

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
ĐẠI HỌC QUỐC GIA TP HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP HỒ CHÍ MINH  
KHOA ĐIỆN - ĐIỆN TỬ**

# **LUẬN VĂN TỐT NGHIỆP**

**ĐỀ TÀI:**

**MÁY TÍNH GIAO TIẾP TELETYPE**

SVTH : PHẠM HÙNG PHONG  
TRƯỜNG VIỆT NAM  
LỚP : 95 KĐĐ  
GVHD : QUÁCH THANH HẢI

THÀNH PHỐ HỒ CHÍ MINH 3 - 2000

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**      Cộng Hòa Xã Hội Chủ Nghĩa Việt  
ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH **Nam**  
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT      Độc Lập - Tự Do - Hạnh Phúc

**KHOA ĐIỆN**  
**BỘ MÔN ĐIỆN TỬ**

**NHIỆM VỤ LUẬN VĂN TỐT NGHIỆP**

Sinh viên thực hiện : PHẠM HÙNG PHONG -TRƯỜNG VIỆT NAM

Ngành : ĐIỆN - ĐIỆN TỬ

Lớp : 95 KĐĐ

TÊN ĐỀ TÀI : **MÁY TÍNH GIAO TIẾP TELETYPE**

1. CÁC SỐ LIỆU BAN ĐẦU: .....

2. NỘI DUNG THUYẾT MINH TÍNH TOÁN: .....

.....

3. CÁC BẢN VẼ : .....

4. GIÁO VIÊN HƯỚNG DẪN : QUÁCH THANH HẢI

5. NGÀY GIAO NHIỆM VỤ : 13 -12 - 1999

6. NGÀY HOÀN THÀNH NHIỆM VỤ : 28 - 2 -2000

Giáo viên hướng dẫn

Thông qua bộ môn  
Ngày.....tháng.....năm.....  
Chủ nhiệm bộ môn





## LỜI NÓI ĐẦU

Thông tin liên lạc là một vấn đề rất quan trọng trong giai đoạn hiện nay và cũng đang được phát triển một cách mạnh mẽ. Hiện nay có rất nhiều hình thức thông tin liên lạc như : vô tuyến, hữu tuyến ... Trong mỗi hình thức lại có nhiều dạng truyền như : truyền hình ảnh, truyền thoại, truyền mã số ...

Teletype là một trong những dạng truyền mã số được phát triển từ kiểu điện tín ngày xưa. Từ hai tín hiệu tích và te để hiểu được một từ hay một chữ có một mã riêng biệt nhóm các tín hiệu ấy lại với nhau.

Sau đó vì lượng thông tin ngày càng nhiều mà kiểu truyền tín hiệu có tốc độ quá chậm so với nhu cầu người ta mới nghĩ ra việc truyền những chỗi xung với hai mức: MARKING và SPACING trong một khung từ gọi là mã BAUDOT. Những xung này được truyền đi với tần số quy ước được gọi là tốc độ BAURATE được định nghĩa là số xung truyền đi trong một giây. Đây chính là phương pháp truyền của Teletype. Máy Teletype được cải tiến rất nhiều từ loại Teletype đầu tiên bằng cơ khí rất cồng kềnh, ồn ào và khó thao tác đến những máy Teletype gọn nhẹ, dễ thao tác.

Với sự bùng nổ hệ thống các máy vi tính như hiện nay, chúng em dùng máy tính để trao đổi dữ liệu như máy Teletype. Đây là đề tài để chúng em hoàn thành luận văn tốt nghiệp. Đề tài nêu lên được hình thức truyền số liệu giữa hai máy tính (giả máy Teletype), không nêu bật hết ưu điểm của máy tính vì truyền theo dạng Teletype chỉ truyền 5 bit ký tự (trong khi đó máy tính truyền được tối đa tới 7 bit ký tự).

## LỜI CẢM TẠ

Chúng em xin chân thành cảm ơn Ban Giám Hiệu và các thầy cô Trường Đại Học Sư Phạm Kỹ Thuật đã chỉ dẫn chúng em trong những tháng năm học tập tại trường.

Trong quá trình thực hiện tập luận văn tốt nghiệp chúng em xin chân thành cảm ơn thầy **Quách Thanh Hải**, giáo viên hướng dẫn, các thầy cô trong Khoa điện và các bạn trong và ngoài lớp đã động viên giúp đỡ chúng em hoàn thành luận văn tốt nghiệp.

Tuy nhiên, do khả năng còn hạn chế và thời gian có hạn, chắc chắn trong tập luận văn không tránh khỏi thiếu sót, mong được sự thông cảm và đóng góp ý kiến của quý thầy cô và các bạn để tập luận văn hoàn chỉnh hơn.

Chúng em xin chân thành cảm ơn.

TP Hồ Chí Minh - Tháng 2 năm 2000  
Nhóm sinh viên thực hiện  
Phạm Hùng Phong  
Trương Việt Nam

# MỤC LỤC

## \* PHẦN GIỚI THIỆU

- I. Tựa đề tài
- II. Nhiệm vụ luận văn tốt nghiệp
- III. Nhận xét giáo viên hướng dẫn
- IV. Nhận xét giáo viên phản biện
- V. Lời cảm tạ
- VI. Lời nói đầu
- VII. Mục lục

## \* PHẦN NỘI DUNG

### PHẦN A : CƠ SỞ LÝ THUYẾT

#### CHƯƠNG DẪN NHẬP

- I. Đặt vấn đề
- II. Mục đích nghiên cứu
- III. Cách thực hiện
- IV. Nhiệm vụ thực hiện

### CHƯƠNG II : CÁC KHÁI NIỆM LIÊN QUAN ĐẾN

#### VIỆC TRUYỀN THÔNG

- I. Truyền thông tuần tự
- II. Truyền thông đồng bộ bất đồng bộ
  - 1. Truyền thông đồng bộ
  - 2. Truyền thông bất đồng bộ
- III. Các khái niệm liên quan đến việc truyền thông
  - 1. Đầu cắm và ổ cắm
  - 2. Tín hiệu bắt tay
  - 3. DTE và DCE
  - 4. Các thông số của trao đổi tin nối tiếp
  - 5. Mạch trao đổi tin nối tiếp của PC
  - 6. Thủ tục trao đổi tin nối tiếp
- IV. Chuẩn giao tiếp RS\_232C
  - 1. Vài nét cơ bản về cổng nối tiếp
  - 2. Đặc điểm kỹ thuật về điện của RS\_232C
  - 3. Các IC kích phát và thu của RS\_232C

### CHƯƠNG III : NGUYÊN LÝ TRUYỀN DỮ LIỆU CỦA TELETYPE

- I. Giải đồ xung
- II. Khung ký tự Teletype
- III. Mã ký tự Teletype
- IV. Tiêu chuẩn giao tiếp máy Teletype
  - 1. Nguyên lý kết nối giữa 2 máy Teletype
  - 2. Giao tiếp dùng dòng điện vòng 20mA

### CHƯƠNG IV : KHÔI GHEP NỐI SONG SONG - NỐI TIẾP

#### VÀ NỐI TIẾP - SONG SONG

- I. Giới thiệu về việc truyền thông tin nối tiếp của PC
- II. Nhiệm vụ của khôi ghép nối song song - nối tiếp

và nối tiếp - song song	15
III. Sơ đồ khối của khối ghép nối	16
IV. Vi mạch trao đổi tin song song - nối tiếp KDB 8251A	17
<b>CHƯƠNG V : GIỚI THIỆU VỀ NGẮT CỦA PC</b>	28
I. Các loại ngắt của PC	28
1. Ngắt cứng	28
2. Ngắt mềm	28
II. Thủ tục xử lý ngắt chương trình	29
III. Giới thiệu về cách sử dụng IRQ4	30
IV. Sử dụng ngắt của ROM-BIOS	31
<b>PHẦN B : THIẾT KẾ PHẦN CỨNG</b>	37
I. Sơ đồ khối	37
II. Sơ đồ nguyên lý	38
III. Nguyên lý hoạt động của mạch	39
IV. Tính toán các linh kiện của mạch	39
<b>PHẦN C : XÂY DỰNG PHẦN MỀM</b>	41
I. Lưu đồ	41
II. Chương trình	
<b>PHẦN D : HƯỚNG THỰC CÔNG</b>	
I. Tổng quát	
II. Tổ chức mạch	
III. Các bước thi công	
IV. Báo cáo kết quả thi công	
* Kết luận	
* Phụ lục	
* Tài liệu tham khảo	



**PHẦN A**  
**CƠ SỞ LÝ THUYẾT**  
**CHƯƠNG DẪN NHẬP**

**I. Đặt vấn đề:**

Thông tin liên lạc luôn là vấn đề được quan tâm đến trong xã hội. Ngay từ thời xa xưa, con người cũng đã biết vận dụng những gì có sẵn như ngọn lửa, ám hiệu . . . để truyền tin.

Ngày nay, việc thu nhập thông tin đầy đủ và kịp thời là điều kiện tiên quyết cho sự thành bại của tất cả các quyết định trong mọi lĩnh vực. Thông tin phải được truyền nhanh chóng từ khắp mọi đơn vị thu thập thông tin về trung tâm, để từ đây xử lý và phản hồi các chỉ thị hoạt động cho các cơ sở.

Máy vi tính ngày càng giữ một vai trò quan trọng trong các lĩnh vực khoa học kỹ thuật và cuộc sống hàng ngày. Các ứng dụng của nó phục vụ cho con người ngày đa dạng. Việc dùng máy tính để truyền số liệu là hết sức thuận lợi, vì ngoài việc thiết kế phần cứng, ta có thể thay đổi phần mềm một cách dễ dàng và nhanh chóng.

**II. Mục đích nghiên cứu:**

Quá trình lựa chọn và thực hiện đề tài này là nhiệm vụ chúng em hoàn tất khóa học đại học.

Ngoài ra, việc nghiên cứu đề tài này để phát huy việc ứng dụng máy tính trong lĩnh vực truyền thông, tạo ra những sản phẩm, thiết bị có tính tiến bộ và hiệu quả giúp ích cho con người.

**III. Cách thức thực hiện:**

Đề tài này được thực hiện như sau:

Tìm hiểu nguyên lý hoạt động, cách thức truyền số liệu giữa hai máy Teletype.

Vận dụng kiến thức đã được học về cấu trúc máy tính để thiết kế mạch truyền số liệu giữa hai máy tính qua cổng COM1. Tìm hiểu các thanh ghi trong UART để viết phần mềm thực hiện việc truyền số liệu.

**IV. Nhiệm vụ thực hiện:**

Nhiệm vụ chính thực hiện là truyền số liệu theo phương thức truyền bất đồng bộ qua cổng COM1 của hai máy tính. Đồng thời, nâng điện áp của đường truyền lên  $\pm 60V$  để thực hiện việc truyền đi xa.

**Đề tài chia thành chia thành 4 phần chính:**

**Phần A** : Đề cập đến các cơ sở lý thuyết, các thuật ngữ được sử dụng trong quá trình truyền thông tuần tự.

**Phần B** : Thiết kế mạch phần cứng để cách ly giữa hai máy tính và nâng cao điện áp đường truyền.

**Phần C** : Viết lưu đồ điều khiển máy tính truyền số liệu qua cổng COM1.

**Phần D** : Hướng thi công.

## **CHƯƠNG II : CÁC KHÁI NIỆM LIÊN QUAN ĐẾN VIỆC TRUYỀN THÔNG**

### **I. Truyền thông tuần tự:**

Hầu hết các máy vi tính lưu trữ và thao tác dữ liệu của chúng theo cách song song. Nghĩa là khi truyền 1 Byte thì các Bit đi cùng một lúc trên các mạch dây song song. Số các Bit gửi đi cùng một lúc thay đổi tùy thuộc vào mỗi loại máy tính khác nhau nhưng thường là 8 hoặc bội số của 8.

Tuy nhiên ngoài việc trao đổi tin song song với một máy tính khác (hoặc một thiết bị ngoài) có dạng tin vào - ra song song, máy tính còn trao đổi tin nối tiếp với máy tính khác hoặc thiết bị ngoài) có dạng tin vào ra từng bit một. Bộ giao tiếp từng tự phải nhận những Byte ở dạng song và gửi đi các bit một cách riêng biệt.

Dữ liệu trên đường truyền trong truyền thông từng tự chỉ ở hai trạng thái là Mark và Space tương ứng với trạng thái điện thế âm và điện thế dương. Bất kỳ dữ liệu truyền nào, trước tiên đều phải chuyển thành một dãy thứ tự các Mark và Space (Mark tương ứng với số 1, Space tương ứng với số 0).

### **II. Truyền thông đồng bộ - bất đồng bộ:**

#### **1. Truyền thông đồng bộ: (Synchronous Communication)**

Quá trình truyền và nhận xảy ra gần như đồng thời (có sự trễ do vận tốc truyền trên đường dây) theo từng bit hay nhóm bit do một máy phát xung nhịp tạo ra. Khi những ký tự được gửi theo một khối ở tốc độ của máy, chúng được đưa ra ngoài một cách đều đặn. Như vậy sẽ không cần thiết thêm vào cho mỗi ký tự truyền những Start bit và Stop bit. Bởi vì một khi ký tự đầu tiên được nhận thì thiết bị nhận có thể tiên đoán một cách chính xác khi nào thì những ký tự tiếp theo sẽ đến. Nói cách khác, thiết bị nhận có thể tự đồng bộ hóa với máy truyền. Phương thức truyền như trên gọi là truyền thông đồng bộ.

#### **Phương pháp này có đặc điểm sau:**

- Nhanh : vì phát và nhận hầu như tức thời.
- Không tin cậy : dễ mất tin.
- Luôn đòi hỏi nguồn phát và nguồn nhận phải sẵn sàng trao đổi tin .

#### **2. Truyền thông bất đồng bộ : (Asynchronous Communication)**

Việc phát và nhận xảy ra không đồng thời, không cùng một nhịp do hai máy phát nhịp thời gian khác nhau điều khiển, dạng tin phát và tin thu không giống nhau. Khi dữ liệu được truyền bởi người sử dụng nhập từ bàn phím, các ký tự nhập luôn luôn được gửi đi và nhận vào một cách bất đồng bộ, bởi vì người sử dụng không thể nhấn phím một cách liên tục và đều đặn. Do đó, khi một máy tính nhận những ký tự, thì giữa những ký tự nhận đó sẽ có những thời gian ngưng khác nhau. Điều này sẽ gây cho máy tính việc không thể biết chính xác được khi nào thì một ký tự kế tiếp sẽ được gửi đến.

Vì thiếu tính liên tục như vậy, cho nên cần phải thêm vào những bit phụ trước và sau ký tự được truyền. Những bit thêm vào này gọi là Start bit, Stop bit.

Phương thức truyền như trên gọi là truyền thông bất đồng bộ. Quá trình phát và nhận được diễn ra như sau:

- Nguồn phát và nguồn nhận đưa tín hiệu yêu cầu trao đổi tin (hay sẵn sàng trao đổi tin).
- Nguồn nhận hoặc nguồn phát đưa tín hiệu xác nhận (chấp nhận yêu cầu).
- Nguồn phát đưa tin vào đường dây số liệu để ghi vào thanh ghi số liệu đệm của khối ghép nối.
- Nguồn nhận nhận số liệu từ khối ghép nối.

**Đặc điểm của phép truyền này là:**

- tin cậy (theo phương thức hỏi đáp hay bắt tay hoặc hội thoại).
- chậm, tốn thiết bị vì có cơ chế hỏi đáp và bộ đệm số liệu.

**III. Các khái niệm liên quan đến việc truyền thông:**

**1. Đầu cắm và ổ cắm: (Plug And Socket)**

Có một vài kiểu khác nhau về đầu cắm và ổ cắm cho những cáp kết nối với thiết bị tuần tự. Bộ kết nối D\_type 25 chân và 9 chân được sử dụng rộng rãi nhất, đôi khi người ta còn gọi là DB\_25 và DB\_9.

Những bộ kết nối gồm có những chân (Pins) hoặc những lỗ cắm (Sockets). Bộ kết nối với những chân cắm (pins) là những bộ kết nối "đực" (male). Bộ kết nối với những lỗ cắm (Sockets) là những bộ kết nối "cái" (Female). Trên mỗi chân cắm hoặc lỗ cắm của bộ kết nối (Connector) đều được đánh số.

**2. Tín hiệu bắt tay: (Handshaking)**

Trong nhiều trường hợp, thiết bị truyền cần biết rằng thiết bị nhận có sẵn sàng nhận tin hay không. Thí dụ ta có thể gửi dữ liệu từ máy này sang máy khác và máy thứ hai không thể xử lý dữ liệu nhanh bằng với tốc độ nhận dữ liệu. Trong trường hợp này, thông tin phải được gửi ngược từ thiết bị nhận tới thiết bị truyền để chỉ ra rằng nó sẵn sàng hoặc không sẵn sàng nhận. Thông tin này gọi là dòng kiểm tra (Flow Control) hoặc tín hiệu bắt tay (Handshaking).

Có hai loại Handshaking là Handshaking phần cứng và handshaking phần mềm. Cả hai loại này đều bao gồm những tín hiệu gửi ngược từ thiết bị nhận đến thiết bị truyền.

Với Handshaking phần cứng: thiết bị nhận gửi một điện thế dương trên đường dây bắt tay khi nó sẵn sàng nhận dữ liệu. Khi máy truyền nhận một điện thế âm, nó biết rằng phải ngừng việc gửi dữ liệu.

Với handshaking phần mềm, tín hiệu bắt tay chứa đựng những ký tự đặc biệt được truyền theo đường dây dữ liệu thay vì trên đường dây bắt tay.

**3. DTE (Data Terminal Equipment) và DCE (Data Communication Equipment)**

DTE : là thiết bị đầu cuối được hiểu tương tự như máy tính.

DCE : được hiểu tương tự như Modem.

**Các chuẩn để phân biệt DTE và DCE:**

- Thiết bị nào sử dụng chân số 2 để xuất dữ liệu thì được hiểu như thiết bị DTE.

- Thiết bị nào sử dụng chân số 2 để nhận dữ liệu thì được hiểu như thiết bị DCE.

Tuy nhiên hai cách phân biệt trên chỉ là tương đối.

#### **4. Các thông số của trao đổi tin nối tiếp:**

- Khoảng cách trao đổi tin: khoảng cách giữa nguồn phát và nguồn thu tin.

Nếu ở khoảng cách gần (dưới 300m) sự thu và phát không cần modem

Nếu ở khoảng cách xa (lớn hơn 300m) cần Modem cho tin cậy.

- Tốc độ trao đổi thông tin: đơn vị được tính là bit trong một giây (bit per second, bps) còn gọi là Baud. Thường có tốc độ 600, 1200, 2400, 4800, 9600 baud (hay bps).

Trao đổi tin không đồng bộ thường có tốc độ chậm (dưới 4800 bps) còn trao đổi tin đồng bộ và lai có thể đạt tới trên 9600 bps. Hiện nay tốc độ trao đổi tin số đã đạt tới cỡ Mbps ( $10^6$  bps).

- Chiều trao đổi tin : trao đổi tin có thể

+ Trên một đường dây duy nhất, có thể có hai chiều đi và về giữa hai nguồn phát và thu tin. Ở một thời điểm chỉ truyền theo một chiều (bán song công)

+ Trên hai đường dây riêng rẽ TxD (phát hay truyền) và RxD (nhận hay thu) với các chiều xác định (đơn công) và tại một thời điểm có thể truyền đồng thời theo cả hai chiều (song công). Tùy mạch khuếch đại đường dây và số đường dây nối (một hoặc hai đường) ta có chiều trao đổi tin khác nhau (đơn công, bán song công hay song công).

#### **5. Mạch trao đổi tin nối tiếp của máy vi tính:**

Tùy lối ra, cách nối mạch và thiết bị ngoài ta có các loại mạch trao đổi tin nối tiếp giữa máy vi tính và thiết bị ngoài khác nhau.

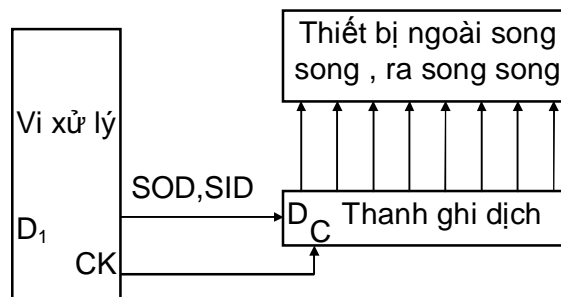
##### **a. Mạch không cần khối ghép nối:**

Đó là sự trao đổi tin với thiết bị ngoài nối tiếp và lối ra hay vào của vi xử lý cũng là nối tiếp (hình 1). Có hai loại lối vào ra của vi xử lý là:

- Lối vào ra nối tiếp riêng biệt (SID, SOD) như của vi xử lý 8085.

- Một chân lối vào ra song song của vi xử lý được dùng cho lối vào ra nối tiếp.

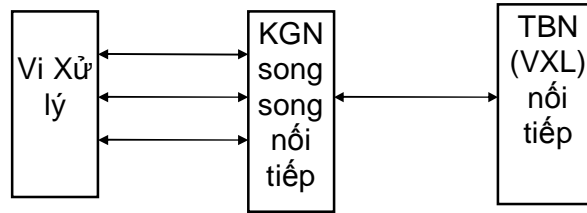
Cả hai trường hợp trên đều đòi hỏi nhiều thời gian trao đổi tin của vi xử lý.



**Hình 1:**

**b. Mạch cần khối ghép nối song song nối tiếp (Hình 2):**

Người ta dùng khối ghép nối song song nối tiếp để biến đổi tín hiệu của vi xử lý (đưa ra một lần) thành tín hiệu nối tiếp truyền cho thiết bị ngoài nối tiếp.



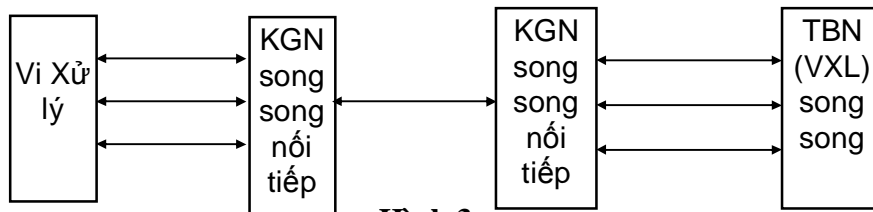
**Hình 2**

**c. Mạch cần khối ghép nối song song nối tiếp và nối tiếp song song (Hình 3):**

Đây là trường hợp trao đổi tin giữa vi xử lý với thiết bị ngoài trao đổi tin song song. Trường hợp này xảy ra khi máy vi tính đặt cách xa thiết bị ngoài và không thể thực hiện trao đổi tin song song được vì tốn nhiều đường dây. Có hai trường hợp:

- Nếu khoảng cách giữa máy vi tính và thiết bị ngoài gần (dưới 300m) không cần Modem.

- Nếu khoảng cách giữa máy vi tính và thiết bị ngoài xa (trên 300m) cần có Modem để điều chế tín hiệu số thành âm tần (tránh nhiễu) và tín hiệu âm tần điều chế thành tín hiệu số.

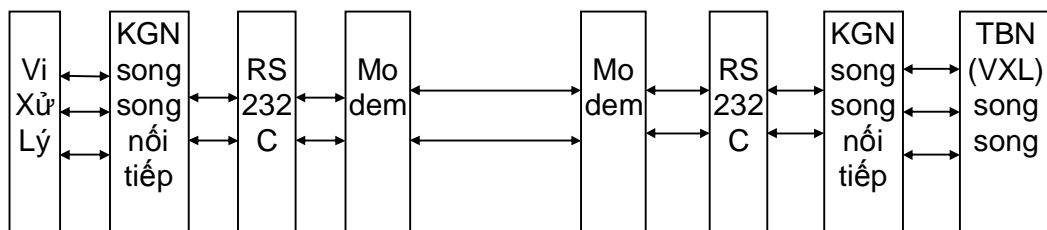


**Hình 3**

**d. Mạch cần khối ghép nối, Modem và khối ghép nối RS 232C**

**(Hình 4):**

Đây là trường hợp tổng quát và thông dụng của trao đổi tin giữa máy tính (song song) với thiết bị ngoài (song song) đặt ở khoảng cách xa (cần modem) và sử dụng chính đường dây điện thoại để trao đổi tin (KGN RS\_232C) biến đổi mức tín hiệu TTL thành mức tín hiệu trên đường dây điện thoại ( $\pm 12 V$ ).



**Hình 4**

## **6. Thủ tục trao đổi tin nối tiếp:**

Thủ tục trao đổi tin giữa một máy vi tính và một thiết bị nhận tin song song (thiết bị đầu cuối, máy in song song, đục băng, ...) thông qua các khối ghép nối song song - nối tiếp và nối tiếp - song song và Modem (Hình 4) theo trình tự sau:

### **a. Thủ tục phát tin TxD:**

- Thiết bị đầu cuối (hay máy vi tính) gửi tín hiệu DTR (Data Terminal Ready - Sự sẵn sàng của thiết bị đầu cuối có số liệu) mức thấp cho Modem báo nó sẵn sàng.

- Modem gửi trả lời thiết bị đầu cuối (TBĐC) bằng tín hiệu DSR (Data Set Ready) mức thấp. Thông thường, modem được đóng mạch nguồn nuôi bởi DTR và báo hiệu đã đóng mạch bởi DSR.

- Nếu thiết bị đầu cuối có một ký tự (Character) sẵn sàng gửi đi, nó gửi RTS (Request To Send - yêu cầu gửi) mức thấp cho Modem.

- Modem gửi tín hiệu CD (Carrier Detect - phát hiện sóng mang) cho TBĐC để báo rằng nó đã liên lạc được với máy vi tính.

- Khi Modem đã hoàn toàn sẵn sàng phát số liệu lên đường dây, nó phát xung nhịp (Modem Clock) và tín hiệu CTR (Clear To Send) tới thiết bị đầu cuối.

- TBĐC gửi các ký tự số liệu (SUD) TxD cho Modem.

- Khi thiết bị đầu cuối gửi xong số liệu, nó nâng mức RTS lên cao báo cho Modem là đã phát xong .

- Modem trả lời thiết bị đầu cuối bằng cách kết thúc tín hiệu CTS về mức cao, báo đã hoàn thành việc truyền tin TxD.

### **b. Thủ tục nhận tin RxD:**

Khi một thiết bị đầu cuối nhận tin nối tiếp từ đường dây, trình tự diễn ra như sau:

- TBĐC thu gửi DTR mức thấp cho modem báo sẵn sàng.

- Modem thu giữ trả lời bởi DSR.

- Modem thu nhận tín hiệu CD từ đường dây và kích thích phát tín hiệu nhịp modem (Modem Clock) cho tín hiệu thu.

- TBĐC phát tín hiệu RTS mức thấp cho modem biết là sẵn sàng thu.

- Modem nhận tín hiệu RTS và phát CTS mức thấp cho thiết bị đầu cuối thu biết modem sẵn sàng nhận tin.

- Modem nhận tín hiệu TxD đã điều chế ở trên đường dây đưa vào bộ giải điều chế và truyền chuỗi tín hiệu RxD cho thiết bị đầu cuối thu.

- Khi thu xong, TBĐC thu nâng RTS lên cao báo cho modem biết việc thu một lời tin đã xong.

- Modem thu nâng mức CTS lên cao để báo đã kết thúc việc thu các tín hiệu RxD.

## **IV. Chuẩn giao tiếp RS 232C:**

### **1. Vài nét cơ bản về cổng nối tiếp:**

Cổng nối tiếp RS\_232C là giao diện phổ biến rộng rãi nhất. Người dùng máy tính PC còn gọi các cổng này là COM1, COM2. Giống như cổng máy in,

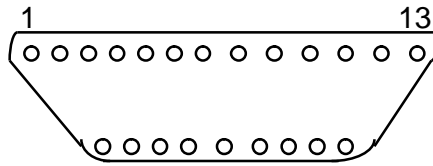
cổng nối tiếp RS\_232C cũng được sử dụng một cách rất thuận tiện cho mục đích đo lường và điều khiển .

Chuẩn RS\_232C dùng với tốc độ truyền dữ liệu là 20 Kbps với khoảng cách truyền lớn nhất gần 15 m. Đây là một dạng giao tiếp dạng TTL và bộ kích đường dây không cân bằng.

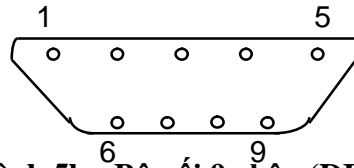
Việc truyền dữ liệu qua cổng RS\_232C được tiến hành theo cách nối tiếp, nghĩa là các bit dữ liệu được gửi đi nối tiếp nhau trên một đường dẫn.

Trên hình 5 là sự bố trí chân của phích cắm RS\_232C ở máy tính PC.





**Hình 5a**: Bộ nối 25 chân (DB\_25)



**Hình 5b**: Bộ nối 9 chân (DB\_9)

**Trong đó :**

- AA : Protective Ground (nối đất bảo vệ)
- TxD : Transmitter Data (truyền dữ liệu)
- RxD : Received Data (nhận dữ liệu)
- RTS :Request To Send (yêu cầu gọi)
- CTS : Clear To Send (xóa việc gọi)
- DSR : Data Set Ready (dữ liệu sẵn sàng)
- SG : Signal Ground (nối đất)
- CD : Carrier Detect (dò sóng mang)
- ST : Select Stanty
- SCF : Secondary Recived Line Signal Det
- SCB : Secondary Clear To Send
- SBA : Secondary Transmitter Data
- DB : Transmitter Signal Element Timing
- SBB :Secondary Received Data
- DD : Received Signal Element Timing
- SCA : Secondary Request To Send
- DTR : Data Terminal Ready ( trạm đầu cuối thiết bị sẵn sàng)
- CG : Signal Quality Detector
- RI : Ring Indicator ( bộ chỉ thị vòng)
- CH : Data Signal Rate Selector
- CI : Data Signal Rate Selector
- DA : Transmitter Signal Element Timing

**2. Đặc điểm kỹ thuật về điện của RS 232C:**

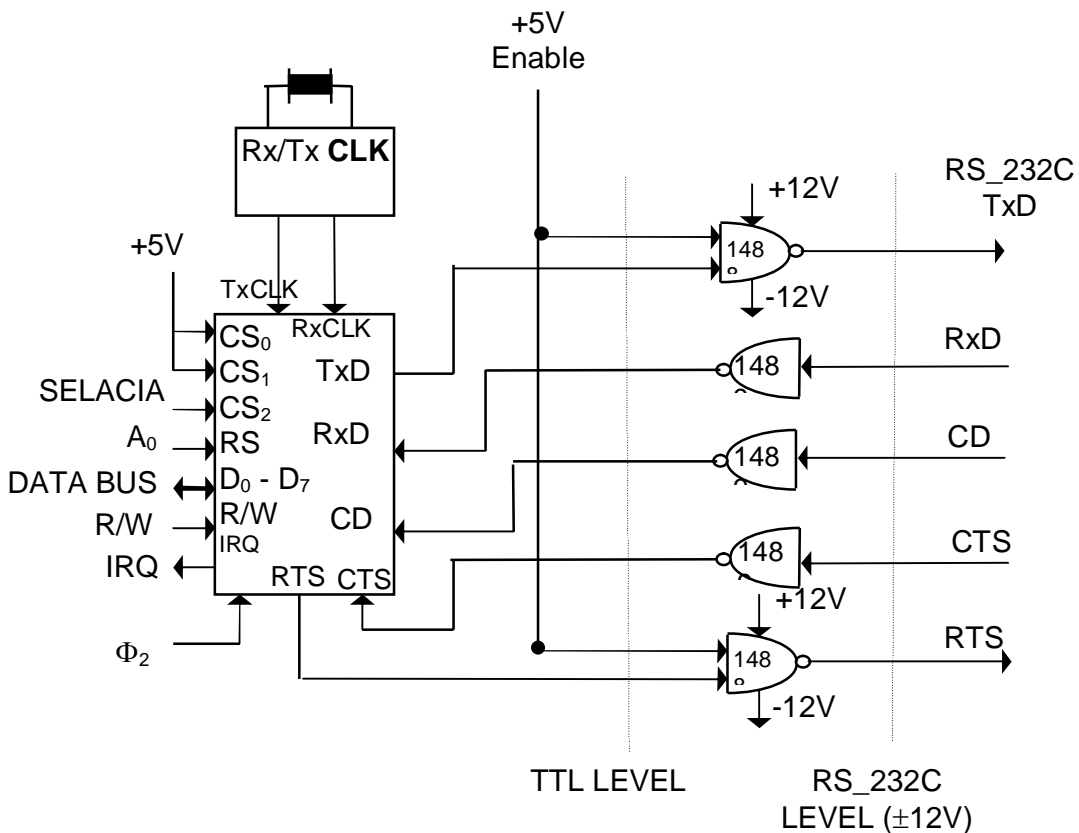
Mức điện áp logic của RS\_232C là hai khoảng điện áp giữa +15 V và -15 V. Các đường dữ liệu sử dụng mức logic âm, logic 1 có điện thế giữa -5V và -15V, logic 0 có điện thế giữa +5V và +15V. Tuy nhiên các đường điều khiển sử dụng mức logic dương, giá trị TRUE từ +5V đến +15V, giá trị FALSE từ -5V đến -15V. Ở chuẩn giao tiếp này, giữa ngõ ra bộ kích phát và ngõ vào bộ thu có mức nhiễu được giới hạn là 2V. Do vậy ngưỡng lớn nhất của ngõ vào là ±3V, trái lại mức ±5V là ngưỡng nhỏ nhất đối với ngõ ra. Ngõ ra bộ kích phát khi không tải có điện áp là ±25V.

**Các đặc điểm về điện khác bao gồm:**

- $R_L$  (điện trở tải) được nhìn từ bộ kích phát phải có giá trị giữa  $7K\Omega$  và  $3K\Omega$ .
- $C_L$  (điện dung tải) được nhìn từ bộ kích phát không được vượt quá  $2500\text{ pF}$ .
- Để ngăn cản sự dao động quá mức, tốc độ thay đổi của điện áp không được vượt quá  $30\text{ V/us}$ .
- Đối với các đường điều khiển, thời gian chuyển của tín hiệu không được vượt quá  $1\text{ ms}$ . Đối với các đường dữ liệu, thời gian chuyển phải không vượt quá  $4\%$  thời gian của 1 bit hoặc  $1\text{ ms}$ .

### 3. Các IC kích phát và thu của RS 232C:

Nhờ tính phổ biến của giao tiếp RS\_232C, người ta đã chế tạo các IC kích phát và thu. Hai vi mạch như vậy được Motorola sản xuất là IC kích phát MC1488 và IC thu 1489 có dạng vỏ vuông. Hình 6 cho thấy 1port RS\_232C được kết nối với ACIA 6850 sử dụng MC1488 và MC1489. Mỗi IC kích phát 1488 nhận một tín hiệu mức TTL và chuyển thành tín hiệu ở ngõ ra tương thích với mức điện áp của RS\_232C, IC thu 1489 phát hiện các mức vào của RS\_232C và chuyển thành các ngõ ra có mức TTL.

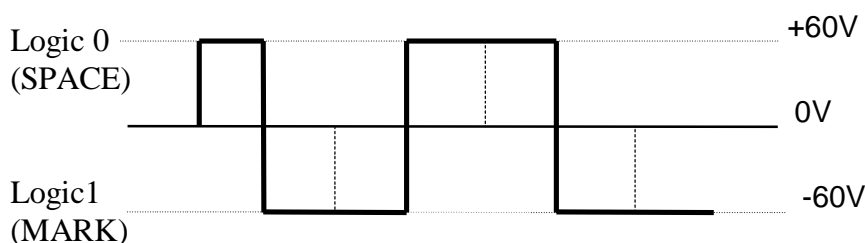


## CHƯƠNG III : NGUYÊN LÝ TRUYỀN DỮ LIỆU CỦA TELETYPE

### I. Giải đồ xung của Teletype:

Teletype truyền dữ liệu kiểu xung. Trước đây người ta truyền với dạng xung MARK ứng với điện áp dương và xung SPACE ứng với điện áp âm hoặc 0V (với các đời máy Teletype bằng cơ khí). Tuy nhiên, sau này khi máy Teletype điện tử ra đời người ta chuyển sang truyền theo mức logic, ứng với mức logic 1 (điện áp âm) là xung MARKING, và mức 0 (điện áp dương) là xung SPACING.

**Mô tả như sau:**

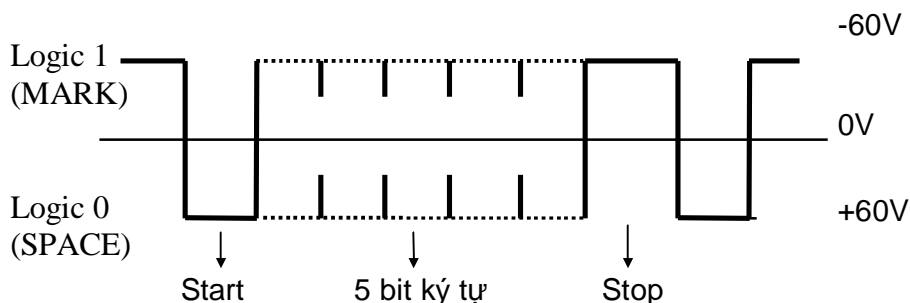


### II. Khung ký tự của Teletype:

Một Frame của Teletype được quy ước như sau:

- 1 bit Start (SPACE)
- 5 bit ký tự (SPACE hoặc MARK)
- 1,5 bit Stop (MARK)

**Mô tả như sau:**



### III. Mã ký tự Teletype:

Các bit ký tự					LETTER	FIGURE
4	3	2	1	0		
0	0	0	0	0	BLANK	BLANK (SP)

0	0	0	0	1	E	3
0	0	0	1	0	LF	LF
0	0	0	1	1	A	-
0	0	1	0	0	SPACE	SPACE
0	0	1	0	1	S	BELL
0	0	1	1	0	I	8
0	0	1	1	1	U	7
0	1	0	0	0	CR	CR
0	1	0	0	1	D	\$
0	1	0	1	0	R	4
0	1	0	1	1	J	'
0	1	1	0	0	N	'
0	1	1	0	1	F	!
0	1	1	1	0	C	:
0	1	1	1	1	K	(
1	0	0	0	0	T	5
1	0	0	0	1	Z	"
1	0	0	1	0	L	)
1	0	0	1	1	W	2
1	0	1	0	0	H	#
1	0	1	0	1	Y	6
1	0	1	1	0	P	0
1	0	1	1	1	Q	1
1	1	0	0	0	O	9
1	1	0	0	1	B	?
1	1	0	1	0	G	&
1	1	0	1	1	FIGURE	FIGURE
1	1	1	0	0	M	.
1	1	1	0	1	X	/
1	1	1	1	0	V	;
1	1	1	1	1	LETTER	LETTER

Hai ký tự LETTER và FIGURE dùng để chuyển đổi bộ chữ của Teletype từ mặt chữ sang mặt số (dấu).

#### **IV. Tiêu chuẩn giao tiếp máy Teletype:**

##### **1. Nguyên lý kết nối giữa hai máy Teletype:**

Các đường tín hiệu của máy Teletype gồm có như sau:

- Một đường vào là đường thu Rx.
- Một đường ra là đường phát Tx.
- Một đường mass.

Khi mở máy, đường phát sẽ từ mức logic 0 (+60V) sẽ được nâng lên mức logic 1 (-60V). Tổng đài viba hoặc máy Teletype đối phương khi nhận được tín hiệu này (đường thu của máy đối phương được Set lên mức logic 1) thì sẽ gửi trả tín hiệu trả lời cũng bằng cách Set đường phát của máy họ lên mức logic 1. Đường thu của máy ta sẽ nhận được mức logic 1. Lúc này coi như hai máy đã được bắt tay và bắt đầu truyền số liệu cho nhau.

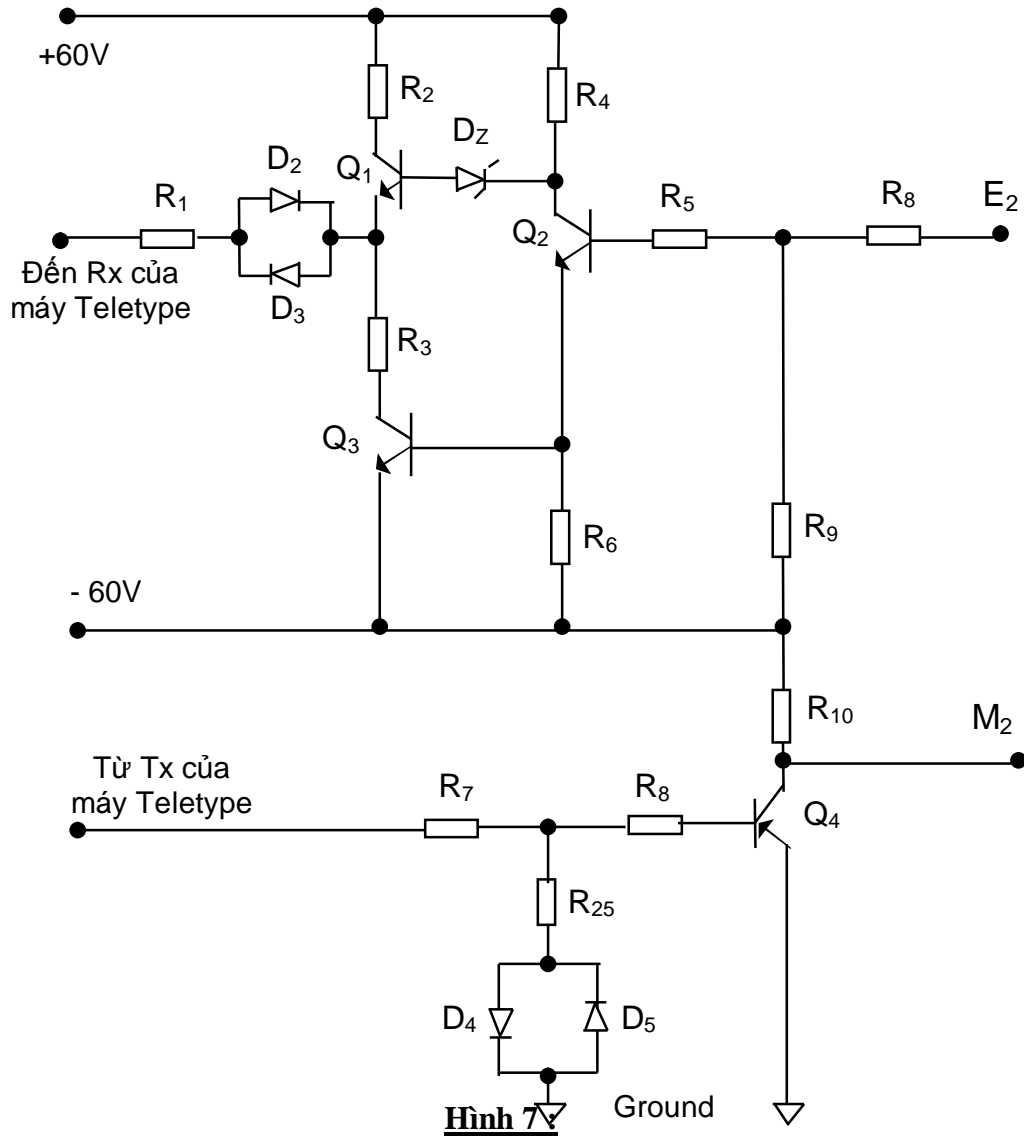
Trường hợp máy ta không nhận được tín hiệu trả lời tức là có sự cố đường dây hoặc vì máy đối phương không mở máy. Đường phát của máy ta sẽ tự động trả về mức logic 0 (+60V) và thông báo hở mạch.

Như vậy ta có thể phân tích mạch kết nối với máy Teletype như hình 7 :

- Khi không bắt tay:  $E_2$  được nối với nguồn -60V  $\rightarrow Q_2$  không dẫn  $\rightarrow Q_3$  cũng không dẫn. còn  $Q_1$  được dẫn báo hòa. do đó, nguồn +60 V qua  $R_2 \rightarrow D_3$  (Led  $D_3$  sáng báo hở mạch)  $\rightarrow R_1$  đến đường dây thu của máy đối phương ( Như vậy đường dây thu của máy đối phương đang ở mức logic 0).

- Tương tự đường dây phát của máy đối phương cũng ở mức logic 0, nên đường dây thu của máy ta cũng ở mức logic 0, cực B của  $Q_4$  được đặt vào một điện áp dương nên  $Q_4$  tắt  $\rightarrow$  điện áp ở  $M_2$  là -60V ( $D_4$  sáng).

- Khi cần gọi, ta cho  $E_2$  nối đất  $Q_2$  dẫn  $\rightarrow Q_3$  dẫn  $\rightarrow Q_1$  tắt. Do đó từ Relay của máy đối phương một dòng điện từ mass qua  $D_2 \rightarrow Q_3 \rightarrow$  về nguồn -60V. Lúc này, đối phương sẽ trả lời bằng cách gọi đến đường dây thu của ta mức logic 1 (dòng-20mA). Do đó,  $Q_4$  dẫn và  $M_2$  coi như nối đất ( $D_2$  sáng). Nhận được tín hiệu này tức là đường dây đã được kết nối và sẵn sàng truyền số liệu.



**2. Giao tiếp dùng dòng điện vòng 20 mA:**

Giao tiếp dùng vòng điện vòng 20mA sử dụng một dòng điện vòng 2 dây đơn giản để truyền dữ liệu nối tiếp. Logic 1 biểu thị bằng dòng điện vòng  $I = 20\text{mA}$  và logic 0 biểu thị bằng dòng  $I = 4 \text{ mA}$ .

## **CHƯƠNG IV:    KHỐI GHÉP NỐI SONG SONG - NỐI TIẾP VÀ NỐI TIẾP - SONG SONG**

### **I. Giới thiệu về việc truyền thông tin nối tiếp của máy vi tính:**

Như đã giới thiệu ở phần trước, tín hiệu Teletype là tín hiệu nối tiếp. Do đó, để giao tiếp với máy vi tính thì phải qua hệ thống phối hợp nối tiếp của máy vi tính. Bởi vì trong máy tính chỉ sử dụng mã ASCII để nhận biết các ký tự và CPU làm việc trên các thanh ghi dùng 8 hoặc 16 bit dữ liệu → CPU làm việc với các Chíp là song song.

Để thu phát nối tiếp từ các thiết bị song song, chúng ta phải biến đổi tin song song thành nối tiếp và nối tiếp thành song song và chế tạo một khối ghép nối có đồng thời 2 chức năng trên để trao đổi tin thu, phát giữa một thiết bị song song với đường dây nối tiếp.

Bộ phối ghép nối tiếp trong máy vi tính cho phép nhận một tín hiệu từ bên ngoài vào và biến đổi thành song song để đưa vào CPU hoặc ngược lại nhận dữ liệu song song từ CPU biến đổi thành tín hiệu nối tiếp gửi ra ngoài.

Sau đây, chúng em xin giới thiệu sơ lược về bộ phối ghép nối tiếp - song song trong máy vi tính IBM PC\_AT (vì chỉ cần phần nối tiếp nên không giới thiệu phần song song của nó).

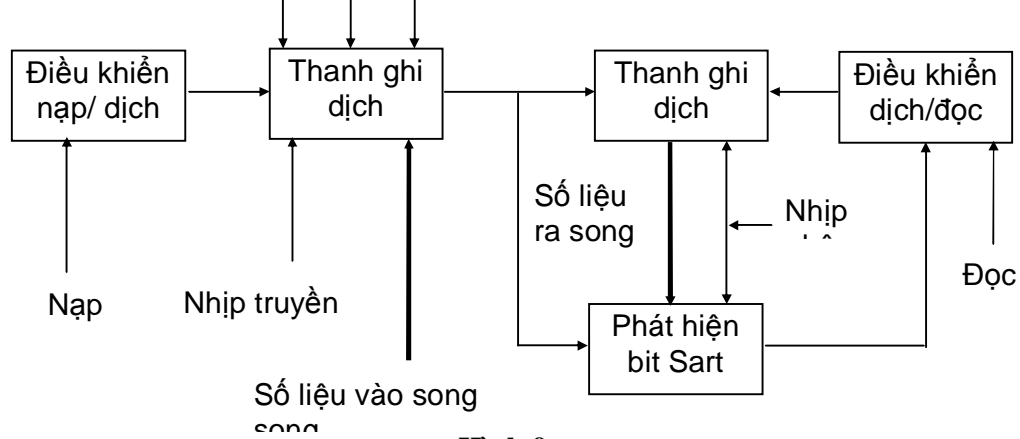
Phần nối tiếp của bộ phối ghép nối tiếp trong máy vi tính AT được chương trình hóa một cách đầy đủ để trợ giúp cho việc liên lạc thông tin dị bộ.

Bộ phối ghép này sẽ tự động thêm hoặc lấy ra những bit Start, Stop và các bit chặn lẻ. Nó có một chương trình đặc biệt tạo ra mã BAUD\_RATE cho phép vận hành từ 50 → 9600 bps. Nó cho phép thiết lập 1 Frame với 5, 6, 7 hoặc 8 bit ký tự với 1; 1,5 hoặc 2 bit Stop. Nó có một hệ thống ngắt ưu tiên dùng để điều khiển phát, thu, báo lỗi, trạng thái đường dây. . .

### **II. Nhiệm vụ của khối ghép nối song song - nối tiếp và nối tiếp song song không đồng bộ:**

Khối ghép nối có nhiệm vụ sau: (Hình 9)

- Thu tin song song từ máy vi tính, thiết bị đầu cuối để biến thành tin nối tiếp để truyền trên đường dây TxD.
- Thu tin nối tiếp từ đường dây RxD để biến thành tin song song cho máy vi tính, thiết bị đầu cuối.
- Chèn và loại trừ các bit khung Start, Stop của khung tin.
- Điều khiển Modem với các tín hiệu hội thoại (phát cho modem DTR, RTS và nhận DSR, CTS từ modem)
- Điều khiển các chế độ đồng bộ, không đồng bộ, số bit tin, số bit Stop.

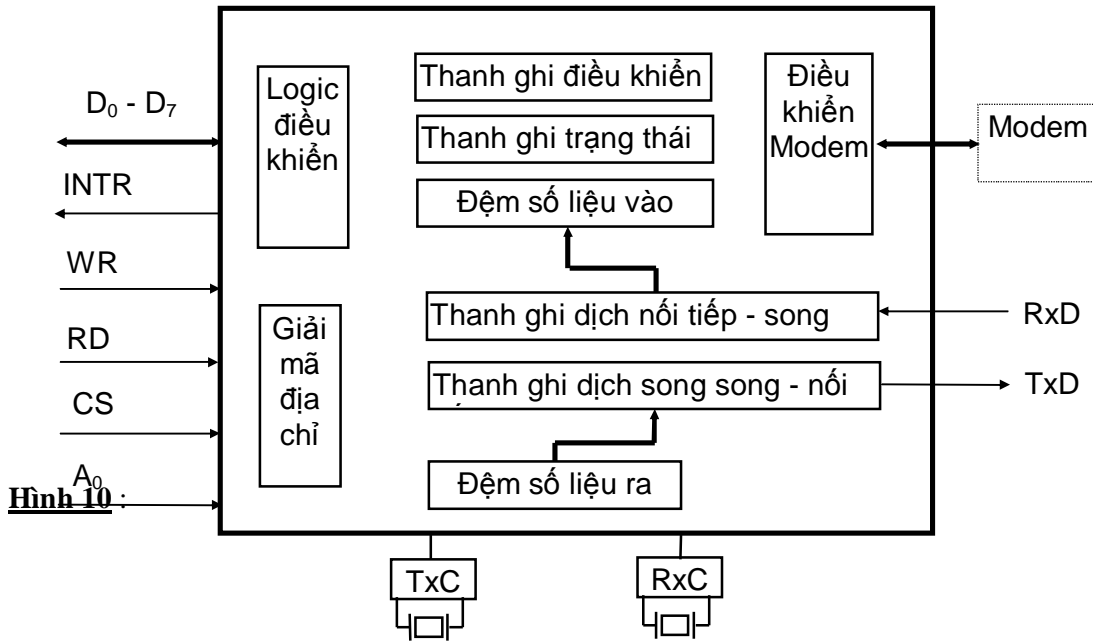


**Hình 9**

### **III. Sơ đồ khối của khối ghép nối:**

Sơ đồ khối của khối ghép nối song song - nối tiếp và nối tiếp - song song như hình 10, gồm:

- Các thanh ghi dịch nối tiếp song song để biến đổi song song - nối tiếp và nối tiếp - song song.
- Các thanh ghi đệm song song vào và ra.
- Thanh ghi điều khiển, định chế độ, số lượng bit tin, số lượng bit Stop.
- Thanh ghi trạng thái để đọc các trạng thái của modem và của khối ghép nối.
- Bộ điều khiển modem.



**Hình 10 :**



#### **IV. Vi mạch trao đổi tin song song - nối tiếp không đồng bộ 8251A:**

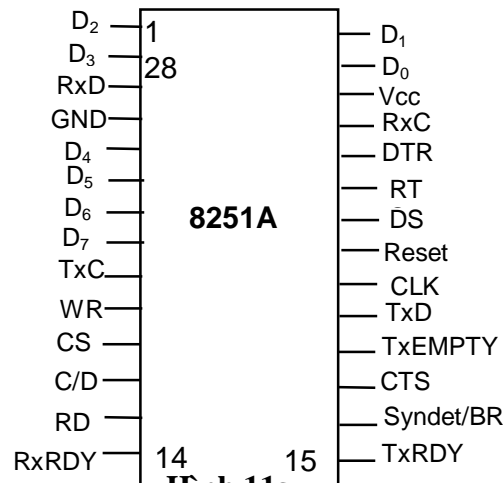
8251A là một USART có khả năng thực hiện Protocol đồng bộ hoặc bất đồng bộ. Ở đây ta chỉ sử dụng để truyền bất đồng bộ.

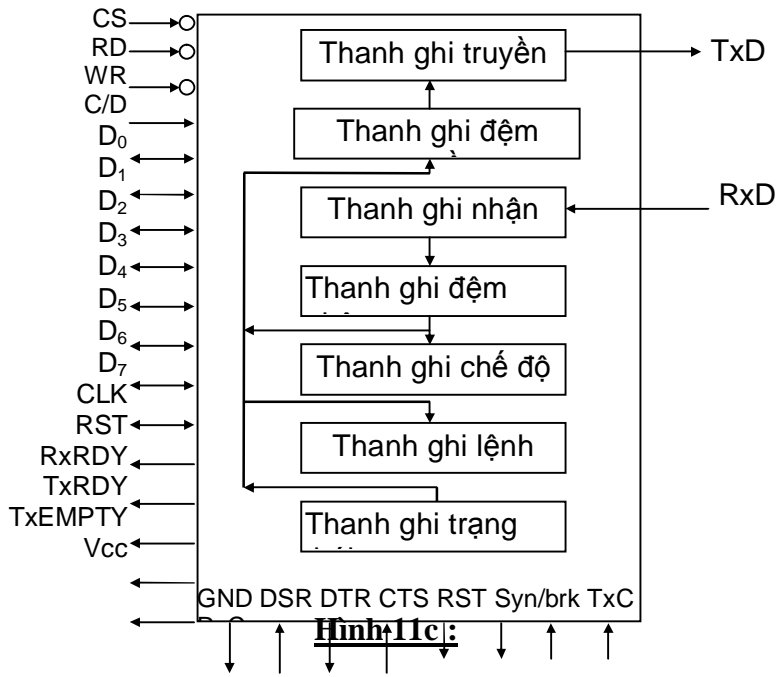
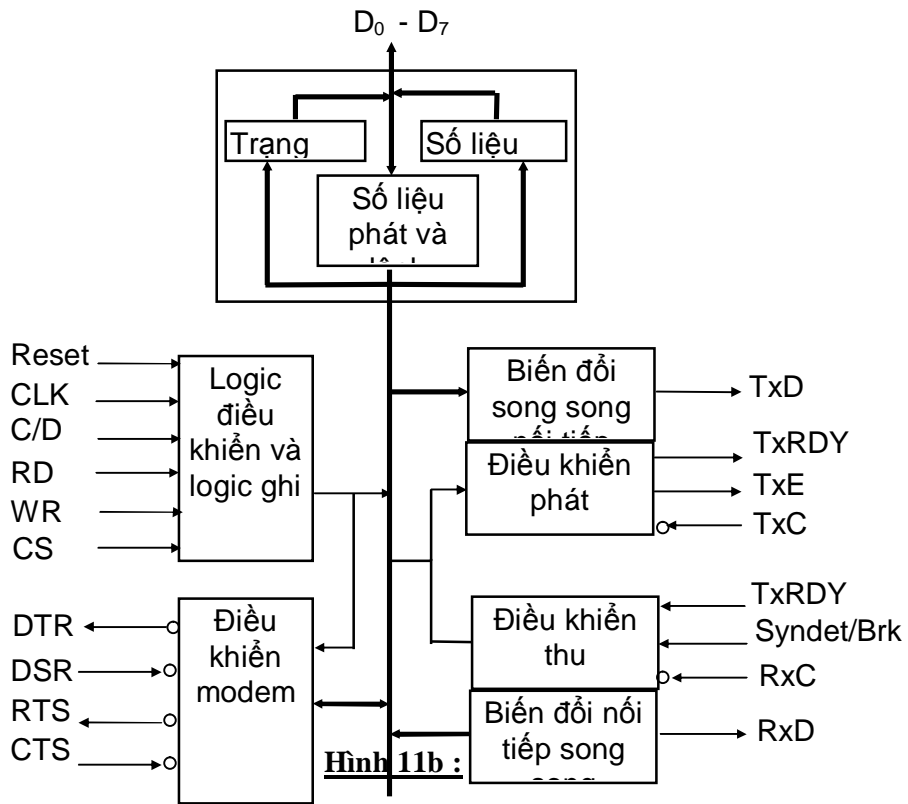
Hãng Intel mô tả 8251A như sau :8251A được thiết kế để truyền dữ liệu với họ  $\mu$ P của Intel. USART nhận các ký tự dữ liệu từ  $\mu$ P ở dạng song song, sau đó đổi chúng thành dạng dữ liệu nối tiếp để phát đi. Đồng thời 8251A có thể thu dòng dữ liệu nối tiếp và đổi chúng thành các ký tự dữ liệu song song gửi đến  $\mu$ P. USART sẽ báo cho  $\mu$ P khi nào có thể nhận 1 ký tự mới từ  $\mu$ P để phát hoặc khi nào đã thu được một ký tự để cho  $\mu$ P đọc.

##### **1. Sơ đồ chân và sơ đồ khối của 8251A:**

Sơ đồ chân và sơ đồ khối của USART 8251A như hình 11 a, b và các thanh ghi như hình 11c.

8251A được chế tạo bằng kỹ thuật NMOS và có 28 chân, tốc độ truyền dữ liệu lên tới 64 Kpbs. Khác với 6402, ở 8251A tất cả dữ liệu đến và từ  $\mu$ P đều đi qua Bus dữ liệu (Data Bus) nối với các chân D<sub>7</sub> - D<sub>0</sub>.





**Trong đó:**

- CS : Chip Select - chọn vi mạch
- RD : Read - đọc
- WR : Write - ghi
- C/D : Control/Data - điều khiển/số liệu
- CLK : Clock - Nhịp
- D<sub>0</sub> - D<sub>7</sub> : Data bus I/O - Bus dữ liệu vào/ra
- TxC : Transmit Clock - Nhịp truyền
- RxC : Receiver Clock - Nhịp nhận
- RST : Reset - Xóa
- DTR : Data Terminal Ready - Số liệu của TBĐC sẵn sàng
- DSR : Data Set Ready - Bộ số liệu sẵn sàng
- RTS : Request To Send - yêu cầu gửi
- CTS : Clear To Send - Xóa gửi
- TxD : Transmit Data - Số liệu truyền
- RxD : Receiver Data - Số liệu nhận
- TxRDY : Transmit Ready - Sẵn sàng truyền
- RxRDY : Receiver Ready - Sẵn sàng nhận
- TxEMPTY : Transmit Register Empty - Thanh ghi truyền rỗng
- Syn/BRK : Synchron / Break Detect - Đồng bộ / phát hiện đứt đoạn tin.

**a. Nhóm tín hiệu ghép nối với vi xử lý gồm:**

- CS (chân 11) nối với bộ giải mã địa chỉ (A<sub>1</sub> - A<sub>n</sub>) để chọn hai thanh ghi (A<sub>0</sub> điều khiển trạng thái, A<sub>0</sub> = 0 đệm số liệu).
- C/D (chân 12) nối với đường dây địa chỉ A<sub>0</sub> để chọn một trong hai cặp thanh ghi trên.
- WR (chân 10) nối với đường dây WR của vi xử lý.
- RD (chân 13) nối với đường dây RD của vi xử lý.
- CLK (chân 20) nối với đường dây CLK của vi xử lý.
- Reset (chân 21) nối với đường dây Reset của vi xử lý.
- D<sub>0</sub> - D<sub>7</sub> (chân 27, 28, 1, 2, 5, 6, 7, 8) nối với các đường dây số liệu D<sub>0</sub> - D<sub>7</sub> của vi xử lý.

**b. Nhóm tín hiệu ghép nối với Modem:**

- DTR (chân 24), RTS (chân 23)
- DSR (chân 22), CTS (chân 17)

**c. Nhóm tín hiệu ghép nối với đường dây truyền - nhận và khối ghép nối:**

- TxEMPTY (chân 18) thanh ghi đệm truyền rỗng.
- TxD (chân 19) cho số liệu truyền.
- RxD (chân 3) cho số liệu nhận.
- TxRDY (chân 15) báo số liệu truyền sẵn sàng.
- RxRDY (chân 14) báo số liệu nhận sẵn sàng.
- Syn/Break : chỉ đồng bộ/ đứt dòng tin.

**d. Nhóm tín hiệu ghép nối với máy phát xung nhịp:**

- TxC : nhịp truyền.
- RxC : nhịp nhận.

## **2. Khởi động 8251A:**

Sau khi Chip 8251A được Reset, một Byte được ghi vào thanh ghi chọn Mode để chọn Protocol mong muốn. Có thể Reset bằng cách dùng chân RST (Reset cứng lúc bật nguồn) hoặc khi ghi bit  $D_1 = 1$  vào thanh ghi lệnh (Reset mềm).

Sau khi thiết lập Mode hoạt động việc phát và thu được điều khiển bằng cách ghi định kỳ từ điều khiển lệnh vào thanh ghi lệnh bao gồm các bước sau:

- Reset chip 8251A (bằng phần cứng hoặc phần mềm)
- Ghi từ chọn Mode mong muốn vào thanh ghi Mode.
- Ghi từ lệnh mong muốn vào thanh ghi lệnh.

□ Phát một ký tự : Để phát một ký tự, bit TxEN trong thanh ghi lệnh phải ở mức logic 1 và chân CTS phải ở mức 0:

+ Đợi cho đến khi chân TxRDY ở mức cao hoặc cho đến khi bit TxRDY trong thanh ghi trạng thái là 1.

+ Ghi ký tự cần phát vào thanh ghi đệm phát.

\* Thu một ký tự : Để thu một ký tự và kiểm tra trạng thái lỗi của ký tự này, ta cần thực hiện các bước sau :

+ Đợi cho đến khi chân RxRDY ở mức cao hoặc cho đến khi bit RxRDY trong thanh ghi trạng thái là một.

+ Đọc trạng thái lỗi từ thanh ghi trạng thái (có thể kết hợp với các bước trước - khi đọc bit RxRDY).

+ Đọc ký tự từ thanh ghi đệm thu.

+ Reset trạng thái lỗi bằng cách ghi bit ER = 1 trong thanh ghi lệnh.

Tốc độ phát và thu được quyết định bởi tần số của xung Clock đưa vào chân RxC/TxC chia theo hệ số (1, 16 hoặc 64) đã chọn trong từ chọn Mode.

## **3. Các thanh ghi của chip 8251A :**

Cũng như CPU, Chip UART chứa đựng một số thanh ghi hoặc bộ nhớ trong, có ba kiểu thanh ghi:

- Những thanh ghi điều khiển: Chúng nhận những lệnh từ CPU và không tự động thay đổi trạng thái.

- Những thanh ghi trạng thái: Chúng thông báo đến CPU những gì xảy ra trong UART.

- Những thanh ghi đệm: Chúng lưu giữ những ký tự chuẩn bị phát đi hoặc chờ xử lý.

Cách truy xuất các thanh ghi phụ thuộc vào cấu trúc của máy tính mà UART được cài đặt. Trong trường hợp máy IBM PC, những giá trị được đặt vào bên trong các thanh ghi nghĩa là dùng lệnh OUT từ chip CPU gửi đến những địa chỉ tương ứng của các thanh ghi. tương tự, những thanh ghi được truy xuất theo cách đọc nghĩa là dùng lệnh IN với các địa chỉ tương ứng.

### **a. Các thanh ghi điều khiển:**

Có 4 thanh ghi điều khiển dùng để nhận lệnh từ CPU:

#### **a1. Thanh ghi điều khiển đường dây: (Line Control Register)**

Thanh ghi này được dùng để SET các thông số liên lạc. Mỗi bit của thanh ghi này có ý nghĩa và được mô tả như sau:

Bit	Ý nghĩa
0	Hai bit này dùng để xác định chiều dài của một từ
1	
2	Số bit Stop
3	Cho phép kiểm tra chẵn lẻ
4	Chọn kiểm tra chẵn lẻ
5	Chọn bit chẵn lẻ
6	Ngắt
7	Bit truy xuất và chốt số liệu

**Mô tả :**

- Bit 0 và 1 : chỉ chiều dài của một từ như sau :

Bit 0	Bit 1	Số ký tự
0	0	5
0	1	6
1	0	7
1	1	8

- Bit 2 : Xác định số STOP bit. Nếu Bit 2 là 0, một Stop bit được tạo ra. Nếu bit 2 là 1, 2 Stop bit được tạo ra trừ khi chiều dài ký tự là 5 (tức là bit 0 và bit 1 bằng 00), lúc này sẽ tạo ra 1.5 Stop bit.

- Bit 3, 4, 5 là những bit lựa chọn sự kiểm tra chẵn lẻ (trong đồ án này không sử dụng nên luôn được Set bằng 0).

- Bit 6 : là bit tạo ra một lệnh ngắt, khi được Set lên 1 nó sẽ treo ngõ ra OUT ở mức logic 0 (SPACING) cho tới khi được SET trở lại bằng 0 (Bit này không được dùng nên luôn luôn được Set bằng 0).

- Bit 7 : là bit duy nhất và chốt số chia (DLAB : Division Latch Access Bit). nó phải được SET lên 1 khi truy xuất việc chốt số chia, tức là lúc cài tốc độ truyền theo BAUD\_RATE. Khi SET trở về 0, nó sẽ cho phép các thao tác đọc hoặc ghi từ vùng đệm thu, vùng đệm phát hoặc thanh ghi cho phép ngắt.

**a2. Thanh ghi điều khiển modem: (Modem Control Register)**

Thanh ghi này điều khiển những tín hiệu bắt tay gửi ra ngoài từ UART. Mỗi bit của thanh ghi này được mô tả như sau :

Bit	Chữ viết tắt	Tên gọi
0	DTR	Dữ liệu trạm cuối sẵn sàng
1	RTS	Yêu cầu gửi
2	OUT1	Ngõ ra 1
3	OUT2	Ngõ ra 2
4	LOOP	Kiểm tra chế độ vòng lặp

- Bit 0 : thường dùng để SET ngõ ra DTR về mức logic 0, cho phép các thiết bị từ xa gửi tín hiệu đến chúng ta hoặc khi nó được SET trở về mức logic 1 yêu cầu thiết bị xa không gửi đến chúng ta nữa. Trong đồ án này ta sử dụng Bit này để gửi ra mạch khi nhận được tín hiệu gọi kết nối từ đối phương. Lúc đó, nó phải được Set lên 1 để xem như bắt tay phần cứng với đối phương.

- Bit 1,2,3,4 : không dùng nên luôn được set bằng 0
- Bit 5,6,7 : luôn Set = 0

**a3. Thanh ghi cho phép ngắt : (Interrupt Enable Register )**

Tám bit thanh ghi này cho phép 4 kiểu điều khiển ngắt tác động lên "chip ngắt" bằng ngõ ra INTRPT. Khi RESET các bit từ 0 đến 3 của thanh ghi này, ta có thể bỏ kiểu ngắt hệ thống của nó. Khi SET các bit từ 0 đến 3, ta chọn kiểu ngắt cho phép.

Bit	Kiểu ngắt
0	Cho phép ngắt để thu dữ liệu
1	Cho phép ngắt báo vùng đệm phát rỗng
2	Cho phép ngắt báo trạng thái đường dây
3	Cho phép ngắt báo trạng thái Modem
4	Bằng 0
5	Bằng 0
6	Bằng 0
7	Bằng 0

**a4. Thanh ghi chốt số chia BAUD RATE:**

BAUD\_RATE được cài đặt phụ thuộc vào 2 thanh ghi mà con số của chúng phải được chia bởi xung nhịp 1.8432 MHz. Kết quả cho ra tần số bằng 16 lần BAUD\_RATE. Hai thanh ghi này mang 2 byte: 1 byte chốt số chia có nghĩa thấp nhất (LSB), và 1 byte chốt số chia có nghĩa cao nhất (MSB). Những số chia này dùng để tạo ra những tốc độ theo BAUD\_RATE khác nhau.

BAUD_RATE	DECIMAL	HEX	MSB	LSB
50	2304	900	9	00
100	1152	480	4	80
300	384	180	1	80
1200	96	60	0	60
2400	48	30	0	30
4800	24	18	0	18
9600	12	0C	0	0C

Tốc độ dây BAUD\_RATE trong bảng không phải là cố định. Những giá trị ở khoảng giữa chúng cũng có thể được tạo ra bằng cách chọn số chia tương ứng.

**b. Các thanh ghi trạng thái:**

Có 3 thanh ghi trạng thái tường trình đến CPU những gì đang xảy ra ở những vùng khác nhau của UART.

**b1. Thanh ghi trạng thái đường dây: (Line Status Register)**

Tám bit thanh ghi này tường trình đến CPU những thông tin trạng thái về sự chuyển biến dữ liệu.

Bit	Chữ viết tắt	Tên gọi
0	DR	Data set ready
1	OE	Overrun error
2	PE	Parity error

3	FE	Frame error
4	BI	Break interrupt
5	THRE	Transmitter hold register empty
6	TSRE	Tx shift register empty
7	= 0	

- Bit 0 : Đây là bit thông báo dữ liệu thu đã sẵn sàng. Nó SET lên một bất kỳ khi nào 1 ký tự vừa mới được nhận và chuyển vào bên trong thanh ghi vùng đệm thu. Bit 0 có thể Reset trở về 0 bằng việc đọc dữ liệu thu từ thanh ghi đệm thu hoặc ghi mức 0 lên nó.

- Bit 1 : Bit này thông báo lỗi tràn. Nó thông báo là dữ liệu bên trong thanh ghi đệm thu chưa được đọc trước khi ký tự kế tiếp được đến, vì vậy nó sẽ phá hỏng ký tự trước.

- Bit 2 : là bit báo lỗi cực tính, thông báo đã thu một ký tự không đúng chuẩn (hoặc lẻ) như đã được chọn trong Bit chọn lựa trong Bit chọn lựa chuẩn lẻ ở thanh ghi điều khiển đường dây.

- Bit 3 : Là bit báo lỗi Frame, nó thông báo đã thu một ký tự không có Stop bit.

- Bit 4 : là một ngắt của phần thu, nó được Set lên 1 bất kỳ khi nào ngõ vào thu dữ liệu ở mức 0 lâu hơn thời gian chiều dài của 1 từ.

- Bit 5 : Bit này thông báo thanh ghi đệm phát rỗng. Nó báo cho bộ điều khiển là sẵn sàng nhận một ký tự mới để phát đi. Bit này Set lên mức 1 khi 1 ký tự nữa mới được chuyển từ thanh ghi lưu giữ phần phát. Nó được Set trở về mức 0 khi CPU nạp một từ mới vào thanh ghi lưu giữ phần phát.

- Bit 6 : Bit này cũng báo phần đệm phát rỗng. Nó được Set lên 1 bất kỳ khi nào mà cả hai thanh ghi đệm phát và thanh ghi dịch chuyển phần phát đều rỗng. Nó được Reset trở về mức 0 khi THR và TSR được chứa ký tự dữ liệu.

- Bit 7 : không dùng.

### **b2 . Thanh ghi trạng thái Modem: (Modem Control Register)**

Thanh ghi này cho những thông tin về trạng thái của những đường bắt tay. Mô tả như sau:

Bit	Tên	Ý nghĩa
0	Delta CTS	CTS bị thay đổi trạng thái
1	Delta DSR	DSR bị thay đổi trạng thái
2	TERI	Đường dây đồ chuông đổi từ ON lên OFF
3	Delta RLSD	Tín hiệu tách sóng bị thay đổi
4	CTS	Ngõ vào xóa để gọi ở mức cao
5	DSR	Ngõ vào dữ liệu sẵn sàng ở mức cao
6	RI	Chỉ thị chuông ở mức cao
7	RLSD	Đường dây tách sóng ở mức cao

### **b3. Thanh ghi đặc tính ngắt: (Interrupt Identification Register)**

Thanh ghi này cung cấp thông tin về trạng thái hiện hành của những ngắt được phát đi.

Bit 0 Set lên 1 là không có một ngắt nào gọi đi. Khi bit 0 ở mức 0, bit 1 và bit 2 chỉ ra ngắt nào đã được gọi đi theo bảng sau :

Bit			Mức ưu tiên	Kiểu ngắt	Nguyên nhân ngắt	Reset lỗi ngắt
2	1	0				
0	0	1	-	0	0	-
1	1	0	Cao nhất	Trạng thái đường dây thu	- Lỗi tràn - Lỗi cực tính - Lỗi khung - Break interrupt	Đọc thanh ghi trạng thái đường dây
1	0	0	Thứ hai	Cho phép thu dữ liệu	Cho phép thu dữ liệu	Đọc thanh ghi đệm thu
0	1	0	Thứ ba	Thanh ghi lưu giữ phần phát rỗng	Thanh ghi lưu giữ phần phát rỗng	Đọc IIR hoặc ghi vào thanh ghi lưu giữ phần phát
0	0	0	Thứ tư	Trạng thái Modem	- CTS - DSR - RI Đã nhận tin	Đọc thanh ghi trạng thái Modem

### **c. Các thanh ghi đệm:**

Kiểu thanh ghi thứ 3 trong UART là các thanh ghi đệm, có hai thanh ghi đệm: Thanh ghi lưu giữ phần phát (Transmitter Holding Register) và thanh ghi đệm thu (Receiver Buffer Register).

#### **c1. Thanh ghi đệm thu:**

Thanh ghi này lưu giữ ký tự sau cùng vừa mới nhận được. Mỗi lần nó được đọc, thanh ghi trạng thái đường dây sẽ chỉ thị vùng đệm thu rỗng cho tới khi nhận được một ký tự khác. Nếu ký tự thứ hai nhận được trước ký tự thứ nhất đọc xong, mỗi lỗi tràn sẽ được thông báo.

#### **c2. Thanh ghi lưu giữ phần phát:**

Thanh ghi lưu giữ phần phát lưu lại ký tự kế tiếp để chuẩn bị phát đi. Ký tự được đặt vào bằng chương trình của người viết. Thanh ghi trạng thái đường dây sẽ thông báo khi một ký tự vừa được phát đi.

**Địa chỉ của các thanh ghi được chọn như sau:**

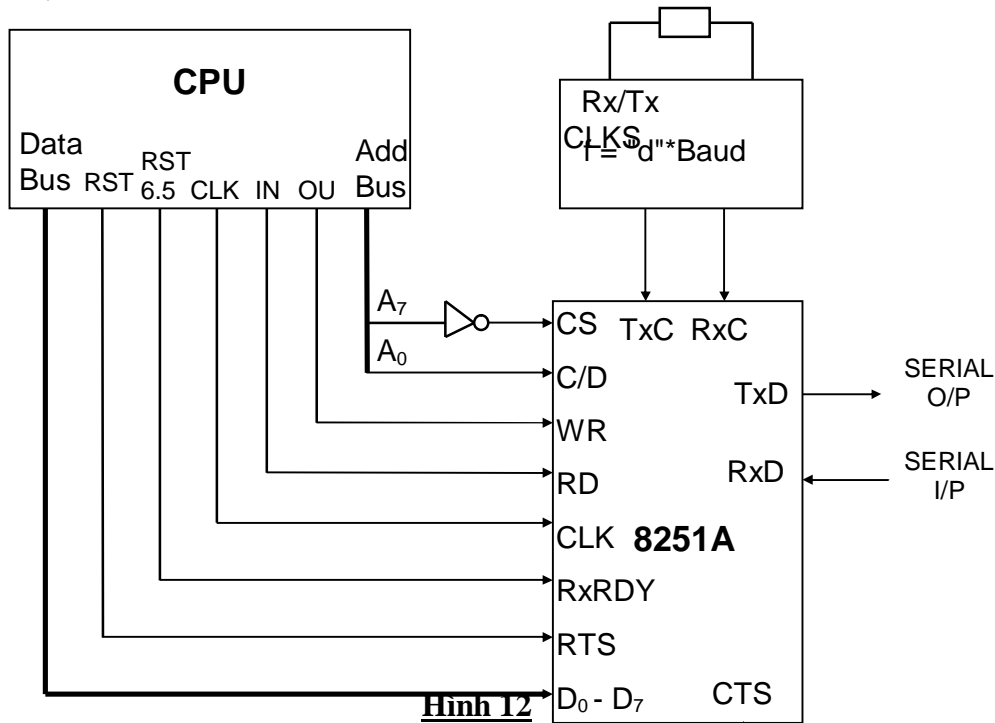
Địa chỉ	Thanh ghi được chọn
XF8	Thanh ghi đệm phát
XF8	Thanh ghi đệm thu
XF8	Thanh ghi chốt số chia LSB
XF9	Thanh ghi chốt số chia MSB
XF9	Thanh ghi cho phép ngắt
XFA	Thanh ghi đặc tính ngắt
XFB	Thanh ghi điều khiển đường dây
XFC	Thanh ghi điều khiển modem
XFD	Thanh ghi trạng thái đường dây



**Trong đó :** Chữ X dùng để chọn cổng COM1 hoặc COM2. X = 3 dùng cho COM1, X = 2 dùng cho COM2.

**4. Giao tiếp giữa 8251A với bộ vi xử lý:**

Hình 12 trình bày kết nối giao tiếp điện hình giữa bộ vi xử lý với USART 8251A.



**Hình 12**

Giao tiếp của 8251A sử dụng đặc tính xuất nhập bộ tích lũy của Intel 8085. Chân RxRDY lên mức cao High sẽ gây ra một ngắt đưa tới  $\mu P$  báo cho CPU biết đã thu một ký tự. Chân TxRDY không được dùng để tạo ra ngắt CPU như trên. Trong trường hợp này  $\mu P$  phải hỏi vòng bit TxRDY của thanh ghi trạng thái trước khi một ký tự mới để phát.

**\* Hỏi vòng (Polling) và sử dụng ngắt để xuất nhập:**

Các chương trình xuất nhập trong hệ thống máy tính thường được phân loại như : hỏi vòng hoặc điều khiển ngắt. Hỏi vòng có nghĩa là  $\mu P$  sẽ ngưng các hoạt động khác một cách định kỳ và hỏi UART đã nhận dữ liệu mới chưa hoặc sẵn sàng phát ký tự kế tiếp chưa. Công việc này thường đòi hỏi  $\mu P$  phải đọc thanh ghi trạng thái của UART. Một chương trình điều khiển ngắt bao hàm ý nghĩa  $\mu P$  vẫn tiếp tục các hoạt động khác và chỉ ngưng khi trả lời một tín hiệu ngắt được gọi tới từ UART.

Hỏi vòng để quyết định trạng thái của UART hạn chế sự cần thiết phải có ngắt phần cứng nhưng hiệu quả sử dụng thời gian làm việc của  $\mu P$  sẽ thấp hơn. Cũng vậy, khi sử dụng hỏi vòng để quyết định trạng thái thu phải hết sức cẩn thận để đảm bảo rằng lỗi tràn không xảy ra. Sự quyết định dùng các ngắt hoặc hỏi vòng để xuất nhập đều phải được cân nhắc chi từng ứng dụng cụ thể,

điều này dựa vào các yếu tố như độ phức tạp của phần cứng, phần mềm và các ưu tiên của nhiệm vụ hệ thống.

**Tóm lại:**

\* Đối với Teletype vì yêu cầu truyền không cao lắm, hơn nữa máy Teletype không thể nào phát đi hoặc thu nhận các Parity bit hoặc kiểm tra các bit này được, nên ta không Set các bit parity.

\* Như đã nêu ở phần trước, vì Teletype có một sự bắt tay phần cứng. Để làm công việc này ta chọn:

- DSR (Data Set Ready): dùng để nhận biết tín hiệu kết nối gọi đến từ máy Teletype tức là khi điện áp đường thu của mạch từ +60V (logic 0) chuyển thành -60V (logic 1) thì DSR sẽ chỉ thị.

- DTR (Data Terminal Ready): sau khi nhận được tín hiệu kết nối, dùng DTR để xuất ra đường phát của mạch từ logic 0 → logic 1 để gọi đến đối phương hoàn thành công việc bắt tay phần cứng.

## CHƯƠNG V : GIỚI THIỆU VỀ NGẮT CỦA MÁY VI TÍNH

### I. Các loại ngắt của một máy vi tính PC:

Người ta chia ngắt thành hai loại : Ngắt cứng và ngắt mềm.

#### 1. Ngắt cứng :

Còn gọi là ngắt ngoài vì do nguyên nhân bên ngoài. Vi xử lý có các lỗi vào dành cho ngắt ngoài. Khi có tín hiệu vào các lỗi này, vi xử lý đang thực hiện lệnh của chương trình sẽ bị dừng.

#### \* Ngắt NMI (Non Maskable Interrupt - Ngắt không che được)

Khi có ngắt này, vi xử lý dừng chương trình sau lệnh đang thực hiện, thanh ghi chỉ thị lệnh và thanh ghi chỉ thị Flag được lưu giữ, hai bit IF (Interrupt Flag) và TF (Trap Flag) bị xóa về 0 để cấm ngắt ngoài tiếp theo và không có bất. Muốn cho phép hay không cho phép ngắt này xảy ra, chúng ta dùng một Trigo để mắc lỗi vào ngắt trước khi đưa vào ngắt trước khi đưa vào lỗi vào ngắt NMI.

#### \* Ngắt INMR:

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

- Được xóa về 0 bởi lệnh CLI (Clear Interrupt).

- Được xác lập lên 1 bởi lệnh STI.

- Thường được nối với lỗi ra yêu cầu ngắt của vi mạch xử lý ưu tiên ngắt (8214, 8159).

#### \* Ngắt Reset:

Đây thực sự không phải là ngắt, chỉ tương ứng với việc treo chương trình tạm thời. Khi có tín hiệu từ ngoài vào lỗi vào Reset hay có lệnh Reset của chương trình, tất cả các thanh ghi của vi xử lý bị xóa về không, trừ thanh ghi đoạn (CS) được nạp địa chỉ đầu tiên của chương trình tự kiểm tra Port.

#### 2. Ngắt mềm: (Hay ngắt bên trong do lệnh của chương trình)

Do vi xử lý gặp các lệnh gây ra ngắt hoặc tình huống đặc biệt khi thực hiện lệnh (ngắt logic và ngắt của hệ điều hành).

- Ngắt do lệnh : Đó là ngắt khi thực hiện các lệnh CALL, HLT, INT.

- Ngắt logic hay các ngoại trừ : xảy ra khi gặp các tình huống đặc biệt sau

:

+ Chia một số cho 0 (vector 0).

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

+ Thực hiện từng bước (vector 1).

+ Điểm dừng chương trình do người sử dụng định hướng (vector

3).

- Ngắt của hệ điều hành: Đó là các ngắt do hệ điều hành quy định để phục vụ trao đổi tin của các thiết bị ngoài như INT 10, INT 16, INT 21 . . .

### II. Thủ tục xử lý ngắt của chương trình:

Khi có một tín hiệu yêu cầu ngắt chương trình đưa vào chân yêu cầu ngắt (INTR), quá trình ngắt chương trình được diễn ra như sau:

- Lưu giữ tin về trạng thái của vi xử lý lúc có tín hiệu yêu cầu ngắt và nơi chương trình chính bị gián đoạn.
- Vi xử lý gửi tín hiệu xác nhận hay cho phép ngắt - INTA và đọc vector ngắt.
- Chuyển sang chương trình con phục vụ ngắt.
- Trở về chỗ chương trình chính bị ngắt và tiếp tục thực hiện chương trình đó.

### **1. Lưu giữ tin về chỗ bị ngắt chương trình:**

Ở cuối mỗi chương trình lệnh, VXL kiểm tra xem có yêu cầu ngắt nào gọi đến hay không. Nếu có yêu cầu, VXL tiến hành lưu trữ tin về nơi bị ngắt chương trình. quá trình lưu giữ tin tiến hành theo các bước sau:

- Giảm con trỏ ngăn xếp.
- Cắm khóa vào ngắt INTR bằng cách xóa bit IF ở thanh ghi cờ.
- Xóa bit cờ bẫy trong thanh ghi cờ.
- Giảm SP đi 2 và nạp nội dung của thanh ghi mạng lệnh hiện hành vào mảng nhớ ngăn xếp có địa chỉ trên.
- Giảm SP đi 2 và nạp nội dung thanh ghi con trỏ lệnh hiện hành vào bộ nhớ ngăn xếp có địa chỉ trên.

### **2. Gửi tín hiệu cho phép ngắt và đọc vec tơ ngắt:**

Sau khi đã lưu trữ tin về vị trí bị ngắt của chương trình chính, VXL gửi tín hiệu xác nhận ngắt INTA (Interrupt Acknowledge) cho khối ghép nối của thiết bị ngoài. Tùy cách tổ chức ngắt và tạo vector ngắt, VXL sử dụng tín hiệu này để đọc vector ngắt tương ứng của khối ghép nối vào thanh ghi chứa A. VXL đọc nội dung của ô nhớ có địa chỉ là vector ngắt để biết được địa chỉ đầu tiên của chương trình con phục vụ ngắt.

### **3. Thực hiện chương trình con phục vụ ngắt:**

Đó là chương trình mà địa chỉ của lệnh đầu tiên nằm trong ô nhớ có địa chỉ là vector ngắt. Kết thúc chương trình con này, có lệnh trở về để VXL tiếp tục thực hiện chương trình chính.

### **4. Thực hiện chương trình chính:**

Sau khi gặp lệnh trở về, VXL tiến hành đọc và hồi phục các tin của VXL lúc bị ngắt chương trình đã ghi nhớ ở chỗ ngắt chương trình. Quá trình đọc ra này xảy ra ngược lại với quá trình ghi vào và nội dung:

- Thanh ghi con trỏ lệnh trở về độ dời của địa chỉ lệnh tiếp theo của chương trình chính bị ngắt trong mảng nhớ lệnh.
- Thanh ghi mảng lệnh về địa chỉ đoạn đầu tiên của vùng nhớ dành cho chương trình chính bị ngắt.
- Thanh ghi Flag lúc bị ngắt chương trình.

## **III. Giới thiệu về cách sử dụng IRQ<sub>4</sub>:**

IRQ<sub>4</sub> là một trong những đường ngắt cứng của máy tính. Trong máy IBM\_PC có 8 đường ngắt cứng và máy tính IBM\_PC\_AT có 16 đường ngắt cứng, được mô tả như sau:

IRQ	Thiết bị
-----	----------

NMI	Ngắt không che
0	Ngắt thời gian
1	Ngắt bàn phím
2	Dự trữ
3	COM2
4	COM1
5	Đĩa cứng
6	Đĩa mềm
7	LPT1

Những đường ngắt này không nối được trực tiếp tới chip CPU, nhưng được thông qua chip xử lý ngắt là PIC\_8259A (Programmable Interrupt Controller). Trong máy IBM PC\_AT có hai bộ điều khiển này. Nó có những mức ưu tiên cho các ngắt để tránh tình trạng lộn xộn xảy ra khi nhiều ngắt đến không theo một trật tự nào cả. Các mức ưu tiên này trong máy IBM PC được chỉ định ở cao nhất là IRQ<sub>0</sub> và thấp nhất là IRQ<sub>7</sub>. Nếu nhiều thiết bị đều yêu cầu sự chú ý bằng cách nâng các đường ngắt của nó lên, bộ điều khiển ngắt sẽ có nhiều ngõ vào ở mức cao thì nó sẽ gọi chúng đến CPU với trật tự của mức độ ưu tiên.

Trong PIC có một thanh ghi cho phép ngắt từ thiết bị gọi đến. Mặc nhiên, IRQ<sub>4</sub> và IRQ<sub>3</sub> là không cho phép. Vì vậy, nếu muốn dùng ngắt cho việc liên lạc thông tin thì phải chỉ thị cho PIC cho phép các đường ngắt tương ứng. Đây là công việc đọc thanh ghi bằng lệnh IN với địa chỉ cổng là 21H và Set Off bit tương ứng (bit 4 cho IRQ<sub>3</sub> và bit 5 cho IRQ<sub>4</sub>) và OUT trở lại giá trị ra Port 21H. Như vậy thì PIC mới cộng thêm các ngắt của UART và cho nó gọi ngắt tới.

Khi PIC nhận được một ngắt, nếu ngắt đó là cho phép và không có một ngắt nào khác gọi đến nữa, nó sẽ đưa mức điện áp dương lên đường ngắt đó đến CPU. Khi đó CPU sẽ công nhận ngắt, nghĩa là trả một tín hiệu về PIC yêu cầu PIC chỉ thị xem ngắt nào vừa mới xảy ra. Lúc đó PIC sẽ gọi một con số (thông qua Data Bus) đến CPU (là con số IRQ + con số 8). Nói cách khác, đối với IRQ<sub>4</sub> nó gọi con số 12 (0Ch), đối với IRQ<sub>3</sub> nó gọi con số 11 (08h), lúc này CPU sẽ thực hiện các phần mục tương ứng là cất địa chỉ chương trình hiện hành vào vùng Stack và thực hiện lệnh Call Far đến bộ nhớ được trỏ bởi vector ngắt của ngắt trong bảng vectơ ngắt. Vì vậy, với ngắt IRQ<sub>4</sub> gọi tới việc CPU thực hiện cũng giống như INT 0Ch của phần mềm.

Nếu bạn không muốn CPU bị phân chia khi có nhiều ngắt, bởi vì nó can thiệp vào một số công việc xấu làm không thể ngắt được, ta có thể chỉ thị đến nó không quan tâm đến cái ngắt đó bằng lệnh STI. Kết quả của lệnh này là chuyển đổi trạng thái cờ IF trong CPU.

Việc Set cờ IF không làm thay đổi bất cứ cái gì trong PIC mà nó vẫn gọi một tín hiệu ngắt đường dây đến CPU khi nó nhận được một ngắt trong những ngắt đường dây. Ngắt đường dây sẽ treo lên mức cao và CPU có thể nhận ra được nguyên nhân của ngắt và thực hiện lệnh STI một lần nữa.

Những ngắt khác gọi đến không nhất thiết sẽ mất đi. Những thiết bị giữ đường IRQ của chúng ở mức cao và PIC hiểu những thiết bị đó đang cần sự chú ý. Nó sẽ chấp nhận gọi các ngắt trong thứ tự ưu tiên của chúng.

Các ngắt có thể bị treo, như vậy kết quả là mất thông tin từ khi thiết bị không tạo ra được ngắt thứ hai cho tới khi ký tự thứ nhất vừa được công nhận.

Có hai lý do tại sao ngắt từ thiết bị không được CPU chú ý ngay lập tức. Thứ nhất là nó bị xung đột với các thiết bị khác có mức ưu tiên cao hơn hoặc có những ngắt đang bị treo. Tuy vậy dùng phương pháp ngắt thường nhanh hơn so với phương pháp hỏi vòng.

Trong chương trình xử lý ngắt phần mềm phải trả lời ngắt đến PIC bằng cách gọi giá trị 20h ra port 20h. Nếu việc này không được thực hiện PIC sẽ không gọi thêm một ngắt nào nữa đến CPU.

#### **IV. Sử dụng ngắt của ROM BIOS:**

ROM BIOS cung cấp 4 hàm xử lý truyền thông tuần tự mà chúng ta có thể truy xuất chúng qua ngắt 14h.

Để sử dụng ngắt này, ta đặt giá trị từ 0 - 3 trong thanh ghi AH để xác định hàm nào sẽ được sử dụng và đặt số thứ tự cổng cần truy xuất vào thanh ghi DX.

Số thứ tự cổng = 0 : COM1

Số thứ tự cổng = 1 : COM2

Ở mức độ chương trình, ta có thể chọn một khối ghép nối tiếp bằng cách gán mã tương ứng vào thanh ghi DX với giá trị:

- 00h cho COM1

- 01h cho COM2

- 02h cho COM3

- 03h cho COM4

**Các hàm của ngắt 14h như sau:**

##### **1. Hàm số 0 (Phục vụ 00h)**

Khởi phát khối ghép nối tiếp, tạo tham số cho cổng nối tiếp:

\* **Vào** :

AH = 0

AL = Tham số truyền

DX = Số thứ tự cổng nối tiếp

\* **Ra** :

AX = Trạng thái đường dây truyền và Modem.

Phục vụ 00h ấn định những thông số khác nhau của khối ghép nối tiếp cũng như RS\_232. Đó là các thông số:

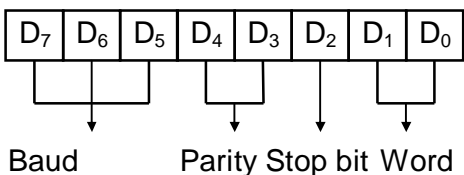
- Số BAUD : Tốc độ trao đổi tin.

- Tính chẵn lẻ.

- Số bit dừng.

- Kích thước ký tự hay số bit tin nối tiếp. những thông số này được tổ hợp trong mã 8 bit, được đặt vào thanh ghi AL, theo thứ tự các bit như hình sau

:



Giá trị trả ra trong AX xem hàm số 3:

- D<sub>7</sub>, D<sub>6</sub>, D<sub>5</sub> mã của vận tốc (tính bằng Baud).
- D<sub>4</sub>, D<sub>3</sub> mã của tính chẵn lẻ.
- D<sub>2</sub> mã của bit dừng.
- D<sub>1</sub>, D<sub>0</sub> mã của các kích thước ký tự.

Các mã trên theo các bảng sau:

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	Vận tốc (bps)
0	0	0	110
0	0	1	150
0	1	0	300
0	1	1	600
1	0	0	1200
1	0	1	2400
1	1	0	4800

D <sub>4</sub>	D <sub>3</sub>	Tính chẵn lẻ
0	0	Không có
0	1	Tính chẵn lẻ
1	0	Không có
1	1	Tính chẵn

D <sub>2</sub>	Số bit dừng
0	Một bit dừng
1	Hai bit dừng

D <sub>1</sub>	D <sub>0</sub>	Kích thước ký tự
0	0	Không dùng
0	1	Không dùng
1	0	7 bit
1	1	8 bit

## 2. Hàm số 1 (Phục vụ 01h)

Hàm này gọi một ký tự đến cổng được chỉ định.

\* **Vào** :

AH = 01h

AL = Ký tự gửi

DX = Số thứ tự cổng nối tiếp

\* **Ra** :

AH = Trạng thái đường truyền

AL = Trạng thái Modem

\* **Goi INT 14h**

Nếu có lỗi xảy ra, bit D<sub>7</sub> sẽ được bật lên 1, các bit còn lại chỉ nội dung lỗi theo mô tả sau đây:

Bit D<sub>1</sub> : Lỗi Overrun

Bit D<sub>2</sub> : Lỗi Parity

Bit D<sub>3</sub> : Lỗi Framing

Bit D<sub>4</sub> : Một Break đã xảy ra

Bit D<sub>5</sub> : THR rỗng

Bit D<sub>6</sub> : THR và Transmitter Shift Register rỗng

Bit D<sub>7</sub> : Quá thời gian

**Chú ý :**

Ký tự sẽ không gửi đến khi đường tín hiệu bắt tay (Handshaking Line) ở trạng thái cao.

**3. Hàm số 2 : (Phục vụ 02h)**

Nhận một ký tự từ cổng chỉ định:

\* **Vào :**

AH = 02h

DX = Số thứ tự cổng nối tiếp

\* **Ra :**

AL = Ký tự nhận

AH = Mã trạng thái

\* **Goi INT 14h**

**Chú ý :**

Hàm này sẽ chờ một ký tự từ cổng cho đến khi nhận được hoặc quá thời hạn chờ. Nếu không có ký tự nào được nhận hoặc việc nhận có lỗi thì bit D<sub>7</sub> của AH lên 1.

**4. Hàm số 3 (Phục vụ 03h)**

Lấy trạng thái của cổng nối tiếp.

Hàm này trả ra thông tin cụ thể và trạng thái của Modem và trạng thái của đường truyền của cổng chỉ định.

\* **Vào :**

AH = 03h

DX = Số thứ tự của cổng nối tiếp

\* **Ra :**

AH = Trạng thái đường truyền

AL = Trạng thái Modem

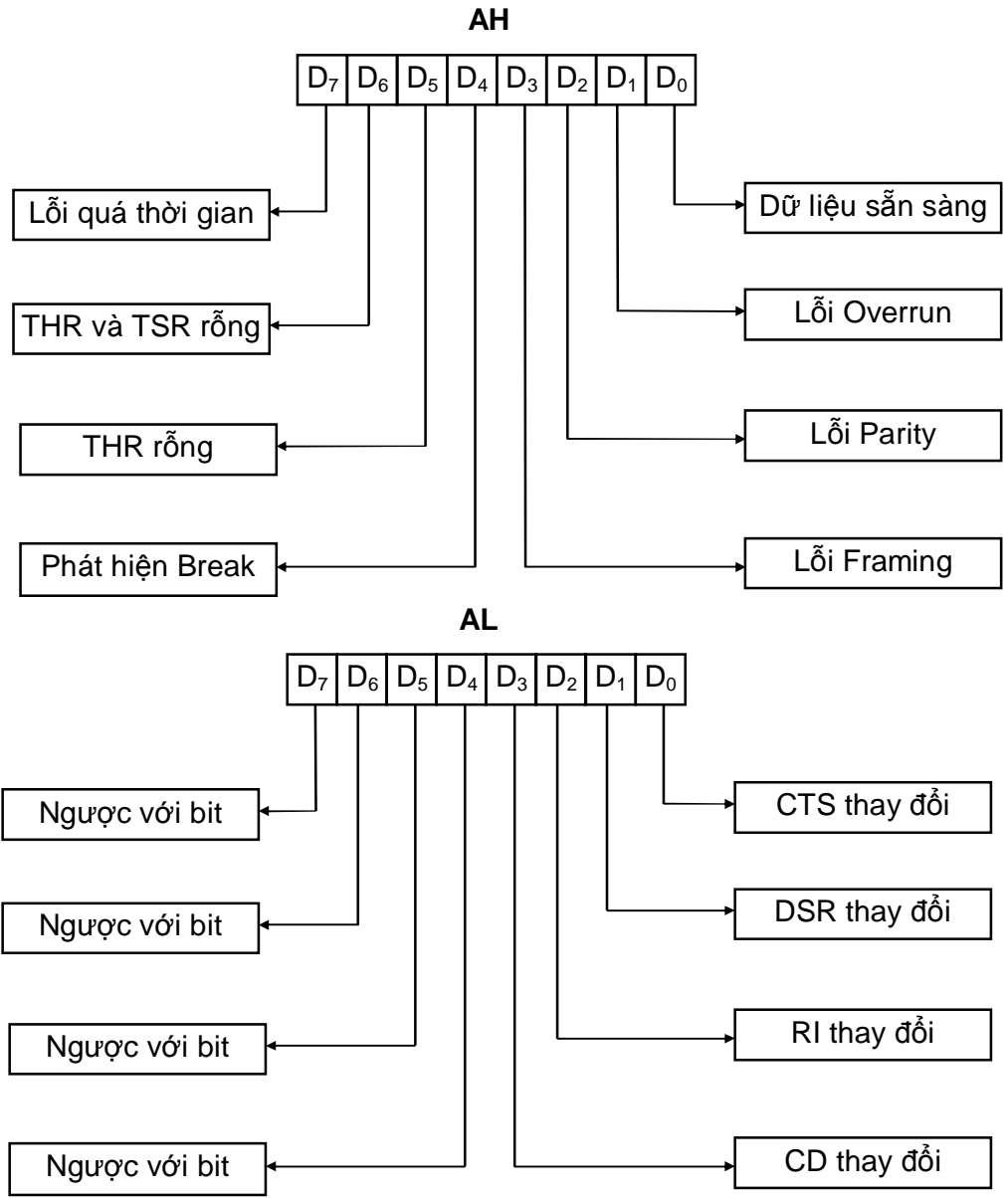
\* **Goi INT 14h.**

Kết quả của chương trình là :

- Trạng thái của đường dây được đặt trong thanh ghi AH, có các bit như hình dưới.

- Trạng thái của Modem được đặt trong thanh ghi AL.



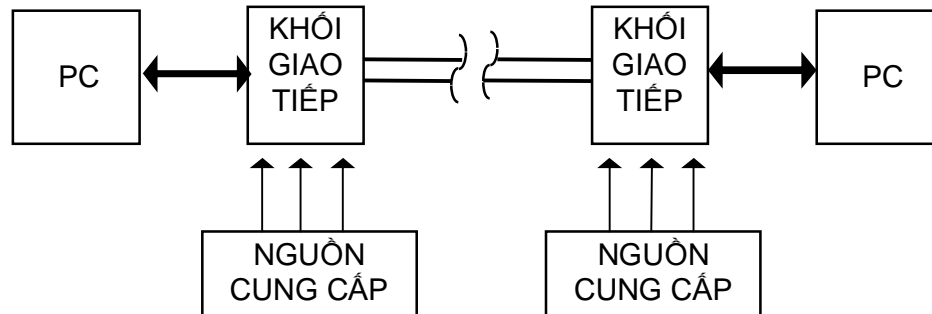


Bit	Ý nghĩa
D <sub>7</sub>	Vượt quá độ trễ 0 = không có sai số 1 = có sai số
D <sub>6</sub>	Thanh ghi dịch chuyển 0 = thanh ghi bận 1 = thanh ghi rỗi
D <sub>5</sub>	Thanh ghi đợi 0 = thanh ghi bận 1 = thanh ghi rỗi
D <sub>4</sub>	Ngắt bởi tín hiệu Break 0 = không biết 1 = có tín hiệu Break
D <sub>3</sub>	Giao thức 0 = không có lỗi 1 = có lỗi
D <sub>2</sub>	Tính chẵn lẻ 0 = không có lỗi 1 = có lỗi
D <sub>1</sub>	Số liệu 0 = không có tràn 1 = có tràn
D <sub>0</sub>	Số liệu đã sẵn sàng 0 = không có số liệu sẵn sàng 1 = số liệu sẵn sàng

## PHẦN B

### THIẾT KẾ PHẦN CỨNG

#### I. Sơ đồ khối:



#### II. Sơ đồ nguyên lý : (Xem sơ đồ)

#### III. Nguyên lý hoạt động:

Bình thường dòng điện qua đường dây thu và phát của máy Teletype là 20mA (mức điện áp là mức 0 tức  $-12^v$ ). Khi máy A muốn phát, DTR được bật lên 1 ( $+12^v$ ) → ngõ ra 7432 ở mức logic 1 → qua IC 74125 để nâng dòng kích  $T_xOK$  sáng và  $Q_1$  dẫn bão hòa → cực C của  $Q_1$  xem như nối mass ( $V_{CEQ1} \approx 0,2^v$ ) → Diode của  $U_7$  dẫn → BJT trong  $U_7$  này dẫn bão hòa → cực C của BJT này được nối mass → ngõ ra của  $U_3$  ở mức logic 1 ( $+12^v$ ) →  $Q_2$  dẫn → một dòng điện đi từ  $+60^v$  qua  $C_{CE}$  của  $Q_2$  → led của  $U_9$  → Led của  $U_{11}$  → mass của nguồn  $60^v$ . Lúc này  $Q_3$  tắt → Led của  $U_8$  tắt, Led của  $U_{10}$  tắt → hai BJT trong  $U_8$  và  $U_{10}$  không dẫn → ngõ ra của 7486 ở mức logic 1 ( $+5^v$ ) → ngõ vào máy tính CTS và RI ở mức logic 0 ( $-12^v$ ) và lúc này đường thu DSR của máy B ở mức logic 1 ( $+12^v$ ).

Khi máy B nhận được tín hiệu DSR này, nó sẽ lập tức gửi trả tín hiệu trả lời cũng bằng cách Set đường phát DTR lên mức logic 1 → đường thu DSR của máy A sẽ nhận được mức logic 1. Lúc này coi như hai máy đã được bắt tay và sẵn sàng truyền dữ liệu.

Khi máy A phát cho máy B thì một chuỗi xung sẽ được đưa qua ngõ ra  $T_xD$  của máy A để đến ngõ vào  $R_xD$  của máy B và lúc này mức logic của  $R_xD$  của máy B sẽ nhịp theo mức nhịp của máy A và được gửi vào máy tính. Ngược lại, khi máy B phát cho máy A thì mức logic ở  $R_xD$  của máy A cũng sẽ nhịp theo mức nhịp của máy B.

Giả sử khi đường truyền hở mạch, thì các Led của  $U_8$ ,  $U_9$ ,  $U_{10}$ ,  $U_{11}$  không dẫn → các BJT của  $U_8$ ,  $U_9$ ,  $U_{10}$ ,  $U_{11}$  tắt → ngõ ra của 7486 ở mức 0, lúc này Led  $T_xOPEN$  và  $R_xOPEN$  sáng → ngõ vào máy tính CTS và RI được tác động ở mức 1 ( $+12^v$ ) → máy tính báo đường truyền hở mạch.

#### IV. Tính toán các linh kiện trong mạch:

Vì các ngõ vào\_ra của cổng COM máy tính ở mức điện áp  $\pm 12^V$  nên ta phải chuyển đổi mức điện áp  $\pm 12^V \rightarrow (0 \div 5)^V$  và ngược lại từ  $(0 \div 5)^V \rightarrow \pm 12^V$  để đáp ứng các ngõ vào\_ra của IC số và các linh kiện khác. Để đảm bảo khả năng chuyển đổi điện áp được chính xác, đơn giản ta chọn IC chuyển đổi điện áp là IC 1488 (chuyển đổi từ  $(0 \div 5)^V \rightarrow \pm 12^V$ ) và IC 1489 (chuyển đổi từ  $\pm 12^V \rightarrow (0 \div 5)^V$ ). Vì 1488 và 1489 là hai IC được tích hợp, chuyên dùng và khả năng chuyển đổi điện áp chính xác hơn các loại Op\_Am và các linh kiện rời khác.

Để đảm bảo an toàn cho máy tính đồng thời tránh gây sự cố đường truyền đối với máy tính ta dùng Opto cách ly điện áp 4N35. Opto 4N35 là loại Opto đơn được dùng trong mạch là vì khả năng ít bị nhiễu đối với các linh kiện (chẳng hạn như các IC số, các led ...) so với Opto đôi.

- Vì dòng thu\_phát của máy Teletype được qui định ở mức 20mA nên chọn dòng qua led là:

$$I_{led} = I_{CQ} = 20 \text{ mA}$$

$$\Rightarrow R_4 = \frac{V_{CC} - V_{CES} - V_{LED}}{I_{LED}} = \frac{5 - 0,2 - 1,2}{20 \text{ mA}} = 180 \Omega$$

$$\rightarrow \text{Chọn } R_4 = 220 \Omega$$

$$\text{Để } Q_1 \text{ dẫn bão hòa chọn } \beta = (1/3 \div 1/4) \beta_{max}$$

$$\text{Thông thường chọn } \beta = 30$$

$$I_B = I_C / \beta = 20 / 30 = 0,6 \text{ mA}$$

$$\text{Transistor } Q_1 \text{ dẫn thì } V_{BE} = 0,7 \text{ v}$$

$$R_{BQ11} = \frac{V_{CC} - V_{BE}}{I_B} = \frac{5 - 0,7}{0,6 \text{ mA}} = 7 \text{ K}\Omega$$

$$\rightarrow \text{chọn } R_1 = 6,8 \text{ K}\Omega$$

Để BJT giao hoán tốt do ảnh hưởng dòng rỉ  $I_{CBO}$  ta chọn  $R_3 = (3,3 \div 10) \text{ K}\Omega$ .

$$\rightarrow \text{chọn } R_3 = 10 \text{ K}\Omega$$

Tương tự đối với các BJT  $Q_2, Q_3, Q_4$  ta chọn :

$$R_8 = R_9 = R_{19} = 10 \text{ K}\Omega$$

Vì các BJT của Opto chỉ hoạt động ở chế độ ngắt dẫn nên chọn dòng cực C là:

$$I_C = 1 \text{ mA}$$

$$R_5 = V_{CC} / I_C = 5 / 1\text{mA} = 5 \text{ K}\Omega$$

$$\rightarrow \text{Chọn } R_5 = 6,8 \text{ K}\Omega$$

Tương tự đối với các BJT của các Opto  $U_8, U_9, U_{10}, U_{11}$  ta cũng chọn dòng:

$$I_C = 1 \text{ mA} \quad \rightarrow \text{Chọn } R_{10} = R_{11} = R_{14} = R_{15} = 6,8 \text{ K}\Omega$$

Vì các  $R_2, R_{12}, R_{16}, R_{17}$  là các điện trở hạn dòng cho LED nên ta chọn bằng  $330\Omega$

→ Chọn  $R_2 = R_{12} = R_{16} = R_{17} = 330\Omega$

Vì dòng thu\_phát là 20mA và điều kiện để BJT ổn định điểm làm việc là:

$$\begin{aligned} V_{CE} &= V_{CC} / 2 = 60 / 2 = 30 \text{ v} \\ R_{13} &= \frac{V_{CC} - V_{CE} - 2V_{LED}}{I_C} = \frac{60 - 30 - 2(1,2)}{20 \text{ mA}} \\ &= 1,37 \text{ K}\Omega \end{aligned}$$

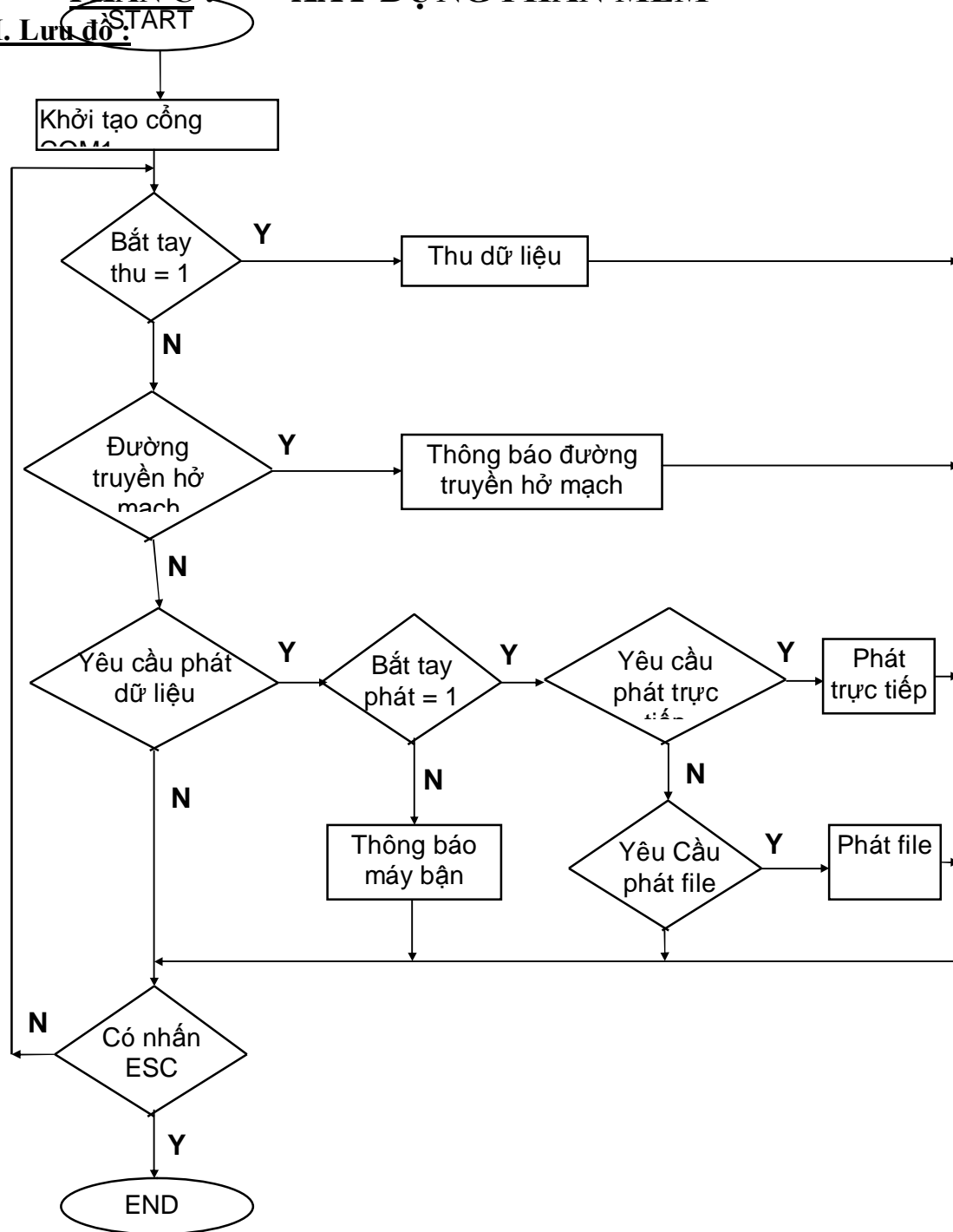
$$I_C = 20 \text{ mA}$$

→ Chọn  $R_{13} = 1,5 \text{ K}\Omega$

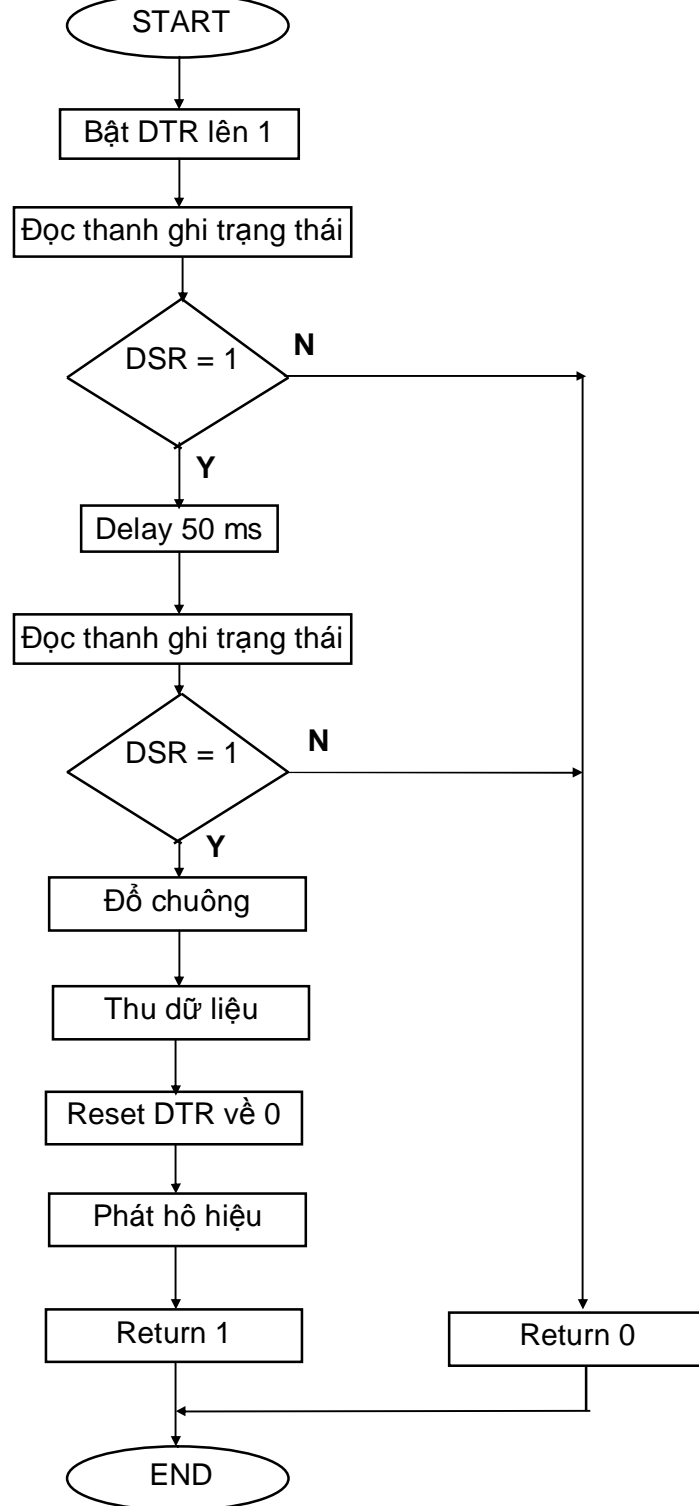
$$\Rightarrow P = R_{13} \cdot I^2 = 1,5 \cdot (0,02)^2 = 0,6 \text{ w}$$

⇒ Chọn  $P = 1 \text{ w}$ .

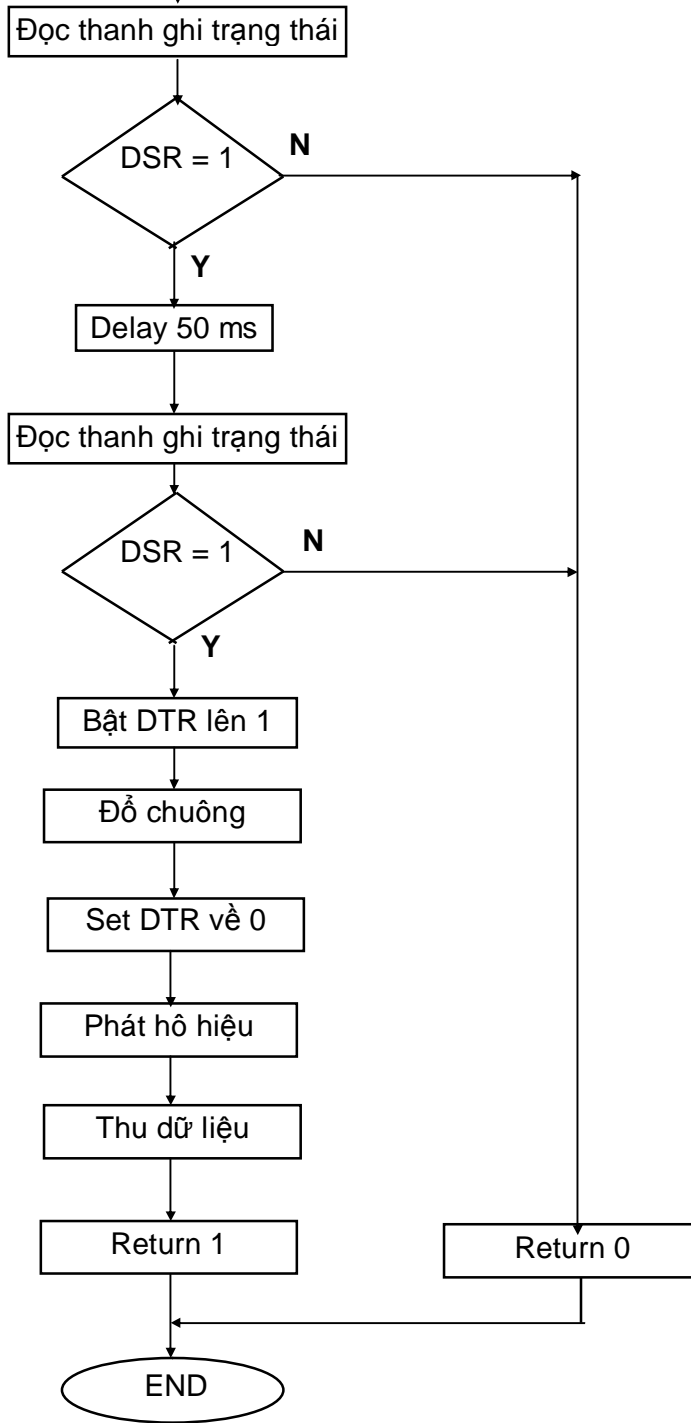
**PHẦN C : XÂY DỰNG PHẦN MỀM**  
**I. Lưu đồ :**



# LƯU ĐỒ HÀM BẮT TAY PHÁT

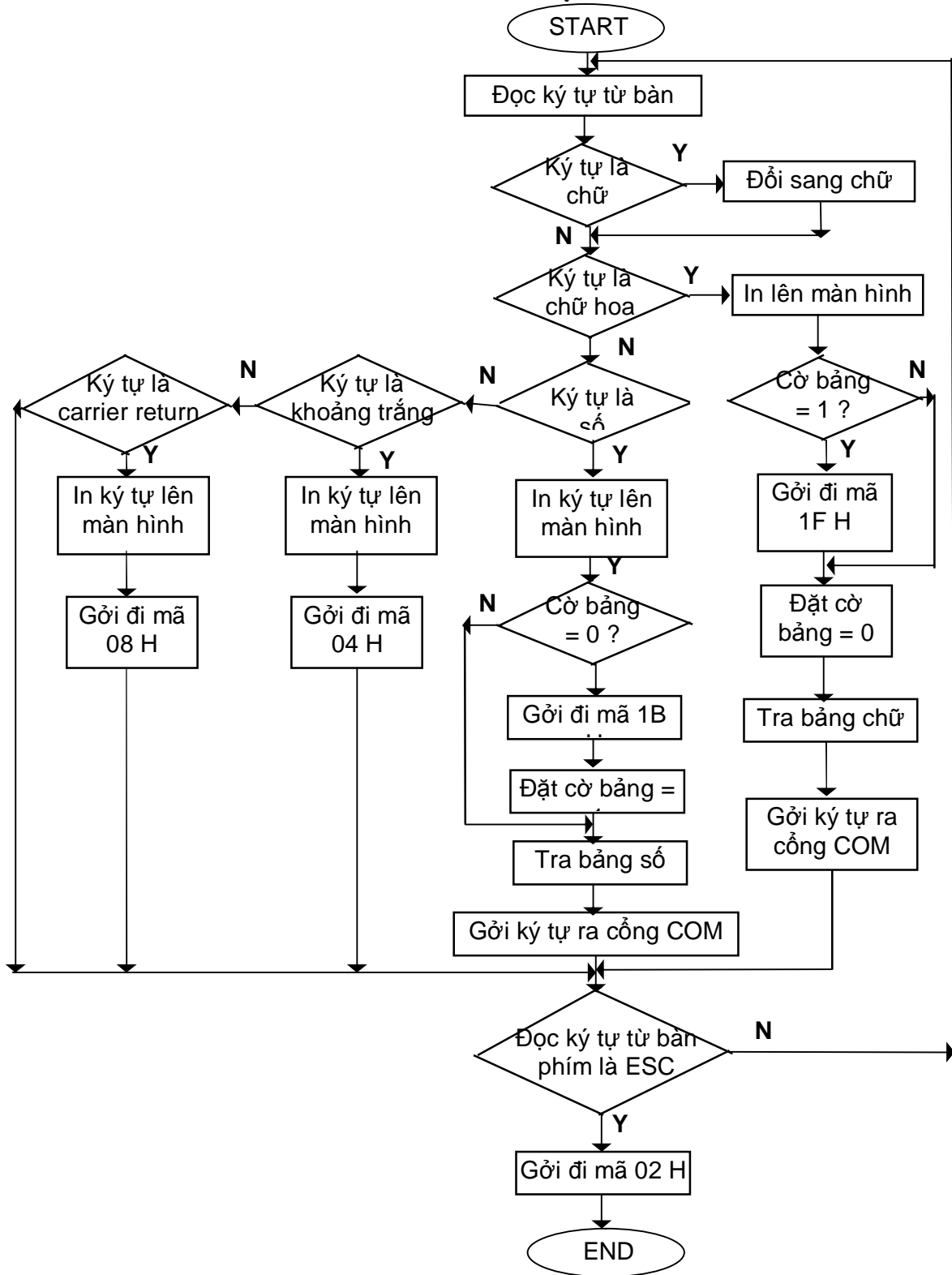


# LUỖ ĐỒ HÀM BẮT TAY THU

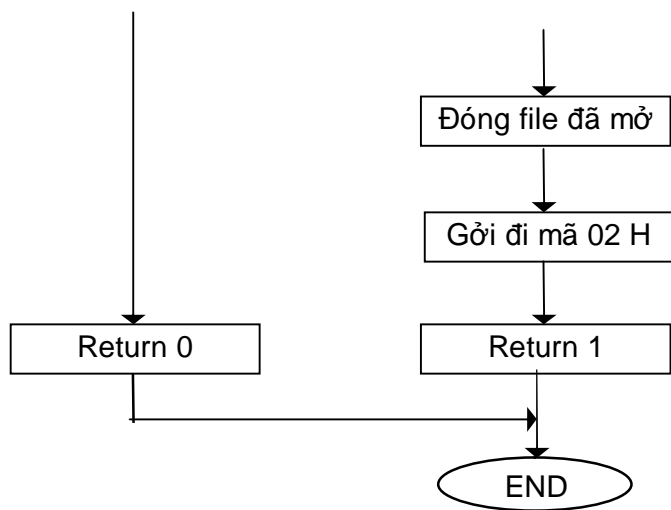


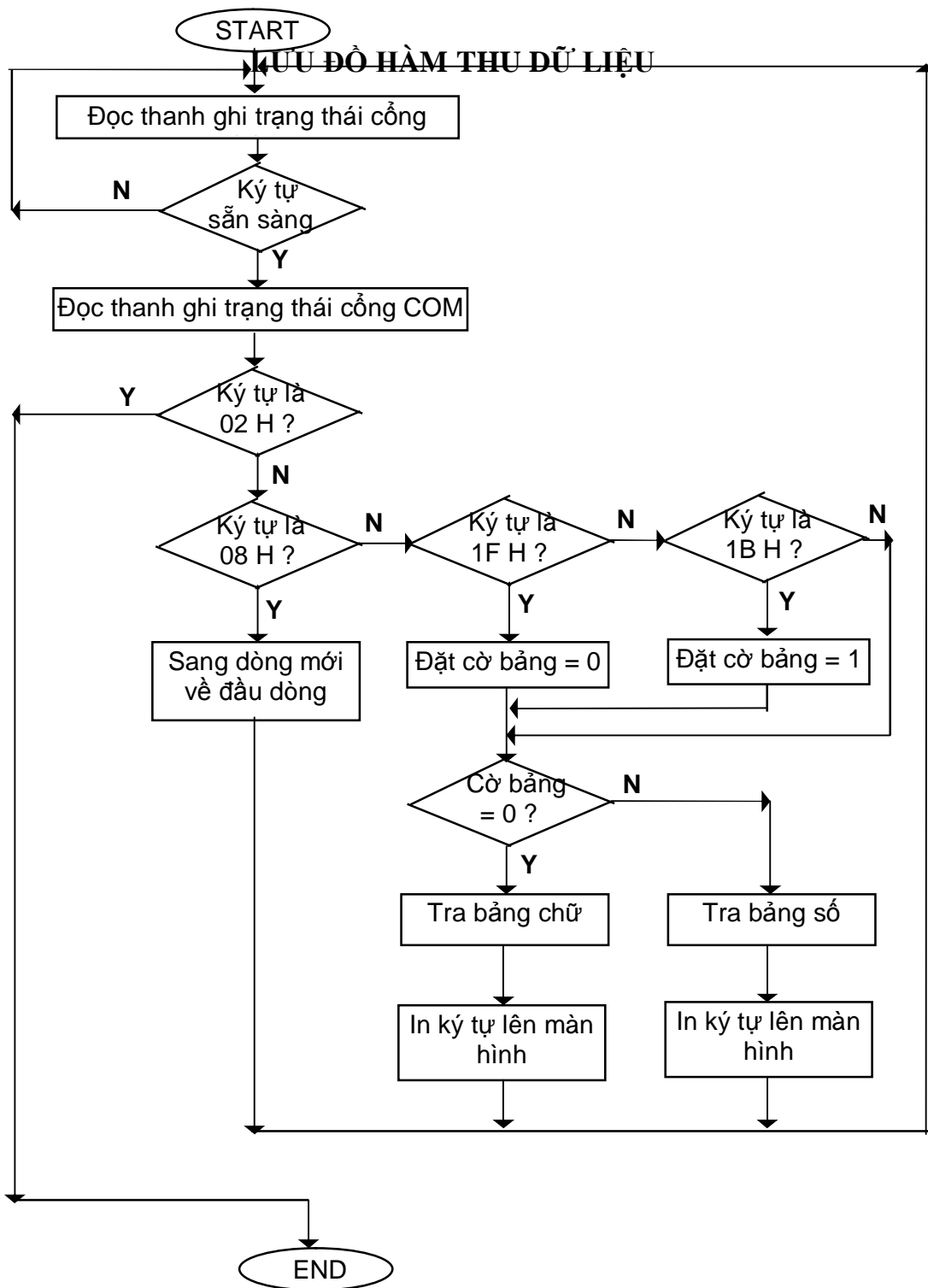


## LƯU ĐỒ HÀM PHÁT TRỰC TIẾP









## II. Chương trình:

```
#include<dos.h>
#include<graphics.h>
#include<stdlib.h>
#include<conio.h>
#include<stdio.h>
#include<ctype.h>
#include<time.h>
#include<c:\bc\bin\serial.h>
#ifndef TRUE
#define FALSE 0
#define ESC 27
#endif TRUE
void main(void)
{
    int k=0;//,mh=0,mode=0;
    char ky_tu;
    void init(void);
    int bat_tay_thu(void);
    int bat_tay_phat(void);
    int kiem_tra_RI(void);
    void phat_truc_tiep(void);
    void thu_du_lieu(void);
    void phat_file(void);
    void de_lay(int);
    void demo(void);
    // initgraph(&mh,&mode,"c:\\bc\\bgi");
    // demo();
    init();
    printf("\n\n\n\t\t\t\t\tBam phim bat ky\r");
    getch();
    do
    {
        if((bat_tay_thu()==1) { thu_du_lieu(); }
        if(kiem_tra_RI()==0)
            { clrscr();
              printf("\n\n\nDUONG THU_PHAT HO MACH\r");
              printf("\n\n\n nhan ESC : thoat.");de_lay(100);goto tiep;
            }

        clrscr();
        printf("\t\t\t\t\t nhan ESC : thoat. Nhan P: phat \n"); de_lay(100);
        tiep:
        if(kbhit())
            {
                ky_tu=getch();
                if((ky_tu=='P') || (ky_tu=='p'))
                    {
```

```

        if(bat_tay_phat()==0){clrscr();printf("\n\t\t\t\t May ban
!");de_lay(100);}
        else
        { printf("\n\r\t\t\tphat FILE nhan F ,phat truc tiep nhan T");
          k=(int)getch();
          switch(k)
          {case 116: { phat_truc_tiep(); break; } //nhan T
            case 102: { phat_file(); break; } //nhan F
          } //end switch
        } // else
        } // end if ky tu = p
    } // end kbhit()
} // end do_while
while((int)ky_tu!=ESC);
} //end main

/*=====*/
//init-- khoi tao cong noi tiep
void init(void)
{
disable();
//cho phep ngat nhan ky tu
outportb((int)&COM -> interrupt_enable,I_CHAR_IN);
//xac dinh kieu truyen va cai toc do truyen
outportb((int)&COM -> format,F_BAUD_LATCH|\
F_NO_BREAK|F_PARITY_NONE|F_STOP2|F_DATA5);
//toc do truyen byte thap
outportb((int)&COM -> baud_l,SPEED & 0xFF);
//toc do truyen byte cao
outportb((int)&COM -> baud_h,SPEED >> 8);
//tra lai truyen thu binh thuong
outportb((int)&COM -> format,F_NORMAL|\
F_NO_BREAK|F_PARITY_NONE|F_STOP2|F_DATA5);
outportb((int)&COM -> out_control,O_OUT1|\
O_OUT2|O_RTS|O_ODTR);
//doc thanh ghi dau vao de xoa co du lieu cua no
//xoa thanh ghi dem thu
(void)inportb((int)&COM -> data);
//xoa thanh ghi cho phep ngat
(void)inportb((int)&COM -> interrupt_enable);
//xoa ngat thanh ghi luu giu phan phat rong
(void)inportb((int)&COM -> interrupt_id);
//xoa thanh ghi trang thai duong day
(void)inportb((int)&COM -> status);
//xoa thanh ghi trang thai modem
(void)inportb((int)&COM -> i_status);
//tra loi ngat
outportb(0x20,0x20);//xoa ngat
enable();

```

```

} //end init

/*=====*/
// ham bat tay phan cung
int bat_tay_thu(void)
{
    void de_lay(int);
    void do_chuong(void);
    void phat_ho_hieu(void);
    void thu_du_lieu(void);
    int DSR =0;
    DSR = inportb((int)&COM -> i_status);
    if(DSR & I_DSR!=1) return(0);
    else
    {
        de_lay(50);
        DSR = inportb((int)&COM -> i_status);
    }
    if(DSR & I_DSR)
    {
        outportb((int)&COM -> out_control,O_DTR);de_lay(2);
        do_chuong();
        outportb((int)&COM -> out_control,O_0DTR);de_lay(2);
        phat_ho_hieu();
//        outportb((int)&COM -> data, 0x00);
        thu_du_lieu();
        return(1) ;
    }
    else return(0);
} //end bat_tay_thu

/*=====*/
int bat_tay_phat(void)
{ void de_lay(int);
  void do_chuong(void);
  void phat_ho_hieu(void);
  void thu_du_lieu(void);
  int dem = 5,DSR=0,ch;
  while(dem != 0)
  {
      outportb((int)&COM -> out_control, O_DTR); de_lay(100);
      DSR=inportb((int)&COM -> i_status);
      if(DSR&I_DSR)
      {
          de_lay(50);
          DSR=inportb((int)&COM -> i_status);
          if(DSR&I_DSR)
          {
              do_chuong();

```

```

        thu_du_lieu();
        outportb((int)&COM -> out_control,O_0DTR);
        phat_ho_hieu();
//        outportb((int)&COM -> data, 0x00);
        return(1);
    }
    else return(0);
}
    dem--;
} //end while
return(0);
} //end bat_tay_phat

/*=====*/
//ham phat truc tiep
void phat_truc_tiep(void)
{ void de_lay(int);
  char ch;
  int ch1=0,*p1,*p2,co_bang2=0;
  int bang_chu2[] =
      {
          0x03,0x19,0x0E,0x09,0x01,
          0x0D,0x1A,0x14,0x06,0x0B,
          0x0F,0x12,0x1C,0x0C,0x18,
          0x16,0x17,0x0A,0x05,0x10,
          0x07,0x1E,0x13,0x1D,0x15,
          0x11,0x04,0x04,0x04,0x04,
          0x04,0x04
      };
  int bang_so2[] =
      {
          0x0D,0x11,0x14,0x09,0x16,
          0x1A,0x04,0x1F,0x12,0x04,
          0x04,0x0C,0x03,0x1C,0x1D,
          0x16,0x17,0x13,0x01,0x0A,
          0x10,0x15,0x07,0x06,0x18,
          0x0E,0x1E,0x04,0x0F,0x04,
          0x19,0x04
      };
  p1 = bang_chu2; p2 = bang_so2;
  clrscr();
  do
  {
    ch = getch();
    if((int)ch== 32) { outportb((int)&COM -> data,0x04);printf("%c",ch);}
    if((int)ch == 13) { outportb((int)&COM -> data,0x08);printf("\n\r"); }
    if((int)ch>=33 && (int)ch<=64)
        {

```



```

        if(co_bang2==0){outportb((int)&COM-
>data,0x1B);de_lay(10);co_bang2=1;}
        printf("%c",ch);
        ch1 = (int)ch - 33; outportb((int)&COM -> data, p2[ch1]);de_lay(2);
    }
    if((int)ch>=97 && (int)ch<=122) ch =toupper(ch);
    if((int) ch>=65 && (int)ch<=90)
    {
        printf("%c",ch);
        if(co_bang2==1){outportb((int)&COM ->
data,0x1F);de_lay(10);co_bang2=0;}
        ch1=(int)ch - 65; outportb((int)&COM -> data, p1[ch1]);
    }
} //end do while
while(ch !=ESC);
outportb((int)&COM -> data,0x02);de_lay(10);
return;
} //end phat_truc_tiep

/*=====*/
void phat_file(void)
{
void de_lay(int);
int c,ch1,*p1,*p2,co_bang2=0;
char t1[14];
FILE *f1;
int bang_chu2[] =
    {
        0x03,0x19,0x0E,0x09,0x01,0x0D,
        0x1A,0x14,0x06,0x0B,0x0F,0x12,
        0x1C,0x0C,0x18,0x16,0x17,0x0A,
        0x05,0x10,0x07,0x1E,0x13,0x1D,
        0x15,0x11,0x04,0x04,0x04,0x04,
        0x04,0x04
    };
int bang_so2[] =
    {
        0x0D,0x11,0x14,0x09,0x16,0x1A,
        0x04,0x0F,0x12,0x04,0x04,0x0C,
        0x03,0x1C,0x1D,0x16,0x17,0x13,
        0x01,0x0A,0x10,0x15,0x07,0x06,
        0x18,0x0E,0x1E,0x04,0x0F,0x04,
        0x19,0x04
    };
p1=bang_chu2; p2=bang_so2;
clrscr();
printf("\n vao ten tep can phat : "); gets(t1);
f1=fopen(t1,"rt");

```

```

if(f1==NULL) { printf("\n KHONG MO DUOC TEP %s ",t1); de_lay(100); goto
thien; }
do
{
c=(int)fgetc(f1);
if(c==32) {printf("%c",c);outportb((int)&COM -> data,0x04);de_lay(2);}
if(c==10) {printf("\n\r");outportb((int)&COM -> data,0x08);de_lay(2);}
if(c >= 33 && c <= 64)
{
printf("%c",c);
if(co_bang2==0) {outportb((int)&COM ->
data,0x1B);de_lay(10);co_bang2=1;}
ch1=c-33;outportb((int)&COM -> data,p2[ch1]);de_lay(2);
}
if(c >= 97 && c <= 122) c = toupper(c);
if(c >= 65 && c <= 90)
{
printf("%c",c);
if(co_bang2==1){outportb((int)&COM -> data,0x1F);de_lay(10);
co_bang2=0;}
ch1 = c - 65;outportb((int)&COM -> data, p1[ch1]);de_lay(2);
}
de_lay(10);
} // end do
while(c!=EOF);
thien: fclose(f1);
outportb((int)&COM -> data,0x02);de_lay(10);
return;
} // end phat file

```

```

/*=====*/
unsigned long int tg_100(void)
{
static struct time ti;
unsigned long int t;
gettime(&ti);
t=((ti.ti_hour*60+ti.ti_min)*60+ti.ti_sec)*100+ti.ti_hund;
return t;
}

```

```

/*=====*/
//Ham thu du lieu
void thu_du_lieu(void)
{
int co_bang=0,ch=0,status=0;
char chu;
char bang_chu[] = {' ','E',10,'A',' ','S',
                  'T','U',13,'D','R','J',
                  'N','F','C','K','T','Z',

```

```

        'L','W','H','Y','P','Q',
        'O','B','G',' ','M','X',
        'V',' ','\0');
char bang_so[] = {' ','3','10','_',' ',' ',' ',
                 '8','7','13','$','4',' ',' ',
                 ' ','!',' ','(','5',' ',' ',
                 ')','2','#','6','0','1',' ',
                 '9','?','&',' ',' ','/',' ',
                 ' ',' '};

char *p1,*p2;
p1 = bang_chu;
p2 = bang_so;
clrscr();
// fprintf(stdprn,"\n\r");
do
{
    status=inportb((int)&COM -> status);
    if((status & S_RxRDY) != 0)
    {
        ch =inportb((int)&COM -> data);
        if(ch==0x02) return;
        if(ch== 0x1F){ co_bang = 0;goto tiep;}
        if(ch== 0x1B){ co_bang = 1;goto tiep;}
        if(ch== 0x08)
        {
            printf("\n\r");
//          fprintf(stdprn," \n\r ");
            goto tiep;
        }
        if(co_bang==0)
        {
            printf("%c",p1[ch]);
//          fprintf(stdprn,"%c",p1[ch]);
        }
        else
        {
            printf("%c",p2[ch]);
//          fprintf(stdprn,"%c",(int)p2[ch]);
        }
    }
} //end if((status & S_RxRDY) != 0)
tiep:
} while(ch!=0x02);//end while
return;
} //end thu du lieu

/*=====*/
void demo(void)
{setbkcolor(BLUE);
setcolor(WHITE);

```

```

        settextstyle(1,0,4);
        settextjustify(0,0);
        outtextxy(20,50,"TRUONG DAI HOC SU PHAM KY THUAT TP HCM ");
        settextstyle(1,0,2);
        settextjustify(0,0);
        outtextxy(20,130,"LUAN VAN TOT NGHIEP : MAY TINH GIAO TIEP
TELETYPE ");
        settextstyle(1,0,2);
        settextjustify(0,0);
        outtextxy(20,170,"SINH VIEN THUC HIEN : TRUONG VIET NAM -
PHAM HUNG PHONG ");
        settextstyle(1,0,2);
        settextjustify(0,0);
        outtextxy(20,210,"GIAO VIEN HUONG DAN : QUACH THANH HAI");
        settextstyle(1,0,2);
        settextjustify(0,0);
        outtextxy(20,250,"NIEN KHOA : 1995 _ 2000 ");
        getch();
        closegraph();
    }

```

```

/*=====*/

```

```

void de_lay(int n)
{ int i;
  unsigned long int t1,t2;
  unsigned long int tg_100(void);
  t1 = tg_100();
  while(1)
  {t2 = tg_100();
   if(t2-t1>n) break;
  }
  return;
}

```

```

/*=====*/

```

```

void do_chuong(void)
{
    int i ;
    for(i=0;i<3;++i){ sound(825);de_lay(100);nosound();de_lay(100);}
    return;
}

```

```

/*=====*/

```

```

void phat_ho_hieu(void)
{
    void de_lay(int);
    int c,ch1,*p1,*p2,co_bang2=0;
    FILE *f1;
    int bang_chu2[] =

```

```

        {
            0x03,0x19,0x0E,0x09,0x01,0x0D,
            0x1A,0x14,0x06,0x0B,0x0F,0x12,
            0x1C,0x0C,0x18,0x16,0x17,0x0A,
            0x05,0x10,0x07,0x1E,0x13,0x1D,
            0x15,0x11,0x04,0x04,0x04,0x04,
            0x04,0x04
        };
    int bang_so2[] =
        {
            0x0D,0x11,0x14,0x09,0x16,0x1A,
            0x04,0x0F,0x12,0x04,0x04,0x0C,
            0x03,0x1C,0x1D,0x16,0x17,0x13,
            0x01,0x0A,0x10,0x15,0x07,0x06,
            0x18,0x0E,0x1E,0x04,0x0F,0x04,
            0x19,0x04
        };
    p1=bang_chu2; p2=bang_so2;
    // clrscr();
    fl=fopen("ho_hieu.c","rt");
    if(fl==NULL) {clrscr(); printf("\n KHONG MO DUOC TEP HO_HIEU.C ");
    de_lay(100); goto thien; }
    do
    {
        c=(int)fgetc(fl);
        if(c==32) {outportb((int)&COM -> data,0x04);de_lay(2);}
        if(c==10) {outportb((int)&COM -> data,0x08);de_lay(2);}
        if(c >= 33 && c <= 64)
        {
            if(co_bang2==0) {outportb((int)&COM ->
data,0x1B);de_lay(10);co_bang2=1;}
            ch1=c-33;outportb((int)&COM -> data,p2[ch1]);de_lay(2);
        }
        if(c >= 97 && c <= 122) c = toupper(c);
        if(c >= 65 && c <= 90)
        {
            if(co_bang2==1){outportb((int)&COM -> data,0x1F);de_lay(10);
co_bang2=0;}
            ch1 = c - 65;outportb((int)&COM -> data, p1[ch1]);de_lay(2);
        }
        de_lay(10);
    } // end do
    while(c!=EOF);
    thien: fclose(fl);
        outportb((int)&COM -> data,0x02);de_lay(10);
    return;
} // end phat HO_HIEU

/*=====*/

```

```
int kiem_tra_RI(void)
{
    int RI=0;
    RI=inportb((int)&COM ->i_status);
    if(RI==0)
        return(1);
    else return(0);
}
```

# PHẦN D

## **HƯỚNG THI CÔNG**

### **I. Tổng quát:**

Mạch giao tiếp là một thiết bị độc lập để ghép nối giữa các máy vi tính với nhau nhằm truyền dữ liệu từ nơi này đến nơi khác một cách nhanh chóng. Ngoài ra, mạch giao tiếp còn chuyển đổi mức điện áp để khoảng cách truyền số liệu giữa hai máy được xa hơn. Mạch giao tiếp có thể được ứng dụng rộng rãi trong các công ty, xí nghiệp . . .

### **II. Tổ chức mạch:**

- Tất cả các IC, các linh kiện điện tử được lắp ráp trên một board mạch.
- Các ngõ vào ra giao tiếp với máy vi tính được hàn với các Connector gắn trên board.

### **III. Các bước thi công:**

- Thiết kế sơ đồ nguyên lý, vẽ mạch.
- Vẽ sơ đồ mạch in dùng phần mềm BM.
- Làm mạch in
- Chuẩn bị linh kiện.
- Lắp ráp linh kiện theo sơ đồ.
- Sau khi lắp ráp tiến hành kiểm tra phần cứng.
- Xây dựng lưu đồ phần mềm điều khiển.
- Viết chương trình bằng ngôn ngữ C.

### **IV. Báo cáo kết quả thi công:**

- Thiết kế và thi công thành công hệ thống mạch giao tiếp.
- Mạch giao tiếp được thiết kế để truyền ở khoảng cách xa, nên trong mạch thi công tồn tại các nhóm linh kiện như :
  - + Nhóm chuyển đổi từ -12V và +12V sang 0V và 5V.
  - + Nhóm chuyển đổi từ 0V và 5V sang -12V và + 12V.
  - + Nhóm nâng điện áp đường truyền lên  $\pm 60$  V để thực hiện truyền xa.

Ngoài ra mạch còn sử dụng linh kiện Opto để cách ly điện áp với máy vi tính.

#### **\* Hạn chế:**

- Do phân bố các linh kiện trên board mạch hẹp nên biến thể và mạch nguồn đặt bên ngoài mạch chính.
- Kỹ thuật lập trình chưa vững vàng nên còn nhiều thiếu sót, nhất là giao diện với người sử dụng chưa đẹp.
- Lần đầu tiên tiến hành thi công, những yêu cầu tối ưu để mạch hoàn chỉnh vẫn chưa đạt được tuy nhiên mạch vẫn đạt được những yêu cầu nhất định

## KẾT LUẬN

Qua 6 tuần tìm hiểu về cấu tạo, nguyên lý truyền dữ liệu máy Teletype và cấu trúc máy tính đã được học với sự hướng dẫn của thầy, cô đề tài luận văn tốt nghiệp đã hoàn thành .

Để làm được điều này. Chúng em cố gắng đem những kiến thức đã học vận dụng vào công việc, thực hiện nhiệm vụ luận văn được giao để có thể đạt kết quả tốt. Tuy thành quả công việc không có gì to lớn nhưng đó là quá trình dẫn dắt chúng em làm quen với việc ứng dụng kiến thức đã học vào thực tế, là thành quả của sinh viên trước khi ra trường

Với những tâm tư nguyện vọng như vậy nên chúng em hết sức cố gắng để thành luận văn đúng thời gian quy định. Do vậy trong quá trình làm việc chúng em không thể tránh khỏi sai sót. Rất mong quý thầy cô thông cảm và chỉ bảo.

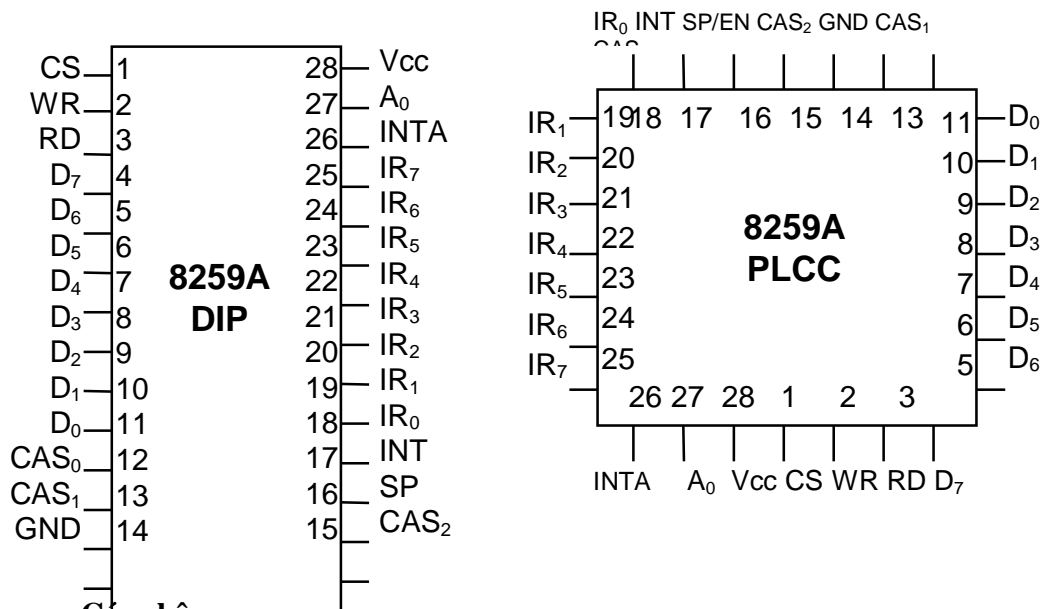


# PHỤ LỤC

## BỘ ĐIỀU KHIỂN NGẮT 8259A

Bộ điều khiển ngắt lập trình được 8259A (PIC : Programable Interrupt Controller) là một vi mạch tích hợp được thiết kế để quản lý các ngắt bên ngoài trong một hệ máy tính. Đây là những kết nối vật lý tới các thiết bị trên PC. Khi một trong số những thiết bị này tác động lên một trong những đường ngắt của PIC, PIC sẽ gửi yêu cầu ngắt cứng cùng với chỉ số của ngắt tới CPU, CPU lấy chỉ số này để truy nhập vào bảng vectơ ngắt để tìm ra thủ tục phục vụ ngắt cứng thích hợp.

8259A có 2 loại hình chữ nhật và hình vuông đều có 28 chân. Chức năng các chân của 8259A hình vuông cũng tương ứng như 8259A hình chữ nhật.



### Các chân :

IR<sub>0</sub> - IR<sub>7</sub> : (chân 18 - 25) Các lối vào yêu cầu ngắt

D<sub>0</sub> - D<sub>7</sub> : (chân 11 - 4) Các bit số liệu (hai chiều)

A<sub>0</sub> : (chân 27) Địa chỉ chọn thanh ghi lệnh

CS (chân 1) : Chọn vi mạch

WR (chân 2) : Lối vào của lệnh ghi

RD (chân 3) : Lối vào của lệnh đọc

CAS<sub>0</sub>, CAS<sub>1</sub>, CAS<sub>2</sub> (chân 12, 13, 15) : Lối vào mắc nối tầng. Một PIC chủ này có thể chọn một trong 8 PIC tớ qua 3 đường dây này.

SP (chân 16) : Trong chế độ không đếm, nếu SP = 1 thì 8259 chủ, SP = 0 thì 8259 tớ.

INTA (chân 26) : Lối vào xác định ngắt.

INT (chân 17) : Lối ra yêu cầu ngắt chương trình.

Trong quá trình hoạt động, 8259A có thể được mô tả gồm có 4 thanh ghi bên trong :

**1. Thanh ghi đáp ứng ngắt : (IRR : Interrupt Request Register)**

Chứa một bit cho từng kênh ngắt từ IRQ<sub>0</sub> đến IRQ<sub>7</sub>. Các bit riêng lẻ phản ánh những kênh nào đang yêu cầu phục vụ ngắt. Thanh ghi IRR có thể được đọc bởi CPU.

**2. Thanh ghi mặt nạ ngắt : (IMR : Interrupt Mask Register)**

Là thanh ghi 8 bit, mỗi bit mô tả một mức ngắt. Một bit được đặt bằng một sẽ ngăn chặn kênh tương ứng phát sinh một ngắt (được che).

**3. Thanh ghi bộ giải ưu tiên : (PR : Priority Resolver Register)**

Xác định xem độ ưu tiên của ngắt có đủ để ngắt một thủ tục phục vụ ngắt đang thực hiện hay không. Điều này phụ thuộc vào sơ đồ ưu tiên được lập trình hóa.

**4. Thanh ghi phục vụ trong : (ISP : In Service Register)**

Chứa một bit cho từng tốc độ ngắt. Bit này được đặt bằng 1 để chỉ dẫn kênh ngắt tương ứng đang được phục vụ. ISR có thể được đọc bởi CPU.

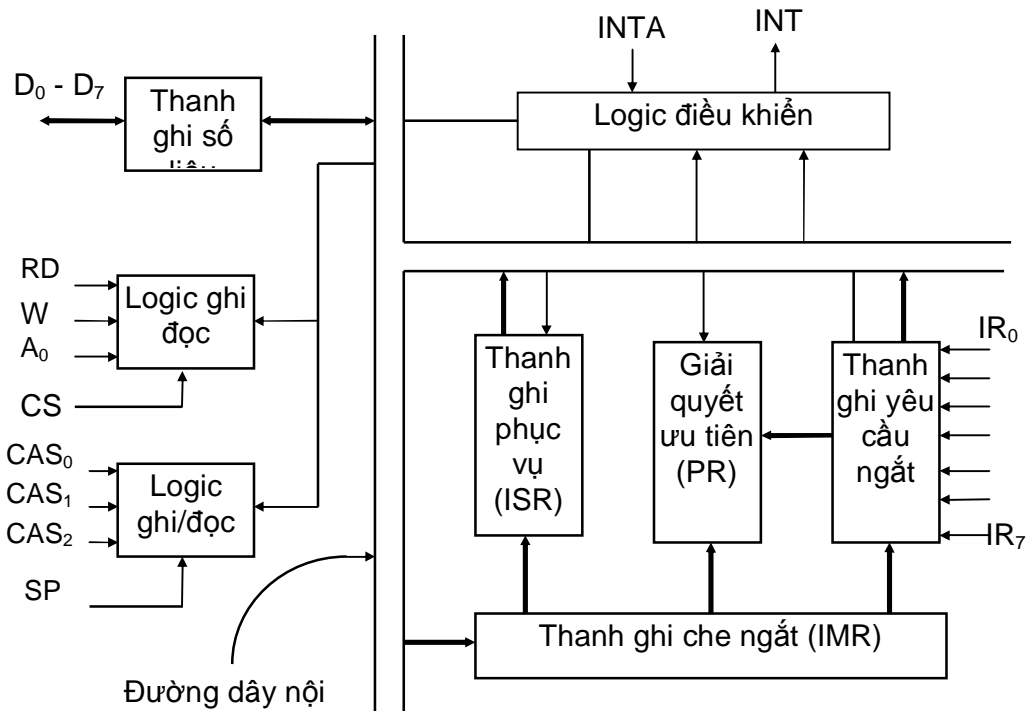
**Các bộ vi xử lý có hai dòng vật lý để báo hiệu các ngắt :**

- Dòng INTR (đáp ứng ngắt) : được điều khiển bởi bộ điều khiển ngắt Intel 8259A.
- Dòng NMI : để báo cho bộ vi xử lý biết về một sự cố nghiêm trọng sắp xảy ra như sắp mất nguồn, lỗi chặn lẻ trong bộ nhớ.
- Khối logic điều khiển : xử lý ngắt, đưa yêu cầu (INT) và nhận xác nhận ngắt (INTA).
- Bộ đếm đường dây số liệu : để đếm ghi vào các thanh ghi và đếm đọc các số liệu từ các thanh ghi.
- Logic điều khiển đọc ghi : tạo các tín hiệu ghi và đọc các thanh ghi đếm.
- Bộ đếm nối tầng/so sánh : để chọn các vi mạch 8259A thứ trong 1 vi mạch 8259A chủ.

**Bảng các giá trị đọc của các mức ưu tiên :**

			D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
Yêu cầu ưu tiên		RS T	1	1	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	1	1	1
Thấp nhất   Cao nhất	0	7	1	1	1	1	1	1	1	1
	1	6	1	1	1	1	0	1	1	1
	2	5	1	1	1	0	1	1	1	1
	3	4	1	1	1	0	0	1	1	1
	4	3	1	1	0	1	1	1	1	1
	5	2	1	1	0	1	0	1	1	1
	6	1	1	1	0	0	1	1	1	1
	7	0	1	1	0	0	0	1	1	1

**Sơ đồ khối của 8259A :**



Để cho phép một ngắt, bit mong muốn trong IMR sẽ được xóa về 0 và ngược lại nếu không cho phép một ngắt, bit tương ứng sẽ bật lên 1.

**Sơ đồ bit của thanh ghi IMR đặt tại cổng I/O 21h như sau :**

Số liệu bit	Đường truyền	Thiết bị bên ngoài
0	IRQ <sub>0</sub>	Ngắt thời gian (Time Keeper Interrup)
1	IRQ <sub>1</sub>	Ngắt bàn phím (Keyboard Interrup)
2	IRQ <sub>2</sub>	Bộ phối ghép màu đồ thị
3	IRQ <sub>3</sub>	Cổng tuần tự số 2 (COM2 hoặc COM4)
4	IRQ <sub>4</sub>	Cổng tuần tự số 1 (COM1 hoặc COM3)
5	IRQ <sub>5</sub>	Ngắt cứng (máy XT), cổng song song (máy AT)
6	IRQ <sub>6</sub>	Ngắt đĩa mềm
7	IRQ <sub>7</sub>	Ngắt máy in (cổng song song số 1)

**\* Có hai khía cạnh ta phải hiểu về PIC :**

- Bất cứ khi nào một ngắt xảy ra, PIC được chặn để nó không sinh ra một ngắt khác. Trạng thái này có thể xóa bằng cách ghi lên thanh ghi ICR của PIC. Thanh ghi này đặt tại cổng I/O của 20h.

Để Reset lại PIC và cho phép ngắt trở lại, thủ tục phục vụ ngắt ISR sẽ ghi giá trị 20h là lệnh kết thúc ngắt, chỉ thị PIC cho phép ngắt trở lại.

- Để cho phép một ngắt, ta xóa bit tương ứng trong thanh ghi mặt nạ ngắt (IMR) về 0. Ngược lại, ta bật bit tương ứng trong thanh ghi IMR lên 1 để không cho phép một ngắt.

- Ngắt cứng còn gọi là ngắt ngoài vì do nguyên nhân bên ngoài. VXL có các lối vào dành cho ngắt ngoài. Khi có tín hiệu vào các lối này, VXL đang thực hiện lệnh của chương trình sẽ bị dừng.

**Ảnh xạ ngắt 8259A lên các thủ tục phục vụ ngắt cứng :**

Đường truyền	Địa chỉ	Thủ tục phục vụ ngắt cứng	Thiết bị bên ngoài
IRQ <sub>0</sub>	20H	08h	Đồng hồ kênh 0
IRQ <sub>1</sub>	24H	09h	Bàn phím
IRQ <sub>2</sub>	28H	OAh	Bộ phối ghép màu đồ thị
IRQ <sub>3</sub>	2CH	OBh	Cổng tuần tự số 2 (COM2 - COM4)
IRQ <sub>4</sub>	30H	OCh	Cổng tuần tự số 1 (COM1 - COM3)
IRQ <sub>5</sub>	34H	ODh	Đĩa cứng (XT) - LPT2 (AT)
IRQ <sub>6</sub>	38H	OEH	Đĩa mềm
IRQ <sub>7</sub>	3CH	OFh	Cổng song song số 1 (LPT1)

## **TÀI LIỆU THAM KHẢO**

- Truyền dữ liệu .Tống Văn On - NXB Trường ĐHBK TP.Hồ Chí Minh
- Kỹ thuật ghép nối máy vi tính . Nguyễn Mạnh Giang - NXB Giáo dục 1998
- Đo lường và điều khiển bằng máy tính - Ngô Diên Tập - NXB KHKT HÀ NỘI 1996
- AT Technical Reference
- XT Technical Reference
- Giáo trình lý thuyết và bài tập ngôn ngữ C. Nguyễn Đình Tê - Hoàng Đức Hải - NXB Giáo dục 1996
- Cấu trúc máy vi tính -Trần Quang Vinh - NXB Giáo dục 1998
- Và một số tài liệu khác