Bài 4:

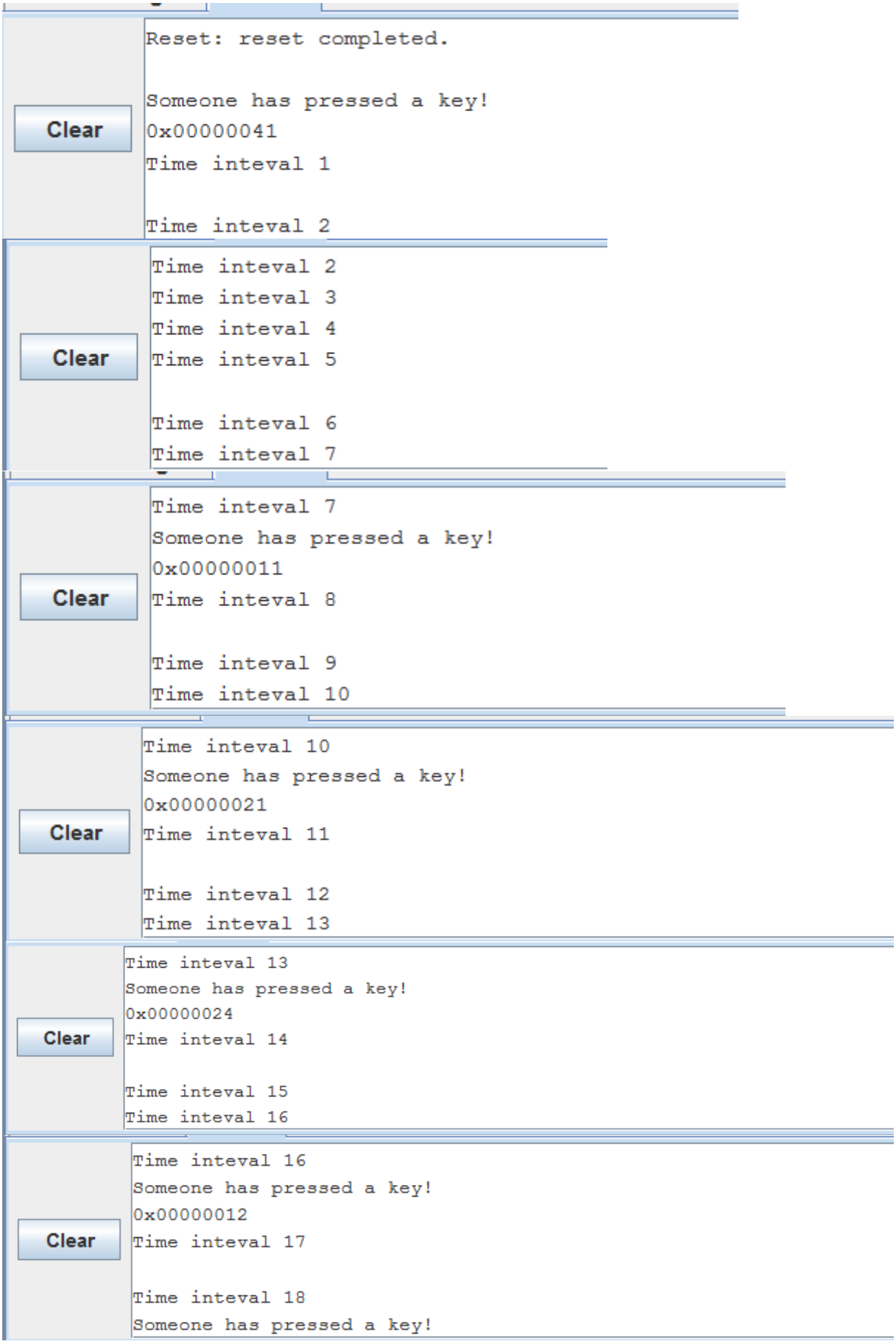
- Mã nguồn

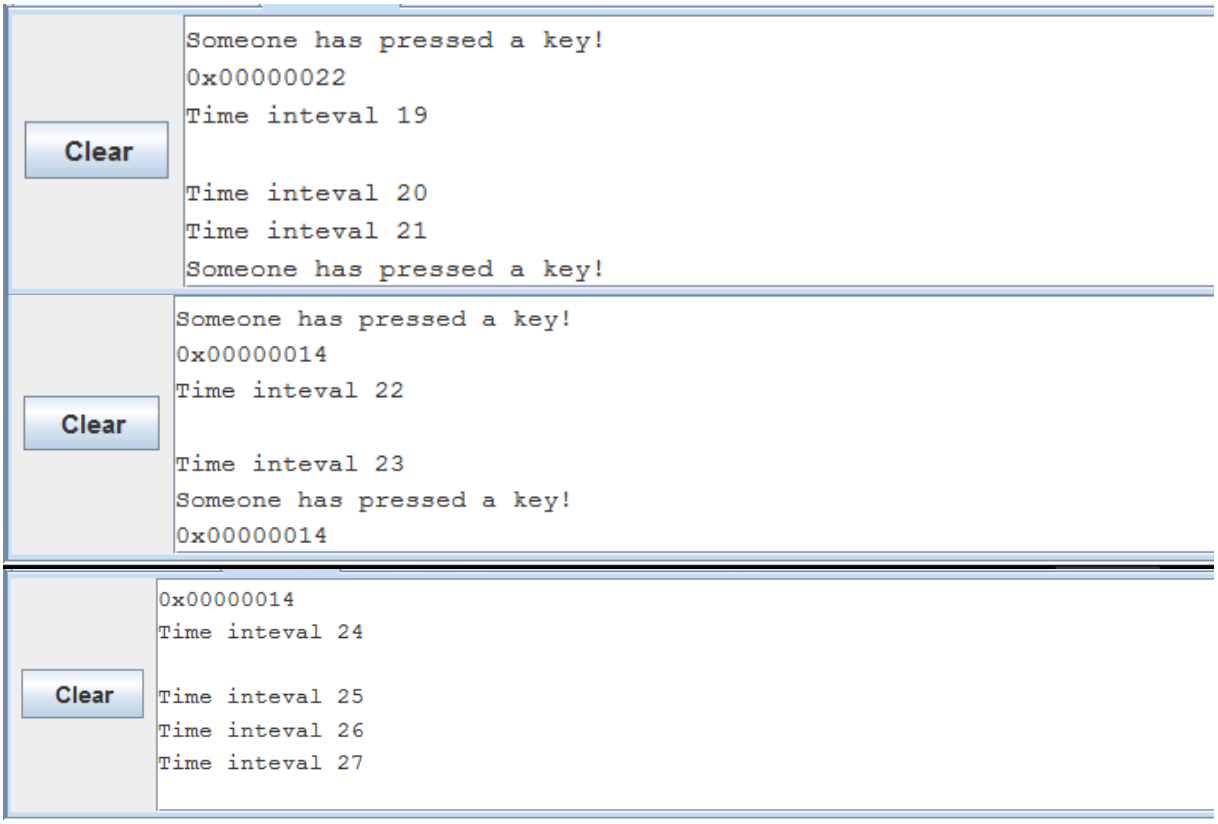
```
.eqv IN_ADRESS_HEXA_KEYBOARD 0xFFFF0012
.eqv OUT_ADRESS_HEXA_KEYBOARD 0xFFFF0014
.eqv COUNTER 0xFFFF0013 # Time Counter
.eqv MASK_CAUSE_COUNTER 0x00000400 # Bit 10: Counter interrupt
.eqv MASK_CAUSE_KEYMATRIX 0x00000800 # Bit 11: Key matrix interrupt

.data
msg_keypress: .asciiz "Someone has pressed a key!\n"
msg_counter: .asciiz "Time inteval!\n"

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# 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)

# Enable the interrupt of TimeCounter of Digital Lab Sim
 li $t1, COUNTER
 sb $t1, 0($t1)

 #-------------------------------------------------------------
 # Loop an print sequence numbers
 #-------------------------------------------------------------

Loop: nop
 nop
 nop
sleep: addi $v0,$zero,32 # BUG: must sleep to wait for Time Counter
 li $a0,200 # sleep 300 ms
 syscall
 nop # WARNING: nop is mandatory here.
 b Loop
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   and $at, $t1,$t2
65   beq $at,$t2, Counter_Intr
66
67  IsKeyMa:li $t2, MASK_CAUSE_KEYMATRIX # if Cause value confirm Key..
68   and $at, $t1,$t2
69   beq $at,$t2, Keymatrix_Intr
70
71  others: j end_process # other cases
72
73  Keymatrix_Intr: li $v0, 4 # Processing Key Matrix Interrupt
74
75   la $a0, msg_keypress
76   syscall
77
78  get_cod: li $t1, IN_ADRESS_HEXA_KEYBOARD
79   li $t2, OUT_ADRESS_HEXA_KEYBOARD
80  start_interrupt_1:
81          li $t3, 0x81 # check row 1 with key 0, 1, 2, 4
82          sb $t3, 0($t1) # must reassign expected row
83          jal interrupt
84
```

```
start_interrupt_2:
        li $t3, 0x82 # check row 2 with key 4, 5, 6, 7
        sb $t3, 0($t1) # must reassign expected row
        jal interrupt

start_interrupt_3:
        li $t3, 0x84 # check row 3 with key 8, 9, A, B
        sb $t3, 0($t1) # must reassign expected row
        jal interrupt

start_interrupt_4:
        li $t3, 0x88 # check row 4 with key C, D, E, F
        sb $t3, 0($t1) # must reassign expected row
        jal end_process

check_after_interrupt_4:
        beq $a0, 0x0, prn_cod
        j next_pc

interrupt:
        lb $a0, 0($t2) # read scan code of key button
        bne $a0, 0x0, prn_cod
        jr $ra
prn_cod:li $v0,34
 syscall
 li $v0,11
 li $a0,'\n' # print endofline
 syscall

 j end_process

Counter_Intr: li $v0, 4 # Processing Counter Interrupt

 la $a0, msg_counter
 syscall

 j end_process

end_process:
 mtc0 $zero, $13 # Must clear cause reg
en_int:
#-----------------------------------------------------------
# Re-enable interrupt
#-----------------------------------------------------------

 li $t1, COUNTER
 sb $t1, 0($t1)

 #-----------------------------------------------------------
 # Evaluate the return address of main routine
 # epc <= epc + 4
 #-----------------------------------------------------------

next_pc:mfc0 $at, $14 # $at <= Coproc0.$14 = Coproc0.epc
 addi $at, $at, 4 # $at = $at + 4 (next instruction)
 mtc0 $at, $14 # Coproc0.$14 = Coproc0.epc <= $at
return: eret # Return from exception
```

- Màn hình chạy


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



- 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: 1
  - 0x14: 5

⇨ Chương trình chạy đúng và in ra MSSV 20194515
Chương trình in ra thời gian => Sau 27 đơn vị thời gian thì in xong MSSV

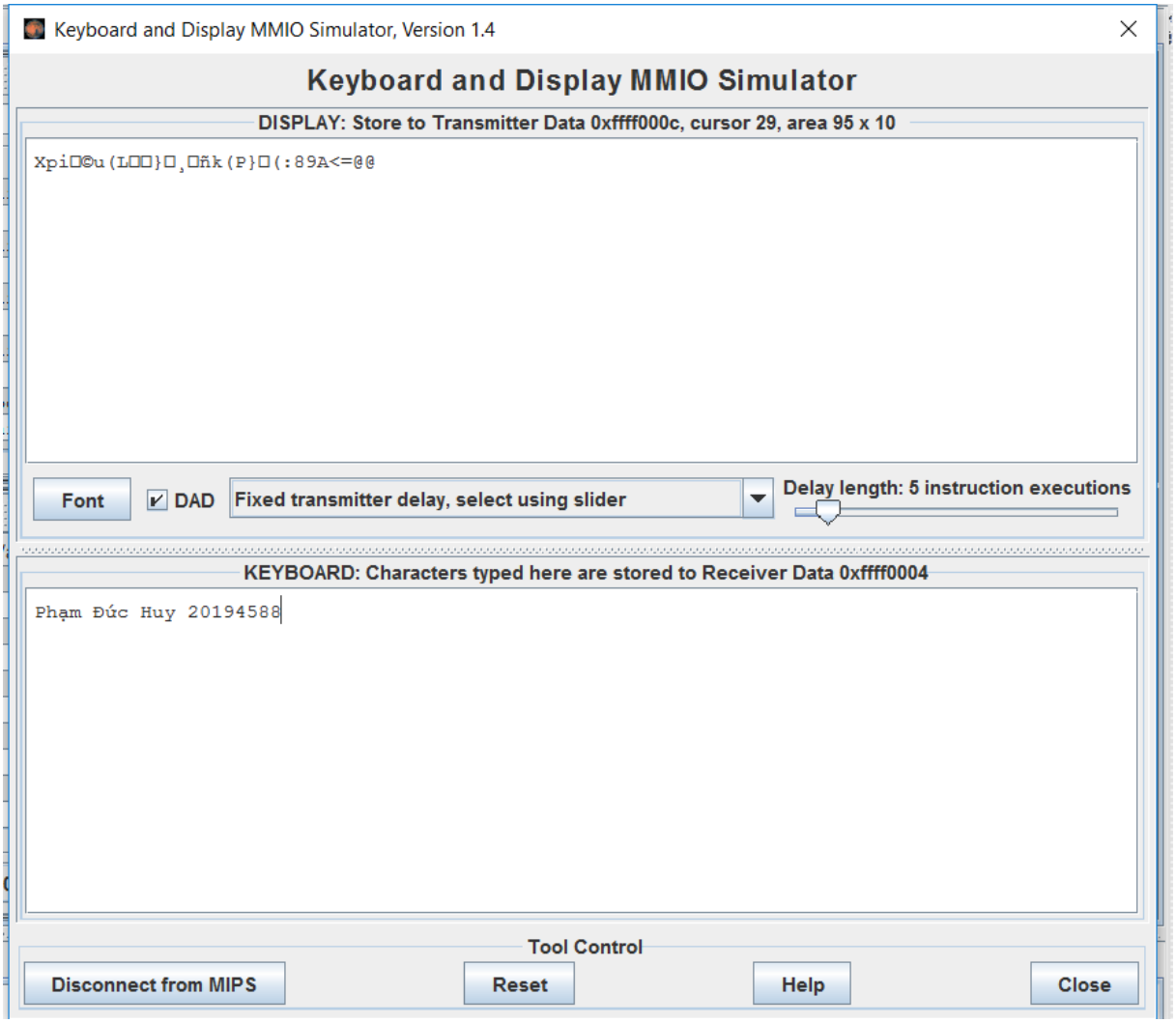
- Chương trình cho phép ngắt đồng thời bằng 2 cách: từ bàn phím Lab Sim và bộ đếm thời gian của Lab Sim
- Tại Coproc0, thanh ghi 13 lưu giá trị để phân biệt kiểu ngắt: 0x400 -> ngắt timer, 0x800 -> ngắt từ bàn phím
- Khi kết nối với Lab Sim và ấn phím bất kì thì chương trình hiển thị message và địa chỉ tương ứng của số.
- Time inteval! 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:

```
.eqv KEY_CODE 0xFFFF0004 # ASCII code from keyboard, 1 byte
.eqv KEY_READY 0xFFFF0000 # =1 if has a new keycode ?
 # Auto clear after lw
.eqv DISPLAY_CODE 0xFFFF000C # ASCII code to show, 1 byte

.eqv DISPLAY_READY 0xFFFF0008 # =1 if the display has already to do

 # Auto clear after sw

.eqv MASK_CAUSE_KEYBOARD 0x0000034 # Keyboard Cause

.text

 li $k0, KEY_CODE
 li $k1, KEY_READY

 li $s0, DISPLAY_CODE
 li $s1, DISPLAY_READY

loop: nop
WaitForKey: lw $t1, 0($k1) # $t1 = [$k1] = KEY_READY
 beq $t1, $zero, WaitForKey # if $t1 == 0 then Polling
MakeIntR: teqi $t1, 1 # if $t0 = 1 then raise an Interrupt
 j loop

#---------------------------------------------------------------
# Interrupt subroutine
#---------------------------------------------------------------

.ktext 0x80000180

get_caus: mfc0 $t1, $13 # $t1 = Coproc0.cause

IsCount: li $t2, MASK_CAUSE_KEYBOARD # if Cause value confirm Keyboard..

 and $at, $t1,$t2
 beq $at,$t2, Counter_Keyboard
 j end_process

Counter_Keyboard:

ReadKey: lw $t0, 0($k0) # $t0 = [$k0] = KEY_CODE
WaitForDis: lw $t2, 0($s1) # $t2 = [$s1] = DISPLAY_READY
 beq $t2, $zero, WaitForDis # if $t2 == 0 then Polling
Encrypt: addi $t0, $t0, 8 # change input key add 8 because MSSV 20194508
ShowKey: sw $t0, 0($s0) # show key
 nop
end_process:

next_pc: mfc0 $at, $14 # $at <= Coproc0.$14 = Coproc0.epc
 addi $at, $at, 4 # $at = $at + 4 (next instruction)
 mtc0 $at, $14 # Coproc0.$14 = Coproc0.epc <= $at
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