GHÉP NỐI VÀ ĐIỀU KHIỂN THIẾT BỊ NGOẠI VI

Người soạn: **TS. Nguyễn Văn Minh Trí**
ThS. Lâm Tăng Đức
Các sinh viên lớp 03SK

TÀI LIỆU LƯU HÀNH NỘI BỘ

LỜI NÓI ĐẦU

Ngày nay, máy vi tính (MVT) đã được cải tiến, nâng cấp và thị trường hoá để đạt tốc độ xử lý cực nhanh, giá thành rẻ. Trong những thập niên qua, MVT đã được áp dụng khắp mọi nơi, thay thế cho các hệ thống cũ nhiều nhược điểm. Đặc biệt, trong hệ thống điện, điện công nghiệp, MVT đã được ứng dụng ngày càng nhiều, mang lại nhiều tính năng ưu việt cho hệ thống. Do đó yêu cầu phát triển lý thuyết về hệ thống có sử dụng MVT đã mở ra nhiều lĩnh vực học mới.

Một ví dụ cụ thể của việc ứng dụng MVT trong công nghiệp là các hệ thống đo lường và điều khiển sử dụng MVT. Trong các hệ thống này, MVT nhận các dữ liệu đo đạt trạng thái vật lý của hệ thống, vi dụ như nhiệt độ, áp suất, điện áp …, và tính toán để đưa ra những quyết định, tín hiệu điều khiển trở lại hệ thống, nhằm đạt được các yêu cầu điều khiển đề ra của hệ thống.

Muốn thiết kế, vận hành và bảo quản tốt hệ đo-điều khiển công nghiệp, người kỹ sư cần nắm vững về kỹ thuật máy tính, cách viết chương trình, kỹ thuật ghép nối thiết bị ngoại vi, việc xử lý số liệu, lý thuyết điều khiển số…

Môn học ghép nối và điều khiển thiết bị ngoại vi (TBNV) nhằm cung cấp các kiến thức cơ bản trong việc tìm hiểu, thiết kế mạch ghép nối ngoại vi với MVT. Các mạch thiết kế phục vụ cho việc đo lường, điều khiển, giao tiếp giữa MVT với MVT, MVT với hệ vi điều khiển.

MỤC LỤC

CHƯƠNG 1: CƠ SỞ GHÉP NỐI THIẾT BỊ NGOẠI VI

1.1. Giới thiệu chung
- 1.1.1. Cấu trúc hệ thống
- 1.1.2. Nhiệm vụ của bộ ghép nối
- 1.1.3. Thủ tục trao đổi dữ liệu của máy vi tính

1.2. Giới thiệu một số vi mạch dùng trong bộ ghép nối
- 1.2.1. Các mạch logic cơ bản
- 1.2.2. Các mạch lật
- 1.2.3. Các mạch thanh ghi đệm
- 1.2.4. Các mạch giải mã địa chỉ
- 1.2.5. Các mạch truyền dữ liệu

CHƯƠNG 2: GHÉP NỐI VÀO RA SONG SONG ĐIỀU KHIỂN BẰNG CHƯƠNG TRÌNH

2.1 Các lệnh vào ra dữ liệu
- 2.1.1 Các lệnh vào ra bằng hợp ngữ
- 2.1.2 Các lệnh vào ra bằng Turbo C
- 2.1.3 Các lệnh vào ra bằng Turbo Pascal

2.2 Ghép nối song song đơn giản: có hay không có đối thoại
- 2.2.1 Cửa vào đơn giản không đối thoại
- 2.2.2 Cửa ra đơn giản không đối thoại
- 2.2.3 Cửa vào đơn giản có đối thoại
- 2.2.4 Cửa ra đơn giản có đối thoại

2.3 Ghép nối song song điều khiển bằng chương trình
- 2.3.1 Sơ đồ khối và chức năng các khối của 8255A
- 2.3.2 Các chế độ làm việc của 8255A
- 2.3.3 Ghép nối với 8255A
- 2.3.4 Lập trình cho 8255A

CHƯƠNG 3: GHÉP NỐI TRAO ĐỔI DỮ LIỆU NỐI TIẾP

3.1 Giới thiệu chung về trao đổi dữ liệu nối tiếp
- 3.1.1 Yêu cầu trao đổi dữ liệu nối tiếp
- 3.1.2 Mạch trao đổi dữ liệu nối tiếp

3.1.3 Thủ tục trao đổi dữ liệu nối tiếp
3.2 Mạch thu phát dị bộ vạn năng UART 8250
3.2.1 Sơ đồ khối và chức năng các khối của UART 8250
3.2.2 Ghép nối với UART 8250
3.2.3 Lập trình cho UART 8250

CHƯƠNG 4: GHÉP NỐI ĐIỀU KHIỂN NGẮT

4.1 Khái niệm và phân loại ngắt
4.1.1 Khái niệm
4.1.2 Phân loại ngắt
4.2 Xử lý ngắt của nhiều thiết bị ngoại vi
4.2.1 Cho phép hoặc cấm ngắt
4.2.2 Sắp xếp ưu tiên ngắt
4.2.3 Xác định nguồn gây ngắt
4.2.4 Tạo vector ngắt
4.3 Vi mạch điều khiển ngắt 8259A
4.3.1 Sơ đồ khối và chức năng các khối của 8259A
4.3.2 Cấu trúc hoạt động của vi mạch 8259A
4.3.3 Ghép nối với 8259A
4.3.4 Lập trình cho 8259A

CHƯƠNG 5: GHÉP NỐI VÀO RA ĐIỀU KHIỂN BẰNG DMA

5.1 Khái niệm DMA cơ bản
5.1.1 Khái niệm DMA
5.1.2 Hoạt động DMAC (DMA Controller) cơ bản
5.1.3 Các chế độ trao đổi dữ liệu DMA
5.2 Bộ điều khiển DMA – 8237A
5.2.1 Sơ đồ khối và chức năng các khối của DMAC 8237A
5.2.2 Ghép nối với 8237A trong hệ vi xử lý 8088
5.2.3 Lập trình cho 8237A

CHƯƠNG 6: GHÉP NỐI TƯƠNG TỰ-SỐ VÀ SỐ-TƯƠNG TỰ

6.1. Giới thiệu một số mạch khuếch đại thuật toán
6.1.1 Các tham số cơ bản của mạch khuếch đại thuật toán
6.1.2 Các sơ đồ cơ bản của bộ khuếch đại thuật toán
6.2 Bộ chuyển đổi số-tương tự DAC
6.2.1 Hoạt động của DAC
6.2.2 Các tham số cơ bản của bộ chuyển đổi của DA
6.2.3 Các mạch DAC điển hình
6.2.4 Ghép nối với DAC
6.2.5 Lập trình xuất dữ liệu ra DAC
6.3 Bộ biến đổi tương tự-số ADC

6.3.1 Nguyên tắc làm việc của ADC
6.3.2 Các tham số cơ bản của bộ chuyển đổi AD
6.3.3 Các phương pháp chuyển đổi AD
6.3.4 Ghép nối với ADC
6.3.5 Lập trình nhận dữ liệu từ ADC

CHƯƠNG 7: ỨNG DỤNG MÁY VI TÍNH TRONG ĐO LƯỜNG VÀ ĐIỀU KHIỂN

7.1. Giới thiệu về cấu trúc máy tính
7.2 Thiết kế mạch ghép nối giữa hệ thống tự động hóa với các cổng của máy vi tính

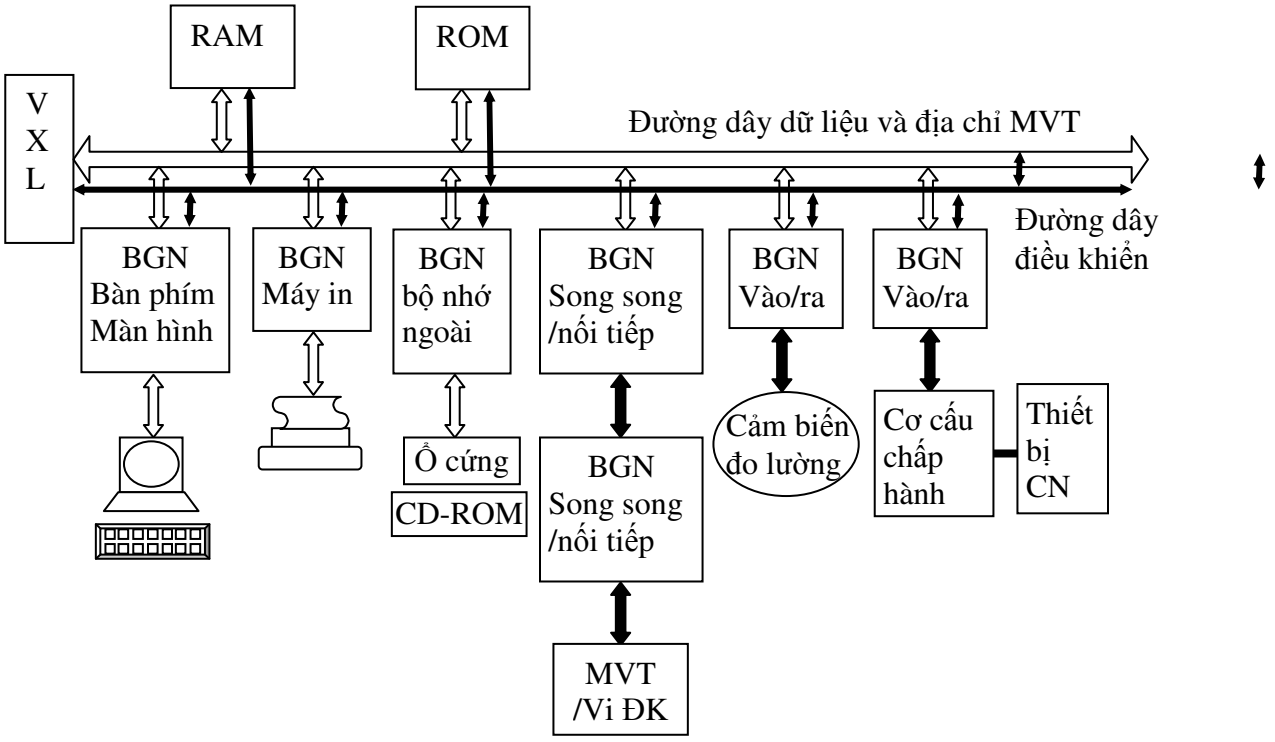
CHƯƠNG 1: CƠ SỞ GHÉP NỐI THIẾT BỊ NGOẠI VI

1.1. Giới thiệu chung

1.1.1 Cấu trúc hệ thống

Máy vi tính hay hệ vi xử lý đều có cấu trúc chung do Phôn Nơi-Man đề xuất gồm khối xử lý trung tâm (CPU), bộ nhớ (M) và các cửa vào/ra (I/O), như trên hình 1.1. Ngoài ra, MVT còn cần phải trao đổi dữ liệu và môi trường bên ngoài, ví dụ giao tiếp với người sử dụng qua bàn phím-màn hình, trao đổi dữ liệu với các thiết bị ngoài thông dụng, các thiết bị ngoài trong hệ đo-điều khiển, và các MVT khác trong mạng. Do đó các bộ ghép nối thiết bị ngoại vi (BGN TBNV) được xây dựng, gồm:

- BGN các thiết bị vào chuẩn như bàn phím, con chuột…
- BGN các thiết bị ra chuẩn như màn hình, máy in…
- BGN các bộ nhớ ngoài chuẩn như ổ cứng, ổ CD-ROM…
- BGN với MVT khác trong mạng nhiều MVT.
- BGN với hệ vi điều khiển, hệ vi xử lý.
- BGN hệ đo lường - điều khiển:



Hình 1.1: Cấu trúc hệ GN trao đổi dữ liệu tin giữa MVT và TBNV

Trong đó: VXL là viết tắt của vi xử lý
RAM là random-access memory
ROM là read-only memory
BGN là bộ ghép nối
CD-ROM là compact disk read-only memory
CN là công nghiệp
ĐK là điều khiển

Đặt biệt trong hệ đo lường - điều khiển, MVT nhận dữ liệu trạng thái vật lý của hệ thống (nhiệt độ, áp suất, điện áp, dòng điện…) dưới dạng tín hiệu điện, từ đầu dò các bộ cảm biến (sensor), bộ chuyển đổi (tranducer), bộ phát hiện (detector). Và MVT còn nhận thông tin về trạng thái sẵn sàng hay bận của thiết bị đo.

MVT sau đó đưa tín hiệu chấp nhận trao đổi dữ liệu với TBNV, thu thập và xử lý dữ liệu, tính toán các tín hiệu điều khiển đưa ra các cơ cấu chấp hành (các van đóng mở, các rơle trong mạch điện, các mạch động lực điều tốc động cơ điện…), hay đưa ra các thông số kỹ thuật cho thiết bị.

Ngoài ra, MVT còn cần lưu trữ dữ liệu trên ổ cứng, đĩa compact (CD-ROM) để tra cứu lúc cần, hiển thị kết quả đo dưới dạng bảng số liệu, dạng đồ thị hay hình vẽ đồ hoạ trên màn hình.

1.1.2 Nhiệm vụ của bộ ghép nối

TBNV được ghép nối với MVT thông qua các phần thích ứng về công nghệ và thích ứng logic như hình 1.2. GN công nghệ làm nhiệm vụ điều chỉnh mức tín hiệu giữa công nghệ sản xuất TBNV và công nghệ sản xuất các mạch cấu trúc nên BUS của MVT. GN logic làm nhiệm vụ tạo các tín hiệu điều khiển TBNV từ những tín hiệu trên BUS hệ thống. Nhiệm vụ của BGN là phối hợp trao đổi dữ liệu giữa MVT và TBNV.

1. *Phối hợp về mức và công suất tín hiệu*

- Mức tín hiệu của MVT thường là mức TTL (0V, 5V), trong khi TBNV có mức điện thoại (±15V, ±48V) hay mức điện công nghiệp (220V/380V).
- Công suất đường dây MVT nhỏ (cỡ chục mA), trong khi công suất cần cho TBNV là rất lớn tùy theo từng hệ thống.

Do đó BGN phải biến đổi điện áp và khuếch đại công suất cho phù hợp giữa MVT và TBNV. Thường dùng vi mạch 3 trạng thái để đưa dữ`liệu ra, đưa dữ liệu vào. Do đó vi mạch sẽ ở trạng thái trở kháng cao khi không có trao đổi dữ liệu, để cô lập TBNV với MVT, để không tiêu thụ một chút công suất nào của đường dây và để bảo vệ MVT.

2. *Phối hợp về dạng dữ liệu*

- Dữ liệu trao đổi của MVT luôn là song song ở dạng số nhị phân, có thể truyền theo 8 bit, 16 bit, 32 bit, 64 bit.
- Dữ liệu của TBNV có thể song song 8 bit và 16 bit, nối tiếp hoặc tín hiệu liên tục.

Do đó BGN sẽ chuyển đổi dạng dữ liệu cho phù hợp giữa MVT và TBNV, ví dụ chuyển đổi từ tín hiệu liên tục sang tín hiệu số.

3. *Phối hợp về tốc độ trao đổi dữ liệu*

- MVT thường hoạt động với tốc độ cao (tần số lên tới hàng trăm MHz) trong khi TBNV thường hoạt động chậm hơn nhiều.

Do đó BGN nhận và lưu dữ liệu từ MVT rồi truyền cho TBNV theo phịp chậm của TBNV, giải phóng cho MVT làm nhiệm vụ khác (phục vụ TBNV khác, chạy chương trình xử lý số liệu hay hiển thị trên màn hình). Tương tự, BGN nhận dữ liệu của TBNV và chờ MVT đọc dữ liệu vào.

4. *Phối hợp về phương thức trao đổi dữ liệu*

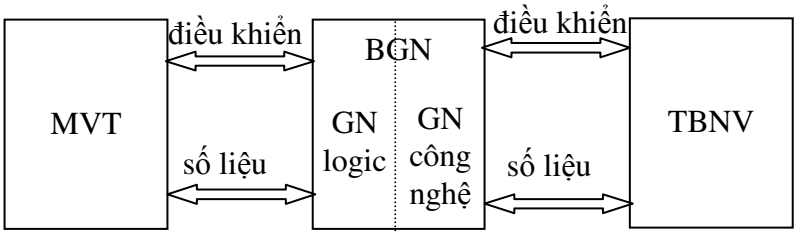
- Việc đọc/xuất dữ liệu do MVT khởi xướng:

Khi đó, MVT đưa lệnh điều khiển tới khởi động BGN hay TBNV, MVT đọc trạng thái của TBNV. Nếu trạng thái TBNV chưa sẵn sàng, MVT sẽ chờ.

Nếu TBNV ở trạng thái sẵn sàng, MVT sẽ tiến hành đọc/xuất dữ liệu.

- Việc đọc/xuất dữ liệu do TBNV khởi xướng:

Khi đó, TBNV đưa yêu cầu trao đổi dữ liệu tới bộ phận xử lý ngắt của BGN. Nếu có nhiều TBNV đồng thời đưa ra yêu cầu, BGN sẽ sắp xếp theo thứ tự ưu tiên định sẵn, rồi đưa từng yêu cầu tới MVT. MVT sau khi nhận tín hiệu yêu cầu, sẽ chuẩn bị và đưa tín hiệu xác nhận sẵn sàng trao đổi. Đến lược BGN nhận và truyền tín hiệu sẵn sàng đến cho TBNV. Sau đó, MVT và TBNV sẽ trao đổi dữ liệu qua trung gian là BGN.



Hình 1.2: GN giữa MVT và TBNV

1.1.3 Thủ tục trao đổi dữ liệu của máy vi tính

MVT trao đổi dữ liệu với TBNV theo một trong hai chế độ:

- Chế độ chương trình:

Gồm các lệnh VÀO, RA và CHUYỂN dữ liệu giữa các thanh ghi.

- Chế độ truy cập thẳng tới bộ nhớ:

BGN lúc này điều khiển sự trao đổi giữa TBNV và bộ nhớ qua trung gian BGN. Các của vào ra của VXL ở trạng thái trở kháng cao. BGN điều khiển mọi hoạt động của bộ nhớ và TBNV, bao gồm:

. Phát địa chỉ cho bộ nhớ và TBNV.

. Phát lệnh đọc $\overline{RD}$ hay ghi $\overline{WR}$ số liệu.

. Các số liệu đọc, ghi được trao đổi giữa bộ nhớ và TBNV thông qua các thanh ghi đệm của BGN.

Ở chế độ trao đổi dữ liệu theo chương trình, việc thực hiện có thể theo một trong ba phương pháp sau (hình 1.3):

- Đồng bộ hay không đối thoại.
- Không đồng bộ hay có đối thoại.
- Ngắt chương trình.

1. Trao đổi đồng bộ:

Sau khi khởi động TBNV, MVT không cần quan tâm tới TBNV có sẵn sàng trao đổi dữ liệu hay không mà đưa luôn lệnh trao đổi dữ liệu. Do đó TBNV phải luôn sẵn sàng trao đổi dữ liệu. Phương pháp trao đổi dữ liệu này có ưu điểm là nhanh, không tốn thời gian chờ đợi. Tuy nhiên nó có nhược điểm là thiếu tin cậy, đôi khi bị mất dữ liệu vì có thể có sự cố làm TBNV chưa sẵn sàng trao đổi.

2. Trao đổi không đồng bộ:

Trình tự trao đổi diễn ra như sau:

- MVT đưa tín hiệu điều khiển tới BGN.
- MVT chờ và kiểm tra trạng thái sẵn sàng trao đổi của TBNV. Nếu chưa sẵn sàng thì đọc là kiểm tra lại trạng thái này.

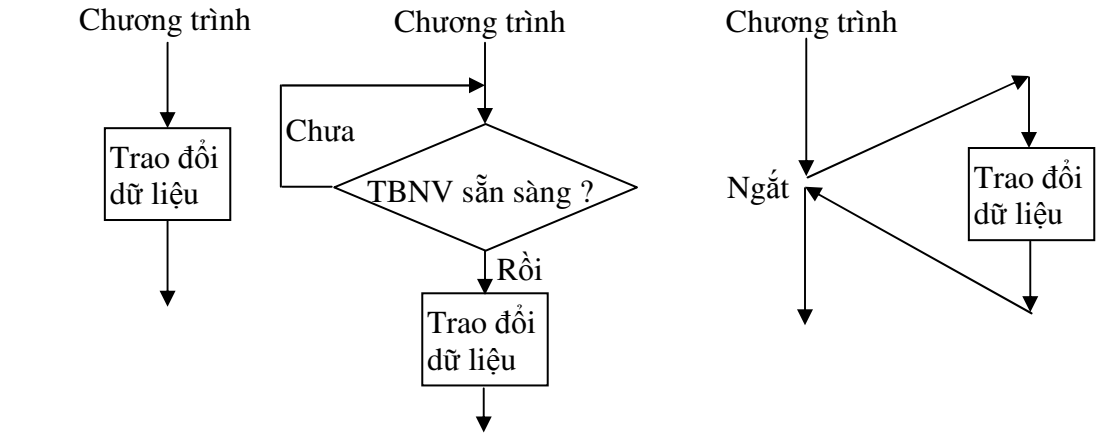
- MVT trao đổi dữ liệu với TBNV khi đã sẵn sàng.

Phương pháp này thường dùng vì tốc độ trao đổi dữ liệu của TBNV chậm hơn so với MVT. Phương pháp trao đổi dữ liệu này có ưu điểm là rất tin cậy, nhưng lại tốn thời gian sử dụng MVT.

3. Trao đổi theo ngắt chương trình:

Phương pháp này khắc phục được nhược điểm của phương pháp trên. Trình tự như sau:

- MVT đang thực hiện chuỗi lệnh của chương trình nào đó.
- TBNV có yêu cầu trao đổi dữ liệu, sẽ gởi tín hiệu yêu cầu trao đổi (yêu cầu ngắt - $\overline{\text{INTR}}$).
- MVT đưa tín hiệu chấp nhận (xác nhận ngắt – INTA).
- Chương trình chính bị ngắt, MVT chuyển sang chương trình con phục vụ ngắt, tức chương trình con trao đổi dữ liệu cho TBNV đã yêu cầu.
- Chương trình chính lại được tiếp tục thực hiện ở chỗ bị ngắt.



a) Đồng bộ b) Không đồng bộ c) Ngắt chương trình

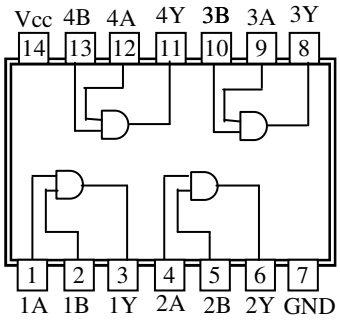
Hình 1.3 : Trao đổi dữ liệu theo chương trình

1.2. Giới thiệu một số vi mạch dùng trong bộ ghép nối

Thông thường BGN có các thanh chốt dữ liệu, bộ đệm 3 trạng thái để nối với BUS và các mạch lật tạo tín hiệu đối thoại.

1.2.1 Các mạch logic cơ bản

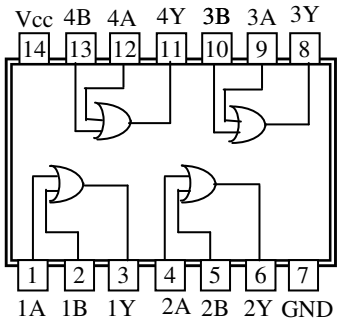
Mạch AND
- SN7408
- SN7409



Bảng trạng thái

A	B	Y
L	L	L
H	L	L
L	H	L
H	H	H

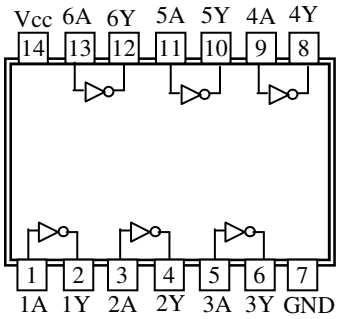
Mạch OR
- SN7432



Bảng trạng thái

A	B	Y
L	L	L
H	L	H
L	H	H
H	H	H

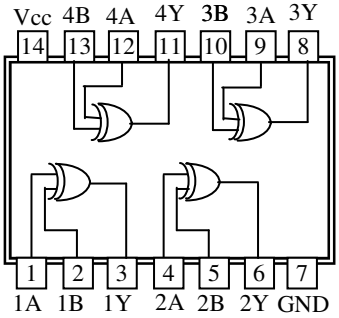
Mạch NOT
- SN7404:
- SN7405
- SN7406



Bảng trạng thái:

A	Y
L	H
H	L

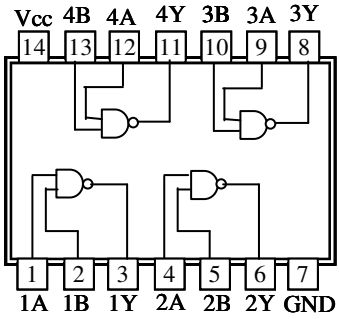
Mạch EX-OR
- SN74136



Bảng trạng thái

A	B	Y
L	L	L
H	L	H
L	H	H
H	H	L

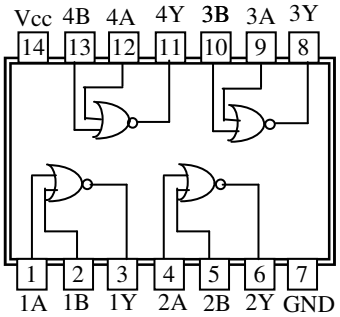
Mạch NAND
- SN7400
- SN7401
- SN7403



Bảng trạng thái:

A	B	Y
L	L	H
H	L	H
L	H	H
H	H	L

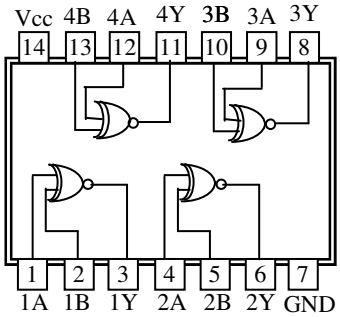
Mạch NOR
- SN7402
- SN7428



Bảng trạng thái

A	B	Y
L	L	H
H	L	L
L	H	L
H	H	L

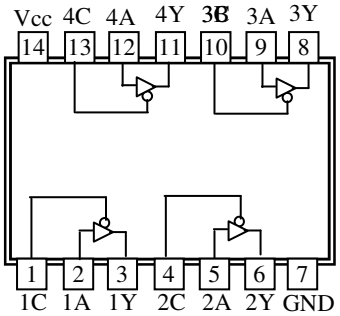
Mạch EX-NOR
- SN74HC266



Bảng trạng thái

A	B	Y
L	L	H
H	L	L
L	H	L
H	H	H

Mạch đệm 3 trạng thái kích hoạt mức thấp
- SN74125
- SN74425



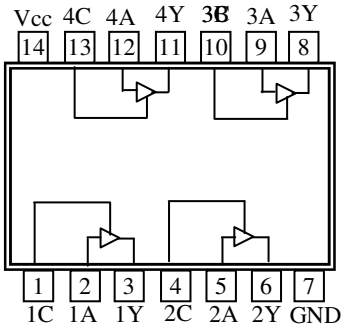
Bảng trạng thái

C	A	Y
H	X	Z_∞
L	L	L
L	H	H

Trong đó Z_∞ là trạng thái trở kháng cao.

Mạch đệm 3 trạng thái kích hoạt mức cao

- SN74126
- SN74426

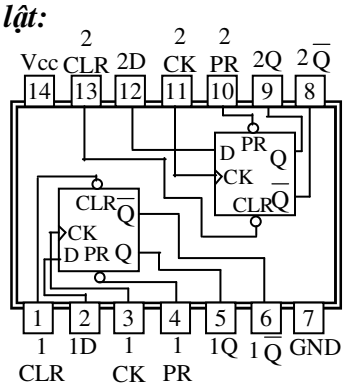


Bảng trạng thái

C	A	Y
L	X	Z_∞
H	L	L
H	H	H

1.2.2 Các mạch lật:

- SN7474



Bảng trạng thái

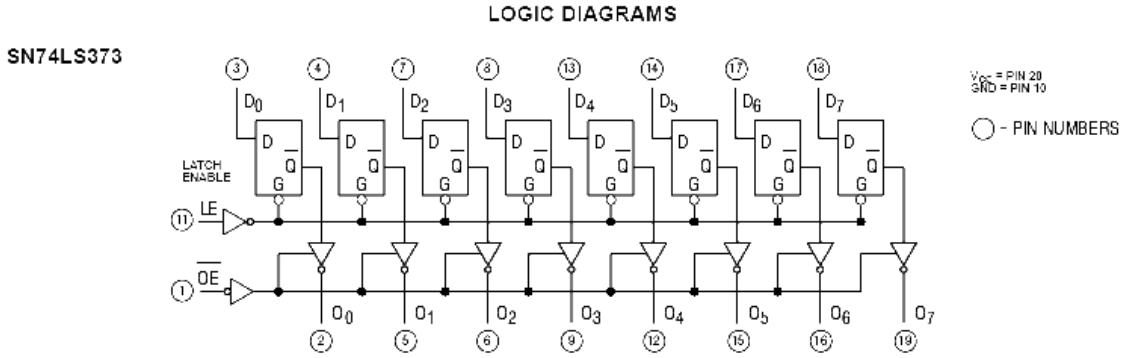
PR	CLR	CK	D	Q	$\overline{Q}$
L	H	X	X	H	L
H	L	X	X	L	H
L	L	X	X	H^*	H^*
H	H	↑	H	H	L
H	H	↑	L	L	H
H	H	L	X	Q_0	$\overline{Q}_0$

Trong đó Q_0 là trạng thái cũ trước đó, H^* là trạng thái cấm.

1.2.3 Các mạch thanh ghi đệm

- SN74LS373, SN74LS374 là loại vi mạch chốt truyền qua 8 bits cùng đệm 3 trạng thái. Đệm và chốt được điều khiển độc lập bằng các đầu vào điều khiển khác nhau là:

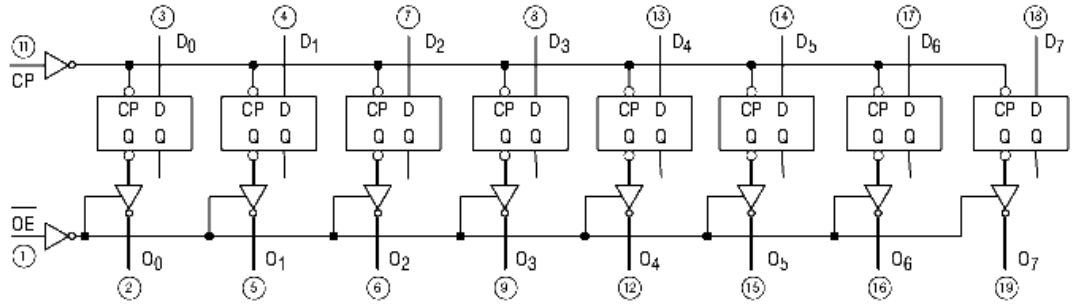
- $\overline{OE}$ (Output Enable): Mở cổng-đệm
- LE (Latch Enable): điều khiển mở chốt
- CP (Clock Input) : Đầu vào xung sườn lên



Bảng sự thật của 74LS373

MODE Hoạt động	Đầu vào			Thanh ghi nội	Đầu ra
	$\overline{OE}$	LE	D_i		O_i
Kích hoạt và đọc thanh ghi	L L	H H	L H	L H	L H
Chốt và đọc thanh ghi	L	L	x	Q_0	Q_0
Chốt và không cho ra	H	x	x	x	Z_∞

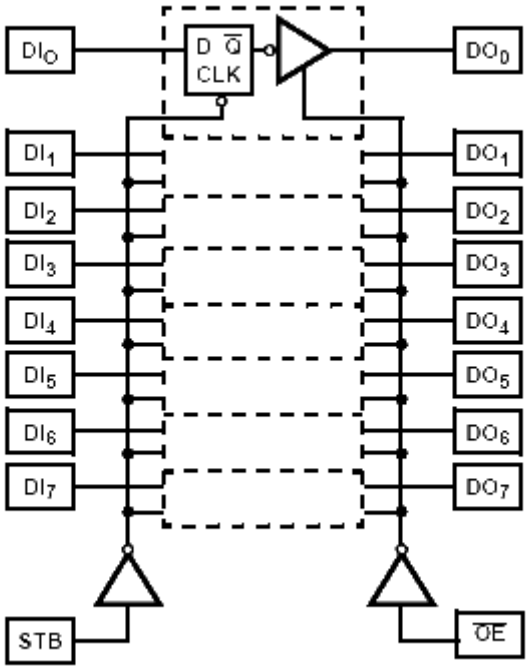
SN74LS374



Bảng sự thật của 74LS373

MODE Hoạt động	Đầu vào			Thanh ghi nội	Đầu ra
	$\overline{OE}$	CP	D_i		O_i
Kích hoạt và đọc thanh ghi	L L	↑ ↑	L H	L H	L H
Chốt và đọc thanh ghi	L	L	x	Q_0	Q_0
Chốt và không cho ra	H	x	x	x	Z_∞

Bộ điều khiển bus chốt hệ 8 loại CMOS 82C82



Bảng sự thật của 82C82

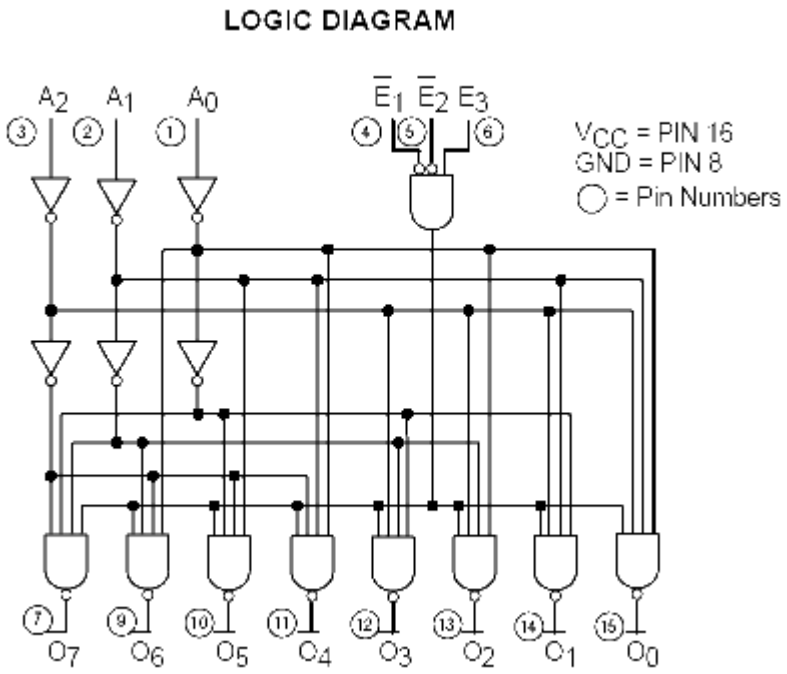
STB	$\overline{OE}$	DI	DO
X	H	X	Z_∞
H	L	L	L
H	L	H	H
↓	L	X	Q_0

Trong đó: H là logic 1 L là logic 0
X là bất kỳ Z_∞ là trạng thái trở kháng cao
↓ là sườn xung âm Q_0 là chốt giá trị của dữ liệu sau cùng

1.2.4 Các mạch giải mã địa chỉ

Một số vi mạch được dùng nhiều trong các hệ thống giải mã của VXL là SN74LS138 (mạch giải mã 3-8), SN74LS139 (hai giải mã 2-4).

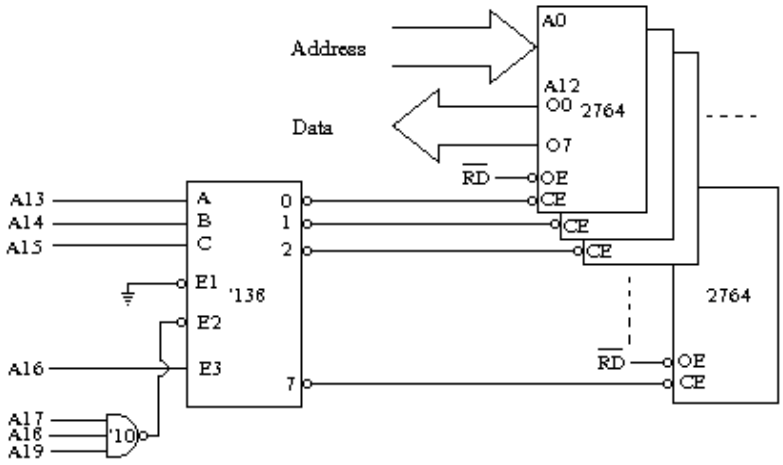
Để minh hoạ cho phương pháp này ta chọn vi mạch 74LS138. Bảng sự thật của vi mạch cho biết tại một thời điểm chỉ có một trong tám đầu ra có mức logic 0. Điều kiện cần để có một trong các mức thấp ở đầu ra là các đầu vào $\overline{E_1}$, $\overline{E_2}$, E_3 phải được kích hoạt, tức là $\overline{E_1}$, $\overline{E_2}$ ở mức thấp và E_3 ở mức cao. Khi 74LS138 đã được kích hoạt, các đầu vào địa chỉ A_0, A_1 và A_2 sẽ lựa chọn đầu ra, điều này cho phép có thể chọn được một trong tám thiết bị nhớ khác nhau tại một thời điểm.



Bảng sự thật của 74LS138

Đầu vào						Output							
Enable			Select										
$\overline{E_1}$	$\overline{E_2}$	E_3	A_2	A_1	A_0	0	1	2	3	4	5	6	7
H	x	x	x	x	x	H	H	H	H	H	H	H	H
x	H	x	x	x	x	H	H	H	H	H	H	H	H
x	x	L	x	x	x	H	H	H	H	H	H	H	H
L	L	H	L	L	L	L	H	H	H	H	H	H	H
L	L	H	L	L	H	H	L	H	H	H	H	H	H
L	L	H	L	H	L	H	H	L	H	H	H	H	H
L	L	H	L	H	H	H	H	H	L	H	H	H	H
L	L	H	H	L	L	H	H	H	H	L	H	H	H
L	L	H	H	L	H	H	H	H	H	H	L	H	H
L	L	H	H	H	L	H	H	H	H	H	H	L	H
L	L	H	H	H	H	H	H	H	H	H	H	H	L

Ví dụ về mạch giải mã địa chỉ cho ROM có thể tham khảo trên hình 1.4.

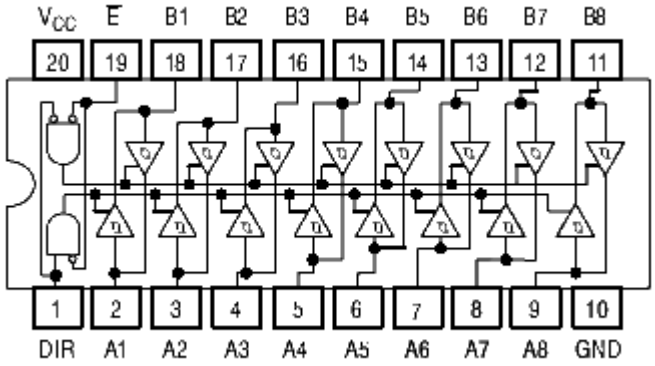


Hình 1.4: Mạch giải mã chọn ROM dùng 74LS138

1.2.5 Các mạch truyền dữ liệu

- SN74LS245

- Các chân A1÷A8, B1÷B8 là các chân gửi/ nhận dữ liệu 2 hướng tương ứng.
- Chân 19 ($\overline{G}$) là chân chọn IC (Chip Enable).
- Chân 1 (DIR) là chân chọn hướng gửi/ nhận dữ liệu (Direction).



Bảng sự thật của 74LS245

G	DIR	Hoạt động
L	L	Dữ liệu vào B, ra A
L	H	Dữ liệu vào A, ra B
H	X	Z_{∞}

CHƯƠNG 2: GHÉP NỐI VÀO RA SONG SONG ĐIỀU KHIỂN BẰNG CHƯƠNG TRÌNH

2.1. Các lệnh vào ra dữ liệu

2.1.1 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: hai loại chuyển dữ liệu 8 hay 16 bít thông qua các cổng vào/ra 8 bít, và hai chuyển dữ liệu 8 hay 16 bít thông qua các cổng 16 bít.

Ví dụ hai lệnh sau đây sẽ 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 2.1: Bảng các lệnh vào ra

Instruction	Data Width	Comment
IN AL,d8	8	Đọc một byte từ cổng vào/ra 8 bít
IN AL,DX	8	Đọc một byte từ cổng vào/ra xác định bởi thanh ghi DX
IN AX,d8	16	Đọc một word từ cổng vào/ra 8 bít
IN AX,DX	16	Đọc một word từ cổng vào/ra xác định bởi thanh ghi DX
OUT d8,AL	8	Gửi một byte ra cổng vào/ra 8 bít
OUT DX,AL	8	Gửi một byte ra cổng vào/ra xác định bởi thanh ghi DX
OUT d8,AX	16	Gửi một word ra cổng vào/ra 8 bít
OUT DX,AX	16	Gửi một word ra cổng vào/ra xác định bởi thanh ghi DX

Chú ý: d8 chứa địa chỉ là cổng vào/ra 8 bít, và DX chứa địa chỉ là cổng vào/ra 16 bít.

2.1.2.Các lệnh vào ra bằng Turbo C

Bảng 2.2: Bảng các lệnh vào ra

Instruction	Data Width	Comment
outportb(addr_port,data_var)	8	Gửi một byte ra cổng vào/ra 8 bít xác định bởi địa chỉ addr_port
Data_var=inportb(addr_port)	8	Đọc một byte từ cổng vào xác định bởi địa chỉ addr_port
outport(addr_port,data_var)	16	Gửi một word ra cổng vào/ra, 8 bít thấp xác định bởi địa chỉ addr_port, 8 bít cao xác định bởi địa chỉ addr_port+1
Data_var=inportb(addr_port)	16	Đọc một word từ cổng vào/ra, 8 bít thấp xác định bởi địa chỉ addr_port, 8 bít cao xác định bởi địa chỉ addr_port+1
outp(addr_port,data_var)	8	Gửi một byte ra cổng vào/ra 8 bít xác định bởi địa chỉ addr_port
Data_var=inp(addr_port)	8	Đọc một byte từ cổng vào xác định bởi địa chỉ addr_port
outpw(addr_port,data_var)	16	Gửi một byte ra cổng vào/ra, 8 bít thấp xác định bởi địa chỉ addr_port, 8 bít cao xác định bởi địa chỉ addr_port+1
Data_var=inp(addr_port)	16	Đọc một word từ cổng vào/ra, 8 bít thấp xác định bởi địa chỉ addr_port, 8 bít cao xác định bởi địa chỉ addr_port+1

Ví dụ các lệnh sau xuất 5 (8 bít 00000101) qua thanh ghi dữ liệu của LPT1

```
#include <dos.h>
#define Dat_reg  0x378
char  data_var=5;
outport(Dat_reg,data_var);
```

2.1.3 Các lệnh vào ra bằng Turbo Pascal

Bảng 2.3: Bảng các lệnh vào ra

Instruction	Data Width	Comment
Port[addr] := var1; Var := port[addr];	8 8	Gửi một byte ra cổng vào/ra 8 bít xác định bởi địa chỉ addr Đọc một byte từ cổng vào xác định bởi địa chỉ addr

Ví dụ:

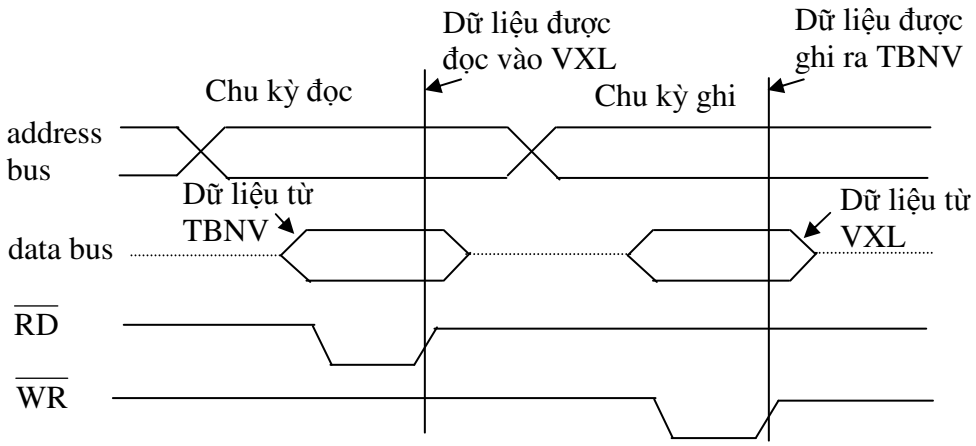
```
Uses crt;
```

Port[$378] := X; Xuất ra giá trị biến X lên đường dẫn dữ liệu (D0 đến D7) của cổng LPT1
Y := port[$379]; Đọc nội dung thanh ghi trạng thái của LPT1 vào biến Y

Chú ý: Một chữ số ở hệ 16 phải thêm dấu: 0x vào trước số đó trong ngôn ngữ C,
và $ vào trước số đó trong ngôn ngữ Pascal.

2.1.4 Giản đồ thời gian các lệnh đọc/ghi dữ liệu:

Bus hệ thống, bao gồm bus dữ liệu (data bus), bus điều khiển (control bus) và bus địa chỉ (address bus), là các đường dẫn truyền thông giữa vi xử lý và TBNV.
Các VXL 8 bit dữ liệu thường có 8 đường dây ở data bus, và 16 đường dây tương ứng 16 đường địa chỉ ở address bus. Hai tín hiệu $\overline{RD}$, $\overline{WR}$ ở control bus thường được sử dụng để xác định thời điểm dữ liệu ổn định trên data bus.
Hình 1.5 là giản đồ thời gian tín hiệu trên bus hệ thống.



Hình 1.5: Giản đồ thời gian tín hiệu trên bus hệ thống.

Mỗi chu kỳ bus (bus cycle) bao gồm việc chuyển 1 từ dữ liệu giữa VXL với bộ nhớ hoặc TBNV. Mỗi chu kỳ bus bắt đầu khi VXL xuất một địa chỉ để chọn một vị trí bộ nhớ hoặc các cổng vào ra. Trong giản đồ này, các bus địa chỉ và dữ liệu được biểu diễn bằng 1 cặp đường thẳng để chỉ thông tin trên bus ổn định. Khi các đường trên sơ đồ cắt ngang nhau diễn tả dữ liệu đã thay đổi. Đường nét đứt là trạng thái thả nổi khi không có thiết bị nào lái nó.

2.2 Ghép nối song song đơn giản: có hay không có đối thoại

2.2.1 Cửa vào đơn giản không đối thoại

Hình 2.1 mô tả một cửa vào đơn giản không cần tín hiệu đối thoại. Tín hiệu vào từ bộ cảm biến (BCB) được nối với bus dữ liệu của MVT thông qua bộ đệm ba trạng thái.

Khi chúng ta lập trình đoạn lệnh:

C: *data1 = inport(addrport);*
Pascal: *data1 := port(addrport);*

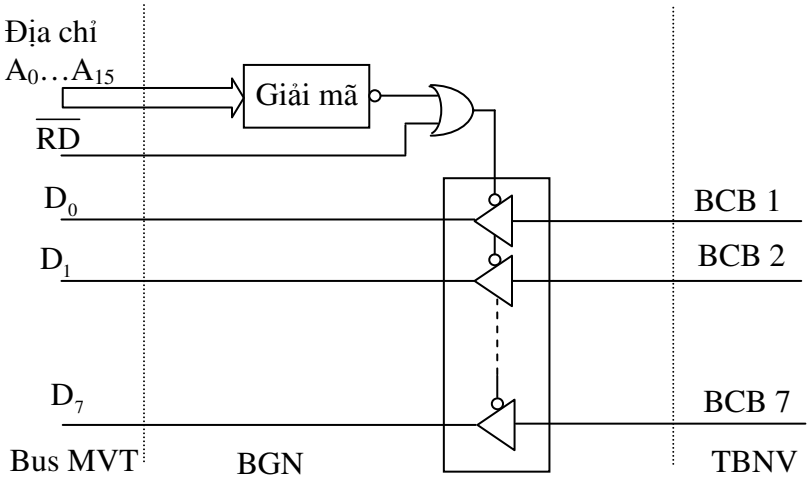
Trình biên dịch sẽ chuyển đoạn lệnh trên thành mã máy, và khi chạy đoạn mã máy này, VXL sẽ tạo ra các tín hiệu:

- $A_0 \ldots A_{15}$ từ addrport
- $\overline{RD}$ 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 sẽ kích hoạt ở mức 0. Do đó đầu ra mạch OR sẽ ở mức 0, kích hoạt cho bộ đệm ba trạng thái mở ra, và dữ liệu từ TBNV sẽ đổ vào bus dữ liệu của VXL. Lúc này, VXL sẽ nhận

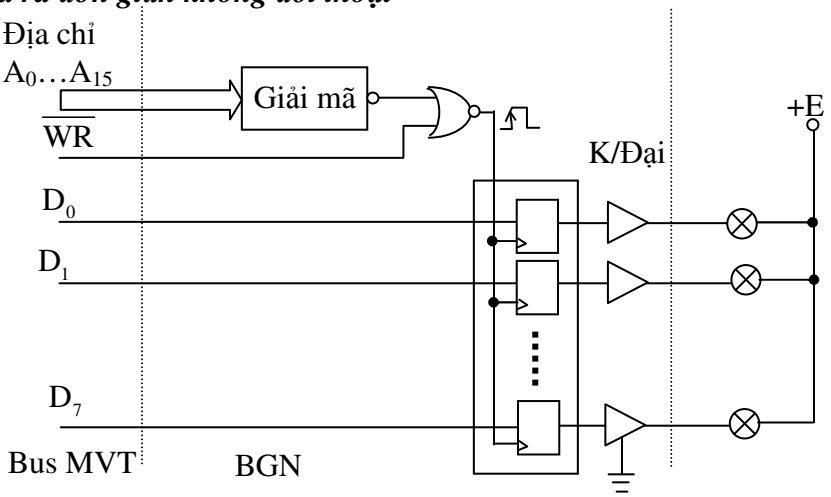
- $D_0, ..., D_7$ gán vào cho biến data1.

Kết thúc chu kỳ lệnh, $\overline{RD}$ trở về mức 1.



Hình 2.1: Cửa vào đơn giản, không có đối thoại

2.2.2 Cửa ra đơn giản không đối thoại



Hình 2.2: Cửa ra đơn giản, không có đối thoại

Tương tự, một cửa ra đơn giản không có đối thoại được thể hiện trên hình 2.2. Dữ liệu từ VXL được gởi ra TBNV qua thanh ghi đệm, dùng để điều khiển đèn. Mạch khuếch đại sẽ nâng công suất lên phù hợp với công suất của đèn.

Khi lập trình phần mềm, chúng ta sẽ gõ vào dòng lệnh

C: *outportb(addrport,data2);*
Pascal: *port(addrport) := data2;*

Trình biên dịch sẽ chuyển đoạn lệnh trên thành mã máy, và khi chạy đoạn mã máy này, VXL sẽ tạo ra các tín hiệu:

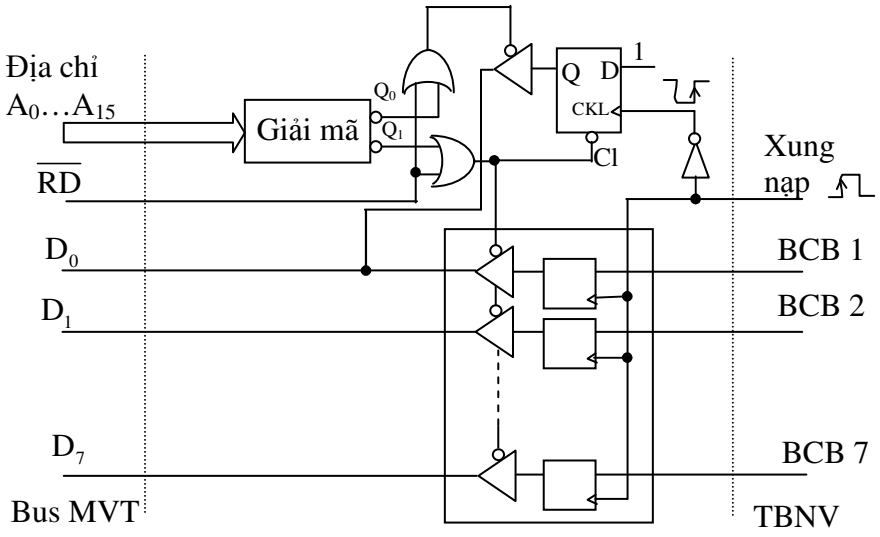
- $A_0 \ldots A_{15}$ từ addrport
- $\overline{WR}$ kích hoạt ở mức 0.

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 sẽ kích hoạt ở mức 0. Do đó đầu ra mạch NOR sẽ nhảy lên mức 1, kích hoạt cho bộ thanh ghi đệm mở ra, và TBNV nhận dữ liệu từ MVT, gồm:

- $D_0, ..., D_7$ tương ứng với biến data2.

Kết thúc chu kỳ lệnh, $\overline{WR}$ trở về mức 1.

2.2.3. Cửa vào đơn giản có đối thoại



Hình 2.3: Cửa vào có đối thoại

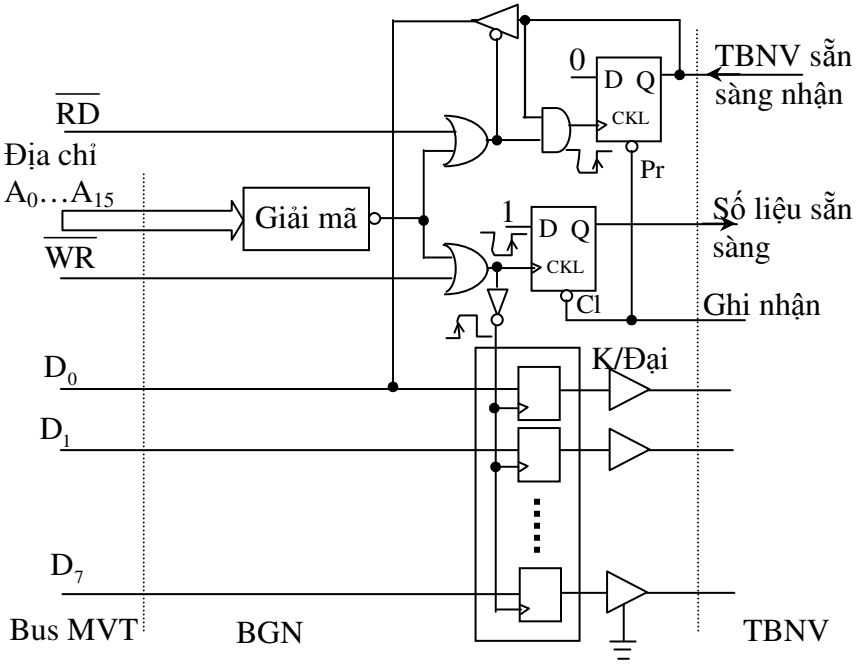
Cửa vào có đối thoại thể hiện trên hình 2.3. TBNV gởi số liệu tới đầu vào các thanh ghi. Khi số liệu đã ổn định, thiết bị ngoại vi gởi tiếp một xung nạp. Sườn lên xung nạp đưa dữ liệu của TBNV vào nhớ trong thanh ghi. Sườn xuống xung nạp kích mạch lật, đưa đầu ra Q = D = 1. Đó là trạng thái sẵn sàng của TBNV. Bộ VXL sẽ kiểm tra trạng thái này bằng thuật toán trong đoạn 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 ba trạng thái, đưa tín hiệu Q = 1 vào đường dữ liệu D_0. Qua đó, dữ liệu với D_0 = 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 ba trạng thái chuyển sang trạng thái trở kháng cao, cách ly chân Q và đường dữ liệu D_0.

Sau khi VXL kiểm tra thấy dữ liệu đã sẵn sàng (bit D_0 = 1), lệnh inportb thứ hai sẽ làm VXL tạo tín hiệu kích mở bộ đệm ba trạng thái, đưa dữ liệu của TBNV đang nhớ trong các thanh ghi vào bus dữ liệu và gán cho biến data4. Cũng bằng xung chọn mạch này, mạch lật sẽ xoá Q về mức logic 0.

2.2.4 Cửa ra đơn giản có đối thoại



Hình 2.4: Cửa ra có đối thoại

Hình 2.4 mô tả cửa ra có đối thoại. Khi TBNV sẵn sàng nhận dữ liệu, TBNV sẽ đưa ra một tín hiệu logic 1. Bộ VXL sẽ kiểm tra trạng thái của TBNV trong đoạn lệnh:

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

Cách hoạt động của VXL tương tự trên, lệnh inportb sẽ nạp dữ liệu có $D_0 = 1$ vào biến data5, biến này dùng để so sánh nhận biết TBNV sẵn sàng chưa. Kết thúc lệnh inportb, đầu ra mạch OR chuyển sang 1, tạo xung lên ở đầu ra mạch AND, làm chuyển trạng thái mạch lật Q = D = 0, đồng thời làm mạch đệm ba trạng thái chuyển sang trạng thái trở kháng cao, cách ly chân Q và đường dữ liệu D_0.

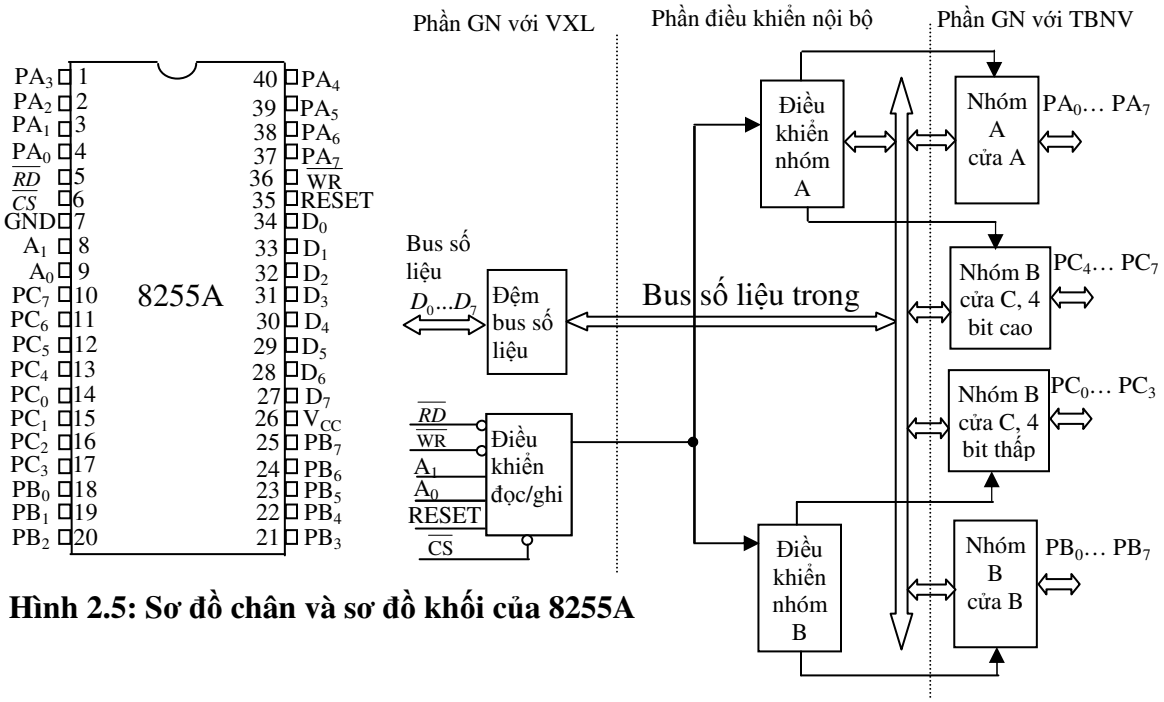
Nếu thấy TBNV sẵn sàng nhận thì lệnh outportb tiếp theo sẽ gởi địa chỉ adrport cùng tín hiệu $\overline{WR} = 0$ tạo ra một xung ra mạch OR. Sườn lên xung này nạp dữ liệu data6 trên bus dữ liệu vào thanh ghi để xuất ra TBNV, sườn xuống đưa tín hiệu Q = D = 1, báo cho TBNV là số liệu sẵn sàng. Sau khi nhận số liệu, TBNV sinh ra tín hiệu ghi nhận để xoá tín hiệu số liệu sẵn sàng (về mức 0) và lập tín hiệu TBNV sẵn sàng nhận (lên mức 1), chờ bộ VXL gởi số liệu tiếp theo.

Trao đổi dữ liệu có đối thoại làm việc chắc chắn hơn trao đổi không đối thoại. Vì TBNV có tốc độ hoạt động chậm, nên số liệu đầu vào VXL có lúc không đúng khi TBNV

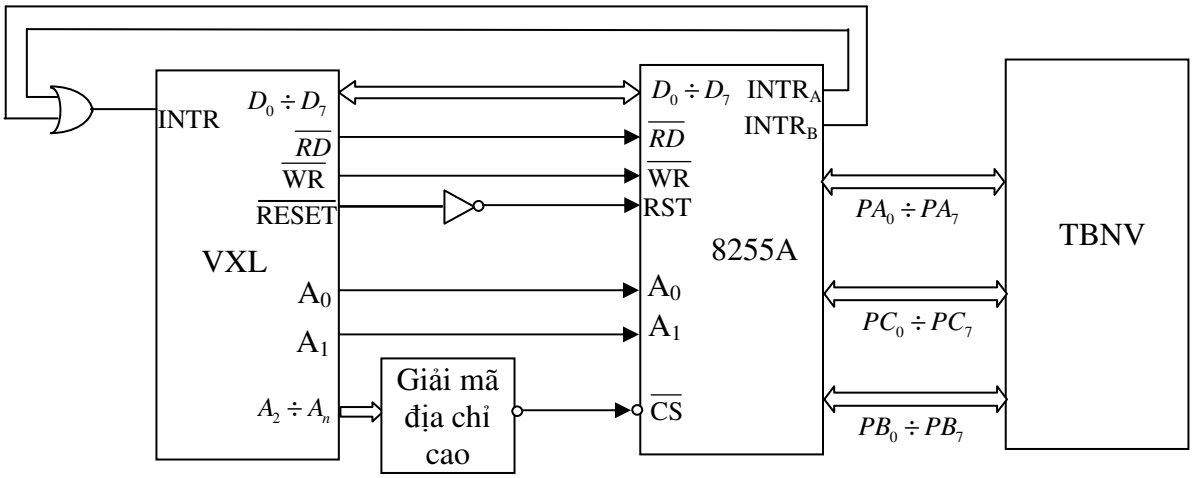
có số liệu chưa ổn định. Do đó một thuật toán cần dùng cho trao đổi không đối thoại là đọc nhiều lần và kiểm tra kết quả các lần với nhau. Nếu kết quả giống nhau thì dữ liệu vào xem như ổn định.

2.3 Ghép nối song song điều khiển bằng chương trình

2.3.1 Sơ đồ khối và chức năng các khối của 8255A



Hình 2.5: Sơ đồ chân và sơ đồ khối của 8255A



Hình 2.6: Sơ đồ GN 8255A với VXL và TBNV

Vi mạch PPI 8255A là vi mạch điều khiển bằng chương trình được áp dụng khá rộng rãi. Hình 1.10 trình bày sơ đồ ghép nối 8255A, gồm có:

1. *Phần ghép nối với VXL:*

- Bộ đệm số liệu để trao đổi dữ liệu hai chiều giữa MVT và đường dây số liệu trong.
- Bộ logic điều khiển đọc viết: tức bộ giải mã địa chỉ lệnh cho các thanh ghi đệm và thanh ghi điều khiển.

Với tổ hợp các tín hiệu địa chỉ (A_0, A_1). chọn vi mạch ($\overline{CS}$), các lệnh đọc ($\overline{RD}$) và ghi ($\overline{WR}$) của VXL, ta có các lệnh ghi và đọc khác nhau cho các cửa (A, B, C) và từ điều khiển (control word) như bảng 2.4.

Bảng 2.4: Bảng trạng thái của 8255A

A_1	A_0	$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	Lệnh (của VXL)	Chiều di chuyển số liệu (với VXL)
0 0 1	0 1 0	0 0 0	0 0 0	1 1 1	Đọc cửa A Đọc cửa B Đọc cửa C	Cửa A → D_0 … D_7 Cửa B → D_0 … D_7 Cửa C → D_0 … D_7
1	1	0	0	1		Không có giá trị
0 0 1 1	0 1 0 1	0 0 0 0	1 1 1 1	0 0 0 0	Ghi cửa A Ghi cửa B Ghi cửa C Ghi thanh ghi điều khiển	D_0 … D_7 → Cửa A D_0 … D_7 → Cửa B D_0 … D_7 → Cửa C D_0 … D_7 → thanh ghi điều khiển
x	x	1	x	x	Vi mạch ở trạng thái trở kháng cao	Không có trao đổi số liệu

2. *Phần ghép nối với TBNV:*

- Cửa A, B: hai thanh ghi đệm số liệu (8 bit) vào hoặc ra tuỳ chương trình
- Cửa C có thể chia là hai nhóm độc lập bằng chương trình: nữa cao (PC_4 … PC_7) và nữa thấp (PC_0 … PC_3). Tuỳ theo chế độ sử dụng trong từ điều khiển, cửa C có thể dùng:
  - Trao đổi số liệu vào hoặc ra ở chế độ 0 (mode 0).
  - Điều khiển hoặc đối thoại với TBNV và VXL khi cửa A và B ở chế độ 1 bằng cách xác lập và xoá từng bit PC_i .
  - Điều khiển hoặc đối thoại với TBNV và VXL khi cửa A và B ở chế độ 1 và 2. Ở chế độ 1 và 2, đọc các bit cửa C, ta biết được trạng thái của các cửa A và B.

3. *Phần các mạch điều khiển nội bộ:*

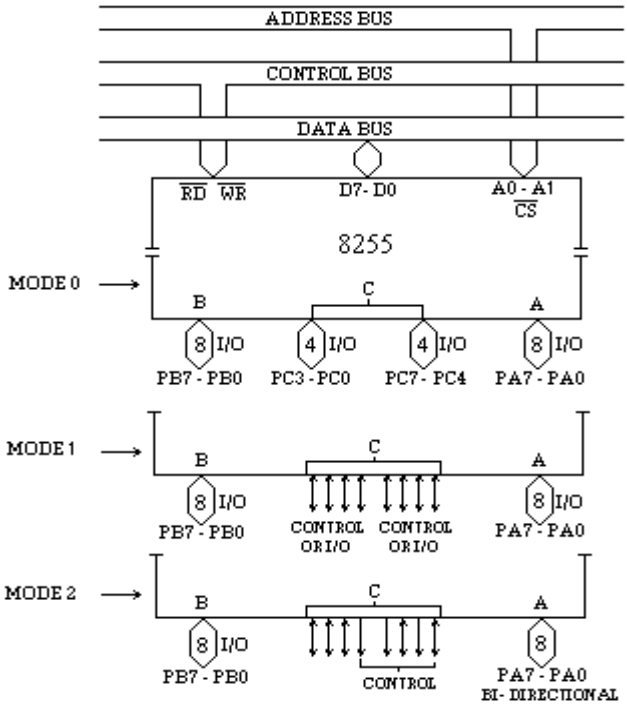
Có các khối điều khiển (nhóm A, nhóm B) các cửa A, B và C.

2.3.2 Các chế độ làm việc của 8255A:

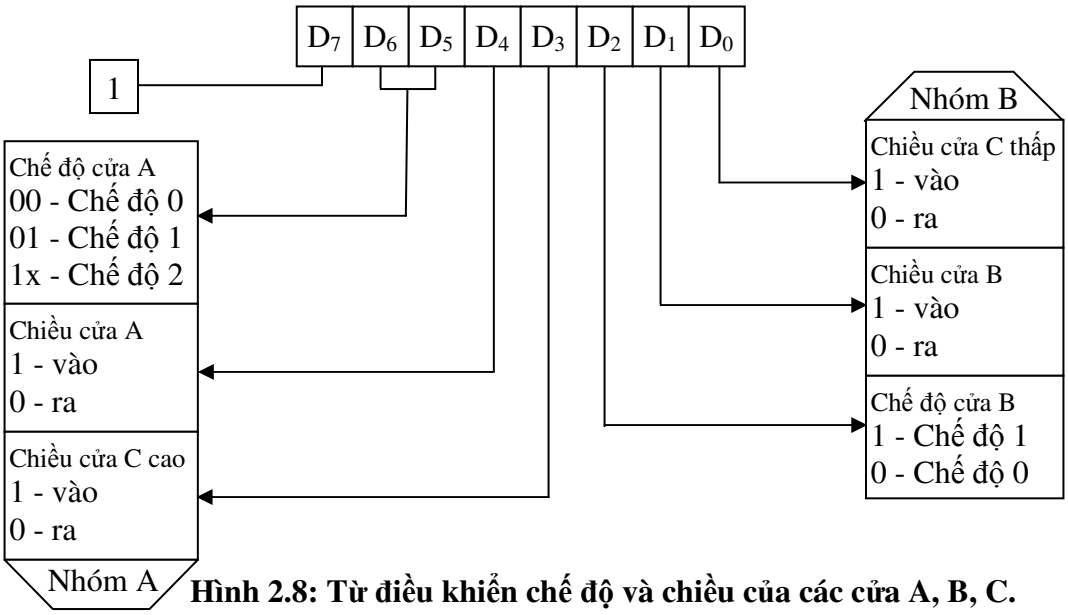
Bộ ghép nối 8255A có ba chế độ làm việc cơ bản như hình 2.7:

- Chế độ 0: vào/ra thông thường
- Chế độ 1: chốt vào/ra
- Chế độ 2: BUS hai chiều

Tuỳ theo từ điều khiển ghi vào thanh ghi điều khiển khi khởi động cho vi mạch, ta có các chế độ làm việc và chiều trao đổi số liệu của các cửa A, B, C khác nhau, như hình 2.8.



Hình 2.7: Tóm tắt các chế độ của 8255A



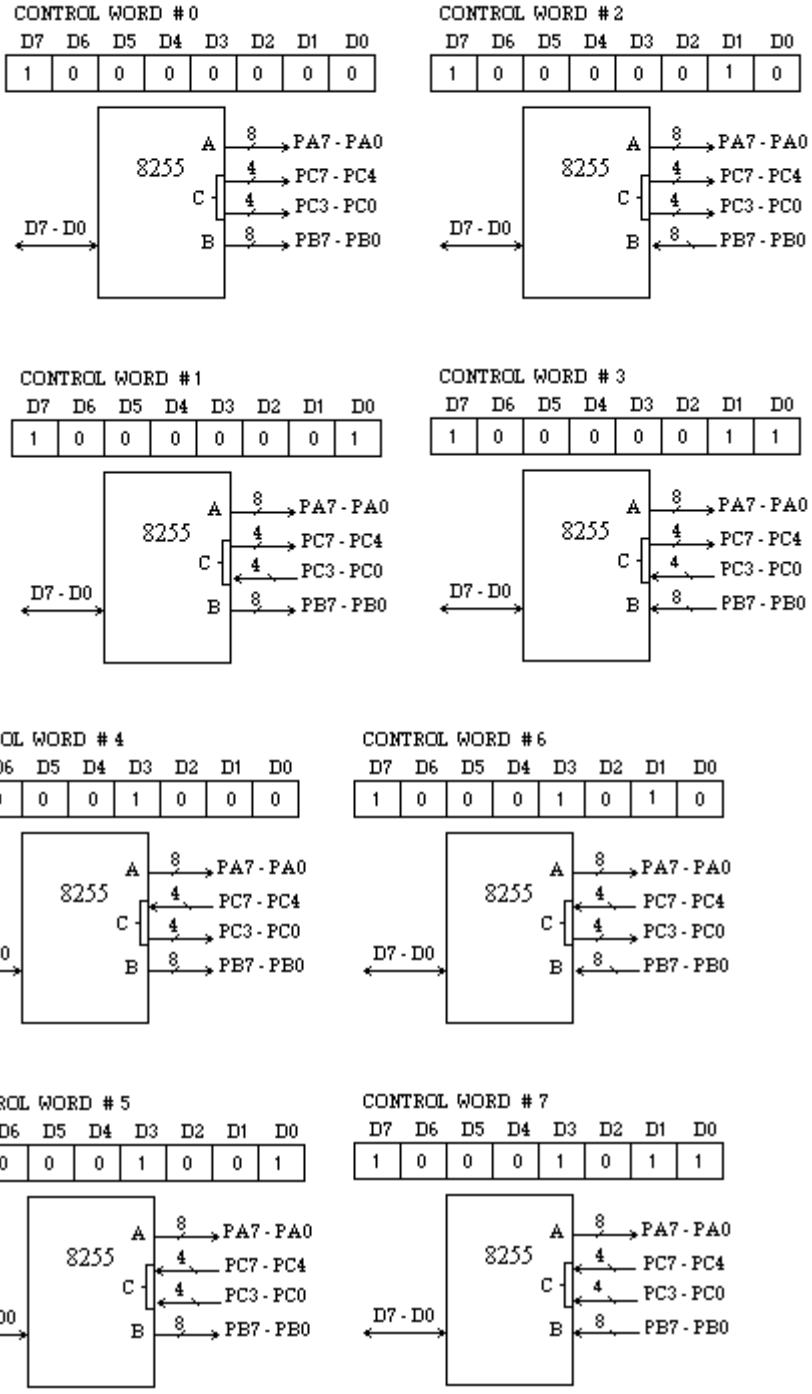
Hình 2.8: Từ điều khiển chế độ và chiều của các cửa A, B, C.

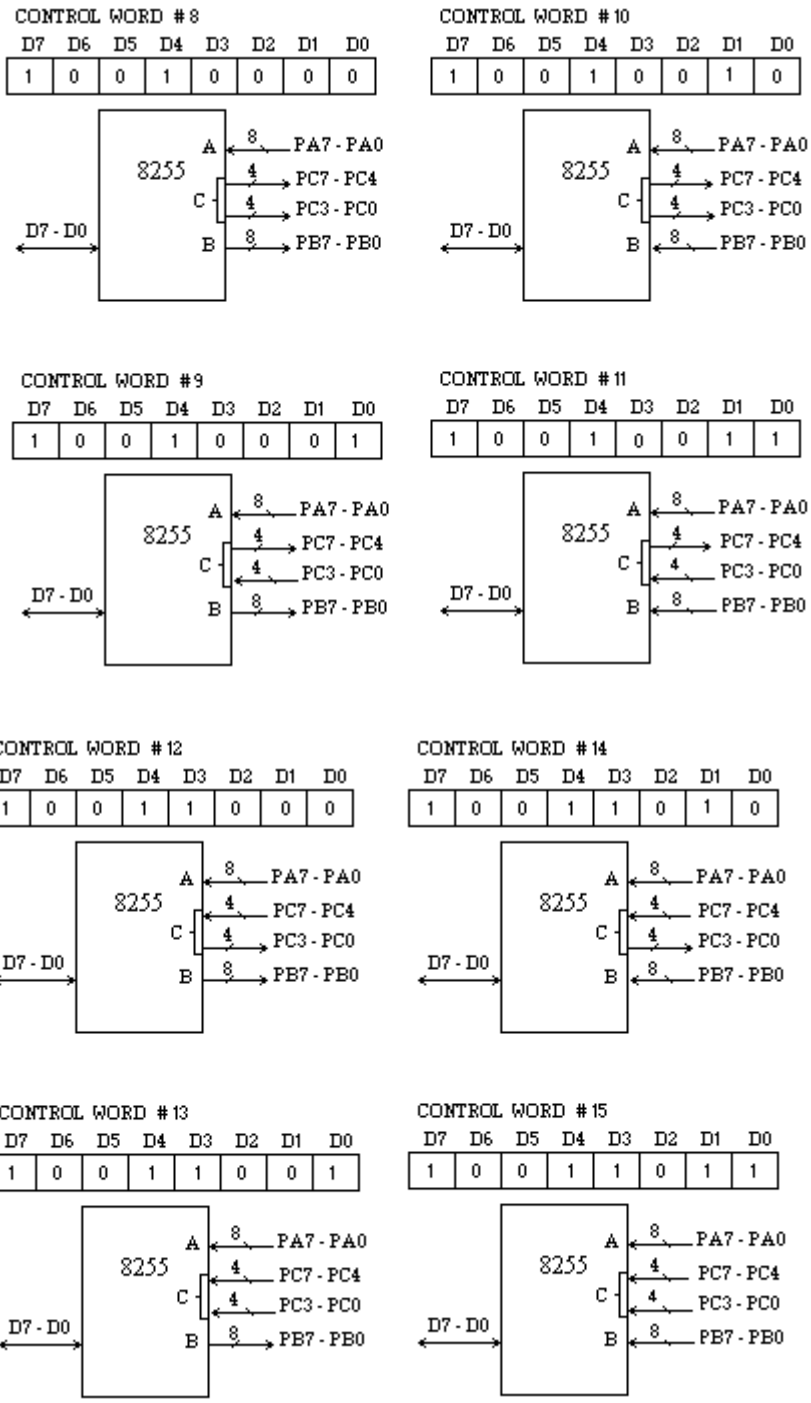
1. Chế độ 0:

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 .

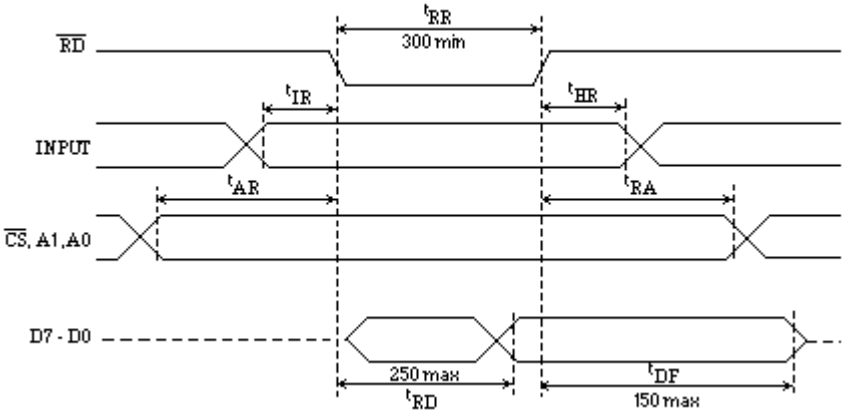
Trong chế độ này, vi mạch có các chức năng sau :

- 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 nhờ sử dụng từ điều khiển và lệnh OUT như sau :

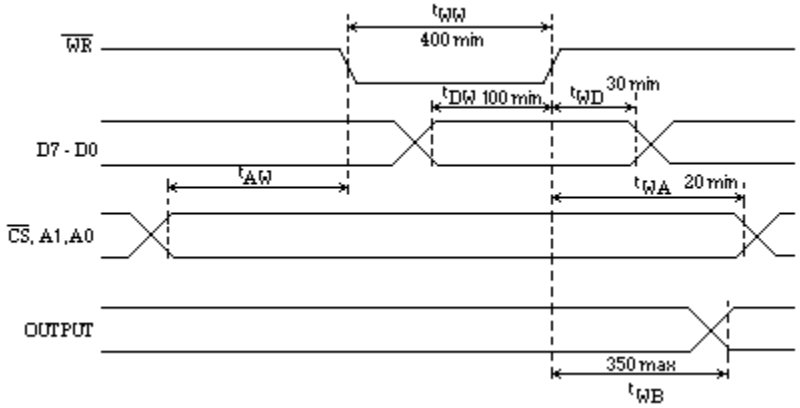




* Đồ thị thời gian đọc /ghi của 8255 trong chế độ này như sau :



Đồ thị thời gian quá trình đọc.



Đồ thị thời gian quá trình ghi.

2. Chế độ 1:

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 (hanshaking 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 chế độ 1 là

- 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.

❖ <u>Các tín hiệu điều khiển vào được dùng trong chế độ vào :</u>

- STB (Strobe input)

Mức thấp của tín hiệu vào này cho phép dữ liệu được đọc vào .

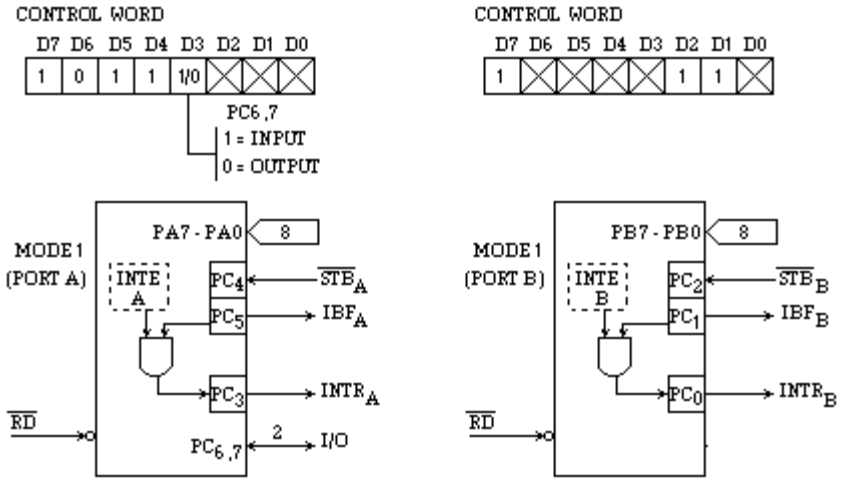
- IBF (Input Buffer Full)

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 $\overline{RD}$.

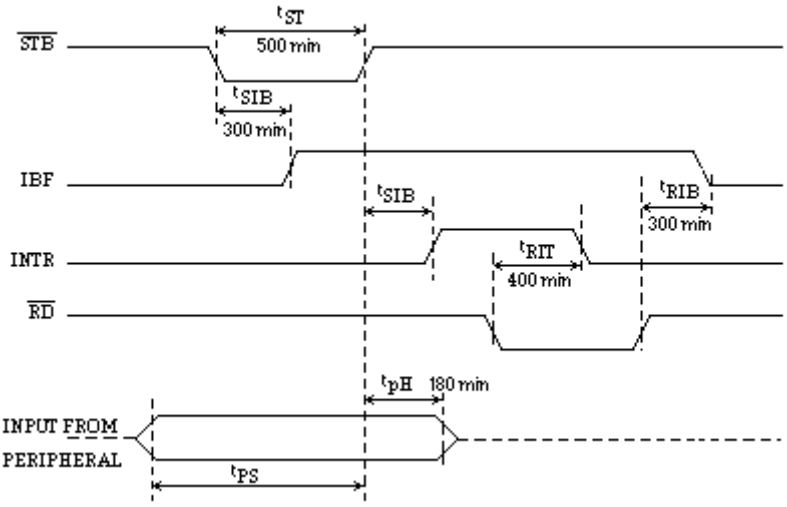
- INTR (Interrupt Request)

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 sau của tín hiệu $\overline{RD}$. 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 và đồ thị thời gian của chế độ này như sau:



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



Đồ thị thời gian 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 $\overline{WR}$ và bị xoá khi tín hiệu vào ACK ở mức thấp .

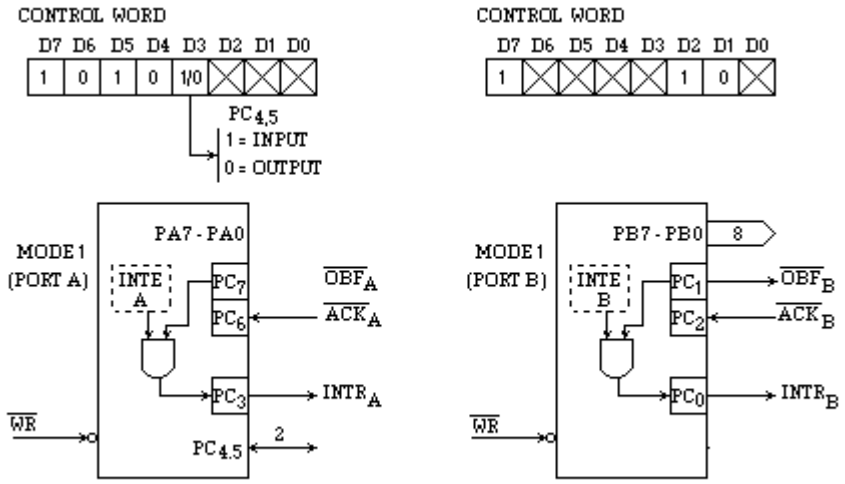
- ACK (Acknowledge Input).

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 phúc đáp từ thiết bị ngoại vi thông báo nó đã nhận được dữ liệu gửi tới từ CPU.

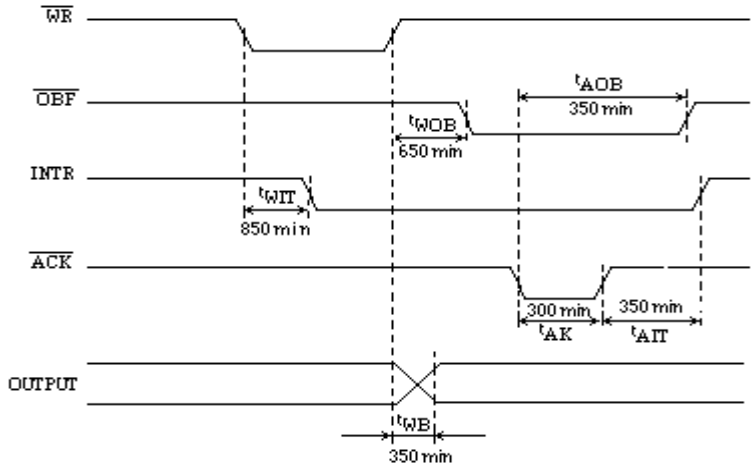
- INTR (Interrupt Request)

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 xuống của tín hiệu $\overline{WR}$.

Các từ lệnh của chế độ này như sau:

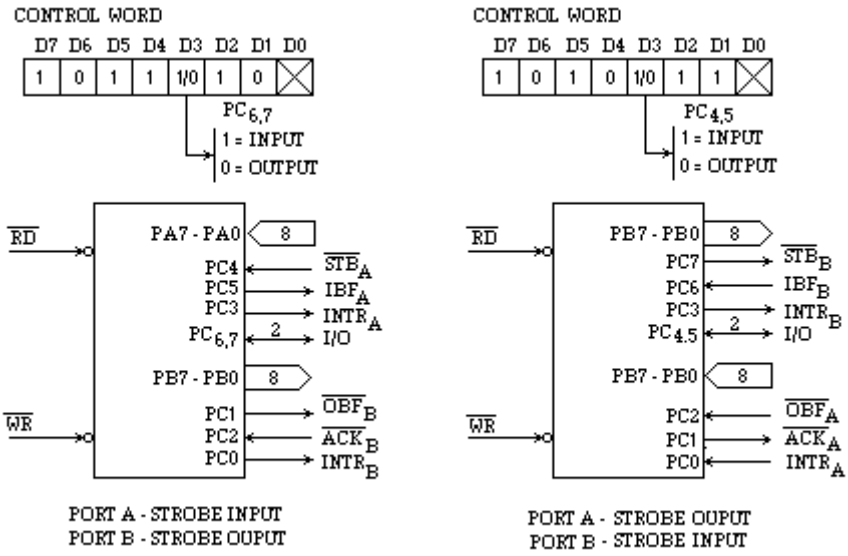


* Đồ thị thời gian của chế độ hoạt động này như sau :



Đồ thị thời gian quá trình ghi.

Trong chế độ 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:



3. Chế độ 2

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 chế độ 1.

Các chức năng cơ bản của chế độ 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 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 .

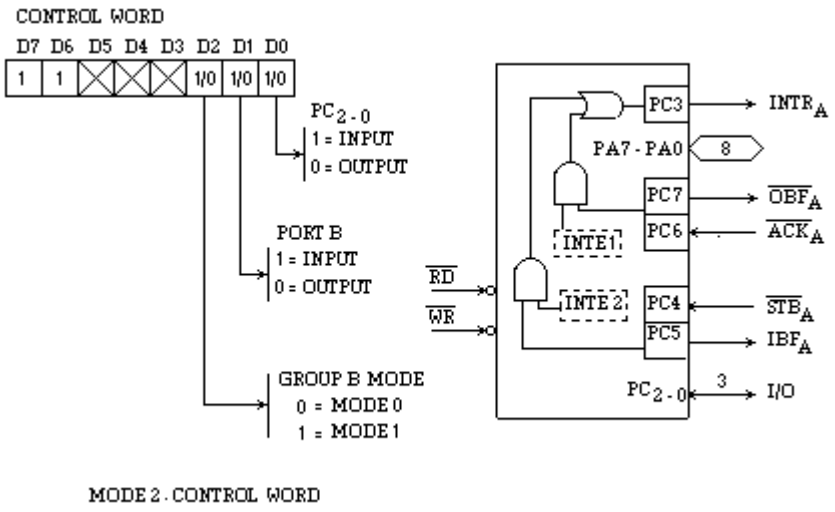
- IBF (Input bufer Full).

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 :



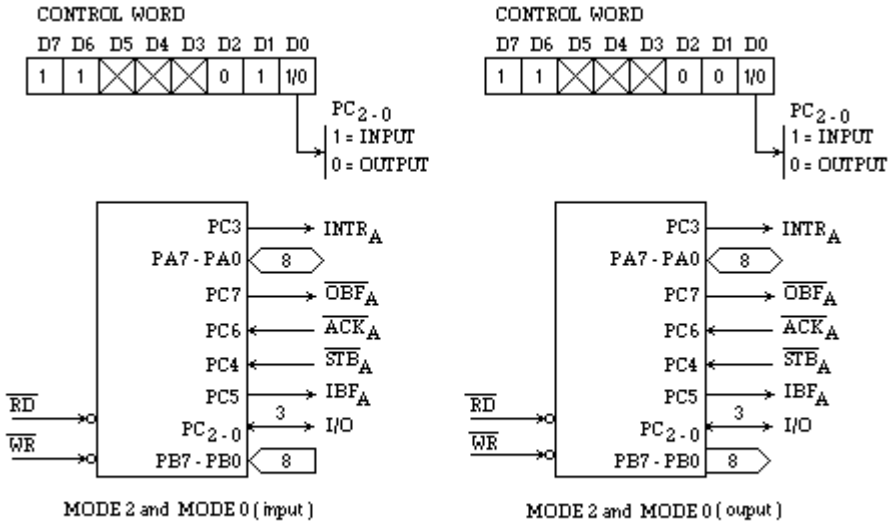
MODE 2 - CONTROL WORD

4. Chế độ kết hợp

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 :

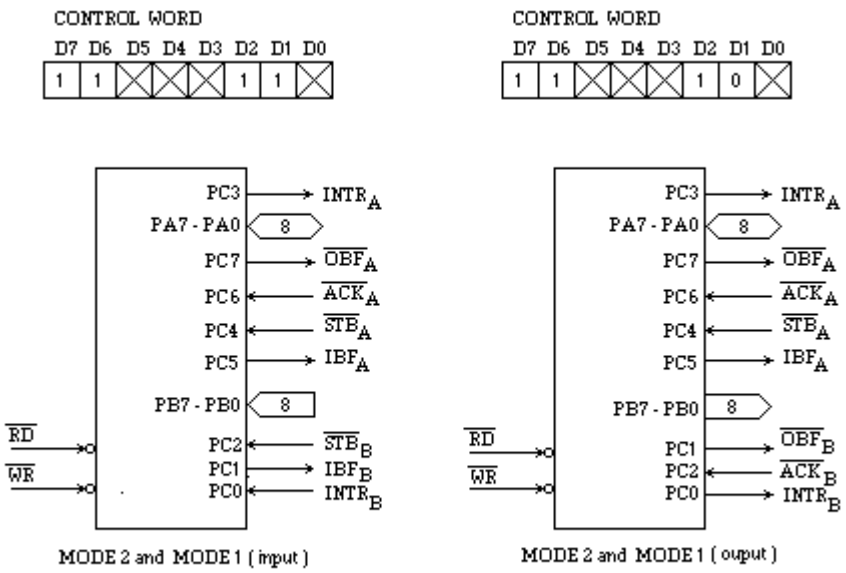
❖ <u>Khi được lập trình là các đường vào tín hiệu</u>

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 hoạ sau :



MODE 2 and MODE 0 (input)

MODE 2 and MODE 0 (ouput)



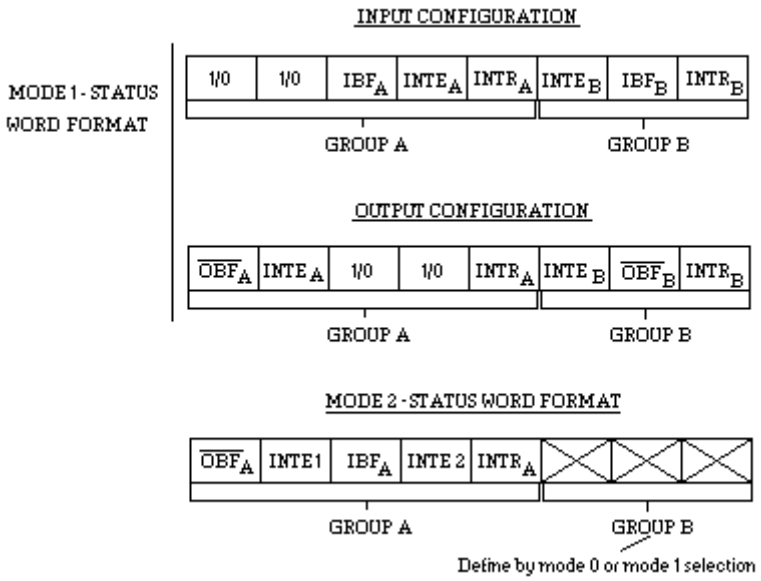
❖ Khi được lập trình là các đường tín hiệu ra

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.

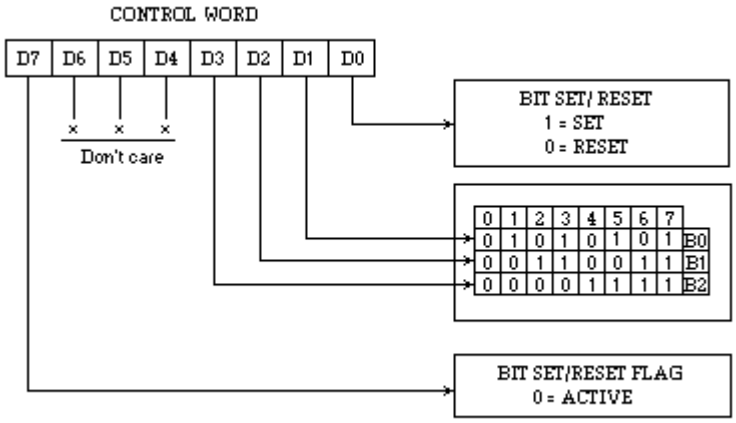
Trong mode 0, cổng C truyền dữ liệu tới hoặc từ thiết bị ngoại vi. Khi 8255 được lập trình hoạt động trong mode 1 hoặc mode 2, cổng C được sử dụng để tạo ra hoặc nhận tín hiệu hội thoại trao đổi với thiết bị ngoại vi. Đọc nội dung của cổng C cho phép người lập trình kiểm tra trạng thái của các thiết 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.



5. 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 lệnh hoạt động trong chế độ như sau:



6. Chức năng điều khiển ngắt (Interrupt control)

Khi 8255 được lập trình hoạt động ở chế độ 1 hoặc chế độ 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 .

2.3.3 Ghép nối với 8255A

Hình 2.9 trình bày một ví dụ một mạch GN 8255A với máy in, máy đọc băng ở chế độ 0. Nửa thấp cửa C là cửa vào, dùng để đọc các trạng thái

- PC_1 cho trạng thái bận của máy in.
- PC_0 cho trạng thái sẵn sàng của máy đọc băng.

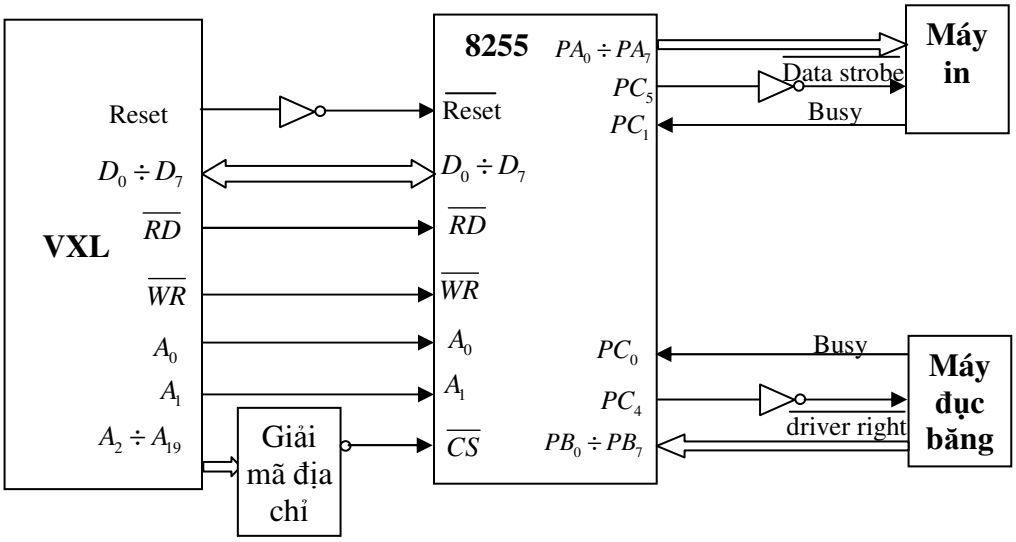
Nửa cao cửa C là cửa ra, dùng để đưa ra tín hiệu điều khiển:

- PC_5 đưa tín hiệu chốt số liệu cho máy in (qua bộ đảo).
- PC_4 để dịch chuyển băng giấy đi 1 vị trí về bên phải (qua bộ khuếch đại đảo).

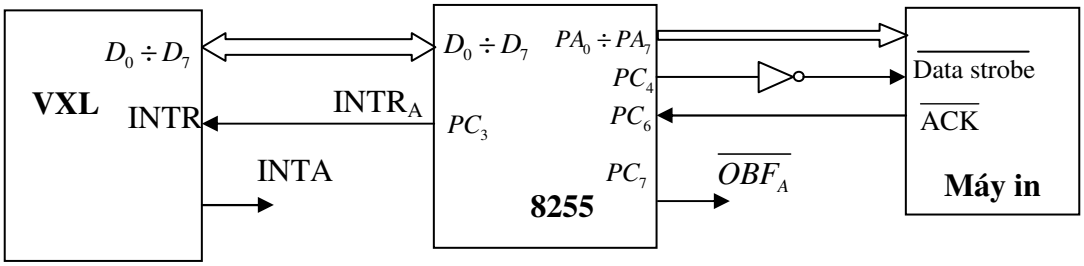
Hình 2.10 mô tả mạch ghép nối với máy in ở chế độ 1. Các tín hiệu đối thoại gồm:

- PC_4 tín hiệu ra để chốt số liệu, có thể dùng PC_7 ($\overline{OBF_A}$) thay thế.
- PC_6 tín hiệu vào cho $\overline{ACK}$, xác nhận đã in số liệu.
- PC_3 dùng để yêu cầu ngắt chương trình, đưa vào chân INTR của VXL.

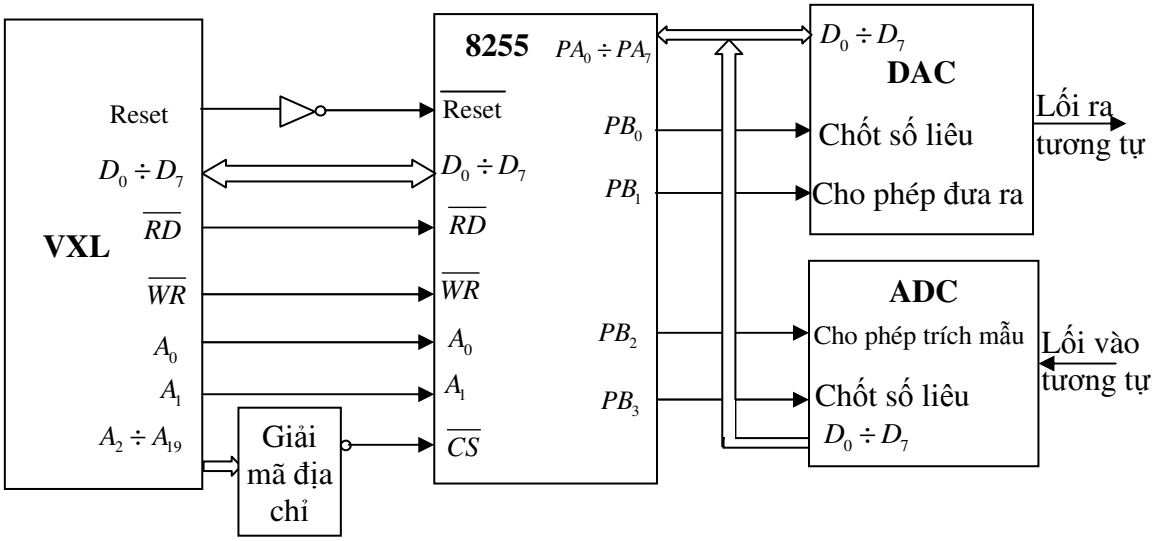
Nhờ sử dụng chế độ 1, các tín hiệu PC_6, PC_7 sẽ tự động thu phát mà không cần lệnh của chương trình, trừ PC_4 phải đưa lệnh. Với việc nối vào INTR của VXL, chương trình phục vụ ngắt không cần lệnh kiểm tra trạng thái của PC_6.



Hình 2.9: Ví dụ GN 8255 ở chế độ 0.



Hình 2.10: Ví dụ GN 8255 ở chế độ 1.



Hình 2.11: Ví dụ GN 8255 ở chế độ kết hợp.

Hình 2.11 mô tả ví dụ ghép nối 8255A ở chế độ 2 của cửa PA và chế độ 0 của cửa PB.

.

2.3.4 Lập trình cho 8255A

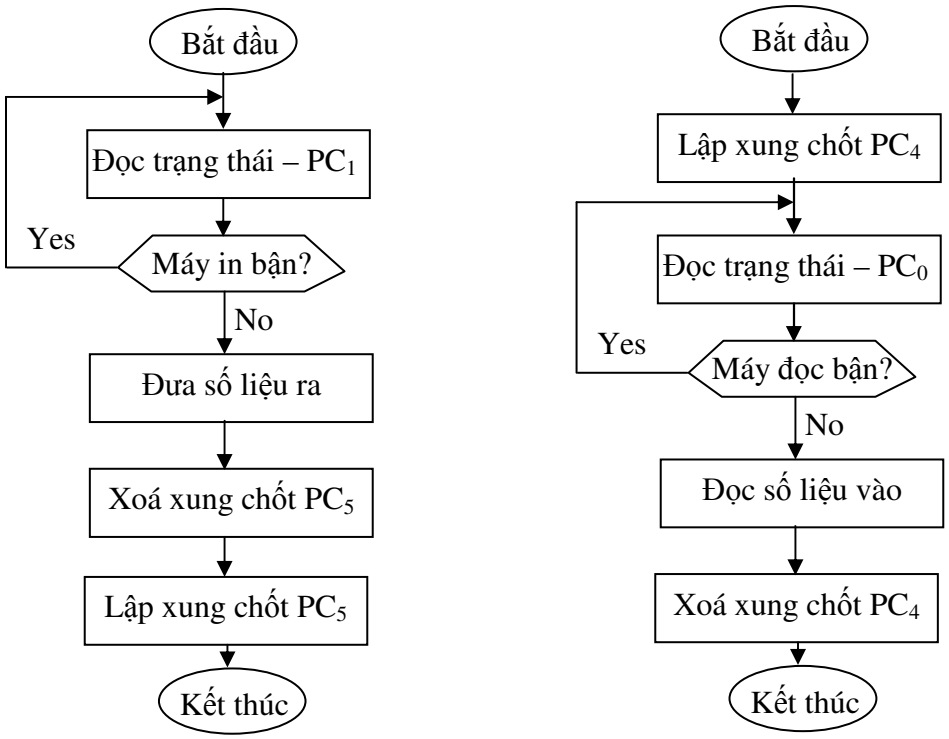
Vi mạch cổng song song lập trình được PPI 8255 có khả năng hoạt động trong các chế độ hoạt động khác nhau các chế độ hoạt động này có thể kết hợp được với nhau tạo khả năng lập trình điều khiển hoạt động một cách linh hoạt.

Khi viết chương trình cho 8255A hoạt động, ta phải :

- Viết lưu đồ thuật toán.
- Xác định từ điều khiển chế độ cho thanh ghi điều khiển, từ trạng thái cho từng mạch cụ thể.
- Chương trình khởi phát bằng lệnh ghi vào thanh ghi điều khiển.
- Đưa nội dung các bít cho các cửa dùng để điều khiển TBNV.
- Đọc và kiểm tra trạng thái : là thanh ghi trạng thái nếu 8255 dùng ở chế độ 1,2; là một cửa ghi trạng thái TBNV nếu ở chế độ 0.
- Đưa số liệu vào hay ghi số liệu ra các cửa.

Ví dụ viết chương trình điều khiển đưa số liệu ra máy in như trên hình 2.9, và địa chỉ cao dành cho máy in xác định là 3FE18h.

- Lưu đồ thuật toán thể hiện trên hình 2.12.



Hình 2.12: Lưu đồ trao đổi dữ liệu giữa MVT và máy in, máy đọc băng.

- Xác định các thanh ghi chế độ và chiều của cửa:

3FE18h $\Rightarrow A_{19} \ldots A_3\ A_2\ A_1\ A_0$
Từ bảng 1.17, ta có địa chỉ sau:
PA có địa chỉ addr_A := 3FE18h
PB có địa chỉ addr_B := 3FE19h
PC có địa chỉ addr_C := 3FE1Ah
Thanh ghi điều khiển addr_dk := 3FE1Bh

- Xác định từ điều khiển và từ trạng thái:

Không dùng ở chế độ lập cờ nên D_7=1
PA ở chế độ 0, cửa ra nên D_6=0 D_5=0 D_4=0
Nửa cửa C cao là cửa ra nên D_3=0
PB ở chế độ 0, cửa vào nên D_2=0 D_1=0
Nửa cửa C thấp là của vào nên D_0=1
Ta có từ điều khiển sau:
Data_dk := 81h (=1000 0001 B)
Từ trạng thái cho các cổng PC:

	PC_7	PC_6	PC_5	PC_4	PC_3	PC_2	PC_1	PC_0	
Cho PC5	0	0	1	0	0	0	0	0	$\Rightarrow$ data_C5 := 20h

Cho PC4	0	0	0	1	0	0	0	0	$\Rightarrow$ data_C4 := 10h
Cho PC_1	0	0	0	0	0	0	1	0	$\Rightarrow$ data_C1 := 02h
Cho PC_0	0	0	0	0	0	0	0	1	$\Rightarrow$ data_C0 := 01h

- Chương trình viết bằng Pascal như sau:

```
Begin
{Gán các biến}
Port[addr_dk] := data_dk;
Repeat
Var1 := port[addr_C];
Until   ((var1 and data_C1)=data_C1);
Port[addr_A] := 's';
Port[addr_C] :=0;
Port[addr_C] := data_C5;
End
```

CHƯƠNG 3: GHÉP NỐI TRAO ĐỔI DỮ LIỆU NỐI TIẾP

3.1 Giới thiệu chung về trao đổi dữ liệu nối tiếp

3.1.1 Yêu cầu trao đổi dữ liệu nối tiếp

Trao đổi dữ liệu nối tiếp là trao đổi lần lượt từng bit, trên một đường dây duy nhất. Do đó ít tốn kém về đường dây, nhưng tốc độ truyền thấp. Sự trao đổi này do các yêu cầu:

- Thiết bị phát hay nhận dữ liệu từng bit. Ví dụ VXL 8085 phát từng bit ở lối ra nối tiếp SOD (chân số 4) và nhận dữ liệu ở lối vào nối tiếp SID (chân số 5); và TBNV nối tiếp như máy cassette, đĩa từ, máy in nối tiếp …
- Khoảng cách giữa hai thiết bị trao đổi dữ liệu là lớn (có thể đến hàng ngàn km hoặc hơn), do vậy việc dùng 8 đường dây truyền dữ liệu song song là rất tốn kém. Ngoài ra, người ta có thể dùng luôn đường dây điện thoại để trao đổi dữ liệu, thông qua bộ điều chế - giải điều chế (MODEM)

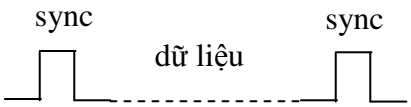
Điều khiển nối tiếp có 2 loại truyền dữ liệu:

-truyền dữ liệu đồng bộ (tốc độ nhanh).

- truyền dữ liệu không đồng bộ (tốc độ chậm).

1. Trong trao đổi dữ liệu đồng bộ:

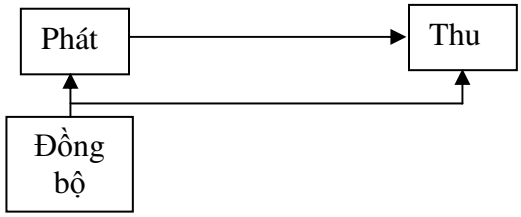
Dữ liệu được truyền 1 lần gồm 1 khối nhiều byte, được chặn đầu và cuối bởi ký tự SYNC(mã ASCII là 16h). Mỗi khối có thể hàng trăm byte hay hàng trăm Kbyte.



Để truyền đi cần có sự đồng bộ về tốc độ giữa máy phát và máy thu, có 2 cách để đồng bộ

- Một máy phát xung đặt ở nơi phát và truyền tới máy thu bởi một đường dây khác. Do đó tốn thêm một đường dây

Hệ này sử dụng 1 máy phát xung đồng bộ để cung cấp xung đồng bộ cho cả hai bên phát và thu.

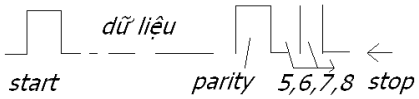


- Máy thu chỉ ở trạng thái chờ, khi có ký tự đồng bộ SYNC thì máy phát xung nhịp trong máy thu bật lên và chuyển sang chế độ thu xung đồng bộ được kích hoạt bởi kí tự SYNC.

2. Trong trao đổi thông tin không đồng bộ:

Các byte được đặt trong một khung và truyền độc lập với nhau, gọi là một lời tin.

Dạng thông tin truyền như sau:



Dữ liệu truyền có thể 5,6 ,7,8 bit; thông thường là 7 bit (nếu truyền số liệu và chữ cái) hay 8 bit (nếu truyền cả các ký tự mở rộng để điều khiển hết File). Một lời tin gồm:

- 1 bit Start thường là mức 0 (+12V). Theo chuẩn RS232C, tín hiệu từ máy tính qua cổng Com, mức điện áp là ±12 V và được qui định:0=12V, 1= -12V.
- 5÷8 bit dữ liệu, ở đây dữ liệu truyền đi là mã ASCII, như số "0" sẽ được truyền đi là mã ASCII = 48 = 030h chứ không phải là truyền đi "0".
- 1 bit bậc dùng để kiểm tra tính chẵn lẻ dữ liệu truyền. Tổng số bit 1 của một lời tin (kể cả bit chẵn lẻ) sẽ được ghi vào bit chẵn lẽ và được kiểm tra so sánh ở nơi phát và nơi thu.
- 1, 1.5 hay 2 bit Stop (tuỳ theo sự lựa chọn trước khi trao đổi) cũng ở mức 1. Số bit Stop thực chất là độ dài của tín hiệu Stop ở mức logic 1.

Bit Start dùng để đồng bộ xung nhịp ở máy phát và máy thu.

Nếu khoảng cách gần dưới 300m, sự thu phát là không cần MODEM.

Ở những khoảng cách lớn người ta không dùng máy phát xung ngoài mà dùng MODEM để tạo xung đồng bộ cho cả máy phát và máy thu. Việc dùng MODEM là để chống nhiễu trên đường truyền. Trên đường dây điện thoại, người ta không phát từng bit 0/1 mà dùng MODEM (modulation-demodulation) để điều chế tín hiệu thành dạng xoay chiều (0÷2400 Hz, 1÷1200 Hz) và truyền tín hiệu xoay chiều đó.

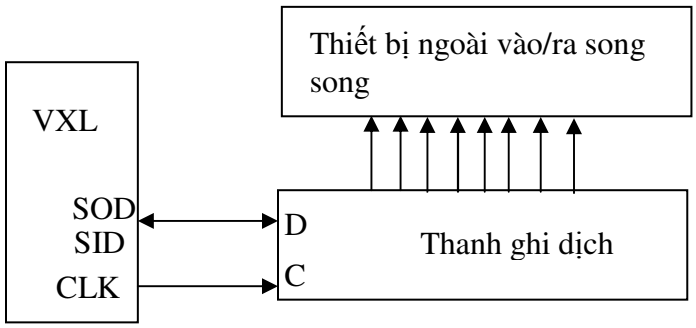
3.1.2 Mạch trao đổi dữ liệu nối tiếp.

1. Mạch không cần bộ ghép nối:

-sử dụng 2 chân ra-vào nối tiếp SID (serial IN) và SOD (serial OUT) của vi xử lý 8085 để trao đổi.

-có thể dùng 1 chân của lối vào ra song song của VXL để làm chân vào ra nối tiếp.

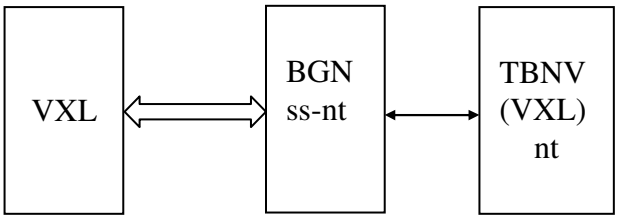
Trao đổi tin kiểu này không phức tạo trong cấu tạo mạch nhưng phức tạp trong quá trình lập trình và theo dõi thiết bị, đòi hỏi nhiều thời gian xử lý.



Hình 3.1: Mạch không cần BGN

2. Mạch trao đổi cần ghép nối song song-nối tiếp:

BGN có chức năng chuyển dữ liệu song song từ máy tính (hệ VXL) thành tín hiệu dạng nối tiếp và truyền đến TBNV. Để truyền chính xác, BGN ngoài chức năng chuyển dữ liệu song song thành nối tiếp, nó còn tạo ra các bit Start, stop, parity để đóng khung dữ liệu. Loại này làm việc đơn giản nhưng không mở rộng được thiết bị ngoại vi, khoảng cách truyền tin ngắn.

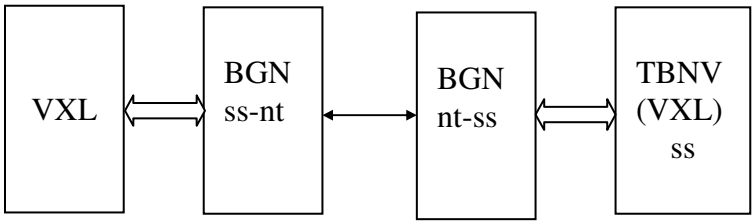


Hình 3.2: Mạch cần BGN ss-nt

Trong đó, song song là ss, nối tiếp là nt.

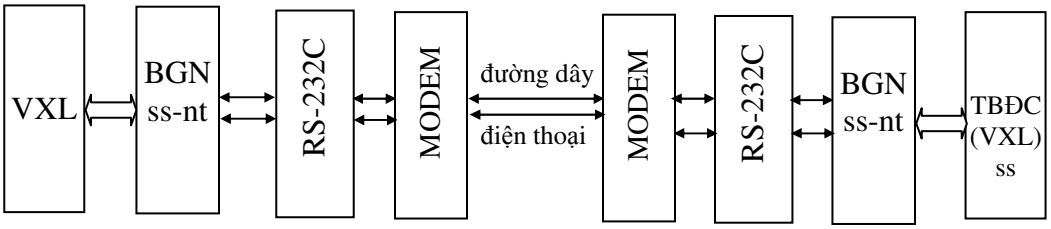
3. Mạch có KGN song song- nối tiếp và nối tiếp-song song:

Hệ này cũng giống như hệ trên, chỉ khác là TBNV làm việc với dữ liệu song song, nên cần có BGN trung gian chuyển dữ liệu song song thành nối tiếp.



Hình 3.3: Mạch cần BGN ss-nt và nt-ss

4. Mạch có BGN, MODEM và RS232C:



Hình 3.4: Mạch cần BGN ss-nt, RS 232 và MODEM.

Hệ này thông dụng và thường xuyên gặp trong thực tế. Trong đó VXL thường là 1 máy vi tính, còn thiết bị đầu cuối (TBĐC) có thể là 1 đối tượng bất kỳ. Nếu TBĐC là máy vi tính thì hệ thống dùng để trao đổi File.

Chú ý:

Chuẩn RS232C có mức điện áp là ±12V.

Chuẩn RS422 có mức điện áp là ±15V.

Chuẩn RS485 có mức điện áp là ±30V (điện thoại).

Để chuyển từ RS232C thành RS485 hay RS422 sẽ có những bộ chuyển (shift) trung gian, thông thường đó là những bộ đệm, vừa tạo mức tín hiệu,vừa phối hợp trở kháng.

3.1.3 Thủ tục trao đổi dữ liệu nối tiếp

1. Thủ tục phát dữ liệu TxD (Tranceiver Data) :

- TBĐC (hay MVT) gửi tín hiệu $\overline{DTR}$ (Data Terminal Ready) ở mức 0 đến MODEM báo hiệu đã sẵn sàng.
- MODEM trả lời TBĐC bằng tín hiệu $\overline{DSR}$ (Date Set Ready) ở mức 0.
- Nếu TBĐC có 1 ký tự sẵn sàng gửi đi, nó gửi $\overline{RTS}$ (Request To Send) đến MODEM
- MODEM gửi tín hiệu CD (Carrier Detect) cho TBĐC để báo hiệu liên lạc đã thông với MTV.
- Khi MODEM sẵn sàng phát số liệu trên đường dây, MODEM phát xung nhịp và tín hiệu $\overline{CTS}$ (Clear To Send) đến TBĐC (xung nhịp ở trên đường dây).
- TBĐC gửi các ký tự số liệu (SDU) cho MODEM.
- Khi TBĐC gửi xong, nó nâng tín hiệu $\overline{RTS}$ lên cao để báo cho modem biết là đã phát xong.
- MODEM trả lời cho TBĐC bằng tín hiệu $\overline{CTS}$ ở mức cao, báo hiệu hoàn thành việc truyền tin.

Qua đó ta thấy việc gửi đi kí tự bằng modem nhiều thủ tục và do đó tốc độ chậm

2. Thủ tục nhận dữ liệu RxD (Receiver Data)

Trong quá trình nhận tin, phải bảo đảm máy tính đã sẵn sàng nhận ,khi đó :

- TBĐC phía thu gửi $\overline{DTR}$ mức thấp cho MODEM báo nó đã sẵn sàng (Data terminal ready)
- MODEM thu gửi tín hiệu trả lời bằng $\overline{DSR}$.
- MODEM thu nhận tín hiệu CD từ đường dây và kích gửi tín hiệu nhịp MODEM cho TBĐC.
- TBĐC phát tín hiệu $\overline{RTS}$ cho modem (Request to Send).
- MODEM gửi tín hiệu $\overline{CTS}$ mức thấp cho TBĐC (Clear to Send) để báo MODEM sẵn sàng nhận tin.
- MODEM nhận *TxD* trên đường dây, sau đó phải điều chế thành các bit 0/1 và lần lượt gửi chuỗi tín hiệu *RxD* cho TBĐC.
- Khi thu xong, TBĐC nâng tín hiệu $\overline{RTS}$ lên cao, báo cho MODEM là đã thu xong
- MODEM trả lời bằng nâng tín hiệu $\overline{CTS}$ lên cao, để báo quá trình nhận đã xong.

Việc tiến hành chuyển dữ liệu ss/nt hay nt/ss được thực hiện bởi các thanh ghi dịch (Shift register). Các thanh ghi này có tác dụng khi nhận 1 byte song song ở đầu vào nó sẽ cho ở đầu ra một chuỗi bit kế tiếp nhau theo thứ tự từ bit thấp đến bit cao của byte. Đối với máy tính, các chức năng trên được thực hiện thông qua vi mạch 8251 của hãng INTEL.

3.2 Mạch thu phát di bộ vạn năng UART 8250

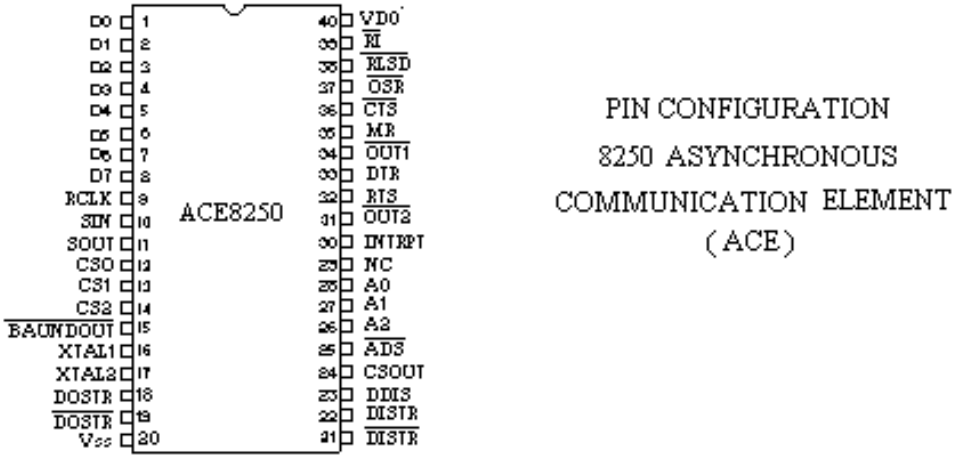
Mạch URAT 8250 (Universal Asynchronous Receiver/Transmitter) được lắp ráp trong các máy IBM XT, là một IC thu phát không đồng bộ có thể lập trình được, nó thực hiện chức năng giao tiếp giữa máy tính và TBNV.

Trong các máy PC/AT hiện nay sử dụng UART 16450 còn trong PS/2 và các AT đời mới lại dùng UART 16550 vì 8250 và 16450 không đạt đến tốc độ 115200 baud được. Thực ra tốc độ này ứng dụng tốt trong kỹ thuật truyền tin trên mạng, còn trong các thiết bị công nghiệp TBNV nói chung, thường chỉ cần tối đa 19200 baud là được, thông dụng là 9600 baud.

3.2.1 Sơ đồ khối và chức năng các khối của UART 8250

1. Mô tả vi mạch:

Vi mạch 8250 có 40 chân như trên hình vẽ :

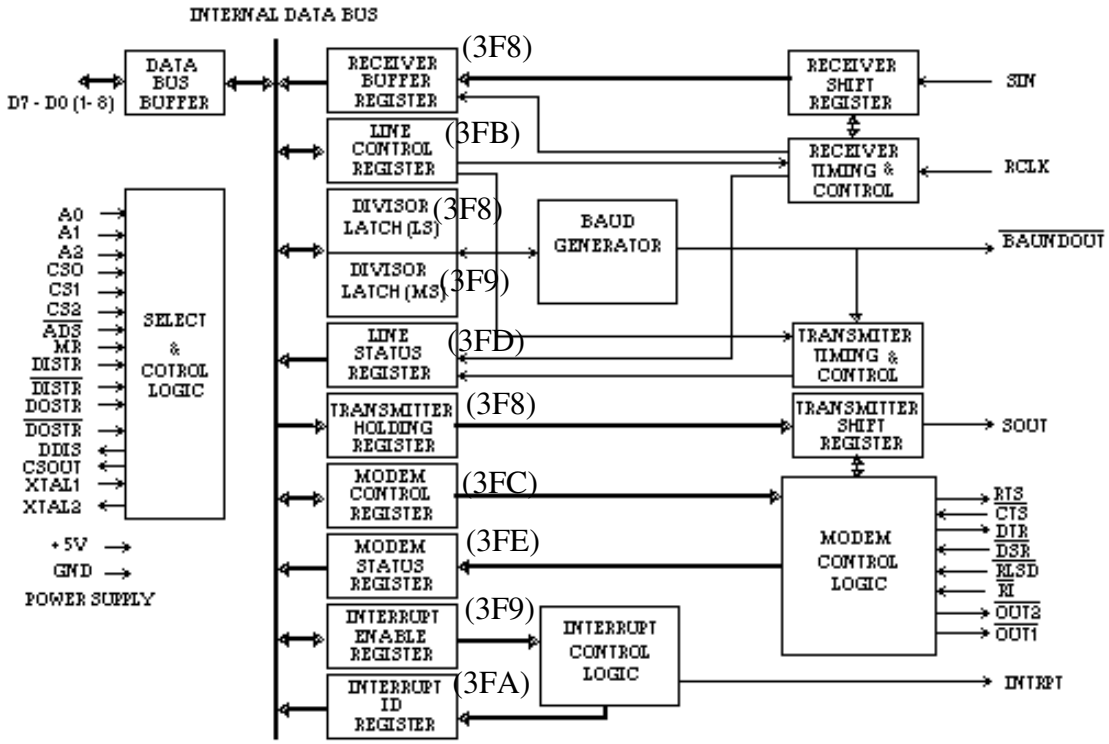


Hình 3.5: Sơ đồ chân 8250

Chức năng các chân gồm:

- Các chân từ 1- 8 (D0- D7): Các chân dữ liệu, 8 chân này được nối với bus dữ liệu. Qua 8 chân dữ liệu này mà dữ liệu được đi từ bus tới dưới dạng song song sau khi qua vi mạch 8250 biến đổi thành tuần tự để đi tới cổng RS232. Dữ liệu từ cổng RS232 tới dưới dạng tuần tự, sau khi đi qua vi mạch 8250 được biến đổi thành song song và qua 8 chân này truyền lên bus.
- Chân số 9 (RCLK): nhận xung đồng hồ đưa ra từ bộ tạo tốc độ.
- Chân số 10 (SIN): nhận dữ liệu tuần tự đi tới từ cổng RS232.
- Chân số 11 (SOUT): Gửi dữ liệu tuần tự sang cổng RS232.
- Chân số 12,13,14 (CS0 -CS2): Chip select - tín hiệu chọn vi mạch.
- Chân số 15 ($\overline{BOUDOUT}$) : Lối ra của bộ tạo tốc độ.
- Chân số 16,17 (XTAL1,XTAL2) :Lấy xung Clock từ bộ phát nhịp bằng thạch anh tới với tần số 3,072 MHz.
- Chân 18(DOSTR): I/O write.
- Chân số 19 ($\overline{\text{DOSTR}}$) : Đảo của I/O write
- Chân số 20 (Vss): nối đất

- Chân số 21 (DISTR) : I/O Read .
- Chân số 22 ($\overline{\text{DISTR}}$): Đảo của I/O Read .
- Chân số 26,27,28 (A0 - A2): nhận địa chỉ truyền tới để giải mã lệnh .
- Chân số 30 (INTRPT) : Interrupt Control Logic .
- Chân số 32 ($\overline{\text{RTS}}$) : Request To Send (yêu cầu gửi) .
- Chân số 33 ($\overline{\text{DTR}}$) : Data Terminal Ready .(Dữ liệu sẵn sàng)
- Chân số 35 (MR) : Reset .
- Chân số 36 ($\overline{\text{CTS}}$) : Clear To Send
- Chân số 37 ($\overline{\text{DSR}}$) : Data Set Ready .
- Chân số 38 ($\overline{\text{DCD}}$) : Data Carrier Detect .
- Chân số 39 ($\overline{\text{RI}}$): Ring Indicate .
- Chân số 40 (VDo) : nối với nguồn một chiều +5V.



Hình 3.6: Sơ đồ khối 8250

Như vậy, 8250 có 11 thanh ghi nội, với cách truy cập như bảng sau:

Bảng 3.1: Địa chỉ các thanh ghi của 8085

DLAB	A_2	A_1	A_0	Đọc/ghi	Thanh ghi

0	0	0	0		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)
1	0	0	1	Đọc/ghi	chốt chia phần cao (MSB)
x	0	1	0	Đọc	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 modem (MSR)
x	1	1	1	Đọc/ghi	nhớ nháp (DM)

Mỗi thanh ghi trong 8250 tương ứng với một địa chỉ cổng, trong đó có hai thang ghi đặc biệt, nó có chức năng có thể thay thế tuỳ thuộc giá trị bit DLAB (divitor latch access bit –Bit truy cập 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 , hai thanh ghi dùng để đệm và nhận dữ liệu, và IER.

2. Mô tả các thanh ghi:

a) Thanh ghi đệm thu (Receiver Buffer Register – RBR) 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 đệm thu (lưu ý thanh ghi có 8 bit). CPU chỉ đọc dữ liệu trong thanh ghi này vì nó thao tác với BUS dữ liệu song song tối thiểu mỗi lần là 1 byte.

b) Thanh ghi đệm phát (Transmitter Holding Register – THR) tương ứng với DLAB = 0, địa chỉ là 3F8h (2F8h)

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

c) Thanh ghi cho phép ngắt (Interrupt Enable Register - IER), bit DLAB = 0 , địa chỉ 3F9h (COM1) hay 2F9h (COM2)

Thanh ghi này thực hiện để cho phép/cấm các nguyên nhân gây ra ngắt khác nhau. Trong khi 8250 hoạt động, có tác động đến CPU thông qua chân INTRPT (chân 30), mỗi bit trong các bit D3, D2, D1, D0 ở mức cao sẽ cho phép các hoạt động tương ứng với các bit để đưa ra nguyên nhân ngắt tương ứng, và cấm ngắt khi ghi 0 vào các bit.

Bit	Chức năng
7-4	Dự trữ, luôn đặt bằng 0.
3	cho phép 8250 phát yêu cầu ngắt trạng thái MODEM

2	cho phép 8250 phát yêu cầu ngắt theo trạng thái dừng truyền nhận hoặc ngắt dòng thông tin (lỗi chẵn lẽ, tràn khung).
1	cho phép 8250 phát yêu cầu ngắt khi thanh ghi đệm phát rỗng.
0	cho phép 8250 phát yêu cầu ngắt khi thanh ghi đệm thu đầy, dữ liệu đã sẵn sàng

d) Thanh ghi chốt chia phần thấp (Divisor Latch - LS) tương ứng DLAB = 1, địa chỉ là 3F8h, hay 2F8h.

e) Thanh ghi chốt chia phần cao (Divisor Latch – MS) tương ứng DLAB = 1, địa chỉ là 3F9h, hay 2F9h.

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

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

- tần số nhịp chuẩn do đồng hồ thạch anh của 8250 phát ra, nó bằng 1,8432 MHZ
- tốc độ tối đa tính toán là 115200 baud.
- tốc độ baud để thực hiện các chân SIN, SOUT.

Bảng chỉ mối liên hệ giữa tốc độ số liệu và các hệ số chia ở dạng hexa decimal như sau:

Baud	**Số chia**	**Baud**	**Số chia**
50	0900	1200	0060
110	0417	2400	0030
150	0300	4800	0018
300	0180	9600	000C
600	00C0	11520	0001

Qua đó ta thấy giá trị trong thanh chốt chia cần 1 byte rưỡi để lưu tất cả. Để thuận tiện, người ta sẽ dùng 2 byte và ghi vào 2 thanh ghi, đó chính là LSB và MSB.

f) Thanh ghi nhận dạng ngắt (Interrupt Identification Register - IIR), địa chỉ là 3FAh, hay 2FAh.

Thanh ghi nhận dạng ngắt (chỉ có thể đọc) chứa mã mức ưu tiên cao nhất của các yêu cầu ngắt (tại chân 30-INPUT của 8250) đang chờ được phục vụ. Nếu có nhiều ngắt cùng một lúc thì IIR sẽ chứa mã ngắt nào cần xử lý trước. Do vậy khi cần xử lý ngắt, CPU cần đọc bit D_0 của thanh ghi này để biết có yêu cầu ngắt và kiểm tra các bit D_1 và D_2 để xác định nguồn gốc các yêu cầu ngắt. Sau khi 8250 bị Reset, chỉ có yêu cầu ngắt ưu tiên số một được phục vụ. Ta có thể thay đổi

điều này bằng cách dùng mặt nạ che đi các bit yêu cầu ngắt nào đó bằng cách ghi vào thanh ghi IIR các giá trị bit thích hợp.
Các bit $D_7 \div D_3$ luôn đặt bằng 0.
D2, D1: mã hoá các yêu cầu ngắt có mức ưu tiên cao nhất đang chờ phục vụ.

D_2	D_1	Mức ưu tiên	Nhận diện ngắt
0	0	4	Trạng thái MODEM ⇒ chương trình đọc trạng thái MODEM
0	1	3	Thanh ghi đệm phát rỗng
1	0	2	TG 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

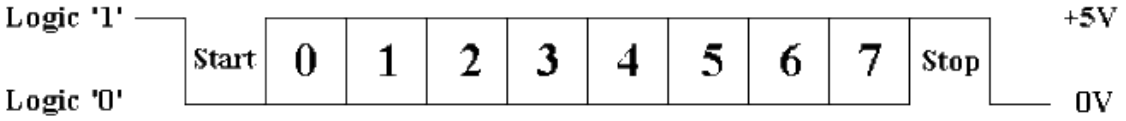
Từ các nhận dạng ngắt trên, VXL sẽ chuyển đến các địa chỉ chứa các chương trình con phục vụ ngắt. VXL sẽ xoá các bit D0, D1, D2 về 0 sau các thao tác:
-đọc thanh ghi trạng thái chuỗi.
-đọc dữ liệu từ thanh ghi đệm thu.
-ghi vào bộ phát hoặc đọc thanh ghi nhận diện ngắt.
-đọc trạng thái thanh ghi thao tác MODEM (RS232-C).

g) Thanh ghi điều khiển dòng (Line Control Register – LCR), địa chỉ là 3FBh, hay 2FBh.

Thanh ghi này còn có tên là thanh ghi định khuôn dạng dữ liệu vì nó quyết định khuôn dạng của dữ liệu truyền trên đường dây.

Bit	Chức năng
7	=1 truy nhập thanh ghi chốt chia (LSB & MSB) để lập tốc độ baud =0 truy nhập IER,THR,RBR
6	Đặt cho phép tín hiệu Break
5-3	B5 B4 B3 Chọn chẵn lẻ x x 0 Non 0 0 1 Odd 0 1 0 Even 1 0 1 High Parity 1 1 1 Low Parity
2	= 1 thì dùng 1.5 bit Stop nếu độ dài từ là 5, còn nếu độ dài từ 6-8 bit thì có 2 bit Stop. = 0 thì dùng 1 bit Stop
1-0	B1 B0 Số bit data 0 0 5 0 1 6 1 0 7 1 1 8

Để hiểu tín hiệu Break, ta xem ví dụ khung dữ liệu trên với 8 bit data, không kiểm tra chẵn lẻ và 1 bit Stop.



Khi đường dây ở trạng thái Mark, mức logic là 1. Bit Start sẽ là mức logic 0. Sau đó từng bit data sẽ được gởi trên đường dây. Bit Stop (logic 1) sau đó được gán vào để kết thúc việc truyền. Sau bit Stop sẽ là bit logic 0, có nghĩa là bit Start của một dữ liệu mới. Nếu không có thêm dữ liệu truyền đi, đường dây phải ở trạng thái rỗi, mức logic 1. Do đó nếu đường dây giữ trạng thái 0 sau bit Stop trong một thời gian bằng độ dài 1 từ, tín hiệu Break sẽ được tạo ra.

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

h) Thanh ghi điều khiển MODEM (MODEM Control Register – MCR), địa chỉ 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 DTS (chân 32) của 8250.

Bit	Chức năng
7-5	Dự 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 của 8250. =0 làm việc ở chế độ thường.
3	=1 : kích hoạt kết xuất OUT2
2	=1 : kích hoạt kết xuất OUT1
1	=1 : tín hiệu tại chân $\overline{\text{RTS}}$ của 8250 được chuyển về mức 0, máy tính phát chuyển tín hiệu $\overline{\text{RTS}}$ tới MODEM để báo hiệu nó đã sẵn sàng phát dữ liệu.
0	=1 : tín hiệu tại chân $\overline{\text{DTR}}$ của 8250 đạt mức 0, lúc này máy tính gửi tín hiệu $\overline{\text{DTR}}$ tới MODEM (hay tới MVT đầu kia nếu giao tiếp không dùng MODEM) để báo hiệu nó sẵn sàng làm việc (tín hiệu gửi tới chân số 4 của DB9 hay chân 20 của DB25).

i) Thanh ghi trạng thái dòng (Line Status Register – LSR) địa chỉ 3FDh

Chức năng thanh ghi này cho biết trạng thái dòng tín hiệu trên đường dây như thế nào, các bit D0÷D5 đều có thể gây ra ngắt nếu các bit tương ứng trong thanh ghi IER được lập .

Các bit trạng thái có ý nghĩa như sau:

Bit	Chức năng

7	Dự trữ
6	=1: báo thanh ghi chuyển phát rỗng (Transmitter Shift Register Empty), báo một kí tự đã được phát đi, bit này bị xoá khi có một kí tự chuyển từ THR sang TSR.
5	=1: báo thanh ghi đệm truyền rỗng, ký tự chuyển từ THR sang TSR (Transmitter Shift Register), bit này bị xoá khi CPU đưa kí tự đến THR.
4	=1: báo nhận được tín hiệu Break, bit này bị xoá khi đọc thanh ghi LSR.
3	=1: báo lỗi khung (framming error) chẳng hạn bit stop =0, bit này bị xoá khi CPU đọc thanh ghi LSR.
2	=1: nhận báo được lỗi kiểm tra chẵn lẽ (parity error), bit này bị xoá khi CPU đọc thanh ghi LSR.
1	=1: có lỗi tràn (over run) xảy ra khi kí tự trước đó bị mất, bit này bị xoá khi CPU đọc thanh ghi LSR.
0	=1: đã nhận được 1 ký tự và để nó trong thanh ghi đệm thu (RBR), bit này bị xoá về 0 khi CPU đọc thanh ghi RBR.

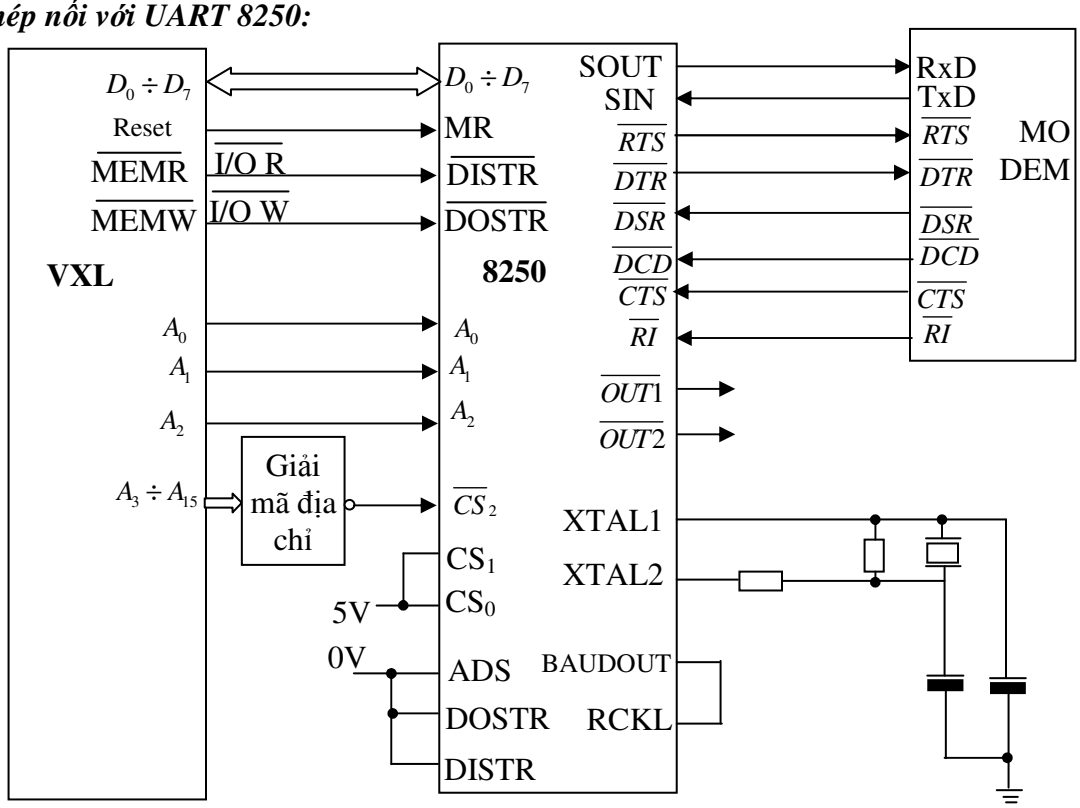
j) Thanh ghi trạng thái modem (MODEM Status Regisster – MSR), địa chỉ 3FEh

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.

Bit	**Chức năng**
7	=1 : DCD đang hoạt động, báo liên lạc với MODEM đã được thiết lập.
6	=1 : RI đang hoạt động
5	=1 : MODEM (hay MVT đầu kia) gởi tín hiệu DSR (Data Set Ready) về máy tính để cho biết đã sẵn sàng nhận dữ liệu (qua chân 6 của DB9 hay DB25)
4	=1 : MODEM (hay MVT đầu kia) gởi tín hiệu CTS (Clear To Send) về máy tính để cho biết đã sẵn sàng nhận dữ liệu (qua chân 8 của DB9 hay chân 5 của DB25).
3	=1 : bit D7 vừa đổi trạng thái
2	=1 : bit D6 vừa đổi trạng thái
1	=1 : bit D5 vừa đổi trạng thái
0	=1 : bit D4 vừa đổi trạng thái

k) Thanh ghi nhớ nháp (Scratch Register), địa chỉ 3FFh
(dành cho CPU, ít sử dụng nên không nghiên cứu)

3.2.2 Ghép nối với UART 8250:



Hình 3.7: Sơ đồ ghép nối với 8250

3.2.3 Lập trình cho UART 8250:

Lưu đồ thuật toán:

a. Khởi phát vi mạch theo trình tự sau:
   - Ghi vào thanh ghi dạng số liệu theo :
     + D_7 = DLAB = 1 để chuẩn bị ghi chốt cho bộ chia tốc độ.
     + Khung tin với số bit Stop
     + Số bit của tin
   - Ghi giá trị bộ chia tần số vào các thanh ghi chốt chia LSB, MSB tuỳ theo tốc độ baud.
   - Ghi các phép yêu cầu ngắt vào thanh ghi yêu cầu ngắt.

b. 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 đưa lệnh DRT (bit D_0-1) điều khiển MODEM chuẩn bị phát.
   - Đọc thanh ghi trạng thái MODEM để kiểm tra các bit DSR (D_5), RI (D_6), DCD (D_7) 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 đệm phát từ MVT.
   - Ghi lệnh RTS (D_1=1) vào thanh ghi điều khiển MODEM để điều khiển phát số liệu.

c. Nhận số liệu nối tiếp:
   - Ghi lệnh DTR (D_0=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 (D_0=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.

Ví dụ 8250 cần trao đổi tin 8 bit, tốc độ 9600 baud, không kiểm tra chẵn lẻ, 2 bit Stop, địa chỉ làm việc đầu 03F8h.
Ghi thanh ghi điều khiển dòng
D0D1=11 , D2=1, D3=0, D4=0, D5=0, D6=0, D7=1 ⇒ data1_LCR=87h
nếu D7=0⇒data2_LCR=07h
addr_LCR=03FBh
Ghi thanh ghi chốt bộ chia
9600 baud ⇒ số chia=000C
data_LS = 0Ch và data_MS = 00h
addr_LS=03F8h và addr_MS=03F9h

Ghi 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 (D_5), RI (D_6), DCD (D_7) = 1 ⇒data1_MSR=E0h

Đọc thanh ghi trạng thái dòng
Addr_LSR=3FDh
Không lỗi D1 D2 D3 = 0; t/g đệm rỗng D5 D6 =1⇒data1_LSR=60h

Chương trình con thực hiện các công việc trên:
1. *Khởi phát vi mạch theo trình tự sau:*

```
port[addr_LCR] :=data1_LCR;
port[addr_LS] :=data_LS;
port[addr_MS] :=data_MS;
port[addr_LCR] :=data2_LCR;
```

```
port[addr_ IER] :=data_IER;
        2. Phát số liệu nối tiếp
repeat
Var1 := port[addr_IIR];
Until ((var1 and THR_rong)=THR_rong);
port[addr_MCR] := data_DRT;
repeat
Var2 := port[addr_MSR];
Until ((var2 and data1_MSR)=data1_MSR);
Repeat
var3 := port[addr_LSR];
until ((var3 and data1_LSR)=data1_LSR);
port[addr_THR] :=$50;  {số liệu truyền đi}
port[addr_MCR] := data_RTS;

      3.Nhận số liệu nối tiếp
port[addr_MCR] := data_DRT;
Repeat
Var2 := port[addr_MSR];
Until ((var2 and data1_MSR)=data1_MSR);
Repeat
var3 := port[addr_LSR];
until ((var3 and data1_LSR)=data1_LSR);
Var4 :=port[addr_RBR];
```

CHƯƠNG 4: GHÉP NỐI ĐIỀU KHIỂN NGẮT

4.1 Khái niệm và phân loại ngắt

4.1.1 Khái niệm

Mỗi khi một thiết bị phần cứng hay một chương trình cần đến sự giúp đỡ của CPU, nó gửi đi một tín hiệu hoặc lệnh gọi là ngắt (Interrupt Request – IRQ) đến bộ vi xử lý chỉ định một công việc cụ thể nào đó mà nó cần CPU thực hiện. Khi bộ vi xử lý nhận được tín hiệu ngắt, nó thường tạm ngưng tất cả các hoạt động khác và kích hoạt một chương trình con đang có trong bộ nhớ gọi là chương trình xử lý ngắt (Interrupt Service Routine _ ISR) tương ứng với từng số liệu ngắt cụ thể. Sau khi chương trình xử lý ngắt làm xong nhiệm vụ, các hoạt động của máy tính sẽ tiếp tục lại từ nơi đã bị tạm dừng lúc xảy ra ngắt.

4.1.2 Phân loại ngắt

Có ba loại ngắt chính, đầu tiên là các ngắt được tạo ra bởi mạch điện của máy tính nhằm đáp lại một sự kiện nào đó như nhấn phím trên bàn phím ... Các ngắt này được bộ điều khiển ngắt 8259A quản lý. 8259A sẽ ấn định mức độ ưu tiên cho từng ngắt rồi gửi đến CPU. Thứ hai là các ngắt do CPU tạo ra khi gặp phải một kết quả bất thường trong khi thực hiện chương trình như chia cho 0 chẳng hạn … Cuối cùng là các ngắt do chính chương trình tạo ra nhằm gọi các chương trình con ở xa đang nằm trong ROM hoặc RAM, các ngắt này gọi là ngắt mềm chúng thường là bộ phận của các chương trình con phục vụ của ROM-BIOS hoặc của DOS.

Ngoài ba loại ngắt trên còn có loại ngắt đặc biệt là ngắt không bị che NMI đòi hỏi CPU phục vụ ngay khi có yêu cầu. Loại ngắt này thường được dùng để báo hiệu sự cố như sụt điệp áp hay lỗi bộ nhớ. Như vậy NMI là mức ngắt có độ ưu tiên cao nhất .

Các ngắt trong PC có thể chia thành 4 nhóm như sau :

- Các ngắt vi xử lý :

Thường gọi là các ngắt logic được thiết kế sẵn trong bộ VXL. Bốn trong số các ngắt này (0,1,3,4) do chính bộ VXL tạo ra, còn ngắt 2 (NMI) sẽ được kích hoạt khi có tín hiệu tạo ra bởi một trong các thiết bị ngoài.

1. Các ngắt cứng

Được thiết kế sẵn trong phần cứng của của PC, tám ngắt trong số các ngắt này (2,8,9,từ Bh đến Fh) được gắn chết vào trong bộ VXL hoặc vào bảng mạch chính của hệ thống. Tất cả các ngắt cứng đều do 8259A điều khiển .

2. Các ngắt mềm

Những ngắt này là một phần của các chương trình ROM -BIOS, các số hiệu dành cho các ngắt của ROM- BIOS là từ 10h đến 1C h và 48h .

Ngoài ra còn có các ngắt DOS và ngắt BASIC phục vụ hệ điều hành DOS và chương trình BASIC .

3. Các ngắt địa chỉ

Bảng 4.1 : Bảng chức năng các ngắt trong máy vi tính

INT (Hex)	IRQ	Common Uses
00	Exception Handlers	Chia một số cho 0
01	Exception Handlers	Thực hiện từng bước
02	Non-Maskable IRQ	Non-Maskable IRQ (Các lỗi chẵn lẻ)
03	Exception Handlers	Điểm dừng chương trình do người sử dụng đặt
04	Exception Handlers	Vượt quá nội dung thanh ghi hay bộ nhớ
05 - 07	Exception Handlers	-
08	Hardware IRQ0	Bộ thời gian hệ thống
09	Hardware IRQ1	Bàn phím
0A	Hardware IRQ2	Redirected
0B	Hardware IRQ3	Cổng nối tiếp COM2/COM4
0C	Hardware IRQ4	Cổng nối tiếp COM1/COM3
0D	Hardware IRQ5	Reserved/Sound Card
0E	Hardware IRQ6	Điều khiển đĩa mềm
0F	Hardware IRQ7	Cổng song song
10 - 6F	Software Interrupts	Ngắt ROM-BIOS hay DOS
70	Hardware IRQ8	Đồng hồ thời gian thực
71	Hardware IRQ9	Redirected IRQ2
72	Hardware IRQ10	Reserved
73	Hardware IRQ11	Reserved
74	Hardware IRQ12	PS/2 Mouse
75	Hardware IRQ13	Math's Co-Processor
76	Hardware IRQ14	Hard Disk Drive
77	Hardware IRQ15	Reserved
78 - FF	Software Interrupts	-

Ba trong số các ngắt này trỏ đến ba bảng rất quan trọng, đó là bảng khởi tạo màn hình, bảng cơ sở đĩa và bảng các ký tự đồ thị. Các bảng này chứa các tham số được ROM

BIOS dùng khi khởi động hệ thống và tạo các ký tự đồ thị. Các số hiệu dành cho các ngắt này là từ 1D hex đến 1F hex.

4.2 Xử lý ngắt của nhiều thiết bị ngoại vi

Thông thường một MVT cần xử lý ngắt của nhiều TBNV, do đó cần giải quyết các vấn đề sau:

- Cho phép hoặc cấm ngắt cho một yêu cầu ngắt của một TBNV.
- Ghi nhận và sắp xếp nhiều yêu cầu ngắt theo thứ tụ ưu tiên.
- Xác định nguồn ngắt do TBNV nào gây ra.
- Tạo vector ngắt, tức địa chỉ ô nhớ của lệnh đầu tiên trong chương trình con phục vụ ngắt.

4.2.1 Cho phép hoặc cấm ngắt

1. *Cho phép hoặc cấm ngắt bằng lệnh*

- Lệnh cấm ngắt DI (Disable interrupt) và cho phép ngắt EI (Enable Interrupt) của 8085.
- Ghi vào thanh ghi cờ nội dung bit IF (Interrupt Flag) các giá trị:

IF =1 : cho phép ngắt
IF=0 : cấm ngắt

2. *Cho phép hoặc cấm ngắt bằng mạch phần cứng*

Với lối vào NMI và yêu cầu ngắt của TBNV, ta có thể dùng các mạch lật để điều khiển cho phép hoặc cấm ngắt.

4.2.2 Sắp xếp ưu tiên ngắt

Theo một trong ba phương pháp sau:

1. *Phương pháp hỏi vòng (polling):*

Dùng chương trình, đọc theo thứ tự ưu tiên các bit trạng thái của các TBNV và kiểm tra chúng. Nếu bit nào xác lập lên 1, tức TBNV tương ứng có yêu cầu ngắt, và VXL chuyển sang chương trình con ngắt phục vụ TBNV đó. Sau đó lại đọc và kiểm tra bit trạng thái TBNV có mức ưu tiên thấp hơn.

Phương pháp này đơn giản về thiết bị nhưng mất thời gian hỏi vòng.

2. *Phương pháp ngắt cứng*

Để tránh mất thời gian hỏi vòng, người ta nối các bit trạng thái của TBNV với mạch OR, lối ra cổng OR nối vào chân INTR của VXL. Khi bất kỳ TBNV nào có yêu cầu ngắt, VXL sẽ nhận biết qua INTR và tiến hành hỏi vòng.

Phương pháp này giảm được thời gian kiểm tra trạng thái khi chưa có yêu cầu ngắt nào, nhưng cũng không thay đổi được thứ tự ưu tiên.

3. *Phương pháp xử lý ưu tiên bằng mạch cứng*

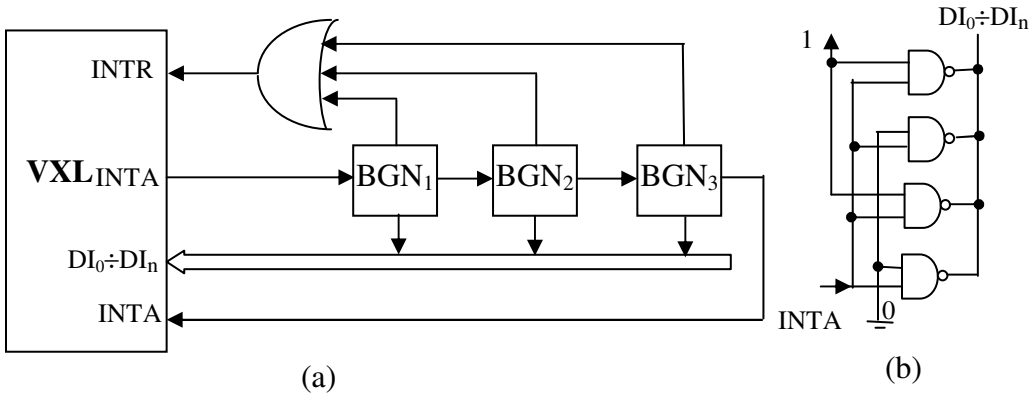
Có hai loại ưu tiên theo vị trí và ưu tiên tuỳ theo mạch so sánh.

❖ *Ưu tiên theo vị trí đặt gần VXL*

Sơ đồ mạch ví dụ trên hình 4.1, với hoạt động như sau:

- Tín hiệu trả lời ngắt INTA (Interrupt Acknowledge) từ VXL đi nối tiếp qua các BGN, qua mạch xử lý ngắt và trở về lại VXL theo một mắt xích kín (daisy chain). BGN nào đặt gần VXL có mức ưu tiên cao

- Nếu một BGN đã ghi nhận ngắt, tín hiệu INTA bị chặn và không truyền tới BGN sau. Do đó, VXL không nhận được tín hiệu INTA, nghĩa là có một yêu cầu ngắt đã xảy ra.



Hình 4.1: Sơ đồ ưu tiên ngắt theo vị trí và vector ngắt

Phương pháp này tuy đơn giản nhưng thứ tự ưu tiên không thay đổi được và nếu có một BGN nào bị hỏng, mạch sẽ luôn báo có yêu cầu ngắt.

❖ *Ưu tiên ngắt dùng mạch so sánh*

Dùng các vi mạch Intel 8214, 8259 có thể thay đổi ưu tiên ngắt bằng chương trình.

4.2.3 Xác định nguồn gây ngắt

Có những phương pháp xác định TBNV nào gây ngắt để chuyển sang chương trình con phục vụ ngắt tưng ứng cho TBNV đó:

- Phương pháp hỏi vòng: VXL đọc và kiểm tra lần lượt trạng thái các TBNV. Nếu kết quả là 1, trạng thái TBNV là sẵn sàng và nguồn ngắt là TBNV đó.
- Phương pháp đọc vector ngắt tạo bởi tín hiệu ngắt INTA trong ưu tiên ngắt theo vị trí.
- Phương pháp đọc vào thanh ghi ưu tiên PR trong sơ đồ dùng mạch so sánh.

4.2.4 Tạo vector ngắt

Vector ngắt chứa địa chỉ đầu của chương trình con phục vụ ngắt. Các vector ngắt của MVT – PC được trình bày trên bảng 4.1. Ta có thể tạo vector ngắt bằng cổng AND colector hở như hình 4.1 (b).

Phương pháp này thay đổi được mức ưu tiên và cách tìm nguồn ngắt, nhưng mạch phức tạp và đắt tiền.

CHƯƠNG 5: GHÉP NỐI VÀO RA ĐIỀU KHIỂN BẰNG DMA

5.1 Khái niệm DMA cơ bản

5.1.1 Khái niệm DMA

Phương pháp trao đổi dữ liệu theo chương trình sẽ chậm do:

- VXL phải giải mã và thực hiện lệnh
- Trao đổi dữ liệu từng byte thông qua thanh chứa AX của VXL.

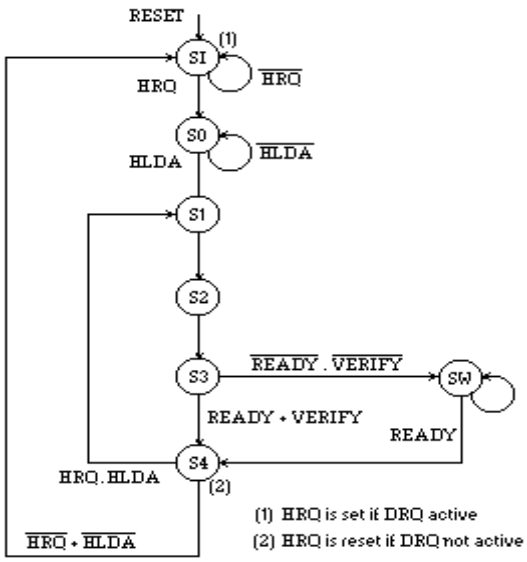
Kỹ thuật vào ra DMA (direct memory access) là phương pháp truy cập trực tiếp tới bộ nhớ hoặc I/O mà không có sự tham gia của CPU. Phương pháp này trao đổi dữ liệu giữa bộ nhớ và thiết bị ngoại vi với tốc độ cao và chỉ bị hạn chế bởi tốc độ của bộ nhớ hoặc của bộ điều khiển DMA. Tốc độ truyền DMA có thể đạt tới 10 ÷ 12 Kbyte/s với các bộ nhớ RAM có tốc độ cao. DMA được ứng dụng trong nhiều mục đích nhưng thông thường nó được dùng trong quá trình "refresh" DRAM, xuất màn hình, đọc ghi đĩa, truyền dữ liệu giữa các vùng nhớ với tốc độ cao .

5.1.2 Hoạt động DMAC (DMA Controller) cơ bản

Hai tín hiệu để yêu cầu và xác nhận trong hệ thống VXL là HOLD được sử dụng để yêu cầu DMA và HLDA là đầu ra xác nhận DMA. Khi tín hiệu HOLD hoạt động (HOLD = 1), DMA được yêu cầu. Bộ VXL trả lời bằng cách kích hoạt tín hiệu HLDA, xác nhận yêu cầu, đồng thời thả nổi các công việc hiện thời cùng các bus dữ liệu và địa chỉ, điều khiển được đặt ở trạng thái trở kháng cao. Trạng thái này cho phép các thiết bị I/O bên ngoài hoặc các bộ VXL khác nắm quyền điều khiển bus hệ thống để truy cập trực tiếp bộ nhớ . Tín hiệu HOLD có mức ưu tiên cao hơn INTR hoặc đầu vào NMI (ngắt không che được) và chỉ sau RESET. Tín hiệu HOLD luôn có hiệu lực tại bất kỳ thời điểm nào trong suốt quá trình thực hiện các lệnh khác của VXL. Chú ý rằng từ lúc tín hiệu HOLD thay đổi cho đến khi tín hiệu HLDA thay đổi đã trải qua một số chu kỳ clock .

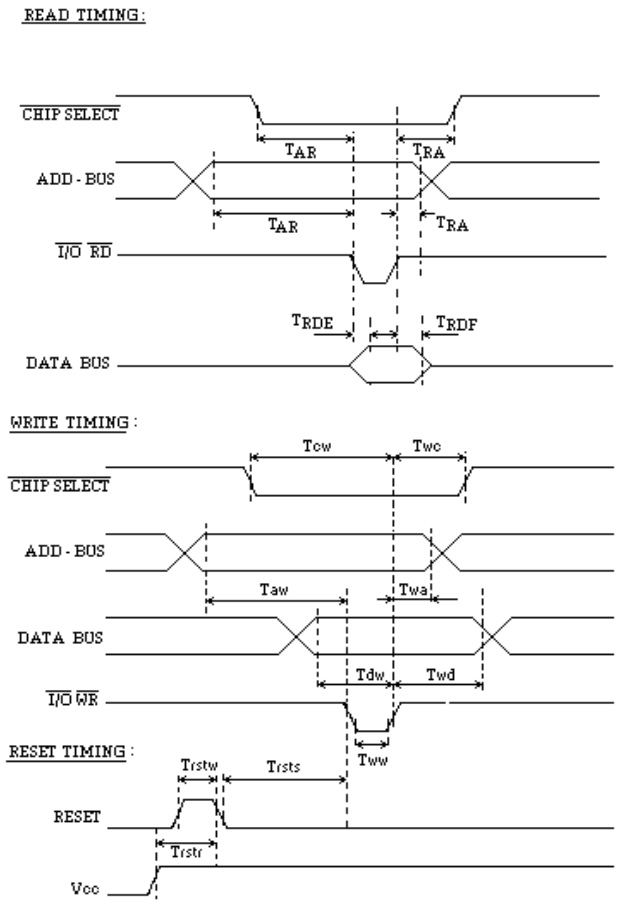
DMA thường được thực hiện giữa thiết bị I/O và bộ nhớ. Quá trình đọc DMA là quá trình đưa dữ liệu từ bộ nhớ ra thiết bị I/O, và ngược lại quá trình ghi DMA là quá trình đưa dữ liệu từ I/O tới bộ nhớ. Trong cả hai chu trình này thiết bị I/O và bộ nhớ được điều khiển đồng thời dẫn đến cần có các tín hiệu điều khiển khác nhau. Để điều khiển quá trình đọc DMA ta cần hai tín hiệu hoạt động MEMR (đọc bộ nhớ) và IOW (ghi I/O). Để điều khiển quá trình ghi ta có hai tín hiệu MEMW (ghi bộ bộ nhớ) và IOR (đọc I/O). Bộ điều khiển DMA cung cấp địa chỉ bộ nhớ và tín hiệu chọn thiết bị I/O cho 8088 trong suốt quá trình DMA. Do tốc độ truyền DMA phụ thuộc vào tốc độ của bộ nhớ và tốc độ của bộ điều khiển DMA nên trong trường hợp tốc độ của bộ điều khiển DMA nhỏ hơn so với bộ nhớ thì bộ điều khiển DMA sẽ làm giảm tốc độ chung của hệ thống .

Hình vẽ sau minh hoạ quá trình hoạt động DMA cơ bản cùng đồ thị thời gian đọc / ghi DMA :



DMA OPERATION STATE DIAGRAM

Đồ thị thời gian đọc / ghi DMA .



5.1.3. Các chế độ trao đổi dữ liệu DMA

+ Trao đổi dữ liệu theo khối: trao đổi nhiều lời dữ liệu, đếm từ giá trị n đến 0.

+ Trao đổi lấy lén chu kỳ từng phần: DMA phát hiện data bus rỗi, rồi thực hiện trao đổi dữ liệu DMA.

DMAC phải có:

- Thiết bị phát hiện đường dây rỗi
- Thiết bị bảo đảm VXL bị treo cho tới khi DMAC không sử dụng đường dây, khiến VXL chờ một thời gian Tw tới khi DMAC thực hiện trao đổi xong trong một phần của trao đổi dữ liệu và tiếp tục nốt ở phần lấy lén chu kỳ sau cho tới khi kết thúc trao đổi cả từ dữ liệu DMA.

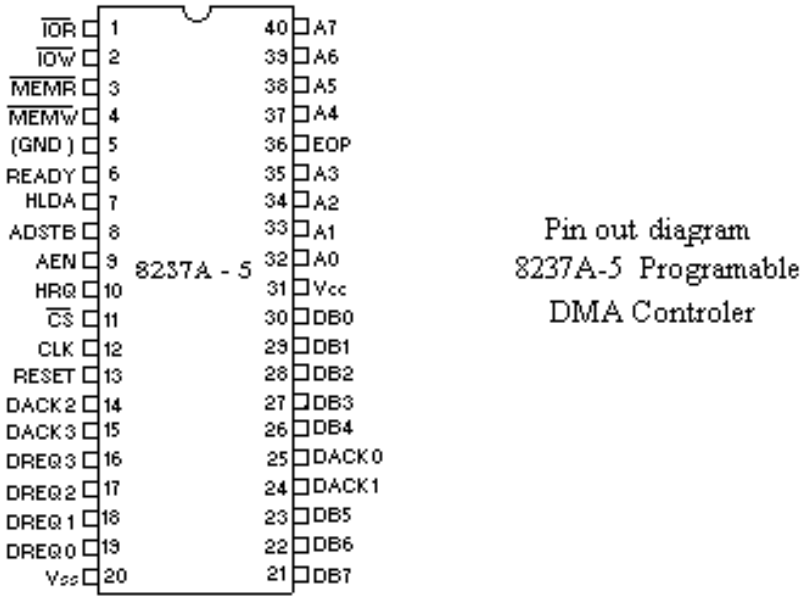
+ Trao đổi lấy lén chu kỳ trong suốt: giống chế độ trên, nhưng DMA bắt VXL chờ tới khi trao đổi xong một từ dữ liệu trọn vẹn.

5.2 Bộ điều khiển DMA – 8237A

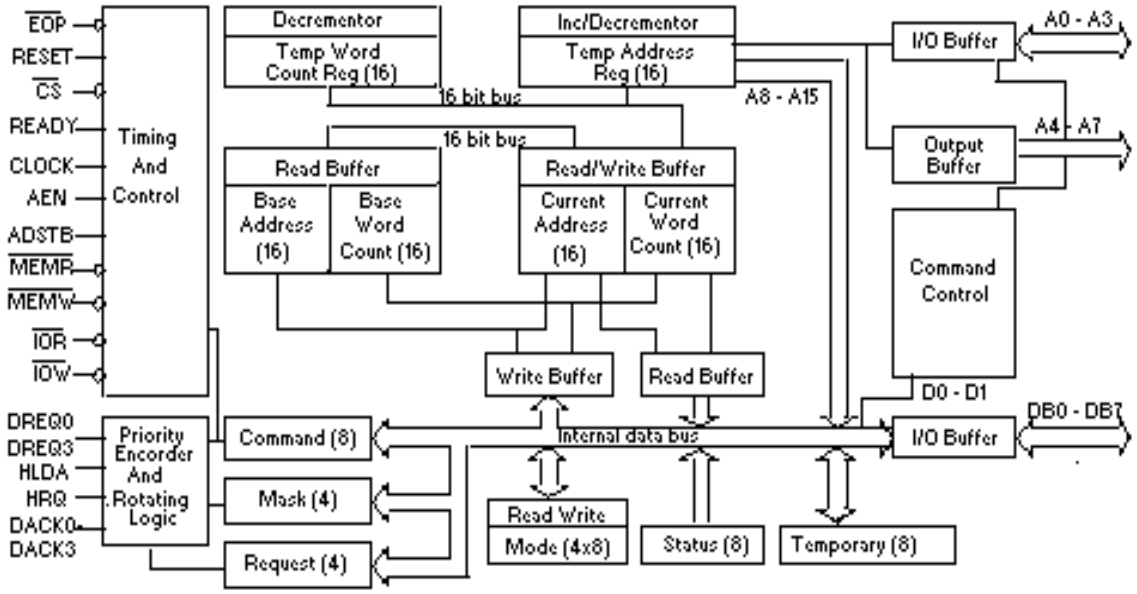
Vi mạch 8237A là một bộ điều khiển DMA cung cấp địa chỉ bộ nhớ và tín hiệu điều khiển trong suốt quá trình DMA và truyền số liệu với tốc độ cao giữa bộ nhớ và thiết bị vào ra. 8237 là vi mạch có 4 kênh tương thích với bộ VXL 8088, các kênh này có thể mở rộng thêm nhiều kênh khác mặc dù đối với hệ thống nhỏ thì 4 kênh này là qúa đủ. 8237 có khả năng thực hiện truyền DMA lên tới 1.6Mbyte/s .Mỗi kênh có khả năng đánh địa chỉ tới 64 Kbyte bộ nhớ và có thể truyền 64 Kbyte theo chương trình.

5.2.1 Sơ đồ khối và chức năng các khối của vi mạch 8237A

1. Sơ đồ khối



Hình 5.1: Vi mạch 8237A -5



Hình 5.2: Sơ đồ khối chi tiết của 8237

- CLK : Được nối với tín hiệu đồng hồ hệ thống .

- CS :Tín hiệu chọn mạch, thông thường nối với đầu ra của một bộ giải mã địa chỉ.

- RESET : Xoá các thanh ghi tạm, thanh ghi trạng thái , thanh ghi lệnh,thanh ghi yêu cầu.

- READY : tạo trạng thái chờ cho 8237A-5 đối với bộ nhớ hoặc I/O có tốc độ thấp.

- HLDA (hold Acknowledge): tín hiệu thông báo CPU đã thả nổi bus địa chỉ, dữ liệu, điều khiển của nó.

- $DREQ_0$ - $DREQ_3$ (DMA request) : Đầu vào yêu cầu truyền DMA cho mỗi kênh của 8237A-5, cực tính của các tín hiệu vào này là có thể lập trình được .

- DB_0 - DB_7 (data bus): Chân nối với bus dữ liệu của bộ VXL 8088 được sử dụng trong quá trình trao đổi dữ liệu

- IOR và IOW : điều khiển đọc và ghi I/O .

- EOP (end of process) : tín hiệu hai chiều báo kết thúc DMA tại cuối chu kỳ DMA .

- A_0- A_3 : Chọn thanh ghi nội cần lập trình và một phần địa chỉ truyền DMA trong quá trình DMA .

- A_7 - A_4 : Cung cấp 1 phần địa chỉ còn lại của DMA .

- HRQ (hold request) : nối với HOLD của VXL tạo yêu cầu DMA .

- $DACK_0$- $DACK_3$: (DMA channel acknowledge) Đầu ra chấp nhận DMA (có thể lập trình được) Thường dùng để chọn thiết bị cần điều khiển DMA.

- AEN : Cho phép chốt địa chỉ nối với DB_0 - DB_7 đồng thời không cho phép các bộ đệm trong hệ thống được nối với VXL.

- ADSTB : (address strobe) : hoạt động như ALE nhưng chỉ sử dụng bởi bộ điều khiển DMA chốt địa chỉ A_8- A_{15} trong suốt quá trình truyền DMA.

-MEMW ,MEMR: Đầu ra điều khiển bộ nhớ đọc, ghi dữ liệu trong chu kỳ DMA tương ứng .

2. Chức năng các khối và các thanh ghi

* Các khối chức năng của vi mạch :

- Khối đệm bus dữ liệu.

Đây là bộ đệm 8 bit ghép nối 8237 với bus hệ thống .

(D_0- D_7) : Các đường bus dữ liệu ba trạng thái, hai chiều. Khi 8237 đang được lập trình điều khiển bởi 8088 CPU, 8 bit dữ liệu của thanh ghi địa chỉ DMA, thanh ghi chế độ (Mode set Register) được gửi đến từ bus dữ liệu. Khi 8088 CPU đọc một thanh ghi địa chỉ DMA, hoặc thanh ghi đếm (terminal count Register), thanh ghi trạng thái dữ liệu được truyền đến 8088 thông qua bus dữ liệu. Trong suốt chu kỳ DMA (khi 8237 đang nắm quyền điều khiển bus dữ liệu), 8237 sẽ gửi 8 bit quan trọng nhất của địa chỉ bộ nhớ (từ một trong số các thanh ghi địa chỉ DMA) tới bộ điều khiển bus. Các bit địa chỉ này sẽ được truyền ngay khi bắt đầu chu kỳ DMA, sau đó bus dữ liệu sẽ được giải phóng để thực hiện quá trình trao đổi dữ liệu với bộ nhớ trong suốt quá trình DMA.

* Các kênh DMA (DMA channel):

Mỗi kênh (CH0 - CH3) bao gồm hai thanh ghi 16 bit :

Thanh ghi địa chỉ DMA và thanh ghi đếm (Terminal Count Register) .

Hai thanh ghi này phải được khởi tạo trước khi các kênh được phép hoạt động. Thanh ghi địa chỉ DMA được đọc cùng với địa chỉ của ô nhớ đầu tiên được truy cập. Giá trị được ghi vào thanh ghi đếm (termianal count register) xác định số các chu kỳ DMA ngay trước khi đầu ra của bộ đếm kết thúc TC (Terminal Count) hoạt động. Nói chung khi muốn có N chu kỳ DMA thì giá trị N-1 phải được đưa tới 14 bit thấp của thanh ghi đếm cổng vào ra (Terminal Count Register). Hai bit cao nhất của thanh ghi này xác định loại hoạt động DMA cho kênh này .

* Khối đọc ghi logic (Read/ Write logic)

Khi 8088 CPU đọc hoặc ghi một trong số các thanh ghi của 8237, khối logic điều khiển đọc ghi nhận lệnh đọc I/O (IOR) hoặc I/O Write (IOW) thông báo cho vi mạch điều khiển bus 8288 giải mã 4 bit địa chỉ thấp nhất (A_0- A_3) và đồng thời ghi nội dung của bus dữ liệu vào thanh ghi địa chỉ (nếu là tín hiệu IOW) hoặc ghi nội dung thanh ghi địa chỉ lên bus dữ liệu (nếu là tín hiệu IOR). Trong suốt quá trình DMA khối logic điều khiển đọc ghi thực hiện lệnh đọc I/O và ghi bộ nhớ (chu kỳ DMA ghi) hoặc tín hiệu ghi I/O và đọc bộ nhớ (chu kỳ DMA đọc). Cần lưu ý rằng trong suốt quá trình điều khiển DMA thiết bị I/O không hoạt động trong chế độ này phải được đặt ở trạng thái "cấm " sử dụng tín hiệu AEN.

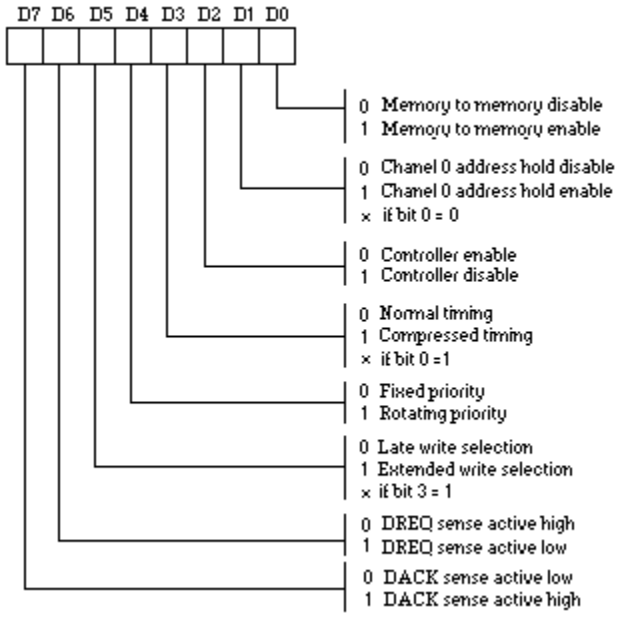
* Các thanh ghi nội:

- Thanh ghi địa chỉ hiện thời (CAR): Lưu 16 bít địa chỉ bộ nhớ trong suốt quá trình truyền DMA. Mỗi kênh đều có một thanh ghi này và CAR tăng hoặc giảm tuỳ thuộc vào cách lập trình.

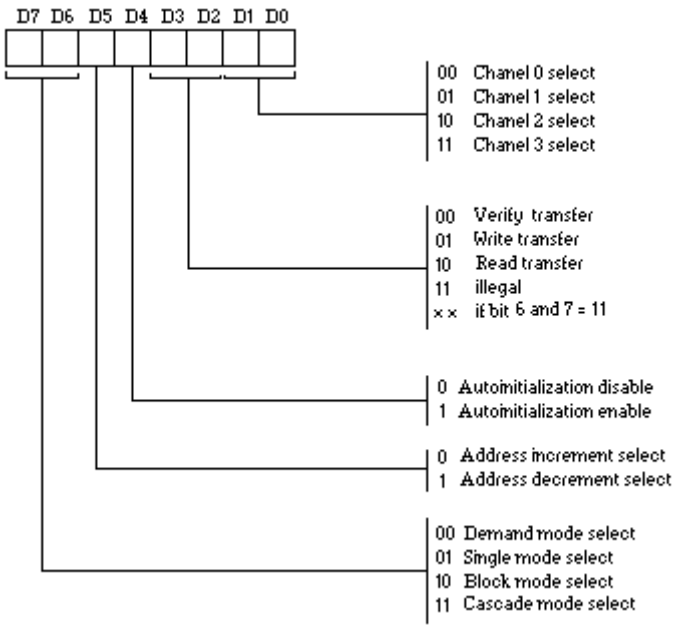
- Thanh ghi đếm từ hiện thời (CWCR): điều khiển số byte được truyền trong quá trình DMA . Số ghi trong thanh ghi này nhỏ hơn số byte đã được truyền 1 đơn vị.

- Thanh ghi địa chỉ cơ sở và từ cơ sở (BWCR): sử dụng khi chọn chế độ tự động kích hoạt cho một kênh ,trong quá trình này thanh ghi BWCR được dùng để gọi lại các thanh ghi CAR và CWCR sau khi DMA kết thúc.

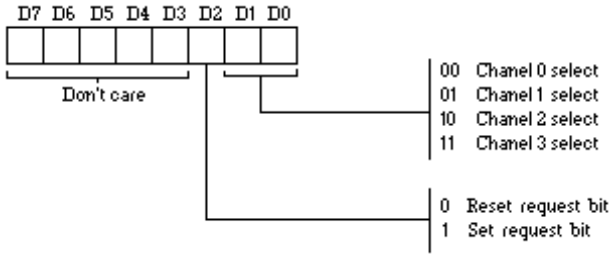
- Thanh ghi lệnh (Command Register): lập trình cho hoạt động của 8237



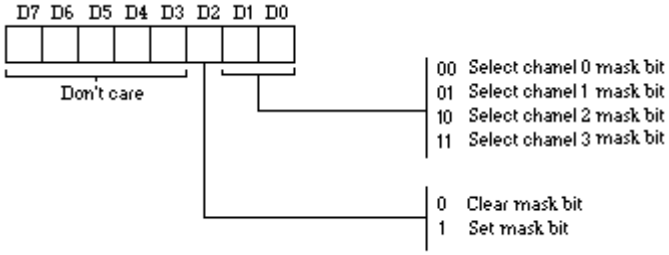
- Thanh ghi chế độ (Mode Registe): Thiết lập chế độ hoạt động cho mỗi kênh .



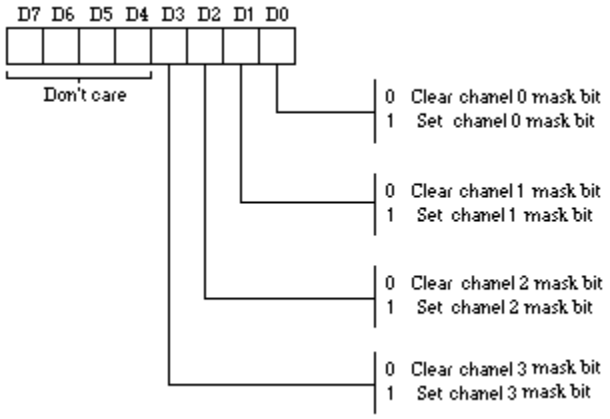
- Thanh ghi yêu cầu (Request Register): dùng yêu cầu truyền DMA bằng phần mềm .



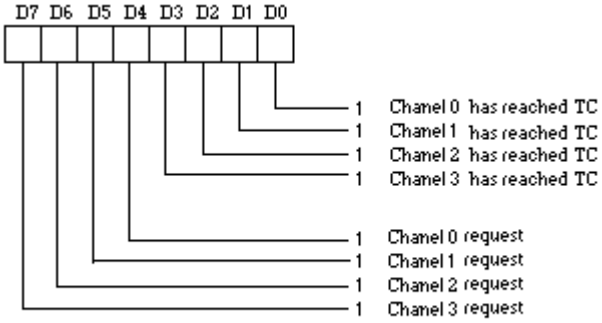
- Thanh ghi che set/reset (Mask register set/reset mode): xoá ,thiết lập việc cấm các kênh .



- Thanh ghi cấm (Mask register): xoá hoặc thiết lập việc cấm của tất cả các kênh bằng một lệnh .



- Thanh ghi trạng thái (Status register -SR): xác định trạng thái của các kênh DMA .



Bảng 5.1: Giới thiệu địa chỉ và các lệnh đọc ghi cho các thanh ghi.

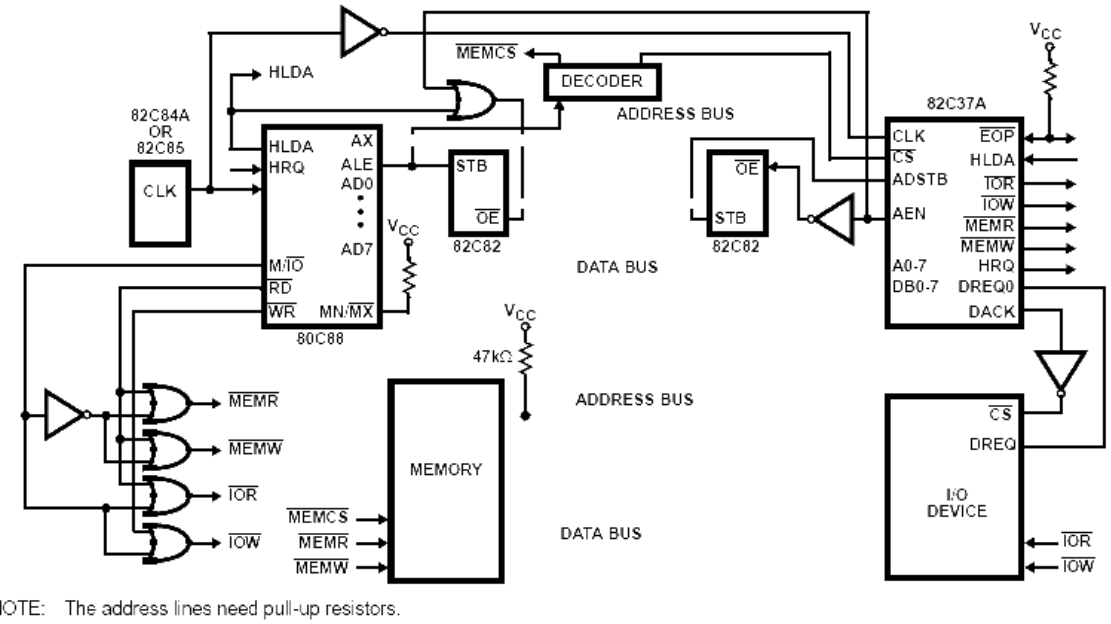
A_3	A_2	A_1	A_0	$\overline{IO}$R	$\overline{IO}$W	Hoạt động
1	0	0	0	0	1	Đọc thanh ghi trạng thái
1	0	0	0	1	0	Ghi thanh ghi lệnh
1	0	0	1	1	0	Ghi thanh ghi yêu cầu
1	0	1	0	1	0	Ghi thanh ghi mặt nạ đơn
1	0	1	1	1	0	Ghi thanh ghi chế độ
1	1	0	0	1	0	Xoá flip-flop con trỏ byte
1	1	0	1	0	1	Đọc thanh ghi tạm thời
1	1	0	1	1	0	Xoá chủ
1	1	1	0	1	0	Xoá thanh ghi mặt nạ
1	1	1	1	1	0	Ghi tất cả các bit của thanh ghi mặt nạ

Kênh	**Thanh ghi**	**Hoạt động**	$\overline{CS}$	$\overline{IOR}$	$\overline{IOW}$	**A_3**	**A_2**	**A_1**	**A_0**	**Int FF**	**Data Bus**
0	địa chỉ cơ sở	Ghi	0	1	0	0	0	0	0	0	$A_7 \div A_0$
	và hiện hành		0	1	0	0	0	0	0	1	$A_8 \div A_{15}$
	địa chỉ hiện	Đọc	0	0	1	0	0	0	0	0	$A_7 \div A_0$
	hành		0	0	1	0	0	0	0	1	$A_8 \div A_{15}$
	Đếm lời cơ sở	Ghi	0	1	0	0	0	0	1	0	$W_7 \div W_0$
	và hiện hành		0	1	0	0	0	0	1	1	$W_{15} \div W_8$
	Đếm lời hiện	Đọc	0	0	1	0	0	0	1	0	$W_7 \div W_0$
	hành		0	0	1	0	0	0	1	1	$W_{15} \div W_8$
1	địa chỉ cơ sở	Ghi	0	1	0	0	0	1	0	0	$A_7 \div A_0$
	và hiện hành		0	1	0	0	0	1	0	1	$A_8 \div A_{15}$
	địa chỉ hiện	Đọc	0	0	1	0	0	1	0	0	$A_7 \div A_0$
	hành		0	0	1	0	0	1	0	1	$A_8 \div A_{15}$
	Đếm lời cơ sở	Ghi	0	1	0	0	0	1	1	0	$W_7 \div W_0$
	và hiện hành		0	1	0	0	0	1	1	1	$W_{15} \div W_8$
	Đếm lời hiện	Đọc	0	0	1	0	0	1	1	0	$W_7 \div W_0$
	hành		0	0	1	0	0	1	1	1	$W_{15} \div W_8$
2	địa chỉ cơ sở	Ghi	0	1	0	0	1	0	0	0	$A_7 \div A_0$
	và hiện hành		0	1	0	0	1	0	0	1	$A_8 \div A_{15}$
	địa chỉ hiện	Đọc	0	0	1	0	1	0	0	0	$A_7 \div A_0$
	hành		0	0	1	0	1	0	0	1	$A_8 \div A_{15}$
	Đếm lời cơ sở	Ghi	0	1	0	0	1	0	1	0	$W_7 \div W_0$
	và hiện hành		0	1	0	0	1	0	1	1	$W_{15} \div W_8$
	Đếm lời hiện	Đọc	0	0	1	0	1	0	1	0	$W_7 \div W_0$
	hành		0	0	1	0	1	0	1	1	$W_{15} \div W_8$
3	địa chỉ cơ sở	Ghi	0	1	0	0	1	1	0	0	$A_7 \div A_0$
	và hiện hành		0	1	0	0	1	1	0	1	$A_8 \div A_{15}$
	địa chỉ hiện	Đọc	0	0	1	0	1	1	0	0	$A_7 \div A_0$
	hành		0	0	1	0	1	1	0	1	$A_8 \div A_{15}$
	Đếm lời cơ sở	Ghi	0	1	0	0	1	1	1	0	$W_7 \div W_0$
	và hiện hành		0	1	0	0	1	1	1	1	$W_{15} \div W_8$
	Đếm lời hiện	Đọc	0	0	1	0	1	1	1	0	$W_7 \div W_0$
	hành		0	0	1	0	1	1	1	1	$W_{15} \div W_8$

5.2.2 Ghép nối với 8237A-5 trong hệ vi xử lý 8088

Vi mạch 8237 được sử dụng trong máy vi tính IBM PC để thực hiện các quá trình DMA trao đổi dữ liệu trực tiếp giữa thiết bị ngoài và máy vi tính .

Tín hiệu AEN và ADSTB ra của 8237A-5 điều khiển đầu ra của chốt 82C82. Trong suốt quá trình hoạt động bình thường của 8088 (AEN=0) các vi mạch chốt địa chỉ cung cấp các bit địa chỉ A_{15}- A_8 .Địa chỉ bus A_7- A_0 được cung cấp trực tiếp bởi 8237A-5 và chứa một phần địa chỉ truyền DMA. Tín hiệu điều khiển MEMR ,MEMW ,IOR ,IOW cũng được tạo bởi bộ điều khiển DMA. Giả thiết địa chỉ đầu cho DMAC là 0070h.



NOTE: The address lines need pull-up resistors.

Hình 5.3: Sơ đồ ghép nối 8237 với 8088

1. *Các trạng thái hoạt động*

Vi mạch 8237A có các trạng thái sau:

- Trạng thái nghỉ SI: trạng thái trở kháng cao, không được chọn mạch ($\overline{CS}$ =1).
- Trạng thái chờ SO: Trạng thái đầu của quá trình trao đổi DMA. 8257 nhận tín hiệu DREQ, xử lý và gởi yêu cầu Hold cho VXL nhưng chưa nhận được tín hiệu xác nhận theo HLDA. Khi nhận được xác nhận, DMAC bắt đầu quá trình trao đổi dữ liệu.
- Trạng thái phát địa chỉ nguồn số liệu S1:
  DMAC phát địa chỉ nguồn số liệu (cửa I/O hoặc Memory)
- Trạng thái phát lệnh đọc cho nguồn số liệu S2:
  DMAC phát lệnh đọc số liệu của nguồn số liệu
- Trạng thái chờ SW:
  DMAC chờ trong thời gian Sw để số liệu được đọc xong ở nguồn phát số liệu, ghi xong vào thanh ghi đệm, chờ Ready của DMAC.

- Trạng thái phát địa chỉ của đích số liệu S3:
  DMAC phát địa chỉ của nơi gởi số liệu (Memory hoặc cửa I/O)
- Trạng thái phát lệnh ghi số liệu vào đích S4:
  DMAC phát lệnh ghi số liệu từ thanh ghi đệm ra cửa vào-ra

2. Các chế độ

Tuỳ nội dung thanh ghi chế độ lúc khởi phát, DMAC có các chế độ sau:
- Chế độ chuyển từng từ đơn (Single Transfer Mode)
- Chế độ chuyển nhiều từ (Block Transfer Mode)
- Chế độ chuyển theo nhu cầu (Demand Transfer Mode) với số lượng từ tuỳ theo người sử dụng điều khiển.
- Chế độ ghép tầng (Cascade Mode) là chế độ chuyển liên tiếp các từ, điều khiển bởi các DMAC mắc nối tầng.

3. Các chế độ ưu tiên của yêu cầu DMAC

Cũng giống như xử lý ngắt, DAMC có hai chế độ ưu tiên:

- Chế độ ưu tiên cố định: kênh 0 có mức ưu tiên cao nhất, kênh 3 có mức ưu tiên thấp nhất.
- Chế độ ưu tiên vòng: kênh vừa được phục vụ sẽ có mức ưu tiên thấp nhất. Khi mới lập chế độ, kênh 0 có mức ưu tiên cao nhất.

4. Khởi phát và đảm bảo chương trình phục vụ

Trước khi đưa DMAC vào hoạt động, chương trình phải khởi phát nó. Quá trình khởi phát gồm 2 giai đoạn:

- Xoá cứng bằng tín hiệu Reset, làm các thanh ghi về 0.
- Ghi các lệnh của chương trình khởi phát: VXL ghi vào các thanh ghi chức năng như địa chỉ ban đầu, số từ trao đổi, số kênh, che chắn yêu cầu của kênh không cần hoạt động, chế độ trao đổi, chiều tăng hay giảm địa chỉ… Khi ghi dữ liệu vào thanh ghi 16 bit, phải ghi lần lượt 2 byte theo thứ tự byte thấp trước, byte cao sau.
- Sau khi được khởi phát, DMAC sẽ hoàn toàn tự động làm việc. Kết thúc chương trình khởi phát này, luôn phải có lệnh ghi bit D_4=1 vào thanh ghi chế độ để cho phép tự khởi động.

5. Kết thúc trao đổi DMA

Có hai cách kết thúc trao đổi dữ liệu:
- Kết thúc tự động: Kết thúc khi bộ đếm số tin trao đổi đếm về 0, hay số đếm kết thúc TC (Terminal Count). Với DMAC 8237, việc kết thúc thể hiện ở tín hiệu Hold kết thúc, tức trở về mức logic 0.

- Kết thúc cữơng bức: đây là tín hiệu từ ngoài vào (do mạch TBNV hoặc do người điều hành can thiếp bằng cách ấn nút lệnh) để kết thúc quá trình trao đổi DMA.

6. Kiểm tra sự kết thúc ở kênh mong muốn

Trong khi DMAC hoạt động, thanh ghi trạng thái luôn được cập nhật, tức luôn ghi trạng thái hiện thời của DMAC. Khi kết thúc, VXL có thể đọc nội dung này của thanh ghi trạng thái để biết rằng kênh nào của DMA đã đạt đến số đếm kết thúc TC, số lời đã trao đổi là bao nhiêu (trường hợp kết thúc cưỡng bức do $\overline{EOP}$ từ ngoài vào) và kênh vào đang chờ phục vụ.

7. Đặc điểm về hoạt động của 8237

8237 có những đặc điểm chính sau:
- Cho phép hoặc cấm riêng biệt từng kênh DMA trong số 4 kênh.
- Trao đổi dữ liệu DMA trực tiếp giữa hai vùng bộ nhớ.
- Cho phép nối tầng các DMAC để tăng số kênh DMA.
- Có thể tạo yêu cầu DMA bằng chương trình
- Với 8237-2, tốc độ trao đổi DMA có thể đạt tới 1,6Mbps. Do đó ngoài phục vụ trao đổi dữ liệu giữa MVT và màn hình, đĩa, DMA còn được dùng trong mạng MVT.

5.2.3 Lập trình cho 8237A

a. Chương trình truyền dữ liệu giữa bộ nhớ-bộ nhớ bằng DMA

Ví dụ cần chuyển khối nhớ có địa chỉ 1000h-13FFh tới khối nhớ 1400h-17FFh thông qua DMA. Chương trình sẽ gồm các bước sau:

- Xoá flip-flop con trỏ byte về 0.
- Lập trình cho các kênh 0 và 1: Đưa địa chỉ nguồn và đích vào kênh 0 và 1.
- Đặt giá trị bộ đếm bằng số byte cần chuyển -1
- Lập trình thông qua thanh ghi chế độ cho kênh 0 và 1
- Lập trình cho thanh ghi lệnh
- Cho phép kênh 0 và đưa ra yêu cầu DMA
- Kiểm tra bit hết dữ liệu. Nếu đã xong thì kết thúc DMA.

Xác định địa chỉ các thanh ghi:

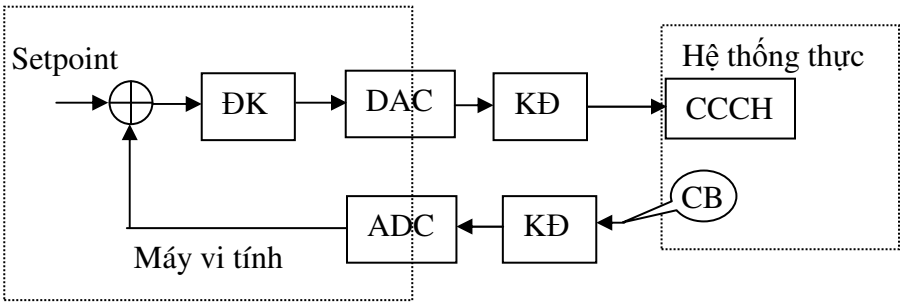
```
Addr_chanel0 = $70;
Addr_chanel1 =$72;
Addr_CT1   =$73  ;// Địa chỉ bộ đếm kênh 1
Addr_SR   =$78   ;// Địa chỉ thanh ghi trạng thái
Addr_CR   =$78  ;// địa chỉ thanh ghi điều khiển
Addr_MR  =$7B ; //địa chỉ thanh ghi chế độ
Addr_MKR = $7F;// địa chỉ thanh ghi cấm
Addr_RR  = $79 ; //địa chỉ thanh ghi yêu cầu
Addr_pFF = $7C ; //địa chỉ flip-flop con trỏ byte
```

```
port[addr_pFF]=$00;// xoá con trỏ FF về 0.
// Đặt địa chỉ nguồn vào kênh 0
port[addr_chanel0] =$00;
port[addr_chanel0] =$10;
// Đặt địa chỉ đích vào kênh 1
port[addr_pFF]=$01;// xoá con trỏ FF về 0.
port[addr_chanel1] =$00;
port[addr_chanel1] =$14;
//Chỉ định số byte cần chuyển 4000
port[addr_pFF]=$01;// xoá con trỏ FF về 0.
port[addr_CT1] =$00;
port[addr_CT1] =$40;
// Đặt chế độ: kênh 0 đọc  1011 1000b= B8h ; kênh 1 ghi: 1011 0101=B5h
port[addr_pFF]=$01;// xoá con trỏ FF về 0.
port[addr_MR]=$B8;
port[addr_MR]=$B5;
// Lập trình thanh ghi lệnh cho phép truyền M-M= 01h
Port[addr_CR]=$01;
// Bỏ mặt nạ che chắn kênh 0 = 0000 1110 b=0Eh
port[addr_MKR]= $0E;
//Yêu cầu DMAC = 0000 0100 b=04h
port[addr_RR] = $04;
//Kiểm tra trạng thái kết thúc đếm: kênh 0 đã truyền hết dữ liệu (D0=1) =01h
Repeat
Var1=port[addr_SR];
Until ((var1 and $01)=$01);
//kết thúc truyền
Return;
```

Sau khi khởi động, hệ thống DMAC sẽ tự hoạt động mà không cần thêm chương trình phụ nào nữa.

CHƯƠNG 6: GHÉP NỐI TƯƠNG TỰ-SỐ VÀ SỐ-TƯƠNG TỰ

Để có thể điều khiển được các máy móc trong nhà máy, các thiết bị y tế chúng ta cần quan tâm đến các ảnh hưởng của môi trường bên ngoài như áp suất, nhiệt độ ... Thường có một số bước để nhận được tín hiệu điện biểu diễn cho các yếu tố đó và biến đổi từ dạng tương tự sang dạng số để các máy tính có thể xử lý được. Hình 6.1 biểu diễn sơ đồ đơn giản của một hệ thống điều khiển tự động dùng máy vi tính.



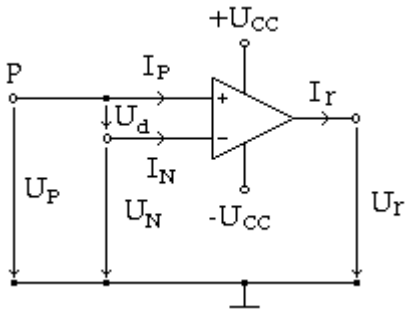
Hình 6.1: Sơ đồ cấu trúc hệ thống điều khiển tự động dùng máy vi tính

Đầu tiên là *cảm biến (CB)* dùng để biến đổi các giá trị vật lý như nhiệt độ, áp suất ... sang một điện áp hay dòng tương ứng. Sau cảm biến thường là bộ *khuếch đại* (KĐ) và lọc. Khâu này được thực hiện bởi một số mạch khuếch đại thuật toán. Sau đó, tín hiệu tương tự này được chuyển sang dạng số bằng bộ *biến đổi tương tự-số* (Analog-Digital Converter-ADC), và được máy vi tính nhận, xử lý, lưu trữ, so sánh với tín hiệu đặt, và tạo ra tín hiệu *điều khiển* (ĐK) tương ứng. Vì máy vi tính chỉ làm việc ở tín hiệu số (logic 0 và 1) nên cần có bộ *chuyển đổi số-tương tự* (Digital-Analog Converter-DAC) để tạo ra tín hiệu điều khiển ở dạng tương tự. Tiếp theo là khâu *khuếch đại* (KĐ) và lệch áp để tạo tín hiệu phù hợp cung cấp cho *cơ cấu chấp hành* (CCCH) của hệ thống thực.

6.1 Giới thiệu một số mạch khuếch đại thuật toán

Khuếch đại thuật toán là một trong số những linh kiện điện tử thường gặp nhất trong kỹ thuật tương tự, vì thế trong kỹ thuật đo lường và điều khiển công nghiệp, khuếch đại thuật toán cũng có mặt trong rất nhiều thiết bị và hệ thống. Khả năng sử dụng của các bộ khuếch đại thuật toán là rất vạn năng, chúng được áp dụng trong nhiều lĩnh vực như các bộ khuếch đại một chiều, các bộ khuếch đại xoay chiều, bộ lọc tích cực, bộ dao động, bộ biến đổi trở kháng, bộ vi phân, bộ tích phân...

6.1.1 Các tham số cơ bản của mạch khuếch đại thuật toán



Hình 6.2: Sơ đồ bộ khuếch đại thuật toán.

- U_d hiệu điện áp vào
- U_P , I_P điện áp vào và dòng điện vào cửa thuận.
- U_N , I_N điện áp vào và dòng điện vào cửa đảo.
- U_r , I_r điện áp ra và dòng điện ra.

Bộ khuếch đại thuật toán khuếch đại hiệu điện áp $U_d = U_P - U_N$ với hệ số khuếch đại $K_0 > 0$. Do đó điện áp ra:

$$U_r = K_0 U_d = K_0 (U_p - U_N)$$

1. Hệ số khuếch đại hiệu K_0

Khi không tải được xác định theo biểu thức sau

$$K_0 = \frac{U_r}{U_d} = \frac{U_r}{U_p - U_N}$$

2. Hệ số khuếch đại đồng pha K_{CM}

Nếu đặt vào cửa thuận và cửa đảo của bộ khuếch đại thuật toán các điện áp bằng nhau, nghĩa là:

$$U_P = U_N = U_{CM} = 0$$

thì $U_d = 0$. Gọi U_{CM} là điện áp vào đồng pha. Lý tưởng ta có $U_r=0$. Tuy nhiên, thực tế không phải như vậy. Do đó giữa điện áp ra và điện áp vào đồng pha có quan hệ tỷ lệ là hệ số khuếch đại đồng pha K_{CM} :

$$K_{CM} = \frac{\Delta U_r}{\Delta U_{CM}}$$

K_{CM} nói chung phụ thuộc vào mức điện áp vào đồng pha.

3. Hệ số nén đồng pha CMRR

Dùng để đánh giá khả năng làm việc của bộ khuếch đại thực so với bộ khuếch đại lý tưởng (K_{CM}=0)

$$CMRR = \frac{K_0}{K_{CM}}$$

4. Dòng vào tĩnh

Là trị trung bình của dòng vào cửa thuận và dòng vào cửa đảo:

$$I_t = \frac{I_p + I_N}{2} \textbf{ với } U_P = U_N = 0$$

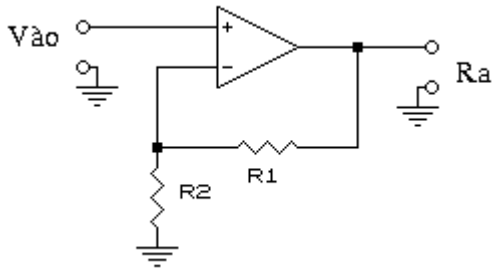
Dòng vào lệch không là hiệu các dòng vào tĩnh ở hai cửa của bộ khuếch đại thuật toán

$$I_0 = I_P - I_N \textbf{ với } U_P = U_N = 0$$

Thông thường $I_0 = 0{,}1I_P$. Trị số của dòng vào lệch không thay đổi theo nhiệt độ. Hiện tượng này gọi là hiện tượng trôi dòng lệch không.

6.1.2 Các sơ đồ cơ bản của bộ khuếch đại thuật toán

1. Sơ đồ khuếch đại không đảo



Hình 6.3: Sơ đồ mạch khuếch đại không đảo

Hệ số khuếch đại của mạch:

$$K = \frac{1}{\left(\frac{1}{K_0} + \frac{R_2}{R_1 + R_2}\right)\left(1 - \frac{1}{CMRR}\right)}$$

Một bộ khuếch đại thuật toán lý tưởng sẽ có:

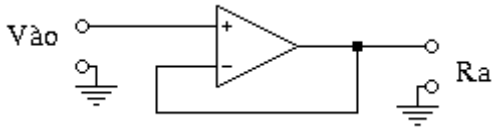
- Hệ số khuếch đại khi không có phản hồi âm lớn vô cùng.
- Điện trở lối vào lớn vô cùng.
- Độ rộng dải thông lớn vô cùng.
- Hệ số nén đồng pha CMRR lớn vô cùng.
- Điện trở lối ra bằng không .
- Thời gian đáp ứng bằng không

Do đó, hệ số khuếch đại của mạch lý tưởng là:

$$K = \frac{R_1 + R_2}{R_2}$$

2. Mạch đệm

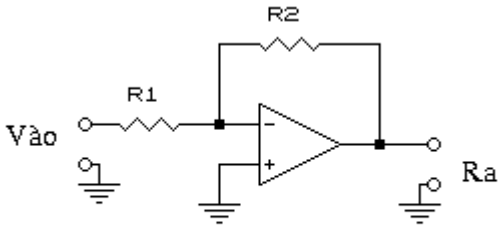
Đây là trường hợp đặc biệt của mạch khuếch đại không đảo



Hình 6.4: Sơ đồ mạch đệm

Mạch có hệ số khuếch đại bằng K=1 và thường dùng để phối hợp trở kháng.

3. Mạch khuếch đại đảo

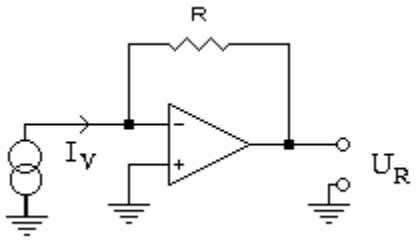


Hình 6.5: Sơ đồ mạch khuếch đại đảo

Hệ số khuếch đại của mạch:

$$K = -\frac{R_2}{R_1}$$

4. Sơ đồ biến đổi dòng điện - điện áp



Hình 6.6: Sơ đồ biến đổi dòng điện - điện áp

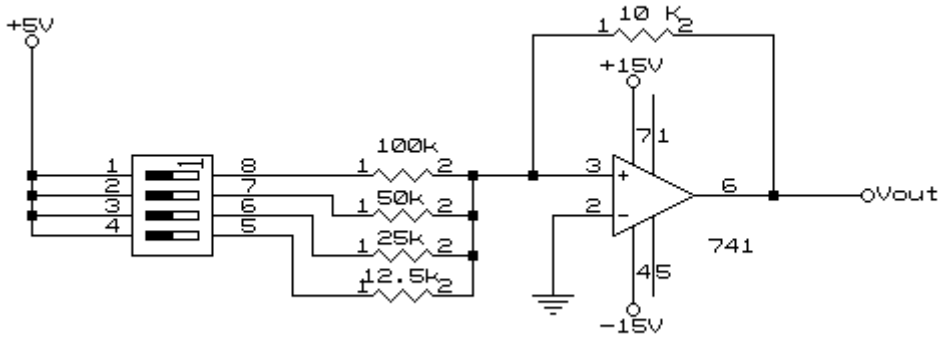
Điện áp ra được tính theo biểu thức:

$$U_R = - R.I_V$$

6.2 Bộ chuyển đổi số-tương tự DAC

Bộ chuyển đổi số tương tự dùng để chuyển tín hiệu ở dạng số sang dạng tương tự. Khâu này hay được sử dụng trong hệ thống điều khiển có dùng máy vi tính. Vì máy vi tính chỉ làm việc với tín hiệu số, trong khi hệ thống thực cần tín hiệu điều khiển ở dạng tương tự.

6.2.1 Hoạt động của DAC

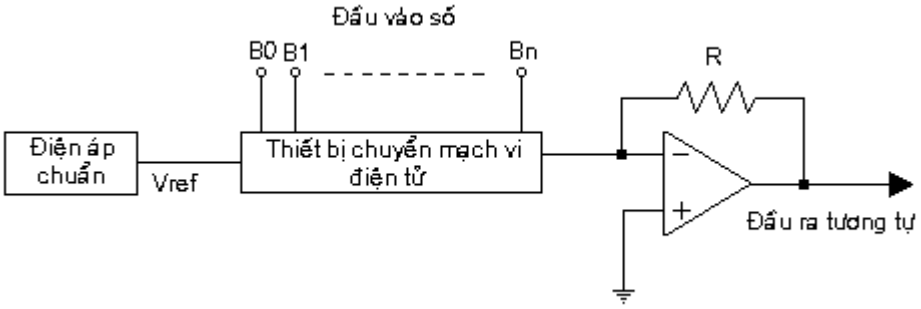


Hình 6.7: Ví dụ về bộ biến đổi DA 4 bít

Mục đích của bộ biến đổi DA, như đã nêu, là biến đổi tín hiệu nhị phân *n* bít thành dòng hay áp tương ứng. Hình 6.7 là một ví dụ về một bộ biến đổi DA 4 bít đơn giản. Dựa vào vị trí của các công tắt B_i (1 là đóng và 0 là mở), điện trở đầu vào sẽ thay đổi theo. Do đó điện áp đầu ra cũng sẽ thay đổi theo :

$$U_r = 10^4.5\left(\frac{B_1}{10^5} + \frac{B_2}{5.10^4} + \frac{B_3}{2{,}5.10^4} + \frac{B_4}{12{,}5.10^3}\right)$$

Về nguyên tắc bộ chuyển đổi số-tương tự tiếp nhận một mã số *n* bít song song hoặc nối tiếp ở lối vào và biến đổi ra dòng điện hoặc điện áp tương ứng ở lối ra. Dòng điện hay điện áp ở lối ra là hàm biến thiên phù hợp theo mã số ở lối vào.



Hình 6.8: Sơ đồ khối DAC

Một bộ DAC hoàn chỉnh bao gồm ba phần tử cơ bản:

- Điện áp quy chiếu ổn định bên ngoài (V_{ref})
- DAC cơ sở
- Khuếch đại thuật toán

Như vậy điện áp đầu ra của bộ biến đổi V_0 sẽ phụ thuộc vào mã nhị phân đầu vào theo công thức sau:

$$\mathbf{V_0 = V_{ref}\,(B_0 2^0 + B_1 2^1 + ... + B_n 2^n)/2^{n+1}}$$

Trong đó B_0 là bít thấp nhất và B_n là bít cao nhất của mã nhị phân đầu vào, V_{ref} là điện áp quy chiếu.

DAC cơ sở cấu tạo bằng các điện trở chính xác và những chuyển mạch tương tự được điều khiển bởi mã số đầu vào . Các chuyển mạch tương tự điều chỉnh dòng điện hay điện áp trích ra từ điện áp quy chiếu và tạo nên dòng điện hay điện áp ở đầu ra tương ứng với mã số đầu vào. Mạch khuếch đại thuật toán dùng ở đây để chuyển đổi dòng thành áp đồng thời có chức năng tầng đệm.

Bộ biến đổi DAC có đặc điểm là đại lượng ra tương tự không liên tục, độ rời rạc của đầu ra phụ thuộc vào số bít của bộ biến đổi, những DAC có số bít đầu vào lớn thì tổng số nấc điện áp ra càng lớn và khoảng cách giữa các nấc càng nhỏ.

6.2.2 Các tham số cơ bản của bộ chuyển đổi DA

- Độ phân giải (Solution): Liên quan đến số bít của một DAC. Nếu số bít là *n* thì số trạng thái của tín hiệu nhị phân là 2^n nghĩa là sẽ có 2^n mức điện thế (hoặc dòng điện) khác nhau, do đó có độ phân giải là $1/2^n$. Độ phân giải càng bé thì điện thế (hoặc dòng điện đầu ra) càng có dạng liên tục, càng gần với thực tế và ngược lại.
- Độ chính xác (Accuracy): Có thể đánh giá chất lượng của một DAC bằng sai số của nó. Đại lượng biểu diễn sai số là độ lệch tối đa giữa đại lượng ra và một đường thẳng nối điểm 0 với điểm FS (Full Scale) trên đặc tuyến chuyển đổi DA.
- Độ tuyến tính (Linearity): Độ tuyến tính của DAC cho biết độ lệch điện áp so với một đường thẳng đi qua những điểm nút của đặc tuyến chuyển đổi. Đó là đặc tính thường gặp nhất với DAC. Đường cong đặc tuyến là đơn điệu nếu sự thay đổi độ lệch trên là không đổi dấu. Để có một DAC đơn điệu, độ lệch này phải lớn hơn 0 cho mỗi nấc thang. Ngoài ra mức độ tuyến tính của DAC phải nhỏ hơn hoặc bằng 1/2 LSB (Least Significant Bit- Tham khảo hình 6.14) để nó trở nên đơn điệu. Như vậy 1/2 LSB là đặc trưng về giới hạn đơn điệu của một DAC.
- Phi tuyến vi sai: là đại lượng cho biết độ lệch giữa giá trị thực tế và lý tưởng cho một nấc điện áp ra ứng với mỗi thay đổi của mã số vào. Đại lượng này cho biết về độ nhẵn của đường cong đặc tuyến đối với DAC.
- Thời gian thiết lập: đối với một DAC là thời gian cần thiết để điện áp ra đạt tới giá trị tới hạn sai số xung quanh giá trị ổn định. Giới hạn này thường là 1/2 LSB hoặc biểu diễn bằng giá trị FS.

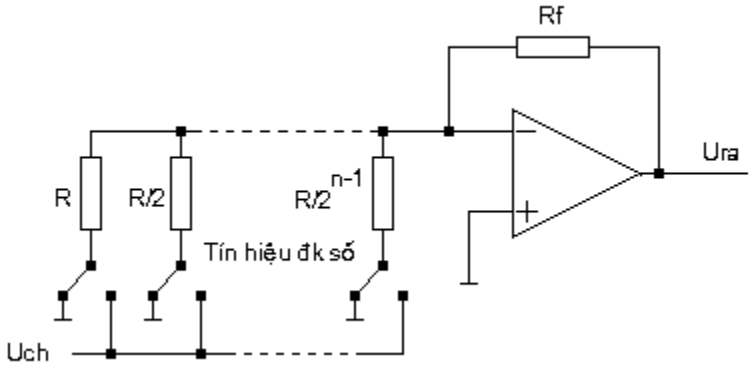
  Thời gian thiết lập trước hết phụ thuộc vào kiểu chuyển mạch, kiểu điện trở và kiểu khuếch đại dùng để xây dựng bộ DAC. Thông thường nó được định nghĩa bằng thời gian từ khi điện áp bắt đầu thay đổi cho tới khi đạt tới vùng giới hạn sai số cho trước. Nó không bao gồm thời gian trễ tính từ khi có sự thay đổi mã số ở đầu vào cho tới khi điện áp ra bắt đầu đáp ứng.

6.2.3 Các mạch DAC điển hình

Các DAC có thể được xây dựng theo một trong những kiểu mạch sau:

- Chuyển đổi DA theo kiểu điện trở trọng lượng
- Chuyển đổi DA theo kiểu mạch R-2R
- Chuyển đổi DA theo phương pháp mã hoá Shannon-Rack

1. Bộ chuyển đổi DA theo kiểu điện trở trọng lượng (Weighted resistor DAC)



Hình 6.9: DAC theo phương pháp điện trở trọng lượng

Mạch gồm một nguồn điện áp chuẩn U_{ch}, các chuyển mạch, các điện trở có giá trị lần lượt là R, R/2, R/4, ... , $R/2^{n-1}$ và các mạch khuếch đại thuật toán.

Với mạch như trên, khi một khoá điện thứ *i* nào được nối với nguồn điện thế chuẩn thì sẽ cung cấp cho bộ KĐTT dòng điện có giá trị là:

$$I_i = \frac{U_{ch}}{R/2^i}$$

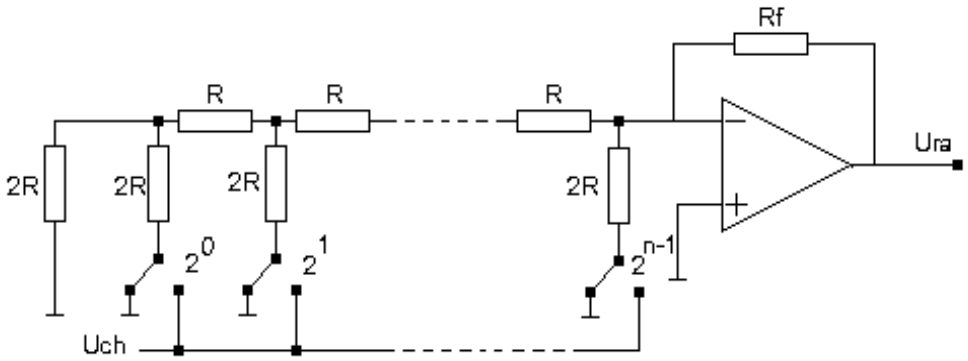
Dòng điện này độc lập với các khoá còn lại. Như vậy có thể thấy ngay rằng biên độ điện áp ra phụ thuộc vào các vị trí được đóng hay mở khoá nghĩa là được nối với điện áp chuẩn U_{ch} hay nói cách khác phụ thuộc vào giá trị các bít tương ứng trong tín hiệu số đưa vào mạch chuyển đổi. Một cách tổng quát, với một DAC có *n* bít thì tín hiệu ra được tính theo công thức:

$$U_r = -U_{ch}\frac{R_f}{R}\left(2^{n-1}.B_{n-1} + \cdots + 2^0.B_0\right)$$

Trong đó $B_0 \div B_{n-1}$ có giá trị "0" hoặc "1".

Mạch có ưu điểm là đơn giản, nhưng nhược điểm là độ chính xác và tính ổn định của kết quả phụ thuộc nhiều vào trị số tuyệt đối của các điện trở và sự ổn định của chúng trong các môi trường khác nhau. Ngoài ra còn phụ thuộc vào tính ổn định và độ chính xác của nguồn điện áp chuẩn.

2. Bộ chuyển đổi DA theo kiểu thang điện trở R-2R (R-2R ladder)



Hình 6.10: DAC theo phương pháp mạch R-2R

DAC với thang điện trở R-2R khắc phục được một số nhược điểm của DAC điện trở trọng lượng.

Mạch chỉ gồm hai điện trở R và 2R mắc theo hình thang với nhiều khoá điện (mỗi khoá điện cho một bít) và một nguồn điện áp chuẩn U_{ch}.

Đại lượng cần tìm là dòng I_{th} chảy vào mạch KĐTT khi có một số khoá điện được nối với U_{ch}. Theo mạch điện ta có:

$$Ura = -I_{th}.R_f$$

Xét tại chuyển mạch tương ứng với bít thứ *i*, nút tương ứng trên mạch hình thang là 2^i . Sử dụng định lý Thevenin, khi đóng chuyển mạch vào U_{ch} thì điện thế tương đương Thevenin tại nút 2^0 sẽ là $U_{ch}/2$ và nguồn tương có nội trở là R, như vậy tại nút 2^1 (tiến về phía mạch KĐTT) ta có nguồn tương đương Thevenin có trị số là $U_{ch}/4$ và nội trở là R.

Từ những kết quả trên suy ra rằng khi di chuyển về phía mạch KĐTT thì trị số điện thế Thevenin tại mỗi nút bằng một nửa trị số tại nút kề cận bên trái nó, và tại nút 2^{n-1} do đặc tính của bộ KĐTT điện thế được coi bằng 0V.

Một cách tổng quát, ta có công thức để tính điện áp ra của một DAC n bít với điện trở hình thang R-2R như sau:

$$U_r = -U_{ch}\frac{R_f}{2^n.R}\left(2^{n-1}.B_{n-1} + \cdots + 2^0.B_0\right)$$

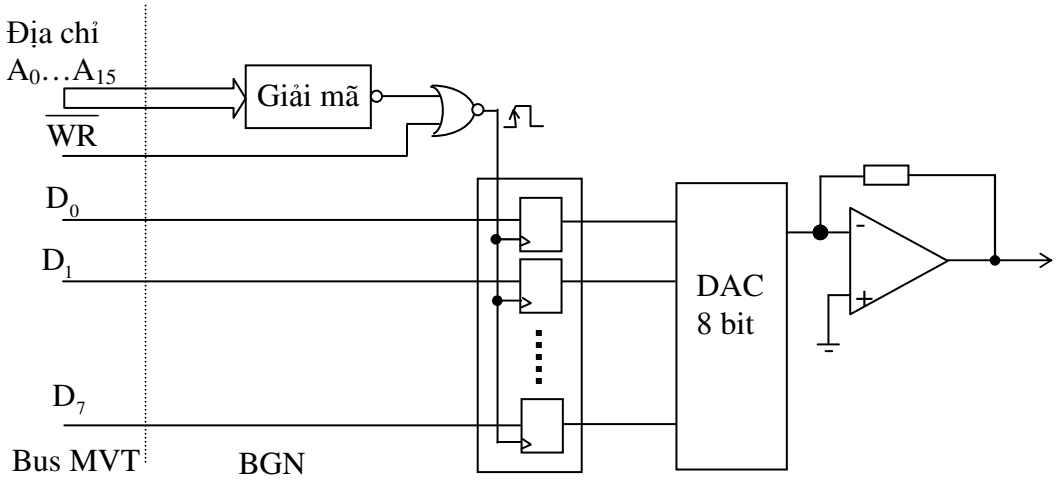
Trong đó $B_0 \div B_{n-1}$ có giá trị "0" hoặc "1".

Các DAC theo thang điện trở phải dùng số điện trở khá lớn, ví dụ nếu một DAC *n* bít thì cần dùng 2(*n*-1) điện trở trong khi phương pháp điện trở trọng lượng chỉ phải dùng *n* thôi. Nhưng bù lại độ chính xác và tính ổn định của tín hiệu ra được đảm bảo tốt hơn.

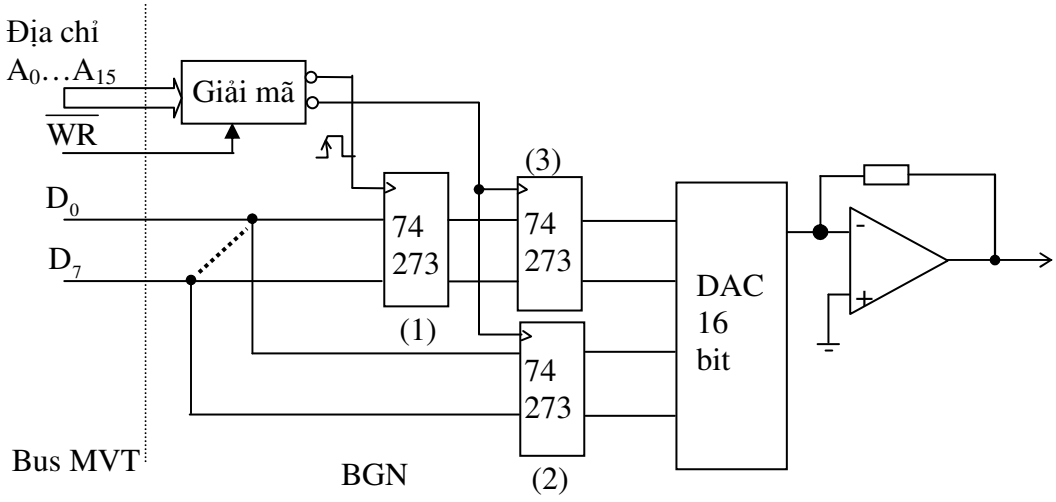
6.2.4 Ghép nối với DAC

Về nguyên tắc một bộ DAC có thể ghép nối tương thích với hầu hết các bộ VXL. Đối với các bộ DAC 8 bít, công việc thậm chí còn rất đơn giản khi ghép nối với các VXL, lý do là các VXL đều có BUS dữ liệu là bội của 8. Đối với các bộ DAC 12 hay 16 bít ta phải sử dụng các đệm trung gian có số bít tương ứng sau đó tiến hành trao đổi số liệu nhiều lần.

Hình 6.11 và 6.12 biểu diễn mạch ghép nối hệ VXL với một DAC 8 bit và DAC 16 bit.



Hình 6.11: Mạch ghép nối hệ VXL với một DAC 8 bits



Hình 6.12: Mạch ghép nối hệ VXL với một DAC 16 bit

6.2.5 Lập trình xuất dữ liệu ra cho DAC

Đối với DAC 8 bit, ta chỉ cần dùng lệnh xuất dữ liệu

Port[addr_port] := data;

Trong đó địa chỉ cổng *addr_port* sẽ qua mạch giải mã địa chỉ 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, ta cần hai lệnh xuất:

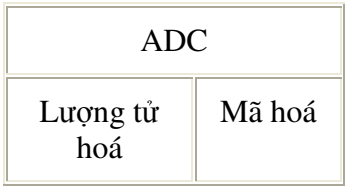
Port[addr_port1] := data1;
Port[addr_port2] := data2;

Trong đó địa chỉ cổng *addr_port1* qua mạch giải mã địa chỉ tạo xung thứ nhất kích mở mạch chốt đệm 74273 thứ nhất, lưu giữ 8 bit thấp của dữ liệu, thể hiện trong *data1*. Lệnh xuất dữ liệu thứ hai theo địa chỉ *addr_port2* sẽ tạo xung thứ hai kích mở mạch chốt đệm 74273 thứ hai (phía dưới) để đưa 8 bit cao của dữ liệu, thể hiện trong *data2*. Đồng thời xung thứ hai này cũng mở chốt đệm 74273 thứ ba để truyền số liệu 8 bit thấp ở đầu ra 74273 thứ nhất sang đầu ra 74273 thứ ba. Kết quả là DAC nhận được 16 bit dữ liệu cùng một lúc.

6.3 Bộ biến đổi tương tự số ADC

Trong phần đầu đã giới thiệu đến việc thu nhận các tín hiệu thông qua các sensor nhiệt, áp suất ... và các mạch khuếch đại thuật toán dùng để khuếch đại và lọc các tín hiệu điện này. Bước tiếp theo là bộ biến đổi AD biến đổi tín hiệu tương tự sang dạng số để có thể làm việc được với CPU.

6.3.1. Nguyên tắc làm việc của ADC



Hình 6.13: Sơ đồ minh hoạ nguyên tắc làm việc của ADC

Tín hiệu tương tự sau khi qua xử lý được đưa vào mạch lấy mẫu. Mạch lấy mẫu có hai nhiệm vụ chính:

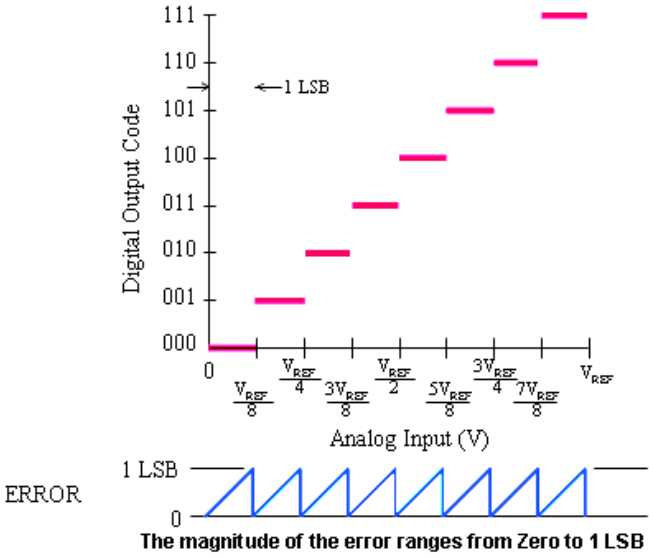
- Lấy mẫu tín hiệu tương tự tại các thời điểm khác nhau và cách đều nhau, nói cách khác đây là quá trình rời rạc hoá tín hiệu về mặt thời gian.
- Lượng tử hoá và mã hoá tín hiệu. Quá trình lượng tử hoá về bản chất là quá trình làm tròn số được thực hiện theo nguyên tắc so sánh, tín hiệu cần chuyển đổi được

so sánh với một đơn vị chuẩn. Còn mã hoá là quá trình xắp xếp lại kết quả đã lượng tử theo một quy luật nhất định tuỳ thuộc vào loại mã yêu cầu ở đầu ra bộ biến đổi (có thể là mã nhị phân hoặc mã hexa).

6.3.2. Các tham số cơ bản của bộ chuyển đổi AD

Các tham số cơ bản của bộ chuyển đổi gồm có: Dải chuyển đổi của điện áp tương tự đầu vào, độ chính xác bộ chuyển đổi và tốc độ chuyển đổi.

- Dải biến đổi của điện áp tương tự đầu vào là khoảng điện áp mà ADC có thể thực hiện chuyển đổi được. Khoảng điện áp này có thể lấy giá trị từ 0 đến một giá trị âm hay dương, hoặc cũng có thể là điện áp hai cực tính $-U_{AM} \div +U_{AM}$.
- Độ chính xác của ADC: Tham số đặc trưng cho độ chính xác của ADC là độ phân giải (Resolution). Tín hiệu ở đầu ra của một ADC là các giá trị số được xắp xếp theo một quy luật nào đó. Số các số hạng ở mã đầu ra (số bít trong từ mã nhị phân) tương ứng với dải biến đổi điện áp vào cho biết mức chính xác của phép biến đổi.



Hình 6.14: Sai số của các bộ chuyển đổi AD và DA

Ví dụ một bộ biến đổi AD có số bít đầu ra là N=12 thì sẽ phân biệt được 2^{12}=2096 mức trong dải biến đổi điện áp đầu vào của nó - U_{AM} chẳng hạn. Như vậy mỗi mức ADC phân biệt được ở đầu vào là U_{AM}/2096. Trong thực tế người ta dùng số bít N để đặc trưng cho độ chính xác của một ADC khi dải biến đổi điện áp đầu vào là không đổi.

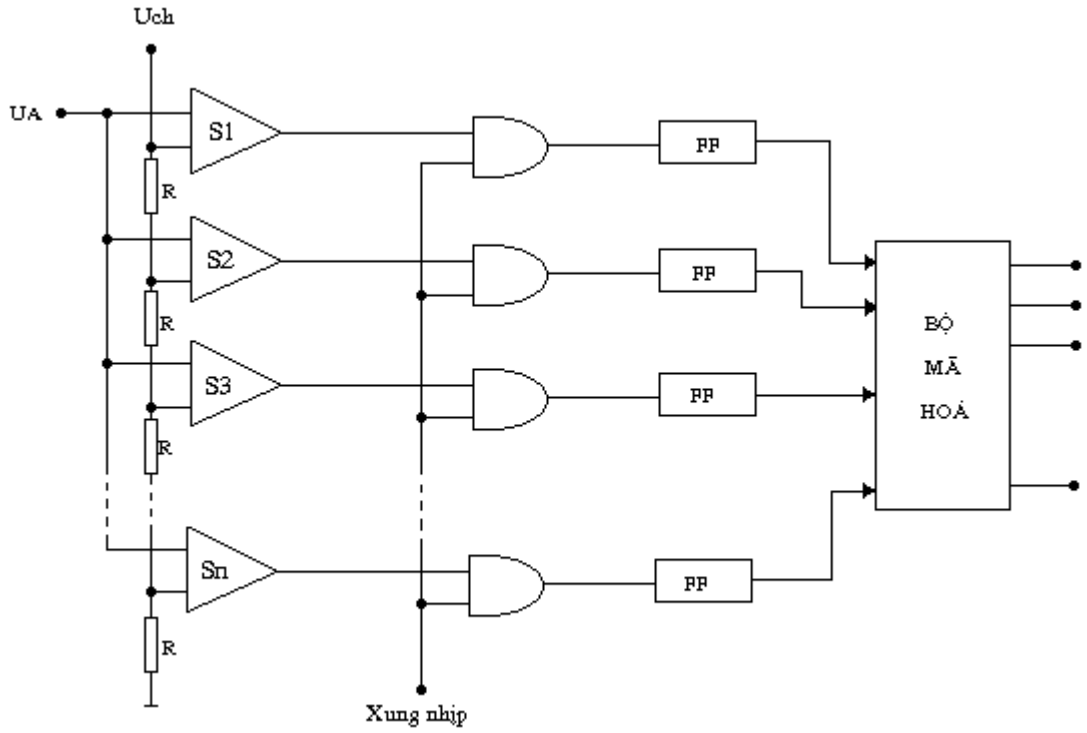
- Tốc độ chuyển đổi cho biết số kết quả chuyển đổi trong một giây, còn được gọi là tần số chuyển đổi f_c. Cũng có thể dùng tham số thời gian chuyển đổi T_c để đặc trưng cho tốc độ chuyển đổi ($T_c = 1/f_c$). Tốc độ chuyển đổi càng cao thì độ chính xác càng giảm và ngược lại.

6.3.3. Các phương pháp chuyển đổi AD

Có nhiều cách để phân loại các bộ biến đổi AD, hay dùng hơn cả là phương pháp phân loại về mặt thời gian. Phương pháp này có ưu điểm là cho phép phán đoán một cách tổng quát về tốc độ chuyển đổi.

- Chuyển đổi song song (Parallel Comparator ADC): Tín hiệu cần chuyển đổi được so sánh cùng một lúc với nhiều giá trị chuẩn, vì vậy các bít được xác định đồng thời và đưa đến đầu ra.
- Chuyển đổi nối tiếp theo mã đếm: Quá trình so sánh được thực hiện từng bước theo quy luật mã đếm. Kết quả chuyển đổi được xác định bằng cách đếm số lượng giá trị tín hiệu tương tự cần chuyển đổi.
- Chuyển đổi nối tiếp theo mã nhị phân: Quá trình so sánh được thực hiện từng bước theo quy luật của mã nhị phân. Các đơn vị chuẩn dùng để so sánh lấy các giá trị giảm dần theo mã nhị phân, do đó các bít được xác định lần lượt từ bít có ý nghĩa nhất MSB (Most Significant Bit) đến bít có ít ý nghĩa nhất LSB (Least Significant Bit).
- Chuyển đổi song song - nối tiếp kết hợp: Trong phương pháp này qua mỗi bước so sánh có thể xác định được tối thiểu là 2 bít đồng thời.

1. Chuyển đổi theo phương pháp song song



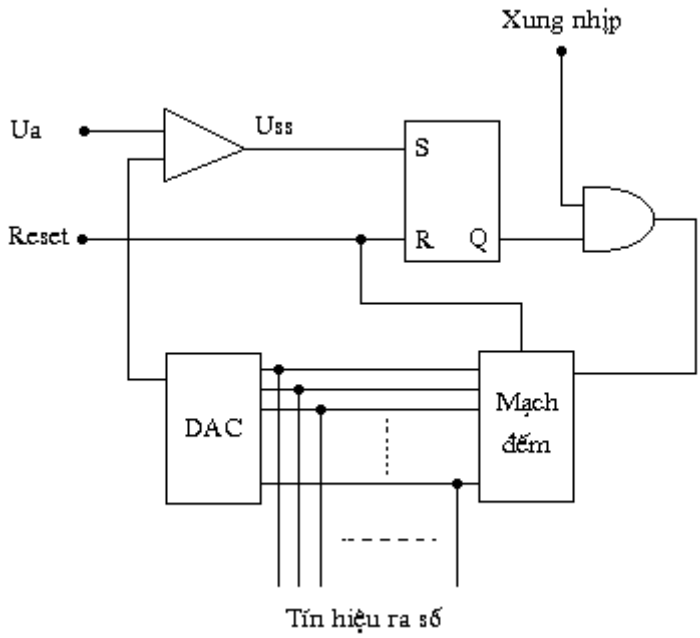
Hình 6.15: Sơ đồ nguyên lý bộ chuyển đổi AD theo phương pháp song song

Nguyên tắc hoạt động của bộ chuyển đổi: Tín hiệu tương tự U_A được đồng thời đưa đến các bộ so sánh từ $S_1 \div S_m$. Điện áp chuẩn U_{ch} được đưa đến đầu vào thứ hai của bộ so sánh qua thang điện trở R. Do đó các điện áp chuẩn đặt vào bộ so sánh lân cận khác nhau một lượng không đổi và giảm dần từ $S_1 \div S_m$. Đầu ra của các bộ so sánh có điện áp chuẩn lấy trên thang điện trở thì có mức logic "1", còn ngược lại có mức logic "0".

Các đầu ra của mạch được nối với một mạch "AND", đầu còn lại của mạch "AND" nối với xung nhịp. Chỉ khi xuất hiện xung nhịp ở đầu vào mạch "AND" thì xung ra của bộ so sánh được vào mạch nhớ Flip - Flop (FF). Như vậy xung nhịp đóng vai trò đảm bảo cho quá trình so sánh và đưa tín hiệu vào bộ nhớ, sau đó bộ mã hoá sẽ biến đổi tín hiệu vào dưới dạng mã đếm thành mã nhị phân (có thể là dạng khác).

Mạch biến đổi song song có ưu điểm là tốc độ chuyển đổi nhanh , nhưng kết cấu mạch khá phức tạp nếu ta tăng độ phân giải của phép biến đổi AD (resolution). Thực vậy để thiết lập một bộ biến đổi có độ phân giải n bít thì cần phải có $(2^n -1)$ bộ so sánh, do vậy phương pháp này chỉ dùng cho các bộ ADC tốc độ cao và độ phân giải thấp.

2. Chuyển đổi AD theo phương pháp đếm



Hình 6.16: Sơ đồ khối mạch ADC kiểu đếm

Mạch bao gồm các phần tử: Bộ so sánh, RS Flip-Flop, cổng "AND" dùng để đóng mở xung nhịp, bộ đếm n bít và một DAC n bít. Các phần tử được mắc thành mạch hồi tiếp.

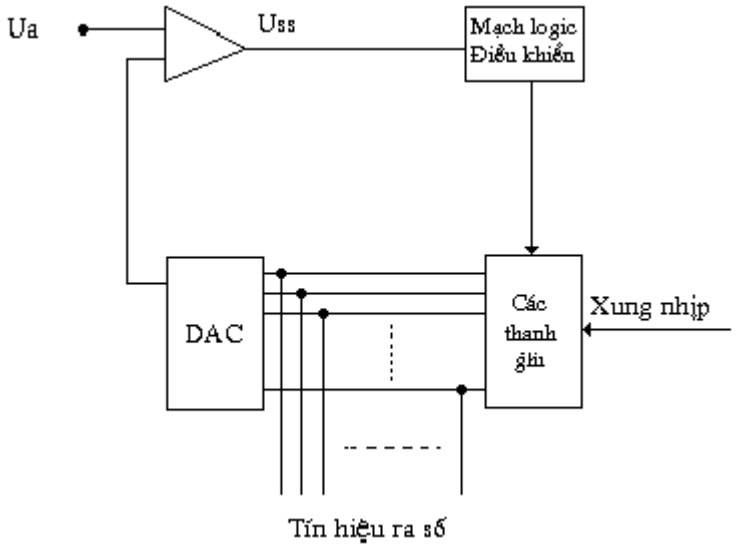
Hoạt động của mạch: Lúc đầu mạch Flip-Flop và mạch đếm được Reset, lối ra Q của FF ở mức logic "0", bộ đếm cũng xoá về 0, do đó điện áp ra U_{DAC} của DAC là 0V. Cổng AND mở để cho các xung nhịp vào mạch đếm. Lúc này điện áp vào U_a lớn hơn U_{ADC} nên U_{ss} ở trị thấp. U_{DAC} tăng dần theo hình bậc 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 FF chuyển lên mức logic "0" làm cổng AND đóng lại. Lúc này nội dung bộ đếm là tín hiệu số n bít tương ứng với tín hiệu tương tự U_a cần chuyển đổi.

Như vậy thời gian chuyển đổi của ADC kiểu 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 ADC kiểu đếm còn có một đặc điểm là thời gian chuyển đổi của mạch sẽ hạn chế tần số biến thiên cao nhất của tín hiệu tương tự đầu vào U_a.

3. Chuyển đổi AD theo phương pháp xấp xỉ liên tiếp



Hình 6.17: Sơ đồ khối mạch ADC xấp xỉ liên tiếp

Chuyển đổi AD theo phương pháp xấp xỉ liên tiếp có ưu điểm lớn là thời gian chuyển đổi tỷ lệ thuận với số bít của mã số và thời gian thiết lập của mạch ghi chứ không phụ thuộc vào độ lớn của tín hiệu chuyển đổi.

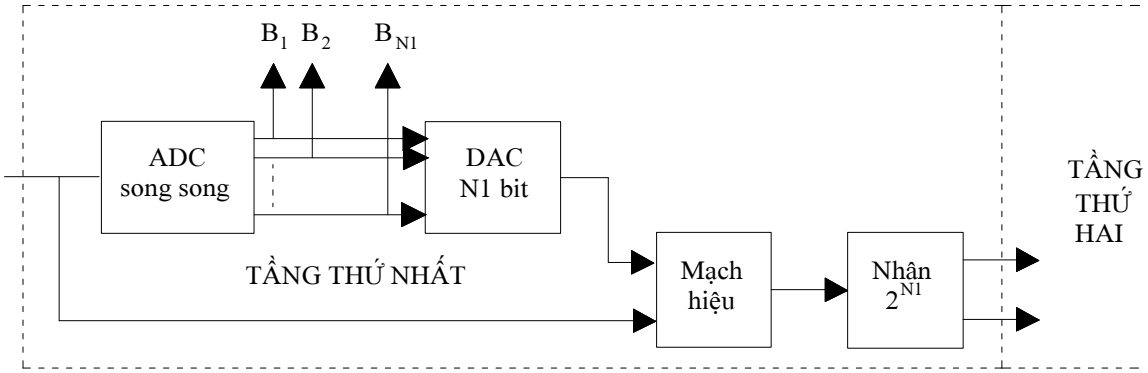
Khởi đầu mạch ghi ở trạng thái Reset, sau đó mạch điều khiển kích thích để bít lớn nhất xuất hiện $B_{n-1} = 1$. Đầu ra DAC sẽ trở thành $U_{ADC} = U_{ch}/2$. Khi đó sẽ xảy ra một trong hai trường hợp:

- Nếu $U_{DAC} > U_a$ thì bít này được bỏ qua (thiết lập "0")
- Nếu $U_{ADC} < U_a$ thì bít này được duy trì (thiết lập "1")

Sau đó lại kích thích để bít kế tiếp xuất hiện $B_{n-2} = 1$. Lúc này đầu ra DAC sẽ đạt được là $U_{ADC} = B_{n-1}\frac{U_{ch}}{2} + \frac{U_{ch}}{4}$. Và công việc so sánh tương tự trên để xác lập bit B_{n-2}.

Công việc so sánh cứ tiếp tục đến khi $U_{DAC} = U_a$ thì quá trình chuyển đổi kết thúc.

4. Chuyển đổi AD theo phương pháp song song- nối tiếp kết hợp

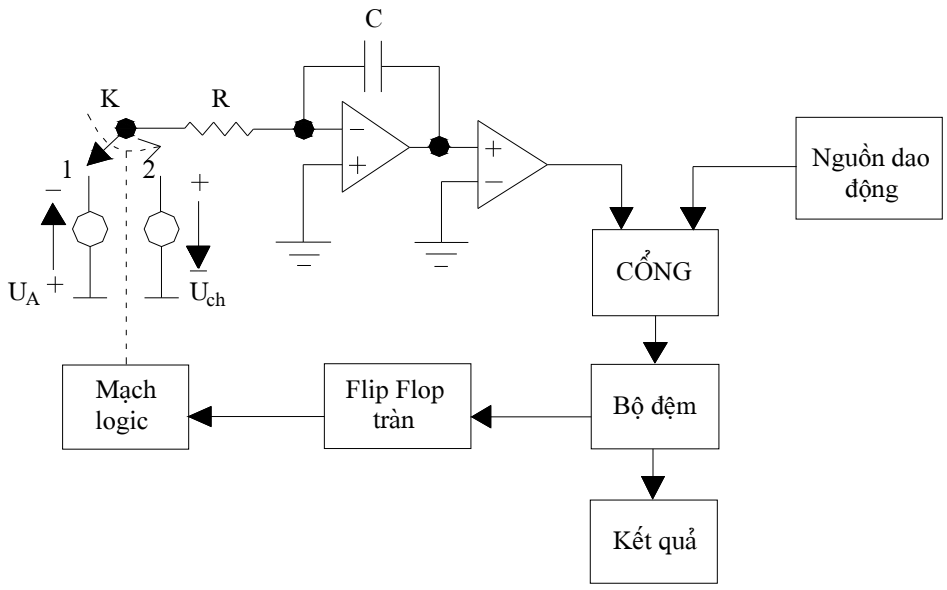


Hình 6.18: Bộ chuyển đổi AD theo phương pháp song song nối tiếp kết hợp.

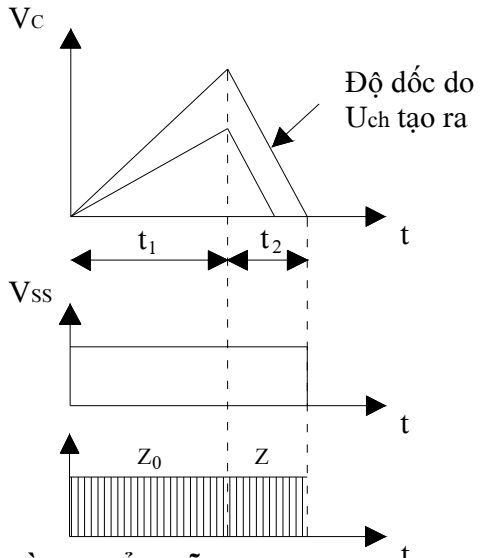
Đây là sự kết hợp phương pháp song song và phương pháp nối tiếp nhằm dung hòa ưu khuyết điểm của hai phương pháp này: giảm bớt độ phức tạp của phương pháp song song và tăng tốc độ chuyển đổi so với phương pháp nối tiếp.

Cũng có thể gọi đây là phương pháp phân đoạn từng nhóm bit, với số bit trong mỗi nhóm $N \geq 2$.

5. Chuyển đổi AD theo phương pháp tích phân hai sườn dốc



Hình 6.19: Bộ chuyển đổi AD theo phương pháp tích phân hai sườn dốc.



Hình 6.20: Đồ thị biểu diễn nguyên lý hoạt động của mạch.

Trong đó, t_1 là thời gian đếm ứng với số xung làm bộ đếm bị tràn.

t_2 là thời gian tích điện áp chuẩn V_{ch}.

V_C là điện áp răng cưa ở đầu ra của bộ tích phân.

V_{SS} là điện áp ra của bộ so sánh.

Z là số xung đếm được.

Z_o là số xung trong thời gian t_0.

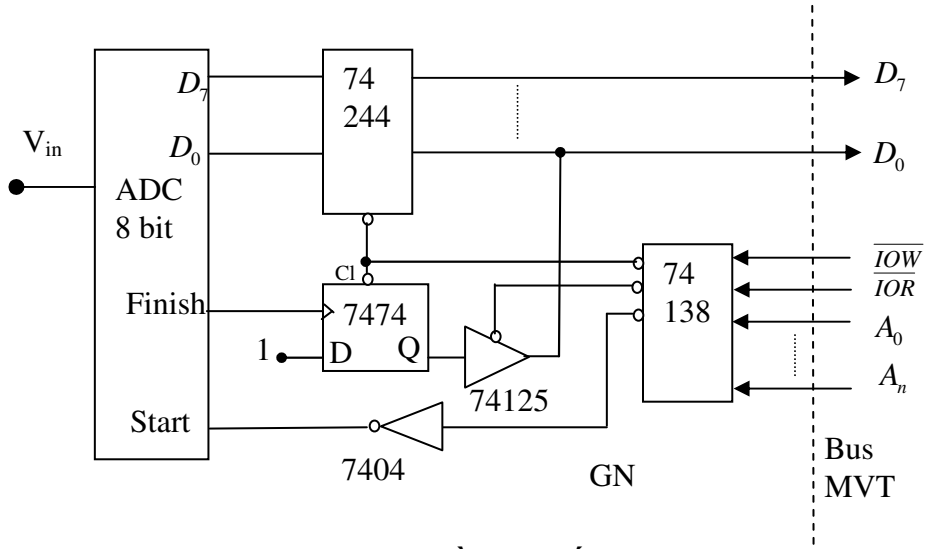
V_{ch} là điện áp chuẩn có cực tính như hình vẽ.

V_A là điện áp vào (cực tính như hình vẽ).

Ở trạng thái đầu tiên, khóa K luôn đặt ở vị trí 1. Mạch tích phân sẽ tích phân V_A, trong khi đó bộ đếm sẽ đếm xung từ nguồn dao động chuẩn tần số f_n. V_A được tích phân trong thời gian t_1 cho đến khi bộ đếm bị tràn (thời điểm t_1). Lúc này mạch logic sẽ điều khiển chuyển khóa K sang vị trí 2 và mạch tích phân sẽ tiếp tục tích phân V_{ch} nhưng với chiều ngược lại vì V_{ch} có cực tính ngược cực tính V_A. Khi tín hiệu ra của bộ tích phân V_C giảm xuống bằng 0 thì mạch so sánh sẽ đóng cổng. Nội dung ghi trong bộ đếm là kết quả biến đổi. Nó tỉ lệ với thời gian tích phân điện áp chuẩn t_2.

6.3.4. Ghép nối với ADC

Phần lớn các ADC cần một tín hiệu Start để bắt đầu quá trình chuyển đổi. Sau khi kết thúc chuyển đổi, một tín hiệu Finish được thiết lập để báo hiệu quá trình chuyển đổi đã kết thúc. Do đó ta có thể dùng mạch ghép nối ra số liệu đơn giản có đối thoại để ghép nối với ADC. Hình 6.18 biểu diễn một mạch ghép nối với một ADC 8 bit. Quá trình chuyển đổi được khởi động bằng một xung dương Start. Kết thúc quá trình này, số liệu sẽ được đưa ra lối ra ADC, đồng thời tín hiệu Finish chuyển từ "0" lên "1" . Lối ra ADC được nối với mạch đệm ba trang thái 74244. Chân xuất của 74244 được nối với chân Clear của mạch lật 7474 để xóa trạng thái sẵn sàng số liệu của ADC. Một xung đọc số liệu ADC sẽ đồng thời xóa luôn đầu ra Q của 7474 về "0" để tránh việc đọc hai lần cùng 1 số liệu. Xung Finish đưa mức "1" ra chân Q của mạch lật 7474 rồi qua mạch đệm 3 trạng thái 74125 để đưa vào đường dữ liệu D_0.



Hình 6.18: Sơ đồ ghép nối ADC với hệ VXL

Đối với ADC 16 bit, chúng ta cần một bộ chốt đệm thứ hai để đọc số liệu 8 bit cao theo một địa chỉ thứ hai.

6.3.5 Lập trình nhận dữ liệu từ ADC

Quá trình nhận dữ liệu từ ADC trên hình 6.18 được thực hiện qua các bước sau:
+ Tạo xung Start để bắt đầu quá trình chuyển đổi AD
+ Kiểm tra việc chuyển đổi đã thực hiện xong ?
+ Đọc dữ liệu vào

Địa chỉ để tạo tín hiệu Start từ 74138 : addr_St
Địa chỉ để tạo tín hiệu mở chốt 74125 : addr_Fh
Địa chỉ để tạo tín hiệu mở đệm 74244: addr_ADC

```
Port[addr_St]:=0;
Repeat
Var1 := port[addr_Fh];
Until ((Var1 AND 1)=1)
//Delay(thoigianchuyendoi) -> khong dung mach doc finish
 Data :=port[addr_ADC];
```

Ghép nối và điều khiển thiết bị ngoại vi

CHƯƠNG 7: ỨNG DỤNG MÁY VI TÍNH TRONG ĐO LƯỜNG VÀ ĐIỀU KHIỂN

7.1 Giới thiệu về cấu trúc máy tính:

Hình 7.1 trình bày các bộ phận trên bo mạch chính của hệ thống máy vi tính (MVT). MVT là một hệ thống vi xử lý bao gồm một bộ xử lý trung tâm CPU, các bộ nhớ ROM, RAM, các vi mạch ghép nối song song, nối tiếp, các bộ điều khiển ngắt và DMA ….

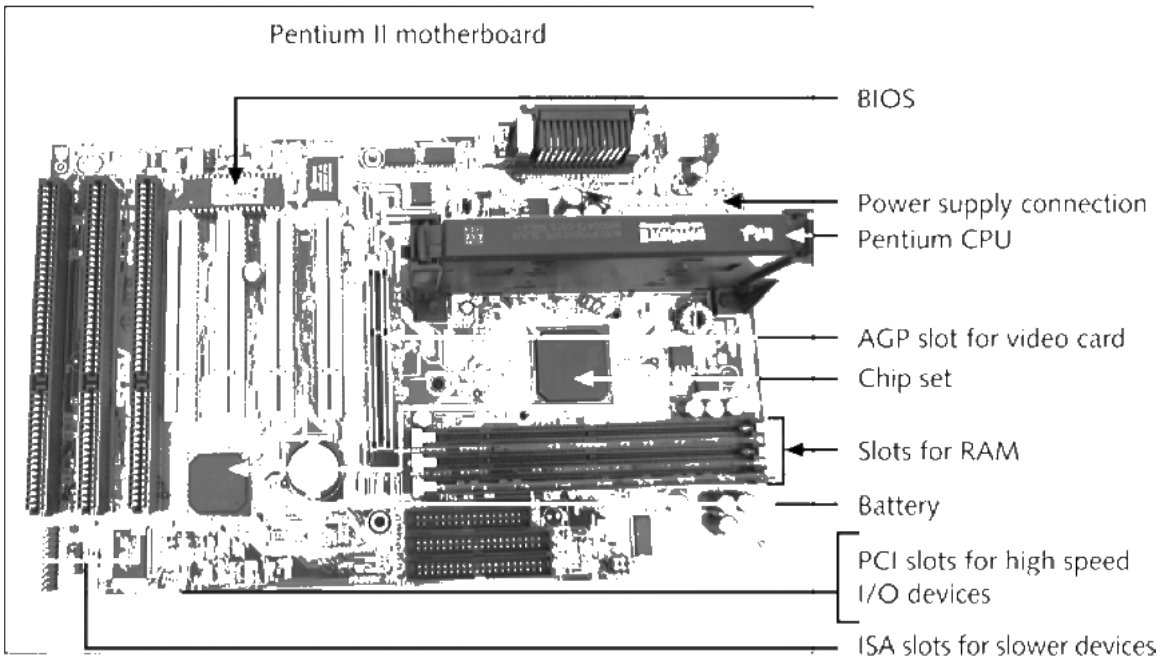


Figure 7.1: All hardware components are either located on the motherboard or directly or indirectly connected to it, because they must all communicate with the CPU

Hình 7.1 và 7.2 cho thấy MVT có ra sẵn một số các cổng cho phép ghép nối với các thiết bị vào ra cơ bản (như màn hình, bàn phím, chuột, máy in…) và các cổng vào ra khác (như cổng USP, khe cắm mở rộng ISA/PCI slots) . Do đó một vấn đề đặt ra là phải thiết kế các mạch ghép nối giữa TBNV với các cổng này, phục vụ cho việc đo lường và điều khiển một hệ thống tự động hóa.

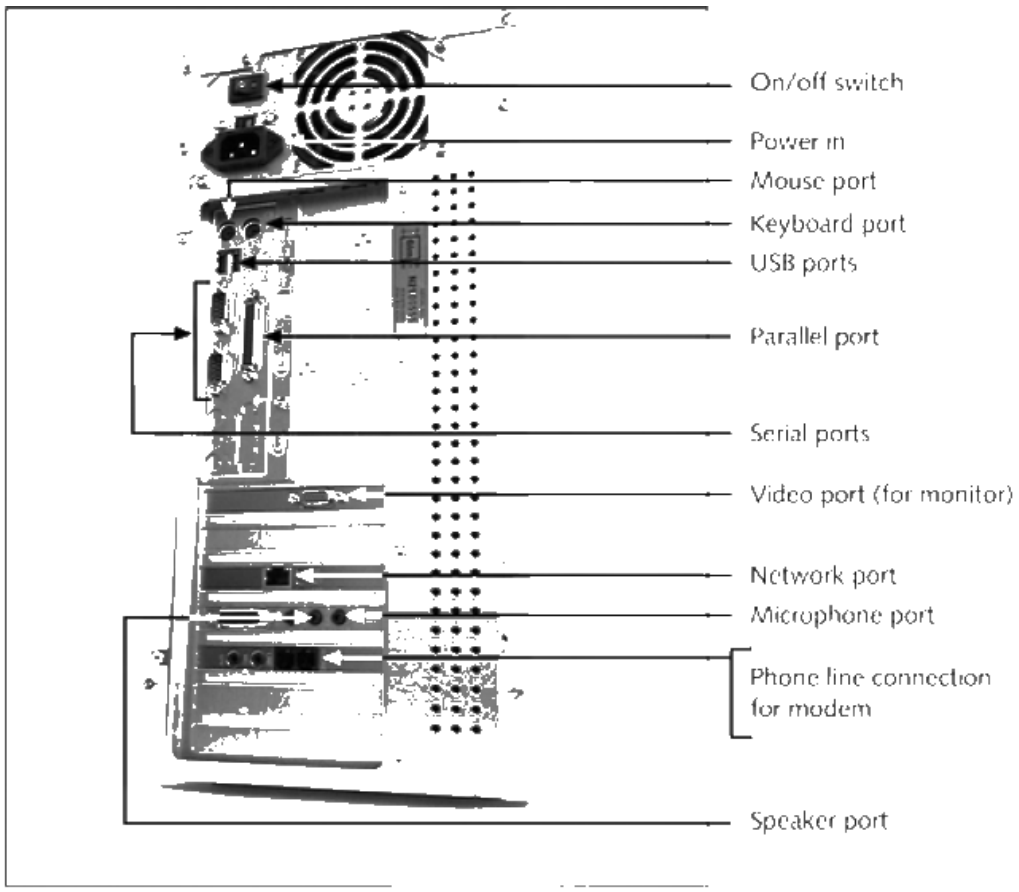


Figure 7.2: Input/output devices connect to the computer case by ports usually found on the back of the case

Chú ý rằng VXL chỉ có một bus dữ liệu, trong khi có rất nhiều thiết bị ngoại vi và ô nhớ cùng muốn trao đổi dữ liệu với VXL qua bus này. Do đó mỗi thiết bị ngoại vi sẽ được cấp một địa chỉ riêng biệt, cho phép nó làm việc với VXL khi địa chỉ riêng biệt đó được đưa ra trên bus địa chỉ. Bảng 7.1 trình bày các địa chỉ vào ra cơ bản cho các thiết bị ngoại vi trong máy vi tính IPM.

Bảng 7.1: Địa chỉ vào/ra cơ bản cho các thiết bị trong các máy vi tính IPM PC

Base Address	Device
0F - 1F	First DMA Controller 8237 A-5
20 - 3F	First Interrupt Controller 8259A, Master
40 - 5F	8254-2
60 - 6F	8042, Keyboard

70 - 7F	Real Time Clock, NMI mask
80 - 9F	DMA Page Register, 74LS612
87	DMA Channel 0
83	DMA Channel 1
81	DMA Channel 2
82	DMA Channel 3
8B	DMA Channel 5
89	DMA Channel 6
8A	DMA Channel 7
8F	Refresh
A0 - BF	Second Interrupt Controller 8259A, Slave
C0 - DF	Second DMA Controller 8237 A-5
F0	Clear 80287 Busy
F1	Reset 80287
F8 - FF	80287 Math coprocessor
170-177	Secondary ATA Hard Disk Controller
1F0 - 1F7	Primary ATA Hard Disk Controller
200 - 20F	Game Controller/Joystick
210 - 217	Expansion Unit
278 - 27F	LPT2 Parallel port

280 - 29F	LCD on Wyse 2108 PC SMC Elite default factory setting
2E8 – 2EF	COM4
2E1	GPIB Adapter 0
2F8 - 2FF	COM2
2E2 – 2E3	Data acquisition
300 - 31F	Prototype Card
300 - 31F	3Com EtherLink default factory setting
320 - 32F	XT Hard disk interface
378 - 37F	LPT1 Parallel port
380 – 38C	SLDC/Secondary Bi-sync interface
3A0 – 3A9	Primary Bi-sync interface
3B0 – 3BB	Monochrome display
3C0 – 3CF	EGA display control
3D0 – 3DF	Color/Graphics Display (CGA)
3E8 – 3EF	COM3
3F0 – 3F7	Floppy disk controller
3F8 – 3FF	COM1

7.2 Thiết kế mạch ghép nối giữa hệ thống tự động hóa với các cổng của máy vi tính

Có thể nói việc ứng dụng máy vi tính trong việc điều khiển tự động một hệ thống thực tế là rất đa dạng. Dưới đây là một vài ứng dụng đó, được giao cho sinh viên tự tìm hiểu, thiết kế và trình bày.

Đề tài 1: THIẾT KẾ MẠCH GHÉP NỐI CỔNG MÁY IN LPT1 ĐỂ ĐO CÂN ĐIỆN TỬ ÔTÔ.

Đề tài 2: THIẾT KẾ MẠCH GHÉP NỐI KHE CẮM MỞ RỘNG ĐỂ ĐO VÀ ĐIỀU KHIỂN TỐC ĐỘ ĐỘNG CƠ ĐIỆN MỘT CHIỀU ĐM.

Đề tài 3: THIẾT KẾ MỘT KIT VI ĐIỀU KHIỂN 89C51 CÓ GHÉP NỐI CỔNG NỐI TIẾP COM1 CỦA MÁY TÍNH ĐỂ ĐO NHIỆT ĐỘ VÀ BÁO CHÁY.

Đề tài 4: THIẾT KẾ MẠCH GHÉP NỐI CỔNG NỐI TIẾP RS232 ĐỂ ĐO ÁNH SÁNG VÀ ĐIỀU KHIỂN ĐÈN.

Đề tài 5: THIẾT KẾ MỘT KIT VI ĐIỀU KHIỂN PIC 16F877 CÓ GHÉP NỐI CỔNG NỐI TIẾP COM1 CỦA MÁY TÍNH ĐỂ ĐO VÀ ĐIỀU KHIỂN NỒNG ĐỘ ÔXY TRONG KHÔNG KHÍ.

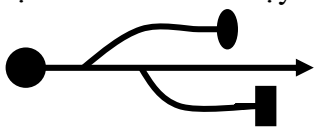
Đề tài 6: THIẾT KẾ MẠCH GHÉP NỐI CỔNG NỐI TIẾP ĐA NĂNG USB ĐỂ ĐIỀU KHIỂN MỘT HÀNH TRÌNH DÙNG ĐỘNG CƠ BƯỚC.

Đề tài 6: Thiết kế một mạch ghép nối với cổng đa năng USB dùng để điều khiển hành trình động cơ bước

PHẦN I: GIỚI THIỆU VỀ CỔNG ĐA NĂNG USB

I. Giới Thiệu Chung Về Bus USB

Cổng USB về thực chất là một bus vì qua đó có thể đấu nối đồng thời rất nhiều thiết bị ngoại vi với chủng loại khác nhau. Vì vậy có thể gọi bus USB là một bus nối tiếp đa năng



1. Những nét chung về bus USB:

- Bus USB là một bus nối tiếp
- Dữ liệu truyền trên bus theo từng bus kế tiếp nhau .Đặc biệt, dữ liệu truyền trên cùng các đường dẫn theo hai hướng.
- Bus USB cho phép đấu nối đến 127 thiết bị
- Dữ liệu truyền với tốc độ đến 12 Mbips
- ✓ VớI bus USB 1.0 ,tốc độ truyền có thể lớn hơn 12Mbps
- ✓ VớI bus USB 2.0 ,tốc độ truyền có thể lên đến 480Mbps
- Bus USB có mối liên hệ chặt chẽ với đặc tính "Plus and Play" được gọi là đấu /ngắt nóng
- Điện áp nguồn nuôi có thể nhận được từ bus.Các thiết bị nhỏ có thể sử dụng điện áp từ trên bus mà không cần nguồn nuôi riêng.

❋Các đặc tính cơ bản của bus USB là:

- ➢ Các cuộc truyền đẳng thời,có thể hiểu là truyền liên tục, hỗ trợ các tín hiệu vidio và âm thanh.Với các cuộc truyền đẳng thời, các thiết bị truyền và nhận dữ liệu theo kiểu được đảm bảo và có thể đoán trước.
- ➢ USB cũng được hỗ trợ các thiết bị không đẳng thời hay thiết bị có quyền ưu tiên cao nhất , các thiết bị đẳng thời hoặc không đẳng thời tồn tại cùng thời điểm.
- ➢ Các thông số kỹ thuật cũng có đặc tính cắm và chạy, các cáp nối và cách kết nối đều được tiêu chuẩn hóa rộng rãi trong công nghiệp.
- ➢ Các hub được sắp xếp theo nhiều tầng với khả năngg mở rộng gần đến mức lớn nhất và thao tác xảy ra đồng thời.
- ➢ Tốc độ truyền 12Mbps với các kích thước khác nhau.
- ➢ Hỗ trợ nhiều yêu cầu dải thông thiết bị từ một vài kbps đến 19Mbps.
- ➢ Hỗ trợ tốc độ truyền dữ liệu của thiết bị trên một phạm vi rộng các giá trị thông qua việc điều tiết kích thước bộ đệm gói dữ liệu và cơ chế tiềm ẩn.
- ➢ Có khả năng cắm nóng ,nghĩa là cho phép các thiết bị ngoại vi có thể được đấu nối mà không cần phải tắt nguồn nuôi cung cấp điện cho máy tính ,có thể đấu/ngắt và thay đổi lại cấu hình thiết bị ngoại vi một cách linh hoạt khả năng quản lý được tăng cường với các chế độ "ngủ "và "nằm lỳ "trên phạm vi hệ thống.

- Tự nhận dạng thiết bị ngoại vi ,tự động vẽ bản đồ chức năng đối với phần mềm điều khiển và cấu hình
- Hỗ trợ cho các thiết bị loại khác nhau, với nhiều chức năng khác nhau.
- Điều khiển luồng dữ liệu thông qua bộ đệm bằng việc quản lý giao thức đặt sẵn bên trong .
- Có cơ chế xử lý lỗi/ hoàn trả lỗi.
- Hỗ trợ khả năng nhận dạng các thiết bị mắc lỗi.
- Giao thức đơn giản trong việc thực hiện và tích hợp

2. Đấu nối và cáp:

- Bus USB có hai kiểu đấu nối khác nhau: kiểu A và kiểu B
- Bus USB sử dụng một cáp bốn sợi để đấu nối với các thiết bị ,trong đó một cặp đường truyền hai sợi xoắn được dùng làm các đường dẫn dữ liệu vi phân, còn hai cặp kia được dùng làm đường dẫn 5V và đường nối đất chung GND

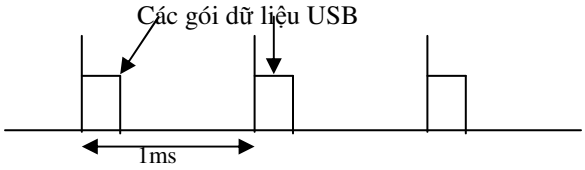
Chân	Tên gọi	Màu dây	Mô tả
1	Vcc	Đỏ	+5VDC
2	D-	Trắng	Dữ liệu-
3	D+	Xanh lục	Dữ liệu+
4	GND	Đen	NốI đất

Bảng đấu nối ở bus USB

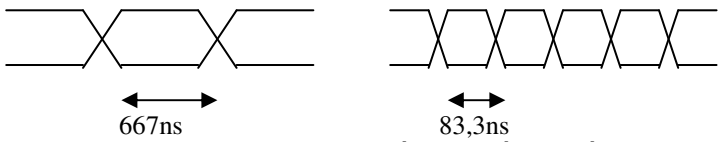
- Các máy tính PC đời mới nhất đều là kiểu A.Còn các trường hợp khác thiết bị thường có ổ cắm kiểu B.
- Việc đấu nối với máy tính được thực hiện bằng một cáp kiểu A-B
- Các cáp dùng để kéo dài khoảng cách từ máy tính đến thiết bị thường là kiểu A-A
- Khi kết nối các thiết bị với bus USB ta thường phải phân biệt rõ các thiết bị sử dụng nguồn nuôi riêng ,chẳng hạn máy in ,với các thiết bị nhận nguồn nuôi qua bus

3. Truyền dữ liệu nối tiếp:

- Toàn bộ dữ liệu được trao đổi đều có một khung dúng bằng 1 ms .Trong phạm vi một khung ,nhiều gói dữ liệu kế tiếp dành cho nhiều các thiết bị khác nhau có thể được xử lý,trong đó có những gói cần gửi với tốc độ thấp và những gói cần gửi với tốc dộ cao cùng tồn tại trong một xung.

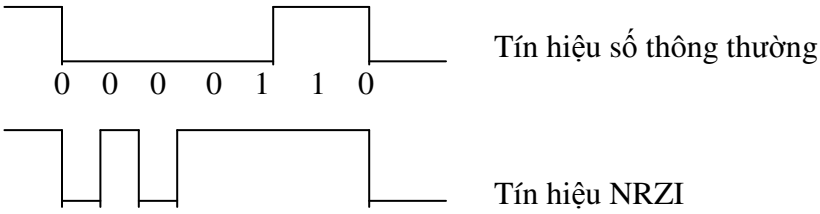


Hình 1: Các gói dữ liệu trong những khung truyền 1 ms



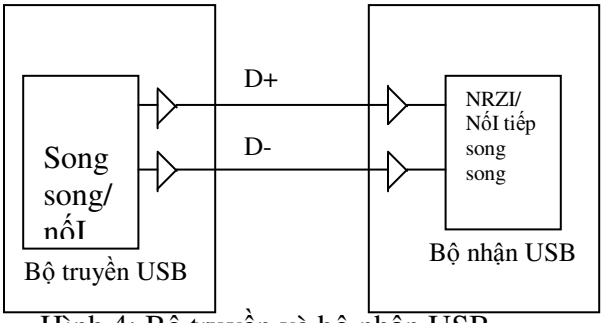
Hình 2: Các tín hiệu tốc độ thấp và tốc độ cao

- Khi cần ghép nối nhiều thiết bị USB ,ta cần có một hộp phân phối hay thường gọi là hub.Hub cũng cho phép tránh xảy ra tình trạng tín hiệu tốc độ cao được chuyển giao tới thiết bị có tốc độ thấp
- Do không có tín hiệu giữ nhịp được truyền tách riêng ra nên xung giữ nhịp cần phải được tạo ra từ chính tín hiệu mang thông tin về dữ liệu.Nên kỹ thuật "không trở về mức 0" (gọi tắt là NRZI) đã được sử dụng.Khi đó,
- ✓ Dữ liệu ứng với gía trị 0 dẫn đến sự thay đổi của mức điện áp .
- ✓ Còn ứng với giá trị 1 lại giữ nguyên mức điện áp



Hình 3: Tín hiệu NRZI

- Mã hóa và giải mã tín hiệu là công việc của riêng phần cứng .Bộ nhận cần phải phục hồi và loại ra tín hiệu giữ nhịp,nhận và giải mã dữ liệu



Hình 4: Bộ truyền và bộ nhận USB

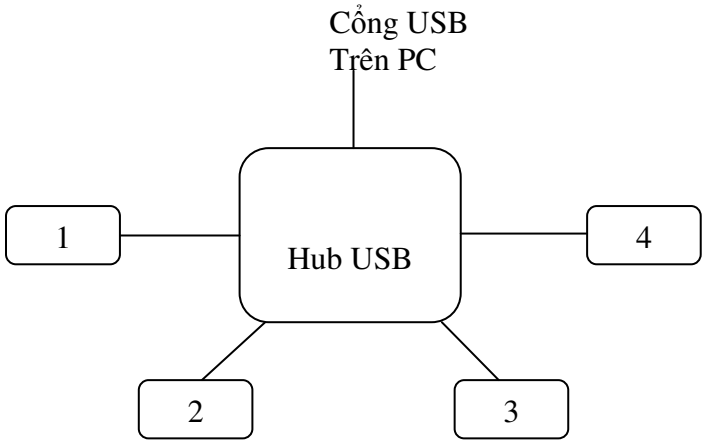
- Để thực hiện quá trình đồng bộ trong khi truyền ,người ta sử dụng một phương pháp rất là độc đáo :
- ✓ Khi dòng dữ liệu chứa 6 số 1 kế tiếp nhau thí một số 0 được bộ truyền tự động thêm vào để bắt buộc xảy ra sự thay đổi mức ,bộ nhận sẽ tự động loại bỏ số 0 này ra khỏi dòng dữ liệu.
- ✓ Cũng nhằm mục đích đồng bộ , mỗi gói dữ liệu đều có dấu hiệu đặc biệt ,đó là byte đồng bộ(Sync-byte)
- Trên thực tế bộ truyền và bộ nhận luôn được thiết kế trên cùng một vi mạch:
- ✓ Mỗi thiết bị USB có chứa một khối SIE đảm nhận nhiệm vụ này
- ✓ Để thực hiện việc trao đổi dữ liệu giữa SIE và các phần còn lại của thiết bị còn cần đến một bộ nhớ đệm FIFO .Tất cả các công việc khác do SIE thực hiện
- ✓ Thông thường ,một thiết USB có nhiều bộ nhớ FIFO ,đóng vai trò trung gian trong các cuộc truyền dữ liệu.
- Phần mềm USB tạo ra các đường ống dẫn tới các điểm cuối riêng lẻ .Mỗi đường ống là một kênh logic dẫn tới một điểm cuối trong một thiết bị .Một thiết bị có thể sử

dụng nhiều đường ống dồng thời ,vì thế tốc độ truyền dữ liệu tổng cộng được nâng lên.

II. Các HUB

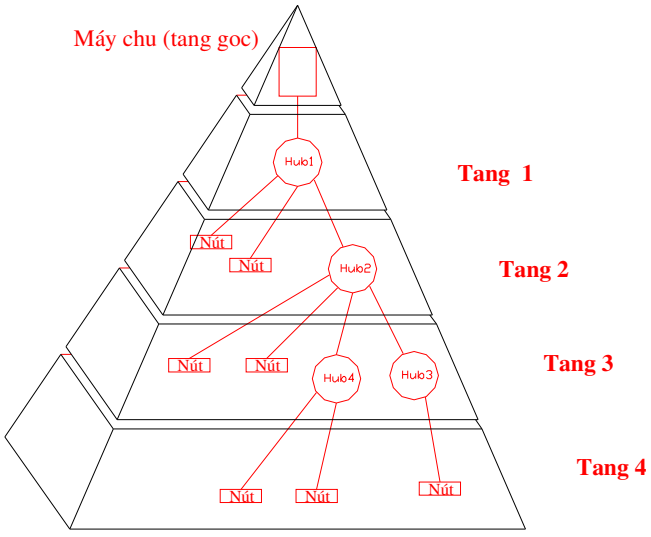
Để đầu nối nhiều thiết bị ngoại vi USB ta cần 1 hub hoặc nhiều hub. Hub là hộp phân phối bus có nhiều cổng.

Một hus ngoài có một cổng hướng về máy chủ và 4 cổng ra thiết bị ghép nối ,ngay trong máy tính PC cũng có một hub .Một loại hub trong để tạo ra 2 cổng USB kiểu A ở phía sau may vi tính.Hub này gọi là hub gốc và hub này đặt ngay trên mạch chính.(hình 5)



Hình 5: Cách đấu nối hình sao ở một hub USB

Ở cổng ra thiết bị ghép nốicủa một hub có thể đấu thêm một hub khác .Như vậy hình thành một cấu trúc phân tầng.(hình 6)

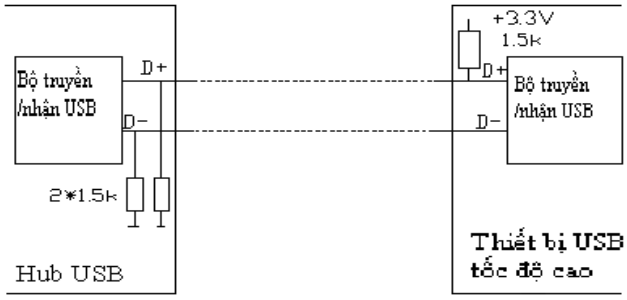


Hình 6: Cấu trúc phân tầng của các hub khi đấu nối vào bus USB

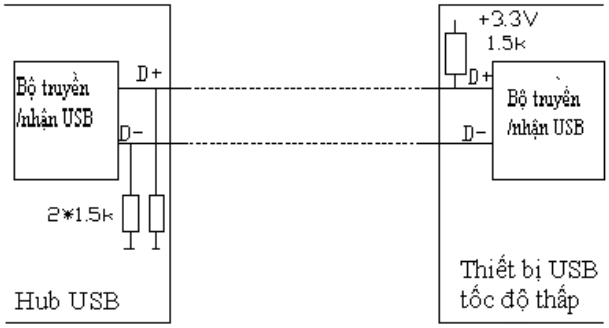
Mỗi hub và mỗi dây cáp đều gây ra sự làm trễ tín hiệu, nhưng thời gian ttrễ không vượt quá thời giá trị cực đại được qui định. Bub USB cho phép tổng cộng đến 7 hub đấu nối kế tiếp nhau, như vậy có nhiều nhất là 127 thiết bị có thể đấu nối vào một bus USB.

Một nhiệm vụ của hub nhận biết các thiết bị mới được đấu nối vào và còn phân biệt đó là thiết bị tốc độ cao hay thiết bị tốc độ thấp. Ngoài ra hub có thể xóa đi 1 thiết bị đã đấu nối vào bus nhưng sau đó lại được tháo rời khỏi hệ thống. Để thực hiện các nhiệm vụ này có những trạng thái bus được tạo ra theo cách đặc biệt.

Một cổng USB không được sử dụng sẽ không được kích hoạt nghĩa là hub không gửi khung dữ liệu tới.Cả hai đường dẫn dữ liệu đều ở mức low và có một điện trở 15k. Mỗi thiết bị ngoại vi USB đều có một điện trở 1,5k nối một trong hai đường dẫn tín hiệu với nguồn +3,3V. Thiết bị tốc độ cao điện trở này nối với nguồn +3,3Vvới đường dẫn D+, thiết bị tốc độ thấp thì nói với đường dẫn D-. Hub cũng có thể nhận biết kiểu thiết bị và có thể xây dựng mối kết nối dữ liệu với tốc độ truyền thích hợp .(hình 7, 8)



Hình 7: Nhận dạng một thiết bị tốc độ cao



Hình 8: Nhận dạng một thiết bị tốc độ thấp

Khi lần đầu tiên đặt lại chế độ cho bus, cả hai đường dẫn dữ liệu được nối mass trong khoảng thời gian 10ms. Khối SIE của thiết bị ngoai vi nhận biết trạng thái này và xóa đi một tín hiệu reset của bộ vi điều khiển được đấu nối vào. Sau đấy bộ điều khiển bắt đầu với việc thực hiện chương trình bộ điều khiển của nó và sẵn sàng trình diện ở hệ thống.

Hub cũng cung cấp điện áp nguồn nuôi cho thiết bị. Khi khởi động mỗi thiết bị USB được phép tiêu thụ dòng điện đến 100 mA. Nếu như cần một dòng lớn hơn thì nhu cầu này cần phải được khai báo .Dòng điện tiêu thụ tổng cộng có thể đạt tới 500mA. Một

hub ngoài chỉ có thể cung cấp 100 mA cho mõi cổng ra thiết bị ghép nối bởi vì nó được phép tiếp nhận tổng cộng không quá 500 mA và yêu cầu riêng cho chính bản thân hub.

III. Phần cứng và phần mềm máy chủ USB.

Máy chủ USB tương tác với các thiết bị USB thông qua bộ điều khiển ở máy chủ. Máy chủ chịu trách nhiệm về những công việc sau:

- Phát hiện việc kết nối hoặc loại bỏ của các thiết bị USB.
- Quản lý việc điều khiển dòng dữ liệu giữa máy chủ và các thiết bị USB.
- Quản lý dòng dữ liệu giữa máy chủ và các thiết bị USB.
- Thống kê trạng thái tính hoạt động của hệ thống.
- Cung cấp dòng điện đã được đặt để hạn chế công suất cho các thiết bị USB được kết nối .

Phần mềm hệ thống USB trên máy chủ quản lý sự tương tác giữa các thiết bị USB và phần mềm thiết bị dựa trên máy chủ. Có 5 vùng tương tác giữa phần mềm hệ thống USB và phần mềm thiết bị ,cụ thể là:

- Điểm danh định cấu hình thiết bị.
- Truyền dữ liệu trong chế độ đẳng thời.
- Truyền dữ liệu trong chế độ không đồng bộ.
- Quản lý năng lượng.
- Quản lý thông tin về thiết bị và bus.

Bất cứ lúc nào có thể phần mềm USB sử dụng các giao diện hệ thống máy chủ hiện hữu để quản lý các mối quan hệ tương tác ở trên.

Các thanh ghi bộ điều khiển máy chủ USB
Thanh ghi nhận dạng nhà cung cấp
VID (vendor Identìication register)
Địa chỉ offset: 00-01h
Giá trị mặc định : 8086h
Thuộc tính: chỉ đọc
Thanh ghi VID chứa số nhận dạng nhà cung cấp .Thanh ghi này ,cùng với thanh ghi nhận dạng thiết bị xác địng duy nhất bất kỳ thiết bị PCI nào .Việc ghi vào thanh ghi này không có tác dụng .Cách viết các bit 15:0 dùng cho số nhận dạng nhà cung cấp .Đây là một giá trị 16 bit được Intel gán.

IV. Kết nối hệ thống USB.

Kết nối các thiết bị USB

Tất cả các thiết bị USB đấu nối vào USB qua cổng đặt trên một loại thiết bị USB chuyên dụng, có tên là hub. Các hub chỉ cho thấy tình trạng kết nối hoặc loại bỏ của một thiết bị USB theo từng trạng thái cổng của nó.

Các máy chủ hỏi (queries) hub để xác định nguyên nhân của thông báo. Hub trả lời (đáp ứng) bằng cách nhận dạng cổng đã sử dụng để kết nối thiết bị USB. Máy chủ cho phép cổng và định địa chỉ thiết bị USB với một đường ống điều khiển bằng cách sử dụng địa chỉ mặc định USB (khi được nối vào lần đầu hoặc khi chúng khởi động lại).

Máy chủ xác định xem liệu thiết bị USB mới được kết nối là một hub hay là một thiết bị chức năng và gán cho thiết bị USB một địa chỉ USB duy nhất

Nếu thiết bị USB đã kết nối là một hub và các thiết bị USB đã được kết nối với cổng của hub thì thủ tục ở trên tiếp diễn với từng thiết bị USB đã được kết nối. Nếu thiết bị USB đã kết nối là một thiết bị chức năng, thì các thông báo về việc kết nối sẽ được phần mềm USB gửi tới phần mềm máy chủ được quan tâm đến

Loại bỏ các thiết bị USB

Khi một thiết bị USB đã được loại bỏ khỏi một trong số các cổng thì hub tự động vô hiệu hoá cổng và cung cấp một thông tin chỉ báo về việc loại bỏ thiết bị ra khỏi máy chủ.

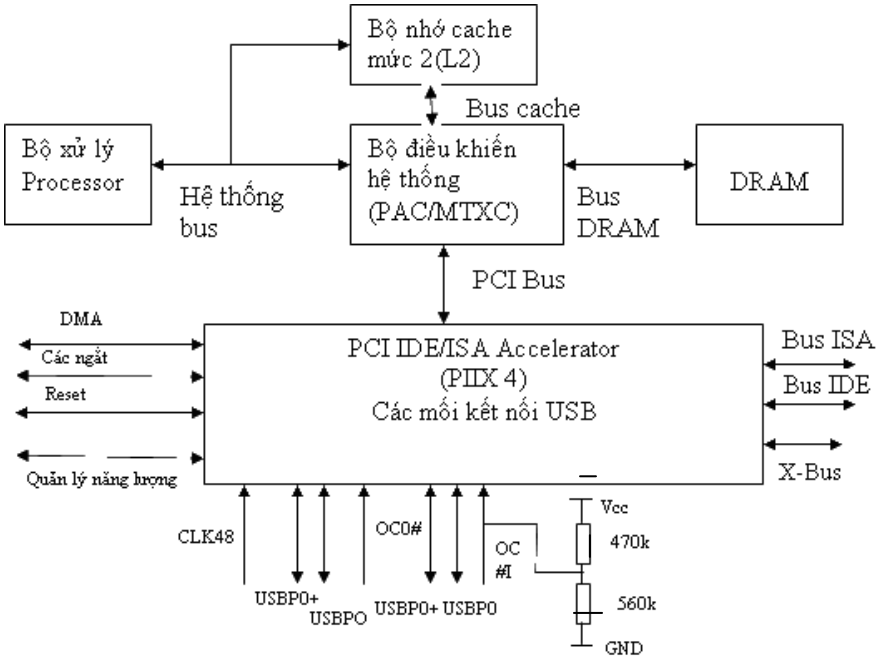
Sau đó máy chủ loại bỏ các số liệu dã biết về thiết bị USB, nếu thiết bị USB bị loại bỏ là là một hub thì qúa trình loại bỏ phải được thực hiện với tất cả các thiết bị USB mà trước đó được nối vào hub. Nếu thiết bị USB đã loại bỏ là một thiết bị chức năng thì các thông báo loại bỏ được gởi tới phần mềm máy chủ được quan tâm đến.

Kết nối hệ thống USB

Hình 3-55 cho thấy một ví dụ về kết nối cổng USB vào trong hệ thống. Lối vào CLK48 có đặt đồng hồ tinh thể 48 MHz(sai số 2500 ppm) được tạo ra nhờ bộ cộng hưởng thạch anh và được sử dụng để tạo tốc độ dữ liệu bằng 12MHz và một khoảng khung khởi tạo bằng 1,0ms (500ppm).

Thiết bị tăng tốc PVIIDE/ISA (cầu PCI) có một thiết kế thích hợp với tốc độ bằng 1,5Mbps hoặc 12 Mbps. Nguồn nuôi dùng cho thiết bị được lấy từ nguồn điện áp -5v trên bản mạch chính (Vcc). Như vậy cần có những biện pháp nào đó để chống ngắn mạch, chẳng hạn bằng một cầu chì 2A. Hệ thống giám sát của cá đường dẫn điện áp nguồn nuôi cho USB với các đường OC#1 và OC#0 (ở phía dưới của hình 3-54).

Khi thiết bị cầu PCI phát hiện ra một lỗi trên đường dẫn điện áp nuôi thì nó sẽ vô hiệu hoá cổng USB tương ứng. Trên các đường dữ liệu, có một điện trở nối tiếp bằng 27(ôm), dùng để giới hạn dòng điện khi xảy ra ngắn mạch với đất(GND). Như vậy dòng sẽ bị giới hạn ở mức nhỏ hơn 185mA. Các cuộn cảm mắc nối tiếp mắc nối tiếp cũng được tính đến trên đường dẫn điện áp nguồn nuôi, để tín hiệu nhiễu trên các thiết bị bên ngoài không gây ảnh hưởng tới nguồn nuôi bản mạch chính.



❋*PHIÊN BẢN USB 2.0*

Các kỹ thuật chính của bus USB 2.0 là:

- Tốc độ thấp (1,5Mbps): các thiết bị tương tác(interactive),thông thường là 10-100Kbps.
- Tốc độ cao(full-speed,12Mbps)các ứng dụng với điện thoại âm thanh.
- Tốc độ rất cao (480 Mbps)các ứng dụng video và bộ nhớ thông thường là 25-400Mbps.

Cổng USB 2.0 sẽ đẩy nhanh quá trình hướng tới một máy tính PC trong tương lai với các đặc tính legacy-free.

V. Các kiểu truyền USB.

Các thiết bị USB có thể trao đổi dữ liệu với máy tính PC theo 4 kiểu hoàn toàn khác nhau:

+ Truyền điều khiển: Để điều khiển phần cứng các yêu cầu điều khiển (control request) được truyền hay gọi là truyền điều khiển (control transfers). Chúng làm việc với mức ưu tiên cao và có khả năng kiểm soát lỗi tự động. Tốc độ truyền lớn vì có đến 64 byte trong một yêu cầu(request) có thể được truyền.

+Truyền ngắt: các thiết bị cung cấp một lượng dữ liệu nhỏ, tuần hoàn, chẳng hạn như bàn phím, chuột, đều sử dụng kiểu truyền ngắt (interrupt transfers). Khác với sự phỏng đoán từ tên gọi, ở đây không có một ngắt nào được thiết bị xoá. Cách tốt nhất là hệ thống sẽ hỏi theo chu kỳ, chẳng hạn 10ms một lần, xem có các dữ liệu mới gởi đến. Thông thường có đến 8 byte có thể được truyền.

+ Truyền theo khối: Khi có dung lượng dữ liệu lớn cần truyền và cần kiểm soát lỗi truyền nhưng lại không có yêu cầu thúc ép về thời gian truyền thì dữ liệu thường được truyền theo khối (bulk transfers). Các ứng dụng của phương pháp này như: máy in và máy quét hình (scanner). Tốc độ truyền dữ liệu phụ thuộc vào mức độ đấu tải trên bus USB.

+ Truyền đẳng thời: Khi có lượng lớn dữ liệu với tốc độ dữ liệu đã được quy định, chẳng hạn như dùng cho card âm thanh, thì thường áp dụng kiểu truyền đẳng thời (isochronous transfers). Theo cách truyền này một giá trị tốc độ dữ liệu xác định được duy trì. Việc hiệu chỉnh lỗi không được thực hiện vì những lỗi truyền lẻ tẻ cũng không gây ảnh hưởng đáng kể đến chất lượng truyền.

Với các ứng dụng trong lĩnh vực đo lường và điều khiển , kiểu truyền thứ nhất (truyền điểu khiển) thường được sử dụng vì vừa có độ an toàn dữ liệu cao vừa có tốc độ truyền lớn .Ngoài ra còn dễ dàng thực hiện được giao thức truyền riêng để trong một số trường hợp ứng dụng có thể tiếp tục xử lý.

.

VI. Gọi phần mềm điều khiển.

Chức năng của một phần mềm điều khiển:

Là mắt xích liên kết giữa phần cứng và phần mềm hệ thống.Trong môi trường Windows 98 về trước, không có một chương trình ứng dụng nào có thể truy cập trực tiếp lên phần cứng, cách tốt nhất là gọi ra một phần mềm điều khiển thiết bị đó, phần mềm

này trao đổi trực tiếp với phần cứng hoặc qua lớp đệm nằm sâu hơn. Phần mềm điều khiển này được chế tạo và cung cấp bởi các nhà sản xuất thiết bị đó.

Hoạt động :

Tất cả các phần mềm điều khiển USB đều dựa trên Win32_Driver_Model. Tất cả các lời gọi phần mềm điều khiển đều thông qua phần mềm quản lý vào ra (I/O manager). Các chương trình người dùng khác nhau gửi gói yêu cầu gần như đồng thời tới phần mềm quản lý vào/ra, phần mềm này sau đấy sẽ phân phối các gói này tới các phần mềm điều khiển cụ thể.

Tất cả các yêu cầu đều được gởi đi dưới dạng các gói, và chuyển tiếp giữa các lớp đệm riêng lẻ. Để biết rõ về cách thức truy nhập lên các thiết bị, ta xem xét đến 5 hàm của Windows sau:

CreateFile( )	Mở một tệp hoặc một thiết bị
CloseHandle( )	Đóng một tệp hoặc một thiết bị
ReadFile( )	Đọc hoặc nhận dữ liệu
WriteFile( )	Ghi hoặc truyền dữ liệu
DeviceloControl( )	Thực hiện các chức năng cụ thể của phần mềm điều khiển

Ví dụ xét đoạn chương trình sau (Gửi 1 byte qua COM2 bằng Delphi):

```
Handle:=CreateFile.(Pchar('COM2'),GENERIC_WRITE,0,NIL,OPEN)_EXISTING,0,0);
Byt:=85;
WriteFile(Handle,Byt1,1,Count,NIL);
CloseHandle(Handle);
```

Giải thích lệnh:

Trước hết , Cổng **COM2** được mở bằng hàm **CreateFile( )** để sau đấy gửi một Byte bằng hàm **WriteFile( ).** Cuối cùng Hàm **CloseFile( )** đóng kênh thông tin lại. Tên của phần mềm điều khiển ở đây là **COM2** mang ý nghĩa thể hiện mối liên quan trực tiếp với giao diện phần cứng "**COM2**".

PHẦN II: ĐỘNG CƠ BƯỚC.

I. Đặc điểm chung về động cơ bước

Động cơ bước thực chất là động cơ đồng bộ hoạt động dưới tác dụng của các xung rời rạc và kế tiếp nhau. Khi một xung dòng điện hoặc điện áp đặt vào cuộn dây phần ứng của động cơ bước, thì roto (phần cảm) của động cơ sẽ quay đi một góc nhất định, và được gọi là bước của động cơ, khi các xung dòng điện đặt vào cuộn dây phần ứng liên tục thì roto sẽ quay liên tục.

Vị trí của trục động cơ bước được xác bằng số lượng xung, và vận tốc của động cơ tỷ lệ với tần số xung, và được xác định bằng số bước/giây (second). Tính năng làm việc của động cơ bước được đặt trưng bởi bước được thực hiện, đặt tính góc (quan hệ của mômen điện từ theo gốc giữa trục của Roto và trục của từ trường tổng), tần số xung giới hạn sao cho các quá trình quá độ, khi hoàn thành một bước có thể tắt đi trước khi bắt đầu bước tiếp theo. Tính năng mở máy của động cơ, được đặt trưng bởi tần số xung cực đại có thể mở máy mà không làm cho Roto mất đồng bộ (bỏ bước). Tuỳ theo kết cấu của từng loại động cơ, mà tần số động cơ có thể tiếp nhận được từ 10 đến 10.000 Khz.

Bước của động cơ (giá trị của góc giữa hai vị trí ổn định kề nhau của Roto) càng nhỏ thì độ chính xác trong điều khiển càng cao. Bước của động cơ phụ thuộc vào số cuộn dây phần ứng, số cực của Stato, số răng của Roto và phương pháp điều khiển bước đủ hoặc điều khiển nữa bước. Tùy theo yêu cầu về độ chính xác và kết cấu của động cơ, mà bước của động cơ thay đổi trong giới hạn từ 1800 - 0,180. Trong đó: động cơ bước nam châm vĩnh cửu dạng cực móng và có từ trở thay đổi từ 60 - 450, động cơ bước có từ trở thay đổi có góc bước nằm trong giới hạn từ: 1,80- 300, và động cơ bước hỗn hợp có góc bước thay đổi trong khoảng 0,360 - 150. Các giá trị góc của các loại động cơ kể trên được tính trong chế độ điều khiển bước đủ.

Chiều quay của động cơ bước không phụ thuộc vào chiều dòng điện chạy trong các cuộn dây phần ứng, mà phụ thuộc vào thứ tự cuộn dây phần ứng được cấp xung điều khiển. Nhiệm vụ này do bộ chuyển phát thực hiện.

Số cuộn dây phần ứng (hay cò gọi là cuộn dây pha) của động cơ bước được chế tạo từ 2 - 5 cuộn dây pha (hay còn gọi là bối dây) và được đặt đối diện nhau trong các rãnh ở Stato. Đối với cuộn dây phải có hai cuộn dây thì chỉ dùng cho điều khiển lưỡng cực (cuộn dây có cực tính thay đổi), với 4 cuộn dây có thể dùng cho cả hai chế độ điều khiển lưỡng cực và điều khiển đơn cực

Động cơ bước là một thiết bị được sử dụng rộng rãi dùng để chuyển các xung điện thành chuyển động cơ học. Ở một số ứng dụng, chẳng hạn như bộ điều khiển đĩa, máy in kim ma trận và robot, thì động cơ bước đươc dùng để điều khiển chuyển động.

II. Nguyên lý hoạt động của động cơ bước.

1.Động cơ nam châm vĩnh cửu.

Nguyên lý làm việc của động cơ này là dựa vào tác động của một trường điện từ trên một mômen điện từ, tức là tác động giữa một trường điện từ và một hoặc nhiều nam châm vĩnh cửu. Roto của động cơ tạo thành một hoặc nhiều cặp từ và mômen điện từ của nam châm được đặt thẳng hàng trên từ trường quay do các cuộn dây tạo nên.

Động cơ có hai cuộn dây lắp ở hai cực của Stato, và một nam châm vĩnh cửu ở Roto. Khi kích thích một cuộn dây của Stato (đồng thời ngắt điện cuộn kia) sẽ tạo nên hai cực Bắc (North) và Nam (South) của nam châm. Roto sẽ thẳng đứng với hướng từ trường.

Nếu ta cho dòng điện vào cuộn dây W1 thì vị trí 1 và 3 của Stato tương ứng sẽ là cực Nam và cực Bắc.

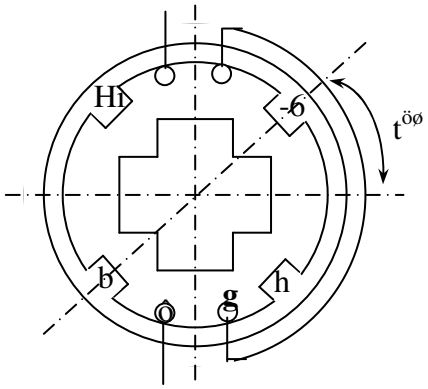
Giả sử trục của nam châm vĩnh cửu của Roto đang lệch với trục 1-3 một góc (dưới tác dụng của lực hút do các cực trái dấu của nam châm sẽ sinh ra một lực quay Roto về vị trí 1, vị trí này gọi là vị trí cân bằng. Sau đó cho dòng điện I2 vào cuộn dây W2 (lúc này dòng điện ở cuộn W1 bị ngắt), thanh nam châm sẽ quay nhanh đến vị trí 2 một góc 900 nếu việc cấp điện liên tục và tuần tự vào cuộn dây W1, W2, W1, W2, …. Và đảo chiều dòng điện sau mỗi bước, thanh nam châm sẽ quay thành những vòng tròn, từ một phần tư vòng tròn đến một phần tư vòng tròn khác.

Các cuộn dây của Stato gọi là các pha. Động cơ bước có thể có nhiều pha: 2, 3, 4, 5 pha, nó được cấp điện cuộn này sang cuộn khác với việc đảo chiều dòng điện sau mỗi bước quay. Chiều các động cơ phụ thuộc vào thứ tự cung cấp điện cho các cuộn dây và hướng của từ trường.

2. Động cơ bước từ trở thay đổi.

Nguyên lý làm việc của động cơ bước từ trở thay đổi dựa trên cơ sở định luật cảm ứng điện từ, tức là dựa trên sự tác động giữa một trường điện từ và một Roto có từ trở thay đổi theo góc quay.

Cấu trúc tiêu biểu của động cơ có bước từ thay đổi.



Roto động cơ điện được chế tạo bằng vật liệu dẫn từ, trên bề mặt Roto thường có nhiều răng. Mỗi răng của Roto hoặc của Stato gọi là một cực. Trên hai cực đối diện được mắc nối tiếp hai cuộn dây (ví dụ như cuộn dây AA') tạo thành một phần của động cơ. Như vậy động cơ như hình vẽ có ba pha A, B, C, từ trở thay đổi theo góc quay của răng. Khi các răng của Roto đứng thẳng hàng với các cực của Stato, từ trở ở đó sẽ nhỏ nhất. Nếu ta cho dòng điện chạy vào cuộn dây BB' nó sẽ tạo nên từ trường kéo cực gần nó nhất của roto và làm Roto quay một góc 300 theo chiều ngược chiều kim đồng hồ. Nếu dòng điện được đưa vào cuộn dây CC', Roto lại tiếp tục quay một góc 300 nữa … Các cuộn dây AA', BB', CC' gọi là các pha.

Hướng quay của động cơ không phụ thuộc vào chiều của dòng điện mà phụ thuộc vào thứ tự cấp điện cho cuộn dây. Nhiệm vụ này do các mạch logic trong bộ chuyển phát thực hiện. Với cách thay đổi thứ tự hoặc thay đổi cách kích thích các cuộn dây ta cũng làm thay đổi các vị trí góc quay.

Động cơ bước có từ trở thay đổi có chuyển động êm, số bước lớn và tần số làm việc cũng khá lớn (từ 2 đến 5 Khz).

Một số công thức tính cho động cơ bước từ trở thay đổi

Nr : Số răng của roto.
Ns : Số răng của stato.
Np : Số pha.
Pr : Góc độ răng roto.
ϕs : Góc bước.
Rs : Giá trị bước.
X = Ns/Np : Số răng stato cho một pha.

Góc độ răng giữa hai răng kề nhau được xác định như sau

$$\Pr = \frac{369^0}{Nr} \quad ; \quad Ps = \frac{360^0}{\Pr}$$

Góc bước được xác định bởi biểu thức:

$$\phi s = \frac{360^0}{Nr.Np} \text{ (độ/bước)}$$

Giá trị bước

$$Rs = \frac{360}{\phi s}$$

Nếu tần số xung là f thì tốc độ Roto là:

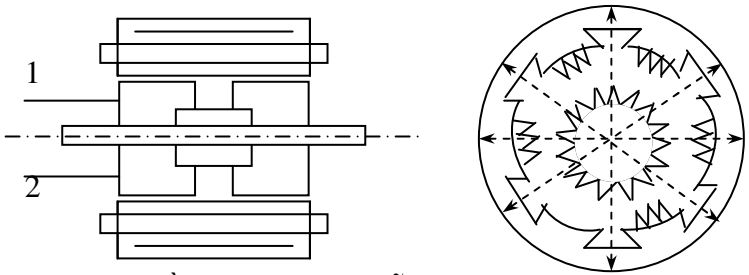
$$\psi = \frac{69f}{Rs} = \frac{60f}{Np.Nr} = \frac{\phi s}{6} \text{(vòng/phút)}$$

Số răng Roto cho một pha:

$$X = \frac{Nr}{Np+1}$$

3.Động cơ hỗn hợp.

Động cơ hỗn hợp là sự kết hợp nguyên tắt làm việc của động cơ có bước nam châm vĩnh cửu và động cơ bước có từ trở thay đổi nhằm có được đặt tính tốt nhất của hai loại kể trên là momen lớn và số bước lớn.



Sơ đồ động cơ bước hỗn hợp.

Động cơ gồm hai nửa Roto (1) và (2). Nửa(1) có nhiều răng trên Roto, nửa (2) là nam châm vĩnh cửu. Do đó có sự kết hợp giữa hai phần nên tạo ra sự kích thích roto mạnh hơn. Động cơ loại này có số bước đạt đến 400 bước, nhưng giá thành đắt.

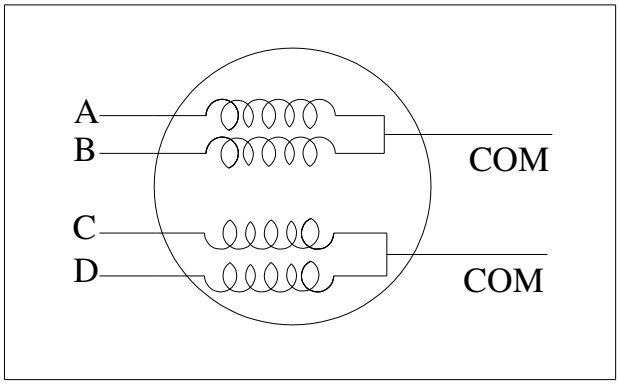
III. Các phương pháp điều khiển động cơ bước.

Mỗi động cơ bước đều có phần quay roto là nam châm vĩnh cửu, được bao xung quanh là phần tĩnh, gọi là stato. Động cơ bước có 4 cuộn dây stato được sắp xếp theo cặp qua tâm đối xứng. Động cơ bước dạng này gọi là động cơ bước 4 pha. Điểm giữa cho

phép thay đổi chiều dòng điện của một trong hai lõi khi một cuộn dây được nối đất, do đó đổi cực của stato. Lưu ý rằng, trục của động cơ thông thường thì quay tự do, còn trục động cơ bước thì quay theo từng bước cố định, lặp lại và đếm từng vị trí cụ thể. Động cơ quay như vậy là từ cơ sở lý thuyết từ trường: các cực cùng dấu đẩy nhau và các cực ngược dấu hút nhau. Chiều quay được xác định bởi từ trường của stato, mà từ trường này thì do dòng điện chạy qua lõi cuộn dây gây nên. Khi hướng của dòng thay đổi thì cực của từ trường cũng thay đổi theo, gây ra chuyển động ngược lại của động cơ (đảo chiều). Động cơ bước ở đây có 6 đầu dây: 4 đầu của cuộn dây stato và 2 đầu dây chung điểm giữa của các cặp dây. Khi chuỗi xung nguồn được cấp đến từng cuộn dây stato thì động cơ sẽ quay. Mỗi chuỗi xung có thể có cấp độ chính xác khác nhau. Bảng sau giới thiệu chuỗi 4 bước thông thường.

Bảng 1

Chiều kim đồng hồ	Bước	Cuộn dây A	Cuộn dây B	Cuộn dây C	Cuộn dây D	Chiều quay bộ đếm
↓	1	1	0	0	1	↑
	2	1	1	0	0	
	3	0	1	1	0	
	4	0	0	1	1	



Hình1: **Bố trí cuộn dây Stato.**

Lưu ý chúng ta có thể bắt đầu với chuỗi xung nào đó trong bảng 1, song khi đã bắt đầu bằng chuỗi xung nào thì cần phải tiếp tục theo đúng thứ tự của chuỗi xung đó. Ví dụ, nếu bắt đầu bằng bước thứ 3 là chuỗi (0110) thì cần tiếp tục với chuỗi của bước 4 rồi sau đó lặp lại 1,2,3 v.v.

Một số thông số và khái niệm.

*Góc bước (step Angle):*Câu hỏi đặt ra là mỗi bước có độ dịch chuyển là bao nhiêu? Điều này phụ thuộc vào cấu trúc bên trong của động cơ, đặc biệt là số răng của stato và roto. Góc bước là độ quay nhỏ nhất của một bước. Các động cơ khác nhau có góc bước khác nhau. Bảng 2 giới thiệu góc bước của một số chủng loại động cơ, trong đó có dùng thuật ngữ số bước trong một vòng. Đây là tổng số bước cần để quay hết một vòng 360^0

Bảng 2: Góc bước của động cơ.

Góc bước	Số bước/vòng
0.72	500
1.8	200
2.0	180
2.5	144
5.0	72
7.5	48
15	24

Dường như trái với ấn tượng ban đầu, động cơ bước không cần nhiều đầu dây ở cuộn stato để có góc bước nhỏ hơn. Tất cả động cơ bước nói ở phần này chỉ dùng 4 đầu ở cuộn dây stato và 2 đầu dây chung ở nút giữa. Mặc dù nhiều hang sản xuất chỉ dùng một đầu dây chung, song vẫn phải có 4 đầu dây stato.

Quan hệ số bước/giây và số vòng quay/phút RPM.

Quan hệ giữa số vòng quay/phút RPM với số bước của một vòng quay và số bước/giây là quan hệ trực quan và được biểu diễn như sau:

Số bước trong giây = RPM x số bước trong vòng quay/60

Chuỗi xung bốn bước và số răng trên roto.

Chuỗi xung chuyển mạch trình bày ở bảng 1 được gọi là chuỗi chuyển mạch 4 bước, bởi vì sau 4 bước thì hai cuộn dây giống nhau sẽ được bật lên ON. Vậy sau 4 bước này động cơ quay được bao nhiêu? Sau khi thực hiện xong 4 bước thì roto chỉ quay, được một bước răng. Do vậy, ở động cơ 200 bước/vòng thì roto có 50 răng vì 50*4=200 bước cần để quay hết một vòng. Như vậy, có thể kết luận là góc bước tối thiểu luôn là hàm số của răng trên roto. Nói cách khác răng càng nhỏ thì roto quay được càng nhiều răng.

Ngoài chuỗi xung 4 bước còn có chuỗi xung 8 bước, chuỗi này còn được gọi là (half-stepping), vì ở chuỗi 8 bước thì mỗi bước là một nữa của góc bình thường.

Bảng 3: Chuỗi xung 8 bước.

Chiều kim đồng hồ	Bước	Cuộn dây A	Cuộn dây B	Cuộn dây C	Cuộn dây D	Chiều quay bộ đếm
↓	1	1	0	0	1	↑
	2	1	0	0	0	
	3	1	1	0	0	
	4	0	1	0	0	
	5	0	1	1	0	
	6	0	0	1	0	
	7	0	0	1	1	
	8	0	0	0	1	

Tốc độ động cơ.

Tốc độ động cơ được đo bằng số bước trong một giây là một hàm của tốc độ chuyển mạch.Bằng việc thay đổi thời gian ta có thể đạt được các tốc độ quay khác nhau.

Moment giữ.

Moment giữ được định nghĩa là lượng moment ngoài cần thiết để làm quay trục động cơ từ vị trí giữ của nó với điều kiện trục động cơ đang đứng yên hay đang quay với tốc độ RPM = 0. Đại lượng này được đo bằng tỷ lệ diện áp và dòng cấp đến động cơ. Đơn vị của moment giữ là kilogram – centimet.

Chuỗi 4 bước điều khiển dạng sóng.

Ngoài chuỗi 4 bước và 8 bước nói trên, còn có một chuỗi khác được gọi là chuỗi 4 bước dạng sóng. Chuỗi này được giới thiệu ở bảng 4. Chuỗi 8 bước trình bày ở bảng 3 là sự kết hợp đơn giãn của các chuỗi 4 bước thường và chuỗi 4 bước điều khiển dạng sóng cho ở bảng 1 và bảng 4.

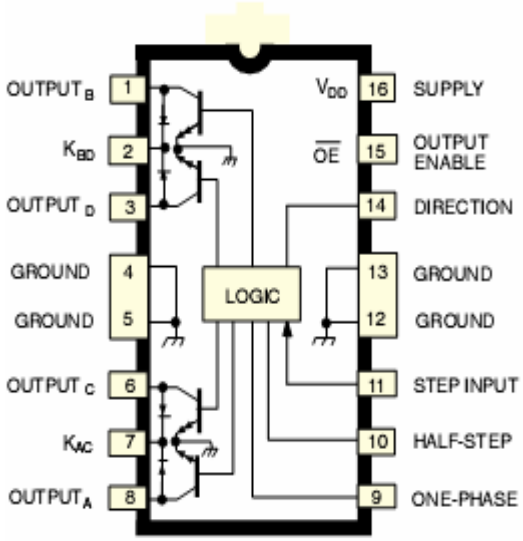
Bảng 4: Chuỗi xung 4 bước.

Chiều kim đồng hồ	Bước	Cuộn dây A	Cuộn dây B	Cuộn dây C	Cuộn dây D	Chiều quay bộ đếm
↓	1	1	0	0	0	↑
	2	0	1	0	0	
	3	0	0	1	0	
	4	0	0	0	1	

PHẦN III: CẤU TRÚC CHỨC NĂNG CỦA CÁC CON VI MẠCH.

I. Chip UCN5804B.

Đây là chip trung tâm điều khiển động cơ bước, và điều khiển động có bước theo phước pháp 4 bước thông dụng(hay gọi là phương pháp bước đủ).



Sơ đồ chân của chip UCN5804B

Chân 9,10,14,15 của chip dùng để điều khiển ngõ ra theo một chu kỳ, bước đứ, haff-step và điều khiển hoạt động ngõ ra.

Chip này hoạt động hay không phụ thuộc vào tín hiệu xung ngõ vào trên chân 11 để thúc đẩy động cơ làm việc. Tín hiệu này có thể được cấp nguồn bên ngoài hay từ công tắc tác động bằng tay.

❋**Đặc điểm của chip điều khiển này là:**

- Dòng điện lớn nhất ở ngõ ra là 1.5A.
- Điện áp chịu đựng ở ngõ ra là 35V.
- Tiêu chuẩn hoạt động bước đủ, nữa bước và bước điều khiển dạng sóng.
- Sử dụng các Diod để bảo vệ bên trong chip.
- Khả năng điều khiển ngõ ra OE và điều khiển trực tiếp.
- Reset lại năng lượng.
- Có mạch tản nhiệt tích hợp bên trong.

Bảng để điều khiển kiểu hoạt động.

Drive Format	Pin 9	Pin 10
Two-Phase	L	L
One-Phase	H	L
Half-Step	L	H
Step-Inhibit	H	H

❋ **Các kiểu hoạt động của chip**.

WAVE-DRIVE SEQUENCE

Half Step = L, One Phase = H				
Step	**A**	**B**	**C**	**D**
POR	ON	OFF	OFF	OFF
1	ON	OFF	OFF	OFF
2	OFF	ON	OFF	OFF
3	OFF	OFF	ON	OFF
4	OFF	OFF	OFF	ON

TWO-PHASE DRIVE SEQUENCE

Half Step = L, One Phase = L				
Step	**A**	**B**	**C**	**D**
POR	ON	OFF	OFF	ON
1	ON	OFF	OFF	ON
2	ON	ON	OFF	OFF
3	OFF	ON	ON	OFF
4	OFF	OFF	ON	ON

HALF-STEP DRIVE SEQUENCE

Half Step = H, One Phase = L				
Step	A	B	C	D
POR	ON	OFF	OFF	OFF
1	ON	OFF	OFF	OFF
2	ON	ON	OFF	OFF
3	OFF	ON	OFF	OFF
4	OFF	ON	ON	OFF
5	OFF	OFF	ON	OFF
6	OFF	OFF	ON	ON
7	OFF	OFF	OFF	ON
8	ON	OFF	OFF	ON

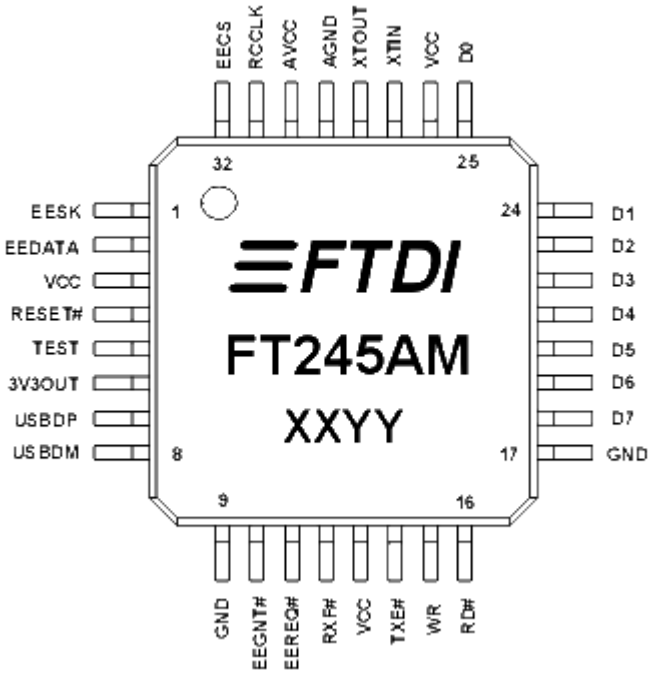
II. Chip FTDI- FT245AM.

1.Tổng quan về chip FTDI-FT245AM.

Đây là chip để chuyển dữ liệu từ kiểu nối tiếp của cổng USB sang truyền song song để điều khiển động cơ bước thông qua chip 5804BM. Với các dắc tính sau:

- 3.3V LDO điện áp điều chỉnh: là máy điều chỉnh điện áp LDO 3.3V trong quá trình USB truyền dữ liệu từ bộ nhớ đệm ra ngõ ra.
- USB truyền nhận: Khối truyền nhận USB cung cấp 1 giao diện vật lý của USB 1.1 đến cáp USB. Tại ngõ ra cung cấp mức điện áp 3.3V để điều khiển tín hiệu, trong khi một máy thu khác và 2 tín hiệu kết thúc từ máy thu cung cấp dữ liệu từ cổng USB, SEO và USB dò tìm điều kiện.
- USB DPLL: Khối này dễ dàng nhận ra các bit mã hóa NRZI dữ liệu từ USB cung cấp đến, và tự tạo ra xung clock và dữ liệu đến SIE block.
- Dao động 6 Mhz: Dao ddoognj tạo ra từ dao động thạch anh ngoài chip hay từ hiện tượng cộng hưởng của các phần tử.
- SIE (Serial Interface Engine): đảm nhận nhiệm vụ truyền nhận dữ liệu.
- USB Protocol Engine: quản lý luồng dữ liệu từ thiết bị điều khiển USB.
- Fifo Receive Buffer (128 byte): Bộ đệm USB nhận dữ liệu tối đa 128 byte, dữ liệu được đọc và ghi từ bộ nhớ FIFO thông qua chân nối tiếp truyền thông dữ liệu USB.
- 384 byte bộ nhớ nhận dữ liệu, 128 byte bộ nhớ truyền tín hiệu mức cao
-

2.Sơ đồ chân.



Chân	Tín hiệu	Loại	Chức năng
7	USBDP	I/O	Tín hiệu dữ liệu USB phụ thuộc váo trở 1.5k để giảm điện áp ngõ ra còn 3.3V
8	USBDM	I/O	Kết nối chân dữ liệu âm của USB
6	3.3VOUT	OUT	Chân tạo điện áp 3.3V tại ngõ ra
27	XTIN	IN	Lấy tín hiệu từ dao động 6Mhz từ dao động thạnh anh
28	XTOUT	OUT	Đưa tín hiệu 6Mhz ra chân dao động thạch anh
31	RCCLK	I/O	RC-Timer đảm bảo xung clock trên mode Sleep và tác động mức Low trong suốt qua trình chọn mode Reset hay Sleep
4	RESET	IN	Reset toàn bộ thiết bị sử dụng network RC ngoài
32	EECS	I/O	Tùy chọn EEPROM-chip chọn
1	EESK	I/O	Tùy chọn EEPROM-xung đồng hồ
2	EEDATA	I/O	Tùy chọn EEPROM-I/O dữ liệu
5	TEST	IN	Đặt thiết bị vào mode Text- thiết bị phải được nối đất.
25	D0	I/O	Bus dữ liệu truyền trực tiếp Bit0
24	D1	I/O	Bus dữ liệu truyền trực tiếp Bit1
23	D2	I/O	Bus dữ liệu truyền trực tiếp Bit2
22	D3	I/O	Bus dữ liệu truyền trực tiếp Bit3
21	D4	I/O	Bus dữ liệu truyền trực tiếp Bit4
20	D5	I/O	Bus dữ liệu truyền trực tiếp Bit5
19	D6	I/O	Bus dữ liệu truyền trực tiếp Bit6
18	D7	I/O	Bus dữ liệu truyền trực tiếp Bit7

16	RD#	IN	Byte dữ liệu D0….D7được tác động mức thấp bởi dữ liệu truyền từ FIFO
15	WR	IN	Ghi byte dữ liệu trên D0…..D7 vào FIFO truyền khi WR được tác động cao $\rightarrow$thấp
14	TXE#	OUT	Khi tác động high, không dữ liệu vào FIFO. Khi low dữ liệu có thể được ghi vào FIFO.
12	RXF#	OUT	Khi ở mức cao, không đọc dữ liệu từ FIFO. Khi ở mức thấp có thể được đọc bởi RD# sau đó chuyển từ thấp sang mức cao.
11	EEREQ#	IN	Yêu cầu EEPROM kết nối để xử lý qua bus dữ liệu
10	EEGN#	OUT	Ở mức thấp, cho phép EEPROM nối để xử lý qua bus dữ liệu
3,13,26	V_{CC}	PWR	Thiết bị chuyển điện áp từ 4.4V – 5.25V
9,17	GND	PWR	
30	AV_{CC}	PWR	Thiết bị nguồn xung clock

PHẦN IV: NGUYÊN LÝ HOẠT ĐỘNG CỦA MẠCH.

Nguồn được cấp bởi một nguồn DC bên ngoài hay nguồn DC P1. Điện áp có thể dao động từ 6$\rightarrow$30V, phụ thuộc vào tốc độ của động cơ bước. Động cơ bước sử dụng hầu hết các dong trong mạch, vì vậy nó được cấp nguồn thông qua trở R1,R2. Đây là trở giới hạn dòng đến động cơ và cho phép động cơ hoạt động với nguồn điện áp cung cấp lớn hơn điện áp giới hạn cho phép động cơ làm việc.

Điện áp ổn định ở trạng thái (không tải) của mạch thu được bằng cách điều chỉnh điện áp ngõ và dưới 5V với con chip điều chỉnh điện áp LM78L05. Đây là một IC điều chỉnh điện áp, với C7, C1, C5 cung cấp thêm vào điện áp lọc.

U1-UCN5804BM có chức năng cung cấp dòng 100mA, đây là trung tâm bộ điều khiển động cơ bước làm việc theo từng phương pháp điều khiển. Nó gồm 1 CMOS xắp xếp và lưu giữ mức logic phần điện áp lưỡng cức ngõ ra để trực tiếp điều khiển động cơ. UNC5804BM có thể phát tín hiệu điều khiển với 3 mode khác nhau: FULL-STEP, HALF-STEP, FULL-STEP WAVE.

D1-4 là một tổ hợp Diod để ngăn cản nguy hiểm đến chip UNC5804BM nếu công tắc ngõ ra ở mức thấp khi động cơ đang hoạt động với tải lớn.

Mỗi LED1-4 sáng tương ứng với ngõ ra làm việc ở mức thấp và là vật có ích để nhận xét trạng thái làm việc của ngõ ra. Trở R3 cung cấp dòng điện giới hạn đến LED.

PIN 9,10,14 và 15 của chip UCN5804BM để điều khiển ngõ ra theo các phương pháp điều khiển tùy chọn. Công tắc DIP1-4 cho phép điều khiển bằng tay mỗi hướng làm việc của động cơ bước theo bảng trạng thái sau:

FUNCTION	S4-PIN 9	S3-PIN 10	FUNCTION	S2-PIN 14	S1-PIN 15
FULL-STEP 2-PHASE	L	L	FORWARD	L	
FULL-STEP WAVE DRIVE	H	L	REVERSE	H	
HALF-STEP	L	H	OUTPUT ENABLED		L
STEP INHIBIT	H	H	OUTPUT DISABLED		H

FIGURE 6

Ở đây ta dùng mạch ghép nối để diều khiển động cơ bước hoạt động bằng máy tính, UCN5804BM phụ thuộc vào chân 11 có được tác động hay không để thúc đẩy động cơ làm việc. Tín hiệu này được cấp từ chip FT245-USB.

Chip FT245_USB chuyển tín hiệu lấy ra từ cổng USB sang dạng tín hiệu truyền song song 8 bit dữ liệu tác động đến chip UCN5804BM để điều khiển động cơ bước làm việc theo các mode định sẵn và tốc độ quy định. Ta chỉ cần truyền dữ liệu vào các chân điều khiển của chip UCN5804BM theo sơ đồ kết nối để điều khiển động cơ làm việc theo 1 trong 3 phương pháp trên.

Đề tài 4: **Thiết kế mạch cổng nối tiếp RS232 mạch này làm nhiệm vụ đo ánh sáng và điều khiển đèn.**

Chương I: Giới thiệu chung về truyền thông nối tiếp

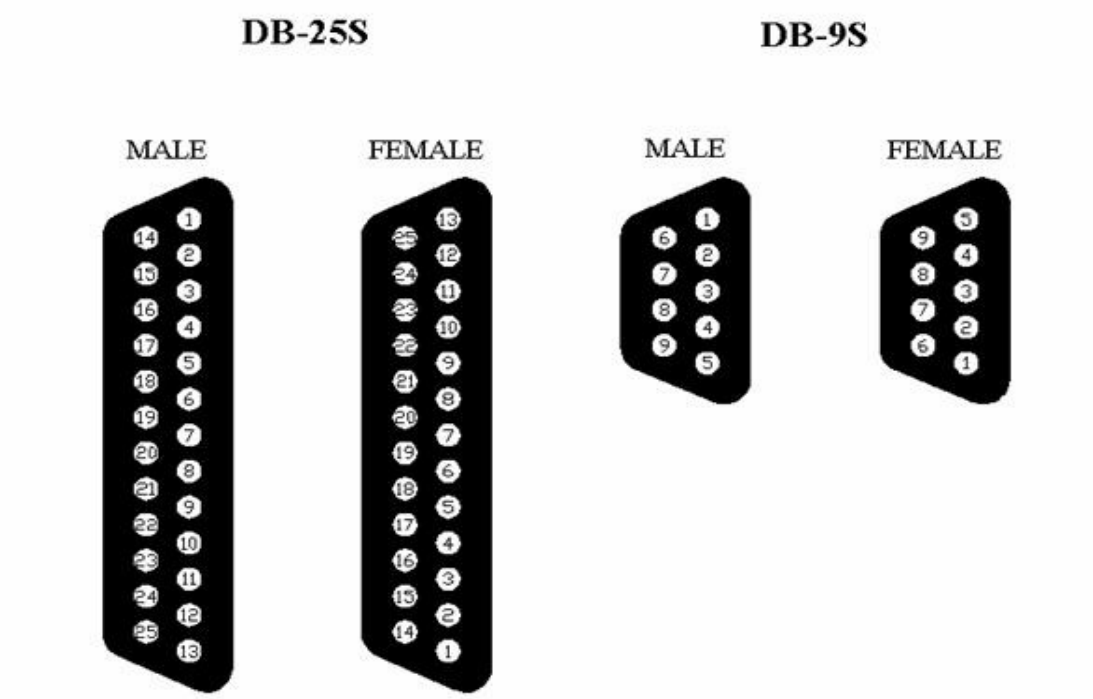
Hiện nay các tiêu chuẩn truyền thông cũng theo hai hướng phân biệt nhau dựa vào cách truyền : song song hay nối tiếp. Cách truyền song song rất dễ bị nhiễu tác động nên không thể truyền đi xa được, do đó cũng ít được sử dụng. Truyền nối tiếp cũng có 2 loại : đồng bộ hay không đồng bộ. Trong cách truyền đồng bộ, dãy ký tự được truyền sẽ kèm theo ký tự đồng bộ là SYN (mã ASCII là 22). Phương thức này cho tốc độ truyền khá cao nhưng do mạch xử lý truyền và nhận (bao gồm mạch thêm ký tự đồng bộ, phát hiện và báo sai…) khá phức tạp nên chỉ dùng trong các ứng dụng có yêu cầu cao về tốc độ truyền. Còn trong các ứng dụng thông thường, nhất là các ứng dụng trong lĩnh vực điều khiển tự động, thì không có yêu cầu về tốc độ mà yêu cầu về độ tin cậy nhưng mạch thực hiện đơn giản, rẻ tiền. Khi đó, cách truyền không đồng bộ rất phù hợp. Theo cách truyền này thì các ký tự được truyền riêng rẽ, phân làm từng frame có bit bắt đầu, các bit dữ liệu của ký tự cần truyền, bit chẵn lẻ (để kiểm tra lỗi đường truyền), và các bit kết thúc.Trong khuôn khổ Luận Văn này ,ta chỉ sử dụng chuẩn truyền thông RS-232C (RS :Recommended Standard) ,là một chuẩn truyền nối tiếp bất đồng bộ rất phổ biến hiện nay.

Chuẩn này lần đầu tiên được giới thiệu vào năm 1962 do hiệp hội kỹ thuật điện tử EIA (Electronics Industries Association) đưa ra như là chuẩn giao tiếp truyền thông giữa máy tính và thiết bị ngoại vi như :modem, máy vẽ, mouse, máy tính khác ……

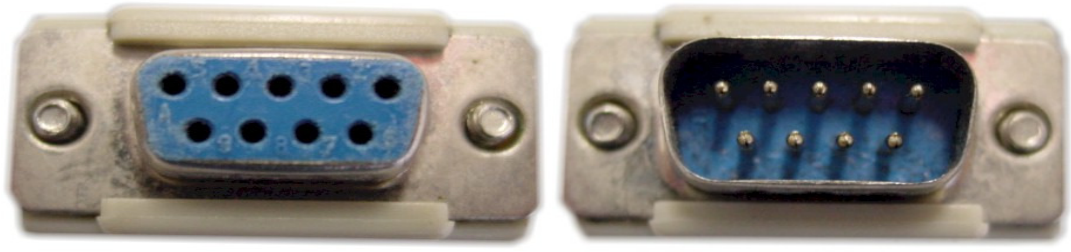
1. Cấu tạo cổng nối tiếp:

Bảng 1.1 Sắp xếp chân của cổng nối tiếp ở máy tính:

9 chân	25 chân	Chức năng
1	8	DCD _ Data Carrier Detect (Lối
2	3	vào)
3	2	RxD _ Receive Data (Lối vào)
4	20	TxD _ Transmit Data (Lối ra)
5	7	DTR _ Data Terminal Ready (Lối
6	6	ra)
7	4	GND _ Ground (Nối đất)
8	5	DSR _ Data Set Ready (Lối vào)
9	22	RTS _ Request to Send (Lối ra)
		CTS _ Clear to Send (Lối vào)
		RI _ Ring Indicator (Lối ra)



.



2. Các Chuẩn Truyền Nối Tiếp :

Ở dạng nối tiếp ta có rất nhiều chuẩn truyền như: RS-232, RS-422, RS-423, RS-449, RS-485 … do không đi sâu vào các chuẩn truyền này nên ta chỉ khảo sát sơ 2 chuẩn truyền RS-232 và RS-485 :

Bảng 1.2. So sánh các tiêu chuẩn truyền EIA:

Thông số	RS_232	RS_422	RS_423	RS_485
Cable length(max)	15m (50FT)	1.2km(4000FT)	1.2km(4000FT)	1.2km(4000FT)
Baud Rate (tốc độ baud)	20Kps/15m	10Mbps/12m 10Mbps/120m 100Kbps/1.2km	100Kbs/9m 10Kbps/90m 1Kbps/1.2km	10Mbps/12m 1Mbps/120m 100Kbps/1.2km

Mode	Unbalanced	Balanced Differental	Unbalnced Differental	Balanced Differental
Driver No	1	1	1	32
Receiver No	1	10	10	32
Logic 0	+5V÷+15 V	+2v ÷ +5V	+3.6V ÷ +6V	+1.5V ÷ +5V
Logic 1	-5V÷-15V	-2V ÷ -5V	-3.6V÷-6V	-1.5V÷-5V
Community (truyền thông)	2V	1.8V	3.4V	1.3V
Cable/Signal (Cáp / tín hiệu)	1	2	2	2
Methode (Phương thức)	Simplex Half_duplex Full_duplex	Simplex Half_duplex Full_duplex	Simplex Half_duplex Full_duplex	Simplex Half_duplex Full_duplex
Short circuit current	500mA	150mA	150mA	150mA

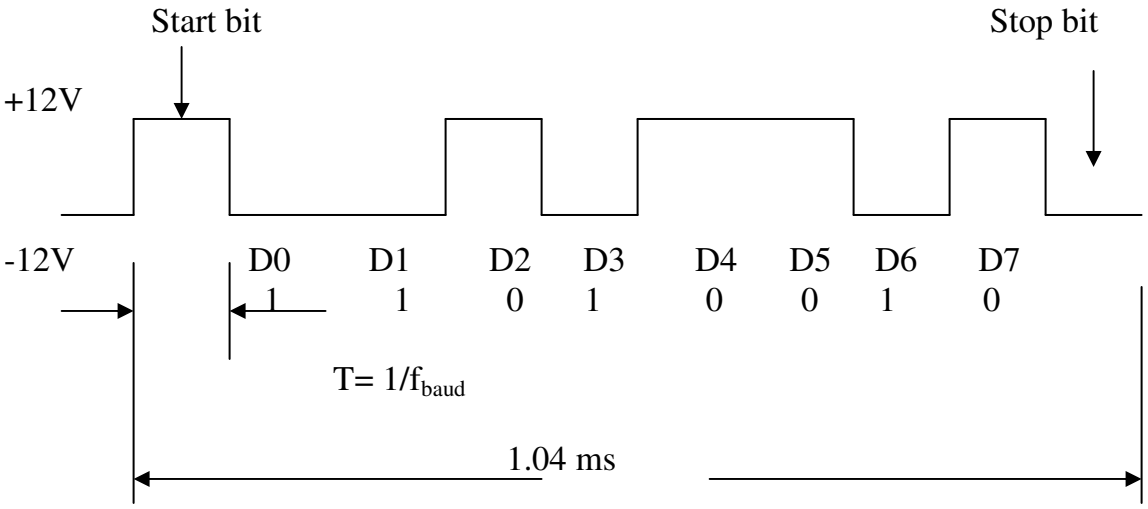
3. Phương thức truyền dữ liệu của RS-232 :

Dữ liệu trao đổi diễn ra trên hai đường dẫn TxD và RxD. Qua chân cắm ra TxD, máy tính gởi dữ liệu của nó đến các thiết bị khác. Trong khi đó dữ liệu mà máy tính nhận được, lại được dẫn đến chân nối RxD. Các tín hiệu khác đóng vai trò như là tín hiệu hỗ trợ khi trao đổi thông tin và vì vậy không phải trong mọi ứng dụng đều dùng đến.

Các bit dữ liệu được gởi đi theo kiểu đảo ngược, nghĩa là các bit có giá trị "1" sẽ có mức điện áp LOW, các bit có giá trị "0" sẽ có mức điện áp HIGH. Mức tín hiệu nhận và truyền qua chân RxD và TxD thông thường nằm trong khoảng –12V đến +12V. Mức điện áp đối với mức HIGH nằm giữa +3V đến +12V.

Một chuỗi dữ liệu truyền đi theo dạng nối tiếp nhau trên một đường dẫn: bắt đầu bằng một bit khởi đầu (Start bit), tiếp theo đó là các bit dữ liệu (data bit), bit thấp đi trước. Số bit dữ liệu nằm trong khoảng 5 đến 8 bit, tiếp đó là bit kiểm tra chẳn lẻ (Parity) và cuối cùng là bit kết thúc (stop bit). Hình thức truyền này có khả năng dùng cho những khoảng cách lớn , bởi vì các khả năng gây nhiễu là nhỏ hơn là dùng cổng song song. Tốc độ truyền được thiết lập bằng tham số Baudrate, là số bit truyền đi trong 1 giây, thông thường là 300, 600, 1500, 2400, 4800, 9600 và 19200.

Một nhược điểm không nhỏ của cổng nối tiếp là 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 gởi cũng như nhận.



Dòng dữ liệu trên cổng RS_232 với tốc độ baud 9600baud.

Một trong những yêu cầu quan trọng của RS-232 là thời gian chuyển đổi từ một mức logic này tới mức logic khác không vượt qúa 4% thời gian 1 bit. Vì thế ở tốc độ 19200 thời gian chuyển mức logic phải nhỏ hơn .

Vấn đề này làm giới hạn chiều dài đường truyền . Với tốc độ truyền 19200 baud có thể truyền xa nhất là 50ft (1ft = 30.48cm, 15.24cm)

Một trong những vấn đề quan trọng cần chú ý khi sử dụng RS-232 là mạch thu phát không cân bằng (đơn cực). Điều này có nghĩa là tín hiệu vào được so với đất. Vì vậy, nếu điện thế tại hai điểm đất của hai mạch thu phát không bằng nhau thì sẽ có dòng điện chạy trên đất. Kết quả sẽ có áp rơi trên dây đất (V= I.R) sẽ làm suy yếu tín hiệu logic. Nếu truyền tín hiệu đi xa , R sẽ tăng dẫn đến áp rơi trên đất sẽ lớn dần đến lúc tín hiệu logic sẽ rơi vào vùng không xác định và mạch thu sẽ không nhận đúng dữ liệu được truyền từ mạch phát. Chính sự không cân bằng trên mạch thu phát là một trong những nguyên nhân giới hạn đường truyền.

Chương II : Giới thiệu các linh kiện dùng trong mạch

I. Chip ADC804:

Chíp ADC804 là bộ chuyển đổi tương tư số thuộc họ ADC804 của hãng national semiconductor. Chíp này cũng được nhiều hãng khác sản xuất. Điện áp nuôi là +5v và độ phân giải 8 bit. Thời gian chuyển đổi được định nghĩa là thời gian mà bộ ADC cần đề chuyển môt đầu vào tương tự thành một số nhị phân. Đối với ADC 804 thời gian chuyển đổi phụ thuộc vào tần số đồng hộ được cấp tới chân CLK va CLKIN và không bé hơn 110μs.

CS chọn chip

Là chân chọn chíp, đầu vào tích cực mức thấp được sử dụng để kich hoạt chíp ADC804. Để tri cập ADC804 thì chân này ở mức thấp.

RD - Đọc

Đây là một tín hiệu vào tích cực mức thấp. các bộ ADC chuyển đổi đầu vào tương tự thành số nhị phân và giữ nó ở một thanh ghi trong. RD được sử dụng để có dữ liệu được đã chuyển đổi tới đầu ra của ADC 804. khi CS=0 nếu có một xung cao xuông thấp áp đến chân RD thì dữ liệu ra dạng số 8 bit được đưa tới các chân dữ liệu D0-D7.

WR - Ghi

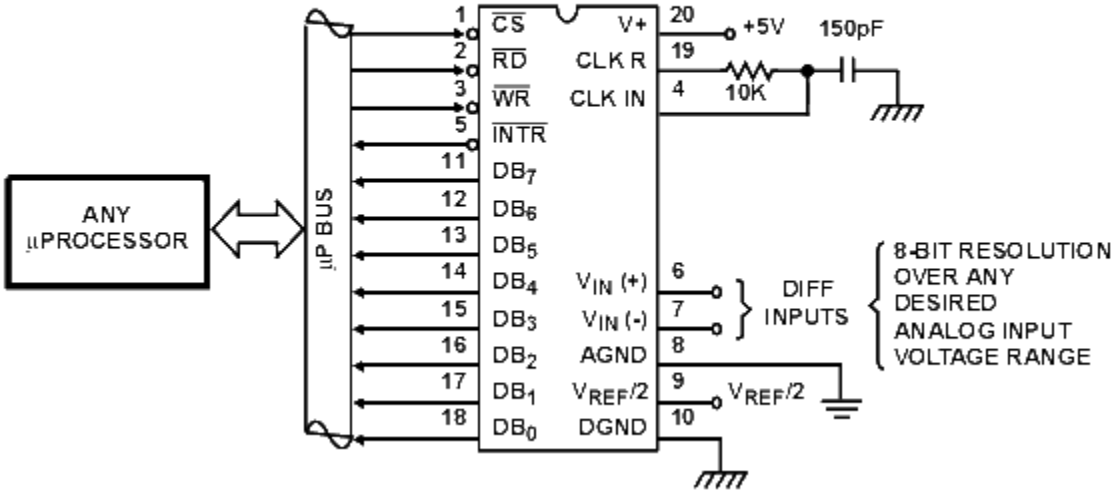
Đây là chân vào tích cực mức thấp được dung để báo cho ADC 804 bắt đầu quá trình chuyển đổi. Nếu CS=0 khi WR tạo ra xung cao xuống thấp thì bộ ADC 804 bắt đầu tiến hành chuyển đổi giá thị đầu tương tự V_{in} về số nhị phân 8 bit.

CLK IN và CLK R

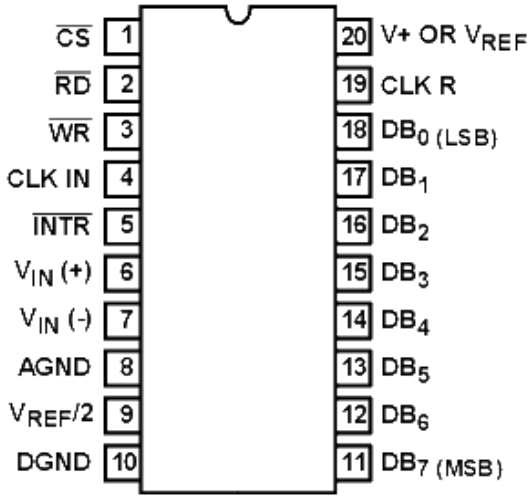
CLK IN là chân vào nối tới đồng hồ ngoài khi động hồ ngoài được sử dụng để tạo thời gian. Tuy nhiên 804 cũng có mộ bộ tạo xung đồng hồ trên chíp để dùng đồng hồ trên chíp của 804 thì chân CLK IN và CLK – R được nối tới tụ điện và một điện trở. với tần số

$$f = \frac{1}{1.1RC}$$

Typical Application Schematic



ADC0802, ADC0803, ADC0804
(PDIP, CERDIP)
TOP VIEW



Ngắt INTR

Ngắt hay còn gọi là kết thúc việc chuyển đổi. đây là chân ra tích cực mức thấp. Bình thường chân này ở trạng thái cao và khi việc chuyển đổi hoàn tất thì nó xuống thấp để báo cho CPU biết là dữ liệu được chuyển đổi đã sẵn sàng để lấy đi. Sau khi INTR xuống thấp, cần đặt CS=0 và gửi một xung cao xuống thấp tới chân RD để đưa dữ liệu ra.

Vin (+) và Vin(-)

Đây la hai đầu vào tương tự vi sai, Trong đó $V_{in} = V_{in}(+) - V_{in}(-)$

$V_{in}(-)$ được nối xuống thấp

V_{in} (+) được dùng làm dầu vào tương tự và được chuyển đổi về dạng số.

Vcc

Là chân nguồn nuôi.

Vref/2

Chân 9 là điện áp đầu vàođược dùng là điện áp tham chiếu.

D0- D7

D0-D7 là các chân ra dữ liệu Các chân nay được đệm ba trang thái và dữ liệu đã được chuyển đổi chỉ được truy cập khi chân CS = 0 và chân RD đưa xuông mức thấp.

$$D_{out} = \frac{Vin}{kichthuocbuoc}$$

Chân đất tương tự và chân đất số.

Đây là những chân đầu vào cấp đất cho cả tính hiệu tương tự và số, Tương tự nối toi Vin và chân đất số nối tới Vcc.

II. Cảm biến nhiệt độ (họ LM35):

Bộ cảm biến (Transducer) chuyển đổi các đại lượng vật lý, ví dụ như nhiệt độ, cường độ ánh sáng, lưu tốc và tốc độ thành các tín hiệu điện. Phụ thuộc vào bộ cảm biến mà đầu ra có thể là tín hiệu dạng điện áp,dòng, trở kháng hay dung kháng. Bộ cảm biến nhiệt đáp ứng sự thay đổi nhiệt độ bằng cách thay đổi trở kháng, song đáp ứng này không tuyến tính.

Bộ cảm biến nhiệt độ LM35 của hãng National Semiconductor Corp là bộ cảm biến nhệt tuyến tính đơn giản và dễ được sử dụng rộng rãi.

LM35 là họ cảm biến nhiệt mạch tích hợp chính xác cao có điện áp đầu ra tỷ lệ tuyến tính với nhiệt độ theo thang độ Celsius.

Nhiệt độ (0^0)	Trở kháng của cảm biến (K Ω)
0	29.49
25	10
50	3.893
75	1.7
100	0.817

Trở kháng của bộ cảm biến nhiệt theo nhiệt độ

Họ cảm biến này không yêu cầu cân chỉnh ngoài vì vốn nó đã được cân chỉnh . Họ này cho ra điện áp 10mV ứng với thay đổi nhiệt độ là 1^0C. Bảng sau giới thiệu một số thông số kỹ thuật chính của họ LM35.

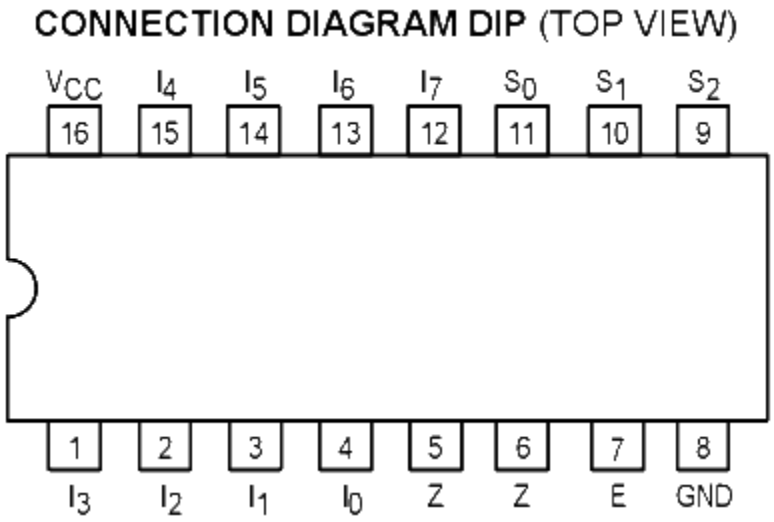
Mã sản phẩm	Dải nhiệt độ	Độ chính xác	Đầu ra
LM35A	-55^0C tới $+150^0$C	+ 1^0C	10mV/F

LM35	-55^0C tới $+150^0$C	$+ 1,5^0$C	10mV/F
LM35CA	-40^0C tới $+110^0$C	$+ 1^0$C	10mV/F
LM35C	-40^0C tới $+110^0$C	$+ 1,5^0$C	10mV/F
LM35D	0^0C tới $+100^0$C	$+ 12^0$C	10mV/F

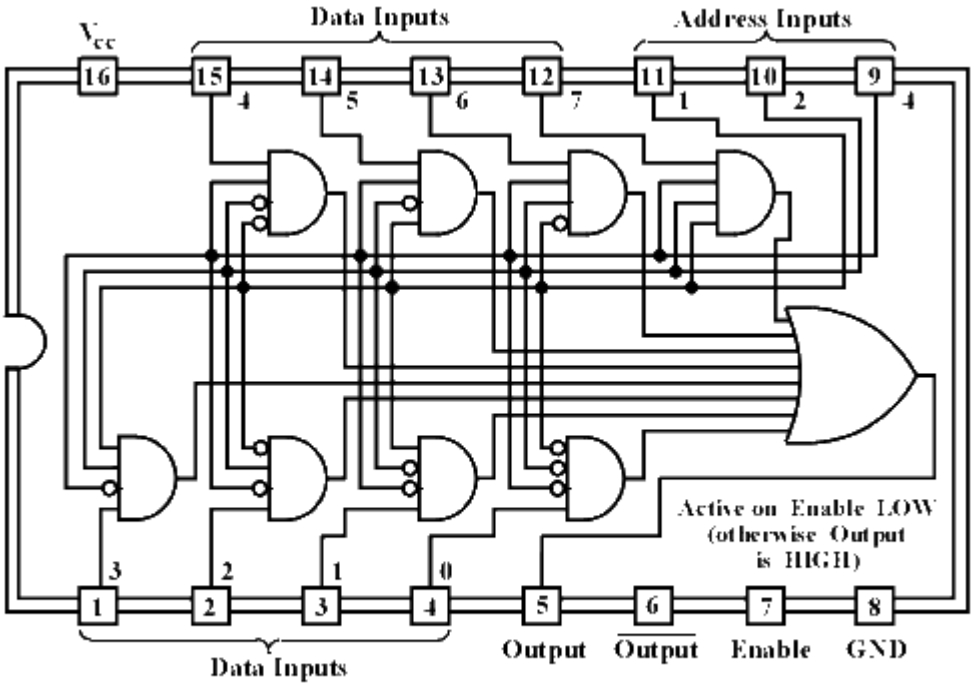
Thông số kỹ thuật chính của cảm biến nhiệt họLM35.

III. Chip 74LS151:

74LS151 là một chip có 8 đầu vào số đa chức năng tốc độ cao.Nó cung cấp trong một gói, khả năng chọn một bit từ 8 bit nguồn. 74LS151 có thể sử dụng như một chức năng của máy phát để phát những mức logic chức năng với 4 giá trị.



$S_0 \div S_2$: Chọn đầu vào
$I_0 \div I_7$: Ngõ vào đa chức năng
E : Chân vào tác động (tác động ở mức thấp)
Z : Ngõ ra đa chức năng
$\overline{Z}$: Ngõ ra bổ sung đa chức năng
V_{cc} : Chân nguồn
GND : Chân nối đất



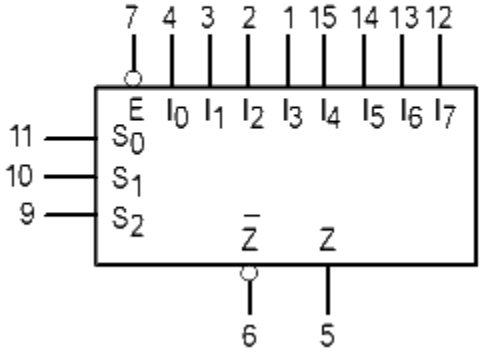
74151: 8-INPUT MULTIPLEXER

TRUTH TABLE

$\overline{E}$	S_2	S_1	S_0	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	$\overline{Z}$	Z
H	X	X	X	X	X	X	X	X	X	X	X	H	L
L	L	L	L	L	X	X	X	X	X	X	X	H	L
L	L	L	L	H	X	X	X	X	X	X	X	L	H
L	L	L	H	X	L	X	X	X	X	X	X	H	L
L	L	L	H	X	H	X	X	X	X	X	X	L	H
L	L	H	L	X	X	L	X	X	X	X	X	H	L
L	L	H	L	X	X	H	X	X	X	X	X	L	H
L	L	H	H	X	X	X	L	X	X	X	X	H	L
L	L	H	H	X	X	X	H	X	X	X	X	L	H
L	H	L	L	X	X	X	X	L	X	X	X	H	L
L	H	L	L	X	X	X	X	H	X	X	X	L	H
L	H	L	H	X	X	X	X	X	L	X	X	H	L
L	H	L	H	X	X	X	X	X	H	X	X	L	H
L	H	H	L	X	X	X	X	X	X	L	X	H	L
L	H	H	L	X	X	X	X	X	X	H	X	L	H
L	H	H	H	X	X	X	X	X	X	X	L	H	L
L	H	H	H	X	X	X	X	X	X	X	H	L	H

H = HIGH Voltage Level
L = LOW Voltage Level
X = Don't Care

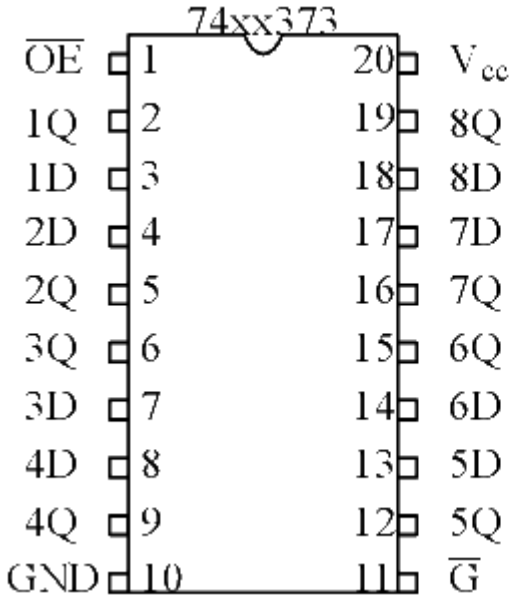
LOGIC SYMBOL



V_{CC} = PIN 16
GND = PIN 8

IV. Chip 74LS373:

Intel giới thiệu 74LS373. 74LS373 là một bộ cài đặt 8 mức flip-flops tác động nhanh. Nhiệm vụ của 74LS373 được hiển thị ở bản dưới. " Hi-Z" có nghĩa là ngõ ra ở trạng thái trở kháng cao. Ngõ ra của mạch được biết như ngõ ra 3 trạng thái. Ngõ ra 3 trạng thái được sử dụng để điều khiển các bus nơi mà các thiết bị khác nhau sẽ được tác động đủ để đóng con 3 trạng thái để điều khiển 1 bus.



$\overline{OE}$	$\overline{G}$	xD	xQ
H	X	X	Hi−Z
L	H	L	L
L	H	H	H
L	L	X	Q_o

Hi-Z : Ngõ ra trở kháng cao
Q_o : Mức ngõ ra trước khi cấu hình ngõ vào được thiết lập

$\overline{OE}$ (Output Enable): Mở cổng đệm

Chương III : Mạch thiết kế ghép nối

I. Sơ đồ nguyên lý của mạch thiết kế mạch:

II. Nguyên lý hoạt động của mạch:

Mạch đo ánh sáng và điều khiển đèn thông qua cổng nối tiếp RS-232. Mạch sử dụng quang trở để đo ánh sáng điều khiển đèn, với điện áp chuẩn 2.5V lấy từ Diod Zener REF25Z, sau đó qua mạch khuếch đại đưa LM358 đưa vào điện áp chuẩn 2.5V vào AD0804. AD0804 chuyển đổi tín hiệu tương tự từ quang trở thành tín hiệu số từ $D0 \rightarrow D7$.

Đầu vào quang trở dùng trở $330\,\Omega$ để giảm dòng vào, 2 diod 4148 tạo ra ngưỡng dẫn.

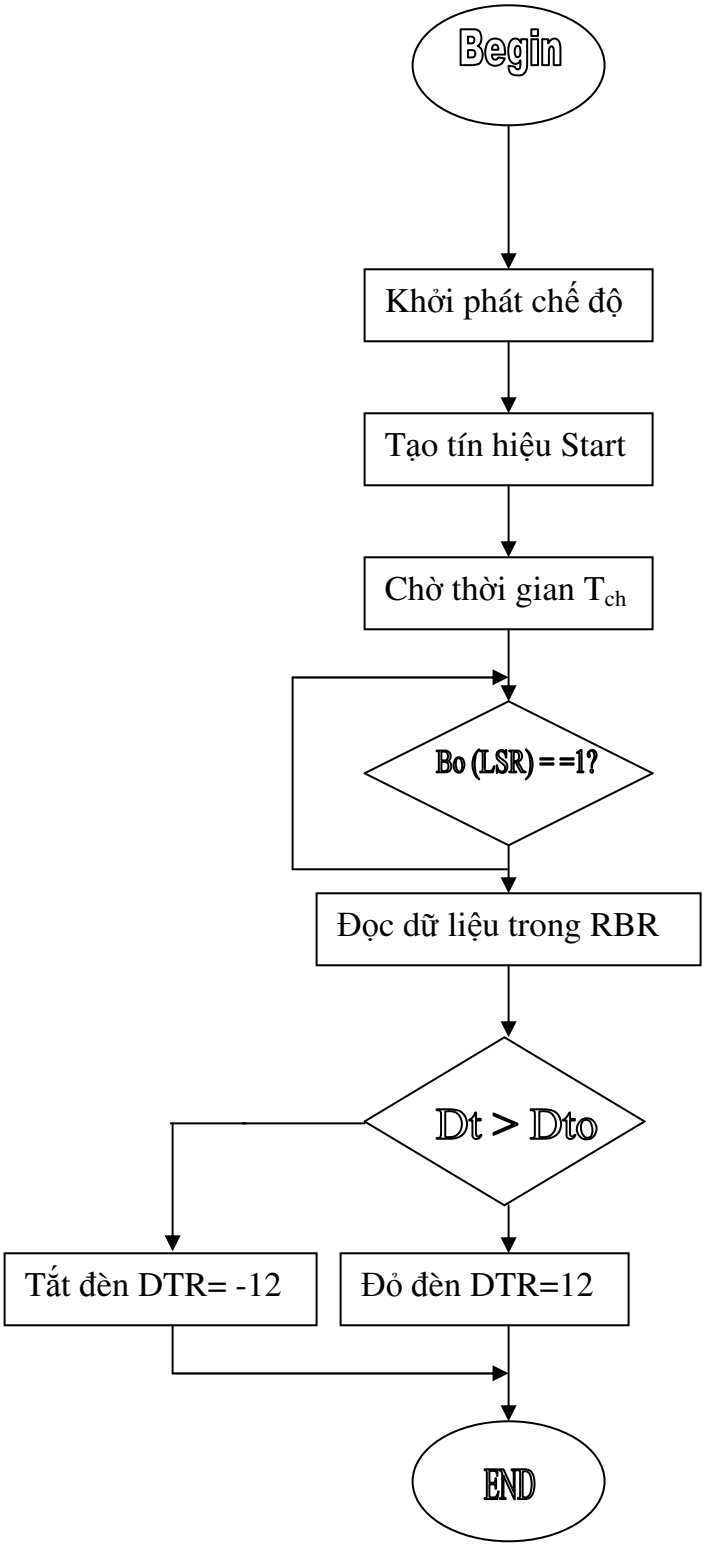
Khi chân $\overline{WR}$ được kích mức logic 0 từ chân RTS của cổng nối tiếp RS-232 thì AD0804 bắt đầu làm việc. Sau khi kết thúc 8 chu kỳ INTR báo kết thúc. Tín hiệu tương tự khi qua con AD0804 chuyển thành tín hiệu số, thông qua Đảo 7414 đưa bit 1 vào chân LE (chip đệm) và OE nối mass sẽ cho thông dữ liệu. Đầu ra kích con ba trạng thái thông dữ liệu từ 8 đầu vào $D0 \rightarrow D7$ khi đầu ra bằng đầu vào, mạch còn lại chuyển tín hiệu truyền từ song song qua nối tiếp.

Thông qua mạch dao động (7414, 10K, 10nF) sẽ tạo xung đưa vào IC74161 đếm từ $000 \rightarrow 111$, sau đó quay trở lại và cứ đếm mãi như vậy.

Q_D được kích mức logic 0 sẽ kích mở 74151(A), sau 8 lần đếm cứ lần lượt đưa dữ liệu ra đầu ra Y đến chân RxD của cổng RS-232 thông qua máy tính điều khiển, điện áp sẽ được đưa ra chân DTR +12V, dòng qua Diod 20mA làm Diod sáng lên kích mở Tranzitor, mở C828 và H106. Lúc này Role điện từ đóng nối nguồn AC đèn sáng lên.

$Q_AQ_BQ_C$ lần lượt tác động như bảng trạng thái sau:

Q_A	Q_B	Q_C	Y
0	0	0	D_0
1	0	0	D_1
0	1	0	D_2
0	0	1	D_3

Khi kết thúc 111 chuyển qua 011 D_7 tắt. Qua đầu đảo điều khiển tương tự 74151(B) có được bit dữ liệu $D_4 \rightarrow D_7$. Sau bit stop qua 3 bit Break. Vậy cuối cung thu được 8 bit data,0 parity, 1 stop, 9600 Baud, 6 bit Break.

III. Lưu đồ thuật toán: