

ASSIGNMENT LAB 11

-Họ tên: Hồ Tuấn Huy

-MSSV: 20225856

Lab 4:

Assignment 4

Create a new project, type in, and build the program of Home Assignment 4.

Upgrade the source code so that it could detect all 16 key buttons, from 0 to F.

Code:

```
.eqv IN_ADRESS_HEX_A_KEYBOARD 0xFFFF0012
.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 interval!\n"
#~~~~~
# MAIN Procedure
#~~~~~
.text
main:
#-----
# Enable interrupts you expect
#-----
# Enable the interrupt of Keyboard matrix 4x4 of Digital Lab Sim
li $t1, IN_ADRESS_HEX_A_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 and 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
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

```

Kết quả:

Khi nhấn lần lượt các nút 2, 5, a, e trên Digital Lab Sim:

```
Time interval!
Time interval!
Someone has pressed a key!
Time interval!
Time interval!
Time interval!
Time interval!
Time interval!
Someone has pressed a key!
Time interval!
Time interval!
Time interval!
Time interval!
```

```
Time interval!
Someone has pressed a key!
Time interval!
Time interval!
Time interval!
Time interval!
Time interval!
Time interval!
Time interval!
Time interval!
Time interval!
Someone has pressed a key!
Time interval!
```

Giải thích: Chương trình minh họa cách tiếp cận cơ bản dựa trên ngắt để xử lý ngắt từ bộ đếm thời gian và bàn phím:

- Trước hết ta thiết lập thủ tục chính để thiết lập ngắt cho bàn phím ma trận và bộ đếm thời gian.
- Thiết lập vòng lặp để chương trình vào vòng lặp vô hạn, chờ ngắt xảy ra.
- Khi có ngắt thì kiểm tra nguyên nhân và hiển thị thông báo tương ứng:
 - + Nếu nguyên nhân từ bàn phím thì in ra thông báo "Someone has pressed a key!".
 - + Nếu nguyên nhân từ bộ đếm thì in ra thông báo "Time interval!".
- Sau khi xử lý ngắt thì bật lại ngắt và trở về chương trình chính.

Lab 5:

Assignment 5

Create a new project, type in, and build the program of Home Assignment 5.

Code:

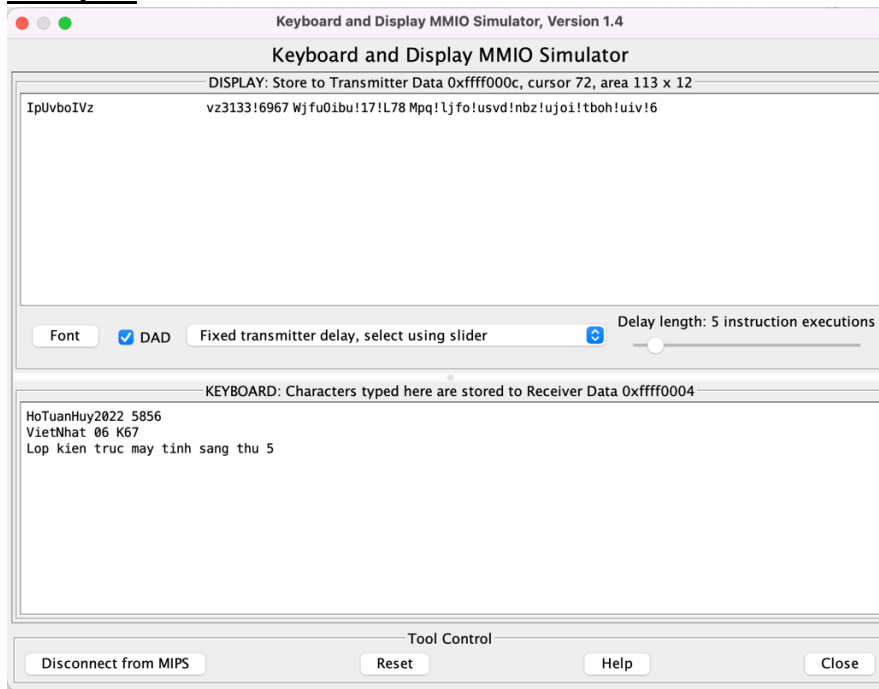
```
.eqv KEY_CODE 0xFFFF0004 # ASCII code from keyboard, 1 byte
.eqv KEY_READY 0xFFFF0000 # =1 if has a new keycode ?
# Auto clear after lw
.eqv DISPLAY_CODE 0xFFFF000C # ASCII code to show, 1 byte
.eqv DISPLAY_READY 0xFFFF0008 # =1 if the display has already to do
# Auto clear after sw
.eqv MASK_CAUSE_KEYBOARD 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, WaitForKey # 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

```

Kết quả:



Giải thích: Chương trình thực hiện việc xử lý ngắt từ bàn phím, mã hoá ký tự đầu vào và hiển thị ký tự lên màn hình:

- Trước hết thiết lập các thanh ghi.
- Vòng lặp chính:
 - + Chờ mã phím mới (KEY_READY) bằng cách kiểm tra liên tục (Polling).
 - + Khi có mã phím mới (KEY_READY=1) thì kích hoạt ngắt.
- Thủ tục để xử lý ngắt: được đặt tại địa chỉ 0x80000180.
 - + Đọc thanh ghi nguyên nhân ngắt để xác định nguyên nhân.
 - + Nếu ngắt từ bàn phím (MASK_CAUSE_KEYBOARD) thì đọc mã phím từ KEY_CODE.
 - + Chờ màn hình sẵn sàng (DISPLAY_READY), mã hoá ký tự đầu vào (thêm 1 vào mã ASCII) rồi hiển thị ký tự lên màn hình (DISPLAY_CODE).
 - + Điều chỉnh địa chỉ trở về và trở lại chương trình chính.