

Laboratory Exercise 11

Interrupts & IO programming

Đỗ Hải Dương – 20194528

Assignment 4

Code:

```
1 .eqv IN_ADDRESS_HEX_KEYBOARD 0xFFFF0012
2 .eqv OUT_ADDRESS_HEX_KEYBOARD 0xFFFF0014
3 .eqv COUNTER 0xFFFF0013          # Time Counter
4 .eqv MASK_CAUSE_COUNTER 0x00000400 # Bit 10: Counter interrupt
5 .eqv MASK_CAUSE_KEYMATRIX 0x00000800 # Bit 11: Key matrix interrupt
6 .data
7 msg_keypress: .ascii "Someone has pressed a key!\n"
8 msg_counter: .ascii "Time interval!"
9 #-----
10 # MAIN Procedure
11 #-----
12 .text
13 main:
14 #-----
15 # Enable interrupts you expect
16 #-----
17 # Enable the interrupt of Keyboard matrix 4x4 of Digital Lab Sim
18     li $t1, IN_ADDRESS_HEX_KEYBOARD
19     li $t3, 0x80          # bit 7 = 1 to enable
20     sb $t3, 0($t1)
21 # Enable the interrupt of TimeCounter of Digital Lab Sim
22     addi $s5, $0, 0
23     li $t1, COUNTER
24     sb $t1, 0($t1)
25
26 #-----
27 # Loop and print sequence numbers
```

```

28  #-----
29  Loop:
30      nop
31      nop
32      nop
33
34  sleep:
35      addi $v0,$zero,32      # BUG: must sleep to wait for Time Counter
36      li $a0,1000           # sleep 300 ms
37      syscall
38      nop                   # WARNING: nop is mandatory here.
39      b Loop
40
41  end_main:
42  #-----
43  # GENERAL INTERRUPT SERVED ROUTINE for all interrupts
44  #-----
45  .ktext 0x80000180
46  IntSR:
47  #-----
48  # Temporary disable interrupt
49  #-----
50  dis_int:
51      li $t1, COUNTER      # BUG: must disable with Time Counter
52      sb $zero, 0($t1)
53      # no need to disable keyboard matrix interrupt
54  #-----

```

```

55  # Processing
56  #-----
57  get_caus:
58      mfc0 $t1, $13        # $t1 = Coproc0.cause
59
60  IsCount:
61      li $t2, MASK_CAUSE_COUNTER      # if Cause value confirm Counter..
62      and $at, $t1,$t2
63      beq $at,$t2, Counter_Intr
64
65  IsKeyMa:
66      li $t2, MASK_CAUSE_KEYMATRIX    # if Cause value confirm Key..
67      and $at, $t1,$t2
68      beq $at,$t2, Keymatrix_Intr
69
70  others:
71      j end_process      # other cases
72
73  Keymatrix_Intr:
74      li $v0, 4          # Processing Key Matrix Interrupt
75      la $a0, msg_keypress
76      syscall
77
78      li $t1, IN_ADRESS_HEX_A_KEYBOARD
79      li $t2, OUT_ADRESS_HEX_A_KEYBOARD
80
81      li $t3, 0x81      # check row 4 and re-enable bit 7
82

```

```

82     jal check
83
84     li $t3, 0x82      # check row 4 and re-enable bit 7
85     jal check
86
87     li $t3, 0x84      # check row 4 and re-enable bit 7
88     jal check
89
90     li $t3, 0x88      # check row 4 and re-enable bit 7
91
92 check:
93     sb $t3, 0($t1)    # must reassign expected row
94     lb $a0, 0($t2)
95     bne $a0, 0x0, prn_cod
96     jr $ra
97
98 prn_cod:
99     li $v0, 34
100    syscall
101    li $v0, 11
102    li $a0, '%n'      # print endofline
103    syscall
104    j end_process
105
106 Counter_Intr:
107     li $v0, 4          # Processing Counter Interrupt
108     la $a0, msg_counter
109
110
111     syscall
112
113     addi $s5, $s5, 1    # count = count + 1
114     addi $v0, $zero, 1
115     add $a0, $s5, $zero # print auto sequence number
116     syscall
117
118     li $v0, 11
119     li $a0, '%n'      # print endofline
120     syscall
121     j end_process
122
123 end_process:
124     mtc0 $zero, $13    # Must clear cause reg
125
126 en_int:
127     #-----
128     # Re-enable interrupt
129     #-----
130     li $t1, COUNTER
131     sb $t1, 0($t1)
132     #-----
133     # Evaluate the return address of main routine
134     # epc <= epc + 4
135     #-----
136
137 next_pc:
138     mfc0 $at, $14      # $at <= Coproc0.$14 = Coproc0.epc
139
140     addi $at, $at, 4    # $at = $at + 4 (next instruction)
141     mtc0 $at, $14      # Coproc0.$14 = Coproc0.epc <= $at
142
143 return: eret

```

Kết quả : 20194528

Mars Messages	Run I/O
<div>Clear</div>	Someone has pressed a key! 0x00000041 Time interval! 1
	Someone has pressed a key! 0x00000011 Time interval! 2
	Someone has pressed a key! 0x00000021 Time interval! 3
	Someone has pressed a key! 0x00000024 Time interval! 4
	Someone has pressed a key! 0x00000012 Time interval! 5
	Someone has pressed a key! 0x00000022 Time interval! 6
	Someone has pressed a key! 0x00000041 Time interval! 7
	Time interval! 8 Someone has pressed a key! 0x00000014

Assignment 5

Code:

```
1  .eqv KEY_CODE 0xFFFF0004      # ASCII code from keyboard, 1 byte
2  .eqv KEY_READY 0xFFFF0000     # =1 if has a new keycode ?
3                                  # Auto clear after lw
4  .eqv DISPLAY_CODE 0xFFFF000C  # ASCII code to show, 1 byte
5  .eqv DISPLAY_READY 0xFFFF0008 # =1 if the display has already to do
6                                  # Auto clear after sw
7  .eqv MASK_CAUSE_KEYBOARD 0x00000034 # Keyboard Cause
8  .text
9      li $k0, KEY_CODE
10     li $k1, KEY_READY
11
12     li $s0, DISPLAY_CODE
13     li $s1, DISPLAY_READY
14
15 loop:    nop
16
17 WaitForKey:
18     lw $t1, 0($k1)              # $t1 = [$k1] = KEY_READY
19     beq $t1, $zero, WaitForKey  # if $t1 == 0 then Polling
20
21 MakeIntR:
22     teqi $t1, 1                  # if $t0 = 1 then raise an Interrupt
23     j loop
24 #-----
25 # Interrupt subroutine
26 #-----
27 .ktext 0x80000180
28 get_caus:
29     mfc0 $t1, $13               # $t1 = Coproc0.cause
30
31 IsCount:
32     li $t2, MASK_CAUSE_KEYBOARD # if Cause value confirm Keyboard..
33     and $at, $t1, $t2
34     beq $at, $t2, Counter_Keyboard
35     j end_process
36
37 Counter_Keyboard:
38
39 ReadKey:
40     lw $t0, 0($k0)              # $t0 = [$k0] = KEY_CODE
41
42 WaitForDis:
43     lw $t2, 0($s1)              # $t2 = [$s1] = DISPLAY_READY
44     beq $t2, $zero, WaitForDis  # if $t2 == 0 then Polling
45
46 Encrypt:
47     addi $t0, $t0, 5            # change input key
```

```

48
49 ShowKey:
50     sw $t0, 0($s0)    # show key
51     nop
52 end_process:
53
54 next_pc:
55     mfc0 $at, $14      # $at <= Coproc0.$14 = Coproc0.epc
56     addi $at, $at, 4    # $at = $at + 4 (next instruction)
57     mtc0 $at, $14      # Coproc0.$14 = Coproc0.epc <= $at
58
59 return:    eret        # Return from exception
60

```

Kết quả:

