Laboratory Exercise 11

Interrupts & IO programming

Assignment 1

**# Source code đã chỉnh sửa**

.data

.eqv IN\_ADRESS\_HEXA\_KEYBOARD 0xFFFF0012

.eqv OUT\_ADRESS\_HEXA\_KEYBOARD 0xFFFF0014

.text

main: li $t1, IN\_ADRESS\_HEXA\_KEYBOARD

li $t2, OUT\_ADRESS\_HEXA\_KEYBOARD

li $t3, 0x01 # check row 4 with key C, D, E, F

li $t4, 0x08

polling:

sb $t3, 0($t1)

lb $a0, 0($t2)

print:

li $v0, 34

syscall

sleep:

li $a0, 100

li $v0, 32

syscall

back\_to\_polling:

beq $t3, $t4, reset

sll $t3, $t3, 1

j polling

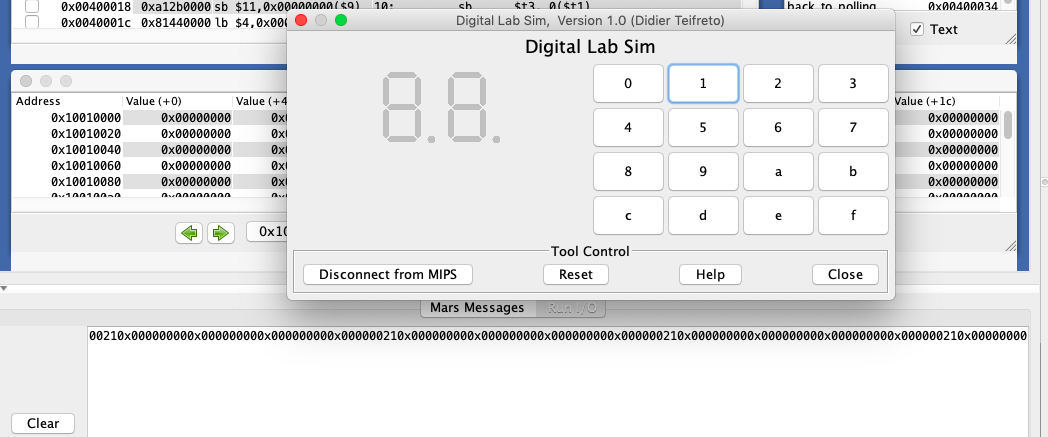
reset:

li $t3, 0x01

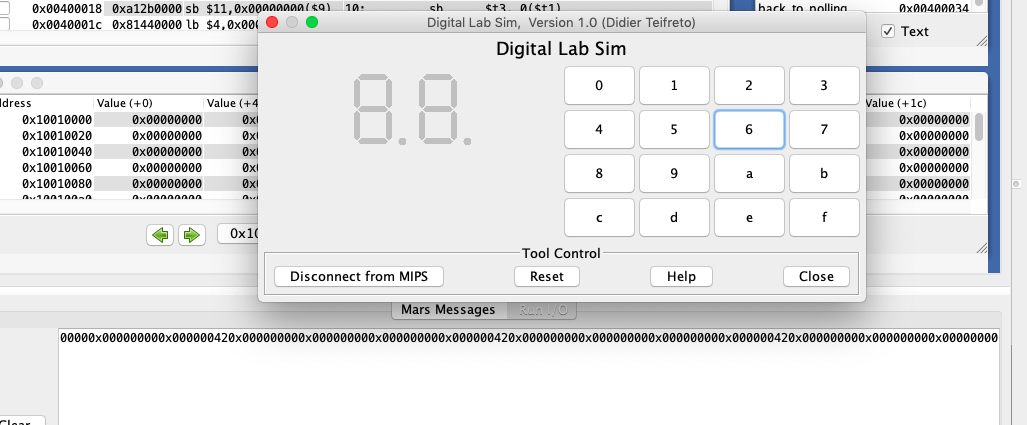
jal polling

**# Demo**

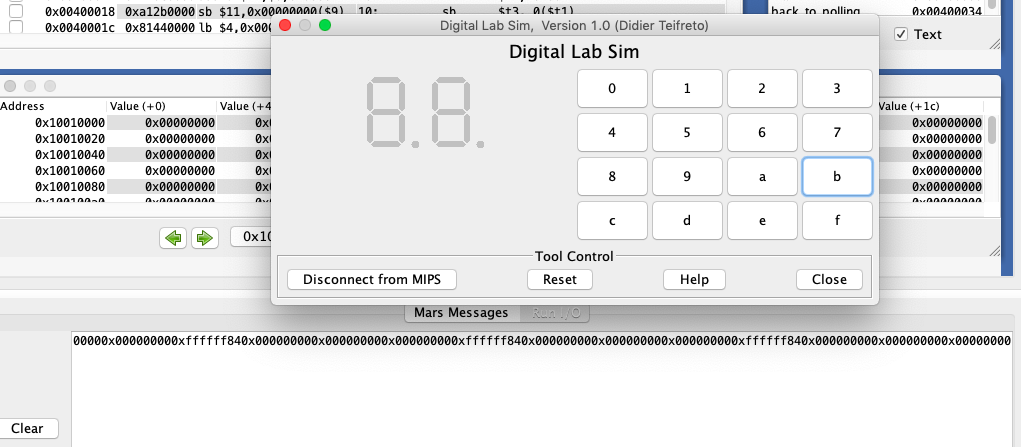
1. Row 1



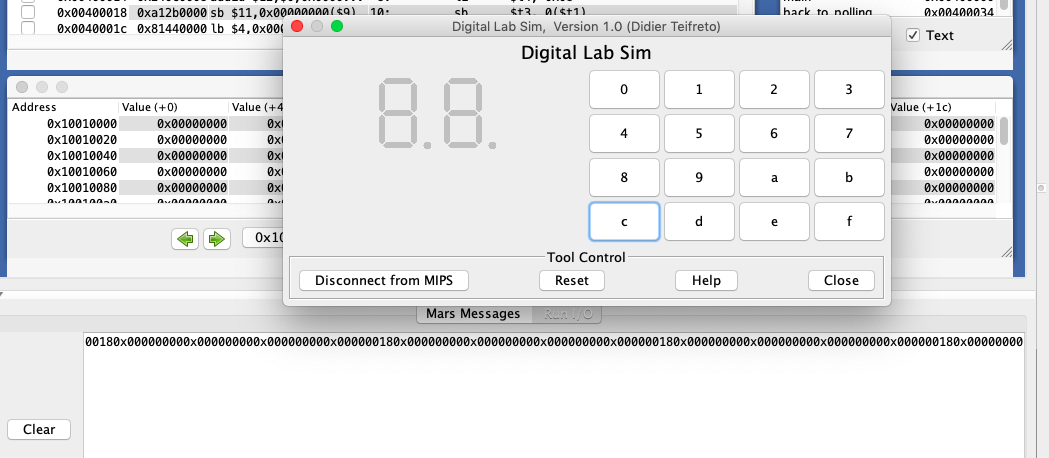
1. Row 2



1. Row 3



1. Row 4



Assignment 2

**# Source Code**

.eqv IN\_ADRESS\_HEXA\_KEYBOARD 0xFFFF0012

.data

Message: .asciiz "Oh my god. Someone's presed a button.\n"

.text

main:

li $t1, IN\_ADRESS\_HEXA\_KEYBOARD

li $t3, 0x80 # bit 7 of = 1 to enable interrupt

sb $t3, 0($t1)

Loop:

nop

nop

nop

nop

b Loop

end\_main:

.ktext 0x80000180

IntSR:

addi $v0, $zero, 4 # show message

la $a0, Message

syscall

next\_pc:

mfc0 $at, $14

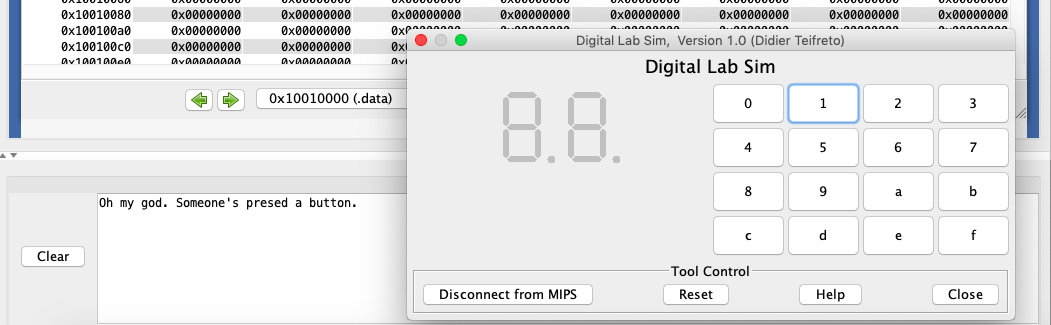
addi $at, $at, 4

mtc0 $at, $14

return:

eret

**# Demo**



**# Giải thích:**

* Ấn một kí tự bất kỳ bằng thiết bị Digital Lab Sim thì message sẽ được in ra màn hình.

Assignment 3

**# Source code đã sửa đổi**

.eqv IN\_ADRESS\_HEXA\_KEYBOARD 0xFFFF0012

.eqv OUT\_ADRESS\_HEXA\_KEYBOARD 0xFFFF0014

.data

Message: .asciiz "Key scan code "

.text

main:

li $t1, IN\_ADRESS\_HEXA\_KEYBOARD

li $t3, 0x80 # bit 7 = 1 to enable

li $t4, 1

sb $t3, 0($t1)

xor $s0, $s0, $s0

Loop: addi $s0, $s0, 1

prn\_seq:

addi $v0, $zero, 1

add $a0, $s0, $zero

syscall

prn\_eol:

addi $v0, $zero, 11

li $a0, '\n'

syscall

sleep:

addi $v0, $zero, 32

li $a0, 300

syscall

nop

b Loop

end\_main:

.ktext 0x80000180

IntSR:

addi $sp, $sp, 4

sw $ra, 0($sp)

addi $sp, $sp, 4

sw $at, 0($sp)

addi $sp, $sp, 4

sw $v0, 0($sp)

addi $sp, $sp, 4

sw $a0, 0($sp)

addi $sp, $sp, 4

sw $t1, 0($sp)

addi $sp, $sp, 4

sw $t3, 0($sp)

prn\_msg:

addi $v0, $zero, 4

la $a0, Message

syscall

get\_cod:

li $t1, IN\_ADRESS\_HEXA\_KEYBOARD

li $t3, 0x80

add $t3, $t3, $t4

sb $t3, 0($t1)

li $t1, OUT\_ADRESS\_HEXA\_KEYBOARD

lb $a0, 0($t1)

beq $a0, $zero, continue # Nếu output = 0 thì chuyển qua # hàng khác

j prn\_cod # Output # 0 🡪 kết quả trên # hàng đó

continue:

sll $t4, $t4, 1 # tăng index hàng thêm 1

beq $t4, 8, prn\_cod # 80 + i\*2 = hexa của hàng

j get\_cod # tiếp theo

prn\_cod:

li $v0, 34

syscall

li $v0, 11

li $a0, '\n'

syscall

next\_pc:

mfc0 $at, $14

addi $at, $at, 4

mtc0 $at, $14

restore:

lw $t3, 0($sp)

addi $sp, $sp, -4

lw $t1, 0($sp)

addi $sp, $sp, -4

lw $a0, 0($sp)

addi $sp, $sp, -4

lw $v0, 0($sp)

addi $sp, $sp, -4

lw $ra, 0($sp)

addi $sp, $sp, -4

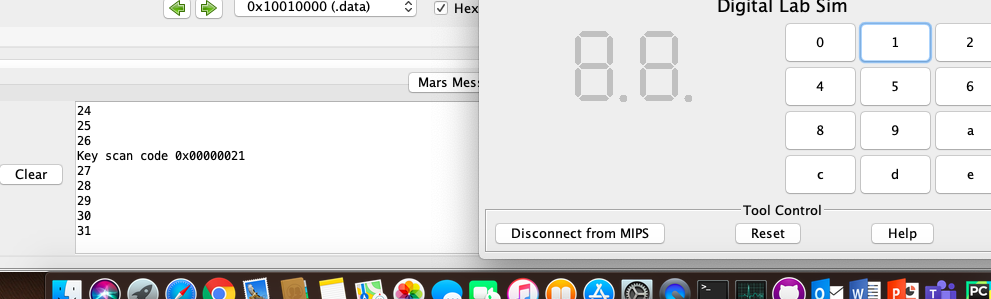
lw $ra, 0($sp)

addi $sp, $sp, -4

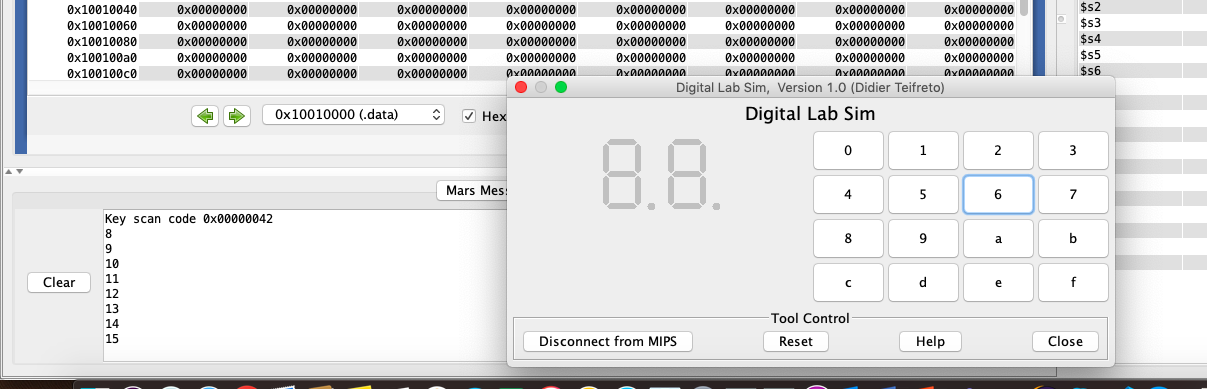
return: eret

🡪 Sửa code hàm get\_cod

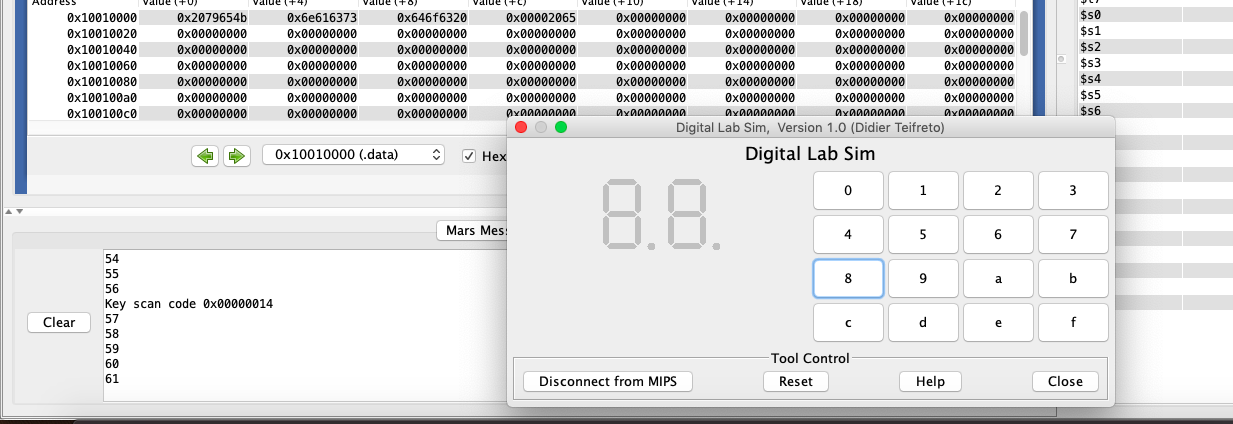
1. Row 1



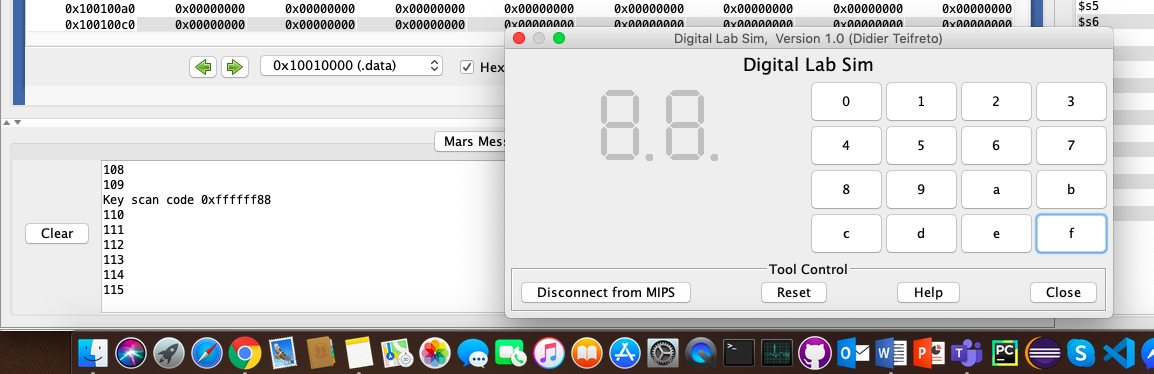
1. Row 2



1. Row 3



1. Row 4



Assignment 4

**# Source Code**

.eqv IN\_ADRESS\_HEXA\_KEYBOARD 0xFFFF0012

.eqv COUNTER 0xFFFF0013

.eqv MASK\_CAUSE\_COUNTER 0x00000400

.eqv MASK\_CAUSE\_KEYMATRIX 0x00000800

.data

msg\_keypress: .asciiz "Someone has pressed a key!\n"

msg\_counter: .asciiz "Time inteval!\n"

.text

main:

li $t1, IN\_ADRESS\_HEXA\_KEYBOARD

li $t3, 0x80 # bit 7 = 1 to enable

sb $t3, 0($t1)

li $t1, COUNTER

sb $t1, 0($t1)

Loop:

nop

nop

nop

sleep: addi $v0, $zero, 32

li $a0, 200

syscall

nop

b Loop

end\_main:

.ktext 0x80000180

IntSR:

dis\_int:

li $t1, COUNTER # BUG: must disable with Time Counter

sb $zero, 0($t1)

get\_caus:

mfc0 $t1, $13

IsCount:

li $t2, MASK\_CAUSE\_COUNTER

and $at, $t1, $t2

beq $at, $t2, Counter\_Intr

IsKeyMa:

li $t2, MASK\_CAUSE\_KEYMATRIX

and $at, $t1, $t2

beq $at, $t2, Keymatrix\_Intr

others: j end\_process

Keymatrix\_Intr:

li $v0, 4 # Processing Key Matrix Interrupt

la $a0, msg\_keypress

syscall

j end\_process

Counter\_Intr:

li $v0, 4 # Processing Counter Interrupt

la $a0, msg\_counter

syscall

j end\_process

end\_process:

mtc0 $zero, $13

en\_int:

li $t1, COUNTER

sb $t1, 0($t1)

next\_pc:

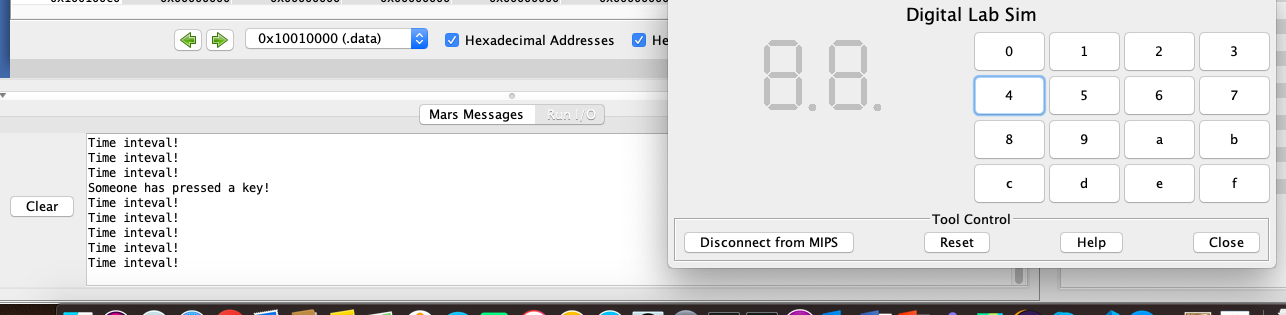
mfc0 $at, $14

addi $at, $at, 4

mtc0 $at, $14

return: eret

**# Demo**

****

**# Giải thích:**

* Vòng lặp mãi mãi chỉ kết thúc nếu người dùng ấn dừng.
* Sau mỗi khoảng thời gian ngắn thì message “Time interval!” sẽ được in ra console.
* Tuy nhiên nếu người dùng nhập số bằng thiết bị Digital Lab Sim thì message “Some one has pressed a key!” sẽ được in ra console.

Assignment 5

**# Source Code**

.eqv KEY\_CODE 0xFFFF0004

.eqv KEY\_READY 0xFFFF0000

.eqv DISPLAY\_CODE 0xFFFF000C

.eqv DISPLAY\_READY 0xFFFF0008

.eqv MASK\_CAUSE\_KEYBOARD 0x0000034

.text

li $k0, KEY\_CODE

li $k1, KEY\_READY

li $s0, DISPLAY\_CODE

li $s1, DISPLAY\_READY

loop: nop

WaitForKey:

lw $t1, 0($k1)

beq $t1, $zero, WaitForKey

MakeIntR:

teqi $t1, 1

j loop

.ktext 0x80000180

get\_caus:

mfc0 $t1, $13

IsCount:

li $t2, MASK\_CAUSE\_KEYBOARD

and $at, $t1, $t2

beq $at, $t2, Counter\_Keyboard

j end\_process

Counter\_Keyboard:

ReadKey:

lw $t0, 0($k0)

WaitForDis:

lw $t2, 0($s1)

beq $t2, $zero, WaitForDis

Encrypt:

addi $t0, $t0, 1

ShowKey:

sw $t0, 0($s0)

nop

end\_process:

next\_pc:

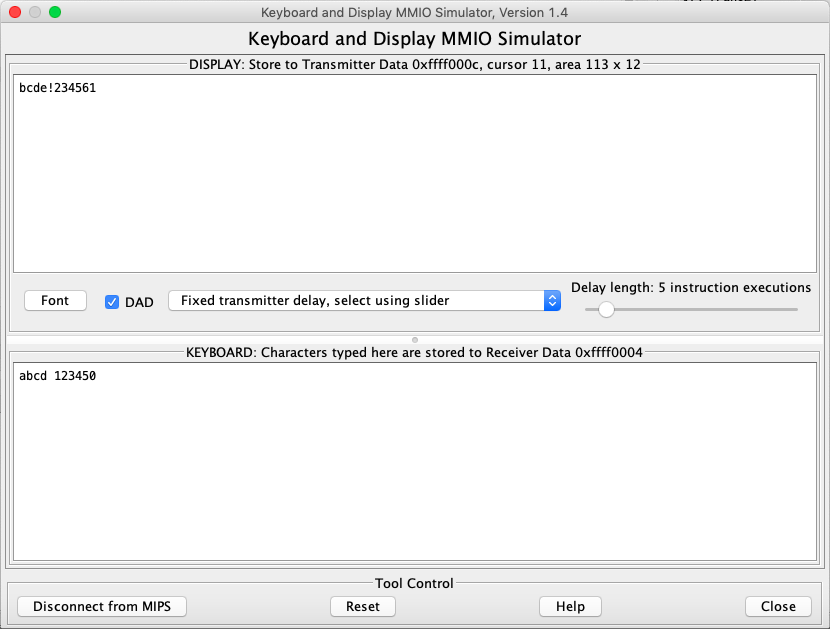
mfc0 $at, $14

addi $at, $at, 4

mtc0 $at, $14

return: eret

**# Demo**

****

**# Giải thích**

* Chương trình chỉ kết thúc khi người dùng ấn dừng.
* Các kí tự nhập vào sẽ được tính bằng mã ASCII của kí tự đó + 1.