

PRACTICE EXERCISES OF THE MICROPROCESSORS & MICROCONTROLLERS

Instructor: The Tung Than

Student's name: Nguyen Quoc Truong An

Student code: 21521810

PRACTICE REPORT NO 2

LAB2: COMMUNICATION WITH 7-SEGMENT LED AND TIMER

I. Student preparation

- Knowledge of Timer configuration.
- Learn how to use 7-segment led (7-segment light).

II. Practice content

- 1) Present and draw a flowchart of the LED scanning algorithm applied to display 7-segment led. **(2 points)**

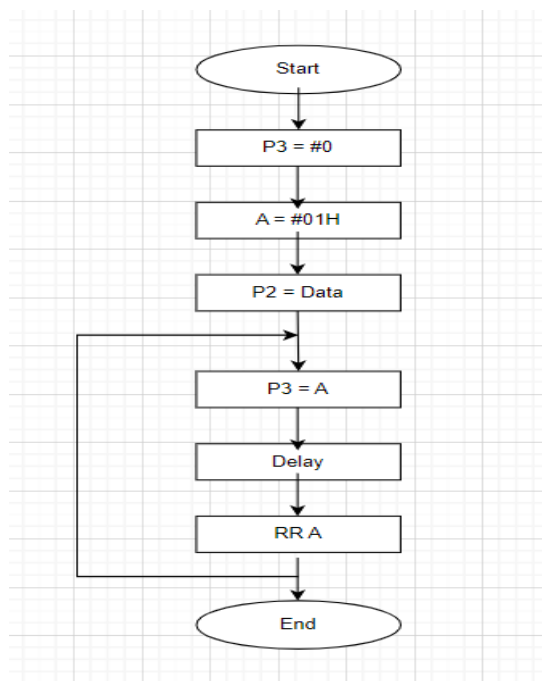
-Thuật toán quét led là giải thuật bật tắt led lần lượt trong khoảng thời gian rất ngắn (thông thường với tốc độ > 25 Hz) sao cho mắt người không nhận ra sự thay đổi.

-Có 2 cách phổ biến để quét led:

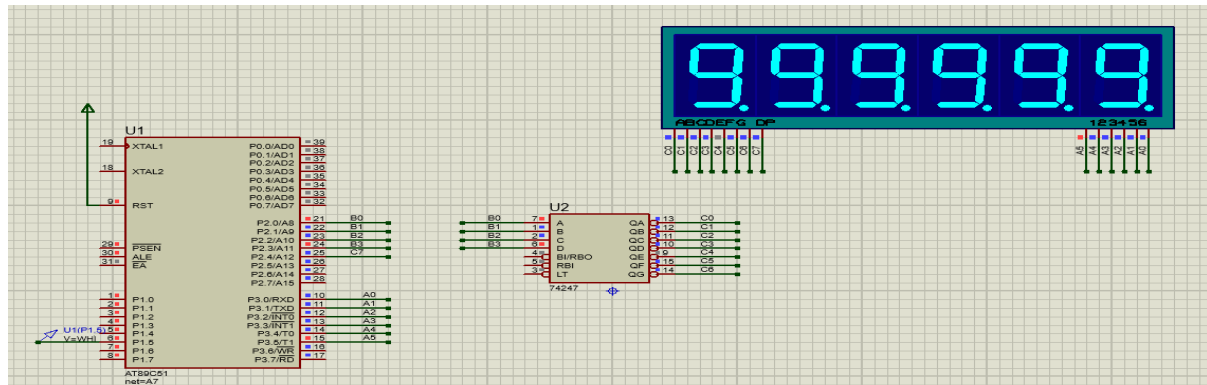
+Sử dụng lệnh SETB và CLR để bật tắt các led.

+Sử dụng lệnh RR để chuyển trạng thái bật tắt.

-Lưu đồ giải thuật quét led trên port P3 để hiển thị 7-segment led và port dữ liệu P2, IC giải mã 74247:



- Để led sáng lần lượt, cần làm cho tín hiệu xuất ra giá trị 1 một khoảng thời gian để có thể nhận biết được. Đầu tiên cho P3 tắt, sau đó làm cho bit A.0 lên 1, sử dụng lệnh xoay phải dữ liệu trên thanh ghi A, mỗi lần xoay giá trị 1 sẽ chuyển lần lượt qua A.1 - A.2 - A.3 - A.4 - A.5 - A.6 - A.7 - A.0, mỗi lần xoay xuất tín hiệu ra P3 sẽ thấy led sáng lần lượt từ led 1 đến led 8.



```

$NOMOD51
$INCLUDE (8051.MCU)

; Reset Vector
ORG 0000H
JMP START

ORG 0100H
START:
    MOV TMOD, #10H          ; Timer 1 chế độ 1
    MOV TL1, #78H           ; Nạp giá trị cho Timer 1
    MOV TH1, #0ECH
    SETB TR1                ; Bật Timer 1
    MOV R0, #9              ; Gán R0 = 9 (R0 sẽ là giá trị hiển thị của LED)

LOOP:
    CALL DISPLAY             ; Gọi hàm để hiển thị LED
    JMP LOOP

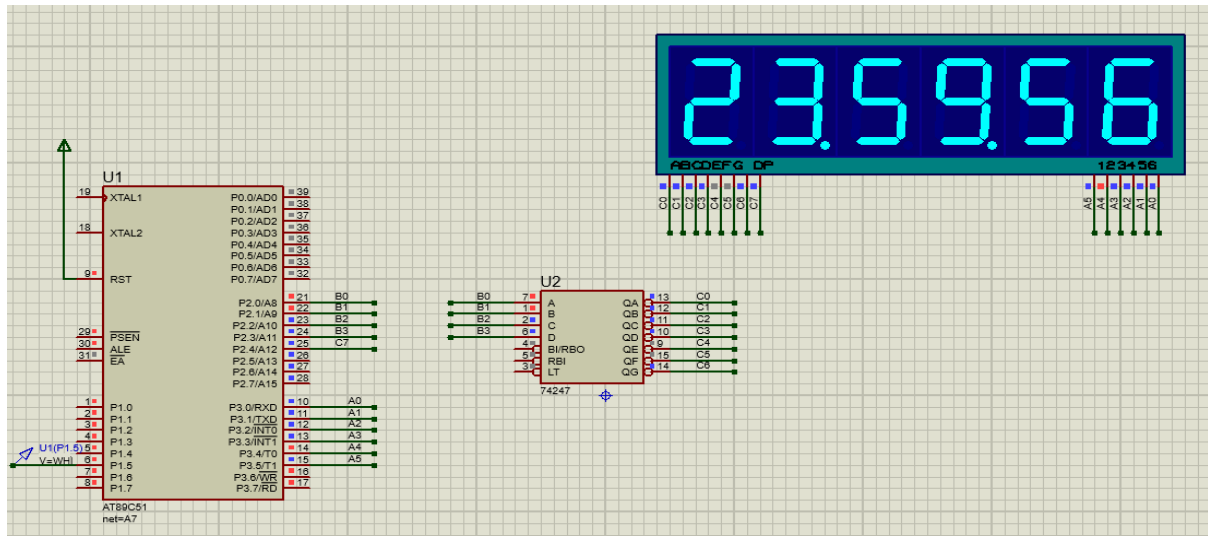
DISPLAY:
    MOV A, #01H             ; Gán A = 0x01 (0b0000.0001)
    HT1:
        MOV P3, A           ; Gán P3 = A (Bật LED ở vị trí bit 1 tương ứng trong A)
        MOV P2, R0
        CALL Delay_5ms      ; Delay 5ms để mắt người đủ thấy giá trị LED
        RR A                ; Xoay phải A để đổi vị trí LED sáng
        CJNE A, #01H, HT1   ; Lặp lại việc hiển thị
    RET
RET

DELAY_5MS:
    ; Hàm delay 5ms bằng Timer 1
    MOV TL1, #78H
    MOV TH1, #0ECH
    SETB TR1
    JNB TF1, $
RET

END

```

- 2) Using the 8051 microcontroller's Timer, design a clock circuit with 24h format with the initial time set in the source code. (2 points)



```

$NOMOD51
$INCLUDE (8051.MCU)

; Reset Vector
ORG 0000H
JMP START

ORG 001BH           ; Interrupt vector của Timer1
LJMP ISR_TIMER1

ORG 0100H
START:
    MOV TMOD, #10H   ; Timer1 che do 1
    MOV TL1, #78H    ; Nap gia tri delay 5ms cho Timer1
    MOV TH1, #0ECH
    MOV IE, #88H     ; Bat enable cho Interrupt của Timer1
    SETB TR1         ; Bat Timer1
                    ; Khoi tao gia tri ban dau của dong ho
    MOV R0, #55      ; R0 luu gia tri Giay
    MOV R1, #59      ; R1 luu gia tri Phut
    MOV R2, #23      ; R2 luu gia tri Gio
    MOV R3, #200     ; So lan lap Delay de tang gia tri Giay, Delay 5ms => 200*5ms = 1s

LOOP:
    CALL DISPLAY     ; Goi ham hien thi
    JMP LOOP

DISPLAY:
    SETB P3.0        ; Bat LED hien thi don vi Giay o pin P3.0
    MOV A, R0
    MOV B, #10
    DIV AB            ; Chia lay du de tach hang don vi của Giay
    MOV P2, B         ; Gan P2 = B, tuc la hien thi hang don vi của Giay
    SETB P2.4        ; Xoa dau cham o LED don vi của Giay
    CALL DELAY_5MS    ; Delay 5ms
    CLR P3.0         ; Tat LED o pin P3.0

```

SETB P3.1 ; Bat LED hien thi hang chuc cua Giay o pin P3.1

MOV A, R0

MOV B, #10

DIV AB ; Chia lay nguyen de tach hang don vi cua Giay

MOV P2, A ; Gan P2 = A, tuc la hien thi hang chuc cua Giay

SETB P2.4 ; Xoa dau cham o LED don vi cua Giay

CALL DELAY_5MS ; Delay 5ms

CLR P3.1 ; Tat LED o pin P3.1

SETB P3.2 ; Xu ly don vi Phut, tuong tu nhu xu ly don vi Giay phia tren

MOV A, R1 ; Xu ly hang don vi Phut

MOV B, #10

DIV AB

MOV P2, B

CLR P2.4

CALL DELAY_5MS

CLR P3.2

SETB P3.3 ; Xu ly hang chuc Phut

MOV A, R1

MOV B, #10

DIV AB

MOV P2, A

SETB P2.4

CALL DELAY_5MS

CLR P3.3

SETB P3.4 ; Xu ly don vi Gio, tuong tu nhu xu ly don vi Giay phia tren

MOV A, R2 ; Xu ly hang don vi Gio

MOV B, #10

DIV AB

MOV P2, B

CLR P2.4

CALL DELAY_5MS

CLR P3.4

SETB P3.5 ; Xu ly hang chuc Gio

MOV A, R2

MOV B, #10

DIV AB

MOV P2, A

SETB P2.4

CALL DELAY_5MS

CLR P3.5

RET

DELAY_5MS: ; Ham delay 5ms

MOV TL1, #78H

MOV TH1, #0ECH

SETB TR1

JNB TF1, \$; Khong can phai xoa co TF1 vi khi Timer1 tran ham xu ly ngat se tu xoa bang phan cung

RET

ISR_TIMER1: ; Ham xu ly khi Timer1 tran, ; XU LY DIGITAL CLOCK

DJNZ R3, NOT_EQUAL ; Kiem tra delay du 200*5ms = 1s de tang Giay, neu khong bang thi thoat ham xu ly ngat

MOV R3, #200 ; Nap lai so lan lap delay 5ms

INC R0 ; Tang Giay

MOV A, R0 ; Gan A = Giay

MOV B, #60 ; Gan B = 60

CJNE A, B, NOT_EQUAL ; Kiem tra Giay = 60?, neu khong bang thi thoat ham xu ly ngat

MOV R0, #0 ; Neu Giay = 60 => reset Giay = 0 va tang Phut

INC R1 ; Tang Phut

```

MOV A, R1          ; Gan A = Phut
MOV B, #60         ; Gan B = 60
CJNE A, B, NOT_EQUAL ; Kiem tra Phut = 60?, neu khong bang thi thoat ham xu ly ngat
MOV R1, #0         ; Neu Phut = 60 => reset Phut = 0 va tang Gio
INC R2            ; Tang Gio

MOV A, R2          ; Gan A = Gio
MOV B, #24         ; Gan B = 24
CJNE A, B, NOT_EQUAL ; Kiem tra Gio = 24?, neu khong bang thi thoat ham xu ly ngat
MOV R2, #0         ; Neu Gio = 24 => reset Gio = 0

NOT_EQUAL:        ; Thoat ham xu ly ngat

RETI

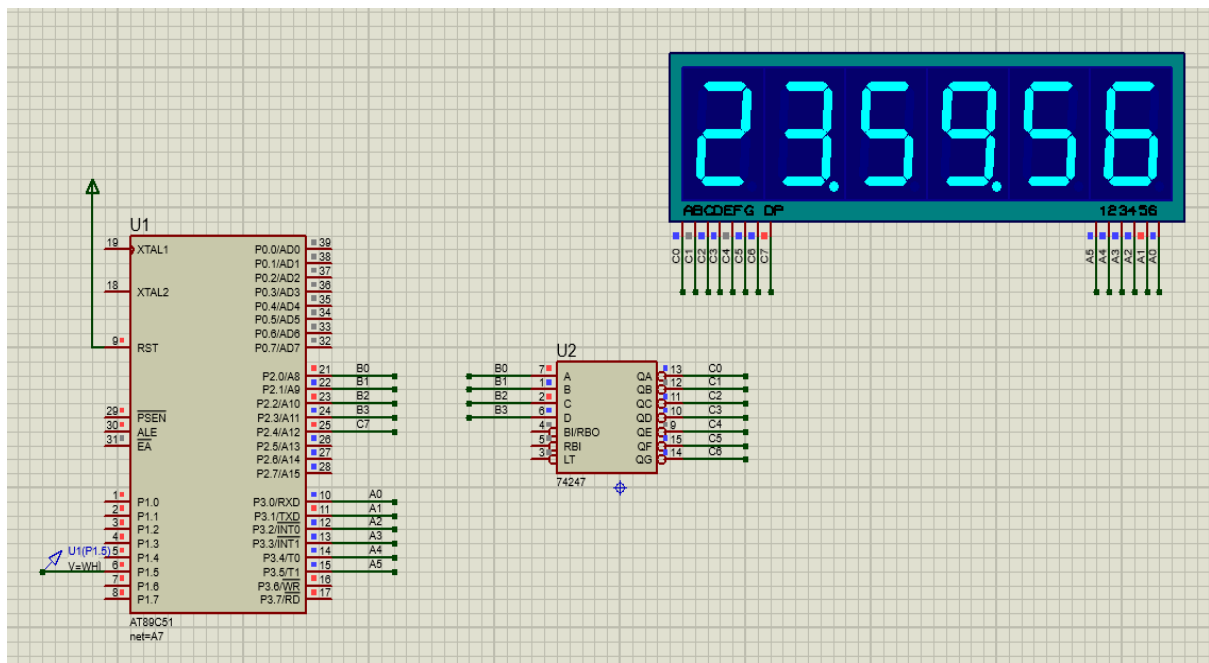
END

```

III. Exercises

With the above clock design, use a loop to create a delay instead of Timer. State the advantages and disadvantages of the two methods.

-Đồng hồ sử dụng hàm Delay bằng vòng lặp:



```

$NOMOD51
$INCLUDE (8051.MCU)

; Reset Vector
ORG 0000H
JMP START

ORG 0100H
START:
    MOV TMOD, #10H          ; Timer1 che do 1
    MOV TL1, #78H           ; Nap gia tri delay 5ms cho Timer1
    MOV TH1, #0ECH
    SETB TR1                ; Bat Timer1
                               ; Khoi tao gia tri ban dau cua dong ho

    MOV R0, #55              ; R0 luu gia tri Giay
    MOV R1, #59              ; R1 luu gia tri Phut
    MOV R2, #23              ; R2 luu gia tri Gio
    MOV R3, #33              ; Xac dinh so lan lap de tao ra 1s de xac dinh thoi diem tang gia tri Giay
                               ; Mot vong lap DISPLAY delay 6*5ms => 33 * 6*5ms ~ 1s

LOOP:
                               ; Thuc hien hien thi Thoi Gian
    DJNZ R3, NOT_INC_SECOND  ; Kiem tra xem da lap du Delay khoang 1s chua, neu chua KHONG tang Giay
    CALL CLOCK                ; Neu da Delay du 1s, Goi ham CLOCK de tinh toan Thoi Gian
    NOT_INC_SECOND:
    CALL DISPLAY               ; Hien thi Thoi Gian

JMP LOOP

DISPLAY:
    SETB P3.0                 ; Bat LED hien thi don vi Giay o pin P3.0
    MOV A, R0
    MOV B, #10
    DIV AB                     ; Chia lay du de tach hang don vi cua Giay
    MOV P2, B                  ; Gan P2 = B, tuc la hien thi hang don vi cua Giay
    SETB P2.4                  ; Xoa dau cham o LED don vi cua Giay
    CALL DELAY_5MS             ; Delay 5ms
    CLR P3.0                   ; Tat LED o pin P3.0

    SETB P3.1                  ; Bat LED hien thi hang chuc cua Giay o pin P3.1
    MOV A, R0
    MOV B, #10
    DIV AB                     ; Chia lay nguyen de tach hang don vi cua Giay
    MOV P2, A                  ; Gan P2 = A, tuc la hien thi hang chuc cua Giay
    SETB P2.4                  ; Xoa dau cham o LED don vi cua Giay
    CALL DELAY_5MS             ; Delay 5ms
    CLR P3.1                   ; Tat LED o pin P3.1

    SETB P3.2                  ; Xu ly don vi Phut, tuong tu nhu xu ly don vi Giay phia tren
    MOV A, R1
    MOV B, #10
    DIV AB
    MOV P2, B
    CLR P2.4
    CALL DELAY_5MS
    CLR P3.2

    SETB P3.3                  ; Xu ly hang chuc Phut
    MOV A, R1
    MOV B, #10
    DIV AB
    MOV P2, A

```

```
SETB P2.4
CALL DELAY_5MS
CLR P3.3
```

```
SETB P3.4 ; Xu ly don vi Gio, tuong tu nhu xu ly don vi Giay phia tren
```

```
MOV A, R2 ; Xu ly hang don vi Gio
```

```
MOV B, #10
```

```
DIV AB
```

```
MOV P2, B
```

```
CLR P2.4
```

```
CALL DELAY_5MS
```

```
CLR P3.4
```

```
SETB P3.5 ; Xu ly hang chuc Gio
```

```
MOV A, R2
```

```
MOV B, #10
```

```
DIV AB
```

```
MOV P2, A
```

```
SETB P2.4
```

```
CALL DELAY_5MS
```

```
CLR P3.5
```

```
RET
```

```
DELAY_5MS: ; Ham delay 5ms xu dung vong lap
MOV R4, #10 ; Thoi gian Delay = 10 * 250 * 2us = 5ms
```

```
DELAY:
```

```
MOV R5, #250
```

```
DJNZ R5, $
```

```
DJNZ R4, DELAY
```

```
RET
```

```
CLOCK:
```

```
MOV R3, #33 ; Nap lai gia tri lap trong khoang 1s
```

```
INC R0 ; Tang Giay
```

```
MOV A, R0 ; Gan A = Giay
```

```
MOV B, #60 ; Gan B = 60
```

```
CJNE A, B, NOT_EQUAL ; Kiem tra Giay = 60?, neu khong bang thi thoat ham xu ly ngat
```

```
MOV R0, #0 ; Neu Giay = 60 => reset Giay = 0 va tang Phut
```

```
INC R1 ; Tang Phut
```

```
MOV A, R1 ; Gan A = Phut
```

```
MOV B, #60 ; Gan B = 60
```

```
CJNE A, B, NOT_EQUAL ; Kiem tra Phut = 60?, neu khong bang thi thoat ham xu ly ngat
```

```
MOV R1, #0 ; Neu Phut = 60 => reset Phut = 0 va tang Gio
```

```
INC R2 ; Tang Gio
```

```
MOV A, R2 ; Gan A = Gio
```

```
MOV B, #24 ; Gan B = 24
```

```
CJNE A, B, NOT_EQUAL ; Kiem tra Gio = 24?, neu khong bang thi thoat ham xu ly ngat
```

```
MOV R2, #0 ; Neu Gio = 24 => reset Gio = 0
```

```
NOT_EQUAL: ; Thoat ham xu ly ngat
```

```
RET
```

```
END
```

****So sánh điểm mạnh điểm yếu của 2 phương pháp:**

-Sử dụng Interrupt và Định thời của Timer

+**Điểm mạnh:** Giảm tải công việc cho vi xử lý bằng việc sử dụng module riêng biệt để tạo Delay. Đạt độ chính xác cao về thời gian Delay và giảm tiêu tốn tài nguyên hệ thống.

+**Điểm yếu:** Phức tạp hơn phương pháp dùng vòng lặp trong việc cài đặt và sử dụng.

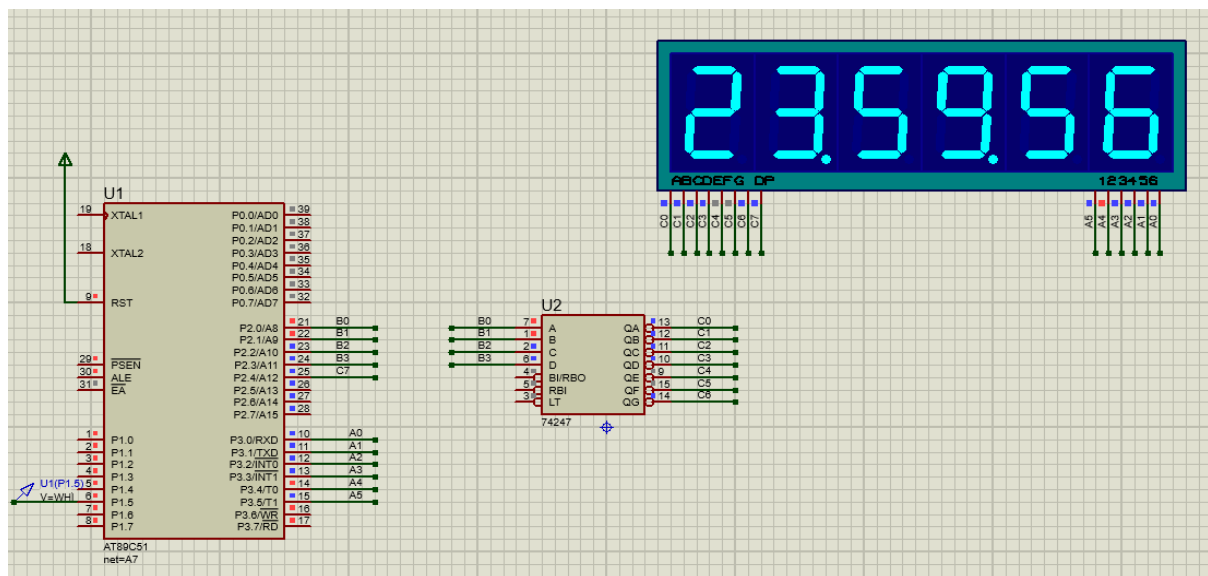
-Sử dụng Delay bằng vòng lặp:

+**Điểm mạnh:** Đơn giản, dễ hiểu, dễ thực hiện cài đặt.

+**Điểm yếu:** Tốn nhiều tài nguyên hơn, độ chính xác thấp, không phù hợp dùng cho hệ thống thực hiện nhiều tác vụ trong một lúc.

IV. Report

1) Design result (screenshot and pasted in the report). (1 point)



2) Explain the operating principle of the effects, accompanied by a video (send a Google Drive link) to demonstrate the circuit operation in case the instructor cannot run the design file. (3 points)

*** Google Drive link:**

https://drive.google.com/drive/folders/1U6AUY_03FUteWkzbA22Wr-ICrMxsmY3H?usp=sharing

****Giải thích code:**

-Ý tưởng tổng quan để giải quyết bài toán: Sử dụng 3 thanh ghi lưu trữ Giây, Phút, Giờ. Tạo Delay 5ms bằng Timer1 chế độ 1 và có sử dụng Ngắt (Interrupt) của bộ định thời Timer1. Khi Timer1 tràn, sự kiện ngắt được thực thi, đồng thời biến đếm số lần Delay 5ms được xét đến, nếu biến đếm cho biết đã Delay 5ms 200 lần ($200 * 5ms = 1s$) thì thực hiện tăng giá trị

Giây. Sau khi tăng giá trị Giây thì kiểm tra, nếu Giây = 60 thì thực hiện tăng giá trị Phút và reset Giây về 0. Tiếp tục kiểm tra Phút, nếu Phút = 60 thì thực hiện tăng giá trị Giờ đồng thời reset Phút về 0. Sau cùng kiểm tra giá trị Giờ, nếu Giờ = 24 thì thực hiện reset Giờ về 0. Ngoài ra, hàm hiển thị DISPLAY luôn được gọi để cập nhật và hiển thị giá trị của Giây, Phút, Giờ lên Led 7-Segments. Hệ thống đồng hồ có sử dụng thêm giải mã IC74247 và 6 Led 7-Segments CA (Common Anot) để hỗ trợ hiển thị thời gian.

*Giải thích chi tiết code:

-Phần khai báo và khởi tạo ban đầu:

```

$NOMOD51
$INCLUDE (8051.MCU)

; Reset Vector
ORG 0000H
JMP START

ORG 001BH          ; Interrupt vector của Timer1
LJMP ISR_TIMER1

ORG 0100H
START:
    MOV TMOD, #10H ; Timer1 chế độ 1
    MOV TL1, #78H  ; Nạp giá trị delay 5ms cho Timer1
    MOV TH1, #0ECH
    MOV IE, #88H   ; Bật enable cho Interrupt của Timer1
    SETB TR1       ; Bật Timer1
                    ; Khởi tạo giá trị ban đầu của đồng hồ
    MOV R0, #55    ; R0 lưu giá trị Giây
    MOV R1, #59    ; R1 lưu giá trị Phút
    MOV R2, #23    ; R2 lưu giá trị Giờ
    MOV R3, #200   ; Số lần lặp Delay để tăng giá trị Giây, Delay 5ms => 200*5ms = 1s

```

+ Ta khai báo vector của Interrupt của Timer1 (ISR_TIMER1) tại vị trí vùng nhớ 0x001B.

+ Chọn sử dụng Timer1 ở chế độ 1. Giá trị cần delay là 5ms, tần số bộ thạch anh là 12MHz => tần số clock = 1MHz => chu kỳ clock là 1us. Vậy giá trị cần nạp cho Timer1 là: $2^{16} - 5 \cdot 10^{-3} / 10^{-6} = 60536 = 0xEC78$.

+ Gán IE = 0x88 => bật chế độ ngắt của bộ định thời Timer1 và bật bit TR1 để khởi động bộ định thời Timer1.

+ Ta khởi tạo giá trị ban đầu của đồng hồ là 23:59:55 lần lượt nạp vào R2, R1, R0.

+ Vì hàm delay 5ms nên cần delay 200 lần mới thực hiện tăng giá trị Giây chứa trong R0 => ta khởi tạo R3 = 200 để kiểm tra đã delay đủ 1s chưa.

-Phần vòng lặp:

```
LOOP:
    CALL DISPLAY      ; Goi ham hien thi
    JMP LOOP
```

+Trong phần này ta gọi hàm hiển thị ‘DISPLAY’ để hiển thị liên tục giá trị của đồng hồ.

-Phần hàm hiển thị (DISPLAY):

```
DISPLAY:
    SETB P3.0          ; Bat LED hien thi don vi Giay o pin P3.0
    MOV A, R0
    MOV B, #10
    DIV AB             ; Chia lay du de tach hang don vi cua Giay
    MOV P2, B          ; Gan P2 = B, tuc la hien thi hang don vi cua Giay
    SETB P2.4          ; Xoa dau cham o LED don vi cua Giay
    CALL DELAY_5MS     ; Delay 5ms
    CLR P3.0           ; Tat LED o pin P3.0
    SETB P3.1          ; Bat LED hien thi hang chuc cua Giay o pin P3.1
    MOV A, R0
    MOV B, #10
    DIV AB             ; Chia lay nguyen de tach hang don vi cua Giay
    MOV P2, A          ; Gan P2 = A, tuc la hien thi hang chuc cua Giay
    SETB P2.4          ; Xoa dau cham o LED don vi cua Giay
    CALL DELAY_5MS     ; Delay 5ms
    CLR P3.1           ; Tat LED o pin P3.1

    SETB P3.2          ; Xu ly don vi Phut, tuong tu nhu xu ly don vi Giay phia tren
    MOV A, R1
    MOV B, #10
    DIV AB
    MOV P2, B
    CLR P2.4
    CALL DELAY_5MS
    CLR P3.2

    SETB P3.3          ; Xu ly hang chuc Phut
    MOV A, R1
    MOV B, #10
    DIV AB
    MOV P2, A
    SETB P2.4
    CALL DELAY_5MS
    CLR P3.3

    SETB P3.4          ; Xu ly don vi Gio, tuong tu nhu xu ly don vi Giay phia tren
    MOV A, R2
    MOV B, #10
    DIV AB
    MOV P2, B
    CLR P2.4
    CALL DELAY_5MS
    CLR P3.4

    SETB P3.5          ; Xu ly hang chuc Gio
    MOV A, R2
    MOV B, #10
    DIV AB
    MOV P2, A
    SETB P2.4
    CALL DELAY_5MS
    CLR P3.5
    RET
```

+ Với hàm hiển thị ‘DISPLAY’, ta thực hiện hiển thị Giây, Phút, Giờ ra 6 Led 7-Segments tương ứng. Sử dụng phương pháp quét led để hiển thị. Các chân Anot của Led được nối tương ứng với P3.0 -> P3.5. Tại một thời điểm chỉ có một trong các pin này của port P3 được bật, đồng nghĩa với chỉ có một Led sáng tại một thời điểm, sau đó delay trong khoản thời gian 5ms và tắt, Led tiếp theo sẽ được bật.

+ Đoạn code trên ta bật tắt lần lượt các pin P3.0 -> P3.5, và để hiển thị giá trị Giây, Phút, Giờ trên từng Led riêng lẻ, ta dùng lệnh ‘DIV AB’ để tách hàng đơn vị và hàng chục của các giá trị Giây, Phút, Giờ sau đó gán cho port P2 để hiển thị.

-Phần hàm Delay 5ms:

```

DELAY_5MS:                ; Ham delay 5ms
    MOV TL1, #78H
    MOV TH1, #0ECH
    SETB TR1
    JNB TF1, $             ; Không cần phải xóa cờ TF1 vì khi Timer1 tràn hàm xử lý ngắt sẽ tự xóa bằng phản cung
    RET

```

+ Theo như tính toán trên phần khởi tạo, ta có giá trị cần nạp cho Timer1 để delay 5ms là 0xEC78 được nạp lần lượt vào thanh ghi TH1 và TL1. Thực hiện nhảy tại chỗ khi cờ TF1 chưa được bật (tức Timer1 chưa tràn). Trong hàm ‘DELAY_5MS’ ta không cần xóa cờ TF1 vì khi Timer1 tràn, cờ TF1 sẽ tự động được xóa khi hàm xử lý Ngắt (Interrupt) được gọi.

-Phần hàm xử lý Ngắt (Interrupt):

```

ISR_TIMER1:                ; Ham xử lý khi Timer1 tràn, ; XU LY DIGITAL CLOCK
    DJNZ R3, NOT_EQUAL     ; Kiểm tra delay dư 200*5ms = 1s để tăng Giây, nếu không bằng thì thoát hàm xử lý ngắt
    MOV R3, #200           ; Nạp lại số lần lặp delay 5ms
    INC R0                 ; Tăng Giây
    MOV A, R0              ; Gán A = Giây
    MOV B, #60             ; Gán B = 60
    CJNE A, B, NOT_EQUAL   ; Kiểm tra Giây = 60?, nếu không bằng thì thoát hàm xử lý ngắt
    MOV R0, #0             ; Nếu Giây = 60 => reset Giây = 0 và tăng Phút
    INC R1                 ; Tăng Phút
    MOV A, R1              ; Gán A = Phút
    MOV B, #60             ; Gán B = 60
    CJNE A, B, NOT_EQUAL   ; Kiểm tra Phút = 60?, nếu không bằng thì thoát hàm xử lý ngắt
    MOV R1, #0             ; Nếu Phút = 60 => reset Phút = 0 và tăng Giờ
    INC R2                 ; Tăng Giờ

    MOV A, R2              ; Gán A = Giờ
    MOV B, #24             ; Gán B = 24
    CJNE A, B, NOT_EQUAL   ; Kiểm tra Giờ = 24?, nếu không bằng thì thoát hàm xử lý ngắt
    MOV R2, #0             ; Nếu Giờ = 24 => reset Giờ = 0

    NOT_EQUAL:            ; Thoát hàm xử lý ngắt

    RETI
END

```

+ Trong hàm xử lý Ngắt (Interrupt), ta thực hiện theo trình tự sau:

- Kiểm tra có Delay đủ 1s (thông qua 200 lần delay 5ms), nếu chưa đủ 1s thì thoát khỏi hàm Ngắt. Nếu đã delay đủ 1s, tăng giá trị R0, tức tăng Giây lên 1 đơn vị.
- Kiểm tra Giây = 60?, nếu không bằng thoát khỏi hàm Ngắt, nếu Giây = 60, thực hiện tăng giá trị Phút lên 1 đơn vị, tức tăng R1 và reset Giây về 0 (R0=0).
- Kiểm tra Phút = 60?, nếu không bằng thoát khỏi hàm Ngắt, nếu Phút = 60 thực hiện tăng giá trị Giờ lên 1 đơn vị, tức tăng R2 và reset Phút về 0 (R1 = 0).
- Kiểm tra Giờ = 24?, nếu không bằng thoát khỏi Ngắt, nếu Giờ = 24 thì reset Giờ về 0 (R2 = 0).
- Thoát khỏi Ngắt.

⇒ **Tổng hợp các đoạn code đã giải thích ở trên ta được một chương trình hoàn chỉnh có chức năng thực hiện đếm và hiển thị thời gian Giờ:Phút:Giây.**

3) Exercise report. (2 points)

*Giải thích code:

-Ý tưởng: Ta thực hiện tính thời gian delay trong lúc gọi hàm 'DISPLAY', trong 1 vòng lặp hàm 'DISPLAY' delay khoảng thời gian $6 \times 5\text{ms} = 30\text{ms}$. Vậy ta cần lặp 33 lần để xấp xỉ 1s. Khi đã tính đủ thời gian 1s, ta thực hiện xử lý và hiển thị Giây, Phút, Giờ tương tự như phần đã trình bày bên trên.

-Phần khai báo và khởi tạo:

```
$NOMOD51
$INCLUDE (8051.MCU)

; Reset Vector
ORG 0000H
JMP START

ORG 0100H
START:
    MOV TMOD, #10H      ; Timer1 che do 1
    MOV TL1, #78H       ; Nap gia tri delay 5ms cho Timer1
    MOV TH1, #0ECH
    SETB TR1            ; Bat Timer1
                        ; Khoi tao gia tri ban dau cua dong ho
    MOV R0, #55          ; R0 luu gia tri Giay
    MOV R1, #59          ; R1 luu gia tri Phut
    MOV R2, #23          ; R2 luu gia tri Gio
    MOV R3, #33          ; Xac dinh so lan lap de tao ra 1s de xac dinh thoi diem tang gia tri Giay
                        ; Mot vong lap DISPLAY delay 6*5ms => 33 * 6*5ms ~ 1s
```

-Phần vòng lặp:

```
LOOP:                                ; Thực hiện hiển thị Thời Gian
    DJNZ R3, NOT_INC_SECOND          ; Kiểm tra xem đã lặp đủ Delay khoảng 1s chưa, nếu chưa KHÔNG tăng Giay
    CALL CLOCK                       ; Nếu đã Delay đủ 1s, Gọi hàm CLOCK để tính toán Thời Gian
    NOT_INC_SECOND:
    CALL DISPLAY                     ; Hiển thị Thời Gian

    JMP LOOP
```

-Phần hiển thị (DISPLAY):

```
DISPLAY:
    SETB P3.0                        ; Bật LED hiển thị đơn vị Giay ở pin P3.0
    MOV A, R0
    MOV B, #10
    DIV AB                            ; Chia lấy dư để tách hàng đơn vị của Giay
    MOV P2, B                        ; Gán P2 = B, tức là hiển thị hàng đơn vị của Giay
    SETB P2.4                        ; Xóa dấu chấm ở LED đơn vị của Giay
    CALL DELAY_5MS                   ; Delay 5ms
    CLR P3.0                         ; Tắt LED ở pin P3.0

    SETB P3.1                        ; Bật LED hiển thị hàng chục của Giay ở pin P3.1
    MOV A, R0
    MOV B, #10
    DIV AB                            ; Chia lấy nguyên để tách hàng đơn vị của Giay
    MOV P2, A                        ; Gán P2 = A, tức là hiển thị hàng chục của Giay
    SETB P2.4                        ; Xóa dấu chấm ở LED đơn vị của Giay
    CALL DELAY_5MS                   ; Delay 5ms
    CLR P3.1                         ; Tắt LED ở pin P3.1

    SETB P3.2                        ; Xử lý đơn vị Phút, tương tự như xử lý đơn vị Giay phía trên
    MOV A, R1
    MOV B, #10
    DIV AB                            ; Xử lý hàng đơn vị Phút
    MOV P2, B
    CLR P2.4
    CALL DELAY_5MS
    CLR P3.2

    SETB P3.3                        ; Xử lý hàng chục Phút
    MOV A, R1
    MOV B, #10
    DIV AB
    MOV P2, A
    SETB P2.4
    CALL DELAY_5MS
    CLR P3.3

    SETB P3.4                        ; Xử lý đơn vị Gio, tương tự như xử lý đơn vị Giay phía trên
    MOV A, R2
    MOV B, #10
    DIV AB                            ; Xử lý hàng đơn vị Gio
    MOV P2, B
    CLR P2.4
    CALL DELAY_5MS
    CLR P3.4

    SETB P3.5                        ; Xử lý hàng chục Gio
    MOV A, R2
    MOV B, #10
    DIV AB
    MOV P2, A
    SETB P2.4
    CALL DELAY_5MS
    CLR P3.5

    RET
```

-Phần hàm Delay 5ms:

```
DELAY_5MS:                ; Ham delay 5ms xu dung vong lap
    MOV R4, #10            ; Thoi gian Delay = 10 * 250 * 2us = 5ms
    DELAY:
        MOV R5, #250
        DJNZ R5, $
        DJNZ R4, DELAY
    RET
```

+ Thời gian delay được tính bằng: $10 * 250 * 2\mu s = 5ms$

-Phần hàm xử lý tăng thời gian:

```
CLOCK:
    MOV R3, #33            ; Nap lai gia tri lap trong khoang 1s
    INC R0                 ; Tang Giay
    MOV A, R0              ; Gan A = Giay
    MOV B, #60             ; Gan B = 60
    CJNE A, B, NOT_EQUAL   ; Kiem tra Giay = 60?, neu khong bang thi thoat ham xu ly ngay
    MOV R0, #0             ; Neu Giay = 60 => reset Giay = 0 va tang Phut
    INC R1                 ; Tang Phut

    MOV A, R1              ; Gan A = Phut
    MOV B, #60             ; Gan B = 60
    CJNE A, B, NOT_EQUAL   ; Kiem tra Phut = 60?, neu khong bang thi thoat ham xu ly ngay
    MOV R1, #0             ; Neu Phut = 60 => reset Phut = 0 va tang Gio
    INC R2                 ; Tang Gio

    MOV A, R2              ; Gan A = Gio
    MOV B, #24             ; Gan B = 24
    CJNE A, B, NOT_EQUAL   ; Kiem tra Gio = 24?, neu khong bang thi thoat ham xu ly ngay
    MOV R2, #0             ; Neu Gio = 24 => reset Gio = 0

    NOT_EQUAL:            ; Thoat ham xu ly ngay
    RET
END
```

⇒ Tổng hợp các đoạn code trên ta được chương trình đếm và hiển thị
Giờ:Phút:Giây sử dụng Delay bằng vòng lặp.

**So sánh điểm mạnh điểm yếu của 2 phương pháp:

-Sử dụng Interrupt và Định thời của Timer

- +**Điểm mạnh:** Giảm tải công việc cho vi xử lý bằng việc sử dụng module riêng biệt để tạo Delay. Đạt độ chính xác cao về thời gian Delay và giảm tiêu tốn tài nguyên hệ thống.
- +**Điểm yếu:** Phức tạp hơn phương pháp dùng vòng lặp trong việc cài đặt và sử dụng.

-Sử dụng Delay bằng vòng lặp:

- +**Điểm mạnh:** Đơn giản, dễ hiểu, dễ thực hiện cài đặt.
- +**Điểm yếu:** Tốn nhiều tài nguyên hơn, độ chính xác thấp, không phù hợp dùng cho hệ thống thực hiện nhiều tác vụ trong một lúc.

V. References

Thuật toán thiết kế Digital Clock

Timer/Counter

Interrupt 8051