BÁO CÁO TUẦN 10

THỰC HÀNH KIẾN TRÚC MÁY TÍNH

Vũ Tuấn Kiệt

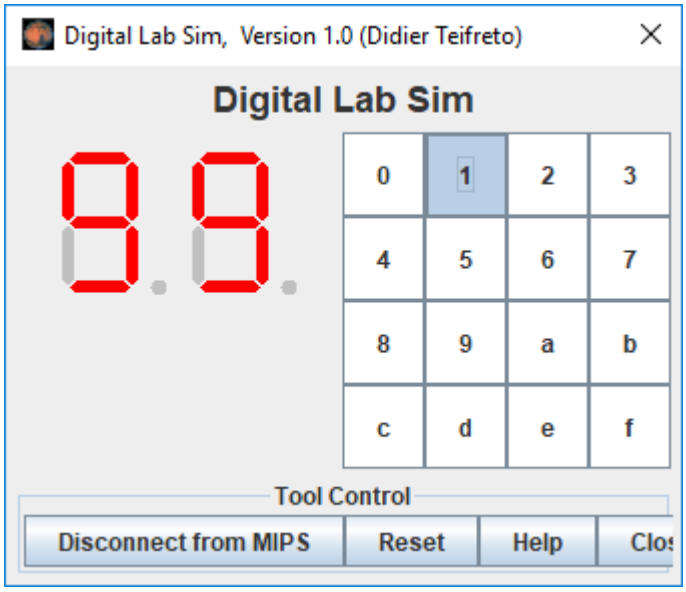
Assignment 1:

Mã nguồn:

```
.eqv  SEVENSEG_RIGHT    0xFFFF0010    # Dia chi cua den led 7 doan trai.


.eqv  SEVENSEG_LEFT   0xFFFF0011       # Dia chi cua den led 7 doan phai

# Mã số sinh viên 20194599

.text

main:

        li      $a0, 111            # giá trị 1 thì tất cả các thanh e là 1(không tính dấu chấm)

                                    # 8 bit cuối thì phải là 01101111

        jal     SHOW_7SEG_LEFT          # show


        li      $a0,  111                   # giá trị 1 thì tất cả các thanh e là 1(không tính dấu chấm)

                                    # 8 bit cuối thì phải là 01101111

        jal     SHOW_7SEG_RIGHT         # show

exit:

        li    $v0, 10

        syscall

endmain:
```

```
SHOW_7SEG_LEFT:
        li   $t0,  SEVENSEG_LEFT        # Gán địa chỉ
        sb   $a0,  0($t0)          # Gán giá trị
        jr   $ra

SHOW_7SEG_RIGHT:
        li   $t0,  SEVENSEG_RIGHT      # Gán địa chỉ
        sb   $a0,  0($t0)          # Gán giá trị
        jr   $ra
```

Kết quả:



Assginment 2:

Mã nguồn:

```
.eqv  SEVENSEG_RIGHT  0xFFFF0010      # Dia chi cua den led 7 doan trai.
```

```
.eqv  SEVENSEG_LEFT   0xFFFF0011       # Dia chi cua den led 7 doan phai


.text
main:
input:
        li      $v0, 5                  # Đọc số nguyên dương
        syscall
        li      $a0, 10
        blt     $v0, $a0, input         # Nếu mà nó nhỏ hơn 10 thì dừng
end_input:
        div     $v0, $a0            # lấy số vừa nhập chia cho 10 để lấy chữ số cuối
        mflo    $v0
        mfhi    $s1
        div     $v0, $v0, $a0           # chia tiếp cho 10 để lấy chữ số gần cuối
        mflo    $v0
        mfhi    $s0
        li      $t0, 0                  # Số để so sánh


        beq     $s0, $t0, set_0
        addi    $t0, $t0, 1
        beq     $s0, $t0, set_1
        addi    $t0, $t0, 1
        beq     $s0, $t0, set_2
```

```
        addi  $t0, $t0, 1
        beq   $s0, $t0, set_3
        addi  $t0, $t0, 1
        beq   $s0, $t0, set_4
        addi  $t0, $t0, 1
        beq   $s0, $t0, set_5
        addi  $t0, $t0, 1
        beq   $s0, $t0, set_6
        addi  $t0, $t0, 1
        beq   $s0, $t0, set_7
        addi  $t0, $t0, 1
        beq   $s0, $t0, set_8
        addi  $t0, $t0, 1
        beq   $s0, $t0, set_9
next:
        jal     SHOW_7SEG_LEFT          # show

        li      $t0, 0
        beq   $s1, $t0, set_01
        addi  $t0, $t0, 1
        beq   $s1, $t0, set_11
        addi  $t0, $t0, 1
        beq   $s1, $t0, set_21
        addi  $t0, $t0, 1
        beq   $s1, $t0, set_31
```

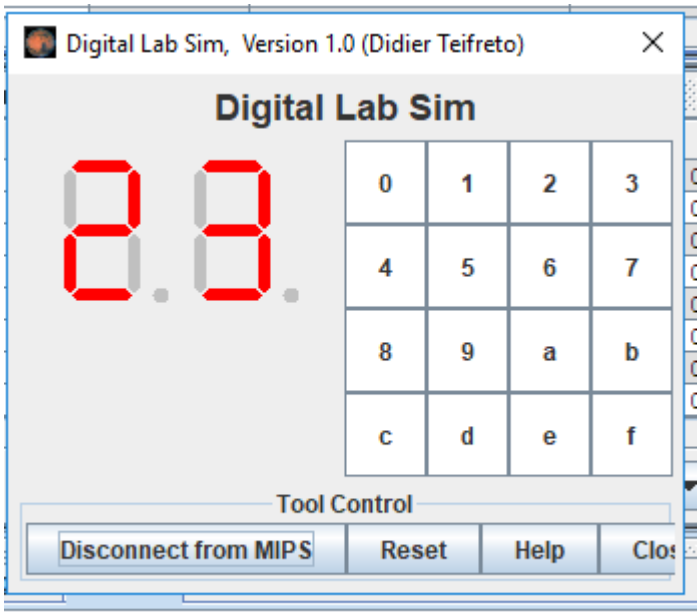
```
        addi  $t0, $t0, 1
        beq   $s1, $t0, set_41
        addi  $t0, $t0, 1
        beq   $s1, $t0, set_51
        addi  $t0, $t0, 1
        beq   $s1, $t0, set_61
        addi  $t1, $t0, 1
        beq   $s0, $t0, set_71
        addi  $t1, $t0, 1
        beq   $s1, $t0, set_81
        addi  $t0, $t0, 1
        beq   $s1, $t0, set_91
next1:
        jal   SHOW_7SEG_RIGHT           # show
        j     exit
# Đặt chỉ số hiển thị cho led
set_0:
        ori   $a0, $0, 0x3f
        j     next
set_1:
        ori   $a0, $0, 0x06
        j     next
set_2:
        ori   $a0, $0, 0x5b
        j     next
```

```
set_3:
        ori     $a0, $0, 0x4f
        j       next
set_4:
        ori     $a0, $0, 0x66
        j       next
set_5:
        ori     $a0, $0, 0x6d
        j       next
set_6:
        ori     $a0, $0, 0x7d
        j       next
set_7:
        ori     $a0, $0, 0x7
        j       next
set_8:
        ori     $a0, $0, 0x7f
        j       next
set_9:
        ori     $a0, $0, 0x6f
        j       next

set_01:
        ori     $a0, $0, 0x3f
        j       next1
```

```
set_11:
        ori     $a0, $0, 0x06
        j       next1
set_21:
        ori     $a0, $0, 0x5b
        j       next1
set_31:
        ori     $a0, $0, 0x4f
        j       next1
set_41:
        ori     $a0, $0, 0x66
        j       next1
set_51:
        ori     $a0, $0, 0x6d
        j       next1
set_61:
        ori     $a0, $0, 0x7d
        j       next1
set_71:
        ori     $a0, $0, 0x7
        j       next1
set_81:
        ori     $a0, $0, 0x7f
        j       next1
set_91:
```

```
        ori     $a0, $0, 0x6f
        j       next1
exit:
        li      $v0, 10
        syscall
endmain:

SHOW_7SEG_LEFT:
        li      $t1,  SEVENSEG_LEFT     # Gán địa chỉ
        sb      $a0,  0($t1)            # Gán giá trị
        jr      $ra

SHOW_7SEG_RIGHT:
        li      $t1,  SEVENSEG_RIGHT  # Gán địa chỉ
        sb      $a0,  0($t1)            # Gán giá trị
        jr      $ra
```

Kết quả:




```
*** user input : 23
```

Assginment 3:

Mã nguồn:

```
.eqv  SEVENSEG_RIGHT  0xFFFF0010      # Dia chi cua den led 7 doan trai.


.eqv  SEVENSEG_LEFT   0xFFFF0011      # Dia chi cua den led 7 doan phai


.text
main:
input:
        li      $v0, 12                 # Đọc ký tự
        syscall
        li      $a0, 10
end_input:
```

```
        div   $v0, $a0          # lấy số vừa nhập chia cho 10 để lấy chữ số cuối
        mflo  $v0
        mfhi  $s1
        div   $v0, $v0, $a0         # chia tiếp cho 10 để lấy chữ số gần cuối
        mflo  $v0
        mfhi  $s0
        li    $t0, 0                # Số để so sánh

        beq   $s0, $t0, set_0
        addi  $t0, $t0, 1
        beq   $s0, $t0, set_1
        addi  $t0, $t0, 1
        beq   $s0, $t0, set_2
        addi  $t0, $t0, 1
        beq   $s0, $t0, set_3
        addi  $t0, $t0, 1
        beq   $s0, $t0, set_4
        addi  $t0, $t0, 1
        beq   $s0, $t0, set_5
        addi  $t0, $t0, 1
        beq   $s0, $t0, set_6
        addi  $t0, $t0, 1
        beq   $s0, $t0, set_7
```

```
        addi  $t0, $t0, 1
        beq   $s0, $t0, set_8
        addi  $t0, $t0, 1
        beq   $s0, $t0, set_9
next:
        jal     SHOW_7SEG_LEFT          # show

        li      $t0, 0
        beq   $s1, $t0, set_01
        addi  $t0, $t0, 1
        beq   $s1, $t0, set_11
        addi  $t0, $t0, 1
        beq   $s1, $t0, set_21
        addi  $t0, $t0, 1
        beq   $s1, $t0, set_31
        addi  $t0, $t0, 1
        beq   $s1, $t0, set_41
        addi  $t0, $t0, 1
        beq   $s1, $t0, set_51
        addi  $t0, $t0, 1
        beq   $s1, $t0, set_61
        addi  $t1, $t0, 1
        beq   $s0, $t0, set_71
        addi  $t1, $t0, 1
        beq   $s1, $t0, set_81
```

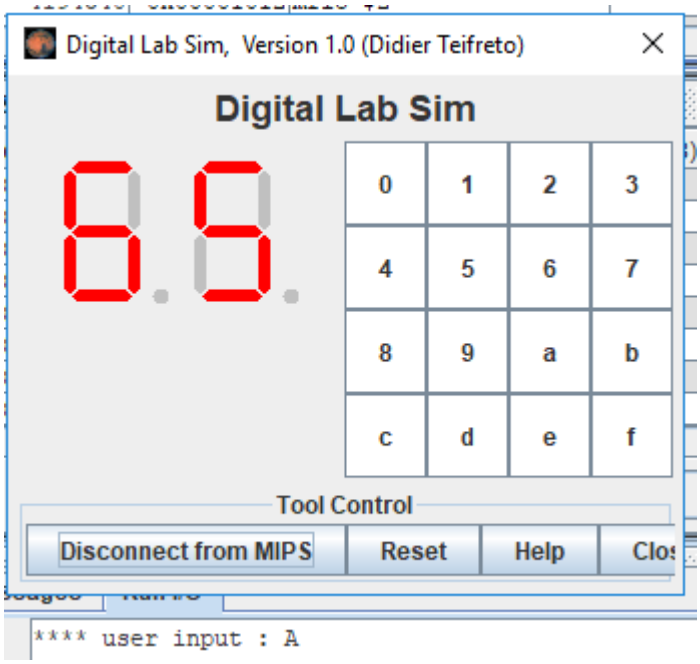
```
        addi  $t0, $t0, 1
        beq   $s1, $t0, set_91
next1:
        jal   SHOW_7SEG_RIGHT           # show
        j     exit
# Đặt chỉ số hiển thị cho led
set_0:
        ori   $a0, $0, 0x3f
        j     next
set_1:
        ori   $a0, $0, 0x06
        j     next
set_2:
        ori   $a0, $0, 0x5b
        j     next
set_3:
        ori   $a0, $0, 0x4f
        j     next
set_4:
        ori   $a0, $0, 0x66
        j     next
set_5:
        ori   $a0, $0, 0x6d
        j     next
set_6:
```

```
        ori     $a0, $0, 0x7d
        j       next
set_7:
        ori     $a0, $0, 0x7
        j       next
set_8:
        ori     $a0, $0, 0x7f
        j       next
set_9:
        ori     $a0, $0, 0x6f
        j       next

set_01:
        ori     $a0, $0, 0x3f
        j       next1
set_11:
        ori     $a0, $0, 0x06
        j       next1
set_21:
        ori     $a0, $0, 0x5b
        j       next1
set_31:
        ori     $a0, $0, 0x4f
        j       next1
set_41:
```

```
        ori     $a0, $0, 0x66
        j       next1
set_51:
        ori     $a0, $0, 0x6d
        j       next1
set_61:
        ori     $a0, $0, 0x7d
        j       next1
set_71:
        ori     $a0, $0, 0x7
        j       next1
set_81:
        ori     $a0, $0, 0x7f
        j       next1
set_91:
        ori     $a0, $0, 0x6f
        j       next1
exit:
        li      $v0, 10
        syscall
endmain:

SHOW_7SEG_LEFT:
        li      $t1,  SEVENSEG_LEFT     # Gán địa chỉ
        sb      $a0,  0($t1)            # Gán giá trị
```

```
        jr      $ra


SHOW_7SEG_RIGHT:
        li      $t1,  SEVENSEG_RIGHT  # Gán địa chỉ
        sb      $a0,  0($t1)          # Gán giá trị
        jr      $ra
```

Kết quả:



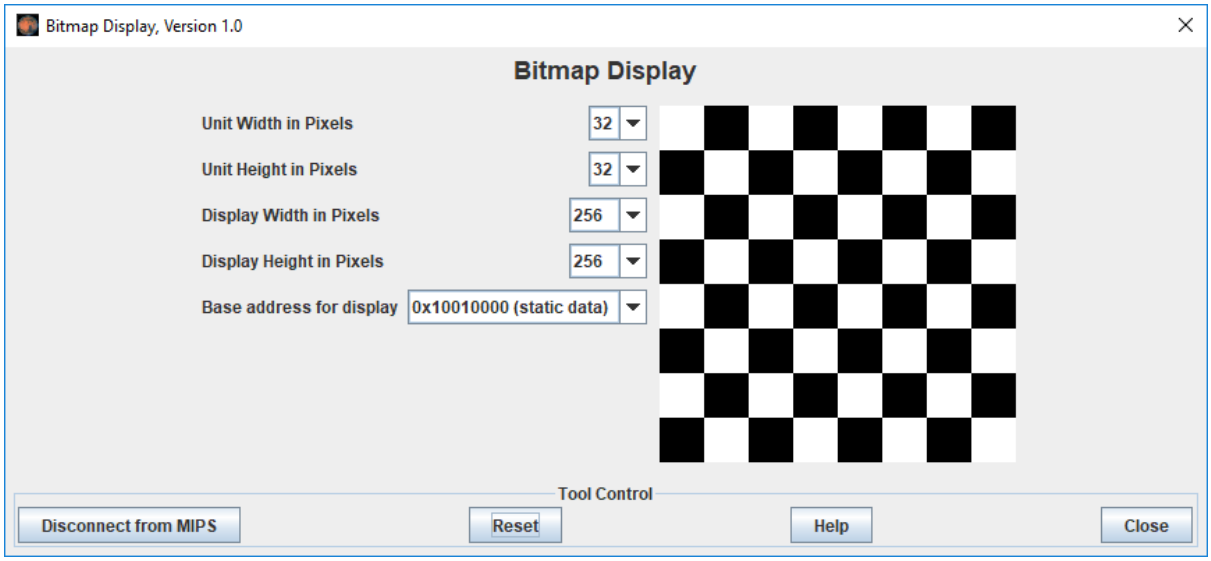
Assginment 4:

Mã nguồn:

```
.eqv  MONITOR_SCREEN        0x10010000
.eqv  WHITE             0x00FFFFFF
.text
        li      $k0, MONITOR_SCREEN
        li      $s0, WHITE
        li      $t0, 0                  # biến để đếm hàng
        li      $t2, 8                  # 8 hàng 8 cột
```

```
        li      $t3, 2
loop1:
        beq     $t0, $t2, exit              # hàng >= 8 thì dừng
        li      $t1, 0                      # biến để đếm cột
        div     $t0, $t3            # kiểm tra xem hàng có chẵn
        mfhi    $t4
loop2:
        beq     $t1, $t2, endloop2      # cột >= 8 thì đến hàng tiếp theo
        div     $t1, $t3            # kiểm tra xem cột có chẵn
        mfhi    $t5
        xor     $t6, $t4, $t5
        beq     $t6, $0, print
next:
        addi    $t1, $t1, 1
        addi    $k0, $k0, 4
        j       loop2
endloop2:
        addi    $t0, $t0, 1
        j       loop1
print:
        sw      $s0, 0($k0)
        j       next
exit:
```

Kết quả:



Assignment 5:

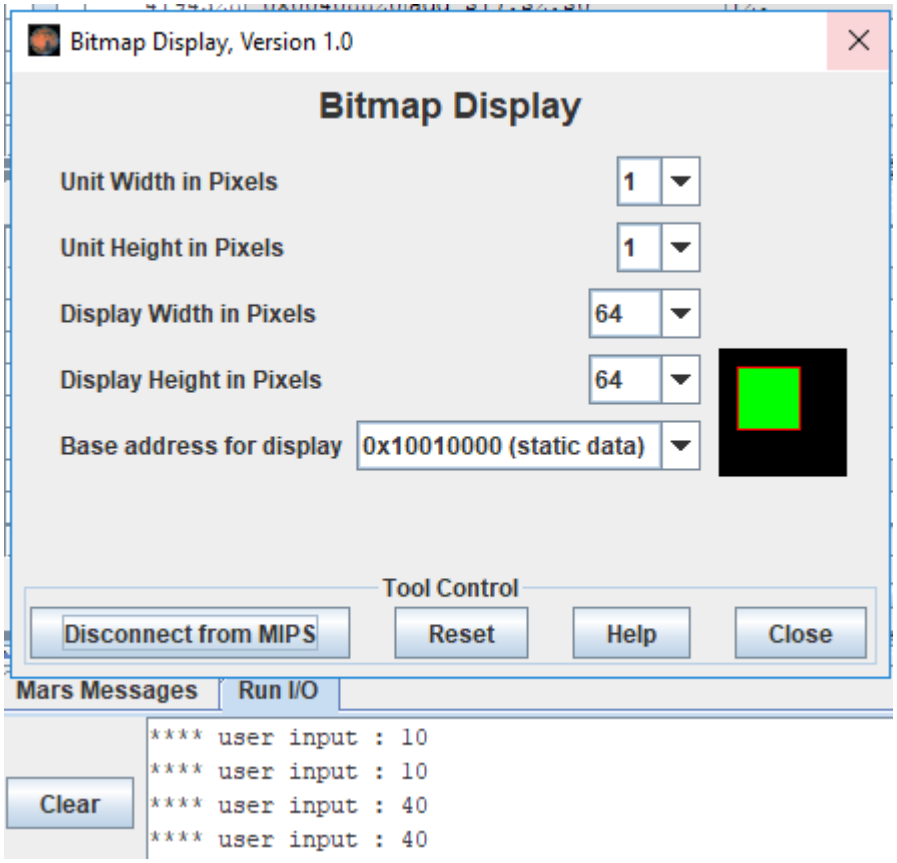
Mã nguồn:

```
.eqv  MONITOR_SCREEN        0x10010000
.eqv  RED          0x00FF0000
.eqv  GREEN              0x0000FF00


.text
        # x1 < x2   y1 < y2
        li      $k0, MONITOR_SCREEN
        li      $s0, RED

        li      $v0, 5
        syscall
        add   $s1, $v0, $0              # x1

        li      $v0, 5
        syscall
```

```
        add   $s2, $v0, $0              # y1

        li      $v0, 5
        syscall
        add   $s3, $v0, $0              # x2

        li      $v0, 5
        syscall
        add   $s4, $v0, $0              # y2

        addi  $t0, $s2, -1              # biến đếm hàng
loop11:
        bgt   $t0, $s4, next
        addi  $t1, $s1, -1              # biến đếm cột
loop21:
        bgt   $t1, $s3, endloop21
        jal     print
        addi  $t1, $t1, 1
        j       loop21
endloop21:
        addi  $t0, $t0, 1
        j       loop11
next:

        li      $s0, GREEN
```

```
        add   $t0, $0, $s2              # biến đếm hàng
loop1:
        beq   $t0, $s4, exit
        add   $t1, $0, $s1              # biến đếm cột
loop2:
        beq   $t1, $s3, endloop2
        jal   print
        addi  $t1, $t1, 1
        j     loop2
endloop2:
        addi  $t0, $t0, 1
        j     loop1
print:
        addi  $t2, $0,  64
        mul   $t2, $t2, $t0
        add   $t2, $t2, $t1
        sll   $t2, $t2, 2
        add   $t2, $t2, $k0
        sw    $s0, 0($t2)
        jr    $ra
exit:
```

Kết quả:


Bitmap Display, Version 1.0
Bitmap Display
Unit Width in Pixels
1
Unit Height in Pixels
1
Display Width in Pixels
64
Display Height in Pixels
64
Base address for display
0x10010000 (static data)
Tool Control
Disconnect from MIPS
Reset
Help
Close
Mars Messages
Run I/O
Clear
**** user input : 10
**** user input : 10
**** user input : 40
**** user input : 40