

CHƯƠNG 8

Các lệnh một bit và lập trình

8.1 Lập trình với các lệnh một bit.

Trong hầu hết các bộ vi xử lý (BVXL) thì dữ liệu được truy cập theo từng byte. Trong các bộ vi xử lý địa chỉ theo byte này thì các nội dung của một thanh ghi, bộ nhớ RAM hay cổng đều phải được truy cập từng byte một. Hay nói cách khác, lượng dữ liệu tối thiểu có thể được truy cập là một byte. Ví dụ, trong bộ vi xử lý Pentium cổng vào/ ra (I/O) được định hướng theo byte, có nghĩa là để thay đổi một bit thì ta phải truy cập toàn bộ 8 bit. Trong khi đó có rất nhiều ứng dụng thì ta phải chỉ cần thay đổi giá trị của một bit chẳng hạn như là bật hoặc tắt một thiết bị. Do vậy khả năng đánh địa chỉ đến từng bit của 8051 rất thích hợp cho ứng dụng này. Khả năng truy cập đến từng bit một thay vì phải truy cập cả byte làm cho 8051 trở thành trong những bộ vi điều khiển (BVĐK) 8 bit mạnh nhất trên thị trường. Vậy những bộ phận nào của CPU, RAM, các thanh ghi, cổng I/O hoặc ROM là có thể đánh địa chỉ theo bit được. Vì ROM chỉ đơn giản dữ mã chương trình thực thi nên nó không cần khả năng đánh địa chỉ theo bit. Tất cả mọi mã lệnh đều định hướng theo byte chỉ có các thanh ghi, RAM và các cổng I/O là cần được đánh địa chỉ theo bit. Trong 8051 thì rất nhiều vị trí của RAM trong một số thanh ghi và tất cả các cổng I/O là có thể đánh địa chỉ theo từng bit. Dưới đây ta chỉ đi sâu vào từng phần một.

8.1.1 Các lệnh một bit.

Các lệnh dùng các phép tính một bit được cho ở bảng 8.1. Trong phần này chúng ta làm về các lệnh này và đưa ra nhiều ví dụ về cách sử dụng chúng, các lệnh một bit khác mà chỉ liên quan đến cờ nhớ CY (Carry Flag) sẽ làm ở mục khác.

Bảng 8.1: Các lệnh một bit của 8051

Lệnh	Chức năng
SETB bit	Thiết lập bit (bit bằng 1)
CLR bit	Xoá bit về không (bit = 0)
CPL bit	Bù bit (bit = NOT bit)
JB bit, đích	Nhảy về đích nếu bit = 1
JNB bit, đích	Nhảy về đích nếu bit = 0
JBC bit, đích	Nhảy về đích nếu bit = 1 và sau đó xoá bit

8.1.2 Các cổng I/O và khả năng đánh địa chỉ theo bit.

Bộ vi điều khiển 8051 có bốn cổng I/O 8 bit là P0, P1, P2 và P3. Chúng ta có thể truy cập toàn bộ 8 bit hoặc theo một bit bất kỳ mà không làm thay đổi các bit khác còn lại. Khi truy cập một cổng theo từng bit, chúng ta sử dụng các cú pháp “SETB Y, X” với X là số của cổng 0, 1, 2 hoặc 3, còn Y là vị trí bit từ 0 đến 7 đối với các bit dữ liệu do đến 7. Ví dụ “SETB P1.5” là thiết lập bit cao số 5 của cổng 1. Hãy nhớ rằng do là bit có nghĩa thấp nhất LSB và D7 là bit có nghĩa là cao nhất MSB. Xem ví dụ 8.1.

Ví dụ 8.1: Viết các chương trình sau:

- Tạo một sóng vuông (hàm xung vuông) với độ đầy xung 50% trên bit 0 của cổng 1.
- Tạo một hàm xung vuông với 66% độ đầy xung trên bit 3 của cổng 1.

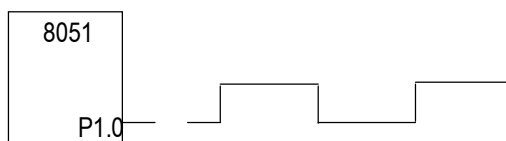
Lời giải:

- Hàm xung vuông với độ đầy xung 50% có nghĩa là trạng thái “bật” và “tắt” (hoặc phân cao và thấp của xung) có cùng độ dài. Do vậy ta chốt P1.0 với thời gian giữ chậm giữa các trạng thái.

```

HERE: SETB P1.0      ;Thiết lập bit 0 cổng 1 lên 1.
      LCALL DELAY    ;Gọi chương trình con giữ chậm DELAY
      CLR P1.0       ;P1.0 = 0
      SJMP HERE      ;Tiếp tục thực hiện nó.
                        Có thể viết chương trình này theo cách khác:
HERE: CPL P1.0       ;Bù bit 0 của cổng 1.
      LCALL DELAY    ;Gọi chương trình con giữ chậm DELAY
      SJMP HERE      ;Tiếp tục thực hiện nó.

```

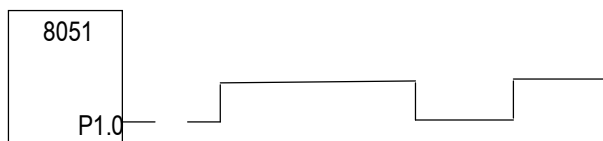


b) Hàm xung vuông với độ đầy xung 66% có nghĩa là trạng thái “bật” có độ dài gấp đôi trạng thái “tắt”.

```

BACK: SETB P1.3      ;Thiết lập bit 3 cổng 1 lên 1.
      LCALL DELAY    ;Gọi chương trình con DELAY
      LCALL DELAY    ;Gọi chương trình con DELAY lần nữa.
      CLR P1.3       ;Xóa bit 3 của cổng 1 và 0.
      LCALL DELAY    ;Gọi chương trình con DELAY
      SJMP BACK      ;Tiếp tục thực hiện nó.

```



Lưu ý rằng, khi mã “P1.0” được hợp dịch nó trở thành “SETB 90H” vì P1.0 có địa chỉ trong RAM là 90h. Từ hình vẽ 8.1 ta thấy rằng các địa chỉ bit cho P0 là 80H đến 87H và cho P là 90H đến 97H v.v... Hình 8.1 cũng chỉ ra tất cả các thanh ghi có khả năng đánh địa chỉ theo bit.

Bảng 8.2: Khả năng đánh địa chỉ theo bit của các cổng.

P0	P1	P2	P3	Port's 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 8.2:

Đối với các lệnh dưới đây thì trạng thái của bit nào của SFR sẽ bị tác động (hãy sử dụng hình 8.1).

- a) SETB 86H, b) CLR 87H, c) SETB 92H
b) SETB DA7H, e) CLR 0F2H, f) SETB OE7H

Lời giải

- a) SETB 86H là dành cho SETB P0.6
b) CLR 87H là dành cho CLR P0.7
c) SETB 92H là dành cho SETB P1.2

d) SETB	0A7H là dành cho SETB P2.7
e) CLR	0F2H là dành cho CLR D2 của thanh ghi B
f) SETB	0E7H là dành cho SETB ACC.7 (bit D7 của thanh ghi A)

8.1.3 Kiểm tra một bit đầu vào.

Lệnh JNB (nhảy nếu bit = 0) và JB (nhảy nếu bit bằng 1) cũng là các phép thao tác đơn bit được sử dụng rộng rãi. Chúng cho phép ta hiển thị một bit và thực hiện quyết định phụ thuộc vào việc liệu nó là 0 hay là 1.

Ví dụ 8.3: Giả sử bit P2.3 là một đầu vào và biểu diễn điều kiện của một lò. Nếu nó bật lên 1 thì có nghĩa là lò nóng. Hãy hiển thị liên tục, mỗi khi nó lên cao thì hãy gửi một xung cao-xuống-thấp (Aigh-to-low) đến cổng P1.5 để bật còi báo.

Lời giải:

```
HERE:  JNB  P2.3, HERE    ; Duy trì hiển thị cao.
        SETB P1.5        ; Thiết lập P1.5 = 1
        CLR  P1.5        ; Thực hiện chuyển xung từ cao-xuống-thấp
```

Các lệnh JNB và JB có thể được dùng đối với các bit bất kỳ của các cổng I/O 0, 1, 2 và 3 vì tất cả các cổng này đều có khả năng đánh địa chỉ theo bit. Tuy nhiên, cổng 3 hầu như để dùng cho các tín hiệu ngắt và truyền thông nối tiếp và thông thường không dùng cho bất cứ vào/ ra theo bit hoặc theo byte nào. Điều này sẽ được bàn ở chương 10 và 11.

8.1.4 Các thanh ghi và khả năng đánh địa chỉ theo bit.

Trong tất cả các cổng I/O đều có khả năng đánh địa chỉ theo bit thì các thanh ghi lại không được như vậy. Ta có thể nhìn thấy điều đó từ hình 8.1: Chỉ thanh ghi B, PSW, IP, IE, ACC, SCON và TCON là có thể đánh địa chỉ theo bit, ở đây ta sẽ tập trung vào các thanh ghi A, B và PSW còn các thanh ghi khác sẽ đề cập ở các chương sau. Từ hình 8.1 hãy để ý rằng cổng PO được gán địa chỉ bit 80H-87H. Còn đại chỉ bit 88-8FH được gán cho thanh ghi TCON.

Cuối cùng địa chỉ bit F0-F7H được gán cho thanh ghi B. Xét ví dụ 8.4 và 8.5 về việc sử dụng các thanh ghi này với khả năng đánh địa chỉ theo bit.

Byte address	Bit address								
FF									
F0	E7	F6	F5	F4	F3	F2	F1	F0	B
E0	E7	E6	E5	E4	E3	E2	E1	E0	ACC
D0	D7	D6	D5	D4	D3	D2	D1	D0	PSW
B8	--	--	--	BC	BB	BA	B9	B8	IP
B0	B7	B6	B5	B4	B3	B2	B1	B0	F3
A8	AF	--	--	AC	AB	AA	A9	A8	IE
A0	A7	A6	A5	A4	A3	A2	A1	A0	P2
99	not bit addressable								SBUF
98	9F	9E	9D	9C	9B	9A	99	99	SCON
90	97	96	95	94	93	92	91	90	P1
8D	not bit addressable								TH1
8C	not bit addressable								TH0
8B	not bit addressable								TL1
8A	not bit addressable								TL0
89	not bit addressable								TMOD
88	8F	8E	8D	8C	8B	8A	89	88	TCON
87	not bit addressable								PCON
83	not bit addressable								DPH
82	not bit addressable								DPL
81	not bit addressable								SP
80	87	86	85	84	83	82	81	80	P0
Special Function Registers									

Hình 8.1: Địa chỉ theo Byte và bit của bộ nhớ RAM các thanh ghi chức năng đặc biệt.

Ví dụ 8.4: Hãy viết chương trình để kiểm tra xem thanh ghi tích lũy có chứa một số chẵn không? Nếu có thì chia nó cho 2, nếu không thì hãy làm chẵn nó và sau đó chia nó cho 2.

Lời giải:

```

MOV B, #2          ; Gán B = 2
JNB ACC 0, YES     ; DO của thanh ghi A có bằng 0?
JNC A              ; Nếu có thì nhảy về YES
YES: DIX AB        ; Nếu là số lẻ thì tăng lên 1 để thành chẵn
                  ; Chia A/B

```

Ví dụ 8.5: Hãy viết đoạn chương trình để kiểm tra xem các bit 0 và 5 của thanh ghi B có giá trị cao không? Nếu không phải thì đặt chúng lên 1 và lưu vào thanh ghi bộ.

Lời giải:

```
JNB  OFOH, NEXT-1      ; Nhảy về NEXT-1 nếu B.0 = 0
SETB OFOH               ; Đặt B.0 = 1
NEXT-1: JNB  OF5H, NEXT-2 ; Nhảy về NEXT-2 nếu B.5 = 0
SETB OF5H               ; Đặt B.5 = 1
NEXT-2: MOV   RO, B      ; Chuyển thanh ghi B
```

CY	AC	--	RS1	RS0	OV	--	P
RS1	RS0	Register Bank				Address	
0	0	0				00H - 07H	
0	1	1				08H - 0FH	
1	0	2				10H - 17H	
1	1	3				18H - 1FH	

Hình 8.2: Các bit của thanh ghi PSW.

Như đã nói ở chương 2, trong thanh ghi PSW có hai bit dành riêng để chọn các bảng thanh ghi. Khi RESET thì bằng 0 được chọn, chúng ta có thể chọn các bảng bất kỳ khác bằng cách sử dụng khả năng đánh địa chỉ theo bit của PSW.

Ví dụ 8.6: Hãy viết chương trình để lưu thanh ghi tích lũy vào R7 của bảng 2.

Lời giải:

```
CLR     PSW.3
SETB    PSW.4
MOV      R7,A
```

Ví dụ 8.7: Trong khi có hai lệnh JNC và JC để kiểm tra bit cờ nhớ CY thì lại không có các lệnh cho bit cờ tràn (OV) làm thế nào để ta có thể viết mã kiểm tra OV.

Lời giải: Cờ OV là bit PSW.2 của thanh ghi PSW. PSW là thanh ghi có thể đánh địa chỉ theo bit, do vậy ta có thể sử dụng lệnh sau để kiểm tra cờ OV:

```
JB      PSW.2, TARGET    ; Nhảy về TARGET nếu OV = 1
```

8.15 Vùng nhớ RAM có thể đánh địa chỉ theo bit.

Trong 128 byte RAM trong của 8051 thì chỉ có 16 byte của nó là có thể đánh địa chỉ theo bit được. Phần còn lại được định dạng byte. Các vùng RAM có thể đánh địa chỉ theo bit là 20H đến 2FH. Với 16 byte này của RAM có thể cung cấp khả năng đánh địa chỉ theo bit là 128 bit, vì $16 \times 8 = 128$. Chúng được đánh địa chỉ từ 0 đến 127. Do vậy, những địa chỉ bit từ 0 đến 7 dành cho byte đầu tiên, vị trí RAM trong 20H và các bit từ 8 đến 0FH là địa chỉ bit của byte thứ hai của vị trí RAM trong 21H v.v... Byte cuối cùng của 2FH có địa chỉ bit từ 78H đến 7FH (xem hình 8.3). Lưu ý rằng các vị trí RAM trong 20H đến 2FH vừa có thể đánh địa chỉ theo byte vừa có thể đánh địa chỉ theo bit.

Để ý từ hình 8.3 và 8.1 ta thấy rằng các địa chỉ bit 00 - 7FH thuộc về các địa chỉ byte của RAM từ 20 - 2FH và các địa chỉ bit từ 80 đến F7H thuộc các thanh ghi đặc biệt SFR, các cổng P0, P1, v.v...

Ví dụ 8.8: Hãy kiểm tra xem các bit sau đây thuộc byte nào? Hãy cho địa chỉ của byte RAM ở dạng Hex.

- a) SETB 42H ; Set bit 42H to 1
- b) CLR 67H ; Clear bit 67
- c) CLR 0FH ; Clear bit 0FH
- d) SETB 28H ; Set bit 28H to 1

e) CLR 12 ; Clear bit 12 (decimal)
f) SETB 05

Lời giải:

- a) Địa chỉ bit 42H của RAM thuộc bit D2 của vị trí RAM 28H.
- b) Địa chỉ bit 67H của RAM thuộc bit D7 của vị trí RAM 20H.
- c) Địa chỉ bit 0FH của RAM thuộc bit D7 của vị trí RAM 21H.
- d) Địa chỉ bit 28H của RAM thuộc bit D0 của vị trí RAM 25H.
- e) Địa chỉ bit 12H của RAM thuộc bit D4 của vị trí RAM 21H.
- f) Địa chỉ bit 05H của RAM thuộc bit D5 của vị trí RAM 20H.

Ví dụ 8.9: Trạng thái của các bit P1.2 và P1.3 của cổng vào/ra P1 phải được lưu cất trước khi chúng được thay đổi. Hãy viết chương trình để lưu trạng thái của P1.2 vào vị trí bit 06 và trạng thái P1.3 vào vị trí bit 07.

Lời giải:

```
CLR      06      ;Xoá địa chỉ bit 06
CLR      07      ; Xoá địa chỉ bit 07
JNB      P1.2, OVER ;Kiểm tra bit P1.2 nhảy về OVER nếu P1.2 = 0
SETB     06      ; Nếu P1.2 thì thiết lập vị trí bit 06 = 0
OVER:    JNB      P1.3, NEXT ;Kiểm tra bit P1.3 nhảy về NEXT nếu nó = 0
SETB     07      ;Nếu P1.3 = 1 thì thiết lập vị trí bit 07 = 1
NEXT:    ....
```

Các câu hỏi ôn luyện:

1. Tất cả các cổng I/O của 8051 đều có khả năng đánh địa chỉ theo bit? (đúng sai)
2. Tất cả mọi thanh ghi của 8051 đều có khả năng đánh địa chỉ theo bit? (đúng sai)
3. Tất cả các vị trí RAM của 8051 đều có khả năng đánh địa chỉ theo bit? (đúng sai)
4. Hãy chỉ ra những thanh ghi nào sau đây có khả năng đánh địa chỉ theo bit:
a) A, b) B, (c) R4 (d) PSW (e) R7
5. Trong 128 byte RAM của 8051 những byte nào có khả năng đánh địa chỉ theo bit. Hãy liệt kê chúng.
6. Làm thế nào để có thể kiểm tra xem bit D0 của R3 là giá trị cao hay thấp.
7. Hãy tìm xem các bit dấu thuộc những byte nào? Hãy cho địa chỉ của các byte RAM theo số Hex:
a) SETB 20 b) CLR 32 c) SETB 12H
d) SETB 95 e) SETB 0ETB 12H
8. Các địa chỉ bit 00 - 7FH và 80 - F7H thuộc các vị trí nhớ nào?
9. Các cổng P0, P1, P2 và P3 là một bộ phận của SFR? (đúng sai)
10. Thanh ghi TCON có thể đánh địa chỉ theo bit (đúng sai)

8.2 Các phép toán một bit với cờ nhớ CY.

Ngoài một thực tế là cờ nhớ CY được thay đổi bởi các lệnh lô-gíc và số học thì trong 8051 còn có một số lệnh mà có thể thao tác trực tiếp cờ nhớ CY. Các lệnh này được cho trong bảng 8.3.

Trong các lệnh được chỉ ra sau trong bảng 8.3 thì chúng ta đã trình bày công dụng của lệnh JNC, CLR và SETB trong nhiều ví dụ trong một số chương trước đây. Dưới đây ta tiếp tục làm quen với một số ví dụ về cách sử dụng một số lệnh khác từ bảng 8.3.

Một số lệnh cho trong bảng 8.3 làm việc với các phép toán lô-gíc AND và OR. Các ví dụ ở mục này sẽ chỉ ra cách sử dụng chúng như thế nào?

Ở chương tiếp theo chúng ta sẽ chỉ ra nhiều ví dụ hơn về việc sử dụng của các lệnh đơn trong phạm vi các ứng dụng thực tế.

Bảng 8.3: Các lệnh liên quan đến cờ nhớ CY

Lệnh	chức năng
SETB C	Thực hiện (tạo) CY = 1
CLR C	Xoá bit nhớ CY = 0
CPL C	Bù bit nhớ
MOV b, C	Sao chép trạng thái bit nhớ vào vị trí bit b = CY
MOV C, b	Sao chép bit b vào trạng thái bit nhớ CY = b
JNC đích	Nhảy tới đích nếu CY = 0
JC đích	Nhảy tới đích nếu CY = 1
ANL C, bit	Thực hiện phép AND với bit b và lưu vào CY
ANL C, /bit	Thực hiện phép AND với bit đảo và lưu vào CY
ORL C, bit	Thực hiện phép OR với bit và lưu vào CY
ORL C, /bit	Thực hiện phép OR với bit đảo và lưu vào CY

Ví dụ 8.10: Hãy viết một chương trình để lưu cất trạng thái của các bit P1.2 và P1.3 vào vị trí nhớ tương ứng trong RAM 6 và 7.

Lời giải:

```

MOV C, P1.2    ; Lưu trạng thái P1.2 vào CY.
MOV 06, C      ; Lưu trạng thái CY vào bit 6 của RAM
MOV C, P1.3    ; Lưu trạng thái P1.2 vào CY
MOV 07, C      ; Lưu trạng thái CY vào vị trí RAM 07

```

Ví dụ 8.11:

Giả sử vị trí nhớ 12H trong RAM giữ trạng thái của việc có điện thoại hay không. Nếu nó ở trạng thái cao có nghĩa là đã có một cuộc gọi mới vì nó được kiểm tra lần cuối. Hãy viết một chương trình để hiển thị “có lời nhắn mới” (“New Message”) trên màn hình LCD nếu bit 12H của RAM có giá trị cao. Nếu nó có giá trị thấp thì LCD hiển thị “không có lời nhắn mới” (“No New Message”).

Lời giải:

```

MOV C, 12H      ; Sao trạng thái bit 12H của RAM vào CY
JNC NO          ; Kiểm tra xem cờ CY có giá trị cao không.
MOV DPTR, # 400H ; Nếu nó nạp địa chỉ của lời nhắn.
LCAL DISPLAY    ; Hiển thị lời nhắn.
SJMP NEXT       ; Thoát
NO:  MOV DSTR, #420H ; Nạp địa chỉ không có lời nhắn.
     LCAL DISPLAY    ; Hiển thị nó.
EXIT:           ; Thoát
; _____ data to be displayed on LCD

ORG 400H
YES-MG: DB "NEW Message"
ORG 420H
NO-MG:  DB "No New Message"

```

Ví dụ 8.12:

Giả sử rằng bit P2.2 được dùng để kiểm tra đèn ngoài và bit P2.5 dùng để kiểm tra đèn trong của một toà nhà. Hãy trình bày làm thế nào để bật đèn ngoài và tắt đèn trong nhà.

Lời giải:

```
SETB  C           ; Đặt CY = 1
ORL   C, P2.2, C  ; Thực hiện phép OR với CY
MOV   P2.2, C     ; Bật đèn nếu nó chưa bật.
CLR   C           ; Xoá CY = 0
ANL   C, P2.5     ; CY = (P2.5 AND CY)
MOV   P2.5, C     ; Tắt nó nếu nó chưa tắt.
```

Câu hỏi ôn luyện:

1. Tìm trạng thái của cờ CY sau đoạn mã dưới đây:

a) CLR	A	b) CLR	C	c) CLR	C
ADD	A, #OFFH	JNC	OVER	JC	OVER
JWC	OVER	SETB	C	CPL	C
CPL	C	OVER:	...	OVER:	...
OVER:	...				

2. Hãy trình bày cách làm thế nào để lưu trạng thái bit P2.7 vào vị trí bit 31 của RAM.

3. Hãy trình bày các chuyển trạng thái bit 09 của RAM đến bit P1.4.

8.3 Đọc các chân đầu vào thông qua chốt cổng.

Trong việc đọc cổng thì một số lệnh đọc trạng thái của các chân cổng, còn một số lệnh khác thì đọc một số trạng thái của chốt cổng trong. Do vậy, khi đọc các cổng thì có hai khả năng:

1. Đọc trạng thái của chân vào.
2. Đọc chốt trong của cổng ra.

Chúng ta phải phân biệt giữa hai dạng lệnh này vì sự lẫn lộn giữa chúng là nguyên nhân chính của các lỗi trong lập trình cho 8051, đặc biệt khi đã kết nối với phần cứng bên ngoài. Trong phần này ta bàn về sơ qua các lệnh này. Tuy nhiên, độc giả phải nghiên cứu và hiểu về các nội dung của chủ đề này và về hoạt động bên trong của các cổng được cho trong phụ lục Appendix C2.

8.3.1 Các lệnh đọc cổng vào.

Như đã nói ở chương 4 thì để biến một bit bất kỳ của cổng 8051 nào đó thành một cổng đầu vào, chúng ta phải ghi (lô-gíc cao) vào bit đó. Ssu khi cấu hình các bit của cổng là đầu vào, ta có thể sử dụng những lệnh nhất định để nhận dữ liệu ngoài trên các chân vào trong CPU. Bảng 8.4 là những lệnh nói trên.

Bảng 8.4: Các lệnh đọc một cổng vào.

Giả lệnh	Ví dụ	Mô tả
MOV A, PX	MOV A, P2	Chuyển dữ liệu ở chân P2 vào ACC
JNB PX.Y, ...	JNB P2.1, đích	Nhảy tới đích nếu, chân P2.1 = 0
JB PX.Y,	JB P1.3, đích	Nhảy đích nếu, chân P1.3 = 1
MOV C, PX.Y	MOV C, P2.4	Sao trạng thái chân P2.4 vào CY

8.3.2 Đọc chốt cho cổng đầu ra.

Một số lệnh nội dung của một chốt cổng trong thay cho việc đọc trạng thái của một chân ngoài. Bảng 8.5 cung cấp danh sách những lệnh này. Ví dụ, xét lệnh “ANL P1, A”. Trình tự thao tác được thực hiện bởi lệnh này như sau:

1. Nó đã chốt trong của một cổng và chuyển dữ liệu đó vào trong CPU.
2. Dữ liệu này được AND với nội dung của thanh ghi A.
3. Kết quả được ghi ngược lại ra chốt cổng.
4. Dữ liệu tại chân cổng được thay đổi và có cùng giá trị như chốt cổng.

Từ những bàn luận trên ta kết luận rằng, các lệnh đọc chốt cổng thường đọc một giá trị, thực hiện một phép tính (và có thể thay đổi nó) sau đó ghi ngược lại ra chốt cổng. Điều này thường được gọi “Đọc-sửa-ghi”, (“Read-Modify-Write”). Bảng 8.5 liệt kê các lệnh đọc-sửa-ghi sử dụng cổng như là toán hạng đích hay nói cách khác, chúng ta chỉ được dùng cho các cổng được cấu hình như các cổng ra.

Bảng 8.5: Các lệnh đọc một chốt (Đọc-sửa-ghi).

giả lệnh	Ví dụ
ANL PX	ANL P1, A
ORL PX	ORL P2, A
XRL PX	XRL P0, A
JBC PX.Y, đích	JBC P1.1, đích
CPL PX	CPL P1.2
INC PX	INC P1
DEC PX	DEC P2
DJN2 PX.Y, đích	DJN2 P1, đích
MOV PX.Y, C	MOV P1.2, C
CLR PX.Y	CLR P2.3
SETB PX.Y	SETB P2.3

Lưu ý: Chúng ta nên nghiên cứu phần C2 của phụ lục Appendix C nếu ta nối phần cứng ngoài vào hệ 8051 của mình. Thực hiện sai các chỉ dẫn hoặc nối sai các chân có thể làm hỏng các cổng của hệ 8051.

8.4 Tóm lược.

Chương này đã mô tả một trong các đặc tính mạnh nhất của 8051 là phép toán một bit. Các phép toán một bit này cho phép lập trình viên thiết lập, xoá, di chuyển và bù các bit riêng rẽ của các cổng, bộ nhớ hoặc các thanh ghi.

Ngoài ra có một số lệnh cho phép thao tác trực tiếp với cờ nhớ CY. Chúng ta cũng đã bàn về các lệnh đọc các chân cổng thông qua việc đọc chốt cổng.

8.5 Các câu hỏi kiểm tra.

1. Các lệnh “SETB A”, “CLR A”, “CPL A” đúng hay sai?
2. Các cổng vào/ ra nào và các thanh ghi nào có thể đánh địa chỉ theo bit.
3. Các lệnh dưới đây đúng hay sai? Đánh dấu lệnh đúng.

- | | |
|--------------|--------------|
| a) SETB P1 | e) SETB B4 |
| b) SETB P2.3 | f) CLR 80H |
| c) CLR ACC.5 | g) CLR PSW.3 |
| d) CRL 90H | h) CLR 87H |

4. Hãy viết chương trình tạo xung vuông với độ đầy xung 75%, 80% trên các chân P1.5 và P2.7 tương ứng.

5. Viết chương trình hiển thị P1.4 nếu nó có giá trị cao thì chương trình tạo ra một âm thanh (sóng dung vuông 50% độ đầy xung) trên chân P2.7.

6. Nhưng địa chỉ bit nào được gán cho các cổng P0, P1, P2 và P3 cho các thanh ghi PCON, A, B và PSW.

7. Những địa chỉ bit dưới đây thuộc về cổng hay thanh ghi nào?

- | | | | | |
|--------|--------|--------|--------|--------|
| a) 85H | b) 87H | c) 88H | d) 8DH | e) 93H |
| f) A5H | g) A7H | h) B3H | i) D4H | j) D8H |

8. Hãy viết chương trình lưu các thanh ghi A, B vào R3 và R5 bằng nhớ 2 tương ứng.
9. Cho một lệnh khác cho “CLR C”, so sánh chúng.
10. Làm thế nào để kiểm tra trạng thái các bit cờ OV, CY, P và AC. Hãy tìm địa chỉ bit của các cờ này.
11. Các cùng nhớ 128 byte của RAM thì những vùng nào là đánh địa chỉ theo bit được? Hãy đánh dấu chúng.
12. Các địa chỉ sau thuộc vùng RAM nào?
a) 05H b) 47 c) 18H d) 2DH e) 53H
g) 15H h) 67H h) 55H i) 14H k) 37FH
13. Các địa chỉ nhỏ hơn 80H được gán cho địa chỉ 20-2FH của RAM phải không? (Đúng/ sai).
14. Viết các lệnh để lưu cờ CY, AC, D vào vị trí bit 4, 16H và 12H tương ứng.
15. Viết chương trình kiểm tra D7 của thanh ghi A. Nếu D7 = 1 thì gửi thông báo sang LCD báo rằng ACC có một số âm.