**­­­­Computer Architecture Lab Report Week 10**

**Full name: Nguyễn Hồng Phúc**

**Student ID: 20225659**

Assignment 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, 0x6F # set value for segments

jal SHOW\_7SEG\_LEFT # show

li $a0, 0x6D # set value for segments

jal SHOW\_7SEG\_RIGHT # show

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

jr $ra

#---------------------------------------------------------------

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

# param[in] $a0 value to shown

# remark $t0 changed

#---------------------------------------------------------------

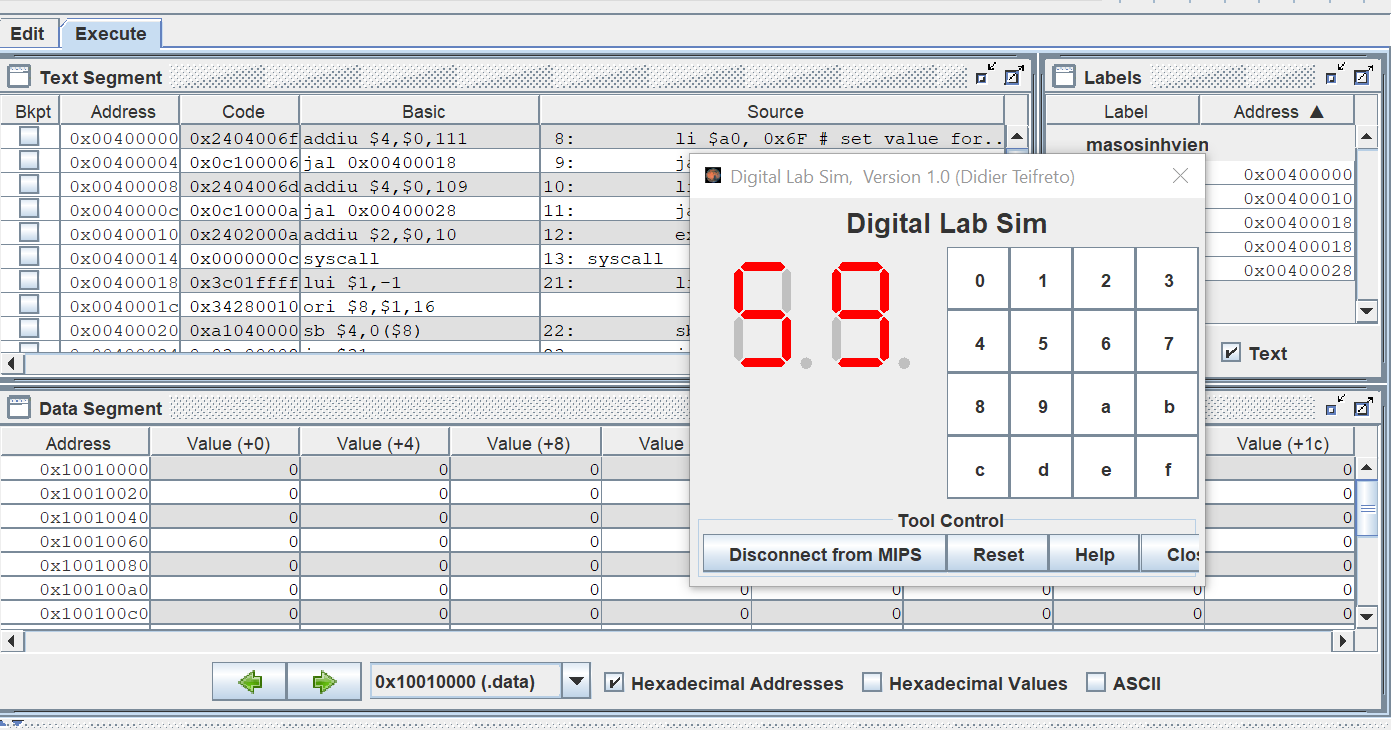
SHOW\_7SEG\_RIGHT: li $t0, SEVENSEG\_RIGHT # assign port's address

sb $a0, 0($t0) # assign new value

jr $ra

**Kết quả:**

Mở Digital Lab Sim trong Tool, kết nối nó với MIPS, chạy và hiển thị kết quả mssv là 20225659, số 59 là số cần hiện ra



Kết quả đúng với yêu cầu bài.

Assignment 2

**Code:**

.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

message: .asciiz "Nhap vao mot so nguyen: "

.text

main:

li $v0, 4

la $a0, message

syscall

li $v0, 5

syscall

move $s0, $v0

li $t2, 10

div $s0, $t2

mfhi $t1

case0r:

bne $t1, 0, case1r

li $a0, 0x3F

jal SHOW\_7SEG\_RIGHT

j defaultr

case1r:

bne $t1, 1, case2r

li $a0, 0x6

jal SHOW\_7SEG\_RIGHT

j defaultr

case2r:

bne $t1, 2, case3r

li $a0, 0x5B

jal SHOW\_7SEG\_RIGHT

j defaultr

case3r:

bne $t1, 3, case4r

li $a0, 0x4F

jal SHOW\_7SEG\_RIGHT

j defaultr

case4r:

bne $t1, 4, case5r

li $a0, 0x66

jal SHOW\_7SEG\_RIGHT

j defaultr

case5r:

bne $t1, 5, case6r

li $a0, 0x6D

jal SHOW\_7SEG\_RIGHT

j defaultr

case6r:

bne $t1, 6, case7r

li $a0, 0x7D

jal SHOW\_7SEG\_RIGHT

j defaultr

case7r:

bne $t1, 7, case8r

li $a0, 0x7

jal SHOW\_7SEG\_RIGHT

j defaultr

case8r:

bne $t1, 8, case9r

li $a0, 0x7F

jal SHOW\_7SEG\_RIGHT

j defaultr

case9r:

bne $t1, 9, defaultr

li $a0, 0x6F

jal SHOW\_7SEG\_RIGHT

j defaultr

defaultr:

sub $s0, $s0, $t1

div $s0, $t2

mflo $t3

div $t3, $t2

mfhi $t1

case0l:

bne $t1, 0, case1l

li $a0, 0x3F

jal SHOW\_7SEG\_LEFT

j defaultl

case1l:

bne $t1, 1, case2l

li $a0, 0x6

jal SHOW\_7SEG\_LEFT

j defaultl

case2l:

bne $t1, 2, case3l

li $a0, 0x5B

jal SHOW\_7SEG\_LEFT

j defaultl

case3l:

bne $t1, 3, case4l

li $a0, 0x4F

jal SHOW\_7SEG\_LEFT

j defaultl

case4l:

bne $t1, 4, case5l

li $a0, 0x66

jal SHOW\_7SEG\_LEFT

j defaultl

case5l:

bne $t1, 5, case6l

li $a0, 0x6D

jal SHOW\_7SEG\_LEFT

j defaultl

case6l:

bne $t1, 6, case7l

li $a0, 0x7D

jal SHOW\_7SEG\_LEFT

j defaultl

case7l:

bne $t1, 7, case8l

li $a0, 0x7

jal SHOW\_7SEG\_LEFT

j defaultl

case8l:

bne $t1, 8, case9l

li $a0, 0x7F

jal SHOW\_7SEG\_LEFT

j defaultl

case9l:

bne $t1, 9, defaultl

li $a0, 0x6F

jal SHOW\_7SEG\_LEFT

j defaultl

defaultl:

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

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:**

Nhập số 58:

A screenshot of a computer

Description automatically generated

Nhập số 100:

A screenshot of a computer

Description automatically generated

Nhập số 9:

A screenshot of a computer

Description automatically generated

Assignment 3

**Code:**

.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

message: .asciiz "Nhap vao mot ky tu: "

.text

main:

li $v0, 4

la $a0, message

syscall

li $v0, 12

syscall

move $s0, $v0

li $t1, -1

For:

addi $t1, $t1, 1

beq $s0, $t1, EndFor

j For

EndFor:

#Vong For dung de lay gia tri Ascii roi luu vao thanh $t1

move $s0, $t1

li $t2, 10

div $s0, $t2

mfhi $t1

case0r:

bne $t1, 0, case1r

li $a0, 0x3F

jal SHOW\_7SEG\_RIGHT

j defaultr

case1r:

bne $t1, 1, case2r

li $a0, 0x6

jal SHOW\_7SEG\_RIGHT

j defaultr

case2r:

bne $t1, 2, case3r

li $a0, 0x5B

jal SHOW\_7SEG\_RIGHT

j defaultr

case3r: bne $t1, 3, case4r

li $a0, 0x4F

jal SHOW\_7SEG\_RIGHT

j defaultr

case4r: bne $t1, 4, case5r

li $a0, 0x66

jal SHOW\_7SEG\_RIGHT

j defaultr

case5r: bne $t1, 5, case6r

li $a0, 0x6D

jal SHOW\_7SEG\_RIGHT

j defaultr

case6r: bne $t1, 6, case7r

li $a0, 0x7D

jal SHOW\_7SEG\_RIGHT

j defaultr

case7r: bne $t1, 7, case8r

li $a0, 0x7

jal SHOW\_7SEG\_RIGHT

j defaultr

case8r: bne $t1, 8, case9r

li $a0, 0x7F

jal SHOW\_7SEG\_RIGHT

j defaultr

case9r: bne $t1, 9, defaultr

li $a0, 0x6F

jal SHOW\_7SEG\_RIGHT

j defaultr

defaultr:

sub $s0, $s0, $t1

div $s0, $t2

mflo $t3

div $t3, $t2

mfhi $t1

case0l:

bne $t1, 0, case1l

li $a0, 0x3F

jal SHOW\_7SEG\_LEFT

j defaultl

case1l:

bne $t1, 1, case2l

li $a0, 0x6

jal SHOW\_7SEG\_LEFT

j defaultl

case2l:

bne $t1, 2, case3l

li $a0, 0x5B

jal SHOW\_7SEG\_LEFT

j defaultl

case3l:

bne $t1, 3, case4l

li $a0, 0x4F

jal SHOW\_7SEG\_LEFT

j defaultl

case4l:

bne $t1, 4, case5l

li $a0, 0x66

jal SHOW\_7SEG\_LEFT

j defaultl

case5l:

bne $t1, 5, case6l

li $a0, 0x6D

jal SHOW\_7SEG\_LEFT

j defaultl

case6l:

bne $t1, 6, case7l

li $a0, 0x7D

jal SHOW\_7SEG\_LEFT

j defaultl

case7l:

bne $t1, 7, case8l

li $a0, 0x7

jal SHOW\_7SEG\_LEFT

j defaultl

case8l:

bne $t1, 8, case9l

li $a0, 0x7F

jal SHOW\_7SEG\_LEFT

j defaultl

case9l:

bne $t1, 9, defaultl

li $a0, 0x6F

jal SHOW\_7SEG\_LEFT

j defaultl

defaultl:

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

jr $ra

#---------------------------------------------------------------

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

# param[in] $a0 value to shown

# remark $t0 changed

#---------------------------------------------------------------

SHOW\_7SEG\_RIGHT:

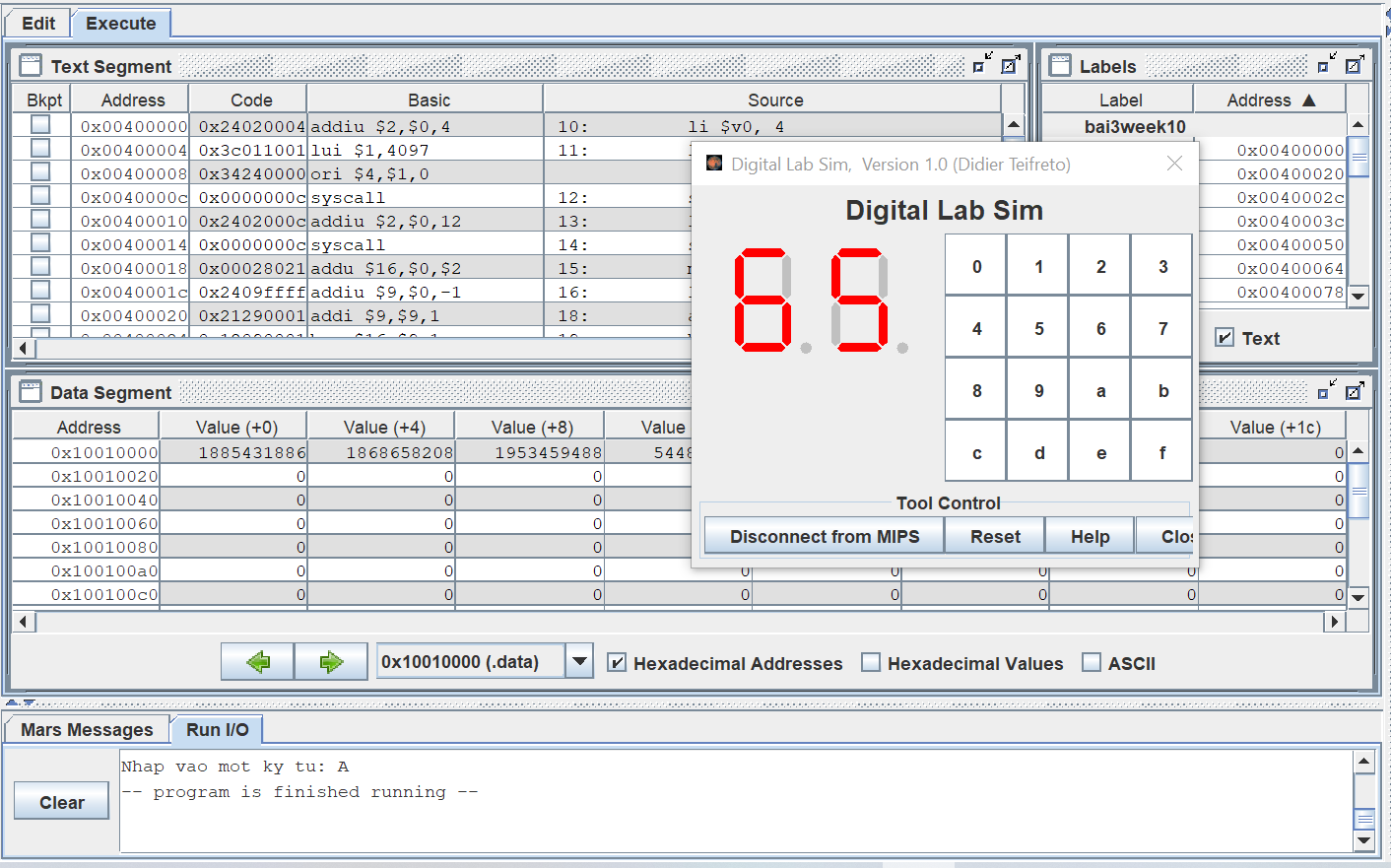
li $t0, SEVENSEG\_RIGHT # assign port's address

sb $a0, 0($t0) # assign new value

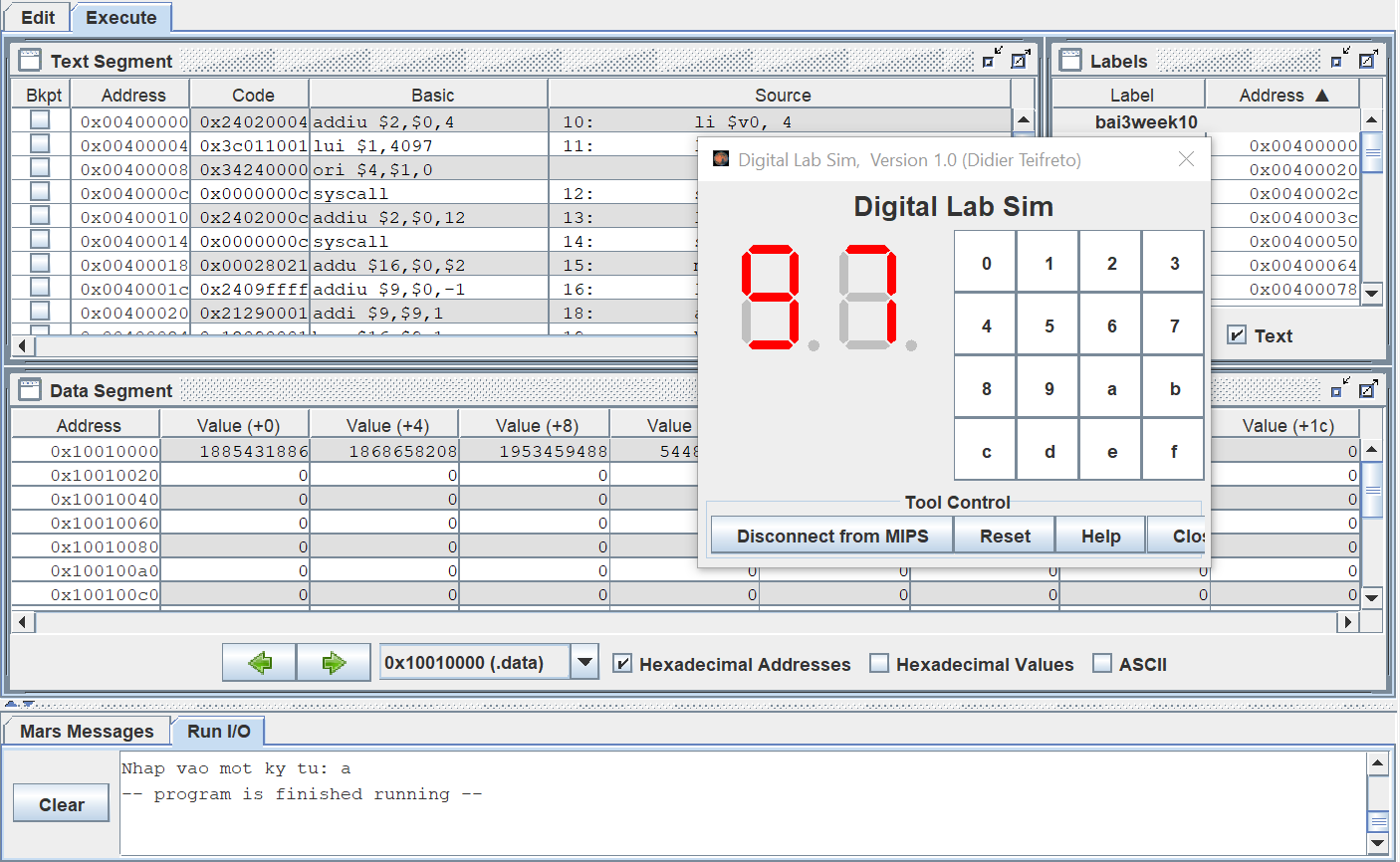
jr $ra

**Kết quả:**

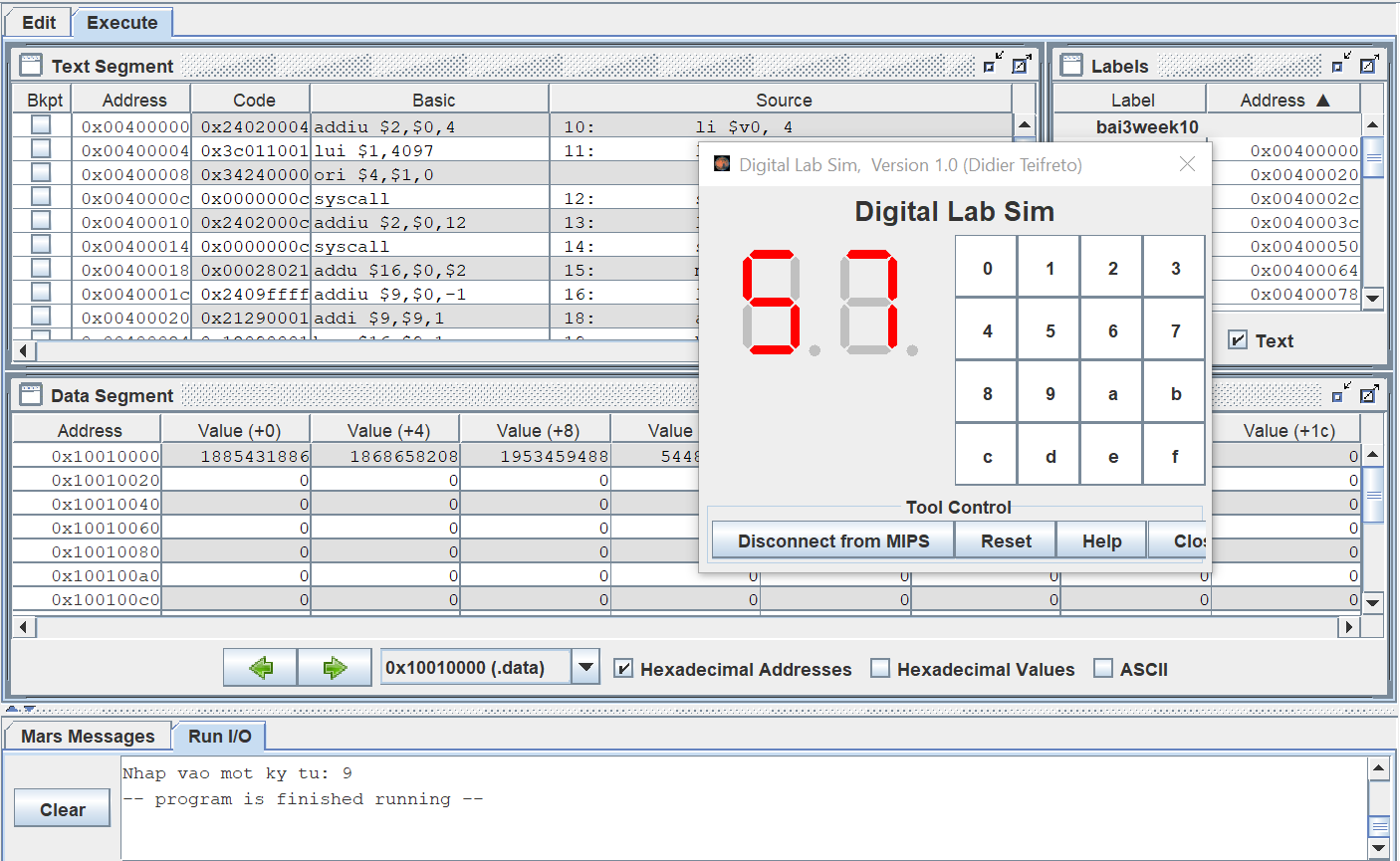
Nhập chữ A:



Nhập chữ a:



Nhập số 9:



Assignment 4

**Code:**

.eqv MONITOR\_SCREEN 0x10010000

.eqv YELLOW 0x00FFFF00

.eqv WHITE 0x00FFFFFF

.text

li $k0, MONITOR\_SCREEN

li $s0, 2

li $t0, -1 # Khoi tao j

For1: addi $t0, $t0, 1

beq $t0, 8, Exit

li $t1, -1 # Khoi tao i

For2: addi $t1, $t1, 1

beq $t1, 8, EndFor2

div $t0, $s0

mfhi $t2

div $t1, $s0

mfhi $t3

bne $t2, 0, Next

bne $t3, 0, Paint2

j Paint1

Next:

beq $t3, 0, Paint2

Paint1:

sll $s1, $t0, 3

add $s1, $s1, $t1

sll $s1, $s1, 2

add $s2, $s1, $k0

li $t4, YELLOW

sw $t4, 0($s2)

j For2

Paint2:

sll $s1, $t0, 3

add $s1, $s1, $t1

sll $s1, $s1, 2

add $s2, $s1, $k0

li $t4, WHITE

sw $t4, 0($s2)

j For2

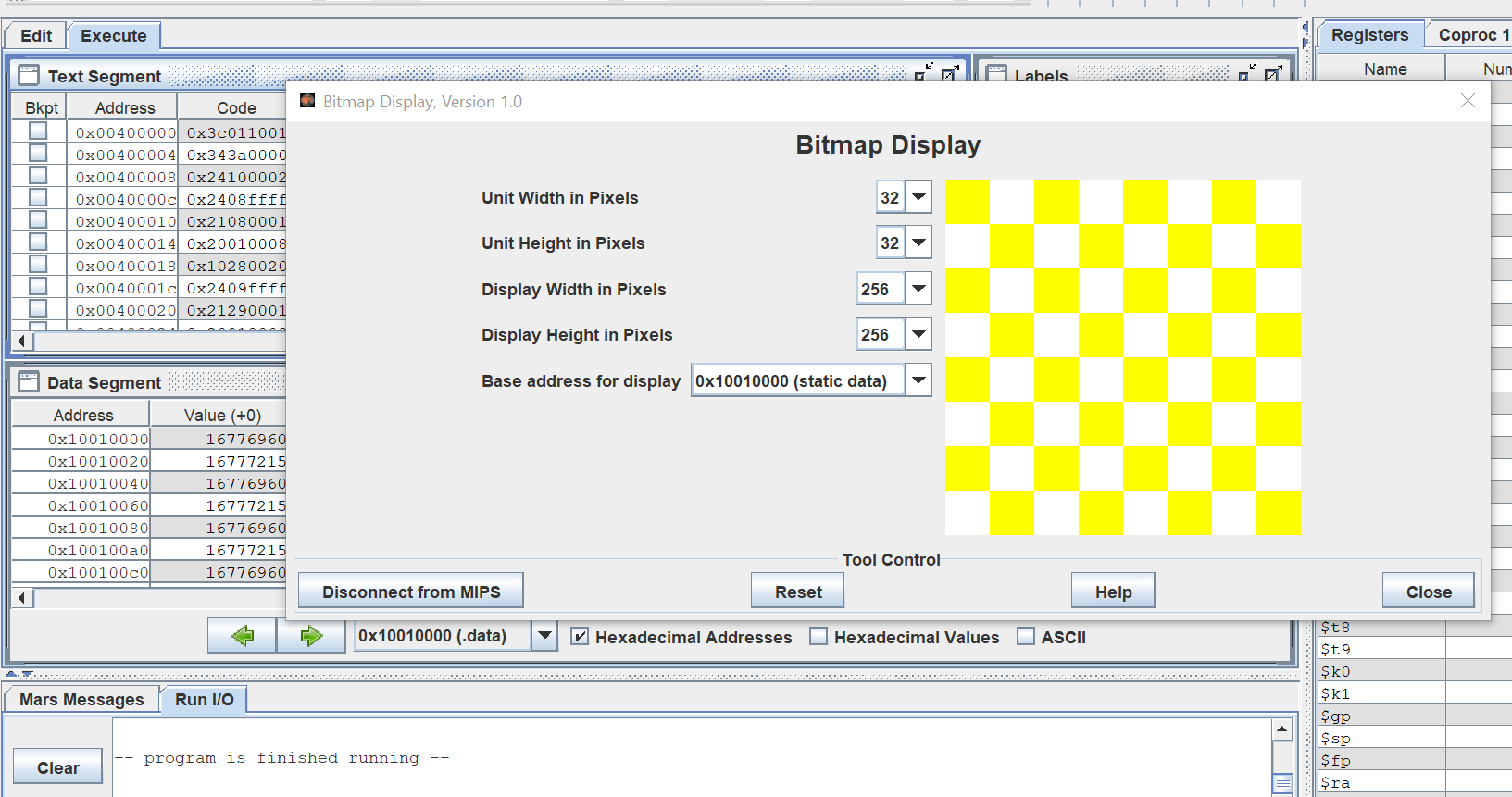
EndFor2:

j For1

Exit: li $v0, 10

Syscall

**Kết quả:**



Assignment 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

li $v0, 4

la $a0, y1

syscall

li $v0, 5

syscall

move $s1, $v0

NhapX2: li $v0, 4

la $a0, x2

syscall

li $v0, 5

syscall

move $s2, $v0

beq $s2, $s0, Error1

NhapY2: li $v0, 4

la $a0, y2

syscall

li $v0, 5

syscall

move $s3, $v0

beq $s3, $s1, Error2

j next

Error1: li $v0, 4

la $a0, error1

syscall

j NhapX2

Error2: li $v0, 4

la $a0, error2

syscall

j NhapY2

next:

slt $t0, $s0, $s2

slt $t1, $s1, $s3

beq $t0, 0, Case3

beq $t1, 0, Case2

Case1:

add $v0, $s1, $zero

For1:

bgt $v0, $s3, Exit

add $v1, $s0, $zero

For2:

bgt $v1, $s2, EndFor2

beq $v0, $s1, InVien1

beq $v0, $s3, InVien1

beq $v1, $s0, InVien1

beq $v1, $s2, InVien1

sll $t8, $v0, 6

add $t8, $t8, $v1

sll $t8, $t8, 2

li $a1, GREEN

add $a2, $k0, $t8

sw $a1, 0($a2)

add $v1, $v1, 1

j For2

InVien1:

sll $t8, $v0, 6

add $t8, $t8, $v1

sll $t8, $t8, 2

li $a1, RED

add $a2, $k0, $t8

sw $a1, 0($a2)

add $v1, $v1, 1

j For2

EndFor2:

add $v0, $v0, 1

j For1

Case2:

add $v0, $s3, $zero

For3:

bgt $v0, $s1, Exit

add $v1, $s0, $zero

For4:

bgt $v1, $s2, EndFor4

beq $v0, $s1, InVien2

beq $v0, $s3, InVien2

beq $v1, $s0, InVien2

beq $v1, $s2, InVien2

sll $t8, $v0, 6

add $t8, $t8, $v1

sll $t8, $t8, 2

li $a1, GREEN

add $a2, $k0, $t8

sw $a1, 0($a2)

add $v1, $v1, 1

j For4

InVien2:

sll $t8, $v0, 6

add $t8, $t8, $v1

sll $t8, $t8, 2

li $a1, RED

add $a2, $k0, $t8

sw $a1, 0($a2)

add $v1, $v1, 1

j For4

EndFor4:

add $v0, $v0, 1

j For3

Case3:

beq $t1, 0, Case4

add $v0, $s1, $zero

For5:

bgt $v0, $s3, Exit

add $v1, $s2, $zero

For6:

bgt $v1, $s0, EndFor6

beq $v0, $s1, InVien3

beq $v0, $s3, InVien3

beq $v1, $s0, InVien3

beq $v1, $s2, InVien3

sll $t8, $v0, 6

add $t8, $t8, $v1

sll $t8, $t8, 2

li $a1, GREEN

add $a2, $k0, $t8

sw $a1, 0($a2)

add $v1, $v1, 1

j For6

InVien3:

sll $t8, $v0, 6

add $t8, $t8, $v1

sll $t8, $t8, 2

li $a1, RED

add $a2, $k0, $t8

sw $a1, 0($a2)

add $v1, $v1, 1

j For6

EndFor6:

add $v0, $v0, 1

j For5

Case4:

add $v0, $s3, $zero

For7:

bgt $v0, $s1, Exit

add $v1, $s2, $zero

For8:

bgt $v1, $s0, EndFor8

beq $v0, $s1, InVien4

beq $v0, $s3, InVien4

beq $v1, $s0, InVien4

beq $v1, $s2, InVien4

sll $t8, $v0, 6

add $t8, $t8, $v1

sll $t8, $t8, 2

li $a1, GREEN

add $a2, $k0, $t8

sw $a1, 0($a2)

add $v1, $v1, 1

j For8

InVien4:

sll $t8, $v0, 6

add $t8, $t8, $v1

sll $t8, $t8, 2

li $a1, RED

add $a2, $k0, $t8

sw $a1, 0($a2)

add $v1, $v1, 1

j For8

EndFor8:

add $v0, $v0, 1

j For7

Exit:

li $v0, 10

syscall

**Kết quả:**

Nhập điểm 1 có toạ độ (15,38) và điểm 2 có toạ độ (45,57)

A screenshot of a computer

Description automatically generated