

Thực hành kiến trúc máy tính

Week 10

Bùi Quang Hưng – 20225849

Assignment 1:

- Code:

```
.eqv SEVENSEG_LEFT 0xFFFF0011      # Địa chỉ của đèn led 7 đoạn trái.

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

.eqv SEVENSEG_RIGHT 0xFFFF0010     # Địa chỉ của đèn led 7 đoạn phải

.text

main:

    li $a0, 0x66                    # set number 5 for segments
    jal SHOW_7SEG_LEFT              # show
    nop

    li $a0, 0x6f                    # set number 9 for segments
    jal SHOW_7SEG_RIGHT            # show
    nop

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

nop

jr $ra

nop

#-----

# 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

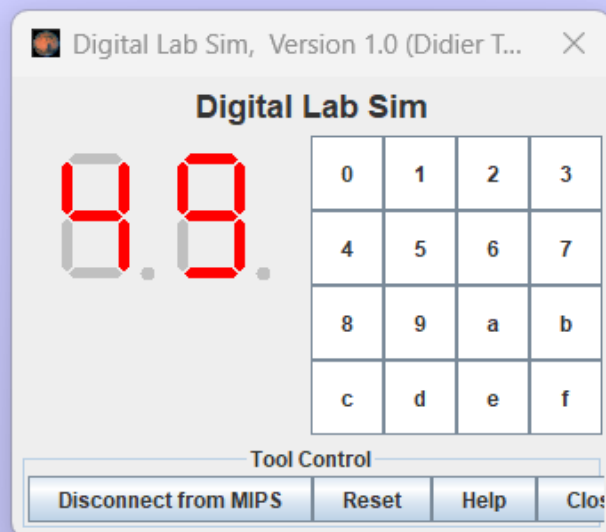
    nop

    jr $ra

    nop

```

- **Nhận xét và kết quả:** Mở Digital Lab Sim trong Tool, kết nối nó với MIPS, chạy và hiển thị kết quả. Bùi Quang Hưng mã số sinh viên 20225849



➔ Kết quả đúng với lý thuyết

Assignment 2:

- Code:

```
.eqv SEVENSEG_LEFT 0xFFFF0011      # Địa chỉ của đèn led 7 đoạn trái.

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

.eqv SEVENSEG_RIGHT 0xFFFF0010     # Địa chỉ của đèn led 7 đoạn phải

.data

message: .asciiz "Nhập vào một số nguyên: "

.text

main:

    li $v0, 4

    la $a0, message

    syscall

    li $v0, 5

    syscall

    move $s0, $v0

    li $t2, 10

    div $s0, $t2

    mfhi $t1

    # So sánh giá trị của t1 với các giá trị từ 0 đến 9 và hiển thị tương ứng

    beq $t1, 0, R0

    beq $t1, 1, R1

    beq $t1, 2, R2

    beq $t1, 3, R3
```

beq \$t1, 4, R4

beq \$t1, 5, R5

beq \$t1, 6, R6

beq \$t1, 7, R7

beq \$t1, 8, R8

beq \$t1, 9, R9

R0:

li \$a0, 0x3F

jal SHOW_7SEG_RIGHT

j sothuhai

R1:

li \$a0, 0x06

jal SHOW_7SEG_RIGHT

j sothuhai

R2:

li \$a0, 0x5B

jal SHOW_7SEG_RIGHT

j sothuhai

R3:

li \$a0, 0x4F

jal SHOW_7SEG_RIGHT

j sothuhai

R4:

li \$a0, 0x66

jal SHOW_7SEG_RIGHT

j sothuhai

R5:

li \$a0, 0x6D

jal SHOW_7SEG_RIGHT

j sothuhai

R6:

li \$a0, 0x7D

jal SHOW_7SEG_RIGHT

j sothuhai

R7:

li \$a0, 0x07

jal SHOW_7SEG_RIGHT

j sothuhai

R8:

li \$a0, 0x7F

jal SHOW_7SEG_RIGHT

j sothuhai

R9:

li \$a0, 0x6F

jal SHOW_7SEG_RIGHT

j sothuhai

sothuhai:

sub \$s0, \$s0, \$t1

div \$s0, \$t2

mflo \$t3

div \$t3, \$t2

mfhi \$t1

beq \$t1, 0, L0

beq \$t1, 1, L1

beq \$t1, 2, L2

beq \$t1, 3, L3

beq \$t1, 4, L4

beq \$t1, 5, L5

beq \$t1, 6, L6

beq \$t1, 7, L7

beq \$t1, 8, L8

beq \$t1, 9, L9

L0:

li \$a0, 0x3F

jal SHOW_7SEG_LEFT

j exit

L1:

li \$a0, 0x6

jal SHOW_7SEG_LEFT

j exit

L2:

li \$a0, 0x5B

jal SHOW_7SEG_LEFT

j exit

L3:

li \$a0, 0x4F

jal SHOW_7SEG_LEFT

j exit

L4:

li \$a0, 0x66

jal SHOW_7SEG_LEFT

j exit

L5:

li \$a0, 0x6D

jal SHOW_7SEG_LEFT

j exit

L6:

```
li $a0, 0x7D
jal SHOW_7SEG_LEFT
j exit
```

L7:

```
li $a0, 0x7
jal SHOW_7SEG_LEFT
j exit
```

L8:

```
li $a0, 0x7F
jal SHOW_7SEG_LEFT
j sothuhai
```

L9:

```
li $a0, 0x6F
jal SHOW_7SEG_LEFT
j exit
```

exit:

```
li $v0, 10
syscall
```

endmain:

```
li $v0, 10
syscall
```

#-----

Function SHOW_7SEG_RIGHT : turn on/off the 7seg

param[in] \$a0 value to shown

remark \$t0 changed

#-----

+TH1: Nhập số 26



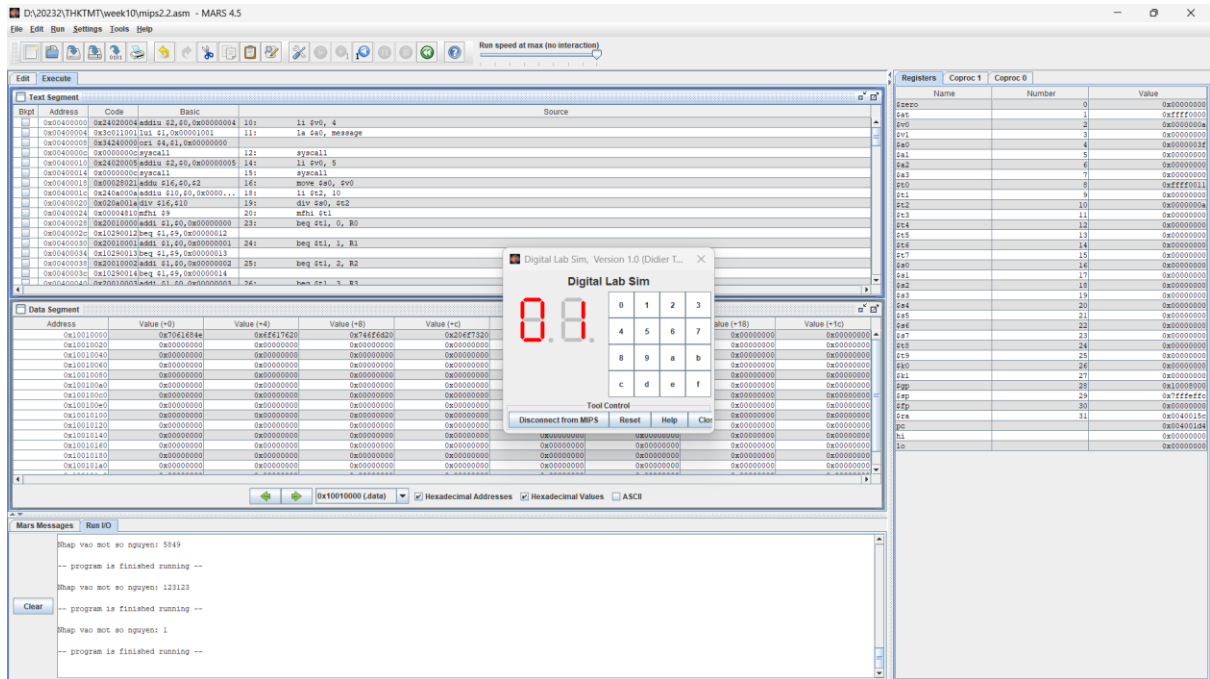
D:\2023\THKMT\week10\mps2.2.asm - MARS 4.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Test Segment

Dispt	Address	Code	Basic	Source
0x00400000	0x00400000	addiu \$2,\$0,0x00000004	101	li \$rv0, 4
0x00400004	0x00400004	li \$t0,0x11111111	111	li \$a0, message
0x00400008	0x00400008	0x34240000	121	syscall
0x0040000c	0x0040000c	0x00000000	141	li \$rv0, 5
0x00400010	0x00400010	0x00000000	151	syscall
0x00400014	0x00400014	0x00000000	161	move \$a0, \$rv0
0x00400018	0x00400018	0x00000000	181	li \$t2, 10
0x0040001c	0x0040001c	0x00000000	191	div \$a0, \$t2
0x00400020	0x00400020	0x00000000	201	mfhi \$t1
0x00400024	0x00400024	0x00000000	231	beq \$t1, 0, R0
0x00400028	0x00400028	0x00000000	241	beq \$t1, 1, R1
0x0040002c	0x0040002c	0x00000000	251	beq \$t1, 2, R2
0x00400030	0x00400030	0x00000000	261	beq \$t1, 3, R3
0x00400034	0x00400034	0x00000000	271	beq \$t1, 4, R4
0x00400038	0x00400038	0x00000000	281	beq \$t1, 5, R5
0x0040003c	0x0040003c	0x00000000	291	beq \$t1, 6, R6
0x00400040	0x00400040	0x00000000	301	beq \$t1, 7, R7
0x00400044	0x00400044	0x00000000	311	beq \$t1, 8, R8
0x00400048	0x00400048	0x00000000	321	beq \$t1, 9, R9
0x0040004c	0x0040004c	0x00000000	331	beq \$t1, 10, R10
0x00400050	0x00400050	0x00000000	341	beq \$t1, 11, R11
0x00400054	0x00400054	0x00000000	351	beq \$t1, 12, R12
0x00400058	0x00400058	0x00000000	361	beq \$t1, 13, R13
0x0040005c	0x0040005c	0x00000000	371	beq \$t1, 14, R14
0x00400060	0x00400060	0x00000000	381	beq \$t1, 15, R15
0x00400064	0x00400064	0x00000000	391	beq \$t1, 16, R16
0x00400068	0x00400068	0x00000000	401	beq \$t1, 17, R17
0x0040006c	0x0040006c	0x00000000	411	beq \$t1, 18, R18
0x00400070	0x00400070	0x00000000	421	beq \$t1, 19, R19
0x00400074	0x00400074	0x00000000	431	beq \$t1, 20, R20
0x00400078	0x00400078	0x00000000	441	beq \$t1, 21, R21
0x0040007c	0x0040007c	0x00000000	451	beq \$t1, 22, R22
0x00400080	0x00400080	0x00000000	461	beq \$t1, 23, R23
0x00400084	0x00400084	0x00000000	471	beq \$t1, 24, R24
0x00400088	0x00400088	0x00000000	481	beq \$t1, 25, R25
0x0040008c	0x0040008c	0x00000000	491	beq \$t1, 26, R26
0x00400090	0x00400090	0x00000000	501	beq \$t1, 27, R27
0x00400094	0x00400094	0x00000000	511	beq \$t1, 28, R28
0x00400098	0x00400098	0x00000000	521	beq \$t1, 29, R29
0x0040009c	0x0040009c	0x00000000	531	beq \$t1, 30, R30
0x004000a0	0x004000a0	0x00000000	541	beq \$t1, 31, R31
0x004000a4	0x004000a4	0x00000000	551	beq \$t1, 32, R32
0x004000a8	0x004000a8	0x00000000	561	beq \$t1, 33, R33
0x004000ac	0x004000ac	0x00000000	571	beq \$t1, 34, R34
0x004000b0	0x004000b0	0x00000000	581	beq \$t1, 35, R35
0x004000b4	0x004000b4	0x00000000	591	beq \$t1, 36, R36
0x004000b8	0x004000b8	0x00000000	601	beq \$t1, 37, R37
0x004000bc	0x004000bc	0x00000000	611	beq \$t1, 38, R38
0x004000c0	0x004000c0	0x00000000	621	beq \$t1, 39, R39
0x004000c4	0x004000c4	0x00000000	631	beq \$t1, 40, R40
0x004000c8	0x004000c8	0x00000000	641	beq \$t1, 41, R41
0x004000cc	0x004000cc	0x00000000	651	beq \$t1, 42, R42
0x004000d0	0x004000d0	0x00000000	661	beq \$t1, 43, R43
0x004000d4	0x004000d4	0x00000000	671	beq \$t1, 44, R44
0x004000d8	0x004000d8	0x00000000	681	beq \$t1, 45, R45
0x004000dc	0x004000dc	0x00000000	691	beq \$t1, 46, R46
0x004000e0	0x004000e0	0x00000000	701	beq \$t1, 47, R47
0x004000e4	0x004000e4	0x00000000	711	beq \$t1, 48, R48
0x004000e8	0x004000e8	0x00000000	721	beq \$t1, 49, R49
0x004000ec	0x004000ec	0x00000000	731	beq \$t1, 50, R50
0x004000f0	0x004000f0	0x00000000	741	beq \$t1, 51, R51
0x004000f4	0x004000f4	0x00000000	751	beq \$t1, 52, R52
0x004000f8	0x004000f8	0x00000000	761	beq \$t1, 53, R53
0x004000fc	0x004000fc	0x00000000	771	beq \$t1, 54, R54
0x00400100	0x00400100	0x00000000	781	beq \$t1, 55, R55
0x00400104	0x00400104	0x00000000	791	beq \$t1, 56, R56
0x00400108	0x00400108	0x00000000	801	beq \$t1, 57, R57
0x0040010c	0x0040010c	0x00000000	811	beq \$t1, 58, R58
0x00400110	0x00400110	0x00000000	821	beq \$t1, 59, R59
0x00400114	0x00400114	0x00000000	831	beq \$t1, 60, R60
0x00400118	0x00400118	0x00000000	841	beq \$t1, 61, R61
0x0040011c	0x0040011c	0x00000000	851	beq \$t1, 62, R62
0x00400120	0x00400120	0x00000000	861	beq \$t1, 63, R63
0x00400124	0x00400124	0x00000000	871	beq \$t1, 64, R64
0x00400128	0x00400128	0x00000000	881	beq \$t1, 65, R65
0x0040012c	0x0040012c	0x00000000	891	beq \$t1, 66, R66
0x00400130	0x00400130	0x00000000	901	beq \$t1, 67, R67
0x00400134	0x00400134	0x00000000	911	beq \$t1, 68, R68
0x00400138	0x00400138	0x00000000	921	beq \$t1, 69, R69
0x0040013c	0x0040013c	0x00000000	931	beq \$t1, 70, R70
0x00400140	0x00400140	0x00000000	941	beq \$t1, 71, R71
0x00400144	0x00400144	0x00000000	951	beq \$t1, 72, R72
0x00400148	0x00400148	0x00000000	961	beq \$t1, 73, R73
0x0040014c	0x0040014c	0x00000000	971	beq \$t1, 74, R74
0x00400150	0x00400150	0x00000000	981	beq \$t1, 75, R75
0x00400154	0x00400154	0x00000000	991	beq \$t1, 76, R76
0x00400158	0x00400158	0x00000000	1001	beq \$t1, 77, R77
0x0040015c	0x0040015c	0x00000000	1011	beq \$t1, 78, R78
0x00400160	0x00400160	0x00000000	1021	beq \$t1, 79, R79
0x00400164	0x00400164	0x00000000	1031	beq \$t1, 80, R80
0x00400168	0x00400168	0x00000000	1041	beq \$t1, 81, R81
0x0040016c	0x0040016c	0x00000000	1051	beq \$t1, 82, R82
0x00400170	0x00400170	0x00000000	1061	beq \$t1, 83, R83
0x00400174	0x00400174	0x00000000	1071	beq \$t1, 84, R84
0x00400178	0x00400178	0x00000000	1081	beq \$t1, 85, R85
0x0040017c	0x0040017c	0x00000000	1091	beq \$t1, 86, R86
0x00400180	0x00400180	0x00000000	1101	beq \$t1, 87, R87
0x00400184	0x00400184	0x00000000	1111	beq \$t1, 88, R88
0x00400188	0x00400188	0x00000000	1121	beq \$t1, 89, R89
0x0040018c	0x0040018c	0x00000000	1131	beq \$t1, 90, R90
0x00400190	0x00400190	0x00000000	1141	beq \$t1, 91, R91
0x00400194	0x00400194	0x00000000	1151	beq \$t1, 92, R92
0x00400198	0x00400198	0x00000000	1161	beq \$t1, 93, R93
0x0040019c	0x0040019c	0x00000000	1171	beq \$t1, 94, R94
0x004001a0	0x004001a0	0x00000000	1181	beq \$t1, 95, R95
0x004001a4	0x004001a4	0x00000000	1191	beq \$t1, 96, R96
0x004001a8	0x004001a8	0x00000000	1201	beq \$t1, 97, R97
0x004001ac	0x004001ac	0x00000000	1211	beq \$t1, 98, R98
0x004001b0	0x004001b0	0x00000000	1221	beq \$t1, 99, R99
0x004001b4	0x004001b4	0x00000000	1231	beq \$t1, 100, R100
0x004001b8	0x004001b8	0x00000000	1241	beq \$t1, 101, R101
0x004001bc	0x004001bc	0x00000000	1251	beq \$t1, 102, R102
0x004001c0	0x004001c0	0x00000000	1261	beq \$t1, 103, R103
0x004001c4	0x004001c4	0x00000000	1271	beq \$t1, 104, R104
0x004001c8	0x004001c8	0x00000000	1281	beq \$t1, 105, R105
0x004001cc	0x004001cc	0x00000000	1291	beq \$t1, 106, R106
0x004001d0	0x004001d0	0x00000000	1301	beq \$t1, 107, R107
0x004001d4	0x004001d4	0x00000000	1311	beq \$t1, 108, R108
0x004001d8	0x004001d8	0x00000000	1321	beq \$t1, 109, R109
0x004001dc	0x004001dc	0x00000000	1331	beq \$t1, 110, R110
0x004001e0	0x004001e0	0x00000000	1341	beq \$t1, 111, R111
0x004001e4	0x004001e4	0x00000000	1351	beq \$t1, 112, R112
0x004001e8	0x004001e8	0x00000000	1361	beq \$t1, 113, R113
0x004001ec	0x004001ec	0x00000000	1371	beq \$t1, 114, R114
0x004001f0	0x004001f0	0x00000000	1381	beq \$t1, 115, R115
0x004001f4	0x004001f4	0x00000000	1391	beq \$t1, 116, R116
0x004001f8	0x004001f8	0x00000000	1401	beq \$t1, 117, R117
0x004001fc	0x004001fc	0x00000000	1411	beq \$t1, 118, R118
0x00400200	0x00400200	0x00000000	1421	beq \$t1, 119, R119
0x00400204	0x00400204	0x00000000	1431	beq \$t1, 120, R120
0x00400208	0x00400208	0x00000000	1441	beq \$t1, 121, R121
0x0040020c	0x0040020c	0x00000000	1451	beq \$t1, 122, R122
0x00400210	0x00400210	0x00000000	1461	beq \$t1, 123, R123
0x00400214	0x00400214	0x00000000	1471	beq \$t1, 124, R124
0x00400218	0x00400218	0x00000000	1481	beq \$t1, 125, R125
0x0040021c	0x0040021c	0x00000000	1491	beq \$t1, 126, R126
0x00400220	0x00400220	0x00000000	1501	beq \$t1, 127, R127
0x00400224	0x00400224	0x00000000	1511	beq \$t1, 128, R128
0x00400228	0x00400228	0x00000000	1521	beq \$t1, 129, R129
0x0040022c	0x0040022c	0x00000000	1531	beq \$t1, 130, R130
0x00400230	0x00400230	0x00000000	1541	beq \$t1, 131, R131
0x00400234	0x00400234	0x00000000	1551	beq \$t1, 132, R132
0x00400238	0x00400238	0x00000000	1561	beq \$t1, 133, R133
0x0040023c	0x0040023c	0x00000000	1571	beq \$t1, 134, R134
0x00400240	0x00400240	0x00000000	1581	beq \$t1, 135, R135
0x00400244	0x00400244	0x00000000	1591	beq \$t1, 136, R136
0x00400248	0x00400248	0x00000000	1601	beq \$t1, 137,



→ Kết quả đúng với lý thuyết

Assignment 3:

- Code:

```
.eqv SEVENSEG_LEFT 0xFFFF0011      # Địa chỉ của đèn led 7 đoạn trái.
```

Bit 0 = đoạn a;

Bit 1 = đoạn b; ...

Bit 7 = dấu .

```
.eqv SEVENSEG_RIGHT 0xFFFF0010     # Địa chỉ của đèn led 7 đoạn phải
```

```
.data
```

```
message: .ascii "Nhập vào kí tự: "
```

```
.text
```

```
main:
```

```
    li $v0, 4
```

```
    la $a0, message
```

```
    syscall
```

```
    # Nhập ký tự
```

```
    li $v0, 12
```

```
syscall  
move $s0, $v0    # Lưu mã ASCII của kí tự vào $s0
```

```
li $t2, 10  
div $s0, $t2  
mfhi $t1
```

So sánh giá trị của t1 với các giá trị từ 0 đến 9 và hiển thị tương ứng

```
beq $t1, 0, R0  
beq $t1, 1, R1  
beq $t1, 2, R2  
beq $t1, 3, R3  
beq $t1, 4, R4  
beq $t1, 5, R5  
beq $t1, 6, R6  
beq $t1, 7, R7  
beq $t1, 8, R8  
beq $t1, 9, R9
```

R0:

```
li $a0, 0x3F  
jal SHOW_7SEG_RIGHT  
j sothuhai
```

R1:

```
li $a0, 0x06  
jal SHOW_7SEG_RIGHT  
j sothuhai
```

R2:

```
li $a0, 0x5B  
jal SHOW_7SEG_RIGHT
```

j sothuhai

R3:

li \$a0, 0x4F

jal SHOW_7SEG_RIGHT

j sothuhai

R4:

li \$a0, 0x66

jal SHOW_7SEG_RIGHT

j sothuhai

R5:

li \$a0, 0x6D

jal SHOW_7SEG_RIGHT

j sothuhai

R6:

li \$a0, 0x7D

jal SHOW_7SEG_RIGHT

j sothuhai

R7:

li \$a0, 0x07

jal SHOW_7SEG_RIGHT

j sothuhai

R8:

li \$a0, 0x7F

jal SHOW_7SEG_RIGHT

j sothuhai

R9:

li \$a0, 0x6F

jal SHOW_7SEG_RIGHT

j sothuhai

sothuhai:

sub \$s0, \$s0, \$t1

div \$s0, \$t2

mflo \$t3

div \$t3, \$t2

mfhi \$t1

beq \$t1, 0, L0

beq \$t1, 1, L1

beq \$t1, 2, L2

beq \$t1, 3, L3

beq \$t1, 4, L4

beq \$t1, 5, L5

beq \$t1, 6, L6

beq \$t1, 7, L7

beq \$t1, 8, L8

beq \$t1, 9, L9

L0:

li \$a0, 0x3F

jal SHOW_7SEG_LEFT

j exit

L1:

li \$a0, 0x6

jal SHOW_7SEG_LEFT

j exit

L2:

li \$a0, 0x5B

jal SHOW_7SEG_LEFT

j exit

L3:

li \$a0, 0x4F

jal SHOW_7SEG_LEFT

j exit

L4:

li \$a0, 0x66

jal SHOW_7SEG_LEFT

j exit

L5:

li \$a0, 0x6D

jal SHOW_7SEG_LEFT

j exit

L6:

li \$a0, 0x7D

jal SHOW_7SEG_LEFT

j exit

L7:

li \$a0, 0x7

jal SHOW_7SEG_LEFT

j exit

L8:

li \$a0, 0x7F

jal SHOW_7SEG_LEFT

j sothuhai

L9:

li \$a0, 0x6F

jal SHOW_7SEG_LEFT

j exit

exit:

```
li $v0, 10
```

```
syscall
```

```
endmain:
```

```
li $v0, 10
```

```
syscall
```

```
#-----
```

```
# Function SHOW_7SEG_RIGHT : 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:
```

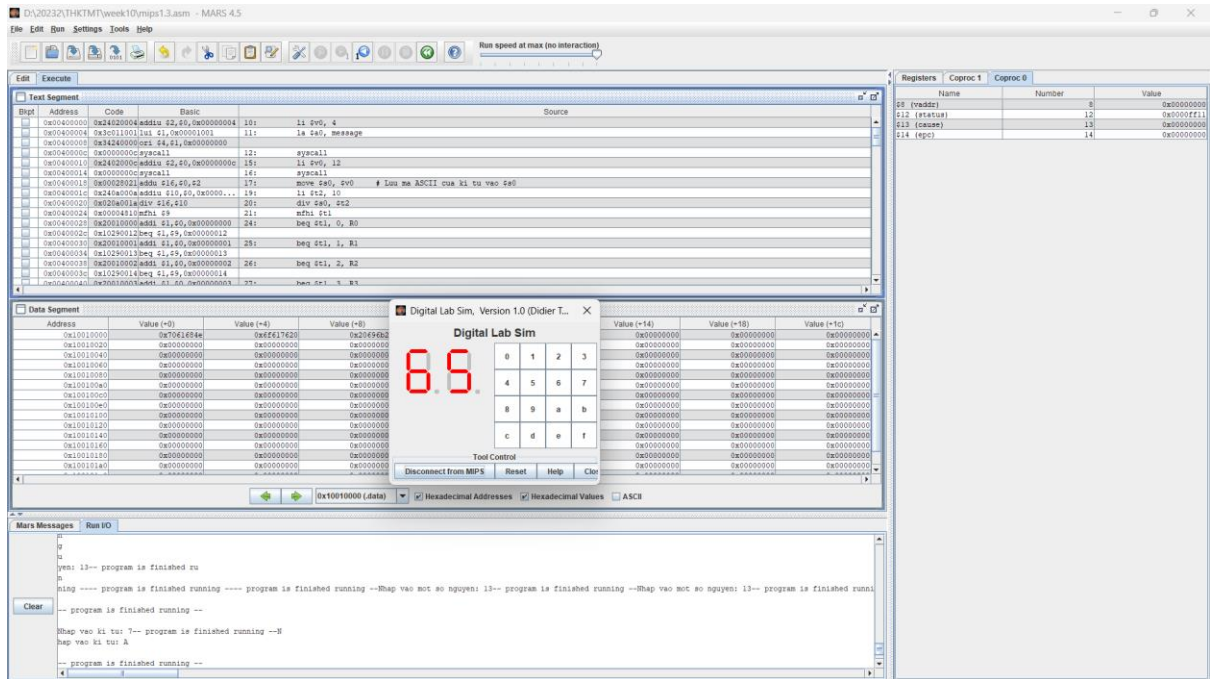
```
li $t0, SEVENSEG_RIGHT # assign port's address
```

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

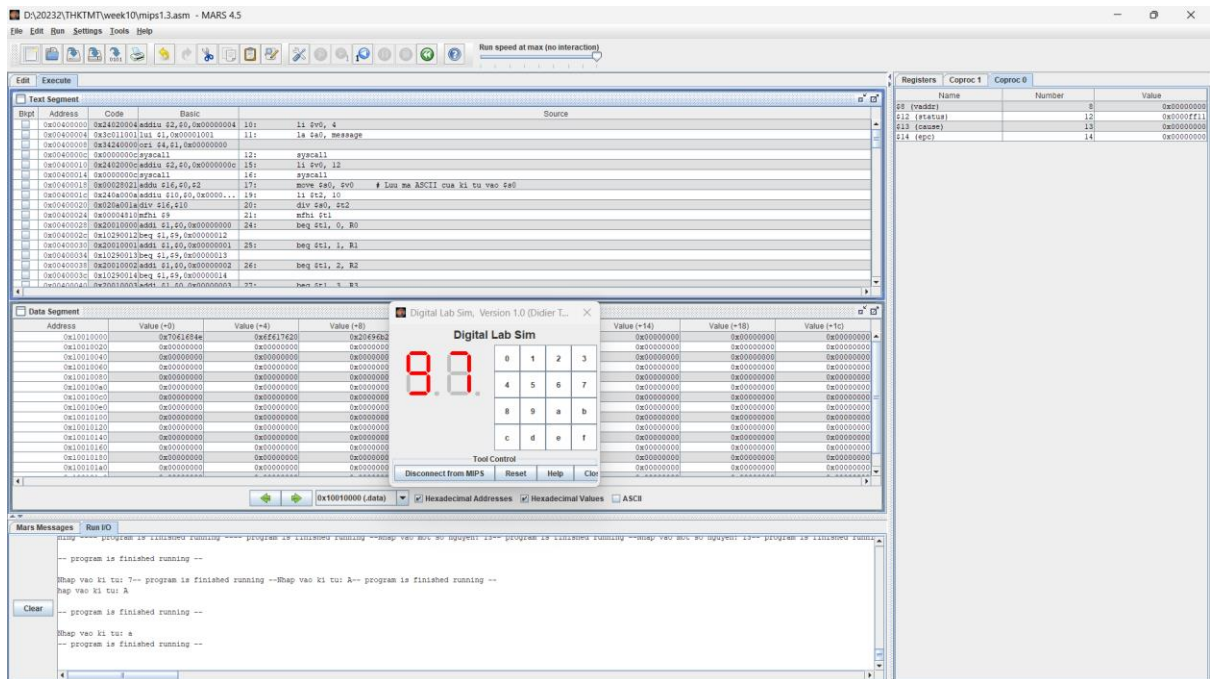
```
jr $ra
```

- Kết quả:

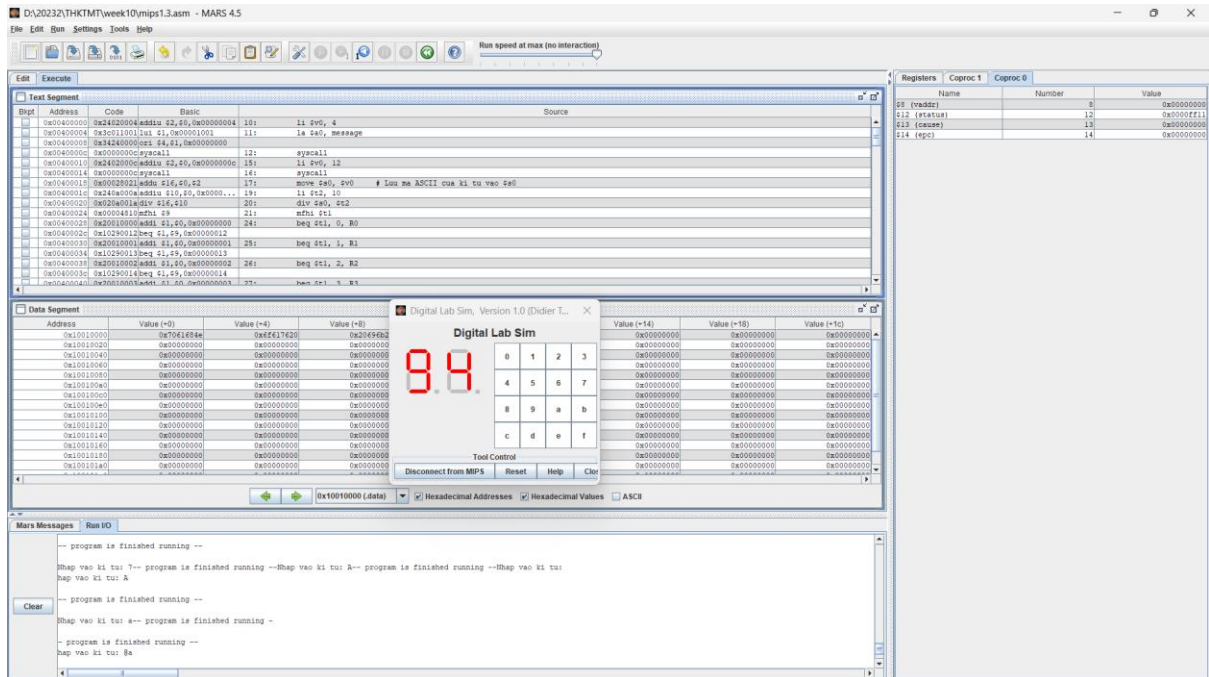
+ Nhập vào kí tự “A”



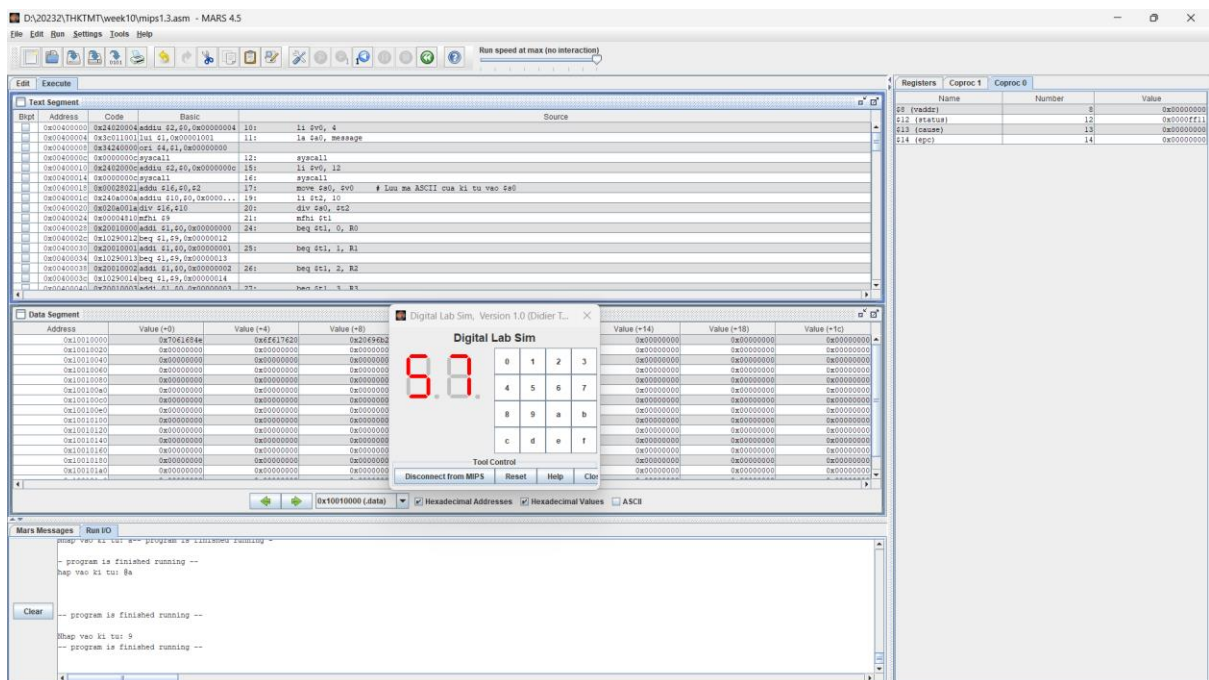
- Nhập vào kí tự “a”



- Nhập vào kí tự “@”



- Nhập vào kí tự là số, ở đây nhập số 9



➔ Kết quả đúng với lý thuyết

Assignment 4

- Code:

```
.eqv MONITOR_SCREEN 0x10010000    #Địa chỉ bắt đầu của bộ nhớ màn hình
```

```
.eqv RED    0x00EC145C    #Các giá trị màu tương ứng
```

```
.eqv BLUE   0x000000FF
```

```

.eqv WHITE  0x00FFFFFF
.eqv YELLOW 0x00FFFF00

.data

row1: .word
0x000000FF,0x00FFFF00,0x000000FF,0x00FFFF00,0x000000FF,0x00FFFF00,0x000000FF,
0x00FFFF00

row2: .word
0x00FFFF00,0x000000FF,0x00FFFF00,0x000000FF,0x00FFFF00,0x000000FF,0x00FFFF00
,0x000000FF


.text


    li $k0, MONITOR_SCREEN      #Nap dia chi bat dau cua man hinh
    li $s0, 8

    la $a0,row1                  #Luu dia chi tung dong
    jal print_screen

    la $a0,row2
    jal print_screen

    la $a0,row1
    jal print_screen

    la $a0,row2
    jal print_screen

    la $a0,row1
    jal print_screen

    la $a0,row2
    jal print_screen

    la $a0,row1
    jal print_screen

```

```

        la $a0,row2
        jal print_screen
end_main:
        li $v0,10                #ket thuc chuong trinh
        syscall

print_screen:
        li $t1,0

loop_print:
        beq $t1,32,end          #kiem tra xem ket thuc row chua
                                #i= 7 ?

        add $t4,$a0,$t1          #Lay gia tri cua row[i] => $t0
        lw $t0,0($t4)            #

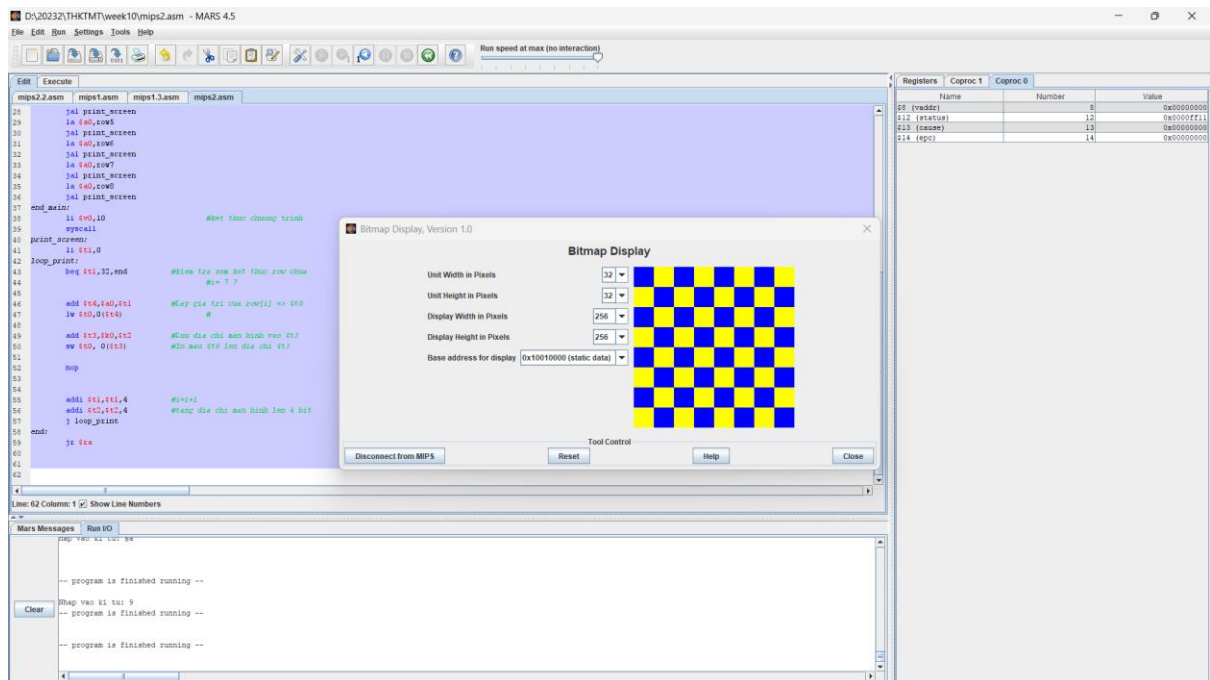
        add $t3,$k0,$t2          #Luu dia chi man hinh vao $t3
        sw $t0, 0($t3)           #In mau $t0 len dia chi $t3

        nop

        addi $t1,$t1,4           #i=i+1
        addi $t2,$t2,4           #tang dia chi man hinh len 4 bit
        j loop_print
end:
        jr $ra

```

- Kết quả:



➔ Kết quả đúng với lý thuyết

Assignment 5:

- Code:

.data

message: .asciiz "Nhap vao 4 so x1,y1,x2,y2(sau moi lan nhap vao 1 so an enter) : "

.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 \$v0 , 4

la \$a0 , message

syscall

li \$v0 , 5

syscall

move \$s0 , \$v0

```

li $v0 , 5
syscall
move $s1, $v0
li $v0 , 5
syscall
move $s2, $v0
li $v0 , 5
syscall
move $s3, $v0 #nhap vao du lieu
main:
blt $s0,$s2,next #if $x1 < x2
move $t0 , $s0
move $s0 , $s2
move $s2, $t0
next:
blt $s1,$s3,solve #if $x1 < x2
move $t1 , $s1
move $s1 , $s3
move $s3, $t1
solve:
li $k0, MONITOR_SCREEN

addi $s0,$s0,-1
addi $s1,$s1,-1
addi $s2,$s2,-1
addi $s3,$s3,-1

li $a1 , 8
li $a2 ,4

```

```
mul $t0 , $s1, $a1
add $t0 , $t0, $s0
mul $t0 , $t0 , 4
add $k0 , $k0 , $t0
```

```
move $k1,$k0
move $t1,$s1 # i = y1
loop:
bgt $t1,$s3,end
move $t2,$s0 #j = x1
beq $t1,$s1,loop_print1
beq $t1,$s3,loop_print1
j loop_print2
loop_print1:
bgt $t2,$s2,next_row
li $t0, RED
sw $t0, 0($k1)
addi $t2 , $t2 , 1
addi $k1, $k1, 4
j loop_print1
loop_print2:
bgt $t2,$s2,next_row
beq $t2,$s0,print1
beq $t2,$s2,print1
li $t0 , YELLOW
sw $t0,0($k1)
addi $t2, $t2 , 1
addi $k1, $k1, 4
j loop_print2
```

```

printl:

li $t0 , RED

sw $t0 , 0($k1)

addi $t2 , $t2 , 1

addi $k1 , $k1 , 4

j loop_print2

next_row:

addi $t1,$t1 , 1

addi $k0 , $k0 , 32

move $k1 , $k0

j loop

end:

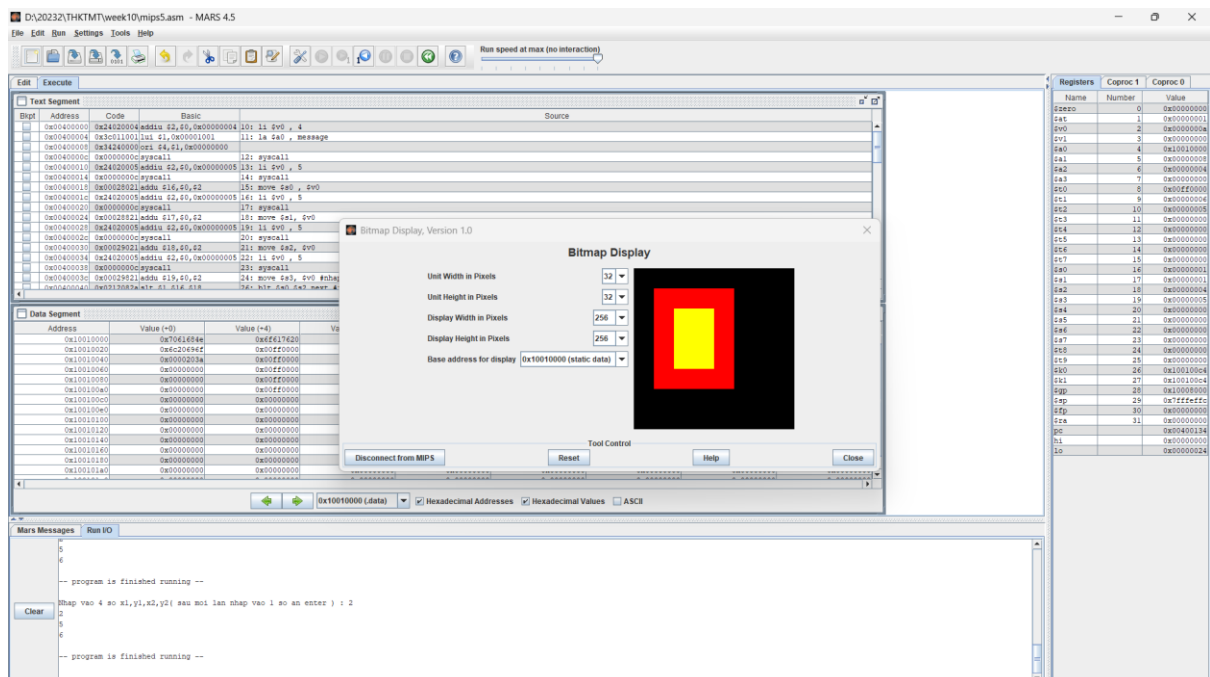
li $v0 , 10

syscall

```

- Kết quả:

Mở bitmap display trong tool sau khi ấn vào nút thực thi (hình cò lê), kết nối bitmap với MIPS, chỉnh độ rộng, độ dài đơn vị điểm ảnh là 8 pixel, chỉnh độ rộng, độ dài của màn hình bitmap là 512 pixel. Khi đó, ở màn hình bitmap có $(256/32)*(256/32) = 64$ điểm ảnh. Chạy để hiện kết quả.



➔ Kết quả đúng với lí thuyết