

## GIAO TIẾP QUA CỔNG SONG SONG

Cổng song song thường dùng để giao tiếp máy vi tính với đối tượng bên ngoài nhờ các đặc điểm:

- Lập trình đơn giản, dễ kết nối
- Tốc độ nhanh.

Khuyết điểm của cổng song song là khoảng cách ngắn do tính chống nhiễu kém. Theo tiêu chuẩn IEEE 1284 năm 1994 có 5 chế độ hoạt động cho cổng song song.

- SPP :cổng song song chuẩn có 3 mode là:

- + Compatibility: xuất 8 bit
- + Nibble: nhập 4 bit
- + Byte: cổng hai chiều

- EPP: cổng song song tăng cường (*enhanced parallel port*)

- ECP: cổng mở rộng khả năng (*extended capability port*)

Tùy loại mainboard có thể hỗ trợ cả 5 mode hay chỉ vài mode.

Cổng SPP có thể truyền dữ liệu song song 8 bit từ máy tính ra với vận tốc 50 Kbytes/sec đến 150 Kbytes/sec. Khi muốn nhập dữ liệu vào máy tính có thể dùng mode Nibble truyền 4 bit hay mode Byte truyền 8 bit.

Cổng EPP và ECP dùng thêm phần cứng hỗ trợ nên vận tốc truyền nhanh hơn, có thể đến 2 Mbytes/sec, thu phát song song 8 bit.

Thay đổi chế độ của cổng song song bằng cách vào BIOS

SETUP của máy tính khi khởi động máy.

Các máy tính không có cổng song song có thể sử dụng chuyển đổi USB - Parallel Port hay PCMCIA (Personal Computer Memory Card International Association) - Parallel Port.



cuu duong than cong . com

Parallel PCMCIA Card SPP/ECP/EPP LPT Port True Printer Port CardBus

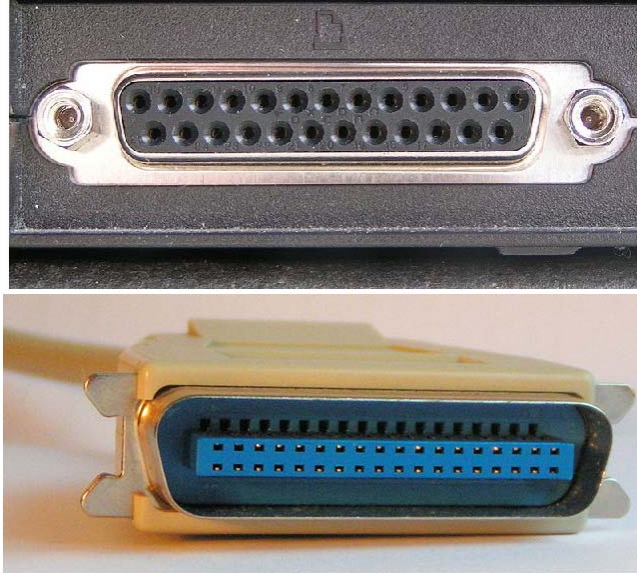
Ngoài nhiệm vụ giao tiếp, cổng song song còn dùng làm khóa bản quyền dongle, phần mềm có bảo vệ đọc mã khóa từ vi mạch trên dongle, nếu không có sẽ chỉ cho phép sử dụng ở chế độ giới hạn.



cuu duong than cong . com

### 6.1 Cổng SPP

Cổng song song có đầu nối 25 chân cái thường dùng để kết nối với máy in đầu nối Centronics 34 chân.



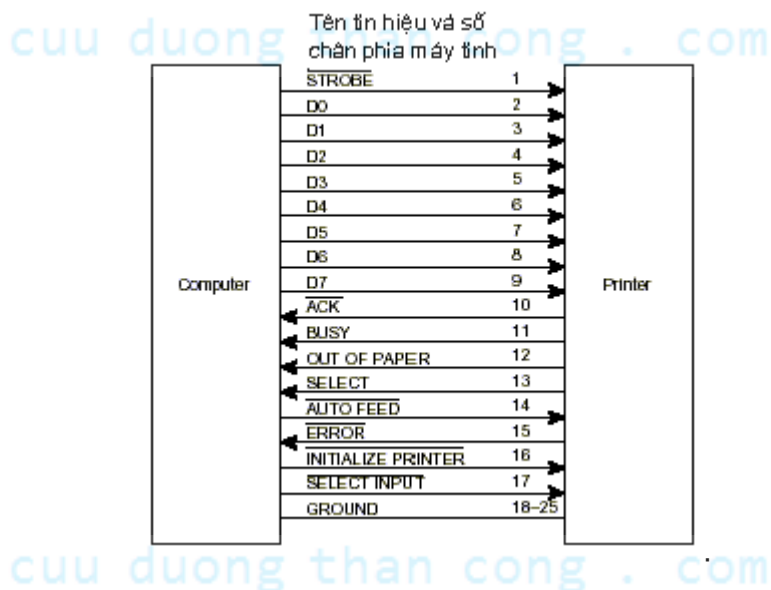
Bảng 6.1 cho sơ đồ chân và ý nghĩa các chân của cổng SPP khi dùng với máy in, dấu “/” có nghĩa tích cực thấp. Ví dụ, chân 15 là /Error hướng vào, nếu chân này xuống mức 0 là có lỗi. Cột Đảo ghi chữ Có tức là tín hiệu được đảo mức, ví dụ chân 17 khi đưa mức logic 0 ứng với chân này vào thanh ghi điều khiển thì ở chân 17 xuất hiện mức 1.

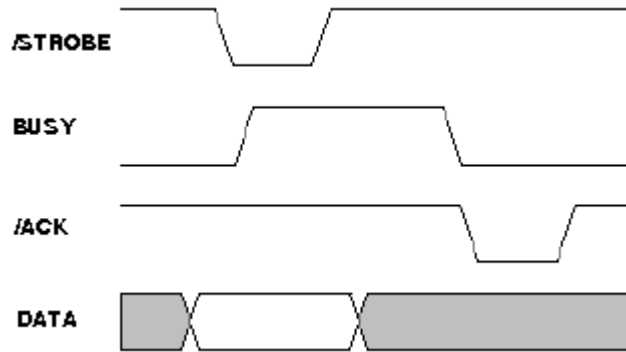
**Bảng 6.1:** Sơ đồ chân của đầu nối phía máy tính (cổng SPP) và phía máy in

| Chân<br>D – 25 | Chân<br>Centronics | Tín hiệu<br>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            |        |            |    |

Ngõ ra cổng song song tương thích với cổng TTL, dòng cấp và thu khoảng vài  $mA$  đến  $16mA$  tùy loại kết cấu phần cứng (công nghệ ASIC). Giao tiếp cổng song song với máy in thường được thực hiện theo H.6.1





Hình 6.1: Sơ đồ kết nối và bắt tay

Ban đầu dữ liệu 8 bit được đưa ra, máy tính đọc chân Busy nếu ở mức thấp thì máy in rảnh, sẽ đưa ra tín hiệu Strobe rộng 1μsec và máy in đọc dữ liệu, đọc xong sẽ báo trở lại bằng  $\overline{ACK}$  rộng 5μsec. Nếu máy in bận thì Busy sẽ ở mức cao.

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

378H cho LPT1 (line printer 1)

278H cho LPT2

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..

- 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.

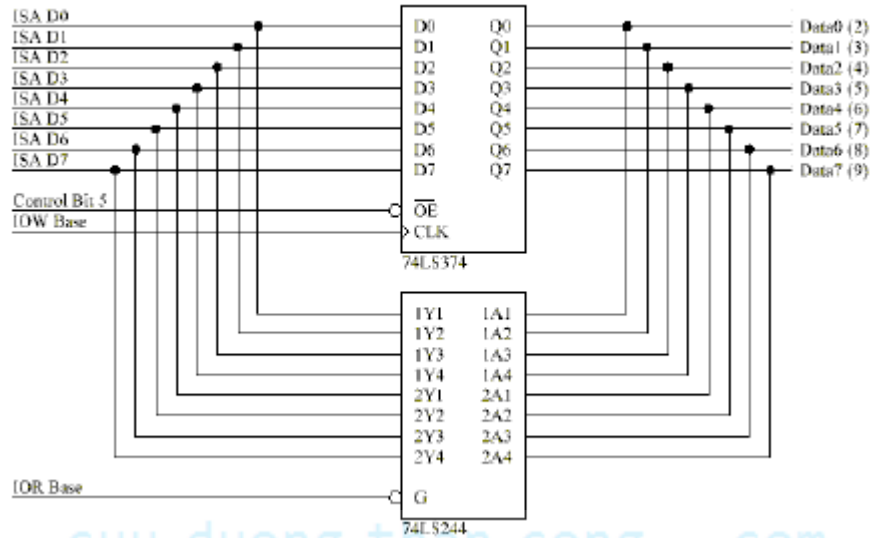
| D7   |      |           |        | D0     |      |   |   |
|------|------|-----------|--------|--------|------|---|---|
| Busy | /ACK | Paper Out | Select | /Error | /IRQ | X | X |
| 11   | 10   | 12        | 13     | 15     |      |   |   |

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

| D7 |   |                         |                      | D0                     |                           |                              |                |
|----|---|-------------------------|----------------------|------------------------|---------------------------|------------------------------|----------------|
| X  | X | Cho phép cổng hai chiều | Cho phép IRQ qua ACK | Chọn máy in /Select 17 | Khởi động máy in /Init 16 | Xuống hàng /Auto Linefeed 14 | Kích /Strobe 1 |

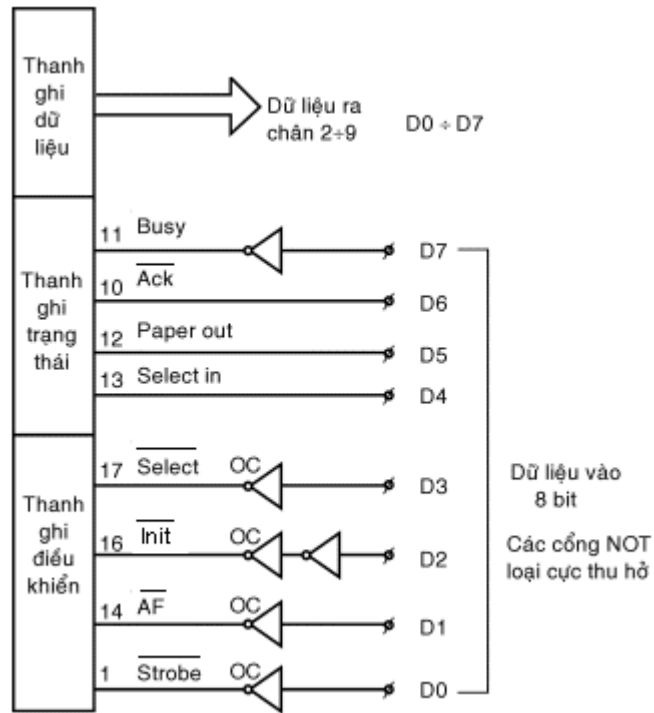
Các chân 1, 11, 14 và 17 được đảo phân cực, bit D6 thanh ghi trạng thái (chân số 10) từ 1 → 0 thì gây ra ngắt IRQ7 nếu được cho phép bởi D4 của thanh ghi điều khiển = 1.



**Hình 6.2** Sơ đồ thanh ghi dữ liệu hai hướng

Một số mainboard (H6.2) 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.

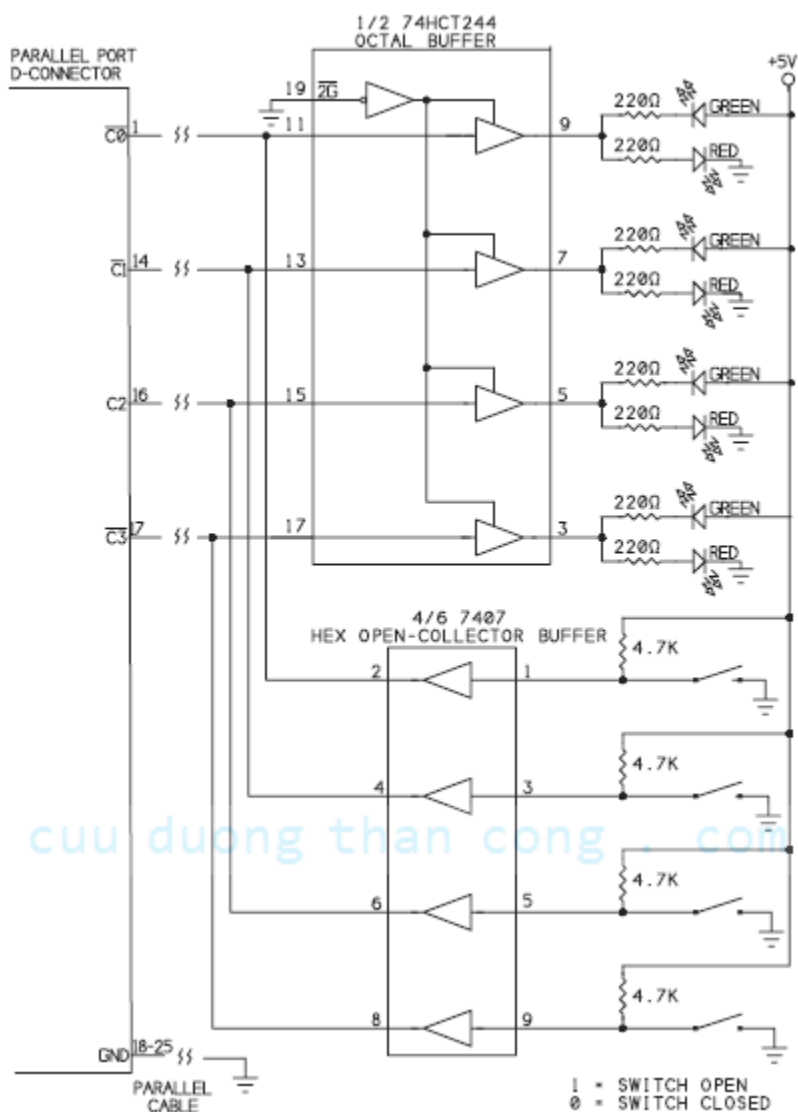
Các chân của thanh ghi điều khiển có ngõ ra cực thu hở nên có thể nhận tín hiệu vào nếu trước đó ta đã nạp 8 bit sao cho các ngõ ra ứng với thanh ghi này lên 1 (sơ đồ H 6.3). Do các tín hiệu /Busy, /Select, /AF và /Strobe đã được đảo phân cực nên ta thêm các cổng đảo, logic đọc vào phản ánh đúng mức tín hiệu.



**Hình 6.3:** Giao tiếp song song hai chiều qua cổng SPP

cuu duong than cong . com

cuu duong than cong . com



Trong một số trường hợp, các chân ra của thanh ghi điều khiển 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 vào thanh ghi trạng thái, đọc hai lần sẽ được 8 bit dữ liệu (H6.4).

Trong sơ 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

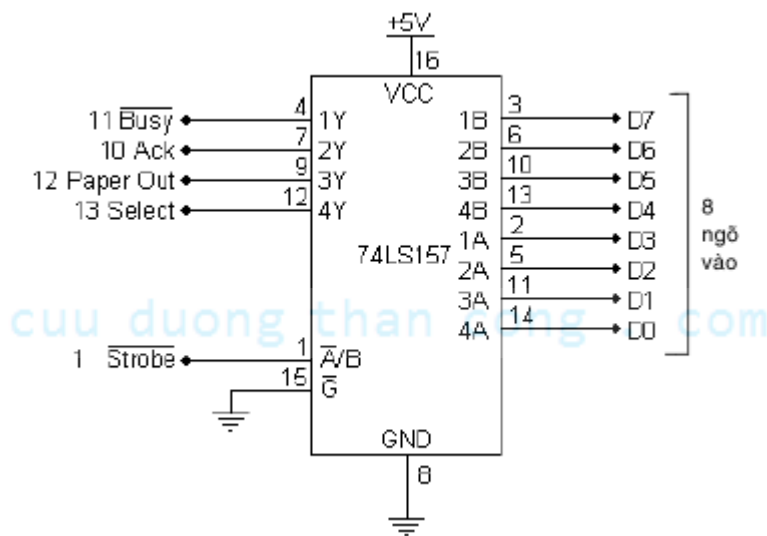


sau:

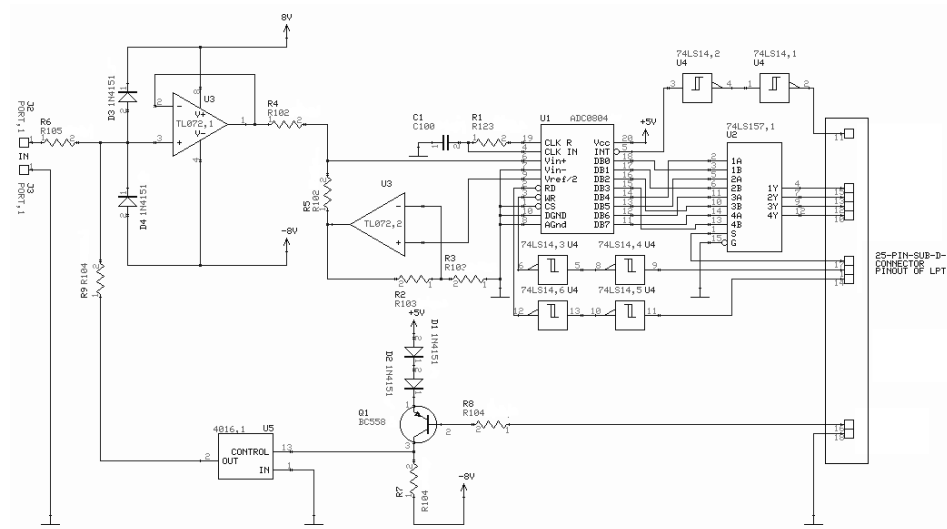
1- Xác định địa chỉ gốc của cổng LPT

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 điều khiển, đọc 4 bit cao của thanh ghi trạng thái (4 bit thấp của dữ liệu vào), rồi dời phải 4 bit, sau đó cho D0 của thanh ghi điều khiển bằng 0, đọc 4 bit cao của thanh ghi trạng thái (4 bit cao 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).



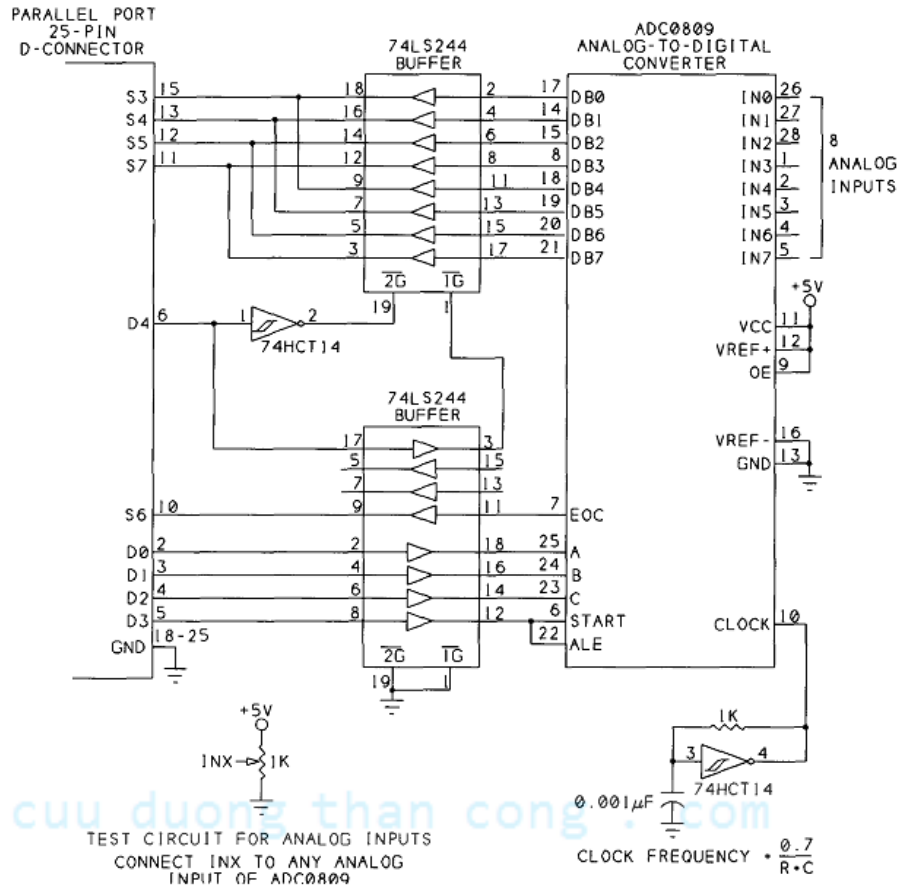
Hình 6.4: Vào 8 bit với 74LS157



Hình 6.5: Chuyển đổi AD dùng ADC0804

cuu duong than cong . com

cuu duong than cong . com



Hình 6.6 AD dùng ADC0809

**Ví dụ lập trình**

Xuất trị của biến oval1 ra thanh ghi dữ liệu và oval2 ra thanh ghi điều khiển, nhập giá trị inval từ thanh ghi trạng thái.

*Trong ngôn ngữ C*

```
# define DATA 0X378
# define STATUS DATA + 1
# define CONTROL DATA + 2
int oval1, oval2, inval;
oval1 = 0X81;
outportb (DATA, oval1);
oval2 = 0X08;
outportb (CONTROL, OVAL2 ^ 0x0b);
inval = ((inportb (STATUS) ^ 0X80);
```

*Trong ngôn ngữ BASIC*

```

DATA = &H378
STATUS = DATA + 1
CONTROL = DATA + 2
OVAL1 = 129
OUT DATA, OVAL1
OVAL2 = 75
OUT CONTROL, OVAL2 XOR 11
INVAL = ((INP (STATUS) XOR 128) AND &HF8) / 8

```

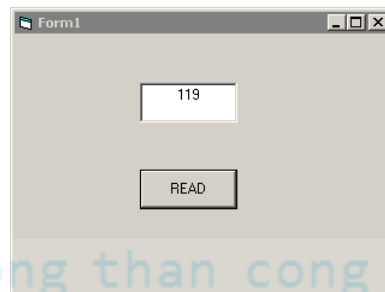
### Lập trình cho sơ đồ H6..4 đọc 8 bit dùng C

```

outportb(CONTROL, inportb(CONTROL) | 0x01); /* Chọn nửa byte thấp (A) */
a = (inportb(STATUS) & 0xF0); /* Đọc nửa byte thấp */
a = a >> 4; /* Dời phải 4 bit */
outportb(CONTROL, inportb(CONTROL) & 0xFE); /* Chọn nửa byte cao (B) */
a = a | (inportb(STATUS) & 0xF0); /* Đọc nửa byte cao */
byte = byte ^ 0x88;

```

### Lập trình cho sơ đồ H6..4 đọc 8 bit dùng VB



```

Private Declare Function Inp32 Lib "inout32.dll" (ByVal adr As Integer) As Integer
Private Declare Sub Out32 Lib "inout32.dll" (ByVal adr As Integer, ByVal data As Integer)
Private Sub cmdRead_Click()
Dim hbytethap, hbytecao, data
Out32 &H37A, 1
hbytethap = Inp32(&H379) And &HFF
hbytethap = hbytethap \ 16
Out32 &H37A, 0
hbytecao = Inp32(&H379) And &HF0
data = hbytecao Or hbytethap
data = data Xor &H88
txtIn.Text = data
End Sub

```

### Lập trình sơ đồ Hình 6.6

```

Private Declare Function Inp32 Lib "inout32.dll" (ByVal adr As Integer) As Integer
Private Declare Sub Out32 Lib "inout32.dll" (ByVal adr As Integer, ByVal data As Integer)
Private Declare Sub Sleep Lib "kernel32.dll" (ByVal delay As Long)

```

```

Private Sub Command1_Click()
Timer1.Enabled = 1

End Sub

Private Sub Command2_Click()
End
End Sub

Private Sub Form_Load()
Timer1.Interval = 1000
Timer1.Enabled = 0
End Sub

Private Sub Timer1_Timer()
For i = 0 To 7
    Out32 &H378, 8 + i
    Out32 &H378, i
    Sleep (1)
    dl = (Inp32(&H379) And &HFF) \ 8
    dl = (dl \ 16) * 8 + dl Mod 8
    Out32 &H378, 16
    dh = (Inp32(&H379) And &HFF) \ 8
    dh = (dh \ 16) * 8 + dh Mod 8
    data = (dh * 16 + dl) Xor &H88
    Text1(i).Text = Round(data * 5 / 256, 1)
Next
End Sub

```

### Sử dụng ngắt

Khi bit 4 của thanh ghi điều khiển ở mức 1 sẽ cho phép ngắt nếu chân ACK (số 9) chuyển từ 1 xuống 0, chương trình phục vụ ngắt IRQ7 sẽ được gọi (cũng có lúc ngắt bị tác động khi ACK chuyển từ 0 lên 1). Có thể viết một chương trình kiểm tra xem ngắt có tác động không. Về phần cứng chỉ cần nối chân 9 (D7) với chân 10 (ACK). Vào Control panel –System kiểm tra địa chỉ và số ngắt của cổng LPT.

```

#include <dos.h>
#define PORTADDRESS 0x378
#define IRQ 7
#define DATA PORTADDRESS+0

```

```

#define STATUS PORTADDRESS+1
#define CONTROL PORTADDRESS+2
#define PIC1 0x20
#define PIC2 0xA0
int interflag; /* Cờ ngắt */
int picaddr; /* Địa chỉ gốc của PIC) */
void interrupt (*oldhandler)();
void interrupt parISR() /* Interrupt Service Routine (ISR) */
{
    interflag = 1;
    outportb(picaddr,0x20); /* End of Interrupt (EOI) */
}
void main(void)
{
    int c;
    int intno; /* Số vectơ ngắt */
    int picmask; /* Mặt nạ */
    if (IRQ >= 2 && IRQ <= 7)
    {
        intno = IRQ + 0x08;
        picaddr = PIC1;
        picmask = 1;
        picmask = picmask << IRQ;
    }
    if (IRQ >= 8 && IRQ <= 15)
    {
        intno = IRQ + 0x68;
        picaddr = PIC2;
        picmask = 1;
        picmask = picmask << (IRQ-8);
    }
    if (IRQ < 2 || IRQ > 15)
    {
        printf("IRQ Out of Range\n");
        exit();
    }
    outportb(CONTROL, inportb(CONTROL) & 0xDF); /* Chiều xuất */
    outportb(DATA,0xFF);
    oldhandler = getvect(intno); /* Lưu vectơ ngắt cũ */
    setvect(intno, parISR); /* Đặt vectơ ngắt mới */
    outportb(picaddr+1,inportb(picaddr+1) & (0xFF - picmask)); /* Gõ mặt nạ */
    outportb(CONTROL, inportb(CONTROL) | 0x10); /* Cho phép IRQ */
    clrscr();
    printf("Parallel Port Interrupt Polarity Tester\n");
    printf("IRQ %d : INTNO %02X : PIC Addr 0x%X : Mask\n",IRQ,intno,picaddr,picmask);
    interflag = 0; /* Reset Interrupt Flag */
    delay(10);
    outportb(DATA,0x00); /* Cho ACK tác động */
    delay(10); /* Chờ */
    if (interflag == 1) printf("Interrupts Occur on High to Low Transition of ACK.\n");
    else
    {

```

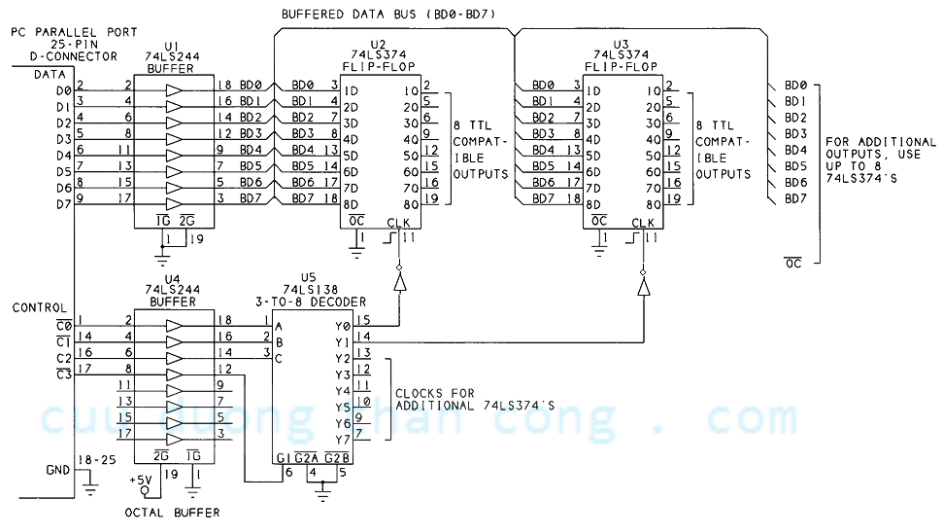
```

        outportb(DATA,0xFF); /* Low to High Transition */
        delay(10); /* wait */
        if (interflag == 1) printf("Interrupts Occur on Low to High Transition of
        ACK.\n");
        else printf("No Interrupt Activity Occurred. \nCheck IRQ Number, Port
        Address and Wiring.");
    }

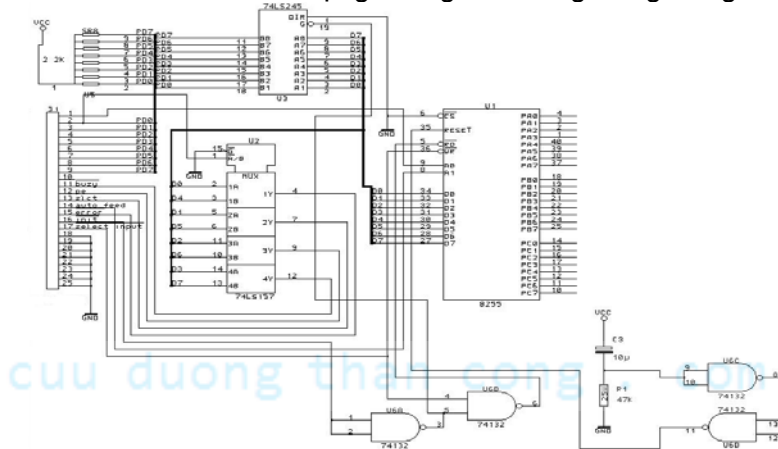
    outportb(CONTROL, inportb(CONTROL) & 0xEF); /* Cấm ngắt */
    outportb(picaddr+1, inportb(picaddr+1) | picmask); /* Che */
    setvect(intno, oldhandler); /* Hồi phục ngắt trước khi thoát */
}

```

Ta có thể mở rộng khả năng giao tiếp của cổng song song bằng cách thêm các vi mạch như 8255, 74LS374



Hình 6.7 Mở rộng số ngõ ra cổng song song

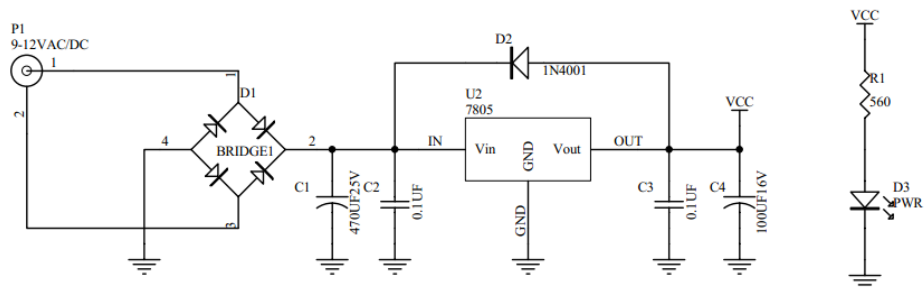
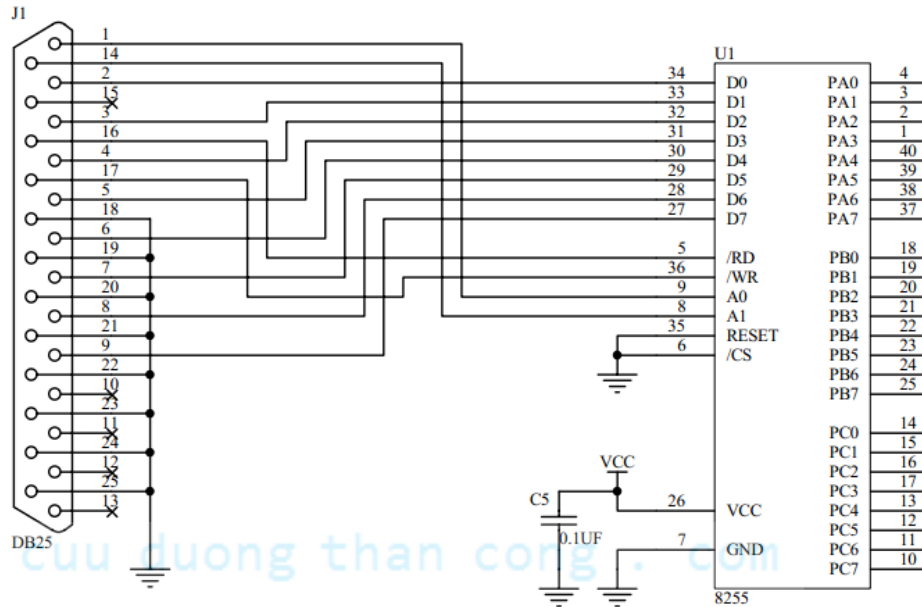


Hình 6.8 Dùng 8255 mở rộng ngõ vào ra

Chân điều khiển:

Xuất từ DR ra port của 8255: chân 14,1(/c1,/c0) chọn thanh ghi, chân 17(/c3)=0, chân 16(c2)=1

Đọc port của 8255 vào SR(/s7, s5, s4, s3): chân 14,1 chọn thanh ghi, chân 17=1, chân 16=0. Đọc nửa byte thấp: chân 2 (D0)=0  
 Đọc nửa byte cao: chân 2 (D0)=1

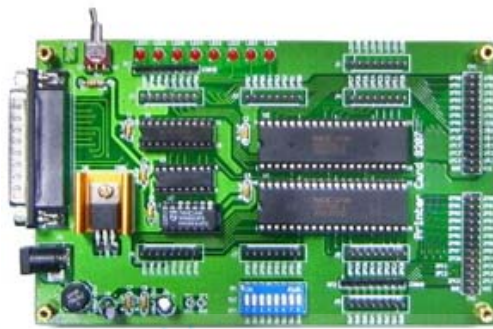


Hình 6.9 Dùng 8255 mở rộng ngõ vào ra



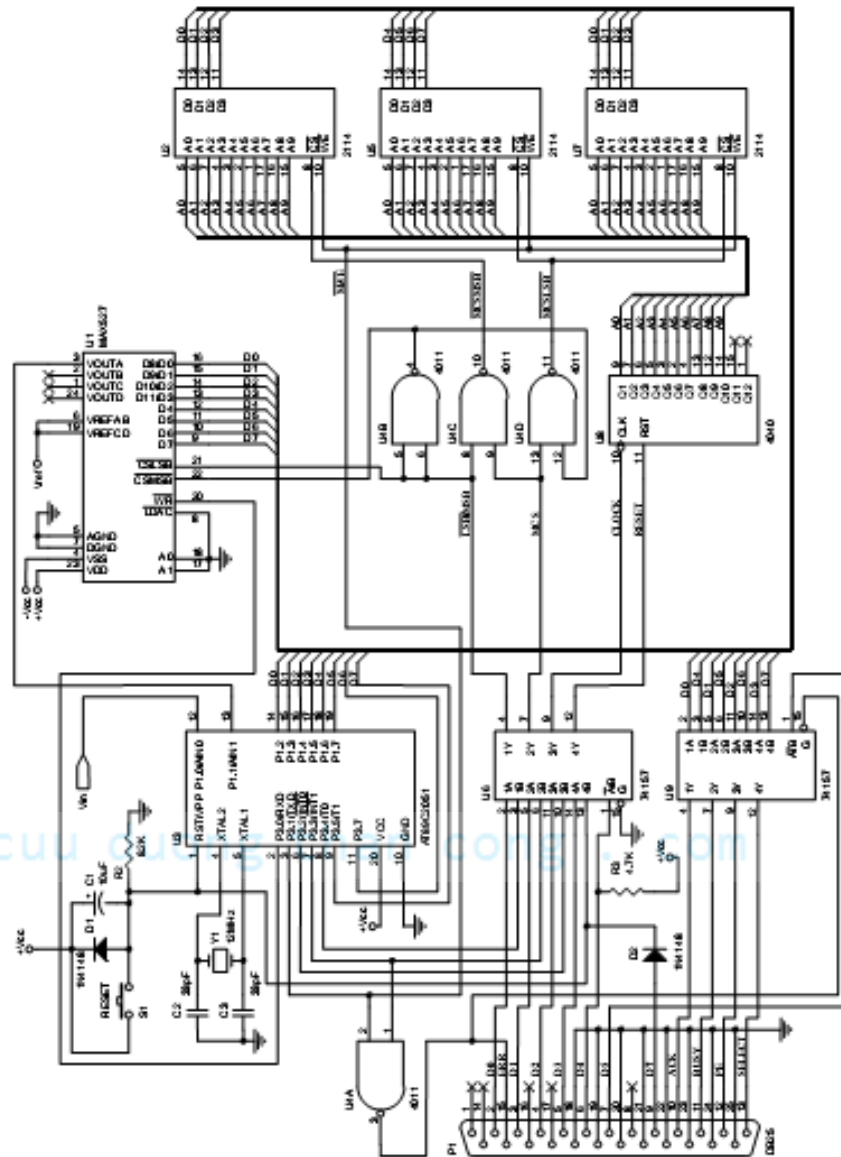


Hình 6.10 Mạch giao tiếp cổng song song với một PIA 8255



Hình 6.11 Mạch giao tiếp cổng song song với hai PIA 8255

cuu duong than cong . com



Hình 6.12

## 6.2 CỔNG EPP (ENHANCED PARALLEL PORT)

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.

**Bảng 6.2**

| Chân       | Tên             | Vào/Ra   | Chức năng                                                      |
|------------|-----------------|----------|----------------------------------------------------------------|
| 1          | Ghi /Write      | Ra       | Khi thấp xuất dữ liệu ra<br>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                                                           |

| Pin   | SPP Signal      | EPP Signal     | IN/OUT | Function                                                                           |
|-------|-----------------|----------------|--------|------------------------------------------------------------------------------------|
| 1     | Strobe          | Write          | Out    | A low on this line indicates a Write, High indicates a Read                        |
| 2-9   | Data 0-7        | Data 0-7       | In-Out | Data Bus. Bi-directional                                                           |
| 10    | Ack             | Interrupt      | In     | Interrupt Line. Interrupt occurs on Positive (Rising) Edge.                        |
| 11    | Busy            | Wait           | In     | Used for handshaking. A EPP cycle can be started when low, and finished when high. |
| 12    | Paper Out / End | Spare          | In     | Spare - Not Used in EPP Handshake                                                  |
| 13    | Select          | Spare          | In     | Spare - Not Used in EPP Handshake                                                  |
| 14    | Auto Linefeed   | Data Strobe    | Out    | When Low, indicates Data transfer                                                  |
| 15    | Error / Fault   | Spare          | In     | Spare - Not used in EPP Handshake                                                  |
| 16    | Initialize      | Reset          | Out    | Reset - Active Low                                                                 |
| 17    | Select Printer  | Address Strobe | Out    | When low, indicates Address transfer                                               |
| 18-25 | Ground          | Ground         | GND    | Ground                                                                             |

### So sánh SPP và EPP

cuu duong than cong . com

| Enhanced Parallel Port (EPP) |                | Extended Capabilities Port (ECP) |                |
|------------------------------|----------------|----------------------------------|----------------|
| Signal Name                  | DB-25 (25-pin) | Signal Name                      | DB-25 (25-pin) |
| Write                        | 1              | Host CLK                         | 1              |
| Data 0                       | 2              | Data 0                           | 2              |
| Data 1                       | 3              | Data 1                           | 3              |
| Data 2                       | 4              | Data 2                           | 4              |
| Data 3                       | 5              | Data 3                           | 5              |
| Data 4                       | 6              | Data 4                           | 6              |
| Data 5                       | 7              | Data 5                           | 7              |
| Data 6                       | 8              | Data 6                           | 8              |
| Data 7                       | 9              | Data 7                           | 9              |
| Interrupt                    | 10             | Periph CLK                       | 10             |
| Wait                         | 11             | Periph ACK                       | 11             |
| Spare                        | 12             | Ack Reverse                      | 12             |
| Spare                        | 13             | X-Flag                           | 13             |
| Data Strobe                  | 14             | Host ACK                         | 14             |
| Spare                        | 15             | Periph Request                   | 15             |
| Reset                        | 16             | Reverse Request                  | 16             |
| Address Strobe               | 17             | 1284 Active                      | 17             |
| Ground                       | 18             | Ground                           | 18             |
| Ground                       | 19             | Ground                           | 19             |
| Ground                       | 20             | Ground                           | 20             |
| Ground                       | 21             | Ground                           | 21             |
| Ground                       | 22             | Ground                           | 22             |
| Ground                       | 23             | Ground                           | 23             |
| Ground                       | 24             | Ground                           | 24             |
| Ground                       | 25             | Ground                           | 25             |

Cổng EPP có thêm một số thanh ghi như trong bảng 6.3:

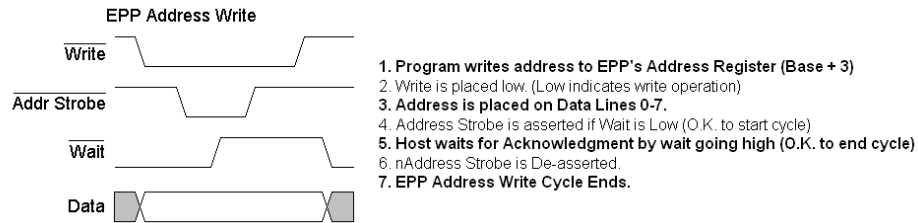
**Bảng 6.3**

| 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 | Dùng để truyền 16/32 bit   |           |
| Gốc + 6 |                            |           |
| Gốc + 7 |                            |           |

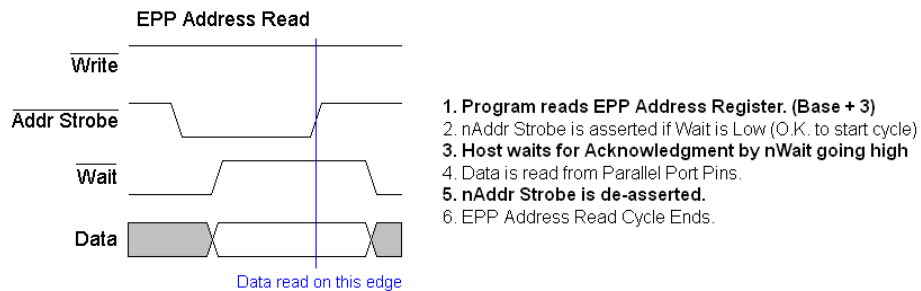
Ba thanh ghi đầu giống như trong SPP. Muốn truyền địa chỉ theo EPP ta đưa địa chỉ vào thanh ghi gốc +3 và mạch logic sẽ tạo

các tín hiệu cần thiết.

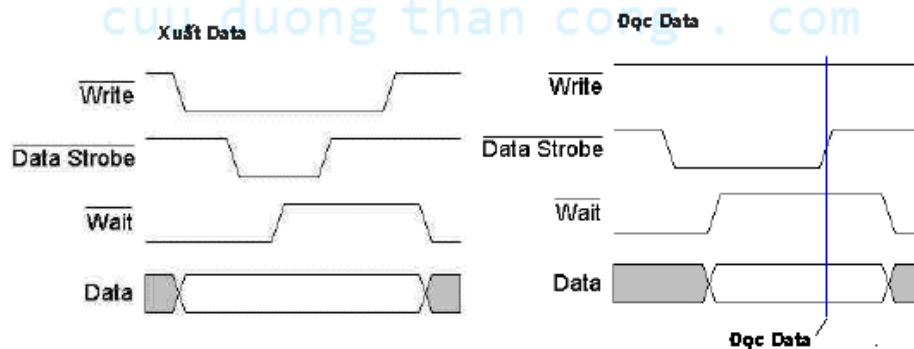
Muốn truyền dữ liệu theo EPP ta đưa dữ liệu vào thanh ghi gốc +4 và mạch logic sẽ tạo các tín hiệu cần thiết.



Hình 6.13 Xuất địa chỉ



Hình 6.14 Đọc địa chỉ



Hình 6.15 Giản đồ xuất nhập dữ liệu

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 /Wait 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

và 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 số 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  $10\mu s$  đường /Wait không tác động khi /Data Strobe hay /Address strobe tác động thì bit này sẽ được đặt lên 1.



### 6.3 CỔNG ECP

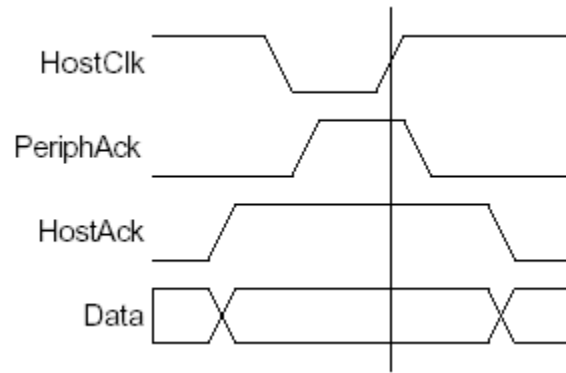
Cổng ECP được phát triển bởi Hewlett Packard và Microsoft, sử dụng phần cứng hỗ trợ cho việc truyền dữ liệu nên có vận tốc truyền nhanh, tương tự cổng EPP. Đặc điểm của cổng ECP là nén dữ liệu khi truyền, cho phép tăng tốc độ truyền dữ liệu. Cổng ECP dùng 11 thanh ghi từ gốc+0 đến gốc+7 và gốc+400H đến gốc+402H. Chân cổng ECP được quy định trong bảng 6.4.

**Bảng 6.4**

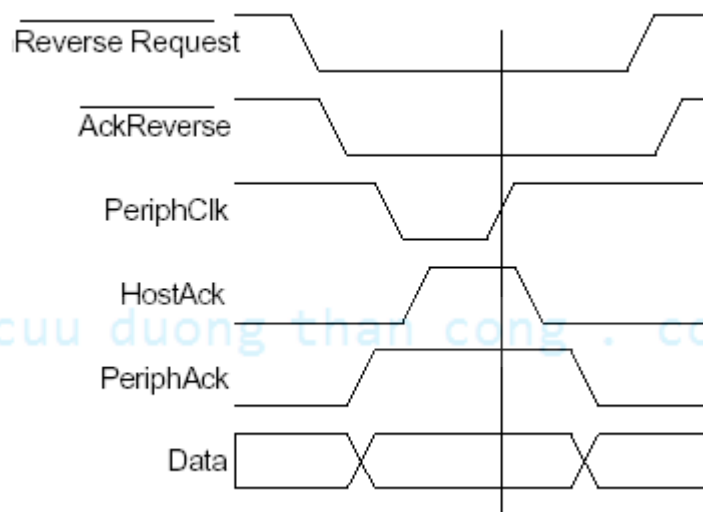
| Chân    | Tín hiệu ECP     | Hướng     | Chức năng                                                                                                                    |
|---------|------------------|-----------|------------------------------------------------------------------------------------------------------------------------------|
| 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 gửi 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 Periph Ack 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 hai 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'.

cuu duong than cong . com



**Hình 6.6:** Truyền dữ liệu từ máy tính ra ngoại vi



**Hình 6.7:** 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.

**Bảng 6.5**

| 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 |



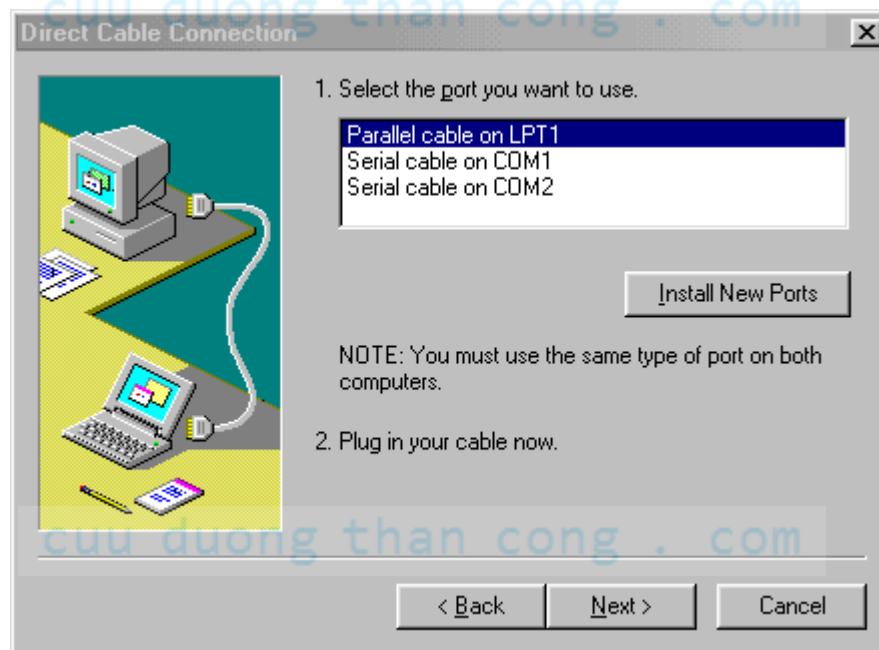
|             |                                      |           |
|-------------|--------------------------------------|-----------|
| Gốc + 2     | Thanh ghi điều khiển (mọi mode)      | Đọc / Ghi |
| Gốc + 400 H | Data FIFO (mode FIFO cổng song song) | Đọc / Ghi |
|             | Data FIFO (ECP)                      | 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.

Chi tiết về các thanh ghi có thể xem trong các tài liệu chuyên sâu hay địa chỉ [www.beyondlogic.org](http://www.beyondlogic.org), [www.lvr.com](http://www.lvr.com).

## 6.4 GHÉP NỐI HAI 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.



**Hình 6.8:** Giao diện ghép nối hai máy tính**1- Ghép nối song song (Laplink cable):** Dùng cáp nối với hai đầu đực 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   |

**2- Ghép nối nối tiếp:** Dùng cáp nối hai 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   |

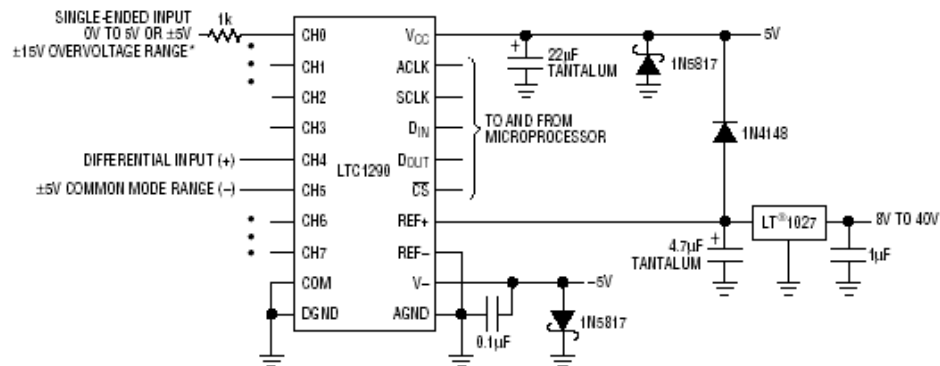
Truyền bằng đường song song nhanh gấp tám đến mười lần truyền nối tiếp.

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

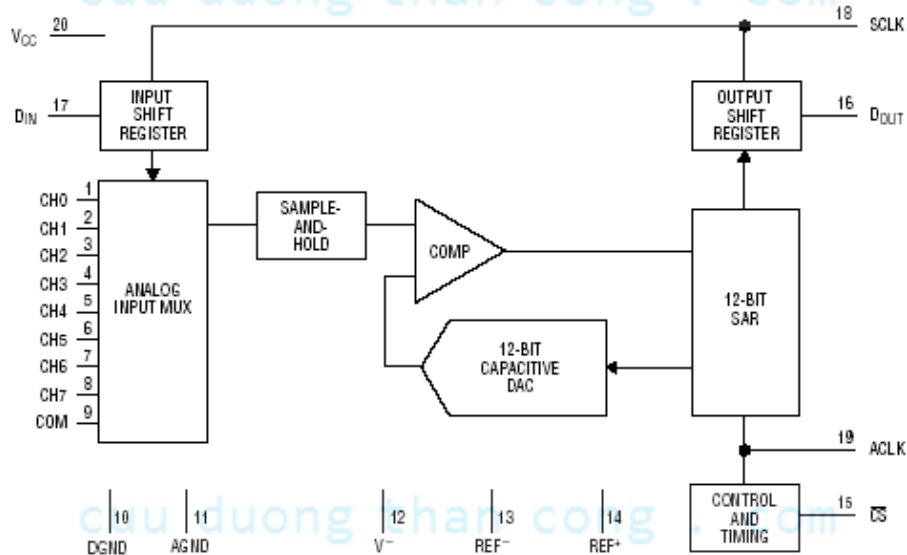
| Chân  |   | Chân  | Chân    |   | Chân    |
|-------|---|-------|---------|---|---------|
| 1     | → | 10    | 15      | ← | 17      |
| 2 ÷ 9 | ↔ | 2 ÷ 9 | 14      | → | 11      |
| 10    | ← | 1     | 16      | → | 12      |
| 11    | ← | 14    | 17      | → | 15      |
| 12    | ← | 16    | 18 ÷ 25 | ↔ | 18 ÷ 25 |
| 13    | ← | 13    |         |   |         |

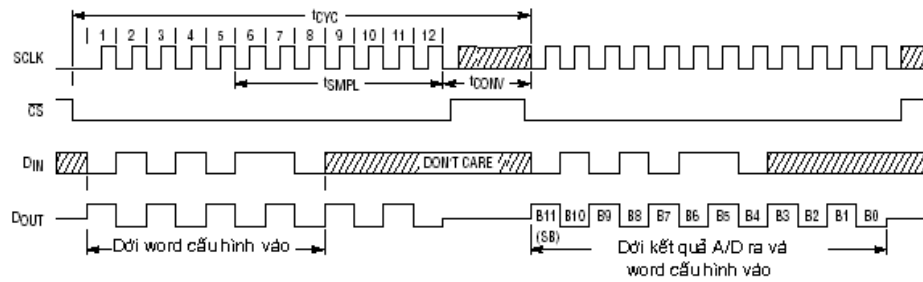
## 6.5 CARD CHUYỂN ĐỔI 8 KÊNH 12 BIT DÙNG CỔNG MÁY IN

Sử dụng vi mạch LTC 1290 8 kênh 12 bit ta có thể thực hiện mạch chuyển đổi 12 bit ghép với máy tính qua cổng máy in. Vi mạch LTC 1290 là sản phẩm của hãng Linear technology đổi A/D dùng phương pháp xấp xỉ liên tiếp 12 bit đơn cực hay 11 bit cộng dấu, thời gian đổi  $13\mu s$  max, kết quả đo đưa ra ngoài dạng nối tiếp. Hình 6.9 cho sơ đồ áp dụng.



Hình 6.9: Sơ đồ áp dụng LTC1290





**Hình 6.10** Sơ đồ khối LTC1290 và giản đồ thời gian

Vi mạch hoạt động với xung đồng bộ SCLK. Khi  $\overline{CS}$  chuyển từ 1 xuống 0, cạnh lên của SCLK đưa 8 bit từ Din vào thanh ghi dờ để định chế độ hoạt động của vi mạch, cạnh xuống của SCLK đưa kết quả chuyển đổi lần trước từ thanh ghi dờ nối tiếp ra Dout. Khi đã đủ 8 bit Din vào cấu hình chuyển đổi theo bảng sau, B7 là bit Din vào trước.

| B7            | B6  | B5        | B4        | B3                    | B2          | B1              | B0              |
|---------------|-----|-----------|-----------|-----------------------|-------------|-----------------|-----------------|
| Đơn/<br>visai | Dấu | Chọn<br>1 | Chọn<br>0 | Đơn cực/<br>Lưỡng cực | MSB/<br>LSB | Chiều dài<br>từ | Chiều dài<br>từ |

B7 = 1: 8 ngõ vào đơn so với chân COM, B6 B5 B4 chọn một trong 8 ngõ vào.

B7 = 0: 8 ngõ vào chia thành 4 cặp vi sai 01, 23, 45, 67, bit B5 B4 chọn một trong 4 cặp còn bit B6 = 1 sẽ ấn định các đầu 0, 2, 4, 6 dương so với 1, 3, 5, 7; B6 = 0 ấn định ngược lại.

B3 = 1: chọn đổi đơn cực, tín hiệu vào từ 0 đến  $V_{ref} = +5V$

B3 = 0: chọn đổi lưỡng cực, tín hiệu vào  $-V_{ref}$  (800H) đến  $V_{ref} - 1LSB$  (7FFH), bit đầu là bit dấu

B2 = 1: cho biết sẽ truyền MSB đầu tiên ra Dout

= 0: cho biết sẽ truyền LSB đầu tiên ra Dout

B1 và B0 cho biết chiều dài từ của Dout.

| B1 | B0 |           |
|----|----|-----------|
| 0  | 0  | 8 bit     |
| 0  | 1  | cắt nguồn |
| 1  | 0  | 12 bit    |
| 1  | 1  | 16 bit    |

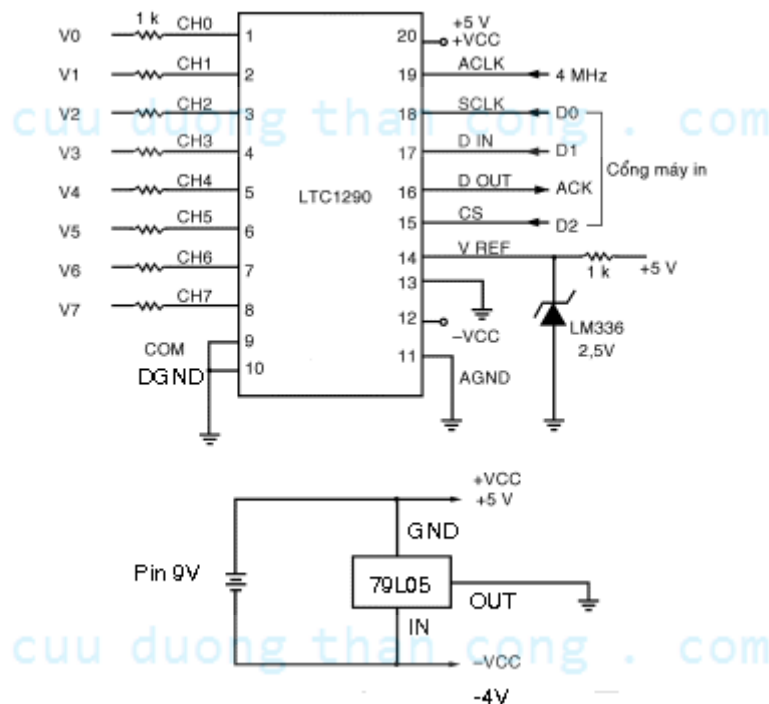
B1= 0, B0= 1: vi mạch ở chế độ cắt nguồn lúc đó  $\overline{CS} = 1$  dòng tiêu thụ khoảng  $10\mu A$ .

Sau khi đã đủ số xung nhịp SCLK được ấn định bởi chiều dài

từ  $\overline{CS}$  được đưa lên mức 1, bắt đầu chuyển đổi theo xung nhịp ACLK, tần số tối đa 4MHz, sau đó từ 48 đến 52 xung ACLK, nếu  $\overline{CS}$  xuống mức 0 thì kết quả đổi sẽ đưa ra Dout theo cấu hình đã đặt trước còn Din sẽ đưa vào và đặt cấu hình mới. Các thông tin chi tiết hơn về vi mạch có thể tìm trong trang web: [www.linear-tech.com](http://www.linear-tech.com).

Mạch chuyển đổi H6.10 cho phép đổi tín hiệu đơn  $0V \div 2,5V$  hay tín hiệu lưỡng cực  $(-2,5V) \div (+2,5V)$ . Xung nhịp SCLK và dữ liệu Din xuất ra từ cổng máy in còn kết quả đổi Dout đưa vào ACK của cổng.

```
{===== CHƯƠNG TRÌNH ĐIỀU KHIỂN CHO CARD AD 12 BIT =====}
unit ltc1290;
interface
const
    vref = 2.48;
function adc_read (ch: integer): real;
implementation
```



Hình 6.10

```

const
    sclk = 1;
    din = 2;
    dout = 64;
    cs = 4;
function adc_read;
const
    wconst = 1 + 32 + 64; {vào đơn, lưỡng cực, MSB đầu}
    chconst: array [0..7] of byte = (0, 2, 8, 10, 4, 6, 12, 14);
    delay = 10; {có thể thay đổi tùy máy tính}
    aout = $378;
    ain = $379;
var
    i: integer;
    win: word;
    result: word;
begin
    for i:= 1 to delay do begin end;
    win := wconst + chconst [ch];
    port [aout] := 0; {drop CS} {dời word cấu hình vào}
    for i := 1 to 12 do begin
        port [aout] := (win and 1)* din;
        port [aout] := (win and 1)* din + sclk;
        win := win div 2;
        port [aout] := 0;
    end;
    port [aout] := cs;
    for i := 1 to delay do begin end;
    port [aout] := 0; {nhận kết quả chuyển đổi}
    result := 0;
    win := wconst + chconst [ch];
    for i := 1 to 12 do begin
        port [aout] := (win and 1)* din;
        port [aout] := (win and 1)* din + sclk;
        result := result*2;
        if port [ain] and dout <> 0 then result := result + 1;
        port [aout] := 0;
    end;
    port [aout] := cs;
    adc_read := vrel*((result + 2048) and 4095) / 2048 - vref;
end;
{===== CHƯƠNG TRÌNH DEMO =====}
program testltc;
uses crt, ltc1290
var i: integer;
begin
    writeln ('Nhấn phím để thoát');
    while not keypressed do begin
        for i := 0 to 7 do write (adc_read(i) :4:4, '');
        writeln ('V');
    end;
end;

```

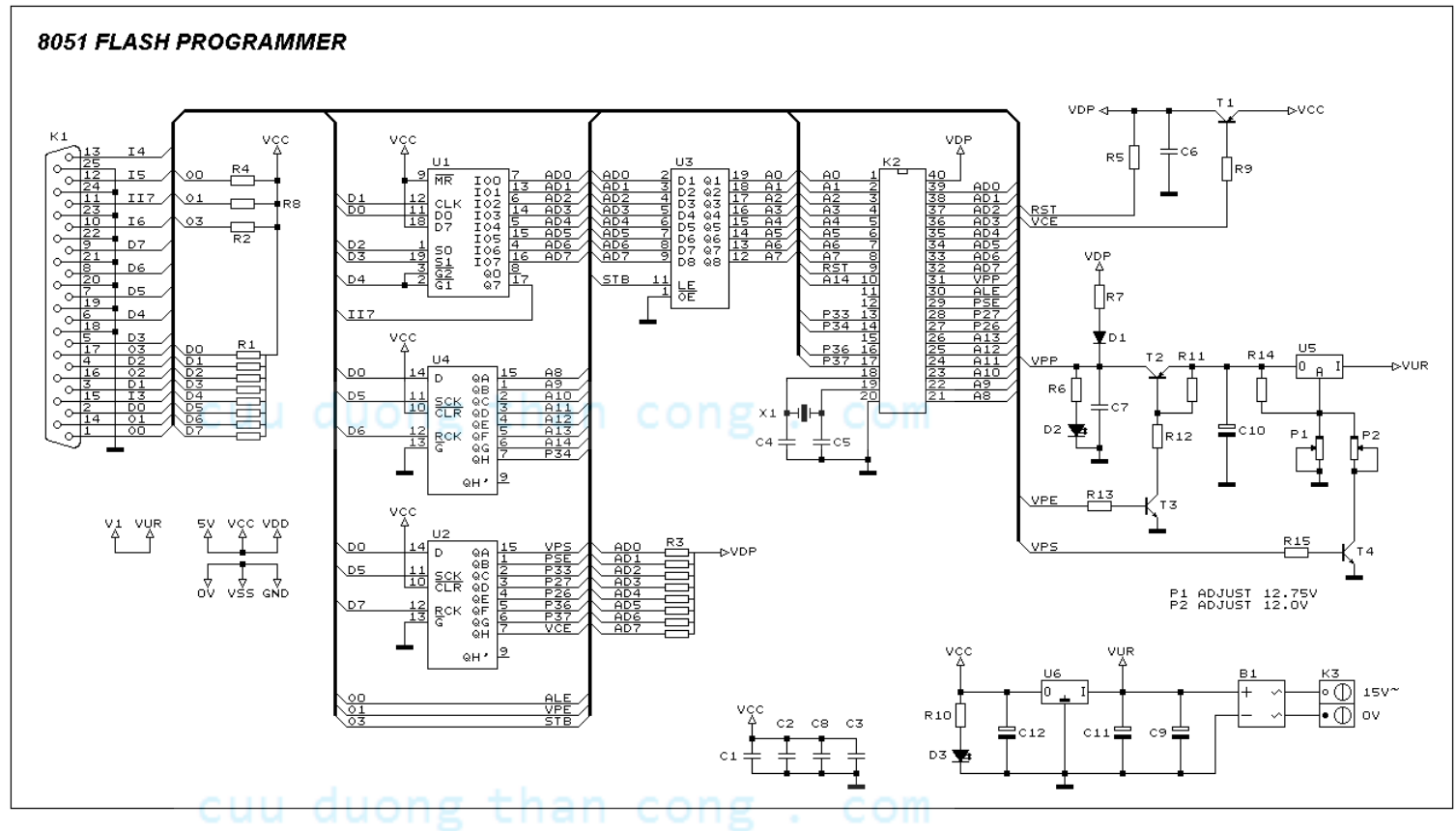
```
        delay (1000);  
    end;  
end.
```

## 6.6 MẠCH LẬP TRÌNH VI ĐIỀU KHIỂN ATMEL 89C

Máy tính thường dùng để lập trình cho vi điều khiển hay EPROM qua cổng song song hay nối tiếp. Mục này trình bày bộ lập trình cho vi mạch họ Atmel 89C51/52/55/1051/2051/4051

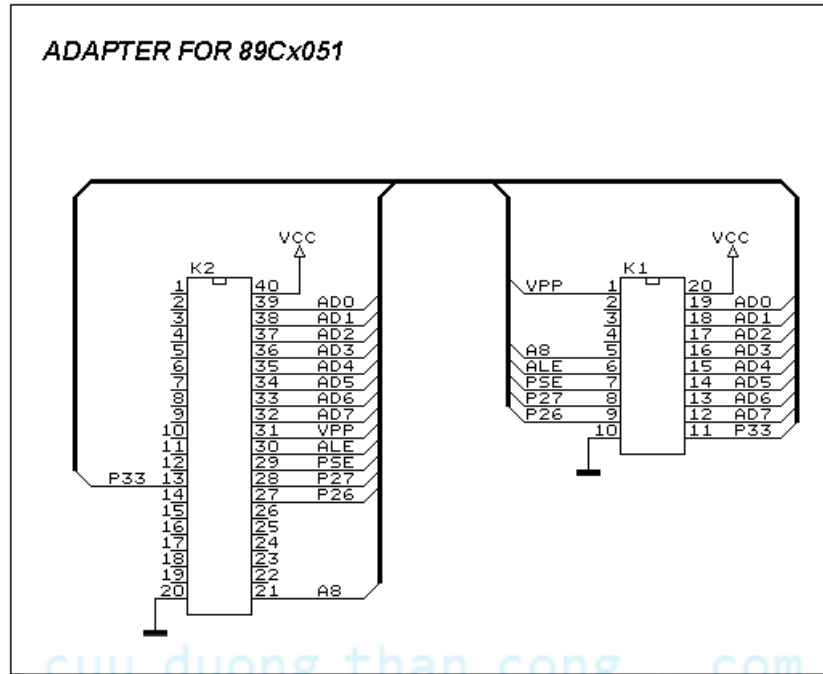
cuu duong than cong . com

cuu duong than cong . com





U1 điều khiển dữ liệu giữa máy tính và bộ điều khiển, U3 cài byte địa chỉ thấp U4 cài byte địa chỉ cao, U2 tạo các tín hiệu điều khiển, U6 cấp nguồn 5V, U5 tạo các điện áp lập trình. chi tiết ở địa chỉ <http://chaokhun.kmitl.ac.th>.



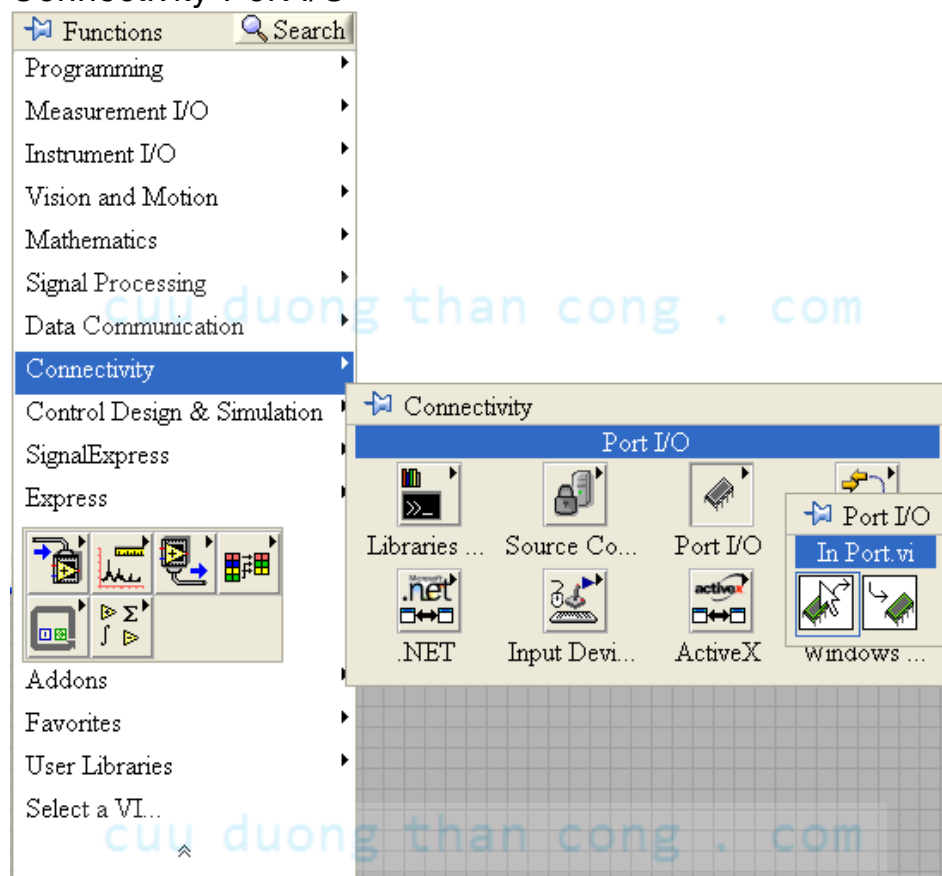
Bảng linh kiện:

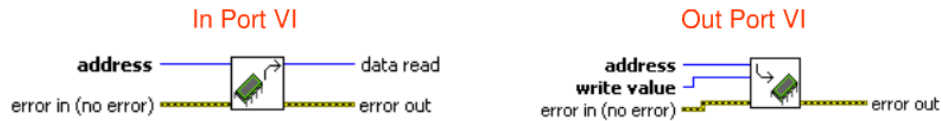
|                          |                        |
|--------------------------|------------------------|
| C1,C2,C3,C6,C8           | 100nF                  |
| C4,C5                    | 33pF                   |
| C7                       | 2.2nF                  |
| C9                       | 1000uF 25V             |
| C10,C11,C12              | 10uF 25V               |
| D1                       | 1N4148                 |
| D2                       | LED RED 3mm            |
| D3                       | LED GREEN 3mm          |
| K2                       | ZIF SOCKET 40 WAY      |
| P1                       | 5K MULTITURN           |
| P2                       | 50K MULTITURN          |
| R1,R3                    | RESISTOR ARRAY 4K7 x 8 |
| R2,R4,R8,R11,R12,R13,R15 | 4K7                    |
| R7,R5                    | 1K                     |
| R6                       | 3K3                    |
| R9                       | 2K7                    |
| R10                      | 680R                   |

|       |                 |
|-------|-----------------|
| R14   | 220R            |
| T2,T1 | 2N3906          |
| T3,T4 | 2N3904          |
| U1    | 74HC299         |
| U4,U2 | 74HC595         |
| U3    | 74HC573         |
| U5    | LM317LZ         |
| U6    | LM7805          |
| X1    | CRYSTAL 4.0 MHz |

## 6.7 LẬP TRÌNH CỔNG SONG SONG DÙNG LABVIEW

LabView giao tiếp cổng song song qua hai công cụ In port.vi và Out port.vi trong thư mục Functions-Connectivity-Port I/O





Address: địa chỉ 16 bit

Data read: 1 byte, 2 byte (hai địa chỉ kế nhau kể từ address), 4 byte (bốn địa chỉ kế tiếp kể từ address)

Write value có thể là 1, 2, 4 byte

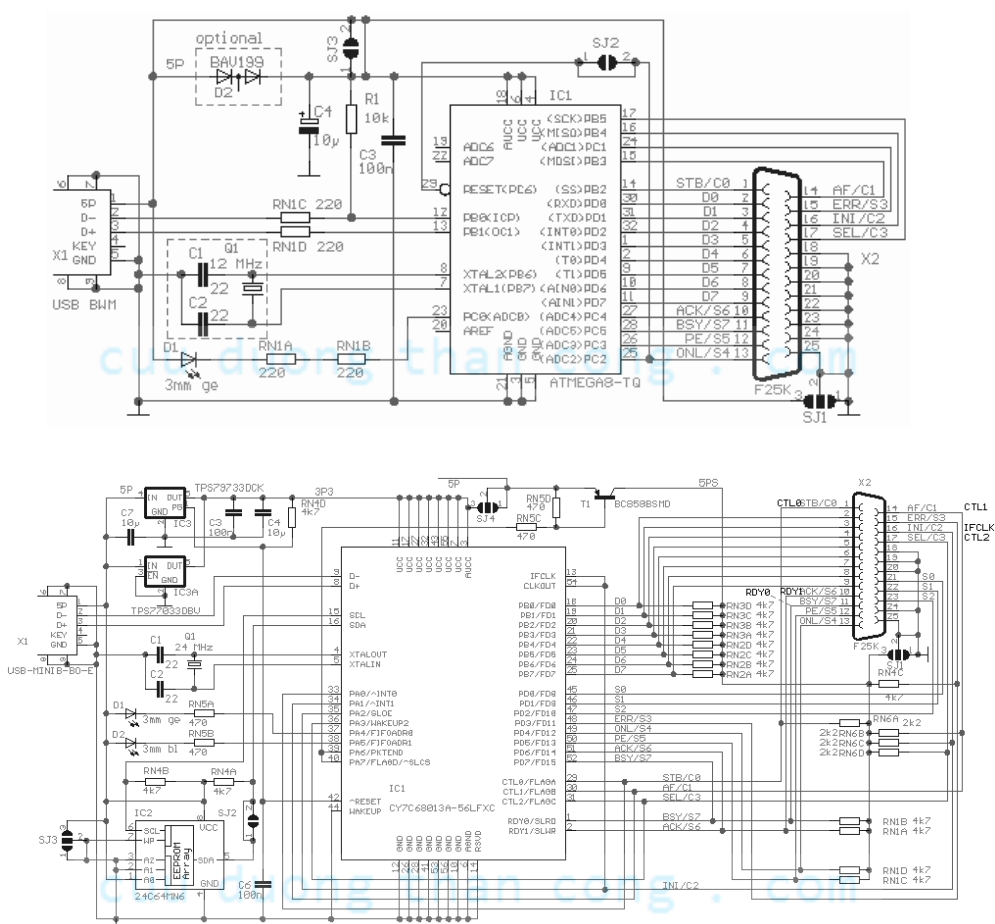
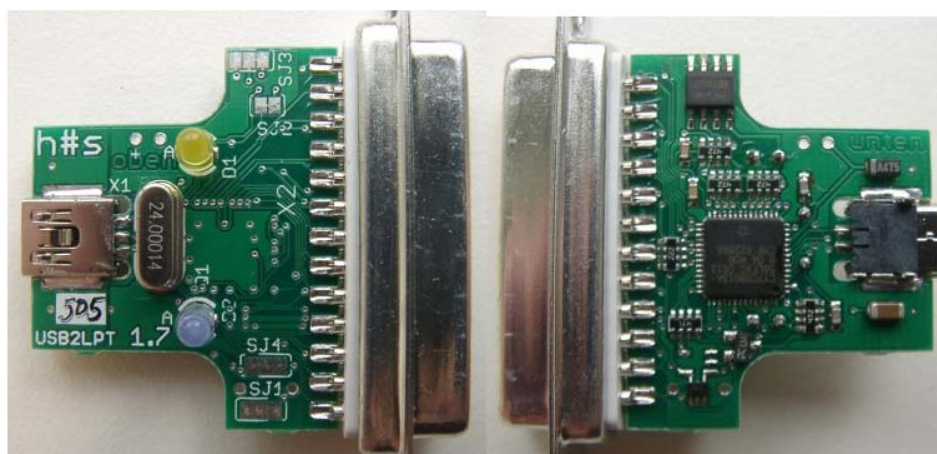
## 6.8 CHUYỂN ĐỔI USB SANG CỔNG LPT

Nhiều máy tính không có cổng song song có thể dùng vi mạch chuyển đổi từ USB sang cổng song song. Vi mạch thường dùng là vi điều khiển giao tiếp USB với máy tính (ATMEGA, CYPRESS, PIC18), phần mềm driver trên máy tính sẽ tạo cổng song song ảo, truyền dữ liệu qua USB, phần mềm firmware trên vi điều khiển tạo cổng song song dựa trên các port của vi điều khiển (<http://www-user.tu-chemnitz.de/>)

cuu duong than cong . com



cuu duong than cong . com



-----O-----

**Bài tập gợi ý**

- 1/ Nghiên cứu ghép 8255 với cổng máy in để tăng số ngõ logic.
- 2/ Thiết kế và lập trình mạch mở rộng cổng máy in 56 bit ra và 8 bit vào.
- 3/ Thiết kế mạch điều khiển động cơ bước 4 pha dùng cổng song song và viết chương trình quay  $n$  xung với tần số  $f$  Hz.
- 4/ Nghiên cứu hoạt động và viết chương trình cho mạch ghi AT89C51.

cuu duong than cong . com

cuu duong than cong . com