

IT3280 –THỰC HÀNH KIẾN TRÚC MÁY TÍNH

Tuần 10:

Họ và tên	Phạm Quốc Cường
Mã số sinh viên	20225604

Assignment 1:

Code:

#Laboratory Exercise 10 Home Assignment 1

.data

Number: .word 0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F, 0x6F

.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, 0x3F # set value for segments

jal SHOW_7SEG_LEFT #show

nop

la \$s0, Number

lw \$a0, 0(\$s0) # set value for segments

jal SHOW_7SEG_RIGHT # show

loop:

addi \$s0, \$s0, 4

loop0to9: lw \$a0, 0(\$s0) # set value for segments

jal SHOW_7SEG_RIGHT # show

addi \$s0, \$s0, 4

li \$t1, 0x6F

bne \$t1, \$a0, loop0to9

nop

subi \$s0, \$s0, 4

loop9to0: lw \$a0, 0(\$s0) # set value for segments

jal SHOW_7SEG_RIGHT # show

subi \$s0, \$s0, 4

li \$t1, 0x3F

bne \$t1, \$a0, loop9to0

j loop

#-----

Function SHOW_7SEG_LEFT

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

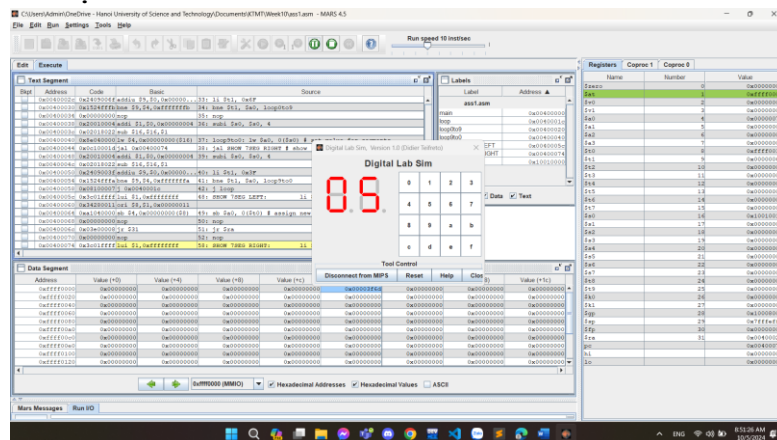
#-----

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

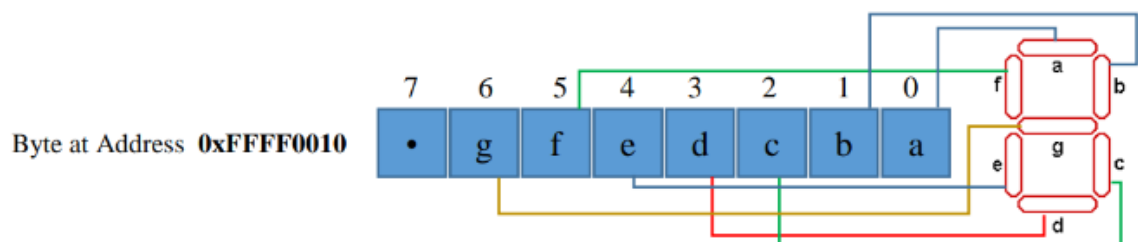
Result:

Digital Lab Sim in ra số từ 0 -> 9 rồi từ 9 -> 0

Ví dụ:



Explain:



*Bảng mã hóa

Digit	a	b	c	d	e	f	g	Hex
0	1	1	1	1	1	1	0	0x3F

1	0	1	1	0	0	0	0	0X06
2	1	1	0	1	1	0	1	0X5B
3	1	1	1	1	0	0	1	0X4F
4	0	1	1	0	0	1	1	0X66
5	1	1	0	1	1	0	1	0X6D
6	1	0	1	1	1	1	1	0X7D
7	1	1	1	0	0	0	0	0X07
8	1	1	1	1	1	1	1	0X7F
9	1	1	1	0	0	1	1	0X6F

- Giá trị 0x3F (số 0) được gán vào thanh ghi \$a0 rồi hiển thị lên led trái.
- Địa chỉ của mảng Number được nạp vào thanh ghi \$s0. Sau đó, giá trị đầu tiên của mảng Number được nạp vào \$a0 và hiển thị trên led phải.
- Một vòng lặp được thực hiện để lặp qua từng số trong mảng Number.
- Giá trị tiếp theo của mảng Number được nạp vào \$a0 và hiển thị lên led phải
- Vòng lặp tiếp tục cho đến khi gặp số 9 (0x6F), sau đó, vòng lặp sẽ thực hiện lặp lại từ 9 về 0.

Assignment 2:

Code:

```
.eqv MONITOR_SCREEN 0x10010000    #Địa chỉ bắt đầu của bộ nhớ màn hình
.eqv RED 0x00FF0000                #Các giá trị màu tương ứng
.eqv GREEN 0x0000FF00
.eqv BLUE 0x000000FF
.eqv WHITE 0x00FFFFFF
.eqv YELLOW 0x00FFFF00
.eqv BLACK 0x00000000
.text
li $k0, MONITOR_SCREEN            #Nạp địa chỉ bắt đầu của màn hình
addi $k0, $k0, 28
li $t0, RED
sw $t0, 0($k0)
add $t1, $zero, $zero
li $t2, 7
nop

loop:
    li $t0, BLACK
    sw $t0, 0($k0)
    nop
```

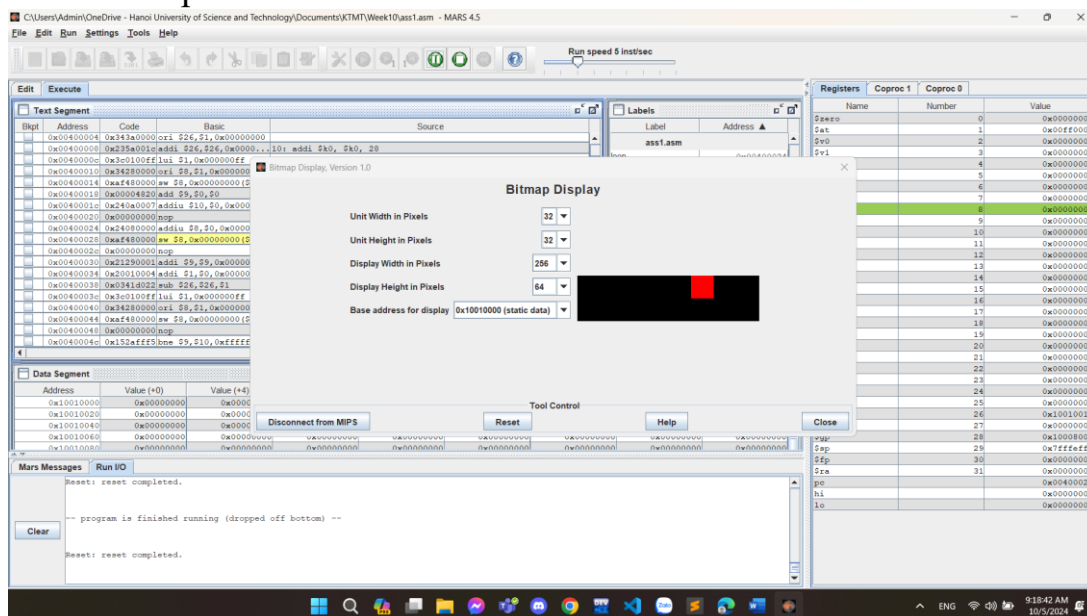
```
addi $t1, $t1, 1
subi $k0, $k0, 4
```

```
li $t0, RED
sw $t0, 0($k0)
nop
```

```
bne $t1, $t2, loop
```

Result:

Chương trình sẽ vẽ ra một hình vuông màu đỏ 2x2 ở phía bên trái rồi di chuyển dần về bên phải.



Explain:

- Như cài đặt trên của màn hình hiển thị, mỗi unit sẽ có chiều dài bằng chiều rộng bằng 32 pixels, chiều dài và chiều rộng hiển thị lần lượt là 256 pixels và 64 pixels.
- Địa chỉ của màn hình được nạp vào thanh ghi \$k0, sau đó được tăng thêm 28 bytes. Điều này có thể đại diện cho vị trí bắt đầu của một dải pixels trên màn hình.
- Khởi tạo ô đỏ đầu tiên ở vị trí ngoài cùng bên phải.
- Sau mỗi vòng lặp, màu đen được ghi vào địa chỉ pixel, địa chỉ giảm đi 4 byte, giá trị màu đỏ được ghi vào địa chỉ mới tạo cảm giác ô đỏ đang di chuyển từ phải sang trái.