

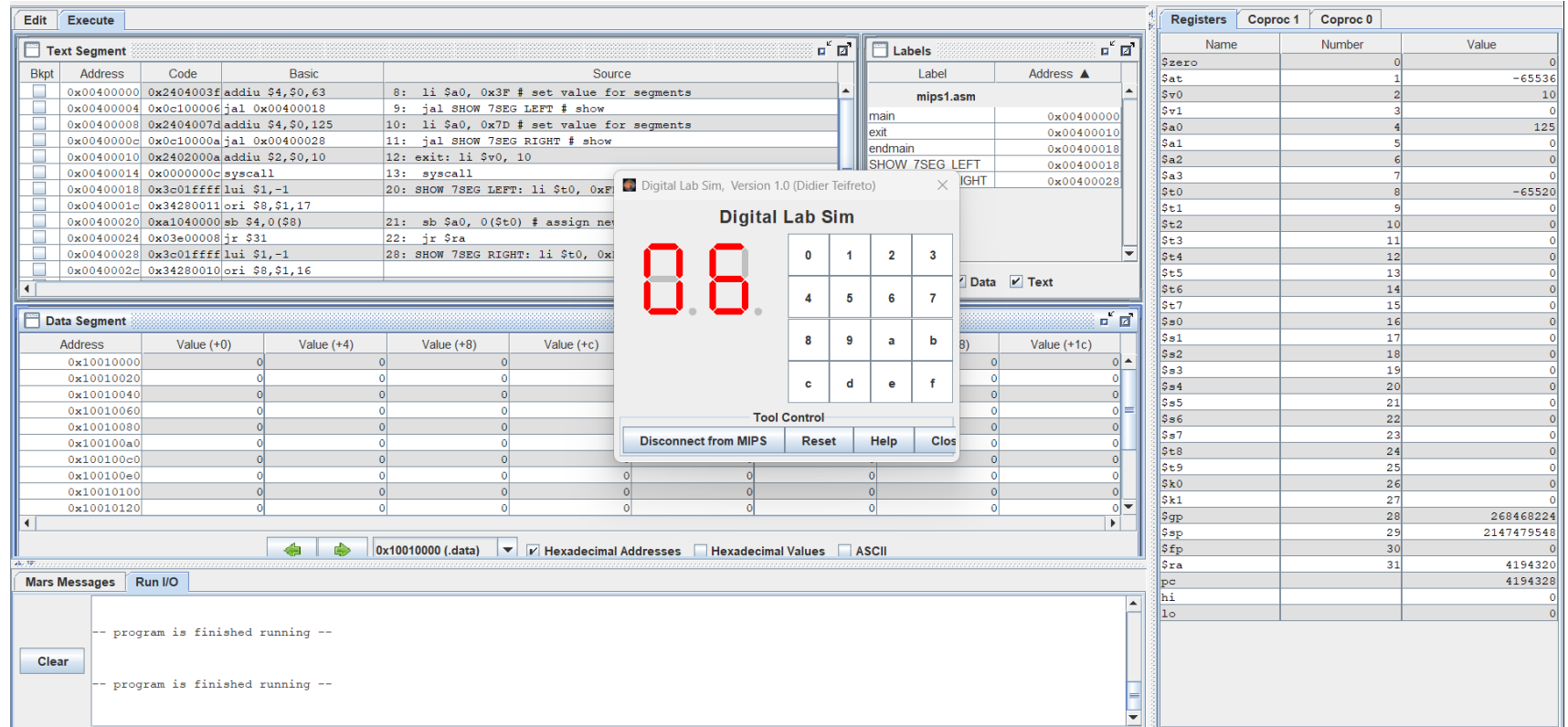
Laboratory Exercise 10

Tools: LED PORT, BITMAP DISPLAY

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

```
1  .eqv SEVENSEG_LEFT 0xFFFF0011 # Địa chỉ của đèn led 7 đoạn trái.
2  # Bit 0 = đoạn a;
3  # Bit 1 = đoạn b; ...
4  # Bit 7 = dấu .
5  .eqv SEVENSEG_RIGHT 0xFFFF0010 # Địa chỉ của đèn led 7 đoạn phải
6  .text
7  main:
8  li $a0, 0x3F # set value for segments
9  jal SHOW_7SEG_LEFT # show
10 li $a0, 0x7D # set value for segments
11 jal SHOW_7SEG_RIGHT # show
12 exit: li $v0, 10
13 syscall
14 endmain:
15 #-----
16 # Function SHOW_7SEG_LEFT : turn on/off the 7seg
17 # param[in] $a0 value to shown
18 # remark $t0 changed
19 #-----
20 SHOW_7SEG_LEFT: li $t0, SEVENSEG_LEFT # assign port's address
21 sb $a0, 0($t0) # assign new value
22 jr $ra
23 #-----
24 # Function SHOW_7SEG_RIGHT : turn on/off the 7seg
25 # param[in] $a0 value to shown
26 # remark $t0 changed
27 #-----
28 SHOW_7SEG_RIGHT: li $t0, SEVENSEG_RIGHT # assign port's address
29 sb $a0, 0($t0) # assign new value
30 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

The screenshot shows the Digital Lab Sim software interface. The main window displays MIPS assembly code in the 'Text Segment' pane. The code includes instructions like `addiu $2,$0,5`, `syscall`, `addi $1,$0,100`, `div $2,$1`, `mflo $2`, `mfhi $5`, `addi $1,$0,10`, `div $5,$1`, `mflo $5`, `mflo $17`, `jal 0x00400044`, and `jal 0x004000e8`. The 'Registers' pane on the right shows the state of various registers, including `$zero`, `$at`, `$v0`, `$v1`, `$a0`, `$a1`, `$a2`, `$a3`, `$t0`, `$t1`, `$t2`, `$t3`, `$t4`, `$t5`, `$t6`, `$t7`, `$s0`, `$s1`, `$s2`, `$s3`, `$s4`, `$s5`, `$s6`, `$s7`, `$t8`, `$t9`, `$k0`, `$k1`, `$gp`, `$sp`, `$fp`, `$ra`, `pc`, `hi`, and `lo`. A 'Digital Lab Sim' window is overlaid in the center, showing a 7-segment display with the number '48' and a numeric keypad. The 'Mars Messages' pane at the bottom shows the output: 'Reset: reset completed.', '12345', and '-- program is finished running --'.

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

```

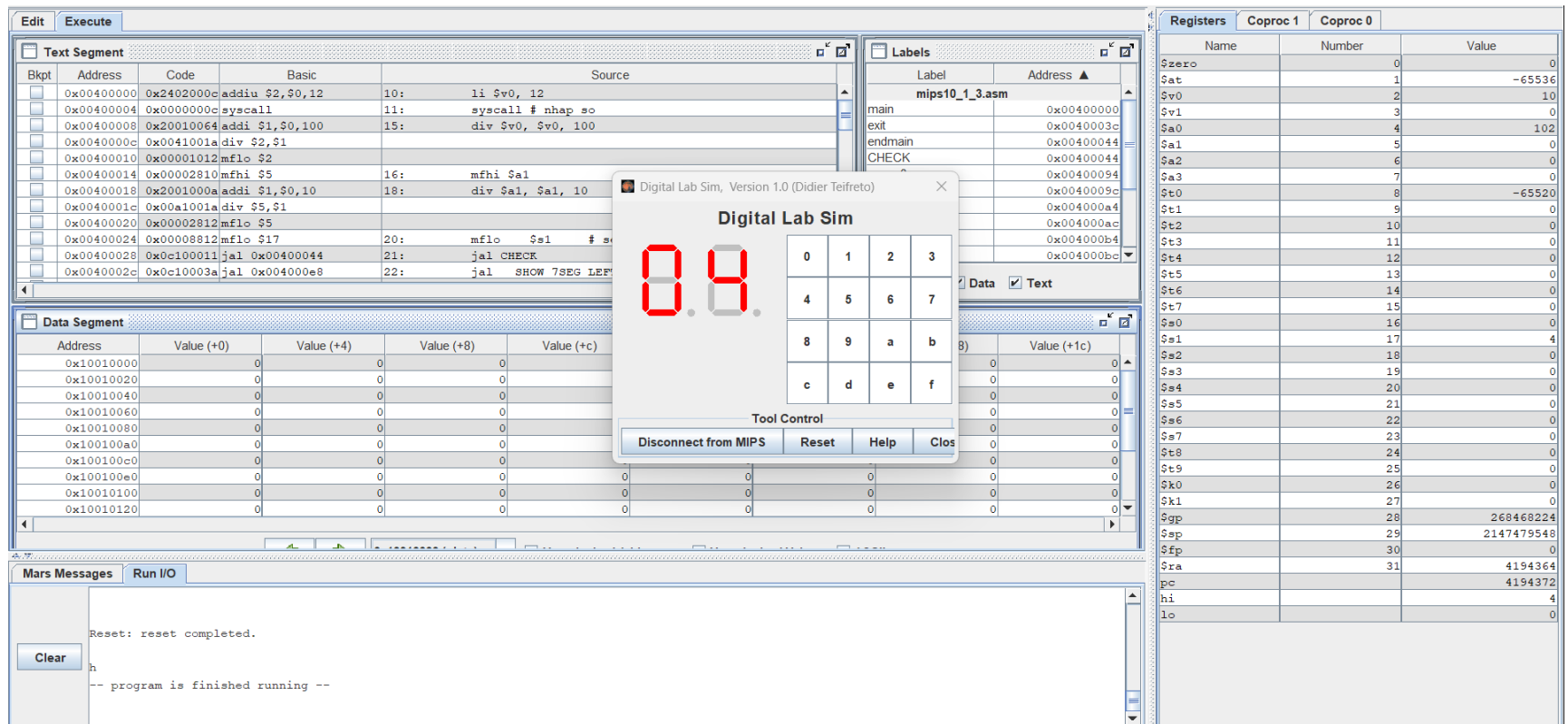
```

#-----
# 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:

The screenshot shows the MIPS simulator interface. The main window displays the Text Segment with instructions and their addresses. A 'Bitmap Display' window is open, showing a 16x16 checkerboard pattern. The Registers window on the right shows the state of the processor registers. The bottom panel shows the 'Mars Messages' window with the message 'Reset: reset completed.' and '-- program is finished running --'.

Label	Address	Code	Basic	Source
	0x00400000	0x24100002	addiu \$t6,\$t0,2	6: li \$s0, 2
	0x00400004	0x24090000	addiu \$t9,\$t0,0	7: li \$t1, 0
	0x00400008	0x20010008	addi \$t1,\$t0,8	9: beq \$t1, 8, end_loop1
	0x0040000c	0x10290011	beq \$t1,\$t9,17	
	0x00400010	0x0130001a	div \$t9,\$t6	10:
	0x00400014	0x00005010	mfhi \$t0	11:
	0x00400018	0x29410008	slli \$t1,\$t0,8	13:
	0x0040001c	0x1020000b	beq \$t1,\$t0,11	
	0x00400020	0x20010008	addi \$t1,\$t0,8	14:
	0x00400024	0x71215802	mul \$t1,\$t9,\$t1	
	0x00400028	0x016a5820	add \$t1,\$t1,\$t0	15:
	0x0040002c	0x000b5880	sll \$t1,\$t1,2	16:

Name	Number	Value
\$zero	0	0
\$at	1	8
\$v0	2	10
\$v1	3	0
\$a0	4	0
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$a4	8	16777215
\$a5	9	8
\$a6	10	9
\$a7	11	252
\$t0	12	0
\$t1	13	0
\$t2	14	0
\$t3	15	0
\$t4	16	2
\$t5	17	0
\$t6	18	0
\$t7	19	0
\$t8	20	0
\$t9	21	0
\$s0	22	0
\$s1	23	0
\$s2	24	0
\$s3	25	0
\$s4	26	0
\$s5	27	0
\$s6	28	268468224
\$s7	29	2147475548
\$s8	30	0
\$s9	31	0
\$pc		4194396
\$hi		0
\$lo		56

Mars Messages: Reset: reset completed. -- program is finished running --