

Assignment 1 - LED PORT

Mã nguồn:

```
.eqv SEVENSEG_LEFT 0xFFFF0011 # Dia chi cua den led 7 doan trai.
# Bit 0 = doan a;
# Bit 1 = doan b; ...
# Bit 7 = dau .
.eqv SEVENSEG_RIGHT 0xFFFF0010 # Dia chi cua den led 7 doan phai

.text
main:
    li $a0, 0x7f # set value for segments
    jal SHOW_7SEG_LEFT # show
    nop
    li $a0, 0x66 # set value for segments
    jal SHOW_7SEG_RIGHT # show
    nop
exit: li $v0, 10
     syscall
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
                nop
                jr $ra
                nop
#-----

# 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
                 nop
                 jr $ra
                 nop
```

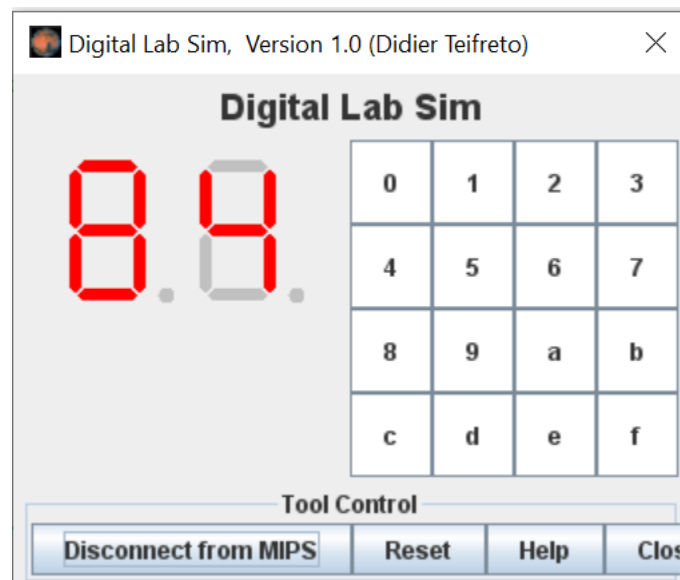
Giải thích:

- Hiển thị 2 số cuối MSSV (8 và 4).
 - Muốn hiển thị số 8, chúng ta cho đèn sáng các thanh a-b-c-d-e-f-g. Chuyển sang mã nhị phân 8 bit ta được $0111\ 1111 = 0x3F$.
Thay vào chương trình:

```
li $a0, 0x7f # set value for segments
jal SHOW_7SEG_LEFT # show
```

- Muốn hiển thị số 4, chúng ta cho đèn sáng các thanh b-c-f-g. Chuyển sang mã nhị phân 8 bit ta được $0110\ 0110 = 0x66$.
Thay vào chương trình:

```
li $a0, 0x66 # set value for segments
jal SHOW_7SEG_RIGHT # show
```

Kết quả thực hiện:

Assignment 2 - BITMAP DISPLAY

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 GREEN 0x0000FF00
.eqv BLUE 0x000000FF
.eqv WHITE 0x0FFFFFFF
.eqv YELLOW 0x00FFFF00
.text
li $k0, MONITOR_SCREEN #Nạp địa chỉ bắt đầu của màn hình

li $s1, 0
li $s2, 252
li $s3, 4
li $s4, WHITE
jal paint    # Background = WHITE
nop

b1:
li $s1, 40
li $s2, 212
li $s3, 32
li $s4, RED
jal paint
nop

b2:
li $s1, 40
li $s2, 52
li $s3, 4
li $s4, RED
jal paint
nop

b3:
li $s1, 52
li $s2, 212
li $s3, 32
li $s4, RED
jal paint
nop

b4:
li $s1, 200
li $s2, 212
li $s3, 4
li $s4, RED
jal paint
nop

b5:
```

```
li $s1, 52
li $s2, 56
li $s3, 4
li $s4, WHITE
jal paint
nop

b6:
li $s1, 212
li $s2, 214
li $s3, 4
li $s4, WHITE
jal paint
nop

b7:
li $s1, 108
li $s2, 112
li $s3, 4
li $s4, RED
jal paint
nop
j end

paint:
add $t2, $zero, $s1 # $s1 = begin
loop:
bgt $t2, $s2, endloop # $s2 = end
nop
add $t1, $k0, $t2 # $t1 = address of place
add $t0, $zero, $s4 # $s4 = COLOR
sw $t0, 0($t1)
add $t2, $t2, $s3 # $s3 = step
j loop
nop
endloop:
jr $ra
end:
```

Giải thích:

- Mục tiêu của chương trình là in ra chữ B.
- Chương trình con *paint* thực hiện tô màu cho các vùng địa chỉ được chỉ định (mỗi pixel cách nhau 4 bit địa chỉ)
Nhận đầu vào là 4 giá trị (địa chỉ tương đối với \$k0):
 - \$s1: Địa chỉ ô nhớ bắt đầu
 - \$s2: Địa chỉ ô nhớ kết thúc
 - \$s3: bước nhảy - step (Nếu tô theo hàng ngang thì $\text{step} = 4$, nếu tô theo hàng dọc thì $\text{step} = 4 * 8 = 32$)
 - \$s4: Giá trị màu cần tô.

```

paint:
add $t2, $zero, $s1           # $s1 = begin
loop:
bgt $t2, $s2, endloop         # $s2 = end
nop
add $t1, $k0, $t2             # $t1 = address of place
add $t0, $zero, $s4           # $s4 = COLOR
sw $t0, 0($t1)
add $t2, $t2, $s3             # $s3 = step
j loop
nop
endloop:
jr $ra

```

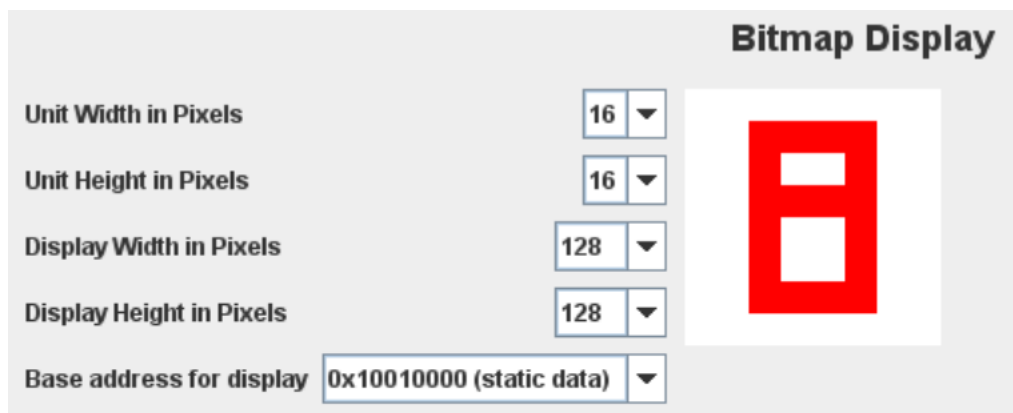
- Ban đầu, chúng ta tô màu toàn bộ background bằng nền trắng

```

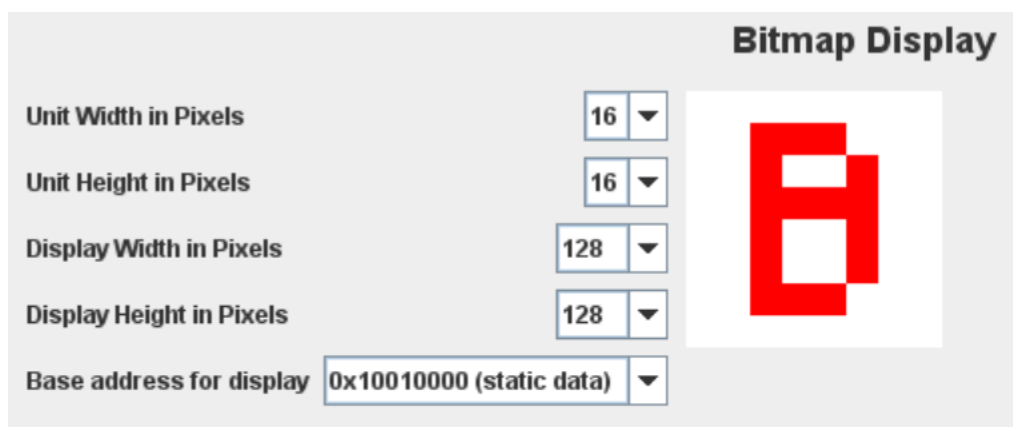
li $s1, 0
li $s2, 252
li $s3, 4
li $s4, WHITE
jal paint           # Background = WHITE
nop

```

- Từ các bước *b1* đến *b5*, chúng ta lần lượt dùng hàm *paint* để tô màu ra số 8 trên bảng BITMAP.



- Các bước *b6* và *b7*, hoàn thành chữ B bằng cách tô màu trắng 2 góc trên và dưới bên phải.



Ở mảng Data Segment:

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	
0x10010000	0x00ffffff	0x00ffffff	0x00ffffff	0x00ffffff	0x00ffffff	0x00ffffff	
0x10010020	0x00ffffff	0x00ffffff	0x00ff0000	0x00ff0000	0x00ff0000	0x00ffffff	
0x10010040	0x00ffffff	0x00ffffff	0x00ff0000	0x00ffffff	0x00ffffff	0x00ff0000	
0x10010060	0x00ffffff	0x00ffffff	0x00ff0000	0x00ff0000	0x00ff0000	0x00ff0000	
0x10010080	0x00ffffff	0x00ffffff	0x00ff0000	0x00ffffff	0x00ffffff	0x00ff0000	
0x100100a0	0x00ffffff	0x00ffffff	0x00ff0000	0x00ffffff	0x00ffffff	0x00ff0000	
0x100100c0	0x00ffffff	0x00ffffff	0x00ff0000	0x00ff0000	0x00ff0000	0x00ffffff	

Ta thấy ở địa chỉ đầu tiên 0X10010000 bằng với giá trị thanh ghi \$k0 (MONITOR_SCREEN). Giá trị 0x00FFFFFF tương ứng với màu trắng. Giá trị 0x00FF0000 tương ứng với giá trị màu đỏ, có vị trí tương ứng với chữ B màu đỏ nền trắng. Mỗi *Value* cách nhau 4 bit tương ứng với mỗi pixel.