

**GIÁO TRÌNH
ĐO LƯỜNG VÀ
ĐIỀU KHIỂN MÁY TÍNH**

Chương 1

GIAO TIẾP QUA RÃNH CẮM MÁY TÍNH

Máy tính tương thích IBM là loại máy tính phổ biến trên thế giới, tùy theo ứng dụng có thể phân thành:

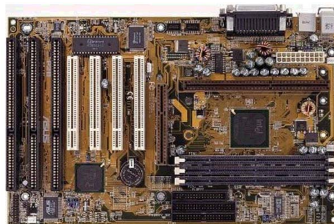
- Máy tính để bàn (*Desktop Personal Computer - Desktop PC*) sử dụng trong văn phòng, công sở cho cá nhân hay máy trạm trong mạng.
- Máy tính chủ (*Server PC*) dùng làm máy chủ trong mạng.
- Máy tính công nghiệp (*Industrial PC*) dùng trong môi trường công nghiệp, chịu điều kiện khắc nghiệt về môi trường như nhiệt độ, độ ẩm, chấn động, va chạm và yêu cầu cao về độ cứng, rắn chắc, chống va đập và xác suất hư hỏng thấp.
- Máy tính panel (*Panel PC*) có màn hình và bàn phím kết hợp và gắn trên bảng điều khiển, thông thường màn hình loại tinh thể lỏng TFT LCD, một số màn hình touchscreen với phím bấm lập trình trực tiếp trên màn hình.
- Máy tính kiểu nhúng (*Embedded PC*) có máy tính (vi xử lý) được kết hợp vào một thiết bị khác (nhúng).

Cấu tạo máy tính thông thường có nguồn cấp, mainboard gồm CPU, bộ nhớ, các chip điều khiển phụ trợ chipset, đĩa cứng, đĩa mềm, đĩa CD và các rãnh cắm dùng cho các card mở rộng ISA, EISA, VESA, PCI,... và các đầu nối cho máy in, màn hình, bàn phím, chuột, modem, USB (*Universal Serial Bus*), Fire Wire, hồng ngoại (IrDA), mạng.

Máy tính công nghiệp chế tạo dưới dạng các board cắm trên một đế, mainboard chứa CPU cũng chế tạo dưới dạng board cắm, đĩa cứng đôi khi được thay bằng đĩa thể rắn (*SSD - Solid State Disk*) thực chất là bộ nhớ không bốc hơi lập trình được, dung lượng có thể lên đến 1 Gbyte, có vận tốc nhanh hơn và bền hơn đĩa cứng.

Máy tính sử dụng trong hệ thống đo lường điều khiển phải giao tiếp với ngoại vi, có nhiều cách giao tiếp như:

- Qua các card đo lường, điều khiển gắn vào rãnh cắm trên mainboard máy tính.
- Giao tiếp qua cổng máy in song song
- Giao tiếp qua cổng nối tiếp RS-232
- Giao tiếp qua cổng nối tiếp USB, Fire Wire
- Giao tiếp dùng cổng hồng ngoại



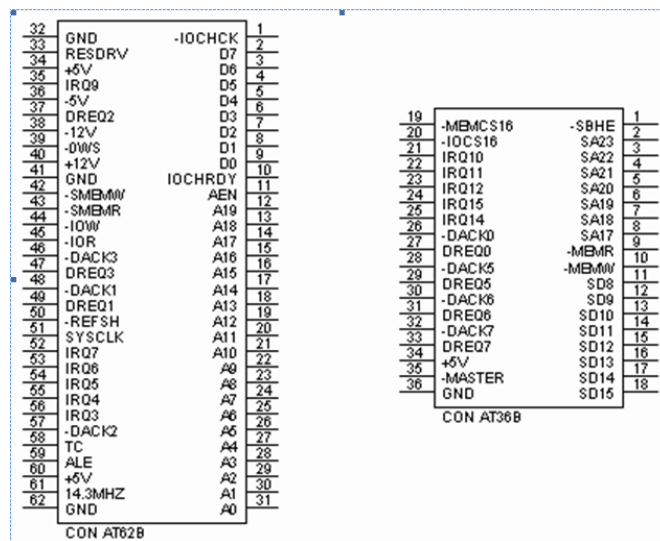
Hình 1.1 Mainboard Pentium II và Pentium IV



I. GIAO TIẾP QUA RÃNH CẮM ISA.

1. Giới thiệu chung.

Rãnh cắm thông dụng nhất là rãnh ISA (*Industry Standard Architecture*) do IBM đưa ra năm 1980 cho máy 8086 XT (*Extended Technology*), sau đó là ISA 16 bit cho máy AT (*Advanced Technology*) và trở thành chuẩn AT Bus. Hiện nay các mainboard P4 không còn rãnh cắm này tuy nhiên việc nghiên cứu rãnh cắm ISA vẫn là cần thiết. Rãnh cắm ISA có màu đen trên mainboard gồm hai phần, phần đầu 62 chân, mỗi hàng 31 chân dùng cho trao đổi dữ liệu 8 bit, phần thứ hai 36 chân, mỗi hàng 18 chân dùng hỗ trợ thêm khi cần dữ liệu 16 bit. Sơ đồ chân rãnh cắm được cho trong hình 1.2.



Hình 1.2: Sơ đồ chân rãnh cắm ISA

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

| | |
|---|--|
| SA19 ÷ SA0 (System Address bus 19 ÷ 0) (I/O) | Tuyến đại chỉ 20 bit dùng truy cập bộ nhớ 1MB và ngoại vi. Có thể dùng với LA23 và 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 bởi vi xử lý hay bộ 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 ÷ 17 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 vụ cho DMA (Direct Access Memory) để trao đổi thông tin trực tiếp với bộ nhớ. DRQ sẽ ở mức cao 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 |
| - IOHRDY (I/O Channel Ready) (I) | Cho phép các board chậm bắt vi xử lý chờ 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 đến 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 báo 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 đọc 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 (Terminal 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 |

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 1.2:



Bảng 1. 2: Các địa chỉ ngoại vi đã sử dụng của một máy Pentium 2

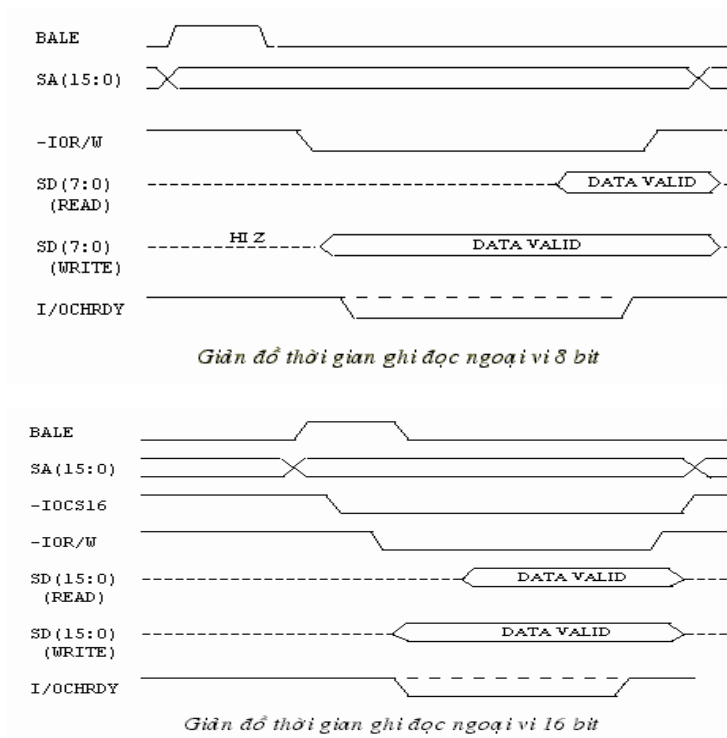
| | |
|-----------|--|
| 000 – 00F | Truy cập bộ nhớ trực tiếp (DMA Direct memory access controller) |
| 020 – 021 | Điều khiển ngắt (PIC Programmable interrupt controller) |
| 040 – 043 | Timer hệ thống (System timer) |
| 060 – 060 | Bàn phím (Keyboard) |
| 061 – 061 | Loa trong (System speaker) |
| 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 – 0DF | DMA |
| 0F0 – 0FF | Đồng xử lý số học (Numeric data processor) |
| 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 Joystick) |
| 208 – 20F | Dành cho mainboard (Motherboard 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 Controller) |
| 376 – 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 (Standard Floppy Disk Controller) |
| 3F6 – 3F6 | Điều khiển IDE (Primary IDE controller) |
| 3F8 – 3FF | Cổng truyền thông 1 (COM1) |

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.

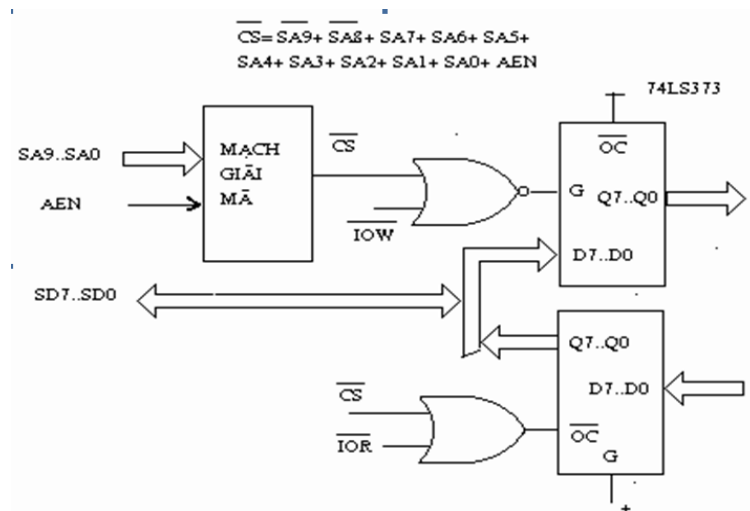
Bảng 1. 3: Các ngắt của một máy Pentium 2

| | |
|----|--|
| 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 PCI Speakerphone Modem) |
| 9 | IRQ cho PCI |
| 10 | Điều khiển ổ đĩa cứng |
| 11 | Điều khiển USB (Inter 8237-1 AB/EB PCI to USB Universal Host Controller) |
| 11 | IRQ cho PCI |
| 12 | Chuột PS/2 |
| 13 | Đồng xử lý số học |
| 14 | Điều khiển IDE thứ nhất |
| 15 | Điều khiển IDE thứ hai |

Tuyến ISA có nhiều chân nhưng thực tế sử dụng ta chỉ cần các đường địa chỉ SA0..SA9, dữ liệu SD0..SD15, điều khiển -IOR, -IOW, AEN, -IOCS16, -SBHE, ngõ vào ngắt IRQ, xung nhịp CLK và nguồn. Các ngoại vi được gọi là port, thường là 8 bit và việc xuất nhập thực hiện theo 8 bit, nếu xuất nhập 16 bit thì dùng hai port địa chỉ kế nhau.



Việc thiết kế card ISA xuất nhập 8 bit tương đối đơn giản, bao gồm mạch giải mã địa chỉ, mạch chốt dữ liệu ra và mạch đệm dữ liệu vào. Ví dụ sử dụng địa chỉ 300 ta dùng mạch logic tạo tín hiệu /CS, mạch chốt và đệm dùng vi mạch 74LS373

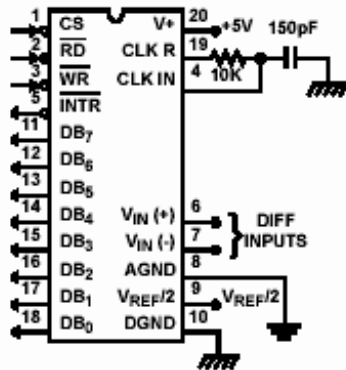


Hình 1.3: Sơ đồ nguyên lý xuất nhập

2. GIỚI THIỆU MỘT SỐ IC THÔNG DỤNG.

2.1 Vi mạch ADC và DAC.

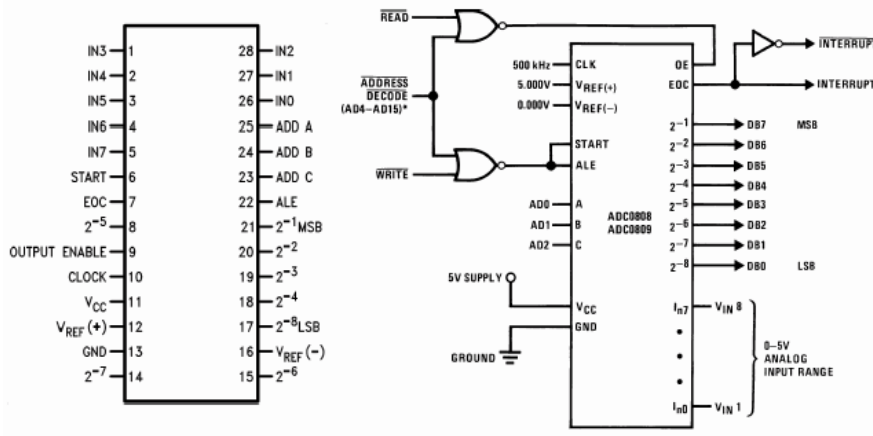
ADC0804: ADC 8 bit xấp xỉ liên tiếp (National Semiconductor)



Bắt đầu đổi: $\overline{CS}=0$, $\overline{RD}=0$, \overline{WR} : xung âm, sau thời gian chuyển đổi $< 0.1ms$, \overline{INTR} xuống 0 báo đổi xong.
 Đọc dữ liệu: $\overline{CS}=0$, $\overline{WR}=1$, \overline{RD} có xung âm, reset \overline{INTR} lên mức 1, dữ liệu N bit xuất ra khi $\overline{RD}=0$, $\overline{RD}=1$ tuyến dữ liệu ở trạng thái tổng trở cao.
 $0 \leq V_{IN(+)} - V_{IN(-)} \leq 5V$
 $1LSB = V_{REF}/256$
 V_{REF} là 2 lần điện áp ở chân 9
 $N = 256 * (V_{IN(+)} - V_{IN(-)}) / V_{REF}$, $0 \leq N \leq 255$
 Chân 9 có thể để hở, lúc đó $V_{REF} = V+$, hoặc nối vào đó một nguồn $Max(V_{IN(+)} - V_{IN(-)})/2$

Hình 2.1. Sơ đồ chân IC ADC0804

ADC0808/0809: ADC 8 kênh 8 bit.



Hình 2.2. Sơ đồ chân IC ADC0808/0809

Chọn kênh: Các chân ABC dùng để chọn 1 trong 8 chân đầu vào analog, việc chọn thực hiện ở cạnh lên của chân ALE.

Chuyển đổi: START có một xung dương, làm EOC xuống 0, sau thời gian chuyển đổi EOC lên mức 1.

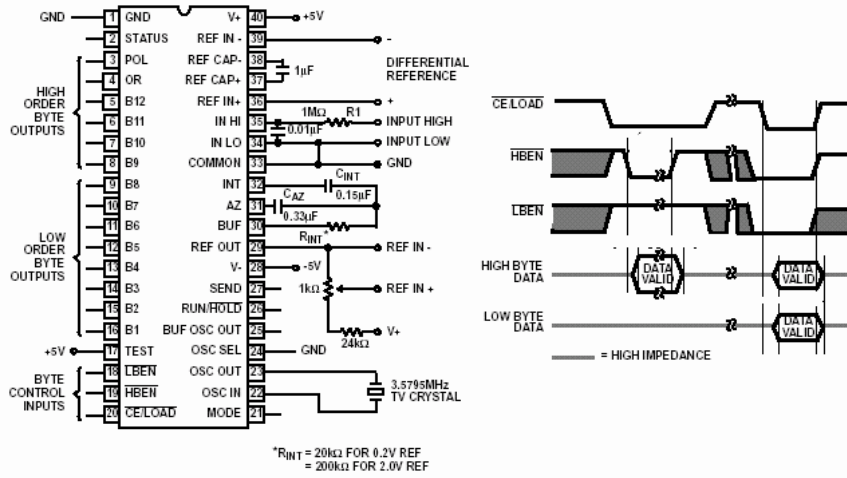
Đọc dữ liệu: $OE = 1$, $N = 256 * (V_{IN} - V_{REF(-)}) / (V_{REF(+)} - V_{REF(-)})$.

ICL7109: ADC tích phân 12 bit nhị phân (Harris Semiconductor).

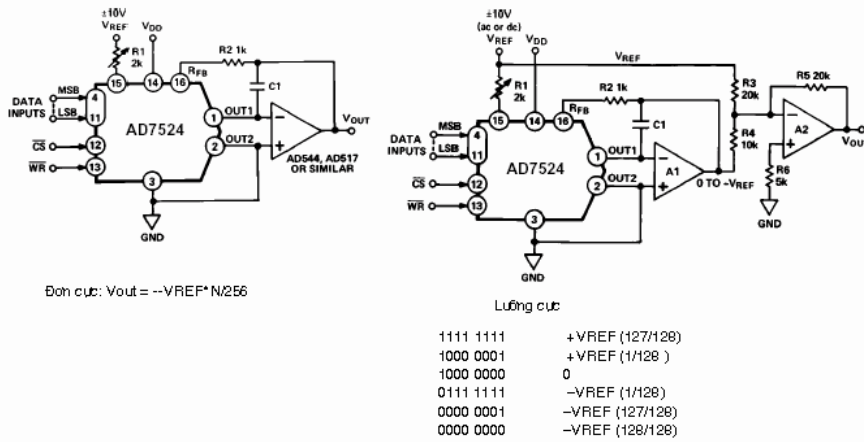
Kết quả chuyển đổi điện áp vì sai giữa hai chân INHI và INLO được xuất ra 12 bit $B_{12} \dots B_1$ theo công thức $N = 2048 * V_{INA} / V_{REF}$, cực tính điện áp chỉ bởi POL (logic 1: dương).

Nếu quá tải OR on. Các chân này ở trạng thái tổng trở cao. Khi chuyển đổi xong chân STATUS logic 0. Điện áp giữa $V+$ và REF OUT là 2.8V ổn định. ICL7109 có nhiều cách để đọc kết quả. Nếu chân MODE để hở là chế độ DIRECT, $\overline{CELOAD} = 0$, $\overline{HBEN} = 0$ thì các chân dữ liệu tích cực.

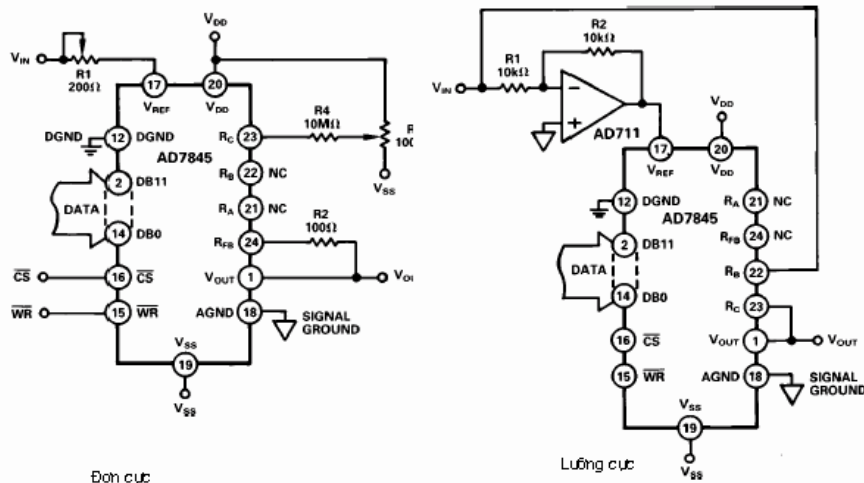
Hình 2.3. Sơ đồ chân IC L7109



Hình 2.4: Sơ đồ chân IC DAC 8 BIT



Hình 2.5: Sơ đồ chân IC DAC 12 BIT



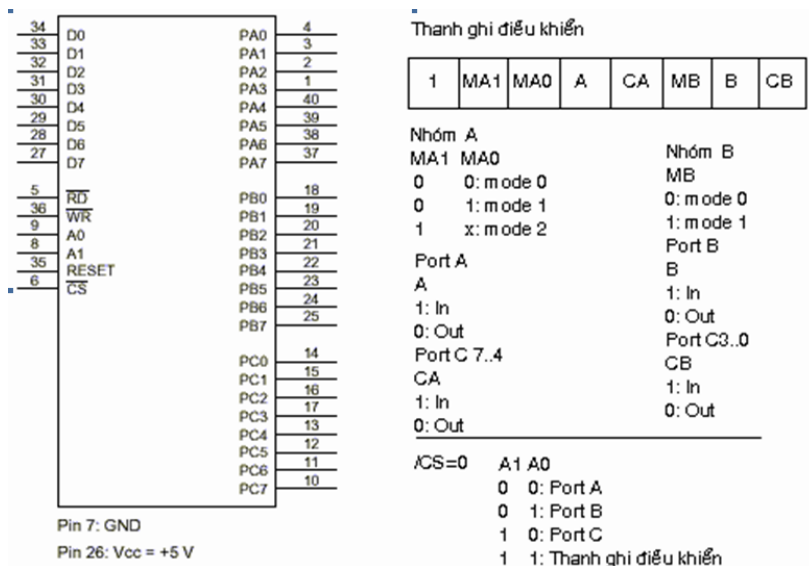
2.2 Vi mạch giao tiếp số.

Vi mạch 8255: Xuất nhập song song 24 bit gồm port A, port B và port C, thanh ghi điều khiển, còn 3 mode hoạt động:

Mode 0: Vào ra trực tiếp, các port xuất nhập độc lập.

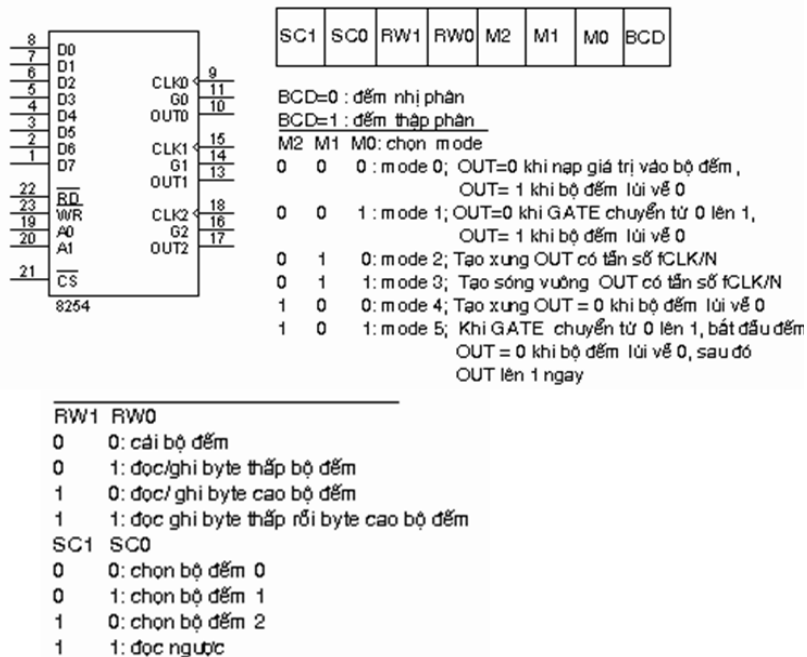
Mode 1: Vào ra bắt tay, Port A và B xuất hay nhập với tín hiệu điều khiển từ port C.

Mode 2: Vào ra 2 hướng cho Port A với tín hiệu điều khiển từ PC cao.



Hình 2.6: Sơ đồ chân và cấu trúc vi mạch 8255

Vi mạch 8254: Đếm/dịnh thì, gồm 3 bộ đếm lùi nhị phân 16 bit và thanh ghi điều khiển. Mỗi bộ đếm có 6 mode, xung nhịp vào CLK, tín hiệu ra OUT và tín hiệu điều khiển GATE.



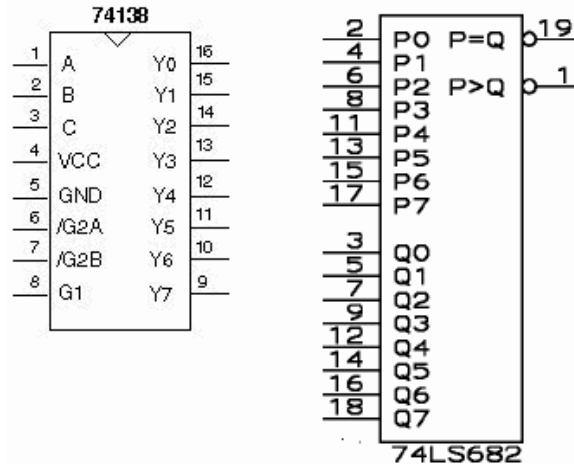
Hình 2.7: Sơ đồ chân và cấu trúc vi mạch 8254

2.3. Vi mạch giải mã.

74LS138: Giải mã 3 ra 8.

74LS139: Hai bộ giải mã 2 ra 4.

74LS682: So sánh nhị phân 8 bit.



Hình 2.8: Sơ đồ chân các vi mạch giải mã 74LS138 và 74LS682

2.4. Một số Card ISA.

Sau đây trình bày card peripheral adapter 16 bit và hai card thu thập số liệu gắn vào rãnh ISA (Hình 2.9, 2.10 và 2.11).

Card peripheral adapter sử dụng hai vi mạch 74245 đệm tuyến hai chiều cho số liệu, hai vi mạch 74244 đệm tuyến địa chỉ và điều khiển. Vi mạch 74138 giải mã địa chỉ 300 đến 3FF, khi truy cập vùng địa chỉ máy, chân Y₄ của U₅ (I/O decode) xuống thấp đưa vào U₈ (74LS244) cho phép xuất ra đường MEMW và I/O decode 8 bit. Cầu nối 8/16 bit cho phép chọn chế độ 8 bit hay 16 bit. Khi chọn chế độ 8 bit thì cầu nối hở, 8 bit cao được truyền khi đường địa chỉ A₀ ở mức cao (địa chỉ lẻ) và 8 bit thấp được truyền khi đường địa chỉ A₀ ở mức thấp (địa chỉ chẵn).

Mạch chuyển đổi AD và DA 8 bit trình bày ở hình 2.11 và 2.12. Hình 2.12 là mạch chuyển đổi AD dùng IC 7109.

Vi mạch ADC ICL 7109 chuyển đổi tín hiệu analog ra dạng số nhị phân 12 bit và ghép nối với máy tính qua vi mạch giao tiếp song song 8255 dùng 2 port A và B, port C điều khiển các chân LBEN, HBEN, R/H.

Vi mạch ICL 7109 hoạt động ở mode direct, khi R/H = 1 chuyển đổi liên tục, khi R/H = 0 sẽ ngừng chuyển đổi.

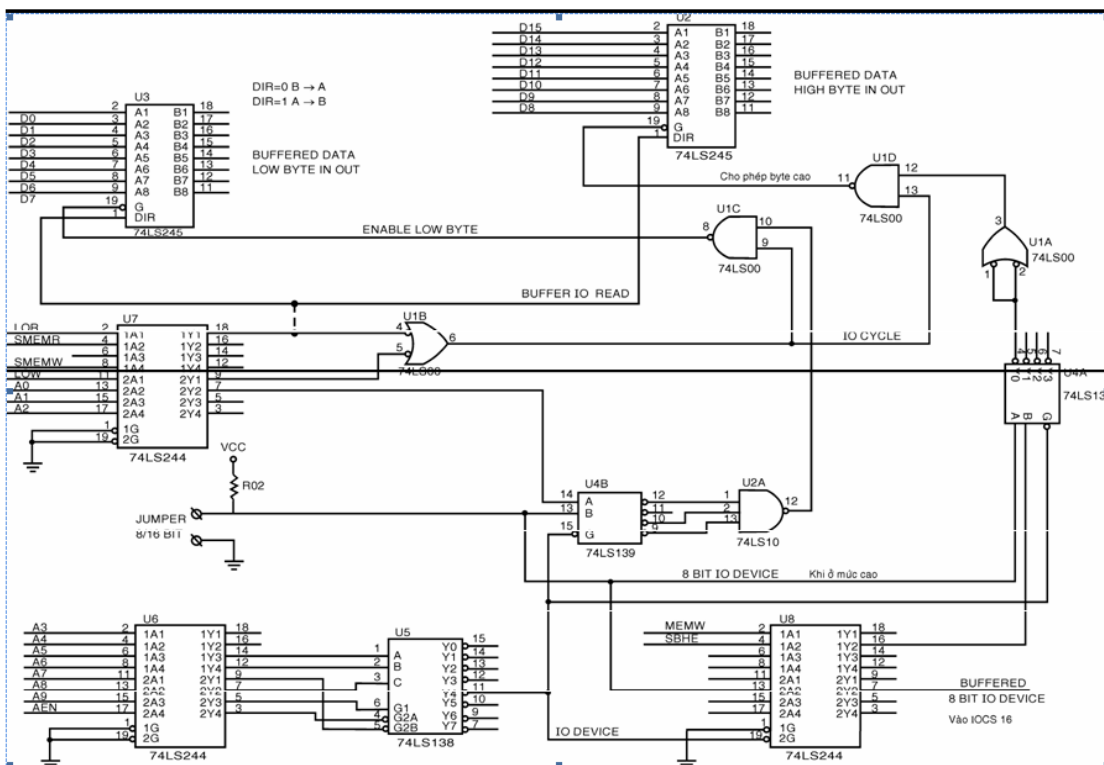
Khi /LBEN ở mức 0 thì xuất byte thấp ra port A, khi /HBEN ở mức 0 thì xuất byte cao gồm 4 bit dữ liệu cao, PDL (cực tính), OR (quá tải) và Status (trạng thái).

Quá trình đọc kết quả như sau:

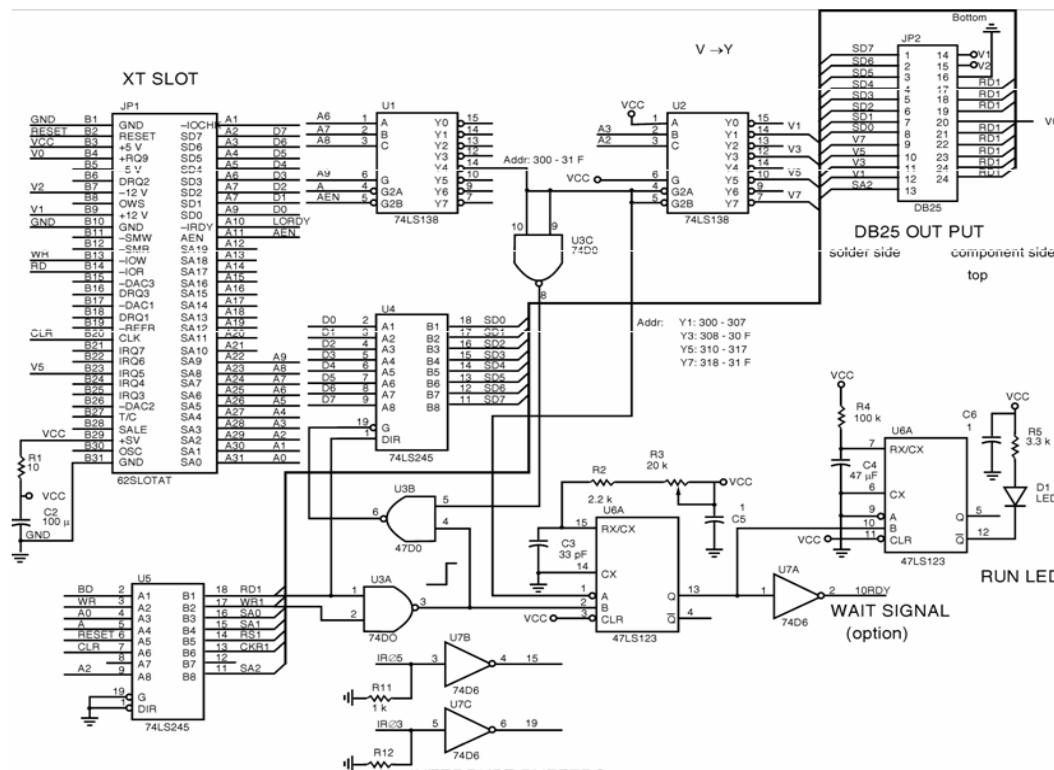
Cho R/H lên mức cao để bắt đầu chuyển đổi, sau đó chờ Status xuống mức) đổi xong, rồi cho LBEN và HBEN mức 0 để đọc dữ liệu vào.

Hình 2.13 là bộ đếm dùng IC 8254.

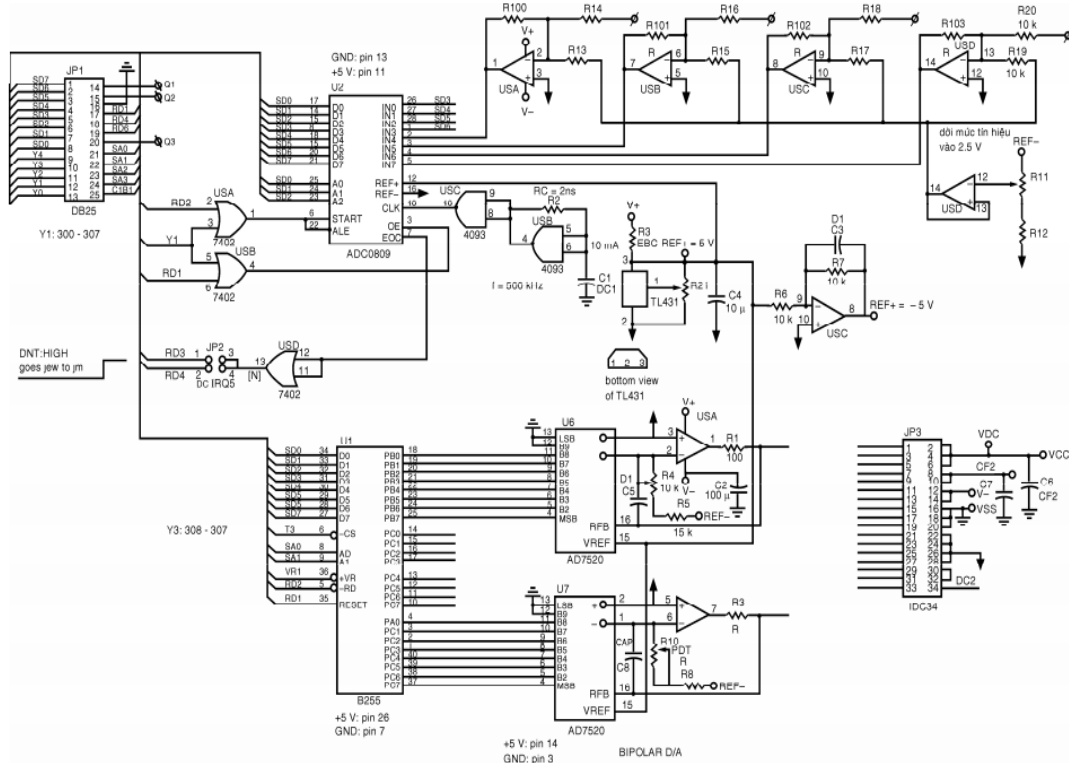
Hình 2.13 a và b trình bày mạch giao tiếp 16 bit dùng 2 IC 8255.



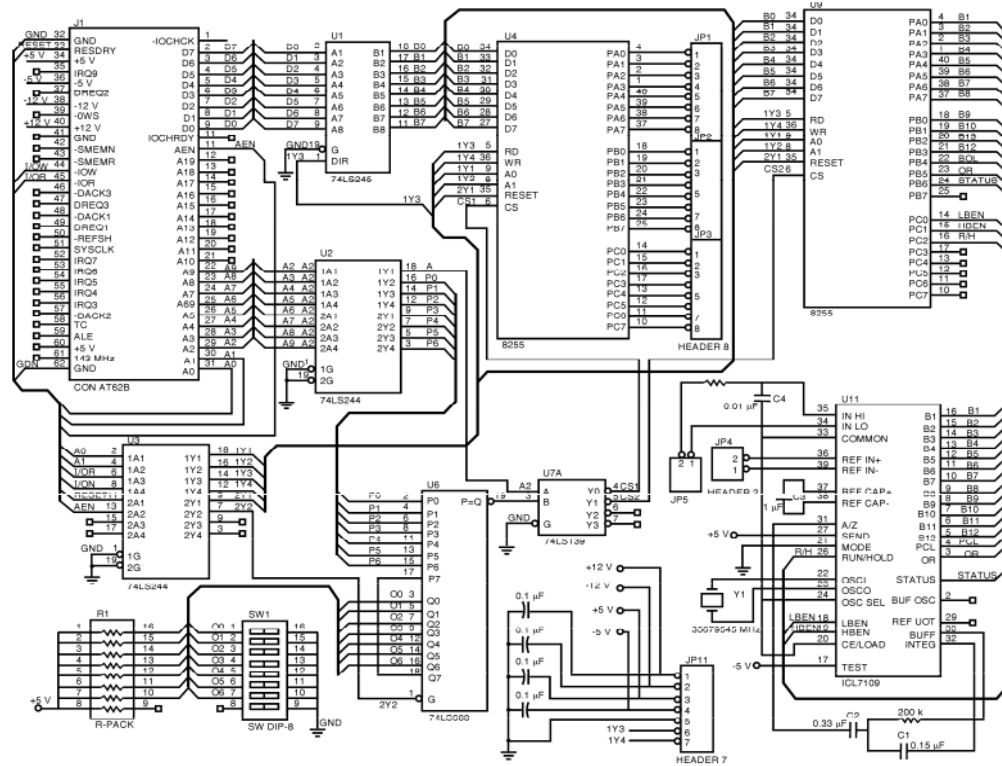
Hình 2.9: Peripheral adapter 16 bit



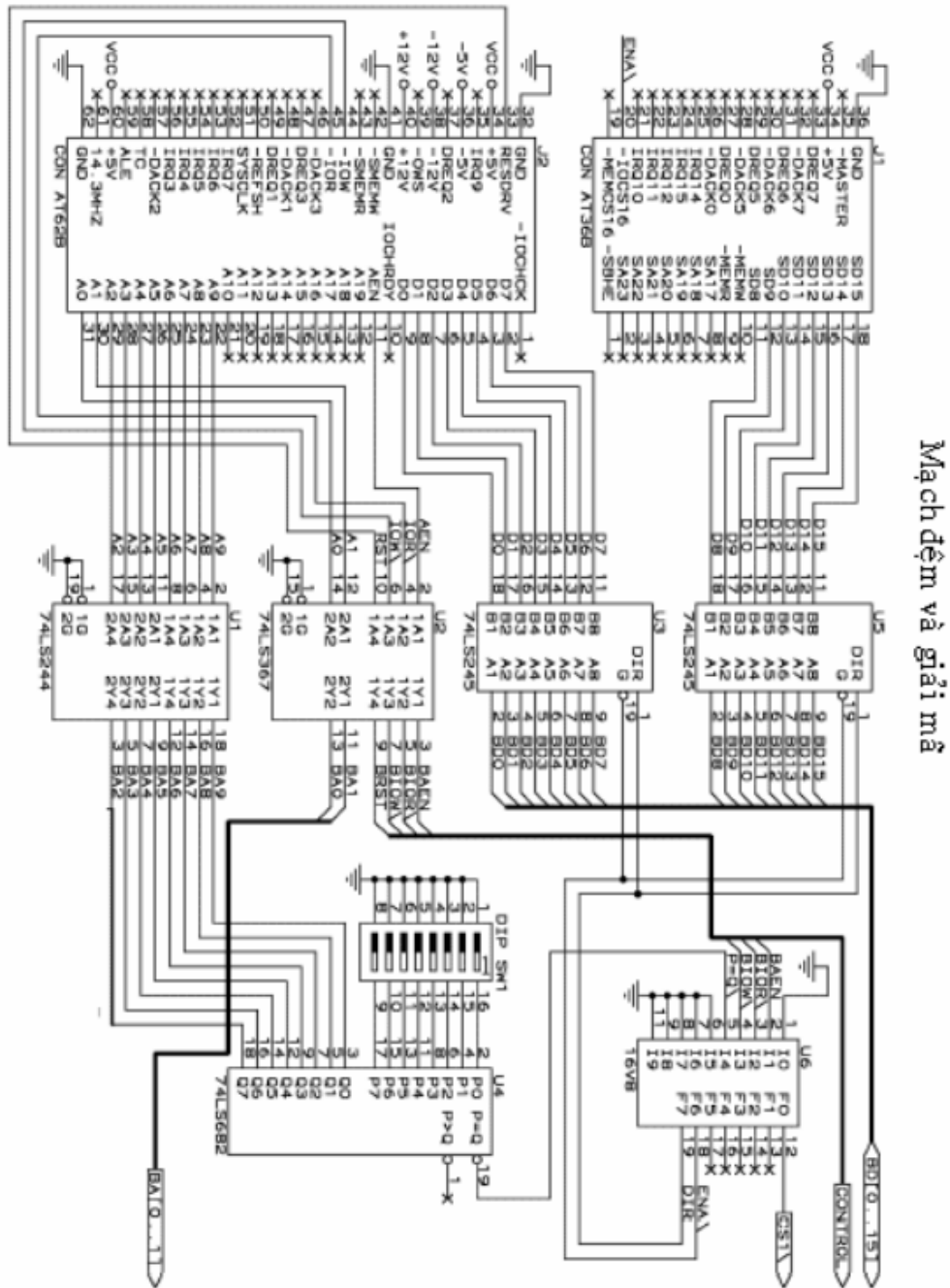
Hình 2.10: 8 bit Interface Card



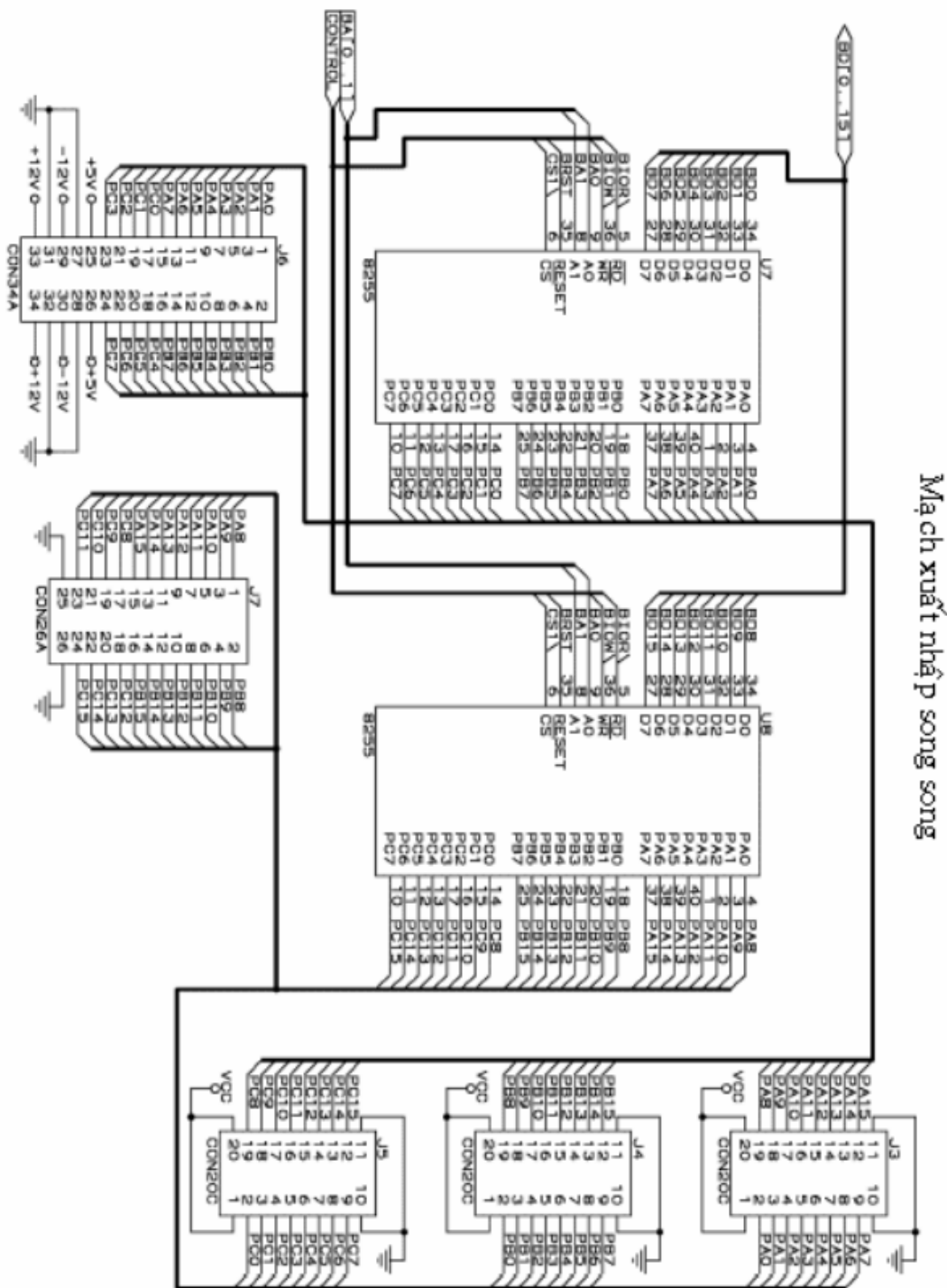
Hình 2.11: Mạch chuyển đổi AD và DA



Hình 2.12: Card chuyển đổi AD 12 bit



Hình 2.13a: Mạch giao tiếp song song 16 bit slot ISA, phần đệm và giải mã địa chỉ.



Hình 2.13b: Mạch giao tiếp song song 16 bit slot ISA, phần xuất nhập ngoại vi



II – GIAO TIẾP QUA RÃNH CẮM PCI

Rãnh cắm PCI (peripheral Component Interconnect) có màu trắng trên main board cho phép giao tiếp ngoại vi 32 hay 64 bit, vận tốc nhanh lên đến 132MB/s so với rãnh cắm ISA 16 bit có vận tốc 35 MB/s.

Nhờ vận tốc cao nên rãnh PCI thường dùng cho card màn hình, sau đó nó được sử dụng để cho các card khác như card mạng, modem nội, âm thanh ... và dần dần các main board đời mới không còn 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 2 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 lắp ráp card giao tiếp PCI khó thực hiện mà phải dùng card chính hãng.

Năm 1998 các hãng Compaq, Hewlett – Packard, 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 do 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 hai 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 33MHz, 66MHz ... |
| RST# | Tín hiệu reset |
| ADO ÷ AD31 | Tuyến địa chỉ khi FRAME# ở mức thấp. |
| C/BE0 ÷ 3# BUS (Command Bytes Enables) | Cho biết loại của truyền dữ liệu (đọc/viết bộ nhớ, ngoại vi ...) |
| PAR | Kiểm tra parity của ADO 31 và C/BE0 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 dữ liệu do initiator bắt đầu thông qua C/BE và IRDY còn target trả lời thông qua TRDY# và STOP# và STOP#. |
| LOCK# | Tín hiệu initiator báo dành riêng một số địa chỉ của target. |
| IDSEL (Initialiation DeviaSelect) | 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# (<i>SnoopBackoff</i>) SDONE (<i>Snoop done</i>) | Dùng cho card memory |
| PRSENT 1 ÷ 2# | Cho biết board có cắm vào slot và công suất tiêu thụ của board đó. |
| CLKRUN# (<i>Clock Running</i>) | Cho phép điều khiển xung nhịp CLK. |
| MGEN (<i>66 MHz enable</i>) | Cho biết xung nhịp là 33MHz hay 66 MHz. |
| 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 REQ64# và ACK64#, PAR64 |
| REQ 64# (<i>Request 64 bit transfer</i>) | |
| ACK 64# (<i>Acknowledge 64 bit transfer</i>) | |
| TCK (<i>Test clock</i>) TDI (<i>Test data input</i>) TDO (<i>Test output</i>) TMS (<i>Test mode Select</i>) TRST# (<i>Test Reset</i>) | Các tín hiệu thử |

Đặc điểm của các board cắm trên tuyến PCI là dữ liệu có thể truyền không thông qua CPU chủ do đó vận tốc xử lý tín hiệu nhanh hơn.

Card DT 300 của hãng Data Translation cho phép đổi 16 tín hiệu analog ra số phân giải 16 bit với vận tốc 250.000 mẫu/sec. Đổi ra số analog 2 kênh 16 bit, xuất nhập digital 32bit.

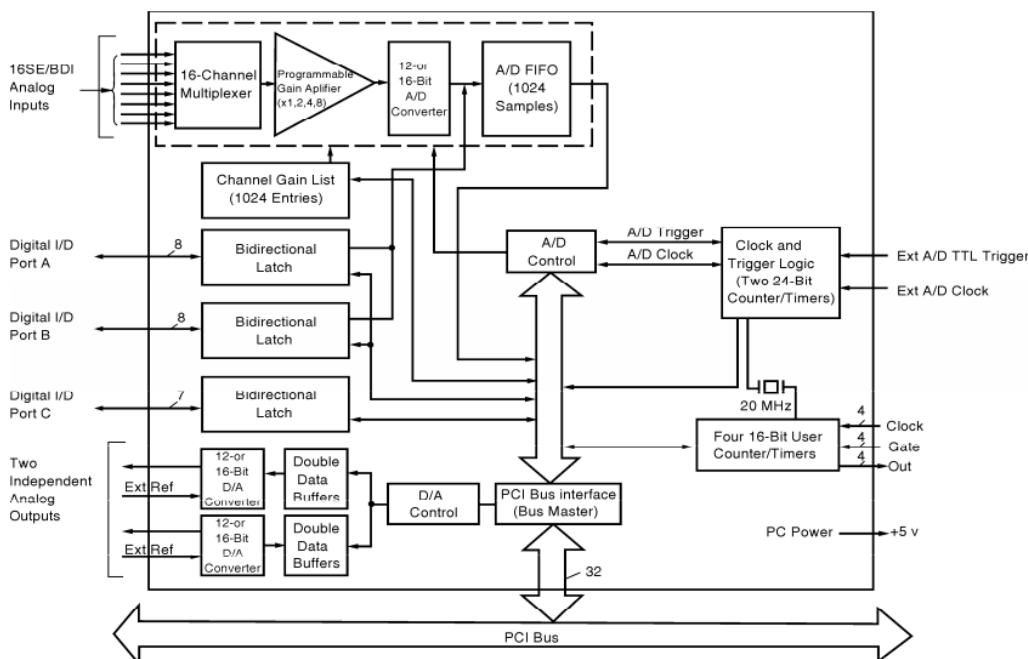
| Pin | 5V system environment | | Pin | 3.3V system environment | | comments |
|-----|-----------------------|-----------|-----|-------------------------|-------------|--------------|
| | side B | side A | | side B | side A | |
| 1 | -12V | TRST# | 1 | -12V | TRST# | 32-bit start |
| 2 | TCK | +12V | 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 | PRSENT1# | Reserved | 9 | PRSENT1# | Reserved | |
| 10 | Reserved | +5V (I/O) | 10 | Reserved | +3.3V (I/O) | |
| 11 | PRSENT2# | Reserved | 11 | PRSENT2# | 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 | 33 | 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 | 32-bit end |
| | Connector Key | | | Connector Key | | 64-bit spacer |
| | Connector Key | | | Connector Key | | 64-bit 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 |

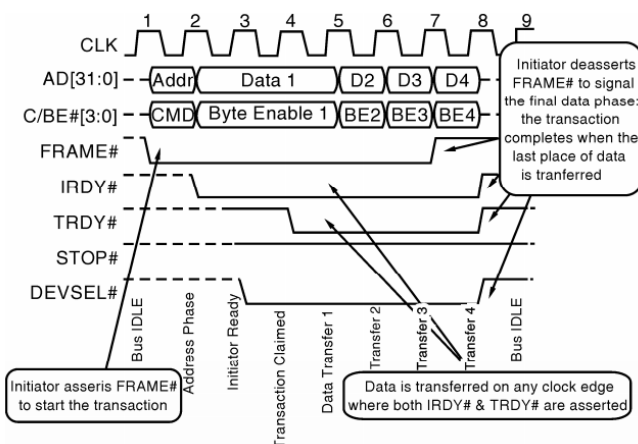
Bảng 2.1: Sơ đồ chân rãnh cắm PCI 64 bit



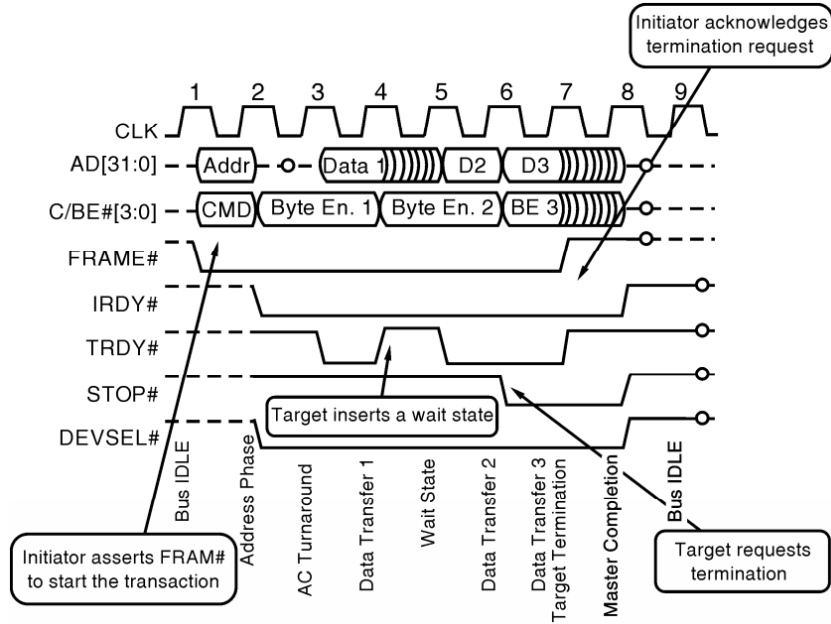
Hình 2.1: Sơ đồ khối card DT300 của hãng Data Translation

Bảng 2.2: Lệnh PCI (từ C/BE#)

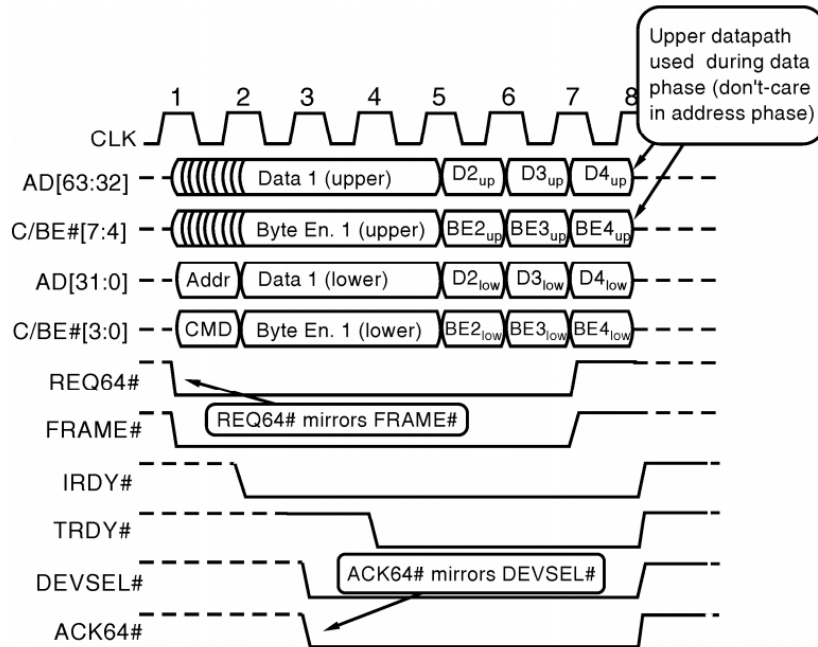
| C/BE# | Lệnh | C/BE# | Lệnh |
|-------|------------------------|-------|--------------------------|
| 0000 | Chấp nhận ngắt | 1000 | Dự trữ |
| 0001 | Chu kỳ đặc biệt | 1001 | Dự trữ |
| 0010 | Đọc Ngoại vi xuất nhập | 1010 | Đọc Cấu hình (Với IDSEL) |
| 0011 | Ghi Ngoại vi xuất nhập | 1011 | Ghi Cấu hình (Với IDSEL) |
| 0100 | Dự trữ | 1100 | Đọc nhiều ô nhớ |
| 0101 | Dự trữ | 1101 | Chu kỳ địa chỉ kép |
| 0110 | Đọc bộ nhớ | 1110 | Đường đọc bộ nhớ |
| 0111 | Ghi bộ nhớ | 1111 | Ghi bộ nhớ và cấm |



Hình 2.2: Giảm đồ thời gian truyền 4 từ kép 32 bit từ Initiator (master) đến target (slave)



Hình 2.3: Giảm độ thời gian đọc 2 từ kép từ target đến Initiator

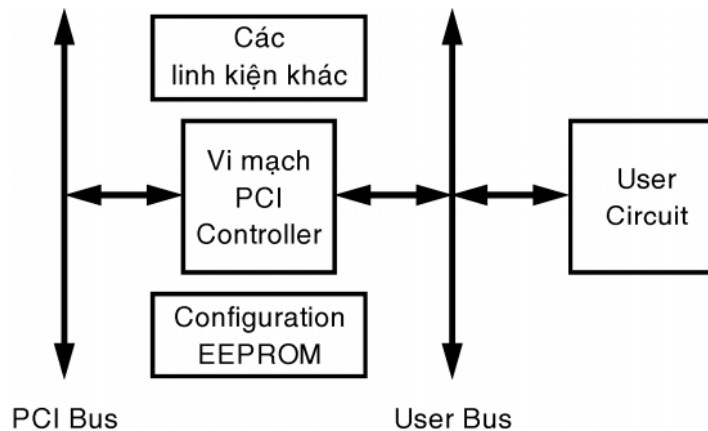


Hình 2.4: Giảm độ thời gian truyền 64 bit

III – THIẾT KẾ CARD GIAO TIẾP RẪNH PCI.

Trong mục trước chúng ta đã nghiên cứu các card ISD, việc thiết kế các card này tương đối đơn giản. Do các main board đời mới không còn hỗ trợ tuyến ISA nên 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 tạo 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 dàng hơn nếu dùng các bộ PCI development kit có sẵn. Các bộ kit này giúp tạo các ứng dụng PCI khác nhau cùng với software kèm theo.

Cấu trúc chung card PCI như sau:



Hình 3.1 – Sơ đồ cấu trúc chung của một card PCI

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 Technology, ISPLSI 1032E của Lattice ... Phần tử thứ hai là EEPROM dùng để chứa thông tin về Card phục vụ chp 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 để kết nối với các linh kiện thông thường.

Chương 2

CARD THU THẬP TÍN HIỆU VÀ ĐIỀU KHIỂN

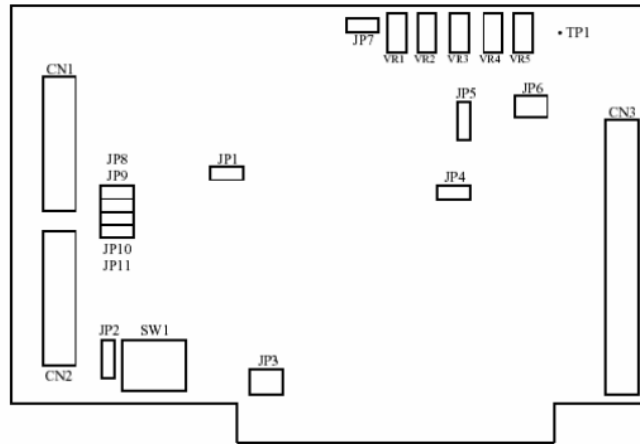
2.1. CARD PCL-818L ADVANTECH.

2.1.1. Sơ đồ khối, chức năng và cấu trúc của Card.

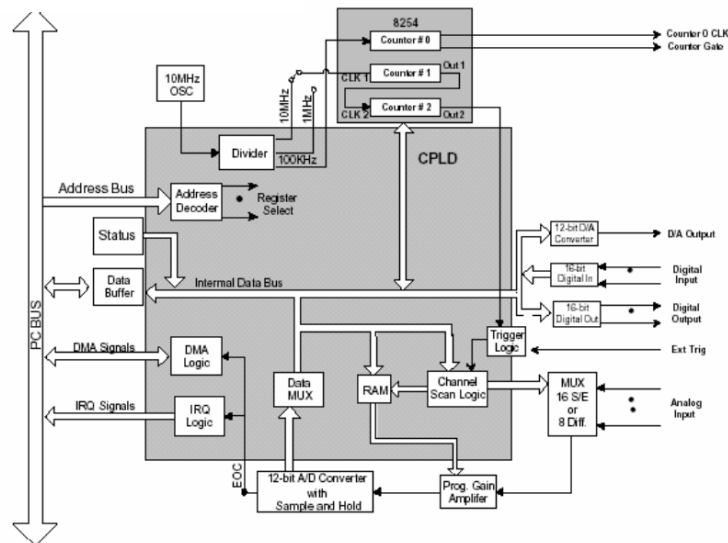
PCI-818L là card ISA có nhiều chức năng dùng để đo lường và điều khiển. Do tính năng ưu việt của card, việc tìm hiểu hoạt động của nó rất cần thiết để tiếp cận kỹ thuật thu thập số liệu bằng máy tính DSA (Data Acquisition System).

Các chức năng chính của Card:

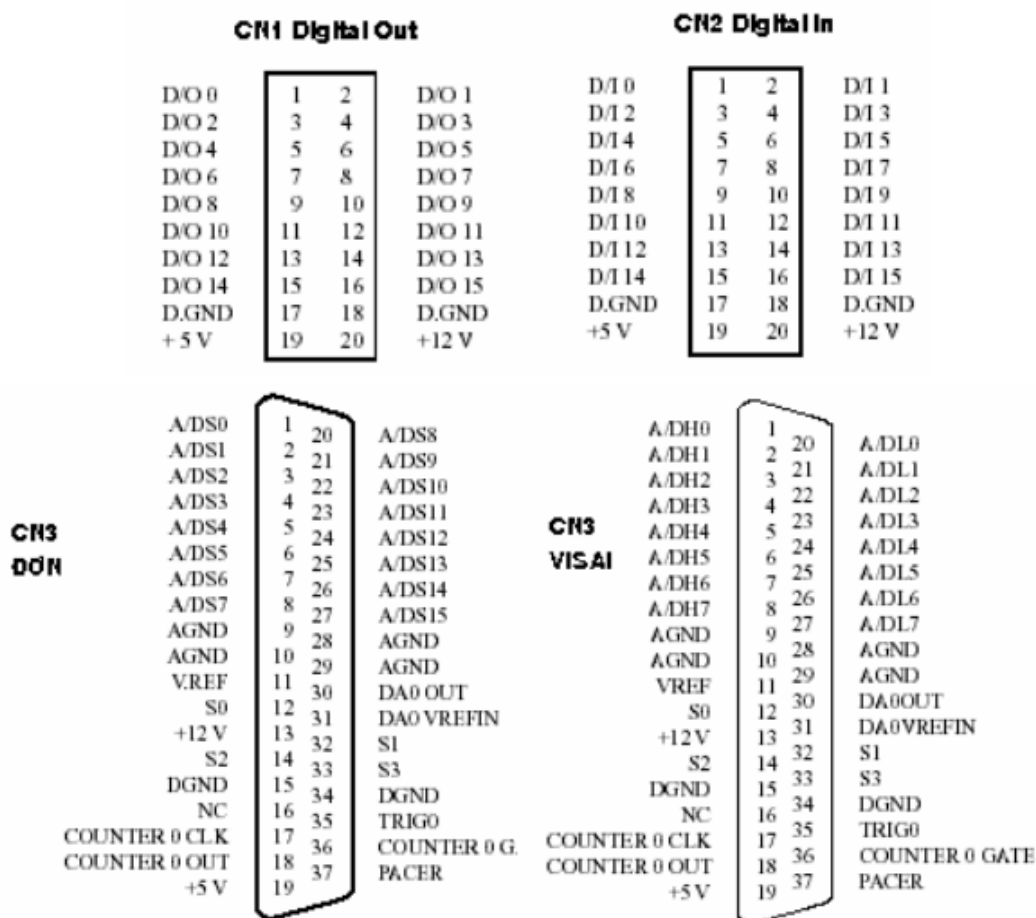
- Chuyển đổi A/D 16 kênh 12 bit với tốc độ lấy mẫu 40000/s.
- Chuyển đổi D/A 1 kênh 12 bit.
- 16 đầu vào digital TTL.
- 16 đầu ra digital TTL.
- 1 bộ đếm/ định thời 16 bit cho người dùng.



Hình 2.1- Vị trí các cầu nối, biến trở và đầu nối của Card PCI-818L Advantech.



Hình 2.2 – Sơ đồ khối card PCI-818L Advantech



Hình 2.3 – Sơ đồ các chân của Card PCI-818L Advantech

Trong đó:

A/DS: Vào Analog (đơn).

A/DH: Vào Analog cao (vi sai).

A/DL: Vào Analog thấp (vi sai).

AGND: Mass Analog.

D/A: Ra Analog.

DO: Ra Digital.

DI: Vào Digital

DGND: Mass Digital và nguồn.

CLK: Clock Counter 0 của 8254.

GATE: Gate Counter 0 của 8254.

OUT: Tín hiệu ra của 8254.

VREF: Nguồn chuẩn trong.

VREF in: Nguồn chuẩn ngoài.

2.1.2. Các thanh ghi của Card.

PCI-818L có 16 thanh ghi, địa chỉ gốc có thể chọn bởi công tắc Sw₁, gồm 6 tiếp điểm, chọn các đường địa chỉ SA₄ ÷ SA₉, thường đặt địa chỉ gốc là BASE 300h.

Các thanh ghi của Card được trình bày trong bảng sau:



| Địa chỉ | Đọc | Viết |
|---------|--------------------------|-------------------------------------|
| BASE+0 | Byte thấp A/D và số kênh | Kích mềem A/D |
| BASE+1 | Byte cao A/D | Điều khiển tầm A/D |
| BASE+2 | Quét kênh MUX | Quét kênh MUX và chỉ tầm điều khiển |
| BASE+3 | Byte thấp Digital Input | Byte thấp Digital Output |
| BASE+4 | N/A | Byte thấp Analog Output |
| BASE+5 | N/A | Byte cao Digital Output |
| BASE+6 | N/A | N/A |
| BASE+7 | N/A | N/A |
| BASE+8 | Trạng thái | Xóa yêu cầu ngắt |
| BASE+9 | Điều khiển | Điều khiển |
| BASE+10 | N/A | Cho phép Counter |
| BASE+11 | Byte cao Digital Output | Byte cao Digital Output |
| BASE+12 | Counter 0 | Counter 0 |
| BASE+13 | Counter 1 | Counter 1 |
| BASE+14 | Counter 2 | Counter 2 |
| BASE+15 | N/A | Điều khiển counter |

Bảng 2.1 – Các thanh ghi của Card PCI-818L Advantech.

a. Các thanh ghi cơ số +0 và +1.

Khi đọc thanh ghi:

| BASE+0 (đọc) - Chứa Byte thấp A/D và số kênh | | | | | | | | |
|--|------|------|-----|-----|-----|-----|-----|-----|
| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| Value | AD3 | AD2 | AD1 | AD0 | C3 | C2 | C1 | C0 |
| BASE+1 (đọc) - Byte cao A/D | | | | | | | | |
| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| Value | AD11 | AD10 | AD9 | AD8 | AD7 | AD6 | AD5 | AD4 |

Trong đó: $AD_{11} \div AD_0$ là dữ liệu Analog đổi sang Digital.
 $C_3 \div C_0$ là số đầu vào Analog tương ứng.

Khi ghi vào thanh ghi BASE +0: Kích mềem bộ A/D, khởi động ADC. Có thể kích bộ A/D bằng phầm mềem hoặc bằng xung Clock trên board (pacer), có thể bằng xung ngoài. Các bit 0 và 1 trong BASE+9 sẽ chọn nguồn kích. Nếu chọn kích mềem thì ta chỉ việc ghi vào thanh ghi BASE+0 bất kỳ một giá trị nào cho mỗi lần kích. Nếu là kích ngoài thì cầu nối JP3 chọn nguồn kích là DI0 (CN2) hay TRIG0 (CN3).

Điều khiển tầm A/D (BASE+1):

Mỗi kênh A/D đều có một tầm điện áp đầu vào riêng cho nó, được điều khiển bởi mã lưu trữ trong RAM của PCI-818L và được đặt bởi cầu nối JP7. Nếu muốn thay đổi tầm cho một kênh, chọn kênh như là kênh Start ở thanh ghi BASE+2, quét kiểu MUX, rồi ghi mã vào bit 0 và 1 của thanh ghi BASE+1.



| BASE+1 (Ghi) - Điều khiển tằm A/D | | | | | | | | |
|-----------------------------------|-----|-----|-----|-----|-----|-----|----|----|
| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| Value | N/A | N/A | N/A | N/A | N/A | N/A | G1 | G0 |

| Tầm điện áp ngõ vào | | Mã tằm | |
|---------------------|----------|--------|---|
| JP7 = 5 | JP7 = 10 | | |
| ±5V | ±10V | 0 | 0 |
| ±2.5V | ±5V | 0 | 1 |
| ±1,25V | ±2,5V | 1 | 0 |
| ±0,625V | ±1,25V | 1 | 1 |

Mã tằm và JP7

b. Thanh ghi quét phân kênh (BASE+2).

- Đọc/ghi BASE+2 để điều khiển/đọc số kênh A/D được quét. Nửa byte cao chỉ kênh Stop, nửa byte thấp chỉ kênh Start. Việc quét phân kênh MUX được khởi động đến kênh Start khi ghi vào thanh ghi này. Mỗi Trigger A/D sẽ chuyển đến kênh đo tiếp theo.

- Khi kích chuyển đổi liên tục, MUX sẽ quét từ kênh Start đến kênh Stop rồi lặp lại từ đầu. Ví dụ, nếu kênh Start là 4 và Stop là 7 thì quét tuần tự 4,5,6,7. Nếu cài đặt ở chế độ 8 đầu vào vì sai thì các bit CH₃ và CL₃ phải là 0.

- Nếu chỉ chọn một kênh để biến đổi A/D thì cài đặt kênh Start và Stop cùng một trị số là số của kênh cần thực hiện biến đổi A/D.

| BASE+2 (Viết) - Các kênh quét đầu và cuối | | | | | | | | |
|---|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| Value | CH3 | CH2 | CH1 | CH0 | CL3 | CL2 | CL1 | CL0 |

- CH₃ ÷ CH₀ là kênh Stop. CL₃ ÷ CL₀ là kênh Start.

- Nửa bit thấp của thanh ghi quét phân kênh CL₃ đến CL₀ cũng có tác dụng như một con trỏ khi lập trình tầm điện áp vào A/D. Khi đặt kênh Start là N, thì mã tằm viết vào thanh ghi BASE+1 là cho kênh N.

c. Các thanh ghi xuất/nhập số (BASE+3).

- PCL-818L có 16 đầu vào số và 16 đầu ra số riêng biệt. Các kênh I/O này dùng chung Port có địa chỉ BASE+3 và BASE+11.

Khi đọc:

| BASE+3 (đọc port) - Byte thấp Digital Input | | | | | | | | |
|---|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| Value | D17 | D16 | D15 | D14 | D13 | D12 | D11 | D10 |

| BASE+11 (đọc port) - Byte cao Digital Input | | | | | | | | |
|---|------|------|------|------|------|------|-----|-----|
| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| Value | DI15 | DI14 | DI13 | DI12 | DI11 | DI10 | DI9 | DI8 |

Khi ghi:

| BASE+3 (viết port) - Byte thấp Digital Input | | | | | | | | |
|--|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| Value | DO7 | DO6 | DO5 | DO4 | DO3 | DO2 | DO1 | DO0 |

| BASE+11 (viết port) - Byte cao Digital Input | | | | | | | | |
|--|------|------|------|------|------|------|-----|-----|
| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| Value | DO15 | DO14 | DO13 | DO12 | DO11 | DO10 | DO9 | DO8 |

d. Thanh ghi xuất Analog D/A (BASE+4/5).

Khi ghi:

| BASE+4 - Byte thấp ngõ ra D/A | | | | | | | | |
|-------------------------------|-----|-----|-----|-----|----|----|----|----|
| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| Value | DA3 | DA2 | DA1 | DA0 | x | x | x | x |

| BASE+5 - Byte cao ngõ ra D/A | | | | | | | | |
|------------------------------|------|------|-----|-----|-----|-----|-----|-----|
| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| Value | DA11 | DA10 | DA9 | DA8 | DA7 | DA6 | DA5 | DA4 |

Trong đó: DA₁₁ ÷ DA₀ là dữ liệu Digital sang Analog.

- Tầm điện áp ra được chọn nhờ cầu nối JP4 và JP5. Nếu JP4 đặt ở IN thì JP5 chọn nguồn chuẩn trong là -5V hay -10V, áp ra của D/A sẽ là 0 đến +5V hay 0 đến +10V. Nếu JP4 đặt ở EXT thì điện áp ra D/A là kết quả nhân số Digital trong hai thanh ghi với điện áp đặt vào chân số 31 VREF IN (-10V .. +10V) của đầu nối CN₃ chia cho 4095.

e. Thanh ghi trạng thái BASE+8.

- Đọc BASE+8 để nhận thông tin về cấu hình và hoạt động A/D.

- Ghi vào BASE+8 một giá trị bất kỳ thì nó sẽ xóa bit INT của BASE+8, còn những bit dữ liệu khác không đổi.

| BASE+8 - Trạng thái A/D | | | | | | | | |
|-------------------------|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| Value | EOC | N/A | MUX | INT | CN3 | CN2 | CN1 | CN0 |

- EOC: End of Conversion.

EOC = 0: Đã đổi xong, kết quả đổi chứa trong BASE+0 và BASE+1.

EOC = 1: Đang chuyển đổi A/D.

- MUX: Chọn 8 kênh vi sai hoặc 16 kênh đơn, phản ánh vị trí cầu nối JP6.

MUX = 0: 8 kênh vi sai.

MUX = 1: 16 kênh đơn.

- INT: Tín hiệu ngắt.

INT = 0: Dữ liệu không có giá trị (không có một biến đổi nào kể từ khi bit INT bị xóa).

INT = 1: A/D đã biến đổi xong, dữ liệu có giá trị.

Nếu bi INTE = 1 (BASE+9) thì khi đổi xong một kênh, tín hiệu ngắt sẽ gửi đến PC qua chân IRQn (IRQn được chọn bởi các bit I₂ ÷ I₀ trong BASE+9). Dù thanh ghi trạng thái A/D là chỉ đọc nhưng khi ghi vào nó một giá trị bất kỳ sẽ xóa bit INT, còn các bit khác không đổi.

- CN₃ ÷ CN₀: Khi EOC = 0 thì các bit này chứa số kênh kế tiếp sẽ được biến đổi.

- Lưu ý: Nếu kích bộ A/D bằng xung clock trên board (pacer) hoặc xung ngoài thì phần mềm phải kiểm tra bit INT trước khi đọc dữ liệu (không phải bit EOC). EOC có thể bằng 0 trong 2 trường hợp:

- Biến đổi đã hoàn tất.
- Không có một biến đổi nào đã được bắt đầu.

Do đó phần mềm phải đợi tín hiệu INT = 1 trước khi đọc số liệu chuyển đổi. Rồi cần phải xóa bit INT bằng cách ghi bất kì giá trị nào vào thanh ghi trạng thái BASE+8.

f. Thanh ghi điều khiển (BASE+9).

Đọc/ghi thanh ghi BASE+9 để nhận/cung cấp thông tin về chế độ hoạt động của PCL-818L.



| BASE+9 - Điều khiển chế độ hoạt động | | | | | | | | |
|--------------------------------------|------|----|----|----|----|------|-----|-----|
| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| Value | INTE | I2 | I1 | I0 | x | DMAE | ST1 | ST0 |

- INTE: Cấm/cho phép ngắt.
INTE = 0: Cấm ngắt.
INTE = 1: Cho phép ngắt.
- Nếu DMAE = 0: PCL-818L sẽ phát ra một tín hiệu ngắt khi nó hoàn tất một chuyển đổi A/D. Vậy cấu hình INTE = 1. DMAE = 0 dùng để báo cho CPU biết bằng ngắt là đã đổi A/D xong.
- Nếu DMAE = 1: PCL-818L sẽ phát một tín hiệu ngắt khi nó nhận một tín hiệu đếm tràn T/C (Terminal Count) từ bộ điều khiển DMA của máy tính để chỉ rằng chuyển đổi truyền DMA đã hoàn tất. Truyền DMA bị dừng bởi ngắt gây ra bởi tín hiệu T/C. (Xem DMAE bên dưới).
- I₂ ÷ I₀: Chọn số ngắt cho data interrupt hoặc truyền data DMA (Không được trùng với số ngắt của thiết bị khác). Ba bit này chọn số ngắt từ 2 (010) đến 7 (111).
- DMAE: Cấm/cho phép PCL-818L truyền DMA.
DMAE = 0: Cấm truyền DMA.
DMAE = 1: Cho phép truyền DMA. Mỗi biến đổi A/D sẽ khởi động hai tín hiệu yêu cầu ngắt liên tiếp. Các tín hiệu này cho phép bộ điều khiển DMA 8237 truyền 2 byte dữ liệu chuyển đổi A/D từ PCL-818L đến bộ nhớ. Chọn kênh DMA 1 hay 3 nhờ cầu nối JP1.
- Lưu ý: Phải lập trình bộ điều khiển DMA và thanh ghi trong DMA 8237 của máy tính trước khi đặt DMAE = 1.
- ST₁ ÷ ST₀: Chọn nguồn kích.

| Nguồn kích | ST1 | ST0 |
|------------|-----|-----|
| Kích mềm | 0 | X |
| Kích ngoài | 1 | 0 |
| Kích Pacer | 1 | 1 |

g. Thanh ghi cho phép đếm định thời (BASE+10).

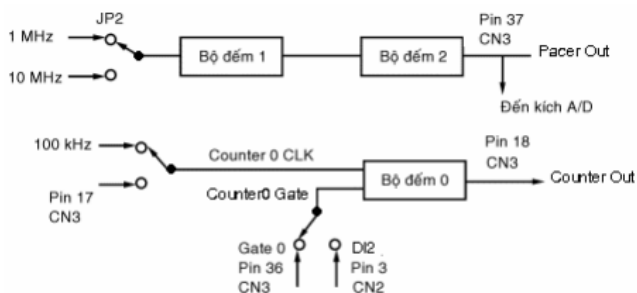
Card PCL-818L của vi mạch 8254 gồm ba bộ đếm 0, 1, 2 và sử dụng hai bộ đếm 1, 2 còn bộ đếm 0 cho người dùng. Xung nhịp cho bộ đếm 1 có thể chọn là 10Mhz hay 1Mhz nhờ cầu nối JP2, đầu ra của bộ đếm 1 nối vào ngõ nhịp của bộ đếm 2, đầu ra của bộ đếm 2 đưa vào kích A/D (kích pacer), vậy tần số kích phụ thuộc vào tần số xung nhịp và số ghi vào 2 bộ đếm 1, 2.

Tần số của pacer là $f_{clk}/(Div_1 * Div_2)$ với $f_{clk} = 1Mhz$ hay $10Mhz$, Div_1 và Div_2 là số đặt trong bộ đếm 1 và bộ đếm 2.

Bộ đếm 0 có đầu vào Clk là 100Khz hay xung nhịp ngoài ở chân 17 CN₃ chọn nhờ BASE+10, chân GATE là DI₂ (chân 3 CN₂) hay GATE0 (chân 36 CN₃) chọn nhờ đầu nối JP3.

Ghi vào BASE+10 để cho phép hoặc cấm bộ đếm/định thời của PCL-818L tạo xung.

| BASE+10 - Cho phép pacer | | | | | | | | |
|--------------------------|----|----|----|----|----|----|-----|-----|
| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| Value | x | x | x | x | x | x | TC1 | TC0 |





- TC0: Cấm/cho phép pacer.

TC0 = 0: Cho phép pacer.

TC0 = 1: Pacer được điều khiển bởi TRIG0 (chân 35 CN₃). Tín hiệu này chặn xung trigger gửi từ pacer đến bộ A/D khi nó bằng 0.

- TC1: Chọn chế độ nguồn xung nhịp đầu vào Counter 0.

TC1 = 0: Counter 0 nhận xung clock ngoài (chân 17 CN₃).

TC1 = 1: Counter 0 nhận xung clock 100Khz ở bên trong.

h. Các thanh ghi điều khiển và đọc/ghi bộ đếm.

Bộ định thời 8254 sử dụng 4 thanh ghi BASE+12, BASE+13, BASE+14, BASE+15. Các chức năng của các thanh ghi này được liệt kê trong bảng sau:

| Thanh ghi | Chức năng | Thanh ghi | Chức năng |
|-----------|-------------------|-----------|-------------------|
| BASE+12 | Counter 0 đọc/ghi | BASE+14 | Counter 2 đọc/ghi |
| BASE+13 | Counter 1 đọc/ghi | BASE+15 | Điều khiển bộ đếm |

Do bộ đếm 8254 có cấu trúc 16 bit, nên mỗi dữ liệu đọc/ghi được chia làm 2 byte: byte thấp (LSB), byte cao (MSB). Để tránh phạm lỗi đọc/ghi sai, cần chú ý thao tác đọc ghi từng đôi (tức là mỗi lần đọc/ghi 2 byte) và theo đúng thứ tự byte.

2.1.3. Chuyển đổi A/D, D/A & DO.

a. Chuyển đổi A/D.

Có thể kích đổi A/D bằng phần mềm, bằng xung ngoài hay bằng pacer on board. Bit 1 và 0 của thanh ghi BASE+9 sẽ chọn nguồn Trigger tương ứng.

Khi chọn kích thước pacer tần số kích từ 2.5Mhz đến 71 phút một xung. Khi chọn kích ngoài, nguồn kích định bởi cầu nối JP3 chọn tín hiệu kích là TRIG0 (chân 35 CN₃) hay DI0 (chân 1 CN₂).

Việc truyền kết quả A/D cho máy tính có thể thực hiện bằng chương trình điều khiển, bằng ngắt hay DMA.

Các bước hình thành để chuyển đổi A/D với trigger bằng phần mềm và truyền kết quả A/D bằng chương trình điều khiển:

- Đặt tầm vào cho mỗi kênh A/D.
- Đặt kênh vào bằng cách chỉ rõ tầm quét kênh.
- Kích đổi A/D bằng cách ghi vào BASE+0 cho một số bất kỳ nào đó.
- Kiểm tra chuyển đổi đã kết thúc chưa bằng cách đọc bit EOC của thanh ghi trạng thái.
- Đọc kết quả chuyển đổi ở thanh ghi BASE+0 và BASE+1.
- Chuyển dữ liệu từ số nhị phân thành số nguyên.

b. Chuyển đổi D/A.

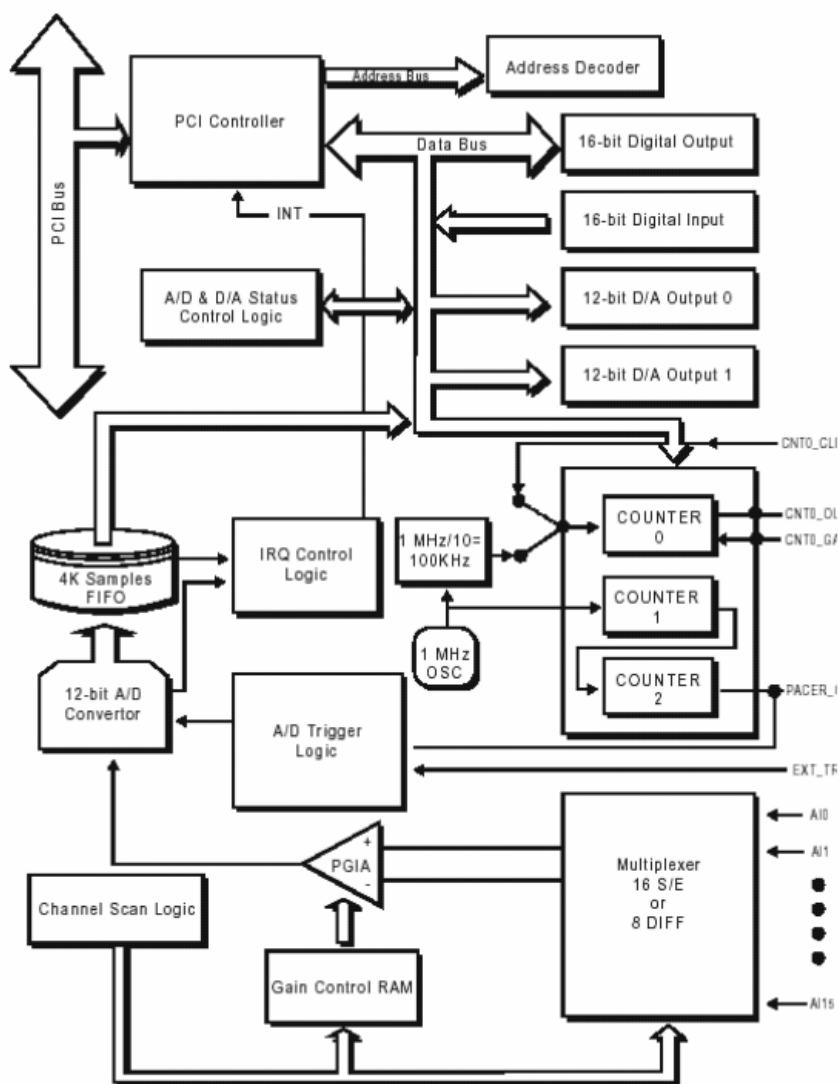
Ghi vào thanh ghi BASE+4 và BASE+5. Khi ghi dữ liệu cho kênh D/A phải ghi byte thấp trước. Byte thấp này được giữ tạm thời trong một thanh ghi và không được xuất ra. Sau khi ghi xong byte cao thì khi đó byte cao và byte thấp được kết hợp lại để chuyển thành D/A.

c. Digital input và output (DI&DO).

Đọc DI từ thanh ghi BASE+3 (byte thấp) và thanh ghi BASE+11 (byte cao). Sau khi đọc dữ liệu sẽ không được chốt, đường vào sẽ ở trạng thái thứ ba. Có thể xuất ra DO cũng bằng cách dùng hai thanh ghi BASE+3 và BASE+11 này. Thanh ghi sẽ chốt giá trị sẽ ghi (có thể đọc lại). Để an toàn nên ghép nối các đầu vào/ra digital thông qua ghép quang.

2.2. Card PCI 1710.

Card PCI 1710 có chức năng tương tự Card PCL-818L nhưng được gắn vào Slot PCI.
Sơ đồ khối của Card như sau:



Hình 2.2. Sơ đồ khối Card PCI 1710

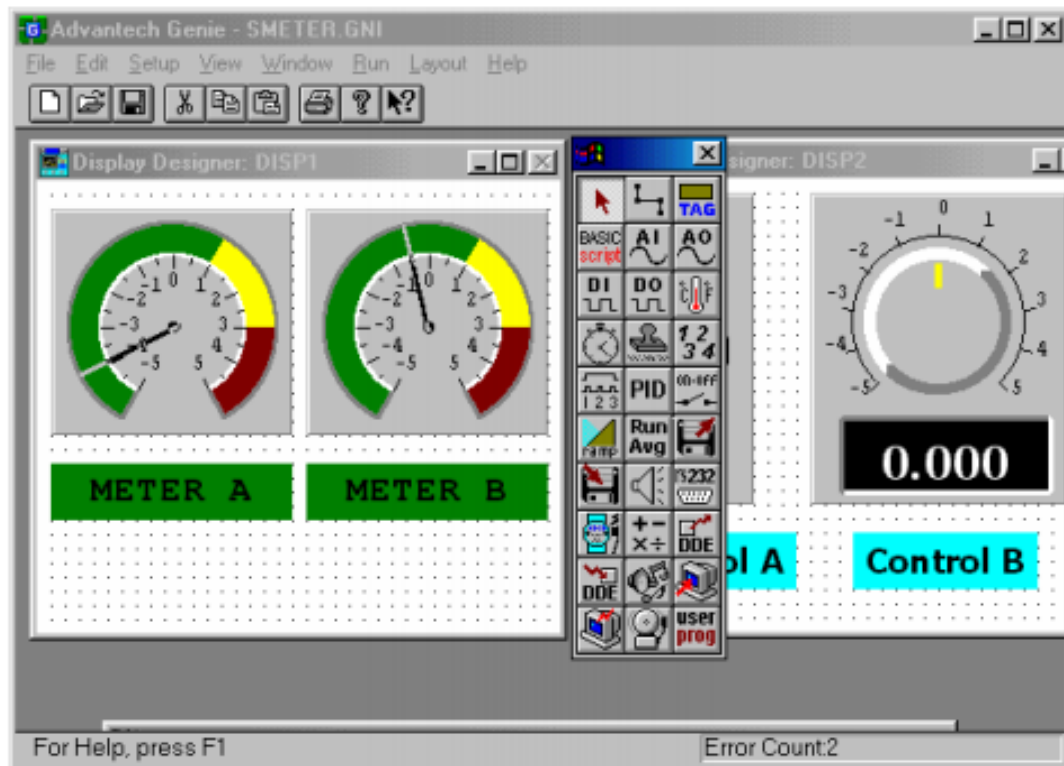
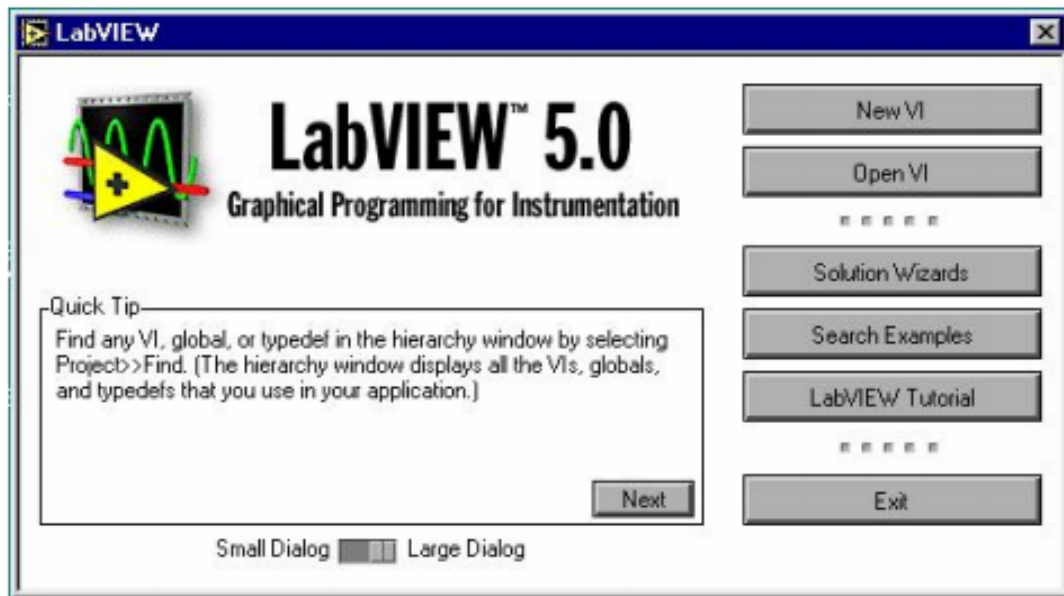
2.2.1. Các phần mềm lập trình cho Card.

Thay vì phải dùng ngôn ngữ lập trình để lập trình cho Card, ta có thể dùng các phần mềm chuyên dụng của hãng, có đặc tính.

- Giảm thời gian lập trình.
- Trình bày dữ liệu dưới dạng bảng biểu, đồ thị.
- Lưu trữ dữ liệu trong file.
- Kết nối với các phần mềm khác.

Các phần mềm thông dụng hiện nay thường được sử dụng như là: LabVIEW (Laboratory Virtual Instrument Engineering Workbench), VISA của National Instruments, GenieDAQ, VisiDAQ của Advantech

Dưới đây là giao diện một số phần mềm thu thập số liệu:



Hình 2.3. Giao diện một số phần mềm thu thập số liệu

Hình ảnh 2 loại Card thông dụng sử dụng 2 chuẩn khe cắm ISA và PCI trên main board máy tính trong ứng dụng đo lường và điều khiển máy tính đã nêu ở trên:



Card PCL818



Card PCI 1710

**GIÁO TRÌNH
ĐO LƯỜNG VÀ
ĐIỀU KHIỂN MÁY TÍNH**

Chương 3

GIAO TIẾP QUA CỔNG SONG SONG, NỐI TIẾP VÀ USB

3.1. Giao tiếp qua cổng song song.

a. Giới thiệu chung.

Cổng song song thường được 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 main board 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 50Kbyte/sec đến 150Kbyte/sec. Khi muốn nhập dữ liệu vào máy tính có thể dùng mode Nibble truyền 4 bit hay Byte truyền 8 bit.

Cổng EPP và ECP dùng thêm phần cứng hỗ trợ nên vận tốc truyền nhanh hơn, có thể đến 2Mbyte/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.

b. Cổng SPP.

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

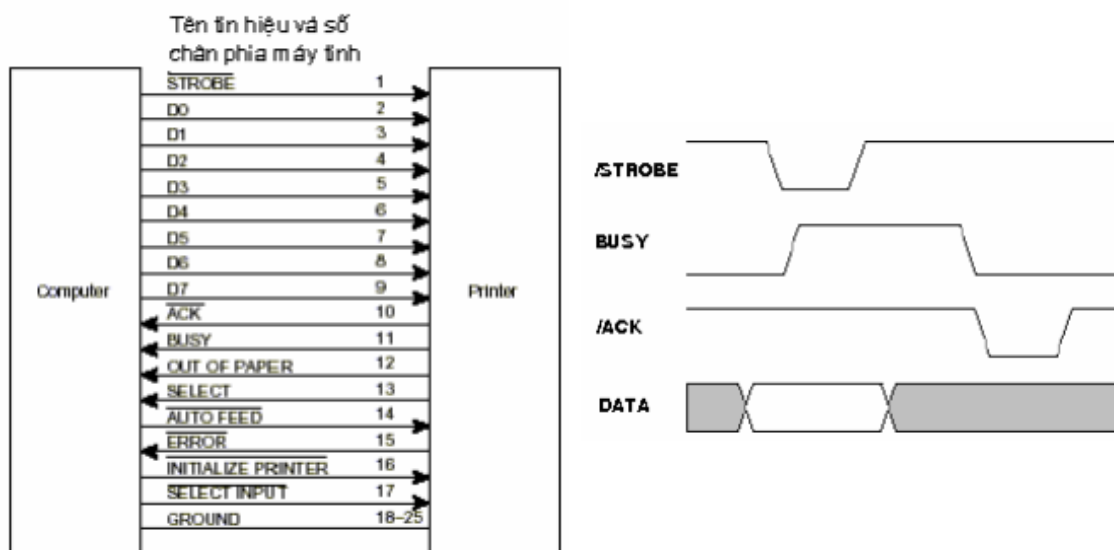
Bảng 3.1 cho sơ đồ chân và ý nghĩa các chân của cổng SPP khi dùng với máy in, dấu “/” có ý nghĩa tích cực thấp. Ví dụ, chân 15 là /Error hướng vào, nếu chân này xuống mức 0 là có lỗi. Cột đảo ghi chữ “C” tức là tín hiệu được đảo mức, ví dụ chân 17 khi đưa mức logic 0 ứng với chân này vào thanh ghi điều khiển thì ở chân 17 xuất hiện mức 1.

Đầu ra cổng song song tương thích với cổng TTL, dòng cấp và thu vào khoảng vài mA đến 16mA tùy theo kết cấu phần cứng (Công nghệ ASIC).

Giao tiếp qua cổng song song thường được thực hiện theo hình 3.1.

| 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 | | | |

Bảng 3.1. Sơ đồ chân của đầu nối phía máy tính (cổng SPP) và phía máy in



Hình 3.1. 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 tín hiệu Strobe rộng 1 sec và máy in đọc dữ liệu, đọc xong sẽ báo trở lại bằng ACK đảo rộng 5 sec. 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 điều khiển và thanh ghi trạng thái. 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 có địa chỉ gốc +1 là thanh ghi chỉ đọc dùng để nhận tín hiệu từ ngoài vào, có 5 tín hiệu vào.

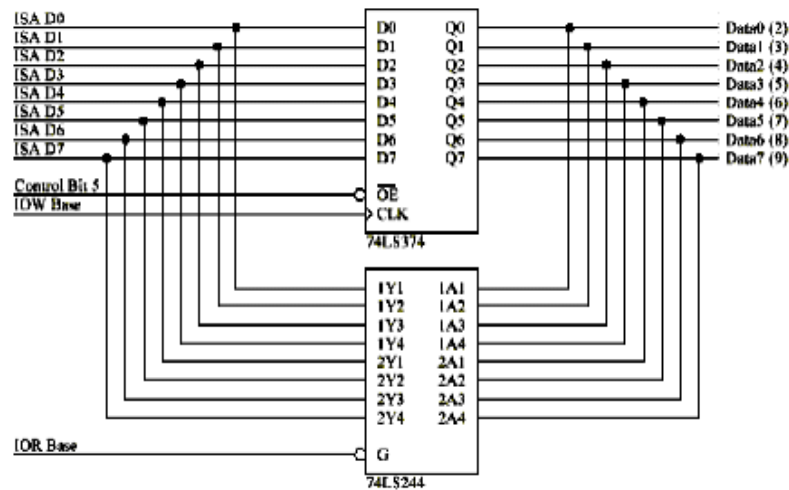
| D7 | | | | D0 | | | |
|------|------|-----------|--------|--------|------|---|---|
| Busy | /ACK | Paper Out | Select | /Error | /IRQ | X | X |
| 11 | 10 | 12 | 13 | 15 | | | |

Chú ý: 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 | Khởi động máy in /Init | Xuống hàng /Auto Linefeed | Kích /Strobe |
| | | | | 17 | 16 | 14 | 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.

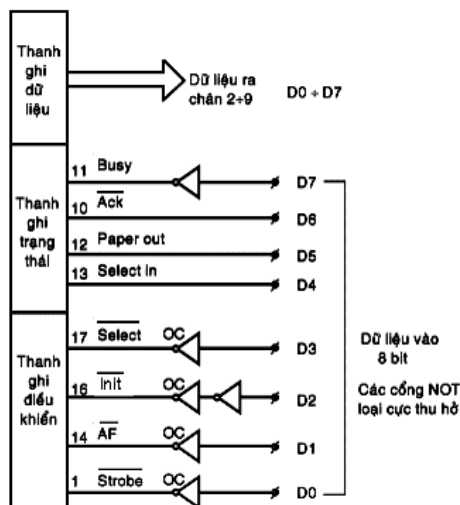


Hình 3.2. Sơ đồ thanh ghi dữ liệu 2 hướng

Một số main board (Hình 3.2) hỗ trợ giao tiếp 2 chiều qua thanh ghi dữ liệu, bit D₅ của thanh ghi dữ liệu 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 đầu ra cực thu hờ nên có thể nhận tín hiệu vào nếu trước đó ta đã nạp 8 bit sao cho các đầu ra ứng với thanh ghi này lên 1 (Hình 3.3). 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.



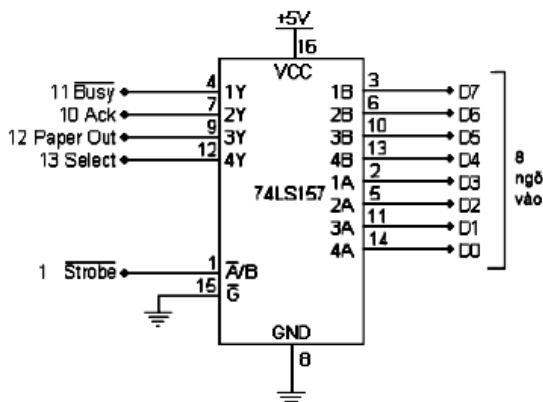
Hình 3.3. Giao tiếp song song 2 chiều qua cổng SPP

Trong một số trường hợp, các chân ra của thanh ghi điều khiển không dùng loại cực thu hờ 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 (Hình 3.4).

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 2 lần đọc ta được 1 byte rồi exclusive OR với 88H để đảo 2 bit D₀ và D₃ (do Busy đảo).



Hình 3.4. Vào 8 bit với 74LS157

Ví dụ lập trình:

Xuất trị của biến $oval_1$ ra thanh ghi dữ liệu và $oval_2$ 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
inoval1, oval2, inval;
oval1 = 0X81;
outportb (DATA, oval1);
oval2 = 0X08;
outportb (CONTROL, OVAL2^0x0b); //exclusive or
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
```

- Sử dụng ngắt:

Khi bit 4 của thanh ghi điều khiển ở mức 1 sẽ cho phép ngắt nếu chân ACK 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 1 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.

c. 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 500KB/sec đến 2 MB/sec 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 3.2.

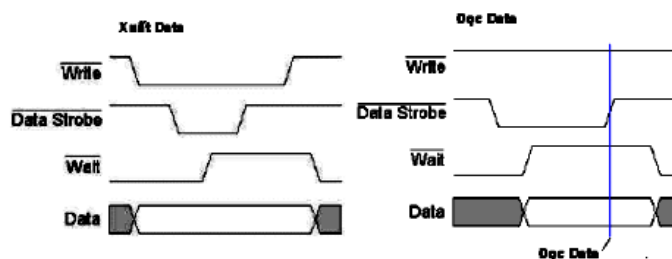
| Chân | Tên | Vào/Ra | Chức năng |
|------------|-----------------|----------|--|
| 1 | Ghi /Write | Ra | Khi thấp xuất dữ liệu ra Khi cao đọc vào |
| 2 - 9 | Data 0 - 7 | Vào - Ra | Tuyến dữ liệu hai hướng |
| 10 | Ngắt Interrupt | Vào | Ngắt ở cạnh lên |
| 11 | Chờ /Wait | Vào | Dùng để bắt tay, chu 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 3.3

Bảng 3.3.

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



Hình 3.5. Giải 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 trong thanh ghi gốc +4. Nếu /Wait mức thấp thì /Data Strobe sẽ ở mức thấp chờ khi /Write 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 10µs đường /Wait không tác động khi /Data Strobe hay /Address strobe tác động thì bit này sẽ được đặt lên 1.

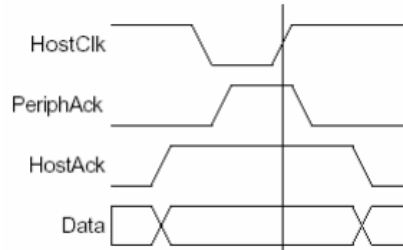
d. Cổng ECP.

Cổng ECP được phát triển bởi HP 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 qui định trong bảng 3.4.

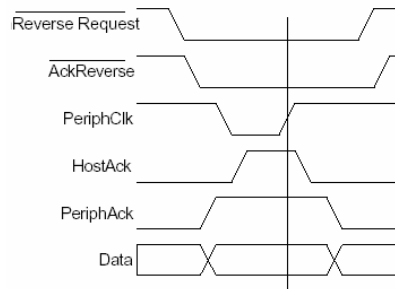
Bảng 3.4.

| 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 HostAck và PeriphAck ở mức cao là truyền dữ liệu. Nếu máy tính gửi lệnh, HostAck mức thấp; nếu ngoại vi gửi lệnh thì PeriphAck 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 kí tự 'A' thì gửi byte 24 (Run length Count) sau đó gửi byte 'A'.



Hình 3.6. Truyền dữ liệu từ máy tính ra ngoại vi.



Hình 3.7. Truyền dữ liệu từ ngoại vi vào máy tính

Ngoại vi nhận được byte 24 trong chu kì lệnh sẽ lặp lại byte 'A' ở chu kì 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.

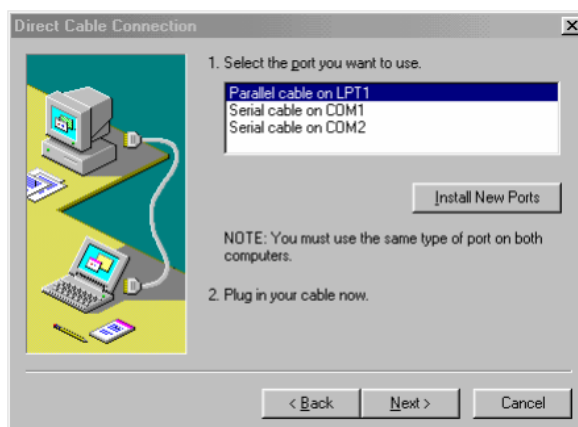
Bảng 3.5.

| 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.

e. Ghép nối hai máy tính.

Hai máy tính có thể ghép với nhau qua cổng song song hay 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 3.8. Giao diện ghép nối hai máy tính

1. Ghép nối song song (Laplink cable): Dùng cáp nối 2 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 2 đầu cái DB9 hay DB25).

Truyền bằng đường song song nhanh gấp 8 đến 10 lần truyền nối tiếp. Nếu cổng song song hai máy có cấu hình ECP thì vận tốc truyền còn nhanh hơn, dưới đây là cáp nối với cổng ECP.

| | 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 |

Bảng 3.6. Dùng cáp nối DB9 hay DB 25

| 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 | | | |

Bảng 3.7. Cáp nối cổng ECP

3.2. Giao tiếp qua cổng nối tiếp.

a. Cấu trúc cổng COM.

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 3 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.

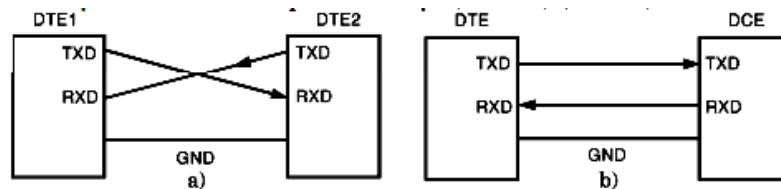
Các thiết bị ghép nối nối tiếp chia làm 2 đoạn DTE (Data Terminal Equipment) và DCE (Data Communication Equipment). DCE là các thiết bị trung gian như modem, còn DTE là các thiết bị như máy tính, vi điều khiển, PLC, là nguồn tạo ra dữ liệu hay tiếp nhận dữ liệu để xử lí. Có thể ghép nối DTE với DTE hoặc DCE, DCE với DTE hoặc DCE. Tín hiệu truyền nối tiếp theo dạng xung chuẩn RS232 của EIA (Electronics Industry Associations), mức logic 0 còn gọi là Space giữa +3V 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 hình 3.9 sau:

Hình 3.9



Khi không truyền đường dây sẽ ở trạng thái Mark, khi bắt đầu truyền, xung Start được truyền (+10V) 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 hình 3.10a.

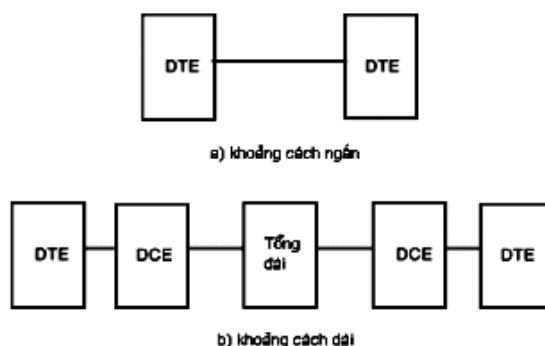


Hình 3.10

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 ground | 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 |

Thường sử dụng các sơ đồ kết nối hình 3.11.



Hình 3.11: a) Kết nối trực tiếp; b) Kết nối qua modem

Tín hiệu truyền nối tiếp dưới dạng các bit, số bit trong một giây 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 56000baud, 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 và 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 gửi trở về CTS và máy tính sẽ gửi dữ liệu, nếu máy tính không nhận được CTS sẽ không gửi dữ liệu. Điều khiển lưu lượng bằng phần mềm dùng hai kí tự Xon và Xoff, Khi modem muốn máy tính ngừng truyền sẽ gửi đi kí tự Xoff (ASCII 19), còn khi modem rảnh nó sẽ gửi kí tự Xon (ASCII 17).

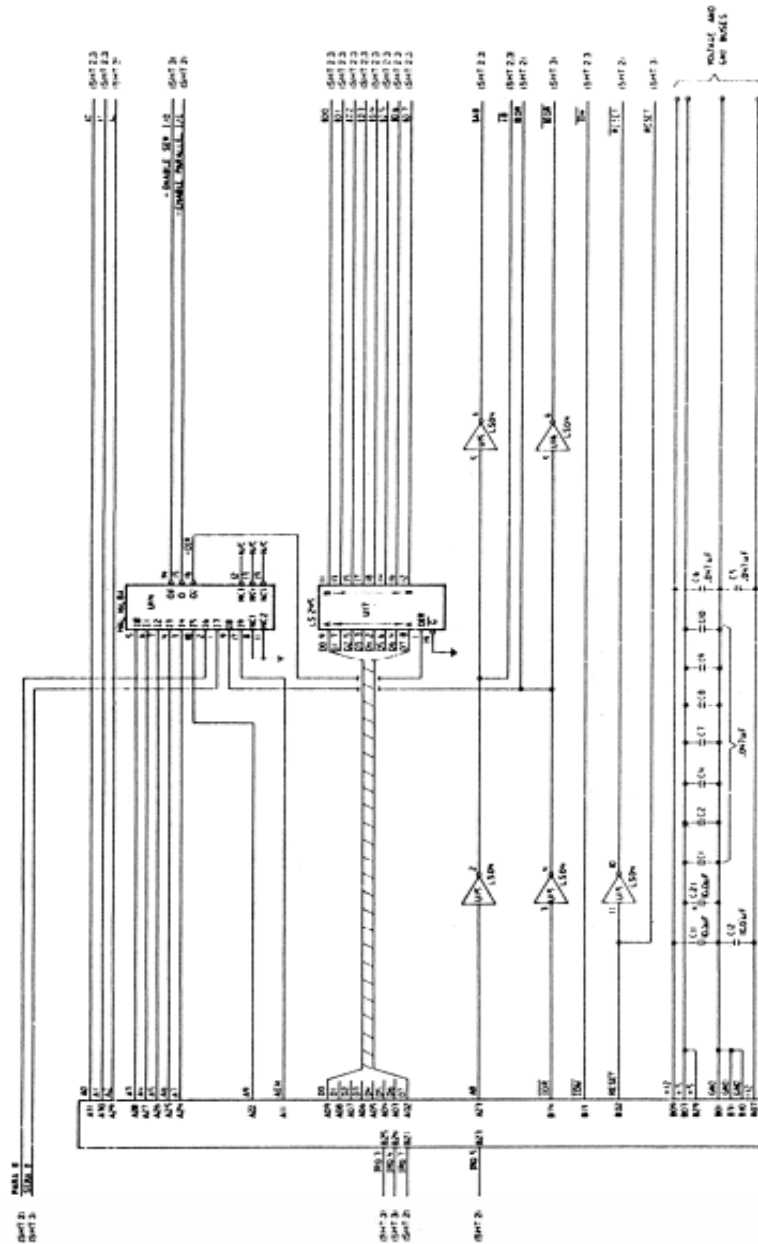
Việc trao đổi dữ liệu của máy tính được thực hiện thông qua vi mạch UART (Universal Asynchronous Receiver Transmitter), còn với vi điều khiển hay PLC thì có các vi mạch chuyên dụng hoặc được tích hợp trong vi xử lí. Các máy tính đời mới dùng công nghệ ASIC sử dụng chip đa năng làm nhiệm vụ giao tiếp 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.

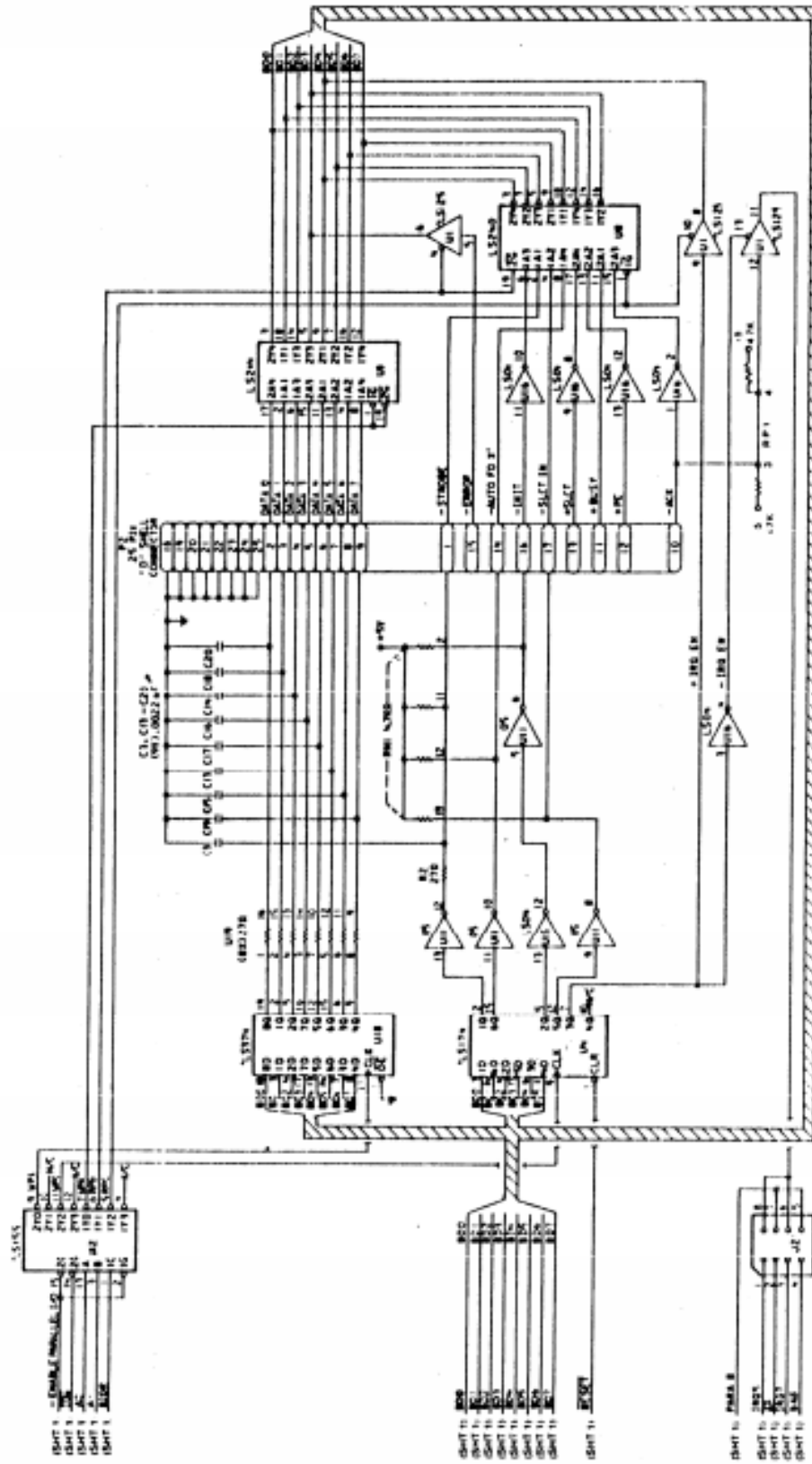
Bảng dưới đây cho địa chỉ gốc cổng COM và các thông tin khác đặc trưng cho từng cổng COM.

| 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 |

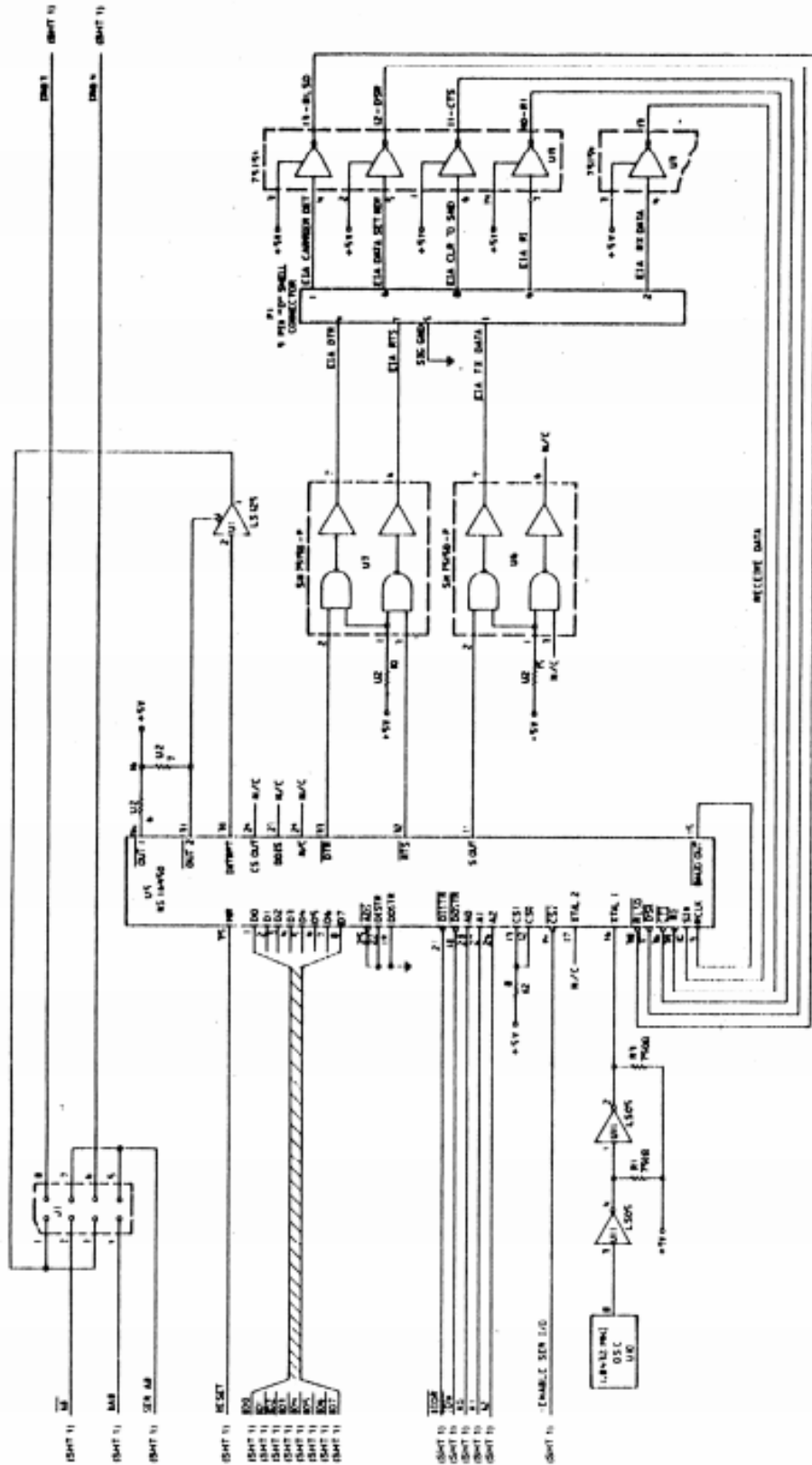
Hình 3.12 cung cấp sơ đồ card giao tiếp LPT/COM của máy XT giúp ta có khái niệm về cách ghép bus ISA với UART. Các chân UART có mức TTL nên cần mạch chuyển mức để ghép với mức RS232. Hình 3.13 cho sơ đồ chân của một số UART thông dụng. Bảng 3.8 cung cấp ý nghĩa các chân của 16550.



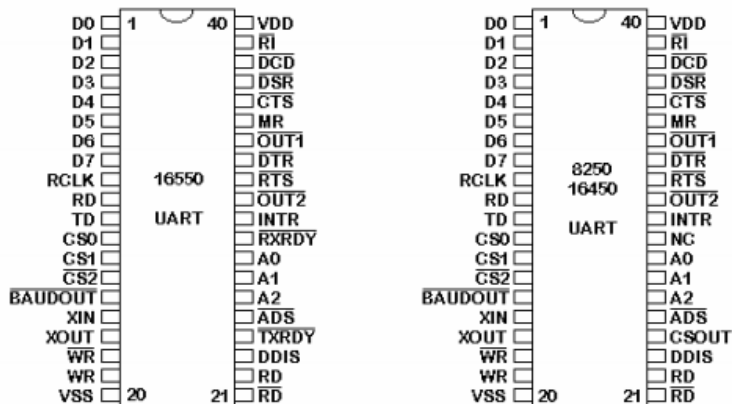
Hình 3.12a. Card giao tiếp LPT/COM phân đệm



Hình 3.12b: Card giao tiếp LPT/Com phần LPT



Hình 3.12c: Card giao tiếp LPT/COM phần COM



Hình 3.13. Sơ đồ chân UART

| Chân | Tên | Ý nghĩa |
|--------|--------------|--|
| 1:8 | D0:D7 | Data Bus |
| 9 | RCLK | Xung nhịp vào, tần số = Baud Rate * 16 |
| 10 | RD | RXD |
| 11 | TD | TXD |
| 12..14 | CS0,CS1,/CS2 | Ba chân chọn chip |
| 15 | /BAUDOUT | Xung nhịp ra, tần số = Baud Rate * 16 |
| 16 | XIN | Dao động vào bộ chia tần |
| 17 | XOUT | Dao động ra |
| 18 | /WR | Ghi dữ liệu, đảo |
| 19 | WR | Ghi dữ liệu, không đảo |
| 20 | VSS | Mass |
| 21 | /RD | Đọc dữ liệu, đảo |
| 22 | RD | Đọc dữ liệu, không đảo |
| 23 | DDIS | Driver Disable, cấm driver |
| 24 | /TXRDY | Transmitter Ready, sẵn sàng phát |
| 25 | /ADS | Address Strobe. Cài địa chỉ |
| 26..28 | A2, A1, A0 | Chọn thanh ghi |
| 29 | /RXRDY | Receive Ready |
| 30 | INTR | Interrupt Output, yêu cầu ngắt |
| 31 | nOUT2 | User Output 2 |
| 32 | /RTS | Request to Send |
| 33 | /DTR | Data Terminal Ready |
| 34 | /OUT1 | User Output 1 |
| 35 | MR | Master Reset |
| 36 | /CTS | Clear To Send |
| 37 | /DSR | Data Set Ready |
| 38 | nDCD | Data Carrier Detect |
| 39 | nRI | Ring Indicator |
| 40 | VDD | + 5 Volts |

Bảng 3.8. Ý nghĩa các chân của UART 16550

| Địa chỉ gốc | DLAB | Đọc/Ghi | Tên | Chức năng |
|-------------|------|---------|------|--|
| + 0 | 0 | Ghi | THR | Transmitter Holding Register Thanh ghi giữ thông tin truyền |
| | 0 | Đọc | RBR | Receiver Buffer Register Thanh ghi đệm thu |
| | 1 | Đọc/Ghi | BRDL | Baud Rate Divisor Latch Cài số chia byte thấp |
| + 1 | 0 | Đọc/Ghi | IER | Interrupt Enable Register Thanh ghi cho phép ngắt |
| | 1 | Đọc/Ghi | BRDH | Cài số chia byte cao |
| + 2 | | Đọc | IIR | Interrupt Identification Register Thanh ghi nhận dạng ngắt |
| | | Ghi | FCR | FIFO Control Register |
| + 3 | | Đọc/Ghi | LCR | Line Control Register Thanh ghi điều khiển đường dây |
| + 4 | | Đọc/Ghi | MCR | Modem Control Register Thanh ghi điều khiển modem |
| + 5 | | Đọc | LSR | Line Status Register Thanh ghi trạng thái đường dây |
| + 6 | | Đọc | MSR | Modem Status Register Thanh ghi trạng thái Modem |
| + 7 | | Đọc/Ghi | | Scratch Register |

Bảng 3.9. Các thanh ghi của UART

| Baud Rate | BRDH | BRDL |
|-----------|------|------|
| 4800 | 00 | 18 |
| 9600 | 00 | 0C |
| 19200 | 00 | 06 |
| 38400 | 00 | 03 |
| 57600 | 00 | 02 |
| 115200 | 00 | 01 |

Bảng 3.10

| | |
|----|---|
| D5 | Cho phép kiểu công suất thấp (16750) |
| D4 | Cho phép kiểu ngủ (16750) |
| D3 | Cho phép ngắt khi có lỗi modem |
| D2 | Cho phép ngắt khi có lỗi truyền, thu |
| D1 | Cho phép ngắt khi THR trống (đã truyền dữ liệu) |
| D0 | Cho phép ngắt khi RBR đầy (đã nhận dữ liệu) |

Bảng 3.11. Thanh ghi cho phép ngắt



| | | | |
|-----------|-------------------------------|-------|--|
| Bit 6 - 7 | D6 | D7 | |
| | 0 | 0 | Không có FIFO |
| | 0 | 1 | Cho phép FIFO nhưng không dùng |
| | 1 | 1 | Cho phép FIFO |
| Bit 5 | Cho phép FIFO 64 byte (16750) | | |
| Bit 4 | Không dùng | | |
| Bit 3 | 0 | | Không dùng |
| | 1 | | Ngắt hết thời gian (16550) |
| Bit 1 - 2 | Bit 2 | Bit 1 | |
| | 0 | 0 | Ngắt trạng thái modem, xóa khi đọc MSR |
| | 0 | 1 | Ngắt THR trống, xóa khi ghi vào THR |
| | 1 | 0 | Ngắt RBR đầy, xóa khi đọc RBR |
| | 1 | 1 | Lỗi truyền thu |
| Bit 0 | 0 | | Có ngắt |
| | 1 | | Không có ngắt |

Bảng 3.12. Thanh ghi nhận dạng ngắt IIR

DLAB (Divisor Latch Access Bit) là bit 7 của LCR, khi DLAB = 1 cho phép đặt bộ chia tần trong UART để được vận tốc baud mong muốn.

UART dùng tinh thể dao động 1.8432 MHz chia cho 16 được tần số 115200Hz. Tần số này được chia bởi bộ đếm lập trình được 16 bit, số chia chứa trong hai thanh ghi địa chỉ gốc +1 (byte cao) và gốc +0 (byte thấp) được sử dụng khi DLAB = 1.

Nếu muốn vận tốc truyền 9600 thì số chia là 12 hay 000CH, byte cao là 00H, byte thấp là 0CH.

Khi DLAB = 0 ghi vào địa chỉ gốc +0 để truyền đi 8 bit nối tiếp, đọc địa chỉ gốc +0 nhận được 8 bit dữ liệu truyền đến.

Thanh ghi IER (gốc +1) cho phép ngắt vì xử lý khi có biến cố trên đường truyền.

Khi có ngắt xảy ra, bit 0 của IIR ở mức 0, loại ngắt chỉ bởi bit 1 và bit 2, ngắt lỗi truyền thu có ưu tiên cao nhất, còn ngắt do modem ưu tiên thấp nhất.

| | | | | |
|-----------|---|-------|---------------|------------------------------|
| Bit 7 | 1 : cài đặt số chia DLAB | | | |
| | 0 : truy xuất RBR, THR, IER | | | |
| Bit 6 | Khi bằng 1 cho phép truyền tín hiệu BREAK, đường dây ở mức 0 (+12V) trong thời gian lớn hơn một khung | | | |
| Bit 3 + 5 | Bit 5 | Bit 4 | Bit 3 | |
| | X | X | 0 | Không kiểm tra cờ chắn/lỗi |
| | 0 | 0 | 1 | Khi lẻ |
| | 0 | 1 | 1 | Khi chẵn |
| | 1 | 0 | 1 | Bit kiểm tra cờ chắn lẻ là 1 |
| | 1 | 1 | 1 | Bit kiểm tra cờ chắn lẻ là 0 |
| Bit 2 | 0 : Một bit stop | | | |
| | 1 : Hai bit stop khi số bit dữ liệu là 6, 7, 8. Khi số bit dữ liệu là 5 thì dùng 1,5 bit stop | | | |
| Bit 0, 1 | Bit 1 | Bit 0 | | |
| | 0 | 0 | 5 bit dữ liệu | |
| | 0 | 1 | 6 bit dữ liệu | |
| | 1 | 0 | 7 bit dữ liệu | |
| | 1 | 1 | 8 bit dữ liệu | |

Bảng 3.14. Thanh ghi điều khiển đường truyền LCR xác định khung truyền

| | |
|-------|---|
| Bit 5 | Dùng cho 16750 |
| Bit 4 | Mode loopback dùng để kiểm tra hoạt động UART |
| Bit 3 | Điều khiển mở ra Aux Output 2 |
| Bit 2 | Điều khiển mở ra Aux Output 1 |
| Bit 1 | Cho RTS lên 1 |
| Bit 0 | Cho DTR lên 1 |

Bảng 3.15. Thanh ghi điều khiển modem MCR

| | |
|-------|--|
| Bit 7 | Sai trong FIFO |
| Bit 6 | Thanh ghi giữ truyền THR và thanh ghi chờ trống |
| Bit 5 | THR trống |
| Bit 4 | Ngắt Break, đường truyền ở logic 0 thời gian dài hơn truyền một byte |
| Bit 3 | Sai khung truyền, bit cuối không phải là bit stop |
| Bit 2 | Sai parity |
| Bit 1 | Khi dữ liệu tới không đọc kịp làm ghi chống lên RBR |
| Bit 0 | Báo có dữ liệu ở RBR |

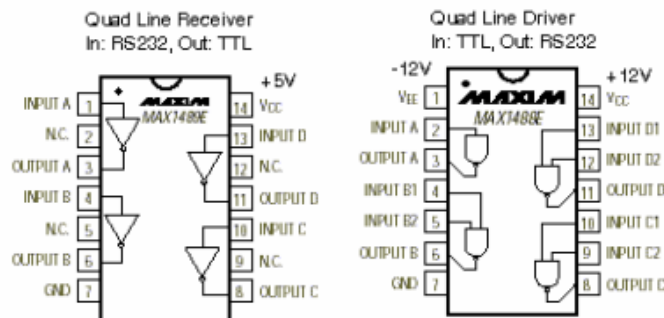
Bảng 3.16. Thanh ghi trạng thái đường dây LSR

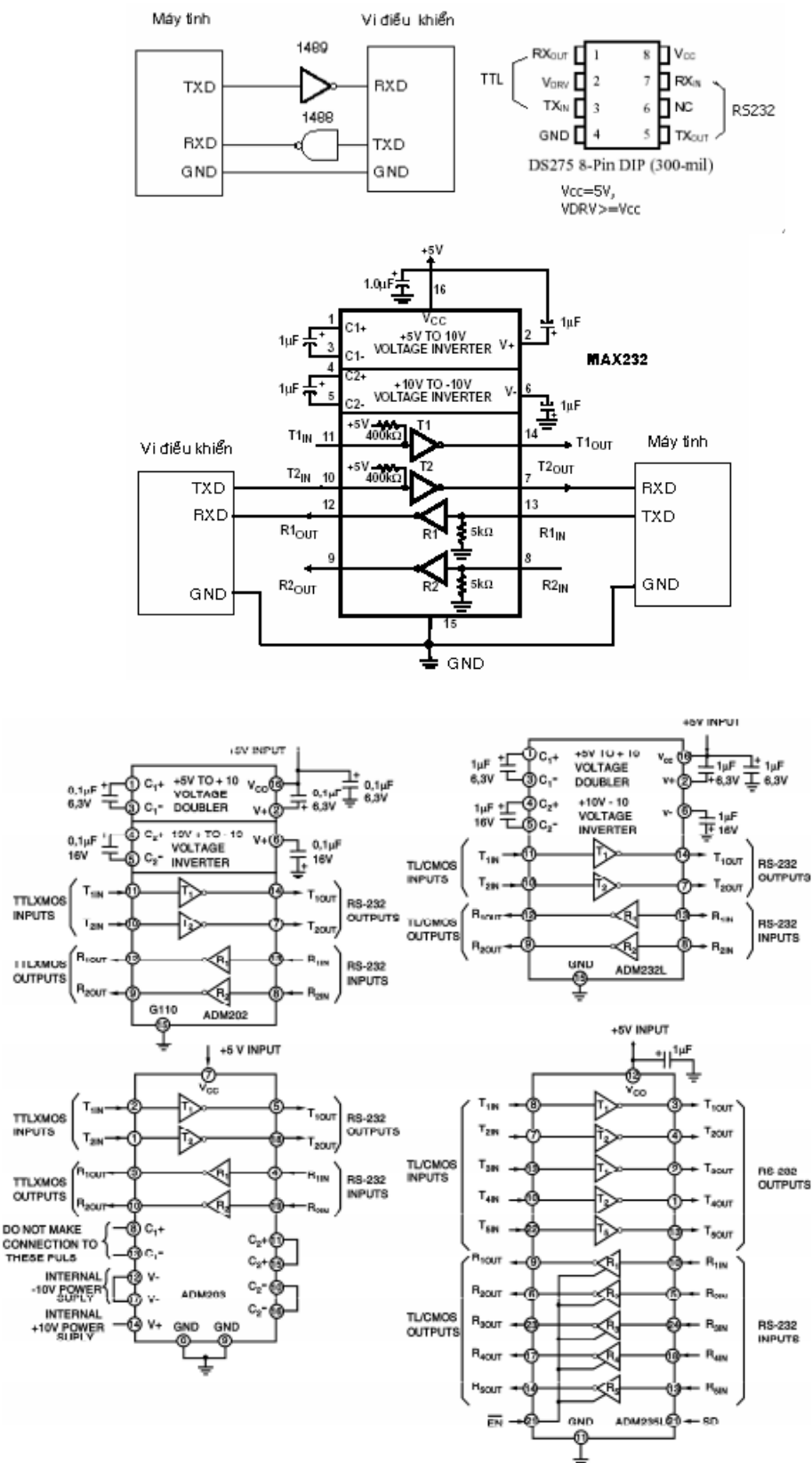
| | |
|-------|----------------------------------|
| Bit 7 | Logic của CD, báo có sóng mang |
| Bit 6 | Logic của RI, báo có chuông |
| Bit 5 | Logic của DSR |
| Bit 4 | Logic của CTS |
| Bit 3 | Mức 1 khi có CD đổi trạng thái |
| Bit 2 | Mức 1 khi RI đổi từ thấp lên cao |
| Bit 1 | Mức 1 khi DSR thay đổi |
| Bit 0 | Mức 1 khi CTS thay đổi |

Bảng 3.17. Thanh ghi trạng thái modem MSR

b. Mạch chuyển mức.

Khi ghép cổng COM máy tính với vi điều khiển hay mạch TTL cần phải có mạch chuyển mức TTL → 232 và ngược lại, các vi mạch thường dùng là cặp MC1488-MC1489, MAX232 (Maxim) hoặc DS 275 (Dallas) (Hình 3.14), SN75150-SN75154 (Hình 3.12)... Loại MAX232 thông dụng hơn cả vì chỉ cần nguồn +5V, nguồn ±10V do mạch dao động 16Khz bên trong cung cấp.





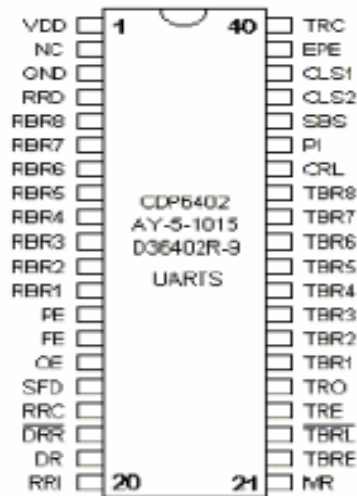
Hình 3.14. Một số vi mạch chuyển đổi TTL → RS232 thông dụng

c. Mạch giao tiếp công nối tiếp.

Sử dụng vi mạch CDP6402 ta có thể chuyển đổi số liệu song song ở ngoại vi ra tín hiệu nối tiếp và ngược lại để ghép nối với công nối tiếp. Vi mạch này chuyển đổi dữ liệu song song ra nối tiếp và ngược lại theo chuẩn RS232. Có hai tuyến dữ liệu song song riêng cho phân thu và phát.

Tín hiệu TXD từ cổng COM được đổi sang mức TTL nhờ vi mạch MAX232 đưa vào chân RRI (Receiver Register In) của 6402 và đổi thành tín hiệu song song 8 bit RBR1÷RBR8 (Receiver Register). Tín hiệu song song 8 bit TBR1÷TBR8 (Transmitter Buffer Register) được đổi thành tín hiệu nối tiếp ra chân TRO (Transmitter Register Out) sau đó nhờ MAX 232 đổi sang điện áp thích hợp vào chân RXD. Vận tốc truyền được xác định bởi tần số tín hiệu ở chân RRC (Receiver Register Clock), TRC (Transmitter Register Clock). Nhờ vi mạch dao động chia tần 74HC4060 có thể thay đổi các vận tốc truyền khác nhau.

Khuôn dạng truyền được xác định bởi các chân PI (Parity Inhibit) SBS (Stop Bit Select) CLS1,2 (Character Length Select) và ÊP (Even Parity Select), các tín hiệu vào được cài bởi CRL (Control Register Load).



Hình 3.15. UART CD6402

| | |
|------------|--------------------------|
| PI: | 1: không kiểm tra parity |
| | 0: có kiểm tra parity |
| SBS: | 1: hai bit stop |
| | 0: một bit stop |
| CLS2, CLS1 | 00: 5 bit data |
| | 01: 6 bit |
| | 10: 7 bit |
| | 11: 8 bit |
| EPE | 1: parity chẵn |
| | 0: parity lẻ |

Bảng 3.18

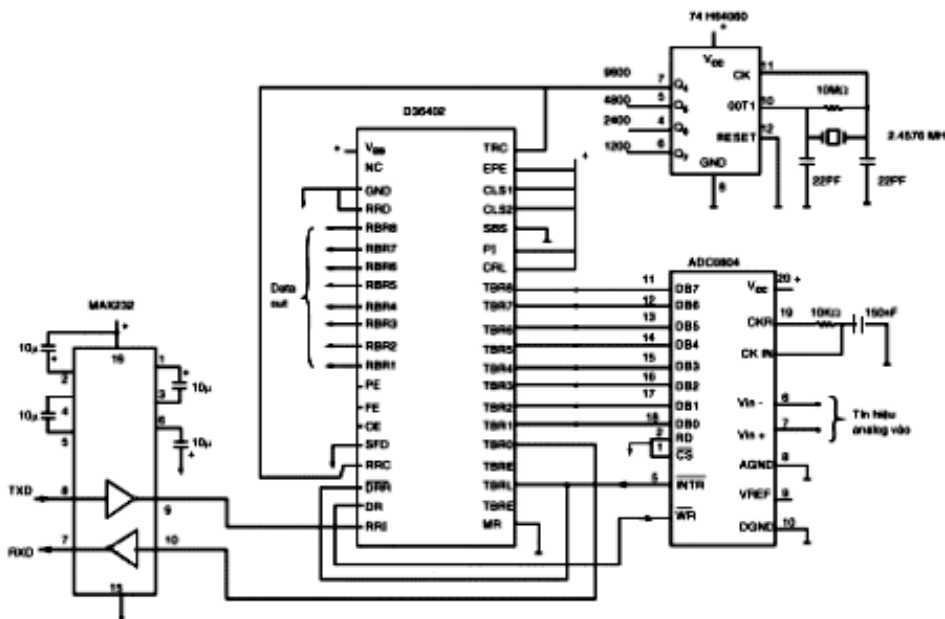
Khi một byte được truyền tới vi mạch 6402 từ TXD, chân DR (Data Receiver) sẽ chuyển mức 1, byte truyền tới được xuất ra song song ở RBR1 ÷ RBR8, muốn xóa DR ta cho DRR đảo (Data Receiver Reset) ở mức 0. Chân TBRL đảo (Transmitter Buffer Register Load) ở mức 0 sẽ nạp data song song ở TBR1 ÷ TBR8 vào thanh ghi đệm truyền, khi chân này chuyển sang mức cao sẽ truyền dữ liệu đi nối tiếp ở TRO.

| | | | |
|------|-----------------------------------|---|---|
| PE | Cờ lỗi | 1 | Có lỗi |
| OE | Overrun error | 1 | Đã nhận dữ liệu nhưng DRR chưa tác động |
| SFD | Status flag disable | 1 | Cấm các cờ báo |
| TRE | Transmitter Register empty | 1 | Thanh ghi truyền trống |
| DR | Data received | 1 | Đã nhận dữ liệu |
| TBRE | Transmitter buffer Register empty | 1 | Thanh ghi đệm truyền trống |

Bảng 3.19. Trạng thái bộ truyền thu nối tiếp thông báo ra ngoài qua các chân



Trong sơ đồ hình 3.16, một vi mạch ADC0804 1 kênh 8 bit được dùng để đối áp tương đồng ra số 8 bit đưa vào TBR1 ÷ TBR8, khi 6402 nhận 1 byte từ TXD nó sẽ cho DR ở mức cao đưa vào chân WR đảo của ADC0804 bắt đầu chuyển đổi AD. Đối xong INTR đảo tác động đưa vào TBRL đảo, nạp đủ 8 bit đã đổi vào thanh ghi đệm truyền và truyền đi nối tiếp, đồng thời đưa vào DRR đảo làm xóa DR ngưng đổi cho đến khi có một byte mới vào RRI.

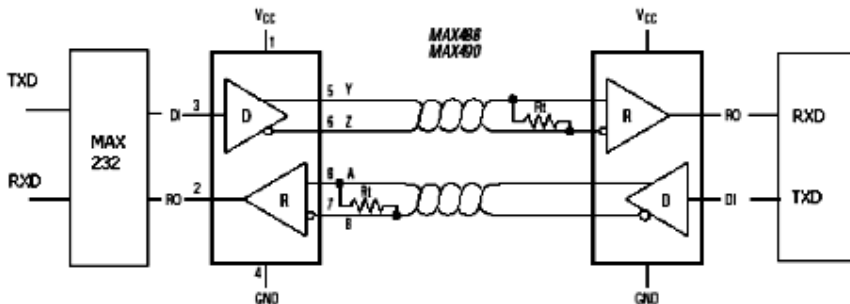


Hình 3.16. Chuyển đổi AD dùng cổng nối tiếp 9600, 8, E, I.

d. Mạng 485.

Chuẩn RS dùng đường truyền không cân bằng vì các tín hiệu đều lấy điểm chuẩn là đường mass chung, bị ảnh hưởng của nhiễu tác động, do đó tốc độ truyền và khoảng cách truyền bị giới hạn.

Khi cần tăng khoảng cách và tốc độ truyền, phương pháp truyền hai dây vi sai trở nên hữu hiệu hơn vì hai dây có đặc tính giống nhau, tín hiệu truyền đi là tín hiệu số điện áp giữa hai dây, do đó loại trừ được nhiễu chung. Hai chuẩn thường dùng là RS422 và RS485. Tuy nhiên chuẩn RS422 ít thông dụng hơn RS485.

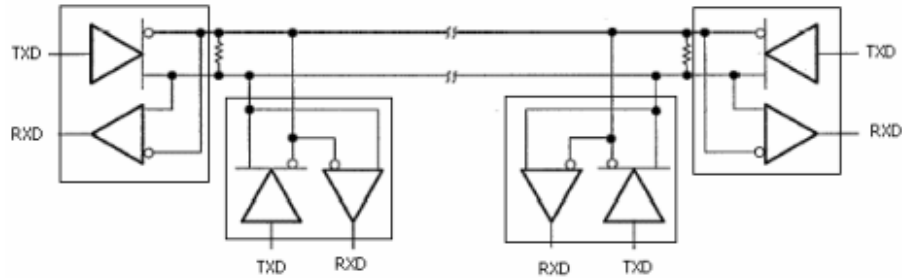


Hình 3.17. Truyền tín hiệu song song dùng RS422

Chuẩn RS422 dùng 4 dây do đó cho phép truyền song công, tức là cùng lúc có thể thu phát. Tín hiệu từ máy tính theo chuẩn RS232 có mức điện áp ±12V phải đổi sang mức TTL0 ÷ 5V dùng vi mạch MAX232, sau đó tín hiệu đơn được đổi sang tín hiệu vi sai dùng vi mạch MAX485 (488, 489, 490, 491 ...). Điện áp vi sai phải lớn hơn 200mV. Nếu $V_{AB} > 200mV$ ta coi như trị logic 1 được truyền, còn nếu $V_{AB} < -200mV$ thì trị logic 0 được truyền.

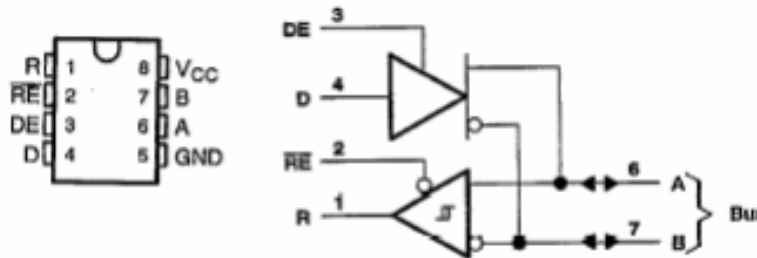
Chuẩn RS422 không cho phép nhiều hơn hai thiết bị truyền nhận tin trên đường dây, vì vậy chuẩn RS485 thông dụng hơn.

Với chuẩn RS485 ta có thể nối 32 thiết bị thu phát trên hai dây, có khoảng cách tối đa 1200m và vận tốc truyền đến 10MBit/sec. Hai điện trở kết thúc 120Ω được nối với hai đầu xa nhất của mạng, dây dẫn là loại xoắn đôi 26AWG. Chuẩn này dùng vi mạch lái SN75176 hay họ Maxim MAX481, 483, 485, 487, 488, 489, 490, 1487. Tiêu biểu là vi mạch MAX485 chuyển đổi từ tín hiệu đơn sang tín hiệu vi sai, có chân điều khiển cho đầu ra vi mạch ở tổng trở cao, nhờ vậy có thể nối chung nhiều vi mạch lái với nhau.



Hình 3.18. Mạng RS485

MAX485 gồm bộ lái và bộ thu, tín hiệu vào bộ lái D logic TTL đổi thành hai tín hiệu A và B, khi tín hiệu điều khiển DE mức thấp thì hai chân AB cách li với mạch. Tín hiệu vào bộ thu là A và B, tín hiệu ra R logic TTL tùy thuộc hiệu điện áp giữa A và B, khi /RE logic 1 thì R cách li với vi mạch.



MAX485, SN75176

Mạch lái

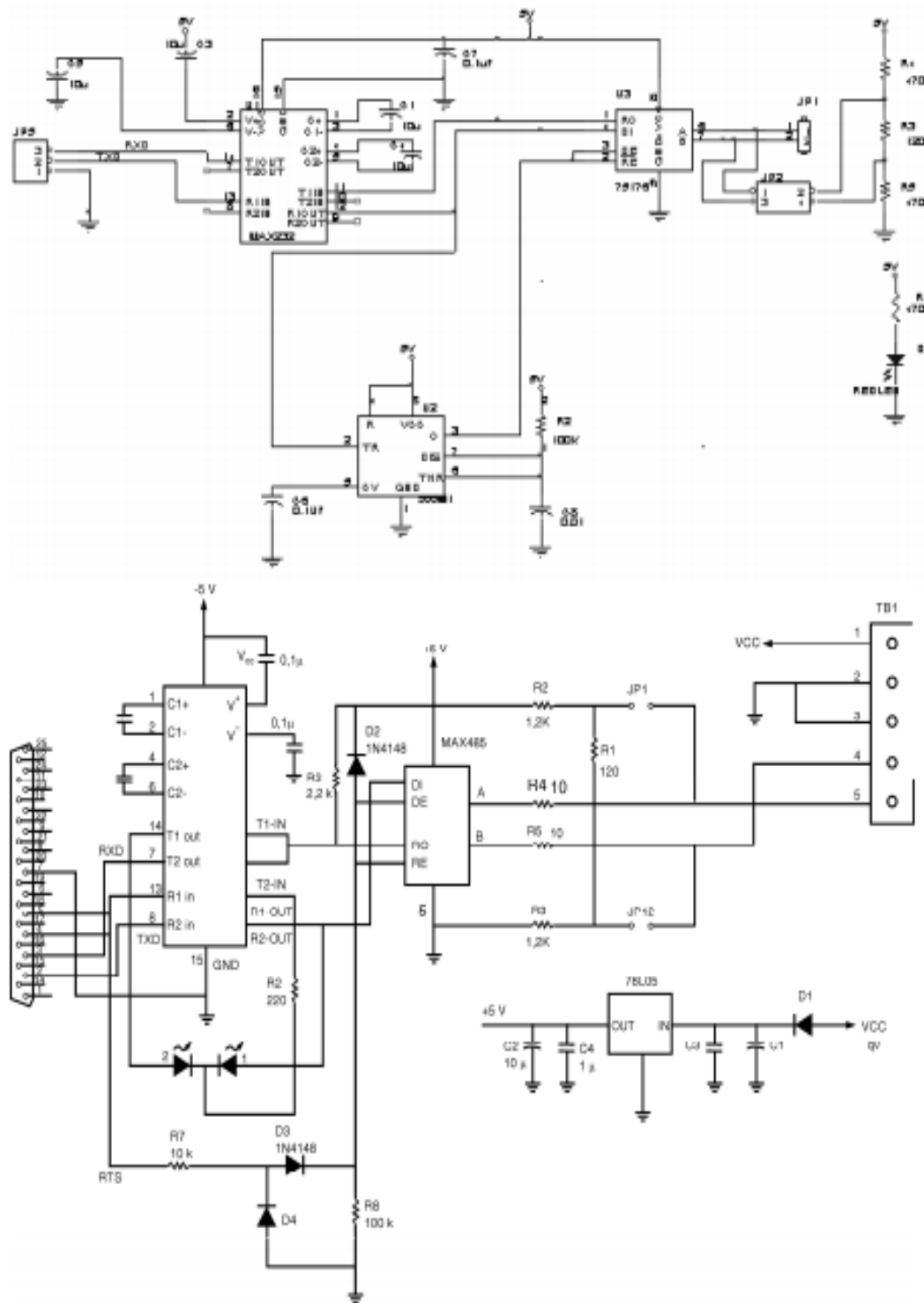
| Input D | Control DE | OUTPUT | |
|---------|------------|--------|---|
| | | A | B |
| H | H | H | L |
| L | H | L | H |
| X | L | Z | Z |

Mạch thu

| VID=VA-VB | Control /RE | Output R |
|----------------------------|-------------|----------|
| $VID \geq 0.2V$ | L | H |
| $-0.2V \leq VID \leq 0.2V$ | L | ? |
| $VID \leq -0.2V$ | L | L |
| X | H | Z |
| Ngõ vào hở | L | H |

Hình 3.19. Vi mạch MAX485 và bảng thực trị

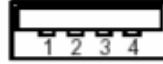
Mạng 485 làm việc theo chế độ Master – Slave, master cho DE mức 1 để truyền dữ liệu, còn các slave có DE=0, /RE=0 chờ nhận dữ liệu. Khi master muốn nhận dữ liệu, DE=0, /RE=0 còn slave phát sẽ có DE=1, /RE=1. Điều khiển các đường DE, /RE bằng tín hiệu RTS hay mạch định thời.



Hình 3.20. Mạch chuyển đổi RS232 →485

3.3. Cổng USB.

Ngày nay các máy tính đều có trang bị ít nhất hai cổng USB (Universal Serial Bus) để kết nối với máy in, camera, chuột, thanh nhớ Flash Rom, modem ... Đặc điểm của USB là vận tốc truyền lớn, Plug and Play, có thể gắn và tháo nóng, không cần nguồn cung cấp cho thiết bị và có thể kết nối nhiều thiết bị trên một Bus chung. USB là sản phẩm chung của nhiều công ty như Intel, Compaq, HP, Lucent, Microsoft, NEC, Philips. Có hai chuẩn USB1.1 (năm 1998) và USB2.0 (năm 2000) nhanh hơn. Ba vận tốc truyền là 480Mb/s, 12Mb/s và 1.5Mb/s. Cổng USB có 4 chân gồm hai dây nguồn và hai dây tín hiệu vì sai.



| Pin Number | Cable Colour | Function |
|------------|--------------|---------------------|
| 1 | Red | V_{BUS} (5 volts) |
| 2 | White | D- |
| 3 | Green | D+ |
| 4 | Black | Ground |

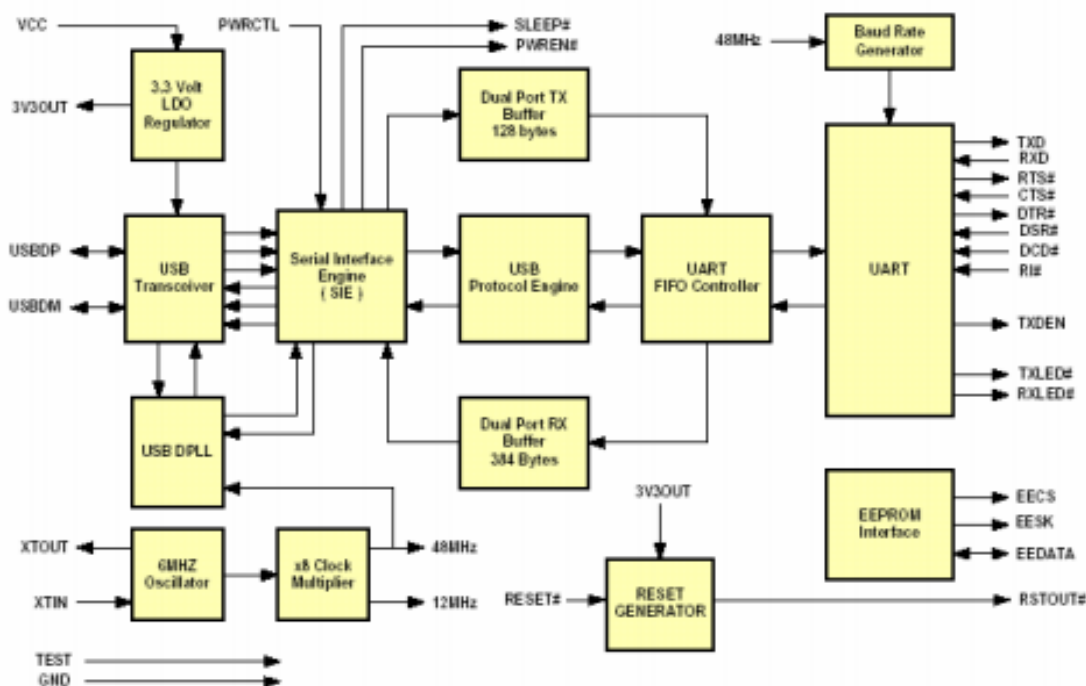
Hình 3.21. Sơ đồ chân cổng USB phía máy tính

Khi cắm thiết bị vào cổng USB điện áp trên dây 2 và 3 thay đổi báo cho bộ điều khiển USB có thiết bị gắn vào và bắt đầu một loạt trao đổi thông tin nhận dạng thiết bị gắn vào để nạp Driver phù hợp cho thiết bị. Muốn gắn nhiều thiết bị vào một cổng ta dùng Hub. Số lượng thiết bị USB tối đa là 127.

Nhiều máy tính không còn thiết kế cổng COM, do đó gây bất tiện khi cần giao tiếp nối tiếp với các thiết bị không hỗ trợ USB. Nhiều hãng đã chế tạo các mạch chuyển đổi từ USB sang RS232 hay RS422, RS485. Nguyên tắc là dùng một vi mạch làm giao tiếp với máy tính theo chuẩn USB và giao tiếp với các thiết bị khác theo chuẩn của cổng COM. Phần mềm driver sẽ coi thiết bị như là cổng COM bình thường và ta lập trình giao tiếp với thiết bị ngoại vi như là với cổng COM, qua trung gian mạch chuyển đổi.

Ví dụ, xét sản phẩm của hãng FTDI (Future Technology Devices International Ltd.) dùng vi mạch FT232BM, sơ đồ khối vi mạch trình bày ở hình 3.22. Phân sau mô tả các khối chính:

- *3.3V LDO Regulator: tạo nguồn 3.3V cho các khối khác.*
- *USB Transceiver: Lái tuyến dữ liệu.*
- *USB DPLL: Vòng khóa pha.*
- *Serial Interface Engine: Chuyển đổi song song nối tiếp, nén tín hiệu và kiểm tra chống sai.*
- *USB Protocol Engine: Tạo và kiểm tra giao thức USB.*
- *Dual Port TX Buffer: Chứa dữ liệu truyền.*
- *Dual Port RX Buffer: Chứa dữ liệu thu.*
- *UART FIFO Controller: Điều khiển truyền dữ liệu giữa Buffer và thanh ghi UART.*
- *UART: Truyền và thu dữ liệu theo chuẩn RS232.*
- *EEPROM Interface: Chứa thông số nhận dạng, nếu không có linh kiện này thì dùng thông số do nhà sản xuất cài sẵn trong chip.*

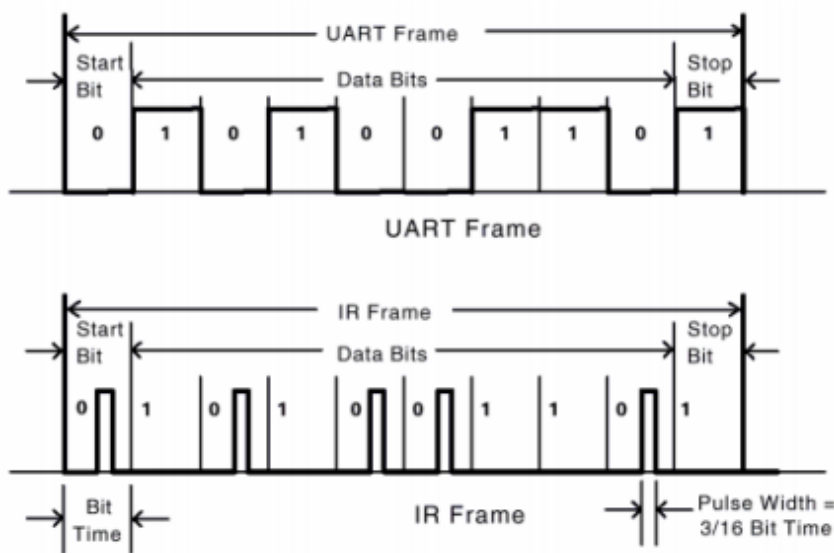


Hình 3.23. Sơ đồ khối vi mạch FT232 chuyển đổi USB → COM

3.4. Cổng hồng ngoại.

Cổng hồng ngoại, (IrDA: InfraRed Data Association) thường được trang bị trên máy tính xách tay để kết nối với thiết bị số như máy tính, điện thoại di động, camera số ... sử dụng sóng tần số 875nm, khoảng cách liên lạc chừng 1m.

Chuẩn IrDA 1.0 có vận tốc truyền 2400 đến 115.200kb/s, tương tự như chuẩn UART (hình 3.24). Chuẩn IR 1.1 có ba vận tốc truyền 0.576, 1.152 và 4Mb/s truyền tin theo gói gồm hai byte start, địa chỉ, dữ liệu, CRC và bit Stop.



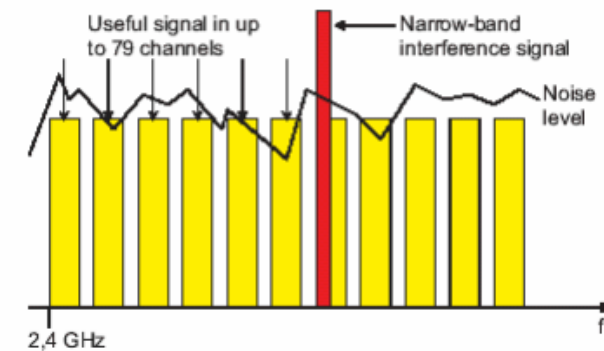
Hình 3.24. Chuẩn IR 1.0

3.5. Mạng.

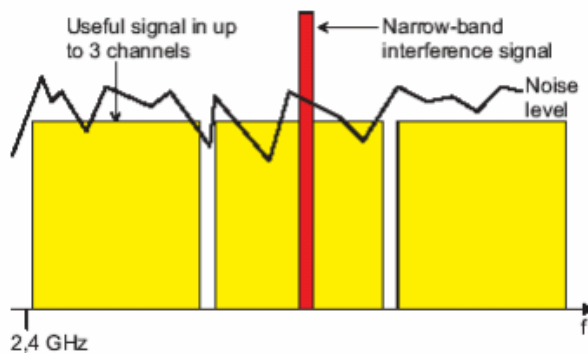
Hệ thống sản xuất lớn bao gồm các máy tính, PLC, bộ điều khiển quá trình .. tất cả kết nối qua mạng hai dây và/hay vô tuyến. Trong công nghiệp có nhiều loại mạng khác nhau như Profibus (Process Field Bus), CAN (Controler Area Network), DeviceNet, Modbus, ASI, Ethernet Công nghiệp, DH485 ... Các công ty lớn về tự động hóa như Siemens, OMRON, Allen-Bradley, Schneidei sản xuất rất nhiều thiết bị mạng và các mạng con của họ cũng rất đa dạng, tuy nhiên phần lớn đều theo chuẩn mạng mở, tức là có thể ghép các thiết bị mạng của nhiều hãng chung với nhau, tất nhiên là phải theo một chuẩn nào đó. Máy tính nối với mạng thông qua card mạng.

Những năm gần đây phổ biến mạng không dây sử dụng dải tần 2.4GHz và 5.7GHz, đó là các mạng LAN không dây, Blue Tooth, WiFi, GPRS (General Packet Radio Service) và WAP (Wireless Application Protocol). Do tính chất truyền tin đa đường, nhiều kênh truyền cao, công suất phát không lớn và yêu cầu bảo mật nên nhiều kỹ thuật hiện đại được sử dụng, ví dụ như kỹ thuật trải phổ. Có hai kỹ thuật trải phổ sử dụng, FHSS trải phổ nhảy tần (Frequency Hopping Spread Spectrum) tần số sóng mang thay đổi ngẫu nhiên trong 79 tần số; kỹ thuật khác là trải phổ chuỗi trực tiếp DSSS (Direct Sequence Spread Spectrum) một bit thông tin được mã hóa thành một chuỗi bit ngẫu nhiên. Hai phương pháp này giúp trải rộng dải tần tín hiệu, do đó làm giảm ảnh hưởng của nhiễu dải tần hẹp và khó xem trộm thông tin. Các thiết bị không dây ở gần nhau tạo thành một tế bào, sự truyền tin sang tế bào khác thực hiện nhờ các bộ lặp lại vô tuyến, còn gọi là điểm truy cập.

Kỹ thuật truy cập mạng là CSMA/CA tránh xung đột. Khi một trạm muốn truyền nhận thấy môi trường tự do, nó sẽ gửi RTS cho biết thời gian truyền, đối tác gửi trả lại CTS và sự truyền tin bắt đầu, các trạm khác biết khi nào kết thúc sự truyền và sẽ chờ đợi. Khi kết thúc truyền, đối tác gửi ACK báo truyền tin thành công.



Kỹ thuật FHSS



Kỹ thuật DSSS

Hình 3.25. Kỹ thuật trải phổ

Chương 4

BỘ ĐIỀU KHIỂN LẬP TRÌNH ĐƯỢC

4.1. Giới thiệu tổng quan.

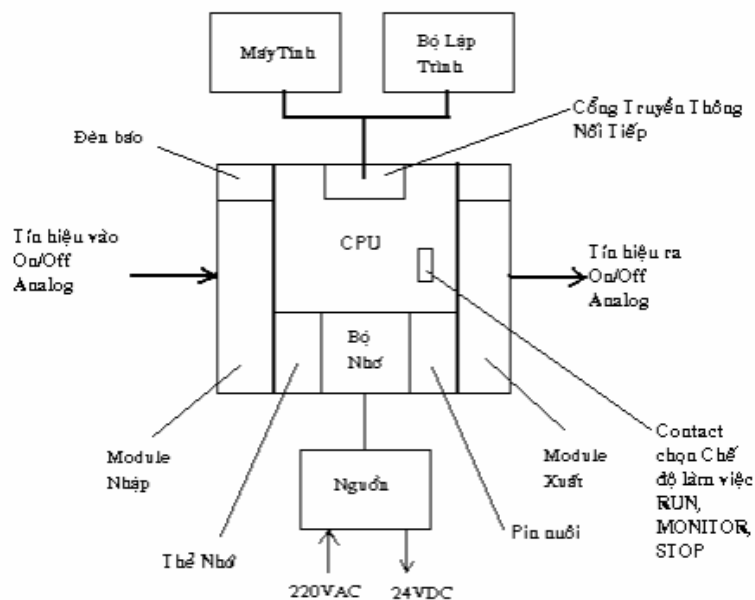
Trong hệ thống tự động thường gặp những thiết bị làm việc theo kiểu tuần tự, theo qui luật if ... then ... else với tín hiệu vào và ra có hai mức, ví dụ như contact hành trình, role. Các sơ đồ này có thể thực hiện bằng role và mạch định thời nhưng với sơ đồ phức tạp số lượng role khá lớn, độ tin cậy kém và nhiều khi không đạt yêu cầu. Từ những năm 70 để đáp ứng yêu cầu có những thiết bị điều khiển thay thế sơ đồ role, đã xuất hiện bộ điều khiển logic lập trình được (programmable Logic Controller-PLC) và ngày càng hoàn thiện, được áp dụng rộng rãi trong công nghiệp (PLC của hãng Allen Bradley Corporation sản xuất năm 1977 sử dụng vi xử lý 8080).

Các PLC đầu tiên chỉ thực hiện được các phép tính logic, tín hiệu vào và ra là tín hiệu rời rạc, còn hiện nay PLC có thể thực hiện được các phép tính số học, logic và làm việc được với cả tín hiệu liên tục, trong một số trường hợp PLC được sử dụng thay cho máy tính (một số hãng dùng từ PC-Programmable Controller để chỉ PLC).

Một hệ thống phức tạp thường gồm máy tính (vi xử lý) thực hiện những công việc phức tạp và PLC thực hiện các công việc mang tính chất tuần tự. Máy tính và PLC kết nối với nhau qua đường truyền nối tiếp và trao đổi thông tin cho nhau. Nhiều máy tính và PLC kết nối với nhau theo mạng điều khiển.

PLC gồm các thành phần chính sau:

- Khối CPU (Vi xử lý).
- Khối nhớ RAM, ROM, EPROM, EEPROM.
- Khối nhập.
- Khối xuất.
- Bộ lập trình cầm tay.
- Nguồn.
- Pin nuôi.
- Thẻ nhớ.
- Module mở rộng.



Hình 4.1. Cấu trúc PLC



Chương trình điều hành của nhà sản xuất, chứa trong bộ nhớ ROM (EPROM), thực hiện các công việc sau:

- Kiểm tra hoạt động bản thân PLC.
- Đọc tín hiệu vào ở khối nhập.
- Chuyển đổi chương trình người dùng chứa ở RAM hay thẻ nhớ sang mã máy của vi xử lý để vi xử lý thực hiện.
- Xuất tín hiệu ra khối xuất.
- Giao tiếp vi xử lý với bộ lập trình cầm tay (hand held programming console) hay với máy tính.
- Giao tiếp nối tiếp RS-232 hoặc RS485.

Chương trình người dùng đưa vào PLC, tùy trường hợp, từ bộ lập trình cầm tay, bàn phím trên PLC hay từ máy tính và chứa vào RAM, một nguồn pin nuôi RAM khi cắt điện nguồn, có một tụ điện trị số khá lớn mắc song song với chân cấp nguồn của RAM để đảm bảo chương trình và dữ liệu cần thiết vẫn còn lưu lại một thời gian sau khi cắt nguồn PLC hay pin. Trong trường hợp cần thiết PLC hỗ trợ nạp chương trình vào thẻ nhớ EPROM hay EEPROM.

Bộ nguồn cho PLC có thể lấy từ nguồn xoay chiều hay nguồn một chiều 24V.

Bộ lập trình cầm tay và máy tính lập trình ghép nối với PLC qua đường truyền nối tiếp.

PLC có thể chế tạo dưới dạng khối gắn kết gồm các khối nguồn xử lý, bộ nhớ, khối nhập và xuất cùng chung trong một vỏ nhựa, hoặc theo dạng module (đơn thể) gồm module nguồn, module CPU và các module nhập xuất, module chức năng ...

PLC nhận tín hiệu vào và xuất tín hiệu ra dạng ON/OFF song song, nối tiếp hay dạng tương tự. Với các module phù hợp có thể cho PLC phát ra các tiếng nói cảnh báo hay hướng dẫn.

Các module chức năng giúp mở rộng khả năng của PLC như khuếch đại đo nhiệt độ, điều khiển quá trình vòng kín, điều khiển vị trí, ghép nối modem, mạng công nghiệp.

Quá trình điều khiển có thể hiển thị lên màn hình kèm với các thông số trạng thái nhờ phần mềm giao diện người-máy (HMI Human Machine Interface). Màn hình thường kết hợp với các phím bấm (OP Operator Panel) để điều khiển và quan sát thông số quá trình.

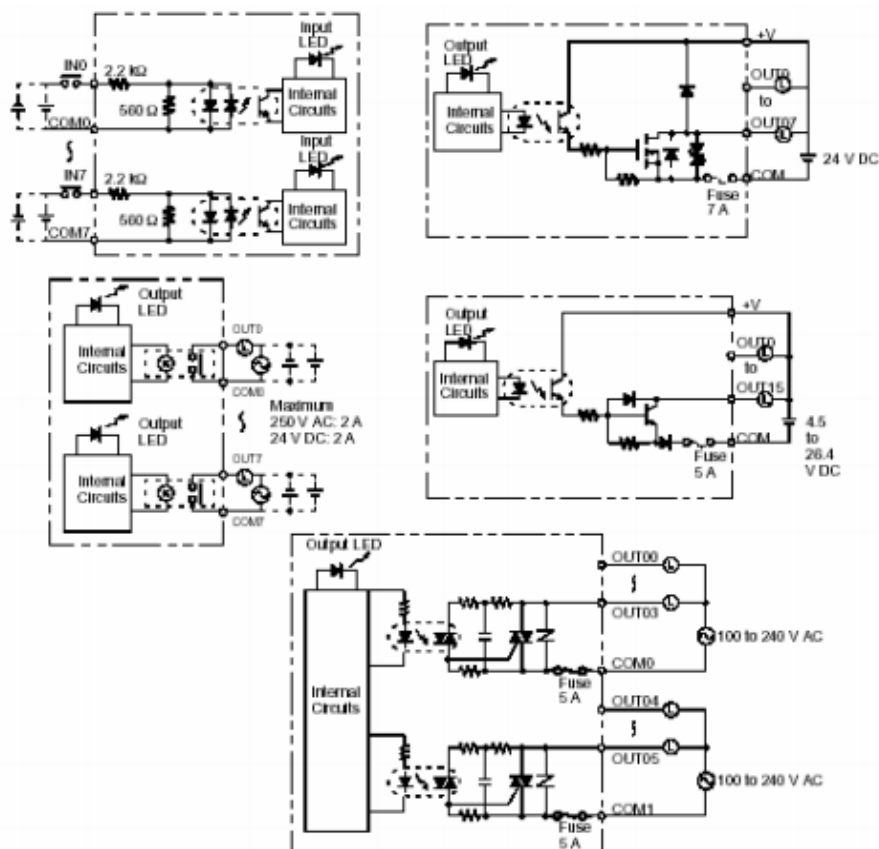
PLC được thiết kế để làm việc trong môi trường công nghiệp do đó mức tín hiệu logic vào là 24V; đối với tín hiệu tương tự nhỏ từ cặp nhiệt hay nhiệt điện trở, có sẵn khối khuếch đại chống nhiễu và khô trôi đi kèm... Do PLC làm việc theo chu kỳ quét nên nó không đáp ứng với tín hiệu thay đổi quá nhanh, điều này hạn chế áp dụng PLC cho việc điều khiển vòng kín các đối tượng có quán tính nhỏ nhưng lại gia tăng độ tin cậy chống nhiễu của thiết bị.

Các tín hiệu xuất/nhập số và tương tự của PLC thường được ghép nối thông qua optocoupler để bảo đảm an toàn. Hình 4.2 trình bày sơ đồ khối nhập và xuất số.

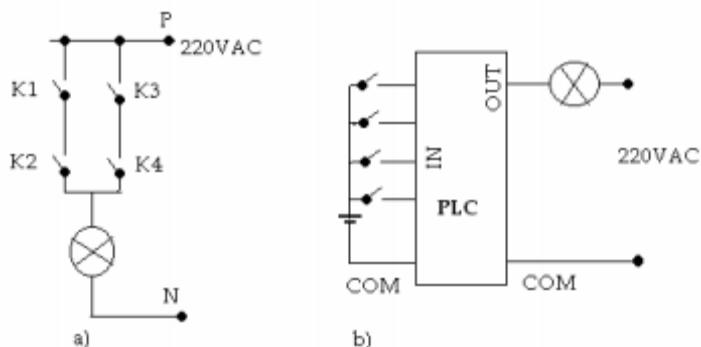
PLC thực hiện chương trình chứa trong bộ nhớ người dùng (UM-User Memory) theo chu kỳ quét. Một chu kỳ quét bắt đầu từ lệnh đầu tiên và kết thúc ở lệnh cuối cùng. Ở mỗi chu kỳ quét PLC đọc trạng thái đầu vào, thực hiện chương trình, cập nhật đầu ra. Thời gian thực hiện chu kỳ quét từ 0,1ms đến hàng chục ms tùy theo vận tốc xử lý của CPU và độ dài của chương trình. Thời gian thực hiện một lệnh cơ bản nhất khoảng dưới 1 μ s.

Chương trình PLC được viết dưới 3 dạng:

- Giản đồ thang (Ladder diagram – LAD).
- Khối hàm (Control System Flowchart – CSF, FBD Function Block Diagram).
- Bảng phát biểu (Statement list – STL)



Hình 4.2. Sơ đồ khối nhập và xuất số

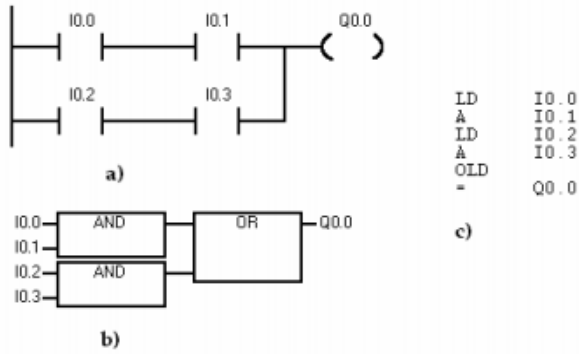


Hình 4.3: a) Mạch tiếp điểm; b) Sơ đồ kết nối PLC

Phương pháp giản đồ thang tương tự sơ đồ role, dạng FBD giống như các sơ đồ trong kỹ thuật số, còn dạng STL tương tự các dòng lệnh của vi xử lí. Tùy theo hãng chế tạo có thể lập trình cho PLC bằng một hay nhiều dạng biểu diễn ở trên.

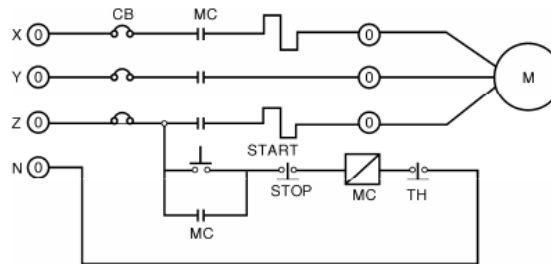
Ví dụ: Xét sơ đồ tắt mở đèn dùng 4 tiếp điểm (Hình 4.3)

Ta có thể hiển thị chương trình bằng 3 dạng như Hình 4.4



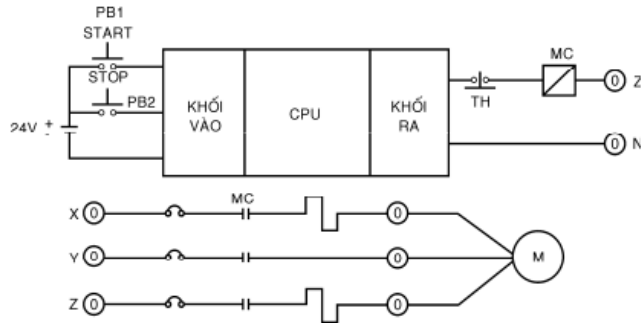
Hình 4.4: a) Dạng LAB; b) Dạng FBD; c) Dạng STL

Ví dụ: Điều khiển động cơ xoay chiều theo sơ đồ Hình 4.5a.



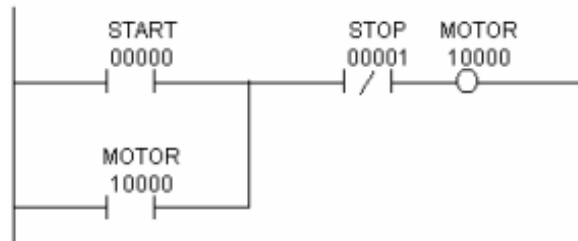
Hình 4.5

Sơ đồ hình 4.5 biến đổi thành sơ đồ điều khiển dùng PLC như hình 4.5b.



Hình 4.5b

Chương trình điều khiển dạng LAB (Hình 4.5c)



Hình 4.5c

Các nút nhấn PB1 và PB2 nối với hai đầu vào có địa chỉ lần lượt là 00000 và 00001. Cuộn dây contactor MC nối với đầu ra có địa chỉ 10000. Chương trình dạng STL như sau:

| Rung No. | Prg. Addr. | Instruction | Address | Name |
|----------|------------|-------------|---------|-------|
| 0001 | 0000 | LD | 00000 | START |
| | 0001 | OR | 10000 | MOTOR |
| | 0002 | AND NOT | 00001 | STOP |
| | 0003 | OUT | 10000 | MOTOR |

Việc lập trình cho PLC được thực hiện theo các bước sau:

- Xác định thứ tự làm việc của máy.
- Vẽ lưu đồ hệ thống.
- Gán các địa chỉ xuất nhập.
- Viết chương trình dạng LAD hay STL và nạp vào PLC.
- Kiểm tra chương trình và sửa lỗi.
- Gắn các đầu nhập và xuất cho PLC.
- Chạy chương trình và sửa lỗi.
- Lưu lại chương trình trên hai địa hoặc/và giấy.

Có rất nhiều hãng sản xuất PLC với nhiều kiểu khác nhau và khó liệt kê hết được:

OMRON: ZEN, CPM1A, CPM2, C200H, CQM1H, CS1.

SIEMENS: LOGO, S5-90U, S5-95U, S5-115U, S5-135U, S5-155U, S7-200, S7-300 ...

ALLEN-BRADLEY: Micrologic1000, SLC500, PLC5, LOGIX.

MITSUBISHI Alpha, FX, Melsec-Q.

SCHNEIDER TSX.

Dưới đây ta sẽ phân tích hoạt động của PLC hãng OMRON và SIEMENS.



a)LOGO

b) S7-200



c) S7-300 và màn hình OP

Hình 4.6: PLC SIEMENS



a)CPM

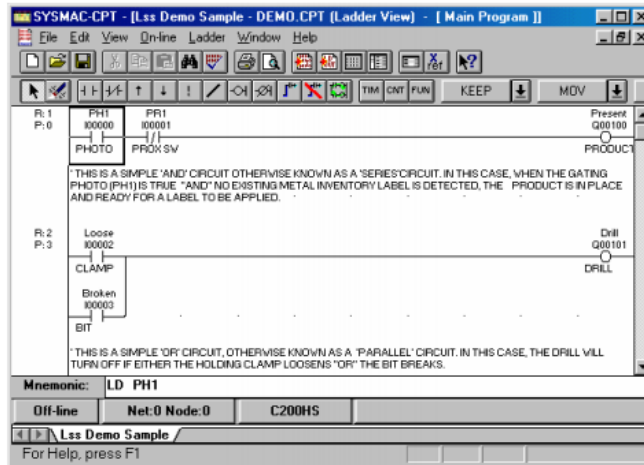
b)C200

Hình 4.7: PLC OMRON

4.2. PLC OMRON

4.2.1. Phần mềm lập trình.

Phần mềm lập trình cho PLC OMRON rất đa dạng. Dạng LAD và dạng STL được đưa vào PLC thông qua máy tính với các phần mềm lập trình như Sysmac Support Software (SSS), Syswin, Sysmac-CPT, CX-Programmer. Ngoài ra còn có thể lập trình dạng STL nhờ bộ lập trình cầm tay (Programming console).



Hình 4.8: Giao diện phần mềm CPT

Các phần mềm lập trình giúp soạn thảo, sửa chữa chương trình, kết nối với PLC, điều khiển PLC ở ba chế độ RUN, STOP và MPNITOR. Chế độ Stop (Program) dùng để nạp chương trình từ máy tính xuống PLC (download) hay sao chép chương trình trong bộ nhớ PLC lên máy tính (upload), ở chế độ RUN và Monitor giá trị các đầu vào ra, các ô nhớ, timer, counter được hiển thị trên chương trình, riêng ở chế độ MONITOR có thể thay đổi nội dung các ô nhớ. Chương trình chứa trong PLC có thể cài mật mã để tránh sao chép trộm.

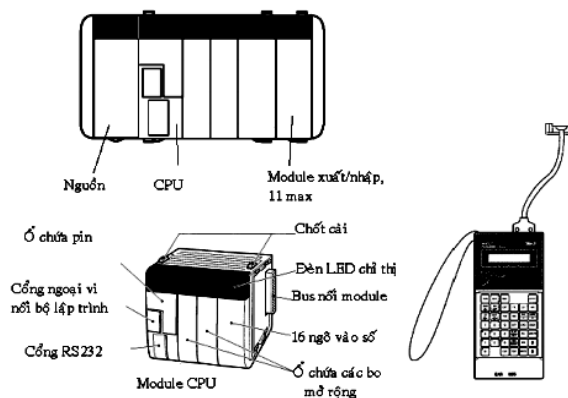
4.2.2. Sơ lược về cấu hình PLC OMRON.

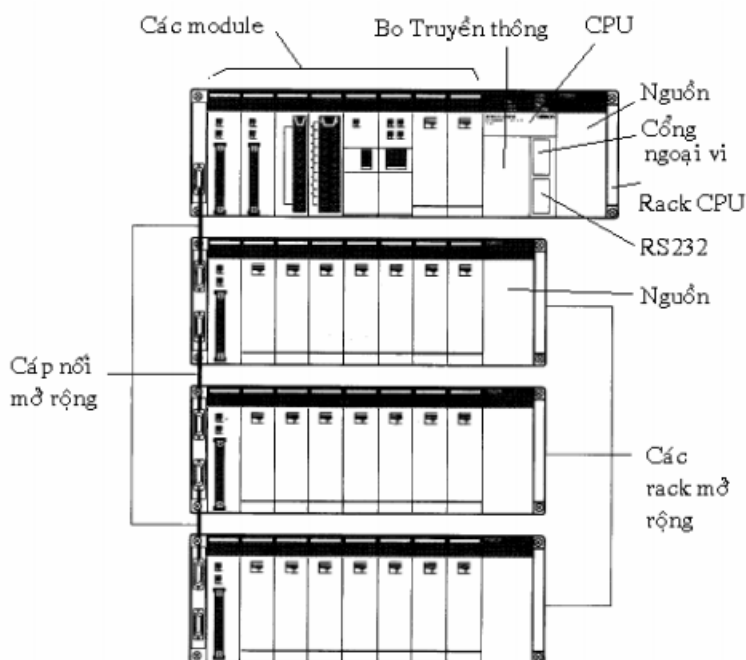
Trong phần này chúng ta chỉ khảo sát ba loại là CQM1, CPM1 và C200H. CQM1 có cấu trúc dạng module nguồn, CPU và các module xuất/nhập. Có thể ghép tối đa đến 11 module xuất/nhập. Nếu dùng module mở rộng thì có thể ghép thêm đến 5 module xuất/nhập. Các module ghép với nhau thông qua Bus nối bên hông, toàn bộ đặt trên đường ray (rail).

Loại C200H có cấu trúc giá (rack) gồm các module gắn trên mặt đế (back plane), giá CPU gồm module nguồn, CPU, các module xuất/nhập, số module gắn vào tùy loại đế, tối đa là 10, muốn thêm module thì dùng các giá đỡ mở rộng, tối đa 3 giá mở rộng.

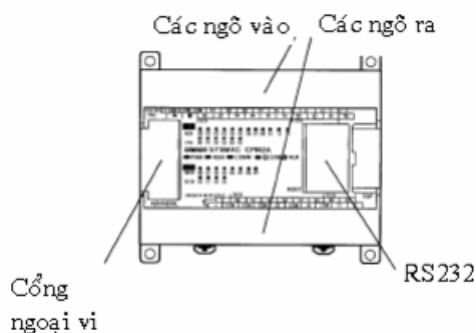
Loại CPM1 có cấu trúc đơn khối, gọn nhẹ, có thể thêm 3 khối mở rộng để tăng khả năng cho PLC.

Hình 4.9. PLC CQM1





:Hình 9.10: C200H



Hình 4.10: PLC CPM2

4.2.3. Cấu trúc địa chỉ bộ nhớ PLC OMRON.

Bộ nhớ PLC OMRON chia làm nhiều vùng IR, SR, TR, HR, AR, TC, DM và UM. Tùy theo kiểu PLC mà các vùng nhớ này có các độ dài khác nhau. Một số vùng nhớ có thể truy xuất theo từng bit hay từ (word = 16 bit), một số chỉ có thể truy xuất theo từ. Một số vùng nhớ được lưu trữ số liệu nhờ tụ điện, thời gian lưu trữ 20 ngày, nếu có pin nuôi thời gian lưu trữ là 5 năm.

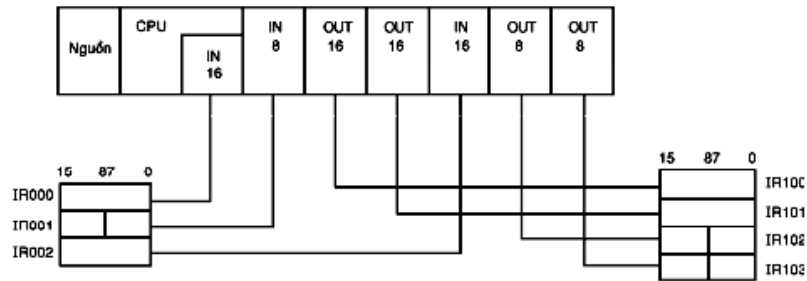
a) Vùng nhớ IR (Internal Relay) chia làm hai vùng nhớ:

- Vùng nhập: Tương ứng với các đầu vào của thiết bị nhập, có thể xử lý theo từ hay bit.
- Vùng xuất: Tương ứng với các đầu ra của thiết bị.

Địa chỉ của các đầu xuất/nhập phụ thuộc kiểu PLC và vị trí các module xuất/nhập.

Địa chỉ từ xuất/nhập tính từ trái sang, bắt đầu từ IR000 cho khối nhập và IR100 cho khối xuất, do đã có sẵn khối nhập gắn sẵn trên khối CPU nên địa chỉ khối nhập gắn thêm vào sẽ bắt đầu từ IR001. Với CPU 11/21-E có tối đa 128 bit xuất/nhập còn với CPU 4X-E tối đa là 192 bit. Địa chỉ tính theo từ gồm 3 số, còn địa chỉ tính theo bit thêm hai số từ 00 đến 15 sau địa chỉ từ.

PLC COM1



Hình 4.11: Qui định địa chỉ module xuất/nhập.

CPM1A

| Nhập | Xuất | Số hiệu |
|------------------------|------------------------|-------------|
| 6 điểm 00000..00005 | 4 điểm 01000..01003 | CPM1A-10CDR |
| 12 điểm | 8 điểm | CPM1A-20CDR |
| 18 điểm | 12 điểm | CPM1A-30CDR |
| 24 điểm | 16 điểm | CPM1A-40CDR |

Có thể gắn thêm đơn vị xuất/nhập mở rộng để thêm đầu xuất/nhập số tương tự cho CPM1A-40 và CPM2.

C200H: PLC C200H được sắp xếp ngược với PLC CQM1, địa chỉ tính từ rãnh bên trái nhất của giá, khối I/O được gắn địa chỉ bắt đầu từ IR000.

| | | | | | |
|--------|--------|--------|-----|-----|-------|
| IR 000 | IR 001 | IR 002 | --- | CPU | Nguồn |
|--------|--------|--------|-----|-----|-------|

- b) Vùng làm việc: Dùng làm vùng nhớ dữ liệu, các vùng nhập và xuất ở trên nếu không liên kết với các module nhập/xuất cũng có thể dùng làm vùng nhớ dữ liệu.
- c) Vùng nhớ SR: (Special Relay): Dùng cho các chức năng đặc biệt như cờ hiệu, tạo xung và làm vùng nhớ dữ liệu.
- d) Vùng nhớ HR (Holding Relay): Dùng để chứa dữ liệu được lưu khi mất điện.
- e) Vùng nhớ LR (Link Relay): Dùng để trao đổi dữ liệu giữa hai PLC.
- f) Vùng nhớ AR (Auxiliary Relay): Dùng làm cờ hiệu và bit điều khiển, được lưu khi mất điện.
- g) Vùng nhớ TC (Timer Counter: Bộ đếm định thời): Dùng cho các lệnh Timer/Counter.
- h) Vùng nhớ TR (Temporary Relay): Dùng để chứa tạm trạng thái ON/OFF của các nhánh rẽ.
- i) Vùng nhớ DM (Data Memory): Chứa thông số cấu hình của PLC và dùng làm vùng nhớ dữ liệu, chỉ truy xuất theo từ, có một số ô nhớ chỉ đọc. Nội dung được giữ lại khi mất điện. Một số ô nhớ dùng để ghi cấu hình (DM6600 ... DM6655).
- j) Vùng nhớ UM (User Program Area) chứa chương trình người sử dụng, dung lượng tùy CPU, được lưu khi mất điện.

Ngoài ra còn có một số vùng nhớ khác được mô tả chi tiết trong tài liệu của nhà sản xuất. Các lệnh tham chiếu bộ nhớ phải ghi rõ tên vùng nhớ, trừ IR và SR.

Dưới đây là các địa chỉ vùng nhớ theo từ (Bảng 4.1)

| Vùng nhớ | | CQM 1 | CPM1 |
|-----------------|-----------------|------------------------------------|-------------------|
| IR | Vùng nhập | IR 000 + IR 011 | IR 000 + IR 009 |
| | Vùng xuất | IR 100 + IR 111 | IR 010 + IR 019 |
| | Vùng làm việc | IR 012 + IR 095 | IR 200 + IR 231 |
| | | IR 112 + IR 195 | |
| IR 216 + IR 219 | | | |
| | IR 224 + IR 229 | | |
| SR | | SR 244 + SR 255 | SR 232 + SR 255 |
| Vùng mở rộng | | IR 200 + IR 215 IR 240 + IR 243 | |
| TR | | TR 0 + TR 7 (bit) | TR 0 + TR 7 |
| HR | | HR 00 + HR 99 | HR 00 + HR 19 |
| AR | | AR 00 + AR 27 | AR 00 + AR 15 |
| LR | | LR 00 + LR 63 | LR 00 + LR 15 |
| TC | | TC000 + TC511 | TC 000 + TC 127 |
| DM | | DM 0000 + DM 6655 | DM 0000 + DM 6655 |

Bảng 4.1

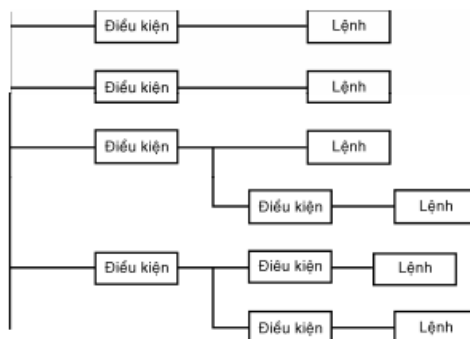
Các vùng nhớ có thể truy cập bit thì thêm hai số từ 00 đến 15 sau địa chỉ từ khi truy cập bit. Với Timer/Counter thì địa chỉ từ kiểm thêm nhiệm vụ là bit trạng thái.

Địa chỉ vùng nhớ IR, SR của C200H khác với CQM1 và CPM1A.

| | |
|-----|-----------------|
| IR1 | IR 000 + IR 235 |
| SR1 | SR 236 + SR 255 |
| SR2 | SR 256 + SR 299 |
| IR2 | IR 300 + IR 511 |

4.3. Các lệnh cơ bản của PLC OMRON.

Chương trình LAD có cấu trúc như hình 4.12 gồm các network, mỗi network gồm các điều kiện, khối điều kiện và lệnh kết nối với nhau, có thể có một lệnh hay nhiều lệnh. Các lệnh được thực hiện theo thứ tự từ trái sang phải và từ trên xuống dưới. Network có dòng chú thích để chương trình dễ hiểu.



Hình 4.12: Sơ đồ chương trình tuyến tính

Khối điều kiện là điều kiện đơn hay tổ hợp logic các điều kiện đơn. Điều kiện đơn biểu thị bằng một tiếp điểm thường mở hay thường đóng. Tổ hợp các điều kiện đơn là kết hợp các tiếp điểm nối tiếp hay/và song song. Mỗi tiếp điểm tương ứng với một bit nhập/xuất hay bit nhớ. Một tiếp điểm thường mở sẽ đóng nếu bit tương ứng ON, một tiếp điểm thường đóng sẽ đóng nếu bit tương ứng OFF.

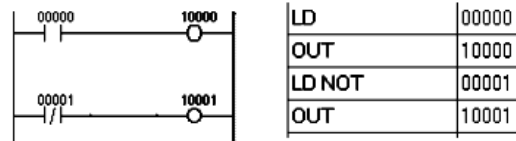
4.3.1. LOAD và LOAD NOT.

Điều kiện đầu của một khối logic trong giản đồ thang tương ứng với lệnh LOAD (LD) đọc tiếp điểm thường mở hay LOAD NOT (LD NOT) đọc tiếp điểm thường đóng.

4.3.2. OUTPUT và OUTPUT NOT. (OUT và OUT NOT)

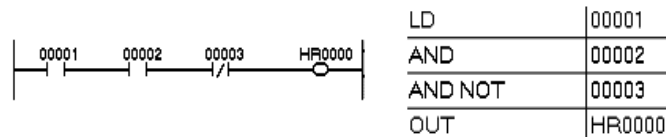
Hai lệnh này điều khiển một bit xuất nhập hay một bit nhớ.

- Lệnh OUT b: Toán hạng sẽ ON nếu điều kiện ON.
- Lệnh OUT NOT b: Toán hạng sẽ ON nếu điều kiện OFF.



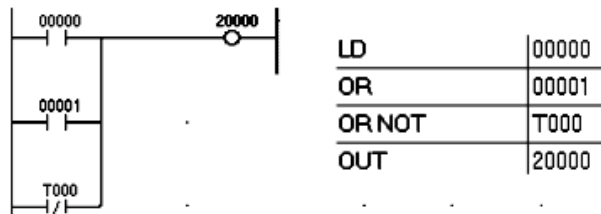
4.3.3. AND, AND NOT.

Hai lệnh này dùng để ghép hai điều kiện nối tiếp nhau:

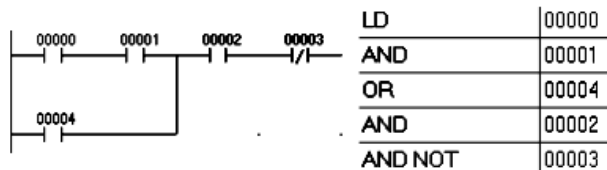


4.3.4. OR, OR NOT.

Ghép hai tiếp điểm song song nhau:



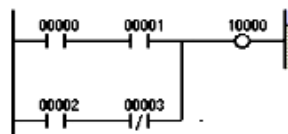
Các lệnh AND, AND NOT, OR, OR NOT có thể kết hợp với nhau:



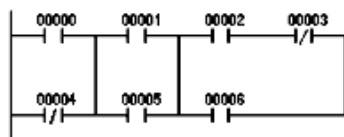
4.3.5. AND LOAD và OR LOAD.

Lệnh AND LOAD (AND LD) kết hợp các khối logic nối tiếp nhau. Lệnh OR LOAD (OR LD) kết hợp các khối logic song song.

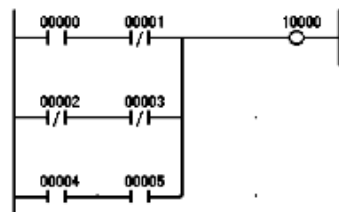
Hình 4.13: Cho các ví dụ sử dụng lệnh AND LD và OR LD.



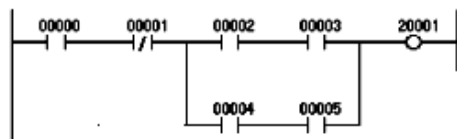
| | |
|---------|-------|
| LD | 00000 |
| AND | 00001 |
| LD | 00002 |
| AND NOT | 00003 |
| OR LD | |
| OUT | 10000 |



| | |
|---------|-------|
| LD | 00000 |
| OR NOT | 00004 |
| LD | 00001 |
| OR | 00005 |
| AND LD | |
| LD | 00002 |
| AND NOT | 00003 |
| OR | 00006 |
| AND LD | |



| | |
|---------|-------|
| LD | 00000 |
| AND NOT | 00001 |
| LD NOT | 00002 |
| AND NOT | 00003 |
| OR LD | |
| LD | 00004 |
| AND | 00005 |
| OR LD | |
| OUT | 10000 |



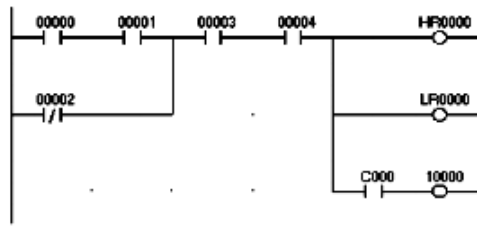
| | |
|---------|-------|
| LD | 00000 |
| AND NOT | 00001 |
| LD | 00002 |
| AND | 00003 |
| LD | 00004 |
| AND | 00005 |
| OR LD | |
| AND LD | |
| OUT | 20001 |

Hình 4.13. Ví dụ các lệnh cơ bản

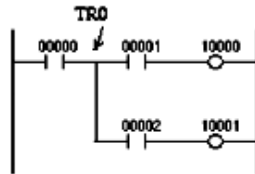
4.3.6. Mã hóa nhiều lệnh bên phải;

Trong trường hợp có nhiều lệnh được thực hiện với cùng điều kiện, ta sẽ viết chương trình STL theo thứ tự từ trên xuống dưới. Trường hợp các lệnh có điều kiện khác nhau ta sẽ dùng các biến nhớ trung gian TR hay dùng lệnh INTERLOCK.

Dùng bit TR: Có 8 bit nhớ TR0 ÷ TR7. Kết quả điều kiện ở điểm rẽ nhánh chứa trong một bit TR.

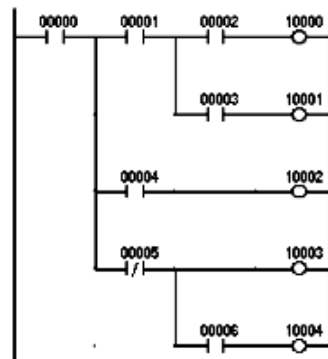


| | |
|--------|--------|
| LD | 00000 |
| AND | 00001 |
| OR NOT | 00002 |
| AND | 00003 |
| AND | 00004 |
| OUT | HR0000 |
| OUT | LR0000 |
| AND | C000 |
| OUT | 10000 |



| | |
|-----|-------|
| LD | 00000 |
| OUT | TR0 |
| AND | 00001 |
| OUT | 10000 |
| LD | TR0 |
| AND | 00002 |
| OUT | 10001 |

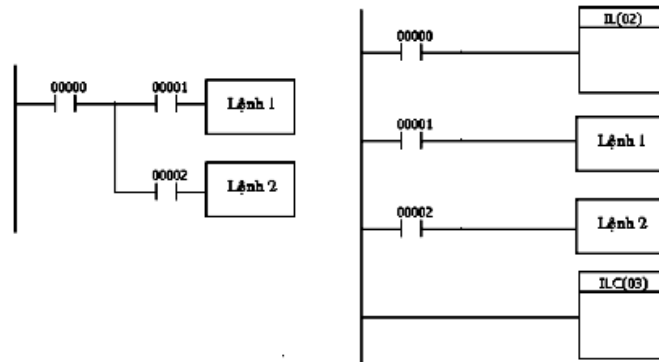
Đôi khi có thể sắp xếp lại sơ đồ để loại bỏ bit TR.



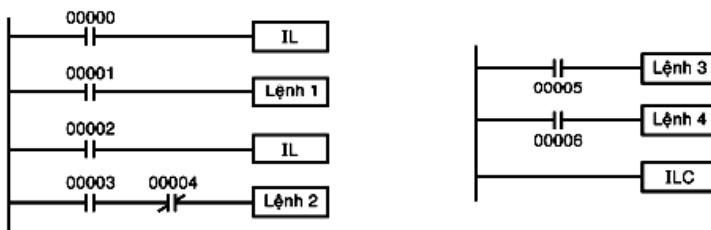
| | | | |
|-----|-------|---------|-------|
| LD | 00000 | LD | TR0 |
| OUT | TR0 | AND | 00004 |
| AND | 00001 | OUT | 10002 |
| OUT | TR1 | LD | TR0 |
| AND | 00002 | AND NOT | 00005 |
| OUT | 10000 | OUT | 10003 |
| LD | TR1 | AND | 00006 |
| AND | 00003 | OUT | 10004 |
| OUT | 10001 | | |

Dùng INTERLOCK IL (02) và INTERLOCK CLEAR ILC (03).

Lệnh IL lưu trữ điều kiện và sử dụng cho các lệnh nằm giữa IL và ILC. Nếu điều kiện cho IL ON thì thực hiện các lệnh nằm giữa IL và ILC, nếu điều kiện cho IL OFF thì không thực hiện các lệnh này. Có thể dùng nhiều lệnh IL và một ILC.



Trong chương trình sau, nếu IR00000 off thì các lệnh 1 đến 4 từ IL đến ILC không được thực hiện vì điều kiện off. Nếu IR00000 on thì lệnh 1 được thực hiện tùy trạng thái IR00001, trạng thái của IR00002 được xét để làm điều kiện cho IL kế.

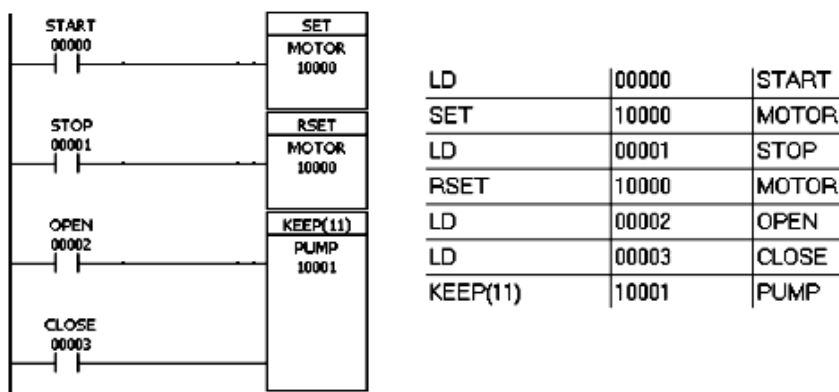


4.3.7. Lệnh SET và RESET (RSET).

- Lệnh SET b: Bit b on khi điều kiện on và giữ nguyên b on khi điều kiện trở thành off.
- Lệnh RSET b: Bit b off khi điều kiện on và giữ nguyên off khi điều kiện trở thành off.

4.3.8. Lệnh KEEP b.

Làm bit b on khi S on và bit b off khi R on. Các lệnh KEEP SET, RSET, DIFU, DIFD dùng với các bit IR, SR, AR, HR, LR riêng biệt OUT dùng với IR, SR, AR, HR, LR, TR.



4.3.9. Lệnh vi phân lên và vi phân xuống.

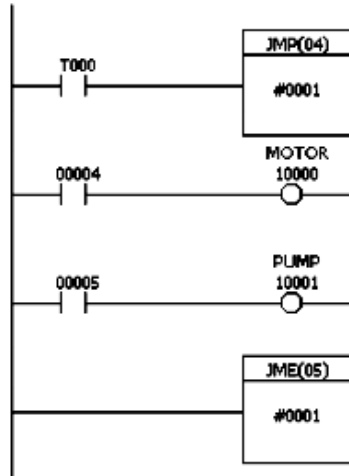
- DIFU(13) b: Bit b on trong một chu kỳ khi điều kiện từ off sang on.
- DIFD(14) b: Bit b on trong một chu kỳ khi điều kiện từ on sang off.

4.3.10. Lệnh JUMP (JMP (04)) và JME (05).

Lệnh JMP nn và JME nn đóng khung một đoạn chương trình. Nếu điều kiện cho lệnh JMP là on thì coi như không có lệnh JMP và chương trình thực hiện bình thường. Nếu điều kiện cho JMP off thì bỏ qua các lệnh trong khoảng JMP và JME nhưng vẫn giữ nguyên trạng thái các bit nhớ cũng như timer và counter, nn là số từ 00 đến 99. Các số bit được dùng một lần trong chương trình, riêng lệnh JMP 00 có thể dùng nhiều lần với chỉ một lệnh JME 00.

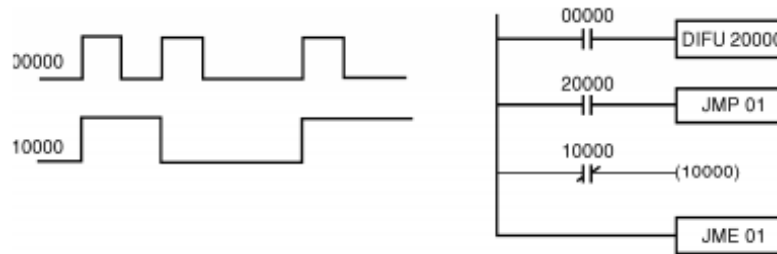
Ví dụ: Khi T000 on đoạn chương trình giữa JMP 01 và JME 01 được thực hiện, bit 10000 và 10001 phụ thuộc điều kiện 00000 và 00001, khi T000 off trạng thái của bit 10000 và 10001 được giữ nguyên.

Khi nhấn contact 00000 lệnh DIFU 20000 làm 20000 ON trong một chu kỳ quét do đó 10000 sẽ ON, đến chu kỳ quét sau 20000 OFF nên không thực hiện lệnh OUT 10000 mà giữ nguyên trạng thái của 10000. Khi nhấn 00000 lần nữa thì thực hiện lệnh OUT 10000 với điều kiện OFF do đó 10000 sẽ OFF.

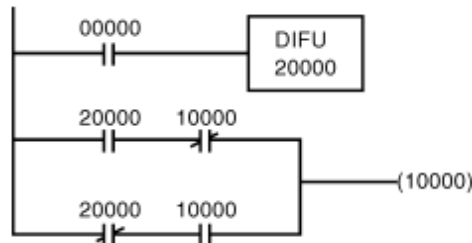


| | |
|---------|-------|
| LD | T000 |
| JMP(04) | #0001 |
| LD | 00004 |
| OUT | 10000 |
| LD | 00005 |
| OUT | 10001 |
| JME(05) | #0001 |

Ví dụ:



Nếu không dùng lệnh JMP JME thì có thể dùng chương trình sau:

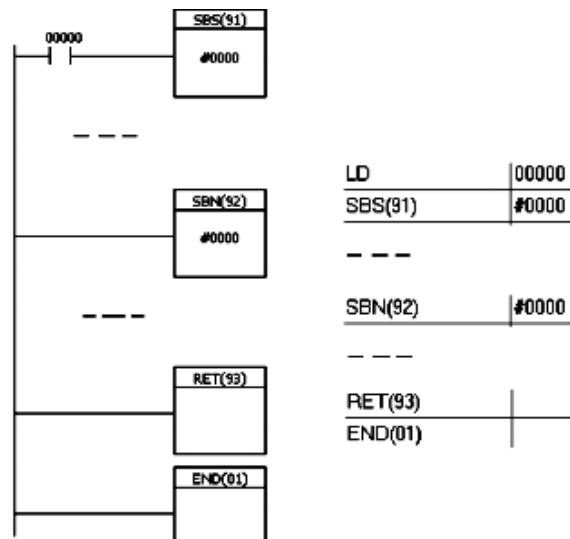


4.3.11. Lệnh chương trình con SBS (91), SBN (92), RET (93).

Lệnh SBN nnn và RET đóng khung chương trình con còn lệnh SBS nnn dùng để gọi chương trình con. Mỗi chương trình con có một số hiệu từ 000 đến 255. Chương trình con được đặt ở đoạn cuối của chương trình chính, trước lệnh END. Các lệnh DIFU, DIFD không nên đặt trong chương trình con. Dùng chương trình con có ưu điểm là chương trình dễ đọc và thời gian thực hiện chương trình ngắn hơn.

Lệnh END (01) là lệnh cuối cùng của chương trình.

4.3.12. Địa chỉ gián tiếp.





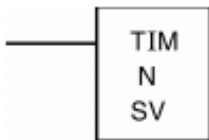
Địa chỉ gián tiếp được thực hiện qua vùng nhớ DM và ký hiệu là *DM. Từ nhớ *DM sẽ chứa địa chỉ của từ nhớ DM muốn sử dụng.

4.3.13. Lệnh vi phân.

Lệnh vi phân ký hiệu bởi dấu @ đứng trước lệnh. Lệnh này chỉ thực hiện một lần khi điều kiện đi từ OFF sang ON.

4.4. Các lệnh định thời bộ đếm.

4.4.1. Lệnh TIMER.

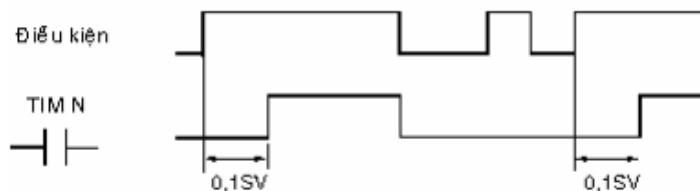


N: Số từ 0 ÷ 511 tùy loại CPU.

SV: Giá trị đặt BCD, 0000 đến 9999 là nội dung ô nhớ.

IR, SR, AR, DM, HR, LR hay hằng số #.

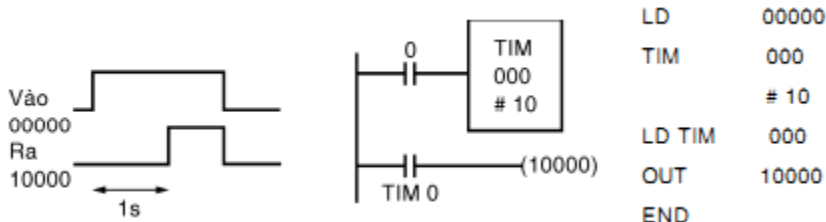
Khi điều kiện off mạch định thời Reset về SV, cờ TIM N off, khi điều kiện on nội dung mạch định thời giảm cứ mỗi 0.1s; sau thời gian 0.1SV giây, cờ TIM N sẽ ON cho đến khi điều kiện OFF hay ngắt điện nguồn. Lệnh TIM dùng với IL sẽ Reset khi điều kiện chỉ IL là OFF.



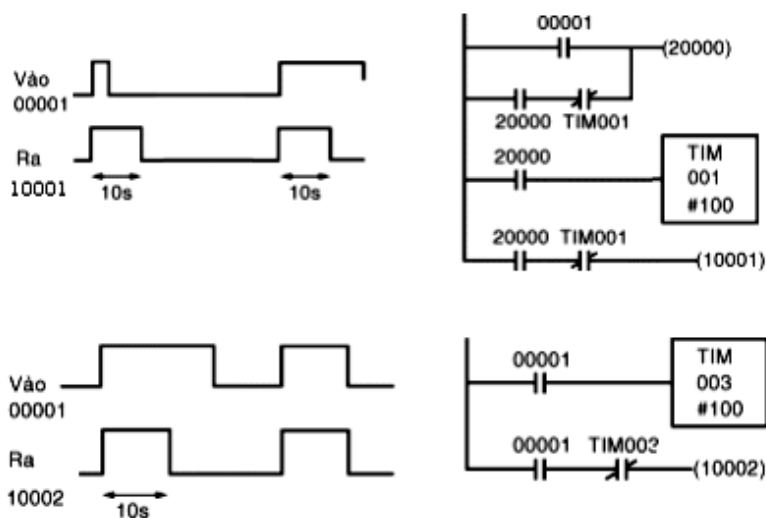
TIMH (15) cũng giống TIM nhưng thời gian trên là 0.01SV và N trong khoảng 000 ÷ 015.

Sau đây là các áp dụng thường gặp của Timer.

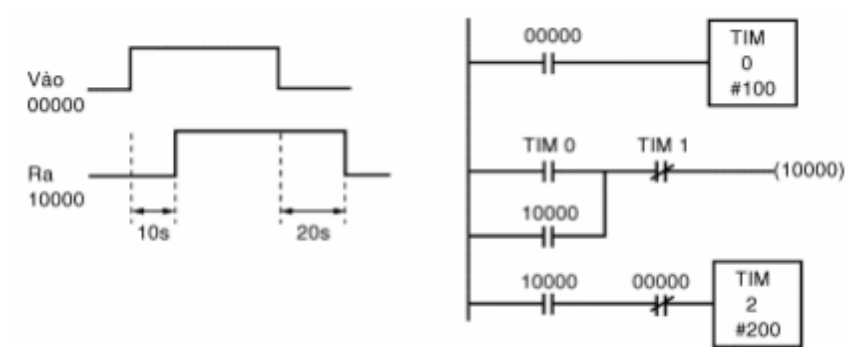
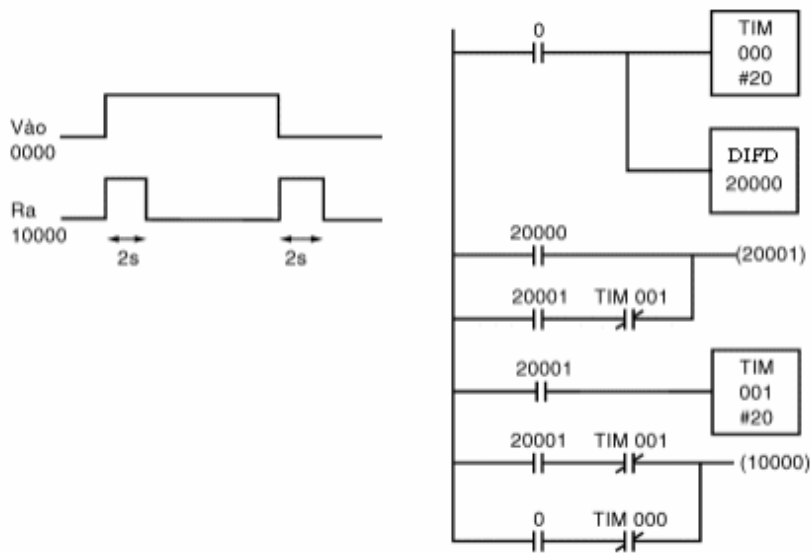
Mạch ON Delay:



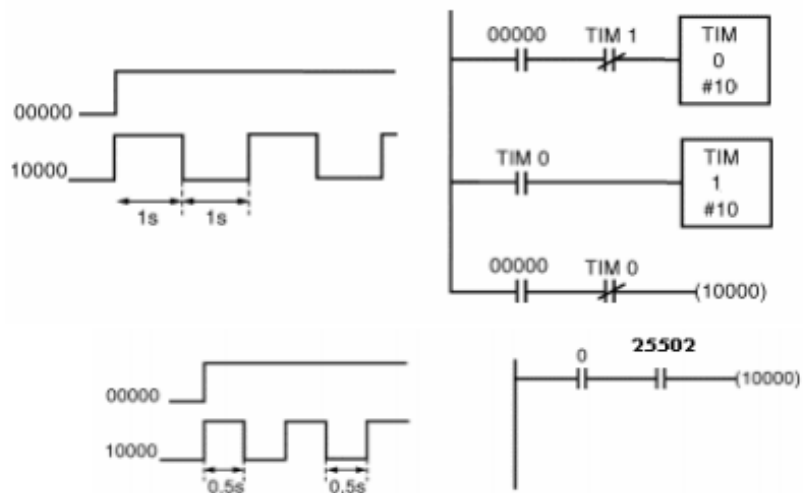
Mạch đơn ổn:



Mạch ON/OFF Delay:

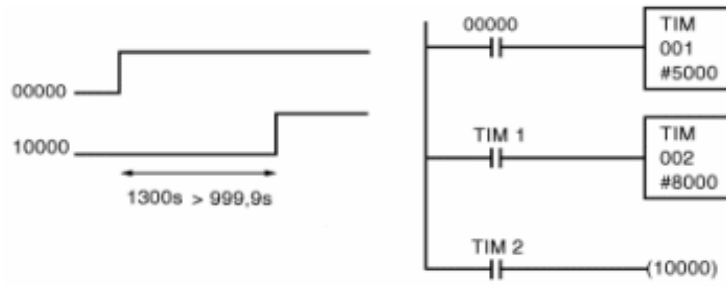


Mạch nhấp nháy:

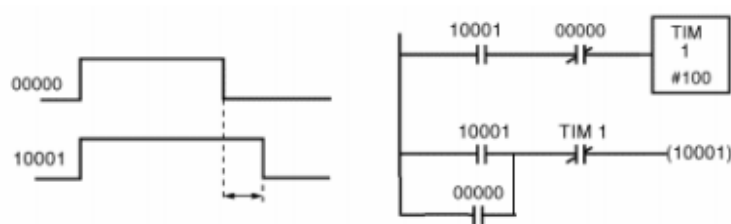




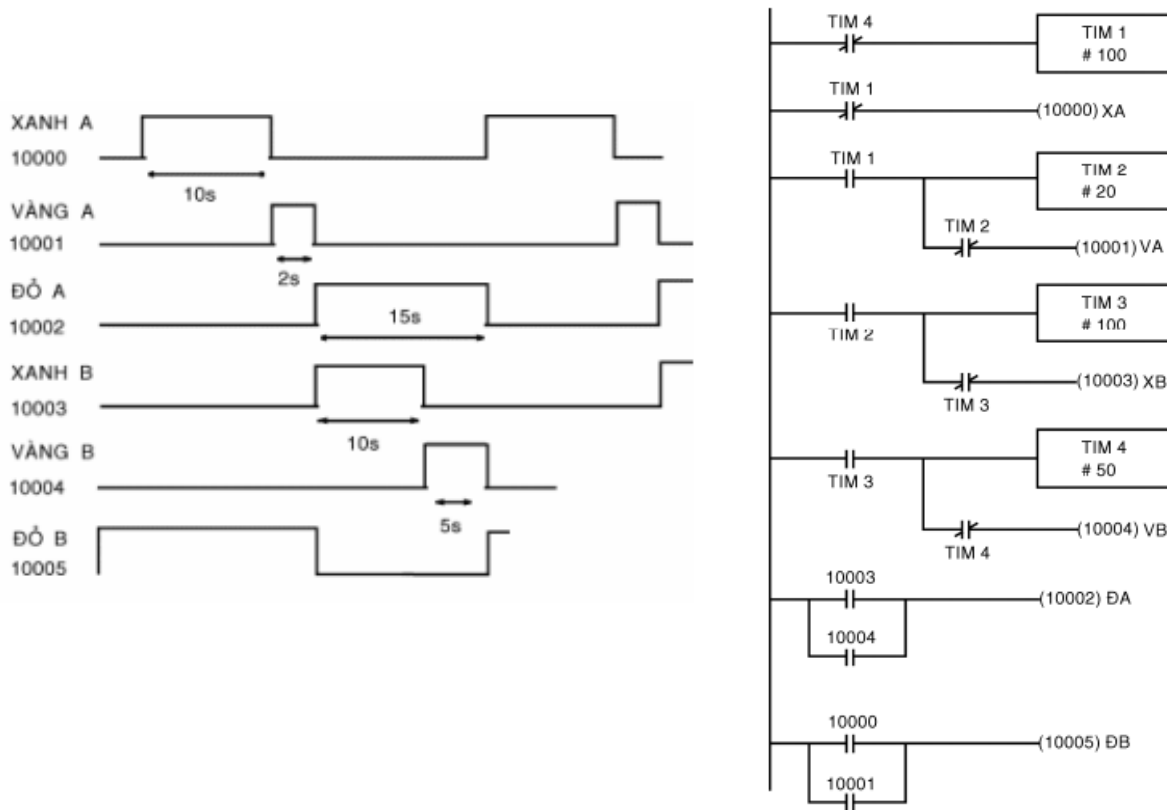
Mạch định thời dài:



Mạch OFF Delay



Đèn giao thông:

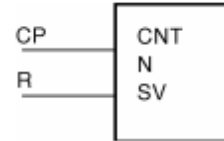


4.4.2. Lệnh đếm CNT.

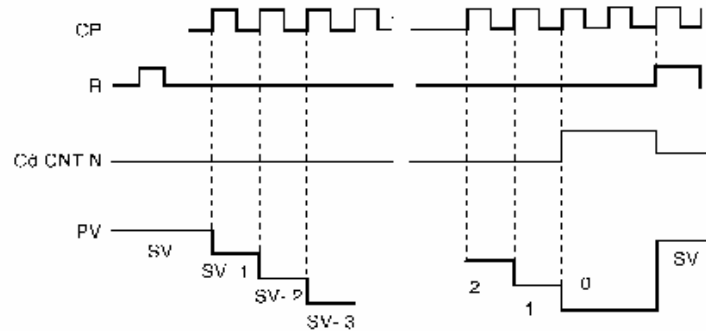
N: Số từ 0 đến 511 tùy loại CPU.

SV: Trị đặt cho bộ đếm (BCD): 0000,... 9999.

IR, SR, AR, DM, HR, LR, #.



R là đầu vào xóa, khi R từ OFF sang ON nội dung PV của bộ đếm được đặt ở SV. Khi R trở về OFF, bộ đếm bắt đầu hoạt động, khi có xung CP từ OFF sang ON, PV sẽ giảm đi 1. PV không thay đổi khi CP từ ON sang OFF. Khi PV = 0 thì nội dung của CNT giữ nguyên ở 0, còn CNT N sẽ ON và giữ nguyên ở ON cho đến khi được Reset bởi R.



Nội dung CNT không bị xóa khi ngắt nguồn hay trong đoạn chương trình IL.

Chú ý là trong PLC có sẵn một số bit đặc biệt sau:

25400: Xung nhịp chu kỳ 1 phút; 25401:0.02 sec.

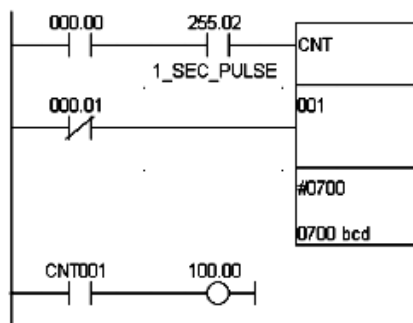
25500: 0.1 sec; 25501:0.2 sec; 25502:1.0 sec.

25313: Cờ luôn luôn ON; 25314: Cờ luôn luôn OFF.

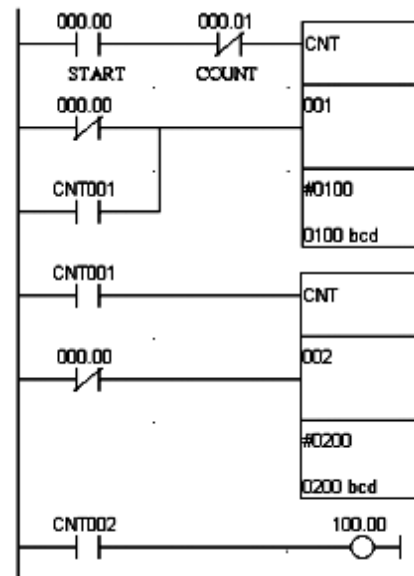
25315: Cờ ON ở chu kỳ đầu.

Ví dụ: Mạch định thời 11'40''

Mạch đếm số lượng lớn 20000 xung.



```
LD 0000
AND 25502
LD NOT 00001
CNT #0700
LD CNT 001
OUT 10002
```





Mạch đóng gói: Đóng gói 10 quả táo cho vào hộp.

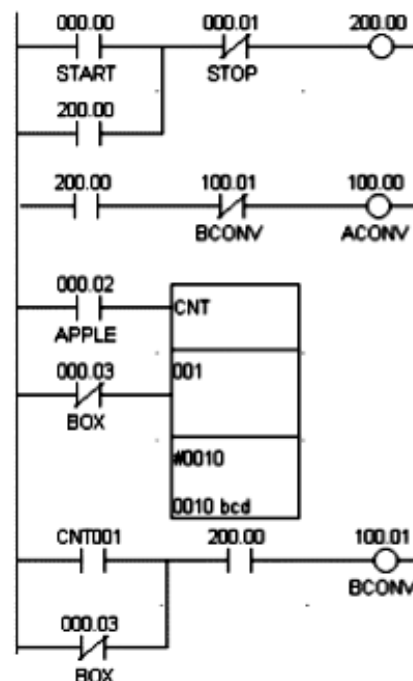
Cảm biến táo: 00002

Cảm biến hộp: 00003

Băng chuyền táo: 10000

Băng chuyền hộp: 10001

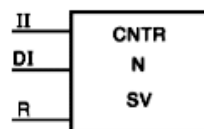
Khi bấm Start, nếu không có hộp che cảm biến thì băng chuyền hộp chạy, đưa vào vị trí nhận táo. Lúc đó băng chuyền hộp ngừng, băng chuyền táo chạy, đưa táo vào hộp. Đếm đủ 10 táo, băng chuyền hộp chạy, bộ đếm Reset khi hộp rời khỏi vị trí, băng chuyền táo ngừng, chờ hộp mới vào vị trí.



4.4.3. Bộ đếm thuận nghịch CNTR (12).

N: 0 ÷ 511 tùy loại CPU

SV: HR, LR, IR, SR, AR, DM, #.

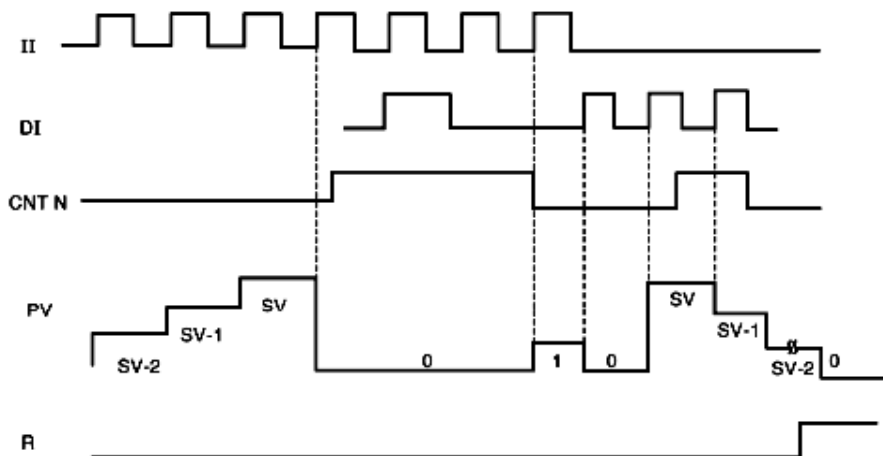


Khi R từ OFF sang ON bộ đếm được xóa về 0, PV = 0. Khi R OFF bộ đếm chuẩn bị đếm.

Khi II từ OFF sang ON thì PV tăng lên 1.

Khi DI từ OFF sang ON thì PV giảm đi 1.

Nếu II và DI cùng lúc từ OFF sang ON thì PV không thay đổi. Khi giảm từ 0, PV sẽ bằng SV và contact CNT N sẽ ON cho đến khi PV giảm. Khi tăng quá SV, PV sẽ bằng 0 và CNT N sẽ ON cho đến khi PV tăng. PV không bị ảnh hưởng bởi ngắt nguồn hay IL.



4.5. Lệnh di chuyển.

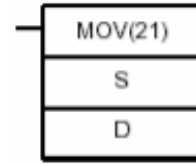
4.5.1. Di chuyển ô nhớ.

MOV (2,1): Chuyển một từ từ S đến D.

S: IR, SR, AR, DM, HR, TC, LR, #.

D: IR, SR, AR, DM, HR, LR.

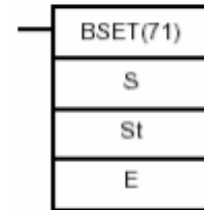
MVN(22): Chép đảo của từ từ S sang D.



BSET: Chép một từ S sang một khối nhớ từ St đến E

S: IR, SR, AR, DM, HR, TC, LR, #.

St, E: IR, SR, AR, DM, HR, TC, LR.



XFER: Chép một khối N ô nhớ bắt đầu từ S sang khối N ô nhớ bắt đầu từ D.

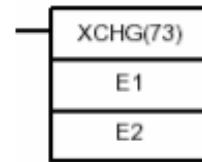
N: IR, SR, AR, DM, TC, LR, #.

S, D: IR, SR, AR, DM, HR, TC, LR.

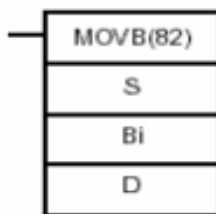


XCHG: Trao đổi 2 ô nhớ E1 và E2.

E1, E2: IR, SR, AR, DM, HR, TC, LR



4.5.2. Di chuyển một số bit.



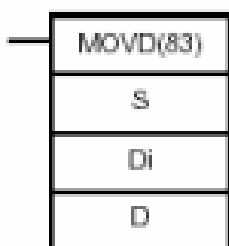
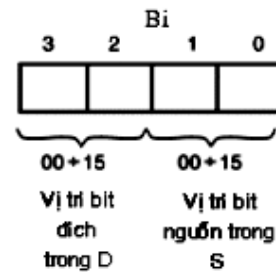
MOVB chép một bit của từ S sang một bit của từ D.

Vị trí bit chỉ bởi Bi.

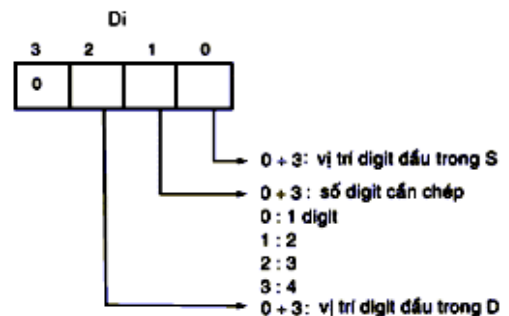
S: IR, SR, AR, DM, HR, LR, #.

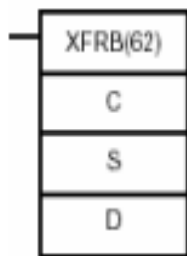
D: IR, SR, AR, DM, HR, LR.

Bi: IR, SR, AR, DM, HR, TC, LR, #.



MOVD tương tự như MOVB nhưng chép từ 1 đến 4 digit.



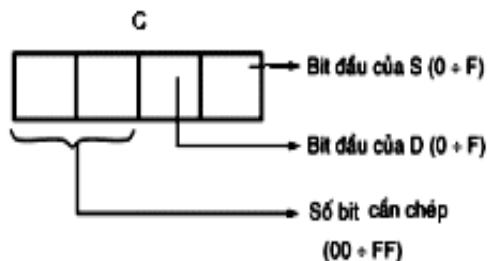


XFRB (62): Chuyển một số bit từ F sang D, qui định bởi C.

C: IR, SR, AM, DM, LR, HR, TC, #.

S: IR, SR, AM, DM, LR, HR, TC.

D: IR, SR, AM, DM, LR, HR.



4.6. Lệnh Logic.

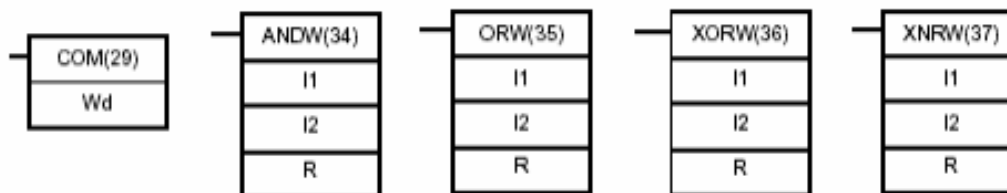
COM: Đảo của các bit Wd.

ANDW: AND hai từ.

ORW: OR hai từ.

XORW: XOR hai từ.

XNRW: Exclusive Nor hai từ.



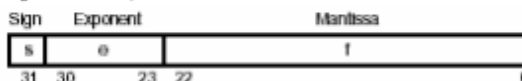
4.7. Lệnh số học.

4.7.1. Các loại số.

PLC OMRON tính toán chủ yếu trên số thập phân BCD 4 hay 8 bit không dấu, số nhị phân có dấu và không dấu 16 bit, 32 bit. Số nhị phân không dấu 16 bit từ 0000 (0) đến FFFF (65,535), 32 bit từ 00000000 (0) đến FFFFFFFF (4,294,967,295). Số nhị phân có dấu 16 bit dùng mã bù hai, bit 15 là bit dấu, từ 8000 (-32,768) đến FFFF (-1) và 0000 (0) đến 7FFF (32,767). Số nhị phân có dấu 32 bit có giá trị từ 80000000 (-2,147,483,648) đến FFFFFFFF (-1) và 00000000 (0) đến 7FFFFFFF (2,147,483,647).

Trong một số trường hợp sử dụng số chấm nổi (số thực), chiếm 32 bit biểu thị bằng dấu s, số mũ e và định trị f:

$$(-1)^s 2^{e-127} (1 + f * 2^{-23})$$



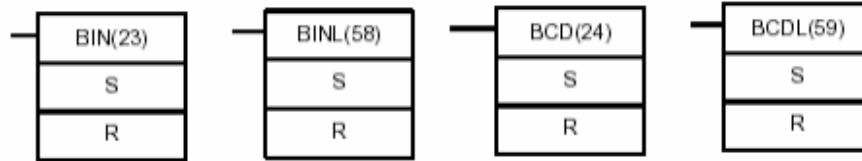
Các cờ hiệu liên quan lệnh số học là:

- N: Cờ âm 25401.
- OF: Cờ tràn dương 25404.
- UF: Cờ tràn âm 25405.
- ER: Lệnh sai 25503.
- CY: Cờ nhớ 25504.

- GR: Cờ nhỏ hơn 25505.
- EQ: Cờ bằng 25506.
- LE: Cờ lớn hơn 25507.

Cờ CY được Set/Reset bởi lệnh STC/CTC

4.7.2. Lệnh đổi dữ liệu BCD – Nhị phân.



BIN: Đổi số BCD 4 digit trong S ra số nhị phân trong R.

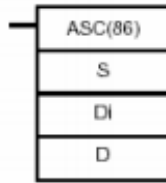
S: IR, SR, AR, DM, HR, TC, LR.

R: IR, SR, AR, DM, HR, LR.

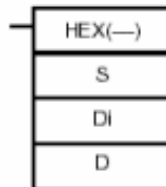
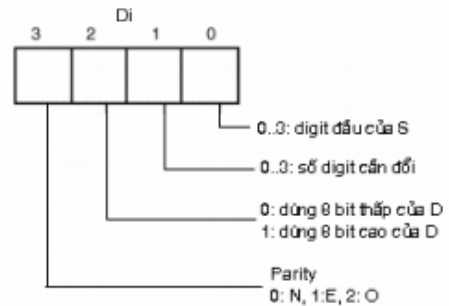
BCD: Đổi số nhị phân 16 bit trong S ra số BCD trong R, nếu kết quả lớn hơn 9999 thì cờ ER báo và R không đổi.

Các lệnh BINL và BCDL dùng cho từ kép.

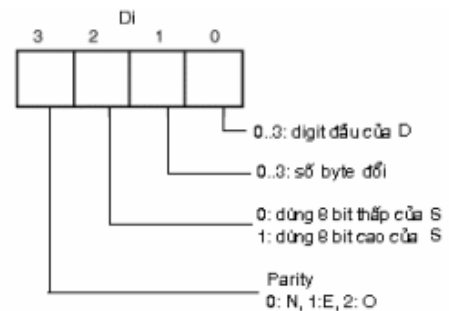
4.7.3. Lệnh đổi HEX-ASCII.



Đổi số nhị phân 4 bit 0... F trong S ra byte mã ASCII trong D theo qui định của D_i



Đổi mã ASCII của số hex 0...F trong S ra số hex trong D

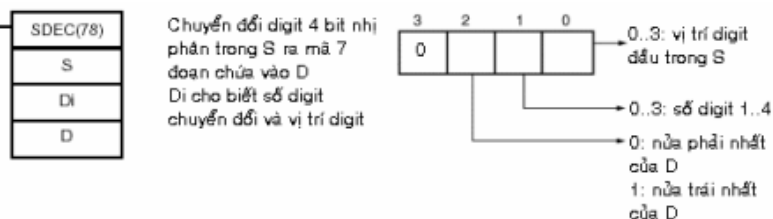


4.7.4. Lệnh đổi số thực – số nguyên.

- | | |
|----------------------------|------|
| Số thực – số nguyên 16 bit | FIX |
| Số thực – số nguyên 16 bit | FIXL |
| Số nguyên 16 bit – số thực | FLT |
| Số nguyên 16 bit – số thực | FLTL |



4.7.5. Giải mã 7 đoạn.

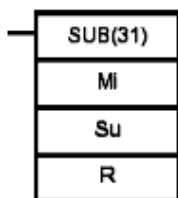
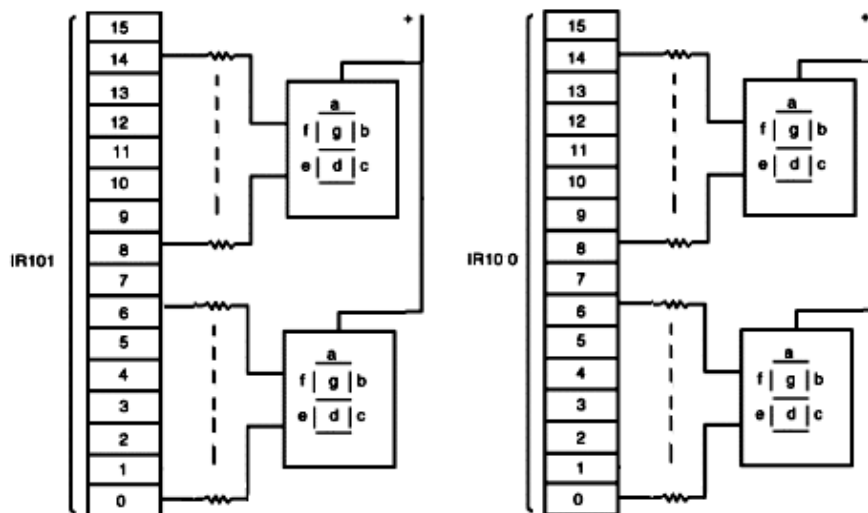


Ví dụ: Muốn hiển thị nội dung ô nhớ DM0000 ra 4 đèn 7 đoạn gắn ở module xuất IR100 và IR101, ta dùng đèn 7 đoạn anode chung, nguồn cấp từ 5V đến 24V, điện trở nối tiếp tùy theo áp nguồn. Lệnh SDEC DM 0000 #0030 100.

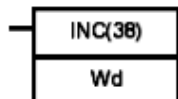
4.7.6. Lệnh BCD không dấu 4 số thập phân.



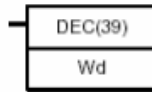
Cộng hai số dạng BCD Au và Ad với bit nhớ CY, kết quả chứa vào R.
 Nếu cờ tràn (quá 9999) thì bit nhớ CY là ON.



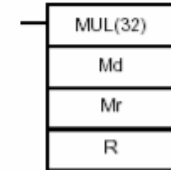
Trừ hai số BCD Mi, Su với bit nhớ CY, kết quả chứa vào R, nếu kết quả âm thì CY là ON và R chứa kết quả phụ 10, muốn có kết quả đúng lấy 0 trừ số này.
 Khi tính toán phải xét CY để chỉnh lại kết quả.



Tăng nội dung BCD của Wd lên 1
 IR, SR, HR, LR, AR, DM
 Cờ EQ on khi kết quả là 0



Giảm Wd đi 1, cờ EQ tác động khi kết quả là 0.

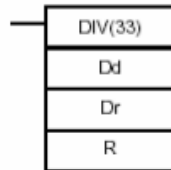


Nhân hai số Md và Mr, kết quả chứa trong R+1 và R.

Nếu có nhớ thì CY = 1.

Md, Mr: IR, HR, LR, AR, SR, DM, TC, #.

R: IR, HR, LR, AR, SR, DM.



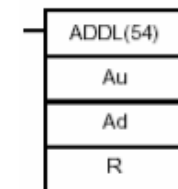
Chia Dd cho Dr, kết quả chứa trong R, dư số trong R+1.

Ví dụ: Trừ ô nhớ HR20 với ô nhớ 120, kết quả chứa vào DM0100

```

LC          00003
CLC(41)
@SUBL(55)  HR 20      120      DM 1000
AND        25502
#BSET(71)  #0000     DM0000    DM0001
CLC(41)
@SUBL(55)  DM0000    DM0100    DM0100
  
```

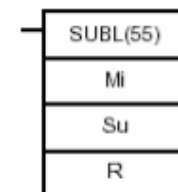
4.7.7. Lệnh BCD không dấu 8 digit.



Cộng Au+1, Au với Ad+1, Ad và CY. Kết quả chứa vào R+1, R.

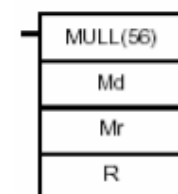
Au, Ad: IR, SR, AR, LR, DM, HR, TC

R: IR, SR, AR, LR, DM, HR.



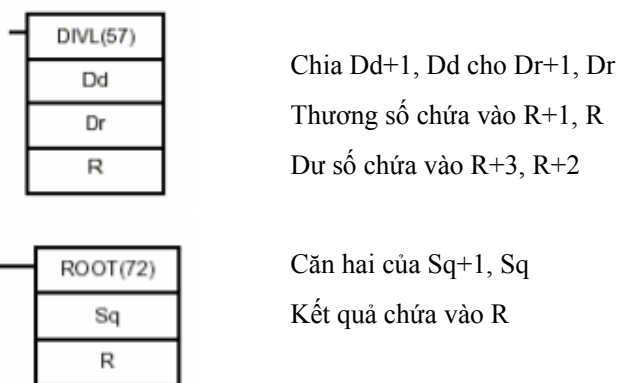
Trừ Mi+1, Mi với Su+1, Su và CY

Kết quả chứa vào R+1, R



Nhân Md+1, Md với Mr+1, Mr

Kết quả chứa vào R+3, R+2, R+1, R



4.7.8. Lệnh số học nhị phân 16 bit không dấu.



ADB: CY ON: Kết quả quá FFFF, OF ON: Kết quả quá 7FFF, UF ON: Kết quả nhỏ hơn 8000, EQ ON: Kết quả 0, N ON: Bit 15 kết quả là 1. Sử dụng các cờ OF, UF khi muốn cộng trừ số có dấu.

SBB: CY ON khi $Mi < Su + CY$ (số không dấu), các cờ khác giống lệnh ADB.

4.7.9. Lệnh số học nhị phân 32 bit không dấu.



ADBL: CY ON: Kết quả quá FFFFFFFF, OF ON: Kết quả quá 7FFFFFFF, UF ON: Kết quả nhỏ hơn 80000000, EQ ON: Kết quả 0, N ON: Bit 15 R+1 là 1. Sử dụng các cờ OF, UF khi muốn cộng trừ số có dấu.

SBBL: CY ON: khi $Mi < Su + CY$ (số không dấu), các cờ khác giống lệnh ADB.

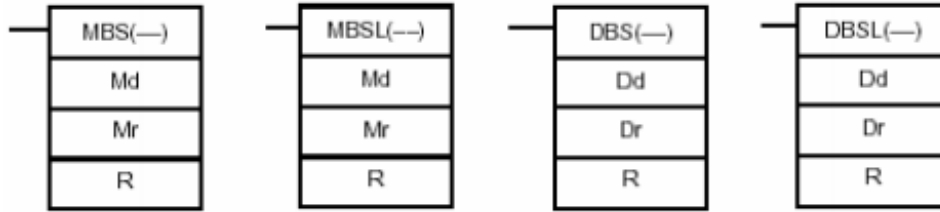
Lệnh nhân chia nhị phân có dấu.

MBS: Nhân 16 bit có dấu, N ON khi bit 15 của R+1 ON.

MBSL: Nhân 32 bit có dấu, N ON khi bit 15 của R+3 ON.

DBS: Chia 16 bit có dấu, N ON khi bit 15 của R ON.

DBSL: Chia 32 bit có dấu, N ON khi bit 15 của R+1 ON.

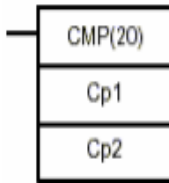


4.7.10. Lệnh số thực.

| | |
|-------------------|------------------|
| Cộng +F | Tang TAN |
| Trừ -F | Cung sin ASIN |
| Nhân *F | Cung cos ACOS |
| Chia /F | Cung tang ATAN |
| Đổi độ ra rad RAD | Căn bậc hai SQRT |
| Đổi rad ra độ DEG | Mũ EXP |
| Sin SIN | Log LOG |
| Cosin COS | |

4.8. Lệnh so sánh.

4.8.1. So sánh hai ô nhớ.



So sánh hai từ nhị phân không dấu Cp1 và Cp2

Cp1, Cp2: IR, SR, AR, HR, TC, LR, #.

Khi so sánh với PV của Timer/Counter giá trị là số BCD.

| Bit | Cp1<Cp2 | Cp1=Cp2 | Cp1>Cp2 |
|----------|---------|---------|---------|
| GR 25505 | OFF | OFF | ON |
| EQ 25506 | OFF | ON | OFF |
| LE 25507 | ON | OFF | OFF |

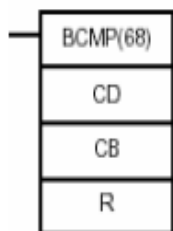
Kết quả so sánh phải đặt liền sau lệnh CMP để bảo đảm giá trị vì trong chương trình có thể có nhiều lệnh so sánh.

Ví dụ: So sánh nếu #1000<(DM0000)<#2000 thì 10000 ON

```

LD 0000          AND 20000
CMP DM0000 #1000  CMP DM0000 #2000
AND 25505        AND 25507
OUT 20000        OUT 10000
LD 00000
    
```

4.8.2. So sánh khoảng.



So sánh số CD với khối 32 ô nhớ bắt đầu từ CB, kết quả chứa trong R.

$CB \leq CD \leq CB+1$: Bit 0 của R ON.

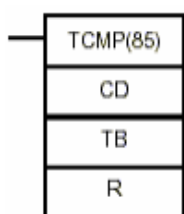
$CB+1 \leq CD \leq CB+i+1$: Bit I của R ON.

$CB+30 \leq CD \leq CN+31$: Bit 15 của R ON.

CD: IR, SR, DM, HR, TC, LR, AR, #; CB: IR, SR, DM, HR, TC, LR.

R: IR, SR, DM, HR, TC, LR, AR.

4.8.3. So sánh bằng.

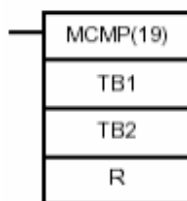


So sánh CD với 16 ô nhớ bắt đầu từ TB, kết quả so sánh chứa vào 16 bit của R. Ví dụ nếu CD bằng TB+i thì bit i của R là ON.

TB, R: IR, SR, DM, HR, TC, LR.

CD: IR, SR, DM, HR, TC, LR, #.

4.8.4. So sánh hai bằng.

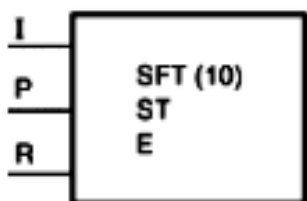


So sánh hai bằng TB1 và TB2 dài 16 bit từ.

Nếu hai từ thứ i bằng nhau thì bit thứ i của R OFF.

4.9. Lệnh ghi dời.

4.9.1. Dời trái.



Thanh ghi dời gồm một loạt từ nhớ 16 bit bắt đầu ở St và chấm dứt ở E. Khi R ON mọi bit của thanh ghi ở trạng thái OFF. Khi R OFF thanh ghi chuẩn bị hoạt động.

Khi P từ OFF sang ON trạng thái ở đầu vào I chứa vào bit 0 của St mọi bit trước của thanh ghi được dời về bên trái, bit cao nhất của

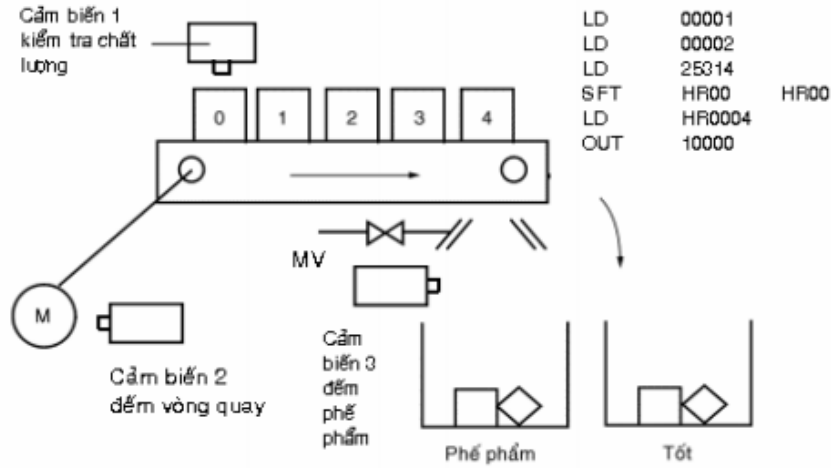
E mất đi. St, E: IR, SR, AR, HR, LR

Ví dụ: Chương trình phát hiện và loại bỏ phế phẩm.

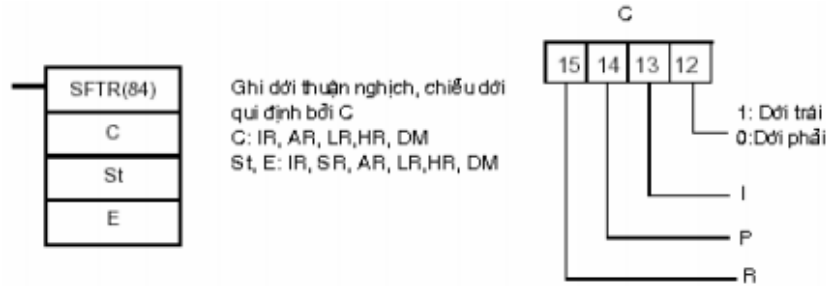
Cảm biến 1: Sẽ phát tín hiệu báo khi phát hiện phế phẩm và đưa vào thanh ghi (tín hiệu I).

Cảm biến 2: Phát một xung mỗi khi có một sản phẩm mới vào băng chuyền dùng làm xung nhịp cho thanh ghi (tín hiệu P).

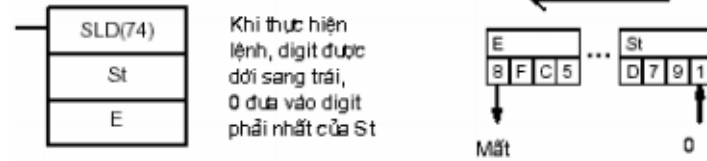
Khi phế phẩm đến vị trí số 3 (4 xung kể từ khi cảm biến 1 báo) sẽ được đẩy vào thùng chứa phế phẩm bởi van từ MV.



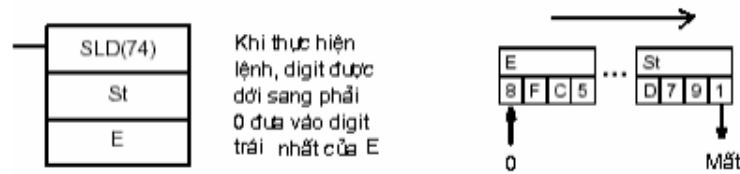
4.9.2. Dời thuận nghịch.



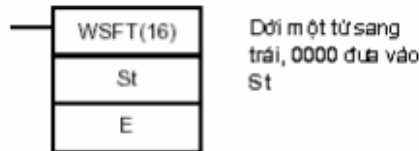
4.9.3. Dời digit sang trái.



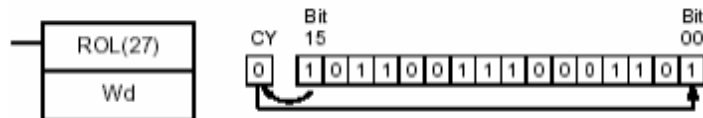
4.9.4. Dời digit sang phải.



4.9.5. Dời từ.

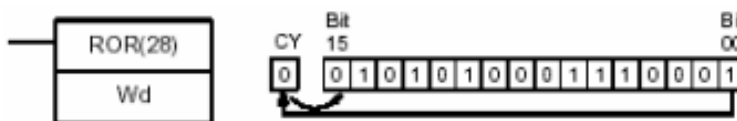


4.9.6. Quay trái.





4.9.7. Quay phải.



4.10. Chức năng ngắt (CQM1).

Chức năng ngắt cho phép PCL nhảy đến chương trình phục vụ ngắt khi có yêu cầu mà không cần phải thường xuyên đọc trạng thái đầu vào.

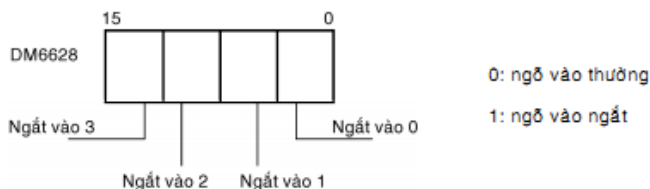
Có ba loại ngắt:

- Ngắt đầu vào.
- Ngắt thời gian.
- Ngắt bộ đếm vận tốc cao.

Ngắt đầu vào: Có 4 đầu vào ngắt theo ưu tiên sau:

- Ngắt vào 0 (IR00000) > ngắt vào 1 (IR00001) > ngắt vào 2 (IR00002) > ngắt vào 3 (IR00003).

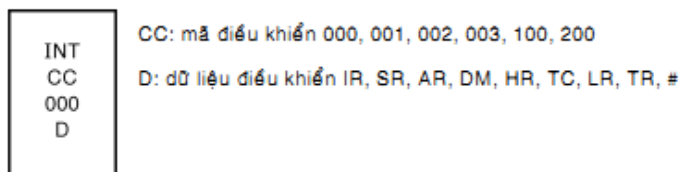
Các ngắt này gọi các hàm con theo thứ tự SBN 000 ÷ SBN 003. Muốn sử dụng ngắt đầu vào phải đặt nội dung DM 6628.



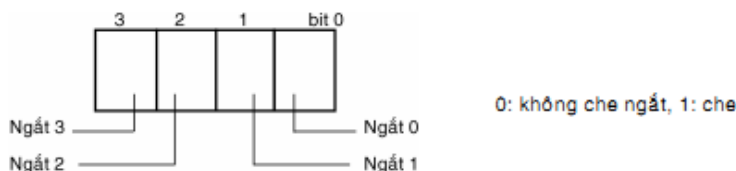
Ví dụ: Muốn dùng đầu vào IR00000, IR00001 làm đầu vào ngắt thì đặt (DM6628)=0011.

Ngắt có thể che hay không che với lệnh INT (89).

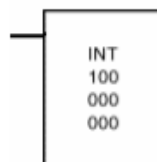
Lệnh này có dạng tổng quát.



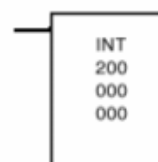
Muốn che ngắt hay không che dùng CC = 000 và D có dạng sau:



Khi bị che, tác động của đầu vào ngắt được ghi lại nhưng không thực hiện, khi xóa che sẽ lập tức nhảy đến chương trình con phục vụ ngắt, trừ khi nó được xóa ngắt bởi CC = 001 và bit tương ứng của D là 1.



Che mọi ngắt



Không che mọi ngắt

Độc trạng thái che hay không che với CC = 002, bit tương ứng trong D sẽ ON nếu bị che.

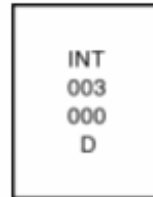
Có thể đặt đầu vào ngắt ở chế độ đếm, nghĩa là ngắt chỉ xảy ra sau khi có một số lượng xung ở đầu vào ngắt. Số lượng xung đặt trước ở các địa chỉ sau:

| | | | |
|--------|--------|--------|--------|
| Ngắt 0 | SR 244 | Ngắt 2 | SR 246 |
| Ngắt 1 | SR 245 | Ngắt 3 | SR 247 |

Nếu nội dung các ô nhớ trên là 0 thì ở chế độ ngắt thường, nội dung ô nhớ phải từ 0001 đến FFFF để ở chế độ ngắt đếm. Tần số xung đếm tối đa 1 KHz.

Sau khi đặt giá trị cho các ô nhớ trên dùng lệnh INT để cho phép ngắt đếm hoạt động.

Nếu bit tương ứng trong D là 0 thì hoạt động ở chế độ đếm và cho phép ngắt, nếu là 1 thì không tác động. Khi có một tín hiệu ngắt vào bộ đếm sẽ tăng lên 1 và khi bằng trị đặt thì gây ra ngắt.



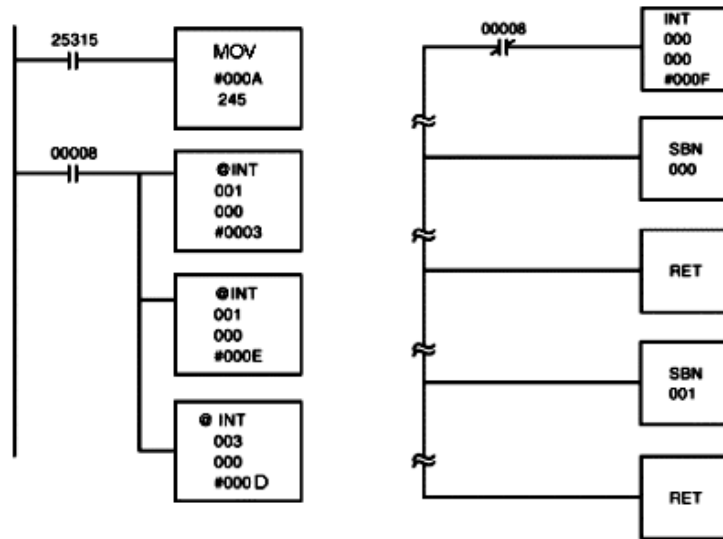
Bộ đếm sử dụng các ô nhớ sau:

| | | | |
|--------|--------|--------|--------|
| Ngắt 0 | SR 248 | Ngắt 2 | SR 250 |
| Ngắt 1 | SR 249 | Ngắt 3 | SR 251 |

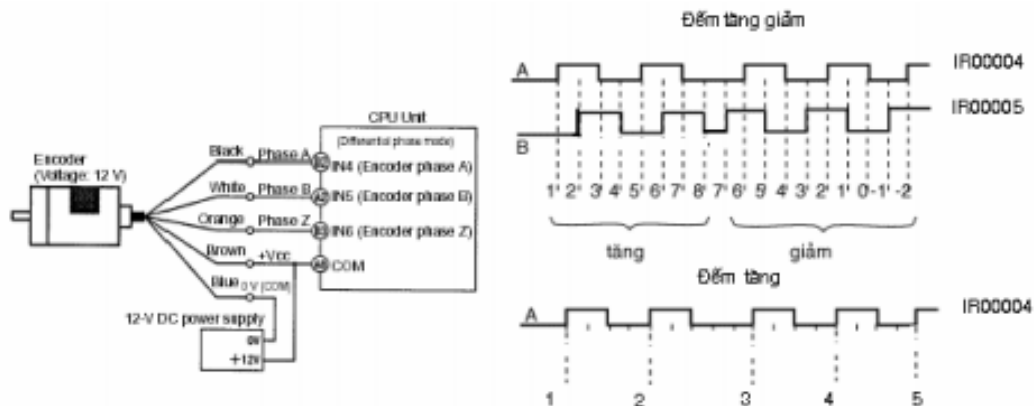
Nội dung ô nhớ là nội dung bộ đếm trừ đi 1.

Ví dụ: Dùng ngắt 0 chế độ ngắt đầu vào và ngắt 1 chế độ ngắt đếm. Đặt DM 6628:0011

Lập trình cho PLC như sau:



Bộ đếm vận tốc cao (HSC High Speed Counter) được dùng để đếm số xung từ encoder ra số theo chế độ ngắt. PLC có thể có nhiều HSC, HSC0 được lắp sẵn trên PLC. Dùng bo mạch mở rộng để thêm HSC.

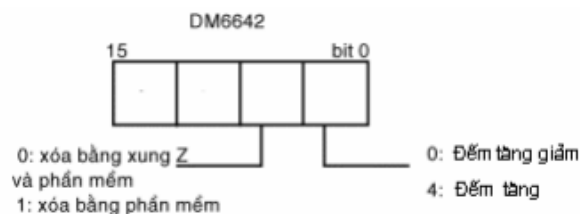




Sử dụng HSC0 ở chế độ đếm tăng hay đếm giảm, khi dùng đếm tăng giảm tần số xung đếm 4 lần so với xung thực tế, điều này tăng độ phân giải của encoder, xung A, B và Z của encoder đưa vào các đầu IR 00004 ÷ IR 00006. Tần số xung vào khi đếm tăng giảm là 2.5Khz và 5 Khz khi đếm tăng.

| | |
|-----------------------------------|---------------------|
| SR 231 | SR 230 |
| <i>Đếm tăng giảm</i> | <i>Đếm tăng</i> |
| F 0032768 + 00032767 (- 32767) | 00000000 + 00065535 |

HSC0 được khởi động bằng cách đặt cấu hình:



Muốn xóa bộ đếm dùng một trong hai phương pháp:

- Xóa phần mềm: cho SR 25200 ON.
- Xóa bằng phần mềm và xung Z: khi xung Z ON và SR 25200 ON, HSC0 được xóa khi cấp nguồn hay khi bắt đầu hoạt động.

Muốn đọc nội dung PV của bộ đếm, ta đọc nội dung hai ô nhớ SR 231, SR 230 hoặc dùng lệnh PRV 000 000 P1, nội dung SR231 và SR230 sẽ chứa vào P1+1 và P1.

Gọi chương trình phục vụ ngắt bằng lệnh so sánh bảng CTBL 000 C TB.

C= 000: so sánh HSC0 với các giá trị BCD 8 digit ghi trong bảng TB, nếu bằng thì gọi một trong các chương trình con (tối đa 16 giá trị) số 0 đến 255

C= 001 so sánh HSC0 với tối đa 8 khoảng, mỗi khoảng có giới hạn dưới và trên

TB: địa chỉ đầu của bảng

| TB | Số giá trị |
|------|------------------------------------|
| TB+1 | Giá trị so sánh số 1, 4 digit thấp |
| TB+2 | Giá trị so sánh số 1, 4 digit cao |
| TB+3 | Số chương trình con |
| TB+4 | Tương tự cho giá trị số 2 |

| TB | Giới hạn dưới số 1, 4 digit thấp |
|------|----------------------------------|
| TB+2 | Giới hạn dưới số 1, 4 digit cao |
| TB+3 | Giới hạn trên số 1, 4 digit thấp |
| TB+4 | Giới hạn trên số 1, 4 digit cao |
| TB+5 | Số chương trình con |
| TB+6 | Tương tự cho tầm số 2 |

Ví dụ: So sánh HSC0 với 1000 và 2000, gọi các chương trình con 101 và 102.

Ta đặt nội dung các ô nhớ.

| | |
|--------------|--------------|
| DM6642: 0114 | DM0003: 0101 |
| DM0000: 0002 | DM0004: 2000 |
| DM0001: 1000 | DM0005: 0000 |
| DM0002: 0000 | DM0006: 0102 |

Và viết chương trình

```
LD 25315
CTBL 000 000 DM0000
---
SBN 101
--
RET
SBN 102
...
RET
```




Ngắt thời gian: Có 3 ngắt thời gian 0, 1 và 2 được điều khiển nhờ lệnh STIM với hai chế độ hoạt động:

- Ngắt đơn ổn: Gọi chương trình con một thời gian sau khi STIM hoạt động.
- Ngắt chu kỳ: Gọi chương trình con theo chu kỳ cách nhau một khoảng thời gian.

Lệnh STIM (69) có dạng sau: STIM C1 C2 C3: Từ điều khiển C1 dùng để chọn chế độ khởi động, ngừng và đọc giá trị của Timer.

| Chức năng | Timer | C ₁ |
|-------------------------|-------|----------------|
| Khởi động chế độ đơn ổn | 0 | 000 |
| | 1 | 001 |
| | 2 | 002 |
| Khởi động chế độ chu kỳ | 0 | 003 |
| | 1 | 004 |
| | 2 | 005 |
| Ngừng timer | 0 | 010 |
| | 1 | 011 |
| | 2 | 012 |
| Đọc trị PV của timer | 0 | 006 |
| | 1 | 007 |
| | 2 | 008 |

Chế độ đơn ổn: C1 đặt như trên, C2+1 chứa đơn vị thời gian từ 0005 đến 0320 (0.5ms đến 32ms) và C2 chứa số lần đếm của đơn vị thời gian.

Như vậy khoảng thời gian từ lúc STIM thực hiện đến khi gọi chương trình con ngắt là: $(C2) \cdot (C2+1) \cdot 0.1ms = 0.5ms$ đến $319.968 ms$.

Nếu C2 là hằng số thì thời gian timer là số đó x 1ms.

C3: Từ 0000 đến 0255 là số chương trình con phục vụ.

Chế độ chu kỳ: C2; C2+1; C3: Như chế độ đơn ổn.

Chế độ ngừng: C2 = 000; C3 = 000.

Khi chương trình con được gọi, timer tự reset và hoạt động trở lại.

Chế độ đọc thời gian hiện tại của timer:

C2: Chỉ số lần bộ đếm đơn vị thời gian đã giảm.

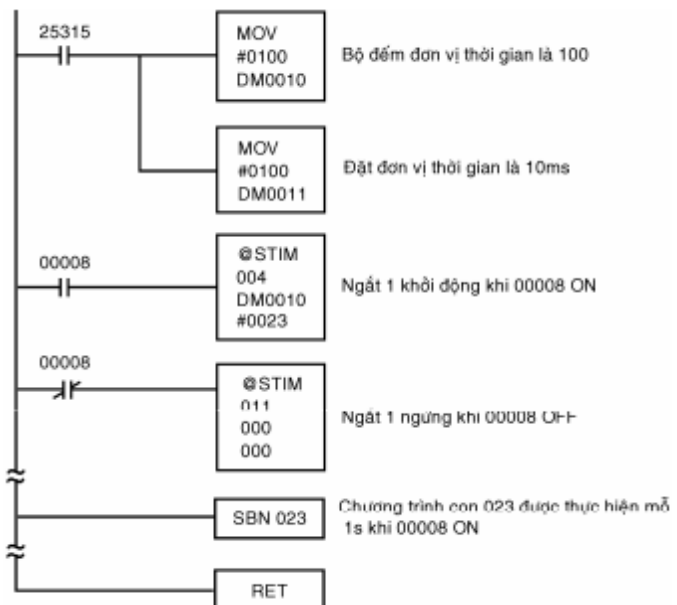
C2+1: Chứa khoảng thời gian trong đơn vị thời gian.

C3: Chứa địa chỉ ô nhớ nhận thông số thời gian đã trôi qua từ lần giảm trước.

Thời gian tổng cộng là: $[(C2) \cdot (C2+1) + C3] \cdot 0.1ms$.

Ngắt timer2 không dùng khi HSC0 đã dùng. Ngắt timer 0 không dùng khi sử dụng SPED phát xung.

Ví dụ: Dùng ngắt thời gian 1 theo kiểu chu kỳ cứ 1s gọi chương trình con số 23 một lần.



4.11. Xử lý Analog.

Có các module AD DA và các lệnh dành riêng cho việc điều khiển.

Khối nhập analog 12 bit CQM1 - AD041.

Module này có 4 đầu vào điện áp hay dòng điện chiếm 4 ô nhớ, có địa chỉ n đến n+3, n tùy thuộc vị trí gắn module và loại PLC.

Điện áp vào tối đa $\pm 15V$.

Dòng vào tối đa $\pm 30mA$.

Chọn chế độ nhờ DIP Switch.

Thời gian chuyển đổi 2.5ms/kênh.

Độ chính xác 1%.

Khi chọn tầm $\pm 10V$ trong ô nhớ sẽ chứa số F830 ÷ 07D0H (-2000 ÷ +2000)

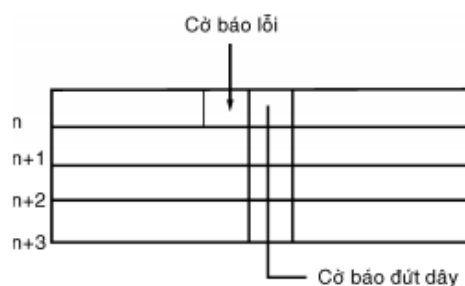
0V ÷ +10V: Tầm đổi 0030 ÷ 0FD0H (0048 ÷ 4048).

1V ÷ 5V hay 4mA ÷ 20mA số đổi là 0030 ÷ 0FD0.

Tổng trở nhận áp 1M Ω max, dòng 250 Ω .

Trường hợp đặt ở chế độ trung bình sẽ lấy 8 trị số đổi rồi lấy trung bình, chu kỳ lấy trung bình ~ 72ms.

Trường hợp đặt tầm đổi 1V ÷ 5V (4mA ÷ 20mA) khi tín hiệu vào < 0.95V (hay dòng nhỏ hơn 4mA) sẽ báo đứt dây ở bit 12 của mỗi ô nhớ. Khi có lỗi thì báo bằng bit 13 của từ nhớ đầu.



Khối xuất analog CQM1 – DA021.

Module có hai đầu ra áp và hai đầu ra dòng vi sai, thời gian đổi 0.5ms/2 điểm. Đổi 11 bit ra điện áp hay dòng, mỗi đầu ra chiếm một ô nhớ.

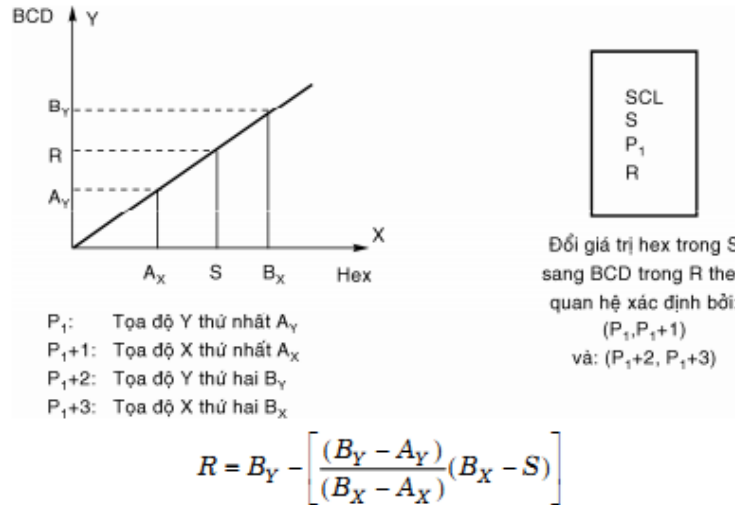
0000 ÷ 07FF → 0V ÷ 10V 0mA ÷ 20mA

F800 ÷ 07FF → -10V ÷ + 10V

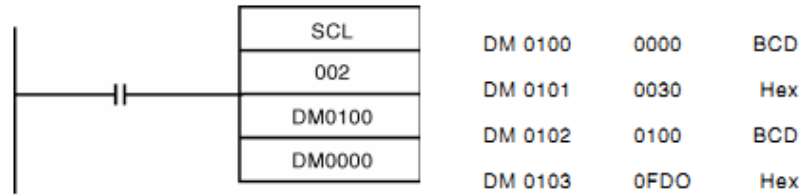
Các lệnh liên quan đến tín hiệu analog và điều khiển quá trình:

| | | | |
|--------------------|----------|---------------------------|---------|
| Tỷ lệ: | SCL (66) | Tạo hàm: | APR (-) |
| Tìm tối đa: | MAX (-) | Điều khiển PID: | PID |
| Tìm tối thiểu: | MIN (-) | Tạo xung: | PULS |
| Trị trung bình | AVG (-) | Tạo xung | SPED |
| Lấy tổng | SUM (-) | Tạo xung tần số thay đổi | PLS2 |
| Điều khiển gia tốc | ACL | Tạo xung bề rộng thay đổi | PWM |

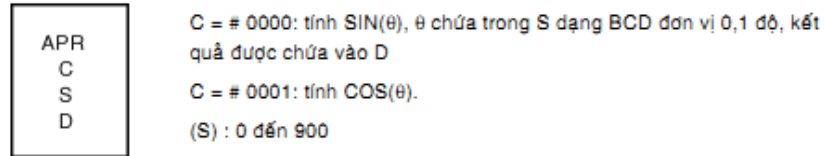
Tỷ lệ SCL: Đổi số nhị phân 4 bit digit sang số BCD 4 digit với tỷ lệ khác.



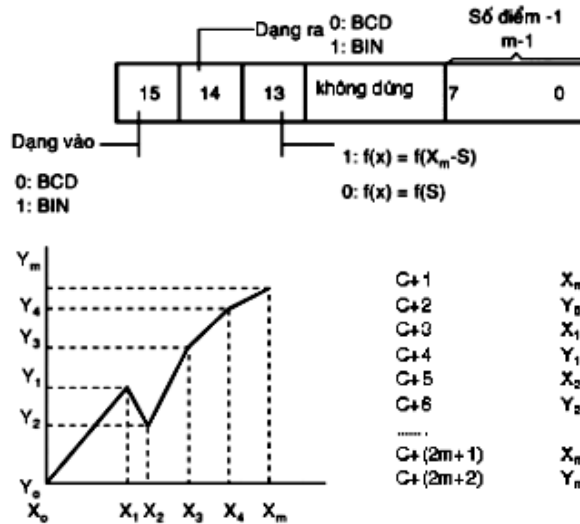
Ví dụ: Đổi số hex đọc từ khối analog in địa chỉ 002 tầm 0030 ÷ 0FD0H sang 0000 ÷ 0100BCD.



Tạo hàm APR (Arithmetic Process).

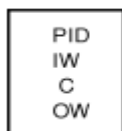


Nếu C là địa chỉ, ARP tính hàm nội suy $f(x)$, f chứa trong địa chỉ bắt đầu từ C, x chứa trong S. Hàm $f(x)$ là hàm tuyến tính từng đoạn và biểu thị bằng đồ thị, ghi trong bảng từ C+1 đến C+2m+2, C xác định số đoạn, dạng dữ liệu vào và ra. BCD hay BIN.



Điều khiển PID.

Hàm PID dùng thông số đặt trong C đến C+6 để tính OW dựa theo IW và SV:



IW: trị đo, nhị phân
 OW: tín hiệu ra khỏi điều khiển PID, nhị phân
 C: địa chỉ đầu bảng thông số

C: Trị đặt SV, nhị phân.

C+1: Dải tỷ lệ P $1 \div 9999$ ứng với dải tỷ lệ 0.1% ... 999.9%.

C+2: Hệ số tích phân BCD $T_{IK} = T_{I\gamma}$

C+3: Hệ số vi phân BCD $T_{DK} = T_{D/\gamma}$.

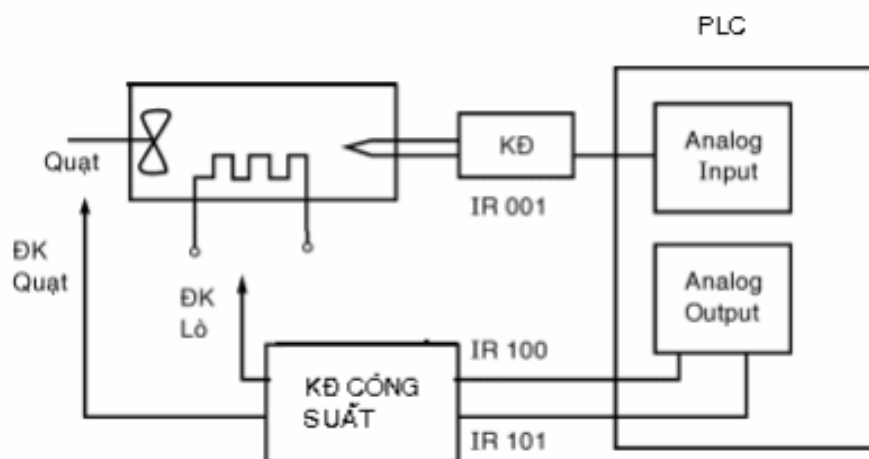
C+4: Chu kỳ lấy mẫu γ từ 00.01s đến 99.99s.

C+5: Bit 4 ÷ Bit 15 thông số lọc thường chọn là 0.65 (000 BCD), bit 0 ÷ 3: 0 – PID ngược, 1 – PID thuận.

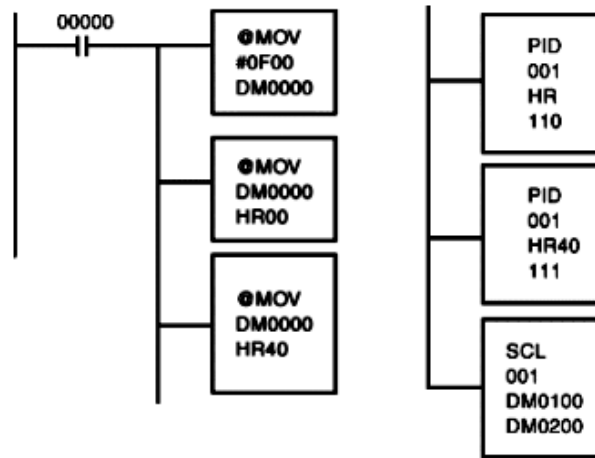
C+6: Bit 0 ÷ 3: Số bit của biến ra, giá trị 0 ... 8 ứng với số bit 8 ... 16 bit; bit 4 ÷ 7: đơn vị thời gian của thời gian lấy mẫu, 0: đơn vị 100ms, 1: đơn vị 10ms; Bit 8 ÷ 11: tầm IW (như OW).

Các ô nhớ từ C + 7 đến C+32 phải để trống.

Ví dụ: Điều khiển nhiệt độ dùng điện trở đốt và quạt thổi.



| Thông số điều khiển lò | | Thông số điều khiển quạt | | Thông số đổi BIN-BCD | |
|------------------------|-------------------------|--------------------------|--------|----------------------|------|
| HR00 | DM0000 (trị đặt) | HR40 | DM0000 | DM0000 | 0000 |
| HR01 | 0080 (P) | HR41 | 0060 | DM0101 | 0000 |
| HR02 | 0200 (T _i) | HR42 | 0150 | DM0102 | 0200 |
| HR03 | 0100 (T _d) | HR43 | 0000 | DM0103 | 0FFF |
| HR04 | 0001 (T _s) | HR44 | 0001 | | |
| HR05 | 0000 | HR45 | 0001 | | |
| HR06 | 0404 (T _{đm}) | HR46 | 0404 | | |



Phát xung.

Đôi khi cần phát xung tần số cao hay xung điều rộng ra ngoại vi để điều khiển động cơ bước hay điều khiển kiểu điều rộng xung. Dùng khối transistor ta có thể phát xung ở một trong các địa chỉ IR 100 đến IR 115. Ghi vào ô nhớ DM6615 từ 00xx, xx từ 00 đến 15 tùy theo muốn dùng địa chỉ IR nào.

Dùng lệnh PULS 000 000 P1 để ấn định số xung sẽ phát, là nội dung ô nhớ P1+1, P1 từ 00000001 đến 16777215.

Lệnh SPED D M F qui định cách phát xung, D = 000..150 chọn đầu ra 00...15 của từ đã qui định trong DM6615, M=000 là mode phát số lượng xung do lệnh PULS qui định, M=001 mode liên tục phát xung liên tục, F là tần số xung từ 0002...0100 nhân với 10Hz. Khi đang phát xung muốn đổi tần số ta thực hiện lệnh SPED với F thay đổi, nếu D = 000 thì ngừng phát xung.

4.12. Truyền thông.

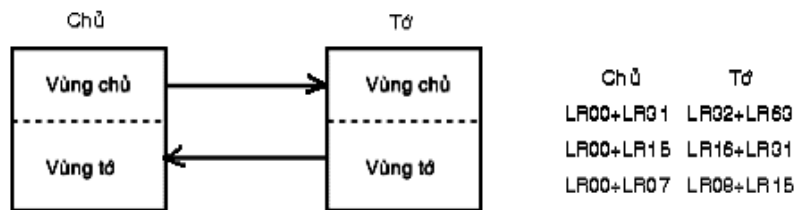
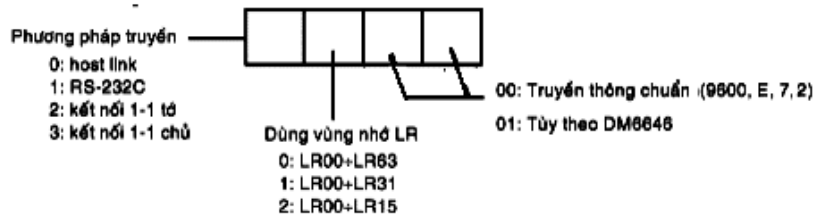
Có thể ghép nối PLC với nhau và máy tính với nhiều PLC thông qua các kết nối sau:

- Kết nối 1 - 1 cho phép nối hai PLC qua cáp nối RS-232.
- Host link nối một máy tính với một PLC qua cáp RS-232 hay một máy tính và nhiều PLC qua cáp 485.
- Controller link: Nối nhiều PLC với nhau qua hai dây (với module mạng).
- Ethernet: Nối nhiều máy tính và nhiều PLC (với module mạng) ...

Ngoài ra các module xuất nhập có thể nối đến PLC từ xa bằng cách dùng hai dây theo mạng Combo Bus S, Combo Bus D.

4.12.1. Kết nối 1 – 1:

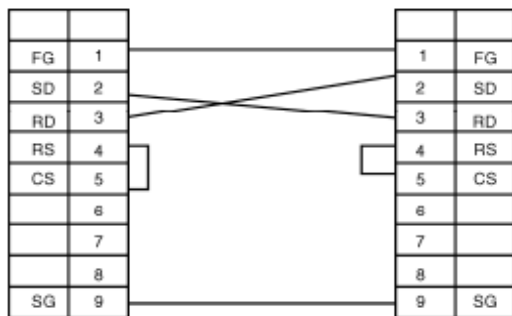
Hai PLC kết nối với nhau theo chế độ chủ - tớ, đặt cấu hình qua ô nhớ DM 6645



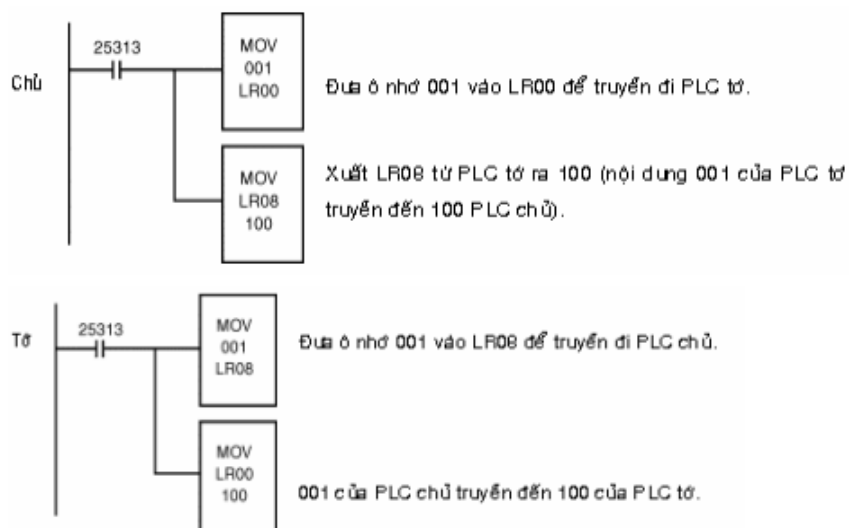


Ví dụ: Kết nối 2 PLC dùng vùng nhớ LR00 đến LR15.

Đặt DM6645: Chủ 3200
Tớ 2200

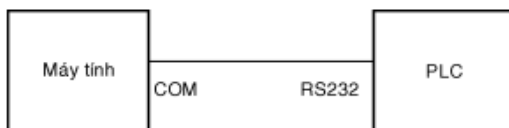


Sơ đồ đấu nối dây cáp 1 - 1

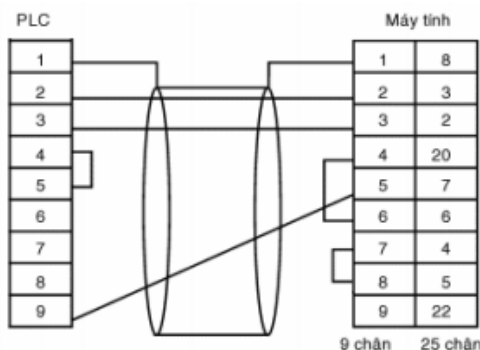


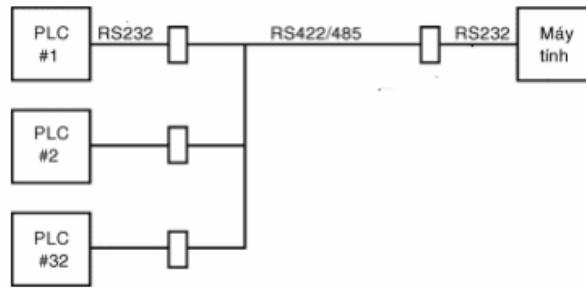
4.12.2. Host link.

Dùng để ghép nối máy tính với PLC qua cáp nối RS-232. Nếu muốn ghép một máy tính với nhiều PLC ta phải dùng bộ chuyển đổi RS-232 ↔ RS-485 cho phép ghép với tối đa 32 PLC.



Sơ đồ nối dây như sau:



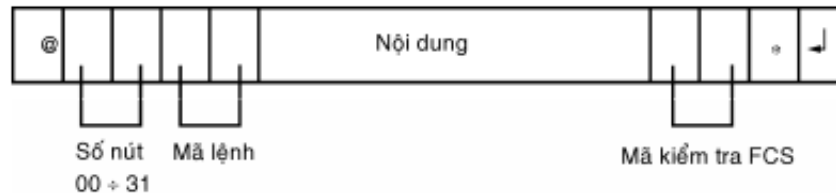


Thông qua host link có thể dùng máy tính để lập trình cho PLC hay đọc ghi bộ nhớ của PLC, từ PLC có thể truyền thông tin cho máy tính dùng lệnh TXD.

Đặt cấu hình dùng DM 6645 với cấu hình chuẩn là 0000.

Nếu dùng RS-422/485 thì mỗi PLC được đánh số nút từ 0000 đến 0031 trong DM6648.

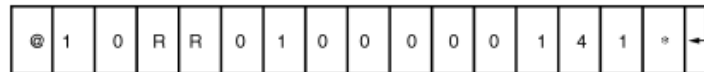
Máy tính truyền tin đến PLC theo dạng sau:



Một khối truyền dài tối đa 131 ký tự, nếu dài hơn 131 thì tách ra nhiều khối, mỗi khối kết thúc bằng \leftarrow (CHRS(13)). Khối cuối kết thúc bằng * \leftarrow .

FCS (Frame Check Sequence) là kết quả phép EXCLUSIVE OR các byte truyền từ đầu đến trước FCS và đổi thành hai ký tự ASCII. Khi nhận thông tin, máy tính hay PLC tính FCS rồi so sánh với FCS đã nhận.

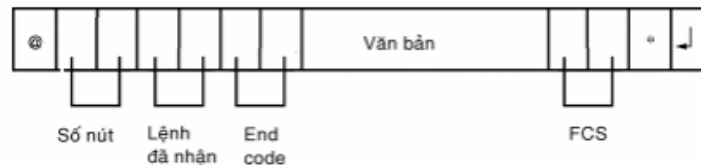
Ví dụ: Truyền lệnh đọc ô nhớ 0100 ở PLC số nút 10.



Tính:

| | | |
|------|-----|----------|
| EXOR | @ | 01000000 |
| | 1 | 00110001 |
| | 0 | 00110000 |
| | FCS | 01000001 |
| | | 4 1 |

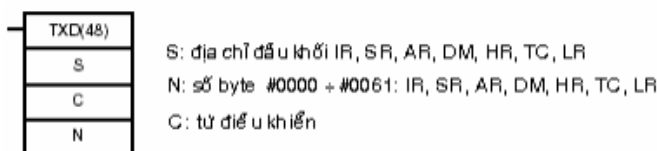
Khi nhận được thông tin từ máy tính, PLC tương ứng sẽ trả lời theo khối sau:



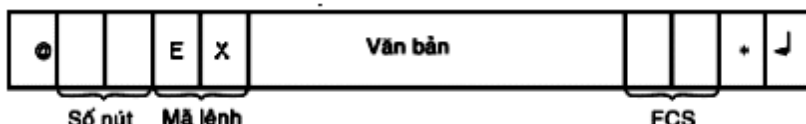
End code cho biết kết quả giao tiếp. Nếu giao tiếp đẹp thì End code là 00.

Bảng đầy đủ các lệnh truyền từ máy tính đến PLC có thể tham khảo thêm trong tài liệu chi tiết của PLC OMRON.

PLC CQM1 cũng có thể chủ động truyền thông tin cho máy tính dùng lệnh TXD.



TXD đổi các byte nhị phân từ S đến S+N/2-1 ra mã ASCII, mỗi byte nhị phân đổi thành hai byte ASCII và truyền theo chuẩn qui định bởi C. Nội dung của C thay đổi tùy theo cách thức truyền và công nối tiếp, trường hợp đơn giản nhất là C = #0000, byte cao nhất của S được truyền đầu tiên. Khi muốn truyền phải kiểm tra bit AR0805 (cờ báo truyền xong) là ON mới được truyền. Khi lệnh TXD được thực hiện sẽ truyền theo dạng sau:

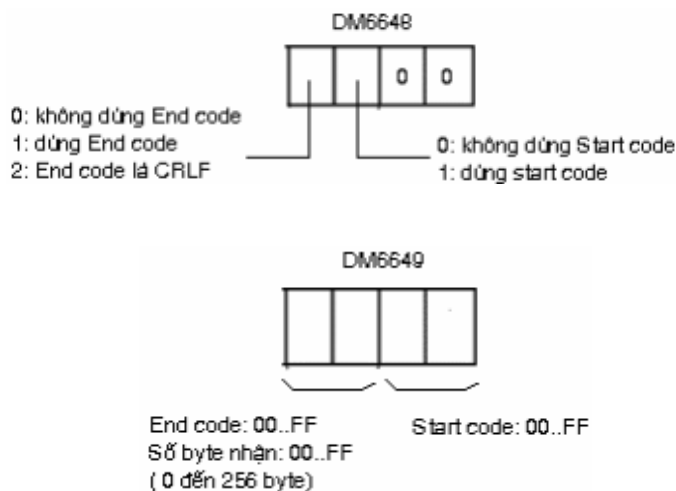


Ví dụ: Truyền 10 byte chứa trong DM0000 đến DM0004 theo chuẩn Host link dùng lệnh TXD DM0000 #0000 #0010 dữ liệu truyền đi (ASCII) là @00EX12341234123412341234123459*cr, giả sử các ô nhớ chứa số 1234. Máy tính phải có chương trình nhận dữ liệu.

Dùng TXD cho phép máy tính không cần thường xuyên đọc thông tin từ PLC mà PLC sẽ tự động truyền khi có nhu cầu. Nếu máy tính muốn trả lời thì truyền theo giao thức Host link như đã trình bày ở trên.

4.12.3. Truyền thông tự do.

Đặt ô nhớ DM6645 là 1000, dùng lệnh TXD để truyền và RXD để thu. Giao thức truyền do người dùng qui định bởi hai ô nhớ DM6648 và DM6649.

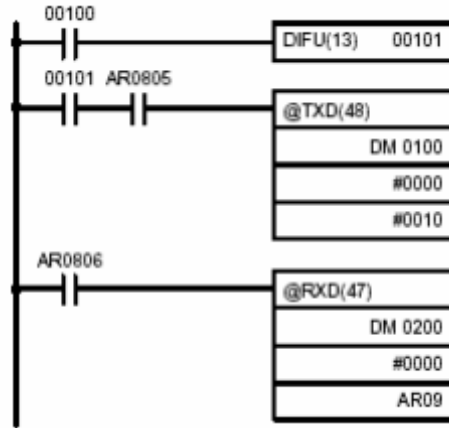


Lệnh TXD giống như trong phần Host link nhưng N có thể đến 0256, dữ liệu truyền đi được kèm thêm Start code, End code hay không tùy theo DM6848.

Máy tính truyền dữ liệu xuống PLC phải theo giao thức định bởi PLC. Khi PLC nhận dữ liệu xong, cờ thu AR0806 ON, tác động đến lệnh RXD, các byte ASCII được chuyển thành số nhị phân 0..F, thông tin về nhận dữ liệu chứa trong các ô nhớ sau:

| | |
|-------------------|---|
| AR 0800 ..AR 0803 | Mã sự cố cổng RS-232C BCD) 0: Thu bình thường, 1: Sai parity, 2: Sai Frame, 3: Tràn |
| AR 0804 | Sai truyền thông |
| AR0805 | Truyền xong |
| AR0905 | Thu xong |
| AR 0807 | Cờ tràn, dữ liệu mất vì không đọc kịp |
| AR 09 | Số byte đã nhận (BCD) |

Ví dụ: Truyền 10 byte trong bảng kể từ ô nhớ DM0100 và nhận dữ liệu cất vào bảng kể từ DM0200. Đặt DM6645 = 1000, DM6648 = 2000, không Start code, End code là CRLF



Cho bit SR25209 ON để reser cổng RS232.

Toàn bộ kiến thức cơ bản về PLC OMRON đã được trình bày trong chương 4. Để có thể nghiên cứu chuyên sâu hơn độc giả có thể tham khảo thêm các tài liệu chính thống của PLC OMRON.

Với lượng kiến thức đã trình bày ở trên cũng đủ để bạn đọc có thể sử dụng cơ bản bộ điều khiển lập trình được PLC OMRON ở trình độ cao đẳng dành cho sinh viên các trường đào tạo nghề.