

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 CÔNG 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 CÔNG 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

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

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

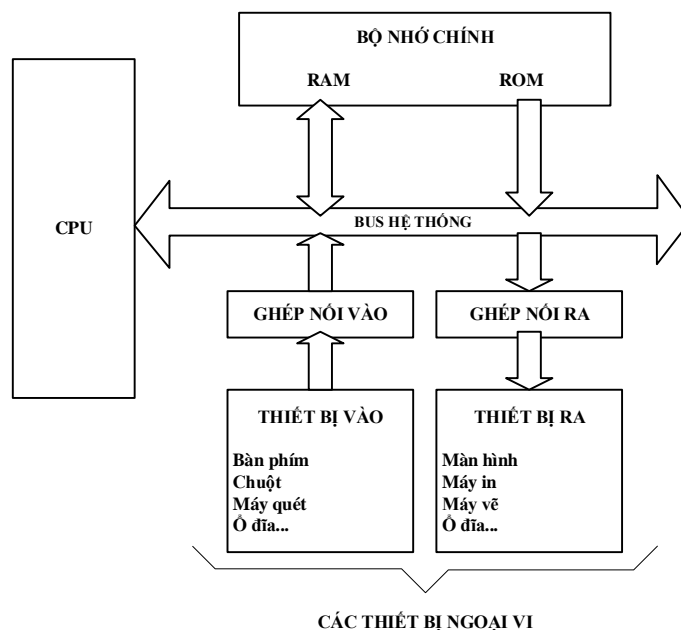
Khối xử lý trung tâm CPU (Central Processing Unit). Khối này được chế tạo bởi mạch vi điện tử có độ tích hợp rất cao.

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.

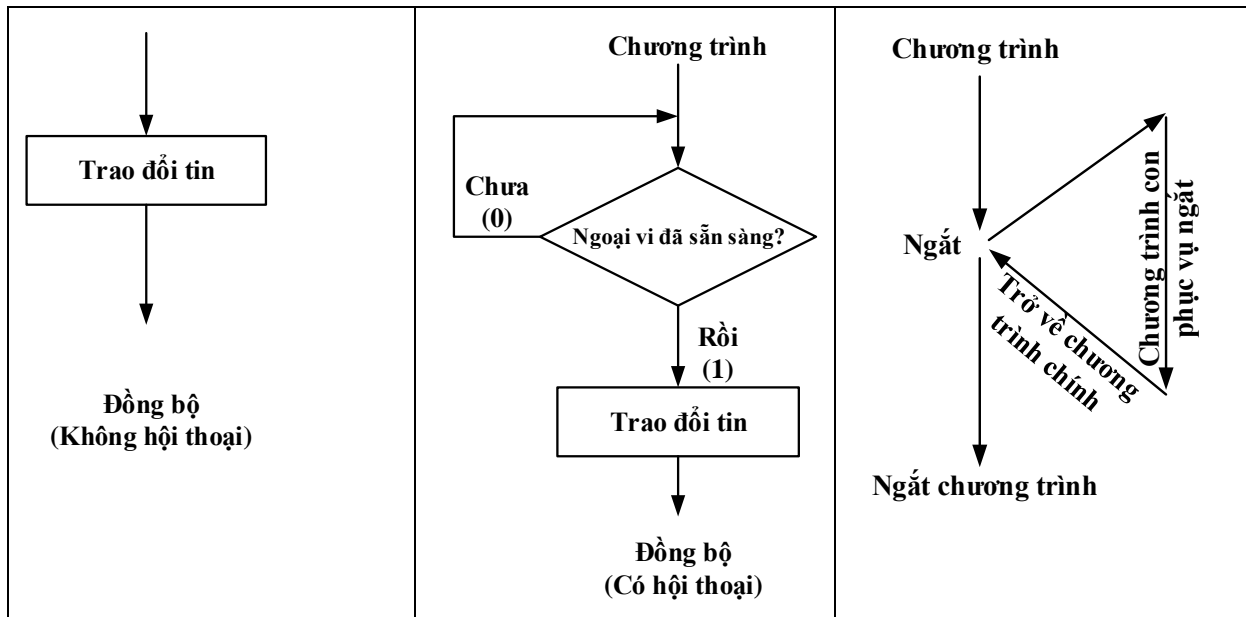
Trao đổi đồng bộ: máy tính không quan tâm xem ngoại vi có sẵn sàng trao đổi tin hay không mà đưa luôn các lệnh trao đổi tin (đọc, ghi hay chuyển số liệu).

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.

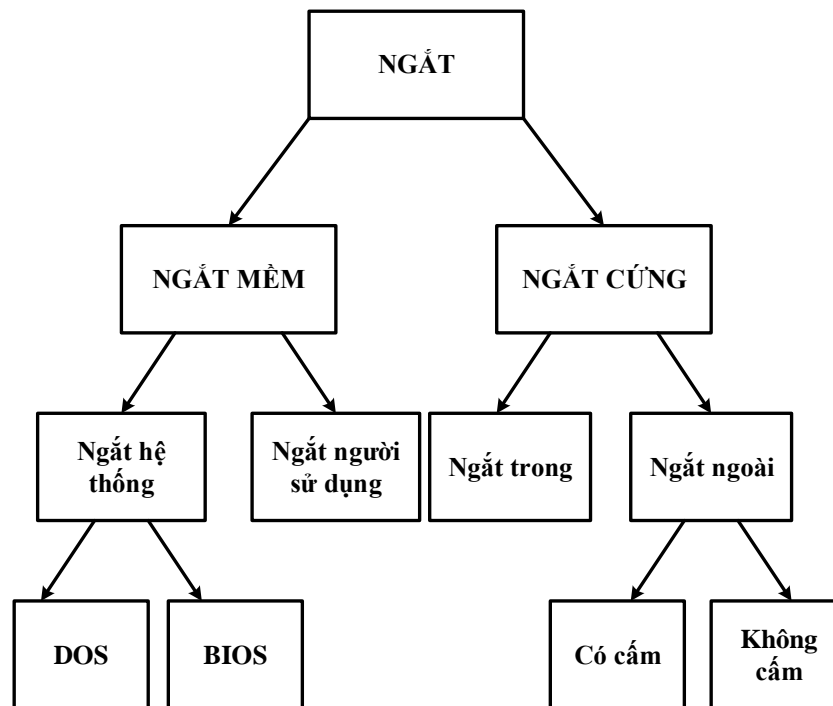
Ưu điểm : phương pháp này khá đơn giản, có độ tin cậy, chỉ trao đổi tin khi ngoại vi đã sẵn sàng.

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 đệm 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 (vectơ 0).

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

Thực hiện từng bước (vectơ 1).

Điểm dừng chương trình do người sử dụng định trước (vectơ 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 mốc nổi (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 lối vào ngắt INTR bằng cách dùng 1 cổng logic OR với:

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

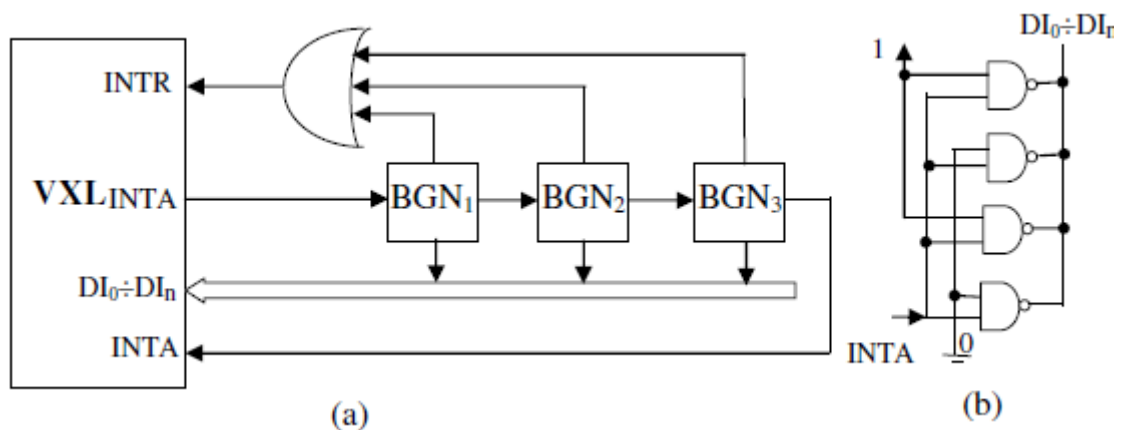
Phương pháp này giảm thời gian kiểm tra trạng thái khi chưa có yêu cầu nhưng không thay đổi được thứ tự ưu tiên 1 cách linh động.

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.



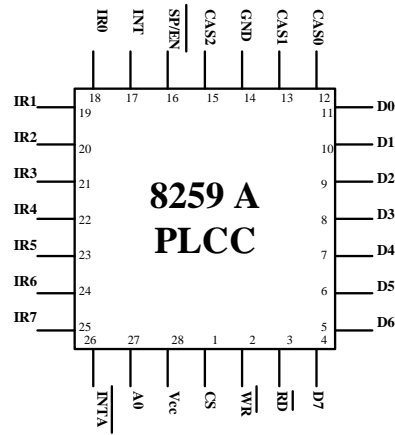
Hình 1.4. Sơ đồ ưu tiên ngắt theo vị trí và vector ngắt.

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

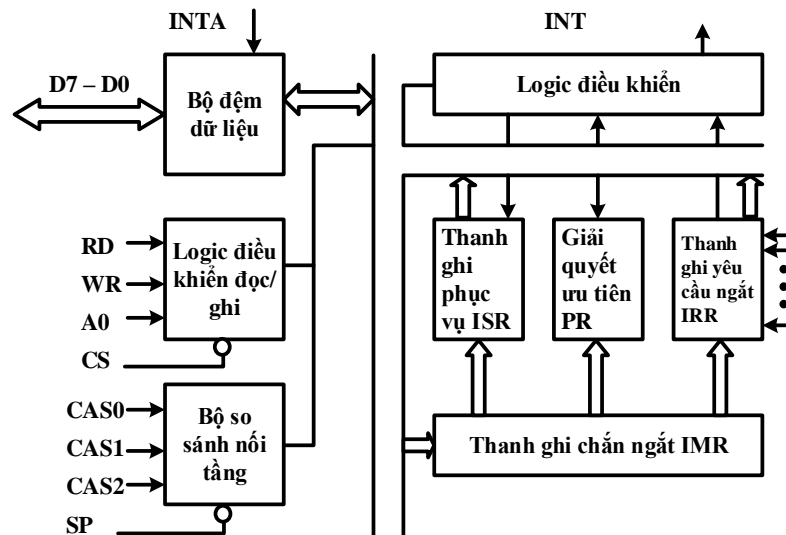
Có hai dạng PIC 8259A là:

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 4
- 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 kí sinh hình thành giữa các đường dẫn có thể làm biến dạng tín hiệu. Khoảng cách giới hạn cực đại là 8m. Thông thường chỉ 1,5 đến 2m vì lí do an toàn dữ liệu.

Nếu sử dụng khoảng cách ghép nối trên 3m thì các đường dây tín hiệu và đường dây nối đất phải được 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.

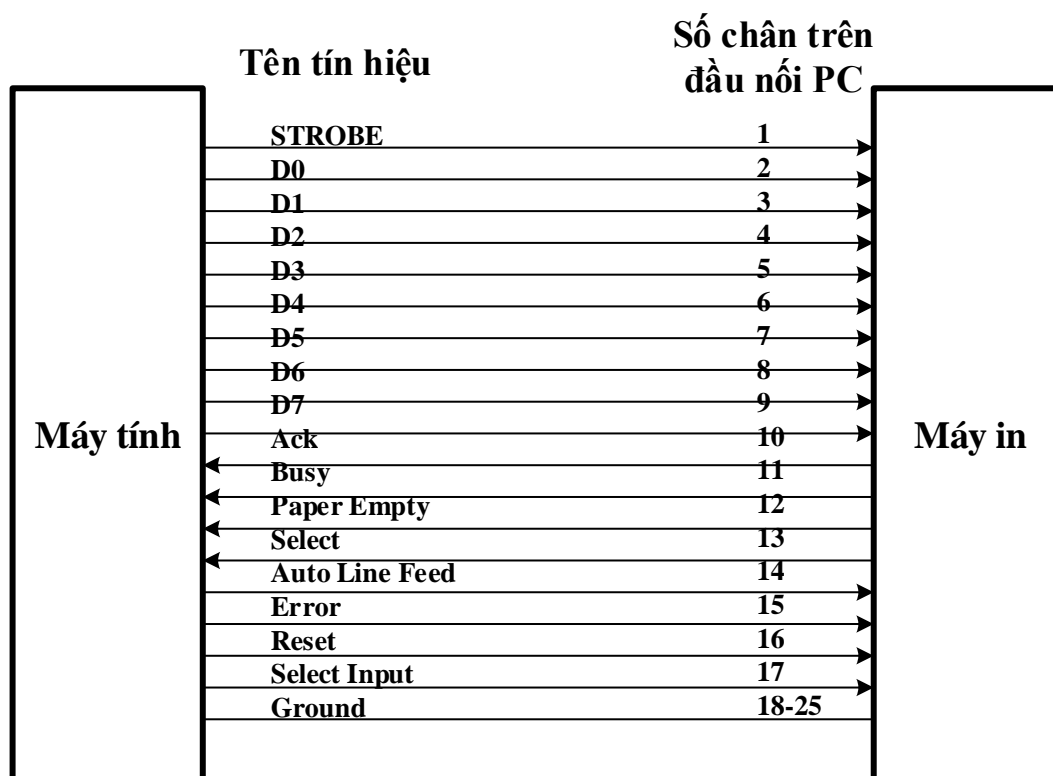
Với nhiều mục đích sử dụng thì khoảng cách này hoàn toàn thỏa đáng, tuy vậy cũng có những ứng dụng đòi hỏi phải truyền trên khoảng cách xa hơn. Trong trường hợp đó sự dụng khả năng ghép nối khác (như ghép nối qua cổng RS232).

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)

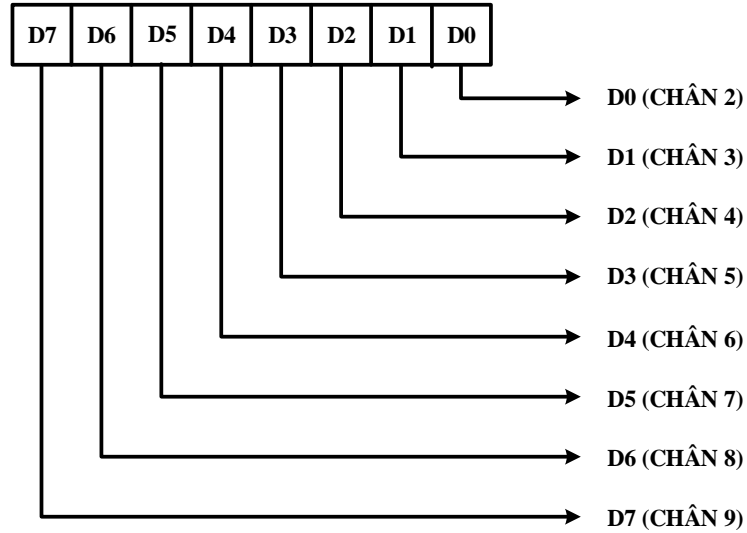
Sơ đồ nối dây và chiều tín hiệu giữa máy tính và máy in



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.

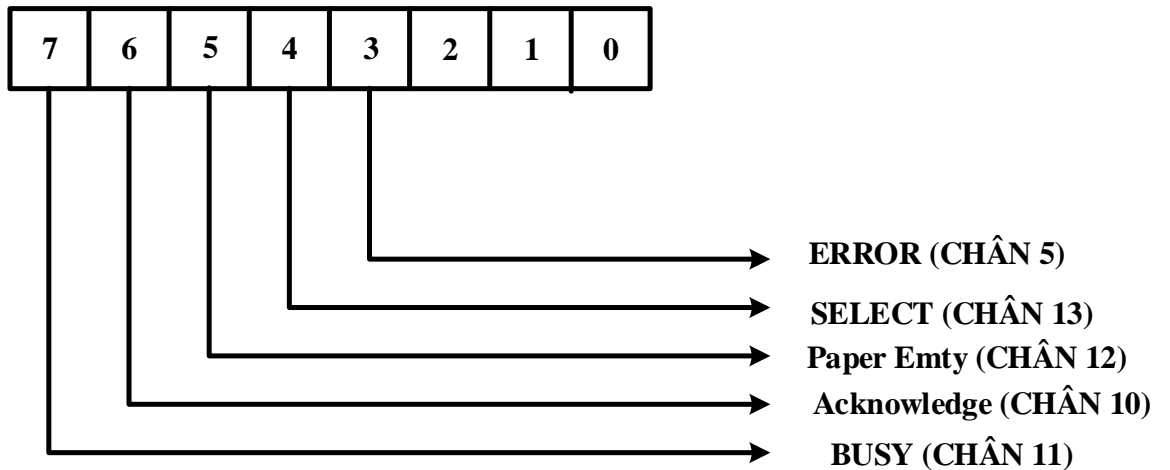
THANH GHI DỮ LIỆU ĐỊA CHỈ (278H, 378H, 2BCH, 3BCH)



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

Thanh ghi trạng thái (chỉ đọc):

THANH GHI TRẠNG THÁI ĐỊA CHỈ (279H, 379H, 2BDH, 3BDH)



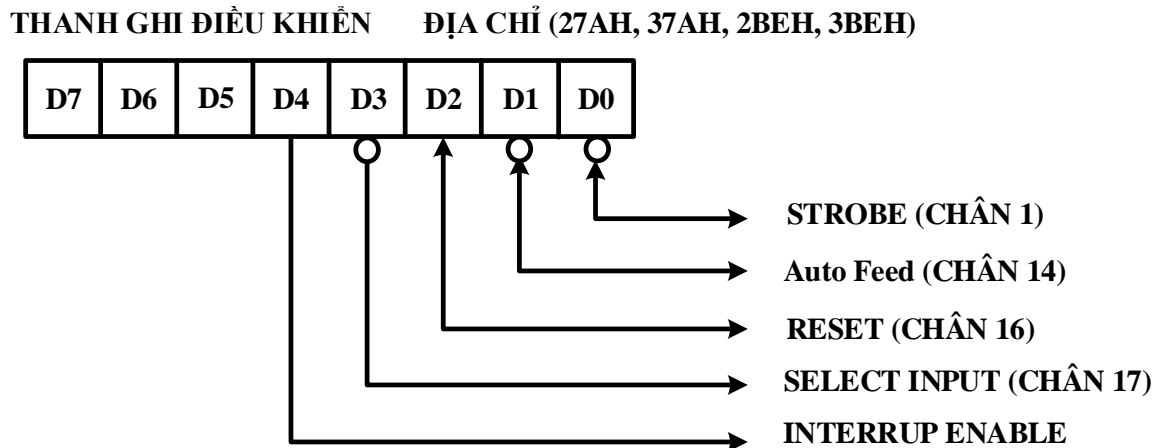
Hình 2.3. Sơ đồ định dạng của thanh ghi trạng thái

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:



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);`

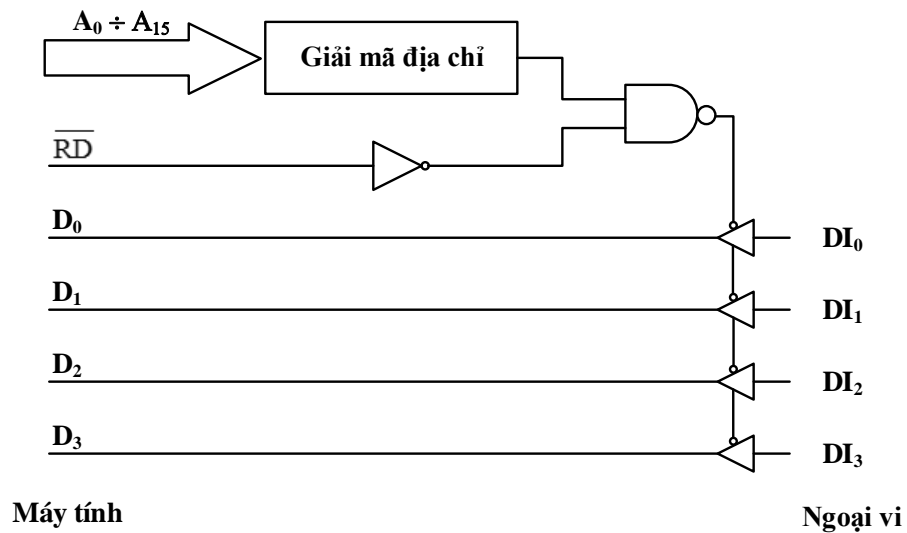
Để đọc thông tin của thanh ghi trạng thái và sắp xếp vào các biến đứng ở bên trái:

`status = inportb(0x379);`

2.3. Khối 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 đệm 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 (addrport);`

Pascal: `data1 = port (addrport);`

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

A0...A15 từ addrport.

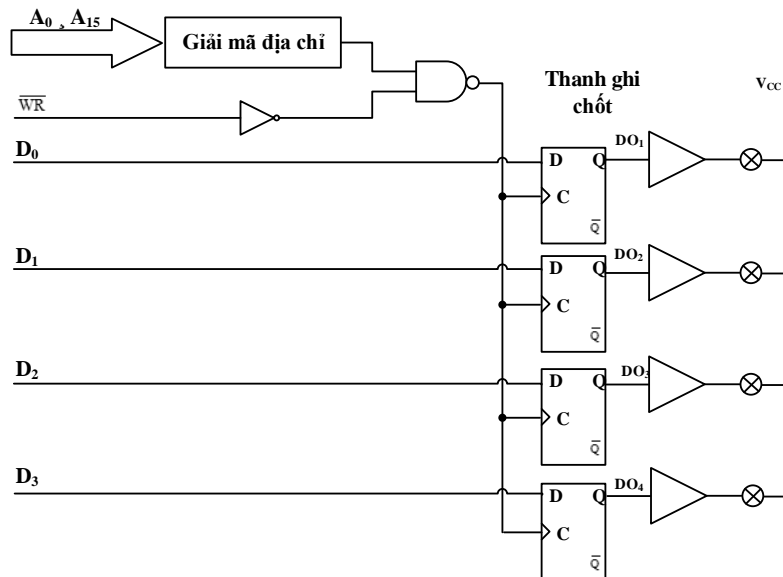
/RD kích hoạt ở mức 0.

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.

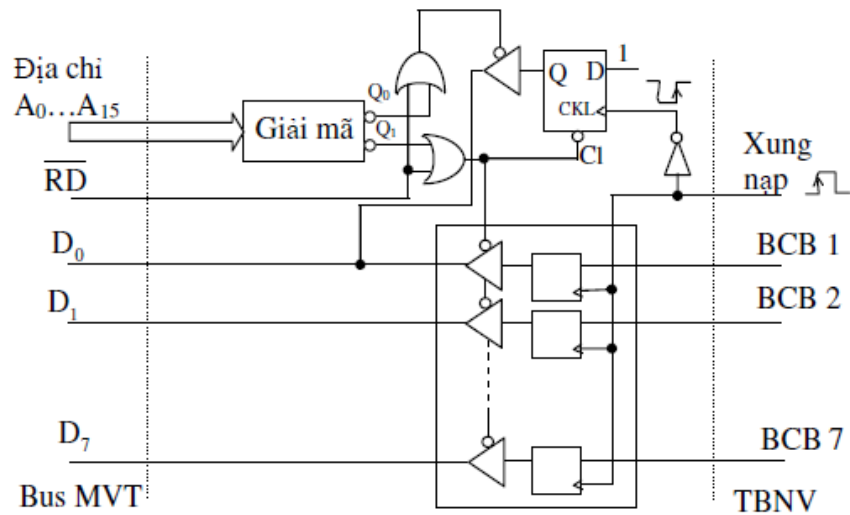
Bộ giải mã sẽ so sánh các giá trị trên đường địa chỉ với địa chỉ cổng cho trước, nếu trùng địa chỉ đầu ra bộ giải mã kích hoạt ở mức 1, lỗi ra cổng NAND = 0 → kích hoạt cho bộ đệm 3 trạng thái mở ra và ngoại vi nhận dữ liệu từ máy tính.

$D0...D7$ 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: Lôi 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 Triggers D sẽ xóa Q về mức logic 0.

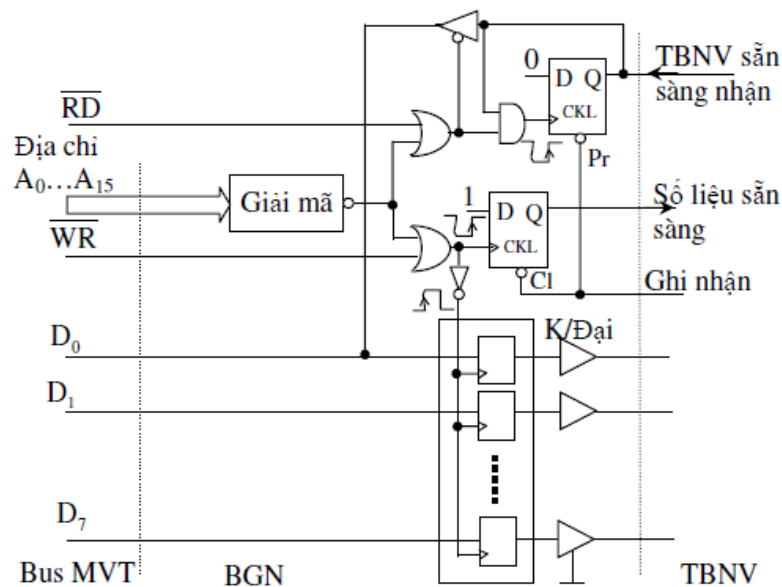
2.3.4. Cửa ra đơn giản có đối thoại

Khi ngoại vi sẵn sàng nhận dữ liệu, ngoại vi sẽ đưa ra 1 tín hiệu logic 1. VXL kiểm tra trạng thái của ngoại vi bằng:

```
#define BIT0 0x01;
do
data5 = inportb(addrport);
while (data5 & BIT0 != BIT0);
outportb(addrport, data6);
```

Lệnh inportb sẽ nạp D0 = 1 vào biến data5, biến này dùng để so sánh nhận biết ngoại vi đã sẵn sàng chưa.

Kết thúc lệnh inportb → I = đầu ra OR = 1 → đầu ra AND = 1 → Q = D = 0 → Mạch đếm 3 trạng thái ở trạng thái Z cao → cách ly chân Q và đường dữ liệu D0.



Hình 2.8. Sơ đồ cửa ra có đối thoại thiết bị ngoại vi.

Nếu thấy ngoại vi sẵn sàng nhận thì lệnh outportb tiếp theo sẽ gửi địa chỉ addrport và tín hiệu /WR = 0 để tạo xung ra mạch OR.

Sườn dương của xung này nạp dữ liệu data6 trên bus dữ liệu vào thanh ghi để xuất ra ngoại vi.

Sườn âm đưa tín hiệu Q = D = 1, báo cho ngoại vi là dữ liệu đã sẵn sàng.

Sau khi nhận dữ liệu, ngoại vi sinh ra tín hiệu ghi nhận để xóa tín hiệu dữ liệu sẵn sàng (về mức 0) và lập tín hiệu ngoại vi sẵn sàng nhận (lên mức 1), chờ VXL gửi dữ liệu tiếp theo.

Trao đổi dữ có đối thoại làm việc chắc chắn hơn trao đổi không đối thoại.

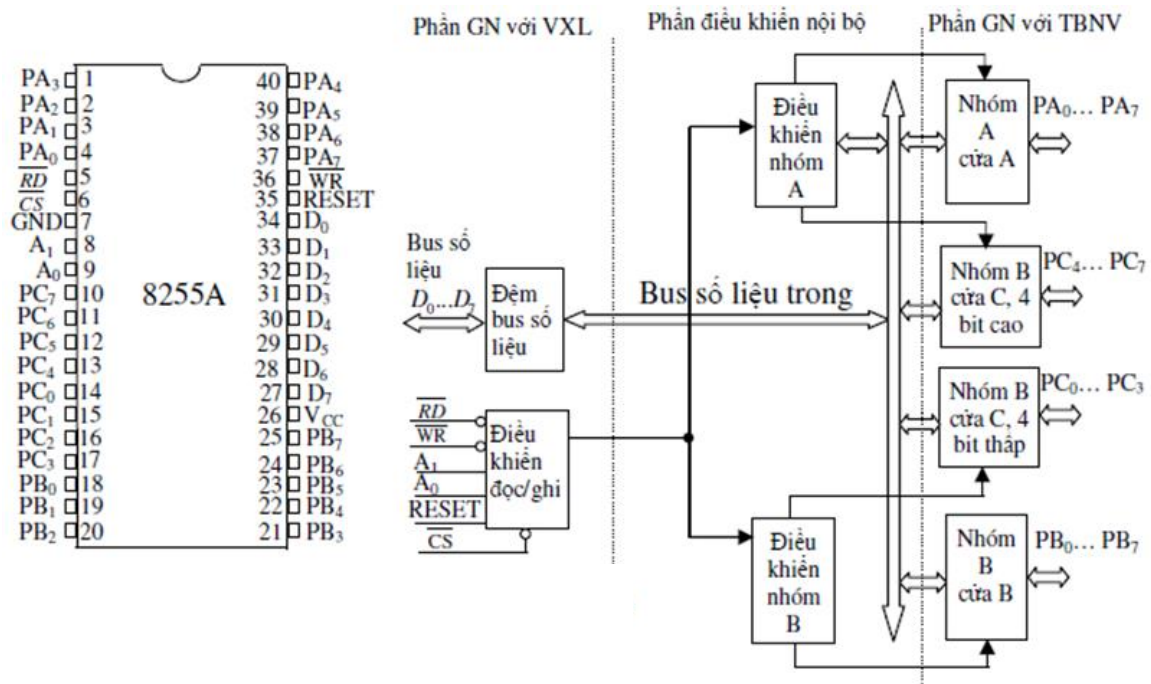
Ngoại vi có tốc độ hoạt động chậm nên dữ liệu đầu vào VXL có lúc không đúng khi ngoại vi có dữ liệu chưa ổn định.

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.

- /CS = 0 thì cho phép 8255A thông tin với CPU.
- /RD = 0 thì cho phép CPU đọc số liệu hoặc thông tin trạng thái từ 8255A.
- /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 công còn lại và được giải mã bên ngoài để chọn 8255A

Bảng 2.5. Bảng chức năng của 8255A

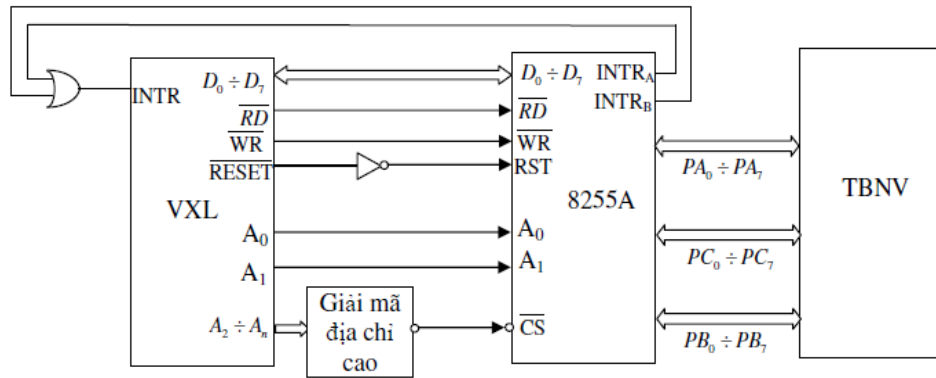
A1	A0	Chức năng
0	0	Cổng A
0	1	Cổng B
1	0	Cổng C
1	1	Thanh ghi lệnh

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

A1	A0	/RD	/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.

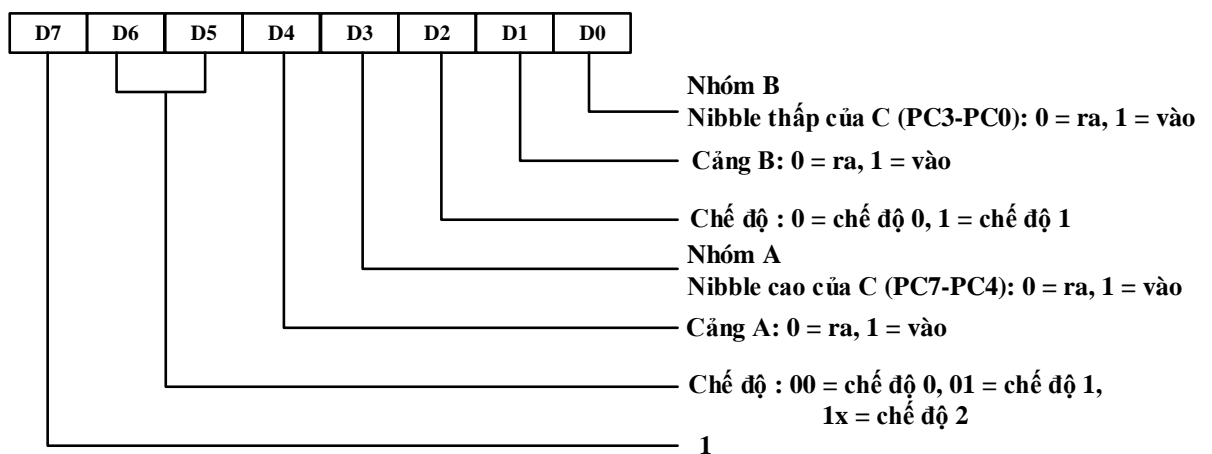
Vi mạch 8255 có ba chế độ hoạt động cơ bản (Mode) được chọn bởi phần mềm.

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.

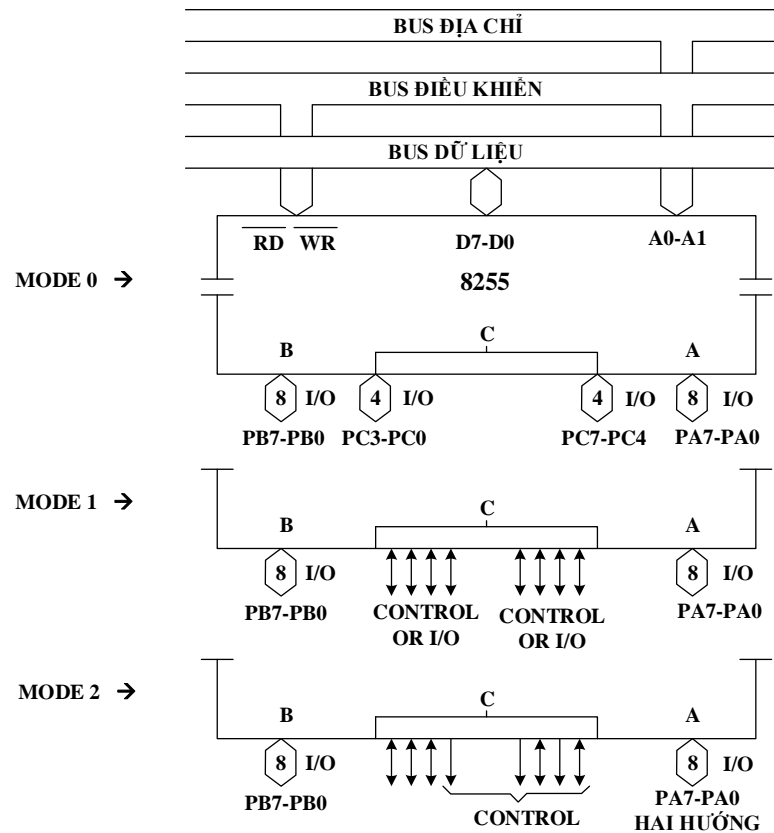
- Mode 0 : Chế độ vào/ ra cơ bản
- Mode 1 : Chế độ vào/ ra có hội thoại (Strobe Input/Output)
- Mode 2 : Chế độ vào/ ra bus hai chiều .

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



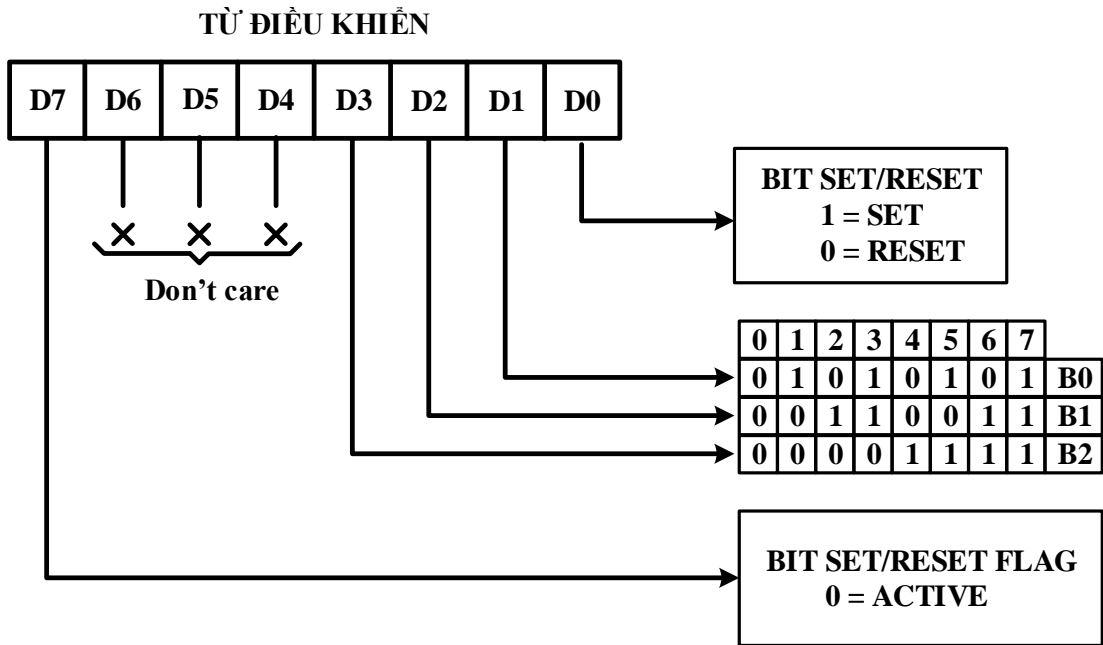
Hình 2.12. Các chế độ hoạt động của thanh ghi điều khiển

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.

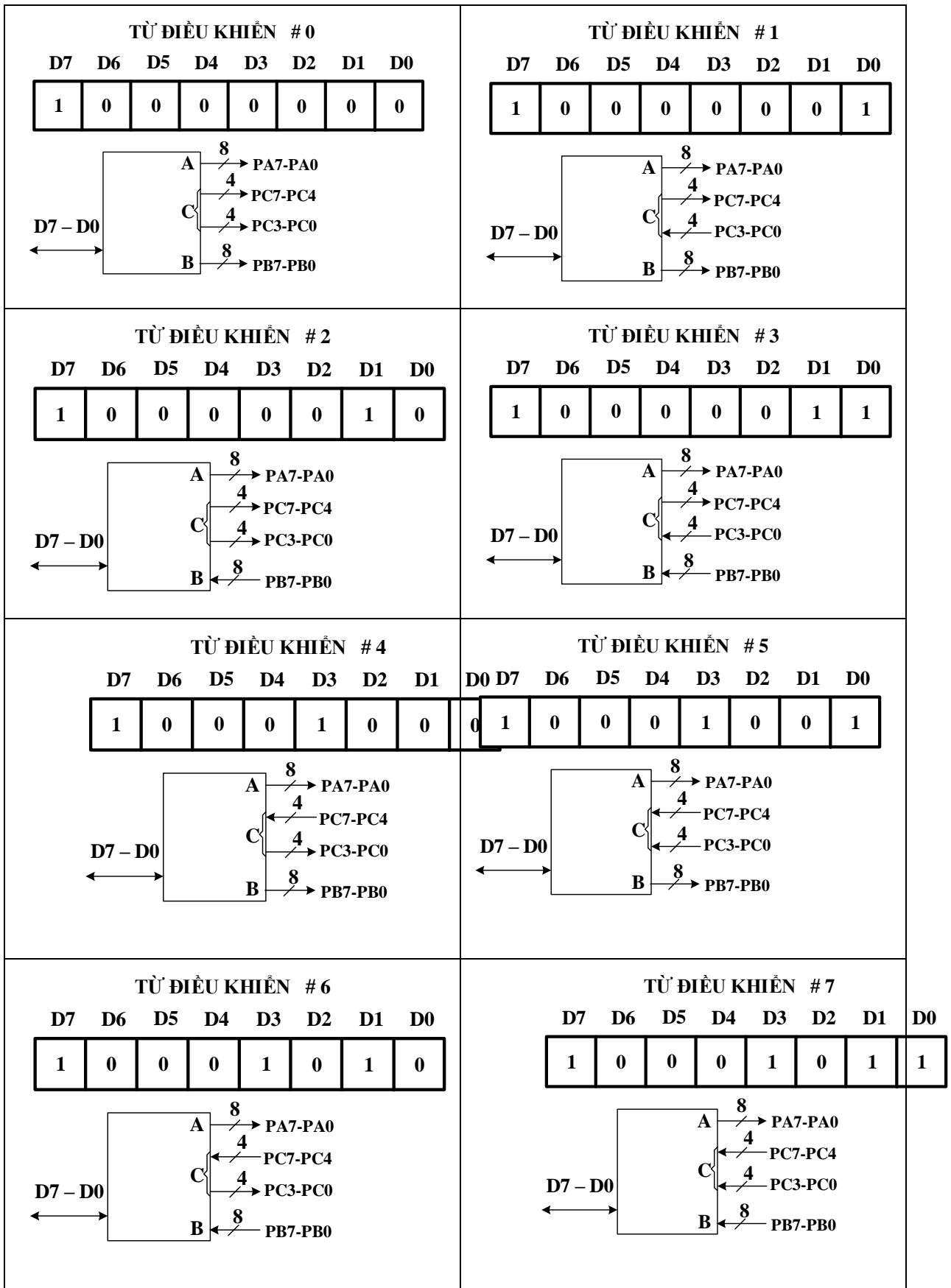
1/ Mode 0 (Basic input/output)

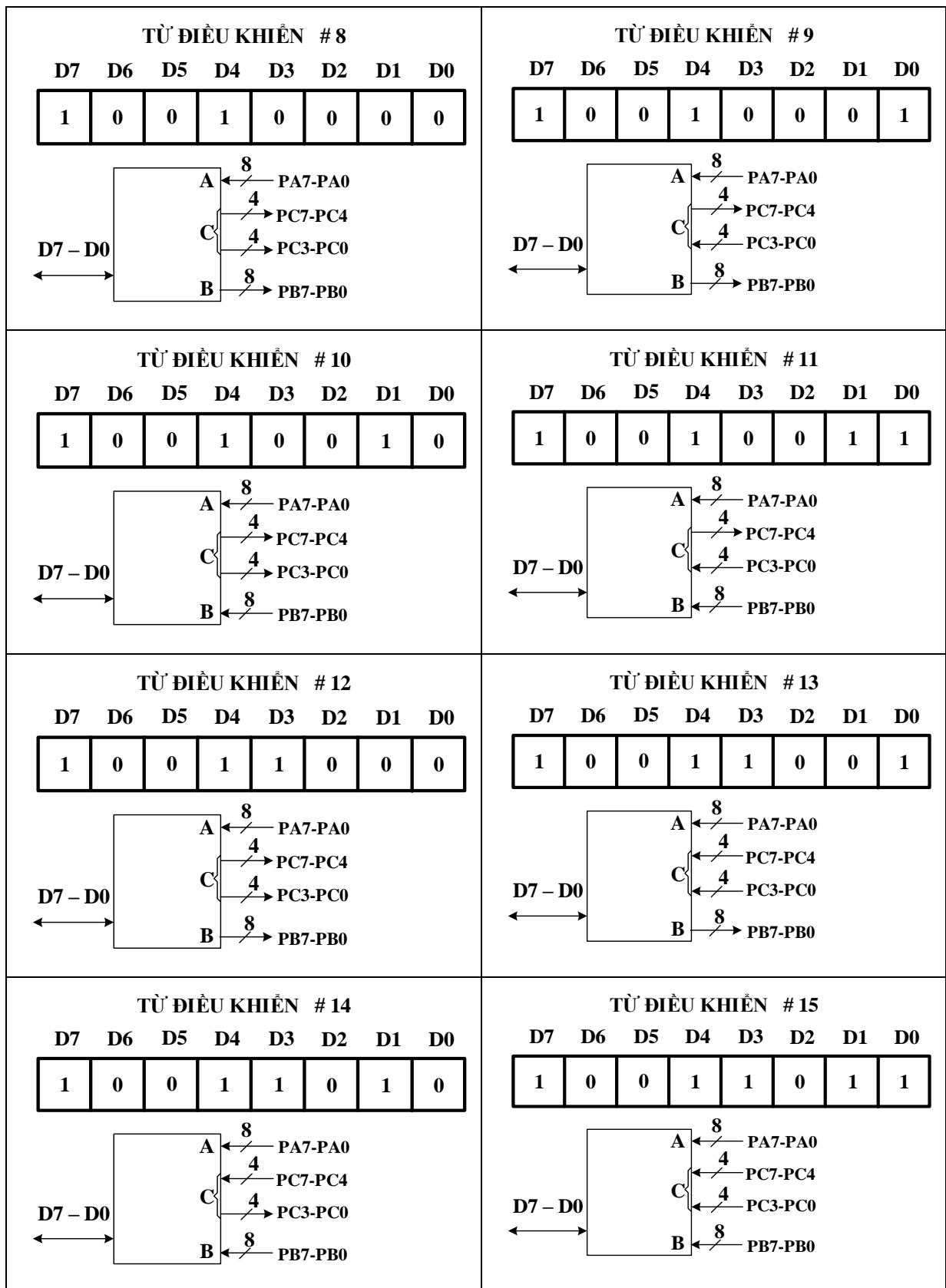
Đây là chế độ vào ra cơ bản của vi mạch, nó đảm bảo dữ liệu được đưa ra hoặc ghi vào các cổng riêng biệt.

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:





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.

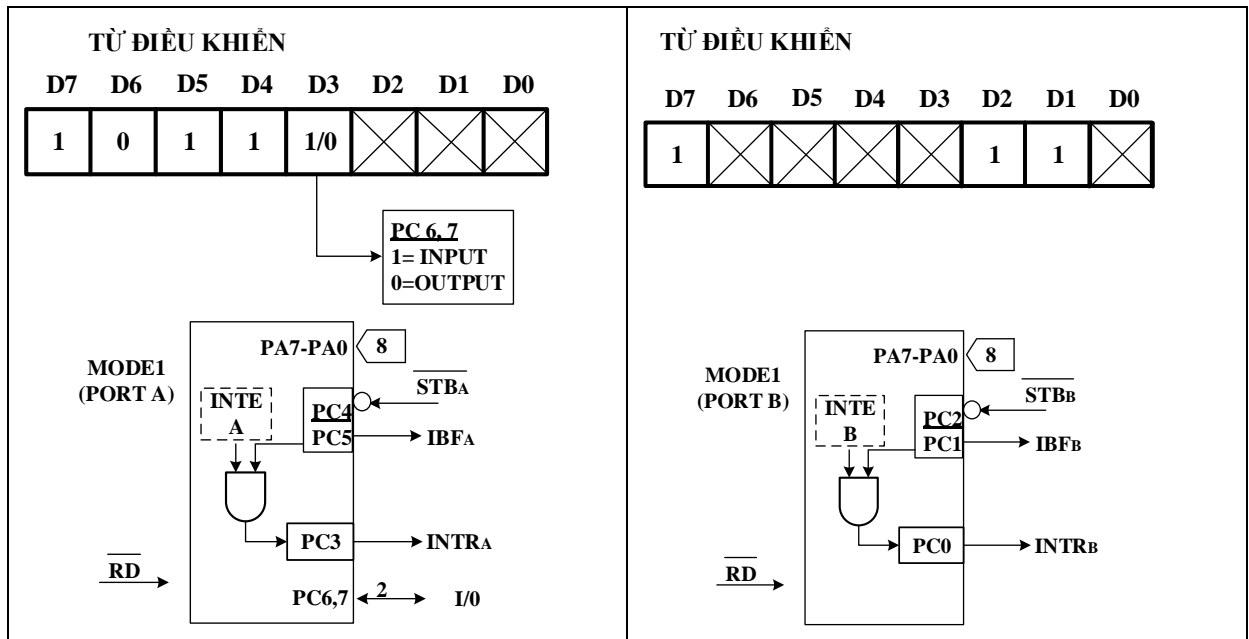
Tín hiệu IBF được xác lập khi tín hiệu STB đang ở mức thấp và được khởi tạo lại khi có sườn dương của đầu vào RD .

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 $STB = 1$.

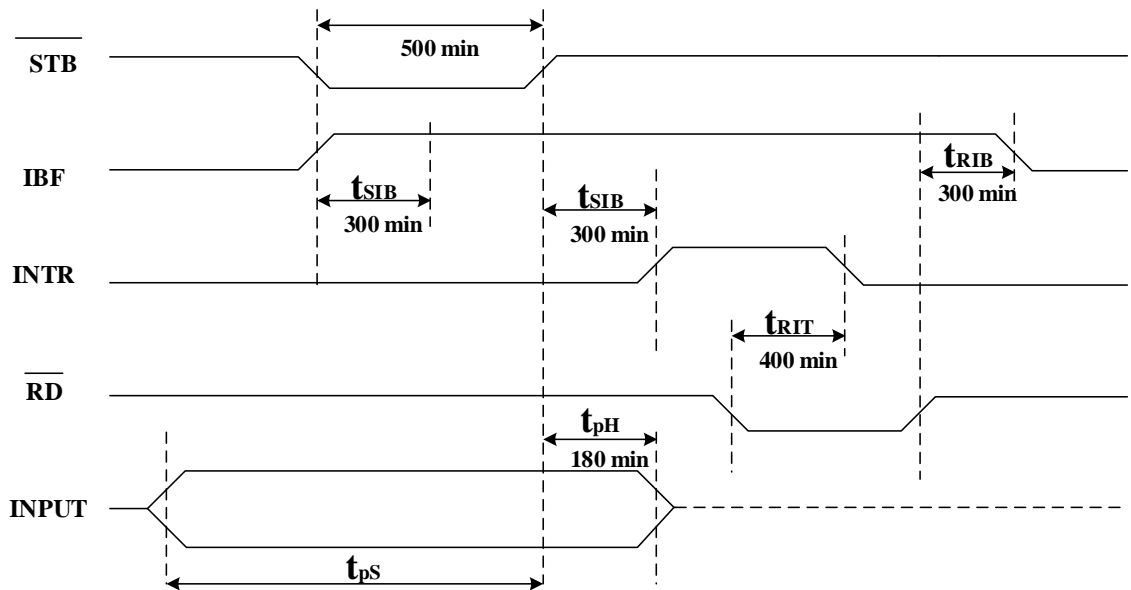
Tín hiệu $IBF=1$ và $INTE =1$: Tín hiệu này được khởi tạo lại khi ở sườn âm của tín hiệu RD. Chức năng này cho phép các thiết bị vào yêu cầu ngắt tới CPU một cách đơn giản bằng cách đưa dữ liệu của nó ra cổng.

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.

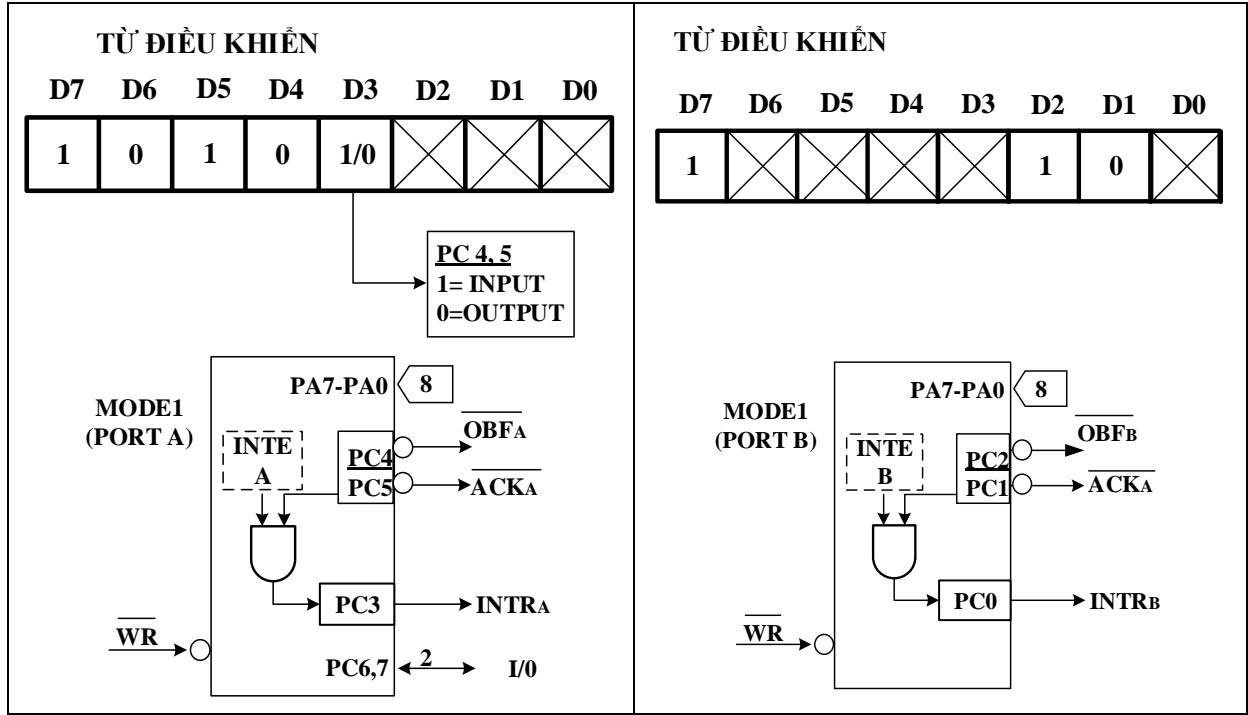
Các tín hiệu điều khiển ra được dùng trong chế độ ra.

/OBF (Output Buffer 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.

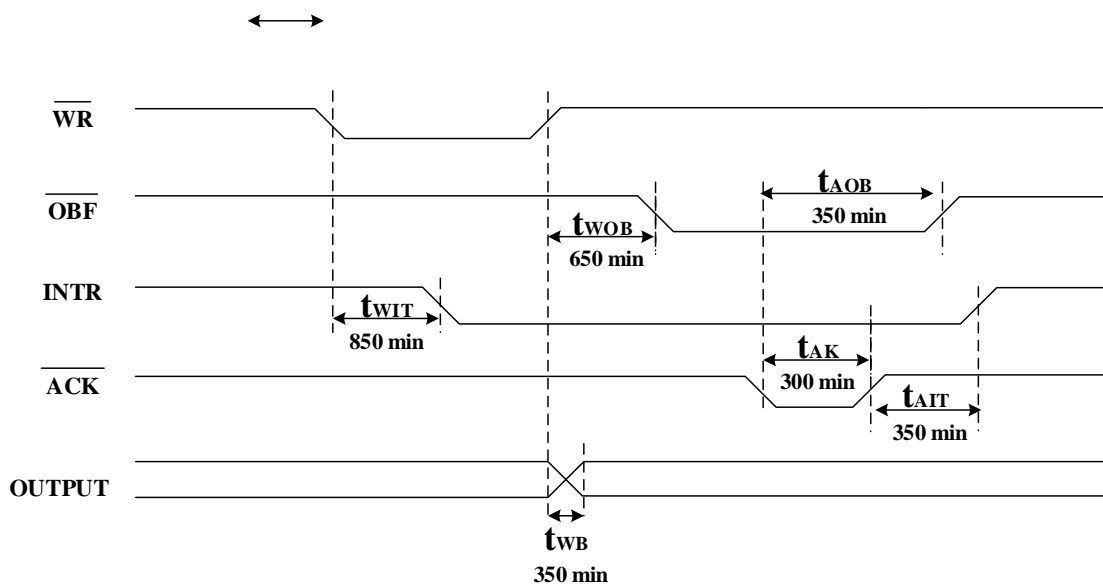
INTR (Interrupt Request): Mức cao của tín hiệu ra này được sử dụng để yêu cầu ngắt CPU khi một thiết bị ngoài đã nhận được dữ liệu truyền từ CPU. Tín hiệu INTR xác lập khi tín hiệu ACK = "1", OBF = "1" và INTE = "1". Tín hiệu này được khởi tạo lại ở sườn âm của tín hiệu WR .

Các từ lệnh của chế độ này như sau :



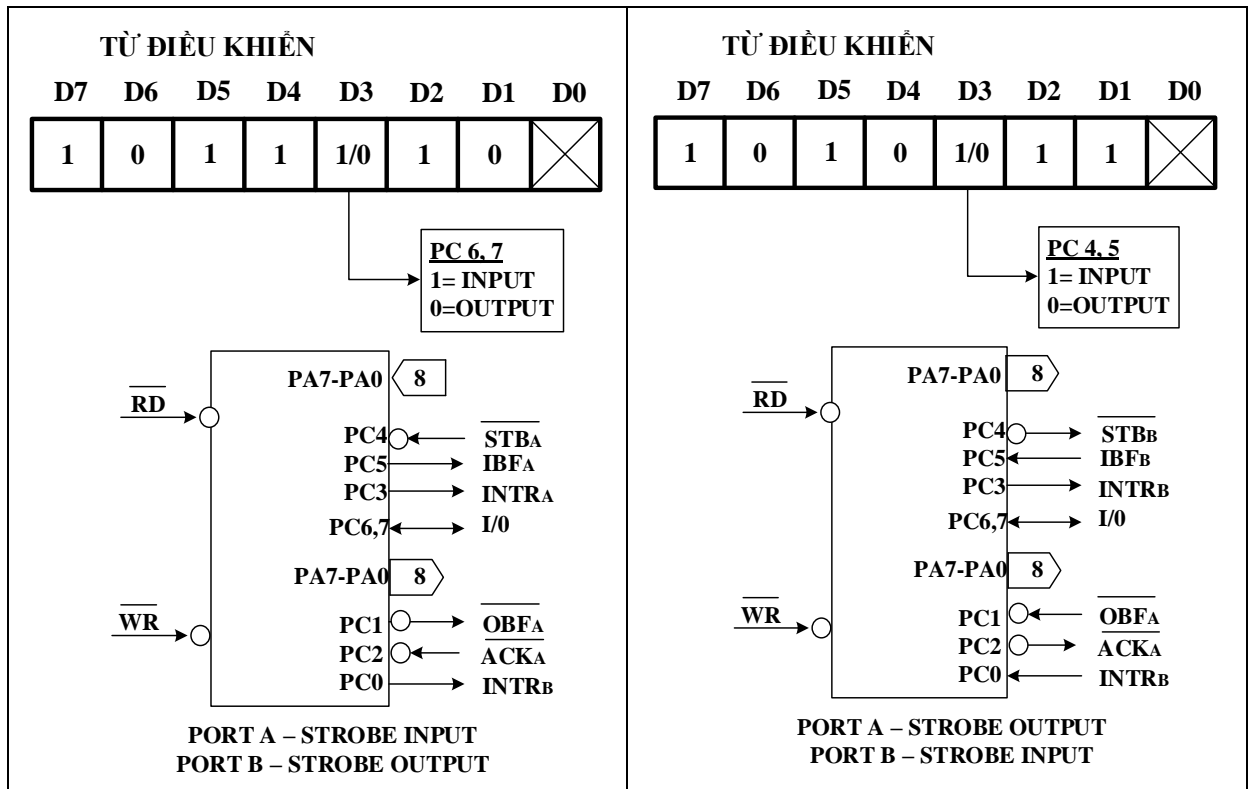
Hình 2.17. Sơ đồ minh họa các từ lệnh điều khiển dùng trong chế độ ra.

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 xoá bit PC4 .

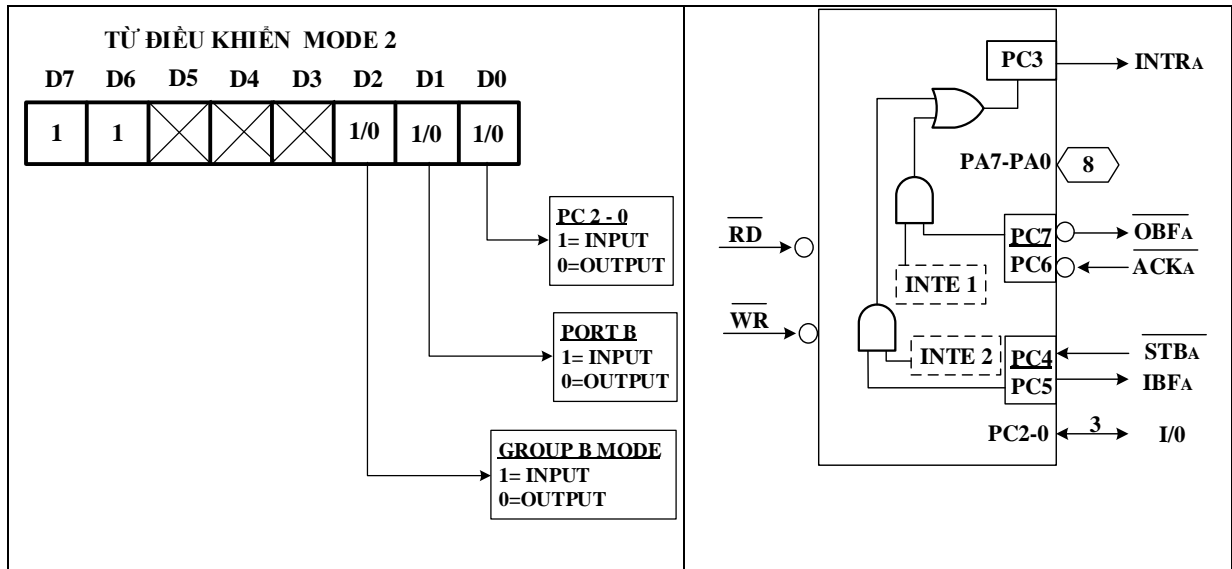
Các tín hiệu điều khiển vào .

/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 :



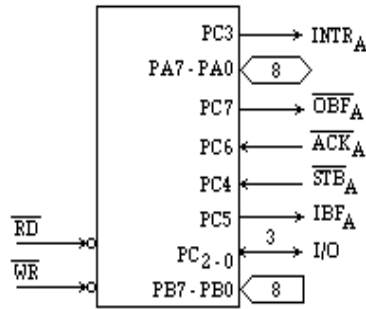
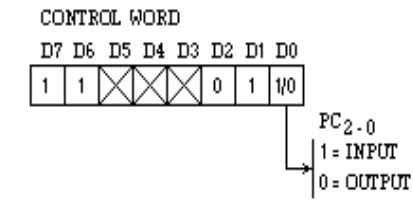
Hình 2.20. Sơ đồ minh họa các từ lệnh hoạt động trong mode 2.

4/ Kết hợp các chế độ hoạt động

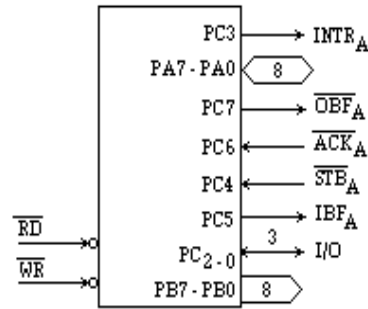
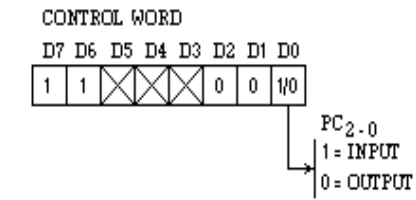
Ngoài việc hoạt động riêng rẽ theo từng chế độ, 8255 còn có khả năng hoạt động đồng thời kết hợp các chế độ khi không phải tất cả các bit trong cổng C được sử dụng để điều khiển hoặc dành cho trạng thái. Các bit còn lại có thể được sử dụng để thực hiện các chức năng sau :

Khi được lập trình là các đường vào tín hiệu :

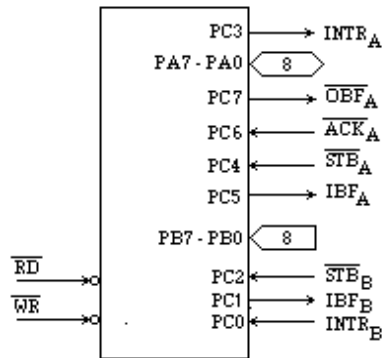
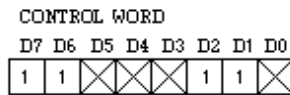
Tất cả các đường vào tín hiệu đều có thể được truy cập trong suốt quá trình đọc cổng C thông thường. Như trong hình vẽ minh họa sau :



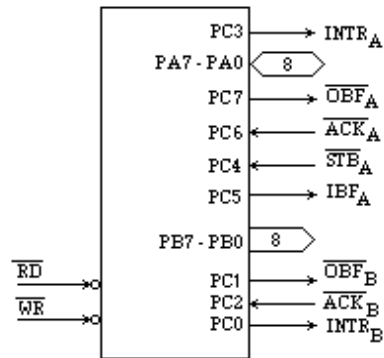
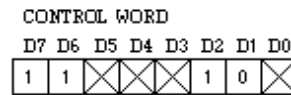
MODE 2 and MODE 0 (input)



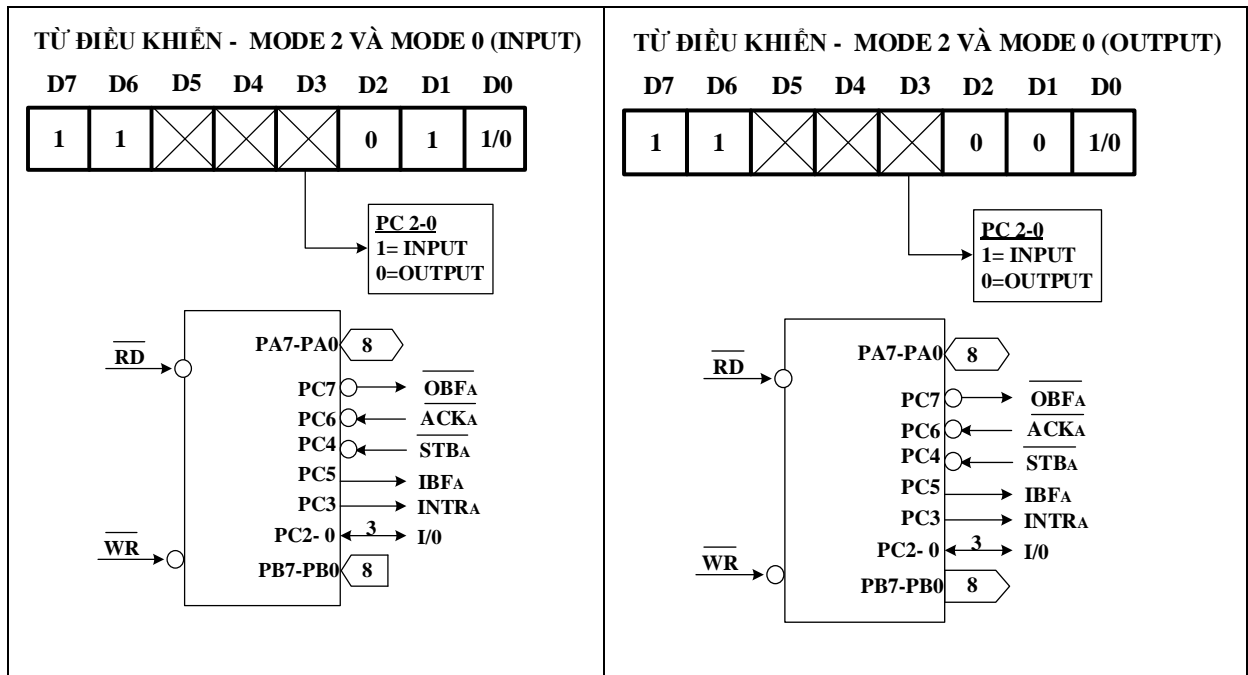
MODE 2 and MODE 0 (output)



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.

Độc 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.

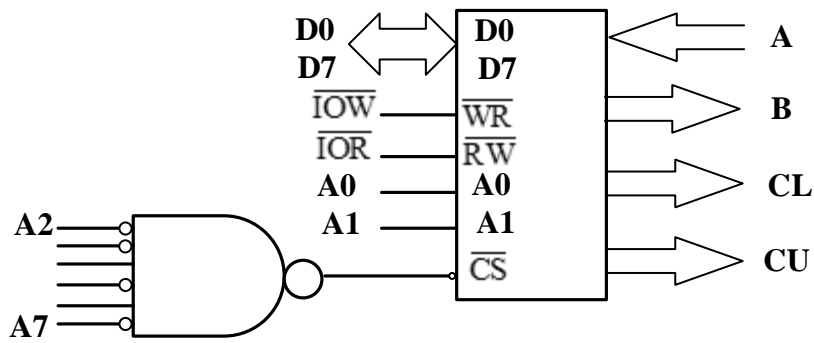
Độc nội dung của cổng C cho phép người lập trình kiểm tra trạng thái của các thiết bị ngoại vi và điều khiển quá trình trao đổi dữ liệu.

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:

```

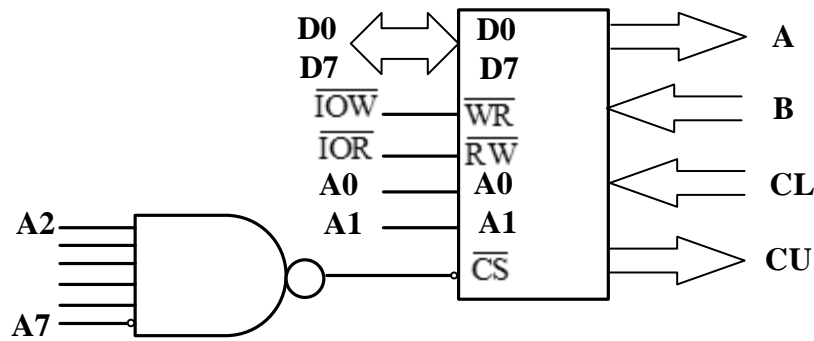
MOV AL,90H ; byte điều khiển PA = in, PB = out, PC = out
OUT 53H,AL ; gửi đến thanh ghi điều khiển
IN AL,50H ; gửi dữ liệu từ PA
OUT 51H,AL ; gửi đến PB
OUT 50H,AL ; gửi đến PC

```

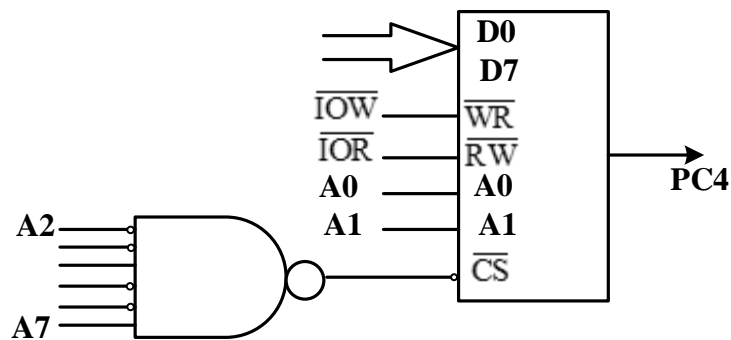
CÂU HỎI ÔN TẬP

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*)