

Laboratory Exercise 11_2 – Report:

Interrupts & IO programming

Lê Văn Duẩn - 20194508

1. Assignment 4

- Mã nguồn:

```
1 .eqv IN_ADDRESS_HEX_A_KEYBOARD 0xFFFF0012
2 .eqv OUT_ADDRESS_HEX_A_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
7 .data
8 msg_keypress: .ascii "Someone has pressed a key!\n"
9 msg_counter: .ascii "Time interval! count: "
10
11 #-----
12 # HAIN Procedure
13 #-----
14
15 .text
16 main:
17 #-----
18 # Enable interrupts you expect
19 #-----
20 # Enable the interrupt of Keyboard matrix 4x4 of Digital Lab Sim
21
22 li $t1, IN_ADDRESS_HEX_A_KEYBOARD
23 li $t3, 0x80 # bit 7 = 1 to enable
24 sb $t3, 0($t1)
25
26 # Enable the interrupt of TimeCounter of Digital Lab Sim
27 li $k0, 0
28 li $t1, COUNTER
29 sb $t1, 0($t1)
30
31 #-----
32 # Loop an print sequence numbers
33 #-----
34
35 Loop: nop
36     nop
37     nop
```

```

38 sleep: addi $v0,$zero,32 # BUG: must sleep to wait for Time Counter
39 li $a0,200 # sleep 300 ms
40 syscall
41 nop # WARNING: nop is mandatory here.
42 b Loop
43 end_main:
44
45 #-----
46 # GENERAL INTERRUPT SERVED ROUTINE for all interrupts
47 #-----
48
49 .ktext 0x80000180
50
51 IntSR:
52 #-----
53 # Temporary disable interrupt
54 #-----
55
56 dis_int:li $t1, COUNTER # BUG: must disable with Time Counter
57 sb $zero, 0($t1)
58 # no need to disable keyboard matrix interrupt
59 #-----
60 # Processing
61 #-----
62
63 get_caus:mfc0 $t1, $13 # $t1 = Coproc0.cause
64 IsCount:li $t2, MASK_CAUSE_COUNTER # if Cause value confirm Counter..
65 and $at, $t1,$t2
66 beq $at,$t2, Counter_Intr
67
68 IsKeyMa:li $t2, MASK_CAUSE_KEYMATRIX # if Cause value confirm Key..
69 and $at, $t1,$t2
70 beq $at,$t2, Keymatrix_Intr
71
72 others: j end_process # other cases
73

```

```

74 Keymatrix_Intr: li $v0, 4 # Processing Key Matrix Interrupt
75
76 la $a0, msg_keypress
77 syscall
78
79 get_cod: li $t1, IN_ADRESS_HEX_A_KEYBOARD
80 li $t2, OUT_ADRESS_HEX_A_KEYBOARD
81 start_interrupt_1:
82 li $t3, 0x81 # check row 1 with key 0, 1, 2, 4
83 sb $t3, 0($t1) # must reassign expected row
84 jal interrupt
85
86 start_interrupt_2:
87 li $t3, 0x82 # check row 2 with key 4, 5, 6, 7
88 sb $t3, 0($t1) # must reassign expected row
89 jal interrupt
90
91 start_interrupt_3:
92 li $t3, 0x84 # check row 3 with key 8, 9, A, B
93 sb $t3, 0($t1) # must reassign expected row
94 jal interrupt
95
96 start_interrupt_4:
97 li $t3, 0x88 # check row 4 with key C, D, E, F
98 sb $t3, 0($t1) # must reassign expected row
99 jal end_process
100
101 check_after_interrupt_4:
102 beq $a0, 0x0, prn_cod
103 j next_pc
104
105 interrupt:
106 lb $a0, 0($t2) # read scan code of key button
107 bne $a0, 0x0, prn_cod
108 jr $ra
109 prn_cod:li $v0,34

```

```

110 syscall
111 li $v0,11
112 li $a0,'\n' # print endofline
113 syscall
114
115 j end_process
116
117 Counter_Intr: li $v0, 4 # Processing Counter Interrupt
118
119 la $a0, msg_counter
120 syscall
121 addi $k0, $k0, 1
122
123 li $v0, 1
124 add $a0, $0, $k0
125 syscall
126
127 li $v0,11
128 li $a0,'\n' # print endofline
129 syscall
130
131 j end_process
132
133 end_process:
134 mtc0 $zero, $13 # Must clear cause reg
135 en_int:
136 #-----
137 # Re-enable interrupt
138 #-----
139
140 li $t1, COUNTER
141 sb $t1, 0($t1)
142
143 #-----
144 # Evaluate the return address of main routine
145 # epc <= epc + 4
146
147 #-----
148 next_pc: mfc0 $at, $14 # $at <= Coproc0.$14 = Coproc0.epc
149 addi $at, $at, 4 # $at = $at + 4 (next instruction)
150 mtc0 $at, $14 # Coproc0.$14 = Coproc0.epc <= $at
151 return: eret # Return from exception
152

```

- Kết quả chạy mô phỏng:

```
Time interval! count: 1
Someone has pressed a key!
0x00000041
Someone has pressed a key!
0x00000011
Someone has pressed a key!
0x00000021
Someone has pressed a key!
0x00000024
Time interval! count: 2
Someone has pressed a key!
0x00000012
Someone has pressed a key!
0x00000022
Someone has pressed a key!
0x00000011
Someone has pressed a key!
0x00000014
Time interval! count: 3
Time interval! count: 4
```

Clear

- Giải thích:

. Chương trình cho phép ngắt đồng thời bằng 2 cách: từ bàn phím Lab Sim và bộ đếm thời gian của Lab Sim

Tại Coproc0, thanh ghi 13 lưu giá trị để phân biệt kiểu ngắt: 0x400 -> ngắt timer, 0x800 -> ngắt từ bàn phím

Khi kết nối với Lab Sim và ấn phím bất kì thì chương trình hiển thị message và địa chỉ tương ứng của số. Như ví dụ trên nhập và MSSV 20194508 thì khi ấn chương trình sẽ hiển thị từng số (2 -> 0x41, 0->0x11, 1->0x21, 9->0x24, ...)

Time interval! xuất hiện khi trong khoảng 200ms như định sẵn không có ngắt bằng cách ấn bàn phím thì sẽ in thông báo này ra màn hình.

2. Assignment 5

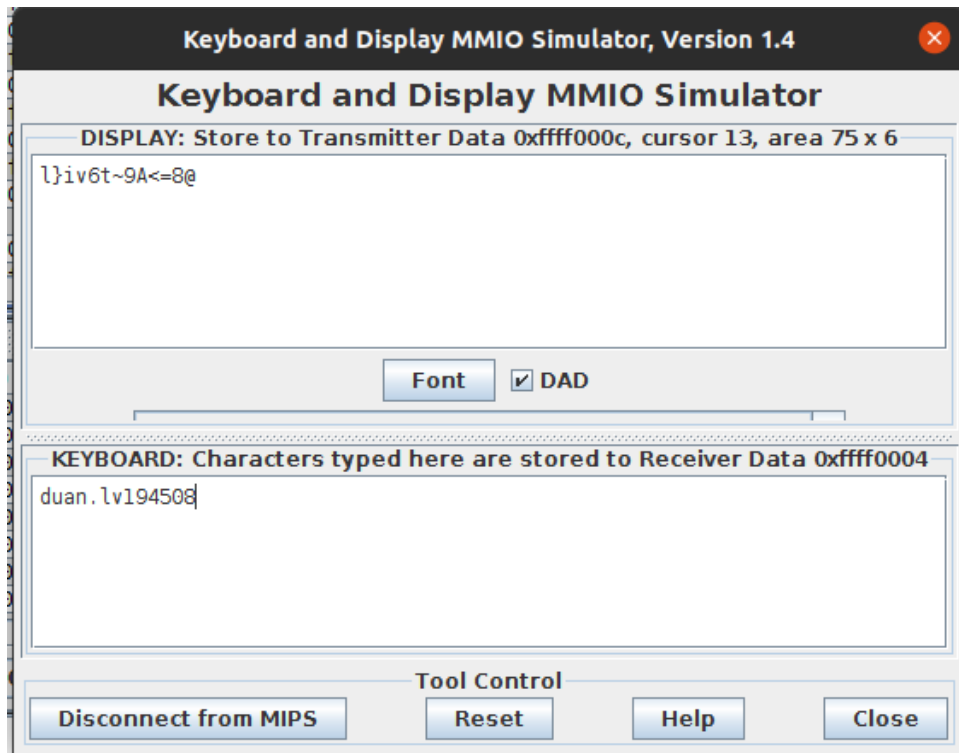
- Mã nguồn:

```

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
6  .eqv DISPLAY_READY 0xFFFF0008 # =1 if the display has already to do
7
8  # Auto clear after sw
9
10 .eqv MASK_CAUSE_KEYBOARD 0x00000034 # Keyboard Cause
11
12 .text
13
14 li $k0, KEY_CODE
15 li $k1, KEY_READY
16
17 li $s0, DISPLAY_CODE
18 li $s1, DISPLAY_READY
19
20 loop: nop
21 WaitForKey: lw $t1, 0($k1) # $t1 = [$k1] = KEY_READY
22 beq $t1, $zero, WaitForKey # if $t1 == 0 then Polling
23 MakeIntR: teqi $t1, 1 # if $t0 = 1 then raise an Interrupt
24 j loop
25
26 #-----
27 # Interrupt subroutine
28 #-----
29
30 .ktext 0x80000180
31
32 get_caus: mfc0 $t1, $13 # $t1 = Coproc0.cause
33
34 IsCount: li $t2, MASK_CAUSE_KEYBOARD # if Cause value confirm Keyboard..
35
36 and $at, $t1, $t2
37 beq $at, $t2, Counter_Keyboard
38
39
40 Counter_Keyboard:
41
42 ReadKey: lw $t0, 0($k0) # $t0 = [$k0] = KEY_CODE
43 WaitForDis: lw $t2, 0($s1) # $t2 = [$s1] = DISPLAY_READY
44 beq $t2, $zero, WaitForDis # if $t2 == 0 then Polling
45 Encrypt: addi $t0, $t0, 8 # change input key add 8 because HSSV 20194508
46 ShowKey: sw $t0, 0($s0) # show key
47 nop
48 end_process:
49
50 next_pc: mfc0 $at, $14 # $at <= Coproc0.$14 = Coproc0.epc
51 addi $at, $at, 4 # $at = $at + 4 (next instruction)
52 mtc0 $at, $14 # Coproc0.$14 = Coproc0.epc <= $at
53 return: eret # Return from exception
54

```

- Kết quả chạy:



- Giải thích:

Sử dụng teq hoặc teqi để cho phép ngắt mềm

Tool keyboard không tự tạo ra ngắt mềm khi bấm vì thế chúng ta cần sử dụng teq hoặc teqi

Chương trình sẽ cho phép ngắt mềm khi nhập kí tự vào keyboard và hiển thị mã hóa theo số cuối của MSSV (lệch 8 giá trị so với kí tự nhập vào vì mssv 20194508)