Thực hành Kiến trúc máy tính tuần 10

Họ tên: Đỗ Hoàng Minh Hiếu

MSSV: 20225837

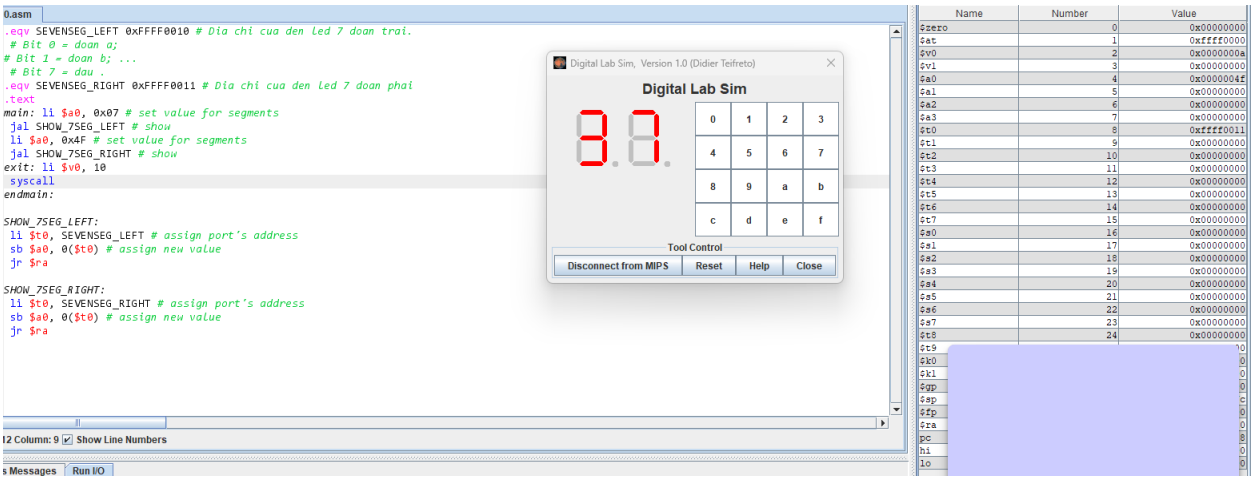
Bài 1:

Code:

```
.eqv SEVENSEG_LEFT 0xFFFF0010 # Dia chi cua den led 7 doan trai.
 # Bit 0 = doan a;
# Bit 1 = doan b; ...
 # Bit 7 = dau .
.eqv SEVENSEG_RIGHT 0xFFFF0011 # Dia chi cua den led 7 doan phai
.text
main: li $a0, 0x07 # set value for segments
 jal SHOW_7SEG_LEFT # show
 li $a0, 0x4F # set value for segments
 jal SHOW_7SEG_RIGHT # show
exit: li $v0, 10
 syscall
endmain:

SHOW_7SEG_LEFT:
 li $t0, SEVENSEG_LEFT # assign port's address
 sb $a0, 0($t0) # assign new value
 jr $ra

SHOW_7SEG_RIGHT:
 li $t0, SEVENSEG_RIGHT # assign port's address
 sb $a0, 0($t0) # assign new value
 jr $ra
```

Chạy thử:

Với MSSV 20225837 thì ta cần in ra số 37. Kết quả chạy thử:



=> Chương trình chạy đúng.

Bài 2:

Code:

```
.eqv SEVENSEG_LEFT 0xFFFF0011 # Dia chi cua den led 7 doan trai
.eqv SEVENSEG_RIGHT 0xFFFF0010 # Dia chi cua den led 7 doan phai
.data
	message: .asciiz "Nhap vao mot so nguyen: "
.text
main:
	li $v0, 4
	la $a0, message
	syscall # Display message

	li $v0, 5
	syscall #Read a number
	bltz $v0, main #Nhap lai neu $v0 < 0

	move $s0, $v0
	li $t2, 10
	div $s0, $t2
	mfhi $t1 # Chia lay so cuoi
	jal num_0
	nop
	jal SHOW_7SEG_RIGHT # show
	nop
```

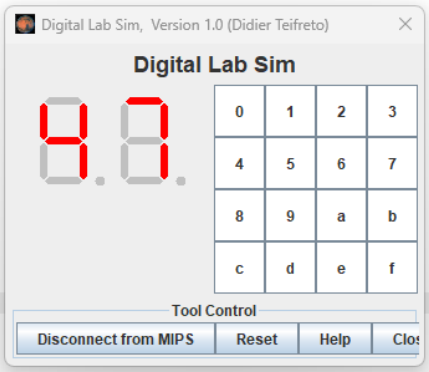
```
        mflo $t1
        div $t1, $t2
        mfhi $t1 #Lay so thu 2
        jal num_0
        nop
        jal SHOW_7SEG_LEFT # show
        nop
exit: li $v0, 10
        syscall
endmain:

num_0:
        bne $t1, $zero, num_1
        li $a0, 0x3F
        jr $ra
num_1:
        li $s1, 1
        bne $t1, $s1, num_2
        li $a0, 0x6
        jr $ra
num_2:
        li $s1, 2
        bne $t1, $s1, num_3
        li $a0, 0x5B
        jr $ra
num_3:
        li $s1, 3
        bne $t1, $s1, num_4
        li $a0, 0x4F
        jr $ra
num_4:
        li $s1, 4
        bne $t1, $s1, num_5
        li $a0, 0x66
        jr $ra
```

```
num_5:
        li $s1, 5
        bne $t1, $s1, num_6
        li $a0, 0x6D
        jr $ra
num_6:
        li $s1, 6
        bne $t1, $s1, num_7
        li $a0, 0x7D
        jr $ra
num_7:
        li $s1, 7
        bne $t1, $s1, num_8
        li $a0, 0x7
        jr $ra
num_8:
        li $s1, 8
        bne $t1, $s1, num_9
        li $a0, 0x7F
        jr $ra
num_9:
        li $s1, 9
        li $a0, 0x6F
        jr $ra
SHOW_7SEG_LEFT:
 li $t0, SEVENSEG_LEFT # assign port's address
 sb $a0, 0($t0) # assign new value
 jr $ra

SHOW_7SEG_RIGHT:
 li $t0, SEVENSEG_RIGHT # assign port's address
 sb $a0, 0($t0) # assign new value
 jr $ra
```

Chạy thử chương trình:

TH1: Nhập vào số 47

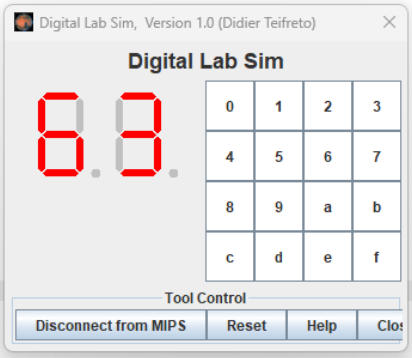
```
num_6:
    li $s1, 6
    bne $t1, $s1, num_7
    li $a0, 0x7D
    jr $ra
num_7:
    li $s1, 7
    bne $t1, $s1, num_8
    li $a0, 0x7
    jr $ra
num_8:
    li $s1, 8
    bne $t1, $s1, num_9
    li $a0, 0x7F
    jr $ra
num_9:
    li $s1, 9
    li $a0, 0x6F
    jr $ra
SHOW_7SEG_LEFT:
 li $t0, SEVENSEG_LEFT # assign port's address
 sb $a0, 0($t0) # assign new value
 jr $ra

SHOW_7SEG_RIGHT:
 li $t0, SEVENSEG_RIGHT # assign port's address
 sb $a0, 0($t0) # assign new value
 jr $ra
```




```
Line: 85 Column: 35  Show Line Numbers
Mars Messages  Run I/O
Nhap vao mot so nguyen: 47

-- program is finished running --
Clear
```

TH2: Nhập số 87658763

```
num_6:
    li $s1, 6
    bne $t1, $s1, num_7
    li $a0, 0x7D
    jr $ra
num_7:
    li $s1, 7
    bne $t1, $s1, num_8
    li $a0, 0x7
    jr $ra
num_8:
    li $s1, 8
    bne $t1, $s1, num_9
    li $a0, 0x7F
    jr $ra
num_9:
    li $s1, 9
    li $a0, 0x6F
    jr $ra
SHOW_7SEG_LEFT:
 li $t0, SEVENSEG_LEFT # assign port's address
 sb $a0, 0($t0) # assign new value
 jr $ra

SHOW_7SEG_RIGHT:
 li $t0, SEVENSEG_RIGHT # assign port's address
 sb $a0, 0($t0) # assign new value
 jr $ra
```

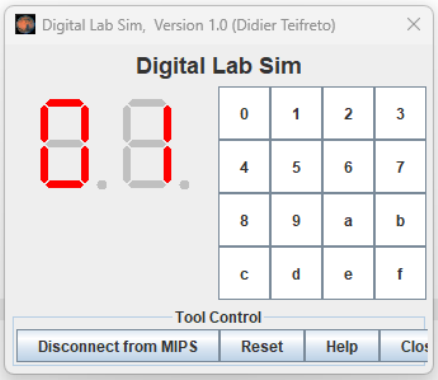



```
Line: 85 Column: 35  Show Line Numbers
Mars Messages  Run I/O
Nhap vao mot so nguyen: 87658763

-- program is finished running --
Clear
```

TH3: Nhập số 1

```
num_6:
    li $s1, 6
    bne $t1, $s1, num_7
    li $a0, 0x7D
    jr $ra
num_7:
    li $s1, 7
    bne $t1, $s1, num_8
    li $a0, 0x7
    jr $ra
num_8:
    li $s1, 8
    bne $t1, $s1, num_9
    li $a0, 0x7F
    jr $ra
num_9:
    li $s1, 9
    li $a0, 0x6F
    jr $ra
SHOW_7SEG_LEFT:
 li $t0, SEVENSEG_LEFT # assign port's address
 sb $a0, 0($t0) # assign new value
 jr $ra

SHOW_7SEG_RIGHT:
 li $t0, SEVENSEG_RIGHT # assign port's address
 sb $a0, 0($t0) # assign new value
 jr $ra
```

jr $t1 Jump register unconditionally : Jump to statement whose address is in $t1



Line: 85 Column: 35 Show Line Numbers

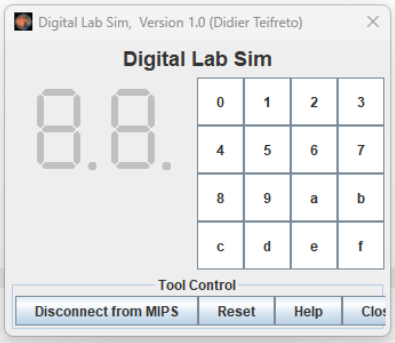
Mars Messages | Run I/O

```
Nhap vao mot so nguyen: 1

-- program is finished running --
```

Clear

TH4: Nhập vào số âm bất kỳ

```
num_6:
    li $s1, 6
    bne $t1, $s1, num_7
    li $a0, 0x7D
    jr $ra
num_7:
    li $s1, 7
    bne $t1, $s1, num_8
    li $a0, 0x7
    jr $ra
num_8:
    li $s1, 8
    bne $t1, $s1, num_9
    li $a0, 0x7F
    jr $ra
num_9:
    li $s1, 9
    li $a0, 0x6F
    jr $ra
SHOW_7SEG_LEFT:
 li $t0, SEVENSEG_LEFT # assign port's address
 sb $a0, 0($t0) # assign new value
 jr $ra

SHOW_7SEG_RIGHT:
 li $t0, SEVENSEG_RIGHT # assign port's address
 sb $a0, 0($t0) # assign new value
 jr $ra
```



Line: 85 Column: 35 Show Line Numbers

Mars Messages | Run I/O

```
Nhap vao mot so nguyen: -6458443
Nhap vao mot so nguyen:
```

Clear

(Nhập lại nếu input là một số âm)

Bài 3:

Code:

```
.eqv SEVENSEG_LEFT 0xFFFF0011 # Dia chi cua den led 7 doan trai
.eqv SEVENSEG_RIGHT 0xFFFF0010 # Dia chi cua den led 7 doan phai
.data
	message: .asciiz "Nhap vao mot ky tu: "
.text
main:
	li $v0, 4
	la $a0, message
	syscall # Display message

	li $v0, 12
	syscall #Read a number

For:
	addi $t1, $t1, 1
	beq $v0, $t1, EndFor
	j For
EndFor:

	li $t2, 10
	div $t1, $t2
	mfhi $t1 # Chia lay so cuoi
	jal num_0
	nop
	jal SHOW_7SEG_RIGHT # show
	nop

	mflo $t1
	div $t1, $t2
	mfhi $t1 #Lay so thu 2
	jal num_0
	nop
	jal SHOW_7SEG_LEFT # show
	nop
```

```
exit: li $v0, 10
      syscall
endmain:

num_0:
      bne $t1, $zero, num_1
      li $a0, 0x3F
      jr $ra
num_1:
      li $s1, 1
      bne $t1, $s1, num_2
      li $a0, 0x6
      jr $ra
num_2:
      li $s1, 2
      bne $t1, $s1, num_3
      li $a0, 0x5B
      jr $ra
num_3:
      li $s1, 3
      bne $t1, $s1, num_4
      li $a0, 0x4F
      jr $ra
num_4:
      li $s1, 4
      bne $t1, $s1, num_5
      li $a0, 0x66
      jr $ra
num_5:
      li $s1, 5
      bne $t1, $s1, num_6
      li $a0, 0x6D
      jr $ra
num_6:
      li $s1, 6
      bne $t1, $s1, num_7
```

```
        li $a0, 0x7D
        jr $ra
num_7:
        li $s1, 7
        bne $t1, $s1, num_8
        li $a0, 0x7
        jr $ra
num_8:
        li $s1, 8
        bne $t1, $s1, num_9
        li $a0, 0x7F
        jr $ra
num_9:
        li $s1, 9
        li $a0, 0x6F
        jr $ra
SHOW_7SEG_LEFT:
 li $t0, SEVENSEG_LEFT # assign port's address
 sb $a0, 0($t0) # assign new value
 jr $ra

SHOW_7SEG_RIGHT:
 li $t0, SEVENSEG_RIGHT # assign port's address
 sb $a0, 0($t0) # assign new value
 jr $ra
```

Chạy thử chương trình:

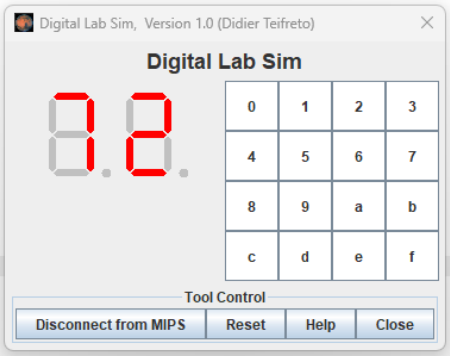
TH1: Nhập vào ký tự ‘H’

```
num_6:
    li $s1, 6
    bne $t1, $s1, num_7
    li $a0, 0x7D
    jr $ra
num_7:
    li $s1, 7
    bne $t1, $s1, num_8
    li $a0, 0x7
    jr $ra
num_8:
    li $s1, 8
    bne $t1, $s1, num_9
    li $a0, 0x7F
    jr $ra
num_9:
    li $s1, 9
    li $a0, 0x6F
    jr $ra
SHOW_7SEG_LEFT:
 li $t0, SEVENSEG_LEFT # assign port's address
 sb $a0, 0($t0) # assign new value
 jr $ra

SHOW_7SEG_RIGHT:
 li $t0, SEVENSEG_RIGHT # assign port's address
 sb $a0, 0($t0) # assign new value
 jr $ra
```

Line: 86 Column: 8 Show Line Numbers

Mars Messages | Run I/O

```
Nhap vao mot ky tu: H
-- program is finished running --
```



TH2: Nhập ký tự '/'

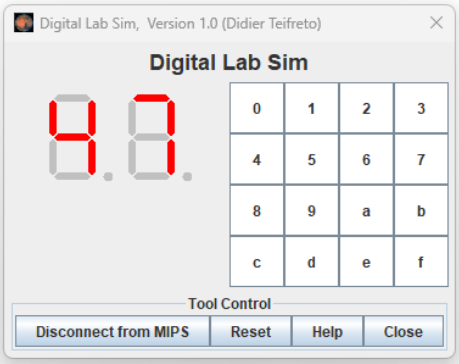
lab10.asm

```
    li $t2, 10
    div $t1, $t2
    mfhi $t1 # Chia lay so cuoi
    jal num_0
    nop
    jal SHOW_7SEG_RIGHT # show
    nop

    mflo $t1
    div $t1, $t2
    mfhi $t1 #Lay so thu 2
    jal num_0
    nop
    jal SHOW_7SEG_LEFT # show
    nop
exit: li $v0, 10
    syscall
endmain:

num_0:
    bne $t1, $zero, num_1
    li $a0, 0x3F
    jr $ra
num_1:
    li $s1, 1
    bne $t1, $s1, num_2
    li $a0, 0x6
    jr $ra
```

Line: 86 Column: 8 Show Line Numbers

Mars Messages | Run I/O

```
Nhap vao mot ky tu: /
-- program is finished running --
```

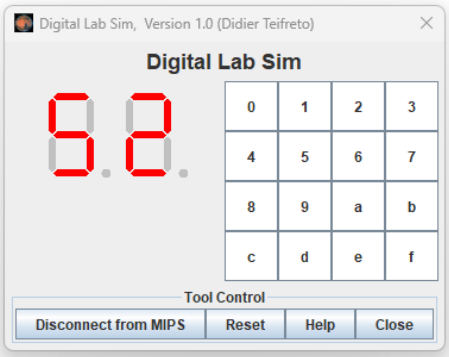


TH3: Nhập ký tự '4'

lab10.asm

```
19
20      li $t2, 10
21      div $t1, $t2
22      mfhi $t1 # Chia lay so cuoi
23      jal num_0
24      nop
25      jal SHOW_7SEG_RIGHT # show
26      nop
27
28      mflo $t1
29      div $t1, $t2
30      mfhi $t1 #Lay so thu 2
31      jal num_0
32      nop
33      jal SHOW_7SEG_LEFT # show
34      nop
35  exit: li $v0, 10
36      syscall
37  endmain:
38
39  num_0:
40      bne $t1, $zero, num_1
41      li $a0, 0x3F
42      jr $ra
43  num_1:
44      li $s1, 1
45      bne $t1, $s1, num_2
46      li $a0, 0x6
47      jr $ra
```

Line: 86 Column: 8 Show Line Numbers



Mars Messages | Run I/O

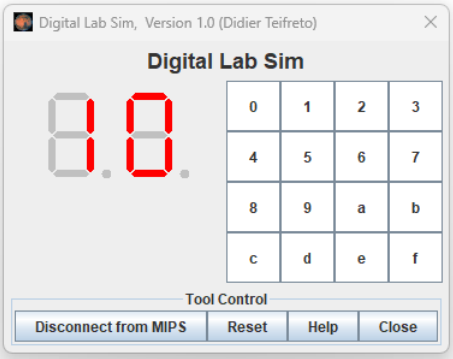
```
Nhap vao mot ky tu: 4
-- program is finished running --
```

TH4: Không nhập gì

lab10.asm

```
19
20      li $t2, 10
21      div $t1, $t2
22      mfhi $t1 # Chia lay so cuoi
23      jal num_0
24      nop
25      jal SHOW_7SEG_RIGHT # show
26      nop
27
28      mflo $t1
29      div $t1, $t2
30      mfhi $t1 #Lay so thu 2
31      jal num_0
32      nop
33      jal SHOW_7SEG_LEFT # show
34      nop
35  exit: li $v0, 10
36      syscall
37  endmain:
38
39  num_0:
40      bne $t1, $zero, num_1
41      li $a0, 0x3F
42      jr $ra
43  num_1:
44      li $s1, 1
45      bne $t1, $s1, num_2
46      li $a0, 0x6
47      jr $ra
```

Line: 86 Column: 8 Show Line Numbers



Mars Messages | Run I/O

Clear

```
Reset: reset completed.

Nhap vao mot ky tu:

-- program is finished running --
```

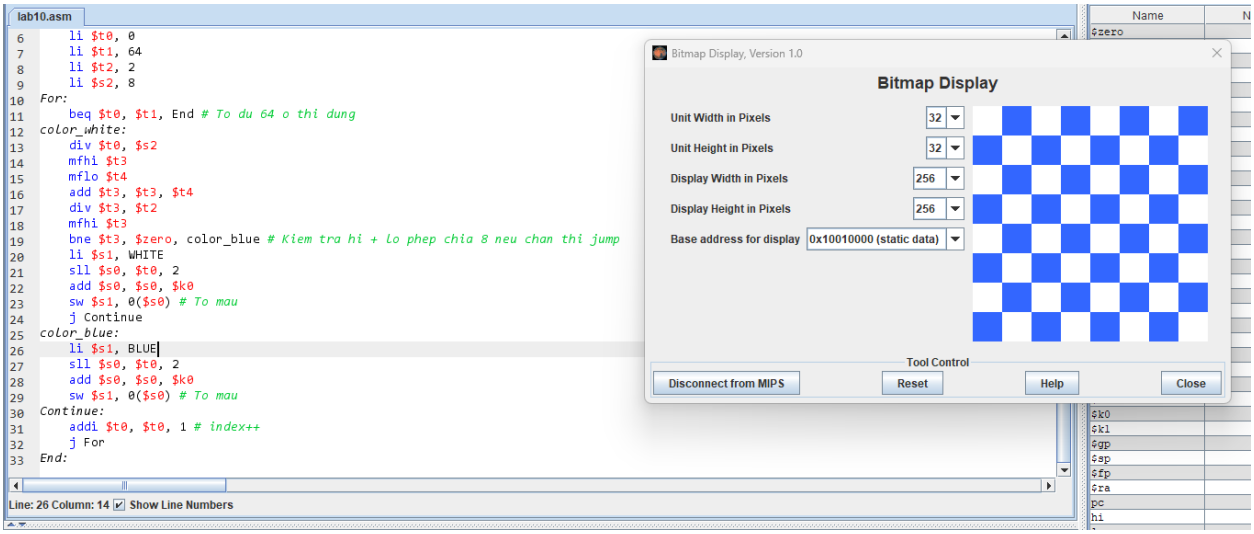
Bài 4:

Code:

```
.eqv MONITOR_SCREEN 0x10010000
.eqv BLUE 0x003366FF
.eqv WHITE 0x00FFFFFF
.text
        li $k0, MONITOR_SCREEN
        li $t0, 0
        li $t1, 64
        li $t2, 2
        li $s2, 8
For:
        beq $t0, $t1, End # To du 64 o thi dung
color_white:
        div $t0, $s2
        mfhi $t3
        mflo $t4
        add $t3, $t3, $t4
        div $t3, $t2
        mfhi $t3
        bne $t3, $zero, color_blue # Kiem tra hi + lo phep chia 8 neu chan thi jump
        li $s1, WHITE
        sll $s0, $t0, 2
        add $s0, $s0, $k0
        sw $s1, 0($s0) # To mau
        j Continue
color_blue:
        li $s1, BLUE
        sll $s0, $t0, 2
        add $s0, $s0, $k0
        sw $s1, 0($s0) # To mau
Continue:
        addi $t0, $t0, 1 # index++
        j For
End:
```

Chạy thử chương trình:

Cài Display Width và Height là 256, Unit là 32 để chia thành 64 ((256/32)^2) điểm ảnh.



Bài 5:

Code:

```
.eqv MONITOR_SCREEN 0x10010000
.eqv RED 0x00FF0000
.eqv GREEN 0x0000FF00
.data
        x1: .asciiz "Nhap x1: "
        y1: .asciiz "Nhap y1: "
        x2: .asciiz "Nhap x2: "
        y2: .asciiz "Nhap y2: "
        error1: .asciiz "Error: x2 phai khac x1. Moi nhap lai!\n"
        error2: .asciiz "Error: y2 phai khac y1. Moi nhap lai!\n"
.text
        li $k0, MONITOR_SCREEN
        li $v0, 4
        la $a0, x1
        syscall
        li $v0, 5
        syscall
        move $s0, $v0 # $s0 = x1
```

```
        li $v0, 4
        la $a0, y1
        syscall
        li $v0, 5
        syscall
        move $s1, $v0  # $s1 = y1

NhapX2:
        li $v0, 4
        la $a0, x2
        syscall
        li $v0, 5
        syscall
        move $s2, $v0  # $s2 = x2
        beq $s2, $s0, Error1

NhapY2:
        li $v0, 4
        la $a0, y2
        syscall
        li $v0, 5
        syscall
        move $s3, $v0  # $s3 = y2
        beq $s3, $s1, Error2
        j Continue

Error1: li $v0, 4
        la $a0, error1
        syscall
        j NhapX2
Error2: li $v0, 4
        la $a0, error2
        syscall
        j NhapY2
# Doi 2 diem ve trai tren cung va phai duoi cung
Continue:
```

```
        bge $s3, $s1, Calc_X
        add $s4, $s3, $zero
        add $s3, $s1, $zero
        add $s1, $s4, $zero
Calc_X:
        bge $s2, $s0, Tsugi
        add $s4, $s2, $zero
        add $s2, $s0, $zero
        add $s0, $s4, $zero
# In vien ngang va doc va in mien
Tsugi:
        sll $t0, $s0, 6
        add $t0, $t0, $s1
        sll $t1, $s0, 6
        add $t1, $t1, $s3
        jal InVienNgang
        nop

        sll $t0, $s2, 6
        add $t0, $t0, $s1
        sll $t1, $s2, 6
        add $t1, $t1, $s3
        jal InVienNgang
        nop

        sll $t0, $s0, 6
        add $t0, $t0, $s1
        sll $t1, $s2, 6
        add $t1, $t1, $s1
        jal InVienDoc
        nop

        sll $t0, $s0, 6
        add $t0, $t0, $s3
        sll $t1, $s2, 6
        add $t1, $t1, $s3
```

```
        jal InVienDoc
        nop

        addi $s0, $s0, 1
        addi $s1, $s1, 1
        addi $s2, $s2, -1
        addi $s3, $s3, -1

        sll $t0, $s0, 6
        add $t0, $t0, $s1
        sll $t1, $s0, 6
        add $t1, $t1, $s3
        sll $t2, $s2, 6
        add $t2, $t2, $s3
For:
        jal InMien1
        nop
        beq $t1, $t2, End_for
        addi $t0, $t0, 64
        addi $t1, $t1, 64
        j For
End_for:
        li $v0, 10
        syscall
InVienNgang:
        sll $t2, $t0, 2
        li $a1, RED
        add $a2, $k0, $t2
        sw $a1, 0($a2)
        beq $t0, $t1, End_InVienNgang
        add $t0, $t0, 1
        j InVienNgang
End_InVienNgang:
        jr $ra

InVienDoc:
```

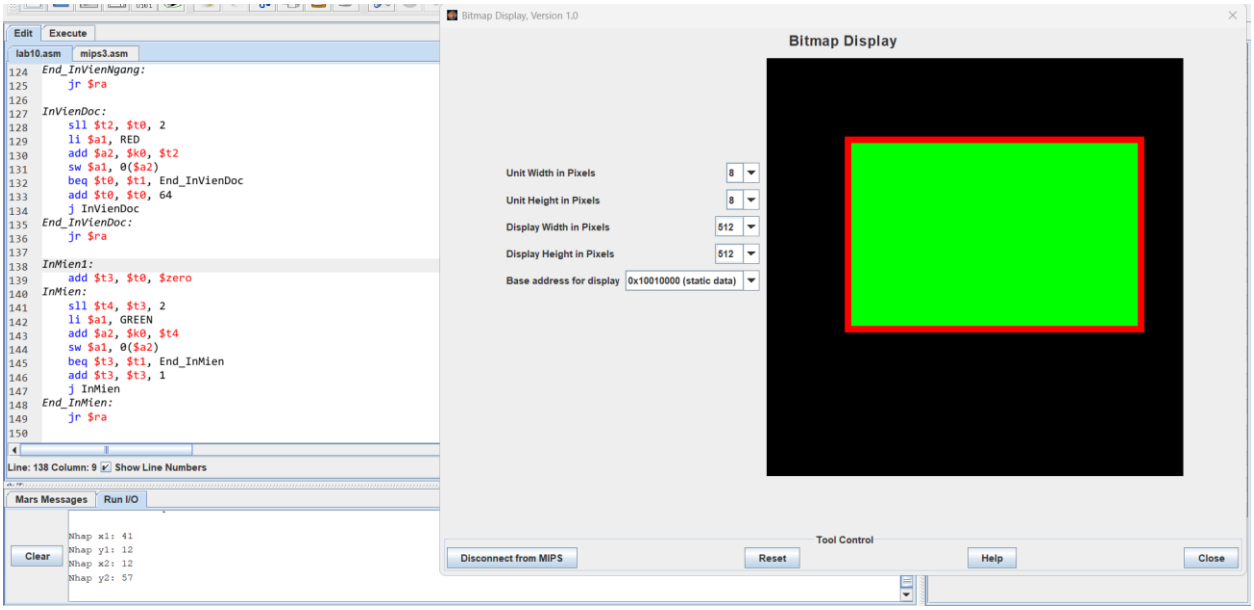
```
        sll $t2, $t0, 2
        li $a1, RED
        add $a2, $k0, $t2
        sw $a1, 0($a2)
        beq $t0, $t1, End_InVienDoc
        add $t0, $t0, 64
        j InVienDoc
End_InVienDoc:
        jr $ra

InMien1:
        add $t3, $t0, $zero
InMien:
        sll $t4, $t3, 2
        li $a1, GREEN
        add $a2, $k0, $t4
        sw $a1, 0($a2)
        beq $t3, $t1, End_InMien
        add $t3, $t3, 1
        j InMien
End_InMien:
        jr $ra
```

Chạy thử chương trình:

Cài Display Width và Height là 512, Unit là 8 để chia thành 4096 ((512/8)^2) điểm ảnh.

Ta mặc định x là số hàng, y là số cột

Với $(x_1, y_1) = (41, 12)$ và $(x_2, y_2) = (12, 57)$


Edit
Execute
lab10.asm
mips3.asm
End_InVienNgang:
jr $ra
InVienDoc:
sll $t2, $t0, 2
li $a1, RED
add $a2, $k0, $t2
sw $a1, 0($a2)
beq $t0, $t1, End_InVienDoc
add $t0, $t0, 64
j InVienDoc
End_InVienDoc:
jr $ra
InMien1:
add $t3, $t0, $zero
InMien:
sll $t4, $t3, 2
li $a1, GREEN
add $a2, $k0, $t4
sw $a1, 0($a2)
beq $t3, $t1, End_InMien
add $t3, $t3, 1
j InMien
End_InMien:
jr $ra
Line: 138 Column: 9
Show Line Numbers
Mars Messages
Run I/O
Clear
Nhap x1: 41
Nhap y1: 12
Nhap x2: 12
Nhap y2: 57
Bitmap Display, Version 1.0
Bitmap Display
Unit Width in Pixels
8
Unit Height in Pixels
8
Display Width in Pixels
512
Display Height in Pixels
512
Base address for display
0x10010000 (static data)
Tool Control
Disconnect from MIPS
Reset
Help
Close