

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



BÁO CÁO THỰC HÀNH MÔN VI XỬ LÝ – VI ĐIỀU KHIỂN
BÀI THỰC HÀNH SỐ 5: SỬ DỤNG INTERRUPT

Sinh viên thực hiện:

Trần Ngọc Ánh

22520077

Giảng viên hướng dẫn: Phạm Minh Quân

Mã lớp: CE103.O22

TP. HỒ CHÍ MINH, 26 THÁNG 5 NĂM 2024

1. Trình bày giải thuật xử lý 2 button có chức năng sau:

- **Button A: Pause/Resume đồng hồ bấm giờ**
- **Button B: Reset đồng hồ**

Button A: Pause/Resume

a) Khởi tạo:

- Biến isRunning để lưu trạng thái đồng hồ (đang chạy hoặc tạm dừng)
- Biến timer để lưu thời gian bấm giờ

b) Xử lý sự kiện click Button A:

- Nếu isRunning là true: thay đổi trạng thái isRunning sang false, dừng bộ đếm thời gian.
- Nếu isRunning là false: thay đổi trạng thái isRunning sang true, bắt đầu bộ đếm thời gian.

c) Cập nhật thời gian:

Nếu isRunning là true: cập nhật giá trị timer theo thời gian trôi qua, hiển thị giá trị timer lên màn hình.

Button B: Reset

Xử lý sự kiện click Button B:

- Đặt giá trị timer về 0
- Cập nhật hiển thị thời gian trên màn hình
- Thay đổi trạng thái isRunning sang false (tùy chọn)

2. Sử dụng AT89C51/AT89C52 kết hợp với module 4 đèn 7-segment LED và 2 nút bấm trên, thiết kế 1 mạch đồng hồ bấm giờ Sport clock với khả năng đếm chính xác đến 1% giây, khoảng đếm từ 00.00 giây đến 99.99 giây và có 2 nút điều khiển Pause/Resume và Reset.

a) Source code:

```
org 00h
REST:
MOV R0, #00
MOV R2, #00
CALL HIEN THI
JB P1.0, $
JMP LAP
```

LAP:

CALL HIENITHI

CALL DELAY

INC R0

CJNE R0, #100, LAP

MOV R0, #00

JMP TANG

TANG:

INC R2

CJNE R2, #100, LAP

INC R0

JMP REST

STOP:

JB P1.0, \$

JMP LAP

HIENITHI:

mov DPTR, #MALED

mov A, R0

mov B, #10

div AB

movc A, @A+DPTR

mov P2, A

clr P3.0

setb P3.0

mov A, B

movc A, @A+DPTR

```
mov P2,A
clr P3.1
Setb P3.1

MOV A, R2
MOV B, #10
DIV AB
MOVC A, @A+DPTR
MOV P2, A
CLR P3.2
SETB P3.2
MOV A,B
MOVC A, @A+DPTR
MOV P2,A
CLR P3.3
SETB P3.3
RET
```

DELAY:

```
MOV TMOD, #01H
MOV TH0, #0E7H
MOV TL0, #00H
SETB TR0
```

KTN:

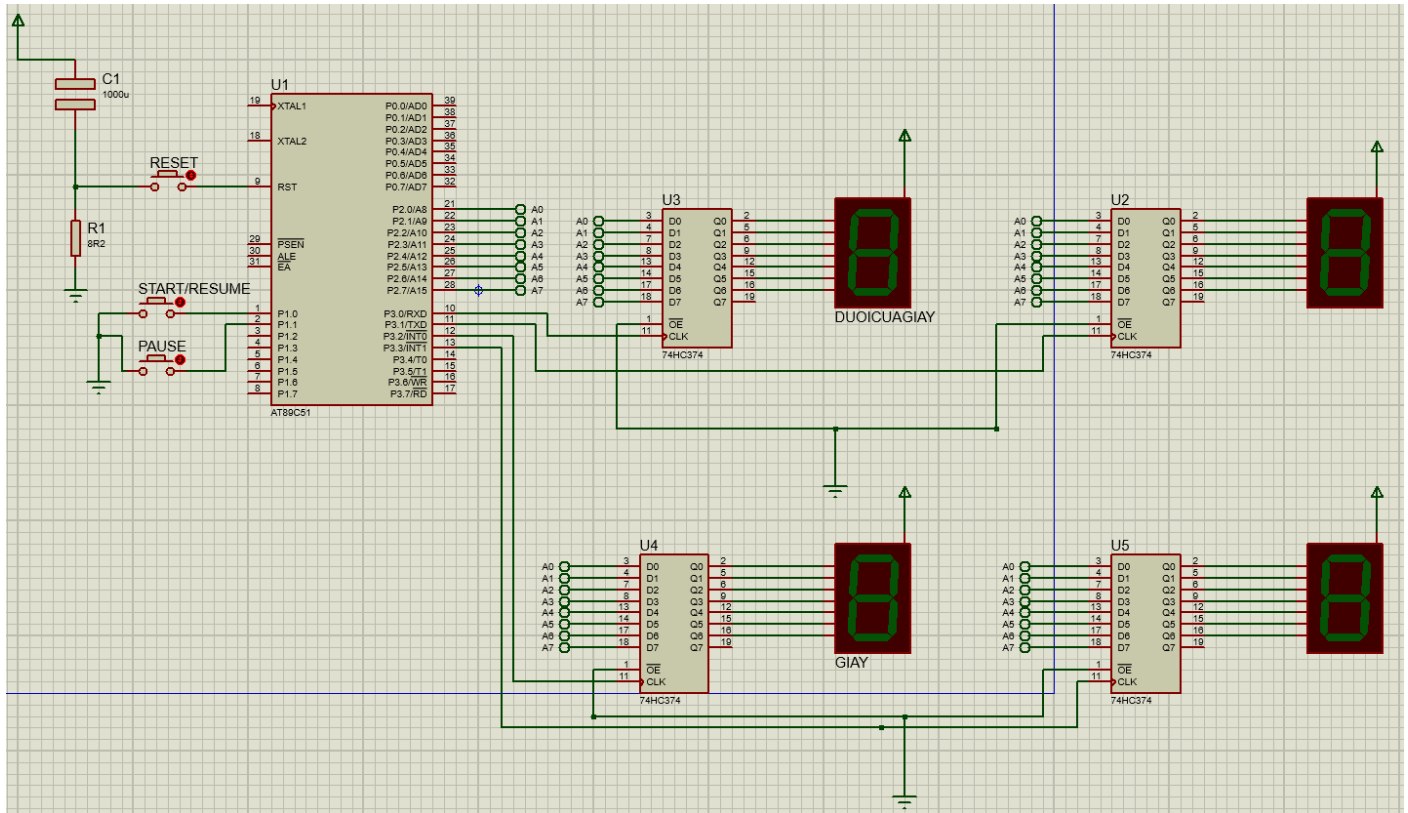
```
JNB P1.1, STOP
JNB TF0, $
CLR TR0
CLR TF0
RET
```

MALED:

DB 0C0H, 0F9H, 0A4H, 0B0H, 99H, 92H, 82H, 0F8H, 80H, 90H

END

b) Mạch mô phỏng:



c) Giải thích nguyên lý hoạt động:

Đoạn code Assembly này được viết cho vi điều khiển AT89C51/AT89C52, sử dụng module 4 đèn 7-segment LED và 2 nút bấm để thực hiện chức năng đồng hồ bấm giờ thể thao. Đồng hồ có khả năng đếm chính xác đến 1% giây, phạm vi đếm từ 00.00 giây đến 99.99 giây và có hai nút điều khiển:

- Nút Pause/Resume: Tạm dừng/Tiếp tục đếm thời gian.
- Nút Reset: Khởi động lại đồng hồ về 00.00 giây.

Phân tích code:

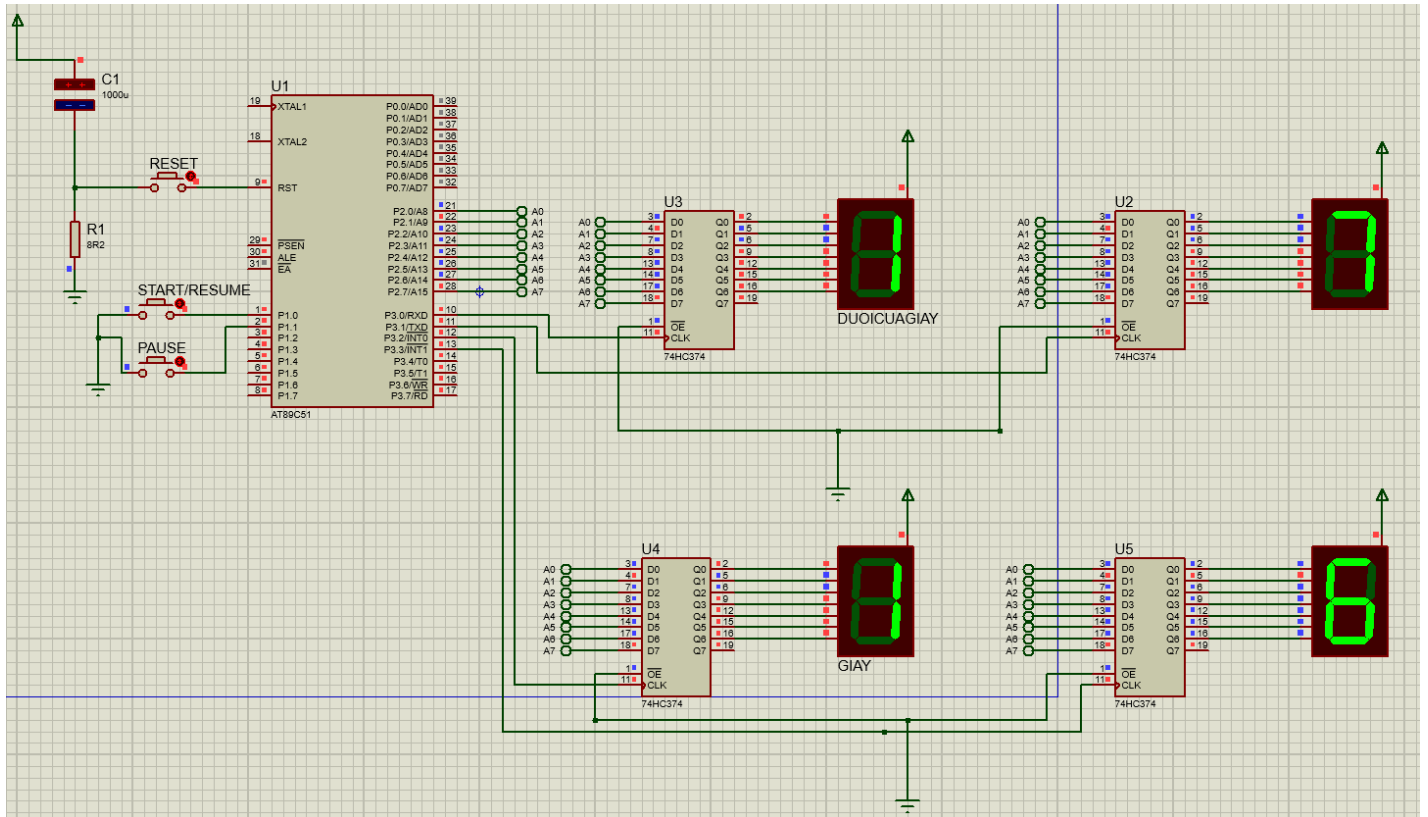
- Vùng nhớ:
 - **ORG 00h**: Vùng nhớ bắt đầu cho chương trình chính (REST).
 - **MALED**: Vùng nhớ chứa dữ liệu hiển thị ký tự lên LED 7-segment.
- Biến:

- **R0**: Lưu trữ giá trị giây.
- **R2**: Lưu trữ giá trị phút.
- **TMOD**: Chế độ hẹn giờ.
- **TH0**: Giá trị cao của thanh ghi hẹn giờ.
- **TL0**: Giá trị thấp của thanh ghi hẹn giờ.
- **TR0**: Kiểm soát hoạt động của hẹn giờ.
- **Hàm chính:**
 - **REST**: Hàm chính, khởi tạo biến, gọi hàm **HIENTHI** và thực hiện vòng lặp chính.
 - **LAP**: Hàm cập nhật và hiển thị thời gian lên LED, thực hiện vòng lặp 100 lần mỗi giây.
 - **TANG**: Hàm tăng giá trị giây.
 - **HIENTHI**: Hàm hiển thị giá trị giây và phút lên LED.
 - **DELAY**: Hàm tạo độ trễ 1ms.
 - **KTN**: Hàm kiểm tra cờ ngắt, dừng chương trình khi cờ ngắt được kích hoạt.
- **Hiển thị LED:**
 - Hàm **HIENTHI** sử dụng **DPTR** để truy cập dữ liệu trong mảng **MALED**.
 - Mảng **MALED** chứa dữ liệu hiển thị ký tự lên LED 7-segment tương ứng với giá trị của **R0** và **R2**.
 - Các hàm **MOVC** và **RET** giúp truy cập và hiển thị dữ liệu lên LED.
- **Hẹn giờ:**
 - Hàm **DELAY** sử dụng chế độ hẹn giờ T0 (TMOD) để tạo độ trễ 1ms.
 - Các thanh ghi **TH0** và **TL0** được thiết lập để tạo ra chu kỳ 1ms.
 - Cờ **TR0** được bật để kích hoạt hẹn giờ.
 - Hàm **KTN** kiểm tra cờ ngắt TF0, khi cờ ngắt được kích hoạt, hàm sẽ tắt cờ **TR0** và cờ **TF0**.

d) Kết quả:

Video mô phỏng lại quá trình mạch chạy trên phần mềm Proteus được đính kèm trong link này:

https://drive.google.com/file/d/1kHl0kKpcDdEHgznXgxmr5S_rDI7Snm2T/view?usp=sharing



3. Bổ sung thêm 2 nút bấm vào đồng hồ với chức năng sau:

- Nút bấm C: tăng số giây đếm lên 1 giây.
- Nút bấm D: giảm số giây xuống 1 giây.

a) Source code:

```
; Reset Vector
org 0000h
jmp Start
org 0003h
jmp buttonA
org 0013h
jmp buttonB

; CODE SEGMENT
org 0100h

Start:
mov IE, #10000101b
MOV R0, #00
MOV R1, #00
mov R4, #1

Main:
```

```
MOV R3, #10
MOV DPTR, #NUMBER
```

```
LOOP:
```

```
MOV A, R0
MOV B, #10
DIV AB
MOV P2, #00010000B
MOVC A, @A+DPTR
MOV P1, A
CALL DELAY
MOV A, B
MOV P2, #00100000B
MOVC A, @A+DPTR
MOV P1, A
CALL DELAY
MOV A, R1
MOV B, #10
DIV AB
MOV P2, #00000100B
MOVC A, @A+DPTR
MOV P1, A
CALL DELAY
MOV A, B
MOV P2, #00001000B
MOVC A, @A+DPTR
MOV P1, A
CALL DELAY

DJNZ R3, LOOP
    jb P3.0, p10
    jnb P3.0, $
    cjne R1, #99, increase
    jmp p10
```

```
increase:
```

```
    inc R1
p10:
    jb P3.1, continue
    jnb P3.1, $
```



```

        cjne R1, #0,decrease
        mov R1,#00
        jmp p10
decrease:
        dec R1

continue:
        cjne R4,#1,LOOP

        INC R0
        CJNE R0,#100,Main
        MOV R0,#00
        INC R1
        CJNE R1,#60,Main
        MOV R1,#00
JMP Main

NUMBER: DB 0C0H, 0F9H, 0A4H, 0B0H, 099H, 092H, 082H, 0F8H,
080H, 090H

DELAY:
        mov R6, #5
        Delay1:
        mov R7, #20
        djnz R7, $
        djnz R6, Delay1
ret

buttonA:
        jnb P3.2, $
        mov A,R4
        cpl A
        mov R4,A
reti

buttonB:
        mov R0, #0
        mov R1, #0
reti

```


- **Main**: Vòng lặp chính, thực hiện cập nhật và hiển thị thời gian.
- **LOOP**: Vòng lặp hiển thị từng chữ số lên LED.
- **DELAY**: Tạo độ trễ 1ms.
- **buttonA**: Xử lý ngắt nút Pause/Resume (chưa sửa).
- **buttonB**: Xử lý ngắt nút Reset (chưa sửa).
- **buttonC**: Xử lý ngắt nút tăng giây (đã thêm).
- **buttonD**: Xử lý ngắt nút giảm giây (đã thêm).
- **Hiển thị LED**: giống như phiên bản trước, code sử dụng **DPTR** và **NUMBER** để hiển thị giá trị giây và phút lên LED.
- **Xử lý nút bấm mới**:
 - **buttonC**: Kiểm tra trạng thái nút P3.2 (nút C). Nếu nút được nhấn, đổi giá trị **R4** để tạm dừng đếm (1) hoặc tiếp tục đếm (0).
 - **buttonD**: Kiểm tra trạng thái nút P3.3 (nút D). Nếu nút được nhấn, thiết lập lại **R0** và **R1** về 0 (khởi động lại đồng hồ).

d) Kết quả:

Video mô phỏng lại quá trình mạch chạy trên phần mềm Proteus được đính kèm trong link này:

https://drive.google.com/file/d/1pYqgeKB1IZecireXRST5FKGfgN__9Yfg/view?usp=sharing

