

TÀI LIỆU HƯỚNG DẪN LẬP TRÌNH CHO HỆ VI XỬ LÝ <phần I>

Khi một bài toán thực tế được đưa ra, ta cần phải tìm hiểu cặn kẽ các yêu cầu do bài toán đặt ra và thực hiện giải quyết theo các bước sau:

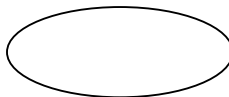
- *Bước 1: Định hướng giải quyết.*
- *Bước 2: Lựa chọn phương án thiết kế phần cứng.*
- *Bước 3: Lập lưu đồ thuật toán cho thiết kế phần mềm.*
- *Bước 4: Thể hiện lưu đồ thuật toán bằng các lệnh cụ thể.*

Các bước 1 và 2 đã được trình bày trong các tài liệu hướng dẫn thiết kế các mạch thí nghiệm, ở đây chỉ chú trọng vào giải quyết các bước 3 và 4. Trong giai đoạn thiết kế phần mềm, bước 3 không chỉ là bước tiên phong mà còn là bước có ý nghĩa quyết định đối với sự thành công và chất lượng của sản phẩm. Bước 4 chỉ là bước hoàn thiện cuối cùng của việc giải quyết một bài toán.

I. Cách xây dựng một lưu đồ thuật toán:

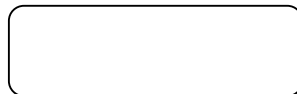
Trước hết nói về lưu đồ thuật toán, như đã đề cập đến trong phần các kiến thức căn bản trong Kỹ thuật Vi xử lý, lưu đồ thuật toán có vai trò đặc biệt quan trọng trong thiết kế phần mềm. Về cơ bản, một lưu đồ thuật toán có thể được xây dựng bằng các biểu tượng sau đây:

1. Biểu tượng 1:



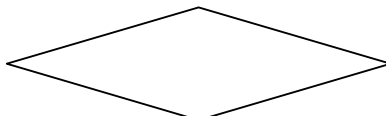
Đây là biểu tượng dùng để chỉ sự bắt đầu hay kết thúc một nhiệm vụ hay một chương trình. Biểu tượng này chỉ có một đầu vào hoặc một đầu ra.

2. Biểu tượng 2:



Đây là biểu tượng dùng để diễn đạt sự thực hiện một thao tác nào đó. Biểu tượng này có một đầu vào và một đầu ra.

3. Biểu tượng 3:



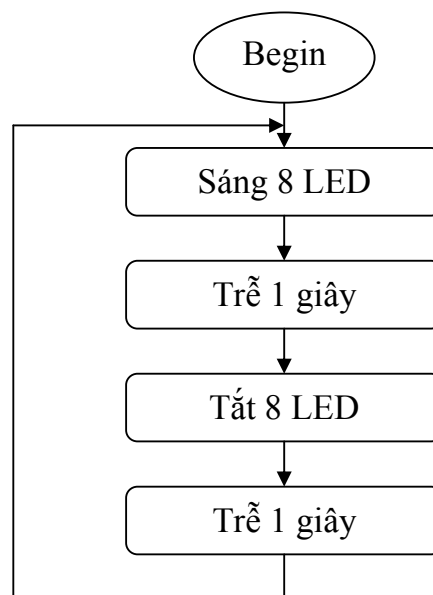
Đây là biểu tượng dùng để diễn đạt sự lựa chọn một trong hai phương án đúng hoặc sai đối với điều kiện hay tình huống được nêu ra trong đó. Biểu tượng này chỉ có một đầu vào nhưng lại có hai đầu ra.

Lưu ý là phải đặc biệt tuân thủ số lượng các đầu vào ra đối với mỗi loại biểu tượng.

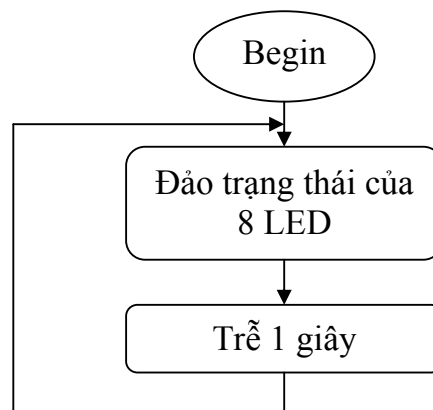
Để làm rõ hơn cách xây dựng một lưu đồ cụ thể, MicroStudy sẽ đưa ra một bài toán mẫu đơn giản trên cơ sở phần mạch phần cứng thí nghiệm và lưu đồ mẫu để giải quyết bài toán đó.

Bài toán đặt ra: nhấp nháy 8 LED đơn nối với cổng P1 theo chu kỳ 1 giây sáng và 1 giây tắt.

Như vậy sau khi phân tích yêu cầu do bài toán đặt ra, lưu đồ thuật toán phải được xây dựng như sau:



Hoặc nếu không quan tâm đến trạng thái bắt đầu của sự nhấp nháy (tức là không cần biết sáng trước hay tắt trước), ta có thể có được lưu đồ đơn giản hơn:



Với lưu đồ thuật toán thứ nhất ta có thể thực hiện bước 4 như sau:

```
#include    <sfr51.inc>
            org    00h                ;địa chỉ reset của vi điều khiển họ 8051
            ljmp   main

            org    40h                ;đặt chương trình chính bắt đầu từ địa chỉ này
            để tránh không đề lên vùng các vectơ ngắt
main:
            mov    p1,#0                ;cho sáng 8 đèn LED nối với P1
            lcall  delay1s                ;trễ 1 giây
            mov    p1,#0ffh            ;tắt 8 đèn LED nối với P1
            lcall  delay1s                ;trễ 1 giây
            sjmp   main                ;quay trở lại tiếp tục như thế

delay1s:                                ; đây là chương trình con tạo trễ 1 giây
            mov    r1,#10
loop1:
            mov    r2,#100
loop2:
            mov    r3,#100
loop3:
            nop
            nop
            nop
            nop
            nop
            nop
            nop
            nop
            nop
            djnz   r3,loop3
            djnz   r2,loop2
            djnz   r1,loop1
            ret

end                                                ;kết thúc chương trình
```

Với lưu đồ thuật toán thứ hai, chương trình có thể được viết là:

```
#include    <sfr51.inc>
            org    00h
            ljmp   main

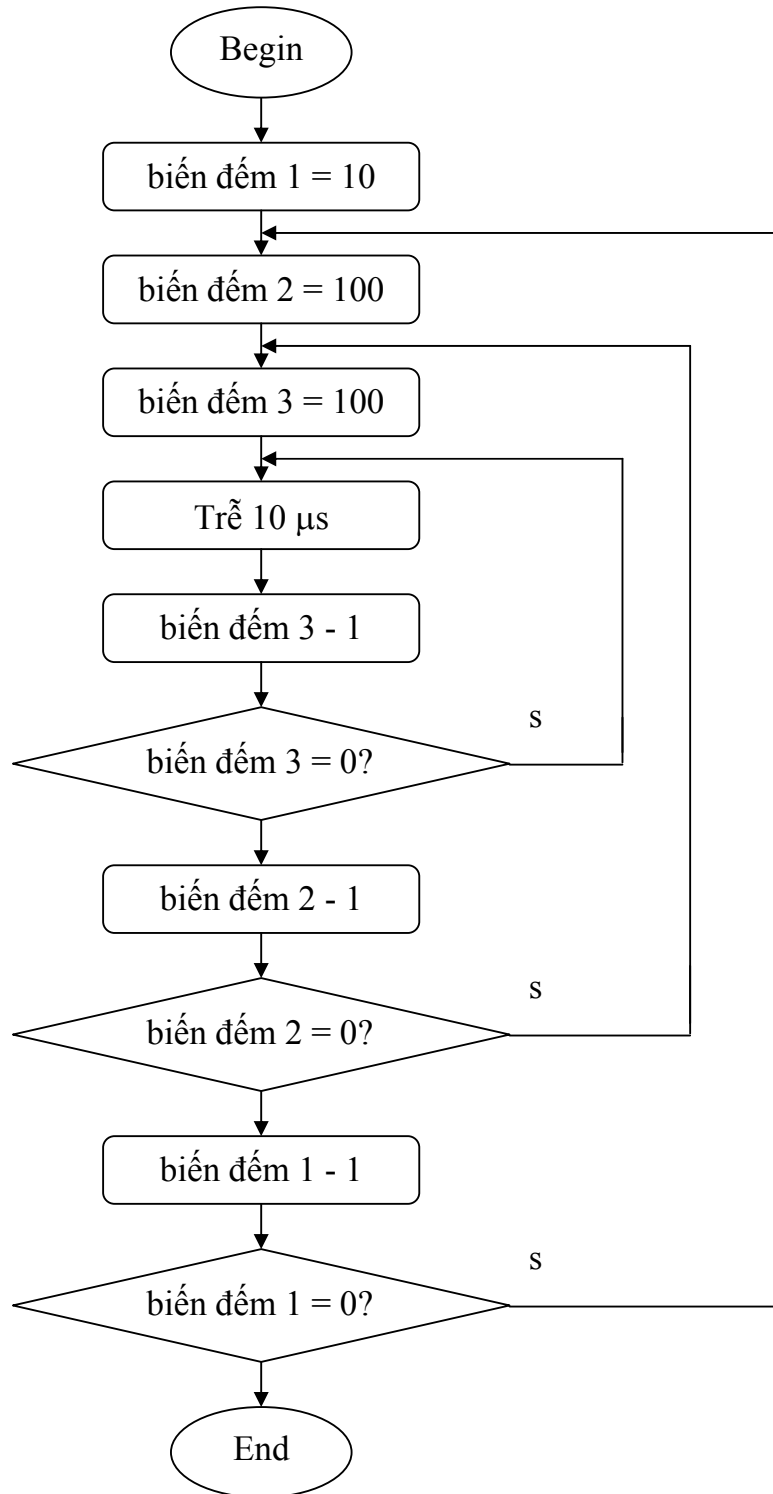
            org    40h
main:
```

```
        mov    a,p1        ;lấy giá trị hiện thời của cổng P1
        cpl    a           ;đảo nó đi (lật trạng thái, 0 thành 1, 1 thành 0)
        mov    p1,a        ;đưa ra trở lại P1 để thực hiện đảo trạng thái
các đèn LED
        lcall  delay1s
        sjmp   main
```

```
delay1s:
        mov    r1,#10
loop1:
        mov    r2,#100
loop2:
        mov    r3,#100
loop3:
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        djnz   r3,loop3
        djnz   r2,loop2
        djnz   r1,loop1
        ret
```

end

Trong hai chương trình trên có sử dụng chương trình con “delay1s”. Chương trình con này sử dụng các lệnh NOP để tiêu thụ thời gian chết khoảng $1\mu s$ với thạch anh 12MHz. Có 3 vòng lặp để tạo ra $10(\text{của vòng ngoài cùng}) * 100(\text{của vòng giữa}) * 100(\text{của vòng trong cùng}) * 10\mu s$ (do 8 lệnh NOP và một lệnh djnz gây ra) = $1,000,000\mu s = 1$ giây. Lưu đồ thuật toán cho chương trình con “delay1s” như sau:



II. Bài tập thực hành:

Viết lưu đồ thuật toán giải quyết các bài toán sau:

- *Sáng lần lượt các LED nối với cổng P1 (theo thứ tự P1.0 đầu tiên, P1.7 sau cùng rồi lại quay về P1.0). Tại một thời điểm chỉ có một LED sáng và thời gian sáng của các LED đều là 1 giây.*
- *Kiểm tra và đáp ứng lại sự kiện phím bấm ở cổng P3.2. Nếu phím bấm thì bật sáng cả 8 LED nối với cổng P1. Nhả phím ra thì tắt tất cả 8 LED.*
- *Hiển thị số 123 ở các LED 7 thanh.*
- *Hiển thị số đếm tăng dần bằng các LED 7 thanh (đếm từ 000 đến 999 rồi lại về 000, mỗi giây tăng số đếm một lần).*
- *Xác định phím bấm trong số 16 phím của ma trận phím và hiển thị số thứ tự của phím được bấm lên các LED 7 thanh (từ 1 đến 16, không có phím bấm thì hiện số 0).*