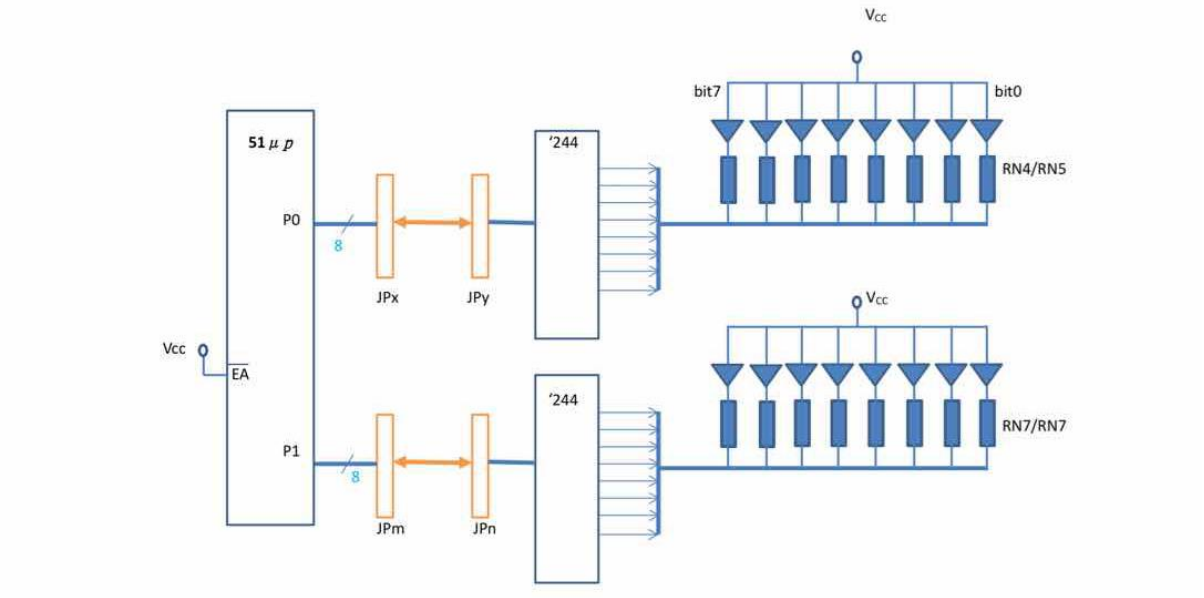
Microprocessor Lab 2 Report

0416106彭敬樺
0416109周才錢

Subject and Goal:

This lab is about using µ-Vision 51IDE residing on MegaWin82G516 to:

- Access every individual LED for ON/OFF control in the 2 sets of discrete LED modules.
- Organize display patterns in static or dynamic form can be achieved as required.



Preparations:

- Power cable and required connection from the output to the led input is established. The on/off of led is controlled by the output of port P0 and P2 that is connected to LED wiring port.
- Check the correctness and check if there are any defective on the board by test running Lab 1 program.

Operating Procedure:

- Jumper-wiring for circuit setup
- Code preparation
- Task execution:
    - Start IDE51 emulation,
    - Start execution and troubleshooting if necessary.

Code Preparation:

```
        org     0
        mov     sp, #50H
        clr     c
        mov     a, #0feH
        mov     R7, a

        mov     a, #0fH
mk1:
        cpl     a
        mov     r6, a
        mov     p1, A
        mov     a, r7
        mov     p0, a
        call    delay
        rlc     a
        mov     r7, a
        mov     a, r6
        jc      mk1
```

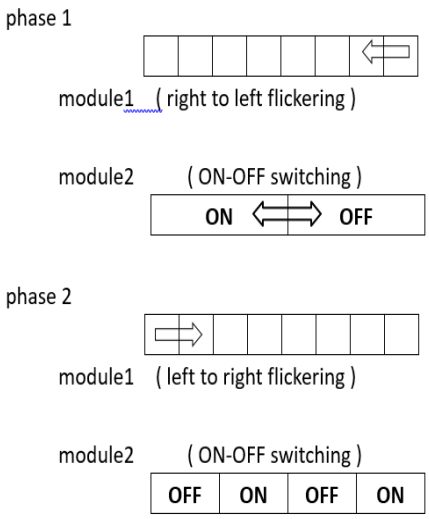
```
        mov     a, #0ccH
mk2:
        cpl     a
        mov     r6, a
        mov     p1, a
        mov     a, r7
        mov     p0, a
        call    delay
        rrc     a
        mov     r7,a
        mov     a, r6
        jc      mk2

        mov     a, #0f0H
mk3:
        cpl     a
        mov     r6, a
        mov     p0, a
        mov     a, r7
```

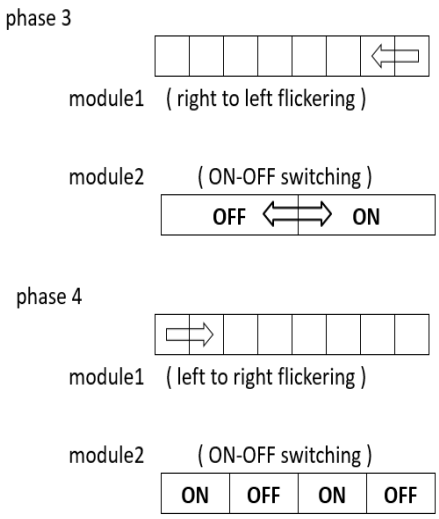
```
        mov     p1, a
        call    delay
        rlc     a
        mov     r7, a
        mov     a, r6
        jc      mk3

        mov     a, #0ccH
mk4:
        cpl     a       ; XXX
        mov     r6, a
        mov     p0, a
        mov     a, r7
        mov     p1, a
        call    delay
        rrc     a
        mov     r7,a
        mov     a, r6
        jc      mk4
```

```
        mov     a, #0fH
        jmp     mk1
delay:
        push    5
            ; push R5???
        push    6
        push    7
        mov     r5, #2
dd1:
        mov     r6, #200
dd2:
        mov     r7, #250
        djnz    r7, $
        djnz    r6, dd2
        djnz    r5, dd1
        pop     7
        pop     6
        pop     5
        ret
```

Observation:

- The initial code provided did not result in the wanted light phase in depicted in the graph below





- The corrected code below will be the one used in the demonstration and has fulfilled the goals of the requires LED phase pattern.

- The changed code below has no more bug or warning message popup. Therefore, lab 2 has been completed and the process has not been a harsh path due to sufficient preparation from reading material prepared before the start of the lab.

revised code:

```
org 0
mov sp, #50H
clr c
mov a, #0feH
mov R7, a
mov a, #0fH
mk1:
cpl a
mov r6, a
mov p2, A
mov a, r7
mov p0, a
call delay
rlc a
mov r7, a
mov a, r6
jc mk1

mov a, #0ccH
mk2:
cpl a
mov r6, a
mov p2, a
mov a, r7
mov p0, a
call delay
rrc a
mov r7,a
mov a, r6
jc mk2

mov a, #0f0H
mk3:
cpl a
mov r6, a
mov p2, a
mov a, r7
mov p0, a
call delay
rrc a
mov r7,a
mov a, r6
jc mk3

mov a, #033H
mk4:
cpl a
mov r6, a
mov p2, a
mov a, r7
mov p0, a
call delay
rrc a
mov r7,a
mov a, r6
jc mk4
mov a, #0fH
jmp mk1
delay:
push 5
; push R5???
push 6
push 7
mov r5, #15
dd1:
mov r6, #200
dd2:
mov r7, #250
djnz r7, $
djnz r6, dd2
djnz r5, dd1
pop 7
pop 6
pop 5
ret
end
```

Comprehensive evaluation:

- The stack pointer is initialized to address of 50H, which has pass through the main written code. The content of the stack will be assigned when the push function is called and the value of stack pointer will be increased by 1. In "delay" function, after 3 push function is called, which change the value of address 50H into content of R5, address 51H into content of R6, and address 52H into R7. The final value of stack pointer after this is 53H. Before exiting the "delay" function 3 pop function is called and the value of stack pointer will be 50H after the operation, but the value of each address has not been change
- The instruction line that is currently running when observed can be determined. Information needed is to calculate the phase it is currently in to determine which loop it is in. Next, by observing the location of flickering LED, we can determine the value of each register due to the fact that each light condition has a specific register value to activate at a determined phase
- The instruction line marked by XX can have huge impact towards the led behavior. If it is removed, the led behavior for the wiring connected to P2 (in revised code) for phase 4 will change. The correspond LED will not change from the initial condition in the phase 4, but will change to normal after getting to phase 1.