

Báo cáo TH Kiến trúc máy tính tuần 10:

Bài 1: Hiện thị 2 chữ số cuối của MSSV:

Code:

```
.eqv SEVENSEG_LEFT 0xFFFF0010 # Địa chỉ của đèn led 7 đoạn trái.

# Bit 0 = đoạn a;
# Bit 1 = đoạn b; ...
# Bit 7 = dấu .

.eqv SEVENSEG_RIGHT 0xFFFF0011 # Địa chỉ của đèn led 7 đoạn phải
.text
main:
    li $a0, 0x5B # set value for segments
    jal SHOW_7SEG_LEFT # show
    li $a0, 0x5B # set value for segments
    jal SHOW_7SEG_RIGHT # show
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
    jr $ra

#-----
# Function SHOW_7SEG_RIGHT : turn on/off the 7seg
# param[in] $a0 value to shown
```

remark \$t0 changed

#-----

SHOW_7SEG_RIGHT:

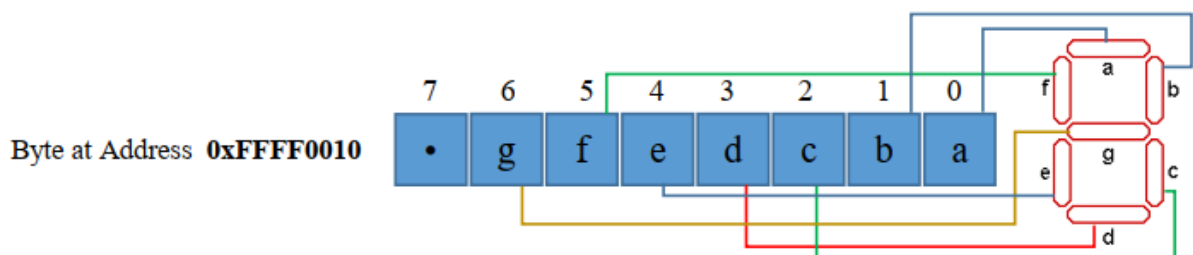
li \$t0, SEVENSEG_RIGHT # assign port's address

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

jr \$ra

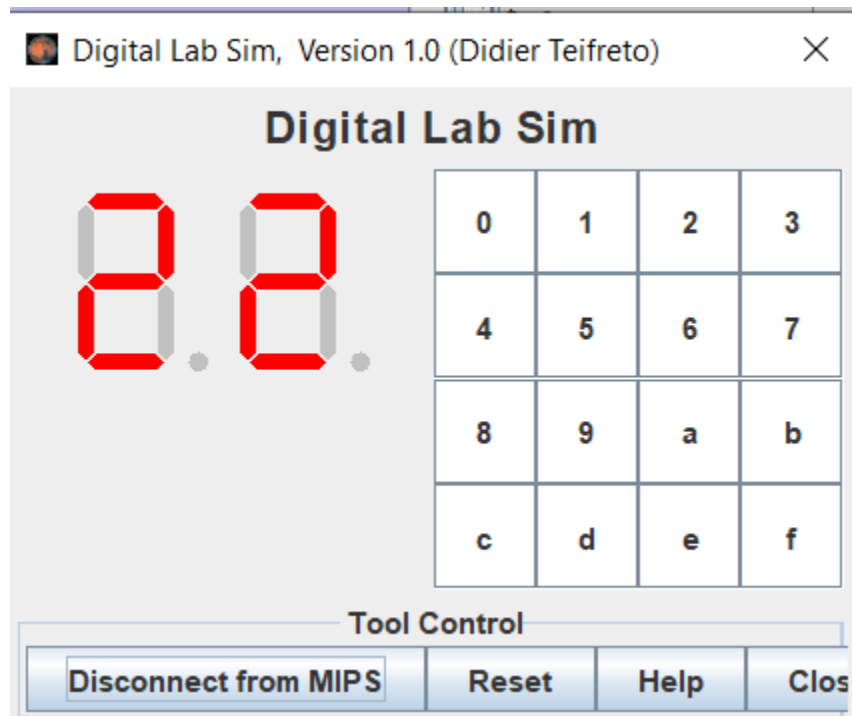
Kết quả:

-Ta cần hiện thị 2 chữ số cuối của MSSV 20194522



- Để hiện thị số 2 thì cần hiện thị các thanh a, b, g, e, d của Sevenseg_left và Sevenseg_right:

- Khi đó mã Hexa: 0101 1011 = 0x5B.



Bài 2: Nhập vào một số nguyên, hiện thị 2 chữ số cuối của số nguyên đó:

Code:

```
.eqv SEVENSEG_LEFT 0xFFFF0010 # Địa chỉ của đèn led 7 đoạn trái.
# Bit 0 = đoạn a;
# Bit 1 = đoạn b; ...
# Bit 7 = dấu .

.eqv SEVENSEG_RIGHT 0xFFFF0011 # Địa chỉ của đèn led 7 đoạn phải

.data
    Mess1: .asciiz "Nhập số nguyên: "

.text

# s1 : Số nguyên vừa nhập
# t3 : Số nguyên cuối cùng thu nhất
# t4 : Số nguyên cuối cùng thu hai

main:

    jal input
    li $t1,10
    div $s1,$t1
    mflo $t2
    mfhi $t3#first
    div $t2,$t1
    mfhi $t4 #second

    add $a0,$zero,$t3

    jal Switch                # set value for segments
    jal SHOW_7SEG_LEFT      # show
    add $a0,$zero,$t4
```

```

        jal Switch                # set value for segments
        jal SHOW_7SEG_RIGHT      # show
exit:
        li $v0, 10
        syscall
endmain:

```

Nhap du lieu

input:

```

        la $a0,Mess1
        li $v0,4                # Nhap so nguyen:
        syscall

        li $v0,5
        syscall                # Nhap du lieu tu ban phim
        add $s1,$zero,$v0
        jr $ra

```

#-----

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

jr \$ra

#-----

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

jr \$ra

Switch:

li \$t8,0

beq \$a0,\$t8,CASE0

li \$t8,1

beq \$a0,\$t8,CASE1

li \$t8,2

beq \$a0,\$t8,CASE2

li \$t8,3

beq \$a0,\$t8,CASE3

li \$t8,4

beq \$a0,\$t8,CASE4

li \$t8,5

beq \$a0,\$t8,CASE5

li \$t8,6

beq \$a0,\$t8,CASE6

li \$t8,7

beq \$a0,\$t8,CASE7

li \$t8,8

beq \$a0,\$t8,CASE8

li \$t8,9

beq \$a0,\$t8,CASE9

CASE0:

li \$a0,0x3f

jr \$ra

CASE1:

li \$a0,0x6

jr \$ra

CASE2:

li \$a0,0x5b

jr \$ra

CASE3:

li \$a0,0x4f

jr \$ra

CASE4:

li \$a0,0x66

jr \$ra

CASE5:

li \$a0,0x6d

jr \$ra

CASE6:

li \$a0,0x7d

jr \$ra

CASE7:

li \$a0,0x7

jr \$ra

CASE8:

li \$a0,0x7f

jr \$ra

CASE9:

li \$a0,0x6f

jr \$ra

Kết quả:

-Ý tưởng: Sau khi nhập số nguyên. Ta lấy số nguyên chia cho div 10 để lấy số dư và lưu vào các thanh ghi tương ứng t3, t4.

- Sau khi tách được số ta sẽ tạo switch case để lấy ra giá trị tương ứng lưu vào \$a0 rồi in ra segment.

- Chuyển các số hiện thị tương ứng thành Switch case:

	.	g	f	e	d	c	b	a	Hexa
0	0	0	1	1	1	1	1	1	0x3f
1	0	0	0	0	0	1	1	0	0x6
2	0	1	0	1	1	0	1	1	0x5b
3	0	1	0	0	1	1	1	1	0x4f
4	0	1	1	0	0	1	1	0	0x66
5	0	1	1	0	1	1	0	1	0x6d
6	0	1	1	1	1	1	0	1	0x7d
7	0	0	0	0	0	1	1	1	0x7
8	0	1	1	1	1	1	1	1	0x7f
9	0	1	1	0	1	1	1	1	0x6f

```
Clear
Nhap so nguyen: **** user input : 123453
-- program is finished running --
```

Digital Lab Sim, Version 1.0 (Didier Teifreto)

Digital Lab Sim

88.

0	1	2	3
4	5	6	7
8	9	a	b
c	d	e	f

Tool Control

Disconnect from MIPS Reset Help Clos

Bài 3: Nhập vào ký tự, hiển thị 2 chữ số cuối của mã ASCII của ký tự đó

Code:

```
.eqv SEVENSEG_LEFT 0xFFFF0010 # Địa chỉ của đèn led 7 đoạn trái.

# Bit 0 = đoạn a;
# Bit 1 = đoạn b; ...
# Bit 7 = dấu .

.eqv SEVENSEG_RIGHT 0xFFFF0011 # Địa chỉ của đèn led 7 đoạn phải

.data
    Mess1: .asciiz "Nhập kí tự:"

.text
input:
    la $a0,Mess1
    li $v0,4
    syscall
    li $v0,12
    syscall
    add $s1,$zero,$v0

main:
    li $t1,10
    div $s1,$t1
    mflo $t2
    mfhi $t3#first
    div $t2,$t1
    mfhi $t4 #second
    add $a0,$zero,$t3
    jal Switch
    jal SHOW_7SEG_LEFT # show
    add $a0,$zero,$t4
    jal Switch
```



```

jal SHOW_7SEG_RIGHT # show
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
jr $ra
#-----
# 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
jr $ra
Switch:
    li $t8,0
    beq $a0,$t8,CASE0
    li $t8,1
    beq $a0,$t8,CASE1
    li $t8,2
    beq $a0,$t8,CASE2
    li $t8,3
    beq $a0,$t8,CASE3

```

```
li $t8,4
beq $a0,$t8,CASE4
li $t8,5
beq $a0,$t8,CASE5
li $t8,6
beq $a0,$t8,CASE6
li $t8,7
beq $a0,$t8,CASE7
li $t8,8
beq $a0,$t8,CASE8
li $t8,9
beq $a0,$t8,CASE9
```

CASE0:

```
li $a0,0x3f
jr $ra
```

CASE1:

```
li $a0,0x6
jr $ra
```

CASE2:

```
li $a0,0x5b
jr $ra
```

CASE3:

```
li $a0,0x4f
jr $ra
```

CASE4:

```
li $a0,0x66
jr $ra
```

CASE5:

```
li $a0,0x6d
```

jr \$ra

CASE6:

li \$a0,0x7d

jr \$ra

CASE7:

li \$a0,0x7

jr \$ra

CASE8:

li \$a0,0x7f

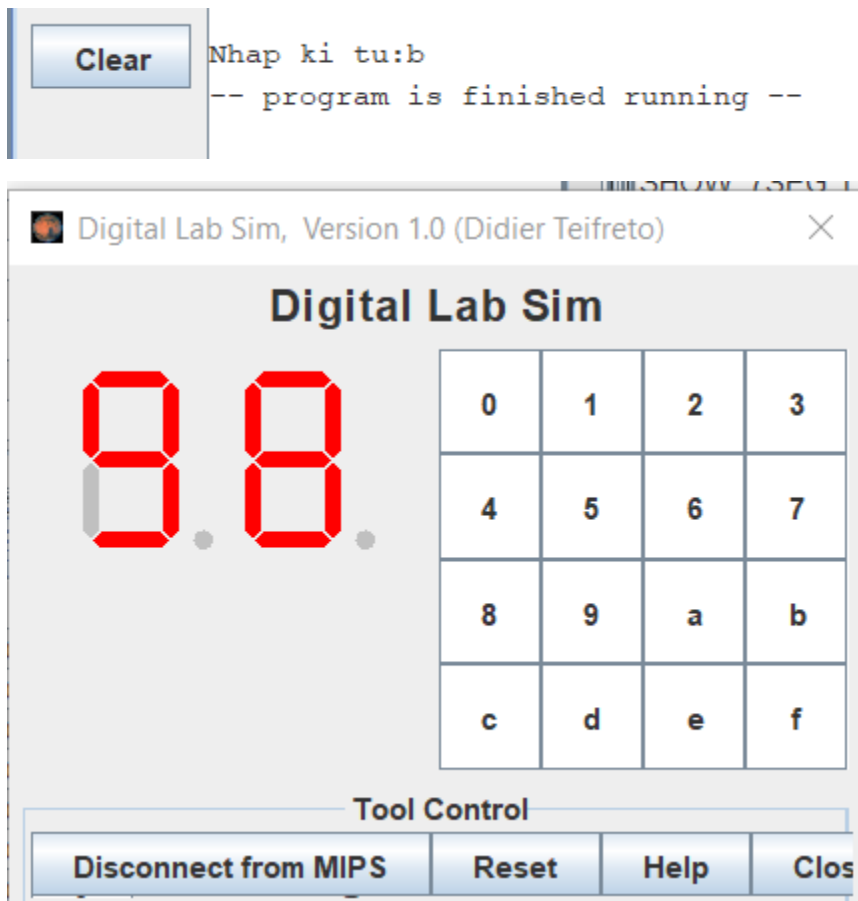
jr \$ra

CASE9:

li \$a0,0x6f

jr \$ra

Kết quả: Cách làm tương tự bài 3



Bài 4. Vẽ ô bàn cờ vua kích thước 8x8

Code:

```
.eqv MONITOR_SCREEN 0x10010000
```

```
.eqv WHITE 0x00FFFFFF
```

```
.text
```

```
li $k0, MONITOR_SCREEN
```

```
li $a0, WHITE
```

```
li $a2, -1 # i = -1
```

```
li $a1, 65
```

```
li $a3, 2
```

```
li $s0, 8
```

```
loop:
```

```
addi $a2, $a2, 1
```

```
beq $a2, $a1, exit
```

```
div $a2, $a3
```

```
mfhi $t0 # du cua hang /2
```

```
div $a2, $s0
```

```
mflo $t1
```

```
div $t1, $a3
```

```
mfhi $t1 # so cot
```

```
bne $t0, $t1, loop
```

```
sll $t0, $a2, 2
```

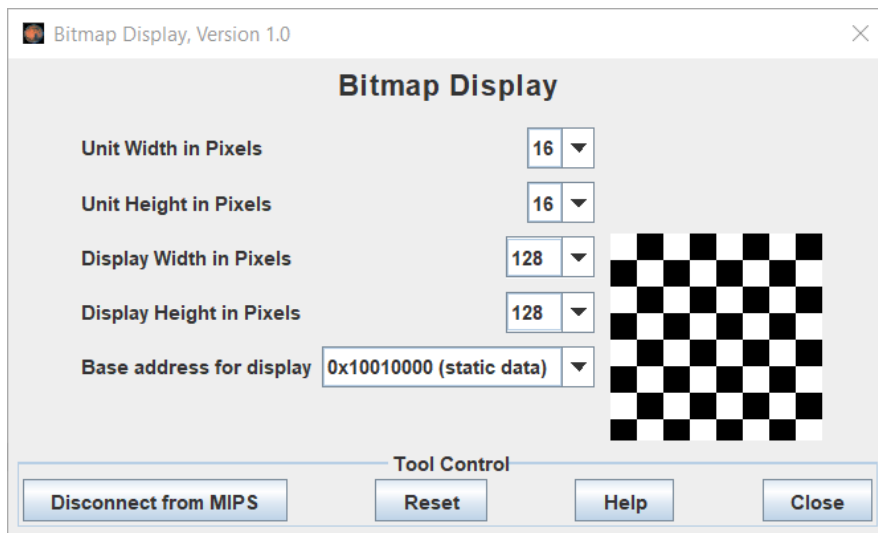
```
add $t0, $t0, $k0
```

```
sw $a0, 0($t0) # White
```

```
j loop
```

```
exit:
```

Kết quả:



Bài 5. Nhập vào 2 điểm có tọa độ (x1, y1) và (x2, y2), vẽ và tô màu hình chữ nhật có 2 góc là 2 điểm trên.

Code:

.data

Mess1: .asciiz "Nhap toa do: "

Mess2: .asciiz "\n "

MessX1: .asciiz "x1: "

MessY1: .asciiz "y1: "

MessX2: .asciiz "x2: "

MessY2: .asciiz "y2: "

.eqv MONITOR_SCREEN 0x10010000

.eqv RED 0x00FF0000

.eqv GREEN 0x0000FF00

.eqv BLUE 0x000000FF

.eqv WHITE 0x00FFFFFF

.eqv YELLOW 0x00FFFF00

.text

```
li $k0, MONITOR_SCREEN
```

```
main:
```

```
# ve vien hcn
```

```
jal input
```

```
li $t1, 0 # i = 0
```

```
li $t2, 0 # j = 0
```

```
li $t0, RED
```

```
jal loop1
```

```
# ve hcn
```

```
addi $s1,$s1,1
```

```
addi $s2,$s2,1
```

```
addi $s3,$s3,-1
```

```
addi $s4,$s4,-1
```

```
li $t1, 0 # i = 0
```

```
li $t2, 0 # j = 0
```

```
li $t0, GREEN
```

```
jal loop1
```

```
exit:
```

```
li $v0, 10
```

```
syscall
```

```
#Vong lap
```

```
loop1:
```

```
slti $t3, $t1, 16
```

```
beq $t3, $zero, endloop1
```

j loop2

loop2:

sll \$t4, \$t1, 6

slti \$t3, \$t2, 16

beq \$t3, \$zero, endloop2

sll \$t5, \$t2, 2

add \$t5, \$t4, \$t5

add \$t5, \$t5, \$k0 # $t5 = k0 + 64i + 4j$

addi \$t2, \$t2, 1 # tang j len 1

j if

endloop2:

addi \$t1, \$t1, 1 # tang i len 1

li \$t2, 0

j loop1

endloop1:

jr \$ra

kiem tra dieu kien de ve

if:

slt \$t3, \$s2, \$t1 # $y1 < i$

beqz \$t3, loop2 # neu $t1 \leq x1$ tro ve vong lap loop2

```
slt $t3, $t1, $s4      # i < y2
beqz $t3, loop2         # neu t1 >= x2 tro ve vong lap loop2
```

```
slt $t3, $s1, $t2      # x1 < j
beqz $t3, loop2
```

```
slt $t3, $t2, $s3      # x2 < j
beqz $t3, loop2
```

```
sw $t0, 0($t5)
j loop2
```

nhap du lieu

input:

```
la $a0, Mess1
li $v0, 4      # Nhap toa do:
syscall
```

```
la $a0, Mess2
li $v0, 4
syscall
```


la \$a0,MessX1

li \$v0,4 # x1:

syscall

li \$v0,5

syscall # Luu du lieu tu ban phim

addi \$s1,\$v0,-1

la \$a0,MessY1

li \$v0,4 # y1:

syscall

li \$v0,5

syscall # Luu du lieu tu ban phim

addi \$s2,\$v0,-2

la \$a0,Mess1

li \$v0,4 # Nhap toa do:

syscall

la \$a0,Mess2

li \$v0,4

syscall

la \$a0,MessX2

li \$v0,4 # x2:

```
syscall
```

```
li $v0,5
```

```
syscall      # Luu du lieu tu ban phim
```

```
addi $s3,$v0,2
```

```
la $a0,MessY2
```

```
li $v0,4      # y2:
```

```
syscall
```

```
li $v0,5
```

```
syscall      # Luu du lieu tu ban phim
```

```
addi $s4,$v0,1
```

```
jr $ra
```

Kết quả:

Clear	Nhap toa do:
	x1: 3
	y1: 4
	Nhap toa do:
	x2: 8
	y2: 8

Bitmap Display

Unit Width in Pixels

16 ▼

Unit Height in Pixels

16 ▼

Display Width in Pixels

256 ▼

Display Height in Pixels

256 ▼

Base address for display 0x10010000 (static data) ▼

