



KHOA VÔ TUYẾN ĐIỆN TỬ

BỘ MÔN KỸ THUẬT VI XỬ LÝ

KỸ THUẬT VI XỬ LÝ VÀ LẬP TRÌNH HỢP NGỮ

Giáo viên: Nguyễn Khoa Sang

HỌC VIỆN KỸ THUẬT QUÂN SỰ - 2016

Chương 6. Lập trình OnChip 80C51 và 89C51

Họ vi điều khiển 8051[?]

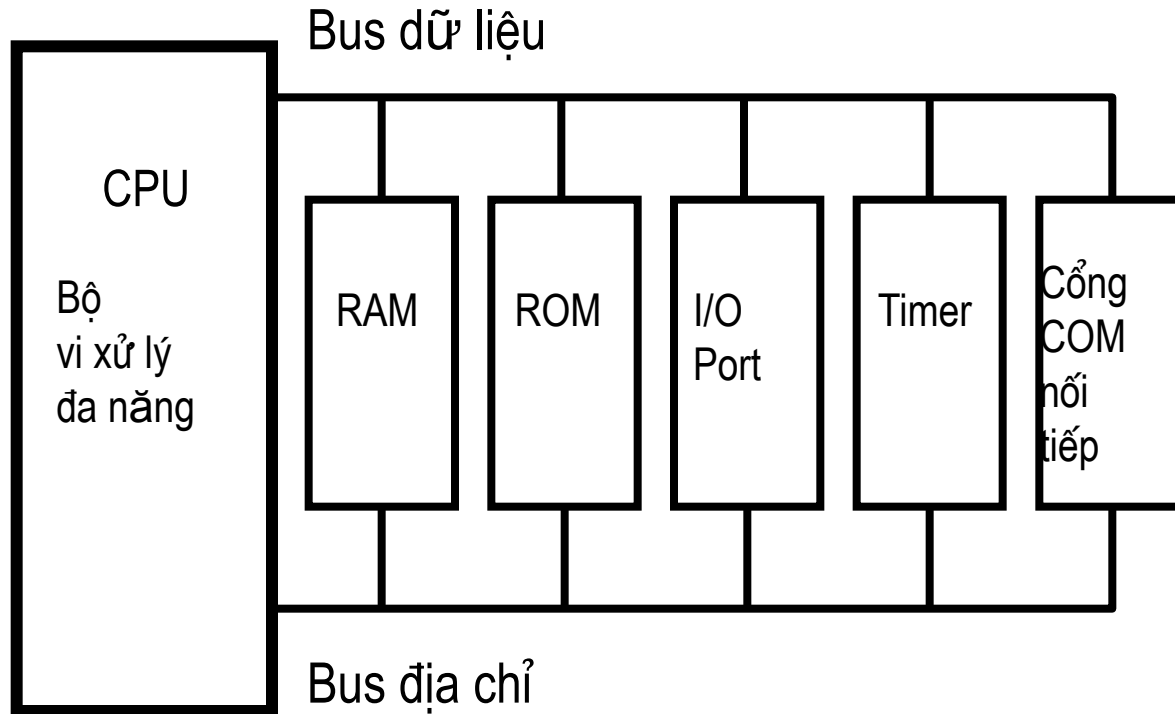
NỘI DUNG

1. Bộ vi điều khiển và hệ nhúng

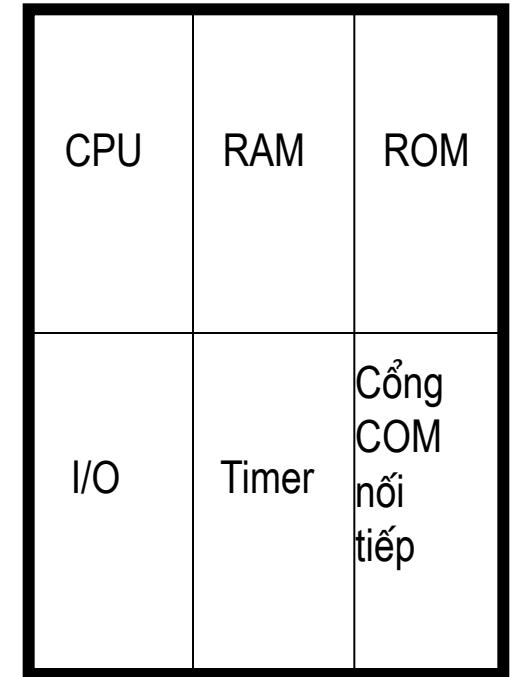
2. Tổng quan họ 8051

1. Bộ vi điều khiển và hệ nhúng

a) Bộ vi điều khiển và bộ vi xử lý



(a) Hệ vi xử lý



(b) Bộ vi điều khiển

Hình 1.1. Hệ vi xử lý và vi điều khiển

*** Bộ VXL đa năng**

- Bộ VXL đa năng như: 80486, Pentium, 68040
- Hệ thống với bộ vi xử lý đa năng, cần được bổ sung RAM, ROM, cổng I/O và bộ định thời.
- Ưu: + Tính linh hoạt cao.
 - + Tùy theo từng ứng dụng để quyết định cấu hình, như chọn dung lượng RAM, ROM, cổng vào ra phù hợp.
- Nhược: Cổng kênh hơn, đắt tiền hơn

*** Bộ vi điều khiển:**

- Có sẵn trên chip bộ vi xử lý, bộ nhớ RAM, ROM, cổng vào ra và bộ định thời.
- Thích hợp trong nhiều ứng dụng với giá rẻ, không gian sử dụng nhỏ.
- Ví dụ ứng dụng: Bộ điều khiển TV, điều khiển quạt, điều khiển máy giặt...
- Một số nhà sản xuất đã tích hợp vào trong bộ vi điều khiển bộ ADC và một số thiết bị ngoại vi khác.

b) Hệ thống nhúng (hệ nhúng)

- **Hệ nhúng** (embedded system) là một hệ chuyên dụng được sử dụng để thực hiện một hoặc một số chức năng cụ thể.
- Hệ nhúng được hiểu là thiết bị “nhúng” trong một hệ thống.
- Bộ vi xử lý và vi điều khiển được sử dụng rộng rãi trong nhiều hệ nhúng.

Ví dụ về các hệ nhúng

- Máy in là ví dụ về một hệ nhúng. Bộ xử lý chỉ được dùng cho công việc nhận và in dữ liệu.
- Bàn phím, modem, bộ điều khiển đĩa, thẻ âm, bộ điều khiển CD-ROM, chuột đều là các hệ nhúng
- Mỗi thiết bị ngoại vi đều có một bộ vi điều khiển để thực hiện chỉ một công việc là ví dụ về hệ nhúng.

Bảng 1.1. Một số sản phẩm nhúng dùng VĐK

Thiết bị gia đình	Các thiết bị điện Máy điện đàm Máy điện thoại Hệ thống an toàn Bộ mở cửa nhà xe Máy trả lời Máy Fax	Máy tính gia đình Tivi Truyền hình cáp VCR Máy quay camera Điều khiển từ xa Trò chơi điện tử	Nhạc cụ điện tử Máy may Điều khiển ánh sáng Đồ chơi Dụng cụ tập thể hình
Thiết bị văn phòng	Điện thoại Máy tính Hệ thống an toàn	Máy Fax Lò vi sóng Máy sao chụp	Máy in lazer Máy in màu Máy nhắn tin
Thiết bị tự động	Máy tính hành trình Điều khiển động cơ Túi đệm khí	Đo lường Hệ thống bảo mật Điều khiển truyền tin Giải trí	Điều hoà nhiệt độ Điện thoại Chìa khoá điện tử

2. Tổng quan họ 8051

- a) Sơ lược sự phát triển 8051**
- b) Vi điều khiển 8051**
- c) Các phiên bản khác của 8051**

a) Sơ lược sự phát triển 8051

- 8051 do Intel sản xuất năm 1981.
- Là bộ vi xử lý 8 bit và làm cơ sở cho các hệ VĐK sau này
- 8051 tích hợp trên chip bộ nhớ ROM, RAM, bộ định thời, các cổng vào ra, cổng nối tiếp, ngắt...
- Hệ vi điều khiển còn được gọi "System on chip" (Hệ thống trên chip).

- Intel cho phép các hãng khác sản xuất chip dựa trên cấu trúc 8051 với điều kiện bảo đảm tính tương thích của các chương trình.

- Một số hãng sản xuất chip VĐK thông dụng

- Atmel
- Microchip Technology
- Dallas Semiconductor
- Siemens
- Zilog
- Philips
- Motorola
- AMD....

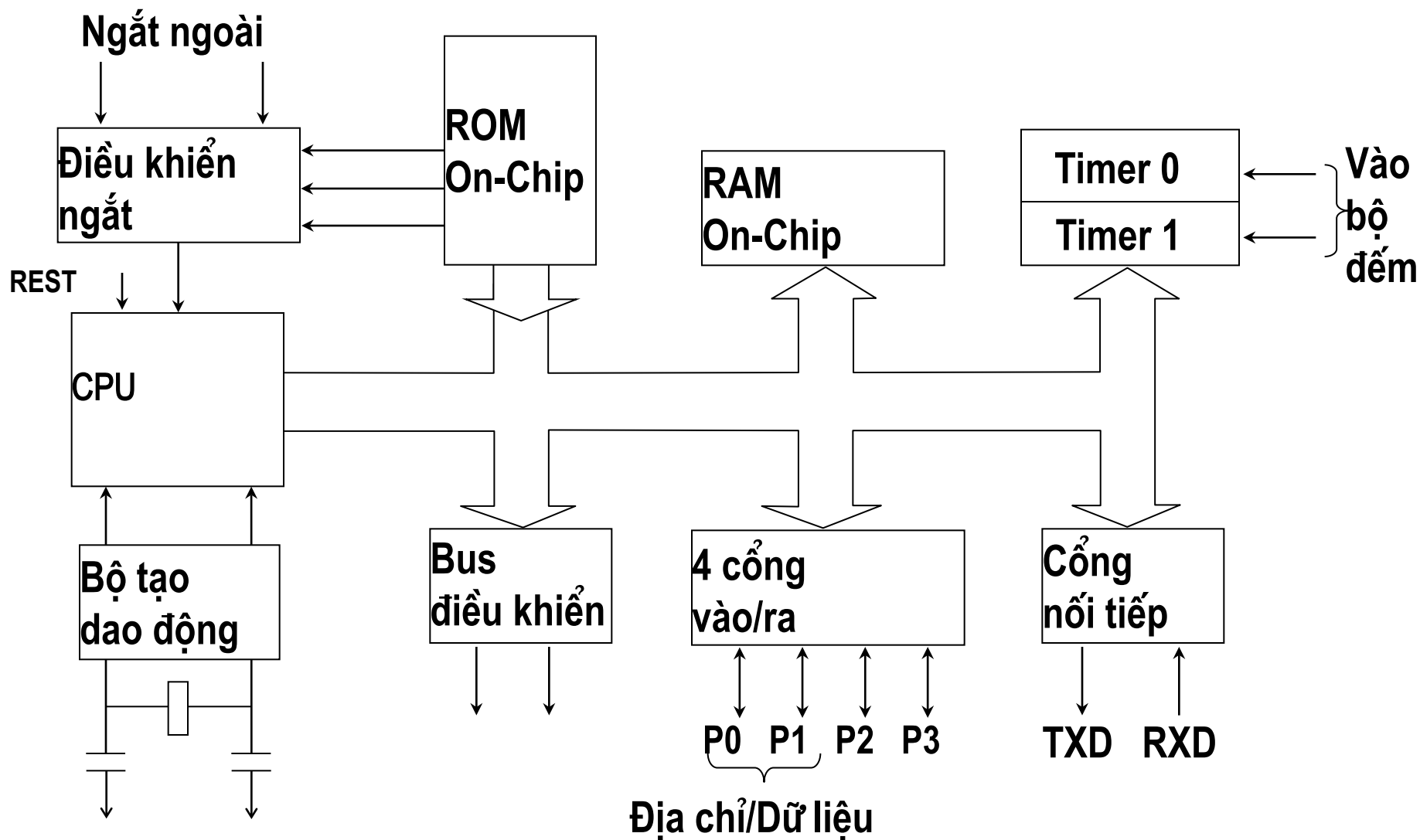
- Nhiều hãng đã bán thư viện VĐK cho ASIC.

b) Họ vi điều khiển 8051 của Intel

Vi điều khiển 8051

Thông số	Số lượng
ROM	4K byte
RAM	128 byte
Bộ định thời	2
Cổng I/O 8 bit	4 cổng 32 chân
Cổng nối tiếp	1
Nguồn ngắt	6

* Cấu trúc khối bộ vi điều khiển 8051



Hình 1.2. Sơ đồ khối bộ vi điều khiển 8051

06 nguồn ngắt của 8051

1. RESET

2. INT0

3. INT1

4. TIMER 0

5. TIMER 1

6. Cổng COM²

- **Bộ vi điều khiển 8052**

- **8052 là một thành viên của họ 8051.**

- **Thông số của 8052:**

- + 256 byte RAM**

- + 8 KB ROM**

- + 3 bộ định thời**

- + 8 nguồn ngắt**

*** Bộ vi điều khiển 8031**

- 8031 là phiên bản 8051 không ROM trên chip
- Để dùng 8031 cần bổ sung ROM ngoài
- ROM ngoài của 8031 có thể lên đến 64 KB
- Để truy cập hết bộ nhớ ROM ngoài, cần dùng hai cổng.
- 8031 còn lại hai cổng để sử dụng.

Tập hợp họ vi điều khiển 8051 của Intel

Thông số	8051	8052	8031
ROM	4K byte	8K byte	0
RAM	128 byte	256 byte	128 byte
Bộ định thời	2	3	2
Chân vào/ra	32	32	32
Cổng nối tiếp	1	1	1
Nguồn ngắt	6	8	6

c) Các phiên bản khác của 8051 * Bộ vi điều khiển 8751

- 4K byte bộ nhớ UV-EPROM (Ultraviolet) trên chip.
- Để sử dụng cần có bộ đốt PROM và bộ xoá UV-EPROM .
- Cần đến 20 phút để xoá ROM trước khi lập trình.
- Nhiều nhà sản xuất đã cho ra mắt phiên bản Flash ROM và NV-RAM (Non-volatile).

*** Bộ VĐK AT8951 của Atmel**

- Là phiên bản 8051 có bộ nhớ Flash ROM
- Không cần sử dụng bộ đốt PROM
- Phiên bản AT89C51 được lập trình qua cổng COM của máy tính IBM PC

Bảng 1.4. Các phiên bản 8051 của Atmel (Flash ROM)

Ký hiệu*	ROM	RAM	Chân I/O	Timer	Ngắt	Vcc	Đóng vỏ
AT89C51	4K	128	32	2	6	5V	40
AT89LV51	4K	128	32	2	6	3V	40
AT89C1051	1K	64	15	1	3	3V	20
AT89C2051	2K	128	15	2	6	3V	20
AT89C52	8K	128	32	3	8	5V	40
AT89LV52	8K	128	32	3	8	3V	40

Ghi chú: * Chữ C trong ký hiệu AT89C51 là CMOS.

*** Bộ vi điều khiển DS5000 của Dallas Semiconductor**

- Là một phiên bản phổ biến của 8051
- Bộ nhớ ROM trên chip là NV-RAM (Non-volatile)
- Có khả năng nạp chương trình vào ROM trên chip qua cổng COM máy tính trong khi chip vẫn được nối với hệ thống.
- NV-RAM cho phép thay đổi nội dung ROM theo từng byte (bộ nhớ Flash và EPROM phải được xoá hết trước khi lập trình lại).

Bảng 1.6. Các phiên bản 8051 của Dallas Semiconductor

Mô hình linh kiện	ROM	RAM	Chân I/O	Timer	Ngắt	Vcc	Đóng vỏ
DS5000-8	8K	128	32	2	6	5V	40
DS5000-32	32K	128	32	2	6	5V	40
DS5000T-8	8K	128	32	2	6	5V	40
DS5000T-32	32K	128	32	2	6	5V	40

Ghi chú: * Chữ “T” sau ký hiệu “5000” là có đồng hồ thời gian thực.

Bảng 1.7. Các phiên bản về tốc độ của DS5000

Tên chip	NV- RAM	Tốc độ
DS5000-8-8	8K	8MHz
DS5000-8-12	8K	12MHz
DS5000-32-8	32K	8MHz
DS5000T-32-12	32K	8MHz
DS5000-32-12	32K	12MHz
DS5000T-8-12	8K	12MHz

Họ 8051 của hãng Philips

- Philips Corporation có dải lựa chọn bộ vi điều khiển rất rộng.
- Nhiều sản phẩm của Philips tích hợp luôn một số chức năng nh:
 - Bộ biến đổi ADC
 - Bộ biến đổi DAC
 - Cổng I/O mở rộng
 - Các phiên bản OTP và Flash

Phiên bản OTP của 8051

- OTP (One Time Programmable) - Phiên bản (Lập trình một lần) của 8051.
- Các phiên bản Flash và NV-RAM thường được dùng để phát triển sản phẩm mẫu.
- Sau khi sản phẩm mẫu được hoàn tất thì sử dụng phiên bản OTP để sản xuất hàng loạt nhằm giảm giá thành.

Tập hợp các phiên bản của 8051

- **8751** là phiên bản 8051 có bộ nhớ UV-PROM (Ultraviolet)
- **AT89C51** là phiên bản Flash ROM của Atmel Corp, nạp qua cổng COM
- **DS5000** là phiên bản NV-RAM (Non-volatile RAM) của Dallas Semiconductor
- **Phiên bản OTP** (Lập trình một lần)

Tổ chức cổng vào/ra

Nội dung

1. Bố trí chân

2. Tổ chức cổng

3. Lập trình vào ra qua cổng

1. Bố trí chân của 8051

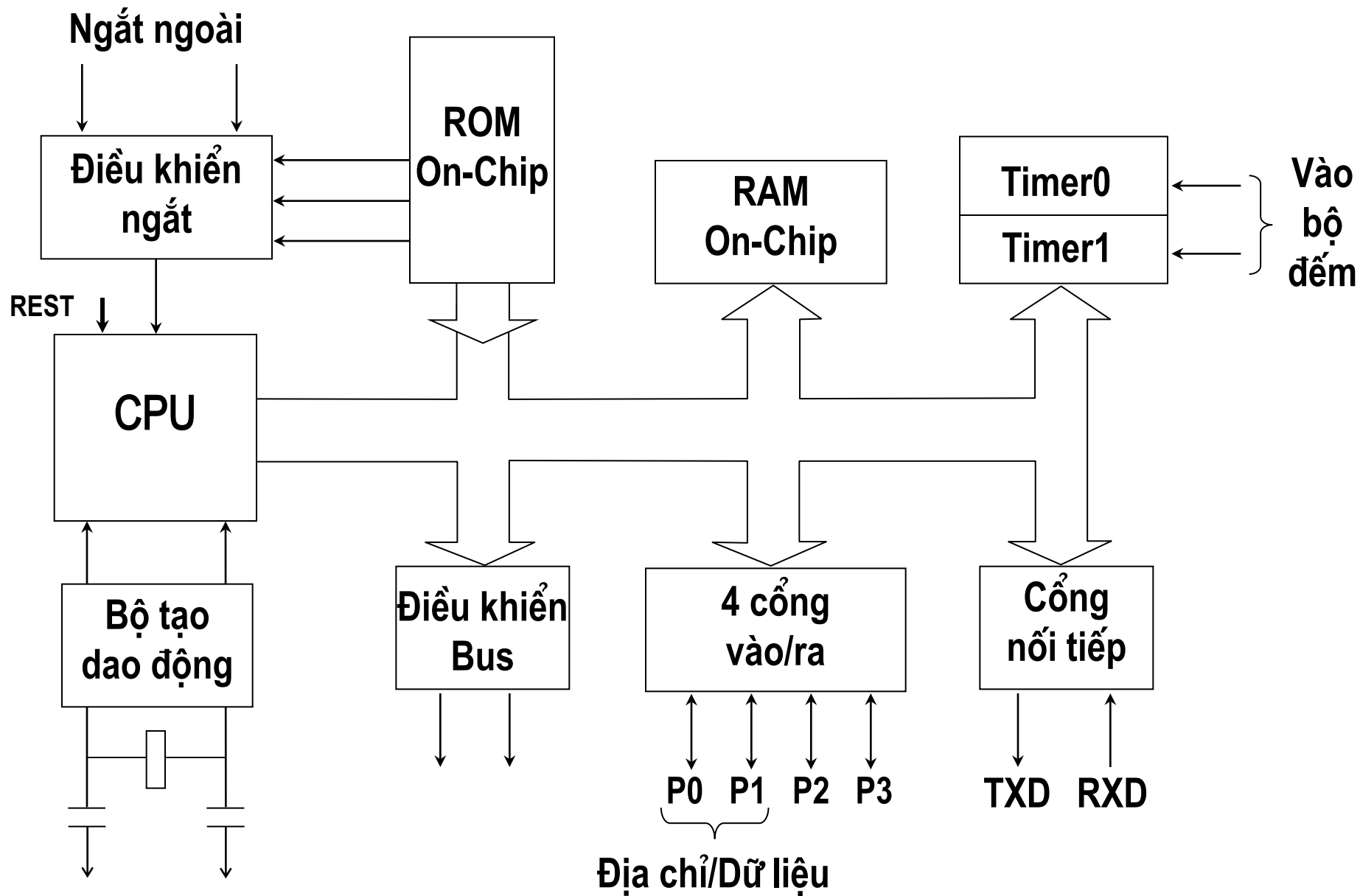
1.1 Tổ chức khối của 8051

1.2 Bố trí chân

1.1 Tổ chức khối của 8051

Có 7 khối chính:

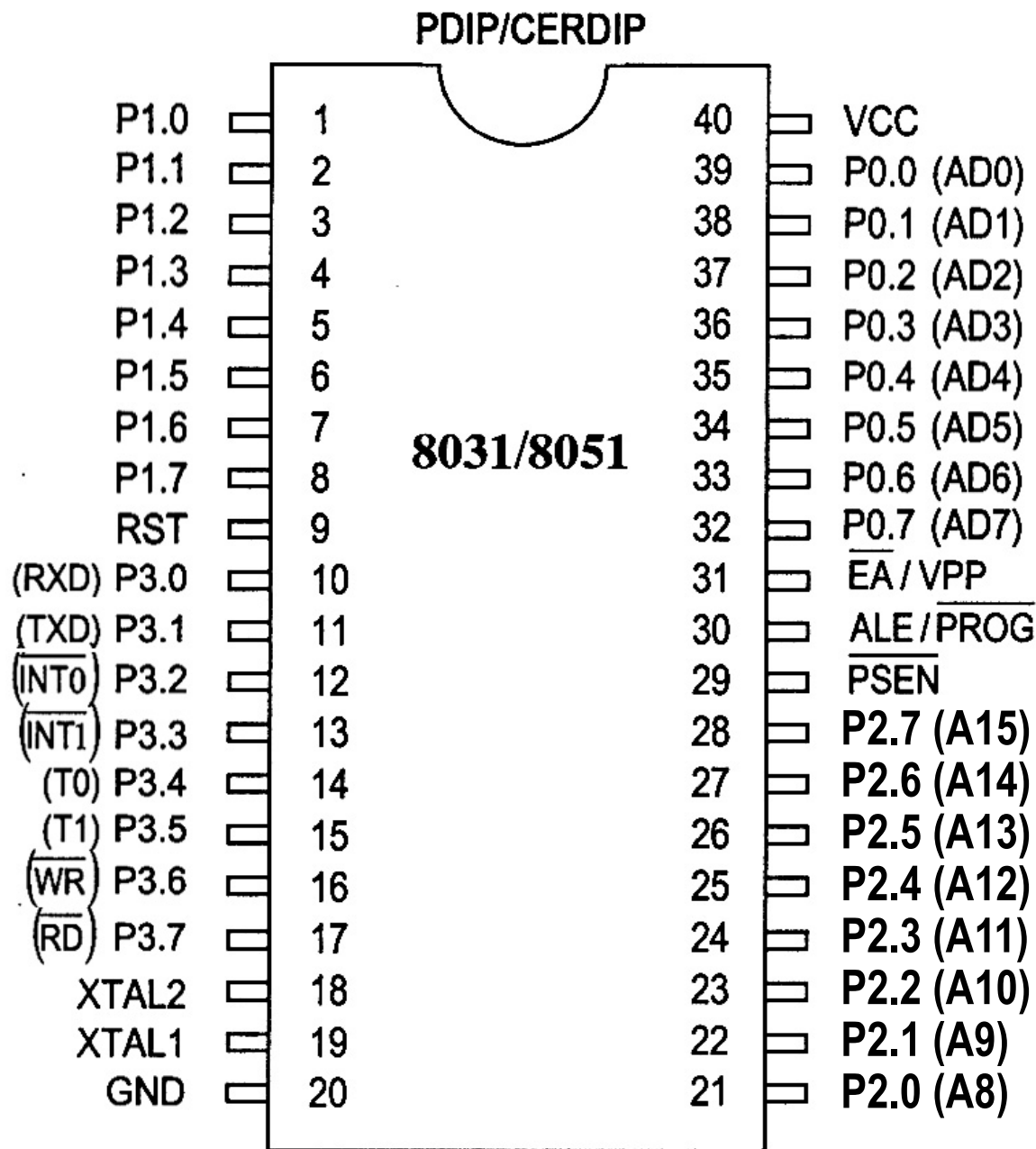
- **Khối CPU**
- **Khối tạo dao động**
- **Bộ nhớ ROM, RAM**
- **Khối điều khiển ngắt**
- **Khối điều khiển bus**
- **Khối Timer/Counter**
- **Khối cổng vào ra**



Hình 1.2. Sơ đồ khối bộ vi điều khiển 8051

1.2 Bố trí chân

- Các thành viên họ 8051 (8751, 89C51, DS5000...) đều có 40 chân.
- Có các kiểu đóng vỏ:
 - + **DIP** (Dual In Line Package) Hai hàng chân.
 - + **QFP** (Quad Flat Package) Vỏ dẹt vuông
 - + **LCC** (Leadless Chip Carrier) Không có chân đỡ
- Thường dùng dạng 40 chân kiểu DIP.



Bố trí chân của VĐK 8051

Bố trí 40 chân

40 chân được phân thành các nhóm:

- a) Nhóm chân nguồn nuôi**
- b) Nhóm chân điều khiển địa_chỉ/Dữ_liệu**
- c) Nhóm chân tạo dao động**
- d) Nhóm chân điều khiển ngắt**
- e) Nhóm chân cổng**

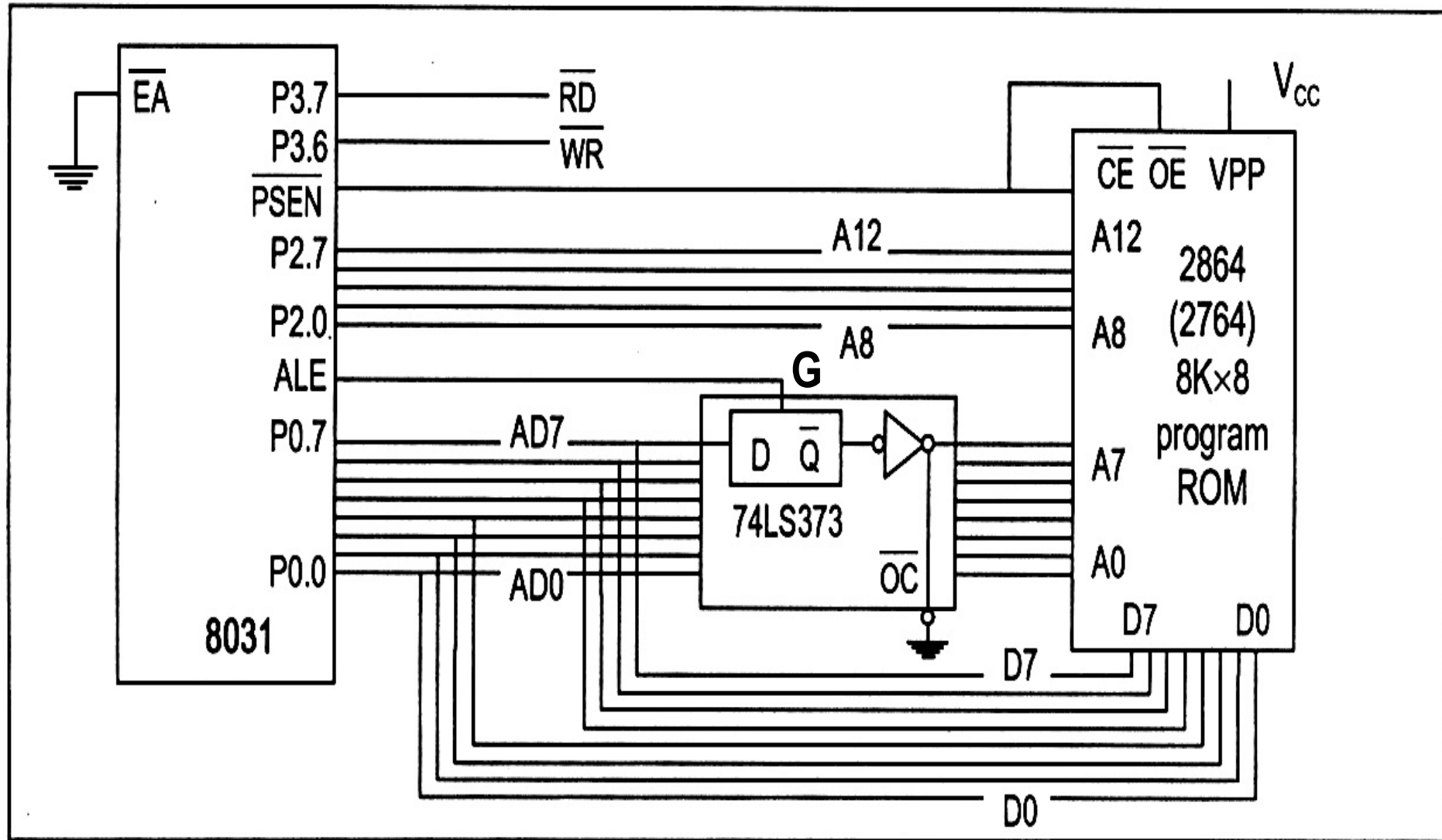
a) Nhóm chân nguồn nuôi

- **VCC (40): Cấp nguồn +5V.**
- **GND (20): Chân đất.**

- a) Nhóm chân nguồn nuôi**
- b) Nhóm chân điều khiển Địa_chỉ/Dữ_liệu**
- c) Nhóm chân tạo dao động**
- d) Nhóm chân điều khiển ngắt**
- e) Nhóm chân cổng**

b) Nhóm chân điều khiển địa chỉ/dữ liệu.

Có 3 chân: \overline{EA} , \overline{PSEN} , ALE



Nối ghép 8031/51 với bộ nhớ ROM ngoài

3 chân: \overline{EA} , \overline{PSEN} , ALE

▪ **\overline{EA}/V_{pp}** : Truy cập bộ nhớ ngoài (External Access) là chân vào.

- Để truy cập bộ nhớ ngoài, chân này được nối đất (với VĐK 8031, 8032).
- Với các VĐK có ROM trên chip (8751, 98C51 hoặc DS5000) chân này được nối nguồn V_{cc} .

PSEN: (Program Store Enable)- **Cho phép cất chương trình**

- **Chân ra, mức tích cực thấp.**
- **Được nối tới chân OE của ROM cho phép cất/tải chương trình tới bộ nhớ ROM ngoài.**
- **PSEN Làm việc kết hợp với chân EA, ALE.**

ALE: Cho phép chốt địa chỉ (Address Latch Enable).

- Chân ra, tích cực cao.
- Khi nối 8031 với bộ nhớ ngoài thì cổng P0 có thể cấp địa chỉ và dữ liệu.
- Chân ALE được dùng để phân kênh địa chỉ/dữ liệu bằng cách nối tới chân G của chip 74LS373.
 - Khi $ALE = 0$ thì P0 cấp dữ liệu.
 - Khi $ALE = 1$ thì P0 cấp địa chỉ

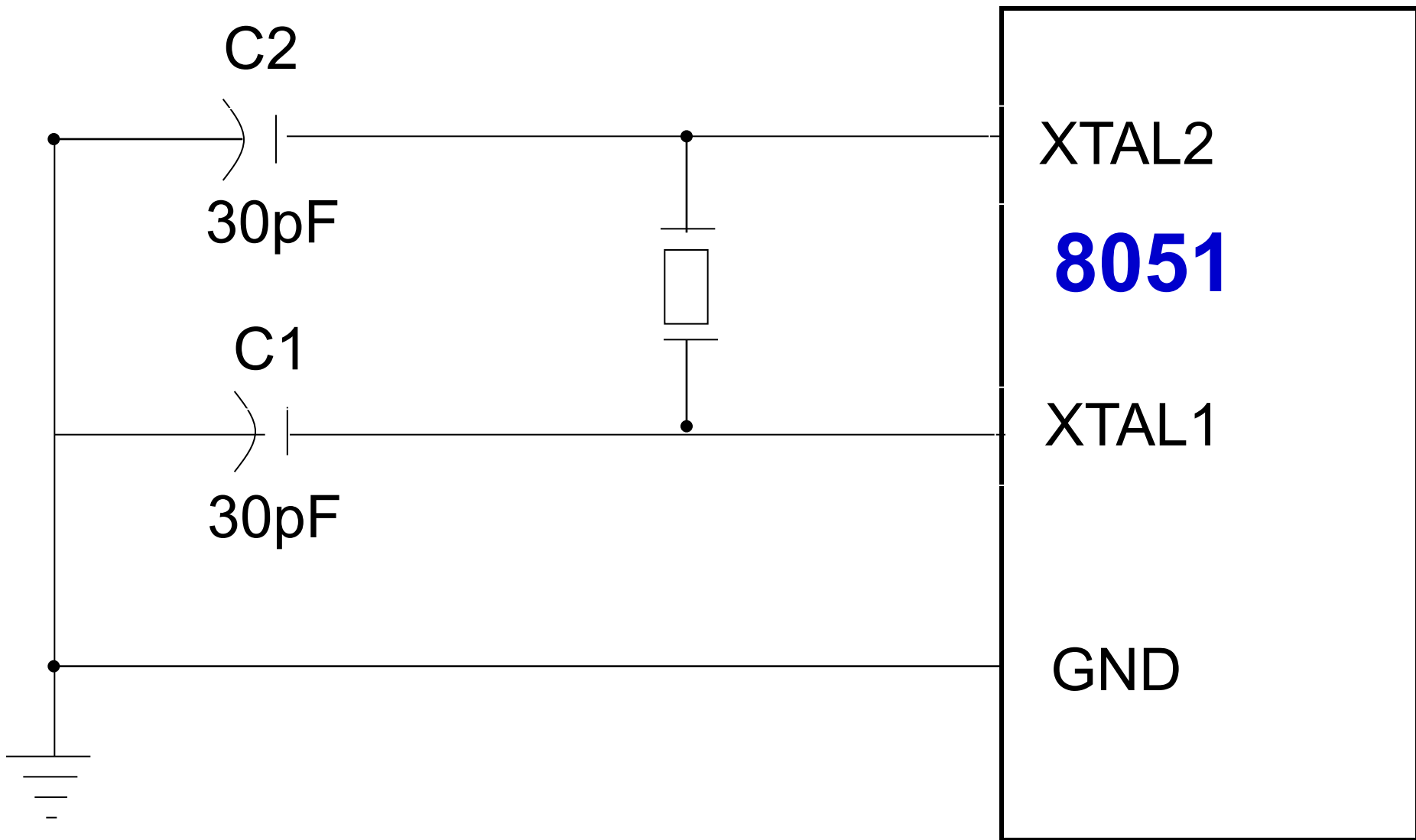
c) Nhóm chân tạo dao động

- **XTAL1 (19) và XTAL2 (18):**

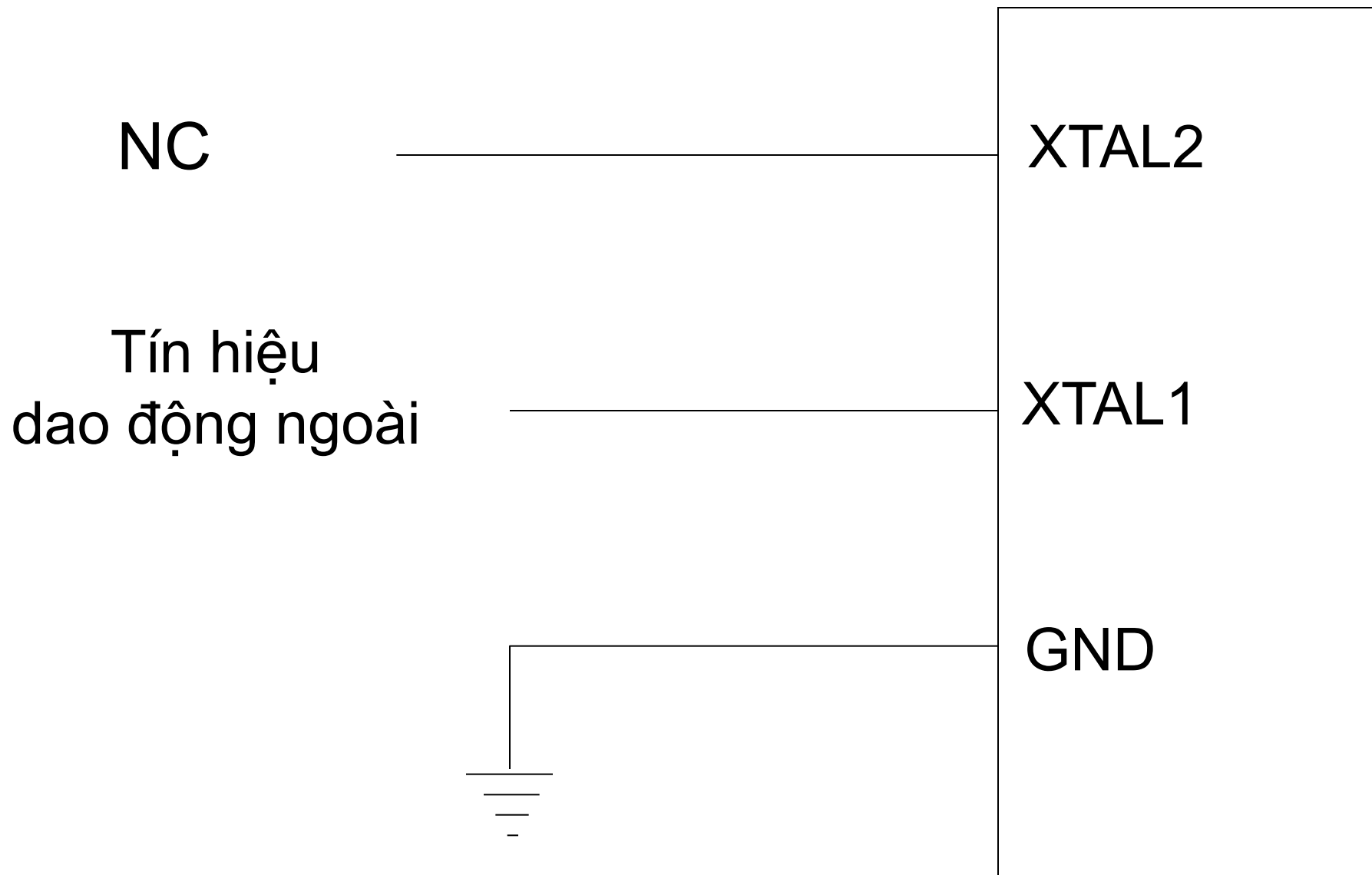
Nối dao động.

+ Khi mắc dao động thạch anh, cần có 2 tụ điện 30pF.

+ Khi dùng dao động ngoài - nối tới chân XTAL1, còn chân XTAL2 để[?] hở.



Hình 4.2. a) Nối thạch anh



Hình 4.2. b) Nối đồng hồ ngoài

Chu kỳ máy:

**Chu kỳ máy của hệ 8051 bằng 12
lần chu kỳ đồng hồ**

Ví dụ 1:

Cho các tần số đồng hồ:

a) $F_{xtal}=11,0592\text{MHz}$

b) $F_{xtal}=16\text{MHz}$.

Xác định chu kỳ máy hệ 8051.

Giải:

a) $T_{xtal} = 1/11,0592\text{MHz} = 0,0904\mu\text{s}.$

Chu kỳ máy =

$T_{mc} = 12T_{xtal} = 12 \times 0,0904 = 1,085\mu\text{s}.$

b. $T_{xtal} = 1/16\text{MHz} = 0,0625\mu\text{s}$

$T_{mc} = 12T_{xtal} = 12 \times 0,0625 = 0,75\mu\text{s}.$

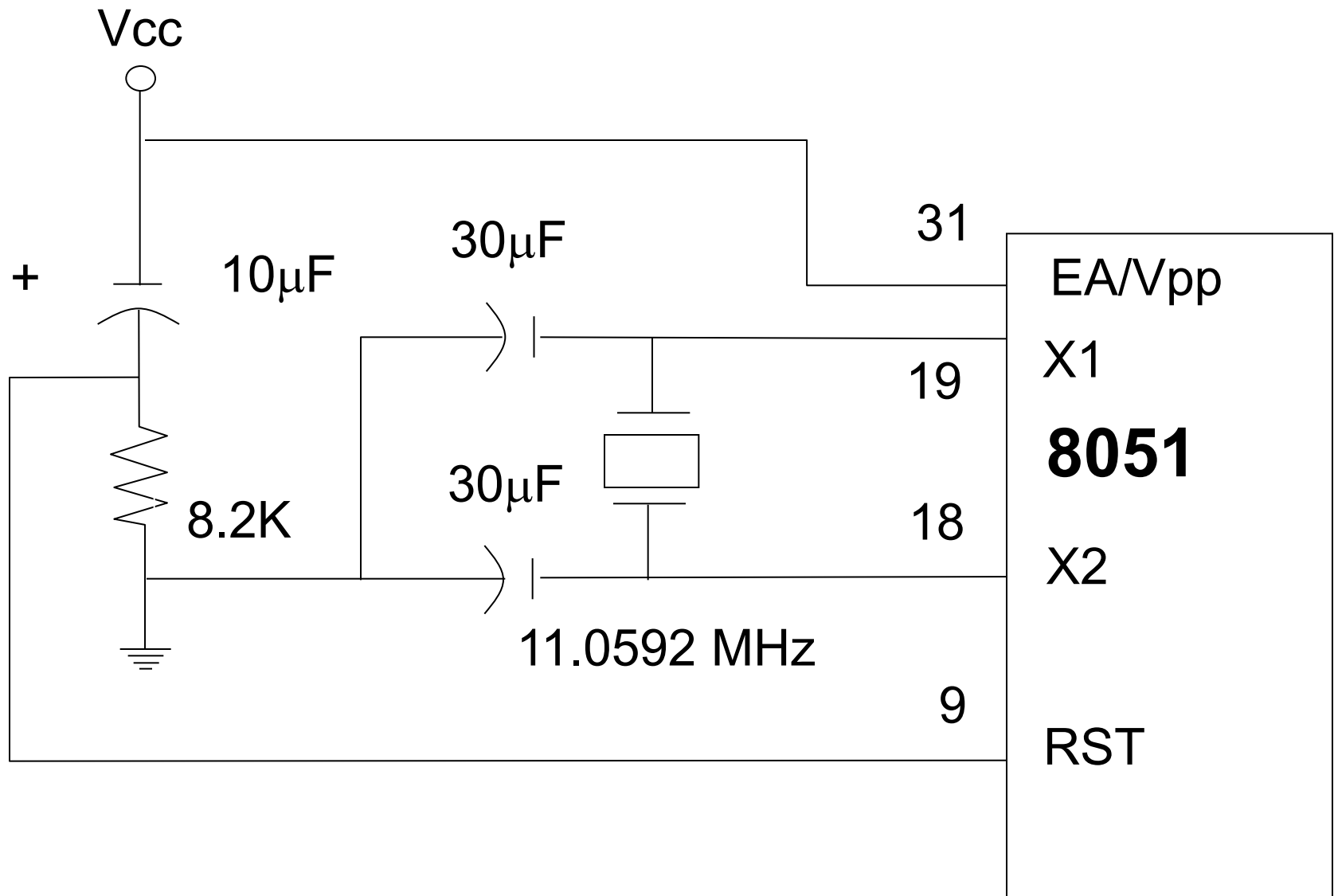
d) Nhóm chân điều khiển ngắt

8051 có 6 ngắt gồm:

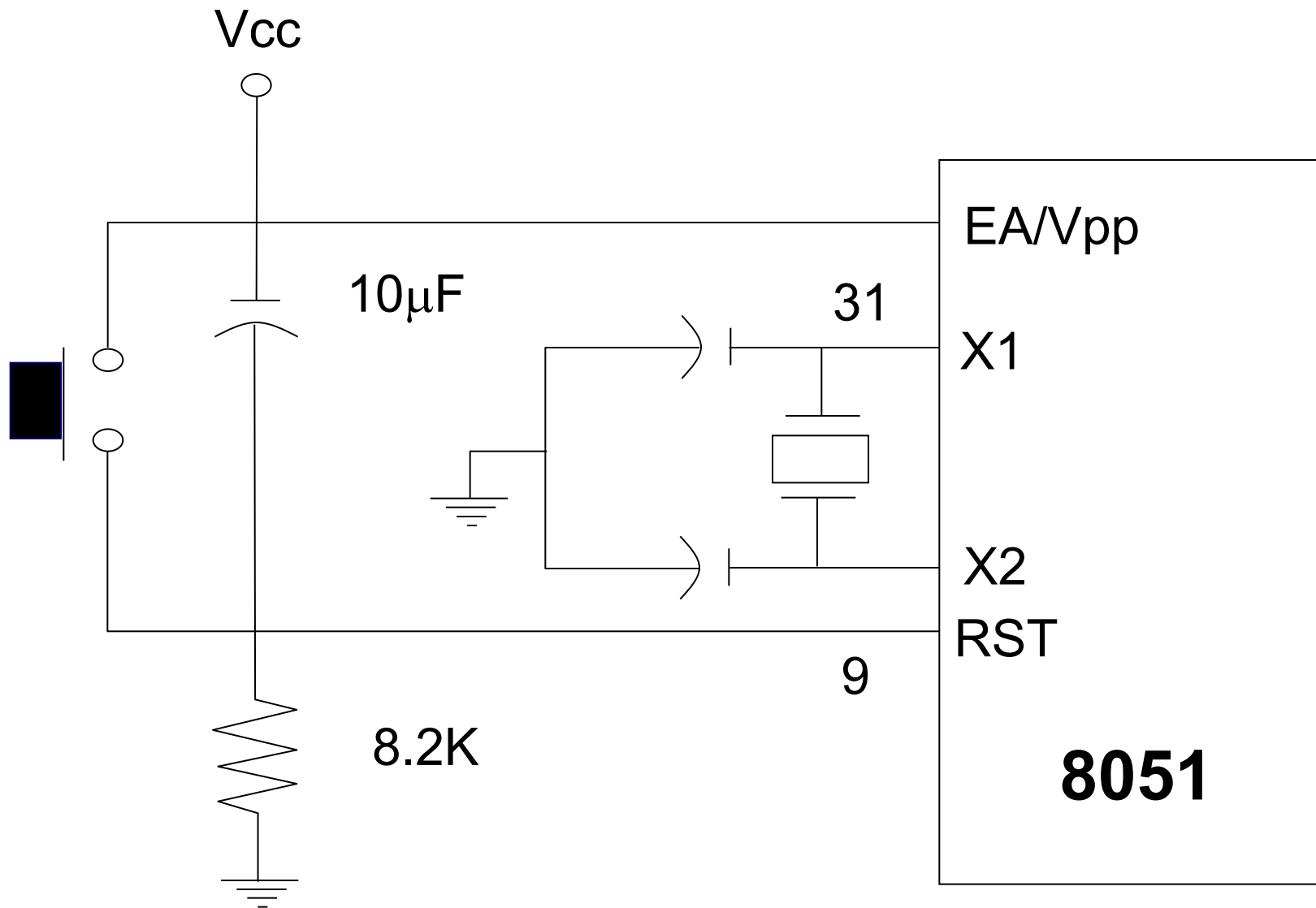
- **Chân RESET**
- **Ngắt ngoài INT0**
- **Ngắt ngoài INT1**
- **Ngắt TIMER0**
- **Ngắt TIMER1**
- **Ngắt cổng COM**

RST: Khởi động lại (RESET) chân 9.

- Bình thường ở mức thấp.
- Tích cực mức cao.
- Khi có xung cao kích hoạt, bộ vi điều khiển kết thúc mọi hoạt động hiện tại và tiến hành khởi động lại.



Hình 4.3. a) Mạch nối chân RST



Hình 4.3. Mạch Reset với phím khởi động

Bảng 4.1. Giá trị một số thanh ghi sau RESET

Register	Reset Value
PC	0000
ACC	0000
B	0000
PSW	0000
SP	0000
DPTR	0007

Lệnh	Tên	Đchỉ
ACC*	Thanh chứa A	0E0H
B*	Thanh ghi B	0F0H
PSW*	Từ trạng thái chương trình	0D0H
SP	Con trỏ ngăn xếp	81H
DPTR	Con trỏ dữ liệu hai byte	
DPL	Byte thấp của DPTR	82H
DPH	Byte cao của DPTR	83H
P0*	Cổng 0	80H
P1*	Cổng 1	90H

2. Tổ chức các cổng 8051

Có tất cả 04 cổng:

2.1 Cổng P0

2.2 Cổng P1

2.3 Cổng P2

2.4 Cổng P3

Chung về 4 cổng của 8051

- 4 cổng: P0, P1, P2, P3. Mỗi cổng có 8 chân, tất cả 32 chân.
- Các cổng ngầm định được cấu hình làm cổng ra.
- Để làm cổng vào cần lập trình.

2.1 Cổng P0

P0 thực hiện được 3 chức năng:

a) P0 làm cổng ra

b) P0 làm cổng vào

c) P0 dùng chuyển địa chỉ

a) P0 làm cổng ra

- P0 là cổng 8 bit, dồn kênh AD AD0-AD7.
- Ngầm định P0 làm cổng ra.
- Để làm cổng vào cần nối thêm điện trở kéo 10 k Ω (vì cổng thiết kế dạng cực máng hở) và cần lập trình.

Ví dụ 2:

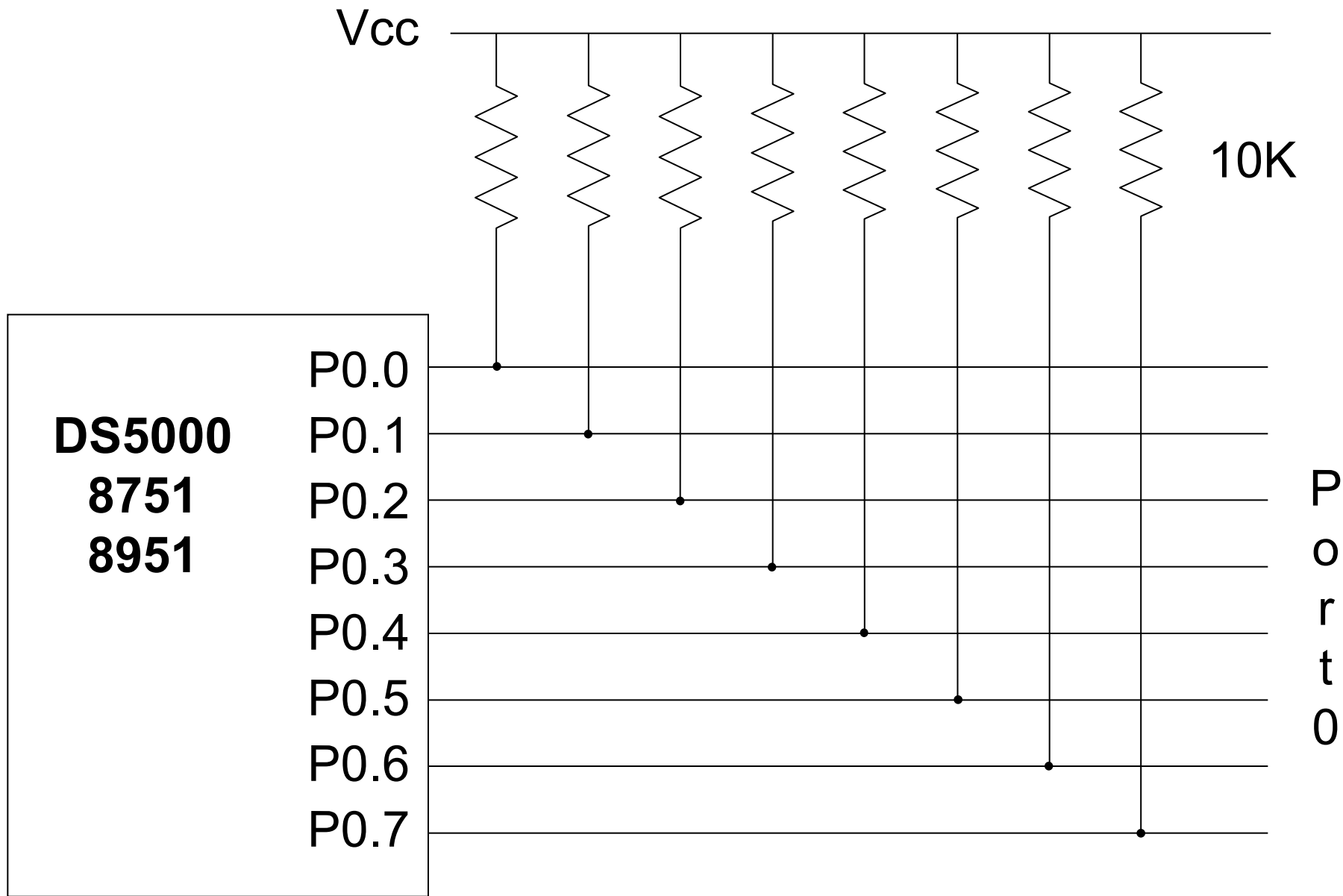
**Viết đoạn chương trình tạo dãy
xung vuông liên tục trên các chân
cổng P0.**

Giải:

```
                MOV        A, #55H  
BACK:          MOV        P0, A  
                ACALL      DELAY  
                CPL        A  
                SJMP       BACK
```

b) P0 làm cổng vào

- Sau khi có các điện trở nối tới cổng P0, để làm cổng vào cần lập trình bằng cách ghi 1 tới tất cả các bit của cổng.



Hình 4.4. Mắc điện trở kéo cổng P0

P0 làm công vào

Ví dụ 3:

Viết đoạn chương trình liên tục nhận dữ liệu tại P0 và gửi ra P1.

Giải:

MOV A , #0FFH

MOV P0 , A

BACK : MOV A , P0

MOV P1 , A

SJMP BACK

c) Cổng P0 dùng chuyển địa chỉ

P0 là cổng dồn kênh nên có thể dùng để chuyển dữ liệu hoặc địa chỉ

- Nếu $ALE=1$ thì P0 cấp địa chỉ A0-A7.
- Nếu $ALE = 0$ thì P0 cấp dữ liệu D0-D7.

2.2 Cổng P1

- a) P1 làm cổng ra
- b) P1 làm cổng vào

a) P1 làm công ra

Ví dụ 4:

Viết đoạn chương trình tạo dãy xung vuông liên tục trên các chân cổng P1.

a) P1 làm cổng ra

- **P1 có sẵn các điện trở kéo bên trong.**
- **Khi Reset, P1 được cấu hình làm cổng ra.**
- **P1 làm cổng vào cần lập trình.**

Ví dụ 4: P1 làm công ra

**Viết đoạn chương trình tạo dãy
xung vuông liên tục trên các chân
công P1.**

Giải:

```
LAP:  MOV    A, #55H
      MOV    P1, A
      ACALL  DELAY
      MOV    A, #0AAH
      MOV    P1, A
      ACALL  DELAY
      SJMP   LAP
```

b) P1 làm cổng vào

- Để P1 làm cổng vào cần lập trình bằng cách ghi 1 đến tất cả các bit của cổng.

Ví dụ 5:

Viết đoạn chương trình:

- a) Cấu hình P1 làm cổng vào.**
- b) Liên tục nhận dữ liệu tại P1.**
- c) Chuyển dữ liệu nhận được vào các thanh ghi R6 và R5.**

Giải

MOV A,#0FFH

MOV P1,A

MOV A,P1

MOV R6,A

ACALL DELAY

MOV A,P1

MOV R5,A

2.3 Cổng P2

- a) P2 làm cổng ra
- b) P2 làm cổng vào
- c) P2 dùng chuyển địa chỉ

a) P2 làm cổng ra

- Cổng ra là chế độ ngầm định.
- Để làm cổng vào cần lập trình

Ví dụ 6: P2 làm công ra

**Viết đoạn chương trình tạo dãy
xung vuông liên tục tại các chân
công P2.**

Giải:

```
BACK: MOV    P2,#55H  
      ACALL  DELAY  
      MOV    P2,#0AAH  
      ACALL  DELAY  
      SJMP   BACK
```


b) P2 làm cổng vào

- Khác với P0, để P2 làm cổng vào không cần mắc điện trở kéo
- Cần lập trình bằng cách ghi 1 tới tất cả các chân cổng.

Ví dụ 7:

Viết đoạn chương trình, đầu tiên cấu hình P2 làm cổng vào, sau đó dữ liệu nhận được từ P2 được gửi liên tục đến P1.

Chương trình

```
        MOV        A,#0FFH
        MOV        P2,A
BACK:   MOV        A,P2
        MOV        P1,A
        SJMP       BACK
```

c) P2 chuyển địa chỉ

- **P2 được dùng với P0 để tạo ra địa chỉ 16 bit.**
- **P0 cấp 8 bit địa chỉ thấp A0-A7**
- **P2 cấp 8 bit địa chỉ cao A8-A15**
- **8031 có bộ nhớ ngoài 64KB nên cần bus địa chỉ 16 bit.**
- **Khi cấp địa chỉ thì P2 không dùng cho I/O.**

2.4 Cổng P3

a) P3 làm cổng ra

b) P3 làm cổng vào.

c) P3 làm chức năng đặc biệt.

- P3 có thể dùng làm cổng vào hoặc ra tương tự các cổng khác.
- Ứng dụng chính cổng P3 là cung cấp tín hiệu đặc biệt, như ngắt, truyền thông nối tiếp, điều khiển đọc, ghi...

Chức năng quan trọng của P3

Bit P3	Chức năng	Chân
P3.0	Nhận dữ liệu (RXD)	10
P3.1	Phát dữ liệu (TXD)	11
P3.2	Ngắt 0 (INT0)	12
P3.3	Ngắt 1 (INT1)	13
P3.4	Bộ định thời 0 (T0)	14
P3.5	Bộ định thời 1(T1)	15
P3.6	Ghi (WR)	16
P3.7	Đọc (RD)	17

- P3.0, P3.1 dùng cho thu/phát dữ liệu trong truyền tin nối tiếp.
- P3.2, P3.3 dành cho ngắt ngoài INT0, INT1.
- P3.4, P3.5 dùng làm bộ định thời Timer0 và Timer1.
- P3.6, P3.7 dùng ghi, đọc bộ nhớ ngoài - RD, WR.
- Trong các hệ 8751, 89C51 hoặc DS5000, P3.6, P3.7 dùng làm vào/ra.

3. Lập trình vào/ra qua cổng²

a) Phương pháp truy cập byte

b) Phương pháp truy cập bit

a) Phương pháp truy cập byte

- **Dùng thanh ghi**
- **Dùng giá trị trực tiếp**
- **Dùng kỹ thuật Đọc-Sửa-Ghi**

Phương pháp dùng Thanh ghi:

Sử dụng thanh ghi truy cập 8 bit công.

Ví dụ 8:

Lập trình dùng thanh ghi A tạo xung vuông tại cổng P1

```
BACK:MOV    A,#55H  
      MOV    P1,A  
      ACALL  DELAY  
      MOV    A,#0AAH  
      MOV    P1,A  
      ACALL  DELAY  
      SJMP   BACK
```

Phương pháp dùng giá trị trực tiếp

Dữ liệu là giá trị hằng số 8 bit sử dụng trong lệnh truy cập cổng.

Ví dụ 9:

Viết lại đoạn chương trình trên dưới dạng dùng giá trị trực tiếp.

BACK:

```
MOV    P1,#55H  
ACALL  DELAY  
MOV    P1,#AAH  
ACALL  DELAY  
SJMP   BACK
```

Phương pháp kết hợp Đọc-Sửa-Ghi

- Là phương pháp rút gọn nhờ kết hợp đồng thời cả ba thao tác: Đọc-Sửa-Ghi trong 1 lệnh.

Ví dụ 10:

Viết lại đoạn chương trình trên dưới dạng dùng kỹ thuật kết hợp Đọc_Sửa_Ghi.

Ví dụ:

MOV P1,#55H

LAP: XOR P1,#0FFH

ACALL DELAY

SJMP LAP

3. Lập trình vào/ra qua cổng

a) Phương pháp truy cập byte

b) Phương pháp truy cập bit

b) Phương pháp truy cập bit

- 8051 có khả năng truy cập từng bit công và không làm thay đổi các bit còn lại.

- Truy cập bit dùng lệnh CPL
- Truy cập bit dùng lệnh SETB, CLR

Truy cập bit dùng lệnh CPL

Ví dụ 11:

**Viết đoạn chương trình tạo dãy
xung vuông tại chân P1.2 dùng lệnh
đảo bit**

BACK: CPL P1.2

ACALL DELAY

SJMP BACK

Truy cập bit dùng lệnh SET CLR bit

Ví dụ 12:

Viết đoạn chương trình tạo dãy xung vuông tại chân P1.2 dùng lệnh SETB và CLR.

AGAIN:

SETB P1.2

ACALL DELAY

CLR P1.2

ACALL DELAY

SJMP AGAIN

Bảng 4.3. Địa chỉ bit của các cổng

P0	P1	P2	P3	Bit
P0.0	P1.0	P2.0	P3.0	D0
P0.1	P1.1	P2.1	P3.1	D1
P0.2	P1.2	P2.2	P3.2	D2
P0.3	P1.3	P2.3	P3.3	D3
P0.4	P1.4	P2.4	P3.4	D4
P0.5	P1.5	P2.5	P3.5	D5
P0.6	P1.6	P2.6	P3.6	D6
P0.7	P1.7	P2.7	P3.7	D7

Ví dụ 14:

Viết chương trình thực hiện các công việc sau:

- a) Duy trì kiểm tra bit P1.2 đến khi lên cao.**
- b) Khi P1.2 lên cao, ghi giá trị 45H vào P0.**
- c) Tạo sườn xung âm (xung Cao-Xuống-Thấp) ở chân P2.3.**

Giải:

SETB P1.2

MOV A,#45H

AGAIN:

JNB P1.2,AGAIN

MOV P0,A

SETB P2.3

CLR P2.3

Bộ đếm, bộ định thời và lập trình

NỘI DUNG

1. Bộ định thời và lập trình

2. Bộ đếm và lập trình

1. Bộ định thời và lập trình

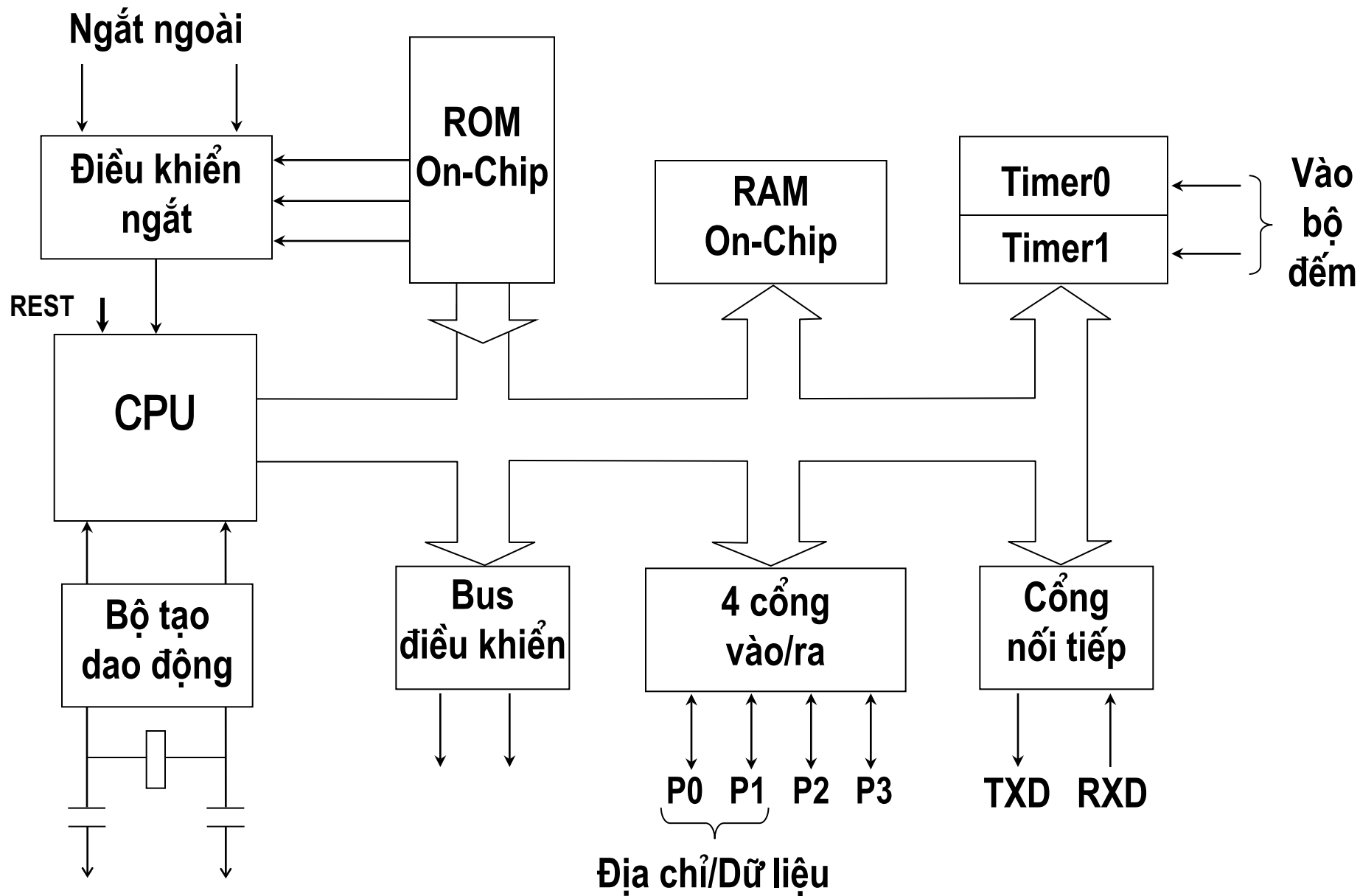
a) Bộ định thời

b) Lập trình chế độ 1

c) Lập trình chế độ 2

a) Bộ định thời

- Bộ định thời
- Thanh đếm
- Thanh ghi chế độ
TMOD



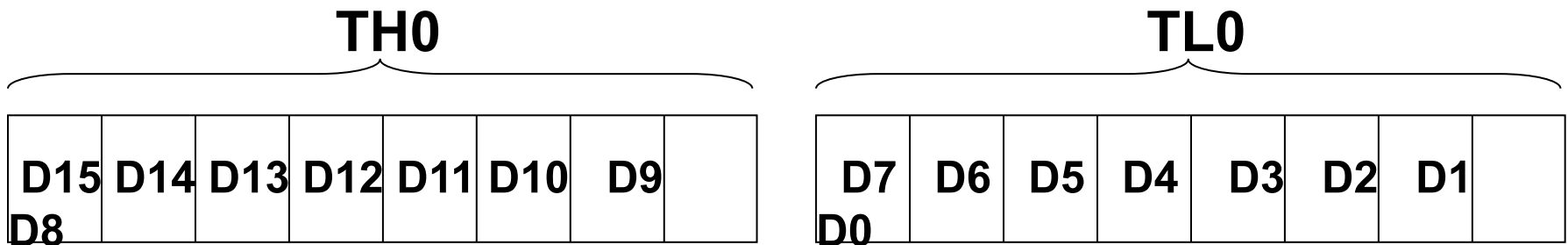
Hình 1.2. Sơ đồ khối bộ vi điều khiển 8051

Bộ định thời

- 8051 có 2 bộ định thời là **Timer0** và **Timer1**.
- Mỗi bộ định thời có 1 thanh ghi 16 bit làm **thanh đếm**.
- Mỗi thanh đếm có hai nửa độc lập là byte thấp (L) và byte cao (H).

Thanh đếm của Timer0

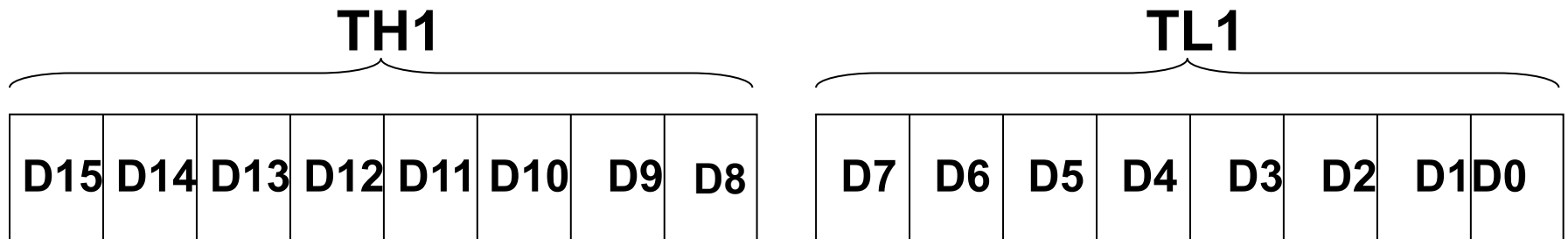
- Là thanh ghi 16 bit gồm byte thấp TL0 và byte cao TH0.
- Có thể truy cập thanh ghi này như đối với các thanh ghi A, B, R0, R1, R2 ...
- Ví dụ: **MOV TL0, #4FH**
MOV R4, TH0



Thanh ghi bộ định thời Timer0

Thanh đếm của Timer1

- Thanh ghi Timer1 16 bit và có byte thấp TL1 và byte cao TH1.
- Truy cập thanh ghi này giống như của Timer0.



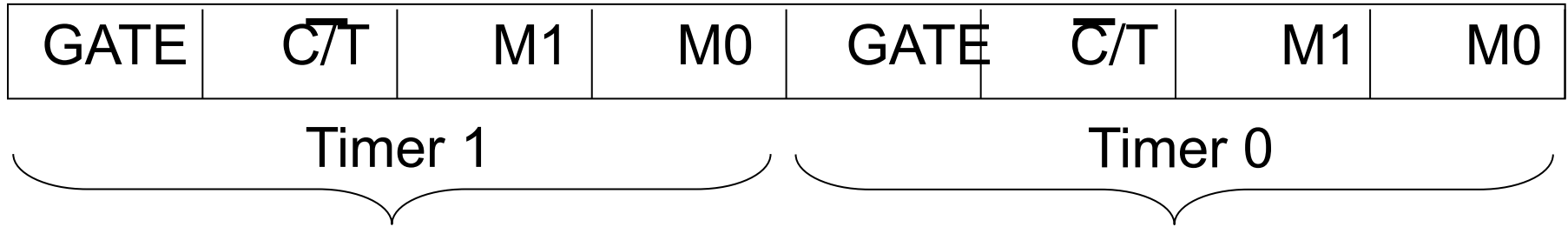
Thanh ghi bộ định thời Timer1

Thanh ghi chế độ TMOD của bộ định thời

- Thanh ghi TMOD 8 bit dùng để thiết lập các chế độ làm việc cho cả Timer0 và Timer1.
- 4 bit thấp TMOD dành cho Timer0
- 4 bit cao dành cho Timer1.
- Trong 4 bit đó:
 - 2 bit thấp để thiết lập chế độ
 - 2 bit cao để thiết lập kiểu hoạt động

(MSB)

(LSB)



- **GATE:** Nếu $GATE=1$, $TRx=1$ thì Timer/Counter được mở khi $INTx=1$ (kích hoạt cứng)
 Nếu $GATE=0$, T/C mở khi $TRx=1$ (kích mềm)
- $\overline{C}/T=0$ là Timer, kích bằng đồng hồ thạch anh;
 $=1$ là Counter, kích bằng xung vào Tx

M1 Chế độ bit 1

M0 Chế độ bit 0

Hình A.8. Thanh ghi TMOD

Bit chọn chế độ M1, M0

M1	M0	Mode	Chế độ hoạt động
0	0	0	Bộ định thời 13 bit
0	1	1	Bộ định thời 16 bit
1	0	2	Chế độ 8 bit tự nạp lại THx lưu giá trị sẽ tự nạp vào TLx mỗi khi tràn
1	1	3	Chế độ bộ định thời chia tách

Bit C/\bar{T} (Bộ đếm/Bộ định thời)

- Là bit chọn chế độ tạo trễ hay bộ đếm.
- Nếu $C/\bar{T}=0$ thì bộ định thời làm bộ tạo trễ. Nguồn đồng hồ là tần số thạch anh.
- Nếu $C/\bar{T}=1$ thì bộ định thời làm bộ đếm, tức đếm xung T0 (đưa đến chân P3.4) hoặc T1 (chân P3.5).

GATE

- Để chọn cách khởi động/dừng bộ định thời bằng phần mềm hoặc phần cứng.
- Gate=0 khởi động/dừng bộ định thời bằng phần mềm
 - Lệnh SETB TRx để khởi động
 - CLR TRx để dừng
- GATE=1 khởi động/dừng bộ định thời bằng phần cứng

Các chế độ hoạt động

- M0, M1 dùng để chọn chế độ 0, 1, và 2 của Timer0, Timer1.
- Chế độ 0: Bộ định thời 13 bit.
- Chế độ 1: Bộ định thời 16 bit
- Chế độ 2: Bộ định thời 8 bit.
- Chế độ được sử dụng rộng rãi là chế độ 1 và 2.

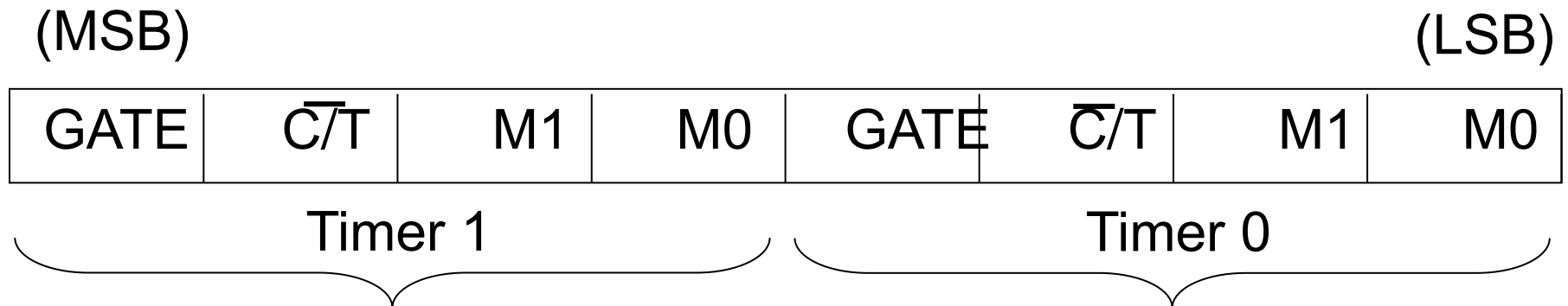
Ví dụ 1:

Xác định chế độ và bộ định thời của các trường hợp sau:

a) **MOV TMOD,#01H**

b) **MOV TMOD,#20H**

c) **MOV TMOD,#12H**



Giải:

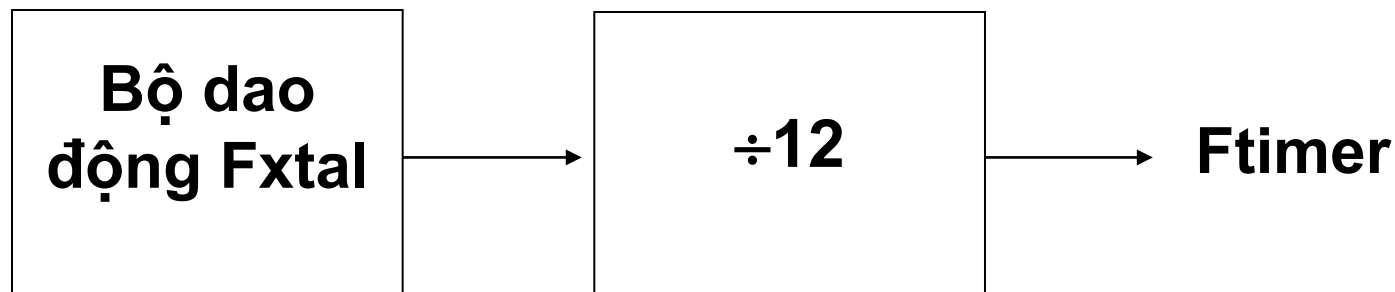
Chuyển số hexa sang nhị phân và đối chiếu với hình 9.3 ta có:

a) TMOD=00000001 ;Timer0, Mode1.
b) TMOD=00100000 ;Timer1, Mode2.
c) TMOD=00010010 ;Timer0, Mode2
;và Timer1,Mode1

Tần số đồng hồ bộ định thời

- Nếu $C/\overline{T} = 0$ thì thạch anh trên chip là nguồn tần số của đồng hồ bộ định thời.
- Chu kỳ bộ định thời luôn bằng 12 lần chu kỳ thạch anh.

$$T_{\text{timer}} = 12T_{\text{xtal}}$$



Ví dụ 2:

Cho thạch anh có tần số nh sau:

a) 12MHz

b) 16MHz

c) 11,0592MHz

Xác định tần số và chu kỳ bộ định thời.

Giải:

a) $F=12 \text{ MHz} \Rightarrow T=1/f \Rightarrow T_{\text{timer}}=12/f=1 \text{ } \mu\text{s}$

b) $F=16 \text{ MHz} \Rightarrow T=1/16=0,0625 \text{ } \mu\text{s}$

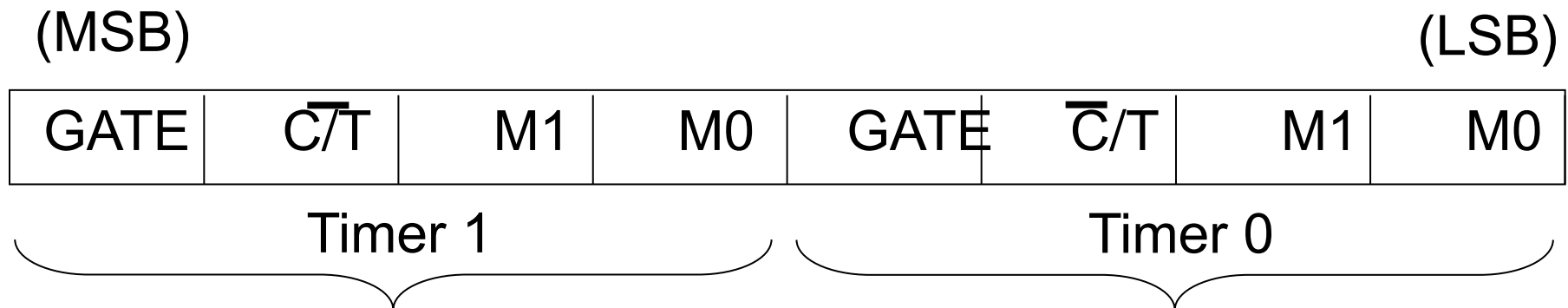
và $T_{\text{timer}}=12T=0,75 \text{ } \mu\text{s}$

c) $F= 11,0592 \text{ MHz} \Rightarrow T=1/f=0,0904 \text{ } \mu\text{s}$

$T_{\text{timer}}=12T = 1,085 \text{ } \mu\text{s}$

Ví dụ 3:

Xác định TMOD để lập trình Timer0, Mode2, nguồn đồng hồ là thạch anh, khởi động/dừng bộ định thời bằng phần mềm.



Giải:

- **TMOD = 0000 0010=02H** Timer0, Mode2
- **C/T = 0** đồng hồ từ nguồn XTAL
- **GATE=0** dùng phần mềm khởi động và dừng Timer0

b) Lập trình Mode1

Một số chú ý:

- 1. Mode1 dùng thanh đếm 16 bit, nên giá trị nạp TH-TL từ 0-FFFFH .**
- 2. Sau khi nạp TH-TL thì khởi động bộ định thời bằng lệnh:**
 - SETB TR0 cho Timer0**
 - SETB TR1 cho Timer1**

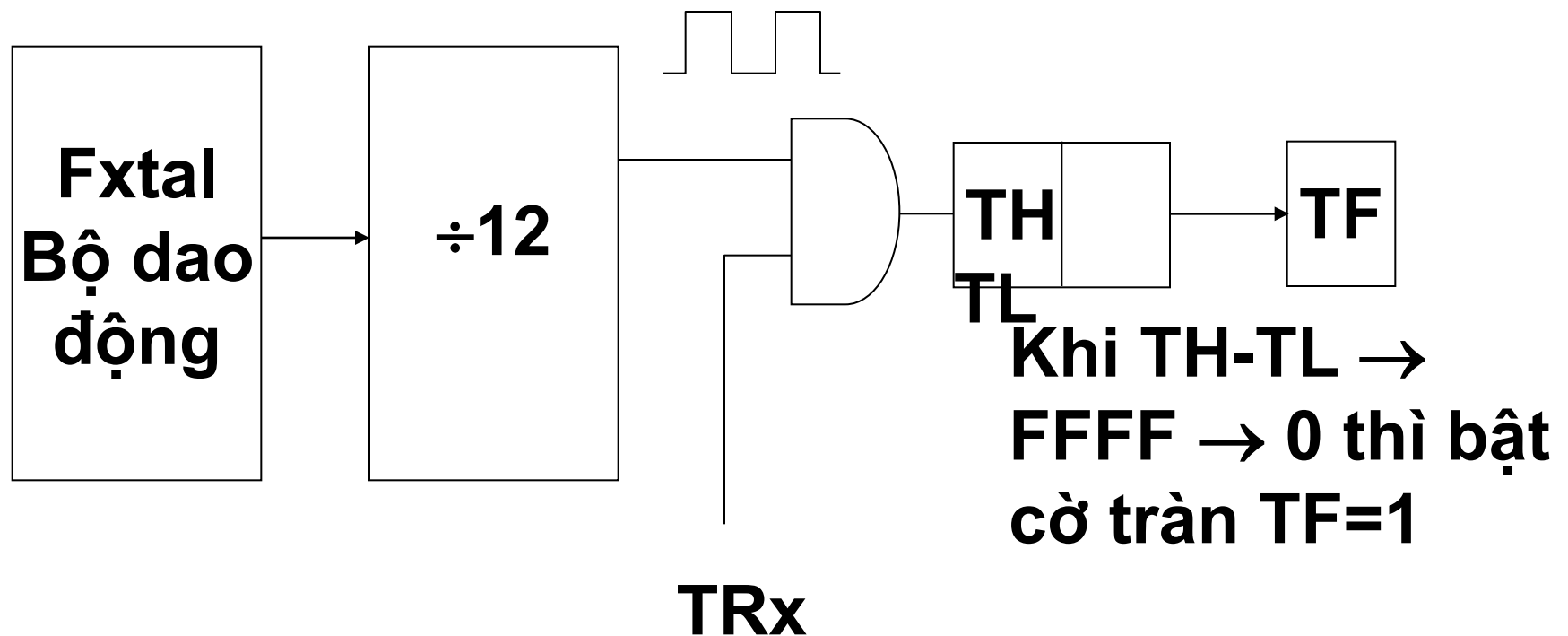
3. Sau đó bộ định thời bắt đầu đếm tăng cho đến khi đạt max FFFFH.

4. Khi đạt max thì bộ đếm quay về 0 và lập cờ TF=1.

5. Khi cờ TF=1, để dừng bộ định thời dùng lệnh

- **CLR TR0 cho Timer0**
- **CLR TR1 cho Timer1**

6. Để lặp lại quá trình đếm, các thanh ghi TH và TL phải nạp lại giá trị ban đầu và cờ TF cần xoá về 0.



Bật cờ TF bộ định thời

Các bước lập trình Timer chế độ 1

Chọn Timer0 (Timer1) để làm việc

1. Nạp TMOD chọn chế độ làm việc

2. Nạp TLx và THx

3. Khởi động Timer bằng lệnh SETB TRx

4. Kiểm tra cờ tràn TFx bằng lệnh "JNB TFx, đích". Thoát khi TFx=1

5. Dừng Timer bằng lệnh CLR TRx

6. Xoá cờ TF bằng lệnh CLR TFx cho vòng lặp tiếp theo.

7. Lặp lại bước 2 để nạp lại TL và TH.

Ví dụ 4:

Viết đoạn chương trình dùng Timer0 tạo xung vuông trên chân P1.5 có độ đầy xung 50%. Cho TH0=FFH, TL0=F2H.

Các bước lập trình Timer chế độ 1

Chọn Timer0 (Timer1) để làm việc

1. Nạp TMOD chọn chế độ làm việc

2. Nạp TLx và THx

3. Khởi động Timer bằng lệnh SETB TRx

4. Kiểm tra cờ tràn TFx bằng lệnh "JNB TFx, đích". Thoát khi TFx=1

5. Dừng Timer (CLR TRx)

6. Xoá cờ TF (CLR TFx) và đảo bit để tạo xung.

7. Lặp lại 2, nạp lại TL và TH.

Giải:

```
MOV      TMOD,#01    ;Chọn Timer0,  
Mode1  
HERE:  
MOV      TL0,#0F2H  
MOV      TH0,#0FFH  
CPL  P1.5            ;Sử dụng chân P1.5  
ACALL DELAY  
SJMP  HERE           ;Nạp lại TH,TL
```

;--- tạo trễ dùng Timer0

DELAY:

SETB

TR0

;Khởi

động

Timer0

AGAIN:

JNB TF0,AGAIN ;Lặp đến khi TF0=1

CLR TR0 ;Dừng Timer0

CLR TF0 ;Xoá cờ bộ Timer0

RET

Phân tích chương trình:

Các bước thực hiện chương trình trên:

1. Nạp TMOD.

2. Nạp giá trị FFF2H vào TH0 - TL0.

3. SETB TR để bắt đầu đếm

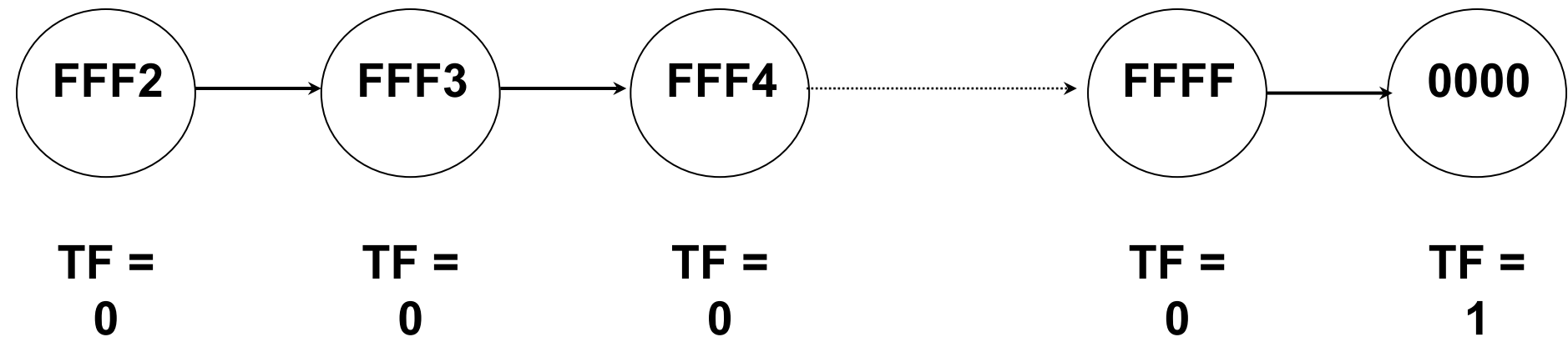
4. Timer0 đếm tăng theo xung đồng hồ thạch anh, đếm qua các trạng thái FFF3, FFF4 ...đến FFFFH rồi thêm 1 xung nữa quay về 0 và bật cờ TF0 = 1.

- 6. Tại thời điểm này lệnh JNB sẽ thoát khỏi vòng lặp và rẽ nhánh xuống thực hiện lệnh sau JNB.**
- 7. Đảo P1.5 để tạo xung.**
- 8. Lệnh “CLR TR0” dừng bộ Timer0.**

Ví dụ 5:

- Tính chu kỳ xung tạo ra do chương trình ở ví dụ trước, cho $F_{xtal} = 11,0592 \text{ MHz}$.

Với TH_TL=FFF2H số đếm bằng 14.



Giải:

- $T_{xtal} = 1/F_{xtal} = 1/11,0592\text{MHz} = 0,0904\mu\text{s}$
- $T_{timer} = 12T_{xtal} = 12 \times 0,0904 = 1,085\mu\text{s}.$
- Số đếm $N = \text{FFFFH} - \text{FFF2H} + 1 = 0\text{EH} = 14\text{D}.$
- $T_{trễ} = N \times 1,085\mu\text{s} = 14 \times 1,085\mu\text{s} = 15,19\mu\text{s}$
nửa chu kỳ.
- $T_{trễ\text{ cả chu kỳ}} = 2 \times 15,19\mu\text{s} = 30,38\mu\text{s}$ là
thời gian trễ do bộ định thời tạo ra

Ví dụ 6:

Tính chính xác tần số xung vuông được tạo ra trên chân P1.5 ở ví dụ vừa nêu có tính đến thời gian trễ thực hiện các lệnh.

Giải:

Dùng bảng phụ lục bổ sung chu kỳ máy

Số chu kỳ máy

LAP2:	MOV TL0,#0F2H	2
	MOV TH0,#0FFH	2
	SETB TR0	1
LAP:	JNB TF0 LAP	14
	CPL P1.5	1
	CLR TR0	1
	CLR TF0	1
	SJMP LAP	2

;--- Tạo trễ bằng timer 0

DELAY:

SETB TR0 1

AGAIN: JNB TF0,AGAIN

14

CLR TR0 1

CLR TF0 1

RET 1

Cộng 24

Thời gian trễ là $T = 2 \times 24 \times 1.085 \mu s = 52,08 \mu s$ và tần số là $F = 19,20 KHz$.

Ví dụ 7:

Tính thời gian trễ do Timer0 tạo ra ở đoạn chương trình sau (bỏ qua thời gian thực hiện của các lệnh):

MOV TMOD,#01

HERE:

MOV TL0,#3EH

MOV TH0,#0B8H

SETB P2.3

SETB TR0

AGAIN:

JNB TF0,AGAIN

CLR TR0

CLR TF0

CLR P2.3

Giải:

Số đếm của bộ định thời:

$$N = (FFFF - B83E + 1) = 47C2H = 18370D$$

$$\text{Thời gian trễ} = 18370 \times 1,085\mu s = 19,93145ms.$$

Ví dụ 8:

Chọn TH và TL ở ví dụ 9.7 để được thời gian trễ cực đại, bỏ qua thời gian thực hiện các lệnh vòng lặp.

Giải:

Thời gian trễ lớn nhất khi $TH=TL=0$, lúc đó bộ định thời đếm từ 0 đến FFFFH và quay về 0.

```
CLR    P2.3                ;Xoá P2.3  
MOV     TMOD,#01          ;Timer0,Mode1(16
```

bit)

HERE:

```
MOV     TL0,#0
```

```
MOV     TH0,#0
```

```
SETB    P2.3
```

```
SETB    TR0                ;Khởi động bộ Timer0
```

AGAIN:

JNB	TF0,AGAIN	;Lặp lại nếu TF0=0
CLR	TR0	;Dừng bộ định thời
CLR	TF0	;Xoá cờ TF0
CLR	P2.3	

**TH=TL=0 thì bộ định thời đếm tăng từ 0 đến FFFFH rồi quay về 0 và bật cờ TF.
Kết quả đếm được 65536.
Thời gian trễ = $65536 \times 1.085\mu s = 71,1065ms$.**

Tạo thời gian trễ cho trước

Cho trước thời gian trễ. Xác định giá trị nạp cho TH và TL. Giả sử tần số thạch anh là 11,0592MHz.

Lập trình tạo thời gian trễ cho tróc giả thiết $F_{xtal}=11,0592\text{Mhz}$

- 1. Chia thời gian trễ cho $1,085\mu\text{s}$
được trị N**
- 2. Tính $M=65536 - N$**
- 3. Chuyển M sang dạng hexa $yyxx$.**
- 4. Nạp $TL = xx$ và $TH = yy$.**

Ví dụ 9:

Viết chương trình cho Timer0 tạo xung trên chân P2.3 có độ rộng 5ms.

Cho $F_{xtal} = 11,0592\text{MHz}$.

Giải:

- Tần số $F_{xtal} = 11,0592\text{MHz}$ nên khoảng thời gian khi bộ đếm tăng 1 là $1,085\mu\text{s}$.
- Độ rộng xung 5ms cần số đếm $N = 5\text{ms}/1,085\mu\text{s} = 4608$.
- Vậy $65536 - 4608 = 60928\text{D} = \text{EE00H}$.
- Do đó giá trị nạp vào $\text{TH}=\text{EEH}$ và $\text{TL}=00$.

Chương trình

CLR	P2.3	;Xoá bit P2.3
MOV TMOD,#01		;Timer0, Mode1 (16 bit

HERE:

MOV TL0,#0		
MOV	TH0,#EEH	
SETB	P2.3	;Tạo xung
SETB	TR0	;Khởi động Timer0

AGAIN:

JNB TF0,AGAIN	;Lặp đến khi TF0=1
CLR TR0	;Dừng bộ định thời
CLR TF0	;Xoá cờ TF0

Ví dụ 10:

Cho $F_{xtal}=11,0592$ MHz.

Viết chương trình tạo sóng xung vuông tần số 50 Hz ở chân P2.3.

Giải:

Các bước thực hiện như sau:

a) Tính chu kỳ sóng vuông:

$$1/50\text{Hz}=20\text{ms}$$

b) Thời gian nửa chu kỳ phần cao:

$$0.5T=10\text{ ms.}$$

c) Số đếm tạo thời gian

$$10\text{ms}/1,085\mu\text{s}=9216$$

d) Số cần nạp 65536-

$$9216=56320\text{D}=\text{DC00H},$$

Như vậy TH = DCH và TL = 00H.

Chương trình

MOV TMOD,#10H	;Timer0,Mode1
AGAIN:	
MOV TL1,#00	
MOV TH1,#0DCH	
SETB TR1	;Khởi động Timer1
BACK:	
JNB TF1,BACK	;Lặp đến khi TF1=1
CLR TR1	;Dừng bộ định thời
CPL P2.3	;Tạo xung P1.5
CLR TF1	;Xoá cờ TF1
SJMP AGAIN	;Nạp lại

Tạo thời gian trễ lớn

Tạo thời gian trễ phụ thuộc vào:

- Tần số thạch anh F_{xtal}
- Thanh ghi 16 bit của bộ định thời ở chế độ 1
- Thời gian trễ lớn nhất khi $TH=TL=0$ cho $T_{delaymax}=71ms$.
- Nếu cần thời gian trễ lớn hơn có thể sử dụng thêm vòng lặp.

Ví dụ 11:

Xác định thời gian trễ của chương trình sau:

MOV TMOD,#10H
MOV R3,#200

LAP2:

MOV TL1,#08
MOV TH1,#01
SETB TR1

LAP:

JNB TF1,LAP
CLR TR1
CLR TF1
DJNZ R3,LAP2

Giải:

$$\text{TH-TL} = 0108\text{H} = 264\text{D}$$

$$65536 - 264 = 65272.$$

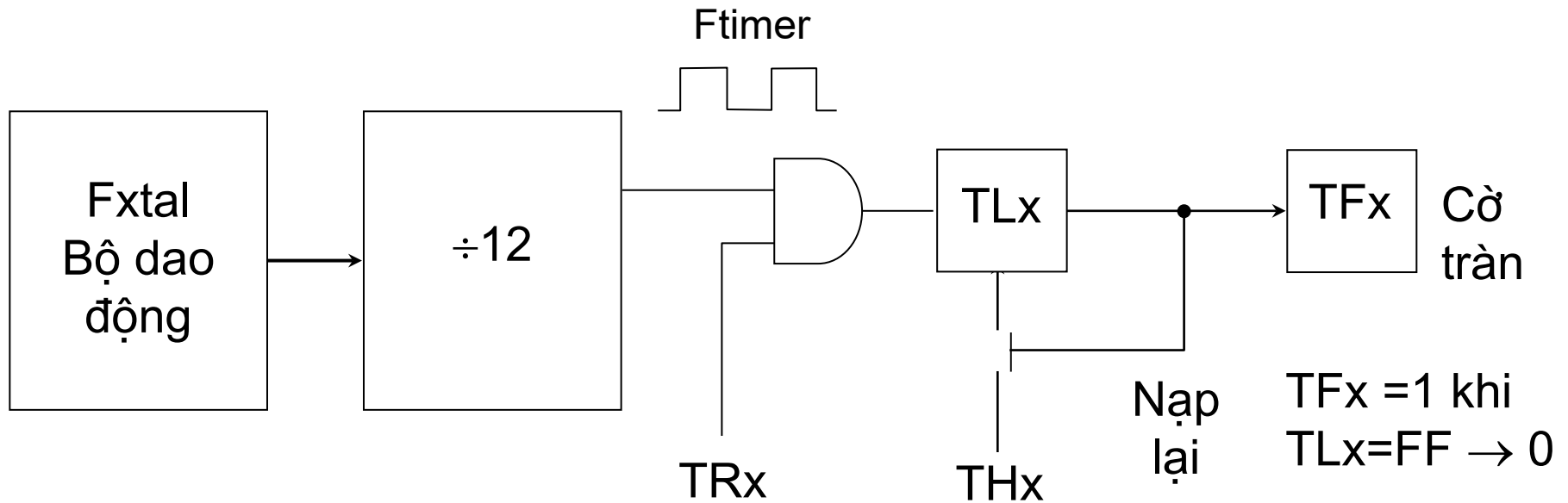
$$65272 \times 1,085 \mu\text{s} = 70,820 \text{ ms}.$$

$$\text{Vậy } 200 \times 70,820 \text{ ms} = 14,164024 \text{ s}.$$

c) Lập trình chế độ 2 (chế độ tự nạp lại)

- Lưu đồ làm việc chế độ 2
- Đặc điểm chế độ 2
- Các bước lập^u trình

Lưu đồ làm việc chế độ 2



Đặc điểm chế độ 2

1. Là bộ đếm 8 bit nên giá trị nạp TH từ 0-FFH.
2. Sau khi khởi động bộ định thời, TH được sao vào TL và TL được đếm tăng cho đến khi đạt max là FFH.
3. Khi TL từ FFH về 0, cờ TF=1 thì TL được tự động nạp lại từ TH.
4. Để đếm lặp lại, cần xoá TF=0 và bộ định thời tự động nạp số đếm mà không cần lập trình.

Chú ý:

- Chế độ 2 còn gọi là chế độ tự nạp.
- Chế độ 2 có nhiều ứng dụng, trong đó có thiết lập tần số baud trong truyền thông nối tiếp.

Các bước lập trình chế độ 2

- 1. Nạp TMOD**
- 2. Nạp TH**
- 3. Khởi động bộ định thời**
- 4. Giám sát cờ TF**
- 5. Xoá cờ TF**
- 6. Lặp lại bước 4**

Ví dụ 12:

Cho $F_{xtal} = 11,0592\text{MHz}$.

Sử dụng Timer1 lập trình tạo xung vuông trên chân P1.0 có chu kỳ $T=544,66\mu\text{s}$.

Các bước lập trình chế độ 2

- 1. Nạp TMOD**
- 2. Nạp TH**
- 3. Khởi động bộ định thời**
- 4. Giám sát cờ TF**
- 5. Xoá cờ TF**
- 6. Lặp lại bước 4**

Giải:

a) Xác định hệ số đếm

$$N = T_{\text{delay}} / T_{\text{timer}} =$$

$$544,66 \mu\text{s} / (2 \times 1,085 \mu\text{s}) = 251$$

$$\text{Số nạp vào TH} = 256 - 251 = 5$$

MOV TMOD,#20H ;Timer1/Mode2

MOV TH1,#5

SETB TR1 ;Khởi động Timer1

BACK:

JNB TF1,BACK

CPL P1.0

CLR TF1

SJMP BACK ;Chế độ 2 tự nạp lại

Ví dụ 13:

Cho $F_{xtal} = 11,0592\text{MHz}$.

**Lập trình chế độ 2 để tạo chu kỳ
xung lớn nhất**

Giải:

Để có được chu kỳ T lớn nhất (tần số nhỏ nhất) thì $TH=0$.

Khi đó $T=2 \times 256 \times 1,085 \mu s = 555,52 \mu s$ Tần số nhỏ nhất: $1/T=1,8$ KHz.

Tạo thời gian lớn ở chế độ 2

- Chế độ 2 cho phép tạo $t_{\max}=555 \mu\text{s}$.
- Để tạo thời gian trễ lớn, cần đặt trong vòng lặp.

Ví dụ 14:

Xác định tần số xung vuông trên chân P1.0 ở đoạn chương trình sau:

```
MOV  TMOD,#02H
```

AGAIN:

```
MOV  TH0 #0
```

```
MOV  R5,#250
```

```
ACALL DELAY
```

```
CPL  P1.0
```

```
SJMP AGAIN
```

DELAY:

```
SETB  TR0
```

BACK:

JNB TF1,BACK

CLR TR0

CLR TF0

DJNZ R5,DELAY

RET

Giải:

Chu kỳ xung:

$$T = 2 \times (250 \times 256 \times 1,085 \mu s) = 138,88 ms$$

$$\text{Tần số } F = 1/T = 7,2 \text{ Hz.}$$

Ví dụ 15:

Cho chế độ 2. Hãy xác định giá trị cần nạp (dạng Hexa) vào TH cho các trường hợp sau:

- a) MOV TH1,#-3**
- b) MOV TH1,#-12**
- c) MOV TH0,#-48**
- d) MOV TH0,#-60**
- e) MOV TH1,#-200**

Giải:

Số thập phân

-3

-12

-48

-60

-200

Số bù hai (giá trị TH)

FDH

F4H

D0H

C4H

38H

Nạp số đếm âm ở chế độ 2

- Chế độ 2 cho phép nạp TH số đếm âm.
- Trình hợp dịch tự dịch số âm.
- Cách làm này giải phóng công việc cho lập trình viên.

Ví dụ:

- `MOV TH0, #-100`
- Trình hợp dịch sẽ tính $-100D = 9CH$ và thực hiện `MOV TH0, #9CH`.

Ví dụ 16:

Hãy xác định thời gian trễ được tạo ra trong đoạn chương trình sau, cho $F_{xtal}=11,0592\text{ MHz}$.


```
MOV  TMOD,#02H
MOV  TH0,#-150
SETB      TR0
```

AGAIN:

```
SETB      P1.3
ACALL DELAY
CLR  P1.3
ACALL DELAY
SJMP AGAIN
```

DELAY:

BACK:

```
SETB      TR0
JNB  TF0,BACK
CLR  TR0
CLR  TF0
RET
```

Giải:

Số đếm là 150.

$$\text{Tdelay} = 150 \times 1.085 \mu\text{s} = 162 \mu\text{s}$$

Chế độ 0

- Chế độ 0 giống chế độ 1, trừ bộ đếm là 13 bit.
- Giá trị đếm: từ 0 đến 1FFFF.
- Khi đếm đến 1FFFFH thì bộ đếm sẽ quay về 0 và bật cờ TF.
- Chế độ 0 ít được sử dụng mà để tương thích với bộ VDK thế hệ trước

2. Bộ đếm và lập trình

a) Bộ đếm

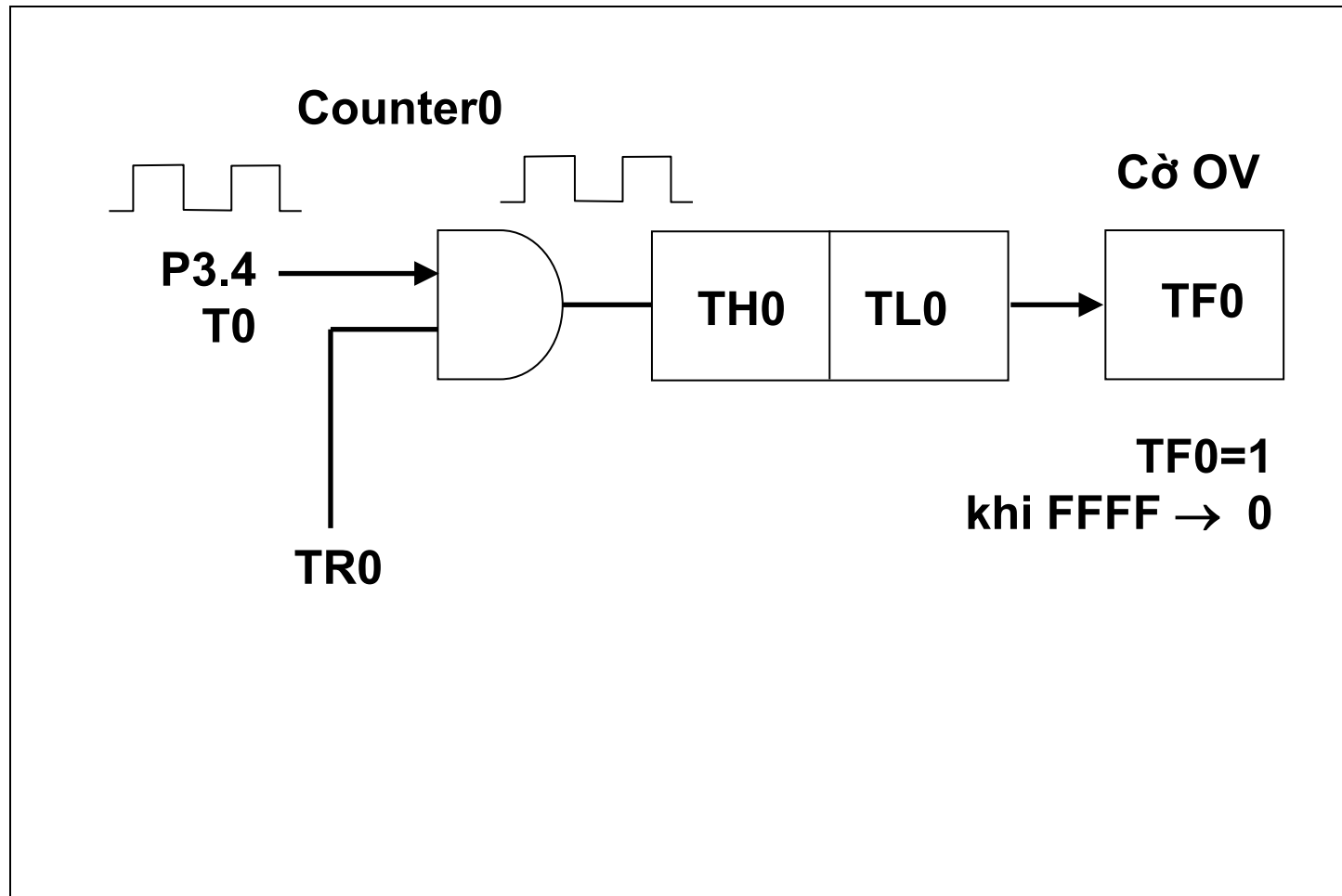
b) Đặc điểm lập trình

c) Các bước lập trình

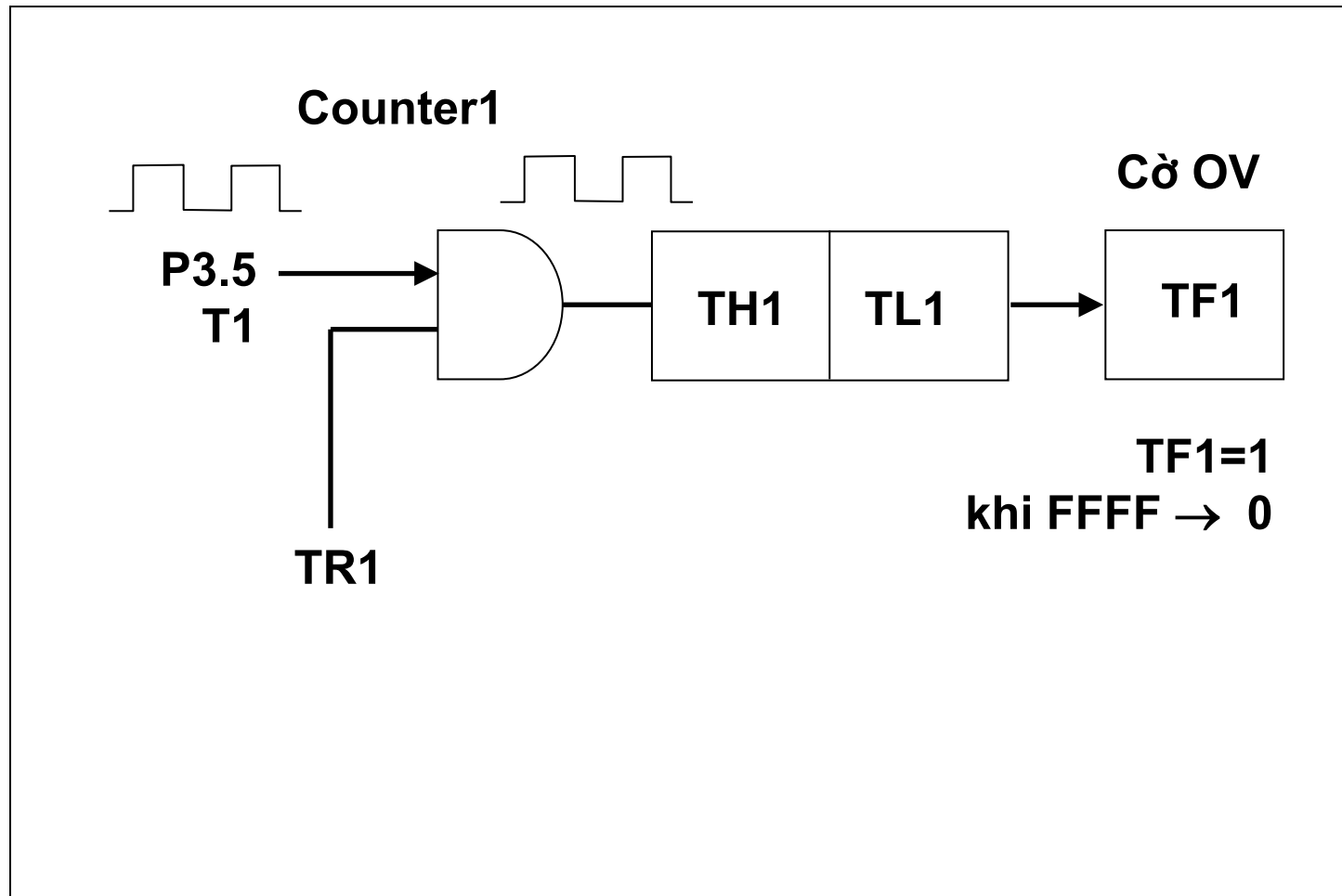
a) Bộ đếm

- Timer ngoài chức năng tạo trễ còn được dùng để đếm xung (Counter).
- Bộ đếm làm việc tương tự bộ định thời, chỉ khác về nguồn tần số.
- Nguồn tần số bộ đếm là xung bên ngoài làm tăng thanh ghi TH và TL.
- Nguồn tần số bộ định thời là tần số thạch anh.

Bộ đếm Counter0 làm việc chế độ 1



Bộ đếm Counter1 làm việc chế độ 1



Bit $\overline{C/T}$ của thanh ghi TMOD

- Khi làm Counter $\overline{C/T}=1$, nguồn xung bên ngoài đặt vào chân T0 (P3.4) hoặc T1 (P3.5).
- Khi làm Timer $\overline{C/T}=0$ nguồn xung là thạch anh.
- Như vậy TMOD của Counter tăng thêm 4 so với TMOD Timer.

Bảng 9.1. Chân cổng P3 dùng cho Timer0 và Timer1

Chân	Chân cổng	Chức năng	Mô tả
14	P3.4	T0	Đầu vào từ ngoài của Timer0
15	P3.5	T1	Đầu vào từ ngoài của Timer1

b) Đặc điểm lập trình

Lập trình giống với bộ định thời, chỉ khác về nguồn tần số nên khác TMOD.

Có 2 lưu ý:

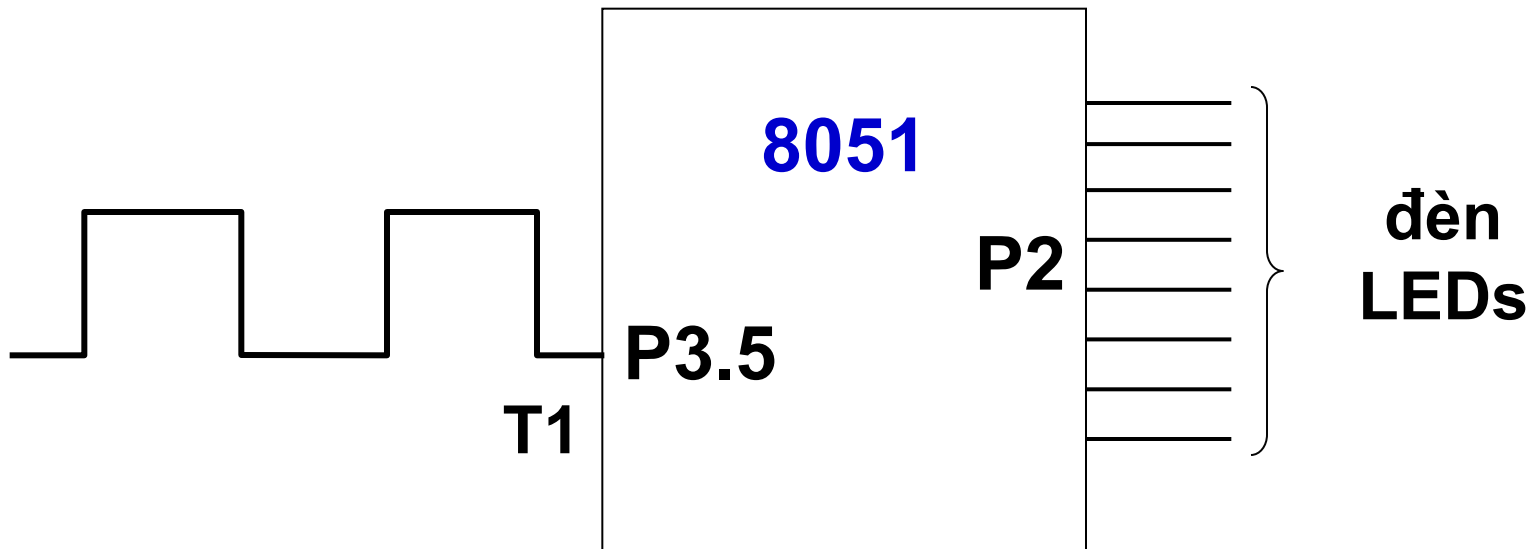
- **Vì $C/\bar{T}=1$ nên với Counter0**
 - **TMOD=05 ở mode1**
 - **TMOD=06 ở mode2**
- **Cho P3.4=1, (P3.5=1) làm đầu vào**

c) Các bước lập trình bộ đếm

- 1. Nạp TMOD**
- 2. Nạp số đếm TH, TL**
- 3. Đặt $P3.4=1$ ($P3.5=1$) làm đầu vào**
- 4. Khởi tạo bộ đếm**
- 5. Giám sát cờ tràn**
- 6. Xóa TR, TF**
- 7. Lặp lại**

Ví dụ 17:

Cho xung đồng hồ cấp tới chân T1.
Viết chương trình Counter1, Mode 2 để đếm xung và hiển thị số đếm TL1 trên cổng P2.



Đếm xung cổng T1 và hiển thị trên đèn LED

Giải:

MOV TMOD,#60H

MOV TH1,#0

SETB P3.5

AGAIN:

SETB TR1

BACK:

MOV A,TL1

MOV P2,A

JNB TF1,BACK

CLR TR1

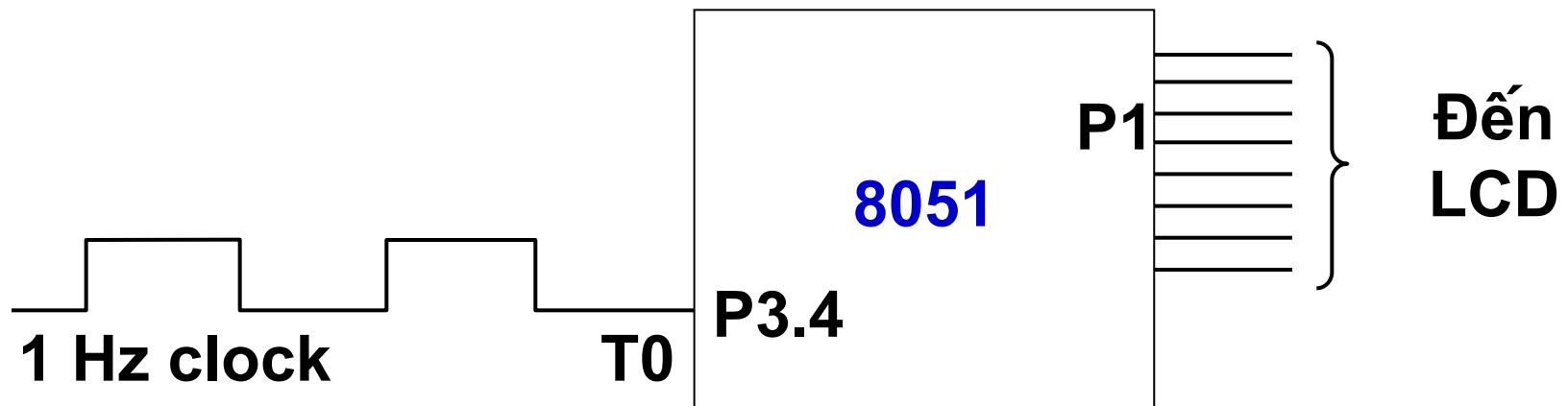
CLR TF1

SJMP AGAIN

Ví dụ 18:

Cho dãy xung có tần số 1 Hz được nối tới chân vào P3.4.

Viết chương trình hiển thị bộ đếm 0 trên màn LCD. Giá trị ban đầu của TH0 là -60.



Giải:

Để hiển thị số đếm TL trên LCD cần chuyển dữ liệu 8 bit nhị phân về mã ASCII.

```
ACALL LCD_SETUP  
MOV      TMOD,#06H  
MOV      TH0,#-60  
SETB     P3.4
```

LAP:

```
SETB     TR0
```

LAP2:

```
MOV    A,TL0  
ACALL  CONV  
ACALL  DISLAY  
JNB    TF0,LAP2  
CLR    TR0  
CLR    TF0  
SJMP   LAP
```

;Chuyển vào R2,R3,R4

;Chuyển m□nhị phân về m□ASCII hiển thị trên LCD

```
CONV: MOV  B,#10
      DIV  AB
      MOV  R2,B                ;Lu số thấp
      MOV  B,#10
      DIV  AB
      ORL   A,#30H
      MOV  R4,A                ;R4 Lu MSB
      MOV  A,B
      ORL   A,#30H
      MOV  R3,A                ;Lu số ASCII vào R3
      MOV  A,R2
      ORL   A,#30H
      MOV  R2,A                ;Lu số ASCII vào R2
      RET
```

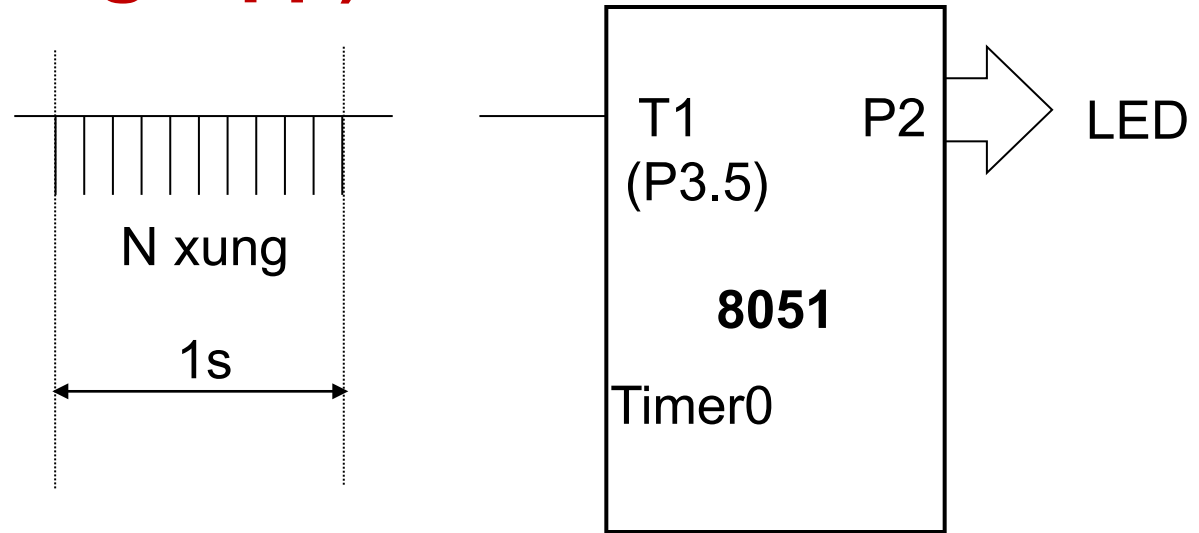
Tóm tắt bài 3

Bộ Timer/Counter làm việc dưới 2 dạng: là Timer và Counter.

- **Dạng Timer: Đếm xung thạch anh**
 - TMOD=01 đếm 16 bit (Timer0)
 - TMOD=02 đếm 8 bit tự nạp (Timer0)
- **Dạng Counter: Đếm xung ngoài**
 - TMOD=05 đếm 16 bit (Counter0)
 - TMOD=06 đếm 8 bit, tự nạp (Counter0)

(Timer1, Counter1 tương tự)

Ví dụ 19: (VD tổng hợp)



Lập trình đo tốc độ động cơ bằng cách liên tục đếm số xung N từ encoder trong thời gian 1s theo cách sau:

- Dãy xung N đưa đến chân T1 của Counter1.
- Dùng Timer0 tạo thời gian 1s.
- Sau 1s số xung 16 bit đếm được lưu ở R2 (byte thấp) và R3 (byte cao).

(gợi ý: $1s = 15 \times 61444 \times 1,085 \mu s$, $61444D = F004H$. Bỏ qua thời gian thực hiện lệnh)

```
    SETB P3.5
    MOV TMOD, #51H
    MOV TH1, #0
    MOV TL1, #0
LAP3: MOV R0, 15
LAP2: MOV TH0, #0FH
      MOV TL0, #0FCH
      SETB TR0
      SETB TR1
LAP:  JNB TF0, LAP
      CLR TF0
      DJNB R0, LAP2
      MOV R2, TL1
      MOV R3, TH1
      CLR TR1
      CLR TR0
      SJMP LAP3
```

Tổ chức Ngắt và lập trình

NỘI DUNG

- 1. Tổ chức ngắt**
- 2. Lập trình ngắt bộ định thời**
- 3. Lập trình ngắt ngoài**
- 4. Lập trình ngắt truyền thông nối tiếp**
- 5. Lập trình ưu tiên ngắt**

1. Tổ chức ngắt

- a) Giới thiệu
- b) Phương pháp phục vụ thiết bị
- c) Thanh ghi cho phép ngắt IE

a) Giới thiệu

06 ngắt của 8051

Ngắt	Địa chỉ	Chân
RESET	00H	9
Ngắt ngoài INT0	03H	12
Ngắt Timer0 (TF0)	0BH	(P3.2)
Ngắt ngoài INT1	13H	13
Ngắt Timer1 (TF1)	1BH	(P3.3)
Ngắt cổng COM (RI và TI)	23H	

b) Phương pháp phục vụ thiết bị

- Một bộ vi điều khiển có thể phục vụ một số thiết bị.
- Có hai cách phục vụ thiết bị, đó là
 - Phương pháp thăm dò (Polling)
 - Phương pháp ngắt (Interrupt)

Phương pháp thăm dò:

- Bộ VĐK liên tục kiểm tra tình trạng của thiết bị và khi điều kiện được đáp ứng thì tiến hành phục vụ thiết bị.
- Sau đó, bộ vi điều khiển chuyển sang kiểm tra trạng thái của thiết bị tiếp theo cho đến khi tất cả đều được phục vụ.

Đặc điểm của phương pháp thăm dò:

- Chiếm nhiều thời gian của bộ VĐK
- Phục vụ thiết bị theo thứ tự định sẵn mà không theo thứ tự ưu tiên.
- Kinh tế hơn trong xây dựng mạch

Phương pháp ngắt:

- Khi có một thiết bị cần được phục vụ thì thiết bị sẽ báo cho bộ vi điều khiển bằng cách gửi một tín hiệu ngắt.
- Khi nhận được tín hiệu này, bộ vi điều khiển ngừng mọi công việc đang thực hiện để chuyển sang phục vụ thiết bị.
- Mỗi ngắt có Chương trình con phục vụ ngắt ISR (Interrupt Service Routine)

Đặc điểm của Phương pháp ngắt:

- Tiết kiệm được nhiều thời gian cho bộ VĐK
- Bộ VĐK phục vụ thiết bị có thể theo mức ưu tiên được gán.
- Có thể bỏ qua một yêu cầu phục vụ của thiết bị.
- Chi phí xây dựng hệ có thể cao hơn.

Bảng vectơ ngắt

- Mỗi ngắt luôn có một trình phục vụ ngắt gọi là ISR
- Khi một ngắt được kích hoạt thì bộ vi điều khiển chạy ISR tương ứng
- Mỗi ISR có địa chỉ được lưu cố định trong bộ nhớ.
- Tập hợp các ô nhớ lưu địa chỉ của tất cả các ISR được gọi là bảng vector ngắt

Trình tự thực hiện ngắt

Khi một ngắt được kích hoạt thì:

- **Bộ VĐK kết thúc lệnh hiện tại và lưu địa chỉ của lệnh kế tiếp (PC) vào ngăn xếp.**
- **Lưu lại trạng thái hiện hành của tất cả các ngắt.**
- **Nhảy tới địa chỉ ISR của ngắt trong bảng vectơ ngắt và thực hiện ISR đó.**
- **Kết thúc ISR, VĐK gặp lệnh RETI và trở về nơi bị ngắt và thực hiện tiếp lệnh sau ngắt.**

06 ngắt của 8051

8051 có 5 ngắt dành cho người dùng. Nhiều tài liệu xem RESET là ngắt thứ 6

- **TF0, TF1** (Timer0, Timer1): 02 ngắt bộ định thời. Địa chỉ ở bảng vector ngắt là 0BH và 1BH.
- **INT0, INT1**: 02 ngắt phần cứng dành cho thiết bị ngoài nối tới chân 12 (P3.2) và 13 (P3.3) cổng P3. Tên gọi khác **EX1** và **EX2**. Địa chỉ ở bảng vector ngắt là 03H và 013H.

- **(RI+TI)** Ngắt truyền thông nối tiếp phục vụ thu/phát dữ liệu. Địa chỉ trong bảng vector ngắt là 023H.
- **RESET:** Khi RESET được kích hoạt mức cao, thì 8051 nhảy về 0, là địa chỉ bật lại nguồn

c) Thanh ghi cho phép ngắt

- Để sử dụng ngắt, cần thiết lập thanh ghi cho phép ngắt **IE** (Interrupt Enable).
- Thanh ghi IE cho phép hoặc cấm các ngắt.
- Thanh ghi IE có thể định địa chỉ bit.

Thanh ghi cho phép ngắt IE

D7	D6	D5	D4	D3	D2	D1	D0
EA	--	ET	ES	ET	EX	ET	EX
		2		1	1	0	0

- EA IE.7** Nếu EA=0 thì che tất cả các ngắt.
Nếu EA=1 thì ngắt được mở/cấm bằng cách đặt/xoá bit tương ứng.
- ET2 IE.5** Cho_phép/cấm ngắt tràn Timer2 (8952)
- ES IE.4** Cho_phép/cấm ngắt cổng nối tiếp
- ET1 IE.3** Cho_phép/cấm ngắt tràn Timer1
- EX1 IE.2** Cho_phép/cấm ngắt INT1
- ET0 IE.1** Cho_phép/cấm ngắt tràn Timer0
- EX0 IE.0** Cho_phép/cấm ngắt INT0

Thực hiện khi cho phép một ngắt

1. Đặt Bit $EA=1$ ($IE.7=1$) cho phép sử dụng ngắt.
2. Tiếp theo, dùng ngắt nào thì thiết lập lên 1 bit ngắt tương ứng ($IE.x=1$).
3. $EA=0$ thì cấm tất cả các ngắt.

Ví dụ 1:

Hãy viết các lệnh thực hiện:

1. Cho phép các ngắt: Cổng COM (ES), Timer0 (ET0) và INT1 (EX1)
2. Cấm ngắt Timer0.
3. Dùng 1 lệnh cấm tất cả các ngắt

Thanh ghi cho phép ngắt IE

D7	D6	D5	D4	D3	D2	D1	D0
EA	--	ET2	ES	ET1	EX1	ET0	EX0

Giải:

1. MOV IE,#96H ;Cho phép ngắt ES, ET0, EX1

Cũng có thể dùng lệnh định địa chỉ bit:

SETB IE.7

SETB IE.4

SETB IE.1

SETB IE.2

2. CLR IE.1 ;Timer0

3. CLR IE.7 ;Cấm tất cả các ngắt

2. Lập trình ngắt bộ định thời

- a) Cách thức sử dụng ngắt bộ định thời**
- b) Lập trình ngắt bộ định thời chế độ 2**
- c) Lập trình ngắt bộ định thời chế độ 1**

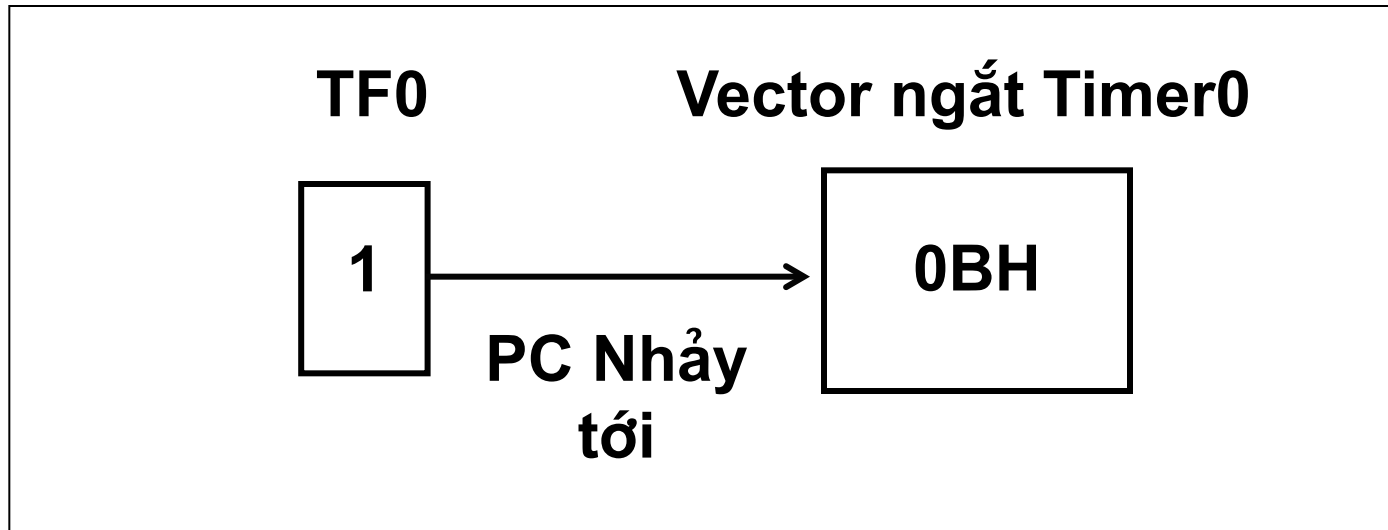
a) Cách thức sử dụng ngắt bộ định thời

- Khi Timer đạt Max (FFH hoặc FFFFH) và về 0 thì Cờ **TF=1**.
- Phương pháp giám sát cờ TF=1 cho phép tạo được thời gian nhưng vì VĐK phải liên tục kiểm tra cờ TF nên tốn thời gian của VĐK.
- Phương pháp khác hiệu quả hơn là sử dụng ngắt.

Cách thức làm việc Ngắt bộ định thời

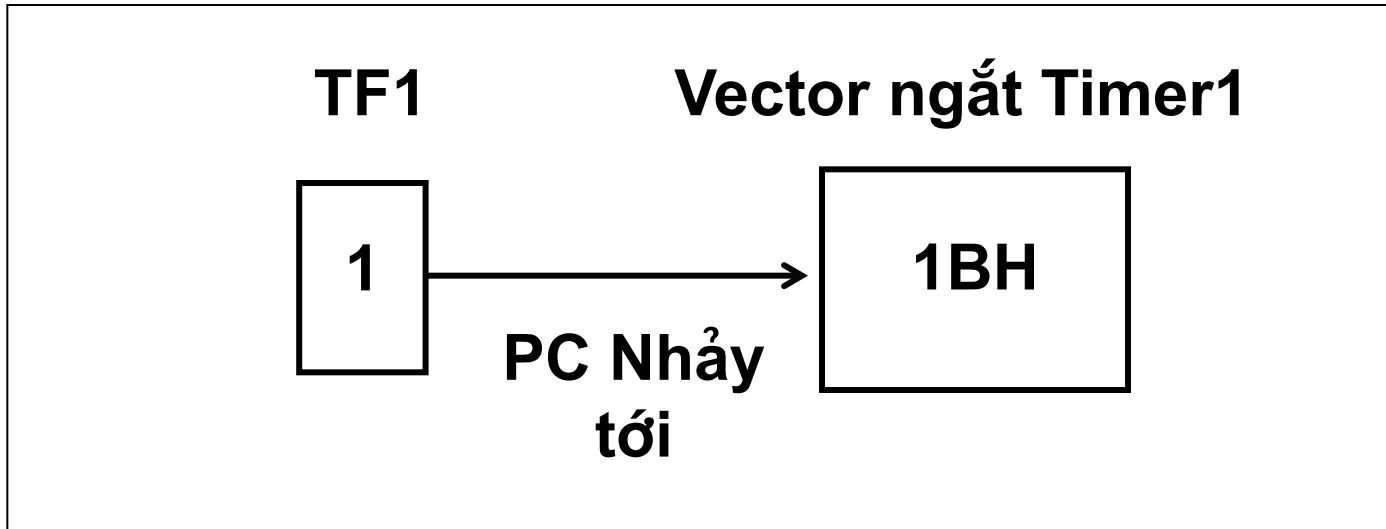
- Đầu tiên thiết lập cho phép ngắt IE.
- Khi đếm tràn thì cờ $TFx=1$, bộ VĐK ngừng mọi công việc đang thực hiện, nhảy tới bảng vector ngắt để chạy ISR.
- Lập trình ISR tại bảng vectơ ngắt

Bộ định thời Timer0



Hình 11.3. Cờ ngắt bộ định thời TF0

Bộ định thời Timer1



Hình 11.3. Cờ ngắt bộ định thời TF1

Đặc điểm lập trình Timer dùng ngắt

- 1. Thiết lập chế độ làm việc.**
 - **TMOD=01 (hoặc 02)**
 - **Nạp TLx, THx**
- 2. Thiết lập thanh ghi IE cho phép ngắt**
 - **IE = 82H với Timer0**
 - **IE = 88H với Timer1**
- 3. Lập trình tại địa chỉ ngắt tương ứng**
 - **0BH với Timer0**
 - **1BH với Timer1**

D7	D6	D5	D4	D3	D2	D1	D0
EA	--	ET	ES	ET	EX	ET	EX
		2		1	1	0	0

- EA IE.7** Nếu EA=0 thì che tất cả các ngắt.
Nếu EA=1 thì ngắt được mở/cấm bằng cách đặt/xoá bit tương ứng.
- ET2 IE.5** Cho_phép/cấm ngắt tràn Timer2 (8952)
- ES IE.4** Cho_phép/cấm ngắt cổng nối tiếp
- ET1 IE.3** Cho_phép/cấm ngắt tràn Timer 1
- EX1 IE.2** Cho_phép/cấm ngắt INT1
- ET0 IE.1** Cho_phép/cấm ngắt tràn Timer 0
- EX0 IE.0** Cho_phép/cấm ngắt INT0

Thanh ghi cho phép ngắt IE

Bảng 11.1. Bảng vector ngắt của 8051

Ngắt	Địa chỉ	Chân
RESET	00H	9
Ngắt ngoài INT0	03H	12
Ngắt Timer0 (TF0)	0BH	(P3.2)
Ngắt ngoài INT1	13H	13
Ngắt Timer1 (TF1)	1BH	(P3.3)
Ngắt cổng COM (RI và TI)	23H	

b) Lập trình ngắt bộ định thời chế độ 2

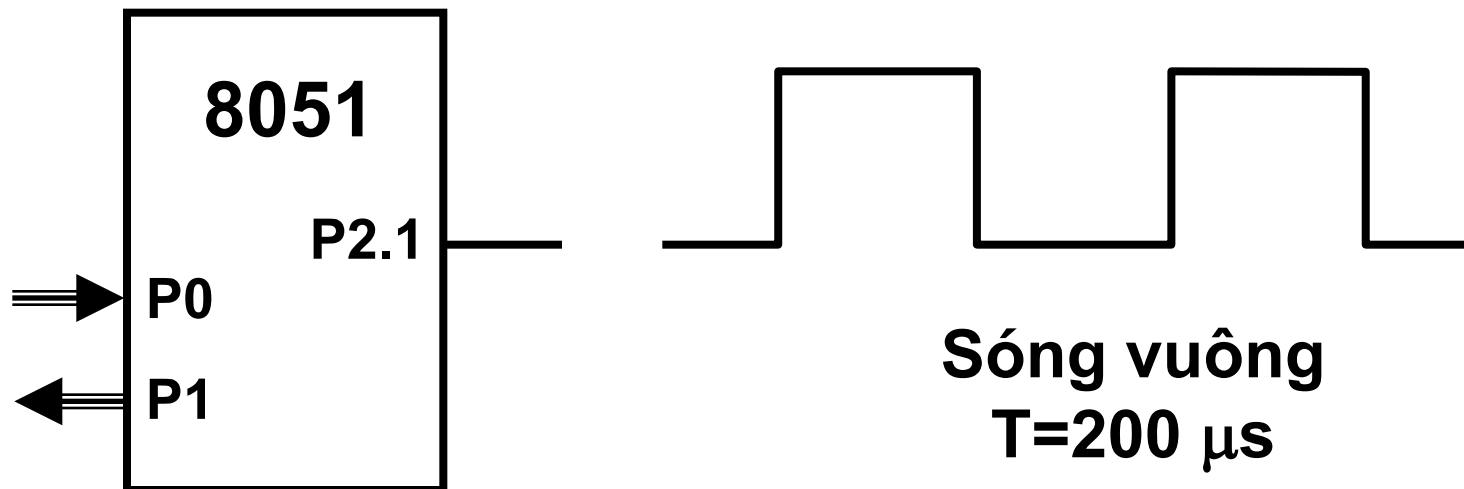
Các bước lập trình

- 1. Lập trình tránh bảng vector ngắt.**
- 2. Thiết lập TMOD và nạp TH.**
- 3. Thiết lập chế độ hỗ trợ nếu cần.**
- 4. Thiết lập thanh ghi IE.**
- 5. Khởi tạo bộ đếm.**
- 6. Tạo vòng lặp xử lý công việc và đợi ngắt.**
- 7. Lập trình tại địa chỉ ngắt.**

Ví dụ 2:

Cho $F_{xtal} = 11,0592\text{MHz}$, lập trình để:

- a) Liên tục nhận dữ liệu 8 bit ở cổng P0.
- b) Gửi dữ liệu nhận được đến cổng P1.
- c) Đồng thời, dùng Timer0 tạo sóng vuông chu kỳ $200\text{ }\mu\text{s}$ trên chân P2.1.



$T_{vd\kappa} = 1/11,0592 \text{ MHz} = 0,0904 \mu\text{s}$

$T_{\text{timer}} = 12T = 12 \times 0,0904 \mu\text{s} = 1.085 \mu\text{s}$

Nạp TH0: $N = 200 \mu\text{s} / 2 / 1,085 \mu\text{s} = 92$

Giải:

**ORG 0H
LJMP MAIN**

**ORG 0BH ;ISR của Timer0
CPL P2.1
RETI**

;-Chương trình chính

ORG 30H

MAIN:

MOV TMOD,#02H ;Timer0, Mode2, tự nạp

MOV P0,#0FFH

MOV TH0,#-92 ;Đặt TH0=A4H (-92)

MOV IE,#82H ;Cho phép Timer0

SETB TR0 ;Khởi động Timer0

BACK:

MOV A,P0

MOV P1,A

SJMP BACK

END

Lưu ý ở chương trình trên:

1. Trình ứng dụng tránh bảng vector ngắt và dùng địa chỉ sau bảng vectơ ngắt là 30H.
2. ISR của Timer0 có kích thước nhỏ nên được đặt trong không gian nhớ của ngắt tại bảng vector ngắt, từ địa chỉ 0BH.
3. Thiết lập chế độ dùng ngắt là lệnh “MOV IE, #82H”.

4. Trong khi nhận dữ liệu tại P0 gửi sang P1, thì Timer0 vẫn đếm và mỗi khi đếm về 0 cờ TF0=1, VĐK lập tức nhảy đến địa chỉ 0BH để thực hiện ISR Timer0.

5. Trước khi kết thúc ISR lập trình viên không cần xóa cờ “CLR TF0” vì VĐK tự động xóa cờ TF khi nhảy đến bảng vector ngắt.

c) Lập trình ngắt bộ định thời chế độ 1

- **Đặc điểm lập trình**
- **Lập trình xung đối xứng**
- **Lập trình tạo xung không đối xứng**

Đặc điểm lập trình

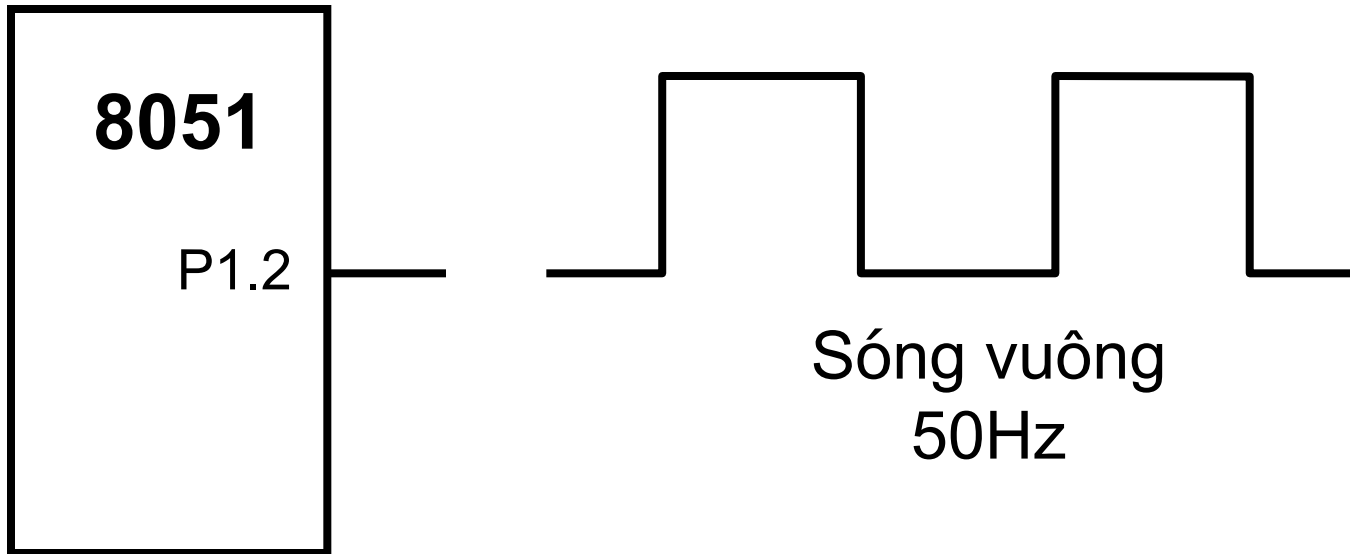
- Tương tự với lập trình chế độ 2
- Cũng lập trình tại bảng vectơ ngắt 0BH hoặc 1BH
- Điểm lưu ý là cần nạp lại THx, TLx (do chế độ này không tự nạp lại TH, TL).

Các bước lập trình chế độ 1

- 1. Lập trình tránh bảng vector ngắt.**
- 2. Thiết lập TMOD và nạp TL, TH.**
- 3. Thiết lập chế độ hỗ trợ nếu cần.**
- 4. Thiết lập thanh ghi IE.**
- 5. Khởi tạo bộ đếm.**
- 6. Tạo vòng lặp xử lý công việc và đợi ngắt.**
- 7. Lập trình tại địa chỉ ngắt và nạp lại TH, TL.**

Ví dụ 3:

Viết chương trình tạo sóng vuông tần số 50Hz trên chân P1.2, sử dụng ngắt Timer1, giả thiết $F_{xtal} = 11,0592\text{MHz}$.



Giải:

```
ORG    0  
LJMP   MAIN  
  
ORG    1BH  
CPL    P1.2  
MOV    TL1,#0  
MOV    TH1,#0DCH  
RETI
```

ORG 30H

MAIN:

MOV TMOD,#10H

MOV TL1,#0

MOV TH1,#0DCH

MOV IE,#88H

SETB TR1

HERE:

SJMP HERE

END

- **Lập trình chế độ 1 tạo xung không đối xứng**
 - Xung không đối xứng có phần dương và âm khác nhau.
 - Trường hợp xung ngắn có thể sử dụng độ trễ của các lệnh.
 - Trường hợp xung dài thì lập trình đếm như thông thường.

Ví dụ 4:

Viết chương trình:

- a) Nhận dữ liệu ở cổng P0**
 - b) Gửi dữ liệu nhận được ra cổng P1.**
 - c) Tạo sóng vuông chân P2.1 có mức cao $1085\mu s$ và mức thấp $15\mu s$.**
- Cho $F_{xtal}=11,0592MHz$, dùng Timer1.**

Các bước lập trình chế độ 1

- 1. Lập trình tránh bảng vector ngắt.**
- 2. Thiết lập TMOD và nạp TL, TH.**
- 3. Thiết lập chế độ hỗ trợ nếu cần.**
- 4. Thiết lập thanh ghi IE.**
- 5. Khởi tạo bộ đếm.**
- 6. Tạo vòng lặp xử lý công việc và đợi ngắt.**
- 7. Lập trình tại địa chỉ ngắt tạo xung và nạp lại TH, TL.**

Đặc điểm lập trình Timer dùng ngắt

- 1. Thiết lập chế độ làm việc.**
 - **TMOD=01 (hoặc 02)**
 - **Nạp TLx, THx**
- 2. Thiết lập thanh ghi IE cho phép ngắt**
 - **IE = 82H với Timer0**
 - **IE = 88H với Timer1**
- 3. Lập trình tại địa chỉ ngắt tương ứng**
 - **0BH với Timer0**
 - **1BH với Timer1**

Giải:

**Số đếm $N=1085\mu s/1,085=1000>255$
nên cần dùng Mode1.**

```
ORG 0H  
LJMP MAIN
```

```
ORG 1BH           ;ISR Timer1  
LJMP ISR_T1
```

ORG 30H

MAIN:

MOV P0,#0FFH

MOV TMOD,#10H ;Timer1, Mode1

MOV TL1,#018H ;-1000D=FC18H

MOV TH1,#0FCH

MOV IE,#88H ;Cho phép ngắt

Timer1

SETB TR1 ;Khởi động Timer 1

BACK:

MOV A,P0

MOV P1,A

SJMP BACK

ISR_T1:

CLR TR1

CLR P2.1

MOV R2,#4 ;2 MC

HERE:

DJNZ R2,HERE ;8MC=4×2 MC

MOV TL1,#18H ;2 MC

MOV TH1,#0FCH ;2 MC

SETB TR1

SETB P2.1

RETI

END

3. Lập trình ngắt ngoài

- a) Giới thiệu**
- b) Lập trình ngắt kích phát mức**
- c) Lập trình ngắt kích phát sườn**
- d) Thanh ghi TCON**

a) Giới thiệu

- **8051 có 2 ngắt ngoài:**
 - **$\overline{\text{INT0}}$ - chân P3.2 Địa chỉ 3H**
 - **$\overline{\text{INT1}}$ - chân P3.3 Địa chỉ 13H**
- **Có hai cách kích hoạt ngắt ngoài:**
 - **Theo mức**
 - **Theo sườn**

b) Lập trình ngắt kích phát mức

- Ngắt kích phát mức là chế độ mặc định khi Reset VĐK.
- Các chân $\overline{INT0}$ và $\overline{INT1}$ bình thường ở mức cao.
- Ngắt được kích hoạt khi đặt xung mức thấp.
- Trước lệnh RETI thì chân \overline{INT} phải ở mức cao, nếu không VĐK hiểu là có ngắt mới kích hoạt.

Đặc điểm lập trình ngắt kích phát mức

- **Thiết lập thanh ghi cho phép ngắt IE**
 - **IE = 81H với INT0**
 - **IE = 84H với INT1**
- **Lập trình tại địa chỉ ngắt tương ứng**
 - **03H cho INT0**
 - **13H cho INT1**
- **Kết thúc ISR của ngắt là lệnh RETI**

Thanh ghi cho phép ngắt IE

D7	D6	D5	D4	D3	D2	D1	D0
EA	--	ET2	ES	ET1	EX1	ET0	EX0

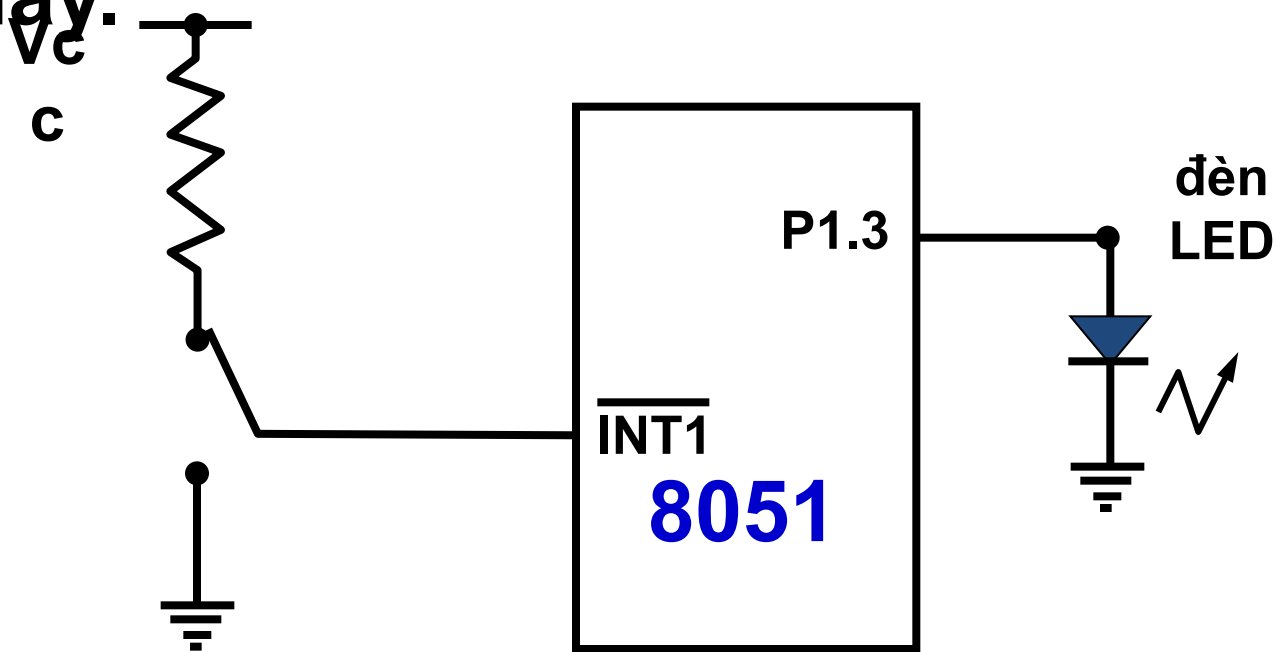
- EA IE.7** Nếu EA=0 thì che tất cả các ngắt.
Nếu EA=1 thì ngắt được mở/cấm bằng cách đặt/xoá bit tương ứng.
- ET2 IE.5** Cho_phép/cấm ngắt tràn Timer2 (8952)
- ES IE.4** Cho_phép/cấm ngắt cổng nối tiếp
- ET1 IE.3** Cho_phép/cấm ngắt tràn Timer1
- EX1 IE.2** Cho_phép/cấm ngắt INT1
- ET0 IE.1** Cho_phép/cấm ngắt tràn Timer0
- EX0 IE.0** Cho_phép/cấm ngắt INT0

Các bước lập trình ngắt kích phát mức

- 1. Lập trình tránh bảng vectơ ngắt.**
- 2. Thiết lập thanh ghi IE.**
- 3. Tạo vòng lặp đợi ngắt.**
- 4. Lập trình tại địa chỉ ngắt.**

Ví dụ 5:

- Cho chân INT1 được nối đến công tắc bình thường ở mức cao.
- Đèn LED nối đến chân P1.3 và bình thường ở chế độ tắt.
- Hãy lập trình để mỗi khi nhấn công tắc xuống thấp thì đèn LED sáng trong vài phần giây.



Các bước lập trình ngắt kích phát mức

- 1. Lập trình tránh bảng vectơ ngắt.**
- 2. Thiết lập thanh ghi IE.**
- 3. Tạo vòng lặp đợi ngắt.**
- 4. Lập trình tại địa chỉ ngắt.**

Giải:

```
ORG 0H  
LJMP      MAIN
```

```
ORG 13H  
SETB      P1.3  
MOV R3,#255  
BACK:     DJNZ R3,BACK  
CLR P1.3  
RETI
```

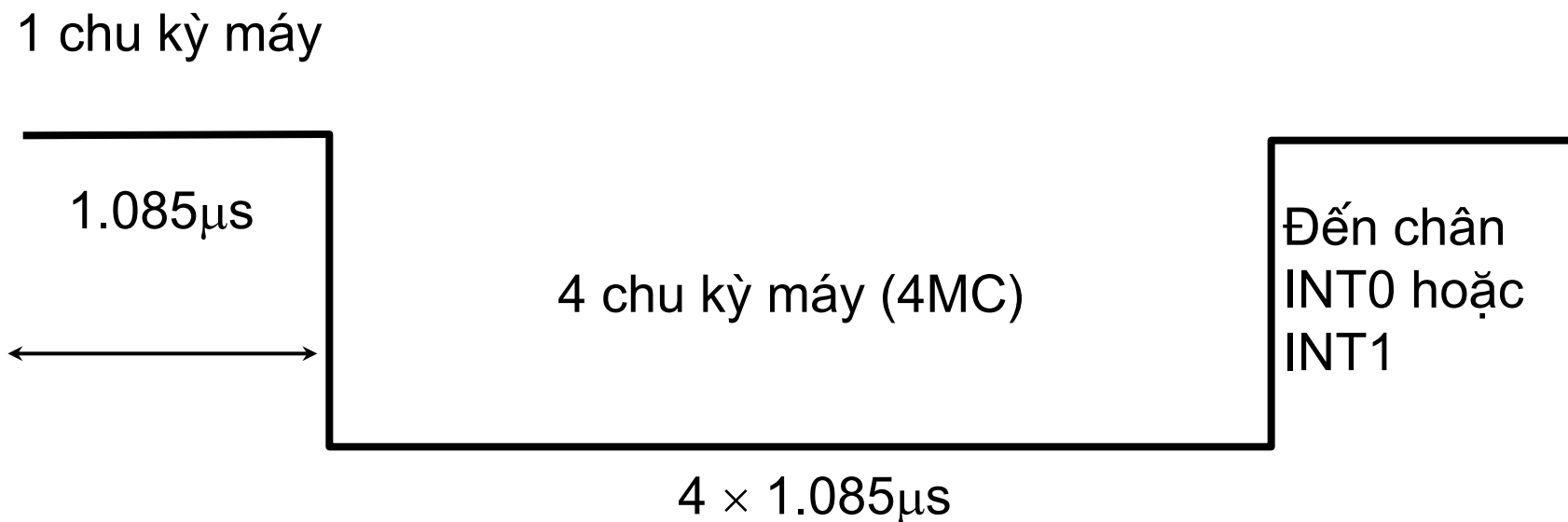
```
ORG 30H  
MAIN:  
MOV IE,#84h  
HERE: SJMP HERE  
END
```

Thời gian kích hoạt ngắt kích phát mức

- **Khi ngắt được kích hoạt, VĐK lấy mẫu chân INTx trong 1 chu kỳ máy để xác định mức thấp.**
- **Chân INTx không được lên cao quá sớm trước khi VĐK nhảy đến ISR vì như vậy ISR sẽ không được thực hiện.**
- **Chân INTx cũng không thể ở mức thấp quá lâu, sau cả lệnh RETI vì như vậy thì coi như có một ngắt mới được kích hoạt.**

■ Tóm lại:

Để ngắt được kích hoạt, INTx cần ở mức thấp 4 chu kỳ máy và không được lâu hơn.



Ghi chú: Khi Reset thì cả IT0 (TCON.0) và IT1 (TCON.2) đều ở mức thấp và ngắt ngoài được kích phát mức

Hình 11.5. Thời gian tối thiểu của ngắt kích phát mức thấp ($F_{xtal} = 11,0592\text{MHz}$)

c) Lập trình ngắt kích phát sườn

- **Đặc điểm lập trình ngắt kích phát sườn**
- **Các bước lập trình ngắt kích phát sườn**
- **Thời gian kích phát sườn**

Đặc điểm lập trình ngắt kích phát sườn

- **Thiết lập IE cho phép ngắt:**
 - **IE=81H với INT0**
 - **IE=84H với INT1**
- **Thiết lập chế độ ngắt kích phát sườn:**
 - **SETB IT0 (hoặc TCON.0) với INT0**
 - **SETB IT1 (hoặc TCON.2) với INT1**
- **Lập trình tại địa chỉ tương ứng:**
 - **3H với INT0**
 - **13H với INT1**

TF1	TR1	TF0	TR0	IE1	IT1	IE0
IT0						

TF1	TCON.7	Cờ tràn Timer1, TF1=1 khi Timer1 tràn, TF1=0 khi VĐK nhảy đến ISR.
TR1	TCON.6	Bật/tắt Timer1
TF0	TCON.5	Như TF1 nhưng cho Timer0.
TR0	TCON.4	Bật/tắt Timer0.
IE1	TCON.3	Dùng cho INT1 kích phát sườn, IE1=1 khi có sườn xuống, IE1=0 khi ngắt được xử lý.
IT1	TCON.2	IT1=1 Kích phát bằng sườn xuống IT1=0 kích phát mức.
IE0	TCON.1	Như IE1 nhưng cho INT0.
IT0	TCON.0	Như IT1 nhưng cho INT0

Hình 11.6. Thanh ghi TCON (Timer/Counter)

Các bước lập trình ngắt kích phát sườn

- 1. Lập trình tránh bảng vectơ ngắt.**
- 2. Thiết lập kích phát sườn**
- 3. Thiết lập thanh ghi IE.**
- 4. Tạo vòng lặp đợi ngắt.**
- 5. Lập trình tại địa chỉ ngắt.**

Thanh ghi cho phép ngắt IE

D7	D6	D5	D4	D3	D2	D1	D0
EA	--	ET	ES	ET	EX	ET	EX
		2		1	1	0	0

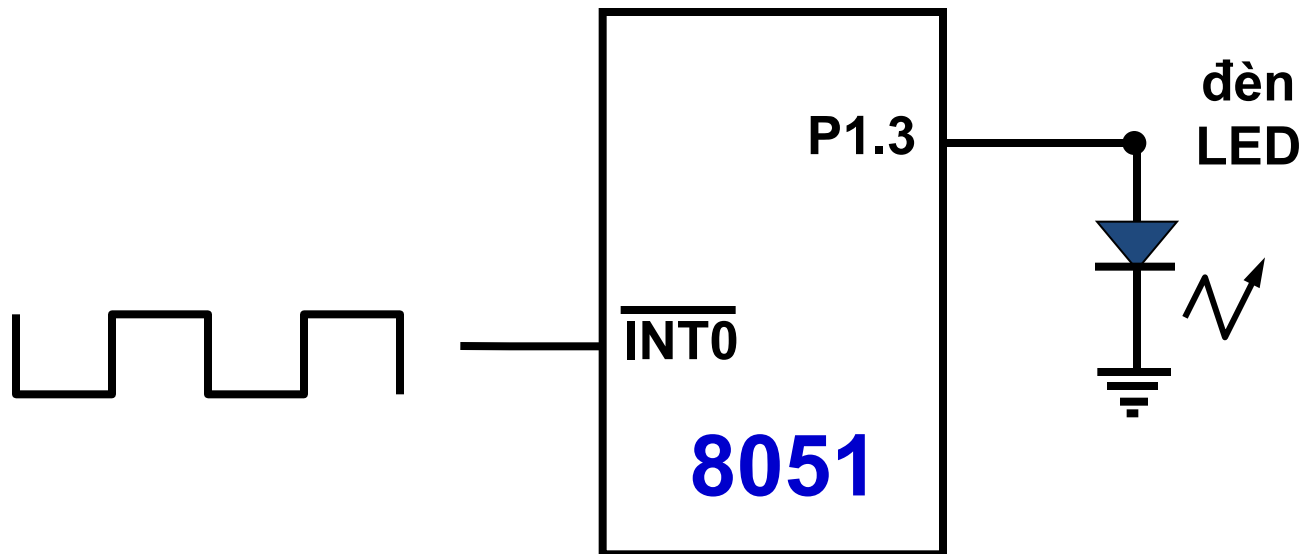
- EA IE.7** Nếu EA=0 thì che tất cả các ngắt.
Nếu EA=1 thì ngắt được mở/cấm bằng cách đặt/xoá bit tương ứng.
- ET2 IE.5** Cho_phép/cấm ngắt tràn Timer2 (8952)
- ES IE.4** Cho_phép/cấm ngắt cổng nối tiếp
- ET1 IE.3** Cho_phép/cấm ngắt tràn Timer1
- EX1 IE.2** Cho_phép/cấm ngắt INT1
- ET0 IE.1** Cho_phép/cấm ngắt tràn Timer0
- EX0 IE.0** Cho_phép/cấm ngắt INT0

Ví dụ 6:

Cho:

- INT0 (P3.2) nối với bộ tạo xung.
- P1.3 nối với đèn LED.

Viết chương trình bật sáng LED 1 lúc mỗi khi có sườn xuống (giả thiết tần số xung đủ chậm).



Giải:

```
ORG 0  
LJMP MAIN
```

```
ORG 03H  
SETB P1.3  
MOV R3,#225
```

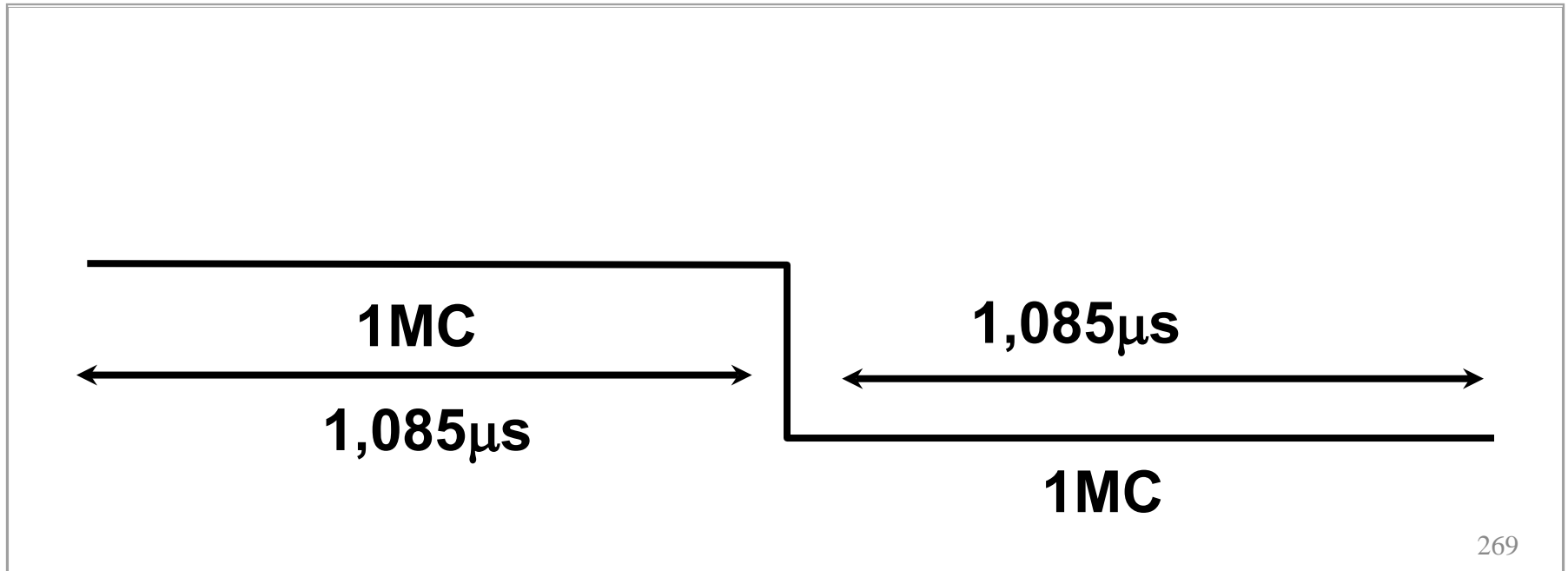
```
BACK: DJNZ R3,BACK  
CLR P1.3  
RETI
```

```
ORG 30H  
Main: SETB TCON.0  
MOV IE,#81H  
HERE: SJMP HERE  
END
```

Thời gian kích phát sườn

Để bộ VĐK nhận biết được quá trình chuyển sườn xung từ cao xuống thấp:

- Nguồn ngắt ngoài phải giữ ở mức cao tối thiểu 1 chu kỳ máy.
- Sau đó duy trì mức thấp cũng tối thiểu 1 chu kỳ máy.



d) Thanh ghi TCON

D7				D0			
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

TF1	TCON.7	Cờ tràn Timer1.
TR1	TCON.6	Bit điều khiển bật/tắt Timer1.
TF0	TCON.5	Tương tự nh TF1 nhưng cho Timer0.
TR0	TCON.4	Tương tự nh TR1 nhưng cho Timer0.
IE1	TCON.3	Cờ cho phép ngắt INT1 kích phát sườn,
IT1	TCON.2	Xác định kiểu kích phát sườn xuống (1)
IE0	TCON.1	Tương tự như IE1 nhưng cho INT0.
IT0	TCON.0	Tương tự như IT1 nhưng cho INT0

Hình 11.6. Thanh ghi TCON (Timer/Counter)

IT0 và IT1

- IT0 và IT1 (TCON.0 và TCON.2) được dùng để xác định kiểu ngắt kích phát sườn.
- Khi Reset, IT0=IT1=0, tức ngầm định thiết lập cho ngắt kích phát mức thấp.
- Đặt IT0=IT1=1 chuyển sang kích phát sườn.

Ví dụ 7:

Phân biệt lệnh RET và RETI? Có thể dùng RET thay cho RETI trong ISR được không?

Trả lời:

Giống:

- RET và RETI giống nhau về cách làm việc, đều lấy hai byte đỉnh ngăn xếp nạp vào bộ đếm chương trình để VĐK thực hiện.

Khác:

- Lệnh RETI còn có nhiệm vụ xóa cờ ngắt ($IE_x=0$) để báo ngắt đã kết thúc và VĐK có thể nhận 1 ngắt mới.
- Lệnh RET không có nhiệm vụ xóa cờ ngắt ($IE_x=1$) nên VĐK không thể nhận 1 ngắt mới.

- **Kết luận:**

- **Lệnh RET không thay RETI được.**
- **Để nhận ngắt kích phát sườn âm tiếp theo Lập trình viên không cần xóa cờ IEx mà lệnh RETI sẽ tự xóa.**

TR0 và TR1

- TR0 và TR1(hay TCON.4 và TCON.6)
- Được dùng để khởi động và dừng bộ định thời Timer0 và Timer1 tương ứng.
- Có thể sử dụng lệnh:
 - SETB TRx và CLR TRx hoặc
 - SETB TCON.4 và CLR TCON.4

TF0 và TF1

- **TF0 và TF1 là bit D5 (TCON.5) và D7 (TCON.7).**
- **Dùng để báo Timer0 và Timer1 bị tràn và quay về 0.**
- **Có thể dùng lệnh:**
 - **JNB TFx, đích và CLR TFx**
 - **SETB TCON.5, đích và CLR TCON.5**

Tóm tắt:

- Bộ vi điều khiển 8051 có 6 ngắt
- Người dùng có thể lập trình cho 5 ngắt là INT0, INT1, TF0, TF1 và TI+RI.
- Có thể thiết lập Cho_phép/Cấm_ngắt.
- Có thể lập trình ưu tiên ngắt.

A. Lập trình ngắt Timer

- **Timer0: IE = 82H, Địa chỉ 0BH**
- **Timer1: IE = 88H, Địa chỉ 1BH**

B. Lập trình ngắt ngoài kích phát mức

- **INT0: IE = 81H, Địa chỉ 03H**
- **INT1: IE = 84H, Địa chỉ 13H**

C. Lập trình ngắt ngoài kích phát s- uờn

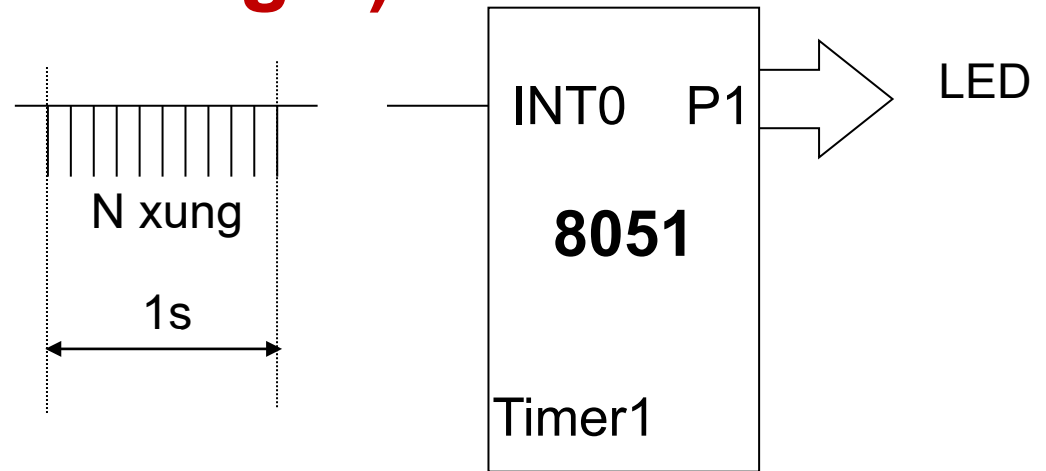
▪ INT0:

- SETB IT0
- IE = 81H, Địa chỉ 03H

▪ INT1:

- SETB IT1
- IE = 84H, Địa chỉ 13H

Ví dụ 10: (tổng hợp cho ngắt)



Lập trình đo tốc độ động cơ bằng cách liên tục đếm xung N từ encoder trong thời gian 1s (giả thiết $N < 256$) theo cách sau:

- Dãy xung N đưa đến chân **INT0** (P3.2) kích phát mức
- Dùng **Timer1** tạo thời gian 1s làm việc chế độ ngắt.
- Sau 1s số xung đếm được chuyển tới cổng **P1**.

(gợi ý: $1s = 30 \times 30722 \times 1,085\mu s$, $30722D = 7802H$).

Các bước lập trình

- Tránh bảng vectơ ngắt
- Nạp TMOD
- Nạp R1 lặp 1s cho Timer1
- Xóa R0 để đếm xung INT0
- Nạp TH1, TL1, khởi động Timer1
- Nạp IE khởi động INT0
- Tạo vòng lặp đợi 2 ngắt
- Lập trình tại địa chỉ ngắt INT0 đếm xung N
- Lập trình tại địa chỉ ngắt Timer1 đếm 1s

ORG 0
LJMP Main

ORG 30H

Main:

MOV TMOD, #10H
MOV R1, 30
:
CLR R0
MOV TH1, #87H
MOV TL1, #FEH
SETB TR1
MOV IE, #81H
LAP: SJMP LAP

ORG 3
INC R0
RETI

ORG 1BH
LJMP Tao_1s

ORG 100H

Tao_1s:

DJZ R3, Thoat
MOV TH1, #87H
MOV TL1, #FEH
RETI

Thoat:

MOV A, R0
MOV P1, A
CLR TR1
CLR TF1
LJMP Main

END

Mức ưu tiên từ cao xuống thấp

Ngắt	Cờ ngắt	Địa chỉ trình phục vụ ngắt	Số thứ tự ngắt
Ngắt ngoài 0	IE0	0003h	0
Timer 0	TF0	000Bh	1
Ngắt ngoài 1	IE1	0013h	2
Timer 1	TF1	001Bh	3
Ngắt truyền thông	RI/TI	0023h	4

D7

D0

--	--	PT2	PS	PT1	PX1	PT0	PX0
----	----	-----	----	-----	-----	-----	-----

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 -- chưa dùng.
- Bit D5 hay **PT2** là Bit ưu tiên ngắt Timer2 (dùng cho 8052)
- Bit D4 hay **PS** là Bit ưu tiên ngắt cổng nối tiếp
- Bit D3 hay **PT1** là Bit ưu tiên ngắt Timer1
- Bit D2 hay **PX1** là mức ưu tiên ngắt ngoài 1
- Bit D1 hay **PT0** là mức ưu tiên ngắt Timer 0
- Bit D0 hay **PX0** là mức ưu tiên ngắt ngoài 0

Truyền thông nối tiếp

Nội dung

- 1. Cơ sở của truyền thông nối tiếp**
- 2. Chuẩn RS232 và nối ghép**
- 3. Lập trình truyền thông nối tiếp bằng phương pháp thăm dò**
- 4. Lập trình truyền thông nối tiếp bằng phương pháp ngắt**

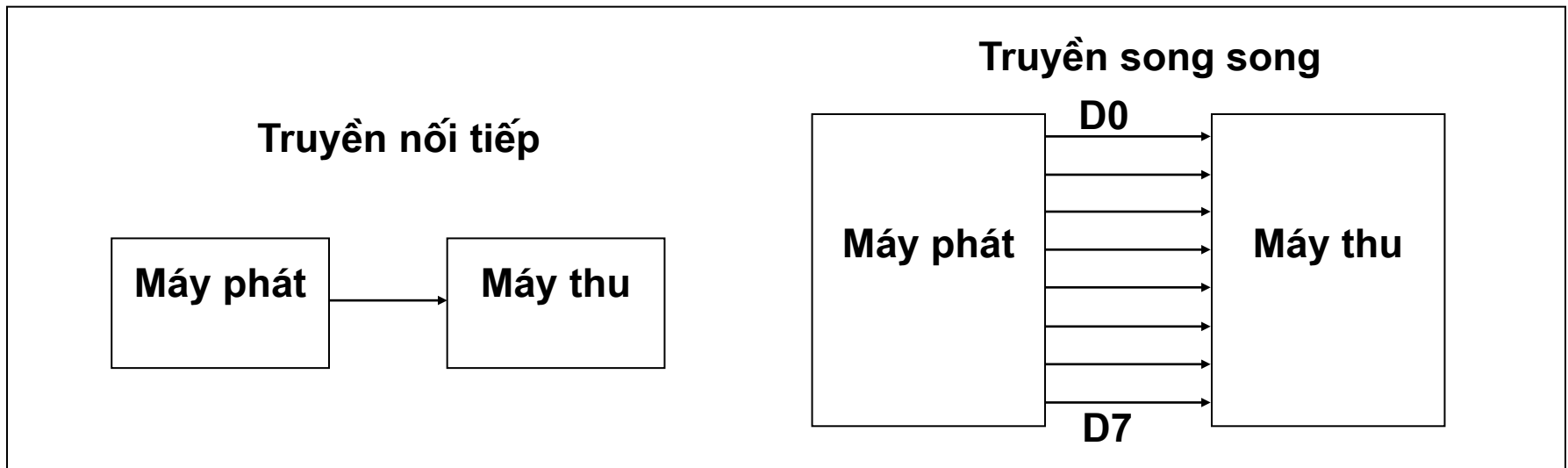
1. Cơ sở truyền thông nối tiếp

- a) Truyền nối tiếp và truyền song song**
- b) Truyền đồng bộ và dị bộ**
- c) Các chế độ thu phát dữ liệu**
- d) Định khung dữ liệu**
- e) Tốc độ truyền dữ liệu**
- f) Phân loại thiết bị truyền dữ liệu**

a) Truyền nối tiếp và truyền song song

Có 2 phương pháp truyền: Nối tiếp và Song song.

- **Truyền song song:** Truyền nhiều bit đồng thời, cự ly truyền hạn chế.
- **Truyền nối tiếp:** Truyền qua 1 đường dây dữ liệu, có thể truyền dữ liệu đi xa.



Hình 10.1. Truyền dữ liệu song song và nối tiếp

Thanh ghi dịch

- ở đầu phát, để truyền nối tiếp, byte dữ liệu được chuyển thành các bit nối tiếp nhờ thanh ghi dịch **vào-song song-ra-nối tiếp**.
- Dữ liệu sau đó được truyền qua một đường truyền.
- ở đầu thu cũng có một thanh ghi dịch **vào-nối tiếp-ra-song song** gói chúng thành từng byte một.

Modem

- Nếu dữ liệu được truyền qua đường thoại thì cần chuyển các số 0 và 1 sang âm thanh dạng sóng hình sin.
- Việc chuyển đổi này do thiết bị gọi là Bộ điều chế/giải điều chế **Modem** (Modulator/Demodulator) thực hiện.
- Khi cự ly truyền ngắn không cần điều chế.
- Để truyền dữ liệu đi xa yêu cầu một Modem điều chế.

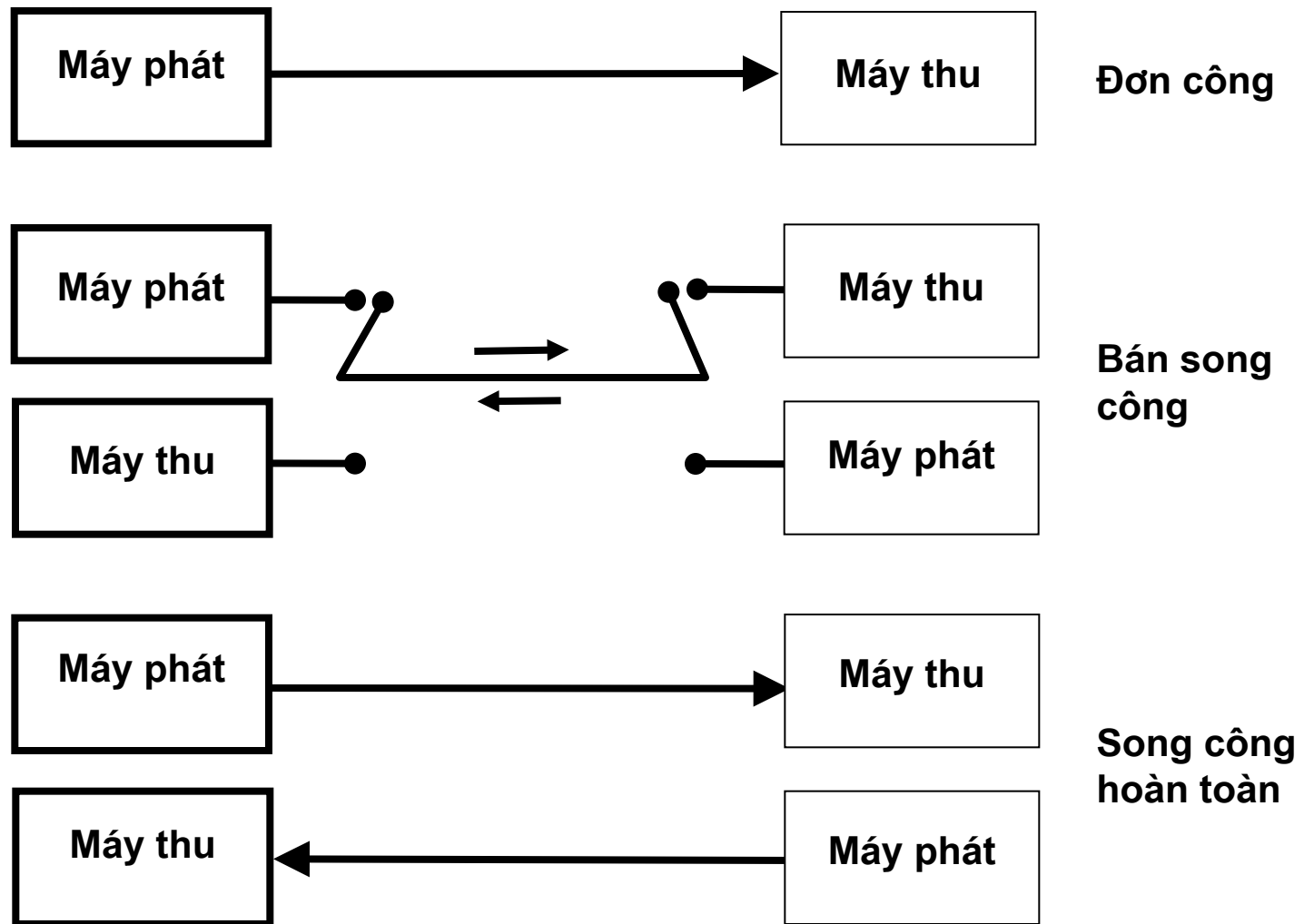
b) Phương pháp Đồng bộ, Dị bộ

Có 2 phương pháp truyền nối tiếp:

- Phương pháp **Đồng bộ** mỗi lần truyền một khối dữ liệu.
- Phương pháp **Dị bộ** chỉ truyền từng byte.
- Các hãng IC sản xuất các bộ:
 - Thu-phát dị bộ tổng hợp UART (Universal Asynchronous Receiver Transmitter) và
 - Thu-phát đồng-dị bộ tổng hợp USART (Universal Synchronous-Asynchronous Receiver Transmitter).

c) Các chế độ thu phát dữ liệu

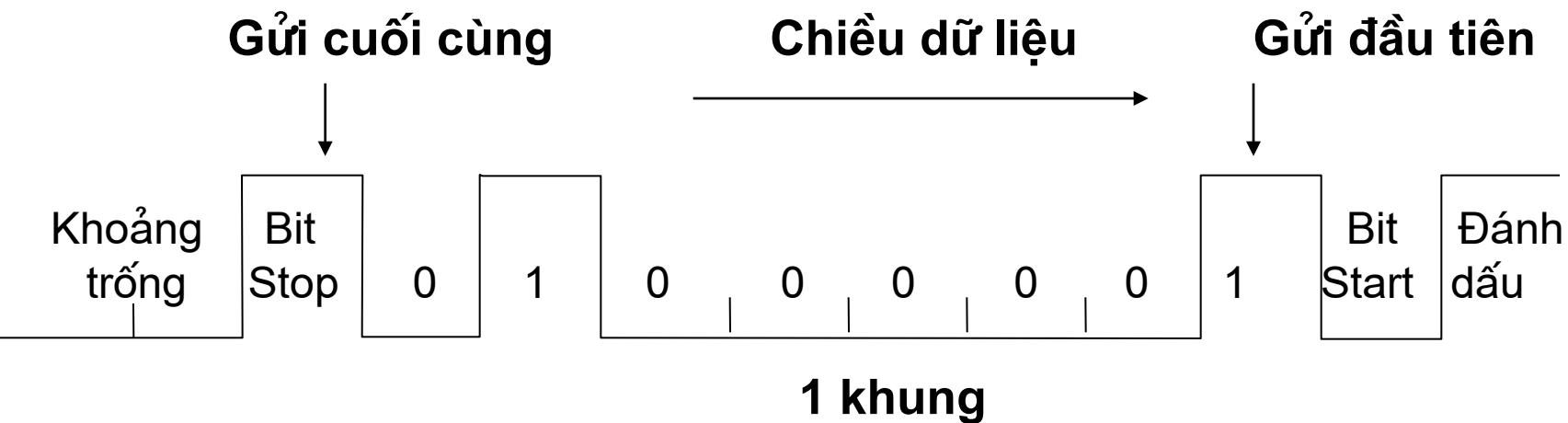
- Truyền **Đơn công**: dữ liệu chỉ được truyền theo 1 chiều.
- Truyền **Song công**: dữ liệu vừa có thể phát vừa có thể thu.
 - **Bán song công**: Nếu tại mỗi thời điểm, dữ liệu chỉ truyền một chiều.
 - **Song công hoàn toàn** nếu dữ liệu truyền được cả hai chiều đồng thời.



Hình 10.2. Các chế độ thu phát dữ liệu

d) Định khung dữ liệu truyền dị bộ

- Định khung dữ liệu: mỗi ký tự được bố trí giữa bit khởi động và bit dừng.
- Bit khởi động luôn chỉ có một bit và có giá trị thấp
- Bit dừng có thể có 1 hoặc 2 bit, có giá trị cao.
- Giao thức truyền (Protocol): qui tắc tổ chức dữ liệu và kết nối.



Định khung ký tự A, m□ASCII 41H

e) Tốc độ truyền dữ liệu

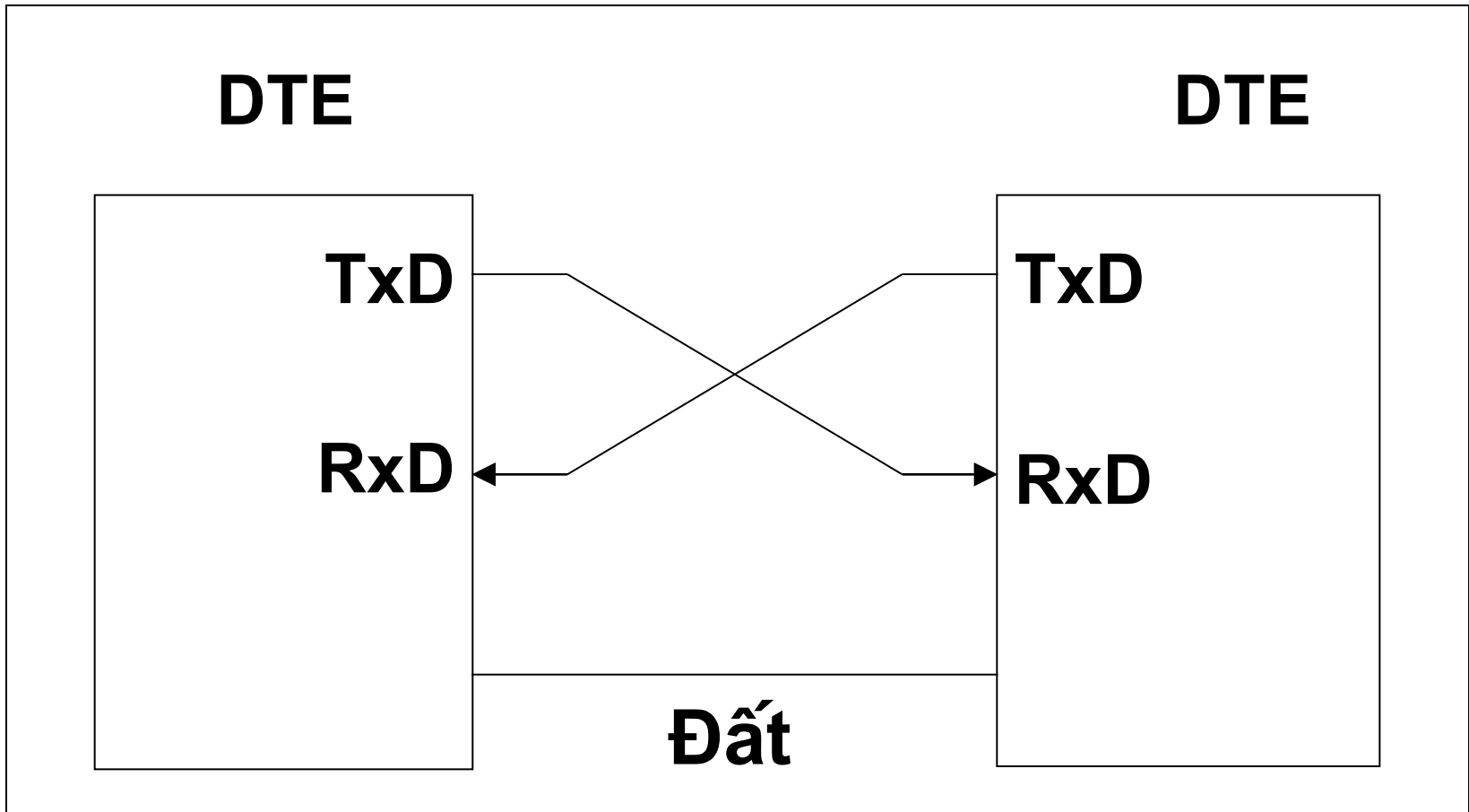
Có 2 khái niệm:

- **Bps** (Bit Per Second- bit/giây) là đơn vị tốc độ truyền nối tiếp.
- **Baud** là đơn vị đo dùng cho modem, là số lần thay đổi tín hiệu trong một giây.
- Đối với đường truyền thì tốc độ baud và bps là một.
- Truyền dữ liệu nối tiếp không đồng bộ thường bị hạn chế dưới 100.000 bps.

f) Phân loại thiết bị truyền dữ liệu

Thiết bị truyền dữ liệu được phân biệt thành:

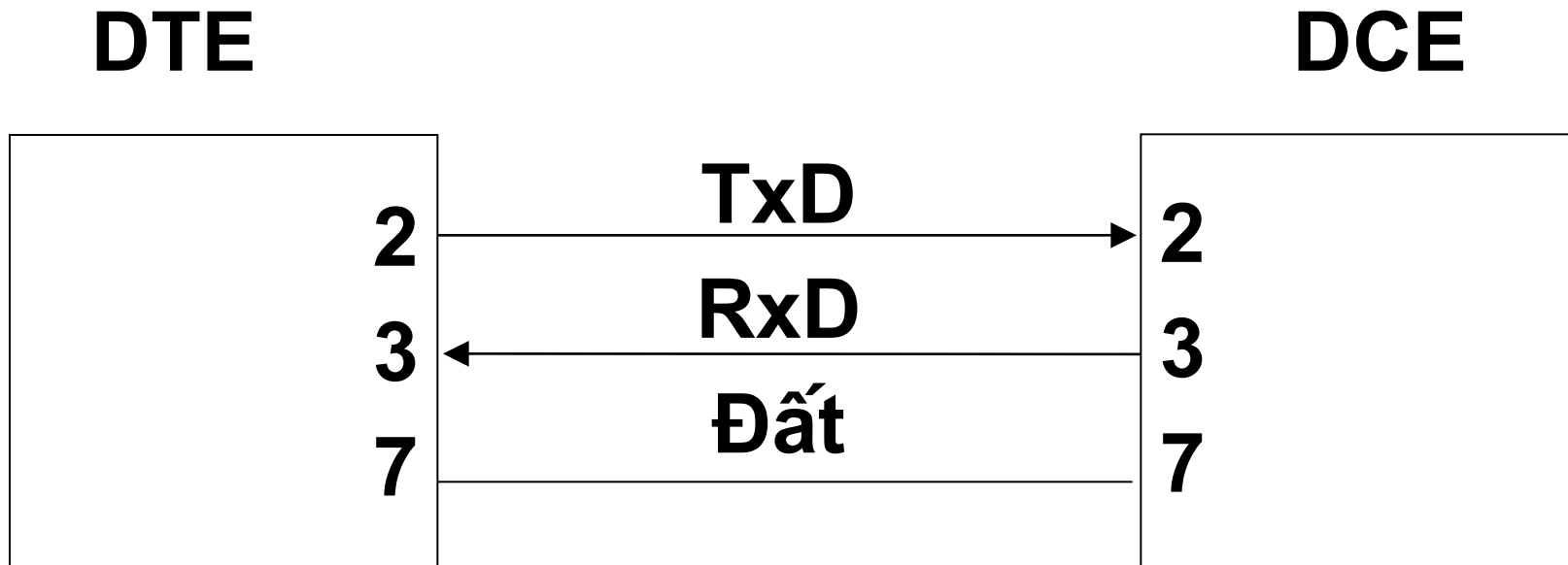
- **Thiết bị đầu cuối DTE (Data Terminal Equipment).**
 - DTE có thể² là máy tính và các thiết bị đầu cuối gửi và nhận dữ liệu.
- **Thiết bị truyền thông DCE (Data Communication Equipment).**
 - DCE có thể² là các modem.



Hình 10.6. Ghép nối DTE-DTE

DTE -Data Terminal Equipment

Ghép nối DTE-DCE



Hình 9.7. Ghép nối DTE-DCE

DTE -Data Terminal Equipment

DCE -Data Communication Equipment

2. Chuẩn RS232 và nối ghép với 8051

a) Chuẩn RS232

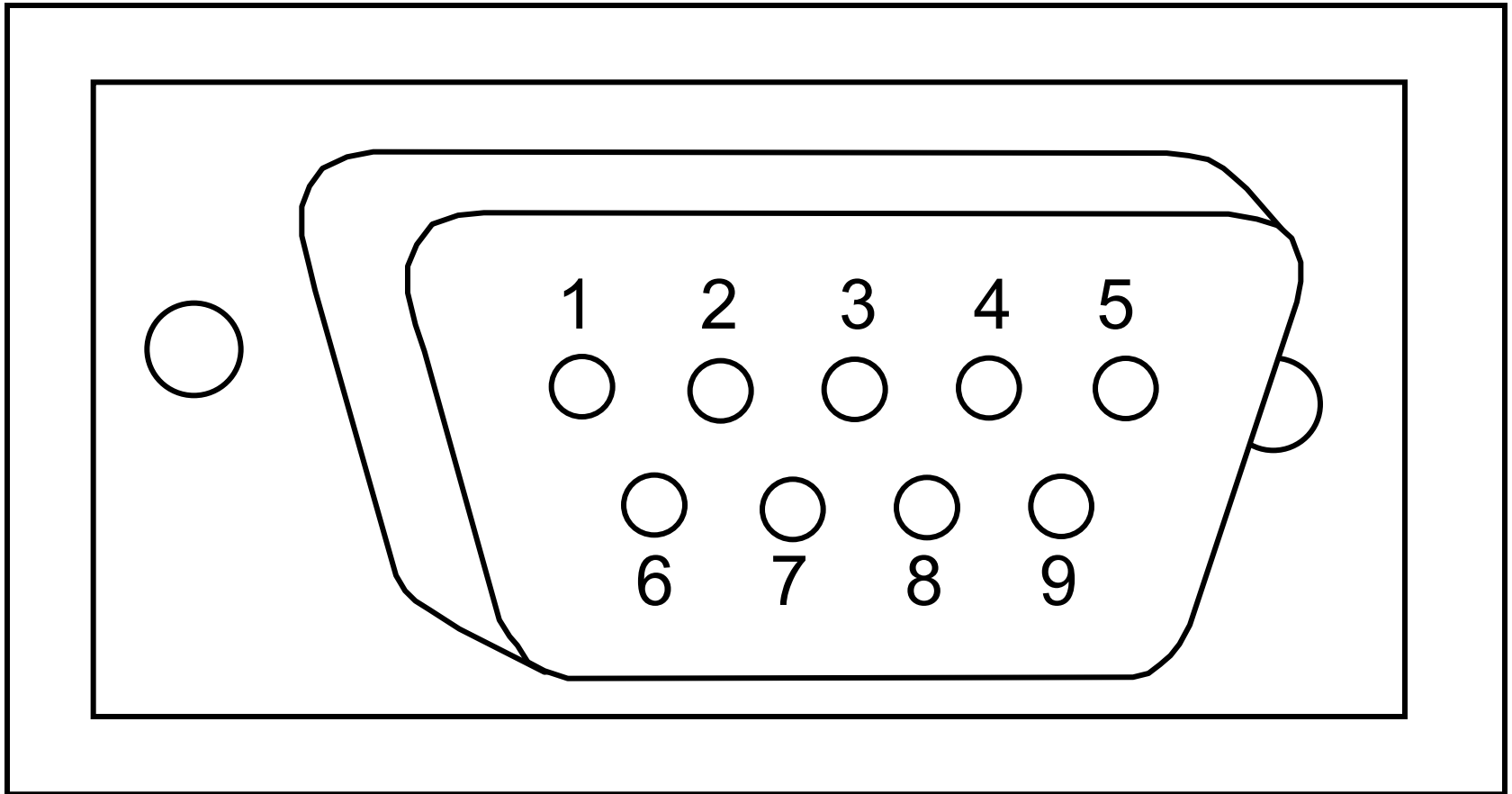
b) Nối ghép RS232 với 8051

a) Chuẩn RS232

- **Chuẩn RS232 được Hiệp hội Công nghiệp Điện tử EIA xây dựng năm 1960.**
- **Chuẩn này được cải tiến thành RS232A (1963), RS232B (1965) và RS232C (1969).**
- **Ngày nay, RS232 là chuẩn giao diện I/O được sử dụng rất rộng rãi.**

Mức điện áp chuẩn RS232

- Các mức áp vào/ra của RS232 không tương thích với TTL (Do chuẩn này ra đời d□ lâu)
- ở RS232:
 - Mức 1 là từ -3V đến -25V
 - Mức 0 là từ +3V đến +25V
 - Khoảng từ -3V đến +3V không xác định.
- Để kết nối với máy tính cần có bộ biến đổi áp như MAX232 chuyển mức TTL sang RS232 và ngược lại.
- Các chip IC MAX232 được dùng để điều khiển đường truyền



Hình 10.5. Sơ đồ đầu nối DB - 9 của RS232

Bảng 10.2. Bố trí chân DB-9 ở máy tính IBM PC

Chân	Mô tả	ý nghĩa
1	Data carrier detect (DCD)	Tách tín hiệu mang dữ liệu
2	Received data (RxD)	Dữ liệu thu
3	Transmitted data (TxD)	Dữ liệu truyền
4	Data terminal ready (DTR)	DTE sẵn sàng
5	Signal ground (GND)	Đất của tín hiệu
6	Data set ready (DSR)	Báo DCE sẵn sàng
7	Request to send (RTS)	Yêu cầu gửi
8	Clear to send (CTS)	Xoá để gửi
9	Ring indicator (RL)	Báo chuông

Các tín hiệu bắt tay truyền dữ liệu nối tiếp

- **DTR** (Data Terminal Ready) - Thiết bị DTE gửi tín hiệu DTR đến DCE để báo DTE ở trạng thái sẵn sàng.
- **DSR** (Data Set Ready) - DCE phát tín hiệu DSR để báo DCE ở trạng thái sẵn sàng.

- **RTS** (Request To Send) - DTE gửi tín hiệu RTS để báo cho DCE biết có một byte cần gửi.
- **CTS** (Clear To Send) - DCE báo DTE biết DCE sẵn sàng nhận để bắt đầu truyền dữ liệu.

- **CD** (Carrier Detect) - DCE gửi tín hiệu CD để báo cho DTE rằng đã phát hiện được sóng mang và kết nối giữa DTE với Modem đã được thiết lập.

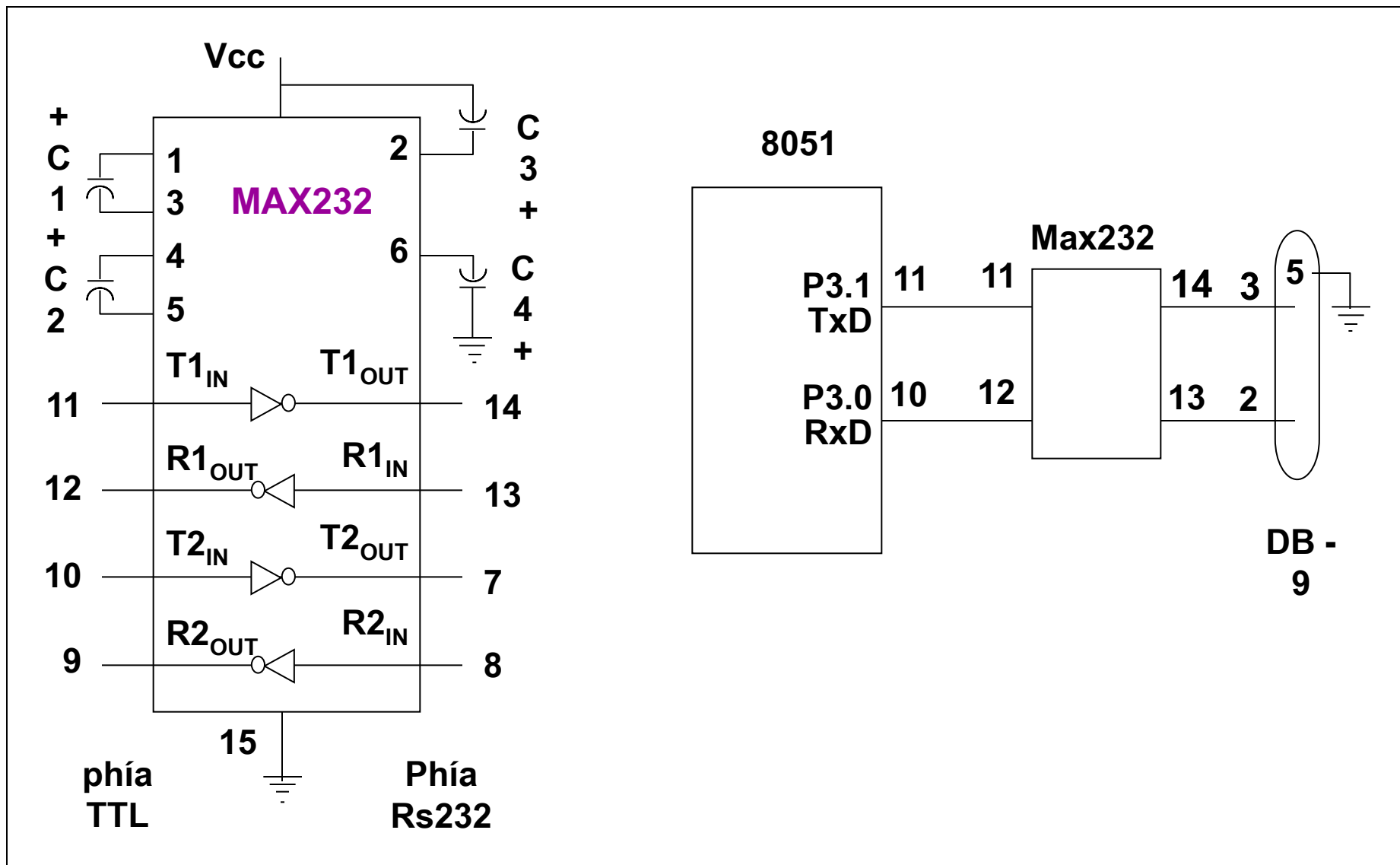
- **RI** (Ring Indicator) - DCE gửi RI tới DTE để báo chuông đang reo. Tín hiệu này tắt hoặc mở đồng bộ với tiếng chuông. Là tín hiệu ít dùng nhất trong 6 tín hiệu bắt tay.

Tóm tắt quá trình truyền tin PC-Modem

- DTR và DSR của máy tính và modem báo hiệu máy tính và modem đang làm việc tốt và ở trạng thái sẵn sàng.
- Tín hiệu RTS và CTS điều khiển luồng dữ liệu.
 - Khi máy tính muốn gửi dữ liệu thì đầu tiên gửi tín hiệu RTS cho modem
 - Nếu modem đã sẵn sàng (có chỗ chứa dữ liệu) thì modem gửi lại cho máy tính tín hiệu CTS.

b) Nối ghép RS232 với 8051

- 8051 dùng 2 chân cho thu-phát nối tiếp là RxD và TxD ở cổng P3.
- Để các chân này tương thích với RS232 cần có thêm một bộ điều khiển đường truyền .
- Bộ điều khiển như vậy có thể là chip MAX232.



**Hình 10.7. a) Sơ đồ cấu trúc của MAX232
b) Sơ đồ nối ghép MAX232 với 8051**

Bộ điều khiển đường truyền MAX232

- MAX232 chuyển đổi giữa các mức điện áp RS232 và TTL.
- Một điểm mạnh của chip MAX232 là dùng nguồn +5v, cùng điện áp nguồn của 8051.
- MAX232 có hai bộ điều khiển đường truyền là thu và phát dữ liệu.

- Điều khiển phát cho TxD là T1 và T2.
- Nhiều ứng dụng chỉ dùng cặp T1/R1 ứng với TxD/RxD của 8051. Cặp R2/T2 không dùng.
- T1 của MAX232 có T1_{in} (11) và T1_{out} (14).
- Chân T1_{in} là ở phía TTL và được nối tới chân TxD của bộ vi điều khiển.
- T1_{out} là ở phía RS232 được nối tới chân TxD của đầu nối DB của RS232.

- Chân R1 có $R1_{in}$ (13) và $R1_{out}$ (12).
- $R1_{in}$ (chân 13) ở phía RS232 được nối tới chân TxD đầu nối DB của RS232 và chân $R1_{out}$ (chân 12) ở phía TTL được nối tới chân RxD của bộ VĐK.
- MAX232 cần thêm 4 tụ điện có trị thông dụng là $22\mu F$.

Bộ điều khiển MAX233

- Để tiết kiệm không gian trên bảng mạch, nhiều nhà thiết kế sử dụng chip MAX233.
- MAX233 có các chức năng như MAX232 song không cần các tụ điện.
- MAX233 đắt hơn đáng kể so với MAX232 và không tương thích với MAX232 (bố trí chân khác nhau).

3. Lập trình truyền thông nối tiếp bằng phương pháp thăm dò

- a) Tốc độ baud cổng COM máy tính và VĐK**
- b) Thanh ghi SBUF**
- c) Thanh ghi SCON**
- d) Phát dữ liệu nối tiếp bằng thăm dò**
- e) Thu dữ liệu nối tiếp bằng thăm dò**

a) Tốc độ baud cổng COM máy tính

100 Baud

150 Baud

300 Baud

600 Baud

1200 Baud

2400 Baud

4800 Baud

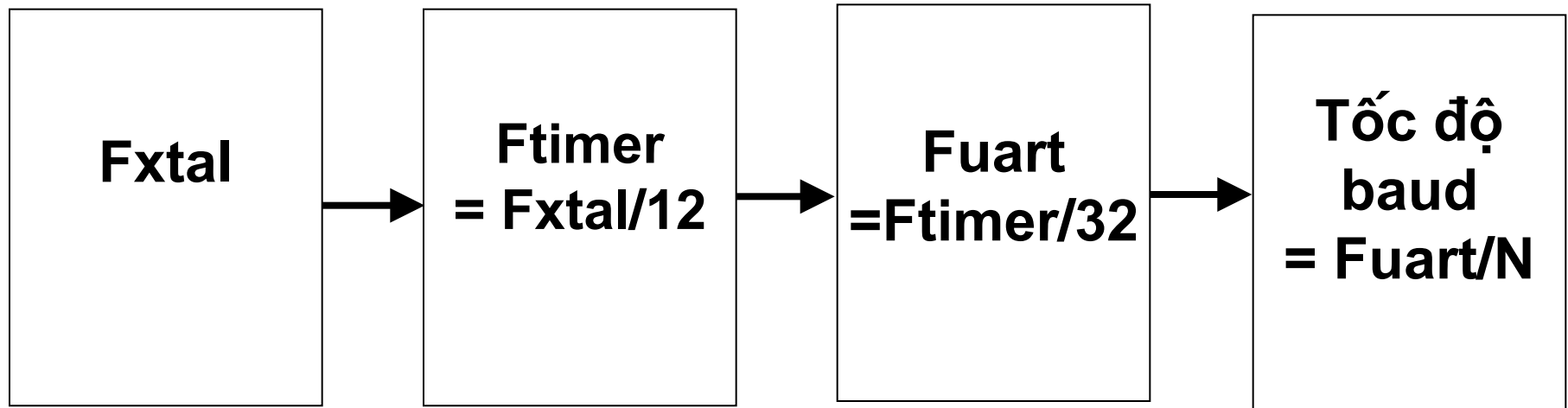
9600 Baud

19200 Baud

- Để thực hiện truyền thông giữa PC và VĐK thì tốc độ baud của VĐK phải phù hợp với tốc độ baud của cổng COM máy tính PC.

Tốc độ truyền dữ liệu cổng COM của 8051

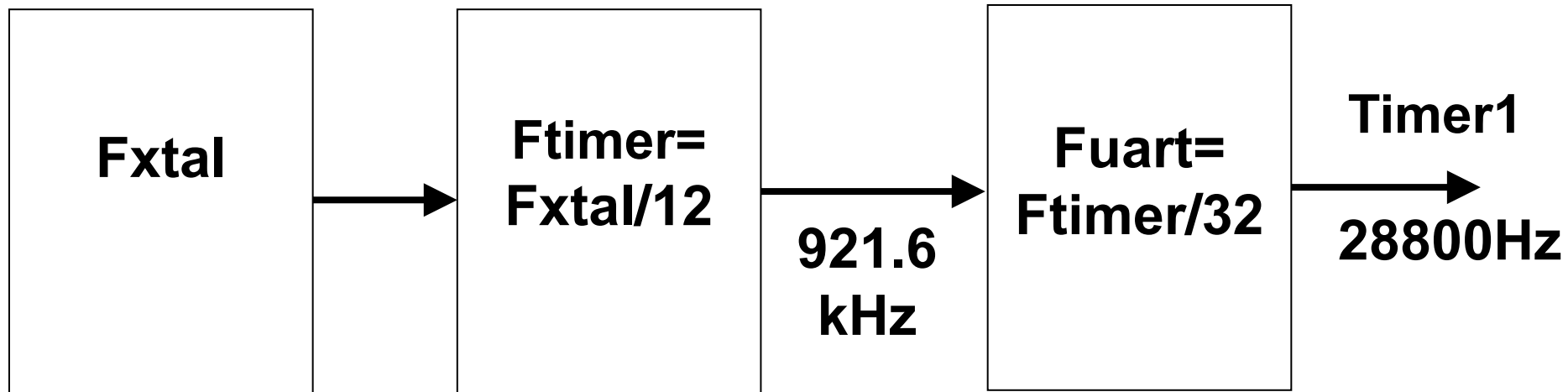
- $F_{\text{timer}} = F_{\text{xtal}}/12$
- $F_{\text{uart}} = F_{\text{timer}}/32$
- Tốc độ Baud của 8051 = F_{uart}/N



Ví dụ 1:

Cho tần số $F_{xtal}=11,0592\text{MHz}$.
Xác định giá trị TH1 để có tốc độ baud là:

- a) 9600; b) 2400; c) 1200.



Bảng 10.4. Giá trị TH1 của Timer1 ứng với các tốc độ baud

Tốc độ baud	TH1 (thập phân)	TH1 (số hexa)
9600	-3	FD
4800	-6	FA
2400	-12	F4
1200	-24	E8

b) Thanh ghi SBUF

- **SBUF là thanh ghi 8 bit dùng cho thu phát nối tiếp của 8051.**
- **Khi phát qua đường TxD, cần chuyển byte dữ liệu vào thanh ghi SBUF.**
- **Khi thu qua đường RxD, SBUF lưu byte dữ liệu thu được.**
- **SBUF có thể được mọi thanh ghi của 8051 truy cập.**

Ví dụ 2:

Viết lệnh:

- a) Gửi ký tự “D” vào thanh ghi SBUF**
- b) Gửi 8 bit thanh ghi A vào SBUF**
- c) Lấy nội dung SBUF gửi vào thanh ghi A**

Giải:

MOV SBUF,#“D”

MOV SBUF,A

MOV A,SBUF

Định khung dữ liệu khi thu, phát

- Khi phát, 8 bit dữ liệu gửi vào thanh ghi SBUF. Byte này sẽ được phát nối tiếp qua chân TxD và định khung với bit Start, Stop.
- Khi thu, khung dữ liệu gồm bit Start, 8 bit dữ liệu, bit Stop được VĐK chuyển qua chân RxD rồi mở khung, tức bỏ bit Start và Stop lấy dữ liệu và chuyển vào SBUF.
- Cách định khung do thanh ghi SCON xác định

c) Thanh ghi điều khiển nối tiếp SCON

SCON là thanh ghi 8 bit được dùng cho một số công việc, như định khung dữ liệu, xác định bit Start, số bit Stop, số bit dữ liệu.

SM0	SM1	SM2	REN	TB8	RB8	TI	RI
-----	-----	-----	-----	-----	-----	----	----

SM0	SCON.7	Xác định mode làm việc cổng nối tiếp
SM1	SCON.6	Xác định mode làm việc cổng nối tiếp
SM2	SCON.5	Truyền thông giữa các bộ VXL (SM2=0)
REN	SCON.4	Cho_phép/cấm thu (thiết lập=phần mềm)
TB8	SCON.3	Không sử dụng rộng rãi
RB8	SCON.2	Không sử dụng rộng rãi
TI	SCON.1	Cờ ngắt phát. Đặt bằng phần cứng khi bắt đầu bit Stop ở Mode 1. Xoá bằng phần mềm
RI	SCON.0	Cờ ngắt thu. Đặt bằng phần cứng khi bắt đầu bit Stop ở Mode 1. Xoá bằng phần mềm.

**Hình A.7. Thanh ghi điều khiển nối tiếp SCON
(định địa chỉ bit)**

SM0, SM1

Xác định chế độ định khung dữ liệu.

SM0	SM1	Mode	Ghi chú	Tốc độ
0	0	0	8bit đồng bộ	
Fxtal/12				
0	1	1	8bit UART	Do Timer1
1	0	2	9bit UART	Fxtal/64
1	1	3	9bit UART	Do Timer1

Chế độ 1: Định khung dữ liệu 8 bit, 1 bit Start và 1 bit Stop. Tốc độ truyền thay đổi được dùng Timer1.

- **Chế độ 0:** Là chế độ đồng bộ duy nhất. Chân RxD dùng thu phát dữ liệu. Chân TxD làm tín hiệu xung nhịp. Tốc độ truyền cố định và bằng $1/12$ tần số thạch anh.
- **Chế độ 1:** Là chế độ truyền dị bộ 8 bit. Định khung dữ liệu 8 bit, 1 bit Start và 1 bit Stop. Tốc độ truyền thay đổi được nhờ Timer1.

- **Chế độ 2:** Là chế độ truyền nhị bộ 9 bit, gồm 8 bit trong thanh ghi SBUF và bit RB8 (thu) hoặc TB8 (phát) trong thanh ghi SCON. Tốc độ truyền cố định và bằng $1/32$ hoặc $1/64$ tần số thạch anh (phụ thuộc vào giá trị SMOD của thanh ghi PCON).
- **Chế độ 3:** Là chế độ truyền nhị bộ 9 bit và khác với chế độ 2 là tốc độ truyền thay đổi được.

SM2

Dùng trong hệ đa xử lý của 8051.

REN: Receive Enable - SCON.4

- REN=1 cho phép thu dữ liệu (qua RxD)
- REN=0 khóa bộ thu.
- Có thể dùng lệnh “SETB SCON.4” và “CLR SCON.4”.

TB8 và RB8

- **TB8 (Transfer Bit 8)** bit dữ liệu phát thứ 9, dùng cho chế độ 2 và 3.
- **RB8 (Receive Bit 8)** bit dữ liệu thu thứ 9 ở chế độ 2, 3.
- Tài liệu không dùng TB8, RB8 (TB8=RB8=0)

TI và RI - Transmit Interrupt và Receive Interrupt.

- **Bình thường TI=0.**
- **Khi 8051 kết thúc phát 1 ký tự 8 bit thì cờ TI=1 để báo bộ VĐK sẵn sàng phát byte tiếp theo.**

- Khi 8051 nhận dữ liệu nối tiếp qua chân RxD thì tiến hành lấy ra 8 bit dữ liệu (bỏ bit Start và Stop) và chuyển vào thanh ghi SBUF.
- Khi quá trình này hoàn tất, cờ RI=1 để báo VĐK đã nhận xong 1 byte và cần cất đi nếu không sẽ bị ghi đè.

Giá trị SCON thường dùng

SM0	SM1	SM2	REN	TB8	RB8	TI	RI
0	1	0	1	0	0	0	0

SCON=0101.0000=50H

d) Phát dữ liệu nối tiếp bằng thăm dò

Đặc điểm lập trình phát nối tiếp

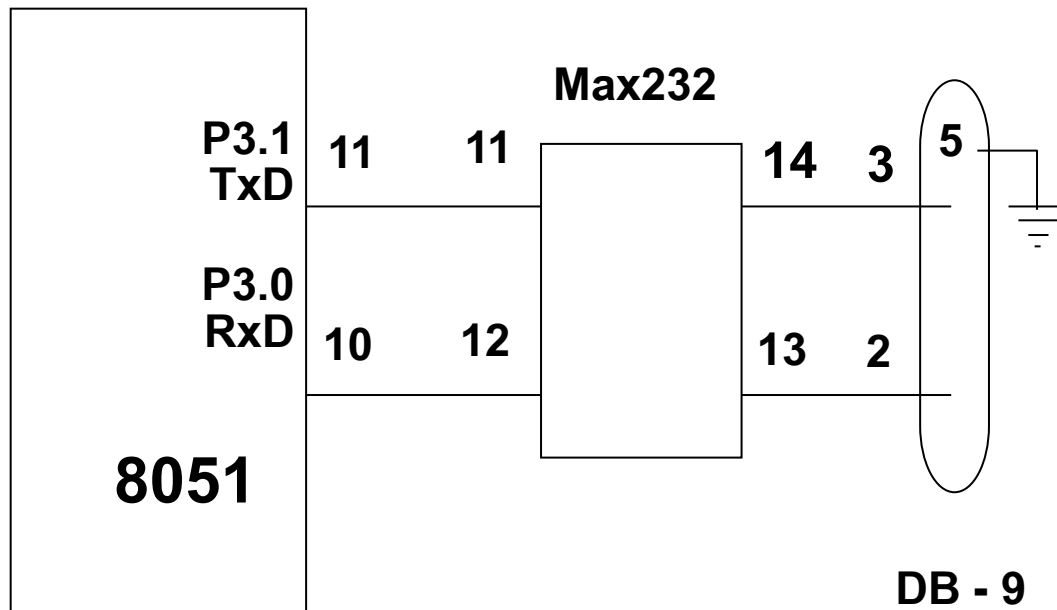
- **Phát dữ liệu nối tiếp sử dụng Timer1, Mode2 nên TMOD=20H.**
- **Tốc độ baud được thiết lập bởi TH1.**
- **SCON=50H (Timer1, Mode1, định khung 8 bit dữ liệu, 1 bit Start và 1 bit Stop)**

Các bước lập trình phát nối tiếp

- 1. Nạp TMOD=20H và TH1.**
- 2. Nạp SCON=50H.**
- 3. Khởi động Timer1.**
- 4. Ghi dữ liệu phát vào SBUF.**
- 5. Giám sát cờ phát TI.**
- 6. Xóa cờ TI.**
- 7. Lặp lại 4.**

Ví dụ 3:

Viết chương trình phát nối tiếp liên tục ký tự “A” với tốc độ 4800 baud.



Giải:

MOV TMOD,#20H

MOV TH1,#-6

;4800 baud

MOV SCON,#50

;8 bit,1 bit Stop

SETB TR1

AGAIN:

MOV SBUF,#"A"

;Phát "A"

HERE:

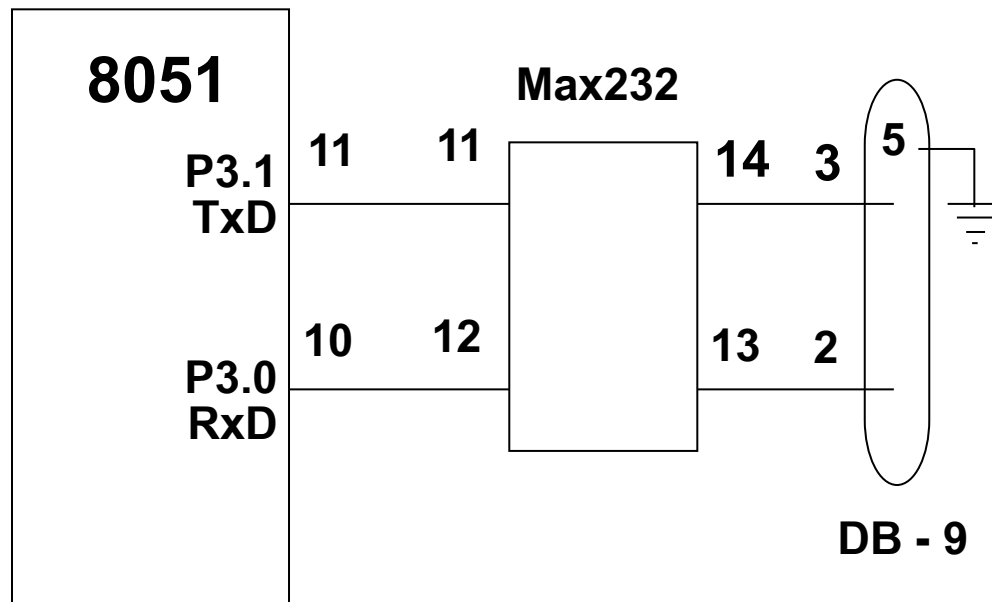
JNB TI,HERE

CLR TI

SJMP AGAIN

Ví dụ 4:

Viết chương trình fát liên tục chữ “HAY” tốc độ 9600 baud (khung 8 bit dữ liệu, 1 bit Stop).



Giải:

```
MOV  TMOD,#20H  
MOV  TH1,#-3  
MOV  SCON,#50H  
SETB TR1
```

AGAIN:

```
MOV  A,# "D"  
ACALL TRANS  
MOV  A,# "E"
```

```
ACALL TRANS
MOV  A,# "P"
ACALL TRANS
SJMP AGAIN
```

```
TRANS:
```

```
    MOV  SBUF,A
```

```
HERE:
```

```
    JNB      TI,HERE
```

```
    CLR  TI
```

```
    RET
```

Vai trò cờ TI

Các bước thực hiện phát ký tự qua TxD:

- Ghi byte ký tự cần phát vào SBUF.
- Phát bit START, 8 bit DL, bit STOP
- Phát xong STOP, TI được lập (TI=1) thông báo đã truyền xong ký tự.
- Trước khi phát byte tiếp theo cờ TI cần được xoá về 0 để báo byte là mới và có thể phát đi được.

b) Thu dữ liệu nối tiếp bằng thăm dò

Đặc điểm lập trình thu dữ liệu

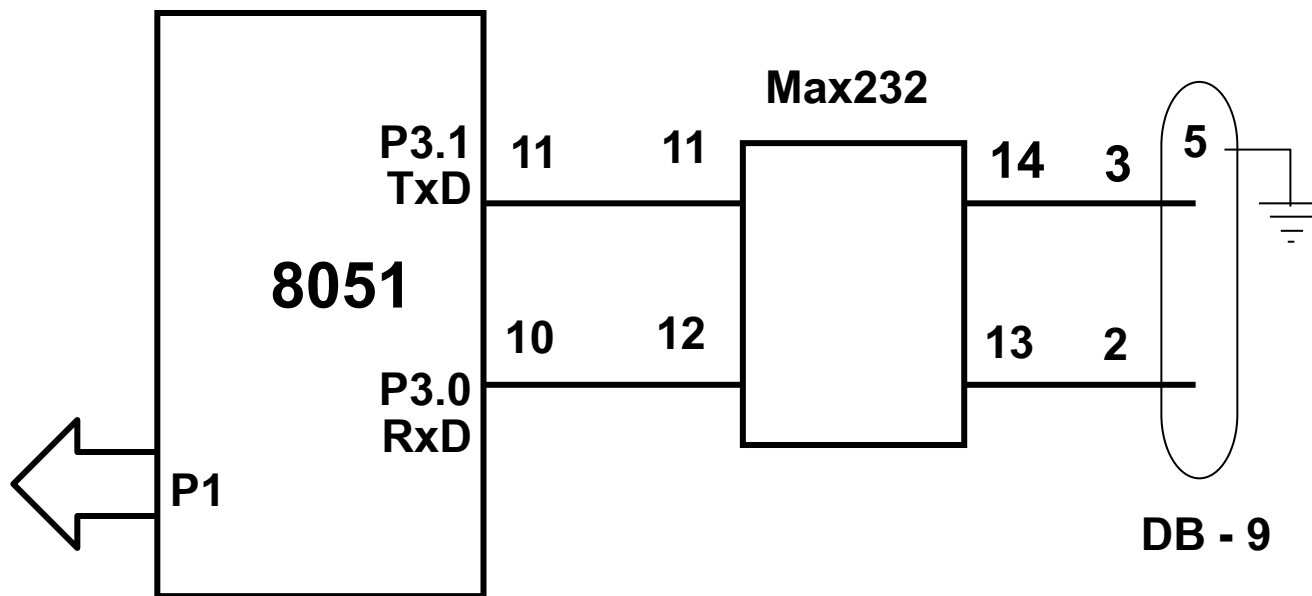
- Tương tự với phát dữ liệu.
- Có 2 điểm khác:
 - Thực hiện giám sát cờ thu RI (không phải TI)
 - Dữ liệu thu được cần lấy ra từ SBUF (không phải nạp vào SBUF)

Các bước lập trình thu dữ liệu

- 1. Nạp TMOD=20H và TH1.**
- 2. Nạp SCON=50H.**
- 3. Khởi động Timer1.**
- 4. Giám sát cờ RI .**
- 5. Khi RI=1 cất dữ liệu có trong SBUF.**
- 6. Xóa cờ RI.**
- 7. Lặp lại bước 4.**

Ví dụ 5:

Lập trình thu dữ liệu nối tiếp và gửi ra cổng P1. Tốc độ baud 4800, 8 bit dữ liệu, 1 bit Stop.



Giải:

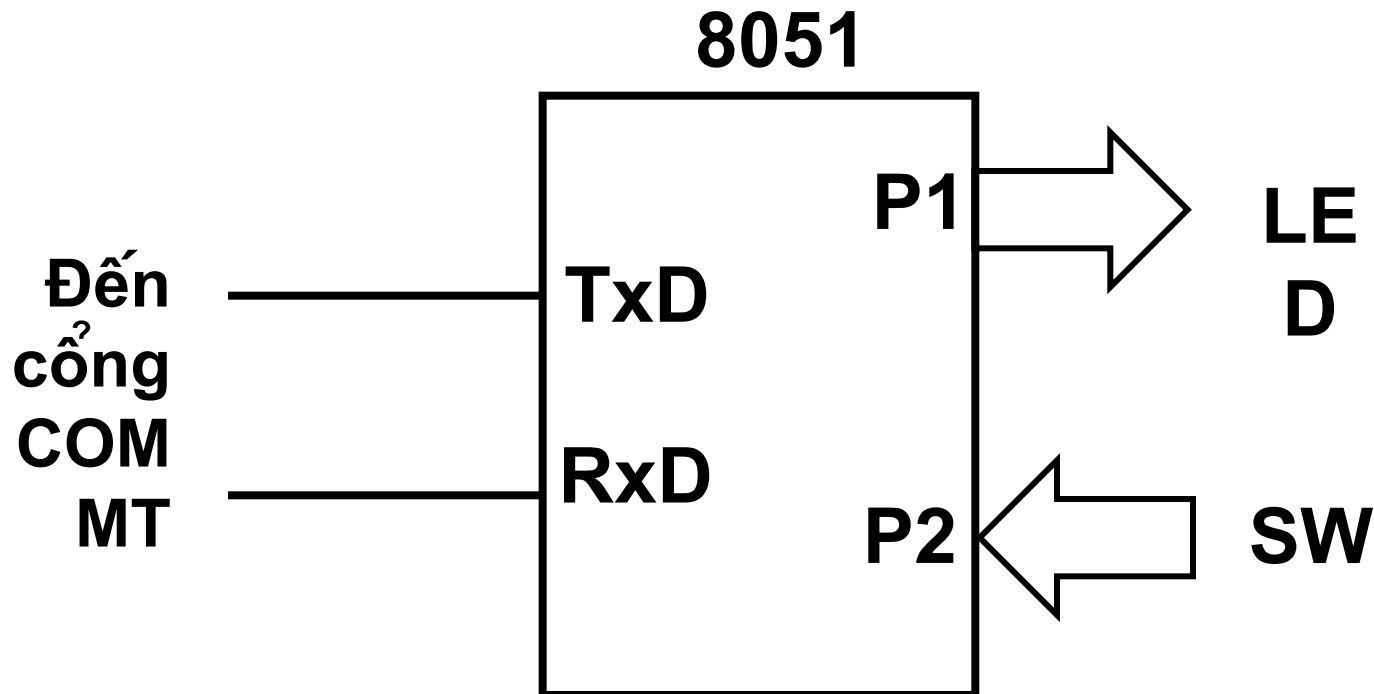
```
MOV  TMOD,#20H
MOV      TH1,#-6
MOV  SCON,#50H
SETB      TR1
```

HERE:

```
JNB  RI,HERE
MOV      A,SBUF
MOV  P1,A
CLR  RI
SJMP HERE
```

Ví dụ 6: Thu phát nối tiếp

Cho: Cổng COM 8051 nối với máy tính, cổng P1 nối LED, P2 cổng vào nối từ các công tắc chuyển mạch.



Viết chương trình:

- a) Gửi thông báo “Đ□ sẵn sàng” tới máy tính.**
- b) Nhận dữ liệu tại các chuyển mạch nối với P2 và fát nối tiếp tới PC.**
- c) Thu dữ liệu PC gửi đến và chuyển tới đèn LED đang nối đến P1.**

Thực hiện a) một lần, còn b) và c) chạy liên tục với tốc độ 4800 baud.

Các bước lập trình

1. Nạp TMOD=20H, TH1, P2 làm cổng vào.
2. Nạp SCON=50.
3. Khởi động Timer1.
4. Nạp địa chỉ Thông Báo vào DPTR
5. Kiểm tra **Fát Thông Báo** nếu xong Thực hiện **Truyền thông NT**, nếu chưa thì tiếp tục **Fat Thông báo**.
6. Viết CT con **Thu, Fat NT**.
7. Định nghĩa Thông báo.

```
ORG 0
MOV P2,#0FFH
MOV TMOD,#20H
MOV TH1,#0FAH
MOV SCON,#50H
SETB TR1
MOV DPTR,#Thong_bao
```

```
Lap: CLR A
      MOVC A,@A+DPTR
      JZ Truyen_thong
      ACALL Fat_NT
      INC DPTR
      SJMP Lap
```

Truyen_thong:

```
MOV      A,P2  
ACALL Fat_NT  
ACALL Thu_NT  
MOV      P1,A  
SJMP Truyen_thong
```

Fat_NT:

```
MOV SBUF,A
```

Lap2: JNB TI,Lap2

```
CLR TI
```

```
RET
```


Thu_NT:

JNB RI,Thu_NT

MOV A,SBUF

CLR RI

RET

Thong_bao:

DB “Đ□sẵn sàng”,0

END

Cờ RI trong việc thu dữ liệu

- **Trình tự thu dữ liệu qua chân RxD:**
 - Thu bit Start.
 - Tiếp theo thu 8 bit dữ liệu. 8 bit này được chuyển vào SBUF.
 - Khi thu xong bit Stop thì bật $RI=1$ để báo byte dữ liệu đã thu xong.
- Khi $RI=1$, cần cất dữ liệu ở SBUF. Cất xong thì xóa cờ RI về 0 để thu dữ liệu tiếp theo.

Tăng gấp đôi tốc độ baud của 8051

Có hai cách tăng tốc độ Baud:

1. Tăng tần số thạch anh gây không tương thích tốc độ Baud.
2. Thay đổi 1 bit trong thanh ghi điều khiển hiệu năng PCON (Power Control).

D7

D0

SM0D	-	-	-	GF0	GF0	PD	IDL
------	---	---	---	-----	-----	----	-----

Thanh ghi PCON

Thanh ghi PCON

- Thanh ghi PCON có 8 bit.
- Bit dành cho truyền thông nối tiếp là D7 SMOD (Serial Mode).
- Khi bật nguồn, D7=0.
- Lập D7=1 nhân đôi được tốc độ baud.
- Lưu ý, PCON không phải là thanh ghi định địa chỉ bit).

Ví dụ 7:

Viết đoạn chương trình lập D7=1 trong thanh ghi PCON

Giải:

MOV A,PCON

SETB ACC.7

MOV PCON,A

SMOD = 0 (D7=0)

$F_{uart} = F_{thạch\ anh} / (12 \times 32)$.

Nếu $F_{xtal} = 11,0592\text{MHz}$ thì:

Thì $F_{uart} = 28,8\text{ KHz}$.

SMOD = 1 (D7=1)

$F_{uart} = F_{xtal} / (12 \times 16)$

Với $F_{xtal} = 11,0592\text{ MHz}$ thì

$F_{uart} = F_{xtal} / (12 \times 16) = 57,6\text{ KHz}$.

Bảng 10.5. Tốc độ baud với SMOD=0 và SMOD=1

TH1 (thập phân)	TH1 (hexa)	Tốc độ baud	
		SMOD=0	SMOD=1
-3	FD	9.600	19.200
-6	FA	4.800	9.600
-12	F4	2.400	4.800
-24	F8	1.200	2.400

Ví dụ 8:

Cho tần số $F_{xtal} = 11,0592\text{MHz}$. Xác định:

- a) Nhiệm vụ của chương trình sau:**
- b) Tốc độ truyền dữ liệu là bao nhiêu**

```
MOV A,PCON
SETB ACC.7
MOV PCON,A
MOV TMOD,#20H
MOV TH1,-3
MOV SCON,#50H
SETB TR1
MOV A,#"B"
```

Lap1:

```
CLR TI
MOV SBUF,A
```

Lap2:

```
JNB TI,Lap2
SJMP Lap1
```

Giải:

a) Chương trình phát liên tục chữ B (42H)

**b) Vì $F_{xtal}=11,0592\text{MHz}$ và $SMOD=1$ nên
 $F_{uart}= 28,8\text{Khz} \times 2=57,6 \text{ Khz}$**

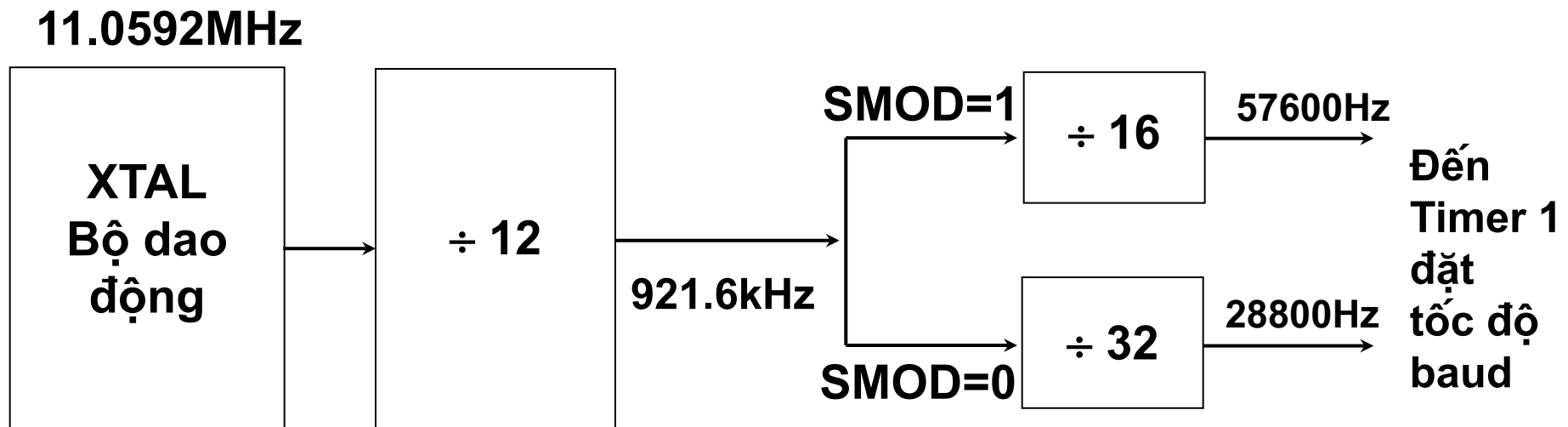
c) $57,6\text{KHz}/3 = 19.200$ là tốc độ cần xác định

Ví dụ 9:

Cho $F_{xtal} = 11,0592\text{MHz}$ và $SMOD=1$.
tìm TH1 để đặt tốc độ.

a) 9600 baud

b) 4800 baud



Giải:

**XTAL=11,0592MHz và SMOD=1 Thì
FTimer1=57,6KHz.**

a) $57.600/9600=6$ nên TH1=-6=FAH

b) $57.600/4800=12$ nên TH1=-12=F4H

Ví dụ 10:

▪ Tính tốc độ baud nếu $TH1=-2$, $SMOD=1$ và tần số $F_{xtal}=11.0592\text{MHz}$. Tốc độ này có được máy tính IBM PC và tương thích hỗ trợ không?

Tốc độ truyền dữ liệu cổng COM máy tính PC

100 Baud

150 Baud

300 Baud

600 Baud

1200 Baud

2400 Baud

4800 Baud

9600 Baud

19200 Baud

Giải:

- $F_{xtal}=11,0592\text{MHz}$ và $SMOD=1$ thì $F_{uart}=57,6\text{kHz}$.
- Tốc độ baud $=57,6\text{Khz}/2=28,800$. Tốc độ này không được IBM PC hỗ trợ nhưng có thể lập trình để PC truyền dữ liệu với tốc độ này.
- Phần mềm nhiều modem cũng như Hyperterminal của Windows có thể hỗ trợ tốc độ này và một số tốc độ khác nữa.

4. Lập trình truyền thông nối tiếp bằng phương pháp ngắt

- a) Cờ ngắt phát, cờ ngắt thu**
- b) Phát nối tiếp dùng ngắt**
- c) Thu nối tiếp dùng ngắt**

a) Cờ ngắt phát TI và ngắt thu RI

■ Cờ ngắt phát TI (Transfer Interrupt)

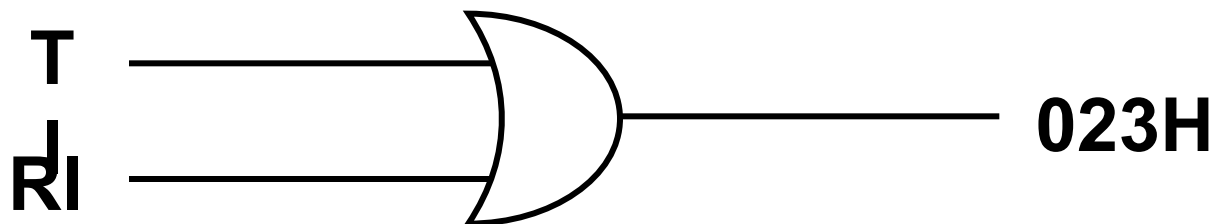
Khi bit cuối cùng của khung dữ liệu STOP được phát thì TI=1 để thông báo thanh ghi SBUF sẵn sàng phát byte tiếp theo.

■ Cờ ngắt thu RI (Receive Interrupt)

Khi toàn bộ khung dữ liệu kể cả bit Stop đã được nhận xong thì RI=1 để chuẩn bị nhận byte tiếp theo.

Cờ TI, RI phương pháp thăm dò và ngắt

- Cả hai FFáp: đều bật cờ TI, RI nh nhau.
- Điểm khác:
 - FFáp thăm dò liên tục kiểm tra cờ, gây tốn thời gian VĐK.
 - FFáp ngắt chỉ kiểm tra cờ mỗi khi ngắt được kích hoạt do đó tiết kiệm thời gian VĐK.



Hình 11.7. Thu và phát dùng một ngắt

- 8051 chỉ có 1 địa chỉ 23H chung cho cả ngắt phát và ngắt thu nên khi ngắt được kích hoạt thì hoặc RI=1 hoặc TI=1.
- Vậy tại địa chỉ 23H luôn phải kiểm tra cờ gây ngắt là TI hay RI để đáp ứng.

b) Phát nối tiếp dùng ngắt

Đặc điểm lập trình

- **Thiết lập cơ bản giống FFáp thăm dò.**
- **Điểm khác:**
 - **Tránh bảng vectơ ngắt**
 - **Nạp $IE=90H$ để dùng ngắt truyền thông nối tiếp.**
 - **Lập trình tại địa chỉ ngắt $TI+RI$ là $23H$**

D7	D6	D5	D4	D3	D2	D1	D0
EA	--	ET	ES	ET	EX	ET	EX
		2		1	1	0	0

- EA IE.7** Nếu EA=0 thì che tất cả các ngắt.
Nếu EA=1 thì ngắt được mở/cấm bằng cách đặt/xoá bit tương ứng.
- ET2 IE.5** Cho_phép/cấm ngắt tràn Timer2 (8952)
- ES IE.4** Cho_phép/cấm ngắt cổng nối tiếp
- ET1 IE.3** Cho_phép/cấm ngắt tràn Timer 1
- EX1 IE.2** Cho_phép/cấm ngắt INT1
- ET0 IE.1** Cho_phép/cấm ngắt tràn Timer 0
- EX0 IE.0** Cho_phép/cấm ngắt INT0

Thanh ghi cho phép ngắt IE

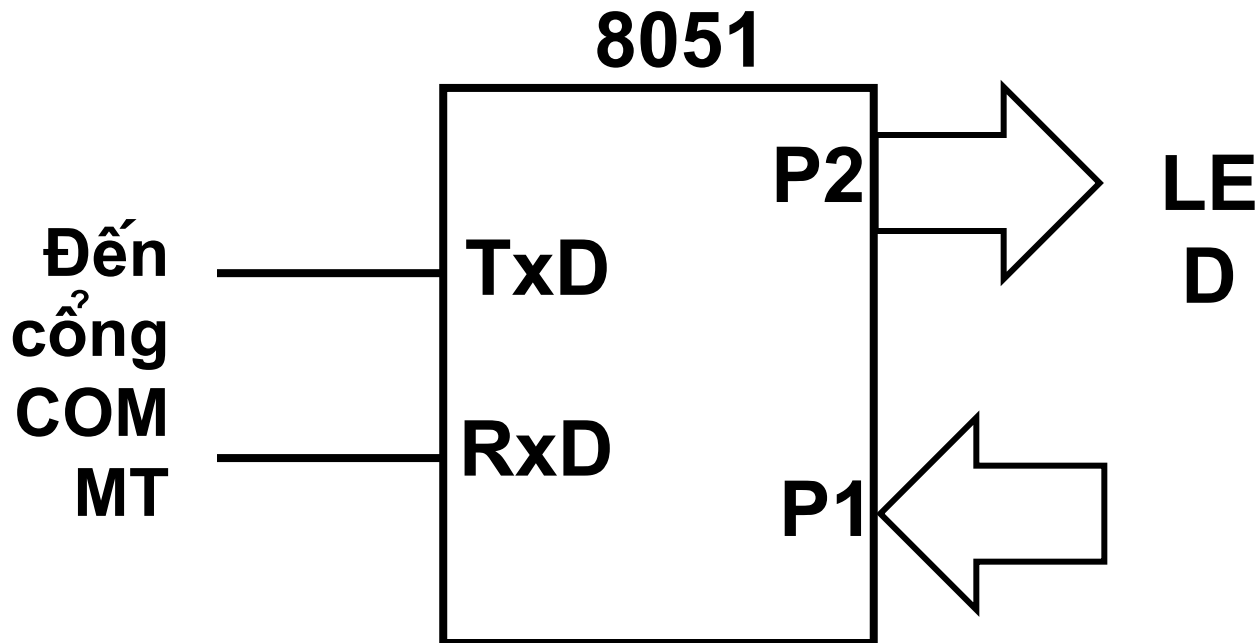
Các bước lập trình phát NT dùng ngắt

- 1. Tránh bảng vectơ ngắt**
- 2. Nạp TMOD =20H và TH1.**
- 3. Nạp SCON=50H.**
- 4. Nạp IE=90H.**
- 5. Khởi tạo Timer.**
- 6. Tạo vòng lặp đợi ngắt.**
- 7. Lập trình tại địa chỉ 23H.**

Ví dụ 11: Viết chương trình phát nối tiếp.

- 1. Nhận dữ liệu từ cổng P1.**
- 2. Gửi dữ liệu nhận được tới cổng P2.**
- 3. Fát dữ liệu nối tiếp tới cổng COM với tốc độ 9600 baud**

Cho tần số Fxatl=11,0592MHz.



Các bước lập trình phát NT dùng ngắt

- 1. Tránh bảng vectơ ngắt**
- 2. Nạp TMOD=20H, TH1 và P1 cổng vào.**
- 3. Nạp SCON=50H.**
- 4. Nạp IE=90H.**
- 5. Khởi tạo Timer.**
- 6. Lập vòng lặp xử lý dữ liệu và đợi ngắt.**
- 7. Lập trình tại địa chỉ 23H.**

Giải:

ORG 0

LJMP MAIN

ORG 23H

LJMP Truyen_NT

MAIN:

MOV P1,#0FFH

MOV TMOD,#20h

MOV TH1,#0FDH

MOV SCON,#50H

MOV IE,#90H

SETB TR1

BACK:

**MOV A,P1
MOV SBUF,A
MOV P2,A
SJMP BACK**

ORG 100H

Truyen_NT:

**JB TI,FAT
MOV A,SBUF
CLR RI
RETI**

FAT:

**CLR TI
RETI
END**

Bảng 11.1. Bảng vector ngắt của 8051

Ngắt	Địa chỉ	Chân
RESET	0H	9
Ngắt ngoài INT0	3H	12
Ngắt bộ Timer0 (TF0)	BH	(P3.2)
Ngắt ngoài INT1	13H	13
Ngắt bộ Timer1 (TF1)	1BH	(P3.3)
Ngắt cổng COM (RI và TI)	23H	

Lu ý:

Tại địa chỉ ngắt, cần kiểm tra cả cờ TI và cờ RI vì cả hai đều gọi cùng một ngắt

c) Thu nối tiếp dùng ngắt

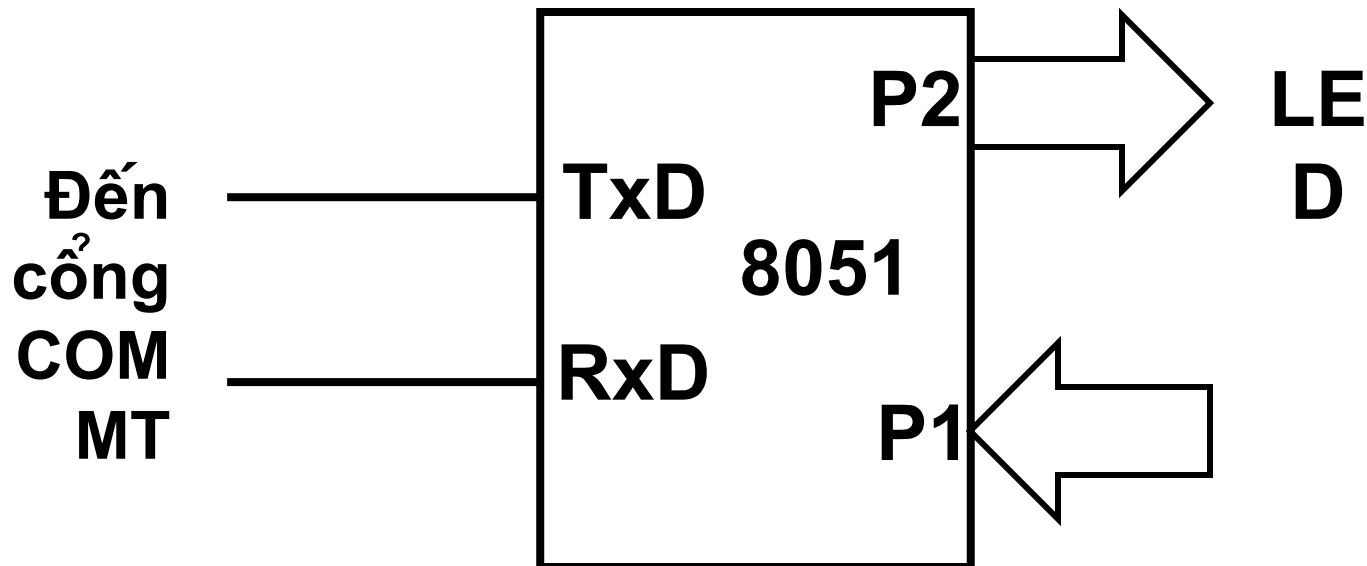
Đặc điểm lập trình

- Các bước lập trình giống phát nối tiếp dùng ngắt, nh Tránh bảng vectơ ngắt, thiết lập $IE=90H$, thiết lập $SCON....$
- Tại địa chỉ ngắt 23 H (của $TI+RI$) vẫn kiểm tra cờ RI , TI .

Ví dụ 12:

Viết chương trình thu nối tiếp:

1. Nhận dữ liệu ở cổng P1.
2. Gửi dữ liệu nhận được đến cổng P2.
3. Dữ liệu thu được ở cổng COM gửi đến P0 với 9600 baud. Fxtal=11,0592MHz.



Các bước lập trình thu NT dùng ngắt

- 1. Tránh bảng vectơ ngắt**
- 2. Nạp TMOD =20H, TH1, P1 cổng vào.**
- 3. Nạp SCON=50H.**
- 4. Nạp IE=90H.**
- 5. Khởi tạo Timer.**
- 6. Tạo vòng lặp xử lý dữ liệu và đợi ngắt.**
- 7. Lập trình tại địa chỉ 23H.**

Lu ý:

- Về cơ bản, chương trình thu nối tiếp giống với phát nối tiếp.
- Điểm khác:
 - Phát cần lệnh gửi dữ liệu vào SBUF "MOV SBUF, A", thu không dùng lệnh này.
 - Thu cần cần lấy dữ liệu ra từ SBUF "MOV A, SBUF".

Xoá cờ RI và TI trước lệnh RETI

- **Lập trình viên cần xoá cờ RI và TI trước RETI để ngắt tiếp theo được đáp ứng.**
- **Đây là điểm khác với ngắt ngoài và ngắt bộ định thời, ở đó VĐK sẽ tự động xoá cờ, không cần lập trình viên thực hiện.**

Tóm tắt bài truyền thông nối tiếp:

- **TTNT có 2 PP lập trình: Ngắt và thăm dò.**
- **PP thăm dò:**
 - **VĐK liên tục giám sát cờ TI, RI nên bị chiếm nhiều thời gian.**
 - **Cờ TI và RI được giám sát riêng trong quá trình phát hoặc thu.**

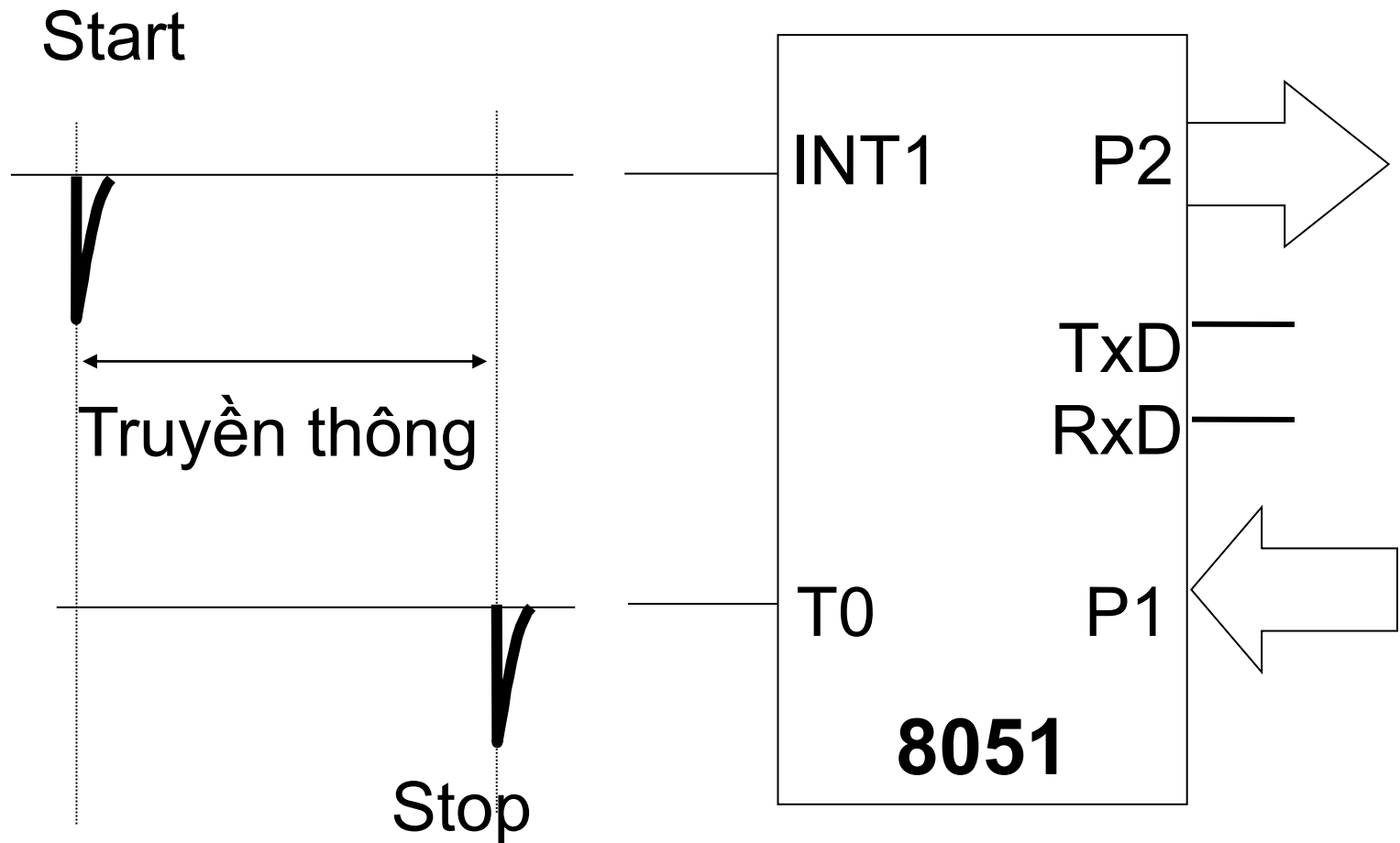
■ PP ngắt:

- VĐK chỉ kiểm tra cờ Tl, Rl mỗi khi ngắt được kích hoạt nên tiết kiệm nhiều thời gian cho VĐK.
- Tại địa chỉ ngắt 23 H luôn kiểm tra cả 2 cờ Tl và Rl.

Ví dụ 14: Bài tổng hợp truyền thông nối tiếp

Lập trình thực hiện truyền thông nối tiếp trong thời gian giữa cặp 2 xung START và STOP theo cách như sau:

- Xung START đưa đến ngắt **INT1** để bắt đầu quá trình truyền thông nối tiếp sử dụng phương pháp ngắt. Nhận dữ liệu ở cổng **P1** phát nối tiếp, thu dữ liệu nối tiếp và gửi đến cổng **P2**.
- Xung STOP đưa đến chân **T0** (P3.4) của COUNTER0 làm việc chế độ giám sát cò tràn TF0 để kết thúc truyền thông sau đó chương trình đợi cặp 2 xung START STOP tiếp theo.
- Cho Fxtal=11,0592 MHz, tốc độ baud 9600.



- T0 (Counter0): phương pháp giám sát cờ tràn TF0
- Truyền thông nối tiếp: phương pháp ngắt

ORG 0H
LJMP MAIN

ORG 30H

MAIN:

SETB P3.4
MOV P1,#0FFh
MOV TMOD,#26H
MOV TH1,#FDH
MOV TH0,#FFH
MOV SCON,#50H
MOV IE,#84H

LAP:

SJMP LAP

ORG 13H
LJMP XU_LY
RETI

ORG 100

XU_LY:

SETB IE.4
SETB TR1
SETB TR0

LAP2:

MOV A, P1
MOV SBUF, A
JNB TF0, LAP2
CLR TF0
CLR TR0
CLR TR1
LJMP MAIN
RET

ORG 23H
JB TI, FAT_NT
MOV A, SBUF
MOV P2,A
CLR RI
RETI

FAT_NT
CLR TI
RETI

END

A. Lập trình ngắt Timer

- **Timer0: IE = 82H, Địa chỉ 0BH**
- **Timer1: IE = 88H, Địa chỉ 1BH**

B. Lập trình ngắt ngoài kích phát mức

- **INT0: IE = 81H, Địa chỉ 03H**
- **INT1: IE = 84H, Địa chỉ 13H**

C. Lập trình ngắt ngoài kích phát sườn

- **INT0: SETB IT0**
IE = 81H, Địa chỉ 03H
- **INT1: SETB IT1**
IE = 84H, Địa chỉ 13H