

Họ và tên: Bùi Văn Anh

MSSV: 20184026

Học phần: Thực hành kiến trúc máy tính

Mã lớp: 122032

Báo cáo LAB10

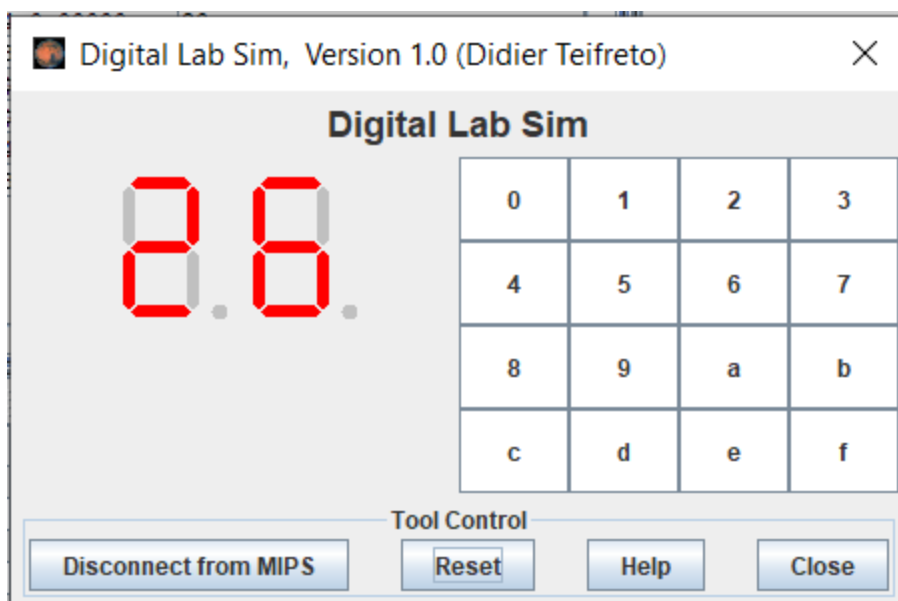
1. Assignment 1

```
.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, 0x5B                  # 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
```

Kết quả:



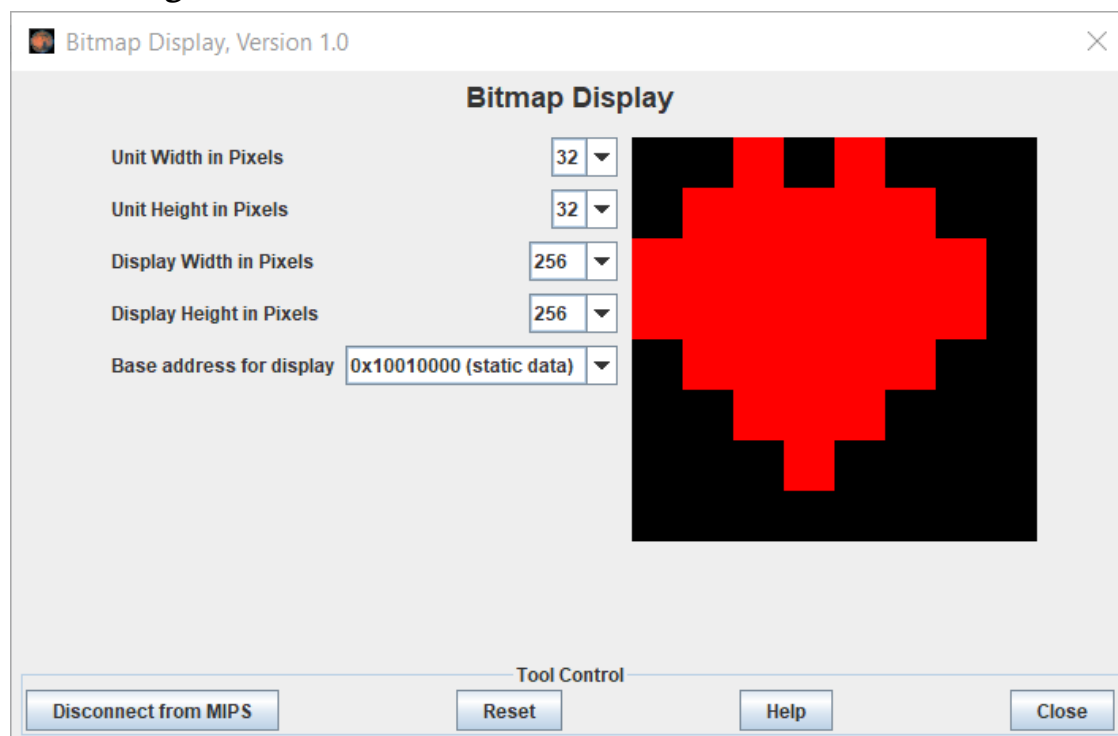
Số 2: Các bit sáng là a,b,d,e,g tương ứng với bit số 0,1,3,4,6 bằng 1

=> 1101101 = 0x5B

Số 6: Các bit sáng là a,c,d,e,f,g tương ứng với bit số 0,2,3,4,5,6 bằng 1

=> 1011111 = 0x7D

2. Assignment 2



Ví dụ: Vẽ hình trái tim

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

```
.eqv RED 0x00FF0000 #Các giá trị màu thường sử dụng
```

```
.eqv GREEN 0x0000FF00
```

```
.eqv BLUE 0x000000FF
```

```
.eqv WHITE 0x00FFFFFF
```

```
.eqv YELLOW 0x00FFFF00
```

```
.text
```

```
li $k0, MONITOR_SCREEN #Nhập địa chỉ bắt đầu của màn hình
```

```
#(0,2)=> 8
```

```
li $t0, RED
```

```
sw $t0, 8($k0)
```

```
nop
```

```
#(0,4)=>16
```

```
li $t0, RED
```

```
sw $t0, 16($k0)
```

```
nop
```

```
#1,1 =>36
```

```
li $t0, RED
```

```
sw $t0, 36($k0)
```

```
nop
```

```
li $t0, RED
```

```
sw $t0, 40($k0)
```

```
nop
```

```
#1,3 => 44
```

```
li $t0, RED
```

```
sw $t0, 44($k0)
```

```
nop
```

```
li $t0, RED
```

```
sw $t0, 48($k0)
```

```
nop
```

```
#1,5 =>52
```

```
li $t0, RED
```

```
sw $t0, 52($k0)
```

```
nop
```

```
#2,0 => 64
```

```
li $t0, RED
```

```
sw $t0, 64($k0)
```

```
nop
```

```
li $t0, RED
```

```
sw $t0, 68($k0)
```

```
nop
```

```
li $t0, RED
```

```
sw $t0, 72($k0)
```

```
nop
```

```
li $t0, RED
```

```
sw $t0, 76($k0)
```

```
nop
```

```
li $t0, RED
```

```
sw $t0, 80($k0)
```

```
nop
```

```
li $t0, RED
```

```
sw $t0, 84($k0)
```

```
nop
```

```
#2,6 => 88
```

```
li $t0, RED
```

```
sw $t0, 88($k0)
```

```
nop
```

```
#3,0=> 96
```

```
li $t0, RED
```

```
sw $t0, 96($k0)
```

```
nop
```

```
li $t0, RED
```

```
sw $t0, 100($k0)
```

```
nop
```

```
li $t0, RED
```

```
sw $t0, 104($k0)
```

```
nop
```

```
li $t0, RED
```

```
sw $t0, 108($k0)
```

```
nop
```

```
li $t0, RED
```

```
sw $t0, 112($k0)
```

```
nop
```

```
li $t0, RED
```

```
sw $t0, 116($k0)
```

```
nop
```

```
#3,6 => 120
```

```
li $t0, RED
```

```
sw $t0, 120($k0)
```

```
nop
```

```
#4,1=> 132
```

```
li $t0, RED
```

```
sw $t0, 132($k0)
```

```
nop
```

```
li $t0, RED
```

```
sw $t0, 136($k0)
```

```
nop
```

```
li $t0, RED
```

```
sw $t0, 140($k0)
```

```
nop
```

```
li $t0, RED
```

```
sw $t0, 144($k0)
```

```
nop
```

```
#4,5 => 148
```

```
li $t0, RED
```

```
sw $t0, 148($k0)
```

```
nop
```

```
#5,2 => 168
```

```
li $t0, RED
```

```
sw $t0, 168($k0)
```

```
nop
```

```
li $t0, RED
```

```
sw $t0, 172($k0)
```

```
nop
```

```
#5,4 => 176
```

```
li $t0, RED
```

```
sw $t0, 176($k0)
```

```
nop
```

```
#6,3 => 204
```

```
li $t0, RED
```

```
sw $t0, 204($k0)
```

```
nop
```

3. Assignment 3

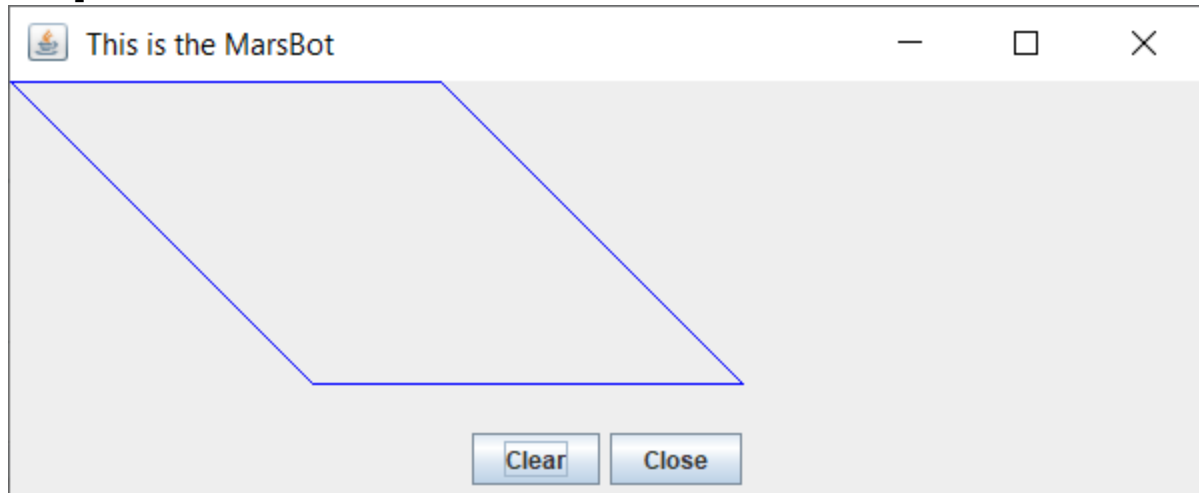
Sửa phần main để vẽ hình thoi:

```

main:      jal TRACK    # draw track line
           nop
           addi $a0, $zero, 90 # Marsbot rotates 90* and start running
           jal ROTATE
           nop
           jal GO
           nop
sleep1:    addi $v0,$zero,32 # Keep running by sleeping in 1000 ms
           li $a0,10000
           syscall
           jal UNTRACK # keep old track
           nop
           jal TRACK # and draw new track line
           nop
goDOWN:    addi $a0, $zero, 135 # Marsbot rotates 135*
           jal ROTATE
           nop
sleep2:    addi $v0,$zero,32 # Keep running by sleeping in 10000 ms
           li $a0,10000
           syscall
           jal UNTRACK # keep old track
           nop
           jal TRACK # and draw new track line
           nop
goAHEAD:   addi $a0, $zero, 270 # Marsbot rotates 270*
           jal ROTATE
           nop
sleep3:    addi $v0,$zero,32 # Keep running by sleeping in 2000 ms
           li $a0,10000
           syscall
           jal UNTRACK # keep old track
           nop
           jal TRACK # and draw new track line
           nop
goUP:      addi $a0,$zero,315
           jal ROTATE
           nop
sleep4:    addi $v0,$zero,32
           li $a0,10000
           syscall
end_main:

```

Kết quả:



4. Assignment 4

Đọc văn bản sau đó chuyển mã ASCII , rồi cộng thêm 1. Sau đó chuyển ASCII thành văn bản và in ra màn hình. Nhập vào nếu là “exit” thì thoát

CODE:

```
.eqv KEY_CODE 0xFFFF0004 # ASCII code from keyboard, 1 byte
.eqv KEY_READY 0xFFFF0000 # =1 if has a new keycode ?
# Auto clear after lw
.eqv DISPLAY_CODE 0xFFFF000C # ASCII code to show, 1 byte
.eqv DISPLAY_READY 0xFFFF0008 # =1 if the display is already to do
# Auto clear after sw
.eqv e 0x65
.eqv x 0x78
.eqv i 0x69
.eqv t 0x74
.text
li $k0, KEY_CODE
li $k1, KEY_READY
li $s0, DISPLAY_CODE
li $s1, DISPLAY_READY
loop: nop
WaitForKey: lw $t1, 0($k1) # $t1 = [$k1] = KEY_READY
             nop
             beq $t1, $zero, WaitForKey # if $t1 == 0 then Polling
             nop
#-----
ReadKey:     lw $t0, 0($k0) # $t0 = [$k0] = KEY_CODE
             j check_e
             nop
#-----
WaitForDis:  lw $t2, 0($s1) # $t2 = [$s1] = DISPLAY_READY
             nop
             beq $t2, $zero, WaitForDis # if $t2 == 0 then Polling
             nop
#-----
Encrypt:     addi $t0, $t0, 1 # change input key
#-----
ShowKey:     sw $t0, 0($s0) # show key
             nop
#-----
j loop
nop
check_e:     beq $t3,e,check_x
             bne $t0,e,WaitForDis
             add $t3,$t0,$zero
             j WaitForKey
```

```

check_x:    beq $t4,x,check_i
            bne $t0,x,WaitForDis
            add $t4,$t0,$zero
            j  WaitForDis
check_i:    beq $t5,i,check_t
            bne $t0,i,WaitForDis
            add $t5,$t0,$zero
            j  WaitForDis
check_t:    beq $t0,t,exit
            j  reset
reset:      li $t3,0
            li $t4,0
            li $t5,0
            j  WaitForDis
exit:      li $v0,10
            syscall

```

Kết quả:

