BÀI BÁO CÁO TUẦN 10

Họ và tên: Nguyễn Trọng Khánh Duy

MSSV: 20210284

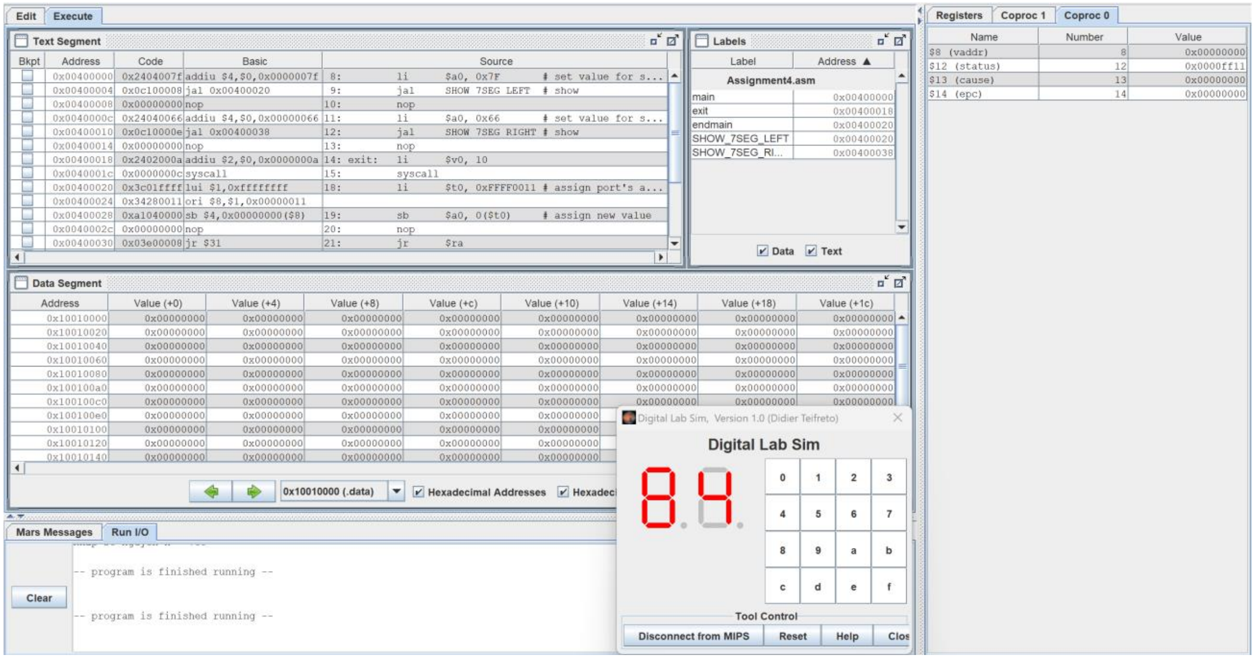
Assignment 1:

CODE:

```
.eqv SEVENSEG_LEFT 0xFFFF0011 # Dia chi cua den led 7 doan trai.
.eqv SEVENSEG_RIGHT 0xFFFF0010  # Dia chi cua den led 7 doan phai
.text
main:
# MSSV 20210284 => 2 so cuoi la 84
# so 8 khi chuyen sang den led 7 doan : 0111 1111 (bit) = 7F (hex)
# so 4 khi chuyen sang den led 7 doan : 0110 0110 (bit) = 66 (hex)
    li  $a0, 0x7F   # set value for segments
    jal     SHOW_7SEG_LEFT  # show
    nop
    li  $a0, 0x66   # set value for segments
    jal     SHOW_7SEG_RIGHT # show
    nop
exit:   li  $v0, 10
    syscall
```

```
endmain:
SHOW_7SEG_LEFT:
    li   $t0, SEVENSEG_LEFT # assign port's address
    sb   $a0, 0($t0)      # assign new value
    nop
    jr   $ra
    nop
SHOW_7SEG_RIGHT:
    li   $t0, SEVENSEG_RIGHT # assign port's address
    sb   $a0, 0($t0)      # assign new value
    nop
    jr   $ra
    nop
```

Kết quả:



Text Segment

Bkpt	Address	Code	Basic	Source
□	0x00400000	0x2404007f	addiu $4,$0,0x0000007f	8: li $a0, 0x7F # set value for s...
□	0x00400004	0x0c100008	jal 0x00400020	9: jal SHOW 7SEG LEFT # show
□	0x00400008	0x00000000	nop	10: nop
□	0x0040000c	0x24040066	addiu $4,$0,0x00000066	11: li $a0, 0x66 # set value for s...
□	0x00400010	0x0c10000e	jal 0x00400038	12: jal SHOW 7SEG RIGHT # show
□	0x00400014	0x00000000	nop	13: nop
□	0x00400018	0x2402000a	addiu $2,$0,0x0000000a	14: exit: li $v0, 10
□	0x0040001c	0x0000000c	syscall	15: syscall
□	0x00400020	0x3c01ffff	lui $1,0xffffffff	18: li $t0, 0xFFFF0011 # assign port's a...
□	0x00400024	0x34280011	ori $8,$1,0x00000011	
□	0x00400028	0xa1040000	sb $4,0x00000000($8)	19: sb $a0, 0($t0) # assign new value
□	0x0040002c	0x00000000	nop	20: nop
□	0x00400030	0x03e00008	jr $31	21: jr $ra

Labels

Label	Address ▲
Assignment4.asm	
main	0x00400000
exit	0x00400018
endmain	0x00400020
SHOW_7SEG_LEFT	0x00400020
SHOW_7SEG_RI...	0x00400038

Name	Number	Value
$8 (vaddr)	8	0x00000000
$12 (status)	12	0x0000ff11
$13 (cause)	13	0x00000000
$14 (epc)	14	0x00000000

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000			
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000			
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000			
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000			

```
-- program is finished running --


-- program is finished running --
```

Assignment 2

CODE:

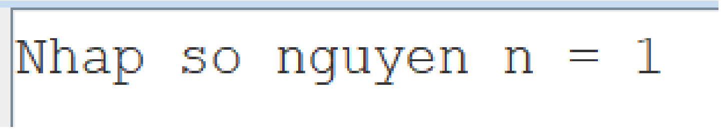
```
.eqv SEVENSEG_LEFT 0xFFFF0011 # Dia chi cua den led 7 doan trai.
.eqv SEVENSEG_RIGHT 0xFFFF0010  # Dia chi cua den led 7 doan phai
.data
    nhapso: .asciiz "Nhap so nguyen n = "

.text
main:
    li  $v0,4
    la  $a0,nhapso
    syscall
    li  $v0, 5
    syscall
    add $s1, $v0, $zero # n = $s1
    div $s2, $s1, 10
    mfhi    $t1     # so hang don vi
    div $s3, $s2, 10
    mfhi    $t2     # so hang chuc
    addi    $s4, $zero, -1  # khoi tao check = -1
    add $t3, $zero, $t2 # $t3 = so hang chuc
Check_so:
So_0:   bne $t3, 0, So_1
```

```
    li  $a0, 0x3F
    j   in_so
So_1:   bne $t3, 1, So_2
    li  $a0, 0x06
    j   in_so
So_2:   bne $t3, 2, So_3
    li  $a0, 0x5B
    j   in_so
So_3:   bne $t3, 3, So_4
    li  $a0, 0x4F
    j   in_so
So_4:   bne $t3, 4, So_5
    li  $a0, 0x66
    j   in_so
So_5:   bne $t3, 5, So_6
    li  $a0, 0x6D
    j   in_so
So_6:   bne $t3, 6, So_7
    li  $a0, 0x7D
    j   in_so
So_7:   bne $t3, 7, So_8
    li  $a0, 0x07
    j   in_so
So_8:   bne $t3, 8, So_9
```

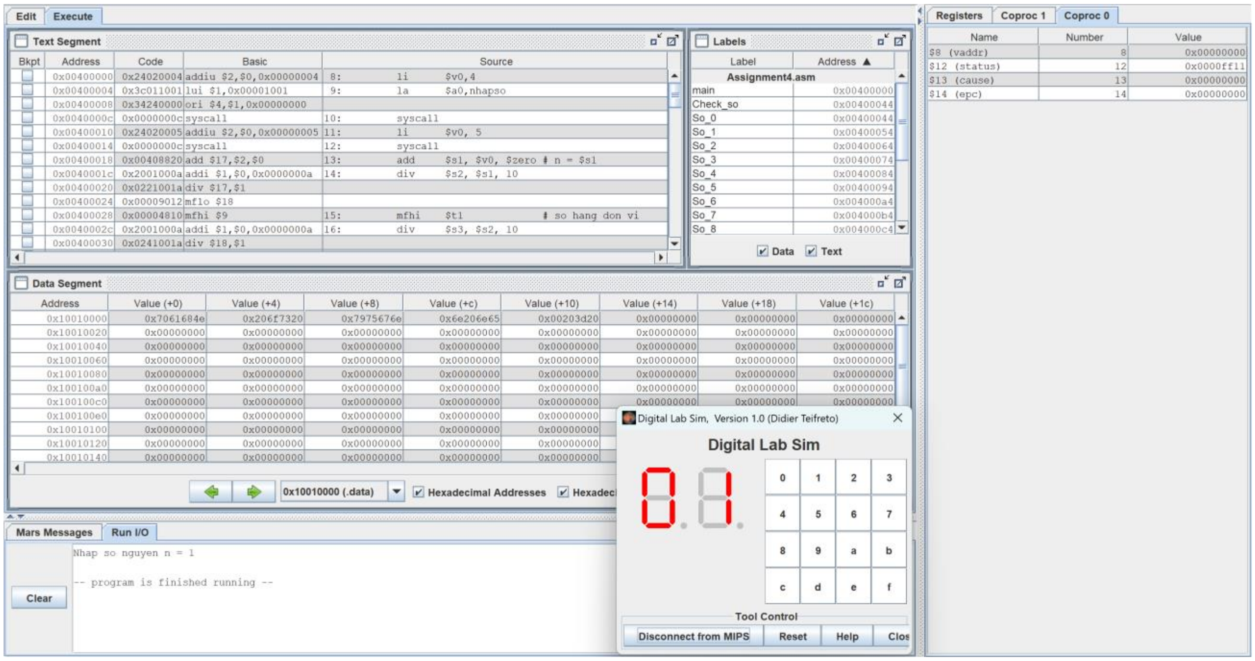
```
    li  $a0, 0x7F
    j   in_so
So_9:   bne $t3, 9, exit
    li  $a0, 0x6F
in_so:
    beq $s4, $zero, in_so_don_vi# if check = -1 in
so hang chuc
                    # if check = 0 in so hang don
vi
in_so_chuc:
    jal     SHOW_7SEG_LEFT  # show
    nop
    addi    $s4, $s4, 1
    add $t3, $zero, $t1 # $t3 = so hang don vi
    j Check_so

in_so_don_vi:
    jal     SHOW_7SEG_RIGHT # show
    nop
exit:   li  $v0, 10
    syscall
endmain:

SHOW_7SEG_LEFT:
    li  $t0, SEVENSEG_LEFT # assign port's address
    sb   $a0, 0($t0)      # assign new value
    nop
    jr   $ra
    nop
SHOW_7SEG_RIGHT:
    li   $t0, SEVENSEG_RIGHT # assign port's address
    sb   $a0, 0($t0)      # assign new value
    nop
    jr   $ra
    nop
```

Kết quả:

TH1: Khi nhập số có 1 chữ số:



```
Nhap so nguyen n = 1
```



Text Segment

Bkpt	Address	Code	Basic	Source
□	0x00400000	0x24020004	addiu $2,$0,0x00000004	8: li $v0,4
□	0x00400004	0x3c011001	lui $1,0x00001001	9: la $a0,nhapso
□	0x00400008	0x34240000	ori $4,$1,0x00000000	
□	0x0040000c	0x0000000c	syscall	10: syscall
□	0x00400010	0x24020005	addiu $2,$0,0x00000005	11: li $v0, 5
□	0x00400014	0x0000000c	syscall	12: syscall
□	0x00400018	0x00408820	add $17,$2,$0	13: add $s1, $v0, $zero # n = $s1
□	0x0040001c	0x2001000a	addi $1,$0,0x0000000a	14: div $s2, $s1, 10
□	0x00400020	0x0221001a	div $17,$1	
□	0x00400024	0x00009012	mflo $18	
□	0x00400028	0x00004810	mfhi $9	15: mfhi $t1 # so hang don vi
□	0x0040002c	0x2001000a	addi $1,$0,0x0000000a	16: div $s3, $s2, 10
□	0x00400030	0x0241001a	div $18,$1	

Labels

Label	Address ▲
Assignment4.asm	
main	0x00400000
Check_so	0x00400044
So_0	0x00400044
So_1	0x00400054
So_2	0x00400064
So_3	0x00400074
So_4	0x00400084
So_5	0x00400094
So_6	0x004000a4
So_7	0x004000b4
So_8	0x004000c4

☑ Data ☑ Text

Name	Number	Value
$8 (vaddr)	8	0x00000000
$12 (status)	12	0x0000ff11
$13 (cause)	13	0x00000000
$14 (epc)	14	0x00000000

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x7061684e	0x206f7320	0x7975676e	0x6e206e65	0x00203d20	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000			
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000			
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000			
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000			

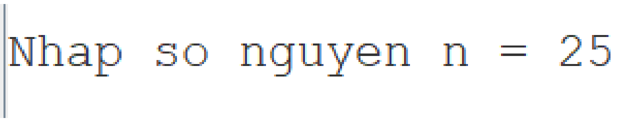
0x10010000 (.data) ☑ Hexadecimal Addresses ☑ Hexadeci

```
Nhap so nguyen n = 1

-- program is finished running --
```

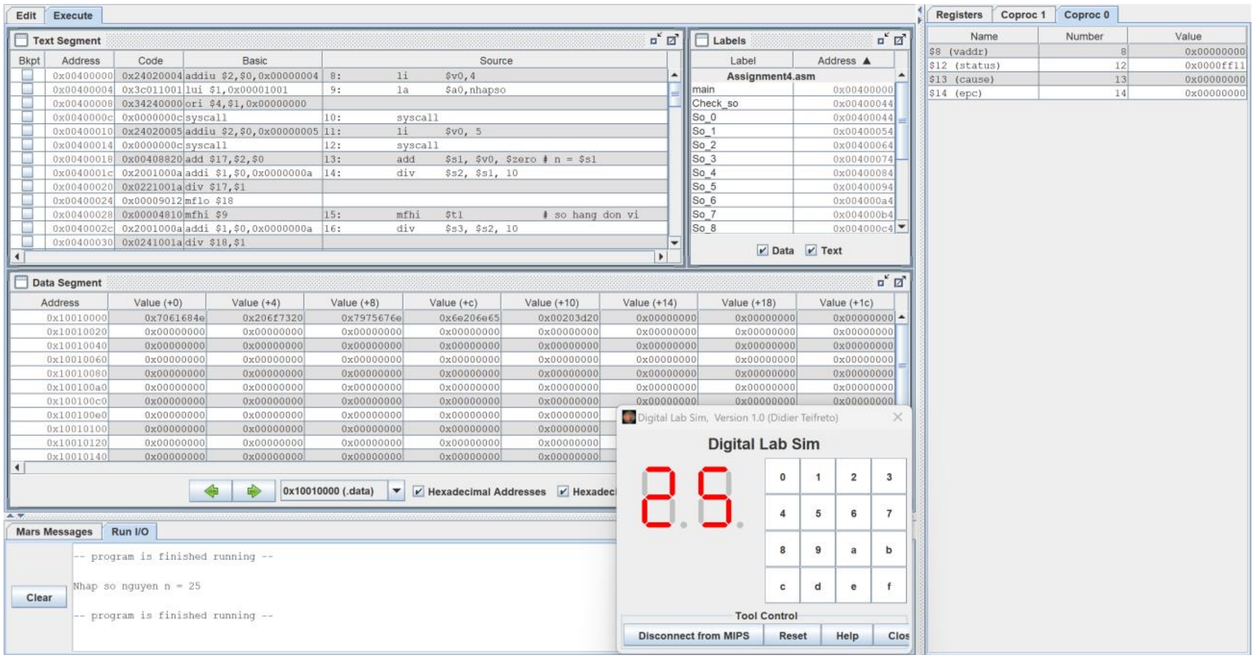
Digital Lab Sim, Version 1.0 (Didier Teifreto)

Digital Lab Sim

TH2: Khi nhập một số có 2 chữ số:



```
Nhap so nguyen n = 25
```



Bkpt	Address	Code	Basic	Source
□	0x00400000	0x24020004	addiu $2,$0,0x00000004	8: li $v0,4
□	0x00400004	0x3c011001	lui $1,0x00001001	9: la $a0,nhapso
□	0x00400008	0x34240000	ori $4,$1,0x00000000	
□	0x0040000c	0x0000000c	syscall	10: syscall
□	0x00400010	0x24020005	addiu $2,$0,0x00000005	11: li $v0, 5
□	0x00400014	0x0000000c	syscall	12: syscall
□	0x00400018	0x00408820	add $17,$2,$0	13: add $s1, $v0, $zero # n = $s1
□	0x0040001c	0x2001000a	addi $1,$0,0x0000000a	14: div $s2, $s1, 10
□	0x00400020	0x0221001a	div $17,$1	
□	0x00400024	0x00009012	mflo $18	
□	0x00400028	0x00004810	mfhi $9	15: mfhi $t1 # so hang don vi
□	0x0040002c	0x2001000a	addi $1,$0,0x0000000a	16: div $s3, $s2, 10
□	0x00400030	0x0241001a	div $18,$1	

Label	Address ▲
Assignment4.asm	
main	0x00400000
Check_so	0x00400044
So_0	0x00400044
So_1	0x00400054
So_2	0x00400064
So_3	0x00400074
So_4	0x00400084
So_5	0x00400094
So_6	0x004000a4
So_7	0x004000b4
So_8	0x004000c4

☑ Data ☑ Text

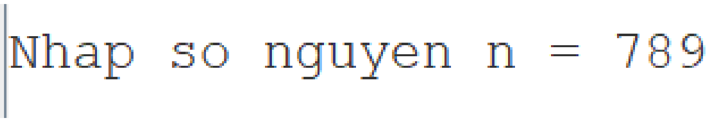
Name	Number	Value
$8 (vaddr)	8	0x00000000
$12 (status)	12	0x0000ff11
$13 (cause)	13	0x00000000
$14 (epc)	14	0x00000000

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x7061684e	0x206f7320	0x7975676e	0x6e206e65	0x00203d20	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000			
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000			
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000			
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000			

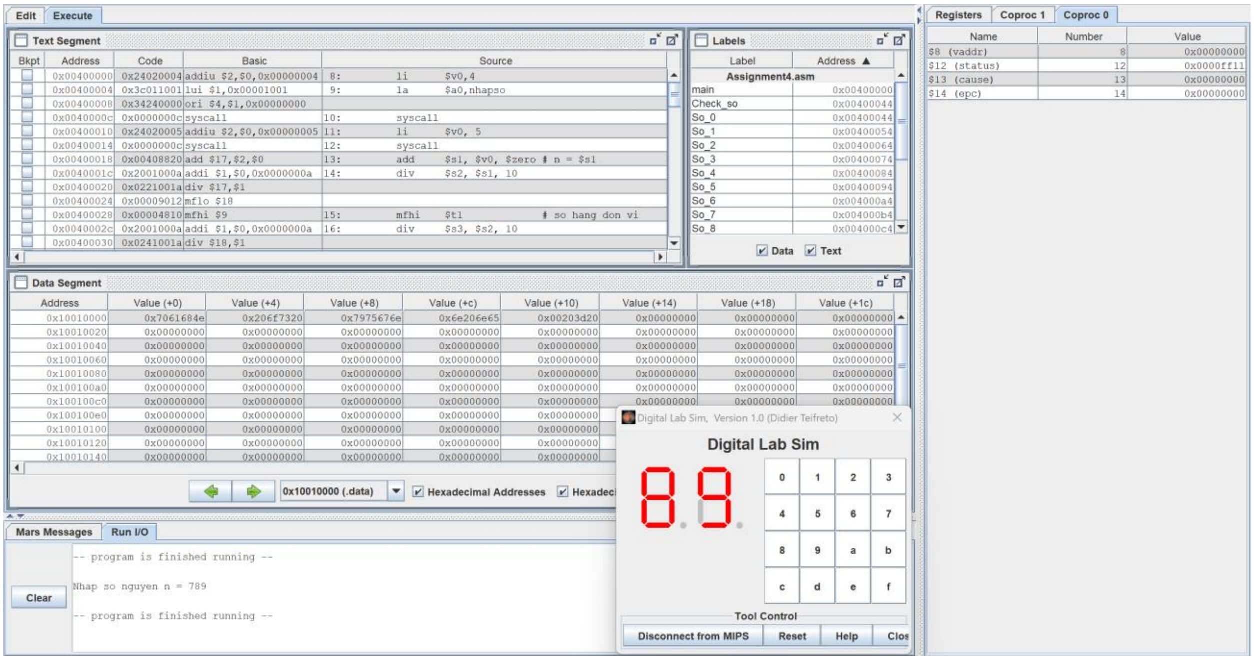
0x10010000 (.data) ☑ Hexadecimal Addresses ☑ Hexadeci

```
-- program is finished running --

Nhap so nguyen n = 25

-- program is finished running --
```

TH3: Khi nhập một số có 3 chữ số:



```
Nhap so nguyen n = 789
```



Bkpt	Address	Code	Basic	Source
□	0x00400000	0x24020004	addiu $2,$0,0x00000004	8: li $v0,4
□	0x00400004	0x3c011001	lui $1,0x00001001	9: la $a0,nhapso
□	0x00400008	0x34240000	ori $4,$1,0x00000000	
□	0x0040000c	0x0000000c	syscall	10: syscall
□	0x00400010	0x24020005	addiu $2,$0,0x00000005	11: li $v0, 5
□	0x00400014	0x0000000c	syscall	12: syscall
□	0x00400018	0x00408820	add $17,$2,$0	13: add $s1, $v0, $zero # n = $s1
□	0x0040001c	0x2001000a	addi $1,$0,0x0000000a	14: div $s2, $s1, 10
□	0x00400020	0x0221001a	div $17,$1	
□	0x00400024	0x00009012	mflo $18	
□	0x00400028	0x00004810	mfhi $9	15: mfhi $t1 # so hang don vi
□	0x0040002c	0x2001000a	addi $1,$0,0x0000000a	16: div $s3, $s2, 10
□	0x00400030	0x0241001a	div $18,$1	

Label	Address ▲
Assignment4.asm	
main	0x00400000
Check_so	0x00400044
So_0	0x00400044
So_1	0x00400054
So_2	0x00400064
So_3	0x00400074
So_4	0x00400084
So_5	0x00400094
So_6	0x004000a4
So_7	0x004000b4
So_8	0x004000c4

☑ Data ☑ Text

Name	Number	Value
$8 (vaddr)	8	0x00000000
$12 (status)	12	0x0000ff11
$13 (cause)	13	0x00000000
$14 (epc)	14	0x00000000

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x7061684e	0x206f7320	0x7975676e	0x6e206e65	0x00203d20	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000			
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000			
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000			
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000			

0x10010000 (.data) ☑ Hexadecimal Addresses ☑ Hexadeci

```
-- program is finished running --

Nhap so nguyen n = 789

-- program is finished running --
```

Assignment 3

CODE:

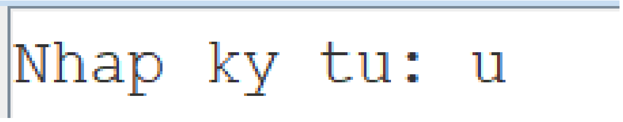
```
.eqv SEVENSEG_LEFT 0xFFFF0011 # Dia chi cua den led 7 doan trai.
.eqv SEVENSEG_RIGHT 0xFFFF0010  # Dia chi cua den led 7 doan phai
.data
    nhapkytu:   .asciiz "Nhap ky tu: "

.text
main:
    li  $v0,4
    la  $a0,nhapkytu
    syscall
    li  $v0, 12
    syscall
    add $s1, $v0, $zero # n = $s1
    div $s2, $s1, 10
    mfhi    $t1     # so hang don vi
    div $s3, $s2, 10
    mfhi    $t2     # so hang chuc
    addi    $s4, $zero, -1  # khoi tao check = -1
    add $t3, $zero, $t2 # $t3 = so hang chuc
Check_so:
So_0:   bne $t3, 0, So_1
```

```
    li  $a0, 0x3F
    j   in_so
So_1:   bne $t3, 1, So_2
    li  $a0, 0x06
    j   in_so
So_2:   bne $t3, 2, So_3
    li  $a0, 0x5B
    j   in_so
So_3:   bne $t3, 3, So_4
    li  $a0, 0x4F
    j   in_so
So_4:   bne $t3, 4, So_5
    li  $a0, 0x66
    j   in_so
So_5:   bne $t3, 5, So_6
    li  $a0, 0x6D
    j   in_so
So_6:   bne $t3, 6, So_7
    li  $a0, 0x7D
    j   in_so
So_7:   bne $t3, 7, So_8
    li  $a0, 0x07
    j   in_so
So_8:   bne $t3, 8, So_9
```

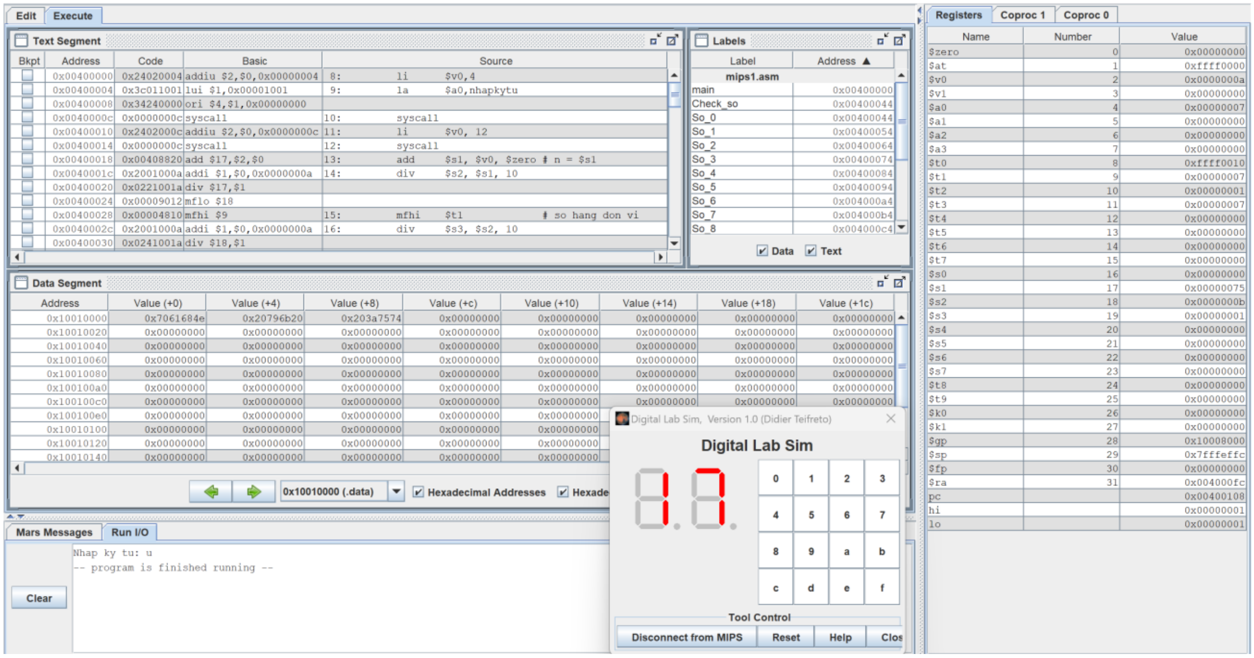
```
    li  $a0, 0x7F
    j   in_so
So_9:   bne $t3, 9, exit
    li  $a0, 0x6F
in_so:
    beq $s4, $zero, in_so_don_vi# if check = -1 in
so hang chuc
                    # if check = 0 in so hang don
vi
in_so_chuc:
    jal     SHOW_7SEG_LEFT  # show
    nop
    addi    $s4, $s4, 1
    add $t3, $zero, $t1 # $t3 = so hang don vi
    j Check_so

in_so_don_vi:
    jal     SHOW_7SEG_RIGHT # show
    nop
exit:   li  $v0, 10
    syscall
endmain:

SHOW_7SEG_LEFT:
    li  $t0, SEVENSEG_LEFT # assign port's address
    sb   $a0, 0($t0)       # assign new value
    nop
    jr   $ra
    nop
SHOW_7SEG_RIGHT:
    li   $t0, SEVENSEG_RIGHT # assign port's address
    sb   $a0, 0($t0)       # assign new value
    nop
    jr   $ra
    nop
```

Kết quả:

Nhập ký tự thường:



```
Nhap ky tu: u
```



Text Segment

Bkpt	Address	Code	Basic	Source
□	0x00400000	0x24020004	addiu $2,$0,0x00000004	8: li $v0,4
□	0x00400004	0x3c011001	lui $1,0x00001001	9: la $a0,nhapkytu
□	0x00400008	0x34240000	ori $4,$1,0x00000000	
□	0x0040000c	0x0000000c	syscall	10: syscall
□	0x00400010	0x2402000c	addiu $2,$0,0x0000000c	11: li $v0, 12
□	0x00400014	0x0000000c	syscall	12: syscall
□	0x00400018	0x00408820	add $17,$2,$0	13: add $s1, $v0, $zero # n = $s1
□	0x0040001c	0x2001000a	addi $1,$0,0x0000000a	14: div $s2, $s1, 10
□	0x00400020	0x0221001a	div $17,$1	
□	0x00400024	0x00009012	mflo $18	
□	0x00400028	0x00004810	mfhi $9	15: mfhi $t1 # so hang don vi
□	0x0040002c	0x2001000a	addi $1,$0,0x0000000a	16: div $s3, $s2, 10
□	0x00400030	0x0241001a	div $18,$1	

Labels

Label	Address ▲
mips1.asm	
main	0x00400000
Check_so	0x00400044
So_0	0x00400044
So_1	0x00400054
So_2	0x00400064
So_3	0x00400074
So_4	0x00400084
So_5	0x00400094
So_6	0x004000a4
So_7	0x004000b4
So_8	0x004000c4

☑ Data ☑ Text

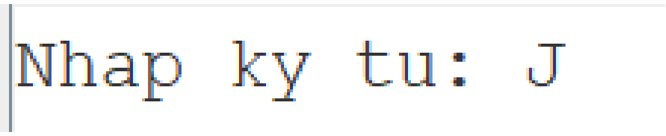
Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x7061684e	0x20796b20	0x203a7574	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000			
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000			
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000			
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000			

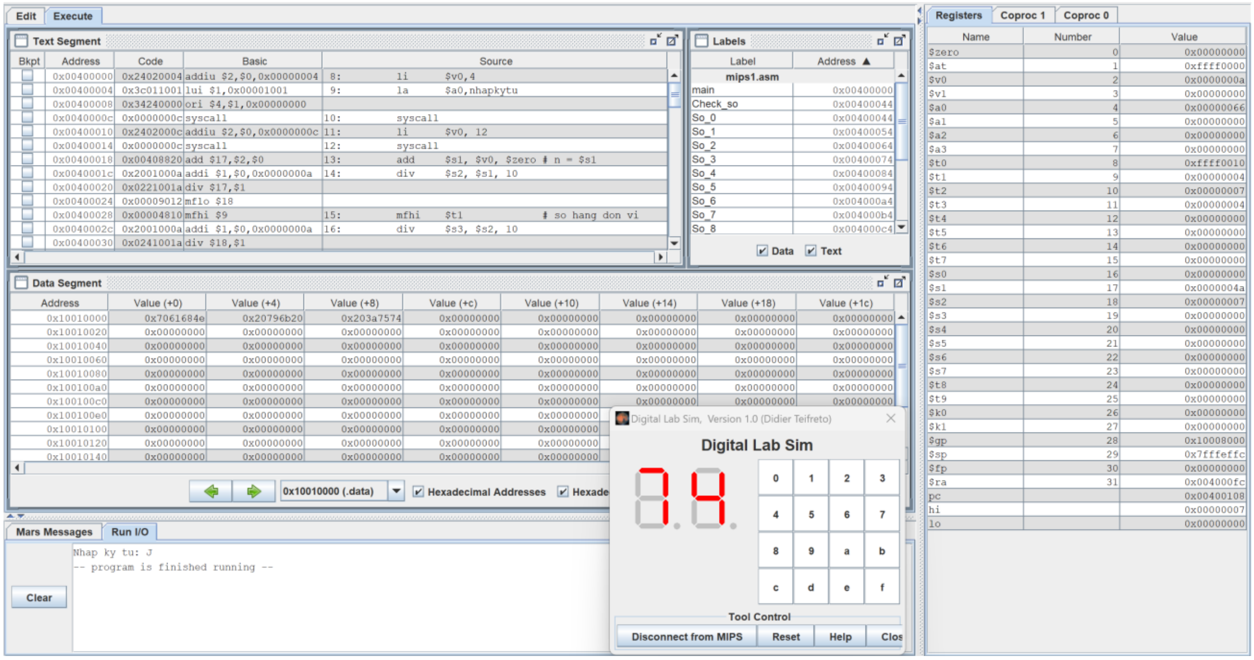
0x10010000 (.data) ☑ Hexadecimal Addresses ☑ Hexade

```
Nhap ky tu: u
-- program is finished running --
```

Name	Number	Value
$zero	0	0x00000000
$at	1	0xffff0000
$v0	2	0x0000000a
$v1	3	0x00000000
$a0	4	0x00000007
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0xffff0010
$t1	9	0x00000007
$t2	10	0x00000001
$t3	11	0x00000007
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x00000075
$s2	18	0x0000000b
$s3	19	0x00000001
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x004000fc
pc		0x00400108
hi		0x00000001
lo		0x00000001

Nhập ký tự in hoa: 

```
Nhap ky tu: J
```



Bkpt	Address	Code	Basic	Source
□	0x00400000	0x24020004	addiu $2,$0,0x00000004	8: li $v0,4
□	0x00400004	0x3c011001	lui $1,0x00001001	9: la $a0,nhapkytu
□	0x00400008	0x34240000	ori $4,$1,0x00000000	
□	0x0040000c	0x0000000c	syscall	10: syscall
□	0x00400010	0x2402000c	addiu $2,$0,0x0000000c	11: li $v0, 12
□	0x00400014	0x0000000c	syscall	12: syscall
□	0x00400018	0x00408820	add $17,$2,$0	13: add $s1, $v0, $zero # n = $s1
□	0x0040001c	0x2001000a	addi $1,$0,0x0000000a	14: div $s2, $s1, 10
□	0x00400020	0x0221001a	div $17,$1	
□	0x00400024	0x00009012	mflo $18	
□	0x00400028	0x00004810	mfhi $9	15: mfhi $t1 # so hang don vi
□	0x0040002c	0x2001000a	addi $1,$0,0x0000000a	16: div $s3, $s2, 10
□	0x00400030	0x0241001a	div $18,$1	

Label	Address ▲
mips1.asm	
main	0x00400000
Check_so	0x00400044
So_0	0x00400044
So_1	0x00400054
So_2	0x00400064
So_3	0x00400074
So_4	0x00400084
So_5	0x00400094
So_6	0x004000a4
So_7	0x004000b4
So_8	0x004000c4

☑ Data ☑ Text

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x7061684e	0x20796b20	0x203a7574	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000			
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000			
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000			
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000			

0x10010000 (.data) ☑ Hexadecimal Addresses ☑ Hexade

```
Nhap ky tu: J
-- program is finished running --
```

Name	Number	Value
$zero	0	0x00000000
$at	1	0xffff0000
$v0	2	0x0000000a
$v1	3	0x00000000
$a0	4	0x00000066
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0xffff0010
$t1	9	0x00000004
$t2	10	0x00000007
$t3	11	0x00000004
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x0000004a
$s2	18	0x00000007
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x004000fc
pc		0x00400108
hi		0x00000007
lo		0x00000000

Assignment 4

CODE:

```
.eqv MONITOR_SCREEN 0x10010000 #Dia chi bat dau cua bo nho man hinh
.eqv RED 0x00FF0000 #Cac gia tri mau thuong su dung
.eqv GREEN 0x0000FF00
.eqv BLUE 0x000000FF
.eqv WHITE 0x00FFFFFF
.eqv YELLOW 0x00FFFF00
.text
    li  $k0, MONITOR_SCREEN     #Nap dia chi bat dau cua man hinh
    li  $t2, 2          # khoi tao 2
    li  $t4, 8          # khoi tao 8
    li  $t1, -1         # i = -1
    li  $t4, 0          # j = 0
    li  $s1, 4          # $s1 = 4
    add $k0, $k0, -4        # $k0 = $k0 - 4
FOR:    addi    $t1, $t1, 1     # i = 0, i ++
    addi    $t4, $t4, 1     # j++
    beq $t1, 72, EXIT       # i = 72 stop
    add $k0, $k0, $s1       # $k0 += 4
    div $t1, $t2        # i / 2
    mfhi    $t3         # $t3 = i % 2
    bne $t4, 8, continue    # j = 8 => i++
    li  $t4, 0          # j = 0
    addi    $t1, $t1, 1     # i++
continue:
    beq $t3, $zero, doi_mau
    li  $t0, RED
    sw  $t0, 0($k0)
    nop
    j   FOR

doi_mau:
    li  $t0, BLUE
    sw  $t0, 0($k0)
    nop
    j FOR

EXIT:   li  $v0, 10
    syscall
```

Giải thích code:

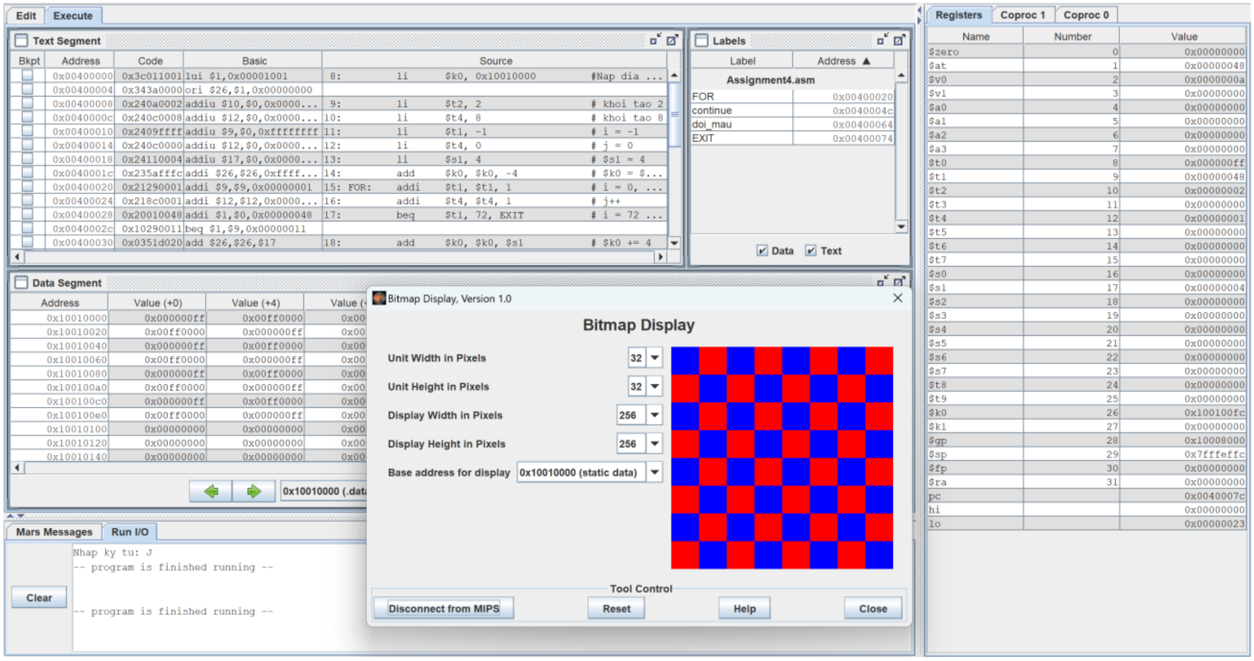
- Dùng một biến i chạy từ 0 – 71 thì dừng

- Dùng một biến j chạy từ 1 – 8

- Khi i /2 dư 1 thì sẽ in màu đỏ, i / 2 dư 0 thì in màu xanh

- Vì bảng là 8x8 nên mỗi lần j = 8 sẽ tăng i thêm 1 => i sẽ tăng 2 đơn vị khi xuống dòng => khi xuống dòng i sẽ giữ nguyên màu. Vì có 8 dòng nên sẽ tăng i lên 8 đơn vị nên phải đặt cho i dừng khi i = 72

Kết quả:



Text Segment

Bkpt	Address	Code	Basic	Source
□	0x00400000	0x3c011001	lui $1,0x00001001	8: li $k0, 0x10010000 #Nap dia ...
□	0x00400004	0x343a0000	ori $26,$1,0x00000000	
□	0x00400008	0x240a0002	addiu $10,$0,0x0000...	9: li $t2, 2 # khoi tao 2
□	0x0040000c	0x240c0008	addiu $12,$0,0x0000...	10: li $t4, 8 # khoi tao 8
□	0x00400010	0x2409ffff	addiu $9,$0,0xffffffff	11: li $t1, -1 # i = -1
□	0x00400014	0x240c0000	addiu $12,$0,0x0000...	12: li $t4, 0 # j = 0
□	0x00400018	0x24110004	addiu $17,$0,0x0000...	13: li $s1, 4 # $s1 = 4
□	0x0040001c	0x235afffc	addi $26,$26,0xffff...	14: add $k0, $k0, -4 # $k0 = $...
□	0x00400020	0x21290001	addi $9,$9,0x00000001	15: FOR: addi $t1, $t1, 1 # i = 0, ...
□	0x00400024	0x218c0001	addi $12,$12,0x0000...	16: addi $t4, $t4, 1 # j++
□	0x00400028	0x20010048	addi $1,$0,0x00000048	17: beq $t1, 72, EXIT # i = 72 ...
□	0x0040002c	0x10290011	beq $1,$9,0x00000011	
□	0x00400030	0x0351d020	add $26,$26,$17	18: add $k0, $k0, $s1 # $k0 += 4

Labels

Label	Address ▲
Assignment4.asm	
FOR	0x00400020
continue	0x0040004c
doi_mau	0x00400064
EXIT	0x00400074

☑ Data ☑ Text

Data Segment

Address	Value (+0)	Value (+4)	Value (+
0x10010000	0x000000ff	0x00ff0000	0x00
0x10010020	0x00ff0000	0x000000ff	0x00
0x10010040	0x000000ff	0x00ff0000	0x00
0x10010060	0x00ff0000	0x000000ff	0x00
0x10010080	0x000000ff	0x00ff0000	0x00
0x100100a0	0x00ff0000	0x000000ff	0x00
0x100100c0	0x000000ff	0x00ff0000	0x00
0x100100e0	0x00ff0000	0x000000ff	0x00
0x10010100	0x00000000	0x00000000	0x00
0x10010120	0x00000000	0x00000000	0x00
0x10010140	0x00000000	0x00000000	0x00

0x10010000 (.data

Bitmap Display settings:

Unit Width in Pixels	32
Unit Height in Pixels	32
Display Width in Pixels	256
Display Height in Pixels	256
Base address for display	0x10010000 (static data)

Run I/O:

```
Nhap ky tu: J
-- program is finished running --


-- program is finished running --
```

Registers

Name	Number	Value
$zero	0	0x00000000
$at	1	0x00000048
$v0	2	0x0000000a
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x000000ff
$t1	9	0x00000048
$t2	10	0x00000002
$t3	11	0x00000000
$t4	12	0x00000001
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x00000004
$s2	18	0x00000000
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x100100fc
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x0040007c
hi		0x00000000
lo		0x00000023

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
```

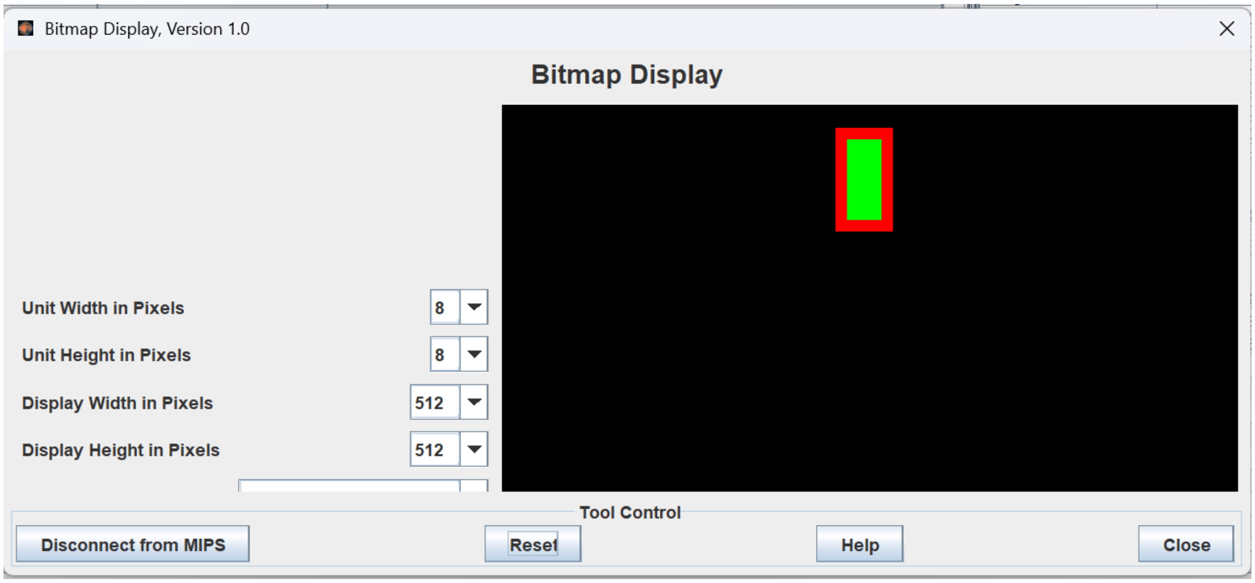
```
    la  $a0, error1
    syscall
    j   NhapX2
Error2: li  $v0, 4
    la  $a0, error2
    syscall
    j   NhapY2
Tsugi:
    slt $t0, $s0, $s2
    slt $t1, $s1, $s3

    beq $t0, 0, Case3
    beq $t1, 0, Case2
Case1:  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

Case2:  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
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
```

```
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:



Unit Width in Pixels 8

Unit Height in Pixels 8

Display Width in Pixels 512

Display Height in Pixels 512