

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

TRẦN THỊ THÚY HÀ

BÀI GIẢNG

**THIẾT BỊ NGOẠI VI
VÀ KỸ THUẬT GHÉP NỐI**

Hà nội, 2014

MỤC LỤC

MỤC LỤC	1
MỤC LỤC HÌNH VẼ	4
MỞ ĐẦU	7
CHƯƠNG 1. CƠ SỞ GHÉP NỐI THIẾT BỊ NGOẠI VI	8
1.1. Giới thiệu chung	8
1.1.1. Cấu trúc hệ thống.....	8
1.1.2. Chức năng của khối ghép nối	9
1.2. Thủ tục trao đổi dữ liệu.....	11
1.2.1. Chế độ trao đổi dữ liệu của máy tính.....	11
1.2.2. Các loại ngắt	12
1.2.3. Xử lý ngắt của nhiều thiết bị ngoại vi.....	16
1.2.4. Vi mạch xử lý ngắt.....	17
CÂU HỎI ÔN TẬP.....	19
CHƯƠNG 2. GHÉP NỐI QUA CỔNG SONG SONG.....	20
2.1. Ghép nối qua cổng song song.....	20
2.1.1. Giới thiệu chung về cổng song song.....	20
2.1.2. Các thanh ghi ở cổng song song	21
2.2. Các lệnh vào/ra dữ liệu	24
2.2.1. Các lệnh vào / ra bằng hợp ngữ	24
2.2.2. Các lệnh vào / ra bằng Turbo C	24
2.3. Khối ghép nối song song đơn giản	25
2.3.1. Cửa vào đơn giản.....	25
2.3.2. Cửa ra đơn giản.....	26
2.3.3. Cửa vào có đối thoại	27
2.3.4. Cửa ra đơn giản có đối thoại.....	29
2.4. Ghép nối song song điều khiển bằng chương trình	30
2.4.1. Sơ đồ khối và chức năng các khối của 8255A.....	30
2.4.2. Chế độ làm việc của 8255A.....	32
2.5. Ví dụ	43
CÂU HỎI ÔN TẬP.....	45
CHƯƠNG 3. GHÉP NỐI QUA CỔNG NỐI TIẾP	47
3.1. Giới thiệu chung về trao đổi dữ liệu nối tiếp	47
3.1.1. Nguồn gốc.....	47
3.1.2. Ưu, nhược điểm của giao diện nối tiếp RS232	48

3.1.3. Đặc điểm của chuẩn RS232	48
3.1.4. Thủ tục trao đổi dữ liệu.....	49
3.2. Một số chuẩn ghép nối.....	50
3.2.1. RS422.....	50
3.2.2. Chuẩn RS423A	51
3.2.3. Chuẩn RS485	51
3.2.4. So sánh các chuẩn ghép nối	52
3.3. Lập trình cho cổng RS-232.....	54
3.3.1. Bộ truyền nhận không đồng bộ vạn năng 8250	54
3.3.2. Các thanh ghi của UART 8250.....	56
3.3.3. Lập trình cho cổng RS-232.....	62
3.3.4. Giao tiếp Two-Wire I2C.....	66
3.3.5. Giao tiếp SPI.....	69
3.4. Cổng USB	71
3.4.1. Chuẩn USB	71
3.4.2. Đầu nối và cáp nối	71
3.4.3. Mô tả hệ thống.....	73
3.4.4. Cấu trúc thành phần	73
3.5. Đo và tạo thời gian chính xác bằng chip PIT – 8253/54	75
3.5.1. Giới thiệu chung	75
3.5.2. Cấu tạo của PIT – 8253/54	76
3.5.3. Lập trình cho 8253.....	77
3.5.4. Ghép nối PIT 8253/54	81
3.6. Ví dụ	81
CÂU HỎI ÔN TẬP.....	84
CHƯƠNG 4. GHÉP NỐI SỐ - TƯƠNG TỰ, TƯƠNG TỰ - SỐ	87
4.1. Tổng quan	87
4.2. Bộ biến đổi số-tương tự (DAC).....	87
4.3. Các tham số của bộ biến đổi D/A	88
4.4. Ghép nối bộ biến đổi D/A với máy tính	89
4.5. Lập trình cho DAC	90
4.6. Bộ biến đổi tương tự - số (ADC).....	91
4.6.1. Tham số đặc trưng của ADC	91
4.6.2. Bộ biến đổi A/D kiểu so sánh song song.....	92
4.6.3. Bộ biến đổi A/D theo phương pháp đếm	92
4.6.4. Bộ biến đổi A/D theo phương pháp xấp xỉ liên tiếp.....	93

4.6.5. Bộ biến đổi A/D hai sườn xung	94
4.6.6. Ghép nối các bộ biến đổi A/D với máy tính	95
4.6.7. Lập trình cho ADC	96
4.7. Ví dụ	97
CÂU HỎI ÔN TẬP.....	97
TÀI LIỆU THAM KHẢO	99

PDF

MỤC LỤC HÌNH VẼ

Hình 1. 1. Sơ đồ khối chức năng của một hệ máy tính.....	8
Hình 1. 2. Các chế độ trao đổi dữ liệu.....	12
Hình 1. 3. Phân loại các ngắt.....	13
Hình 1. 4. Sơ đồ chân của 8259A.....	18
Hình 1. 5. Sơ đồ khối của 8259A.....	19
Hình 2.1. Sơ đồ kết nối giữa máy tính và máy in.....	21
Hình 2.2. Sơ đồ định dạng của thanh ghi dữ liệu.....	22
Hình 2.3. Sơ đồ định dạng của thanh ghi trạng thái.....	22
Hình 2.4. Sơ đồ định dạng của thanh ghi điều khiển.....	23
Hình 2.5. Sơ đồ cửa vào đơn giản.....	26
Hình 2.6. Sơ đồ cửa ra đơn giản.....	27
Hình 2.7. Sơ đồ cửa vào có đối thoại thiết bị ngoại vi.....	28
Hình 2.8. Sơ đồ cửa ra có đối thoại thiết bị ngoại vi.....	29
Hình 2.9. Sơ đồ chân và sơ đồ kết nối của 8255A.....	30
Hình 2.10. Sơ đồ ghép nối của 8255A với thiết bị ngoại vi.....	32
Hình 2.11. Sơ đồ định dạng của từ điều khiển.....	32
Hình 2.12. Các chế độ hoạt động của thanh ghi điều khiển.....	33
Hình 2.13. Sơ đồ định dạng của từ điều khiển.....	34
Hình 2.14. Các cấu hình hoạt động của Mode 0.....	36
Hình 2.15. Sơ đồ minh họa các từ lệnh điều khiển dùng trong chế độ vào.....	38
Hình 2.16. Đồ thị thời gian của chế độ vào mô tả quá trình đọc.....	38
Hình 2.17. Sơ đồ minh họa các từ lệnh điều khiển dùng trong chế độ ra.....	39
Hình 2.18. Đồ thị thời gian của chế độ ra mô tả quá trình ghi.....	39
Hình 2.19. Sơ đồ minh họa các từ lệnh hoạt động trong mode 1.....	40
Hình 2.20. Sơ đồ minh họa các từ lệnh hoạt động trong mode 2.....	41
Hình 2.21. Sơ đồ minh họa các chế độ hoạt động của từ lệnh điều khiển.....	43
Hình 2.22. Hình vẽ ví dụ 1.....	44
Hình 3.1. Sơ đồ định dạng thanh ghi điều khiển đường truyền LCR.....	57
Hình 3.2. Sơ đồ định dạng thanh ghi trạng thái đường truyền LSR.....	59
Hình 3.3. Sơ đồ định dạng thanh ghi cho phép ngắt IER.....	60
Hình 3.4. Sơ đồ ghép nối với 8250.....	62
Hình 3.5. Sơ đồ chân của UART 8250.....	64
Hình 3.6. Sơ đồ khối của 8250.....	65
Hình 3.7. Một mạng TWI.....	67
Hình 3.8. Quá trình truyền 1 bit dữ liệu I2C.....	68
Hình 3.9. Lưu đồ thuật toán quá trình truyền nhận dữ liệu I2C.....	68
Hình 3.10. Giao diện SPI.....	70
Hình 3.11. Truyền nhận dữ liệu trên SPI.....	71

Hình 3.12. Cấu trúc cổng USB.	72
Hình 3.13. Sơ đồ tổng quát của hệ thống.....	73
Hình 3.14. Sơ đồ giao tiếp cổng USB với bộ điều khiển trung tâm (kit Arduino).....	75
Hình 3.15. Sơ đồ khối và sơ đồ chân của PIT 8253/54.....	76
Hình 3.16. Định dạng từ điều khiển.....	78
Hình 3.17. Ghép nối 8253/54 với PC qua khe cắm ISA.....	81
Hình 4.1. Sơ đồ cấu trúc hệ thống điều khiển tự động.....	87
Hình 4.2. Bộ biến đổi D/A cơ bản.....	88
Hình 4.3. Mạch biến đổi D/A kiểu R-2R.	88
Hình 4.4. Điều khiển DAC bằng máy tính.	90
Hình 4.5. Ghép nối vi xử lý với DAC 16 bit.....	90
Hình 4.6. Bộ biến đổi A/D song song.....	92
Hình 4.7. Bộ biến đổi A/D theo phương pháp đếm.....	93
Hình 4.8. Bộ biến đổi A/D theo phương pháp đếm.....	94
Hình 4.9. Bộ biến đổi A/D hai sườn xung.....	95
Hình 4.10. Sơ đồ ghép nối ADC với vi xử lý.....	96
Hình 4.11. Nối ghép ADC 804 với 8255.....	97

MỤC LỤC BẢNG

<i>Bảng 1. 1. Bảng chức năng các chân của thanh ghi trạng thái.</i>	22
<i>Bảng 1. 2. Bảng chức năng các chân của thanh ghi điều khiển.</i>	23
<i>Bảng 1. 3. Bảng chức năng của 8255A.</i>	Error! Bookmark not defined.
<i>Bảng 1. 4. Bảng điều khiển đọc/viết của 8255A.</i>	Error! Bookmark not defined.
<i>Bảng 2.1. Bảng chức năng các chân của thanh ghi trạng thái.</i>	22
<i>Bảng 2.2. Bảng chức năng các chân của thanh ghi điều khiển.</i>	23
<i>Bảng 2.3. Bảng các lệnh vào/ra.</i>	24
<i>Bảng 2.4. Bảng các lệnh vào/ra.</i>	24
<i>Bảng 2.5. Bảng chức năng của 8255A.</i>	31
<i>Bảng 2.6. Bảng điều khiển đọc/viết của 8255A.</i>	31
<i>Bảng 3.1. Bảng so sánh các chuẩn ghép nối.</i>	52
<i>Bảng 3.2. Bảng mô tả các thanh ghi của UART.</i>	56
<i>Bảng 3.3. Bảng mô tả chức năng của thanh ghi điều khiển MODEM.</i>	61
<i>Bảng 3.4. Bảng mô tả chức năng của thanh ghi trạng thái MODEM.</i>	61
<i>Bảng 3.5. Các lệnh của 8250.</i>	65
<i>Bảng 3.6. Sắp xếp chân ở cổng USB.</i>	72
<i>Bảng 3.7. Bảng chế độ hoạt động của 8253.</i>	77
<i>Bảng 3.8. Tóm tắt hoạt động của xung cửa GATE 8253/54.</i>	79
<i>Bảng 3.9. Định dạng của từ điều khiển.</i>	80
<i>Bảng 3.10. Định dạng cho lệnh đọc thanh ghi điều khiển.</i>	81
<i>Bảng 3.11. Định dạng byte trạng thái.</i>	81

MỞ ĐẦU

Công nghệ thông tin truyền thông và kỹ thuật máy tính có một thời gian dài xây dựng và phát triển. Hiện nay, máy tính được đưa vào ứng dụng trong mọi lĩnh vực khoa học kỹ thuật và xã hội. Máy tính ngày càng được phát triển cả về tính năng hoạt động, ứng dụng với kích thước ngày càng nhỏ.

Nghiên cứu, phát triển các hệ thống thiết bị ngoại vi và các kỹ thuật ghép nối máy tính với các thiết bị khác được triển khai trên tất cả các trường đại học công nghệ, kỹ thuật trên thế giới.

Các kỹ thuật ghép nối máy tính được phát triển từ các bước với các kỹ thuật và giao thức kết nối như UART, RS232, RS485, LPT, IEEE 1394, USB 1.0, USB 2.0, USB 3.0, VGA, DVI, HDMI, Wifi, bluetooth,... thông dụng của máy tính; các kết nối song song, nối tiếp, I2C, SPI, ADC, DAC, timer, interrupt... của các chip thông dụng hiện nay.

Giảng dạy lý thuyết, thực hành và định hướng các nghiên cứu tương lai cho các sinh viên học và sử dụng được các kiến thức học được về kỹ thuật ghép nối máy tính và các thiết bị ngoại vi do đó rất quan trọng và có nhu cầu lớn trong xã hội.

Môn học này và các môn học tương đương đang được giảng dạy rộng rãi trong các trường kỹ thuật công nghệ trong nước như: Đại học Bách khoa Hà Nội, Đại học Công nghệ, Đại học Quốc gia Hà nội và nhiều trường khác. Các sinh viên của các trường được giảng dạy lý thuyết kết hợp với các bài thực hành dựng sẵn. Sau đó, sinh viên được làm quen với các bài tập lớn, bài tập lớn tự thiết kế, chế tạo các mạch đơn giản kết nối PC, MCU với các mạch điện, hệ thống điều khiển, thu thập dữ liệu và truyền tin cụ thể.

Môn học 'Thiết bị ngoại vi và kỹ thuật ghép nối' cũng được giảng dạy cho các sinh viên trên các trường đại học hàng đầu trên thế giới và khu vực như: MIT, UIUC, NUS, NTU...

CHƯƠNG 1. CƠ SỞ GHÉP NỐI THIẾT BỊ NGOẠI VI

1.1. Giới thiệu chung

1.1.1. Cấu trúc hệ thống

Máy vi tính có thành phần cơ bản là bộ vi xử lý, nó gồm có các khối chức năng như hình 1.1.

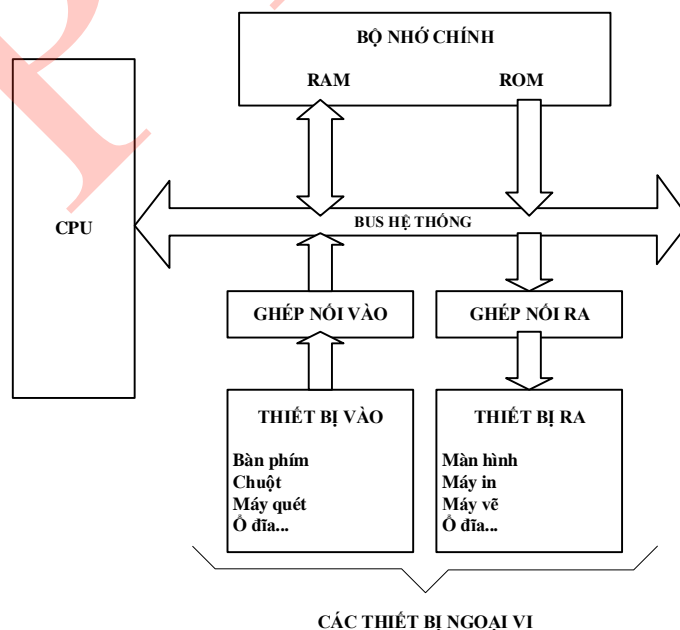
Khối xử lý trung tâm CPU (Central Processing Unit). Khối này được chế tạo bởi mạch vi điện tử có độ tích hợp rất cao.

Khối Bộ nhớ chính hay gọi là bộ nhớ trong gồm có các bộ nhớ bán dẫn ROM và RAM có tốc độ truy cập nhanh dùng để chứa các chương trình điều khiển hoạt động của hệ thống.

Khối mạch ghép nối vào / ra là các mạch điện tử cho phép CPU liên lạc được với các thiết bị ngoại vi như bàn phím, màn hình, ổ đĩa, chuột... hoặc các thiết bị ngoại vi như bộ biến đổi số - tương tự DAC, tương tự - số ADC, mạch vào/ra số I/O.

Khối đồng hồ hệ thống (system clock) là khối tạo ra xung clock để duy trì hoạt động và đồng bộ hoạt động của CPU với các bộ phận liên quan.

Ngoài ra, trong máy tính còn được cài sẵn hệ điều hành để điều hành các chức năng cơ bản của hệ thống máy tính.



Hình 1. 1. Sơ đồ khối chức năng của một hệ máy tính

1.1.2. Chức năng của khối ghép nối

Máy tính PC được xuất xưởng có thể được người sử dụng nâng cấp, mở rộng cấu hình bằng cách ghép nối thêm các card mở rộng hoặc các thiết bị ngoại vi. Các nhà sản xuất máy tính đã dự trữ sẵn các rãnh cắm mở rộng trên bản mạch chính, các cổng ghép nối: song song (LPT), nối tiếp (COM).

Đây chính là những vị trí mà kỹ thuật ghép nối máy tính có thể tác động vào. Nhờ có kỹ thuật ghép nối máy tính mà khả năng của máy tính được mở rộng thêm rất nhiều: ví dụ có thể xây dựng được các hệ thống đo lường hoàn toàn tự động.

Để ghép nối máy tính với các thiết bị ngoại vi có bốn khả năng để lựa chọn:

- Ghép nối qua cổng máy in hay còn gọi là cổng song song.
- Ghép nối qua cổng RS 232 hay còn gọi là cổng nối tiếp.
- Ghép nối qua rãnh cắm mở rộng trên bản mạch chính.
- Ghép nối qua cổng USB.

Nhiệm vụ của khối ghép nối là phối hợp trao đổi tin giữa máy tính và thiết bị ngoại vi về mức và công suất của tín hiệu, về dạng tin, tốc độ và phương thức trao đổi.

Phối hợp về mức và công suất tín hiệu:

- Mức tín hiệu của máy tính thường là mức (0V, 5V) trong khi đó thiết bị ngoại vi có thể là $\pm 15V$, $\pm 48V$.
- Công suất của máy tính thường nhỏ (cỡ chục mA), các thiết bị ngoại vi cần có công suất lớn hơn. Khối ghép nối phải biến đổi công suất để phù hợp (khuếch đại đường dây, khuếch đại công suất).

Phối hợp về dạng tin:

- Tín tức trao đổi của máy tính luôn là song song, có thể truyền theo 8 bit, 16 bit, 32 bit, 64 bit.
- Tín của thiết bị ngoại vi có thể là song song hoặc nối tiếp. Khi trao đổi song song thường là 8 bit và 16 bit.

Phối hợp về tốc độ trao đổi thông tin:

- Máy tính thường hoạt động với tốc độ cao (cỡ 100MHz), thiết bị ngoại vi thường hoạt động chậm hơn nên khối ghép nối nhận tin nhanh từ máy tính rồi truyền tin chậm cho thiết bị ngoại vi để giải phóng máy tính thực thi nhiệm vụ khác.

Phối hợp về phương thức trao đổi tin:

- Khối ghép nối phải có cách trao đổi tin diễn ra theo trình tự nhất định:
- Nếu việc trao đổi tin do máy tính khởi xướng:
 - Máy tính đưa lệnh điều khiển để khởi động thiết bị ngoại vi hay khởi động khối ghép nối.
 - Máy tính đọc trả lời sẵn sàng trao đổi hay trạng thái sẵn sàng của thiết bị ngoại vi. Nếu có trạng thái sẵn sàng mới trao đổi tin, nếu không thì chờ và đọc lại trạng thái.
 - Máy tính trao đổi tin khi đọc thấy trạng thái sẵn sàng
- Nếu việc trao đổi tin do thiết bị ngoại vi khởi xướng:
 - Thiết bị ngoại vi đưa yêu cầu trao đổi tin vào bộ phận xử lý ngắt của khối ghép nối, để đưa ra yêu cầu ngắt chương trình cho máy tính.
 - Nếu có nhiều thiết bị ngoại vi yêu cầu đồng thời thì khối ghép nối sắp xếp theo ưu tiên định sẵn, rồi đưa ra yêu cầu trao đổi tin cho máy tính.
 - Máy tính nhận yêu cầu, và đưa tín hiệu xác nhận sẵn sàng trao đổi.
 - Khối ghép nối nhận và truyền tín hiệu xác nhận cho thiết bị ngoại vi.
 - Thiết bị ngoại vi trao đổi tin với khối ghép nối và khối ghép nối trao đổi tin với máy tính (nếu đưa tin vào) và ngược lại nếu lấy tin ra.

Chức năng của khối ghép nối:

- Nhận tín hiệu (listener): Nhận thông báo địa chỉ từ máy tính; Nhận thông báo về trạng thái từ thiết bị ngoại vi; Nhận lệnh điều khiển từ máy tính.
- Nguồn tín hiệu (talker): Phát địa chỉ cho khối chức năng của thiết bị ngoại vi; Phát lệnh cho thiết bị ngoại vi; Phát yêu cầu hay trạng thái của thiết bị ngoại vi cho máy tính; Phát số liệu cho thiết bị ngoại vi hay cho máy tính.

Chức năng điều khiển (controller):

- Khối ghép nối có cả 2 nhiệm vụ nguồn nhận và nguồn phát lệnh: Phát địa chỉ cho từng khối chức năng của thiết bị ngoại vi; Truyền lệnh cho từng khối; Nhận lệnh từ khối điều khiển khác; Nhận yêu cầu trao đổi tin ở các khối chức năng...

Chương trình phục vụ trao đổi tin cho khối ghép nối:

- Mỗi khối ghép nối cần viết chương trình trao đổi tin (ngôn ngữ Assembly) và khi sử dụng người dùng cần viết chương trình ứng dụng (ngôn ngữ Turbo Pascal hay C++).
- Với chương trình phục vụ trao đổi tin, có các thao tác:
 - Khởi động khối ghép nối: ghi các lệnh xác định chế độ (mode) với lệnh điều khiển khối ghép nối và thiết bị ngoại vi.
 - Ghi che chắn và cho phép ngắt.
 - Đọc trạng thái thiết bị ngoại vi (bằng lệnh đọc - IN) và xử lý ngắt theo cách hỏi vòng (polling) hoặc xử lý ngắt bằng mạch điện tử (phần cứng).
 - Ghi số liệu ra: Từ thanh ghi A của vi xử lý đưa tin về số liệu ($D_0 \div D_n$) bằng lệnh viết \overline{WR} hay đưa ra (OUT) thanh ghi đệm viết.
 - Đọc tin số liệu ($D_0 \div D_n$) bằng lệnh đọc \overline{RD} hay đưa vào (IN) thanh ghi chứa A.

1.2. Thủ tục trao đổi dữ liệu

1.2.1. Chế độ trao đổi dữ liệu của máy tính

Quá trình trao đổi dữ liệu giữa CPU và các ngoại vi gồm có các bước:

- Xác định xem có thiết bị nào sẵn sàng trao đổi thông tin với CPU hay không?
- Nếu có thì xác định xem đó là thiết bị nào?
- Giải quyết tranh chấp nếu có nhiều thiết bị cùng muốn trao đổi thông tin với CPU.
- Phát ra các tín hiệu điều khiển và đồng bộ để trao đổi dữ liệu

Có ba phương pháp trao đổi dữ liệu:

- Đồng bộ hay không điều kiện (không hội thoại).
- Không đồng bộ hay có điều kiện sẵn sàng của ngoại vi (có hội thoại).
- Ngắt chương trình.

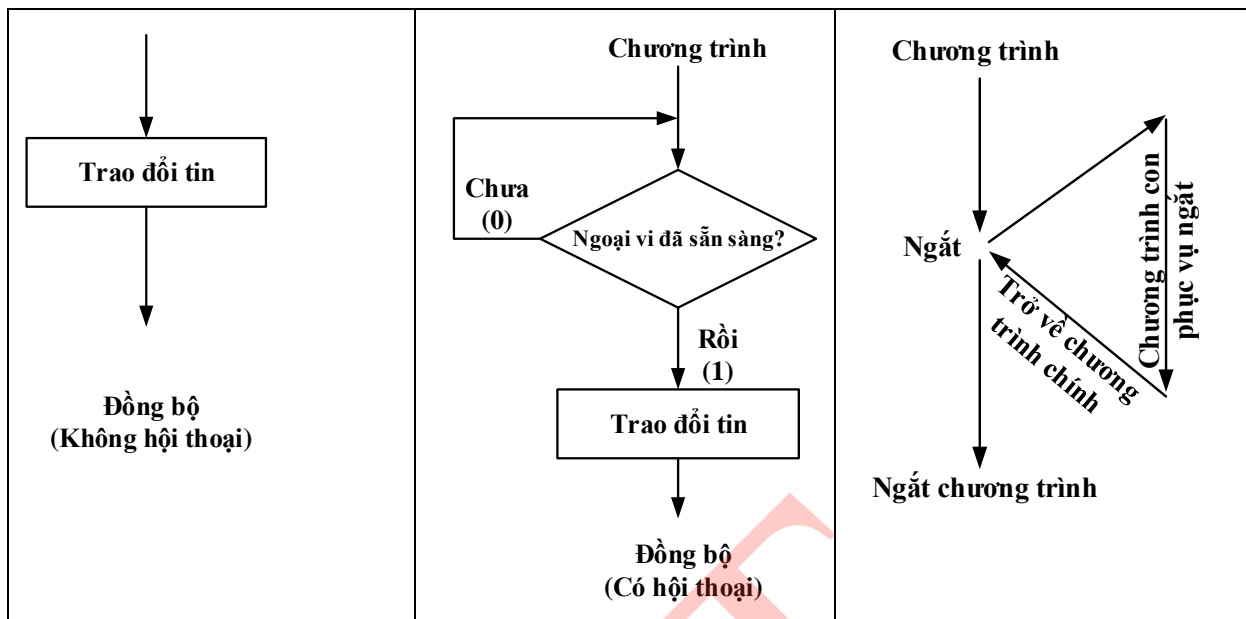
Trao đổi đồng bộ: máy tính không quan tâm xem ngoại vi có sẵn sàng trao đổi tin hay không mà đưa luôn các lệnh trao đổi tin (đọc, ghi hay chuyển số liệu).

Phương pháp trao đổi này chỉ được thực hiện khi:

- Ngoại vi luôn sẵn sàng trao đổi tin.
- Tốc độ trao đổi tin giữa máy tính và ngoại vi luôn phù hợp nhau.

Ưu điểm: Trao đổi theo phương pháp này nhanh, không tốn thời gian chờ.

Nhược điểm: Mặc dù phương thức trao đổi nhanh nhưng thiếu tin cậy, có khả năng bị mất tin vì có thể có sự cố làm cho ngoại vi chưa sẵn sàng trao đổi.



Hình 1. 2. Các chế độ trao đổi dữ liệu

Trao đổi đồng bộ hay hỏi trạng thái. Trong phương pháp này, chương trình liên tục kiểm tra trạng thái của các ngoại vi để xem có yêu cầu nào cần trao đổi dữ liệu không. Nếu có ngoại vi nào đó có yêu cầu phục vụ (trạng thái sẵn sàng - READY) thì chương trình sẽ cấp một chương trình con để phục vụ rồi tiếp tục kiểm tra ngoại vi kế tiếp. Quá trình này được thực hiện cho đến ngoại vi cuối cùng và sẽ được lặp lại.

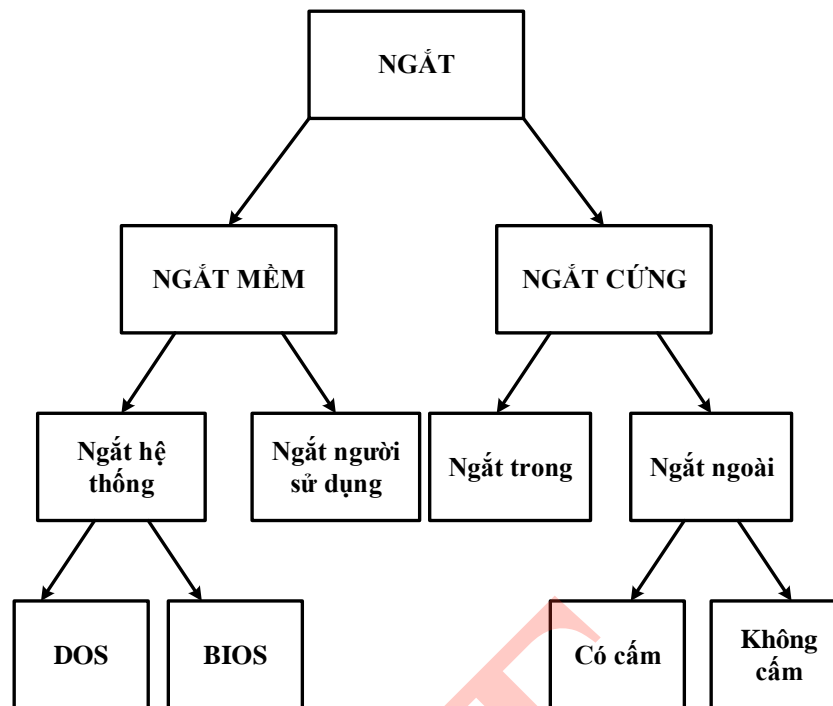
Ưu điểm : phương pháp này khá đơn giản, có độ tin cậy, chỉ trao đổi tin khi ngoại vi đã sẵn sàng.

Nhược điểm: tốn thời gian sử dụng máy tính khi số lượng ngoại vi tăng đồng thời khả năng đáp ứng tức thời với yêu cầu phục vụ rất thấp vì CPU phải kiểm tra tất cả các trạng thái của các ngoại vi trước khi quay về ngoại vi có yêu cầu phục vụ.

Trao đổi theo ngắt chương trình. Máy tính đang thực hiện chuỗi lệnh của chương trình nào đó, nếu ngoại vi có yêu cầu trao đổi tin, sẽ gửi tín hiệu yêu cầu ngắt (\overline{INTR}) tới CPU. CPU sẽ thực hiện nốt lệnh hiện tại và đưa tín hiệu chấp nhận (xác nhận ngắt-INTA). Chương trình chính bị ngắt, máy tính chuyển sang chương trình con phục vụ ngắt tức thì. Sau khi xong việc chương trình chính lại được tiếp tục thực hiện ở chỗ bị ngắt.

1.2.2. Các loại ngắt

Ngắt là quá trình CPU tự ngưng hoạt động hiện tại khi có một yêu cầu ngắt gửi đến để chuyển sang thực hiện một chương trình con phục vụ ngắt tương ứng. Sau khi thực hiện xong thì quay trở lại thực hiện tiếp công việc đang dở trên.



Hình 1. 3. Phân loại các ngắt.

Các loại ngắt có thể được phát ra từ một trong ba nguồn:

Nguồn thứ nhất là tín hiệu bên ngoài được đưa vào chân ngắt không thể che được NMI hoặc được đưa vào chân yêu cầu ngắt INTR. Ngắt xảy ra do tín hiệu được đưa vào một trong các chân nói trên được gọi là *ngắt cứng*.

Nguồn ngắt thứ hai là các câu lệnh gọi ngắt INT và được gọi là *ngắt mềm*.

Nguồn ngắt thứ ba là các lỗi phát sinh trong quá trình thực hiện câu lệnh (ví dụ chia một số cho 0). Các ngắt này xảy ra trong CPU được gọi là *ngoại lệ*.

Ngắt cứng: CPU có các lỗi dành cho ngắt ngoài. Khi có tín hiệu vào của lỗi vào này VXL sẽ dừng chương trình đang chạy.

Ví dụ: VXL Intel 8080 có 2 lỗi vào ngắt: INT và Reset.

VXL Intel 8085 có 5 lỗi vào ngắt: INTR, Reset, RST 6,5; RST 7,5; TRAP).

VXL IAP x86 có 3 lỗi vào ngắt: INTR, Reset, NMI.

Ngắt NMI (No Masquable Interrupt)- ngắt không che được:

Có 3 nguồn gây ra ngắt:

- Sai số chẵn/lẻ trong bộ nhớ.
- Lỗi ở cửa vào/ra.
- Bộ đồng xử lý báo lỗi.

Khi có ngắt này, VXL dừng chương trình sau lệnh đang thực hiện.

Thanh ghi chỉ thị lệnh (IP) và thanh ghi chỉ thị Flag được lưu giữ.

2 bit IF (Interuppt Flag) và TF (Trap Flag) bị xóa về 0 để cấm ngắt ngoài tiếp theo và không có bấy.

Muốn cho phép hay không cho phép ngắt này xảy ra dùng 1 trigơ mắc lỗi vào ngắt trước khi đưa tới lỗi ngắt NMI.

Ngắt INTR:

Ngắt này cho phép ngắt ($IF = 1$) hay cấm ngắt ($IF = 0$) vào thanh ghi Flag.

INTR được xóa về 0 nhờ lệnh CLI (Clear Interrupt).

INTR được xác lập lên 1 nhờ lệnh STI (Set Interrupt).

INTR thường được nối với lỗi ra yêu cầu ngắt của vi mạch xử lý ưu tiên ngắt.

Ngắt INTR có 16 mức ngắt với mức ưu tiên giảm dần.

Bảng 1.1. Các mức ngắt của ngắt INTR

IRQ 0	Lỗi ra bộ đếm thời gian	IRQ 8	Ngắt nhịp thời gian thực
IRQ 1	Bàn phím (bộ đếm ra)	IRQ 9	Phần mềm định hướng tới INT 0Ah (IRQ 2)
IRQ 2	Ngắt từ 8259A thứ 2	IRQ 10	Dành riêng (dự trữ)
IRQ 3	Cổng nối tiếp 2 không đồng bộ	IRQ 11	Dành riêng (dự trữ)
IRQ 4	Cổng nối tiếp 1 không đồng bộ	IRQ 12	Dành riêng (dự trữ)
IRQ 5	Cổng song song 2 trong máy AT	IRQ 13	Đồng xử lý
IRQ 6	Điều khiển đĩa mềm	IRQ 14	Điều khiển đĩa cứng
IRQ 7	Cổng song song 1 (Máy in)	IRQ 15	Dành riêng

Ngắt RESET.

Thực sự không phải là ngắt, chỉ là treo chương trình tạm thời.

Khi Reset được khởi động lại hoặc xóa về trạng thái ban đầu hay có lệnh Reset của chương trình → các thanh ghi của $VXL = 0$, trừ thanh ghi đoạn mã (CS) được nạp địa chỉ đầu tiên của chương trình tự kiểm tra Port.

Ngắt mềm: Ngắt bên trong do lệnh của chương trình

Khi VXL gặp các lệnh gây ra ngắt hoặc tình huống đặc biệt khi thực hiện lệnh (ngắt logic) và ngắt của hệ điều hành.

Ngắt do lệnh: là ngắt khi thực hiện lệnh: Call, HLT, INT

Ngắt logic: xảy ra khi gặp tình huống đặc biệt sau:

Chia một số cho 0 (vecto 0).

Vượt quá nội dung thanh ghi hay bộ nhớ (vecto 4).

Thực hiện từng bước (vecto 1).

Điểm dừng chương trình do người sử dụng định trước (vecto 3).

Ngắt của hệ điều hành: là các ngắt do hệ điều hành để phục vụ trao đổi tin của ngoại vi (bàn phím, màn hình, máy in...) như INT 10, INT16, INT 21...

Ngắt mềm chia làm 2 loại: Ngắt BIOS và ngắt DOS.

Ngắt BIOS (ROM-BIOS).

Ngắt của hệ điều hành và ngắt logic:

Bảng 1.2. Các loại ngắt của hệ điều hành và ngắt logic.

0	Chia cho 0	3	Điểm dừng
1	Chế độ từng bước 1 (Dùng để chạy thử chương trình)	4	Vượt quá nội dung

Ngắt để điều khiển ngoại vi:

Bảng 1.3. Các loại ngắt để điều khiển ngoại vi.

10	Điều khiển màn hình	16	Bàn phím (cùng với INT 9h)
11	Xác định thiết bị của máy tính (1 bit cho thiết bị)	17	Máy in
12	Xác định kích thước bộ nhớ	18	Điểm vào của BASIC chứa trong ROM
13	Các hành động đọc và viết lên đĩa	19	Điểm vào của phần chương trình móc nối (Boot:7 C00h)
14	Vào/ra cho liên lạc không đồng bộ	1A	Giờ của ngày (đọc và ghi ngày)
15	Hành động vào/ra lên cassette		

Ngắt phát bởi chương trình:

1B: Ngắt chương trình bởi CTRL – BREAK.

1C: Kiểm tra bởi bộ đếm đồng hồ.

Ngắt để truy nhập vào bảng thông số:

1D: Bảng các thông số Video có trong ROM. (Dùng để khởi phát khối điều khiển Video 6845).

1E: Bảng các số liệu của các đặc tính phụ (đồ họa)

b) Ngắt DOS (gọi các chức năng)

Bảng 1. 1. Các loại ngắt DOS.

20	Kết thúc chương trình (trả việc điều khiển cho MS-DOS)	25	Đọc trực tiếp trên đĩa với địa chỉ tuyệt đối.
21	Gọi những chức năng của DOS	26	Viết trực tiếp trên đĩa với địa chỉ tuyệt đối.
22	Kết thúc của 1 chương trình (điểm trên một chương trình con tiếp theo của chương trình chính)	27	Chấm dứt chương trình thường trú

23	Dừng khẩn cấp (CTRL-BREAK)	28÷3F	Các ngắt dừng cho bên trong của DOS
24	Dừng khi bộ xử lý có lỗi nghiêm trọng		

1.2.3. Xử lý ngắt của nhiều thiết bị ngoại vi

Tại một thời điểm có nhiều yêu cầu trao đổi tin từ các ngoại vi gửi đến yêu cầu ngắt chương trình nên phải sắp xếp yêu cầu để giải quyết:

- Cho phép hoặc cấm ngắt (che chắn) 1 yêu cầu ngắt của 1 ngoại vi.
- Ghi nhận và sắp xếp nhiều yêu cầu ngắt theo thứ tự ưu tiên.
- Xác định nguồn ngắt (xác định ngoại vi gây ra ngắt chương trình).
- Tạo vectơ ngắt, tức tạo địa chỉ ô nhớ trong đó chứa địa chỉ của lệnh đầu tiên của chương trình con phục vụ ngắt.

Cho phép ngắt và cấm ngắt:

Có hai cách bằng lệnh: Cấm ngắt DI (Disable Interrupt) và cho phép ngắt EI (Enable Interrupt). Hai lệnh này là của VXL Intel 8085 cho các lỗi vào ngắt RST 5,5; RST 6,5; RST 7,5 và INTR.

Ghi vào bit IF (Interrupt Flag) của thanh ghi Flag của VXL Intel 8086 các giá trị:

IF = 1: cho phép ngắt.

IF = 0: cấm ngắt.

Cho phép ngắt và cấm ngắt bằng mạch:

Với các lỗi vào ngắt không che được NMI (Nonmaskable Interrupt) hay các lỗi vào yêu cầu ngắt bất kỳ của ngoại vi người ta dùng các trigơ cho phép và cấm ngắt. Khi xác lập 1 (cho phép) và xóa về 0 (cấm ngắt).

Lỗi vào yêu cầu ngắt của ngoại vi được đưa vào 1 chân của cổng logic và chân kia lấy từ lỗi ra của trigơ. Lỗi ra cổng logic đưa vào NMI hoặc INTR hoặc các lỗi vào của sơ đồ xử lý ưu tiên ngắt.

Sắp xếp ưu tiên ngắt:

Nếu có nhiều hơn 1 yêu cầu ngắt thì phải ghi nhận và sắp xếp. Ghi nhận dùng trigơ (7475). Trước khi đưa tới các thanh ghi, các yêu cầu ngắt đi qua các bit của thanh ghi cho phép/che chắn.

Việc sắp xếp ưu tiên phục vụ ngắt dùng 1 trong 3 phương pháp:

Phương pháp chương trình hay hỏi vòng (polling):

Dùng chương trình đọc theo thứ tự ưu tiên các bit trạng thái của ngoại vi và kiểm tra chúng. Nếu bit trạng thái được đọc (=1) thì ngoại vi tương ứng yêu cầu trao đổi tin → VXL chuyển sang chương trình con phục vụ ngoại vi đó.

Sau đó lại đọc và kiểm tra bit trạng thái của ngoại vi có mức ưu tiên thấp hơn.

Ưu điểm: Đơn giản về thiết bị (chỉ cần có các thanh ghi trạng thái).

Nhược điểm: Máy tính luôn hỏi vòng các trạng thái của các ngoại vi. Tốn thời gian hỏi vòng các bit trạng thái của các ngoại vi. Thay đổi thứ tự ưu tiên bằng cách đọc trạng thái ngoại vi có ưu tiên nhất và bằng phương pháp ghi che yêu cầu ngắt.

Phương pháp ngắt cứng:

Tránh mất thời gian hỏi vòng, người ta chỉ tiến hành hỏi vòng khi có ngắt chương trình. Tức là có 1 trong nhiều ngoại vi gây ra ngắt chương trình ở lỗi vào ngắt INTR bằng cách dùng 1 cổng logic OR với:

- Các lỗi vào nối với các bit trạng thái của các ngoại vi.
- 1 lỗi ra nối với INTR của VXL.

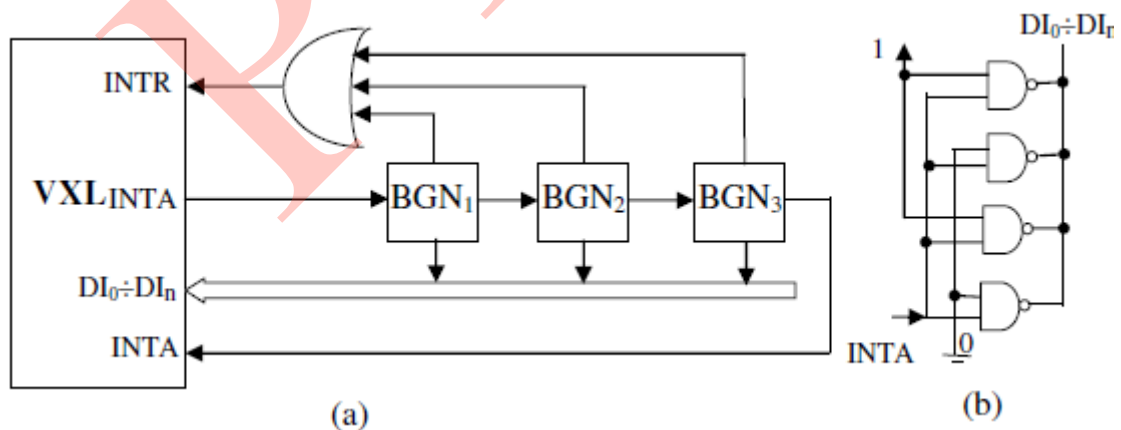
Phương pháp này giảm thời gian kiểm tra trạng thái khi chưa có yêu cầu nhưng không thay đổi được thứ tự ưu tiên 1 cách linh động.

Phương pháp xử lý ưu tiên bằng mạch cứng:

Có 2 loại: Ưu tiên theo vị trí lắp đặt gần VXL và Ưu tiên tùy ý dùng mạch so sánh.

Ưu tiên theo vị trí đặt gần VXL:

Từ sơ đồ ví dụ hình 1.4 có hoạt động như sau: Tín hiệu trả lời ngắt INTA (Interrupt Acknowledge) từ VXL đi nối tiếp qua bộ ghép nối BGN, mạch xử lý ngắt và trở về lại VXL theo một mắt xích (daisy chain). Bộ ghép nối nào đặt gần VXL có mức ưu tiên cao. Nếu một bộ ghép nối đã ghi nhận ngắt, tín hiệu INTA bị chặn và không truyền tới bộ ghép nối sau. Do đó, VXL không nhận được tín hiệu INTA, nghĩa là có một yêu cầu đã xảy ra.



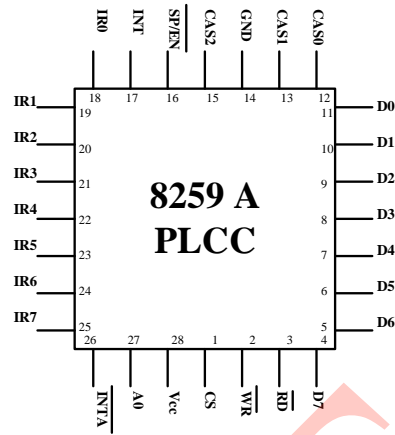
Hình 1.4. Sơ đồ ưu tiên ngắt theo vị trí và vector ngắt.

1.2.4. Vi mạch xử lý ngắt

Bộ điều khiển ngắt cứng PIC 8259A.

Trên thực tế có nhiều loại vi xử lý ngắt của nhiều hãng nổi tiếng như ZILOG, MOTOROLA, INTEL.... Nhưng vi mạch xử lý ngắt sử dụng trong máy tính IBM PC và tương thích là bộ vi điều khiển PIC 8259A.

Vi mạch này có các ưu điểm là: sắp xếp và xử lý ngắt tốt.



Hình 1. 4. Sơ đồ chân của 8259A

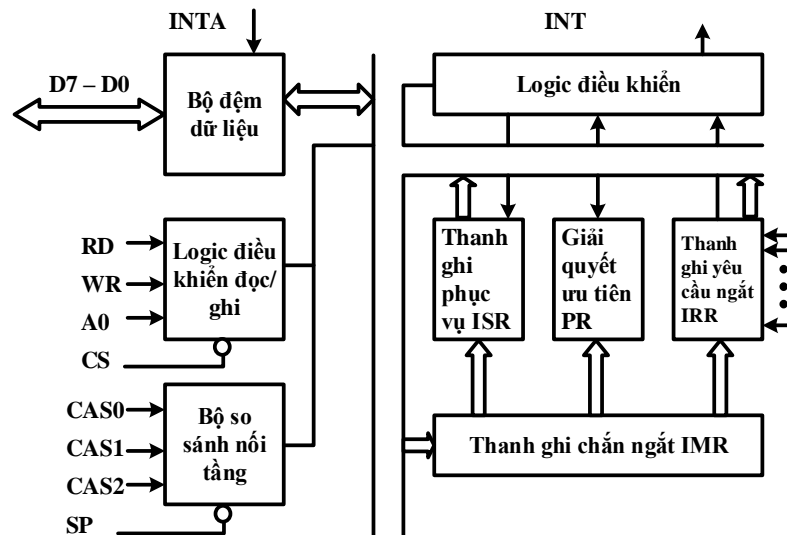
Có hai dạng PIC 8259A là:

DIP (Dual Inline Package): Loại hai hàng chân

PLCC: Loại có 4 hàng chân

Trong đó:

- CS (1) : Chip select chọn mạch điều khiển
- WR (2): Write (lối vào của lệnh Ghi)
- RD (3) : Read (lối vào lệnh đọc)
- D0 ÷ D7 (11÷4): Các bit dữ liệu hai chiều
- CAS0 (12), CAS1 (13), CAS2 (15): Lối vào mắc tầng của PIC chủ với PIC tớ
- SP/EN (16): Slave Programming /ANble
- INT (17): Lối ra yêu cầu ngắt
- IR0 ÷ IR7 (18÷25): các vào lối yêu cầu ngắt
- INTA (26): Lối ra yêu cầu ngắt
- A0 (27): Địa chỉ chọn thanh ghi lệnh
- Vcc (28): Nguồn nuôi



Hình 1. 5. Sơ đồ khối của 8259A

Trong đó:

- Bộ đệm dữ liệu (Data Buffer): được sử dụng khuếch đại dữ liệu
- Logic đọc/ ghi (Read/Write Logic): Điều khiển đọc ghi
- Bộ đệm và so sánh nối tầng: Nối PIC 8259A thành PIC 3
- Logic điều Khiển (Control Logic): tạo các tín hiệu ghi và đọc các thanh ghi đệm
- Thanh ghi ISR (In Service Register): chứa các chương trình con xử lý ngắt
- PR (Priority Resolver): Giải quyết ưu tiên
- IRR (Interrupt Request Register): thanh ghi tám mức ưu tiên từ ngoại vi.
- IMR (Interrupt Mask Register): cho người lập trình biết ngắt có che được hay không?

CÂU HỎI ÔN TẬP

1. So sánh hai phương pháp hồi vòng và ngắt.
2. Địa chỉ bắt đầu của chương trình con phục vụ ngắt là bao nhiêu?
3. Ưu nhược điểm của hai phương pháp vào/ra bằng chương trình và vào/ra bằng DMA?
4. Trong trường hợp nào thì vào/ra bằng DMA lại nhanh hơn bằng chương trình?
5. Tại sao truyền số liệu bằng DMA nhanh hơn truyền bằng chương trình?
6. Vì sao phải dùng lệnh IRET thay cho lệnh RET thông thường ở cuối thủ tục phục vụ ngắt?
7. Địa chỉ của chương trình con phục vụ ngắt 4 là 0010:0082. Hãy xác định thứ tự và địa điểm địa chỉ này được cất trong bảng vector ngắt?

CHƯƠNG 2. GHÉP NỐI QUA CỔNG SONG SONG

2.1. Ghép nối qua cổng song song

2.1.1. Giới thiệu chung về cổng song song

Cổng song song: Dữ liệu được truyền qua cổng này theo cách song song, cụ thể dữ liệu được truyền 8 bit đồng thời hay còn gọi byte nối tiếp bit song song.

Cổng máy in: Lí do là hầu hết các máy in đều được nối với máy tính qua cổng này.

Cổng Centronic: Đây là tên của một công ty đã thiết kế ra cổng này. Centronic là tên một công ty chuyên sản xuất máy in kiểu ma trận đứng hàng đầu thế giới. Chính công ty này đã nghĩ ra kiểu thiết kế cổng ghép nối máy in với máy tính.

2.1.1.1. Mức điện áp cổng

Cổng song song đều sử dụng mức điện áp tương thích TTL (Transistor - Transistor - Logic) từ $0V \rightarrow +5V$ trong đó:

- $0V$ là mức logic LOW.
- $2V \div 5V$ là mức logic HIGH.

Khi ghép nối với cổng song song chỉ ghép nối những thiết bị ngoại vi có mức điện áp tương thích TTL. Nếu thiết bị ngoại vi không có mức điện áp tương thích TTL thì phải áp dụng biện pháp ghép mức hoặc ghép cách ly qua bộ ghép nối quang.

2.1.1.2. Khoảng cách ghép nối

Khoảng cách cực đại giữa thiết bị ngoại vi và máy tính ghép qua cổng song song thường bị hạn chế do hiện tượng cảm ứng giữa các đường dẫn và điện dung ký sinh hình thành giữa các đường dẫn có thể làm biến dạng tín hiệu. Khoảng cách giới hạn cực đại là 8m. Thông thường chỉ 1,5 đến 2m vì lí do an toàn dữ liệu.

Nếu sử dụng khoảng cách ghép nối trên 3m thì các đường dây tín hiệu và đường dây nối đất phải được xoắn với nhau thành từng cặp để giảm thiểu ảnh hưởng của nhiễu; hoặc có thể sử dụng cáp dẹt, trên đó mỗi đường dữ liệu được đặt giữa hai đường dây nối đất.

2.1.1.3. Tốc độ truyền dữ liệu

Tốc độ truyền dữ liệu qua cổng song song phụ thuộc vào phần cứng được sử dụng. Trên lý thuyết tốc độ có thể đạt đến 1Mb/s nhưng với khoảng cách truyền hạn chế trong phạm vi 1m.

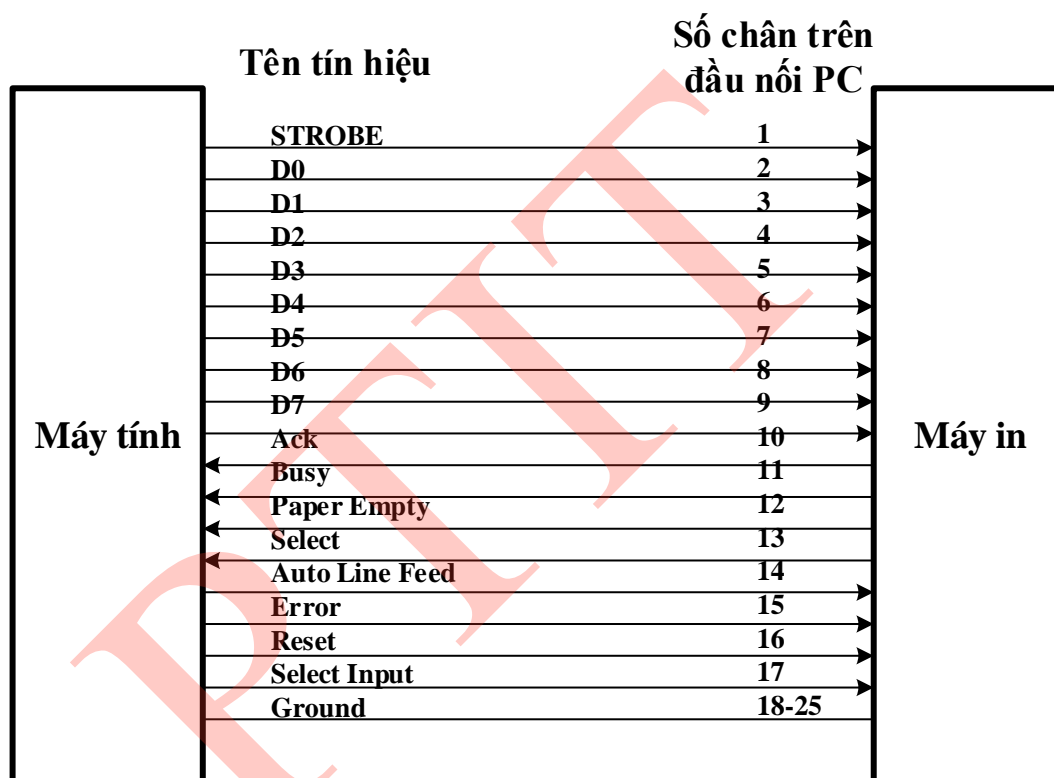
Với nhiều mục đích sử dụng thì khoảng cách này hoàn toàn thỏa đáng, tuy vậy cũng có những ứng dụng đòi hỏi phải truyền trên khoảng cách xa hơn. Trong trường hợp đó sự dụng khả năng ghép nối khác (như ghép nối qua cổng RS232).

2.1.2. Các thanh ghi ở cổng song song

Các đường dẫn của cổng song song được nối với ba thanh ghi 8 bit khác nhau:

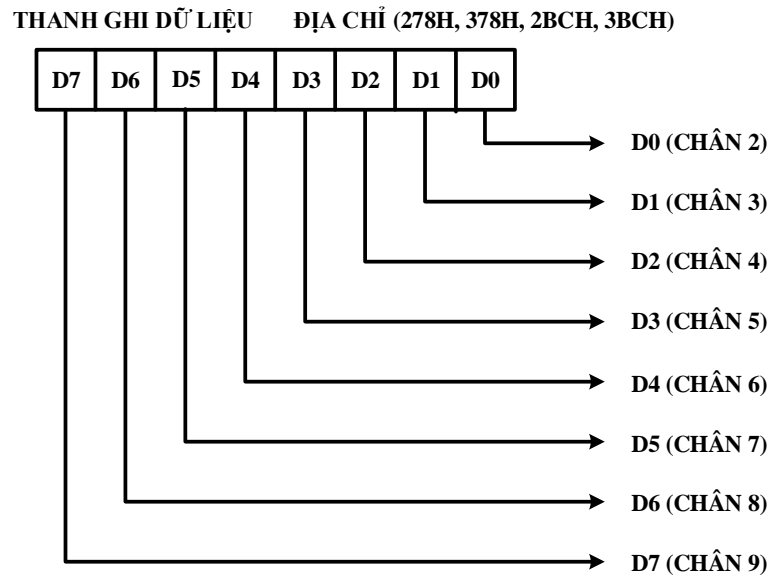
- Thanh ghi dữ liệu (Data Register).
- Thanh ghi trạng thái (Status Register)
- Thanh ghi điều khiển (Control Register)

Sơ đồ nối dây và chiều tín hiệu giữa máy tính và máy in



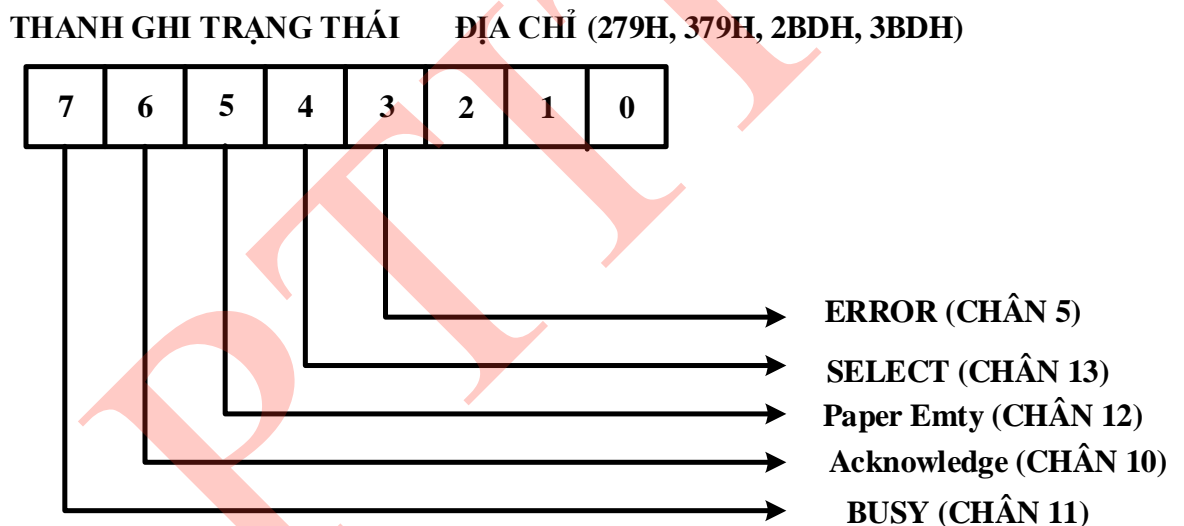
Hình 2.1. Sơ đồ kết nối giữa máy tính và máy in.

Thanh ghi dữ liệu: chứa dữ liệu từ D0 ÷ D7, dữ liệu này được truyền theo 2 hướng có thể vào/ra.



Hình 2.2. Sơ đồ định dạng của thanh ghi dữ liệu.

Thanh ghi trạng thái (chỉ đọc):



Hình 2.3. Sơ đồ định dạng của thanh ghi trạng thái

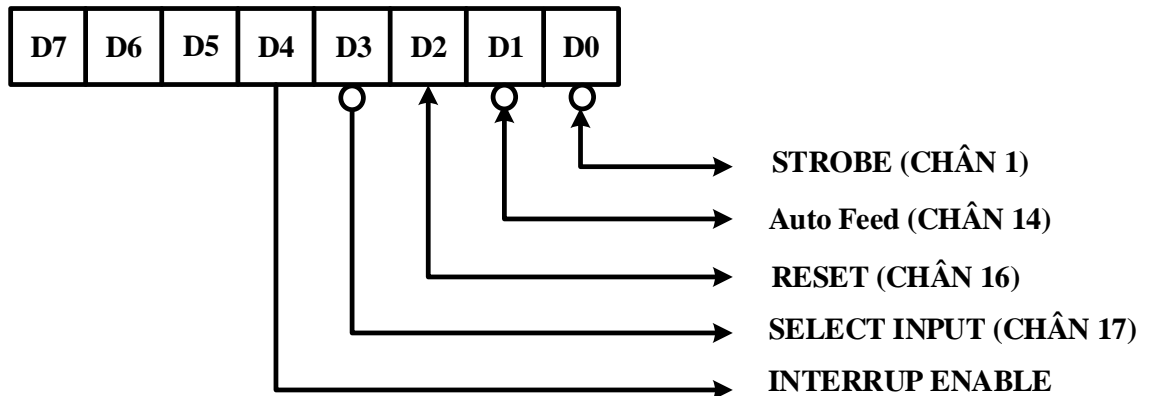
Bảng 2.1. Bảng chức năng các chân của thanh ghi trạng thái.

BSY	/ACK	PAP	OFON	/FEH	X	X	X
11	10	12	13	15	Không sử dụng		

- BSY = 1: không bận; 0: bận
- /ACK = 1: đang truyền dữ liệu; 0: đã truyền xong
- PAP = 1: không có giấy; 0: có giấy
- OFON = 1 : máy in sẵn sàng (Online); = 0 : Offline.
- /FEH = 1: không bị lỗi; = 0: bị lỗi (hết giấy, máy in offline, lỗi máy in)

Thanh ghi điều khiển:

THANH GHI ĐIỀU KHIỂN ĐỊA CHỈ (27AH, 37AH, 2BEH, 3BEH)



Hình 2.4. Sơ đồ định dạng của thanh ghi điều khiển

Bảng 2.2. Bảng chức năng các chân của thanh ghi điều khiển.

X	X	X	IRQ	/DSL	/INI	/ALF	/STR
Không sử dụng (= '1')				17	16	14	1

- IRQ = đòi hỏi ngắt cứng 1: cho phép; 0: không cho phép
- /DSL = 1: không chọn máy in; 0: máy in được chọn
- /INI = 1: không khởi động máy in; 0: khởi động máy in
- /ALF = 1 : máy in không tự động xuống dòng; = 0 : máy in tự động xuống dòng.
- STR = 1: không truyền dữ liệu đến máy in; = 0: truyền dữ liệu đến máy in

Như sơ đồ trên đã trình bày 8 đường dữ liệu dẫn tới thanh ghi dữ liệu còn 4 đường dẫn điều khiển **Strobe, Auto Linefeed, Reset, Select Input** dẫn tới thanh ghi điều khiển.

Năm đường dẫn trạng thái **Acknowledge, Busy, Paper Empty, Select, Error** tới thanh ghi trạng thái.

Thanh ghi dữ liệu hay 8 đường dẫn dữ liệu không phải là đường dẫn 2 hướng trong tất cả các loại máy tính nên dữ liệu chỉ có thể được *xuất ra* qua các đường dẫn này cụ thể từ **D0** đến **D7**.

Thanh ghi điều khiển hai hướng, hay nói chính xác hơn: Bốn bit có giá trị thấp được sắp xếp ở các chân **1, 14, 16, 17**. Thanh ghi trạng thái chỉ có thể được đọc và vì vậy được gọi là một hướng.

2.2. Các lệnh vào/ra dữ liệu

2.2.1. Các lệnh vào / ra bằng hợp ngữ

Lệnh để nhận dữ liệu từ thiết bị vào/ra là INput và một lệnh để gửi số liệu ra thiết bị vào/ra là OUTput.

Có bốn cách dùng khác nhau của mỗi lệnh này:

- 2 loại chuyển dữ liệu 8 hay 16 bit thông qua các cổng vào/ra 8 bit
- 2 loại chuyển dữ liệu 8 hay 16 bit thông qua các cổng 16 bit.

Ví dụ: 2 lệnh sau đọc các byte của đường dẫn dữ liệu từ cổng LPT1:

MOV DX,378H ; - nạp địa chỉ của thanh ghi dữ liệu vào DX

IN AL,DX ; - đọc thông tin trên đường dẫn dữ liệu (D0 đến D7) của LPT1 sang thanh ghi AL

Bảng 2.3. Bảng các lệnh vào/ra

Cấu trúc lệnh	Số bit dữ liệu	Chức năng
IN AL,d8	8	Đọc một byte từ cổng vào/ra 8 bit
IN AL,DX	8	Đọc một byte từ cổng vào/ra xác định bởi thanh DX
IN AX,d8	16	Đọc một word từ cổng vào/ra 8 bit
IN AX,DX	16	Đọc một word từ cổng vào/ra xác định bởi thanh DX
OUT d8, AL	8	Gửi một byte ra cổng vào/ra 8 bit
OUT DX, AL	8	Gửi một word ra cổng vào/ra xác định bởi thanh DX
OUT d8, AX	16	Gửi một word ra cổng vào/ra 8 bit
OUT DX, AX	16	Gửi một word ra cổng vào/ra xác định bởi thanh DX

Lưu ý : d8 chứa địa chỉ là cổng vào/ra 8 bit; DX: chứa địa chỉ là cổng vào/ ra 16 bit

2.2.2. Các lệnh vào / ra bằng Turbo C

Xuất ra dữ liệu: outportb (cổng, giá trị);

Nhập vào dữ liệu: x = inportb (cổng);

Bảng 2.4. Bảng các lệnh vào/ra

Cấu trúc lệnh	Số bit dữ liệu	Chức năng
Data_var=inportb(addr_port)	8	Đọc một byte từ cổng vào xác định bởi địa chỉ addr_port.
Outportb(addr_port, data_var)	8	Gửi một byte ra cổng vào/ra 8 bit xác định bởi địa

Cấu trúc lệnh	Số bit dữ liệu	Chức năng
		Chỉ <code>addr_port</code> .
<code>Data_var=inportb(addr_port)</code>	16	Đọc một byte từ cổng vào/ra 8 bit thấp xác định bởi địa chỉ <code>addr_port</code> , 8 bit cao xác định bởi địa chỉ <code>addr_port + 1</code> .
<code>Outportb(addr_port, data_var)</code>	16	Gửi một byte ra cổng vào/ra 8 bit thấp xác định bởi địa chỉ <code>addr_port</code> , 8 bit cao xác định bởi địa chỉ <code>addr_port + 1</code> .
<code>Data_var=inportb(addr_port)</code>	8	Đọc một byte từ cổng vào/ra 8 bit xác định bởi địa chỉ <code>addr_port</code> .
<code>Outportb(addr_port, data_var)</code>	8	Gửi một byte ra cổng vào/ra 8 bit xác định bởi địa chỉ <code>addr_port</code> .

Ví dụ: Xuất giá trị $5_{10} = (0000\ 0101)_2$ qua thanh ghi dữ liệu của LPT1.

Outportb (0x378, 5)

hoặc

#include <dos.h> --tệp đề mục cần có phải móc nối bằng câu lệnh này.

#define Dat_reg 0x378

char data_var=5;

outport(Dat_reg,data_var);

hoặc

#define Dat_reg 0x378

outport(Dat_reg, 5);

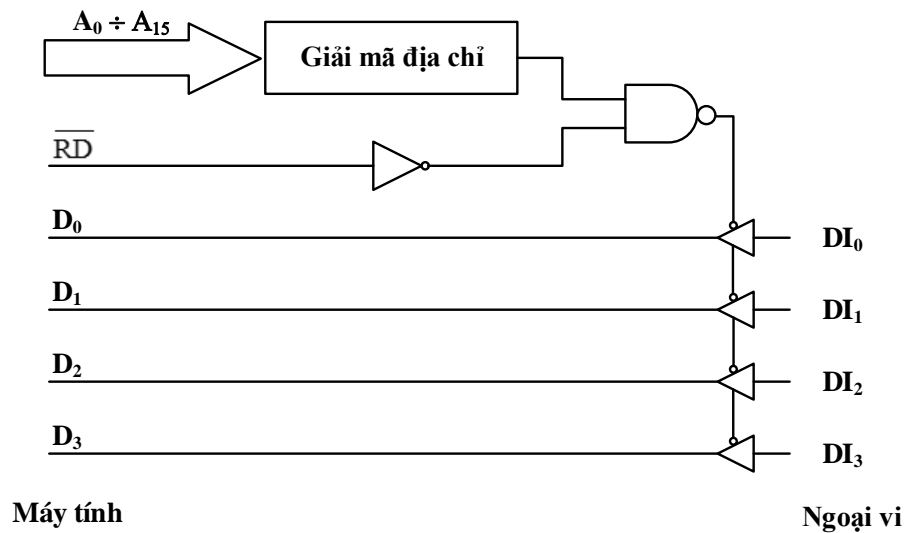
Để đọc thông tin của thanh ghi trạng thái và sắp xếp vào các biến đứng ở bên trái:

status = inportb (0x379);

2.3. Khởi ghép nối song song đơn giản

2.3.1. Cửa vào đơn giản

Cửa vào đơn giản: gồm bộ giải mã địa chỉ - lệnh và các cổng 3 trạng thái để đưa trực tiếp số liệu song song từ thanh ghi đếm số liệu của ngoại vi vào đường dây số liệu ($D0 \div Dn$) của máy tính.



Hình 2.5. Sơ đồ cửa vào đơn giản

Tín hiệu vào từ bộ cảm biến được nối với bus dữ liệu thông qua cổng 3 trạng thái.

Lập trình:

C: `data1 = inport (addport);`

Pascal: `data1 = port (addport);`

Trình biên dịch chuyển đoạn lệnh thành mã máy, khi đó VXL sẽ tạo ra các tín hiệu:

A0...A15 từ addrport.

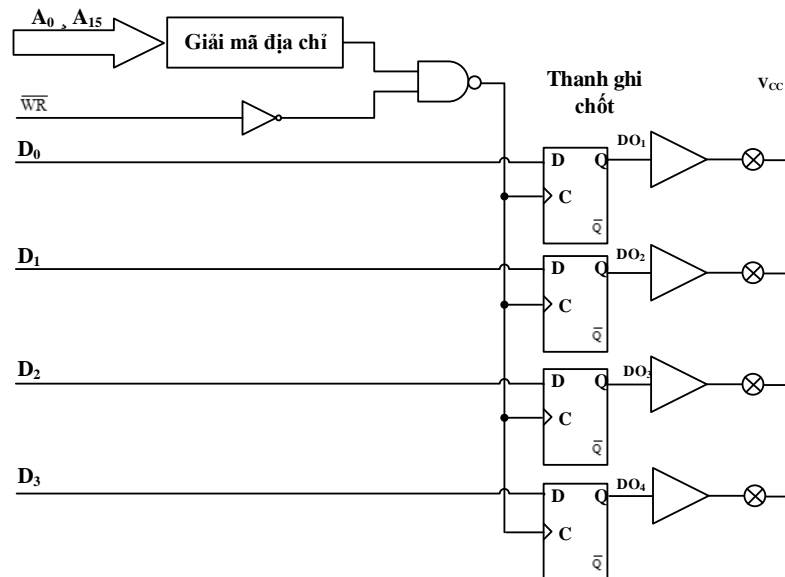
/RD kích hoạt ở mức 0.

Bộ giải mã sẽ so sánh các giá trị trên đường địa chỉ với địa chỉ cổng cho trước, nếu trùng địa chỉ đầu ra bộ giải mã kích hoạt ở mức 1, lỗi ra cổng NAND = 0 → kích hoạt cho bộ đệm 3 trạng thái mở ra và dữ liệu từ ngoại vi sẽ đi vào bus dữ liệu của VXL.

VXL sẽ nhận D0...D7 gán vào biến data1. Kết thúc chu kỳ lệnh /RD = 1.

2.3.2. Cửa ra đơn giản

Cửa ra đơn giản: gồm : gồm bộ giải mã địa chỉ - lệnh và có thêm các thanh ghi chốt số liệu ra để ghi số liệu từ máy tính ra. Lỗi ra có thể thêm sơ đồ ba trạng thái để cách ly ngoại vi với đường dây của máy tính.



Hình 2.6. Sơ đồ cửa ra đơn giản.

Dữ liệu từ VXL được gửi ra từ ngoại vi qua thanh ghi đệm dùng để điều khiển đèn. Mạch khuếch đại sẽ nâng công suất lên phù hợp với công suất của đèn.

Lệnh:

C: `outportb(addrport, data2);`

Pascal: `portb(addrport) := data2;`

Lúc này VXL sẽ tạo ra các tín hiệu:

A0...A15 từ addrport.

/WD kích hoạt ở mức 0.

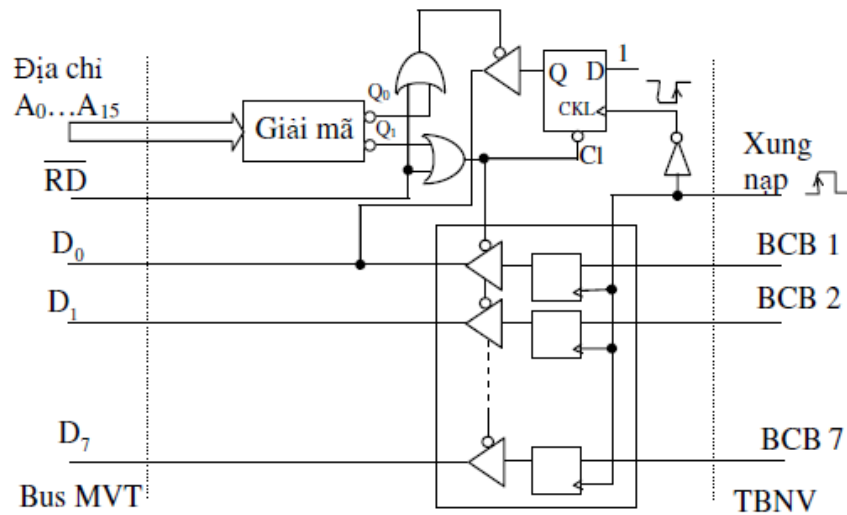
Bộ giải mã sẽ so sánh các giá trị trên đường địa chỉ với địa chỉ cổng cho trước, nếu trùng địa chỉ đầu ra bộ giải mã kích hoạt ở mức 1, lỗi ra cổng NAND = 0 → kích hoạt cho bộ đệm 3 trạng thái mở ra và ngoại vi nhận dữ liệu từ máy tính.

D0...D7 tương ứng với biến data2.

Kết thúc chu kỳ lệnh /WD = 1.

2.3.3. Cửa vào có đối thoại

Cửa vào có đối thoại: Lỗi vào có các trigơ D có xung Clock là tín hiệu ghi số liệu vào



Hình 2.7. Sơ đồ cửa vào có đối thoại thiết bị ngoại vi.

Ngoại vi gửi dữ liệu tới đầu vào các thanh ghi. Khi dữ liệu đã ổn định ngoại vi gửi tiếp một xung nạp. Sườn dương của xung nạp đưa dữ liệu của ngoại vi vào nhớ trong thanh ghi, sườn âm của xung nạp kích thích mạch lật, đưa đầu ra $Q = D = 1$. Đây là trạng thái sẵn sàng của ngoại vi.

VXL sẽ kiểm tra trạng thái bằng lệnh:

```
#define BIT0 0x01;
#define addrport1 0x00F8;
#define addrport3 0x00F9;
do
data3 = inportb(addrport1);
while (data3 & BIT0 != BIT0);
data4 = inportb(addrport2);
```

Bằng lệnh *inportb* đầu tiên, VXL sẽ đưa địa chỉ *addrport1* ra các chân địa chỉ, kích hoạt $\overline{RD} = 0$. Đầu ra $OR = 0$, mở mạch đệm 3 trạng thái, đưa tín hiệu $Q = 1$ vào đường dữ liệu D0.

$D0 = 1$ sẽ được gán cho biến *data3*.

Kết thúc chu kỳ lệnh, $\overline{RD} = 1$, đầu ra cổng OR chuyển sang mức 1 làm mạch đệm 3 trạng thái chuyển sang trạng thái Z cao, cách ly chân Q và đường dữ liệu D0.

Sau khi VXL kiểm tra thấy dữ liệu đã sẵn sàng ($D0 = 1$), lệnh *inport* thứ 2 sẽ làm VXL tạo tín hiệu kích mở bộ đệm 3 trạng thái để đưa dữ liệu của ngoại vi đang nhớ trong các thanh ghi vào bus dữ liệu và gán cho biến *data4*.

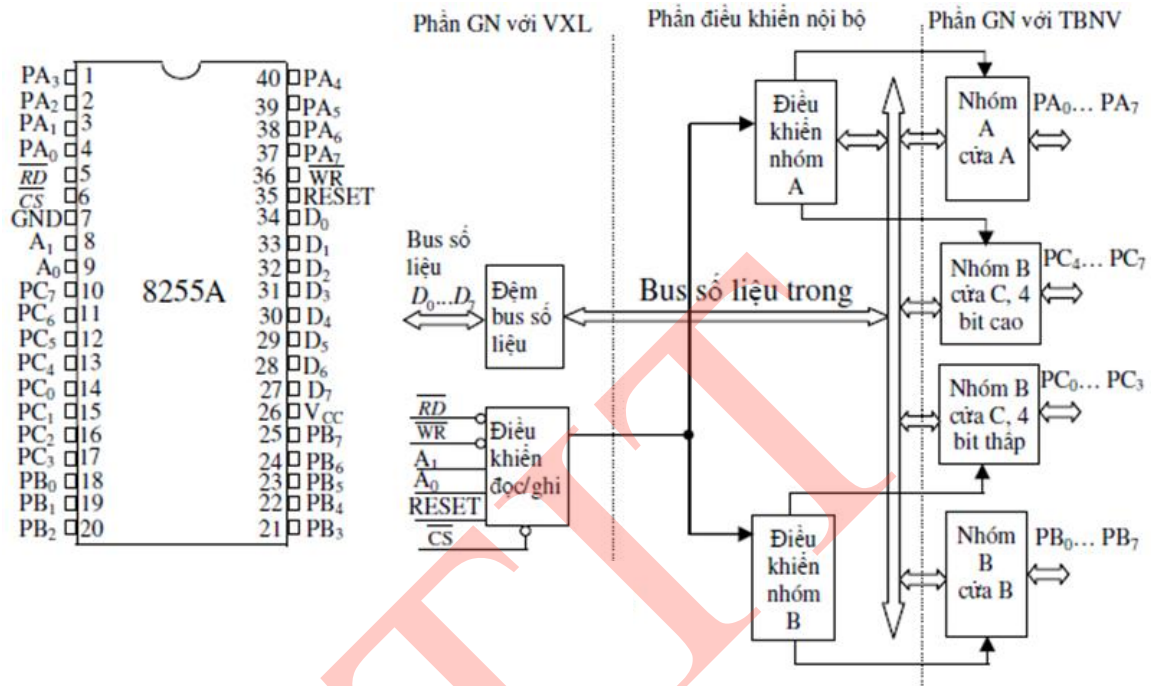
Cũng bằng xung chọn mạch này Trigrơ D sẽ xóa Q về mức logic 0.

Nên 1 thuật toán cần dùng cho trao đổi không đối thoại là đọc nhiều lần và kiểm tra kết quả các lần với nhau.

Nếu kết quả giống nhau thì dữ liệu vào là ổn định

2.4. Ghép nối song song điều khiển bằng chương trình

2.4.1. Sơ đồ khối và chức năng các khối của 8255A



Hình 2.9. Sơ đồ chân và sơ đồ kết nối của 8255A

8255A là vi mạch giao tiếp song song rất thông dụng trong toàn các hệ thống.

Vi mạch này có 24 đầu của 3 cổng vào/ra 8 bit 2 hướng:

- PA0-PA7: cổng A, PB0-PB7: cổng B, PC0-PC7: cổng C được chia thành 2 nhóm A và B.
- Nhóm A: gồm cổng A và nibble cao của cổng C (CU);
- Nhóm B: gồm cổng B và nibble thấp của cổng C (CL).
- Vi mạch này hoạt động trong 3 chế độ (0, 1 và 2)

Bộ đệm số liệu là loại 8 bit – 2 hướng – 3 trạng thái được dùng để phối ghép 8255A với bus hệ thống của máy tính. Số liệu sẽ được bộ đệm phát hoặc thu thông qua việc chạy các lệnh OUT hoặc IN của CPU. Các từ điều khiển và thông tin trạng thái cũng được truyền qua bộ đệm này.

Các mạch logic đọc/viết và điều khiển quản lý các quá trình truyền số liệu, truyền các từ điều khiển bên trong và bên ngoài vi mạch. Nó chấp nhận các tín hiệu

vào từ các bus địa chỉ và điều khiển của CPU và lần lượt phát ra các lệnh tới cả 2 nhóm điều khiển A và B.

- $\overline{CS} = 0$ thì cho phép 8255A thông tin với CPU.
- $\overline{RD} = 0$ thì cho phép CPU đọc số liệu hoặc thông tin trạng thái từ 8255A.
- $\overline{WR} = 0$ thì cho phép CPU viết số liệu hoặc từ điều khiển vào 8255A.
- Hai chân A1, A0 thường được nối với 2 bit địa chỉ thấp nhất của bus địa chỉ.
- 2 chân này dùng để chọn thanh ghi nội để lập trình hoặc hoạt động.

Để 8255A có thể đọc ghi thì chân CS phải ở mức logic 0 và địa chỉ I/O phải được nối với chân A1 và A0, không cần quan tâm đến các chân địa chỉ cổng còn lại và được giải mã bên ngoài để chọn 8255A

Bảng 2.5. Bảng chức năng của 8255A

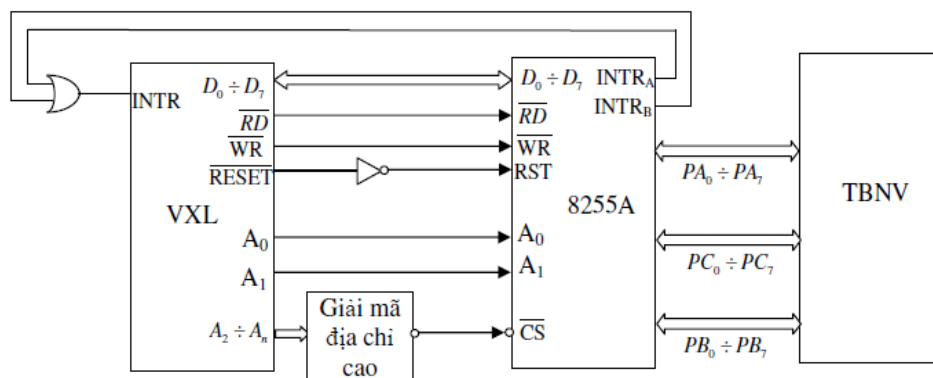
A1	A0	Chức năng
0	0	Cổng A
0	1	Cổng B
1	0	Cổng C
1	1	Thanh ghi lệnh

Bảng 2.6. Bảng điều khiển đọc/viết của 8255A.

A1	A0	\overline{RD}	\overline{WR}	Chức năng
0	0	0	1	Đọc cổng A vào bus dữ liệu
0	1	0	1	Đọc cổng B vào bus dữ liệu
1	0	0	1	Đọc cổng C vào bus dữ liệu
1	1	0	1	Đọc từ điều khiển vào bus dữ liệu
0	0	1	0	Viết từ bus số liệu vào cổng A
0	1	1	0	Viết từ bus số liệu vào cổng B
1	0	1	0	Viết từ bus số liệu vào cổng C
1	1	1	0	Viết từ bus số liệu vào từ điều khiển

Chân RESET = 1 sẽ xóa các thanh ghi điều khiển và đặt tất cả các cổng A, B, C ở chế độ vào

2.4.2. Chế độ làm việc của 8255A



Hình 2.10. Sơ đồ ghép nối của 8255A với thiết bị ngoại vi.

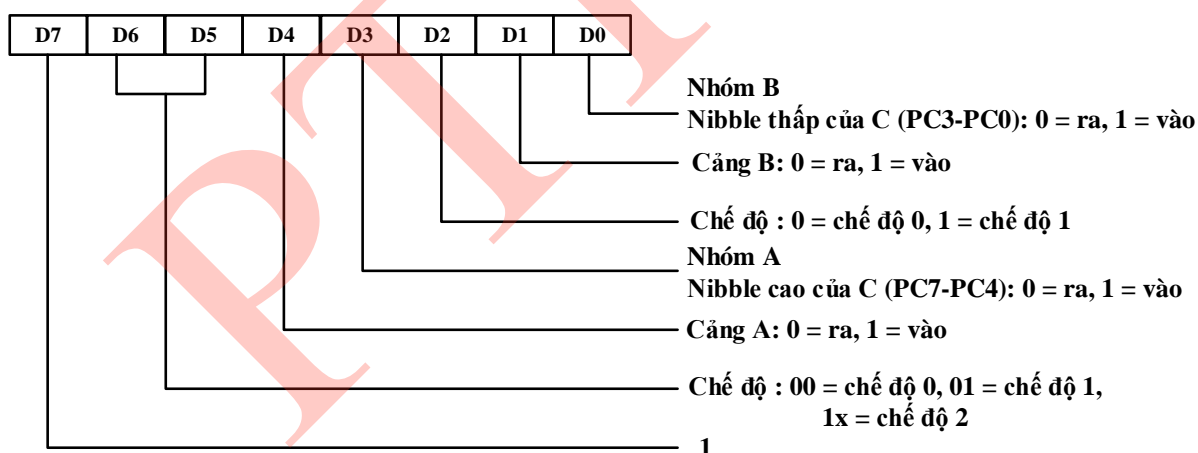
Vi mạch 8255 có ba chế độ hoạt động cơ bản (Mode) được chọn bởi phần mềm.

Sau khi được Reset tất cả các cổng của 8255A sẽ được đặt ở chế độ vào (tức là 24 chân của 3 cổng A, B, C ở trạng thái Z cao).

Trong quá trình chạy chương trình, bằng các lệnh viết ra địa chỉ của thanh ghi từ điều khiển có thể chọn các chế độ hoạt động tùy ý.

2.4.2.1. Định dạng của từ điều khiển

➤ Chế độ viết từ điều khiển (D7 = 1)



Hình 2.11. Sơ đồ định dạng của từ điều khiển.

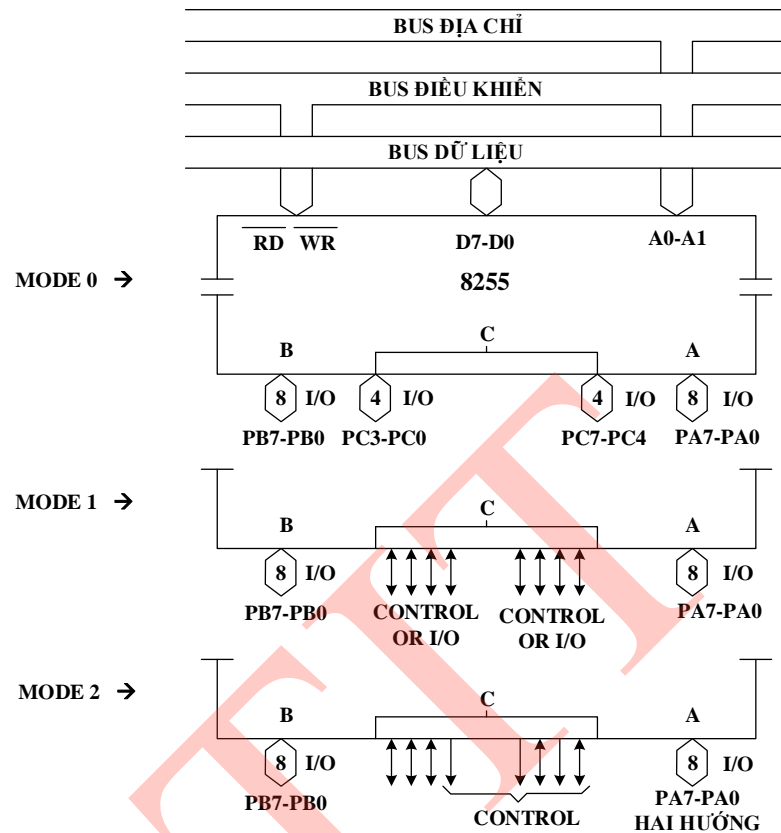
- Mode 0 : Chế độ vào/ ra cơ bản
- Mode 1 : Chế độ vào/ ra có hội thoại (Strobe Input/Output)
- Mode 2 : Chế độ vào/ ra bus hai chiều .

Tùy theo từ điều khiển ghi vào thanh ghi điều khiển khi khởi động cho vi mạch mà ta có chế độ làm việc và chiều trao đổi dữ liệu của các cổng A, B, C khác nhau.

Khi tín hiệu RESET bằng 1 tất cả các cổng được đưa về trạng thái là các cổng vào (Input port) nghĩa là tất cả 24 đường của ba cổng đều ở trạng thái trở kháng cao).

Sau khi tín hiệu RESET bằng 0 thì 8255 có thể duy trì trạng thái trên nếu như không có sự thiết lập trạng thái bổ xung nào nữa.

Trong quá trình thực hiện chương trình của hệ thống, bất kỳ một chế độ nào khác có thể được chọn nhờ sử dụng lệnh OUT



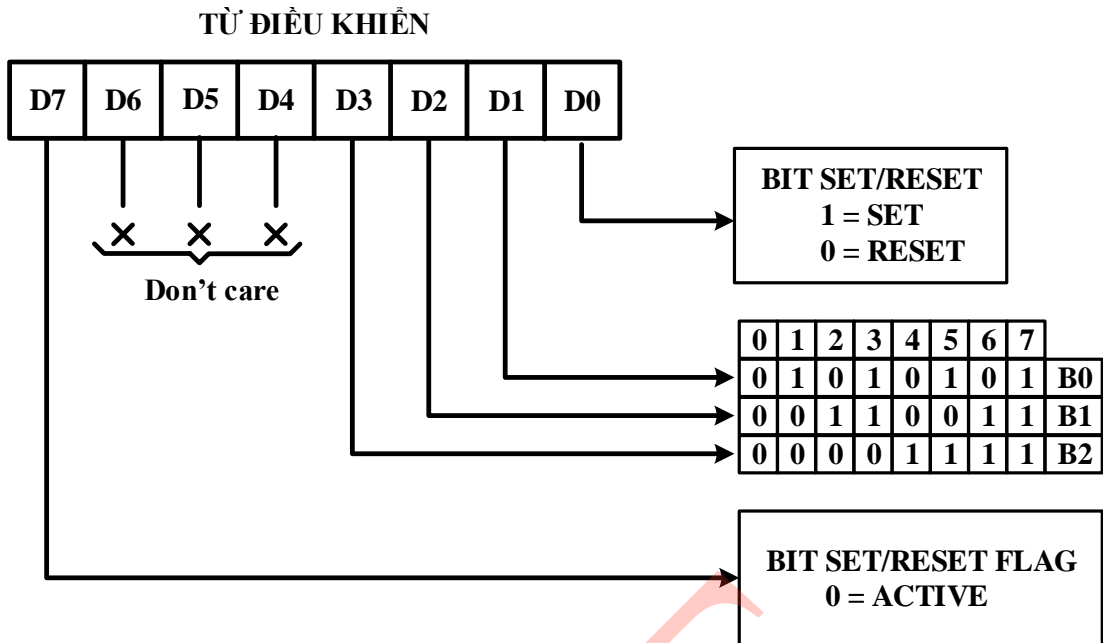
Hình 2.12. Các chế độ hoạt động của thanh ghi điều khiển

2.4.2.2. Chức năng xoá bit và lập bit (single bit set/reset)

Bất cứ bit nào trong 8 bit của cổng C đều có thể được thiết lập hoặc được xoá nhờ sử dụng các lệnh ghi ra thanh ghi điều khiển các bit phù hợp.

Như vậy khi cổng C đang được dùng để điều khiển, ghi trạng thái cho cổng A hoặc cổng B thì những bit của cổng C có thể được xoá hoặc được thiết lập nhờ sử dụng các lệnh xoá, thiết lập bit như khi cổng C là cổng ra dữ liệu.

Từ điều khiển hoạt động trong chế độ như sau :



Hình 2.13. Sơ đồ định dạng của từ điều khiển.

2.4.2.3. Chức năng điều khiển ngắt (Interrupt control)

Khi 8255 được lập trình hoạt động ở mode 1 hoặc mode 2 các tín hiệu điều khiển của nó có thể được sử dụng như yêu cầu ngắt tới CPU.

Tín hiệu yêu cầu ngắt tạo bởi cổng C có thể bị cấm hay được phép nhờ xoá hay xác lập mạch lật INTE sử dụng chức năng lập xoá bit như đã nêu trên.

Chức năng này cho phép người lập trình có thể cho phép hoặc không cho phép một thiết bị I/O ngắt CPU mà không gây ảnh hưởng tới bất kỳ một thiết bị nào khác trong cấu trúc ngắt.

2.4.2.4. Các chế độ

Các cổng A, B, C có thể hoạt động ở 3 chế độ 0, 1 và 2 khác nhau.

1/ Mode 0 (Basic input/output)

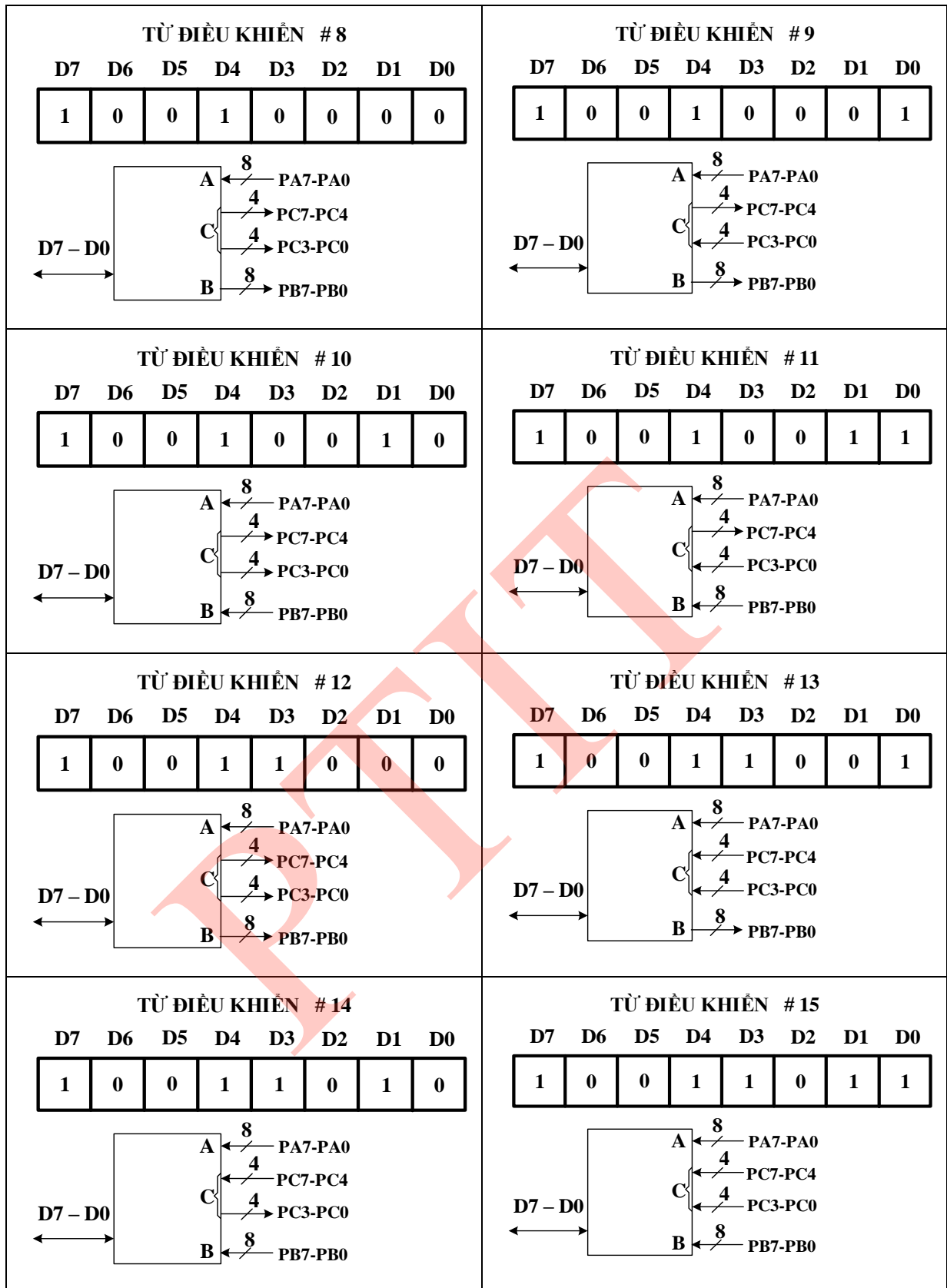
Đây là chế độ vào ra cơ bản của vi mạch, nó đảm bảo dữ liệu được đưa ra hoặc ghi vào các cổng riêng biệt.

Trong chế độ này, vi mạch có các chức năng sau :

- Vi mạch hoạt động gồm hai cổng 8 bit và hai cổng 4 bit
- Các cổng có thể là cổng vào hoặc cổng ra.
- Các tín hiệu ra được chốt lại.
- Các tín hiệu vào không được chốt.

Trong chế độ này 8255 có 16 cấu hình hoạt động vào/ra:

<div><div>TỪ ĐIỀU KHIỂN #0</div><div><div>D7D6D5D4D3D2D1D0</div><div><div>10000000</div></div></div><div><div><div>D7 - D0</div><div><div>A8PA7-PA0</div><div><div>C4PC7-PC4</div><div><div>C4PC3-PC0</div><div><div>B8PB7-PB0</div></div></div></div></div></div></div></div>	<div><div>TỪ ĐIỀU KHIỂN #1</div><div><div>D7D6D5D4D3D2D1D0</div><div><div>10000001</div></div></div><div><div><div>D7 - D0</div><div><div>A8PA7-PA0</div><div><div>C4PC7-PC4</div><div><div>C4PC3-PC0</div><div><div>B8PB7-PB0</div></div></div></div></div></div></div></div>
<div><div>TỪ ĐIỀU KHIỂN #2</div><div><div>D7D6D5D4D3D2D1D0</div><div><div>10000010</div></div></div><div><div><div>D7 - D0</div><div><div>A8PA7-PA0</div><div><div>C4PC7-PC4</div><div><div>C4PC3-PC0</div><div><div>B8PB7-PB0</div></div></div></div></div></div></div></div>	<div><div>TỪ ĐIỀU KHIỂN #3</div><div><div>D7D6D5D4D3D2D1D0</div><div><div>10000011</div></div></div><div><div><div>D7 - D0</div><div><div>A8PA7-PA0</div><div><div>C4PC7-PC4</div><div><div>C4PC3-PC0</div><div><div>B8PB7-PB0</div></div></div></div></div></div></div></div>
<div><div>TỪ ĐIỀU KHIỂN #4</div><div><div>D7D6D5D4D3D2D1</div><div><div>1000100</div></div></div><div><div><div>D7 - D0</div><div><div>A8PA7-PA0</div><div><div>C4PC7-PC4</div><div><div>C4PC3-PC0</div><div><div>B8PB7-PB0</div></div></div></div></div></div></div></div>	<div><div>TỪ ĐIỀU KHIỂN #5</div><div><div>D0D7D6D5D4D3D2D1D0</div><div><div>010001001</div></div></div><div><div><div>D7 - D0</div><div><div>A8PA7-PA0</div><div><div>C4PC7-PC4</div><div><div>C4PC3-PC0</div><div><div>B8PB7-PB0</div></div></div></div></div></div></div></div>
<div><div>TỪ ĐIỀU KHIỂN #6</div><div><div>D7D6D5D4D3D2D1D0</div><div><div>10001010</div></div></div><div><div><div>D7 - D0</div><div><div>A8PA7-PA0</div><div><div>C4PC7-PC4</div><div><div>C4PC3-PC0</div><div><div>B8PB7-PB0</div></div></div></div></div></div></div></div>	<div><div>TỪ ĐIỀU KHIỂN #7</div><div><div>D7D6D5D4D3D2D1D0</div><div><div>10001011</div></div></div><div><div><div>D7 - D0</div><div><div>A8PA7-PA0</div><div><div>C4PC7-PC4</div><div><div>C4PC3-PC0</div><div><div>B8PB7-PB0</div></div></div></div></div></div></div></div>



Hình 2.14. Các cấu hình hoạt động của Mode 0.

2/ Mode 1 (Strobe input/output)

Trong chế độ này cổng A và cổng B sử dụng các đường dây tín hiệu của cổng C để tạo hoặc tiếp nhận tín hiệu hội thoại (handshaking signal) nghĩa là mọi quá trình trao đổi dữ liệu của từng cổng đều dùng các tín hiệu hội thoại.

Các chức năng cơ bản của Mode 1.

- Vi mạch hoạt động gồm hai nhóm ,nhóm A và nhóm B.
- Mỗi nhóm chứa một cổng 8 bit và một cổng điều khiển 4 bit.
- Cổng 8 bit có thể là cổng vào, hoặc cổng ra, cả hai cổng vào ra đều là cổng chốt.
- Các cổng 4 bit được sử dụng để điều khiển và xác định trạng thái của các cổng 8 bit.

Các tín hiệu điều khiển vào được dùng trong chế độ vào :

/STB (Strobe input): Mức thấp của tín hiệu vào này cho phép dữ liệu được đọc vào .

IBF (Input Buffer Full): Mức cao của tín hiệu ra này chỉ ra rằng dữ liệu đã được ghi vào cổng chốt, về bản chất đây là một tín hiệu xác nhận.

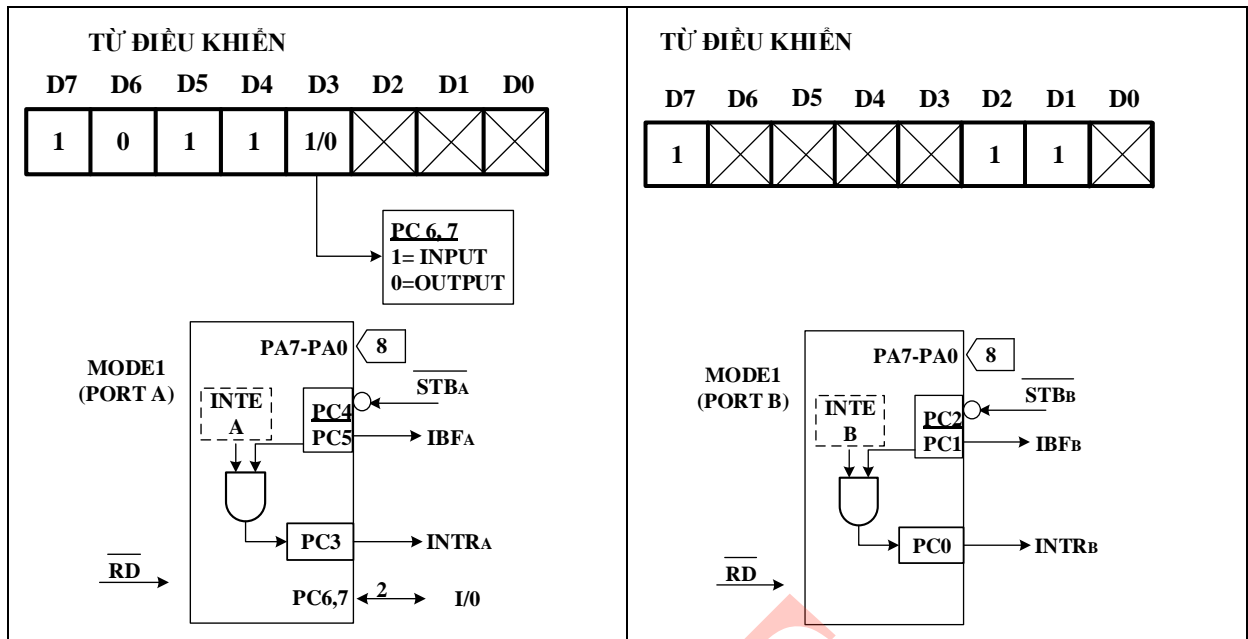
Tín hiệu IBF được xác lập khi tín hiệu STB đang ở mức thấp và được khởi tạo lại khi có sườn dương của đầu vào RD .

INTR (Interrupt Request): Mức cao của tín hiệu ra này có thể được sử dụng để yêu cầu ngắt tới CPU.

Khi một thiết bị vào yêu cầu phục vụ, tín hiệu INTR được xác lập bởi tín hiệu $STB = 1$.

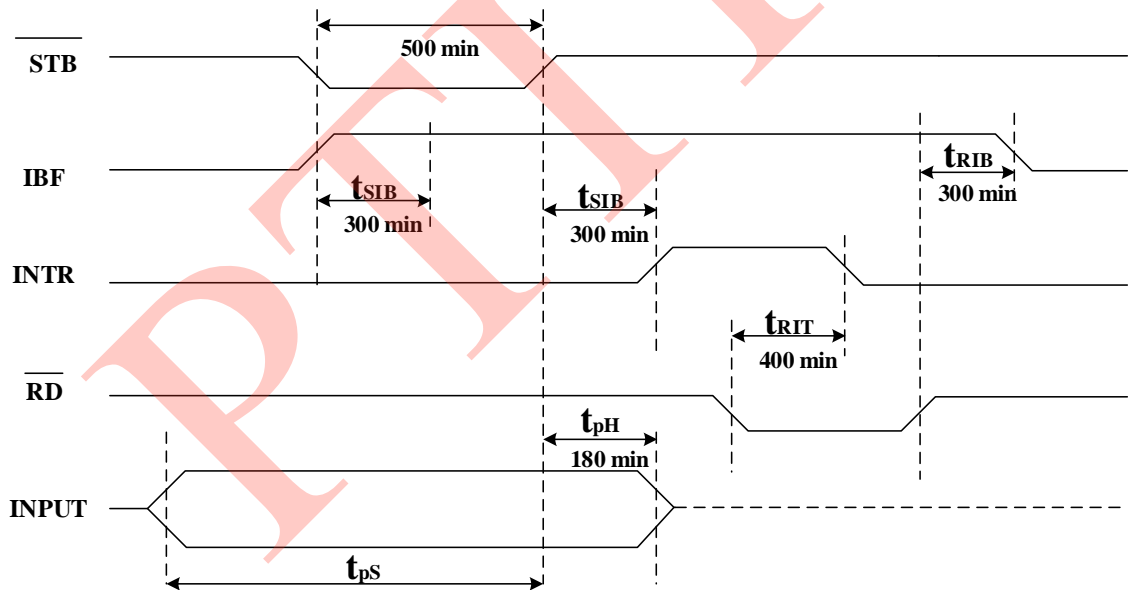
Tín hiệu $IBF=1$ và $INTE = 1$: Tín hiệu này được khởi tạo lại khi ở sườn âm của tín hiệu RD. Chức năng này cho phép các thiết bị vào yêu cầu ngắt tới CPU một cách đơn giản bằng cách đưa dữ liệu của nó ra cổng.

Các từ lệnh điều khiển.



Hình 2.15. Sơ đồ minh họa các từ lệnh điều khiển dùng trong chế độ vào.

Đồ thị thời gian của chế độ vào được minh họa bởi hình 2.16.



Hình 2.16. Đồ thị thời gian của chế độ vào mô tả quá trình đọc.

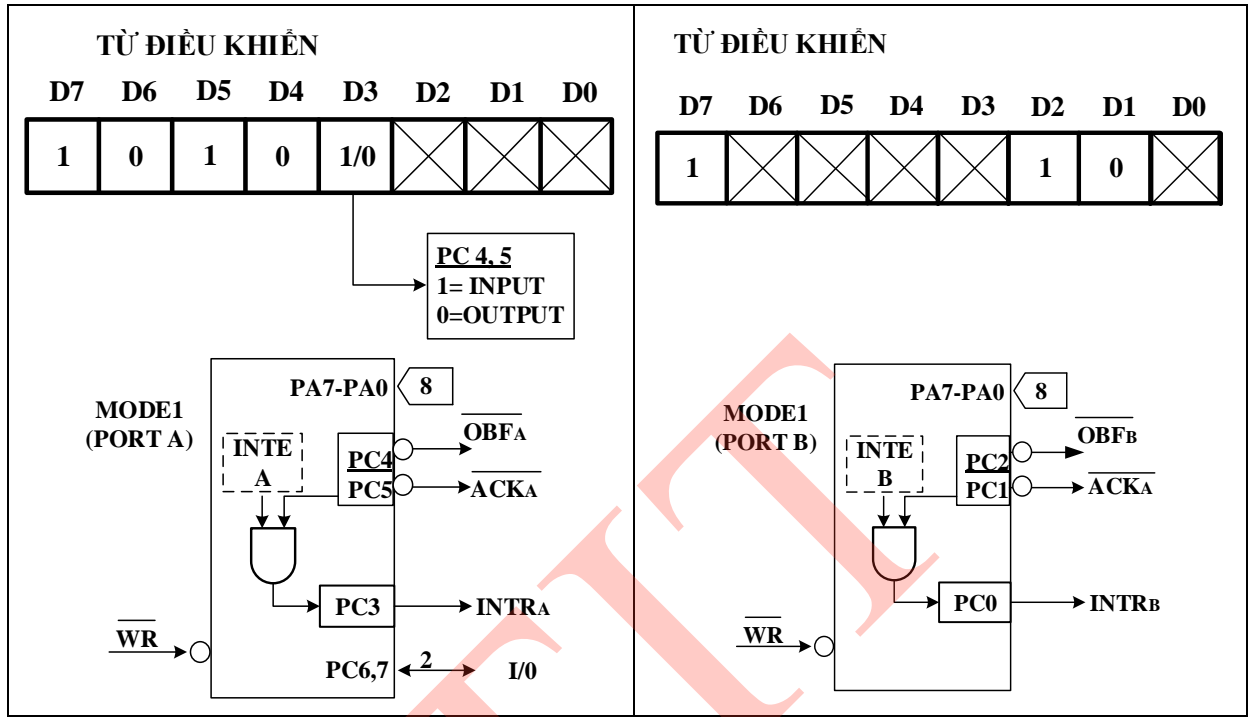
Các tín hiệu điều khiển ra được dùng trong chế độ ra.

/OBF (Output Bufer Full F/F): Tín hiệu ra OBF chuyển về mức thấp để thông báo rằng CPU đã ghi dữ liệu ra một cổng xác định. Tín hiệu OBF F/F sẽ được xác lập ở sườn lên của tín hiệu WR và bị xoá khi tín hiệu vào ACK ở mức thấp.

/ACK (Acknowledge Input): Mức thấp của tín hiệu vào này thông báo cho 8255 rằng dữ liệu từ cổng A hoặc cổng B đã được chấp nhận. Về bản chất đây là tín hiệu trả lời từ thiết bị ngoại vi thông báo nó đã nhận được dữ liệu gửi tới từ CPU.

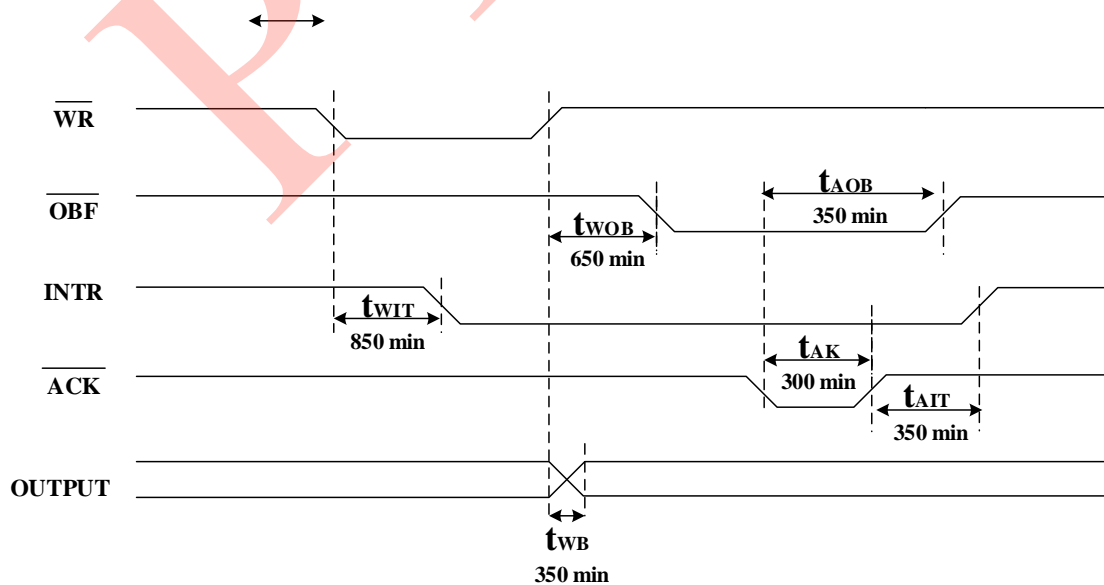
INTR (Interrupt Request): Mức cao của tín hiệu ra này được sử dụng để yêu cầu ngắt CPU khi một thiết bị ngoài đã nhận được dữ liệu truyền từ CPU. Tín hiệu INTR xác lập khi tín hiệu ACK = "1", OBF = "1" và INTE = "1". Tín hiệu này được khởi tạo lại ở sườn âm của tín hiệu WR .

Các từ lệnh của chế độ này như sau :



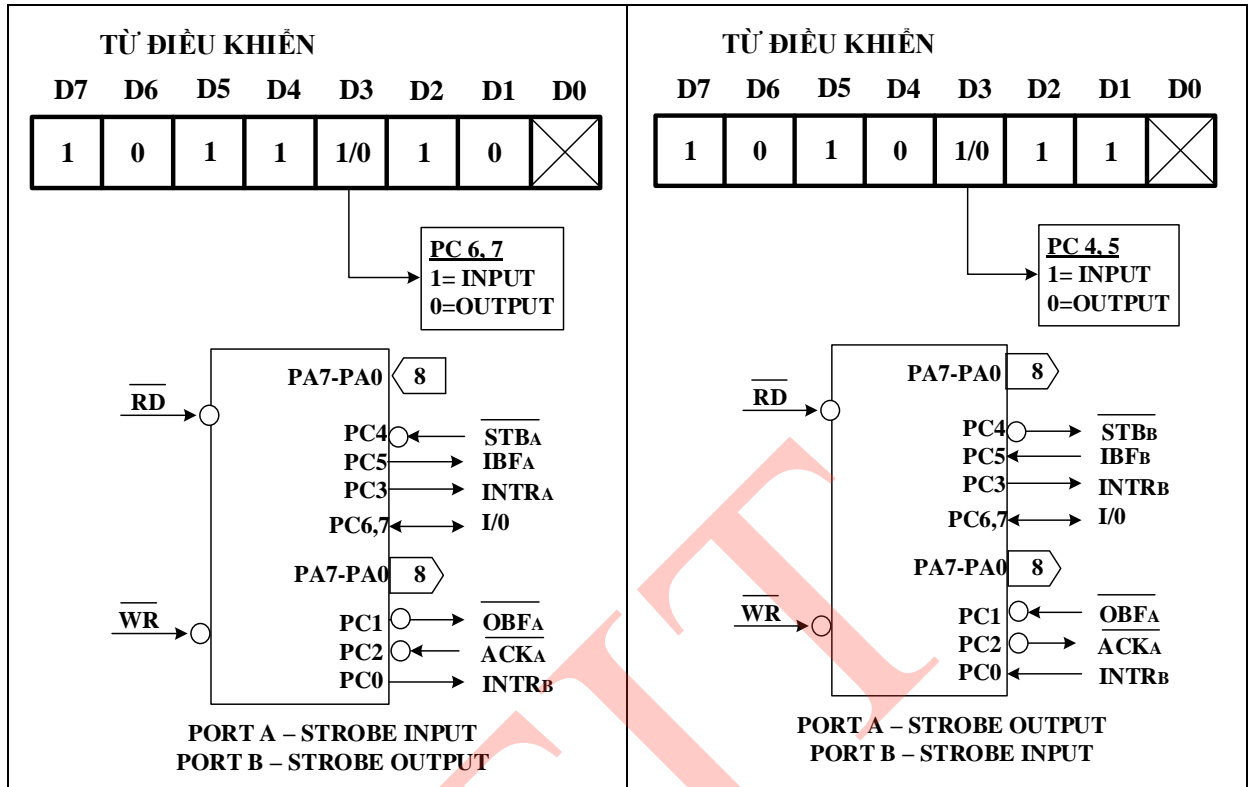
Hình 2.17. Sơ đồ minh họa các từ lệnh điều khiển dùng trong chế độ ra.

Đồ thị thời gian của chế độ hoạt động này như sau :



Hình 2.18. Đồ thị thời gian của chế độ ra mô tả quá trình ghi.

Trong mode 1, hai cổng A và B có thể được lập trình một cách riêng biệt là cổng vào hoặc cổng ra để hoạt động trong các ứng dụng vào /ra có hội thoại khác nhau. Các từ lệnh hoạt động trong chế độ này như sau.



Hình 2.19. Sơ đồ minh họa các từ lệnh hoạt động trong mode 1.

3/ Mode 2 (*Strobed bi-directional Bus I/O*).

Chế độ hoạt động này cung cấp khả năng trao đổi dữ liệu với các thiết bị ngoại vi sử dụng một đường truyền 8 bit để vừa truyền vừa nhận dữ liệu (Bus vào/ra hai chiều).

Các tín hiệu hội thoại được dùng trong chế độ này để điều khiển việc truyền dữ liệu cũng tương tự như mode 1.

Các chức năng cơ bản của Mode 2.

Trong chế độ này chỉ có nhóm A được sử dụng. Cổng A là cổng vào/ ra hai chiều 8 bit. Các tín hiệu vào/ra đều được chốt lại. 5 bit của cổng C được sử dụng làm cổng điều khiển, trạng thái cho cổng A 8 bit.

Các tín hiệu điều khiển ra .

/OBF (Output buffer full): Tín hiệu ra OBF ở mức thấp thông báo CPU đã khi dữ liệu ra cổng A.

/ACK (Acknowledge): Mức thấp của tín hiệu vào này cho phép bộ đệm ra bus dữ liệu của cổng A gửi dữ liệu. Ngược lại, bộ đệm ra này sẽ ở trạng thái trở kháng cao .

INTE 1 (Tín hiệu INTE phối hợp với tín hiệu OBF). Điều khiển bởi việc lập bit hoặc xóa bit PC4 .

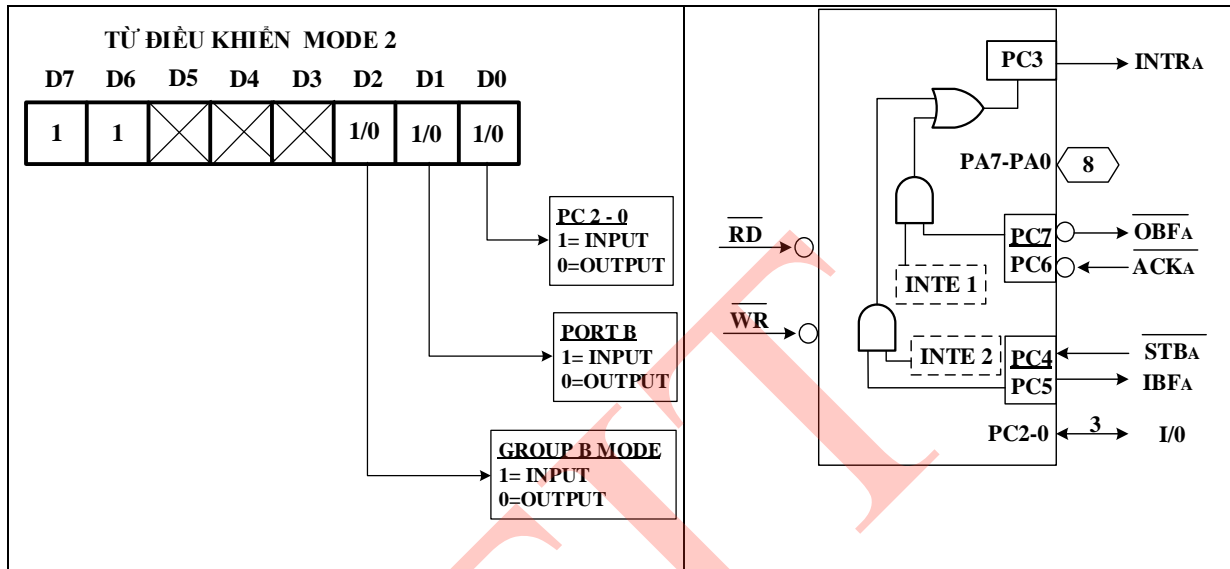
Các tín hiệu điều khiển vào .

/STB (Srobe input): Mức thấp của tín hiệu vào này đọc dữ liệu vào cổng vào chốt.

IBF (Input bufer Full): Mức cao của tín hiệu ra này chỉ ra rằng tín hiệu đã được ghi vào cổng vào chốt .

INTE2 (Tín hiệu INTE phối hợp với tín hiệu IBF): Điều khiển bởi việc xoá /lập bit của PC4

Các từ lệnh hoạt động trong chế độ này như sau :



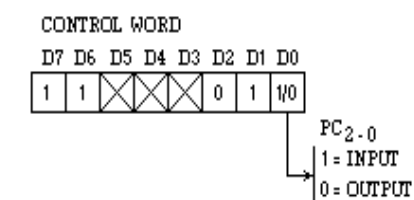
Hình 2.20. Sơ đồ minh họa các từ lệnh hoạt động trong mode 2.

4/ Kết hợp các chế độ hoạt động

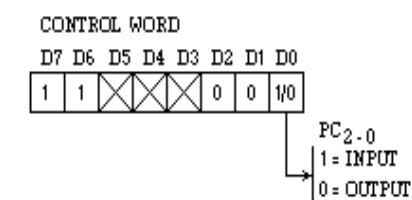
Ngoài việc hoạt động riêng rẽ theo từng chế độ, 8255 còn có khả năng hoạt động đồng thời kết hợp các chế độ khi không phải tất cả các bit trong cổng C được sử dụng để điều khiển hoặc dành cho trạng thái. Các bit còn lại có thể được sử dụng để thực hiện các chức năng sau :

Khi được lập trình là các đường vào tín hiệu :

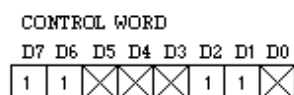
Tất cả các đường vào tín hiệu đều có thể được truy cập trong suốt quá trình đọc cổng C thông thường. Như trong hình vẽ minh họa sau :



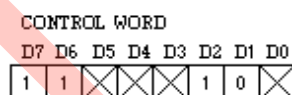
MODE 2 and MODE 0 (input)



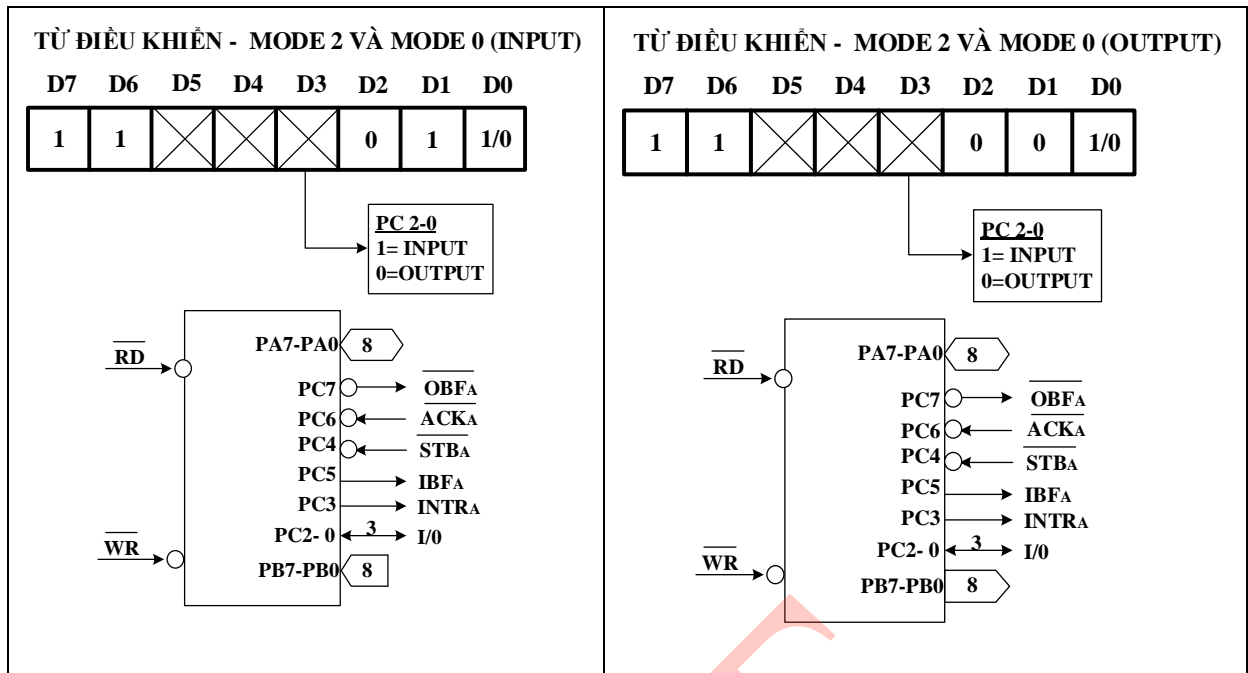
MODE 2 and MODE 0 (output)



MODE 2 and MODE 1 (input)



MODE 2 and MODE 1 (output)



Hình 2.21. Sơ đồ minh họa các chế độ hoạt động của từ lệnh điều khiển.

Khi được lập trình là các đường tín hiệu ra .

Các bit trong số các bit cao của cổng C (PC7- PC4) phải được truy cập một cách riêng rẽ bằng cách sử dụng chức năng xoá /lập bit.

Các bit trong số các bit thấp của cổng C có thể truy cập bằng chức năng xoá /lập bit hoặc dùng 3 bit tương ứng ghi ra cổng C.

Đọc trạng thái cổng C.

Trong mode 0, cổng C truyền dữ liệu tới hoặc từ thiết bị ngoại vi. Khi 8255 được lập trình hoạt động trong mode 1 hoặc mode 2, cổng C được sử dụng để tạo ra hoặc nhận tín hiệu hội thoại trao đổi với thiết bị ngoại vi.

Đọc nội dung của cổng C cho phép người lập trình kiểm tra trạng thái của các thiết bị ngoại vi và điều khiển quá trình trao đổi dữ liệu.

Không có lệnh đặc biệt nào dùng để đọc thông tin trạng thái từ cổng C mà chỉ có thao tác đọc thông thường thực hiện chức năng này.

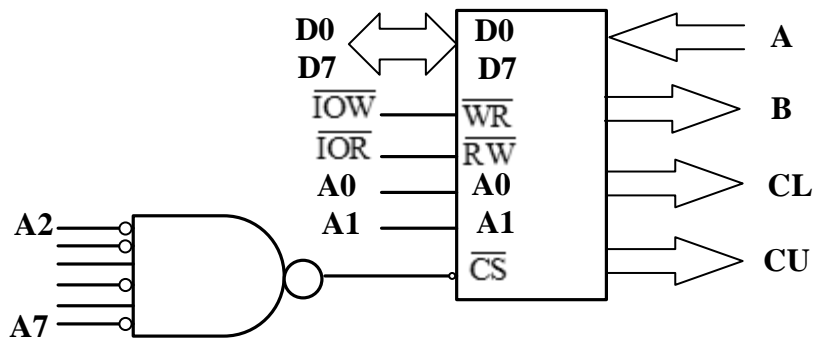
2.5. Ví dụ

Ví dụ 1: Cho hình sau với cấu hình portA là đầu vào, portB là đầu ra và tất cả các bit của portC là đầu ra.

a) Tìm địa chỉ port gán cho A, B, C và thanh ghi điều khiển.

b) Tìm byte (word) điều khiển cho cấu hình này.

c) Lập trình cho các port cho đầu vào dữ liệu từ port A và gửi nó cho 2 port B và C.



Hình 2.22. Hình vẽ ví dụ 1.

Giải:

a) Địa chỉ port gán cho A, B, C và thanh ghi điều khiển:

/CS	A1	A0	Address	Port
0101 00	0	0	50H	Port A
0101 00	0	1	51H	Port B
0101 00	1	0	52H	Port C
0101 00	1	1	53H	Thanh ghi điều khiển

b) Byte (word) điều khiển cho cấu hình này:

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	0	0	0	0
active	Mode 0		A vào	C ra		B ra	C ra

Byte điều khiển có giá trị = 90H hay 1001 0000

c) Chương trình:

```

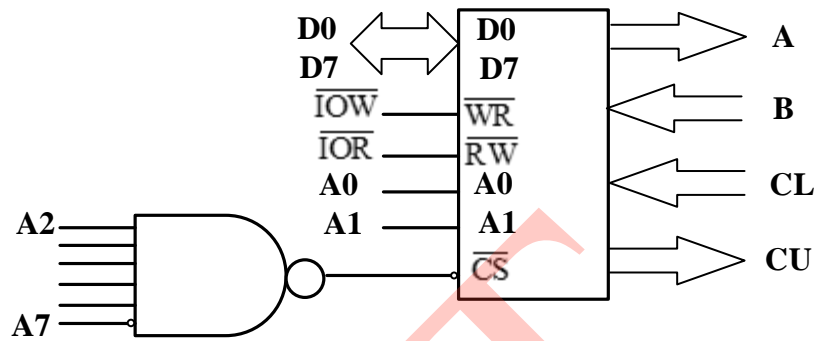
MOV  AL,90H ; byte điều khiển PA = in, PB = out, PC = out
OUT  53H,AL ; gửi đến thanh ghi điều khiển
IN   AL,50H ; gửi dữ liệu từ PA
OUT  51H,AL ; gửi đến PB
OUT  50H,AL ; gửi đến PC

```

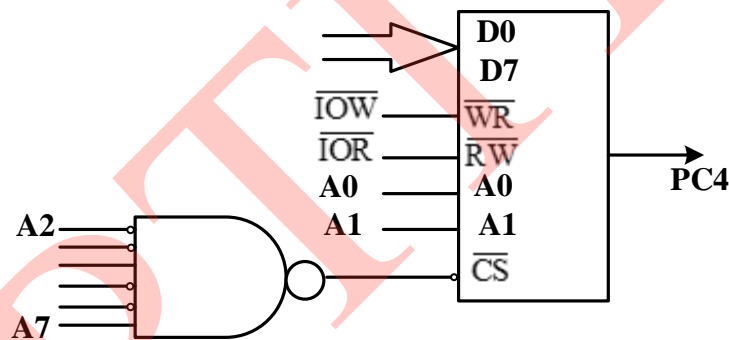
CÂU HỎI ÔN TẬP

1. Cho hình sau với cấu hình portA là đầu ra, portB là đầu vào và PC0-PC3 là đầu vào; PC4-PC7 là đầu ra.

- a) Tìm địa chỉ port gán cho A, B, C và thanh ghi điều khiển.
- b) Tìm byte (word) điều khiển cho cấu hình này.
- c) Lập trình cho 8255 để nhận dữ liệu từ port B và gửi sang port A. Đồng thời lấy dữ liệu từ PCL và gửi nó cho PCU.



2. Lập trình PC4 của 8255 để tạo xung có độ rộng 50ms với độ lấp đầy là 50%.



3. Lập chương trình cho 8255 để thực hiện đếm các số hệ mười từ 0 đến 255 (00h đến FFh) và viết chúng ra cổng A với thời gian 1 giây một lần đếm. (**Cho địa chỉ portA = 300**)

4. Lập chương trình cho 8255 để thực hiện đọc byte từ cổng A rồi viết byte này ra cổng C. Nếu một phím bất kỳ được nhấn hoặc tất cả các bit trên cổng A đều bằng 1 thì chương trình được kết thúc. (**Cho địa chỉ portA = 300**)

5. Viết chương trình cho 8255 để xuất ra cổng A hai số BCD tương ứng với các số thập phân từ 0 đến 99. (**Cho địa chỉ portA = 300**)

6. Viết chương trình cho 8255 để kiểm tra và thông báo ra màn hình máy vi tính giá trị của các bit trong cổng A ở chế độ 0. mode vào. (**Cho địa chỉ portA = 300**)

7. Một bộ đèn điều khiển giao thông đơn giản được lắp trên 3 bit của cổng C. Hãy viết chương trình cho 8255 để điều khiển các đèn theo quy luật sau:

- Đèn vàng luôn sáng sau đèn xanh, đèn đỏ luôn sáng sau đèn vàng và đèn xanh luôn sáng sau đèn đỏ.
- Mỗi đèn sáng trong 2 phút
- Các đèn sẽ sáng khi mức logic của các bit tương ứng bằng 1
- Hệ sẽ tắt hết các đèn và ngừng hoạt động khi bit PA7 bằng 1
(*Cho địa chỉ portA = 300*)

PTE

CHƯƠNG 3. GHÉP NỐI QUA CỔNG NỐI TIẾP

3.1. Giới thiệu chung về trao đổi dữ liệu nối tiếp

Ngày nay, ghép nối qua cổng nối tiếp là một kỹ thuật được sử dụng rộng rãi nhất bởi số lượng và chủng loại các thiết bị ngoại vi ghép nối qua cổng này đứng hàng đầu trong các khả năng ghép nối với máy tính. Qua cổng nối tiếp có thể ghép nối modem, chuột, bộ biến đổi A/D, D/A, các thiết bị đo lường ...

Ghép nối qua cổng nối tiếp là dữ liệu được truyền qua cổng kiểu nối tiếp nghĩa là tại một thời điểm chỉ có một bit được truyền dọc theo một đường dẫn. Đặc điểm này cho phép tạo ra sự khác biệt so với các cách ghép nối khác chẳng hạn cách truyền thông theo kiểu song song trong đó nhiều bit được gửi đồng thời. Ưu điểm chính của kỹ thuật này là sử dụng một đường truyền và một đường nhận cho nên việc điều khiển trở nên đơn giản.

Cổng này có tên là RS232 hoặc V.24. RS232 là tên một tiêu chuẩn quy định các đặc tính cho cổng nối tiếp, còn V.24 là tên của cổng này được áp dụng ở các nước Tây Âu.

So với các khả năng ghép nối khác tốc độ truyền qua cổng nối tiếp chậm, tốc độ thường sử dụng là 19600 bit/s/20m. Tốc độ truyền ở các modem đời mới nhất cũng chỉ đạt 56,6Kbit/s. Về sau có một số tiêu chuẩn nối tiếp khác ra đời như RS422, RS485 cho phép truyền với tốc độ cao hơn và khoảng cách dài hơn: ví dụ RS422 10Mbit/s/hàng ngàn km. Một số chuẩn khác còn cho phép sử dụng trên mạng máy tính.

3.1.1. Nguồn gốc

Ngày nay, ghép nối qua cổng nối tiếp là một kỹ thuật được sử dụng rộng rãi nhất bởi số lượng và chủng loại các thiết bị ngoại vi ghép nối qua cổng này đứng hàng đầu trong các khả năng ghép nối với máy tính. Qua cổng nối tiếp có thể ghép nối modem, chuột, bộ biến đổi A/D, D/A, các thiết bị đo lường ...

Ghép nối qua cổng nối tiếp là dữ liệu được truyền qua cổng kiểu nối tiếp nghĩa là tại một thời điểm chỉ có một bit được truyền dọc theo một đường dẫn. Đặc điểm này cho phép tạo ra sự khác biệt so với các cách ghép nối khác chẳng hạn cách truyền thông theo kiểu song song trong đó nhiều bit được gửi đồng thời. Ưu điểm chính của kỹ thuật này là sử dụng một đường truyền và một đường nhận cho nên việc điều khiển trở nên đơn giản.

Cổng này có tên là RS232 hoặc V.24. RS232 là tên một tiêu chuẩn quy định các đặc tính cho cổng nối tiếp, còn V.24 là tên của cổng này được áp dụng ở các nước Tây Âu.

Ghép nối qua cổng nối tiếp RS232 là một trong những kỹ thuật được sử dụng rộng rãi để ghép nối các thiết bị ngoại vi với máy tính. Nó là một chuẩn giao tiếp nối tiếp dùng định dạng không đồng bộ, kết nối nhiều nhất là 2 thiết bị, chiều dài kết nối lớn nhất cho phép để đảm bảo dữ liệu là 12.5 đến 25.4m, tốc độ 20kbit/s đôi khi là tốc độ 115kbit/s với một số thiết bị đặc biệt.

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

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

3.1.2. Ưu, nhược điểm của giao diện nối tiếp RS232

Ưu điểm của giao tiếp nối tiếp RS232:

- Khả năng chống nhiễu của các cổng nối tiếp cao.
- Thiết bị ngoại vi có thể tháo lắp ngay cả khi máy tính đang được cấp điện.
- Các mạch điện đơn giản có thể nhận được điện áp nguồn nuôi qua cổng nối tiếp.

+ Nhược điểm của giao tiếp nối tiếp RS232:

- Tốc độ truyền dữ liệu bị hạn chế. Ví dụ như với tốc độ 9600 baud cho phép truyền nhiều nhất là 960 byte mỗi giây. Khuôn dạng dữ liệu (Frame) cần phải được thiết lập như nhau ở cả hai bên gửi cũng như nhận.
- Chiều dài đường truyền hạn chế.

3.1.3. Đặc điểm của chuẩn RS232

- Trong chuẩn RS232 có mức giới hạn trên và dưới (logic 0 và 1) là $\pm 12V$. Hiện nay đang được cố định trở kháng tải trong phạm vi từ $3000\ \Omega$ - 7000Ω .
- Mức logic 1 có điện áp nằm trong khoảng $-3V$ đến $-12V$, mức logic 0 từ $+3V$ đến $12V$.
- Tốc độ truyền nhận dữ liệu cực đại là 100kbps (ngay nay có thể lớn hơn).
- Các lỗi vào phải có điện dung nhỏ hơn 2500pF.

- Trở kháng tải phải lớn hơn 3000Ω nhưng phải nhỏ hơn 7000Ω .
- Độ dài của cáp nối giữa máy tính và thiết bị ngoại vi ghép nối qua cổng nối tiếp RS232 không vượt qua 15m nếu chúng ta không sử dụng model.
- Các giá trị tốc độ truyền dữ liệu chuẩn : 50, 75, 110, 750, 300, 600, 1200, 2400, 4800, 9600, 19200, 28800, 38400....56600, 115200 bps.

3.1.4. Thủ tục trao đổi dữ liệu

+ *Quá trình truyền dữ liệu*

Truyền dữ liệu qua cổng nối tiếp RS232 được thực hiện không đồng bộ. Do vậy nên tại một thời điểm chỉ có một bit được truyền (1 kí tự). Bộ truyền gửi một bit bắt đầu (bit start) để thông báo cho bộ nhận biết một kí tự sẽ được gửi đến trong lần truyền bit tiếp theo. Bit này luôn bắt đầu bằng mức 0. Tiếp theo đó là các bit dữ liệu (bits data) được gửi dưới dạng mã ASCII(có thể là 5,6,7 hay 8 bit dữ liệu). Sau đó là một Parity bit (kiểm tra bit chẵn, lẻ hay không) và cuối cùng là bit dừng - bit stop có thể là 1; 1,5 hay 2 bit dừng.

+ *Tốc độ Baud*

Đây là một tham số đặc trưng của RS232. Tham số này chính là đặc trưng cho quá trình truyền dữ liệu qua cổng nối tiếp RS232 là tốc độ truyền nhận dữ liệu hay còn gọi là tốc độ bit. Tốc độ bit được định nghĩa là số bit truyền được trong thời gian 1 giây hay số bit truyền được trong thời gian 1 giây. Tốc độ bit này phải được thiết lập ở bên phát và bên nhận đều phải có tốc độ như nhau (Tốc độ giữa vi điều khiển và máy tính phải chung nhau 1 tốc độ truyền bit).

Ngoài tốc độ bit còn một tham số để mô tả tốc độ truyền là tốc độ Baud. Tốc độ Baud liên quan đến tốc độ mà phần tử mã hóa dữ liệu được sử dụng để diễn tả bit được truyền còn tốc độ bit thì phản ánh tốc độ thực tế mà các bit được truyền. Vì một phần tử báo hiệu sự mã hóa một bit nên khi đó hai tốc độ bit và tốc độ baud là phải đồng nhất.

Một số tốc độ Baud thường dùng: 50, 75, 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 28800, 38400, 56000, 115200 ... Trong thiết bị họ thường dùng tốc độ là 19200.

Khi sử dụng chuẩn nối tiếp RS232 thì yêu cầu khi sử dụng chuẩn là thời gian chuyển mức logic không vượt quá 4% thời gian truyền 1 bit. Do vậy, nếu tốc độ bit càng cao thì thời gian truyền 1 bit càng nhỏ thì thời gian chuyển mức logic càng phải nhỏ. Điều này làm giới hạn tốc Baud và khoảng cách truyền.

+ *Bit chẵn lẻ hay Parity bit*

Đây là bit kiểm tra lỗi trên đường truyền. Thực chất của quá trình kiểm tra lỗi khi truyền dữ liệu là bổ sung thêm dữ liệu được truyền để tìm ra hoặc sửa một số lỗi trong quá trình truyền. Do đó trong chuẩn RS232 sử dụng một kỹ thuật kiểm tra chẵn lẻ.

Một bit chẵn lẻ được bổ sung vào dữ liệu được truyền để cho thấy số lượng các bit "1" được gửi trong một khung truyền là chẵn hay lẻ.

Một Parity bit chỉ có thể tìm ra một số lẻ các lỗi như 1,3,5,7,9... Nếu như một bit chẵn được mắc lỗi thì Parity bit sẽ trùng giá trị với trường hợp không mắc lỗi vì thế không phát hiện ra lỗi. Do đó trong kỹ thuật mã hóa lỗi này không được sử dụng trong trường hợp có khả năng một vài bit bị mắc lỗi.

3.2. Một số chuẩn ghép nối

RS232 vẫn còn một số nhược điểm: Khoảng cách truyền còn hạn chế (15 m), tốc độ truyền chưa cao (100 Kbps). Vì vậy có một số tiêu chuẩn đã được ra đời nhằm khắc phục những nhược điểm do chuẩn RS232 để lại. Các tiêu chuẩn đó là RS422, RS423A, RS485, tất cả các chuẩn này đều bắt đầu bằng RS.

3.2.1. RS422

Là tiêu chuẩn đầu tiên được cải tiến từ tiêu chuẩn RS232C trong đó cả 2 đặc điểm khoảng cách và tốc độ truyền đều được cải tiến cụ thể: RS422 cho phép tăng tốc độ cỡ vài Mbit/s. Các vi mạch thông thường được sử dụng cho chuẩn RS232C như: MAX232 (Maxim), LT232 (Linear Technology) đều không thể sử dụng cho chuẩn này. Ngoài ra khoảng cách truyền cũng được cải thiện từ 15m -> 1200m, tất nhiên khi đó phải chấp nhận một tốc độ truyền vừa phải cỡ như 90 Kbps.

Cải tiến:

Về mặt bản chất vẫn truyền dữ liệu theo kiểu nối tiếp nhưng cách truyền thay đổi. Cụ thể là mức logic của tín hiệu không được tính theo đường Mas (0v) mà được tính theo điện áp vi sai (chênh lệch điện áp giữa 2 đường dẫn) khi đó bộ đệm đường dẫn của RS422 tạo ra một điện áp vi sai ~ 5v và truyền trên 2 sợi dây xoắn, sau đó ở bên nhận sẽ có một bộ phối hợp mức để đo vi sai điện áp để phân biệt giữa mức HIGH và mức LOW.

RS422 quy định một cặp tín hiệu được sử dụng để truyền chứ không phải một tín hiệu như trong RS232, cặp này bao gồm: Tín hiệu không đảo ký hiệu (A) và tín hiệu đảo (B). Chênh lệch điện áp giữa tín hiệu A và B khoảng 2-6V và sẽ xảy ra 2 trường hợp:

UA âm so với UB -> A có mức logic 1(mức đầy)

UA dương so với UB -> A có mức logic 0(mức trống)

Các giá trị điện áp này kết hợp với các thông số đã được quy định trở kháng đường cáp sẽ đặt ra một giới hạn thực tế cho độ dài của đường truyền cực đại. Tính toán cho thấy khoảng cách có thể đạt đến 1200m và đây là khoảng cách lí tưởng đối với rất nhiều ứng dụng. Trong trường hợp sử dụng đường truyền ngắn có thể nâng tốc độ truyền lên 10 Mbps. Nguyên nhân xa của việc cải thiện được khoảng cách truyền là tín hiệu nhiều khi ảnh hưởng đến một đường tín hiệu cũng đồng thời gây ảnh hưởng đến đường tín hiệu kia và như vậy khi so sánh với nhau sẽ bị bù trừ. Vì vậy để thoả mãn được điều kiện này thì hai đường tín hiệu phải nằm rất sát nhau. Trên thực tế người ta giải quyết bằng cách xoắn hai sợi dây lại với nhau. Khi đó không chỉ độ bền cơ học của cáp được tăng lên mà ảnh hưởng của nhiễu cũng được bù trừ thỏa đáng.

3.2.2. Chuẩn RS423A

Đặc điểm của chuẩn này là người ta sử dụng điện áp không cân bằng (không đối xứng). Nó chỉ sử dụng một đường dẫn để truyền giống như RS232 nhưng thông số điện đã được cải tiến để có tốc độ cao hơn và đường truyền dài hơn, cụ thể là giảm điện áp xuống chỉ còn 0-6V trong đó 0V~HIGH(1) 6V~LOW(0). Các tín hiệu của chuẩn RS232C thường được sử dụng để tạo ra tín hiệu điều khiển trên cùng một cáp truyền trong khi các tín hiệu của RS423A được sử dụng cho cả dữ liệu và cho cả việc phân chia khoảng thời gian.

3.2.3. Chuẩn RS485

Đây là kết quả trực tiếp của việc cải tiến RS422 theo hướng cho nhiều thành viên có thể tham gia vào cuộc truyền. Ta đã biết các chuẩn vừa trình bày ở trên chỉ cho phép hai thành viên tham gia truyền dữ liệu với nhau, trong trường hợp muốn đưa thêm thành viên thứ ba vào thì giải pháp về phần cứng rất phức tạp trong khi nhiều ứng dụng thực tế cần có khả năng truyền dữ liệu giữa một số thành viên với nhau. Có thể nói việc đưa ra chuẩn này thực chất là cho phép hình thành một bus dữ liệu trên đó có nhiều hơn hai thành viên tham gia, cho đến nay con số có thể đạt đến 32. Về mặt giải pháp kĩ thuật thì việc đưa chuẩn nối tiếp trở thành một bus đòi hỏi phải có sự hỗ trợ của các vi mạch 3 trạng thái (HIGH, LOW, điện trở cao). Mức logic 1 được ấn định tương tự RS422 cụ thể từ -1,5V~6V, mức logic 0 nằm trong khoảng 1,5V~6V.

Bộ đệm đường dẫn của chuẩn này cũng tạo ra một điện áp vi sai 5V trên hai đường dẫn /truyền dữ liệu và mức điện áp cũng được tính theo phương pháp vi sai. Chính nhờ cải tiến này mà việc ghép nối nhiều thiết bị đo lường (máy tính) với nhau trở nên hết sức đơn giản.

Ứng dụng của RS485

Một thí dụ sử dụng RS485 là theo dõi lượng hàng xuất nhập ở các kho xăng dầu. Các bể có thể chứa nhiều loại xăng khác nhau, tổng cộng có thể đến 32 bể chứa được quản lý đồng thời.

Người ta quản lý bằng cách mỗi bể đưa vào một đầu đo và gắn cho đầu đo đó một địa chỉ mà thực chất là một mã số. Các thông tin về đầu đo sẽ được chuyển qua RS485 vào máy tính. Tại bộ xử lý máy tính sẽ đọc các số liệu cùng với các mã số kèm theo. Trên cơ sở đó sẽ biết ngay là số lượng đo lường được gửi từ bể nào về, các số liệu này được đem kết hợp với kích thước hình học, nhiệt độ bể, nhiệt độ môi trường từ đó hình thành một cơ sở dữ liệu cho phép quản lý lượng xăng dầu đang được bảo quản trong kho, lượng xuất nhập trong ngày, tuần, tháng.

3.2.4. So sánh các chuẩn ghép nối

Giao diện nối tiếp RS232C(V.24) là một giao diện điện áp thuần túy trong đó các mức logic HIGH hoặc LOW đều là các mức điện áp nằm trong khoảng 3->12V hoặc -3->-12V là các mức logic đọc được và được tính so với một đường mas (GND) chung. Ưu điểm đặc biệt của nó là việc xử lý đơn giản theo nghĩa tìm lỗi hoặc viết phần mềm điều khiển rất nhanh.

Chính vì vậy việc ghép nối qua cổng này được áp dụng rất phổ biến và khả năng giao tiếp cũng rất lớn theo nghĩa là rất nhiều loại thiết bị ngoại vi của nhiều hãng khác nhau với nhiều mẫu mã khác nhau đều có thể ghép nối được với cổng này. Nhược điểm của giao diện này trước hết là khoảng cách truyền còn hạn chế và tốc độ truyền dữ liệu chưa cao.

Các giao diện RS422 và RS485 đều là các giao diện vi sai điện áp đối xứng. Trong trường hợp này việc truyền dữ liệu được tiến hành trên đường dẫn vi sai điện áp và do vậy hai đường dẫn được xoắn với nhau thành cặp khác hẳn với trường hợp của RS232 trong đó mức điện áp trên đường truyền và đường nhận được tính so với đường mas chung. Vì vậy, thông tin nhận được là từ điện áp vi sai giữa hai đường truyền và trên hai dây dẫn chứ không phải là từ giá trị điện áp tuyệt đối tính so với điện thế của đường mas chung. Do cách truyền này mà nhiễu điện từ nói chung không ảnh hưởng đến quá trình truyền dữ liệu. Kết quả là có thể tăng tốc độ truyền lên tới 10 Mbps.

Bảng 3.1. Bảng so sánh các chuẩn ghép nối.

	V24/RS232C	RS422	RS 485
Bản chất liên kết	Điểm-điểm	Điểm-điểm	bus
Loại giao diện	Điện áp ko đối	Điện áp đối xứng	Vi phân điện áp đối xứng với khả

	V24/RS232C	RS422	RS 485
	xúng		năng 3 trạng thái
Khả năng chống nhiễu	Thấp	Cao	Cao
Số bộ đệm cực đại	1	1	32
Số bộ nhận cực đại	1	1	32
Độ dài cực đại của đường truyền	15 m	1200 m	1200m
Tốc độ truyền cực đại	100 kbaud	10 MBaud	10 MBaud
Điện áp lỗi ra của bộ đệm: - Không tải. - Có tải	$\pm 15V$ $\pm 5V$	$\pm 5V$ $\pm 2V$	$\pm 5V$ $\pm 1.5V$
Điện trở lỗi ra của bộ đệm	$3k\Omega \div 7k\Omega$	100Ω	$54k\Omega \div 60k\Omega$
Điện trở lỗi vào của bộ đệm	$3k\Omega \div 7k\Omega$	$>4k\Omega$	$>12k\Omega$
Độ nhảy của bộ nhận	$\pm 3V$	$\pm 200mV$	$\pm 200mV$

Nhận xét:

Giữa RS422 và RS 485 có một sự khác nhau về cơ bản đó là: Chuẩn RS422 thực hiện chức năng liên kết điểm-điểm, trong khi bộ đệm của giao diện RS485 có thể được chuyển mạch trong trạng thái điện trở cao, vì vậy cho phép hình thành liên kết bus.

Trong một số trường hợp người ta phải hạn chế ảnh hưởng giữa các thành viên của khối ghép nối đặc biệt là trong trường hợp đường truyền dài, khi đó người ta phải thực hiện cách li về điện giữa hai thành viên bằng cách đưa vào bộ ghép nối quang. Trong bộ ghép nối quang bao gồm hai linh kiện: Diot phát quang LED (Light Emitting Diode) và Transistor quang. Khi có dòng điện chạy qua theo hướng thuận thì diot sẽ phát ra ánh sáng và dội vào cực gốc của transtor quang làm cho nó chuyển sang trạng thái dẫn (có dòng điện chạy qua). Như vậy, việc có dòng điện chạy qua diot bao giờ

cũng kéo theo dòng điện đi qua transistor, nhờ vậy mà người ta có thể dùng bộ ghép nối quang vào mục đích truyền dữ liệu. Ưu điểm của nó là giữa bên truyền và bên nhận được cách li về điện, vì vậy nếu có sự cố xảy ra một bên thì sẽ không kéo theo sự cố ở bên kia.

3.3. Lập trình cho cổng RS-232

Dữ liệu được gửi đi dọc theo đường truyền TxD (Transmit Data) và nhận bằng đường truyền RxD (Receive Data) với đường trở về đất chung.

Các đường dẫn khác được sử dụng để bắt tay (handshaking) và chia thành 2 loại vào và ra:

Các đường dẫn bắt tay lối vào:

RI (Ring Indicate): Báo chuông.

DSR (Data Set Ready): Dữ liệu sẵn sàng.

CTS (Clear to Send): Xóa để truyền

Các đường dẫn bắt tay lối ra:

RTS (Request to Send): Yêu cầu truyền.

DTR (Data Terminal Ready): Đầu cuối dữ liệu sẵn sàng.

Đóng vai trò chủ đạo trong quá trình truyền thông nối tiếp là IC 8250.

3.3.1. Bộ truyền nhận không đồng bộ vạn năng 8250

UART 8250 (Universal Asynchronous Receiver/Transmitter) đảm nhiệm các chức năng chính:

- Chuyển đổi tín hiệu song song từ CPU thành tín hiệu nối tiếp để truyền ra khỏi máy tính và chuyển đổi tín hiệu nối tiếp từ ngoại vi thành song song để chuyển tới CPU.
- Bổ sung các bit: Start, chẵn/lẻ và Stop vào mỗi ký tự để hình thành khung truyền, đồng thời tách các bit đó ra từ khung truyền nhận được.
- Duy trì trạng thái các bit riêng biệt được phát ra với tốc độ truyền dữ liệu thích hợp, tính toán các bit chẵn/lẻ trên từng ký tự nhận và truyền đồng thời thông báo cho các hệ thống biết bất kỳ sai sót đã được phát hiện.
- Thiết lập các tín hiệu bắt tay phần cứng thích hợp và thông báo trạng thái của các mạch đó.

UART 8250 là chip cơ bản, hiện nay các chip thông dụng là UART 16450, 16550A, 16750... Cấu trúc và chức năng của chúng dựa trên nền tảng của 8250

Lập trình cho 8250 được thực hiện bằng cách đọc và viết các thanh ghi của nó. Các thanh ghi đó là:

Base Address	Mode	Name	
+0 (DLAB=0)	Write	Transmitter Holding Buffer	THR
+0 (DLAB=0)	Read	Receiver Buffer	RBR
+0 (DLAB=1)	Rd/Wr	Divisor Latch Low Byte	DLL
+1 (DLAB=0)	Rd/Wr	Interrupt Enable Register	IER
+1 (DLAB=1)	Rd/Wr	Divisor Latch High Byte	DLM
+2	Read	Interrupt Identification Register	IIR
+2	Write	FIFO Control Register	FCR
+3	Rd/Wr	Line Control Register	LCR
+4	Rd/Wr	Modem Control Register	MCR
+5	Read	Line Status Register	LSR
+6	Read	Modem Status Register	MSR
+7	Rd/Wr	Scratch Register	SCR

PC có 4 cổng nối tiếp là COM1 - COM4 được phân biệt qua các vị trí địa chỉ trong vùng vào/ ra của PC và các số ngắt tương ứng. Địa chỉ đầu tiên của UART (của thanh ghi đệm truyền / nhận) gọi là địa chỉ cơ sở.

Thông thường địa chỉ cơ sở và IRQ được quy định nhờ các cầu nối (jumper) trên card vào/ra hoặc trên bản mạch chính.

Name	Port address	IRQ
COM 1	3F8h	IRQ4
COM 2	2F8h	IRQ3
COM 3	3E8h	IRQ4
COM 4	2E8h	IRQ3

Ta thấy COM 1, 3 và COM 2, 4 đều dùng chung một kênh ngắt, do vậy ở cùng 1 thời điểm chỉ có thể sử dụng 1 phần mềm cho ngắt của 1 trong 2 cổng và chỉ có thể sử dụng nhiều hơn 2 cổng nối tiếp khi trong chương trình không sử dụng ngắt

10 thanh ghi trong UART 8250 chia ra làm 3 loại:

- Thanh ghi điều khiển (Control Register): dùng để nhận và thực hiện các lệnh từ CPU.
- Thanh ghi trạng thái (Status Register): dùng để thông báo cho CPU biết về trạng thái của UART hay UART đang làm gì?
- Thanh ghi đệm (Buffer Register): dùng để giữ ký tự trong lúc truyền hoặc xử lý.

3.3.2. Các thanh ghi của UART 8250

Bảng 3.2. Bảng mô tả các thanh ghi của UART.

DLA B	A 2	A1	A0	Đọc/Ghi	Thanh ghi	Địa chỉ	
0	0	0	0	Đọc/Ghi	Đệm thu (RBR) và đệm phát (THR)	3F8 (2F8)	đ/c cơ sở
0	0	0	1	Đọc/Ghi	Cho phép yêu cầu ngắt (IER)	3F9 (2F9)	ĐCCS+1
1	0	0	0	Đọc/Ghi	Chốt chia phần thấp (LSB)	3F8 (2F8)	ĐCCS
1	0	0	1	Đọc	Chốt chia phần cao (MSB)	3F9 (2F9)	ĐCCS+1
x	0	1	0	Đọc/Ghi	Nhận dạng ngắt (IIR)	3FA (2FA)	ĐCCS+2
x	0	1	1	Đọc/Ghi	Điều khiển dòng (LCR)	3FB (2FB)	ĐCCS+3
x	1	0	0	Đọc/Ghi	Điều khiển MODEM (MC)	3FC (2FC)	ĐCCS+4
x	1	0	1	Đọc/Ghi	Trạng thái dòng (LSR)	3FD (2FD)	ĐCCS+5
x	1	1	0	Đọc/Ghi	Trạng thái modem (MSR)	3FE (2FE)	ĐCCS+6
x	1	1	1	Đọc/Ghi	Nhớ nháp (DM)	3FF (2FF)	ĐCCS+7

Mỗi thanh ghi trong 8250 tương ứng với một địa chỉ cổng, trong đó có hai thanh ghi đặc biệt, nó có chức năng có thể thay thế tùy thuộc giá trị bit DLAB (divisor latch access bit – Bit truy nhập chốt – DLAB là bit D7 của thanh ghi dạng số liệu)

- Nếu DLAB = 1 thì thanh ghi thực hiện chức năng chốt chia phần cao và phần thấp.
- Nếu DLAB = 0 thì hai thanh ghi dùng để đệm và nhận dữ liệu, và IER.

Thanh ghi giữ (holding register): Thanh ghi này thực chất là các bộ đệm được chuyên dùng để giữ một ký tự, ký tự này hoặc đã được nhận nhưng chưa được đọc hoặc đã gửi đến cổng nối tiếp nhưng chưa được truyền đi.

3.3.2.1. Thanh ghi đệm thu (Receiver Buffer Register – RBR)

Thanh ghi đếm thu ứng với DLAB = 0, thanh ghi có địa chỉ là 3F8h (COM1) hay 2F8h (COM2).

Khi 8250 nhận được ký tự qua chân SIN (chân 10), ký tự được chuyển vào thanh ghi dịch, tại đây nó được tháo khung (các bit Start, Stop, Parity) và nạp song song vào thanh ghi đếm thu. CPU chỉ đọc dữ liệu trong thanh ghi này vì nó thao tác với BUS dữ liệu song song tối thiểu 1 byte một lần.

3.3.2.2. Thanh ghi đếm phát (Transmitter Holding Register – THR)

Thanh ghi đếm phát ứng với DLAB = 0, thanh ghi có địa chỉ là 3F8h (COM1) hay 2F8h (COM2).

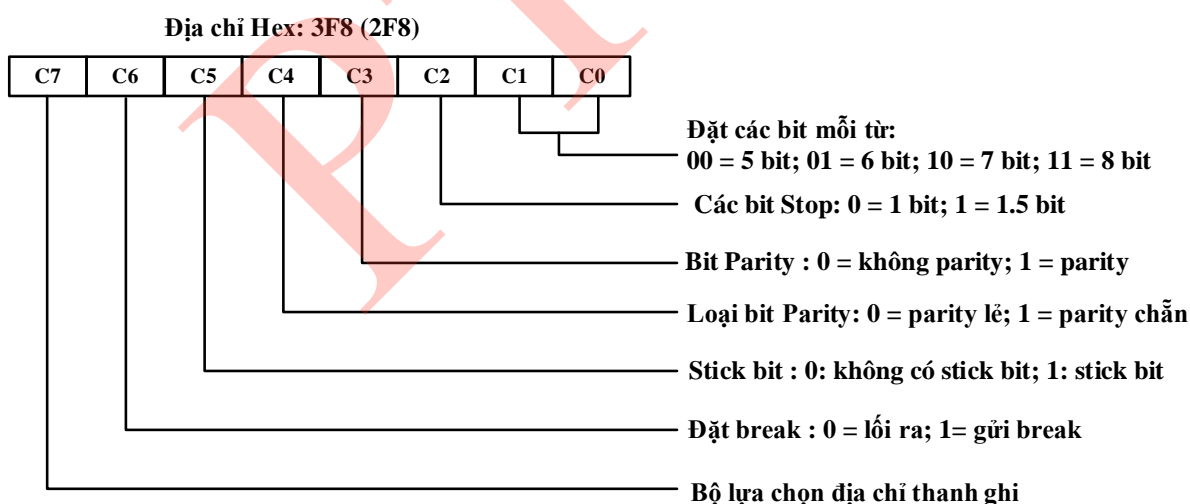
Ký tự cần phát đi phải được ghi từ thanh ghi vào và thanh ghi này nằm chờ (holding) cho đến khi ký tự trước nó được phát đi, sau đó nó đưa vào thanh ghi dịch của bộ phát. Tại thanh ghi dịch này nó được tháo khung và đưa ra từng bit một vào chân SOU (chân 11) của 8250. Dữ liệu tại đầu ra ở dạng nối tiếp.

3.3.2.3. Thanh ghi điều khiển đường truyền LCR (Line Control Register)-DLAB

Thanh ghi này lưu giữ các tham số được người lập trình thiết lập và xác định khuôn mẫu khung truyền của cuộc trao đổi thông tin.

Các thông tin được lưu giữ trong thanh ghi: số các bit dữ liệu, số lượng bit Stop và kiểu chẵn/lẻ.

Chức năng các bit của thanh ghi LCR:



Hình 3.1. Sơ đồ định dạng thanh ghi điều khiển đường truyền LCR

- Bit C0, C1 chỉ rõ số các bit dữ liệu trong từng ký tự được truyền.
- Bit C2 chỉ rõ số các bit Stop trong mỗi khung truyền.
- Bit C3 quy định có sử dụng bit chẵn/lẻ ko?
- Bit C4 Kiểu chẵn/lẻ đã được chọn.

- Bit C5: nếu như C3 và C5 = 1 thì khi bộ truyền xuất ra 1 ký tự, bộ nhận tại chỗ (local) sẽ phát hiện như là 1 giá trị logic 1.
- Bit C6 được quy định là bit dừng. Bit C6 = 1 thì nó buộc SOUT (Serial out hay TxD) = 0 cho đến khi 1 giá trị 0 được cất giữ vào bit 6. Nhờ bit này mà máy tính có thể báo hiệu cho thiết bị đầu cuối biết là đã được nối như 1 phần của hệ thống truyền thông.
- Bit C7 = 1 để truy nhập các chốt số chia. Các chốt này là những thanh ghi cất giữ số chia đối với tín hiệu clock, số này quy định tốc độ baud của hệ truyền thông nối tiếp. Mỗi lần tốc độ baud được đặt lại thì bit C7 = 0.

3.3.2.4. Thanh ghi tốc độ baud

Thanh này không lưu giữ giá trị của tốc độ baud mà chỉ chứa số chia 16 bit (2 thanh ghi 8 bit), số này suy ra tốc độ baud.

Tốc độ baud được đặt bằng cách nạp số chia 16 bit đặt ở địa chỉ bộ đệm TX/RX.

Giá trị cất giữ trên 2 thanh ghi này phụ thuộc vào tần số của bộ cộng hưởng thạch anh của 8250 (chân 16, 17).

Tần số là 18 432MHz; 24 576 MHz; 3072MHz

Để đạt tốc độ truyền mong muốn, giá trị tính toán được đặt trong hai thanh ghi được gọi là thanh ghi chốt chia tốc độ được tính theo công thức:

$$\text{Bộ chia} = \frac{\text{Tần số nhịp chuẩn}}{16 \times \text{tốc độ baud mong muốn}}$$

Tần số nhịp chuẩn do đồng hồ thạch anh của 8250 phát ra, có giá trị 18432 MHz.

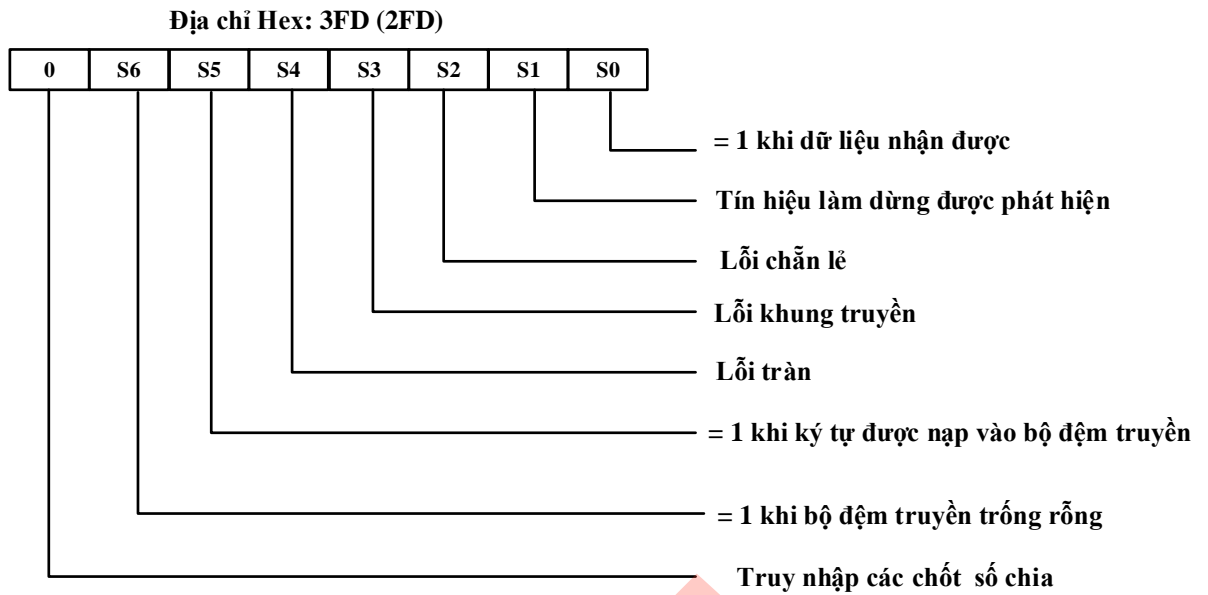
Tốc độ tối đa là 115200 baud. Dưới đây là bảng chuyển đổi giữa giá trị baud và số chia.

Baud	Số chia	Baud	Số chia
50	0900	1200	0060
110	0417	2400	0030
150	0300	4800	0018
300	0180	9600	0018
600	00C0	11520	0001

Nhận xét: giá trị trong thanh chốt chia cần 1.5 byte để lưu tất cả. Để thuận tiện người ta sẽ dùng 2 byte và ghi vào hai thanh ghi là LSB và MSB.

3.3.2.5. Thanh ghi trạng thái đường truyền LSR- line Status Register

Gồm 8 bit chứa thông tin về quá trình truyền dữ liệu qua cổng nối tiếp cần cung cấp cho bộ VXL.



Hình 3.2. Sơ đồ định dạng thanh ghi trạng thái đường truyền LSR

Bit S0: thông báo cho biết dữ liệu đã nhận được. Bit 0 = 1: dữ liệu đã được nhận và sẵn sàng để bộ xử lý đọc.

Bit S1 = 1: ký tự nhận trước đó đã bị mất vì nó ko được đọc trước khi 1 ký tự mới được nhận nên ký tự mới đã ghi đè lên ký tự trước.

Bit S2 = 1: ký tự được nhận có tính chẵn lẻ sai. Khi thanh ghi trạng thái đường truyền LSR được đọc thì bit này = 0.

Bit S3 = 1: có lỗi khung truyền, (khi ký tự đã nhận có bit Stop ko hợp lệ).

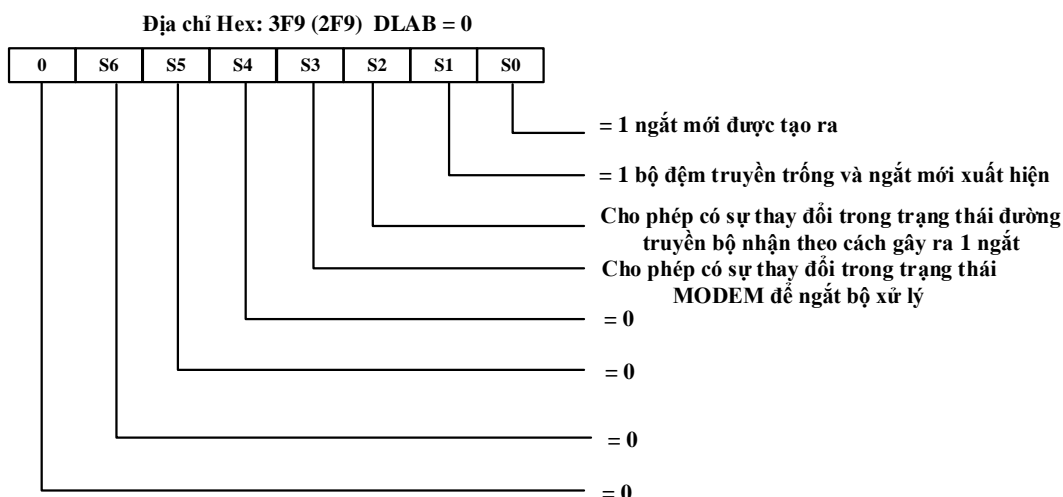
Bit S4 là bit gián đoạn ngắt (break interrupt bit)

Khi đường dây ở trạng thái Mark (ứng với mức logic 1). Bit Start sẽ có mức logic 0. Sau đó từng bit data sẽ được gửi lên đường dây. Bit Stop (logic 1) sau đó được gán vào để kết thúc truyền. Sau bit Stop là bit logic 0, điều này có nghĩa đây là bit Start của một dữ liệu mới.

Nếu không có thêm dữ liệu truyền đi, đường dây phải ở trạng thái rỗi tương ứng mức logic 1. Do đó, nếu đường dây giữ trạng thái 0 sau bit Stop trong một thời gian bằng độ dài 1 từ thì tín hiệu Break sẽ được tạo ra.

Tín hiệu Break là tín hiệu báo cho CPU hay MODEM biết ký tự truyền đã xòn và đang tạm ngắt việc truyền.

3.3.2.6. Thanh ghi cho phép ngắt- IER : Interrupt Enable Register



Hình 3.3. Sơ đồ định dạng thanh ghi cho phép ngắt IER

3.3.2.7. Thanh ghi nhận dạng ngắt – IIR: Interrupt Identification Register

Thanh này chứa đựng mã, nhận dạng điều kiện (ngắt) nào đang yêu cầu chú ý. Giữa các ngắt có phân cấp mức độ ưu tiên.

Các bit D7÷D3 luôn đặt bằng 0. D2, D1: mã hóa các yêu cầu ngắt có mức ưu tiên cao nhất đang chờ phục vụ.

D1	D2	Mức ưu tiên	Nhận dạng ngắt
0	0	4	Trạng thái MODEM → chương trình đọc trạng thái MODEM
0	1	3	Thanh ghi đệm phát rỗng
1	0	2	Thanh ghi đệm thu đầy, số liệu đã sẵn sàng
1	1	1	Trạng thái dòng đã thay đổi

D0 = 0: có yêu cầu ngắt.

D0 = 1: không có yêu cầu ngắt.

Từ các nhận dạng ngắt trên, VXL sẽ chuyển đến các địa chỉ chứa các chương trình con phục vụ ngắt. VXL sẽ xóa các bit D0, D1, D2 về giá trị 0 sau các thao tác:

- Đọc thanh ghi trạng thái chuỗi.
- Đọc dữ liệu từ thanh ghi đệm thu.
- Ghi vào bộ phát hoặc đọc thanh ghi nhận diện ngắt.
- Đọc trạng thái thanh ghi thao tác MODEM (RS232-C)

3.3.2.8. Thanh ghi điều khiển MODEM (MODEM Control Register - MCR)

Địa chỉ thanh ghi 3FCh. Thanh ghi này còn gọi là thanh ghi điều khiển các tín hiệu ra của MODEM vì nó cho phép điều khiển các tín hiệu tại các chân DTR (chân 33) và chân DTS (chân 32) của 8250.

Bảng 3.3. Bảng mô tả chức năng của thanh ghi điều khiển MODEM

Bit	Chức năng
7-5	Dữ trữ, luôn được thiết lập về 0
4	= 1: cho phép 8250 làm việc ở chế độ nối vòng cục bộ để kiểm tra các chức năng. = 0: làm việc ở chế độ thường.
3	= 1: kích hoạt OUT2
2	= 1: kích hoạt OUT1
1	= 1: tín hiệu tại chân /RTS của 8250 được chuyển về mức 0, máy tính phát chuyển tín hiệu /RTS tới MODEM để báo hiệu nó đã sẵn sàng phát dữ liệu.
0	=1: tín hiệu tại chân /DTR của 8250 được chuyển về mức 0, máy tính phát chuyển tín hiệu /DTR tới MODEM (hoặc máy tính) để báo hiệu nó đã sẵn sàng làm việc (tín hiệu gửi tới chân số 4 của DB9 hay chân 20 của DB25).

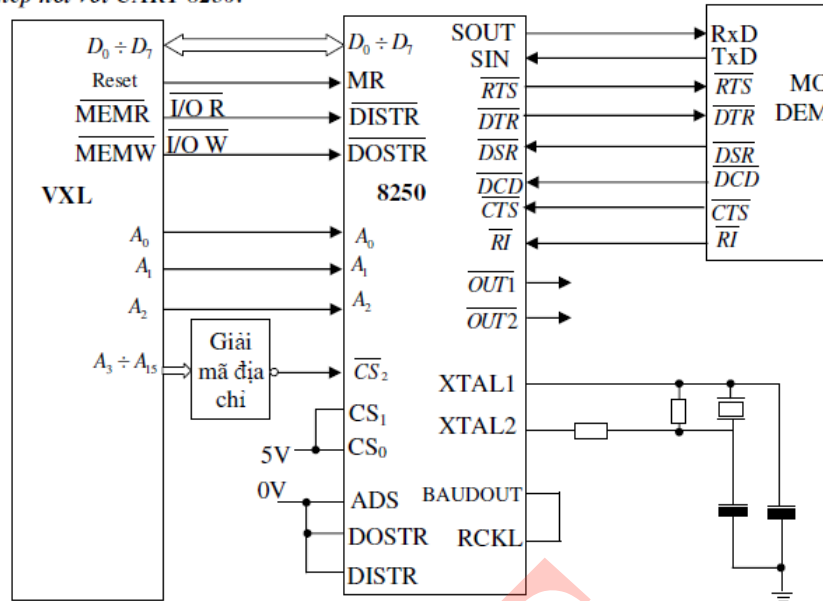
3.3.2.9. Thanh ghi trạng thái MODEM (MODEM Status Register).

Địa chỉ thanh ghi 3FEh. Thanh ghi này còn được gọi là thanh ghi trạng thái vào từ RS232C vì nó cho biết trạng thái hiện thời của các tín hiệu điều khiển MODEM.

Bảng 3.4. Bảng mô tả chức năng của thanh ghi trạng thái MODEM

Bit	Chức năng
7	= 1: DCD đang hoạt động.
6	= 1: RI đang hoạt động.
5	= 1: MODEM (hoặc máy tính) gửi DSR (Data Set Ready) về máy tính để cho biết sẵn sàng nhận dữ liệu (qua chân 6 của DB9 và DB25).
4	= 1: MODEM (hoặc máy tính) gửi CTS (Clear To Send) về máy tính để cho biết sẵn sàng nhận dữ liệu (qua chân 8 của DB9 và chân 5 của DB25).
3	= 1: D7 vừa đổi trạng thái.
2	= 1: D6 vừa đổi trạng thái.
1	= 1: D5 vừa đổi trạng thái.
0	= 1: D4 vừa đổi trạng thái.

Ghép nối với UART 8250:



Hình 3.4. Sơ đồ ghép nối với 8250

3.3.3. Lập trình cho cổng RS-232

Lưu đồ thuật toán:

- Khởi phát vi mạch theo trình tự sau:
- Ghi vào thanh ghi dạng số liệu theo:
 - o D7 = DLAB = 1 để chuẩn bị ghi chốt cho bộ chia tốc độ.
 - o Khung tin với số bit Stop.
 - o Số bit của tin.
- Ghi giá trị bộ chia tần số vào các thanh ghi chốt chia LSB, MSB tùy theo tốc độ baud.
- Ghi các phép yêu cầu ngắt vào thanh ghi yêu cầu ngắt.

Phát số liệu nối tiếp:

- Đọc thanh ghi nhận dạng ngắt để biết bộ đệm rỗng, có thể phát tin.
- Ghi vào thanh ghi điều khiển MODEM để kiểm tra các bit DSR (bit D0, D1) điều khiển MODEM chuẩn bị phát.
- Đọc thanh ghi trạng thái MODEM để kiểm tra các bit DSR (D5), RI (D6), DCD (D7) xem đã chuẩn bị phát chưa? (giá trị chúng = 1).
- Đọc thanh ghi trạng thái dòng để xem có sai số không và hai thanh truyền và đệm có trống không? (để đưa tin ra)
- Ghi tin ra vào thanh ghi đếm phát từ máy tính.
- Ghi lệnh RTS (D1 = 1) vào thanh ghi điều khiển MODEM để điều khiển phát số liệu.

Nhận số liệu nối tiếp.

- Ghi lệnh DTR ($D0 = 1$) vào thanh ghi điều khiển MODEM.
- Đọc trạng thái MODEM DSR, RI, DCD ở thanh ghi trạng thái MODEM.
- Đọc thanh ghi trạng thái dòng để biết đã có số liệu thu chưa ($D0 = 1$); hoặc đọc thanh ghi nhận diện ngắt để biết thêm số liệu thu.
- Đọc số liệu vào từ thanh ghi đệm số liệu vào.

3.3.3.1. Thủ tục trao đổi dữ liệu nối tiếp

Thủ tục phát TxD (Tranceiver Data):

Thiết bị đầu cuối gửi tín hiệu /DTR (Data Terminal Ready) ở mức 0 đến MODEM báo hiệu đã sẵn sàng. MODEM trả lời thiết bị đầu cuối bằng tín hiệu /DSR (Date Set Ready) ở mức 0.

Nếu thiết bị đầu cuối có 1 ký tự sẵn sàng gửi đi nó gửi /RST (Request To Send) đến MODEM. MODEM gửi tín hiệu CD (Carrier Detect) cho thiết bị đầu cuối để báo hiệu liên lạc đã thông.

Khi MODEM sẵn sàng phát số liệu trên đường dây thì nó phát xung và tín hiệu /CTS (Clear To Send) đến thiết bị đầu cuối. Thiết bị đầu cuối gửi các ký tự số liệu cho MODEM

Sau đó thiết bị đầu cuối nâng tín hiệu /RST lên mức cao để báo cho MODEM biết là đã phát xong. MODEM trả lời bằng cách gửi tín hiệu /CTS ở mức cao để báo đã hoàn thành việc truyền tin.

Thủ tục nhận dữ liệu RxD (Receiver Data).

Thiết bị đầu cuối phía thu gửi /DTR mức thấp cho MODEM báo nó đã sẵn sàng nhận. MODEM thu gửi tín hiệu trả lời bằng /DSR.

MODEM thu nhận tín hiệu CD từ đường dây và kích gửi tín hiệu nhịp MODEM cho thiết bị đầu cuối. Thiết bị đầu cuối phát tín hiệu /RTS.

MODEM gửi tín hiệu /CTS mức thấp cho thiết bị đầu cuối để báo MODEM sẵn sàng nhận tin. MODEM nhận TxD trên đường dây, rồi điều chế thành bit 0/1 và lần lượt gửi chuỗi tín hiệu RxD cho thiết bị đầu cuối.

Khi thu xong, thiết bị đầu cuối nâng tín hiệu /RTS lên cao báo cho MODEM là đã thu xong. MODEM trả lời bằng việc nâng tín hiệu /CTS lên cao để báo đã nhận xong.

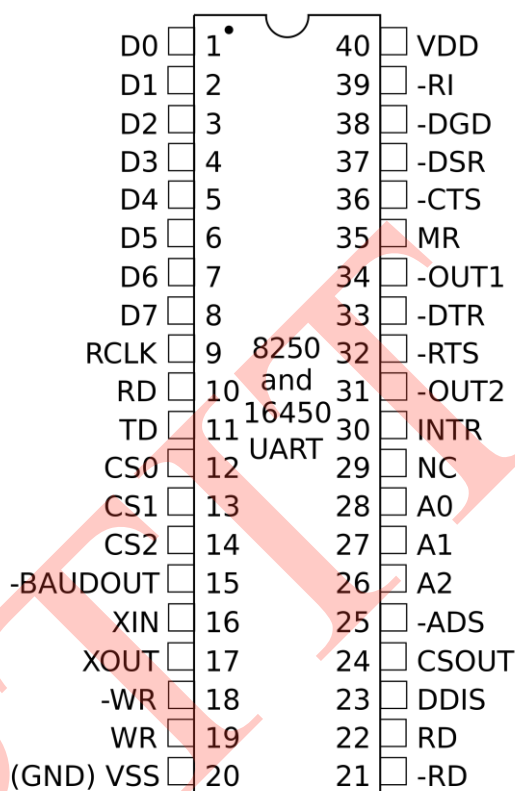
3.3.3.2. Mạch thu phát không đồng bộ vạn năng UART 8250

Mạch UART 8250 (Universal Asynchronous Receiver/Transmitter) được lắp ráp trong các máy IBM XT. UART là IC thu phát không đồng bộ có thể lập trình được, nó thực hiện chức năng giao tiếp giữa máy tính và ngoại vi.

Trong các máy tính PC/AT hiện nay sử dụng UART 16450. Còn trong các máy tính AT đời mới sử dụng UART 16550 vì 8250 và 16450 không đạt đến tốc độ 115200 baud.

Thực ra tốc độ này còn ứng dụng tốt trong kỹ thuật truyền tin trên mạng, còn trong các thiết bị công nghiệp ngoại vi nói chung chỉ cần tối đa 19200 baud là được, thông dụng nhất là 9600 baud

Sơ đồ khối và chức năng



Hình 3.5. Sơ đồ chân của UART 8250

Chức năng các chân gồm:

Các chân từ 1÷8 (D0÷D7): là các chân dữ liệu.

Chân 9 (RCLK): nhận xung đồng hồ đưa ra từ bộ tạo tốc độ.

Chân 10 (SIN): nhận dữ liệu tuần tự đi tới từ cổng RS232.

Chân 11 (SOUT): gửi dữ liệu tuần tự sang cổng RS232.

Chân 12, 13, 14 (CS0÷CS2): Chọn chip.

Chân 15 (/BOUDOUT): Lỗi ra của bộ tạo tốc độ.

Chân 16, 17 (XTAL1, XTAL2): Lấy xung Clock từ bộ phát nhịp bằng thạch anh tới với tần số 3072MHz.

Chân 18 (DOSTR): đầu viết I/O .

Chân 19 (/DOSTR): đảo của đầu viết I/O .

Chân 20 (Vss): đất.

Chân 21 (DISTR): đầu đọc I/O.

Chân 22 (/DISTR): đảo của đầu đọc I/O.

Chân 26, 27, 28 (A0÷A2): nhận địa chỉ truyền tới để giải mã lệnh.

Chân 30 (INTRPT): Interrupt Control Logic (Logic điều khiển ngắt).

Chân 32 (/RTS): Request To Sent (yêu cầu gửi).

Chân 33 (/DTR): Data Terminal Ready (dữ liệu sẵn sàng).

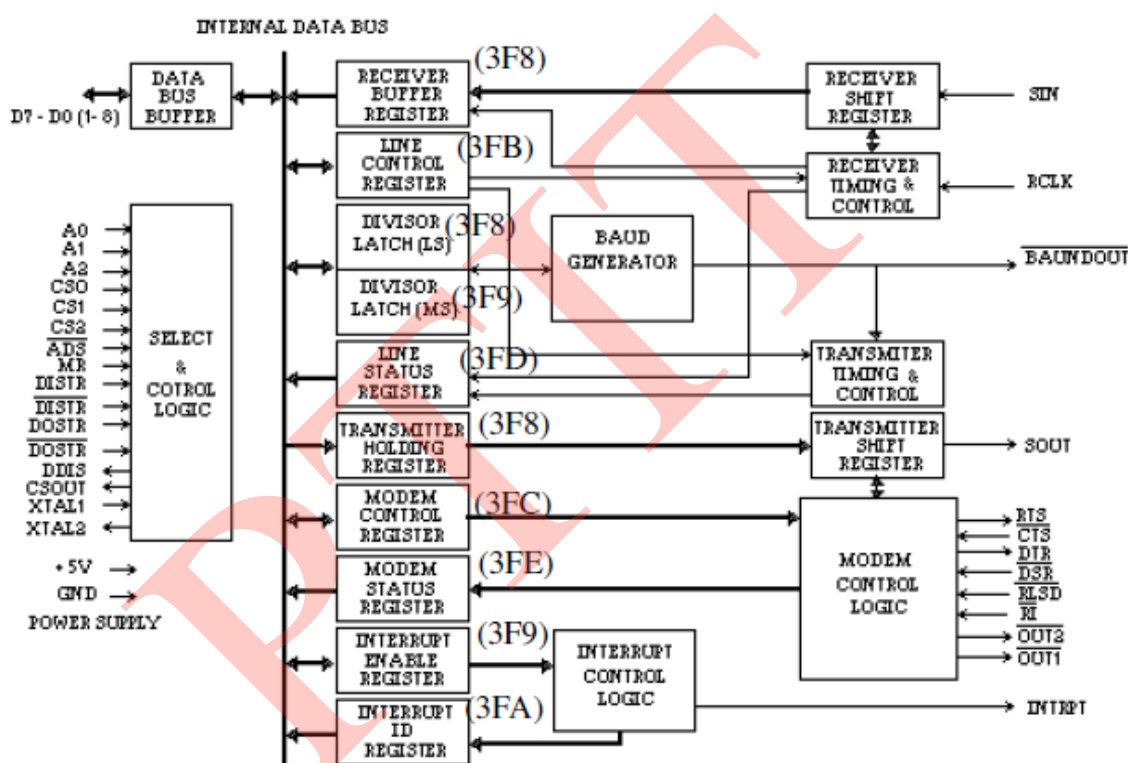
Chân 35 (MR): Reset.

Chân 36 (/CTS): Clear To Send.

Chân 37 (/DSR): Data Set Ready.

Chân 39 (/RI): Ring Indicate.

Chân 40 (VDo): Nối với nguồn 1 chiều 5V.



Hình 3.6. Sơ đồ khối của 8250.

8250 có 11 thanh ghi nội truy nhập theo bảng sau:

Bảng 3.5. Các lệnh của 8250.

DLAB	A2	A1	A0	Đọc/Ghi	Thanh ghi
0	0	0	0	Đọc/Ghi	Đệm thu (RBR) và đệm phát (THR)
0	0	0	1	Đọc/Ghi	Cho phép yêu cầu ngắt (IER)
1	0	0	0	Đọc/Ghi	Chốt chia phần thấp (LSB)

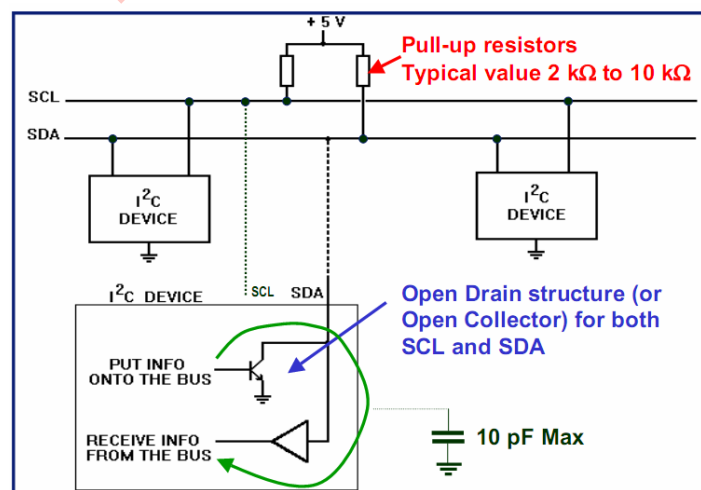
DLAB	A2	A1	A0	Đọc/Ghi	Thanh ghi
1	0	0	1	Đọc	Chốt chia phần cao (MSB)
x	0	1	0	Đọc/Ghi	Nhận dạng ngắt (IIR)
x	0	1	1	Đọc/Ghi	Điều khiển dòng (LCR)
x	1	0	0	Đọc/Ghi	Điều khiển MODEM (MC)
x	1	0	1	Đọc/Ghi	Trạng thái dòng (LSR)
x	1	1	0	Đọc/Ghi	Trạng thái dòng (MSR)
x	1	1	1	Đọc/Ghi	Nhớ nháp (DM)

3.3.4. Giao tiếp Two-Wire I2C

3.3.4.1. Giới thiệu

TWI (Two-Wire Serial Intereafce) là một module truyền thông nối tiếp đồng bộ trên các chip AVR dựa trên chuẩn truyền thông I2C. I2C là viết tắt của từ Inter-Integrated Circuit là một chuẩn truyền thông do hãng điện tử Philips Semiconductor sáng lập và xây dựng thành chuẩn năm 1990. Phiên bản mới nhất của I2C là V3.0 phát hành năm 2007.

TWI (I2C) là một truyền thông nối tiếp đa chip chủ (tạm dịch của cụm từ multi-master serial computer bus). Khái niệm “multi-master” (“đa chip chủ”) được hiểu là trong trên cùng một bus có thể có nhiều hơn một thiết bị làm Master, đồng thời một Slave có thể trở thành một Master nếu nó có khả năng. Ví dụ trong một mạng TWI của nhiều AVR kết nối với nhau, bất kỳ một AVR nào đều có thể trở thành Master ở một thời điểm nào đó. Tuy nhiên nếu một mạng dùng một AVR điều khiển các chip nhớ (như EEPROM AT24C1024 chẳng hạn) thì khái niệm “multi-master” không tồn tại vì các chip nhớ được thiết kế sẵn là Slave, không có khả năng trở thành Master.



Hình 3.7. Một mạng TWI

TWI (I2C) được thực hiện trên 2 đường SDA (Serial DATA) và SCL (Serial Clock) trong đó SDA là đường truyền/nhận dữ liệu và SCL là đường xung nhịp.

Nhiệm vụ vai trò của Master và Slave:

+ Master

- Giữ vai trò điều khiển Bus I2C.
- Tạo xung Clock (SCL) trong suốt quá trình giao tiếp.
- Tạo các tín hiệu Start bắt đầu quá trình truy xuất.
- Phát địa chỉ của thiết bị Slave cần truy xuất.
- Gửi tín hiệu R/W tới Slave.
- Truyền dữ liệu tới thiết bị Slave.
- Nhận dữ liệu từ Slave
- Tạo tín hiệu Not-ACK khi kết thúc nhận từ Slave.
- Tạo tín hiệu Stop kết thúc quá trình truy xuất.

+ Slave

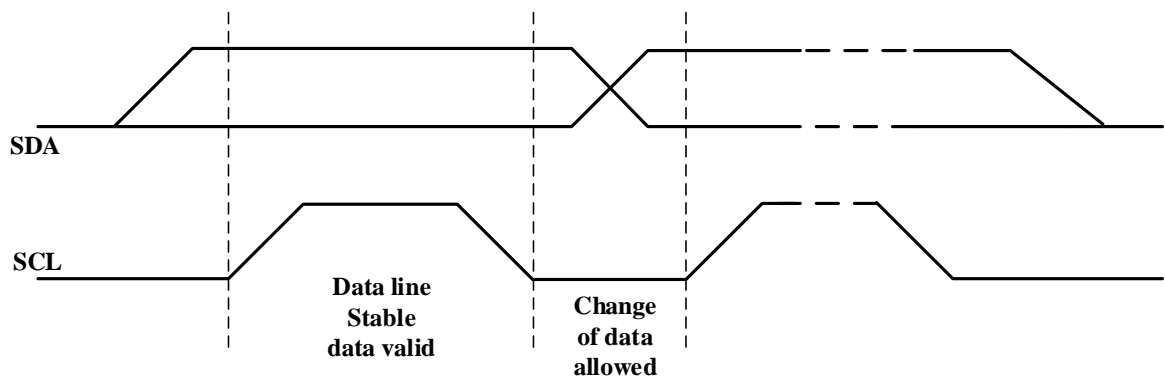
- Nhận địa chỉ và bit R/W từ Master (Chỉ “Response” khi đúng địa chỉ của Slave) .
- Nhận dữ liệu từ Master gửi bit ACK sau mỗi 8 Clock.
- Truyền dữ liệu tới Master, chờ ACK từ Master để tiếp tục gửi.
- Thực hiện chức năng chuyên dụng của khối thiết bị ngoại vi (RAM, EEPROM, ADC,DAC ngoài...).

Mỗi thiết bị ngoại vi tham gia vào bus I2C đều có một địa chỉ duy nhất, nhằm phân biệt giữa các thiết bị với nhau. Độ dài địa chỉ là 7 bit, điều đó có nghĩa là trên một bus I2C có thể phân biệt tối đa 128 thiết bị. Khi thiết bị Master muốn giao tiếp với ngoại vi nào trên bus I2C, nó sẽ gửi 7 bit địa chỉ của thiết bị đó ra bus ngay sau xung START. Byte đầu tiên được gửi sẽ bao gồm 7 bit địa chỉ và một bit thứ 8 điều khiển hướng truyền.

Mỗi một thiết bị ngoại vi sẽ có một địa chỉ riêng do nhà sản xuất quy định. Địa chỉ đó có thể là cố định hay thay đổi. Riêng bit điều khiển hướng sẽ quy định chiều truyền dữ liệu. Nếu bit này bằng “0” có nghĩa là byte dữ liệu tiếp theo sau sẽ được truyền từ Master đến Slave, còn ngược lại nếu bằng “1” thì các byte theo sau byte đầu tiên sẽ là dữ liệu từ Slave gửi đến Master. Việc thiết lập giá trị cho bit này do Master thi hành, Slave sẽ tùy theo giá trị đó mà có sự phản hồi tương ứng đến Master.

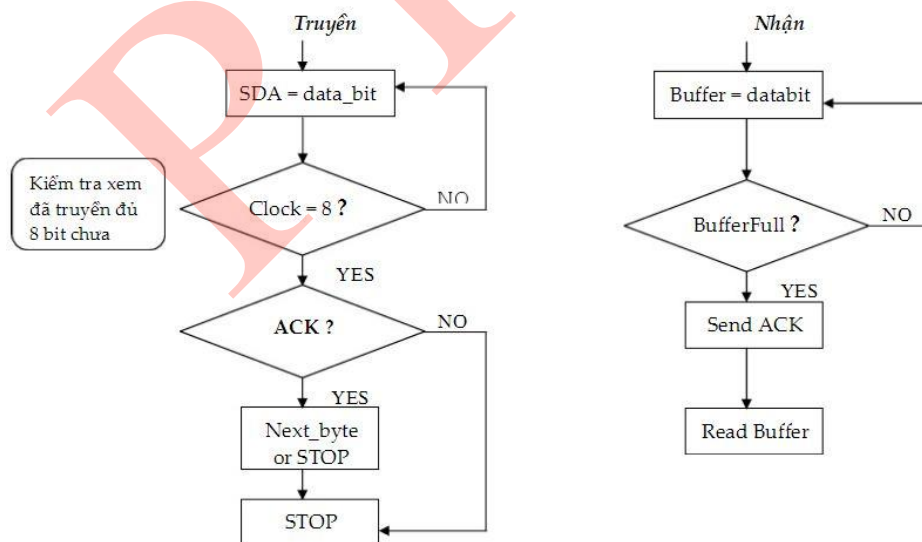
3.3.4.2. Quá trình truyền nhận dữ liệu

Dữ liệu được truyền trên bus I2C theo từng bit, bit dữ liệu được truyền đi tại mỗi sườn dương của xung đồng hồ trên dây SCL, quá trình thay đổi bit dữ liệu xảy ra khi SCL đang ở mức thấp.



Hình 3.8. Quá trình truyền 1 bit dữ liệu I2C

Mỗi byte dữ liệu được truyền có độ dài là 8 bits. Số lượng byte có thể truyền trong một lần là không hạn chế. Mỗi byte được truyền đi theo sau là một bit ACK để báo hiệu đã nhận dữ liệu. Bit có trọng số cao nhất (MSB) sẽ được truyền đi đầu tiên, các bit sẽ được truyền đi lần lượt. Sau 8 xung clock trên dây SCL, 8 bit dữ liệu đã được truyền đi. Lúc này thiết bị nhận, sau khi đã nhận đủ 8 bit dữ liệu sẽ kéo SDA xuống mức thấp tạo một xung ACK ứng với xung clock thứ 9 trên dây SDA để báo hiệu đã nhận đủ 8 bit. Thiết bị truyền khi nhận được bit ACK sẽ tiếp tục thực hiện quá trình truyền hoặc kết thúc.



Hình 3.9. Lưu đồ thuật toán quá trình truyền nhận dữ liệu I2C

Một byte truyền đi có kèm theo bit ACK là điều kiện bắt buộc, nhằm đảm bảo cho quá trình truyền nhận được diễn ra chính xác. Khi không nhận được đúng địa chỉ hay khi muốn kết thúc quá trình giao tiếp, thiết bị nhận sẽ gửi một xung Not-ACK

(SDA ở mức cao) để báo cho thiết bị chủ biết, thiết bị Master sẽ tạo xung STOP để kết thúc hay lặp lại một xung START để bắt đầu quá trình mới.

3.3.5. Giao tiếp SPI

3.3.5.1. Giới thiệu

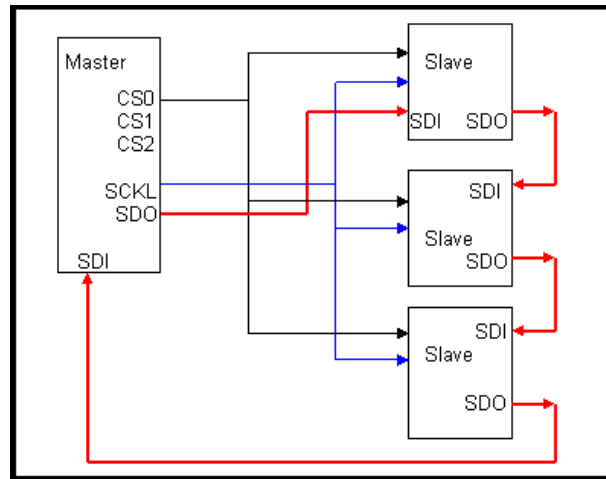
SPI (Serial Peripheral Bus) là một chuẩn truyền thông nối tiếp tốc độ cao do hãng Motorola đề xuất. Đây là kiểu truyền thông Master-Slave, trong đó có 1 chip Master điều phối quá trình truyền thông và các chip Slaves được điều khiển bởi Master vì thế truyền thông chỉ xảy ra giữa Master và Slave. SPI là một cách truyền song công (full - duplex) nghĩa là tại cùng một thời điểm quá trình truyền và nhận có thể xảy ra đồng thời. SPI đôi khi được gọi là chuẩn truyền thông “4 dây” vì có 4 đường giao tiếp trong chuẩn này đó là SCK (Serial Clock), MISO (Master Input Slave Output), MOSI (Master Output Slave Input) và SS (Slave Select). Hình 3.10 thể hiện một kết SPI giữa một chip Master và 3 chip Slave thông qua 4 đường.

SCK: Xung giữ nhịp cho giao tiếp SPI, vì SPI là chuẩn truyền đồng bộ nên cần 1 đường giữ nhịp, mỗi nhịp trên chân SCK báo 1 bit dữ liệu đến hoặc đi. Đây là điểm khác biệt với truyền thông không đồng bộ mà chúng ta đã biết trong chuẩn UART. Sự tồn tại của chân SCK giúp quá trình truyền ít bị lỗi và vì thế tốc độ truyền của SPI có thể đạt rất cao. Xung nhịp chỉ được tạo ra bởi chip Master.

MISO – Master Input / Slave Output: nếu là chip Master thì đây là đường Input còn nếu là chip Slave thì MISO lại là Output. MISO của Master và các Slaves được nối trực tiếp với nhau..

MOSI – Master Output / Slave Input: nếu là chip Master thì đây là đường Output còn nếu là chip Slave thì MOSI là Input. MOSI của Master và các Slaves được nối trực tiếp với nhau.

SS – Slave Select: SS là đường chọn Slave cần giao tiếp, trên các chip Slave đường SS sẽ ở mức cao khi không làm việc. Nếu chip Master kéo đường SS của một Slave nào đó xuống mức thấp thì việc giao tiếp sẽ xảy ra giữa Master và Slave đó. Chỉ có 1 đường SS trên mỗi Slave nhưng có thể có nhiều đường điều khiển SS trên Master, tùy thuộc vào thiết kế của người dùng.



Hình 3.10. Giao diện SPI

3.3.5.2. Quá trình truyền nhận dữ liệu

Các chân SPI trong AVR: Các chân giao tiếp SPI cũng chính là các chân PORT thông thường, vì thế nếu muốn sử dụng SPI chúng ta cần xác lập hướng cho các chân này. Trên chip Atmega8, các chân SPI như sau:

SCK – PB7 (chân 8)

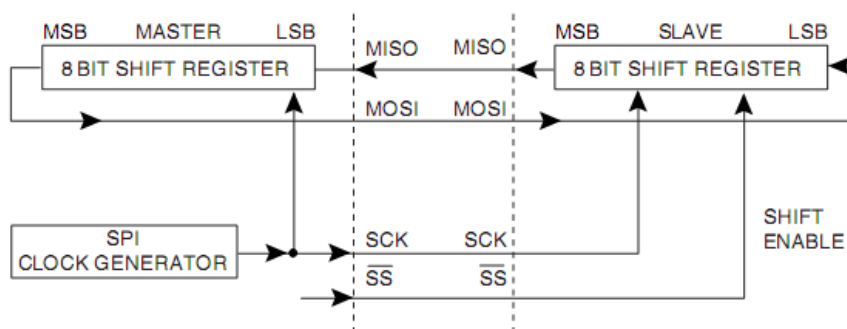
MISO – PB6 (chân 7)

MOSI – PB5 (chân 6)

SS – PB4 (chân 5)

Khi chip AVR được sử dụng làm Slave, bạn cần set các chân SCK input, MOSI input, MISO output và SS input. Nếu là Master thì SCK output, MISO output, MOSI input và khi này chân SS không quan trọng, chúng ta có thể dùng chân này để điều khiển SS của Slaves hoặc bất kỳ chân PORT thông thường nào.

Hoạt động của SPI: mỗi chip Master hay Slave có một thanh ghi dữ liệu 8 bits. Cứ mỗi xung nhịp do Master tạo ra trên đường giữ nhịp SCK, một bit trong thanh ghi dữ liệu của Master được truyền qua Slave trên đường MOSI, đồng thời một bit trong thanh ghi dữ liệu của chip Slave cũng được truyền qua Master trên đường MISO. Do 2 gói dữ liệu trên 2 chip được gửi qua lại đồng thời nên quá trình truyền dữ liệu này được gọi là “song công”. Hình 3.11 mô tả quá trình truyền 1 gói dữ liệu thực hiện bởi module SPI trong AVR, bên trái là chip Master và bên phải là Slave.



Hình 3.11. Truyền nhận dữ liệu trên SPI

Cực của xung giữ nhịp, phase và các chế độ hoạt động: cực của xung giữ nhịp (Clock Polarity) được gọi tắt là CPOL là khái niệm dùng chỉ trạng thái của chân SCK ở trạng thái nghỉ. Ở trạng thái nghỉ (Idle), chân SCK có thể được giữ ở mức cao (CPOL=1) hoặc thấp (CPOL=0). Phase (CPHA) dùng để chỉ cách mà dữ liệu được lấy mẫu (sample) theo xung giữ nhịp. Dữ liệu có thể được lấy mẫu ở cạnh lên của SCK (CPHA=0) hoặc cạnh xuống (CPHA=1). Sự kết hợp của SPOL và CPHA làm nên 4 chế độ hoạt động của SPI. Nhìn chung việc chọn 1 trong 4 chế độ này không ảnh hưởng đến chất lượng truyền thông mà chỉ cốt sao cho có sự tương thích giữa Master và Slave.

3.4. Cổng USB

3.4.1. Chuẩn USB

Universal Serial Bus là một chuẩn truyền dữ liệu cho bus ngoại vi do hai hãng Intel và Microsoft phối hợp phát triển. Một cổng USB đơn có thể được dùng để kết nối tới 127 thiết bị ngoại vi như chuột, modem, bàn phím, máy in, máy scan, ...

Mục tiêu hướng tới khi sử dụng USB:

- Dễ dàng mở rộng các thiết bị đầu cuối của PC.
- Cung cấp giải pháp chi phí thấp song vẫn hỗ trợ truyền dẫn với tốc độ lên đến 480 Mb/s.
- Hỗ trợ ứng dụng thời gian thực.
- Giao thức linh hoạt cho các chế độ hỗn hợp.
- Tích cực công nghệ thiết bị tiện nghi.
- Đa năng do có nhiều thiết bị có thể ghép nối với PC qua cổng USB.

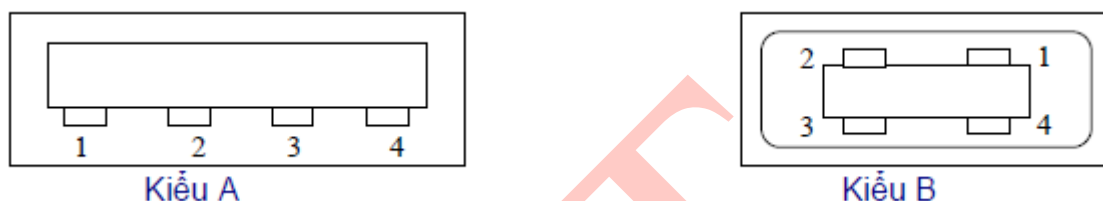
3.4.2. Đầu nối và cáp nối

USB có hai kiểu đầu nối khác nhau được gọi là A, B. Bus USB sử dụng cáp nối 4 sợi dây để nối với các thiết bị ghép nối. Trong đó có một cặp đường truyền 2 sợi xoắn được dùng làm đường dẫn dữ liệu vi phân, ký hiệu là D+ và D-. Còn một cặp

kia dùng làm đường 5V và đường nối đất chung. Cáp nối luôn được thực hiện liên kết 1:1. Sự sắp xếp các chân ở đầu nối cáp tuân theo những quy định sau:

Bảng 3.6. Sắp xếp chân ở cổng USB

Chân	Tên gọi	Màu dây
1	Vcc	Đỏ
2	D-	Trắng
3	D+	Xanh lục
4	GND	Đen



Hình 3.12. Cấu trúc cổng USB.

Hiện nay, ổ cắm USB của máy tính đều là kiểu A, qua đó có thể đấu trực tiếp thiết bị USB vào máy tính. Các thiết bị có tốc độ thấp như chuột có thể đấu thẳng vào ổ cắm thông qua một phích cắm cũng kiểu A.

Trong các trường hợp khác thiết bị thường có một ổ cắm kiểu B, muốn nối với máy tính phải sử dụng một cáp kiểu A,B. Trong trường hợp cần nối dài cáp, tức là để tăng khoảng cách giữa máy tính PC tới thiết bị ghép nối, người ta sử dụng cáp A,A. Cho đến nay các cáp USB đều được các nhà sản xuất cung cấp dưới dạng hoàn chỉnh trên đó đầu cắm, độ dài, chất lượng bọc kim chống nhiễu đều không thể thay đổi được. Vì vậy, tùy theo mục đích sử dụng ta phải lựa chọn thông số cáp cho chính xác từ chiều dài cho đến đầu nối.

Thông qua ổ cắm USB có thể lấy ra điện áp + 5V với dòng điện tiêu thụ 100 mA. Trong một số trường hợp có thể lấy tới 500 mA. Hai đường dẫn dữ liệu D +, D - cho phép đấu nối với các linh kiện USB đặc biệt chẳng hạn như là một số vi điều khiển tín hiệu ở chân D +, D - là các tín hiệu vi phân với mức điện áp = 0 - 3,3V. Điện áp nguồn nuôi cho bus có thể tăng đến 5,25V và khi chịu dòng tải lớn có thể giảm xuống 4,2V. Một vi mạch ổn áp trong trường hợp này có thể tạo ra một điện áp ổn định +3,3V. Toàn bộ hệ thống có thể thiết kế sao cho khi chịu dòng tải lớn điện áp nguồn cũng không vượt quá + 4,2V. Khi thiết bị ghép nối cần dòng tiêu thụ >100mA cần xem xét kỹ khả năng cung cấp và chịu tải của các linh kiện phía trong máy tính để tránh nguy hiểm. Khi ghép nối một thiết bị với bus USB, thường phải phân biệt rõ các

thiết bị sử dụng nguồn nuôi riêng như máy in với các thiết bị nhận điện áp nguồn nuôi qua bus. Trong một số trường hợp cả hai chế độ nguồn nuôi có thể cùng tồn tại để lựa chọn theo cách thiết kế của bus.

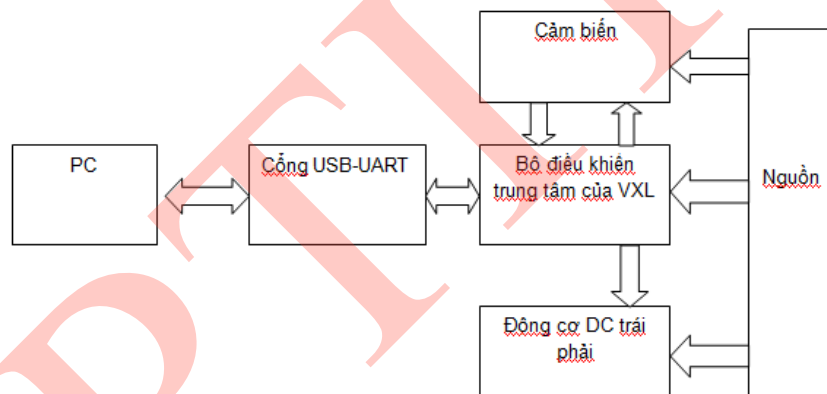
3.4.3. Mô tả hệ thống

Một hệ thống USB được mô tả bởi 3 định nghĩa: kết nối USB, các thiết bị USB, USB host. Kết nối USB bao gồm kiến trúc bus, những mối quan hệ Inter-Layer, các mô hình luồng dữ liệu, lập trình USB.

Các thiết bị USB: gồm có hub cung cấp điểm lắp thêm vào USB và thiết bị hiểu được giao thức USB. Mô hình sơ lược về truyền nhận dữ liệu qua cổng USB.

Tín hiệu điều khiển từ giao diện được lập trình trên C# được truyền qua chuẩn giao tiếp USB-UART đưa tới bộ điều khiển trung tâm, tại đây tín hiệu được xử lý và truyền tới các động cơ DC điều chỉnh hướng.

Bộ điều khiển trung tâm phát tín hiệu và nhận dữ liệu từ cảm biến rồi truyền về PC thông qua cổng USB-UART.



Hình 3.13. Sơ đồ tổng quát của hệ thống

3.4.4. Cấu trúc thành phần

3.4.4.1. Truyền nhận dữ liệu USB

Để truyền dữ liệu thì thông tin được chia nhỏ lại thành các dạng gói thông tin:

- Gói thông báo: chỉ ra các điểm đầu cuối của quá trình truyền dữ liệu.
- Gói bắt đầu khung truyền: bắt đầu khung và số khung đáp ứng khoảng cách thời gian chính xác để bắt đầu khung khác.
- Gói dữ liệu: 8bit chứa thông tin cần truyền.

Thực hiện kết nối USB gồm các thành phần:

Thiết bị USB vật lý: Một phần cứng ở đầu cuối của cáp USB thi hành một số chức năng hữu dụng với người sử dụng.

Client Software: Phần mềm thi hành trên máy chủ, đáp ứng cho thiết bị USB. Phần mềm này có thể được cung cấp bởi hệ điều hành hoặc được cung cấp cùng thiết bị USB.

USB System Software: Là phần mềm hỗ trợ USB trong một hệ điều hành cụ thể. Phần mềm hệ thống USB thường được cung cấp kèm với hệ điều hành, nó không phụ thuộc vào một thiết bị USB cụ thể nào cũng như phần mềm client cụ thể nào.

USB Host Controller: Bao gồm phần cứng và phần mềm cho phép các thiết bị USB kết nối với Host.

USB Bus Interface Layer: Cung cấp kết nối vật lý, gói tin, báo hiệu giữa Host và một thiết bị.

USB Device Layer: Cho thấy hệ điều hành hỗ trợ thiết bị USB cần có một phần mềm hệ thống để thực hiện việc quản lý thiết bị USB logic.

Function Layer: cung cấp khả năng được thêm vào cho host qua phần mềm của thiết bị USB trên host.

Quá trình xử lý tổng quát:

Một trong những nhiệm vụ của hub là phát hiện sự cắm vào hoặc tháo ra của các thiết bị. Mỗi hub có một điểm ngắt để báo cáo những sự kiện trên cho máy chủ. Khi hệ thống khởi động, máy chủ thăm dò hub gốc của nó để biết có hay không các thiết bị được gắn vào, các thiết bị đó có thể là các hub bổ sung và các thiết bị khác gắn vào hub gốc. Sau khi khởi động, máy chủ tiếp tục thăm dò định kỳ để báo có thiết bị mới được gắn vào hay không hoặc một thiết bị vừa được tháo ra.

Để tìm hiểu một thiết bị mới, máy chủ gửi một chuỗi các yêu cầu tới hub của thiết bị để hub này thiết lập một đường truyền giữa máy chủ và thiết bị. Sau đó máy chủ sẽ tìm hiểu về thiết bị bằng cách gửi các yêu cầu chuẩn của USB tới điểm cuối 0 của thiết bị bằng kiểu truyền điều khiển. Tất cả các thiết bị USB phải hỗ trợ truyền điều khiển, các yêu cầu chuẩn và điểm cuối 0.

Đối với người sử dụng thì quá trình tìm hiểu máy chủ là ẩn và hoàn toàn tự động ngoại trừ một số thông báo cho biết có một thiết bị mới được gắn vào và việc cài đặt thiết bị có thành công hay không. Đôi khi, trong lần đầu tiên sử dụng, người dùng cần phải chọn bộ điều khiển (driver) thích hợp hoặc chỉ cho máy chủ nơi mà nó có thể tìm thấy bộ điều khiển cho thiết bị.

Các trạng thái khi thiết bị không hoạt động.

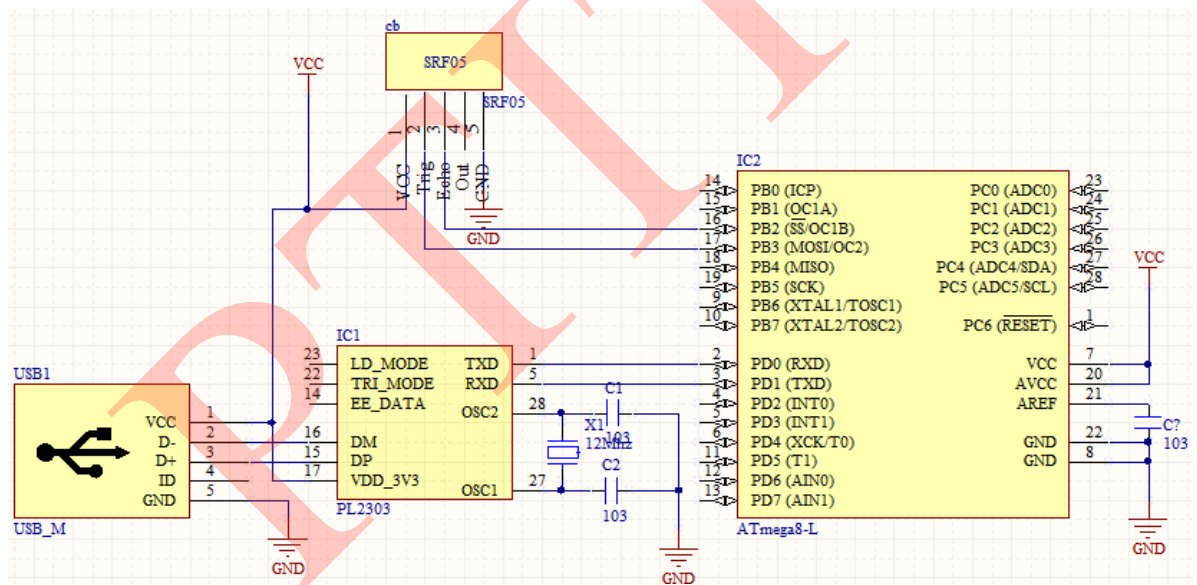
Khi quá trình tìm hiểu được hoàn thành, windows đưa thiết bị mới vào mục Device Manager trong Control Panel. Còn khi người sử dụng gỡ thiết bị ra khỏi hệ thống thì windows cũng gỡ thiết bị ra khỏi danh sách trong Device Manager.

Trạng thái tháo ra: Nếu hub không cấp nguồn tới các đường bus của thiết bị thì thiết bị ở trạng thái tháo ra. Hiện tượng thiết bị không được cấp nguồn mặc dù vẫn đang gắn vào hub xảy ra khi hub phát hiện sự quá tải về dòng từ phía thiết bị hoặc hub nhận được yêu cầu từ phía máy chủ bắt buộc nó ngắt nguồn.

Trạng thái treo: Thiết bị sẽ ở vào trạng thái treo khi nó phát hiện thấy đã 3ms trôi qua mà không thấy bất kỳ một tín hiệu nào trên bus. Các thiết bị đều phải hỗ trợ trạng thái này tức thiết bị phải có khả năng phát hiện khoảng thời gian 3ms không tích cực của bus và đưa thiết bị vào trạng thái treo.

Chuẩn USB-UART.

USB cho phép người sử dụng dễ dàng hơn nhờ hỗ trợ tính năng cài đặt Plug-and-Play (cắm và chạy) và đặc biệt là tính năng cắm gỡ nóng (có thể cắm và gỡ thiết bị USB trong khi máy tính vẫn đang hoạt động mà không cần phải khởi động lại). Cấp USB gồm hai sợi nguồn (+5V và dây chung GND) cùng một cặp gồm hai sợi dây xoắn để mang dữ liệu. Trên sợi nguồn máy tính có cấp nguồn lên tới 500mA ở điện áp 5V một chiều



Hình 3.14. Sơ đồ giao tiếp công USB với bộ điều khiển trung tâm (kit Arduino)

3.5. Đo và tạo thời gian chính xác bằng chip PIT – 8253/54

3.5.1. Giới thiệu chung

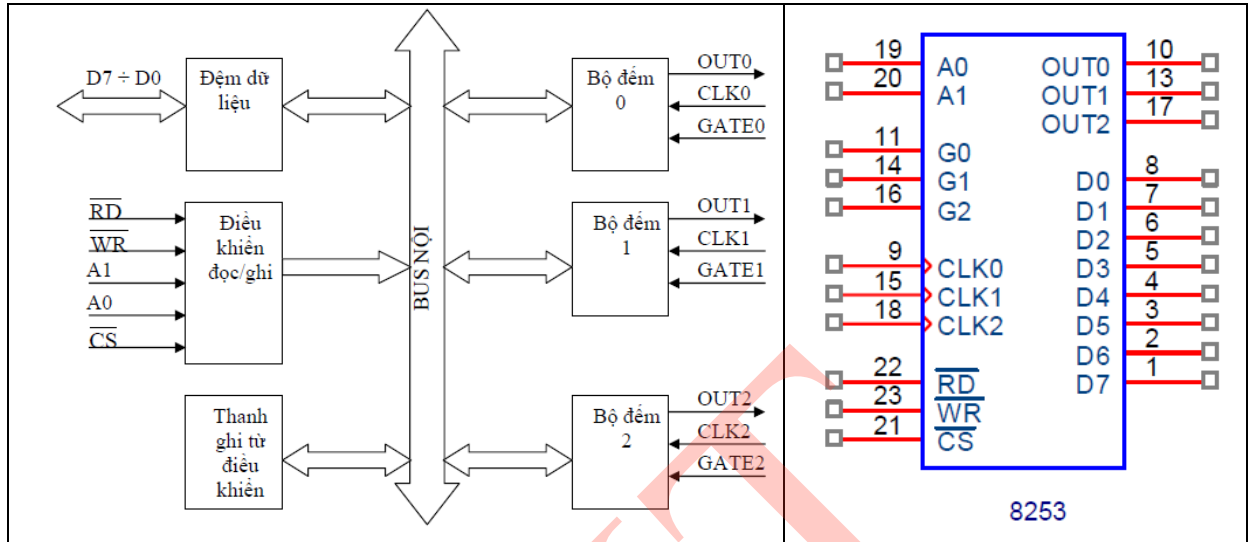
Muốn đo hoặc tạo ra những xung nhịp định thời chính xác và ổn định bằng một hệ thống ghép nối máy tính người ta sử dụng vi mạch định thời có thể lập trình được là PIT-8253 (Programmable Interval Timer).

PIT-8253 khi được lắp trên bản mạch ghép nối cắm vào khe cắm mở rộng của PC có thể phục vụ cho nhiều mục đích như: tạo máy phát sóng tỷ lệ khả trình, đếm các sự

kiện, tạo đồng hồ thời gian thực, tạo các xung đơn, đo khoảng cách thời gian chính xác, đo tần số, điều khiển mô tơ bước...

3.5.2. Cấu tạo của PIT – 8253/54

Sơ đồ khối và sơ đồ chân của PIT 8253/54 được mô tả ở hình 3.15.



Hình 3.15. Sơ đồ khối và sơ đồ chân của PIT 8253/54

Bộ đệm dữ liệu là loại 8 bit-2 hướng-3 trạng thái dùng để phối ghép 8253 với bus hệ thống của PC.

Số liệu sẽ được bộ đếm phát hoặc thu bằng việc chạy lệnh OUT hoặc IN của CPU.

Bộ đệm dữ liệu có 3 chức năng cơ bản:

- Định trình cho chế độ hoạt động của 8253/54.
- Nạp các thanh ghi của bộ đếm.
- Đọc các giá trị đếm.

Mạch logic điều khiển đọc/viết nhận các tín hiệu vào từ bus hệ thống và lần lượt phát các tín hiệu điều khiển cho thiết bị hoạt động. Nó bị cấm hoặc cho phép bởi tín hiệu chọn chip CS nhằm không để một hoạt động nào xảy ra khi mà chip chưa được chọn bởi hệ thống PC.

Chân /RD khi ở mức thấp (= 0) cho phép CPU đọc số liệu từ 8253 trong dạng 1 giá trị các bộ đếm.

Chân /WR khi ở mức thấp (= 0) cho phép CPU viết dữ liệu vào 8253, đó là các thông tin về chế độ hoạt động hoặc nạp các số đếm ban đầu vào bộ đếm.

Hai chân A0, A1 nối với 2 bit địa chỉ thấp nhất của bus địa chỉ PC. Khi kết hợp với chân RD, WR sẽ cho phép truy cập một trong ba bộ đếm hoặc thanh ghi điều khiển theo quy luật sau từ bảng :

Bảng 3.7. Bảng chế độ hoạt động của 8253.

/CS	/RD	/WR	A1	A0	Chức năng
0	1	0	0	0	Nạp bộ đếm 0
0	1	0	0	1	Nạp bộ đếm 1
0	1	0	1	0	Nạp bộ đếm 2
0	1	0	1	1	Viết thanh ghi lệnh
0	0	1	0	0	Đọc bộ đếm 0
0	0	1	0	1	Đọc bộ đếm 1
0	0	1	1	0	Đọc bộ đếm 2
0	0	1	1	1	Đọc thanh ghi lệnh (8254)
1	X	X	X	X	Không tích cực
0	1	1	X	X	Không hoạt động

Thanh ghi từ điều khiển được chọn khi $A1A0=11$. Thông tin được lưu giữ trong thanh ghi này sẽ điều khiển:

- Chế độ hoạt động của các bộ đếm.
- Chọn cách đếm (nhị phân hay BCD).
- Phương thức đọc/viết của mỗi thanh ghi đếm số liệu của các bộ đếm.

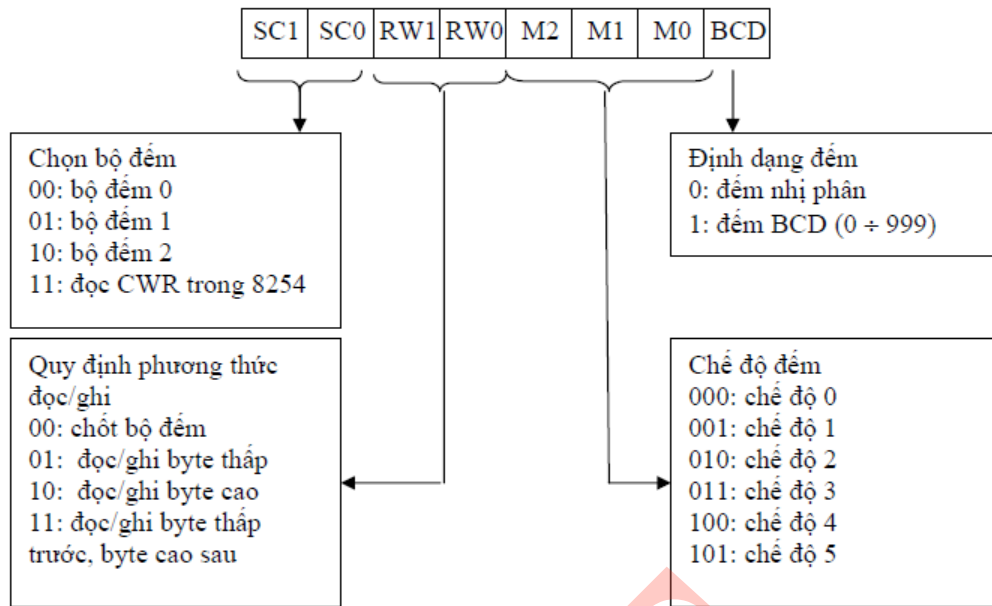
Đối với 8253 thanh ghi này chỉ được viết. Đối với 8254 thanh ghi này có thể đọc/viết.

Ba bộ đếm 0, 1, 2 là loại đếm lùi, 16 bit và có thể đặt giá trị đếm ban đầu. Mỗi bộ đếm có 2 lối vào là xung clock – CLK và GATE, 1 lối ra OUT. Khi GATE = 1, bộ đếm được phép đếm và ngược lại

3.5.3. Lập trình cho 8253

3.5.3.1. Định dạng từ điều khiển

Dùng các lệnh vào/ra đơn giản để lập trình cho các hoạt động của 8253. Đầu tiên phải viết từ điều khiển vào thanh ghi từ điều khiển. Định dạng từ điều khiển như sau:



Hình 3.16. Định dạng từ điều khiển.

3.5.3.2. Chế độ đếm

PIT 8253/54 có sáu chế độ đếm từ chế độ 0 đến chế độ 5.

Chế độ 0 (Interrupt on Terminal Count): Khi kết thúc đếm thì phát ra một sườn xung lên (thường dùng làm tín hiệu yêu cầu ngắt trong các hệ thống). Tín hiệu lỗi ra ở mức thấp ngay sau khi chọn chế độ và cho tới khi số xung đếm kết thúc thì sẽ chuyển lên mức cao.

Việc nạp lại thanh ghi bộ đếm sẽ tạo ra các kết quả sau:

- Nạp byte đầu tiên (byte thấp) sẽ làm ngừng bộ đếm hiện tại.
- Nạp byte thứ 2 (byte cao) sẽ khởi phát quá trình đếm mới.

Chế độ 1 (Programmable Monoflop): Phát ra một xung đơn với thời gian tồn tại xung biến đổi được. Tín hiệu lỗi ra OUT chuyển xuống mức thấp tại sườn âm của xung clock đầu tiên (khởi phát đếm) và sẽ chuyển lên mức cao khi bộ đếm kết thúc.

Bộ đếm được khởi động bởi sườn dương của xung cửa GATE. Trong quá trình đếm, nếu xung cửa xuất hiện nhiều sườn dương, thì với mỗi lần như vậy bộ đếm lại được đếm từ giá trị ban đầu.

Chế độ này cho phép tạo ra các xung đơn có độ rộng biến đổi tùy theo giá trị nạp ban đầu.

Chế độ 2 (Rate Generator): Phát ra các sóng tỷ lệ. Các xung nhịp đưa vào bộ đếm được chia bởi giá trị nạp trong thanh ghi bộ đếm. Tín hiệu lỗi ra xuống mức thấp trong chu kỳ đầu tiên và sau đó chuyển lên mức cao trong các chu kỳ còn lại. Quá trình cứ tiếp tục như vậy.

Ví dụ: muốn chia 4 tần số lỗi vào ta để lỗi ra có 1 chu kỳ ở mức thấp và 3 chu kỳ còn lại ở mức cao.

Chế độ 3 (Square-Wave Generator): Phát ra sóng xung vuông. Chế độ này tương tự như chế độ 2 nhưng xung lỗi ra là sóng vuông khi giá trị đếm chẵn và sẽ thêm một chu kỳ ở mức cao khi giá trị đếm lẻ.

Chế độ 4 (Software-triggered Pulse): Tạo các xung bằng phần mềm. Chế độ này giống như chế độ 2 nhưng xung của GATE không khởi động quá trình đếm mà sẽ đếm ngay khi số đếm ban đầu được nạp. Ngay sau khi nạp chế độ hoạt động, lỗi ra ở mức cao để đếm và khi kết thúc đếm lỗi ra xuống mức thấp trong 1 chu kỳ xung đếm rồi lại trở lại mức cao.

Chế độ 5 (Hardware-triggered Pulse): tạo các xung bằng xung của GATE. Cũng giống như chế độ 2, sườn dương của xung Gate khởi động quá trình đếm sau đó xuống mức thấp trong 1 chu kỳ nhịp khi việc đếm kết thúc. Khi quá trình đếm chưa kết thúc, nếu xung của xuất hiện những sườn dương thì cứ mỗi lần như vậy bộ đếm lại được đếm lại từ giá trị ban đầu..

Tóm tắt vai trò của xung GATE như sau:

Bảng 3.8. Tóm tắt hoạt động của xung của GATE 8253/54.

Trạng thái tín hiệu Chế độ	Mức thấp hoặc sườn âm	Sườn dương	Mức cao
0	Cấm bộ đếm	- - -	Cho phép đếm
1	- - -	- Khởi phát đếm - Reset lỗi ra sau xung nhịp tiếp theo	- - -
2	- Cấm đếm - Đặt lỗi ra ở mức cao ngay lập tức	Khởi phát đếm	Cho phép đếm
3	- Cấm đếm - Đặt lỗi ra ở mức cao ngay lập tức	Khởi phát đếm	Cho phép đếm
4	Cấm bộ đếm	- - -	Cho phép đếm
5	- - -	Khởi phát đếm	- - -

3.5.3.3. Viết bộ đếm

Chương trình phải thực hiện hai thao tác viết bộ đếm:

- Viết từ điều khiển để xác lập chế độ hoạt động của các bộ đếm.
- Nạp số đếm ban đầu cho các bộ đếm.

Sau khi được nạp giá trị ban đầu, bộ đếm sẽ thực hiện đếm lùi từ đó xuống 0.

Giá trị ban đầu ứng với hệ số chia tần của bộ đếm trong các chế độ 2 và 3.

Số đếm ban đầu được nạp là 0 tương đương với việc nạp số đếm cực đại 2^{16} (đếm nhị phân) hoặc 10^4 (đếm BCD)

3.5.3.4. Đọc bộ đếm

Có hai cách chọn lựa cho việc đọc bộ đếm trong 8253/54:

- Đọc trực tiếp bởi một hoặc hai lệnh IN.
- Dùng lệnh chốt bộ đếm (counter latch command).

Nếu cần đọc giá trị tức thời của bộ đếm khi quá trình đếm chưa kết thúc thì phải sử dụng các lệnh chốt bộ đếm hoặc lệnh đọc thanh ghi từ điều khiển (Read-back Command). Lệnh đọc này chỉ có trong 8254. Tại thời điểm lệnh chốt bộ đếm được viết vào thanh ghi từ điều khiển, giá trị đếm được chốt vào thanh ghi đếm ra của PIT.

Định dạng của từ điều khiển cho lệnh chốt bộ đếm được mô tả ở bảng 3.9.

Bảng 3.9. Định dạng của từ điều khiển.

D7	D6	D5	D4	D3	D2	D1	D0
SC ₁	SC ₀	0	0	X	X	X	X
2 bit chọn bộ đếm được chốt: 00: chọn bộ đếm 0 01: chọn bộ đếm 1 10: chọn bộ đếm 2				X: tùy chọn (có thể đặt = 0)			

Lệnh đọc thanh ghi từ điều khiển cho phép chốt giá trị đếm và chốt byte lưu trữ trạng thái của bộ đếm được chọn.

Định dạng cho lệnh đọc thanh ghi điều khiển được mô tả ở bảng 3.10.

Bảng 3.10. Định dạng cho lệnh đọc thanh ghi điều khiển.

D7	D6	D5	D4	D3	D2	D1	D0
1	1	Count	Status	Count2	Count1	Count0	0
		= 0: chốt giá trị đếm; = 1: ko xác định	= 0: chốt byte trạng thái của bộ đếm; =1: ko xác định	= 1: chọn bộ đếm 2	= 1: chọn bộ đếm 1	= 1: chọn bộ đếm 0	

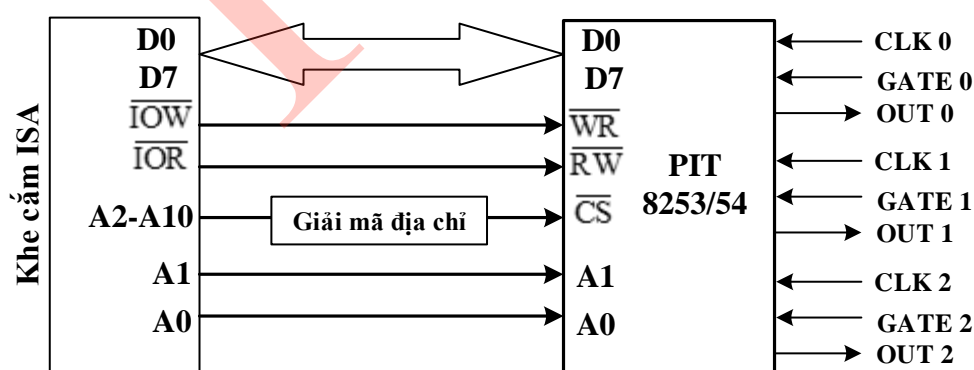
Định dạng byte trạng thái được mô tả ở bảng 3.11.

Bảng 3.11. Định dạng byte trạng thái.

D7	D6	D5	D4	D3	D2	D1	D0
Pin	Null	RW ₁	RW ₀	M ₂	M ₁	M ₀	BCD
Trạng thái của OUT _x ; (x=0,1,2) = 1: lỗi ra mức cao. = 0: thấp	Bộ đếm đã được nạp giá trị ban đầu chưa? =1: chưa nạp, chưa đọc được. = 0: nạp rồi đọc được	Giống định dạng từ điều khiển					

3.5.4. Ghép nối PIT 8253/54

Sơ đồ khối ghép nối 8253/54 với các chân tín hiệu trên khe cắm mở rộng như 8255A được mô tả ở hình 3.17.



Hình 3.17. Ghép nối 8253/54 với PC qua khe cắm ISA.

3.6. Ví dụ

Ví dụ 1: 8250 cần trao đổi 8 bit, tốc độ 9600 baud, không kiểm tra chẵn/lẻ, 2bit Stop, địa chỉ làm việc đầu 03F8h.

- Tìm địa chỉ cho thanh ghi điều khiển trạng thái – LCR. (Ghi)
- Tìm địa chỉ cho thanh ghi chốt bộ chia– LS. (Ghi)
- Tìm địa chỉ cho thanh ghi cho phép ngắt– IER. (Ghi)
- Tìm địa chỉ cho thanh ghi nhận dạng ngắt– IIR. (Đọc)
- Tìm địa chỉ cho thanh ghi điều khiển MODEM – MCR. (Ghi)
- Tìm địa chỉ cho thanh ghi trạng thái MODEM – MSR. (Đọc)
- Tìm địa chỉ cho thanh ghi trạng thái đường truyền – LSR. (Đọc)

Giải:

- Thanh ghi điều khiển trạng thái:

$D0D1 = 11; D2 = 1; D3 = 0; D4 = 0; D5 = 0; D6 = 0;$

Nếu $D7 = 1 \rightarrow data1_LCR = 87h;$

Nếu $D7 = 0 \rightarrow data2_LCR = 07h$

$addr_LCR = 03FBh$

- Thanh ghi chốt bộ chia:

Tốc độ 9600 baud \rightarrow số chia = 000C

$\rightarrow data_LS = 0Ch$ và $data_MS = 00h$

$addr_LS = 03F8h$

$addr_MS = 03F9h$

- Thanh ghi cho phép ngắt không làm việc, cấm ngắt:

$data_IER = 00h$

$addr_IER = 03F9h$

- Đọc thanh ghi nhận dạng ngắt:

$THR_rong = 02h$

$addr_IIR = 03FAh$

- Ghi thanh ghi điều khiển MODEM:

$addr_MCR = 3FCh$

$data_DRT = 01h$

$data_RTS = 02h$

- Đọc thanh ghi trạng thái MODEM:

$addr_MSR = 3FEh$

$DSR(D5), RI(D6), DCD(D7) = 1$

$\rightarrow data_MSR = E0h$

- Đọc thanh ghi trạng thái dòng:

$addr_LSR = 3FDh$

Không lỗi $D1 D2 D3 = 0$; thời gian đệm rỗng $D5 D6 = 1$

$\rightarrow data1_LSR = 60h$

Ví dụ 2: Đặt 8250 (COM 1) ở chế độ truyền 1200 baud, 8 bit dữ liệu, 1 bit Stop, không bit chắn lẻ. Hãy lập đoạn chương trình để:

- Đặt các thông số cho cổng nối tiếp.
- Kiểm tra bộ đệm truyền (THR).
- Kiểm tra thanh ghi trạng thái đường truyền (LSR)

Giải:

Các bước:

- Khởi tạo
 - Chốt số chia để quy định tốc độ baud
 - Lựa chọn số bit dữ liệu, bit Stop và kiểu chắn lẻ.
 - Bắt đầu thực hiện truyền/nhận:
 - Thanh ghi LSR kiểm tra xem thanh ghi đệm truyền THR đang trống rỗng (nếu đúng thì việc xuất ra được thực hiện).
 - LSR kiểm tra xem có dữ liệu nhận được đang chờ để đọc
- Đặt thông số cho cổng nối tiếp:
- ```
/*Dat toc do baud 1200*/
Baud_rate = 1843200 /(1200*16);
outp (0x3FB, 0x80); /*dat 1 vao DLAB*/
outp (0x3F8, (baud_rate & 0xFF)); /*load LSB into latch*/
outp (0x3F9, ((baud_rate & FF00)>>4)); /*load MSB into latch*/
'Now set the format
outp (0x1, FB, 3); /*8, 1 and No parity*/
```
- Kiểm tra thanh ghi bộ đệm truyền THR
- ```
/* Loop until THRE*/  
While (((status = inp (0x3FD)) & 40)); /*while bit 5 is low*/  
Outp (0x3F8, var);  
Kiểm tra thanh ghi trạng thái đường truyền LSR  
/* check for flag*/  
if ((status = inp (0x3FD)) & 01); /*if bit 0 is hi*/  
Var = inp (0x3F8); /* get the character */
```

Ví dụ 3. Viết đoạn chương trình cho PIT 8253/54:

- Đặt bộ đếm 0 (địa chỉ là 304h) thực hiện đếm 16 bit hoạt động ở chế độ 0.
- Nạp bộ đếm 0 một giá trị thích hợp, như là OUT 0 = 1 sau 0.025sec

Giải:

Giả sử xung clock được nối với PCLK (**14.7456/6 MHz**) → chu kỳ là **6/14.7456** microseconds. Do vậy, hệ số nạp của bộ đếm là $(0.025 \times 14.7456 \times 10^6)/6 = 61440 = \mathbf{F000h}$.

Nạp giá trị 30h vào thanh ghi Control Word Register, và nạp giá trị F000h vào bộ đếm Counter0 :

MOV AL, 30h ; load the control word into AL

OUT 0Eh, AL ; and send it to the Control Register since the 8253 PIT chip is connected to the low byte of the data bus, two write cycles are required to load F000h into counter0

MOV AL, 00h ; load the low byte of F000h

OUT 08h, AL ; into low byte of Counter0

MOV AL, F0h ; load the high byte of F000h

OUT 08h, AL ; into high byte of Counter0

CÂU HỎI ÔN TẬP

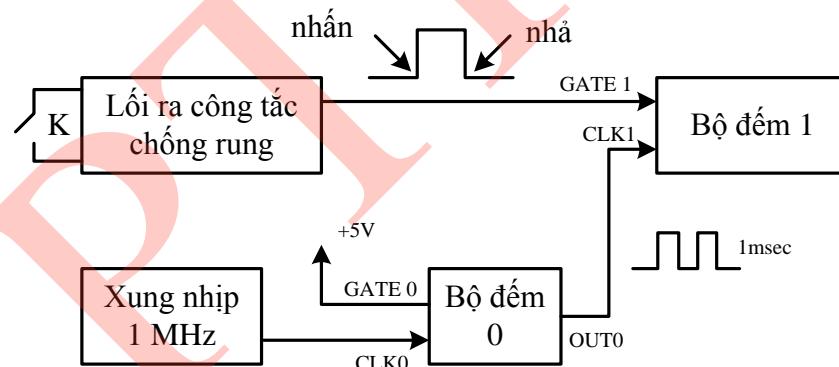
1. Lập trình để truyền 1 ký tự từ COM 1 và nhận nó qua cổng này.
2. Viết chương trình để truyền dòng chữ “Khoa KTDĐT1” từ máy tính qua cổng COM1 hiển thị trên LCD.
3. Viết chương trình để truyền dòng chữ “2010-2014” từ máy tính qua cổng COM1 hiển thị trên LED 7 đoạn, có hiệu ứng nhấp nháy.
4. Viết chương trình để truyền dòng chữ “HEN GAP LAI” từ máy tính qua cổng COM1 hiển thị trên LCD, chữ chạy từ phải sang trái.
5. Viết chương trình tìm các số nguyên tố nhỏ hơn 100 và hiển thị kết quả trên máy tính thông qua giao tiếp cổng COM.
6. Xây dựng chương trình đơn giản để thực hiện việc truyền lệnh điều khiển từ máy tính tới vi điều khiển 8051 để bật tắt thiết bị điện thông qua cổng COM (yêu cầu mô hình hệ thống, chức năng các thiết bị và chương trình chương trình).
7. Nối lối ra của máy phát xung nhịp 1 MHz tới lối vào CLK2 của bộ đếm 2 chip PIT 8253; nối lối vào GATE2 lên + 5V để luôn cho phép đếm. Tạo chương trình phát sóng vuông qua bộ đếm 2 trong chip PIT -8253 với tần số bất kì được nhập từ bàn phím. **(Cho địa chỉ C0 = 304)**
8. Nối lối ra của máy phát xung nhịp 1 MHz (tần số có thể được điều chỉnh tăng hoặc giảm liên tục từng bước 100Hz một bằng việc nhấn phím “T” (tăng) hoặc nhấn phím “G” (giảm)) tới lối vào CLK2 của bộ đếm 2 chip PIT 8253; nối lối vào GATE2 lên + 5V để luôn cho phép đếm. Tạo chương trình phát sóng vuông qua bộ đếm 2 trong chip PIT -8253 với tần số bất kì được nhập từ bàn phím. **(Cho địa chỉ C0 = 304)**

9. Nối lối ra của máy phát xung nhịp 1 MHz tới lối vào CLK2 của bộ đếm 2 chip PIT 8253; nối lối vào GATE2 lên +5V để luôn cho phép đếm. Viết chương trình để phát ra các xung tỷ lệ lặp lại có độ rộng xung thỏa mãn điều kiện sau: mức thấp 40 μ s mức cao 200 μ s. **(Cho địa chỉ C0 = 304)**

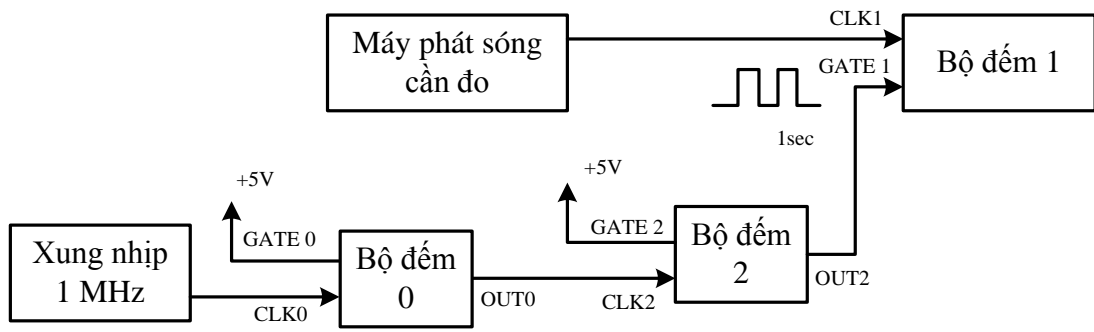
10. Nối đầu ra của công tắc tạo xung đơn vào lối vào CLK1 của bộ đếm 1 vi mạch PIT-8254. Viết chương trình liên tục hiển thị số xung được đếm. Nhấn công tắc để tạo xung. **(Cho địa chỉ C0 = 304)**

11. Nối đầu ra của công tắc tạo xung đơn vào lối vào CLK0 của bộ đếm 0 vi mạch PIT-8254. Viết chương trình liên tục hiển thị số xung được đếm. Nhấn công tắc để tạo xung. Giá trị đếm ban đầu luôn là 1111 1111 1111 1111 = FFFFh. **(Cho địa chỉ C0 = 304)**

12. Thực hiện đo thời gian chính xác giữa 2 sự kiện (thí dụ giữa các sự kiện nhấn và nhả công tắc chống rung K) như sơ đồ hình sau. Các xung đếm chuẩn có độ rộng 1ms được tạo ra từ nguồn xung nhịp 1MHz (có sai số tương đối về tần số rất thấp 10^{-5} đến 10^{-6}) và lối ra OUT0 của bộ đếm 0 được đưa tới lối vào CLK1 của bộ đếm 1. Bộ đếm sẽ chỉ đếm được số xung trong khoảng thời gian giữa 2 lần nhấn công tắc K tức là khi giá trị GATE1 bằng 1. **(Cho địa chỉ C0 = 304)**



13. Sơ đồ tạo một bộ đo tần số xung tuần hoàn chính xác dùng vi mạch PIT-8253/54 như hình sau. Viết chương trình cho sơ đồ. **(Cho địa chỉ C0 = 304)**



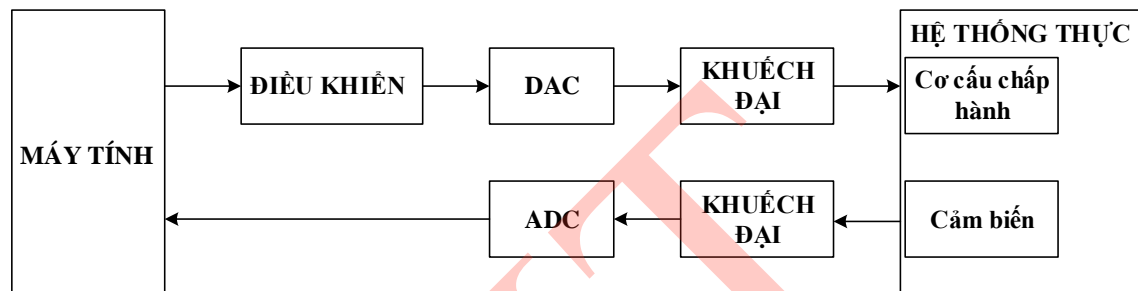
Đo tần số

CHƯƠNG 4. GHÉP NỐI SỐ - TƯƠNG TỰ, TƯƠNG TỰ - SỐ

4.1. Tổng quan

Máy vi tính là thiết bị số, nhiều thiết bị ngoại vi dùng trong điều khiển và đo lường là thiết bị thu/phát các tín hiệu tương tự như nhiệt độ, áp suất ... Do đó, các bộ biến đổi tín hiệu số sang tín hiệu tương tự (DAC) và tín hiệu tương tự sang tín hiệu số (ADC) là phần tử trung gian giữa máy tính và các thiết bị ngoại vi.

Hình 4.1 minh họa một hệ thống điều khiển của máy tính với thiết bị ngoại vi.



Hình 4.1. Sơ đồ cấu trúc hệ thống điều khiển tự động.

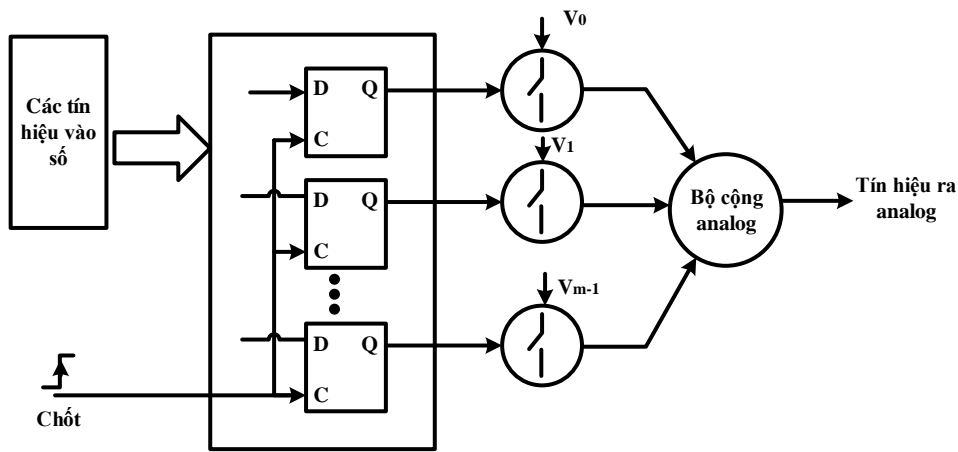
Các bộ cảm biến dùng để biến đổi các giá trị vật lý như nhiệt độ, áp suất... sang điện áp hay dòng điện tương tự. Sau các bộ cảm biến thường là các bộ khuếch đại và bộ lọc. Các tín hiệu ở đầu ra bộ khuếch đại là các tín hiệu tương tự nên chúng được đưa qua các bộ biến đổi tương tự sang số (ADC- Analog to Digital Converter) để chuyển sang tín hiệu số. Máy tính sẽ tiếp nhận tín hiệu này để xử lý, lưu trữ hay so sánh rồi đưa ra các lệnh điều khiển tới hệ thống cơ cấu chấp hành thông qua bộ biến đổi số sang tương tự (DAC- Digital to Analog Converter) và bộ khuếch đại.

4.2. Bộ biến đổi số-tương tự (DAC)

Các bộ biến đổi D/A thường dựa trên nguyên lý cơ bản như hình 4.2. Trong sơ đồ nguyên lý bao gồm một mạch cộng analog để cộng tất cả điện thế analog tỷ lệ với từng bit nhị phân. Các trigơ D là các mạch chốt m bit nhị phân của tín hiệu đầu vào. Mỗi bit thứ i sẽ điều khiển các công tắc tương ứng nhằm cho qua điện thế V_i tỷ lệ với các trọng số thứ i tới bộ cộng mỗi khi giá trị của nó đạt mức 1. Do vậy, đầu ra của bộ cộng có giá trị là:

$$V = B_{m-1} \cdot V_{m-1} + B_{m-2} \cdot V_{m-2} + \dots + B_0 \cdot V_0$$

Trong đó B_i là giá trị nhị phân (0 hoặc 1) của bit thứ i .

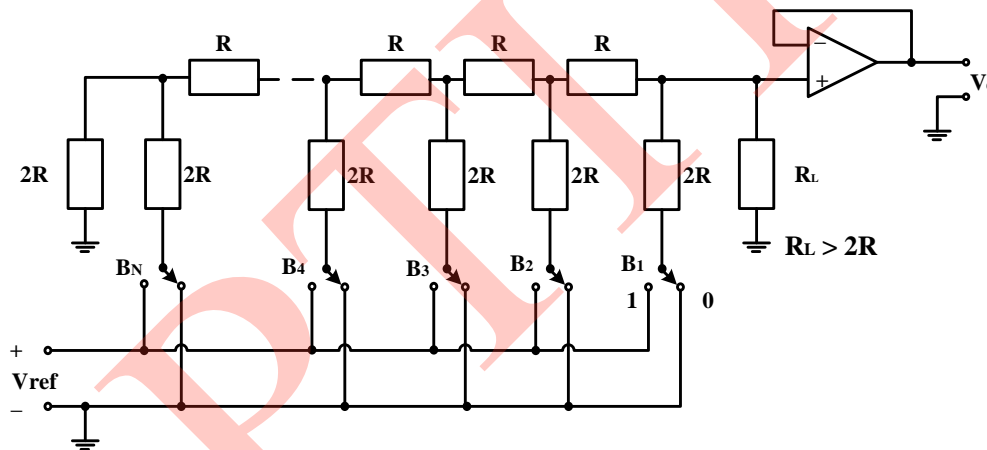


Hình 4.2. Bộ biến đổi D/A cơ bản.

Hiện nay, loại DAC thông dụng nhất là loại dùng thang các điện trở R-2R để cấp dòng cho bộ cộng analog, sơ đồ minh họa hình 4.3.

Điện áp đầu ra tỷ lệ với giá trị các bit đầu vào là:

$$V_0 = -V_{\text{ref}} \left(\frac{B_1}{2^1} + \frac{B_2}{2^2} + \dots + \frac{B_N}{2^N} \right)$$



Hình 4.3. Mạch biến đổi D/A kiểu R-2R.

4.3. Các tham số của bộ biến đổi D/A

Độ phân giải

Độ phân giải là tỉ số giữa giá trị cực tiểu đối với giá trị cực đại của điện áp đầu ra, về trị số tỉ số này tương ứng với tỉ số giá trị cực tiểu đối với giá trị cực đại của tín hiệu số đầu vào.

Thí dụ đối với DAC 10 bit, có độ phân giải là:

$$\frac{0000000001}{1111111111} = \frac{1}{2^{10}-1} = \frac{1}{1023} \approx 0,001$$

Độ phân giải của DAC cũng có thể biểu thị bằng số bit tín hiệu số đầu vào.

Điện áp ra toàn thang

Điện áp ra toàn thang là điện áp cực đại ứng với trường hợp tất cả các bit lỗi vào đều bằng 1. Giá trị điện áp ra cực đại của bộ biến đổi luôn nhỏ hơn giá trị danh định ít nhất là 1 bit có trọng số nhỏ nhất. Ví dụ, đối với bộ biến đổi D/A 10 bit, điện áp ra toàn thang là 10V thì giá trị của bit LSB là $10V/1024 \approx 9.77 \text{ mV}$. Điện áp ra lớn nhất của bộ biến đổi nếu được điều chỉnh đúng là $10V - 0.00977 \approx 9.9902V$.

Độ tuyến tính

Độ tuyến tính là độ lệch của đầu ra so với đường thẳng nối điểm chuyển đổi từ trạng thái toàn 0 (các công tắc đều tắt) đến trạng thái toàn 1 (các công tắc đều đóng). Trong trường hợp lý tưởng, độ lệch của đầu ra không được vượt quá giá trị $\pm 1/2$ bit LSB nhằm duy trì độ chính xác toàn cục. Trên thực tế có rất nhiều bộ biến đổi D/A có sai số tuyến tính lớn hơn giá trị này mà vẫn cho ra kết quả đúng.

Độ chính xác

Độ chính xác của bộ biến đổi D/A là sự sai lệch giữa giá trị lỗi ra thực tế và giá trị lỗi ra mong muốn. Nó thường được xác định theo số phần trăm của điện áp hoặc dòng điện toàn thang. Trong trường hợp lý tưởng, sai số lớn nhất không vượt quá giá trị $\pm 1/2$ bit LSB.

Thời gian xác lập dòng điện, điện áp đầu ra

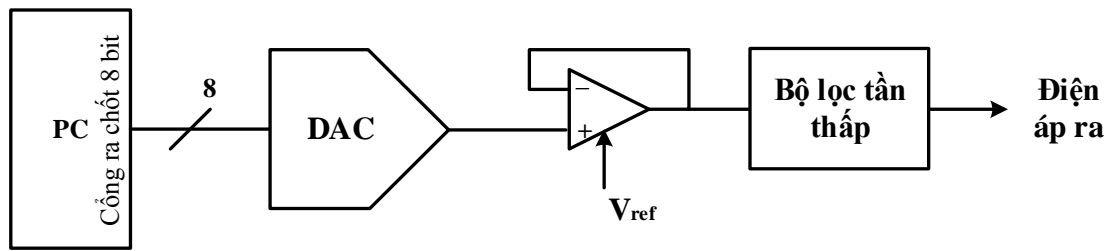
Thời gian xác lập, là thời gian từ khi tín hiệu số được đưa vào đến khi dòng điện hoặc điện áp đầu ra ổn định. Điều này có nghĩa đây là thời gian để bộ biến đổi đạt được khoảng $\pm 1/2$ bit LSB so với giá trị xác lập cuối cùng.

Ngoài các tham số trên còn một số tham số khác như: các mức logic cao, thấp, điện trở và điện dung đầu vào. Dải động, điện trở và điện dung đầu ra v.v.

4.4. Ghép nối bộ biến đổi D/A với máy tính

Bộ biến đổi D/A có thể tương thích với hầu hết các bộ vi xử lý. Đối với các bộ biến đổi D/A 8 bit thì việc ghép nối rất đơn giản vì các vi xử lý đều có BUS dữ liệu là bội của 8. Đối với các bộ biến đổi D/A 12 hay 16 bit cần phải sử dụng các bộ đệm trung gian.

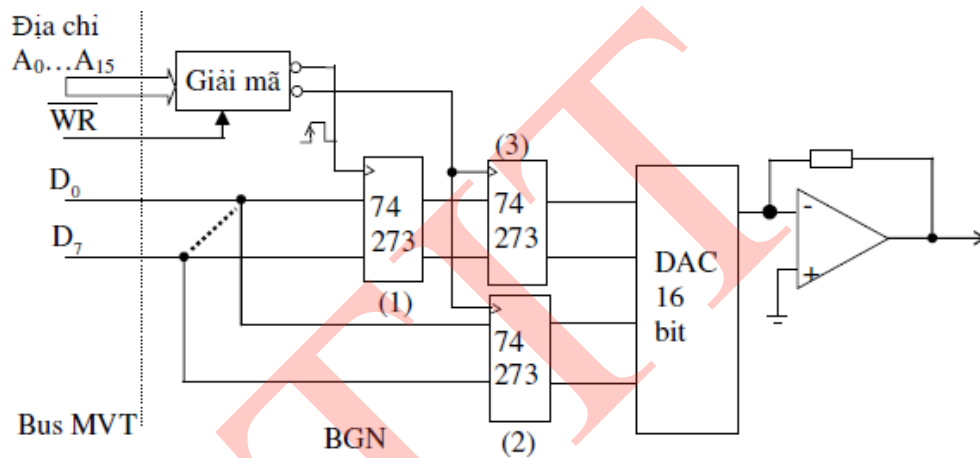
Thông thường các bộ biến đổi D/A là loại có dòng ra tỷ lệ với mã đầu vào nên người ta sử dụng mạch khuếch đại thuật toán để làm bộ biến đổi dòng điện thành điện áp. Do điện áp đầu ra biến đổi theo các mức lượng tử nên phải dùng một bộ lọc tần thấp để là trơn tín hiệu. Có thể sử dụng các cổng vào/ra của vi mạch PPI-8255 làm cổng ra 8 bit. Nếu số bit vào của D/A có 12 hay 16 bit thì phải sử dụng 2 cổng của 8255, có thể dùng thêm các mạch chốt dữ liệu 74244... Khi lập trình xuất ra cổng các số là hàm biến đổi theo thời gian, từ đó có thể tạo được các sóng sin, răng cưa, vuông hay tỷ lệ... Đồng thời độ ổn định tín hiệu ra chỉ phụ thuộc vào nguồn điện áp chuẩn V_{ref} cấp cho DAC.



Hình 4.4. Điều khiển DAC bằng máy tính.

Để ghép nối ta nối các lối vào của bộ biến đổi với một cổng (port). Trong trường hợp số bit của bộ biến đổi D/A (ví dụ 12 bit) lớn hơn số bit của cổng (ví dụ 8 bit) cần phải thiết kế mạch phần cứng và phần mềm hợp lý. Ví dụ, nối 8 bit thấp của 12 bit đầu vào của bộ biến đổi với một cổng và 4 bit cao còn lại với cổng khác.

Hình 4.5 minh họa cách ghép nối vi xử lý với bộ biến đổi D/A 16 bit.



Hình 4.5. Ghép nối vi xử lý với DAC 16 bit

4.5. Lập trình cho DAC

Lập trình xuất dữ liệu cho DAC 8 bit.

`Port[addr_port]:=data;`

Trong đó địa chỉ cổng `addr_port` sẽ qua mạch giải mã địa chỉ để tạo xung kích mở mạch đệm đưa dữ liệu số vào đầu vào DAC. Số liệu cần chuyển đổi được đặt trong `data`.

Đối với DAC 16 bit cần 2 lệnh xuất:

`Port[addr_port1]:=data1;`

`Port[addr_port2]:=data2;`

Trong đó địa chỉ cổng `addr_port1` sẽ qua mạch giải mã địa chỉ để tạo xung kích mở mạch chốt đệm 74273 (hình 4.5) thứ nhất, lưu giữ 8 bit thấp của dữ liệu, số liệu cần chuyển đổi được đặt trong `data1`. Lệnh xuất dữ liệu thứ 2 theo địa chỉ cổng `addr_port2` sẽ tạo xung kích mở mạch chốt đệm 74273 (hình 4.5) thứ hai, lưu giữ 8 bit

cao của dữ liệu, số liệu cần chuyển đổi được đặt trong *data2*. Đồng thời xung kích thứ hai cũng mở chốt đệm 74273 thứ ba. Do vậy DAC đồng thời nhận được 16 bit dữ liệu.

4.6. Bộ biến đổi tương tự - số (ADC)

Các bộ biến đổi A/D có nhiệm vụ biến đổi điện thế hoặc dòng điện thành dạng mã dạng số tương ứng tại đầu ra. Nếu số bit đầu ra là N tương ứng với độ phân giải là N bit. Các tham số của bộ biến đổi A/D tương tự như bộ biến đổi D/A.

Có rất nhiều cách để thực hiện biến đổi A/D. Phần sau giới thiệu một số phương pháp biến đổi A/D thông dụng.

4.6.1. Tham số đặc trưng của ADC

Dải biến đổi của điện áp tương tự ở đầu vào.

Là khoảng điện áp mà bộ chuyển đổi AD có thể thực hiện chuyển đổi được. Khoảng điện áp đó có thể lấy trị số từ 0 đến một trị số dương hoặc âm nào đó hoặc cũng có thể là điện áp có hai cực tính từ $-U_{Am}$ đến $+U_{Am}$.

Độ phân giải.

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

Ví dụ: Một ADC có số bit ở đầu ra $N = 12$ có thể phân biệt được $2^{12} = 4096$ mức trong dải biến đổi điện áp vào của nó. Độ phân biệt của một ADC được ký hiệu là Q và được xác định bởi biểu thức sau:

$$Q = \frac{U_{Am}}{2^N - 1}$$

Q là giá trị của một mức lượng tử hoá hoặc còn gọi là một LSB.

Thông thường các ADC có số bit từ 3 đến 12. Ngoài ra còn có một số các ADC đạt được độ chính xác có số bit từ 14 đến 16 bit.

Đặc tuyến truyền đạt lý tưởng của ADC là một đường bậc thang đều và có độ dốc trung bình bằng 1. Đặc tuyến thực có sai số lệch không, nghĩa là nó không xuất phát tại giá trị ứng với $\frac{1}{2}$ LSB. Nó là hình bậc thang không đều do ảnh hưởng của sai số khuếch đại, méo phi tuyến.

Tốc độ chuyển đổi.

Tốc độ chuyển đổi là số chuyển đổi trong một giây gọi là tần số chuyển đổi f_c . Cũng có thể dùng tham số thời gian chuyển đổi T_c để đặc trưng cho tốc độ chuyển đổi. T_c là thời gian cần thiết cho một lần chuyển đổi.

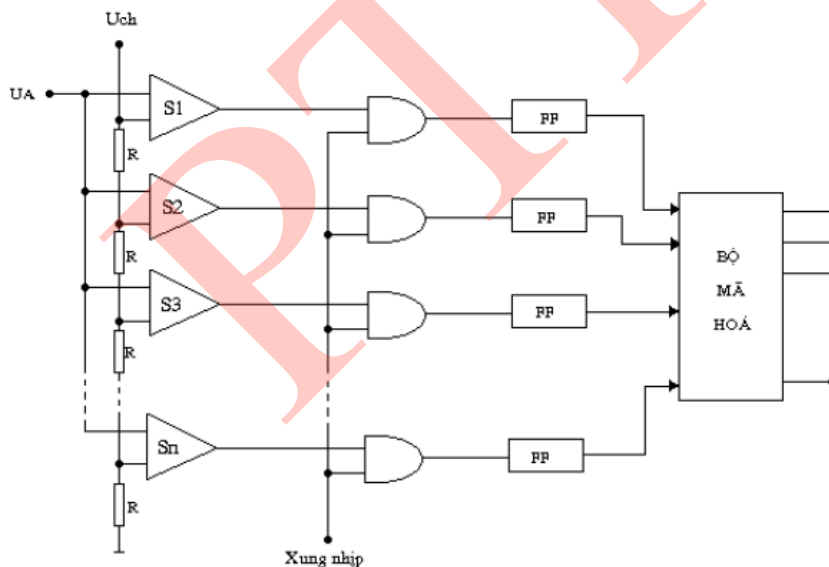
Chú ý rằng $f_c \neq 1/T_c$. Thường $f_c < 1/T_c$ và giữa các lần chuyển đổi còn có một khoảng thời gian cần thiết cho ADC phục hồi lại trạng thái ban đầu.

4.6.2. Bộ biến đổi A/D kiểu so sánh song song

Trong phương pháp chuyển đổi song song, tín hiệu tương tự U_A được đồng thời đưa tới các bộ so sánh $S_1 \div S_n$. Điện áp chuẩn được đưa tới đầu vào thứ hai của các bộ so sánh, thông qua thang điện trở R . Do đó các điện áp chuẩn đặt vào bộ so sánh lần lượt khác nhau một lượng không đổi và giảm dần từ bộ so sánh S_1 đến từ bộ so sánh S_n . Đầu ra của bộ so sánh có điện áp vào lớn hơn điện áp chuẩn lấy trên thang điện trở có mức logic “1” các đầu ra còn lại có mức logic “0”. Tất cả các đầu ra được nối với mạch AND (và), một đầu mạch AND được nối với mạch tạo xung nhịp chỉ khi có xung nhịp đưa đến đầu vào mạch AND thì các xung trên đầu ra mạch so sánh mới đưa tới mạch nhớ FF (Flip - Flop).

Như vậy cứ sau một khoảng thời gian bằng một chu kỳ của xung nhịp lại có một tín hiệu được biến đổi và đưa tới đầu ra. Xung nhịp đảm bảo cho quá trình so sánh kết thúc mới đưa tới tín hiệu vào bộ nhớ.

Mạch biến đổi song song có tốc độ chuyển đổi nhanh, vì quá trình so sánh được thực hiện song song, nhưng mạch phức tạp với số linh kiện quá lớn. Với bộ chuyển đổi N bit để phân biệt được 2^N mức lượng tử hóa phải dùng $(2^N - 1)$ bộ so sánh. Vì vậy phương pháp này chỉ dùng trong các ADC yêu cầu số bit N nhỏ và tốc độ chuyển đổi cao.



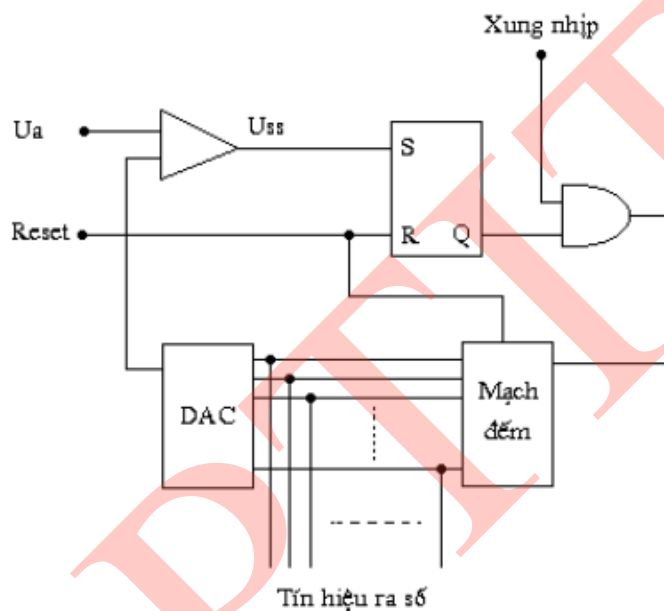
Hình 4.6. Bộ biến đổi A/D song song.

4.6.3. Bộ biến đổi A/D theo phương pháp đếm

Hình 4.7 trình bày sơ đồ nguyên tắc của ADC làm việc theo phương pháp đếm đơn giản. Mạch bao gồm bộ so sánh, trigơ RS, cổng AND dùng để đóng mở xung nhịp, bộ đếm n bit và một bộ DAC n bit. Các phần tử được mắc thành mạch hồi tiếp.

Lúc đầu mạch trigơ và mạch đếm được Reset, lối ra Q của trigơ có giá trị 0, bộ đếm cũng bị xóa về trạng thái 0, do vậy, điện áp ra U_{DAC} của bộ biến đổi D/A có giá trị là 0V. Cổng AND mở để cho các xung nhịp vào mạch đếm. Lúc này điện áp vào U_a lớn hơn U_{ADC} nên U_{SS} ở giá trị thấp. U_{DAC} tăng dần theo hình bậc thang vì mạch đếm liên tục đổi trạng thái từ thấp lên cao, khi U_{DAC} đủ lớn hơn U_a thì U_{SS} chuyển lên cao làm cho lối ra Q của trigơ chuyển lên mức logic 0 làm cổng AND đóng lại. Lúc này nội dung bộ đếm là tín hiệu số n bit tương ứng với tín hiệu tương tự U_a cần chuyển đổi.

Thời gian chuyển đổi của ADC theo phương pháp đếm phụ thuộc vào độ lớn của tín hiệu tương tự U_a và tần số xung nhịp. Nếu U_a càng lớn thì thời gian chuyển đổi càng dài, nếu xung nhịp cao thì thời gian chuyển đổi ngắn. Ngoài ra thời gian chuyển đổi sẽ hạn chế tần số biến thiên cao nhất của tín hiệu tương tự đầu vào U_a .



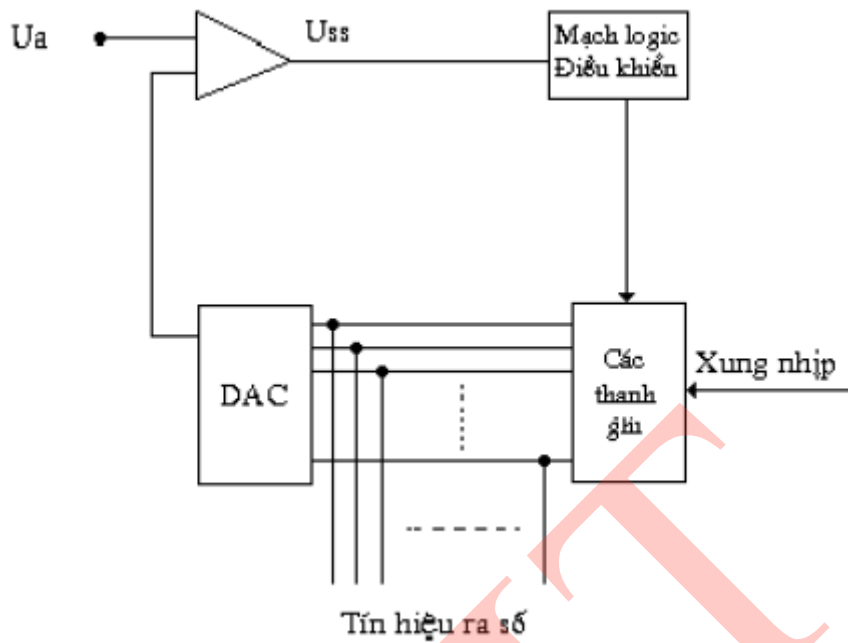
Hình 4.7. Bộ biến đổi A/D theo phương pháp đếm

4.6.4. Bộ biến đổi A/D theo phương pháp xấp xỉ liên tiếp

Bộ chuyển đổi A/D theo phương pháp xấp xỉ liên tiếp có ưu điểm là thời gian chuyển đổi tỷ lệ thuận với số bit của mã số và thời gian thiết lập của thanh ghi mà không phụ thuộc của tín hiệu chuyển đổi.

ADC có N bit tương ứng với quá trình biến đổi xảy ra N bước. Đầu tiên thanh ghi xuất ra số nhị phân có giá trị số bằng $\frac{1}{2}$ toàn dải, nghĩa là số bit có trọng số lớn nhất MSB có giá trị 1, các bit còn lại có giá trị 0. Như vậy, điện áp lối ra của bộ biến đổi D/A có giá trị bằng $\frac{1}{2}$ điện áp toàn dải. Nếu điện áp ra của bộ biến đổi D/A lớn hơn điện áp vào thì lối ra của bộ so áp U_{SS} sẽ ở mức logic 0 và điều khiển thanh ghi xóa bit MSB và đặt bit tiếp theo của số xuất ra lên mức logic 1. Nếu điện áp này vẫn

nhỏ hơn điện áp vào U_a thì thế lối ra của bộ so áp có mức logic 1 sẽ điều khiển thanh ghi giữ nguyên giá trị MSB bằng 1 và đặt bit tiếp theo của số xuất ra lên giá trị 1. Quá trình cứ lặp lại N lần và giá trị chuyển đổi số mong muốn.



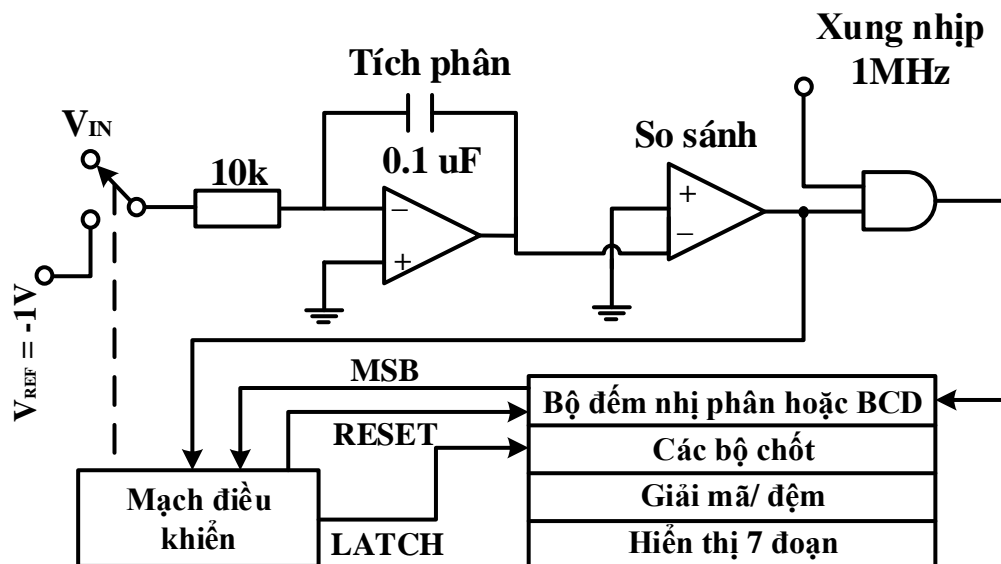
Hình 4.8. Bộ biến đổi A/D theo phương pháp đếm.

4.6.5. Bộ biến đổi A/D hai sườn xung

Bộ biến đổi A/D hai sườn xung cung cấp độ phân giải cao với giá thành hạ.

Đầu tiên, mạch điều khiển sẽ xóa tất cả các bộ đếm về giá trị 0 và nối lối vào của mạch tích phân với điện áp vào cần chuyển đổi. Giả sử điện áp vào là dương làm cho lối ra của mạch tích phân lúc này sẽ âm dần. Khi lối ra của mạch tích phân giảm xuống dưới 0 thì lối ra bộ so sánh sẽ chuyển lên mức cao. Lối ra bộ so sánh ở mức cao cho phép mở cổng AND đưa xung nhịp 1 MHz tới chuỗi các bộ đếm. Sau một số đếm nhất định, thường là 1000, mạch điều khiển sẽ chuyển lối vào bộ tích phân xuống mức điện áp chuẩn âm, đồng thời reset tất cả các bộ đếm về giá trị 0.

Với điện áp lối vào âm, lối ra mạch tích phân sẽ tăng dần và vượt qua giá trị 0V thì lối ra của bộ so sánh chuyển xuống mức thấp, cấm xung nhịp điều khiển bộ đếm. Số đếm cần thiết để lối ra của bộ tích phân quay về 0V tỷ lệ với điện áp vào. Vì điện trở và tụ điện trên mạch tích phân được dùng cho cả tích phân điện áp đầu vào và điện áp chuẩn nên những biến đổi nhỏ trong giá trị của chúng (do ảnh hưởng của nhiệt độ) sẽ không ảnh hưởng đến độ chính xác của quá trình biến đổi.



Hình 4.9. Bộ biến đổi A/D hai sườn xung.

4.6.6. Ghép nối các bộ biến đổi A/D với máy tính

Khi bộ biến đổi A/D sử dụng các bộ so sánh song song thì bộ biến đổi tạo ra các giá trị lỗi ra nhanh hơn khả năng đọc chúng của máy tính nhiều lần. Do vậy, cần phải dùng phương pháp truy cập bộ nhớ trực tiếp DMA. Khi vi xử lý thả nổi bus, chip điều khiển DMAC sẽ điều khiển bus và cho dữ liệu từ bộ biến đổi A/D truyền trực tiếp vào các ô nhớ liên tiếp.

Phần lớn các bộ biến đổi A/D loại sườn xung được thiết kế để ghép nối với các bộ hiển thị 7 đoạn. Do đó, chúng thường xuất dữ liệu dạng mã BCD hoặc mã 7 đoạn. Để đọc dữ liệu từ bộ biến đổi này thường dùng phương pháp hỏi vòng.

Các bộ biến đổi A/D theo phương pháp xấp xỉ liên tiếp thường có các lỗi ra cho từng bit một. Mã ở lỗi ra là dạng mã nhị phân. Có thể nối các lỗi ra song song của bộ biến đổi với số chân yêu cầu của một cổng vào và đọc số liệu bằng chương trình phần mềm. Ngoài các đường dữ liệu, có hai đường tín hiệu khác của bộ biến đổi xấp xỉ liên tiếp cần phải nối máy tính. Đầu tiên, máy tính phát ra tín hiệu START CONVERT (khởi phát) đến bộ biến đổi để báo cho nó thực hiện một quá trình chuyển đổi A/D. Bộ biến đổi phát ra tín hiệu EOC hoặc Finish (kết thúc) để báo hiệu quá trình biến đổi đã hoàn tất và dữ liệu trên các lỗi ra đã ổn định.

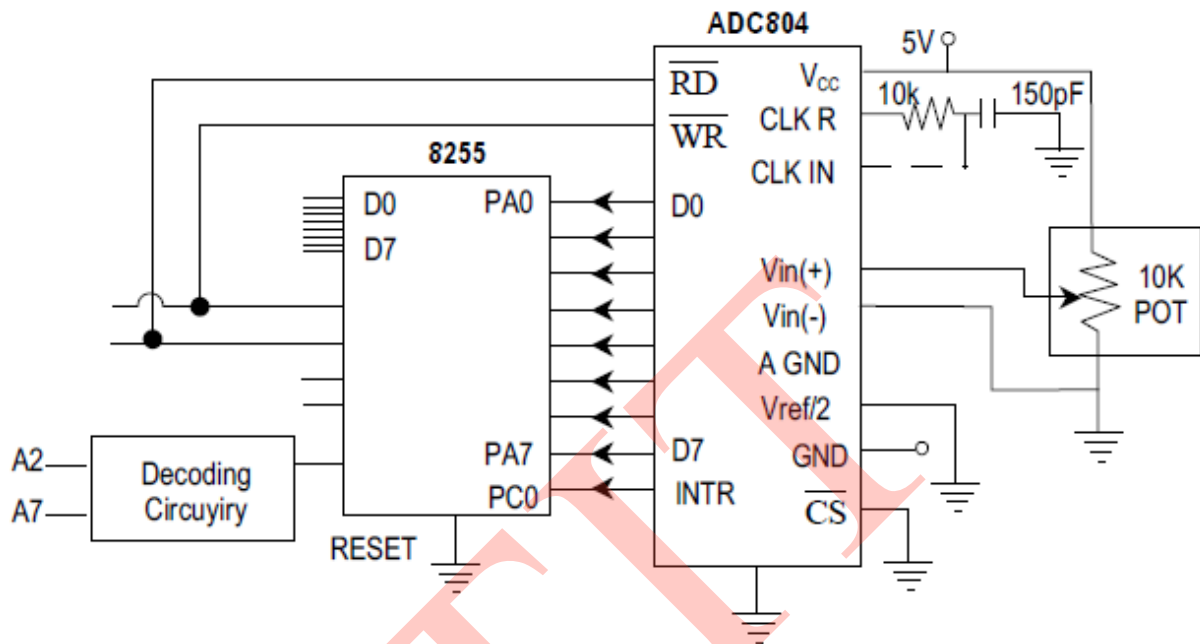
Until ((Var1 AND 1) = 1)

//Delay(thoigianchuyendoi) → không dùng mach doc finish

Data := port[addr_ADC];

4.7. Ví dụ

Ví dụ: Nối ghép ADC với 8255.



Hình 4.11. Nối ghép ADC 804 với 8255

```

MOV     A, #80H      ; Từ điều khiển với PA = đầu ra và PC = đầu vào
MOV     R1, #CRPORT  ; Nạp địa chỉ cổng điều khiển
MOVX    @R1, A       ; Đặt PA = đầu ra và PC = đầu vào
BACK:   MOV     R1, #CPORT ; Nạp địa chỉ cổng C
        MOVX    A, @R1    ; Đọc địa chỉ cổng C để xem ADC đã sẵn sàng chưa
        ANL     A, #00000001B ; Che tất cả các bit cổng C để xem ADC đã sẵn
                                ; sàng chưa
        JNZ     BACK      ; Giữ hiển thị PC0 che EOC
                                ; Kết thúc hội thoại và bây giờ nhận dữ liệu của ADC

MOV     R1, #APORT    ; Nạp địa chỉ PA
MOVX    A, @R1        ; A = đầu vào dữ liệu tương tự

```

CÂU HỎI ÔN TẬP

- Viết chương trình biến đổi các số 0, 255, 511, 1023, 2047, 4095 thành các tín hiệu tương tự ở lối ra DAC của chip DAC-80.
- Viết chương trình phát ra một sóng hình răng cưa với tần số và biên độ lớn nhất có thể được bằng bộ biến đổi D/A.

3. Viết chương trình phát ra một sóng hình răng cưa có thể biến đổi liên tục và tần số bằng việc nhấn các phím trên bàn phím bằng bộ biến đổi D/A.
4. Viết chương trình phát ra một sóng xung vuông với độ rộng biến đổi được bằng bộ biến đổi D/A.
5. Viết chương trình phát ra một sóng xung vuông có thể biến đổi liên tục và tần số bằng việc nhấn các phím trên bàn phím bằng bộ biến đổi D/A.
6. Viết chương trình biến đổi tương tự - số A/D để biến đổi điện áp một chiều 10V theo phương pháp đếm.
7. Viết chương trình biến đổi tương tự - số A/D để biến đổi điện áp một chiều 10V theo phương pháp xấp xỉ liên tục.
8. Viết chương trình biến đổi tương tự - số A/D để biến đổi điện áp một chiều 10V theo phương pháp hai sườn xung.

TÀI LIỆU THAM KHẢO

1. *Slide bài giảng Thiết bị ngoại vi và Kỹ thuật ghép nối* - Trần Thị Thúy Hà, Học viện công nghệ Bưu chính Viễn thông 2012.
2. *Kỹ thuật ghép nối máy vi tính*, Nguyễn Mạnh Giang, NXB Giáo dục 1997 .
3. *Nguyên lý phần cứng và kỹ thuật ghép nối máy vi tính*, Trần Quang Vinh, NXB Giáo dục 2002.
4. *Ghép nối và điều khiển thiết bị ngoại vi*, Nguyễn Văn Minh Trí, Lâm Tăng Đức.
5. *Đo lường và điều khiển bằng máy tính*, Ngô Diên Tập, NXB Khoa học và Kỹ thuật 1999.
6. *Cấu trúc máy vi tính*, Trần Quang Vinh, Đại học Quốc gia Hà nội, 2005.
7. *Microprocessor and interfacing*, Douglas V.Hall, Glencoe 1993.
8. *Microcomputer Interfacing, a practical guide for technicians, engineers and scientific*, Joseph J. Comer, Prentice hall, 1991.
9. *Peripheral design handbook*, Intel, 1991.
10. *PC Interfacing and Data Acquisition - Techniques for Measurement Instrumentation and Control*, Kelvin Jame, Newnes, 2000.
11. *Visual Basic for Electronic Engineerings Applications*, Vincent Him, Newnes, 2005.
12. *USB complete*, Jan Axelson, Newnes, 2001.
13. *Serial port complete*, Jan Axelson, Newnes, 2001
14. *Parallel port complete*, Jan Axelson, Newnes, 2001.
15. *The Windows serial port programming handbook*, Ying Bai, CRC Press LLC, 2005.