# Thực hành kiến trúc máy tính – tuần 10

Trần Khánh Quỳnh – 20225762

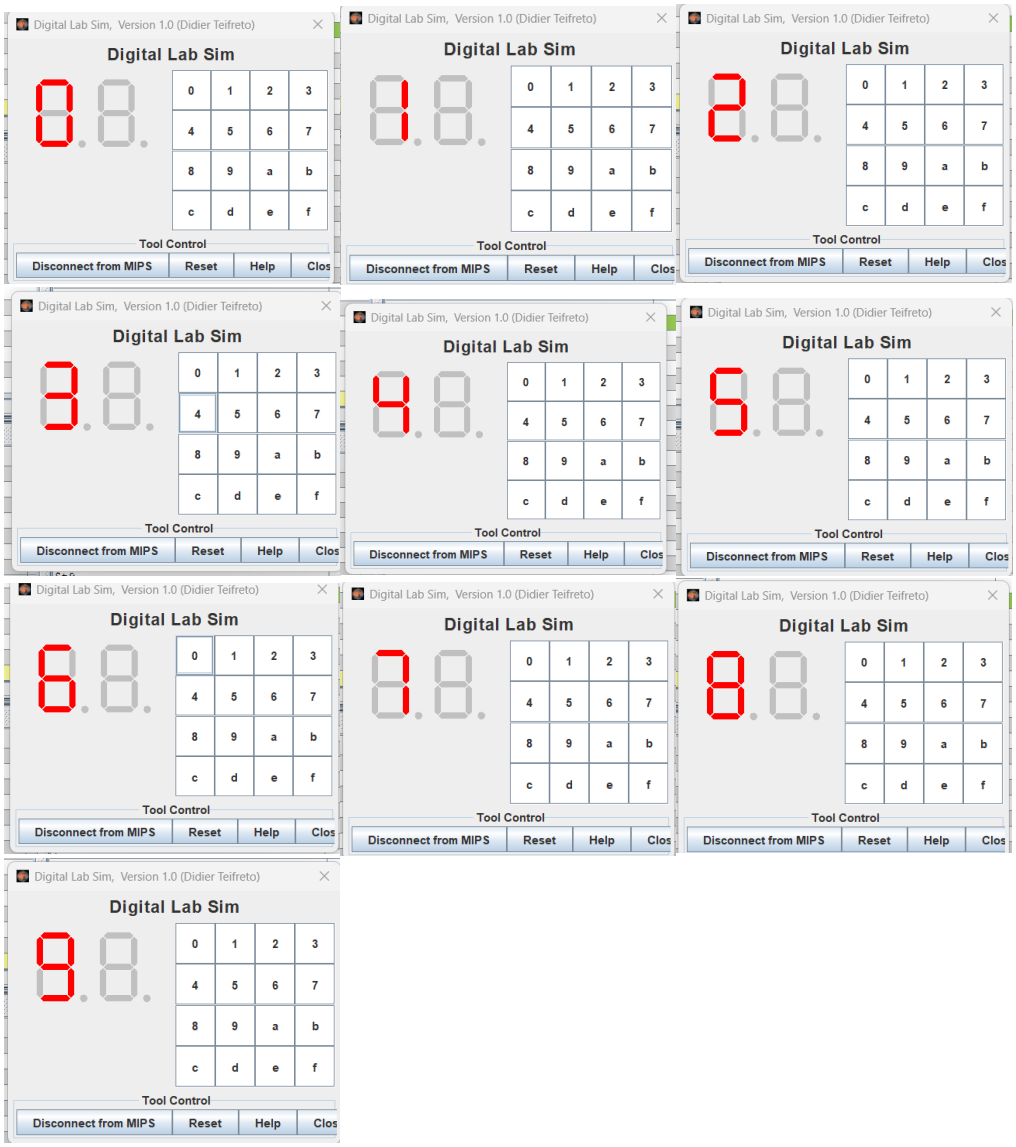
## 1. Assignment 1

-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


.data
A: .word 0x3F, 0x6,0x5B,0x4F, 0X66,0X6D, 0X7D, 0X7, 0X7F, 0X6F
.text
main:
      la $s1, A         #load the address of the array
      addi $t1, $0, 0   #i = 0
      addi $t2, $0, 10  #n = 10
      li $t3, 1         #i++
      li $t4, 1
      li $t5, -1
loop_0to9:
      lw $a0, ($s1)            #set value for segments
      jal SHOW_7SEG_LEFT       #show
      nop
      addi $s1, $s1, 4  #point to the next element
      add $t1, $t1, $t3
      beq $t1, $t2, change_direction
      j loop_0to9
loop_9to0:
   lw $a0, ($s1)
   jal SHOW_7SEG_LEFT
   nop
   addi $s1, $s1, -4
   add $t1, $t1, $t3
   beq $t1, $0, change_direction
```

```
    j loop_9to0
change_direction:
    sub $t3, $0, $t3    #change the direction i++ --> i-- --> i++ --> ...
    beq $t3, $t4, loop_0to9
    beq $t3, $t5, loop_9to0

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

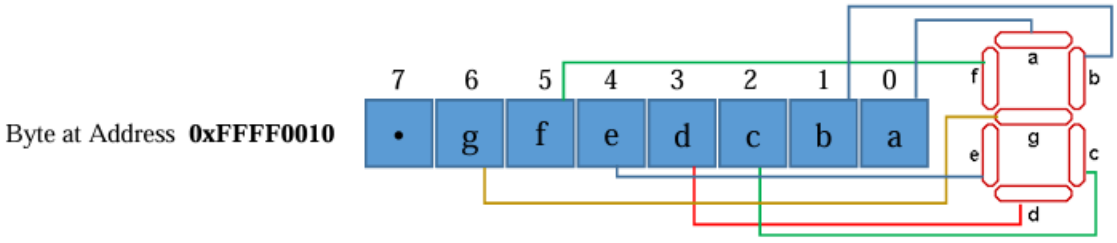
-Kết quả chạy:




LED đếm các số từ 0 đến 9, rồi sau đó lại từ 9 về 0, rồi tiếp tục lặp lại.

-Giải thích:

Thanh ghi $a0 dùng để lưu giá trị muốn hiển thị lên LED




Ví dụ, nếu muốn hiển thị số 0, dấu chấm và thanh g sẽ tắt nên giá trị của bit là 0, a,b,c,d,e,f hiển thị nên giá trị của bit là 1

→Mã nhị phân $a0 = 0x00111111 = 0x3F. Tương tự với các số còn lại

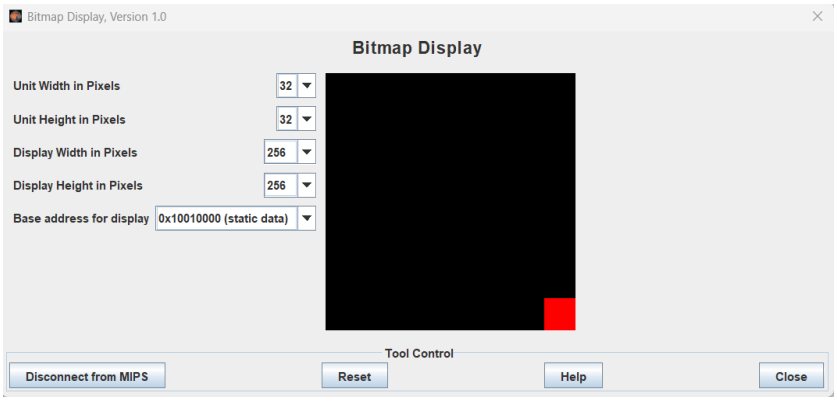
Thanh ghi $t1 là biến i để chạy vòng lặp, thanh ghi $t2 để lưu lại số phần tử trong mảng. Thanh ghi $s1 lưu địa chỉ của mảng A, mảng A chứa các giá trị từ 0 đến 9 để có thể hiển thị lên LED. Thanh ghi $t3 dùng để chuyển hướng đếm, nếu như đếm từ 0 đến 9 thì $t1 hay i sẽ cộng với 1, còn nếu như đếm từ 9 đến 0 thì $t1 hay i sẽ giảm đi 1.

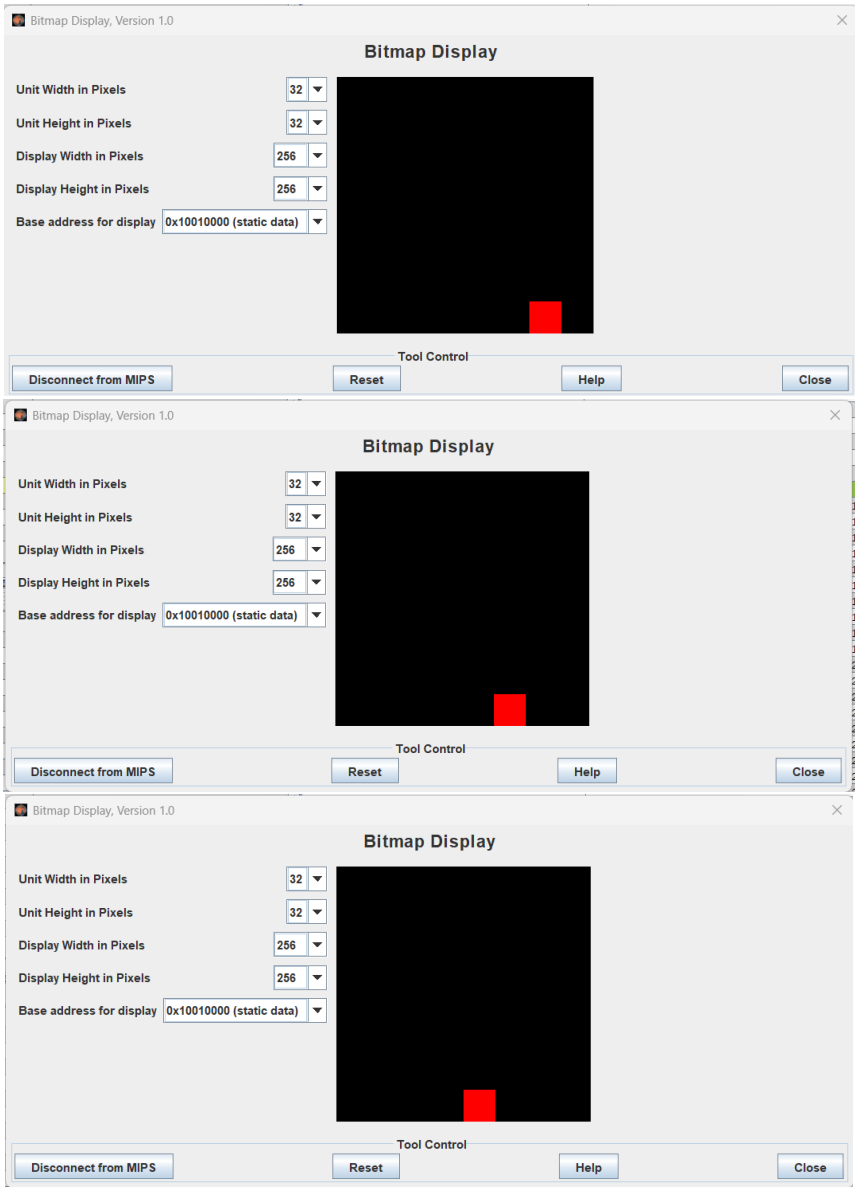
## 2. Assignment 2:

-Mã nguồn

```
.eqv MONITOR_SCREEN 0x10010000   #Dia chi bat dau cua bo nho man hinh
.eqv RED            0x00FF0000   #Cac gia tri mau thuong su dung
.eqv GREEN          0x0000FF00
.eqv BLUE           0x000000FF
.eqv WHITE          0x00FFFFFF
.eqv YELLOW         0x00FFFF00
.eqv BLACK        0x00000000
.text
       li $k0, MONITOR_SCREEN
       addi $k0, $k0, 256  #Dat dia chi bat dau la o cuoi cung goc phai hinh
loop:  li $t0, RED  #To mau do cho o vuong
       li $t1, BLACK #To de mau den len o mau do
       sw  $t0, ($k0)
       sw $t1, ($k0)
       addi $k0, $k0, -4   #Di chuyen
       bne $k0, $0, loop
       nop
exit:
```

-Kết quả chạy:






-Giải thích:

Ta hiển thị theo màn hình 256x256, mỗi pixel là 32x32 thì sẽ có 64 ô. Đặt vị trí đầu tiên của hình vuông ở ô cuối cùng bên phải, hàng cuối cùng → địa chỉ bắt đầu là địa chỉ monitor_screen + 256.

Để di chuyển, ta lần tô màu đỏ cho ô đó, sau đó tô đè màu đen lên rồi giảm địa chỉ lưu trong thanh ghi k0 sang ô cạnh đó bên trái, rồi tiếp tục vòng lặp thì ta có ô đỏ di chuyển dần từ phải sang trái