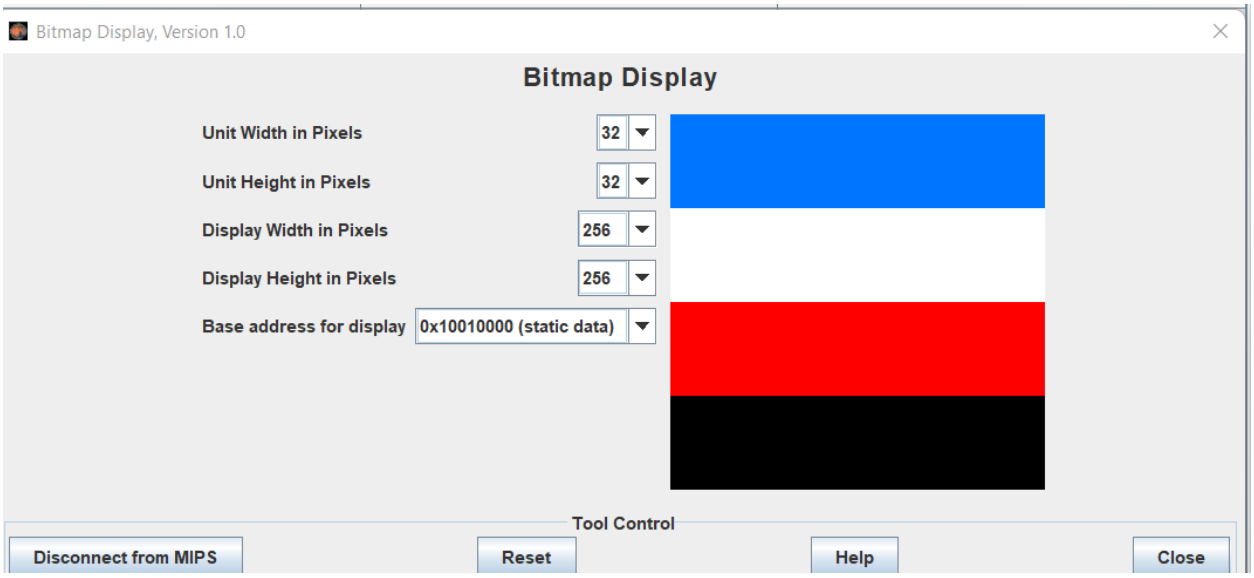
**Code**

```
.eqv MONITOR_SCREEN 0x10010000 # Địa chỉ bắt đầu của bộ nhớ màn hình
.eqv BLUE 0x000075FF # Màu xanh
.eqv WHITE 0xFFFFFFFF # Màu trắng
.eqv RED 0x00FF0000 # Màu đỏ

.text
main:
    li $k0, MONITOR_SCREEN # Load địa chỉ bắt đầu của bộ nhớ màn hình

    # Vẽ phần màu xanh của lá cờ Pháp
    li $t0, BLUE
    sw $t0, 0($k0) # Dòng đầu tiên
    sw $t0, 4($k0)
    sw $t0, 8($k0)
    sw $t0, 12($k0)
    sw $t0, 16($k0)
    sw $t0, 20($k0)
    sw $t0, 24($k0)
    sw $t0, 28($k0)
    sw $t0, 32($k0)
```

```
sw $t0, 36($k0)
sw $t0, 40($k0)
sw $t0, 44($k0)
sw $t0, 48($k0)
sw $t0, 52($k0)
sw $t0, 56($k0)
sw $t0, 60($k0)

# Vẽ phần màu trắng của lá cờ Pháp
li $t0, WHITE
sw $t0, 64($k0)
sw $t0, 68($k0)
sw $t0, 72($k0)
sw $t0, 76($k0)
sw $t0, 80($k0)
sw $t0, 84($k0)
sw $t0, 88($k0)
sw $t0, 92($k0)
sw $t0, 96($k0)
sw $t0, 100($k0)
sw $t0, 104($k0)
sw $t0, 108($k0)
sw $t0, 112($k0)
sw $t0, 116($k0)
sw $t0, 120($k0)
sw $t0, 124($k0)
```

```
    # Vẽ phần màu đỏ của lá cờ Pháp
    li $t0, RED
    sw $t0, 128($k0)
    sw $t0, 132($k0)
    sw $t0, 136($k0)
    sw $t0, 140($k0)
    sw $t0, 144($k0)
    sw $t0, 148($k0)
    sw $t0, 152($k0)
    sw $t0, 156($k0)
    sw $t0, 160($k0)
    sw $t0, 164($k0)
    sw $t0, 168($k0)
    sw $t0, 172($k0)
    sw $t0, 176($k0)
    sw $t0, 180($k0)
    sw $t0, 184($k0)
    sw $t0, 188($k0)

exit:
    li $v0, 10 # Kết thúc chương trình
    syscall
endmain:
```

**Result**




**Explain:**

Kết quả đúng với lý thuyết

**Assigment 3**

**Code**

```
.eqv HEADING 0xffff8010 # Integer: An angle between 0 and 359
 # 0 : North (up)
# 90: East (right)
 # 180: South (down)
 # 270: West (left)
.eqv MOVING 0xffff8050 # Boolean: whether or not to move
.eqv LEAVETRACK 0xffff8020 # Boolean (0 or non-0):
 # whether or not to leave a track
.eqv WHEREX 0xffff8030 # Integer: Current x-location of MarsBot
.eqv WHEREY 0xffff8040 # Integer: Current y-location of MarsBot
.text
```

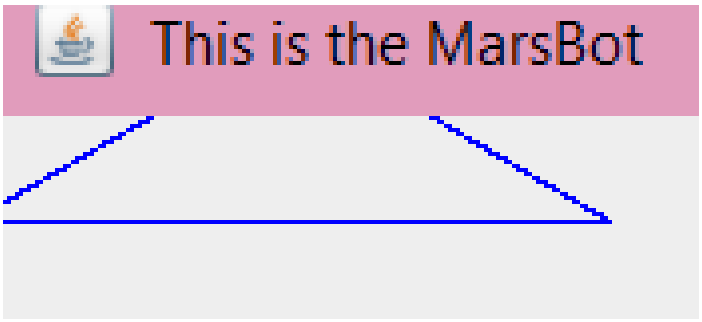
```
main: jal draw_track # draw track line
 nop
 addi $a0, $zero, 90 # Marsbot rotates 90* and start running
 jal rotate_marsbot
 nop
 jal start_moving
 nop

sleep1: addi $v0,$zero,32 # Keep running by sleeping in 4000 ms
 li $a0,4000
 syscall

 jal stop_tracking # keep old track
 nop
 jal draw_track # and draw new track line
 nop

goDOWN: addi $a0, $zero, 120 # Marsbot rotates 180 degrees
 jal rotate_marsbot
 nop

sleep2: addi $v0,$zero,32 # Keep running by sleeping in 2000 ms
 li $a0,2000
 syscall
 jal stop_tracking # keep old track
 nop
```

```
 jal draw_track # and draw new track line
 nop

goLEFT: addi $a0, $zero, 270 # Marsbot rotates 270 degrees
 jal rotate_marsbot
 nop

sleep3: addi $v0,$zero,32 # Keep running by sleeping in 6000 ms
 li $a0,6000
 syscall
 jal stop_tracking # keep old track
 nop
 jal draw_track # and draw new track line
 nop

goASKEW: addi $a0, $zero, 60 # Marsbot rotates 120 degrees
 jal rotate_marsbot
 nop
 jal stop_tracking # keep old track
 nop
 jal draw_track # and draw new track line
 nop

end_main:

#-----------------------------------------------------------
```

```
# start_moving: to start running
# param[in] none
#-------------------------------------------------------
start_moving: li $at, MOVING # change MOVING port
 addi $k0, $zero,1 # to logic 1,
 sb $k0, 0($at) # to start running
 nop
 jr $ra
 nop
#-------------------------------------------------------
# STOP: to stop running
# param[in] none
#-------------------------------------------------------
STOP: li $at, MOVING # change MOVING port to 0
 sb $zero, 0($at) # to stop
 nop
 jr $ra
 nop
#-------------------------------------------------------
# draw_track: to start drawing line
# param[in] none
#-------------------------------------------------------
draw_track: li $at, LEAVETRACK # change LEAVETRACK port
 addi $k0, $zero,1 # to logic 1,
 sb $k0, 0($at) # to start tracking
 nop
```

```
 jr $ra
 nop
#-----------------------------------------------------------
# stop_tracking: to stop drawing line
# param[in] none
#-----------------------------------------------------------
stop_tracking: li $at, LEAVETRACK # change LEAVETRACK port to 0
 sb $zero, 0($at) # to stop drawing tail
 nop
 jr $ra
 nop
#-----------------------------------------------------------
# rotate_marsbot: to rotate the robot
# param[in] $a0, An angle between 0 and 359
# 0 : North (up)
# 90: East (right)
# 180: South (down)
# 270: West (left)
#-----------------------------------------------------------
rotate_marsbot: li $at, HEADING # change HEADING port
 sw $a0, 0($at) # to rotate robot
 nop
 jr $ra
 nop
```

**Result**




**Explain**

Chương trình trên vẽ hình thang theo cách lệnh đúng lý thuyết.