

CHƯƠNG 13

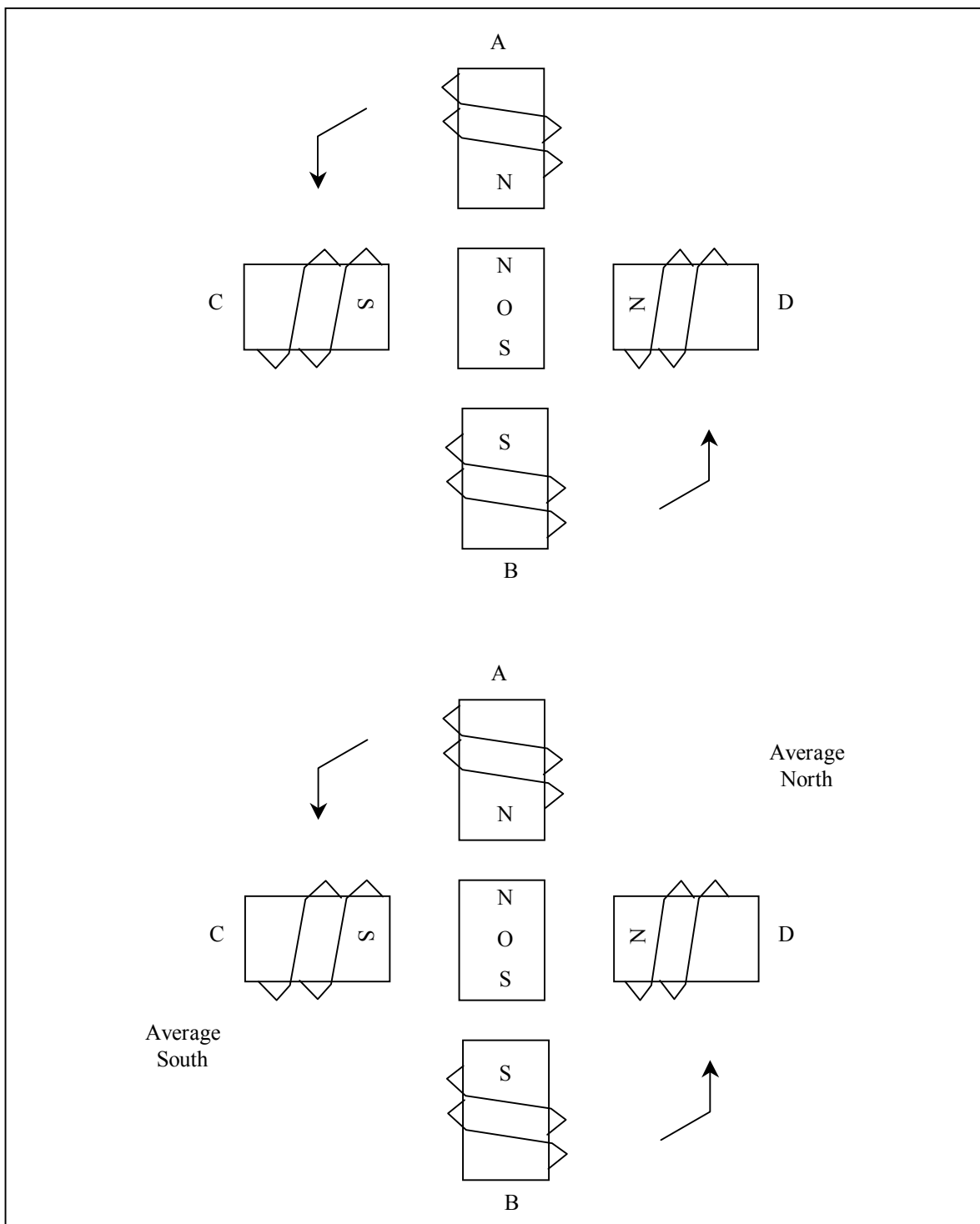
Phối ghép với thế giới kiểu II động cơ bước, bàn phím và các bộ DAC

13.1 Phối ghép với một động cơ bước.

Phần này bắt đầu với việc giới thiệu tổng quan về hoạt động của các động cơ bước. Sau đó chúng ta mô tả cách phối ghép một động cơ bước với bộ vi điều khiển 8051. Cuối cùng ta sử dụng các chương trình hợp ngữ để trình diễn điều khiển góc và hướng quay của động cơ bước.

13.1.1 Các động cơ bước.

Động cơ bước là một thiết bị sử dụng rộng rãi để chuyển các xung điện thành chuyển động cơ học. Trong các ứng dụng chẳng hạn như các bộ điều khiển đĩa, các máy in kim ma trận và các máy rô-bốt thì động cơ bước được dùng để điều khiển chuyển động. Mỗi động cơ bước đều có phần quay rôto là nam châm vĩnh cửu (cũng còn được gọi là trục dẫn - shaft) được bao bọc xung quanh là một **đứng** yên gọi stato (xem hình 131.1). Hầu hết các động cơ bước đều có chung có 4 stato mà các cuộn dây của chúng được bố trí theo cặp đối xứng với điểm giữa chung (xem hình 13.2), Kiểu động cơ bước này nhìn chung còn được coi như động cơ bước 4 pha. Điểm giữa cho phép một sự thay đổi của hướng dòng của một trong hai lõi khi một cuộn dây được nối đất tạo ra sự thay đổi cực của stato. Lưu ý rằng, trục của một động cơ truyền thống thì quay tự do, còn trục của động cơ bước thì chuyển động theo một độ tăng cố định lặp lại để cho phép ta chuyển dịch nó đến một vị trí chính xác. Chuyển động cố định lặp lại này có được là nhờ kết quả của lý thuyết từ trường cơ sở là các cực cùng dấu thì đẩy nhau và các cực khác dấu thì hút nhau. Hướng quay được xác định bởi từ trường của stato. Từ trường của stato được xác định bởi dòng chạy qua lõi cuộn dây. khi hướng của dòng thay đổi thì cực từ trường cũng thay đổi gây ra chuyển động ngược lại của động cơ (đảo chiều). Động cơ bước được nối ở đây có 6 đầu dây: 4 đầu của cuộn dây stato và hai đầu dây chung điểm giữa của các cặp dây. Khi chuỗi xung nguồn được cấp đến mỗi cuộn dây stato thì động cơ sẽ quay. Có một số chuỗi xung được sử dụng rộng rãi với cấp độ chính xác khác nhau. Bảng 13.1 trình bày chuỗi 4 bước thông thường.



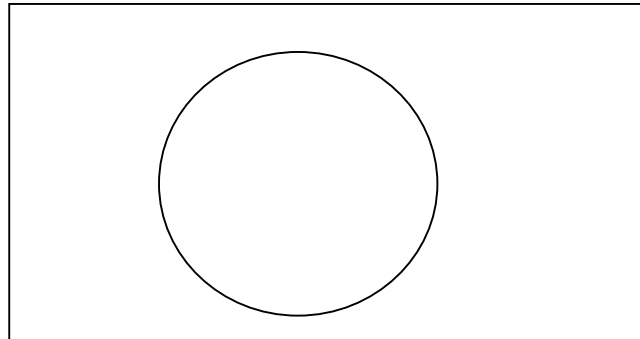
Hình 13.1: Căn chỉnh rôto.

Bảng 13.1: Chuỗi nguồn nuôi 4 bước thông thường.

Chiều kim đồng hồ ↓	Bước	Cuộn dây A	Cuộn dây B	Cuộn dây C	Cuộn dây D	↑ Chiều quay bộ đếm
	1	1	0	0	1	
	2	1	1	0	0	
	3	0	1	1	0	
	4	0	0	1	1	

Bảng13.2: Các góc bước của động cơ bước.

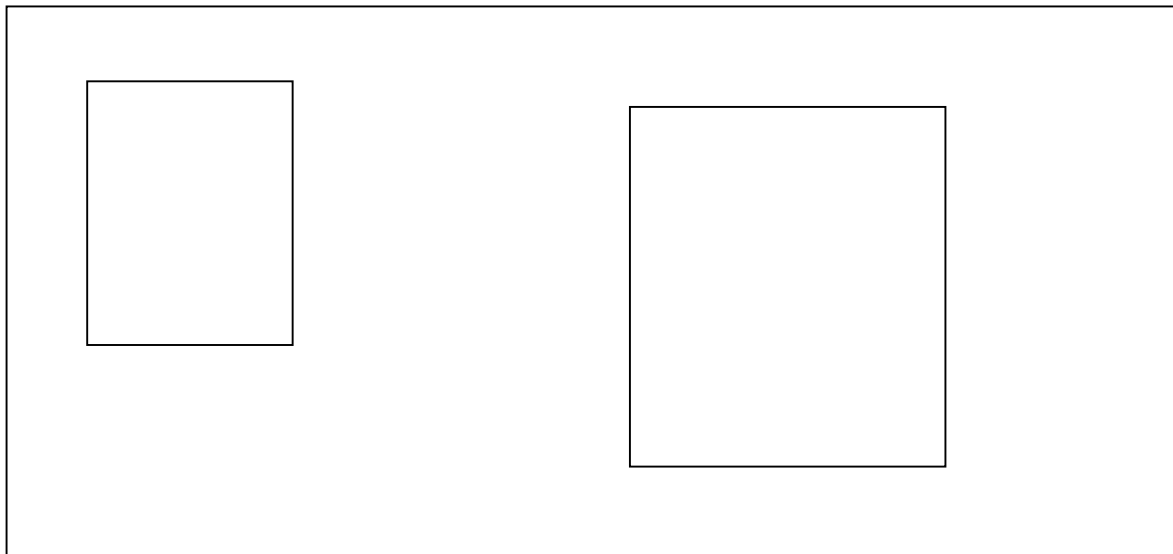
Góc bước	Số bước/ vòng
0.72	500
1.8	200
2.0	180
2.5	144
5.0	72
7.5	48
15	24



Bảng13.2: Các góc bước của động cơ bước.

Hình 13.2:

Hình 13.2: Bố trí các cuộn dây của stato.



Hình 13.3: Phối ghép 8051 với một động cơ bước.

Cần phải nhớ rằng mặc dù ta có thể bắt đầu với các chuỗi bất kỳ trong bảng 13.1. Nhưng khi đã bắt đầu thì ta phải tiếp tục với các chuỗi theo đúng thứ tự. Ví dụ ta bắt đầu bước thứ ba là chuỗi (0110) thì ta phải tiếp tục với chuỗi của bước 4 rồi sau đó 1, 2, 3 v.v...

Ví dụ 13.1:

Hãy mô tả kết nối 8051 với động cơ bước của hình 13.3 và viết một chương trình quay nó liên tục.

Lời giải:

Các bước dưới đây trình bày việc kết nối 8051 với động cơ bước và lập trình của nó.

1. Sử dụng một ôm kế để đo trở kháng của các đầu dây. Điều này xác định đầu chung (COM) nào được nối tới cuộn dây nào?
2. Các dây chung được nối tới đầu dương của nguồn cấp cho động cơ. Trong nhiều động cơ thì + 5V là đủ.
3. Bốn đầu củ cuộn dây stato được điều khiển bởi 4 bit của cổng P1 trong 8051 (P1.0 - P1.3). Tuy nhiên, vì 8051 không đủ dòng để điều khiển các cuộn dây động cơ bước nên ta phải sử dụng một bộ điều khiển chẳng hạn như ULN2003 để cấp năng lượng cho stato. Thay cho ULN2003 ta có thể sử dụng các bóng bán dẫn làm các bộ điều khiển như chỉ ra trên hình 13.4. Tuy nhiên ta để ý rằng, nếu các bóng bán dẫn được sử dụng như các bộ điều khiển chúng ta cũng phải sử dụng các đi ốt để ngăn dòng cảm ứng được tạo ra khi tắt cuộn dây. Một lý do mà ULN2003 được ưu chuộng hơn các bóng bán dẫn như các bộ điều khiển là nó có đi ốt bên trong để ngăn cảm ứng điện từ ngược.

```

BACK:      MOV      A, # 66H          ; Nạp chuỗi xung bước
           MOV      P1, A           ; Xuất chuỗi xung đến động cơ
           RR        A              ; Quay theo chiều kim đồng hồ
           ACALL     DELAY          ; Chờ
           SJMP      BACK          ; Tiếp tục chạy
           -----

DELAY
           MOV      R2, # 100
H1:        MOV      R3, # 255
H2:        DJNZ     R3, H2
           DJNZ     R2, H1
           RET

```

Hãy thay đổi giá trị của DELAY để đặt tốc độ quay. Ta có thể sử dụng lệnh đơn bit SETB và CLR thay cho lệnh RRA để tạo ra chuỗi xung.

13.1.2 Góc bước (Step Angle).

Vậy mỗi bước có độ dịch chuyển là bao nhiêu? Điều này phụ thuộc vào cấu trúc bên trong của động cơ, đặc biệt là số răng của stato và rô to. Góc bước là độ quay nhỏ nhất của một bước. Các động cơ khác nhau có các góc bước khác nhau. Bảng 13.2 trình bày một số góc bước đối với các động cơ khác nhau. Bảng 13.2 có sử dụng thuật ngữ số bước trong một vòng (Steps per revolution). Đây là tổng số bước cần để quay hết một vòng 360^0 (chẳng hạn $180 \text{ bước} \times 2^0 = 360^0$).

Cần phải nói rằng dường như trái ngược với ấn tượng ban đầu. Một động cơ bước không cần nhiều đầu dây cho stato hơn để có các bước nhỏ hơn. Tất cả mọi động cơ bước được nói ở đây chỉ có 4 đầu dây cho cuộn dây stato và 2 đầu dây chung cho nút giữa. Mặc dù nhiều hãng sản xuất chỉ dành một đầu chung thay cho hai thì họ cũng vẫn phải có 4 đầu cuộn dây stato.

13.1.3 Quan hệ số bước trong giây và số vòng quay trong phút RPM.

Quan hệ giữa số vòng quay trong phút RPM (revolutions per minute), số bước trong vòng quay và số bước trong vòng giây là quan hệ thuộc về trực giác và nó được biểu diễn như sau:

$$\text{Số bước trong giây} = \frac{\text{RPM} \times \text{Số bước trong vòng quay}}{60}$$

13.1.4 Chuỗi xung bốn bước và số răng trên rô to.

Chuỗi xung chuyển mạch được trình bày trong bảng 13.1 được gọi là chuỗi chuyển mạch 4 bước bởi vì sau 4 bước thì hai cuộn dây giống nhau sẽ được bật “ON”. Vậy độ dịch chuyển của 4 bước này sẽ là bao nhiêu? Sau mỗi khi thực hiện 4 bước này thì rô to chỉ dịch được một bước răng. Do vậy, trong động cơ bước với 200 bước/ vòng thì rô to của nó có 50 răng vì $50 \times 4 = 200$ bước cần để quay hết một vòng. Điều này dẫn đến một kết luận là góc bước tối thiểu luôn là hàm của số răng trên rô to. Hay nói cách khác góc bước càng nhỏ thì rô to quay được càng nhiều răng. Hãy xét ví dụ 13.2.

Ví dụ 13.2:

Hãy tính số lần của chuỗi 4 bước trong bảng 13.1 phải cấp cho một động cơ bước để tạo ra một dịch chuyển 80° nếu động cơ góc bước là 2° .

Lời giải:

Một động cơ có góc bước là 2° thì phải có những đặc tính sau: góc bước 2° , số bước/ vòng là 180° , số răng của rô to là 45, độ dịch chuyển sau mỗi chuỗi 4 bước là 8° . Vậy để dịch chuyển 80° thì cần 40 chuỗi 4 bước vì $10 \times 4 \times 2 = 80$.

Nhìn vào ví dụ 13.2 thì có người sẽ hỏi vậy muốn dịch chuyển đi 45° thì làm thế nào khi góc bước là 2° . Muốn có độ phân giải nhỏ hơn thì tất cả mọi động cơ bước đều cho phép chuỗi chuyển mạch 8 bước, chuỗi 8 bước cũng còn được gọi chuỗi nửa bước (half - stepping), vì trong chuỗi 8 bước dưới đây thì mỗi bước là một nửa của góc bước bình thường. Ví dụ, một động cơ có góc bước là 2° có thể sử dụng góc bước 1° nếu áp dụng chuỗi ở bảng 13.3.

Bảng 13.3: Chuỗi xung 8 bước.

Bước	Cuộn A	Cuộn B	Cuộn C	Cuộn D
1	1	0	0	1
2	1	0	0	0
3	1	1	0	0
4	0	1	0	0
5	0	1	1	0
6	0	0	1	0
7	0	0	1	1
8	0	0	0	1

Chiều
kim
đồng
hồ
↓

↑
Chiều
quay
bộ
đếm

13.1.5 Tốc độ động cơ.

Tốc độ động cơ được đo bằng số bước trong một giây (bước/giây) là một hàm của tốc độ chuyển mạch. Để ý trong ví dụ 13.1 ta thấy rằng bằng việc thay đổi độ thời gian trễ ta có thể đạt được các tốc độ quay khác nhau.

13.1.6 Mô men giữ.

Dưới đây là một định nghĩa về mô men giữ:

Mô men giữ là lượng mô men ngoài cần thiết để làm quay trục động cơ từ vị trí giữ của nó với điều kiện trục động cơ đang đứng yên hoặc đang quay với tốc độ

RPM = 0. Đại lượng này được đo bằng tỷ lệ điện áp và dòng cấp đến động cơ. Đơn vị của mô men giữ là kilôgam - centimet (hay ounce - inch).

13.1.7 Chuỗi 4 bước điều khiển dạng sóng.

Ngoài các chuỗi 4 bước và 8 bước đã nói trên đây còn có một chuỗi khác được gọi là chuỗi 4 bước dạng sóng. Nó được trình bày trong bảng 13.4. Để ý 8 bước trong bảng 13.3 là một sự kết hợp đơn giản của các chuỗi 4 bước thường và chuỗi 4 bước điều khiển dạng sóng được cho ở bảng 13.1 và 13.4.

Chiều kim đồng hồ ↓	Bước	Cuộn dây A	Cuộn dây B	Cuộn dây C	Cuộn dây D	↑ Chiều quay bộ đếm
	1	1	0	0	0	
	2	0	1	0	0	
	3	0	0	1	0	
	4	0	0	0	1	

Hình 13.4: Sử dụng các bóng bán dẫn để điều khiển động cơ bước.

13.2 Phối ghép 8051 với bàn phím.

Các bàn phím và LCD là những thiết bị vào/ra được sử dụng rộng rãi nhất của 8051 và cần phải thấu hiểu một cách cơ bản về chúng. Ở phần này trước hết ta giới thiệu các kiến thức cơ bản về bàn phím với cơ cấu ấn phím và tách phím, sau đó giới thiệu về giao tiếp 8051 với bàn phím.

13.2.1 Phối ghép bàn phím với 8051.

Ở mức thấp nhất các bàn phím được tổ chức dưới dạng một ma trận các hàng và các cột. CPU truy cập cả hàng lẫn cột thông qua các cổng. Do vậy, với hai cổng 8 bit thì có thể nối tới một bàn phím 8×8 tới bộ vi xử lý. Khi một phím được ấn thì một hàng và một cột được tiếp xúc, ngoài ra không có sự tiếp xúc nào giữa các hàng và các cột. Trong các bàn phím máy tính IBM PC có một bộ vi điều khiển (bao gồm một bộ vi xử lý, bộ nhớ RAM và EPROM và một số cổng tất cả được bố trí trên một chip) chịu trách nhiệm phối ghép phần cứng và phần mềm của bàn phím. Trong những hệ thống như vậy, nó là chức năng của các chương trình được lưu trong EPROM của bộ vi điều khiển để quét liên tục các phím, xác định xem phím nào đã được kích hoạt và gửi nó đến bo mạch chính. Trong phần này nghiên cứu về cơ cấu 8051 quét và xác định phím.

13.2.2 Quét và xác định phím.

Hình 13.5 trình bày một ma trận 4×4 được nối tới hai cổng. Các hàng được nối tới một đầu ra và các cột được nối tới một cổng vào. Nếu không có phím nào được ấn thì việc đóng cổng vào sẽ hoàn toàn là 1 cho tất cả các cột vì tất cả được nối tới dương nguồn V_{CC} . Nếu tất cả các hàng được nối đất và một phím được ấn thì một trong các cột sẽ có giá trị 0 vì phím được ấn tạo đường xuống đất. Chức năng của bộ vi điều khiển là quét liên tục để phát hiện và xác định phím được ấn.

Hình 13.5

Hình 13.5: Nối ghép bàn phím ma trận tới các cổng.

13.2.3 Nối đất các hàng và đọc các cột.

Để phát hiện một phím được ấn thì bộ vi điều khiển nối đất tất cả các hàng bằng cách cấp 0 tới chốt đầu ra, sau đó nó đọc các hàng. Nếu dữ được đọc từ các

cột là $D3 - D0 = 1101$ thì không có phím nào được ấn và quá trình tiếp tục cho đến khi phát hiện một phím được ấn. Tuy nhiên, nếu một trong các bit cột có số 0 thì điều đó có nghĩa là việc ấn phím đã xảy ra. Ví dụ, nếu $D3 - D0 = 1101$ có nghĩa là một phím ở cột 1 được ấn. Sau khi một ấn phím được phát hiện, bộ vi điều khiển sẽ chạy quá trình xác định phím. Bắt đầu với hàng trên cùng, bộ vi điều khiển nối đất nó bằng cách chỉ cấp mức thấp tới chân $D0$, sau đó nó đọc các cột. Nếu dữ liệu đọc được là toàn số 1 thì không có phím nào của hàng này được ấn và quá trình này chuyển sang hàng kế tiếp. Nó nối đất hàng kế tiếp, đọc các cột và kiểm tra xem có số 0 nào không? Quá trình này tiếp tục cho đến khi xác định được hàng nào có phím ấn. Sau khi xác định được hàng có phím được ấn thì công việc tiếp theo là tìm ra phím ấn thuộc cột nào. Điều này thật là dễ dàng vì bộ vi điều khiển biết tại thời điểm bất kỳ hàng nào và cột nào được truy cập. Hãy xét ví dụ 13.3.

Ví dụ 13.3:

Từ hình 13.5 hãy xác định hàng và cột của phím được ấn cho các trường hợp sau đây:

- a) $D3 - D0 = 1110$ cho hàng và $D3 - D0 = 1011$ cho cột.
- b) $D3 - D0 = 1101$ cho hàng và $D3 - D0 = 0111$ cho cột.

Lời giải:

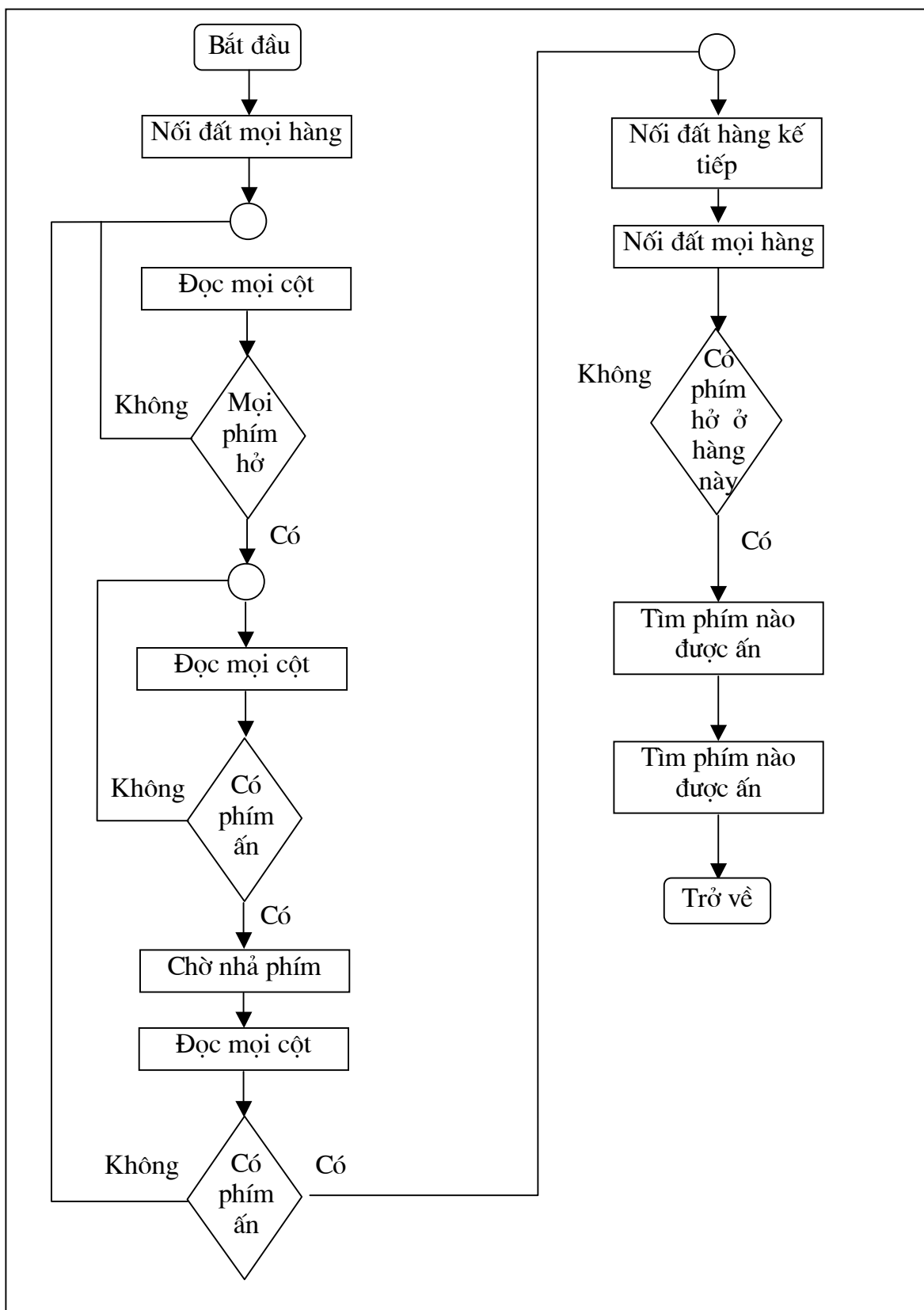
Từ hình 13.5 cột và hàng có thể được sử dụng xác định phím.

- a) Hàng thuộc $D0$ và cột thuộc $D2$, do vậy phím số 2 đã được ấn.
- b) Hàng thuộc $D1$ và cột thuộc $D3$, do vậy phím số 7 đã được ấn.

Chương trình 13.1 là chương trình hợp ngữ của 8051 để phát hiện và xác định sự kích hoạt phím. Trong chương trình này $P1$ và $P2$ được giả thiết là cổng ra và cổng vào tương ứng. Chương trình 13.1 đi qua 4 giai đoạn chính sau đây.

1. Khẳng định phím trước đó đã được nhả, các số không là đầu ra tới tất cả các hàng cùng một lúc và các cột được đọc và được kiểm tra chừng nào tất cả mọi cột đều cao. Khi tất cả các cột được phát hiện là đều cao thì chương trình chờ một thời gian ngắn trước khi nó chuyển sang giai đoạn kế tiếp để chờ một phím được ấn.
2. Để biết có một phím nào được ấn các cột được quét đi quét lại trong vòng vô tận cho đến khi có một cột có số 0. Hãy nhớ rằng các chốt đầu ra được nối tới các hàng vẫn có các số 0 ban đầu (được cấp ở giai đoạn 1) làm cho chúng được nối đất. Sau khi phát hiện ấn phím, nó đợi 20ms chờ cho phím nhả ra và sau đó quét lại các cột. Điều này phục vụ hai chức năng: a) nó đảm bảo rằng việc phát hiện ấn phím đầu tiên không bị sai do nhiễu và b) thời gian giữ chậm là 20ms ngăn ngừa việc ấn cùng một phím như là nhiều lần ấn. Nếu sau 20ms giữ chậm mà phím vẫn được ấn nó chuyển sang giai đoạn kế tiếp để phát hiện phím ấn thuộc hàng nào, nếu không nó quay trở vòng lặp để phát hiện có một phím ấn thật.
3. Để phát hiện ấn phím thuộc hàng, nó nối đất mỗi hàng tại một thời điểm, đọc các cột mỗi lần. Nếu nó phát hiện tất cả mọi cột đều cao, điều này có nghĩa là ấn phím không thuộc hàng đó, do vậy nó nối đất hàng kế tiếp và tiếp tục cho đến khi phát hiện ra hàng có phím ấn. Khi tìm hàng có phím ấn, nó thiết lập địa chỉ bắt đầu cho bảng trình bày giữ các mã quét (hoặc giá trị ASCII) cho hàng đó và chuyển sang giai đoạn kế tiếp để xác định phím.
4. Để xác định phím ấn, nó quay các bit cột, mỗi lần một bit vào cờ nhớ và kiểm tra xem nó có giá trị thấp không? Khi tìm ra số 0, nó kéo mã ASCII dành cho phím đó ra từ bảng trình bày. Nếu không tìm được số 0 thì nó tăng con trỏ để chỉ đến

phần tử kế tiếp của bảng trình bày. Hình 13.6 trình bày lưu đồ quá trình tìm phím ấn này.



Hình 13.6: Lưu đồ tìm phím ấn của chương trình 13.1.

Trong khi việc phát hiện ấn là chuẩn cho tất cả mọi bàn phím thì quá trình xác định phím nào được ấn lại không giống nhau. Phương phát sử dụng bảng trình bày được đưa ra trong chương trình 13.1 có thể được sửa đổi để làm việc với bất kỳ ma trận kích thước 8×8 nào. Hình 13.6 là lưu đồ thuật toán của chương trình 13.1 để quét và xác định phím ấn.

Có những chip IC chẳng hạn như MM74C924 của hãng National Semiconductor kết hợp việc quét và giải mã bàn phím tất cả vào một chip. Các chip như vậy sử dụng sự kết hợp các bộ đếm và các cổng lô gíc (không phải bộ vi điều khiển) để thực thi các khái niệm được trình bày trong chương trình 13.1 dưới đây.

Chương trình 13.1:

- ; Chương trình con bàn phím Keyboard này gửi mã ASCII
- ; Cha phím được ấn đến chân P0.1
- ; Các chân P1.0 – P1.3 được nối tới các hàng còn P2.0 – P2.3 tới các cột.

13.3 Phối ghép một DAC với 8051.

Phần này sẽ trình bày cách phối ghép một bộ biến đổi số tương tự DAC với 8051. Sau đó minh họa tạo một sóng hình sin trên máy hiện sóng sử dụng bộ DAC.

13.3.1 Bộ biến đổi số - tương tự DAC.

Bộ biến đổi - tương tự DAC là một thiết bị được sử dụng rộng rãi để chuyển đổi các xung số hoá về các tín hiệu tương tự. Trong phần này ta giới thiệu cơ sở phối ghép một bộ DAC với 8051.

Xem lại các kiến thức điện tử số ta thấy có hai cách tạo ra bộ DAC: Phương pháp trọng số nhị phân và phương trình thang $R/2R$. Nhiều bộ DAC dựa trên các mạch tổ hợp, bao gồm MC1408 (DAC808) được sử dụng trong phần này đều sử dụng phương pháp hình thang $R/2R$ vì nó có thể đạt độ chính xác cao hơn. Tiêu chuẩn đánh giá một bộ DAC đầu tiên là độ phân giải hàm của số đầu vào nhị phân. Các độ phân giải chúng là 8, 10 và 12 bit. Số các đầu vào bit dữ liệu quyết định độ phân giải của bộ DAC, vì số mức đầu ra tương tự bằng 2^n với n là đầu vào bit dữ liệu. Do vậy, một bộ DAC 8 bit như DAC808 chẳng hạn có 256 mức đầu ra điện áp (dòng điện) rời rạc. Tương tự như vậy, một bộ DAC 12 bit cho 4096 mức điện áp rời rạc. cũng có các bộ DAC 16 bit nhưng chúng rất đắt.

13.3.2 Bộ biến đổi DAC MC1408 (hay DAC808).

Trong bộ DAC808 các đầu vào số được chuyển đổi thành dòng (I_{out}) và việc nối một điện trở tới chân I_{out} ta chuyển kết quả thành điện áp. dòng tổng được cấp bởi chân I_{out} là một hàm số nhị phân ở các đầu vào D0 – D7 của DAC808 và tham chiếu I_{ref} như sau:

$$I_{out} = I_{ref} \left(\frac{D7}{2} + \frac{D6}{4} + \frac{D5}{8} + \frac{D4}{16} + \frac{D3}{32} + \frac{D2}{64} + \frac{D1}{128} + \frac{D0}{256} \right)$$

Trong đó D0 là bit thấp nhất LSB và D7 là bit cao nhất MSB đối với các đầu vào

