

Bài 4:

- Mã nguồn

```
1  .eqv IN_ADDRESS_HEXА_KEYBOARD 0xFFFF0012
2  .eqv OUT_ADDRESS_HEXА_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: .asciiz "Someone has pressed a key!\n"
9  msg_counter: .asciiz "Time interval!\n"
10
11 #~~~~~
12 # MAIN 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_HEXА_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 $t1, COUNTER
28 sb $t1, 0($t1)
29
30 #-----
31 # Loop an print sequence numbers
32 #-----
33
34 Loop: nop
35     nop
36     nop
37 sleep: addi $v0,$zero,32 # BUG: must sleep to wait for Time Counter
38     li $a0,200 # sleep 300 ms
39     syscall
40     nop # WARNING: nop is mandatory here.
41     b Loop
42 end_main:
```

```

43
44 #~~~~~
45 # GENERAL INTERRUPT SERVED ROUTINE for all interrupts
46 #~~~~~
47
48 .ktext 0x80000180
49
50 IntSR:
51 #-----
52 # Temporary disable interrupt
53 #-----
54
55 dis_int:li $t1, COUNTER # BUG: must disable with Time Counter
56 sb $zero, 0($t1)
57 # no need to disable keyboard matrix interrupt
58 #-----
59 # Processing
60 #-----
61
62 get_caus:mfc0 $t1, $13 # $t1 = Coproc0.cause
63 IsCount:li $t2, MASK_CAUSE_COUNTER # if Cause value confirm Counter..
64
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
82 start_interrupt_1:
83     li $t3, 0x81 # check row 1 with key 0, 1, 2, 4
84     sb $t3, 0($t1) # must reassign expected row
85     jal interrupt

```

```

85 start_interrupt_2:
86     li $t3, 0x82 # check row 2 with key 4, 5, 6, 7
87     sb $t3, 0($t1) # must reassign expected row
88     jal interrupt
89
90 start_interrupt_3:
91     li $t3, 0x84 # check row 3 with key 8, 9, A, B
92     sb $t3, 0($t1) # must reassign expected row
93     jal interrupt
94
95 start_interrupt_4:
96     li $t3, 0x88 # check row 4 with key C, D, E, F
97     sb $t3, 0($t1) # must reassign expected row
98     jal end_process
99
100 check_after_interrupt_4:
101     beq $a0, 0x0, prn_cod
102     j next_pc
103
104 interrupt:
105     lb $a0, 0($t2) # read scan code of key button
106     bne $a0, 0x0, prn_cod
107     jr $ra
108 prn_cod: li $v0, 34
109     syscall
110     li $v0, 11
111     li $a0, '\n' # print endofline
112     syscall
113
114     j end_process
115
116 Counter_Intr: li $v0, 4 # Processing Counter Interrupt
117
118     la $a0, msg_counter
119     syscall
120
121     j end_process
122
123 end_process:
124     mtc0 $zero, $13 # Must clear cause reg
125 en_int:
126     #-----
127     # Re-enable interrupt
128     #-----
129
130     li $t1, COUNTER
131     sb $t1, 0($t1)
132
133     #-----
134     # Evaluate the return address of main routine
135     # epc <= epc + 4
136     #-----
137
138 next_pc: mfc0 $at, $14 # $at <= Coproc0.$14 = Coproc0.epc
139     addi $at, $at, 4 # $at = $at + 4 (next instruction)
140     mtc0 $at, $14 # Coproc0.$14 = Coproc0.epc <= $at
141 return: eret # Return from exception

```

- Màn hình chạy

	Reset: reset completed.
Clear	Someone has pressed a key! 0x00000041 Time interval 1 Time interval 2
Clear	Time interval 2 Time interval 3 Time interval 4 Time interval 5 Time interval 6 Time interval 7
Clear	Time interval 7 Someone has pressed a key! 0x00000011 Time interval 8 Time interval 9 Time interval 10
Clear	Time interval 10 Someone has pressed a key! 0x00000021 Time interval 11 Time interval 12 Time interval 13
Clear	Time interval 13 Someone has pressed a key! 0x00000024 Time interval 14 Time interval 15 Time interval 16
Clear	Time interval 16 Someone has pressed a key! 0x00000012 Time interval 17 Time interval 18 Someone has pressed a key!

Clear	Someone has pressed a key!
	0x00000022
	Time interval 19
	Time interval 20
	Time interval 21
	Someone has pressed a key!
Clear	Someone has pressed a key!
	0x00000014
	Time interval 22
	Time interval 23
	Someone has pressed a key!
	0x00000014
Clear	0x00000014
	Time interval 24
	Time interval 25
	Time interval 26
	Time interval 27

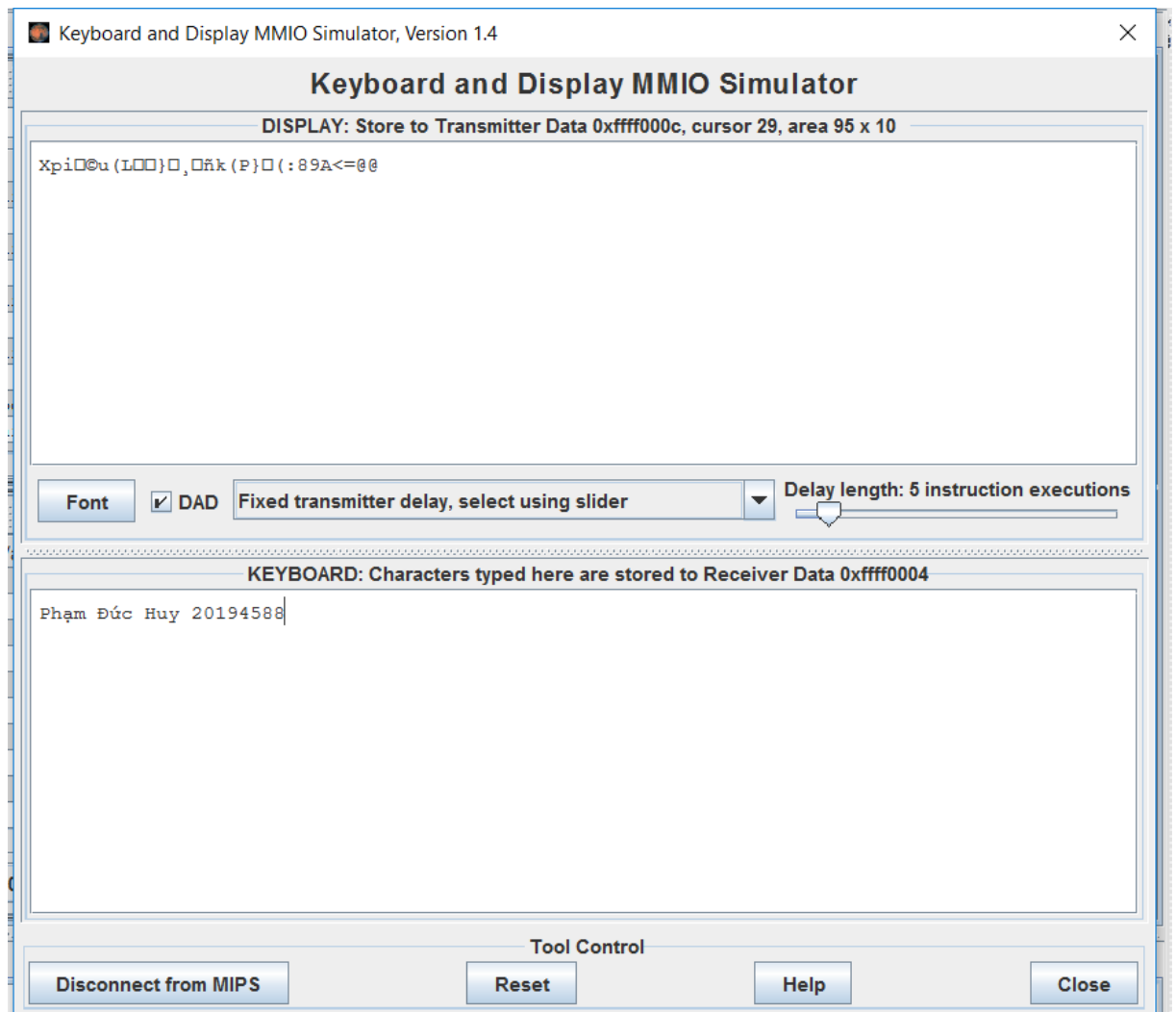
- Giải thích
 - Màn hình in ra các số tương ứng với MSSV
 - 0x41: 2
 - 0x11: 0
 - 0x21: 1
 - 0x24: 9
 - 0x12: 4
 - 0x22: 5
 - 0x14: 8
 - 0x14: 8
 - ⇒ Chương trình chạy đúng và in ra MSSV 20194588
 - Chương trình in ra thời gian => Sau 27 đơn vị thời gian thì in xong MSSV
 - Chương trình cho phép ngắt 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.
 - 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.

Bài 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 0x0000034 # 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
23 beq $t1, $zero, WaitForKey # if $t1 == 0 then Polling
24 MakeIntR: teqi $t1, 1 # if $t0 = 1 then raise an Interrupt
25
26 j loop
27
28 #-----
29 # Interrupt subroutine
30 #-----
31
32 .ktext 0x80000180
33
34 get_caus: mfc0 $t1, $13 # $t1 = Coproc0.cause
35
36 IsCount: li $t2, MASK_CAUSE_KEYBOARD # if Cause value confirm Keyboard..
37
38 and $at, $t1, $t2
39 beq $at, $t2, Counter_Keyboard
40
41 j end_process
42
43 Counter_Keyboard:
44
45 ReadKey: lw $t0, 0($k0) # $t0 = [$k0] = KEY_CODE
46
47 WaitForDis: lw $t2, 0($s1) # $t2 = [$s1] = DISPLAY_READY
48
49 beq $t2, $zero, WaitForDis # if $t2 == 0 then Polling
50 Encrypt: addi $t0, $t0, 8 # change input key add 8 because MSSV 20194508
51 ShowKey: sw $t0, 0($s0) # show key
52
53 nop
54 end_process:
55
56 next_pc: mfc0 $at, $14 # $at <= Coproc0.$14 = Coproc0.epc
57
58 addi $at, $at, 4 # $at = $at + 4 (next instruction)
59 mtc0 $at, $14 # Coproc0.$14 = Coproc0.epc <= $at
60
61 return: eret # Return from exception
```

- Màn hình 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