

CHƯƠNG 11

Lập trình các ngắt

Một ngắt là một sự kiện bên trong hoặc bên ngoài làm ngắt bộ vi điều khiển để báo cho nó biết rằng thiết bị cần dịch vụ của nó. Trong chương này ta tìm hiểu khái niệm ngắt và lập trình ngắt.

11.1 Các ngắt của 8051.

11.1.1 Các ngắt ngược với thăm dò.

Một bộ vi điều khiển có thể phục vụ một vài thiết bị, có hai cách để thực hiện điều này đó là sử dụng các ngắt và thăm dò (polling). Trong phương pháp sử dụng các ngắt thì mỗi khi có một thiết bị bất kỳ cần đến dịch vụ của nó thì nó báo cho bộ vi điều khiển bằng cách gửi một tín hiệu ngắt. Khi nhận được tín hiệu ngắt thì bộ vi điều khiển ngắt tất cả những gì nó đang thực hiện để chuyển sang phục vụ thiết bị. Chương trình đi cùng với ngắt được gọi là trình dịch vụ ngắt ISR (Interrupt Service Routine) hay còn gọi là trình quản lý ngắt (Interrupt handler). Còn trong phương pháp thăm dò thì bộ vi điều khiển hiển thị liên tục tình trạng của một thiết bị đã cho và điều kiện thỏa mãn thì nó phục vụ thiết bị. Sau đó nó chuyển sang hiển thị tình trạng của thiết bị kế tiếp cho đến khi tất cả đều được phục vụ. Mặc dù phương pháp thăm dò có thể hiển thị tình trạng của một vài thiết bị và phục vụ mỗi thiết bị khi các điều kiện nhất định được thỏa mãn nhưng nó không tận dụng hết công dụng của bộ vi điều khiển. Điểm mạnh của phương pháp ngắt là bộ vi điều khiển có thể phục vụ được rất nhiều thiết bị (tất nhiên là không tại cùng một thời điểm). Mỗi thiết bị có thể nhận được sự chú ý của bộ vi điều khiển dựa trên mức ưu tiên được gán cho nó. Đối với phương pháp thăm dò thì không thể gán mức ưu tiên cho các thiết bị vì nó kiểm tra tất cả mọi thiết bị theo kiểu hơi vòng. Quan trọng hơn là trong phương pháp ngắt thì bộ vi điều khiển cũng còn có thể che hoặc làm lơ một yêu cầu dịch vụ của thiết bị. Điều này lại một lần nữa không thể thực hiện được trong phương pháp thăm dò. Lý do quan trọng nhất là phương pháp ngắt được ưu chuộng nhất là vì phương pháp thăm dò làm lãng phí thời gian của bộ vi điều khiển bằng cách hỏi dò từng thiết bị kể cả khi chúng không cần đến dịch vụ. Nhằm để tránh thì người ta sử dụng phương pháp ngắt. Ví dụ trong các bộ định thời được bàn đến ở chương 9 ta đã dùng lệnh “JNB TF, đích” và đợi cho đến khi bộ định thời quay trở về 0. Trong ví dụ đó, trong khi chờ đợi thì ta có thể làm việc được gì khác có ích hơn, chẳng hạn như khi sử dụng phương pháp ngắt thì bộ vi điều khiển có thể đi làm các việc khác và khi cờ TF bật lên nó sẽ ngắt bộ vi điều khiển cho dù nó đang làm bất kỳ điều gì.

11.1.2 Trình phục vụ ngắt.

Đối với mỗi ngắt thì phải có một trình phục vụ ngắt ISR hay trình quản lý ngắt. Khi một ngắt được gọi thì bộ vi điều khiển phục vụ ngắt. Khi một ngắt được gọi thì bộ vi điều khiển chạy trình phục vụ ngắt. Đối với mỗi ngắt thì có một vị trí cố định trong bộ nhớ để giữ địa chỉ ISR của nó. Nhóm các vị trí nhớ được dành riêng để gửi các địa chỉ của các ISR được gọi là bảng véc tơ ngắt (xem hình 11.1).

11.1.3 Các bước khi thực hiện một ngắt.

Khi kích hoạt một ngắt bộ vi điều khiển đi qua các bước sau:

1. Nó kết thúc lệnh đang thực hiện và lưu địa chỉ của lệnh kế tiếp (PC) vào ngăn xếp.

2. Nó cũng lưu tình trạng hiện tại của tất cả các ngắt vào bên trong (nghĩa là không lưu vào ngăn xếp).
3. Nó nhảy đến một vị trí cố định trong bộ nhớ được gọi là bảng véc tơ ngắt nơi lưu giữ địa chỉ của một trình phục vụ ngắt.
4. Bộ vi điều khiển nhận địa chỉ ISR từ bảng véc tơ ngắt và nhảy tới đó. Nó bắt đầu thực hiện trình phục vụ ngắt cho đến lệnh cuối cùng của ISR là RETI (trở về từ ngắt).
5. Khi thực hiện lệnh RETI bộ vi điều khiển quay trở về nơi nó đã bị ngắt. Trước hết nó nhận địa chỉ của bộ đếm chương trình PC từ ngăn xếp bằng cách kéo hai byte trên đỉnh của ngăn xếp vào PC. Sau đó bắt đầu thực hiện các lệnh từ địa chỉ đó.

Lưu ý ở bước 5 đến vai trò nhạy cảm của ngăn xếp, vì lý do này mà chúng ta phải cẩn thận khi thao tác các nội dung của ngăn xếp trong ISR. Đặc biệt trong ISR cũng như bất kỳ chương trình con CALL nào số lần đẩy vào ngăn xếp (Push) và số lần lấy ra từ nó (Pop) phải bằng nhau.

11.1.4 Sáu ngắt trong 8051.

Thực tế chỉ có 5 ngắt dành cho người dùng trong 8051 nhưng nhiều nhà sản xuất đưa ra các bảng dữ liệu nói rằng có sáu ngắt vì họ tính cả lệnh tái thiết lập lại RESET. Sáu ngắt của 8051 được phân bố như sau:

1. RESET: Khi chân RESET được kích hoạt từ 8051 nhảy về địa chỉ 0000. Đây là địa chỉ bật lại nguồn được bàn ở chương 4.
2. Gồm hai ngắt dành cho các bộ định thời: 1 cho Timer0 và 1 cho Timer1. Địa chỉ của các ngắt này là 000B4 và 001B4 trong bảng véc tơ ngắt dành cho Timer0 và Timer1 tương ứng.
3. Hai ngắt dành cho các ngắt phần cứng bên ngoài chân 12 (P3.2) và 13 (P3.3) của cổng P3 là các ngắt phần cứng bên ngoài INT0 và INT1 tương ứng. Các ngắt ngoài cũng còn được coi như EX1 và EX2 vị trí nhớ trong bảng véc tơ ngắt của các ngắt ngoài này là 0003H và 0013H gán cho INT0 và INT1 tương ứng.
4. Truyền thông nối tiếp có một ngắt thuộc về cả thu và phát. Địa chỉ của ngắt này trong bảng véc tơ ngắt là 0023H.

Chú ý rằng trong bảng 11.1 có một số giới hạn các byte dành riêng cho mỗi ngắt. Ví dụ, đối với ngắt INT0 ngắt phần cứng bên ngoài 0 thì có tổng cộng là 8 byte từ địa chỉ 0003H đến 000AH dành cho nó. Tương tự như vậy, 8 byte từ địa chỉ 000BH đến 0012H là dành cho ngắt bộ định thời 0 là TI0. Nếu trình phục vụ ngắt đối mặt với một ngắt đã cho mà ngắn đủ đặt vừa không gian nhớ được. Nếu không vừa thì một lệnh LJMPP được đặt vào trong bảng véc tơ ngắt để chỉ đến địa chỉ của ISR, ở trường hợp này thì các byte còn lại được cấp cho ngắt này không dùng đến. Dưới đây là các ví dụ về lập trình ngắt minh họa cho các điều trình bày trên đây.

Từ bảng 11.1 cùng để ý thấy một thực tế rằng chỉ có 3 byte của không gian bộ nhớ ROM được gán cho chân RESET. Đó là những vị trí địa chỉ 0, 1 và 2 của ROM. Vị trí địa chỉ 3 thuộc về ngắt phần cứng bên ngoài 0 với lý do này trong chương trình chúng ta phải đặt lệnh LJMPP như là lệnh đầu tiên và hướng bộ xử lý lệnh khỏi bảng véc tơ ngắt như chỉ ra trên hình 11.1.

Bảng 11.1: Bảng véc tơ ngắt của 8051.

Ngắt	Địa chỉ ROM	Chân
Bật lại nguồn (RESET)	0000	9
Ngắt phân cứng ngoài (INT0)	0003	12 (P3.2)
Ngắt bộ Timer0 (TF0)	000B	
Ngắt phân cứng ngoài 1 (INT1)	0013	13 (P3.3)
Ngắt bộ Timer1 (TF1)	001B	
Ngắt COM nối tiếp (RI và TI)	0023	

11.1.5 Cho phép và cấm ngắt.

Khi bật lại nguồn thì tất cả mọi ngắt đều bị cấm (bị che) có nghĩa là không có ngắt nào sẽ được bộ vi điều khiển đáp ứng nếu chúng được kích hoạt. Các ngắt phải được kích hoạt bằng phần mềm để bộ vi điều khiển đáp ứng chúng. Có một thanh ghi được gọi là cho phép ngắt IE (Interrupt Enable) chịu trách nhiệm về việc cho phép (không che) và cấm (che) các ngắt. Hình 11.2 trình bày thanh ghi IE, lưu ý rằng IE là thanh ghi có thể đánh địa chỉ theo bit.

Từ hình 11.2 ta thấy rằng D7 của thanh ghi IE được gọi là bit cho phép tất cả các ngắt EA (Euable All). Bit này phải được thiết lập lên 1 để phần còn lại của thanh ghi hoạt động được. Bit D6 chưa được sử dụng. Bit D54 được dành cho 8051, còn bit D4 dùng cho ngắt nối tiếp v.v...

11.1.6 Các bước khi cho phép ngắt.

Để cho phép một ngắt ta phải thực hiện các bước sau:

1. Bit D7 của thanh ghi IE là EA phải được bật lên cao để cho phép các bit còn lại của thanh ghi nhận được hiệu ứng.
2. Nếu EA = 1 thì tất cả mọi ngắt đều được phép và sẽ được đáp ứng nếu các bit tương ứng của chúng trong IE có mức cao. Nếu EA = 0 thì không có ngắt nào sẽ được đáp ứng cho dù bit tương ứng của nó trong IE có giá trị cao.

Để hiểu điểm quan trọng này hãy xét ví dụ 11.1.

Hình 11.2: Thanh ghi cho phép ngắt IE.

D7							D0
EA	--	ET2	ES	ET1	EX1	ET0	EX0

EA IE.7 Nếu EA = 0 thì mọi ngắt bị cấm
Nếu EA = 1 thì mỗi nguồn ngắt được cho phép hoặc bị cấm bằng các bật hoặc xóa bit cho phép của nó.

-- IE.6 Dự phòng cho tương lai

ET2 IE.5 Cho phép hoặc cấm ngắt tràn hoặc thu của Timer2 (8051)

ES IE.4 Cho phép hoặc cấm ngắt cổng nối tiếp

ET1 IE.3 Cho phép hoặc cấm ngắt tràn của Timer1

EX1 IE.2 Cho phép hoặc cấm ngắt ngoài 1

ET0 IE.1 Cho phép hoặc cấm ngắt tràn của Timer0

EX0 IE.0 Cho phép hoặc cấm ngắt ngoài 0

* Người dùng không phải ghi 1 vào bit dự phòng này. Bit này có thể dùng cho các bộ vi điều khiển nhanh với đặc tính mới

Ví dụ 11.1:

Hãy chỉ ra những lệnh để a) cho phép ngắt nối tiếp ngắt Timer0 và ngắt phần cứng ngoài 1 (EX1) và b) cấm (che) ngắt Timer0 sau đó c) trình bày cách cấm tất cả mọi ngắt chỉ bằng một lệnh duy nhất.

Lời giải:

a) MOV IE, #10010110B ; Cho phép ngắt nối tiếp, cho phép ngắt Timer0 và cho phép ngắt phần cứng ngoài.

Vì IE là thanh ghi có thể đánh địa chỉ theo bit nên ta có thể sử dụng các lệnh sau đây để truy cập đến các bit riêng rẽ của thanh ghi:

```
SETB IE.7      ; EA = 1, Cho phép tất cả mọi ngắt
SETB IE.4      ; Cho phép ngắt nối tiếp
SETB IE.1      ; Cho phép ngắt Timer1
SETB IE.2      ; Cho phép ngắt phần cứng ngoài 1
```

(tất cả những lệnh này tương đương với lệnh “MOV IE, #10010110B” trên đây).

b) CLR IE.1 ; Xoá (che) ngắt Timer0

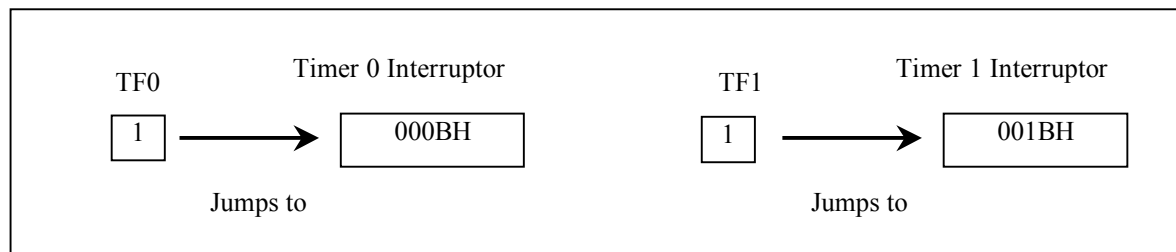
c) CLR IE.7 ; Cấm tất cả mọi ngắt.

11.2 Lập trình các ngắt bộ định thời.

Trong chương 9 ta đã nói cách sử dụng các bộ định thời Timer0 và Timer1 bằng phương pháp thăm dò. Trong phần này ta sẽ sử dụng các ngắt để lập trình cho các bộ định thời của 8051.

11.2.1 Cờ quay về 0 của bộ định thời và ngắt.

Trong chương 9 chúng ta đã nói rằng cờ bộ định thời TF được đặt lên cao khi bộ định thời đạt giá trị cực đại và quay về 0 (Roll - over). Trong chương trình này chúng ta cũng chỉ ra cách hiển thị cờ TF bằng lệnh “JNB TF, đích”. Khi thăm dò cờ TF thì ta phải đợi cho đến khi cờ TF được bật lên. Vấn đề với phương pháp này là bộ vi điều khiển bị trói buộc khi cờ TF được bật lên và không thể làm được bất kỳ việc gì khác. Sử dụng các ngắt giải quyết được vấn đề này và tránh được sự trói buộc của bộ vi điều khiển. Nếu bộ ngắt định thời trong thanh ghi IE được phép thì mỗi khi nó quay trở về 0 cờ TF được bật lên và bộ vi điều khiển bị ngắt tại bất kỳ việc gì nó đang thực hiện và nhảy tới bảng véc tơ ngắt để phục vụ ISR. Bằng cách này thì bộ vi điều khiển có thể làm những công việc khác cho đến khi nào nó được thông báo rằng bộ định thời đã quay về 0. Xem hình 11.3 và ví dụ 11.2.



Hình 11.3: Ngắt bộ định thời TF0 và TF1.

Hãy để những điểm chương trình dưới đây của chương trình trong ví dụ 11.2.

1. Chúng ta phải tránh sử dụng không gian bộ nhớ dành cho bảng véc tơ ngắt. Do vậy, ta đặt tất cả mã khởi tạo tại địa chỉ 30H của bộ nhớ. Lệnh LJMP là lệnh đầu

tiên mà 8051 thực hiện khi nó được cấp nguồn. Lệnh LJMP lái bộ điều khiển tránh khỏi bảng véc tơ ngắt.

2. Trình phục vụ ISR của bộ Timer0 được đặt ở trong bộ nhớ bắt đầu từ địa chỉ 000BH và vì nó quá nhỏ đủ cho vào không gian nhớ dành cho ngắt này.
3. Chúng ta cho phép ngắt bộ Timer0 với lệnh “MOV IE, #1000 010H” trong chương trình chính MAIN.
4. Trong khi dữ liệu ở cổng P0 được nhận vào và chuyển liên tục sang công việc P1 thì mỗi khi bộ Timer0 trở về 0, cờ TF0 được bật lên và bộ vi điều khiển thoát ra khỏi vòng lặp BACK và đi đến địa chỉ 000BH để thực hiện ISR gắn liền với bộ Timer0.
5. Trong trình phục vụ ngắt ISR của Timer0 ta thấy rằng không cần đến lệnh “CLR TF0” trước khi lệnh RETI. Lý do này là vì 8051 xóa cờ TF bên trong khi nhảy đến bảng véc tơ ngắt.

Ví dụ 11.2:

Hãy viết chương trình nhân liên tục dữ liệu 8 bit ở cổng P0 và gửi nó đến cổng P1 trong khi nó cùng lúc tạo ra một sóng vuông chu kỳ 200 μ s trên chân P2.1. Hãy sử dụng bộ Timer0 để tạo ra sóng vuông, tần số của 8051 là XTAL = 11.0592MHz.

Lời giải:

Ta sử dụng bộ Timer0 ở chế độ 2 (tự động nạp lại) giá trị nạp cho TH0 là $100/1.085\mu s = 92$.

; - - Khi khởi tạo vào chương trình main tránh dùng không gian.

; Địa chỉ dành cho bảng véc tơ ngắt.

ORG 0000H

CPL P2.1

; Nhảy đến bảng véc tơ ngắt.

;

; - - Trình ISR dành cho Timer0 để tạo ra sóng vuông.

ORG 0030H

; Ngay sau địa chỉ bảng véc-tơ ngắt

MAIN: TMOD, #02H

; Chọn bộ Timer0, chế độ 2 tự nạp lại

MOV P0, #0FFH

; Lấy P0 làm cổng vào nhận dữ liệu

MOV TH0, # - 92

; Đặt TH0 = A4H cho - 92

MOV IE, #82H

; IE = 1000 0010 cho phép Timer0

SETB TR0

; Khởi động bộ Timer0

BACK: MOV A, P0

; Nhận dữ liệu vào từ cổng P0

MOV P1, A

; Chuyển dữ liệu đến cổng P1

SJMP BACK

; Tiếp tục nhận và chuyển dữ liệu

; Chứng nào bị ngắt bởi TF0

END

Trong ví dụ 11.2 trình phục vụ ngắt ISR ngắn nên nó có thể đặt vừa vào không gian địa chỉ dành cho ngắt Timer0 trong bảng véc tơ ngắt. Tất nhiên không phải lúc nào cũng làm được như vậy. Xét ví dụ 11.3 dưới đây.

Ví dụ 11.3:

Hãy viết lại chương trình ở ví dụ 11.2 để tạo sóng vuông với mức cao kéo dài 1085 μ s và mức thấp dài 15 μ s với giả thiết tần số XTAL = 11.0592MHz. Hãy sử dụng bộ định thời Timer1.

Lời giải:

Vì $1085\mu s$ là $1000 \times 1085\mu s$ nên ta cần sử dụng chế độ 1 của bộ định thời Timer1.

```
; - - Khi khởi tạo tránh sử dụng không gian dành cho bảng véc tơ ngắt.
      ORG    0000H
      LJMP   MAIN                ; Chuyển đến bảng véc tơ ngắt.
;
; - - Trình ISR đối với Timer1 để tạo ra xung vuông
      ORG    001BH                ; Địa chỉ ngắt của Timer1 trong bảng véc tơ ngắt
      LJMP   ISR-T1              ; Nhảy đến ISR
;
; - - Bắt đầu các chương trình chính MAIN.
MAIN:  ORG    0030H                ; Sau bảng véc tơ ngắt
      MOV    TMOD, #10H           ; Chọn Timer1 chế độ 1
      MOV    P0, #0FFH           ; Chọn cổng P0 làm đầu vào nhận dữ liệu
      MOV    TL1, #018H          ; Đặt TL1 = 18 byte thấp của - 1000
      MOV    TH1, #0FCH          ; Đặt TH1 = FC byte cao của - 1000
      MOV    IE, #88H            ; IE = 10001000 cho phép ngắt Timer1
      SETB   TR1                 ; Khởi động bộ Timer1
BACK:  MOV    A, P0               ; Nhận dữ liệu đầu vào ở cổng P0
      MOV    P1, A               ; Chuyển dữ liệu đến P1
      SJMP   BACK                ; Tiếp tục nhận và chuyển dữ liệu
;
; - - Trình ISR của Timer1 phải được nạp lại vì ở chế độ 1
ISR-T1: CLR    TR1                ; Dừng bộ Timer1
      CLR    P2.1                ; P2.1 = 0 bắt đầu xung mức thấp
      MOV    R2, #4               ; 2 chu kỳ máy MC (Machine Cycle)
HERE:  DJNZ   R2, HERE            ;  $4 \times 2 \text{ MC} = 8 \text{ MC}$ 
      MOV    TL1, #18H           ; Nạp lại byte thấp giá trị 2 MC
      MOV    TH1, #0FCH          ; Nạp lại byte cao giá trị 2 MC
      SETB   TR1                 ; Khởi động Timer1 1 MC
      SETB   P2.1                ; P2.1 = 1 bật P2.1 trở lại cao
      RETI                       ; Trở về chương trình chính
      END
```

Lưu ý rằng phân xung mức thấp được tạo ra bởi 14 chu kỳ mức MC và mỗi $\text{MC} = 1.085\mu s$ và $14 \times 1.085\mu s = 15.19\mu s$.

Ví dụ 11.4:

Viết một chương trình để tạo ra một sóng vuông tần số 50Hz trên chân P1.2.

Ví dụ này tương tự ví dụ 9.12 ngoại trừ ngắt Timer0, giả sử $\text{XTAL} = 11.0592\text{MHz}$.

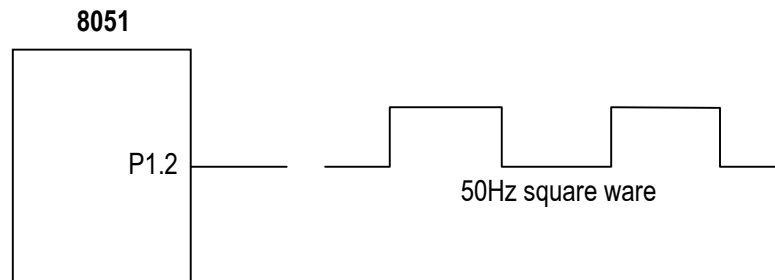
Lời giải:

```
ORG    0
LJMP   MAIN
ORG    000BH                ; Chương trình con phục vụ ngắt cho Timer0
CPL    P1.2
MOV    TL0, #00
MOV    TH0, #0DCH
RETI
ORG    30H
; ----- main program for initialization
MAIN:  MOV    TMOD, #0000001B    ; Chọn Timer0 chế độ 1
```

```

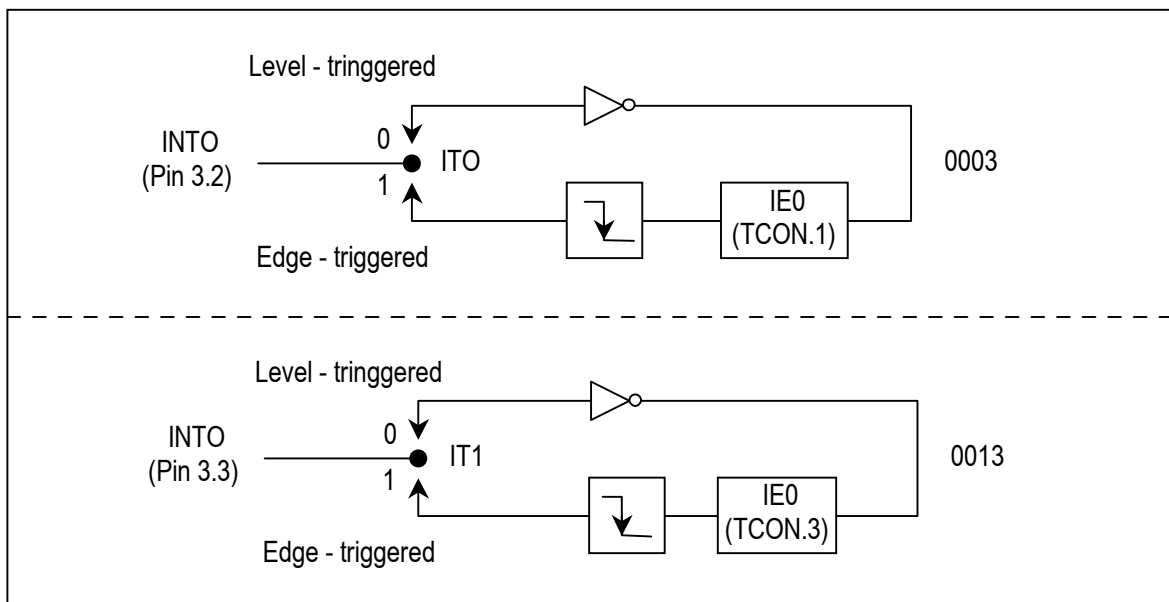
MOV    TL0, # 0DCH
MOV    IE, # 82H           ; Cho phép ngắt Timer0
SETB   TR0
HERE:  SJMP HERE
END

```



11.3 Lập trình các ngắt phần cứng bên ngoài.

Bộ vi điều khiển 8051 có hai ngắt phần cứng bên ngoài là chân 12 (P3.2) và chân 13 (P3.3) dùng cho ngắt INT0 và INT1. Khi kích hoạt những chân này thì 8051 bị ngắt tại bất kỳ công việc nào mà nó đang thực hiện và nó nhảy đến bảng véc tơ ngắt để thực hiện trình phục vụ ngắt.



11.3.1 Các ngắt ngoài INT0 và INT1.

Chỉ có hai ngắt phần cứng ngoài trong 8051 là INT0 và INT1. Chúng được bố trí trên chân P3.2 và P3.3 và địa chỉ của chúng trong bảng véc tơ ngắt là 0003H và 0013H. Như đã nói ở mục 11.1 thì chúng được ghép và bị cấm bằng việc sử dụng thanh ghi IE. Vậy chúng được kích hoạt như thế nào? Có hai mức kích hoạt cho các ngắt phần cứng ngoài: Ngắt theo mức và ngắt theo sườn. Dưới đây là mô tả hoạt động của mỗi loại.

11.3.2 Ngắt theo mức.

Ở chế độ ngắt theo mức thì các chân INT0 và INT1 bình thường ở mức cao (giống như tất cả các chân của cổng I/O) và nếu một tín hiệu ở mức thấp được cấp tới chúng thì nó ghi nhận ngắt. Sau đó bộ vi điều khiển dừng tất cả mọi công việc nó

đang thực hiện và nhảy đến bảng véc tơ ngắt để phục vụ ngắt. Điều này được gọi là ngắt được kích hoạt theo mức hay ngắt theo mức và là chế độ ngắt mặc định khi cấp nguồn lại cho 8051. Tín hiệu mức thấp tại chân INT phải được lấy đi trước khi thực hiện lệnh cuối cùng của trình phục vụ ngắt RETI, nếu không một ngắt khác sẽ lại được tạo ra. Hay nói cách khác, nếu tín hiệu ngắt mức thấp không được lấy đi khi ISR kết thúc thì nó không thể hiện như một ngắt khác và 8051 nhảy đến bảng véc tơ ngắt để thực hiện ISR. Xem ví dụ 11.5.

Ví dụ 11.5.

Giả sử chân INT1 được nối đến công tắc bình thường ở mức cao. Mỗi khi nó xuống thấp phải bật một đèn LED. Đèn LED được nối đến chân P1.3 và bình thường ở chế độ tắt. Khi nó được bật lên nó phải sáng vài phần trăm giây. Chừng nào công tắc được ấn xuống thấp đèn LED phải sáng liên tục.

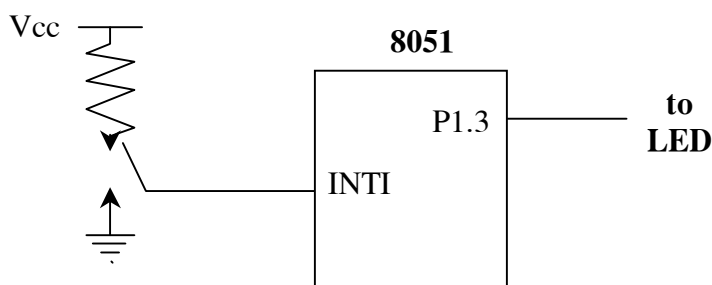
Lời giải:

```

                ORG    0000H
                LJMP    MAIN                ; Nhảy đến bảng véc tơ ngắt
; - - Chương trình con ISR cho ngắt cứng INT1 để bật đèn LED.
                ORG    0013H                ; Trình phục vụ ngắt ISR cho INT1
                SETB    P1.3                ; Bật đèn LED
                MOV     R3, # 255           ;
BACK:           DJNZ    R3, BACK            ; Giữ đèn LED sáng một lúc
                CLR     P1.3                ; Tắt đèn LED
                RETI                        ; Trở về từ ISR
; - - Bắt đầu chương trình chính Main.
                ORG     30H
MAIN:           MOV     IE, #10000100B     ; Cho phép ngắt dài
                SJMP    HERE                ; Chờ ở đây cho đến khi được ngắt
END

```

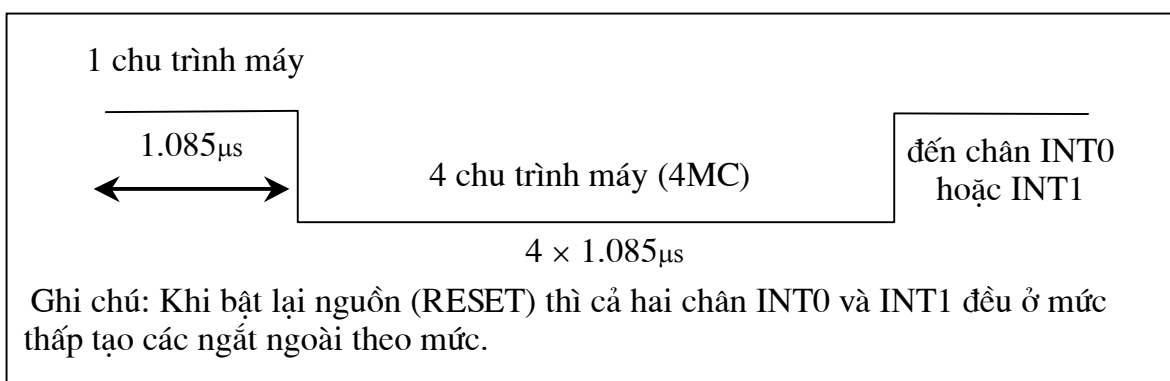
Ấn công tắc xuống sẽ làm cho đèn LED sáng. Nếu nó được giữ ở trạng thái được kích hoạt thì đèn LED sáng liên tục.



Trong chương trình này bộ vi điều khiển quay vòng liên tục trong vòng lặp HERE. Mỗi khi công tắc trên chân P3.3 (INT1) được kích hoạt thì bộ vi điều khiển thoát khỏi vòng lặp và nhảy đến bảng véc tơ ngắt tại địa chỉ 0013H. Trình ISR cho INT1 bật đèn LED lên giữ nó một lúc và tắt nó trước khi trở về. Nếu trong lúc nó thực hiện lệnh quay trở về RETI mà chân INT1 vẫn còn ở mức thấp thì bộ vi điều khiển khởi tạo lại ngắt. Do vậy, để giải quyết vấn đề này thì chân INT1 phải được đưa lên cao tại thời điểm lệnh RETI được thực hiện.

11.3.3 Trích mẫu ngắt theo mức.

Các chân P3.2 và P3.3 bình thường được dùng cho vào - ra nếu các bit INT0 và INT1 trong thanh ghi IE không được kích hoạt. Sau khi các ngắt phần cứng trong thanh ghi IE được kích hoạt thì bộ vi điều khiển duy trì trích mẫu trên chân INTn đối với tín hiệu mức thấp một lần trong một chu trình máy. Theo bảng dữ liệu của nhà sản xuất của bộ vi điều khiển thì “chân ngắt phải được giữ ở mức thấp cho đến khi bắt đầu thực hiện trình phục vụ ngắt ISR. Nếu chân INTn được đưa trở lại mức cao trước khi bắt đầu thực hiện ISR thì sẽ chẳng có ngắt nào xảy ra”. Tuy nhiên trong quá trình kích hoạt ngắt theo mức thấp nên nó lại phải đưa lên mức cao trước khi thực hiện lệnh RETI và lại theo bảng dữ liệu của nhà sản xuất thì “nếu chân INTn vẫn ở mức thấp sau lệnh RETI của trình phục vụ ngắt thì một ngắt khác lại sẽ được kích hoạt sau khi lệnh RETI được thực hiện”. Do vậy, để bảo đảm việc kích hoạt ngắt phần cứng tại các chân INTn phải khẳng định rằng thời gian tồn tại tín hiệu mức thấp là khoảng 4 chu trình máy và không được hơn. Điều này là do một thực tế là ngắt theo mức không được chốt. Do vậy chân ngắt phải được giữ ở mức thấp cho đến khi bắt đầu thực hiện ISR.



Hình 11.5: Thời gian tối thiểu của ngắt theo mức thấp (XTAL = 11.0592MHz)

11.3.4 Các ngắt theo sườn.

Như đã nói ở trước đây trong quá trình bật lại nguồn thì 8051 làm các chân INT0 và INT1 là các ngắt theo mức thấp. Để biến các chân này trở thành các ngắt theo sườn thì chúng ta phải viết chương trình cho các bit của thanh ghi TCON. Thanh ghi TCON giữ các bit cờ IT0 và IT1 xác định chế độ ngắt theo sườn hay ngắt theo mức của các ngắt phần cứng IT0 và IT1 là các bit D0 và D2 của thanh ghi TCON tương ứng. Chúng có thể được biểu diễn như TCON.0 và TCON.2 vì thanh ghi TCON có thể đánh địa chỉ theo bit. Khi bật lại nguồn thì TCON.0 (IT0) và TCON.2 (IT1) đều ở mức thấp (0) nghĩa là các ngắt phần cứng ngoài của các chân INT0 và INT1 là ngắt theo mức thấp. Bằng việc chuyển các bit TCON.0 và TCON.2 lên cao qua các lệnh “SETB TCON.0” và “SETB TCON.2” thì các ngắt phần cứng ngoài INT0 và INT1 trở thành các ngắt theo sườn. Ví dụ, lệnh “SETB TCON.2” làm cho INT1 mà được gọi là ngắt theo sườn trong đó khi một tín hiệu chuyển từ cao xuống thấp được cấp đến chân P3.3 thì ở trường hợp này bộ vi điều khiển sẽ bị ngắt và bị cưỡng bức nhảy đến bảng véo ngắt tại địa chỉ 0013H để thực hiện trình phục vụ ngắt. Tuy nhiên là với giải thiết rằng bit ngắt đã được cho phép trong thanh ghi IE.

D7

D0

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
-----	-----	-----	-----	-----	-----	-----	-----

Hình 11.6: Thanh ghi TCON.

- Bít TF1 hay TCON.7 là cờ tràn của bộ Timer1. Nó được lập bởi phần cứng khi bộ đếm/ bộ định thời 1 tràn, nó được xoá bởi phần cứng khi bộ xử lý chỉ đến trình phục vụ ngắt.
- Bít TR1 hay TCON.6 là bít điều khiển hoạt động của Timer1. Nó được thiết lập và xoá bởi phần mềm để bật/ tắt Timer1.
- Bít TF0 hay TCON.5 tương tự như TF1 dành cho Timer0.
- Bít TR0 hay TCON.4 tương tự như TR1 dành cho Timer0.
- Bít IE1 hay TCON.3 cờ ngắt ngoài 1 theo sườn. Nó được thiết lập bởi CPU khi sườn ngắt ngoài (chuyển từ cao xuống thấp) được phát hiện. Nó được xoá bởi CPU khi ngắt được xử lý. Lưu ý: Cờ này không chốt những ngắt theo mức thấp.
- Bít IT1 hay TCON.2 là bít điều khiển kiểu ngắt. Nó được thiết lập và xoá bởi phần mềm để xác định kiểu ngắt ngoài theo sườn xuống hay mức thấp.
- Bít IE0 hay TCON.1 tương tự như IE1 dành cho ngắt ngoài 0.
- Bít IT0 hay TCON.0 tương tự như bít IT1 dành cho ngắt ngoài 0.

Xét ví dụ 11.6, chú ý rằng sự khác nhau duy nhất giữa ví dụ này và ví dụ 11.5 là ở trong hàng đầu tiên của MAIN khi lệnh “SETB TCON.2” chuyển ngắt INT1 về kiểu ngắt theo sườn. Khi sườn xuống của tín hiệu được cấp đến chân INT1 thì đèn LED sẽ bật lên một lúc. Đèn LED có thời gian sáng phụ thuộc vào độ trễ bên trong ISR của INT1. Để bật lại đèn LED thì phải có một sườn xung xuống khác được cấp đến chân P3.3. Điều này ngược với ví dụ 11.5. Trong ví dụ 11.5 do bản chất ngắt theo mức của ngắt thì đèn LED còn sáng chừng nào tín hiệu ở chân INT1 vẫn còn ở mức thấp. Nhưng trong ví dụ này để bật lại đèn LED thì xung ở chân INT1 phải được đưa lên cao rồi sau đó bị hạ xuống thấp để tạo ra một sườn xuống làm kích hoạt ngắt.

Ví dụ 11.6:

Giả thiết chân P3.3 (INT1) được nối với một máy tạo xung, hãy viết một chương trình trong đó sườn xuống của xung sẽ gửi một tín hiệu cao đến chân P1.3 đang được nối tới đèn LED (hoặc một còi báo). Hay nói cách khác, đèn LED được bật và tắt cùng tần số với các xung được cấp tới chân INT1. Đây là phiên bản ngắt theo sườn xung của ví dụ 11.5 đã trình bày ở trên.

Lời giải:

```

ORG    0000H
LJMP   MAIN
; - - Trình phục vụ ngắt ISR dành cho ngắt INT1 để bật đèn LED
ORG    0013H                ; Nhảy đến địa chỉ của trình phục vụ ngắt INT1
SETB   P1.3                 ; Bật đèn LED (hoặc còi)
MOV     R3, #225
BACK:   DJNZ  R3, HERE        ; giữ đèn LED (hoặc còi) một lúc
CLR     P1.3                 ; Tắt đèn LED (hoặc còi)
RETI                                ; Quay trở về từ ngắt
; - - Bắt đầu chương trình chính
ORG     30H
SETB    TCON.2               ; Chuyển ngắt INT1 về kiểu ngắt theo sườn xung

```

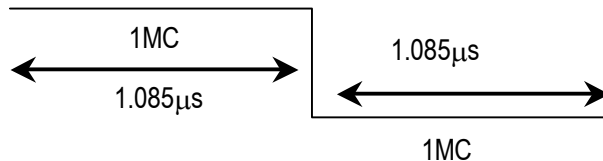
```

MOV    IE, #10001B    ; Cho phép ngắt ngoài INT1
HERE:  SJMP  HERE      ; Dừng ở đây cho đến khi bị ngắt
END

```

11.3.5 Trình mẫu ngắt theo sườn.

Trước khi kết thúc phần này ta cần trả lời câu hỏi vậy thì ngắt theo sườn được trích mẫu thường xuyên như thế nào? Trong các ngắt theo sườn, nguồn ngoài phải giữ ở mức cao tối thiểu là một chu trình máy nữa để đảm bảo bộ vi điều khiển nhìn thấy được sự chuyển dịch từ cao xuống thấp của sườn xung.



Thời hạn xung tối thiểu để phát hiện ra các ngắt theo sườn xung với tần số XTAL = 11.0592MHz

Sườn xuống của xung được chốt bởi 8051 và được giữ bởi thanh ghi TCON. Các bit TCON.1 và TCON.3 giữ các sườn được chốt của chân INT0 và INT1 tương ứng. TCON.1 và TCON.3 cũng còn được gọi là các bit IE0 và IE1 như chỉ ra trên hình 11.6. Chúng hoạt động như các cờ “ngắt đang được phục vụ” (Interrupt-in-server). Khi một cờ “ngắt đang được phục vụ” bật lên thì nó báo cho thế giới thực bên ngoài rằng ngắt hiện nay đang được xử lý và trên chân INTn này sẽ không có ngắt nào được đáp ứng chừng nào ngắt này chưa được phục vụ xong. Đây giống như tín hiệu báo bận ở máy điện thoại. Cần phải nhấn mạnh hạt điểm dưới đây khi quan tâm đến các bit IT0 và IT1 của thanh ghi TCON.

1. Khi các trình phục vụ ngắt ISR kết thúc (nghĩa là trong thanh ghi thực hiện lệnh RETI). Các bit này (TCON.1 và TCON.3) được xóa để báo rằng ngắt được hoàn tất và 8051 sẵn sàng đáp ứng ngắt khác trên chân đó. Để ngắt khác được nhận và thì tín hiệu trên chân đó phải trở lại mức cao và sau đó nhảy xuống thấp để được phát hiện như một ngắt theo sườn.
2. Trong thời gian trình phục vụ ngắt đang được thực hiện thì chân INTn bị làm ngưng không quan tâm đến nó có bao nhiêu lần chuyển dịch từ cao xuống thấp. Trong thực tế nó là một trong các chức năng của lệnh RETI để xóa bit tương ứng trong thanh ghi TCON (bit TCON.1 và TCON.3). Nó báo cho ta rằng trình phục vụ ngắt xấp kết thúc. Vì lý do này mà các bit TCON.1 và TCON.3 được gọi là các cơ báo “ngắt đang được phục vụ” cờ này sẽ lên cao khi một sườn xuống được phát hiện trên chân INT và dừng ở mức cao trong toàn bộ quá trình thực hiện ISR. Nó chỉ bị xóa bởi lệnh RETI là lệnh cuối cùng của ISR. Do vậy, sẽ không bao giờ cần đến các lệnh xóa bit này như “CLR TCON.1” hay “CLR TCON.3” trước lệnh RETI trong trình phục vụ ngắt đối với các ngắt cứng INT0 và INT1. Điều này không đúng với trường hợp của ngắt nối tiếp.

Ví dụ 11.7:

Sự khác nhau giữa các lệnh RET và RETI là gì? Giải thích tại sao ta không thể dùng lệnh RET thay cho lệnh RETI trong trình phục vụ ngắt.

Lời giải:

Các hai lệnh RET và RETI đều thực thi các hành vi giống nhau là lấy hai byte trên đỉnh ngăn xếp vào bộ đếm chương trình và đưa 8051 trở về nơi đó đã bỏ đi. Tuy nhiên, lệnh RETI còn thực thi một nhiệm vụ khác nữa là xoá cờ “ngắt đang được phục vụ” để báo rằng ngắt đã kết thúc và 8051 có thể nhập một ngắt mới trên chân này. Nếu ta dùng lệnh RET thay cho RETI như là lệnh cuối cùng của trình phục vụ ngắt như vậy là ta đã vô tình khoá mọi ngắt mới trên chân này sau ngắt đầu tiên vì trạng thái của chân báo rằng ngắt vẫn đang được phục vụ. Đây là trường hợp mà các cờ TF0, TF1, TCON.1 và TCON.3 được xoá bởi lệnh RETI.

11.3.6 Vài điều bổ xung về thanh ghi TCON.

Bây giờ ta xét kỹ về các bit của thanh ghi TCON để hiểu vai trò của nó trong việc duy trì các ngắt.

11.3.6.1 Các bit IT0 và IT1.

Các bit TCON.0 và TCON.2 được coi như là các bit IT0 và IT1 tương ứng. Đây là các bit xác định kiểu ngắt theo sườn xung hay theo mức xung của các ngắt phân cứng trên chân INT.0 và INT.1 tương ứng. Khi bật lại nguồn cả hai bit này đều có mức 0 để biến chúng thành ngắt theo tín hiệu mức thấp. Lập trình viên có thể điều khiển một trong số chúng lên cao để chuyển ngắt phân cứng bên ngoài thành ngắt theo ngưỡng. Trong một hệ thống dựa trên 8051 đã cho thì một khi ta đã đặt về 0 hoặc 1 thì các bit này sẽ không thay đổi vì người thiết kế đã cố định kiểu ngắt là ngắt theo sườn hay theo mức rồi.

11.3.6.2 Các bit IE0 và IE1.

Các bit TCON.1 và TCON.3 còn được gọi là IE0 và IE1 tương ứng. Các bit này được 8051 dùng để bám kiểu ngắt theo sườn xung. Nói khác là nếu IT0 và IT1 bằng 0 thì có nghĩa là các ngắt phân cứng là ngắt theo mức thấp, các bit IE0 và IE1 không dùng đến làm gì. Các bit IE0 và IE1 được 8051 chỉ dùng để chốt sườn xung từ cao xuống thấp trên các chân INT0 và INT1. Khi có chuyển dịch sườn xung trên chân INT0 (hay INT1) thì 8051 đánh dấu (bật lên cao) các bit IEx trên thanh ghi TCON nhảy đến bảng vec tơ ngắt và bắt đầu thực hiện trình phục vụ ngắt ISR. Trong khi 8051 thực hiện ISR thì không có một sườn xung nào được ghi nhận trên chân INT0 (hay INT1) để ngăn mọi ngắt trong ngắt. Chỉ trong khi thực hiện lệnh RETI ở cuối trình phục vụ ngắt ISR thì các bit IEx mới bị báo rằng một sườn xung cao xuống thấp mới trên chân INT0 (hay INT1) sẽ kích hoạt ngắt trở lại. Từ phần trình bày trên ta thấy rằng các bit IE0 và IE1 được 8051 sử dụng bên trong để báo có một ngắt đang được xử lý hay không. Hay nói cách khác là lập trình viên không phải quan tâm đến các bit này.

11.3.6.3 Các bit TR0 và TR1.

Đây là những bit D4 và D6 (hay TCON.4 và TCON.6) của thanh ghi TCON. Các bit này đã được giới thiệu ở chương 9 chúng được dùng để khởi động và dừng các bộ định thời Timer0 và Timer1 tương ứng. Vì thanh ghi TCON có thể đánh địa chỉ theo bit nên có thể sử dụng các lệnh “SETB TRx” và “CLR TRx” cũng như các lệnh “SETB TCON.4” và “CLR TCON.4”.

11.3.6.4 Các bit TF0 và TF1.

Các bit này là D5 (TCON.5) và D7 (TCON.7) của thanh ghi TCON mà đã được giới thiệu ở chương 9. Chúng ta được sử dụng bởi các bộ Timer0 và Timer1 tương ứng để báo rằng các bộ định thời bị tràn hay quay về không. Mặc dù ta đã dùng các lệnh “JNB TFx, đích” và “CLR TFx” nhưng chúng ta cũng không thể sử

dụng các lệnh như “SETB TCON.5, đích” và “CLR TCON.5” vì TCON là thanh ghi có thể đánh địa chỉ theo bit.

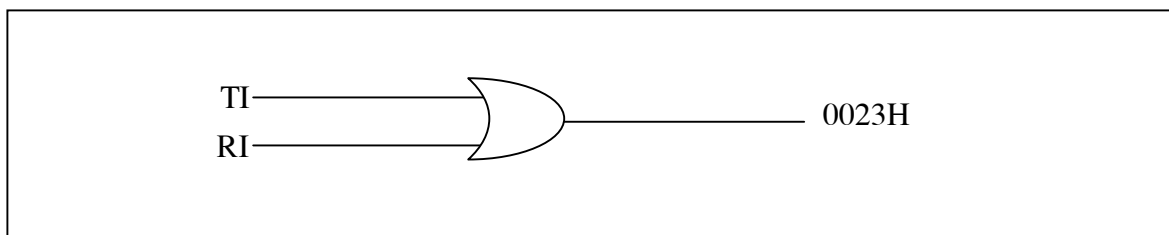
11.4 Lập trình ngắt truyền thông nối tiếp.

Trong chương 10 chúng ta đã nghiên cứu về truyền thông nối tiếp của 8051. Tất cả các ví dụ trong chương ấy đều sử dụng phương pháp thăm dò (polling). Ở chương này ta khám phá truyền thông dựa trên ngắt mà nó cho phép 8051 làm việc rất nhiều việc ngoài việc truyền và nhận dữ liệu từ cổng truyền thông nối tiếp.

11.4.1 Các cờ RI và TI và các ngắt.

Như đã nói ở chương 10 thì cờ ngắt truyền TI (Transfer interrupt) được bật lên khi bit cuối cùng của khung dữ liệu, bit stop được truyền đi báo rằng thanh ghi SBUF sẵn sàng truyền byte kế tiếp. Trong trường hợp cờ RI (Receive Interrupt) thì nó được bật lên khi toàn bộ khung dữ liệu kể cả bit stop đã được nhận. Hay nói cách khác khi thanh ghi SBUF đã có một byte thì cờ RI bật lên báo rằng byte dữ liệu nhận được cần lấy đi cất vào nơi an toàn trước khi nó bị mất (bị ghi đè) bởi dữ liệu mới nhận được. Chừng nào còn nói về truyền thông nối tiếp thì tất cả mọi khái niệm trên đây đều áp dụng giống như nhau cho dù sử dụng phương pháp thăm dò hay sử dụng phương pháp ngắt. Sự khác nhau duy nhất giữa hai phương pháp này là ở cách phục vụ quá trình truyền thông nối tiếp như thế nào. Trong phương pháp thăm dò thì chúng ta phải đợi cho cờ (TI hay RI) bật lên và trong lúc chờ đợi thì ta không thể làm gì được cả. Còn trong phương pháp ngắt thì ta được báo khi 8051 đã nhận được một byte hoặc nó sẵn sàng chuyển (truyền) byte kế tiếp và ta có thể làm các công việc khác trong khi truyền thông nối tiếp đang được phục vụ.

Trong 8051 chỉ có một ngắt dành riêng cho truyền thông nối tiếp. Ngắt này được dùng cho cả truyền và nhận dữ liệu. Nếu bit ngắt trong thanh ghi IE (là bit IE.4) được phép khi RI và TI bật lên thì 8051 nhận được ngắt và nhảy đến địa chỉ trình phục vụ ngắt dành cho truyền thông nối tiếp 0023H trong bảng véctơ ngắt để thực hiện nó. Trong trình ISR này chúng ta phải kiểm tra các cờ TI và RI để xem cờ nào gây ra ngắt để đáp ứng một cách phù hợp (xem ví dụ 11.8).



Hình 11.7: Ngắt truyền thông có thể do hai cờ TI và RI gọi.

11.4.2 Sử dụng cổng COM nối tiếp trong 8051.

Trong phần lớn các ứng dụng, ngắt nối tiếp chủ yếu được sử dụng để nhận dữ liệu và không bao giờ được sử dụng để truyền dữ liệu nối tiếp. Điều này giống như việc báo chuông để nhận điện thoại, còn nếu ta muốn gọi điện thoại thì có nhiều cách khác ngắt ta chứ không cần đến đồ chuông. Tuy nhiên, trong khi nhận điện thoại ta phải trả lời ngay không biết ta đang làm gì nếu không thuộc gọi sẽ (mất) đi qua. Tương tự như vậy, ta sử dụng các ngắt nối tiếp khi nhận dữ liệu đi đến để sao chép cho nó không bị mất: Hãy xét ví dụ 11.9 dưới đây.

Ví dụ 11.8:

Hãy viết chương trình trong đó 8051 đọc dữ liệu từ cổng P1 và ghi nó tới cổng P2 liên tục trong khi đưa một bản sao dữ liệu tới cổng COM nối tiếp để thực hiện truyền nối tiếp giả thiết tần số XTAL là 11.0592MHz và tốc độ baud là 9600.

Lời giải:

```

                ORG    0
                LJMP   MAIN
                ORG    23H
                LJMP   SERIAL                ; Nhảy đến trình phục vụ ngắt truyền thông nối tiếp
MAIN:          MOVQ P1, # 0FFH              ; Lấy cổng P1 làm cổng đầu vào
                MOV    TMOD, # 20h          ; Chọn Timer1, chế độ 2 tự nạp lại
                MOV    TH1, # 0FDH         ; Chọn tốc độ baud = 9600
                MOV    SCON, # 50H         ; Khung dữ liệu: 8 bit dữ liệu, 1 stop à cho phép REN
                MOV    IE, # 10010000B     ; Cho phép ngắt nối tiếp
                SETB   TR1                  ; Khởi động Timer1
BACK:          MOV    A, P1                ; Đọc dữ liệu từ cổng P1
                MOV    SBUF, A              ; Lấy một bản sao tới SBUF
                MOV    P2, A                ; Gửi nó đến cổng P2
                SJMP   BACK                 ; ở lại trong vòng lặp
;
; -----Trình phục vụ ngắt cổng nối tiếp
                ORG    100H
SERIAL:        JB     TI, TRANS             ; Nhảy đến cờ TI cao
                MOV    A, SBUF              ; Nếu không tiếp tục nhận dữ liệu
                CLR    RI                   ; Xoá cờ RI vì CPU không làm điều này
                RETI                        ; Trở về từ trình phục vụ ngắt
TRANS:         CLR    TI                   ; Xoá cờ TI vì CPU không làm điều này
                RETI                        ; Trở về từ ISR
                END

```

Trong vấn đề trên thấy chú ý đến vai trò của cờ TI và RI. Thời điểm một byte được ghi vào SBUF thì nó được đóng khung và truyền đi nối tiếp. Kết quả là khi bit cuối cùng (bit stop) được truyền đi thì cờ TI bật lên cao và nó gây ra ngắt nối tiếp được gọi khi bit tương ứng của nó trong thanh ghi IE được đưa lên cao. Trong trình phục vụ ngắt nối tiếp, ta phải kiểm tra cả cờ TI và cờ RI vì cả hai đều có thể gọi ngắt hay nói cách khác là chỉ có một ngắt cho cả truyền và nhận.

Ví dụ 11.9:

Hãy viết chương trình trong đó 8051 nhận dữ liệu từ cổng P1 và gửi liên tục đến cổng P2 trong khi đó dữ liệu đi vào từ cổng nối tiếp COM được gửi đến cổng P0. Giả thiết tần số XTAL là 11.0592MHz và tốc độ baud 9600.

Lời giải:

```

                ORG    0
                LJMP   MAIN
                ORG    23H
                LJMP   SERIAL                ; Lấy cổng P1 là cổng đầu vào
                ORG    03H
MAIN:          MOV    P1, # FFH
                MOV    TMOD, # 20H          ; Chọn Timer và chế độ hai tự nạp lại
                MOV    TH1, # 0FDH         ; Khung dữ liệu: 8 bit dữ liệu, 1 stop, cho phép REN

```

```

MOV    SCON, # 50H          ; Cho phép ngắt nối tiếp
MOV    IE, # 10010000B      ; Khởi động Timer1
SETB   TR1                  ; Đọc dữ liệu từ cổng P1
BACK:  MOV    A, P1          ; Gửi dữ liệu đến cổng P2
        MOV    P2, A        ; ở lại trong vòng lặp
        SJMP   BACK

; -----Trình phục vụ ngắt cổng nối tiếp.
ORG    100H
SERIAL: JB    TI, TRANS      ; Nhảy nếu Ti cao
        MOV    A, SBUF       ; Nếu không tiếp tục nhận dữ liệu
        MOV    P0, A        ; Gửi dữ liệu đầu vào đến cổng P0
        CLR    RI           ; Xóa cờ RI vì CPU không xóa cờ này
        RETI                ; Trở về từ ISR
TRANS:  CLS    TI           ; Xóa cờ TI và CUP không xóa cờ này.
        RETI                ; ; trở về từ ISR
END

```

11.4.3 Xóa cờ RI và TI trước lệnh RETI.

Để ý rằng lệnh cuối cùng trước khi trở về từ ISR là RETI là lệnh xóa các cờ RI và TI. Đây là điều cần thiết bởi vì đó là ngắt duy nhất dành cho nhận và truyền 8051 không biết được nguồn gây ra ngắt là nguồn nào, do vậy trình phục vụ ngắt phải được xóa các cờ này để cho phép các ngắt sau đó được đáp ứng sau khi kết thúc ngắt. Điều này tương phản với ngắt ngoài và ngắt bộ định thời đều được 8051 xóa các cờ. Các lệnh xóa các cờ ngắt bằng phần mềm qua các lệnh “CLR TI” và “CLR RI”. Hãy xét ví dụ 11.10 dưới đây và để ý đến các lệnh xóa cờ ngắt trước lệnh RETI.

Ví dụ 11.10:

Hãy viết một chương trình sử dụng các ngắt để thực hiện các công việc sau:

- Nhận dữ liệu nối tiếp và gửi nó đến cổng P0.
- Lấy cổng P1 đọc và truyền nối tiếp và sao đến cổng P2.
- Sử dụng Timer0 tạo sóng vuông tần số 5kHz trên chân P0.1 giải thiết tần số XTAL = 11.0592MHz và tốc độ baud 4800.

Lời giải:

```

ORG    0
LJMP   MAIN
ORG    000BH          ; Trình phục vụ ngắt dành cho Timer0
CPL    P0.1           ; Tạo xung ở chân P0.1
RETI                ; Trở về từ ISR
ORG    23H            ; Nhảy đến địa chỉ ngắt truyền nối tiếp
LJMP   SERIAL         ; Lấy cổng P1 làm cổng đầu vào
ORG    30H
MAIN :  MOV    P1, # 0FFH ; Chọn Timer0 và Timer1 chế độ 2 tự nạp lại
        MOV    TMOD, # 22H ; Chọn Timer0 và Timer1 chế độ 2 tự nạp lại
        MOV    TH1, # 0F6H ; Chọn tốc độ baud 4800
        MOV    SCON, # 50H ; Khung dữ liệu: 8 bit dữ liệu, 1 stop, cho phép REN
        MOV    TH0, # - 92 ; Tạo tần số 5kHz
        MOV    IE, # 10010010B ; Cho phép ngắt nối tiếp
        SETB   TR1        ; Khởi động Timer1
        SETB   TR0        ; Khởi động Timer0
BACK:  MOV    A, P1        ; Đọc dữ liệu từ cổng P1
        MOV    SBUF, A    ; Lấy một lần bản sao dữ liệu
        MOV    P2, A      ; Ghi nó vào cổng P2

```

```

                SJMP    BACK                ; ở lại trong vòng lặp
; ----- Trình phục vụ ngắt cổng nối tiếp.
                ORG     100H
SERIAL:         JB      TI, TRANS          ; Nhảy nếu TI vào
                MOV     A, SBUF            ; Nếu không tiếp tục nhận dữ liệu
                MOV     P0, A              ; Gửi dữ liệu nối tiếp đến P0
                CLR     RI                  ; Xoá cờ RI vì 8051 không làm điều này
                RETI                         ; Trở về từ ISR
TRANS:          CLR     TI                  ; Xoá cờ TI vì 8051 không xoá
                RETI                         ; Trở về từ ISR.
                END

```

Trước khi kết thúc phần này hãy để ý đến danh sách tất cả mọi cờ ngắt được cho trong bảng 11.2. Trong khi thanh ghi TCON giữ 4 cờ ngắt còn hai cờ TI và RI ở trong thanh ghi SCON của 8051.

Bảng 11.2: Các bit cờ ngắt.

Ngắt	Cờ	Bit của thanh ghi SFR
Ngắt ngoài 0	IE0	TCON.1
Ngắt ngoài 1	IE1	TCON.3
Ngắt Timer0	TF0	TCON.5
Ngắt Timer1	TF1	TCON.7
Ngắt cổng nối tiếp	TI	SCON.1
Ngắt Timer2	TF2	T2CON.7 (TA89C52)
Ngắt Timer2	EXF2	T2CON.6 (TA89C52)

11.5 Các mức ưu tiên ngắt trong 8051.

11.5.1 Các mức ưu tiên trong quá trình bật lại nguồn.

Khi 8051 được cấp nguồn thì các mức ưu tiên ngắt được gán theo bảng 11.3. Từ bảng này ta thấy ví dụ nếu các ngắt phần cứng ngoài 0 và 1 được kích hoạt cùng một lúc thì ngắt ngoài 0 sẽ được đáp ứng trước. Chỉ sau khi ngắt INT0 đã được phục vụ xong thì INT1 mới được phục vụ vì INT1 có mức ưu tiên thấp hơn. Trong thực tế sơ đồ mức ưu tiên ngắt trong bảng không có ý nghĩa gì cả mà một quy trình thăm dò trong đó 8051 thăm dò các ngắt theo trình tự cho trong bảng 11.3 và đáp ứng chúng một cách phù hợp.

Bảng 11.3: Mức ưu tiên các ngắt trong khi cấp lại nguồn.

Mức ưu tiên cao xuống thấp	
Ngắt ngoài 0	INT0
Ngắt bộ định thời 0	TF0
Ngắt ngoài 1	INT1
Ngắt bộ định thời 1	TF1
Ngắt truyền thông nối tiếp	(RI + TI)

Ví dụ 11.1:

Hãy bình luận xem điều gì xảy ra nếu các ngắt INT0, TF0 và INT1 được kích hoạt cùng một lúc. Giả thiết rằng các mức ưu tiên được thiết lập như khi bật lại nguồn và các ngắt ngoài là ngắt theo sườn xung.

Lời giải:

Nếu ba ngắt này được kích hoạt cùng một thời điểm thì chúng được chốt và được giữ ở bên trong. Sau đó kiểm tra tất cả năm ngắt theo trình tự cho trong bảng 11.3. Nếu một ngắt bất kỳ được kích hoạt thì nó được phục vụ theo trình tự. Do vậy, khi cả ba ngắt trên đây cùng được kích hoạt một lúc thì ngắt ngoài 0 (IE0) được phục vụ trước hết sau đó đến ngắt Timer0 (TF0) và cuối cùng là ngắt ngoài 1 (IE1).

D7		D0					
--	--	PT2	PS	PT1	PX1	PT0	PX0

Hình 11.8: Thanh ghi mức ưu tiên ngắt IP, bit ưu tiên = 1 là mức ưu tiên cao, bit ưu tiên = 0 là mức ưu tiên thấp.

- Bit D7 và D6 hay IP.7 và IP.6 - chưa dùng.
- Bit D5 hay IP.5 là bit ưu tiên ngắt Timer2 (dùng cho 8052)
- Bit D4 hay IP.4 là bit ưu tiên ngắt cổng nối tiếp
- Bit D3 hay IP.3 là bit ưu tiên ngắt Timer1
- Bit D2 hay IP.2 là mức ưu tiên ngắt ngoài 1
- Bit D1 hay IP.1 là mức ưu tiên ngắt Timer 0
- Bit D0 hay IP.0 là mức ưu tiên ngắt ngoài 0

Người dùng không được viết phần mềm ghi các số 1 vào các bit chưa dùng vì chúng dành cho các ứng dụng tương lai.

11.5.2 Thiết lập mức ưu tiên ngắt với thanh ghi IP.

Chúng ta có thể thay đổi trình tự trong bảng 11.3 bằng cách gán mức ưu tiên cao hơn cho bất kỳ ngắt nào. Điều này được thực hiện bằng cách lập trình một thanh ghi gọi là thanh ghi mức ưu tiên ngắt IP (Interrupt Priority). Trên hình 11.8 là các bit của thanh ghi này, khi bật lại nguồn thanh ghi IP chứa hoàn toàn các số 0 để tạo ra trình tự ưu tiên ngắt theo bảng 11.3. Để một ngắt nào đó mức ưu tiên cao hơn ta thực hiện đưa bit tương ứng lên cao. Hãy xem ví dụ 11.12.

Một điểm khác nữa cần được làm sáng tỏ là mức ưu tiên ngắt khi hai hoặc nhiều bit ngắt trong thanh ghi IP được đặt lên cao. Trong trường hợp này thì trong khi các ngắt này có mức ưu tiên cao hơn các ngắt khác chúng sẽ được phục vụ theo trình tự cho trong bảng 11.3. Xem ví dụ 11.13.

Ví dụ 11.12:

- a) Hãy lập trình thanh ghi IP để gán mức ưu tiên cao nhất cho ngắt INT1 (ngắt ngoài 1) sau đó.
- b) Hãy phân tích điều gì xảy ra khi INT0, INT1 và TF0 được kích hoạt cùng lúc. Giả thiết tất cả các ngắt đều là các ngắt theo sườn.

Lời giải:

- a) MOV IP, #0000 0100B ; Đặt bit IP.2 = 1 để gán INT1 mức ưu tiên cao nhất. Lệnh “SETB IP.2” cũng tác động tương tự bởi vì IP là thanh ghi có thể đánh địa chỉ theo bit.
- b) Lệnh trong bước a) gán mức ưu tiên cao hơn INT1 so với các ngắt khác, do vậy khi INT0, INT1 và TF0 được kích hoạt cùng lúc thì trước hết INT1 được phục vụ

trước rồi sau đó đến INT0 và cuối cùng là TF0. Điều này là do INT1 có mức ưu tiên cao hơn hai ngắt kia ở bước a). Sau khi thực hiện xong ngắt INT1 thì 8051 trở về phục vụ ngắt còn lại theo trình tự ưu tiên trong bảng 11.3.

Ví dụ 11.13:

Giả thiết rằng sau khi bật lại nguồn thì mức ưu tiên ngắt được thiết lập bởi lệnh “MOV IP, #0000 1100B”. Hãy bình luận về quá trình các ngắt được phục vụ như thế nào?

Lời giải:

Lệnh “MOV IP, #0000 1100B” (chữ B là giá trị thập phân) thiết lập ngắt ngoài (INT1) và ngắt bộ Timer1 (TF1) có mức ưu tiên cao hơn các ngắt khác. Tuy nhiên, vì chúng được thăm dò theo bảng 11.3 nên chúng sẽ được phục vụ theo trình tự sau:

Mức ưu tiên cao nhất: Ngắt ngoài 1 (INT1)
Ngắt bộ Timer 1 (TF1)
Ngắt ngoài 0 (INT0)
Ngắt bộ Timer0 (TF0)

Mức ưu tiên thấp nhất: Ngắt cổng truyền thông nối tiếp (RI + RT).

11.5.3 Ngắt trong ngắt.

Điều gì xảy ra nếu 8051 đang thực hiện một trình phục vụ ngắt thuộc một ngắt nào đó thì lại có một ngắt khác được kích hoạt? Trong những trường hợp như vậy thì một ngắt có mức ưu tiên cao hơn có thể ngắt một ngắt có mức ưu tiên thấp hơn. Đây gọi là ngắt trong ngắt. Trong 8051 một ngắt ưu tiên thấp có thể bị ngắt bởi một ngắt có mức ưu tiên cao hơn chứ không bị ngắt bởi một ngắt có mức ưu tiên thấp hơn. Mặc dù tất cả mọi ngắt đều được chốt và gửi bên trong nhưng không có ngắt mức thấp nào được CPU quan tâm ngay tức khắc nếu 8051 chưa kết thúc phục vụ các ngắt mức cao.

11.5.4 Thu chop ngắt bằng phần mềm (Triggering).

Có nhiều lúc ta cần kiểm tra một trình phục vụ ngắt bằng con đường mô phỏng. Điều này có thể được thực hiện bằng các lệnh đơn giản để thiết lập các ngắt lên cao và bằng cách đó buộc 8051 nhảy đến bảng véc tơ ngắt. Ví dụ, nếu bit IE dành cho bộ Timer1 được bật lên 1 thì một lệnh như “SETB TF1” sẽ ngắt 8051 ngừng thực hiện công việc đang làm bất kỳ và buộc nó nhảy đến bảng véc tơ ngắt. Hay nói cách khác, ta không cần đợi cho Timer1 quay trở về 0 mới tạo ra ngắt. Chúng ta có thể gây ra một ngắt bằng các lệnh đưa các bit của ngắt tương ứng lên cao.

Như vậy ở chương này chúng ta đã biết ngắt là một sự kiện bên trong hoặc bên ngoài gây ra ngắt bộ vi điều khiển để báo cho nó biết rằng thiết bị cần được phục vụ. Mỗi một ngắt có một chương trình đi kèm với nó được gọi là trình phục vụ ngắt ISR. Bộ vi điều khiển 8051 có sáu ngắt, trong đó năm ngắt người dùng có thể truy cập được. Đó là hai ngắt cho các thiết bị phần cứng bên ngoài INT0 và INT1, hai ngắt cho các bộ định thời là TF0 và TF1 và ngắt dành cho truyền thông nối tiếp.

8051 có thể được lập trình cho phép hoặc cấm một ngắt bất kỳ cũng như thiết lập mức ưu tiên cho nó theo yêu cầu của thuật toán ứng dụng.