

ĐẠI HỌC THÁI NGUYÊN
KHOA CÔNG NGHỆ THÔNG TIN
Bộ môn Công nghệ điều khiển tự động

Phạm Đức Long

Giáo trình
GHÉP NỐI THIẾT BỊ NGOẠI VI

Thái Nguyên 9 - 2007

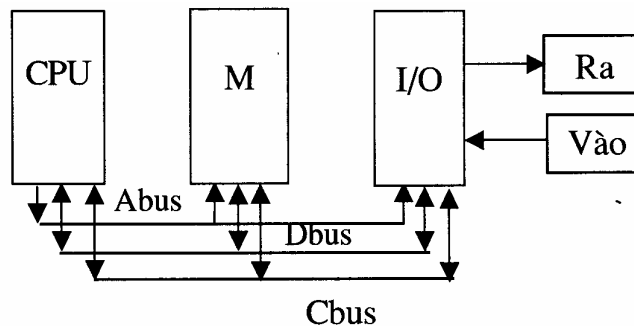
Chương I
Tổng quan về máy tính, thiết bị ngoại vi của máy tính
và các phương pháp trao đổi, biến đổi dữ liệu

1.1. Máy tính và các thành phần cấu thành

1.1.1. Hệ vi xử lý kinh điển

- Bộ vi xử lý là 1 thành phần không thể thiếu được để xây dựng các hệ thống tính toán xử lý, nhưng chỉ riêng bộ vi xử lý thì chưa đủ, nó còn phải được kết hợp với các thành phần khác như bộ nhớ và thiết bị ngoại vi mới tạo nên hệ vi xử lý.

- Hình 1.1 là sơ đồ tổng quát của hệ vi xử lý kinh điển được áp dụng cho các hệ tính toán nhỏ và các máy tính thế hệ đầu.



Hình 1.1 Hệ vi xử lý kinh điển

- CPU - Bộ xử lý trung tâm (Central Processing Unit): Bộ não của máy tính gồm các mạch vi điện tử có độ tích hợp rất cao (hàng triệu tranzito trong 1 chip). Nó gồm có các phần:

- + CU (Control Unit - Khối điều khiển có chức năng): đọc mã lệnh dưới dạng tập hợp các bit 0/1 từ các ô nhớ trong bộ nhớ. Giải mã các lệnh thành dãy các xung điều khiển để điều khiển các khối khác thực hiện như điều khiển ALU, điều khiển ra ngoài CPU,

- + ALU (Arithmetic Logic Unit - Khối tính toán số học và logic): Tổ hợp các mạch logic điện tử phức tạp cho phép thực hiện các thao tác trên các thanh ghi như +, -, *, /, AND, OR, NOT...

- + Các thanh ghi (Registers):

Một CPU có thể có nhiều thanh ghi: Thanh ghi gồm những phần tử nhớ thật nhớ) liên hệ với nhau 1 cách hợp lý, có thể lưu giữ được 1 trong 2 trạng thái thông tin (0 hoặc 1). Thanh ghi thực chất là 1 bộ nhớ được cấy ngay trong CPU. Vì tốc độ truy cập các thanh ghi nhanh hơn là với bộ nhớ chính RAM nên nó được dùng để lưu trữ các dữ liệu tạm thời cho các quá trình tính toán, xử lý của CPU. CPU 808 có 14 thanh ghi.

- Các thanh ghi đoạn 16 bit (8 bit phần thấp và 8 bit phần cao):

CS: Code Segment	Thanh ghi đoạn mã
DS: Data Segment	Thanh ghi đoạn dữ liệu
SS: Stack Segment	Thanh ghi đoạn Stack
ES: extra Segment	Thanh ghi đoạn dữ liệu mở rộng

Nội dung các thanh ghi đoạn chỉ ra địa chỉ đầu (segment) của 4 đoạn trộm bộ nhớ (địa chỉ cơ sở). Địa chỉ của các ô nhớ nằm trong đoạn tính được bằng cách cộng thêm vào địa chỉ cơ sở 1 giá trị gọi là địa chỉ lệch (offset)

- Thanh ghi con trỏ lệnh IP (bộ đếm chương trình) chứa địa chỉ của lệnh sắp thực hiện: Các chương trình máy tính là tập hợp của các lệnh. CPU sẽ lấy từng lệnh ra để chạy. Để điều khiển chính xác việc thực hiện này cần có một bộ đếm chương trình- đó chính là IP. Thanh ghi con trỏ lệnh IP kết hợp với thanh ghi CS chỉ ra địa chỉ đầy đủ của lệnh sắp thực hiện là CS:IP.

- Các thanh ghi dữ liệu: AX, BX, CX, DX. Chúng có độ dài 16 bit gồm 8 bit phần thấp và 8 bit phần cao ($AX=AH+AL$, $BX=BH+BL$, $CX=CH+CL$, $DX=DH+DL$).

- Các thanh ghi con trỏ, chỉ số 16 bit: SP, BP, SI, DI.

- Thanh ghi cờ 16 bit sử dụng 9 bit cho phép biết trạng thái hoạt động của CPU và điều khiển cho phép hay không cho phép ngắt ngoài loại che được.

Để xem các thanh ghi hoạt động ra sao ta có thể dùng chương trình debug với lệnh T (Chạy từng bước) và lệnh R (xem các thanh ghi).

Chẳng hạn: C:\debugfile.exe ↵

Ta cũng có thể dùng 1 chương trình Pascal đơn giản sau để xem hoạt động của các thanh ghi:

begin

Repeat

asm

xor ax,ax

moy al,1

add Al,5

sub al,3

...

end;

until keypressed

end.

ấn *Alt+D* rồi vào mục *Registers*. Sau đó ấn nhả phím *F7* để chạy chương trình và xem các thanh ghi trong CPU đang hoạt động ra sao. Từ máy 386 các thanh ghi đa năng và thanh ghi cờ có độ lớn gấp đôi (32 bit), các thanh ghi đoạn (4 thanh ghi) độ lớn vẫn là 16 bit.

- Bộ nhớ - *Memory*: Gồm có hai loại

- ROM: Vi mạch nhớ ROM chứa các chương trình và số liệu cố định, không bị mất khi ngắt điện cung cấp. Trong một hệ vi xử lý, các chương trình khởi động hệ thống, các chương trình vào/ra cơ sở và có thể cả một số chương trình ứng dụng cụ thể được chứa trong ROM.

- RAM: Vi mạch nhớ RAM khi ngắt điện nguồn nuôi sẽ bị mất nội dung lưu trữ. RAM có thể lưu giữ một phần chương trình hệ thống, một số số liệu của hệ thống, các chương trình ứng dụng, các kết quả trung gian của quá trình tính toán, xử lý.

- Khối phối ghép vào/ra (*I/O*): Đây là khối phục vụ giao tiếp giữa các thiết bị ngoài và hệ trung tâm (*hệ trung tâm bao gồm CPU + M*). Các thiết bị ngoài có thể là thiết bị vào hoặc thiết bị ra. Thiết bị vào ví dụ như phím điều khiển để thay đổi thông số chương trình, điều khiển hoạt động của hệ vi xử lý. Thiết bị ra như các thiết bị hiển thị: LED 7seg, LCD (*với các hệ vi xử lý nhỏ*), màn hình (*với máy tính PC*). Do đặc điểm hoạt động của thiết bị ngoài và hệ trung tâm có sự khác nhau về tốc độ làm việc, mức vật lý, phương thức làm việc nên một số trường hợp (như với máy tính PC) cần có bộ phối ghép đệm, đảm bảo cho các khối thiết bị ngoài giao tiếp được với hệ trung tâm. Bộ ghép giữa bus hệ thống và thiết bị ngoài gọi là cổng. Mỗi cổng có một địa chỉ xác định.

- Hệ thống bus: Là tập hợp các đường dây dẫn ghép nối các chân địa chỉ, dữ liệu, các chân tín hiệu điều khiển của 3 khối đã nêu trên.

- Abus: Nối các đường dây địa chỉ của CPU với 2 khối M và I/O. Khả năng phân biệt địa chỉ của CPU phụ thuộc số chân địa chỉ của nó. Số này có thể là 16, 20, 24, 36 chân. Chỉ có CPU mới có khả năng phát ra tín hiệu địa chỉ.

- Dbus: Dùng để vận chuyển dữ liệu. Độ rộng của nó 8, 16, 32, 64 bit. Dbus có tính 2 chiều. Các phần tử có đầu ra nối thẳng với bus dữ liệu đều phải được trang bị đầu ra 3 trạng thái để có thể làm việc bình thường với bus này.

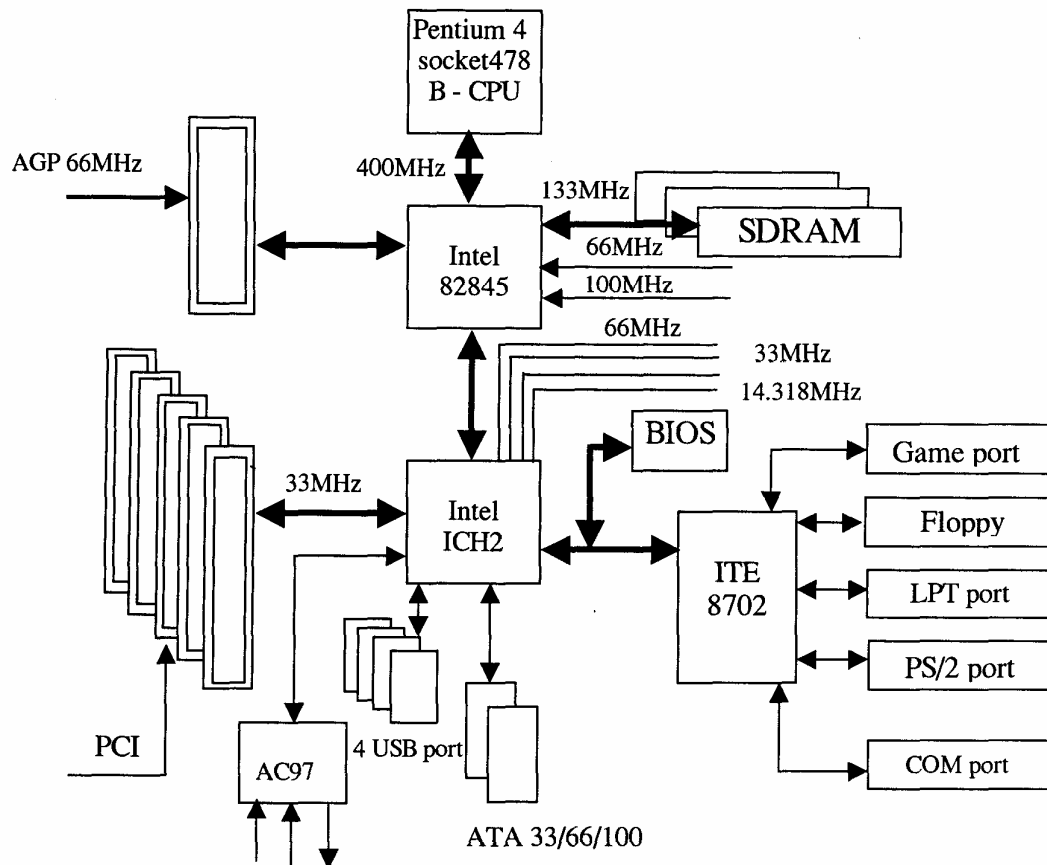
- Cbus: Gồm nhiều đường dây tín hiệu khác nhau. Mỗi tín hiệu có 1 chiều xác định. Các tín hiệu trên Cbus bao gồm các tín hiệu điều khiển từ CPU như dứt đọc viết tín hiệu trạng thái từ bộ nhớ, thiết bị ngoại vi báo cho CPU như INTR, HOLD...

- Hoạt động của hệ: Dữ liệu được đưa vào hệ xử lý có thể từ các thiết bị nhớ ngoài hoặc trực tiếp qua cổng vào đưa vào RAM. Chương trình xử lý có thể là chương trình đã nạp sẵn trong ROM hoặc được nạp từ bộ nhớ ngoài vào RAM. CPU thực hiện chương trình theo chu trình:

- + Lấy lệnh
- + Giải mã lệnh
- + Điều khiển thực hiện lệnh

Trong quá trình thực hiện nếu có tác động ngắt hoặc yêu cầu DMA CPU sẽ đáp ứng các yêu cầu này sau đó lại quay trở lại chu trình hoạt động chính.

1.1.2. Máy tính PC hiện nay



Hình 1.2. Sơ đồ khối máy tính PC hiện nay

Máy tính PC hiện đại ngày nay gồm các thành phần được nối với nhau qua các chipset (Ví dụ trên hình 1.2). Kiến trúc này nhằm thực hiện mục đích phân chia các cấu thành có tốc độ làm việc tương đương nhau vào cùng một nhóm nhằm khai thác triệt để khả năng hoạt động của CPU và các thành phần cấu thành trong hệ thống.

- Chipset là 1 chip tích hợp chức năng của nhiều con chip làm nhiệm vụ điều khiển hoạt động giữa CPU và các lớp bộ nhớ, thiết bị ngoại vi có tốc độ làm việc khác nhau, giúp cho các lớp cấu thành có tốc độ làm việc, kiểu dữ liệu khác nhau có thể hoạt động đồng bộ trong hệ thống. Các chipset hiện nay được các hãng sản xuất mainboard tích hợp bằng công nghệ ASIC (Application Specific Integrated Circuit).

- Chipset cầu bắc nối các thành phần có tốc độ làm việc cao như cảm đồ hoạ AGP, SDRAM, CPU.

- Chipset cầu nam nối các thành phần có tốc độ tương đối cao như các thiết bị nối vào khe cắm PCI, USB, đĩa cứng ATA...

- Chipset phụ nối chipset cầu nam với các thiết bị có tốc độ chậm hơn như: bàn phím, đĩa mềm, cổng máy in song song, cổng truyền tin nối tiếp.

- Kiến trúc này cho phép toàn bộ hệ thống phát huy hết năng lực hoạt động của mỗi cấu thành mà không ảnh hưởng đến nhau do tốc độ làm việc khác nhau cũng như kiểu dữ liệu khác nhau.

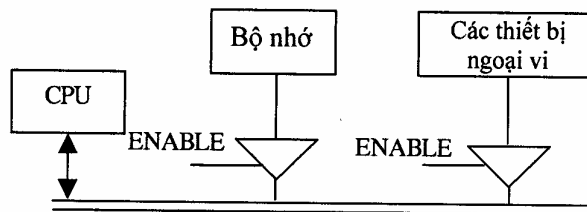
1.1.3. Hệ thống bus

- Hệ thống bus đã được giới thiệu ở trên gồm 3 thành phần (bus địa chỉ, bus dữ liệu và bus điều khiển). Trong đó bus dữ liệu có tính 2 chiều hoàn toàn có nghĩa là dữ liệu có thể từ CPU tới bộ nhớ hoặc thiết bị ngoại vi và ngược lại dữ liệu có thể từ bộ nhớ hoặc thiết bị ngoại vi tới CPU trên mỗi dây bit. Trên bus địa chỉ chiều của tín hiệu chỉ là từ CPU tới bộ nhớ hoặc thiết bị ngoại vi-bus địa chỉ có tính một chiều. Trên bus điều khiển chiều của tín hiệu với từng dây bit cũng là một chiều nhưng có thể là từ CPU ra như các tín hiệu điều khiển đọc/viết, tín hiệu trả lời ngắt, trả lời yêu cầu treo,... và cũng có thể từ bộ nhớ hoặc thiết bị ngoại vi tới CPU như tín hiệu yêu cầu ngắt, yêu cầu treo, yêu cầu đợi,... Có thể nói bus điều khiển có tính hai chiều không hoàn toàn.

- Thiết bị 3 trạng thái: Dùng để tránh xung đột bus.



Hình 1.3 Mạch 3 trạng thái



Hình 1.3. Thiết bị ngoại vi và bộ nhớ nối vào bus qua mạch ba trạng thái

Do việc bộ nhớ và các thiết bị ngoại vi dùng chung bus khi giao tiếp với CPU nên cần có cơ chế đảm bảo sao cho tại một thời điểm sử dụng chỉ có một thiết bị vật lý (bộ nhớ, thiết bị ngoại vi) nối với bus. Tín hiệu điều khiển ENABLE cho phép bộ nhớ hoặc thiết bị ngoại vi nào nối với bus. Thiết bị không nối với bus sẽ ở trạng thái trở kháng cao không gây ảnh hưởng đến thiết bị nối với bus.

Bảng 1.1

ENABLE	IN	OUT
0	0	0
1	1	1
1	0	z (trở kháng cao)
1	0	z (trở kháng cao)

Có hai phương pháp truyền thông tin trên bus là phương pháp đồng bộ và phương pháp không đồng bộ, trong máy tính PC sử dụng truyền đồng bộ là chủ yếu do không cách giữa các đối tượng nội bộ không lớn. Trong phương pháp này CPU điều khiển toàn bộ quá trình truyền thông tin qua các tín hiệu điều khiển ghi/đọc.

Quá trình truyền thông tin trên hệ thống bus được thực hiện trong các khoảng thời gian khác nhau. Chu kỳ bus là khoảng thời gian được CPU dùng để thực hiện một thao tác truyền thông tin nhất định với một đối tượng nhất định. Mỗi một chu kỳ bus kéo dài trên nhiều chu kỳ nhịp xung đồng hồ máy tính. Có các chu kỳ bus cơ bản như sau: nhập lệnh, đọc bộ nhớ, ghi bộ nhớ, đọc cổng vào/ra, ghi vào cổng vào/ra, ngắt.

1.2. Thiết bị ngoài của máy tính

1.2.1. Kiểu cơ - điện tử

Bao gồm các thiết bị được tạo nên từ nhiều thành phần phối hợp với nhau: Các thành phần cơ cấu cơ khí thường đảm nhiệm các chức năng theo yêu cầu chính của thiết bị như in, vẽ, đóng cắt thiết bị... Các thành phần cơ cấu điện, điện tử đảm nhiệm chức năng dẫn động, khuếch đại công suất. Việc phối hợp hoạt động giữa các cơ cấu điện - điện tử - cơ khí do các chương trình trong máy tính hoặc trong các hệ vi xử lý thực hiện.

1.2.2. Kiểu từ - điện tử

Thường được ứng dụng trong các thành phần lưu trữ thông tin. Đây là các vật thể trên bề mặt chứa lớp bột từ có tính từ dư. Thông tin cần lưu trữ được chuyển đổi sang các tín hiệu điện có mức điện áp "0" hoặc "1" sau đó cho các dòng điện này chạy trong cuộn dây đầu từ để hoá các lớp bột từ trên vật chứa thông tin.

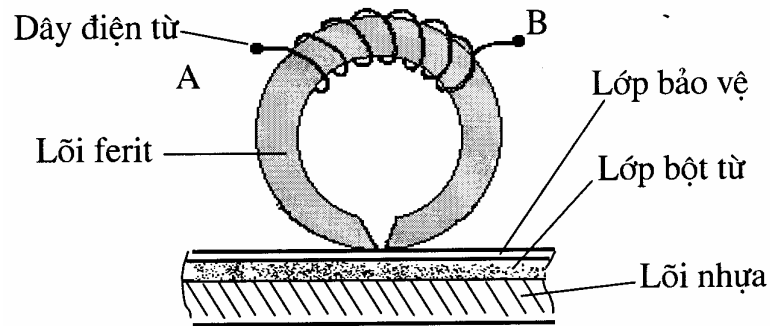
Đầu từ: làm chức năng viết và đọc ra thông tin trên vật chứa thông tin. Đầu từ được cấu tạo từ các vòng xuyên ferit có một khe hở để tập trung từ thông khi từ hoá các hạt từ trên bề mặt vật chứa thông tin.

Nguyên lý ghi_đọc từ:

- Gồm 2 thành phần chính:

- Đầu từ: Là 1 lõi ferit hình xuyên, có khe từ. Trên lõi có quấn cuộn dây điện từ.

Các đầu ra của cuộn dây nối vào mạch thu phát thông tin



Hình 1.4 Nguyên lý đọc ghi từ

- Đĩa từ: Là đĩa nhựa dẻo, trên bề mặt có phủ một lớp bột từ có đặc tính lưu giữ từ.

• Hoạt động:

- Ghi: Thông tin cần ghi vào đĩa ở dạng 0- 1 được biến đổi thành tín hiệu điện.

(Ví dụ theo chuẩn TTL: 0 : 0.. + 0,8 Volt

1 : + 2,8..+ 5Volt)

Các tín hiệu điện 0- 1 này chạy trên cuộn dây đầu từ sẽ tạo ra từ trường tỉ lệ với 0- 1. Trong khi đĩa từ quay quá trình từ hoá bề mặt đĩa theo thông tin đưa vào được thực hiện và kết quả là ở các vị trí khác nhau của đĩa sẽ lưu giữ các phần đĩa nhiễm từ với các mức độ khác nhau.

- Đọc: Ngược với quá trình ghi

1.2.3. Kiểu quang - điện tử

• Đĩa CD-ROM (Compact Disk Read Only Memory)

- Đĩa CD được phát minh vào năm 1982. Các tiêu chuẩn đầu tiên Reed Book hai hãng SONY và PHILIPS đưa ra. Với sự phát triển kỹ thuật các tiêu chuẩn nào cũng thay đổi; nhưng cơ bản vẫn dựa trên cơ sở Reed Book.

- Đĩa CD ngày nay không những được sử dụng trong lĩnh vực nghe nhìn mà còn được dùng làm bộ nhớ dung lượng lớn. Sự khác nhau giữa CD Player (các mạng nghe nhạc dùng đĩa CDRom) và bộ đọc đĩa CD ROM ghép với máy tính PC các bộ đọc CD ROM này có thêm bộ ghép nối để truyền số liệu tới bus hệ thống của PC và các linh kiện ghép nối đảm bảo cho CPU truy nhập các số liệu từ đĩa với những lệnh phần mềm.

Cấu tạo đĩa:

Đường kính: 4.75 inches

Dày: 1,2 mm

Lỗ ở giữa có đường kính: 15mm

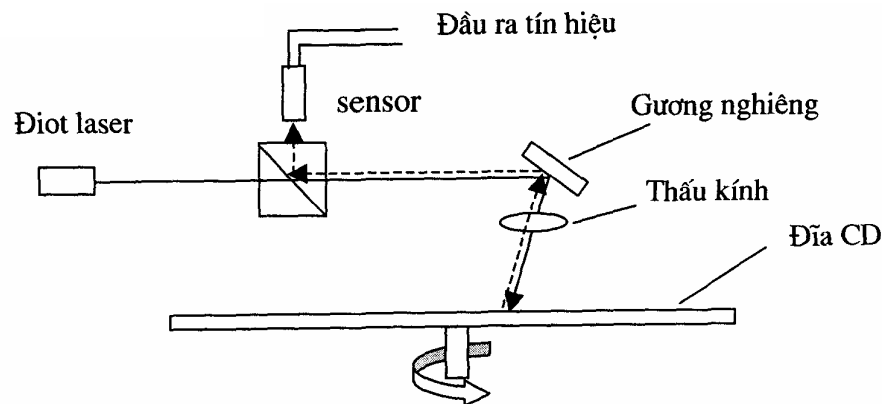
Dung lượng phổ biến hiện nay: 650MB -700MB

- Đĩa CD có những rãnh phản xạ ánh sáng được phủ bởi bột nhôm và sau đó phủ lớp sơn bóng để bảo vệ.

- Khi đĩa CD chế tạo, thông tin được đưa vào trong đĩa CD dưới các rãnh được phủ nhôm dưới dạng pits (Sự lõm xuống) và lands (Sự lồi lên); những lồi lõm này chính là biểu hiện của các bit. Pits và lands được sắp xếp dọc theo đường tròn ở quanh trục bao phủ toàn bộ bề mặt đĩa CD, lượn vòng từ trong ra ngoài. Không như (ra hát các loại đĩa CD bắt đầu ghi từ mép trong ra ngoài).

- Do có cấu tạo đặc biệt nên tốc độ truyền dữ liệu và thời gian thâm nhập của đĩa CD-ROM chưa cao so với đĩa cứng.

Nguyên tắc hoạt động:



Hình 1.5. Nguyên lý đọc đĩa CDROM

Sensor:	Cảm biến	Diode Laser:	Đĩa phát laser
Be am Splitter:	Bộ phân tích tia sáng	Bit Signal:	Tín hiệu số nhị phân(bit)
Renected be am:	Tia phản xạ ----->	Sensing beam:	Tia tới ———>
Movable Mirror:	Gương chuyển động	Optical Disk:	Đĩa quang

Tia laser từ một laser phát ra được hội tụ qua hệ thống quang học hội tụ lên các rãnh trên bề mặt đĩa CD- ROM. Ta đã biết thông tin được ghi bởi các pits và lands. Cường độ tia phản xạ sẽ yếu đi khi gặp chỗ lõm. Trong ổ đĩa có 1 sensor thu, nhạy với cường độ tia phản xạ. Cường độ tia phản xạ phụ thuộc vào các chỗ lồi lõm mà nó đi qua, tức là phụ thuộc các thông tin ghi trên đĩa. Đầu ra của sensor là các tín hiệu thông tin đã được chuyển sang dạng điện.

• Đĩa CD-WR (ổ đĩa CD ghi - đọc) và đĩa DVD (Digital Versatile Disc). Đĩa quang công nghệ số đa dụng: DVD xuất hiện đầu tiên vào tháng 3 năm 1997. Một đĩa DVD có thể lưu trữ thông tin gấp 7 lần một đĩa CD. Đầu đọc đĩa DVD có thể đọc được đĩa CD ROM (ngược lại thì không). DVD có đường kính và độ dày giống như đĩa CD, chúng cùng được làm bằng vật liệu như nhau và phương pháp chế tạo như nhau. Giống

như một đĩa CD, dữ liệu trong một DVD được mã hoá trong các chỗ lõm và lồi nhỏ dọc theo rãnh trên đĩa. Một DVD gồm có một vài lớp plastic, tổng cộng bề dày 1.2 mm. Mỗi một lớp được tạo bởi phun khuôn poly carbon plastic (*created by injection molding polycarbonate plastic*). Các quá trình này định dạng một đĩa có các bướu vi nhỏ (bumps) xếp đơn, liên tục và dài xoắn theo rãnh dữ liệu. Một khi xoá các các mẫu polycarbon đã được định dạng, một lớp phản xạ mỏng thì được thổi tới đĩa, bao phủ các bướu. Nhôm được sử dụng sau các lớp bên trong, nhưng một lớp vàng bán phản xạ thì được sử dụng cho các lớp ngoài, cho phép tia laser chiếu vào xuyên qua lớp ngoài tới lớp trong. Sau tất cả các lớp đã được làm, chúng được phủ sơn và ổn định bằng tia hồng ngoại. Với các đĩa một mặt, nhãn của chúng được in lựa ở mặt không có dữ liệu. Với đĩa hai mặt nhãn đĩa chỉ được in trên chỗ trống không ghi dữ liệu giữa đĩa. Mặt cắt các phần khác nhau của các kiểu đĩa khác nhau như các hình 1.6

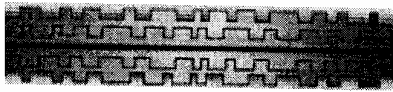
Một mặt, lớp đơn (4.7GB)



Một mặt, lớp kép (8.5GB)

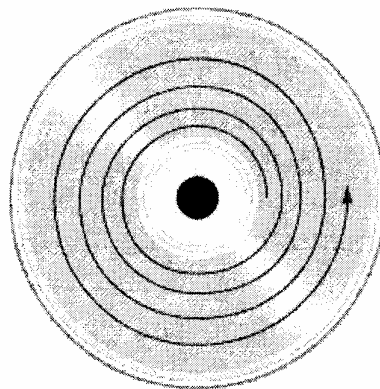


Hai mặt, lớp kép (17GB)

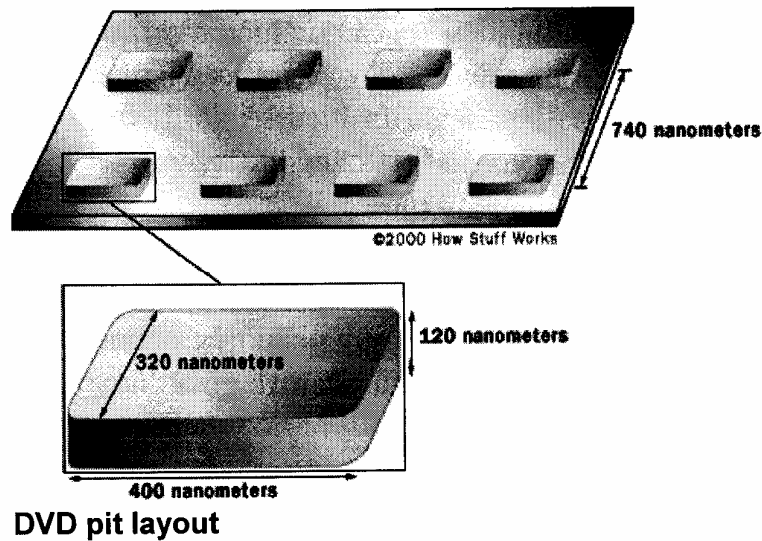


Hình 1.6. Các dạng DVD và dung lượng

Mỗi một lớp có thể viết của DVD có một rãnh xoắn chứa dữ liệu. Với loại DVD một mặt (single-layer DVD) rãnh xoắn luôn luôn chạy từ trong ra ngoài. Như vậy rãnh xoắn bắt đầu ở tâm nghĩa là DVD một mặt có thể nhỏ hơn 12c nếu yêu cầu.



Hình 1.7. Đường ghi dữ liệu trên DVD



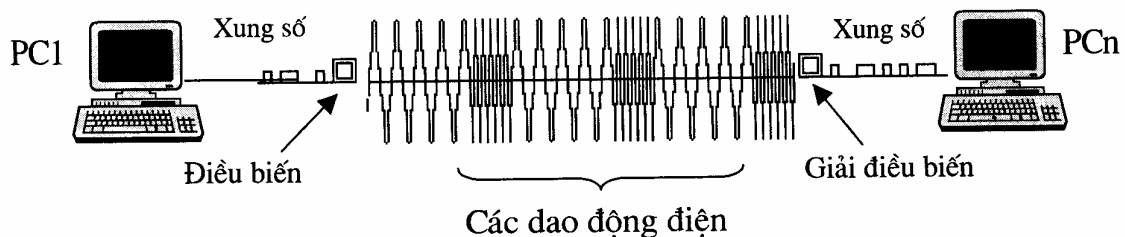
Hình 1.8. Dữ liệu trên DVD

Trên đĩa DVD các pits được thay bằng các các bướu (bumps) trên rãnh dữ liệu. Chúng xuất hiện như các pits trên mặt nhôm, nhưng ở phía tia laser đọc tới đúng là các bumps. Quan sát bằng kính hiển vi có thể thấy các bướu trên rãnh xoắn của DVD. Nếu trải rãnh xoắn này ra có thể tới chiều dài 48km (với DVD hai mặt, lớp kép). Để đọc được các bướu nhỏ này cần phải có các cơ cấu cơ-điện tử chính xác trong thiết bị đọc đĩa.

1.2.4 Thiết bị xử lý tín hiệu

Có thể kể ra đây các thiết bị như modem, các thiết bị phục vụ cho mạng, các bộ A/D, DIA. (Các bộ A/D và D/A sẽ được khảo sát kỹ ở chương sau).

- MODEM: Có gốc từ các chữ tiếng Anh Modulator DEModulator. Là 1 thiết bị sử dụng để kết nối máy tính với đường truyền điện thoại, thông qua đó trao đổi thư điện tử, truy nhập mạng Internet hoặc thực hiện bài toán đo lường điều khiển tới 1 vị trí ở xa...



Hình 1.9. Hoạt động của MODEM

Tín hiệu từ máy PCI ra là tín hiệu số, các tín hiệu này qua bộ phận điều biến là tín hiệu Analog được truyền trên đường dây điện thoại. Trước khi vào máy PCn được giải điều biến(Điều chế) lại thành tín hiệu số để thích hợp với PCn. Quá trình truyền và

nhận tín hiệu từ PCn đến PCI cũng tương tự.

- Thiết bị phục vụ cho mạng máy tính:

- REPEATER: Bộ chuyển tiếp tín hiệu. Repeater làm việc ở tầng vật lý trong mô hình OSI. Nó có một cổng vào và một cổng ra. Chức năng chính của nó là khuếch đại tín hiệu, bù lại những suy giảm trên đường truyền. Repeater không có khả năng liên kết các LAN khác nhau về giao thức truyền thông ở tầng liên kết dữ liệu.

- HUB: Là repeater nhiều cổng. Nó thực hiện chuyển tiếp tất cả các tín hiệu điện đến từ một cổng tới tất cả các cổng còn lại sau khi đã khuếch đại.

- ROUTER: Bộ chọn đường - làm việc trên tầng network của mô hình OSI. Router thường có nhiều hơn 2 cổng. Nó tiếp nhận tín hiệu vật lý từ một cổng, chuyển đổi về dạng dữ liệu, kiểm tra địa chỉ của mạng rồi chuyển dữ liệu đến cổng tương ứng. Router dùng các bảng chọn đường (routing table) để lưu trữ các ánh xạ cổng - địa chỉ network. Router dùng để liên kết các LAN có thể khác nhau và chuẩn LAN nhưng dùng một giao thức mạng ở tầng Network. Chẳng hạn hai mạng Ethernet và Token ring cùng với giao thức truyền thông TCP/IP. Router cũng dùng để liên kết hai mạng ở rất xa nhau chẳng hạn bằng leased line hoặc ISDN. Trong trường hợp như vậy mỗi mạng phải dùng một router. Các route- dùng cho mục đích này gọi là remote router.

- BRIDGE: Làm việc với tầng thứ hai của mô hình OSI - tầng liên kết dữ liệu. Nó được thiết kế để có khả năng nhận tín hiệu vật lý, chuyển đổi về dạng dữ liệu và chuyển tiếp dữ liệu. Bridge có 2 cổng. Sau khi nhận tín hiệu vật lý và chuyển đổi về dạng dữ liệu từ một cổng, bridge kiểm tra địa chỉ đích, nếu địa chỉ này là của một node liên kết với chính cổng nhận tín hiệu, nó bỏ qua việc xử lý. Trong trường hợp ngược lại dữ liệu được chuyển tới cổng còn lại, tại cổng này dữ liệu được chuyển đổi thành tín hiệu vật lý và gửi đi. Để kiểm tra một node được liên kết với cổng nào của nó, bridge dùng một bảng địa chỉ cập nhật động. Vì nguyên lý hoạt động nói trên mà tốc độ truyền thông qua 2 cổng của bridge chậm hơn so với repeater. Bridge được dùng để liên kết các LAN có cùng giao thức tầng liên kết dữ liệu. Các LAN này có thể khác nhau về môi trường truyền dẫn vật lý Bridge cũng được dùng cho mục đích chia một LAN thành nhiều LAN thành phần, và mỗi mạng LAN thành phần này tập hợp những node có tần suất giao tiếp với nhau thường xuyên, các mạng LAN thành phần có nhu cầu giao tiếp với nhau không thường xuyên.

- SWITCH: Làm việc như một Bridge nhiều cổng. Khác với HUB - nhận tín hiệu từ một cổng rồi chuyển tiếp tới tất cả các cổng còn lại, switch nhận tín hiệu vật lý, chuyển đổi thành dữ liệu, từ một cổng, kiểm tra địa chỉ đích rồi gửi tới một cổng tương ứng. Switch được thiết kế để liên kết các cổng của nó với bảng thông rất lớn.

1.3. Các chuẩn ghép nối

1.3.1. Định nghĩa

- Đ/n: Các tiêu chuẩn ghép nối bao gồm các tiêu chuẩn về cơ khí và các thông số điện và các thủ tục về truyền tin sao cho thông tin ở các thiết bị ngoại vi và các hệ thống xử lý khác nhau có thể trao đổi được với nhau một cách an toàn và thuận tiện.
- Phân bố địa chỉ cổng vào/ra trong máy vi tính:

Theo kiến trúc máy tính IBM-PC 1 KB địa chỉ đầu tiên trong bản đồ bộ nhớ được dành cho địa chỉ của các thiết bị ngoại vi

Bảng 1.2. Sắp xếp của vùng địa chỉ vào/ra của máy tính PC

Địa chỉ (hexa) vào ra	Chức năng
000-01F	Bộ điều khiển DMA 1 (8232)
020-03F	Bộ điều khiển ngắt (8259)
040-043	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)
0C-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 trò chơi
278-27F	Cổng song song 2 (LPT2)
2B0-2DF	Card EGA 2
2E8-2EF	Cổng nối tiếp 4 (COM4)
2F8-2FF	Cổng nối tiếp 2 (COM2)
300-31F	Dùng cho cam mở rộng
320-32F	Bộ điều khiển đĩa cứng
3 60-3 6F	Cổng nối mạng LAN
378-37F	Cổng song song 1 (LPT1)
3 8 0-3 8F	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 (COM3)
3F0-3F7	Bộ điều khiển đĩa mềm
3F8-3FF	Cổng nối tiếp 1 (COM1)

1.3.2. Các đặc tả cho từng kiểu ghép nối

- Chuẩn RS232, 485:

Sau 1 thời gian dài không chính thức, đến năm 1962. Hiệp hội các nhà công nghiệp điện tử (*EIA: The Electronics Industries Association*) đã cho ban hành tiêu chuẩn RS-32 áp dụng cho cổng nối tiếp (RS Recommended Standard - Tiêu chuẩn đã giới thiệu). Có 2 loại: RS 232B và RS232C. Tiêu chuẩn đang được áp dụng hiện nay là RS232C. Ở các nước Tây Âu chuẩn này còn gọi là chuẩn V.24

- Chuẩn cơ khí: Dùng các đầu nối DB25 hoặc DB9 Sẽ giới thiệu kỹ ở chương 5

- Chuẩn về điện:

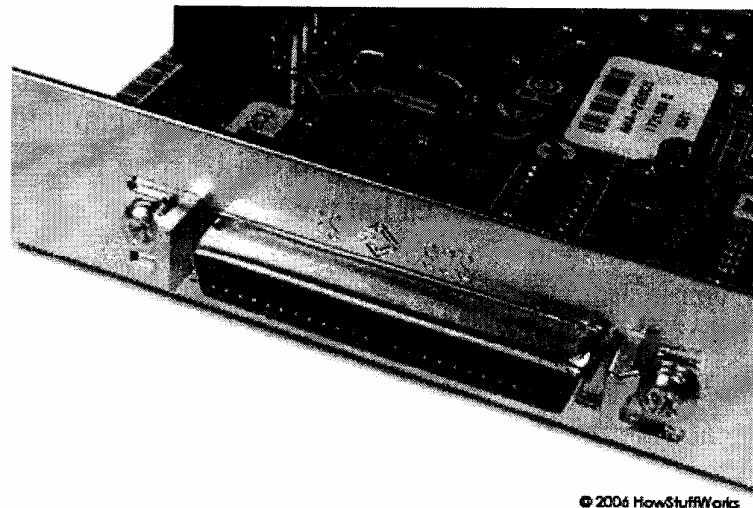
- + Mức "1" các điện áp từ -3V đến -12V. Mức "0" các điện áp từ +3V đến + 12V

- + Tốc độ truyền nhận cực đại 100 kbit/giây

- + Tốc độ truyền chuẩn: 50, 75, 110, 150, 300... 1200, 2400, 4800, 9600, 19.200, 28.800... 56.000baud,

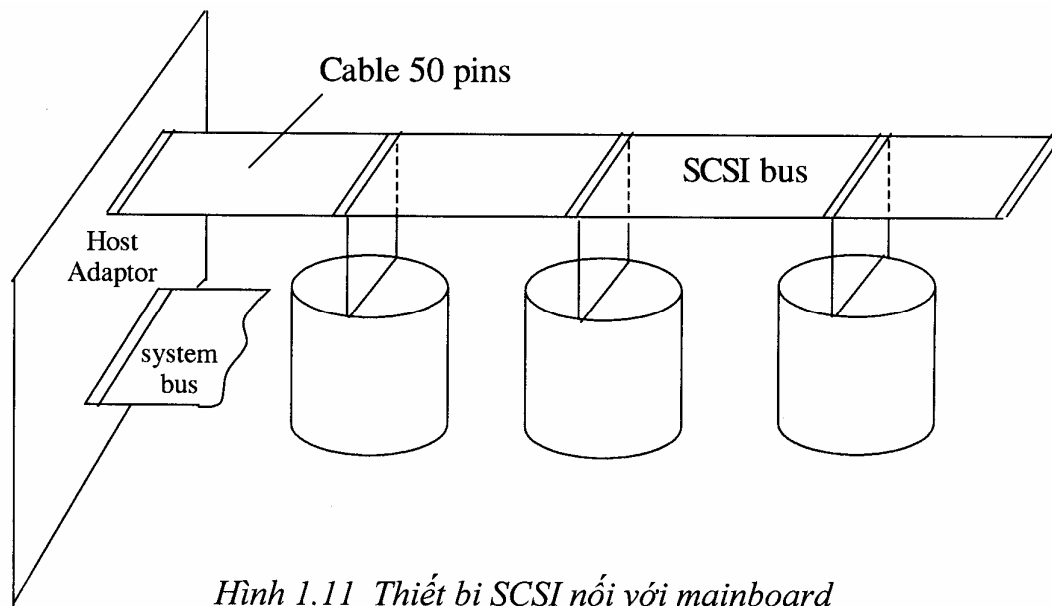
SCSI: *Small Computer System Interface*. Do Macintosh đề xuất năm 1984. Tốc độ chuyển vận dữ liệu max >320MB/s. Cho phép quản lý tới 16 thiết bị

- Chuẩn cơ khí: 50 chân (có thể biến thể 68 hoặc 80 chân với một số loại SCSI khác),



Hình 1.10. Đầu nối SCSI 50 chân

- Cấu tạo:



Hình 1.11 Thiết bị SCSI nối với mainboard

SCSI có ba loại cơ bản:

- **SCSI-1:** Loại nguyên thủy được phát triển từ năm 1984, SCSI-1 hiện nay đã lỗi thời nó có độ rộng bus 8 bit và tốc độ xung nhịp làm việc 5 MHz.

- **SCSI-2:** Ra đời 1994, có đặc điểm kỹ thuật sử dụng *Common Command Set* (CCS) - gồm 18 lệnh dùng chung trợ giúp cho các loại SCSI. Có thể sử dụng xung nhịp đồng hồ gấp đôi tới 10MHz (*chế độ fast*) hoặc sử dụng chế độ gấp đôi bề rộng bus dữ liệu (*chế độ wide*) hoặc chế độ gấp đôi cả hai thuộc tính (*fast/wide*). SCSI-2 cũng có thể thêm vào lệnh hàng đợi (*command queuing*), cho phép thiết bị lưu trữ và phân quyền ưu tiên từ máy tính chủ (host computer).

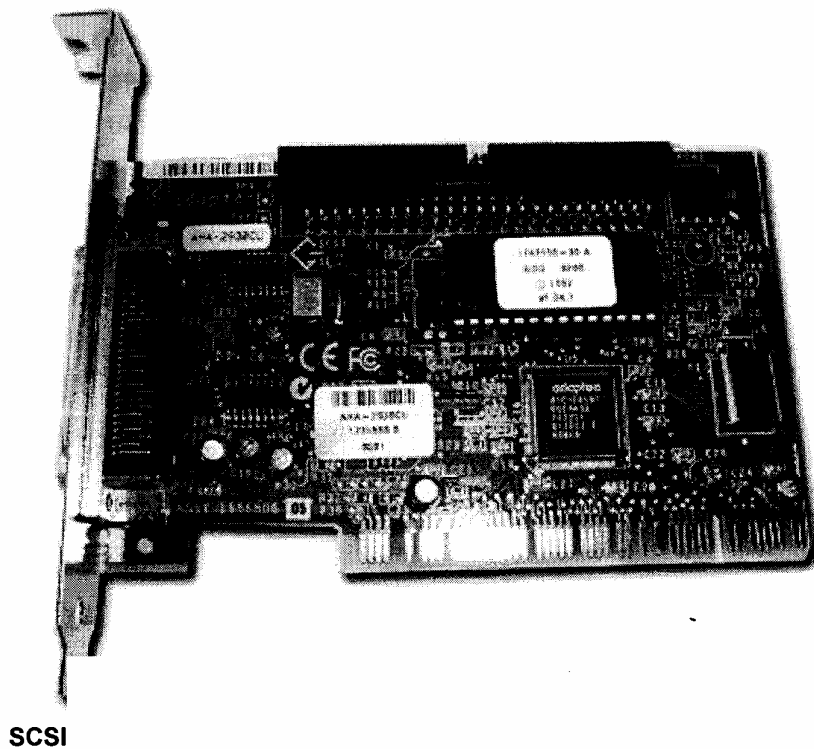
SCSI-3: Xuất hiện đầu tiên 1995. Một tập các tiêu chuẩn được gọi là *SCSI Parallel Interface* (SPI), là cách các thiết bị SCSI truyền tin với các thiết bị khác. Đặc tả kỹ thuật phổ biến của SCSI-3 gắn thuật ngữ Ultra, như Ultra với SPI, Ultra2 với SPI-2 và Ultra3 với SPI-3.

Bảng 1.3

Tên	Đặc tả kỹ thuật	Số thiết bị	Độ rộng bus	Tốc độ bus	MBps
Asynchronous	- SCSI-1	8	8 bits	5 MHz	4 MBps
Synchronous SCSI	- SCSI -1	8	8 bits	5 MHz	5 MBps
Wide	- SCSI -2	16	16 b its	5 MHz	10 MBps
Fast	- SCSI -2	8	8 bits	10 MHz	10
Fast/wide	- SCSI -2	16	16 b its	10 MHz	20 MBps

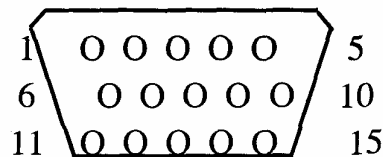
Ultra	SCSI-3 SPI	8	8 bits	20 MHz	20 MBps
Ultra/Wide	SCSI-3 SPI	8	16 bits	20 MHz	40 MBps
Ultra2	SCSI-3 SPI	8	8 bits	40 MHz	40 MBps
Ultra2/Wide	SCSI-3 SPI2	16	16 bits	40 MHz	80 MBps
Ultra3	SCSI-3 Spl-3	16	16 bits	40 MHz	160 MBps
Ultra320	SCSI-3 Spl-4	16	16 bits	80 MHz	320 MBps

Các kiểu SCSI truyền dữ liệu bit song song. Ultra320 SCSI sử dụng dữ liệu kiểu đóng gói (packeted data). Kiểu SCSI mới nhất hiện nay được gọi là *Serial Attached SCSI* (SAS), sử dụng các lệnh SCSI nhưng dữ liệu truyền nối tiếp. SAS sử dụng liên kết nối tiếp point-to-point di chuyển dữ liệu với tốc độ 3.0 gigabits trên giây, mỗi một cổng SAS có thể hỗ trợ hoạt động cho 128 thiết bị.



Hình 1.12. SCSI controller

- USB: Universal Serial Bus được phát minh 1997. Chuẩn công nghiệp mới này dùng 1 loại đầu nối vừa với tất cả để thay cho mọi cổng cũ khác trên PC. Ta có thể cắm nhiều loại thiết bị vào cổng USB: màn hình, bàn phím, chuột, modem, máy in, máy quét... Ta còn có thể cắm 1 chuỗi thiết bị ngoại vi cái này nối cái kia, nghĩa là ta có 1 chuỗi thiết bị chạy từ 1 cổng duy nhất trên PC. Một số sản phẩm USB như máy quét và Camera số có thể hoạt động không cần dây cắm điện riêng - Dây nối USB có khả năng cung cấp nguồn điện. Cổng USB hoạt động nhanh gấp 10 lần cổng song song, gấp 100 lần cổng nối tiếp. Phiên bản USB hiện nay đang sử dụng là USB 2.0.
- ISA: Rãnh cắm theo tiêu chuẩn ISA Industry Standard Architecture. có 32 chân x 2= 64 trên 2 mặt. Bề rộng bus dữ liệu 8 bit. Tốc độ truyền dữ liệu lớn nhất max= 8Mbit/giây với loại ISA mở rộng: 16 bit và với EISA: 32 bit. Vùng địa chỉ thiết bị ngoại vi dành cho các card mở rộng từ 300h đến 31Fh. Từ 1999 khe cắm ISA không còn được sử dụng cho máy tính PC nữa.
- PCI: Do công ty Intel xây dựng nên. PCI hoạt động với bề rộng bus dữ liệu 32 bit. Hiện nay loại PCIX phát triển hơn hoạt động với bề rộng bus dữ liệu 64 bit.
- EIDE: Đầu nối cơ khí 40 chân (*pins*) thường được sử dụng trong các máy tính và trong các thiết bị recorder, các thiết bị thu thập dữ liệu. Để ghép nối các thiết bị với hệ trung tâm cần có IDE Controller. Một dây nối IDE cho phép cắm 2 thiết bị trên nó (một master, một slaver).
- Giao diện cho thiết bị hiển thị chuẩn



Hình 1.13. Các tín hiệu trên đầu nối ra màn hình CRT

- | | |
|---------------------------------|----------------------------|
| 1: Red Video | 2: Green Video |
| 3: Blue Video | 4: Ground |
| 5: Selftest | 6: Red Ground |
| 7: Green Ground | 8: Blue Ground |
| 9: No Connection | 10: Digital Ground |
| 11: Ground | 12: Reserved (SDA for DDC) |
| 13: Horizontal Sync. | 14: Vertical Sync. |
| 15: No Connection (SCL for DDC) | |

- Giao diện cho ổ đọc đĩa mềm: 34 chân dùng để nối bộ điều khiển đĩa mềm FDC

Controller với ổ đọc đĩa mềm

1.4. Các phương pháp vào ra dữ liệu

CPU thực hiện trao đổi thông tin với các thiết bị ngoại vi và thế giới bên ngoài thông qua thiết bị giao diện.

Thiết bị giao diện là thiết bị có thể lập trình. Mỗi thiết bị giao diện đều có 3 loại thanh ghi, mỗi loại thực hiện 1 chức năng khác nhau:

- + data register (thanh ghi dữ liệu): thực hiện chức năng bộ đệm tạm chứa dữ liệu vào/ra.

- + status register (thanh ghi trạng thái): chứa thông tin phản ánh trạng thái làm việc của thiết bị giao diện và thiết bị ngoại vi

- + control register (thanh ghi điều khiển): nhận và chứa các từ điều khiển xác lập chế độ làm việc của thiết bị.

Mỗi một thanh ghi có 1 địa chỉ xác định gọi là địa chỉ cổng.

Khi CPU đưa một dữ liệu ra ngoài (thực hiện bằng lệnh OUT xuất dữ liệu ra cổng có địa chỉ xác định) thực chất là CPU đưa dữ liệu ra thanh ghi dữ liệu của thiết bị giao diện, thiết bị giao diện sẽ chuyển nó thành dạng thích hợp với thiết bị ngoại vi rồi mới đưa ra ngoài cho thiết bị ngoại vi.

Khi thiết bị ngoại vi gửi một dữ liệu cho máy tính, dữ liệu này được đưa vào thanh ghi dữ liệu trong thiết bị giao diện. CPU nhập dữ liệu từ ngoài bằng cách đọc thanh ghi dữ liệu đệm này.

Thiết bị giao diện chỉ giúp CPU kết nối một cách thích hợp về mặt vật lý với các thiết bị bên ngoài, nhưng chưa đảm bảo tính tin cậy của quá trình trao đổi thông tin. Điều này xuất phát từ một thực tế khách quan là nhịp làm việc của CPU khác xa với nhịp và tốc độ làm việc của thiết bị ngoại vi. Để CPU có thể thực hiện trao đổi thông tin với các thiết bị ngoại vi với độ tin cậy cao cần phải áp dụng các phương pháp trao đổi dữ liệu thích hợp, các phương pháp này được gọi là các phương pháp vào/ ra dữ liệu,

Có thể phân chia các phương pháp vào/ra dữ liệu thành 2 nhóm:

- Vào ra do CPU chủ động:

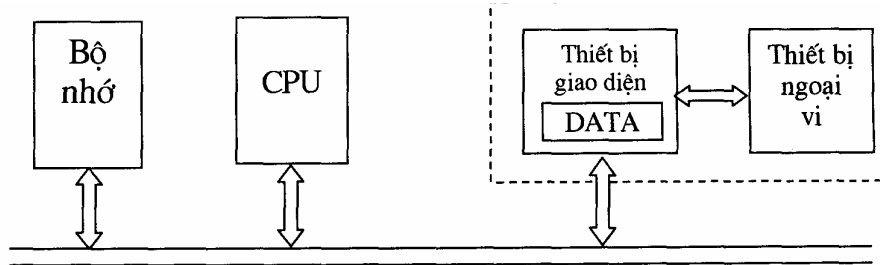
- + Vào ra theo định trình
- + Vào ra có thăm dò (phương pháp hỏi vòng - polling)

- Vào ra do thiết bị ngoại vi chủ động:

- + Vào ra bằng ngắt
- + Vào ra DMA

1.4.1 Vào/Ra theo định trình

Đây là phương pháp mà trong đó quá trình vào/ra dữ liệu được thực hiện theo một chu kỳ xác định trước, nhờ các lệnh vào/ra (lệnh IN hoặc OUT) và CPU không quan tâm đến trạng thái của thiết bị vào/ra (bao gồm thiết bị giao diện và thiết bị ngoại vi).

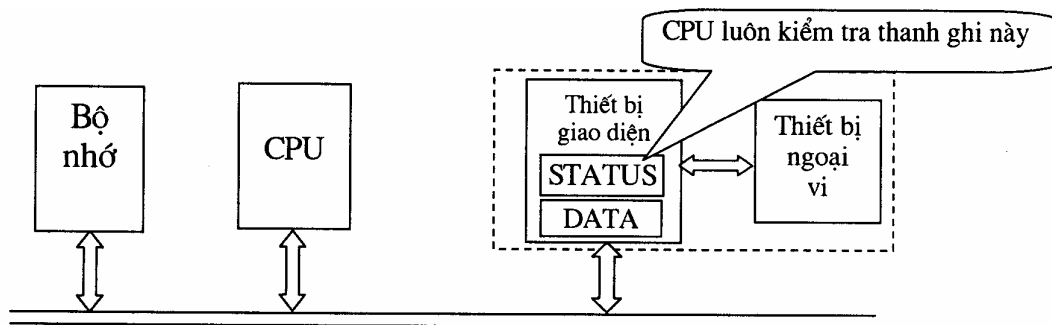


Hình 1.14. Sơ đồ vào ra theo định trình

Phương pháp vào/ra theo định trình thích hợp với những quá trình vào/ra có chu kỳ cố định và có thể xác định trước.

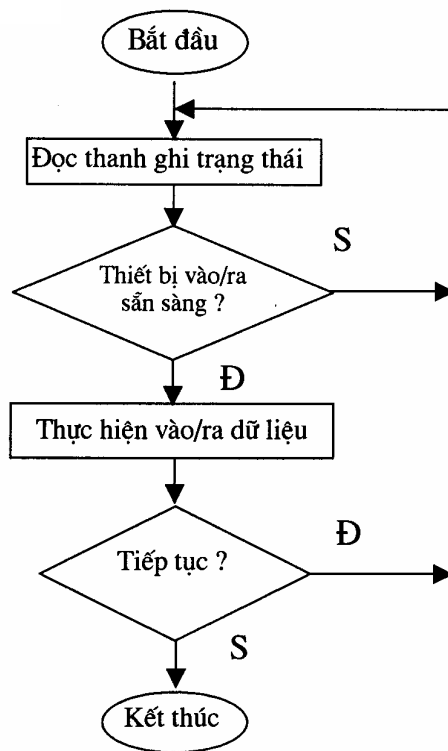
1.4.2. Vào/Ra theo phương pháp hỏi vòng

Trong mỗi thiết bị giao diện thường có ít nhất một thanh ghi trạng thái chứa thông tin phản ánh trạng thái làm việc của thiết bị này và thiết bị ngoại vi. Khi thực hiện phương pháp vào/ra có thăm dò, chương trình vào ra dữ liệu luôn thực hiện kiểm tra trạng thái sẵn sàng làm việc của thiết bị trước khi thực hiện thực sự việc vào/ra dữ liệu.



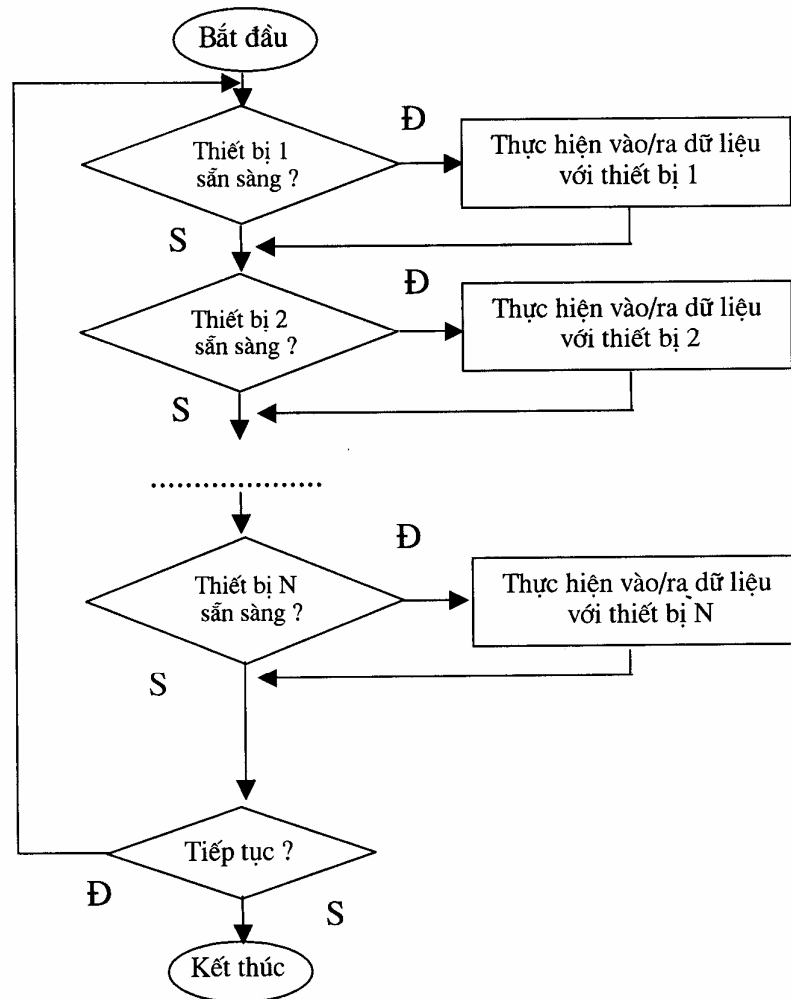
Hình 1.15. Sơ đồ vào ra kiểu thăm dò

Quá trình vào/ra dữ liệu có thăm dò như sau:



Hình 1.16. Thuật toán vào/ ra kiểu thăm dò với một thiết bị

Quá trình vào/ra với nhiều thiết bị theo phương pháp thăm dò:



Hình 1.17. Vào ra dữ liệu kiểu thăm dò với nhiều thiết bị

Ưu điểm của phương pháp này: Do CPU luôn kiểm tra trạng thái sẵn sàng làm việc của thiết bị trước khi thực hiện vào/ra nên vào/ra dữ liệu kiểu này có độ tin cậy cao.

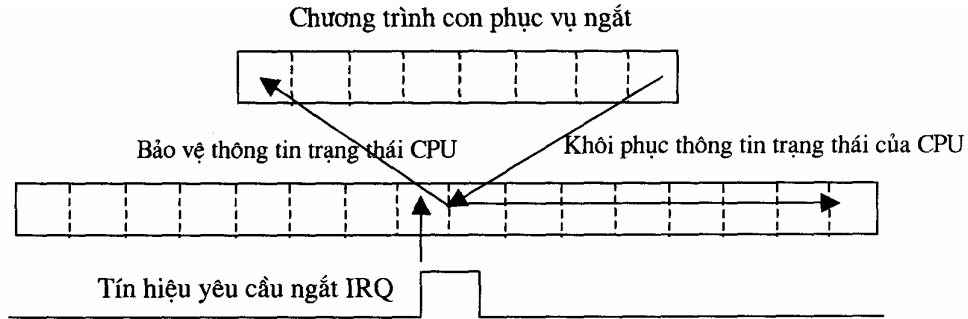
Nhược điểm: Do CPU luôn phải kiểm soát lần lượt trạng thái làm việc của các thiết bị nên tốc độ vào/ra dữ liệu chậm. Hơn nữa nếu CPU chỉ làm 1 công việc là vào/ra dữ liệu (theo phương pháp thăm dò) thì hiệu quả không cao, ngược lại nếu CPU đồng thời phải thực hiện nhiều loại công việc hơn thì thời gian làm việc của

CPU sẽ bị chia xẻ, đồng thời độ tin cậy của phương pháp vào/ra theo thăm dò cũng bị giảm đi rất nhiều.

1.4.3. Vào / Ra bằng ngắt

Ngắt là việc CPU dừng chương trình chính đang thực hiện để thực hiện một chương trình con khác là chương trình con phục vụ ngắt (ISR Interrupt Service routine). Có hai loại ngắt cứng và ngắt mềm. Vào ra dữ liệu bằng ngắt cứng là phương pháp vào/ra dữ liệu trong đó thiết bị vào/ra chủ động quá trình vào/ra dữ liệu nhờ hệ

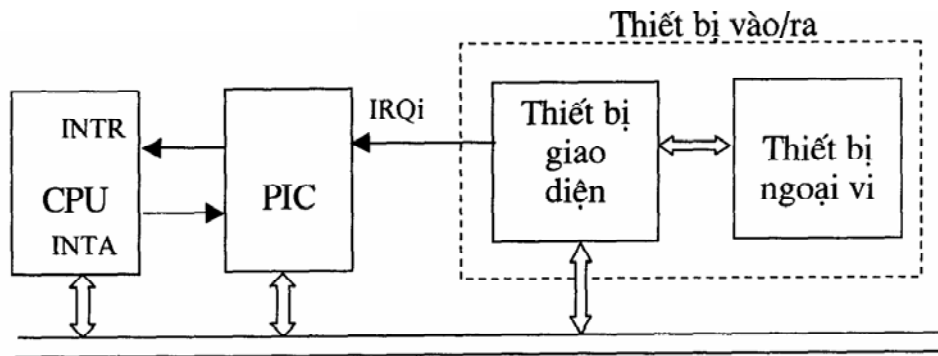
thống ngắt cứng.



Hình 1.18. Sơ đồ quá trình ngắt

Thông thường quá trình vào/ra bằng ngắt cứng được trợ giúp bởi thiết bị điều khiển ngắt PIC (Programmable Interrupt Controller). PIC có chức năng ghi nhận các yêu cầu ngắt IRQ và cung cấp cho CPU số hiệu ngắt đại diện cho địa chỉ của chương trình con phục vụ ngắt và tương ứng yêu cầu ngắt IRQ.

Cấu trúc của hệ thống ngắt cứng:



Hình 1.19. Bộ điều khiển ngắt PIC trong hệ thống

Quá trình vào/ra dữ liệu bằng ngắt cứng:

- CPU đang thực hiện chương trình chính
- Thiết bị vào/ra có yêu cầu phục vụ phát ra tín hiệu IRQ cho PIC.
- Thiết bị PIC phát ra tín hiệu INT cho CPU, yêu cầu CPU phục vụ
- CPU hoàn thành nốt lệnh đang thực hiện
- CPU phát tín hiệu INTA (Interrupt Acknowledge) trả lời PIC, báo sẵn sàng phục vụ quá trình ngắt.
- PIC phát ra số hiệu ngắt (là con số đại diện cho địa chỉ của chương trình con phục vụ ngắt, và tương ứng với tín hiệu IRQ) cho CPU.
- Dựa trên số ngắt này CPU kích hoạt và thực hiện chương trình con phục vụ ngắt

ISR để thực hiện vào/ra dữ liệu

- Khi ISR kết thúc thì CPU quay lại tiếp tục thực hiện chương trình chính đang thực hiện,

Ưu điểm của phương pháp này:

- CPU thực hiện vào/ra dữ liệu ngay khi có yêu cầu từ thiết bị vào/ra. Điều này làm cho quá trình vào/ra dữ liệu có độ tin cậy rất cao. -

- CPU chỉ phục vụ thiết bị vào/ra khi có yêu cầu (khi thiết bị vào/ra đã sẵn sàng cho việc truyền dữ liệu), do vậy làm tăng hiệu quả làm việc của CPU. Do những ưu điểm này mà phương pháp vào/ra dùng ngắt cứng được dùng để vào/ra dữ liệu với phần lớn các thiết bị ngoại vi chuẩn của máy tính như: bàn phím, máy in, thiết bị vào ra nối tiếp...

Tuy nhiên với phương pháp này quá trình chuyển dữ liệu giữa bộ nhớ và thiết bị vào/ra vẫn phải qua CPU và quá trình vào/ra dữ liệu vẫn do CPU thực hiện, nên đây chưa phải là phương pháp vào/ra nhanh nhất.

MỘT SỐ VÍ DỤ THAY VECTO NGẮT

- Lập trình bằng Pascal

```
uses dos,crt;
```

```
var oldvec: pointer; { chứa địa chỉ của chương trình xử lý ngắt }
```

```
R: Registers;
```

```
procedure thayngat; interrupt;
```

```
begin writeln(" Hello ");
```

```
end;
```

```
asm
```

```
mov ah,4Ch
```

```
int 21h
```

```
end;
```

```
end;
```

```
BEGIN
```

```
getintvec($ 13,oldvec);
```

```
setintvec($13,Add(thayngat));
```

```
intr($ 13,r);
```

```
readln;
setintvec($13,oldvec);
```

END.

- Lập trình bằng C:

```
#include <stdio. h>
#include <dos. h>
#include <conio. h>
#define INTR OX1C /* The clock tick interrupt */
#ifdef _cplusplus
#define _CPPARGS...
#else
#define CPPARGS
#endif
void interrupt (*oldhandler)(_CPPARGS); int count 0;
void interrupt handler(_CPPARGS)
{
    /* increase the global counter */
    count++;
    /* call the old routine */
    oldhandler();
}
int main(void)
{ /* save the old interrupt vector */
    oldhandler = getvect(INTR);
    /* install the new interrupt handler */ setvect(INTR, handler);
    /* loop until the counter exceeds 20 */
    while (count < 20)
        printf("count is %d\n",count);
    /* reset the old ~nterrupt handler */
    setvect(INTR, oldhandler);
    return 0;
}
```

- Lập trình bằng hợp ngữ:

Giới thiệu các hàm 35h, 25h của ngắt 21h:

+ Hàm 25h: Thiết lập vectơ ngắt

Vào: AH=25h, AL: Số hiệu ngắt

DS:DX= vectơ ngắt

Ra: không

+ Hàm 35h: Lấy vectơ ngắt

Vào: AH:35h

AL: số hiệu ngắt

Ra: ES:BX: vectơ ngắt

; Thủ tục cắt vectơ ngắt cũ và thiết lập vectơ ngắt mới

; vào: AL: số hiệu ngắt

DI = địa chỉ bộ đệm vectơ ngắt cũ

SI = địa chỉ bộ đệm chứa vectơ ngắt mới

; cắt cũ:

mov an,35h

mov ai, số hiệu ngắt cần lấy

mov [di],bx

mov [DI+2], ES

; thiết lập mới

MOV DX,[SI]

PUSH DS

MOV DS,[SI+2]

MOV AH,25h

máy al, số hiệu ngắt cần thiết lập

INT 21h

POP DS

RET

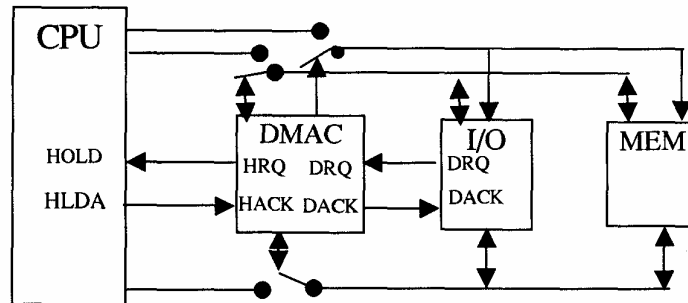
1.4.4. Vào / Ra theo phương pháp DMA

Có nhiều quá trình trao đổi dữ liệu đòi hỏi tốc độ vào/ra dữ liệu nhanh hơn khả năng vào/ra dữ liệu của các phương pháp vào/ra dữ liệu bằng chương trình như đã

trình bày ở trên, ví dụ quá trình chuyển dữ liệu từ các ổ đĩa vào bộ nhớ và ngược lại. Lúc này có thể sử dụng phương pháp vào/ra dữ liệu kiểu DMA, là quá trình vào/ra dữ liệu trực tiếp giữa bộ nhớ và thiết bị ngoại vi không qua CPU.

Trong quá trình DMA việc chuyển dữ liệu không được điều khiển bởi CPU mà bởi một thiết bị phần cứng là bộ điều khiển DMAC (DMA Controller).

Sơ đồ quá trình như sau:



Hình 1.20. Vào ra dùng DMA

- DMAC được xác lập chế độ làm việc, nhận thông tin về địa chỉ đầu khối nhớ chứa dữ liệu và kích thước khối dữ liệu cần truyền.

- CPU đang hoạt động bình thường nếu giữa thiết bị vào/ra và bộ nhớ có yêu cầu DMA thì:

- Thiết bị vào/ra phát tín hiệu DRQ cho DMAC
- DMAC phát tín hiệu $HRQ = 1$ đến chân HOLD của CPU, yêu cầu CPU đi vào chế độ DMA (yêu cầu treo CPU).
- CPU thực hiện nốt chu kỳ máy.
- CPU phát tín hiệu HLDA trả lời cho DMAC và tự tách ra khỏi hệ thống bus.
- Quyền điều khiển hệ thống bus thuộc về DMAC
- DMAC làm chủ bus hệ thống, tạo tín hiệu DACK trả lời thiết bị yêu cầu, phát địa chỉ ô nhớ lên bus địa chỉ, phát các tín hiệu điều khiển ghi/đọc thiết bị vào/ra và các tín hiệu điều khiển ghi/đọc bộ nhớ và thực hiện điều khiển toàn bộ quá trình chuyển dữ liệu giữa thiết bị vào/ra và bộ nhớ.
- Kết thúc quá trình DMA, DMAC gửi tín hiệu $HRQ = 0$ đến chân HOLD trả lại quyền điều khiển bus cho CPU,

- CPU tiếp tục trở lại kiểu hoạt động bình thường.

1.5. Các thiết bị chuyển đổi dữ liệu:

1.5.1. Khái niệm - Định lý lấy mẫu của Shannon

Trong thực tế, các loại thông tin dữ liệu hầu hết ở dạng tương tự và liên tục' theo thời gian. Song thế giới bên trong máy vi tính hoàn toàn bằng số, và rời rạc. Để đưa các thông tin dữ liệu tương tự vào máy tính số cần phải có một thiết bị, có khả năng chuyển các tín hiệu tương tự này thành tín hiệu số. Thiết bị đó gọi là các bộ chuyển đổi AD (ADC- Analog Digital Convertor).

Ngược lại, để khôi phục các mẫu tín hiệu đã lấy mẫu và sau khi xử lý đưa ra môi trường bên ngoài ta cần bộ chuyển đổi D/A (DAC - Digital Analog Convertor).

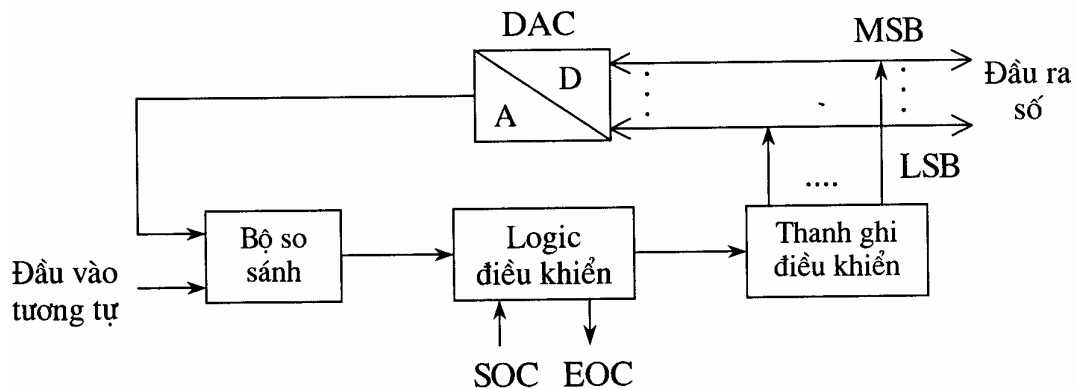
Bộ chuyển đổi tương tự - số làm nhiệm vụ chuyển đổi những thông tin, dữ liệu tương tự biểu diễn đặc tính của các đại lượng vật lý trong thế giới tự nhiên sang dạng mã số. Mã số được dùng trong quá trình xử lý tin, tính toán trong hệ thống máy tính và các hệ thống đo điều khiển số. Nó thực hiện hai chức năng cơ bản là lượng tử hoá và mã hoá.

Chuyển đổi tương tự - số thực hiện chức năng chuyển đổi thông tin dạng tương tự (thường là tín hiệu điện áp) sang dạng số (mã nhị phân).

Mạch chuyển đổi tương tự - số có cấu trúc như Hình 1.21.

Mạch ADC nhận tín hiệu tương tự dạng điện áp ở đầu vào và chuyển nó thành dạng số ở đầu ra. Độ rộng dữ liệu đầu ra có thể là 4, 8 bit, 12, 14 bit, Mạch ADC bao gồm bộ so sánh, logic điều khiển, thanh ghi điều khiển và mạch chuyển đổi số - tương tự DAC.

- Mạch ADC thực hiện hai thao tác cơ bản là: *lượng tử hoá* và *mã hoá*



Hình 1.21. Sơ đồ khối của ADC

- Lượng tử hoá là gán giá trị của tín hiệu tương tự - liên tục vào vùng các giá trị tương tự - rời rạc. Vùng giá trị này có nhiều mức, phụ thuộc chất lượng của ADC, mỗi mức tương tự - rời rạc cách nhau một khoảng lượng tử.

- Mã hoá là gán một mã nhị phân cho từng giá trị tương tự - rời rạc đó. Bộ so sánh thực hiện chức năng lượng tử hoá, gán giá trị tương tự tại thời điểm lấy mẫu vào vùng các giá trị tương tự - rời rạc bằng cách liên tục so sánh giá trị tương tự cần chuyển đổi với các giá trị tương tự được sinh ra trong quá trình chuyển đổi. Khi hai giá trị này xấp

xi nhau thì một tín hiệu được sinh ra báo hiệu quá trình lượng tử hoá đã xong.

Logic điều khiển cho phép khởi động và báo kết thúc quá trình chuyển đổi.

Thanh ghi điều khiển thực hiện chức năng mã hoá, tạo giá trị số trong quá trình chuyển đổi.

Quá trình chuyển đổi từ tín hiệu dạng điện áp tương tự sang dạng số được tiến hành trong chu kỳ lấy mẫu - chuyển đổi. Chu kỳ này được bắt đầu bởi tín hiệu bắt

$$X_a(t) = \sum_{n=-\infty}^{\infty} x\left(\frac{n}{2F_{\max}}\right) \left\{ \frac{\sin \pi(2F_{\max}t - n)}{\pi(2F_{\max}t - n)} \right\},$$

đầu chuyển đổi SOC (*Suất of Conversion*) và quá trình chuyển đổi báo hiệu kết thúc bằng tín hiệu EOC (*End of Conversion*),

Lấy mẫu là sự biến đổi tín hiệu analog liên tục theo thời gian xâu) thành các mẫu rời rạc $x_a(n.T_s)$ thứ tự theo thời gian với khoảng thời gian lấy mẫu cách đều nhau là T_s giây, được biểu diễn bằng: $\{f_n\} = \{f(nT)\}$

tại $t_n = nT$, $n = 0, \dots, \infty$,

Một chu kỳ lấy mẫu là T_s giây và khi đó $F_s = 1/T_s$ Hz là tần số lấy mẫu. Với mọi T_s nào đi chăng nữa thì tín hiệu nhận được sau khi lấy mẫu vẫn luôn luôn là tín hiệu rời rạc. Để khôi phục lại tín hiệu xoá từ các mẫu $x_a(n.T_s)$ thì nảy sinh ra điều kiện ràng buộc đối với cách lựa chọn giá trị T_s . Điều kiện này được phát biểu thành định lý lấy mẫu Shannon-someya, nó giới hạn quan hệ giữa thời gian lấy mẫu và những đặc tính của tín hiệu cần lấy mẫu. Khi chúng ta lấy mẫu 1 cách tự do sẽ xảy ra sự bóp méo với với tín hiệu lấy mẫu.

Một tín hiệu tương tự $x_a(t)$ có giải tần số giới hạn từ 0 đến $F_{\max}$ Hz, tín hiệu $x_a(t)$ sẽ được khôi phục lại bằng cách sử dụng các giá trị lấy mẫu $x_a(n.T_s)$ tại mọi khoảng thời gian $1/2F_{\max}$ giây như sau:

Trong đó $x(n/2F_{\max})$ là giá trị lấy mẫu của $x_a(t)$ tại $t = n/2F_{\max}$;

$n = -\infty, \dots, \dots, \infty$; $T_s = 1/2F_{\max}$ là chu kỳ lấy theo yêu cầu, và $F_s = 2F_{\max}$ là tần số lấy mẫu cho phép nhỏ nhất. Trong trường hợp sóng âm truyền qua đường dây điện thoại có giải tần số với $F_{\max} = 4\text{kHz}$ thì chu kỳ lấy mẫu sẽ là $T_s = 1/2 \cdot 4000$ giây (hay $T_s = 1/8\text{mili}$ giây),

Định lý Shannon Phát biểu như sau:

Một tín hiệu tương tự $x_a(t)$ có giải phổ hữu hạn với giới hạn trên là $F_{\max}(\text{Hz})$ (tức là phổ bằng 0 khi f nằm ngoài giải $-F_{\max} \dots F_{\max}$). Ta sẽ chỉ có thể khôi phục lại $x_a(t)$

một cách chính xác từ các mẫu $x_a(n.T_s)$ nếu như.

$$F_s > 2 F_{max} \text{ hay } T_s < 1/(2 F_{max})$$

Định lý Shannon cho chúng ta giới hạn nhỏ nhất của tần số lấy mẫu tín hiệu tương tự chuyển đổi thành tín hiệu số để sau đó khôi phục lại các tín hiệu gốc từ tín hiệu số mà không bị sai lệch.

1.5.2. Chuyển đổi AfD

Chỉ tiêu kỹ thuật chủ yếu của ADC:

- Độ phân giải của ADC biểu thị bằng số bit của tín hiệu số đầu ra. Số bộ càng nhiều thì sai số lượng tử càng nhỏ, độ chính xác càng cao.

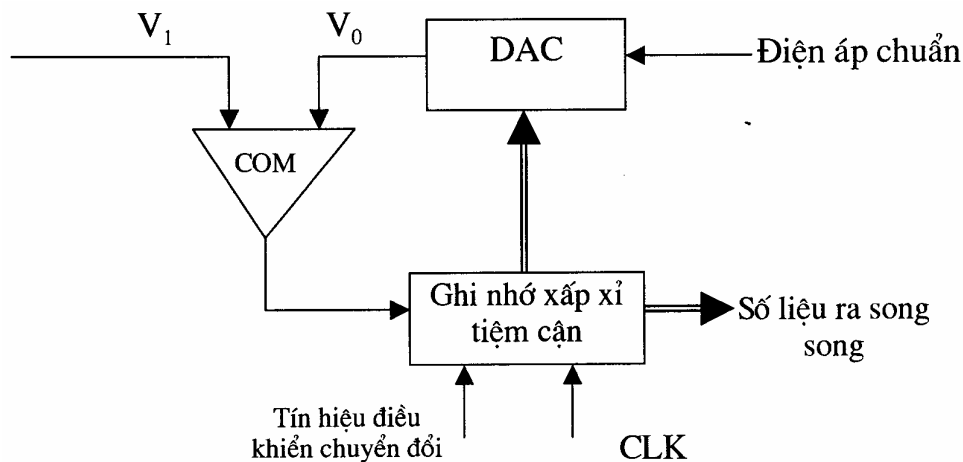
- Tốc độ chuyển đổi: Tốc độ biến đổi tín hiệu tương tự thành tín hiệu số. Chủ yếu phụ thuộc kiểu ADC,

- Các thông số khác: Mức logic của tín hiệu số đầu ra, khả năng chịu tải, Hệ số nhiệt độ, công suất tiêu hao...

a. A/D xấp xỉ tiệm cận

Đặc điểm: Tốc độ chuyển đổi nhanh - đến vài chục micro giây. Giá thành đắt từ vài chục (ADC0809 - 40.000) đến vài trăm nghìn (ADI220 200.000) - (1/2005). AD574 (650.000 - 1/2005)

Nguyên lý:



Hình 1.22. Nguyên lý làm việc của ADC xấp xỉ tiệm cận

COM: Bộ so sánh

DAC: Chuyển đổi số tương tự

CLK: Xung đồng hồ điều khiển

Trước khi thực hiện chuyển đổi, bộ nhớ phải xóa về 0. Bắt đầu chuyển đổi, xung

đồng hồ lập bit MSB trong bộ nhớ ở mức 1, Số liệu ra của bộ nhớ là 1000... 0. Tín hiệu số này được DAC chuyển đổi thành điện áp tương tự V_0 để đưa vào so sánh với điện áp vào cần chuyển đổi V_i .

Nếu $V_0 > V_i$ thì tín hiệu số quá lớn, trong bộ nhớ xấp xỉ tiệm cận bit MSB bị xoá về 0.

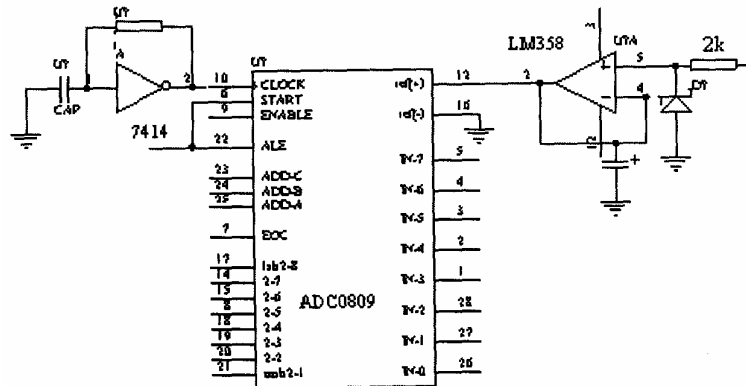
Nếu $V_0 < V_i$: bit MSB vẫn giữ nguyên = 1.

Tiếp theo cũng với phương pháp như trên xung đồng hồ lập bit có trọng số tiếp theo = 1. Sau khi so sánh logic mạch xác định giá trị 1 này có duy trì không hay chuyển thành 0. Cứ như vậy tiếp tục cho đến bit LSB thì xong.

Kết thúc quá trình giá trị số trong bộ nhớ chính là giá trị chuyển đổi cần mong muốn.

quá trình tương tự quá trình cân 1 vật bằng cân. Các quả cân được lựa chọn từ to đến bé. Mạch ADC xấp xỉ tiệm cận điển hình ADC0809: Quét động 8 kênh bằng logic địa chỉ,

- Độ phân giải 8 bit
- Tốc độ chuyển đổi $100\mu s$



Hình 1.23. Mạch điện ADC 0809

Bảng 1.4. Tổ hợp tín hiệu chọn đầu vào của ADC0809

C	B	A	Kênh chọn
0	0	0	IN0
0	0	1	IN1
0	1	0	IN2
0	1	1	IN3

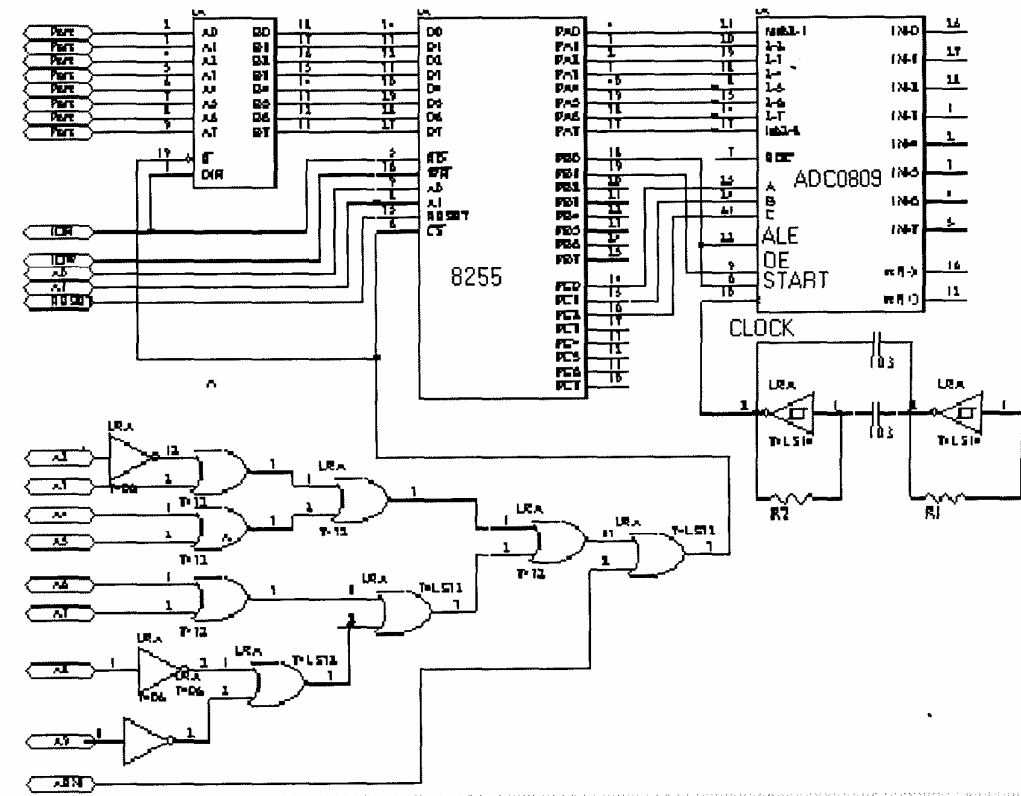
1	D	0	IN4
1	0	1	IN5
1	1	0	IN6
1	1	1	IN7

Nguyên tắc làm việc:

Điện áp chuẩn so sánh đưa vào chân số 12. Xung dao động đồng hồ đưa vào chân số 10. Tổ hợp trạng thái của các chân A, B, C: xác định đầu vào nào trong tám đầu IN0.. IN7. Khi có xung dương kích hoạt chân START (Nối với ALE) ADC bắt đầu làm việc. Trong thời gian chuyển đổi chân EOC đứng ở mức LOW sau khoảng 100 μ s (Nếu CLOCK = 640 KHz) sẽ chuyển sang HIGH báo hiệu kết thúc quá trình biến đổi. Khi chân ENABLE (pin 9) được đưa lên mức HIGH có thể đọc dữ liệu ra ở các chân dữ liệu.

Hình 1.24 là sơ đồ của một cảm ghép nối có sử dụng ADC0809 qua khe cắm ISA của máy tính PC.

Nếu cần loại ADC nhiều đầu vào hơn người ta có thể chọn ADC0816 hoặc 0817 có độ phân giải 8 bit với 16 đầu vào. Thời gian chuyển đổi một kênh của ADC là 100 μ s với clock 640kHz,



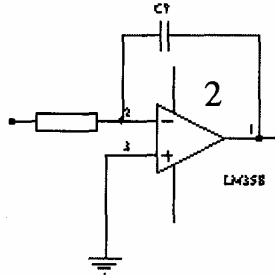
Hình 1.24. Ghép nối ADC0809 trên cam qua khe cắm ISA

b. A/D tích phân 2 sườn dốc

VD ICL7 109, ICL7 107, ICL7 135...

Đặc điểm: Độ chính xác cao, ổn định(10-14 bit). Tốc độ chuyển đổi chậm (vài trăm miligiây),

Nguyên lý: Nhắc lại mạch tích phân



Hình 1.25. Mạch tích phân

Từ điều kiện cân bằng tại nút 2 $I_r = I_c$ ta có:

$$-C \frac{dU_r}{dt} = \frac{U_v}{R} \quad \text{và} \quad U_r = 1/CR \int_0^t U_v dt + U_{r0}$$

U_{r0} là điện áp trên tụ C khi trôi là hằng số tích phân xác định từ điều kiện ban đầu). Thường khi $t=0$ $U_v=0$ và $U_r=0$ nên ta có:

$$U_r = 1/T \int_0^t U_v dt$$

Với $T = RC$ là hằng số tích phân của mạch. Khi tín hiệu vào thay đổi từng nấc, tốc độ thay đổi của điện áp ra sẽ bằng:

$$\frac{\Delta U_r}{\Delta t} = -\frac{U_v}{RC}$$

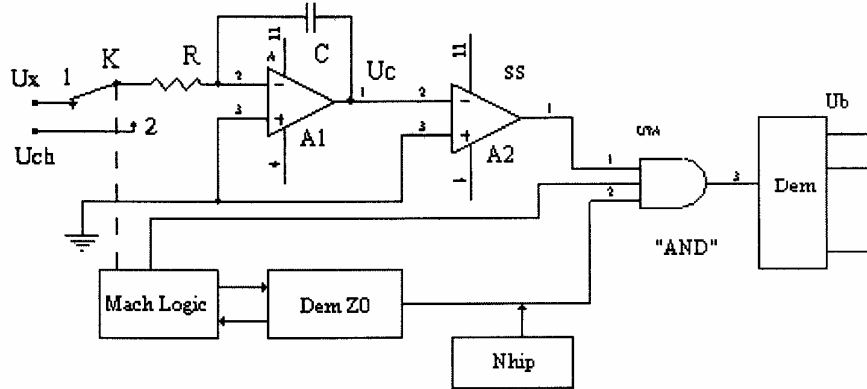
Nghĩa là ở đầu ra bộ tích phân sẽ có điện áp tăng (hay giảm tuyến tính theo thời gian). Đối với tín hiệu hình sin bộ tích phân sẽ là bộ lọc tần thấp, quay pha tín hiệu hình sin đi 90° và hệ số khuếch đại của nó tỉ lệ nghịch với tần số.

Theo Hình 1.26 khi mạch logic điều khiển khoá K ở vị trí 1 thì U_x (Điện áp tương tự cần chuyển đổi) nạp điện cho tụ điện C thông qua điện trở R. Trên đầu ra mạch tích phân A_1 có điện áp:

$$U' C = \frac{1}{RC} \int U_x dt = \frac{1}{RC} U_x t$$

Giả thiết thời gian nạp cho tụ là tỉ ta có điện áp hạ trên tụ sau thời gian tỉ là:

$$U'_{c1} = \frac{U_x}{RC} t_1 \quad (1)$$



Hình 1.26. Mạch nguyên lý của ADC tích phân hai sườn dốc

U_{C1} tỉ lệ với U_x Tùy theo U_x lớn hay bé, đặc tuyến $U'_c(t)$ có độ dốc khác nhau như trên hình đặc tuyến. Trong thời gian t_1 bộ đếm Z_0 cũng đếm các xung nhịp. Hết thời gian t_1 khoá K được mạch logic điều khiển sang vị trí 2 đồng thời tín hiệu từ mạch logic cũng được đưa đến mạch "AND" làm cho mạch này thông đối với xung nhịp. Tại thời điểm này mạch đếm ở đầu ra bắt đầu đếm, đồng thời mạch đếm Z_0 được mạch logic điều khiển về vị trí nghỉ. Khi K ở vị trí 2 điện áp chuẩn U_{ch} có giá trị ngược với U_x bắt đầu nạp điện cho tụ C theo chiều ngược lại, phương trình nạp là:

$$U'_c'' = -\frac{U_{ch}}{RC} t$$

Sau một khoảng thời gian t_2 thì

$$U''_{c2} = -\frac{U_{ch}}{RC} t_2 \quad (2)$$

Giả thiết rằng sau thời gian t_2 thì $U''_c = U'_c$ nghĩa là điện áp trên tụ U_c bằng 0, (1) và (2):

$$\frac{U_x}{RC} t_1 = \frac{U_{ch}}{RC} t_2 \quad \text{hoặc} \quad t_2 = \frac{U_x}{U_{ch}} t_1$$

$$Z_0 = t_1 f_n \quad \text{suy ra} \quad t_1 = \frac{Z_0}{f_n} \quad \text{thay vào công thức của } t_2 \text{ có} \quad t_2 = \frac{U_x}{U_{ch}} \frac{Z_0}{f_n}$$

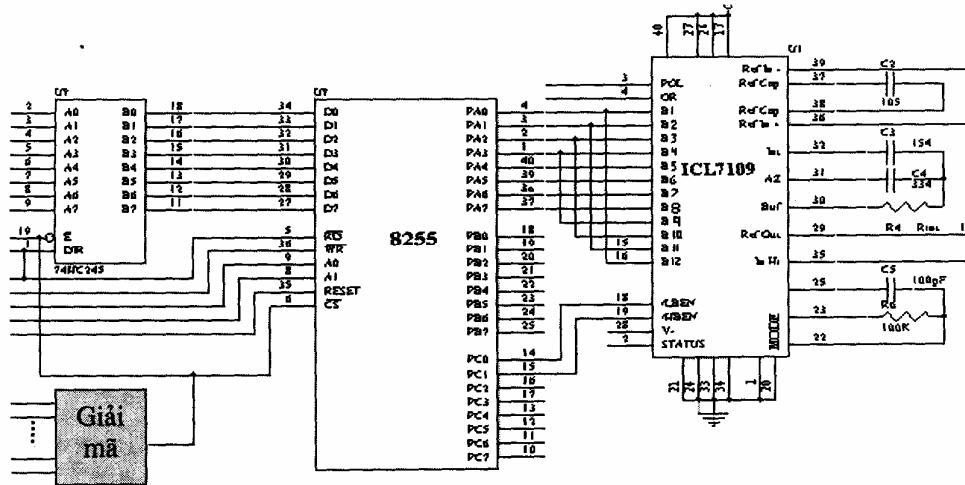
Do đó số xung nhịp đếm được nhờ mạch đếm ở đầu ra trong khoảng thời gian t_2 là: sau thời gian t_2 mạch đếm ra bị ngắt vì $U_c = 0$ và mạch logic đóng cổng "và":

$$Z = t_2 f_n = \frac{U_x}{U_{ch}} Z_0$$

Theo công thức này ta thấy số xung đếm đầu ra tỷ lệ với điện áp tương tự cần

chuyển đổi. Kết quả đếm không phụ thuộc vào thông số của mạch, không phụ thuộc vào tần số xung nhịp f_n . Theo phương pháp này ta đã làm cho điện áp U_x tỷ lệ với thời gian (t_1, t_2) rồi đem số xung nhịp xuất hiện trong khoảng thời gian đó.

ADC làm việc theo nguyên lý này có độ chính xác cao nhưng tốc độ chuyển đổi chậm.



Hình 1.27. Ghép nối ICL 7109 với 8255A

Giới thiệu ADC tích phân 2 sườn dốc ICL7 1 09

Đặc tính:

- Là loại ADC tích phân 2 sườn dốc 12 bits (*Polarity and Over ranger*).
- Có các đầu ra 3 trạng thái tương thích TTL và kiểu UART cho giao diện song song hoặc nối tiếp đơn giản.
- Đầu vào RUNfHOLD và đầu ra STATUS có thể sử dụng cho kiểm tra và điều khiển chuyển đổi.
- Dùng thạch anh 3,58MHz để làm dao động có thể thực hiện 7,5 chuyển đổi /giây. Có thể dùng các phần tử R và C lắp thành bộ dao động.
- Giới hạn nhiệt độ sử dụng từ -55°C tới $+125^{\circ}\text{C}$.
- Năng lượng tiêu thụ rất nhỏ (dòng điện tiêu thụ khi làm việc 3 mA).
- Điện áp cung cấp +5V, -5V.

ADC 7109 có thể làm việc ở 2 chế độ: Phụ thuộc điện áp đưa vào chân 21 (MODE):

- Khi pin 21 = low: Chế độ "Direct".
- Khi pin 21 = hgh: Chế độ sử dụng UART with Handshake.

Các công thức tính toán:

- OSILATOR FREQUENCY:

$$f_{OSC} = 0,45/RC$$

$$C_{osc} > 50pF; R_{osc} > 50k\Omega$$

$$f_{osc}(Typ) = 60kHz \text{ or } f_{osc}(Typ) = 3,58MHz$$

- OSILATOR PERIOD

$$t_{osc} = RC/0,45$$

$$t_{osc} = 1/113,58MHz \text{ Crystal}$$

- INTEGRATION CLOCK FREQUENCY

$$f_{clock} = f_{osc}(RC \text{ Mode})$$

$$f_{clock} = f_{osc}/58 \text{ (Crystal)}$$

$$t_{clock} = 1/FC_{LOCK}$$

- INTEGRATION PERIOD

$$T_{INT} = 2048 \times t_{CLOCK}$$

- 60/50 REJECTION CRITERION

$$t_{INT}/t_{60HZ} \text{ or } t_{INT}/t_{50HZ}: \text{Integer}$$

- OPTIMUM INTEGRATION CURRENT IINT:

$$i_{INT} = 20\mu A$$

- FULL SCALE ANALOG INPUT VOLTAGE

$$V_{INFS} \text{ Typically} = 200mV \text{ or } 2V$$

- INTEGRATE RESISTOR

$$R_{INT} = V_{INFS} / i_{INT}$$

- INTEGRATE CAPACITOR

$$C_{INT} = (t_{INT}) (i_{INT}) / V_{INT}$$

- INTEGRATOR OUTPUT VOLTAGE SWING

$$V_{INT} = (t_{INT}) (I_{INT}) / C_{INT}$$

- V_{INT} MAXIMUM SWING

$$(V_{-} + 0,5V) < V_{INT} < (V_{+} - 0,5V)$$

$$V_{INT}(TYP): 2V$$

- DISPLAY COUNT

$$COUNT = 2048 \times V_{IN} / V_{ref}$$

- CONVERSION CYCLE

$$t_{CYC}: t_{clock} \times 8192$$

(In Free Run Mode, Run/HOLD: 1)

$$\text{When } f_{clock} = 60\text{kHz } t_{CYC} = 133 \text{ ms}$$

- COMMON MODE INPUT VOLTAGE

$$(V_{-} + 2,0V) < V_{IN} < (V_{+} - 2,0V).$$

- AUTO-ZERO CAPACITOR

$$0,01\mu F < CAZ < 1\mu F$$

- REFERENCE CAPACITOR

$$0,1\mu F < C_{REF} < 1\mu F$$

- V_{REF}

Biased between V_{+} and V_{-}

$$V_{REF} - V_{+} - 2,8VV$$

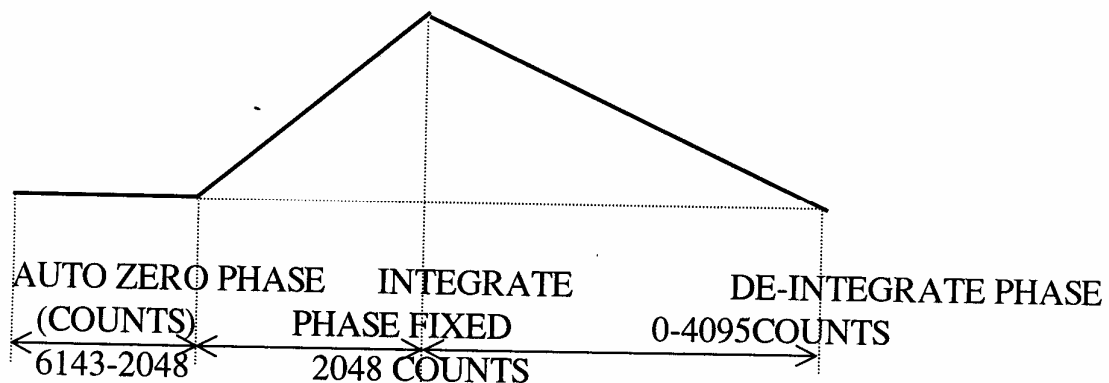
Regulation lost when V_{+} to $V_{-} \leq 6,4V$

V_{REF} is not used, float output pin

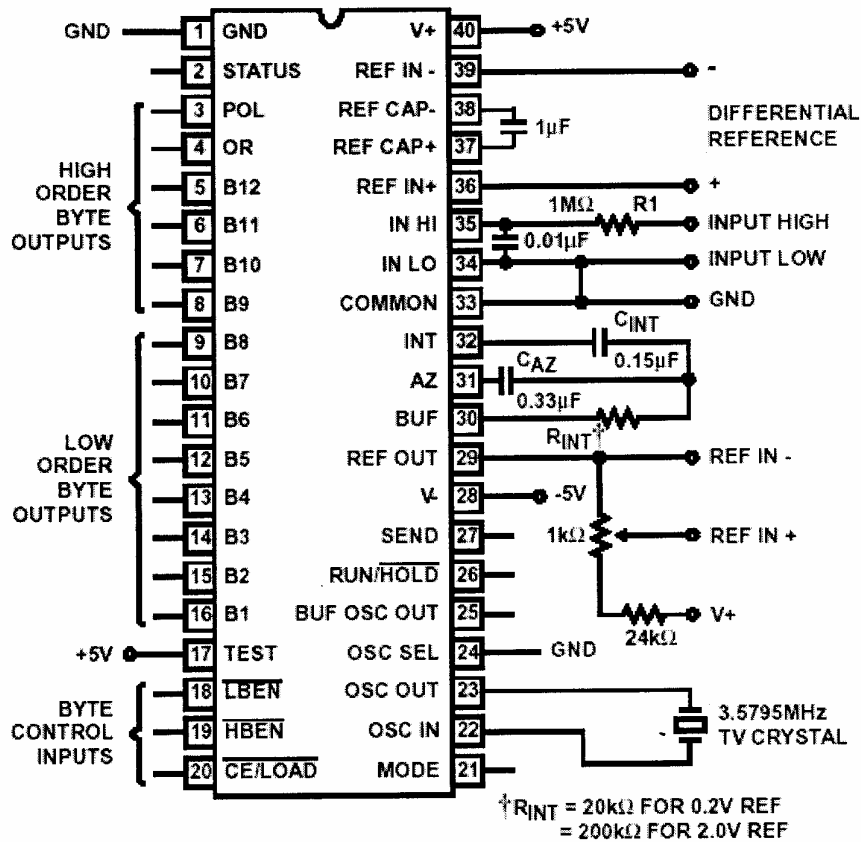
- POWER SUPPLY: DUAL $\pm 5,0V$

$V_{+} = +5V$ to GND

$V_{-} = -5V$ to GND



$$\text{TOTAL CONVERSION TIME} = 8192 \times t_{TLOCK} \text{ (IN FREE RUN - MODE)}$$

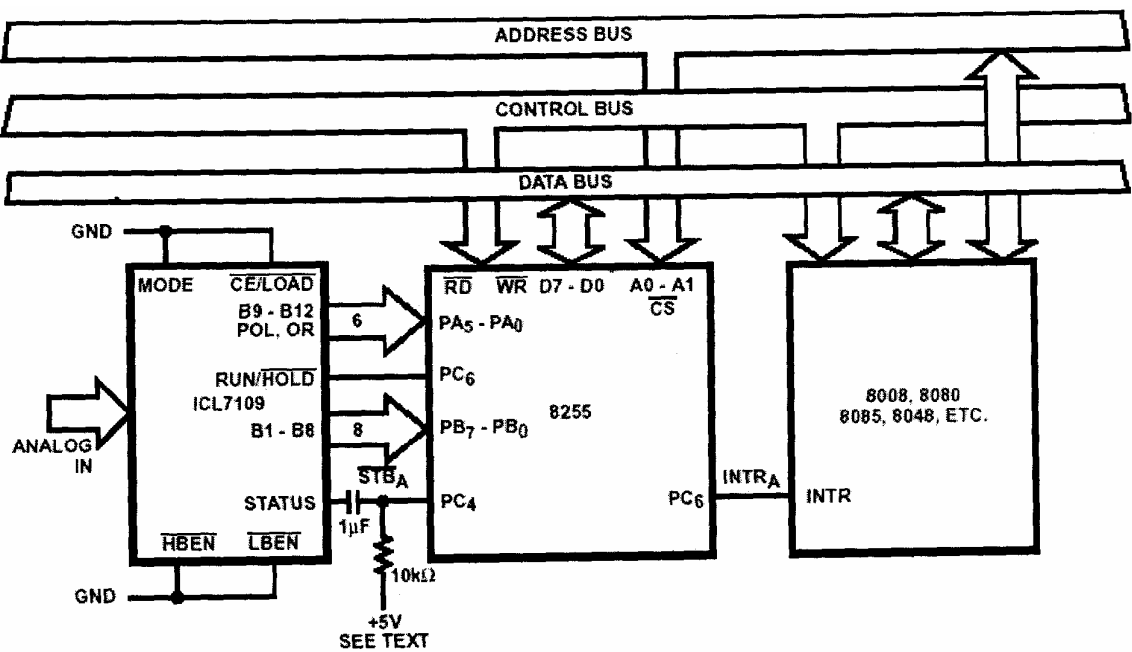


Hình 1.28 Mạch test ICL 7109

7109 được cấp nguồn dương (GND to V+) +6.0V và nguồn âm (GND to V-) -9V. Trong sử dụng thường áp dụng các mức điện áp V+ = +5V và V- = -5V. Dao động tạo xung nhịp cho 7109 đưa vào hai chân 22 và 23 có tần số 3.5795MHz. Sử dụng chân chân 21 "mode" để đặt chế độ làm việc trực tiếp (direct) hoặc chế độ bắt tay (handshake) giao tiếp với truyền tin nối tiếp. Điện áp analog đầu vào giữa chân 34 và 35. Tín hiệu digital 12 bit gồm 8 bit thấp từ B1 đến B8 (các chân 16-9) và 4 bit cao (các chân từ 8-5). Hai bit phân cực POL chân 3 và quá thang OR chân 4. Trong chế độ direct: Sau khi $\overline{CELOAD}$ tác động ở mức thấp quá trình chuyển đổi sau 330miligiây. Bằng cách cho các chân $\overline{BLEN}$ và $\overline{HBEN}$ tác động chúng ta sẽ đọc được 8 bit thấp và 4 bit cao của dữ liệu.

Do các cổng nối vào bus của các vi xử lý thế hệ cũ và các vi điều khiển thường chỉ là 8 bit nên có thể ghép qua hai cổng của vi mạch 8255 như Hình 1.29 hoặc ghép vào một cổng như Hình 1.27

Quá trình đọc dữ liệu dù theo sơ đồ nào cũng phải thực hiện hai lần, đọc vào hai biến sau đó được xử lý để thu được giá trị thực tế.

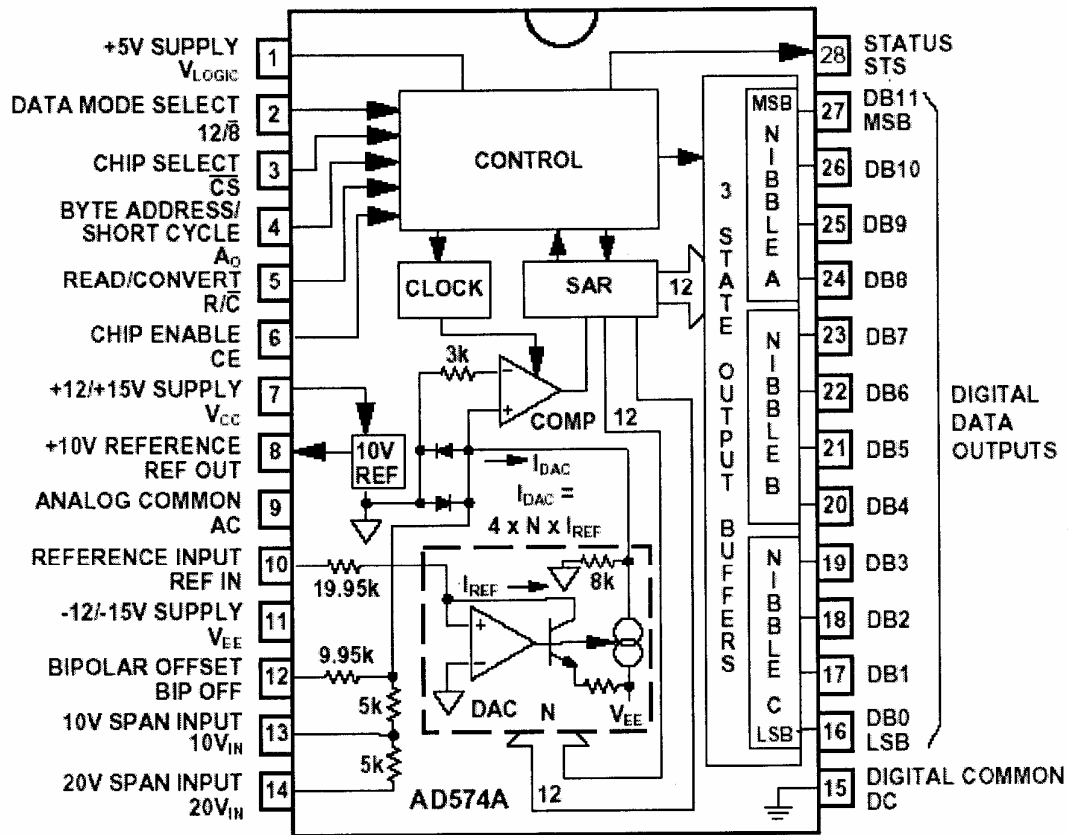


Hình 1.29. Ghép nối ICL7109 với 8255 và vi xử lý

• Giới thiệu ADC AD574

Thông số kỹ thuật của ADC 574:

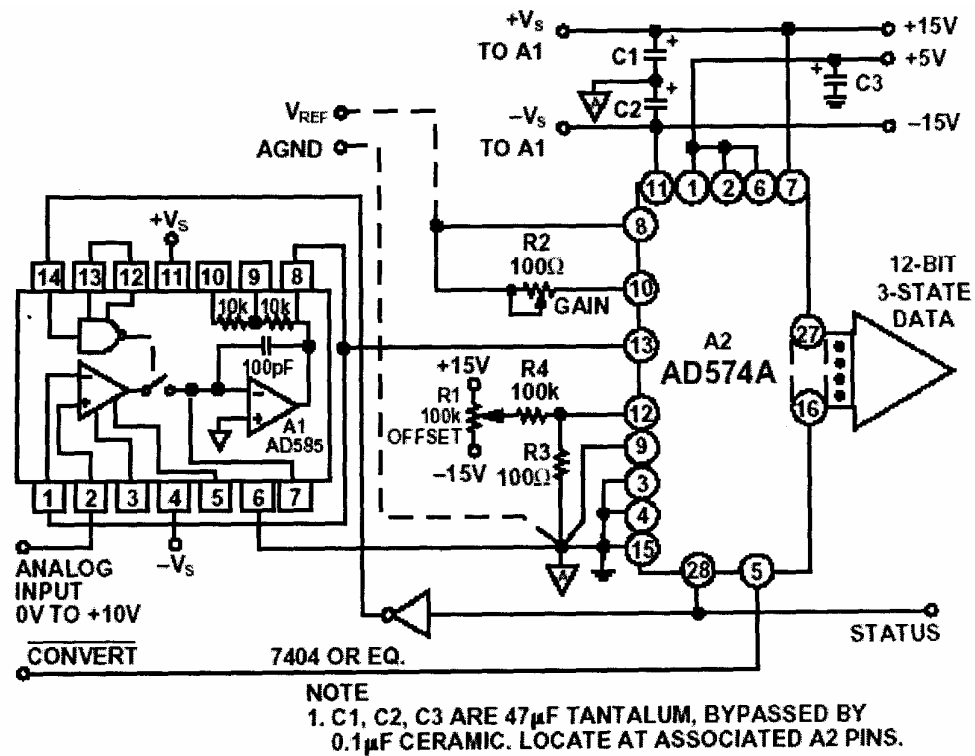
- + Độ phân giải 12 bit
- + Có giao diện bus 8 bộ và 16 bit với các bộ vi xử lý
- + Làm việc tuyến tính trong khoảng nhiệt độ từ 0°C đến $+70^{\circ}\text{C}$ (loại AD574A J, K, L và từ -55°C trên $+125^{\circ}\text{C}$ (loại AD574A s, T, U).
- + Thời gian chuyển đổi $25\mu\text{s}$
- + Tương thích (cả về chân) với ADC chuyển đổi tốc độ cao AD674B ($15\mu\text{s}$) AD774B ($10\mu\text{s}$)
- + Tiêu thụ năng lượng thttp: 390mW



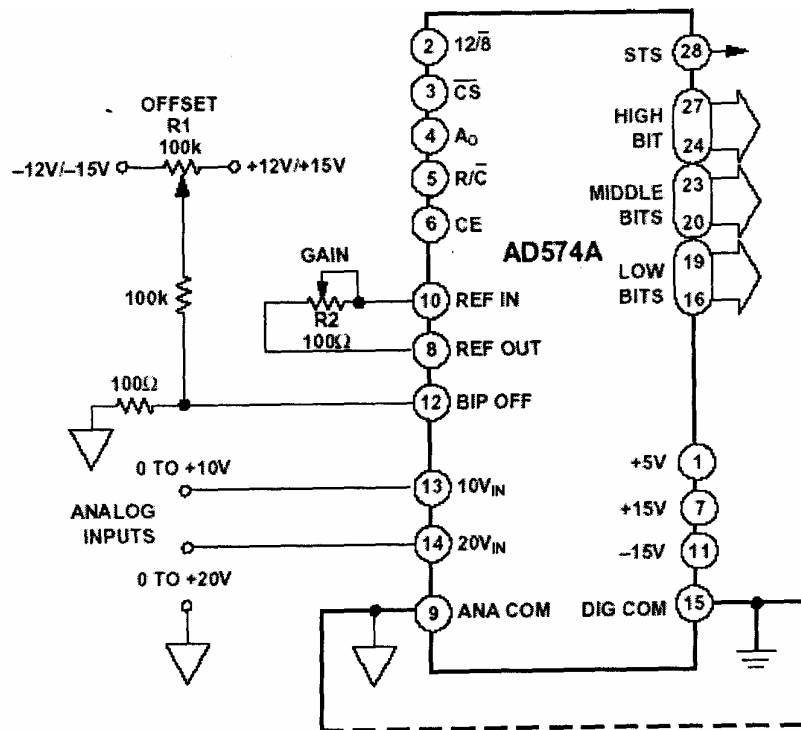
Hình 1.30. Cấu tạo AD574A

+ AD574A có một đầu vào tương tự và đầu ra số 12 bit, có bộ tạo chuẩn và bộ tạo xung nhịp sẵn bên trong. Đầu ra có bộ đệm 3 trạng thái để ghép trực tiếp vào kênh dữ liệu của bộ vi xử lý. AD574A được thiết kế với thành phần chính là hai chip LSI gồm cả hai mạch tương tự và số, điều này tạo cho vi mạch có đặc tính linh hoạt và hiệu suất cao với năng lượng thấp nhất. Chip thứ nhất là bộ chuyển đổi ADC 12 bit AD 565A hiệu suất cao và bộ tạo điện áp chuẩn. Nó còn có bộ chuyển mạch dòng đầu ra tốc độ cao và tập hợp điện trở màng mỏng được tinh chỉnh bằng laser. Chip thứ hai sử dụng mạch LSI gồm thanh ghi xấp xỉ liên tục, bộ điều khiển logic, bộ đệm ba trạng thái, bộ tạo xung nhịp và bộ hiệu chỉnh đầu vào, bộ so sánh có độ trôi thấp. Xem hình 2.5.

+ AD 574A có thể ghép nối với một trong các bus 8,12 hoặc 16 bit của hệ vi xử lý không cần bộ đệm cổng hoặc các phần tử điều khiển. Cả 12 bit dữ liệu đầu ra có thể đọc một lần hoặc hai lần mỗi lần 8 bit (lần đầu 8 bit dữ liệu lần sau 4 bit dữ liệu và 4 bit zero).



Hình 1.31. AD574A với AD585 Sample and Hold



Hình 1.32. Tín hiệu đầu vào đơn cực

Khởi tạo ADC theo bảng 1.5

Bảng 1.5. Bảng sự thật của ADC574A

CE	$\overline{CS}$	R/C	12/8	A ₀	Operation
0	X	X	X	X	None
X	1	X	X	X	None
1	0	0	X	0	Initiate 12-Bit Conversion
1	0	0	X	1	Initiate 8 Bit Conversion
1	0	1	Pin 1	X	Enable 12-Bit. Parallel Output
1	0	1	Pin 15	1	Enable 8 Most Significant Bit
1	0	1	Pin 15	0	Enable 4 BSBs + 4 Trailing Zeroes

Các tín hiệu CE, $\overline{CS}$ và R/C điều khiển chuyển đổi của bộ chuyển đổi. Trạng thái R/C khi cả CE và $\overline{CS}$ xác nhận bộ chuyển đổi đang đọc hay đang chuyển đổi. Khi đọc thì R/C = 1 còn khi chuyển đổi R/C = 0. Các đầu vào thanh ghi A₀ và 12/8 điều khiển dạng dữ liệu và độ dài chuyển đổi. Đường A₀ thường nối với bit LSB của bus địa chỉ. Nếu việc chuyển đổi bắt đầu với A₀ = 0 thì ở đầu ra sẽ nhận được đủ cả 12 bit. Nếu A₀ = 1 thì chuyển đổi 8 bit thấp. Trong quá trình thao tác đọc dữ liệu, giá trị A₀ còn xác định bộ đệm 3 trạng thái chứa 8 MSB (với A₀ = 0) hoặc 4 LSB (khi A₀ = 1). Chân 12/8 xác định dữ liệu đầu ra được xếp thành 2 word 8 bit (12/8 nối với chân DIG COM) hoặc một word đơn 12 bit (chân 12/8 nối với v_{logic}). Chân 12/8 không tương thích với TTL nên phải ghim về DIG hoặc v_{logic}.

Dạng thức 8 bit (khi A₀: 1) chứa 4 LSB giá trị biến đổi kèm theo 4 bit zero. Cách tổ chức này cho phép các đường dữ liệu ra hoàn toàn trùng hợp và giao diện trực tiếp với bus 8 bit mà không cần bộ đệm 3 trạng thái. Tín hiệu đầu ra STS chỉ thị trạng thái của bộ chuyển đổi. Khi STS = 1 là lúc bắt đầu chuyển đổi và STS = 0 khi chuyển đổi xong. Đặc tính thời gian chuyển đổi cơ bản của AD574A

Bảng 1.6. Thời gian chuyển đổi - các kiểu chuyển đổi

Symbol	Parameter	Min	Typ	Max	Units
t _{DSC}	STS Delay from CE			100	
t _{HEC}	CE Pulse Width	300			ns
t _{SSC}	$\overline{CS}$ to CE Setup	300			ns
t _{HSC}	$\overline{CS}$ Low During CE High	200			ns
t _{SRC}	R $\overline{C}$ to CE Setup	250			ns
t _{HRC}	R $\overline{C}$ Low During CE High	200			ns
t _{SAC}	A ₀ to CE Setup	0			Ns
t _{HAC}	A ₀ Valid During CE High	300			ns
t _C	Conversion Time				
	8-Bit Cycle	10		24	μs
	12-Bit Cycle	15		35	μs

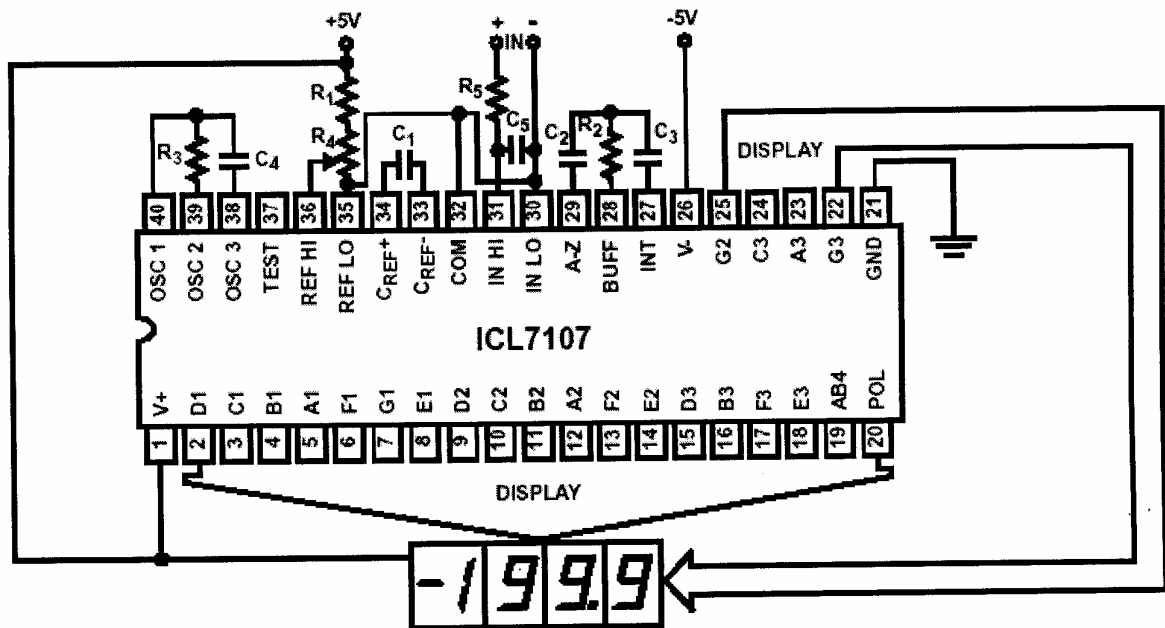
Khi thao tác chuyển đổi, chân $R/\overline{C}$ cần ở mức thấp trước lúc CE và $\overline{CS}$ xuất hiện. Nếu $R/\overline{C} = 1$ và thao tác đọc dữ liệu xuất hiện đồng thời có thể tạo thành sự tranh chấp trong bus hệ thống. Có thể dùng CE hoặc $\overline{CS}$ để khởi tạo một chuyển đổi. Nếu sử dụng $\overline{CS}$ để kích hoạt chuyển đổi ADC hoặc không có thời gian thiết lập riêng thì phải kéo dài thời gian hiệu lực của các tín hiệu cần thiết (thời gian đảm bảo ít nhất là 200ns trong khi $R/\overline{C}$, cE và $\overline{CS}$ có hiệu lực). Đầu CE có thời gian trễ nhỏ hơn chân $\overline{CS}$ nên điều khiển chuyển đổi nhanh hơn.

- Giới thiệu ADC 7106/7107

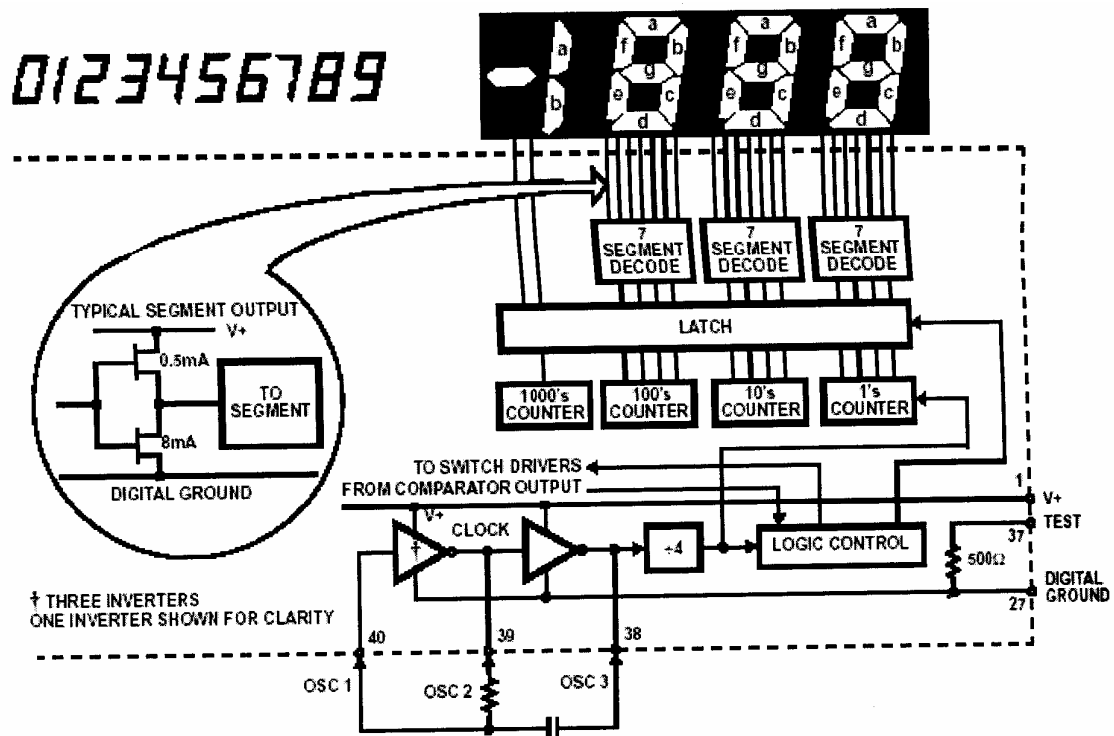
Là loại ADC tích phân 2 sườn dốc:

- + một đầu vào, độ phân giải $3^{1/2}$ digit, tiêu thụ năng lượng thấp
- + có thể ghép trực tiếp LCD và LED 7 đoạn. Rất thuận tiện cho các ứng dụng cần có hiển thị trong công nghiệp.
- + nhiệt độ làm việc 0°C đến 70°C .
- + Điện áp cung cấp: ICL7106, $V+$ to $V-$ 5V. ICL7107, $V+$ to GND 6V, $V-$ to GND 9V

$C_1 = 0.1\mu\text{F}$	$C_5 = 0.02\mu\text{F}$	
$C_2 = 0.47\mu\text{F}$	$R_1 = 24\text{k}\Omega$	
$C_3 = 0.22\mu\text{F}$	$R_2 = 47\text{k}\Omega$	$R_4 = 1\text{k}\Omega$
$C_4 = 100\text{pF}$	$R_3 = 100\text{k}\Omega$	$R_5 = 1\text{M}\Omega$



Hình 1.33. Mạch thử ICL 7107



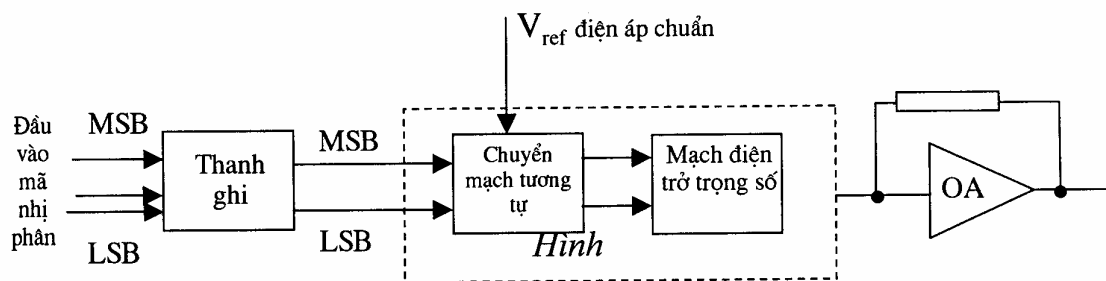
Hình 1.34. Phần hiển thị số của ICL7107

Chính xác cao, nhiễu thấp và giá thành thấp. Chúng được sử dụng rất tốt cho các băng tần đến 1MHz như tiếng nói, âm thanh.

1.5.3. Chuyển đổi D/A

Chuyển đổi số - tương tự DAC thực hiện chức năng chuyển đổi thông tin dạng mã số (mã nhị phân) sang dạng tương tự (dạng điện áp một chiều).

DAC có cấu trúc tổng quát như sau:



Hình 1.37. Chuyển đổi DAC

Đầu vào của DAC là dữ liệu ở dạng mã số (mã nhị phân). Độ rộng của dữ liệu vào (tính bằng bit) xác định độ chính xác của đầu ra tương tự.

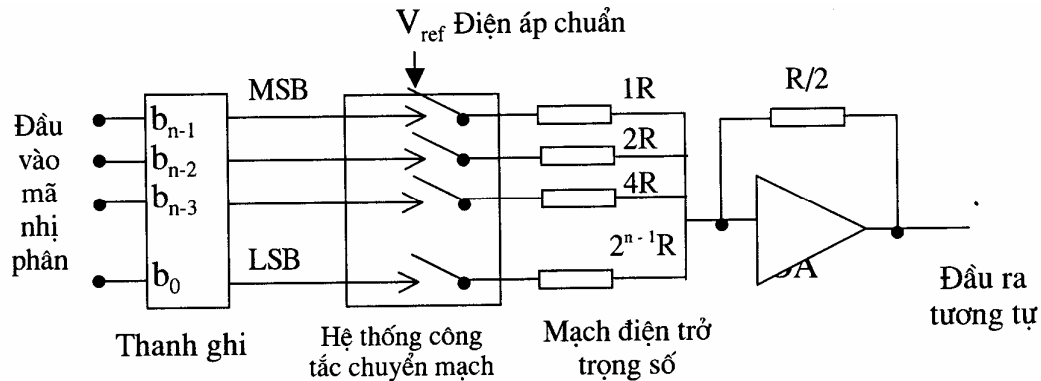
Độ rộng dữ liệu của đầu vào 8bit, 10 bit, 12 bit, 16 bit... tương ứng với khoảng tín hiệu tương tự có mức lượng tử từ 256 mức, 1024, 4096, 65536... mức.

Thanh ghi có chức năng chốt dữ liệu cần chuyển đổi,

Mạch xác định trọng số, bao gồm bộ chuyển mạch tương tự và mạch điện trở, cùng tín hiệu điện áp chuẩn V_{ref} sẽ tạo đầu ra điện áp hoặc dòng điện có mức tương ứng với giá trị của mã số ở đầu vào.

Mạch khuếch đại thuật toán (OA) có chức năng chuyển đổi dòng-áp cho đầu ra tương tự.

- DAC loại cộng



Hình 1.38. Nguyên lý DAC loại cộng

Mức logic ở đầu ra của thanh ghi quyết định việc kết nối điện áp chuẩn V_{ref} lên các điện trở trong mạch điện trở trọng số ở đầu vào của mạch khuếch đại thuật toán.

Mạch khuếch đại thuật toán đóng vai trò một bộ cộng

Mạch điện trở trọng số được tổ chức như sau:

$$1R = 2^0 R \text{ ứng với MSB - bit có ý nghĩa lớn nhất}$$

$$2R = 2^1 R$$

$$4R = 2^2 R$$

...

$$2^{n-1} R \text{ ứng với LSB - bit có ý nghĩa nhỏ nhất}$$

Điện trở phản hồi trên mạch khuếch đại thuật toán có trị số $1/2 R$

Điện áp ở đầu ra mạch OA là:

$$V_0 = - V_{ref} \frac{R/2}{R_{IN}}$$

trong đó:

$$\frac{1}{R_{IN}} = \frac{1}{2^0 R} b_{n-1} + \frac{1}{2^1 R} b_{n-2} + \frac{1}{2^2 R} b_{n-3} + \dots + \frac{1}{2^{n-1} R} b_0$$

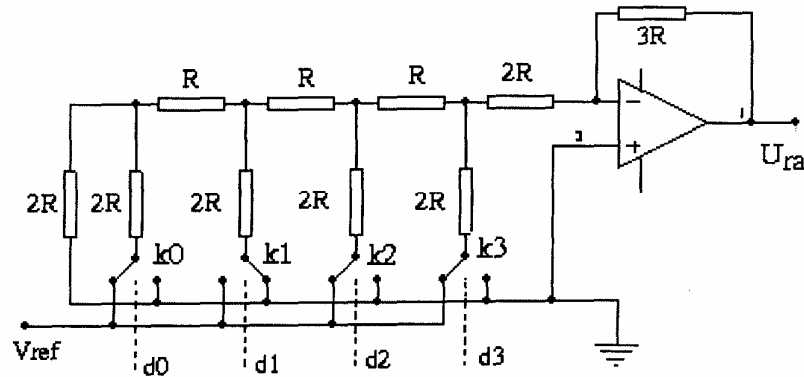
với $b_i = 0$ nếu bit tương ứng là "0"

$b_i = 1$ nếu bit tương ứng là "1"

Từ đây ta có điện áp đầu ra:

$$V_0 = -\frac{V_{ref}}{2} \left(\frac{1}{2^0} b_{n-1} + \frac{1}{2^1} b_{n-2} + \frac{1}{2^2} b_{n-3} + \dots + \frac{1}{2^{n-1}} b_0 \right)$$

DAC điện trở hình T



Hình 1.39 DAC điện trở hình T

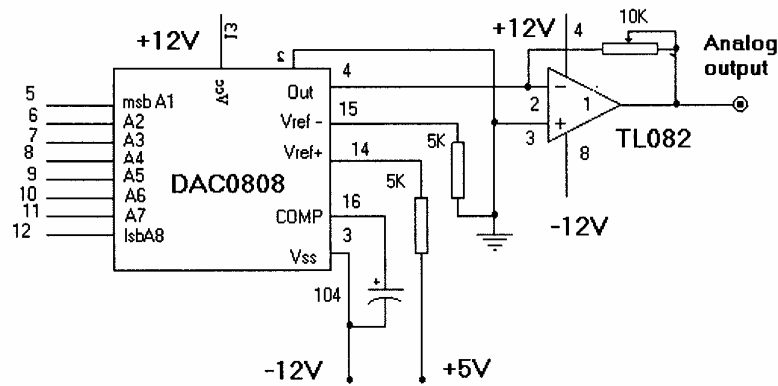
Mạch chuyển đổi DAC được tạo ra bằng cách mắc thang điện trở này với mạch khuếch đại thuật toán OA qua hệ thống chuyển mạch K_i . Với cách mắc này dòng điện I_i và các dòng nhánh qua các chuyển mạch K_i là cố định ($I_i/2, I_i/4, \dots$) và không phụ thuộc vào vị trí của chuyển mạch do cả hai vị trí của K_i đều được nối đất thật hoặc là qua lõi vào khuếch đại thuật toán OA (đất giả).

Các chuyển mạch K_i được điều khiển bằng các bit d_i . Khi $d_i = 1$ chuyển mạch được đóng sang bên phải; khi $d_i = 0$ chuyển mạch được đóng sang bên trái. Trên hình vẽ giá trị của các bit $d_3 d_2 d_1 d_0$ là 1101.

Với dữ liệu đầu vào có độ rộng n mạch thang điện trở-chuyển mạch- khuếch đại thuật toán cho điện áp đầu ra U_{ra} :

$$U_{ra} = -\frac{V_{ref}}{2^n} (d_{n-1} 2^{n-1} + d_{n-2} 2^{n-2} + \dots + d_1 2^1 + d_0 2^0)$$

- Sơ đồ sử dụng ADC 0808



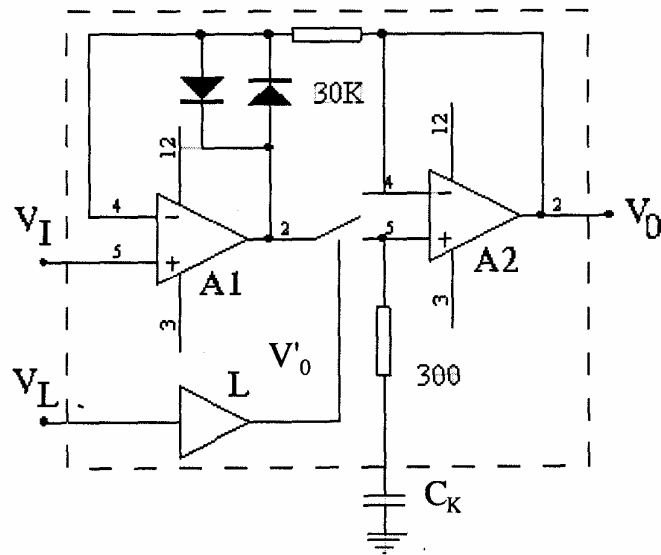
Hình 1.40 Sử dụng DAC 0808

Giá trị đầu vào số được đưa vào qua các chân từ số 5 đến chân số 12. Điện áp đầu ra Out tại chân số 4 được tính theo công thức:

$$V_0 = -V_{REF} \left(\frac{A1}{2} + \frac{A2}{4} + \dots + \frac{A8}{256} \right)$$

Mạch điện lấy mẫu và giữ mẫu: Khi sử dụng ADC để số hóa các tín hiệu tương tự việc thực hiện gồm lượng tử hóa và mã hóa. Trong pha đầu có quá trình lấy mẫu. Nếu sử dụng các ADC có tốc độ chuyển đổi thấp để thực hiện thì không đảm bảo điều kiện định lý shanon khi khôi phục tín hiệu. Để đảm bảo quá trình lấy mẫu đỡ bị mất mát thông tin người ta sử dụng các mạch lấy mẫu và giữ mẫu.

Trong hình này transistor trường kênh N là chuyển mạch lấy mẫu. Nếu \$V_L\$ ở mức cao thì T thông tín hiệu vào \$V_I\$ nạp cho tụ điện \$C_h\$ thông qua \$R_I\$ và T. Giả sử \$R_I = R_f\$ dòng điện đầu vào khuếch đại thuật toán bằng 0. Khi đó sau quá trình nạp điện thì \$V_0 = -V_I = V_C\$. Nếu \$V_L\$ về mức thấp rồi thì T ngắt. Do đó điện áp \$V_C\$ trên \$C_h\$ và \$V_0\$ duy trì một thời gian. Thời gian duy trì này càng dài nếu \$C_h\$ không rò và trở kháng vào bộ khuếch đại thuật toán càng lớn. Nhược điểm là tốc độ lấy mẫu thấp do \$C_I\$ nạp qua \$R_I\$ và T; mà \$R_I\$ không thể quá nhỏ do ảnh hưởng đến trở kháng vào của mạch khuếch đại thuật toán. Mạch cải tiến dùng trong vi mạch lấy mẫu và giữ mẫu LF 198 như sau:



Hình 1.41. Vi mạch lấy mẫu và giữ mẫu LF198

A1, A2 là các bộ khuếch đại thuật toán. S là chuyển mạch điện tử, L là mạch điều khiển S. Khi S nối A1 và A2 công tác ở chế độ bộ lặp (hệ số khuếch đại = 1 nên $V_0 = V'_0 = V_I$). Tụ điện bên ngoài C_K mắc nối tiếp với $R_2 (=300)$ thông xuống đất sẽ có điện áp bằng V_I . Khi $V_L = 0$, S ngắt điện áp trên tụ, do đó điện áp ra về duy trì. V_I có thể biến thiên đáng kể trước khi S được nối thông trở lại cho lần là mẫu kế tiếp do đó. V_0 có thể biến thiên rất lớn làm hỏng chuyển mạch S. D1 và D2 dùng để bảo vệ quá áp này. Nếu V'_0 quá chênh so với V_0 thì diốt sẽ thông giảm V'_0 trong phạm vi $V_I + V_D$ với và là điện áp rơi trên một. Còn khi S thông - lúc lấy mẫu thì $V'_0 = V_0$ D1 và D2 đều hở mạch, mạch bảo vệ không ảnh hưởng gì đến việc lấy mẫu.

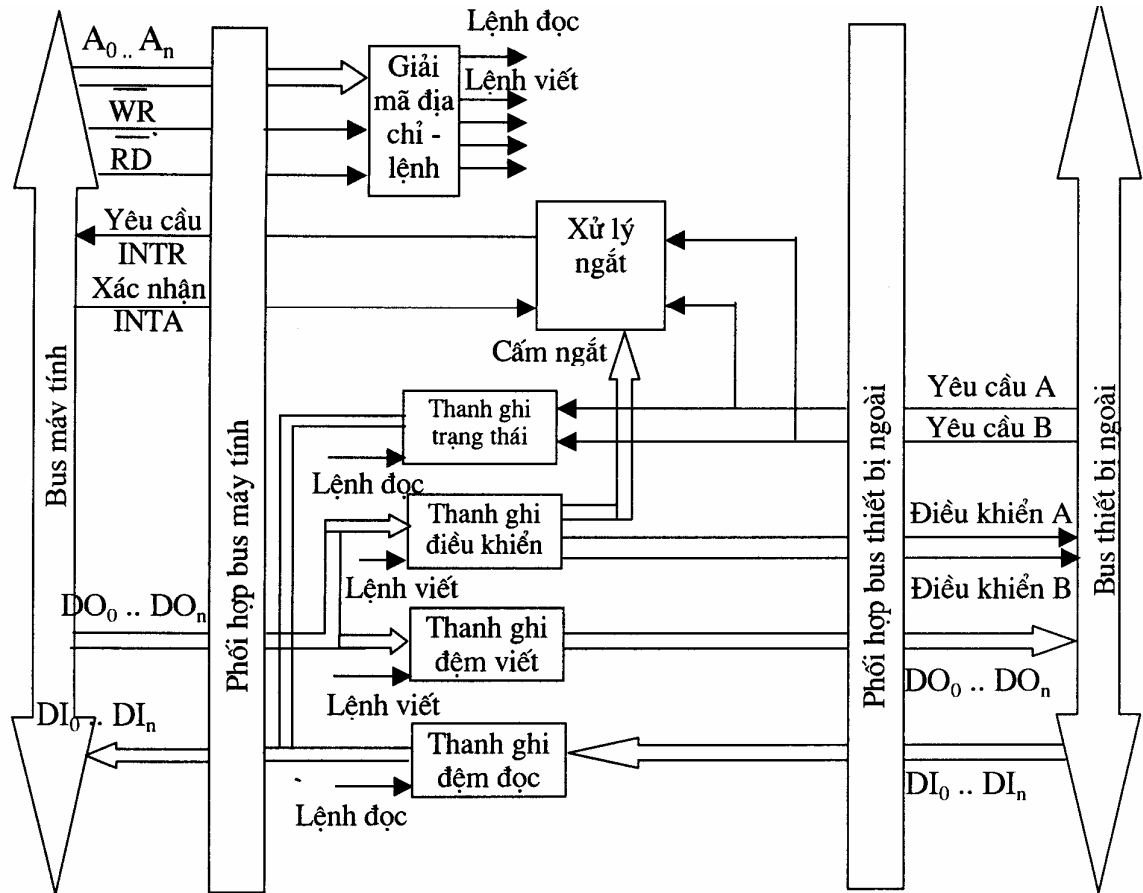
Sơ với sơ đồ trên thì điện trở R_1 đã được thay bằng mạch KĐTT A1 có điện trở nhỏ hơn nhiều nên khi nạp cho C_K sẽ nhanh hơn.

CÂU HỎI VÀ BÀI TẬP

1. Kiến trúc "hi-performance" trong máy tính PC giải quyết vấn đề gì?
2. Chipset là gì?
3. Hãy mô tả khái niệm *tần số bus* ở mức vật lý?
4. So sánh các đặc điểm kỹ thuật chính của mỗi loại ADC với nhau: tốc độ chuyển đổi, độ phân giải, khả năng giao tiếp với thiết bị khác?
5. Thiết kế mạch chuyển đổi dùng ADC thông dụng như ADC0809, ADC0804 có ghép với mạch lấy mẫu và giữ mẫu?
6. Thiết kế mạch chuyển đổi dùng ADC ICL7109, 7107 đo nhiệt độ dùng sensor LM335.

Chương II
Cấu trúc chung của 1 modul ghép nối

2.1. Mô hình tổng thể



Hình 2.1. Sơ đồ khối một khối ghép nối

Việc trao đổi thông tin, dữ liệu giữa thiết bị ngoài và hệ trung tâm (CPU+bộ nhớ) cơ bản là viết vào và đọc ra. Khi đưa ra thì dữ liệu được đưa qua các thanh ghi đệm viết. Khi đọc vào thì dữ liệu được đưa qua các thanh ghi đệm đọc. Các tín hiệu giải mã địa chỉ và điều khiển đọc viết được phối hợp giữa bộ giải mã và điều khiển logic cho phép thiết bị nào hoạt động.

Khi dùng phương pháp polling có thể kiểm tra điều kiện đưa dữ liệu vào hoặc viết dữ liệu ra thông qua các thanh ghi trạng thái.

Nếu vào ra dữ liệu dùng ngắt thì tín hiệu yêu cầu ngắt ngoài từ thiết bị ngoại vi, thông qua bộ xử lý ngắt tác động đến chân INTR của CPU. Nếu CPU chấp nhận ngắt nó phát tín hiệu INTA ra ngoài, ngắt được thực hiện, dữ liệu được viết vào / đọc ra bằng chương trình của chúng ta thay cho chương trình phục vụ ngắt ISR.

Có thể thực hiện kỹ thuật DMA để vào/ra dữ liệu được sự giám sát của DMAC (trong sơ đồ này không mô tả).

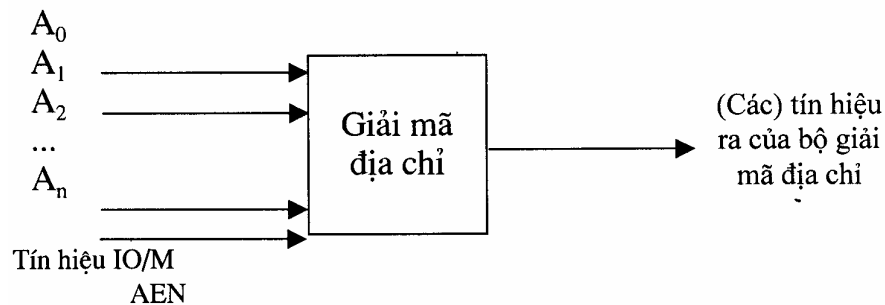
2.2. Cấu trúc các khối

2.2.1. Khối giải mã địa chỉ - nhiệm vụ, cấu tạo

- Mục đích của giải mã địa chỉ là để xác định được ô nhớ hay thiết bị ngoại vi mà CPU cần làm việc.

- Khối giải mã địa chỉ có nhiều đầu vào chân địa chỉ có thể kết hợp một số tín hiệu điều khiển và có một hoặc nhiều đầu ra giải mã địa chỉ.

- Tín hiệu đầu ra của bộ giải mã địa chỉ có thể là mức thấp (LOW) hoặc mức cao (HIGH)



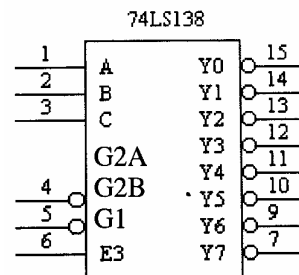
Hình 2.2 Bộ giải mã địa chỉ

- Các vi mạch thường được sử dụng để xây dựng bộ giải mã địa chỉ là các vi mạch AND, OR, NOT, NAND ghép nối với nhau theo yêu cầu cụ thể,

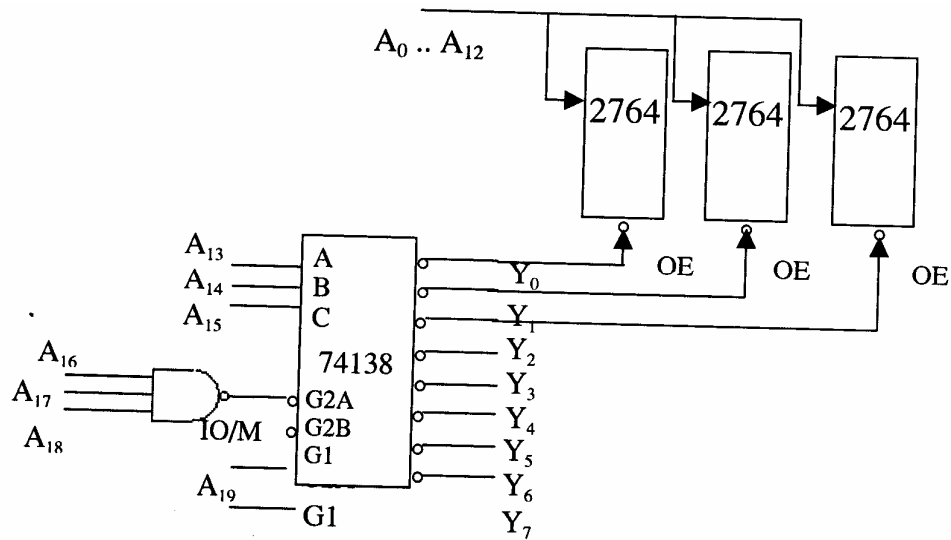
- Có thể sử dụng các mảng logic lập trình (GAL, PAL) để xây dựng các bộ giải mã địa chỉ.

- Có thể sử dụng vi mạch giải mã nhiều đầu ra chẳng hạn như 74LS138

Hoạt động của 74LS138



Hình 2.3. Vi mạch 74LS138



Hình 2.4. Dùng 74138 giải mã địa chỉ cho 3 vi mạch

Vi mạch có 8 đầu ra giải mã tác động ở mức thấp từ Y_0 đến Y_7 . Việc chọn đầu ra giải mã nào do tổ hợp tín hiệu các chân A, B, C. Để vi mạch hoạt động tín hiệu ở các chân G2A, G2B, G1 phải đảm bảo đồng thời như sau:

$G2A=0$, $G2B=0$ và $G1=1$

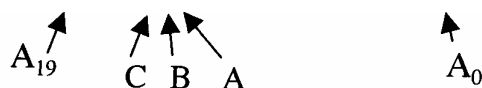
Bảng 2.1. Tác động của vi mạch

Input					Outputs							
Enable		Select										
G	G2*	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	H	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	H	H
H	L	H	H	H	H	H	H	H	H	H	H	L

* $G2 = G2A + G2B$

H = High Level, L = Low Level, X = Don't Care

Ví dụ sử dụng vi mạch 74138 giải mã cho 3 vi mạch ROM 2764, địa chỉ của ô nhớ đầu tiên là F0000h tức là 1111 0000 0000 0000 0000



Có nhiều phương án để giải mã cho một yêu cầu cụ thể khi sử dụng 74138.

2.2.2. Khối đệm dữ liệu - nhiệm vụ, cấu tạo

Là khối trung gian để chứa dữ liệu trước khi được chuyển vào hay chuyển ra khỏi hệ trung tâm.

- Các vi mạch hay được sử dụng:

+ Đệm 1 chiều: 74244

Vi mạch là bộ đệm gồm 2 nửa 4 bit. Nửa thứ nhất gồm các bit vào là 1A1,..., 1A4 và đầu ra là 1Y1,..., 1Y4. Nửa thứ hai là 2A1,..., 2A4 và các đầu ra là 2Y1,..., 2Y4.

Cho phép nửa thứ nhất làm việc khi chân điều khiển 1G (chân số 1) ở mức thấp

Cho phép nửa thứ hai làm việc khi chân điều khiển 2G (chân số 19) ở mức thấp

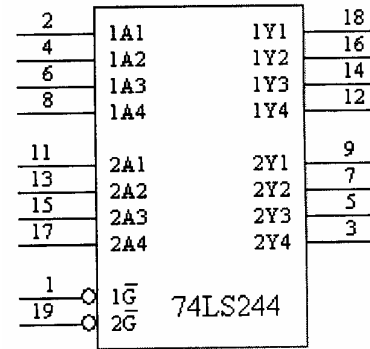
Như vậy bằng cách điều khiển các chân 1 và 19 có thể điều khiển dữ liệu qua hai nửa 4 bit riêng biệt hoặc đồng thời.

+ Đệm 2 chiều: 74245

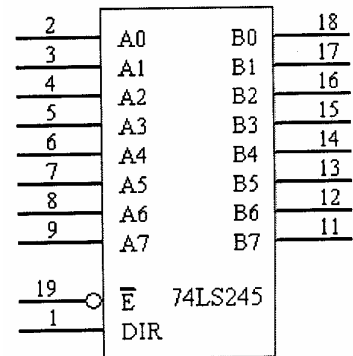
Đây là bộ đệm hai chiều 8 bit. Để dữ liệu qua được bộ đệm chân E (chân số 19) phải mức thấp. Chiều dữ liệu qua bộ đệm phụ thuộc vào mi logic của chân DIR (chân số 1). Khi DIR = 0 Các bit dữ liệu chỉ được phép từ B qua A. Khi DIR = 1 Các bit dữ liệu chỉ được phép từ A qua B.

+ Mạch chốt 74373

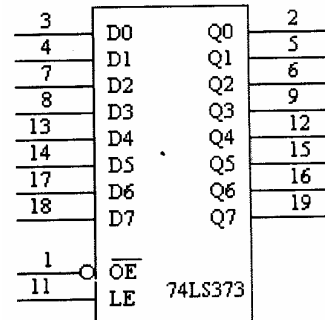
Vi mạch chốt có đầu ra 3 trạng thái (TRI-STATE). Để vi mạch làm việc được thì chân OE (chân số 1) phải ở mức thấp. Khi LE (chân số 11) phải ở mức thấp. Khi LE (chân số 11) phải ở mức cao dữ liệu tới các đầu D_i được đưa qua Q_i. Khi LE ở mức thấp dữ liệu được chốt giữ lại không đưa ra cho đến khi LE ở mức cao.



Hình 2.5 Vi mạch 74LS244



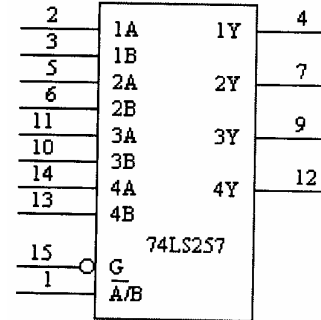
Hình 2.6. Vi mạch 74LS245



Hình 2.7. Vi mạch 74LS373

+ Vi mạch dồn kênh 74LS257

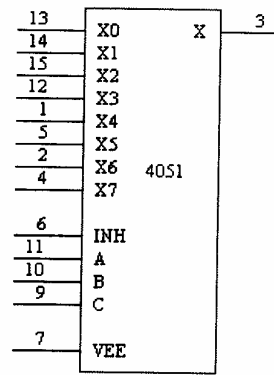
Tám bit vào (các chân 2,3,5,6,10,11,13,14) đến vi mạch, chỉ có 4 đường ra (1Y,2Y,3Y,4Y) được điều khiển qua làm 2 lần bằng giá trị chân A/B: Lần 1 chân này =0 các bit thấp (D0-D3) qua; lần 2 A/B =1 các bit cao (D4- D7) qua. Một ví dụ sử dụng vi mạch 74LS257 có thể xem trong trang 58.



Hình 2.8. Vi mạch 74LS257

+ Vi mạch chuyển mạch CD4051

Để vi mạch làm việc chân INH (chân số 6) ở mức thấp. Thay đổi tổ hợp A B C sẽ chọn được các đầu X_0 đến X_7 . Đầu vào có thể là X, đầu ra là một trong các đầu X_i (tách kênh) hoặc đầu vào là một trong các đầu X_i đầu ra là X (dồn kênh),

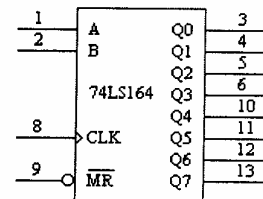


Hình 2.9. Vi mạch CD4051

+ Thanh ghi dịch 74LS164

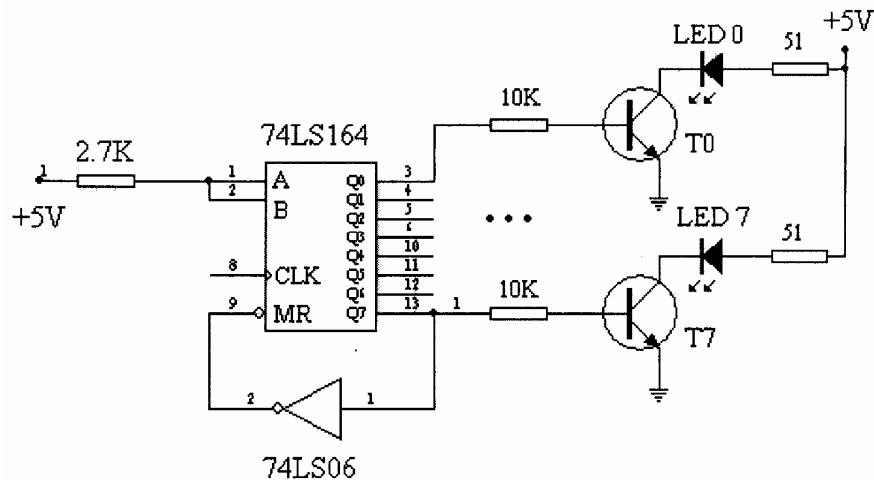
Bảng 2.2. Hoạt động của 74LS164

OPERATING	INPUTS			OUTPUTS	
	MR	A	B	Q_0	$Q_1 - Q_7$
Reset (Clear)	L	X	X	L	L - L
Shift	H	L	L	L	$q_0 - q_6$
	H	L	h	L	$q_0 - q_6$
	H	h	l	L	$q_0 - q_6$
	H	h	h	H	$q_0 - q_6$



Hình 2.10. Vi mạch 74LS164

Tín hiệu nối tiếp đưa vào A, B hoặc A và B. Các đầu ra song song Q_0 đến Q_7 . Xung nhịp làm việc được đưa vào CLK. Sau mỗi xung CLK trạng thái của Q_i lại được chuyển đến Q_{i+1} và đầu vào được chuyển đến Q_0 . Chân MR (chân số 9) là chân xóa các đầu ra Q_i . Khi MR ở mức thấp các đầu ra song song được xóa về 0.



Hình 2.11. Mạch điều khiển đèn sáng dần bằng 74LS164

Để hiểu sự làm việc của 74LS164 ta có thể xem sơ đồ hình.... Chân A và B nối với +5V qua điện trở 2.7K tương đương với đưa mức 1 vào A, B. Xung nhịp làm việc từ một bộ dao động khác đưa vào chân CLK. Khi bắt đầu làm việc các đầu Q_i (Q_0 đến Q_7) ở mức thấp các đèn T_0 đến T_7 khoá nên các LED không sáng.

- Khi có xung CLK đầu tiên mức 1 của A, B được đưa vào Q_0 . Đèn T_0 mở → LED₀ sáng. Trạng thái của Q_0 được chuyển sang Q_1 , Q_1 chuyển sang Q_2 ,... Các đèn T_1 .. T_7 vẫn khoá và các LED₁.. LED₇ vẫn không sáng.

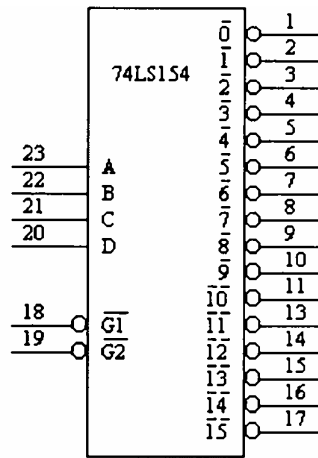
- Khi có xung CLK thứ hai → đèn T_0 mở trạng thái Q_0 được chuyển sang Q_1 → đèn T_1 mở... LED₀, LED₁ sáng...

-...

- Sau xung thứ 8 đèn T_7 mở, các đèn đều sáng. Trạng thái 1 của Q_7 được đưa về MR làm cho các đầu Q_i trở lại bằng 0 và lại bắt đầu một chu kỳ mới.

+ Vi mạch giải mã 74LS154

Vi mạch hoạt động khi G1 (chân số 18) và G2 (chân số 19) ở mức thấp. Tùy theo tổ hợp ABCD nào mà các đầu ra (từ chân 1 đến chân 17) sẽ được kích hoạt ở mức thấp.



Hình 2.12 Vi mạch 74154

Giới thiệu mạch bảng chữ điện tử

Quét hàng dùng 74154

Quét cột dùng 74164

2.2.3. Khối logic điều khiển thiết bị - nhiệm vụ, cấu tạo

- Đảm bảo sự phối hợp hoạt động của các vi mạch trong thiết bị 1 cách đúng đắn để việc đưa dữ liệu vào/ra giữa hệ trung tâm và thiết bị ngoại vi thực hiện được.

- Các tín hiệu điều khiển của khối này gồm:

- + $\overline{WR}$: Điều khiển đọc
- + $\overline{RD}$: Điều khiển viết
- + ALE: Xung cho phép chốt địa chỉ
- + $IO/\overline{M}$: Phân biệt truy nhập bộ nhớ hay thiết bị ngoại vi
- + RESET: Tín hiệu khởi động lại CPU
- + AEN: Phân biệt kiểu vào ra dữ liệu AEN=1 Thực hiện DMA, AEN =0 vào ra qua CPU.

Các ghép nối trong khối cần đảm bảo sao cho:

+ Khi tín hiệu giải mã địa chỉ đưa vào khối giải mã địa chỉ đúng thì các vi mạch đệm, cổng phải mở ra

+ Các tín hiệu điều khiển đọc/ viết của CPU phải nối vào các chân điều khiển đọc/viết của các vi mạch. RD đảm bảo chiều dữ liệu vào. WR đảm bảo chiều dữ liệu ra. Tín hiệu reset của CPU phải được nối vào các chân reset của các vi mạch cổng.

+ Dữ liệu thông qua vi mạch đệm nối vào bus dữ liệu của hệ thống.

2.3. Phần mềm điều khiển thiết bị

- Dùng để điều khiển việc vào ra dữ liệu, được viết bằng các ngôn ngữ lập trình (C, C++, Pascal, delphi, VB, assembly,...) truy nhập qua các cổng điều khiển thiết bị.

- Chương trình điều khiển thiết bị phải được viết tương ứng với phần cứng. Nếu phần mềm điều khiển của thiết bị này dùng cho thiết bị khác thì sẽ không thực hiện được việc điều khiển. Có hai loại phần mềm điều khiển:

- + Viết theo yêu cầu riêng, cụ thể với một phần cứng không thông dụng (Giống như "may đo"). Các phần mềm này không tích hợp được vào hệ điều hành.

- + Các driver là các chương trình điều khiển thiết bị được viết theo các qui chuẩn để có thể tích hợp vào các hệ điều hành. Các phần mềm điều khiển thiết bị, cụ thể được viết để chạy trên nền hệ điều hành nào đó đảm bảo việc vào ra dữ liệu cho 1 thiết bị phần cứng cụ thể. Các nhà cung cấp thiết bị rời như cam màn hình, cam âm thanh, các Ổ USB, modem, máy in, máy quét... đã viết sẵn các driver cho mỗi loại thiết bị. Họ có thể bắt tay với nhà cung cấp hệ điều hành - như Microsoft chẳng hạn để thống nhất mã sản phẩm của mình VID (Vendor ID). Khi ghép thiết bị vào hệ thống, chương trình của hệ điều hành sẽ thông qua các mã này để phát hiện ra thiết bị nào của hãng nào được ghép vào hệ thống để khởi động driver tương ứng đã cung cấp cho nhà sản xuất hệ điều hành. Điều này giải thích việc các hệ điều hành của Microsoft gần đây như Windows 2000, XP tự động nhận biết được hầu hết các thiết bị ngoại vi thông dụng.

- Một số giao diện không cho phép lập trình trực tiếp mà phải thông qua các hàm API như các giao diện PCI, USB,

- Một số hệ điều hành không cho phép lập trình trực tiếp vào các địa chỉ cổng mà phải thông qua các hàm API như Windows XP.

Các hệ điều hành mới như windows NT, 2000, XP không cho phép đọc viết trực tiếp các cổng. Để truy nhập cổng cần thông qua các hàm API hoặc qua các thư viện của từng ngôn ngữ lập trình trên windows. Trong chương tiếp theo có giới thiệu chi tiết sử dụng thư viện inoutport. để lập trình với VB truy nhập cổng song song của máy tính PC.

CÂU HỎI VÀ BÀI TẬP

1. Hãy liệt kê các vi mạch đệm (1 chiều, 2 chiều), cách sử dụng mà bạn biết?
2. Một vi mạch đệm và một vi mạch chốt khác nhau cơ bản ở điểm nào?
3. Hãy liệt kê các vi mạch dồn/tách kênh, cách sử dụng?
4. Hãy liệt kê các vi mạch giải mã, cách sử dụng?
5. Hãy liệt kê các vi mạch thanh ghi dịch, cách sử dụng?

6. Dữ liệu từ trong bộ nhớ máy tính ra ngoài thiết bị ngoại vi và ngược lại có phải là liên tục không hay gián đoạn? Vai trò của khối logic với vấn đề này?

7. Tại sao hệ điều hành tại nhận diện được thiết bị ngoại vi của các hãng khác nhau? Hãy cho biết mã VID một số thiết bị trên máy tính của bạn: ổ cứng, bàn phím, cam mạng,...?

8. Xây dựng mạch quét sử dụng ROM và 74LS154 để điều khiển dãy đèn

9. Tìm hiểu sử dụng Inpout32.dll?

Chương III
Thiết kế các ghép nối máy tính qua các giao diện

3.1. Ghép nối máy tính qua các giao diện

3.1.1. Ghép nối qua cổng song song

a) Các thanh ghi của giao diện vào ra song song

Đặc điểm:

- Cùng 1 thời điểm truyền được 4, 8, 16, 32 bộ dữ liệu (Mỗi bit dữ liệu có 1 đường dẫn dữ liệu riêng),

- Gửi nhanh, nhận nhanh,

- Khoảng cách trao đổi thông tin ngắn do ảnh hưởng của điện dung ký sinh, do mức điện áp, do kinh tế.

Điện hình trong máy tính đó là cổng song song (Cổng LPT - Line PrinTer port).

Hiện nay trong máy tính PC sử dụng giao diện song song chuẩn Centronics do hãng Centronics thiết kế. Có 4 cổng song song với các địa chỉ

3BCh, 378h, 278h, 2BCh

Khi máy tính mới bật lên, BIOS (Basic Input Output System) sẽ quyết định số hiệu địa chỉ cổng máy in song song. Đầu tiên nó sẽ tìm ở địa chỉ 3BCh nếu tìm thấy cổng song song ở đây thì cổng LPT1 sẽ là 3BCh. Sau đó BIOS tìm tiếp cổng song song ở địa chỉ 378h; nếu có đây sẽ là LPT2... Nhưng ở bước đầu nếu không thấy cổng song song ở 3BCh thì 378h sẽ là địa chỉ của cổng LPT1. Trên mainboard máy tính PC hiện nay do vấn đề giảm giá thành và do xuất hiện các giao diện vào/ra mới có nhiều tính năng ưu việt nên các nhà sản xuất chỉ chế tạo một cổng máy in song ghép với địa chỉ 378h.

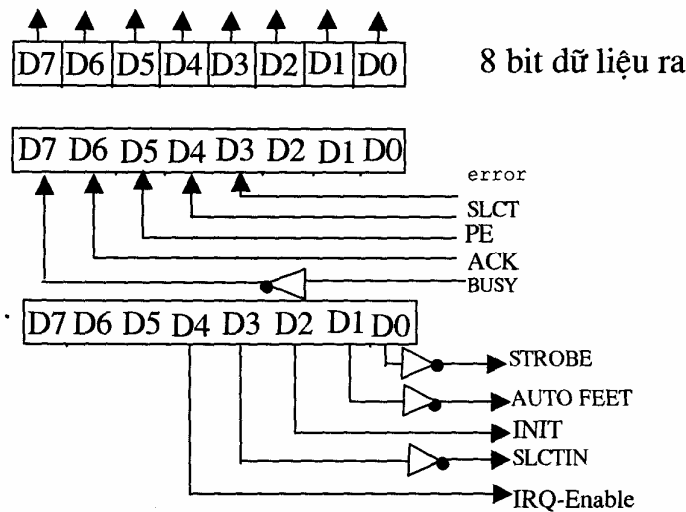
Các thanh ghi: Cổng máy in song song có 3 thanh ghi:

- Thanh ghi địa chỉ cơ sở: 378h, 278h, 3BCh, 2BCh là thanh ghi dữ liệu (Data Register). Dữ liệu từ trong máy tính được đưa ra qua thanh ghi này. Có 1 số hãng cho phép có thể đưa dữ liệu vào máy tính qua thanh ghi này. Đó là cổng song song kiểu vào ra hai chiều (*bi-directional mode*).

- Thanh ghi Địa chỉ cơ sở + 1: 379h, 279h, 3BDh, 2BDh có 5 bit vào máy tính Có thể sử dụng các bit vào này để kiểm tra trạng thái của các thiết bị ngoại vi hoặc đưa dữ liệu vào máy tính.

- Thanh ghi địa chỉ cơ sở + 2: Thanh ghi điều khiển địa chỉ 37Ah, 27Ah 3BCh, 2

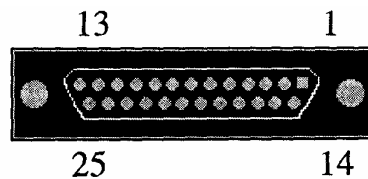
BCh. Gồm 5 bit ra,



Hình 3.1. Các thanh ghi của cổng máy in song song

Giao diện song song trên các máy vi tính hiện nay thường được ghép với máy in ở địa chỉ 378h. Giao diện song song còn cho phép khả năng truyền nhận dữ liệu ra vào máy tính trong các ứng dụng đo lường và điều khiển bằng máy tính.

Hình dáng bên ngoài của đầu nối cổng song song loại 25 chân trên hình vẽ - Nhìn vào máy tính, chân hình vuông là số 1.



Hình 3.2. Đầu nối cổng máy in 25 chân trên máy tính

Các chân như sau. (Thứ tự từ 1 - 25)

STROBE:	Lối ra (Output)	(1)
D ₀	Lối ra dữ liệu D ₀	(2)
D ₁	Lối ra dữ liệu D ₁	(3)
D ₂	Lối ra dữ liệu D ₂	(4)
D ₃	Lối ra dữ liệu D ₃	(5)
D ₄	Lối ra dữ liệu D ₄	(6)
D ₅	Lối ra dữ liệu D ₅	(7)
D ₆	Lối ra dữ liệu D ₆	(8)
D ₇	Lối ra dữ liệu D ₇	(9)

ACK Lỗi vào Input (10)

BUSY Lỗi vào Máy in bận (11)

PE Lỗi vào Hết giấy (12)

SLCT Lỗi vào Select (Lựa chọn) (13)

AF Lỗi ra Autofeed (Tự chọn) (14)

ERROR Lỗi vào error (Lỗi) (15)

INIT Lỗi ra Đặt lại máy in (16)

SLCTIN Lỗi ra Select in (17)

Các chân từ 18 đến 25

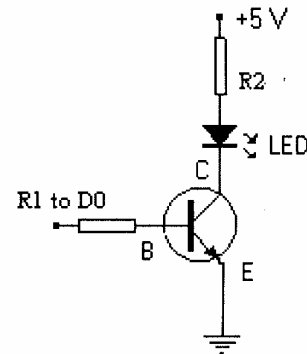
GND Nối đất

Nếu bạn quan sát kỹ có thể thấy số hiệu các chân được in nổi trên đầu cắm nhựa.

b) Điều khiển ra ngoài qua cổng song song

- Đặc điểm: Công suất ra của các đường tín hiệu không lớn, nếu cần điều khiển công suất lớn cần ghép thêm các mạch phụ.

- VD: Hãy khảo sát một mạch điện điều khiển qua cổng máy in song song đơn giản nhất như sau. Đây là mạch điện khóa dùng tranzistor. Bình thường tranzistor ở chế độ khóa, không có dòng collector (giống như công tắc ở vị trí ngắt điện), không có dòng chạy qua R2, qua LED nên đèn LED tắt. Khi ta đưa mức điện áp "1" vào R1 tranzistor dẫn điện (giống như công tắc ở vị trí dẫn điện), dòng chạy qua R2, qua LED, sẽ làm đèn LED sáng. Muốn đèn tắt ta đưa tín hiệu mức "0" tới R1.



Hình 3.3. Mạch điện đóng/mở LED

- Chương trình điều khiển LED có thể viết bằng Pascal

Muốn đưa 1 giá trị ra cổng ta dùng

port[\$địa chỉ cổng]:= giá trị;

Muốn đọc giá trị của cổng vào 1 biến ta dùng

biến:=port[\$địa chỉ cổng];

- Nếu sử dụng ngôn ngữ C:

Để đưa 1 byte ra cổng có thể dùng

outportb(Địa chỉ cổng, giá trị);

VD: outportb(0x378,0x80)

hoặc để đưa 1 word ra cổng dùng

outport(Địa chỉ cổng, giá trị);

Muốn đọc cổng có thể dùng

biến = inportb(Địa chỉ cổng);

hoặc ***biến = inport(Địa chỉ cổng)***

VD: val = inportb(0x3F8);

- Nếu dùng hợp ngữ, để đưa 1 giá trị ra 1 cổng:

mov dx, cổng

mov ax, giá trị

out dx, al

Để đọc cổng

mov dx, cổng

in al, dx

mov biến, al

Program controlLED;

uses crt;

const n=500;

begin

clrscr;

Write (' Chương trình điều khiển đèn LED ');

repeat

port [\$378]:= port [\$378] or \$01; {chân điều khiển nối với D0}

delay (n);

port [\$378]:=\$00;

delay (n);

until keypressed;

port [\$378]:=\$00

end.

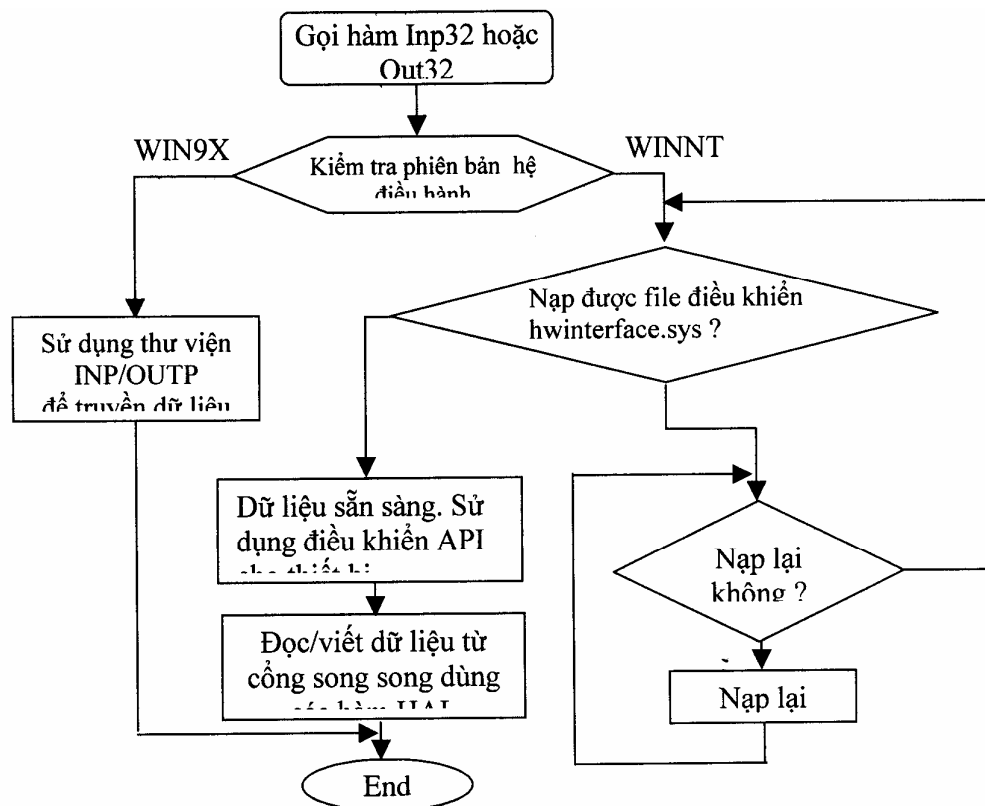
Lập trình trên Windows:

Nếu viết các chương trình điều khiển bằng Pascal, C trên DOS hoặc hợp ngữ như trên thì chương trình điều khiển của chúng ta chỉ chạy được trên Windows 9X, còn trên Windows NT, 2000, XP thì không chạy được vì các hệ điều hành này không cho phép đọc viết trực tiếp các cổng. Có thể viết chương trình thực hiện truy xuất dữ liệu qua cổng song song chạy trên các hệ điều hành windows bằng cách sử dụng thư viện liên kết động Inpout32.dll của tác giả Jan Axelson (<http://www.lvr.com>)

Thư viện này có một số đặc điểm như sau:

- + Làm việc với các phiên bản windows (WIN 98, NT, 200 and XP)
- + Sử dụng phương pháp điều khiển được nhúng trong dự
- + Không yêu cầu phần mềm hoặc điều khiển thiết bị đặc biệt nào đi kèm
- + Các hàm điều khiển sẽ được tự động cài đặt và cấu hình khi dò được nạp vào
- + Không sử dụng hàm API nào đi kèm mà chỉ sử dụng hai hàm Inp32 và Out32
- + Dễ dàng sử dụng với VC++ và VB

Sơ đồ làm việc như Hình 3.4



Hình 3.4. Hoạt động của Inpout32.dll

Có hai khối xây dựng sẵn quan trọng của DLL:

1) Một điều khiển thiết bị kiểu nhân được nhúng trong DLL ở dạng nhị phân được thông qua Hwinterface.sys cho phép gọi 3 hàm 'DriverEntry', 'hwinterfaceunload', 'hwinterface Device Control' thực hiện nạp gỡ điều khiển thiết bị và liên kết với Deviceiocontrol API thực hiện đọc viết tùy theo mã chương trình sử dụng.

2) Bản thân DLL: Các hàm trong Inpout32.dll được thực hiện trong hai file "inpout32drv.cpp" và "osversion.cpp", osversion.cpp kiểm tra phiên bản của hệ điều hành còn "inpout32drv.cpp" thì cài đặt điều khiển thiết bị kiểu nhân (đã nói trên), nạp nó, thực hiện đọc/viết qua cổng song song...

Hai hàm chính thực hiện là:

1) 'Inp32', đọc dữ liệu từ thanh ghi cổng song song. Cú pháp:

OutPortAddress, ValueToWrite

Ví dụ: Out &h378, &h55

2) 'Out32', viết dữ liệu vào thanh ghi cổng song song. Cú pháp:

ValueRead = Inp(PortAddress)

Ví dụ: ValueRead = Inp(&h378)

Ngoài ra còn một số hàm khác có thể thực hiện trong Inpout32.dll như:

1) 'DllMain', được gọi khi dò được nạp hay gỡ, khi dò được nạp nó kiểm tra hệ điều hành và nạp nếu cần thiết,

2) 'Closedriver', đóng các điều khiển thiết bị đã mở trước khi gỡ điều khiển thiết bị (driver).

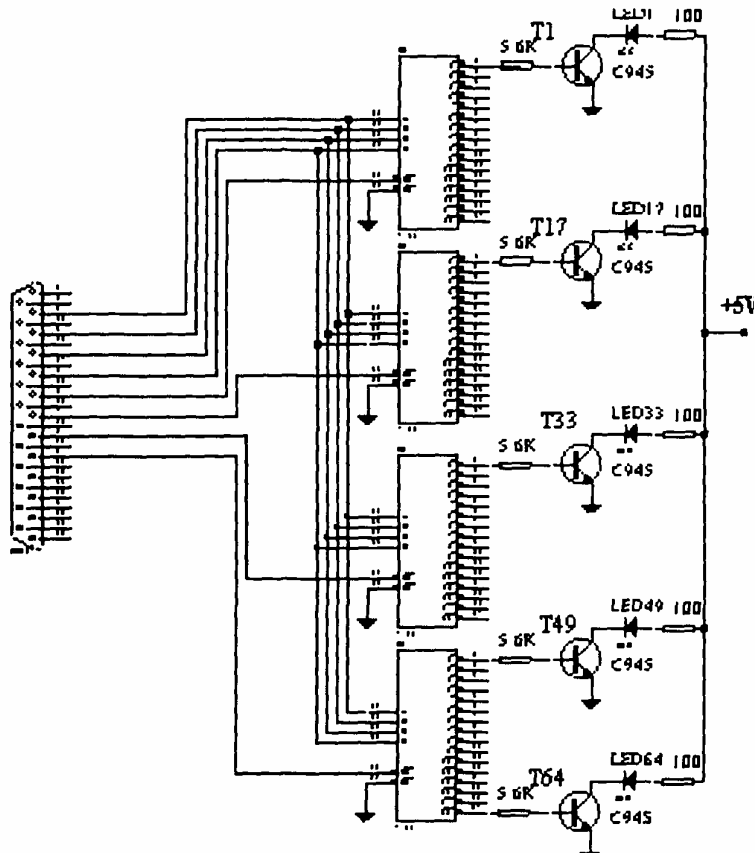
3) 'Opendriver', mở một liên kết tới điều khiển thiết bị hwinterface.

4) 'inst', lấy ra 'hwinterface.sys' từ dạng nhị phân đưa tới thư mục 'systemroot\drivers' và tạo một dịch vụ. Hàm này được gọi khi hàm 'Opendriver' thực hiện không thành công.

5) 'start', khởi động dịch vụ hwinterface sử dụng Control Manager APIS.

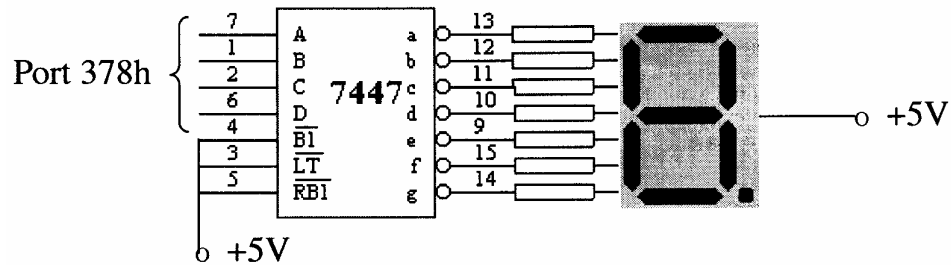
6) 'Systemversion' Kiểm tra phiên bản hệ điều hành và trả về mã thích hợp. Có thể dowload do này và mã nguồn từ trang web <http://www.logix4u.net/> sau đó thực hiện theo hướng dẫn của tác giả để sử dụng,

Trên đây chúng ta đã đề cập đến việc điều khiển ra ngoài qua cổng máy in với 8 bit. Để tăng số bộ điều khiển ra có thể kết hợp sử dụng các vi mạch giải mã, chuyển kênh... như 4051, 74154... Sơ đồ sử dụng 74154 cho phép điều khiển ra ngoài 64 bit qua cổng máy in song song như hình 3.5:



Hình 3.5. Điều khiển 64 tin qua cổng máy in

Điều khiển anh LED 7segment qua cổng máy in



Hình 3.6. Dùng vi mạch giải mã BCD- 7seg hiển thị tình đèn 7 đoạn

Chân A của 7447 được nối với D0 của 378h

Chân B của 7447 được nối với D1 của 378h

Chân C của 7447 được nối với D2 của 378h

Chân D của 7447 được nối với D3 của 378h

Các số hiển thị từ 0 đến 9 được đổi ra số BCD rồi đưa vào ABCD

Muốn hiển thị số 0 đưa vào 0000 các thanh a, b, c, d, e, f sẽ sáng

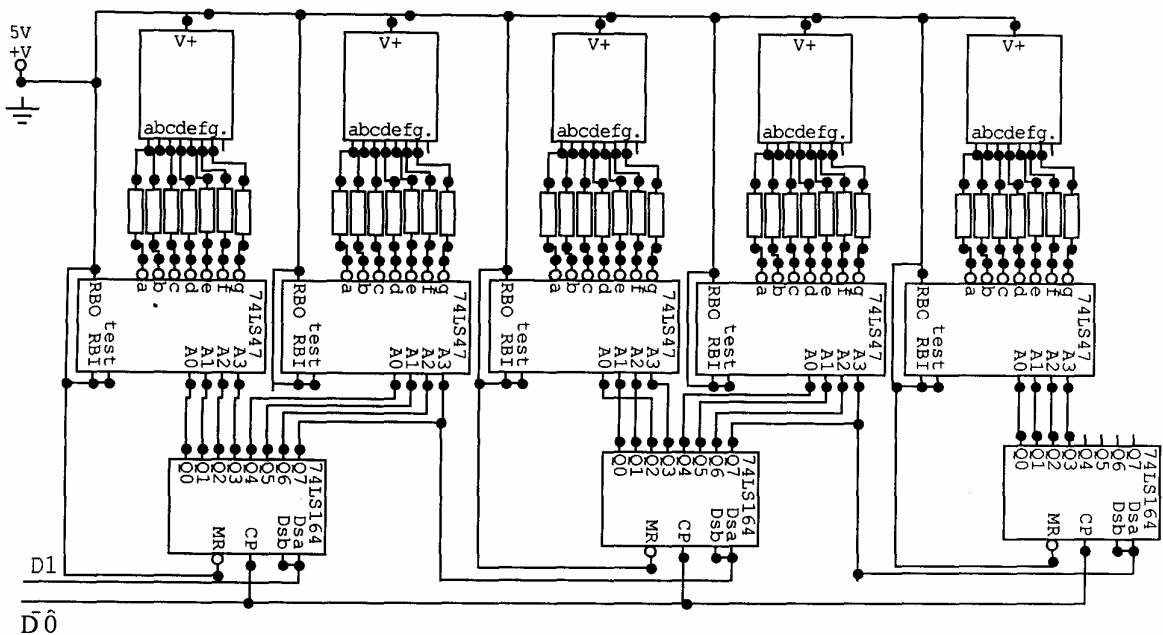
Muốn hiển thị số 1 đưa vào 0001 các thanh b, c sẽ sáng

...

Muốn hiển thị số 9 đưa vào 1001 các thanh a, b, c, f, g sẽ sáng.

Theo cách này với 8 bit của cổng 378h có thể hiển thị hai LED. Nếu sử dụng thêm các mạch phụ có thể tăng thêm số bộ điều khiển qua cổng 378h. Chẳng hạn có thể sử dụng thêm mạch dồn kênh, 8255A... Tuy nhiên việc điều khiển cũng vẫn bị hạn chế.

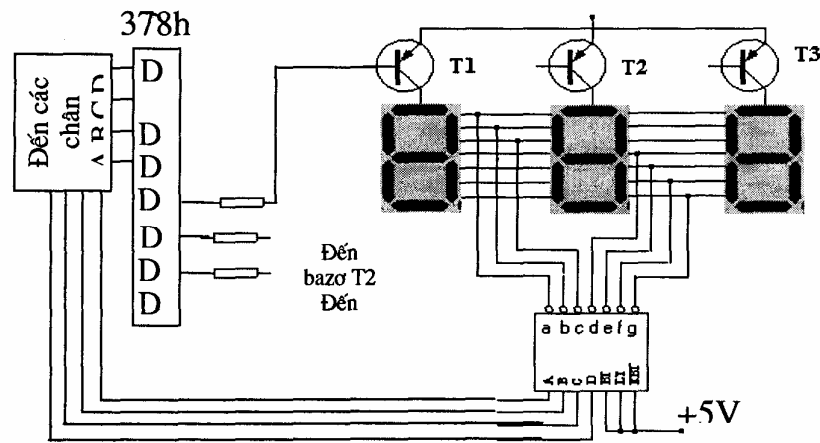
Để tăng số LED điều khiển được ta có thể sử dụng sơ đồ sau, về mặt lý thuyết có thể điều khiển được số LED không hạn chế theo phương pháp hiển thị ảnh và dữ liệu truyền cho các LED nối tiếp qua 1 bit của cổng LPT. Hoạt động của mạch điện như sau: Để 5 đèn LED sáng sử dụng 5 vi mạch 7447 và 3 vi mạch 74164. Các số cần hiển thị ra các đèn LED 7 đoạn được đổi ra số BCD sau đó được truyền nối tiếp qua các bit DI của cổng máy in đến vi mạch 74164 và vi mạch 7447 rồi mới tới đèn LED. Vi mạch 74164 là thanh ghi dịch nên sau khi đưa từng bit vào chân A, B tiếp theo đưa xung CLK = 1 (Qua bit D₀ của 378h) vào chân CP, các bit dữ liệu đầu vào sẽ được đưa vào thanh ghi dịch và được dịch dần sang phải, mỗi xung clock dịch một bit. Như vậy thì số cuối cùng của dãy số cần hiện sẽ được hiển thị ở đèn LED đầu tiên sau đó cứ dịch dần đến đèn LED cuối cùng. Khi các bit được đưa ra hết thì các đèn sẽ hiện các con số theo yêu cầu.



Hình 3.7. Hiển thị nhiều LED bằng 2 bit điều khiển

Điều khiển hiển thị động LED 7segment qua cổng máy in.

Khi sử dụng hiển thị tĩnh dòng điện tiêu thụ để đất LED rất lớn do các LED sáng liên tục. Để giảm dòng điện sử dụng, tăng số LED điều khiển có thể dùng phương pháp hiển thị động như sơ đồ sau:



Hình 3.8. Hiển thị động 3 đèn LED 7 đoạn

Thực hiện: Các LED không được cấp nguồn sáng liên tục mà được cấp sáng 60 lần/giây. Do tính lưu ảnh trên võng mạc của người mà ta có cảm giác các đèn sáng liên tục không nhấp nháy trong khi năng lượng tiêu thụ cho đèn giảm đi rất nhiều.

Các bit D0 đến D3 vào 4 đầu 7447 hiển thị các số từ 0 đến 9.

Các bit D4, D4, D6 sẽ cho đèn nào thông → đèn 7 đoạn đó sẽ sáng

VD cần hiển thị số 178. Thuật toán điều khiển như sau:

B1:

+ Cổng 378h nhận giá trị 0001 0001 - (Các đèn đều có thể sáng số 1 nhưng chỉ đèn 1 có thể sáng).

+ Trễ 2ms

B2:

+ Cổng 378h nhận giá trị 0010 0111 - (Các đèn đều có thể sáng số 7 nhưng chỉ đèn 2 có thể sáng).

+ Trễ 2ms

+ Cổng 378h nhận giá trị 0100 1000 - (Các đèn đều có thể sáng số 8 nhưng chỉ đèn 3 có thể sáng),

+ Trễ 2ms

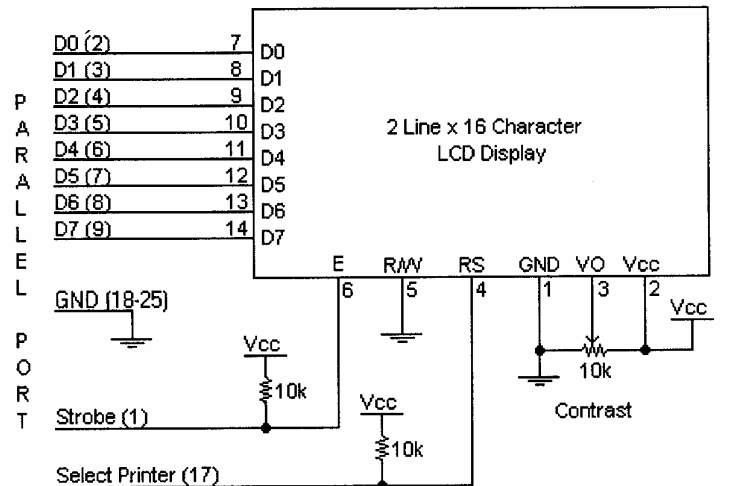
Việc này lặp 60 lần / giây

Với bit D7 ta có thể mắc thêm 1 đèn 7 đoạn nữa

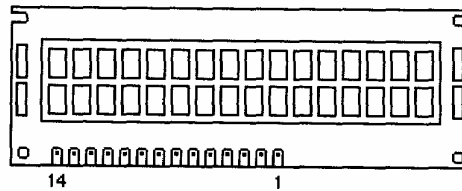
Điều khiển LCD:

Ví dụ này được áp dụng cho tất cả các loại cổng máy in song song dù có phải là loại Bi-direct hay không. LCD được ghép vào cổng song song là loại LCD hai đường 16 ký tự

rất thông dụng có thể mua trên thị trường với giá khoảng 70 nghìn VNĐ (1-2007).



Hình 3.9. Mạch điện LCD ghép với cổng máy in song song



Các chân *Enable* (chân E số 6) và chân *Register Select* (chân RS số 4) được nối với bit STROUBE và Select Printer của thanh ghi điều khiển. Thanh ghi này có đầu ra open collector / open drain output. Hầu hết các cổng song song đều có các điện trở kéo lên ở bên trong, một số thì không có. Vì vậy nếu thêm vào một điện trở kéo lên lớn sẽ làm mạch có thể dùng cho nhiều thể loại máy tính khác nhau. Chân 5 của LCD được nối đất để nó chỉ làm việc ở mô-đem hiển thị. Điều chỉnh độ tương phản của LCD bằng chiết áp LCD nối các chân 1, 2, 3. Nguồn cung cấp là +5V, chú ý cho thêm tụ lọc.

LCD hai đường 16 ký tự có thể sử dụng loại thông dụng HD44780. Mã nguồn của chương trình điều khiển:

Programming - Source Code

```
/* LCD Module Software */
/* 17th May 1997 */
/* Copyright 1997 Cung Peacock */
/* WWW - http://www.senet.com.au/cpeacock*/
/* Email - cpeacock@senet.com.au*/
/* */
/* Register Select must be connected to Select
```

```

Printer (PIN 17) * /

/* Enable must be connected to Strobe (PINI) */

/* DATA 0: 7 Connected to DATA 0: 7 * /

#include <dos.h>

# include <string.h>

#define PORTADDRESS 0x378/*Enter Your Port Address Here */

#define DATA PORTADDRESS+0

#define STATUS PORTADDRESS+1

#define CONTROL PORTADDRESS+2

void main (void)

{

char string [ ] = { "Testing 1, 2, 3

"

"It' Works ! "

char init [10];

int count;

int len;

init [0]: 0x0F; /* Init Display */

init [1] = 0x01; / * Clear Display * /

init [2] = 0x38t;/* Dual Line / 8 Bits */

outportb (CONTROL, inportb (CONTROL) & 0XDF);

/* Reset Control Port - Make sure Forward Direction

*/ outportb (CONTROL, inportb (CONTROL) | 0x08);

/* Set Select Printer (Register Select) */

for (count: 0; count <= 2; count ++)

{

outportb (DATA, init [count]);

outportb (CONTROL, inportb (CONTROL) | 0x01);

/* Set Strobe (Enable) */

delay (20);

```

```

/* Larger Delay for INIT */
outportb (CONTROL, inportb (CONTROL) & 0xFE),
/* Reset Strobe (Enable) */
delay (20);
/* Larger Delay for INIT */
outportb (CONTROL, inportb (CONTROL) & 0xF 7);
/* Reset Select Printer (Register Select) */
len = strlen (string);
for (count = 0; count < len; count++)
{
outportb (DATA, string [count]);
outportb (CONTROL, inportb (CONTROL) | 0x01);
/* Set Strobe */
delay (2);
outportb (CONTROL, inportb (CONTROL) & 0xFE);
/* Reset Strobe */
delay (2);

```

Chương trình trên được viết bằng Borland C, nếu muốn sử dụng bộ dịch của Microsoft thì thay *outportb()* bằng *outp()* và *inportb()* bằng *inp()*.

Vòng lặp đầu tiên thực hiện bật và khởi tạo LCD. Các lệnh này gửi tới thanh ghi lệnh (*Instruction Register*) của LCD được điều khiển bởi chân lựa chọn thanh ghi (chân 4). Khi chân 4 ở mức thấp thanh ghi lệnh được chọn, khi nó ở mức cao thanh ghi dữ liệu được chọn. Đường *Select Printer* của cổng máy in song song có mạch đảo nên khi ta viết '1' tới bit 3 thì đường *Select Printer* ở mức thấp. Câu lệnh: *outportb (CONTROL, inportb (CONTROL) 10 x 08);* sẽ làm set bit 3 và không thay đổi các bit khác.

Sau khi đặt một byte dữ liệu vào các chân dữ liệu phải ra lệnh cho modul LCD đọc dữ liệu. Để làm việc này sử dụng chân *Enable*. Dữ liệu được đưa vào module LCD trong mức cao hoặc mức thấp. Chân *Strobe* cũng có mạch đảo vì vậy set bit 0 của thanh ghi điều khiển (Control Register) chúng ta nhận được đường *Strobe* ra mức thấp. Sau khi đợi một khoảng thời gian trễ đường này trở lại mức cao cho byte tiếp theo. Sau khi khởi tạo modul LCD chúng ta muốn gửi text tới nó. Các ký tự được gửi tới cổng dữ liệu (*Data Port*) của LCD, vì vậy chúng ta muốn xoá bit 3. Một lần nữa chúng ta lại phải thay đổi một bit sử dụng câu lệnh *outportb (CONTROL, inportb (CONTROL) &*

0xF7);. khi đó chúng ta thiết lập một vòng lặp for khác đọc một byte từ chuỗi và gửi nó tới LCD. Việc này lặp với độ dài của chuỗi. Sự trễ phù hợp với các máy phổ biến nhất. Nếu các LCD không được khởi tạo tương thích thì cần tăng trễ lên, tương tự như vậy nếu panel là các ký tự bỏ cách ví dụ Tst hoặc bị lặp ký tự như TTeessttiinngg là chúng ta có thể có lỗi kết nối *Enable* và cần kiểm tra lại *Enable* và *Strobe*.

- Điều khiển motor bước qua cổng máy in.

Động cơ bước là động cơ khi chuyển động dịch chuyển từng bước một. Mỗi bước có thể 0.9^0 , 1.8^0 , 3.6^0 tùy loại động cơ. Động cơ bước có thể hai pha, bốn pha hoặc nhiều pha. Động cơ bước thực chất là một động cơ không đồng bộ dùng để biến đổi các tín hiệu điều khiển dưới dạng các xung điện rời rạc kế tiếp nhau thành các chuyển động góc quay hoặc các chuyển động của rô to và có khả năng cố định rô to vào những vị trí cần thiết.

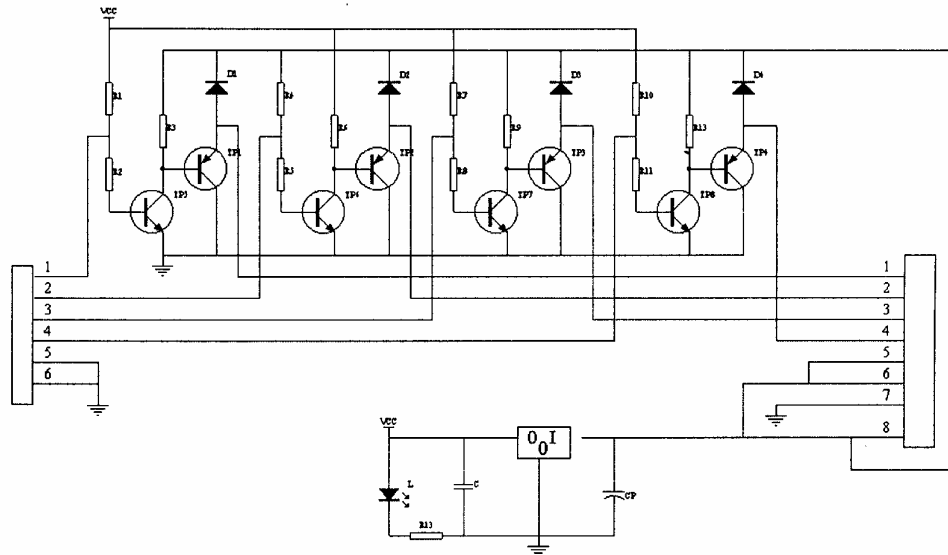
Ta có thể dùng 4 bit của thanh ghi 378h cổng máy in kết hợp với mạch công suất để điều khiển để điều khiển một động cơ bước 4 pha như sơ đồ hình 3.10 như các bit D0, D1, D2, D3 được nối vào các đầu vào điều khiển của mạch công suất. VI ạch điện cho phép thực hiện điều khiển động cơ theo các phương pháp :

- + Cả bước
- + Nửa bước
- + Vi bước

Việc kích các cuộn dây có thể lcuộn, 2 cuộn

Bảng 3.1. Tín hiệu kích các cuộn dây

Kích 1 pha				Kích 2 pha				Kích 1 - 2 pha			
	D1	D2	D3	D0	D1	D2	D3	D0	D1	D2	D3
1	0	0	0	1	1	0	0	1	0	0	0
0	1	0	0	0	1	1	0	1	1	0	0
0	0	1	0	0	0	1	1	0	1	0	0
0	0	0	1	1	0	0	1	0	1	1	0
1	0	0	0	1	1	0	0	0	0	1	0
0	1	0	0	0	1	1	0	0	0	1	1
0	0	1	0	0	0	1	1	0	0	0	1
0	0	0	1	1	0	0	1	1	0	0	1

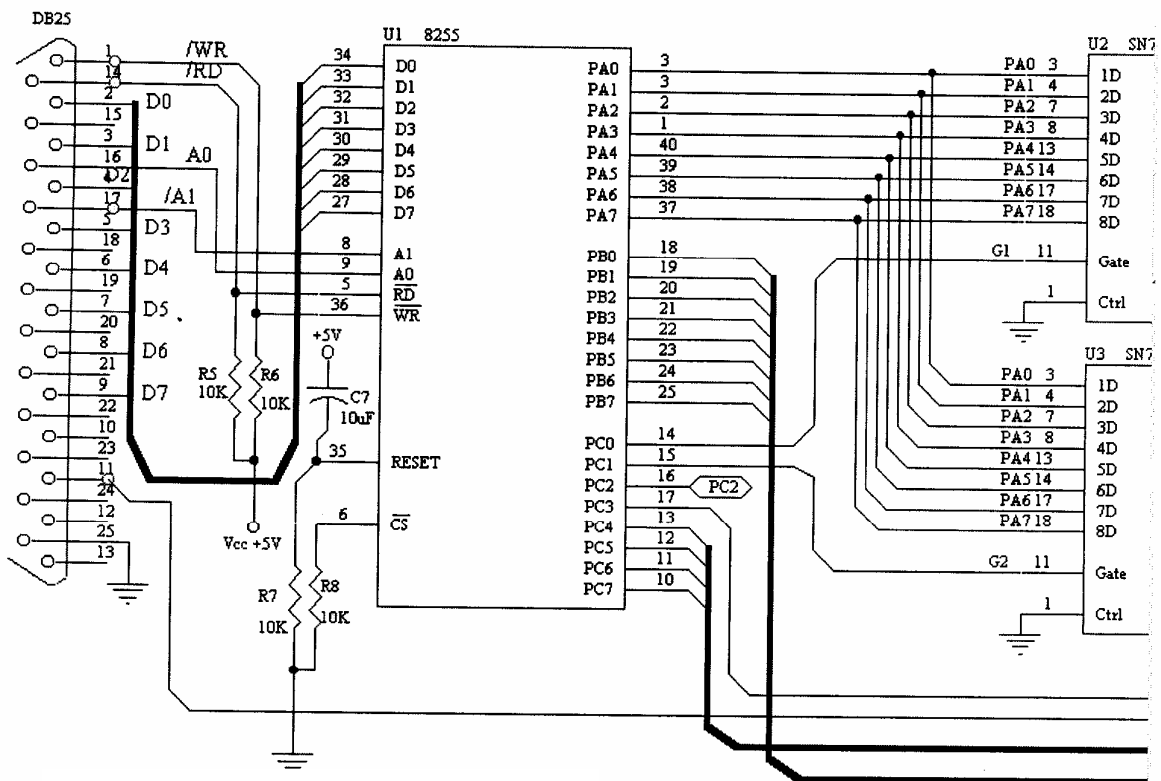


Hình 3.10. Mạch điện điều khiển mô tơ bước 4 cuộn dây

- Điều khiển một mạch DAC qua cổng máy in

Xem Hình 1.40

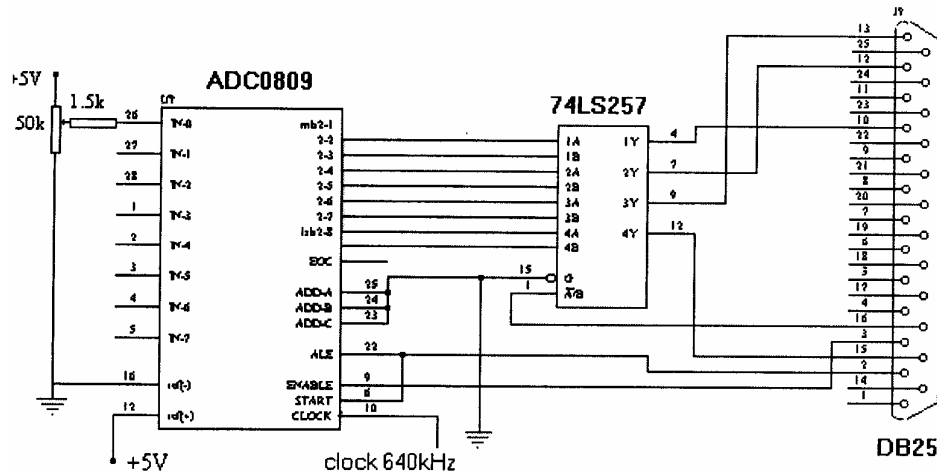
- Ghép 8255 với cổng máy in song song



Hình 3.11. Ghép 8255A với cổng máy in

- Dữ liệu được đưa từ máy tính sang máy in trong quá trình in:
 - c) Đưa dữ liệu vào máy vi tính qua cổng song song

- Thanh ghi dữ liệu của cổng máy in song song



Hình 3.12. Đưa dữ liệu số vào máy tính qua thanh ghi 379h của cổng máy in

Một số cổng máy in sử dụng giao diện bi-direct cho phép truyền dữ liệu 8bit qua thanh ghi dữ liệu theo hai hướng. Với loại này, để cho phép nhận dữ liệu vào qua địa chỉ cơ sở người ta phải thiết lập bit D5 (hoặc có thể D6 - hoặc cả hai) [] lên mức cao. Nói chung trong thực tế các hãng sản xuất mainboard không phải khi nào cũng chế tạo giao diện song song Centronics kiểu bi-direct. Theo một số báo cáo thực nghiệm [3] cho thấy khi thiết lập bit D6 rồi thì cổng 378h chỉ vào được 4 bit thấp thậm chí một số trường hợp không tác dụng gì.

(Xem thêm các sơ đồ ghép nối 8255A, ADC 7109, 8254,... với cổng song song trong các tài liệu tham khảo).

3.1.2. Ghép nối qua cổng nối tiếp

a) Giao diện RS232, 485

- Giao diện RS-232: Chuẩn RS 232 là giao diện phổ biến do EIA (Electro Industrie American) đưa ra từ năm 1969. Do tính thuận tiện của nó nên hầu hết các thiết bị khi cần 1 giao diện ghép nối máy tính người ta hay sử dụng RS232, nó có mặt trong rất nhiều thiết bị.

- Truyền tin nối tiếp: Trong kiểu truyền tin nối tiếp, dữ liệu vào/ra máy tính theo từng bit. (Giải thích "truyền").

- Truyền tin nối tiếp có 2 kiểu: đồng bộ và không đồng bộ

- Chuẩn cơ khí: Dùng 2 loại DB9 và DB29 Ma le và Female



Hình 3.13. Đầu nối cơ khí DB25 và DB9

Bảng 3.2. Chức năng các chân tín hiệu giao diện nối tiếp

DB9	DB25	Chức năng
1	8	DCD- Data Carrier Detect Lỗi vào
2	3	RxD - Receive Data Lỗi vào
3	2	TxD - Transmit Data Lỗi ra
4	20	DTR - Data Temlinal Ready Lỗi ra
5	7	GND - Nối đất
6	6	DSR - Data Sét Ready Lỗi vào
7	4	RTS - Request to Send Lỗi ra
8	5	CTS - Clear to Send Lỗi vào
9	22	RI - Ring Indicator Lỗi vào

- Mức tín hiệu logic nằm trong khoảng - 12V đến + 12V

Mức 1: -3 V đến - 12V

Mức 0: Từ +3V đến + 12V

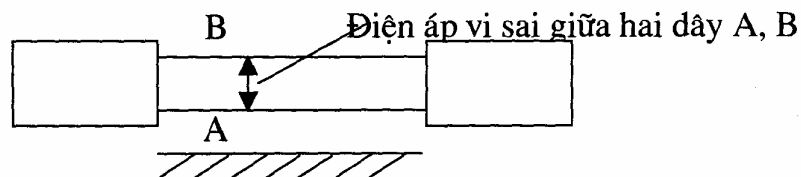
- Tốc độ truyền: 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19.200. Thường dùng nhất là 4800 và 9600 baud. Với tín hiệu có hai mức 0 và 1 một baud = 1bit/giây.

- Khi không có thiết bị phụ trợ thì khoảng cách truyền an toàn qua RS 232 giữa 2 thiết bị chỉ từ 15-20m mặc dù về lý thuyết cho phép truyền đến 100feet.

• Giao diện RS-422

- Giao diện truyền tin nối tiếp RS-422 có tốc độ cao hơn: đến 10Mbit/giây và khoảng cách truyền lớn hơn: Đến 1200m,

Mức logic không tính theo sự thay đổi mức điện áp tín hiệu so với mass mà tính theo điện áp vi sai giữa hai dây dẫn. Bộ đệm đường dẫn của RS-422 tạo ra một điện áp vi sai khoảng 5V truyền qua hai dây xoắn. Bộ phối hợp mức bên nhận sẽ đo điện áp vi sai để phân biệt hai mức "1" và "0".



Hình 3.14. Tín hiệu của chuẩn RS 422

- Trong chuẩn RS-422 một cặp tín hiệu được sử dụng để truyền dữ liệu chứ không

phải 1 tín hiệu. Cặp này là:

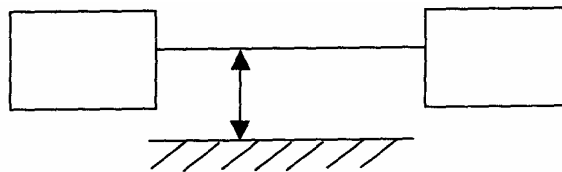
- + Tín hiệu không đảo A
- + Tín hiệu đảo B

Chênh lệch điện áp giữa A và B trong khoảng 2V đến 6V. Khi A âm so với B có mức "1". Khi A dương so với B có mức "0",

- Giao diện RS-423A

Sử dụng điện áp không đối xứng. Tiêu chuẩn chỉ dùng một đường dẫn để truyền, giống như trong RS-232. Tuy vậy các thông số đã được cải tiến để có tốc độ truyền cao hơn, độ dài truyền xa hơn. Giá trị mức điện áp từ 0-6V: 0V: mức "1"; 6V: mức "0".

Điện áp tín hiệu so với mass



Hình 3.15 Tín hiệu của chuẩn RS 423

- Giao diện RS – 485:

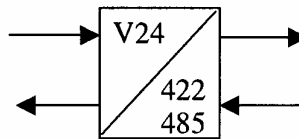
Là sự mở rộng của RS-422. Tuy nhiên có sự cải tiến: cùng độ dài đường truyền và tốc độ tiêu chuẩn, ghép nối này cho phép nhiều hơn hai thành viên tham gia (có thể tới 32 thành viên tham gia)

Mức "1" chênh lệch điện áp vi sai: - 1.5V đến -6V

Mức "0" chênh lệch điện áp vi sai: +1.5V đến +6V

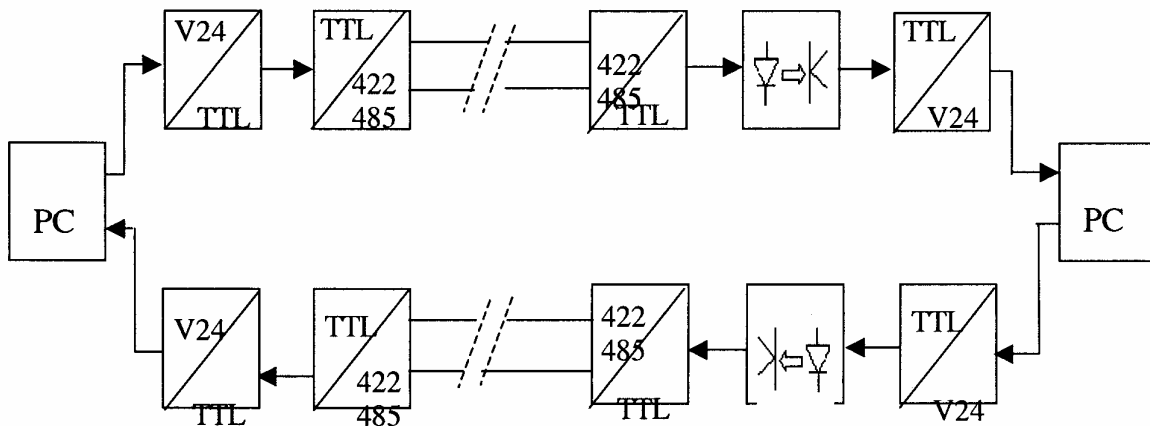
Tốc độ truyền đến 10MBaud

- Bộ ghép nối RS232- RS485



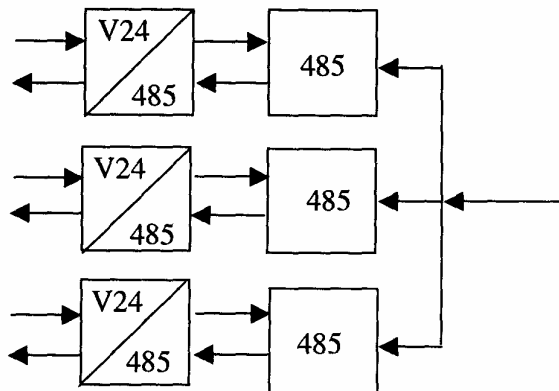
Hình 3.16. Chuyển đổi RS232-RS485

Phối hợp hai chiều



Hình 3.17. Phối hợp 2 chiều RS232-RS485

Sử dụng chuẩn 485 có thể ghép thành bus



Hình 3.18. Thiết bị ghép vào bus RS485

b) Vào ra dữ liệu bằng Polling-truyền tin qua cổng nối tiếp

• Các thanh ghi UART8250:

Tốc độ truyền tính qua cổng truyền tin nối tiếp điều khiển được nhờ vào việc chọn xung đồng hồ vào X tall và việc chọn số chia của tần số này (Đề tại thanh ghi LSB và MSB). Ví dụ nếu ta dùng tần số đồng hồ 1.8432 MHz tại X tall ta muốn tốc độ phát là 9600 baud ta tìm số chia để ghi vào LSB và MSB như sau:

Số chia = Tần số vào/(Tốc độ phát x 16) = $1.8432 \cdot 10^6 / (9600 \cdot 16) = 12$ Vì mạch có 8 thanh ghi-trong đó thanh ghi địa chỉ cơ sở tùy theo giá trị của bit DLAB (bit D7 của thanh ghi điều khiển đường truyền) có thể làm chức năng đệm thu, đệm phát hoặc là thanh ghi cho phép tạo yêu cầu ngắt, thanh ghi số chia phần thập như trong bảng sau:

Bảng 3.3. Các thanh ghi cổng truyền tin nối tiếp

DLA	A2	A1	A0	Chọn ra
0	0	0	0	Thanh ghi đệm thu, Thanh ghi gửi phát(XF8)
0	0	0	1	Thanh ghi cho phép tạo yêu cầu ngắt (IER) XF9

1	0	0	0	Thanh ghi cho số chia phần thấp(LSB)
1	0	0	1	Thanh ghi cho số chia phần cao(MSB)
x	0	1	0	Thanh ghi nhận dạng nguồn gốc yêu cầu ngắt
x	0	1	1	Thanh ghi điều khiển đường dây (LSR - XFB)
x	1	0	0	Thanh ghi điều khiển modem (MCR)
x	1	0	1	Thanh ghi trạng thái đường dây (LSR XFD)
x	1	1	0	Thanh ghi trạng thái modem
x	1	1	1	Thanh ghi nháp dành cho CPU ít sử dụng

Trong các ứng dụng thu phát thông thường, sử dụng Topology Point - To - Point Nghe Modem sử dụng 3 dây các thanh ghi được sử dụng là:

- Thanh ghi địa chỉ cơ sở XF8
- Thanh ghi LSR địa chỉ XFD

Ta sẽ giới thiệu 2 thanh ghi này

- Thanh ghi địa chỉ cơ sở là thanh ghi tại đây ta có thể nhận 1 dữ liệu đến từ 1 hệ vi xử lý (máy tính) khác hoặc phát đi 1 dữ liệu. Vùng đệm có độ lớn 1 byte. Đây là vùng đệm kép, bao gồm đệm giữ và đệm phát: Trong khi 1 ký tự đang được truyền đi ở đệm phát thì 1 ký tự khác có thể được đưa từ CPU sang đệm giữ.

Trong máy tính PC có 4 cổng truyền tin nối tiếp. Địa chỉ cơ sở của các cổng như sau:

- COM1: 3F8-3FFh dùng IRQ4
- COM2: 2F8-2FFh dùng IRQ3
- COM3: 3E8-3EFh dùng IRQ4
- COM4: 2E8-2EFh dùng IRQ3

Trong các Mainboard thường chỉ có sẵn 2 cổng COM1 và COM2. Muốn sử dụng COM3 và COM4 ta phải cắm thêm 1 cam phối ghép I/O ngoài.

Thanh ghi trạng thái đường truyền LSR (Line Status Register): có địa chỉ = Địa chỉ cơ sở +5,

Dạng thức của thanh ghi

D7	D6	D5	D4	D3	D2	D1	D0
----	----	----	----	----	----	----	----

D0: RXDR: Receiver data ready:

= 1 đã nhận được 1 dữ liệu và để nó trong thanh ghi đệm thu RBR. bit này bị xóa

khi CPU đọc dữ liệu ở RBR.

D1: OR - Lỗi do thu đề

D2: PE – Lỗi Parity

D3: FE – Lỗi khung

D4: BI - Có sự gián đoạn trong khi truyền

D5: THRE: Transmitter holding register empty (Thanh ghi phát rỗng)

= 1 khi ký tự đã được chuyển từ THR sang TSR bit này bị xoá khi CPU đưa ký tự tới thanh ghi THR.

D6: TSRE - Thanh ghi dịch phát rỗng,

D7 luôn bằng 0,

Việc truyền tin qua các cổng nối tiếp của máy tính PC được tiến hành theo các bước sau:

- *Init các cổng: Tên cổng truyền (nhận), tốc độ phát (thu), khuôn dạng gói tin.*

- *Test thanh ghi LSR với các bit trạng thái tương ứng để quyết định gửi, nhận dữ liệu theo 1 giao thức nào đó.*

Ví dụ:

1 đoạn chương trình khởi đầu cho COM2:

```
mov al, 80h
```

```
mov dx, 2fbh
```

```
out dx, al
```

```
mov al, 24; tốc độ 4800
```

```
mov dx, 2f8h
```

```
out dx, al
```

```
mov al, 0
```

```
mov dx, 2f9h
```

```
out dx, al
```

```
mov al, lah; từ khuôn dạng dữ liệu: Parity chẵn, 1 bit stop, 7 bit mã
```

```
mov dx, 2fb
```

```
out dx, al
```

Nếu dùng Pascal:

```
(* chương trình doi thu ky tu tu 1 PC khac den 11/14/2000*)
uses crt, dos;
var kytu: byte;
i,j : longint;
BEGIN
port [$3fb]:=$80;
port [$3f8]:=12;
port [$3f9]:=0;
port [$3fb]:=2;
i =1; j:=0;
Repeat
clrscr;
port [$3fd]:= port [$3fd] and $01;
    if port [$3fd] <>$01 then { DOI THU }
begin gotoxy (30,12); writeln (' Dang doi thu tin hieu... ');
delay(50);
end
else THU KY TU}
begin
kyt u::port [$3f 8 ];
writeln ('Ky tu nhan duoc: ', chr (kytu), ' So ky tu thu duoc: '); delay (10); {port
[$3f8]:=$5; }
i =i+1;
end;
{ PHAT TRA LAI }
asm
@ 1: mov dx, 03 fdh
in al, dx
and al, 20h
cmp al, 20h
je @2
```

```

jmp @1
@ 2: máy ai, kytu
mov dx, 3f8h
out dx, al
end;
Until keypressed; { writeln (' So ky tu nhan sai ', j); }
END.

uses crt, dos;
const a=$2f8;
var kt_phat, kt_thu:byte;
begin
{writeln;
writeln (' Write (' Dang phai chu U... '); }
port [a+3]:=$80;
port ra]:=12; {9600}
port [ a + 1]:=0;
port [ a+3 ]:= 2; { none parity, 7 }
{i:=1, j:=0;}
repeat
port [a]:-$55; delay(1);
Repeat
port [$3fd]:= port [$3fd] and $01;
until port [$3fd]=$01;
kt_thu:=port [$ 3f 8];
writeln (' Thu được ', chi (kt_thu));
until keypressed;
end.

```

- Thu dữ liệu qua cổng nối tiếp

Có thể sử dụng cổng nối tiếp để thu dữ liệu. Một trong những trường hợp dễ thấy là sử dụng modem để truyền nhận tin qua cổng nối tiếp. Khi đó phải lập trình cho modem với các hệ lệnh riêng. Ở đây chỉ giới thiệu một số khả năng ghép nối trực tiếp

thiết bị với cổng truyền tin nối tiếp.

- + Ghép nối ADC ICL7 109 với cổng truyền tin nối tiếp
- + Ghép nối ADC MAX 186 với cổng truyền tin nối tiếp

c) Vào ra dữ liệu dùng ngắt

So sánh sử dụng polling và ngắt

- + Polling chiếm thời gian CPU, chậm
- + Dùng ngắt đáp ứng kịp thời nhu cầu truyền tin có dung lượng lớn
- + Viết chương trình ngắt phức tạp bởi liên quan đến các vấn đề: cấm ngắt, cho ngắt, nhận dạng nguồn ngắt, thay vector ngắt...

- Các nguồn gây ngắt trong 8250A và mức ưu tiên:

Có thể cấm hoặc cho phép ngắt 8250A bằng thanh ghi ĐCCS +1 khi bit DLAB =0.

Khuôn dạng thanh ghi cho phép ngắt (IER Interrupt enable register)

0	0	0	0	MODE	RLINE	TXEMP	RXRD
---	---	---	---	------	-------	-------	------

MODEM = 1 Cho phép các thay đổi trạng thái của modem gây ngắt

RLINE = 1 Cho phép các tín hiệu trạng thái báo đường dây thu gây ngắt

TXEMPTY =1 Cho phép gây ngắt khi đệm giữ phát rỗng

RxRDY = Cho phép gây ngắt khi đệm thu đầy

8250A chỉ có 1 đầu tác động ngắt đến CPU trong khi đó nó 4 nguồn gây ra ngắt vì vậy cần sử dụng thanh ghi nhận dạng nguồn ngắt IIR - Interrupt Identification Register,

ID0 = 0 Có yêu cầu ngắt = 1 Không có

0	0	0	0	0	IDI	ID2	IDO
---	---	---	---	---	-----	-----	-----

Tổ hợp ID1 và ID2 mã hoá yêu cầu ngắt có mức ưu tiên cao nhất đang chờ được phục vụ.

Bảng 3.4. Các nguồn ngắt nối tiếp và giá trị các bộ thanh ghi nhận dạng nguồn ngắt

ID2	ID1	Mức	Tên loại ngắt	Nguồn gốc ngắt	IDi bị xoá khi
1	1	1	Trạng thái đường thu	Lỗi khung, thu dề, lỗi parity gián đoạn khi thu	Đọc LSR
1	0	2	Đệm thu đầy	Đệm thu đầy	Đọc RBR
0	1	3	Đệm giữ phát rỗng	Đệm giữ phát rỗng	Đọc IIR ghi THR
0	0	4	Trạng thái modem	ΔCTS, ΔDSR, ΔRI, ΔRLSD	Đọc MSR

Trình tự của một chương trình thu/ phát dùng ngắt

- + Cắm báo ngắt cứng
- + Xoá nguồn báo ngắt
- + Đặt khuôn dạng truyền
- + Đặt số chia
- + DLAB=0
- + Xoá báo ngắt phát
- + Xoá báo ngắt nhận
- + Đặt nguồn ngắt là thu hay phát
- + Bảo vệ ngắt cũ
- + Thay vectơ ngắt mới
- + Cho phép báo ngắt cứng

Các lệnh thu hay nhận dữ liệu được viết trong chương trình của chúng ta dưới dạng chương trình phục vụ ngắt. Khi có ngắt (thu hay nhận) chương trình này sẽ thực hiện thu hay nhận dữ liệu.

3.1.3. Ghép nối qua cổng USB

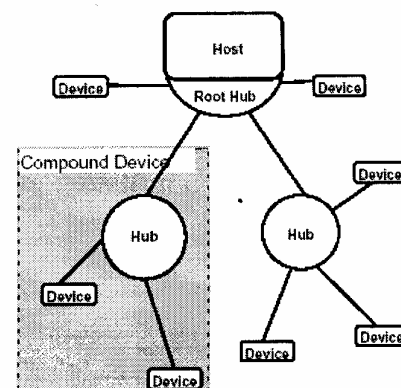
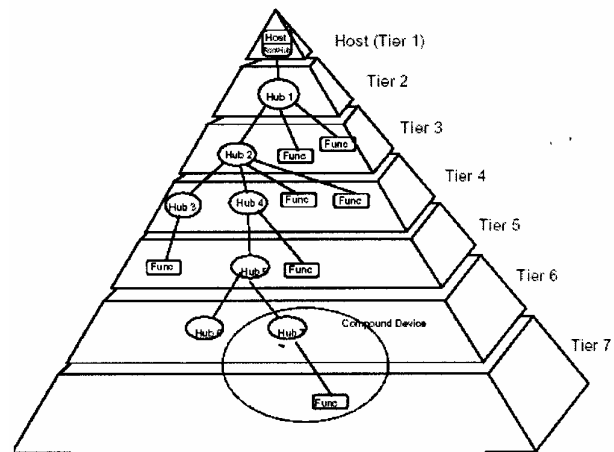
a) Cấu tạo của hệ thống USB

Được phát triển từ năm 1995. Phiên bản USB đầu tiên là USB 1.0 và phiên bản đang sử dụng hiện nay (2005) là USB 2.0. Mục đích chính là tạo ra một bus ngoài mở rộng ghép nối các thiết bị ngoại vi một cách dễ dàng. Các thiết bị USB có thể được cắm chồng lên nhau, cắm không cần tắt điện hay khởi động lại máy tính. Có thể ghép vào một bản mạch máy tính 127 thiết bị USB -

Tuy nhiên hiện nay trên các mainboard thường nhà chế tạo chỉ làm sẵn 2,4 hoặc 8 đầu nối USB. Số

lượng chủng loại thiết bị ghép với máy tính hiện nay đã có hàng trăm loại bao gồm máy in, máy ảnh, bàn phím, con chuột, modem, máy quét, ổ flash.... Có các loại USB:

- Low speed: Tốc độ 10-100 kb/s. Ví dụ như bàn phím, con chuột, bút từ, điều khiển trò chơi (joystick).



- Medium speed: Tốc độ 500kb/s - 10Mb/s như ISDN, pho ne, audio, compressed Video (Videonén).

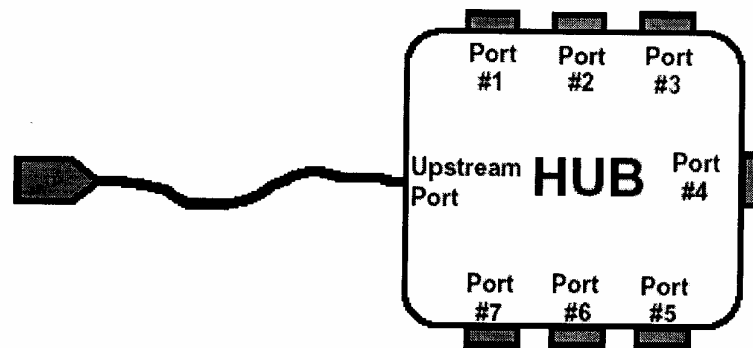
- High speed: Tốc độ 25-480Mb/s như Video, disk.

• Kiến trúc USB:

- Bus topology: Là cách nối các thiết bị USB với host USB. Topology của USB là hình sao thứ bậc. Một hub là trung tâm của sao. Kết nối giữa các đoạn dây dẫn là liên kết điểm - điểm. (*Root tier trên main board - Hub: các bộ tập trung-Node: các thiết bị USB*). Cần nhắc lại cấu trúc hiện đại của PC với sự tham gia của USB.

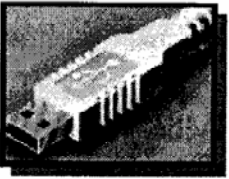
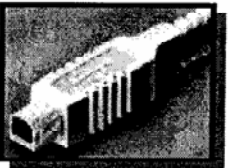
- USB host: Trong mỗi hệ USB chỉ có một. Có giao diện đến host Computer: Bộ điều khiển host (USB host controller) có nhiệm vụ chuyển đổi dữ liệu giữ format dữ liệu sử dụng và format USB, nó có thể được thực hiện trong một tí hợp phần cứng, phần mềm hoặc firmware. Một loạt hub cung cấp 1 hoặc một số điểm liên kết vào.

- Giao diện vật lý:



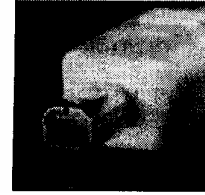
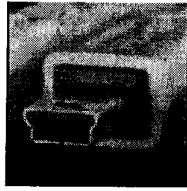
Hình 3.21. Kiến trúc HUB USB

- Giao diện cơ khí, điện:

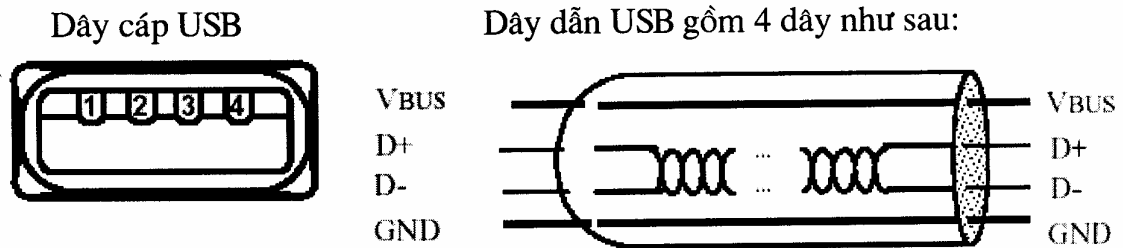
Series "A" Connectors	Series "B" Connectors
♦ Series "A" plugs are always oriented upstream towards the <i>Host System</i>  "A" Plugs (From the USB Device)	♦ Series "B" plugs are always oriented downstream towards the USB Device  "B" Plugs (From the Host System)

Hình 3.22. Hai kiểu đầu nối USB chính

Ngoài ra còn có một số kiểu đầu nối "mini" như sau:



Hình 3.23. Đầu nối USB kiểu mini



Hình 3.24. Dây cáp USB

V_{bus} (1) Red

Nguồn cung cấp cho USB (+5v)

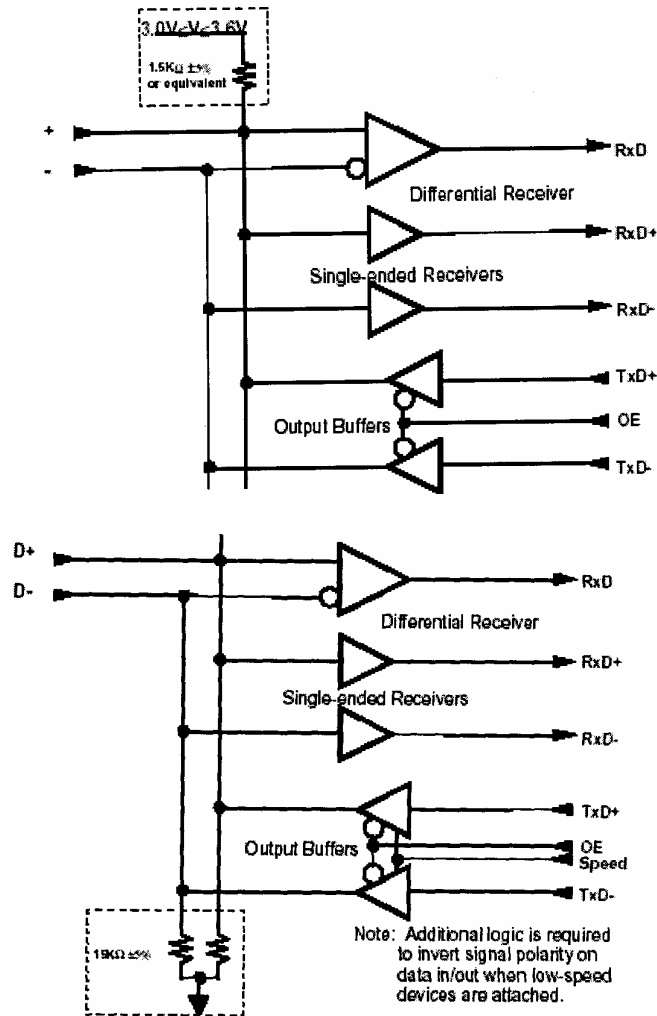
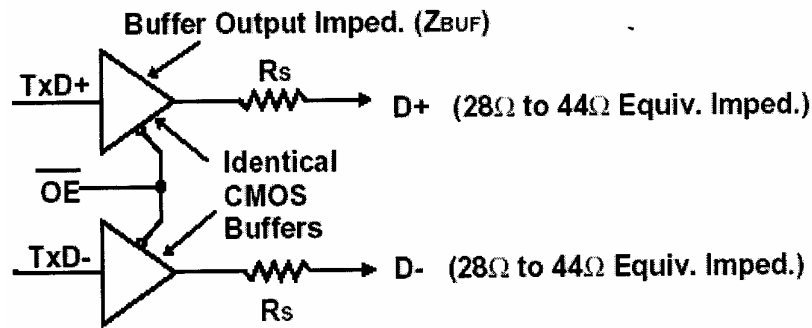
D⁻ (2) White

Hai dây D⁺ và D⁻ là hai dây truyền dữ liệu vì sai

D⁺ (3) Green

GND (4) Black

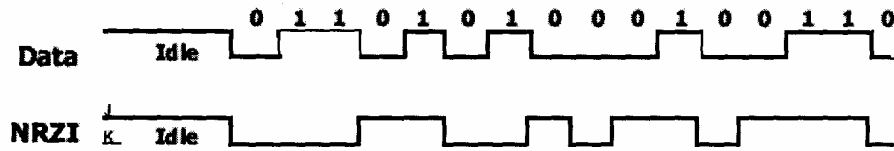
Dây nối đất (Vỏ máy) 0V



Hình 3.25. Các kết nối điện của đầu ra host hoặc Hub

+ Mã hoá USB

Dữ liệu truyền trên bus USB được mã hoá theo kiểu NRZI (Non Return to Zero Invert - Không đảo về không) theo sơ đồ sau với luật: 0 → 1: Không đổi; 1 → 0: Thay đổi; Nếu có 2 tín hiệu 0 liên tiếp: thay đổi.



Hình 3.26. Sơ đồ mã hoá NRZI

• Phần mềm USB: Phần mềm hệ thống USB chứa hai lớp

- + Một lớp trên của các điều khiển thiết bị USB
- + Một lớp dưới của các hàm USB

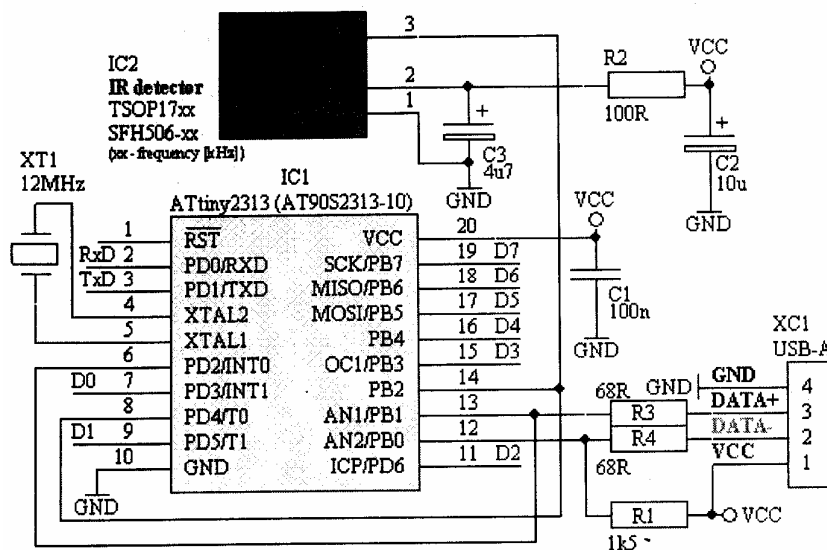
Lớp thấp của các hàm USB thực hiện bởi các tác vụ:

- + Điều khiển các hàm truyền tin giữa các thiết bị USB và Host
- + Nạp và gỡ các điều khiển thiết bị tại những thời điểm thích hợp
- + Truyền nhận các Frame và các packet USB

Khi thiết kế một thiết bị ngoại vi trao đổi dữ liệu qua USB các hàm điều khiển nhập xuất dữ liệu là các hàm có sẵn của các ngôn ngữ lập trình bậc cao chạy trên windows. Tùy theo ngôn ngữ VC hoặc Delphi, C# các hàm này có khuôn dạng khác nhau. Khi sử dụng ngôn ngữ. nào chúng ta có thể tham khảo trợ giúp của ngôn ngữ đó để sử dụng các hàm USB thích hợp:

b) Ví dụ sơ đồ ghép nối USB

Mạch điện sử dụng AVR AT90 S2313-10 có thể thu các mã hồng ngoại



Hình 3.2.7. AVR ghép với máy tính qua USB

Hoạt động:

Các dữ liệu D0.. D7 có thể được thu vào vi điều khiển AT90S2313-10 được

chương trình nạp trong vi điều khiển biến đổi thành dữ liệu dạng NRZI (như đã nói ở trên) rồi đưa tới máy tính PC qua hai dây D^+ và D^- . Chương trình trên máy tính PC (Được viết bằng các ngôn ngữ có giao diện truy nhập API như VC++, Delphi, C#) đọc các dữ liệu này chuyển thành dạng dữ liệu thông thường sau đó ghi vào file, hiển thị... tùy theo yêu cầu cụ thể.

Trên sơ đồ có sử dụng một bộ dò hồng ngoại cho phép nhận tín hiệu điều khiển qua mạch phát hồng ngoại. Khi có tín hiệu hồng ngoại sẽ gây ra ngắt vi điều khiển. Chương trình phục vụ ngắt sẽ thực hiện nhiệm vụ thu dữ liệu (hoặc tác động khác theo yêu cầu cụ thể).

3.1.4. Ghép nối qua các khe cắm ở rộng

a) Ghép nối qua khe cắm ISA

- ISA: Sử dụng với độ rộng bus 8 bit theo tiêu chuẩn ISA(Industry Standard Architecture): có 62 đường tín hiệu trên 2 mặt trên đó có các đường địa chỉ, dữ liệu, tín hiệu điều khiển, cấp nguồn...
- EISA: ISA mở rộng thêm 1 rãnh thứ 2 gồm 36 chân cho phép sử dụng độ rộng bus dữ liệu 16bit, 32 bit.
- Giải mã cho 1 card mở rộng qua khe cắm EISA:

Máy vi tính PC dành 1 KB địa chỉ cho các thiết bị ngoại vi. Mỗi thiết bị có 1 địa chỉ riêng.

VD: Các cổng COM: 3F8h, 2F8h, 3E8h, 2E8h

Các cổng máy in: 3BCh, 378h, 278h, 2BCh

Cổng bàn phím: 60h

Ổ đĩa cứng : 1F0h, 3F0h...

Card mở rộng của chúng ta cần giải mã vào các địa chỉ khác với các địa chỉ của các thiết bị ngoại vi (Nếu giải mã trùng vào sẽ xung đột cổng). Các địa chỉ từ 300h đến 31Fh được dành cho card mở rộng. Việc giải mã được thực hiện qua 10 bộ địa chỉ A0..

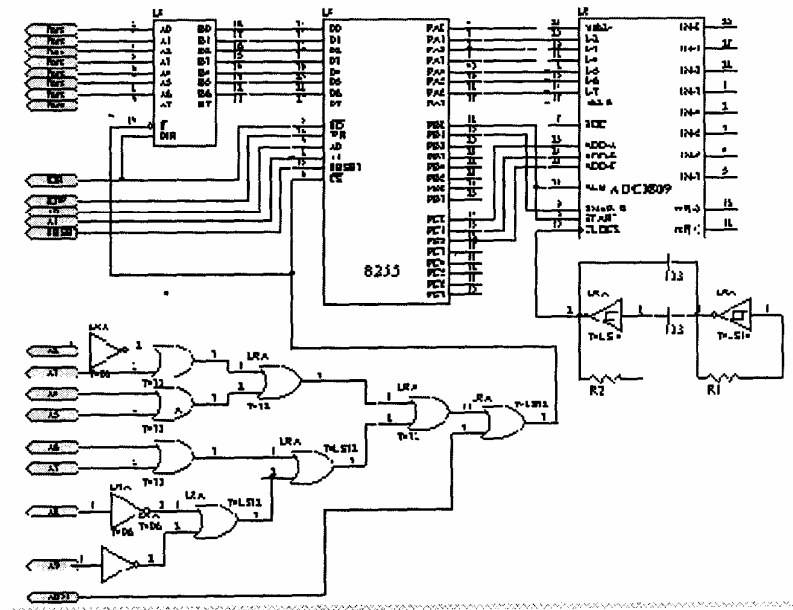
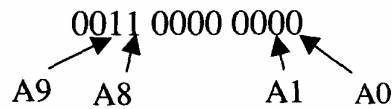
A9 và tín hiệu AEN,

AEN=1 Truy nhập DMA

AEN =0 Truy nhập qua vi xử lý. Khi giải mã cho card ta sử dụng AEN =0.

Các tín hiệu giải mã tùy theo tín hiệu chọn chip có thể là mức thấp hoặc mức cao.

VD: Giải mã cho 1 card tại địa chỉ 300h



Hình 3.28. Sơ đồ card ghép nối qua giao diện ISA

Khi các tín hiệu địa chỉ vào đúng: A8, A9 =1, A2, A3, A4, A5, A6, A7 =0 sẽ cho tín hiệu đầu ra của mạch giải mã (đưa vào CS của 8255) là mức thấp cho phép mạch làm việc,

Giải thích tác dụng của câu lệnh là tín hiệu địa chỉ

b) Ghép nối qua khe cắm PCI

• Cấu hình một hệ thống bus PCI

Bus nội bộ *Peripheral Component Interconnect (PCI)* là một giao diện truyền dữ liệu giữa các cấu thành điều khiển ngoại vi và hệ trung tâm (vi xử lý + bộ nhớ). Bus PCI là một bus dữ liệu và địa chỉ 32, 64 bit. Để có thể hiểu đầy đủ về kiến trúc phần cứng của PCI cần quan tâm đến các chủ đề:

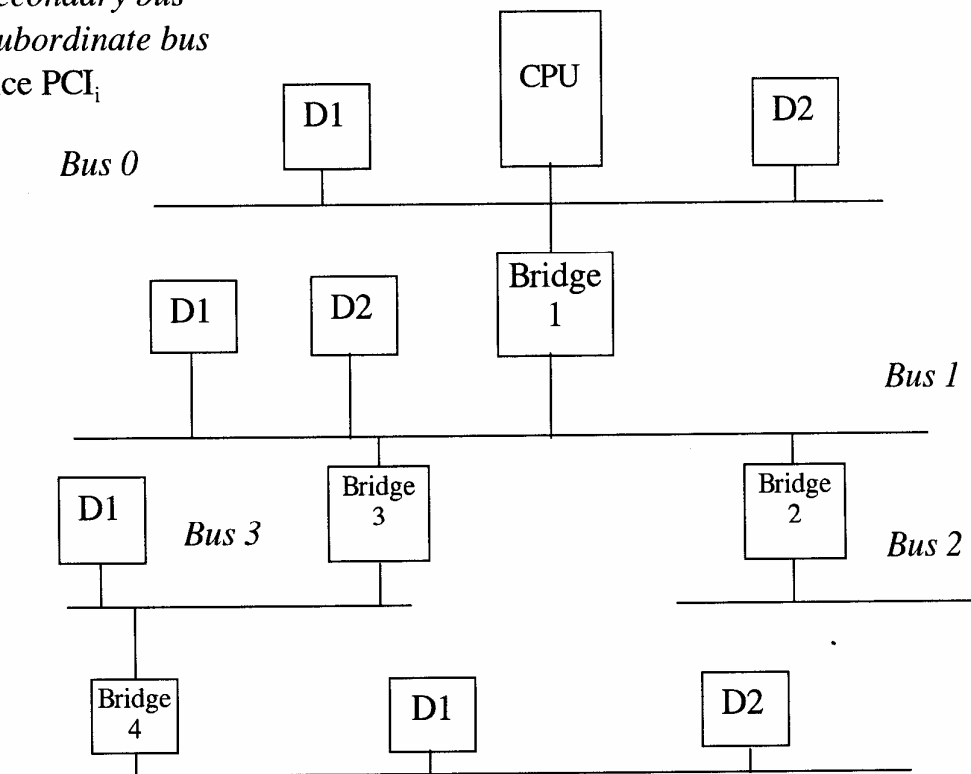
- + Không gian địa chỉ
- + Kích thước dữ liệu
- + Byte thứ tự
- + Các vector ngắt
- + Các thiết bị đa năng

Bus 0: Primary bus

Bus 1: Secondary bus

Bus n: Subordinate bus

Di: Device PCI_i

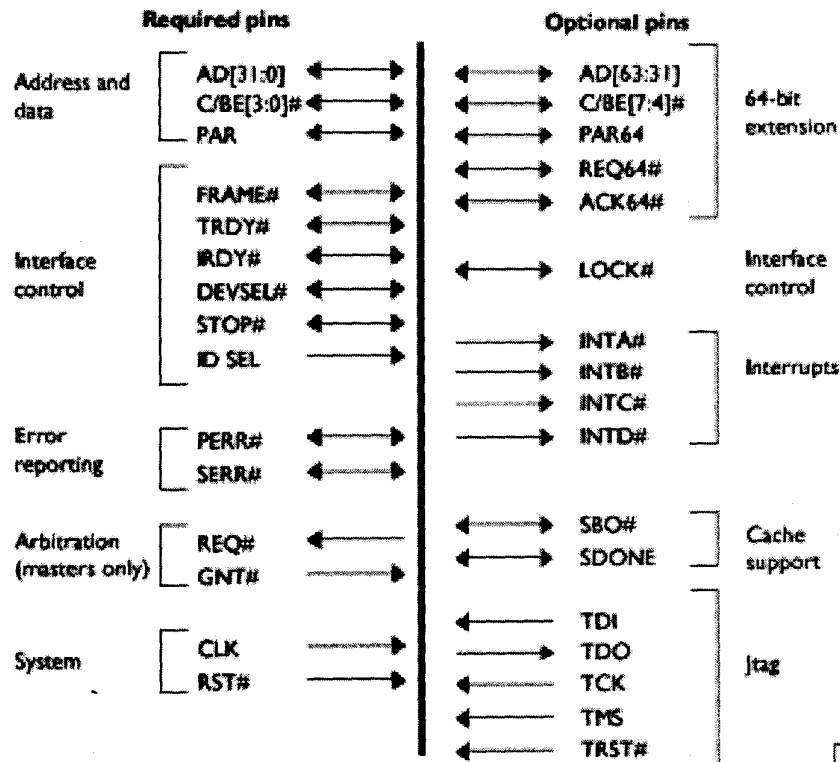


Hình 3.29. Sơ đồ hệ PCI

Tần số làm việc clock của PCI từ 0-33MHZ, tác động theo sườn lên,

Các nhóm tín hiệu qua khe cắm PCI,

Bus PCI được sử dụng trên mainboard máy tính PC có màu trắng, 62 chân trên mỗi mặt. Chức năng của mỗi chân trên khe cắm này có thể tham khảo ([3]. Ngô Diễm Tập, *Đo lường và điều khiển bằng máy tính*).



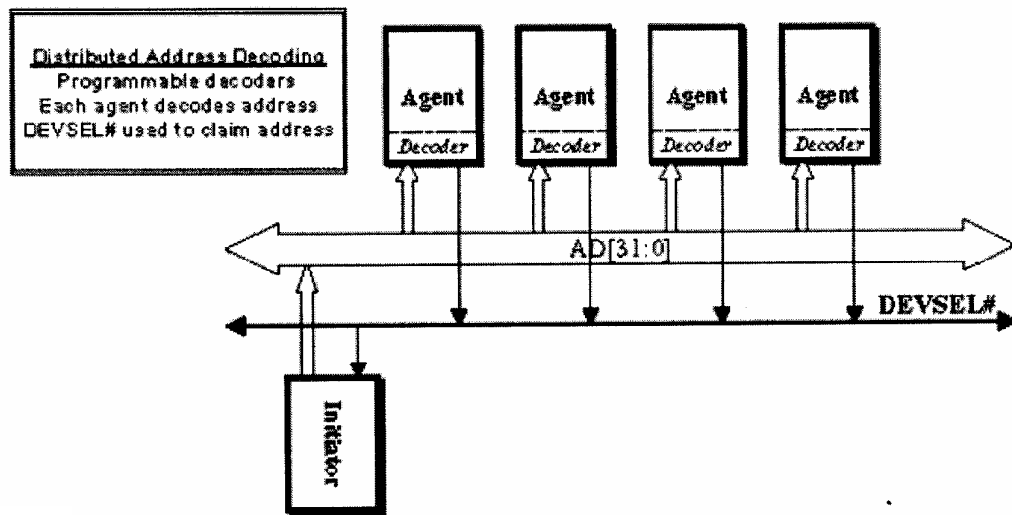
Hình 3.30. Các tín hiệu ở khe cắm PCI

Ý nghĩa và tác dụng của các chân tín hiệu:

- + AD[31 -0] Các chân địa chuẩn liệu
- + C/BE (x) Cho phép lệnh/Byte
- + PAR Parity được sử dụng cho AD0-AD31 và C/BE -3
- + FRAME Sử dụng để báo chu kỳ là một pha địa chỉ hay một pha dữ liệu
- + TRDY Thiết bị đích (Target) sẵn sàng
- + IRDY Thiết bị khởi tạo (Initiator) sẵn sàng
- + DEVSEL Chọn thiết bị
- + STOP Xác nhận bởi target. Yêu cầu Master để dừng chu kỳ đang truyền
- + IDSEL Khởi tạo chọn thiết bị
- + PERR Lỗi Parity
- + SERR Lỗi hệ thống
- + REQ Yêu cầu một di chuyển PCI
- + GNT (Grant) Công nhận sử dụng PCI là hợp lệ
- + RTS Reset

- + CLK Xung nhịp đồng hồ
 - + LOCK Điều khiển khoá tài nguyên trên bus PCI
 - + INT A, B, C, D Các tín hiệu ngắt
- Các tên gọi (terms): Việc trao đổi dữ liệu giữa các thiết bị trên bus PCI theo phương thức truyền đồng bộ. Việc trao đổi được thực hiện giữa các thiết bị khởi tạo- thiết bị đích. Một số tên gọi được sử dụng.
 - + Initiator: Thiết bị khởi tạo hay còn gọi là thiết bị Master. Thiết bị chiếm bus và khởi tạo truyền dữ liệu. Mỗi một Initiator phải có một target,
 - + Target: Thiết bị đích hay còn gọi là thiết bị Slave là đích của truyền dữ liệu (đọc hoặc viết).
 - + Agent: Bất kỳ một Initiator hoặc Target trên bus PCI. Đây là cái chưa được xác nhận là Initiator hoặc Target trong khi thiết lập liên kết.

Giải mã địa chỉ phân bổ trước (Distributed Address Decoding)

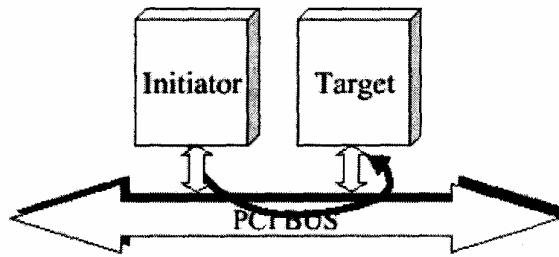


Hình 3.31. Giải mã địa chỉ phân bổ trước

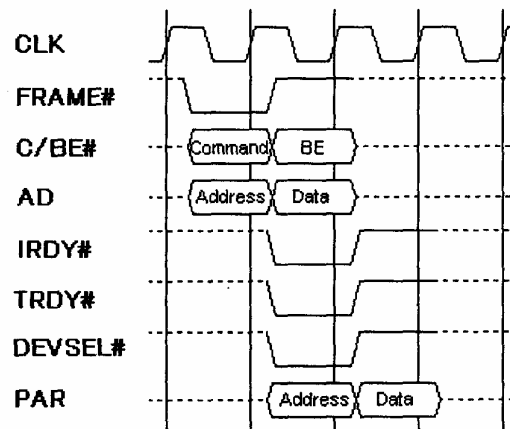
PCI sử dụng giải mã địa chỉ phân bổ trước. Một giao dịch (transaction) bắt đầu qua bus PCI. Mỗi một đích tiềm năng trên bus giải mã các địa chỉ PCI của giao dịch để xác định nó có thuộc về không gian địa chỉ được sở hữu của đích không. Một target có thể được phân cho một không gian địa chỉ lớn hơn các target khác và cũng có thể phản ứng ở nhiều địa chỉ hơn. Mỗi target xác nhận địa chỉ giao dịch của mình bằng tín hiệu DEVSEL#. Bộ giải mã của mỗi Agent lập trình được giải mã địa chỉ DEVSEL# được sử dụng cho xác nhận địa chỉ. Xem trên hình vẽ ta thấy một Initiator khởi động một giao dịch trên bus. Các nhóm tín hiệu lệnh và dữ liệu của nó gửi đi trên bus (đường mũi tên màu trắng to) được các Agent giải mã địa chỉ. Nếu Agent nào đúng địa chỉ nó sẽ gửi DEVSEL# về Initiator (mũi tên màu đen nhỏ) và khi đó nó trở thành Target. Sau

đó là giao dịch truyền nhận dữ liệu giữa Initiator và Target.

Đọc/viết dữ liệu: Hình 3.32 cho ta thấy việc viết từ một Initiator vào Target

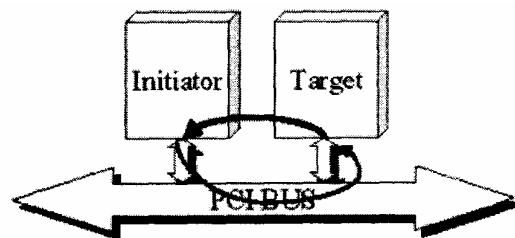


Hình 3.32. Viết từ Initiator



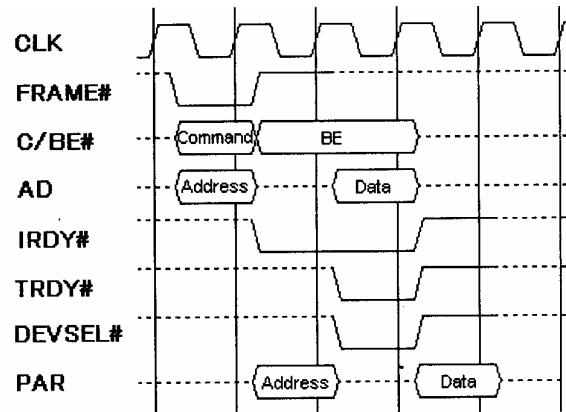
Hình 3.33. Biểu đồ thời gian của quá trình viết

Các thiết bị đích (target) nhận ra sự bắt đầu của máy chu kỳ bus bởi tín hiệu FRAME# được xác nhận ở xung đồng hồ thứ hai. Ở một chu kỳ bus được khởi tạo, lệnh và địa chỉ được dẫn tới mỗi một thiết bị nối với bus và một thiết bị đích tìm thấy chu kỳ thì sẽ gửi tín hiệu xác nhận DEVSEL#. Sự trả lời này của target có thể sau một xung clock (giải mã tốc độ cao - trình speed decode) hoặc sau 2 hay 3 xung clock (giải mã tốc độ trung bình hoặc giải mã tốc độ chậm). Sau khi trả lời DEVSEL#, thiết bị đích xác nhận IRDY# (Viết dữ liệu có hiệu lực) và nhận dữ liệu vào thiết bị đích. Đồng thời thiết bị đích biết rằng việc truyền là chu kỳ cuối cùng bởi tín hiệu FRAME# cắt xác nhận (*deasserted*) và IRDY# với ý nghĩa dữ liệu hiện thời là cuối cùng. Khi dữ liệu cuối cùng đã được lấy vào điều khiển thiết bị đích nó sẽ điều khiển DEVSEL# và TRDY# cắt xác nhận (*deasserted*) sau đó DEVSEL# và TRDY#



Hình 3.34 Quá trình đọc

sẽ được giải phóng ở xung đồng hồ tiếp theo và thiết bị đích kết thúc chu kỳ bus.



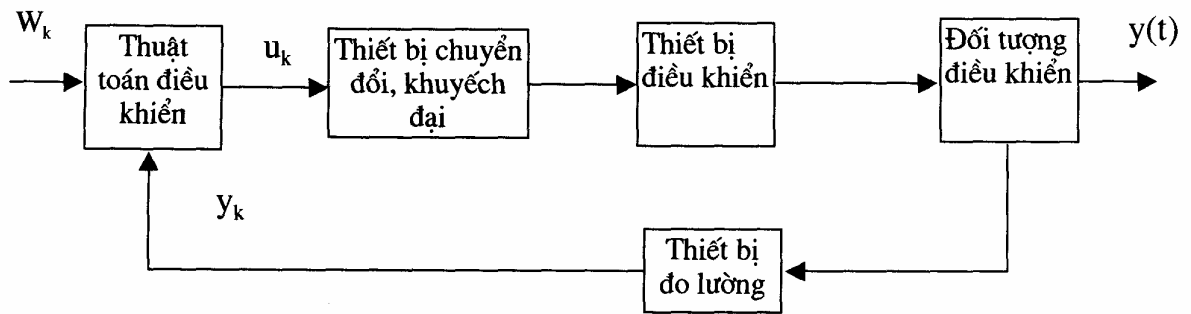
Hình 3.35. Biểu đồ thời gian của quá trình đọc

Như sự dẫn dắt chu kỳ cần có một xung đồng hồ của chu kỳ đệm bởi vì các agent điều khiển bus AD đã thay đổi. Vì vậy chu kỳ đọc nhiều so với chu kỳ viết và một xung đồng hồ là phải cần thiết. Các thiết bị đích (target) nhận ra sự bắt đầu của máy chu kỳ bus bởi tín hiệu FRAME# được xác nhận ở xung đồng hồ thứ hai. Ở một chu kỳ bus được khởi tạo lệnh và địa chỉ được dẫn tới mỗi một thiết bị nối với bus và một thiết bị đích tìm thấy chu kỳ thì sẽ gửi tín hiệu xác nhận DEVSEL#. Trong trường hợp này bởi thiết bị đích trả lời trong hai xung đồng hồ nên việc giải mã là giải mã tốc độ trung bình. Điều đó là cần thiết để đưa dữ liệu vào bus vì cần có thời gian thay đổi của các agent - cái nào thì không quan tâm đến bus (không đúng địa chỉ) cái nào lấy điều khiển trong bus AD (đúng địa chỉ),

Sau hai xung đồng hồ từ lúc bắt đầu chu kỳ bus thiết bị đích đặt dữ liệu đọc vào bus AD và xác nhận TRDY#. Ở xung đồng hồ tiếp theo thiết bị đích kiểm tra IRDY#. Nếu IRDY# đã được xác nhận (Initiator có thể nhận dữ liệu), việc truyền dữ liệu hoàn thành. Và sau đó tín hiệu FRAME# cắt xác nhận, với ý nghĩa dữ liệu hiện thời là cuối cùng. Thiết bị đích nhận ra nó và kết thúc giao dịch. Sau khi dữ liệu cuối cùng được truyền, thiết bị đích điều khiển DEVSEL# và TRDY# cắt xác nhận và hai tín hiệu này sẽ kết thúc ở xung đồng hồ tiếp theo và giao dịch trên bus chấm dứt.

3.2. Ghép nối máy tính với các thiết bị đo lường và điều khiển

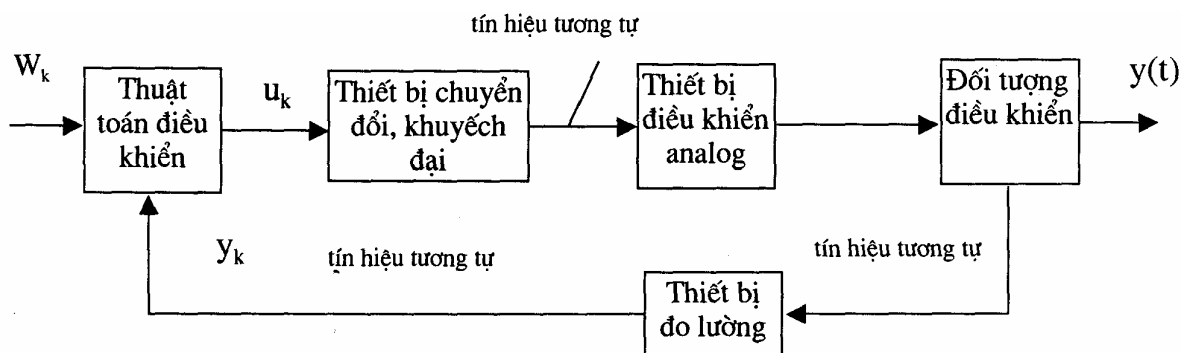
3.2.1. Mô hình tổng quát



Hình 3.36 Mô hình điều khiển tổng quát

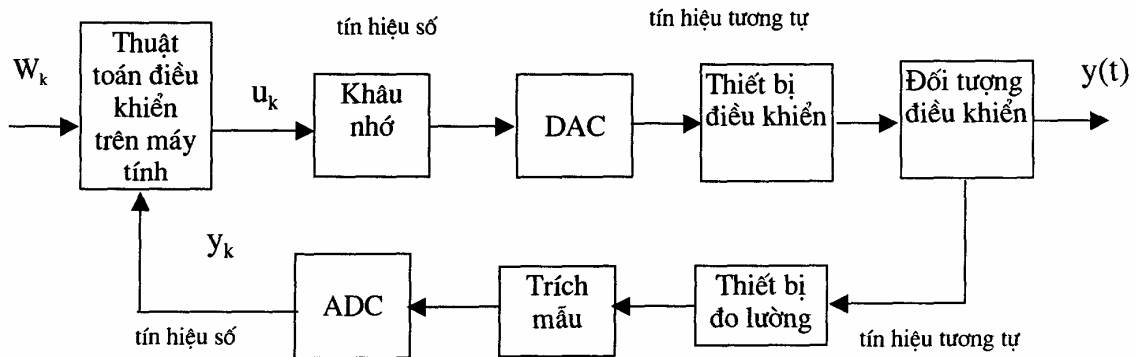
3.2.2. Các phương pháp điều khiển

a) Điều khiển tương tự



Hình 3.37. Mô hình điều khiển tương tự

b) Điều khiển số



Hình 3. Mô hình điều khiển tương tự

W_k : Giá trị đặt

$y(t)$: Giá trị thực

u_k : Điều khiển đầu ra

Sai lệch điều chỉnh $e_k = w_k - y(t)$

CÂU HỎI VÀ BÀI TẬP

1. Xem các vùng địa chỉ thiết bị ngoại vi trên máy tính mình đang sử dụng qua Program/Accessories/System/Tools/Windows Information,
2. Viết chương trình truyền tin nối tiếp giữa hai máy tính bằng VB sử dụng đối tượng MS COM.
3. Viết chương trình điều khiển ra ngoài cổng máy in song song chạy trên các loại windows làm nhấp nháy một đèn LED dùng inoutport.dll
4. Tìm trên mạng Internet các vi xử lý USB.
5. Tìm hiểu chip biến đổi USB/RS232, RS232/RS485.
6. Tìm trong các ngôn ngữ lập trình (VC, VB, Delphi, C#) các hàm truy nhập cổng máy tính PC.
7. Xây dựng mạch ghép ADC 7109, 574, 0816,... với cổng máy in song song, nối tiếp.
8. Lắp ráp sơ đồ DAC 0808 với cổng máy in song song và điều khiển một bóng đèn 12V công suất nhỏ (5, 12, 21 W) sáng tối dần bằng chương trình máy tính.
9. Xây dựng mạch ghép một LCD 1 đường với cổng máy in song song và viết chương trình hiển thị ký tự trên LCD.
10. Xây dựng một ma trận bằng các LED (rời hoặc khối) ghép với cổng máy in song song và hiển thị ký tự qua ma trận này bằng chương trình máy tính theo phương pháp quét động.
11. Xây dựng một mô hình điều khiển motor 1 chiều kiểu cầu H qua cổng máy in song song. Viết chương trình điều khiển.
12. Xây dựng hệ thống đo U, I, f của dòng điện xoay chiều bằng máy tính.
13. Xây dựng hệ thống vẽ đường đặc tuyến của linh kiện bán dẫn bằng máy tính
14. Xây dựng hệ thống ghép nối máy tính qua cổng song song hoặc nối tiếp thu thập dữ liệu tự động gồm ADC, bộ ghép nối, chương trình lưu cơ sở dữ liệu chạy trên các loại Windows.
15. Tự viết chương trình thu ảnh lưu vào máy tính qua Webcam,
16. Tự viết chương trình thu âm thanh lưu vào máy tính PC,

Chương IV

Ghép nối máy tính - máy tính

4.1. Ghép nối đơn giản qua cổng song song

Theo sơ đồ các chân nối với nhau như sau:

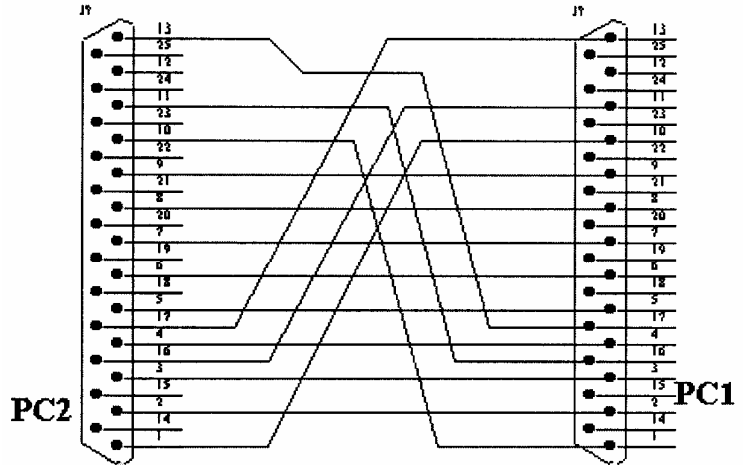
+ Từ D0 đến D7 các chân của mỗi cổng PC1 và PC2 tương ứng nối với nhau

+ ACK - STB

+ INIT - BUSY

+ SLCTIN-SLCT

Như vậy dữ liệu từ một máy (giả sử PC1) chuyển sang máy kia (PC2) qua thanh ghi 378h.

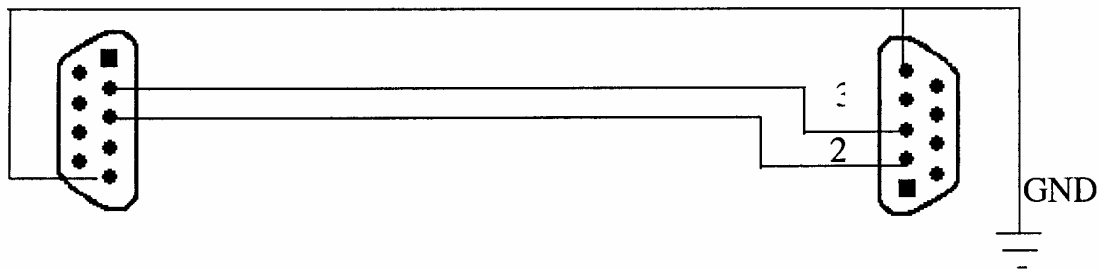


Hình 4.1 Ghép nối đơn giản qua cổng LPT

Chương trình thu bên PC2 phải thiết lập bit D5 của thanh ghi 37Ah của máy này lên 1 để 378h của máy này là cổng vào. Sau khi một byte được truyền, qua STB máy PC2 sẽ báo cho PC1 biết có thể nhận tiếp dữ liệu. PC1 lại gửi tiếp... Truyền/nhận qua cổng song song chỉ có thể thực hiện trong khoảng cách vài mét.

4.2. Ghép nối đơn giản qua cổng nối tiếp

Để thực hiện truyền dữ liệu giữa hai máy tính PC có thể thực hiện kết nối qua cổng truyền tin nối tiếp theo sơ đồ hình 4.2.



Hình 4.2. Ghép nối đơn giản qua cổng nối tiếp

4.3. Vòng dòng điện

Trong truyền tin nối tiếp, để có thể thực hiện truyền với các khoảng cách xa có thể sử dụng các giao diện RS-422, RS-485 hoặc sử dụng giao diện RS-232 kết hợp với vòng dòng điện.

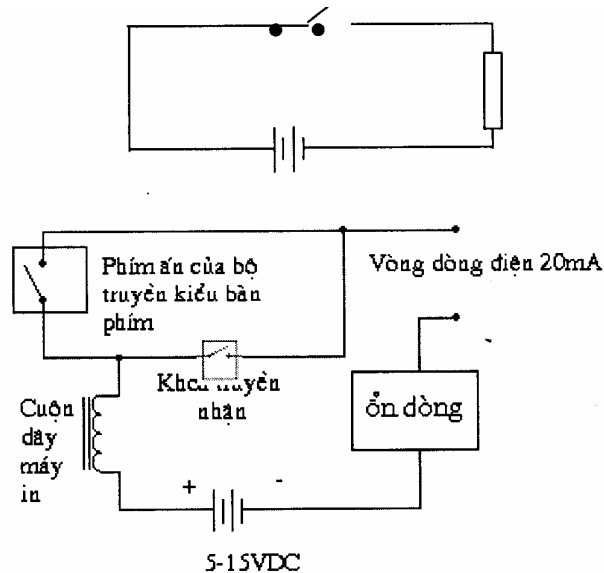
+ Vòng dòng điện 20mA sử dụng tín hiệu dòng điện để mang dữ liệu còn RS-232 dùng các mức điện áp

+ Vòng dòng điện 20mA ra đời trước RS - 232, mô tả trạng thái logic qua tác động cho hoặc không cho một dòng điện có cường độ 20mA đi qua mạch.

- Khái niệm về dòng dòng điện

Logic "0": $\leq 2\text{mA}$

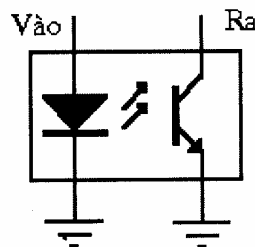
Logic "1": 20mA



Hình 4.3. Sơ đồ vòng dòng điện

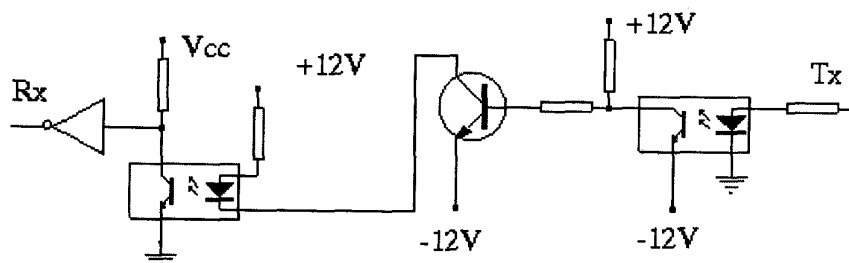
Ghép nối quang

Hoạt động của phần tử ghép quang: Khi có dòng điện chạy qua một, một sẽ phát quang, làm cho tranzito quang dẫn điện. Như vậy tín hiệu đã truyền được từ bên "vào" đến bên "ra" mặc dù hai bên không nối trực tiếp. Nếu bên "vào" có điện áp cao thì cũng không truyền điện áp cao này sang bên "ra" mà chỉ có tín hiệu được truyền qua.



Hình 4.4. Phần tử ghép quang

Một sơ đồ ghép nối quang thu - phát như sau



Hình 4.5. Ghép nối thu phát cách ly

4.4. Ghép nối qua mạng LAN

Các máy tính ghép nối với nhau thành mạng máy tính. Với loại mạng cục bộ có 3 kiểu topology:

- Hình sao
- Hình vòng
- Kiểu bus

Mỗi máy tính cần có một cảm mạng để ghép với đường dây hoặc thiết bị mạng. Các thiết bị mạng thông dụng có thể xem trang 11, 12.

CÂU HỎI VÀ BÀI TẬP

1. Xây dựng mạch ghép nối nối tiếp giữa hai máy tính ghép nối RS485.
2. Xây dựng mạch ghép nối nối tiếp giữa hai máy tính ghép nối RS232 có dùng vòng dòng điện, ghép quang.
3. Viết chương trình để từ một máy tính ghép với mạng LAN hoặc Internet thu thập dữ liệu/ điều khiển từ/ tới một hoặc nhiều máy tính ghép với mạng sử dụng Windows socket.

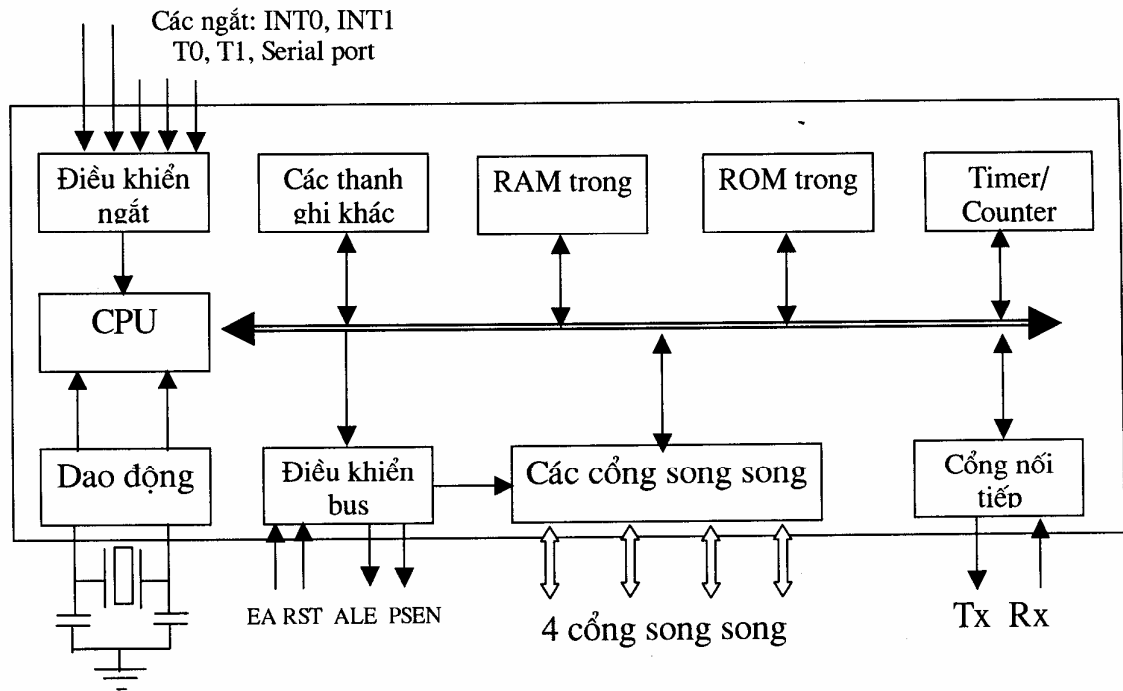
Chương V
Ghép nối máy tính - hệ vi xử lý

5.1. Họ vi điều khiển 8x51/52

5.1.1. Tổng quát về vi điều khiển 8x51/52

8x51/52 là họ vi điều khiển có các đặc tính kỹ thuật cơ bản như sau:

- Là vi điều khiển 8 bit. Có 4kB/ 8kB ROM trong 128/1256 byte RAM trong.
- Khả năng địa chỉ hoá:
 - + 64K bộ nhớ chương trình
 - + 64K bộ nhớ dữ liệu
- Có 128 bytes nhớ RAM trong
- Có 2 Time/Counters
- Cổng nối tiếp, 4 cổng vào ra song song



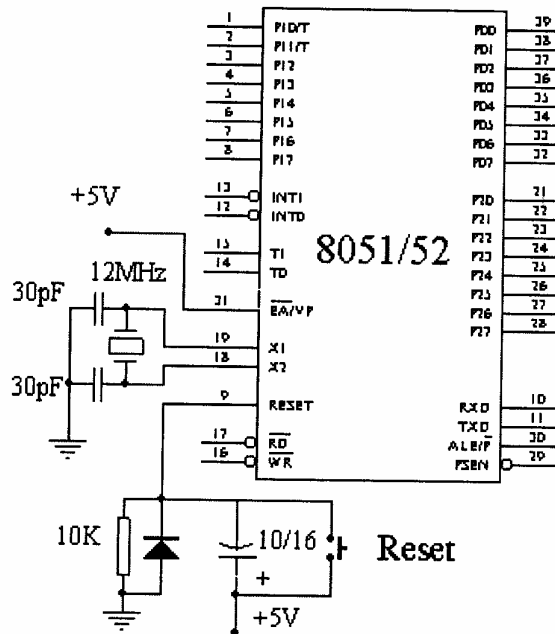
Hình 5.1. Sơ đồ khối vi điều khiển 8x51

- Có bộ điều khiển ngắt logic với 5 nguồn ngắt (8x51) hoặc 6 nguồn ngắt (8x52) - 22 thanh ghi có chức năng đặc biệt SFR (Special function registers).

8051 có thể đánh địa chỉ 64K bộ nhớ dữ liệu ngoài và 64K bộ nhớ chương trình ngoài. 8051 có 2 tín hiệu đọc phân biệt: RD# và PSEN#:

- RD#: Được kích hoạt khi byte được đọc từ bộ nhớ dữ liệu bên ngoài.

-PSEN#: Được kích hoạt khi byte được đọc từ bộ nhớ chương trình bên ngoài



Hình 5.2. Sơ đồ vi điều khiển 8x51/52 tối thiểu có thể hoạt động

5.1.2. Truyền tin nối tiếp trong vi điều khiển 8x51/52

Vi điều khiển 8x51/52 có cổng nối tiếp nằm trên chip. Chức năng quan trọng của cổng nối tiếp là biến đổi dữ liệu từ song song thành nối tiếp để đẩy lên đường truyền và biến đổi dữ liệu vào từ nối tiếp thành song song.

Việc truy nhập phần cứng của cổng nối tiếp thông qua các chân TXD và RXD của 8x52 và đó cũng là 2 bit của Port 3:

P3.1 (Tx/D) là chân 11

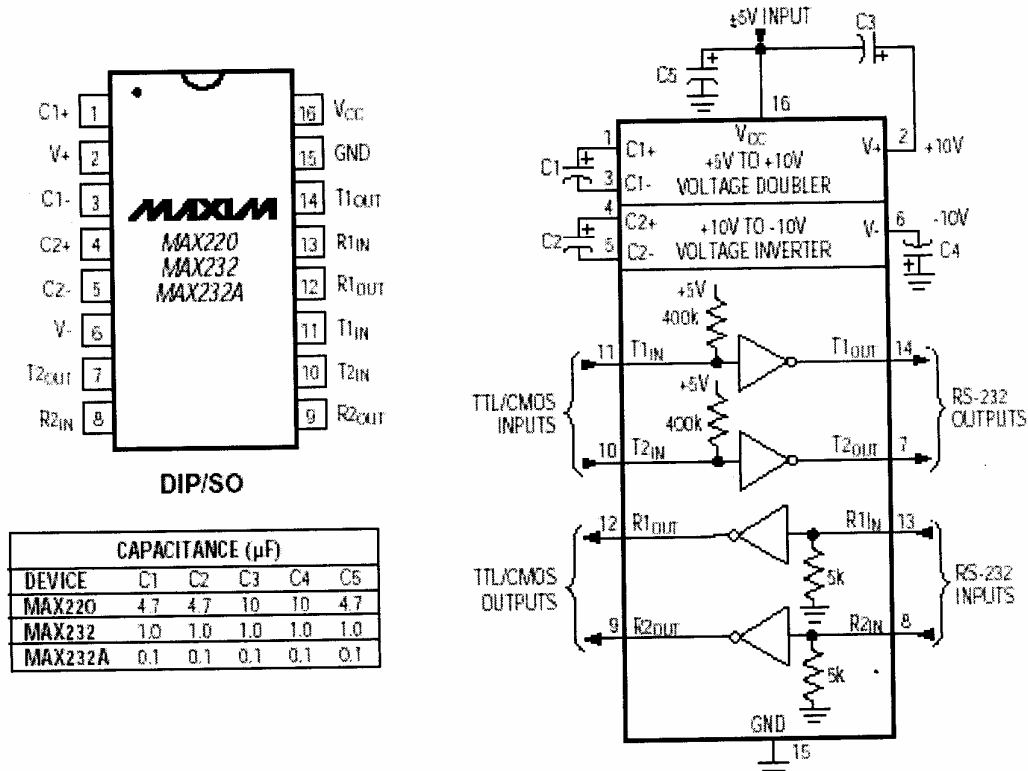
P3.0 (Rx/D) là chân 10

Cổng nối tiếp của 8x52 có thể truyền 2 chiều đồng thời (full duplex) và ký tự có thể được nhận và lưu trữ trong bộ đệm trong khi ký tự thứ 2 đã được nhận và nếu CPU đọc ký tự thứ nhất trước khi ký tự thứ hai được nhận thì dữ liệu không bị mất. Có 2 thanh ghi chức năng đặc biệt (Special Function Register) để phần mềm qua đó truy nhập cổng nối tiếp là SBUF và SCON. SBUF (Serial port buffer) có địa chỉ 99h được xem như 2 buffer. Khi ghi dữ liệu vào SBUF là truyền dữ liệu còn khi đọc dữ liệu từ SBUF là nhận dữ liệu từ đường truyền SCON (Serial port Control register) có địa chỉ 98h là thanh ghi có thể đánh địa chỉ theo từng bit bao gồm bit trạng thái và bộ điều khiển. Bit điều khiển xác lập chế độ điều khiển cho cổng nối tiếp và bit trạng thái cho biết ký tự được truyền hay là được nhận Bit trạng thái được kiểm tra bằng phần mềm hoặc lập trình để gây ra ngắt.

Chương trình truyền tin nối tiếp với vi điều khiển 8x51/52 có thể viết theo kiểu

polling hoặc theo kiểu ngắt. Việc chọn loại nào là tùy theo yêu cầu của công việc cụ thể. Với các ứng dụng đơn giản, ít dữ liệu thì chương trình kiểu polling có thể đáp ứng được. Với các ứng dụng yêu cầu vào ra phức tạp, dữ liệu nhiều, nhanh nên sử dụng phương pháp truyền tin dùng ngắt.

Khi sử dụng truyền tin nối tiếp với vi điều khiển, để tăng cường độ lớn của tín hiệu truyền tin thường sử dụng vi mạch khuếch đại truyền tin nối tiếp MAX232. Sơ đồ sử dụng như hình dưới:

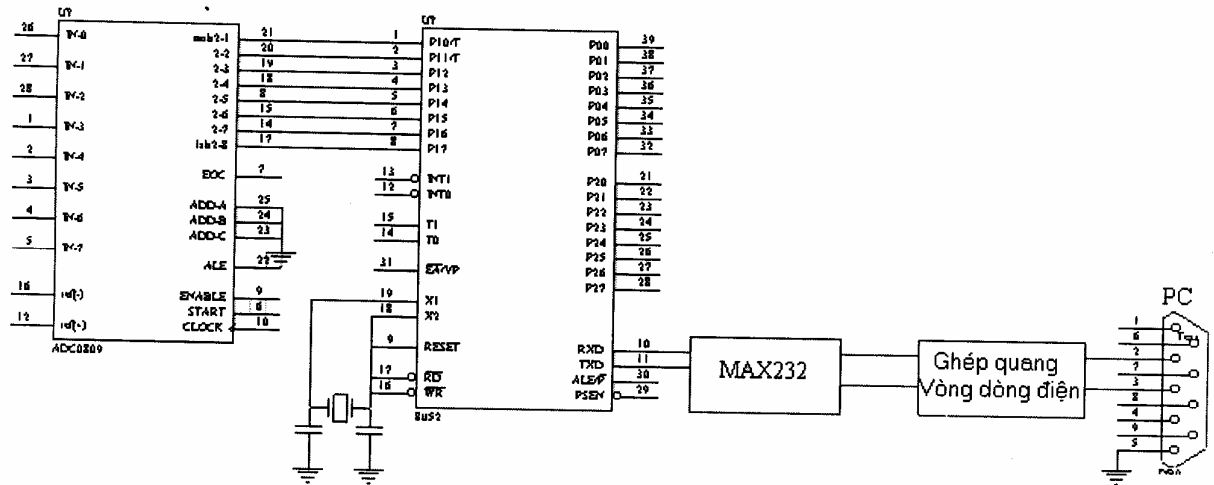


Hình 5.3. Vi mạch khuếch đại truyền tin nối tiếp MAX232

5.2. Ghép nối hệ vi xử lý - máy tính PC

Phần cứng: Ghép nối điển hình giữa một máy tính và vi điều khiển 8x51/52 là thông qua cổng truyền tin nối tiếp, giao diện RS-232,

Để tăng khoảng cách ghép nối có thể phối hợp ghép vòng dòng điện hoặc dùng bộ chuyển đổi RS232/RS-485. Cả hai cách này đều cho phép tăng khoảng cách truyền tới 1000- 1200m,



Hình 5.4. Ghép nối vi điều khiển 8x51/52 với máy tính PC

Phần mềm:

+ Chương trình trên vi điều khiển: Được viết bằng hợp ngữ hoặc bằng C sau khi dịch rồi được nạp vào bộ nhớ trong của vi điều khiển (nếu chương trình lớn có thể phải chứa trong bộ nhớ ngoài), thực hiện các chức năng thu dữ liệu truyền về máy tính host, xử lý theo lệnh của host...

+ Chương trình trên host: Viết bằng các ngôn ngữ bậc cao thực hiện thu dữ liệu từ hệ vi xử lý gửi về, cập nhật dữ liệu, truyền tới các máy tính khác, điều khiển thiết bị chấp hành, gởi lệnh tới vi xử lý... Các chương trình hiện nay chạy trên các hệ điều hành Windows 2000, XP truy xuất các cổng phải thông qua các hàm của các ngôn ngữ bậc cao hoặc các hàm API của Windows. Các chức năng được viết dưới dạng các luồng (*thread*).

5.3. Ghép nối máy tính với chuột và bàn phím

Bàn phím và con chuột là những thiết bị quen thuộc ghép nối với máy tính PC. Chúng dùng để nhập dữ liệu bằng tay từ người sử dụng, được dùng để tác động vào các biểu tượng hoặc mệnh đề điều khiển máy tính. Trong bàn phím và con chuột đều có các vi điều khiển. Hệ vi xử lý xây dựng trên các vi điều khiển này về một số thiết bị điện, điện tử khác trong quá trình làm việc truyền tin với hệ trung tâm (CPU và bộ nhớ trung tâm) theo phương pháp truyền tin nối tiếp đồng bộ. Trước đây con chuột thường được nối qua cổng truyền tin nối tiếp (COM1 hoặc COM2) còn ngày nay trên máy tính PC cả bàn phím và con chuột đều là kiểu PS/2.

The PS/2 Mouse/keyboard Protocol

Giao diện vật lý:

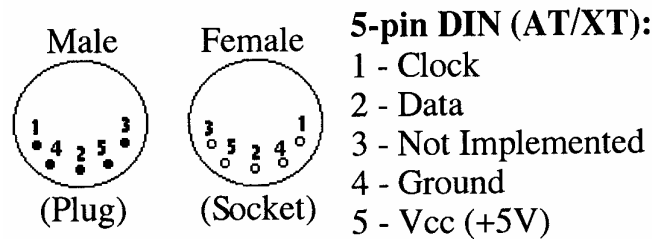
PS/2 - "Personal System/2" được nhắc đến cuối những năm 80 bởi IBM. Đầu nối

cổng PS/2 có hai kiểu: 5 chân DIN hoặc 6 chân mini-DIN. Cả hai kiểu đầu nối đều giống nhau về mặt điện, chúng chỉ khác nhau về các chân. Điều này có nghĩa là hai kiểu đầu nối có thể thay đổi dễ dàng cho nhau bằng các đầu nối chuyển đổi.

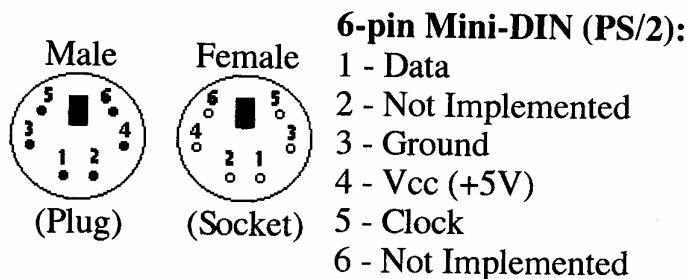
Bàn phím PC sử dụng một trong hai kiểu đầu nối 6-chân mini-DIN hoặc 5-chân DIN connector. Nếu chúng ta có bàn phím 6-chân mini-DIN mà máy tính lại là 5-chân DIN (hoặc ngược lại) thì cần có sự đầu nối chuyển đổi như trên (hoặc có thể cắt dây ra nối lại chuyển đổi với nhau). Bàn phím với 6-chân mini-DIN thường được gọi là bàn phím "PS/2", trong khi loại 5-chân DIN thường được gọi là các thiết bị "AT" (các bàn phím "XT" cũng thường sử dụng 5-chân DIN, nhưng thuộc loại cũ và hầu như không được làm trong những năm gần đây nữa). Các bàn phím hiện nay là loại PS/2, AT, hoặc USB.

Cáp nối bàn phím hoặc con chuột với máy tính thường dài khoảng 6 feet và có 4 đến 6 dây bọc nhựa được bọc bởi lớp giấy kim loại mỏng. Nếu chúng ta muốn có dây dài hơn chúng ta phải mua dây cáp PS/2 mở rộng có bán tại các cửa hàng vật liệu điện. Chúng ta có thể nối nhiều đoạn dây cáp mở rộng lại hoặc mua luôn độ dài dây cần thiết. Nếu nối nhiều mối nối có thể làm truyền tin kém.

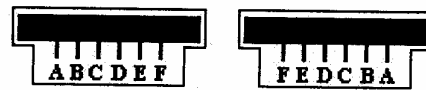
Các chân của mỗi kiểu đầu nối như Hình 5.5 a, b:



Hình 5.5a



Hình 5.5 a, b. Đầu nối chuột/bàn phím PS/2 và kiểu AT/XT

**6-pin SDL:**

A - Not
 Implemented
 B - Data
 C - Ground
 D - Clock
 E - Vcc (+5V)
 F - Not
 Implemented

Hình 5.6. Đầu nối chuột/bàn phím PS/2 6 chân

Giao diện điện:

Chú ý: trong phần này thuật ngữ "host" dùng để chỉ thiết bị mà chuột hoặc bàn phím nối đến ví như máy tính PC còn thuật ngữ "device" là để chỉ bàn phím hoặc con chuột.

Bàn phím hoặc chuột được cung cấp qua nguồn đơn (Vcc/Ground). Dòng tiêu thụ không lớn hơn 275 ma từ host và phải tránh được sự tăng điện đột ngột. Những sự tăng đột ngột về điện được gây ra bởi nguyên nhân "hot-plugging" (tức là rút/cắm bàn phím hoặc con chuột khi máy tính đang chạy). Những mainboard cũ còn có cả cầu chì bảo vệ cổng bàn phím và con chuột. Khi những cầu chì này bị nổ đã không mang lại lợi ích gì cho người tiêu dùng. Kiểu mới nhất các mainboard sử dụng cầu chì tự nối lại "Poly" là một hướng giải quyết vấn đề.

Tóm tắt: Đặc tính nguồn cung cấp

Vcc = +4.5V to +5.5V.

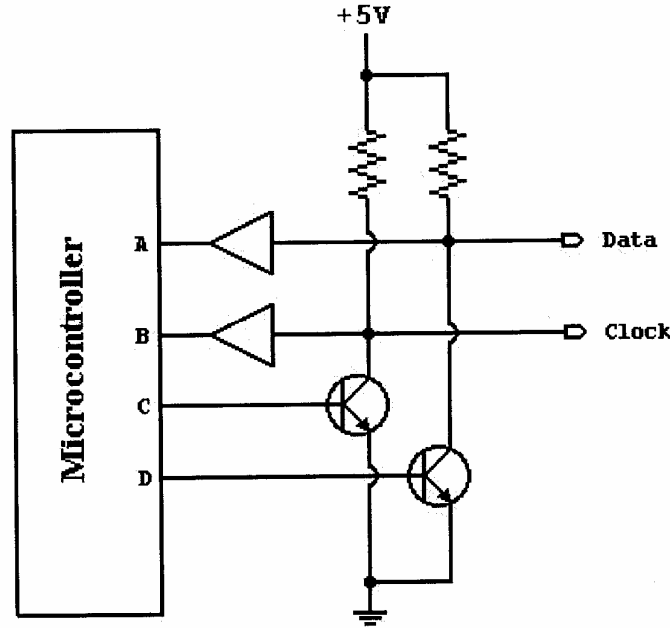
Dòng lớn nhất = 275 mA.

Các đường dây Data và Clock đều là hở collector có điện trở kéo lên. Một giao diện "open-collector" có hai trạng thái có thể: trở kháng thấp hoặc trở kháng cao. Trong trạng thái "low", một transistor sẽ kéo mạch (nhe) xuống mức nối đất. Trong trạng thái hình impedance" giao diện đóng vai trò như một mạch điện hở và không điều khiển mạch cao hay thấp. Và lại một điện trở kéo lên "pullup" được nối giữa bus và Vcc như vậy bus được kéo lên cao nếu không có thiết bị nào trên bus tác động kéo nó xuống thấp. Giá trị chính xác của điện trở này là không quan trọng (1~10 kOhms); các điện trở lớn cho kết quả nguồn tiêu thụ nhỏ và các điện trở nhỏ hơn cho kết quả nối lên nhanh hơn. Một giao diện mạch hở collector được xem dưới đây:

Hình 5.7: Giao diện mạch open-collector chung. Dữ liệu và Clock được đọc vào các chân A và B của vi điều khiển. Cả các dây có điện áp +5V, nhưng có thể kéo tới đất bởi xác nhận mức "1" trên C và D.

Và kết quả là, Data tương ứng giá trị D, bị đảo ngược., và Clock tương ứng giá trị

C, bị đảo ngược.



Hình 5.7. Giao diện mạch bàn phím

Truyền tin: Mô tả chung

Chuột và bàn phím PS/2 thực hiện một giao thức truyền tin nối tiếp đồng bộ hai chiều. Bus nghỉ ("idle") khi cả hai đường dây là cao (open-collector). Trạng thái này chỉ khi bàn phím hoặc chuột được cho phép bắt đầu truyền dữ liệu; Host có sự điều khiển cuối cùng trên bus và có thể cấm truyền tin ở bất kỳ thời điểm nào bằng cách kéo đường Clock xuống thấp (pulling the Clock nhe low).

Thiết bị luôn phát ra tín hiệu clock. Nếu host muốn gửi dữ liệu, đầu tiên nó phải cấm truyền tin từ thiết bị bằng cách kéo clock low. Host khi đó kéo dây Data xuống thấp (Data low) và giải thoát dây Clock (releases Clock). Đây là trạng thái "Request-to-send" và các tín hiệu thiết bị bắt đầu sinh ra các xung clock.

Tóm tắt: Các trạng thái bus

Data = high, Clock = high: Trạng thái nghỉ - *khe state*.

Data = high, Clock = low: Cấm truyền tin - *Communication Inhibited*.

Data = low, Clock = high: *Host Request-to-send*

Các dữ liệu được truyền một byte kế tiếp nhau và mỗi một byte được truyền trong một khung tin (frame) chứa 11 - 12 bits. Các bit đó là:

- 1 start bit. Bit này luôn bằng 0.
- 8 data bits, bit LSB truyền trước.

- 1 parity bit (parity lẻ).
- 1 stop bit. Bit này luôn bằng 1.
- 1 acknowledge bit (host-to-device communication giấy)

Bit parity được lập (set) nếu có tổng con số chẵn các bit 1 trong các bit dữ liệu và xoá (reset - 0) nếu có tổng con số bit 1 là lẻ trong các bit dữ liệu. Con số các bit 1 trong các bit dữ liệu cộng với bit parity luôn luôn phải thêm vào một con số lẻ (parity lẻ) (*The number of 1's in the data bits plus the parity bit always add up to an odd number (odd parity.)*) Điều này sử dụng để tìm sự hỏng. Bàn phím/con chuột phải kiểm tra (chếch) bit này và nếu không đúng nó sẽ phản ứng như là nó thu một lệnh sai.

Dữ liệu truyền từ thiết bị tới host đọc trong sườn xuống (*falling edge*) của tín hiệu clock signal; dữ liệu truyền từ host tới thiết bị thì đọc trong sườn lên (*rising edge*). Tần số xung clock phải trong giới hạn 10 - 16.7 kHz. Nghĩa là xung clock phải cao trong 30 - 50 micro giây và thấp trong 30 - 50 micro giây. Nếu chúng ta thiết kế một bàn phím, con chuột, hoặc một thiết bị nào đó nối với host chúng ta có thể biến đổi/lấy mẫu dương dữ liệu ở giữa mỗi một xung (*modify/sample the Data near in the middle of each cell*). Tức là 15 - 25 micro giây thích hợp sau clock thay đổi. Mặt khác bàn phím/con chuột luôn sinh ra xung tín hiệu clock, nhưng host luôn có điều khiển sau cùng (ultimate control) qua truyền tin.

Truyền tin: Device-to-Host

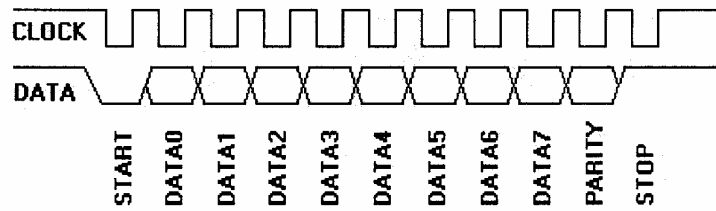
Các đường Data và Clock đều là hở collector (open collector). Một điện trở thì được nối giữa mỗi một dây với +5V, như vậy trạng thái nghỉ của bus là cao. Khi bàn phím hoặc con chuột muốn gửi thông tin, đầu tiên nó kiểm tra (checks) đường Clock chắc chắn là mức cao. Nếu chưa phải, host cấm truyền tin và thiết bị phải đưa vào đệm dữ liệu sẽ truyền cho đến khi host giải thoát Clock (releases Clock). Đường dây Clock phải tiếp tục cao ở ít nhất 50 micro giây trước khi thiết bị có thể bắt đầu truyền dữ liệu của nó.

Như đã đề cập đến trong phần trước, bàn phím và con chuột sử dụng giao thức nối tiếp với khung tin 11 bit. Các bit đó là:

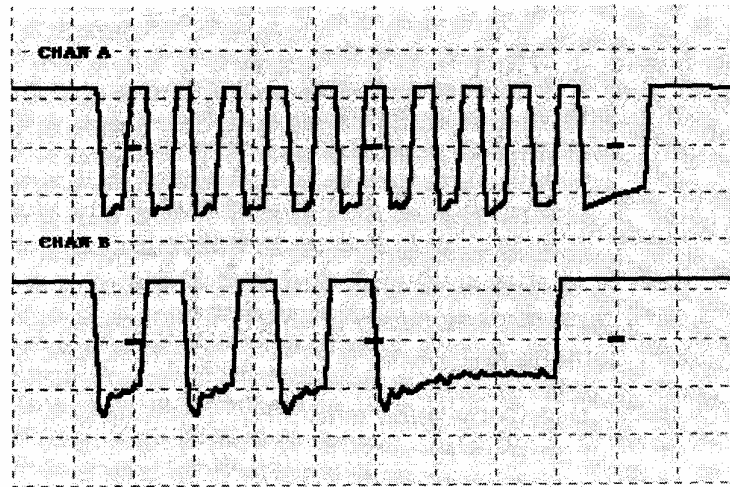
- 1 start bit. Bit này luôn bằng 0.
- 8 data bits, LSB trước.
- 1 parity bit (odd parity).
- 1 stop bit. Bit này luôn bằng 1.

Bàn phím/chuột viết một bit vào đường Data khi Clock là cao, và nó là đọc bởi host khi Clock là thấp. Hình 5.58 và 5.59 minh họa điều này.

Hình 5.58: Truyền tin Device-to-host. Đường Data thay đổi trạng thái khi Clock là cao và dữ liệu ổn định khi Clock là thấp.



Hình 5.8. Truyền tin từ thiết bị tới host



Hình 5.9. Mã quét phím "Q" gửi từ bàn phím tới cổng bàn phím
kênh A là Clock, kênh B là dữ liệu

Tần số clock là 10- 16.7 kHz. Thời gian từ sườn lên của một xung Clock tới một thay đổi trạng thái dữ liệu (Dữ liệu transition) ít nhất phải là 5 micro giây. Thời gian từ thay đổi trạng thái dữ liệu tới sườn xuống của một xung clock ít nhất phải 5 micro giây và không lớn hơn 25 micro giây.

Host có thể cấm truyền tin ở bất cứ lúc nào bằng cách kéo đường Clock thấp (low) ít nhất 100 microseconds. Nếu một sự truyền bị cấm trước xung clock thứ 11 thiết bị (device) phải hỏng việc truyền hiện tại và phải sẵn sàng truyền lại (sự truyền) hiện tại "chunk" của dữ liệu khi host giải thoát Clock (releases Clock). Một sự "chunk" của dữ liệu có thể là a ma ke co de, break co de, device ID, mouse movement packet, etc. Ví dụ nếu một bàn phím bị ngắt khi đang gửi byte thứ hai của một break co de hai byte, nó sẽ cần phải truyền lại cả các byte của break code đó.

Nếu host kéo clock thấp trước khi biến đổi clock high-to-low đầu tiên, hoặc sau sườn xuống của xung clock cuối cùng, bàn phím/con chuột không cần phải truyền lại bất kỳ dữ liệu nào. Ngoài ra nếu dữ liệu mới đã được tạo lập thì cần thiết phải truyền, nó sẽ phải giữ ở đệm cho đến khi host giải thoát Clock (releases Clock). Bàn phím có 1 vùng đệm 16 byte cho mục đích này. Nếu trên bàn phím bấm thêm vào khi đã đầy vùng đệm thì phím bấm vào sẽ bị bỏ qua cho đến khi vùng đệm còn chỗ trống. Các con chuột chỉ lưu trữ các gói tin di chuyển hiện thời phổ biến cho việc truyền.

Truyền tin Host-to-Device:

Gói tin (packet) được gửi đi khác một ít trong truyền tin host-to-device...

Trước hết, thiết bị PS/2 sinh ra tín hiệu clock. Nếu host muốn gửi dữ liệu, trước hết nó phải dây Clock và Data vào trạng thái "Request-to-send" như sau:

Cấm truyền tin bởi việc kéo dây Clock xuống thấp ít nhất 100 micro giây.

Gắn vào "Request-to-send" bằng cách kéo dây Data xuống thấp, sau đó giải thoát Clock (release Clock).

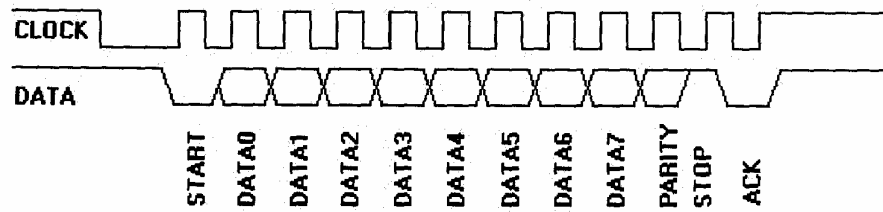
Thiết bị sẽ kiểm tra trạng thái này trong một thời khoảng không vượt quá 1 0 milli giây. Khi thiết bị tìm được trạng thái này, nó sẽ bắt đầu phát ra các tín hiệu Clock và clock trong 8 bit dữ liệu và 1 bit stop. Host thay đổi dây Data chỉ khi dây Clock là mức thấp, và dữ liệu được đọc bởi thiết bị khi dây Clock ở mức cao. Việc này ngược với điều xuất hiện trong truyền tin device-to-host.

Sau khi bit stop đã được thu, thiết bị sẽ xác nhận (acknowledge) byte thu bởi mang dây Data xuống thấp và phát ra một xung clock cuối. Nếu host không giải thoát dây Data sau xung clock thứ 11, thiết bị sẽ tiếp tục phát các xung clock cho đến khi dây Data được giải thoát' (thiết bị khi đó sẽ phát ra một sự hỏng). Host có thể bỏ việc truyền (abort transmission) ở thời điểm trước xung clock thứ 11 (acknowledge) bằng cách giữ dây Clock ở mức thấp ít nhất 100 micro giây. Đây là các bước host gửi dữ liệu đến một thiết bị PS/2:

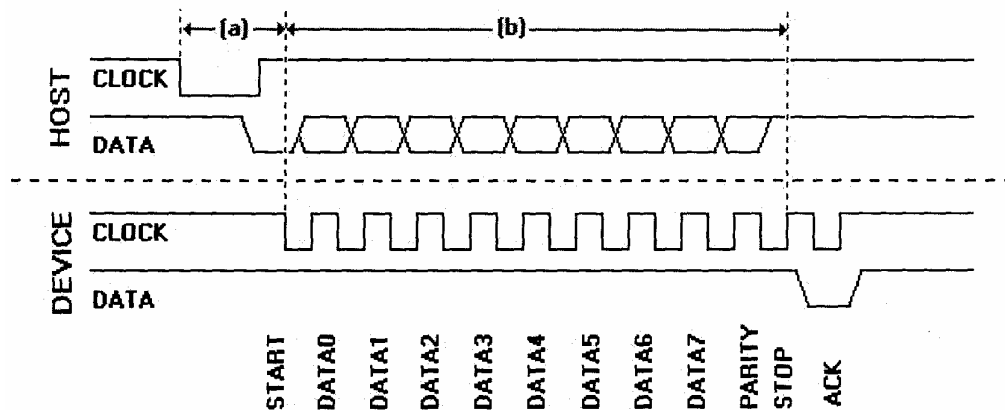
- 1) Làm cho dây Clock ở mức thấp ít nhất 100 micro giây.
- 2) Làm cho dây Data ở mức thấp.
- 3) Giải thoát dây Clock (Release the Clock line).
- 4) Đợi thiết bị mang dây Clock đến mức thấp.
- 5) Set/reset dây Data để truyền bit dữ liệu đầu tiên
- 6) Đợi thiết bị mang dây Clock đến mức cao.
- 7) Đợi thiết bị mang dây Clock đến mức thấp.
- 8) Liên tục làm các bước 5-7 với 7 bit dữ liệu khác và bit parity
- 9) Release dây Data..
- 10) Đợi thiết bị mang dây Data đến mức thấp.
- 11) Đợi thiết bị mang dây Clock đến mức thấp.
- 12) Đợi thiết bị giải thoát (release) dây Data và dây Clock

Hình 5.10 cho xem biểu đồ này và Hình 5.11 tách rời thời gian cho xem các tín hiệu được sinh ra bởi host, và bởi thiết bị PS/2. Chú ý sự thay đổi trong thời gian với bit "ack" sự truyền dữ liệu xuất hiện khi đường dây Clock là mức cao (đúng hơn khi

nó là thấp như trong trường hợp với 11 bit khác).



Hình 5.10 Truyền từ host đến thiết bị



Hình 5.11. Chi tiết hoá truyền tin host đến thiết bị

Tham khảo Hình 5.11 đây có hai lượng thời gian host đợi (host looks for). (a) is the time it takes the device to begin generating clock pulses after the host initially takes the Clock nhe low, which must be no greater than 15 ms. (b) is the thuê ít takes for the packet to bê sau, which must be no greater than 2ms. Nếu mỗi một khoảng thời gian này không phù hợp, host sẽ phát ra một sự hỏng. Tức khắc sau khi "ack" được nhận, host có thể mang dây Clock xuống thấp cấm truyền tin trong khi nó xử lý dữ liệu. Nếu lệnh gửi từ host yêu cầu sự đáp ứng đáp ứng này phải được nhận ít hơn 20 ms sau khi host giải phóng dây Clock. Nếu điều này không xảy ra, host sinh ra một sự hỏng (enor).

CÂU HỎI VÀ BÀI TẬP

1. Giao diện PS/2 là gì? Tại sao chuột và bàn phím đều ghép chung PS/2 mà không xung đột?
2. Xây dựng hệ thống thu thập dữ liệu: Tín hiệu analog qua ADC đưa vào hệ vi xử lý (8x51 hoặc PSOC) sau đó chuyển về host PC qua RS 485 hoặc RS232 ghép vòng dòng điện.
3. Cũng nhiệm vụ trên và chương trình trên máy tính PC chuyển dữ liệu thu được về một máy server ghép với host PC quan mạng LAN hoặc Intemet.

TÀI LIỆU THAM KHẢO

- [1]. William Stallings, *Computer organization and architecture*, Prentice- Han, Inc, 1996.
- [2]. Adrew S. Tanenbaum, *The Modern Operating System*.
- [3]. Peter H. Anderson, *Use of a PC Printer Portfor Control and Data Acquisition*, Department of Electrical Engineering Morgan State University, 1996.
- [4]. Ngô Diễm Tập, *Đo lường và điều khiển bằng máy tính*, NXBKHKHT, 2001.
- [5]. Văn Thế Minh, *Kỹ thuật vi xử lý*, NXBGD, 1997.
- [6]. Vũ Chân Hưng, *Giáo trình kiến trúc máy vi tính*, NXB GTVT, 2003.
- [7]. Ngô Diễm Tập, *Kỹ thuật ghép nối máy tính*, NXBKHKHT, 1999.
- [8]. Nguyễn Nam Trung, *Cấu trúc máy vi tính và thiết bị ngoại vi*, NXB KHKHT, 2000.
- [9]. Nguyễn Mạnh Giang, *Kỹ thuật ghép nối máy tính*, NXBGD, 1998.
- [10]. Vũ Chân Hưng, *Tập bài giảng "ứng dụng kỹ thuật vi xử lý và máy vi tính trong đo lường và điều khiển"*.
- [11]. Các tài liệu kỹ thuật ADC, DAC, các vi mạch số - tương tự của các hãng.
- [12]. Một số trang web:
 - + <http://w.w.w.msn.com>
 - + <http://w.w.w.ntfs.com>
 - + <http://w.w.w.cs.arizona.edu/>
 - + <http://w.w.w.science.unitn.it/>
 - + <http://w.w.w.howstuffworks.com>
 - + <http://w.w.w.Storage review.com>

MỤC LỤC

Trang

Chương I. Tổng quan về máy tính, thiết bị ngoài của máy tính và các phương pháp trao đổi, biến đổi dữ liệu	2
1.1. Máy tính và các thành phần cấu thành	2
1.1.1. Hệ vi xử lý kinh điển	2
1.1.2. Máy tính PC hiện nay	5
1.1.3. Hệ thống bus	6
1.2. Thiết bị ngoài của máy tính	7
1.2.1. Kiểu cơ - điện tử	7
1.2.2. Kiểu từ - điện tử	7
1.2.3. Kiểu quang - điện tử	8
1.2.4. Thiết bị xử lý tín hiệu	11
1.3. Các chuẩn ghép nối	12
1.3.1. Định nghĩa	12
1.3.2. Các đặc tả cho từng kiểu ghép nối	14
1.4. Các phương pháp vào ra dữ liệu	18
1.4.2. Vào/Ra theo phương pháp hỏi vòng	19
1.4.3. Vào / Ra bằng ngắt	21
1.4.4. Vào / Ra theo phương pháp DMA	25
1.5. Các thiết bị chuyển đổi dữ liệu:	26
1.5.1. Khái niệm - Định lý lấy mẫu của Shannon	26
1.5.2. Chuyển đổi A/D	29
a. A/D xấp xỉ tiệm cận	29
b. A/D tích phân 2 sườn dốc	31
1.5.3. Chuyển đổi D/A	43
Chương II. Cấu trúc chung của 1 modul ghép nối	48
2.1. Mô hình tổng thể	48
2.2. Cấu trúc các khối	49
2.2.1. Khối giải mã địa chỉ - nhiệm vụ, cấu tạo	49
2.2.2. Khối đệm dữ liệu - nhiệm vụ, cấu tạo	51
2.2.3. Khối logic điều khiển thiết bị - nhiệm vụ, cấu tạo	54

2.3. Phần mềm điều khiển thiết bị.....	54
<i>Chương III. Thiết kế các ghép nối máy tính qua các giao diện.....</i>	57
3.1. Ghép nối máy tính qua các giao diện.....	57
3.1.1. Ghép nối qua cổng song song.....	57
a) Các thanh ghi của giao diện vào ra song song.....	57
b) Điều khiển ra ngoài qua cổng song song.....	59
c) Đưa dữ liệu vào máy vi tính qua cổng song song.....	70
3.1.2. Ghép nối qua cổng nối tiếp.....	71
a) Giao diện RS232, 485.....	71
b) Vào ra dữ liệu bằng Polling-truyền tin qua cổng nối tiếp.....	74
c) Vào ra dữ liệu dùng ngắt.....	79
3.1.3. Ghép nối qua cổng USB.....	80
3.1.4. Ghép nối qua các khe cắm ở rộng.....	85
a) Ghép nối qua khe cắm ISA.....	85
b) Ghép nối qua khe cắm PCI.....	86
3.2. Ghép nối máy tính với các thiết bị đo lường và điều khiển.....	91
3.2.1. Mô hình tổng quát.....	91
3.2.2. Các phương pháp điều khiển.....	92
a) Điều khiển tương tự.....	92
<i>Chương IV. Ghép nối máy tính - máy tính.....</i>	94
4.1. Ghép nối đơn giản qua cổng song song.....	94
4.2. Ghép nối đơn giản qua cổng nối tiếp.....	94
4.3. Vòng dòng điện.....	94
4.4. Ghép nối qua mạng LAN.....	96
<i>Chương V. Ghép nối máy tính - hệ vi xử lý.....</i>	97
5.1. Họ vi điều khiển 8x51/52.....	97
5.1.1. Tổng quát về vi điều khiển 8x51/52.....	97
5.1.2. Truyền tin nối tiếp trong vi điều khiển 8x51/52.....	98
5.2. Ghép nối hệ vi xử lý - máy tính PC.....	99
5.3. Ghép nối máy tính với chuột và bàn phím.....	100
TÀI LIỆU THAM KHẢO.....	108