Laboratory Exercise 11

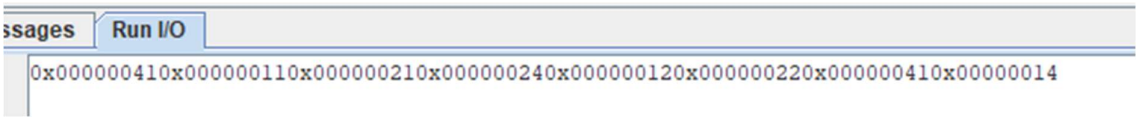
Interrupts & IO programming

Đỗ Hải Dương – 20194528

Assignment 1

```
.eqv IN_ADRESS_HEXA_KEYBOARD 0xFFFF0012
# 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.
.eqv OUT_ADRESS_HEXA_KEYBOARD 0xFFFF0014
.text
main:
        li $t1, IN_ADRESS_HEXA_KEYBOARD
        li $t2, OUT_ADRESS_HEXA_KEYBOARD
        li $t3, 0x1     # check row 1 with key 0, 1, 2, 3
        li $t4, 0x2     # check row 2 with key 4, 5, 6, 7
        li $t5, 0x4     # check row 3 with key 8, 9, a, b
        li $t6, 0x8     # check row 4 with key c, d, e, f

polling:
polling_row1:
        sb $t3, 0($t1)  # must reassign expected row
        lb $a0, 0($t2)  # read scan code of key button
        bnez $a0, print

polling_row2:
        sb $t4, 0($t1)  # must reassign expected row
        lb $a0, 0($t2)  # read scan code of key button
        bnez $a0, print

polling_row3:
        sb $t5, 0($t1)  # must reassign expected row
        lb $a0, 0($t2)  # read scan code of key button
        bnez $a0, print

polling_row4:
        sb $t6, 0($t1)  # must reassign expected row
        lb $a0, 0($t2)  # read scan code of key button
        bnez $a0, print

print:  li $v0, 34      # print integer (hexa)
        syscall

sleep:  li $a0, 100     # sleep 100ms
        li $v0, 32
        syscall


back_to_polling:
        j polling       # continue polling

```

Kết quả : 20194528



Giải thích :

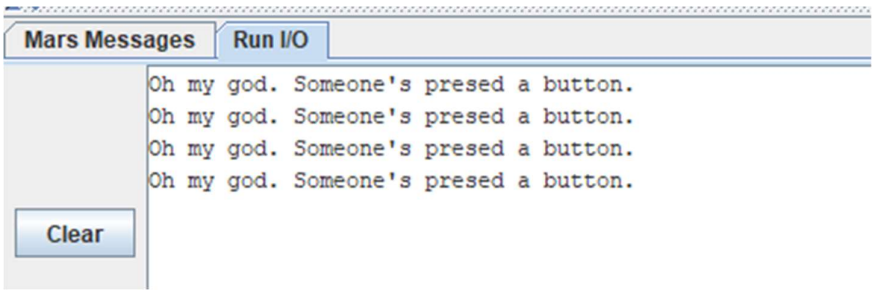
- Lưu các giá trị lần lượt t1 là địa chỉ của chỉ số dòng , t2 là địa chỉ phím được ấn để in ra màn hình .
- Lưu t3, t4, t5, t6 lần lượt cho các chỉ số của dòng 1 (gồm 0,1,2,3), 2 (4, 5, 6, 7), 3 (8, 9, a, b) và 4 (c, d, e, f)
- Vòng lặp polling :
  - ⇨ Kiểm tra từng dòng xem phím được nhập có thuộc dòng đó hay không, bằng cách check a0 . Nếu a0 khác 0 thì phím đó vừa được nhấn , in ra phím được nhấn bằng print.

Assignment 2

```
1  .eqv IN_ADRESS_HEXA_KEYBOARD 0xFFFF0012
2
3  .data
4  Message: .asciiz "Oh my god. Someone's presed a button.\n"
5  #~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
6  # MAIN Procedure
7  #~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
8  .text
9  main:
10         #-------------------------------------------------------------
11         # Enable interrupts you expect
12         #-------------------------------------------------------------
13         # Enable the interrupt of Keyboard matrix 4x4 of Digital Lab Sim
14         li $t1, IN_ADRESS_HEXA_KEYBOARD
15         li $t3, 0x80            # bit 7 of = 1 to enable interrupt
16         sb $t3, 0($t1)
17         #-------------------------------------------------------------
18         # No-end loop, main program, to demo the effective of interrupt
19         #-------------------------------------------------------------
20 Loop:    nop
21         nop
```

```
22          nop
23          nop
24          b Loop                      # Wait for interrupt
25  end_main:
26  #~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
27  # GENERAL INTERRUPT SERVED ROUTINE for all interrupts
28  #~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
29  .ktext 0x80000180
30          #---------------------------------------------------
31          # Processing
32          #---------------------------------------------------
33  IntSR:
34          addi $v0, $zero, 4          # show message
35          la $a0, Message
36          syscall
37          #---------------------------------------------------
38          # Evaluate the return address of main routine
39          # epc <= epc + 4
40          #---------------------------------------------------
41
42  next_pc:
43          mfc0 $at, $14 # $at <= Coproc0.$14 = Coproc0.epc
44          addi $at, $at, 4 # $at = $at + 4 (next instruction)
45          mtc0 $at, $14 # Coproc0.$14 = Coproc0.epc <= $at
46  return:
47          eret # Return from exception
48
```

Kết quả:



Giải thích :

- Lưu địa chỉ giá trị nhập từ bàn phím vào t1, t3=0x80, lưu thanh ghi t3 vào phần tử đầu tiên mang t1
- Nhảy đến địa chỉ 0x80000180 và viết code ở đây
- In messgage "Oh my god. Someone's presed a button.\n", sau đó lưu địa chỉ lệnh kế tiếp vào thanh ghi $14
- Gán nội dung thanh ghi $14 vào thanh ghi pc

Assignment 3

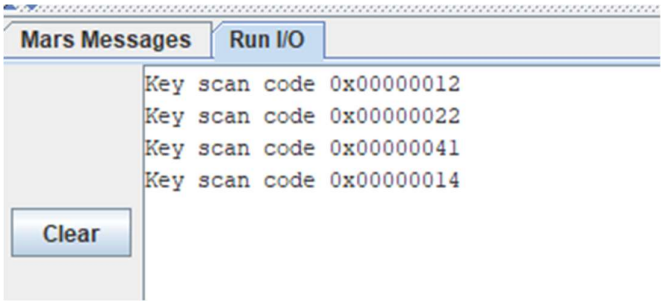
```
.eqv IN_ADRESS_HEXA_KEYBOARD 0xFFFF0012
.eqv OUT_ADRESS_HEXA_KEYBOARD 0xFFFF0014

.data
Message: .asciiz "Key scan code "

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# MAIN Procedure
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

.text
main:
        #-----------------------------------------------------------
        # Enable interrupts you expect
        #-----------------------------------------------------------
        # Enable the interrupt of Keyboard matrix 4x4 of Digital Lab Sim
        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 $at because we may change it later
        sw $at,0($sp)
        addi $sp,$sp,4 # Save $sp 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)
```

```
64          addi $sp,$sp,4 # Save $t1 because we may change it later
65          sw $t1,0($sp)
66          addi $sp,$sp,4 # Save $t3 because we may change it later
67          sw $t3,0($sp)
68
69          #-----------------------------------------------------
70          # Processing
71          #-----------------------------------------------------
72  prn_msg:
73          addi $v0, $zero, 4
74          la $a0, Message
75          syscall
76          li $t6, 0x1
77          li $t3, 0x81                        # check row 4 and re-enable bit 7
78  get_cod:
79          li  $t1, IN_ADRESS_HEXA_KEYBOARD
80          bgt $t3, 0x88, reset_getcod         # check row 4 and re-enable bit 7
81          sb  $t3, 0($t1)                     # must reassign expected row
82          li  $t1, OUT_ADRESS_HEXA_KEYBOARD
83          lb  $a0, 0($t1)
84          bnez $a0, prn_cod
```

```
85          mul $t6, $t6, 2
86          add $t3, $t6, 0x80
87          j get_cod
88  prn_cod:
89          li  $v0,34
90          syscall
91          li  $v0,11
92          li  $a0,'¥n'      # print endofline
93          syscall
94          #-----------------------------------------------------
95          # Evaluate the return address of main routine
96          # epc <= epc + 4
97          #-----------------------------------------------------
98  next_pc:
99          mfc0 $at, $14     # $at <= Coproc0.$14 = Coproc0.epc
100         addi $at, $at, 4 # $at = $at + 4 (next instruction)
101         mtc0 $at, $14     # Coproc0.$14 = Coproc0.epc <= $at
102         #-----------------------------------------------------
103         # RESTORE the REG FILE from STACK
104         #-----------------------------------------------------
105 restore:
```

```
106         lw $t3, 0($sp)    # Restore the registers from stack
107         addi $sp,$sp,-4
108         lw $t1, 0($sp)    # Restore the registers from stack
109         addi $sp,$sp,-4
110         lw $a0, 0($sp)    # Restore the registers from stack
111         addi $sp,$sp,-4
112         lw $v0, 0($sp)    # Restore the registers from stack
113         addi $sp,$sp,-4
114         lw $ra, 0($sp)    # Restore the registers from stack
115         addi $sp,$sp,-4
116         lw $ra, 0($sp)    # Restore the registers from stack
117         addi $sp,$sp,-4
118
119 return:
120         eret              # Return from exception
121
122 reset_getcod:
123         li  $t3, 0x81
124         li  $t6, 0x1
125         j get_cod
126
```

Kết quả: 4528



Giải thích:

- Bật theo dõi toàn bộ bàn phím . Tăng vòng lặp thêm 1 khi nhấn nút và in ra màn hình .
- Nhảy đến địa chỉ 0x80000180 và lưu các biến $ra, $at, $v0, $a0, $t1, $t3 vào stack .
- In đoạn massage"Kay scan code "
- Kiểm tra xem phím nhấn có phải dòng 1 không ? Đúng thì nhảy đến nhãn prn_cod
- Kiểm tra xem phím nhấn có phải dòng 2 không ? Đúng thì nhảy đến nhãn prn_cod
- Kiểm tra xem phím nhấn có phải dòng 3 không ? Đúng thì nhảy đến nhãn prn_cod
- Kiểm tra xem phím nhấn có phải dòng 4 không ? Đúng thì nhảy đến nhãn prn_cod
- Sau đó in mã của phím được nhấn và ký tự xuông dòng. Lưu địa chỉ lệnh kế tiếp vào thanh ghi $14.