

Bài 1:

- Mã nguồn

```
1  #-----
2  # col 0x1 col 0x2 col 0x4 col 0x8
3  #
4  # row 0x1      0      1      2      3
5  #              0x11   0x21   0x41   0x81
6  #
7  # row 0x2      4      5      6      7
8  #              0x12   0x22   0x42   0x82
9  #
10 # row 0x4      8      9      a      b
11 #              0x14   0x24   0x44   0x84
12 #
13 # row 0x8      c      d      e      f
14 #              0x18   0x28   0x48   0x88
15 #
16 #-----
17 # command row number of hexadecimal keyboard (bit 0 to 3)
18 # Eg. assign 0x1, to get key button 0,1,2,3
19 # assign 0x2, to get key button 4,5,6,7
20 # NOTE must reassign value for this address before reading,
21 # eventhough you only want to scan 1 row

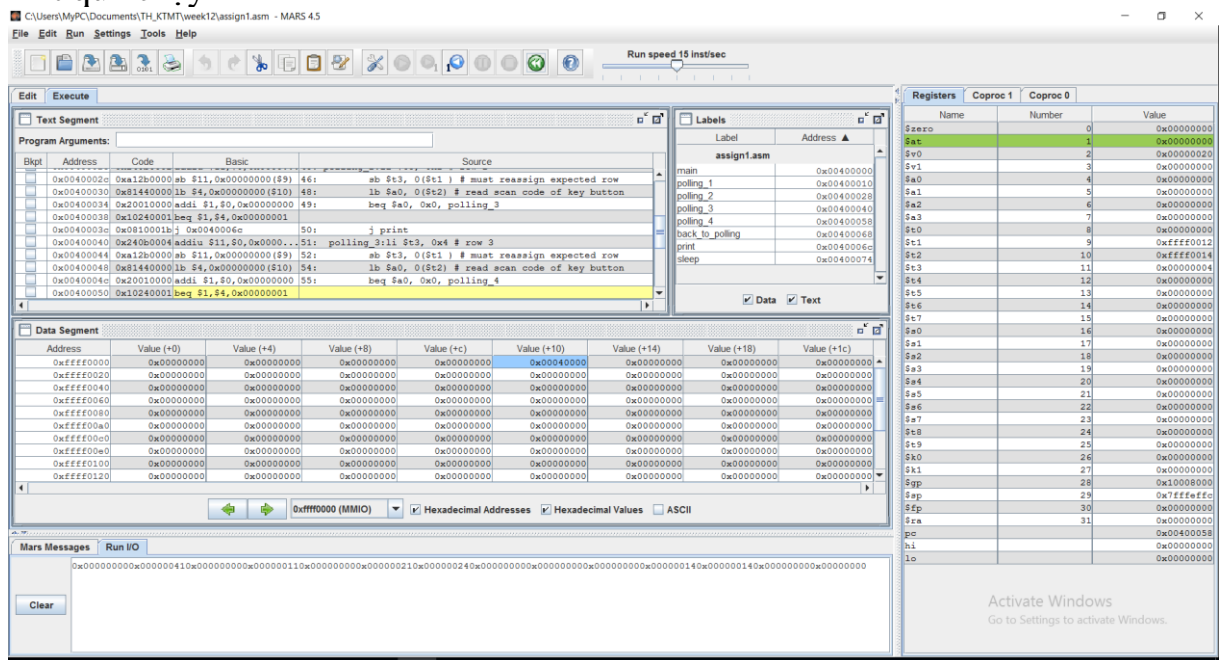
22
23 .eqv IN_ADDRESS_HEX_KEYBOARD 0xFFFF0012
24
25 # receive row and column of the key pressed, 0 if not key pressed
26 # Eg. equal 0x11, means that key button 0 pressed.
27 # Eg. equal 0x28, means that key button D pressed.
28
29 .eqv OUT_ADDRESS_HEX_KEYBOARD 0xFFFF0014
30 .text
31
32 main:    li $t1, IN_ADDRESS_HEX_KEYBOARD
33
34          li $t2, OUT_ADDRESS_HEX_KEYBOARD
35
36          #li $t3, 0x08 # check row 4 with key C, D, E, F
37
38 polling_1:li $t3, 0x1 # row 1
39          sb $t3, 0($t1) # must reassign expected row
40
41          lb $a0, 0($t2) # read scan code of key button
42
```

```

43         beq $a0, 0x0, polling_2
44         j print
45 polling_2:li $t3, 0x2 # row 2
46         sb $t3, 0($t1) # must reassign expected row
47
48         lb $a0, 0($t2) # read scan code of key button
49         beq $a0, 0x0, polling_3
50         j print
51 polling_3:li $t3, 0x4 # row 3
52         sb $t3, 0($t1) # must reassign expected row
53
54         lb $a0, 0($t2) # read scan code of key button
55         beq $a0, 0x0, polling_4
56         j print
57 polling_4:li $t3, 0x8 # row 4
58         sb $t3, 0($t1) # must reassign expected row
59
60         lb $a0, 0($t2) # read scan code of key button
61         j print
62 back_to_polling: j polling_1 # continue polling
63
64 print:
65         li $v0, 34 # print integer (hexa)
66
67         syscall
68 sleep: li $a0, 100 # sleep 100ms
69
70         li $v0, 32
71         syscall
72         j back_to_polling
73
74
75

```

- **Kết quả chạy**



- **Giải thích**

Yêu cầu: Check toàn bộ các ký tự từ 0 -> F

In ra kết quả khi nhập mã số sinh viên từ bàn phím

- 0x41 là số 2;
- 0x11 là số 0;
- 0x21 là số 1;
- 0x24 là số 9;
- 0x12 là số 4;
- 0x22 là số 5;
- 0x14 là số 8;
- 0x14 là số 8;

MSSV: 20194588 là Kết quả như ảnh trên

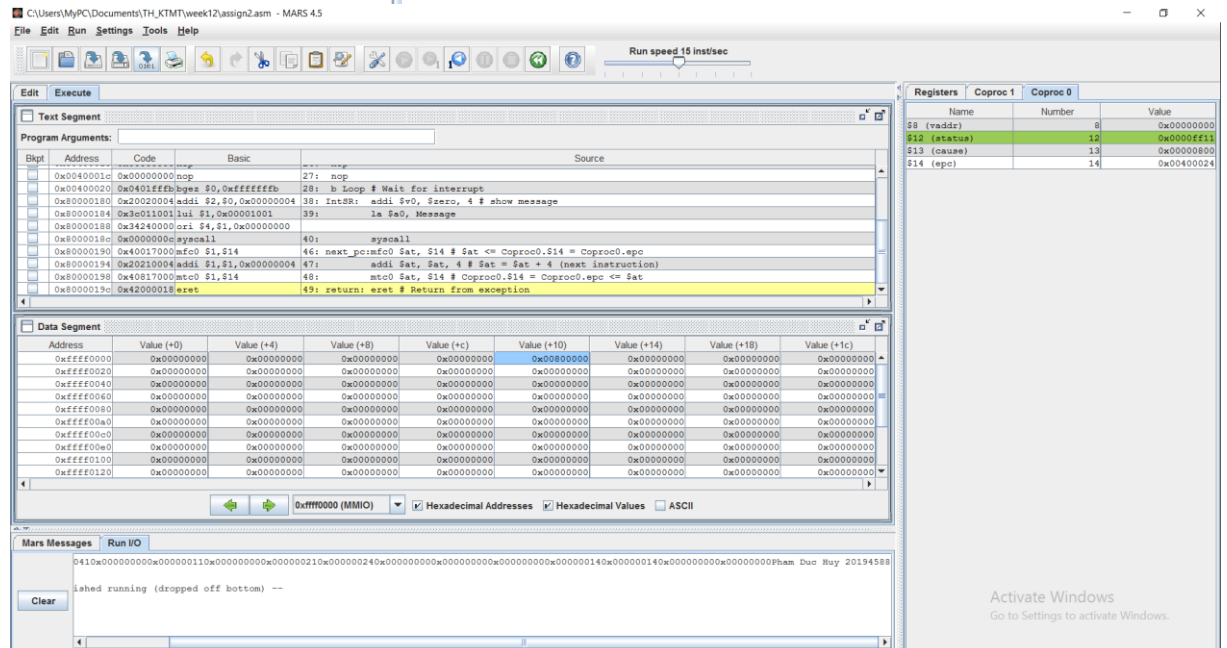
Bài 2:

o Mã nguồn

```
1 .eqv IN_ADDRESS_HEX_A_KEYBOARD 0xFFFF0012
2 .data
3
4 Message: .asciiz "Pham Duc Huy 20194588\n"
5 #~~~~~
6 # MAIN Procedure
7 #~~~~~
8
9 .text
10 main:
11 #-----
12 # Enable interrupts you expect
13 #-----
14 # Enable the interrupt of Keyboard matrix 4x4 of Digital Lab Sim
15
16 li $t1, IN_ADDRESS_HEX_A_KEYBOARD
17 li $t3, 0x80 # bit 7 of = 1 to enable interrupt
18 sb $t3, 0($t1)
19
20 #-----
21 # No-end loop, main program, to demo the effective of interrupt
22 #-----
23
24 Loop: nop
25     nop
26     nop
27     nop
28     b Loop # Wait for interrupt
29 end_main:
30 #~~~~~
31 # GENERAL INTERRUPT SERVED ROUTINE for all interrupts
32 #~~~~~
33
34 .ktext 0x80000180
35 #-----
36 # Processing
37 #-----
38 IntSR: addi $v0, $zero, 4 # show message
39         la $a0, Message
40         syscall
41 #-----
42 # Evaluate the return address of main routine
43 # epc <= epc + 4
44 #-----
45
46 next_pc:mfc0 $at, $14 # $at <= Coproc0.$14 = Coproc0.epc
47         addi $at, $at, 4 # $at = $at + 4 (next instruction)
48         mtc0 $at, $14 # Coproc0.$14 = Coproc0.epc <= $at
49 return: eret # Return from exception
50
```

o Kết quả chạy

Pham Duc Huy 20194588



- Giải thích
Khi nhấn phím bất kì từ 0 -> F thì sẽ hiện ra tên và mssv

Bài 3:

- Mã nguồn

```
1  .eqv IN_ADRESS_HEXА_KEYBOARD 0xFFFF0012
2  .eqv OUT_ADRESS_HEXА_KEYBOARD 0xFFFF0014
3  .data
4
5  Message: .asciiz "Key scan code "
6
7  #~~~~~
8  # MAIN Procedure
9  #~~~~~
10
11 .text
12 main:
13
14 #-----
15 # Enable interrupts you expectre
16 #-----
17 # Enable the interrupt of Keyboard matrix 4x4 of Digital Lab Sim
18
19 li $t1, IN_ADRESS_HEXА_KEYBOARD
20 li $t3, 0x80 # bit 7 = 1 to enable
21 sb $t3, 0($t1)
22
23 #-----
24 # Loop an print sequence numbers
25 #-----
26 xor $s0, $s0, $s0 # count = $s0 = 0
27
28 Loop: addi $s0, $s0, 1 # count = count + 1
29
30 prn_seq: addi $v0, $zero, 1
31 add $a0, $s0, $zero # print auto sequence number
32 syscall
33
34 prn_eol: addi $v0, $zero, 11
35 li $a0, '\n' # print endofline
36 syscall
37
38 sleep: addi $v0, $zero, 32
39 li $a0, 300 # sleep 300 ms
40 syscall
41 nop # WARNING: nop is mandatory here.
42 b Loop # Loop
```

```

43 end_main:
44
45 #~~~~~
46 # GENERAL INTERRUPT SERVED ROUTINE for all interrupts
47 #~~~~~
48
49 .ktext 0x80000180
50
51 #-----
52 # SAVE the current REG FILE to stack
53 #-----
54
55 IntSR: addi $sp,$sp,4 # Save $ra because we may change it later
56 sw $ra,0($sp)
57 addi $sp,$sp,4 # Save $ra because we may change it later
58 sw $at,0($sp)
59 addi $sp,$sp,4 # Save $ra because we may change it later
60 sw $v0,0($sp)
61 addi $sp,$sp,4 # Save $a0, because we may change it later
62 sw $a0,0($sp)
63 addi $sp,$sp,4 # Save $t1, because we may change it later
64 sw $t1,0($sp)
65 addi $sp,$sp,4 # Save $t3, because we may change it later
66 sw $t3,0($sp)
67 #-----
68 # Processing
69 #-----
70
71 prn_msg:addi $v0, $zero, 4
72 la $a0, Message
73 syscall
74
75 get_cod:li $t1, IN_ADDRESS_HEX_A_KEYBOARD
76
77 interrupt_1: li $t3, 0x81 # check row 1 and re-enable bit 7
78 sb $t3, 0($t1) # must reassign expected row
79 li $t1, OUT_ADDRESS_HEX_A_KEYBOARD
80 lb $a0, 0($t1)
81 beq $a0, 0x0, interrupt_2
82 j prn_cod
83
84 interrupt_2: li $t3, 0x82 # check row 2 and re-enable bit 7

```

```

85  sb $t3, 0($t1) # must reassign expected row
86  li $t1, OUT_ADDRESS_HEX_A_KEYBOARD
87  lb $a0, 0($t1)
88  beq $a0, 0x0, interrupt_3
89  j prn_cod
90
91  interrupt_3: li $t3, 0x84 # check row 3 and re-enable bit 7
92  sb $t3, 0($t1) # must reassign expected row
93  li $t1, OUT_ADDRESS_HEX_A_KEYBOARD
94  lb $a0, 0($t1)
95  beq $a0, 0x0, interrupt_4
96  j prn_cod
97
98  interrupt_4: li $t3, 0x88 # check row 4 and re-enable bit 7
99  sb $t3, 0($t1) # must reassign expected row
100 li $t1, OUT_ADDRESS_HEX_A_KEYBOARD
101 lb $a0, 0($t1)
102 beq $a0, 0x0, next_pc
103
104 prn_cod: li $v0, 34
105
106 syscall
107 li $v0, 11
108 li $a0, '\n' # print endofline
109 syscall
110
111 #-----
112 # Evaluate the return address of main routine
113 # epc <= epc + 4
114 #-----
115
116 next_pc: mfc0 $at, $14 # $at <= Coproc0.$14 = Coproc0.epc
117 addi $at, $at, 4 # $at = $at + 4 (next instruction)
118 mtc0 $at, $14 # Coproc0.$14 = Coproc0.epc <= $at
119
120 #-----
121 # RESTORE the REG FILE from STACK
122 #-----
123
124 restore: lw $t3, 0($sp) # Restore the registers from stack
125 addi $sp, $sp, -4
126 lw $t1, 0($sp) # Restore the registers from stack
127
128 addi $sp, $sp, -4
129 lw $a0, 0($sp) # Restore the registers from stack
130 addi $sp, $sp, -4
131 lw $v0, 0($sp) # Restore the registers from stack
132 addi $sp, $sp, -4
133 lw $ra, 0($sp) # Restore the registers from stack
134 addi $sp, $sp, -4
135
136 return: eret # Return from exception

```

- Kết quả chạy

