

Báo cáo tuần 10 (P1)

Thực hành kiến trúc máy tính

Họ tên: Phan Minh Anh Tuấn
MSSV: 20205227

Mục lục

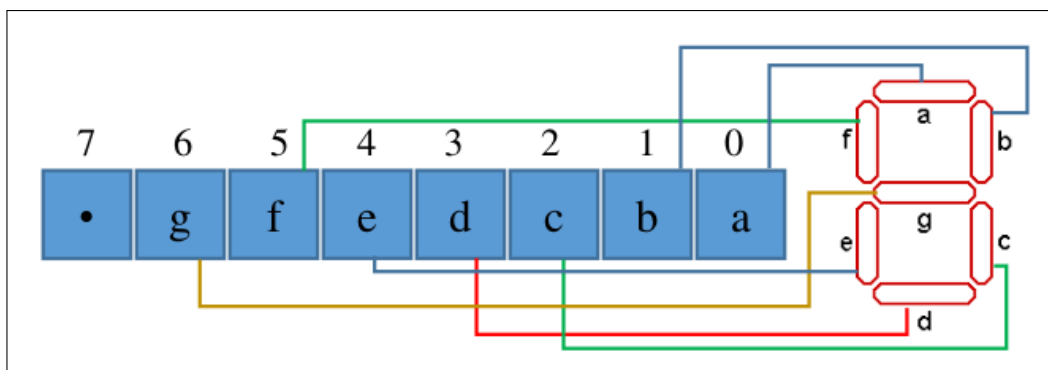
1	Assignment 1	2
1.1	Phân tích đề bài	2
1.2	Triển khai MIPS	2
2	Assignment 2	5
2.1	Triển khai MIPS	5

1 Assignment 1

1.1 Phân tích đề bài

Đề bài: Create a new project, type in, and build the program of Home Assignment 1. Show different values on LED.

MSSV: 20205227 → Cần hiển thị hai số 27



Hình 1: Cấu trúc để hiển thị số LED 7 thanh

Hiển thị số 2: Giá trị nhị phân là 11011011, giá trị hexa là 0xDF

Hiển thị số 7: Giá trị nhị phân là 10000111, giá trị hexa là 0x87

1.2 Triển khai MIPS

```
.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
```

Bước 1: Truyền địa chỉ LED 7 thanh bên trái và bên phải

```

main:
    li $a0, 0xDB          # set value for segments
    jal SHOW_7SEG_LEFT    # show
    nop
    li $a0, 0x87          # set value for segments
    jal SHOW_7SEG_RIGHT   # show
    nop

```

Bước 2: Truyền giá trị 0xDB, 0x87 vào \$a0 để hiển thị số trên LED 7 thanh

```

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

```

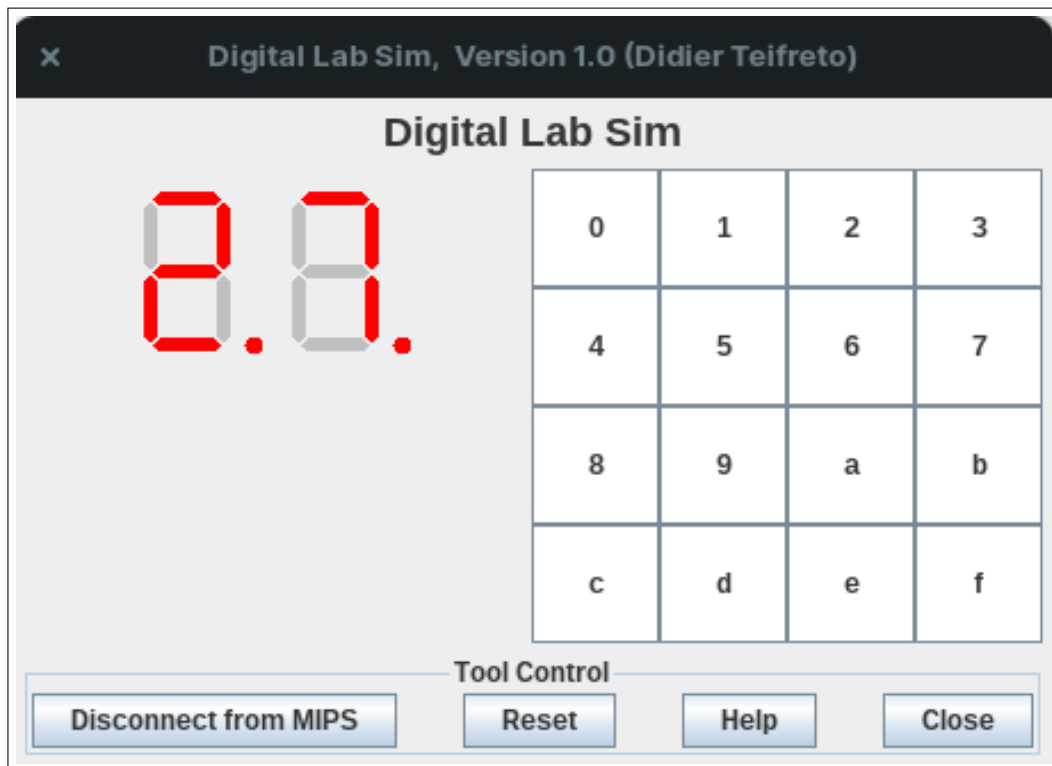
Bước 3: In giá trị của LED trái

```

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

```

Bước 4: In giá trị của LED phải



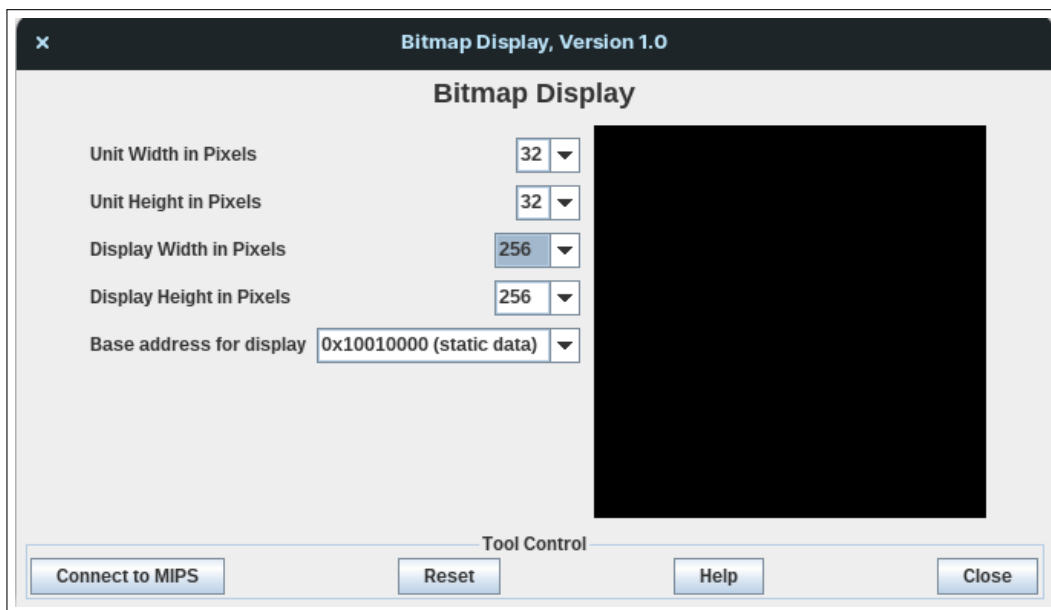
Hình 2: Kết quả của đoạn code

2 Assignment 2

Đề bài: Create a new project, type in, and build the program of Home Assignment 2. Draw something.

2.1 Triển khai MIPS

Thực hiện vẽ trên bitmap display với các thông số như hình vẽ



Hình 3: Các thông số của bitmap display

<code>.eqv</code>	<code>MONITOR_SCREEN</code>	<code>0x10010000</code>	<i>#Địa chỉ bắt đầu của bộ nhớ màn hình</i>
<code>.eqv</code>	<code>RED</code>	<code>0x00FF0000</code>	<i>#Các giá trị màu thường sử dụng</i>
<code>.eqv</code>	<code>GREEN</code>	<code>0x0000FF00</code>	
<code>.eqv</code>	<code>BLUE</code>	<code>0x000000FF</code>	
<code>.eqv</code>	<code>WHITE</code>	<code>0x00FFFFFF</code>	
<code>.eqv</code>	<code>YELLOW</code>	<code>0x00FFFF00</code>	

Hình 4: Gán các địa chỉ của màn hình và màu sắc vào các biến

```

.text
    li $k0, MONITOR_SCREEN          #Nap địa chỉ bắt đầu của màn hình

    li $t0, RED
    sw $t0, 36($k0)
    nop
    li $t0, RED
    sw $t0, 40($k0)
    nop
    li $t0, RED
    sw $t0, 44($k0)
    nop
    li $t0, RED
    sw $t0, 48($k0)
    nop
    li $t0, RED
    sw $t0, 52($k0)
    nop
    li $t0, RED
    sw $t0, 76($k0)
    nop
    li $t0, RED
    sw $t0, 108($k0)
    nop
    li $t0, RED
    sw $t0, 140($k0)
    nop
    li $t0, RED
    sw $t0, 172($k0)
    nop

```

Hình 5: Vẽ chữ T (thủ công)

```

        li $t0, WHITE
        addi $t1, $k0, 0
        addi $t3, $k0, 256

loop:
        lw $t2, 0($t1)
        bne $t2, $0, next
        sw $t0, 0($t1)

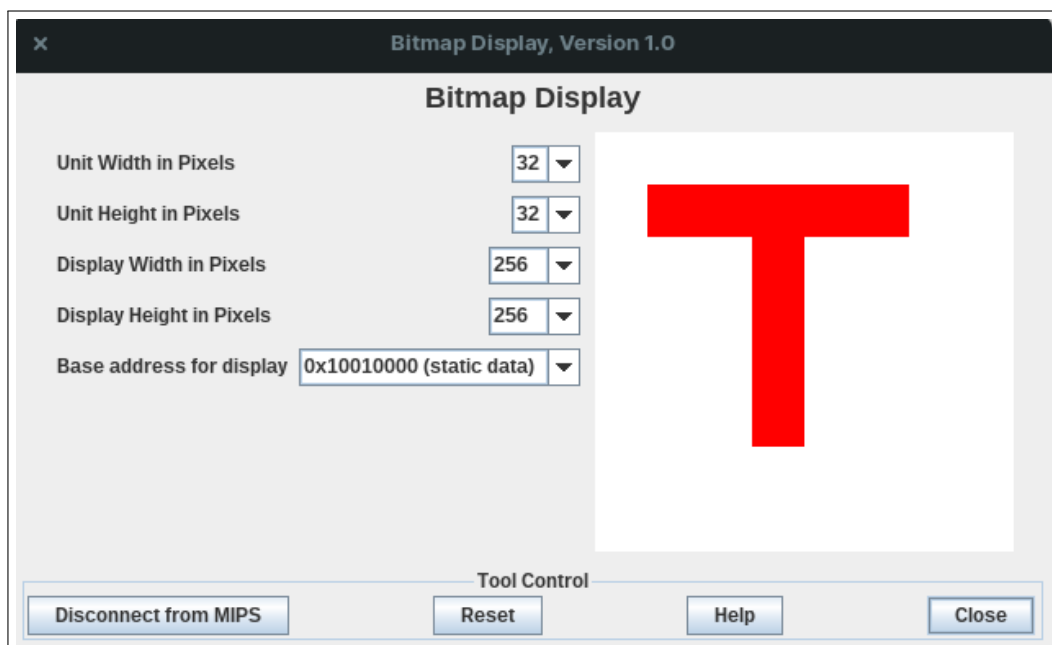
next:
        addi $t1, $t1, 4
        beq $t1, $t3, done
        j loop

done:

```

Hình 6: Vẽ nền trắng

Giải thích: \$t0 lưu giá trị của màu trắng, \$k0 lưu địa chỉ của màn hình. Với giá trị từ \$t1 đến \$t3, nếu địa chỉ tại ô đó chưa được tô trắng, thực hiện lệnh "sw \$t0, 0(\$t1)". Tăng giá trị của thanh ghi \$t1 và kiểm tra điều kiện lặp.



Hình 7: Kết quả của đoạn code