ASSIGNMENT LAB 11

-Họ tên: Hồ Tuấn Huy
-MSSV: 20225856

Lab 1:

Assignment 1

Create a new project, type in, and build the program of Home Assignment 1.
Upgrade the source code so that it could defect all 16 key buttons, from 0 to F.

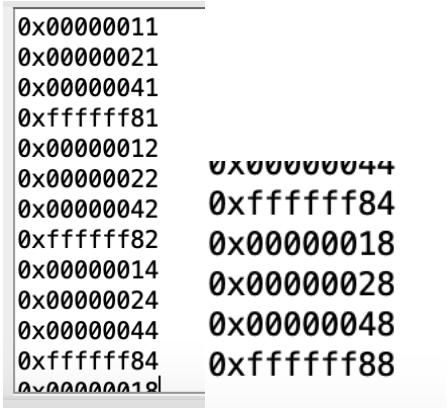
Code:

```
#-----------------------------------------------------
# col 0x1 col 0x2 col 0x4 col 0x8
#
# row 0x1 0 1 2 3
# 0x11 0x21 0x41 0x81
#
# row 0x2 4 5 6 7
# 0x12 0x22 0x42 0x82
#
# row 0x4 8 9 a b
# 0x14 0x24 0x44 0x84
#
# row 0x8 c d e f
# 0x18 0x28 0x48 0x88
#
#-----------------------------------------------------
# command row number of hexadecimal keyboard (bit 0 to 3)
# Eg. assign 0x1, to get key button 0,1,2,3
# assign 0x2, to get key button 4,5,6,7
# NOTE must reassign value for this address before reading,
# eventhough you only want to scan 1 row
# receive row and column of the key pressed, 0 if not key pressed
# Eg. equal 0x11, means that key button 0 pressed.
# Eg. equal 0x28, means that key button D pressed.
.data
n: .asciiz "\n"
.text
main:
li $t1, 0xFFFF0012
li $t2, 0xFFFF0014
li $s0, 0x10
li $t3, 0x01
li $t4, 0x02
li $t5, 0x04
```

```
li $t6, 0x08
li $t0, 0
polling:
beq $t0, 100, exit
sb $t3, 0($t1 ) # must reassign expected row
lb $a0, 0($t2) # read scan code of key button
bne $a0, $0, print
sb $t4, 0($t1)
lb $a0, 0($t2)
bne $a0, $0, print
sb $t5, 0($t1)
lb $a0, 0($t2)
bne $a0, $0, print
sb $t6, 0($t1)
lb $a0, 0($t2)
bne $a0, $0, print
j continue

print:
li $v0, 34 # print integer (hexa)
syscall
la $a0, n
li $v0, 4
syscall
continue: addi $a0, $t0, 1
sleep:
li $a0, 3000 # sleep 3000ms
li $v0, 32
syscall
back_to_polling:
j polling
exit:
```

Kết quả: Tương ứng với 16 phím được bấm từ 0 đến f ta thu được:



Giải thích:

- Trước hết ta khởi tạo: thiết lập các địa chỉ và giá trị cần thiết.
- Vòng lặp để kiểm tra:
  + Ghi các giá trị $t3 đến $t6 lần lượt vào địa chỉ $t1.
  + Đọc giá trị từ địa chỉ $t2 vào $a0.
  + Nếu giá trị đọc được khác 0 thì in ra giá trị đó, lặp lại quá trình cho đến khi đọc được 0.
- Chờ 3000ms=3s mỗi lần kiểm tra.

Lab 2:

Assignment 2

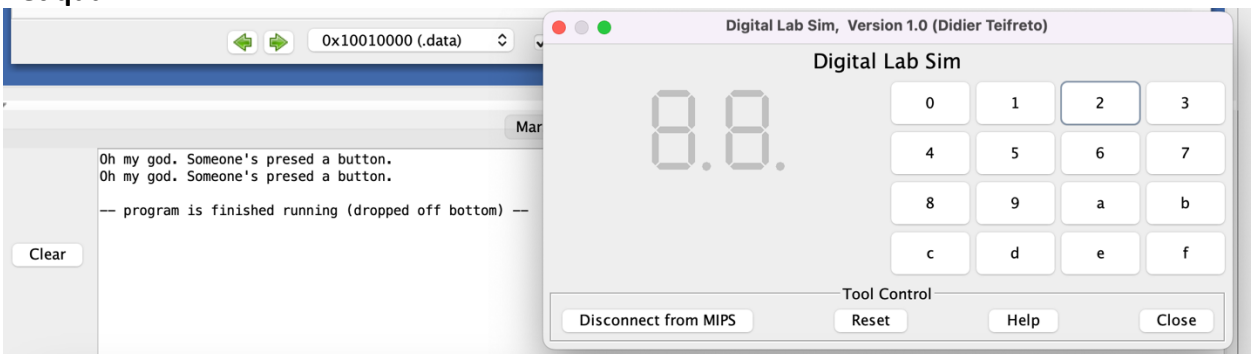
Create a new project, type in, and build the program of Home Assignment 2.
Upgrade the source code so that it could defect all 16 key buttons, from 0 to F.

Code:

```
.eqv IN_ADRESS_HEXA_KEYBOARD 0xFFFF0012
.data
Message: .asciiz "Oh my god. Someone's presed a button.\n"
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# MAIN Procedure
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
.text
main:
#-------------------------------------------------------
# Enable interrupts you expect
#-------------------------------------------------------
# Enable the interrupt of Keyboard matrix 4x4 of Digital Lab Sim
li $t1, IN_ADRESS_HEXA_KEYBOARD
li $t3, 0x80 # bit 7 of = 1 to enable interrupt
sb $t3, 0($t1)
#-------------------------------------------------------
# No-end loop, main program, to demo the effective of interrupt
#-------------------------------------------------------
Loop: nop
nop
nop
nop
b Loop # Wait for interrupt
end_main:
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# GENERAL INTERRUPT SERVED ROUTINE for all interrupts
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
.ktext 0x80000180
#---------------------------------------------------------
# Processing
#---------------------------------------------------------
```

```
IntSR: addi $v0, $zero, 4 # show message
la $a0, Message
syscall
#---------------------------------------------------------
# Evaluate the return address of main routine
# epc <= epc + 4
#---------------------------------------------------------
next_pc:mfc0 $at, $14 # $at <= Coproc0.$14 = Coproc0.epc
addi $at, $at, 4 # $at = $at + 4 (next instruction)
mtc0 $at, $14 # Coproc0.$14 = Coproc0.epc <= $at
return: eret # Return from exception
```

Kết quả:



Giải thích:

- Load địa chỉ của bàn phím vào thanh ghi $t1.
- Load giá trị 0x80 vào thanh ghi $t3, giá trị này có bit thứ 7 là 1 dùng để kích hoạt ngắt.
- Ghi giá trị trong $t3 vào địa chỉ trong $t1 (kích hoạt ngắt cho bàn phím).
- Vòng lặp chính dùng để chờ ngắt xảy ra.
- Trình phục vụ ngắt dùng để in thông báo ra màn hình và chuẩn bị trở về chương trình chính (khi ngắt xảy ra).
- Sau đó cập nhật địa chỉ trở về và tiếp tục thực thi chương trình chính sau khi ngắt được xử lý.

Lab 3:

Assignment 3

Create a new project, type in, and build the program of Home Assignment 3.
Upgrade the source code so that it could defect all 16 key buttons, from 0 to F.

Code:

```
.eqv IN_ADRESS_HEXA_KEYBOARD 0xFFFF0012
.eqv OUT_ADRESS_HEXA_KEYBOARD 0xFFFF0014
.data
Message: .asciiz "Key scan code "
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# MAIN Procedure
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
.text
main:
```

```
 #-----------------------------------------------------
 # Enable interrupts you expect
 #-----------------------------------------------------
 # Enable the interrupt of Keyboard matrix 4x4 of Digital Lab
 li $t1, IN_ADRESS_HEXA_KEYBOARD
 li $t3, 0x80 # bit 7 = 1 to enable
 sb $t3, 0($t1)
#-----------------------------------------------------
 # Loop an print sequence numbers
 #-----------------------------------------------------
 xor $s0, $s0, $s0 # count = $s0 = 0
Loop: addi $s0, $s0, 1 # count = count + 1
prn_seq:addi $v0,$zero,1
 add $a0,$s0,$zero # print auto sequence number
 syscall
prn_eol:addi $v0,$zero,11
 li $a0,'\n' # print endofline
 syscall
sleep: addi $v0,$zero,32
 li $a0,300 # sleep 300 ms
 syscall
 nop # WARNING: nop is mandatory here.
 b Loop # Loop
end_main:
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# GENERAL INTERRUPT SERVED ROUTINE for all interrupts
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
.ktext 0x80000180
 #-----------------------------------------------------
 # SAVE the current REG FILE to stack
 #-----------------------------------------------------
IntSR: addi $sp,$sp,4 # Save $ra because we may change it later
 sw $ra,0($sp)
 addi $sp,$sp,4 # Save $ra because we may change it later
 sw $at,0($sp)
 addi $sp,$sp,4 # Save $ra because we may change it later
 sw $v0,0($sp)
 addi $sp,$sp,4 # Save $a0, because we may change it later
 sw $a0,0($sp)
 addi $sp,$sp,4 # Save $t1, because we may change it later
 sw $t1,0($sp)
 addi $sp,$sp,4 # Save $t3, because we may change it later
 sw $t3,0($sp)
 #-----------------------------------------------------
```

```
 # Processing
 #-----------------------------------------------------
prn_msg:addi $v0, $zero, 4
 la $a0, Message
 syscall
get_cod:li $t1, IN_ADRESS_HEXA_KEYBOARD
 li $t3, 0x81 # check row 1and re-enable
 sb $t3, 0($t1) # must reassign expected row
 li $t1, OUT_ADRESS_HEXA_KEYBOARD
 lb $a0, 0($t1)
 bne $a0,$zero,prn_cod

 li $t1, IN_ADRESS_HEXA_KEYBOARD
 li $t3, 0x82 # check row 2and re-enable
 sb $t3, 0($t1) # must reassign expected row
 li $t1, OUT_ADRESS_HEXA_KEYBOARD
 lb $a0, 0($t1)
 bne $a0,$zero,prn_cod

 li $t1, IN_ADRESS_HEXA_KEYBOARD
 li $t3, 0x84 # check row 3and re-enable
 sb $t3, 0($t1) # must reassign expected row
 li $t1, OUT_ADRESS_HEXA_KEYBOARD
 lb $a0, 0($t1)
 bne $a0,$zero,prn_cod

 li $t1, IN_ADRESS_HEXA_KEYBOARD
 li $t3, 0x88 # check row 4 and re-enable
 sb $t3, 0($t1) # must reassign expected row
 li $t1, OUT_ADRESS_HEXA_KEYBOARD
 lb $a0, 0($t1)
 bne $a0,$zero,prn_cod

prn_cod:li $v0,34
 syscall
 li $v0,11
 li $a0,'\n' # print endofline
 syscall
 #-----------------------------------------------------
 # Evaluate the return address of main routine
 # epc <= epc + 4
 #-----------------------------------------------------
next_pc:mfc0 $at, $14 # $at <= Coproc0.$14 = Coproc0.epc
 addi $at, $at, 4 # $at = $at + 4 (next instruction)
```

```
 mtc0 $at, $14 # Coproc0.$14 = Coproc0.epc <= $at
 #-----------------------------------------------------
 # RESTORE the REG FILE from STACK
 #-----------------------------------------------------
restore:lw $t3, 0($sp) # Restore the registers from stack
 addi $sp,$sp,-4
 lw $t1, 0($sp) # Restore the registers from stack
 addi $sp,$sp,-4
 lw $a0, 0($sp) # Restore the registers from stack
 addi $sp,$sp,-4
 lw $v0, 0($sp) # Restore the registers from stack
 addi $sp,$sp,-4
 lw $ra, 0($sp) # Restore the registers from stack
 addi $sp,$sp,-4
return: eret # Return from exception
```

Kết quả: Khi nhập 9 và 7:

Giải thích: Chương trình liên tục quét bàn phím và in ra mã của các phím được nhấn trên màn hình:

- Trước hết ta có các hằng số địa chỉ, đại diện cho địa chỉ nhập và xuất.
- Phần .text-main procedure là mã lệnh chính của chương trình. Đầu tiên kích hoạt ngắt cho bàn phím và sau đó lặp lại việc in ra mã của các phím được nhấn. Mỗi lần lặp thì

chương trình in ra một số thứ tự tăng dần rồi in ra dòng mới và tạm dừng trong một khoảng thời gian.

- Phần xử lý ngắt chung thì khi có ngắt, chương trình sẽ kiếm tra từng hàng của bàn phím và in ra mã các phím được nhấn, trong trường hợp trên là mã 9 và 7.
- Phần get_cod kiểm tra từng hàng của bàn phím và in ra mã các phím được nhấn tương ứng từng hàng.
- next_pc là phần tính toán địa chỉ trả về cho chương trình sau khi xử lý xong một ngắt.
- Phần restore để khôi phục các thanh ghi từ stack sau khi xử lý xong một ngắt.