

Báo cáo week10

Phùng Ngọc Vinh – 20194719

Câu 1 : Hiển thị 2 chữ số cuối của MSSV

Mã nguồn:

```
.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, 0x6F            # set value for segments
    jal SHOW_7SEG_LEFT      # show
    li  $a0, 0x6             # 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

#-----

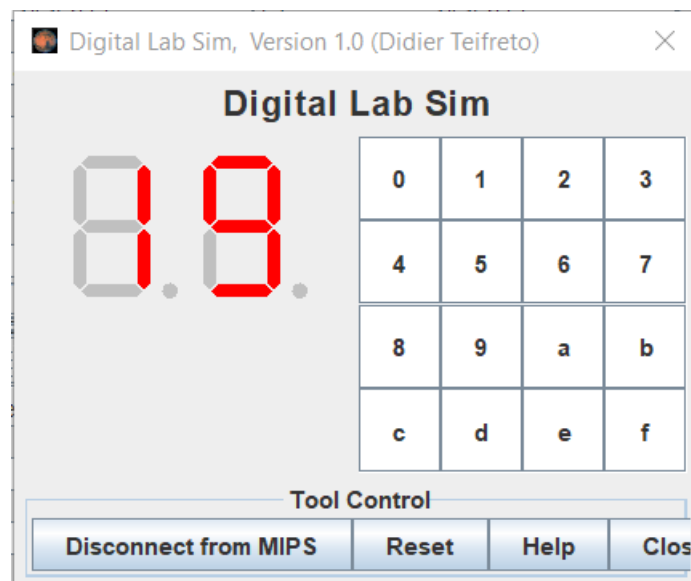
# Function SHOW_7SEG_RIGHT : turn on/off the 7seg
# param[in] $a0  value to shown
# remark  $t0 changed
#-----

SHOW_7SEG_RIGHT: li $t0, SEVENSEG_RIGHT  # assign port's
address

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

```

Kết quả chạy:



Câu 2: Nhập vào một số nguyên, hiển thị 2 chữ số cuối của số nguyên đó

Mã nguồn:

```
.eqv SEVENSEG_LEFT  0xFFFF0010    # Dia chi cua den led 7 doan trai.
                        #   Bit 0 = doan a;
                        #   Bit 1 = doan b; ...
                        #   Bit 7 = dau .

.eqv SEVENSEG_RIGHT 0xFFFF0011    # Dia chi cua den led 7 doan phai

.data
mess: .asciiz "Nhap so nguyen:"

.text
main:
    li $v0,51
    la $a0, mess
    syscall
    addi $s0,$0,10
    div $a0,$s0
    mfhi $t1
    mflo $t2

Left:
    jal check

endLeft:
```

```

        jal SHOW_7SEG_LEFT      # show
Right:
        div $t2,$s0
        mfhi $t1
        jal check
endRight:
        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
```

check:

```
li $t8,0
```

```
beq $t1,$t8,CASE0
```

```
li $t8,1
```

```
beq $t1,$t8,CASE1
```

```
li $t8,2
```

```
beq $t1,$t8,CASE2
```

```
li $t8,3
```

```
beq $t1,$t8,CASE3
```

```
li $t8,4
```

```
beq $t1,$t8,CASE4
```

```
li $t8,5
```

```
beq $t1,$t8,CASE5
```

```
li $t8,6
```

```
beq $t1,$t8,CASE6
```

```
li $t8,7
```

```
beq $t1,$t8,CASE7
```

```
li $t8,8
```

```
beq $t1,$t8,CASE8
```

```
li $t8,9
```

```
beq $t1,$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:

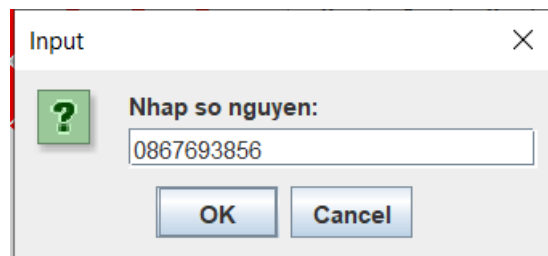
```
li $a0,0x6f
```

```
jr $ra
```

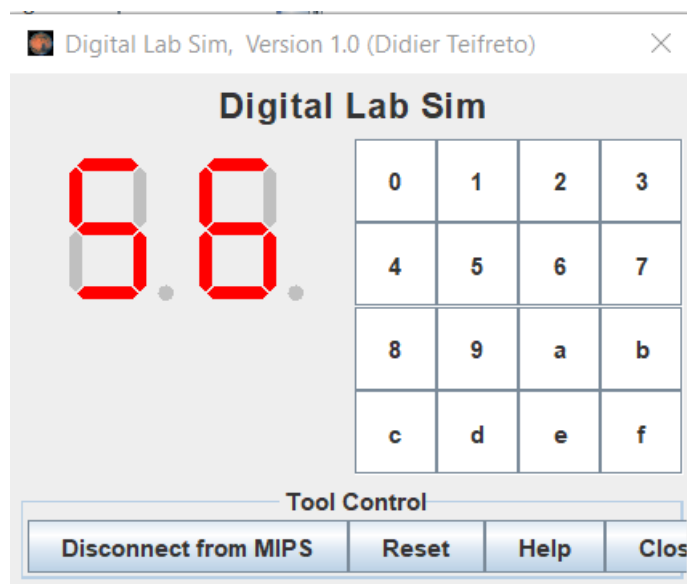
Kết quả:

*TH1: Số nguyên có từ 2 chữ số trở lên:

- Nhập số nguyên:



Input dialog box with a green question mark icon. The text "Nhập số nguyên:" is displayed above a text input field containing "0867693856". Below the input field are "OK" and "Cancel" buttons.



Digital Lab Sim, Version 1.0 (Didier Teifreto)

Digital Lab Sim

08.

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

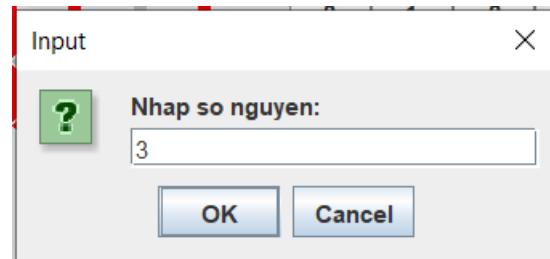
Tool Control

Disconnect from MIPS Reset Help Clos

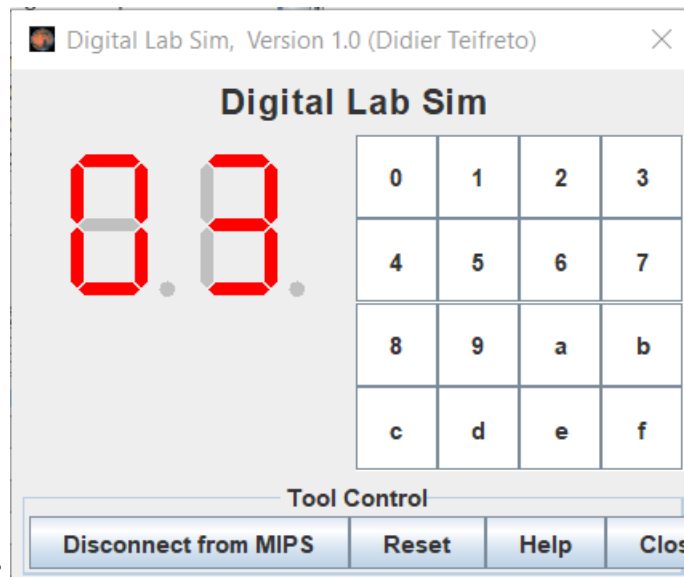
-Hiển thị:

*TH2: Số nguyên là số trong khoảng [0;9] chữ số trở lên:

- Nhập số nguyên:



A small dialog box titled "Input" with a close button (X) in the top right corner. It contains a green square icon with a question mark, the text "Nhập số nguyên:" (Enter integer:), a text input field containing the number "3", and two buttons labeled "OK" and "Cancel" at the bottom.



The main window of the "Digital Lab Sim" software, version 1.0 by Didier Teifreto. It features a title bar, a close button (X), and a main area with the title "Digital Lab Sim". On the left, there are two red 7-segment displays showing the number "08". To the right of the displays is a 4x4 grid of buttons labeled with hexadecimal digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f. At the bottom, there is a "Tool Control" section with four buttons: "Disconnect from MIPS", "Reset", "Help", and "Close".

- Hiển thị:

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

Mã nguồn:

```
.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
mess: .asciiz "Nhap ki tu:"

.text
main:
    la $a0,mess
    li $v0,4
    syscall
    li $v0,12
    syscall
    addi $s0,$0,10
    div $v0,$s0
    mfhi $t1
    mflo $t2
Left:
    jal check
endLeft:
    jal SHOW_7SEG_LEFT    # show
Right:
    div $t2,$s0
    mfhi $t1
    jal check
endRight:

```

```

        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
check:
        li $t8, 0
        beq $t1, $t8, CASE0

```

```
li $t8,1
beq $t1,$t8,CASE1
li $t8,2
beq $t1,$t8,CASE2
li $t8,3
beq $t1,$t8,CASE3
li $t8,4
beq $t1,$t8,CASE4
li $t8,5
beq $t1,$t8,CASE5
li $t8,6
beq $t1,$t8,CASE6
li $t8,7
beq $t1,$t8,CASE7
li $t8,8
beq $t1,$t8,CASE8
li $t8,9
beq $t1,$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

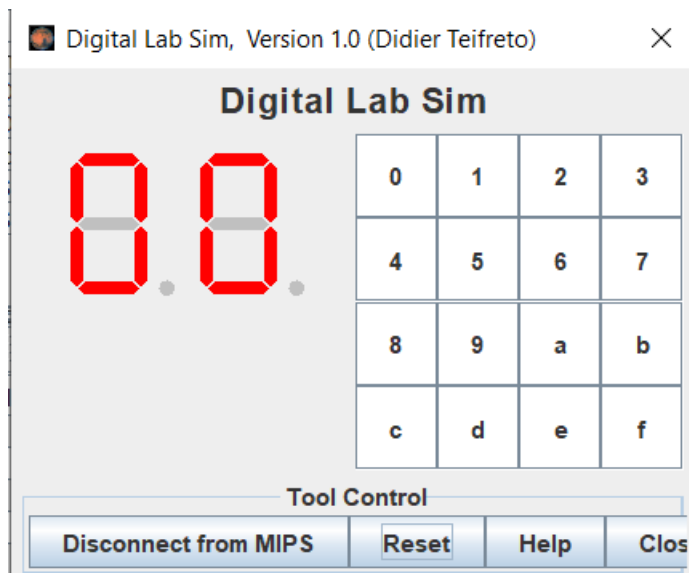
jr \$ra

Kết quả:

-nhập kí tự 'd' từ bàn phím:

```
Nhap ki tu:d
-- program is finished running --
```

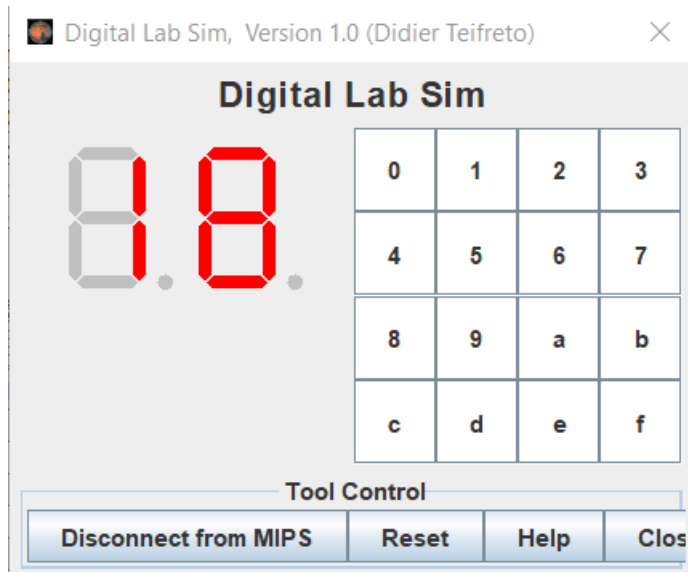
-hiển thị



-nhập kí tự 'v' từ bàn phím:

```
Nhap ki tu:v
-- program is finished running --
```

-hiển thị:



Câu 4: Vẽ ô bàn cờ vua kích thước 8x8

Mã nguồn:

```
.eqv MONITOR_SCREEN 0x10010000
.eqv WHITE      0x00FFFFFF

.text
    li $k0, MONITOR_SCREEN
    addi $s0,$0,0
    addi $t2,$0,65
```

```

        addi $t3,$0,16
loop:
        beq $s0,$t2,endloop
        div $s0,$t3
        mfhi $t4
        addi $t7,$0,3
        addi $t6,$0,4
        addi $t5,$0,0
        beq $t4,$0,re1
        addi $s0,$s0,1
re2:
        beq $t5,$t6,loop
        sll $t1,$s0,2
        add $t1,$t1,$k0
        li $t0, WHITE
        sw $t0, 0($t1)

        addi $s0,$s0,2
        beq $t5,$t7,re3
tang:
        addi $t5,$t5,1
        j re2
endloop:

```

re1:

beq \$t5,\$t6,loop

sll \$t1,\$s0,2

add \$t1,\$t1,\$k0

li \$t0, WHITE

sw \$t0, 0(\$t1)

addi \$s0,\$s0,2

addi \$t5,\$t5,1

j re1

re3:

sub \$s0,\$s0,1

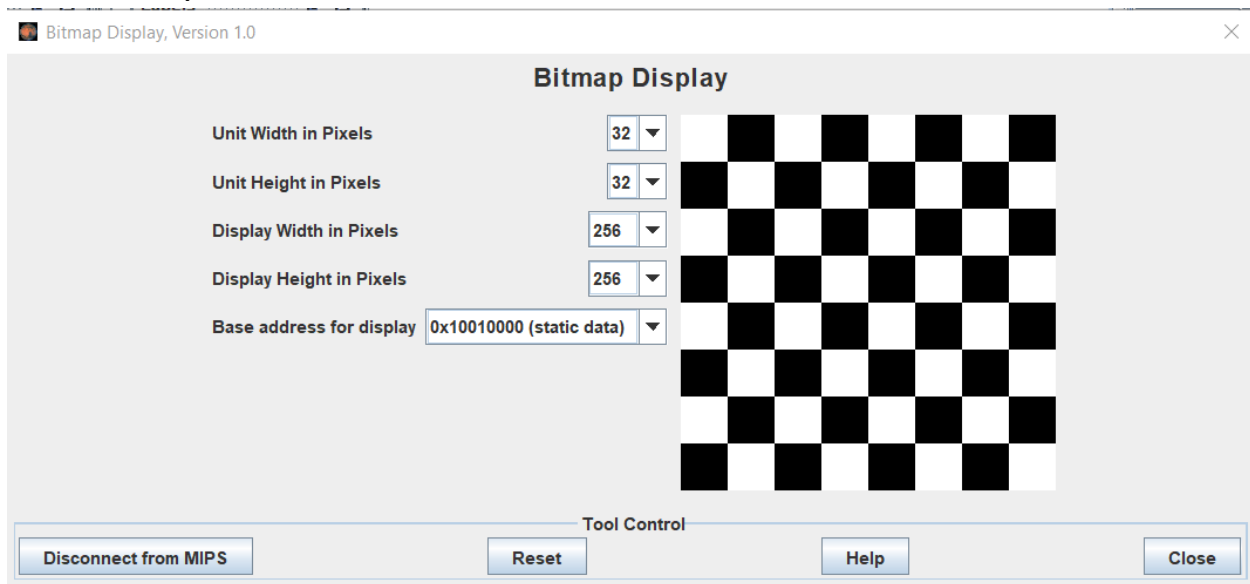
j tang

exit:

li \$v0, 10

syscall

Kết quả:



Câu 5: Vẽ hình chữ nhật

Mã nguồn:

```
.eqv MONITOR_SCREEN 0x10010000
.eqv RED 0x00FF0000
.eqv GREEN 0x0000FF00
.eqv BLUE 0x000000FF
.eqv WHITE 0x00FFFFFF
.eqv YELLOW 0x00FFFF00
.data
Message1: .asciiz "Nhap hoanh do x1"
Message2: .asciiz "Nhap tung do y1"
Message3: .asciiz "Nhap hoanh do x2"
Message4: .asciiz "Nhap tung do y2"
.text
```

main:

li \$v0, 51 # doc hoan h do x1 la \$a0, Message1

la \$a0, Message1

syscall

addi \$s1,\$a0,0

li \$v0, 51 # doc tung do y1 la \$a0, Message2

la \$a0, Message2

syscall

addi \$s2,\$a0,0

li \$v0, 51 # doc hoan h do x2 la \$a0, Message3

la \$a0, Message3

syscall

addi \$s3,\$a0,0

li \$v0, 51 # doc tung do y2 la \$a0, Message4

la \$a0, Message4

syscall

addi \$s4,\$a0,0

li \$k0, MONITOR_SCREEN

check_hoanh:

slt \$t0,\$s1,\$s3

beqz \$t0,luu1

addi \$t4,\$s1,0

```
        addi $t5,$s3,0
        j check_tung
luu1:
        addi $t4,$s3,0
        addi $t5,$s1,0
check_tung:
        slt $t0,$s2,$s4
        beqz $t0,luu2
        addi $t6,$s2,0
        addi $t7,$s4,0
        j end_check
luu2:

        addi $t6,$s4,0
        addi $t7,$s2,0
end_check:
        mul $s1,$t4,8
        add $s1,$s1,$t6
        mul $s1,$s1,4
        add $k0,$k0,$s1
        sub $s2,$t5,$t4
        sub $s3,$t7,$t6
        addi $s2,$s2,1
```

```
addi $s3,$s3,1
mul $s4,$s3,4
mflo $s4
li $s5,32
sub $s5,$s5,$s4
li $t0, 0
li $t1, 0
```

loop1:

```
beq $t0, $s2, endloop1
```

loop:

```
beq $t1, $s3, endloop
li $t2, YELLOW
sw $t2, 0($k0)
addi $k0, $k0, 4
addi $t1,$t1,1
j loop
```

endloop:

```
addi $t0,$t0,1
li $t1, 0
add $k0, $k0, $s5
j loop1
```

endloop1:

```
subi $k0, $k0, 32
```

```
addi $v0,$k0,0
```

```
li $t0, 0
```

```
v1:
```

```
beq $t0, $s3, endv1
```

```
li $t2, BLUE
```

```
sw $t2, 0($v0)
```

```
addi $v0, $v0, 4
```

```
addi $t0,$t0,1
```

```
j v1
```

```
endv1:
```

```
addi $v0,$k0,0
```

```
li $t0, 0
```

```
v2:
```

```
beq $t0, $s2, endv2
```

```
li $t2, BLUE
```

```
sw $t2, 0($v0)
```

```
subi $v0, $v0, 32
```

```
addi $t0,$t0,1
```

```
j v2
```

```
endv2:
```

```
addi $v0,$v0,32
```

```
addi $k0,$v0,0
```

```
li $t0, 0
```

```
v3:
```

```
beq $t0, $s3, endv3
```

```
li $t2, BLUE
```

```
sw $t2, 0($k0)
```

```
addi $k0, $k0, 4
```

```
addi $t0,$t0,1
```

```
j v3
```

```
endv3:
```

```
subi $k0,$k0,4
```

```
li $t0, 0
```

```
v4:
```

```
beq $t0, $s2, endv4
```

```
li $t2, BLUE
```

```
sw $t2, 0($k0)
```

```
addi $k0, $k0, 32
```

```
addi $t0,$t0,1
```

```
j v4
```

```
endv4:
```

```
endmain:
```

Kết quả:

*Nhập A(x1,y1)

Input

×

?

Nhap hoành đo x1

5

OK

Cancel

Input

×

?

Nhap tung đo y1

6

OK

Cancel

*Nhập B(x2,y2)

Input

×

?

Nhap hoành đo x2

4

OK

Cancel

Input

×

?

Nhap tung đo y2

7

OK

Cancel

*Hiển thị

Bitmap Display, Version 1.0

×

Bitmap Display

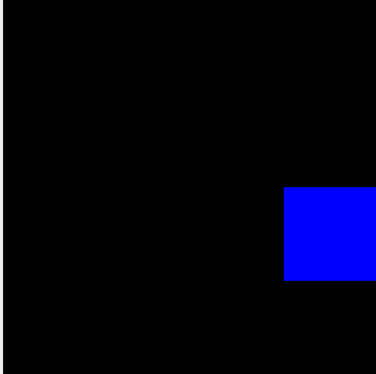
Unit Width in Pixels32

Unit Height in Pixels32

Display Width in Pixels256

Display Height in Pixels256

Base address for display0x10010000 (static data)



Tool Control

Disconnect from MIPS

Reset

Help

Close

*Nhập A(x1,y1)

Input

×

?

Nhap hoành đo x1

1

OK

Cancel

Input

×

?

Nhap tung đo y1

1

OK

Cancel

*Nhập B(x2,y2)

Input

×

?

Nhap hoành đo x2

6

OK

Cancel

Input

×

?

Nhap tung đo y2

6

OK

Cancel

*Hiển thị

Bitmap Display, Version 1.0

×

Bitmap Display

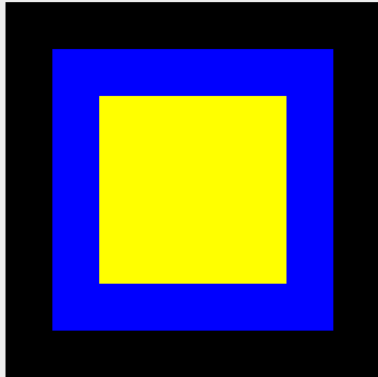
Unit Width in Pixels32

Unit Height in Pixels32

Display Width in Pixels256

Display Height in Pixels256

Base address for display0x10010000 (static data)



Tool Control

Disconnect from MIPS

Reset

Help

Close