

BÁO CÁO THỰC HÀNH KIẾN TRÚC MÁY TÍNH – TUẦN 10

Họ và tên: Nguyễn Mạnh Tùng

MSSV: 20225682

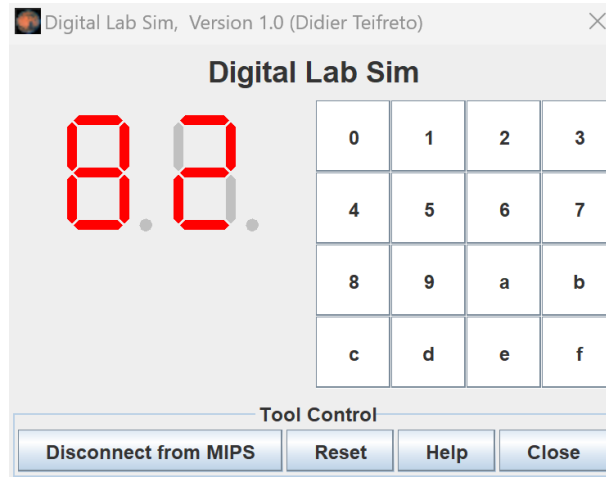
Assignment 1:

- Code:

+ Sửa lại phần set value cho hai thanh led trái và led phải để hiện thị hai số cuối của MSSV (20225682 -> 82)

```
.eqv SEVENSEG_LEFT 0xFFFF0010 # Địa chỉ led trái
.eqv SEVENSEG_RIGHT 0xFFFF0011 # Địa chỉ led phải
.text
main:
    li $a0, 0x5B                # set value for segments
    jal SHOW_7SEG_LEFT          # show
    li $a0, 0x7                 # set value for segments
    jal SHOW_7SEG_RIGHT         # show
    exit: li $v0, 10
    syscall
endmain:
SHOW_7SEG_LEFT: li $t0, SEVENSEG_LEFT # assign port's address
sb $a0, 0($t0)                  # assign new value
jr $ra
SHOW_7SEG_RIGHT: li $t0, SEVENSEG_RIGHT # assign port's address
sb $a0, 0($t0)                  # assign new value
jr $ra
```

- Kết quả:



→ Kết quả đúng với lý thuyết

Assignment 2:

- Code:

```
.eqv SEVENSEG_LEFT 0xFFFF0010      #Dia chi led trai
.eqv SEVENSEG_RIGHT 0xFFFF0011     #Dia chi led phai

.data

msg_1: .ascii "Nhap vao mot so nguyen: "

.text

main:

    li $v0, 4
    la $a0, msg_1
    syscall

    li $v0, 5
    syscall

    move $s0, $v0
```

li \$t1, 100

div \$s0, \$t1

mfhi \$t2

li \$t3, 10

div \$t2, \$t3

mfhi \$s1 #chu so hang don vi

mflo \$s2 #chu so hang chuc

beq \$s1, 0, case0

beq \$s1, 1, case1

beq \$s1, 2, case2

beq \$s1, 3, case3

beq \$s1, 4, case4

beq \$s1, 5, case5

beq \$s1, 6, case6

beq \$s1, 7, case7

beq \$s1, 8, case8

beq \$s1, 9, case9

case0:

li \$s3, 0x3F

j continue1

case1:

li \$s3, 0x06

j continue1

case2:

li \$s3, 0x5B

j continue1

case3:

li \$s3, 0x4F

j continue1

case4:

li \$s3, 0x66

j continue1

case5:

li \$s3, 0x6D

j continue1

case6:

li \$s3, 0x7D

j continue1

case7:

li \$s3, 0x07

j continue1

case8:

li \$s3, 0x7F

j continue1

case9:

li \$s3, 0x6F

j continue1

continue1:

beq \$s2, 0, case_0

beq \$s2, 1, case_1

beq \$s2, 2, case_2

beq \$s2, 3, case_3

beq \$s2, 4, case_4

beq \$s2, 5, case_5

beq \$s2, 6, case_6

beq \$s2, 7, case_7

beq \$s2, 8, case_8

beq \$s2, 9, case_9

case_0:

li \$s4, 0x3F

j continue2

case_1:

li \$s4, 0x06

j continue2

case_2:

li \$s4, 0x5B

j continue2

case_3:

li \$s4, 0x4F

j continue2

case_4:

li \$s4, 0x66

j continue2

case_5:

li \$s4, 0x6D

j continue2

case_6:

li \$s4, 0x7D

j continue2

case_7:

li \$s4, 0x07

j continue2

case_8:

li \$s4, 0x7F

j continue2

case_9:

li \$s4, 0x6F

j continue2

continue2:

move \$a0, \$s3 # set value for segments

jal SHOW_7SEG_LEFT # show

move \$a0, \$s4 # set value for segments

jal SHOW_7SEG_RIGHT # show

exit:

li \$v0, 10

syscall

endmain:

SHOW_7SEG_LEFT:

li \$t0, SEVENSEG_LEFT # assign port's address

sb \$a0, 0(\$t0) # assign new value

jr \$ra

SHOW_7SEG_RIGHT:

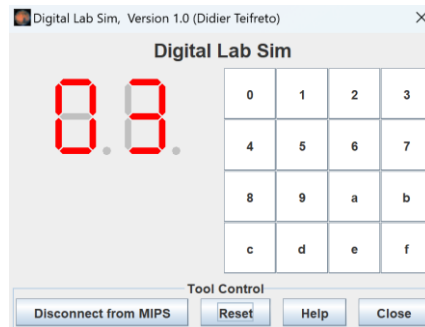
li \$t0, SEVENSEG_RIGHT # assign port's address

sb \$a0, 0(\$t0) # assign new value

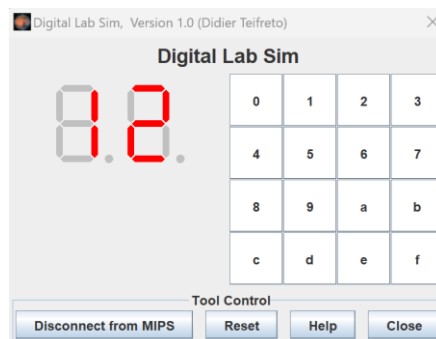
jr \$ra

- Kết quả:

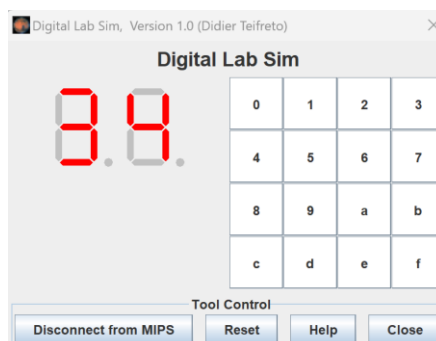
+ Nhập 3:



+ Nhập 12:



+ Nhập 1234:



➔ Kết quả đúng với lí thuyết.

Assignment 3:

- Code:

```
.eqv SEVENSEG_LEFT 0xFFFF0010      #Dia chi led trai
.eqv SEVENSEG_RIGHT 0xFFFF0011     #Dia chi led phai

.data

msg_1: .asciiz "Nhap vao mot ki tu: "

.text

main:

    li $v0, 4
    la $a0, msg_1
    syscall

    li $v0, 12
    syscall

    move $s0, $v0      $luu ma ascii vao $s0


    li $t1, 100
    div $s0, $t1
    mfhi $t2


    li $t3, 10
    div $t2, $t3

    mfhi $s1      #chu so hang don vi
    mflo $s2      #chu so hang chuc


    beq $s1, 0, case0
    beq $s1, 1, case1
```


beq \$s1, 2, case2

beq \$s1, 3, case3

beq \$s1, 4, case4

beq \$s1, 5, case5

beq \$s1, 6, case6

beq \$s1, 7, case7

beq \$s1, 8, case8

beq \$s1, 9, case9

case0:

li \$s3, 0x3F

j continue1

case1:

li \$s3, 0x06

j continue1

case2:

li \$s3, 0x5B

j continue1

case3:

li \$s3, 0x4F

j continue1

case4:

li \$s3, 0x66

j continue1

case5:

li \$s3, 0x6D

j continue1

case6:

li \$s3, 0x7D

j continue1

case7:

li \$s3, 0x07

j continue1

case8:

li \$s3, 0x7F

j continue1

case9:

li \$s3, 0x6F

j continue1

continue1:

beq \$s2, 0, case_0

beq \$s2, 1, case_1

beq \$s2, 2, case_2

beq \$s2, 3, case_3

beq \$s2, 4, case_4

beq \$s2, 5, case_5

beq \$s2, 6, case_6

beq \$s2, 7, case_7

beq \$s2, 8, case_8

beq \$s2, 9, case_9

case_0:

li \$s4, 0x3F

j continue2

case_1:

li \$s4, 0x06

j continue2

case_2:

li \$s4, 0x5B

j continue2

case_3:

li \$s4, 0x4F

j continue2

case_4:

li \$s4, 0x66

j continue2

case_5:

li \$s4, 0x6D

j continue2

case_6:

li \$s4, 0x7D

j continue2

case_7:

li \$s4, 0x07

j continue2

case_8:

li \$s4, 0x7F

j continue2

case_9:

li \$s4, 0x6F

j continue2

continue2:

move \$a0, \$s3 # set value for segments

jal SHOW_7SEG_LEFT # show

move \$a0, \$s4 # set value for segments

jal SHOW_7SEG_RIGHT # show

exit:

li \$v0, 10

syscall

endmain:

SHOW_7SEG_LEFT:

li \$t0, SEVENSEG_LEFT # assign port's address

sb \$a0, 0(\$t0) # assign new value

jr \$ra

SHOW_7SEG_RIGHT:

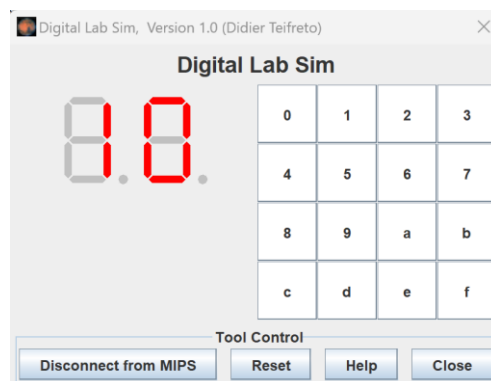
li \$t0, SEVENSEG_RIGHT # assign port's address

sb \$a0, 0(\$t0) # assign new value

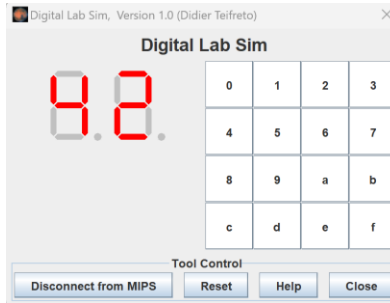
jr \$ra

- Kết quả:

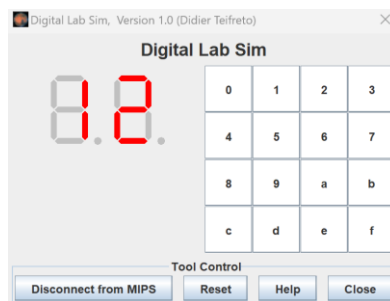
+ Nhập: enter



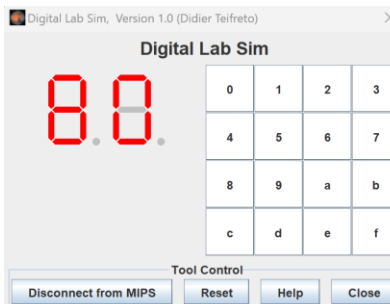
+ Nhập kí tự đặc biệt: *



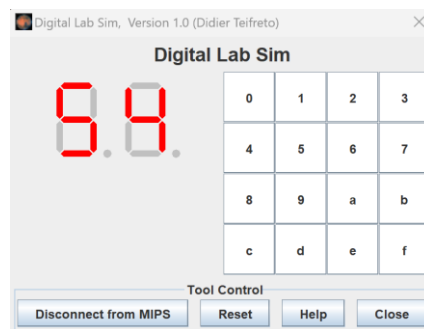
+ Nhập chữ thường: p



+ Nhập chữ hoa: P



+ Nhập số: 6



→ Kết quả đúng với lí thuyết.

Assignment 4:

- Code:

```
.eqv MONITOR_SCREEN 0x10010000

.eqv BLUE 0x000000FF

.eqv WHITE 0x00FFFFFF

.text

main:

    li $k0, MONITOR_SCREEN

    li $s0, 0        #i=0

    li $s1, 0        #j=0

for_row:

    li $s1, 0        #reset j=0

for_column:

    # vi tri 0

    mul $t0, $s0, 8    # chi so hang

    add $t0, $t0, $s1    # chi so hang + cot

    sll $t0, $t0, 2    # dia chi byte

    # to mau

    beq $t0, $zero, set_white # o dau tien = white

    andi $t1, $s0, 1    # hang chan or le

    andi $t2, $s1, 1    # cot chan or le

    bne $t1, $t2, set_blue # hang # cot -> xanh

set_white:

    li $t3, WHITE

    sw $t3, 0($k0)    # to mau trang

j end
```

set_blue:

li \$t3, BLUE

sw \$t3, 0(\$k0) # to mau xanh

end:

next

addi \$k0, \$k0, 4

j++

addi \$s1, \$s1, 1

blt \$s1, 8, for_column

i++

addi \$s0, \$s0, 1

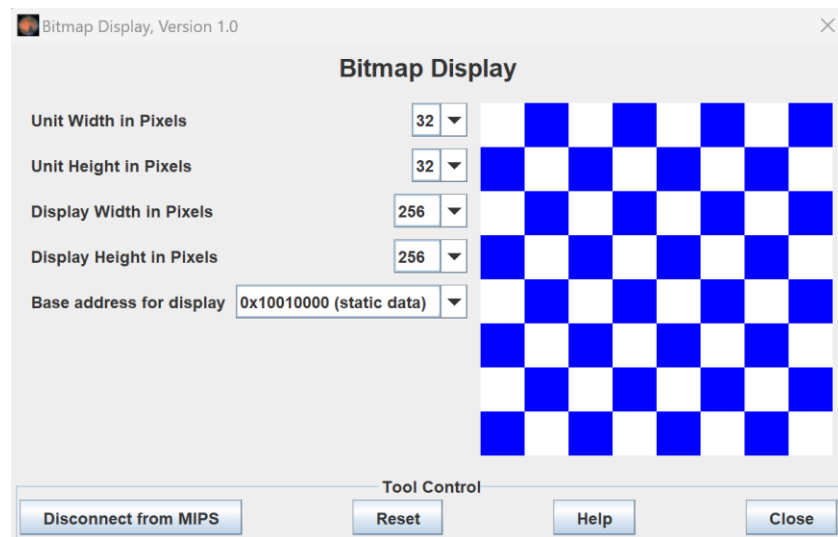
blt \$s0, 8, for_row

exit

li \$v0, 10

syscall

- Kết quả:



→ Kết quả đúng với lí thuyết.

Assignment 5:

- Code:

```
.eqv MONITOR_SCREEN 0x10010000

.eqv RED 0x00FF0000

.eqv GREEN 0x0000FF00

.data

    x1: .asciiz "Nhap x1: "
    y1: .asciiz "Nhap y1: "
    x2: .asciiz "Nhap x2: "
    y2: .asciiz "Nhap y2: "
    error1: .asciiz "Error: x2 phai khac x1. Moi nhap lai!\n"
    error2: .asciiz "Error: y2 phai khac y1. Moi nhap lai!\n"

.text

    li $k0, MONITOR_SCREEN
    li $v0, 4
    la $a0, x1
    syscall

    li $v0, 5
    syscall

    move $s0, $v0

    li $v0, 4
    la $a0, y1
    syscall

    li $v0, 5
    syscall

    move $s1, $v0
```


NhapX2:

```
li $v0, 4
la $a0, x2
syscall
li $v0, 5
syscall
move $s2, $v0
beq $s2, $s0, Error1
```

NhapY2:

```
li $v0, 4
la $a0, y2
syscall
li $v0, 5
syscall
move $s3, $v0
beq $s3, $s1, Error2
j Tsugi
```

Error1:

```
li $v0, 4
la $a0, error1
syscall
j NhapX2
```

Error2:

```
li $v0, 4
la $a0, error2
syscall
```

j NhapY2

Tsugi:

Kiểm tra vị trí của điểm x1, y1 và x2, y2

slt \$t0, \$s0, \$s2

slt \$t1, \$s1, \$s3

Xử lý trường hợp $x1 < x2$ và $y1 < y2$

beq \$t0, 0, Case3

beq \$t1, 0, Case2

Trường hợp $x1 > x2$ và $y1 > y2$

Case1:

Lặp qua từng pixel và tô màu

add \$v0, \$s1, \$zero

For1:

bgt \$v0, \$s3, Exit

add \$v1, \$s0, \$zero

For2:

bgt \$v1, \$s2, EndFor2

beq \$v0, \$s1, InVien1

beq \$v0, \$s3, InVien1

beq \$v1, \$s0, InVien1

beq \$v1, \$s2, InVien1

sll \$t8, \$v0, 6

add \$t8, \$t8, \$v1

sll \$t8, \$t8, 2

li \$a1, GREEN

add \$a2, \$k0, \$t8

sw \$a1, 0(\$a2)

add \$v1, \$v1, 1

j For2

InVien1:

sll \$t8, \$v0, 6

add \$t8, \$t8, \$v1

sll \$t8, \$t8, 2

li \$a1, RED

add \$a2, \$k0, \$t8

sw \$a1, 0(\$a2)

add \$v1, \$v1, 1

j For2

EndFor2:

add \$v0, \$v0, 1

j For1

Trường hợp $x1 < x2$ và $y1 > y2$

Case2:

Lặp qua từng pixel và tô màu

add \$v0, \$s3, \$zero

For3:

bgt \$v0, \$s1, Exit

add \$v1, \$s0, \$zero

For4:

bgt \$v1, \$s2, EndFor4

beq \$v0, \$s1, InVien2

beq \$v0, \$s3, InVien2

beq \$v1, \$s0, InVien2

beq \$v1, \$s2, InVien2

sll \$t8, \$v0, 6

add \$t8, \$t8, \$v1

sll \$t8, \$t8, 2

li \$a1, GREEN

add \$a2, \$k0, \$t8

sw \$a1, 0(\$a2)

add \$v1, \$v1, 1

j For4

InVien2:

sll \$t8, \$v0, 6

add \$t8, \$t8, \$v1

sll \$t8, \$t8, 2

li \$a1, RED

add \$a2, \$k0, \$t8

sw \$a1, 0(\$a2)

add \$v1, \$v1, 1

j For4

EndFor4:

add \$v0, \$v0, 1

j For3

Trường hợp $x1 > x2$ và $y1 < y2$

Case3:

beq \$t1, 0, Case4

add \$v0, \$s1, \$zero

For5:

bgt \$v0, \$s3, Exit
add \$v1, \$s2, \$zero

For6:

bgt \$v1, \$s0, EndFor6
beq \$v0, \$s1, InVien3
beq \$v0, \$s3, InVien3
beq \$v1, \$s0, InVien3
beq \$v1, \$s2, InVien3
sll \$t8, \$v0, 6
add \$t8, \$t8, \$v1
sll \$t8, \$t8, 2
li \$a1, GREEN
add \$a2, \$k0, \$t8
sw \$a1, 0(\$a2)
add \$v1, \$v1, 1
j For6

InVien3:

sll \$t8, \$v0, 6
add \$t8, \$t8, \$v1
sll \$t8, \$t8, 2
li \$a1, RED
add \$a2, \$k0, \$t8
sw \$a1, 0(\$a2)
add \$v1, \$v1, 1
j For6

EndFor6:

add \$v0, \$v0, 1

j For5

Trường hợp $x1 > x2$ và $y1 > y2$

Case4:

add \$v0, \$s3, \$zero

For7:

bgt \$v0, \$s1, Exit

add \$v1, \$s2, \$zero

For8:

bgt \$v1, \$s0, EndFor8

beq \$v0, \$s1, InVien4

beq \$v0, \$s3, InVien4

beq \$v1, \$s0, InVien4

beq \$v1, \$s2, InVien4

sll \$t8, \$v0, 6

add \$t8, \$t8, \$v1

sll \$t8, \$t8, 2

li \$a1, GREEN

add \$a2, \$k0, \$t8

sw \$a1, 0(\$a2)

add \$v1, \$v1, 1

j For8

InVien4:

sll \$t8, \$v0, 6

add \$t8, \$t8, \$v1

sll \$t8, \$t8, 2

li \$a1, RED

add \$a2, \$k0, \$t8

sw \$a1, 0(\$a2)

add \$v1, \$v1, 1

j For8

EndFor8:

add \$v0, \$v0, 1

j For7

Exit:

li \$v0, 10

syscall

- Kết quả: (x1,y2)=(20,40); (x2,y2)=(10,20)

The screenshot displays a MIPS simulator interface. On the left, the assembly code is shown with instructions like `sll $t8, $t8, 2`, `li $a1, RED`, `add $a2, $k0, $t8`, `sw $a1, 0($a2)`, `add $v1, $v1, 1`, `j For8`, `EndFor8:`, `add $v0, $v0, 1`, `j For7`, `Exit:`, `li $v0, 10`, and `syscall`. Below the code, the memory segments are displayed, including the Text Segment and Data Segment. The Data Segment shows memory addresses and values, with a red rectangle highlighting the area from address 0x10010000 to 0x10010010. On the right, the Bitmap Display window shows a black screen with a red rectangle, indicating the drawing area. The Tool Control panel at the bottom includes buttons for Disconnect from MIPS, Reset, Help, and Close.