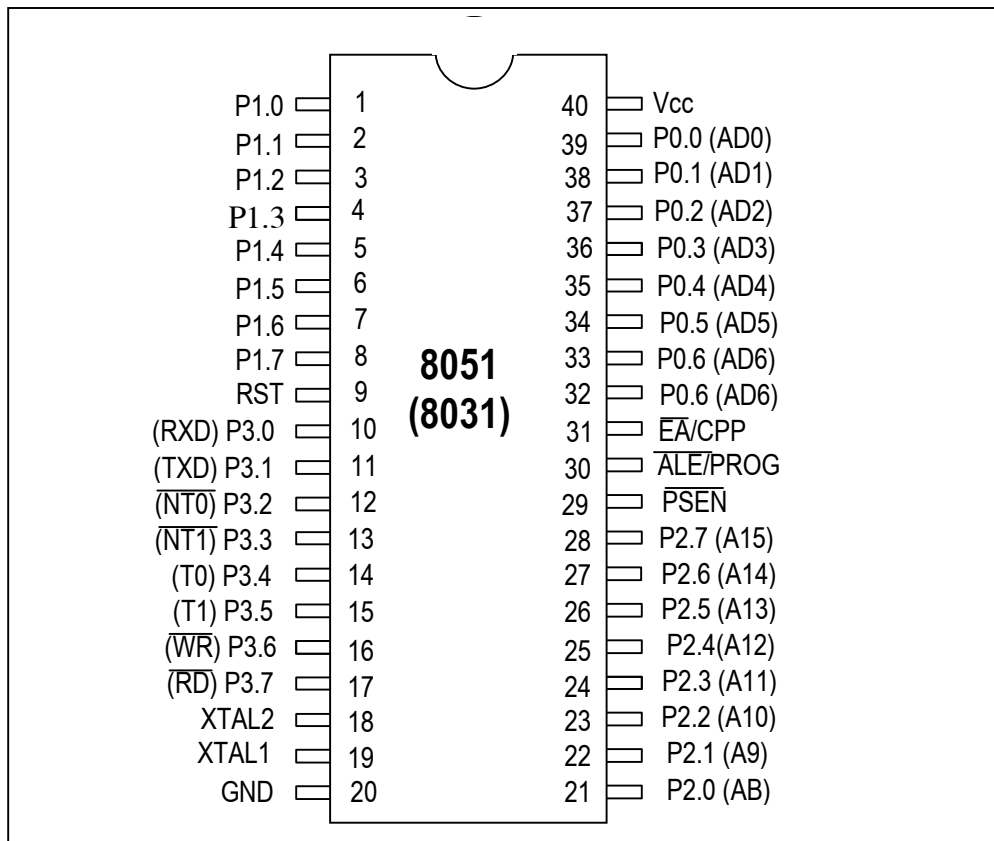


CHƯƠNG 4

Lập trình cho cổng vào - ra I/O

4.1 Mô tả chân của 8051.

Mặc dù các thành viên của họ 8051 (ví dụ 8751, 89C51, DS5000) đều có các kiểu đóng vỏ khác nhau, chẳng hạn như hai hàng chân DIP (Dual In-Line Package) dạng vỏ dẹt vuông QFP (Quad Flat Package) và dạng chip không có chân đỡ LLC (Leadless Chip Carrier) thì chúng đều có 40 chân cho các chức năng khác nhau như vào ra I/O, đọc \overline{RD} , ghi \overline{WR} , địa chỉ, dữ liệu và ngắt. Cần phải lưu ý rằng một số hãng cung cấp một phiên bản 8051 có 20 chân với số cổng vào-ra ít hơn cho các ứng dụng yêu cầu thấp hơn. Tuy nhiên, vì hầu hết các nhà phát triển chính sử dụng chip đóng vỏ 40 chân với hai hàng chân DIP nên ta chỉ tập chung mô tả phiên bản này.



Hình 4.1: Sơ đồ bố trí chân của 8051.

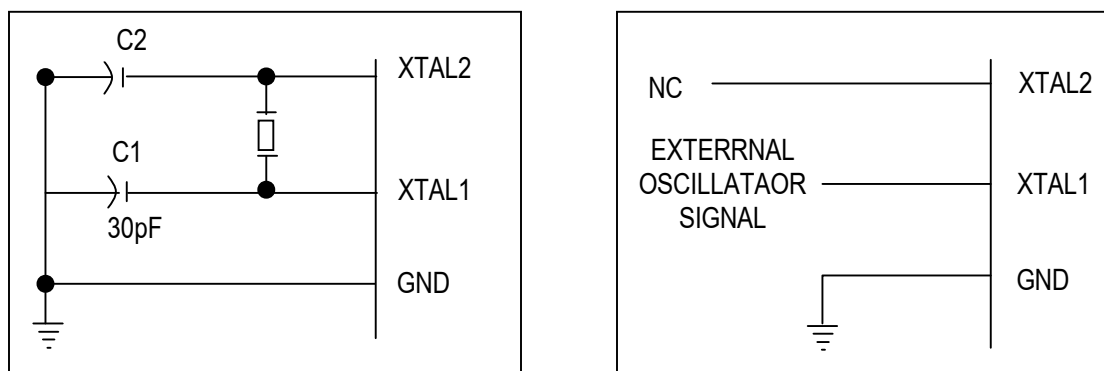
Trên hình 4.1 là sơ đồ bố trí chân của 8051. Ta thấy rằng trong 40 chân thì có 32 chân dành cho các cổng P0, P1, P2 và P3 với mỗi cổng có 8 chân. Các chân còn lại được dành cho nguồn V_{CC} , đất GND, các chân giao động XTAL1 và XTAL2 tái lập RST cho phép chốt địa chỉ ALE truy cập được địa chỉ ngoài \overline{EA} , cho phép cắt chương trình \overline{PSEN} . Trong 8 chân này thì 6 chân V_{CC} , GND, XTAL1, XTAL2, RST và \overline{EA} được các họ 8031 và 8051 sử dụng. Hay nói cách khác là chúng phải được

nối để cho hệ thống làm việc mà không cần biết bộ vi điều khiển thuộc họ 8051 hay 8031. Còn hai chân khác là $\overline{\text{PSEN}}$ và ALE được sử dụng chủ yếu trong các hệ thống dựa trên 8031.

1. Chân V_{CC} : Chân số 40 là V_{CC} cấp điện áp nguồn cho chip. Nguồn điện áp là +5V.
2. Chân GND: Chân GND: Chân số 20 là GND.
3. Chân XTAL1 và XTAL2:

8051 có một bộ giao động trên chip nhưng nó yêu cầu có một xung đồng hồ ngoài để chạy nó. Bộ giao động thạch anh thường xuyên nhất được nối tới các chân đầu vào XTAL1 (chân 19) và XTAL2 (chân 18). Bộ giao động thạch anh được nối tới XTAL1 và XTAL2 cũng cần hai tụ điện giá trị 30pF. Một phía của tụ điện được nối xuống đất như được trình bày trên hình 4.2a.

Cần phải lưu ý rằng có nhiều tốc độ khác nhau của họ 8051. Tốc độ được coi như là tần số cực đại của bộ giao động được nối tới chân XTAL. Ví dụ, một chip 12MHz hoặc thấp hơn. Tương tự như vậy thì một bộ vi điều khiển cũng yêu cầu một tinh thể có tần số không lớn hơn 20MHz. Khi 8051 được nối tới một bộ giao động tinh thể thạch anh và cấp nguồn thì ta có thể quan sát tần số trên chân XTAL2 bằng máy hiện sóng. Nếu ta quyết định sử dụng một nguồn tần số khác bộ giao động thạch anh chẳng hạn như là bộ giao động TTL thì nó sẽ được nối tới chân XTAL1, còn chân XTAL2 thì để hở không nối như hình 4.2b.



Hình 4.2: a) Nối XTAL tới 8051 b) Nối XTAL tới nguồn đồng bộ ngoài.

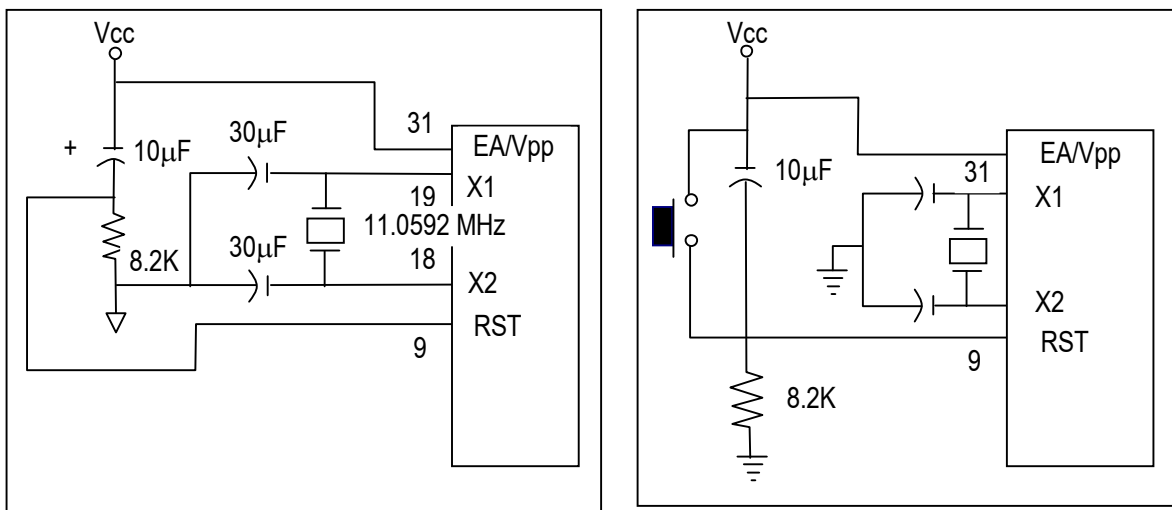
4. Chân RST.

Chân số 9 là chân tái lập RESET. Nó là một đầu vào và có mức tích cực cao (bình thường ở mức thấp). Khi cấp xung cao tới chân này thì bộ vi điều khiển sẽ tái lập và kết thúc mọi hoạt động. Điều này thường được coi như là sự tái bật nguồn. Khi kích hoạt tái bật nguồn sẽ làm mất mọi giá trị trên các thanh ghi. Bảng 4.1 cung cấp một cách liệt kê các thanh ghi của 8051 và các giá trị của chúng sau khi tái bật nguồn.

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
	0000

Lưu ý rằng giá trị của bộ đếm chương trình PC là 0 khi tái lập để ép CPU nạp mã lệnh đầu tiên từ bộ nhớ ROM tại vị trí ngăn nhớ 0000. Điều này có nghĩa là ta phải đặt dòng đầu tiên của mã nguồn tại vị trí ngăn nhớ 0 của ROM vì đây là mã CPU tỉnh thức và tìm lệnh đầu tiên. Hình 4.3 trình bày hai cách nối chân RST với mạch bật nguồn.



Hình 4.3: a) Mạch tái bật nguồn RESET.

b) Mạch tái bật nguồn với Debounce.

Nhằm làm cho đầu vào RESET có hiệu quả thì nó phải có tối thiểu 2 chu kỳ máy. Hay nói cách khác, xung cao phải kéo dài tối thiểu 2 chu kỳ máy trước khi nó xuống thấp.

Trong 8051 một chu kỳ máy được định nghĩa bằng 12 chu kỳ dao động như đã nói ở chương 3 và được trình bày tại vị trí 4.1.

5. Chân \overline{EA} :

Các thành viên họ 8051 như 8751, 98C51 hoặc DS5000 đều có ROM trên chip lưu cất chương trình. Trong các trường hợp như vậy thì chân \overline{EA} được nối tới V_{CC}. Đối với các thành viên củ họ như 8031 và 8032 mà không có ROM trên chip thì mã chương trình được lưu cất ở trên bộ nhớ ROM ngoài và chúng được nạp cho 8031/32. Do vậy, đối với 8031 thì chân \overline{EA} phải được nối đất để báo rằng mã chương trình được cất ở ngoài. \overline{EA} có nghĩa là truy cập ngoài (External Access) là chân số 31 trên vỏ kiểu DIP. Nó là một chân đầu vào và phải được nối hoặc với V_{CC} hoặc GND. Hay nói cách khác là nó không được để hở.

Ở chương 14 chúng ta sẽ trình bày cách 8031 sử dụng chân này kết hợp với $\overline{\text{PSEN}}$ để truy cập các chương trình được cất trên bộ nhớ ROM ở ngoài 8031. Trong các chip 8051 với bộ nhớ ROM trên chip như 8751, 89C51 hoặc DS5000 thì EA được nối với V_{CC} .

Ví dụ 4:

Hãy tìm chu kỳ máy đối với a) XTAL = 11.0592MHz b) XTAL = 16MHz.

Lời giải:

a) $11.0592\text{MHz}/12 = 921.6\text{kHz}$.

Chu kỳ máy = $1/921.6\text{kHz} = 1.085\mu\text{s}$.

b) $16\text{MHz}/12 = 1.333\text{MHz}$

Chu kỳ máy = $1/1.333\text{MHz} = 0.75\mu\text{s}$.

Các chân mô tả trên đây phải được nối mà không cần thành viên nào được sử dụng. Còn hai chân dưới đây được sử dụng chủ yếu trong hệ thống dựa trên 8031 và sẽ được trình bày chi tiết ở chương 11.

6. Chân $\overline{\text{PSEN}}$:

Đây là chân đầu ra cho phép cất chương trình (Program Store Enable) trong hệ thống dựa trên 8031 thì chương trình được cất ở bộ nhớ ROM ngoài thì chân này được nối tới chân OE của ROM. Chi tiết được bàn ở chương 14.

7. Chân ALE:

Chân cho phép chốt địa chỉ ALE là chân đầu ra và được tích cực cao. Khi nối 8031 tới bộ nhớ ngoài thì cổng 0 cũng được cấp địa chỉ và dữ liệu. Hay nói cách khác 8031 dồn địa chỉ và dữ liệu qua cổng 0 để tiết kiệm số chân. Chân ALE được sử dụng để phân kênh địa chỉ và dữ liệu bằng cách nối tới chân G của chip 74LS373. Điều này được nói chi tiết ở chương 14.

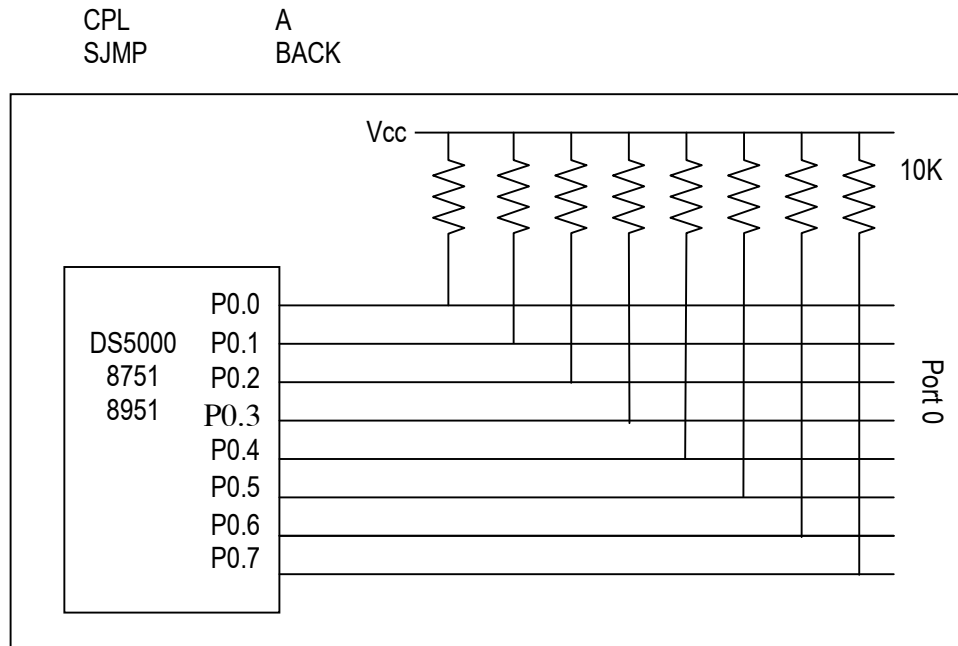
8. Các chân cổng vào ra và các chức năng của chúng.

Bốn cổng P0, P1, P2 và P3 đều sử dụng 8 chân và tạo thành cổng 8 bit. Tất cả các cổng khi RESET đều được cấu hình như các đầu ra, sẵn sàng để được sử dụng như các cổng đầu ra. Muốn sử dụng cổng nào trong số các cổng này làm đầu vào thì nó phải được lập trình.

9. Cổng P0.

Cổng 0 chiếm tất cả 8 chân (từ chân 32 đến 39). Nó có thể được dùng như cổng đầu ra, để sử dụng các chân của cổng 0 vừa làm đầu ra, vừa làm đầu vào thì mỗi chân phải được nối tới một điện trở kéo bên ngoài $10\text{k}\Omega$. Điều này là do một thực tế là cổng P0 là một mảng mở khác với các cổng P1, P2 và P3. Khái niệm mảng mở được sử dụng trong các chip MOS về chừng mực nào đó nó giống như Cô-lec-tơ hở đối với các chip TTL. Trong bất kỳ hệ thống nào sử dụng 8751, 89C51 hoặc DS5000 ta thường nối cổng P0 tới các điện trở kéo, Xem hình 4.4 bằng cách này ta có được các ưu điểm của cổng P0 cho cả đầu ra và đầu vào. Với những điện trở kéo ngoài được nối khi tái lập cổng P0 được cấu hình như một cổng đầu ra. Ví dụ, đoạn mã sau đây sẽ liên tục gửi ra cổng P0 các giá trị 554 và AAH.

```
BACK:    MOV     A, #554
          MOV     P0, A
          ACALL  DELAY
```



Hình 4.4: Cổng P0 với các điện trở kéo.

- a) Cổng P0 đầu vào: Với các điện trở được nối tới cổng P0 nhằm để tạo nó thành cổng đầu vào thì nó phải được lập trình bằng cách ghi 1 tới tất cả các bit. Đoạn mã dưới đây sẽ cấu hình P0 lúc đầu là đầu vào bằng cách ghi 1 đến nó và sau đó dữ liệu nhận được từ nó được gửi đến P1.

b)

	MOV	A, #FFH	; Gán A = FF dạng Hex
	MOV	P0, A	; Tạo cổng P0 làm cổng đầu vào bằng cách
			; Ghi tất cả các bit của nó.
BACK:	MOV	A, P0	; Nhận dữ liệu từ P0
	MOV	P1, A	; Gửi nó đến cổng 1
	SJMP	BACK	; Lặp lại

b) Vai trò kép của cổng P0: Như trình bày trên hình 4.1, cổng P0 được gán AD0 - AD7 cho phép nó được sử dụng vừa cho địa chỉ, vừa cho dữ liệu. Khi nối 8051/31 tới bộ nhớ ngoài thì cổng 0 cung cấp cả địa chỉ và dữ liệu 8051 dồn dữ liệu và địa chỉ qua cổng P0 để tiết kiệm số chân. ALE báo nếu P0 có địa chỉ hay dữ liệu khi ALE - 0 nó cấp dữ liệu D0 - D7. Do vậy, ALE được sử dụng để tách địa chỉ và dữ liệu với sự trợ giúp của chốt 74LS373 mà ta sẽ biết cụ thể ở chương 14.

10. Cổng P1.

Cổng P1 cũng chiếm tất cả 8 chân (từ chân 1 đến chân 8) nó có thể được sử dụng như đầu vào hoặc đầu ra. So với cổng P0 thì cổng này không cần đến điện trở kéo vì nó đã có các điện trở kéo bên trong. Trong quá trình tái lập thì cổng P1 được cấu hình như một cổng đầu ra. Ví dụ, đoạn mã sau sẽ gửi liên tục các giá trị 55 và AAH ra cổng P1.

	MOV	A, #55H	
BACK:	MOV	P1, A	

```
ACALL DELAY
SJMP      BACK
```

Cổng P1 như đầu vào: Để biến cổng P1 thành đầu vào thì nó phải được lập trình bằng cách ghi một đến tất cả các bit của nó. Lý do về điều này được bàn ở mục lục Appendix C.2. Trong đoạn mã sau, cổng P1 lúc đầu được cấu hình như cổng đầu vào bằng cách ghi 1 vào các bit của nó và sau đó dữ liệu nhận được từ cổng này được cất vào R7, R6 và R5.

```
MOV      A, #0FFH      ; Nạp A = FF ở dạng hex
MOV      P1, A          ; Tạo cổng P1 thành cổng đầu vào bằng
                        ; cách ghi 1 vào các bit của nó.
MOV      A, P1          ; Nhận dữ liệu từ P1
MOV      R7, A          ; Cất nó vào thanh ghi R7
ACALL    DELAY          ; Chờ
MOV      A, P1          ; Nhận dữ liệu khác từ P1
MOV      R6, A          ; Cất nó vào thanh ghi R6
ACALL    DELAY          ; Chờ
MOV      A, P1          ; Nhận dữ liệu khác từ cổng P1
MOV      R5, A          ; Cất nó vào thanh ghi R5
```

11. Cổng P2:

Cổng P2 cũng chiếm 8 chân (các chân từ 21 đến 28). Nó có thể được sử dụng như đầu vào hoặc đầu ra giống như cổng P1, cổng P2 cũng không cần điện trở kéo vì nó đã có các điện trở kéo bên trong. Khi tái lập, thì cổng P2 được cấu hình như một cổng đầu ra. Ví dụ, đoạn mã sau sẽ gửi liên tục ra cổng P2 các giá trị 55H và AAH. Đó là tất cả các bit của P2 lên xuống liên tục.

```
BACK:    MOV      A, #55H
          MOV      P2, A
          ACALL    DELAY
          CPL      A
          SJMP     BACK
```

a) Cổng P2 như đầu vào.

Để tạo cổng P2 như đầu vào thì nó phải được lập trình bằng cách ghi các số 1 tới tất cả các chân của nó. Đoạn mã sau đây đầu tiên cấu hình P2 là cổng vào bằng cách ghi một đến tất cả các chân của nó và sau đó dữ liệu nhận được từ P2 được gửi liên tục đến P1.

```
BACK:    MOV      A, 0FFH      ; Gán A giá trị FF dạng Hex
          MOV      P2, A      ; Tạo P2 là cổng đầu vào bằng cách
                        ; ghi một đến các chân của nó
          MOV      A, 2        ; Nhận dữ liệu từ P2
          MOV      P1, A      ; Gửi nó đến P1
          SJMP     BACK        ; Lặp lại
```

b) Vai trò kép của P2.

Trong các hệ thống dựa trên 8751, 89C51 và DS5000 thì P2 được dùng như đầu ra đơn giản. Tuy nhiên trong hệ thống dựa trên 80312 thì cổng P2 phải được dùng cùng với P0 để tạo ra địa chỉ 16 bit đối với bộ nhớ ngoài. Như chỉ ra trên hình 4.1 cổng P2 cũng được chỉ định như là A8 - A15 báo chức năng kép của nó. Vì một bộ 8031 có khả năng trung cấp 64k byte bộ nhớ ngoài, nó cần một đường địa chỉ 16 bit. Trong khi P.0 cung cấp 8 bit thấp qua A0 - A7. Công việc của P2 là cung cấp các bit địa chỉ A8 - A15. Hay nói cách khác khi 8031 được nối tới bộ nhớ ngoài thì P2 được dùng cho 8 bit của địa chỉ 16 bit và nó không thể dùng cho vào ra. Điều này sẽ được trình bày chi tiết ở chương 14.

Từ những trình bày trên đây ta có thể kết luận rằng trong các hệ thống dựa trên các bộ vi điều khiển 8751, 89C51 hoặc DS5000 thì ta có 3 cổng P0, P1 và P2 cho các thao tác vào ra và như thế là có thể đủ cho các ứng dụng với hầu hết các bộ vi điều khiển. Còn cấp P3 là để dành cho ngắt và ta sẽ cùng bàn dưới đây.

11 - Cổng P3:

Cổng P3 chiếm tổng cộng là 8 chân từ chân 10 đến chân 17. Nó có thể được sử dụng như đầu vào hoặc đầu ra. Cổng P3 không cần các điện trở kéo cũng như P1 và P2. Mặc dù cổng P3 được cấu hình như một cổng đầu ra khi tái lập, nhưng đây không phải là cách nó được ứng dụng phổ biến nhất. Cổng P3 có chức năng bổ xung là cung cấp một số tín hiệu quan trọng đặc biệt chẳng hạn như các ngắt. Bảng 4.2 cung cấp các chức năng khác của cổng P3. Thông tin này áp dụng cho cả 8051 và 8031.

Bảng 4.2: Các chức năng khác của cổng P3

Bít của cổng P3	Chức năng	chân số
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 (TO)	14
P3.5	1 Bộ định thời 1 (T1)	15
P3.6	Ghi (WR)	16
P3.7	Đọc (RD)	17

Các bit P3.0 và P3.1 được dùng cho các tín hiệu nhận và phát dữ liệu trong truyền thông dữ liệu nối tiếp. Xem chương 10 để biết các chúng được nối ghép như thế nào. Các bit P3.2 và P3.3 được dành cho các ngắt ngoài và chúng được trình bày chi tiết ở chương 11. Bit P3.4 và P3.5 được dùng cho các bộ định thêm 0 và 1 và chi tiết được trình bày ở chương 9. Cuối cùng các bit P3.6 và P3.7 được cấp cho các tín hiệu ghi và đọc các bộ nhớ ngoài được nối tới các hệ thống dựa trên 8031. Chương 14 sẽ trình bày cách chúng được sử dụng như thế nào trong các hệ thống dựa trên 8031. Trong các hệ thống dựa trên 8751, 89C51 hoặc DS5000 thì các chân P3.6 và P3.7 được dùng cho vào - ra còn các chân khác của P3 được sử dụng bình thường trong vai trò chức năng thay đổi.

4.2 Lập trình vào - ra: thao tác bit.

4.2.1 các cách khác nhau để truy cập toàn bộ 8 bit.

Trong đoạn mã dưới đây cũng như trong nhiều ví dụ vào ra trước đây toàn bộ 8 bit của cổng P1 được cập.

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

Đoạn mã trên chốt mỗi bit của P1 một cách liên tục. Chúng ta đã thấy một biến thể của chương trình trên trước đó. Bây giờ ta có thể viết lại đoạn mã trên theo cách hiệu quả hơn bằng cách truy cập trực tiếp cổng mà không qua thanh ghi tổng như sau:

```
BACK:    MOV      P1, # 55H
          ACALL    DELAY
          MOV      P1, #00H
          CALL     DELAY
          SJMP     BACK
```

Ta có thể viết một dạng khác của đoạn mã trên bằng kỹ thuật đọc - sửa đổi ghi như ở mục 4.2.2 dưới đây.

4.2.2 Đặc điểm Đọc- sửa đổi - ghi (Read - Modify – Write).

Các cổng trong 8051 có thể được truy cập bằng kỹ thuật được gọi là Đọc-sửa đổi-ghi. Đặc điểm này tiết kiệm rất nhiều dòng lệnh bằng cách kết hợp tất cả 3 thao tác: 1đọc cổng, 2 sửa đổi nó và 3 ghi nó ra cổng vào một lệnh đơn. Đoạn mã dưới đây trước hết đặt 01010101 (nhị phân) vào cổng 1. Sau đó lệnh “XLR P1, #0FFH” thực hiện phép lô-gích OR loại trừ là XOR trên cổng p1 với 1111 1111 (nhị phân) và sau đó ghi kết quả trở lại cổng P1.

```
AGAIN:    MOV      P1, #55H      ; P1 = 01010101
          XLR      P1, # 0FFH    ; EX - OR P1 với 1111 1111
          ACALL    DELAY
          SJMP     AGAIN
```

Lưu ý rằng lệnh XOR của 55H và FFH sẽ cho kết quả là AAH. Tương tự như vậy lệnh XOR của AAH với FFH lại cho giá trị kết quả là 55H. Các lệnh lô-gích được trình bày ở chương 7.

4.2.3. Khả năng đánh địa chỉ theo bit của các cổng

Có nhiều lúc chúng ta cần truy cập chỉ 1 hoặc 2 bit của cổng thay vì truy cập cả 8 bit của cổng. Một điểm mạnh của các cổng 8051 là chúng có khả năng truy cập từng bit riêng rẽ mà không làm thay đổi các bit còn lại trong cổng đó ví dụ, đoạn mã dưới đây chốt bit P1.2 liên tục:

```
BACK:    CPL      P1.2          ; Lấy bù 2 chỉ riêng bit P1.2
```



```

ACALL DELAY
SJMP     BACK

```

Một biến thể khác của đoạn mã trên là:

```

AGCN:    SETB     P1.2           ; Chỉ thay đổi bit P1.2 lên cao
          ACALL    DELAY
          CLR      P1.2         ; Xoá bit P1.2 xuống thấp
          ACALL    DELAY
          SJMP     AGAIN

```

Lưu ý rằng bit P1.2 là bit thứ 3 của cổng P1, vì bit thứ nhất là P1.0 và bit thứ hai là P1.1 v.v...

Bảng 4.3 trình bày các bit của các cổng vào ra của 8051. Xem ví dụ 4.2 về thao tác bit của các bit vào - ra. Lưu ý rằng trong ví dụ 4.2 các bit không dùng đến là không bị ảnh hưởng. Đây là khả năng đánh địa chỉ theo bit của các cổng vào - ra và là một trong những điểm mạnh nhất của bộ vi điều khiển 8051.

Ví dụ 4.2: hãy viết chương trình thực hiện các công việc sau:

- Duy trì hiển thị bit P1.2 cho đến khi nó lên cao
- Khi P1.2 lên cao, hãy ghi giá trị 45H vào cổng P0
- Gửi một xung cao xuống thấp (H-to-L) tới P2.3

Lời giải:

```

          SET      P1.2           ; Tạo bit P1.2 là đầu vào
          MOV      A, #45H        ; Gán A = 45H
AGAIN:    JNB      P1.2, AGAIN    ; Thoát khi P1.2 = 1
          MOV      P0, A          ; Xuất A tới cổng P0
          SETB     P2.3           ; Đưa P2.3 lên cao
          CLR      P2.3          ; Tạo P2.3 xuống thấp để có xung H-T0-L

```

Trong chương trình này lệnh “JNB P1.2, AGCN” (JNB có nghĩa là nhảy nếu không bit) ở lại vòng lặp cho đến khi P1.2 chưa lên cao. Khi P1.2 lên cao nó thoát ra khỏi vòng lặp ghi giá trị 45H tới cổng P0 và tạo ra xung H-to-L bằng chuỗi các lệnh SETB và CLR.