**Thực hành kiến trúc máy tính tuần 36**

**Full name: Lê Quang Khải**

**Student ID: 20225638**

Assignment 1

.eqv SEVENSEG\_LEFT 0xFFFF0011 # Dia chi cua den led 7 doan trai.

 # Bit 0 = doan a;

 # Bit 1 = doan b; ...

 # Bit 7 = dau .

.eqv SEVENSEG\_RIGHT 0xFFFF0010 # Dia chi cua den led 7 doan phai

.text

main:

    li $a0, 7 # set value for segments

    jal SHOW\_7SEG\_LEFT # show

    nop

    li $a0, 109 # set value for segments

    jal SHOW\_7SEG\_RIGHT # show

    nop

exit:   li $v0, 10

    syscall

endmain:

#---------------------------------------------------------------

# Function SHOW\_7SEG\_LEFT : turn on/off the 7seg

# param[in] $a0 value to shown

# remark $t0 changed

#---------------------------------------------------------------

SHOW\_7SEG\_LEFT:

    li $t0, SEVENSEG\_LEFT # assign port's address

    sb $a0, 0($t0) # assign new value

    nop

    jr $ra

    nop

#---------------------------------------------------------------

# Function SHOW\_7SEG\_RIGHT : turn on/off the 7seg

# param[in] $a0 value to shown

# remark $t0 changed

#---------------------------------------------------------------

SHOW\_7SEG\_RIGHT:

    li $t0, SEVENSEG\_RIGHT # assign port's address

    sb $a0, 0($t0) # assign new value

    nop

    jr $ra

    nop

**Hiển thị các số khác nhau:**

- Hiển thị 75: A screenshot of a computer program

Description automatically generated

- Hiển thị 19:

A screenshot of a computer

Description automatically generated

- Hiển thị 54:

A screenshot of a computer

Description automatically generated

Assignment 2:

.eqv MONITOR\_SCREEN 0x10010000 #Dia chi bat dau cua bo nho man hinh

.eqv RED 0x00FF0000 #Cac gia tri mau thuong su dung

.eqv GREEN 0x0000FF00

.eqv BLUE 0x000000FF

.eqv WHITE 0x00FFFFFF

.eqv YELLOW 0x00FFFF00

.eqv MAGENTA 0x00FF00FF

.eqv CYAN 0x0000FFFF

.eqv PURPLE 0x00800080

.text

    li $k0, MONITOR\_SCREEN #Nap dia chi bat dau cua man hinh

    li $t1, 0 # counter

    li $t2, 0

    li $t3, 0

loop\_red:

    li $t2, 1

        beq $t1, 8, next\_line\_red

    li $t0, RED

    sw $t0, 0($k0)

    addi $k0, $k0, 4

    addi $t1, $t1, 1

    j loop\_red

next\_line\_red:

    li $t1, 0

loop\_green:

    li $t2, 1

        beq $t1, 8, next\_line\_green

    li $t0, GREEN

    sw $t0, 0($k0)

    addi $k0, $k0, 4

    addi $t1, $t1, 1

    j loop\_green

next\_line\_green:

       li $t1, 0

loop\_blue:

    li $t2, 1

        beq $t1, 8, next\_line\_blue

    li $t0, BLUE

    sw $t0, 0($k0)

    addi $k0, $k0, 4

    addi $t1, $t1, 1

    j loop\_blue

next\_line\_blue:

       li $t1, 0

loop\_white:

    li $t2, 1

        beq $t1, 8, next\_line\_white

    li $t0, WHITE

    sw $t0, 0($k0)

    addi $k0, $k0, 4

    addi $t1, $t1, 1

    j loop\_white

next\_line\_white:

       li $t1, 0

loop\_yellow:

    li $t2, 1

        beq $t1, 8, next\_line\_yellow

    li $t0, YELLOW

    sw $t0, 0($k0)

    addi $k0, $k0, 4

    addi $t1, $t1, 1

    j loop\_yellow

next\_line\_yellow:

       li $t1, 0

loop\_magenta:

    li $t2, 1

        beq $t1, 8, next\_line\_magenta

    li $t0, MAGENTA

    sw $t0, 0($k0)

    addi $k0, $k0, 4

    addi $t1, $t1, 1

    j loop\_magenta

next\_line\_magenta:

       li $t1, 0

loop\_cyan:

    li $t2, 1

        beq $t1, 8, next\_line\_cyan

    li $t0, CYAN

    sw $t0, 0($k0)

    addi $k0, $k0, 4

    addi $t1, $t1, 1

    j loop\_cyan

next\_line\_cyan:

       li $t1, 0

loop\_purple:

    li $t2, 1

        beq $t1, 8, next\_line\_purple

    li $t0, PURPLE

    sw $t0, 0($k0)

    addi $k0, $k0, 4

    addi $t1, $t1, 1

    j loop\_purple

next\_line\_purple:

       li $t1, 0

**Kết quả:**

**A screenshot of a computer display

Description automatically generated**

Assignment 3:

.eqv HEADING 0xffff8010 # Integer: An angle between 0 and 359

.eqv MOVING 0xffff8050 # Boolean: whether or not to move

.eqv LEAVETRACK 0xffff8020 # Boolean (0 or non-0): # whether or not to leave a track

.eqv WHEREX 0xffff8030 # Integer: Current x-location of MarsBot

.eqv WHEREY 0xffff8040 # Integer: Current y-location of MarsBot

.text

main:

    addi $a0, $zero, 120 # Marsbot rotates 90\* and start running

    jal ROTATE

    jal GO

sleep1:

    addi $v0,$zero, 32 # Keep running by sleeping in 5000 ms

    li $a0, 5000

    syscall

    jal TRACK # and draw new track line

go\_150:

    addi $a0, $zero, 150 # Marsbot rotates 150\*

    jal ROTATE

sleep2:

    addi $v0,$zero, 32 # Keep running by sleeping in 3000 ms

    li $a0, 5000

    syscall

    jal UNTRACK # keep old track

    jal TRACK # and draw new track line

goLEFT:

    addi $a0, $zero, 270 # Marsbot rotates 270\* đi ngang sang trái

    jal ROTATE

sleep3:

    addi $v0,$zero, 32 # Keep running by sleeping in 5000 ms

    li $a0, 5000

    syscall

    jal UNTRACK # keep old track

    jal TRACK # and draw new track line

goUP:

    addi $a0, $zero, 30 # Marsbot rotates 30\*

    jal ROTATE

sleep4:

    addi $v0,$zero,32 # Keep running by sleeping in 5000 ms

    li $a0,5000

    syscall

    jal UNTRACK

    jal STOP

end\_main:

    li $v0, 10

    syscall

GO:

    li $at, MOVING # change MOVING port

    addi $k0, $zero, 1 # to logic 1,

    sb $k0, 0($at) # to start running

    jr $ra

STOP:

    li $at, MOVING # change MOVING port to 0

    sb $zero, 0($at) # to stop

    jr $ra

TRACK:

    li $at, LEAVETRACK # change LEAVETRACK port

    addi $k0, $zero, 1 # to logic 1

    sb $k0, 0($at) # to start tracking

    jr $ra

UNTRACK:

    li $at, LEAVETRACK # change LEAVETRACK port to 0

    sb $zero, 0($at) # to stop drawing tail

    jr $ra

ROTATE:

    li $at, HEADING # change HEADING port

    sw $a0, 0($at) # to rotate robot

    jr $ra

**Kết quả:**

**A screenshot of a computer

Description automatically generated**

Assignment 4:

.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

.text

    li $k0, KEY\_CODE

    li $k1, KEY\_READY

    li $s0, DISPLAY\_CODE

    li $s1, DISPLAY\_READY

    li $t3, 0

loop:

    nop

WaitForKey:

    lw $t1, 0($k1) # $t1 = [$k1] = KEY\_READY

    beq $t1, $zero, WaitForKey # if $t1 == 0 then Polling

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:

    jal check\_exit

    jal check\_uppercase

ShowKey:

    sw $t0, 0($s0) # show key

    bge $t3, 4, end\_main # khi $t3 = 4 là đã chạy hết chữ exit

    nop

    j loop

end\_main:

    li $v0, 10

    syscall

check\_uppercase:

    bgt $t0, 'Z', check\_lowercase

    blt $t0, 'A', check\_number

    addi $t0, $t0, 32

    jr $ra

check\_lowercase:

    bgt $t0, 'z', default

    blt $t0, 'a', default

    subi $t0, $t0, 32

    jr $ra

check\_number:

    blt $t0, '0', default

    bgt $t0, '9', default

    jr $ra

default:

    addi $t0, $zero, 32 # $t0 = \*

    jr $ra

check\_exit:

    beq $t0, 'e', check\_e

    beq $t0, 'x', check\_x

    beq $t0, 'i', check\_i

    beq $t0, 't', check\_t

    li $t3, 0

    jr $ra

check\_e:

    bne $t3, 0, return

    addi $t3, $t3, 1

    jr $ra

check\_x:

    bne $t3, 1, return

    addi $t3, $t3, 1

    jr $ra

check\_i:

    bne $t3, 2, return

    addi $t3, $t3, 1

    jr $ra

check\_t:

    bne $t3, 3, return

    addi $t3, $t3, 1

    jr $ra

return:

    li $t3, 0

    jr $ra

**Kết quả:**

**A screenshot of a computer

Description automatically generated**