BÁO CÁO THỰC HÀNH

THỰC HÀNH KIẾN TRÚC MÁY TÍNH

Tuần 11:

Họ và tên	Phạm Quốc Cường
Mã số sinh viên	20225604

Assignment 4:

Code:

```
.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! count: "
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~
# 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 $k0, 0
        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:

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~
# GENERAL INTERRUPT SERVED ROUTINE for all interrupts
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~
.ktext 0x80000180
IntSR:
#----------------------------------------------------
# Temporary disable interrupt
#----------------------------------------------------
dis_int:li $t1, COUNTER # BUG: must disable with Time Counter
        sb $zero, 0($t1)
# no need to disable keyboard matrix interrupt
```

```
#---------------------------------------------------
# Processing
#---------------------------------------------------
get_caus:mfc0 $t1, $13 # $t1 = Coproc0.cause
IsCount:li $t2, MASK_CAUSE_COUNTER   # if Cause value confirm Counter..
        and $at, $t1,$t2
        beq $at,$t2, Counter_Intr
IsKeyMa:li $t2, MASK_CAUSE_KEYMATRIX # if Cause value confirm Key..
        and $at, $t1,$t2
        beq $at,$t2, Keymatrix_Intr
others: j end_process # other cases
Keymatrix_Intr: li $v0, 4 # Processing Key Matrix Interrupt
        la $a0, msg_keypress
        syscall
get_cod: li $t1, IN_ADRESS_HEXA_KEYBOARD
 li $t2, OUT_ADRESS_HEXA_KEYBOARD
start_interrupt_1:
li $t3, 0x81 # check row 1 with key 0, 1, 2, 4
sb $t3, 0($t1) # must reassign expected row
jal interrupt

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 interrupt
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
        addi $k0, $k0, 1

        li $v0, 1
        add $a0, $0, $k0
        syscall

        li $v0,11
        li $a0,'\n' # print endofline
        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
```

Result:

```
Someone has pressed a key!
0x00000021
Time inteval! count: 1

Someone has pressed a key!
0x00000042
Someone has pressed a key!
0x00000044
Someone has pressed a key!
0x00000048
Time inteval! count: 2
```

Explain:

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 trên nhập số lần lượt từ 4 hàng.

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.

Assignment 5:

Code:

```
.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..
```

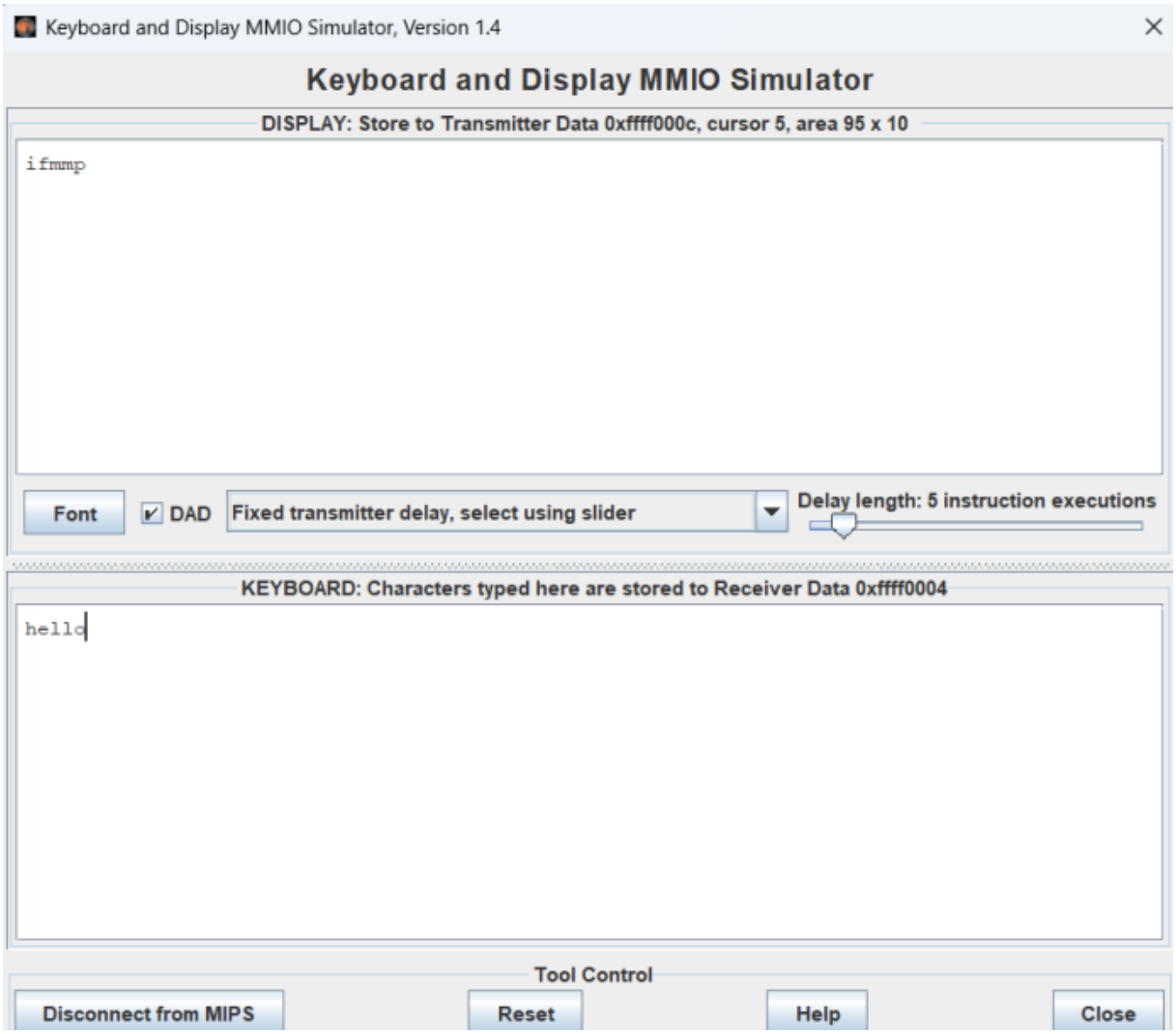
```
        and $at, $t1,$t2
        beq $at,$t2, Counter_Keyboard
        j end_process

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, 1 # change input key
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
```

Result:



Explain:

Sử dụng teq hoặc teq 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 chạy vòng lặp vô tận và chờ cờ sẵn sàng của bàn phím (KEY_READY), khi có phím nhấn KEY_READY = 1, lệnh ngắt teqi được kích hoạt

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 bằng cách cộng mã ASCII thêm 1 (Encrypt) và in ra màn hình.