

NGÔ DIỄN TẬP

ĐO LƯỜNG VÀ ĐIỀU KHIỂN



BẢNG MÁY TÍNH



NHA XUẤT BẢN
KHOA HỌC VÀ KỸ THUẬT

Ngô Diên Tập

ĐO LƯỜNG VÀ ĐIỀU KHIỂN BẰNG MÁY TÍNH



NHÀ XUẤT BẢN KHOA HỌC VÀ KỸ THUẬT
HÀ NỘI - 1997

LỜI NÓI ĐẦU

Việc ứng dụng máy vi tính vào kỹ thuật đo lường và điều khiển đã đem lại những kết quả đầy tính ưu việt. Các thiết bị, hệ thống đo lường và điều khiển ghép nối với máy tính có độ chính xác cao, thời gian thu thập số liệu ngắn, nhưng còn đáng quan tâm hơn là mức độ tự động hoá trong việc thu thập và xử lý các kết quả đo, kể cả việc lập bảng thống kê cũng như in ra kết quả.

Cuốn sách này được biên soạn nhằm giới thiệu nguyên lý hoạt động của các mạch đo lường và điều khiển ghép nối với máy tính. Hàng trăm ví dụ minh hoạ sẽ giúp cho bạn đọc dễ dàng làm quen dần với các mạch điện từ đơn giản đến phức tạp, từ việc tự tiến hành lắp ráp đến có thể làm chủ những hệ thống đo lường và điều khiển phức tạp.

Để các hệ thống đo lường và điều khiển ghép nối với máy tính hoạt động được thì ngoài phần mạch điện còn cần có một chương trình được nạp vào máy tính. Các chương trình dùng cho các hệ thống đo lường và điều khiển sẽ được trình bày riêng trong một cuốn sách khác; còn trong cuốn sách này các chương trình minh hoạ được giới thiệu đều viết bằng QBasic. Theo cách nghĩ của chúng tôi, với ngôn ngữ này bạn đọc sẽ dễ dàng làm quen, nhưng đồng thời cũng đủ khả năng để minh hoạ sự hoạt động của mạch điện. Các chương trình minh hoạ này được giới thiệu trước hết nhằm mục đích đọc hiểu, nên để tránh sự hiểu lầm, những lời chú thích trong chương trình được viết bằng tiếng Việt có dấu; và chúng ta đều biết BASIC không thể hiểu được các chữ có dấu này; nên khi gõ chương trình vào máy tính ta phải bỏ đi các dấu. Sau khi đã làm chủ được hệ thống, ta có thể dùng các công cụ lập trình khác (chẳng hạn C++ ; DELPHI...) để tạo ra những sản phẩm phần mềm có chất lượng và hình thức trình bày trên màn hình tốt hơn.

Mặc dù đã dành thời gian thích đáng nhưng không tránh khỏi một số lỗi vẫn có thể còn sót lại trong cuốn sách. Rất mong bạn đọc gần xa chỉ dẫn cho. Thư từ góp ý xin gửi về Nhà xuất bản Khoa học và Kỹ thuật, 70 Phố Trần Hưng Đạo, Hà nội.

Tác giả

MỤC LỤC

	Trang
□ Lời nói đầu	3
□ Mục lục	4
□ Chương 1. Giao diện của máy tính trong đo lường và điều khiển	7
● 1.1 Cổng ghép nối với máy in	7
- 1.1.1 Vài nét cơ bản về cổng ghép nối với máy in	7
- 1.1.2 Trao đổi với các đường dẫn tín hiệu	9
- 1.1.3 Chương trình kiểm tra cổng máy in	12
- 1.1.4 Môđun cơ sở vào/ ra dùng cho cổng máy in	18
● 1.2 Cổng nối tiếp RS 232	22
- 1.2.1 Vài nét cơ bản về cổng nối tiếp	22
- 1.2.2 Sự trao đổi với đường dẫn dữ liệu	25
- 1.2.3 Bộ gửi nối tiếp	30
- 1.2.4 Bộ phát nhận không đồng bộ vạn năng CDP 6402	32
- 1.2.5 Môđun cơ sở vào/ ra 8 bit không có nguồn phụ	37
- 1.2.6 Môđun cơ sở vào/ ra 8 bit với MAX 232	41
- 1.2.7 Môđun cơ sở vào/ ra 8 bit cách ly về điện	45
● 1.3 Rãnh cắm trong máy tính PC	48
- 1.3.1 Sự sắp xếp chân trên rãnh cắm	48
- 1.3.2 Giải mã địa chỉ và nối bus dữ liệu	53
□ Chương 2. Các mạch số	60
● 2.1 Quản lý theo bit	60
● 2.2 Ghép nối kiểu role	62
● 2.3 Bộ điều khiển mini theo chương trình đã nhớ (SPS)	67
● 2.4 Vi mạch ghép nối 8243	79
- 2.4.1 Mô tả	79
- 2.4.2 16 lối ra TTL với 8243	82
- 2.4.3 Môđun vào/ ra 32 bit	86
● 2.5 Giao diện vào/ ra trên rãnh cắm PC	94
- 2.5.1 Vi mạch ghép nối 8255	94
- 2.5.2 Giao diện vào/ ra 24 bit trên rãnh cắm PC	97

- 2.5.3 Giao diện vào/ ra 48 bit trên rãnh cắm PC	99
● 2.6 Các bộ đếm vạn năng	100
- 2.6.1 Vi mạch đếm 8253	100
- 2.6.2 Ghép nối bộ đếm với rãnh cắm trong máy tính PC	110
- 2.6.3 Môđun đếm ở cổng nối tiếp và cổng song song	114
- 2.6.4 Những ứng dụng của vi mạch đếm 8253	128
- Bộ đếm xung 16 bit	128
- Bộ đếm xung 32 bit	129
- Bộ đếm tần số 16 bit	129
- Bộ đếm tần số 7 chữ số	130
- Bộ đếm tần số thấp với độ chính xác cao	132
- Bộ so dây đàn ghita	133
- Bộ phát xung lập trình được	135
- Đồng hồ role lập trình được	136
- Thiết bị kiểm tra phản ứng	138
- Đo điện dung	139
- Điều khiển vòng quay	140
- Bộ biến đổi D/A 16 bit	142
- Bộ tổng hợp tần số	143
□ Chương 3. Các mạch điều khiển với bộ biến đổi A/D	153
● 3.1 Ghép nối bộ biến đổi A/D 8 bit 1 kênh trực tiếp vào cổng nối tiếp	153
● 3.2 Bộ biến đổi A/D 8 bit 1 kênh với ADC 0804	157
- 3.2.1 Các thông số kỹ thuật và cách làm việc	157
- 3.2.2 Cách tạo ra các dải đo khác nhau	159
- 3.2.3 Môđun biến đổi A/D 8 bit với dải đo chuyển đổi được	164
● 3.3 Môđun biến đổi A/D 8 bit 8 kênh dùng vi mạch ADC 0809	172
● 3.4 Môđun biến đổi A/D 12 bit 1 kênh dùng vi mạch ICL 7109	173
● 3.5 Ghép nối môđun biến đổi A/D 12 bit 8 kênh trực tiếp vào cổng nối tiếp	179
● 3.6 Các ứng dụng của bộ biến đổi A/D	188
- 3.6.1 Đo dòng điện	188
- 3.6.2 Đo điện áp vi phân	189
- 3.6.3 Đo điện trở	191
- 3.6.4 Đo điện dung	193
- 3.6.5 Đo tần số	194
- 3.6.6 Đo nhiệt độ	195
- 3.6.7 Thiết bị tự ghi y(t)	200

□ Chương 4. Các mạch điều khiển với bộ biến đổi D/A	207
● 4.1 Cách hoạt động của bộ biến đổi D/A	208
● 4.2 Bộ biến đổi D/A 8 bit AD 7524	209
● 4.3 Bộ biến đổi D/A 8 bit ZN 426	210
● 4.4 Bộ biến đổi D/A 12 bit AD 7545	211
● 4.5 Những ứng dụng của bộ biến đổi D/A	212
- 4.5.1 Môđun biến đổi D/A 12 bit	212
- 4.5.2 Bộ phát còi báo hiệu điều khiển bằng máy vi tính	214
- 4.5.3 Bộ nguồn lập trình được	215
- 4.5.4 Nguồn dòng lập trình được	216
- 4.5.5 Bộ đặt độ khuếch đại lập trình được	218
- 4.5.6 Mạch tích phân lập trình được	219
- 4.5.7 Bộ tạo hàm lập trình được	220
- 4.5.8 Môđun để vẽ đường đặc trưng	221
□ Chương 5. Sự ghép nối bus ở cổng nối tiếp RS 232	224
● 5.1 Phần cứng	224
● 5.2 Phần mềm	231
● 5.3 Các môđun bus khác nhau	234
- 5.3.1 Môđun bus với 8 lối ra TTL	235
- 5.3.2 Môđun bus với 8 lối ra ghép nối quang	236
- 5.3.3 Môđun bus với 8 lối ra dùng với role	238
- 5.3.4 Môđun bus với 8 lối vào TTL	240
- 5.3.5 Môđun bus với 8 lối vào ghép nối quang	242
- 5.3.6 Môđun bus với một lối vào analog	244
- 5.3.7 Môđun bus với tám lối vào analog	246
- 5.3.8 Môđun bus với 24 đường dẫn vào/ ra	248
- 5.3.9 Môđun bus với bộ đếm 8253	251
- 5.3.10 Môđun bus với một lối ra analog	254
□ 6. Phụ lục	256
Phụ lục 1. Thông số của một số khuếch đại thuật toán	256
Phụ lục 2. Vài nét về QBASIC	265
Phụ lục 3. Một số loại rãnh cắm trong máy tính PC	271
□ 7. Tài liệu tham khảo	279

Chương 1

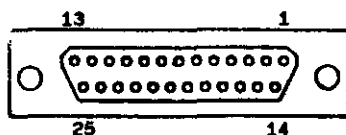
GIAO DIỆN CỦA MÁY TÍNH TRONG ĐO LƯỜNG VÀ ĐIỀU KHIỂN

1.1 CỔNG GHÉP NỐI VỚI MÁY IN

1.1.1 VÀI NÉT CƠ BẢN VỀ CỔNG GHÉP NỐI VỚI MÁY IN

Cổng nối với máy in hay thường gọi là *giao diện Centronics* có lẽ không có gì xa lạ đối với người sử dụng máy tính. Việc nối máy in với máy tính được thực hiện qua ổ cắm 25 chân ở phía sau máy tính. Nhưng đây không chỉ là chỗ nối với máy in mà khi sử dụng máy tính vào mục đích đo lường và điều khiển thì việc ghép nối cũng thực hiện qua ổ cắm này. Qua cổng này dữ liệu được truyền đi song song, nên đôi khi còn được gọi là *cổng ghép nối song song* và tốc độ truyền dữ liệu cũng đạt đến mức lớn đáng kể. Tất cả các đường dẫn của cổng này đều tương thích TTL, nghĩa là chúng đều cung cấp một mức điện áp nằm giữa 0 và 5 V. Do đó ta còn cần phải lưu ý là ở các ổ các đường dẫn lối vào cổng này không được đặt các mức điện áp quá lớn. Sự sắp xếp các chân ra ở cổng máy in với tất cả các đường dẫn được mô tả trên hình 1.1.

Ta thấy rõ là bên cạnh 8 bit dữ liệu còn có những đường dẫn tín hiệu khác, tổng cộng người sử dụng có thể trao đổi một cách riêng biệt với 17 đường dẫn, bao gồm 12 đường dẫn ra và 5 đường dẫn vào. Bởi vì tám đường dẫn dữ liệu D0 - D7 không phải là đường dẫn hai chiều trong tất cả các loại máy tính, nên sau đây ta sẽ thấy là D0 - D7 chỉ có thể được sử dụng như là lối ra. Các lối ra khác nữa là STROBE, AUTOFEED (AF), INIT và SELECT IN (SLCTIN). Khi trao đổi thông tin với máy in, các đường dẫn này có những chức năng xác định. Thí dụ, INIT = 0 thực hiện một quá trình khởi động lại (Reset) ở máy in, còn STROBE có nhiệm vụ ghi các bit dữ liệu đã được gửi từ máy in bằng một xung Low vào trong bộ nhớ của máy in.



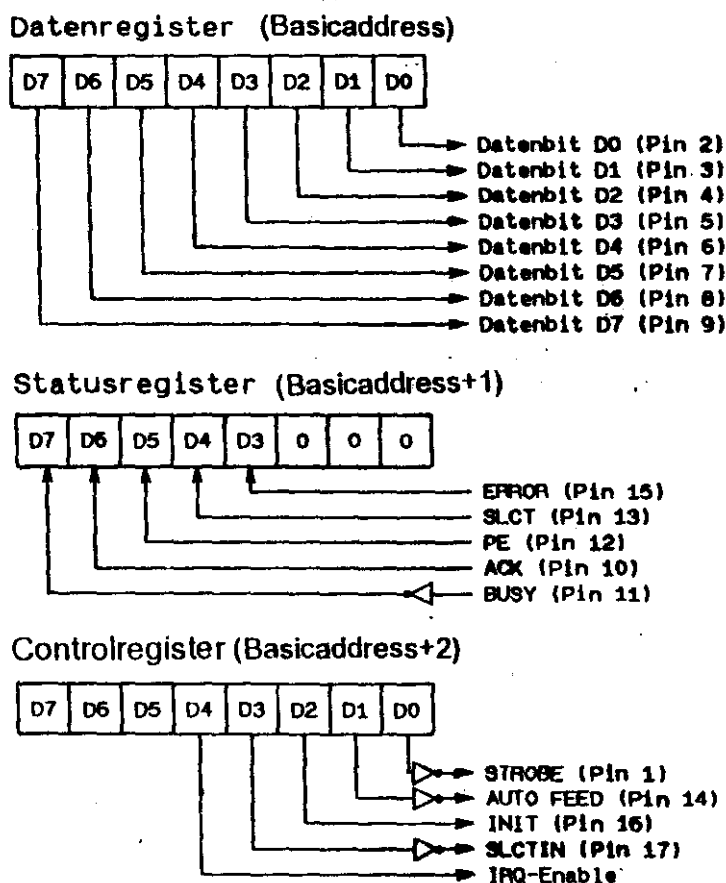
Chân	Ký hiệu	Vào/Ra	Mô tả
1	STROBE	Lỗi ra (Output)	⌋ : Byte được in
2	D0	Lỗi ra	Đường dữ liệu D0
3	D1	Lỗi ra	Đường dữ liệu D1
4	D2	Lỗi ra	Đường dữ liệu D2
5	D3	Lỗi ra	Đường dữ liệu D3
6	D4	Lỗi ra	Đường dữ liệu D4
7	D5	Lỗi ra	Đường dữ liệu D5
8	D6	Lỗi ra	Đường dữ liệu D6
9	D7	Lỗi ra	Đường dữ liệu D7
10	ACK	Lỗi vào (Input)	Acknowledge (xác nhận)
11	BUSY	Lỗi vào	1 : Máy in bận
12	PE	Lỗi vào	Hết giấy
13	SLCT	Lỗi vào	Select (lựa chọn)
14	AF	Lỗi ra	Auto Feed (tự nạp)
15	ERROR	Lỗi vào	Error (Lỗi)
16	INIT	Lỗi ra	0 : Đặt lại máy in
17	SLCTIN	Lỗi ra	Select in
18	GND		Nối đất
19	GND		
20	GND		
21	GND		
22	GND		
23	GND		
24	GND		
25	GND		

Hình 1.1 Bố trí chân ở cổng máy in của máy tính PC.

Công máy in cũng có những đường dẫn lối vào, nhờ vậy mà sự bắt (chéo) tay giữa máy tính và máy in được thực hiện. Chẳng hạn, khi mà máy in không còn đủ chỗ trong bộ nhớ thì máy in sẽ gửi đến máy tính một bit trạng thái (BUSY = 1); điều đó có nghĩa là máy in tại thời điểm này đang bận, không nên gửi thêm các byte dữ liệu khác đến nữa.

Khi hết giấy ở máy in thì máy tính sẽ thông báo là PAPER EMPTY (PE). Đường dẫn lối vào tiếp theo là: ACKNOWLEDGE (ACK), SELECT (SLCT) và ERROR. Tổng cộng máy tính PC có 5 lối vào hướng tới máy in.

1.1.2 TRAO ĐỔI VỚI CÁC ĐƯỜNG DẪN TÍN HIỆU



Hình 1.2 Thanh ghi ở cổng máy in của máy tính PC.

Tất cả các đường dẫn tín hiệu vừa được giới thiệu cho phép trao đổi qua các địa chỉ bộ nhớ của máy tính PC. 17 đường dẫn của cổng máy in sắp xếp thành ba thanh ghi: ghi dữ liệu, ghi trạng thái và ghi điều khiển. Hình 1.2 chỉ ra sự sắp xếp của các đường dẫn tín hiệu tới các bit dữ liệu riêng biệt của thanh ghi.

Địa chỉ đầu tiên đạt đến được của cổng máy in được xem như là địa chỉ cơ bản. Ở các máy tính PC được chế tạo gần đây địa chỉ cơ bản của cổng máy in được sắp xếp như sau:

LPT1 (Cổng máy in thứ nhất)	⇒	Địa chỉ cơ bản = 378 (Hex) hoặc là 3BC (Hex) ở máy tính Laptop
LPT2 (Cổng máy in thứ hai)	⇒	Địa chỉ cơ bản = 278 (Hex)

Địa chỉ cơ bản đồng nhất với thanh ghi dữ liệu. Thanh ghi trạng thái được đặt tới dưới địa chỉ cơ bản + 1. Ở đây cần chú ý rằng mức logic của BUSY (chân 11) được sắp xếp ngược với thanh ghi trạng thái. Thanh ghi điều khiển với bốn đường dẫn lối ra của nó đặt dưới địa chỉ "địa chỉ cơ bản + 2". Ở đây lại cần chú ý tới sự đảo ngược của các tín hiệu STROBE, AUTOFEED và SLCTIN.

Các địa chỉ cơ bản của cổng máy in của máy tính PC được đặt ở những địa chỉ bộ nhớ xác định và có thể được đọc ra bằng chương trình dưới đây. Địa chỉ cơ bản của LPT1 đúng như giá trị 16 bit trong các địa chỉ 408 (hex) và 409 (hex). Cả hai địa chỉ tiếp sau: 40A (hex) và 40B (hex) chứa đựng địa chỉ cơ bản của LPT2.

```

'=====
'  Program: BASADR
'
'  Function: Chương trình này thông báo địa chỉ cơ
'  bản của hai cổng máy in đầu tiên LPT1 và LPT2
'=====
DEF SEG = 0
CLS
'----- Địa chỉ cơ bản của LPT1-----
basadr.low = PEEK (&H408)

```

```

basadr.high = PEEK (&H409)
basadr.lpt1 = basadr.low + 256 * basadr.high
IF basadr.lpt1 = 0 THEN
PRINT "LPT1: không tồn tại"
ELSE
PRINT "LPT1: Địa chỉ cơ bản là";
Hexs (basadr.lpt1); "(hex)"
END IF

```

----- Địa chỉ cơ bản của LPT2 -----

```

basadr.low = PEEK (&H40A)
basadr.high = PEEK (&H40B)
basadr.lpt2 = basadr.low + 256 * basadr.high
IF basadr.lpt2 = 0 THEN
PRINT "LPT2: không tồn tại" ELSE
PRINT "LPT2: Địa chỉ cơ bản là";
HEXS (basadr.lpt2); "(hex)"
END IF
END

```

Khi ta muốn trao đổi với thanh ghi trong một ngôn ngữ lập trình thì cách tốt nhất là lúc bắt đầu chương trình ta thực hiện một sự khởi tạo để khẳng định các địa chỉ thanh ghi. Trong QBasic sự khởi tạo này được viết như sau:

<i>basadr = &H378</i>	<i>REM Basicaddress</i>
<i>datreg = basadr</i>	<i>REM Dateneregister</i>
<i>startreg = basadr + 1</i>	<i>REM Statusregister</i>
<i>contrreg = basadr + 2</i>	<i>REM Controlregister</i>

Các đường dẫn tín hiệu riêng biệt cho phép lập trình như dưới đây:

Ghi lên Dateneregister:	<i>OUT datreg, outbyte</i>
Đọc từ Statusregister:	<i>inbyte = INP (startreg)</i>
Ghi lên Controlregister:	<i>OUT contrreg, outbyte</i>

Các outbyte và inbyte chứa đựng các số trong hệ đếm mười nằm trong khoảng 0 đến 255. Ta có thể thông báo mẫu bit tương ứng của outbyte và inbyte qua giá trị của các bit dữ liệu riêng biệt. Muốn thế ta cần biết rằng D0 chiếm giá trị là 2^0 , D1 chiếm giá trị là 2^1 , D2 chiếm giá trị là 2^2 v. v. Khi mà thanh ghi trạng thái chứa giá trị đã đọc vào inbyte = 125 thì các bit dữ liệu sau được phép thông báo:

D7 = 1	⇒	128
D6 = 0	⇒ +	0
D5 = 0	⇒ +	0
D4 = 1	⇒ +	16
D3 = 1	⇒ +	8
D2 = 0	⇒ +	0
D1 = 0	⇒ +	0
D0 = 0	⇒ +	0 ⇒ Tổng số = 128 + 16 + 8 = 152

Nói cách khác: BUSY là Low (mức thấp), ACK là Low, PE là Low, SLCT là High (mức cao), ERROR là High.

1.1.3 CHƯƠNG TRÌNH KIỂM TRA CỔNG MÁY IN

Dưới đây xin giới thiệu một chương trình viết bằng QBasic cho phép trao đổi với tất cả các đường dẫn của cổng máy in một cách riêng biệt. Chương trình này có tên là LPTTEST.BAS. Nếu như bạn đọc chưa có kinh nghiệm làm quen với QBasic thì trước khi thử nghiệm chương trình này nên đọc thêm phần phụ lục 2 ở cuối sách.

Địa chỉ cơ bản của cổng máy in có thể được thông báo bằng chương trình có tên BASADR.BAS.

Chương trình LPTTEST được viết như sau:

```

'=====
' Program: LPTTEST
' Function: Chương trình kiểm tra để trao đổi với
'          tất cả các đường dẫn ở cổng máy in
'=====

```

```

COLOR 0, 15
CLS
a$ = Chương trình kiểm tra cổng máy in
a = LEN (a$): b = (80 - a) / 2 - 1
LOCATE 1, B: PRINT CHR$ (201);
STRING (a + 2, CHR$ (205)); CHR$ (187)
LOCATE 2, b: PRINT CHR$ (186)
LOCATE 2, b + 2. PRINT a$
LOCATE 2, a + b + 2: PRINT CHR$ (186)
LOCATE 3, b: PRINT CHR$ (200);
STRING$ (a + 2, CHR$ (205)); CHR$ (188)
PRINT
LOCATE 5,15:PRINT " Địa chỉ cơ bản của
                    cổng máy in (Hex):378"

basadr = &H378
datreg = basadr           'Datenregister
statreg = basadr + 1      'Statusregister
contrreg = basadr + 2     'Controlregister
LOCATE 7, 10: PRINT "Các đường dẫn lối ra"
LOCATE 7, 50: PRINT "Các đường dẫn lối vào"
LOCATE 8, 10: PRINT "-----"
LOCATE 8, 50: PRINT "-----"
LOCATE 9, 10: PRINT "Đường dẫn dữ liệu D0 ="
LOCATE 9, 50: PRINT "ERROR ="
LOCATE 10, 10: PRINT "Đường dẫn dữ liệu D1 ="
LOCATE 10, 50: PRINT "SLCT ="
LOCATE 11, 10: PRINT "Đường dẫn dữ liệu ="
LOCATE 11, 50: PRINT "PE ="
LOCATE 12, 10: PRINT "Đường dẫn Dữ liệu D3 ="
LOCATE 12, 50: PRINT "ACK ="
LOCATE 13, 10: PRINT "Đường dẫn Dữ liệu D4 ="
LOCATE 13, 50: PRINT "BUSY ="
LOCATE 14, 10: PRINT "Đường dẫn Dữ liệu D5 ="
LOCATE 15, 10: PRINT "Đường dẫn Dữ liệu D6 ="
LOCATE 16, 10: PRINT "Đường dẫn Dữ liệu D7 ="

```

```

LOCATE 17, 10: PRINT "STROBE ="
LOCATE 18, 10: PRINT "AUTO FEED ="
LOCATE 19, 10: PRINT "INIT ="
LOCATE 20, 10: PRINT "SLCTIN =" PRINT
LOCATE 22, 1: PRINT "Hướng dẫn:"
LOCATE 22,15: PRINT CHR$(24);CHR$(25);":Lựa chọn lối ra"
LOCATE 22, 35: PRINT "ESC: Kết thúc"
LOCATE 22, 50: PRINT "F1: Đảo lối ra "
LOCATE 23, 1: PRINT "-----"
'-----Tất cả đường dẫn lối ra đặt bằng 1-----
FOR i = 9 TO 20
LOCATE i, 30: PRINT 1
NEXT
OUT datreg, 255: OUT contrreg, (31 XOR 11)
ZEILE = 9: spalte = 10
DO
'
' Đọc dòng hiện tại (cột 10 đến 24) vào như là text$
'
"text$ =" "
FOR i = 0 TO 14"
text$ = text$ + CHR$( SCREEN (ZEILE, spalte + i))
NEXT
'-----test$ : mô tả đảo-----
COLOR 15, 0: LOCATE ZEILE, spalte
PRINT text$
COLOR 0, 15 'Sự xuất dữ liệu tiếp theo trở lại bình thường'
'-----Bắt đầu vòng lặp, hỏi lại bàn phím-----
DO
taste$ = INKEYS 'Chờ ấn phím
inbyte=INP (statreg) 'Đọc vào thanh ghi trạng thái
FOR i = 3 TO 7
LOCATE 6 + i, 60
IF i < 7 THEN PRINT (inbyte AND 2 ^ i) \ 2 ^ i
IF i=7 THEN PRINT (inbyte AND 2^i)\2^i) XOR 1

```

```

NEXT
LOOP UNTIL taste$ <> " "
-----Kết thúc vòng lặp (Phím đã nhấn)-----
LOCATE ZEILE, spalte
PRINT text$
'-----Phím nào đã được nhấn-----
SELECT CASE ASC (RIGHT$ (taste$, 1))
CASE 27: EXIT DO           'Phím ESC
CASE 59                     'Phím F1
LOCATE ZEILE, 30
PRINT VAL (CHR$ (SCREEN (ZEILE, 31))) XOR 1
CASE 72: IF ZEILE > 9 THEN
ZEILE = ZEILE - 1           'Phím mũi tên lên
CASE 80: IF ZEILE < 20 THEN
ZEILE = ZEILE + 1           'Phím mũi tên xuống
END SELECT
'-----Xuất dữ liệu-----
d.byte = 0
FOR i = 9 TO 16
d.byte = d.byte + VAL (CHR$ (SCREEN (I, 31)))
* 2 ^ (i - 9)
NEXT
OUT datreg, d.byte          'Xuất Datenbyte
s.byte = 0
FOR i = 17 TO 20
s.byte = s.byte + VAL (CHR$ (SCREEN (i, 31)))
* 2 ^ (i - 17)
NEXT
s.byte = s.byte XOR 11      'Chú ý phép đảo !
OUT contrreg, s.byte        'Xuất ra Controlregister
LOOP
END

```

Sau khi khởi động chương trình, màn hình sẽ được trình bày như trên hình 1.3

Chương trình kiểm tra công máy in

Địa chỉ cơ bản của công máy in (Hex): 378

Duong dan loi ra		Duong dan loi vao	
Duong_dan Du_lieu D0	= 1	ERROR	= 1
Duong_dan Du_lieu D1	= 1	SLCT	= 1
Duong_dan Du_lieu D2	= 1	PE	= 0
Duong_dan Du_lieu D3	= 1	ACK	= 1
Duong_dan Du_lieu D4	= 1	BUSY	= 0
Duong_dan Du_lieu D5	= 1		
Duong_dan Du_lieu D6	= 1		
Duong_dan Du_lieu D7	= 1		
STROBE	1		
AUTO FEED	1		
INIT	1		
SLCTIN	1		

Huong_dan: ↑↓ : Lua_chon_loi_ra ESC : Ket_thuc F1: Dao_loi_ra

Hình 1.3 Trình bày trên màn hình của chương trình LPTTEST.

Bằng phím mũi tên ta có thể lựa chọn một đường dẫn dữ liệu hoặc là một bit điều khiển và đảo ngược lại bằng phím F1. Giống như các bit được mô tả trên màn hình, chúng cũng xuất hiện ở cổng máy in của máy tính PC. Điều đó có nghĩa là bằng sự xác nhận của phím F1 mức hiện tại của tất cả các đường dẫn dữ liệu và đường dẫn điều khiển được đưa ra. Đồng thời, các bit trạng thái được hỏi lại và chỉ ra trên màn hình.

Ứng dụng: Điều khiển máy in.

Một ứng dụng của chương trình LPTTEST là dùng để gửi một văn bản ra máy in. Bình thường thì cái gì mà có thể nhìn thấy trong sự vội vã của máy tính PC thì ở đây ta có thể cho vào từng bit một.

1. GIAO DIỆN CỦA MÁY TÍNH

Bước	D7	D6	D5	D4	D3	D2	D1	D0	(Hex)
1 Bật máy in									
2 Chương trình LPTTEST bắt đầu									
3 Xếp sát mã ASCII của " T "									
4 Các xung STROBE âm	0	1	0	1	0	1	0	0	\$ 54
5 Xếp sát mã ASCII của " e "									
6 Các xung STROBE âm	0	1	1	0	0	1	0	1	\$ 65
7 Xếp sát mã ASCII của " s "									
8 Các xung STROBE âm	0	1	1	1	0	0	1	1	\$ 73
9 Xếp sát mã ASCII của " t "									
10 Các xung STROBE âm	0	1	1	1	0	1	0	0	\$ 74
11 Xếp sát FF									
12 Các xung STROBE âm	0	0	0	0	1	1	0	1	\$ 0D
13 Xếp sát FF									
14 Các xung STROBE âm	0	0	0	0	1	0	1	0	\$ 0A
15 Xếp sát mã ASCII của " O "									
16 Các xung STROBE âm	0	1	0	0	1	1	1	1	\$ 4F
17 Xếp sát mã ASCII của " K "									
18 Các xung STROBE âm	0	1	0	0	1	0	1	1	\$ 4B
19 Xếp sát FF									
20 Các xung STROBE âm	0	0	0	0	1	1	0	0	\$ 0C

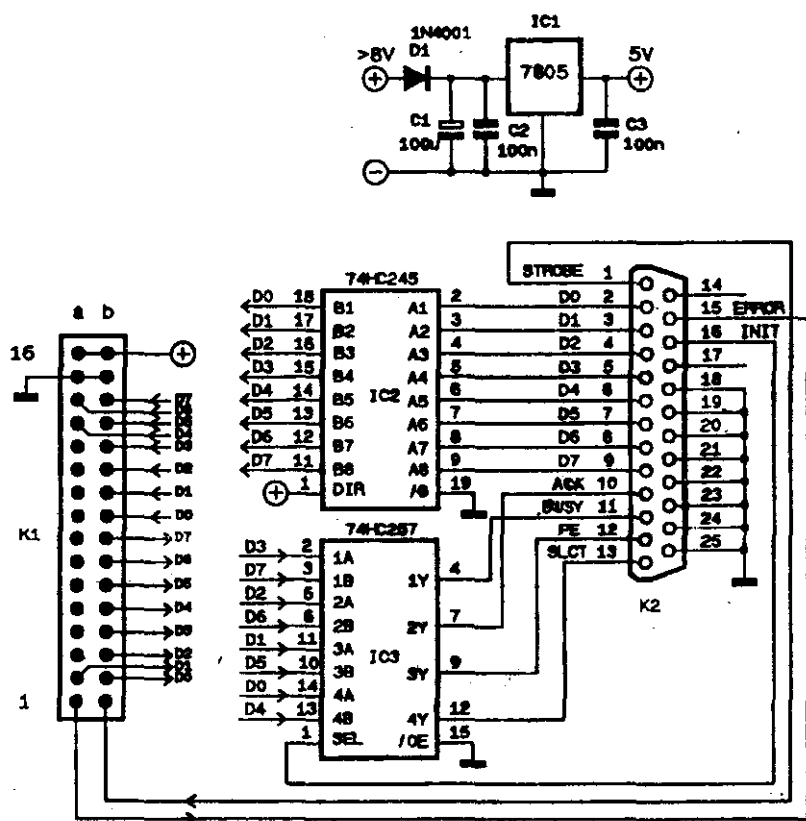
Thí dụ ta cần phải in ra từ "Test" và trên dòng tiếp theo là chữ "OK". Từng bước diễn ra là trước hết ta xếp sát mã ASCII của các chữ cái đã gặp ở các đường dẫn dữ liệu, sau đó bằng một xung âm STROBE các dữ liệu được chuyển giao sang máy in. Từ đó ta suy ra rằng sau khi khởi động chương trình bit điều khiển STROBE có mức cao (High) nên ta nhận được một xung âm STROBE qua việc đảo hai lần bằng phím F1. Sau đó theo cách này từ "Test" sẽ được gửi đi, tiếp theo là dãy điều khiển cho việc bắt đầu một dòng mới. Đây là các dãy điều khiển CARRIAGE RETURN (lui về đầu dòng, viết tắt là CR) và LINE FEED (xuống dòng, viết tắt là LF). Sau đây ta có thể gửi đi mã ASCII của "OK". Như là bước cuối cùng chỉ có một FORM FEED (FF) được gửi để nạp một trang. Do vậy máy in cho ra cả trang.

Bảng thống kê trên đây một lần nữa tóm tắt các bước riêng biệt.

1.1.4 MÔĐUN CƠ SỞ 8 BIT VÀO / RA DÙNG CHO CỔNG MÁY IN

Trong mục trước ta đã thấy trong số 17 đường dẫn lối vào chỉ có 5 đường được sử dụng như là lối vào. Tất nhiên bằng một mạch phụ thêm không mấy tốn kém ta cũng có thể đọc vào nhiều hơn là 5 bit.

Trên hình 1.4 chỉ ra sơ đồ mạch của một môđun đơn giản mà ta có thể nối trực tiếp với cổng máy in. Môđun này có thể gửi ra 8 bit cũng như đọc vào 8 bit qua cổng máy in.



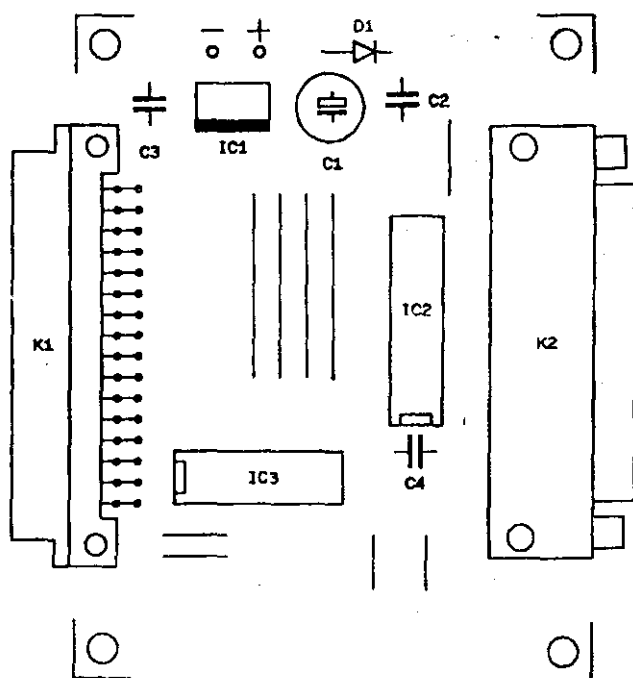
Hình 1.4 Sơ đồ của môđun cơ sở dùng cho cổng máy in.

Các đường dẫn dữ liệu đi tới bộ đệm 74HC245, vì mạch này tách ra các dữ liệu đã đến của máy tính PC từ môđun ghép nối. Để đọc vào 8 bit

dữ liệu, vi mạch 74HC257 được sử dụng, vi mạch này chứa 4 bộ quét động (dồn kênh, multiplexer) với hai lối vào và một lối ra. Nhờ vậy một chu trình đọc vào được chia làm hai nửa. Chân ra $SEL = 0$ có tác dụng làm cho các bit dữ liệu D0, D1, D2 và D3 đến được máy tính PC, trong khi $SEL = 1$ các bit còn lại là D4, D5, D6 và D7 sẽ được đọc vào. Bit điều khiển INIT có tác dụng chuyển mạch đổi với SEL.

Nguồn nuôi cho môđun này lấy điện áp + 5 V từ vi mạch ổn áp 7805 mà điện áp lối vào của nó có thể được lấy từ bộ pin 9 V hoặc bộ nắn điện từ nguồn điện lưới 50 Hz. Dòng điện tiêu thụ chỉ cỡ mA.

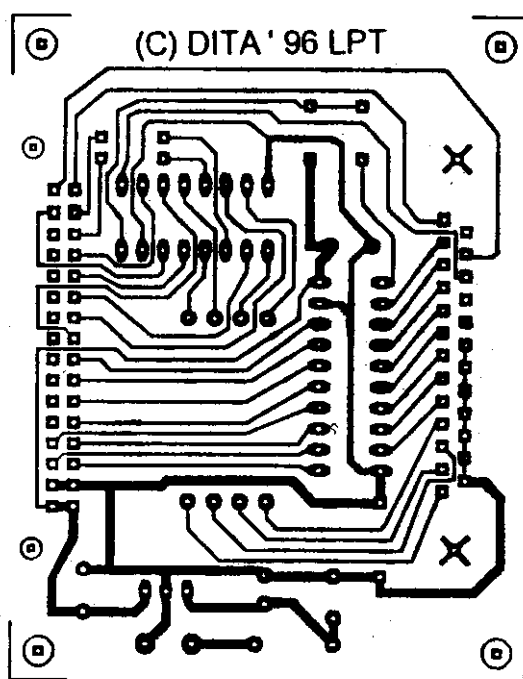
Ngoài 8 lối vào và lối ra còn có một bit có thể đọc vào qua đường dẫn trạng thái và nhờ bit điều khiển STROBE một bit có thể được đọc ra. Tất cả các đường dẫn dữ liệu dẫn đến một đầu nối 32 chân để có thể ghép nối các phần cứng riêng lẻ với môđun này. Một số phần cứng khác nhau sẽ được lần lượt giới thiệu sau này.



Hình 1.5 Sắp xếp linh kiện trên môđun cơ sở dùng cho cổng máy in.

Sơ đồ sắp xếp linh kiện của môđun cơ sở dùng cho cổng máy in như mô tả trên hình 1.5, còn mạch in tương ứng như trên hình 1.6. Nếu như

bạn đọc muốn tự làm thì khi vẽ mạch nên lưu ý là dòng chữ (C) DITA 96 trên mặt hàn linh kiện phải nằm đúng chiều.



Hình 1.6 Sơ đồ mạch in của môđun cơ sở dùng cho cổng máy in.

Để lắp ráp mạch này ta cần có những linh kiện sau:

Linh kiện bán dẫn: C2, C3, C4: 100 nF

IC1: 7805

Điốt bán dẫn:

IC2: 74HC245

D1: 1N 4001

IC3: 74HC 257

Phích cắm:

Tụ điện:

K1: đẹt 32 chân

C1: 100 μ F / 35 V

K2: 25 chân

Việc trao đổi với môđun cơ sở được thực hiện trong QBasic rất đơn giản. Khi cần đưa các dữ liệu ra thì ta dùng lệnh OUT register-address, outbyte. Địa chỉ thanh ghi có thể là địa chỉ của các thanh ghi dữ liệu hoặc là của các thanh ghi điều khiển. Biến outbyte chứa đựng một con số trong hệ đếm mười nằm trong khoảng 0 đến 255.

Thí dụ: OUT&H378, 128

Lệnh này truy nhập lên thanh ghi dữ liệu của cổng máy in. Ở ổ cắm 32 chân det của môđun cơ sở sẽ xuất hiện các bit dữ liệu sau:

$D7 = 1, D6 = 0, D5 = 0, D4 = 0, D3 = 0, D2 = 0, D1 = 0, D0 = 0.$

Chu trình đọc vào được hình thành một cách phức tạp hơn và chương trình dưới đây sẽ làm sáng tỏ hơn nhận xét này.

```

=====
' Unterprogram: lese. lpt
'
' Function: Chương trình này đọc vào một byte qua
' cổng máy in. Kết quả đúng trong biến tính theo
' byte
'
' Hardware: Môđun cơ sở dùng cho cổng máy in đã
' trình bày ở chương 1
'
=====
' -----Set up cổng máy in-----
basard = &H378          'Basicaddress của cổng máy in
datreg = basadr          'Datenregister
statreg = basadr + 1     'Statusregister
contrreg = basadr + 2    'Controlregister
'
' ----- Đọc dữ liệu vào -----
'
OUT contrreg, INP(contrreg) AND(255 - 4) 'init = 0
inbyte1 = INP(statreg) 'Đọc vào từ D0,D1,D2,và D3
OUT contrreg, INP (contrreg) OR 4 'init = 1
inbyte2 = INP(statreg) 'Đọc vào từ D4,D5,D6,và D7
'
' -----Sự sắp xếp của bit dữ liệu đã đọc vào-----
'
inbyte1: Statusregister bit 4 (SLCT) là D0
' Statusregister bit 5 (PE) là D1

```

```

'           Statusregister bit 6 (ACK)   là D2
'           Statusregister bit 5 (BUSY)   là /D3
'   inbyte2: Statusregister bit 4 (SLCT)   là D4
'           Statusregister bit 5 (PE)     là D5
'           Statusregister bit 6 (ACK)   là D6
'           Statusregister bit 5 (BUSY)   là /D7
'

```

```

inbyte = (((inbyte XOR 128) AND &HFO) /16) +
          ((inbyte2 XOR 128) AND &HFO)

```

END SUB

Việc sử dụng một chương trình con có ưu điểm là trong chương trình chính chỉ cần có một lệnh riêng lẻ đã có thể đọc vào 8 bit dữ liệu. Trong QBasic, việc gọi ra được tiến hành bằng lệnh *call lese.lpt (inbyte)*. Sau đó biến inbyte sẽ chứa con số trong hệ đếm mười của các bit dữ liệu đã đọc vào. Khi chú ý tới chương trình con một cách chi tiết hơn, ta sẽ thấy thực ra là hai byte sẽ được đọc vào. Sau đó các bit dữ liệu có giá trị còn cần phải được phân loại và sắp xếp lại. Đồng thời phải chú ý là lúc này điều kiện như INIT = 0 được thực hiện. Lệnh OUT contrreg INP (contrreg) AND (255 - 4) đặt INIT lên Low (thấp) để không gây ảnh hưởng đến các bit khác trong thanh ghi điều khiển. Để đạt được điều này, trước hết nội dung hiện thời của thanh ghi điều khiển được đọc vào sau đó được đặt bằng một liên kết AND (VÀ) với (255 - 4) INIT thành mức Low (thấp).

1.2 CỔNG NỐI TIẾP RS 232

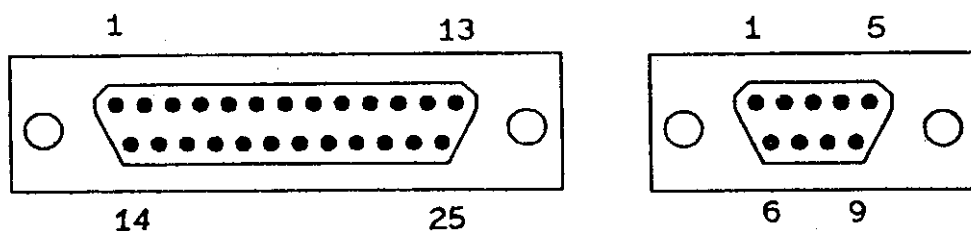
1.2.1 VÀI NÉT CƠ BẢN VỀ CỔNG NỐI TIẾP

Cổng nối tiếp RS-232 là giao diện phổ biến rộng rãi nhất. Người dùng máy tính PC còn gọi các cổng này là COM 1, còn COM 2 để tự do cho các ứng dụng khác. Giống như cổng máy in, cổng nối tiếp RS-232 cũng được sử dụng một cách rất thuận tiện cho mục đích đo lường và điều khiển.

Việc truyền dữ liệu qua cổng RS-232 được tiến hành theo cách nối tiếp, nghĩa là các bit dữ liệu được gửi đi nối tiếp nhau trên một đường

dẫn. Trước hết, loại truyền này có khả năng dùng cho những khoảng cách lớn hơn, bởi vì các khả năng gây nhiễu là nhỏ đáng kể hơn khi dùng một cổng song song. Việc dùng cổng song song có một nhược điểm đáng kể là cáp truyền dùng quá nhiều sợi, và vì vậy rất đắt tiền. Hơn nữa, mức tín hiệu nằm trong khoảng 0 - 5 V đã tỏ ra không thích ứng với khoảng cách lớn.

Cổng nối tiếp RS 232 không phải là một hệ thống bus, nó cho phép dễ dàng tạo ra liên kết dưới hình thức điểm với điểm giữa hai máy cần trao đổi thông tin với nhau. Một thành viên thứ 3 không thể tham gia vào cuộc trao đổi thông tin này. Trên hình 1.7 là sự bố trí chân của phích cắm RS 232 ở máy tính PC.



Chân (loại 9 chân)	Chân (loại 25 chân)	Chức năng	
1	8	DCD - Data Carrier Detect	Lối vào
2	3	RxD - Receive Data	Lối vào
3	2	TxD - Transmit Data	Lối ra
4	20	DTR - Data Terminal Ready	Lối ra
5	7	GND - Nối đất	
6	6	DSR - Data Set Ready	Lối vào
7	4	RTS - Request to Send	Lối ra
8	5	CTS - Clear to Send	Lối vào
9	22	RI - Ring Indicator	Lối vào

Hình 1.7 Sắp xếp chân của cổng nối tiếp ở máy tính PC.

Từ hình vẽ ta thấy ổ cắm nối tiếp RS 232 có tổng cộng 8 đường dẫn chưa kể đường nối đất. Trên thực tế có hai loại phích cắm, một loại có 9 chân và một loại 25 chân. Cả hai loại này đều có chung một đặc điểm khác hẳn với cổng máy in là chỗ nối với máy in ở máy tính PC là ổ cắm, trong khi ở cổng nối tiếp lại là phích cắm nhiều chân.

Việc truyền dữ liệu xảy ra ở trên hai đường dẫn. Qua chân cắm ra TxD (Transmit Data), máy tính gửi các dữ liệu của nó đến máy kia. Trong khi đó các dữ liệu mà máy tính nhận được, lại được dẫn đến chân nối RxD (Receive Data). Các tín hiệu khác đóng vai trò như là những tín hiệu hỗ trợ khi trao đổi thông tin và vì thế không phải trong mọi ứng dụng đều dùng đến.

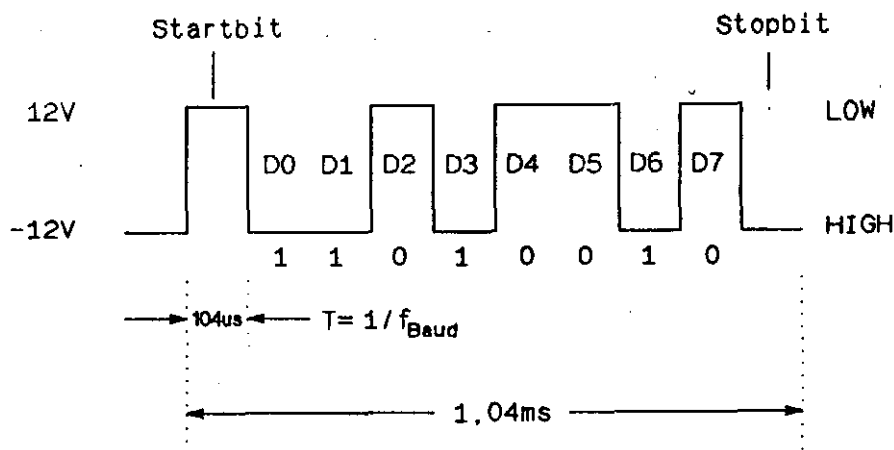
Việc truyền dữ liệu

Mức tín hiệu trên chân ra RxD tùy thuộc vào đường dẫn TxD và thông thường nằm trong khoảng - 12 V đến + 12 V. Các bit dữ liệu được gửi đảo ngược lại. Mức điện áp đối với mức High nằm giữa - 3 V và - 12 V và mức Low nằm giữa + 3 V và + 12 V. Trên hình 1.8 mô tả một dòng dữ liệu điển hình của một byte dữ liệu trên cổng nối tiếp RS 232.

Ở trạng thái tĩnh trên đường dẫn có điện áp - 12 V. Một bit khởi động (Startbit) sẽ mở đầu việc truyền dữ liệu. Tiếp đó là các bit dữ liệu riêng lẻ sẽ đến, trong đó những bit giá trị thấp sẽ được gửi trước tiên. Con số của các bit dữ liệu thay đổi giữa năm và tám. Ở cuối của dòng dữ liệu còn có một bit dừng (Stopbit) để đặt trở lại trạng thái lỗi ra (- 12 V).

Bảng tốc độ baud (đọc là bô) ta thiết lập tốc độ truyền dữ liệu. Các giá trị thông thường là 300; 600; 1.200; 2.400; 4.800; 9.600 và 19.200 baud. Ký hiệu baud tương ứng với số bit được truyền trong một giây. Chẳng hạn như khi tốc độ baud bằng 9.600 có nghĩa là có 9.600 bit dữ liệu được truyền trong mỗi giây. Từ đó ta suy ra rằng còn có một bit bắt đầu và một bit dừng được gửi kèm theo với một byte dữ liệu. Như vậy với mỗi một byte đã có 10 bit được gửi. Nhờ vậy mà có thể ước đoán một cách dễ dàng lượng dữ liệu cực đại đã được truyền. Với tốc độ 9.600 baud cho phép truyền nhiều nhất là 960 byte mỗi giây. Qua cách ước

tính đơn giản này ta cũng thấy được một nhược điểm không nhỏ của cổng truyền nối tiếp là tốc độ truyền dữ liệu bị hạn chế.



Hình 1.8 Dòng dữ liệu trên cổng RS 232 với tốc độ 9.600 baud.

Còn một vấn đề nữa là khuôn mẫu (Format) truyền dữ liệu cần phải được thiết lập như nhau cả ở bên gửi và cũng như ở bên nhận. Các thông số truyền có thể được thiết lập trên máy PC bằng các câu lệnh trên (còn gọi là trong) DOS. Ngay cả trên WINDOWS cũng có những chương trình riêng để sử dụng. Khi đó các thông số truyền dữ liệu như: tốc độ baud, số bit dữ liệu, số bit dừng, bit chẵn lẻ (parity) có thể được thiết lập một cách rất đơn giản.

1.2.2. SỰ TRAO ĐỔI CỦA ĐƯỜNG DẪN TÍN HIỆU

Cũng như ở cổng máy in, các đường dẫn tín hiệu riêng biệt cũng cho phép trao đổi qua các địa chỉ trong máy tính PC. Trong trường hợp này, người ta thường sử dụng những vi mạch có mức độ tích hợp cao để có thể hợp nhất nhiều chức năng trên một chip. Ở máy tính PC thường có một bộ phát/ nhận không đồng bộ vạn năng (gọi tắt là UART: Universal Asynchronous Receiver/Transmitter) để điều khiển sự trao đổi thông tin giữa máy tính và các thiết bị ngoại vi. Phổ biến nhất là vi mạch 8250 của hãng NSC hoặc các thế hệ tiếp theo, chẳng hạn như 16C550. Bộ UART này có 10 thanh ghi để điều khiển tất cả các chức năng của việc nhập vào và xuất ra dữ liệu theo cách nối tiếp. Liên quan đến nội dung

của cuốn sách này chỉ đề cập đến hai điều đáng quan tâm. Đó là thanh ghi điều khiển-môđem và thanh ghi trạng thái-môđem.

Sự sắp xếp của các thanh ghi này như sau:

Thanh ghi điều khiển-môđem (địa chỉ cơ bản +4)

Bit 0:	DTR	(Lỗi ra)	Giá trị 1
Bit 1:	RTS	(Lỗi ra)	Giá trị 2

Thanh ghi trạng thái-môđem (địa chỉ cơ bản +6)

Bit 4:	CTS	(Lỗi vào)	Giá trị 16
Bit 5:	DSR	(Lỗi vào)	Giá trị 32
Bit 6:	RI	(Lỗi vào)	Giá trị 64
Bit 7:	DCD	(Lỗi vào)	Giá trị 128

Lại cũng giống như ở cổng ghép nối với máy in, các thanh ghi được trao đổi qua ô nhớ trong vùng vào/ ra (Input/ Output). Địa chỉ đầu tiên có thể tới được của cổng nối tiếp được gọi là địa chỉ cơ bản (Basic Address). Các địa chỉ ghi tiếp theo được đạt tới bằng việc cộng thêm số thanh ghi đã gặp của bộ UART vào địa chỉ cơ bản.

Địa chỉ cơ bản của cổng nối tiếp của máy tính PC có thể tóm tắt trong bảng các địa chỉ sau:

COM 1 (cổng nối tiếp thứ nhất)	Địa chỉ cơ bản =	3F8 (Hex)
COM 2 (cổng nối tiếp thứ hai)	Địa chỉ cơ bản =	2F8 (Hex)
COM 3 (cổng nối tiếp thứ ba)	Địa chỉ cơ bản =	3E8 (Hex)
COM 4 (cổng nối tiếp thứ tư)	Địa chỉ cơ bản =	2E8 (Hex)

Trong QBasic các tín hiệu qua cổng giao diện nối tiếp RS 232 có thể được trao đổi như sau:

basadr = &H2F8 REM Basicaddress COM2

DTR đặt là - 12 V: *OUT(basadr + 4), INP(basadr + 4) AND (255 - 1)*

DTR đặt là - 12 V: *OUT(basadr + 4), INP(basadr + 4) OR 1*

DTR đặt là - 12 V: *OUT(basadr + 4), INP(basadr + 4) AND (255 - 2)*

DTR đặt là - 12 V: *OUT(basadr + 4), INP(basadr + 4) OR 2*

DCD hỏi lại: *level of dcd = (INP(basadr + 6) AND 16) / 16*

DCD hỏi lại: $level\ of\ dcd = (INP(basadr + 6) AND 32) / 32$
 DCD hỏi lại: $level\ of\ dcd = (INP(basadr + 6) AND 64) / 64$
 DCD hỏi lại: $level\ of\ dcd = (INP(basadr + 6) AND 128) / 128$

Đương nhiên trong hầu hết các ứng dụng của giao diện nối tiếp không đòi hỏi phải lập trình cho tất cả các đường dẫn một cách riêng biệt. Dùng cho mục đích này đã có các chip thông minh được dùng ở phía bên nhận và bên gửi. Sau đó phải kể đến là trong QBasic có những lệnh rất đơn giản phù hợp với biên bản truyền RS 232. Chẳng hạn như có hai máy tính PC ghép nối với nhau qua cổng nối tiếp thì chỉ cần có một lệnh để gửi một byte dữ liệu đúng khuôn mẫu tới bên gửi.

Bằng lệnh OPEN COM giao diện nối tiếp sẽ được mở. Bằng PRINT và INPUT các dữ liệu có thể được gửi và được đọc.

Cú pháp hoàn chỉnh của lệnh được miêu tả một cách đầy đủ trong bảng dưới đây:

*OPEN "COMn": Options1 Options2" [FOR Mode] AS [#] Dateinr%
 LEN= Sentencelength %*

n	Cổng mở để truyền dữ liệu: 1 = COM1; 2 = COM2 v.v
Option1	Các thông số truyền dữ liệu hay được sử dụng nhất [Baud] [,[Parity] [, [Daten] [, [Stop]]]]
	○ Baud là tốc độ truyền dữ liệu tính ra baud của thiết bị được mở: 75; 110; 150; 300; 600; 1200; 2400; 4800; 9600; 19200
	○ Parity chỉ ra là tính chẵn lẻ cần phải được kiểm tra như thế nào: N (không kiểm tra) E (chẵn) O (lẻ) S (ký tự trống) M (đánh dấu) PE (có kiểm tra lỗi)
	○ Daten (dữ liệu) là con số của các bit dữ liệu trên byte: 5, 6, 7, 8
	○ Stop là số của bit Stop: 1; 1,5; 2
	Thông thường thiết lập: 300 baud, chẵn, 7 bit dữ liệu, 1 Stopbit

Option2

Một danh sách các tham số ít được sử dụng hơn, các tham số này được ngăn cách bằng dấu phẩy:

Option	Mô tả
ASC	Khi mở máy trong mã ASCII
BIN	Khi mở máy trong mã nhị phân
CD[m]	Khả năng định thời gian chờ lỗi đối với đường dẫn Data-Carrier-Detect DCD (bằng mili giây: ms)
CS[m]	Khả năng định thời gian chờ lỗi đối với đường dẫn Clear-To-Send (CTS) bằng mili giây (ms)
DS[m]	Khả năng định thời gian chờ lỗi đối với đường dẫn Data-Set-Ready (DS) bằng mili giây (ms)
LF	Gửi một ký tự xuống dòng sau một ký tự lùi về đầu dòng
OP[m]	Khả năng định OPEN COM cần phải chờ bao lâu cho đến khi tất cả các liên kết truyền dữ liệu đã mở (bằng mili giây)
RB[n]	Khả năng định kích thước bộ đệm bên nhận (bằng byte)
RS	Nén lại sự xuất hiện của một yêu cầu gửi
TB[n]	Khả năng định kích thước của bộ đệm truyền (bằng byte)

Mode

INPUT, OUTPUT hoặc RANDOM (Sự thiết lập tiêu chuẩn)

- INPUT chỉ ra rằng giao diện đã mở để nạp dữ liệu vào một cách tuần tự.
- OUTPUT chỉ ra rằng giao diện đã mở để xuất dữ liệu ra một cách tuần tự.
- RANDOM chỉ ra rằng giao diện đã mở trong kiểu

(mode) truy nhập trực tiếp,

Dateinr% Một số nguyên nằm giữa 1 và 255 đặc trưng cho kênh truyền dữ liệu, cho đến lúc nó vẫn mở.

Độ dài câu% Độ lớn bộ đệm trong mode truy nhập trực tiếp (mặc định là 128 byte)

Thí dụ: *OPEN "COM2: 19200, N, 8, 1, CS0, DS0"*
FOR RANDOM AS # 1

Lệnh này mở cổng nối tiếp COM2 với 19.200 baud, không có bit kiểm tra chẵn lẻ (parity), tám bit dữ liệu, một bit dừng như là kênh truyền dữ liệu với số 1.

Qua việc mở giao diện bằng lệnh OPEN cả hai đường dẫn RTS và DTS được chuyển sang + 12 V. Với khuôn mẫu truyền đã được thiết lập, bây giờ các dữ liệu có thể được đọc vào và đưa ra. Việc giữ dữ liệu tuân theo lệnh PRINT #, ký hiệu \$. Ta phải chú ý là không bao giờ được phép quên lập dấu chấm phẩy ở cuối của lệnh ! Bởi vì nếu không thì một ký tự thứ hai, cụ thể là CR (*Carriage Return*) sẽ được gửi kèm. Dấu chấm phẩy sẽ ngăn lại ký tự CR. Ký hiệu \$ có thể là một ký tự nào đó từ bộ ký tự IBM.

Lệnh INPUT \$ (n, #1) đọc n byte qua giao diện nối tiếp. Giá trị được đọc vào là một xâu và có thể được biến đổi bằng lệnh ASC (String \$) thành mẫu bit tương ứng. Để đọc vào một giá trị điện áp của một bộ biến đổi A/D thì cần có lệnh sau: u = ASC (INPUT \$) 1, #1)). Biến u sau đó chứa một giá trị bằng số nằm giữa 0 và 255; (Thông thường, ký hiệu điện áp được viết bằng U, nhưng ở đây điện áp được xem như một biến nên dùng ký hiệu u !).

Bằng PRINT#1, CHR\$(); cho phép gửi một mẫu bit trực tiếp qua cổng nối tiếp. Các thí dụ dưới đây sẽ minh hoạ rõ hơn các ứng dụng của lệnh PRINT#1:

Thí dụ:

Lệnh	Mẫu bit được gửi								Ghi chú
	D7	D6	D5	D4	D3	D2	D1	D0	
PRINT #1, " A ".	0	1	0	0	0	0	0	1	Mã ASCII của A
PRINT #1, " a ".	0	1	1	0	0	0	0	1	Mã ASCII của a
PRINT #1, CHR\$(65)	0	1	0	0	0	0	0	1	có giá trị 65
PRINT #1, CHR\$(0)	0	0	0	0	0	0	0	0	có giá trị 0
PRINT #1, CHR\$(255)	1	1	1	1	1	1	1	1	có giá trị 255
PRINT #1, CHR\$(15)	0	0	0	0	1	1	1	1	có giá trị 15

Ở cuối của một chương trình, trong đó đã dùng lệnh OPEN để mở giao diện nối tiếp, thì lại không được quên lệnh đóng CLOSE. Lệnh CLOSE sẽ đóng giao diện nối tiếp và sau đó đường dẫn DTR và RTS lại được đặt trở lại mức LOW (- 12 V).

1.2.3 BỘ GỬI NỐI TIẾP

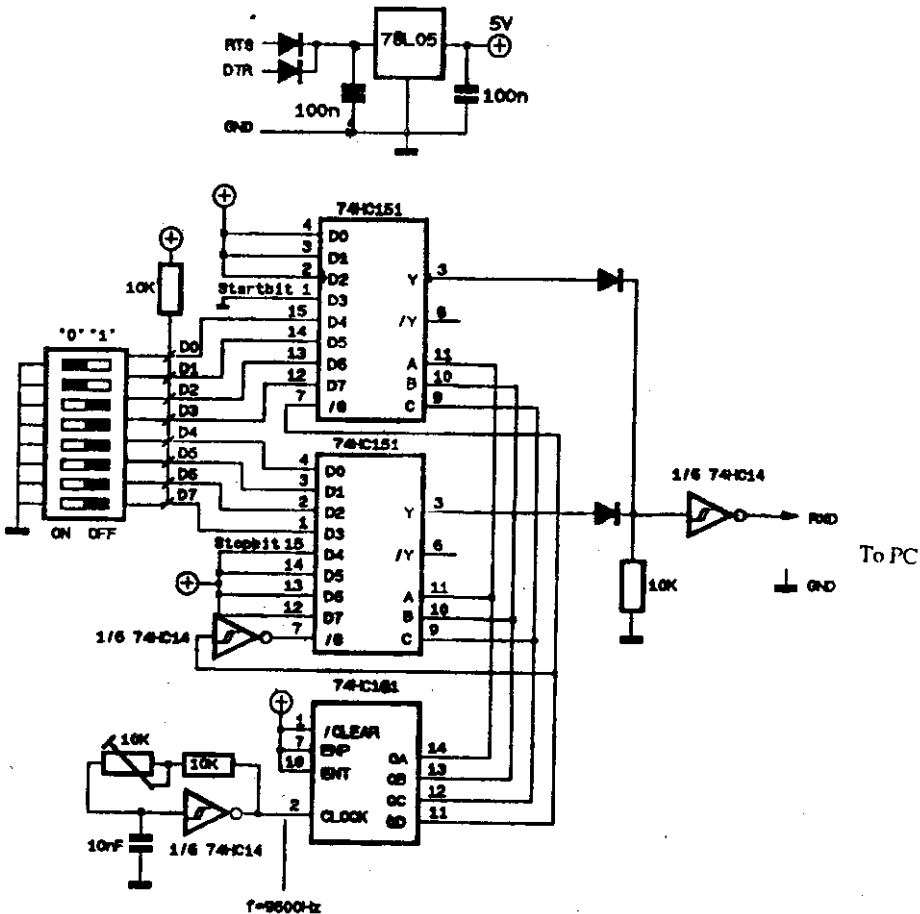
Hình 1.9 chỉ ra một bộ gửi nối tiếp đơn giản, bộ này gửi 8 bit dữ liệu kể sát song song qua giao diện nối tiếp tới máy tính PC. Khuôn mẫu truyền dữ liệu là bit khởi động (Startbit), 8 bit dữ liệu, một bit dừng (Stopbit) không có parity, 9.600 baud.

Ở đây để nối với máy tính PC cần có hai đường dẫn. Lối ra của bộ gửi nối tiếp được dẫn tới chân ra RxD của máy tính PC. Đồng thời hai chân mass được nối với nhau. Bằng mạch điện đơn giản này, tùy từng trường hợp, có thể cho phép theo một phương cách đơn giản hỏi lại điện áp của bộ biến đổi A/D hoặc mẫu bit điều khiển.

Mạch điện có thể được mô tả như sau:

Nguồn nuôi được lấy trực tiếp từ hai dây dẫn RTS và DTR. Khi hai giao diện đều đã được mở đường dẫn có điện áp + 12 V và mỗi đường dẫn có thể cung cấp một dòng trên 5 mA. Bộ ổn áp 78L05 tạo ra một điện áp + 5 V ổn định. Ở chân 2 của bộ đếm nhị phân 74HC161 có một tín hiệu với tần số 9.600 Hz, nghĩa là đúng bằng tốc độ baud. Ở lối ra QD... QA của bộ đếm nhị phân thay đổi từ 0000 lúc ban đầu với sự giữ nhịp của trạng thái logic của chúng phù hợp với hệ thống số nhị phân rồi đạt đến 1111. Sự giữ nhịp tiếp theo dẫn trở lại trạng thái đếm 0000

và cuộc chơi lại bắt đầu từ đầu. Các lối ra của bộ đếm dẫn đến các lối vào địa chỉ A, B, C của hai bộ quét động của vi mạch loại 74HC151. Các lối vào D0... D7 của nó được truyền qua kế tiếp nhau sang lối ra y. Bit khởi động mở đầu cuộc truyền dữ liệu. Sau đấy các Bit dữ liệu đã được đặt trước ở chuyển mạch DIP được dẫn tới lối ra y và nhờ vậy đến được máy tính PC.

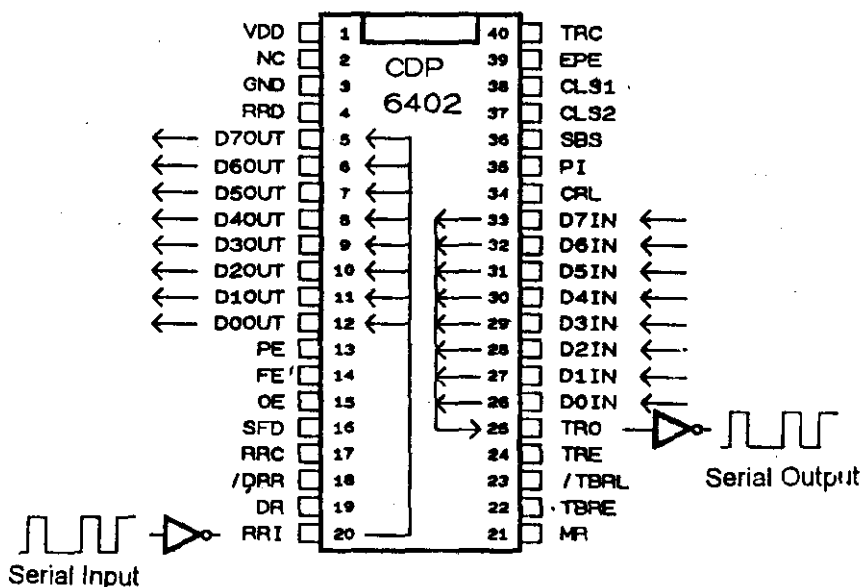


Hình 1.9 Bộ gửi nối tiếp.

Sau khi một byte dữ liệu đầy đủ đã được truyền là tiếp đến bit dừng, bit này được gửi cho đến khi bộ đếm 74HC161 lựa chọn lại bit khởi động ở trạng thái đếm 0011 (3 dec.). Cả hai điốt cùng với điện trở 10 kΩ tạo nên một cổng HOẶC (OR), cổng này kế tiếp hai lối ra của bộ quét động. Cổng HOẶC này là cần thiết bởi vì lối ra QD của bộ đếm luôn luôn chỉ kích hoạt một bộ quét động, trong khi lối ra kia lưu lại trong trạng thái điện trở cao.

Để đọc vào các byte dữ liệu, tất nhiên là trong mỗi ngôn ngữ lập trình ta có thể viết một chương trình tương ứng. Một khả năng khác để giao tiếp trực quan (Visualization) là dùng các chương trình chạy trong WINDOWS. Khi đó có thể thiết lập tất cả các thông số truyền một cách thuận tiện. Các dữ liệu nhận được sẽ được mô tả trên màn hình như là *Xâu*. Chẳng hạn như khi ta thiết lập ở bộ gửi nối tiếp mẫu bit với giá trị 65 (dec.) thì trên màn hình sẽ hiện lên chữ A.

1.2.4 BỘ PHÁT NHẬN-KHÔNG ĐỒNG BỘ VẠN NĂNG CDP 6402



Hình 1.10 Bố trí chân của UART CDP 6402.

Những tính chất sau đặc trưng cho vi mạch CDP 6402

- ❑ Công suất tiêu thụ không đáng kể: 7,5 mW ở 3,2 MHz và $V_{cc} = 5$ V.
- ❑ Tốc độ baud: đến 200 kBaud khi điện áp nguồn nuôi là + 5 V.
đến 400 kBaud khi điện áp nguồn nuôi là + 10 V.
- ❑ Điện áp nguồn nuôi từ 4 V đến 10,5 V.
- ❑ Đặt khuôn mẫu truyền dữ liệu bằng phần cứng.
- ❑ Sử dụng đơn giản.
- ❑ Giá thành gần 10 USD (năm 1996).

Như ở trong mục trước đã đề cập đến, để truyền dữ liệu qua giao diện nối tiếp đã có các chip được tích hợp ở mức độ cao. Một linh kiện loại này là bộ UART CDP 6402 của hãng HARIS. Bộ UART (Universal Asynchronous Receiver/ Transmitter) này chứa trên cùng chip một bộ gửi và bộ nhận nối tiếp hoạt động hoàn toàn độc lập với nhau. Bộ gửi nối tiếp truyền đi sau một xung khởi động các dữ liệu xếp kề sát qua một đường dẫn tới bộ nhận và gửi kèm theo một cách tự động các bit khởi động và bit dừng. Bên bộ nhận lại có được các dữ liệu đã nối tiếp đến để sử dụng song song. Điểm đáng lưu ý ở vi mạch này là khuôn mẫu truyền dữ liệu có thể được thiết lập trước bằng phần cứng qua các mức logic ở các chân. Nhờ vậy mà vi mạch này có thể được sử dụng một cách vạn năng.

Bảng dưới đây mô tả chức năng của các chân riêng biệt:

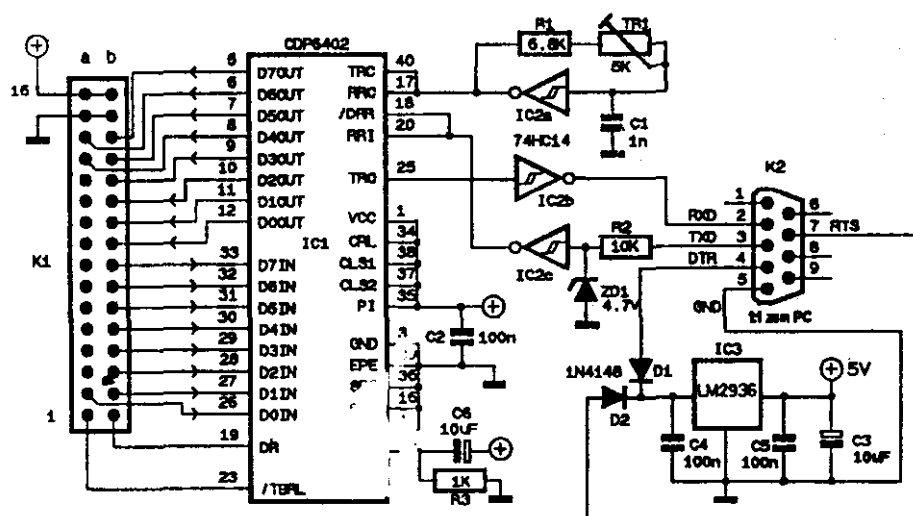
Chân	Ký hiệu	Mô tả
1	VDD	Cực dương của nguồn nuôi
2	NC	Không dùng
3	GND	Mass, Đất, 0 V
4	RRD	Receive Register Disable Khi tín hiệu này dẫn đến mức High thì các đường dẫn lối ra D0OUT đến D7OUT chuyển sang trạng thái điện trở cao
5	D7 OUT	Các bit dữ liệu đã đến theo cách nối tiếp ở chân 20 sẽ xuất hiện theo cách song song ở các lối ra ba trạng thái (tristate): D7OUT đến D0OUT.
6	D6 OUT	
7	D5 OUT	
8	D4 OUT	
9	D3 OUT	
10	D2 OUT	
11	D1 OUT	
12	D0 OUT	
13	PE	Parity Error: Sai số chẵn lẻ

		Một mức logic 1 ở chân này báo hiệu là bit chặn lẻ đã được lập trình không đồng nhất với bit nhận được. Nếu như bit chặn lẻ không được kích hoạt thì chân này nằm ở mức Low
14	FE	Sai số Framming Mức High ở chân này báo hiệu là bit dừng đầu tiên là không có giá trị. FE giữ nguyên High cho đến khi nhận được một bit dừng có giá trị.
15	OE	Sai số Overrun. OE sau đó trở nên High, nếu như một byte mới đã được nhận. Trước khi byte cũ được đọc từ thanh ghi nhận.
16	SFD	Status Flag Disable. Một mức cao ở chân này có nghĩa là các lối ra PE, FE, OE, DE và TBRL trở nên có điện trở cao.
17	RRC	Receiver Register Clock Ở RRC, tín hiệu giữ nhịp của bộ nhận nối tiếp được dẫn đến. Tần số cần phải được thiết lập ở giá trị 16 lần lớn hơn tốc độ
18	DRR	Data Received Reset Một xung Low ở chân này đặt DR (Data Received) trở lại Low.
19	DR	Data Received DR = 1 báo hiệu là các dữ liệu đã được nhận một cách đầy đủ và có mặt ở các lối ra D7OUT đến D0OUT. Trước khi một byte dữ liệu có giá trị tiếp theo có thể được báo hiệu, tín hiệu DR cần phải được đặt lại bằng một xung âm ở DRR (chân 18).
20	RRI	Receiver Register Input Ở chân này tín hiệu lối vào nối tiếp được dẫn đến.

21	MR	Master Reset Với mức High, việc Reset linh kiện sẽ được thực hiện. PE, FE, OE và DR được đặt lại, trong khi TRE, TBRE và TRO được đặt lên mức cao.
22	TBRE	Transmitter Buffer Register Empty Một mức cao ở chân này báo hiệu là thanh ghi của bộ gửi đang trống và sẵn sàng tiếp nhận dữ liệu mới.
23	TBRL	Transmitter Control Register Load Một xung Low sẽ xóa để gửi đi các bit dữ liệu. Bằng sườn dương các dữ liệu xếp kế sát, song song D7IN đến D0IN sẽ được truyền vào thanh ghi của bộ gửi và sau đó được gửi đến bên nhận theo cách nối tiếp với bit khởi động và bit dừng.
24	TRE	Transmitter Register Empty Một mức cao báo hiệu là linh kiện đã làm xong việc gửi
25	TRO	Transmitter Register Output. Các bit dữ liệu xếp kế sát song song D0IN đến D7IN được gửi bao gồm bit khởi động và bit dừng qua đường dẫn TRO tới bên nhận.
26	D0IN	Các bit dữ liệu ở các lối vào này được gửi trực tiếp đến nơi nhận. • D0IN là LSB • D7IN là MSB
27	D1IN	
28	D2IN	
29	D3IN	
30	D4IN	
31	D5IN	
32	D6IN	
33	D7IN	
34	CRL	Control Register Load: Nạp thanh ghi điều khiển. Một mức High nạp các bit điều khiển vào thanh ghi điều khiển.

1.2.5 MÔĐUN CƠ SỞ VÀO/ RA 8 BIT KHÔNG CÓ NGUỒN NUÔI PHỤ

Dưới đây là một mạch có khả năng hoạt động như là môđun ghép nối với giao diện nối tiếp của máy tính PC để nhập và xuất dữ liệu từ 8 tín hiệu TTL. Môđun này có khả năng ghép nối với bộ biến đổi A/D, bộ biến đổi D/A, khối đếm và các bộ phận ngoại vi khác và nhờ vậy cho phép nó được sử dụng một cách vạn năng. Đặc biệt ở đây phải nhấn mạnh là môđun không đòi hỏi có nguồn nuôi phụ. Nguồn nuôi cho môđun này được lấy trực tiếp từ giao diện RS 232 ở máy tính PC. Mạch điện đòi hỏi dòng điện tiêu thụ cỡ mili ampe. Ngoài ra còn có thể ghép nối thêm với các mạch phụ trợ, nên dòng tiêu thụ tổng cộng cực đại cỡ 10 mA. Trên hình 1.11 chỉ ra sơ đồ điện của môđun.



Hình 1.11 Sơ đồ mạch của môđun cơ sở dùng cho cổng nối tiếp không cần nguồn nuôi phụ.

Hạt nhân của mạch điện này là vi mạch UART CDP 6402, linh kiện này đòi hỏi rất ít các linh kiện phụ ở mạch ngoài. Nguồn nuôi được lấy ra từ hai đường tín hiệu DTR và RTS. Điện áp khi ở trạng thái không quá tải là + 12 V đi qua hai diốt đến lối vào của bộ ổn áp LM 2936 của hãng NSC. Vi mạch ổn áp này có hai ưu điểm nổi bật: dòng tiêu thụ của vi mạch này chỉ cỡ 100 μ A, còn sự khác nhau nhỏ nhất giữa điện áp lối vào và lối ra có thể đạt đến 0,2 V. Nhờ vậy vi mạch LM 2936 có thể được xem là đã được dự tính trước để dùng cho mạch điện này. Tín hiệu giữ nhịp dùng cho UART được lấy ra từ một bộ dao động bao gồm một bộ

đảo bằng trigger Schmitt, một tụ điện và một điện trở như minh họa trên hình 1.11. Với các giá trị của tụ điện và điện trở như ghi trên hình vẽ, tần số tương ứng cỡ 153 kHz. Một sự đồng chỉnh chính xác để có được tần số 153.600 Hz được thực hiện bằng trimmer TR1. Tần số đã được thiết lập tương ứng với 16 lần tốc độ baud có giá trị 9.600 baud.

Các bit dữ liệu đã nhận được từ máy tính PC qua đường dẫn TxD ở ổ cắm 9 chân đạt đến môđun cơ sở. Bởi vì tín hiệu này có mức điện áp nằm giữa + 12 V và - 12 V. Còn môđun cơ sở làm việc giữa mức điện áp 0 V và 5 V nên cần có một sự thích ứng về điện áp. Người ta cho tín hiệu truyền qua điện trở R2 tới diốt ổn áp ZD1 (4,7 V) để hạn chế biên độ điện áp cực đại xuống dưới + 5 V. Nhờ vậy mà ở lối vào của bộ đảo tiếp theo mức điện áp trở lại tương thích TTL. Lối ra nối tiếp TRO của bộ UART được dẫn qua một bộ đảo đến chân ra RxD ở phích cắm 9 chân. Để nối ghép với máy tính PC không được phép dùng một cáp nối môđun 0 với các đường dẫn tín hiệu bất chéo, mà nên dùng cáp nối 1:1.

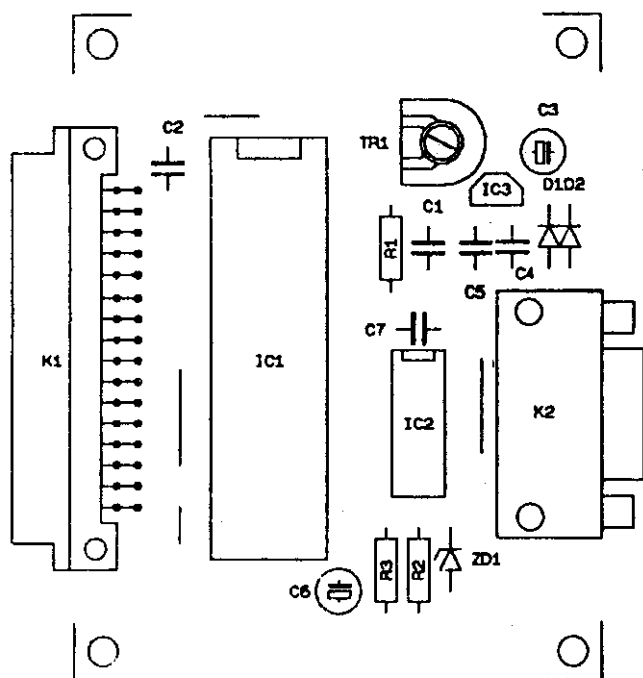
Việc lắp ráp bộ UART tiến hành theo cách thông thường. Đối với việc truyền dữ liệu các mạch của bit điều khiển CLS1 = 1, CLS2 = 1, PI = 1, EPE = 0 và SBS = 0 chỉ ra các khuôn mẫu dữ liệu sau đây: 8 bit dữ liệu, một bit khởi động, một bit dừng, không có bit chặn lẻ. Khuôn mẫu này cũng phải được thiết lập bên máy tính PC vì nếu không thì bộ UART sẽ nhận được những dữ liệu sai.

Nhờ có C6 và R3 mà cứ mỗi lần bật điện vào máy lại xảy ra sự đặt lại (Power-On-Reset). Các dữ liệu lối ra D7OUT đến D0OUT cũng như byte dữ liệu đã được gửi D7IN đến D0IN có mặt ở ổ cắm 32 chân det.

Mạch ở chân ra DDR (chân 18) là rất đáng quan tâm. Để hiểu được ta cần chú ý: Khi các dữ liệu đã nhận được một cách đầy đủ thì tín hiệu DR chuyển sang mức High. Chỉ khi nào nhận được một xung Low ở DDR thì DR mới được đặt trở lại mức Low. Xung âm này được lấy ra từ byte dữ liệu tới tiếp sau đấy. Muốn thế người ta sử dụng bit khởi động, bit này sau bộ đảo sẽ có mức Low đang cần có.

Trên hình 1.12 và 1.13 mô tả sự sắp xếp linh kiện và sơ đồ mạch in của môđun cơ sở.

Các linh kiện cần thiết cho mạch được liệt kê như sau đây:



Hình 1.12 Sắp xếp linh kiện trên môđun cơ sở dùng cho cổng nối tiếp không cần nguồn nuôi phụ.

Linh kiện bán dẫn:

IC1: CDP 6402
 IC2: 74HC14
 IC3: LM 2936Z-5.0
 (hoặc 78L05)

Tụ điện:

C1: 1 nF
 C2, C4, C5: 100 nF
 C3, C6: 10 μ F / 16 V

Điện trở:

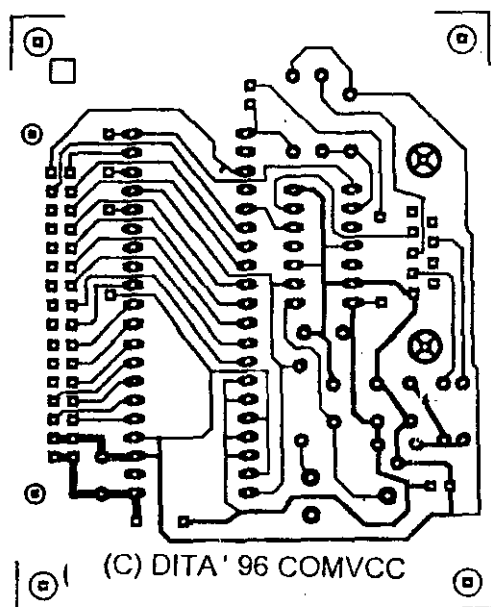
R1: 6,8 k Ω
 R2: 10 k Ω
 R3: 1 k Ω
 TR1: 5 k Ω

Điốt bán dẫn:

D1, D2: 1N 4148
 ZD1: 4,7 V

Phích cắm:

K1: dẹt 32 chân
 K2: 9 chân



Hình 1.13 Sơ đồ mạch in của môđun cơ sở dùng cho cổng nối tiếp không cần nguồn nuôi phụ.

Môđun cơ sở này có thể trao đổi với máy tính PC như thế nào ?

Trong QBasic công việc này được thực hiện một cách rất đơn giản. Bằng lệnh `PRINT #1, CHR $ (outbyte)`, một mẫu bit mong muốn sẽ được các đường dẫn lối ra `D7OUT` đến `D0OUT` của môđun cơ sở. Biến `outbyte` chứa đựng một con số trong hệ đếm mười nằm giữa 0 và 255. Việc mô tả chi tiết về vấn đề này đã được đề cập đến ở trên đây.

Chỉ dẫn: Điều kiện để đọc vào một byte dữ liệu là một xung Low ở tín hiệu `TBRL`. Bằng sườn dương của xung giữ nhịp các dữ liệu xếp kê sát song song `D7IN` đến `D0IN` được chuyển vào thanh ghi bên gửi và sau đấy được gửi nối tiếp với bit khởi động và bit dừng tới máy tính PC.

Việc đọc vào của các bit dữ liệu xếp kê sát song song `D7IN` đến `D0IN` diễn ra sau lệnh `inbyte = ASC (INOUT $ (1, #1))`. Sau đó biến `inbyte` chứa đựng giá trị của các byte dữ liệu như là một giá trị trong hệ đếm mười.

Tín hiệu này cũng có mặt ở ổ cắm 32 chân det. Chương trình dùng làm thí dụ dưới đây một lần nữa làm sáng tỏ hơn việc trao đổi thông tin giữa máy tính PC và môđun cơ sở diễn ra như thế nào.

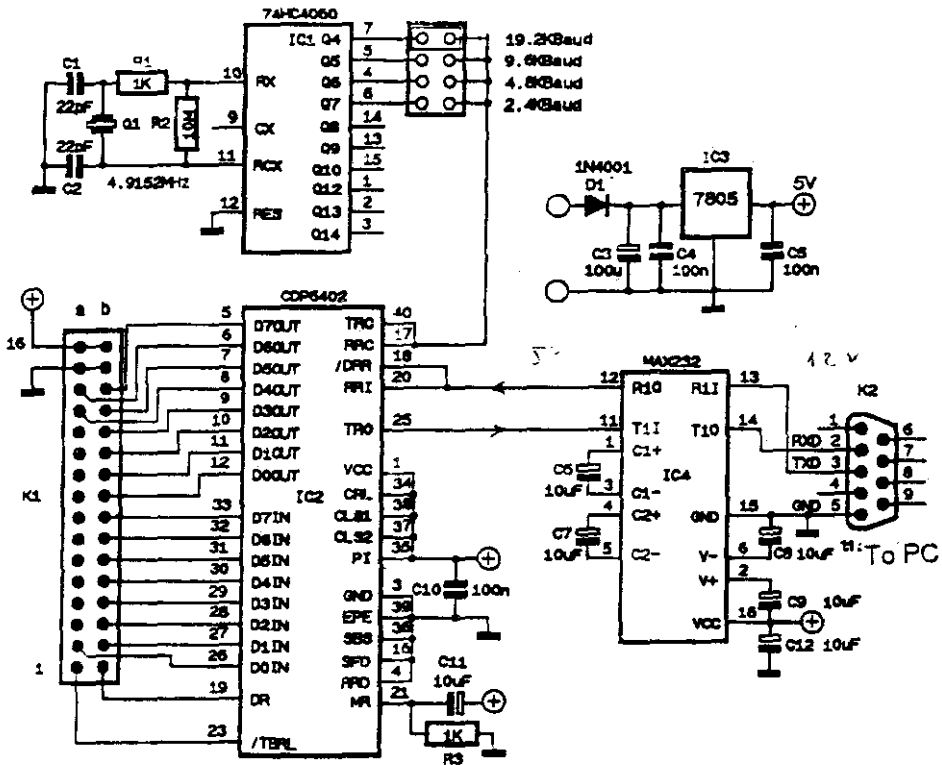
```
'=====
'  Program: COMBSP
'  Function: Chương trình này dùng làm thí dụ minh
'  hoạ một cách rõ ràng hơn xem môđun cơ sở dùng cho
'  giao diện nối tiếp ở mục 1,2,5 đến 1,2,7 có thể
'  được trao đổi từ máy tính PC như thế nào ?
'
'  Hardware: Cần có một môđun cơ sở dùng cho giao
'  diện nối tiếp như ở mục 1,2,5 đến 1,2,7
'-----9600 Baud an COM 2-----
OPEN "com2: 9600,N8,1,CS,DS" FOR RANDOM AS #1
CLS
PRINT "Bạn hãy cho vào con số hệ đếm 10 của byte"
PRINT "dữ liệu được đưa ra (0... 255)"
INPUT outbyte
'===== Xuất ra các byte dữ liệu =====
PRINT #1, CHR$(outbyte);
'===== Kết thúc việc xuất dữ liệu =====
Bây giờ dữ liệu được đọc vào
outbyte = outbyte AND (255 - 1)
PRINT #1, CHR$(outbyte);
outbyte = outbyte OR 1
PRINT #1, CHR$(outbyte);
'----- Đọc vào các byte dữ liệu -----
inbyte = ASC (INPUT$(1, #1))
PRINT "Bit dữ liệu đã đọc vào là:", inbyte

END
```

1.2.6 MÔĐUN CƠ SỞ VÀO/ RA 8 BIT VỚI MAX 232

Trong mục này sẽ giới thiệu một môđun cơ sở khác nữa để ghép nối với giao diện nối tiếp của máy tính PC. Môđun này tương thích hoàn

toàn với ổ cắm 32 chân đặt ở môđun trình bày trong mục 1.2.5. Ở đây, môđun này cũng cho phép cả đọc vào cũng như đưa ra 8 tín hiệu TTL qua giao diện nối tiếp của máy tính PC. Sự khác nhau so với môđun trước đây là ở những thông số kỹ thuật đã được cải tiến.

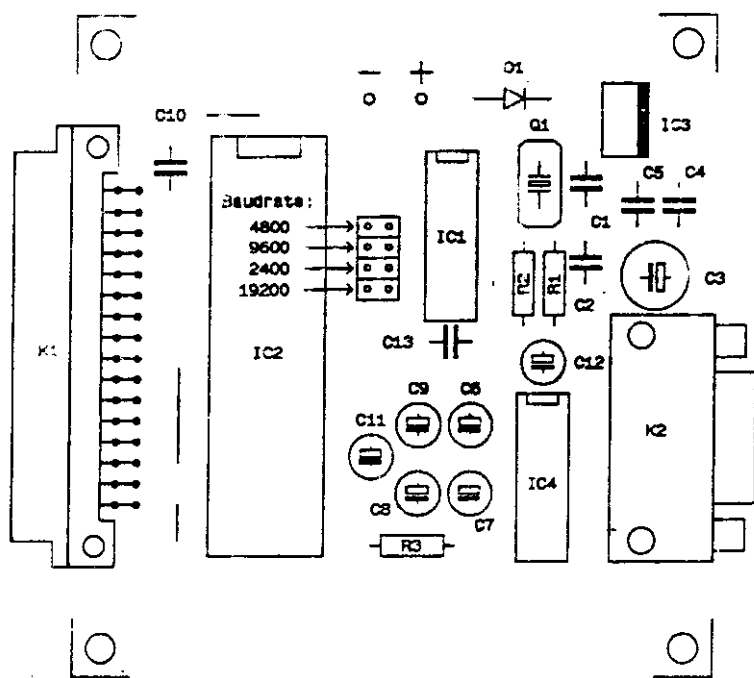


Hình 1.14 Sơ đồ mạch của môđun cơ sở dùng cho cổng nối tiếp với MAX 232.

Môđun cơ sở trên hình 1.14 có chứa vi mạch MAX 232 của hãng MAXIM để thích ứng tín hiệu ở mức (+ 12 V, - 12 V) trên giao diện RS 232.

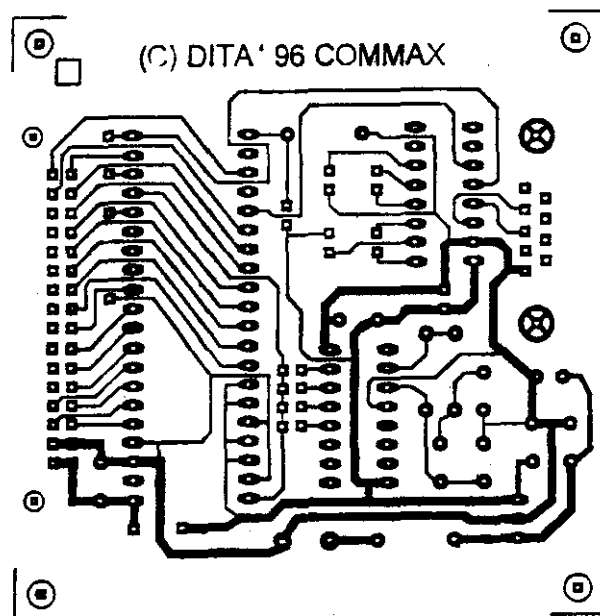
Vi mạch này nhận mức RS 232 đã được gửi từ máy tính PC và biến đổi tín hiệu này thành tín hiệu TTL, để rồi sau đó dẫn tới bộ UART CDP 6402. Tín hiệu từ bộ UART được biến đổi thành mức + 12 V / - 12 V và gửi tới máy tính PC. Theo cách này khoảng cách giữa máy tính PC và môđun cơ sở có thể đạt đến trên 20 m. Một ưu điểm khác nữa của môđun cơ sở là khả năng thiết lập tốc độ baud. Muốn thế, trong mạch

đã sử dụng bộ đếm nhị phân 74HC4060, bộ đếm này có những lối vào phụ thêm để nối trực tiếp với một bộ cộng hưởng thạch anh.



Hình 1.15 Sắp xếp linh kiện trên môđun cơ sở dùng cho cổng nối tiếp với MAX 232.

Bằng việc thay đổi vị trí một cầu nối (Jumper) có thể đặt một cách chính xác các giá trị 2.400; 4.800; 9.600 và 19.200 baud. Một sự tinh chỉnh bổ sung bằng một trimmer ở đây là không cần thiết. Cần chú ý rằng tần số ở chân 17 và 40 của bộ UART cần phải tương ứng với 16 lần tốc độ baud. Chẳng hạn như ở lối ra Q4 (của IC1 74HC4060) có các xung vuông góc với tần số $4,9152 \text{ MHz} : 24 = 307.200 \text{ Hz}$. Tần số này ứng với 16 lần tốc độ baud 19.200 baud. Nguồn nuôi được thiết kế với bộ ổn áp 7805 để có được điện áp 5 V ổn định ở lối ra. Còn ở lối vào cần có một điện áp trên 7,5 V, điện áp này có thể lấy từ một bộ nắn dòng hoặc từ một bộ pin 9 V. Trên hình 1.15 và 1.16 là sơ đồ sắp xếp linh kiện và mạch in của môđun.



Hình 1.16 Sơ đồ mạch in của môđun cơ sở dùng cho cổng nối tiếp với MAX 232.

Các linh kiện cần thiết cho mạch được liệt kê dưới đây:

Linh kiện bán dẫn:

IC1: 74HC4060
 IC2: CDP 6402
 IC3: 7805
 IC4: MAX 232

Tụ điện:

C1, C2: 22 pF
 C3: 100 μ F / 35 V
 C4, C5, C10, C13: 100 nF
 C6, C7, C8, C9, C11, C12:
 10 μ F / 16 V

Điện trở:

R1, R3: 1 k Ω
 R2: 10 M Ω

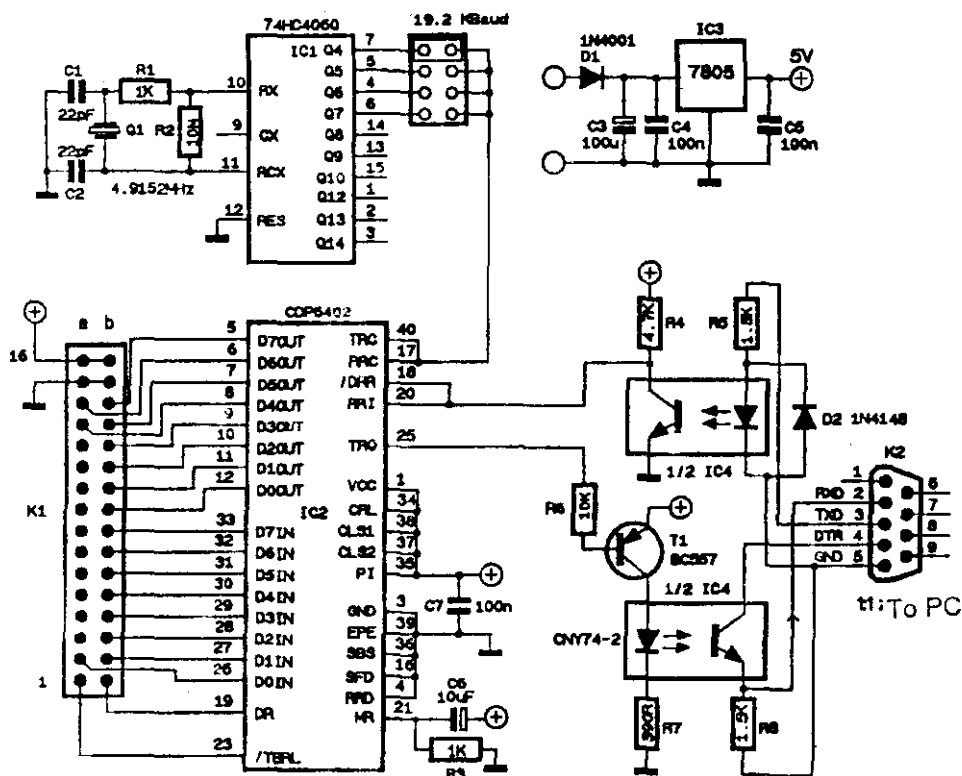
Phích cắm:

K1: dẹt 32 chân
 K2: 9 chân

Điốt bán dẫn:

D1, D2: 1N 4148
 ZD1: 4,7 V

1.2.7 MÔĐUN CƠ SỞ VÀO/ RA 8 BIT CÁCH LY VỀ MẶT ĐIỆN

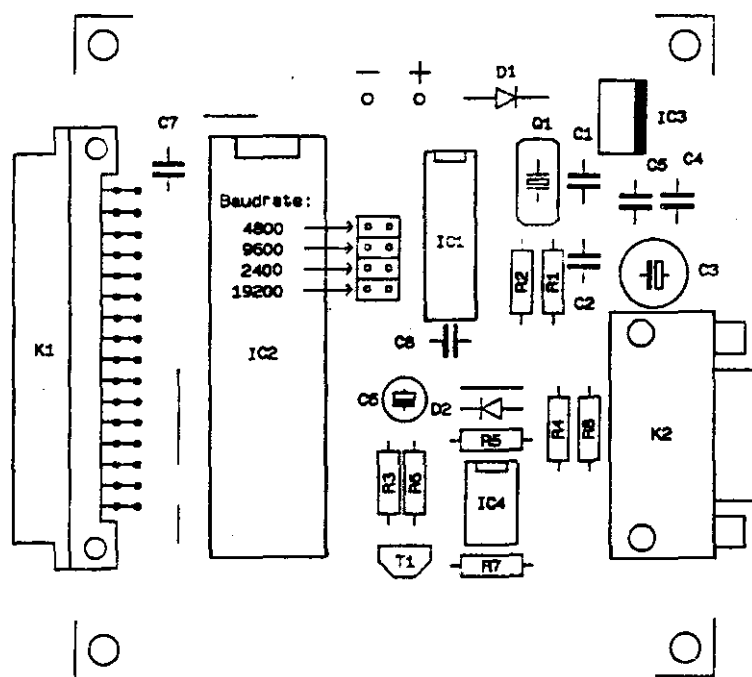


Hình 1.17 Sơ đồ mạch của module cơ sở dùng cho cổng nối tiếp với sự cách ly về điện.

Trong mục này một module khác nữa có thể nối với giao diện nối tiếp sẽ được giới thiệu. Sự khác nhau cơ bản so với các module đã giới thiệu trước đây là sự cách ly về điện giữa máy tính PC và module. Nhờ có linh kiện ghép nối quang (optocoupler) mà tất cả các tín hiệu đến với máy tính PC được cách ly. Do vậy mà không có sự ghép nối trực tiếp về mặt điện giữa đối tượng đo và máy tính PC. Nếu không kể đến tính chất này thì có thể xem như module này đã được mô tả trong mục trên đây, còn sơ đồ điện được minh họa trên hình 1.17. Những đặc điểm của module này có thể tóm tắt như sau:

- ❑ Thiết lập chính xác tốc độ baud bằng bộ cộng hưởng thạch anh.
- ❑ Bốn tốc độ baud đặt được bằng cầu nối: 2.400; 4.800; 9.600 và 19.200 baud.

- ☐ Nguồn nuôi dùng qua bộ nắn điện hoặc bộ pin 9 V.
- ☐ Ngăn cách về điện với máy tính PC.



Hình 1.18 Sắp xếp các linh kiện môđun cơ sở vào/ ra 8 bit cách ly về điện.

Các linh kiện cần dùng để lắp ráp mạch là:

Linh kiện bán dẫn:

IC1: 74HC4060
 IC2: CDP 6402
 IC3: 7805
 IC4: CNY 74-2
 T1: BC 557

Điện trở:

R1, R3: 1 k Ω
 R2: 10 M Ω
 R4: 4,7 k Ω
 R5, R8: 1,5 k Ω
 R6: 10 k Ω
 R7: 390 Ω

Tụ điện:

C1, C2: 22 pF
 C3: 100 μ F / 35 V
 C4, C5, C7, C8: 100 nF
 C6: 10 μ F / 16 V

Điốt bán dẫn:

D1: 1N 4001
 D2: 1N 4148

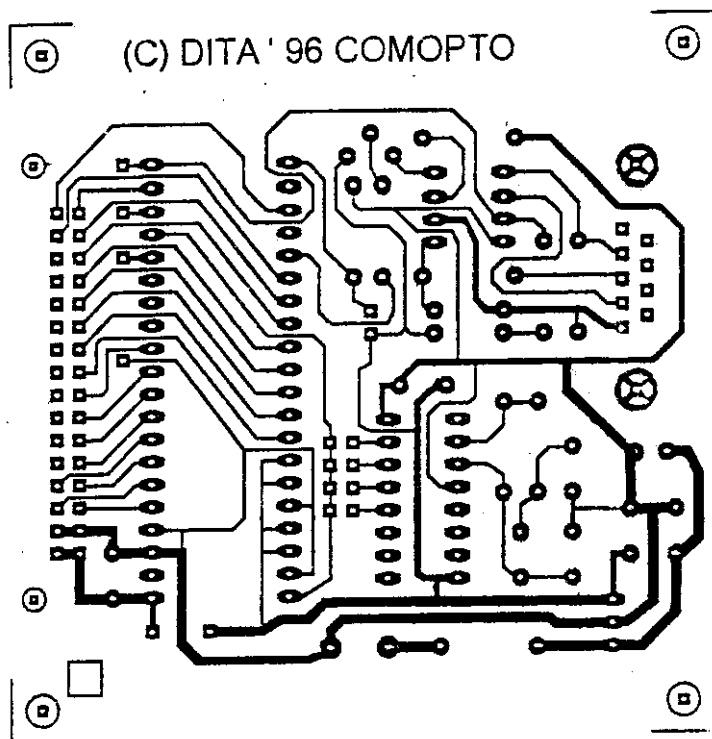
Phích cắm:

Thạch anh:

K1: dẹt 32 chân

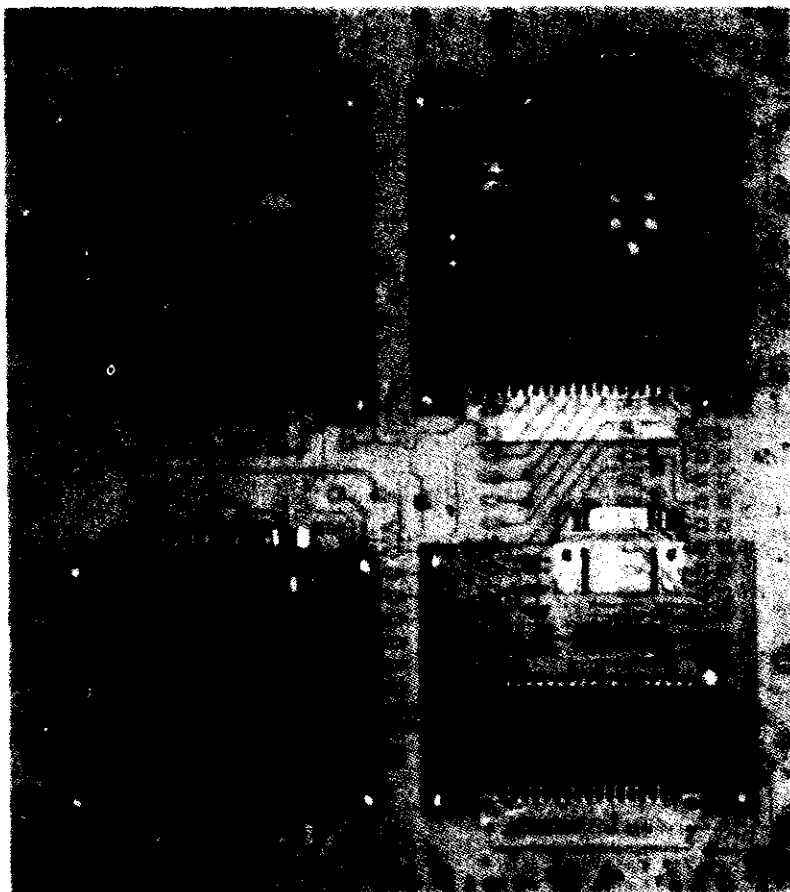
Q1: 4,9152 MHz

K2: 9 chân



Hình 1.19 Sơ đồ mạch in môđun cơ sở vào/ ra 8 bit cách ly về điện.

Trên hình 1.18 và 1.19 là sự sắp xếp linh kiện và mạch in của môđun cơ sở; còn trên hình 1.20 là hình dáng bề ngoài của các môđun cơ sở đã được giới thiệu.



Hình 1.20 Hình dáng bên ngoài của một số môđun cơ sở.

1.3 CÁC RÃNH CẮM TRONG MÁY TÍNH PC

1.3.1 SỰ SẮP XẾP CHÂN TRÊN RÃNH CẮM

Khi một máy tính xuất xưởng thì cả nhà sản xuất và người dùng đều ngầm hiểu là cấu hình chưa hẳn đã hoàn chỉnh, mà tùy từng mục đích sử dụng có thể đưa thêm vào các bản mạch (card) ghép nối để mở rộng khả năng đáp ứng của máy tính. Bên trong máy, ngoài các rãnh cắm dùng cho card vào/ ra (I/O), card màn hình, vẫn còn những rãnh

cắm để trống. Các rãnh cắm này được tiếp tục dùng để ghép nối các bản mạch cắm thêm vào với máy tính PC.

Ở máy tính PC/XT rãnh cắm trong máy tính chỉ có một loại với độ rộng bus là 8 bit và tuân theo tiêu chuẩn ISA (viết tắt từ Industry Standard Architecture). Từ máy tính AT trở đi việc bố trí chân trên rãnh cắm trở nên phức tạp hơn, tùy theo tiêu chuẩn được lựa chọn khi chế tạo máy tính. Các loại rãnh cắm theo những tiêu chuẩn khác nhau có thể kể ra như sau:

- ☐ Rãnh cắm 16 bit theo chuẩn ISA (Industry Standard Architecture).
- ☐ Rãnh cắm PS/2 với 16 bit theo chuẩn MCA
(Micro Channel Architecture).
- ☐ Rãnh cắm PS/2 với 32 bit theo chuẩn MCA
(Micro Channel Architecture).
- ☐ Rãnh cắm 32 bit theo chuẩn EISA
(Extended Industry Standard Architecture).
- ☐ Rãnh cắm 32 bit theo chuẩn VESA VLB (VESA Local Bus Standard).
- ☐ Rãnh cắm 32 bit theo chuẩn PCI
(Peripheral Component Interconnect-Standard).

Cho đến nay phần lớn các card ghép nối dùng trong kỹ thuật đo lường và điều khiển đều được chế tạo để đặt vào rãnh cắm theo chuẩn ISA, cho nên dưới đây sẽ được giới thiệu chi tiết hơn. Trong các trường hợp cần tìm hiểu chi tiết về các rãnh cắm khác, bạn đọc có thể xem thêm phần phụ lục 3 ở cuối sách.

Rãnh cắm theo chuẩn ISA:

Thông thường rãnh cắm có 62 đường tín hiệu dùng cho mục đích thông tin với một card cắm vào. Về cơ bản các đường tín hiệu này được chia ra thành các đường dẫn tín hiệu, đường dẫn địa chỉ và đường dẫn điều khiển. Bởi vì ngay từ các máy tính PC/ XT đã sẵn có các rãnh cắm 62 chân này, trên đó có 8 đường dẫn dữ liệu, nên đôi khi người ta cũng gọi luôn rãnh cắm 62 chân này là rãnh cắm 8 bit. Chỉ những card 8 bit mới được cắm vào rãnh này. Bảng dưới đây chỉ ra sự sắp xếp chân ra của rãnh cắm mở rộng 8 bit.

	Phía mạch in	Phía linh kiện	
GND	B01	A01	/ IOCHCK
Reset	B02	A02	D7
+ 5 V	B03	A03	D6
IRQ2	B04	A04	D5
- 5 V	B05	A05	D4
DREQ2	B06	A06	D3
- 12 V	B07	A07	D2
Dự trữ	B08	A08	D1
+ 12 V	B09	A09	D0
GND	B10	A10	/ IOCHRDY
/MEMW	B11	A11	AEN
/MEMR	B12	A12	A19
/IOW	B13	A13	A18
/IOR	B14	A14	A17
/DACK3	B15	A15	A16
DREQ3	B16	A16	A15
/DACK1	B17	A17	A14
DREQ1	B18	A18	A13
/DACK0	B19	A19	A12
CLK	B20	A20	A11
IRQ7	B21	A21	A10
IRQ6	B22	A22	A9
IRQ5	B23	A23	A8
IRQ4	B24	A24	A7
IRQ3	B25	A25	A6
/DACK2	B26	A26	A5
TC	B27	A27	A4
ALE	B28	A28	A3
+ 5 V	B29	A29	A2
OSC	B30	A30	A1
GND	B31	A31	A0

Về sau máy tính PC/ AT ra đời chúng có thêm một rãnh thứ hai nằm thẳng hàng với rãnh 8 bit kể trên và có 36 chân.

Trên rãnh này có chứa các tín hiệu 16 bit. Nên khi có thêm rãnh cắm này thì người ta gọi chung cả hai rãnh là rãnh cắm 16 bit. Các rãnh cắm từ 32 bit trở lên dùng để ghép thêm vào những card có chất lượng rất cao, và vì vậy trong phạm vi nội dung của cuốn sách này chưa được quan tâm.

Rãnh cắm 16 bit bao gồm rãnh cắm 8 bit và có thêm một rãnh cắm thứ hai. Dưới đây là sự sắp xếp chân ra của rãnh cắm thứ hai:

	Phía mạch in	Phía linh kiện	
/ MEM CS16	D01	C01	SBHE
/ I-O CS16	D02	C02	LA23
IRQ 10	D03	C03	LA22
IRQ 11	D04	C04	LA21
IRQ 12	D05	C05	LA20
IRQ 13	D06	C06	LA19
IRQ 14	D07	C07	LA18
7DARK 4	D08	C08	LA17
DRQ 0	D09	C09	7MEMR
7DARK 5	D10	C10	7MEMW
DRQ 5	D11	C11	SD8
7DARK 6	D12	C12	SD9
DRQ 6	D13	C13	SD10
7DARK 7	D14	C14	SD11
DRQ 7	D15	C15	SD12
+ 5 V	D16	C16	SD13
/MASTER	D17	C17	SD14
GND	D18	C18	SD15

Kích thước lớn nhất của các card ISA 8 bit là:

- ☐ Chiều cao 106,7 mm (hay 4,2 inhxơ).
- ☐ Chiều dài 333,5 mm (hay 13.13 inhxơ).

☐ Chiều dày - kể cả linh kiện - 12,7 mm (hay 0,5 inhxơ).

Còn kích thước lớn nhất của các card ISA 16 bit là:

☐ Chiều cao 121,92 mm (hay 4,8 inhxơ).

☐ Chiều dài 333,5 mm (hay 13,13 inhxơ).

☐ Chiều dày - kể cả linh kiện - 12,7 mm (hay 0,5 inhxơ).

Sự khác nhau giữa hai kích thước này chỉ ở chiều cao mà nguyên nhân là do vỏ của máy tính AT được thiết kế cao hơn loại XT đôi chút.

Từ cách sắp xếp chân ra, rõ ràng là 62 đường tín hiệu nằm cả ở mặt hàn thiếc cũng như mặt sắp đặt linh kiện. Do đó các bản mạch (card) cắm thêm vào bao giờ cũng là những card mạch in hai mặt. Bên cạnh 8 đường dẫn dữ liệu trên card còn có 20 đường dẫn địa chỉ từ A0 đến A19. Trong cuốn sách này các đường dẫn điều khiển còn lại cũng ít được quan tâm đến. Và không phải tất cả các tín hiệu điều khiển dùng để tạo nên một card mở rộng đều được sử dụng, nên thường thì ta cũng chỉ cần quan tâm đến các tín hiệu sau:

Tín hiệu	Hướng	Mô tả
Reset	Lối ra	Sau khi bật máy tính hoặc sau khi ngắt điện, đường dẫn Reset sẽ kích hoạt trong thời gian ngắn để đưa card đã được cắm vào đến một trạng thái ban đầu xác định.
/IOW	Lối ra	Input / Output / Write Tín hiệu này sẽ kích hoạt khi truy nhập ghi lên một card mở rộng. Mức thấp chỉ ra rằng các dữ liệu có giá trị đang chờ để đưa ra ở bus dữ liệu. Các dữ liệu được đón nhận bằng sườn trước.
/IOR	Lối ra	Input / Output / Read Mức thấp của đường dẫn địa chỉ này báo hiệu sự truy nhập đọc trên một card mở rộng. Trong thời gian này các dữ liệu có giá trị cần phải sắp xếp để rồi sau đó được đón

AEN	Lỗi ra	nhận bằng sườn trước. Address Enable Đường dẫn điều khiển AEN dùng để phân biệt chu trình truy nhập DMA và chu trình truy nhập bộ xử lý. Ở mức cao (High) DMA giám sát qua bus địa chỉ và bus dữ liệu; đường dẫn có hiệu lực ở mức thấp (Low). Đường dẫn này cần phải được sử dụng cho quá trình giải mã địa chỉ bởi card mở rộng.
-----	--------	---

1.3.2 SỰ GIẢI MÃ ĐỊA CHỈ VÀ NỐI BUS DỮ LIỆU

Vùng vào/ ra của máy tính PC đã chiếm giữ 64 kByte của bộ nhớ tổng cộng với dung lượng hàng vài MByte trở lên. Vì vậy, vùng vào/ ra của một card mở rộng không được phép bao trùm lên vùng địa chỉ vào/ ra của máy tính. Khi đưa một card mở rộng vào sử dụng, thì việc đầu tiên là ta phải lưu tâm đến điểm đáng chú ý này. Bảng dưới đây sẽ chỉ ra sự sắp xếp của vùng địa chỉ vào/ ra của máy tính PC/AT.

Địa chỉ (Hex) vào / ra	Chức năng
000 - 01F	Bộ điều khiển DMA 1 (8232)
020 - 03F	Bộ điều khiển ngắt (8259)
040 - 04H	Bộ phát thời gian (8254)
060 - 06F	Bộ kiểm tra bàn phím (8242)
070 - 07F	Đồng hồ thời gian thực (MC 146818)
080 - 09F	Thanh ghi trạng DMA (LS 670)
0A0 - 0AF	Bộ điều khiển ngắt 2 (8259)
0CH - 0CF	Bộ điều khiển DMA 2 (8237)
0E0 - 0EF	Dự trữ cho mảng mạch chính
0F8 - 0FF	Bộ đồng xử lý 80x87
1F0 - 1F8	Bộ điều khiển đĩa cứng
200 - 20F	Cổng dùng cho trò chơi (Game)
278 - 27F	Cổng song song 2 (LPT 2)

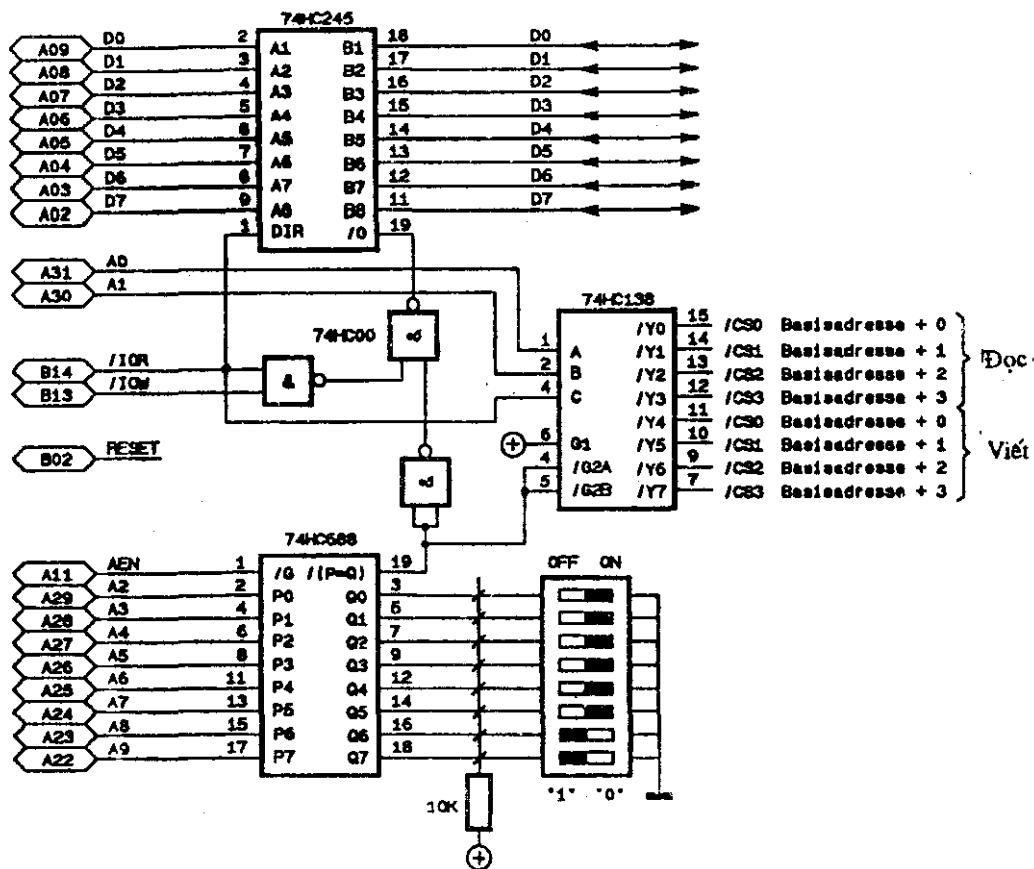
2B0 - 2DF	Card EGA 2
2E8 - 2EF	Cổng nối tiếp 4 (COM 4)
2F8 - 2FF	Cổng nối tiếp 2 (COM 2)
300 - 31F	Dùng cho card mở rộng
320 - 32F	Bộ điều khiển đĩa cứng
360 - 36F	Cổng nối mạng (LAN)
378 - 37F	Cổng song song 1 (LPT 1)
380 - 38F	Cổng nối tiếp đồng bộ 2
3A0 - 3AF	Cổng nối tiếp đồng bộ 1
3B0 - 3B7	Màn hình đơn sắc
3C0 - 3CF	Card EGA
3D0 - 3DF	Card CGA
3E8 - 3EF	Cổng nối tiếp 3 (COM 3)
3F0 - 3F7	Bộ điều khiển đĩa mềm
3F8 - 3FF	Cổng nối tiếp 1 (COM 1)

Từ bảng này ta thấy rõ ràng là các địa chỉ 300 đến 31F (Hex.) đã được dự tính để dùng cho các card mở rộng. Các đường dẫn địa chỉ được sử dụng đối với vùng này là A0 đến A9. Thông thường thì các địa chỉ, mà dưới các địa chỉ này máy tính có thể trao đổi với card mở rộng, có thể đặt được ở chính trên card. Bây giờ nhiệm vụ của tấm bản mạch (card) được gài vào là so sánh các đường dẫn địa chỉ ở máy tính với các địa chỉ đã được thiết lập xem có thống nhất không và thông báo sự đánh giá ở một bộ điều khiển logic. Chỉ khi có sự thống nhất một cách chính xác mới có thể tiến hành sự trao đổi thông tin với máy tính.

Thông thường thì trên một card mở rộng có nhiều khối chức năng như: bộ biến đổi A/D, bộ biến đổi D/A, khối xuất và nhập dữ liệu số, các khối này được trao đổi dưới những địa chỉ khác nhau từ máy tính. Trên hình 1.21 chỉ ra một mạch giải mã điển hình, mà ta sẽ còn gặp lại trên các card mở rộng.

Bộ giải mã địa chỉ 74HC688 so sánh các đường dẫn địa chỉ A2 đến A9 xem có thống nhất với các địa chỉ cơ bản được thiết lập trên card mở rộng bằng chuyển mạch DIP. 74HC688 so sánh hai trong số tám bit

xem có giống nhau không và khi các bit xếp kế sát đồng nhất sẽ tạo ra một tín hiệu Low ở chân 19.



Hình 1.21 Mạch giải mã thường dùng trên một card được cắm thêm vào.

Ngoài cái đó ra nó còn quy định về quyền sử dụng một lối vào được kích hoạt /G. Chừng nào mà chân ra này còn nằm ở mức High (cao) thì tín hiệu lối ra của bộ so sánh vẫn giữ nguyên mức logic 1 độc lập với mức logic xếp kế sát. Bình thường thì chân ra này được nối với tín hiệu AEN. Chỉ khi tín hiệu này dẫn đến mức thấp thì các dữ liệu có giá trị mới nằm ở trên bus. Do được chế tạo bằng công nghệ CMOS mà đối với các tín hiệu của bus PC linh kiện này thực tế tỏ ra là không có tải.

Mạch logic của bộ giải mã sinh ra từ các đường dẫn địa chỉ A0 và A1 tín hiệu đánh giá của bộ so sánh địa chỉ và hai tín hiệu /IOR, /IOW

các tín hiệu lựa chọn đối với các khối chức năng riêng biệt. Vì thế bằng các địa chỉ A0 và A1 cho phép trao đổi với bốn chức năng khác nhau khi đọc vào hoặc khi ghi, card mở rộng cũng chiếm bốn địa chỉ.

Mạch logic của bộ giải mã có chứa hai vi mạch 74HC00 và 74HC138. Ba cổng NAND làm cho bộ đệm bus 74HC245 sau đây chỉ trở nên được kích hoạt ($/G = 0$) khi các điều kiện sau được thực hiện: Thứ nhất là card mở rộng cắm thêm vào đã trao đổi được (chân 19 của vi mạch 74HC688 bằng không) và thứ hai là một chu trình đọc hoặc là chu trình ghi được thực hiện ($I/O = 0$ hoặc là $IOW = 0$). Khi mà chỉ một điều kiện không được thực hiện thì chân ra $/G$ của 74HC245 giữ nguyên là High và bộ đệm bus dừng lại trong trạng thái điện trở cao.

G1	/G2A	/G2B	C	B	A	/Y0	/Y1	/Y2	/Y3	/Y4	/Y5	/Y6	/Y7
			IOR	A1	A0								
X	1	1	X	X	X	1	1	1	1	1	1	1	1
0	X	X	X	X	X	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	1	1	1	1	1	1	1
1	0	0	0	0	1	1	0	1	1	1	1	1	1
1	0	0	0	1	0	1	1	0	1	1	1	1	1
1	0	0	0	1	1	1	1	1	0	1	1	1	1
1	0	0	1	0	0	1	1	1	1	0	1	1	1
1	0	0	1	0	1	1	1	1	1	1	0	1	1
1	0	0	1	1	0	1	1	1	1	1	1	0	1
1	0	0	1	1	1	1	1	1	1	1	1	1	0

Hình 1.22 Bảng chân lý của vi mạch giải mã 74HC138 trên hình 1.21.

Ngoài ra, mạch logic của bộ giải mã còn có chứa một linh kiện đặc biệt nhằm làm giảm đáng kể giá thành của bộ giải mã. Vi mạch 74HC138 có chứa một bộ giải mã 1 trong 8 với ba tín hiệu lựa chọn chip. Nhờ một con số nhị phân, ở linh kiện này ta có thể chọn đúng lối ra từ trong số tám lối ra. Lối ra đã được lựa chọn sau đó sẽ nhận mức Low, trong khi tất cả các lối ra khác vẫn giữ nguyên mức logic 1. Trong mạch điện được minh họa trên hình 1.21, các bit địa chỉ A0 và A1 cũng như

tín hiệu IOR đạt đến các lối vào địa chỉ A, B và C. Còn hình 1.22 mô tả bảng chân lý của vi mạch 74HC138.

Đến đây ta có thể tóm tắt lại là: card mở rộng được cắm thêm vào rãnh cắm sẽ chiếm bốn địa chỉ, trong đó mỗi địa chỉ đều có thể được đọc hoặc là ghi. Vùng địa chỉ của card mở rộng có thể viết như sau:

Địa chỉ vào / ra	Đọc/ghi (R/W)	Lối ra tín hiệu
Địa chỉ cơ bản (chẳng hạn F 300)	Đọc	Y0 = 0
	Ghi	Y4 = 0
Địa chỉ cơ bản + 1 (chẳng hạn F301)	Đọc	Y1 = 0
	Ghi	Y5 = 0
Địa chỉ cơ bản + 2 (chẳng hạn F302)	Đọc	Y2 = 0
	Ghi	Y6 = 0
Địa chỉ cơ bản + 3 (chẳng hạn F303)	Đọc	Y3 = 0
	Ghi	Y7 = 0

Trong QBasic các lệnh cần thiết để ghi và đọc địa chỉ có thể viết dưới dạng:

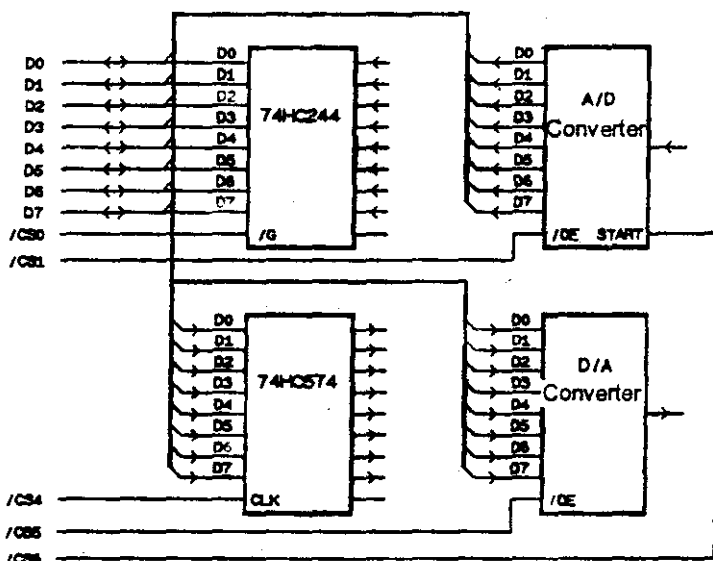
- Chu trình ghi: *OUT port address, outbyte*
- Chu trình đọc: *inbyte = INP (port address)*

Về vấn đề này ta đã có dịp tìm hiểu trong mục 1.1 trước đây.

Bộ giải mã logic đảm nhiệm đồng thời sự điều khiển của bộ đệm bus hai chiều 74HC245, bộ này nối các đường dẫn dữ liệu của rãnh cắm PC với các đường dẫn của card mở rộng. Sự nối ghép này là rất quan trọng, nhờ vậy mà các mức tín hiệu trên đường dẫn dữ liệu không bị ảnh hưởng quá mạnh. Nó có chứa 8 bộ đệm với các lối ra ba trạng thái (Tristate) để trao đổi thông tin giữa các đường dẫn bus dữ liệu theo hai hướng. Hướng được xác định từ mức logic ở lối vào DIR: DIR = 0 chuyển các dữ liệu từ B sang A. Việc chuyển hướng dữ liệu cho phép quản lý một cách đơn giản nhất là bằng tín hiệu /IOR. Ta có thể nối trực tiếp với chân ra DIR. Đương nhiên giả thiết là cổng dữ liệu A được nối với các

đường dữ liệu của rãnh cắm PC và cổng dữ liệu B với đường dẫn dữ liệu của card mở rộng. Qua đó bảo đảm rằng bộ đệm bus 74HC245 chỉ sắp xếp những dữ liệu trên bus dữ liệu của máy tính PC. Khi mà máy tính PC thực hiện một sự truy nhập đọc ($/IOR = 0$).

Các tín hiệu lựa chọn chip CS0 đến CS7 có thể được sử dụng để lựa chọn các khối chức năng khác nhau. Bằng bốn tín hiệu CS0 đến CS3 có thể điều khiển các khối ngoại vi khác nhau để đọc các dữ liệu. Các khối ngoại vi này có thể là bộ biến đổi A/D hoặc là bộ đệm bus, bộ đệm này đẩy ra các dữ liệu theo hướng rãnh cắm PC. Ngược lại các tín hiệu CS4 đến CS7 đóng vai trò kích hoạt khối ngoại vi cần nhập các dữ liệu từ máy tính PC. Các khối này có thể là bộ biến đổi D/A hoặc bộ chốt dữ liệu. Nhờ vậy mà các tín hiệu trigger đối với các phép đo có thể được thực hiện. Hình 1.23 minh họa một vài khả năng cho thấy các tín hiệu này đã được tận dụng như thế nào.



Hình 1.23 Đầu nối tín hiệu CS vào khối ngoại vi.

Khi đọc từ địa chỉ cơ bản thì CS0 nhận mức Low và kích hoạt bộ đệm 74HC244, bộ này đẩy ra 8 bit dữ liệu của nó theo hướng rãnh máy tính PC. Bây giờ máy tính PC có thể đọc các dữ liệu vào bởi vì bộ đệm bus 74HC245 ở thời điểm này cũng đã được kích hoạt. Với CS1 bộ biến đổi A/D trở nên trao đổi được. Điều này xảy ra qua việc đọc địa chỉ

(Basicaddress + 1). Khi ta muốn lưu trữ các dữ liệu lên thiết bị ngoại vi thì cần đến cái gọi là bộ chốt dữ liệu. Vì mạch 74HC574 có chứa 8 mạch D-Flipflop, các mạch này lưu trữ vào linh kiện các dữ liệu xếp kế sát ở lối vào CLK. Khi ta ghép nối CS4 trực tiếp với lối vào CLK của 74HC574 thì khi ghi vào địa chỉ cơ bản các dữ liệu xếp kế sát ở rãnh cảm của PC sẽ được lưu trữ. Tương tự, bộ biến đổi D/A cho phép điều khiển bằng sự chốt dữ liệu.

Trong chu trình ghi dưới địa chỉ (Basicaddress + 2) xung Low của CS6 tác dụng đến sự khởi động của sự biến đổi A/D. Trong trường hợp này thông thường thì hoàn toàn không quan trọng là dữ liệu nào ta gửi đi bởi vì chỉ có xung CS âm được đánh giá.

Chương 2

CÁC MẠCH SỐ**2.1 QUẢN LÝ THEO BIT**

Trong kỹ thuật đo lường và điều khiển, các tín hiệu điều khiển có một ý nghĩa đặc biệt. Còn trong lĩnh vực tự động hoá ta thấy vô số các máy móc được điều khiển bằng các tín hiệu điều khiển số khác nhau. Chẳng hạn, để điều khiển các quá trình trong một trạm rửa xe ô tô, rất nhiều mô tơ và bộ phận chấp hành phải được kích hoạt: mô tơ dùng cho băng chuyền, van dùng cho thiết bị phun nước, động cơ cho máy thổi khô đều mới chỉ là một số được kể ra. Những lệnh để điều khiển việc sắp xếp vị trí được tóm tắt trong một từ điều khiển, trong đó sự sắp xếp vị trí xác định được đặt trong mỗi bit riêng lẻ. Vì thế có thể qua một bit điều khiển một mô tơ của băng chuyền, qua một bit khác điều khiển van của thiết bị phun nước. Sau đó mức logic quyết định liệu bộ phận chấp hành được kích hoạt hay là không. Trong mỗi quan hệ này, điều rất quan trọng là có thể thay đổi một bit xác định mà không làm ảnh hưởng đến trạng thái logic của bit khác. Muốn thế cần có những lệnh nào? Câu trả lời sẽ được tìm thấy trong mục dưới đây.

Đảo (ngược) bit:

Để đảo một bit xác định của một từ điều khiển cần phải thực hiện liên kết LOẠI TRỪ - HOẶC (EXCLUSIVE-OR) từng bit một. Trong liên kết LOẠI TRỪ - HOẶC với 1, bit chịu một tác động đảo, còn trong liên kết LOẠI TRỪ - HOẶC với 0, bit giữ nguyên không thay đổi.

Thí dụ:

	D7	D6	D5	D4	D3	D2	D1	D0	Giá trị
Từ điều khiển cũ bit 3 và bit 5	0	1	0	0	1	0	0	0	76 dec. (hệ 10)
XOR đảo	0	0	1	0	1	0	0	0	40 dec.
Từ điều khiển mới	0	1	1	0	0	1	0	0	100 dec.

Lệnh trong QBasic: *contrword.new = contrword.old XOR (40)*

Đặt bit lên Low

Để đặt một bit xác định của một từ điều khiển lên Low thì một liên kết VÀ cần phải được thực hiện từng bit một. Trong liên kết VÀ với 0, bit được đặt lại lên mức logic 0, còn trong liên kết VÀ với 1 bit giữ nguyên không thay đổi.

Thí dụ:

	D7	D6	D5	D4	D3	D2	D1	D0	Giá trị
Từ điều khiển cũ bit 0 và bit 3	0	1	0	0	1	1	0	1	77 dec.
AND đặt lên LOW	1	1	1	1	0	1	1	0	246 dec.
Từ điều khiển mới	0	1	0	0	0	1	0	0	68 dec.

Lệnh trong QBasic: *controlword.new = controlword.old AND (255-I-8)*

Đặt bit lên High

Để đặt một bit xác định của một từ điều khiển lên High thì một liên kết HOẶC cần phải được thực hiện từng bit một. Trong liên kết HOẶC với 1, bit được đặt lên mức logic 1, còn trong liên kết HOẶC với 0, bit giữ nguyên không thay đổi.

Thí dụ:

	D7	D6	D5	D4	D3	D2	D1	D0	Giá trị
Từ điều khiển cũ lên bit 1	0	1	0	0	1	1	0	1	77 dec.
OR đặt lên High	0	0	0	0	0	0	1	0	2 dec.
Từ điều khiển mới	0	1	0	0	1	1	1	1	79 dec.

Lệnh trong QBasic: *controlword.new = controlword.old OR 2*

Bit hỏi lại

Trong kỹ thuật quá trình thường xuất hiện vấn đề là phải điều khiển một van, mà vị trí cuối cùng của nó *Mở/Đóng* được thông báo trở lại ở các lối vào số của hệ thống tự động. Bài toán của chương trình điều khiển là phải đánh giá mẫu bit kê sát để khẳng định xem tại thời điểm này van đang ở vị trí nào. Từ toàn bộ từ điều khiển phải lọc lựa ra một bit. Điều này xảy ra khi ta liên kết bit cần khảo sát với 1 và các bit còn lại với 0

Thí dụ:

	D7	D6	D5	D4	D3	D2	D1	D0	Giá trị
Từ điều khiển cũ	0	1	0	0	1	0	0	0	77 dec.
Hỏi lại									
AND của bit 3	0	0	1	0	1	0	0	0	8 dec.
Từ điều khiển mới	0	1	1	0	0	1	0	0	8 dec.

Từ kết quả này, bit đang gặp còn phải được lựa chọn. Muốn thế ta đem chia kết quả liên kết với giá trị của bit được lựa chọn.

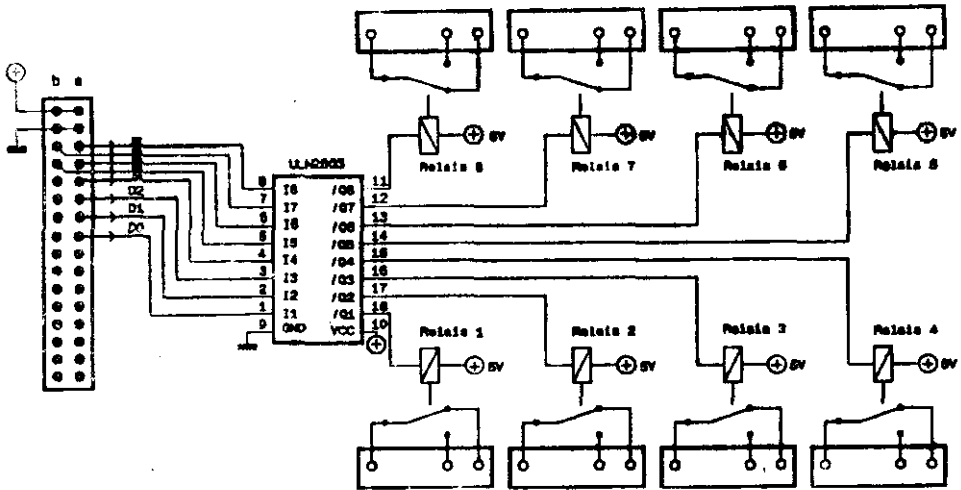
Lệnh trong QBasic là: *level. of. bit 3 = (Controlword AND 2³) / 2³*

2.2 GHÉP NỐI KIỂU ROLE

Trong các mục 1.1 và 1.2 các môđun cơ sở khác nhau đã được giới thiệu, các môđun này có thể được đấu vào hoặc là qua giao diện máy in song song hoặc là giao diện RS 232 nối tiếp.

Dưới đây sẽ giới thiệu một môđun ghép nối kiểu role, môđun này có thể đấu vào một môđun cơ bản nào đó vừa được kể ra ở trên. Nhờ thế mà tám role có thể được điều khiển từ máy tính PC. Mạch điện trên hình 2.1 có chứa một bộ đệm khá đặc biệt có tên là ULN 2803 mà lối ra của nó có thể chịu tải đến 500 mA. Bộ đệm ULN 2803 có chứa 8 tầng đệm với các diốt bảo vệ đã được tích hợp, các tầng này có khả năng điều khiển trực tiếp các role. Điều đáng lưu ý ở đây là mặc dù vi mạch làm việc như một bộ đảo nhưng mà tải lại được đấu với nguồn nuôi dương

vào lối ra của bộ đếm, và vì thế khi lối vào ở mức High cũng có dòng đi qua role.



Hình 2.1 Môđun ghép nối kiểu role.

Chương trình RELAIS.BAS để điều khiển các role riêng biệt có thể được viết ra như dưới đây:

```

' =====
' Program: RELAIS
'
' Function: Bằng chương trình này có thể điều khiển
' 8 role tùy theo cách lựa chọn hoặc là qua cổng nối
' với máy in hoặc là qua cổng nối tiếp (cổng chuột)
' của máy tính PC
'
' Hardware: Cần có một môđun cơ sở đã trình bày ở
' chương 1 cũng như môđun ghép nối role
' =====
COLOR 0, 15
CLS
Nạpvao:
PRINT "Bạn sử dụng cổng ghép nối nào ?"
PRINT
PRINT "Giao diện (Interface) (2)"
INPUT lptcom

```

'-----Khởi tạo cổng đã lựa chọn-----'

```
SELECT CASE lptcom
CASE 1
basadr = &H378 'Basicaddress
datreg = basadr 'Datenregister
statreg = basadr + 1 'Statusregister
contrreg = basadr + 2 'Controlregister
CASE 2
OPEN "COM2: 9600,N,8,1,CS,DS" FOR RANDOM AS #1
CASE ELSE: GOTO Napvao 'Napvao: Input
END SELECT
```

'-----Bắt đầu chương trình riêng-----'

```
CLS
a$ = "R E L A I S - I N T E F A C E"
a = LEN (a$): b = (80 - a) / 2 - 1
LOCATE 1, b: PRINT CHR$ (201); STRING$
(a + 2, CHR$ (205)); CHR$ (187)
LOCATE 2, b: PRINT CHR$ (186)
LOCATE 2, b + 2: PRINT a$
LOCATE 2, a + b + 4: PRINT CHR$ (186)
LOCATE 3, b: PRINT CHR$ (200); STRING$
(a + 2, CHR$ (205)); CHR$ (188)
LOCATE 5, 25: PRINT "Relais 1 ="
LOCATE 7, 25: PRINT "Relais 2 ="
LOCATE 9, 25: PRINT "Relais 3 ="
LOCATE 11, 25: PRINT "Relais 4 ="
LOCATE 13, 25: PRINT "Relais 5 ="
LOCATE 15, 25: PRINT "Relais 6 ="
LOCATE 17, 25: PRINT "Relais 7 ="
LOCATE 19, 25: PRINT "Relais 8 ="
LOCATE 22, 1: PRINT "Hướng dẫn:"
LOCATE 22, 15: PRINT CHR$ (24); CHR$ (25);
":Lựa chọn Role"
LOCATE 22, 35: PRINT "ESC: Kết thúc"
LOCATE 22, 50: PRINT "F1: Đảo role"
```



```

LOCATE 23, 1: PRINT "-----"
'----- Đặt tất cả các Role lên 0 -----
FOR i = 5 TO 19 STEP 2
LOCATE i, 45: PRINT "MỞ"
NEXT
SELECT CASE lptcom
CASE 1: OUT datreg, 0
CASE 2: PRINT #1, CHR$ (0);
END SELECT
ZEILE = 5: spalte = 25
DO
'----Đọc dòng hiện thời như là text (Cột 25 đến 33)----
Text$ = ""
FOR i = 0 TO 8
text$ = text$ + CHR$ (SCREEN (ZEILE, spalte + i))
NEXT
-----text$ mô tả đảo-----
COLOR 15, 0: LOCATE ZEILE, spalte
PRINT text$
COLOR 0, 15
-----Bắt đầu vòng lặp, hỏi lại bàn phím-----
DO
taste$ = INKEYS      Chờ nhấn phím
LOOP UNTIL taste$ <> ""
-----Hết vòng lặp (phím đã được ấn)-----
LOCATE ZEILE, spalte
PRINT text$
'-----Phím nào được nhấn ?-----
SELECT CASE ASC (RIGHTS (taste$, 1))
CASE 27: EXIT DO      'Phím ESC
CASE 59                'Phím F1
LOCATE ZEILE, 45
IF CHR$ (SCREEN (ZEILE, 45)) = "E" THEN
PRINT "MỞ"
ELSE PRINT "ĐÓNG"

```

```

END IF
CASE 72: IF ZEILE > 5 THEN ZEILE =
ZEILE - 2           'Phím mũi tên lên
CASE 80: IF ZEILE < 19 THEN ZEILE =
ZEILE + 2           'Phím mũi tên xuống
END SELECT

```

```

----- Xuất dữ liệu -----
outbyte = 0
FOR i = 5 TO 19 STEP 2
outbyte = outbyte -
((CHR$ (SCREEN (i, 45)) = "E")) * 2 ^ ((i - 5) / 2
NEXT
SELECT CASE lptcom
CASE 1: OUT datreg, outbyte      'Cổng máy in
CASE 2: PRINT #1, CHR$ (outbyte); 'Cổng nối tiếp
END SELECT
LOOP
END

```

GHEP NOI BANG ROLE

	=	Mo
Role 2	=	Mo
Role 3	=	Mo
Role 4	=	Mo
Role 5	=	Mo
Role 6	=	Mo
Role 7	=	Mo
Role 8	=	Mo

Huong_dan: ↑↓ : Lua_chon_role ESC : Ket_thuc F1: Dao_role

Hình 2.2 Trình bày trên màn hình của chương trình RELAIS.BAS.

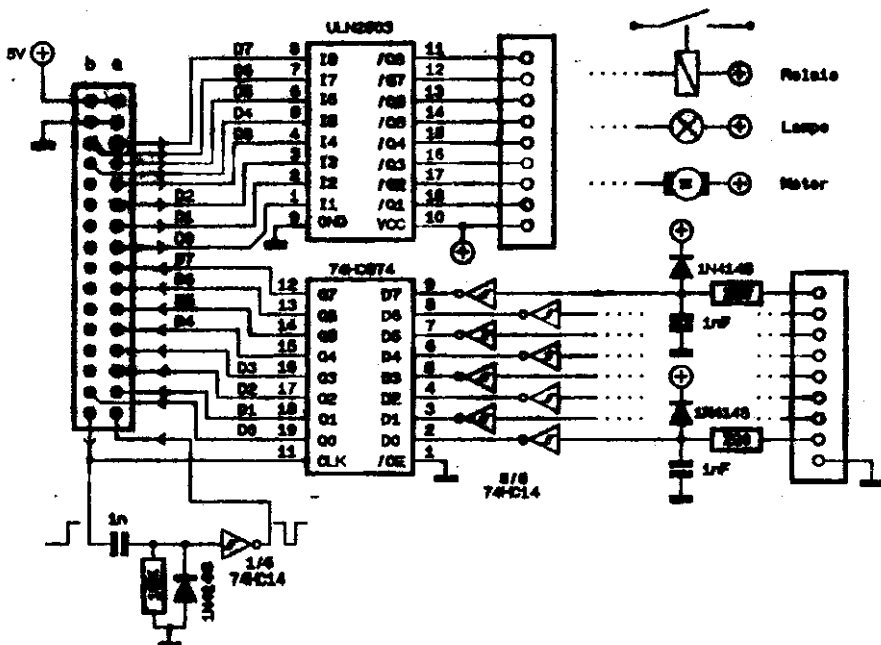
Sau khi khởi động chương trình, người sử dụng phải quyết định xem nên điều khiển bộ ghép nối role qua giao diện nào. Tiếp đó, trên màn hình sẽ xuất hiện cách trình bày như trên hình 2.2.

Thoạt đầu tất cả các role đều được ngắt điện. Bằng phím mũi tên ta có thể lựa chọn một role nào đó và đảo trạng thái bằng phím F1. Sau khi xác nhận bằng phím F1, role sẽ được điều khiển lập tức. Bằng phím ESC ta có thể kết thúc chương trình.

2.3. BỘ ĐIỀU KHIỂN MINI THEO CHƯƠNG TRÌNH ĐÃ NHỚ (SPC)

Mục này mô tả một môđun ghép nối mà ta có thể sử dụng cùng với một môđun cơ sở từ chương 1 như là giao diện điều khiển. Với một phần mềm thích hợp, sau đó có thể thực hiện một quá trình điều khiển theo chương trình đã lưu trữ - thường viết tắt là SPC (từ các chữ tiếng Anh: *Stored-Programmable Control*). Môđun ghép nối trên hình 2.3 thu thập tám tín hiệu nhị phân và có thể qua bộ đệm ULN 2803 điều khiển trực tiếp các bộ phận chấp hành có công suất nhỏ. Dòng tải lớn nhất đạt đến 500 mA, đủ để đệm cho các bóng đèn, role và các mô tơ nhỏ.

Hardware



Hình 2.3 Môđun mini SPC.

Phần cứng

Các tín hiệu lối vào nhị phân được dẫn đến các lối vào của vi mạch 74HC14 qua một mạch RC, vi mạch này có chứa 4 bộ đảo với chức năng của trigger Schmitt. Các diốt được dùng để bảo vệ giao diện khi các điện áp lối vào lớn hơn 5 V. Sau một lần đảo, tín hiệu lối vào được dẫn đến bộ chốt dữ liệu 74HC574. Bằng sườn dương (thay đổi từ Low sang High) của xung giữ nhịp ở lối vào CLK (chân 11) bộ này lưu trữ các dữ liệu kể sát bên trong vào thanh ghi và sau đó đặt vào các lối ra Q để chờ xử lý tiếp. Sau đấy, sự thay đổi của tín hiệu lối vào giữ nguyên không được để ý đến. Tín hiệu CLK được lấy đi ở chân 1b của ổ nối 32 chân. Tùy theo môđun cơ sở được sử dụng mà tín hiệu này xuất phát từ các nguồn khác nhau.

Ở môđun cơ sở dùng cho cổng máy in được mô tả trong mục 1.1.4 tín hiệu STROBE được dẫn qua chân ra 1b tới lối vào CLK của vi mạch 74HC574. Trước khi một byte dữ liệu hiện thời được đọc vào, một sự thay đổi tín hiệu từ Low sang High cần phải được tạo ra bằng tín hiệu STROBE.

Khi sử dụng môđun cơ sở dùng cho giao diện nối tiếp RS 232 (xem mục 1.2) tín hiệu DR (Data Received) của bộ UART (xem thêm hình 1.4) được dẫn qua chân ra 1b đến chân tín hiệu CLK của vi mạch 74HC574. Bởi vì DR được tự động gửi đi từ bộ UART CDP 6402, nên ta không cần quan tâm hơn đến sự lập trình của tín hiệu này. DR chuyển từ Low sang High khi bộ UART đã nhận xong dữ liệu từ máy tính PC.

Để có thể đọc vào các dữ liệu bằng một môđun cơ sở dùng cho cổng nối tiếp thì ngoài ra còn cần có một xung Low để mở đầu việc truyền dữ liệu tới máy tính. Đồng thời xung này được điều khiển bởi tín hiệu DR. Tín hiệu DR được dẫn qua một tụ điện tới một bộ đảo của vi mạch 74HC14, mà ở lối vào của bộ này có đầu qua một điện trở xuống mass. Sau đó lối ra của bộ đảo chuyển sang Low trong khoảng thời gian kéo dài $\tau = -0,7 R.C$ mỗi khi xuất hiện sườn dương của tín hiệu DR. Nói khác đi: Với mỗi byte mà máy tính PC gửi tới giao diện, máy tính PC nhận được một byte trả lời lại, byte này cho trở lại tín hiệu lối vào nhị phân của các đường dẫn dữ liệu D0IN... D7IN

Phần mềm

Phần mềm để điều khiển môđun đã được cất ra theo những ứng dụng của một bộ SPC. Bên cạnh các lối vào hiện thời, các mức tín hiệu của lối ra cũng được chỉ ra trên màn hình. Hình 2.4 chỉ ra sự trình bày của màn hình sau khi khởi động MINISPC.BAS, chương trình này chứa các mã Basic dưới đây:

```
' =====
' Program: MINISPC
' Function: Bằng chương trình này có thể thực hiện
' được chức năng điều khiển qua cổng ghép nối máy in
' hoặc cổng nối tiếp tương tự như một SPC
' Hardware: Cần có một môđun cơ sở đã trình bày ở
' chương 1 cũng như một môđun SPC để điều khiển các
' lối ra số và đọc các lối vào digital
' =====
DECLARE SUB lese. lpt (inbyte)
DECLARE SUB lese. com (inbyte)
DIM E1 (0 TO 7)      '8 input
DIM A1 (0 TO 7)      '8 Output
DIM M1 (0 TO 7)      '8 Merker
DIM SHARED basadr, datreg, statreg, contrreg
COLOR 0, 15
CLS
Nạpvao:
PRINT "Bạn chọn giao diện nào ? "
PRINT
PRINT "Cổng máy in (1) hoặc là cổng nối tiếp (2)"
INPUT lptcom
-----Khởi tạo giao diện đã lựa chọn-----
SELECT CASE lptcom
CASE 1
basadr = &H378  Địa chỉ cơ bản của cổng máy in
datreg = basadr      'Datenregister
statreg = basadr + 1 'Statusregister
```

```

contrreg = basadr + 2    'Controlregister
CASE 2
OPEN "COM 2: 9600,N,8,1,CS,DS" FOR RANDOM AS #1
CASE ELSE: GOTO Napvao
END SELECT

```

-----Bắt đầu chương trình riêng-----

```

CLS
a$ = "M I N I - SPS"
a = LEN (a$): b = (80 - a) / 2 - 1
LOCATE 1, b: PRINT CHR$ (201); STRING$
(a + 2, CHR$ (205); CHR$ (187))
LOCATE 2, b: PRINT CHR$ (186)
LOCATE 2, b + 2: PRINT a$
LOCATE 2, a + b + 3: PRINT CHR$ (186)
LOCATE 3, b: PRINT CHR$ (200); STRING$
(a + 2, CHR$ (205)); CHR$ (188)
PRINT
LOCATE 7, 10: PRINT "Loi_vao_so (Digital)"
LOCATE 7, 50: PRINT "Loi_ra_so (Digital)"
LOCATE 8, 10: PRINT "-----"
LOCATE 8, 50: PRINT "-----"
LOCATE 9, 10: PRINT "E 1.0 ="
LOCATE 9, 50: PRINT "A 1.0 ="
LOCATE 10, 10: PRINT "E 1.1 ="
LOCATE 10, 50: PRINT "A 1.1 ="
LOCATE 11, 10: PRINT "E 1.2 ="
LOCATE 11, 50: PRINT "A 1.2 ="
LOCATE 12, 10: PRINT "E 1.3 ="
LOCATE 12, 50: PRINT "A 1.3 ="
LOCATE 13, 10: PRINT "E 1.4 ="
LOCATE 13, 50: PRINT "A 1.4 ="
LOCATE 14, 10: PRINT "E 1.5 ="
LOCATE 14, 50: PRINT "A 1.5 ="
LOCATE 15, 10: PRINT "E 1.6 ="
LOCATE 15, 50: PRINT "A 1.6 ="

```

```

LOCATE 16, 10: PRINT "E 1.7 ="
LOCATE 16, 50: PRINT "A 1.7 ="
LOCATE 22, 1: PRINT "Hướng dẫn:"
LOCATE 23, 1: PRINT "-----"
LOCATE 22, 15: PRINT "ESC: Kết_thúc"

'-----Đặt tất cả Merker, Input và Output lên Low-----
FOR i = 0 TO 7: M1 (i) = 0: NEXT
FOR i = 0 TO 7: A1 (i) = 0: NEXT
FOR i = 0 TO 7: E1 (i) = 0: NEXT

'-----Bắt đầu vòng lặp-----
DO
'----- Xuất các bit lên màn hình -----
FOR i = 0 TO 7
LOCATE 9 + i, 20: PRINT E1(i)
LOCATE 9 + i, 60: PRINT A1(i)
NEXT
'-----Tiếp theo là các phương trình lôgic-----
' Bài toán điều khiển (Control-problem)
' Chẳng hạn: M1 (3) = (E1(3) AND M1(2) OR M1(3)) AND
' (NOT M1 (0))
' or: A1(0) = E1 (0) AND E1(6)
'----- Thiết lập các byte lỗi ra -----
outbyte = 0
FOR i = 0 TO 7
outbyte = outbyte + A1 (i) * 2 ^ i
NEXT
'----- Xuất outbyte -----
SELECT CASE lptcom
CASE 1: OUT datreg, outbyte 'Cổng máy in
CASE 2: PRINT #1, CHR$(outbyte); 'Cổng nối tiếp
'----- Đọc lỗi vào -----
SELECT CASE lptcom
'----- Qua giao diện máy in -----
CASE 1
OUT contrreg, (INP (contrreg) AND (255 - 1))

```

```

      'STROBE lên mức 0
OUT contrreg, (INP (contrreg) OR 1)
      'STROBE lên 1

CALL lese. lpt (inbyte)
'----- Về giao diện nối tiếp -----
CASE 2: CALL lese.com (inbyte) .
b$ = INPUT$ (LOC (1), #1) 'Xoá toàn bộ bộ đệm
END SELECT

inbyte = inbyte XOR 255 'Phép đảo nhờ 74HC14
FOR i = 0 TO 7
E1 (i) = (inbyte AND 2 ^ i) / 2 ^ i
NEXT

'-----Phím ESC bằng kết thúc-----
taste$ = INKEY$
IF taste$ = CHR$ (27) THEN EXIT DO

LOOP
END
SUB lese.com (inbyte)
'=====
' Unterprogram: lese, com
' Function:Chương trình này đọc vào một byte qua
' cổng nối tiếp. Khi cuộc truyền dữ liệu bị nhiễu
' sẽ có một thông báo nhấp nháy và chương trình kết
' thúc
'=====
i = 0
DO
i = i + 1
'-----Khi mà các byte tồn tại-----
IF LOC (1) >= 1 THEN
in$ = INPUT$ (1, #1)
inbyte = ASC (in$)

```



```

GOTO ketthuc
END IF

'-----Đọc các dữ liệu của phép thử mới-----
FOR k = 1 TO 1000: NEXT
LOOP UNTIL i = 10          'Cực đại 1000 phép thử
'-----Không nhận được ký tự nào -----
CLS
PRINT "Sự truyền dữ liệu bị nhiễu !!!"
PRINT
PRINT "Không nhận được ký tự nào !!!"
PRINT
PRINT "Kiểm tra lại: cáp nối, phần cứng..."
END
ketthuc:
END SUB

SUB lese.lpt (inbyte)
'-----Đọc dữ liệu vào-----
OUT contrreg, 0          'init = 0
inbyte1 = INP (statreg) 'Đọc vào từ D0,D1,D2,và D3
OUT contrreg, 4          'init = 1
inbyte2 = INP (statreg) 'Đọc vào từ D4,D5,D6,và D7
'-----Sắp xếp các bit dữ liệu đã đọc vào-----
inbyte1:  Statusregister bit 4 (SLCT) là D0
          Statusregister bit 5 (PE)      là D1
          Statusregister bit 6 (ACK)     là D2
          Statusregister bit 5 (BUSY)    là /D3
inbyte2:  Statusregister bit 4 (SLCTz0)  là D4
          Statusregister bit 7 (PE)      là D5
          Statusregister bit 6 (ACK)     là D6
          Statusregister bit 5 (BUSY)    là /D7
inbyte = (((inbyte1 XOR 128) AND &HFO) / 16) +
          ((inbyte2 XOR 128 AND &HFO)
END SUB

```

Bên cạnh việc làm cho người sử dụng nhìn thấy lỗi vào và lỗi ra, bài toán điều khiển cũng được lập trình. Giống như trong một SPC, những liên kết nhị phân tương ứng cũng có trong Qbasic.

MINI - SPS	
Loi_vao_so (digital)	Loi_ra_so (digital)
E 1.0 = 1	A 1.0 = 1
E 1.1 = 1	A 1.1 = 1
E 1.2 = 1	A 1.2 = 1
E 1.3 = 1	A 1.3 = 1
E 1.4 = 1	A 1.4 = 1
E 1.5 = 1	A 1.5 = 1
E 1.6 = 1	A 1.6 = 1
E 1.7 = 1	A 1.7 = 1
Huong_dan:	ESC : Ket_thuc

Hình 2.4 Sự trình bày trên màn hình của chương trình MINISPS.BAS.

Chương trình điều khiển của một bộ SPC bao gồm từ một dãy các lệnh điều khiển được xử lý nối tiếp nhau. Sau khi xử lý xong tất cả các lệnh, các lỗi ra sẽ được điều khiển. Ở đây, chúng tôi không đề cập đến những vấn đề cơ sở của kỹ thuật điều khiển theo chương trình đã lưu trữ, mà chỉ dẫn ra ba thí dụ nhằm minh hoạ: danh sách các lệnh (viết tắt là: DSCL) của một bộ SPC có thể được chuyển đổi trong cú pháp QBasic như thế nào.

Trong thí dụ 1, lỗi ra A 1.0 sau đó chỉ trở nên High khi tất cả các lỗi vào ở mức High. Liên kết VÀ (AND) có thể được chuyển giao trên thực tế theo tương quan một với một trong QBasic. Sự khác nhau duy nhất thể hiện ở sự sắp xếp các kết quả liên kết lỗi ra. Trong khi ở DSCL của một bộ SPC, trước hết, các phương trình logic được thể hiện rồi sau đó các kết quả được gán cho lỗi ra; còn trong QBasic thứ tự này được thực

hiện hoàn toàn ngược lại. Biến nhận một giá trị và biến này cần phải đúng để bắt đầu một lệnh.

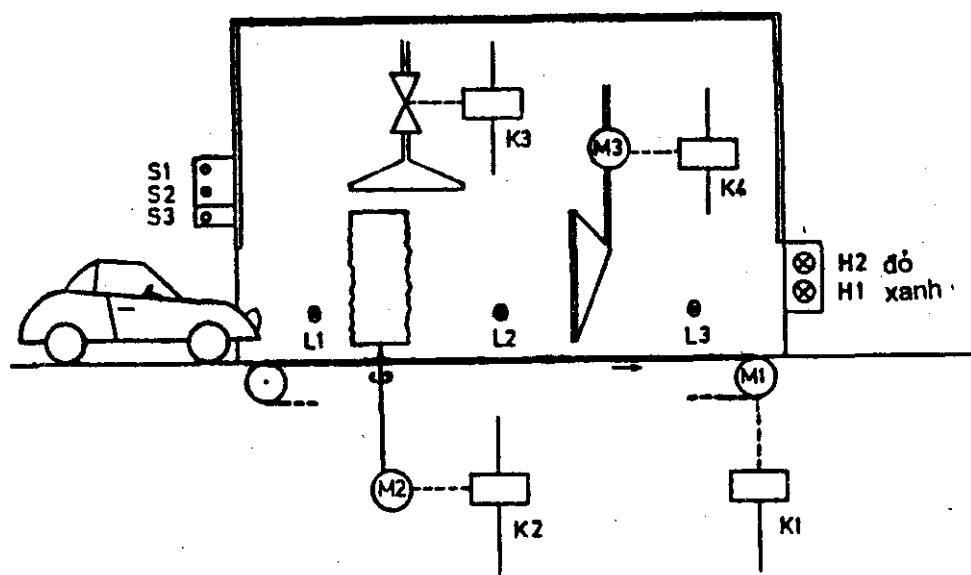
	DSCL của một SPC			QBasic
Thí dụ 1:	U	E	1.0	$A1(0) = E1(0) \text{ AND } E1(1) \text{ AND } E1(2)$
	U	E	1.1	
	U	E	1.2	
	=	A	1.0	
Thí dụ 2:	U	E	1.0	$A1(0) = (E1(0) \text{ AND } E1(1)) \text{ OR } E1(2)$
	U	E	1.1	
	O	E	1.2	
	=	A	1.0	
Thí dụ 3:	U	E	1.0	$A1(0) = (E1(0) \text{ OR } A1(0)) \text{ AND NOT } E1(1)$
	S	A	1.0	
	U	E	1.1	
	R	A	1.0	

Trong thí dụ 2, nên chú ý tới một dãy tuần tự của các lệnh liên kết. Trong bộ SPC, chương trình điều khiển được xử lý theo lệnh đối với lệnh. Nghĩa là lỗi vào E 10 và E 11 chịu một liên kết VÀ, và rồi kết quả lại được liên kết HOẶC với E 1.2. Dãy tuần tự có thể đặt được trong QBasic chỉ qua các dấu ngoặc phụ thêm vào.

Thí dụ 3 minh họa cho việc sử dụng lệnh S để đặt và R để đặt lại một bit. Với sự tác động của lệnh S, lỗi ra được trao đổi A1.0 được đặt lên High, nếu như kết quả liên kết của lệnh điều khiển vừa tiến hành là logic 1. Sau đó, lỗi ra cũng duy trì trạng thái logic, khi kết quả liên kết thay đổi sang 0. Lỗi ra A1.0 chỉ có thể được đặt lại bằng E1.1 = 1. Bởi vì hai lệnh S và R không có chứa trong QBasic nên các hàm này cần phải được giải thích trên số hạng liên kết cơ bản. Từ thí dụ 3 ta thấy được sự chuyển đổi này đã được thực hiện như thế nào.

Ứng dụng: Điều khiển một trạm rửa ô tô.

Ứng dụng dưới đây cho ta một ý niệm nhỏ trong kỹ thuật điều khiển và chỉ cho ta thấy một trạm rửa ô tô có thể được điều khiển từ máy tính như thế nào khi sử dụng môđun SPC ghép nối với môđun cơ sở đã trình bày ở chương 1. Sự mô tả hệ thống của trạm rửa ô tô được chỉ ra trên hình 2.5.



Hình 2.5 Sơ đồ mô tả một trạm rửa ô tô.

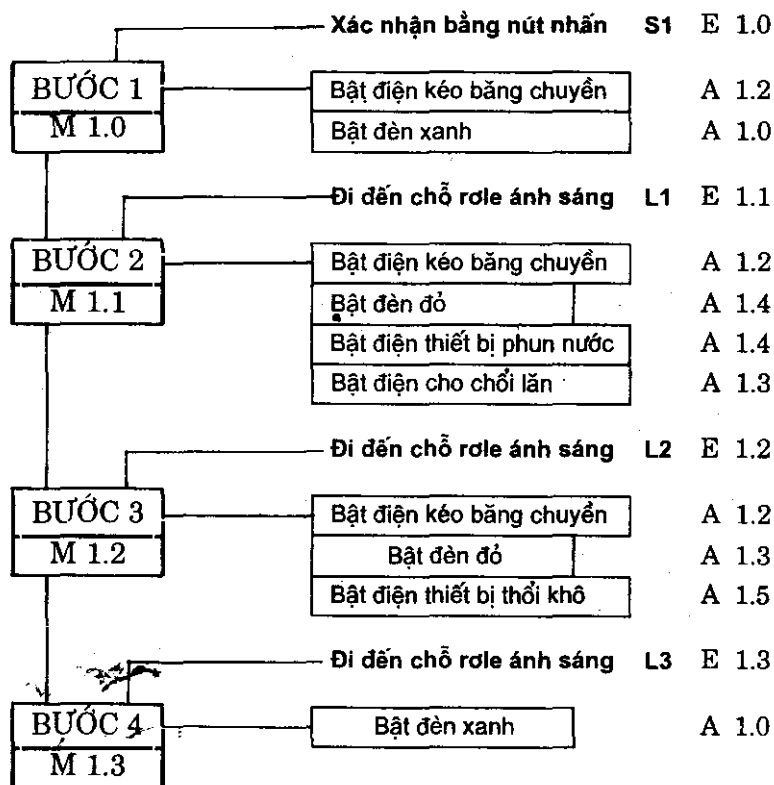
Diễn biến theo thời gian của sự điều khiển quá trình có thể được mô tả như sau:

1- Trạm này được đóng điện bằng công tắc *đóng/ngắt* S2. Sau đó quá trình rửa cần phải được bắt đầu bằng sự xác nhận của nút nhấn khởi động S1. Ngoài ra, băng chuyển được đóng mạch qua role K1.

2- Xe ô tô được di chuyển bằng băng chuyển qua những vị trí khác nhau của trạm. Khi xe chuyển dịch đến role ánh sáng L1 (L1 = 1) thì chổi lăn để làm sạch và thiết bị phun nước cần phải được đóng mạch. Đồng thời đèn xanh tắt và đèn đỏ cần phải được bật lên. Đèn đỏ báo hiệu là có xe đang rửa và xe sau cần phải đứng chờ ở trước lối vào trạm rửa.

3- Khi đến role ánh sáng L2 ($L2 = 1$) thì một mặt van của thiết bị phun nước cần phải được đóng lại, mặt khác chổi lăn để làm sạch phải được ngắt điện. Đồng thời thiết bị thổi khô bắt đầu hoạt động.

4- Khi ô tô di chuyển tiếp đến role ánh sáng L3 ($L3 = 1$) thì quá trình làm khô kết thúc và động cơ thổi phải được ngắt mạch. Ngoài ra đèn đỏ cần phải tắt và đèn xanh được bật lên báo hiệu cho xe tiếp theo có thể đi vào trạm rửa.



Hình 2.6 Sơ đồ chức năng của một trạm rửa ô tô.

Từ sự mô tả quá trình thời gian rõ ràng là sự điều khiển quá trình có thể được chia nhỏ thành bốn bước. Sự chuyển tiếp của một bước sang bước tiếp theo phụ thuộc vào điều kiện chuyển tiếp. Hơn nữa đây là dấu hiệu điển hình của một việc điều khiển quá trình không giống như ở sự điều khiển liên kết (Linkage) các tín hiệu lối vào dẫn trực tiếp đến một sự thay đổi của trạng thái lối ra. Thí dụ chỉ một sự kích hoạt của role ánh sáng L2 không đủ để đóng van chấp hành của thiết bị phun nước.

Còn một điều kiện khác nữa là bước 2 cần phải được kích hoạt, sau đó việc dịch chuyển đến vị trí role ánh sáng L2 dẫn đến hành động mong muốn.

Một khả năng khác để mô tả quá trình thời gian của trạm rửa ô tô là dùng sơ đồ chức năng (so sánh với hình 2.6). Từ sơ đồ này rút ra không chỉ quá trình thời gian mà còn cho phép rút ra tất cả những điều kiện chuyển tiếp từ một bước sang bước tiếp theo.

Bốn bước được kích hoạt kế tiếp nhau trong dãy các quá trình thời gian của chúng. Trước hết là bước 1, sau đấy là bước 2 rồi tiếp đến bước 3 và cuối cùng là bước 4. Mỗi bước có sắp đặt một dấu hiệu bước (Merker). Trong chuỗi các bước này luôn luôn chỉ có một bước, tương ứng với dấu hiệu bước, được kích hoạt. Ta có được điều kiện này bằng cách khi kích hoạt một bước thì bước trước đấy được xóa đi bằng sự lập trình đã định từ trước. Theo cách nhìn này có thể dẫn đến một điều không logic là băng chuyền đã được kích hoạt không chỉ ở một bước. Do đó các lỗi ra mà trong đó băng chuyền cần được điều khiển cần phải được liên kết HOẶC.

Việc lập trình điều khiển quá trình thực hiện theo cách là trước hết chuỗi các bước được lập trình bằng các dấu hiệu bước, và sau đó các lỗi ra riêng biệt được điều khiển bằng các dấu hiệu bước. Để điều khiển quá trình trong một trạm rửa ô tô, trong QBasic có thể dùng các lệnh dưới đây:

```

'-----Tiếp theo là các phương trình logic-----
' Trước hết chuỗi các bước được lập trình bằng các
' dấu hiệu (Merker)
M1 (0) = ((E1(0) AND NOT M1 (7) OR M1 (0))
          AND NOT M1 (1)) 'Bước (Step) 1
M1 (1) = (E1(0) AND M1 (0) OR M1 (1))
          AND (NOT M1 (2)) 'Bước 2
M1 (2) = (E1(2) AND M1 (1) OR M1 (2))
          AND (NOT M1 (3)) 'Bước 3
M1 (3) = (E1(3) AND M1 (2) OR M1 (3))
          AND (NOT M1 (0)) 'Bước 4

```

$M1(0) = (M1(0) \text{ OR } M1(7))$
 $\text{AND NOT } M1(3)$ 'Startmerker'

Dấu hiệu Start ngăn ngừa một quá trình khởi động mới trong khi chẳng hạn bước 3 (M 1.2) đang hoạt tính. Dưới đây là sự điều khiển của các lối ra

$A1(0) = M1(0) \text{ OR } M1(3)$ 'Lối ra A 1.0
 $A1(1) = M1(1) \text{ OR } M1(2)$ 'Lối ra A 1.1
 $A1(2) = M1(0) \text{ OR } M1(1) \text{ OR } M1(2)$ 'Lối ra A 1.2
 $A1(3) = M1(1)$ 'Lối ra A 1.3
 $A1(4) = M1(1)$ 'Lối ra A 1.4
 $A1(5) = M1(2)$ 'Lối ra A 1.5

2.4 VI MẠCH GHÉP NỐI 8243

2.4.1 MÔ TẢ

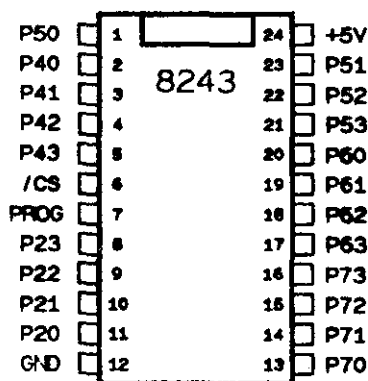
Các vi mạch được tích hợp ở mức độ cao đã làm đơn giản quá trình lắp ráp mạch đi đáng kể. Những thành tựu của sự phát triển trong lĩnh vực điện tử học luôn luôn mang lại những vi mạch có công dụng tốt mà giá cả lại phải chăng. Vi mạch được mô tả dưới đây thuộc loại 8243 tạo ra khả năng làm tăng nhiều lần các đường dẫn vào/ ra số. Điều đáng đề cập ở đây là bus điều khiển và bus dữ liệu cùng tồn tại ở một chân ra và độ rộng bus chỉ là 4 bit. Thêm vào đó còn có một đường dẫn điều khiển đặc biệt, cho nên vi mạch 8243 có tổng cộng năm lối vào điều khiển, qua các lối vào này cho phép điều khiển 16 đường dẫn vào/ ra số. Các lối vào và lối ra lại chia ra 5 cổng và 4 bit. Chỉ với hai vi mạch loại này có thể trao đổi với 32 lối vào và lối ra số ở mức tín hiệu TTL. Sự sắp xếp chân của vi mạch 8243 trên vỏ 24 chân được chỉ ra trên hình 2.7.

Chức năng của các chân riêng biệt có thể được mô tả như sau:

P20 - P23 (gọi chung là cổng 2: Port 2)

Bốn bit P20 - P23 mô tả bus điều khiển và bus dữ liệu theo hai hướng. Tất cả các dữ liệu xuất hiện khi truy nhập lên bốn cổng vào/ ra P4 đến P7 đều được dẫn trên các đường dẫn này. Trong một chương trình đọc,

cổng P2 chứa đựng những dữ liệu đứng xếp hàng của cổng vào/ ra đã được lựa chọn, trong một chu trình ghi các dữ liệu cần phải xuất hiện ở một cổng vào/ ra sẽ được xếp vào. Liệu dữ liệu này liên quan đến chu trình đọc hoặc là chu trình ghi, công việc này sẽ được xác định với sự trợ giúp của từ điều khiển, đồng thời từ này được dẫn đến Port 2. Từ điều khiển 4 bit chứa đựng trước hết là một địa chỉ của cổng vào/ ra được trao đổi và sau đây là mã lệnh để đọc hoặc là ghi (so sánh với hình 2.7)



Từ điều khiển (Controlword) ở Port 2

P23	P22	Lệnh	P21	P20	Cổng (Port)
0	0	Đọc	0	0	Port 4
0	1	Ghi	0	1	Port 5
1	0	ORLD	1	0	Port 6
1	1	ANLD	1	1	Port 7

Hình 2.7 Bố trí chân của vi mạch 8243.

P40 - P 43 (gọi chung là Cổng 4: Port 4)

Bốn bit P40 - P43 tạo nên Port 4. Cổng này có thể được lập trình như là một lối vào hoặc là một lối ra.

P50 - P53 (Cổng 5: Port 5)

Bốn bit P50 - P53 tạo nên Port 5. Cổng này có thể được lập trình như là một lối vào hoặc là một lối ra.

P60 - P63 (Cổng 6)

Bốn bit P60 - P63 tạo nên Port 6. Cổng này có thể được lập trình như là một lối vào hoặc là một lối ra.

P70 - P73 (Cổng 7)

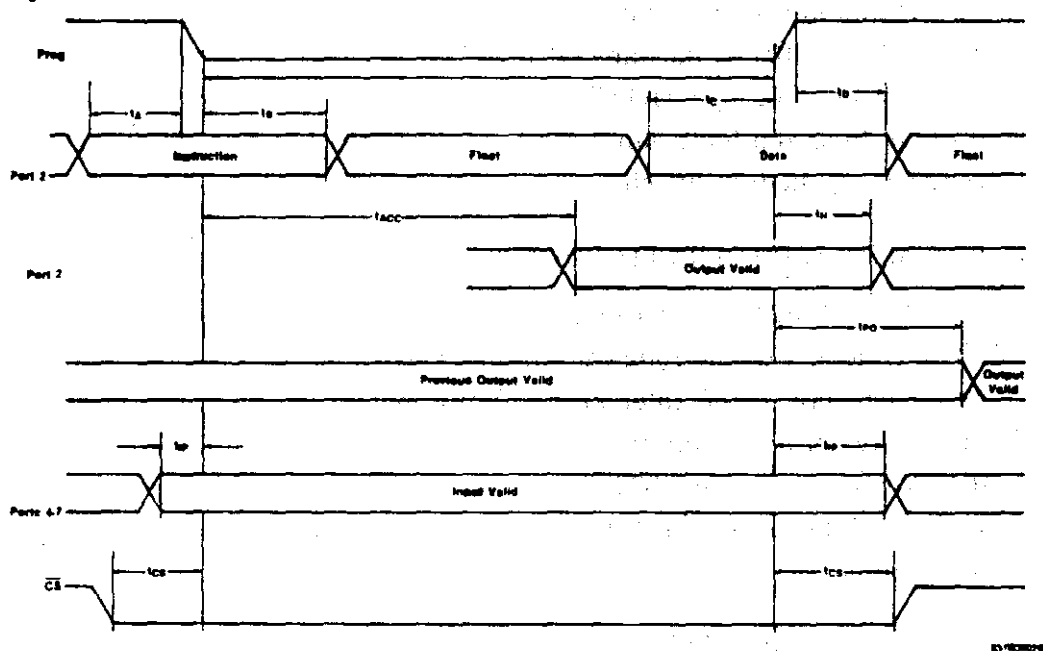
Bốn bit P70 - P73 tạo nên Port 7. Cổng này có thể được lập trình như là một lối vào hoặc là một lối ra.

/CS (Chip Select)

Một mức High ở chân /CS ngăn ngừa việc một lối ra nào đấy hoặc là trạng thái bên trong bị thay đổi. Một cuộc trao đổi thông tin với vi mạch 8243 chỉ có thể tiến hành khi mà đường dẫn Chip Select có mức Low.

PROG

Bằng việc chuyển dời High - Low (sườn âm) ở lối vào PROG vi mạch 8243 đón nhận từ điều khiển đứng xếp hàng ở cổng 2. Còn sự chuyển mức từ Low sang High báo hiệu là ở cổng 2 đang có các dữ liệu chờ xử lý.



Hình 2.8 Giải đồ xung ở một lần truy nhập ghi và đọc của vi mạch 8243.

Một cuộc truyền dữ liệu đầy đủ bằng các cổng vào/ ra diễn biến theo cách sau:

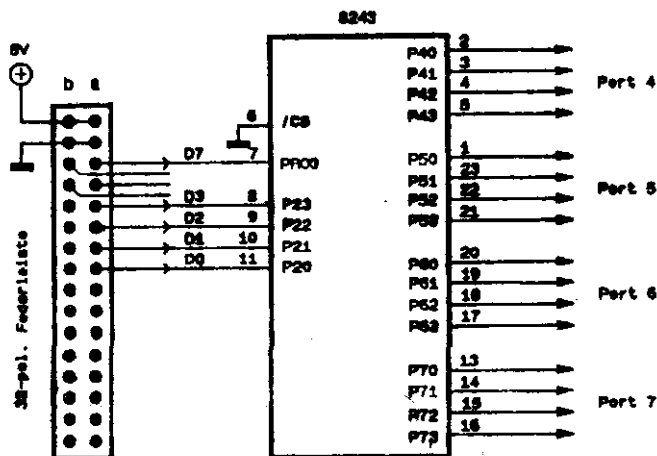
Mỗi một cuộc truyền dữ liệu có hai từ dữ liệu 4 bit tham gia. Từ đầu tiên chứa từ điều khiển, từ tiếp theo là từ dữ liệu. Trong một lần truy nhập đọc, sườn âm ở lối vào PROG lưu trữ từ điều khiển đúng xếp hàng ở cổng 2 vào trong linh kiện. Từ điều khiển báo hiệu cho vi mạch 8243 là bây giờ các dữ liệu của một cổng vào/ ra được đọc. Vì mạch 8243 giải mã từ này ở bên trong và đặt các đường dẫn lối vào đã được chọn lựa vào cổng 2, bây giờ ở đó chúng có thể được gọi ra từ các linh kiện khác. Sườn dương (sườn dốc lên) ở chân ra PROG kết thúc chu trình đọc và chuyển mạch cổng vào/ ra sang trạng thái điện trở cao. Sau đó cổng 2 trở lại chức năng như là một lối vào. Để gửi các dữ liệu tới một cổng vào/ ra mong muốn thì đồng thời từ điều khiển được xếp sát vào, từ này được lưu trữ và giải mã bằng sườn âm (sườn dốc xuống) của tín hiệu PROG trong vi mạch 8243. Bây giờ nửa byte dữ liệu có thể được xếp sát vào cổng 2 và sau đấy có thể xuất hiện ở lối ra được chọn lựa nhờ sườn dương của tín hiệu ở chân PROG.

Thí dụ khi ta muốn xuất dữ liệu qua cổng 5 thì ta cho từ điều khiển P23... P20 = 0101. Chức năng ORLD cũng như ANLD thực hiện từng bit một một liên kết logic HOẶC cũng như VÀ giữa các dữ liệu mới và các dữ liệu đúng xếp hàng sẵn ở cổng. Kết quả liên kết được dẫn tiếp đến cổng vào/ ra đã được lựa chọn.

2.4.2 MUỖI SÁU LỐI RA TTL VỚI 8243

Vì mạch 8243 có khả năng ghép nối với các môđun cơ sở đã trình bày ở chương 1 để mở rộng các tín hiệu vào/ ra đã có mặt ở đó. Sau đấy theo cách này có thể thiết kế ra những mạch điện phức tạp hơn, các mạch này đòi hỏi nhiều hơn là 8 lối ra và lối vào số. Hình 2.9 chỉ ra sự điều khiển của vi mạch 8243 qua ổ nối 32 chân mà tất cả các môđun cơ sở đã trình bày ở chương 1 đều có thể đấu vào ổ nối này như là một giao diện chung.

Các đường dẫn dữ liệu D0 - D3 được nối trực tiếp với cổng 2, còn D7 được dẫn đến chân ra PROG. Với năm đường dẫn dữ liệu được sử dụng sau đó có thể điều khiển 16 lối ra.



Hình 2.9 16 lối ra TTL với vi mạch 8243.

Các câu lệnh dùng trong QBasic để xuất ra một nửa byte dữ liệu qua một trong bốn cổng vào/ra có thể tóm tắt lại theo một cách tốt nhất là đặt vào trong một chương trình con. Sau đó trong chương trình chính ta chỉ cần có một câu lệnh để trao đổi với cổng mong muốn.

Qua một môđun cơ sở đã trình bày ở chương 1 và được đấu vào giao diện nối tiếp của máy tính PC, vi mạch 8243 có thể được trao đổi theo một chương trình như sau:

```

=====
'
' Program: 16BITCOM
'
' Function: Bằng chương trình này có thể qua cổng
' nối tiếp trao đổi với 16 đường dẫn nhu là lối vào
' hoặc là lối ra. Trong đó có 4 Port 4 bit để sử
' dụng.
'
' Hardware: Cần có một môđun cơ sở đã trình bày ở
' chương 1 cũng như một môđun mô tả trên hình 2.9
'
=====
DECLARE SUB OUT. 8243 (daten, port)
'

DIM SHARED outbyte
'----- 9600 baud, nối vào COM2 -----
OPEN "com2: 9600,N,8,1,CS,DS" FOR RANDOM AS #1

```

```

outbyte = 255 'Tất cả các Datenbits của Môđun lên mức 1
PRINT #1, CHR$(outbyte);
CALL OUT. 8243 (15, 4)      ' Tất cả 4 bit ở Port 4 lên High
CLOSE 1
END
SUB OUT. 8243 (daten, port)

```

```

'=====
'  Unterprogram: out.8243
'
'  Function: Chương trình con này điều khiển các
'  đường dữ liệu của vi mạch 8243 sao cho 4 bit dữ
'  liệu có thể được xuất ra qua một cổng (Port)
'=====

'-----Xuất ra Controlword -----
outbyte = (outbyte AND (&HFO)) + port
PRINT #1, CHR$(outbyte);
'---Lập trình controlword: Prog từ High sang Low---
outbyte = outbyte AND (255 - 128)
PRINT #1, CHR$(outbyte);
'-----Xếp các dữ liệu (4 BIT) được xuất ra-----
outbyte = (outbyte AND (&HFO)) + daten
PRINT #1, CHR$(outbyte);
'-----Prog trở lại High-----
outbyte = outbyte OR 128
PRINT #1, CHR$(outbyte);
END SUB

```

Khi mà môđun cơ sở đã trình bày ở mục 1.1.4 tìm thấy ứng dụng cùng với giao diện máy in thì chương trình có thể viết như sau:

```

'=====
'  Program: 16 BITLPT
'
'  Function: Bằng chương trình này có thể trao đổi
'  với 16 đường dẫn lối ra TTL qua cổng máy in
'  song song

```

' Hardware: Cần có một môđun cơ sở mô tả ở chương 1
' cũng như môđun mô tả trên hình 2.9

' =====

DECLARE SUB OUT, 8243 (daten, port)

DIM SHARED outbyte, datreg, statreg, contrreg

' -----Thiết lập chế độ ban đầu-----

basadr = &H378

datreg = basadr 'Datenregister

statreg = basadr+1 'Statusregister

contrreg=basadr+2 'Controlregister

outbyte = 255 'Tất cả các bit dữ liệu
' của môđun lên mức log 1

OUT datreg, outbyte

CALL OUT. 8243 (15, 4) 'Tất cả 4 bit ở Port 4 lên High

CLOSE 1

END

SUB OUT. 8243 (daten, port)

' =====

' Unterprogram: out. 8243

' Function: Chương trình con này điều khiển các
' đường dẫn của vi mạch 8243 sao cho 4 bit có thể
' được xuất ra qua một cổng

' =====

' -----Cho ra controlword -----

outbyte = (outbyte AND (&HFO)) + port

OUT datreg, outbyte

' ---Lập trình controlword: Prog từ high sang low---

outbyte = outbyte AND (255 - 128)

OUT datreg, outbyte

' -----Xếp các dữ liệu (4 BIT) được cho ra-----

outbyte = (outbyte AND &HFO) + daten

OUT datreg, outbyte

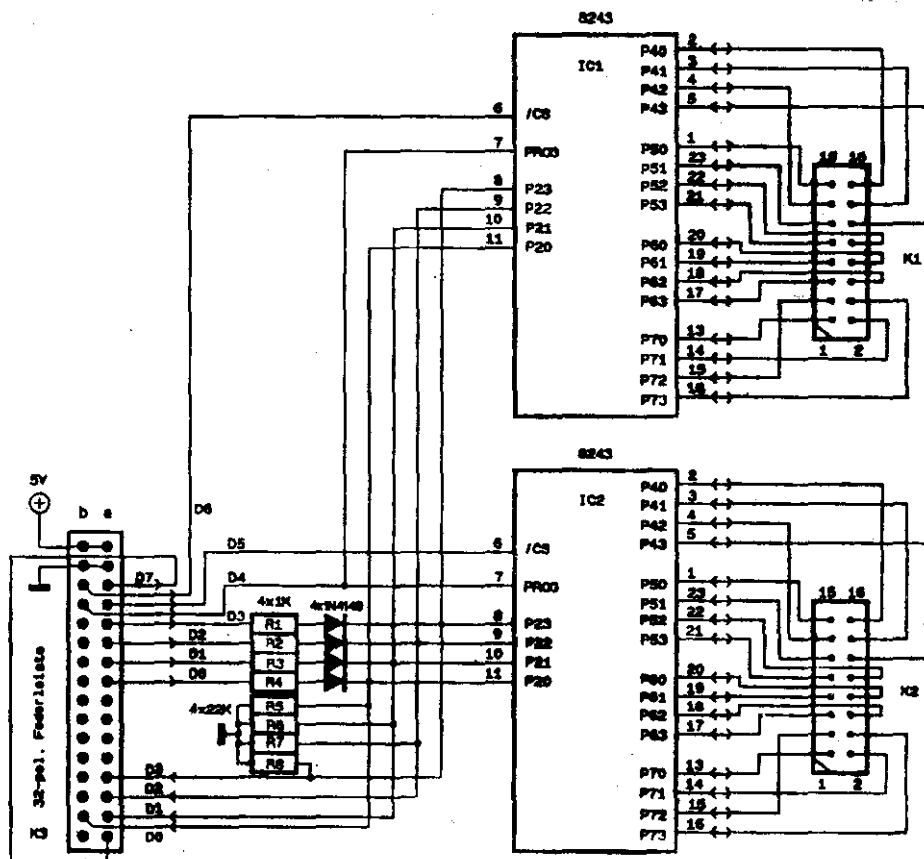
-----Prog trở lại high-----

outbyte = outbyte OR 128

OUT datereg, outbyte

END SUB

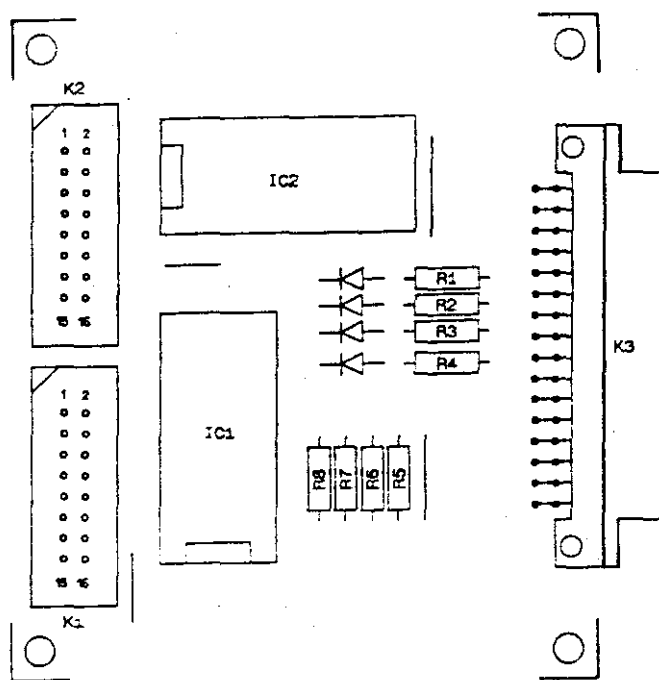
2.4.3 MÔĐUN VÀO/ RA 32 BIT



Hình 2.10 Môđun vào/ ra 32 bit với hai vi mạch 8243.

Trong mạch điện được minh họa trên hình 2.9, bên cạnh 3 đường dẫn lối ra D4, D5 và D6 cũng có tất cả các đường dẫn lối vào đều đã không được sử dụng. Tất cả các đường dẫn tự do này có thể đem dùng cho một vi mạch thứ hai cùng loại 8243. Hình 2.10 chỉ ra mạch điện của môđun giao diện, bằng môđun này ghép nối với môđun cơ sở đã mô tả ở chương 1 có thể sử dụng 32 tín hiệu TTL như là lối vào hoặc là lối ra.

Các đường dẫn dữ liệu D0 đến D3 được dẫn qua các điện trở và điốt đến các bus dữ liệu và bus điều khiển của vi mạch 8243. Các điốt ngăn ngừa các dữ liệu đứng xếp hàng ở cổng 2 gây ra sự chập mạch với các đường dẫn lối ra D0 đến D3 trong khi đọc vào một cổng vào/ ra. Thí dụ trường hợp này có thể xảy ra khi D0 có mức Low và đồng thời ở P20 đang có mức High. Các đường dẫn dữ liệu D5 và D6 lọc lựa bằng một mức Low, đây là mức của hai linh kiện được trao đổi. D7 sinh ra xung Low, mở đầu cuộc gửi đi của các dữ liệu xếp kế sát qua giao diện nối tiếp tới máy tính PC. Tín hiệu này được dẫn trực tiếp tới chân ra TBRL của bộ UART CDP6402 (xem thêm mục 1.2). Khi sử dụng môđun cơ sở dùng với cổng máy in (xem thêm mục 1.1.4) đường dẫn dữ liệu D7 có thể được giữ nguyên không để ý đến. 32 tín hiệu TTL có thể được phân nhánh ở hai ổ nối 16 chân K1 và K2.



Hình 2.11 Sắp xếp linh kiện trên môđun vào/ ra 32 bit.

Hình 2.11 và 2.12 chỉ ra sơ đồ sắp xếp linh kiện cũng như mạch in. Khi bạn đọc muốn sử dụng sơ đồ mạch in này để tự làm ra mạch ghép

nối thì phải chú ý là dòng chữ (C) DT96 - 32BITIO ở phía hàn các linh kiện phải được để cho đúng chiều.

Các linh kiện cần thiết để lắp ráp ra mạch điện được liệt kê như dưới đây:

Linh kiện bán dẫn:

IC1, IC2: 82C43

Điốt bán dẫn:

D1, D2, D3, D4: 1N 4148

Điện trở:

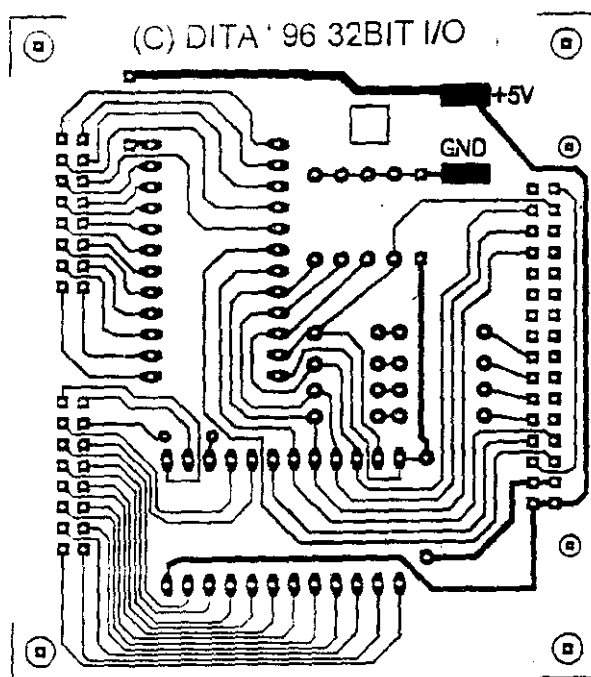
R1, R2, R3, R4 = 1 k Ω

R5, R6, R7, R8: = 22 k Ω

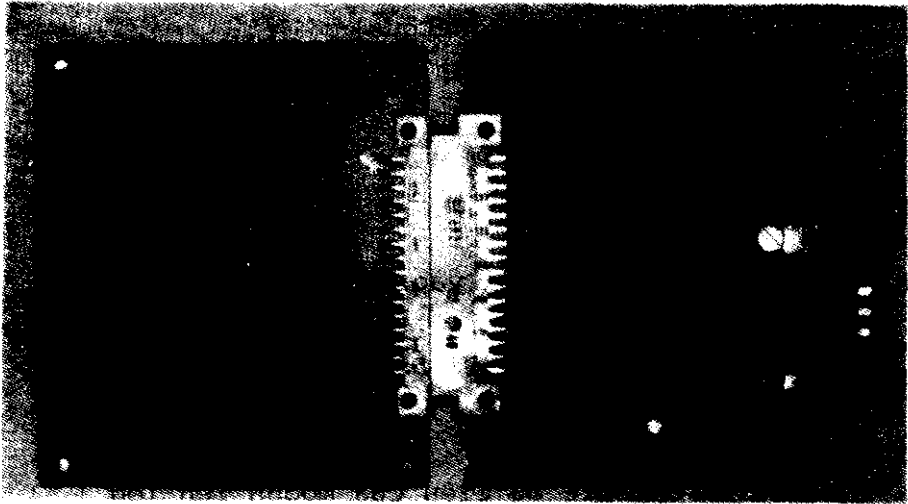
Phích cắm:

K1: dẹt 32 chân

K2: 25 chân



Hình 2.12 Sơ đồ mạch in của môđun vào/ ra 32 bit.



Hình 2.13 Môđun vào/ ra 32 bit cùng với một môđun cơ sở.

Sau khi lắp ráp xong ta có thể đem ghép nối với một môđun cơ sở. Trên hình 2.13 là một môđun vào/ ra 32 bit cùng với một môđun cơ sở dùng với cổng nối tiếp RS 232 đã mô tả ở mục 1.2. Với hai bản mạch này ta có thể trao đổi với 32 tín hiệu TTL như là lối vào hoặc là lối ra qua cổng nối tiếp của máy tính PC.

Phần mềm để điều khiển môđun vào/ ra 32 bit có thể viết như chương trình sau đây:

```

'-----
' Program: 32BITCOM
'
' Function: Bằng chương trình này có thể qua cổng
' nối tiếp trao đổi với 32 đường dẫn như là lối vào
' hoặc là lối ra. Trong đó có 8 Port 4 bit để sử
' dụng. Bằng cách thay đổi đi tít chút, ta cũng có
' thể áp dụng chương trình dưới đây cho cổng máy in.
'
' Hardware: Cần có một môđun cơ sở đã trình bày ở
' chương 1 cũng như một bộ ghép nối vào/ ra 32 bit
'-----

```

```

DECLARE SUB OUT, 8243 (daten,.port)
DECLARE SUB in.8243 (Port, ketqua)
DECLARE SUB lese.com (inbyte)

DIM SHARED outbyte
COLOR 0, 15

'-----9600 baud, nối vào COM2-----

OPEN "com2: 9600,N,8,1,CS,DS" FOR RANDOM AS #1

outbyte = 255

'Tất cả các bit dữ liệu của môđun lên log 1
PRINT #1, CHR$ (outbyte);

```

Again:

```

CLS
a$ = "32 - B I T I / O - I N T E R F A C E"
a = LEN (a$): b = (80 - a) / 2 - 1
LOCATE 1, b: PRINT CHR$ (201); STRING$
(a + 2, CHR$ (205)); CHR$ (187)
LOCATE 2, b: PRINT CHR$ (186)
LOCATE 2, b + 2: PRINT a$
LOCATE 2, a + b + 3: PRINT CHR$ (186)
LOCATE 3, b: PRINT CHR$ (200); STRING$
(a + 2, CHR$ (205)); CHR$ (188)
PRINT

```

label1:

```

INPUT "Vi mạch nào: IC (IC1 = 1 hoặc IC2 = 2) bạn
muốn trao đổi ?"; ic
IF ic < 1 OR ic > 2 THEN GOTO label1
'Bây giờ CS - Signal tương ứng được đặt lên Low
IF ic = 1 THEN outbyte = outbyte AND (255 - 64)
IF ic = 2 THEN outbyte = outbyte AND (255 - 32)
PRINT #1, CHR$ (outbyte);
PRINT

```

label2:

```
INPUT "Qua Port nào (4, 5, 6 hoặc 7) ?"; Port
IF Port < 4 OR Port > 7 THEN GOTO label2
PRINT
INPUT "Đọc vào dữ liệu (1) hoặc là xuất ra (2) ?";
in.out
IF in.out = 1 THEN
CALL in.8243 (Port, ketqua)
ketqua = ketqua AND 15
PRINT "Các bit được đọc vào có giá trị";
ketqua
    'CS-Signal tương ứng được đặt trở lại High
IF ic = 1 THEN outbyte = outbyte OR 64
IF ic = 2 THEN outbyte = outbyte OR 32
PRINT #1, CHR$ (outbyte);
GOTO tieptuc
END IF
PRINT
```

label3:

```
INPUT "Bạn hãy cho dữ liệu bằng số thập phân
(<= 15)"; daten
IF daten > 15 OR daten < 0 THEN GOTO label3:
CALL OUT.8243 (daten, Port)
```

tieptuc: PRINT

```
INPUT "Again (j)"; a$
IF a$ = "j" THEN
GOTO Again
ELSE
GOTO Ketthuc
END IF
```

Ketthuc:

```
CLOSE 1
```

END

```
SUB in. 8243 (Port, ketqua)
```

```

'=====
'  Unterprogram: in. 8243
'  Function: Chương trình này điều khiển các đường
'  dẫn dữ liệu của vi mạch 8243 sao cho 4 bit có thể
'  được đọc vào qua một cổng (Port)
'=====
'-----Xuất ra từ điều khiển-----
outbyte = (outbyte AND (&HFO)) + Port - 4
PRINT #1, CHR$ (outbyte);
'---Lập trình từ điều khiển; Prog từ High sang Low ---
outbyte = outbyte AND &HEF
PRINT #1, CHR$ (outbyte);
'----Đọc dữ liệu vào: TBRL lên LOW và trở lại HIGH---
outbyte = outbyte AND 9255 - 128)
PRINT #1, CHR$ (outbyte);
outbyte = outbyte OR 128
PRINT #1, CHR$ (outbyte);
CALL lese.com (inbyte)
ketqua = inbyte
'----- Prog trở lại high -----
outbyte = outbyte OR 16
PRINT #1, CHR$ (outbyte);

END SUB
SUB lese.com (inbyte)
'=====
'  Unterprogram: lese.com
'  Function: Chương trình con này đọc vào một byte
'  qua cổng nối tiếp. Nếu như cuộc truyền dữ liệu bị
'  nhiễu thì sẽ có một thông báo hiện lên nhấp nháy
'  và chương trình kết thúc !
'=====
i = 0
DO i = i + 1
'-----Khi byte đang có thì loc (1) >=1-----
IF LOC (1) >= 1 THEN in$ = INPUT$ (1, #1)

```

```

inbyte = ASC (in$)
GOTO ketthuc
END IF
'----- Phép thử mới, đọc dữ liệu vào-----
LOOP UNTIL I = 1000 'Cực đại 1000 lần thử
'----- Không nhận được byte nào ! -----
CLS
PRINT "Cuộc truyền dữ liệu bị nhiễu..!"
PRINT
PRINT "Không nhận được ký tự nào!!!!!!"
PRINT
PRINT "Xin hãy kiểm tra lại dây nối, ổ cắm,
      phần cứng....,"
END
ketthuc:
END SUB
SUB OUT.8243 (daten, Port)
'=====
' Unterprogram: out.8243
' Function: Chương trình này điều khiển các đường
' dẫn của vi mạch 8243 sao cho 4 bit có thể được
' xuất ra qua một cổng (Port)
'=====
'-----Xuất ra từ điều khiển-----
outbyte = (outbyte AND (&HFO)) + Port
PRINT #1, CHR$ (outbyte);
'---Lập trình controlword: prog từ high sang low---
outbyte = outbyte AND &HEF
PRINT #1, CHR$ (outbyte);
'-----Xếp các dữ liệu (4 BIT) được cho ra-----
outbyte = (outbyte AND (&HFO)) + Daten
PRINT #1, CHR$ (outbyte);
'-----Prog trở lại mức High -----
outbyte = outbyte OR 16

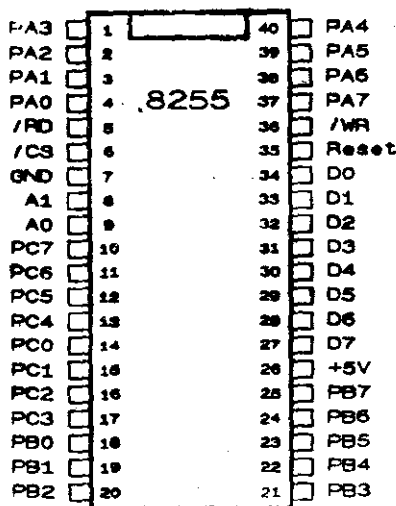
```

```
PRINT #1, CHR$(outbyte);
END SUB
```

2.5 GIAO DIỆN VÀO/ RA TRÊN RÃNH CẮM PC

2.5.1 VI MẠCH GHÉP NỐI 8255

Khối ghép nối 8255 thường hay được sử dụng trên tấm Card cắm thêm vào máy tính PC trong việc thu thập dữ liệu đo lường để xuất và nhập dữ liệu số, trong việc điều khiển quá trình biến đổi A/D. Vi mạch 40 chân này có các chân bố trí như ở trên hình 2.14, trong đó có chứa 24 đường dẫn lối vào/ ra được xếp thành ba cổng song song (cổng A, B và C).



Hình 2.14 Sơ đồ bố trí chân của vi mạch ghép nối 8255.

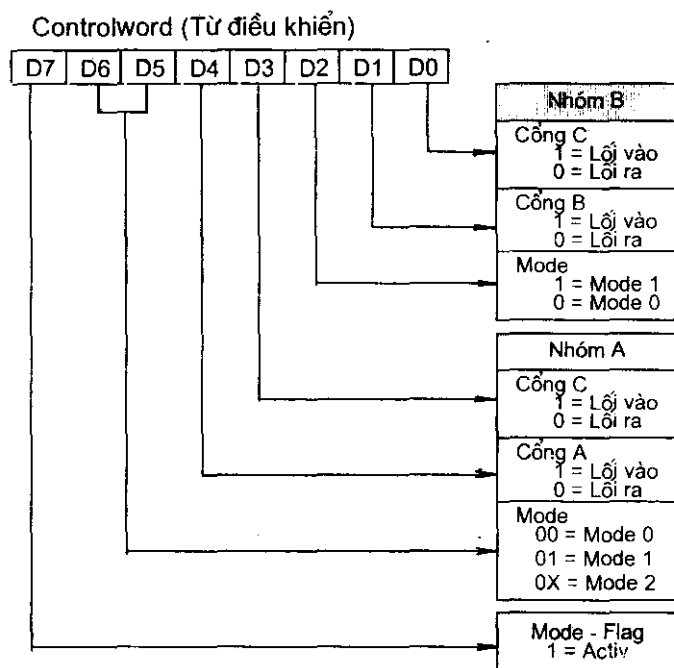
Tính linh hoạt của vi mạch này thể hiện ở khả năng lập trình. Qua một thanh ghi điều khiển, người sử dụng xác định loại hoạt động và cổng nào cần được sử dụng như là lối vào hoặc lối ra. Các chân ra D0 - D7 tạo nên bus dữ liệu hai hướng, rộng 8 bit. Tất cả các dữ liệu khi truy nhập ghi hoặc là đọc được dẫn qua các đường dẫn này. Trạng thái logic ghi/đọc được nhận biết qua các tín hiệu điều khiển /CS, /RD, /WR. Một cuộc trao đổi thông tin với vi mạch 8255 sau đó có thể được tiến hành khi /CS = 0. Khi /RD = 0, sau đó các tín hiệu của cổng được lựa chọn được dẫn đến bus dữ liệu và có thể được gọi bởi các vi mạch khác. Khi /WR = 0, thì mọi việc xảy ra ngược trở lại. Các bit địa chỉ A0 và A1 cùng

với các tín hiệu ghi và đọc báo hiệu cho biết sẽ truy nhập lên cổng nào. Bảng chân lý dưới đây tóm tắt các kết luận vừa mới trình bày ở trên.

A1	A0	/RD	/WR	/CS	Chức năng
0	0	0	1	0	Cửa (Port) A $\Rightarrow$ Bus dữ liệu
0	1	0	1	0	Cửa (Port) B $\Rightarrow$ Bus dữ liệu
1	0	0	1	0	Cửa (Port) C $\Rightarrow$ Bus dữ liệu
0	0	1	0	0	Bus dữ liệu $\Rightarrow$ Cửa (Port) A
0	1	1	0	0	Bus dữ liệu $\Rightarrow$ Cửa (Port) B
1	0	1	0	0	Bus dữ liệu $\Rightarrow$ Cửa (Port) C
1	1	1	0	0	Sự ghi vào của từ điều khiển
X	X	X	X	1	Đường dẫn dữ liệu ở trạng thái điện trở cao (tristate)
X	X	1	1	0	Đường dẫn dữ liệu ở trạng thái điện trở cao (tristate)

Từ bảng này rõ ràng là thanh ghi điều khiển đặt dưới địa chỉ bên trong $A0 = 0$, $A1 = 1$. Trong một chu trình ghi lên thanh ghi điều khiển, người sử dụng khẳng định cổng vào/ra cũng như kiểu hoạt động trong một từ điều khiển. Một mức High ở đường dẫn Reset đặt trở lại thanh ghi điều khiển và định nghĩa toàn bộ 24 đường dẫn như là các lối vào. Sự sắp xếp của từ điều khiển được mô tả trên hình 2.15.

Từ đây ta thấy rõ ràng là 24 đường dẫn của các cổng A, B và C được phân chia thành hai nhóm. Trong đó một nửa của cổng C (PC4...PC7) thuộc vào nhóm A, còn nửa kia thuộc nhóm B. Tổng cộng, người sử dụng có ba kiểu hoạt động khác nhau. Kiểu (mode) 0, kiểu 1 và kiểu 2. Dưới đây sẽ chỉ có kiểu 0 được mô tả. Loại hoạt động này giới thiệu một khả năng xuất và nhập dữ liệu theo cách đơn giản qua 3 cổng A, B, và C.



Hình 2.15 Sự sắp xếp của từ điều khiển ở vi mạch 8255.

Đối với loại hoạt động theo kiểu 0, từ điều khiển có thể tính theo cách sau:

	Port A	Port C Bit 4 ... 7	Port B	Port C Bit 0 ... 3
Lỗi vào:	16	8	2	1
Lỗi ra:	0	0	0	0
Từ điều khiển:	128 +	+	+	+

Thí dụ 1: Kiểu 0, cổng A như là lỗi vào, cổng B như là lỗi ra, cổng C như là lỗi ra.

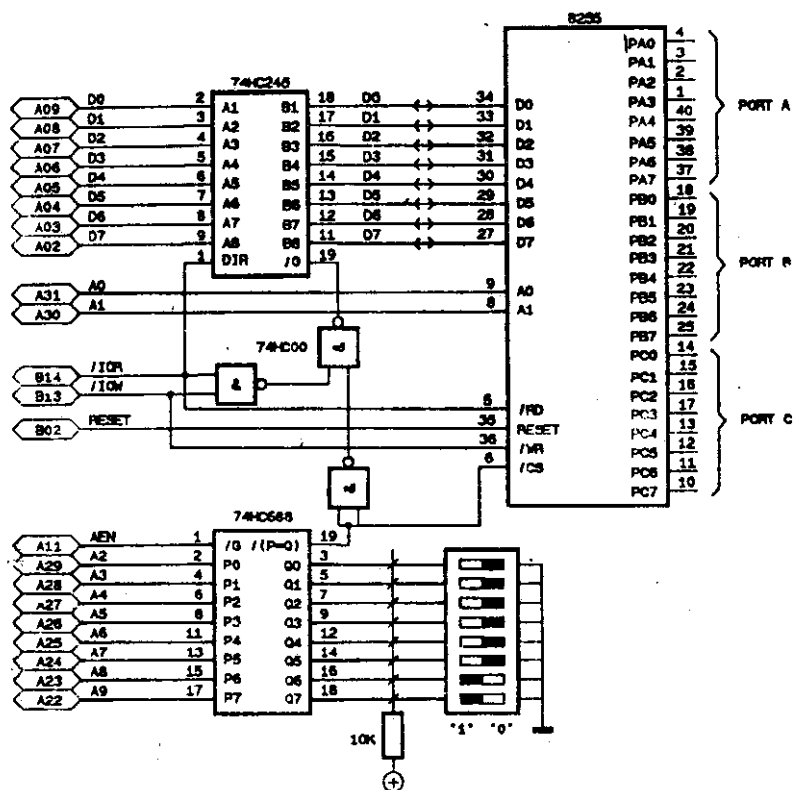
$$\text{Từ điều khiển (Controlword)} = 128 + 16 + 0 + 0 + 0 = 144$$

Thí dụ 2: Kiểu 0, cổng A như là lỗi ra, cổng C (bit 4... 7) như là lỗi vào, cổng B như là lỗi vào và cổng C (bit 0... 3) như là lỗi ra.

$$\text{Từ điều khiển (Controlword)} = 128 + 0 + 8 + 2 + 0 = 138.$$

2.5.2 GIAO DIỆN VÀO/ RA 24 BIT TRÊN RÃNH CẮM PC

Việc ghép nối vi mạch 8255 vào rãnh cắm của máy tính PC có thể quản lý một cách rất đơn giản. Hình 2.16 chỉ ra sơ đồ mạch điện của một giao diện vào/ ra 24 bit dùng cho rãnh máy tính PC. Các tín hiệu ghi và đọc /IOW và /IOR có thể được nối trực tiếp với các tín hiệu tương ứng ở vi mạch 8255. Còn chính đường dẫn Reset của rãnh cắm PC được dẫn đến vi mạch 8255 mà không cần một khối logic phụ thêm nào.



Hình 2.16 Giao diện vào/ ra 24 bit trên rãnh cắm PC.

Địa chỉ cơ bản mà dưới địa chỉ này mạch điện được trao đổi có thể được thiết lập ở chuyển mạch DIP. Chính mạch điện đã ghi tên 4 địa chỉ bắt đầu bằng địa chỉ cơ bản của vùng vào/ ra của một máy tính PC.

Địa chỉ vào/ ra		Chức năng
Địa chỉ cơ bản (Hex)	chẳng hạn \$ 300	Cửa (Port) A
Địa chỉ cơ bản + 1	chẳng hạn \$ 301	Cửa B
Địa chỉ cơ bản + 2	chẳng hạn \$ 302	Cửa C
Địa chỉ cơ bản + 3	chẳng hạn \$ 303	Thanh ghi điều khiển

Khi có sự đồng nhất giữa địa chỉ cơ bản được thiết lập và các địa chỉ đang đứng xếp hàng ở rãnh cắm PC, vi mạch 74HC688 sinh ra ở chân 19 một tín hiệu Low, tín hiệu này kích hoạt vi mạch 8255 qua đường dẫn CS. Nói chung, tín hiệu Low này tạo khả năng cho một cuộc trao đổi thông tin bằng vi mạch ghép nối 8255.

Chương trình dùng làm thí dụ dưới đây sẽ chỉ ra những lệnh nào trong QBasic cần dùng cho cuộc trao đổi thông tin.

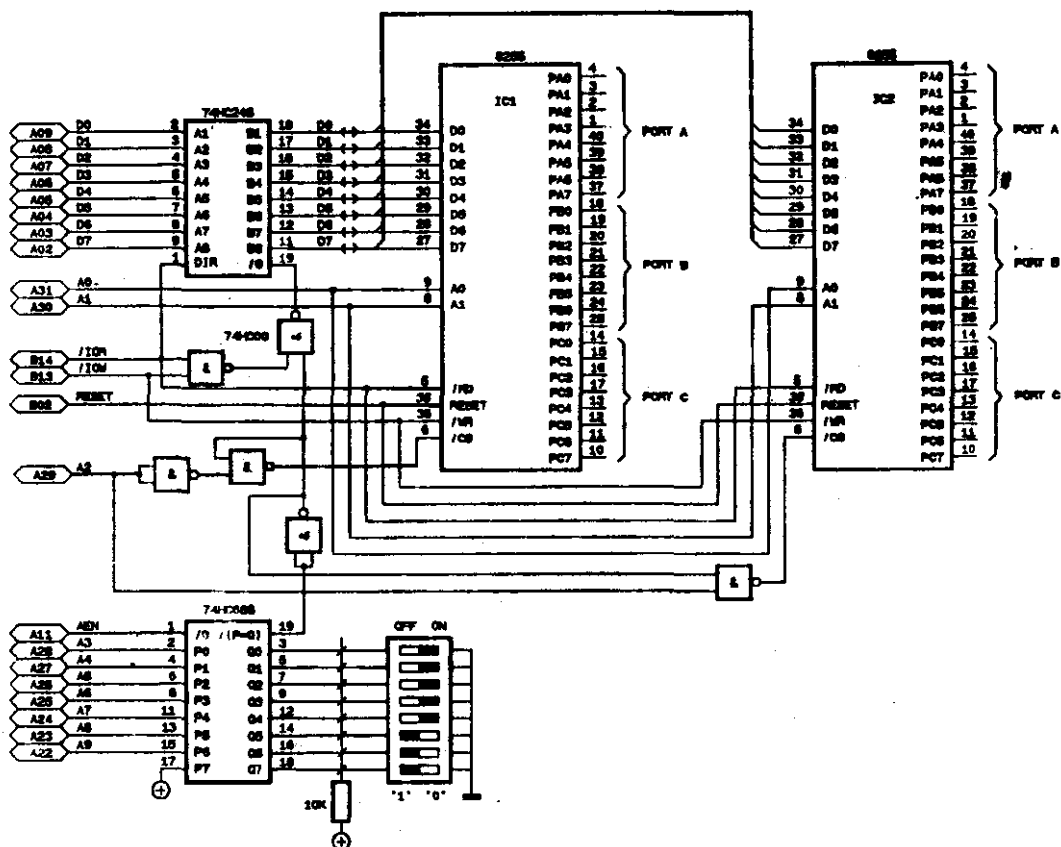
```
'=====
' Program:8255BSP
' Function: Bằng chương trình này có thể trao đổi
' với mạch giao diện I/O ở rãnh cắm PC mô tả trên
' hình 2.16
'=====
'-----Khẳng định các địa chỉ vào/ ra (I/O)-----
basadr = &H300          'Basicaddress, thí dụ $300
porta = basadr          'Port A
portb = basadr+1        'Port B
portc = basadr+2        'Port C
contrreg = basadr + 3    'Controlregister
'-----Thí dụ: Controlword = 144-----
' Port A như là Input, Port B và Port C như là Output
'-----Ghi controlword vào controlregister-----
controlword = 144
OUT contrreg, controlword
'-----Gửi các dữ liệu tới cổng B -----
daten = 255
OUT portb, daten        'Tất cả các bit ở cổng B lên High
```

```

'-----Gửi các dữ liệu tới cổng C -----
daten = 0
OUT portc, daten      'Tất cả các bit ở cổng C lên Low
'----- Đọc các dữ liệu từ cổng A -----
inbyte = INP(porta)
PRINT "Các dữ liệu ở cổng A là: "; inbyte
END

```

2.5.3 GIAO DIỆN VÀO/ RA 48 BIT TRÊN RÃNH CẮM PC



Hình 2.17 Giao diện vào/ ra 48 bit trên rãnh cắm PC.

Khi muốn mở rộng mạch điện của giao diện vào/ra 24 bit thêm 24 đường dẫn vào/ ra nữa, thì phải bổ sung thêm một vi mạch 8255 vào

mạch điện như minh hoạ trên hình 2.17. Mạch logic giải mã phải mở rộng thêm hai cổng NAND nữa và ngoài ra đường dẫn Chip Select của vi mạch 8255 cần phải được điều khiển. Việc lựa chọn xem cho khối ghép nối nào trong hai khối trở nên được kích hoạt do bit địa chỉ A2 ở rãnh cắm PC đảm nhiệm. Khi A2 dẫn tới mức Low và ngoài ra địa chỉ cơ bản đã được đặt đúng (chân 19 của 74HC688 = 0) thì có thể tiến hành trao đổi thông tin bằng vi mạch IC1. Khi A2 = 1 và địa chỉ cơ bản là đúng thì vi mạch 8255 thứ hai trở nên hoạt động.

Mạch điện trên hình 2.17 ghi tên tổng cộng 8 địa chỉ, các địa chỉ này được viết như sau:

Địa chỉ vào / ra		Chức năng
Địa chỉ cơ bản (Hex)	chẳng hạn \$ 300	Cửa A, IC1
Địa chỉ cơ bản + 1	chẳng hạn \$ 301	Cửa B, IC1
Địa chỉ cơ bản + 2	chẳng hạn \$ 302	Cửa C, IC1
Địa chỉ cơ bản + 3	chẳng hạn \$ 303	Thanh ghi điều khiển, IC1
Địa chỉ cơ bản + 4	chẳng hạn \$ 304	Cửa A, IC2
Địa chỉ cơ bản + 5	chẳng hạn \$ 305	Cửa B, IC2
Địa chỉ cơ bản + 6	chẳng hạn \$ 306	Cửa C, IC2
Địa chỉ cơ bản + 7	chẳng hạn \$ 307	Thanh ghi điều khiển, IC2

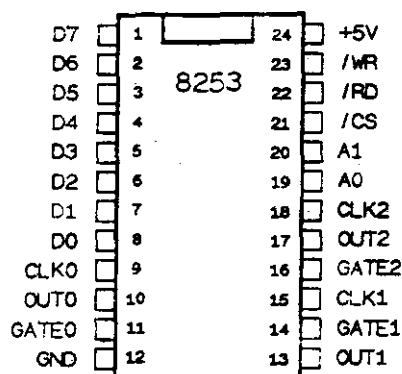
2.6 CÁC BỘ ĐẾM VẠN NĂNG

2.6.1 VI MẠCH ĐẾM 8253

Vi mạch đếm 8253 là một bộ đếm có thể gặp trong nhiều ứng dụng. Con số các ứng dụng khả dĩ nhiều đến mức đáng ngạc nhiên: bộ đếm xung, bộ đếm tần số, bộ phát xung lập trình được, thiết bị thử phản ứng, máy đo điện dung, bộ điều khiển độ sáng, bộ biến đổi D/A v. v... chỉ là một vài thí dụ được kể ra. Để có thể sử dụng được vi mạch này trong thực tế thì điều bắt buộc là phải tìm hiểu kỹ lưỡng nguyên tắc hoạt động của vi mạch này sẽ được mô tả một cách đầy đủ trong mục dưới đây.

Vi mạch đếm 8253 chứa ba bộ đếm lùi 16 bit độc lập với nhau. Mỗi một bộ đếm có ba chân ra tương ứng: lối vào giữ nhịp CLK, cửa GATE và lối ra tín hiệu OUT.

Thoạt nhìn việc sử dụng các bộ đếm lùi có thể là không logic. Để hiểu rõ ta nên chú ý tới cách đặt vấn đề là phải đếm đúng 10.000 xung và tiếp đó là một tín hiệu để kích hoạt. Với mỗi sườn âm ở lối vào CLK, hiện mức ở bộ đếm lại bị giảm đi cho đến khi đạt giá trị bằng 0. Sự giải mã của hiện mức 0 ở bộ đếm (đã đếm 10.000 xung) sau đó được thực hiện ở bên trong của vi mạch theo cách đơn giản đáng kể hơn là sự giải mã của giá trị xuất phát, ở đây là 10.000 nhưng tất nhiên nó có thể nhận các giá trị khác. Hình 2.18 chỉ ra sự sắp xếp chân của vi mạch đếm 8253.



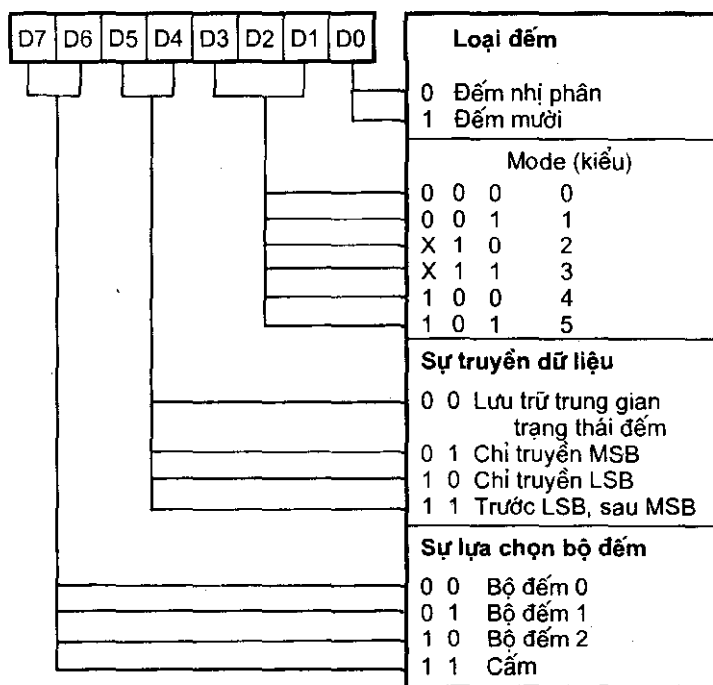
Hình 2.18 Sắp xếp chân của mạch đếm 8253.

Các đường dẫn dữ liệu D0 đến D7 biểu diễn các bus dữ liệu theo hai hướng, qua đó tất cả các dữ liệu khi truy nhập ghi hoặc là đọc được dẫn đến.

Trong một lần truy nhập đọc ($/RD = 0$) hiện mức ở bộ đếm có thể được gọi ra, còn trong một lần truy nhập ghi ($/WR = 0$) hiện mức của bộ đếm được ghi vào một thanh ghi điều khiển và sau đó bộ đếm được nạp bằng các giá trị bắt đầu. Qua các bit địa chỉ A0 và A1 ta đạt đến ba bộ đếm và thanh ghi điều khiển. Bảng chân lý sau chỉ ra một cách nhìn tổng quan về chức năng của các đường dẫn điều khiển ở vi mạch đếm 8253.

A1	A0	/RD	/WR	/CS	Chức năng
0	0	0	1	0	Đọc ra từ bộ đếm 0
0	1	0	1	0	Đọc ra từ bộ đếm 1
1	0	0	1	0	Đọc ra từ bộ đếm 2
0	0	1	0	0	Sự nạp của bộ đếm 0 bằng từ xuất phát
0	1	1	0	0	Sự nạp của bộ đếm 1 bằng từ xuất phát
1	0	1	0	0	Sự nạp của bộ đếm 2 bằng từ xuất phát
1	1	1	0	0	Sự ghi vào của từ điều khiển
X	X	X	X	1	Đường dẫn dữ liệu ở trạng thái điện trở cao (Tristate)
X	X	1	1	0	Đường dẫn dữ liệu ở trạng thái điện trở cao (Tristate)

Từ điều khiển khẳng định kiểu hoạt động và hệ thống đếm, đồng thời kiểm tra sự nạp của thanh ghi đếm. Một bộ đếm được lập trình, trong đó trước hết ta nạp thanh ghi điều khiển bằng số đếm đã gập, kiểu (mode) và hệ thống đếm.



Hình 2.19 Sắp xếp của từ điều khiển ở vi mạch 8253.

Sau đó ta ghi giá trị xuất phát của bộ đếm vào thanh ghi đếm tương ứng. Rồi cứ mỗi lần gặp sườn âm bộ đếm lại đếm lùi đi một. Trong các kiểu hoạt động riêng lẻ, sự đạt tới đích của bộ đếm (hiện mức 0 của bộ đếm) có ý nghĩa khác nhau. Hình 2.19 chỉ ra sự sắp xếp của các từ điều khiển ở vi mạch đếm 8253.

Bit dữ liệu D0 lựa chọn giữa các con số đã mã hoá nhị phân và BCD. Còn D1, D2 và D3 khẳng định 6 kiểu hoạt động khả dĩ và sau này sẽ còn được trình bày một cách đầy đủ. Ở trên đã nhắc tới bộ đếm 16 bit, nhưng mà lại chỉ có một bus dữ liệu 8 bit, nên một giá trị xuất phát 16 bit cần phải được truyền đi thành hai byte.

Trong hầu hết các ứng dụng, trước hết là bit có giá trị thấp nhất LSB (từ tiếng Anh: *Least Significant Bit*) sau đó là bit có giá trị cao hơn MSB (tiếng Anh: *Most Significant Bit*) được truyền đi. Chẳng hạn như để nạp giá trị xuất phát 50.000 vào một bộ đếm, thì trước hết giá trị hàng chục 80 sau đó là 195 trong bộ đếm đã gặp cần phải được ghi. Phép tính tương ứng có thể ghi ra như sau: $50.000 : 256 = 195 \text{ dư } 80$; ở đây 195 là MSB còn 80 là LSB. Từ hình 2.19 có thể tính toán một cách rất đơn giản từ điều khiển dùng cho loại đếm các số nhị phân, quá trình truyền trước hết là LSB rồi đến MSB.

$$\text{Controlword} = 2 \cdot \text{Mode} + 64 \cdot \text{Counter} + 48$$

Khi mà D4 = 0 và D5 = 0, hiện mức ở một bộ đếm có thể lưu trữ trung gian mà không cần làm ngừng quá trình đếm. Khi lưu trữ trung gian mức logic của đường dẫn dữ liệu D0 đến D3 không phải là quan trọng và do đó có thể được đặt lên 0. Thí dụ khi ta muốn lưu trữ hiện mức ở bộ đếm của bộ đếm thì giá trị 128 (dec.) phải được ghi vào thanh ghi điều khiển.

Trong trường hợp tổng quát, từ điều khiển cần thiết để lưu trữ trung gian một hiện mức bộ đếm có thể được tính như sau:

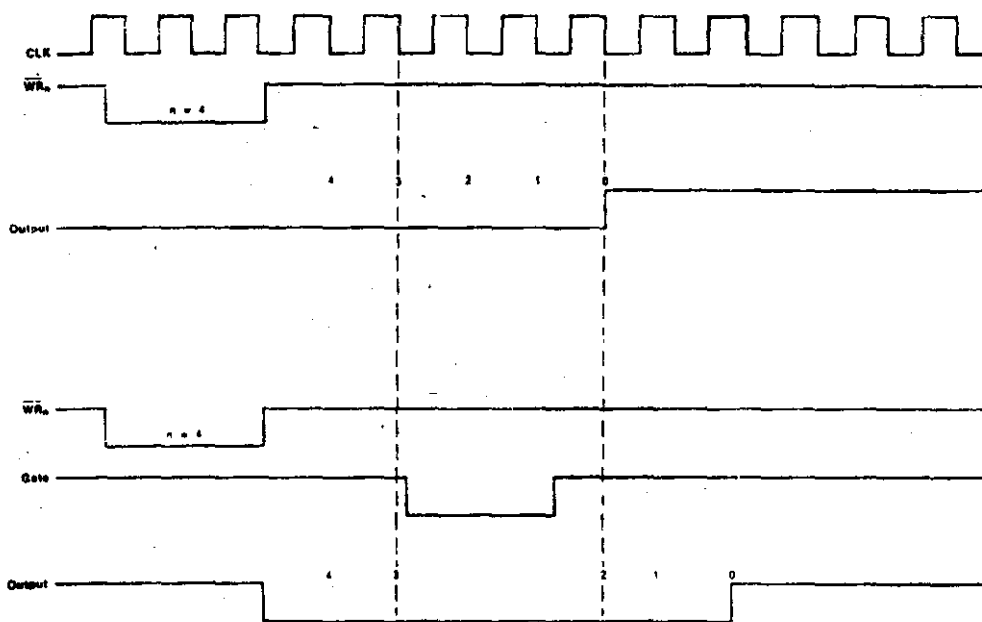
Lưu trữ trung gian trạng thái đếm (Counterstand):

$$\text{Controlword} = 64 \cdot \text{counter} + \text{mode}$$

Sự mô tả kiểu hoạt động riêng biệt có thể dẫn ra như sau:

Kiểu (mode) 0: Sườn dương bị làm trễ

Lỗi ra của bộ đếm (OUT 0/1/2) trước hết được chuyển sang mức Low sau khi có sự khẳng định loại hoạt động diễn ra trong lúc thiết lập trạng thái ban đầu của khối định thời gian Timer. Sau khi đã diễn ra quá trình nạp của thanh ghi đếm mong muốn bằng một giá trị xuất phát, bộ đếm tương ứng sẽ đếm lùi bắt đầu từ giá trị xuất phát, giả định, lỗi vào GATE nằm trên mức High. $GATE = 0$ làm ngừng quá trình đếm đang xảy ra, không kể là lỗi ra OUT thay đổi sang một dạng khác nào đấy. Khi lỗi vào GATE chuyển trở lại High, quá trình đếm bắt đầu với giá trị mà bộ đếm chỉ ra trước khi ngừng. Lỗi ra vẫn giữ nguyên mức Low trong khi đếm lùi. Khi đạt được hiện mức 0 ở bộ đếm thì lỗi ra chuyển sang High và dừng lại ở trạng thái này cho đến khi bộ đếm được cấu hình mới.



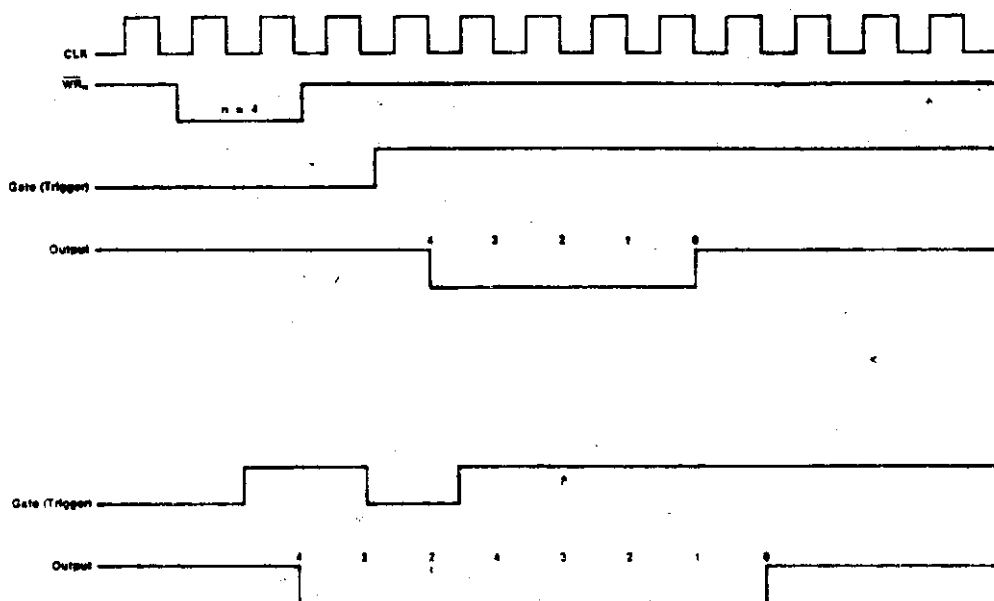
Hình 2.20 Giải đồ xung của kiểu hoạt động 0.

Cần phải chú ý là sau khi đạt đến sự kết thúc quá trình bộ đếm lại tiếp tục đếm lùi bắt đầu bằng giá trị 65.535, độc lập với việc giá trị xuất

phát nào đang đứng trong thanh ghi bộ đếm. Sau đó lối ra OUT dừng lại ở mức High cho đến khi bộ đếm được nạp mới bằng một giá trị xuất phát. Đặc biệt phải nhấn mạnh là sườn âm đầu tiên ở lối vào CLK không được đếm mà chỉ có giá trị xuất phát được đảm nhiệm như là hiện mức bộ đếm hiện thời. Sau đó một sườn âm của xung nhịp được dẫn đến lối vào CLK để làm giảm hiện mức ở bộ đếm đi 1.

Việc ghi mới vào thanh ghi bộ đếm trong khi quá trình đếm đang diễn ra dẫn đến hậu quả là khi ghi vào byte đầu tiên quá trình đếm dừng lại và khi ghi vào byte thứ hai thì quá trình đếm mới được khởi động.

Kiểu hoạt động 1: Mạch monoflop lập trình được



Hình 2.21 Giản đồ xung của kiểu hoạt động 1.

Khi ta chú ý đến giản đồ xung của kiểu hoạt động 1 trên hình 2.21 thì thoạt nhìn có vẻ như không có sự khác nhau cơ bản với kiểu hoạt động 0. Nhưng ấn tượng đầu tiên này không đúng. Ở cả hai kiểu hoạt động, lối ra OUT giữ nguyên ở trạng thái Low trong n chu kỳ giữ nhịp,

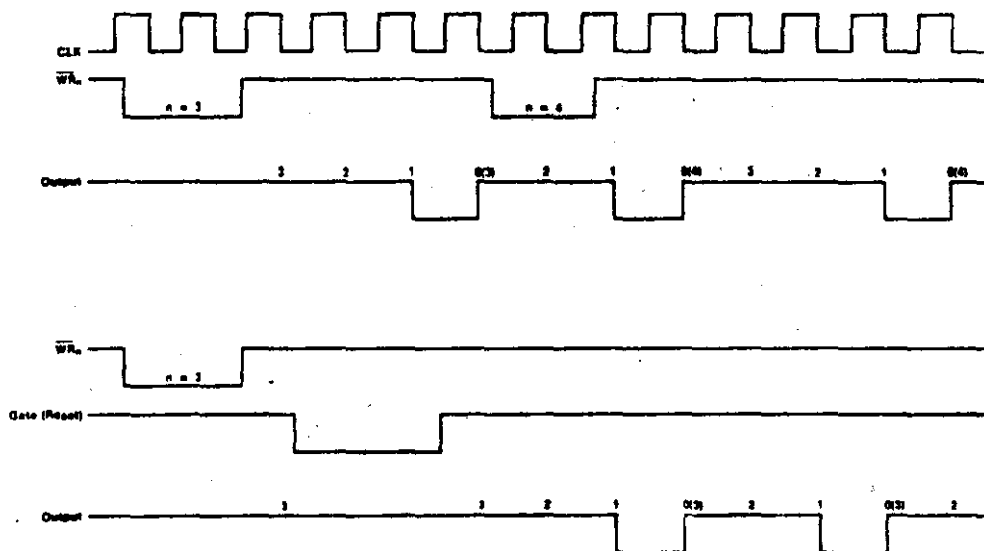
trước khi nó chuyển sang mức High ở hiện mức 0 ở bộ đếm. Trong đó, N là giá trị xuất phát của bộ đếm. Sự khác nhau cơ bản giữa hai kiểu hoạt động thể hiện ở chức năng của lối vào GATE. Sau khi nạp một bộ đếm bằng giá trị xuất phát của nó, lối ra OUT trước hết còn dừng lại ở mức High. Vậy thì sau khi có một sườn dương xuất hiện ở lối vào GATE, quá trình đếm mới bắt đầu, lối ra OUT chuyển sang mức Low bằng sườn âm của xung giữ nhịp đầu tiên ở chân CLOCK, nghĩa là bộ monoflop lập trình được cần phải được khởi động bằng phần cứng. Cũng cần chú ý là với mỗi sườn dương ở lối vào GATE, bộ đếm lại được nạp bằng giá trị xuất phát. Bằng quá trình này có thể kéo dài thời gian Low của mạch monoflop ra vô hạn. Hay là theo ngôn ngữ chuyên môn, ta có thể nói là mạch monoflop tái lập trạng thái được (retriggerable).

Khác với kiểu hoạt động 0, chức năng đếm cũng được bảo đảm tới khi $GATE = 0$. Sau khi đạt được hiện mức 0 ở bộ đếm thì bộ đếm bắt đầu đếm lùi từ giá trị 65.535, điều này xảy ra độc lập với giá trị xuất phát của bộ đếm. Một sự nạp mới của bộ đếm trong khi lối ra là Low, sẽ không có ảnh hưởng đến mức độ kéo dài của trạng thái Low. Hiện mức của bộ đếm có thể được đọc ra bất kỳ lúc nào mà không có ảnh hưởng lên quá trình đếm đang diễn ra.

Kiểu hoạt động 2: Bộ chia $1/n$

Trong kiểu hoạt động này lối ra OUT chuyển sang mức Low ở hiện mức 1 của bộ đếm trong khoảng thời gian một chu kỳ giữ nhịp. Sau khi đạt tới hiện mức 0, bộ đếm lối ra OUT chuyển lại sang mức High và bộ đếm bắt đầu đếm lùi từ giá trị xuất phát. Đây là điểm khác nhau cơ bản so với kiểu hoạt động 0 và 1, ở kiểu hoạt động này sau khi đạt hiện mức đếm 0, bộ đếm bắt đầu đếm lùi từ giá trị 65.535. Ở đây mức độ kéo dài của mức High ở tín hiệu lối ra OUT là $(n - 1)$ chu kỳ giữ nhịp, còn mức Low kéo dài một chu kỳ giữ nhịp nên chu kỳ của tín hiệu lối ra OUT bằng n lần tín hiệu CLOCK. Điều đáng chú ý là bộ đếm được khởi động mới bằng mỗi sườn dương ở lối vào GATE. Nhờ vậy mà lối vào GATE có thể được tận dụng để đồng bộ bộ đếm.

Trong thực tế phải đặc biệt chú ý là sườn âm đầu tiên ở lối vào CLOCK còn chưa được đếm, mà chỉ có giá trị xuất phát được chấp nhận như là hiện mức hiện thời ở bộ đếm.



Hình 2.22 Giải đồ xung của kiểu hoạt động 2.

Một ứng dụng của kiểu hoạt động 2 là sự phân tầng của bộ đếm. Sườn giữ nhịp dương của tín hiệu lối ra OUT ở hiện mức 0 của bộ đếm có thể được dẫn qua một bộ đảo đến lối vào CLOCK của bộ đếm tiếp theo.

Kiểu hoạt động 3: Bộ phát tín hiệu vuông góc.

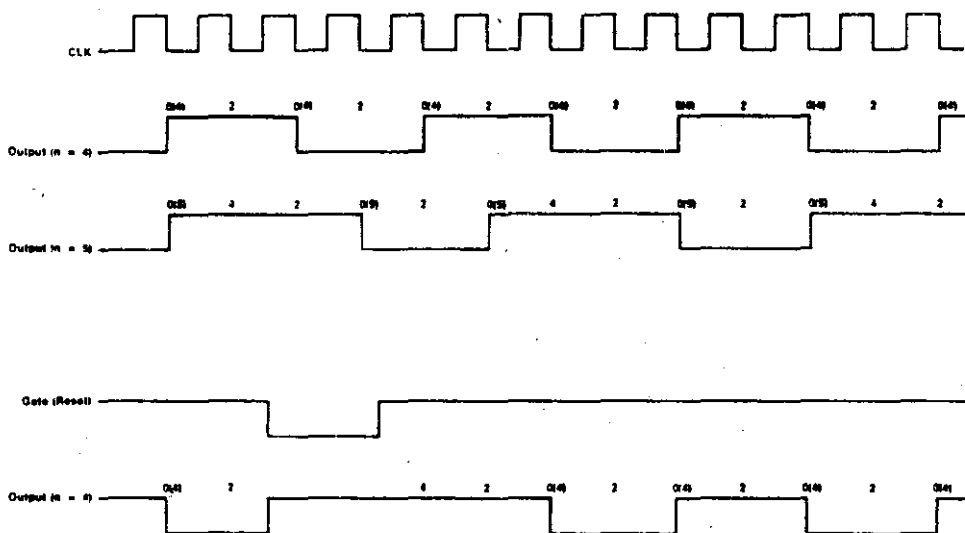
Trong kiểu hoạt động 3, bộ đếm 8253 có thể được sử dụng như là một bộ phát tín hiệu vuông góc lập trình được. Loại hoạt động này nhấn mạnh vai trò của khả năng về công suất của vi mạch đếm 8253. Chi phí cho phần cứng để có được chức năng này bằng các vi mạch logic, rời rạc là rất tốn kém. Trong kiểu hoạt động 3, tín hiệu lối ra OUT có mức High trong nửa đầu của quá trình đếm, trong nửa sau lối ra giữ nguyên ở mức Low. Nếu như giá trị đếm là lẻ thì khi có xung nhịp đầu tiên (sau khi nạp bộ đếm) giá trị này bị bớt đi 1. Các xung nhịp tiếp theo sẽ giảm bớt giá trị này đi 2. Khi đạt đến sự kết thúc quá trình đếm lối ra trở thành Low, và giá trị đếm đầy đủ được nạp trở lại. Xung nhịp kế sát

theo quá trình sẽ giảm bớt hiện mức ở bộ đếm đi 3, còn các xung nhịp tiếp theo lại chỉ giảm bớt đi 2. Quá trình này được lặp lại và nhờ thế - khi mà giá trị đếm là lẻ - lối ra sẽ trở thành High trong $(n+1)/2$ chu kỳ giữ nhịp và trở nên Low trong $(n-1)/2$ chu kỳ giữ nhịp.

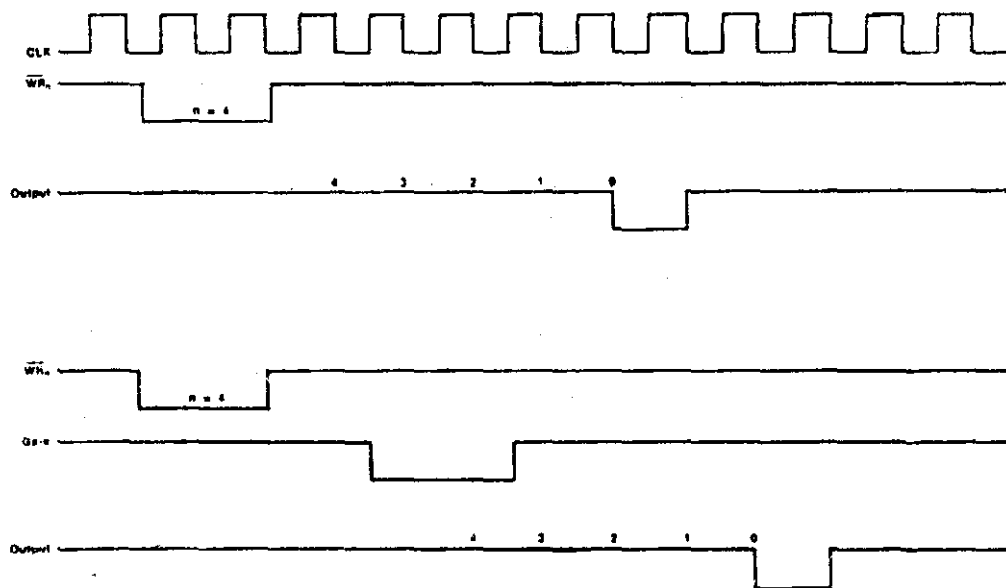
Cứ như thế bộ đếm làm việc giống như một bộ đếm lập trình được.

Hệ số chia có thể lựa chọn được giữa 2 và 65.535. Bằng việc chia tăng hai bộ đếm trong kiểu hoạt động 3 dẫn đến hệ số chia tổng cộng được tính bằng tích của hai hệ số chia. Theo cách này có thể tạo ra các tần số thấp với độ chính xác cao. Chẳng hạn như ta muốn tạo ra tần số so sánh là 50 Hz từ một bộ dao động thạch anh 4 MHz thì cần có một hệ số chia là 80.000.

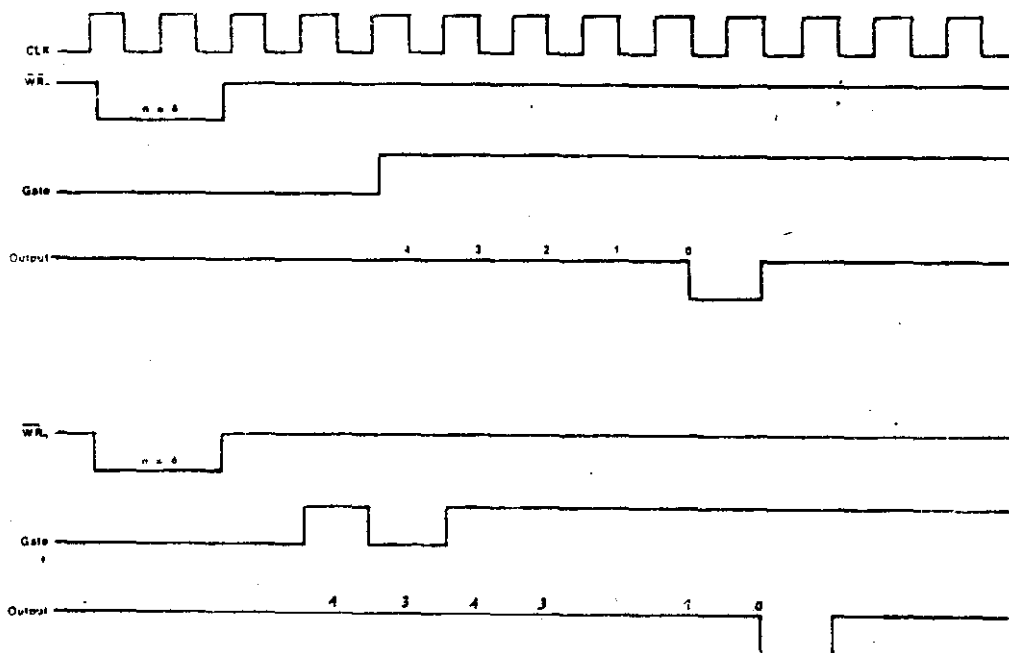
Để có được hệ số chia lớn như thế này, người sử dụng có nhiều giải pháp để lựa chọn. Thường thì cần có hai bộ đếm làm việc theo kiểu hoạt động 3 được đấu kế tiếp nhau. Chẳng hạn bộ đếm 1 có thể được nạp bằng 1.000 còn bộ đếm 2 bằng 80.



Hình 2.23 Giải đồ xung của kiểu hoạt động 3.



Hình 2.24 Giải đồ xung của kiểu hoạt động 4.



Hình 2.25 Giải đồ xung của kiểu hoạt động 5.

Kiểu hoạt động 4: STROBE tái lập trạng thái được bằng phần mềm

Lỗi ra của bộ đếm trước hết chuyển sang trạng thái High sau khi thiết lập trạng thái ban đầu cho khối định thời gian (Timer). Sau khi nạp bộ đếm, quá trình đếm bắt đầu. Khi đạt đến trạng thái kết thúc lỗi ra chuyển sang Low trong một chu kỳ giữ nhịp rồi chuyển trở lại sang High. Khi mà trong một quá trình đếm thanh ghi đếm được nạp mới thì giá trị đếm mới được nạp bằng xung giữ nhịp tiếp theo sau. Trong khi lỗi vào GATE dẫn đến mức Low, thì không có quá trình đếm nào có thể xảy ra.

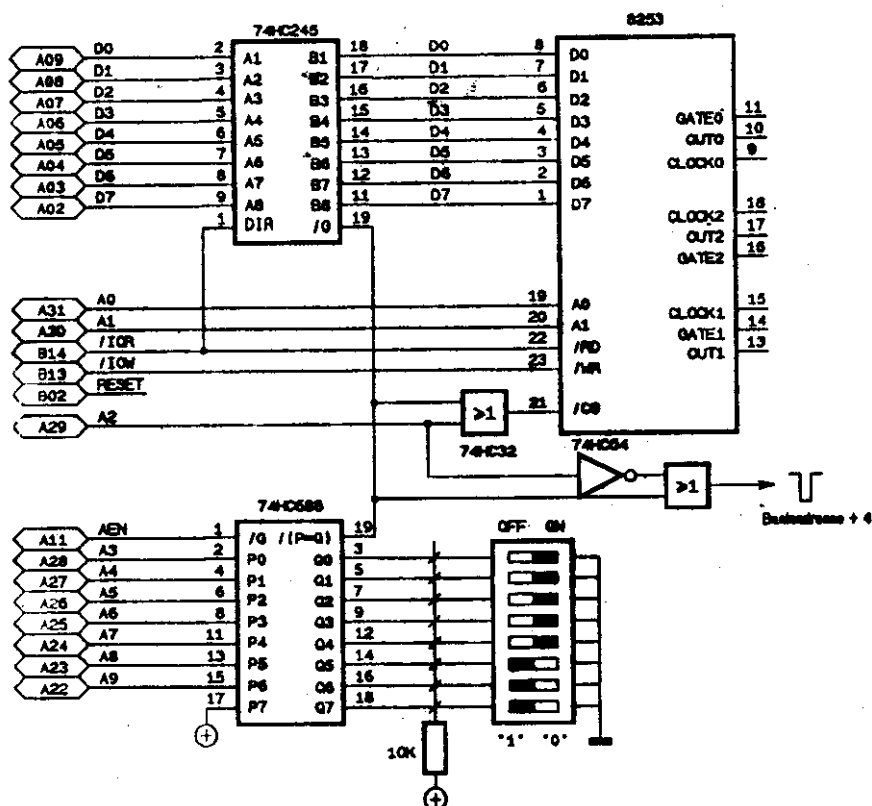
Kiểu hoạt động 5: STROBE tái lập trạng thái được bằng phần cứng

Trong kiểu hoạt động này bộ đếm bắt đầu quá trình đếm bằng sườn dương của xung ở lỗi vào GATE và khi kết thúc quá trình đếm sinh ra một mức Low trong một chu kỳ giữ nhịp. Bộ đếm là có thể tái lập trạng thái được.

2.6.2 GHEP NỐI BỘ ĐẾM VỚI RANH CẮM TRONG MÁY PC

Vi mạch đếm 8253 có thể ghép nối một cách rất đơn giản vào rãnh cắm của máy tính PC. Việc ráp nối tiến hành qua các đường dẫn dữ liệu D0 đến D7, các đường dẫn điều khiển /CS, /WR, /RD cũng như qua các đường dẫn địa chỉ A0 và A1. Hình 2.26 chỉ ra sơ đồ mạch điện dùng cho việc ghép nối bộ đếm. Ta nhận thấy sự có mặt của bộ đếm bus hai hướng 74HC245, đã được mô tả kỹ lưỡng trong mục 1.3, các đường dẫn dữ liệu của rãnh cắm PC được nối với vi mạch đếm qua bộ đếm 74HC245 này. Nhờ vậy mà khi truy nhập lên vi mạch đếm để ghi hoặc là đọc bộ đếm này đóng vai trò chuyển hướng của dòng dữ liệu. Còn vi mạch 74HC688 so sánh các đường dẫn dữ liệu kể sát rãnh máy tính PC theo tính đồng nhất với các địa chỉ cơ bản được đặt trước bởi chuyển mạch DIP, và khi có được sự đồng nhất sẽ cho ra một mức Low ở chân 19. Tín hiệu này được dẫn đến một cổng HOẶC (OR) bằng bit địa chỉ A2 của rãnh cắm của máy tính, lỗi ra của cổng này sẽ kích hoạt vi mạch đếm 8253. Ở đây điều kiện cần là hai tín hiệu lỗi vào có mức Low. Đồng thời tín hiệu A2 được dẫn qua một bộ đảo để tới một cửa HOẶC (OR) khác nữa; cửa này trong một lần truy nhập đọc sẽ tạo ra một xung Low

ngắn hạn; xung Low lại có thể được sử dụng trong nhiều ứng dụng của vi mạch đếm 8253.



Hình 2.26 Ghép nối bộ đếm với rãnh cắm trong máy PC.

Mạch điện chiếm chỗ với 8 địa chỉ bắt đầu từ địa chỉ cơ bản:

Địa chỉ vào/ ra		Chức năng
Địa chỉ cơ bản	chẳng hạn \$ 300	Bộ đếm 0
Địa chỉ cơ bản + 1	chẳng hạn \$ 301	Bộ đếm 1
Địa chỉ cơ bản + 2	chẳng hạn \$ 302	Bộ đếm 2
Địa chỉ cơ bản + 3	chẳng hạn \$ 303	Thanh ghi điều khiển
Địa chỉ cơ bản + 4	chẳng hạn \$ 304	Các xung LOW (so sánh với hình 2.26)
Địa chỉ cơ bản + 5	chẳng hạn \$ 305	Các xung LOW (so sánh với hình 2.26)
Địa chỉ cơ bản + 6	chẳng hạn \$ 306	Các xung LOW (so sánh với hình 2.26)
Địa chỉ cơ bản + 7	chẳng hạn \$ 307	Các xung LOW (so sánh với hình 2.26)

Chương trình 8253BSP dùng làm thí dụ dưới đây chỉ ra cho ta thấy khi lập trình cho một card cắm thêm vào có chứa vi mạch đếm 8253 thì cần những lệnh nào trong QBasic:

```
'=====
' Program:8253BSP
'
' Function: Tín hiệu vuông góc của một bộ dao động
' thạch anh với tần số f = 4 MHz được dẫn đến bộ đếm
' 0 và được chia cho 23580. Tín hiệu lỗi ra được dẫn
' đến lối vào CLOCK của bộ đếm 1, bộ này hoạt động
' như một bộ đếm xung đơn giản (mode 2). Kết quả đếm
' thường xuyên được đọc ra và hiển thị lên màn hình.
'=====
'-----Khẳng định các địa chỉ vào/ ra (I/O)-----
basadr = &H300                'Basicaddress, 300 hex
counter0 = basadr              'Bộ đếm 0
counter1 = basadr + 1          'Bộ đếm 1
counter2 = basadr + 2          'Bộ đếm 2
contrreg = basadr + 3          'Controlregister
'
'-----Thí dụ-----
' Bộ đếm 0: Mode 3, Startvalue: 23580 (Hệ số chia)
' Bộ đếm 1: Mode 2, Startvalue: 65535
'
mode 0 = 3                     'Mode Counter 0
mode 1 = 2                     'Mode Counter 1
start0 = 23580                 'Startvalue Counter 0
start1 = 65535                 'Startvalue Counter 1
controlword0 = 54              'Controlword bộ đếm 0 (Xem text)
controlword1 = 116             'Controlword bộ đếm 1 (Xem text)
'
'-----Tính toán các byte High và Low-----
'
start0.highbyte = start0 \ 256
start0.lowbyte = start0 MOD 256 'Chỉ có phần dư
```



```
start1.highbyte = start1 \ 256
start1.lowbyte = start1 MOD 256 'Chỉ có phần dư
```

-----Cấu hình bộ đếm-----

```
OUT contrreg, controlword 0 'Bộ đếm 0
OUT counter0, start0.lowbyte 'Trước hết Lowbyte
OUT counter0, start0.highbyte 'Sau đến Highbyte
OUT contrreg, controlword 1 'Bộ đếm 1
OUT counter1, start1.lowbyte 'Trước hết Lowbyte
OUT counter1, start1.highbyte 'Sau đến Highbyte
```

DO

-----Nhớ trung gian trạng thái bộ đếm-----

```
OUT contrreg, 64 'Chốt bộ đếm 1
```

-----Đọc kết quả từ bộ đếm-----

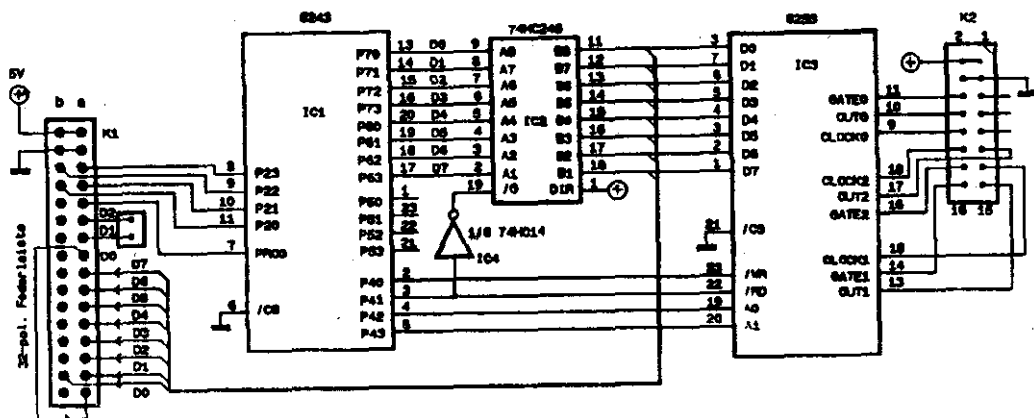
```
stand1.lowbyte = INP(counter1)
                        'Trước hết các byte Low
stand1.highbyte = INP(counter1)
                        'Sau là các byte High
counterstand = stand1.lowbyte + 256 * stand1.highbyte
```

```
PRINT "Kết quả đếm hiện thời của bộ đếm 1 là:";
counterstand
IF INKEY$ = CHR$(27) THEN EXIT DO
LOOP
END
```

2.6.3 MÔĐUN ĐẾM Ở CỔNG NỐI TIẾP VÀ CỔNG SONG SONG

Phần cứng

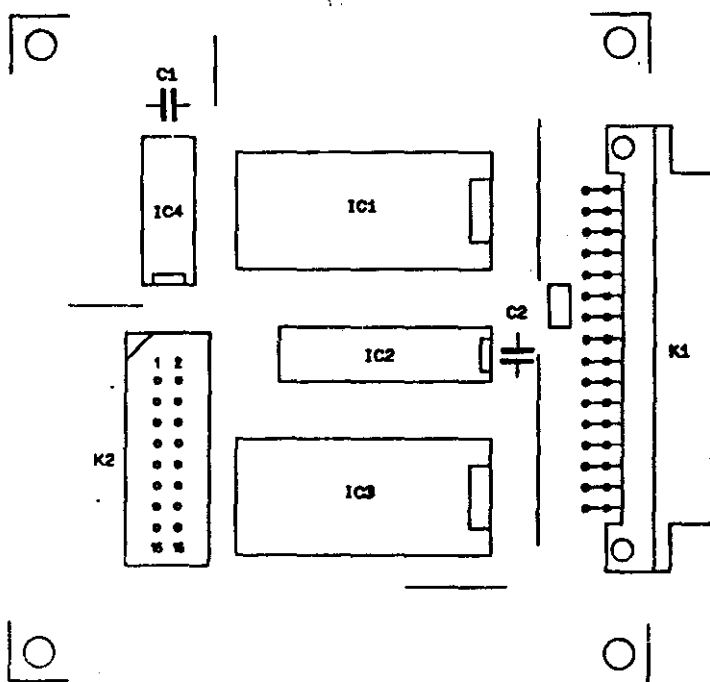
Như đã thấy rõ trong mục vừa rồi, vi mạch đếm 8253 được lắp ráp trên card cắm thêm vào có thể được điều khiển rất đơn giản. Từ đó ta cũng thấy rằng rãnh cắm của máy tính PC bên cạnh 8 đường dẫn dữ liệu hai hướng cũng còn có những tín hiệu điều khiển cũng như tín hiệu địa chỉ. Cuộc trao đổi thông tin chỉ có thể xảy ra khi mà có được truy nhập được lên các tín hiệu này. Qua các môđun đã mô tả ở chương 1, vi mạch đếm 8253 có thể được điều khiển trực tiếp, bởi vì các môđun này có chỉ một vài đường dẫn.



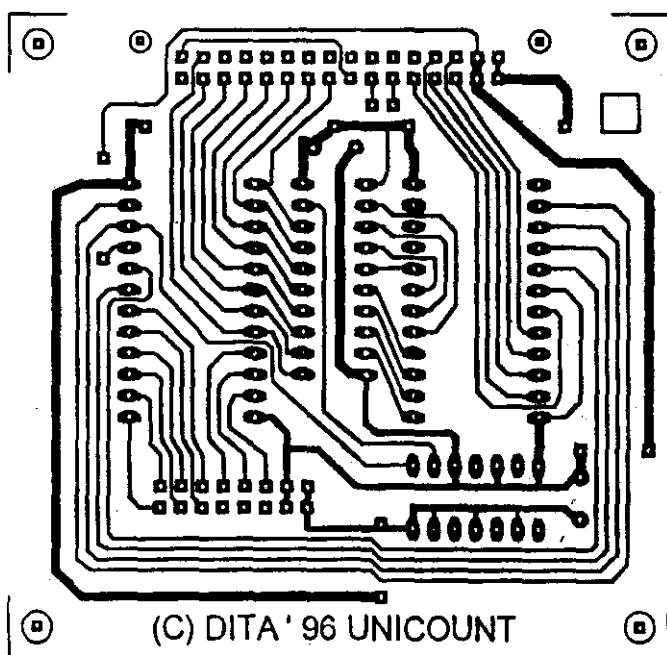
Hình 2.27 Môđun đếm vạn năng.

Tuy nhiên qua việc sử dụng một vi mạch 8243 (xem thêm mục 2.4) vấn đề sẽ không có gì cần quan tâm thêm. Hình 2.27 chỉ ra sơ đồ mạch điện của môđun đếm sử dụng vi mạch 8253, môđun này có thể được ghép nối với bất kỳ một môđun nào trong chương 1. Theo cách này, tùy theo cách lựa chọn mà vi mạch 8253 có thể được điều khiển qua cổng máy in hoặc cổng nối tiếp RS 232. Trong phạm vi chương này, ta sẽ còn có điều kiện để tìm hiểu thêm một vài ứng dụng đáng quan tâm khác nữa. Trước hết ở đây ta xem xét mạch điện mô tả trên hình 2.27.

Vi mạch đếm 8243 tạo ra khả năng điều khiển tất cả các đường dẫn dữ liệu cũng như các đường dẫn địa chỉ của vi mạch 8253. Giữa



Hình 2.28 Sắp xếp linh kiện trên môđun đếm.



Hình 2.29 Sơ đồ mạch in của môđun đếm.

các đường dẫn dữ liệu của hai vi mạch còn có một bộ đệm bus hai hướng 74HC245, vi mạch này sẽ chuyển sang trạng thái điện trở cao mỗi khi truy nhập lên vi mạch đếm 8253 để đọc ($/RD = 0$). Trong mỗi chu trình ghi, các dữ liệu của vi mạch đếm 8253 được dẫn đến ổ nối 32 chân để rồi được đọc từ máy tính PC qua các môđun bus. Ở ổ nối 16 chân K2 các tín hiệu của ba bộ đếm có thể được điều dẫn. Ổ nối này đóng vai trò như một cổng ghép nối đối với các phần cứng riêng biệt. Hình 2.28 mô tả sơ đồ sắp xếp linh kiện của môđun đếm; còn trên hình 2.29 là sơ đồ mạch in của môđun này.

Danh sách các linh kiện dưới đây giới thiệu các vật tư cần thiết để lắp ráp một môđun đếm vạn năng:

Linh kiện bán dẫn: IC1: 82C43

Phích cắm:

IC2: 74HC245

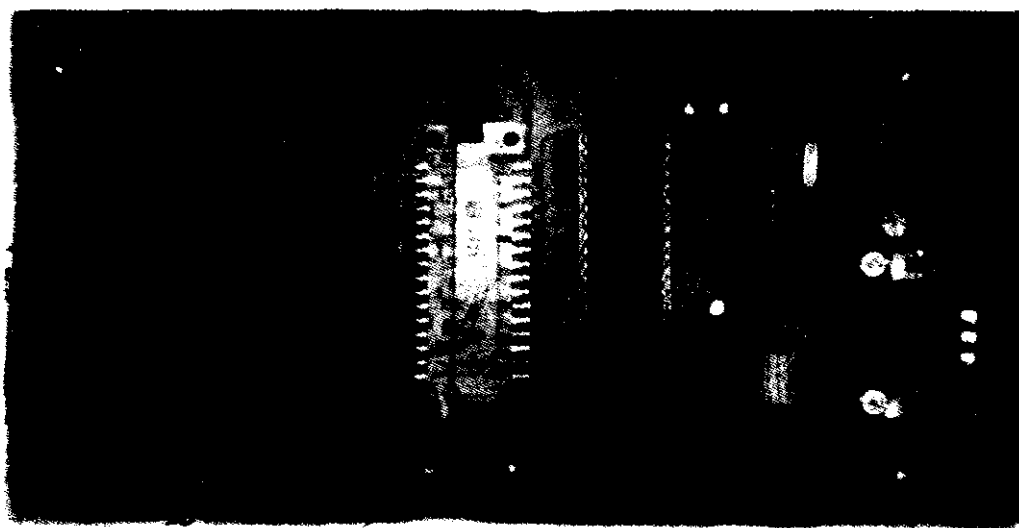
K1: dẹt 32 chân

IC3: 82C 53

K2: 25 chân

IC4: 74HC14

Trên hình 2.30 là hình dáng bên ngoài của môđun đếm ghép nối với một môđun cơ sở đã trình bày ở chương 1.



Hình 2.30 Môđun đếm ghép nối với một môđun cơ sở.

Phần mềm

Phần mềm để điều khiển ba bộ đếm của vi mạch 8253 cùng với môđun đếm được viết ra rất công phu chứ không đơn giản như khi ghép nối bộ đếm vào rãnh cắm của máy tính PC. Những gì mà ở card cắm thêm vào có thể gây tác động chỉ bằng một lệnh, thì ở đây lại cần phải lập trình theo từng bước một. Vì vậy chương trình dùng làm thí dụ dưới đây nên được tìm hiểu kỹ lưỡng. Trong chương trình ta sẽ thấy rõ cả ba bộ đếm được nạp bằng giá trị xuất phát trong các kiểu hoạt động khác nhau, và chỉ ra hiện mức của bộ đếm trên màn hình. Toàn bộ chương trình là một tín hiệu hoàn chỉnh nên ở đây không cần có thêm những lời giải thích.

```

=====
Program: UNICTLPT
Function: Bằng chương trình này có thể trao đổi
với hai bộ đếm lùi 16 bit qua cổng nối với máy
in. Từ đó dẫn đến những ứng dụng như là bộ đếm
xung, bộ đếm tần số, bộ phát xung lập trình được,
đồng hồ ngắt mạch lập trình được v. v...
Hardware: Cần có một môđun cơ sở đã trình bày ở
chương 1 cũng như một môđun đếm vạn năng
=====
DECLARE SUB OUT.PORT4.8243 (nibbel)
DECLARE SUB BYTE.TO.8253 (BYTE)
DECLARE SUB WRITE.DATA.TO.8253 (Counternumber, MODE,
startvalue)
DECLARE SUB READ.DATA.FROM.8253 (Counternumber,
Counterstand)
DECLARE SUB lese.lpt (inbyte)

```

```

DIM SHARED outbyte, datreg, statreg, contrreg

```

```

-----Khởi tạo-----

```

```

basadr = &h378
datreg  = basadr          'Datenregister
statreg  = basadr + 1     'Statusregister
contrreg = basadr + 2     'Controlregister

```

```

outbyte = 255

```

```

COLOR 0, 15

```

```

CLS

```

```

a$ = "T E S T S O F T W A R E FOR COUNTER MODUL"

```

```

a = LEN (a$): b = (80 - a) / 2 - 1

```

```

LOCATE 1, b: PRINT CHR$ (201); STRING$

```

```

(a + 2, CHR$ (205)); CHR$ (187)

```

```

LOCATE 2, b: PRINT CHR$ (186);

```

```

LOCATE 2, b + 2: PRINT a$

```

```

LOCATE 2, a + b + 3: PRINT CHR$ (186)

```

```

LOCATE 3, b: PRINT CHR$ (200); STRING$

```

```

(a + 2, CHR$ (205)); CHR$ (188)

```

```

PRINT

```

```

'---Ở đây mode và giá trị xuất phát có thể được cho vào---

```

```

mod0 = 2          'Kiểu bộ đếm 0

```

```

mod1 = 2          'Kiểu bộ đếm 1

```

```

mod2 = 2          'Kiểu bộ đếm 2

```

```

start0 = 20000    'Giá trị xuấtphát ở bộ đếm 0

```

```

start1 = 45000    'Giá trị xuấtphát ở bộ đếm 1

```

```

start2 = 30000    'Giá trị xuấtphát ở bộ đếm 2

```

```

'-----Vòng lặp đọc vào-----

```

```

CALL WRITE.DATA.TO.8253 (0, mod0, start0)

```

```

CALL WRITE.DATA.TO.8253 (1, mod1, start1)

```

```

CALL WRITE.DATA.TO.8253 (2, mod2, start2)

```

'-- --Sinh ra xung Low bằng đường dẫn dữ liệu D2-----
' (có thể được dùng cho các ứng dụng của 8253)
,

outbyte = outbyte AND (255 - 2)

OUT datreg, outbyte

outbyte = outbyte OR 2

OUT datreg, outbyte
,

'-----Vòng lặp đọc vào-----
,

DO

CALL READ.DATA.FROM.8253 (0, stand0)

CALL READ.DATA.FROM.8253 (1, stand1)

CALL READ.DATA.FROM.8253 (2, stand2)

Impulse& = (start0 - stand0) + start0 * (start1 -
stand1)
,

'-----Xuất dữ liệu ra màn hình-----
,

LOCATE 5, 20: PRINT "COUNTER 0"

LOCATE 5, 40: PRINT "COUNTER 1"

LOCATE 5, 60: PRINT "COUNTER 2"

LOCATE 7, 1: PRINT "MODE:"

LOCATE 7, 20: PRINT mod0

LOCATE 7, 40: PRINT mod1

LOCATE 7, 60: PRINT mod2

LOCATE 9, 1: PRINT "GIA_TRI_XUAT_PHAT:"

LOCATE 9, 20: PRINT start0

LOCATE 9, 40: PRINT start1

LOCATE 9, 60: PRINT start2

LOCATE 11, 1: PRINT "HIEN_MUC_O_BO_DEM:"

LOCATE 11, 20: PRINT stand0

LOCATE 11, 40: PRINT stand1

LOCATE 11, 60: PRINT stand2

```

LOCATE 15, 1: PRINT "TINH_TOAN:"
LOCATE 15, 20: PRINT Impulse &
LOCATE 20, 1: PRINT "ESC: Ngừng lại"
'
'-----Khả năng ngừng bằng phím ESC-----
'
IF INKEY$ = CHR$ (27) THEN EXIT DO

LOOP
CLOSE 1
END
SUB BYTE.TO.8253 (BYTE)
'=====
' Unterprogram: byte.to.8253
' Function: Chương trình này điều khiển đường dẫn
' của 8243 sao cho một byte dữ liệu được xuất ra qua
' các cổng 6 và 7. Byte này được dẫn tới các đường
' dẫn dữ liệu D0 đến D7 của vi mạch đếm 8253.
'=====
nibbel.low = BYTE AND &HF
nibbel.high = (BYTE AND &HFO) / 16
'-----Xuất ra NIBBEL.LOW qua cổng 7-----
Port = 7
'
'-----Xuất ra từ điều khiển-----
'
outbyte = (outbyte AND (&HF)) + port * 16
OUT datreg, outbyte
'
' ---Lập trình từ điều khiển: Prog từ High sang Low---
'
outbyte = outbyte AND (255 - 8)
OUT datreg, outbyte
'
'-----Xếp nửa byte dữ liệu (4 BIT) được cho ra -----

```



```

outbyte = (outbyte AND &HF) + nibbel.low * 16
OUT datreg, outbyte
'
'-----PROG trở lại HIGH-----
'
outbyte = outbyte OR 8
OUT datreg, outbyte
'
'-----Xuất ra NIBBEL.High qua Port 6-----
'
Port = 6
'
'-----Xuất ra từ điều khiển-----
'
outbyte = (outbyte AND (&HF)) + Port * 16
OUT datreg, outbyte
'
'----Lập trình CONTROLWORD: PROG từ HIGH sang LOW----
'
outbyte = outbyte AND (255 - 8)
OUT datreg, outbyte
'
' ----- Xếp nửa byte dữ liệu (4 BIT) được cho ra -----
outbyte = (outbyte AND &HF) + nibbel.high * 16
OUT datreg, outbyte
'
'-----PROG trở lại HIGH-----
'
outbyte = outbyte OR 8
OUT datreg, outbyte
END SUB
SUB lese.lpt (inbyte)
'
'=====
'  Unterprogram: lese.lpt

```

' Function: Chương trình con này đọc vào một byte
' qua cổng máy in. Kết quả đúng ở trong biến inbyte.

'=====

'-----Đọc dữ liệu vào-----

```
OUT contrreg, 0          'init = 0
inbyte1 = INP(statreg)    'đọc vào từ D0,D1,D2,và D3
OUT contrreg, 4          'init = 1
inbyte2 = INP(statreg)    'đọc vào từ D4,D5,D6,và D7
```

'-----Sắp xếp các bit dữ liệu đọc vào-----

```
'inbyte1:      Statusregister bit 4 (SLCT) là D0
'              Statusregister bit 5 (PE)      là D1
'              Statusregister bit 6 (ACK)      là D2
'              Statusregister bit 5 (BUSY) là /D3
'inbyte2:      Statusregister bit 4 (SLCT) là D4
'              Statusregister bit 7 (PE)      là D5
'              Statusregister bit 6 (ACK)      là D6
'              Statusregister bit 5 (BUSY)      là /D7
inbyte = (((inbyte1 XOR 128) AND &HF0) / 16) +
((inbyte2 XOR 128) AND &HF0)
END SUB
```

SUB OUT.PORT4.8243 (nibbel)

'=====

' Unterprogram: out.port.8243
' Function: Chương trình này điều khiển các đường
' dẫn của vi mạch 8243 sao cho một bit dữ liệu được
' xuất ra qua cổng 4. Bit này điều khiển các đường
' dẫn điều khiển của vi mạch 8253 như sau:

```
P43 --> A1 P42 --> A0
P41 --> /RD P40 --> /WR
```

```
'=====
'-----Xuất nửa byte qua cổng 4-----
Port = 4
'
'-----Xuất ra từ điều khiển-----
'
outbyte = (outbyte AND (&HF)) + Port * 16
OUT datreg, outbyte
'
'---Lập trình từ điều khiển: PROG từ HIGH sang LOW---
'
outbyte = outbyte AND (255 - 8)
OUT datreg, outbyte
'
'----- Xếp nửa byte dữ liệu (4 BIT) được cho ra-----
outbyte = (outbyte AND &HF) + nibbel * 16
OUT datreg, outbyte
'
'-----PROG trở lại HIGH-----
'
outbyte = outbyte OR 8
OUT datreg, outbyte
END SUB
SUB READ.DATA.FROM.8253 (Counternummer, Counterstand)

'=====
'  Unterprogram: READ.DATA.FROM.8243
'  Function: Chương trình điều khiển bộ đếm 8253 qua
'  8243 sao cho trạng thái đếm được thông báo lại
'  như là kết quả
'=====
'---Địa chỉ (A1 = 1 và A0 = 1) xếp cho CONTROLWORD---
' D6 --> A0 and D7 --> A1
' đồng thời /RD và /WR phải được đặt lên High
'
```

```

nibbel = 15
CALL OUT.PORT4.8243 (nibbel)
'
'-----Tạo nên từ điều khiển CONTROLWORD-----
'
' 64 tương ứng với: Sự chốt dữ liệu
controlword.8253 = 64 * Counternummer
'
'-----Gửi CONTROLWORD tới 8253 -----
'
CALL BYTE.TO.8253 (controlword.8253)
'
'-----Tạo ra WRITE-SIGNAL /WR: LOW...HIGH----
'
nibbel = nibbel AND &HE
CALL OUT.PORT4.8243 (nibbel)
nibbel = nibbel OR &H1
CALL OUT.PORT4.8243 (nibbel)
'
'-----Xếp địa chỉ cho số bộ đếm-----
'
' D6 tương ứng với bit địa chỉ A0
' D7 tương ứng với bit địa chỉ A1
nibbel = (nibbel AND &H3) + 4 * Counternummer
CALL OUT.PORT4.8243 (nibbel)
'
'-----Tạo ra READ-SIGNAL /RD: LOW...HIGH-----
'
nibbel = nibbel AND &HD
CALL OUT.PORT4.8243 (nibbel)
'
'-----Đọc dữ liệu vào-----
'
CALL lese.lpt (byte.low)

```

```

nibbel = nibbel OR &H2      'Read-Signal trở lại High
CALL OUT.PORT4.8243 (nibbel)
'
'-----Tạo ra READ-SIGNAL /RD: LOW...HIGH -----
'
nibbel = nibbel AND &HD
CALL OUT.PORT4.8243 (nibbel)
'
'-----Đọc dữ liệu vào-----
'
CALL lese.lpt (byte.high)
nibbel = nibbel OR &H2      'Read-Signal trở lại High
CALL OUT.PORT4.8243 (nibbel)
'
Counterstand = (byte.low + 256 * byte.high)

END SUB

SUB WRITE.DATA.TO.8253(Counternummer,MODE, startvalue)
'
'=====
'  Unterprogram: WRITE.DATA.TO.8243
'  Function:Chương trình này điều khiển vi mạch đếm
'  8253 qua vi mạch 8243 sao cho số đếm, kiểu đếm và
'  giá trị xuất phát được gửi tới 8253
'=====
'
'---Xếp các địa chỉ A1=1 và A0=2) cho từ điều khiển ---
'              (từ điều khiển: Controlword)
'  D6 tương ứng với bit địa chỉ A0
'  D7 tương ứng với bit địa chỉ A1
'  nibbel = 15
'  CALL OUT.PORT4.8243 (nibbel)
'
'-----Tạo ra từ điều khiển-----

```

48 tương ứng với: trước hết LSB sau đến MSB

0 tương ứng với: các số nhị phân

controlword.8253 = 64 * Counternummer+48+2 * MODE + 0

'-----Gửi từ điều khiển tới 8253-----'

CALL BYTE.TO.8253 (controlword.8253)

'-----Tạo ra tín hiệu viết /WR: LOW...HIGH-----'

nibble = nibble AND &HE

CALL OUT.PORT4.8243 (nibbel)

nibble = nibble OR &H1

CALL OUT.PORT4.8243 (nibbel)

'-----Xếp các địa chỉ dành cho số bộ đếm-----'

(Số bộ đếm: Counternummer)

' D6 tương ứng với bit địa chỉ A0

' D7 tương ứng với bit địa chỉ A1

nibbel = (nibbel AND 3) + 4 * Counternummer

CALL OUT.PORT4.8243 (nibbel)

'-----Gửi từ dữ liệu (16 bit) tới 8253 HIGHBYTE-----'

' Hãy chú ý tới sự phân chia ra HIGHBYTE & LOWBYTE

low.byte = startvalue MOD 256

high.byte = INT (startvalue / 256)

'-----Gửi đi LOWBYTE-----'

CALL BYTE.TO.8253 (low.byte)

'-----Tạo ra tín hiệu viết /WR: LOW...HIGH-----'

nibble = nibble AND &HE

```
CALL OUT.PORT4.8243 (nibbel) .
nibble = nibble OR &H1
CALL OUT.PORT4.8243 (nibbel)
```

-----Gửi đi HIGHBYTE-----

```
CALL BYTE.TO.8253 (high.byte)
```

-----Tạo ra tín hiệu viết /WR: LOW...HIGH-----

```
nibble = nibble AND &HE
CALL OUT.PORT4.8243 (nibbel)
nibble = nibble OR &H1
CALL OUT.PORT4.8243 (nibbel)
```

END SUB

PHAN MEM KIEM TRA DUNG CHO MODUN DEM

	Bo_Dem 0	Bo_Dem 1	Bo_Dem 2
Mode:	3	1	2
Gia_tri_xuat_phat	65000	32500	3000
Trang_thai_dem	36469	55035	1150
Tinh_toan	28531		

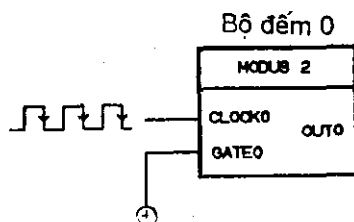
Huong_dan: ESC : Ngung

Hình 2.31 Trình bày trên màn hình của chương trình UNICTLPT.BAS.

Hình 2.31 giới thiệu sự trình bày màn hình sau khi khởi động chương trình.

2.6.4 NHỮNG ỨNG DỤNG CỦA VI MẠCH ĐẾM 8253

2.6.4.1 Bộ đếm xung 16 bit



Hình 2.32 Bộ đếm xung 16 bit.

Ứng dụng đầu tiên của bộ đếm 8253 là để tạo ra một bộ đếm xung đơn giản. Bộ đếm này đếm các sườn tín hiệu chạy đến ở lối vào CLOCK. Tùy theo kiểu được sử dụng, tần số cực đại của tín hiệu lối vào có thể đạt đến 10 MHz. Bởi vì vi mạch 8253 có chứa bộ đếm 16 bit, nên bằng một bộ đếm chỉ có thể đếm đến 65.535. Một bộ đếm như vậy làm việc theo kiểu hoạt động 2 và có thể mô tả trên hình 2.32. Lối vào GATE được đặt lên mức High. Tín hiệu kẻ sát lối vào CLOCK làm giảm dần số các xung chờ đếm bằng sườn âm. Dư đó hoàn toàn có ý nghĩa khi mà ta chọn luôn giá trị lớn nhất có thể 65.535 như là giá trị xuất phát. Số xung có thể được tính như sau:

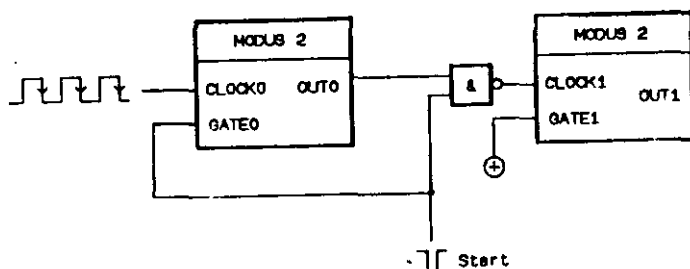
$$\text{Số xung} = \text{Giá trị xuất phát} - \text{Hiện mức ở bộ đếm}$$

Chú ý: Sau khi nạp bộ đếm bằng giá trị xuất phát, sườn âm đầu tiên ở lối vào còn chưa tác động lên quá trình đếm mà chỉ dẫn đến một điều là giá trị bắt đầu được chấp nhận như là hiện mức ở bộ đếm. Xung đầu tiên được đặt như là xung 0.

Sự lập trình để vi mạch hoạt động như là một bộ đếm 16 bit được thực hiện từng bước một. Trước hết từ điều khiển, sau đây là giá trị xuất phát cần phải được ghi vào bộ đếm. Sau sự lưu trữ trung gian của hiện mức của bộ đếm, bộ đếm này có thể được đọc ra một cách yên lặng.

2.6.4.2 Bộ đếm xung 32 bit

Bằng việc nối thêm một bộ đếm 16 bit thứ hai, ta nhận được một bộ đếm 32 bit, để có thể đếm đến 4,29 tỷ xung. Hình 2.33 chỉ ra sơ đồ mạch điện của hai bộ đếm. Cả hai làm việc theo kiểu làm việc 2. Khi quá thang bộ đếm 0 (sự chuyển Low/ High của tín hiệu lối ra OUT) bộ đếm 1 nhận được một xung đếm qua cổng NAND.



Hình 2.33 Bộ đếm xung 32 bit.

Phép đo được bắt đầu bằng một xung khởi động. Bằng xung khởi động, bộ đếm 1 nhận được ở lối vào CLOCK một sườn tín hiệu âm, sườn này chấp nhận giá trị xuất phát như là hiện mức của bộ đếm hiện tại. Bây giờ thì bộ đếm hoàn toàn được thiết lập xong giá trị ban đầu. Để tính các xung chồng chất lên ta có thể dùng công thức sau:

$$\text{Số xung} = (\text{start0} - \text{stand0}) + \text{start0} \cdot (\text{start1} - \text{stand1})$$

Chỉ dẫn: Sau đây là ...

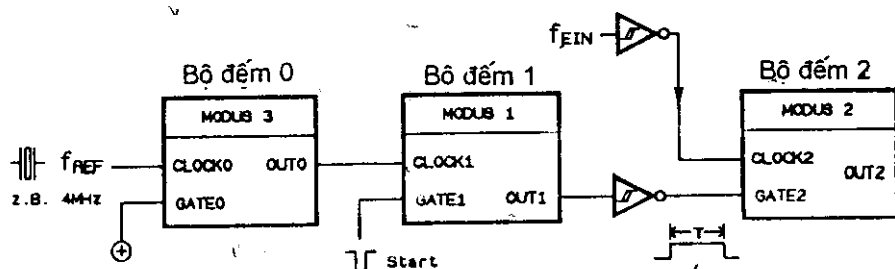
startX giá trị xuất phát của bộ đếm X và

standX hiện mức ở bộ đếm X

2.6.4.3 Bộ đếm tần số 16 bit

Phương pháp được dùng để đo tần số có thể nói ngắn gọn là đếm số xung tín hiệu lối vào trong một thời gian mở cổng được xác định chính xác. Thời gian mở cổng có thể được thực hiện theo kiểu hoạt động 1. Ở đây tín hiệu lối ra giữ nguyên ở mức Low đối với n chu kỳ giữ nhịp ($n =$ giá trị bắt đầu). Khi ta dẫn tín hiệu này qua một bộ đảo đến lối vào

GATE của một bộ đếm khác nữa hoạt động theo kiểu hoạt động 2 thì bộ đếm này sẽ đếm các xung lỗi vào trong thời gian T. Hình 2.34 chỉ ra mạch điện của vi mạch 8253 dùng cho một bộ đếm tần số 16 bit.



Hình 2.34 Bộ đếm tần số 16 bit.

Bộ đếm 0 chia tần số so sánh từ một bộ phát dao động dùng bộ cộng hưởng thạch anh có tần số 4 MHz thành các giá trị nhỏ hơn. Bằng bộ đếm 1 có thể tạo ra những thời gian mở cổng khác nhau. Ngoài ra tất nhiên còn cần có một xung khởi động ở lỗi vào GATE, xung này mở đầu thời gian mở cổng bằng sườn dương. Sau thời gian mở cổng hiện mức ở bộ đếm của bộ đếm không thay đổi nữa. Sau đó hiện mức ở bộ đếm có thể được đọc ra và tần số được tính ra theo công thức sau:

$$f_{in} = \frac{(start2 - stand2) \cdot f_{ref}}{start0 \cdot start1}$$

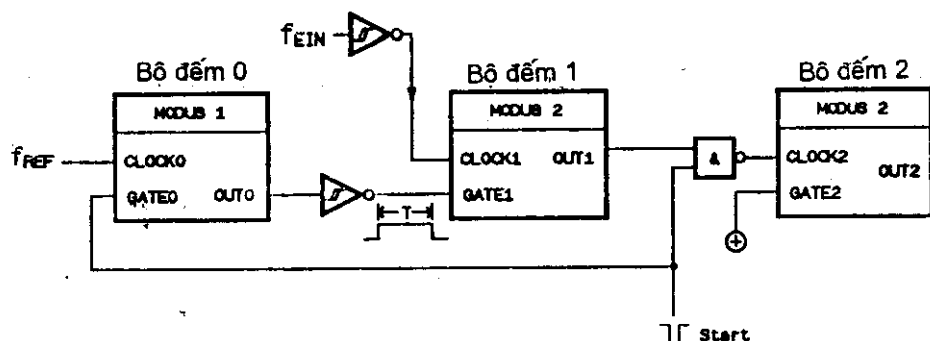
Thí dụ bằng số: $f_{ref} = 4 \text{ MHz}$

start0	start1	t đếm	start2	stand2	f_{in}
4000	1.000	1 s	65.535	10.000	55.535
40	1.000	0,1 s	60.000	20.000	400,00 kHz
4	1.000	0,01 s	65.000	15.000	5 MHz
4000	1.0000	10 s	65.535	5.535	6000,0 Hz

2.6.4.4 Bộ đếm tần số 7 chữ số

Bộ đếm tần số được mô tả dưới đây có thể dùng để đo tần số đến 10 MHz. Các tần số cao hơn có thể đo được bằng cách cho qua bộ chia tần

số thích hợp. Nhưng bộ hiển thị bị hạn chế chỉ đến 7 chữ số. Mạch điện trên hình 2.35 về cơ bản gồm hai phần. Trước hết, một phép đo tần số đòi hỏi có một khoảng thời gian mở cổng được xác định rất chính xác. Khoảng thời gian này đạt được nhờ bộ đếm 0 theo kiểu hoạt động 1. Mặt khác cần có một bộ đếm có thể đếm được ít nhất 10 triệu xung trong một giây ($f_{in} = 10 \text{ MHz}$).



Hình 2.35 Bộ đếm tần số 7 chữ số.

Ngoài ra còn có hai bộ đếm 16 bit. Để bắt đầu phép đo cần có một xung Low, xung này bắt đầu khoảng thời gian mở cổng qua lối vào GATE của bộ đếm 0. Sau đó xung này giữ nhịp bộ đếm 2 qua cổng NAND, qua đó bộ đếm này chấp nhận giá trị xuất phát như là hiện mức của bộ đếm. Tần số lối vào được dẫn đến lối vào giữ nhịp của bộ đếm 1. Cứ mỗi lần quá thang bộ đếm này tạo ra một sự chuyển mức Low sang High, sự chuyển mức này lại giữ nhịp bộ đếm 2 qua cửa NAND. Nói khác đi, bộ đếm 2 đếm tín hiệu quá thang của bộ đếm 1. Sau khoảng thời gian mở cổng hiện mức đếm của bộ đếm 1 và bộ đếm 2 không thay đổi nữa. Sau đấy, tần số của tín hiệu lối vào có thể được thông báo qua công thức tính sau:

$$f_{in} = \frac{(start1 - stand1) + start1 \cdot (start2 - stand2) \cdot f_{ref}}{start0}$$

2.6.4.5 Bộ đếm tần số thấp với độ chính xác cao

Bộ đếm tần số vừa trình bày có nhược điểm nằm chính trong phương pháp đo tần số. Bộ đếm tần số đã đếm số xung trong khoảng thời gian mở cổng đã được xác định rất chính xác. Chẳng hạn như khoảng thời gian mở cổng này là 1 giây, thì số xung đếm được tương ứng với tần số của tín hiệu lối vào. Nếu như ta chấp nhận sự đếm xung có sai số là ± 1 xung thì khi thời gian mở cổng là 1 giây sai số này tương ứng với một độ chính xác của phép đo tần số là ± 1 Hz. Nói khác đi, một tần số có giá trị $f = 100$ Hz được đo với độ chính xác là ± 1 Hz thì sai số tương đối là 1 %. Một tần số có giá trị 10 Hz chỉ có thể đo được với độ chính xác 10 %. Có một khả năng loại bỏ được nhược điểm này là kéo dài khoảng thời gian mở cổng. Nhưng điều này cũng có nghĩa là ta phải chờ đợi thêm 10 giây hoặc là 100 giây trước khi ta nhận được kết quả. Một giải pháp khác, nhưng đồng thời cũng tạo ra một phương pháp đo khác là đo thời gian kéo dài của chu kỳ. Trong phương pháp đo này, đối với khoảng thời gian của một chu kỳ giữ nhịp của tín hiệu lối vào, các xung được đếm với một tần số cao. Để lĩnh hội và sử dụng thí dụ cuối cùng một lần nữa ta đếm ở một tần số lối vào có giá trị 10 Hz trong một khoảng thời gian kéo dài của chu kỳ ($T = 0,1$ s). Các xung với tần số là 4 MHz, thì ta sẽ nhận được kết quả đếm là 400.000. Từ đó, tần số của tín hiệu lối vào có thể được tính như sau:

$$f_{in} = f_{ref} / \text{Kết quả đếm} = 10,0000 \text{ Hz.}$$

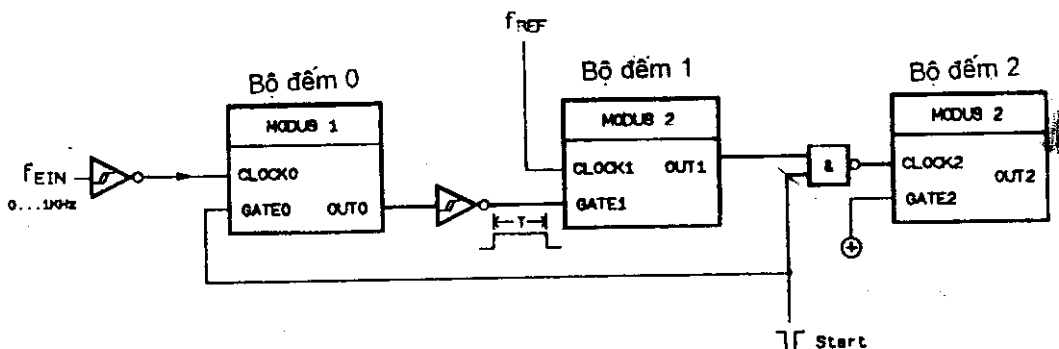
Độ không chính xác của kết quả đo bây giờ chỉ còn bằng $\pm 0,000025$ Hz. Khi ta cho phép có một sai số ± 1 xung khi đếm xung thì ta có thể tính độ không chính xác của phép đo độ kéo dài của chu kỳ như sau:

$$\text{Độ không chính xác} = f_{ref} / (\text{Kết quả đếm})^2$$

Các con số thí dụ:

f_{ref}	Kết quả đếm	f_{in}	Độ không chính xác
4 MHz	500	8.000 Hz	16 Hz
4 MHz	5.000	800 Hz	0,16 Hz
4 MHz	50.000	80 Hz	0,0016 Hz
4 MHz	500.000	8 Hz	0,000016 Hz

Hình 2.36 chỉ ra sơ đồ mạch điện của bộ đếm tần số theo cách đo độ kéo dài của chu kỳ. Nếu để ý kỹ hơn ta sẽ thấy mạch này trên thực tế là giống với bộ đếm tần số đã trình bày ở mục 2.6.4.4; chỉ khác mỗi một điểm là các chân ra f_{in} và f_{ref} đã trao đổi cho nhau.



Hình 2.36 Bộ đếm tần số thấp (< 1 kHz).

Sau thời gian mở cổng T ta có thể tính tần số ở lỗi vào theo công thức sau:

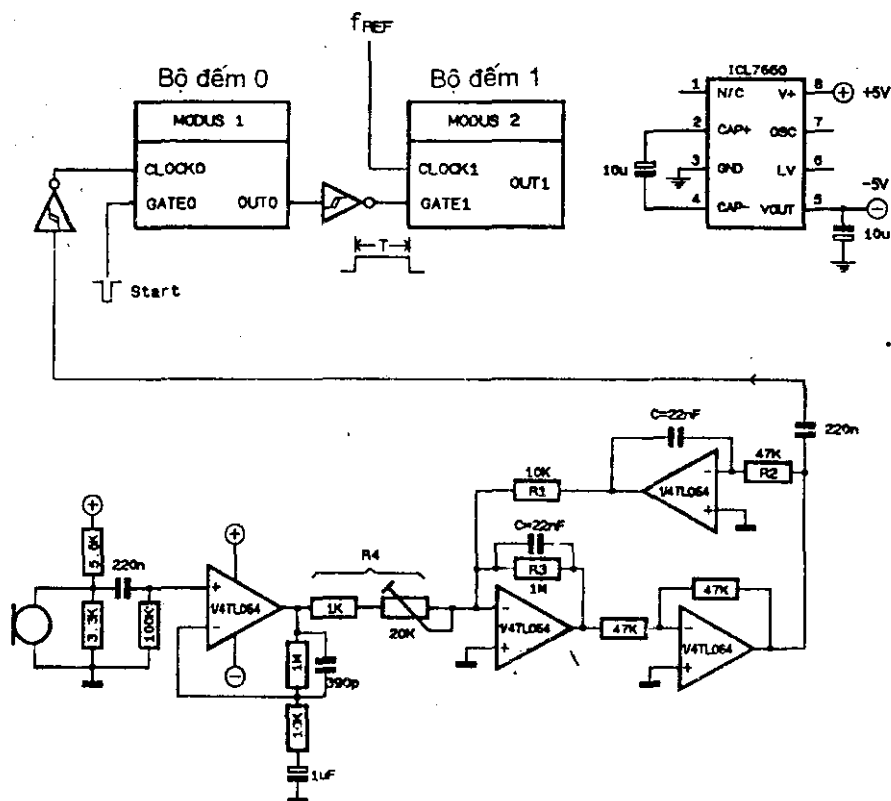
$$f_{in} = \frac{start0 \cdot f_{ref}}{(start1 - stand1) + start1 \cdot (start2 - stand2)}$$

Khi tiến hành lập trình, có một điểm mà ta phải chú ý là do sơ ý có thể làm xuất hiện lỗi chia cho số 0; đây là một lỗi nặng và trong mọi trường hợp ta không được mắc phải. Ngoài ra còn có một điểm nên cân nhắc là điều gì sẽ xảy ra sau khi bộ đếm đã được đặt cấu hình đúng và xung khởi động đã cho bắt đầu phép đo. Sườn âm (đốc xuống) đầu tiên của xung giữ nhịp tín hiệu lỗi vào cần phải đo sẽ đặt lỗi ra OUT0 lên mức Low, qua đó GATE 1 nhận được mức High và quá trình đếm bắt đầu. Sườn âm đầu tiên này trong mọi trường hợp đều phải được chờ đợi, trước khi công thức tính f_{in} được sử dụng đến. Trước đây, hiện mức ở bộ đếm không thay đổi được khi mà số 0 được tạo ra ở mẫu số.

2.6.4.6 Bộ so dây đàn ghita

Phương pháp đo tần số được giải thích trong mục vừa qua có thể sử dụng để lên dây cho thật chính xác các đàn dây. Để dùng như một dụng cụ so dây thì mạch điện này chỉ thiếu phần xử lý tín hiệu. Hình 2.37 chỉ

ra mạch điện của một máy so dây đàn ghita khi sử dụng hai bộ đếm loại 8253. Bộ đếm 0 tạo ra khoảng thời gian mở cổng T, để bộ đếm có thể đếm các xung của tần số so sánh trong khoảng thời gian này. Bởi vì, khoảng thời gian mở cổng T là rất ngắn nên theo cách này chỉ có thể thực hiện 10 đến 100 phép đo mỗi giây.



Hình 2.37 Mạch điện của thiết bị so dây đàn ghi ta dùng vi mạch 8253.

Sự xử lý tín hiệu được tiến hành bằng một tầng tiền khuếch đại và một bộ lọc dải thông bậc bốn (*biquadratic*). Bộ lọc có nhiệm vụ lọc ra tần số đúng từ tổng của các họa âm và đồng thời khuếch đại tín hiệu. Tín hiệu từ micro được dẫn đến tầng khuếch đại thuật toán đầu tiên loại TL 064, ở đây tầng này hoạt động như một tầng khuếch đại điện áp xoay chiều với hệ số khuếch đại cỡ 100. Tín hiệu lối ra được dẫn qua bộ lọc dải thông bậc bốn được lắp ráp từ ba phần còn lại của vi mạch khuếch đại thuật toán kép 4 cùng với các điện trở và tụ điện. Hệ số khuếch đại

ở tần số trung bình có thể được thiết lập chỉ qua sự thay đổi điện trở R_4 .

Để tính toán các giá trị điện trở và tụ điện có thể sử dụng các công thức sau:

$$\text{Tần số trung bình:} \quad f_0 = \frac{1}{2 \cdot \pi \cdot C \cdot \sqrt{R_1 \cdot R_2}}$$

$$\text{Hệ số khuếch đại ở } f_0: \quad v = R_3 / R_4$$

$$\text{Độ rộng dải:} \quad B = \frac{1}{2 \cdot \pi \cdot C \cdot R_3}$$

$$\text{Hệ số phẩm chất:} \quad Q = R_3 / R_1$$

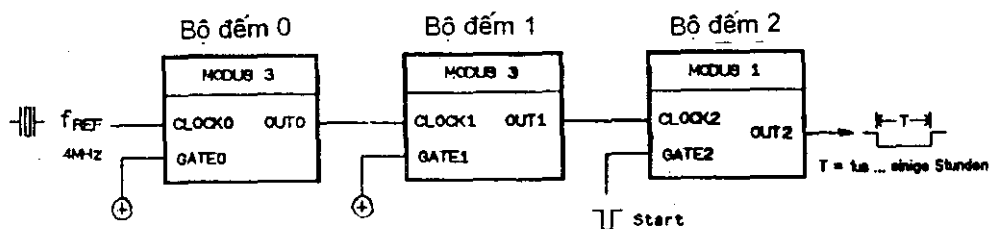
Với các linh kiện được lựa chọn trên hình 2.37 tần số trung bình đạt được là 333 Hz và một độ rộng dải là 7 Hz. Bằng cách thiết lập này có thể lên dây cho các dây đàn E (Mi), H (Si) và e (Mi) trong số tổng cộng 6 dây đàn ghita (E: Mi; A: La; D: Rê; G: Son; H: Si và e: Mi), không cần phải chọn trước một chuyển mạch điện cho các điện trở xác định tần số R_1 và R_2 . Từ đó ta thấy rằng ở dây đàn trầm E ($f = 82,4$ Hz) hoạ âm thứ tư được lọc ra ($f = 4 \times 82,4$ Hz = 329,6 Hz). Hoàn toàn tương tự với dây đàn H ($f = 110$ Hz). Ở đây, hoạ âm thứ ba của dây đàn ($f = 330$ Hz) cho đi qua được nhờ việc sử dụng của bộ lọc dải thông, tất cả các tần số khác sẽ bị hãm lại nó nằm xa tần số f_0 . Dây đàn e mảnh tạo ra tần số tương ứng là 329,6 Hz. Tín hiệu này cũng được dẫn đến được lọc, được khuếch đại ở lối vào của bộ đếm tần số. Tần số của một dây đàn ghita có thể được tính toán như sau:

$$f = \frac{f_{ref} * start0}{(start1 - stand1)}$$

2.6.4.7 Bộ phát xung lập trình được

Với 3 bộ đếm trong vi mạch 8253 có thể tạo ra độ rộng xung rất chính xác cỡ một vài micro giây đến một vài giờ. Hình 2.38 chỉ ra cách nối mạch của ba bộ đếm. Bộ đếm 0 và bộ đếm 1 hoạt động như bộ phát xung vuông góc với hệ số chia đặt được, còn bộ đếm 2 như là một mạch phát xung một trạng thái cân bằng (monoflop) lập trình được. Nguyên tắc hoạt động có thể mô tả rất đơn giản. Sau khi xung khởi động đến GATE 2, bộ đếm 2 tạo ra một tín hiệu Low ở lối ra OUT2 bằng sườn âm

đầu tiên ở CLOCK 2 đối với độ kéo dài là n chu kỳ giữ nhịp (n = giá trị xuất phát của bộ đếm 2).



Hình 2.38 Bộ phát xung lập trình được.

Độ rộng xung T có thể được tính như sau:

$$T = \frac{start0.start1.start2}{f_{ref}}$$

Thí dụ bằng số: $f_{ref} = 4 \text{ MHz}$

start 0	start 1	start 2	T
2	2	10	10 μs
4	100	10.000	1 s
4	600	60.000	1 min

Trong các ứng dụng thực tế cần phải chú ý là chỉ có xung khởi động không thôi thì không đủ để bắt đầu xung có độ rộng T . Ngoài cái đó ra sườn âm của tín hiệu ở CLOCK2 dẫn đến lỗi ra OUT2 nhận mức Low.

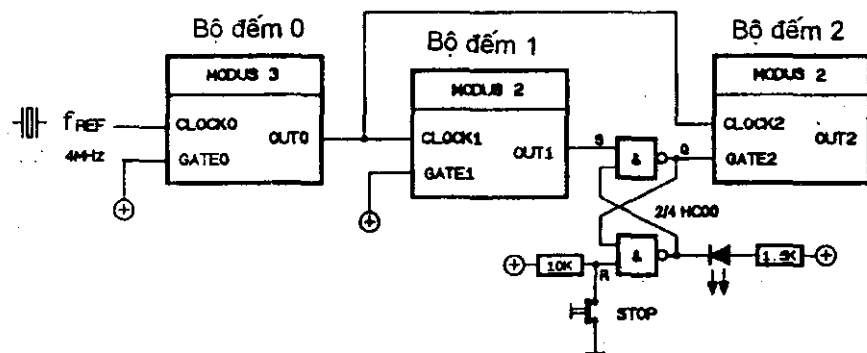
2.6.4.8 Đồng hồ role lập trình được

Đồng hồ lập trình được và có thêm chức năng đóng/ ngắt được mô tả trên hình 2.39 có khả năng thiết lập khoảng đặt thời gian từ vài micro giây đến nửa năm với độ chính xác thạch anh. Khi một môđun cơ sở từ chương 1 được sử dụng cùng với môđun đếm đã mô tả ở mục 2.6 sau khi lập trình cho vi mạch 8253 phần cứng có thể tách rời khỏi máy tính PC. Việc xử lý tín hiệu diễn ra hoàn toàn ở trên phần cứng.

Phần chính của mạch là bộ đếm 1 và bộ đếm 2, ở đây các bộ đếm này làm việc theo kiểu hoạt động 2 và ở hiện mức bộ đếm 1 chúng

2.6.4.9 Thiết bị kiểm tra phản ứng

Hình 2.40 chỉ ra sơ đồ mạch của vi mạch 8253 được sử dụng trong thiết bị thử phản ứng của con người. Bài toán được đặt ra là sau khi điốt phát quang (LED) bùng sáng thì phím phải được ấn. Thời gian phản ứng sau đó có thể được máy tính PC tính và chỉ thị ra trên màn hình. Bộ đếm 0 ở đây hoạt động như một bộ chia lập trình được có giá trị xuất phát là 4.000.



Hình 2.40 Thiết bị thử phản ứng dùng vi mạch 8253.

Do vậy mà tần số lỗi vào ở CLOCK 1 và CLOCK 2 là 1.000 Hz. Lỗi ra OUT 1 của bộ đếm 1 sau n chu kỳ giữ nhịp chuyển sang mức Low và lật trạng thái của mạch RS-Flipflop. Lỗi ra Q trở nên High, đồng thời bộ đếm 2 đếm lùi các xung lỗi vào ở CLOCK 2 bắt đầu từ giá trị xuất phát. Đồng thời lỗi ra /Q của bộ RS-Flipflop chuyển sang Low và vì thế LED bùng sáng. Sự xác nhận của phím nhấn dẫn đến tác động là lỗi ra Q của bộ Flipflop được đặt trở lại và bộ đếm 2 không đếm tiếp nữa. LED sẽ sáng trở lại. Sau đó thời gian phản ứng có thể được thông báo theo công thức sau:

$$f_{reaction} = \frac{start2 - stand2}{f_{ref} \cdot start0}$$

Giá trị xuất phát của bộ đếm 1 không có mặt trong biểu thức tính toán thời gian phản ứng. Nó có thể được sử dụng như là mạch trễ. Theo cách

$$T = \frac{\text{Số xung}}{f_{\text{ref}}} = \frac{\text{start0} \cdot (\text{start1} - \text{stand1})}{f_{\text{ref}}}$$

và nhờ vậy, giá trị của C_x cũng được tính:

$$C_x = \frac{\text{start0} \cdot (\text{start1} - \text{stand1})}{1,1 \cdot R \cdot f_{\text{ref}}}$$

Thí dụ bằng số: $f_{\text{ref}} = 4 \text{ MHz}$, $R = 1 \text{ M}\Omega$.

start 0	start 1	stand 1	C_x .
2	65.000	64.378	283 pF
2	65.000	64.870	59 pF
2	65.000	3.250	28 nF
20	65.000	8.910	255 nF
200	65.000	15.000	2,27 μ F

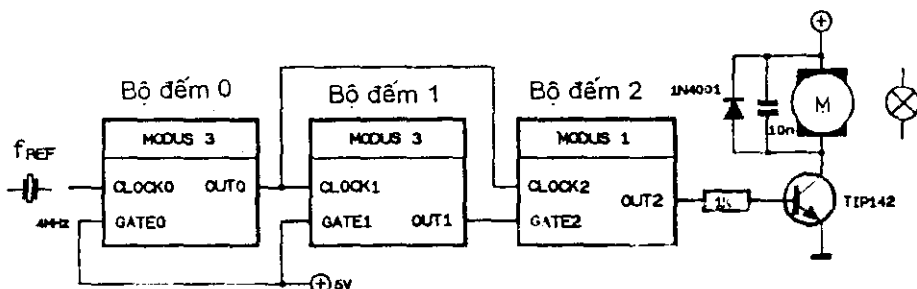
2.6.4.11 Điều khiển vòng quay

Một ứng dụng khác của vi mạch đếm 8253 cũng rất hấp dẫn là dùng để điều khiển vòng quay của motor một chiều. Ứng dụng này dựa trên quá trình điều biến độ rộng xung. Khi đó một tải sử dụng được đấu vào chỉ trong một thời gian ngắn. Bằng việc thay đổi tỷ số Xung/ Nghỉ ta thay đổi được thời gian đóng mạch tính theo phần trăm và do vậy thay đổi được giá trị công suất thực tế của động cơ một chiều. Theo cách này ta cũng có thể điều khiển không chỉ động cơ mà cả các đèn chiếu sáng hoặc lò đốt nóng.

Mạch điện của bộ biến điệu độ rộng xung được chỉ ra trên hình 2.42. Ta nhận thấy là bên cạnh ba bộ đếm của vi mạch 8253 chỉ cần có thêm một tầng đệm dưới dạng một tranzito Darlington. Khi có mức High ở lối ra OUT2 thì chuyển sang trạng thái dẫn.

Ở đây bộ đếm 8253 được sử dụng như là một bộ biến điệu độ rộng xung. Bộ đếm 0 làm việc trong kiểu hoạt động 3 như là bộ chia tần số lập trình được, tín hiệu lối ra của bộ này được dẫn đến các tín hiệu

CLOCK của bộ đếm 1 và bộ đếm 2. Còn bộ đếm 2 đã được cấu hình như là một monoflop lập trình được. Với mỗi sườn dương ở lối vào GATE 2, giá trị xuất phát start 2 trong bộ đếm 2 được nạp và lối ra OUT 2 chuyển sang mức Low sau một khoảng thời gian bằng chu kỳ giữ nhịp của tín hiệu CLOCK 2 kể từ start 2.



Hình 2.42 Sự điều khiển vòng quay dừng vi mạch 8253.

Một thí dụ bằng số có thể làm rõ hơn nguyên tắc làm việc.

Tần số của các tín hiệu GATE 2 là 50 Hz còn tỷ số Xung / Nghỉ cần phải thay đổi được theo tỷ lệ 1: 1000. Do vậy các giá trị xuất phát được khẳng định đối với bộ đếm. Giá trị xuất phát của bộ đếm 0 là start 0 = 80, còn giá trị xuất phát của bộ đếm 1 là 1000. Giá trị xuất phát start 2 của bộ đếm 2 thay đổi được giữa 1 và 1000.

Tần số của tín hiệu CLOCK 2 tính toán ra thành $4 \text{ MHz} : 80 = 50.000 \text{ Hz}$ tương ứng với một chu kỳ bằng $20 \mu\text{s}$. Khi bộ đếm 2 được nạp bằng giá trị 15 thì sẽ xảy ra sự thay đổi dưới đây: Với sườn dương của tín hiệu 50 Hz ở lối vào GATE 2, lối ra OUT 2 sau khoảng thời gian bằng $15 \times 20 \mu\text{s} = 300 \mu\text{s}$ sẽ chuyển sang Low. Sau đấy, lối ra OUT 2 giữ nguyên ở mức High cho đến khi xuất hiện một sườn dương mới của tín hiệu 50 Hz. Độ kéo dài của mức High được tính ra thành $20 \text{ ms} - 300 \mu\text{s} = 19,7 \text{ ms}$. Với một tranzito công suất làm việc ở đây như một bộ đảo, động cơ sẽ được đóng điện trong đúng khoảng thời gian này. Khi mà một giá trị bắt đầu lớn hơn được nạp vào bộ đếm 2, thì thời gian dừng ở mức Low của tín hiệu lối ra OUT 2 sẽ bị kéo dài ra tương ứng và động cơ

nhận được một công suất nhỏ hơn và do vậy vòng quay của nó bị giảm đi.

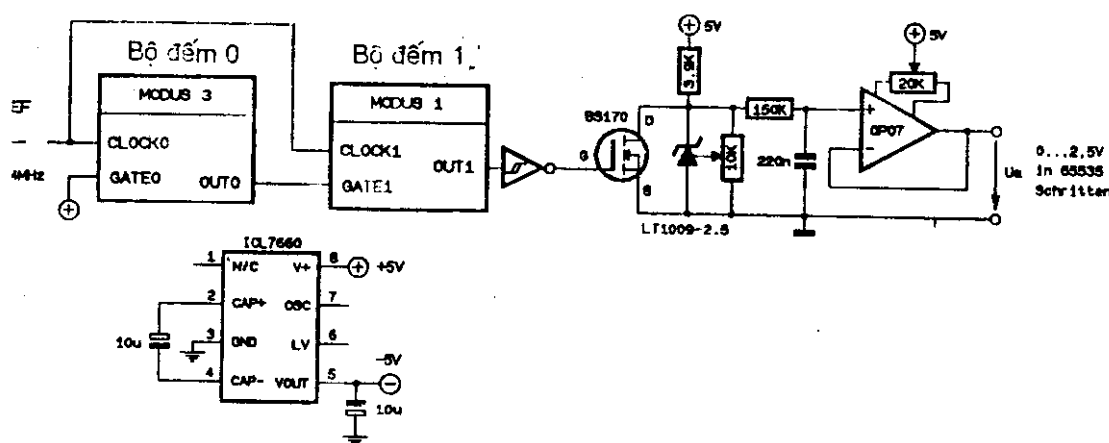
Thời gian đóng mạch T_{on} , tính theo phần trăm, có thể được tính theo công thức sau:

$$T_{on}(\%) = \frac{start1 - start2}{start1} * 100$$

Thí dụ bằng số $f_{ref} = 4 \text{ MHz}$

start 1	start 2	Độ kéo dài của thời gian đóng mạch
1000	1	99,9 %
1000	15	98,5 %
1000	250	75 %
1000	500	50 %
1000	1.000	0 %

2.6.4.12 Bộ biến đổi D/A 16 bit



Hình 2.43 Bộ biến đổi D/A 16 bit dùng vi mạch 8253.

Bằng cách điều biến độ rộng của xung để điều khiển số vòng quay như trình bày ở trong mục trên đây cũng có thể tạo nên một bộ biến đổi D/A. Để hiểu được ứng dụng dưới đây bạn đọc nên tìm hiểu kỹ cách điều khiển vòng quay trình bày ở mục 2.6.4.11.

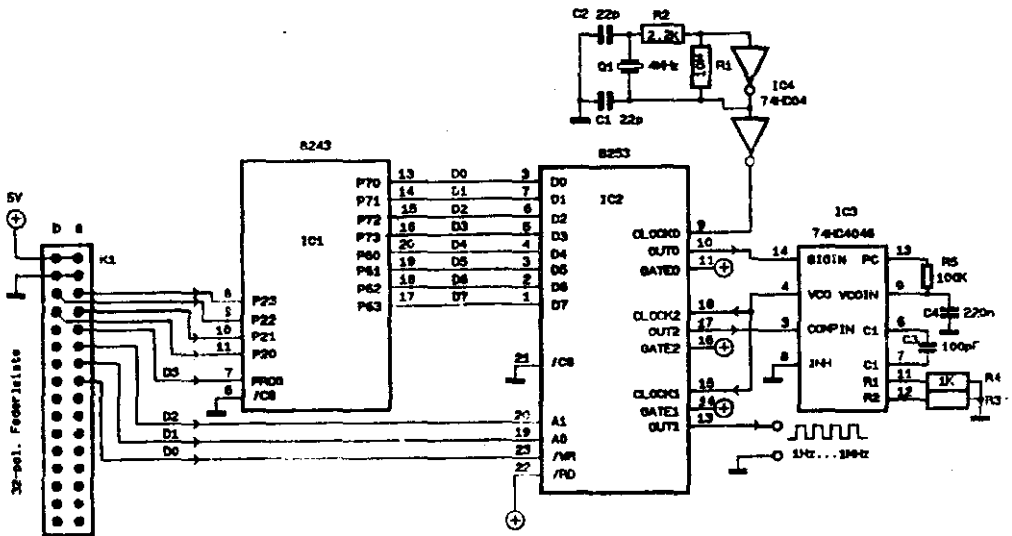
Bằng mạch điện trên hình 2.43 có thể thiết lập điện áp lối ra từ 0 cho đến 2,5 V trong 65.535 bước. Giá trị này tương ứng với một bộ biến đổi D/A 16 bit. Bước nhảy điện áp nhỏ nhất có giá trị chỉ bằng 38,15 μ V. Tín hiệu ra số OUT 1 của bộ điều biến độ rộng xung được dẫn qua một bộ đảo đến tranzito BS 170, ở đây tranzito này hoạt động như là một chuyển mạch. Khi cực của nhận được một điện áp cỡ 5 V thì tranzito trở nên dẫn, trong khi điện áp là 0 V thì tranzito ở trạng thái điện trở cao. Chuỗi xung với tỷ số *Xung/Nghỉ* thay đổi được và biên độ bằng 2,5 V được dẫn đến một mạch RC để lọc ra phần điện áp một chiều. Ở lối vào không đảo của OP 07 có một điện áp một chiều mà ta có thể tính toán theo công thức sau:

$$U_a = 2,5 \text{ V} * \frac{\text{start0} - \text{start1}}{\text{start0}}$$

Công thức trên có được từ cách lập luận như sau: Giá trị trung bình của điện áp được tính từ thương số giữa thời gian có xung và thời gian nghỉ của tín hiệu OUT 1 nhân với biên độ 2,5 V. Thời gian nghỉ được tính bằng giá trị xuất phát của bộ đếm 1. Còn chu kỳ có thể thiết lập bằng giá trị bắt đầu của bộ đếm 0. Thời gian có xung lại được tính như là hiệu số giữa chu kỳ và thời gian nghỉ.

2.6.4.13 Bộ tổng hợp tần số

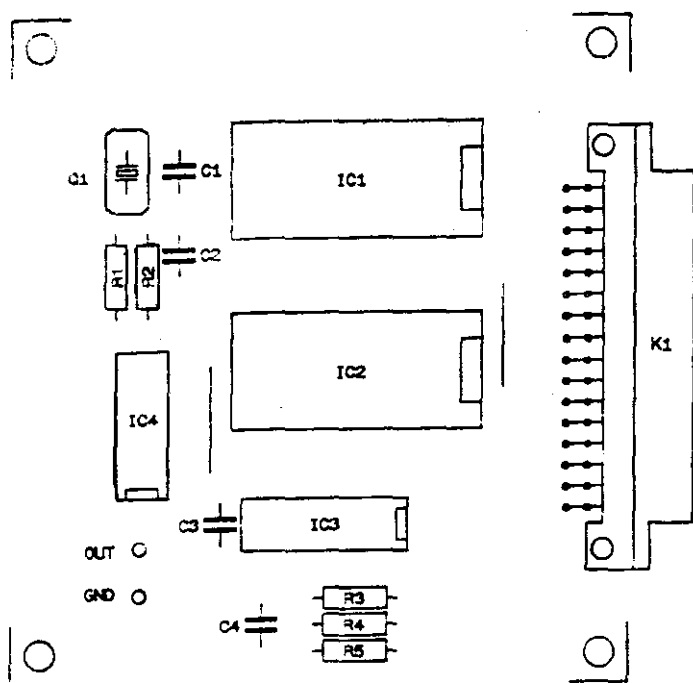
Bộ tổng hợp tần số được giới thiệu dưới đây tạo ra một tín hiệu vuông góc mà tần số của nó có thể được thiết lập một cách chính xác bằng bộ dao động thạch anh giữa 1 Hz và 1 MHz. Sự lựa chọn tần số giữa 100 Hz và 999 kHz được tiến hành một cách chính xác bằng cách lựa chọn 3 con số có nghĩa, còn giữa 1 Hz và 100 Hz với khoảng cách tần số là 1 Hz. Tất cả các tần số có thể được thiết lập một cách thuận tiện từ máy tính PC khi ghép nối với một môđun cơ sở đã mô tả ở chương 1. Trên hình 2.44 là mạch điện của môđun tổng hợp tần số.



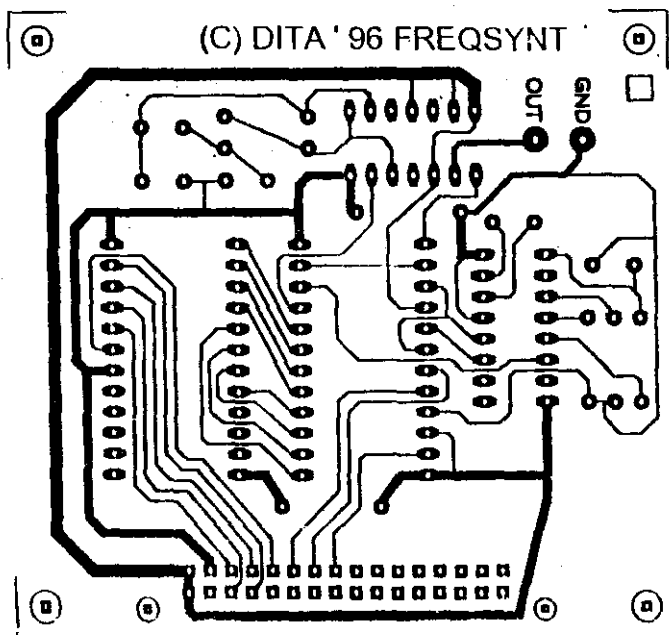
Hình 2.44 Sơ đồ mạch điện của môđun tổng hợp tần số.

Vi mạch ghép nối 8243 ở đây điều khiển 8 đường dẫn dữ liệu của vi mạch 8253. Sự trình bày chi tiết về vi mạch này có thể tìm thấy trong mục 2.4. Ba bộ đếm của vi mạch 8253 đều làm việc như là những bộ chia tần số lập trình được trong kiểu hoạt động 3. Ba bộ chia tần số được nối với vi mạch 74HC4046, vi mạch này chứa một bộ so pha và một bộ dao động điều khiển bằng điện áp (VCO). Vi mạch này cùng với bộ chia tần số tạo nên một mạch điều chỉnh pha, còn được gọi là vòng khoá pha PLL (tiếng Anh: *Phase Lock Loop*). Nhiệm vụ của vi mạch 74HC4046 là kiểm tra hai tần số theo thuộc tính bằng nhau về pha và tần số và tạo ra một tín hiệu dùng cho bộ dao động điều khiển bằng điện áp (VCO). Muốn thế, bộ so pha có hai lối vào được sắp xếp ở chân 14 và 3. Ở lối vào 1 của bộ so pha (chân 14) có tần số so sánh $f = 2 \text{ kHz}$ được tạo ra bởi bộ đếm 0. Ngoài ra, khi tần số so sánh có giá trị 4 MHz , giá trị xuất phát cần phải được lựa chọn bằng 2.000.

Lối vào 2 (chân 3) được nối với lối ra của bộ dao động điều khiển bằng điện áp qua bộ chia tần số lập trình được của bộ đếm 2. Lối ra của bộ so pha điều chỉnh bộ dao động bằng một điện áp một chiều sao cho hai tần số ở chân 14 và chân 3 đồng nhất với nhau. Bởi vì trong vòng lặp phản hồi một bộ chia tần số được đấu mạch với tỷ số chia $n : 1$ ($n = 2 \dots 1000$) nên tần số dao động ở chân 4 bằng n lần tần số so sánh ở



Hình 2.45 Sơ đồ sắp xếp linh kiện của môđun tổng hợp tần số.



Hình 2.46 Sơ đồ mạch in của môđun tổng hợp tần số.

chân 14. Nói cách khác, bộ dao động cần phải làm việc với tần số bằng n lần tần số của tín hiệu so sánh. Thí dụ như khi một tỷ số chia được thiết lập bằng 256 thì tần số dao động ở chân 4 là 512 kHz. Tín hiệu lỗi ra của bộ VCO chia thêm một lần chia tần số phụ thêm bằng bộ chia tần số lập trình được của bộ đếm 1 để đạt tới những số thập phân nhỏ hơn. Hệ số chia có thể được lựa chọn giữa 2 và 65.535.

Sơ đồ sắp xếp linh kiện của môđun tổng hợp tần số như trên hình 2.45.

Để lắp ráp mạch điện ta cần có những linh kiện sau đây:

Linh kiện bán dẫn:

IC1: 82C43

IC2: 82C53

IC3: 74HC4046

IC4: 74HC04

R3: có thể bỏ đi

R5: 100 k Ω

Tụ điện:

C1, C2: 22 pF

C3: 220 nF

Điện trở:

R1: 10 M Ω

R2, R4: 1 k Ω

Phích cắm:

K1: 32 chân

Hình 2.46 chỉ ra sơ đồ mạch in của môđun tổng hợp tần số. Nếu như bạn đọc muốn dùng sơ đồ mạch in này để tự lắp ráp môđun thì phải chú ý là dòng chữ ở phía hàn các linh kiện phải được đọc đúng chiều.

Phần mềm để điều khiển môđun tổng hợp tần số chỉ ra danh sách dưới đây của các lệnh trong QBasic. Ở đây không cần có sự mô tả chương trình bởi vì những lời giải thích trong chương trình đã là đầy đủ.

```
=====
' Program: FRESYCOM
'
```

```
' Function:Chương trình này cho phép thiết lập tần
' số từ 1 Hz đến 1 MHz qua giao diện nối tiếp, với
' độ chính xác của bộ dao động thạch anh. Từ 100 Hz
' đến 999 kHz được chia ra ba vị trí có giá trị để
' lựa chọn tần số. Các tần số được dự tính như sau:
```

```
'         100 k $\Omega$  tương ứng với 1MHz
```

```
'         234 k $\Omega$  tương ứng với 234 kHz
```

' 841 Ω tương ứng với 841 Hz
' 2,34 k Ω tương ứng với 2340 Hz
' 29 Ω tương ứng với 29 Hz
' Bằng cách thay đổi tỷ chút chương trình này cũng
' có thể dùng cho cổng máy in.
'
' Hardware: Cần có một môđun cơ sở từ chương 1 cũng
' như môđun tổng hợp tần số

'=====

DECLARE SUB BYTE.TO.8253 (BYTE)
DECLARE SUB WRITE.DATA.TO.8253 (counternummer, mode,
startvalue)

DIM SHARED outbyte
outbyte = 255

'----- 9600 baud, nối vào COM2 -----

OPEN "com2:9600,N,8,1,CS,DS" FOR RANDOM AS #1

Label1:

COLOR 0, 15

CLS

a\$ = "F R E Q U E N Z S Y N T H E S I Z E R"

a = LEN (a\$): b = (80 - a) / 2 - 1

LOCATE 1, b: PRINT CHR\$ (201); STRING\$

(a + 2, CHR\$ (205)); CHR\$ (187)

LOCATE 2, b: PRINT CHR\$ (186)

LOCATE 2, b + 2: PRINT a\$

LOCATE 2, a + b + 3: PRINT CHR\$ (186)

LOCATE 3, b: PRINT CHR\$ (200); STRING\$

(a + 2, CHR\$ (205)); CHR\$ (188)

PRINT

'-----Bắt đầu chương trình chính-----

INPUT "Hãy cho vào giá trị tần số

(thí dụ 23.6k):", freq\$

a = INSTR (freq\$, "k") 'a: ở vị trí nào là một "k"?

b = INSTR (freq\$, ".") 'b: ở vị trí nào là một "."?

IF a = 0 THEN 'không có "k"

freq = VAL (LEFT\$ (freq\$, LEN (freq\$)))

decade = 3

GOTO Label2 END IF

IF b = 0 THEN 'không có "." được cho vào, thí dụ 123K

freq = VAL (LEFT\$ (freq\$, LEN (freq\$) - 1))

decade = 0

END IF

IF b = 2 THEN '". " ở vị trí thứ ba, thí dụ 2.64K

freq = VAL (LEFT\$ (freq\$, LEN (freq\$) - 1)) * 100

decade = 2

END IF

IF b = 3 THEN '". " ở vị trí thứ ba, thí dụ 45.8K

freq = VAL (LEFT\$ (freq\$, LEN (freq\$) - 1)) * 10

decade = 1

END IF

Label2:

'-----Sự điều khiển bộ đếm 8253-----

CALL WRITE.DATA.TO.8253 (0, 3, 2000)

CALL WRITE.DATA.TO.8253 (1, 3, 2 * 10 ^ decade)

CALL WRITE.DATA.TO.8253 (2, 3, freq)

INPUT "Again"; a\$

IF a\$ = "j" THEN GOTO Label1

```
IF a$ = "n" THEN GOTO ketthuc
ketthuc:
```

```
    CLOSE 1
```

```
    END
```

```
SUB BYTE.TO.8253 (BYTE)
```

```
'=====
```

```
'  Unterprogram: byte.to.8253
```

```
'  Function: Chương trình điều khiển các đường dẫn
'  của 8243 sao cho một byte dữ liệu được xuất ra qua
'  các Port 6 và 7. Byte này được dẫn tới các đường
'  dẫn dữ liệu D0 đến D7 của bộ đếm 8253.
```

```
'=====
```

```
nibbel.low = BYTE AND &HF
```

```
nibbel.high = (BYTE AND &HF0) / 16
```

```
'-----Xuất ra của NIBBEL.LOW qua Port 7-----
Port 7
```

```
,
```

```
'-----Xuất ra CONTROLWORD-----
```

```
,
```

```
outbyte = (outbyte AND (&HF)) + Port * 16
```

```
PRINT #1, CHR$(outbyte);
```

```
,
```

```
'-Lập trình cho CONTROLWORD:PROG Từ HIGH Sang LOW-
```

```
,
```

```
outbyte = outbyte AND (255 - 8)
```

```
PRINT #1, CHR$(outbyte);
```

```
'-----Xếp các dữ liệu (4 BIT) được xuất ra-----
```

```
outbyte = (outbyte AND &HF) + nibbel.low * 16
```

```
PRINT #1, CHR$(outbyte);
```

```
,
```

```
'-----PROG trở lại HIGH-----
```

```

outbyte = outbyte OR 8
PRINT #1, CHR$ (outbyte);
'
'-----Xuất NIBBEL.HIGH qua cổng 6-----
Port = 6
'
'-----Xuất ra từ điều khiển-----
'
outbyte = (outbyte AND (&HF)) + Port * 16
PRINT #1, CHR$ (outbyte);
'
'--Lập trình cho CONTROLWORD: PROG từ HIGH sang LOW--
'
outbyte = outbyte AND (255 -8)
PRINT #1, CHR$ (outbyte);
'
'-----Xếp nửa byte dữ liệu (4 BIT) được xuất ra-----
outbyte = (outbyte AND &HF) + nibbel.high * 16
PRINT #1, CHR$ (outbyte);
'
'-----PROG trở lại HIGH-----
'
outbyte = outbyte OR 8
PRINT #1, CHR$ (outbyte);

END SUB

SUB WRITE.DATA.TO.8253 (Số đếm, Mode, Giá trị xuất
phát)
'
'-----Xếp địa chỉ dùng cho CONTROLWORD-----
'D1 tương ứng với bit địa chỉ A0
'D2 tương ứng với bit địa chỉ A1
outbyte = outbyte OR (2 + 4)

```

```

PRINT #1, CHR$(outbyte);
'
'-----Tạo nên từ điều khiển-----
'
controlword.8253 = 64*Counternumner+48+2* Mode+ 0
'
'-----Xuất CONTROLWORD cho 8253 -----
'
CALL BYTE.TO.8253 (controlword.8253)
'
'----- Tạo ra WRITE-SIGNAL /WR: LOW... HIGH-----
outbyte = outbyte AND (255 - 1)
PRINT #1, CHR$(outbyte);
. outbyte = outbyte OR 1
PRINT #1, CHR$(outbyte);
'-----Sắp xếp địa chỉ cho-----
outbyte = (outbyte AND(255-2-4)+2* Counternumner
PRINT #1, CHR$(outbyte);
'=====Gửi DATENWORD (16 BIT) tới 8253=====
' Chú ý: Sự chia cho LOWBYTE và HIGHBYTE
low.byte = startvalue MOD 256
high.byte = INT (startvalue/ 256)
'-----Gửi LOWBYTE-----
CALL BYTE.TO.8253 (low.byte)
'----- Tạo ra WRITE-SIGNAL /WR: LOW... HIGH-----
outbyte = outbyte AND (255 - 1)
PRINT #1, CHR$(outbyte);
outbyte = outbyte OR 1
PRINT #1, CHR$(outbyte);
'-----Gửi HIGHBYTE-----
CALL BYTE.TO.8253 (high.byte)
'
'----- Tạo ra WRITE-SIGNAL /WR: LOW... HIGH-----
outbyte = outbyte AND (255 - 1)
PRINT #1, CHR$(outbyte);

```

```
outbyte = outbyte OR 1  
PRINT #1, CHR$ (outbyte);  
END SUB
```

Chương 3

**CÁC MẠCH ĐIỀU KHIỂN
VỚI BỘ BIẾN ĐỔI A/D**

Các đại lượng vật lý được đo bởi các bộ cảm biến (sensor) có thể được thu thập và xử lý trong máy tính PC chỉ khi các tín hiệu tương ứng là những giá trị số (digital). Vì thế cần có bộ biến đổi A/D để số hoá các giá trị đo analog dùng cho máy tính. Chương này sẽ giới thiệu tổng quan về các bộ biến đổi A/D khác nhau và chỉ ra cho ta thấy các bộ biến đổi này được sử dụng trong thực tế như thế nào.

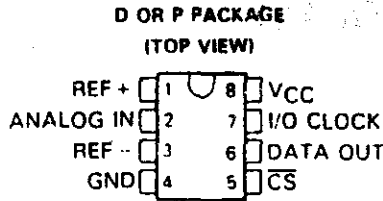
**3.1 GHÉP NỐI BỘ BIẾN ĐỔI A/D 1 KÊNH, 8 BIT
TRỰC TIẾP VÀO CỔNG NỐI TIẾP**

Bộ biến đổi A/D 8 bit có tên TLC 549 là một vi mạch rất hấp dẫn đối với những ứng dụng đơn giản. Vi mạch này chỉ có 8 chân ra và tiêu thụ dòng không đáng kể nên rất thích hợp với cổng ghép nối RS 232. Vi mạch TLC 549 có chứa: bộ giữ nhịp hệ thống, khối lấy mẫu và giữ, bộ biến đổi A/D 8 bit, một thanh ghi dữ liệu và khối logic điều khiển tương ứng trên cùng một chip. Để trao đổi thông tin với các khối chức năng khác, vi mạch này có hai chân ra với tên gọi là I/O-CLOCK và DATA OUT. Các tín hiệu ở chân này là tương thích TTL và làm dễ dàng việc trao đổi thông tin với các thành viên khác.

Các thông số kỹ thuật sau đây đặc trưng cho vi mạch TLC 549:

- ☐ Độ phân giải 8 bit.
- ☐ Có lối vào điện áp so sánh vi phân.
- ☐ Thời gian biến đổi: 17 μ s.
- ☐ Lấy mẫu và giữ (Sample & Hold) trên chip.
- ☐ Thời gian giữ nhịp hệ thống bên trong là 4 MHz.
- ☐ Điện áp nguồn nuôi: 3 ... 6 V.

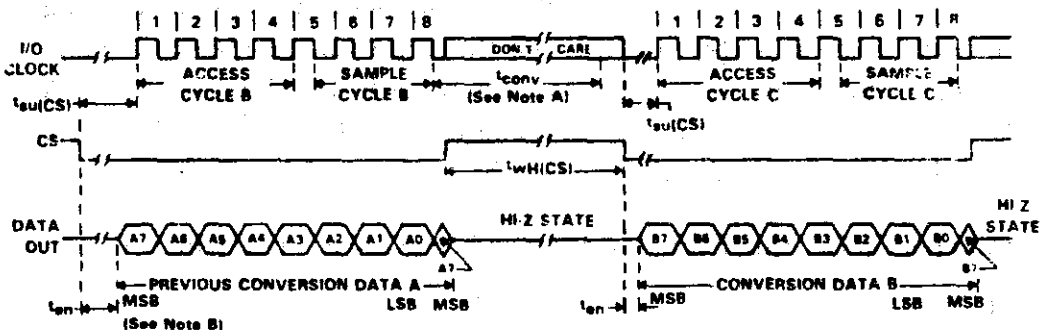
- Dòng tiêu thụ: 1,8 mA.
- Giá thành cỡ 3 USD (năm 1996).



Hình 3.1 Bố trí chân ra của vi mạch TLC 549.

Nguyên tắc hoạt động

Trong hình 3.2 là giản đồ xung có thể dùng để minh họa cho cách làm việc được mô tả dưới đây. Để có thể bắt đầu một lần biến đổi thì cần phải đặt mức Low vào lối vào Chip Select. Bây giờ sẽ thấy bit dữ liệu có giá trị cao nhất D7 của lần biến đổi diễn ra trước đây đã đứng sẵn ở lối ra DATA OUT. Bằng sườn âm của bốn nhịp đầu tiên ở I/O-CLOCK các bit dữ liệu D6, D5, D4 và D3 được dẫn đến lối ra DATA OUT. Sườn dương của xung nhịp tiếp theo ở I/O-CLOCK kích hoạt chức năng lấy mẫu và giữ (*Sample & Hold*) bên trong; tín hiệu analog đang đứng xếp hàng được dò quét và biến đổi thành một mẫu bit số (digital).



Hình 3.2 Giản đồ xung của mạch biến đổi A/D dùng vi mạch TLC 549.

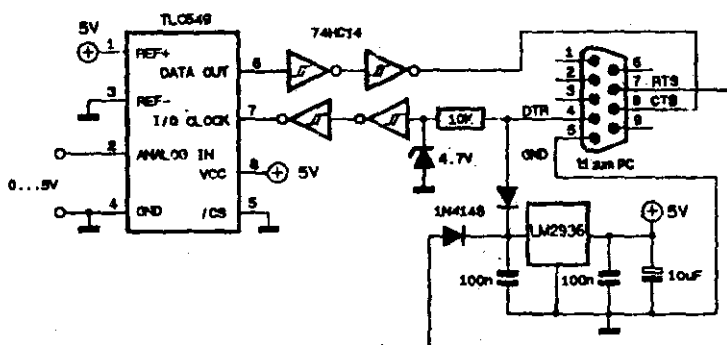
Ba xung nhịp khác nữa ở I/O-CLOCK là cần thiết có để phát ra 3 bit dữ liệu còn dư lại. Tín hiệu giữ nhịp cuối cùng (thứ tám) ở I/O-CLOCK kích hoạt chức năng giữ của khối Sample & Hold bên trong

bằng sườn âm. Trong khoảng thời gian $17 \mu s$ tiếp theo, chân I/O-CLOCK cần phải được giữ ở mức Low để bảo đảm độ chính xác của quá trình biến đổi A/D.

Khi quá trình biến đổi tiếp theo đã kết thúc, bit dữ liệu có giá trị cao nhất D7 đứng xếp hàng ở lối ra DATA OUT và có thể được đọc từ một thành viên khác. Bây giờ, với 7 xung giữ nhịp tiếp theo ở I/O-CLOCK, các bit dữ liệu còn dư lại được dẫn đến lối ra DATA OUT và bắt đầu một cuộc chơi mới.

Phần cứng

Vì mạch TLC 549 có thể đệm trực tiếp vào giao diện nối tiếp của một máy tính PC. Hình 3.3 chỉ ra sơ đồ điện của bộ biến đổi A/D 8 bit có sử dụng vi mạch TLC 549.



Hình 3.3 Bộ biến đổi A/D 8 bit đấu trực tiếp vào cổng nối tiếp của PC.

Điện áp nguồn nuôi + 5 V cung cấp từ vi mạch LM 2936; vi mạch này có dòng điện tĩnh cực kỳ nhỏ cũng như một sụt thế không đáng kể. Điện áp lối vào được lấy từ các tín hiệu RTS và DTR, chúng cho ra một điện áp + 12 V với một điện trở trong cỡ 800Ω . Ngoài ra tín hiệu RTS điều khiển chân I/O-CLOCK của vi mạch TLC 549. Các bit dữ liệu đã được biến đổi được dẫn qua lối ra DATA OUT tới lối vào CTS, mà trạng thái logic của lối vào này được máy tính PC hỏi lại.

Phần mềm

Để vi mạch TLC 549 hoạt động được với giao diện nối tiếp của máy tính PC thì trong QBasic cần có những lệnh sau đây:

```
'=====
' Program: TLC549
' Function: Chương trình này điều khiển bộ biến đổi
' A/D TLC549 qua giao diện nối tiếp của máy tính PC
' và đọc ra điện áp kế sát bộ biến đổi (so sánh với
' mạch điện ở trên hình 3.3.
'=====
COLOR 0, 15
CLS
basadr = &H2F8           'Địa chỉ cơ bản COM2
a$ = "8-Bit A/D Converter ở cổng RS 232 "
a = LEN (a$): b = (80 - a) / 2 - 1
LOCATE 1, b: PRINT CHR$ (201);STRING$ (a+2, CHR$ (205));
CHR$ (187)
LOCATE 2, b: PRINT CHR$ (186)
LOCATE 2, b + 2: PRINT a$
LOCATE 2, a + b + 2: PRINT CHR$ (186)
LOCATE 3, b: PRINT CHR$(200);
STRING$ (a + 2, CHR$ (205));
CHR$ (188)
PRINT
LOCATE 5, 20: PRINT "Điện áp là:"
LOCATE 5, 55: PRINT "Volt"
LOCATE 23, 1: PRINT "Hướng dẫn:"
LOCATE 23, 20: PRINT "ESC: Kết thúc"

UREF = 5                 'Điện áp so sánh là 5 V
'-----DTR lên +12 V, RTS lên - 12 V-----
OUT (basadr + 4), INP (basadr + 4) OR 1
OUT (basadr + 4), INP (basadr + 4) AND (255 - 2)

DO
```

```

valence = 0                                'valence: Giá trị
'-----Đọc A7-----
CTS = (INP (basard+6) AND 16) / 16          'A7 nằm ở CTS
valence = CTS * 2 ^ 7
'-----Đón nhận kết quả-----
FOR i = 6 TO 0 STEP - 1
OUT (basard + 4), INP (basard + 4) OR 2     'RTS = 1
OUT (basadr+4), INP (basadr+4) AND (255-2)   'RTS = 0
CTS = (INP (basadr + 6) AND 16) / 16
valence = CTS * 2 ^ i + valence
NEXT i
OUT (basard + 4), INP (basard + 4) OR 2     'RTS = 1
OUT (basadr + 4), INP (basadr + 4) AND (255- 2) 'RTS = 0
'
'-----Bây giờ diễn ra sự biến đổi-----
'
'-----In ra kết quả của lần biến đổi trước-----
'
volt = valence * UREF / 256
LOCATE 5, 45: PRINT USING "#.###"; volt

IF INKEY$ = CHR$(27) THEN EXIT DO
                                'Làm ngừng bằng phím ESC

LOOP
END

```

3.2 BỘ BIẾN ĐỔI A/D 8 BIT 1 KÊNH VỚI ADC 0804

3.2.1 CÁC THÔNG SỐ KỸ THUẬT VÀ CÁCH LÀM VIỆC

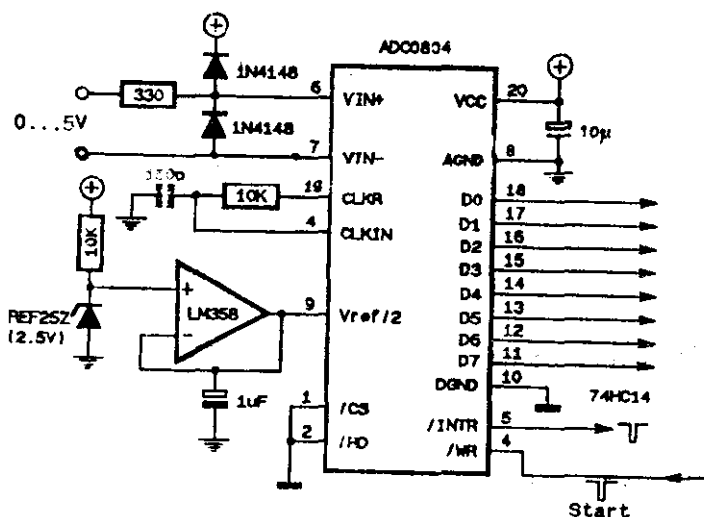
Vì mạch ADC 0804 của hãng NSC là một bộ biến đổi A/D tác động nhanh và giá không cao nên có thể được dùng đến trong nhiều ứng dụng. Trái ngược với TLC 549, vì mạch ADC 0804 quy định về quyền sử dụng các lối ra dữ liệu song song, các lối ra này đều tương thích TTL. Qua một lối vào điều khiển, các lối ra có thể chuyển sang trạng thái

điện trở cao, và nhờ thế mà vi mạch ADC 0804 có thể được đệm vào một bus dữ liệu. Ngoài ra nó còn quy định cả về quyền sử dụng các lối vào điều khiển để đọc và ghi. Lối vào analog là không so sánh với mass mà là lối vào vi phân, nghĩa là vi phân điện áp ở hai chân lối vào VIN+ và VIN- được biến đổi. Điều đáng đề cập đến ở đây là khi điện áp nguồn nuôi là + 5 V, dải điện áp của lối vào analog có thể đạt đến 5 V.

Các thông số kỹ thuật:

- ☐ Bus dữ liệu 8 bit.
- ☐ Có lối vào analog vi phân.
- ☐ Tất cả các tín hiệu tương thích TTL.
- ☐ Bộ phát xung nhịp cùng nằm trên chip.
- ☐ Dải tín hiệu analog lối vào 0 ... 5 V khi điện áp nguồn nuôi là 5 V.
- ☐ Không cần hiệu chỉnh điểm 0.
- ☐ Dòng tiêu thụ: cỡ 1,9 mA.

Hình 3.4 dưới đây mô tả một mạch điện điển hình dùng vi mạch ADC 0804 để tạo ra bộ biến đổi A/D 8 bit.



Hình 3.4 Bộ biến đổi A/D 8 bit dùng vi mạch ADC 0804.

Ở đây ta nên nhận rõ rằng dải điện áp lối vào analog lớn gấp hai lần điện áp so sánh. Điện áp so sánh (ở đây là 2,5 V) được dẫn qua một

bộ lập điện áp để tới bộ biến đổi A/D ADC 0804. Tầng khuếch đại thuật toán là bắt buộc phải có để phối hợp trở kháng, bởi vì điện trở lối vào ở chân 9 với giá trị $1,1\text{ k}\Omega$ là quá nhỏ.

Với điện trở $10\text{ k}\Omega$ và tụ điện 150 pF , bộ phát xung nhịp bên trong được kích hoạt ở các lối vào CLK. Tương ứng với giá trị đã cho của điện trở và tụ điện, tần số giữ nhịp bên trong cỡ 640 kHz .

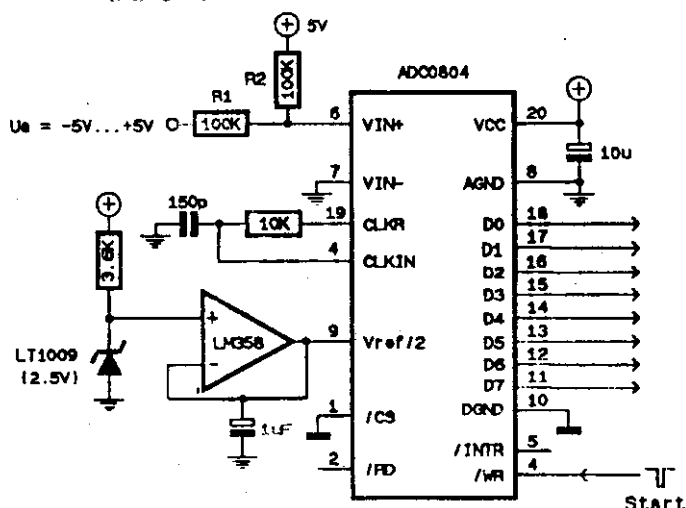
Một lần biến đổi được bắt đầu bằng một xung Low ngắn hạn ở lối vào /WR. Muốn thế điều kiện cần có là một mức Low của tín hiệu /CS. Sau thời gian biến đổi $100\text{ }\mu\text{s}$, lối ra /INTR chuyển sang Low và báo hiệu việc kết thúc quá trình biến đổi. Sau đó qua một mức Low ở lối vào /RD có thể đọc ra các bit dữ liệu. Sự truy nhập để đọc dẫn đến hậu quả là tín hiệu /INTR sẽ chuyển trở lại mức High. Khi mà, như ở trên hình 3.4, lối vào /RD được chuyển hẳn sang mức Low, thì lối ra /INTR chuyển sang Low sau quá trình biến đổi kéo dài 8 chu kỳ giữ nhịp của bộ giữ nhịp bên trong. Ở tần số giữ nhịp là 640 kHz , chu kỳ này là $12,5\text{ }\mu\text{s}$.

3.2.2 CÁCH TẠO RA CÁC DẢI ĐO KHÁC NHAU

3.2.2.1 Dải đo - 5 V...+ 5 V

Khi điện áp nguồn nuôi bằng $+5\text{ V}$ dải điện áp lối vào của vi mạch ADC 0804 là $0\ldots 5\text{ V}$. Qua việc thiết kế mạch thích hợp cho lối vào analog bằng các điện trở cũng có thể tạo ra được các dải đo khác nhau. Để có thể đo các điện áp âm thì điều kiện cần có là điện thế lối vào ở lối vào VIN+ phải được nâng lên một chút sao cho bộ biến đổi A/D nhận được giá trị 0 V như là điện áp nhỏ nhất. Hình 3.5 chỉ ra cho ta thấy dải đo - 5 V đến $+5\text{ V}$ được tạo ra như thế nào. Tín hiệu lối vào được dẫn qua điện trở R1 đến chân ra VIN+ của bộ biến đổi A/D. Đồng thời, lối vào này được nối với điện áp $+5\text{ V}$ qua điện trở R2. Điện áp cuối cùng nhận được có thể tính được nhờ nguyên lý xếp chồng điện áp. Muốn thế trước hết ta tính điện áp ở lối vào VIN+, điện áp này bắt nguồn từ tín hiệu lối vào U_e . Chân ra $+5\text{ V}$ ở điện trở R2 cần phải được đặt một cách giả định để tính toán lên điện thế 0 V . Tiếp đó ta đặt tín hiệu lối vào U_e bằng 0 V và tính điện áp ở lối vào VIN+, nếu coi như điện áp này bắt

nguồn chỉ từ điện áp + 5 V ở đầu điện trở R2. Cuối cùng bằng phép cộng ta gộp hai phần đóng góp lại.



Hình 3.5 Dải đo - 5 V ... + 5 V dùng vi mạch ADC 0804.

Các phương trình dưới đây một lần nữa tóm tắt lại phép tính:

1. Phần đóng góp nhận được từ U_e :

$$VIN+ = U_e \cdot R2 / (R1 + R2) = U_e / 2$$

2. Phần đóng góp nhận được từ + 5 V :

$$VIN+ = 5 V \cdot R1 / (R1 + R2) = 2,5 V$$

3. Tổng của các các phần đóng góp:

$$VIN+ = (U_e / 2) + 2,5 V$$

Thí dụ bằng số:

$$U_e = -5 V \quad \text{dẫn đến} \quad VIN+ = 0 V$$

$$U_e = 0 V \quad \text{dẫn đến} \quad VIN+ = 2,5 V$$

$$U_e = +5 V \quad \text{dẫn đến} \quad VIN+ = +5 V$$

3.2.2.2 Dải đo - 10 V ... + 10 V

Phép đo các điện áp âm hơn giá trị nhỏ nhất - 5 V ở trên có thể được đặt bằng sự dịch chuyển điện thế nhờ một điện trở riêng lẻ. Khi ta

muốn mở rộng dải điện áp âm thì điện áp lỗi vào phải được dẫn đến một bộ chia thế. Hình 3.6 chỉ ra sơ đồ mạch điện của lỗi vào VIN+ dùng cho dải đo - 10 V ... + 10 V. Việc tính toán điện áp, được dẫn đến bộ biến đổi A/D, có thể được thực hiện như trong thí dụ vừa qua bằng nguyên lý xếp chồng điện áp.

1. Phần đóng góp nhận được từ U_e :

$$VIN+ = U_e \cdot R2 / (R1 + R2) = U_e / 2$$

2. Phần đóng góp nhận được từ + 5 V

$$VIN+ = 5 \text{ V} \cdot R1 / (R1 + R2) = 2,5 \text{ V}$$

3. Tổng của các các phần đóng góp:

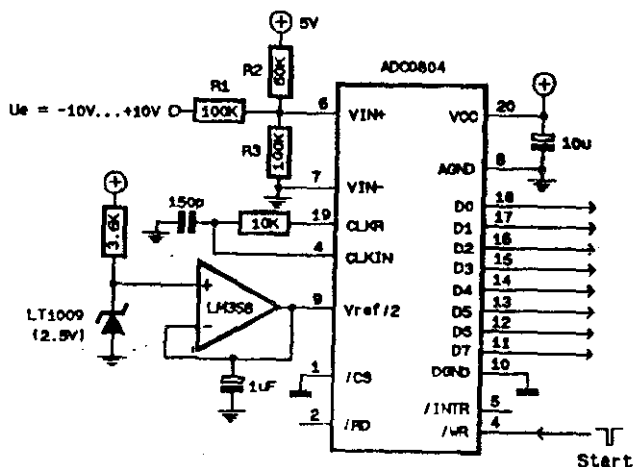
$$VIN+ = (U_e / 2) + 2,5 \text{ V}$$

Thí dụ bằng số:

$$U_e = -10 \text{ V} \text{ dẫn đến } VIN+ = 0 \text{ V}$$

$$U_e = 0 \text{ V} \text{ dẫn đến } VIN+ = 2,5 \text{ V}$$

$$U_e = +10 \text{ V} \text{ dẫn đến } VIN+ = +5 \text{ V}$$



Hình 3.6 Dải đo - 10 V ... + 10 V dùng vi mạch ADC 0804.

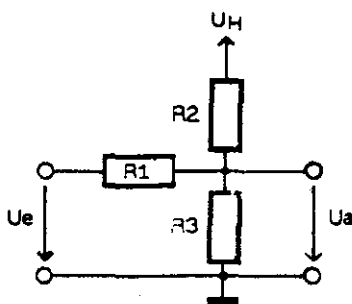
3.2.2.3 Dải đo - $U_{e, \max}$ đến + $U_{e, \max}$

Bài toán mà ta cần phải giải quyết dưới đây là phải tạo ra một dải điện áp lỗi vào nào đó từ - $U_{e, \max}$ đến + $U_{e, \max}$ trong khi điện áp lỗi

ra có thể thay đổi bất đầu từ 0 V và được phép đạt đến một giá trị cực đại $U_{a,max}$ nào đó. Bài toán này có thể giải không cần đến một tầng khuếch đại thuật toán hoặc là một điện áp âm hỗ trợ mà chỉ bằng cách đấu mạch của 3 điện trở. Việc tính toán giá trị điện trở cũng được tiến hành nhờ việc ứng dụng nguyên lý xếp chồng điện áp. Ở đây không dẫn ra toàn bộ quá trình tính toán mà chỉ đưa ra kết quả cuối cùng. Đối với mạng điện trở trên hình 3.7 có thể viết ra công thức sau đây:

$$(R_2 / R_1) = (U_H / U_{e,max})$$

$$\text{và } (R_1 / R_3) = (2 \cdot U_{e,max} / U_{a,max}) - (U_{e,max} / U_H) - 1$$



Hình 3.7 Cách đấu điện trở để dịch chuyển điện áp.

Hai thí dụ dưới đây đủ để minh họa cho việc áp dụng công thức tính toán.

Thí dụ 1: Dải điện áp lối vào là - 10 V...+ 10 V, điện áp phụ trợ có giá trị bằng $U_H = + 5$ V. Điện áp lối ra U_a của tổ hợp các điện trở trên hình 3.7 là 0 ... 5 V. Thí dụ này phù hợp đúng với số liệu áp dụng trên hình 3.6.

$$(R_2 / R_1) = (5 \text{ V} / 10 \text{ V}) = 0,5$$

$$\text{và } (R_1 / R_3) = (2 \cdot 10 \text{ V} / 5 \text{ V}) - (10 \text{ V} / 5 \text{ V}) - 1 = 1$$

Nghĩa là các điện trở R_1 và R_3 cần phải có cùng giá trị, còn R_2 cần phải nhỏ bằng một nửa giá trị của R_1 đã chọn.

Thí dụ 2 : Dải điện áp lối vào là - 25 V ... + 25 V, điện áp phụ trợ là $U_H = 2,5$ V. Điện trở R_2 sau đó có thể được nối trực tiếp với điện

áp so sánh 2,5 V. Điện áp lỗi ra U_a của mạng điện trở cần phải là 0... 2,5 V. Việc tính toán các điện trở R_1 , R_2 và R_3 được thực hiện như sau:

$$(R_2 / R_1) = (2,5 \text{ V} / 25 \text{ V}) = 0,1$$

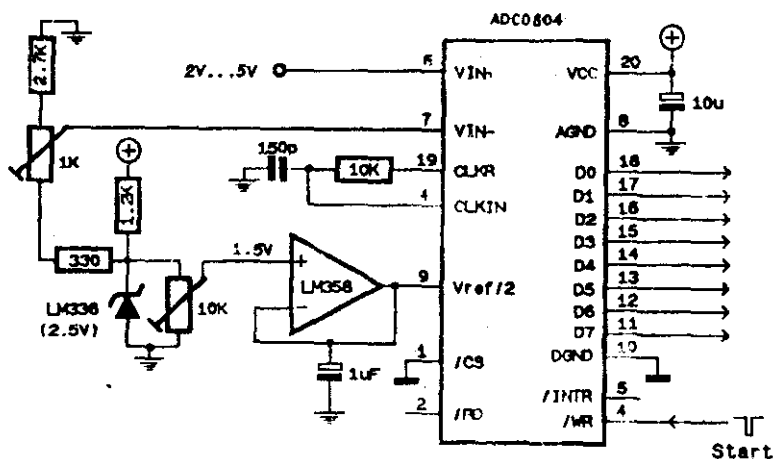
$$\text{và} \quad (R_1 / R_3) = (2 \cdot 25 \text{ V}) - (2,5 \text{ V} / 2,5 \text{ V}) - 1 = 9$$

Một sự lựa chọn khả dĩ của các điện trở có thể viết ra như sau:

$$R_1 = 180 \text{ k}\Omega; R_2 = 18 \text{ k}\Omega; R_3 = 20 \text{ k}\Omega.$$

3.2.2.4 Dải đo: 2 V... 5 V

Bộ biến đổi A/D mang tên ADC 0804 có thể dùng trong một số ứng dụng đáng quan tâm, trong đó có dùng các lối vào vi phân. Như đã giải thích, vi mạch ADC 0804 biến đổi vi phân của hai điện áp lối vào V_{IN+} và V_{IN-} thành các giá trị số. Chân V_{IN-} không bắt buộc phải được đặt lên điện thế 0 V (GND).



Hình 3.8 Dải đo 2 V ... 5 V dùng vi mạch ADC 0804.

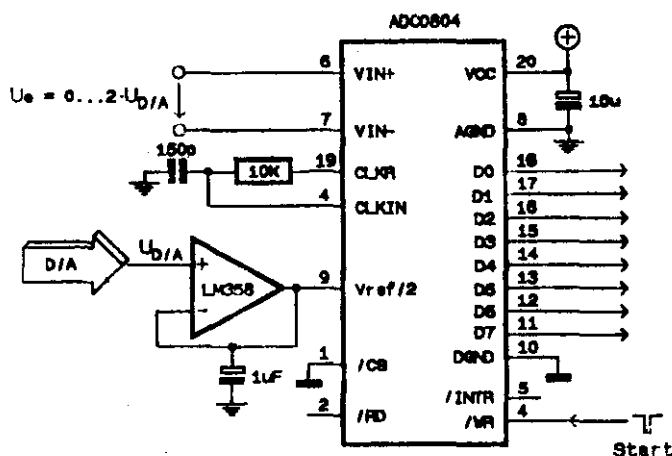
Hình 3.8 chỉ ra một mạch điện cụ thể, trong đó dải điện áp lối vào bắt đầu từ 2 V. Kết quả này có được do ta đặt lối vào V_{IN-} lên điện áp + 2 V. Ngoài ra, chân ra với điện áp so sánh (chân 9) cần phải được đặt lên 1,5 V. Sau đó điện áp lối vào ở vi mạch ADC 0804 được tính toán theo công thức:

$$U_{in} = (\text{Inputbyte} \cdot 3 \text{ V} / 256) + 2 \text{ V}$$

3.2.2.5 Dải đo lập trình được

Bởi vì điện áp so sánh ở vi mạch ADC 0804 được thiết lập từ bên ngoài nên xuất hiện khả năng đặt trước dải đo qua một bộ biến đổi A/D bằng một phần mềm (xem hình 2.9). Qua đó dải đo có thể lập trình được và thoả mãn phương trình:

$$U_{in} = 0 \dots 2 U_{D/A}$$



Hình 3.9 Dải đo chuyển đổi được dùng vi mạch ADC 0804.

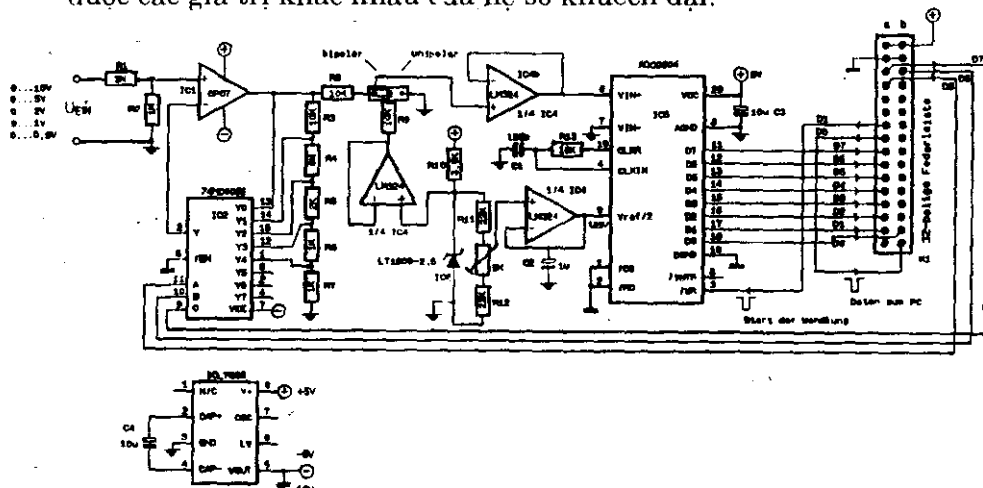
3.2.3 MÔĐUN BIẾN ĐỔI A/D 8 BIT VỚI DẢI ĐO CHUYỂN ĐỔI ĐƯỢC

Phần cứng

Khi ghép nối với môđun cơ sở đã trình bày ở chương 1, vi mạch ADC 0804 có thể được sử dụng để tạo nên một bộ biến đổi A/D 8 bit với tính năng chuyển đổi được dải đo. Bởi vì mỗi môđun cơ sở quy định quyền sử dụng ở lối vào và lối ra tương thích TTL nên sự đấu nối vào vi mạch ADC 0804 không gây nên vấn đề gì. Hình 3.10 chỉ ra sơ đồ mạch in đầy đủ của môđun biến đổi A/D.

Qua ổ nối 32 chân, môđun được nối vào một môđun cơ sở ở chương 1. Các lối ra dữ liệu D7 đến D0 được dẫn đến lối vào của môđun cơ sở qua tiếp điểm ở ổ nối. Khi mà một môđun cơ sở dùng với giao diện nối tiếp (so sánh với mục 3.1) thì cần có thêm một xung Low, xung này mở đầu sự gửi đi các dữ liệu tới máy tính PC. Xung Low này được tạo ra nhờ đường dẫn dữ liệu lối ra D0 của môđun cơ sở.

Các đường dẫn dữ liệu còn tự do là D7, D6 và D5 được dẫn đến bộ quét động (dồn kênh) 8 của 74HC4051. Vi mạch này cùng với bộ khuếch đại thuật toán OP07 và các điện trở phản hồi R3 - R7 tạo nên một tầng khuếch đại lập trình được. Tùy thuộc vào việc điện trở phản hồi ở lối vào không đảo của vi mạch OP07 được nối vào vị trí nào mà ta có thể nhận được các giá trị khác nhau của hệ số khuếch đại.



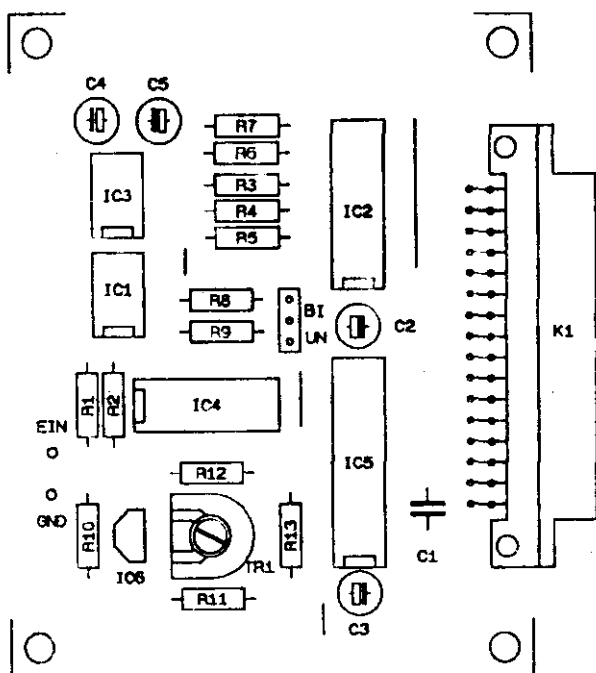
Hình 3.10 Môđun biến đổi A/D 8 bit với dải đo chuyển đổi được.

Bảng sau đây minh họa rõ thêm:

C	B	A	Liên kết	Hệ số khuếch đại
0	0	0	Y - Y0	1
0	0	0	Y - Y1	$1 + R3 / (R4 + R5 + R6 + R7) = 2$
0	0	0	Y - Y2	$1 + (R3 + R4) / (R5 + R6 + R7) = 5$
0	0	0	Y - Y3	$1 + (R3 + R4 + R5) / (R6 + R7) = 2$
1	1	1	Y - Y4	$1 + (R3 + R4 + R5 + R6) / R7 = 20$

Hệ số khuếch đại được lựa chọn sao cho trong từng dải đo và khi điện áp lối vào nhận giá trị cực đại thì điện áp lối ra của tầng khuếch đại OP07 đạt giá trị 2,5 V. Điện áp lối ra của vi mạch OP07 được dẫn qua điện trở R8 đến tầng lặp lại điện áp LM324 và sau đó đến lối vào của bộ biến đổi A/D. Với các cầu nối (Jumper) ở hai điện trở R8/ R9 ta có thể lựa chọn giữa phép đo hai cực (bipolar) hay là một cực (unipolar). Trong phép đo một cực (chỉ có những giá trị điện áp dương) điện áp lối

ra của tầng khuếch đại thuật toán được dẫn đến bộ biến đổi ADC 0804 không thay đổi. Khi các cầu nối được cắm cho phép đo hai cực, thì điện áp lối ra của OP07 phải chịu (một) sự dịch chuyển mức điện thế đi + 2,5 V. Nhưng vì hai điện trở R8 và R9 có cùng một giá trị nên điện áp cuối cùng còn cần phải đem chia đôi. Dải lối vào $\pm 2,5$ V được chuyển đổi theo cách này thành 0... 2,5 V. Hình 3.11 chỉ ra sơ đồ sắp xếp linh kiện trên mạch in, còn trên hình 3.12 ta có thể nhìn thấy sơ đồ mạch in của bộ biến đổi A/D này.

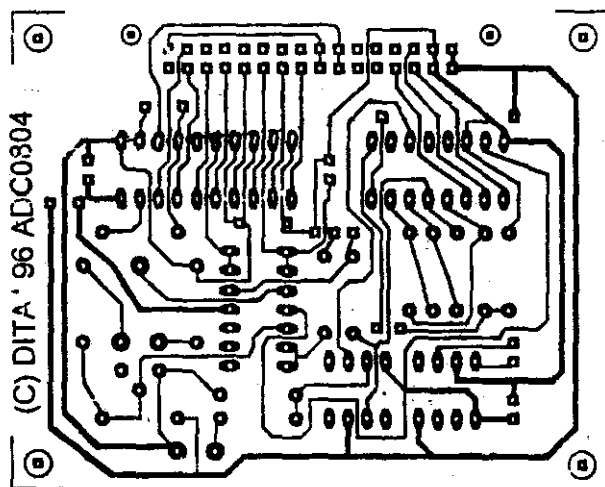


Hình 3.11 Sắp xếp các linh kiện trên bộ biến đổi A/D 8 bit với dải đo chuyển đổi được.

Các linh kiện cần thiết để lắp ráp môđun này được liệt kê ra sau đây:

Linh kiện bán dẫn:

- IC1: OP 07
- IC2: 74HC4051
- IC3: 82C43
- IC4: 82C43
- IC5: 82C43
- IC6: LT1009-2.5 V (hoặc tương tự, chẳng hạn LM 336Z-2.5)



Hình 3.12 Sơ đồ mạch in của bộ biến đổi A/D 8 bit với dải đo chuyển đổi được.

Điện trở:

R1: $3\text{ M}\Omega$

R2: $1\text{ M}\Omega$

R3, R8, R9, R13: $10\text{ k}\Omega$

R4: $6\text{ k}\Omega$

R5: $2\text{ k}\Omega$

R6, R7: $1\text{ k}\Omega$

R10 = $3,9\text{ k}\Omega$

R11, R12 = $27\text{ k}\Omega$

Tụ điện:

C1 = 150 pF

C2 = $1\text{ }\mu\text{F}$

C3, C4, C5 = $10\text{ }\mu\text{F}$

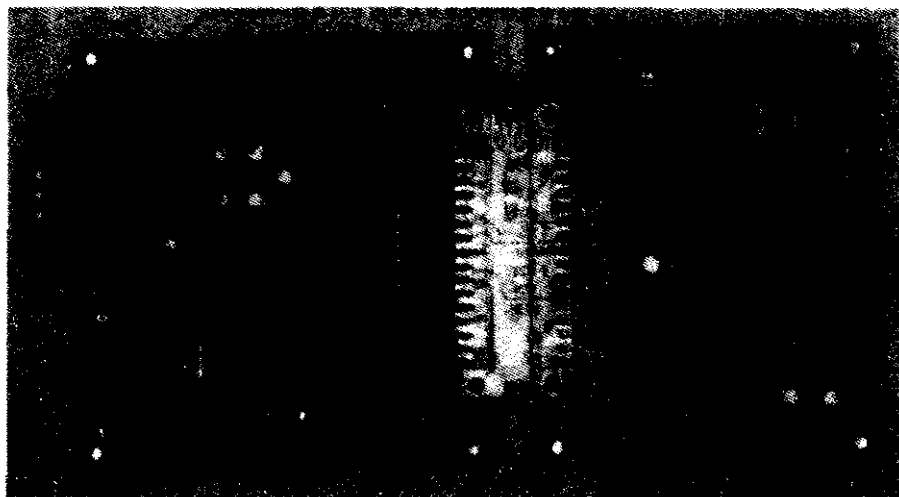
Điốt bán dẫn:

D1, D2, D3, D4: 1N 4148

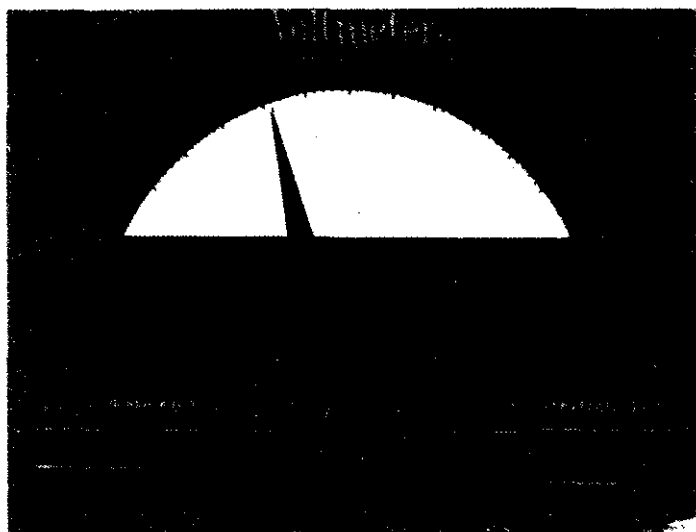
Phích cắm:

K1: dẹt 32 chân

Hình 3.13 chỉ ra tấm mạch của môđun biến đổi A/D 8 bit cùng với một môđun cơ sở đã trình bày ở chương 1.



Hình 3.13 Tấm mạch của môđun biến đổi A/D 8 bit cùng với một môđun cơ sở.



Hình 3.14 Một thí dụ cho màn hình chỉ thị analog.

Bằng phần mềm ta có thể viết ra một chương trình minh họa sự hoạt động của vi mạch ADC 0804 trong đó điện áp ở môđun biến đổi A/D được trình bày dưới dạng một bộ chỉ thị analog. Ở đây việc chuyển dải đo được thực hiện bằng phím chức năng. Ngoài ra bằng phím chức

năng cũng có thể thiết lập tốc độ truyền một cách rất thuận tiện. Trên hình 3.14 chỉ ra một thí dụ về cách trình bày bộ chỉ thị analog trên màn hình.

Phần mềm

Khi ghép nối với môđun cơ sở đã trình bày ở chương 1, bộ biến đổi A/D có thể đấu vào cả cổng máy in cũng như cổng nối tiếp của máy tính PC. Bởi vì chương trình dưới đây với những lời giải thích xen kẽ đã là một tài liệu rất đầy đủ nên ở đây không có thêm những lời mô tả chi tiết.

```
'=====
' Program: 0804 LPT
'
' Function: Chương trình này điều khiển môđun biến
' đổi A/D qua cổng máy in bằng việc chuyển dải đo
' của máy tính PC (xem hình 3.5)
'
' Hardware: Cần có bộ biến đổi A/D mô tả ở hình 3.5
' cũng như môđun cơ sở từ mục 1.2
'=====
```

```
DECLARE SUB lese.lpt (inbyte)
```

```
DIM SHARED outbyte, datereg, statreg, contrreg
```

```
'-----Sử dụng các dãy điều khiển sau-----
```

Valence	1	2	4	8	16	32	64	128
	D0	D1	D2	D3	D4	D5	D6	D7
Khởi động sự biến đổi	1	0/1	x	x	x		Dải đo	
Dải đo 10 V	1	1	x	x	x	0	0	0
Dải đo 5 V	1	1	x	x	x	1	0	0
Dải đo 2 V	1	1	x	x	x	0	1	0
Dải đo 1 V	1	1	x	x	x	1	1	0
Dải đo 0,5 V	1	1	x	x	x	0	0	1
Đón giá trị đo	0/1	1	x	x	x	x	x	x

-----Thiết lập chế độ ban đầu-----

```
basadr = &H378
datreg = basadr           'Datenregister
statreg = basadr + 1      'Statusregister
contrreg = basadr + 2     'Controlregister
```

CLS

```
PRINT "Dải đo nào ?... 10 V (1), 5 V (2), 2 V (3),
1 V (4), 0,5 V (5)"
```

INPUT A

```
IF A = 1 THEN mb = 3:   UMAX = 10 'Dải đo 10 V
IF A = 2 THEN mb = 35:  UMAX = 5  'Dải đo 5 V
IF A = 3 THEN mb = 67:  UMAX = 2  'Dải đo 2 V
IF A = 4 THEN mb = 99:  UMAX = 1  'Dải đo 1 V
IF A = 5 THEN mb = 131: UMAX = .5 'Dải đo 0,5 V
```

DO

-----Bắt đầu sự biến đổi-----

```
OUT datreg, mb - 2
OUT datreg, mb + 2
FOR I = 1 TO: NEXT      'Hãy chờ ít nhất 100 us
-----Đón nhận giá trị đo-----
OUT datreg, mb - 1
OUT datreg, mb + 1
CALL lese.lpt (inbyte)
```

```
umess = INT (inbyte / 256 * UMAX * 1000) / 1000
PRINT "Điện áp là: "; umess; "Volt"
```

```
IF INKEY$ = CHR$ (27) THEN EXIT DO
```

'Ngừng lại bằng phím ESC

LOOP

END

SUB lese.lpt (inbyte)

```

=====
'
'   Unterprogram: lese. lpt
'
'   Function: Chương trình con này đọc vào một byte
'   qua cổng máy in. Kết quả đứng trong biến inbyte
'
=====
'-----Đọc dữ liệu vào-----
'
    UT contrreg, 0          'init = 0
    inbyte1 = INP(statreg)  'Đọcvào từ D0,D1,D2,và D3
    OUT contrreg, 4         'init = 1
    inbyte2 = INP(statreg)  'Đọcvào từ D4,D5,D6,và D7
'
'-----Sắp xếp các bit dữ liệu đã đọc -----
'
'   inbyte1:      Statusregister bit 4 (SLCT) là D0
'                  Statusregister bit 5 (PE)   là D1
'                  Statusregister bit 6 (ACK)  là D2
'                  Statusregister bit 5 (BUSY) là /D3
'   inbyte2:      Statusregister bit 4 (SLCT) là D4
'                  Statusregister bit 7 (PE)   là D5
'                  Statusregister bit 6 (ACK)  là D6
'                  Statusregister bit 5 (BUSY) là /D7
'   inbyte = (((inbyte1 XOR 128) AND &HFO) / 16) +
'             ((inbyte2 XOR 128) AND &HFO)

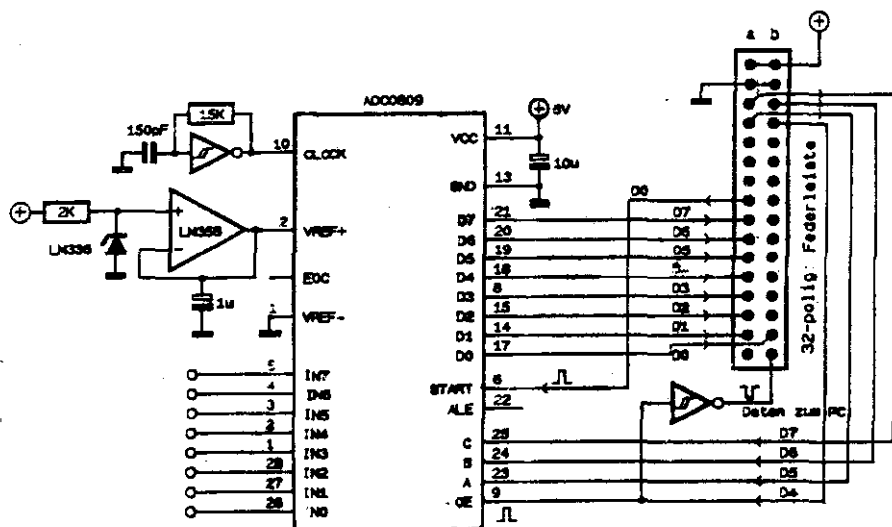
```

END SUB

3.3 MÔĐUN BIẾN ĐỔI A/D 8 BIT 8 KÊNH DÙNG VI MẠCH ADC 0809

Bộ biến đổi A/D mang tên ADC 0809 rất giống với bộ biến đổi A/D ADC 0804 đã mô tả trong mục 3.2. Sự khác nhau cơ bản là ở con số các kênh lối vào đã được tăng lên. Tổng cộng người sử dụng có 8 kênh làm việc hoàn toàn độc lập với nhau để lựa chọn. Ở đây cần chú ý là các điện áp được đo so với điện thế 0 V. Còn một đặc điểm đáng quan tâm hơn là sự tiêu thụ dòng điện của vi mạch hầu như không đáng kể (chỉ cỡ 300 μ A). Thời gian biến đổi cỡ 100 μ s. Các thông số kỹ thuật của bộ biến đổi ADC 0809 được kể ra như sau:

- ☐ Không cần đòi hỏi điều chỉnh điểm 0.
- ☐ Quét động 8 kênh bằng logic địa chỉ.
- ☐ Dải tín hiệu lối vào analog khi điện áp nguồn nuôi là + 5 V.
- ☐ Tất cả các tín hiệu tương thích TTL.
- ☐ Độ phân giải: 8 bit.
- ☐ Thời gian biến đổi: 100 μ s.
- ☐ Dòng tiêu thụ (bình thường): 0,3 mA.



Hình 3.15 Môđun biến đổi A/D 8 bit 8 kênh dùng vi mạch ADC 0809.

Hình 3.15 chỉ ra một mạch điển hình được thiết kế cho bộ biến đổi ADC 0809. Mạch điện này cũng có ổ nối 32 chân như là giao diện để ghép nối, qua đó mỗi môđun cơ sở ở chương 1 có thể đấu nối vào. Theo

cách này ta có thể điều khiển bộ biến đổi A/D ADC 0809 từ máy tính PC qua giao diện máy in hoặc là qua giao diện nối tiếp RS 232. Tín hiệu giữ nhịp dùng cho bộ biến đổi A/D cần phải được tạo ra ở bên ngoài và được dẫn đến chân CLOCK. Điện áp so sánh được đưa qua tầng lặp lại điện áp để đến chân ra V_{REF+} , chân này có điện trở nối vào cỡ 2,5 k Ω . Tám kênh nối vào analog được dẫn đến các chân IN0 đến IN7. Mẫu bit ở các nối vào địa chỉ A, B, C sẽ xác định xem kênh nào phải được lựa chọn. Khi đó, việc lựa chọn tuân theo quy định sắp xếp dưới đây:

C	B	A	Kênh nối vào được kích hoạt
0	0	0	IN0
0	0	1	IN1
0	1	0	IN2
0	1	1	IN3
1	0	0	IN4
1	0	1	IN5
1	1	0	IN6
1	1	1	IN7

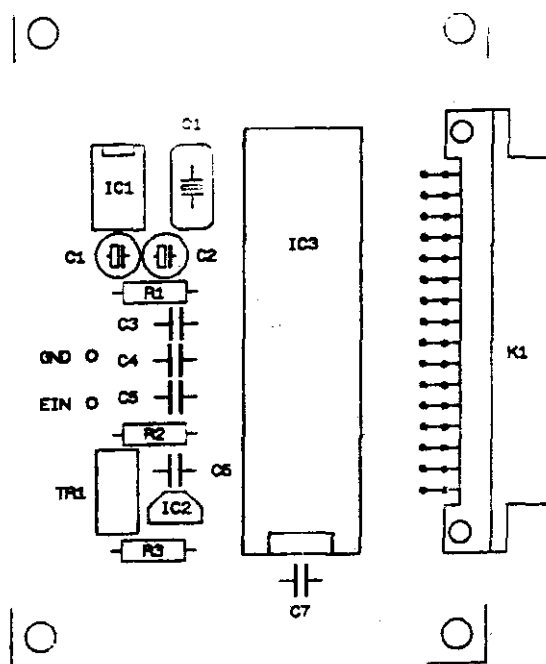
Nguyên tắc làm việc của bộ biến đổi ADC 0809 cũng không có gì phức tạp. Một xung dương ở chân START kích hoạt sự biến đổi. Qua đó mẫu bit ở nối vào địa chỉ A; B và C cũng đồng thời được chốt và xác định kênh cần biến đổi. Trong quá trình biến đổi chân ra EOC đứng ở mức Low, sau cỡ 100 μ s mức này sẽ chuyển sang High và báo hiệu sự kết thúc quá trình biến đổi. Sau đó kết quả của quá trình biến đổi sẽ xếp hàng ở các đường dẫn dữ liệu D0...D7. Khi OE = 1 các đường dẫn có thể được đọc tiếp.

3.4 MÔĐUN BIẾN ĐỔI A/D 12 BIT 1 KÊNH DÙNG VI MẠCH ICL 7109

Tất cả các bộ biến đổi đã trình bày từ trước đến giờ hoạt động theo nguyên tắc gần đúng liên tiếp. Ngược lại bộ biến đổi A/D 12 bit có tên ICL 7109, được sử dụng sau đây, hoạt động theo phương pháp hai độ dốc (dual slop). Trong một khoảng thời gian cố định t , một tụ điện được nạp bằng điện áp U_x qua một điện trở. Theo sự biến đổi của thời gian

Các thông số kỹ thuật:

- ☐ Độ phân giải 12 bit.
- ☐ Các lỗi vào vi phân đối với điện áp và nguồn so sánh.
- ☐ Dòng tiêu thụ (bình thường): cỡ 1,5 mA.
- ☐ Bit chẵn lẻ (parity).
- ☐ Bit quá thang (overflow).
- ☐ Nguồn nuôi ± 5 V.

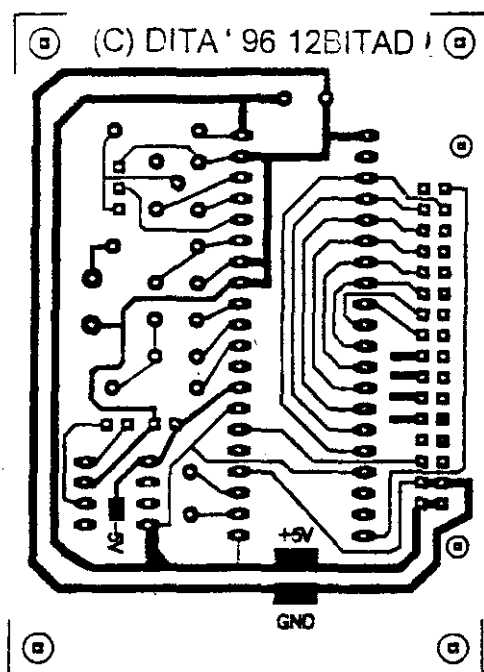


Hình 3.17 Sắp xếp các linh kiện trên bộ biến đổi A/D 12 bit.

Hình 3.16 chỉ ra sự đấu nối linh kiện quanh vi mạch ICL 7109. Ở đây bộ biến đổi A/D được đệm vào theo kiểu bất chéo tay, trong đó chân ra MODE (chân 21) được nối lên điện áp + 5 V. Việc trao đổi thông tin với một môđun cơ sở dùng cho giao diện nối tiếp ở chương 1 là rất đơn giản. Trước hết chân ra RUN/HOLD phải được đặt lên High. Sau đó vi mạch thực hiện một lần biến đổi. Tín hiệu /LOAD sẽ báo hiệu thời điểm nào các dữ liệu có giá trị đúng sẵn sàng. Các dữ liệu được truyền đi

thành hai byte. Trước hết là MSB giá trị cao với bit phân cực POL, bit quá thang (OR) và các bit dữ liệu B12... B9, sau đó là LSB giá trị thấp với các bit dữ liệu còn lại B8... B1. Xung Low ở lối ra LOAD bắt đầu việc gửi đi của các byte dữ liệu đang có giá trị tới máy tính PC. Ở đó chỉ còn lại công việc là đánh giá các bit dữ liệu.

Sự sắp xếp linh kiện của môđun biến đổi A/D 12 bit được minh hoạ trên hình 3.17, còn sơ đồ mạch in được chỉ ra trên hình 3.18.



Hình 3.18 Sơ đồ mạch in của bộ biến đổi A/D 12 bit.

Chương trình 7109COM.BAS chỉ ra cho ta thấy phần cứng trình bày ở trên được trao đổi trong QBasic như thế nào.

```
'=====
' Program: 7109 COM
'
' Function: Bằng chương trình này có thể điều khiển
'           môđun biến đổi A/D 12 bit trên hình 3.16 qua
'           cổng nối tiếp của máy tính PC. Giá trị điện áp đã
'           biến đổi được xuất ra thường xuyên lên màn hình
```


' Hardware: Cần có một môđun cơ sở từ chương 1 cũng
 ' như môđun biến đổi A/D 12 bit

'=====

DECLARE SUB lese.com (inbyte)

CLS

UREF = 5 'UREF = 0,5 Volt

'-----Mở COM2 với 9600 baud-----

OPEN "com2: 9600,N,8,1,CS,DS" FOR RANDOM AS #1

'-----RUN/ HOLD đặt lên High-----

PRINT #1, CHR\$(128);

'-----Bắt đầu vòng lặp liên tục để biến đổi A/D----

DO

'-----Đọc vào HighByte POL, OR, B12-B9-----

'-----và lọc lựa các Bit dữ liệu-----

CALL lese.com (inbyte)

valence = 0 'valence: giá trị

FOR i = 2 TO 5

Bit = (inbyte AND 2 ^ i) / 2 ^ i

valence = valence + Bit * 2 ^ (13 - i)

NEXT

POL = (inbyte AND 2 ^ 0) / 2 ^ 0 'Phân cực

OVR = (inbyte AND 2 ^ 1) / 2 ^ 1 'Quá thang

'-----Đọc vào LowByte: B8-B1-----

CALL lese.com (inbyte)

valence = valence + inbyte

u = INT(10000 * valence*2*REF/4096)/10000

IF OVR = 1 THEN PRINT "Quá thang (Overflow)"

IF POL = 1 AND OVR = 0 THEN

```

PRINT "Điện áp bằng: +"; u; "Volt"
IF POL = 0 AND OVR = 0 THEN
PRINT "Điện áp bằng: -"; u; "Volt"

IF INKEY$ = CHR$ (27) THEN EXIT DO
      'Ngừng lại bằng phím ESC

LOOP

PRINT #1, CHR$ (0);
CLOSE 1
END

SUB lese.com (inbyte)
'=====
'  Unterprogram: lese.com
'  Function: Chương trình này đọc một Byte qua giao
'  diện nối tiếp. Nếu nhĩ sự truyền bị nhiều thì một
'  dòng thông báo sẽ nhấp nháy và chương trình kết
'  thúc
'=====
      i = 0
      DO
      i = i + 1
      ,
      '-----Khi byte đã có mặt-----
      ,
      IF LOC (1) >= 1 THEN
      in$ = INPUT$ (1, #1)
      inbyte = ASC (in$)
      GOTO ketthuc
      END IF
      ,
      '-----Phép thử mới, đọc vào dữ liệu -----
      ,
      LOOP UNTIL i = 1000      'Cực đại là 1000 phép thử

```

```

'-----Không nhận được byte nào-----
'
CLS
PRINT "Sự truyền dữ liệu bị nhiễu !!!!!!"
PRINT
PRINT "Không nhận được ký tự nào !!!!!!"
PRINT
PRINT "Hãy kiểm tra lại ổ cắm, dây nối, phần
cứng..."
END
ketthuc:
END SUB

```

3.5 GHÉP NỐI MÔĐUN BIẾN ĐỔI A/D 8 KÊNH, 12 BIT TRỰC TIẾP VÀO CỔNG NỐI TIẾP

Vi mạch MAX 186 của hãng MAXIM là một bộ biến đổi A/D 8 kênh, 12 bit thuộc thế hệ mới nhất. Tất cả những gì mà vài năm trước đây còn là một mạch điện khá phức tạp thì trong vi mạch MAX186 đã được tích hợp lên trên một chip. Bên cạnh một bộ quét động 8 kênh, khối đánh dấu và giữ (Track & Hold) và một giao diện nối tiếp, vi mạch này còn chứa một nguồn điện áp so sánh cũng như một mạch phát xung giữ nhịp (ở bên trong). Điện áp nguồn nuôi + 5 V hoặc là ± 5 V đều dùng được. Tám lối vào tương tự đối với các phép đo hai cực hoặc là một cực có thể được so sánh với mass hoặc là cấu hình theo kiểu vi phân bằng phân mềm. Khi ghép nối với giao diện RS232 vi mạch này tỏ ra là đặc biệt hấp dẫn bởi vì dòng tiêu thụ nhỏ không đáng kể, chỉ cỡ 1,5 mA, còn điện áp nguồn có thể lấy ra từ các đường dẫn điều khiển của giao diện.

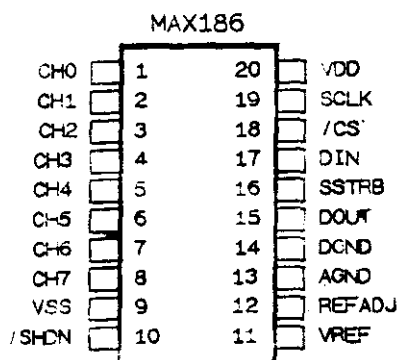
Để tạo ra một bộ biến đổi A/D 12 bit 8 kênh ở giao diện nối tiếp của máy tính PC chỉ cần có một vài linh kiện phụ thêm.

Các thông số kỹ thuật dưới đây đặc trưng cho vi mạch MAX 186:

- ☐ Tám lối vào so sánh với mass (một đầu) hoặc là 4 lối vào vi phân.
- ☐ Độ phân giải 12 bit.

- ☐ Điện áp nguồn nuôi + 5 V hoặc ± 5 V.
- ☐ Dòng tiêu thụ không đáng kể: cỡ 1,5 mA.
- ☐ Có bộ đánh dấu và giữ (Track & Hold) bên trong.
- ☐ Có nguồn so sánh 4,096 V bên trong.
- ☐ Lỗi vào một dấu hoặc là hai dấu, đặt được bằng phần mềm.
- ☐ Thời gian biến đổi lớn nhất: 10 μ s.

Hình 3.19 mô tả cách bố trí chân ra của vi mạch MAX 186.



Hình 3.19 Cách bố trí chân ra của vi mạch MAX 186.

Dưới đây là sự mô tả chức năng của các chân riêng biệt:

Chân	Tên	Chức năng
1	CH0	Các lỗi ra analog
2	CH1	(CH0 tương ứng với kênh 0, vân vân...)
3	CH2	
4	CH3	
5	CH4	
6	CH5	
7	CH6	
8	CH7	
9	Vss	Điện áp nguồn nuôi âm Nối với - 5 V ± 5 % hoặc với đất analog (AGND)
10	/SHDN	SHUT DOWN Khi lỗi vào này dẫn đến mức Low thì tất cả chức

năng của vi mạch không hoạt động. Sau đây dòng tiêu thụ chỉ cỡ 10 μA . Đối với các ứng dụng thông thường (chuẩn) chân này cần phải được bỏ trống. Trong loại hoạt động này phải nối một tụ điện 4,7 μF vào chân U_{REF}

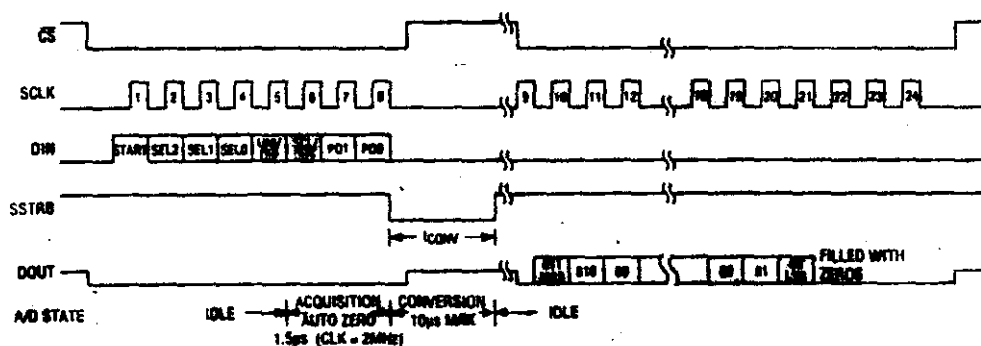
- | | | |
|----|------------------|---|
| 11 | U_{REF} | Điện áp so sánh dùng cho quá trình biến đổi A/D. Chân ra này cần phải được nối qua một tụ điện 4,7 μF xuống mass. Chân ra U_{REF} đồng thời là lối vào điện áp so sánh khi sử dụng một nguồn so sánh bên ngoài. |
| 12 | REFADJ | Qua lối vào này một sự tinh chỉnh cho điện áp so sánh bên trong có thể được thực hiện. Dải tinh chỉnh được có độ rộng là $\pm 1,5 \%$. |
| 13 | AGND | Mass cho mạch tương tự
Đồng thời là điểm so sánh trong phép đo một cực |
| 14 | DGND | Mass cho mạch số |
| 15 | DOUT | Lối ra dữ liệu nối tiếp
Qua lối ra này, các bit dữ liệu là kết quả của bộ biến đổi được truyền ra. Cứ với mỗi sườn âm của tín hiệu thì một bit lại được đưa ra . |
| 16 | SSTRB | Lối ra STROBE nối tiếp
Khi dùng bộ giữ nhịp bên trong, SSTRB sẽ chuyển sang mức Low trong giai đoạn biến đổi. Khi CS = 1, thì sau đó lối ra chuyển sang trạng thái điện trở cao. |
| 17 | DIN | Lối vào dữ liệu nối tiếp
Qua lối vào này một byte điều khiển trong vi mạch MAX 186 được giữ nhịp bằng sườn âm ở SCLK. |
| 18 | /CS | Chip Select (lựa chọn chip)
Byte điều khiển được đọc vào chỉ khi CS = 0. Khi CS chuyển sang mức High thì DOUT ở trạng thái điện trở cao. |
| 19 | SCLK | Lối vào giữ nhịp nối tiếp
SCLK là lối vào giữ nhịp dùng cho việc nạp vào các byte dữ liệu theo cách nối tiếp và dùng cho việc đọc |

ra theo cách nối tiếp các kết quả biến đổi.

20 VDD Điện áp nguồn nuôi dương: $+5\text{ V} \pm 5\%$

Sự truyền dữ liệu

Sự biến đổi theo thời gian (Timing) khi truyền dữ liệu giữa MAX 186 và một phần tử khác được chỉ ra trên hình 3.20 dưới đây:



Hình 3.20 Sự phân bố thời gian khi truyền dữ liệu nối tiếp bằng vi mạch MAX 186.

Đối với quá trình biến đổi A/D, loại hoạt động này sử dụng bộ giữ nhịp bên trong. Còn để truyền đi tín hiệu chỉ cần có ba đường dẫn điều khiển: SCLK, DIN và DOUT. Một cuộc truyền dữ liệu đầy đủ được chia thành ba giai đoạn. Trước hết, một byte điều khiển được giữ nhịp bên trong vi mạch MAX 186, byte điều khiển này khẳng định đúng loại hoạt động của quá trình biến đổi A/D. Tiếp đó, quá trình biến đổi thực sự của tín hiệu lỗi vào analog thành các giá trị digital bắt đầu. Sau khi kết thúc quá trình biến đổi, các kết quả có thể được đọc ra theo cách nối tiếp. Từ hình 3.20 ta thấy rõ là mỗi bit riêng lẻ của byte điều khiển được truyền đi bằng sườn dương của tín hiệu SCLK trong MAX 186. Bit 7 là bit khởi động (Startbit) và luôn có mức High. Với 3 bit tiếp theo SEL 2, SEL 1 và SEL 0 thì lỗi vào analog cần được biến đổi sẽ được lựa chọn. Bit tiếp theo khẳng định phép đo là một cực (unipolar, Bit = 1) hay là

hai cực (bipolar, Bit = 0). Bằng bit 2, người sử dụng có thể đo so với mass (SGL/ DIFF = 1) hay là biến đổi vi phân giữa hai tín hiệu analog. Hai bit cuối cùng để chứng minh cho kiểu hoạt động "giữ nhịp bên trong" bằng PD1 = 1 và PD0 = 0.

Bảng tiếp theo chỉ ra sự sắp xếp của các bit điều khiển SEL 2, SEL 1 và SEL 0 theo lối vào tương tự đã được lựa chọn.

Phép đo một cực (unipolar): SGL/ DIFF = 1

SEL2	SEL1	SEL0	CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7	AGND
0	0	0	+								-
1	0	0		+							-
0	0	1			+						-
1	0	1				+					-
0	1	0					+				-
1	1	0						+			-
0	1	1							+		-
1	1	1								+	-

Phép đo hai cực (bipolar): SGL/ DIFF = 0

SEL2	SEL1	SEL0	CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7
0	0	0	+	-						
0	0	1			+	-				
0	1	0					+	-		
0	1	1							+	-
1	0	0	-	+						
1	0	1			-	+				
1	1	0					-	+		
1	1	1							-	+

Sau khi ghi vào byte điều khiển, tín hiệu SSTRB chuyển sang Low, báo hiệu sự bắt đầu của quá trình biến đổi A/D. Sau khoảng thời gian cỡ 10 μ s quá trình biến đổi kết thúc và tín hiệu SSTRB nhận trở lại mức

mạch như thế này mà điện áp lối vào cần có ở vi mạch LM 2936 sau đó được bảo đảm, chẳng hạn khi DTR chuyển trong thời gian ngắn sang mức Low.

Các tín hiệu điều khiển DTR và RTS được sử dụng không chỉ để tạo nguồn nuôi mà còn để tạo ra bit dữ liệu DIN và xung giữ nhịp đọc cũng như ghi SCLK. Việc thích ứng điện áp giữa mức ± 12 V của đường dẫn DTR và RTS với mức TTL thực hiện chỉ bởi một điốt ổn áp với một điện trở hạn chế dòng. Qua lối vào CTS, các bit dữ liệu đứng xếp hàng ở lối ra DOUT được đọc vào máy tính PC.

Phần mềm

Khi lập trình cho bộ biến đổi A/D 12 bit 8 kênh, trước hết cần chú ý đến sự đảo ngược của tín hiệu DTR và RTS. Việc trao đổi của tín hiệu này cũng như sự hỏi lại của mức logic của CTS đã được mô tả khá đầy đủ trong mục 1.2. Dưới đây là danh sách các lệnh trong chương trình MAX 186.BAS cho phép trao đổi với phần cứng đã minh hoạ trên hình 3.21 và hiển thị các điện áp đo được lên màn hình.

```
'=====
' Program: MAX186
'
' Fuction: Chương trình này điều khiển mạch điện
' trên hình 3.21 qua giao diện nối tiếp của máy
' tính PC, sao cho điện áp được đo và hiển thị lên
' màn hình. Độ phân giải ở mỗi kênh là 12 bit.
'=====
COLOR 0, 15
CLS

basadr = &H2F8      'Địa chỉ cơ bản COM2

a$ = "8-Channel 12-Bit A/D-Converter ở cổng RS232"
a = LEN (a$): b = (80 - a) / 2 - 1
LOCATE 1, b: PRINT CHR$ (201);
STRING$ (a + 2, CHR$ (205)); CHR$ (187)
LOCATE 2, b: PRINT CHR$ (186)
```

```

LOCATE 2, b + 2: PRINT a$
LOCATE 2, a + b + 2: PRINT CHR$ (186)
LOCATE 3, b: PRINT CHR$ (200);
STRING$ (a + 2, CHR$ (205)); CHR$ (188)
PRINT
LOCATE 5, 20: PRINT "Điện áp ở kênh 0:"
LOCATE 7, 20: PRINT "Điện áp ở kênh 1:"
LOCATE 9, 20: PRINT "Điện áp ở kênh 2:"
LOCATE 11, 20: PRINT "Điện áp ở kênh 3:"
LOCATE 13, 20: PRINT "Điện áp ở kênh 4:"
LOCATE 15, 20: PRINT "Điện áp ở kênh 5:"
LOCATE 17, 20: PRINT "Điện áp ở kênh 6:"
LOCATE 19, 20: PRINT "Điện áp ở kênh 7:"
LOCATE 23, 1: PRINT "Hướng dẫn"
LOCATE 23, 20: PRINT "ESC: Kết_thúc"
FOR i=0 TO 7: LOCATE 5+2*i, 55: PRINT "Volt":
NEXT

```

```

'-----DTR and RTS is + 12 V-----

```

```

OUT (basadr + 4), INP (basadr + 4) OR 3

```

```

CH = 0

```

```

'-----Vòng lặp vô tận bắt đầu-----

```

```

'-----Toàn bộ 8 kênh được biến đổi kế tiếp nhau-----

```

```

DO

```

```

HELP = 4 * CH MOD 7

```

```

IF CH = 7 THEN HELP = 7

```

```

Controlbyte = 142+16* HELP 'unipolar,single ended
                        'Giữ nhịp trong

```

```

'-----Tủ điều khiển nối tiếp đếm MAX 186-----

```

```

'-----DIN = /RTS và SCLK = /DTR-----

```

```

FOR i = 7 TO 0 STEP - 1
DIN = (controlbyte AND 2 ^ i) / 2 ^ i
IF DIN = 0 THEN OUT (basadr+4), INP(basadr+4) OR 2
IF DIN = 1 THEN OUT(basadr+4), INP(basadr+4) AND (255-2)

```

-----Tạo ra xung SCLK-----

```

OUT (basadr + 4), INP (basadr + 4) AND (255 - 1)
SCLK = 1
OUT (basadr + 4), INP (basadr + 4) OR 1
SCLK = 0
NEXT

```

-----Bây giờ sự biến đổi diễn ra-----

```

FOR i = 1 TO 5: NEXT      'Chờ ít nhất 10 ms

```

-----Đón nhận kết quả-----

```

valence = 0
FOR i = 11 TO 0 STEP - 1
OUT (basadr + 4), INP (basadr + 4) AND (255 - 1)
SCLK = 1
OUT (basadr + 4), INP (basadr + 4) OR 1
SCLK = 0
CTS = (INP (basadr + 6) AND 16) / 16
valence = valence + CTS * 2 ^ i
NEXT

```

```

FOR i = 1 TO 4
OUT (basadr + 4), INP (basadr+4) AND (255-1)
SCLK = 1
OUT (basadr + 4), INP (basadr + 4) OR 1
SCLK = 0
NEXT

```

-----Tính toán điện áp lỗi ra-----

```
volt = valence * 4.096 / 4096
LOCATE 5 + (2 * CH), 45: PRINT USING "#.###"; volt
IF INKEY$ = CHR$ (27) THEN EXIT DO
CH = CH + 1: IF CH = 8 THEN CH = 0

LOOP
END
```

Hình 3.22 chỉ ra sự trình bày màn hình sau khi khởi động chương trình MAX 186.BAS kể trên.

BỘ BIẾN ĐỔI AD 8 KANH 12 BIT ĐAU VAO CONG RS 232		
Dien_ap o kenh 0	3.009	Volt
Dien_ap o kenh 1	2.673	Volt
Dien_ap o kenh 2	0.742	Volt
Dien_ap o kenh 3	1.079	Volt
Dien_ap o kenh 4	1.399	Volt
Dien_ap o kenh 5	4.095	Volt
Dien_ap o kenh 6	4.095	Volt
Dien_ap o kenh 7	2.060	Volt
Huong_dan: ESC : Ket_thuc		

Hình 3.22 Trình bày trên màn hình khi khởi động chương trình MAX186.BAS.

3.6 CÁC ỨNG DỤNG CỦA BỘ BIẾN ĐỔI A/D

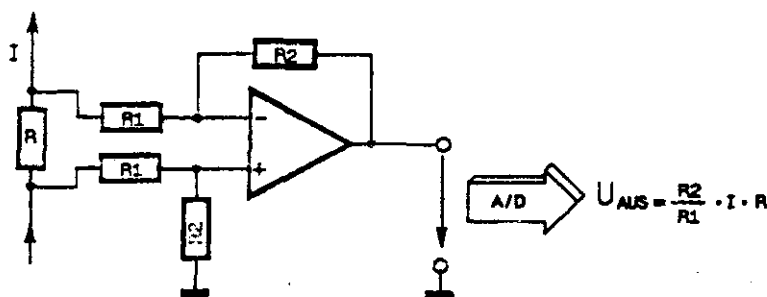
3.6.1 ĐO DÒNG ĐIỆN

Việc đo dòng điện được tiến hành bằng cách áp dụng định luật Ôm, trong đó ta đo sụt điện áp ở trên một điện trở do dòng điện đi qua gây nên. Hình 3.23 chỉ ra một mạch điện, bao gồm tầng khuếch đại thuật

toán và bốn điện trở. Bằng công thức được dẫn ra dưới đây có thể tính được dòng điện I theo điện áp lỗi ra U_{out} (hay là U_{aus}).

$$I = \frac{R_1 * U_{out}}{R_2 * R}$$

Khi áp dụng công thức cần phải chú ý là điện trở R_1 và R_2 tạo thành một cặp và đồng nhất với nhau về giá trị một cách rất chính xác. Độ lệch danh định của cặp này không được phép quá 0,5 %. Một ưu điểm nổi bật của mạch điện này là điện thế đồng nhịp (in-phase) ở điện trở R có thể lớn hơn điện áp nguồn nuôi của khuếch đại thuật toán. Điều này đúng chỉ với một điều kiện là điện áp ở các lối vào của khuếch đại thuật toán nằm trong dải đồng nhịp của tầng khuếch đại.



Hình 3.23 Mạch đo dòng điện.

3.6.2 DO ĐIỆN ÁP VI PHÂN

Trong nhiều ứng dụng của kỹ thuật đo, ta lại cần phải đo vi phân (hiệu số) điện áp rất nhỏ. Như là một thí dụ ta có thể dẫn ra trường hợp đầu đo độ giãn nở, khi chịu tải cực đại cũng lại chỉ cho ra một điện áp cỡ 20... 30mV. Nhưng tín hiệu này lại không so sánh với mass mà có một điện áp đồng nhịp (in-phase) cỡ một vài vôn lại chồng chất lên nó. Để đo tín hiệu loại này cần có một tầng khuếch đại đo minh họa trên hình 3.24.

Ta đều biết đặc trưng cơ bản của tầng khuếch đại đo là trở kháng lối vào cao và sự loại trừ đồng nhịp (in-phase rejection) cũng như hệ số khuếch đại có thể thiết lập được một cách chính xác. Đặc tính này làm

cho nó trở thành phần tử lý tưởng được dùng để khuếch đại chính xác các tín hiệu nhỏ, khi mà tín hiệu này lại được chồng chất lên một tín hiệu đồng nhịp (in -phase) có biên độ lớn hơn.

Hai tầng khuếch đại lồng vào trên hình 3.24 có một điện trở phản hồi chung R_1 . Cách đấu mạch này làm cho hai lối vào của hai khuếch đại thuật toán trở thành hai lối vào vi phân.

Các điện áp lối vào U_1 và U_2 được truyền đi qua mạch phản hồi của hai khuếch đại thuật toán lên hai đầu của điện trở R_1 , vì thế ở đây xuất hiện một giá trị điện áp, tương ứng với vi phân của điện áp lối vào. Điện áp này gây ra một dòng điện chung đi qua điện trở R_1 , dòng điện này lại dẫn đến việc tạo ra một điện áp vi phân đã được khuếch đại ở lối ra. Ưu điểm cơ bản của mạch điện này là ở chỗ qua việc thay đổi một điện trở riêng lẻ (R_1) ta có thể thiết lập được hệ số khuếch đại.

Điện áp đồng nhịp (in-phase) đặt lên đồng thời ở hai lối vào không cho ra tín hiệu trên R_1 và vì thế cũng không được khuếch đại lên. Tất nhiên, nói cho đúng hơn là chúng có xuất hiện ở lối ra của hai khuếch đại thuật toán nhưng với hệ số khuếch đại bằng 1. Việc loại bỏ các thành phần cùng pha này do tầng khuếch đại thuật toán cuối cùng đảm nhiệm, tầng này tác dụng như là bộ khuếch đại trừ.

Tóm lại với tầng khuếch đại đo hay còn gọi là khuếch đại dụng cụ minh hoạ trên hình 3.24 có thể kể ra những ưu điểm sau:

☐ Trở kháng lối vào rất cao

Khác với các bộ khuếch đại trừ đơn giản, mạch lối vào và mạch phản hồi cách ly với nhau. Nguồn tín hiệu bị chịu tải chỉ bởi dòng lối vào của tầng khuếch đại dụng cụ. Vì thế điện trở lối vào có thể nằm trong dải $10^{12} \Omega$.

☐ Can nhiễu không đáng kể ở các bộ khuếch đại trừ đơn giản các nguồn can nhiễu gây ra ảnh hưởng độc lập với nhau:

- a) Can nhiễu vì nhiệt của điện trở lối vào.
- b) Can nhiễu riêng ở lối vào.
- c) Can nhiễu do dòng lối vào của khuếch đại thuật toán.

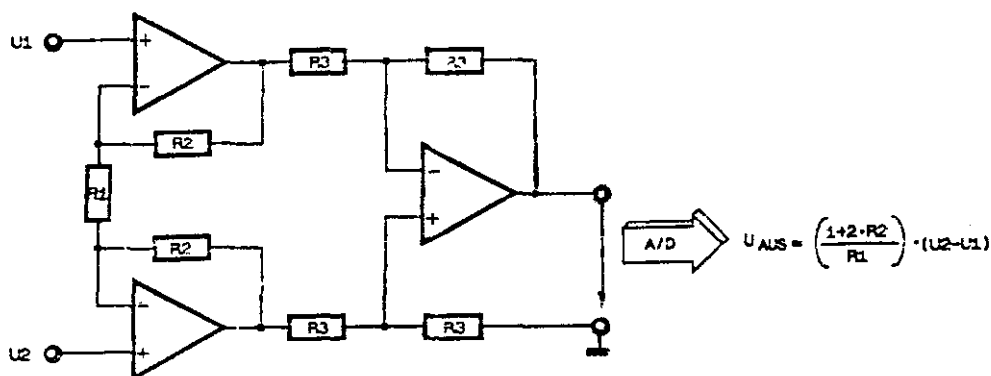
Các thành phần can nhiễu a) và c) được khử đi hoặc là hạn chế khi sử dụng một tầng khuếch đại dụng cụ.

- ❑ Độ không tuyến tính của hệ số khuếch đại không đáng kể.

Độ không tuyến tính của hệ số khuếch đại đạt đến giá trị 0,05 % hoặc tốt hơn thế.

- ❑ Việc đặt hệ số khuếch đại rất đơn giản

Hệ số khuếch đại có thể được xác định chỉ qua sự thay đổi của điện trở R_1 .



Hình 3.24 Tầng khuếch đại dụng cụ.

3.6.3 ĐO ĐIỆN TRỞ

Để đo điện trở cần có một nguồn dòng rất chính xác, để cung cấp dòng điện đi qua điện trở cần phải đo. Sau đó từ sụt thế trên điện trở ta có thể tính ra giá trị điện trở cần xác định. Hình 3.25 chỉ ra một mạch điện để đo điện trở, tuy đơn giản nhưng không kém phần hấp dẫn.

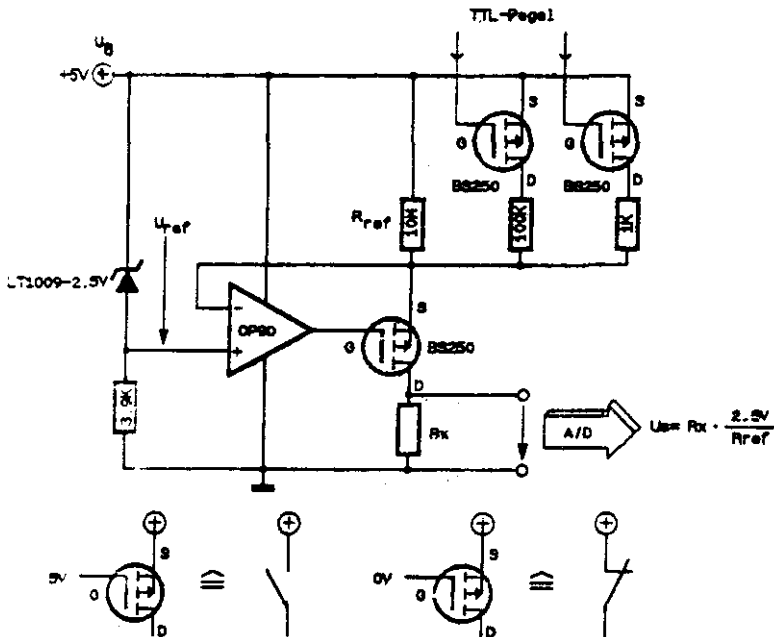
Phần chính của mạch điện này là một tầng khuếch đại thuật toán, có tín hiệu lỗi ra điều khiển một tranzito trường loại MOS.

Mạch điện này hoạt động như thế nào ?

Tầng khuếch đại thuật toán điều khiển dòng điện I_{SD} qua điện áp của sao cho ở lỗi vào đảo và không đảo trên thực tế có cùng một điện áp. Sự sai khác giữa hai điện áp này phải cỡ con số thứ 5 đứng sau dấu

phẩy (phần trăm milivôn). Điện áp ở lối vào không đảo được cho trước bằng một điện áp so sánh và tính theo phương trình:

$$U_+ = U_B - U_{REF} = U_-$$



Hình 3.25 Mạch đo điện trở.

Từ phương trình này có thể suy ra là điện áp ở điện trở so sánh tương ứng với điện áp so sánh. Vì thế qua điện trở so sánh chỉ có một dòng điện không đổi với giá trị $I = U_{REF} / R_{REF}$ chạy qua. Dòng điện này cũng chạy qua tranzito BS 250 và điện trở cần đo. Độ lớn của các điện trở này là không quan trọng. Ở đây cần chú ý là sụt điện áp trên điện trở R_x không thể lớn hơn $U_B - U_{REF}$.

Tiếp đó điện áp trên điện trở R_x có thể được thu thập vào máy tính PC qua một bộ biến đổi A/D. Giá trị chưa biết của điện trở R_x được tính toán theo phương trình sau:

$$R_x = \frac{R_{ref} \cdot U_a}{U_{ref}}$$

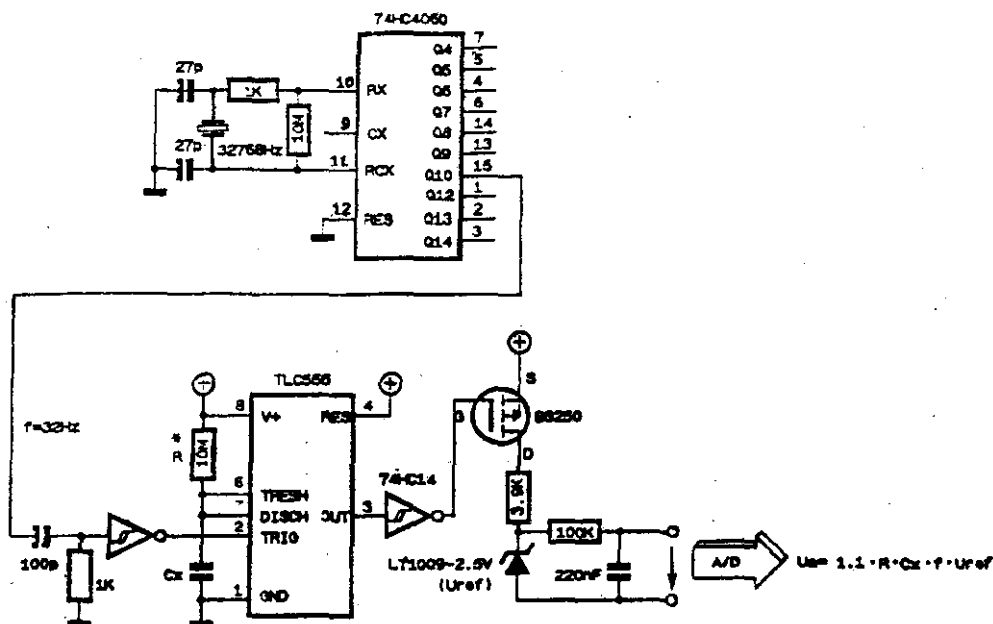
Các dải đo khác nhau có thể tạo ra bằng cách thay đổi điện trở so sánh. Để chuyển đổi dải đo cũng có thể sử dụng luôn tranzito BS 250. Tranzito này được điều khiển ở cực cửa (GATE) bằng mức TTL, còn cách hoạt động thì giống như một chuyển mạch. Khi đặt điện áp 5 V vào cực cửa (GATE) thì tranzito bị cấm tương ứng với một chuyển mạch ở trạng thái ngắt. Khi ta đặt một tín hiệu Low vào cực cửa thì tranzito chuyển mạch sang trạng thái mở và không có sụt áp trên tranzito. Trong mạch điện trên hình 3.25, điện trở so sánh hiệu dụng được tính toán qua mạch song song bao gồm cả các điện trở ở cực máng (Drain) của tranzito đang trong trạng thái dẫn.

3.6.4 ĐO ĐIỆN DUNG

Trong mục 2.6.4.10 ta đã làm quen với một phương pháp đo điện dung. Trong khi mạch điện này làm việc thuần túy với các tín hiệu số (digital) thì một phương pháp đo analog dưới đây cần phải được giới thiệu. Muốn thế cần có một bộ biến đổi điện dung thành điện áp tuyến tính. Một mạch đơn giản dùng cho một bộ biến đổi loại này được mô tả trên hình 3.26.

Tụ điện cần phải đo được sử dụng cùng với một điện trở đo như là hàng số thời gian của một tầng trigger một trạng thái cân bằng (TLC 555). Với mỗi xung âm lật trạng thái (trigger-impul) ở chân 2 của TLC 555, tôi ra OÚT chuyển sang mức High trong thời gian $T = 1,1 \cdot R \cdot C_x$. Bởi vì độ rộng của xung được lặp lại với tần số của tín hiệu trigger (ở đây $f = 32 \text{ Hz}$) nên bằng việc tích phân các xung lối ra sẽ nhận được một điện áp một chiều tỷ lệ với điện dung C . Công thức để tính điện áp lối ra có thể được lấy từ hình 3.26. Để có độ chính xác cao cho phép đo thì ta phải quan tâm đến nhiều đại lượng. Điện trở so sánh R có thể được chọn theo cấp chính xác 0,1 ... 1 %. Tần số của tín hiệu trigger được xác định bởi một bộ dao động thạch anh (quartz) và do vậy gây nên một sai số hoàn toàn không đáng kể. Đại lượng cuối cùng là điện áp so sánh. Loại ổn áp được dùng ở đây là LT 1009-2,5 có độ chính xác cỡ 0,2 %. Tóm lại ta có thể đi đến kết luận là bằng mạch điện minh họa trên hình 3.26, mặc dù việc thiết kế khá đơn giản nhưng ta vẫn có thể đạt được kết quả khá chính xác.

Có thể tạo ra các dải đo điện dung khác nhau qua việc chuyển mạch các điện trở R (so sánh với hình 3.25).



Hình 3.26 Mạch đo điện dung.

Độ rộng xung của TLC 555 cần phải không được vượt quá $2/3$ của chu kỳ của tín hiệu trigger. Nhờ vậy có thể ước đoán được điện dung cần đo cực đại bằng phương trình sau:

$$C_x \leq \frac{2}{3} * \frac{1}{1.1 \cdot f \cdot R}$$

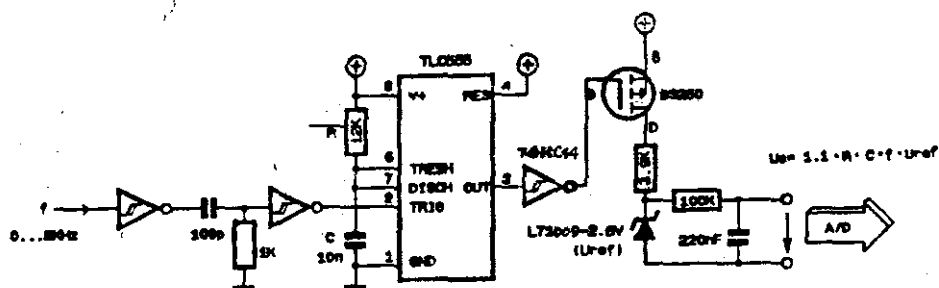
Với các giá trị cho trên hình 3.26 sẽ nhận được một dải đo có giá trị lớn nhất là 2 nF. Để đo được các tụ điện lớn hơn thì điện trở R cần phải chọn nhỏ đi tương ứng.

3.6.5 ĐO TẦN SỐ

Phương pháp đo thông thường được áp dụng để đo tần số là thuần túy digital và thực chất là phải đếm số các xung của tín hiệu lỗi vào trong một khoảng thời gian mở cổng T được xác định chính xác. Phương

pháp này đã được áp dụng ở mục 2.6.4.3. Một phương pháp đo khác sẽ được giới thiệu dưới đây.

Ở đây một điện áp lỗi ra analog được tạo ra tỷ lệ với tần số lỗi vào. Bộ biến đổi cần thiết đã được sử dụng ở trong mục trước đây. Hình 3.27 chỉ ra mạch điện để đo tần số theo phương pháp tương tự.



Hình 3.27 Mạch đo tần số theo phương pháp analog.

Tín hiệu lỗi vào được dẫn đến sau việc tạo ra một xung ở chân 2 của TLC 555. Vì mạch này tạo ra một thời gian xung cố định với $T = 1,1 \cdot R \cdot C$. Chuỗi xung (pulse repetition) được xác định qua tần số của tín hiệu lỗi vào. Khi tần số là nhỏ thì sự tham gia của thành phần điện áp một chiều cũng nhỏ. Khi tần số cao dần thì độ rộng xung tính theo tỷ lệ phần trăm cũng tăng lên và điện áp lỗi ra cũng tăng theo. Mạch điện tương ứng với một bộ điều biến tỷ lệ Xung/ Nghỉ, trong đó tỷ lệ Xung/ Nghỉ được xác định bằng tần số của tín hiệu lỗi vào. Để xác định giá trị của điện trở và tụ điện ta áp dụng quy tắc sau: Giá trị nhỏ nhất của chu kỳ tín hiệu lỗi vào cần phải được chọn bằng 150 % (hay là 2/3) của độ rộng xung T. Khi đó tần số cực đại của tín hiệu lỗi vào có thể được tính như sau:

$$f_{\max} = \frac{2}{3} * \frac{1}{1,1 \cdot R \cdot C}$$

3.6.6 ĐO NHIỆT ĐỘ

Với những hiểu biết về các mạch đo các đại lượng cơ bản đã trình bày ở trên, ta có thể vận dụng để đo nhiều đại lượng khác như nhiệt độ,

Các thông số kỹ thuật dưới đây làm cho vi mạch này trở nên đáng quan tâm:

- ☐ Định thang trực tiếp theo độ Celsius ($^{\circ}\text{C}$).
- ☐ Tín hiệu lối ra bằng $10 \text{ mV}/^{\circ}\text{C}$.
- ☐ Độ chính xác được bảo đảm: không kém hơn $0,5^{\circ}\text{C}$.
- ☐ Độ chính xác trong vùng nhiệt độ phòng: $0,25^{\circ}\text{C}$.
- ☐ Điện áp nguồn nuôi $4 \dots 30 \text{ V}$.
- ☐ Dòng điện tiêu thụ $60 \mu\text{A}$.
- ☐ Mức độ không tuyến tính: loại $0,25^{\circ}\text{C}$.

Trên hình 3.28 chỉ ra một mạch dùng vi mạch LM 35 làm đầu đo nhiệt độ. Điện áp so sánh được thiết lập cỡ $0,64 \text{ V}$. Nhờ vậy có được dải đo $0 - 128^{\circ}\text{C}$ với độ chính xác cỡ $0,5^{\circ}\text{C}$.

Đo nhiệt độ bằng cặp nhiệt điện

Việc sử dụng cặp nhiệt điện để đo nhiệt độ có một số ưu điểm hơn hẳn khi so sánh với một số đầu đo khác. Các cặp nhiệt điện có dải nhiệt độ sử dụng lớn hơn, thời gian đáp ứng nhanh hơn cùng với giá thành cũng phải chăng hơn. Tùy theo từng loại cặp nhiệt mà dải nhiệt độ sử dụng có thể bao trùm từ -200°C đến 1.700°C . Ngoài ra, đường đặc trưng nhiệt của nó có thể xem như là tuyến tính trong từng dải nhiệt độ xác định.

Về mặt cấu tạo, cặp nhiệt bao gồm hai dây dẫn khác nhau được hàn chung lại tại một điểm. Khi ta chế tạo một tiếp xúc giữa hai kim loại ở cùng một nhiệt độ thì cũng sẽ xuất hiện ở hai đầu một điện áp nhỏ. Độ lớn của điện áp này phụ thuộc vào nhiệt độ và vào kim loại được dùng làm cặp nhiệt. Do đó trong các mạch điện tử do khi lắp ráp đã hàn các linh kiện khác nhau vào mạch nên cũng xuất hiện những cặp nhiệt điện, cho dù ta hoàn toàn không mong muốn. Chẳng hạn, do tiếp xúc khi hàn giữa thiếc và đồng cũng xuất hiện một cặp nhiệt điện với sức điện động cỡ $3 \mu\text{V}/^{\circ}\text{C}$.

Với một cặp nhiệt điện ta luôn luôn có thể đo được sự khác nhau về nhiệt độ giữa điểm hàn và đầu kẹp, và vì thế luôn cần có một nhiệt độ

để so sánh. Nhiệt độ so sánh - mà thường chọn là nhiệt độ của dầu kẹp - có thể sử dụng luôn giá trị nhiệt độ 0°C . Một cách khác là ta dùng một cặp nhiệt thứ hai và nhúng vào một cốc nước đá đang tan. Cả hai cặp nhiệt này được đầu nối tiếp với nhau còn kết quả đo được đọc trực tiếp ra nhiệt độ thực. Một khả năng khác nữa là dùng nhiệt độ môi trường và xem như nhiệt độ này là không đổi. Khi sự thăng giáng của nhiệt độ môi trường nhỏ hơn độ phân giải của nhiệt độ muốn đo thì đây là khả năng đơn giản nhất để đo nhiệt độ bằng cặp nhiệt. Mạch điện tử để ghép nối cũng đơn giản. Cặp nhiệt được đầu trực tiếp vào một vi mạch khuếch đại thuật toán có độ trôi theo nhiệt độ và độ lệch điểm 0 (offset) nhỏ, chẳng hạn loại OP 07, còn hệ số khuếch đại mong muốn được đặt bằng một điện trở thích hợp.

Một giải pháp khác chính xác hơn là dùng chỉ một đầu đo nhưng nhiệt độ dầu kẹp được đo bằng một đầu đo thứ hai. Điện áp lấy ra từ đầu đo thứ hai sau đó được điện áp từ cặp nhiệt điện trừ đi. Đầu đo nhiệt độ dầu kẹp thường được dùng là một tranzito có hai chân phát-gốc được nối tắt và cung cấp một điện áp ở lối ra là $2\text{ mV}/^{\circ}\text{C}$ hoặc là dùng một đầu đo bán dẫn như loại AD 590. Đầu đo phụ trợ này cần phải có tiếp xúc nhiệt với dầu kẹp để phản ánh đúng nhiệt độ của đầu đo này.

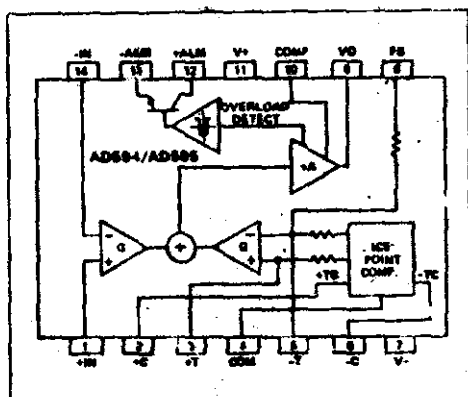
Bảng thông kê dưới đây chỉ ra điện áp lối ra hoặc như vẫn thường gọi là sức điện động nhiệt điện và dải nhiệt độ sử dụng của những cặp nhiệt điện thường gặp nhất.

Loại	Cặp nhiệt	Điện áp lối ra (V/ $^{\circ}\text{C}$) trong dải $0 - 100^{\circ}\text{C}$	Dải nhiệt độ ($^{\circ}\text{C}$)
E	Nikel (10%)/ Chrom-Constantan	67,5	-200 đến + 1.000
J	Sắt-Constantan	54,4	-200 đến + 1.200
K	Nickel (10%)/Chrom-Nickel	40,4	-200 đến + 1.300
R	Platin/ (13%) Rhodium-Platin	7,5	-200 đến + 1.700
S	Platin/(10%) Rhodium-Platin	7,3	-200 đến + 1.700
T	Đồng-Constantan	46,8	-200 đến + 400

Điện áp lối ra được dẫn chứng cho dải 0°C đến 100°C . Nhưng các cặp đều không có quan hệ tuyến tính giữa điện áp lối ra và nhiệt độ.

Trên thực tế người ta thường quan niệm các cặp nhiệt có điện áp lỗi ra tuyến tính trong từng dải 100 độ một, chẳng hạn 0 - 100; 100 - 200; vân vân... Do vậy ta cũng thấy là việc tuyến tính hoá trong toàn bộ dải nhiệt độ sử dụng bằng kỹ thuật mạch điện tử là một vấn đề hoàn toàn không đơn giản. Thực tế cũng cho thấy một giải pháp rất có hiệu quả là tuyến tính hoá bằng phần mềm cùng với một bộ vi xử lý hoặc là một máy tính PC. Các nhà sản xuất cặp nhiệt điện cũng thường cung cấp kèm theo các công thức để theo đó người sử dụng có thể thực hiện việc tuyến tính hoá.

Khi chưa có điều kiện áp dụng giải pháp triệt để là tuyến tính hoá đường đặc trưng của cặp nhiệt bằng phần mềm thì việc tuyến tính hoá bằng phần cứng vẫn là cách dễ thực hiện nhất. Để tuyến tính hoá thì công việc đầu tiên phải làm là khuếch đại tín hiệu nhiệt điện lên, do tín hiệu này nhỏ và một chiều nên thường phải áp dụng phương pháp đóng ngắt để biến tín hiệu một chiều thành tín hiệu xoay chiều, đôi khi ta còn nói là dùng mạch khuếch đại rung. Thực hiện công việc này bằng các linh kiện rời rạc thường khá phiền phức. Cách đơn giản nhất là sử dụng linh kiện đã được chế tạo để chuyên dùng cho việc đo nhiệt độ bằng cặp nhiệt của hãng Analog Device có tên là AD 584 và AD 595.



Hình 3.29 Sơ đồ khối của bộ khuếch đại cặp nhiệt AD 594/AD595.

Trong một vỏ 14 chân có chứa một bộ khuếch đại dụng cụ để khuếch đại tín hiệu nhiệt điện, ngoài ra nhiệt độ trên vỏ của linh kiện đồng thời là nhiệt độ dầu kẹp cặp nhiệt cũng được đo. Trước khi xuất xưởng linh kiện đã được đồng chỉnh cho cặp nhiệt điện Sắt-Constantan (AD 594) hoặc một cặp Nickel-Chrom/Nickel (AD 595).

Đối với các loại cặp nhiệt điện khác việc tuyến tính hoá trong vi mạch này có thể thực hiện được bằng ba điện trở lắp thêm ở bên ngoài.

Cách tốt nhất để nối cặp nhiệt điện với vi mạch là đấu trực tiếp vào vi mạch này.

Khi ghép nối vi mạch vào mạch điện thì nên chú ý là không nên dùng chân cắm, vì như vậy có khả năng tạo thêm ra các cặp nhiệt điện bổ sung. Vi mạch này được chia ra hai loại với cấp chính xác khác nhau tương ứng với sai số định thang cực đại là $\pm 1\%$ và $\pm 3\%$. Về giá thành thì loại rẻ nhất cũng cỡ 20 đôla. Điện áp nhận được ở lối ra của vi mạch cỡ 10 mV/°C. Một điểm đáng lưu ý khi sử dụng vi mạch này là sự cảnh báo khi đứt cặp nhiệt, bởi vì sự cố này có thể dẫn đến những hậu quả nghiêm trọng, chẳng hạn khi điều chỉnh nhiệt độ có thể dẫn đến sự đốt nóng liên tục cho đến khi tất cả đều bị cháy. Để thông báo tình trạng đứt cặp nhiệt, nhà sản xuất đã tạo ra hai lối ra +ALM và -ALM. Khi cặp nhiệt đứt, các tín hiệu lấy ra từ hai chân này có thể bật điện vào mạch còi báo hiệu hoặc ngắt mạch điện nung nóng, vân vân...

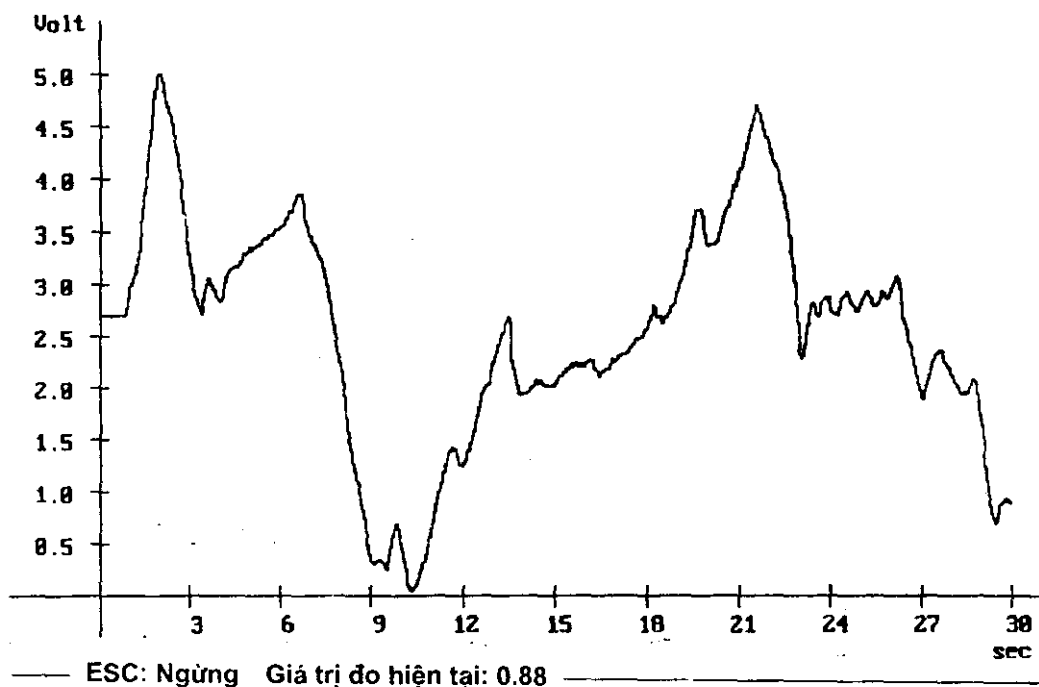
3.6.7 THIẾT BỊ GHI Y(T)

Tất cả các mạch điện đã được giới thiệu từ đầu đến giờ thu thập các thông số quá trình khác nhau qua bộ biến đổi A/D và chuyển tiếp các giá trị digital đến máy tính PC. Bây giờ còn lại vấn đề là người sử dụng muốn quan sát và đánh giá các giá trị đo như thế nào.

Một khả năng mô tả các tín hiệu là dùng thiết bị tự ghi y(t) để biểu diễn sự biến đổi tín hiệu trong một khoảng thời gian xác định, cụ thể là sự biến đổi của biên độ theo trục thời gian. Một ứng dụng điển hình dùng làm thí dụ là mô tả nhiệt độ phòng trong khoảng thời gian là một ngày.

Chương trình dưới đây có tên YTSCHCOM. BAS mô tả một loại thiết bị tự ghi y(t) dưới dạng đơn giản. Đây là một loại thiết bị ảo (virtual). Bằng cách suy nghĩ kỹ lưỡng khi lập trình ta có thể trình bày chương trình một cách thuận tiện hơn và mở rộng thêm những đặc tính khác nữa. Trước tiên chương trình phải mô tả một lối vào và chỉ ra những lệnh nào trong QBasic là cần thiết cho yêu cầu hiển thị. Có rất nhiều lệnh có thể dùng cho đồ họa và cho phép mô tả các giá trị đo một cách có hiệu quả. Bạn đọc có thể tìm hiểu thêm trong nhiều tài liệu tham khảo. Hình 3.29 chỉ ra sự trình bày màn hình của chương trình theo sự diễn biến của một tín hiệu.

THIẾT BỊ TỰ GHI $Y(t)$



Hình 3.30 Sự trình bày của thiết bị tự ghi $y(t)$ trên màn hình.

Chương trình tương ứng có thể được viết như sau:

```

=====
'   Program: YTSCHCOM'
'
'   Function: Chương trình này điều khiển môđun biến
'   đổi A/D bởi sự chuyển đổi dải đo qua cổng nối
'   tiếp của PC (xem thêm hình 3.5) và vẽ các giá trị
'   đo trên đồ thị  $y(t)$ . Thời gian quét có thể được
'   chọn  $> 0,1$  s, tùy ý.
'
'   Hardware: Cần có môđun biến đổi A/D minh hoạ trên
'   hình 3.5
'   cùng với một môđun cơ sở ở mục 1.2
'
=====

```

```
DECLARE SUB lese.com (inbyte)
```

```
SCREEN 12
```

```
'-----Sử dụng các dãy điều khiển sau-----
```

Valence	1	2	4	8	16	32	64	128
(Giá trị)	D0	D1	D2	D3	D4	D5	D6	D7
Khởi động sự biến đổi	1	0/1	x	x	x	Dải đo		
Dải đo 10 V	1	1	x	x	x	0	0	0
Dải đo 5 V	1	1	x	x	x	1	0	0
Dải đo 2 V	1	1	x	x	x	0	1	0
Dải đo 1 V	1	1	x	x	x	1	1	0
Dải đo 0,5 V	1	1	x	x	x	0	0	1
Đón giá trị đo	0/1	1	x	x	x	x	x	x

```
'-----Mở giao diện 9600 baud ở COM2-----
```

```
OPEN "com2: 9600,N,8,1,CS,DS" FOR RANDOM AS #1
```

```
PRINT "Dải đo nào ?..."
```

```
10 V (1), 5 V (2), 2 V (3), 1 V (4), 0,5 V (5)"
```

```
INPUT a
```

```
IF a = 1 THEN mb = 3: Umax = 10 'Dải đo 10 V
```

```
IF a = 2 THEN mb = 35: Umax = 5 'Dải đo 5 V
```

```
IF a = 3 THEN mb = 67: Umax = 2 'Dải đo 2 V
```

```
IF a = 4 THEN mb = 99: Umax = 1 'Dải đo 1 V
```

```
IF a = 5 THEN mb = 131: Umax = .5 'Dải đo 0,5 V
```

```
PRINT "Cho vào thời gian quét:"
```

```
PRINT "Thí dụ nạp vào: 0.1 đối với 0,1 giây"
```

```
PRINT "1.5 đối với 1,5 giây"
```

```
PRINT "25 đối với 25 giây"
```

```
INPUT Scantime
```

CLS

a\$ = "Y (t) - W R I T E R"

a = LEN (a\$): b = (80 - a) / 2 - 1

LOCATE 1, b: PRINT CHR\$ (201);

STRING\$ (a + 2, CHR\$ (205)); CHR\$ (187)

LOCATE 2, b: PRINT CHR\$ (186)

LOCATE 2, b + 2: PRINT a\$

LOCATE 2, a + b + 2: PRINT CHR\$ (186)

LOCATE 3, b: PRINT CHR\$ (200);

STRING\$ (a + 2, CHR\$ (205)); CHR\$ (188)

-----Định nghĩa các biến -----

xmax = 600 'Số giá trị max theo hướng x

ymax = 255 'Biên độ cực đại tương ứng

y1 = .02 * ymax 'Dành cho việc chia thang

x1 = .01 * xmax 'như trên

xneg = .1 * xmax 'Độ dài của đoạn trục âm

yneg = .1 * ymax 'như trên

-----Khẳng định hệ tọa độ -----

WINDOWS (-xneg, -2.3 * yneg)-

(1.05 * xmax, 1.15 * 1.1 * ymax)

-----Vẽ các trục tọa độ-----

LINE (-xneg, 0) - (1.1 * xmax, 0)

LINE (0, -yneg) - (0, 1.1 * ymax)

-----Chia thang và đánh số-----

FOR i = 1 TO 10

LINE (-x1, i * ymax / 10) - (x1, i * ymax / 10)

LINE (i * xmax / 10, -y1) - (i * xmax / 10, y1)

```

LOCATE 26 - 2 * i, 1
PRINT USING "###.#"; umax / 10 * i
LOCATE 4, 3: PRINT "Volt"
LOCATE 27, 11 + 7 * (i - 1)
PRINT USING "#####"; i * xmax / 10 * Thời gian quét;
NEXT i
LOCATE 28, 20: PRINT "Giá trị đo hiện thời:"
LOCATE 28, 76: PRINT "sec."
LOCATE 28, 1: PRINT "ESC: Ngừng lại"
'
'-----Vòng lặp để đọc vào giá trị đo-----
'
FOR i = 0 TO xmax
oldtime = TIMER                'oldtime = Thời điểm t
newtime = oldtime + scantime
                                'newtime = t + thời gian quét
flag = 0
'
'-----Bắt đầu vòng lặp theo thời gian-----
'
DO
curtime = TIMER                'curtime = thời điểm hiện tại
IF curtime >= newtime THEN EXIT DO
IF flag = 0 THEN
'
'-----Bắt đầu phép đo-----
'
PRINT #1, CHR$ (mb - 2);
PRINT #1, CHR$ (mb + 2);
'
'-----Đón nhận giá trị đo-----
'
PRINT #1, CHR$ (mb - 1);
PRINT #1, CHR$ (mb + 1);
CALL lese.com (new.value)

```

```

LOCATE 28,40:PRINT USING "##.##"; new.value/256* ymax
new.value = new.value / 256 * ymax
'
'-----Điền các giá trị vào hệ tọa độ-----
'
LINE (i, old.value) - (i, new.value)
old.value = new.value
flag = 1
END IF

IF INKEY$ = CHR$(27) THEN EXIT FOR

```

Khả năng làm ngừng bằng phím ESC

LOOP

NEXT i

CLOSE 1

END

SUB lese.com (inbyte)

```

'=====
'  Unterprogram: lese.com
'
'  Function: Chương trình này đọc một byte qua giao
'  diện nối tiếp. Nếu như sự truyền bị nhiễu thì một
'  dòng thông báo sẽ nhấp nháy và chương trình kết
'  thúc
'  =====
  i = 0
  DO
    i = i + 1
    '
    '-----Khi Byte đã có mặt loc(1) >= 1-----
    '
    IF LOC (1) >= 1 THEN
      inS = INPUT$(1, #1)
    
```

```
inbyte = ASC (in$)
```

```
GOTO ketthuc
```

```
END IF
```

```
,
```

```
'-----Phép thử mới, đọc vào dữ liệu -----
```

```
,
```

```
LOOP UNTIL i = 1000 'Cực đại là 1000 phép thử
```

```
,
```

```
'-----Không nhận được Byte nào-----
```

```
,
```

```
CLS
```

```
PRINT "Sự truyền dữ liệu bị nhiễu !!!!!!"
```

```
PRINT
```

```
PRINT "Không nhận được ký tự nào !!!!!!"
```

```
PRINT
```

```
PRINT "Hãy kiểm tra lại ổ cắm, cáp nối, phần  
cứng, ..."
```

```
END
```

```
ketthuc:
```

```
END SUB
```



```
-----
```

Chương 4

CÁC MẠCH ĐIỀU KHIỂN VỚI BỘ BIẾN ĐỔI D/A

Để điều khiển một quá trình đo đạc bằng máy tính PC, để cho trước một giá trị nên có hoặc là để sử dụng máy tính như một bộ điều khiển số thì ta cần có được khả năng xuất ra những giá trị analog (tương tự) như dòng điện và điện áp với độ lớn cụ thể được định trước.

Để đạt được mục đích này thường là phải nghĩ ngay đến các bộ biến đổi D/A. Từ 2^n giá trị số (digital) có thể bộ biến đổi D/A sẽ tạo ra 2^n điện áp lối ra khác nhau; các điện áp này được lấy ra từ một điện áp so sánh xác định. Nguồn điện áp so sánh có thể tìm thấy hoặc chính ở bên trong bộ biến đổi D/A hoặc có thể từ một nguồn điện áp so sánh bên ngoài. Bộ biến đổi D/A có thể cho các lối ra là nguồn dòng hoặc nguồn điện áp.

Độ phân giải được chỉ ra như là độ rộng của giá trị số được biến đổi. Vì thế, một bộ biến đổi D/A n bit có thể tạo ra 2^n giá trị lối ra khác nhau.

Khi ta chọn dải điều chỉnh có độ rộng 10 V như thường thấy trong công nghiệp, thì sẽ có những bước nhảy điện áp nhỏ nhất như sau:

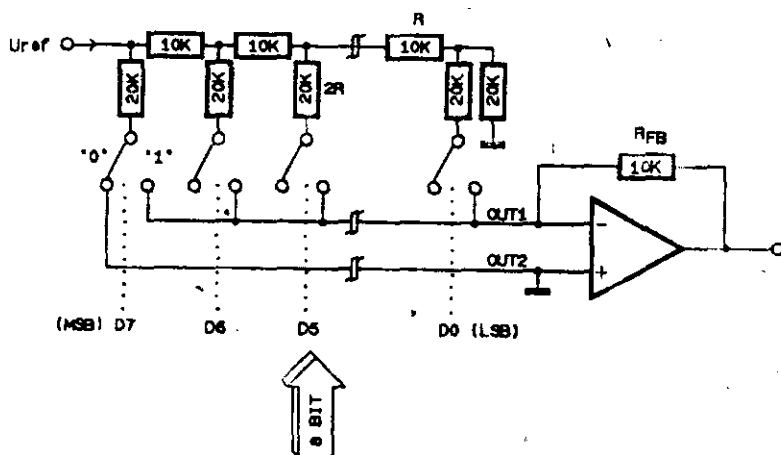
n	Độ phân giải	Điện áp nhỏ nhất
8	1 / 256	39,1 mV
10	1 / 1.024	39,1 mV
12	1 / 4.096	39,1 mV
16	1 / 65.536	39,1 mV

4.1 CÁCH HOẠT ĐỘNG CỦA BỘ BIẾN ĐỔI D/A

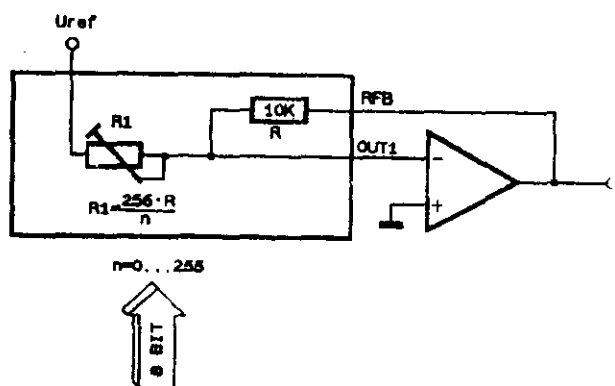
Phương pháp biến đổi được dùng rộng rãi nhất ở các bộ biến đổi D/A là sử dụng một mạng các điện trở R - 2R (xem hình 4.1). Sự sắp xếp khá đặc biệt của các điện trở đã mang lại những ưu điểm nổi bật so với các phương pháp khác. Phần chính của mạng các điện trở có thể xem như là một bộ chia điện áp. Bộ chia này có đặc tính là mỗi điểm nút được đấu tải bằng một điện trở R. Nhờ vậy mà ở mỗi điểm nút dòng điện đi qua được chia theo tỷ lệ 1:1 và đối với bit cao nhất dòng đi qua điện trở được tính bằng biểu thức:

$$I = U_{\text{ref}} / 2R$$

Còn qua điện trở tiếp theo sẽ bằng: $I = U_{\text{ref}} / 2R \cdot 0,5$; vân vân...



Sơ đồ tương đương



Hình 4.1 Cách đấu điện trở R-2R ở bộ biến đổi D/A.

Một bộ chuyển mạch qua lại sẽ xác định liệu dòng điện sẽ đi xuống mass hay là đi qua điểm lấy tổng của mạch. Mức High đặt ở chuyển mạch sao cho dòng điện đi qua điểm lấy tổng và do vậy đóng góp một phần vào dòng điện tổng cộng.

Dòng điện tổng cộng ở lối ra OUT1 sau đó được tính theo biểu thức:

$$I = U_{\text{ref}} \cdot n / 256 \cdot R$$

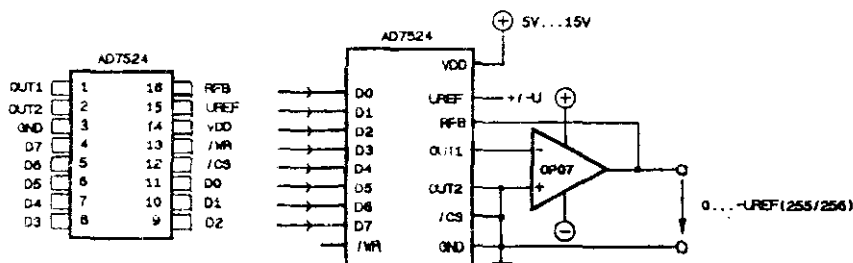
Ở đây n là giá trị của byte dữ liệu (kể sát lối ra). Khi ta nối chân ra OUT 1 với lối vào không đảo của một bộ khuếch đại thuật toán có điện trở R được đấu như điện trở phản hồi thì điện áp lối ra được tính như sau:

$$U_a = - I \cdot R = - U_{\text{ref}} \cdot n / 256$$

Trong các bộ biến đổi A/D chuyên dụng thì điện trở phản hồi được tích hợp ngay ở trên chip của vi mạch. Ngược lại, chỉ trong một số ít bộ biến đổi A/D bộ khuếch đại thuật toán được cùng tích hợp trên chip nên sau đó phải lắp ráp thêm ở bên ngoài. Trên hình 4.1 cũng đồng thời minh họa sự phân bố linh kiện trên mạch tích hợp.

Trong mạch điện này, giá trị tuyệt đối của điện trở R là không quan trọng, và giá trị của R có thể dao động trong khoảng rộng. Khi chú ý tới độ chính xác của quá trình biến đổi thì ta thấy chỉ có việc tạo cặp của các điện trở đóng một vai trò quyết định. Ngày nay, với trình độ phát triển của công nghệ cao thì yêu cầu này có thể được thực hiện với một độ chính xác rất cao.

4.2 BỘ BIẾN ĐỔI D/A 8 BIT AD 7524



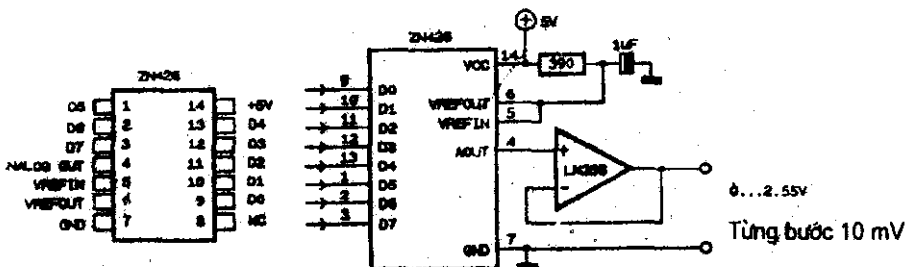
Hình 4.2 Bố trí chân và sơ đồ mạch của bộ biến đổi D/A AD 7524.

Bộ biến đổi AD 7524 làm việc theo nguyên tắc vừa được trình bày. Điện trở R của mạng điện trở cỡ 10 k Ω . Hình 4.2 chỉ ra sự sắp xếp chân và một mạch điện điển hình của bộ biến đổi.

Các lối ra OUT 1 và OUT 2 là các điểm chung của dòng điện. Dòng tiêu thụ của vi mạch AD 7524 nằm trong khoảng của những linh kiện chế tạo theo công nghệ CMOS, cụ thể là chỉ cỡ 1 mA khi điện áp nguồn là + 5 V. Điện áp nguồn ở chân ra VDD có thể được lựa chọn trong khoảng 5 V đến 15 V. Tất cả các lối vào đều tương thích TTL. Các chân ra CS (Chip Select) và WR (Write) khi ghi vào một byte dữ liệu cần phải có mức Low. Sau đó, chân WR chuyển trở lại mức High và byte dữ liệu được lưu trữ lại trong bộ biến đổi D/A. Khi sử dụng vi mạch AD 7524 thì điều đặc biệt đáng quan tâm là ở chân U_{REF} (điện áp so sánh) có thể cho phép đấu vào cả điện áp dương cũng như điện áp âm. Thậm chí điện áp này cũng có thể là một điện áp xoay chiều, có biên độ thay đổi trong phạm vi một giá trị giới hạn.

4.3 BỘ BIẾN ĐỔI D/A 8 BIT ZN 426

Bộ biến đổi D/A có tên ZN 426 là một linh kiện khá đắt tiền và có một nguồn điện áp so sánh bên trong với giá trị 2,55 V. Vi mạch này hoạt động cũng theo phương pháp mạng điện trở, nhưng với lối vào và lối ra có thể hoán đổi cho nhau. Từ hình 4.1 suy ra là chân ra OUT 1 của vi mạch ZN 426 được nối với lối vào điện áp so sánh, chân ra U_{ref} tương ứng với lối ra analog (tương tự). Theo cách này, tầng khuếch đại đảm nhận việc lấy tổng. Nhược điểm của phương pháp này là sự chịu tải không đồng đều của điện áp so sánh và có mức điện áp cao ở chỗ chuyển mạch.



Hình 4.3 Bố trí chân và sơ đồ mạch của bộ biến đổi D/A ZN 426.

Hình 4.3 chỉ ra sự sắp xếp chân ra và mạch điện sử dụng vi mạch ZN 426

Khác với vi mạch AD 7524, điện áp so sánh bên ngoài chỉ được nhận giá trị dương nằm trong khoảng 0 và 13 V. Ngoài ra các lối vào điều khiển phụ như CS và WR để nối với hệ thống bus hoàn toàn không thấy có. Việc lựa chọn vi mạch ZN 426 cho những ứng dụng đơn giản là hoàn toàn đúng đắn.

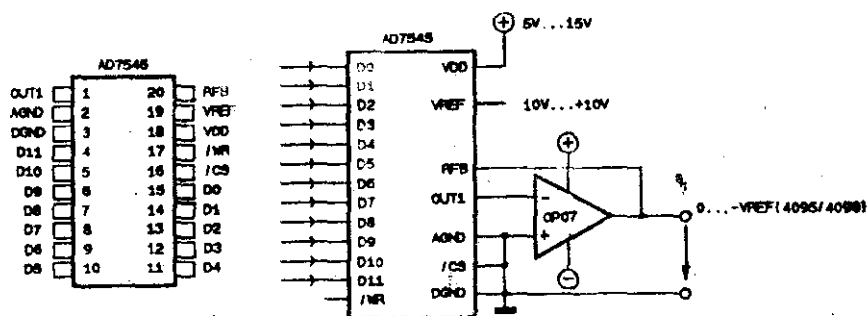
Phương pháp hoán đổi lối vào và lối ra cũng có thể được sử dụng cho vi mạch AD 7524.

Dưới đây là các thông số của vi mạch ZN 426

- ☐ Tương thích với TTL và CMOS 5 V.
- ☐ Nguồn nuôi đơn giản + 5 V.
- ☐ Chỉ cần đấu thêm điện trở và tụ điện phụ.
- ☐ Dòng tiêu thụ (max): 9 mA.

4.4 BỘ BIẾN ĐỔI D/A 12 BIT AD 7545

Hình vẽ dưới đây chỉ ra cách bố trí chân ra của vi mạch biến đổi D/A AD 7545



Hình 4.4 Bố trí chân và sơ đồ mạch của bộ biến đổi D/A AD 7545.

Cách làm việc của vi mạch AD 7545 hoàn toàn giống như ở bộ biến đổi D/A AD 7524 đã mô tả ở mục 4.2. Sự khác nhau duy nhất là ở chỗ số các bit dữ liệu đã được mở rộng. Với 12 bit ta luôn có thể thiết lập 4095 bậc điện áp, trong khi ở bộ biến đổi 8 bit chỉ có thể đặt 255 bậc. Các

đường dẫn dữ liệu truy nhập trực tiếp tới linh kiện. Nhìn vào sự sắp xếp các chân ra, ta thấy là bộ biến đổi D/A có hai chân nối mass: một mass cho mạch analog (AGND) và một mass cho mạch digital (DGND). Đặc điểm này cần được chú ý tới khi tiến hành thiết kế mạch in.

Khi ta muốn xử lý các điện áp trong dải milivon (mV) thì việc tiếp mass (nối với vỏ máy, đôi khi là sự nối đất) một cách thận trọng là không cần thiết. Đối với quá trình biến đổi D/A thì điện áp so sánh là AGND.

Các chân ra AGND và DGND cần phải được nối chung lại chỉ ở một điểm của mạch điện. Chỗ thích hợp nhất là nối trực tiếp vào nguồn nuôi. Trong trường hợp lý tưởng ta dùng một nguồn nuôi riêng cho phần digital và một nguồn nuôi cách ly khác cho phần analog.

Dưới đây là các thông số kỹ thuật của vi mạch AD 7545:

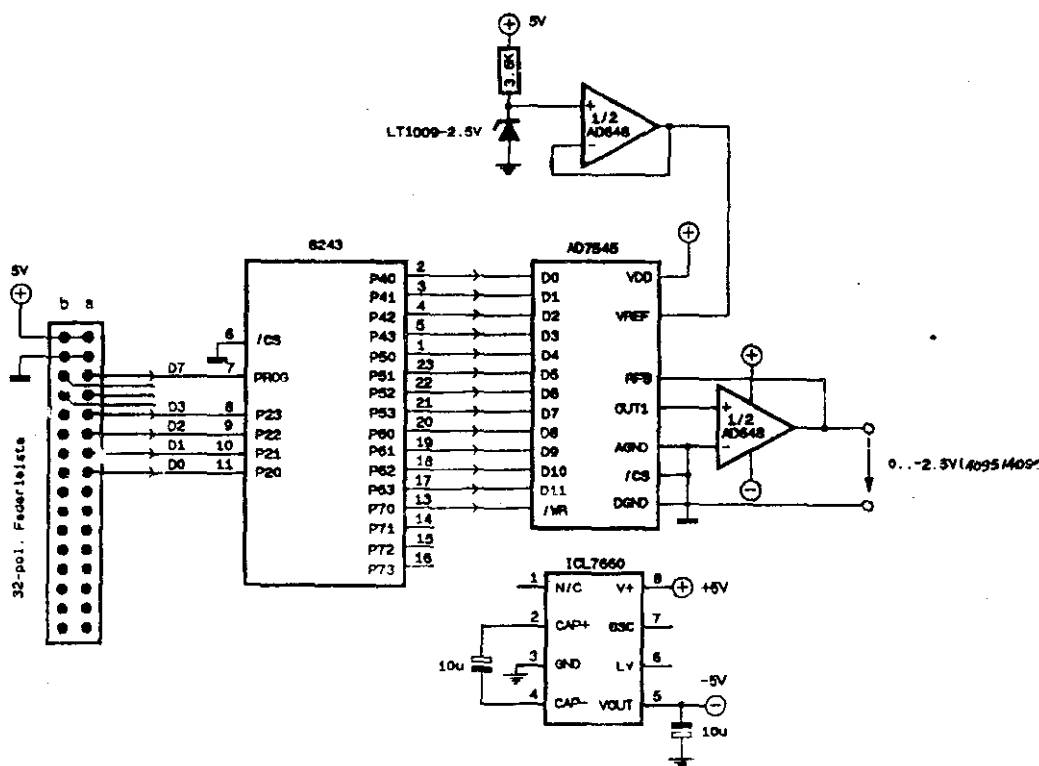
- ☐ Độ phân giải 12 bit.
- ☐ Điện áp nguồn nuôi: + 5 V đến 15 V.
- ☐ Dòng tiêu thụ (max): 2 mA.
- ☐ Điện trở lối vào ở V_{REF} : 11 k Ω (tiêu chuẩn).
- ☐ Thời gian biến đổi: 2 μ s.
- ☐ Giá cỡ 10 USD (năm 1996).

4.5 NHỮNG ỨNG DỤNG CỦA BỘ BIẾN ĐỔI D/A

4.5.1 MÔĐUN BIẾN ĐỔI D/A 12 BIT

Ở chương 1, chúng tôi đã giới thiệu các mạch điện của nhiều môđun cơ sở có 8 lối vào và lối ra TTL có thể đấu vào cả giao diện nối tiếp cũng như vào giao diện song song. Để điều khiển một bộ biến đổi D/A 12 bit môđun cơ sở còn thiếu 4 đường dẫn lối ra. Bằng việc sử dụng vi mạch 8243 mà ta đã tìm hiểu kỹ ở mục 2.4 vấn đề này hoàn toàn được giải quyết. Bây giờ có 16 lối ra TTL có thể được sử dụng như dưới đây. 12 đường dẫn lối ra dẫn trực tiếp đến các đường dẫn dữ liệu của bộ biến đổi D/A. Một lối ra điều khiển lối vào WR, mà qua lối vào này bộ biến đổi D/A lưu trữ các dữ liệu đang đứng xếp hàng. Muốn thế cần có một mức Low ở chân ra này.

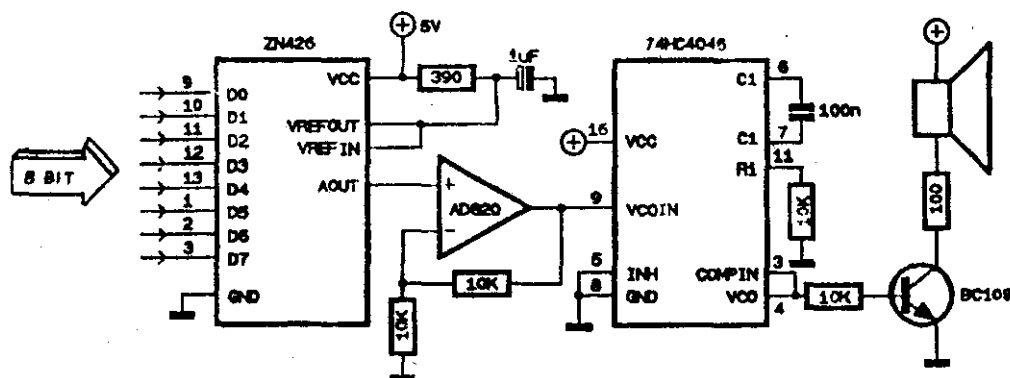
Điện áp so sánh được dẫn đến chân ra V_{ref} qua một tầng lặp lại điện áp. Ở nguồn điện áp so sánh trên hình 4.5 vi mạch ổn áp có độ chính xác cao LT 1009-2,5 V đã được lựa chọn (độ chính xác đạt đến 0,2 %). Điện áp lỗi ra của bộ biến đổi D/A được tạo nên qua một bộ khuếch đại thuật toán làm việc như một bộ khuếch đại đảo. Muốn thế cần có một nguồn nuôi với điện áp âm lấy từ phía vi mạch ICL 7660. Khi điều khiển các môđun có bộ biến đổi D/A cần phải chú ý rằng trước hết phải đặt chân WR lên mức High. Sau đó 12 bit có thể được đặt qua các cổng 4, 5 và 6 vào bộ biến đổi D/A. Bằng một xung nhọn Low ở chân WR các dữ liệu trong bộ biến đổi AD 7545 được lưu trữ và giữ điện áp lỗi ra ở mức tương ứng. Khi chân ra WR thường xuyên được giữ ở mức Low, thì sẽ xuất hiện các bước nhảy điện áp ở lỗi ra của bộ khuếch đại khi ghi các dữ liệu vào. Vì thế các cổng 4, 5 và 6 không thể được ghi một cách đồng thời.



Hình 4.5 Sơ đồ mạch của môđun biến đổi D/A 12 bit.

4.5.2 BỘ PHÁT CÒI BÁO HIỆU ĐIỀU KHIỂN BẰNG MÁY VI TÍNH

Trên hình 4.6 chỉ ra một bộ phát còi báo hiệu đơn giản có thể điều khiển được từ máy tính PC.



Hình 4.6 Bộ phát còi báo hiệu điều khiển bằng máy vi tính.

Phần chính của mạch điện này bên cạnh bộ biến đổi D/A là một linh kiện PLL có tên 74HC4046. Vì mạch này đã được sử dụng ở bộ tổng hợp tần số được trình bày trong mục 2.6.4.13. Phần quan trọng nhất của vi mạch này là một bộ so pha và một bộ dao động được điều khiển bằng điện áp (VCO, viết tắt từ Voltage Controlled Oscillator). Bộ VCO có lẽ là linh kiện có tính vạn năng nhất và thuộc vào loại đắt tiền nhất trên thị trường linh kiện. Nó tạo ra một điện áp có dạng vuông góc rõ nét nhất và có thể tinh chỉnh qua một điện áp một chiều trong một dải rộng.

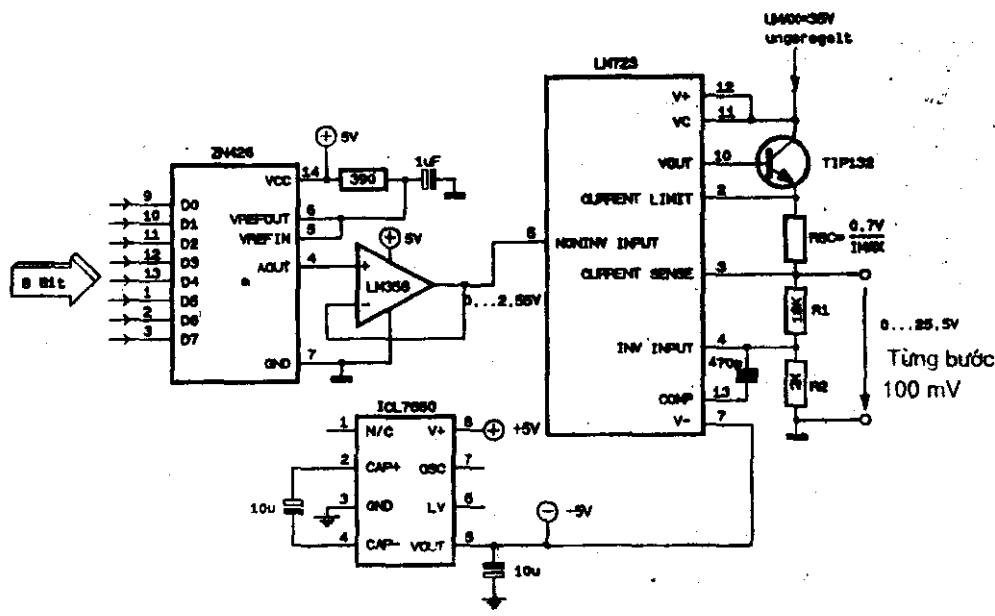
Trong bộ phát còi báo hiệu điều khiển bằng máy tính PC chỉ có linh kiện VCO là đáng quan tâm nhất. Tần số dao động được xác định bởi một tụ điện đấu giữa chân 6 và chân 7, qua một điện trở (có giá trị nhỏ nhất cho phép là 10 k Ω) từ chân 11 xuống mass và điện áp ở chân 9. Điện áp được cho trước từ máy tính PC qua bộ biến đổi D/A ZN 426 và có thể chấp nhận một tiến trình nào đó. Phù hợp với các tiến trình đó ta có thể đạt tới các hiệu ứng tiếng động rất khác nhau.

Điện áp lỗi ra của bộ biến đổi D/A được khuếch đại lên 2 lần. Bộ khuếch đại thuật toán AD 820 có tính chất đặc biệt là điện áp lỗi ra có thể đạt đến 0 V và đến điện áp nguồn nuôi dương. Các thông số kỹ thuật khác bạn đọc có thể tìm thấy trong phụ lục 1 ở cuối sách.

Tín hiệu vuông góc từ bộ VCO được đưa đến một tranzito để điều khiển tải của nó là một loa. Điện trở $100\ \Omega$ có thể được thay thế bằng một chiết áp tinh chỉnh và nhờ thế có thể đặt âm lượng cho bộ phát còi báo hiệu này.

4.5.3 BỘ NGUỒN LẬP TRÌNH ĐƯỢC

Gần đây các bộ nguồn nuôi điện áp thấp đã được thay thế dần hoặc lắp đặt mới bằng những bộ nguồn có thể lập trình được. Giá trị của điện áp (hoặc cả dòng điện) được đặt một cách gián tiếp qua những phím nhấn để có được những giá trị rất chính xác. Cũng có loại có thể ghép nối với máy tính PC để từ đó có thể đặt giá trị điện áp. Ta có thể tìm hiểu các mạch điện loại này qua thí dụ sau đây.



Hình 4.7 Sơ đồ mạch của bộ nguồn lập trình được.

Một bộ biến đổi D/A có thể được ghép nối với vi mạch ổn áp loại LM 723 để tạo nên một bộ nguồn nuôi. Sau đó điện áp lối ra đã ổn định lại có thể được thiết lập một cách rất thuận lợi từ máy tính PC. Và ngay cả diễn biến theo thời gian của điện áp được lập trình trước cũng có thể được thực hiện theo cách này. Hình 4.7 chỉ ra mạch điện của một nguồn nuôi như vậy.

Trong mạch này cũng sử dụng vi mạch ZN 426 như là một bộ biến đổi D/A. Bằng điện áp so sánh bên trong, điện áp lỗi ra ở chân 4 có thể đặt được trong phạm vi giữa 0 và 2,55 V. Bộ lặp lại điện áp LM 358 nối với lỗi ra analog và dẫn tín hiệu đến chân 5 của bộ ổn áp LM 723. Vì mạch này chuyên dùng để lắp ráp các bộ nguồn và bên trong có chứa tất cả các linh kiện cần thiết. Điện áp cho phép cực đại ở chân V_+ so với điện áp V_- cỡ 42 V. Chỉ dẫn này rất quan trọng, nếu như ta muốn ổn định điện áp ra đến 25 V. Với một biến áp có điện áp thứ cấp là 25 V, ta nhận được ở bộ chỉnh lưu trong chế độ không tải một điện áp một chiều cỡ $\sqrt{2} \times 25 \text{ V} = 35 \text{ V}$. Cách ước tính này thường được thực hiện khi thiết kế các bộ nguồn.

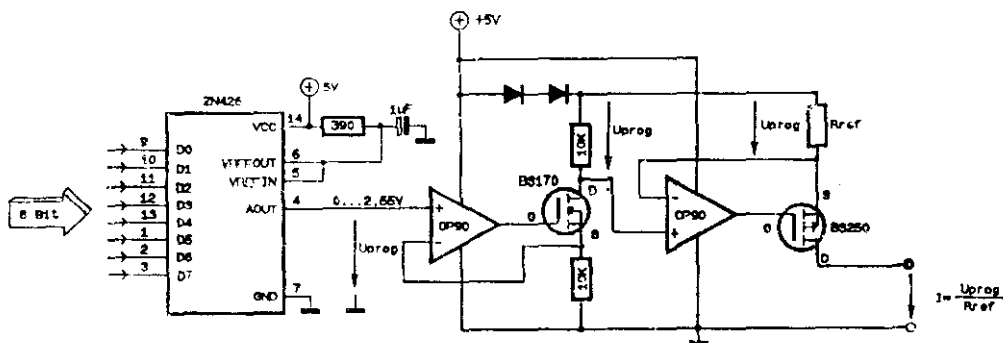
Vi mạch ổn áp LM 723 có chứa một bộ khuếch đại vi phân với các lối vào không đảo (NONINV INPUT) và đảo (INV INPUT). Vi mạch điều chỉnh điện áp lối ra V_{out} sao cho giá trị điện áp ở hai lối vào của tầng khuếch đại vi phân có cùng giá trị (điện áp lối vào bằng 0). Do tỷ số giá trị điện trở $R_1/R_2 = 9 : 1$ nên điện áp lối ra ở chân 3 đúng bằng 10 lần điện áp ở chân 5. Theo cách này ta đạt được giá trị điện áp ở lối ra (chân 3) của bộ nguồn nằm trong khoảng 0 đến 25,5 V.

Một bit tương ứng với một sự thay đổi điện áp cỡ 100 mV. Tranzito TIP 102 hoạt động theo nguyên tắc Darlington có nhiệm vụ khuếch đại dòng lối ra của vi mạch LM 723. Điện trở RSC xác định dòng điện cực đại cho phép, có thể lấy ra từ bộ nguồn này. Khi có một sụt áp cỡ 0,7 V trên điện trở này thì mạch bảo vệ chống quá điện áp ở bên trong hoạt động và dòng tiêu thụ lập tức bị hạn chế lại. Trong cách lắp ráp mạch cũng nên chú ý là chân 7 của vi mạch LM 723 có tên V_- được nối với nguồn điện áp âm - 5 V, và do vậy mà điện áp lối ra có thể đạt được giá trị 0 V. Nếu ta bỏ đi vi mạch ICL 7660 và đặt chân ra V_- trực tiếp xuống mass (0 V) thì điện áp lối ra bắt đầu đặt được từ 2 V.

4.5.4 NGUỒN DÒNG LẬP TRÌNH ĐƯỢC

Các nguồn dòng cần bảo đảm cung cấp cho một mạch tiêu thụ một dòng điện độc lập với điện áp đặt lên mạch tiêu thụ. Trong mục 3.6.3 nguyên tắc này đã được sử dụng để đo điện trở. Hình 4.8 chỉ ra một

mạch điện trong đó dòng điện không đổi có thể được thiết lập qua điện áp lỗi ra của một bộ biến đổi D/A.



Hình 4.8 Sơ đồ mạch của bộ nguồn dòng lập trình được.

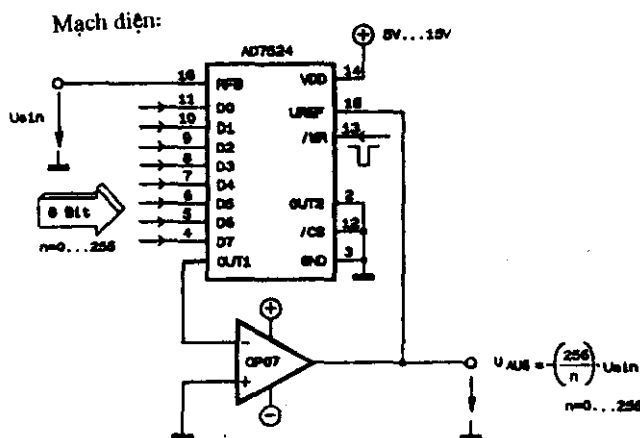
Mạch điện này hoạt động như thế nào ?

Ở lối vào không đảo của bộ khuếch đại thuật toán thứ nhất có đặt điện áp lỗi ra U_{prog} của bộ biến đổi D/A. Điện áp này đồng nhất với điện áp ở lối vào đảo và tương ứng với sụt áp trên điện trở ở 10 kΩ ở chân ra S (cực nguồn) của tranzito BS 170. Do ở cực máng D cũng có đấu một điện trở có cùng giá trị nên sụt điện áp ở đây cũng đúng bằng U_{prog} . Điều này có nghĩa là điện áp trên điện trở R_{ref} cần phải tương ứng đúng với giá trị này. Cả hai bộ khuếch đại thuật toán cùng với tranzito BS 170 thực hiện nhiệm vụ chuyển chỗ điện áp U_{prog} giữa lối ra của bộ biến đổi D/A với mass sao cho ở trên điện trở R_{ref} ta nhận lại đúng điện áp U_{prog} . Điều đó sẽ cho phép có được một dòng điện có giá trị không đổi đi qua tranzito BS 250 (xem trên hình 4.8).

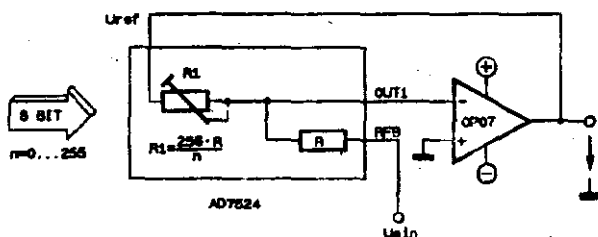
Hai diốt trong mạch có ý nghĩa đặc biệt và đóng vai trò quan trọng đối với chức năng của mạch. Điện áp lỗi ra của bộ biến đổi D/A cỡ 40 mV. Điều này có nghĩa là ở lối vào không đảo của tầng khuếch đại thuật toán thứ hai có một điện áp bằng: $5\text{ V} - 2 \times U_{diốt} - 40\text{ mV} = 3,6\text{ V}$ được thiết lập. Đây vẫn chưa phải là điều không bình thường. Nhưng không có các diốt thì điện áp này sẽ có giá trị đúng bằng 4,96 V, và giá trị này nằm ngoài vùng lối vào của tầng khuếch đại thuật toán.

4.5.5 BỘ ĐẶT ĐỘ KHUẾCH ĐẠI LẬP TRÌNH ĐƯỢC

Hình 4.9 dưới đây chỉ ra một ứng dụng đáng quan tâm của bộ biến đổi D/A được chế tạo bằng công nghệ CMOS và có tên là AD 7524.



Sơ đồ tương đương:



Hình 4.9 Bộ đặt độ khuếch đại lập trình được.

Trong ứng dụng này, các giá trị khác nhau của hệ số khuếch đại có thể được thiết lập qua bộ biến đổi D/A AD 7524. Việc lắp ráp nối bộ biến đổi D/A là không bình thường, bởi vì điện áp lối vào được dẫn đến chân ra RFB. Xem xét trên hình vẽ ta dễ dàng thấy được cách hoạt động của mạch. Mạch điện tương ứng với một tầng khuếch đại đảo, trong đó giá trị của điện trở phản hồi R_1 có thể đặt được qua bộ biến đổi D/A. Điện trở lối vào tương ứng với giá trị của điện trở R , mà ở bộ biến đổi D/A AD 7524 giá trị này cỡ 10 k Ω .

Công thức để tính toán hệ số khuếch đại có dạng

$$v = - 256 / n$$

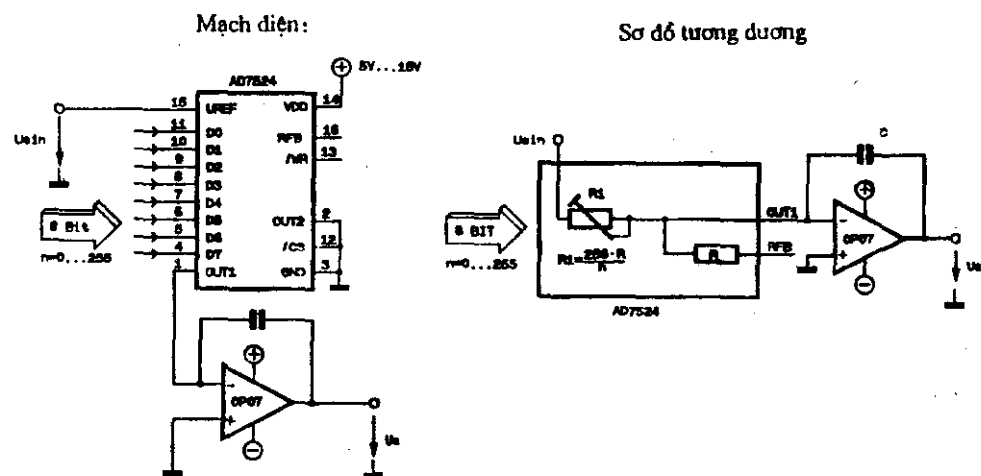
trong đó n là giá trị của các byte dữ liệu ở vi mạch AD 7524 và nằm giữa 0 và 255. Khi ta đặt độ khuếch đại lên hết cỡ ($n = 0$) thì hệ số khuếch đại lớn nhất có thể đặt được là $v = 256$. Giá trị nhỏ nhất có thể đặt được là $v = 256 / 255 = 1,0039$.

4.5.6 MẠCH TÍCH PHÂN LẬP TRÌNH ĐƯỢC

Một ứng dụng khác trong đó bộ biến đổi D/A loại CMOS được sử dụng như một điện trở có giá trị đặt được sẽ được trình bày dưới đây.

Trong ứng dụng này bộ biến đổi D/A được ghép nối với một bộ khuếch đại thuật toán và đóng vai trò như một điện trở trong mạch tích phân (so sánh với hình 4.10).

Trên mạch phản hồi của khuếch đại thuật toán là một tụ điện, tụ này được nạp với dòng điện $I = U_e / R_1$. Qua bộ điện trở R_1 đặt được thành 255 bậc. Chân ra RFB ở vi mạch AD 7524 ở đây giữ nguyên không được dùng đến.

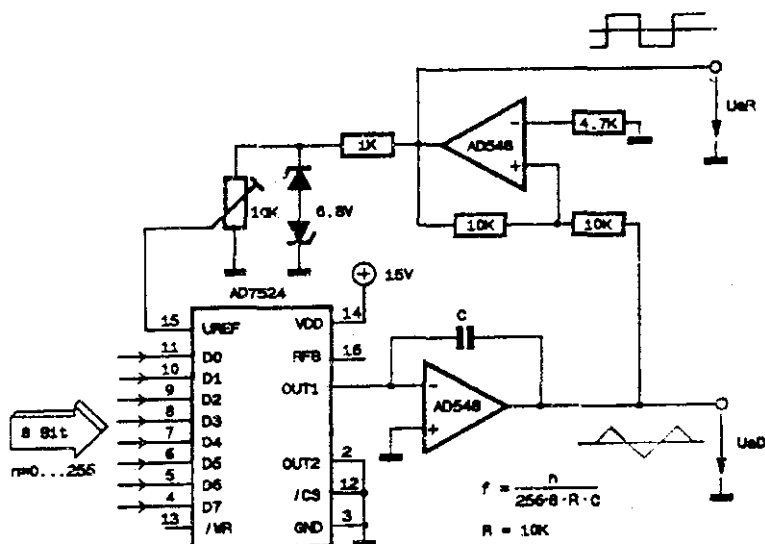


Hình 4.10 Mạch tích phân lập trình được.

Tất nhiên mạch này cũng có một nhược điểm là giá trị chính xác của điện trở R_1 chỉ có thể biết được bằng kỹ thuật đo, còn từ tài liệu kỹ thuật chỉ có thể nhận được giá trị gần đúng. Nhưng ta cũng phải thấy rõ

là trong mạng điện trở $R - 2R$ đối với quá trình biến đổi D/A, độ không chính xác của điện trở R không có ý nghĩa, mà chỉ có tỷ số $2R$ trên R cần phải có giá trị đúng bằng 2. Ngoài ra, mạch trong hình 4.10 có thể được sử dụng rất thích hợp trong các mạch lọc. Qua đó ta có thể thay đổi tần số giới hạn của một bộ lọc tích cực. Chẳng hạn như trong mạch tạo tiếng dây đàn ghita ở hình 3.27 thì tần số cộng hưởng của dải thông có thể đặt được từ máy tính PC. Sau đây điện trở R_2 cần phải được thay thế bằng bộ biến đổi A/D.

4.5.7 BỘ TẠO HÀM LẬP TRÌNH ĐƯỢC



Hình 4.11 Bộ tạo hàm lập trình được.

Trên hình 4.11 là sơ đồ điện của bộ tạo hàm (function generator) với hai loại điện áp lối ra: thứ nhất là tín hiệu vuông góc còn thứ hai là tín hiệu hình tam giác. Tần số của các tín hiệu này có thể đặt được bằng bộ biến đổi D/A AD 7524. Ứng dụng này có nét không bình thường. Trước hết ta cần hiểu rõ xem tần số của tín hiệu lối ra có thể được thiết lập bằng một bộ biến đổi D/A như thế nào. Câu trả lời đã gần như có được, khi ta xem xét kỹ sơ đồ trên hình 4.10. Mạch tích phân được mô tả ở đó có một ý nghĩa đặc biệt đối với bộ tạo hàm lập trình được. Bộ biến đổi A/D hoạt động như một điện trở đặt được và được nạp điện qua tụ

điện C. Bởi vì lối ra OUT 1 có một điện áp gần như điện thế mass nên điện áp lối ra U_{AD} phù hợp đúng với điện áp trên tụ điện nhưng với dấu đã bị thay đổi.

Tín hiệu điện áp hình tam giác được truyền đến bộ khuếch đại thuật toán thứ hai, ở đây bộ khuếch đại này hoạt động như một bộ so sánh điện áp. Khi điện áp đạt đến giá trị điện áp ngưỡng của bộ so sánh điện áp thì tín hiệu lối ra của nó lật từ giá trị cực đại vừa xuất hiện sang một giá trị cực trị khác nhưng với dấu ngược lại.

Sự đảo chiều của điện áp dẫn đến một hậu quả là điện áp lối vào ở U_{ref} cũng đảo chiều. Tụ điện C sau đó được nạp với dấu đã được đặt ngược lại. Một điện áp hình tam giác xuất hiện, còn ở lối ra của bộ khuếch đại thuật toán, mà đang làm việc như bộ so sánh điện áp, ta nhận được một điện áp lối ra có dạng vuông góc. Trong mạch điện này, bộ biến đổi D/A điều khiển dòng một chiều (không đổi) nạp cho tụ điện C. Giá trị của các byte dữ liệu gần kề càng lớn thì dòng điện nạp này cũng càng lớn. Do đó tần số của hai tín hiệu điện áp ở lối ra cũng càng lớn.

Khi lựa chọn khuếch đại thuật toán nên chú ý là tốc độ tăng điện áp (slew rate) của vi mạch có một vai trò rất quan trọng. Với một giá trị $1 \text{ V}/\mu\text{s}$ thì thời gian cần thiết để tăng từ -10 V lên $+10 \text{ V}$ chỉ cỡ 20 micro giây.

4.5.8 MÔĐUN ĐỂ VẼ ĐƯỜNG ĐẶC TRƯNG

Trên hình 4.12 chỉ ra mạch điện của một máy đo có khả năng vẽ các đường đặc trưng (họ đặc tuyến) khi sử dụng cùng với môđun cơ sở đã trình bày ở chương 1. Chẳng hạn ta muốn dùng để đo đường đặc trưng Von-Ampe của một điốt. Khi đó ta sẽ phải đặt lên điốt những điện áp khác nhau và đo dòng điện tương ứng qua điốt. Chắc chắn là một số bạn đọc đã từng làm bài thực tập này trong giờ học về linh kiện bán dẫn. Khi ghi lại các giá trị dòng điện và điện áp tương ứng ta có thể vẽ ra đường đặc trưng Von-Ampe. Bảng mạch điện được giới thiệu dưới đây thì tất cả các công việc nêu trên có thể tiến hành trên máy tính PC.

Điều đó có nghĩa là, với mỗi byte dữ liệu mà máy tính PC gửi, máy tính sẽ nhận được trở lại một byte trả lời có chứa kết quả của quá trình biến đổi A/D.

Nếu như dành thời gian suy nghĩ hơn tí chút ta có thể tạo ra những bản mạch (card) ghép nối, tuy phức tạp hơn, nhưng có thể vẽ ra các họ đường đặc trưng của linh kiện bán dẫn không thua kém gì các máy đo chuyên dụng.

Chương 5

SỰ GHÉP NỐI BUS Ở CỔNG RS 232**5.1 PHẦN CỨNG**

Trong các chương 2, 3 và 4 đã giới thiệu một số mạch điện có thể đem ghép nối cùng với các môđun cơ sở đã trình bày ở chương 1 vào giao diện RS 232 hoặc là vào giao diện máy in của máy tính PC. Nhờ vậy mà người sử dụng có thể thực hiện những phép đo khác nhau mà không cần phải mở máy tính ra.

Cách ghép nối trên đây có một nhược điểm là luôn luôn chỉ có một môđun được sử dụng cho một giao diện. Nhưng trong nhiều trường hợp ta lại muốn trao đổi một cách đồng thời với nhiều môđun ghép nối khác nhau, để rồi theo cách này có thể xây dựng lên một hệ thống đo từ những môđun riêng biệt. Hệ thống có được khả năng này được gọi là hệ thống bus. Rãnh cắm của máy tính cũng tạo nên một hệ thống bus như vậy, và bộ xử lý có thể truy nhập lên nhiều card mở rộng được cắm vào.

Một "bus" điển hình có chứa các đường dẫn dữ liệu mà qua đó các thành viên tham dự được đấu nối vào sẽ trao đổi dữ liệu với nhau. Để lựa chọn các thành viên tham dự đã có các đường dẫn địa chỉ được đặc trưng như là bus địa chỉ. Ngoài ra còn có những tín hiệu điều khiển bổ sung để khẳng định hướng dữ liệu và để đồng bộ quá trình truyền. Chúng được gọi chung là bus điều khiển. Hình 5.1 chỉ ra sơ đồ mạch điện của giao diện bus đối với cổng nối tiếp của một máy tính PC.

Mạch điện này làm việc như thế nào ?

Khi ta so sánh mạch điện này với môđun cơ sở trên hình 1.14 thì sẽ thấy có nhiều điểm chung. Phần chính của mạch điện là một bộ UART CDP 6402. Nó nhận các dữ liệu được gửi nối tiếp của máy tính PC và sắp xếp các byte dữ liệu ở các đường dẫn D0OUT - D7OUT theo cách song song. Qua đó nó gửi các dữ liệu xếp kể sát, song song ở D0IN -

[illegible]

Thế còn vi mạch 74HC573 dùng để làm gì ?

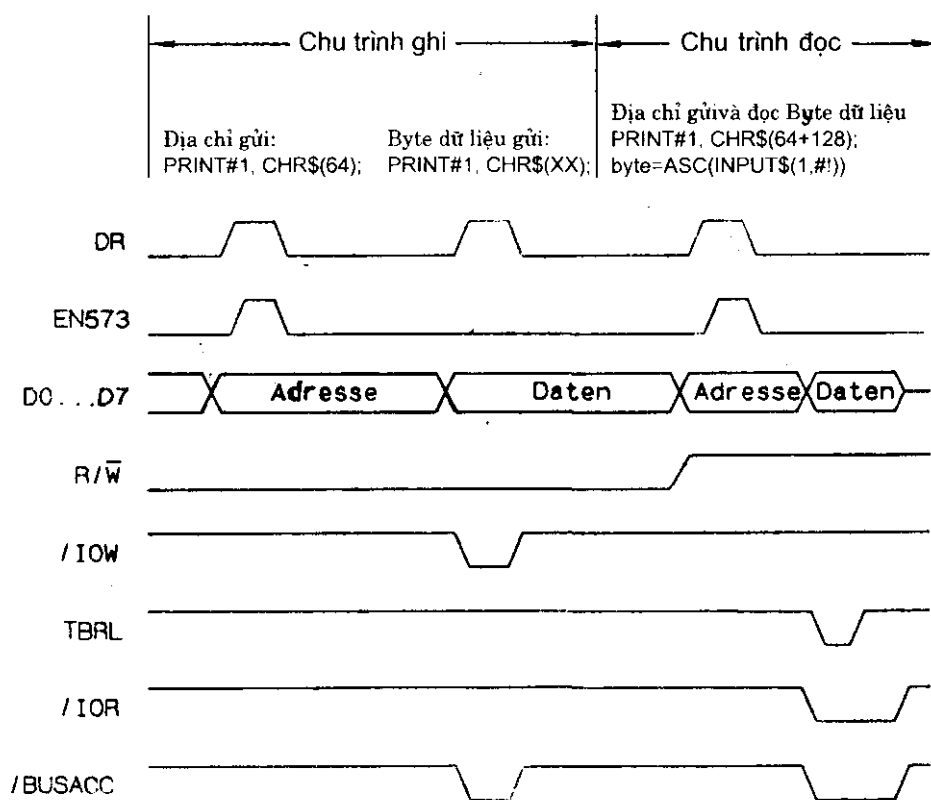
Như trước đây đã giải thích, phần cơ bản tạo nên một hệ thống bus là bus địa chỉ. Qua các đường dẫn địa chỉ, một thành viên cần trao đổi sẽ được lựa chọn. Bởi vì bộ CDP 6402 chỉ có 8 đường dẫn lối ra nên một chu trình ghi, nghĩa là một chu trình mà trong đó các dữ liệu và địa chỉ được gửi, sẽ phải phân chia thành hai giai đoạn. Trước hết máy tính PC gửi các địa chỉ của thành viên cần trao đổi. Các địa chỉ được lưu trữ trong vi mạch 74HC573 và đứng ở các lối ra Q để khi cần thì dùng đến. Sau đấy các dữ liệu được gửi qua bus dữ liệu. Liệu một byte dữ liệu đã tới bộ CDP 6402 có chứa một địa chỉ hay là các dữ liệu, việc này do linh kiện GAL 16V8 lọc lựa. Nó sẽ điều khiển chân 11 của vi mạch 74HC573. Khi chân này nằm ở mức High thì vi mạch 74HC573 sẽ lưu trữ các byte dữ liệu xếp kế sát như là địa chỉ. Ở mức Low, các lối ra Q duy trì trạng thái logic của nó, và cũng cả khi các lối vào D thay đổi mức của nó. Từ 8 bit có thể, sẽ chỉ có bảy được sử dụng để tạo nên địa chỉ. Bit có giá trị cao nhất (R/W) dùng để phân biệt xem liệu một chu trình đọc hay là ghi đang xảy ra (so sánh với hình 5.2). Khi R/W = 1 các dữ liệu có thể được gửi, còn R/W = 0 báo hiệu một chu trình ghi.

Vùng địa chỉ	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
Toàn bộ:	X	1	0	0	X	X	X	X	X	X
100 (Hex)...	0	1	0	0	0	0	0	0	0	0
13F (Hex)...	0	1	0	0	1	1	1	1	1	1
300 (Hex)...	1	1	0	0	0	0	0	0	0	0
33F (Hex)...	1	1	0	0	1	1	1	1	1	1

Bus địa chỉ ở ổ cắm K2 và K3 quyết định quyền sử dụng 10 đường dẫn địa chỉ. Trong đó A6 và A7 được đặt lên đất còn A8 lên + 5 V. Đường dẫn địa chỉ A9 lấy ra lối ra 7Q của vi mạch 74HC573. Thoạt nhìn thì sự sắp xếp này có thể không đem lại ý nghĩa gì. Nhưng ta cần chú ý là bus địa chỉ đã được lựa chọn sao cho hệ thống bus của rãnh cắm PC sẽ đạt được tính tương thích lớn nhất. Điều này tạo ra khả năng xây dựng các môđun bus tương tự như bản mạch (card) cắm thêm vào. Nhờ vậy mà ngay đến những card đơn giản cắm thêm vào cũng có thể hoạt động ở giao diện bus dành cho giao diện nối tiếp.

Ta trở lại một lần nữa với các đường dẫn địa chỉ. Với mạch điện ở hình 5.1 ta có vùng địa chỉ dưới đây đối với hệ thống bus:

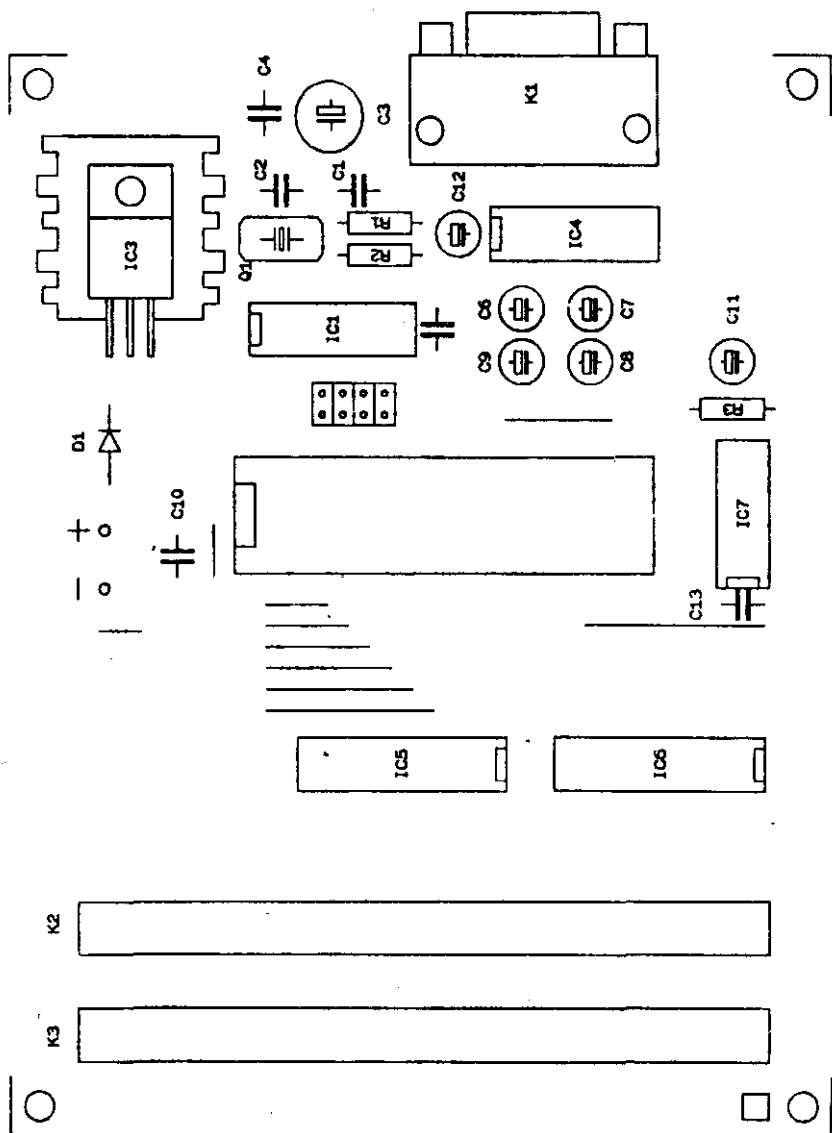
Tổng cộng, người sử dụng có thể truy nhập lên 128 địa chỉ. Ta hãy thử so sánh: ở rãnh cắm PC vùng địa chỉ được quy định đối với card mở rộng là 300... 31F (hex), tương ứng với 32 địa chỉ vào ra (I/O). Vùng địa chỉ này có chứa trong hệ thống bus vừa trình bày. Qua đó các địa chỉ 100... 13F được truy nhập tới.



Hình 5.2 Giản đồ xung dùng cho sự ghép nối bus ở cổng RS 232.

Còn có một thành phần quan trọng khác nữa của một hệ thống bus là bus điều khiển. Bus này được tạo nên qua 4 tín hiệu IOW, IOR, BUSACC và RESET. Tương tự như ở rãnh cắm PC, khi ở mức Low, IOW báo hiệu một chu trình ghi và IOR báo hiệu một chu trình đọc. BUSACC trở nên Low khi một thành viên trong hệ thống bus được trao

đổi. Khi bật điện áp nguồn nuôi, đường dẫn RESET cho ra một xung High ngắn, xung này có thể được sử dụng như là tín hiệu đặt lại. Linh kiện GAL 16V8 tạo nên các tín hiệu điều khiển.

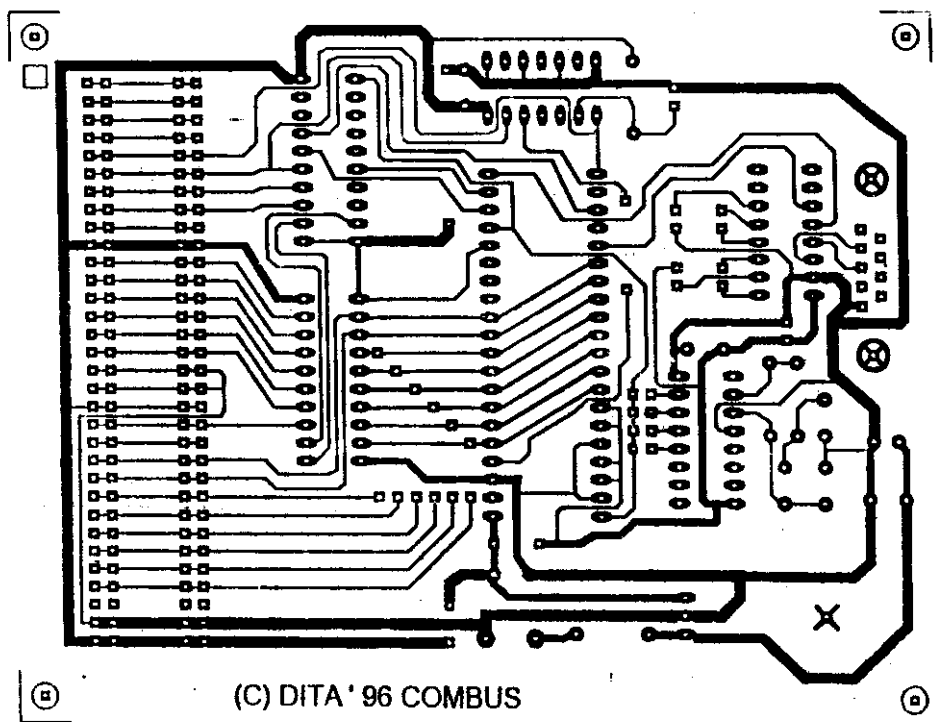


Hình 5.3 Sắp xếp linh kiện của bộ ghép nối bus.

Hình 5.2 chỉ ra sự diễn biến theo thời gian của những tín hiệu khác nhau đã tham dự vào mạch điện được mô tả theo hình 5.1.

Tín hiệu DR (chân 9 của vi mạch CDP 6402) báo hiệu là một byte đã nhận được từ máy tính PC và đang đứng xếp hàng song song ở các đường dẫn lối ra của bộ UART. Bằng xung dương ở chân 11 của vi mạch 74HC573 (hoặc EN 573) các dữ liệu này sẽ được đánh giá như là địa chỉ và được lưu trữ trong linh kiện. Các byte dữ liệu theo sau có chứa những dữ liệu được chuyển giao từ thành viên được trao đổi bằng xung Low của IOW. Nhờ vậy chu trình ghi kết thúc.

Một chu trình đọc được mở đầu trong đó bit có giá trị cao nhất khi gửi được sử dụng làm địa chỉ. Sau đó linh kiện GAL 16V8 nhận biết ra rằng điều này liên quan đến một chu trình đọc. Rồi đến khi $IOR = 0$, một thành viên có thể đặt các dữ liệu của nó lên bus. Trong giai đoạn này, linh kiện GAL 16V8 tạo ra một xung Low ở lối vào TBRL của bộ UART, mở đầu việc gửi tới máy tính PC các dữ liệu xếp kê sát (so sánh với mục 1.2.4). Hình 5.3 chỉ ra sự sắp xếp linh kiện của giao diện bus.



Hình 5.4 Sơ đồ mạch in của giao diện bus.

Để lắp ráp mạch cần có những linh kiện sau đây:

Linh kiện bán dẫn:

IC1: 74HC4060

IC2: CDP6402

IC3: 7805

IC4: MA X232

IC5: 74HC573

IC6: GAL 16V8

IC7: 74HC14

Tụ điện:

C1, C2: 22 pF

C3: 100 μ F

C4, C5, C10, C13: 100 pF

C6, C7, C8, C9, C11, C12:

10 μ F

Điốt bán dẫn:

D1: 1N 4001

Điện trở:

R1, R3: 1 k Ω

R2: 10 M Ω

Phích cắm:

K1: 9 chân

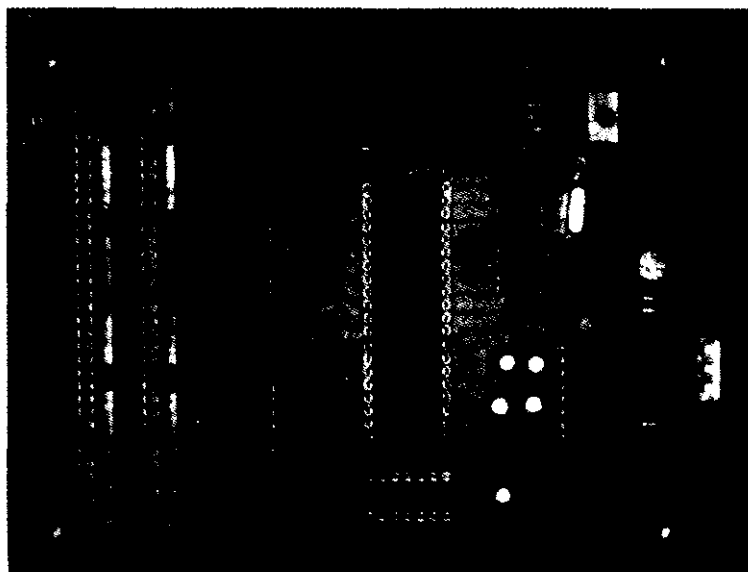
K2, K3: 64 chân

Thạch anh:

Q1: 4,9152 MHz

Trên hình 5.4 mô tả sơ đồ mạch in của giao diện bus. Khi sử dụng hình vẽ này để tự làm mạch in, ta nên chú ý sao cho trên mạch in dòng chữ (C) DITA 96 ghi bên cạnh phải đọc đúng chiều.

Trên hình 5.5 chỉ ra sự sắp xếp linh kiện của bộ ghép nối bus.



Hình 5.5 Bản mạch in và linh kiện của bộ ghép nối bus.

5.2 PHẦN MỀM

Trong mục trình bày về phần cứng ta đã thấy là việc gửi một byte dữ liệu tới một thành viên của hệ thống bus cần có hai lệnh. Trước hết địa chỉ được gửi, sau đấy mới đến byte dữ liệu. Để gửi địa chỉ đi ta cần phải lưu ý một số điểm. Tương tự với rãnh cắm PC, vùng địa chỉ của hệ thống bus là 100... 13F (hex) và 300... 33F (hex). Khó khăn là ở chỗ một địa chỉ nào đấy từ vùng địa chỉ vừa mới kể ra có thể không gửi trực tiếp đến được. Đối với các đường dẫn địa chỉ A0... A9 sẽ cần có 10 bit, nhưng mà bộ UART CDP 6402 lại chỉ có 8 đường dẫn lối ra để sử dụng. Nên để có thể sử dụng các địa chỉ có thực của hệ thống bus trong một chương trình thì cần có một sự chuyển đổi được thực hiện dưới dạng một chương trình con.

Thí dụ sau đây sẽ làm rõ hơn điều này.

Giả sử là cần phải gửi byte dữ liệu 125 (dec.) tới một phần tử nằm trong một nhóm cấu thành của hệ thống bus dưới địa chỉ 300 (hex). Muốn thế cần có những lệnh sau đây trong QBasic.

```
DECLARE SUB out.data (address!, outbyte)

DIM SHARED basadr
CLS
'
'-----Mở giao diện: 9600 baud ở COM2-----
'
OPEN "com2: 9600,N,8,1,CS,DS" FOR RANDOM AS #1
'
'--Xác định các địa chỉ cơ bản và địa chỉ cổng--
'
basadr = &H300      'Địa chỉ cơ bản
port = basadr + 2 'Địa chỉ của cổng trong thí dụ
'
'-----Xuất các Byte dữ liệu ở cổng-----
'
CALL out.data (port, 125)
'
```

CLOSE 1

END

SUB out.data (address, outbyte)

'=====

'

' Unterprogram: out.data

'

' Function: Chương trình cho ra một byte (outbyte) ở
' giao diện bus dưới địa chỉ "address"

'

'=====

'

A9 = (basadr AND 2 ^ 9) / 2 ^ 9

addressbyte = (address AND (255-128-64))+A9*64

PRINT #1, CHR\$ (addressbyte);

PRINT #1, CHR\$ (outbyte);

END SUB

Chương trình con thực hiện sự chuyển đổi cần thiết được gọi vào chương trình chính bằng lệnh *Call*.

Chu trình đọc vào cũng hoàn toàn tương tự. Để truy nhập một lần nữa lên chương trình dùng làm thí dụ trên đây thì một byte dữ liệu cần phải được đọc ra từ phần tử đã được trao đổi dưới địa chỉ 301 (hex). Muốn thế cần có các lệnh dưới đây:

DECLARE SUB inp.data (address!, inbyte)

DIM SHARED basadr

CLS

'

'-----Mở giao diện: 9600 baud ở COM2-----

'

OPEN "com2: 9600,N,8,1,CS,DS" FOR RANDOM AS #1

'

'--Xác định các địa chỉ cơ bản và địa chỉ cổng--

'


```

basadr = &H300      'Địa chỉ cơ bản
port = basadr + 1 'Địa chỉ của cổng trong thí dụ
'
'-----Đọc từ cổng-----
'
CALL inp.data (port, inbyte)
PRINT "Dữ liệu ở cổng với địa chỉ 301"
(hex) là: "; inbyte
'
CLOSE 1
END

```

```

SUB inp.data (address, inbyte)
'=====
'
'  Unterprogram: inp.data
'  Function: Chương trình này đọc vào 1 byte từ giao
'  diện bus qua giao diện nối tiếp của máy tính PC.
'  Khi sự truyền dữ liệu bị nhiễu thì một dòng thông
'  báo hiện lên nhấp nháy và chương trình kết thúc
'
'=====
      A9 = (basadr AND 2 ^ 9) / 2 ^ 9
      addressbyte = (address AND (255-128-64))
      + A9 * 64 + 128
      PRINT #1, CHR$ (addressbyte);

      i = 0
      DO
      i = i + 1
      '
      '-----Khi mà Byte đã có, loc (1) >=1-----
      '
      IF LOC (1) >= 1 THEN
      in$ = INPUT$ (1, #1)

```

```

inbyte = ASC (in$)
GOTO ketthuc
END IF
,
'-----Phép thử mới để đọc dữ liệu vào-----
,
LOOP UNTIL i = 10000      'Cực đại 10000 phép thử
,
'-----Không nhận được byte nào-----
,
CLS
PRINT "Sự truyền dữ liệu bị nhiễu !!!!!"
PRINT
PRINT "Không có ký tự nào được nhận !!!!!"
PRINT
PRINT "Bạn hãy kiểm tra: Sự ghép nối giao diện,
phần cứng..."
END
ketthuc
END SUB

```

Ở đây cần đặc biệt chú ý là bit có giá trị cao nhất đã được đặt ở byte địa chỉ. Đường dẫn này dùng để phân biệt xem liệu việc này liên quan với một chu trình đọc hay là một chu trình ghi (so sánh với hình 5.2).

5.3 CÁC MÔĐUN BUS KHÁC NHAU

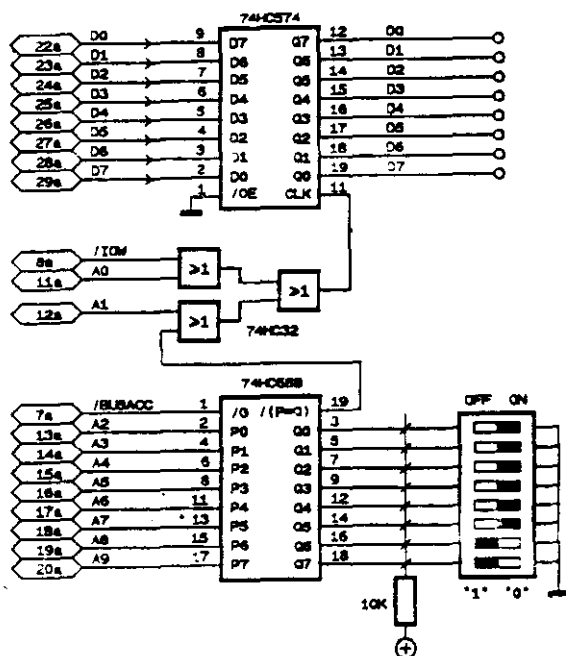
Trong mục này sẽ giới thiệu tiếp một vài môđun bus được ghép nối qua ổ cắm 64 chân của giao diện bus. Mặc dù trên mạch chính chỉ có hai chỗ cắm, nhưng mà với một bản mạch (board) mở rộng đơn giản được nối với bản mạch chính qua một cáp dệt nhiều sợi có thể mở rộng thêm ra đến một số nào đấy. Điều đáng quan tâm ở mạch điện dưới đây là sự giải mã địa chỉ và sự nối bus địa chỉ của một mạch được thực hiện giống hệt như đã gặp ở trên tấm card cắm thêm vào. Nếu như bạn đọc chưa

quen thuộc với rãnh cắm của máy tính PC thì nên xem lại mục 1.3 để có thêm những thông tin cần thiết.

5.3.1 MÔĐUN BUS VỚI 8 LỐI RA TTL

Hình 5.6 chỉ ra một mạch điện đơn giản với 8 lối ra TTL. Các địa chỉ cơ bản của môđun bus có thể đặt được bằng chuyển mạch DIP giống như ở tấm card cắm thêm vào dùng cho máy tính PC. Sự giải mã địa chỉ được tiến hành bằng ba cổng HOẶC. Byte dữ liệu đang đứng xếp hàng D0... D7 được truyền đi bằng sườn dương của xung giữ nhịp ở lối vào CLK của vi mạch 74HC574 vào linh kiện này. Byte dữ liệu đứng xếp hàng ở các đường dẫn lối ra Q0... Q7. Sườn dương được phát ra nhờ sự trợ giúp của tín hiệu IOW, khi mà cả hai đường dẫn địa chỉ A0 và A1 ở mức Low.

Vùng địa chỉ của môđun bus bao gồm một địa chỉ bắt đầu ở 100 (hex) hoặc là 300 (hex) đặt được trên các khoảng cách + 4, chẳng hạn như 300, 304, 308 (hex), vân vân...



Hình 5.6 Sơ đồ mạch của môđun bus với 8 lối ra TTL.

Phần mềm để trao đổi với môđun cơ sở có thể viết như sau:

```

DECLARE SUB out.data (address!, outbyte)
    DIM SHARED basadr
    CLS
    '
    '-----Mở giao diện: 9600 baud ở COM2-----
    '
    OPEN "com2: 9600,N,8,1,CS,DS" FOR RANDOM AS #1
    '
    '---Địa chỉ cơ bản của môđun bus là 108 (hex)---
    '
    basadr = &H108          'Địa chỉ cơ bản
    '
    '-----Đặt tất cả lối ra TTL lên High-----
    '
    CALL out.data (basadr, 255)
    '
    CLOSE #1
    END

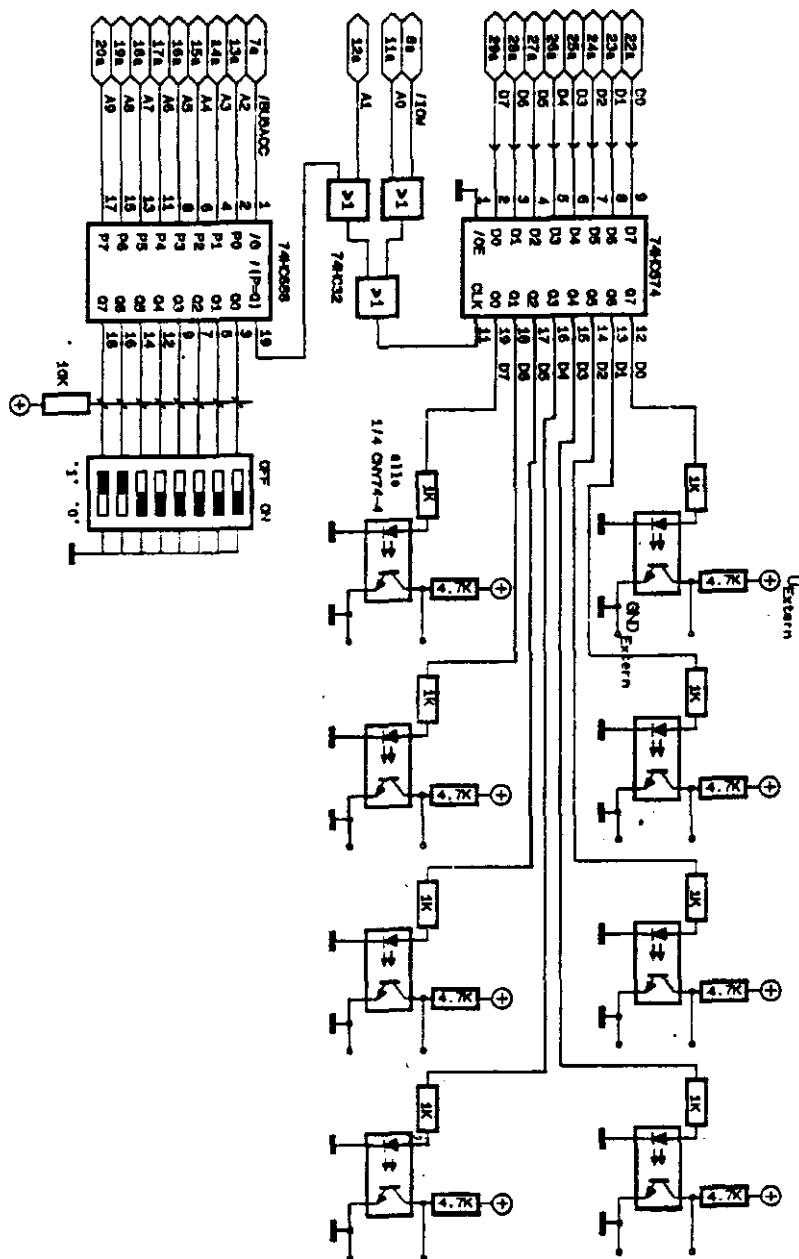
SUB out.data (address, outbyte)
    '      ...Xem mục 5.2 !!
END SUB

```

5.3.2 MÔĐUN BUS VỚI 8 LỐI RA GHÉP NỐI QUANG

Hình 5.7 chỉ ra một sự mở rộng của môđun bus vừa rồi trong đó các lối được điều khiển cách ly với giao diện bus qua bộ ghép nối quang (optocoupler). Ở đây cần phải chú ý rằng mức logic 1 của một đường dẫn dữ liệu được cho ra đảo ngược lại khi đi qua bộ ghép nối quang. Bốn bộ ghép nối quang được chứa chung trong một vi mạch loại CNY 74-4. Khi nối qua một điện trở 1 k Ω lên mức High, sẽ có một dòng điện đi qua điốt phát quang đã được tích hợp trên vi mạch. Dòng điện này thông qua ánh sáng sẽ tác động lên dòng gốc (base) của tranzito quang đặt đối diện và tăng dòng cực góp đi qua điện trở 4,7 k Ω nối với mass. Cần chú

ý là mass này là không đồng nhất với mass của môđun bus được nối với tất cả các vi mạch TTL. Vì thế mà không có sự liên kết về điện (galvanic) giữa hệ thống bus và mức điện áp ở cực góp của một tranzito quang.



Hình 5.7 Sơ đồ mạch của môđun bus với 8 lỗi ra ghép nối quang.

Môđun bus có thể được trao đổi qua một địa chỉ cơ bản. Cả hai đường dẫn địa chỉ A0 và A1 khi đó cần phải có mức Low. Bởi vì sự thiết lập (đặt) địa chỉ cơ bản bắt đầu bằng đường dẫn địa chỉ A2 nên địa chỉ cơ bản có thể được đặt chỉ trên các khoảng cách + 4.

Phần mềm dùng cho môđun bus có thể viết trong QBasic như sau:

```

DECLARE SUB out.data (address!, outbyte)
    DIM SHARED basadr
    CLS
    '
    '-----Mở giao diện: 9600 baud ở COM2-----
    '
    OPEN "com2: 9600,N,8,1,CS,DS" FOR RANDOM AS #1
    '
    '---Địa chỉ cơ bản của môđun bus là 304 (hex)---
    '
    basadr = &H304
    '
    '-----Đặt tất cả lối ra OPTO lên High-----
    '
    CALL out.data (basadr, 0)
    '
    CLOSE 1
    END

SUB out.data (address, outbyte)
    '
    ...Xem mục 5.2 !!
END SUB

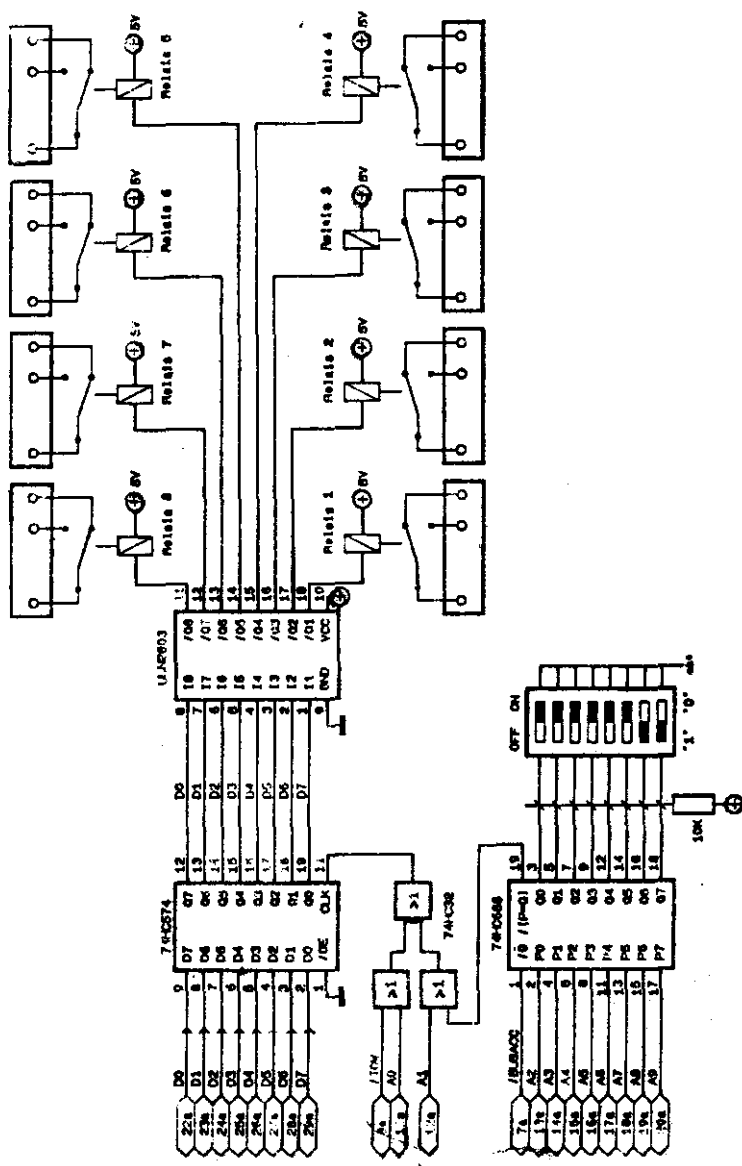
```

5.3.3 MÔĐUN BUS VỚI 8 LỐI RA DÙNG VỚI ROLE

Với môđun bus trong mục 5.3.1 đã có 8 lối ra TTL, các lối vào này có thể điều khiển 8 role qua một vi mạch phụ trợ.

Các lối ra của vi mạch 74HC574 được dẫn đến bộ đệm dùng vi mạch ULN 2803. Nó chứa 8 tầng đệm với mạch cực góp hở, trong đó các

điốt bảo vệ đối với tải cảm ứng đã được tích hợp sẵn. Nhờ thế có thể điều khiển trực tiếp các role. Dòng tải cực đại đạt đến cỡ 500 mA. Sự giải mã địa chỉ là giống hệt với hai môđun bus đã trình bày ở các mục vừa qua.



Hình 5.8 Sơ đồ mạch của môđun bus với 8 lõi ra dùng với role.

Phần mềm để điều khiển môđun bus được viết như sau:

```

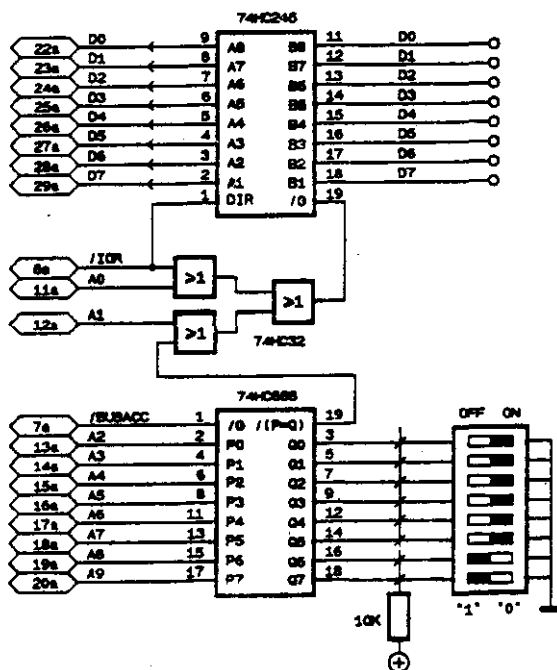
DECLARE SUB out.data (address !, outbyte)
  DIM SHARED basadr
  CLS
  '
  '-----Mở giao diện: 9600 baud ở COM2-----
  '
  OPEN "com2: 9600,N,8,1,CS,DS" FOR RANDOM AS #1
  '
  '---Địa chỉ cơ bản của môđun bus là 300 (hex)----
  '
  basadr = &H300
  '
  '-----Tất cả các Role cần phải được kích hoạt-----
  '
  CALL out.data (basadr, 255)
  '
  CLOSE 1
  END

SUB out.data (address, outbyte)
  '      Xem mục 5.2 !!
END SUB

```

5.3.4 MÔĐUN BUS VỚI 8 LỐI VÀO TTL

Hình 5.9 chỉ ra mạch điện của một môđun bus với tám lối vào TTL. Việc thiết lập địa chỉ cơ bản mà dưới các địa chỉ này môđun được trao đổi có thể tiến hành qua chuyển mạch DIP. Khi có sự thống nhất với địa chỉ bus kế sát, vi mạch 74HC688 tạo ra ở chân 19 một tín hiệu Low, rồi được dẫn tới mạch logic giải mã. Ở chu trình đọc, chân IOR cũng nhận mức Low và chuyển hướng dữ liệu qua lối vào DIR ở bộ đệm bus từ B sang A. Khi các đường dẫn A0 và A1 dẫn đến mức Low thì bộ đệm bus trở nên hoạt động bằng /G = 0. Sau đó tám bit dữ liệu nằm trên bus dữ liệu và được gửi tiếp từ bộ UART CDP 6402 đến máy tính PC theo cách nối tiếp.



Hình 5.9 Sơ đồ mạch của môđun bus với 8 lối vào TTL.

Môđun bus có thể được trao đổi qua một địa chỉ từ vùng địa chỉ của giao diện bus. Ta có thể thiết lập địa chỉ cơ bản bằng chuyển mạch DIP bắt đầu từ 100 (hex) hoặc là 300 (hex) trên khoảng cách + 4.

Các lệnh cần thiết để đọc vào một byte dữ liệu được viết như sau:

```
DECLARE SUB inp.data (address!, inbyte)
```

```
DIM SHARED basadr
```

```
CLS
```

```
'
```

```
'-----Mở giao diện: 9600 baud ở COM2-----
```

```
'
```

```
OPEN "com2: 9600,N,8,1,CS,DS" FOR RANDOM AS #1
```

```
'
```

```
'---Địa chỉ cơ bản của môđun bus là 100 (hex)---
```

```
'
```

```
basadr = &H100
```

```
'
```

```
'-----Đọc vào các byte dữ liệu-----
```

```

CALL inp.data (basadr, inbyte)
PRINT "Byte dữ liệu đã đọc vào là:"; inbyte
'
CLOSE 1
END

```

```

SUB inp.data (address, inbyte)

```

```

'      ...Xem mục 5.2 !!

```

```

END SUB

```

5.3.5 MÔĐUN BUS VỚI TÁM LỖ VÀO GHÉP NỐI QUANG

Trong một số ứng dụng lại cần thiết là các lỗ vào số (digital) của giao diện bus phải cách ly về điện (galvanic). Các bộ ghép nối quang cần dùng cho mục đích này có thể chuyển đổi dòng điện lối vào thành dòng điện lối ra mà không cần có một sự ghép nối bằng điện giữa hai dòng điện.

Hình 5.10 chỉ ra mạch điện của môđun bus với các lỗ vào ghép nối quang.

Đối với sự giải mã địa chỉ, những kết luận tương ứng ở trong mục vừa rồi vẫn giữ nguyên giá trị. Vùng địa chỉ cũng đồng nhất với môđun bus trong hình 5.9.

Để điều khiển môđun bus ta có thể sử dụng các lệnh sau:

```

DECLARE SUB inp.data (address!, inbyte)

```

```

DIM SHARED basadr

```

```

CLS

```

```

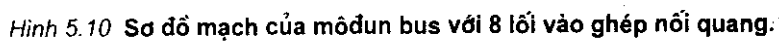
'

```

```

'-----Mở giao diện: 9600 baud ở COM2-----

```



```

OPEN "com2: 9600,N,8,1,CS,DS" FOR RANDOM AS #1
'
'-----Địa chỉ cơ bản của môđun bus là 314 (hex)----
'
basadr = &H314
'
'-----Đọc vào các lối vào Opto-----
'
CALL inp.data (basadr, inbyte)
inbyte = inbyte XOR 255      'Đảo tất cả các bit
PRINT "Byte dữ liệu đã đọc vào là:"; inbyte
'

CLOSE 1

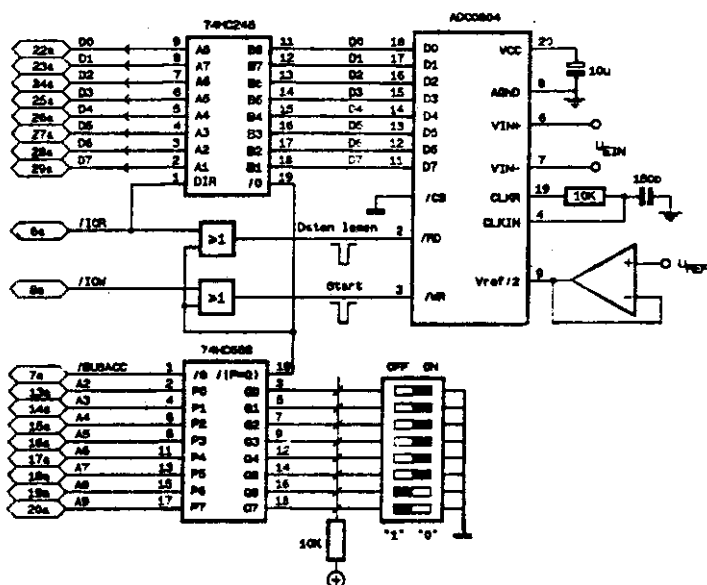
END

SUB inp.data (address, inbyte)
'
'    ...Xem mục 5.2 !!
'
END SUB

```

5.3.6 MÔĐUN BUS VỚI MỘT LỐI VÀO ANALOG

Môđun bus được mô tả dưới đây cho phép đo một điện áp analog. Vì mạch ADC 0804 được sử dụng ở đây như là bộ biến đổi A/D và đã được mô tả kỹ lưỡng trong mục 3.2. Các lối ra dữ liệu của bộ biến đổi A/D được dẫn đến bộ đệm bus có tên là 74HC245. Ở một chu trình đọc (IOR = 0) vì mạch đặt các dữ liệu biến đổi lên bus dữ liệu, để có thể đọc được từ máy tính. Tín hiệu điều khiển IOW mở đầu quá trình biến đổi A/D bằng một xung Low. Muốn thế, cần có một sự truy nhập ghi với một byte dữ liệu nào đó dưới địa chỉ cơ bản của môđun. Các đường dẫn địa chỉ A0 và A1 không tham gia vào mạch logic giải mã. Do đó bắt đầu bằng địa chỉ cơ bản có thể trao đổi với môđun bus qua bốn địa chỉ.



Hình 5.11 Sơ đồ mạch của môđun bus với một lối vào analog.

Đoạn chương trình dưới đây cho thấy môđun bus được trao đổi trong ngôn ngữ QBasic như thế nào.

```

DECLARE SUB inp.data (address!, inbyte)
DECLARE SUB out.data (address!, outbyte)
DIM SHARED basadr
UREF = 2.5           'Điện áp so sánh là 2.5 V
CLS
'-----Mở giao diện: 9600 baud ở COM2-----
OPEN "com2: 9600,N,8,1, CS, DS" FOR RANDOM AS #1
'-----Địa chỉ cơ bản của môđun bus là 118 (hex)---
basadr = &H118
'-----Bắt đầu sự biến đổi A/D-----
CALL out.data (basadr, 0) 'Byte dữ liệu nào đấy

```

'-----Đọc kết quả-----'

```
CALL inp.data (basadr, inbyte)
PRINT "Điện áp là:"; inbyte / 256 * 2 * UREF
CLOSE 1
END
```

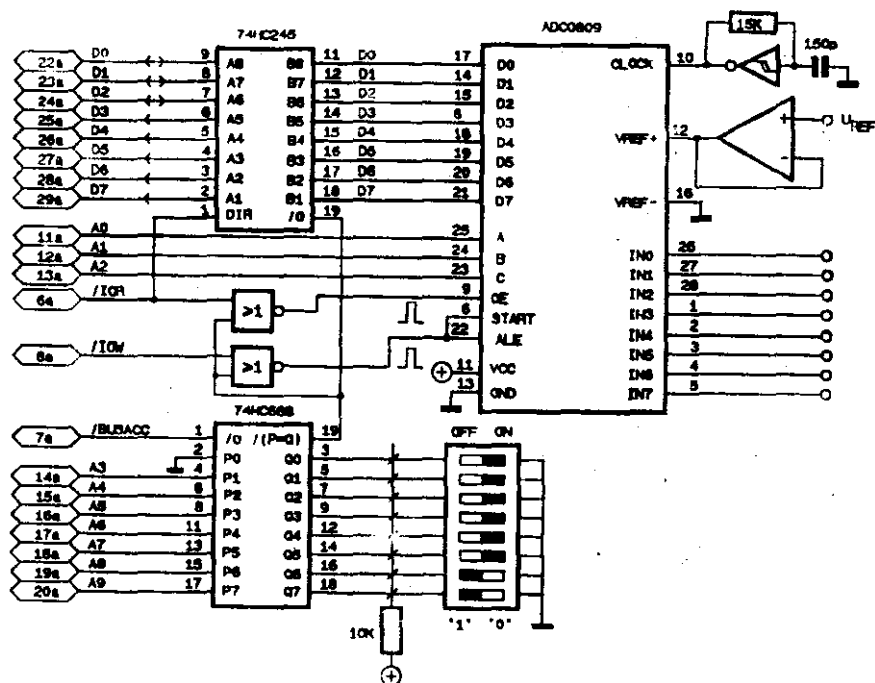
```
SUB inp.data (address, inbyte)
' ...Xem mục 5.2 !!
END SUB
```

```
SUB out.data (address, outbyte)
' ...Xem mục 5.2
END SUB
```

5.3.7 MÔĐUN BUS VỚI TÁM LỐI VÀO ANALOG

Bằng bộ biến đổi A/D ADC 0809 có thể xây dựng lên một môđun bus với tám lối vào analog. Các thông số kỹ thuật của linh kiện này đã được trình bày ở mục 3.3. Ngoài ra cách hoạt động của linh kiện này cũng đã được mô tả chi tiết ở đó.

Hình 5.12 chỉ ra mạch điện của môđun bus. Các đường dẫn dữ liệu của vi mạch ADC 0809 được nối trực tiếp với bộ đệm bus 74HC245. Các lối vào địa chỉ A, B, C, lựa chọn kênh được biến đổi dẫn đến các đường dẫn địa chỉ A0, A1 và A2. Một quá trình biến đổi được bắt đầu trong đó ta gửi một byte dữ liệu nào đó dưới địa chỉ của kênh được biến đổi tới vi mạch ADC 0809. Trong chu trình ghi, một xung dương được tạo ra ở lối vào START, xung này lưu trữ địa chỉ đang xếp hàng vào bộ biến đổi A/D và đồng thời khởi động quá trình biến đổi A/D. Sau đó ta có thể đọc vào các dữ liệu của bộ biến đổi.



Hình 5.12 Sơ đồ mạch của môđun bus với tám lối vào analog.

Vùng địa chỉ của môđun cơ sở bao trùm lên tám địa chỉ mà lối vào analog cần phải biến đổi được lựa chọn bằng các địa chỉ này. Địa chỉ cơ bản có thể được chọn bằng vị trí của các chốt gạt ở trên chuyển mạch DIP.

Đoạn chương trình dưới đây cho thấy môđun bus có thể được trao đổi trong QBasic như thế nào.

```

DECLARE SUB inp.data (address!, inbyte)
DECLARE SUB out.data (address!, outbyte)
DIM SHARED basadr
UREF = 5!                                'Điện áp so sánh là 5.0 V
CLS

'-----Mở giao diện: 9600 baud ở COM2-----
OPEN "com2: 9600,N,8,1, CS, DS" FOR RANDOM AS #1

'---Địa chỉ cơ bản của môđun bus là 118 (hex)---
basadr = &H118

```

```

CH0 = basadr           'Kênh (Channel) 0
CH1 = basadr + 1       'Kênh (Channel) 1
CH2 = basadr + 2       'Kênh (Channel) 2
CH3 = basadr + 3       'Kênh (Channel) 3
CH4 = basadr + 4       'Kênh (Channel) 4
CH5 = basadr + 5       'Kênh (Channel) 5
CH6 = basadr + 6       'Kênh (Channel) 6
CH7 = basadr + 7       'Kênh (Channel) 7
'
'-----Bắt đầu sự biến đổi A/D của kênh 3-----
'
CALL out.data (CH3, 0)   'Byte dữ liệu nào đấy
'
'-----Đọc kết quả-----
'
CALL inp.data (basadr, inbyte)
PRINT "Điện áp bằng:"; inbyte / 256 * UREF
'
CLOSE 1
END

```

```

SUB inp.data (address, inbyte)
'    ...Xem mục 5.2
END SUB

```

```

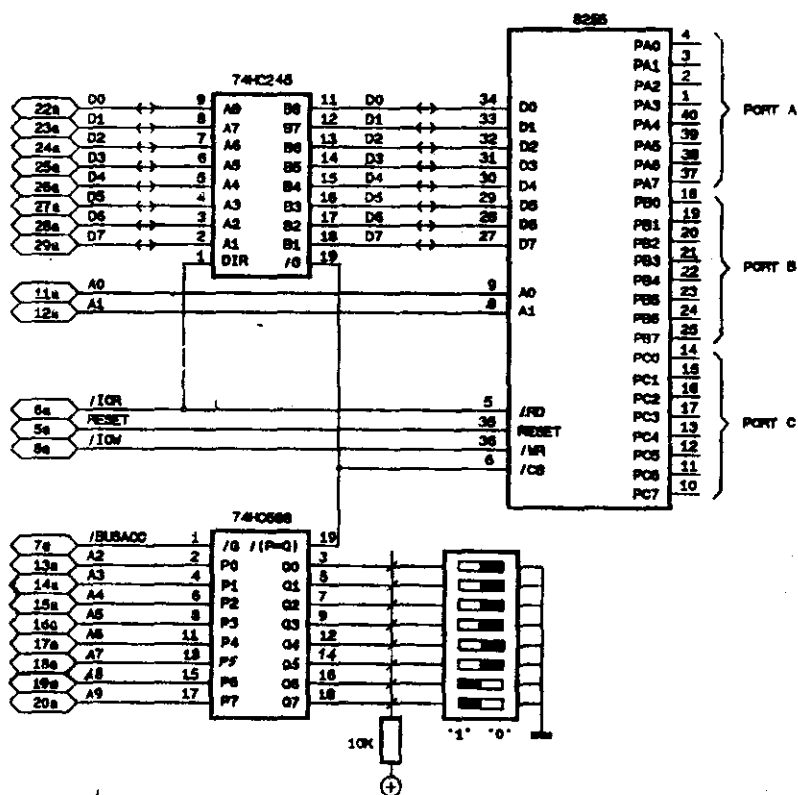
SUB out.data (address, outbyte)
'    ...Xem mục 5.2
END SUB

```

5.3.8 MÔĐUN BUS VỚI 24 ĐƯỜNG DẪN VÀO/ RA

Vi mạch ghép nối với mức độ tích hợp cao (LSI) mang tên 8255, mà ta đã mô tả khá đầy đủ trong mục 2.5, cũng có thể được sử dụng ở giao diện bus dành cho cổng nối tiếp. Hình 5.13 chỉ ra mạch điện của môđun bus đã có sẵn 24 đường dẫn vào ra.

Điều đáng chú ý là, ngoài sự giải mã địa chỉ bằng vi mạch 74HC688 không đòi hỏi có một khối logic giải mã phụ thêm nào. Các tín hiệu IOR, RESET và IOW của bus điều khiển được dẫn trực tiếp đến các lối vào tương ứng của vi mạch 8255. Mặt khác, mạch điện này tương thích với giao diện vào ra (I/O) ở rãnh cắm PC đã minh họa trên hình 2.16.



Hình 5.13 Sơ đồ mạch của môđun bus với 24 lối vào/ ra.

Vùng địa chỉ của môđun bus bao trùm lên trên bốn địa chỉ mà qua đó đạt đến ba cổng cũng như thanh ghi điều khiển. Đoạn chương trình dưới đây chỉ ra cho ta thấy những lệnh nào trong QBasic là cần thiết để trao đổi với ba cổng của 8255.

```
DECLARE SUB inp.data (address!, inbyte)
DECLARE SUB out.data (address!, outbyte)
DIM SHARED basadr
```

CLS

'-----Mở giao diện: 9600 baud ở COM2-----

OPEN "com2: 9600,N,8,1,CS, DS" FOR RANDOM AS #1

'-----Xác định địa chỉ cơ bản và địa chỉ cổng-----

```

basadr = &H300           'Địa chỉ cơ bản
porta = basadr            'Địa chỉ cổng A
porta = basadr + 1        'Địa chỉ cổng B
porta = basadr + 2        'Địa chỉ cổng C
contrreg = basadr + 3     'Địa chỉ của thanh ghi điều khiển

```

'-----Thí dụ: Cổng A: Lối ra -----

'Cổng B: Lối vào

'Cổng C: Lối vào-----

'-----Tù điều khiển dẫn đến 137 (xem text)-----

'-----Gửi tù điều khiển tới-----

CALL out.data (contrreg, 137)

'-----Xuất ra một byte dữ liệu ở cổng A-----

CALL out.data (porta, 143)

'-----Đọc từ cổng C-----

CALL inp.data (portc, inbyte)

PRINT "Các dữ liệu ở chân C là"; inbyte

'-----Đọc từ cổng B-----

CALL inp.data (portb, inbyte)

```
PRINT "Các dữ liệu ở chân B là:"; inbyte
```

```
CLOSE 1
```

```
END
```

```
SUB inp.data (address, inbyte)
```

```
' ...Xem mục 5.2
```

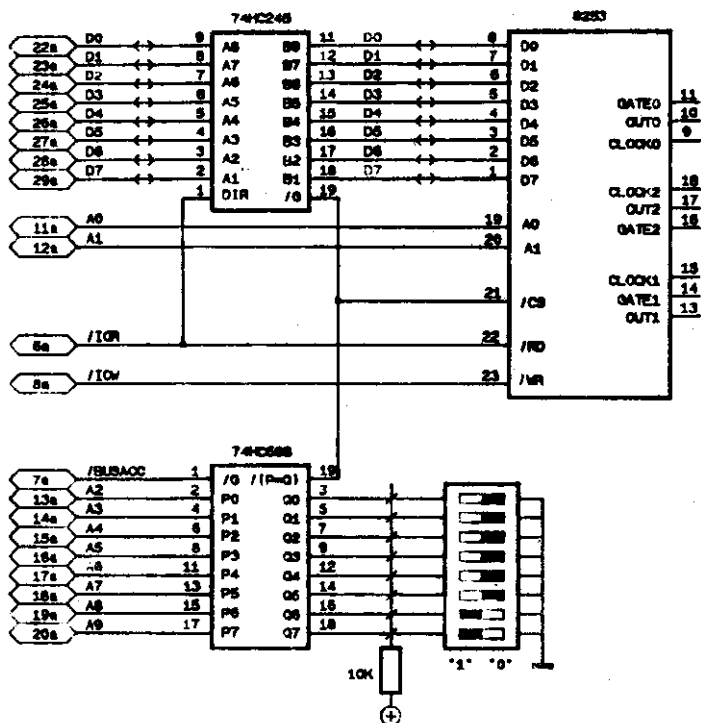
```
END SUB
```

```
SUB out.data (address, outbyte)
```

```
' ...Xem mục 5.2
```

```
END SUB
```

5.3.9 MÔĐUN BUS VỚI VI MẠCH ĐẾM 8253



Hình 5.14 Sơ đồ mạch của môđun bus với vi mạch đếm 8253.

Trong mục 2.6 vi mạch đếm 8253 đã được giới thiệu. Nếu như bạn đọc chưa có những hiểu biết chi tiết về vi mạch này thì nên xem lại mục 2.6 một lần nữa. Hình 5.14 chỉ ra sơ đồ mạch điện của môđun có dùng vi

mạch 8253. Giống như ở vi mạch 8255 đã mô tả trong mục trước đây, ngoài sự giải mã địa chỉ không cần có thêm một mạch logic giải mã nào khác. Việc ghép nối ở hệ thống bus là rất đơn giản. Các đường dẫn địa chỉ A0 và A1 có thể được nối trực tiếp với các chân ra tương ứng ở vi mạch 8253. Điều tương tự cũng xảy ra đúng với tín hiệu điều khiển IOW và IOR.

Khi đã có sự thống nhất giữa địa chỉ được đặt bằng chuyển mạch DIP với các đường dẫn địa chỉ của bus địa chỉ thì vi mạch 8253 sẽ được kích hoạt bằng mức Low ở lối vào CS.

Vùng địa chỉ của linh kiện đếm bao trùm lên bốn địa chỉ mà qua đó đạt đến ba bộ đếm cũng như thanh ghi điều khiển (so sánh thêm với mục 2.6). Đoạn chương trình dưới đây chỉ ra cho ta thấy vi mạch 8253 được trao đổi ở giao diện bus như thế nào.

```

DECLARE SUB inp.data (address!, inbyte)
DECLARE SUB out.data (address!, outbyte)
DIM SHARED basadr
CLS
'
'-----Mở giao diện: 9600 baud ở COM2-----
'
OPEN "com2: 9600,N,8,1,CS,DS" FOR RANDOM AS #1
'
'-----Khẳng định các địa chỉ I/O-----
'
basadr = &H328           'Địa chỉ cơ bản, 328 hex
counter0 = basadr        'Bộ đếm 0
counter1 = basadr + 1    'Bộ đếm 1
counter2 = basadr + 2    'Bộ đếm 2
contrreg = basadr + 3    'Controlregister
'
'-----Thí dụ:-----
' Counter 0: Mode 3, Startvalue: 23580 (Hệ số chia)
' Counter 1: Mode 2, Startvalue: 65535
'

```

```

mode0 = 3           'Kiểu bộ đếm 0
mode1 = 2           'Kiểu bộ đếm 1
start0 = 23580      'Giá trị xuấtphát ở bộ đếm 0
start1 = 65535      'Giá trị xuấtphát ở bộ đếm 1
controlword0 = 54   'Từ điều khiển ở bộ đếm 0
controlword1 = 116  'Từ điều khiển ở bộ đếm 1

```

-----Cấu hình của bộ đếm-----

```

CALL out.data (contrreg, controlword0)   'Bộ đếm 0
CALL out.data (counter0, start0.lowbyte)
           'Trước hết Lowbyte
CALL out.data (counter0, start0.highbyte)
           'Sau đó Highbyte
CALL out.data (contrreg, controlword1)   'Bộ đếm 1
CALL out.data (counter1, start1.lowbyte)
           'Trước hết Lowbyte
CALL out.data (counter1, start1.highbyte)
           'Sau đó Highbyte

```

DO

-----Lưu trữ trung gian trạng thái bộ đếm-----

```

CALL out.data (contrreg, 64)             'Chốt bộ đếm 1

```

-----Đọc ra trạng thái bộ đếm-----

```

CALL inp.data (counter1, stand1.lowbyte)
           'Trước hết Lowbyte
CALL inp.data (counter1, stand1.highbyte)
           'Sau đó Highbyte
counterstand = stand1.lowbyte+256* stand1.highbyte

```

```

PRINT "Trạng thái hiện tại của bộ đếm 1 là:";

```


khoảng 0 ... 2,55 V từng bước 10 mV một. Khi đó, điện áp lỗi ra được lựa chọn tương ứng đúng đến phần mười của byte dữ liệu. Chẳng hạn như khi ta ghi giá trị 129 (dec.) dưới địa chỉ của môđun bus thì một điện áp lỗi ra bằng 1,29 V được thiết lập.

Vùng địa chỉ của môđun bus bao trùm lên một địa chỉ. Địa chỉ này có thể được thiết lập ở trên chuyển mạch DIP bắt đầu từ 100 (hex) và 300 (hex) trên các khoảng cách + 4.

Phần mềm để xuất ra một giá trị điện áp có thể được viết ra như dưới đây:

```

DECLARE SUB out.data (address!, outbyte)
    DIM SHARED basadr
    CLS
    '
    '-----Mở giao diện: 9600 baud ở COM2-----
    '
    OPEN "com2: 9600,N,8,1,CS,DS" FOR RANDOM AS #1
    '
    '----Địa chỉ cơ bản của môđun bus là 330 (hex)---
    '
    basadr = &H330
    '
    '-----Xuất ra 1.78 Volt-----
    '
    CALL out.data (basadr, 178)

    CLOSE 1
    END

SUB out.data (address, outbyte)
    '
    ... xem mục 5.2 !!
END SUB

```

6 PHỤ LỤC

Phụ lục 1

THÔNG SỐ CỦA MỘT SỐ KHUẾCH ĐẠI THUẬT TOÁN

Trong sách đã đề cập nhiều đến các bộ biến đổi A/D cũng như bộ biến đổi D/A cho nên phần này chỉ bổ sung thêm hiểu biết về một số khuếch đại thuật toán.

Khuếch đại thuật toán là một trong số những linh kiện điện tử thường gặp nhất trong kỹ thuật tương tự, và vì thế trong kỹ thuật đo lường và điều khiển khuếch đại thuật toán cũng có mặt trong rất nhiều thiết bị và hệ thống. Dù được dùng như bộ khuếch đại điện áp xoay chiều hay một chiều, dù là được dùng như là bộ lọc tích cực, bộ dao động hay như là bộ biến đổi trở kháng đơn giản thì ta đều phải thừa nhận khả năng sử dụng của chúng là rất vạn năng. Một khuếch đại thuật toán lý tưởng được đặc trưng qua những tính chất sau:

- ☐ Hệ số khuếch đại khi không có phản hồi âm là vô cùng lớn.
- ☐ Điện trở lối vào lớn vô cùng.
- ☐ Độ rộng dải thông lớn vô cùng.
- ☐ Điện trở lối ra bằng không.
- ☐ Thời gian đáp ứng bằng không.

Trên thực tế không có khuếch đại thuật toán lý tưởng, nhưng mà có tồn tại những khuếch đại thuật toán thực có tính chất gần với những tiêu chuẩn đã nêu ở trên. Dưới đây xin dẫn ra những khuếch đại thuật toán mới được sản xuất gần đây cùng với sự mô tả ngắn gọn những đặc tính kỹ thuật của chúng:

- AD 548 Vi mạch AD 548 là một vi mạch khuếch đại thuật toán đơn.
- AD 648 còn AD 658 chứa hai khuếch đại thuật toán. Các khuếch đại thuật toán này được chế tạo để sử dụng trong các ứng dụng, trong đó đòi hỏi công suất nhỏ với tính chất rất đặc sắc là hoạt động với cả dòng một chiều và xoay chiều. Sự phối hợp của điện áp offset và độ trôi nhỏ không đáng kể, dòng tĩnh nhỏ, điện áp nhiễu rất nhỏ cho phép khuếch đại thuật toán

này hoạt động đặc biệt tốt trong các bộ đệm của bộ biến đổi D/A và trong các mạch thông tin để nhập và xuất dữ liệu chạy bằng ắc quy.

Vì mạch AD 548 cho phép người sử dụng giảm tiêu hao công suất đi 85 %. Nó bảo đảm một điện áp offset rất nhỏ - cỡ 2 mV và một độ trôi cỡ 20 mV/ °C. Để đạt được tình trạng hoạt động tối ưu khi chế tạo các vi mạch này người ta đã tiến hành cắt các chip bằng tia laze.

Các thông số kỹ thuật:

○ Dải điện áp nguồn nuôi	$\pm 4,5 \text{ V} \dots \pm 18 \text{ V}$
○ Điện áp offset V_{OS} (max)	2 mV
○ Dòng tĩnh lối vào I_B (max)	20 pA
○ Độ rộng dải khuếch đại GWB	1 MHz
○ Tốc độ tăng điện áp	1 V/ μ s
○ Điện trở lối vào	$10^{12} \Omega$
○ Dòng tiêu thụ (max)	200 μ A (AD 548) 400 μ A (AD 648)

- LF 411 Hai vi mạch này là khuếch đại thuật toán đơn giản và kép có
 LF 412 tốc độ đáp ứng nhanh, có lối vào J-FET với điện áp offset lối vào không đáng kể. Chúng có độ rộng dải khuếch đại lớn và tốc độ tăng điện áp cao. Ngoài ra chúng còn có lối vào J-FET đã được cân chỉnh rất tốt, dòng offset và dòng 0 ở lối vào không đáng kể.

Các thông số kỹ thuật:

○ Dải điện áp nguồn nuôi	$\pm 5 \text{ V} \dots \pm 18 \text{ V}$
○ Điện áp offset V_{OS} (max)	3 mV
○ Dòng tĩnh lối vào I_B (max)	200 pA
○ Độ rộng dải khuếch đại GWB	4 MHz
○ Tốc độ tăng điện áp	15 V/ μ s
○ Điện trở lối vào	$> 10^{12} \Omega$
○ Dòng tiêu thụ (max)	3,4 mA (LF 411) 6,5 mA (LF 412)

LM 324 Vi mạch LM 324 chứa 4 tầng khuếch đại thuật toán trong một vỏ chất dẻo với 14 chân ra. Linh kiện này tỏ ra là lý tưởng đối với những ứng dụng dùng nguồn nuôi đơn giản và sẵn sàng hoạt động từ điện áp nguồn nuôi: $3\text{ V} / \pm 1,5\text{ V}$. Vì thế vi mạch này đặc biệt thích hợp với những mạch điện dùng nguồn nuôi là pin. Mỗi bộ khuếch đại đều được bù trừ bên trong.

Các thông số kỹ thuật:

- Dải điện áp nguồn nuôi $\pm 1,5\text{ V} \dots \pm 15\text{ V}$
/ $3\text{ V} \dots 30\text{ V}$
- Điện áp offset V_{OS} (max) 7 mV
- Dòng tĩnh lối vào I_B (max) 250 nA
- Dải điện áp lối ra (nhỏ nhất) $20\text{ mV} \dots U_B - 4,78\text{ V}$
khi $R_L \geq 2\text{ k}\Omega$
- Dòng tiêu thụ (max) 3 mA

LM 358 Một loại khuếch đại thuật toán công suất nhỏ kép đôi,
LF 412 chuyên dùng cho các mạch hoạt động với nguồn nuôi đơn giản với điện áp từ $3\text{ V} \dots 32\text{ V}$, còn dòng tĩnh cỡ $500\text{ }\mu\text{A}$. Bộ khuếch đại thuật toán này với một lối vào đồng nhịp (in-phase) và một điện áp ra có thể dao động đến điện thế 0 mà có được khả năng nổi bật là có thể hoạt động trực tiếp ở $+5\text{ V}$ và tương thích với tất cả các mức logic.

Các thông số kỹ thuật:

- Dải điện áp nguồn nuôi $1,5\text{ V} \dots \pm 15\text{ V}$
/ $3\text{ V} \dots 32\text{ V}$
- Điện áp offset V_{OS} (max) 7 mV
- Dòng tĩnh lối vào I_B (max) 250 pA
- Dải điện áp lối ra (nhỏ nhất) $20\text{ mV} \dots U_B - 1,5\text{ V}$
khi $R_L \geq 10\text{ k}\Omega$
- Dòng tiêu thụ (max) 2 mA

LMC660 Vi mạch LMC 662 là một vi mạch khuếch đại thuật toán kép
LMC662 2 còn LMC 660 là một vi mạch khuếch đại thuật toán kép 4.

Cả hai đều là linh kiện CMOS lý tưởng đối với chế độ nguồn nuôi đơn giản. Chúng được thiết kế để hoạt động với điện áp nguồn từ + 5 V đến + 15 V và có thể chấp nhận biên độ điện áp lỗi ra lớn đến mức xấp xỉ bằng điện áp nguồn nuôi. Ngoài ra nó chiếm một dải đồng nhịp lỗi vào, đạt đến tận 0 V. Độ trôi của điện áp lỗi vào, nhiễu của dải dải rộng và hệ số khuếch đại điện áp khi có tải thực là tương đương hoặc tốt hơn ở những loại lưỡng cực tương tự thường gặp.

Các thông số kỹ thuật:

○ Dải điện áp nguồn nuôi:	4,75 V ... 15,5 V
○ Điện áp offset V_{OS} (max)	6,3 mV
○ Dòng tĩnh lỗi vào I_B (max)	2 pA
○ Độ rộng dải khuếch đại GWB	1,4 MHz
○ Tốc độ tăng điện áp	1,1 V/ μ s
○ Điện trở lỗi vào	$> 10^{12} \Omega$
○ Dải điện áp lỗi ra (nhỏ nhất)	0,19 V ... 4,78 V $R_L = 2 \text{ k}\Omega, U_B = 5 \text{ V}$
○ Dòng tiêu thụ (max)	2,9 mA (LMC660) 1,8 mA (LMC662)

LP 324 Bốn khuếch đại thuật toán có bù trừ bên trong, hoạt động độc lập với hệ số khuếch đại cao, tiêu thụ dòng điện không đáng kể. Chúng có khả năng được dùng trong các thiết bị điện tử hoạt động bằng pin, chẳng hạn các máy đo cầm tay hoặc các thiết bị khác sử dụng với dòng một chiều cần có dòng điện tiêu thụ không đáng kể.

Các thông số kỹ thuật:

○ Dải điện áp nguồn nuôi:	3 V ... 32 V
○ Điện áp offset V_{OS} (max)	9 mV
○ Dòng tĩnh lỗi vào I_B (max)	20 nA
○ Độ rộng dải khuếch đại GWB	100 kHz
○ Tốc độ tăng điện áp	50 V/ms
○ Dòng tiêu thụ (max)	250 μ A

- OP 90 Các loại này là những bộ khuếch đại chất lượng cao với dòng tiêu thụ cực kỳ nhỏ, có thể dùng được với cả nguồn nuôi đơn giản cũng như với nguồn đối xứng. Giá trị hệ số khuếch đại lớn, điện áp offset vô cùng nhỏ tạo khả năng cho những chức năng chính xác trong các mạch công suất rất nhỏ. Dải điện áp lỗi vào của bộ khuếch đại bao gồm cả điện áp âm, dải này làm thích ứng các tín hiệu lỗi vào với mass cho nên vi mạch có thể hoạt động với nguồn nuôi một cực đơn giản. Nhờ vậy, tín hiệu lỗi ra thấp nhất có thể chấp nhận giá trị điện áp mass, và ngoài ra cách hoạt động “không vào, không ra: zero in, zero.out” là có thể. Mỗi bộ khuếch đại đòi hỏi một dòng tĩnh cỡ 20 μA , nhưng lại có thể cung cấp một dòng điện lớn hơn 5 mA ở tải. Công suất tiêu thụ cực tiểu cho phép vi mạch này có thể dùng trong các thiết bị chạy bằng pin thường hoặc pin mặt trời. Các ứng dụng thường gặp là các máy cầm tay, các bộ khuếch đại đo, bộ biến đổi D/A, các bộ khuếch đại lỗi ra, các bộ so sánh điện áp nhỏ.

Các thông số kỹ thuật:

- | | |
|---------------------------------|--|
| ○ Dải điện áp nguồn nuôi | $\pm 0,8 \text{ V} \dots \pm 18 \text{ V}$
$/ 1,6 \text{ V} \dots 36 \text{ V}$ |
| ○ Điện áp offset V_{OS} (max) | 450 μV (OP 90)
500 μV (OP 290)
1 mV (OP 490) |
| ○ Dòng tĩnh lỗi vào I_B (max) | 15 nA |
| ○ Độ rộng dải khuếch đại GWB | 20 kHz |
| ○ Tốc độ tăng điện áp | 12 V/ms |
| ○ Điện trở lỗi vào | 30 M Ω |
| ○ Dải điện áp lỗi ra (nhỏ nhất) | 0,0005 V ... 4,2 V
$R_L \geq 10 \text{ k}\Omega, U_B = 5 \text{ V}$ |
| ○ Dòng tiêu thụ (max) | 20 mA (OP 90)
40 mA (OP 290)
80 mA (OP 490) |

TLC271 Họ các khuếch đại thuật toán đất tiền này với dòng tiêu thụ
 TLC272 không đáng kể tỏ ra là đặc biệt thích hợp với nguồn nuôi đơn
 TLC274 giản. Khi đặt vào chân này một điện áp cỡ điện áp nguồn
 nuôi thì có nghĩa là ta đã chọn loại thiên áp thấp. Còn khi
 đặt vào một điện áp cỡ một nửa giá trị điện áp nguồn nuôi thì
 khuếch đại thuật toán hoạt động ở chế độ thiên áp trung
 bình. Cuối cùng, nếu như đặt vào một điện áp 0 V thì ta có
 được chế độ thiên áp cao.

Các thông số kỹ thuật của TLC271:

○ Dải điện áp nguồn nuôi:	3 V ... 6 V
○ Điện áp offset V_{OS} (max)	10 mV
○ Dòng tĩnh lối vào I_B (max)	0,6 pA
○ Độ rộng dải khuếch đại GWB	
* Thiên áp thấp	90 kHz
* Thiên áp trung bình	525 kHz
* Thiên áp cao	1,7 MHz
○ Tốc độ tăng điện áp	
* Thiên áp thấp	0,03 V/ μ s
* Thiên áp trung bình	0,46 V/ μ s
* Thiên áp cao	3,6 V/ μ s
○ Điện trở lối vào	$> 10^{12} \Omega$
○ Dòng tiêu thụ (max)	
* Thiên áp thấp	30 μ A
* Thiên áp trung bình	320 μ A
* Thiên áp cao	1,8 mA

*Các thông số kỹ thuật của TLC272 và
 TLC274:*

○ Độ rộng dải khuếch đại GWB	1,7 MHz
○ Tốc độ tăng điện áp	3,6 V/ μ s
○ Dòng tiêu thụ (max)	3,6 mA (TLC272)
	7,2 mA (TLC274)

OP 07 OP 07 là một khuếch đại thuật toán chính xác theo tiêu
 chuẩn công nghiệp được đặc trưng với điện áp offset cực kỳ
 nhỏ với dòng lối vào không đáng kể, nói khác đi nó có điện trở

lối vào cao. Vi mạch này có phần bù tần số bên trong và trong nhiều trường hợp có thể thay thế trực tiếp cho các khuếch đại thuật toán khác.

Các thông số kỹ thuật:

○ Dải điện áp nguồn nuôi:	$\pm 3 \text{ V} \dots \pm 18 \text{ V}$
○ Điện áp offset V_{OS} (max)	$75 \mu\text{V}$
○ Dòng tĩnh lối vào I_B (max)	3 pA
○ Độ rộng dải khuếch đại GWB	$0,6 \text{ MHz}$
○ Tốc độ tăng điện áp	$0,3 \text{ V}/\mu\text{s}$
○ Điện trở lối vào	$60 \text{ M}\Omega$

TL 081 TL 081 là một khuếch đại thuật toán đơn có chân ra dùng cho việc điều chỉnh offset 0. Trong khi TL 082 là một khuếch đại thuật toán kép hai còn TL 084 là kép 4; cả hai đều có cùng một cách bố trí chân đối với nguồn nuôi. Các vi mạch này tỏ ra là đặc biệt thích hợp đối với các bộ tích phân nhanh, các bộ khuếch đại, các bộ lọc tích cực và trong các máy đo.

Các thông số kỹ thuật:

○ Dải điện áp nguồn nuôi:	$\pm 3 \text{ V} \dots \pm 18 \text{ V}$
○ Điện áp offset V_{OS} (max)	15 mV
○ Dòng tĩnh lối vào I_B (max)	400 pA
○ Độ rộng dải khuếch đại GWB	3 MHz
○ Tốc độ tăng điện áp	$13 \text{ V}/\mu\text{s}$
○ Điện trở lối vào	$> 10^{12} \Omega$
○ Dòng tiêu thụ (max)	$2,8 \text{ mA (TL 081)}$ $5,6 \text{ mA (TL 082)}$ $11,2 \text{ mA (TL 084)}$

TL 061 TL 061 là một khuếch đại thuật toán đơn có chân ra dùng cho việc điều chỉnh offset 0. Trong khi TL 062 là một khuếch đại thuật toán kép hai còn TL 064 là kép 4; chúng là thế hệ công suất nhỏ của họ 081. Do mỗi bộ khuếch đại tiêu thụ dòng không đáng kể, cỡ $200 \mu\text{A}$, mà các vi mạch này tỏ ra là lý tưởng đối với những ứng dụng có công suất giới hạn, đặc

biệt là trong các dụng cụ đo cầm tay và các mạch hoạt động bằng pin.

Các thông số kỹ thuật:

○ Dải điện áp nguồn nuôi:	$\pm 3 \text{ V} \dots \pm 18 \text{ V}$
○ Điện áp offset V_{OS} (max)	15 mV
○ Dòng tĩnh lỗi vào I_B (max)	400 pA
○ Độ rộng dải khuếch đại GWB	1 MHz
○ Tốc độ tăng điện áp	3,5 V/ μ s
○ Điện trở lỗi vào	$> 10^{12} \Omega$
○ Dòng tiêu thụ (max)	250 μ A (TL 061) 500 μ A (TL 062) 1 mA (TL 064)

AD 820 AD 820 là một khuếch đại thuật toán đơn có chân ra dùng cho việc điều chỉnh offset 0. Trong khi AD 822 là một khuếch đại thuật toán kép hai. Những bộ khuếch đại thuật toán đời mới này của hãng Analog Device tỏ ra có những tính chất rất tốt. Tín hiệu lỗi ra đạt đến cả điện áp nguồn dương cả điện áp nguồn âm. Chúng được thiết kế để dùng cho các mạch hoạt động với nguồn nuôi đơn giản và chỉ ra một dải điện áp lỗi vào bắt đầu ngay từ 0 V. Mặc dù là dòng tiêu thụ không đáng kể nhưng cả hai vi mạch này đều có tốc độ tăng điện áp (độ dốc) và độ rộng dải cao.

Các thông số kỹ thuật:

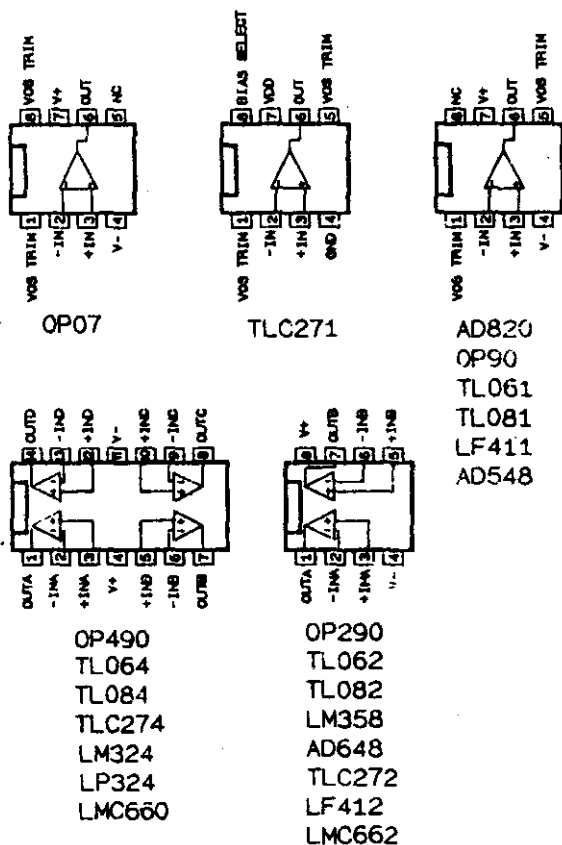
○ Dải điện áp nguồn nuôi	4 V ... 36 V $/ \pm 2 \text{ V} \dots \pm 18 \text{ V}$
○ Điện áp offset V_{OS} (max)	1 mV
○ Dòng tĩnh lỗi vào I_B (max)	20 pA
○ Độ rộng dải khuếch đại GWB	2 MHz
○ Tốc độ tăng điện áp	3,5 V/ μ s
○ Điện trở lỗi vào	$> 10^{12} \Omega$
○ Dải điện áp lỗi ra (nhỏ nhất)	0,004 V ... 4,992 V (không tải) 0,04 V ($I_L = 2 \text{ mA}$)

Dòng tiêu thụ (max)

750 μ A (AD 820)

1,5 mA (AD 822)

Trên hình 6.1 là sự sắp xếp chân ra của một số khuếch đại thuật toán đã mô tả ở trên



Hình 6.1 Sắp xếp chân ra của một số khuếch đại thuật toán.

Phụ lục 2

VÀI NÉT VỀ QBASIC

Chương trình QBasic tạo ra một môi trường cho phép lập chương trình bằng ngôn ngữ Basic. Trong thư mục DOS, bên cạnh chương trình QBasic (mang tên QBasic.EXE) còn thấy có một tệp để thiết lập chế độ ban đầu (QBasic.INI) cũng như một vài chương trình thí dụ có đuôi là BAS. Để xem danh sách các tệp được dùng cho ngôn ngữ QBasic ta có thể dùng lệnh sau:

```
DIR \DOS\QBASIC.*
```

các tệp thí dụ được chỉ ra bằng lệnh:

```
DIR \DOS\*.BAS
```

còn tệp chương trình QBasic.EXE được nạp và khởi động bằng lệnh:

```
QBASIC
```

Trong đó chứa đựng tất cả những gì được yêu cầu đối với việc nạp dữ liệu và việc cho chạy một chương trình được viết ra bằng ngôn ngữ Basic. Việc gọi ra QBasic một cách đầy đủ với tất cả các khả năng lựa chọn (Option) được thực hiện bằng câu lệnh sau:

```
QBasic[/B[/EDITOR[/G[/H[/MBF[/NOHI][[/RUN]Sourcedatei]
```

Option (Lựa chọn) Mô tả

/B Cho phép sử dụng một màn hình đơn sắc với một Card đồ họa màu. Với Option /B, QBASIC sẽ mô tả hình ảnh đơn sắc trên một màn hình màu.

/EDITOR Gọi ra bộ soạn thảo văn bản “ MS-DOS Editor “. Option này cũng có thể được đưa vào dưới dạng ngắn gọn: /ED.

/G Chỉ định cho QBASIC để làm tươi lại (refresh) một màn hình CGA nhanh như có thể (Option này chỉ có thể sử dụng với một màn hình CGA). Khi mà sự làm tươi lại màn hình làm xuất hiện những vết lấm tấm

- thì có thể là phần cứng của bạn không thích hợp với Option này. Khi đó, nếu như ta thích có một màn hình sạch sẽ hơn thì ta hãy khởi động QBASIC không có Option /G.
- /H Chỉ ra số dòng lớn nhất có thể tương thích với phần cứng.
- /MBF Khi chỉ định chức năng biến đổi của QBASIC (CVS<CVD, MKS\$, MKD\$), để xử lý các con số trong khuôn mẫu IEEE như các con số trong khuôn mẫu nhị phân Microsoft.
- /NOHI Cho phép sử dụng một màn hình không thích hợp với cường độ sáng cao. Option này không sử dụng với máy tính Laptop của hãng Compaq.
- Source-datei Là tệp mà QBasic cần phải mở khi khởi động chương trình. Để mở một tệp đã được viết bằng GW-BASIC hoặc là BASICA thì tệp này cần phải được lưu trữ bằng Option, A
- /RUN Source-datei Khi chỉ định QBASIC để mở và cho chạy một chương trình trước khi nó được chỉ ra trên màn hình

QBasic có chứa đựng một phương tiện giúp đỡ trực tuyến (On Line) mà ta có thể tìm đến một cách dễ dàng. Sự trợ giúp đó có thể được sử dụng theo cách sau:

- ☐ Để nhận được sự giải thích về cách dùng một từ khoá của QBasic, ta di chuyển con trỏ đến đó rồi nhấn vào phím F1 hoặc là nhấn vào phím chuột phía bên phải.
- ☐ Để nhận được sự giải thích về một thực đơn QBasic, một lệnh hoặc một hộp đối thoại ta di chuyển con trỏ đến đó rồi nhấn vào phím F1 hoặc là nhấn vào phím chuột phía bên phải.
- ☐ Để xem lại các chủ đề trợ giúp của QBasic, ta di chuyển con trỏ đến đó rồi nhấn phím Alt+H sau đó là một trong các chữ cái được nhấn mạnh (tô đậm hoặc gạch dưới) để lựa chọn một lệnh.

- ❑ Để di chuyển con trỏ tới cửa sổ trợ giúp (Help), hãy ấn Shift+F6.
- ❑ Để lật trang các thông tin trợ giúp lên phía trên hoặc phía dưới, ta nhấn vào phím <Page Down> hoặc <Page Up>.
- ❑ Để sao chép các thông tin trợ giúp (chẳng hạn như các thí dụ chương trình) vào một cửa sổ để nhìn, ta sử dụng các lệnh tương ứng trong thực đơn soạn thảo (Edit).
- ❑ Để đóng cửa sổ trợ giúp (Help) lại, hãy ấn phím ESC.

Để di chuyển con trỏ tới một chủ đề cần trợ giúp, ta ấn vào phím TAB hoặc là vào chữ cái đầu tiên của chủ đề mong muốn. Để xem về một chủ đề hoặc các thông tin về một từ khoá, ta di chuyển con trỏ tới một vị trí nào đó trên chủ đề hoặc là từ khoá rồi sau đó nhấn vào phím F1 (hoặc là vào phím chuột bên phải).

QBasic lưu trữ 20 chủ đề cần trợ giúp sau cùng mà ta đã xem, muốn xem lại các chủ đề này ta nhấn phím Alt+F1 hoặc là nhấn nhiều lần lên bề mặt lựa chọn <Return>.

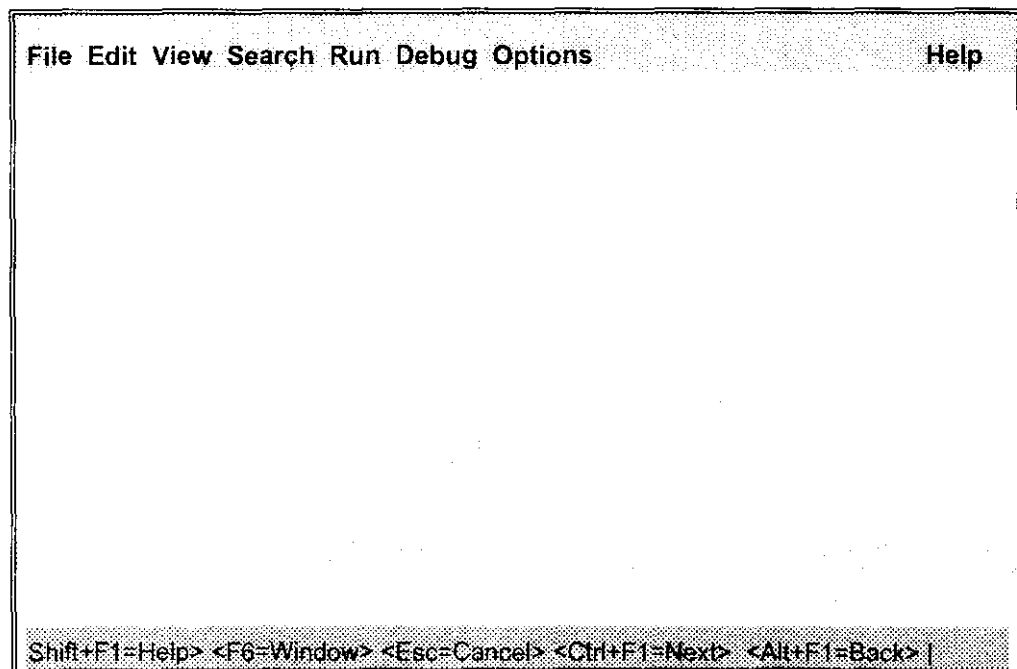
Bảng tóm tắt dưới đây cho ta một cách nhìn tổng quan về các lệnh có thể được sử dụng. Mỗi lệnh đều được mô tả một cách đầy đủ trong thực đơn trợ giúp.

Nhiệm vụ trong khi lập trình	Từ khoá trong danh sách này
Điều khiển việc chạy chương trình	DO...LOOP, END, EXIT, FOR...NEXT, IF...THEN...ELSE, GOSUB...RETURN, GOTO, ON...GOSUB, ON...GOTO, SELECT CASE, STOP, SYSTEM
Các hằng và biến	CONST, DATA, DIM, ERASE, OPTION BASE,
Khai báo.	READ, REDIM, REM, RESTORE, SWAP, TYPE...END TYPE,
Định nghĩa các thủ tục BASIC	CALL, DECLARE, EXIT,

	FUNCTION, RUN,
Và gọi ra	SHELL, SHARED, STATIC, SUB,
Nhập và xuất dữ liệu ra	CLS, CSRLIN, INKEY\$, INP, INPUT, KEY (Gán giá trị), LINE INPUT, LOCATE, LPOOS, LPRINT, LPRINT USING, OPEN COM, OUT, POS, PRINT, PRINT USING, SPC, Hàm- SCREEN, TAB, VIEW PRINT, WAIT, WIDTH
Mô tả bảng đồ hoạ	CIRCLE, COLOR, GET (Grafic), LINE, PAINT, PALLETTE, PCOPY, PMAP, POINT, PRESET, PSET, PUT (Grafic), Lệnh-SCREEN, VIEW, WINDOW,
Lệnh hệ thống từ DOS	CHDIR, KILL, MKDIR, NAME, RMDIR
Xuất và nhập Datei	CLOSE, EOF, FILEATTR, FREEFILEGET (Datei-I/O), INPUT, LOC, LOCK, LOF, OPEN, PUT, (Datei-I/O), Hàm- SEEK, Lệnh-SEEK , UNLOCK, WRITE
Quản lý bộ nhớ	CLEAR, FRE, PEEK, POKE,
Quản lý dãy ký tự	ASC, CHR\$, HEX\$, INSTR, LCASE\$, LEFT\$, LEN, LSET, LTRIM\$, Hàm MID\$, Lệnh- MID\$, OCT\$, RIGHT\$, RSET, RTRIM\$, SPACE\$, STR\$, STRING\$, UCASE\$, VAL,

Thực hiện các phép tính toán học	ABS, ASC, ATN, CDBL, CINT, CLNG, COS, CSNG, CVDMBF, CVSMBF, EXP, INT, LOG, RAN-DOMIZE, RND, SGN, SIN, SQR, TAN, Hàm TIME\$,
Theo dõi biến cố và theo dõi lỗi	COM, ERDEV, ERDEV\$, ERL, ERR, ERROR, KEY (Theo dõi biến cố), ON COM, ON ERROR, ON KEY, ON PEN, ON PLAY, ON STRIG, ON TIME, PEN, PLAY (Theo dõi biến cố), RESUME, RETURN, STRIG, Hàm TIMER, Lệnh TIMER,

Hình vẽ 6.2 dưới đây mô tả sự trình bày màn hình sau khi khởi động chương trình QBasic:



Hình 6.2 Màn hình của QBasic sau khi được khởi động.

Sau khi khởi động từ DOS bằng câu lệnh đã trình bày ở phần trên, ta có thể làm việc trực tiếp với QBasic. Những thao tác đều được nhắc nhở trên dòng thực đơn phía trên và lời giải thích ở phía dưới của màn hình....

Phụ lục 3**MỘT SỐ LOẠI RÃNH CẮM
TRONG MÁY TÍNH PC**

Để giúp bạn đọc khi tìm hiểu các card ghép nối có chất lượng cao, trong phần này sẽ được giới thiệu sự sắp xếp chân ra ở một số rãnh cắm khác với ISA:

□ RÃNH CẮM 32 BIT THEO CHUẨN EISA (Extended Industry Standard Architecture)

Kích thước thông thường của một card EISA là:

- * Chiều cao: 127 mm (5,0 inhxơ)
- * Chiều dài: 333,5 mm (13,13 inhxơ)
- * Chiều dày, bao gồm cả linh kiện: 12,7 mm (0,5 inhxơ)

Từ kích thước này, rõ ràng là một card ISA có thể cắm vừa rãnh cắm EISA. Rãnh này vừa có thể chấp nhận các card ISA 8 và 16 bit vừa có thể duy trì chế độ hoạt động 32 bit của các card ghép nối tuân theo đúng chuẩn EISA. Rãnh cắm EISA được dùng cho bộ xử lý 80386 DX và các thế hệ kế tiếp

Sự sắp xếp chân ra trên rãnh cắm EISA có dạng tương tự như trên rãnh cắm ISA chỉ khác ở vị trí cụ thể và tên gọi của các chân, cho nên để tiện so sánh trong bảng ở trang sau sẽ dẫn ra sự sắp chân của cả hai loại.

EISA	ISA	Phía mạch in	Phía linh kiện	ISA	EISA
GND	GND	B01	A01	/IOCHCK	-CMD
+ 5 V	Reset	B02	A02	D7	-START
+ 5 V	+ 5 V	B03	A03	D6	EXRDY
Dự trữ	IRQ2	B04	A04	D5	-EX32
Dự trữ	- 5 V	B05	A05	D4	GND
(Steg)	DRQ2	B06	A06	D3	(Steg)
Dự trữ	- 12 V	B07	A07	D2	-EX16
Dự trữ	Dự trữ	B08	A08	D1	-SLBURST
+ 12 V	+ 12 V	B09	A09	D0	-MSBURST
M-IO	GND	B10	A10	IOCHRDY	W-R
-LOCK	/MEMW	B11	A11	AEN	GND
Dự trữ	/MEMR	B12	A12	A19	Dự trữ
GND	/IOW	B13	A13	A18	Dự trữ
Dự trữ	/IOR	B14	A14	A17	Dự trữ
-BE3	/DACK3	B15	A15	A16	GND
(Steg)	DREQ3	B16	A16	A15	(Steg)
-BE2	/DACK1	B17	A17	A14	-BE1
-BE0	DREQ1	B18	A18	A13	-LAddress31
GND	/DACK0	B19	A19	A12	GND
+ 5 V	CLK	B20	A20	A11	-LAddress30
-LAddress29	IRQ7	B21	A21	A10	-LAddress28
GND	IRQ6	B22	A22	A9	-LAddress27
-LAddress26	IRQ5	B23	A23	A8	-LAddress25
-LAddress24	IRQ4	B24	A24	A7	GND
(Steg)	IRQ3	B25	A25	A6	(Steg)
LAddress16	/DACK2	B26	A26	A5	LAddress15
LAddress14	TC	B27	A27	A4	LAddress13
+ 5 V	ALE	B28	A28	A3	LAddress12
+ 5 V	+ 5 V	B29	A29	A2	LAddress11
GND	OSC	B30	A30	A1	GND
LAddress10	GND	B31	A31	A0	LAddress9

LAddress8	MEMCS16	D01		C01	-SBHE	LAddress7
LAddress6	I-O CS16	D02		C02	LA23	GND
LAddress5	IRQ 10	D03		C03	LA22	LAddress4
+ 5 V	IRQ 11	D04		C04	LA21	LAddress3
LAddress2	IRQ 12	D05		C05	LA20	GND
(Steg)	IRQ 13	D06		C06	LA19	(Steg)
Data bit 16	IRQ 14	D07		C07	LA18	Data bit 17
Data bit 18	-DARK 0	D08		C08	LA17	Data bit 19
GND	DRQ 0	D09		C09	-MEMR	Data bit 20
Data bit 21	-DARK 5	D10		C10	-MEMW	Data bit 22
Data bit 23	DRQ 5	D11		C11	SD8	GND
Data bit 24	-DARK 6	D12		C12	SD9	Data bit 25
GND	DRQ 6	D13		C13	SD10	Data bit 26
Data bit 27	-DARK 7	D14		C14	SD11	Data bit 28
(Steg)	DRQ 7	D15		C15	SD12	(Steg)
Data bit 29	+ 5 V	D16		C16	SD13	GND
+ 5 V	MASTER	D17		C17	SD14	Data bit 30
+ 5 V	GND	D18		C18	SD15	Data bit 31
-MAKx						-MERQx

□ RÃNH CẮM 32 BIT VÀ 64 BIT THEO CHUẨN VESA VLB (VESA Local Bus Standard)

Việc tạo ra các “local bus” nằm trong ý đồ nhằm đạt được mối liên hệ trực tiếp với bộ xử lý để làm tăng tốc độ truyền dữ liệu, đặc biệt là khi bộ xử lý 803486 ra đời. Còn chữ VESA bắt nguồn từ tên gọi của Video Electronics Standard Association, tổ chức này đã dành nhiều thời gian để tìm kiếm những giải pháp bằng phần cứng để tận dụng tốc độ xử lý của các bộ vi xử lý thế hệ mới.

Rãnh cắm VLB (VESA Local Bus) bao gồm một rãnh cắm ISA 16 Bit và một rãnh mở rộng nằm thẳng hàng với rãnh ISA. Rãnh cắm VLB có đến 116 chân ra được sắp xếp như bảng dưới đây.

		Phía mạch in		Phía linh kiện		
64 bit	32 bit			32 bit	64 bit	
	DAT01	B01		A01	DAT00	
	DAT03	B02		A02	DAT02	
	GND	B03		A03	DAT04	
	DAT05	B04		A04	DAT06	
	DAT07	B05		A05	DAT08	
	DAT09	B06		A06	GND	
	DAT11	B07		A07	DAT10	
	DAT13	B08		A08	DAT12	
	DAT15	B09		A09	Vcc	
	GND	B10		A10	DAT14	
	DAT17	B11		A11	DAT16	
	Vcc	B12		A12	DAT18	
	DAT19	B13		A13	DAT20	
	DAT21	B14		A14	GND	
	DAT23	B15		A15	DAT22	
	DAT25	B16		A16	DAT24	
	GND	B17		A17	DAT26	
	DAT27	B18		A18	DAT28	
	DAT29	B19		A19	DAT30	
	DAT31	B20		A20	Vcc	
DAT62	ADR30	B21		A21	ARD31	DAT63
DAT60	ADR28	B22		A22	GND	
DAT58	ADR26	B23		A23	ADR29	DAT61
	GND	B24		A24	ADR27	DAT59
DAT56	ADR24	B25		A25	ADR25	DAT57
DAT54	ADR22	B26		A26	ADR23	DAT55
	Vcc	B27		A27	ADR21	DAT53
DAT52	ADR20	B28		A28	ADR19	DAT51
DAT50	ADR18	B29		A29	GND	

DAT48	ADR16	B30		A30	ADR17	DAT49
DAT46	ADR14	B31		A31	ADR15	DAT47
DAT44	ADR12	B32		A32	Vcc	
DAT42	ADR10	B33		A33	ADR13	DAT45
DAT40	ADR08	B34		A34	ADR11	DAT44
	GND	B35		A35	ADR09	DAT41
DAT38	ADR06	B36		A36	ADR07	DAT39
DAT36	ADR04	B37		A37	ADR05	DAT37
	WBACK#	B38		A38	GND	
BE4#	BE0#	B39		A39	ADR03	DAT35
	Vcc	B40		A40	ADR02	DAT34
BE5#	BE1#	B41		A41	Dự trữ	LBS64#
BE6#	BE2#	B42		A42	RESET#	LBS64#
	GND	B43		A43	D/ C#	DAT61
BE7#	BE3#	B44		A44	M/ IO#	DAT59
	ADS#	B45		A45	W/ R#	DAT57
(Steg)	(Steg)	B46		A46	(Steg)	(Steg)
(Steg)	(Steg)	B47		A47	(Steg)	(Steg)
	LRDY#	B48		A48	RDYRTN#	
	LDEV<X>#	B49		A49	GND	
	LREQ<X>#	B50		A50	IRQ9	
	GND	B51		A51	BRDY#	
	LGNT<X>#	B52		A52	BLAST#	
	Vcc	B53		A53	ID0	DAT32
	ID2	B54		A54	ID1	DAT33
	ID3	B55		A55	GND	
ACK64#	ID4	B56		A56	LCLK	
	Dự trữ	B57		A57	Vcc	
	LEAD#	B58		A58	LBS16#	

❑ RÃNH CẮM 32 BIT THEO CHUẨN PCI (Peripheral Component Interconnect-Standard)

Tiêu chuẩn này được chính thức đưa ra vào tháng giêng năm 1994 và dự tính áp dụng cho các máy tính PC đến 64 bit, còn tốc độ truyền dữ liệu có thể đạt đến 132 Mbyte mỗi giây. Thông thường, bên cạnh 2 hoặc 3 rãnh cắm PCI, các nhà sản xuất vẫn sắp xếp một vài rãnh cắm ISA để người sử dụng có thể ghép nối một cách linh hoạt tùy theo những card đang sẵn có trong tay.

Bảng dưới đây mô tả sự sắp xếp chân ra trên rãnh cắm PCI, trong đó mặt A là mặt sắp xếp linh kiện còn mặt B là mặt hàn các chân linh kiện.

Chân	5 V, mặt A	5 V, mặt B	3,3 V, mặt A	3,3 V, mặt B	Chú thích
1	- 12 V	TSRT#	- 12 V	TSRT#	Bắt đầu 32bit
2	TCK	+ 12 V	TCK	+ 12 V	
3	GROUND	TMS	GROUND	TMS	
4	TDO	TDI	TDO	TDI	
5	+ 5 V	+ 5 V	+ 5 V	+ 5 V	
6	+ 5 V	INTA#	+ 5 V	INTA#	
7	INTB#	INTC#	INTB#	INTC#	
8	INTD#	+ 5 V	INTD#	+ 5 V	
9	PRSNT1#	Dự trữ	PRSNT1#	Dự trữ	
10	Dự trữ	+ 5 V (I/O)	Dự trữ	+ 3,3 V(I/O)	
11	PRSNT2#	Dự trữ	PRSNT2#	Dự trữ	3,3 V -Steg 3,3 V -Steg
12	GROUND	GROUND	(Steg)	(Steg)	
13	GROUND	GROUND	(Steg)	(Steg)	
14	Dự trữ	Dự trữ	Dự trữ	Dự trữ	
15	GROUND	RST#	GROUND	RST#	
16	CLK	+ 5 V (I/O)	CLK	+ 3,3 V(I/O)	
17	GROUND	GNT#	GROUND	GNT#	
18	REQ#	GROUND	REQ#	GROUND	
19	+ 5 V (I/O)	Dự trữ	+ 3,3 V (I/O)	Dự trữ	
20	AD[31]	AD[30]	AD[31]	AD[30]	
21	AD[29]	+ 3,3 V	AD[29]	+ 3,3 V	
22	GROUND	AD[28]	GROUND	AD[28]	
23	AD[27]	AD[26]	AD[27]	AD[26]	
24	AD[25]	GROUND	AD[25]	GROUND	

Chân	5 V, mặt A	5 V, mặt B	3,3 V, mặt A	3,3 V, mặt B	Chú thích
25	+ 3,3 V	AD[24]	+ 3,3 V	AD[24]	
26	C/BE[3]#	IDSEL	C/BE[3]#	IDSEL	
27	AD[23]	+ 3,3 V	AD[23]	+ 3,3 V	
28	GROUND	AD[22]	GROUND	AD[22]	
29	AD[21]	AD[20]	AD[21]	AD[20]	
30	AD[19]	GROUND	AD[19]	GROUND	
31	+ 3,3 V	AD[18]	+ 3,3 V	AD[18]	
32	AD[17]	AD[16]	AD[17]	AD[16]	
33	C/BE[2]#	+ 3,3 V	C/BE[2]#	+ 3,3 V	
34	GROUND	FRAME#	GROUND	FRAME#	
35	IRDY#	GROUND	IRDY#	GROUND	
36	+ 3,3 V	TRDY#	+ 3,3 V	TRDY#	
37	DEVSEL#	GROUND	DEVSEL#	GROUND	
38	GROUND	STOP#	GROUND	STOP#	
39	LOCK#	+ 3,3 V	LOCK#	+ 3,3 V	
40	PERR#	SDONE	PERR#	SDONE	
41	+ 3,3 V	SBO#	+ 3,3 V	SBO#	
42	SERR#	GROUND	SERR#	GROUND	
43	+ 3,3 V	PAR	+ 3,3 V	PAR	
44	C/BE[1]#	AD[15]	C/BE[1]#	AD[15]	
45	AD[14]	+ 3,3 V	AD[14]	+ 3,3 V	
46	GROUND	AD[13]	GROUND	AD[13]	
47	AD[12]	AD[11]	AD[12]	AD[11]	
48	AD[10]	GROUND	AD[10]	GROUND	
49	GROUND	AD[09]	GROUND	AD[09]	5 V-(Steg)
50	(Steg)	(Steg)	GROUND	GROUND	5 V-(Steg)
51	(Steg)	(Steg)	GROUND	GROUND	
52	AD[09]	C/BE[0]#	AD[09]	C/BE[0]#	
53	AD[07]	+ 3,3 V	AD[07]	+ 3,3 V	
54	+ 3,3 V	AD[06]	+ 3,3 V	AD[06]	
55	AD[05]	AD[04]	AD[05]	AD[04]	
56	AD[03]	GROUND	AD[03]	GROUND	
57	GROUND	AD[02]	GROUND	AD[02]	
58	AD[01]	AD[00]	AD[01]	AD[00]	
59	+ 5 V (I/O)	+ 5 V (I/O)	+ 3,3 V	+ 3,3 V	
60	ACK64#	REQ 64#	ACK64#	REQ64#	
61	+ 5 V	+ 5 V	+ 5 V	+ 5 V	
62	+ 5 V	+ 5 V	+ 5 V	+ 5 V	Hết 32 bit
	(Steg)	(Steg)	(Steg)	(Steg)	Cách ly với
	(Steg)	(Steg)	(Steg)	(Steg)	phần 64 bit

Chân	5 V, mặt A	5 V, mặt B	3,3 V, mặt A	3,3 V, mặt B	Chú thích
63	Dự trữ	GROUND	Dự trữ	GROUND	Bắt đầu 64 bit
64	GROUND	C/BE[7]#	GROUND	C/BE[7]#	
65	C/BE[6]#	C/BE[5]#	C/BE[6]#	C/BE[5]#	
66	C/BE[4]#	+ 5 V (I/O)	C/BE[4]#	+ 3,3 V (I/O)	
67	GROUND	PAR64	GROUND	PAR64	
68	AD[63]	AD[62]	AD[63]	AD[62]	
69	AD[61]	GROUND	AD[61]	GROUND	
70	+5 V (I/O)	AD[60]	+ 3,3 V (I/O)	AD[60]	
71	AD[59]	AD[58]	AD[59]	AD[58]	
72	AD[57]	GROUND	AD[57]	GROUND	
73	GROUND	AD[56]	GROUND	AD[56]	
74	AD[55]	AD[54]	AD[55]	AD[54]	
75	AD[53]	+5 V (I/O)	GROUND	+5 V (I/O)	
76	GROUND	AD[52]	GROUND	AD[52]	
77	AD[51]	AD[50]	AD[51]	AD[50]	
78	AD[49]	GROUND	AD[49]	GROUND	
79	+5 V (I/O)	AD[48]	+ 3,3 V (I/O)	AD[48]	
80	AD[47]	AD[46]	AD[47]	AD[46]	
81	AD[45]	GROUND	AD[45]	GROUND	
82	GROUND	AD[44]	GROUND	AD[44]	
83	AD[43]	AD[42]	AD[43]	AD[42]	
84	AD[41]	+ 5 V (I/O)	AD[41]	+ 5 V (I/O)	
85	GROUND	AD[40]	GROUND	AD[40]	
86	AD[39]	AD[38]	AD[39]	AD[38]	
87	AD[37]	GROUND	AD[37]	GROUND	
88	+ 5 V (I/O)	AD[36]	+ 3,3 V (I/O)	AD[36]	
89	AD[35]	AD[34]	AD[35]	AD[34]	
90	AD[33]	GROUND	AD[33]	GROUND	
91	GROUND	AD[32]	GROUND	AD[32]	Hết 64 bit
92	Dự trữ	Dự trữ	Dự trữ	Dự trữ	
93	Dự trữ	GROUND	Dự trữ	GROUND	
94	GROUND	Dự trữ	GROUND	Dự trữ	



7. TÀI LIỆU THAM KHẢO

- [1] Dembowski, Klaus: *PC-gestuetzte Meß- und Regeltechnik*
Markt & Technik 1991
 - [2] Hans-Joachim Blank, Herbert Bernstein: *PC Schaltungstechnik in der Praxis*
Markt & Technik 1991
 - [3] Link, Wolfgang: *Messen, Steuern und Regeln mit PCs*
Frazis-Verlag 1992
 - [4] Scott Mueller: *Hardware Praxis*
Addison Wesley 1995
 - [5] Kainka, Bukhard: *Messen, Steuern und Regeln ueber die RS 232 Schnittstelle*
Frazis-Verlag 1992
 - [6] *Interface Schaltungen*
Elektor 1990 - 1991
 - [7] U. Tietze, Ch. Schenk: *Halbleiter-Schaltungstechnik*
Springer Verlag, 9th Edition
 - [8] Rose, Michael: *Interfacetechnik*
Vogel-Verlag 1991
 - [9] Ernst Schmeer: *Computergesteuerte Anlagen und Steuerprogramme*
Ummier-Verlag 1995
 - [10] R.J. Brandenburg: *Messen und Auswerten mit MS-DOS Rechnern*
Vogel-Verlag 1994
 - [11] Peter Wratil, Richard Schmidt: *PC/XT/AT Messen, Steuern und Regeln*
Markt & Technik 1991
 - [12] Bosetti P. C., *QBASIC 4.0 /4.5 compact*,
Franzis 1994
 - [13] Drews D. , *QBASIC 4.5 Toolbox*,
Tewi 1993
 - [14] Henning Mittelbach, *Lập trình bằng TURBO Pascal, Version 7.0*
Nhà xuất bản Khoa học và Kỹ thuật, Hà nội 1996
-

ĐO LƯỜNG VÀ ĐIỀU KHIỂN BẰNG MÁY TÍNH

Tác giả : Ngô Diên Tập

Chịu trách nhiệm xuất bản:	PGS. PTS. Tô Đăng Hải
Biên tập:	Đặng Đình Thạch
Trình bày:	PTS Ngô Diên Tập
Sửa bản in:	Quang Ngọc
Chế bản:	Tác giả
Vẽ bìa:	Hương Lan

NHÀ XUẤT BẢN KHOA HỌC VÀ KỸ THUẬT

70 Phố Trần Hưng Đạo, HÀ NỘI

In 1000 cuốn khổ 16x24cm, TẠI CÔNG TY IN HÀNG KHÔNG

Giấy phép xuất bản số: 98-33-2 ngày 8/7/1997

In xong và nộp lưu chiểu tháng 8/1997