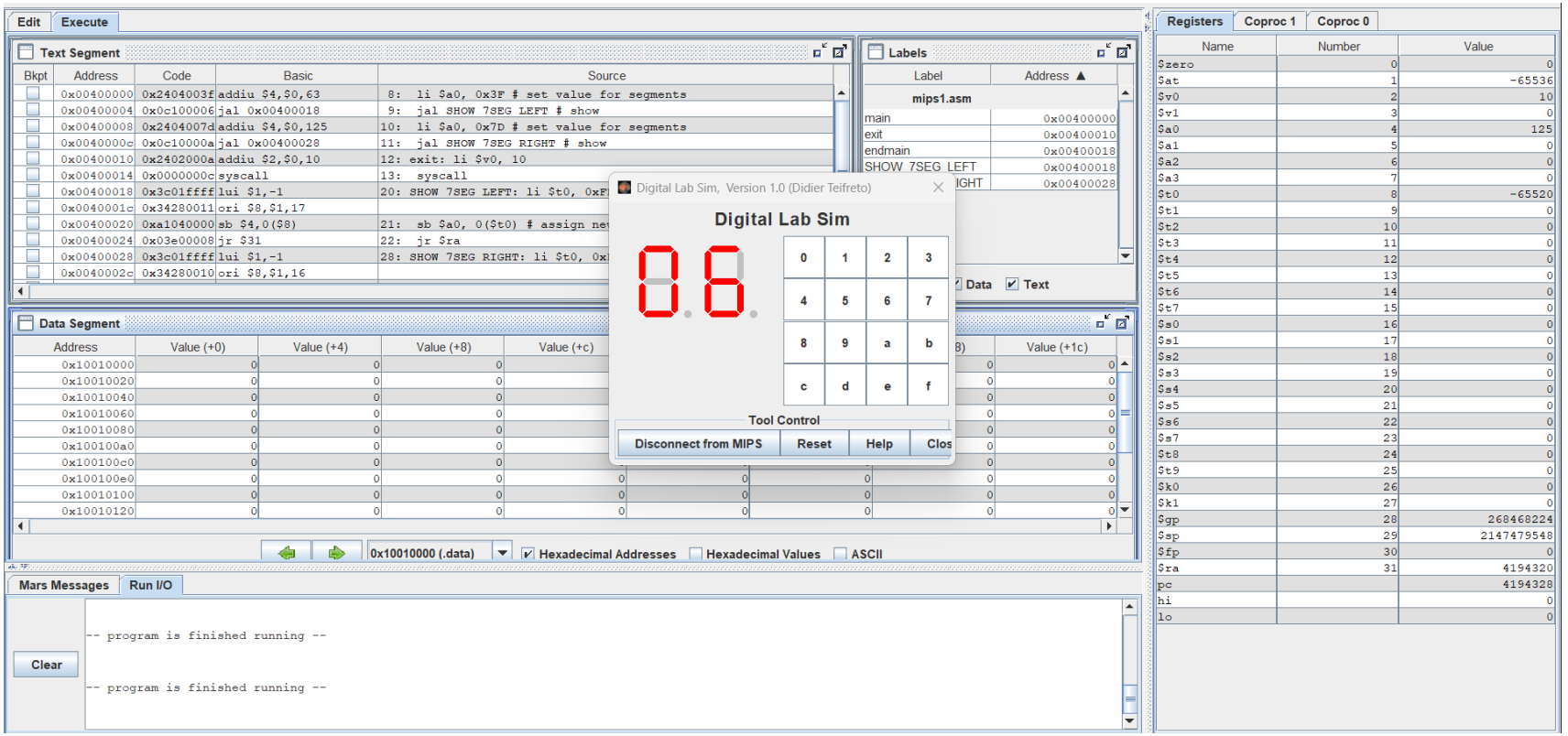
Laboratory Exercise 10

Tools: LED PORT, BITMAP DISPLAY

1. **Assignment 1:** *Hiển thị 2 chữ số cuối của MSSV*

```
.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, 0x3F # set value for segments
 jal SHOW_7SEG_LEFT # show
 li $a0, 0x7D # 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
```

- MSSV của em: 20215006 -> Kết quả mong muốn được in ra: 06
- Thực tế:



2. **Assignment 2:** *Nhập vào một số nguyên từ bàn phím, hiển thị 2 chữ số cuối của số nguyên đó*
   - Mã:

```
#Laboratory Exercise 10, Assignment 2
.eqv SEVENSEG_LEFT    0xFFFF0011    # Address of led 7 (left)
                                    #  Bit0=doana;
                                    # Bit1=doanb;...
                                    # Bit7=dau.
.eqv SEVENSEG_RIGHT   0xFFFF0010    # Address of led 7 (right)

.text
main:
        li $v0, 5
```

```
        syscall # nhap so

        div $v0, $v0, 100
        mfhi $a1

        div $a1, $a1, 10

        mflo    $s1     # set value for segments
        jal CHECK
        jal   SHOW_7SEG_LEFT   # show

        mfhi    $s1     # set value for segments
        jal CHECK
        jal   SHOW_7SEG_RIGHT # show
exit:
        li    $v0, 10
        syscall

endmain:

#---------------------------------check number--------------

CHECK:

        beq $s1, 0, case_0
        beq $s1, 1, case_1
        beq $s1, 2, case_2
        beq $s1, 3, case_3
        beq $s1, 4, case_4
        beq $s1, 5, case_5
        beq $s1, 6, case_6
        beq $s1, 7, case_7
        beq $s1, 8, case_8
```

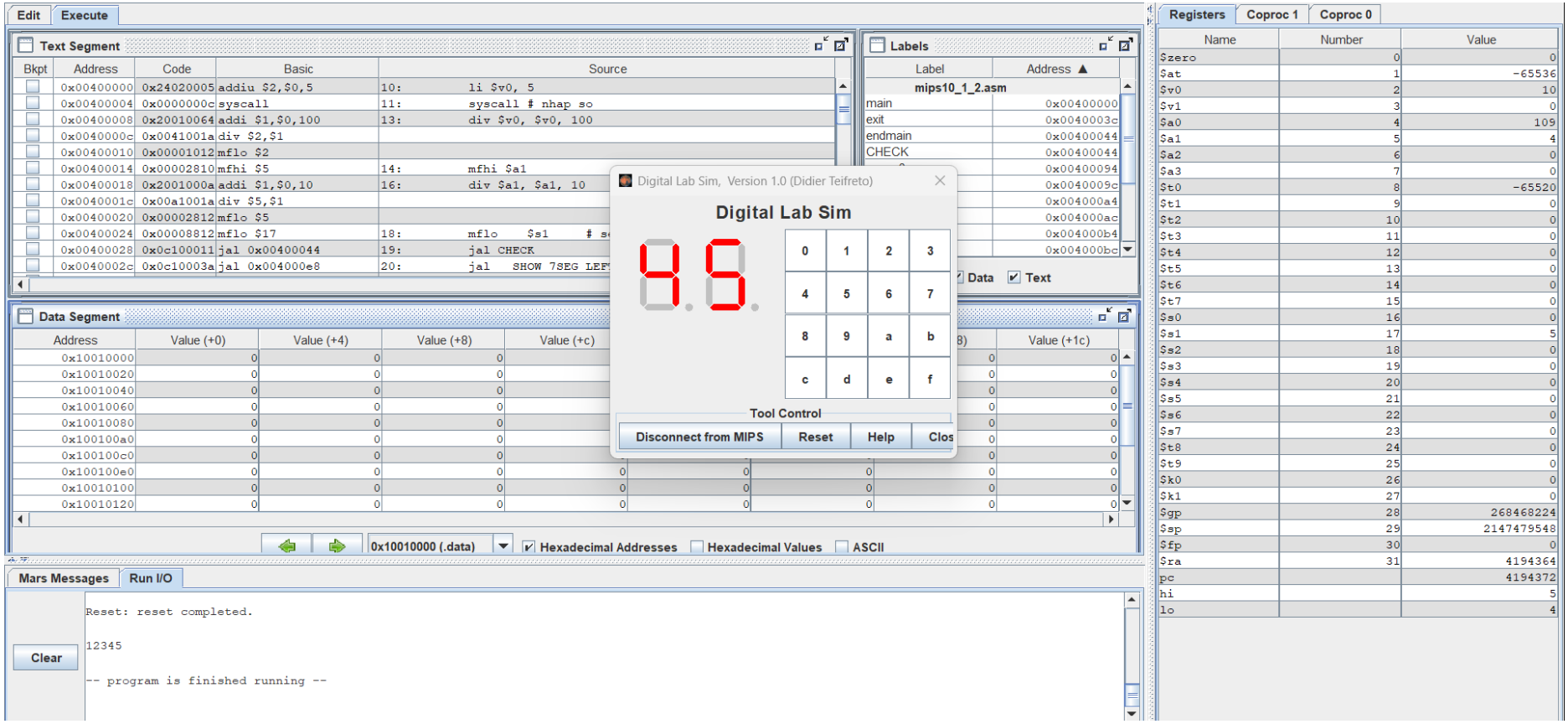
```
        beq $s1, 9, case_9

case_0:
        li $a0, 0x3f
        j end_check
case_1:
        li $a0, 0x06
        j end_check
case_2:
        li $a0, 0x5b
        j end_check
case_3:
        li $a0, 0x4f
        j end_check
case_4:
        li $a0, 0x66
        j end_check
case_5:
        li $a0, 0x6d
        j end_check
case_6:
        li $a0, 0x7d
        j end_check
case_7:
        li $a0, 0x07
        j end_check
case_8:
        li $a0, 0x7f
        j end_check
case_9:
        li $a0, 0x6f
        j end_check
end_check:
```

```
        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
```

- Kết quả
  Input: 12345
  Output: 45



3. **Assignment 3:** *Nhập vào ký tự, hiển thị 2 chữ số cuối của mã ASCII của ký tự đó*
   - Mã chương trình:

```
#Laboratory Exercise 10, Assignment 3
.eqv SEVENSEG_LEFT    0xFFFF0011    # Address of led 7 (left)
                                    #  Bit0=doana;
                                    # Bit1=doanb;...
                                    # Bit7=dau.
.eqv SEVENSEG_RIGHT   0xFFFF0010    # Address of led 7 (right)

.text
main:
```

```
        li $v0, 12
        syscall # nhap so



        div $v0, $v0, 100
        mfhi $a1

        div $a1, $a1, 10

        mflo    $s1     # set value for segments
        jal CHECK
        jal   SHOW_7SEG_LEFT   # show

        mfhi    $s1     # set value for segments
        jal CHECK
        jal   SHOW_7SEG_RIGHT # show
exit:
        li    $v0, 10
        syscall

endmain:

#-----------------------------------check number--------------

CHECK:

        beq $s1, 0, case_0
        beq $s1, 1, case_1
        beq $s1, 2, case_2
        beq $s1, 3, case_3
        beq $s1, 4, case_4
        beq $s1, 5, case_5
```

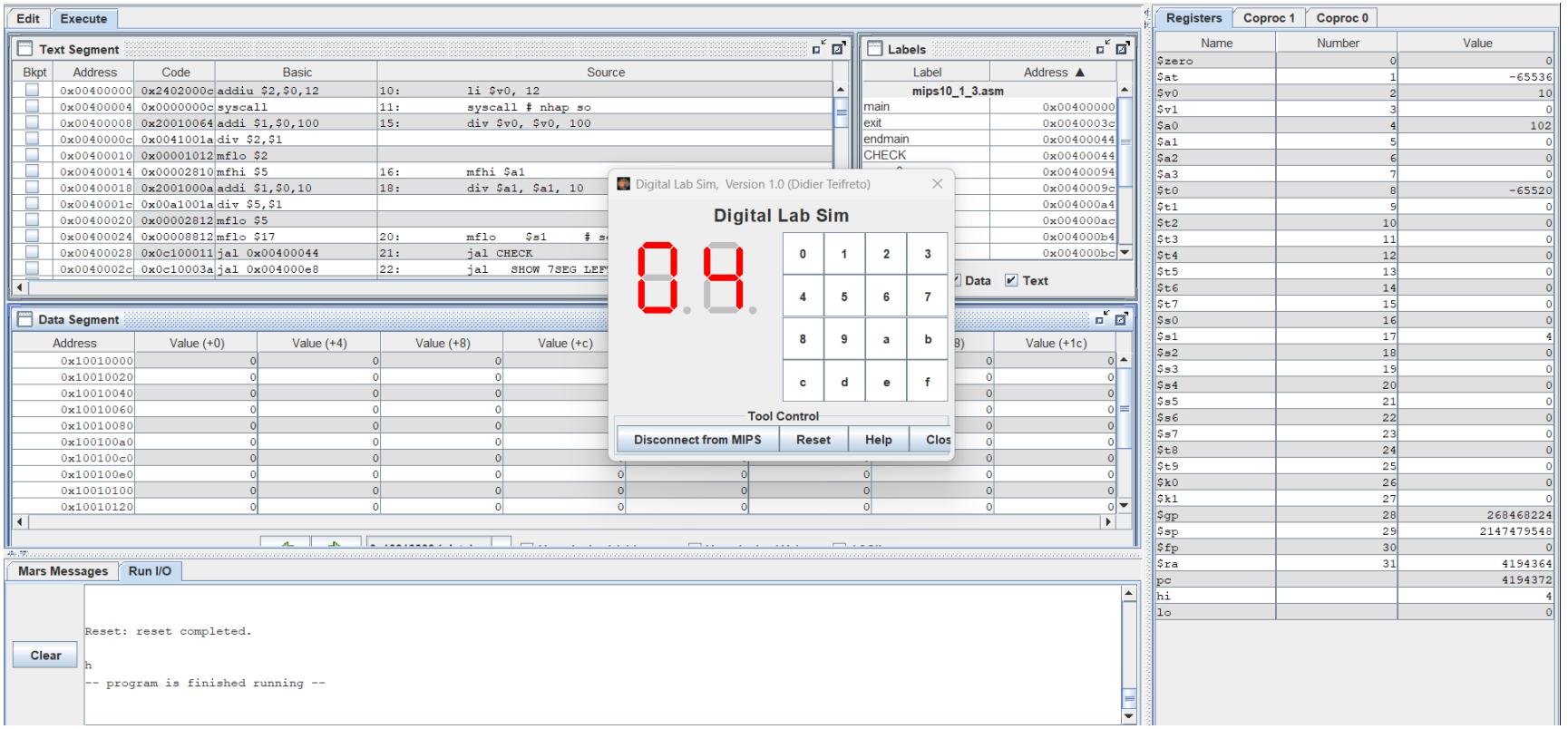
```
        beq $s1, 6, case_6
        beq $s1, 7, case_7
        beq $s1, 8, case_8
        beq $s1, 9, case_9

case_0:
        li $a0, 0x3f
        j end_check
case_1:
        li $a0, 0x06
        j end_check
case_2:
        li $a0, 0x5b
        j end_check
case_3:
        li $a0, 0x4f
        j end_check
case_4:
        li $a0, 0x66
        j end_check
case_5:
        li $a0, 0x6d
        j end_check
case_6:
        li $a0, 0x7d
        j end_check
case_7:
        li $a0, 0x07
        j end_check
case_8:
        li $a0, 0x7f
        j end_check
case_9:
```

```
        li $a0, 0x6f
        j end_check
end_check:
        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
```

- Kết quả:
  Input: h (ASCII code: 104)
  Output: 04



4. **Assignment 4:** *Vẽ ô bàn cờ vua kích thước 8x8, chọn 2 màu bất kỳ (khác màu đen)*
   - Mã chương trình:

```
#.eqv MONITOR_SCREEN 0x10010000
.eqv WHITE 0x00FFFFFF
.data
MONITOR_SCREEN: .word 0x00FFFFFF
.text
    li $s0, 2
    li $t1, 0
    loop1:
    beq $t1, 8, end_loop1
    div $t1, $s0
    mfhi $t2
        loop2:
```

```
    bge $t2, 8, end_loop2
    mul $t3, $t1, 8
    add $t3, $t3, $t2
    sll $t3, $t3, 2
    li $t0, WHITE
    sw $t0, MONITOR_SCREEN($t3)
    addi $t2, $t2, 2
    j loop2
    end_loop2:
  addi $t1, $t1, 1
  j loop1
  end_loop1:
  li $v0, 10
  syscall
```

- Kết quả: