MỤC LỤC

ĐỀ MỤC **TRANG**

BÀI MỞ ĐẦU: TỔNG QUAN VỀ ĐO LƯỜNG..... 2

CHƯƠNG 1: CÁC KHÁI NIỆM VỀ ĐO LƯỜNG VÀ ĐIỀU KHIỂN MÁY TÍNH 4

CHƯƠNG 2: GIAO TIẾP QUA CỔNG SONG SONG (PARALLEL)..... 18

CHƯƠNG 3: GIAO TIẾP QUA CỔNG NỐI TIẾP (SERIAL) 37

TÀI LIỆU THAM KHẢO..... 55

BÀI MỞ ĐẦU: TỔNG QUAN VỀ ĐO LƯỜNG

1. Vai trò của Đo lường và Điều khiển máy tính.

Ngày nay, trên thế giới đang diễn ra quá trình tin học hóa trên nhiều lĩnh vực hoạt động của xã hội loài người. Máy tính nói chung và Đo lường và điều khiển máy tính nói riêng xuất hiện khắp nơi.

Để thực hiện công việc có tích chất nguy hiểm và đòi hỏi độ chính xác cao như điêu khiển xe cộ, máy bay các tín hiệu giao thông hay điều khiển các thiệt bị trong các nhà máy nhiệt điện…Nếu không sử dụng máy tính để điều khiển máy móc thực hiện các công việc đó thì khó mà có thể giải quyết được các bài toán đó. Chính vì vậy ngày nay Đo lường và điều khiển máy tính đóng vai trò quan trọng đời sống và sản xuất hiện nay.

Đo lường giúp chúng ta có thể giải quyết rất nhiều bài toán khó đòi hỏi độ chính xác cao trong các nhà máy sản xuất công nghiệp.

Điều khiển máy tính có thể giúp chúng ta theo dõi, quan sát các hoạt động từ xa như điều khiển và theo dõi các hoạt động dây truyền trong các nhà máy, theo dõi và điều khiển các tín hiệu đèn giao thông trong các thành phố, thị trấn…hay trong các lĩnh vực khác…

Tóm lại Đo lường và điêu khiển máy tính có vai trò không thể thiếu trong đời sống cũng như trong hoạt động sản xuất ngáy nay.

2. Những ứng dụng điều khiển máy tính.

Điều khiển bằng máy tính có ứng dụng rất lớn trong nhiều lĩnh vực nhất là trong nền công nghiệp hiện đại, và nó là một trong những ngành trọng yếu ngày nay.

Các hệ thống máy tính giúp con người trong việc tự động hóa các dây chuyển sản xuất, nhất là trong những môi trường nguy hiểm cho con người ví dụ như: sản xuất ôtô, máy vi tính, đồ điện tử và các thiết bị đòi hỏi độ chính xác cao khác.

Điều khiển máy tính có thể tự thu thập thông tin về giao thông, và dựa vào các thuật toán có sẵn phân luồng và điều khiển giao thông tự động.

Điều khiển máy tính trong quân sự cũng có thể tự điều khiển xe cộ, máy bay, tên lửa đạn đạo trong chiến tranh để tránh tiêu hao sức người, đồng thời độ tin cậy và chính xác lại cao hơn nhiều lần.

Điều khiển máy tính trong y tế được sử dụng để hỗ trợ các bác sĩ trong các ca mổ, đem lại sự chính xác cao khi thao tác và đồng thơi là sự an toàn cho bệnh nhân.

Điều khiển máy tính trong văn phòng sử dụng để điều khiển máy in giúp cho chúng ta có thể rễ ràng in ấn các văn bản một cách nhanh chóng.

Điều khiển máy tính cũng có thể giúp chúng ta quan sát, lưu dữ lại được những hình ảnh hay hoạt động thông qua hệ thống Camera được sử dụng trong các nhà hàng, khách sạn, các hệ thống bán hàng như Siêu thị…

CHƯƠNG 1: CÁC KHÁI NIỆM VỀ ĐO LƯỜNG VÀ ĐIỀU KHIỂN MÁY TÍNH

1. Các khái niệm cơ bản về điều khiển bằng máy tính
   1.1. Định nghĩa và phân loại giao tiếp
   1.2. Các hình thức điều khiển bằng máy tính
2. Giao tiếp qua khe ISA
   2.1. Sơ đồ chân rãnh cắm ISA.
   2.2. Các địa chỉ (Address) mặc định của một số thiết bị thông dụng trong máy tính.
   2.3. Các ngắt (IRQ) mặc định của một số thiết bị thông dụng trong máy tính.
3. Giao tiếp qua khe PCI
   3.1. Các đặc điểm của khe cắm PCI.
   3.2. Sơ đồ chân khe cắm PCI
   3.3. Giới thiệu thiết kế card giao tiếp PCI

Mục tiêu:

- Trình bày được cấu trúc giao tiếp máy tính với thiết bị ngoại vi.
- Trình bày được giao tiếp qua khe ISA và PCI.
- Ý thức học tập, nghiên cứu cao
- Chính xác, cẩn thận, tỉ mỉ khi thiết kế mạch giao tiếp qua khe cắm.

1. Các khái niệm cơ bản và điều khiển bằng máy tính

1.1. Định nghĩa và phân loại giao tiếp

Giao tiếp là tiếp xúc giữa các cổng của máy tính với dây cáp của thiết bị ngoài, qua đó chúng có thể nhận dạng và trao đổi dữ liệu với nhau.

Giao tiếp được phân loại như sau:

- Giao tiếp qua khe ISA
- Giao tiếp qua khe PCI
- Giao tiếp qua cổng song song

- Giao tiếp qua cổng SPP
- Giao tiếp qua cổng EPP
- Giao tiếp qua cổng ECP

- Giao tiếp qua cổng nối tiếp (COM)

1.2. Các hình thức điều khiển bằng máy tính

Có hai hình thức điều khiển chính bằng máy tính

- Điều khiển trực tiếp

Là các điều khiển trực tiếp từ máy tính qua các thiết bị ngoài bằng cách kết nối trực tiếp các thiết bị này với may tính.

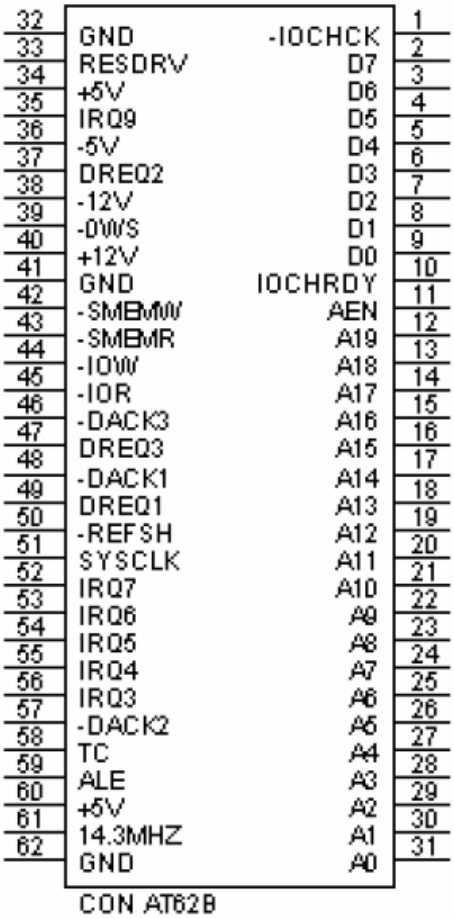
- Điều khiển gián tiếp

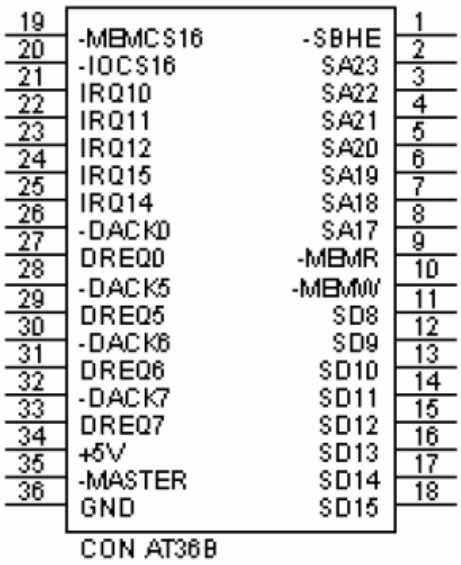
Là các điều khiển khi thiết bị ngoài và máy tính không kết nối trực tiếp với nhau, các thiết bị ngoài này được điều khiển bởi máy tính thông qua các phương tiện gián tiếp. Có nghĩa là các thiết bị ngoài và máy tính được kết nối với nhau thông qua một hệ thống mạng như LAN hay Internet.

2. Giao tiếp qua khe ISA

2.1. Sơ đồ chân rãnh cắm ISA

Rãnh cắm thông dụng nhất là rãnh cắm ISA (*Industry Standard Architecture*) do IBM đưa ra năm 1980 do máy 8086 XT (*Extender Techology*), sau đó là ISA 16 bit cho máy AT (*Advanced Techology*) và trở thành chuẩn AT Bus. Hiện nay các mainboard P4 không còn rãnh cắm này tuy nhiên việc nghiên cứu rãnh cắm ISA cẫn là cần thiết. Rãnh cắm ISA có màu đen trên mainboard gồm 2 phần, phần đầu 62 chân, mỗi hàng có 31 chân dùng cho trao đổi dữ liệu 8 bit, phần thứ 2 36 chân, mỗi hàng 18 chân dùng hỗ trợ thêm khi cần dữ liệu 16 bit. Sơ đồ chân rãnh cắm được mô tả như hình bên dưới:





Sau đây là ý nghĩa vắn tắt các tín hiệu của rãnh cắm (dấu - ở trước báo tín hiệu là tích cực thấp).

SA19 ÷ (System Address bus 19 ÷) (I/O)	Tuyến địa chỉ 20 bit dùng truy cập bộ nhớ 1 Mbyte và ngoại vi. Có thể dùng với LA23 ÷ LA17 truy cập 16 Mbyte bộ nhớ. Khi truy cập ngoại vi dùng 16 bit thấp cho phép truy cập 64K địa chỉ ngoại vi. Ở chế độ đọc hay ghi khi BALE mức cao, địa chỉ được xuất ra và được cài lại ở cạnh xuống của BALE. Các tín hiệu này được điều khiển DMA nhưng cũng có thể được chiếm bởi Card điều khiển gắn vào rãnh cắm.
LA23 ÷ LA17 (Unlatched Address bus 23 ÷ 17)(I/O)	Dùng cùng với SA19 ÷ 0 để truy cập 16 Mbyte bộ nhớ, không được cài lại.
AEN (Address Enable) (O)	Cho phép bộ điều khiển DMA chiếm

	tuyến của vi xử lý khi ở mức cao.
BALE (Buffered Address Latch Enable) (O)	Dùng để cài địa chỉ LA23 ÷ hay dùng để giải mã các địa chỉ này.
CLK (System Clock) (O)	Xung nhịp 4.77 MHz
SD15 ÷ SD0 (System Data) (I/O)	16 bit dữ liệu
–DACK0 ÷ –DACK3, –DACK5 ÷ –DACK7 (DMA Acknowledge) (O)	0÷3 và 5÷7 dùng thông báo cho biết vi xử lý chấp nhận DMA khi có yêu cầu ở các chân DRQ0÷DRQ3 và DRQ5÷DRQ7
DRQ0÷DRQ3, DRQ5÷DRQ7 (DMA Requests) (I)	Dùng khi ngoại vi yêu cầu chiếm tuyến của vi xử lý ISA phục cho DMA (Direct Access Memory) để trao đổi thông tin trực tiếp với bộ nhớ. DRQ sẽ ở mức cao nhất cho đến khi DACK tương ứng ở mức thấp.
– IOCHCK (I/O Channel Check) (I)	Ở mức cao khi có lỗi, ngoài ra có thể do board ISA điều khiển để yêu cầu ngắt NMI
– IOCHRDY (I/O Channel Ready) (I)	Cho phép các board chậm bắt vi xử lý chở bằng cách kéo đường này xuống thấp khi đang ở chu kỳ đọc viết, lúc đó vi xử lý sẽ vào chu kỳ chờ cho đên khi đường này lên mức cao.
– IOR (Read) (I/O)	Báo ngoại vi xuất dữ liệu ra tuyến
– IOW (Write) (I/O)	Báo ngoại vi đọc dữ liệu trên tuyến
IRQ9÷IRQ12, IRQ14÷IRQ15 IRQ3÷IRQ7 (Interrupt Requests)	Tín hiệu vào bào ngoại vi cần ngắt, IRQ sẽ ở mức cao cho đến khi vi xử lý chấp nhận bằng chương trình phục vụ ngắt.
– SMEMR (System Memory Read)(O)	Điều khiển bộ nhớ dưới 1 MB xuất dữ liệu ra.
– SMEMW (System Memory Write) (O)	Điều khiển ghi dữ liệu vào bộ nhớ dưới 1 Mbyte
– MEMR (Memory Read) (O)	Dùng để dọc dữ liệu từ bộ nhớ

– **MEMW** (Memory Write) (O)	Ghi dữ liệu vào bộ nhớ
– **REFRESH** (Memory Refresh) (I/O)	Ở mức thấp nhất trong chu kỳ làm tươi bộ nhớ.
OSC (Oscillator) (O)	Xung nhịp 14.31818 MHz
RESET DRV (Reset Drive) (O)	Tín hiệu reset, ở mức cao khi boot máy
TC (Ter*min*al Count) (O)	Báo đã đếm hết trong hoạt động DMA.
– **MASTER** (I)	Khi board ISA có yêu cầu DMA nhận được DACK, nó sẽ cho Master mức thấp để kiểm soát các tuyến.
– **MEM CS16** (Memory Chip Select 16) (I)	Ở mức thấp khi truyền dữ liệu 16 bit với bộ nhớ.
– **IO CS16** (Chip Select 16) (I)	Do ngoại vi điều khiển ở mức thấp khi muốn truyền dữ liệu 16 bit.
– **OWS** (Zero Wait State) (I)	Do ngoại vi điều khiển ở mức thấp cho biết không cần trạng thái chờ.
– **SBHE** (System Byte High Enable)	Ở mức thấp khi truyền byte cao.

2.2. Các địa chỉ (Address) mặc định của một số thiết bị thông dụng trong máy tính.

Thông qua rãnh cắm ISA có thể truy cập 1024 địa chỉ ngoại vi từ 000 đến 3FF, trong đó một số đã sử dụng cho các thiết bị có sẵn của máy tính như trong bảng sau:

Sau đây là các địa chỉ ngoại vi đã sử dụng của một máy Pentium 2 :

000 – 00F	Truy cập bộ nhớ trực tiếp (DMA Direct memory access controller)
020 – 021	Điều khiển ngắt (OIC Ptogrammable interrupt controller)
040 - 043	Timer hệ thống (System timer)
060 – 060	Bàn phím (Keyboard)
061 - 061	Loa trong (System speaker)
064 - 064	Bàn phím
070- 071	RAM hệ thống và đồng hồ thời gian thực (System CMOS/RTC)
081 - 083	DMA

087 – 087	DMA
089 – 08B	DMA
08F - 091	DMA
0A0 – 0A1	PIC
0C0 – 0CF	DMA
0F0 – 0FF	Đồng sử lý số học (Numeric data ptocessor)
168 – 16F	Điều khiển đĩa cứng (Standard IDE/ESDI Hard Disk controller)
170 – 177	Điều khiển IDE (secondary IDE controller intel 82371 AB/EB)
1F0 – 1F7	Điều khiển IDE (Primary IDE controller)
201 – 201	Que trò chơi (Game port foystick)
208 – 20F	Dành cho Miainboard (Metherboard resources)
220 – 22F	Card âm thanh (ES 1868 Plug and play Audio Drive)
274 – 277	IO read data port for ISA Plug and Play enumerator
2F8 – 2FF	Cổng truyền thông 2 (COM2)
330 – 331	Card âm thanh
36E – 36F	Điều khiển đĩa cứng (Standard IDE/ESDI Hard Disk Comtroller)
367 – 376	Điều khiển IDE (Secondary IDE controller)
378 – 37F	Cổng song song (LPT1)
388 – 38B	Card âm thanh
3B0 – 3BB	Card video S3 Inc. Trio3D/2X (Engineering Release)
3C0 – 3DF	Card video S3 Inc. Trio3D/2X (Engineering Release)
3F2 – 3F5	Điều khiển ổ đĩa mềm (Stardard Floppy Disk Controller)
3F6 – 3F6	Điều khiển IDE (Primary IDE controller)
3F8 – 3FF	Cổng truyền thông 1 (COM1)

2.3. Các ngắt (IRQ) mặc định của một số thiết bị thông dụng trong máy tính.

Các thiết bị ngoại vi thường dùng ngắt để tác động đến CPU yêu cầu làm việc gì đó bằng cách đưa chân IRQ lên mức cao. Các chân này thường được dành sẵn cho các thiết bị cụ thể.

Các ngắt của một máy Pentium 2

0	Timer hệ thống
1	Bàn phím
2	PIC
3	Cổng truyền thông 2
4	Cổng truyền thông 1
5	Card âm thanh
6	Điều khiển ổ đĩa mềm
7	Cổng song song (LPT1)
8	RAM hệ thống và đồng hồ thời gian thực
9	Modem (Motorola SM56 PIC Speakerphone Modem)
9	IRQ cho PCI
10	Điều khiển ổ đĩa cứng
11	Điều khiển USB (Intel 8237-1 AB/EB PCI to USB Universal Host Controller)
11	IRQ cho PCI
12	Chuột PS/2
13	Đồng sử lý số học
14	Điều khiển IDE thứ nhất
15	Điều khiển IDE thứ hai

3. Giao tiếp qua khe PCI

3.1. Các đặc điểm của khe cắm PCI

Rãnh cắm PCI (*Peripheral Component Interconnect*) có màu trắng trên mainboard cho phép giao tiếp ngoại vi 32 hay 64 bit vận tốc nhanh đến 132 Mbyte/s sp với rãnh cắm ISA 16 bit có bận tốc 0 ÷ 5 Mbyte/s.

Nhờ vận tốc cao nên rãnh PCI thường dùng cho card mà hình, sau đó nó được sử dụng để cho các card khác như card mạng, modem nội, âm thanh… và dần dần các mainboard đời mới không dành chỗ cho rãnh ISA nữa.

Các hãng như Advantech, Data Translation… cũng đã sản xuất card giao tiếp ngoại vi cho máy tính dùng rãnh PCI.

Rãnh PCI 64 bit có hai hàng tiếp điểm, mỗi bên 94 tiếp điểm phía A là phía linh kiện còn phía B là phía hàn. Do tính chất phức tạp của tuyến và vận tốc tín hiệu lớn nên việc tự ráp card giao tiếp PCI khó thực hiện mà phỉa dùng card chính hãng.

Năm 1998 các hãng Compaq, Hewlett – Packerd, IBM phối hợp đưa ra chuẩn PCI-X (PCI Express) có đặc tính tốt hơn.

Tuyến PCI 32 bit sử dụng chung 32 đường địa chỉ data ADO – 31, pha địa chỉ so tín hiệu FRAME# điều khiển, sau đó là một hay nhiều pha dữ liệu. Tuyến PCI 64 bit dùng 64 đường ADO – 63 cho địa chỉ và dữ liệu.

Có 2 loại tuyến PCI mức tín hiệu 5V và mức tín hiệu 3,3V.

Sau đây là mô tả các tín hiệu của PCI:

CLK	Xung nhịp 33Hz, 66MHz…
RST#	Tín hiệu reset
ADO ÷ AD31	Tuyến địa chỉ khi FRAME# ở mức thấp
C/BEO ÷ 3#BUS *(Command BytesEnables)*	Cho biết loại của truyền dữ liệu (đọc/viết bộ nhớ, ngoại vi…)
PAR	Kiểm tra parity của ADO÷31 và C/BEO÷3
IRDY# *(Initiator Ready)*	
TRDY# *(Target ready)*	Hai dữ liệu bắt tay giữa bộ phát và bộ nhận dữ liệu trên tuyến PCI
STOP#	Tín hiệu target báo cho initiator để chấm dứt giao dịch, Initiator là chủ của tuyến (bus master) còn target là bus slave. Việc truyền sữ liệu so initiator bắt đầu thông qua C/BE và IRDY còn target trả lời thông qua TRDY# và STOP#.
LOCK#	Tín hiệu initiator báo dành riêng một số địa chỉ target.
IDSEL *(Initialigation Devia Select)*	Tín hiệu chọn chip
DEVSEL# *(Device Select)*	Của nó trên tuyến PCI do target điều khiển khi nó thấy địa chỉ của nó trên tuyến PCI
REQ#	Yêu cầu dùng bus (request)
GNT#	Cho biết yêu cầu

REQ#	Đã được chấp nhận (grant)
PERR# *(Parity Error)*	
SERR# *(System Error)*	Sai hệ thống
INTA#, INTB#, INIC#, INID#	Các tín hiệu ngắt
SBO# (Snoop Backoff) **SDONE** (Snoop done)	Dùng cho card memory
PRSNT 1 ÷ 2#	Cho biết có board cắm vào slot và công xuất tiêu thụ của board đó.
CLKRUN# (Clock Running)	Cho phép điều khiển xung nhịp CLK
MGGEN(66MHz enable)	Cho biết xung nhịp 33MHz hay 66MHz
AD 32 + 63	32 đường địa chỉ và dữ liệu cao trong PCI 64 bit.
C/BE 4 ÷ 7#	Dùng khi truyền 64 bit kết hợp với REQ 64# và ACK 64#, PAR 64
REQ 64# *(Request 64 bit transfer)*	
ACK 64# *(Acknowledge 64 bit transfer)*	
TCK *(Test clock)* **TDI** *(Test data input)* **TDO** *(Test output)* **TMS** *(Test mode Select)* **TRST#** *(Test Reset)*	Các tín hiệu thử

3.2. Sơ đồ chân khe cắm PCI

Sơ đồ chân rãnh cắm PCI 64 bit

PIN	5V system enclronment		PIN	3.3V system envlronment		Comments
	Slde B	Slde A		Slde B	Slde A	
1	-12V	TRST#	1	-12V	TRST#	32-bit start
2	TCK	+12V	2	TCK	+12V	
3	Ground	TMS	3	Ground	TMS	
4	TDO	TDI	4	TDO	TDI	
5	+5V	+5V	5	+5V	+5V	
6	+5V	INTA#	6	+5V	INTA#	
7	INTB#	INTC#	7	INTB#	INTC#	
8	INTD#	+5V	8	INTD#	+5V	
9	PRSNT1#	Reserved	9	PRSNT1#	Reserved	
10	Reserved	+5V (I/O)	10	Reserved	+3.3V (I/O)	
11	PRSNT2#	Reserved	11	PRSNT2#	Reserved	
12	Ground	Ground	12	Connector Key		3.3V key
13	Ground	Ground	13	Connector Key		3.3V key
14	Reserved	Reserved	14	Reserved	Reserved	
15	Ground	RST#	15	Ground	RST#	
16	CLK	+5V (I/O)	16	CLK		
17	Ground	GNT#	17	Ground	+3.3V (I/O)	
18	REQ#	Ground	18	REQ#	Ground	
19	+5V (I/O)	Reserved	19	+3.3V (I/O)	Reserved	
20	AD[31]	AD[30]	20	AD[31]	AD[30]	
21	AD[29]	+3.3V	21	AD[29]	+3.3V	
22	Ground	AD[28]	22	Ground	AD[28]	
23	AD[27]	AD[26]	23	AD[27]	AD[26]	
24	AD[25]	Ground	24	AD[25]	Ground	

25	+3.3V	AD[24]	25	+3.3V	AD[24]	
26	C/BE[3]#	IDSEL	26	C/BE[3]#	IDSEL	
27	AD[23]	+3.3V	27	AD[23]	+3.3V	
28	Ground	AD[22]	28	Ground	AD[22]	
29	AD[21]	AD[20]	29	AD[21]	AD[20]	
30	AD[19]	Ground	30	AD[19]	Ground	
31	+3.3V	AD[18]	31	+3.3V	AD[18]	
32	AD[17]	AD[16]	32	AD[17]	AD[16]	
33	C/BE[2]#	+3.3V	32	C/BE[2]#	+3.3V	
34	Ground	FRAME#	34	Ground	FRAME#	
35	IRDY#	Ground	35	IRDY#	Ground	
36	+3.3V	TRDY#	36	+3.3V	TRDY#	
37	DESVEL#	Ground	37	DESVEL#	Ground	
38	Ground	Stop#	38	Ground	Stop#	
39	LOCK#	3.3V	39	LOCK#	3.3V	
40	PERR#	SDONE	40	PERR#	SDONE	
41	+3.3V	SBO#	41	+3.3V	SBO#	
42	SERR#	Ground	42	SERR#	Ground	
43	+3.3V	PAR	43	+3.3V	PAR	
44	C/BE[1]#	AD[15]	44	C/BE[1]#	AD[15]	
45	AD[14]	+3.3V	45	AD[14]	+3.3V	
46	Ground	AD[13]	46	Ground	AD[13]	
47	AD[12]	AD[11]	47	AD[12]	AD[11]	
48	AD[10]	Ground	48	AD[10]	Ground	
49	Ground	AD[09]	49	M66EN	AD[09]	
50	Connector Key		50	Ground	Ground	5V key
51	Connector Key		51	Ground	Ground	5V key
52	AD[08]	C/BE[0]#	52	AD[08]	C/BE[0]#	
53	AD[07]	+3.3V	53	AD[07]	+3.3V	

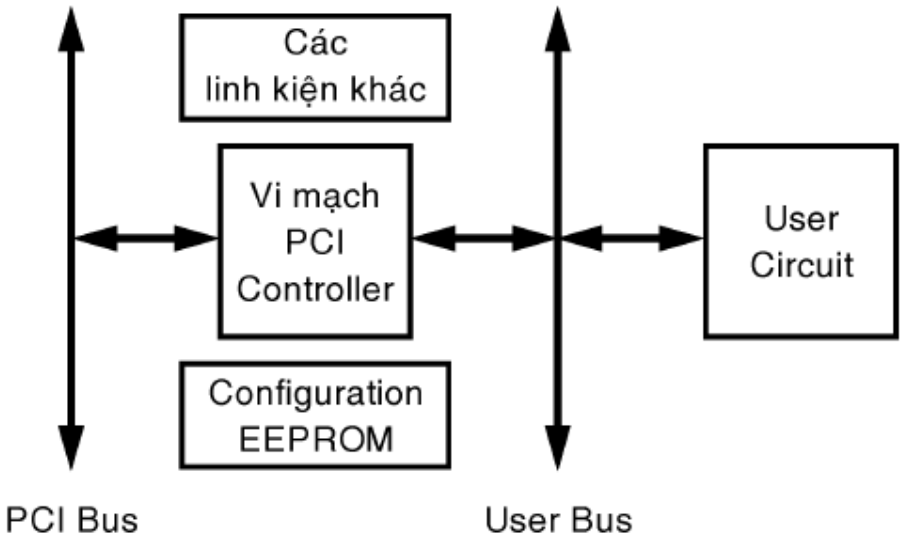
54	+3.3V	AD[06]	54	+3.3V	AD[06]	
55	AD[05]	AD[04]	55	AD[05]	AD[04]	
56	AD[03]	Ground	56	AD[03]	Ground	
57	Ground	AD[02]	57	Ground	AD[02]	
58	AD[01]	AD[00]	58	AD[01]	AD[00]	
59	+5V (I/O)	+5V (I/O)	59	3.3V (I/O)	3.3V (I/O)	
60	ACK 64#	REQ 64#	60	ACK 64#	REQ 64#	
61	+5V	+5V	61	+5V	+5V	
62	+5V	+5V	62	+5V	+5V	32bit end
	Connector Key			Connector Key		64bit spacer
	Connector Key			Connector Key		64bit spacer
63	Reserved	Ground	63	Reserved	Ground	64 bit Start
64	Ground	C/BE[7]#	64	Ground	C/BE[7]#	
65	C/BE[6]#	C/BE[5]#	65	C/BE[6]#	C/BE[5]#	
66	C/BE[4]#	+5V (I/O)	66	C/BE[4]#	+3.3V (I/O)	
67	Ground	PAR 64	67	Ground	PAR 64	
68	AD[63]	AD[62]	68	AD[63]	AD[62]	
69	AD[61]	Ground	69	AD[61]	Ground	
70	+5V (I/O)	AD[60]	70	+3.3V (I/O)	AD[60]	
71	AD[59]	AD[58]	71	AD[59]	AD[58]	
72	AD[57]	Ground	72	AD[57]	Ground	
73	Ground	AD[56]	73	Ground	AD[56]	
74	AD[55]	AD[54]	74	AD[55]	AD[54]	
75	AD[53]	+5V (I/O)	75	AD[53]	+3.3V (I/O)	
76	Ground	AD[52]	76	Ground	AD[52]	
77	AD[51]	AD[50]	77	AD[51]	AD[50]	
78	AD[49]	Ground	78	AD[49]	Ground	
79	+5V (I/O)	AD[48]	79	+3.3V (I/O)	AD[48]	
80	AD[47]	AD[46]	80	AD[47]	AD[46]	

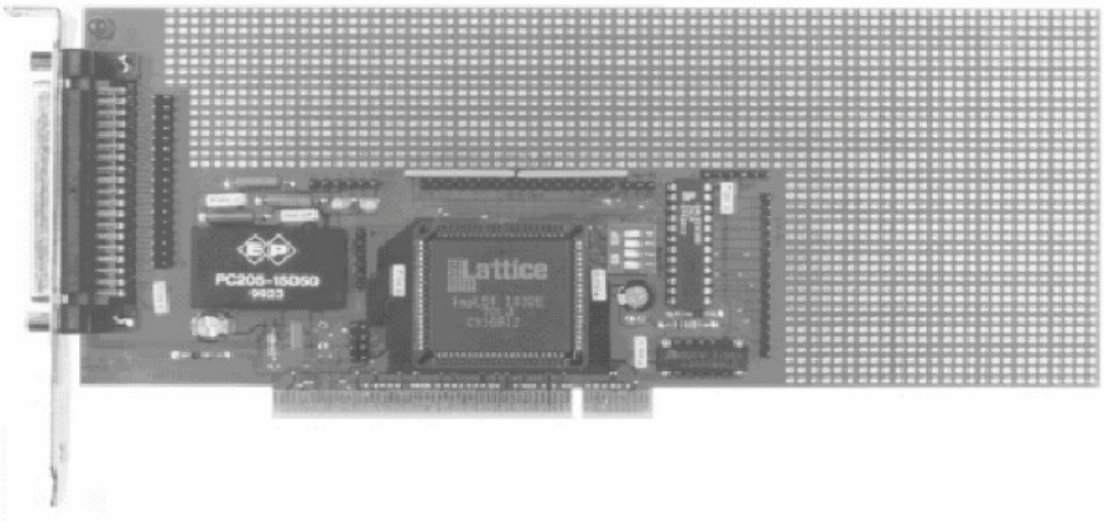
81	AD[45]	Ground	81	AD[45]	Ground	
82	Ground	AD[44]	82	Ground	AD[44]	
83	AD[43]	AD[42]	83	AD[43]	AD[42]	
84	AD[41]	+5V (I/O)	84	AD[41]	+3.3V (I/O)	
85	Ground	AD[40]	85	Ground	AD[40]	
86	AD[39]	AD[38]	86	AD[39]	AD[38]	
87	AD[37]	Ground	87	AD[37]	Ground	
88	+5V (I/O)	AD[36]	88	+3.3V (I/O)	AD[36]	
89	AD[35]	AD[34]	89	AD[35]	AD[34]	
90	AD[33]	Ground	90	AD[33]	Ground	
91	Ground	AD[32]	91	Ground	AD[32]	
92	Reserved	Reserved	92	Reserved	Reserved	
93	Reserved	Ground	93	Reserved	Ground	
94	Ground	Reserved	94	Ground	Reserved	64-bit end

3.3. Giới thiệu thiết kế card giao tiếp PCI

Việc thiết kế các card này tương đối đơn giản. Do các mainboard đời mới không còn hỗ trợ tuyến ISA nên chúng ta phải chuyển sang sử dụng tuyến PCI bằng cách mua các card chuyên dụng của các hãng với phần mềm kèm theo. Do sự phức tạp của tuyến PCI, việc tự thiết kế và chế tạp card PCI tương đối khó khăn, đòi hỏi sử dụng các linh kiện FPGA có mật độ tích hợp cao, mạch in nhiều lớp và công nghệ dán, ngoài ra việc lập trình cho card này cũng không phải dễ dàng mà phải thông qua các hàm Windows API. Việc thiết kế sẽ trở nên dễ ràng nếu dùng các bộ PCI development kit có sẵn. Các bộ kit này giúp tạo ứng dụng PCI khác nhau cùng với software kèm theo.

Cấu trúc chung Card PCI như sau:





Phần tử chính trong Card là vi mạch PCI Controller dùng làm cầu nối giữa tuyến PCI và mạch người dùng, chế tạo bởi các hãng theo công nghệ ASIC, ví dụ như PCI9050/9054 của PLX Technologies, ispLSI 1032E của Lattice… Phần tử thứ 2 là EEPROM dùng để chứa thông tin về card phục vụ cho PnP (Plug and Play) khi khởi động máy tính. User Bus gồm tuyến dữ liệu 16 bit, tuyến địa chỉ và tuyến điều khiển dùng kết nối với các linh kiên thông thường. Thông qua Development Kit chúng ta có thể học tập cách thiết kế card. Một card tiêu chuẩn là PCI-Proto LAB/PLX-M vi mạch (www.pci-tools.com) hoặc card PIKS-161 (www.komcard.com).

CHƯƠNG 2: GIAO TIẾP QUA CỔNG SONG SONG (PARALLEL)

1. Giới thiệu cổng parallel.
   1.1. Sơ đồ chân
   1.2. Chức năng và hoạt động
2. Các đặc tính của cổng Parallel
   2.1. Tốc độ
   2.2. Tính chống nhiễu
3. Giao tiếp cổng SPP
   3.1. Đặc tính của SPP.
   3.2. Sơ đồ chân của SPP
   3.3. Thanh ghi và địa chỉ giao tiếp
   3.4. Giao tiếp 1 chiều
   3.5. Giao tiếp 2 chiều
   3.6. Minh họa giao tiếp song song với TTL 74LS157
4. Giao tiếp cổng EPP
   4.1. Đặc tính của EPP.
   4.2. Sơ đồ chân của EPP
   4.3. Kết nối 2 máy tính qua cổng EPP
      4.3.1. Sơ đồ dây nối EPP
      4.3.2. Triển khai kết nối 2 máy tính (Laplink)
5. Giao tiếp cổng ECP
   5.1. Đặc tính của ECP.
   5.2. Sơ đồ chân của ECP
   5.3. Kết nối 2 máy tính qua cổng ECP
      5.3.1. Sơ đồ dây nối ECP
      5.3.2. Triển khai kết nối 2 máy tính (Laplink)

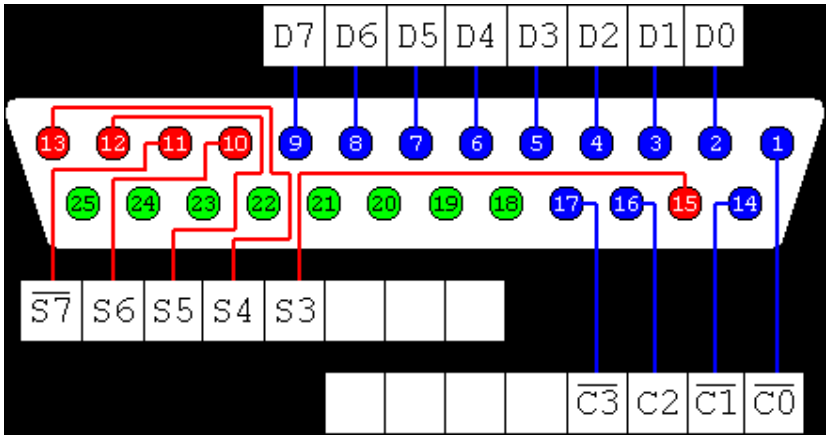
Mục tiêu:

- Trình bày được về chuẩn giao tiếp song song.
- Phân biệt được sự khác biệt giữa các dạng giao tiếp song song SPP, EPP, ECP.
- Trình bày được cấu trúc giao tiếp, phục vụ cho môn học ‘ Lập trình ghép nối máy tính’ sau này
- Thực hiện kết nối 2 máy tính qua cổng song song
- Ý thức học tập, nghiên cứu cao
- Chính xác, cẩn thận, tỉ mỉ khi thiết kế mạch giao tiếp qua cổng song song.

1. Giới thiệu cổng Parallel

1.1. Sơ đồ chân

Đa số các máy vi tính đều trao đổi thông tin thông qua các ngã sau đây: Parallel port, Serial port, USB và Network card. Parallel port là một phần không thể thiếu trong việc sử dụng computer để giao tiếp với các thiết bị điện tử khác.



Dưới đây là liệt kê 25 chân của Parallel Port với tên gọi (Hardware và software) và thứ tự của từng chân.

Chân	Tên signal (*dùng cho hardware*)	Direction/type (*nhìn từ PC*)	Tên signal và thứ tự của bit (*dùng cho software*)	Normal signal line function
1	-STROBE	OC/Pullup	Control register bit 0	kích hoạt thông báo gửi hoặc nhận data, 0 là đọc, 1 là viết
2	D0	Hai chiều	Data register bit 0	bit 0 chứa data
3	D1	Hai chiều	Data register bit 1	bit 1 chứa data

4	D2	Hai chiều	Data register bit 2	bit 2 chứa data
5	D3	Hai chiều	Data register bit 3	bit 3 chứa data
6	D4	Hai chiều	Data register bit 4	bit 4 chứa data
7	D5	Hai chiều	Data register bit 5	bit 5 chứa data
8	D6	Hai chiều	Data register bit 6	bit 6 chứa data
9	D7	Hai chiều	Data register bit 7	bit 7 chứa data
10	-ACK	Input	Status register bit 6	Pulsed low by printer to acknowledge data byte Rising (usually) edge causes IRQ if enabled
11	BUSY	Input	Status register bit 7	Kích hoạt khi Printer đang bận (Busy)
12	NOPAPER	Input	Status register bit 5	Kích hoạt khi printer hết giấy
13	SELECTED	Input	Status register bit 4	Kích hoạt khi printer đang hoạt động
14	-AUTOFEED	OC/Pullup	Control register bit 1	Kích hoạt thông báo data đã sẵn sàng để học hoặc viết
15	-ERROR	Input	Status register bit 3	Kích hoạt khi printer bi lỗi
16	-INITILIZE	OC/Pullup	Control register bit 2	Kích hoạt để printer reset lại vị trí ban đầu
17	-SELECT	OC/Pullup	Control register bit 3	Kích hoạt để đánh dấu printer nhận được valid address
18	Ground			
…	Ground			Chân (18-25) bỏ trống, dùng tùy ý
25	Ground			

1.2. Chức năng và hoạt động

Cổng song song là cổng thường được sử dụng để giao tiếp trong các đề án nhỏ. Cổng này cho phép vào tới 9 bit và ra tới 12 bit vào bất cứ thời điểm nào, do vậy yêu cầu dòng điện của thiết bị ngoại vi phải rất nhỏ để thi hành nhiều nhiệm vụ đơn giản.

Cổng này gồm có 4 đường điều khiển, 5 đường trạng thái và 8 đường dữ liệu. Nó thường được tìm thấy sau mỗi máy PC và có hình dạng là một lổ cắm 25 chân, và là đầu nối loại female. Cũng có loại là đầu nối 25 chân loại male. Đây có thể là cổng nối tiếp RS-232C và do vậy nên hoàn toàn không tương thích.

Cổng song song mới hơn được tiêu chuẩn hóa dưới chuẩn IEEE 1284, và được đưa ra lần đầu năm 1994. Chuẩn này xác định 5 phương thức hoạt động, mà nó được mô tả như sau:

1. Compatibility Mode
2. Nibble Mode
3. Byte Mode
4. EPP Mode
5. ECP Mode

Mục tiêu là để phát họa trình điều khiển mới và những thiết bị mà nó tương thích với mỗi loại khác nhau, và nó cũng tương thích ngược lại với chuẩn Standard Parallel Port (SPP). Capatibility Mode, Nibble Mode và Byte Mode chỉ sử dụng cho phần cứng chuẩn mà có giá trị trên những Card cổng song song nguyên thủy, trong khi đối với cổng ECP và cổng EPP đòi hỏi phải thêm vào một số thuộc tính phần cứng để có thể chạy ở tốc độ nhanh hơn, trong khi vẫn trở về tương thích với chế độ chuẩn SPP.

Compatibility Mode hay là Centronics Mode mà thường được biết tới thì thường chỉ có thể gửi dữ liệu theo một hướng ra mà không thể theo chiều ngược lại, và chỉ truyền với tốc độ đặc trưng là khoảng 50 Kbyte/giây nhưng cũng có khi lên tới 150 Kbyte/giây. Để nhận dữ liệu chúng ta cần phải thay đổi phương thức hoạt động hoặc là Nibble Mode hay là Byte Mode. Nibble Mode có thể nhập vào (input) một nibble (4 bit) theo hướng ngược lại. Ví dụ từ thiết bị đến máy tính. Chế độ Byte Mode dùng cổng song song hai chiều (được tìm thấy trên một số loại Card) có nét đặc trưng là vào một lúc 8 bit, tức là truyền một byte dữ liệu theo hướng ngược lại.

Cổng song song có khả năng mở rộng hay cổng song song nâng cao sử dụng thêm vào một số đặc tính phần cứng để phát ra và quản lý tín hiệu bắt tay. Để đưa một byte ra máy in hay bất cứ một nội dung gì, dùng phương thức Compatility Mode thì chương trình yêu cầu phải theo các bước là:

1. Viết một byte ra Port dữ liệu.

2. Kiểm tra xem máy in có bận hay không. Nếu máy in đang bận, nó sẽ không chấp nhận bất cứ một dữ liệu nào gửi ra, do đó bất cứ dữ liệu nào gửi ra sẽ bị mất.

3. Đưa chân Strobe (chân 1) xuống thấp. Điều này nhằm để thông báo với máy in rằng đang có một dữ liệu trên đường dữ liệu (Data lines: chân 2-9).

4. Đặt chân Strobe lên cao lại sau khi đợi khoảng 5us thì đặt chân Strobe xuống thấp lại (trở lại bước 3).

Điều này sẽ giới hạn tốc độ mà cổng có thể hoạt động. Cổng ECP và cổng EPP dựa vào đó để đánh dấu trên phần cứng để kiểm tra xem máy in có bận hay không. Nếu máy in bận phát ra tín hiệu hay là để dành riêng cho tín hiệu bắt tay. Điều này có nghĩa là chỉ có một chỉ thị vào ra (In/Out) cần để thi hành để gia tăng tốc độ. Cổng loại này cho phép truyền ra với tốc độ từ 1 đến 2 MegaByte/giây. Cổng ECP có một phần thuận lợi nữa là có thể sử dụng kênh DMA và bộ đệm FIFO, do vậy mà dữ liệu truyền ra không cần chỉ thị vào ra (In/Out).

2. Các đặc tính của cổng Parallel

2.1. Tốc độ

Khoảng cách cực đại giữa thiết bị ngoại vi và máy tính ghép qua cổng song song thường bị hạn chế. Lý do là hiện tượng cảm ứng giữa các đường dẫn và điện dung kí sinh hình thành giữa các đường dẫn có thể làm biến dạng tín hiệu. Khoảng cách giới hạn cực đại là 8m. Thông thường chỉ 1,5 đến 2m vì lí do an toàn dữ liệu. Nếu sử dụng khoảng cách ghép nối trên 3m thì các đường dây tín hiệu và đường dây nối đất phải được soắn với nhau thành từng cặp để giảm thiểu ảnh hưởng của nhiễu. Biện pháp khác sử dụng cáp dẹt, trên đó mỗi đường dữ liệu được đặt giữa haiđường dây nối đất.

Tốc độ truyền dữ liệu qua cổng song song phụ thuộc vào phần cứng được sử dụng. Trênl ý thuyết tốc độ có thể đạt đến 1Mb/s nhưng với khoảng cách truyền hạn chế trong phạm vi 1m. Với nhiều mục đích sử dụng thì khoảng cách này hoàn toàn thỏa đáng, tuy vậy cũng có những ứng dụng đòi hỏi phải truyền trên khoảng cách xa hơn. Trong trường hợp đó ta phải nghĩ ngay đến khả năng ghép nối khác (như ghép nối qua cổng RS232).

2.2. Tính chống nhiễu

Cổng song song là cổng thường được sử dụng để giao tiếp trong các đề án nhỏ. Cổng này cho phép vào tới 9 bit và ra tới 12 bit vào bất cứ thời điểm nào,

do vậy yêu cầu dòng điện của thiết bị ngoại vi phải rất nhỏ để thi hành nhiều nhiệm vụ đơn giản.

Cổng này gồm có 4 đường điều khiển, 5 đường trạng thái và 8 đường dữ liệu. Nó thường được tìm thấy sau mỗi máy PC và có hình dạng là một lổ cắm 25 chân, và là đầu nối loại female. Cũng có loại là đầu nối 25 chân loại male. Đây có thể là cổng nối tiếp RS-232C và do vậy nên hoàn toàn không tương thích.

Cổng song song mới hơn được tiêu chuẩn hóa dưới chuẩn IEEE 1284, và được đưa ra lần đầu năm 1994. Chuẩn này xác định 5 phương thức hoạt động, mà nó được mô tả như sau:

1. Compatibility Mode
2. Nibble Mode
3. Byte Mode
4. EPP Mode
5. ECP Mode

Mục tiêu là để phát họa trình điều khiển mới và những thiết bị mà nó tương thích với mỗi loại khác nhau, và nó cũng tương thích ngược lại với chuẩn Standard Parallel Port (SPP). Capatibility Mode, Nibble Mode và Byte Mode chỉ sử dụng cho phần cứng chuẩn mà có giá trị trên những Card cổng song song nguyên thủy, trong khi đối với cổng ECP và cổng EPP đòi hỏi phải thêm vào một số thuộc tính phần cứng để có thể chạy ở tốc độ nhanh hơn, trong khi vẫn trở về tương thích với chế độ chuẩn SPP.

Compatibility Mode hay là Centronics Mode mà thường được biết tới thì thường chỉ có thể gửi dữ liệu theo một hướng ra mà không thể theo chiều ngược lại, và chỉ truyền với tốc độ đặc trưng là khoảng 50 Kbyte/giây nhưng cũng có khi lên tới 150 Kbyte/giây. Để nhận dữ liệu chúng ta cần phải thay đổi phương thức hoạt động hoặc là Nibble Mode hay là Byte Mode. Nibble Mode có thể nhập vào (input) một nibble (4 bit) theo hướng ngược lại. Ví dụ từ thiết bị đến máy tính. Chế độ Byte Mode dùng cổng song song hai chiều (được tìm thấy trên một số loại Card) có nét đặc trưng là vào một lúc 8 bit, tức là truyền một byte dữ liệu theo hướng ngược lại.

Cổng song song có khả năng mở rộng hay cổng song song nâng cao sử dụng thêm vào một số đặc tính phần cứng để phát ra và quản lý tín hiệu bắt tay. Để đưa một byte ra máy in hay bất cứ một nội dung gì, dùng phương thức Compatility Mode thì chương trình yêu cầu phải theo các bước là:

- Viết một byte ra Port dữ liệu.

- Kiểm tra xem máy in có bận hay không. Nếu máy in đang bận, nó sẽ không chấp nhận bất cứ một dữ liệu nào gửi ra, do đó bất cứ dữ liệu nào gửi ra sẽ bị mất.
- Đưa chân Strobe (chân 1) xuống thấp. Điều này nhằm để thông báo với máy in rằng đang có một dữ liệu trên đường dữ liệu (Data lines: chân 2-9).
- Đặt chân Strobe lên cao lại sau khi đợi khoảng 5us thì đặt chân Strobe xuống thấp lại (trở lại bước 3).

Điều này sẽ giới hạn tốc độ mà cổng có thể hoạt động. Cổng ECP và cổng EPP dựa vào đó để đánh dấu trên phần cứng để kiểm tra xem máy in có bận hay không. Nếu máy in bận phát ra tín hiệu hay là để dành riêng cho tín hiệu bắt tay. Điều này có nghĩa là chỉ có một chỉ thị vào ra (In/Out) cần để thi hành để gia tăng tốc độ. Cổng loại này cho phép truyền ra với tốc độ từ 1 đến 2 MegaByte/giây. Cổng ECP có một phần thuận lợi nữa là có thể sử dụng kênh DMA và bộ đệm FIFO, do vậy mà dữ liệu truyền ra không cần chỉ thị vào ra (In/Out).

3. Giao tiếp cổng SPP

3.1. Đặc tính của SPP

Cổng song song có đầu nối 25 chân cái thường dùng để kết nối với máy in đầu nối Centronics 34 chân. SPP (standard parallel port)

3.2. Sơ đồ chân của SPP

Bảng sau cho sơ đồ chân và ý nghĩa các chân của cổng SPP khi dùng với máy in, dấu nhắc "/" có nghĩa tích cực thấp. *Ví dụ,* chân 15 là /Error hướng vào, nếu chân này xuống mức 0 là có lỗi. Cột đảo ghi chữ Có tức là tín hiệu được đảo mức, ví dụ chân 17 khi đưa mức logic 0 ứng với chân này vào thanh ghi điều khiển thì ở chân 17 xuất hiện mức 1.

Chân D – 25	Chân Centronics	Tín hiệu SPP	Hướng	Thanh ghi	Đảo
1	1	/Strobe	Vào/Ra	Điều khiển	Có
2	2	Data 0	Ra	Data	
3	3	Data 1	Ra	Data	
4	4	Data 2	Ra	Data	
5	5	Data 3	Ra	Data	
6	6	Data 4	Ra	Data	
7	7	Data 5	Ra	Data	
8	8	Data 6	Ra	Data	
9	9	Data 7	Ra	Data	
10	10	/Ack	Vào	Trạng thái	
11	11	Busy	Vào	Trạng thái	Có
12	12	Paper Out	Vào	Trạng thái	
13	13	Select	Vào	Trạng thái	
14	14	/Auto-Linefeed	Vào/Ra	Điều khiển	Có
15	32	/Error	Vào	Trạng thái	
16	31	/Init	Vào/Ra	Điều khiển	
17	36	/Select Printer	Vào/Ra	Điều khiển	Có
18 - 25	19 - 30	Mass			

3.3. Thanh ghi và địa chỉ giao tiếp

Việc giao tiếp được thực hiện qua 3 thanh ghi: thanh ghi dữ liệu, thanh ghi điều khiển và thanh ghi trạng thái. Thông thường sử dụng hai địa chỉ gốc:

378H cho LPT1 (Line printer 1)

278H cho LPT 2

Một số máy dùng địa chỉ 03BC.

- Thanh ghi dữ liệu có địa chỉ gốc +0,8 bit, nhận dữ liệu để xuất ra ngoài, dữ liệu được chốt…

	7	6	5	4	3	2	1	0
Tín hiệu máy	D7	D6	D5	D4	D3	D2	D1	D0
Chân số	9	8	7	6	5	4	3	2

- Thanh ghi trạng thái địa chỉ gốc +1 là thanh ghi chỉ đọc dùng để nhận tín hiệu từ ngoài vào, có 5 tín hiệu vào.

D7							D0
Busy	/ACK	Paper Out	Select	/Error	/IRQ	X	X
11	10	12	13	15			

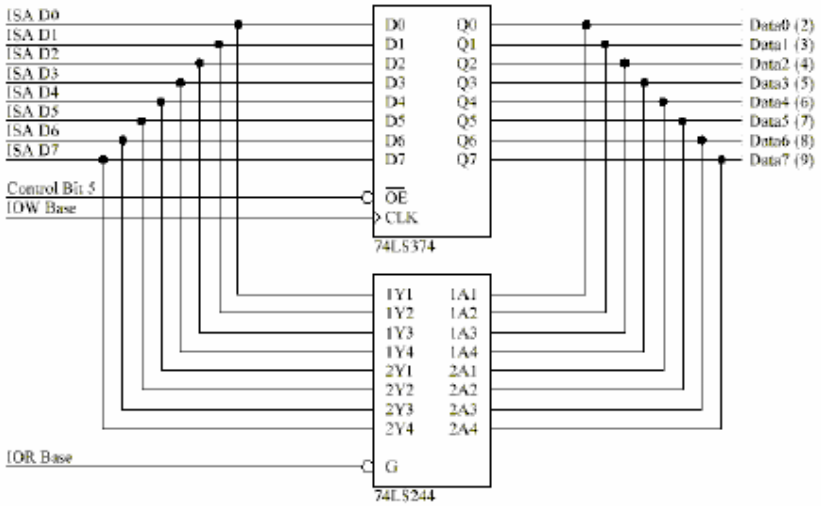
Chú ý là bit Busy được đảo, nghĩa là nếu chân 11 có điện áp +5V thì bit D7 của thanh ghi trạng thái mức logic 0, bit D2 bằng 0 khi có ngắt từ /ACK.

- Thanh ghi điều khiển có 4 đường ra điều khiển, địa chỉ gốc +2, các đường này dùng cực thu hở do đó có thể giao tiếp 2 chiều.

D7 D0

X	X	Cho phép cổng 2 chiều	Cho phép IRQ qua ACK	Chọn máy in /Select 17	Khởi động máy in /Init 16	Xuống bằng /Auto Linefeed 14	Kích /Strobe 1

Các chân 1, 11, 14 và 17 được đảo phần cứng, bit D6 thanh ghi trạng thái (Chân sô 10) từ 1->0 thì gây ra ngắt IRQ7 nếu được cho phép bởi D4 của thanh ghi điều khiển = 1.



Sơ đồ thanh ghi dữ liệu hai hướng

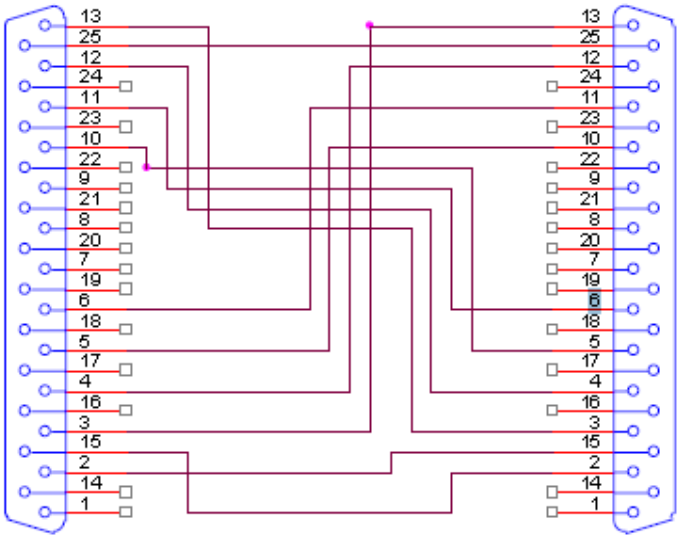
Một số mainboard hỗ trợ giao tiếp hai chiều qua thanh ghi dữ liệu, bit D5 của thanh ghi điều khiển bằng 1 thì cho phép các chân 2…:9 của thanh ghi dữ liệu có chiều đi vào, nghĩa là có thể đưa tín hiệu vào các chân này rồi đọc thanh ghi dữ liệu.

Các chân của thanh ghi điều khiển có ngõ ra cực thu hở nên có thể nhận tín hiệu vào nếu trước đó ta đã nạp 8 bit sao cho các ngõ ra ứng với thanh ghi này lên 1. Do các tín hiệu /BUSY, Select, /AF và /Strobe đã được đảo phần cứng nên ta thêm các cổng đảo, logic đoch vào phản ánh đúng mức tín hiệu.

3.4. Giao tiếp 1 chiều

Quá trình giao tiếp với cổng song song dùng 2 chế độ: chế độ chuẩn SPP và chế độ

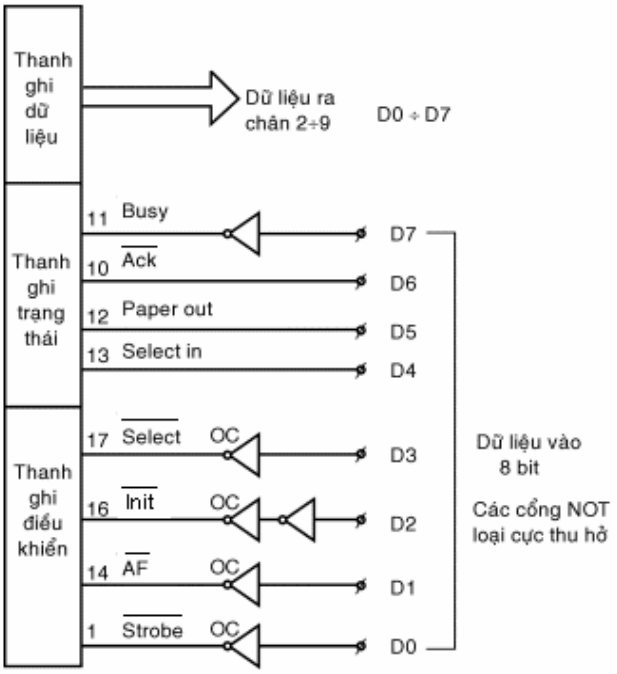
mở rộng. Việc giao tiếp ở chế độ chuẩn mô tả như sau:



Sơ đồ chân kết nối mo tả như sau:

PC1		PC2	
Chức năng	**Chân**	**Chân**	**Chức năng**
D0	2	15	$\overline{\text{ERROR}}$
D1	3	13	SELECT
D2	4	12	PAPER EMPTY
D3	5	10	$\overline{\text{ACK}}$
D4	6	11	BUSY
BUSY	11	6	D4
$\overline{\text{ACK}}$	10	5	D3
PAPER EMPTY	12	4	D2
SELECT	13	3	D1
$\overline{\text{ERROR}}$	15	2	D0
GND	25	25	GND

3.5. Giao tiếp 2 chiều



Giao tiếp song song hai chiều qua cổng SPP

Trong một số trường hợp, các chân ra của thanh ghi không dùng loại cực thu hở mà dùng cột totem chỉ có một chiều xuất, lúc đó có thể dùng bộ phân kênh để đọc 4 bit dữ liệu cào thanh ghi trạng thái, đọc hai lần sẽ được 8 bit dữ liệu.

Trong so đồ chân /Strobe được dùng để chọn nửa byte thấp khi ở mức 0 tức là bit D0 của thanh ghi điều khiển ở mức 1.

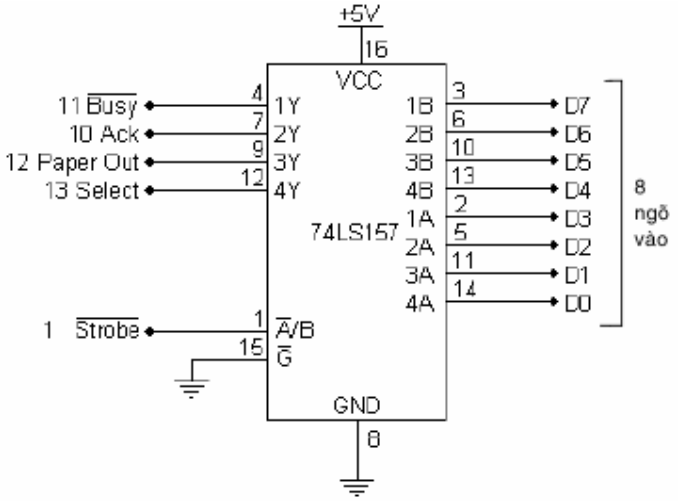
Với sơ đồ này việc xuất nhập dữ liệu 8 bit được thực hiện như sau:

1- Xác định địa chỉ gốc cổng LPT

2- Muốn xuất dữ liệu, ghi dữ liệu 8 bit vào thanh ghi dữ liệu, muốn xuất nhiều hơn 8 bit có thể dùng kèm các tín hiệu Init, Select… bằng cách ghi vào thanh ghi điều khiển byte thích hợp.

3- Muốn đọc dữ liệu, ghi bit D0=1 vào thanh ghi diều khiển, đọc 4 bit cao của thanh ghi trạng thái (4 bit thấp của dữ liệu vào), kết hợp hai lần đọc ta được 1 byte rồi exclusive OR với 88H để đảo hai bit D7 và D3 (do Busy đảo).

3.6. Minh họa giao tiếp song song với TTL 74LS157



Minh họa giao tiếp song song với TTL 74LS157

4. Giao tiếp cổng EPP

4.1. Đặc tính của EPP

Cổng EPP là sản phẩm liên kết của Intel, Xircom và Zenith, có hai chuẩn là EPP1.7 và EPP1.9, vận tốc truyền từ 500 Kbytes/s đến 2 Mbytes/s nhờ sự hỗ trợ của phần cứng và kỹ thuật DMA.

Khi chuyển cổng song song sang chế độ EPP (vào mục Setup khi khởi động máy tính để đặt chế độ) các chân cổng sẽ mang tên gọi và ý nghĩa khác.

4.2. Sơ đồ chân của EPP

Chân	Tên	Vào/Ra	Chức năng
1	Ghi /Write	Ra	Khi thấp xuất dữ liệu ra Khi cao đọc vào
2 - 9	Data 0 - 7	Vào - Ra	Tuyến dữ liệu hai hướng
10	Ngắt Interrupt	Vào	Ngắt ở cạnh lên
11	Chờ /Wait	Vào	Dùng để bắt tay, chu kỳ EPP bắt đầu khi thấp, chấm dứt khi cao
12, 13, 15			Không dùng
14	/Data Strobe	Ra	Khi ở mức thấp truyền dữ liệu
16	/Reset	Ra	Reset, tích cực thấp
17	/Address Strobe	Ra	Khi ở mức thấp truyền địa chỉ
18 – 25			Mass

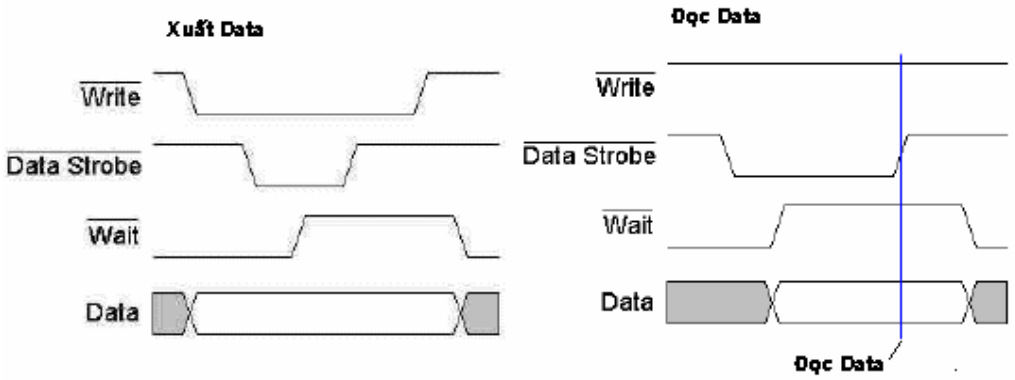
Cổng EPP có thêm một số thanh ghi như trong bảng sau:

Địa chỉ	Tên	Đọc / Ghi
Gốc + 0	Thanh ghi dữ liệu (SPP)	Ghi
Gốc + 1	Thanh ghi trạng thái (SPP)	Đọc
Gốc + 2	Thanh ghi điều khiển (SPP)	Ghi
Gốc + 3	Thanh ghi địa chỉ (EPP)	Đọc / Ghi
Gốc + 4	Thanh ghi dữ liệu (EPP)	Đọc / Ghi
Gốc + 5		
Gốc + 6	Dùng để truyền 16/32 bit	
Gốc + 7		

Ba thanh ghi đầu giống như trong SPP. Muốn truyền dữ liệu theo EPP ta đưa dữ liệu vào thanh ghi gốc +4 và mạch logic sẽ tạo các tín hiệu cần thiết.

4.3. Kết nối hai máy tính qua cổng EPP

4.3.1. Sơ đồ dây nối EPP



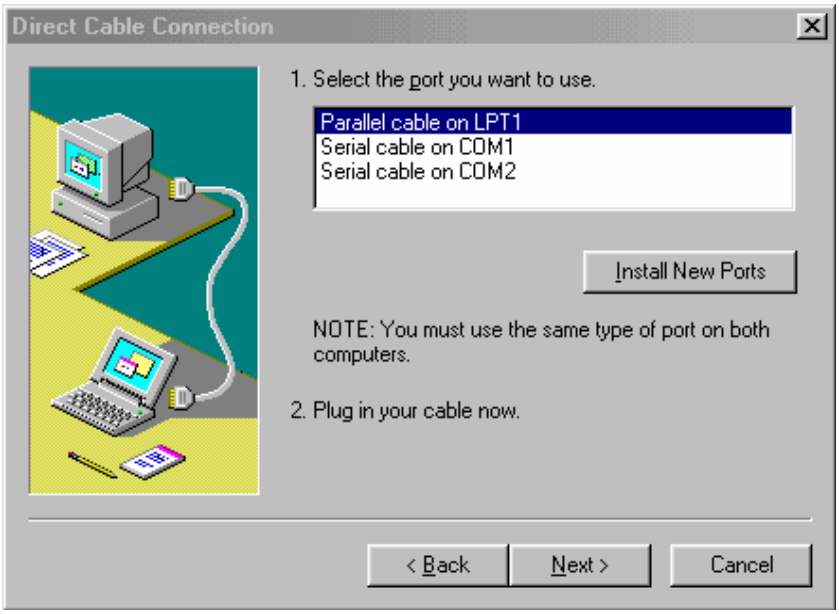
Chân /Write và Data Strobe tích cực thấp chờ cho đến khi /Wait lên mức cao báo bên kia đã nhận dữ liệu, sau đó /Data Strobe và /Write trở lại mức cao kết thúc truyền.

Khi nhận dữ liệu, đầu tiên chương trình đọc thanh ghi gốc +4. Nếu /Write mức thấp thì /Data Strobe sẽ ở mức thấp chờ khi /Wait ở mức cao bên kia báo đã gửi dữ liệu tới, /Data Strobe sẽ ở mức cao cà dữ liệu được đọc vào. Chu kỳ xuất và nhập dữ liệu cũng giống như chu kỳ xuất và nhập địa chỉ. Thường kết hợp địa chỉ và dữ liệu để truy xuất dữ liệu từ các địa chỉ ngoại vi khác nhau.

Phần cứng ngoại vi có nhiệm vụ xử lý các chân /Write, /Data Strobe, /Address Strobe đưa vào các mạch cài hay cho phép và xuất tín hiệu /Wait phù hợp. Thông qua tín hiệu Interrupt (chân sô 10) ngoại vi sẽ tác động đến máy tính, bit 0 của thanh ghi trạng thái là bit báo hết thời gian EPP. Nếu trong khoảng 10µs đường /Wait không tác động khi /Data Strobe hay /Address strobe tác động thì bit này sẽ được đặt lên 1.

4.3.2. Triển khai kết nối 2 máy tính (laplink)

Hai máy tính có thể ghép với nhau qua cổng song song hay cổng nối tiếp để truyền số liệu thông qua tiện ích Direct Cable Connection của hệ điều hành Windows hay tiện ích tương tự của phần mềm Norton Commander.



Giao diện ghép nối hai máy tính

- Ghép nối dùng cáp nối với hai đầu cực DB25, giao tiếp 4 bit

	Chân		Chân
D0	2	→	15
D1	3	→	13
D2	4	→	12
D3	5	→	10
D4	6	→	11
ACK	10	←	5
Busy	11	←	6
Paper out	12	←	4

Select 13 ← 3

Error 15 ← 2

GND 25 ← 25

- Giao tiếp dùng cáp nối 2 đầu cái DB9 hay DB25

	9 chân	25 chân	25 chân	9 chân
GND	5	7	7	5
TXD	3	2	3	2
RTS	7	4	5	8
DSR	1, 6	6	20	4
RXD	2	3	2	3
CTS	8	5	4	7
DTR	4	20	6	1 , 6

Truyền bằng đường song song nhanh gấp tám đến mười lần truyền nối tiếp.

Nếu cổng song song hai máy có cấu hình ECP thì vận tốc truyền còn nhanh hơn nữa, dưới đây là cáp nối với cổng ECP

Chân		Chân	Chân		Chân
1	→	10	15	←	17
2 ÷ 9	←>	2 ÷ 9	14	→	11
10	←	1	16	→	12
11	←	14	17	→	15
12	←	16	18 ÷ 25	<→	18 ÷ 25
13	←	13			

5. Giao tiếp cổng ECP

5.1. Đặc tính của ECP

Cổng ECP được phát triển bởi Hewlett Packard và Microsoft, sử dụng phần cứng hỗ trợ cho việc truyền dữ liệu nên có vận tốc truyền nhanh, tương tự cổng EPP. Đặc điểm của cổng ECP là nén dữ liệu khi truyền, cho phép tăng tốc độ truyền dữ liệu. Cổng ECP dùng 11 thanh ghi từ +0 đến gốc +7 và gốc +400H đến gốc +420H.

5.2. Sơ đồ chân của ECP

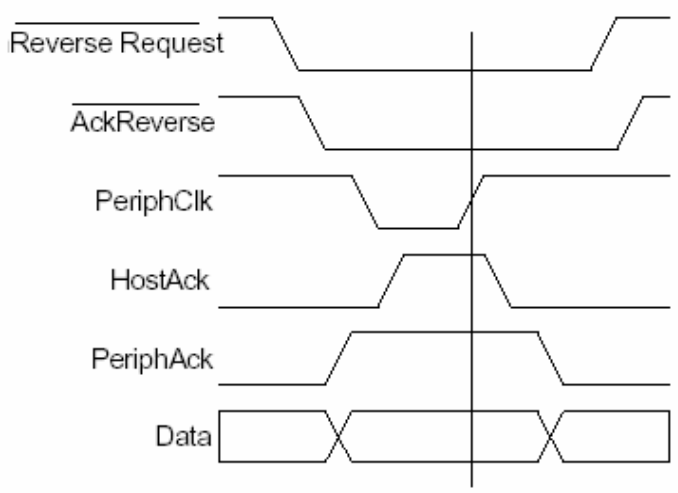
Chân cổng ECP được qui định trong bảng sau:

Chân	Tín hiệu ECP	Hướng	Chức năng
1	HostClk	Ra	Thấp cho biết có dữ liệu, cạnh lên dùng để cài dữ liệu ra
2 – 9	Data 0 – 7	Hai chiều	Tuyến dữ liệu hai chiều
10	PeriphClk	Vào	Thấp cho biết ngoại vi cần đưa dữ liệu vào máy tính, cạnh lên dùng để cài dữ liệu vào
11	PeriphAck	Vào	Khi ngoại vi nhận dữ liệu thì lên mức cao. Nếu ngoại vi gởi dữ liệu thì mức cao báo chu kỳ dữ liệu, mức thấp báo chu kỳ lệnh
12	/AckReverse	Vào	Khi thấp ngoại vi chấp nhận truyền dữ liệu
13	X – Flag	Vào	Cờ mở rộng
14	HostAck	Ra	Ở mức cao báo chu kỳ dữ liệu, ở mức thấp báo chu kỳ lệnh
15	PeriphRequest	Vào	Yêu cầu máy tính nhận số liệu
16	/ ReverseRequest	Ra	Khi ở mức thấp cho biết dữ liệu nhập vào máy tính
18 – 29			Mass

Khi truyền dữ liệu từ máy tính ra ngoại vi thì HostAck sẽ đổi mức, còn truyền dữ liệu từ ngoại vi vào máy tính thì PeriphAck đổi mức. Trên tuyến dữ liệu có thể truyền dữ liệu hay lệnh. Nếu Host Ack và PeriphAck ở mức cao là truyền dữ liệu. Nếu máy tính gửi lệnh, Host Ack mức thấp; nếu ngoại vi gửi lệnh, Periph Ack mức thấp. lệnh gồm 2 loại, nếu bit 7 của tuyến dữ liệu (chân 9) ở mức thấp thì 7 bit còn lại dùng để cho biết thông tin về nén dữ liệu. Nếu bit 7 ở mức cao thì 7 bit còn lại là địa chỉ kênh. Khi truyền dữ liệu nén, đầu tiên truyền số lần lặp lại của byte dữ liệu, sau đó truyền byte dữ liệu, ví dụ truyền 25 byte ký tự ‘A’ thì gửi byte 24 (*Run length count*) sau đó gửi byte ‘A’.

5.3. Kết nối 2 máy tính qua cổng ECP

5.3.1. Sơ đồ dây nối ECP



Truyền dữ liệu từ ngoại vi vào máy tính

Ngoại vi nhận được byte 24 trong chu kỳ lệnh sẽ lặp lại byte 'A' ở chu kỳ dữ liệu 25 lần. Tỷ số nén tối đa 64/1.

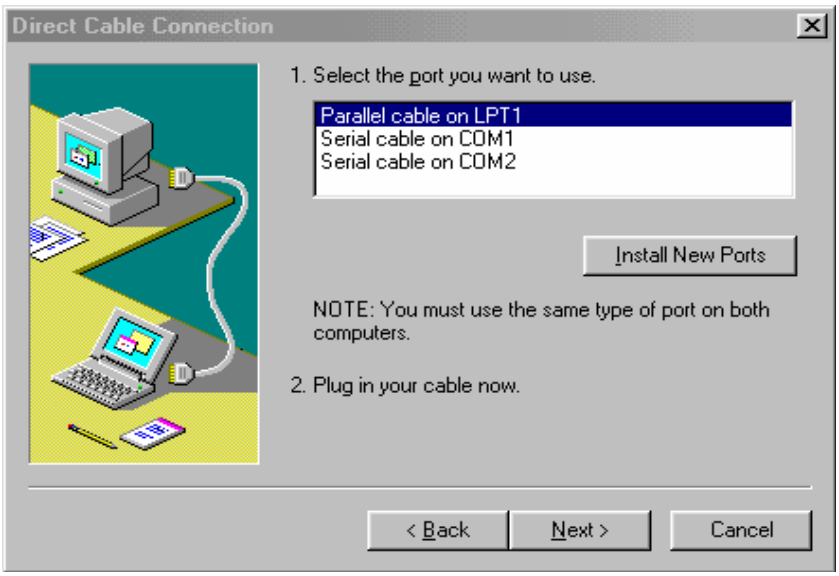
Để tăng tốc độ truyền dữ liệu, cổng ECP dùng các thanh ghi sắp xếp kiểu FIFO và một số thanh ghi phụ.

Địa chỉ	Tên	Đọc / Ghi
Gốc + 0	Thanh ghi dữ liệu (SPP)	Ghi
	Địa chỉ FIFO (ECP)	Đọc / Ghi
Gốc + 1	Thanh ghi trạng thái (mọi mode)	Đọc / Ghi
Gốc + 2	Thanh ghi điều khiển (mọi mode)	Đọc / Ghi
Gốc + 400 H	Data FIFO (mode FIFO cổng song song)	Đọc / Ghi
	Data FIFO (ECP)	Đọc / Ghi
	Test FIFO (mode Test)	Đọc / Ghi
	Thanh ghi cấu hình A	Đọc / Ghi
Gốc + 401 H	Thanh ghi cấu hình B	Đọc / Ghi
Gốc + 402 H	Thanh ghi điều khiển mở rộng	Đọc / Ghi

Thanh ghi điều khiển mở rộng cho phép chọn kiểu hoạt động của cổng song song. Thanh ghi cấu hình A và B sử dụng để đặt cấu hình của cổng ECP.

5.3.2. Triển khai kết nối 2 máy tính

Hai máy tính có thể ghép với nhau qua cổng song song hay cổng nối tiếp để truyền số liệu thông qua tiện ích Direct Cable Connection của hệ điều hành Windows hay tiện ích tương tự của phần mềm Norton Commander.



Giao diện ghép nối hai máy tính

- Ghép nối dùng cáp nối với hai đầu cực DB25, giao tiếp 4 bit

	Chân		Chân
D0	2	→	15
D1	3	→	13
D2	4	→	12
D3	5	→	10
D4	6	→	11
ACK	10	←	5
Busy	11	←	6
Paper out	12	←	4
Select	13	←	3
Error	15	←	2
GND	25	←	25

- Giao tiếp dùng cáp nối 2 đầu cái DB9 hay DB25

	9 chân	25 chân	25 chân	9 chân
GND	5	7	7	5
TXD	3	2	3	2
RTS	7	4	5	8
DSR	1, 6	6	20	4
RXD	2	3	2	3
CTS	8	5	4	7
DTR	4	20	6	1 , 6

Truyền bằng đường song song nhanh gấp tám đến mười lần truyền nối tiếp.

Nếu cổng song song hai máy có cấu hình ECP thì vận tốc truyền còn nhanh hơn nữa, dưới đây là cáp nối với cổng ECP

Chân		Chân	Chân		Chân
1	→	10	15	←	17
2 ÷ 9	←>	2 ÷ 9	14	→	11
10	←	1	16	→	12
11	←	14	17	→	15
12	←	16	18 ÷ 25	<→	18 ÷ 25
13	←	13			

CHƯƠNG 3: GIAO TIẾP QUA CỔNG NỐI TIẾP (SERIAL)

1. Các đặc tính của cổng giao tiếp nối tiếp (COM)
   1.1. Phương thức giao tiếp nối tiếp
   1.2. Đặc tính về tốc độ
   1.3. Đặc tính chống nhiễu
2. Cấu trúc cổng COM
   2.1. Sơ đồ chân cổng kiểu DB-9
   2.2. Sơ đồ chân cổng kiểu DB-25
   2.3. Sơ đồ khối kết nối cổng COM
3. Mạch giao tiếp nối tiếp
   3.1. Mạch chuyển RS-232
   3.2. Lập trình điều khiển nối tiếp đơn giản
   3.3. Giới thiệu mạch chuyển AD qua cổng nối tiếp.

Mục tiêu:

- Trình bày được về chuẩn giao tiếp nối tiếp.
- Trình bày cấu trúc giao tiếp, phục vụ cho môn học 'Lập trình ghép nối máy tính' sau này.
- Thực hiện kết nối 2 máy tính qua cổng nối tiếp
- Ý thức học tập, nghiên cứu cao
- Chính xác, cẩn thận, tỉ mỉ khi thực hiện giao tiếp qua cổng nối tiếp.

1. Các đặc tính của cổng giao tiếp nối tiếp (COM)

1.1. Phương thức giao tiếp nối tiếp

Cổng nối tiếp trên máy tính, thường gọi là cổng COM, được sử dụng để truyền dữ liệu hai chiều giữa máy tính và ngoại vi, có các ưu điểm sau:

- Khoảng cách truyền dài hơn so với cổng song song. Cổng nối tiếp truyền mức 1 từ −3V đến −25V và mức 0 từ +3V đến +25V nên tính chống nhiễu cao hơn, cho phép khoảng cách truyền

- xa hơn.

- Số dây kết nối ít, tối thiểu ba dây.

- Có thể ghép với đường dây điện thoại, cho phép khoảng cách truyền chỉ bị giới hạn bởi mạng tổng đài điện thoại.

- Có thể truyền không dây dùng tia hồng ngoại.

- Ghép nối dễ dàng với vi điều khiển hay PLC.

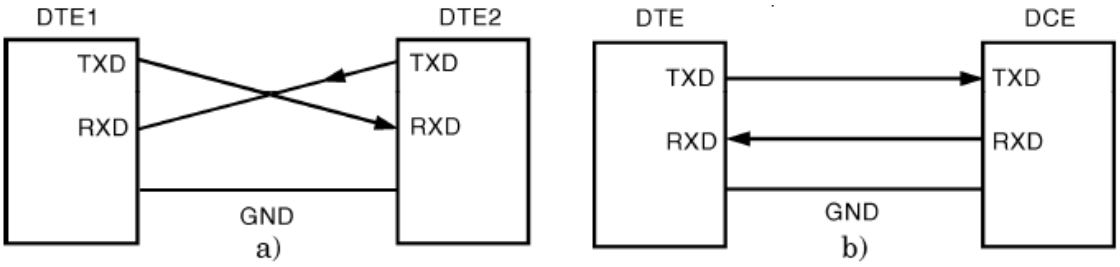
- Cho phép nối mạng.

Các thiết bị ghép nối nối tiếp chia làm hai loại DTE *(Data Terminal Equipment)* và DCE *(Data Communication Equipment)* DCE là các thiết bị trung gian như modem, còn DTE là các thiết bị như máy tính, vi điều khiển, PLC, là nguồn tạo ra dữ liệu hay tiếp nhận dữ liệu để xử lý. Có thể ghép nối DTE với DTE hoặc DCE, DCE với DTE hoặc DCE. Tín hiệu truyền nối tiếp theo dạng xung chuẩn RS 232 của EIA (Electronics Industry Associations), mức logic 0 còn gọi là *Space* giữa +3 và +25V, mức logic 1 còn gọi là *Mark*, ở giữa –3V và –25V.

Từ DTE tín hiệu được truyền giữa hai dây TXD và GND theo khuôn dạng sau:

Khi không truyền đường dây sẽ ở trạng thái Mark, khi bắt đầu truyền, xung Start được truyền (+10 V) sau đó là 8 bit dữ liệu, bit D0 được truyền trước, nếu bit dữ liệu logic 0 thì điện áp đường dây tương ứng là +10V, sau các bit dữ liệu là bit kiểm tra chẵn lẻ rồi bit stop ở logic 1(-10V), DTE nhận tín hiệu truyền ngược trở lại theo đường RXD. Nếu nối hai DTE với nhau thì dùng sơ đồ trên.

Trường hợp nối DTE với DCE thì chân TXD của DCE nhận tín hiệu còn chân RXD phát tín hiệu (nối 1-1).



Cổng COM có hai dạng đầu nối đực D-25 và D-9.

D-25	D-9	Tên tín hiệu	Chiều	Ý nghĩa
2	3	TD, TXD, truyền dữ liệu	xuất	Xuất dữ liệu nối tiếp
3	2	RD, RXD, nhận dữ liệu	nhập	Nhập dữ liệu nối tiếp
4	7	RTS, Request to send	xuất	DTE sẵn sàng trao đổi dữ liệu
5	8	CTS, Clear to send	nhập	Modem sẵn sàng trao đổi dữ liệu
6	6	DSR, Data set ready	nhập	Modem sẵn sàng kết nối
7	5	SG, Signal grourd	Mass	
8	1	CD, Carrier detect, phát giác sóng mang	nhập	Phát giác có tín hiệu trên đường dây
20	4	DTR, Data terminal Ready	xuất	DTE sẵn sàng kết nối
22	9	RI, Ring Indicator	nhập	Modem phát giác tín hiệu chuông

1.2. Đặc tính về tốc độ

Cổng nối tiếp RS232 là một giao diện phổ biến rộng rãi nhất. Người ta còn gọi cổng này là cổng COM1, còn cổng COM2 để tự do cho các ứng dụng khác.

Giống như cổng máy in cổng COM cũng được sử dụng một cách thuận tiện cho việc giao tiếp với thiết bị ngoại vi.

Việc truyền dữ liệu qua cổng COM được tiến hành theo cách nối tiếp. Nghĩa là các bit dữ liệu được truyền đi nối tiếp nhau trên một đường dẫn. Loại truyền này có khả năng dùng cho những ứng dụng có yêu cầu truyền khoảng cách lớn hơn, bởi vì các khả năng gây nhiễu là nhỏ đáng kể hơn khi dùng một cổng song song (cổng máy in).

Cổng COM không phải là một hệ thống bus nó cho phép dễ dàng tạo ra liên kết dưới hình thức điểm với điểm giữa hai máy cần trao đổi thông tin với nhau, một thành viên thứ ba không thể tham gia vào cuộc trao đổi thông tin này.

Phích cắm COM có tổng cộng 8 đường dẫn, chưa kể đến đường nối đất. Trên thực tế có hai loại phích cắm, một loại 9 chân và một loại 25 chân. Cả hai loại này đều có chung một đặc điểm.

Việc truyền dữ liệu xảy ra ở trên hai đường dẫn. Qua chân cắm ra TXD máy tính gởi dữ liệu của nó đến KIT Vi điều khiển. Trong khi đó các dữ liệu mà máy tính nhận được, lại được dẫn đến chân RXD các tín hiệu khác đóng vai trò

như là tín hiệu hổ trợ khi trao đổi thông tin, và vì thế không phải trong mọi trường hợp ứng dụng đều dùng hết.

Vì tín hiệu cổng COM thường ở mức +12V, -12V nên không tương thích với điện áp TTL nên để giao tiếp KIT Vi điều khiển 8051 với máy tính qua cổng COM ta phải qua một vi mạch biến đổi điện áp cho phù hợp với mức TTL, ta chọn vi mạch MAX232 để thực hiện việc tương thích điện áp.

Chính vì từ – 3V tới 3V là phạm vi không được định nghĩa, trong trường hợp thay đổi giá trị logic từ thấp lên cao hoặc từ cao xuống thấp, một tín hiệu phải vượt qua quãng quá độ trong một thời gian ngắn hợp lý. Điều này dẫn đến việc phải hạn chế về điện dung của các thiết bị tham gia và của cả đường truyền. Tốc độ truyền dẫn tối đa phụ thuộc vào chiều dài của dây dẫn. Đa số các hệ thống hiện nay chỉ hỗ trợ với tốc độ 19,2 kBd (chiều dài cho phép 30 – 50 m).

1.3. Đặc tính chống nhiễu

Khả năng chống nhiễu của các cổng nối tiếp cao.

Cổng nối tiếp là một phần của PC ngay khi nó mới ra đời. Với mỗi cổng COM hoặc COMM trong PC là một cổng nối tiếp không đồng bộ được điều khiển bởi các UART. Mỗi cổng COM có thể có giao diện RS-232, RS-485 hoặc cổng có thể để dành cho một modem trong hoặc thiêt bị khác. Mỗi PC có thể có các dạng khác nhau của các cổng nối tiếp như USB, Firewire, và I2C nhưng chúng dùng các giao thức khác nhau và yêu cầu các thành phần khác nhau.

2. Cấu trúc cổng COM

2.1. Sơ đồ chân kiểu cổng DB-9

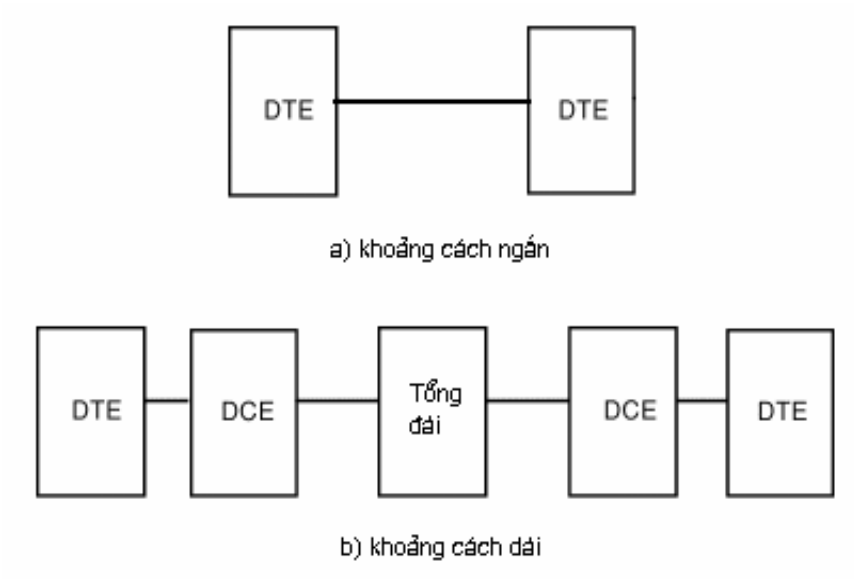
Chân D-9	Tên tín hiệu	Chiều	Ý nghĩa
1	CD, Carrier detect, phát giác sóng mang	Nhập	Phát giác có tín hiệu trên đường dây
2	RD, RXD, nhận dữ liệu	Nhập	Nhập dữ liệu nối tiếp
3	TD, TXD, truyền dữ liệu	Xuất	Xuất dữ liệu nối tiếp
4	DTR, Data terminal Ready	Xuất	DTE sẵn sàng kết nối
5	SG, Signal grourd	Mass	
6	DSR, Data set ready	Nhập	Modem sẵn sàng kết nối
7	RTS, Request to send	Xuất	DTE sẵn sàng trao đổi dữ liệu

8	CTS, Clear to send	Nhập	Modem sẵn sàng trao đổi dữ liệu
9	RI, Ring Indicator	Nhập	Modem phát giác tín hiệu chuông

2.2. Sơ đồ chân cổng kiểu DB-25

Chân D-25	Tên tín hiệu	Chiều	Ý nghĩa
2	TD, TXD, truyền dữ liệu	Xuất	Xuất dữ liệu nối tiếp
3	RD, RXD, nhận dữ liệu	Nhập	Nhập dữ liệu nối tiếp
4	RTS, Request to send	Xuất	DTE sẵn sàng trao đổi dữ liệu
5	CTS, Clear to send	Nhập	Modem sẵn sàng trao đổi dữ liệu
6	DSR, Data set ready	Nhập	Modem sẵn sàng kết nối
7	SG, Signal grourd	Mass	
8	CD, Carrier detect, phát giác sóng mang	Nhập	Phát giác có tín hiệu trên đường dây
20	DTR, Data terminal Ready	Xuất	DTE sẵn sàng kết nối
22	RI, Ring Indicator	Nhập	Modem phát giác tín hiệu chuông

2.3. Sơ đồ kết nối cổng COM



Tín hiệu truyền nối tiếp dưới dạng các bit, số bit trong mộtgiây được gọi là baud, vận tốc truyền thông dụng là 300, 600, 1200, 2400, 4800, 9600, 19200… baud. Nếu dùng vận tốc 9600 baud và khung truyền 8, E, 2 (8 bit dữ liệu, 1 bit kiểm tra chẵn, 2 bit stop) thì truyền một byte chiếm 12 bit vậy một giây truyền được 800 byte, thời gian truyền 1 bit là ~ 0,1msec. Các modem đời mới có thể đạt tốc độ 56000 baud, tuy nhiên các vi mạch truyền nối tiếp có thể đạt tốc độ cao hơn đến 115200 baud (vi mạch 16550) 230400 baud (16C650) vì vậy các modem phải nén tín hiệu trước khi truyền trên đường. Kết nối giữa máy tính (DTE) và modem (DCE) thực hiện theo nguyên tắc các chân cùng tên nối với nhau. Còn khi kết nối DTE và DTE thường dùng sơ đồ sau:

D9	D25			D25	D9
3	2	TD →	RD	3	2
2	3	RD ←	TD	2	3
5	7	SG	SG	7	5
4	20	DTR	DTR	20	4
6	6	DSR ←	→ DSR	6	6
1	8	CD ←	→ CD	8	1
7	4	RTS	RTS	4	7
8	5	CTS ←	→ CTS	5	8

Khi DTE cần truyền dữ liệu thì DTR tích cực đưa về DSR cho biết phía nhận sẵn sàng, đưa về CD cho biết đã nhận được sóng mang của modem ảo. Hai DTE có cùng khung truyền nên RTS và CTS nối với nhau. Đôi khi có thể bỏ đường nối DTR với DSR và CD.

Khi kết nối DTE với DCE, do vận tốc truyền khác nhau, cần điều khiển lưu lượng. Có hai cách là dùng phần cứng và phần mềm. Khi dùng phần cứng sử dụng hai dây RTS và CTS. Nếu DTE muốn truyền sẽ cho RTS tác động, nếu DCE chấp nhận sẽ gởi trở về CTS và máy tính sẽ gởi dữ liệu, nếu máy tính không nhận được CTS sẽ không gởi dữ liệu. Điều khiển lưu lượng bằng phần mềm dùng hai ký tự Xon và Xoff. Khi modem muốn máy tính ngừng truyền sẽ gởi đi ký tự Xoff (ASCII 19) còn khi modem rảnh nó sẽ gởi ký tự Xon (ASCII 17).

Việc trao đổi dữ liệu của máy tính được thực hiện thông qua vi mạch UART *(Universal Asynchronous Receiver Transmitter)* còn với vi điều khiển hay PLC thì có các vi mạch chuyên dụng hoặc được tích hợp trong vi xử lý. Các máy tính đời mới dùng công nghệ ASIC sử dụng chip đa năng làm nhiều nhiệm vụ giao tiếp nối tiếp, song song, cổng trò chơi, điều khiển đĩa, tuy nhiên phần giao tiếp nối tiếp thiết kế tương hợp với các vi mạch UART rời. Các loại vi mạch UART thường gặp là 8250, 8250A, 16450, 16550, 16650, 16750, … 6402.

Các cổng nối tiếp được đánh số COM 1, COM 2, COM 3, COM 4. Bảng sau cho địa chỉ gốc cổng COM và các thông tin khác.

Tên	Địa chỉ gốc	Ngắt	Nơi chứa địa chỉ
COM 1	3F8	4	0000 : 0400
COM 2	2F8	3	0000 : 0402
COM 3	3E8	4	0000 : 0404
COM 4	2E8	3	0000 : 0406

3. Mạch giao tiếp nối tiếp

3.1. Mạch chuyển đổi RS-232

Vấn đề giao tiếp giữa PC và vi điều khiển rất quan trọng trong các ứng dụng điều khiển, đo lường... Ghép nối qua cổng nối tiếp RS232 là một trong những kỹ thuật được sử dụng rộng rãi để ghép nối các thiết bị ngoại vi với máy tính.Nó là một chuẩn giao tiếp nối tiếp dùng định dạng không đồng bộ, kết nối nhiều nhất là 2 thiết bị , chiều dài kết nối lớn nhất cho phép để đảm bảo dữ liệu là 12.5 đến 25.4m, tốc độ 20kbit/s đôi khi là tốc độ 115kbit/s với một số thiết bị đặc biệt. Ý nghĩa của chuẩn truyền thông nối tiếp nghĩa là trong một thời điểm chỉ có một bit được gửi đi dọc theo đường truyền.

Có hia phiên bản RS232 được lưu hành trong thời gian tương đối dài là RS232B và RS232C. Nhưng cho đến nay thì phiên bản RS232B cũ thì ít được dùng còn RS232C hiện vẫn được dùng và tồn tại thường được gọi là tên ngẫn gọn là chuẩn RS232.

Các máy tính thường có 1 hoặc 2 cổng nối tiếp theo chuẩn RS232C được gọi là cổng Com. Chúng được dùng ghép nối cho chuột, modem, thiết bị đo lường...Trên main máy tính có loại 9 chân hoặc lại 25 chân tùy vào đời máy và main của máy tính. Việc thiết kế giao tiếp với cổng RS232 cũng tương đối dễ dàng, đặc biệt khi chọn chế độ hoạt động là không đồng bộ và tốc độ truyền dữ liệu thấp.

+ Khả năng chống nhiễu của các cổng nối tiếp cao.

+ Thiết bị ngoại vi có thể tháo lắp ngay cả khi máy tính đang được cấp điện.

+ Các mạch điện đơn giản có thể nhận được điện áp nguồn nuôi qua công nối tiếp.

+ Trong chuẩn RS232 có mức giới hạn trên và dưới (logic 0 và 1) là +-12V. Hiện nay đang được cố định trở kháng tải trong phạm vi từ 3000 ôm - 7000 ôm.

+ Mức logic 1 có điện áp nằm trong khoảng -3V đến -12V, mức logic 0 từ +-3V đến 12V.

+ Tốc độ truyền nhận dữ liệu cực đại là 100kbps (ngày nay có thể lớn hơn).

+ Các lối vào phải có điện dung nhỏ hơn 2500pF.

+ Trở kháng tải phải lớn hơn 3000 ôm nhưng phải nhỏ hơn 7000 ôm.

+ Độ dài của cáp nối giữa máy tính và thiết bị ngoại vi ghép nối qua cổng nối tiếp RS232 không vượt qua 15m nếu chúng ta không sử model.

+ Các giá trị tốc độ truyền dữ liệu chuẩn :

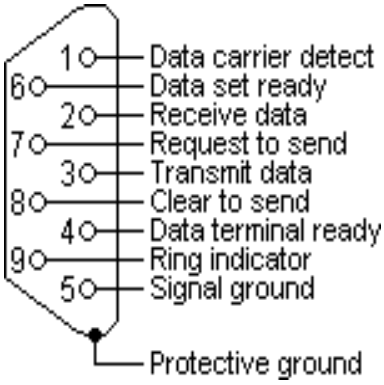
50,75,110,750,300,600,1200,2400,4800,9600,19200,28800,38400....56600,115200 bps.

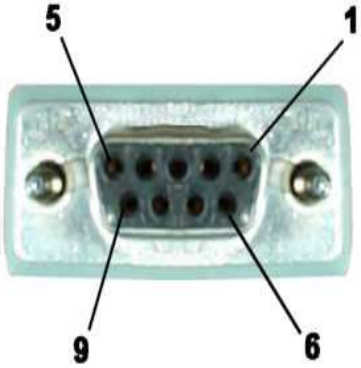
❖ **Cổng RS232 trên PC**

Hầu hết các máy tính cá nhân hiện nay đều được trang bị ít nhất là 1 cổng Com hay cổng nối tiếp RS232. Số lượng cổng Com có thể lên tới 4 tùy từng loại main máy tính. Khi đó các cổng Com đó được đánh dấu là Com 1, Com 2, Com 3...Trên đó có 2 loại đầu nối được sử dụng cho cổng nối tiếp RS232 loại 9 chân

(DB9) hoặc 25 chân (DB25). Tuy hai loại đầu nối này có cùng song song nhưng hai loại đầu nối này được phân biệt bởi cổng đực (DB9) và cổng cái (DB25).

Ta xét sơ đồ chân cổng Com 9 chân:





Trên là các kí hiệu chân và hình dạng của cổng DB9 Chức năng của các chân như sau:

+ chân 1 : Data Carrier Detect (DCD) : Phát tín hiệu mang dữ liệu.

+ chân 2: Receive Data (RxD) : Nhận dữ liệu.

+ chân 3 : Transmit Data (TxD) : Truyền dữ liệu.

+ chân 4 : Data Termial Ready (DTR) : Đầu cuối dữ liệu sẵn sàng được kích hoạt bởi bộ phận khi muốn truyền dữ liệu.

+ chân 5 : Singal Ground (SG) : Mass của tín hiệu.

+ chân 6 : Data Set Ready (DSR) : Dữ liệu sẵn sàng, được kích hoạt bởi bộ truyền khi nó sẵn sàng nhận dữ liệu.

+ chân 7 : Request to Send : yêu cầu gửi,bô truyền đặt đường này lên mức hoạt động khi sẵn sàng truyền dữ liệu.

+ chân 8 : Clear To Send (CTS) : Xóa để gửi ,bô nhận đặt đường này lên mức kích hoạt động để thông báo cho bộ truyền là nó sẵn sàng nhận tín hiệu.

+ chân 9 : Ring Indicate (RI) : Báo chuông cho biết là bộ nhận đang nhận tín hiệu rung chuông.

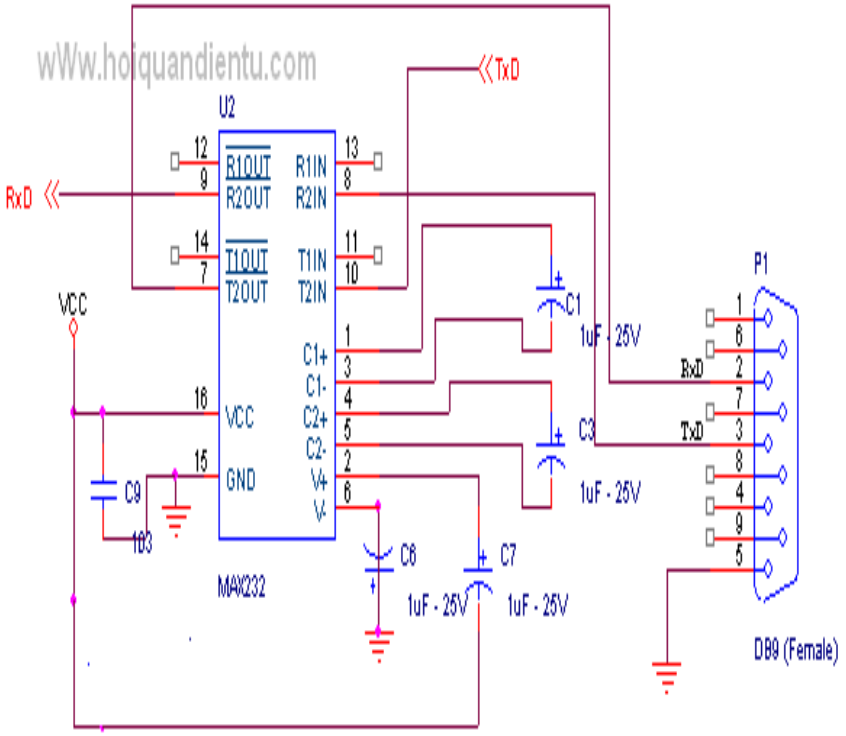
Còn DB28 bây giờ hầu hết các main mới ra đều không có cổng này nữa. Nên tôi không đề cập đến ở đây.

❖ **Mạch chuẩn giao RS232 dùng IC Max232**

Max232 là IC chuyên dùng cho giao tiếp giữa RS232 và thiết bị ngoại vi. Max232 là IC của hãng Maxim. Đây là IC chay ổn định và được sử dụng phổ biến trong các mạch giao tiếp chuẩn RS232. Giá thành của Max232 phù hợp (12K hay 10K) và tích hợp trong đó hai kênh truyền cho chuẩn RS232. Dòng

tín hiệu được thiết kế cho chuẩn RS232 . Mỗi đầu truyền ra và cổng nhận tín hiệu đều được bảo vệ chống lại sự phóng tĩnh điện (hình như là 15KV). Ngoài ra Max232 còn được thiết kế với nguồn +5V cung cấp nguồn công suất nhỏ.

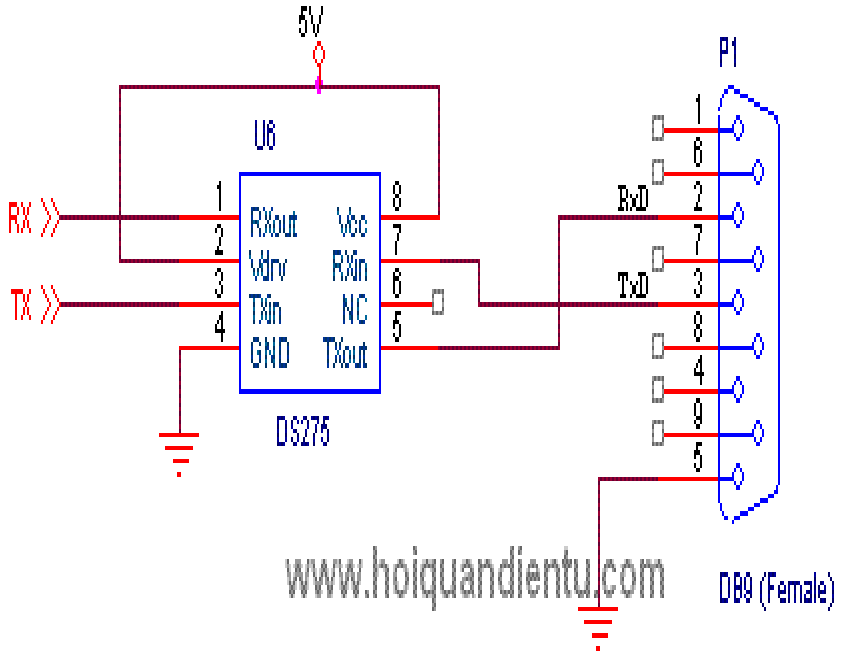
Mạch giao tiếp như sau :



Đây là mạch giao tiếp 1 kênh dùng Max232. Còn giao tiếp 2 kênh thì tương tự. Mạch này được sử dụng khá nhiều trong chuẩn giao tiếp RS232.

❖ **Mạch chuẩn giao tiếp RS232 dùng DS275**

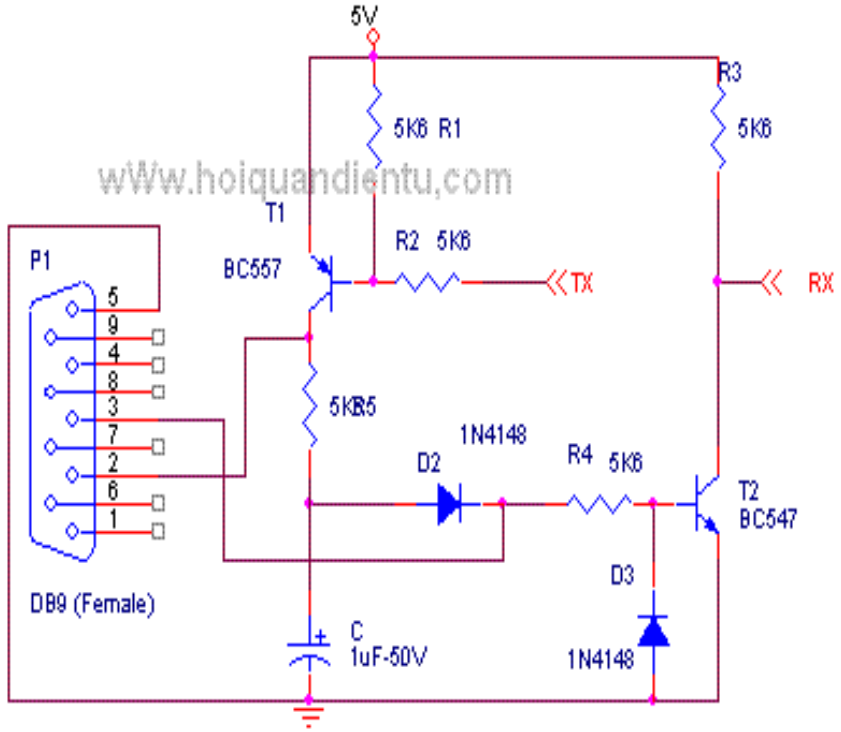
Đây cũng là IC của hãng Maxim. DS275 được dùng trong các mạch giao tiếp của chuẩn RS232 nhưng do nó chỉ là bán song công và dùng trong các thiết kế công suất nhỏ.



Mạch giao tiếp khá đơn giản. Do bán song công nên trong các ứng dụng ít được dùng.

❖ **Mạch chuẩn giao tiếp RS232 dùng transitor**

Mạch sử dụng 2 transior để giao tiếp RS232.



3.2. Lập trình điều khiển nối tiếp đơn giản

/* Chương trình truyền nối tiếp dùng ngôn ngữ C

```
/* Xuất kỳ tự nhận được ra màn hình và đọc ký tự từ bàn phím gởi nối tiếp */
#include <dos.h>
#include <stdio.h>
#include<conio.h>
#define PORT1 0x3F8
/* COM 1 0x3F8 */
/* COM 2 0x2F8 */
/* COM 3 0x3E8 */
/* COM 4 0x2E8 */
void main (void)
{
int C;
int ch;
outportb (PORT1 + 1, 0); /* Cấm ngắt Port 1 */
/* Đặt cấu hình PORT 1 */
outportb (PORT1 + 3, 0x80); /* SET DLAB ON */
outportb (PORT1 + 0, 0x03); /* Set Baud rate - Divisor Latch Low Byte */
/* Default 0x03 = 38,400 BPS */
/* 0x01 = 115,200 BPS */
/* 0x02 = 56,700 BPS */
/* 0x06 = 19,200 BPS */
/* 0x0C = 9,600 BPS */
/* 0x18 = 4,800 BPS */
/* 0x30 = 2,400 BPS */
outportb (PORT1 + 1 , 0x00); /* Set Baud rate - Divisor Latch Hight Byte */
outportb (PORT1 + 3 , 0x03); /* 8 Bits, No Parity, 1 Stop Bit */
outportb (PORT1 + 2 , 0xC7); /* FIFO Control Register */
outportb (PORT1 + 4 , 0x0B); /* Turn on DTR, RTS, and OUT2 */
```

```
printf ("\n. Press ESC to quit \n");
do { c = inportb (PORT1 + 5); /* Xem có nhận được ký tự không. */
if (c & 1) {ch = inportb (PORT1);
printf ("%c" ch);} /* Xuất ký tự ra màn hình */
if (kbhit ()) {ch = getch (); /* Đọc phím bấm */
outportb (PORT1, ch);} /* Gởi ký tự */
} while (ch != 27);
}
```

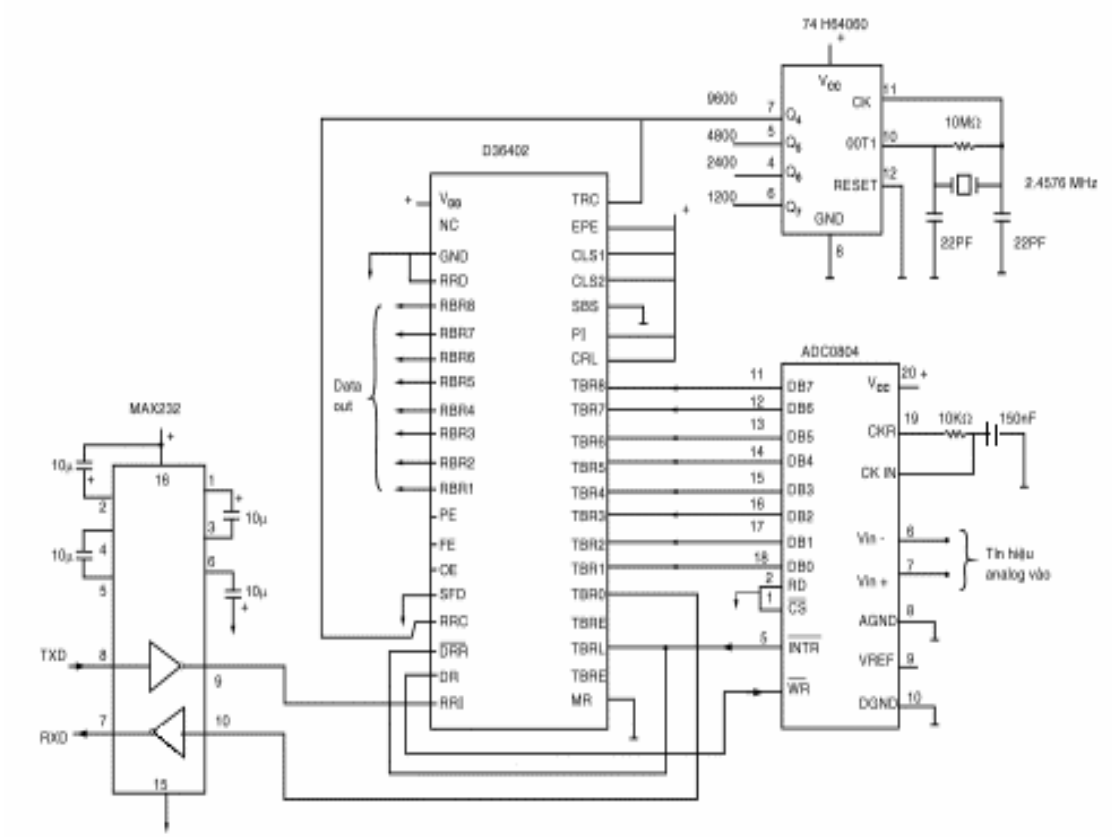
/* Chương trình giao tiếp nối tiếp dùng ngắt*/

```
#include <dos.h>
#include <stdio.h>
#include <conio.h>
#define PORT1 0x2E8
#define INTVECT 0x08 /* Com Port's IRQ here */
int bufferin = 0;
int bufferout = 0;
char ch;
char buffer [1025];
void interrupt (*oldport1isr) ();
void interrupt PORT1INTO () /* Interrupt Service Routine (IRS) for PORT 1 */
{
int c;
do {c = inportb (PORT1 + 5);
if (c & 1) {buffer [bufferin] = inportb (PORT1);
bufferin++;
if (bufferin == 1024) bufferin = 0;}
} while (c & 1);
outportb (0x20, 0x20);
```

```
}
void main (void)
{
int c;
outportb (PORT1 + 1 , 0);
oldport1isr = getvect (INTVECT); /* cất vectơ ngắt cũ */
setvect (INTVECT, PORT1INT); /* đặt vectơ ngắt mới */
/* COM 1 - 0x0C */
/* COM 2 - 0x0B */
/* COM 3 - 0x0C */
/* COM 4 - 0x0B */
outportb (PORT1 + 3 , 0x80); /* SET DLAB ON */
outportb (PORT1 + 0 , 0x03); /* Set Baud rate - Divisor Latch Low Byte */
/* Default 0x03 = 38,400 BPS */
/* 0x01 = 115,200 BPS */
/* 0x20 = 56,700 BPS */
/* 0x06 = 19,200 BPS */
/* 0x0C = 9,600 BPS */
/* 0x18 = 4,800 BPS */
/* 0x30 = 2,400 BPS */
outportb (PORT1 + 1, 0x00); /* Set Baud rate - Divisor Latch Hight Byte */
outportb (PORT1 + 3, 0x03); /* 8 Bits, No Parity, 1 Stop Bit */
outportb (PORT1 + 2, 0xC7); /* FIFO Control Register */
outportb (PORT1 + 4, 0x0B); /* Turn on DTR, RTS, and OUT2 */
outportb (0x21, (inportb (0x21) & 0xF7)); /* Set Programmable Interrupt
Controller */
/* COM 1 (IRQ4) - 0xEF */
/* COM 2 (IRQ3) - 0xF7 */
/* COM 3 (IRQ4) - 0xEF */
/* COM 4 (IRQ3) - 0xF7 */
```

```
outportb (PORT1 + 1 , 0x01); /* Ngắt khi thu */
printf (“\n Press ESC to quit \n”);
do {
if (bufferin ! == bufferout) {ch = buffer [bufferout];
bufferout++;
if (bufferout == 1024) bufferout = 0;
printf (“%C”, ch);}
if (kbhit ()) {c = getch ();
outportb (PORT 1, c);}
} while (c ! = 27);
outportb (PORT 1 + 1 , 0); /* Turn off interrupts - Port 1 */
outportb (0x21, (inportb (0x21) □ 0x08)); /*MASK IRQ using PIC */
/* COM 1 (IRQ4) - 0x10 */
/* COM 2 (IRQ3) - 0x08 */
/* COM 3 (IRQ4) - 0x10 */
/* COM 4 (IRQ3) - 0x08 */
setvect (INTVECT, oldport 1 isr); /* hồi phục vectơ ngắt cũ */
}
```

3.3. Giới thiệu mạch chuyển AD qua cổng nối tiếp



Chuyển đổi AD dùng cổng nối tiếp 9600, 8, E, 1

Chương trình sau viết bằng ngôn ngữ C minh họa hoạt động của mạch.

/* Chương trình đổi AD nối tiếp */

```
#include <dos.h>
#include <stdio.h>
#include <conio.h>
#define PORT1 0x3F8
#define INTVECT 0x0C
int bufferin = 0;
int bufferout = 0;
unsigned char ch;
char buffer [1025];
void interrupt (*oldport1isr) ();
void interrupt PORT1INT /*IRS cho PORT1 */
{
```

```
int c;
do {c = inportb(PORT1 +5);
if (c & 1) {buffer[bufferin] = inportb(PORT1);
bufferin++;
if (bufferin == 1024) {bufferin = 0;}}
} while (c & 1);
outportb (0x10, 0x20);
}
void main (void)
{
int c;
outportb (PORT1 + 1, 0); /* Cấm ngắt port 1 */
oldport1isr = getvect (INTVECT);
setvect (INTVECT, PORT1INT);
/*PORT 1 - Đặt chế độ*/
outportb (PORT1 + 3, 0x80); /* SET DLAB ON */
outportb (PORT1 + 0, 0x0C); /* đặt Baud rate 9600 BPS */
outportb (PORT1 + 1, 0x00);
outportb (PORT1 + 3, 0x03); /* 8 bits, No Parity, 1 Stop Bit */
outportb (PORT1 + 2, 0xC7); /*FIFO Control Register*/
outportb (PORT1 + 4, 0x0B); /*Cho DTR, RTS, và OUT2 ON*/
outportb (0x21, (inportb (0x21) & 0xEF);
outportb (PORT1 + 1, 0x01); /* Interrupt khi nhận data */
printf ("\n\chương trình đổi ADC nối tiếp. Nhấn ESC để thoát \n");
do {
if (bufferin ! == bufferout) {ch = buffer[bufferout];
bufferout++;
if (bufferout == 1024) {bufferout = 0;}
print("%u\n", ch);
```

```
printf("%f volts\n", (float) ch/256*5);}
if (kbhit () {c = getch();
outportb(PORT1, c);}
} while (c != 27);
outportb (PORT1 + 1, 0);
outportb (0x21, (inportb (0x21) □ 0x20));
setvect (INTVECT, oldport1isr);
}
```

TÀI LIỆU THAM KHẢO

- TS Nguyễn Đức Thành. Đo lường và Điều khiển bằng máy tính. NXB: ĐH HCM. Năm 2006
- Ngô Diên Tập. Đo lường và Điều khiển bằng Máy tính. NXB : Khoa học và Kỹ Thuật Hà Nội.