







## LỜI CẢM TẠ

Trong thời gian thực hiện đề tài sinh viên đã học hỏi được nhiều điều bổ ích từ thầy hướng dẫn, từ thầy cô và các bạn.

Sinh viên thực hiện bày tỏ lòng biết ơn đến thầy LÊ THANH ĐẠO, trên cương vị là người hướng dẫn đề tài đã tận tình hướng dẫn và tạo điều kiện cho chúng em hoàn thành tốt đề tài

Sinh viên thực hiện cũng xin chân thành cảm ơn các thầy cô trong khoa Điện Tử cùng các bạn đã đóng góp ý kiến và kinh nghiệm quý báu trong quá trình thực hiện đề tài này.

Tp. Hồ Chí Minh, tháng 02 năm 2007  
Sinh viên thực hiện

## LỜI NÓI ĐẦU

Ngày nay, với sự phát triển của nền kinh tế nước ta đã có nhiều thay đổi to lớn, cuộc sống của người dân cũng ngày càng được nâng cao. Trên đà phát triển đó, vấn đề tự động hóa trong quá trình sản xuất, đáp ứng các hoạt động sinh hoạt hằng ngày của con người một cách tiện ích, nhanh chóng đã trở thành nhu cầu cần thiết.

Nắm được nhu cầu thực tiễn của xã hội nhóm thực hiện tiến hành nghiên cứu và thiết kế một hệ thống BÀI GIỮ XE TỰ ĐỘNG với mong muốn là giải quyết những nhu cầu trên, và lấy đó làm đề tài tốt nghiệp cho mình.

Những kiến thức, năng lực đạt được trong quá trình học tập ở trường sẽ được đánh giá qua đợt bảo vệ đồ án cuối khóa. Vì vậy chúng em cố gắng tận dụng tất cả những kiến thức đã học ở trường cùng với sự tìm tòi nghiên cứu, để có thể hoàn thành tốt đồ án này. Sản phẩm này là kết quả đạt được của những kiến thức đã được tích lũy trong quá trình học tập và nghiên cứu. Đây cũng là thành công đầu tiên của chúng em trước khi ra trường.

Mặc dù chúng em rất cố gắng để hoàn thành tập đồ án này đúng thời hạn, nên không tránh khỏi những thiếu sót mong quý thầy cô thông cảm. Chúng em mong được đón nhận những ý kiến đóng góp của quý thầy cô và các bạn sinh viên.

Xin chân thành cảm ơn.

Nhóm sinh viên thực hiện.

# MỤC LỤC

Dẫn nhập

## **PHẦN A : GIỚI THIỆU CÁC LINH KIỆN CHÍNH CÓ LIÊN QUAN TRONG ĐỀ TÀI**

### **Chương I : Khảo Sát Vi Điều Khiển 8951**

- I. Giới thiệu cấu trúc phần cứng họ MSC – 8951**
- II. Khảo sát sơ đồ chân 8951 và chức năng từng chân**
  1. Sơ đồ chân 8951
  2. Chức năng các chân
- III. Cấu trúc bên trong vi điều khiển**
  1. Tổ chức bộ nhớ
  2. Các thanh ghi có chức năng đặc biệt
  3. Bộ nhớ ngoài ( external memory )
    - 3.1. Truy xuất bộ nhớ mã ngoài
    - 3.2. Truy xuất bộ nhớ dữ liệu ngoài
    - 3.3. Sự giải mã địa chỉ
    - 3.4. Sự đè lên nhau của các vùng nhớ dữ liệu ngoài
    - 3.5. Hoạt động Reset
- IV. Hoạt động Timer của 8951**
  1. Giới thiệu
  2. Các thanh ghi điều khiển Timer
    - 2.1. Thanh ghi điều khiển chế độ Timer TMOD
    - 2.2. Thanh ghi điều khiển chế độ Timer TCON
    - 2.3. Các nguồn xung nhịp cho Timer
    - 2.4. Quá trình bắt đầu đếm,kết thúc đếm và điều khiển Timer
    - 2.5. Sự khởi động và truy xuất các thanh ghi Timer
  3. Các chế độ Timer và cờ tràn ( Timer Modes And Overflow )
    - 3.1. MODE 0 ( Mode Timer 13 bit )
    - 3.2. MODE 1 ( Mode Timer 16 bit )
    - 3.3. MODE 2 ( Mode tự động nạp 8 bit )
    - 3.4. MODE 3 ( Mode Timer tách ra )
- V. Hoạt động Port nối tiếp**
  1. Giới thiệu
  2. Các thanh ghi và chế độ hoạt động của Port nối tiếp
    - 2.1. Thanh ghi điều khiển Port nối tiếp

- 2.2. Thanh ghi dịch đơn 8 bit ( MODE 0 )
- 2.3. UART 8 bit với tốc độ baud thay đổi được ( MODE 1 )
- 2.4. UART 9 bit với tốc độ baud cố định ( MODE 2 )
- 2.5. UART 9 bit với tốc độ baud thay đổi được (MODE 3 )
- 2.6. Quá trình khởi động và truy xuất các thanh ghi của Port nối tiếp
- 2.7. Tốc độ baud của Port nối tiếp
- 3. Tổ chức ngắt trong 8951
  - 3.1. Cho phép và không cho phép ngắt
  - 3.2. Vectơ ngắt
  - 3.3. Ngắt Port nối tiếp

## **VI. Tóm tắt tập lệnh của 8951**

Các chế độ định vị địa chỉ ( Addressing Mode )

- 1. Định vị địa chỉ dùng thanh ghi
- 2. Định vị địa chỉ trực tiếp
- 3. Định vị địa chỉ gián tiếp
- 4. Định vị địa chỉ tức thời
- 5. Định vị địa chỉ tương đối
- 6. Định vị địa chỉ tuyệt đối
- 7. Định vị địa chỉ dài
- 8. Định vị địa chỉ chỉ số

## **Chương II : Khảo Sát Các IC Phát PT2262,IC Thu PT2272,74HC573,DAC0808, TDA2003,EN29LV040,...Và Một Số Mạch Ứng Dụng**

### **I. Giới thiệu IC phát PT2262**

- 1. Sơ đồ chân
- 2. Hoạt động phát RF
  - 2.1. Các mã bit
  - 2.2. Các mã Word
  - 2.3. Các mã khung
  - 2.4. Bộ dao động
- 3. Các mạch ứng dụng của IC phát PT2262
  - 3.1. Mạch phát 4 Data trên băng tần UHF
  - 3.2. Mạch phát địa chỉ ( không Data ) trên băng tần UHF
  - 3.3. Mạch phát địa chỉ ( không Data ) không chế độ Stand - By trên băng tần UHF
  - 3.4. Mạch phát 6 Data hồng ngoại, điều chỉnh điện trở dao động để lấy sóng mang 38 KHz ở chân DOUT

### **II. Giới thiệu IC thu PT2272**

- 1. Sơ đồ khối
- 2. Sơ đồ chân
- 3. Ý nghĩa các chân

4. Mô tả chức năng
  - 4.1. Hoạt động của mạch thu RF
  - 4.2. Bộ dao động
5. Tính hiệu lực của việc truyền tín hiệu
6. Dữ liệu ra được chốt lại hay tức thời
  - 6.1. Loại chốt ( PT2272 – LX )
  - 6.2. Loại tức thời ( PT2272 – MX )
7. Trình tự hoạt động
8. Các mạch ứng dụng
  - 8.1. Ứng dụng cho PT2272 loại không có Data
  - 8.2. Ứng dụng cho PT2272 có 4 Data
  - 8.3. Mạch hồng ngoại cho PT2272 loại 4 Data

### **III. Giới thiệu IC 74HC573**

1. Sơ đồ khối
2. Sơ đồ chân
3. Ý nghĩa các chân
4. Bảng trạng thái
5. Nguyên tắc hoạt động

### **IV. Giới thiệu DAC0808**

1. Sơ đồ chân và kết nối của DAC0808
2. Sơ đồ kết nối bên trong DAC0808

### **V. Giới thiệu TDA2003**

1. Sơ đồ chân, hình dạng và kết nối của TDA2003
2. Sơ đồ kết nối bên trong của TDA2003

### **VI. Giới thiệu EN29LV040**

1. Sơ đồ chân
2. Sơ đồ khối cấu trúc bên trong EN29LV040

## **PHẦN B : THIẾT KẾ PHẦN CỨNG**

### **A. TỔNG QUAN ĐỀ TÀI**

#### **I. Quy trình ra vào bãi**

#### **II. Cơ cấu kỹ thuật**

### **B. SƠ ĐỒ KHỐI TỔNG QUAN ĐỀ TÀI**

#### **I. Chức năng từng khối**

#### **II. Các sơ đồ nguyên lý cụ thể**

1. Mạch phát
2. Mạch thu
3. Mạch điều khiển motor
4. Sơ đồ card giao tiếp máy tính
5. Mạch khuếch đại âm thanh



- C. SƠ ĐỒ NGUYÊN LÝ KHÔI GIAO TIẾP MÁY TÍNH
- D. SƠ ĐỒ NGUYÊN LÝ KHÔI XỬ LÝ TRÊN MÔ HÌNH
- E. SƠ ĐỒ NGUYÊN LÝ KHÔI NHẬP MẬT MÃ

**PHẦN C : THI CÔNG****PHẦN D : PHẦN MỀM**

A.CHƯƠNG TRÌNH VI ĐIỀU KHIỂN PHẦN ÂM THANH

**I. Lưu đồ****II. Phần chương trình VĐK**

B.CHƯƠNG TRÌNH VI ĐIỀU KHIỂN KHÔI XỬ LÝ TRÊN MÔ HÌNH

**I. Lưu đồ****II. Phần chương trình VĐK**

C.CHƯƠNG TRÌNH VI ĐIỀU KHIỂN KHÔI GIAO TIẾP MÁY TÍNH

**I. Lưu đồ****II. Phần chương trình VĐK**

D.THIẾT KẾ GIAO DIỆN VÀ PHẦN MỀM VB

**I. Chương trình Form 1****II. Chương trình Form 2****PHẦN E : KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN CỦA ĐỀ TÀI****I. Tóm tắt nội dung đề tài****II. Kết quả đạt được**

1. Tính khoa học
2. Khả năng triển khai ứng dụng vào thực tiễn
3. Hiệu quả kinh tế

**III. Kết luận****IV.Những hạn chế và hướng phát triển đề tài****TÀI LIỆU THAM KHẢO**

# DẪN NHẬP

## I. ĐẶT VẤN ĐỀ:

Hiện nay đất nước ta đang hội nhập vào sự phát triển chung của nền kinh tế thế giới và khu vực Châu Á bằng nền sản xuất đa dạng và đầy tiềm năng. Nền kinh tế đã và đang phát triển này không chỉ đòi hỏi một số lượng lao động khổng lồ mà còn yêu cầu về trình độ, chất lượng tay nghề, kỹ thuật lao động và thiết bị sản xuất. Trên đà phát triển đó, vấn đề tự động hóa trong quá trình sản xuất, trong các hoạt động sinh hoạt hằng ngày của con người đã trở thành nhu cầu cần thiết. Thoạt đầu vấn đề tự động hóa được thực hiện riêng lẻ từ cơ khí hóa cho đến các mạch điện tử. Ngày nay, với sự xuất hiện của các Chip vi xử lý và máy tính cùng với việc sử dụng rộng rãi của nó đã đẩy vấn đề tự động hóa lên một bước cao hơn và thời lượng nhanh hơn...

Cùng với sự phát triển của đất nước thì chất lượng cuộc sống của người dân cũng được nâng cao. Một số thành phố lớn ở nước ta, việc đi lại bằng xe bốn bánh là rất lớn do đó nhu cầu phải có bãi đỗ và giữ xe là rất cần thiết. Tuy nhiên, ở các khu đô thị lớn thì khoảng không gian trống là rất ít do đó cần xây dựng những bãi đỗ xe và giữ xe nhiều tầng và mang tính chuyên nghiệp.

Từ những nhu cầu thực tế và rất thực tiễn ở nước ta, chúng em quyết định chọn đề tài: “ BÃI GIỮ XE BỐN BÁNH TỰ ĐỘNG ” để nghiên cứu và phát triển. Trong đề tài này có sử dụng ứng dụng Máy Tính và kỹ thuật truyền số liệu không dây vào điều khiển các thiết bị trong hệ thống. Ngoài ra, máy tính còn có phần giao diện rất tiện lợi cho người sử dụng.

## II. GIỚI HẠN ĐỀ TÀI:

Với thời gian sáu tuần thực hiện đề tài, cũng như trình độ chuyên môn có hạn, chúng em đã cố gắng hết sức để hoàn thành đề tài này và đã giải quyết được những vấn đề sau:

- Hoàn thành tập đồ án.
- Thiết kế hoàn chỉnh mô hình bãi đỗ xe tự động.
- Viết chương trình (phần mềm) để đáp ứng các yêu cầu trong cơ cấu kỹ thuật của đề tài.
- Cho hoạt động nhiều lần mô hình chạy tốt cả phần cứng cũng như phần mềm.

Ngoài ra còn một số vấn đề chưa giải quyết được:

- Một số tính năng mang tính chất mô hình.
- Giao diện Visual Basic nghiên cứu một phần nhỏ.

### **III. MỤC ĐÍCH NGHIÊN CỨU:**

Mục đích trước hết khi thực hiện đề tài này là để hoàn tất chương trình môn học để đủ điều kiện ra trường.

- Cụ thể khi nghiên cứu thực hiện đề tài này là chúng em muốn phát huy những thành quả ứng dụng của vi điều khiển nhằm tạo ra những sản phẩm, những thiết bị tiên tiến hơn, và đạt hiệu quả sản xuất cao hơn.
- Mặt khác tập đề án này cũng có thể làm tài liệu tham khảo cho những sinh viên khoá sau. Giúp họ hiểu rõ hơn về những ứng dụng của vi điều khiển.
- Ngoài ra quá trình nghiên cứu thực hiện đề tài là trường để tham gia vào các hoạt động sản xuất của xã hội. một cơ hội để chúng em tự kiểm tra lại những kiến thức đã được học ở trường, đồng thời phát huy tính sáng tạo, khả năng giải quyết một vấn đề theo yêu cầu đặt ra. Và đây cũng là dịp để chúng em tự khẳng định mình trước khi ra trường tham gia vào các hoạt động sản xuất của xã hội.

# PHẦN A :

GIỚI THIỆU CÁC LINH  
KIỆN CHÍNH CÓ LIÊN  
QUAN TRONG ĐỀ TÀI

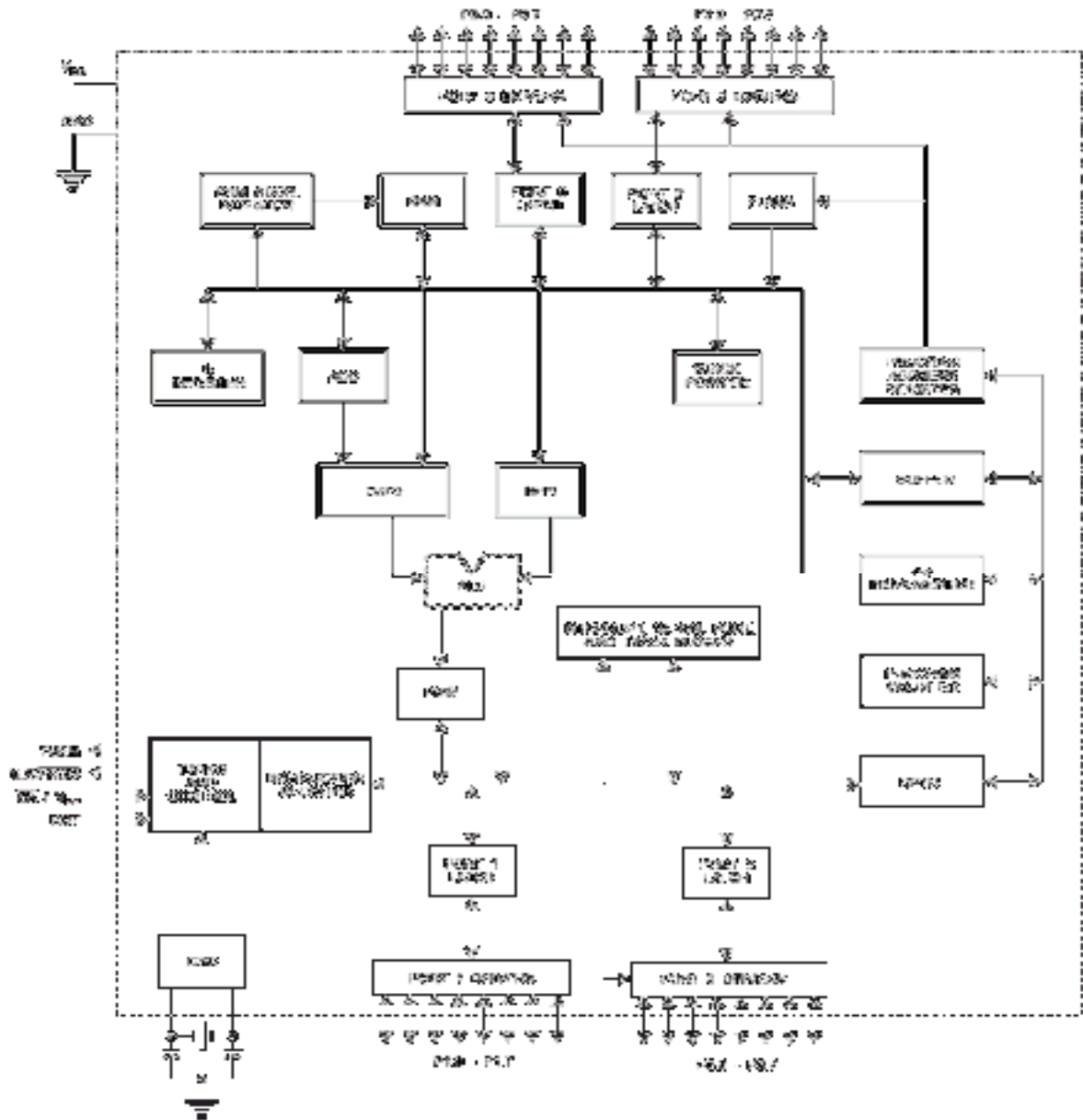
**CHƯƠNG I:****KHẢO SÁT VI ĐIỀU KHIỂN 8951****I. GIỚI THIỆU CẤU TRÚC PHẦN CỨNG HO MSC-51 (8951):**

- Đặc điểm và chức năng hoạt động của các IC họ MSC-51 hoàn toàn tương tự như nhau. Ở đây giới thiệu IC8951 là một họ IC vi điều khiển do hãng Intel của Mỹ sản xuất. Chúng có các đặc điểm chung như sau:

Các đặc điểm của 8951 được tóm tắt như sau :

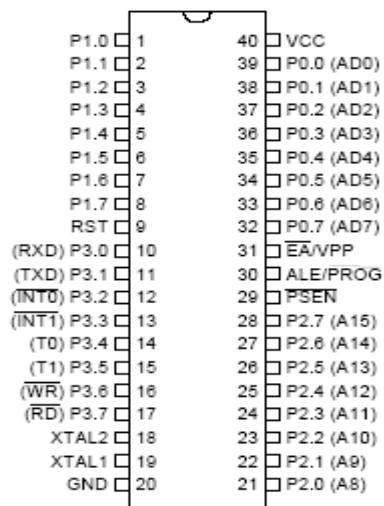
- 8 KB EPROM bên trong.
- 128 Byte RAM nội.
- 4 Port xuất /nhập (Input/Output) 8 bit.
- Giao tiếp nối tiếp.
- 64 KB vùng nhớ mã ngoài
- 64 KB vùng nhớ dữ liệu ngoài.
- Xử lý Boolean (hoạt động trên bit đơn).
- 210 vị trí nhớ có thể định vị bit.
- 4  $\mu$ s cho hoạt động nhân hoặc chia.

**Sơ đồ khối của 8951:**



## II. KHẢO SÁT SƠ ĐỒ CHÂN 8951, CHỨC NĂNG TỪNG CHÂN:

### 1. Sơ đồ chân 8951:



*Sơ đồ chân IC 8951*

### 2. Chức năng các chân của 8951:

- 8951 có tất cả 40 chân có chức năng như các đường xuất nhập. Trong đó có 24 chân có tác dụng kép (có nghĩa 1 chân có 2 chức năng), mỗi đường có thể hoạt động như đường xuất nhập hoặc như đường điều khiển hoặc là thành phần của các bus dữ liệu và bus địa chỉ.

#### **a. Các Port:**

- **Port 0 :**

- Port 0 là port có 2 chức năng ở các chân 32 – 39 của 8951. Trong các thiết kế cỡ nhỏ không dùng bộ nhớ mở rộng nó có chức năng như các đường IO. Đối với các thiết kế cỡ lớn có bộ nhớ mở rộng, nó được kết hợp giữa bus địa chỉ và bus dữ liệu.

- **Port 1:**

- Port 1 là port IO trên các chân 1-8. Các chân được ký hiệu P1.0, P1.1, P1.2, ... có thể dùng cho giao tiếp với các thiết bị ngoài nếu cần. Port 1 không có chức năng khác, vì vậy chúng chỉ được dùng cho giao tiếp với các thiết bị bên ngoài.

- **Port 2:**

- Port 2 là 1 port có tác dụng kép trên các chân 21 - 28 được dùng như các đường xuất nhập hoặc là byte cao của bus địa chỉ đối với các thiết bị dùng bộ nhớ mở rộng.

- **Port 3:**

- Port 3 là port có tác dụng kép trên các chân 10 - 17. Các chân của port này có nhiều chức năng, các công dụng chuyên đổi có liên hệ với các đặc tính đặc biệt của 8951 như ở bảng sau:

Bit	Tên	Chức năng chuyển đổi
P3.0	RxT	Ngõ vào dữ liệu nối tiếp.
P3.1	TxD	Ngõ xuất dữ liệu nối tiếp.
P3.2	INT0\	Ngõ vào ngắt cứng thứ 0.
P3.3	INT1\	Ngõ vào ngắt cứng thứ 1.
P3.4	T0	Ngõ vào của TIMER/COUNTER thứ 0.
P3.5	T1	Ngõ vào của TIMER/COUNTER thứ 1.
P3.6	WR\	Tín hiệu ghi dữ liệu lên bộ nhớ ngoài.
P3.7	RD\	Tín hiệu đọc dữ liệu từ bộ nhớ ngoài.

### b. Các ngõ tín hiệu điều khiển:

- **Ngõ tín hiệu PSEN (Program store enable):**

- PSEN là tín hiệu ngõ ra ở chân 29 có tác dụng cho phép đọc bộ nhớ chương trình mở rộng thường được nối đến chân 0E\ (output enable) của Eprom cho phép đọc các byte mã lệnh.
- PSEN ở mức thấp trong thời gian Microcontroller 8951 lấy lệnh. Các mã lệnh của chương trình được đọc từ Eprom qua bus dữ liệu và được chốt vào thanh ghi lệnh bên trong 8951 để giải mã lệnh. Khi 8951 thi hành chương trình trong ROM nội PSEN sẽ ở mức logic 1.

- **Ngõ tín hiệu điều khiển ALE (Address Latch Enable) :**

- Khi 8951 truy xuất bộ nhớ bên ngoài, port 0 có chức năng là bus địa chỉ và bus dữ liệu do đó phải tách các đường dữ liệu và địa chỉ. Tín hiệu ra ALE ở chân thứ 30 dùng làm tín hiệu điều khiển để giải đa hợp các đường địa chỉ và dữ liệu khi kết nối chúng với IC chốt.
- Tín hiệu ra ở chân ALE là một xung trong khoảng thời gian port 0 đóng vai trò là địa chỉ thấp nên chốt địa chỉ hoàn toàn tự động. Các xung tín hiệu ALE có tốc độ bằng 1/6 lần tần số dao động trên chip và có thể được dùng làm tín hiệu clock cho các phần khác của hệ thống. Chân ALE được dùng làm ngõ vào xung lập trình cho Eprom trong 8951.

- **Ngõ tín hiệu EA\ (External Access):**

- Tín hiệu vào EA\ ở chân 31 thường được mắc lên mức 1 hoặc mức 0. Nếu ở mức 1, 8951 thi hành chương trình từ ROM nội trong khoảng địa chỉ thấp 8 Kbyte. Nếu ở mức 0, 8951 sẽ thi hành chương trình từ bộ nhớ mở rộng. Chân EA\ được lấy làm chân cấp nguồn 21V khi lập trình cho Eprom trong 8951.

- **Ngõ tín hiệu RST (Reset) :**

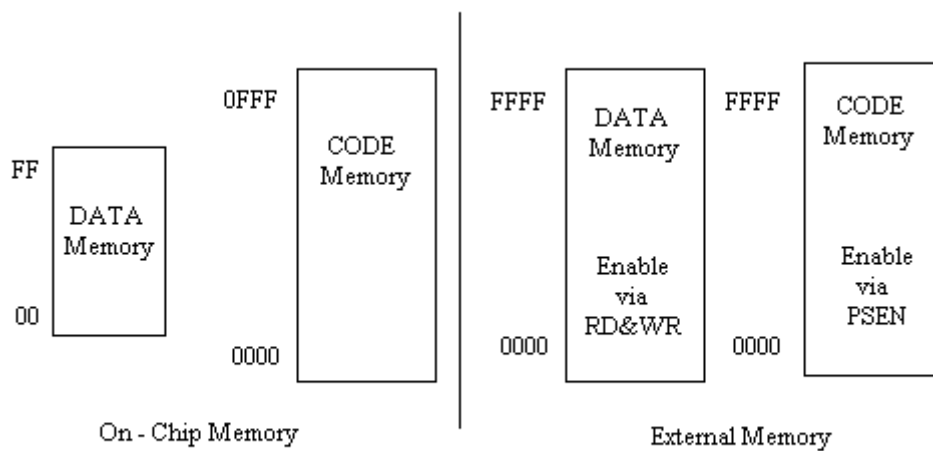
- Ngõ vào RST ở chân 9 là ngõ vào Reset của 8951. Khi ngõ vào tín hiệu này đưa lên cao ít nhất là 2 chu kỳ máy, các thanh ghi bên trong được nạp những giá trị thích hợp để khởi động hệ thống. Khi cấp điện mạch tự động Reset.



- **Các ngõ vào bộ dao động X1, X2:**
  - Bộ dao động được tích hợp bên trong 8951, khi sử dụng 8951 người thiết kế chỉ cần kết nối thêm thạch anh và các tụ như hình vẽ trong sơ đồ. Tần số thạch anh thường sử dụng cho 8951 là 12Mhz.
  - Chân 40 (Vcc) được nối lên nguồn 5V.

### **III. CẤU TRÚC BÊN TRONG VI ĐIỀU KHIỂN :**

#### **1. Tổ chức bộ nhớ:**



**Bảng tóm tắt các vùng nhớ 8951.**

Cấu trúc bộ nhớ RAM bên trong vi điều khiển :

Địa chỉ byte	Địa chỉ bit	Địa chỉ byte	Địa chỉ bit		
7F	RAM đa dụng	FF			
		F0	F7   F6   F5   F4   F3   F2   F1   F0	B	
		E0	E7   E6   E5   E4   E3   E2   E1   E0	ACC	
		D0	D7   D6   D5   D4   D3   D2   D1   D0	PSW	
		B8	-   -   -   BC   BB   BA   B9   B8	IP	
		B0	B7   B6   B5   B4   B3   B2   B1   B0	P.3	
		A8	AF       AC   AB   AA   A9   A8	IE	
		A0	A7   A6   A5   A4   A3   A2   A1   A0	P2	
30			99	không được địa chỉ hoá bit	SBUF
2F		7F   7E   7D   7C   7B   7A   79   78	98	9F   9E   9D   9C   9B   9A   99   98	SCON
2E		77   76   75   74   73   72   71   70	90	97   96   95   94   93   92   91   90	P1
2D		6F   6E   6D   6C   6B   6A   69   68	8D	không được địa chỉ hoá bit	TH1
2C		67   66   65   64   63   62   61   60	8C	không được địa chỉ hoá bit	TH0
2B		5F   5E   5D   5C   5B   5A   59   58	8B	không được địa chỉ hoá bit	TL1
2A		57   56   55   54   53   52   51   50	8A	không được địa chỉ hoá bit	TL0
29		4F   4E   4D   4C   4B   4A   49   48	89	không được địa chỉ hoá bit	TMOD
28		47   46   45   44   43   42   41   40	88	8F   8E   8D   8C   8B   8A   89   88	TCON
27		3F   3E   3D   3C   3B   3A   39   38	87	không được địa chỉ hoá bit	PCON
26		37   36   35   34   33   32   31   30	83	không được địa chỉ hoá bit	DPH
25		2F   2E   2D   2C   2B   2A   29   28	82	không được địa chỉ hoá bit	DPL
24		27   26   25   24   23   22   21   20	81	không được địa chỉ hoá bit	SP
23		1F   1E   1D   1C   1B   1A   19   18	88	87   86   85   84   83   82   81   80	P0
22		17   16   15   14   13   12   11   10			
21		0F   0E   0D   0C   0B   0A   09   08			
20		07   06   05   04   03   02   01   00			
1F		Bank 3			
18					
17		Bank 2			
10					
0F		Bank 1			
08					
07	Bank thanh ghi 0				
00	(mặc định cho R0 -R7)				

**RAM CÁC THANH GHI CHỨC NĂNG ĐẶC BIỆT**

- Bộ nhớ trong 8951 bao gồm EPROM và RAM. RAM trong 8951 bao gồm nhiều thành phần: phần lưu trữ đa dụng, phần lưu trữ địa chỉ hóa từng bit, các bank thanh ghi và các thanh ghi chức năng đặc biệt.

- 8951 có bộ nhớ theo cấu trúc Harvard: có những vùng bộ nhớ riêng biệt cho chương trình và dữ liệu. Chương trình và dữ liệu có thể chứa bên trong 8951 nhưng 8951 vẫn có thể kết nối với 64K byte bộ nhớ chương trình và 64K byte dữ liệu.

**Hai đặc tính cần chú ý là :**

- ◆ Các thanh ghi và các port xuất nhập đã được định vị (xác định) trong bộ nhớ và có thể truy xuất trực tiếp giống như các địa chỉ bộ nhớ khác.
- ◆ Ngăn xếp bên trong Ram nội nhỏ hơn so với Ram ngoại như trong các bộ Microcontroller khác.

RAM bên trong 8951 được phân chia như sau:

- + Các bank thanh ghi có địa chỉ từ 00H đến 1FH.
- + RAM địa chỉ hóa từng bit có địa chỉ từ 20H đến 2FH.
- + RAM đa dụng từ 30H đến 7FH.
- + Các thanh ghi chức năng đặc biệt từ 80H đến FFH.

- **RAM đa dụng:**

- Mặc dù trên hình vẽ cho thấy 80 byte đa dụng chiếm các địa chỉ từ 30H đến 7FH, 32 byte dưới từ 00H đến 1FH cũng có thể dùng với mục đích tương tự (mặc dù các địa chỉ này đã có mục đích khác).

- Mọi địa chỉ trong vùng RAM đa dụng đều có thể truy xuất tự do dùng kiểu địa chỉ trực tiếp hoặc gián tiếp.

- **RAM có thể truy xuất từng bit:**

- 8951 chứa 210 bit được địa chỉ hóa, trong đó có 128 bit có chứa các byte có chứa các địa chỉ từ 20F đến 2FH và các bit còn lại chứa trong nhóm thanh ghi có chức năng đặc biệt.

- Ý tưởng truy xuất từng bit bằng phần mềm là các đặc tính mạnh của microcontroller xử lý chung. Các bit có thể được đặt, xóa, AND, OR, . . . , với 1 lệnh đơn. Đa số các microcontroller xử lý đòi hỏi một chuỗi lệnh đọc – sửa - ghi để đạt được mục đích tương tự. Ngoài ra các port cũng có thể truy xuất được từng bit.

- 128 bit truy xuất từng bit này cũng có thể truy xuất như các byte hoặc như các bit phụ thuộc vào lệnh được dùng.

- **Các bank thanh ghi:**

- 32 byte thấp của bộ nhớ nội được dành cho các bank thanh ghi. Bộ lệnh 8951 hỗ trợ 8 thanh ghi có tên là R0 đến R7 và theo mặc định sau khi reset hệ thống, các thanh ghi này có các địa chỉ từ 00H đến 07H.

- Các lệnh dùng các thanh ghi R0 đến R7 sẽ ngắn hơn và nhanh hơn so với các lệnh có chức năng tương ứng dùng kiểu địa chỉ trực tiếp. Các dữ liệu được dùng thường xuyên nên dùng một trong các thanh ghi này.

- Do có 4 bank thanh ghi nên tại một thời điểm chỉ có một bank thanh ghi được truy xuất bởi các thanh ghi R0 đến R7 để chuyển đổi việc truy xuất các bank thanh ghi ta phải thay đổi các bit chọn bank trong thanh ghi trạng thái.

## 2. Các thanh ghi có chức năng đặc biệt:

- Các thanh ghi nội của 8951 được truy xuất ngầm định bởi bộ lệnh.
- Các thanh ghi trong 8951 được định dạng như một phần của RAM trên chip vì vậy mỗi thanh ghi sẽ có một địa chỉ (ngoại trừ thanh ghi bộ đếm chương trình và thanh ghi lệnh vì các thanh ghi này hiếm khi bị tác động trực tiếp). Cũng như R0 đến R7, 8951 có 21 thanh ghi có chức năng đặc biệt (SFR: Special Function Register) ở vùng trên của RAM nội từ địa chỉ 80H đến FFH.

Chú ý: tất cả 128 địa chỉ từ 80H đến FFH không được định nghĩa, chỉ có 21 thanh ghi có chức năng đặc biệt được định nghĩa sẵn các địa chỉ.

- Ngoại trừ thanh ghi A có thể được truy xuất ngầm như đã nói, đa số các thanh ghi có chức năng đặc biệt SFR có thể địa chỉ hóa từng bit hoặc byte.

- Thanh ghi trạng thái chương trình (PSW: Program Status Word):

Từ trạng thái chương trình ở địa chỉ D0H được tóm tắt như sau:

BIT	SYMBOL	ADDRESS	DESCRIPTION
PSW.7	CY	D7H	Carry Flag
PSW.6	AC	D6H	Auxiliary Carry Flag
PSW.5	F0	D5H	Flag 0
PSW.4	RS1	D4H	Register Bank Select 1
PSW.3	RS0	D3H	Register Bank Select 0
			00=Bank 0; address 00H÷07H
			01=Bank 1; address 08H÷0FH
			10=Bank 2; address 10H÷17H
			11=Bank 3; address 18H÷1FH
PSW.2	OV	D2H	Overflow Flag
PSW.1	-	D1H	Reserved
PSW.0	P	DOH	Even Parity Flag

### Chức năng từng bit trạng thái chương trình

*Cờ Carry CY (Carry Flag):*

- Cờ nhớ có tác dụng kép. Thông thường nó được dùng cho các lệnh toán học: C=1 nếu phép toán cộng có sự tràn hoặc phép trừ có mượn và ngược lại C= 0 nếu phép toán cộng không tràn và phép trừ không có mượn.

*Cờ Carry phụ AC (Auxiliary Carry Flag):*

- Khi cộng những giá trị BCD (Binary Code Decimal), cờ nhớ phụ AC được set nếu kết quả 4 bit thấp nằm trong phạm vi điều khiển 0AH÷ 0FH. Ngược lại AC = 0.

*Cờ 0 (Flag 0):*

- Cờ 0 (F0) là 1 bit cờ đa dụng dùng cho các ứng dụng của người dùng.

*Những bit chọn bank thanh ghi truy xuất:*

- RS1 và RS0 quyết định dãy thanh ghi tích cực. Chúng được xóa sau khi reset hệ thống và được thay đổi bởi phần mềm khi cần thiết.

- Tùy theo RS1, RS0 = 00, 01, 10, 11 sẽ được chọn Bank tích cực tương ứng là Bank 0, Bank1, Bank2, Bank3.

RS1	RS0	BANK
0	0	0
0	1	1
1	0	2
1	1	3

*Cờ tràn OV (Over Flag) :*

- Cờ tràn được set sau một hoạt động cộng hoặc trừ nếu có sự tràn toán học. Khi các số có dấu được cộng hoặc trừ với nhau, phần mềm có thể kiểm tra bit này để xác định xem kết quả có nằm trong tầm xác định không. Khi các số không có dấu được cộng bit OV được bỏ qua. Các kết quả lớn hơn +127 hoặc nhỏ hơn -128 thì bit OV = 1.

*Bit Parity (P):*

- Bit tự động được set hay Clear ở mỗi chu kỳ máy để lập Parity chẵn với thanh ghi A. Sự đếm các bit 1 trong thanh ghi A cộng với bit Parity luôn luôn chẵn. Ví dụ A chứa 10101101B thì bit P set lên một để tổng số bit 1 trong A và P tạo thành số chẵn.

- Bit Parity thường được dùng trong sự kết hợp với những thủ tục của Port nối tiếp để tạo ra bit Parity trước khi phát đi hoặc kiểm tra bit Parity sau khi thu.

*Thanh ghi B:*

- Thanh ghi B ở địa chỉ F0H được dùng cùng với thanh ghi A cho các phép toán nhân chia. Lệnh MUL AB  $\leftarrow$  sẽ nhận những giá trị không dấu 8 bit trong hai thanh ghi A và B, rồi trả về kết quả 16 bit trong A (byte cao) và B(byte thấp). Lệnh DIV AB  $\leftarrow$  lấy A chia B, kết quả nguyên đặt vào A, số dư đặt vào B.

- Thanh ghi B có thể được dùng như một thanh ghi đếm trung gian đa mục đích. Nó là những bit định vị thông qua những địa chỉ từ F0H÷F7H.

□ *Con trỏ Ngăn xếp SP (Stack Pointer) :*

- Con trỏ ngăn xếp là một thanh ghi 8 bit ở địa chỉ 81H. Nó chứa địa chỉ của byte dữ liệu hiện hành trên đỉnh ngăn xếp. Các lệnh trên ngăn xếp bao gồm các lệnh cất dữ liệu vào ngăn xếp (PUSH) và lấy dữ liệu ra khỏi Ngăn xếp (POP). Lệnh cất dữ liệu vào ngăn xếp sẽ làm tăng SP trước khi ghi dữ liệu và lệnh lấy ra khỏi ngăn xếp sẽ làm giảm SP. Ngăn xếp của 8031/8051 được giữ trong RAM nội và giới hạn các địa chỉ có thể truy xuất bằng địa chỉ gián tiếp, chúng là 128 byte đầu của 8951.

- Để khởi động SP với ngăn xếp bắt đầu tại địa chỉ 60H, các lệnh sau đây được dùng:

```
MOV SP , #5F
```

- Với lệnh trên thì ngăn xếp của 8951 chỉ có 32 byte vì địa chỉ cao nhất của RAM trên chip là 7FH. Sở dĩ giá trị 5FH được nạp vào SP vì SP tăng lên 60H trước khi cất byte dữ liệu.

- Khi Reset 8951, SP sẽ mang giá trị mặc định là 07H và dữ liệu đầu tiên sẽ được cất vào ô nhớ ngăn xếp có địa chỉ 08H. Nếu phần mềm ứng dụng không khởi động SP một giá trị mới thì bank thanh ghi 1 có thể cả 2 và 3 sẽ không dùng được vì vùng RAM này đã được dùng làm ngăn xếp. Ngăn xếp được truy xuất trực tiếp bằng các lệnh PUSH và POP để lưu trữ tạm thời và lấy lại dữ liệu, hoặc truy xuất ngầm bằng lệnh gọi chương trình con ( ACALL, LCALL) và các lệnh trở về (RET, RETI) để lưu trữ giá trị của bộ đếm chương trình khi bắt đầu thực hiện chương trình con và lấy lại khi kết thúc chương trình con ...

□ *Con trỏ dữ liệu DPTR (Data Pointer):*

- Con trỏ dữ liệu (DPTR) được dùng để truy xuất bộ nhớ ngoài là một thanh ghi 16 bit ở địa chỉ 82H (DPL: byte thấp) và 83H (DPH: byte cao). Ba lệnh sau sẽ ghi 55H vào RAM ngoài ở địa chỉ 1000H:

```
MOV A , #55H
```

```
MOV DPTR, #1000H
```

```
MOV @DPTR, A
```

- Lệnh đầu tiên dùng để nạp 55H vào thanh ghi A. Lệnh thứ hai dùng để nạp địa chỉ của ô nhớ cần lưu giá trị 55H vào con trỏ dữ liệu DPTR. Lệnh thứ ba sẽ di chuyển nội dung thanh ghi A (là 55H) vào ô nhớ RAM bên ngoài có địa chỉ chứa trong DPTR (là 1000H).

□ *Các thanh ghi Port (Port Register):*

- Các Port của 8951 bao gồm Port0 ở địa chỉ 80H, Port1 ở địa chỉ 90H, Port2 ở địa chỉ A0H, và Port3 ở địa chỉ B0H. Tất cả các Port này đều có thể truy xuất từng bit nên rất thuận tiện trong khả năng giao tiếp.

□ *Các thanh ghi Timer (Timer Register):*

- 8951 có chứa hai bộ định thời/ bộ đếm 16 bit được dùng cho việc định thời được đếm sự kiện. Timer0 ở địa chỉ 8AH (TLO: byte thấp ) và 8CH (THO: byte cao). Timer1 ở địa chỉ 8BH (TL1: byte thấp) và 8DH (TH1: byte cao). Việc khởi động timer được SET bởi Timer Mode (TMOD) ở địa chỉ 89H và thanh ghi điều khiển Timer (TCON) ở địa chỉ 88H. Chỉ có TCON được địa chỉ hóa từng bit .

□ *Các thanh ghi Port nối tiếp (Serial Port Register) :*

- 8951 chứa một Port nối tiếp cho việc trao đổi thông tin với các thiết bị nối tiếp như máy tính, modem hoặc giao tiếp nối tiếp với các IC khác. Một thanh ghi đệm dữ liệu nối tiếp (SBUF) ở địa chỉ 99H sẽ giữ cả hai dữ liệu truyền và dữ liệu nhập. Khi truyền dữ liệu ghi lên SBUF, khi nhận dữ liệu thì đọc SBUF. Các mode vận khác nhau được lập trình qua thanh ghi điều khiển Port nối tiếp (SCON) được địa chỉ hóa từng bit ở địa chỉ 98H.

□ *Các thanh ghi ngắt (Interrupt Register):*

- 8951 có cấu trúc 5 nguồn ngắt, 2 mức ưu tiên. Các ngắt bị cấm sau khi bị reset hệ thống và sẽ được cho phép bằng việc ghi thanh ghi cho phép ngắt (IE) ở địa chỉ A8H. Cả hai được địa chỉ hóa từng bit.

□ *Thanh ghi điều khiển nguồn PCON (Power Control Register):*

- Thanh ghi PCON không có bit định vị. Nó ở địa chỉ 87H chứa nhiều bit điều khiển. Thanh ghi PCON được tóm tắt như sau:

- + Bit 7 (SMOD) : Bit có tốc độ Baud ở mode 1, 2, 3 ở Port nối tiếp khi set.
- + Bit 6, 5, 4 : Không có địa chỉ.
- + Bit 3 (GF1) : Bit cờ đa năng 1.
- + Bit 2 (GF0) : Bit cờ đa năng 2 .
- + Bit 1 (PD) : Set để khởi động mode Power Down và thoát để reset.
- + Bit 0 (IDL) : Set để khởi động mode Idle và thoát khi ngắt mạch hoặc reset.

Các bit điều khiển Power Down và Idle có tác dụng chính trong tất cả các IC họ MSC-51 nhưng chỉ được thi hành trong sự biên dịch của CMOS.

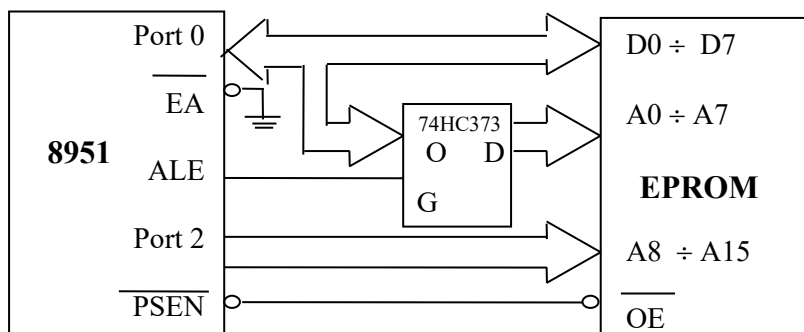
### **3. Bộ nhớ ngoài (external memory) :**

- 8951 có khả năng mở rộng bộ nhớ lên đến 64K byte bộ nhớ chương trình và 64k byte bộ nhớ dữ liệu ngoài. Do đó có thể dùng thêm RAM và ROM nếu cần.

- Khi dùng bộ nhớ ngoài, Port0 không còn chức năng I/O nữa. Nó được kết hợp giữa bus địa chỉ (A0-A7) và bus dữ liệu (D0-D7) với tín hiệu ALE để chốt byte của bus địa chỉ khi bắt đầu mỗi chu kỳ bộ nhớ. Port được cho là byte cao của bus địa chỉ.

### 3.1. Truy xuất bộ nhớ mã ngoài (Accessing External Code Memory) :

- Bộ nhớ chương trình bên ngoài là bộ nhớ ROM được cho phép của tín hiệu PSEN\.  
Sự kết nối phần cứng của bộ nhớ EPROM như sau:

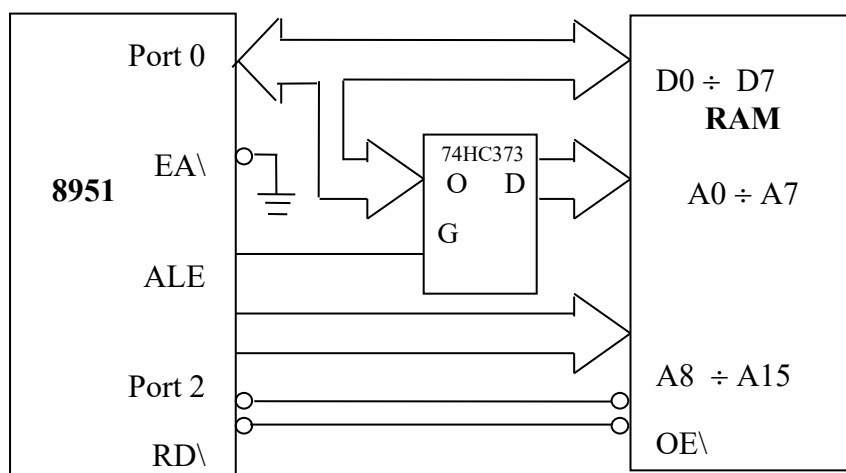


- Trong một chu kỳ máy tiêu biểu, tín hiệu ALE tích 2 lần. Lần thứ nhất cho phép 74HC373 mở cổng chốt địa chỉ byte thấp, khi ALE xuống 0 thì byte thấp và byte cao của bộ đếm chương trình đều có nhưng EPROM chưa xuất vì PSEN\ chưa tích cực, khi tín hiệu lên một trở lại thì Port 0 đã có dữ liệu là Opcode. ALE tích cực lần thứ hai được giải thích tương tự và byte 2 được đọc từ bộ nhớ chương trình. Nếu lệnh đang hiện hành là lệnh 1 byte thì CPU chỉ đọc Opcode, còn byte thứ hai bỏ đi.

### 3.2. Truy xuất bộ nhớ dữ liệu ngoài (Accessing External Data Memory):

- Bộ nhớ dữ liệu ngoài là một bộ nhớ RAM được đọc hoặc ghi khi được cho phép của tín hiệu RD\ và WR. Hai tín hiệu này nằm ở chân P3.7 (RD) và P3.6 (WR). Lệnh MOVX được dùng để truy xuất bộ nhớ dữ liệu ngoài và dùng một bộ đệm dữ liệu 16 bit (DPTR), R0 hoặc R1 như là một thanh ghi địa chỉ.

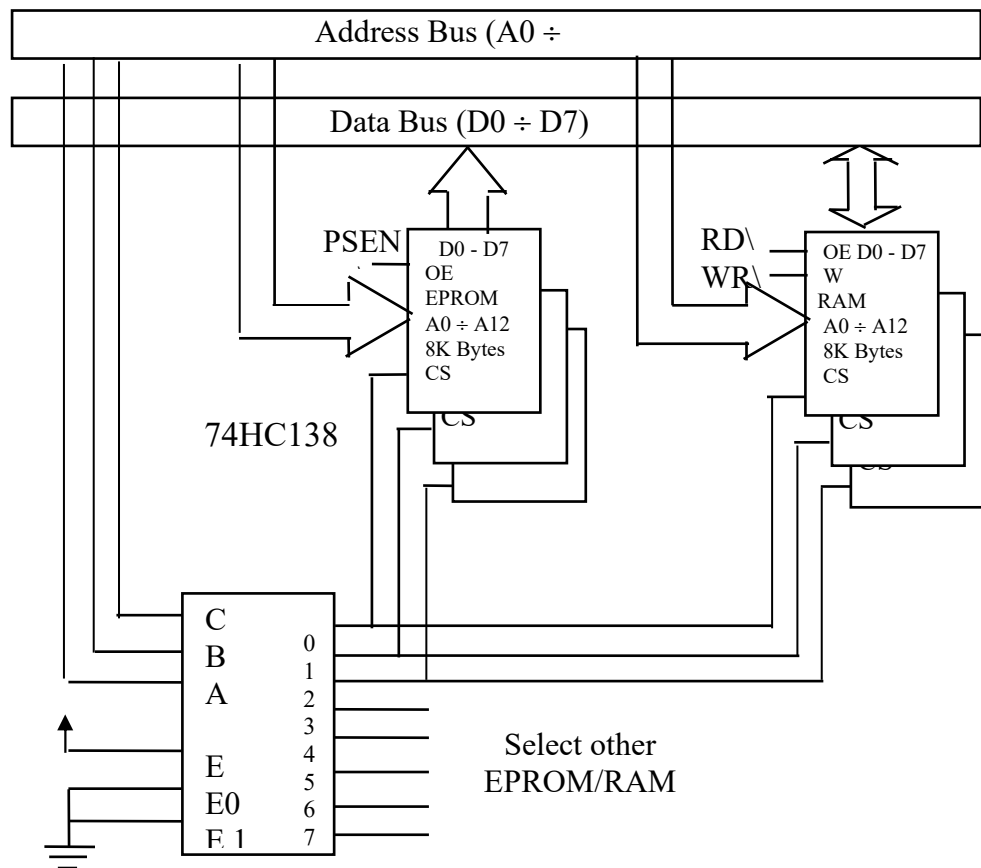
- Các RAM có thể giao tiếp với 8951 tương tự cách thức như EPROM ngoại trừ chân RD\ của 8951 nối với chân OE\ (Output Enable) của RAM và chân WR\ của 8951 nối với chân WE\ của RAM. Sự nối các bus địa chỉ và dữ liệu tương tự như cách nối của EPROM.





### 3.3. Sự giải mã địa chỉ (Address Decoding):

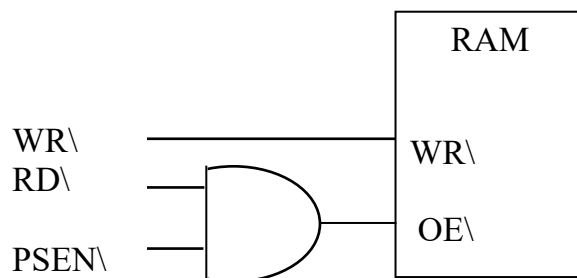
- Sự giải mã địa chỉ là một yêu cầu tất yếu để chọn EPROM, RAM, 8279, ... Sự giải mã địa chỉ đối với 8951 để chọn các vùng nhớ ngoài như các vi điều khiển. Nếu các con EPROM hoặc RAM 8K được dùng thì các bus địa chỉ phải được giải mã để chọn các IC nhớ nằm trong phạm vi giới hạn 8K: 0000H÷1FFFH, 2000H÷3FFFH,...
- Một cách cụ thể, IC giải mã 74HC138 được dùng với những ngõ ra của nó được nối với những ngõ vào chọn Chip CS (Chip Select) trên những IC nhớ EPROM, RAM, ... Hình sau đây cho phép kết nối nhiều EPROM và RAM.



Address Decoding (Giải mã địa chỉ)

### 3.4. Sự đè lên nhau của các vùng nhớ dữ liệu ngoài:

- Vì bộ nhớ chương trình là ROM, nên nảy sinh một vấn đề bất tiện khi phát triển phần mềm cho vi điều khiển. Một nhược điểm chung của 8951 là các vùng nhớ dữ liệu ngoài nằm đè lên nhau, vì tín hiệu PSEN\ được dùng để đọc bộ nhớ mã ngoài và tín hiệu RD\ được dùng để đọc bộ nhớ dữ liệu, nên một bộ nhớ RAM có thể chứa cả chương trình và dữ liệu bằng cách nối đường OE\ của RAM đến ngõ ra một cổng AND có hai ngõ vào PSEN\ và RD\. Sơ đồ mạch như hình sau cho phép cho phép bộ nhớ RAM có hai chức năng vừa là bộ nhớ chương trình vừa là bộ nhớ dữ liệu:

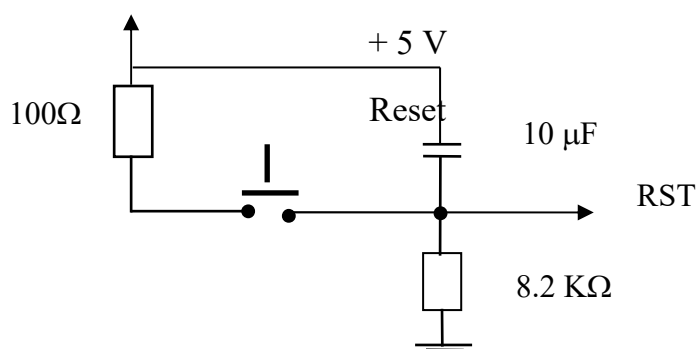


### Overlapping the External code and data space

- Vậy một chương trình có thể được tải vào RAM bằng cách xem nó như bộ nhớ dữ liệu và thi hành chương trình bằng cách xem nó như bộ nhớ chương trình.

### 3.5.Hoạt động Reset:

- 8951 có ngõ vào reset RST tác động ở mức cao trong khoảng thời gian 2 chu kỳ xung máy, sau đó xuống mức thấp để 8951 bắt đầu làm việc. RST có thể kích bằng tay bằng một phím nhấn thường hở, sơ đồ mạch reset như sau:



### Manual Reset , Reset bằng tay.

Trạng thái của tất cả các thanh ghi trong 8951 sau khi reset hệ thống được tóm tắt như sau:

Thanh ghi	Nội dung
Đếm chương trình PC	0000H
Thanh ghi tích lũy A	00H
Thanh ghi B	00H
Thanh ghi thái PSW	00H
SP	07H
DPRT	0000H
Port 0 đến port 3	FFH
IP	XXX0 0000 B
IE	0X0X 0000 B
Các thanh ghi định thời	00H
	00H

SCON SBUF	00H
PCON (MHOS)	0XXX XXXXH
PCON (CMOS)	0XXX 0000 B

- Thanh ghi quan trọng nhất là thanh ghi bộ đếm chương trình PC được reset tại địa chỉ 0000H. Khi ngõ vào RST xuống mức thấp, chương trình luôn bắt đầu tại địa chỉ 0000H của bộ nhớ chương trình. Nội dung của RAM trên chip không bị thay đổi bởi tác động của ngõ vào reset.

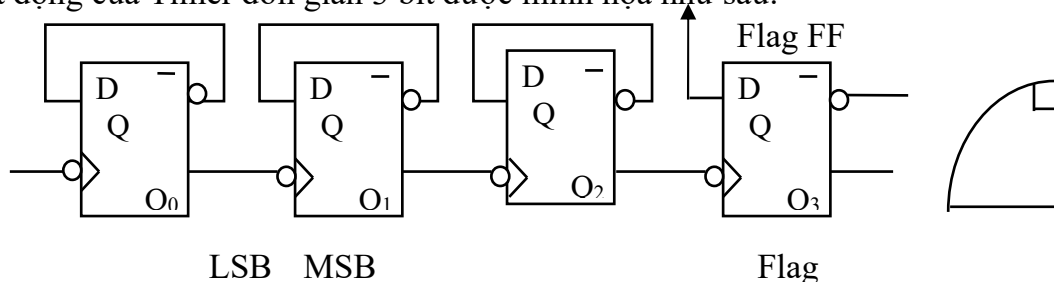
#### **IV. HOẠT ĐỘNG TIMER CỦA 8951:**

##### **1. Giới thiệu :**

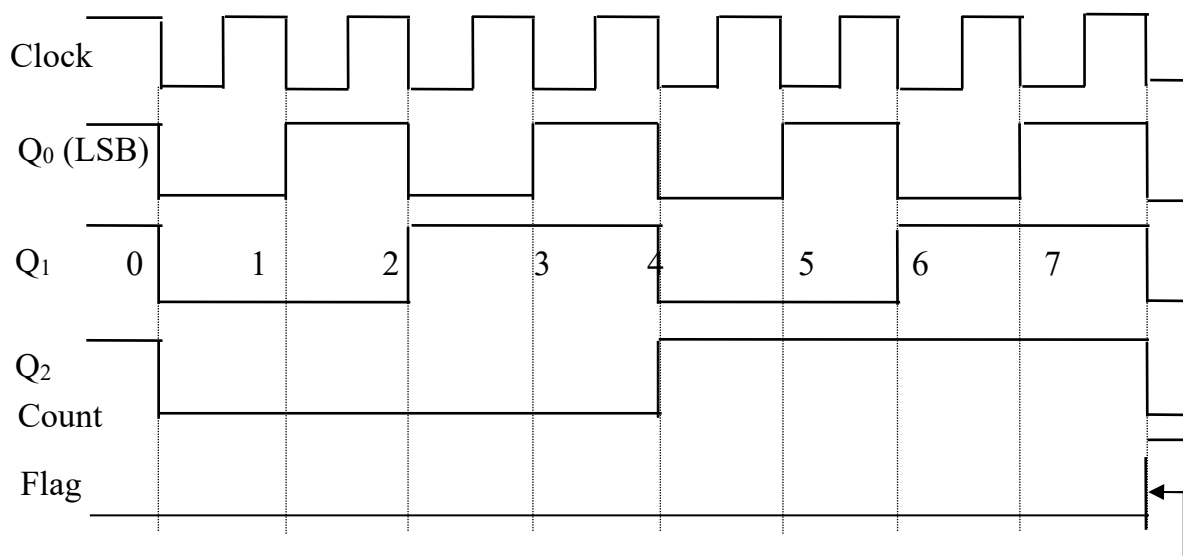
- Bộ định thời của Timer là một chuỗi các Flip Flop được chia làm 2, nó nhận tín hiệu vào là một nguồn xung clock, xung clock được đưa vào Flip Flop thứ nhất là xung clock của Flip Flop thứ hai mà nó cũng chia tần số clock này cho 2 và cứ tiếp tục.

- Vì mỗi tầng kế tiếp chia cho 2, nên Timer n tầng phải chia tần số clock ngõ vào cho  $2^n$ . Ngõ ra của tầng cuối cùng là clock của Flip Flop tràn Timer hoặc cờ mà nó kiểm tra bởi phần mềm hoặc sinh ra ngắt. Giá trị nhị phân trong các FF của bộ Timer có thể được nghĩ như đếm xung clock hoặc các sự kiện quan trọng bởi vì Timer được khởi động. Ví dụ Timer 16 bit có thể đếm đến từ FFFFH sang 0000H.

- Hoạt động của Timer đơn giản 3 bit được minh họa như sau:



**Timer Flip Flops.**



- Trong hình trên mỗi tầng là một FF loại D phủ định tác động cạnh xuống được hoạt động ở mode chia cho 2 (ngõ ra Q\ được nối vào D). FF còn là một bộ chốt đơn giản loại D được set bởi tầng cuối cùng trong Timer. Trong biểu đồ thời gian, tầng đầu đổi trạng thái ở  $\frac{1}{2}$  tần số clock, tầng thứ hai đổi trạng thái ở tần số  $\frac{1}{4}$  tần số clock . . . Số đếm được biết ở dạng thập phân và được kiểm tra lại dễ dàng bởi việc kiểm tra các tầng của 3 FF. Ví dụ số đếm “4” xuất hiện khi  $Q2=1, Q1=0, Q0=0$  ( $4_{10}=100_2$ ).
- Các Timer được ứng dụng thực tế cho các hoạt động định hướng. 8951 có 2 bộ Timer 16 bit, mỗi Timer có 4 mode hoạt động. Các Timer dùng để đếm giờ, đếm các sự kiện cần thiết và sự sinh ra tốc độ của tốc độ Baud bởi sự gắn liền Port nối tiếp.
- Mỗi sự định thời là một Timer 16 bit, do đó tầng cuối cùng là tầng thứ 16 sẽ chia tần số clock vào cho  $2^{16} = 65.536$ .
- Trong các ứng dụng định thời, 1 Timer được lập trình để tràn ở một khoảng thời gian đều đặn và được set cờ tràn Timer. Cờ được dùng để đồng bộ chương trình để thực hiện một hoạt động như việc đưa tới 1 tầng các ngõ vào hoặc gửi dữ liệu đếm ngõ ra. Các ứng dụng khác có sử dụng việc ghi giờ đều đều của Timer để đo thời gian đã trôi qua hai trạng thái (ví dụ đo độ rộng xung). Việc đếm một sự kiện được dùng để xác định số lần xuất hiện của sự kiện đó, tức thời gian trôi qua giữa các sự kiện.
- Các Timer của 8951 được truy xuất bởi việc dùng 6 thanh ghi chức năng đặc biệt như sau :

Timer SFR	Purpose	Address	Bit-Addressable
TCON	Control	88H	YES
TMOD	Mode	89H	NO
TL0	Timer 0 low-byte	8AH	NO
TL1	Timer 1 low-byte	8BH	NO
TH0	Timer 0 high-byte	8CH	NO
TH1	Timer 1 high-byte	8DH	NO

## 2. Các thanh ghi điều khiển timer :

### 2.1. Thanh ghi điều khiển chế độ timer TMOD (timer mode register) :

- Thanh ghi mode gồm hai nhóm 4 bit là: 4 bit thấp đặt mode hoạt động cho Timer 0 và 4 bit cao đặt mode hoạt động cho Timer 1. 8 bit của thanh ghi TMOD được tóm tắt như sau:

Bit	Name	Timer	Description
7	GATE	1	Khi GATE = 1, Timer chỉ làm việc khi INT1=1
6	C/T	1	Bit cho đếm sự kiện hay ghi giờ
			C/T = 1 : Đếm sự kiện
			C/T = 0 : Ghi giờ đều đặn
5	M1	1	Bit chọn mode của Timer 1
4	M0	1	Bit chọn mode của Timer 1
3	GATE	0	Bit công của Timer 0
2	C/T	0	Bit chọn Counter/Timer của Timer 0
1	M1	0	Bit chọn mode của Timer 0
0	M0	0	Bit chọn mode của Timer 0

Hai bit M0 và M1 của TMOD để chọn mode cho Timer 0 hoặc Timer 1.

M1	M0	MODE	DESCRIPTION
0	0	0	Mode Timer 13 bit (mode 8048)
0	1	1	Mode Timer 16 bit
1	0	2	Mode tự động nạp 8 bit
1	1	3	Mode Timer tách ra : Timer 0 : TL0 là Timer 8 bit được điều khiển bởi các bit của Timer 0. TH0 tương tự nhưng được điều khiển bởi các bit của mode Timer 1. Timer 1 : Được ngừng lại.

- TMOD không có bit định vị, nó thường được LOAD một lần bởi phần mềm ở đầu chương trình để khởi động mode Timer. Sau đó sự định giờ có thể dừng lại, được khởi động lại như thế bởi sự truy xuất các thanh ghi chức năng đặc biệt của Timer khác.

## 2.2. Thanh ghi điều khiển timer TCON (timer control register):

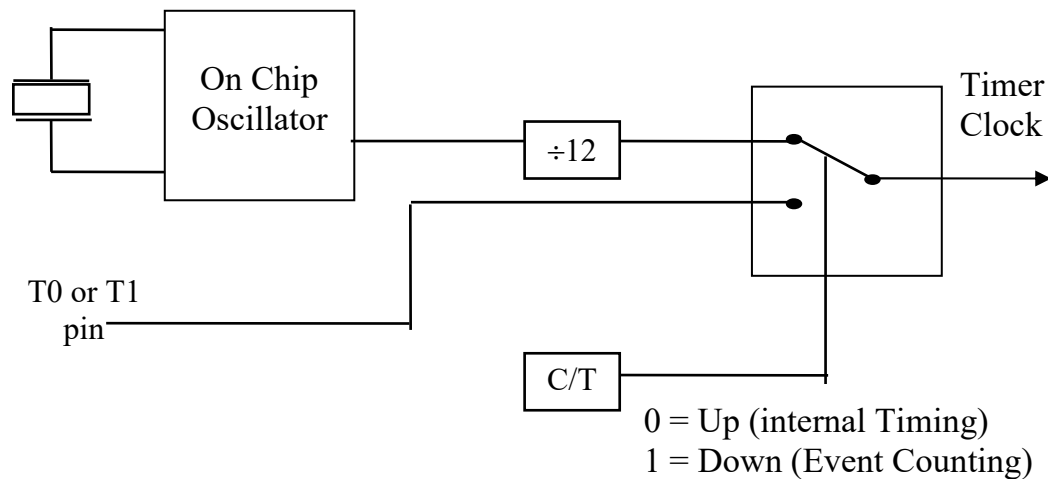
- Thanh ghi điều khiển bao gồm các bit trạng thái và các bit điều khiển bởi Timer 0 và Timer 1. Thanh ghi TCON có bit định vị. Hoạt động của từng bit được tóm tắt như sau :

Bit	Symbol	Bit Address	Description
TCON.7	TF1	8FH	Cờ tràn Timer 1 được set bởi phần cứng ở sự tràn, được xóa bởi phần mềm hoặc bởi phần cứng khi các vector xử lý đến thủ tục phục vụ ngắt ISR
TCON.6	TR1	8EH	Bit điều khiển chạy Timer 1 được set hoặc xóa bởi phần mềm để chạy hoặc ngưng chạy Timer.
TCON.5	TF0	8DH	Cờ tràn Timer 0(hoạt động tương tự TF1)
TCON.4	TR0	8CH	Bit điều khiển chạy Timer 0 (giống TR1)
TCON.3	IE1	8BH	Cờ kiểu ngắt 1 ngoài. Khi cạnh xuống xuất hiện trên INT1 thì IE1 được xóa bởi phần mềm hoặc phần cứng khi CPU định hướng đến thủ tục phục vụ ngắt ngoài.
TCON.2	IT1	8AH	Cờ kiểu ngắt 1 ngoài được set hoặc xóa bằng phần mềm bởi cạnh kích hoạt bởi sự ngắt ngoài.
TCON.1	IE0	89H	Cờ cạnh ngắt 0 ngoài
TCON	IT0	88H	Cờ kiểu ngắt 0 ngoài.

### 2.3. Các nguồn xung nhịp cho timer (clock sources):

- Có hai nguồn xung clock có thể đếm giờ là sự định giờ bên trong và sự đếm sự kiện bên ngoài. Bit C/T trong TMOD cho phép chọn 1 trong 2 khi Timer được khởi động.

Crystal



#### ◆ Đếm giờ bên trong (Interval Timing):

- Nếu bit C/T = 0 thì hoạt động của Timer liên tục được chọn vào bộ Timer được ghi giờ từ dao động trên Chip. Một bộ chia 12 được thêm vào để giảm tần số clock đến 1 giá trị phù hợp với các ứng dụng. Các thanh ghi TLx và THx tăng ở tốc độ 1/12 lần tần số dao động trên Chip. Nếu dùng thạch anh 12MHz thì sẽ đưa đến tốc độ clock 1MHz.

- Các sự tràn Timer sinh ra sau một con số cố định của những xung clock, nó phụ thuộc vào giá trị khởi tạo được LOAD vào các thanh ghi THx và TLx.

#### ◆ Đếm các sự kiện bên trong (Event Counting) :

- Nếu bit C/T = 1 thì bộ Timer được ghi giờ từ nguồn bên ngoài trong nhiều ứng dụng, nguồn bên ngoài này cung cấp 1 sự định giờ với 1 xung trên sự xảy ra của sự kiện. Sự định giờ là sự đếm sự kiện. Con số sự kiện được xác định trong phần mềm bởi việc đọc các thanh ghi Timer. TLx/THx, bởi vì giá trị 16 bit trong các thanh này tăng lên cho mỗi sự kiện.

- Nguồn xung clock bên ngoài đưa vào chân P3.4 là ngõ nhập của xung clock bởi Timer 0 (T0) và P3.5 là ngõ nhập của xung clock bởi Timer 1 (T1).

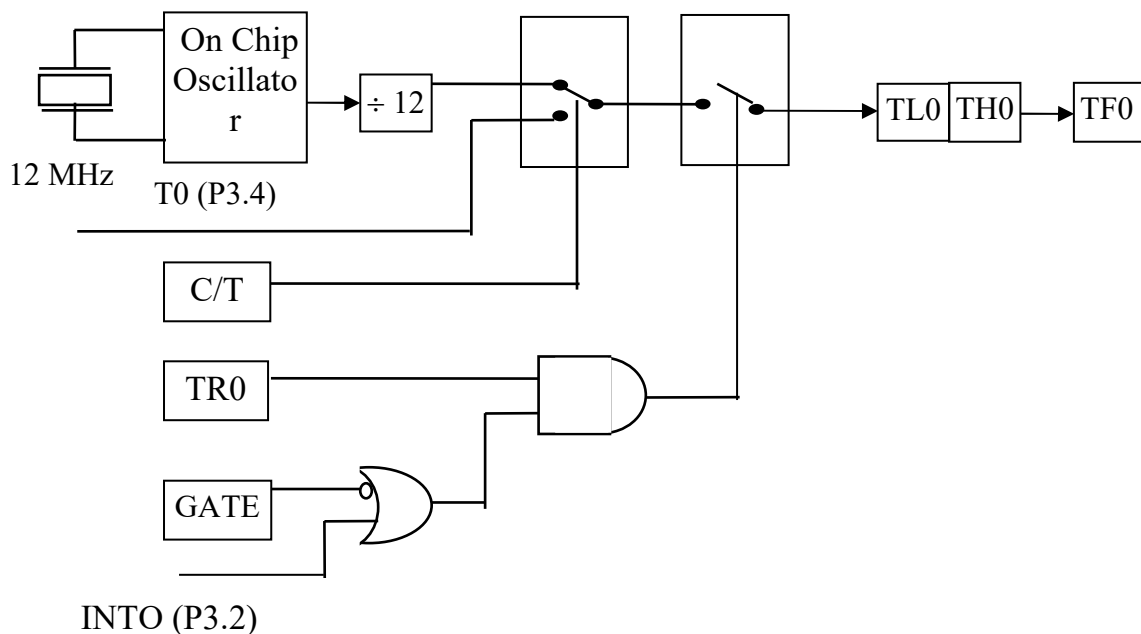
- Trong các ứng dụng đếm các thanh ghi Timer được tăng trong đáp ứng của sự chuyển trạng thái từ 1 sang 0 ở ngõ nhập Tx. Ngõ nhập bên ngoài được thử trong suốt S5P2 của mọi chu kỳ máy: Do đó khi ngõ nhập đưa tới mức cao trong một chu kỳ và mức thấp trong một chu kỳ kế tiếp thì bộ đếm tăng lên một. Giá trị mới xuất hiện trong các thanh ghi Timer trong suốt S5P1 của chu kỳ theo sau một sự chuyển đổi. Bởi vì nó

chiếm 2 chu kỳ máy ( $2\mu s$ ) để nhận ra sự chuyển đổi từ 1 sang 0, nên tần số bên ngoài lớn nhất là 500KHz nếu dao động thạch anh 12 MHz.

#### 2.4. Quá trình bắt đầu đếm, kết thúc đếm và điều khiển timer (starting, stopping and controlling the timer) :

- Bit TRx trong thanh ghi có bit định vị TCON được điều khiển bởi phần mềm để bắt đầu hoặc kết thúc các Timer. Để bắt đầu các Timer ta set bit TRx và để kết thúc Timer ta Clear TRx. Ví dụ Timer 0 được bắt đầu bởi lệnh SETB TR0 và được kết thúc bởi lệnh CLR TR0 (bit Gate= 0). Bit TRx bị xóa sau sự reset hệ thống, do đó các Timer bị cấm bằng sự mặc định.

- Thêm phương pháp nữa để điều khiển các Timer là dùng bit GATE trong thanh ghi TMOD và ngõ nhập bên ngoài INTx. Điều này được dùng để đo các độ rộng xung. Giả sử xung đưa vào chân INTO ta khởi động Timer 0 cho mode 1 là mode Timer 16 bit với TL0/TH0 = 0000H, GATE = 1, TR0 = 1. Như vậy khi INTO = 1 thì Timer “được mở công” và ghi giờ với tốc độ của tần số 1MHz. Khi INTO xuống thấp thì Timer “đóng công” và khoảng thời gian của xung tính bằng  $\mu s$  là sự đếm được trong thanh ghi TL0/TH0.



**Timer Operating Mode 1.**

#### 2.5. Sự khởi động và truy xuất các thanh ghi timer:

- Các Timer được khởi động 1 lần ở đầu chương trình để đặt mode hoạt động cho chúng. Sau đó trong chương trình các Timer được bắt đầu, được xóa, các thanh ghi Timer được đọc và cập nhật ... theo yêu cầu của từng ứng dụng cụ thể.

- Mode Timer TMOD là thanh ghi đầu tiên được khởi gán, bởi vì đặt mode hoạt động cho các Timer. Ví dụ khởi động cho Timer 1 hoạt động ở mode 1 (mode Timer 16bit) và được ghi giờ bằng dao động trên Chip ta dùng lệnh : MOV TMOD, # 00001000B. Trong lệnh này M1 = 0, M0 = 1 để vào mode 1 và C/T = 0, GATE = 0 để cho phép



ghi giờ bên trong đồng thời xóa các bit mode của Timer 0. Sau lệnh trên Timer vẫn chưa đếm giờ, nó chỉ bắt đầu đếm giờ khi set bit điều khiển chạy TR1 của nó.

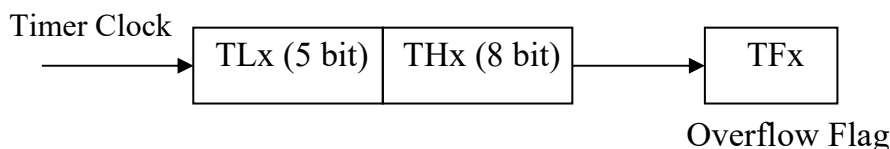
- Nếu ta không khởi gán giá trị đầu cho các thanh ghi TLx/THx thì Timer sẽ bắt đầu đếm từ 0000H lên và khi tràn từ FFFFH sang 0000H nó sẽ bắt đầu tràn TFx rồi tiếp tục đếm từ 0000H lên tiếp . . .
- Nếu ta khởi gán giá trị đầu cho TLx/THx, thì Timer sẽ bắt đầu đếm từ giá trị khởi gán đó lên nhưng khi tràn từ FFFFH sang 0000H lại đếm từ 0000H lên.
- Chú ý rằng cờ tràn TFx tự động được set bởi phần cứng sau mỗi sự tràn và sẽ được xóa bởi phần mềm. Chính vì vậy ta có thể lập trình chờ sau mỗi lần tràn ta sẽ xóa cờ TFx và quay vòng lặp khởi gán cho TLx/THx để Timer luôn luôn bắt đầu đếm từ giá trị khởi gán lên theo ý ta mong muốn.

Đặc biệt những sự khởi gán nhỏ hơn 256  $\mu$ s, ta sẽ gọi mode Timer tự động nạp 8 bit của mode 2. Sau khi khởi gán giá trị đầu vào THx, khi set bit TRx thì Timer sẽ bắt đầu đếm giá trị khởi gán và khi tràn từ FFH sang 00H trong TLx, cờ TFx tự động được set đồng thời giá trị khởi gán mà ta khởi gán cho Thx được nạp tự động vào TLx và Timer lại được đếm từ giá trị khởi gán này lên. Nói cách khác, sau mỗi tràn ta không cần khởi gán lại cho các thanh ghi Timer mà chúng vẫn đếm được lại từ giá trị ban đầu.

### **3. Các chế độ Timer và cờ tràn ( Timer Modes And Overflow ):**

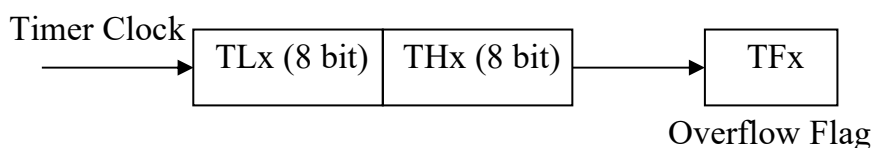
- 8951 có 2 Timer là Timer 0 và timer 1. Ta dùng ký hiệu TLx và Thx để chỉ 2 thanh ghi byte thấp và byte cao của Timer 0 hoặc Timer 1.

#### **3.1. Mode Timer 13 bit (MODE 0) :**



- Mode 0 là mode Timer 13 bit, trong đó byte cao của Timer (Thx) được đặt thấp và 5 bit trọng số thấp nhất của byte thấp Timer (TLx) đặt cao để hợp thành Timer 13 bit. 3 bit cao của TLx không dùng.

#### **3.2. Mode Timer 16 bit (MODE 1) :**

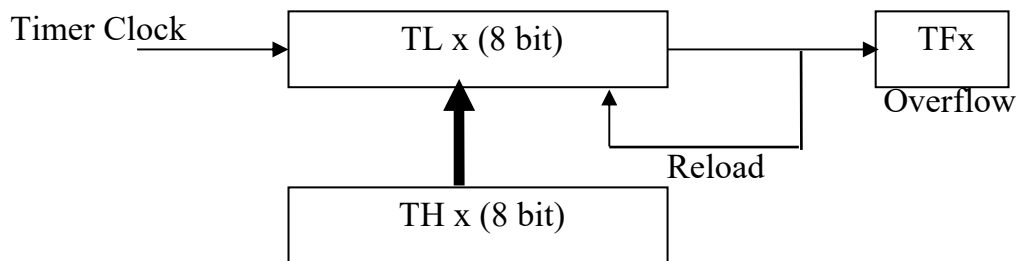


- Mode 1 là mode Timer 16 bit, tương tự như mode 0 ngoại trừ Timer này hoạt động như một Timer đầy đủ 16 bit, xung clock được dùng với sự kết hợp các thanh ghi cao và thấp (TLx, THx). Khi xung clock được nhận vào, bộ đếm Timer tăng lên 0000H,

0001H, 0002H, ..., và một sự tràn sẽ xuất hiện khi có sự chuyển trên bộ đếm Timer từ FFFH sang 0000H và sẽ set cờ tràn Time, sau đó Timer đếm tiếp.

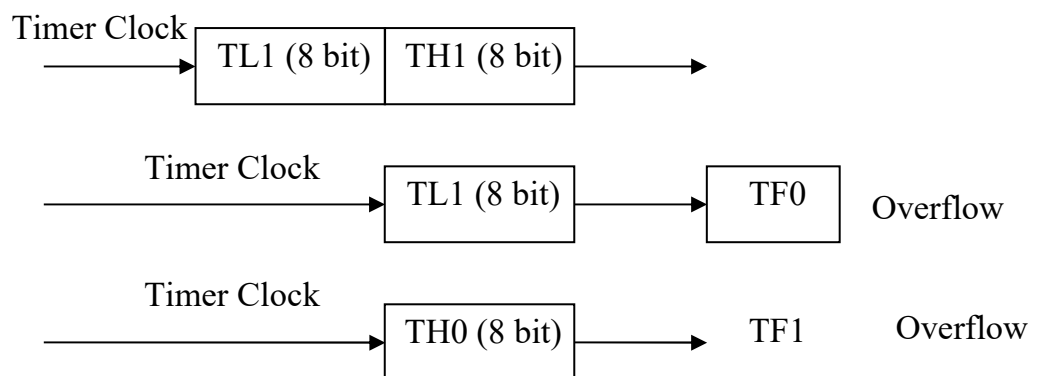
- Cờ tràn là bit TFX trong thanh ghi TCON mà nó sẽ được đọc hoặc ghi bởi phần mềm.
- Bit có trọng số lớn nhất (MSB) của giá trị trong thanh ghi Timer là bit 7 của THx và bit có trọng số thấp nhất (LSB) là bit 0 của TLx. Bit LSB đổi trạng thái ở tần số clock vào được chia  $2^{16} = 65.536$ .
- Các thanh ghi Timer TLx và Thx có thể được đọc hoặc ghi tại bất kỳ thời điểm nào bởi phần mềm.

### 3.3. Mode tự động nạp 8 bit (MODE 2) :



- Mode 2 là mode tự động nạp 8 bit, byte thấp TLx của Timer hoạt động như một Timer 8 bit trong khi byte cao THx của Timer giữ giá trị Reload. Khi bộ đếm tràn từ FFH sang 00H, không chỉ cờ tràn được set mà giá trị trong THx cũng được nạp vào TLx : Bộ đếm được tiếp tục từ giá trị này lên đến sự chuyển trạng thái từ FFH sang 00H kế tiếp và cứ thế tiếp tục. Mode này thì phù hợp bởi vì các sự tràn xuất hiện cụ thể mà mỗi lúc nghỉ thanh ghi TMOD và THx được khởi động.

### 3.4 Mode Timer tách ra (MODE 3) :



- Mode 3 là mode Timer tách ra và là sự khác biệt cho mỗi Timer.
- Timer 0 ở mode 3 được chia là 2 timer 8 bit. TL0 và TH0 hoạt động như những Timer riêng lẻ với sự tràn sẽ set các bit TL0 và TF1 tương ứng.

- Timer 1 bị dừng lại ở mode 3, nhưng có thể được khởi động bởi việc ngắt nó vào một trong các mode khác. Chỉ có nhược điểm là cờ tràn TF1 của Timer 1 không bị ảnh hưởng bởi các sự tràn của Timer 1 bởi vì TF1 được nối với TH0.
- Mode 3 cung cấp 1 Timer ngoại 8 bit là Timer thứ ba của 8951. Khi vào Timer 0 ở mode 3, Timer có thể hoạt động hoặc tắt bởi sự ngắt nó ra ngoài và vào trong mode của chính nó hoặc có thể được dùng bởi Port nối tiếp như là một máy phát tốc độ Baud, hoặc nó có thể dùng trong hướng nào đó mà không sử dụng Interrupt.

## **V. HOẠT ĐỘNG PORT NỐI TIẾP :**

### **1. Giới thiệu :**

- 8951 có một port nối tiếp trong chip có thể hoạt động ở nhiều chế độ trên một dãy tần số rộng. Chức năng chủ yếu là thực hiện chuyển đổi song song sang nối tiếp với dữ liệu xuất và chuyển đổi nối tiếp sang song song với dữ liệu nhập.
- Port nối tiếp cho hoạt động song công (full duplex: thu và phát đồng thời) và đệm thu (receiver buffering) cho phép một ký tự sẽ được thu và được giữ trong khi ký tự thứ hai được nhận. Nếu CPU đọc ký tự thứ nhất trước khi ký tự thứ hai được thu đầy đủ thì dữ liệu sẽ không bị mất.
- Hai thanh ghi chức năng đặc biệt cho phép phần mềm truy xuất đến port nối tiếp là: SBUF và SCON. Bộ đệm port nối tiếp (SBUF) ở địa chỉ 99H nhận dữ liệu để thu hoặc phát. Thanh ghi điều khiển port nối tiếp (SCON) ở địa chỉ 98H là thanh ghi có địa chỉ bit chứa các bit trạng thái và các bit điều khiển. Các bit điều khiển đặt chế độ hoạt động cho port nối tiếp, và các bit trạng thái Báo cáo kết thúc việc phát hoặc thu ký tự. Các bit trạng thái có thể được kiểm tra bằng phần mềm hoặc có thể lập trình để tạo ngắt.

### **2. Các thanh ghi và các chế độ hoạt động của port nối tiếp:**

#### **2.1. Thanh ghi điều khiển port nối tiếp:**

- Chế độ hoạt động của port nối tiếp được đặt bằng cách ghi vào thanh ghi chế độ port nối tiếp (SCON) ở địa chỉ 98H. Sau đây các bản tóm tắt thanh ghi SCON và các chế độ của port nối tiếp:

Bit	Ký hiệu	Địa chỉ	Mô tả
SCON.7	SM0	9FH	Bit 0 của chế độ port nối tiếp
SCON.6	SM1	9EH	Bit 1 của chế độ port nối tiếp
SCON.5	SM3	9DH	Bit 2 của chế độ port nối tiếp. Cho phép truyền thông xử lý trong các chế độ 2 và 3, RI sẽ không bị tác động nếu bit thứ 9 thu được là 0
SCON.4	REN	9CH	Cho phép bộ thu phải được đặt lên 1 để thu các ký tự
SCON.3	TB8	9BH	Bit 8 phát, bit thứ 9 được phát trong chế độ 2 và 3, được đặt và xóa bằng phần mềm.

SCON.2	RB8	9AH	B it 8 thu, bit thứ 9 thu được Cờ ngắt phát. Đặt lên 1 khi kết thúc phát ký tự, được xóa bằng phần mềm Cờ ngắt thu. Đặt lên 1 khi kết thúc thu ký tự, được xóa bằng phần mềm
SCON.1	TI	99H	
SCON.0	RI	98H	

### Tóm tắt thanh ghi chế độ port nối tiếp

SM0	SM1	Chế độ	Mô tả	Tốc độ baud
0	0	0	Thanh ghi dịch	Cố định ( $F_{osc}/12$ )
0	1	1	UART 8 bit	Thay đổi (đặt bằng timer)
1	0	2	UART 9 bit	Cố định ( $F_{osc}/12$ hoặc $F_{osc}/64$ )
1	1	3	UART 9 bit	Thay đổi (đặt bằng timer)

### Các chế độ port nối tiếp

- Trước khi sử dụng port nối tiếp, phải khởi động SCON cho đúng chế độ. Ví dụ, lệnh sau:

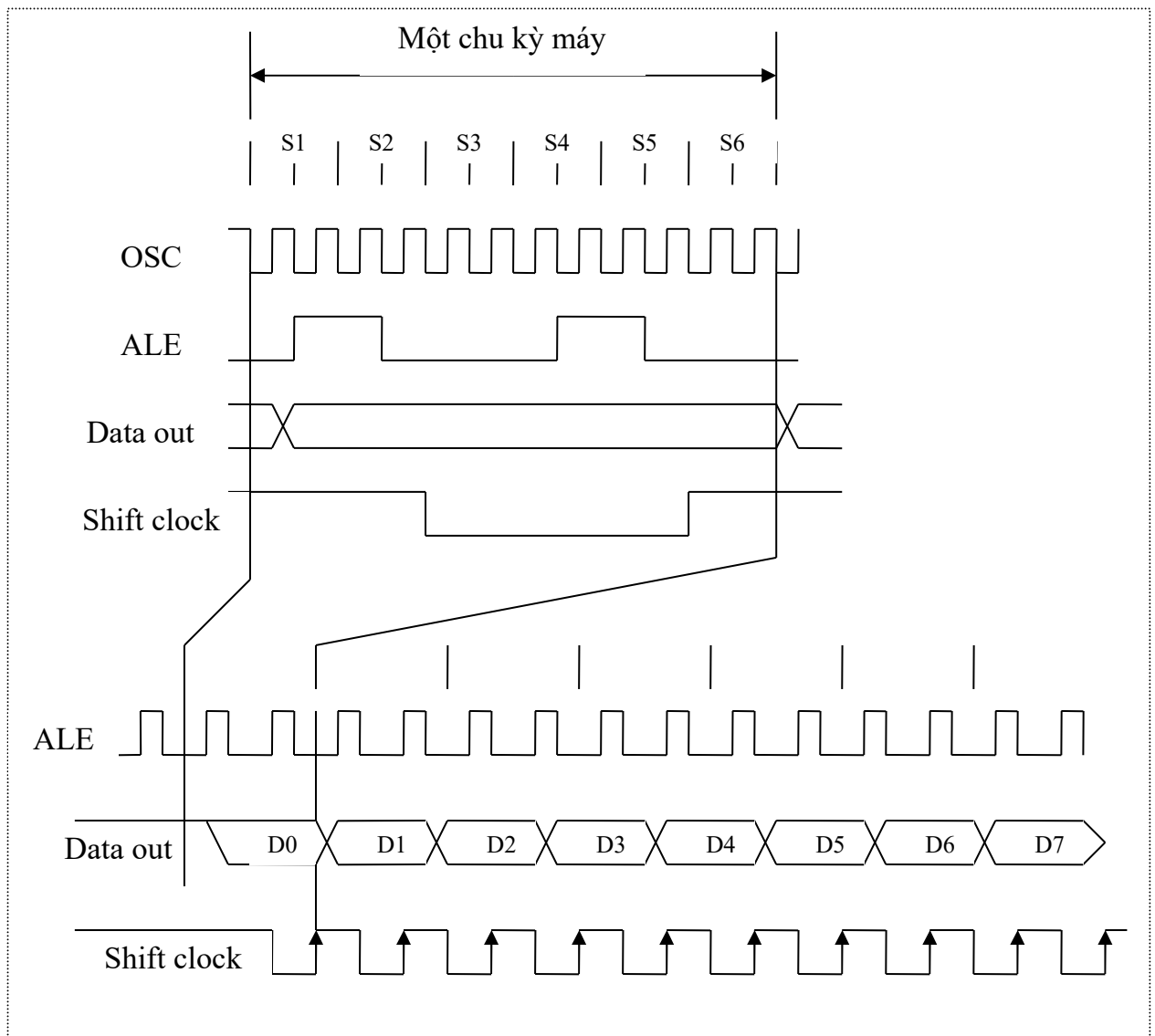
```
MOV SCON, #01010010B
```

Khởi động port nối tiếp cho chế độ 1 (SM0/SM1=0/1), cho phép bộ thu (REN=1) và cờ ngắt phát (TP=1) để bộ phát sẵn sàng hoạt động.

#### 2.2. MODE 0 (Thanh ghi dịch đơn 8 bit):

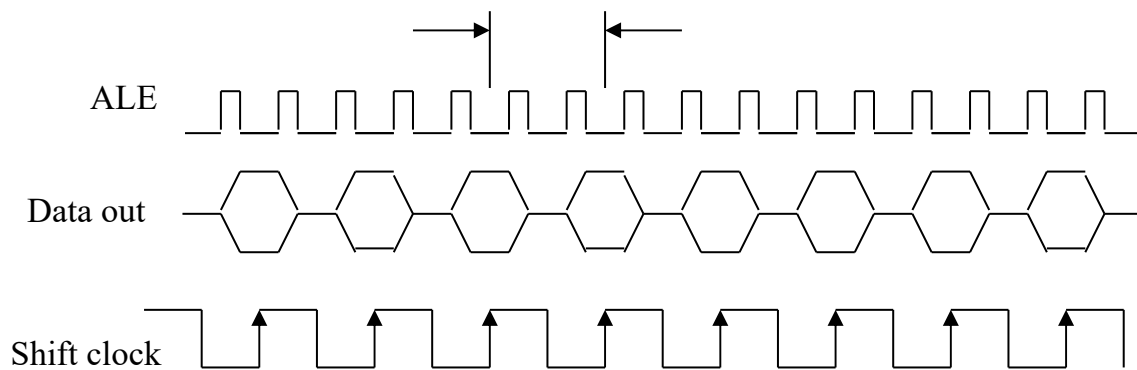
- Chế độ 0 được chọn bằng các thanh ghi các bit 0 vào SM1 và SM2 của SCON, đưa port nối tiếp vào chế độ thanh ghi dịch 8bit. Dữ liệu nối tiếp vào và ra qua RXD và TXD xuất xung nhịp dịch, 8 bit được phát hoặc thu với bit đầu tiên là LSB. Tốc độ baud cố định ở 1/12 tần số dao động trên chip.

- Việc phát đi được khởi động bằng bất cứ lệnh nào ghi dữ liệu vào SBUF. Dữ liệu dịch ra ngoài trên đường RXD (P3.0) với các xung nhịp được gửi ra đường TXD (P3.1). Mỗi bit phát đi hợp lệ (trên RXD) trong một chu kỳ máy, tín hiệu xung nhập xuống thấp ở S3P1 và trở về cao ở S6P1.



### Giải đồ thời gian Port nối tiếp phát ở chế độ 0

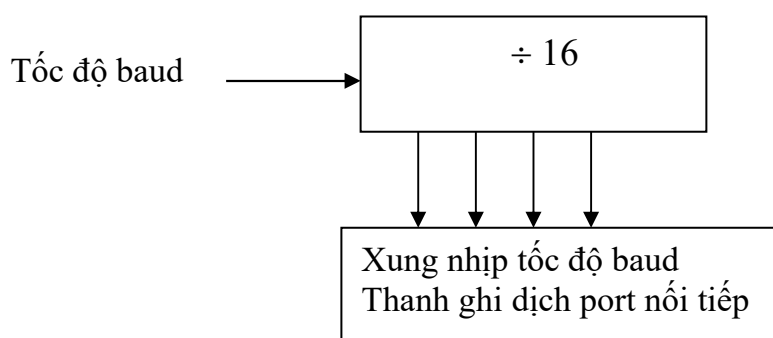
- Việc thu được khởi động khi cho phép bộ thu (REN) là 1 và bit ngắt thu (RI) là 0. Quy tắc tổng quát là đặt REN khi bắt đầu chương trình để khởi động port nối tiếp, rồi xoá RI để bắt đầu nhận dữ liệu. Khi RI bị xoá, các xung nhịp được đưa ra đường TXD, bắt đầu chu kỳ máy kế tiếp và dữ liệu theo xung nhịp ở đường RXD. Lấy xung nhịp cho dữ liệu vào port nối tiếp xảy ra ở cạnh đường của TXD.



**Giản đồ thời gian phát nối tiếp ở chế độ 0**

### 2.3. MODE 1 (UART 8 bit với tốc độ baud thay đổi được):

- Ở chế độ 1, port nối tiếp của 8951 làm việc như một UART 8 bit với tốc độ baud thay đổi được. Một UART (Bộ thu phát đồng bộ vạn năng) là một dụng cụ thu phát dữ liệu nối tiếp với mỗi ký tự dữ liệu đi trước là bit start ở mức thấp và theo sau bit stop ở mức cao. Đôi khi xen thêm bit kiểm tra chẵn lẻ giữa bit dữ liệu cuối cùng và bit stop. Hoạt động chủ yếu của UART là chuyển đổi song song sang nối tiếp với dữ liệu nhập.
- Ở chế độ 1, 10 bit được phát trên TXD hoặc thu trên RXD. Những bit đó là: 1 bit start (luôn luôn là 0), 8 bit dữ liệu (LSB đầu tiên) và 1 bit stop (luôn luôn là 1). Với hoạt động thu, bit stop được đưa vào RB8 trong SCON. Trong 8951 chế độ baud được đặt bằng tốc độ báo tràn của timer 1.
- Tạo xung nhịp và đồng bộ hóa các thanh ghi dịch của port nối tiếp trong các chế độ 1,2 và 3 được thiết lập bằng bộ đếm 4 bit chia cho 16, ngõ ra là xung nhịp tốc độ baud. Ngõ vào của bộ đếm này được chọn qua phần mềm



### 2.4. UART 9 bit với tốc độ baud cố định (MODE 2):

- Khi SM1=1 và SM0=0, cổng nối tiếp làm việc ở chế độ 2, như một UART 9bit có tốc độ baud cố định, 11 bit sẽ được phát hoặc thu: 1bit start, 8 bit data, 1 bit data thứ 9 có thể được lập trình và 1 bit stop. Khi phát bit thứ 9 là bất cứ gì đã được đưa vào TB8 trong SCON (có thể là bit Parity). Khi thu bit thứ 9 thu được sẽ ở trong RB8. Tốc độ baud ở chế độ 2 là 1/32 hoặc 1/16 tần số dao động trên chip.

**2.5. UART 9 bit với tốc độ baud thay đổi được (MODE 3):**

- Chế độ này giống như ở chế độ 2 ngoại trừ tốc độ baud có thể lập trình được và được cung cấp bởi Timer. Thật ra các chế độ 1, 2, 3 rất giống nhau. Cái khác biệt là ở tốc độ baud (cố định trong chế độ 2, thay đổi trong chế độ 1 và 3) và ở số bit data (8 bit trong chế độ 1, 9 trong chế độ 2 và 3).

**2.6. Quá trình khởi động và truy xuất các thanh ghi của PORT nối tiếp:****◆ Cho Phép Thu (Receive Enable):**

- Bit cho phép bộ thu (REN=Receiver Enable) Trong SCON phải được đặt lên 1 bằng phần mềm để cho phép thu các ký tự thông thường thực hiện việc này ở đầu chương trình khi khởi động cổng nối tiếp, timer ... Có thể thực hiện việc này theo hai cách.

Lệnh:

SETB REN ; đặt REN lên 1

Hoặc lệnh

MOV SCON,#XXX1XXXXB ; đặt REN lên 1 hoặc xoá các bit khác trên SCON khi cần (các X phải là 0 hoặc 1 để đặt chế độ làm việc)

**◆ Bit dữ liệu thứ 9 (the 9<sup>th</sup> data bit):**

- Bit dữ liệu thứ 9 cần phát trong các chế độ 2 và 3 phải được nạp vào trong TB8 bằng phần mềm. Bit dữ liệu thứ 9 thu được đặt ở RB8. Phần mềm có thể cần hoặc không cần bit dữ liệu thứ 9, phụ thuộc vào đặc tính kỹ thuật của thiết bị nối tiếp sử dụng (bit dữ liệu thứ 9 cũng đóng vai trò quan trọng trong truyền thông đa xử lý )

**◆ Thêm 1 bit parity:**

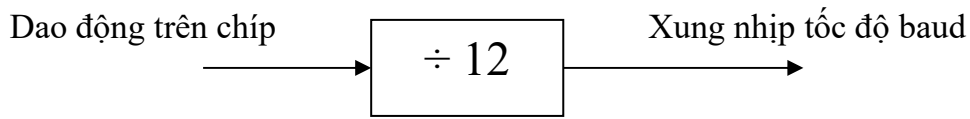
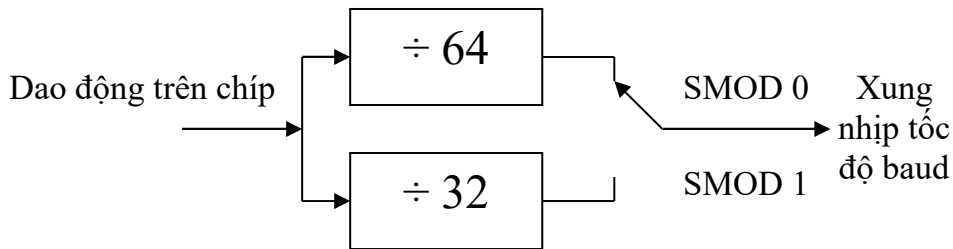
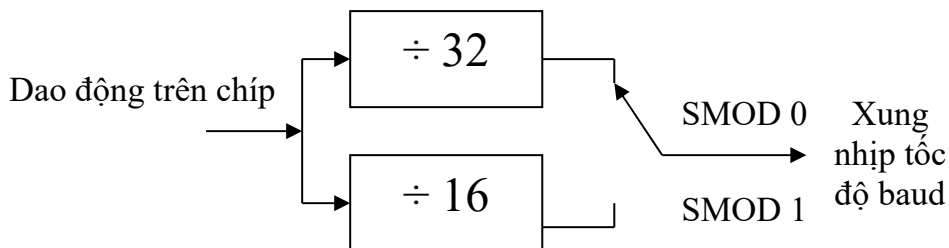
- Thường sử dụng bit dữ liệu thứ 9 để thêm parity vào ký tự. Như đã nhận xét ở chương trước, bit P trong từ trạng thái chương trình (PSW) được đặt lên 1 hoặc bị xoá bởi chu kỳ máy để thiết lập kiểm tra chẵn với 8 bit trong thanh tích lũy.

**◆ Các cờ ngắt:**

- Hai cờ ngắt thu và phát (RI và TI) trong SCON đóng một vai trò quan trọng trong truyền thông nối tiếp dùng 8951/8051. Cả hai bit được đặt lên 1 bằng phần cứng, nhưng phải được xoá bằng phần mềm.

**2.7. Tốc độ baud port nối tiếp :**

- Như đã nói, tốc độ baud cố định ở các chế độ 0 và 2. Trong chế độ 0 nó luôn luôn là tần số dao động trên chip được chia cho 12. Thông thường thạch anh ấn định tần số dao động trên chip nhưng cũng có thể sử dụng nguồn xung nhịp khác.

**a.MODE 0****b.MODE 2****c.MODE 1 và MODE 3**

## 2.8. Các nguồn tạo xung nhịp cho port nối tiếp

- Mặc nhiên sau khi reset hệ thống, tốc độ baud chế độ 2 là tần số bộ dao động chia cho 64, tốc độ baud cũng bị ảnh hưởng bởi 1 bit trong thanh ghi điều khiển nguồn cung cấp (PCON) bit 7 của PCON là bit SMOD. Đặt bit SMOD lên 1 làm gấp đôi tốc độ baud trong các chế độ 1, 2 và 3. Trong chế độ 2, tốc độ baud có thể bị gấp đôi từ giá trị mặc nhiên của 1/64 tần số dao động (SMOD=0) đến 1/32 tần số dao động (SMOD=1)

- Vì PCON không được định địa chỉ theo bit, nên để đặt bit SMOD lên 1 cần phải theo các lệnh sau:

- + MOV A,PCON : nhập vào A giá trị hiện hành của PCON
- + SETB ACC.7 : Set bit của ACC (bit SMOD)
- + MOV PCON,A : ghi giá trị ngược về PCON mà SMOD đã được set.



- Các tốc độ baud trong các chế độ 1 và 3 được xác định bằng tốc độ tràn của timer 1. Vì timer hoạt động ở tần số tương đối cao, tràn timer được chia thêm cho 32 (hoặc 16 nếu SMOD =1 ) trước khi cung cấp tốc độ xung nhịp cho port nối tiếp.

### **3.Tổ chức ngắt trong 8951 (INTERRUPT ORGANIZATION) :**

Vi Điều Khiển có 5 nguồn ngắt:2 nguồn ngắt ngoài,2 ngắt timer và 1 ngắt Port nối tiếp, tất cả các nguồn ngắt bị cấm sau khi reset hệ thống và cho phép bởi phần mềm.

#### **3.1.Cho phép và không cho phép ngắt :**

Mỗi nguồn ngắt được cho phép hoặc không cho phép thông qua thanh ghi chức năng đặc biệt có các bit được địa chỉ hóa IE (Interrupt Enable) tại địa chỉ 0A8H.

<b>BIT</b>	<b>SYMBOL</b>	<b>BIT ADDRESS</b>	<b>DESCRIPTION (1:ENABLE,0:DISABLE)</b>
IE.7	EA	AFH	Global Enable/Disable
IE.6	EA	AEH	Undefined
IE.5	ET2	ADH	Enable Timer 2 Interrupt (8052)
IE.4	ES	ACH	Enable Serial Port Interrupt
IE.3	ET1	ABH	Enable Timer 1 Interrupt
IE.2	EX1	AAH	Enable External 1 Interrupt
IE.1	ET0	A9H	Enable Timer 0 Interrupt
IE.0	EX0	A8H	Enable External 0 Interrupt

#### **3.2.Véctơ ngắt :**

Khi ngắt được chấp nhận giá trị được đưa vào PC (Program Counter) gọi là vector ngắt (Interrupt Vector)

<b>INTERRUPT</b>	<b>FLAG</b>	<b>VECTOR ADDRESS</b>
System Reset	RST	0000 H
External 0	IE0	0003 H
Timer 0	TF0	000B H
External 1	IE1	0013 H
Timer 1	TF1	001B H
Serial Port	RI OR TI	0023 H
Timer 2	TF2 OR EXF2	002B H

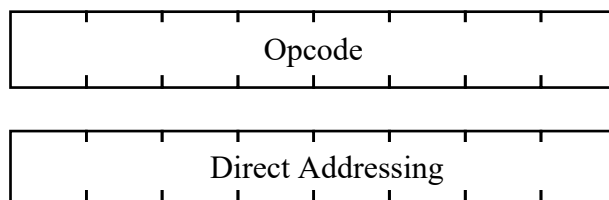
#### **3.3.Ngắt Port nối tiếp**

Ngắt Port nối tiếp xảy ra khi cả 2 cờ ngắt truyền (TI) hoặc cờ ngắt nhận (RI) được đặt. Ngắt truyền xảy ra khi bit cuối cùng trong SBUF truyền xong tức là lúc này thanh ghi SBUF rỗng .Ngắt nhận xảy ra khi SBUF đã hoàn thành việc nhận và đang đợi để đọc tức là lúc này thanh ghi SBUF đầy. Cả hai cờ ngắt này được đặt bởi phần cứng và xóa bằng phần mềm.



## 2. Định địa chỉ trực tiếp (Direct Addressing):

- Sự định địa chỉ trực tiếp có thể truy xuất bất kỳ giá trị nào trên Chip hoặc thanh ghi phần cứng trên Chip. Một byte địa chỉ trực tiếp được đưa vào Opcode để định rõ vị trí được dùng như sau:

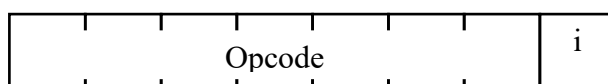


- Tùy thuộc các bit bậc cao của địa chỉ trực tiếp mà một trong 2 vùng nhớ được chọn. Khi bit 7 = 0, thì địa chỉ trực tiếp ở trong khoảng 0÷127 (00H÷7FH) và 128 vị trí nhớ thấp của RAM trên Chip được chọn.

- Tất cả các Port I/O, các thanh ghi chức năng đặc biệt, thanh ghi điều khiển hoặc thanh ghi trạng thái bao giờ cũng được quy định các địa chỉ trong khoảng 128÷255 (80÷FFH). Khi byte địa chỉ trực tiếp nằm trong giới hạn này (ứng với bit 7 = 1) thì thanh ghi chức năng đặc biệt được truy xuất. Ví dụ Port 0 và Port 1 được quy định địa chỉ trực tiếp là 80H và 90H, P0, P1 là dạng thức rút gọn thuật nhớ của Port, thì sự biến thiên cho phép thay thế và hiểu dạng thức rút gọn thuật nhớ của chúng. Chẩn hạn lệnh: MOV P1, A  $\leftarrow$  sự biên dịch sẽ xác định địa chỉ trực tiếp của Port 1 là 90H đặt vào hai byte của lệnh (byte 1 của port 0).

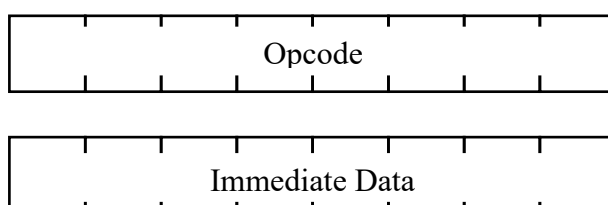
## 3. Định vị địa chỉ gián tiếp (Indirect Addressing):

- Sự định địa chỉ gián tiếp được tượng trưng bởi ký hiệu @ được đặt trước R0, R1 hay DPTR. R0 và R1 có thể hoạt động như một thanh ghi con trỏ mà nội dung của nó cho biết một địa chỉ trong RAM nội ở nơi mà dữ liệu được ghi hoặc được đọc. Bit có trọng số nhỏ nhất của Opcode lệnh sẽ xác định R0 hay R1 được dùng con trỏ Pointer.



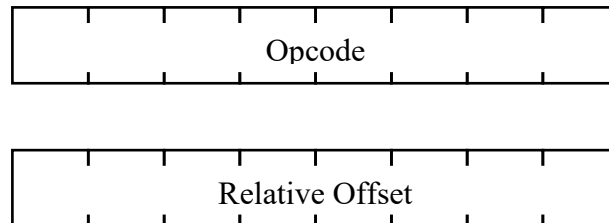
## 4. Định địa chỉ tức thời (Immediate Addressing):

- Sự định địa chỉ tức thời được tượng trưng bởi ký hiệu # được đứng trước một hằng số, 1 biến ký hiệu hoặc một biểu thức số học được sử dụng bởi các hằng, các ký hiệu, các hoạt động do người điều khiển. Trình biên dịch tính toán giá trị và thay thế dữ liệu tức thời. Byte lệnh thêm vô chứa trị số dữ liệu tức thời như sau:



**5. Định địa chỉ tương đối:**

- Sự định địa chỉ tương đối chỉ sử dụng với những lệnh nhảy nào đó. Một địa chỉ tương đối (hoặc Offset) là một giá trị 8 bit mà nó được cộng vào bộ đếm chương trình PC để tạo thành địa chỉ một lệnh tiếp theo được thực thi. Phạm vi của sự nhảy nằm trong khoảng  $-128 \div 127$ . Offset tương đối được gắn vào lệnh như một byte thêm vào như sau:

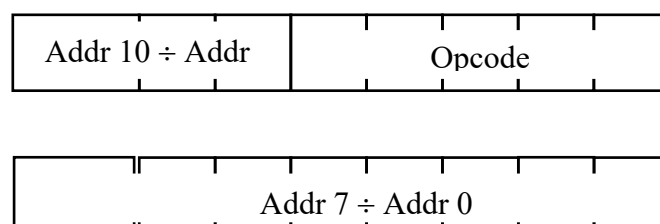


- Những nơi nhảy đến thường được chỉ rõ bởi các nhãn và trình biên dịch xác định Offset Relative cho phù hợp.

- Sự định vị tương đối đem lại thuận lợi cho việc cung cấp mã vị trí độc lập, nhưng bất lợi là chỉ nhảy ngắn trong phạm vi  $-128 \div 127$  byte.

**6. Sự định địa chỉ tuyệt đối (Absolute Addressing):**

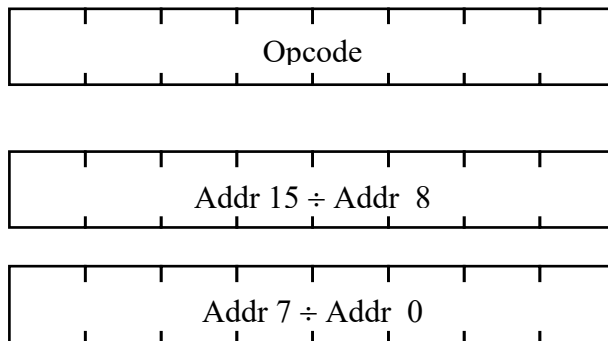
- Sự định địa chỉ tuyệt đối được dùng với các lệnh ACALL và AJMP. Các lệnh 2 byte cho phép phân chia trong trang 2K đang lưu hành của bộ nhớ mã của việc cung cấp 11 bit thấp để xác định địa chỉ trong trang 2K ( $A0 \div A10$  gồm  $A10 \div A8$  trong Opcode và  $A7 \div A0$  trong byte) và 5 bit cao để chọn trang 2K (5 bit cao đang lưu hành trong bộ đếm chương trình là 5 bit Opcode).



- Sự định vị tuyệt đối đem lại thuận lợi cho các lệnh ngắn (2 byte), nhưng bất lợi trong việc giới hạn phạm vi nơi gọi đến và cung cấp mã có vị trí độc lập.

**7. Định địa chỉ dài (Long Addressing) :**

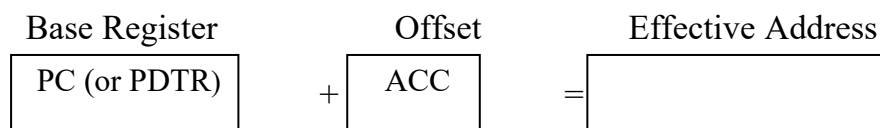
- Sự định vị dài được dùng với lệnh LCALL và LJMP. Các lệnh 3 byte này bao gồm một địa chỉ nơi gọi tới 16 bit đầy đủ là 2 byte và 3 byte của lệnh.



- Ưu điểm của sự định dài là vùng nhớ mã 64K có thể được dùng hết, nhược điểm là các lệnh đó dài 3 byte và vị trí lệ thuộc. Sự phụ thuộc vào vị trí sẽ bất lợi bởi chương trình không thể thực thi tại địa chỉ khác.

**8. Định địa chỉ chỉ số (Index Addressing):**

- Sự định địa chỉ phụ lục dùng một thanh ghi cơ bản (cũng như bộ đếm chương trình hoặc bộ đếm dữ liệu) và Offset (thanh ghi A) trong sự hình thành 1 địa chỉ liên quan bởi lệnh JMP hoặc MOVC.



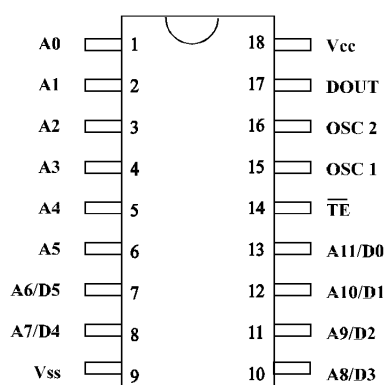
**Index Address**

- Các bảng của lệnh nhảy hoặc các bảng tra được tạo nên một cách dễ dàng bằng cách dùng địa chỉ phụ lục.

**CHƯƠNG II :**  
**KHẢO SÁT CÁC IC PHÁT PT2262, IC THU**  
**PT2272, 74HC573, DAC0808, TDA2003, EN29LV040, ... VÀ**  
**MỘT SỐ MẠCH ỨNG DỤNG**

**I. GIỚI THIỆU IC PHÁT PT2262 :**

**1. Sơ đồ chân:**

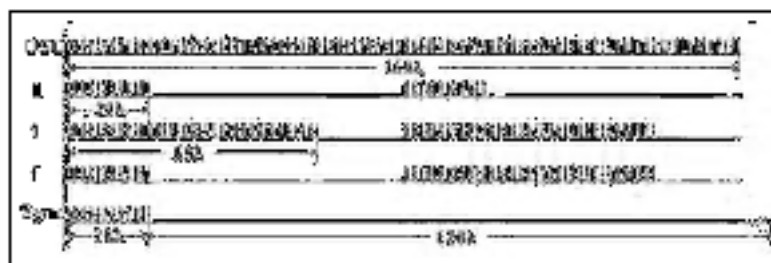


**Hình 1.1.** Sơ đồ chân của PT2262

**2. Hoạt động phát sóng RF:** tần số sóng mang được tạo ra bởi PT2262 là 38Kz.

**2.1. Các mã bit :**

- Các mã bit được điều tần cao hơn tần số sóng mang 38Kz và có tên là bit 0 , bit 1 , hoặc là bit thả nổi .Dạng sóng của chúng được biểu diễn hình bên dưới.



**Hình 1.2.** Dạng sóng bit của PT2262.

Trong đó :  $\lambda$  × chiều dài xung clock.

## 2.2. Các mã Word :

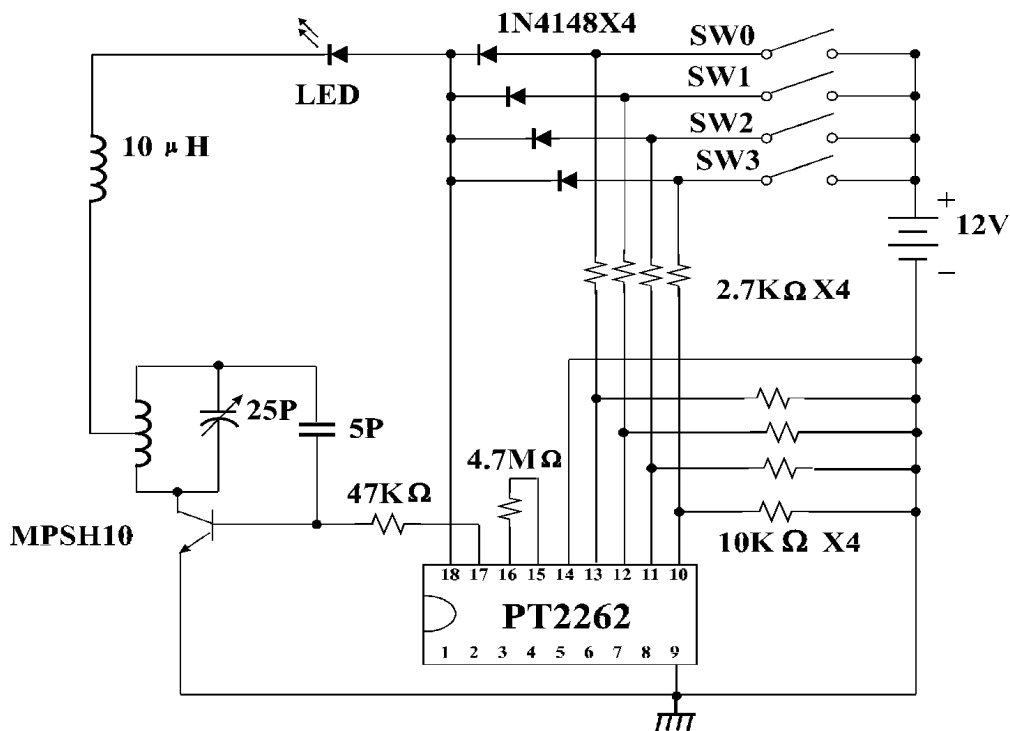
Mã Word cũng được hình thành từ các mã bit và được định dạng giống như các mã khung của RF.

## 2.3. Các mã khung :

Cũng giống như vậy ,mã khung được hình thành từ các mã Word và việc định dạng cũng giống như hoạt động của RF.

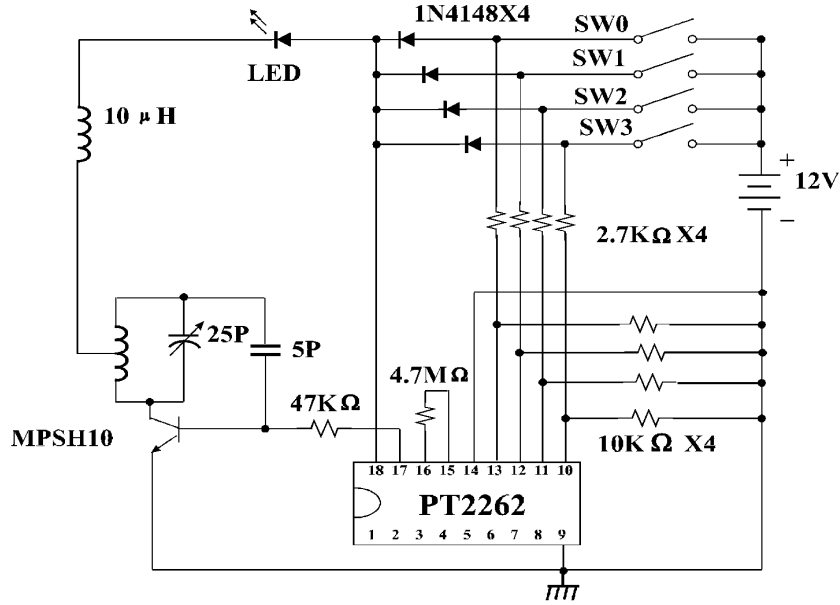
## 2.4. Bộ dao động :

PT2262 được thiết kế đặc biệt cho các ứng dụng điều khiển từ xa và ngõ ra của nó phát ra tần số sóng mang 38Kz . Để lấy được tần số sóng mang 38 KZ tại ngõ ra thì tần số của PT2272 phải đạt 76Kz . Giữa hai chân của PT2262 phải gắn điện trở 440k . Hơn nữa , để kết hợp với tần số giải âm thì giữa hai chân OSC1 và OSC2 phải gắn một điện trở 1M.



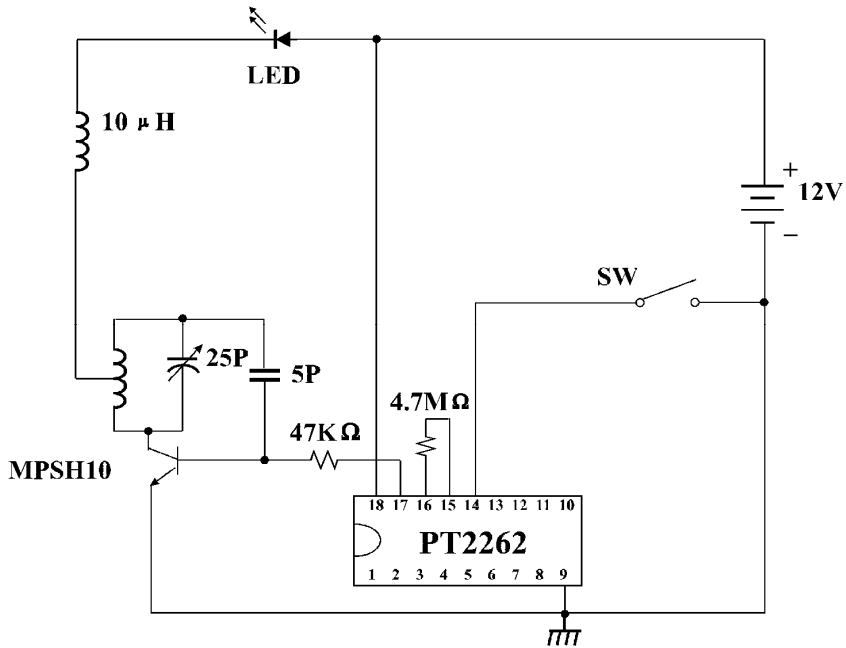
**3. Các mạch ứng dụng của IC phát PT 2262 :**

**3.1. Mạch phát 4 Data trên băng tần UHF :**



**Hình 1.3.** Mạch phát 4 Data trên băng tần UHF

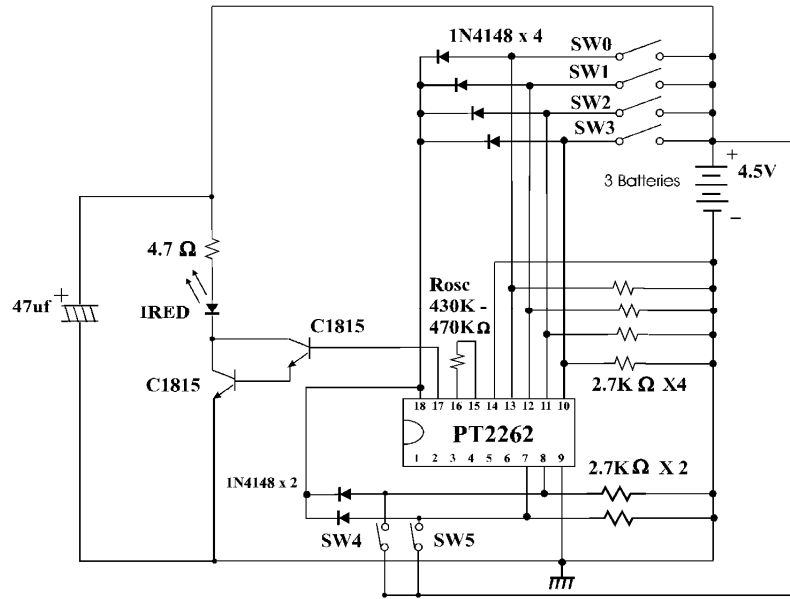
**3.2. Mạch phát địa chỉ (không Data ) trên băng tần UHF :**



**Hình 1.4.** Mạch phát địa chỉ (không Data ) trên băng tần UHF

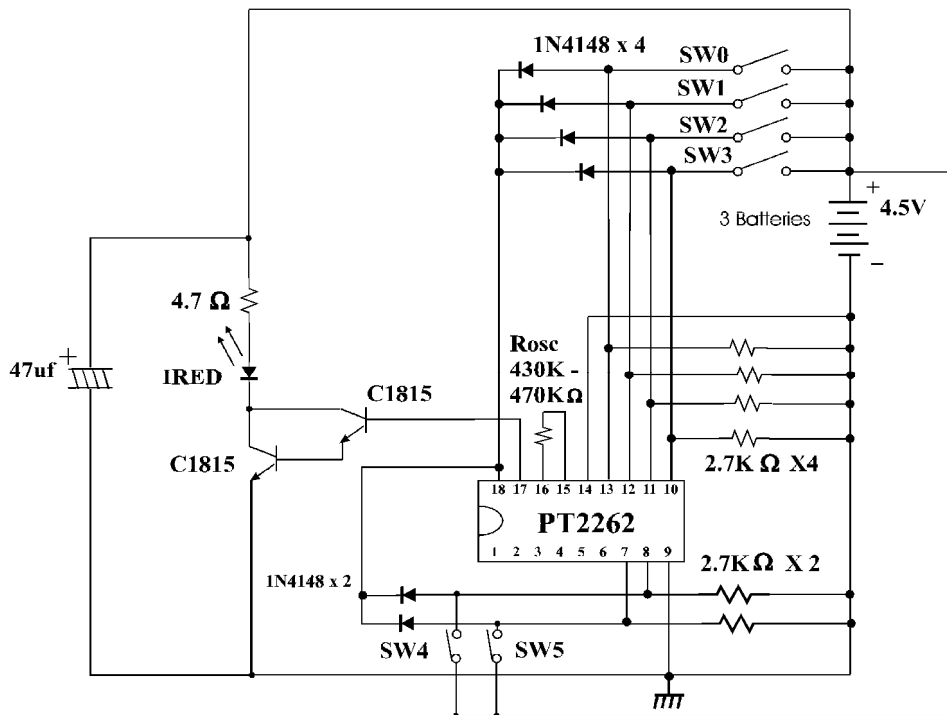


**3.3. Mạch phát địa chỉ (không Data ) không chế độ Stand-By trên băng tần UHF :**



**Hình 1.5.** Mạch phát địa chỉ (không Data ) không chế độ Stand-By trên băng tần UHF

**3.4. Mạch phát 6 Data hồng ngoại , điều chỉnh điện trở dao động để lấy sóng mang 38KHz ở chân DOUT :**

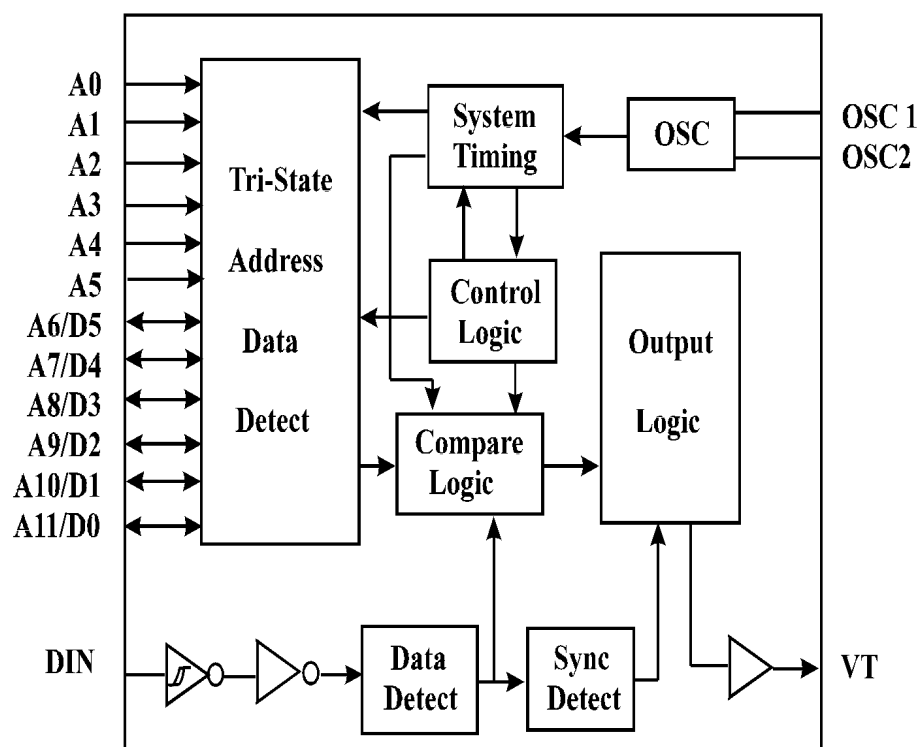


**Hình 1.6.** Mạch phát 6 Data hồng ngoại , điều chỉnh điện trở dao động để lấy sóng mang 38Kz ở chân DOUT.

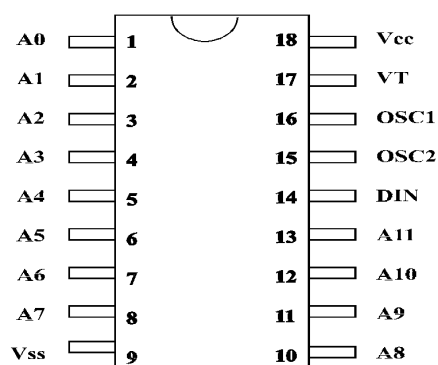
## II. GIỚI THIỆU THU PT 2272 :

Là IC giải mã điều khiển từ xa được kết hợp với PT2262 theo công nghệ CMOS . Nó gồm 12 chân địa chỉ trạng thái khởi động bit và nó cung cấp lớn nhất là  $2^{13}$  địa chỉ .PT2272 có sẵn một số lựa chọn phù hợp với mọi ứng dụng cần thiết : thay đổi ngõ ra data , loại chốt hoặc tức thời.

### 1.Sơ đồ khối:



### 2.Sơ đồ chân:



PT 2272

**Hình 2.2.** Sơ đồ chân của PT2272

**3. Ý nghĩa của các chân:**

Loại 18 chân	Loại 20 chân	Mô tả chức năng
1 → 6	1 → 6	-A <sub>0</sub> → A <sub>5</sub> : chân mã địa chỉ, 6 chân trạng thái khởi động này sẽ được xác định dạng sóng đã được mã hóa từ bit 0 → bit 5. Mỗi chân này có thể được set lên 1 hoặc 0 hoặc thả nổi.
7 → 8 10 → 13	7 → 8 12 → 15	-A <sub>6</sub> → A <sub>11</sub> : chân mã địa chỉ, 6 chân này được sử dụng làm bit địa chỉ cao. -D <sub>0</sub> → D <sub>5</sub> : chân dữ liệu, 6 chân này được sử dụng làm ngõ ra data. Việc sử dụng bit làm địa chỉ cao hay làm ngõ ra data là tùy thuộc vào phiên bản của PT2272. +khi sử dụng vào ngõ vào địa chỉ thì các chân này trở thành các chân trạng thái khởi động và mỗi chân có thể ở mức 0, mức 1 hoặc thả nổi + khi sử dụng làm ngõ ra data, nếu địa chỉ được mã hoá từ dạng sóng nhận về tương thích với địa chỉ thiết lập tại các chân ngõ vào địa chỉ và tương ứng với bit dữ liệu được nhận về là 1 bit thì các chân này sẽ set lên mức 1 (V <sub>cc</sub> ) tức là điện áp từ các chân này là + V <sub>cc</sub> . Ngược lại là - V <sub>cc</sub>
14	16	
15	17	-DIN: các chân dữ liệu vào. Dạng sóng đã được mã hoá được nhận về thì được đưa đến các chân DIN này.
16	18	- Chân dao động số 1.
		- Chân dao động số 2.
17	19	⇒ Giữa hai chân này sẽ được mắc một điện

18	20	trở để xác định tần số căn bản của PT2272.
9	9	- VT: Chân kiểm tra mã nhận về, khi dạng sóng nhận về đúng thì chân VT sẽ ở mức cao.
	10 → 11	- Nguồn dương. - Nguồn âm.. - Không kết nối.

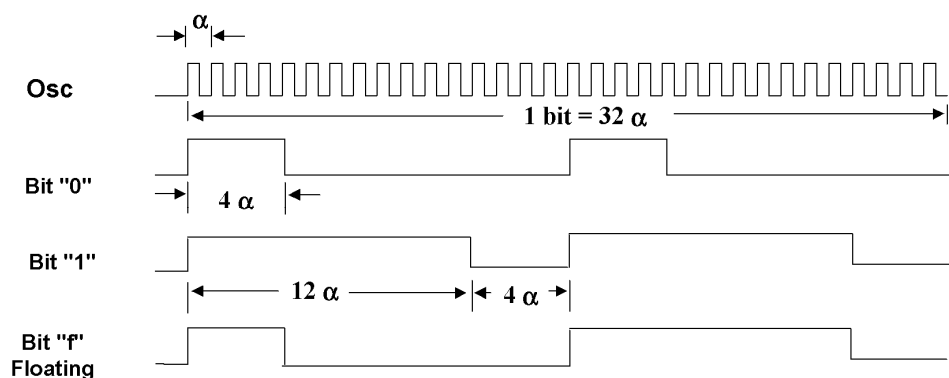
**Bảng 2.1.** Ý nghĩa các chân của PT2272

#### **4. Mô tả chức năng :**

PT 2272 giải mã dạng sóng nhận được đưa vào chân DIN. Dạng sóng được giải mã ở dạng mã Word. Mã Word này gồm có bit dữ liệu, bit địa chỉ, và bit đồng bộ. Các bit địa chỉ được giải mã sẽ đem so sánh với các địa chỉ đã được thiết lập tại các chân ngõ vào địa chỉ. Nếu hai địa chỉ này tương thích với hai mã Word liên tiếp PT2272 sẽ cho các bit dữ liệu tương ứng tại các ngõ ra Data được giải mã thành một bit và ngõ ra VT sẽ ở trạng thái cao.

#### **4.1. Hoạt động của mạch thu RF:**

- **Các mã bit :** mã bit là thành phần cơ bản của dạng sóng đã mã hoá và nó có thể được phân làm hai loại : bit dữ liệu / địa chỉ hoặc bit đồng bộ.
- **Bit địa chỉ / dữ liệu :** mỗi bit AD có thể là bit 0, bit 1, hoặc bit thả nổi nếu nó có các trạng thái tương ứng là trạng thái cao, trạng thái thấp, hoặc trạng thái thả nổi. Mỗi dạng sóng bit như vậy gồm hai chu kỳ xung. Mỗi chu kỳ xung gồm có 16 xung dao động. Xem hình bên dưới.

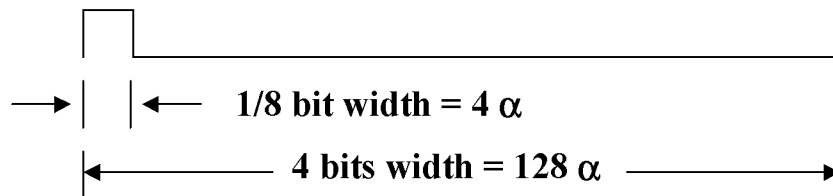


**Hình 2.3.** Dạng sóng bit AD

Trong đó :  $\alpha$  là chu kỳ xung dao động

Bit “ f ” chỉ dành cho bit địa chỉ.

- **Bit đồng bộ** : có độ dài 4 bit và độ rộng xung =  $1/8$  bit AD



**Note : 1 bit =  $32 \alpha$**

**Hình 2.4.** Dạng sóng bit đồng bộ.

- **Các mã Word** : một nhóm các mã bit gọi là mã Word. Một mã Word bao gồm 12 bit AD và theo sau là một bit đồng bộ. 12 bit AD này có thể là bit địa chỉ hoặc là bit dữ liệu tùy theo phiên bản của PT2272. Xem hình bên dưới.

**PT2272:**

A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	SYNC
----	----	----	----	----	----	----	----	----	----	-----	-----	------

**PT2272A-M2/L2:**

A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	D1	D0	SYNC
----	----	----	----	----	----	----	----	----	----	----	----	------

**PT2272-M3/L3:**

A0	A1	A2	A3	A4	A5	A6	A7	A8	D2	D1	D0	SYNC
----	----	----	----	----	----	----	----	----	----	----	----	------

**PT2272-M4/L4:**

A0	A1	A2	A3	A4	A5	A6	A7	D3	D2	D1	D0	SYNC
----	----	----	----	----	----	----	----	----	----	----	----	------

**PT2272-M5/L5:**

A0	A1	A2	A3	A4	A5	A6	D4	D3	D2	D1	D0	SYNC
----	----	----	----	----	----	----	----	----	----	----	----	------

**PT2272-M6/L6:**

A0	A1	A2	A3	A4	A5	D5	D4	D3	D2	D1	D0	SYNC
----	----	----	----	----	----	----	----	----	----	----	----	------

↳

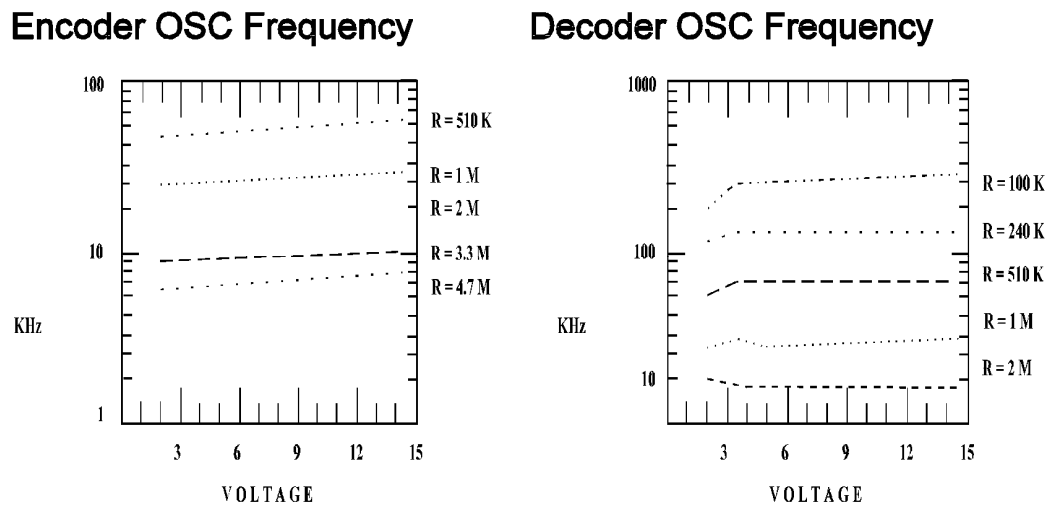
One Complete Code Word

↳

**Hình 2.5.** Mã Word của các phiên bản PT2272

## 4.2. Bộ dao động :

Bộ dao động của PT2272 được xây dựng bằng cách nối một điện trở bên ngoài. Để PT2272 giải mã đúng dạng sóng nhận được thì tần số dao động của PT2272 phải đạt từ 2,5 đến 8 lần tần số phát của PT2262. Để hạn chế méo tốt nhất là cho tần số của PT2272 nằm trong khoảng chính giữa trong bảng bên dưới.



**Hình 2.6.** Tần số dao động của bộ phát và bộ thu.

- Các giá trị điện trở cho bộ dao động :

PT 2262	PT 2272
4.7 MΩ	820 KΩ*
3.3 MΩ	680 KΩ*
1.2 MΩ	200 KΩ**

**Bảng 2.2.** Các giá trị điện trở cho PT2272 và PT2262

Bộ dao động loại này thì giá trị điện trở có thể thay đổi được cho cả PT2272 và PT2262.

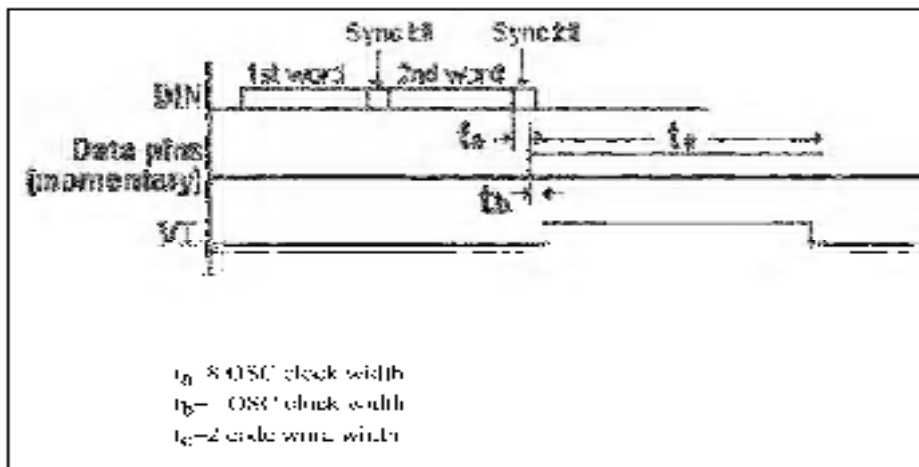
Lưu ý: \* dùng cho mạch khi nguồn cấp cho PT2272 là 5V- 15V

\*\* dùng cho mạch khi nguồn cấp cho PT2272 là 4V- 15V

Có nghĩa là nếu nguồn cấp cho PT2272 dưới 5V thì phải sử dụng giá trị điện trở thấp hơn cho cả hai loại PT2272 và PT2262.

## **5. Tính hiệu lực của sự truyền tín hiệu :**

Khi PT2272 nhận về một mã Word thì đầu tiên nó kiểm tra mã Word này đã đúng chưa. Để làm việc này thì nó phải là một mã Word đúng và bit địa chỉ phải tương thích với địa chỉ được thiết lập tại các chân địa chỉ. Sau khi kiểm tra mã liên tiếp đúng rồi thì PT2272 sẽ điều khiển các chân dữ liệu tương thích với các bit dữ liệu nhận về và làm tăng Vt lên mức cao.



**Hình 2.7.** Giải đồ kiểm tra mã Word

## **6. Dữ liệu ra được chốt lại hay tức thời :**

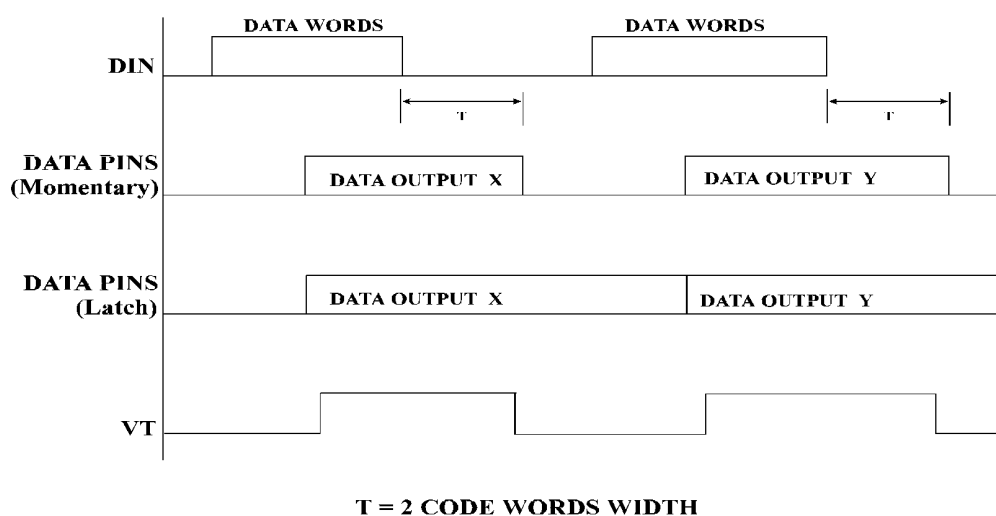
PT2272 sử dụng dữ liệu ngõ ra được chốt lại hay tức thời là tùy thuộc vào phiên bản của PT2272.

### **6.1. Loại chốt (PT2272- LX) :**

Loại này sẽ kích hoạt ngõ ra suốt thời gian PT2262 phát dữ liệu và dữ liệu này được giữ lại trong bộ nhớ cho đến khi dữ liệu khác được đưa vào.

### **6.2. Loại tức thời ( PT2272- MX ):**

Loại này cũng sẽ kích hoạt ngõ ra suốt thời gian PT2262 phát dữ liệu , nhưng dữ liệu này không được giữ lại trong bộ nhớ sau khi sự truyền phát kết thúc . Xem hình bên dưới.



**Hình 2.8.** Giải đồ ngõ ra Data của PT2272.

### **7.Trình tự hoạt động:**

#### **7.1. Giải mã không cùng với chân xuất dữ liệu:**

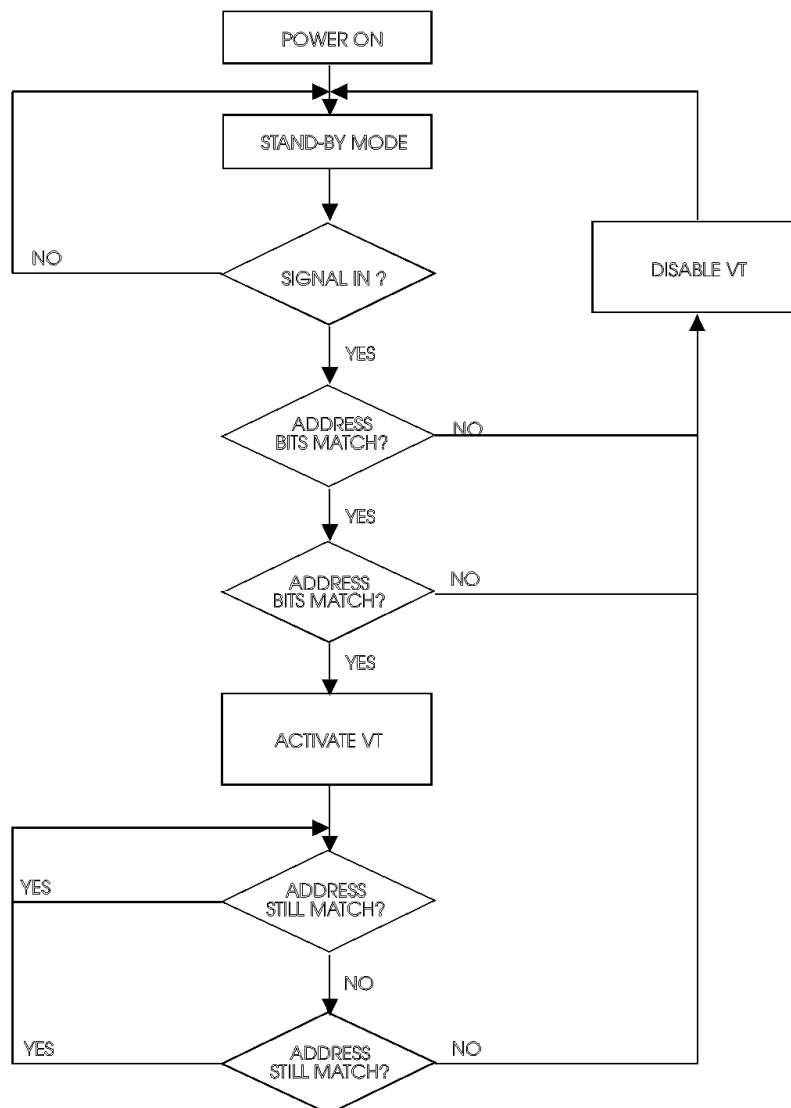
**Bước 1** : Khi mở nguồn , PT2272 sẽ ở chế độ chờ ( Stand – By ).

**Bước 2** : sau đó nó sẽ tìm tín hiệu ,nếu không có tín hiệu nhận về thì nó vẫn ở chế độ chờ. Ngược lại , bit địa chỉ nhận về sẽ đem so sánh với địa chỉ tại các chân địa chỉ.

**Bước 3** : Chân VT chỉ lên mức cao khi nào có hai khung liên tục tương thích với các bit địa chỉ.

**Bước 4**: Sau đó các bit địa chỉ được kiểm tra lần nữa .Nếu có hai bit liên tục bị sai thì sẽ không cho phép chân VT và sẽ trở về chế độ chờ. Ngược lại , thì các bit địa chỉ sẽ được kiểm tra tiếp tục





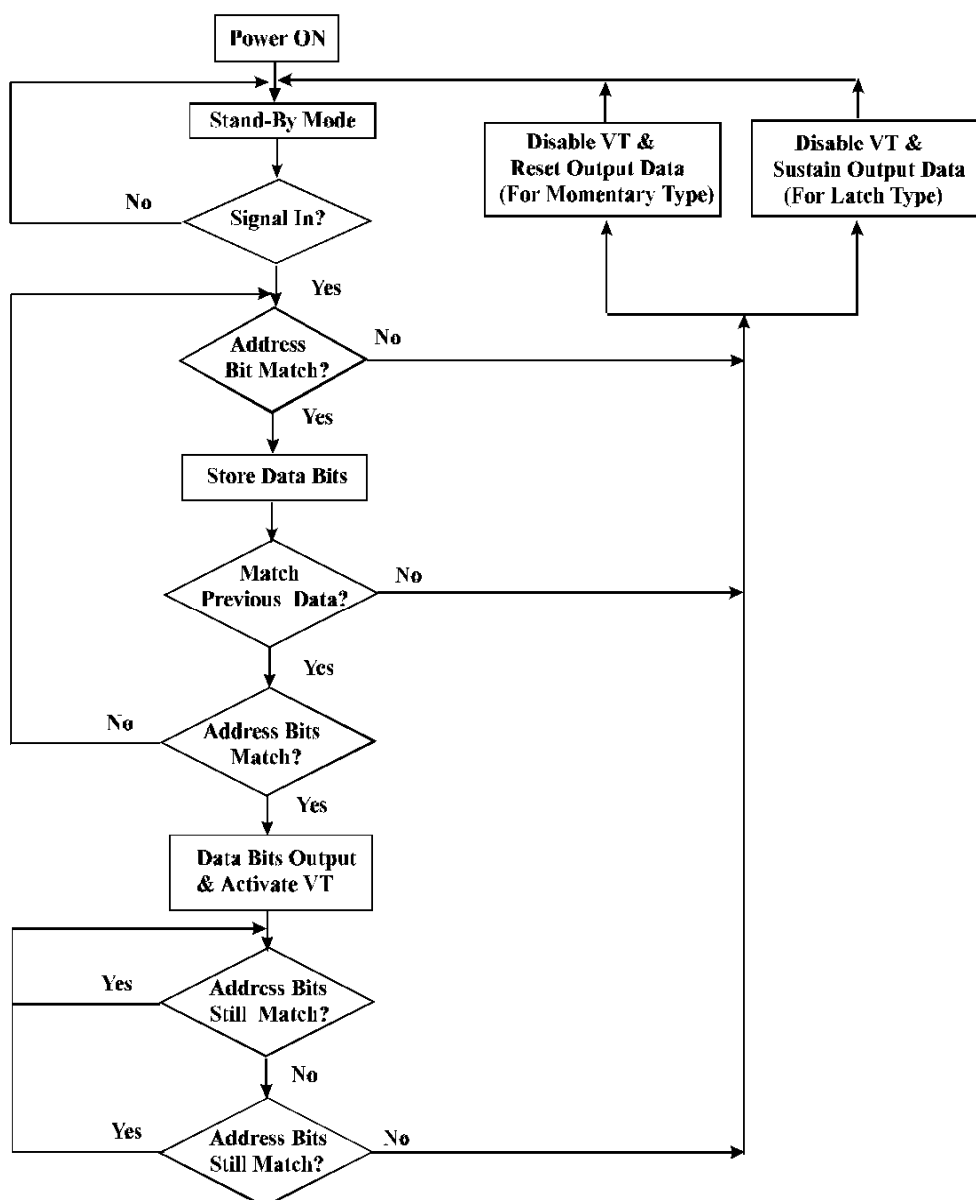
**Hình 2.9.** Trình tự giải mã không cùng với chân xuất dữ liệu.

## 7.2. Giải mã không cùng với chân xuất dữ liệu:

**Bước 1** : Khi mở nguồn , PT2272 sẽ ở chế độ chờ ( Stand – By ).

**Bước 2** : sau đó nó sẽ tìm tín hiệu ,nếu không có tín hiệu nhận về thì nó vẫn ở chế độ chờ. Ngược lại , bit địa chỉ nhận về sẽ đem so sánh với địa chỉ tại các chân địa chỉ.

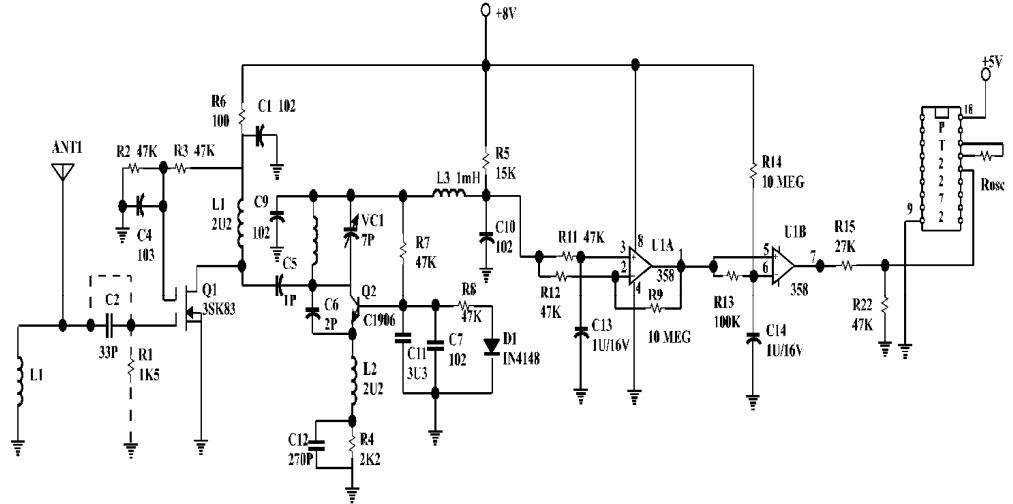
**Bước 3** : Khi nào các bit địa chỉ trong một khung tương thích với địa chỉ của các chân địa chỉ thì bit dữ liệu được lưu trữ trong bộ nhớ .Cũng vậy , khi IC tìm thấy được hai bit liên tục và các dữ liệu giống nhau có cùng bit địa chỉ thì ngõ ra dữ liệu được cho phép và cho phép chân VT. Đối với loại tức thời , dữ liệu ra bị reset trong khi đó , loại chốt thì được giữ lại.



**Hình 2.10.** Trình tự giải mã cùng với chân xuất Data.

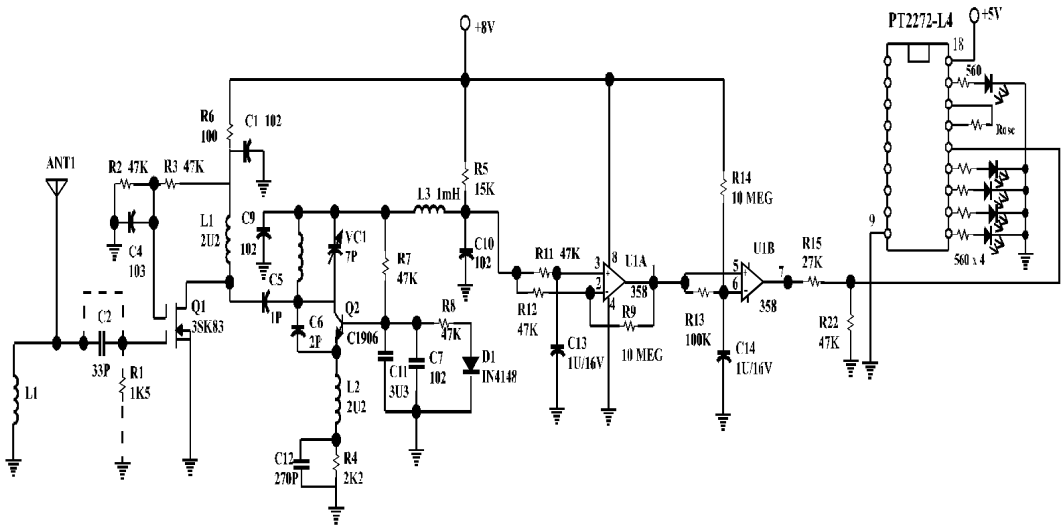
**8. Các mạch ứng dụng :**

**8.1. Ứng dụng cho PT2272 loại không có Data :**



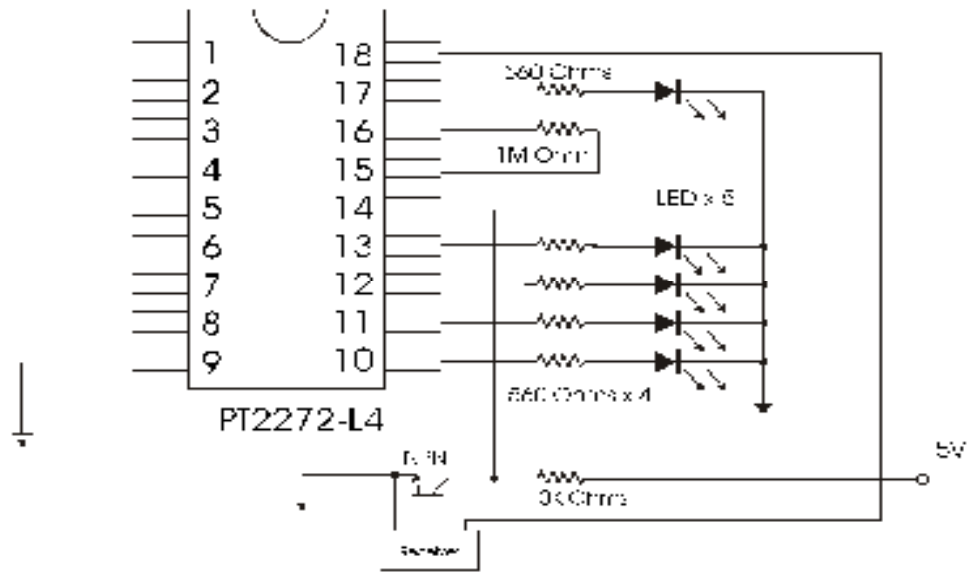
Hình 2.11. Mạch thu sóng RF không có Data.

**8.2. Ứng dụng cho PT2272 loại có 4 Data :**



Hình 2.12. Mạch thu sóng RF có 4 Data.

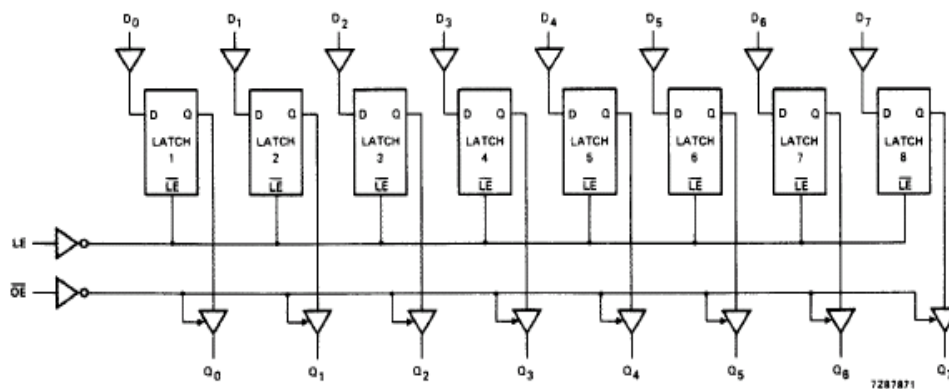
**8.3. Mạch hồng ngoại cho PT2272 loại 4 Data:**



**Hình 2.13.** Mạch hồng ngoại cho PT2272 loại 4 Data

**III. GIỚI THIỆU IC 74HC573 :**

**1. Sơ đồ khối :**



**Hình 3.1.** Sơ đồ khối của 74HC573

## 2. Sơ đồ chân :



**Hình 3.2.** Sơ đồ chân của 74HC573

## 3. Ý nghĩa của các chân :

- Chân 1 :  $\overline{OE}$  . Ngõ vào cho phép chốt ngõ ra 3 trạng thái ( tích cực mức thấp)..
- Chân 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 :  $D_0 \rightarrow D_7$  . Các chân dữ liệu vào .
- Chân 10 : GND . Chân nối đất
- Chân 11 : LE . Ngõ vào cho phép chốt .( tích cực mức cao ) .
- Chân 12, 13 , 14 , 15 , 16 , 17 , 18 , 19 :  $Q_0 \rightarrow Q_7$  . Các ngõ ra chốt 3 trạng thái.
- Chân 20 : chân nối nguồn +5V.

## 4. Bảng trạng thái :

Chế độ hoạt động.	INPUT			Trạng thái 8 bộ chốt	OUTPUTS
	$\overline{OE}$	LE	$D_n$		$Q_0 \rightarrow Q_7$
Cho phép và đọc thanh ghi.	L	H	L	L	L
	L	H	H	H	H
Chốt và đọc thanh ghi.	L	L	l	L	L
	L	L	h	H	H
Chốt thanh ghi và không cho phép ngõ ra.	H	L	l	L	Z
	H	L	h	H	Z

**Bảng 3.1.** Trạng thái các chân của 74HC573

Trong đó : H là mức điện áp cao.

L là mức điện áp thấp.

h là mức điện áp cao trong khoảng thời gian để chân LE chuyển từ mức cao sang mức thấp.

l là mức điện áp thấp trong khoảng thời gian để chân LE chuyển từ mức cao sang mức thấp.

Z là trở kháng cao khi ngõ ra ở trạng thái OFF.

### **5. Nguyên tắc hoạt động :**

- Bên trong 74HC573 gồm 8 DFlip – Flop , có các ngõ vào và ngõ ra không đảo nằm đối diện nhau thuận tiện cho việc giao tiếp với vi xử lý .

Khi chân LE ở mức cao thì Data tại các ngõ vào  $D_n$  được đưa đến bộ chốt Ngõ ra Q của bộ chốt sẽ thay đổi trạng thái tương ứng với sự thay đổi trạng thái của các ngõ vào  $D_n$  .

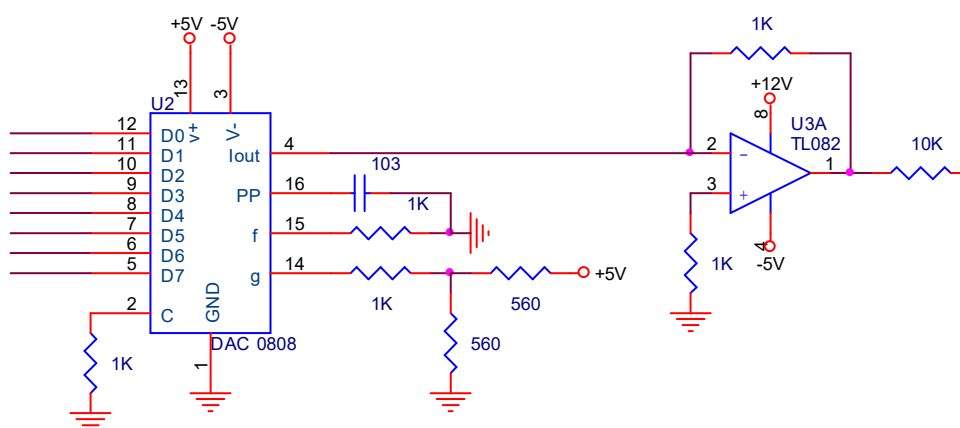
- Khi chân LE xuống mức thấp thì bộ chốt sẽ lưu giữ dữ liệu lại.

- Khi chân OE ở mức thấp , dữ liệu nằm trong 8 bộ chốt bên trong sẽ được đưa đến ngõ ra Q . Khi OE ở mức cao thì các ngõ ra này có trở kháng cao và ở trạng thái OFF.

Hoạt động của chân LE sẽ không ảnh hưởng đến trạng thái của các bộ chốt.

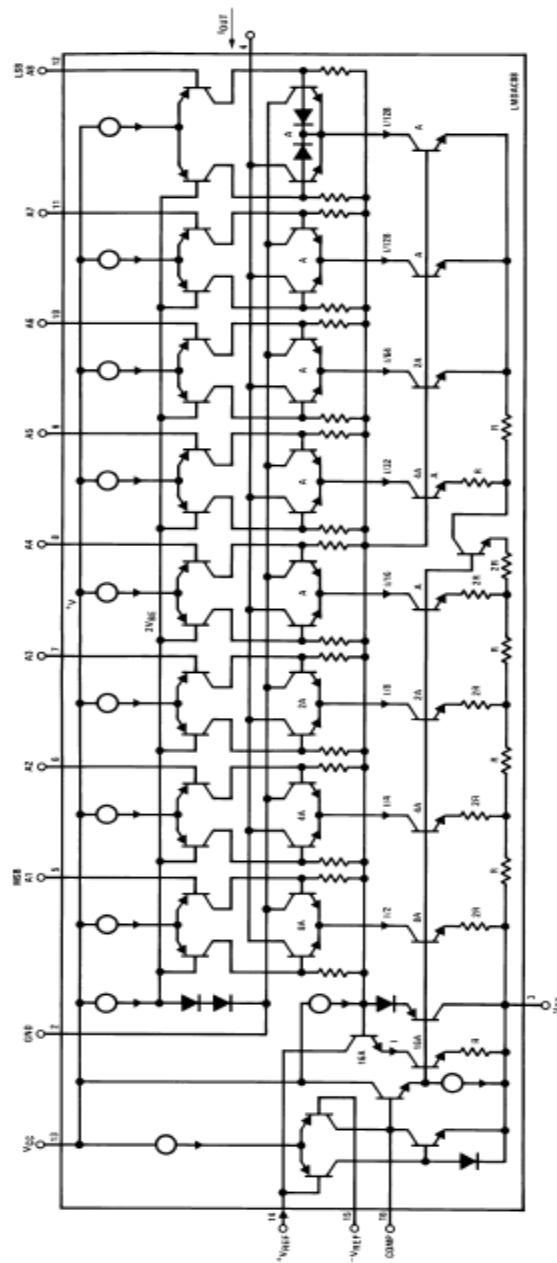
## **IV.GIỚI THIỆU DAC0808 :**

### **1.Sơ đồ chân và kết nối của DAC0808 :**



**Hình 3.1.**Sơ đồ mắc DAC0808.

**2. Sơ đồ kết nối bên trong DAC0808 :**



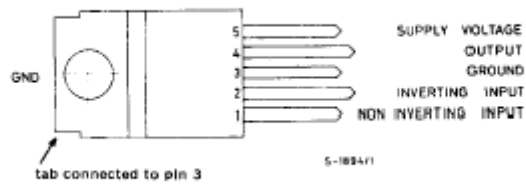
0000607-4

FIGURE 2. Equivalent Circuit of the DAC0808 Series (Note 6)

**Hình 3.2.** Sơ đồ kết nối bên trong của DAC0808.

**V.GIỚI THIỆU TDA2003 :**

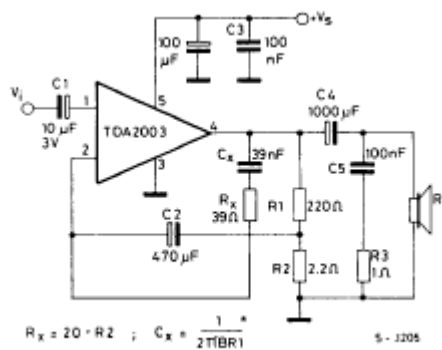
**1.Sơ đồ chân,hình dạng và kết nối của TDA2003 :**



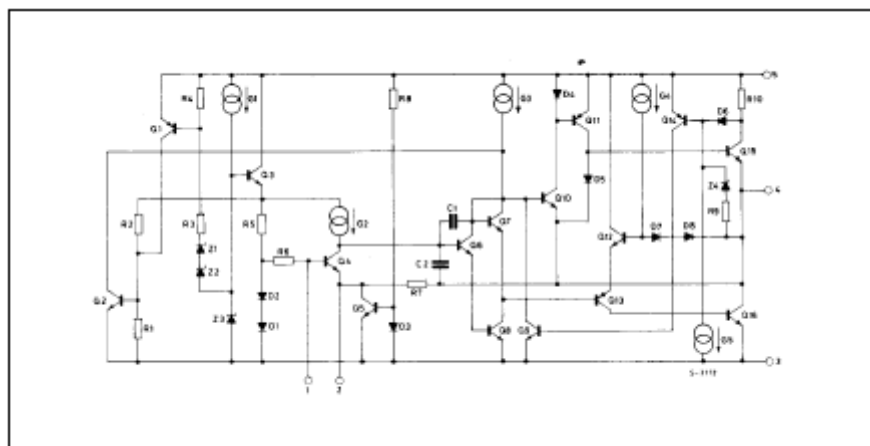
PENTAWATT

ORDERING NUMBERS : TDA 2003H  
TDA 2003V

**Hình 4.1.Hình dạng thực tế của TDA2003.**



**2. Sơ đồ kết nối bên trong của TDA2003 :**



**Hình 4.2.Sơ đồ kết nối bên trong của TDA2003.**



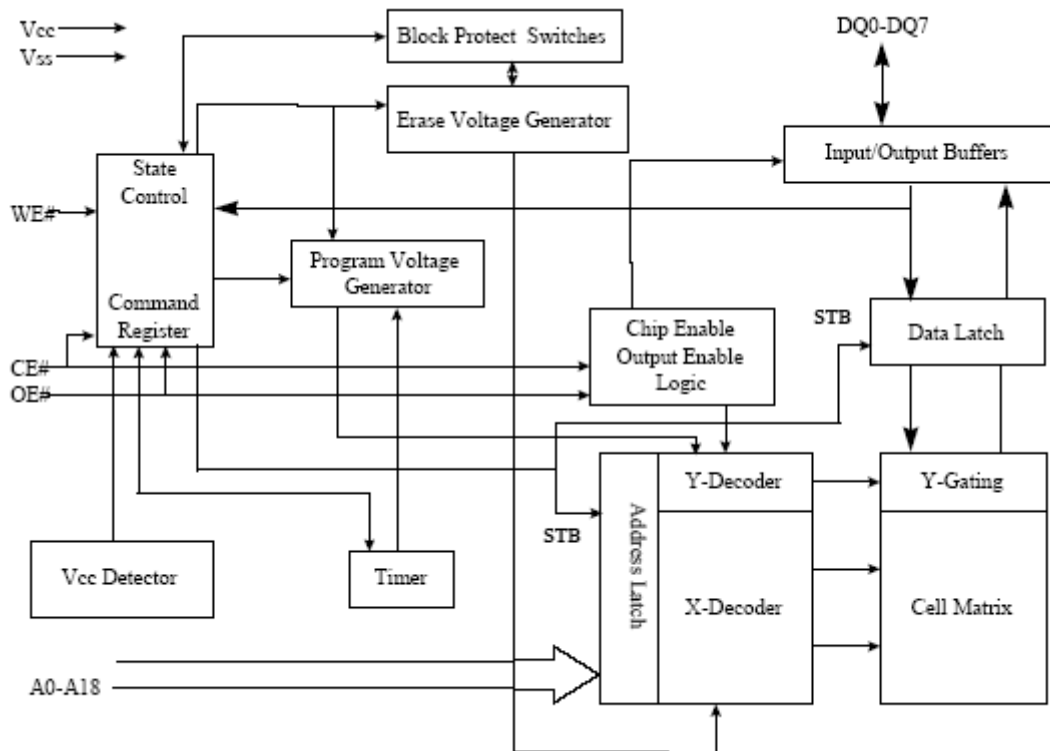
**VI. GIỚI THIỆU EN29LV040 :**

**1. Sơ đồ chân :**



Hình 5.1. Sơ đồ chân EN29LV040

**2. Sơ đồ khối cấu trúc bên trong EN29LV040 :**



# **PHẦN B :**

## **THIẾT KẾ PHẦN CỨNG**

## **A.TỔNG QUAN ĐỀ TÀI :**

\* Mô hình bãi xe gồm có :

- + Sức chứa 5 xe .
- + Hiện thị số lượng xe có trong bãi .
- + Kiểm soát thời gian ra và vào bãi để tính tiền nhưng thanh toán tiền thông qua trạm điều hành .
- + Quản lý về mã code để kiểm soát xe .
- + Lời chào và thông báo bằng âm thanh ở cửa vào và ra của bãi cho chủ xe.

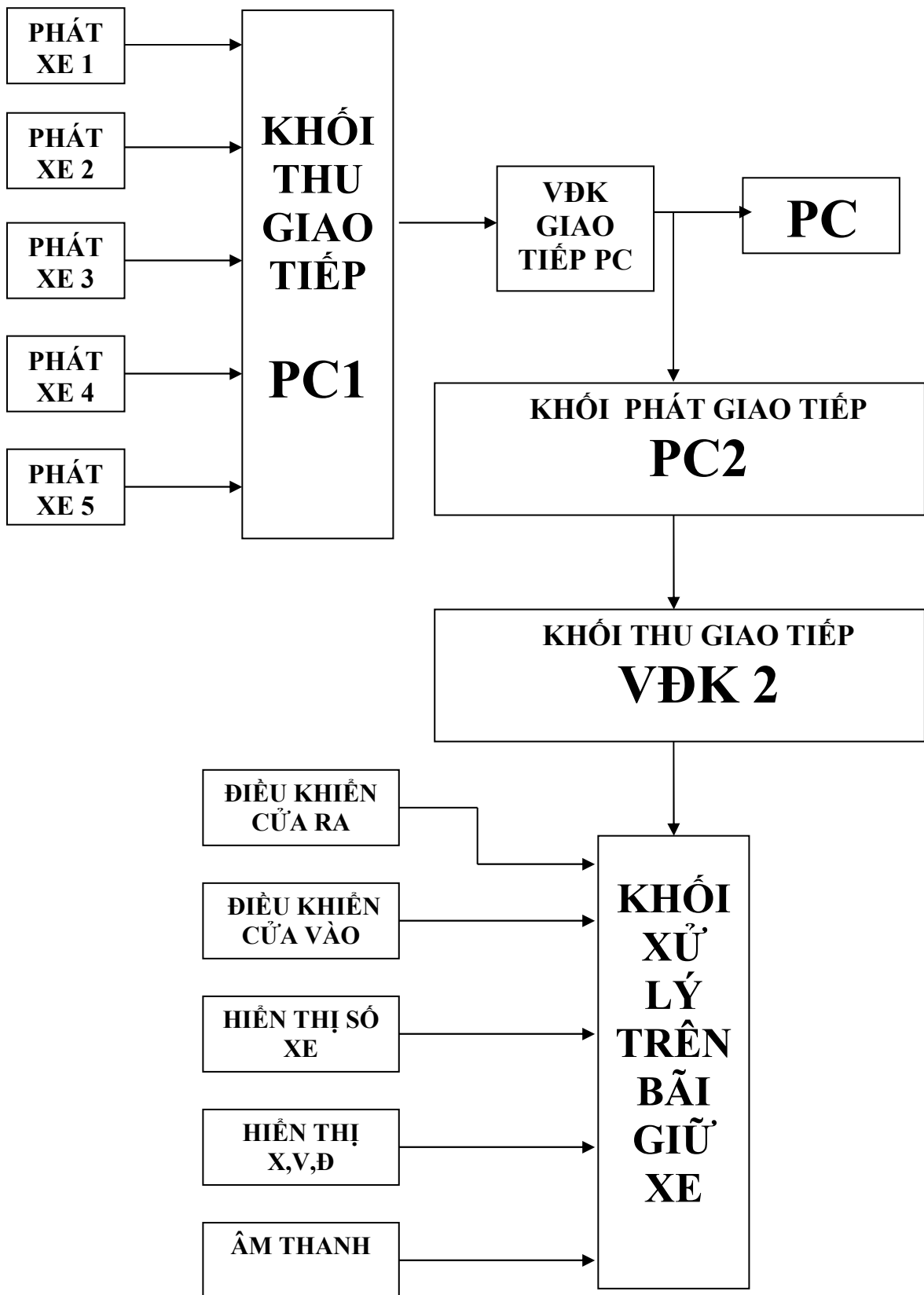
## **I.QUY TRÌNH RA VÀO BÃI XE :**

- Chủ xe muốn đưa xe vào bãi phải liên hệ trạm điều hành, trạm điều hành sẽ cung cấp cho chủ xe một mã code(dành cho xe) và mật mã (dành cho chủ xe).Việc này chỉ diễn ra một lần đối với những xe gửi xe thường xuyên(sau 30 ngày lần gửi cuối thì phần mềm sẽ tự động xóa mã code nếu chủ xe không đưa xe vào bãi một lần nào).
- Khi lấy xe ra chủ xe liên hệ với trạm điều hành để cung cấp mật mã cho trạm điều hành và thanh toán tiền.Sau đó trạm điều hành sẽ mở mã để chủ xe được phép đưa xe ra ngoài .

## **II.CƠ CẤU KỸ THUẬT :**

- Khi xe đến cửa vào bãi cảm biến sẽ nhận diện xe đồng thời kiểm tra mã code của xe . Nếu mã code của xe đã được mở thì VĐK sẽ điều khiển mở cửa đồng thời tác động led hiển thị số lượng xe trong bãi tăng lên . VĐK sẽ kích hoạt bộ đếm giờ tương ứng với mã code của xe vừa vào .
- Khi xe đến cửa ra bãi cảm biến sẽ nhận diện xe đồng thời kiểm tra mã code của xe. Nếu mã code của xe đã mở ( đã thanh toán tiền ),khi đó VĐK sẽ điều khiển cửa ra mở , đồng thời tác động led hiển thị số lượng xe trong bãi giảm xuống . Khi đó mã code của xe vừa ra sẽ mất . Nếu chủ xe chưa thanh toán tiền hoặc chưa hợp lệ về mã thì VĐK cho phát tính hiệu cảnh báo bằng âm thanh và quang báo đồng thời phát tính hiệu báo động về trạm điều hành .
- Phía trước cửa vào bãi có 3 led xanh , vàng , đỏ và 1 led 7 đoạn.
  - + Xanh : bãi xe còn trên 2 chỗ trống .
  - + Vàng : bãi xe chỉ còn 1 chỗ trống .
  - + Đỏ : bãi xe không còn chỗ trống .
  - + Led 7 đoạn : hiển thị số xe có trong bãi.
- VĐK giao tiếp với máy tính được hiển thị bằng phần mềm Visual Basic .

## B.SƠ ĐỒ KHỐI TỔNG QUAN ĐỀ TÀI :

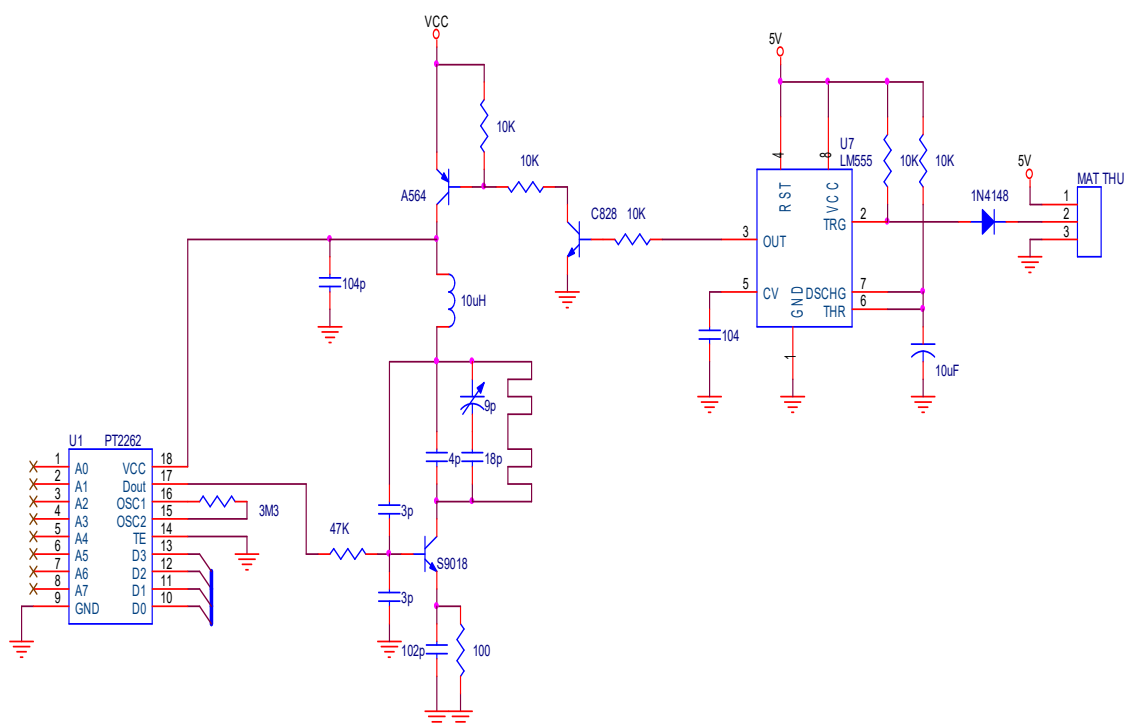


## I. NHIỆM VỤ CỦA TỪNG KHỐI :

- 1. Khối phát xe 1, xe 2, ...:** có nhiệm vụ phát tín hiệu có mã code vừa được cấp ở trạm điều hành đến khối thu giao tiếp với máy tính.
- 2. Khối thu giao tiếp PC1:** sẽ thu tín hiệu vô tuyến mang mã code từ khối phát bao gồm khối phát của xe và khối nhập mật mã. Các khối này được set sẵn mã địa chỉ là 00H.
- 3. Khối giao tiếp máy tính:** khối này nhận tín hiệu dữ liệu từ khối thu giao tiếp 1 sau đó truyền về máy tính. Sau đó chờ lệnh của PC để đưa lệnh này tới khối phát giao tiếp 2.
- 4. Khối phát giao tiếp PC2:** nhận dữ liệu từ khối giao tiếp và truyền dữ liệu này tới khối thu giao tiếp VĐK 2. Cả hai khối này được set chung mã địa chỉ là 01H.
- 5. Khối thu giao tiếp PC2:** nhận dữ liệu từ khối phát giao tiếp PC 2 và truyền dữ liệu này tới khối xử lý trên mô hình VĐK 2.
- 6. Khối xử lý VĐK2 :** nhận dữ liệu từ khối thu giao tiếp VĐK 2, xử lý và điều khiển các thiết bị trên bãi giữ xe.

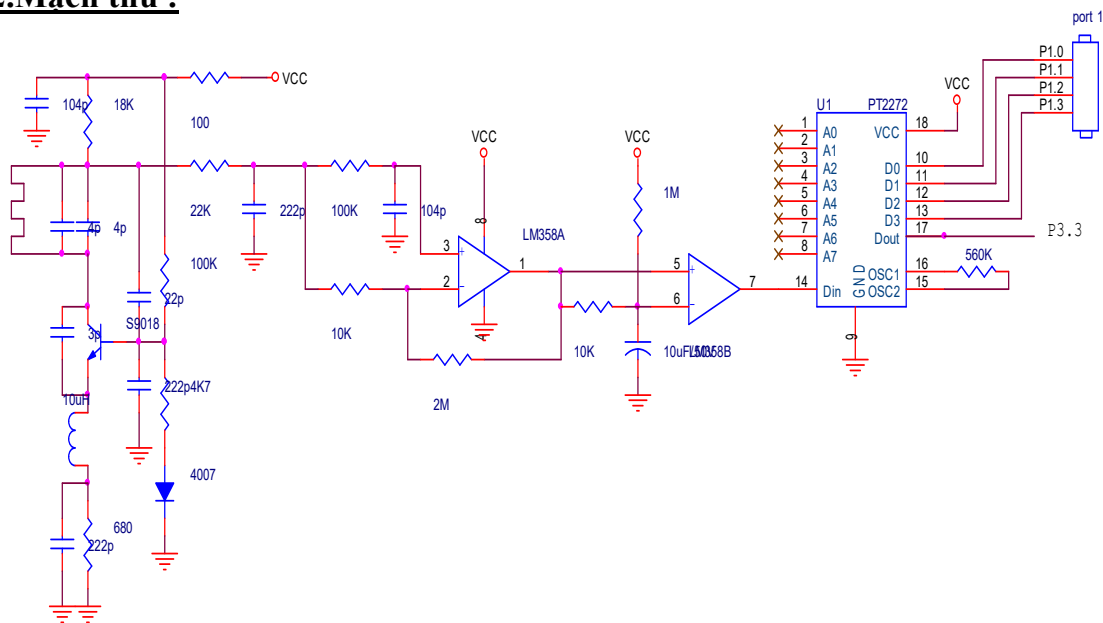
## II. CÁC SƠ ĐỒ NGUYÊN LÝ CỤ THỂ :

### 1. Mạch phát :



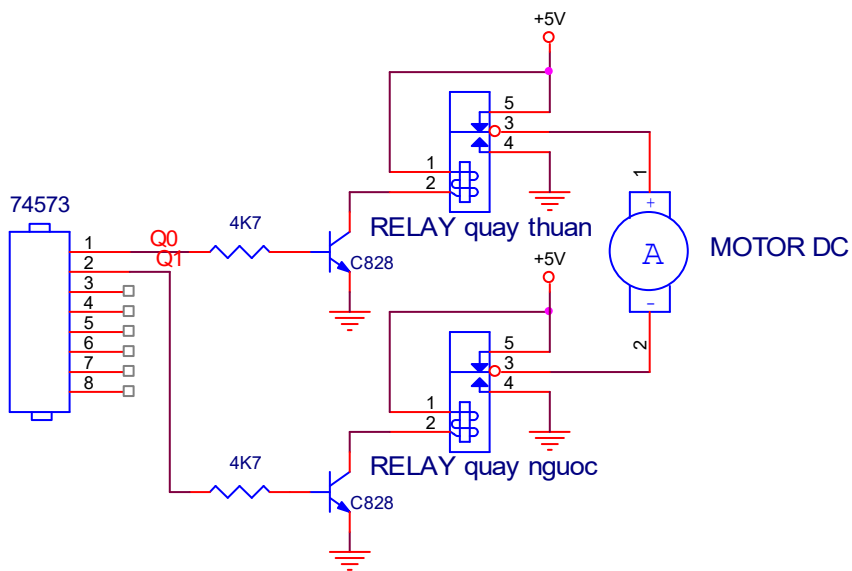
**Hình 2.1.** Mạch phát sóng RF

**2.Mạch thu :**



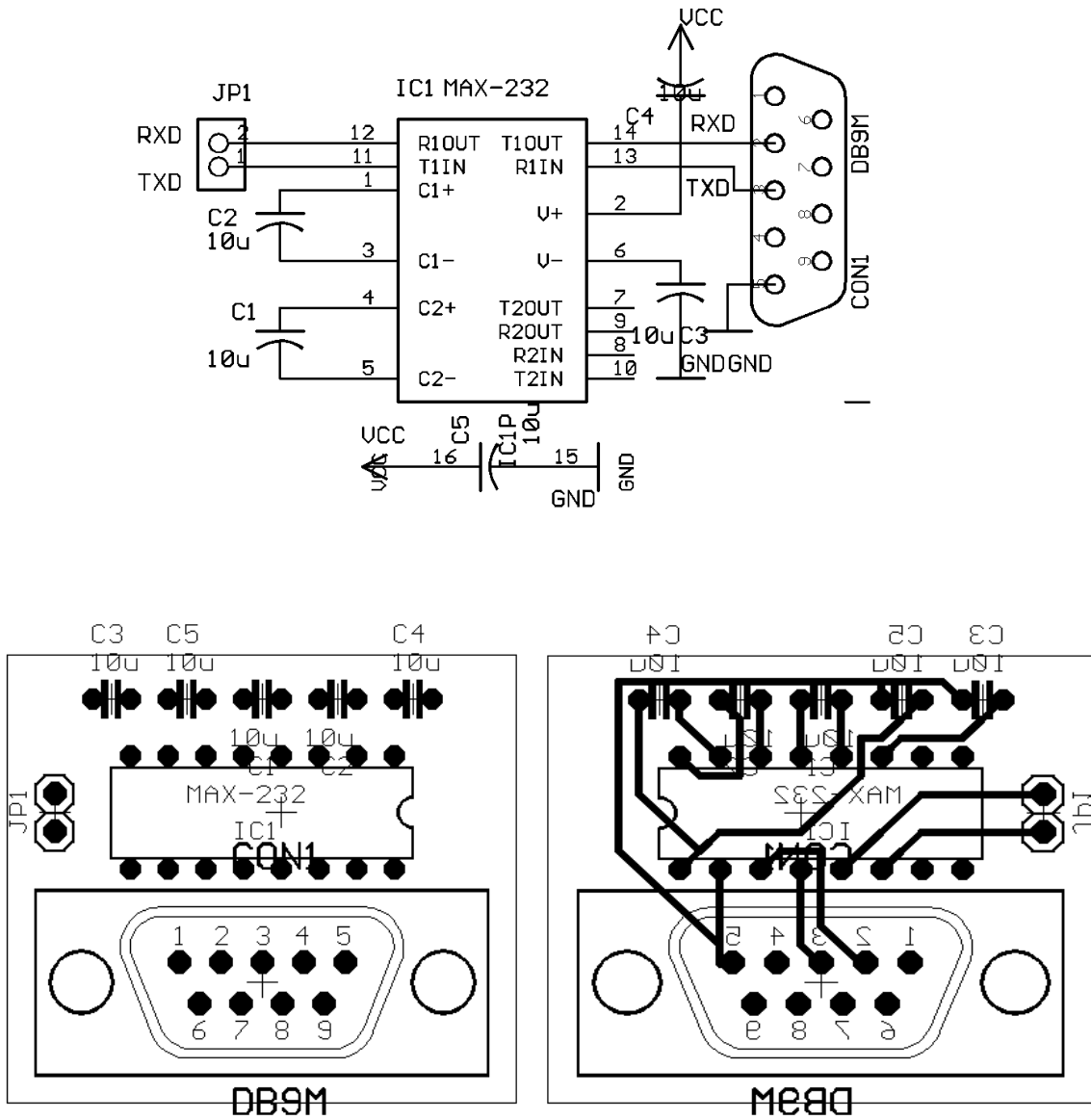
**Hình 2.2.** Mạch thu sóng RF

**3.Mạch điều khiển mortor :**

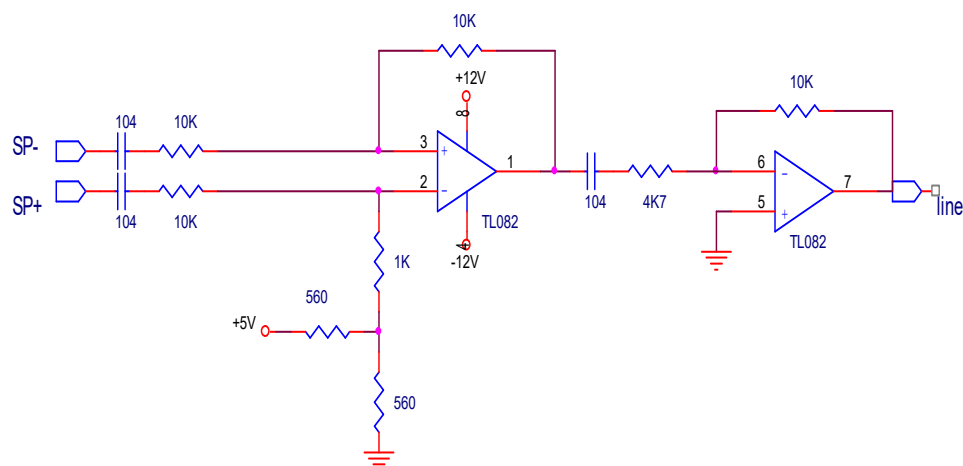


**Hình 2.3.** Mạch điều khiển mortor

**4. Sơ đồ card giao tiếp máy tính :**



**Hình 2.4.** Sơ Đồ Card Giao Tiếp Và Sơ Đồ Mạch In

**5. Mạch khuếch đại âm thanh :****Hình 2.5.** Mạch khuếch đại âm thanh

- Tín hiệu âm thanh trước khi lên đường dây đến máy bị gọi ta cho qua mạch khuếch đại.

- Chọn độ lợi của mạch khuếch đại 1 là  $A_{v3} = 1$  , mạch này chỉ mục đích đệm ngõ vào.

$$\text{Ta có : } A_{v3} = \frac{R_{10}}{R_{11}} = 1$$

- Chọn  $R_{10} = 10k, R_{11} = 10k$  .

- Chọn độ lợi của mạch khuếch đại 2 là  $A_{v0} = 2$

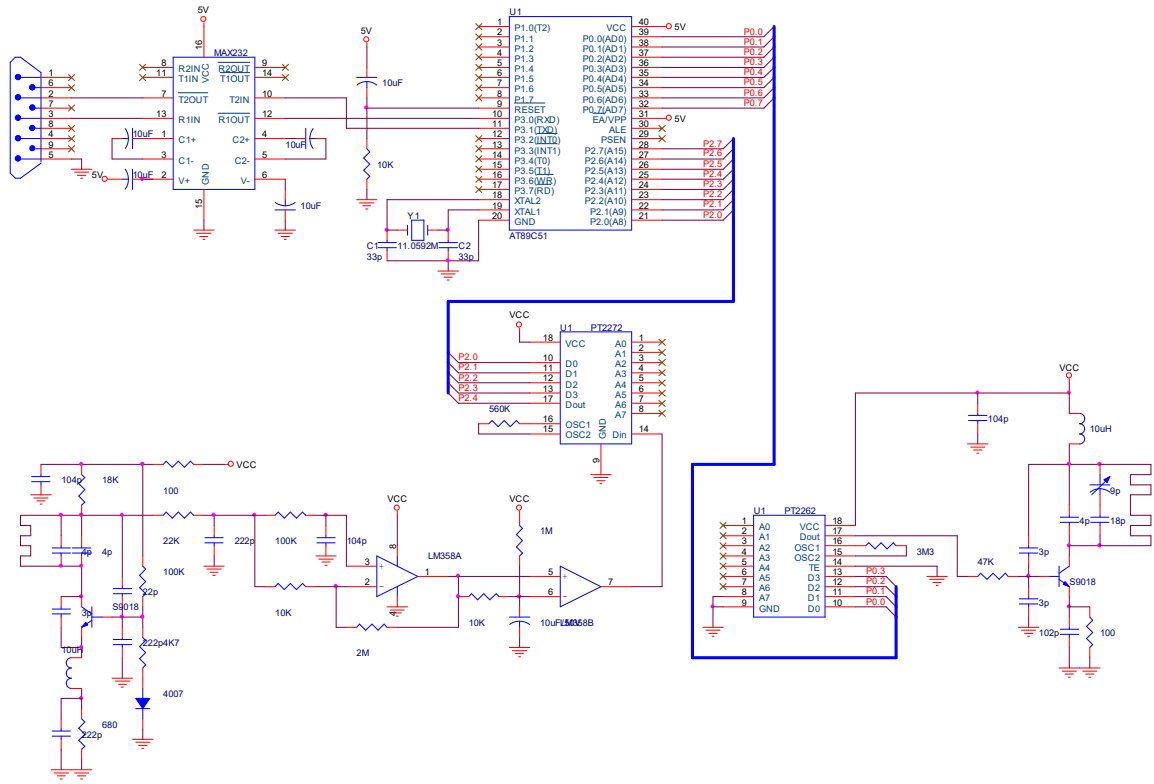
$$\text{Ta có : } A_{v4} = \frac{R_{12}}{R_{13}} = 1$$

- Chọn  $R_{12} = 4.7k, R_{13} = 10k$

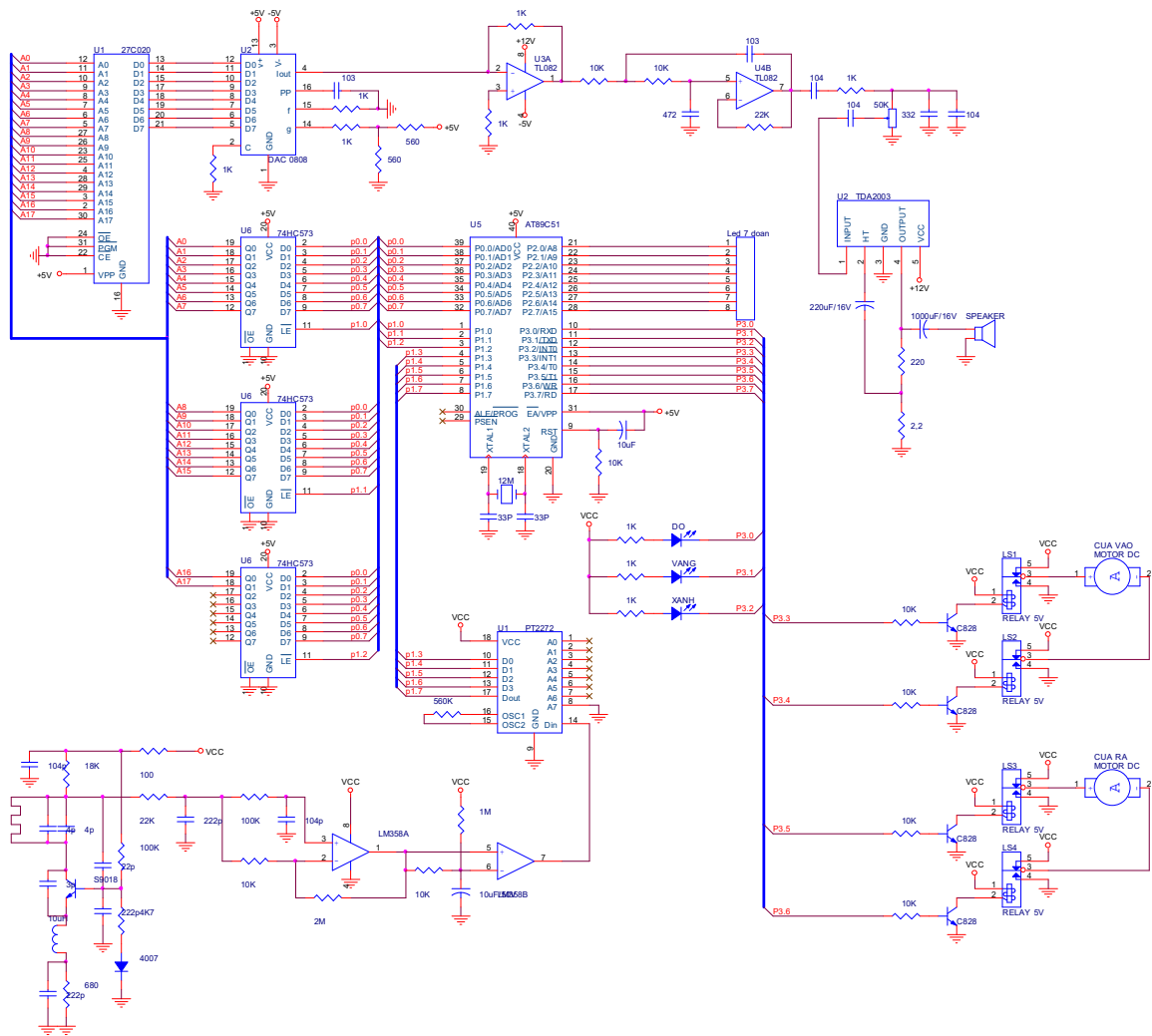
\* Từ sơ đồ khối tổng quan của đề tài sinh viên thực hiện đã đưa ra được các sơ đồ nguyên lý cho các khối.



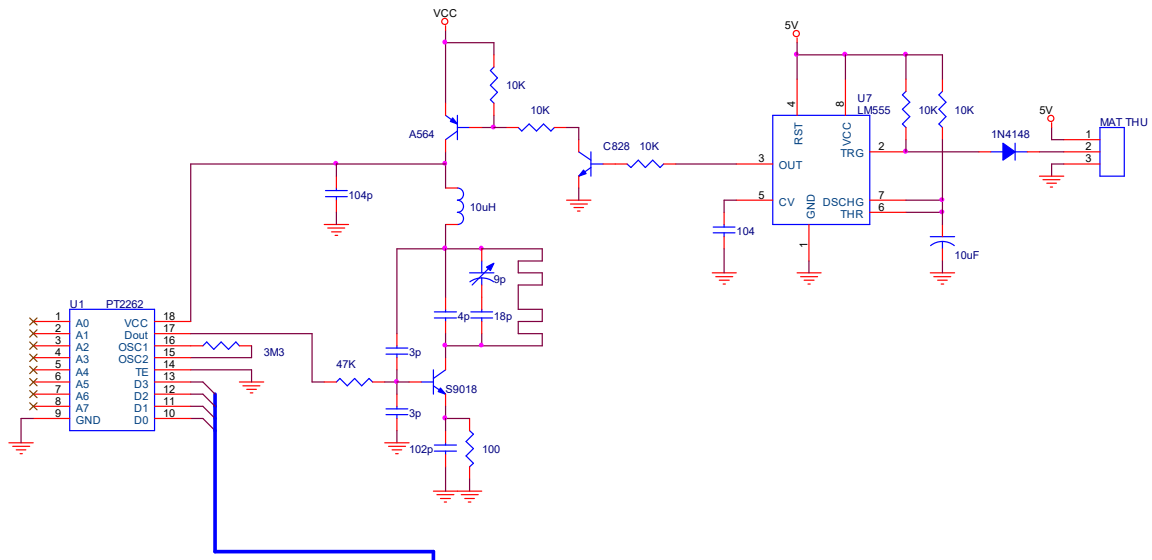
**C.SƠ ĐỒ NGUYÊN LÝ KHỐI GIAO TIẾP MÁY TÍNH :**



### D. SƠ ĐỒ NGUYÊN LÝ KHỞI XỬ LÝ TRÊN BÃI GIỮ XE :



**E.SƠ ĐỒ NGUYÊN LÝ KHỐI NHẬP MẬT MÃ :**



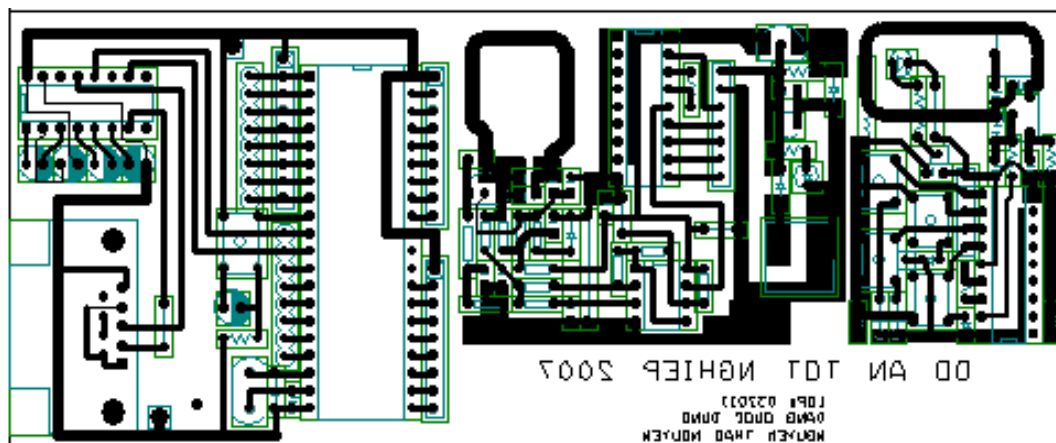
XE-PHIM	D3	D2	D1	D0
XE1	0	0	0	0
XE2	0	0	0	1
XE3	0	0	1	0
XE4	0	0	1	1
XE5	0	1	0	0
PHIM 0	0	1	0	1
PHIM 1	0	1	1	0
PHIM 2	0	1	1	1
PHIM 3	1	0	0	0
PHIM 4	1	0	0	1
PHIM 5	1	0	1	0
PHIM 6	1	0	1	1
PHIM 7	1	1	0	0
PHIM 8	1	1	0	1
PHIM 9	1	1	1	0
ENTER	1	1	1	1

# PHẦN C :

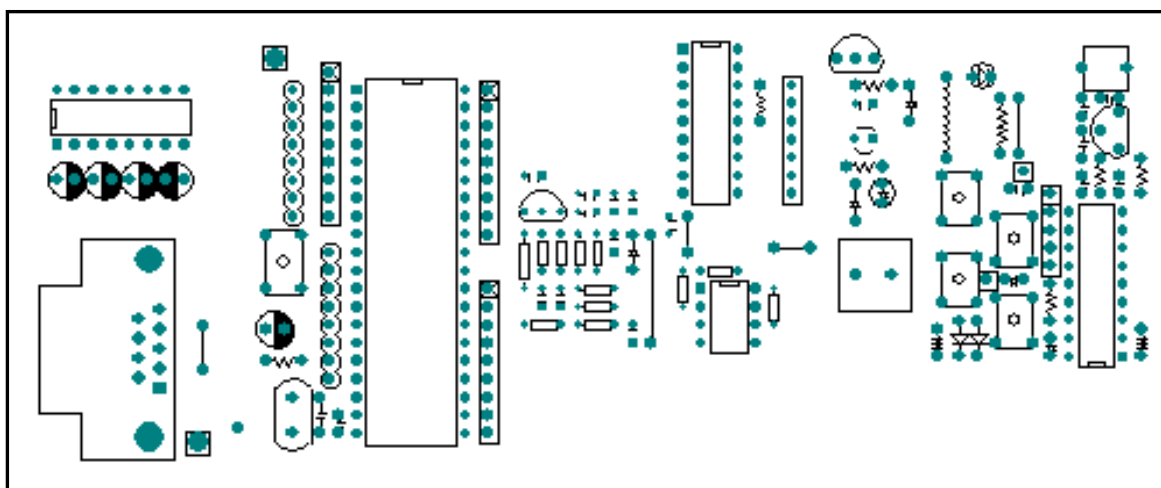
## THI CÔNG

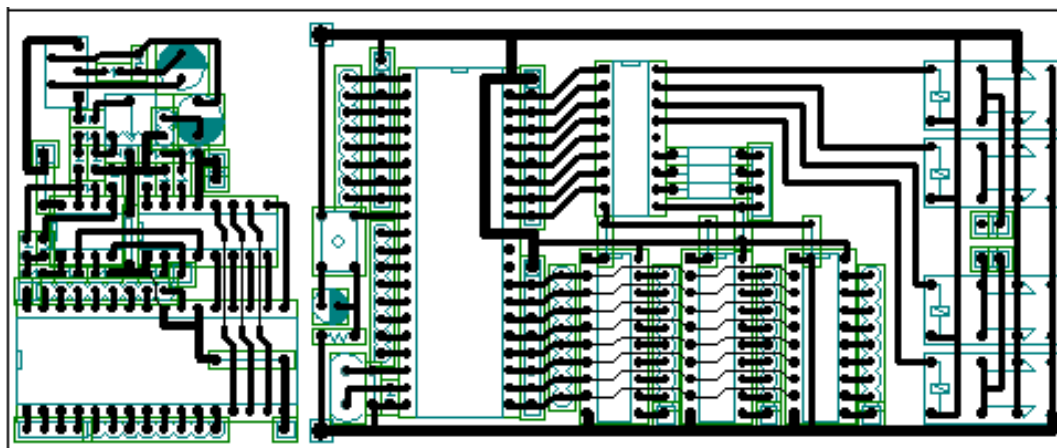
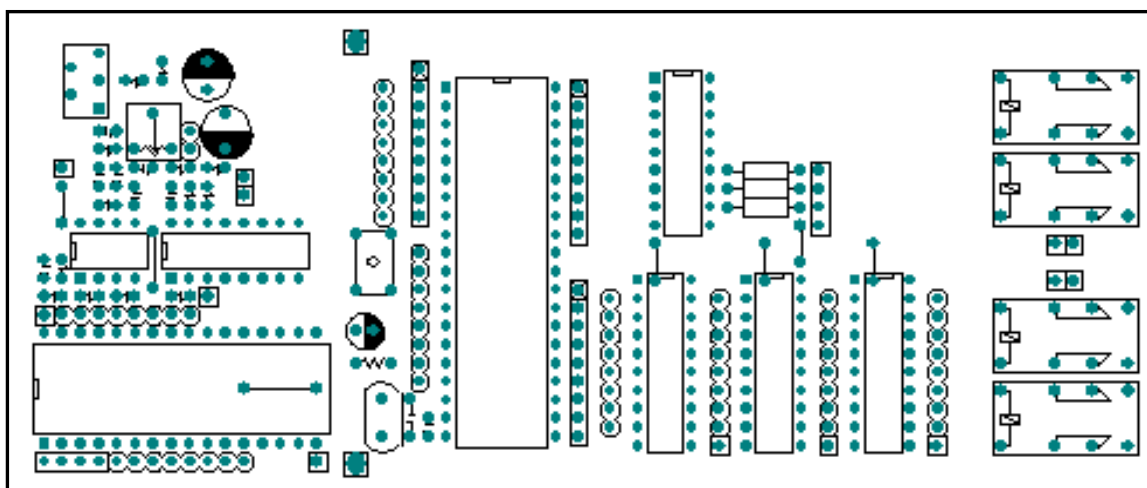
**A. GIAO TIẾP**

**I. SƠ ĐỒ MẠCH IN :**

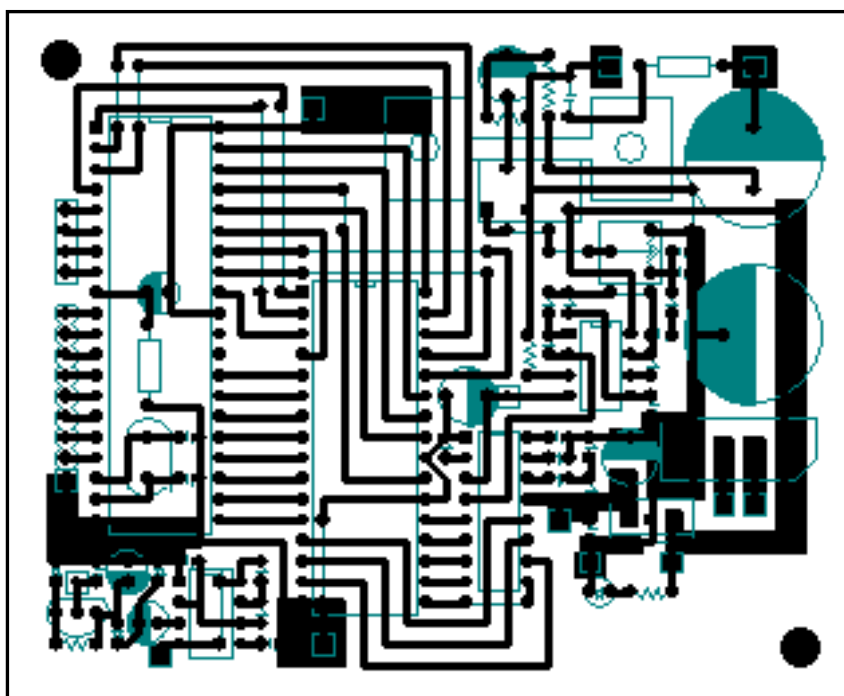


**II. SƠ ĐỒ BỐ TRÍ LINH KIỆN :**

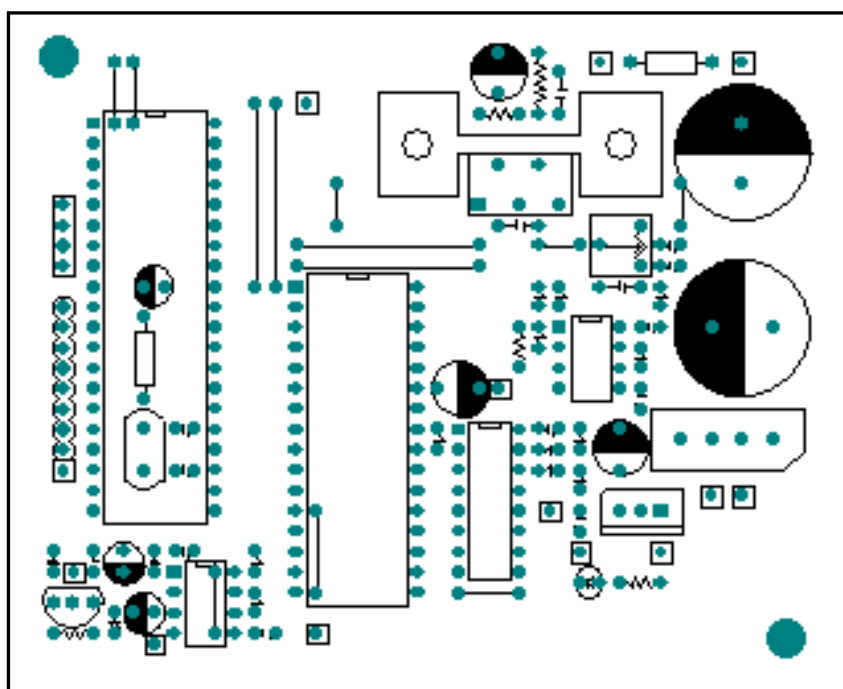


**B. MÔ HÌNH****I. SƠ ĐỒ MẠCH IN :****II. SƠ ĐỒ BỐ TRÍ LINH KIỆN :**

**C. ÂM THANH**  
**I. SƠ ĐỒ MẠCH IN :**



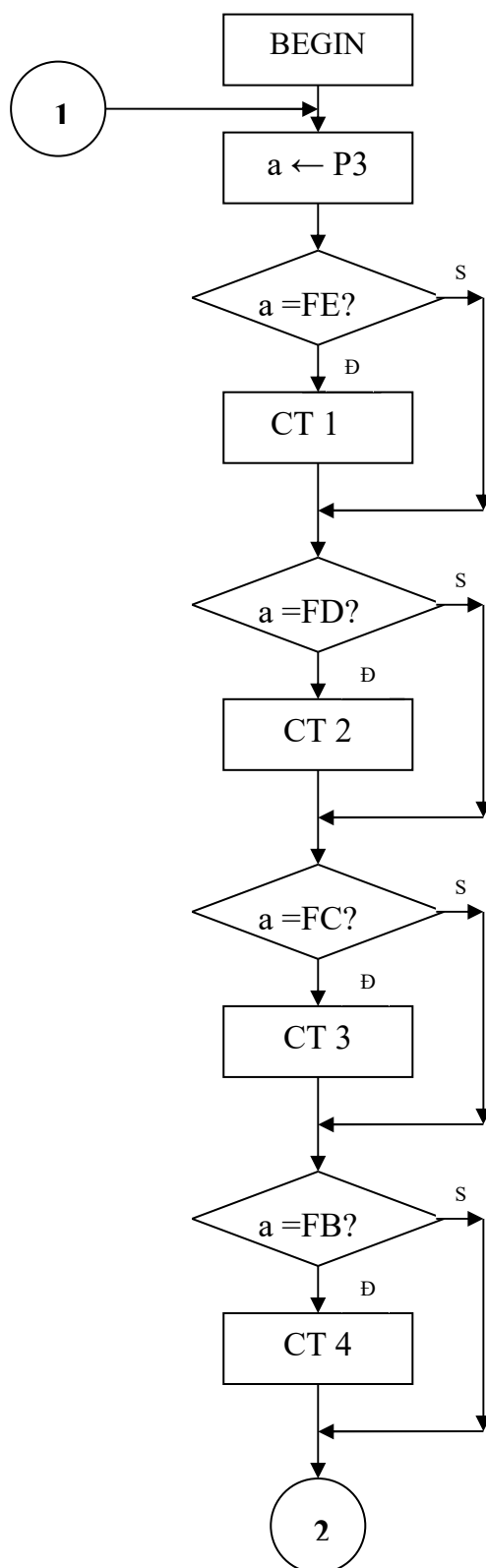
**II. SƠ ĐỒ BỐ TRÍ LINH KIỆN :**

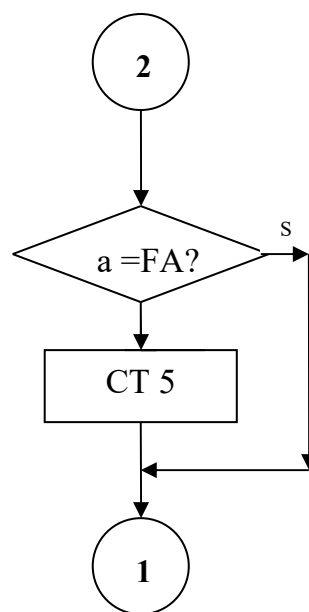


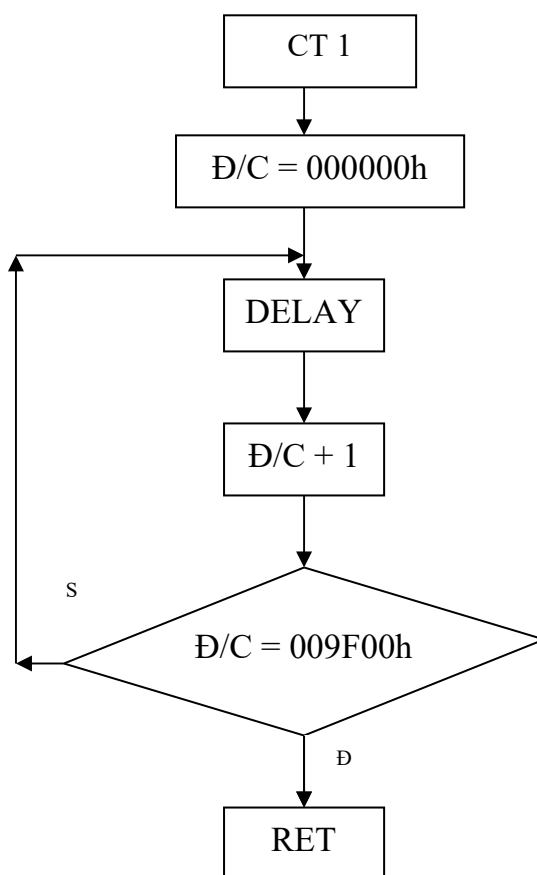
# PHẦN D :

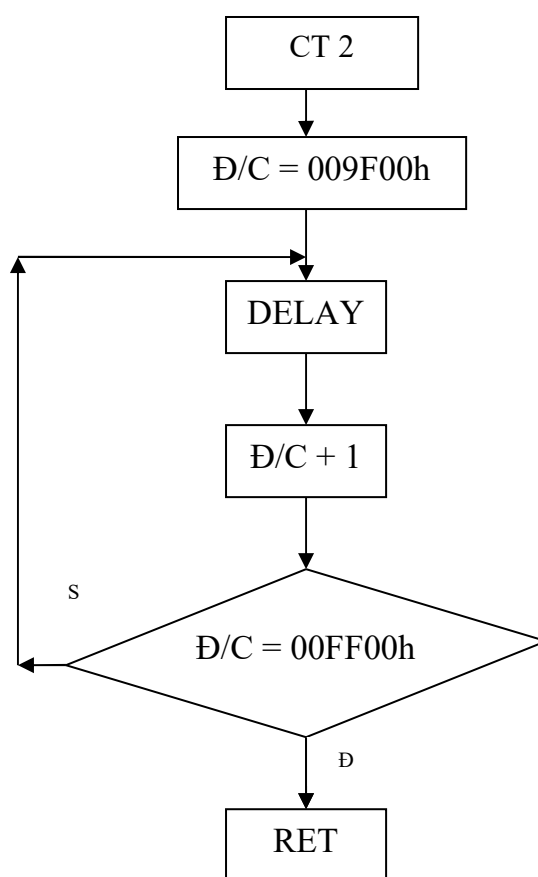
## PHẦN MỀM

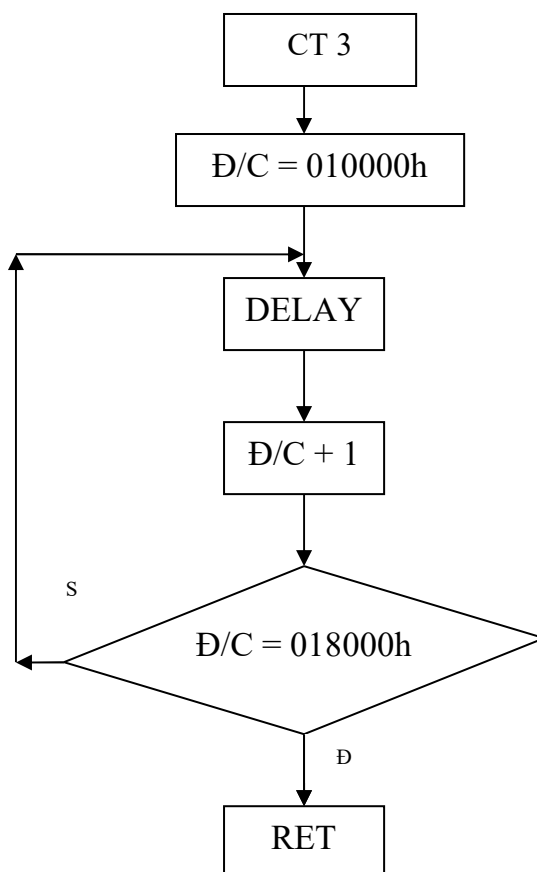


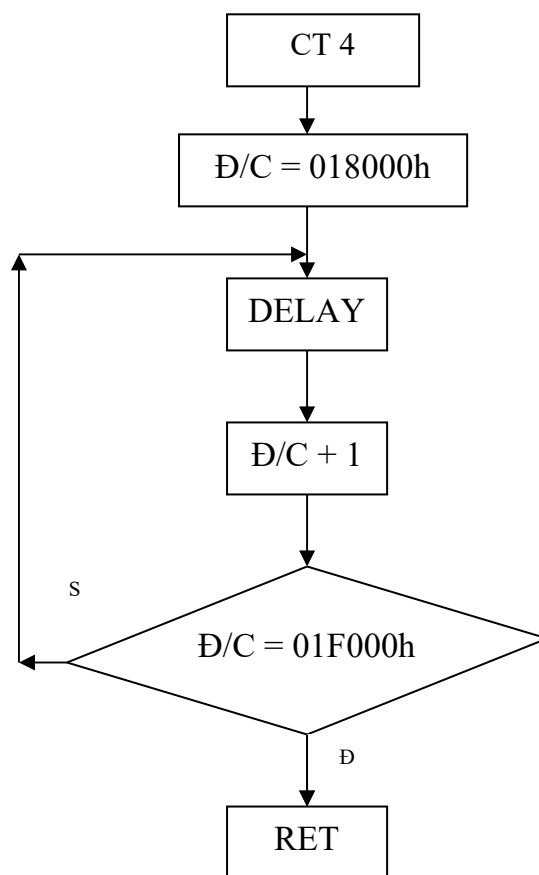
**A.CHƯƠNG TRÌNH VDK PHẦN ÂM THANH:****I.LƯU ĐỒ :**

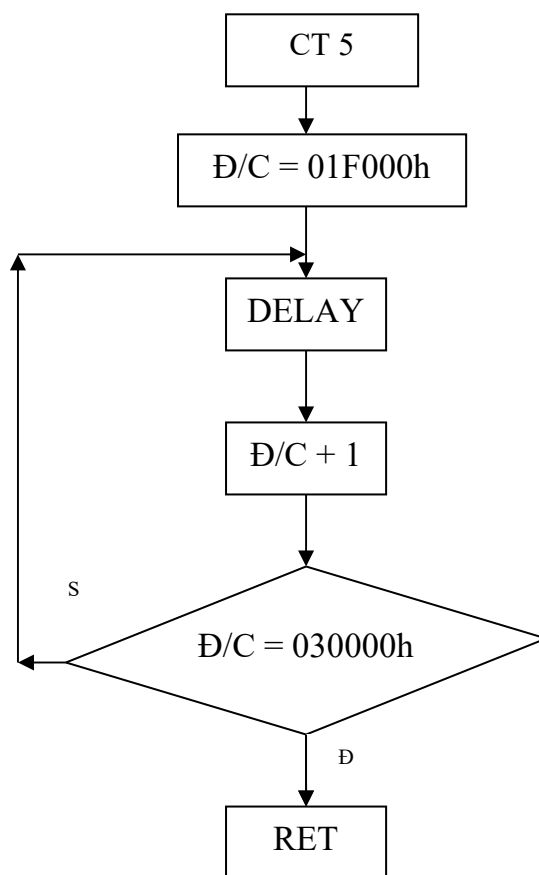












**II. PHẦN CHƯƠNG TRÌNH VĐK :**

```

                                ÂM THANH
                                org      00h

lb:                               mov     a,p3

                                cjne    a,#0feh,y1
                                lcall   ct1
y1:                               cjne    a,#0fdh,y2
                                lcall   ct2
y2:                               cjne    a,#0fch,y3
                                lcall   ct3
y3:                               cjne    a,#0fbh,y4
                                lcall   ct4
y4:                               cjne    a,#0fah,y5
                                lcall   ct5

y5:                               sjmp   lb

ct1:                              mov     p1,#00h
                                mov     p0,#00h
x2:                               mov     p2,#00h

x1:                               lcall  delay

                                inc     p2
                                mov     a,p2
                                cjne    a,#0ffh,x1

                                inc     p0
                                mov     a,p0
                                cjne    a,#09fh,x2

                                ret

ct2:                              mov     p1,#00h
x23:                             mov     p0,#09fh
x22:                              mov     p2,#00h

x21:                              lcall  delay

                                inc     p2
                                mov     a,p2
                                cjne    a,#0ffh,x21

                                inc     p0

```



```
                                mov    a,p0
                                cjne   a,#0ffh,x22

                                ret

ct3:                            mov    p1,#01h
x33:                            mov    p0,#00h
x32:                            mov    p2,#00h

x31:                            lcall  delay

                                inc    p2
                                mov    a,p2
                                cjne   a,#0ffh,x31

                                inc    p0
                                mov    a,p0
                                cjne   a,#080h,x32

                                ret

ct4:                            mov    p1,#01h
x43:                            mov    p0,#80h
x42:                            mov    p2,#00h

x41:                            lcall  delay

                                inc    p2
                                mov    a,p2
                                cjne   a,#0ffh,x41

                                inc    p0
                                mov    a,p0
                                cjne   a,#0f0h,x42

                                ret

ct5:                            mov    p1,#01h
x53:                            mov    p0,#0f0h
x52:                            mov    p2,#00h

x51:                            lcall  delay

                                inc    p2
                                mov    a,p2
                                cjne   a,#0ffh,x51
```

```
inc    p0
mov    a,p0
cjne   a,#0ffh,x52
```

```
mov    p0,#00h
inc    p1
mov    a,p1
cjne   a,#03h,x52
```

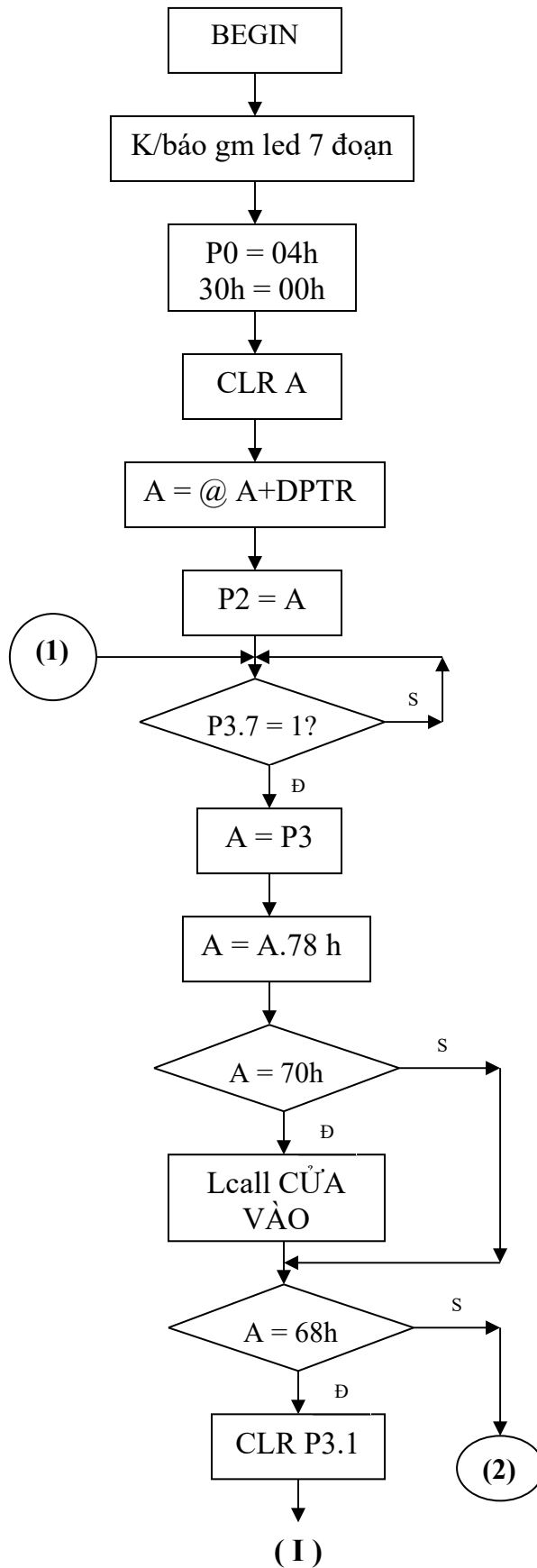
```
ret
```

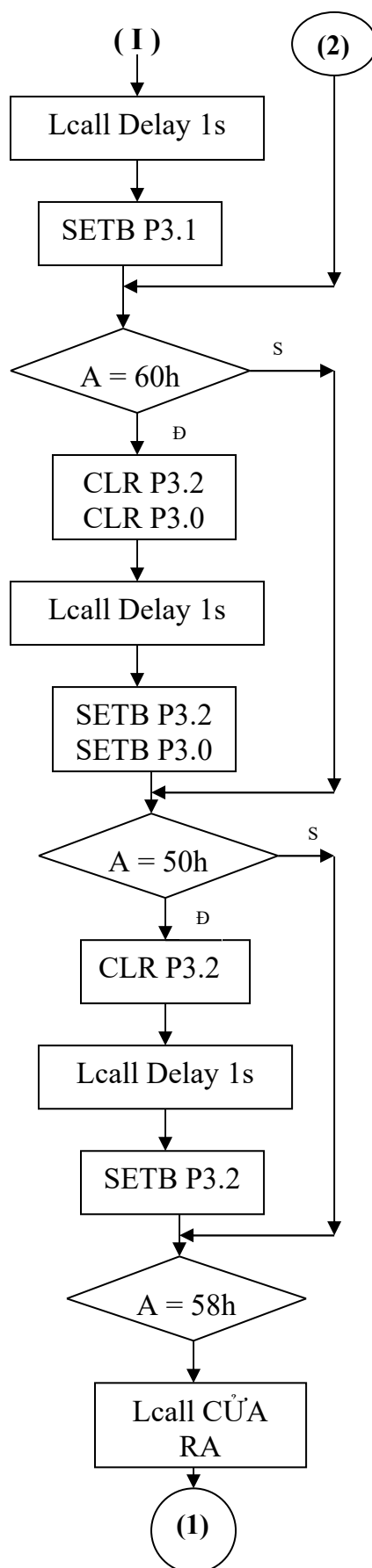
```
delay:
```

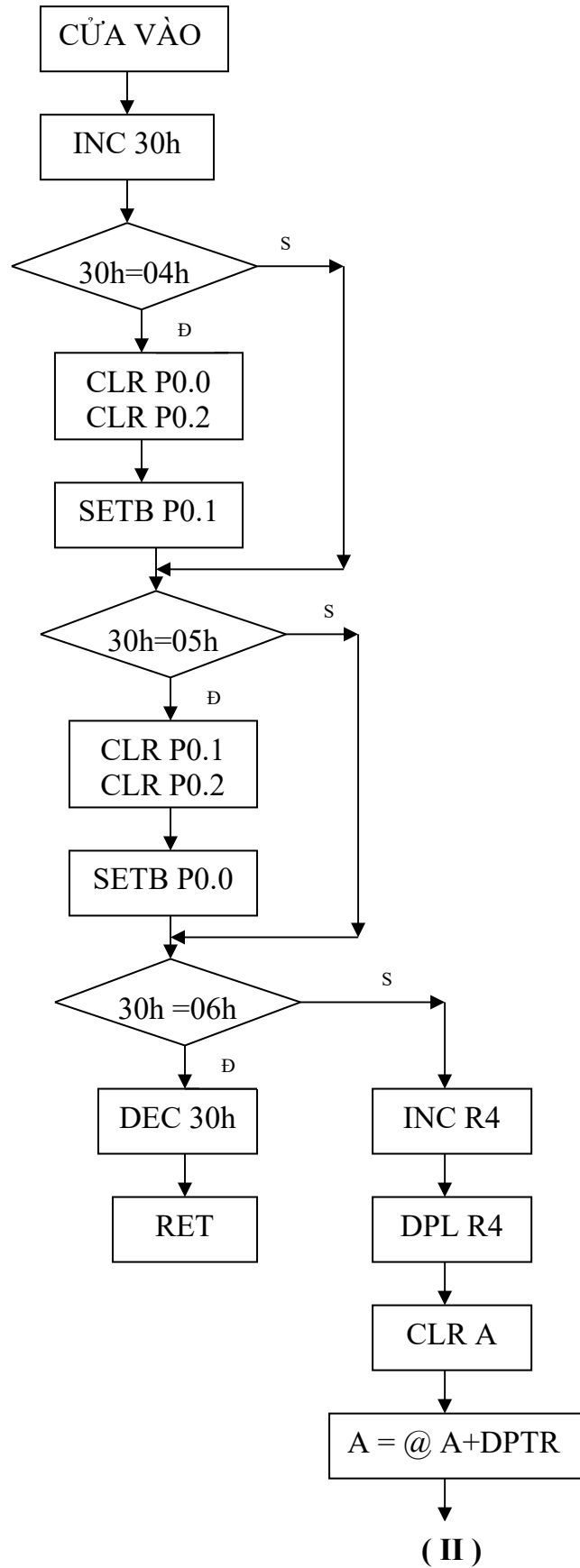
```
mov    r6,#54
del1:  djnz  r6,del1
ret
```

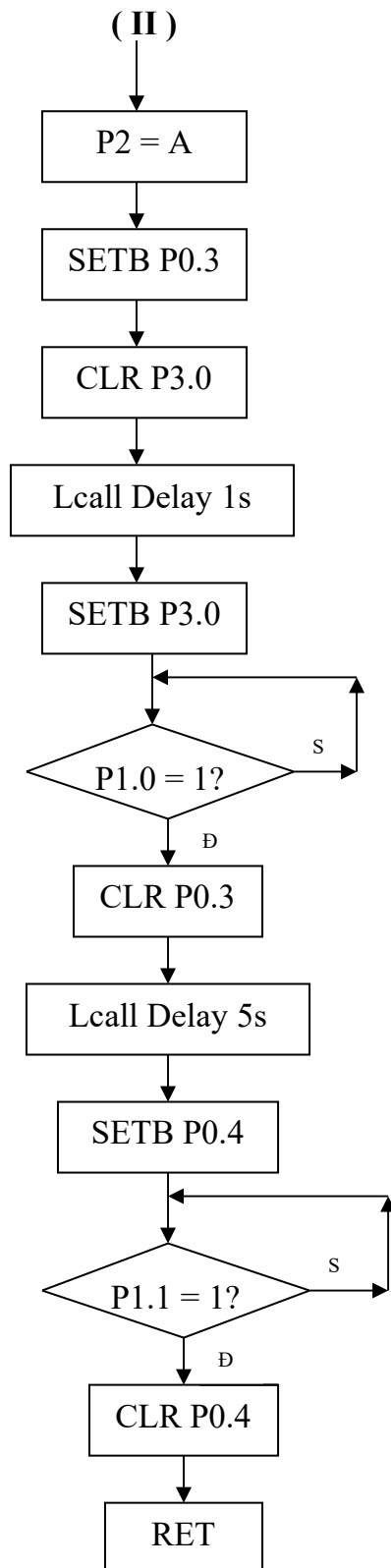
```
end
```

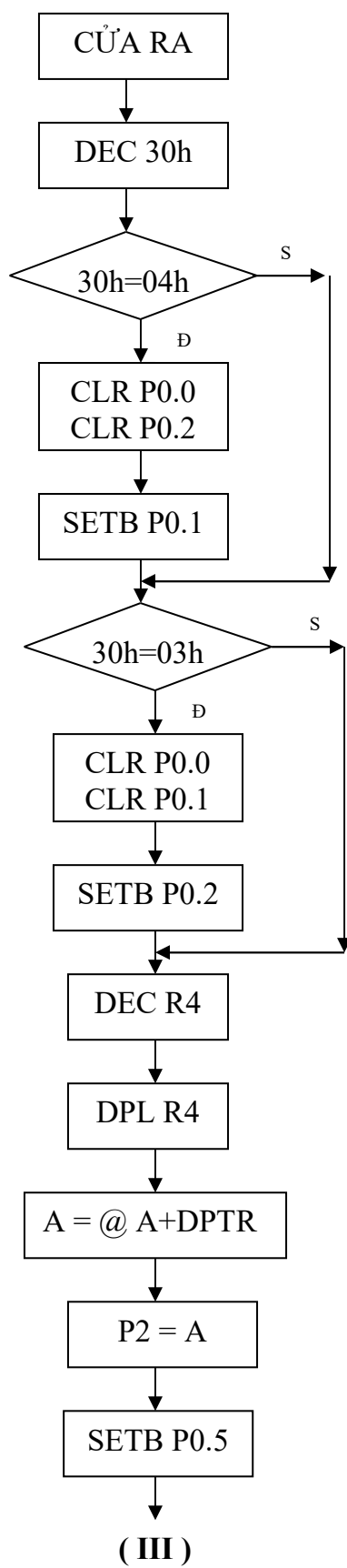
**B.CHƯƠNG TRÌNH VI ĐIỀU KHIỂN KHỐI XỬ LÝ TRÊN MÔ HÌNH :**  
**I.LƯU ĐỒ :**

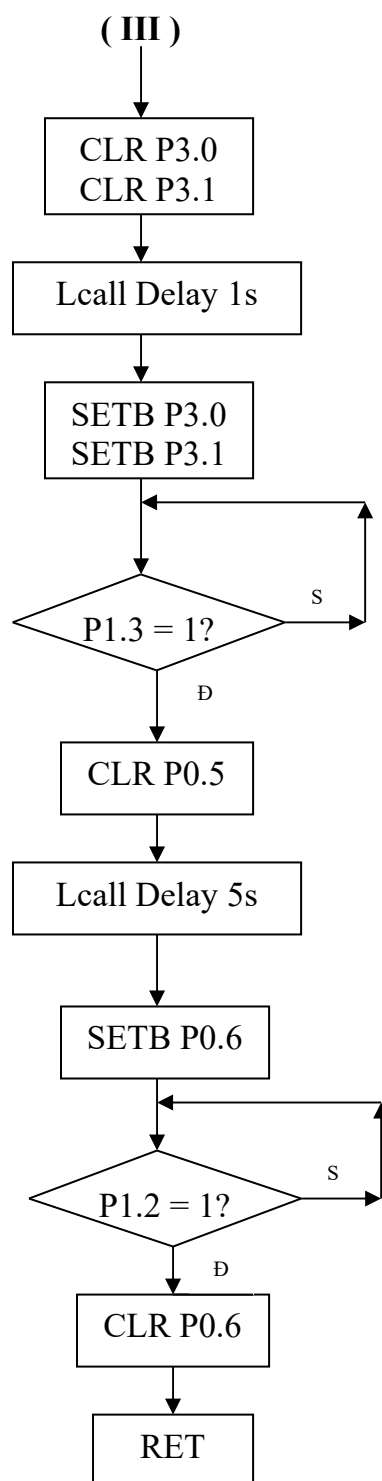














**II. PHẦN CHƯƠNG TRÌNH VĐK :****XỬ LÝ TRÊN MÔ HÌNH**

```

org    0600h
db
0c0h,0f9h,0a4h,0b0h,99h,92h,82h,0f1h,80h,90h
org    00h

mov    r3,#06h
mov    r4,#00h

mov    30h,#00h        ;bien dem so luong xe
trong bai
mov    p0,#04h        ;den xanh

mov    dph,r3
mov    dpl,r4
clr    a
movc   a,@a+dptr
mov    p2,a            ;so 0

lb:
jbp3.7,$

mov    a,p3
anl    a,#78h

cjne   a,#70h,x1      ;mo cua vao
lcall  cua_vao

x1:
cjne   a,#68h,x2
clr    p3.1           ;cho phep am thanh 2 "xin quy
khach vui long nhap mat ma"
lcall  delay1s
setb   p3.1

x2:
cjne   a,#60h,x3
clr    p3.2           ;cho phep am thanh 5 "yeu cau
dong tien"
clr    p3.0
lcall  delay1s
setb   p3.2
setb   p3.0

x3:
cjne   a,#50h,x4
clr    p3.2           ;cho phep am thanh 4 "mat ma
sai, xin quy khach vui long nhap lai"

```

```

                                lcall    delay1s
                                setb    p3.2

x4:                                cjne    a,#58h,x5        ;mo cua ra
                                lcall    cua_ra

x5:                                sjmp    lb

cua_vao:

                                inc     30h

                                mov     a,30h
                                cjne    a,#04h,x6        ;tin hieu den vang
                                clr     p0.0
                                clr     p0.2
                                setb    p0.1

x6:                                cjne    a,#05h,x7        ;tin hieu den do
                                clr     p0.1
                                clr     p0.2
                                setb    p0.0

x7:                                cjne    a,#06h,x8
                                dec     30h
                                ret

x8:

                                inc     r4
                                mov     dpl,r4
                                clr     a
                                movc   a,@a+dptr
                                mov     p2,a        ;hien thi so luong xe tang 1

                                setb    p0.3        ;cho phep mo cua vao

                                clr     p3.0        ;cho phep am thanh 1 "bai de
xe trung SPKT TP.HCM xin kinh chao quy khach"
                                lcall    delay1s
                                setb    p3.0

                                jnb     p1.0,$        ;kiem tra cua da mo het chua
                                clr     p0.3        ;tat tin hieu mo cua vao
                                lcall    delay5s        ;cho xe vao cua vao
                                setb    p0.4        ;dong cua vao
                                jnb     p1.1,$        ;kiem tra cua da dong het chua

```

```

        clr    p0.4        ;tat tin hieu dong cua vao
        ret

cua_ra:
        dec    30h

        mov    a,30h
        cjne   a,#04h,x9    ;tin hieu den vang
        clr    p0.0
        clr    p0.2
        setb   p0.1

x9:
        cjne   a,#03h,x10    ;tin hieu den xanh
        clr    p0.0
        clr    p0.1
        setb   p0.2

x10:
        dec    r4
        mov    dpl,r4
        clr    a
        movc   a,@a+dptr
        mov    p2,a        ;hien thi so luong xe giam 1

        setb   p0.5        ;cho phep mo cua ra

        clr    p3.0        ;cho phep am thanh 3 "chuc
        quy khach thuong lo binh an"

        clr    p3.1
        lcall  delay1s
        setb   p3.0
        setb   p3.1

        jnb    p1.3,$        ;kiem tra cua da mo het chua
        clr    p0.5        ;tat tin hieu mo cua ra

        lcall  delay5s        ;cho xe ra cua
        setb   p0.6        ;dong cua ra
        jnb    p1.2,$        ;kiem tra cua da dong het chua
        clr    p0.6        ;tat tin hieu dong cua ra

        ret

delay1s:
        mov    50h,#02h

```

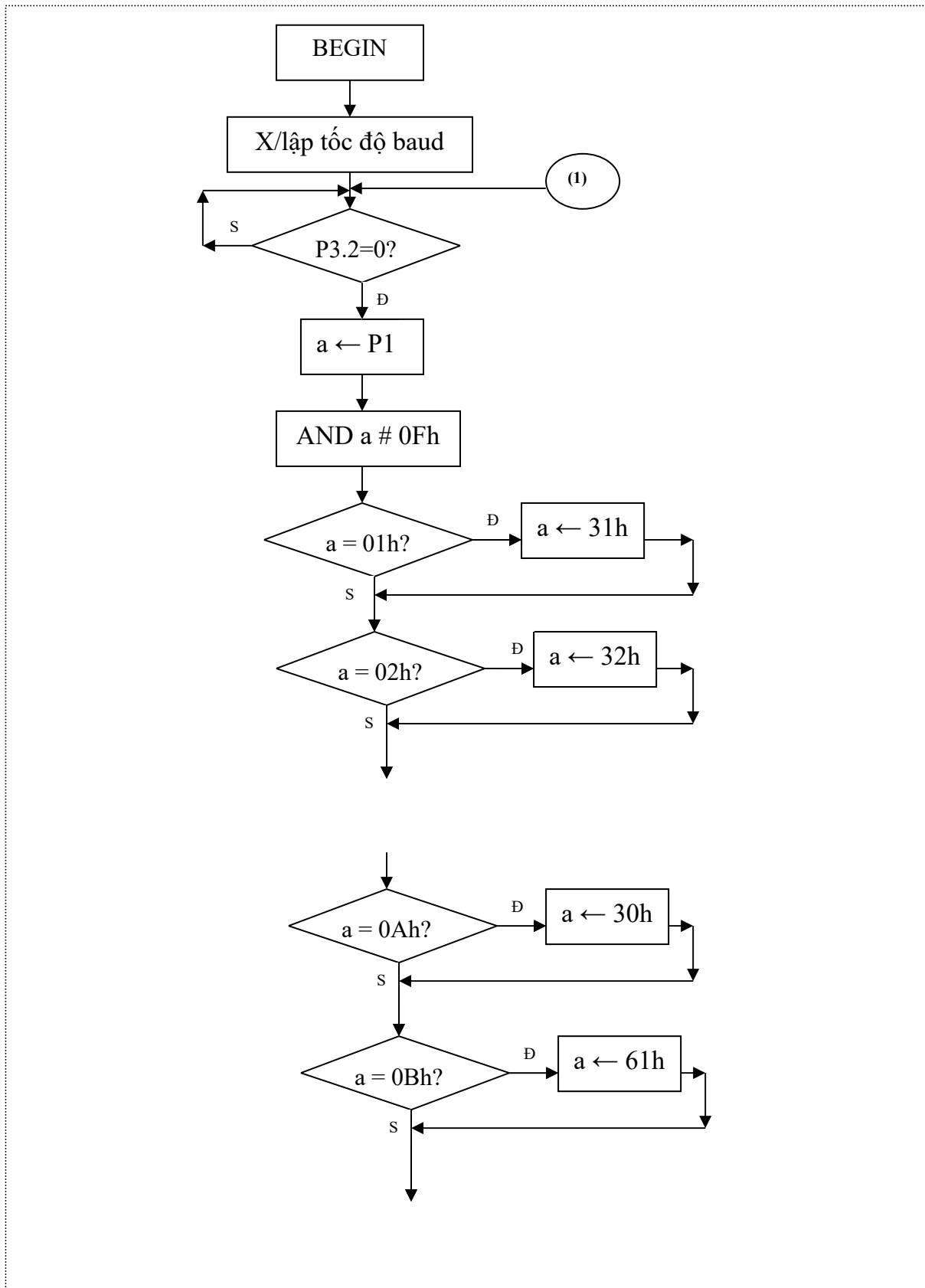
ĐỒ ÁN TỐT NGHIỆP

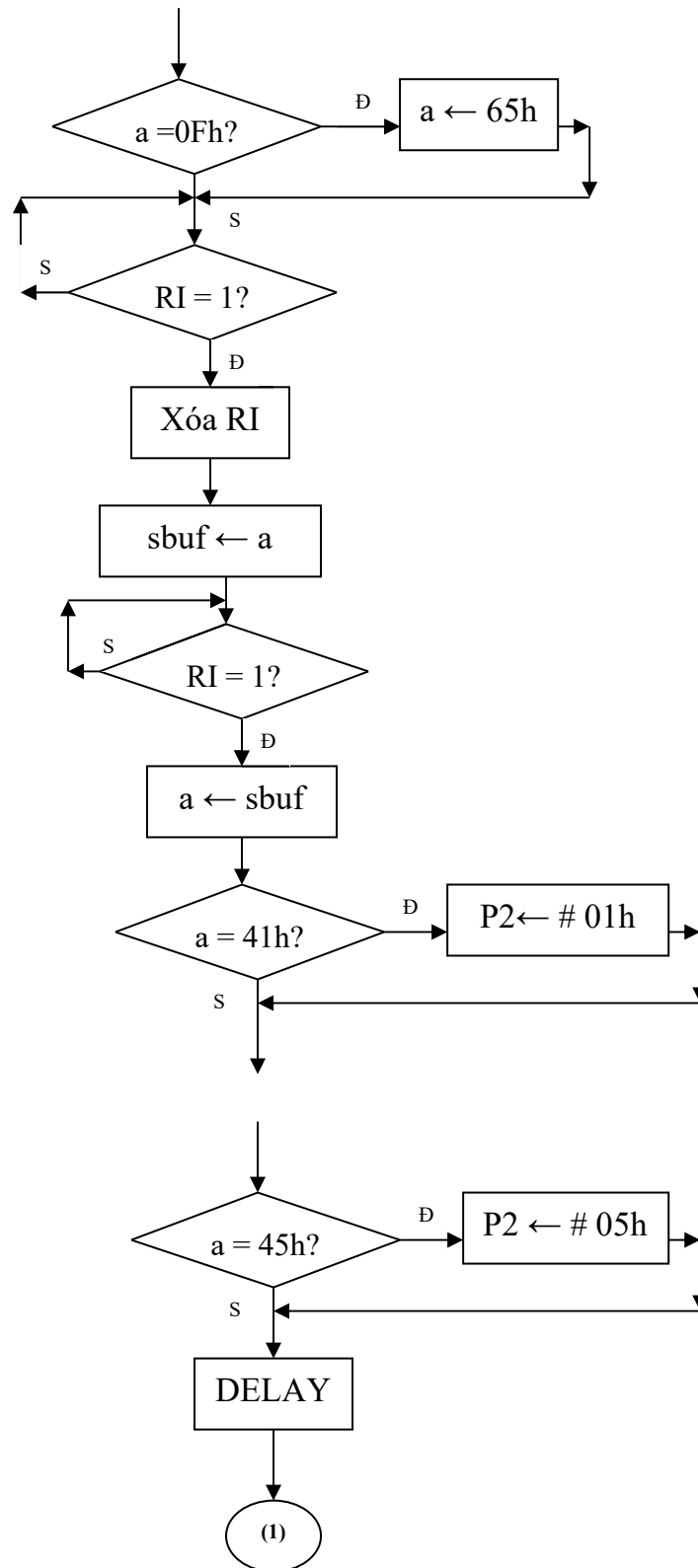
---

```
del2:          mov    51h,#0ffh
del1:          mov    52h,#0ffh
               djnz   52h,$
               djnz   51h,del1
               djnz   50h,del2
               ret
```

```
delay5s:
               mov    53h,#50h
del4:          mov    54h,#0ffh
del3:          mov    55h,#0ffh
               djnz   55h,$
               djnz   54h,del3
               djnz   53h,del4
               ret
```

```
end
```

**C.CHƯƠNG TRÌNH VĐK CỦA KHỐI GIAO TIẾP VỚI MÁY TÍNH :****I.LƯU ĐỒ :**



## II. PHẦN CHƯƠNG TRÌNH VDK : GIAO TIẾP MÁY TÍNH

```

                                org      00h

                                mov      tmod,#20h
                                mov      th1,#-24
                                mov      scon,#52h
                                setb     tr1

lb:                               mov      p2,#00h

                                jbp3.2,$
                                mov      a,p1
                                anl      a,#0fh

                                cjne     a,#01h,x1      ;phim so 1
                                mov      a,#31h

x1:                               cjne     a,#02h,x2      ;phim so 2
                                mov      a,#32h

x2:                               cjne     a,#03h,x3      ;phim so 3
                                mov      a,#33h

x3:                               cjne     a,#04h,x4      ;phim so 4
                                mov      a,#34h

x4:                               cjne     a,#05h,x5      ;phim so 5
                                mov      a,#35h

x5:                               cjne     a,#06h,x6      ;phim so 6
                                mov      a,#36h

x6:                               cjne     a,#07h,x7      ;phim so 7
                                mov      a,#37h

x7:                               cjne     a,#08h,x8      ;phim so 8
                                mov      a,#38h

x8:                               cjne     a,#09h,x9      ;phim so 9
                                mov      a,#39h

x9:                               cjne     a,#0ah,x10     ;phim so 0
                                mov      a,#30h

x10:                              cjne     a,#0bh,x11     ;xe1

```

```

mov    a,#61h    ;a
x11:   cjne    a,#0ch,x12    ;xe2
      mov    a,#62h    ;b
x12:   cjne    a,#0dh,x13    ;xe3
      mov    a,#63h    ;c
x13:   cjne    a,#0eh,x14    ;xe4
      mov    a,#64h    ;d
x14:   cjne    a,#0fh,x15    ;xe5
      mov    a,#65h    ;e
x15:
      jnb    ri,$
      clr    ri
      mov    sbuf,a
      jnb    ri,$
      clr    ri
      mov    a,sbuf
      cjne    a,#41h,x16    ;A cho xe vao bai
      mov    p2,#01h
x16:   cjne    a,#42h,x17    ;B yeu cau nhap ma
      mov    p2,#02h
x17:   cjne    a,#43h,x18    ;C yeu cau dong tien
      mov    p2,#03h
x18:   cjne    a,#44h,x19    ;D cho xe ra bai
      mov    p2,#04h
x19:   cjne    a,#45h,x20    ;E yeu cau nhap lai mat
      ma
      mov    p2,#05h
x20:   lcall   delay
      ljmp   lb
delay: mov    50h,#02h

```



ĐỒ ÁN TỐT NGHIỆP

---

```
del2:          mov    51h,#0ffh
del1:          mov    52h,#0ffh
               djnz  52h,$
               djnz  51h,del1
               djnz  50h,del2
               ret
```

```
end
```

**D. THIẾT KẾ GIAO DIỆN VÀ PHẦN MỀM VB :****Form 1 :**

The screenshot shows a Windows application window with the title bar 'DUNG'. The main content area has a dark blue background with white and red text. At the top, it reads 'TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM', 'KHOA ĐIỆN TỬ', and 'LỚP: 032011'. Below this is the title 'MÔ HÌNH BÃI DỂ XE TỰ ĐỘNG' in large red letters. The form is organized into several sections:

- Top Section:** 'HỌ VÀ TÊN' (Name) with an empty text box; 'NGÀY/GIỜ VÀO BÃI' (Date/Time In) with an empty text box; 'SỐ XE TRONG BÃI' (Number of cars in parking) with a display showing '0 XE'.
- Second Section:** 'SỐ CMND' (ID Number) with an empty text box and a 'ĐĂNG KÝ' (Register) button; 'NGÀY/GIỜ RA BÃI' (Date/Time Out) with a display showing '21:23:06 2/12/2007'; 'THỜI GIAN ĐỂ XE' (Parking duration) with an empty text box and 'GIỜ' (Hours) label.
- Third Section:** 'TÍNH TIỀN' (Calculate) button; 'ĐÃ THU' (Paid) button; 'XÓA' (Delete) button; 'MẬT MÃ' (Password) and 'MÃ CODE' (Code) with empty text boxes; 'ĐƠN GIÁ' (Price) with a display showing '10000 ĐỒNG'.
- Bottom Section:** 'GVHD' (Instructor) with a display showing 'LÊ THANH ĐẠO'; 'SVTH' (Student) with a display showing 'ĐẶNG QUỐC DŨNG NGUYỄN THẢO NGUYÊN'; 'THÀNH TIỀN' (Total amount) with an empty text box and 'VND' label.

**I. CHƯƠNG TRÌNH FROM 1 :**

```
Dim m As String  
Dim n1 As String  
Dim n2 As String  
Dim n3 As String  
Dim n4 As String  
Dim n As String
```

```
Dim k As Integer
```

```
Dim j As String
```

```
Dim ma1 As String  
Dim ma2 As String  
Dim ma3 As String  
Dim ma4 As String  
Dim ma5 As String
```

```
Dim code1 As String  
Dim code2 As String  
Dim code3 As String  
Dim code4 As String  
Dim code5 As String
```

```
Dim ngayvao As String  
Dim giovao As String  
Dim ngayra As String  
Dim giora As String  
Dim ngay As String  
Dim gio As String
```

```
Private Sub Command1_Click()  
Form1.Visible = False  
Form2.Visible = True  
Form2.Show
```

```
End Sub
```

```
Private Sub Command2_Click()  
If Text2.Text = Form2.Text1.Text Then  
If Text3.Text = Form2.Text6.Text Then  
Text4.Text = Form2.Text21.Text 'gio vao
```

```
Text17.Text = Form2.Text26.Text 'ngay vao  
Text7.Text = Form2.Text11.Text 'mat ma1  
Text8.Text = Form2.Text16.Text 'ma code1  
End If  
End If
```

```
If Text2.Text = Form2.Text2.Text Then  
If Text3.Text = Form2.Text7.Text Then  
Text4.Text = Form2.Text22.Text 'gio vao  
Text17.Text = Form2.Text27.Text 'ngay vao  
Text7.Text = Form2.Text12.Text 'mat ma2  
Text8.Text = Form2.Text17.Text 'ma code2  
End If  
End If
```

```
If Text2.Text = Form2.Text3.Text Then  
If Text3.Text = Form2.Text8.Text Then  
Text4.Text = Form2.Text23.Text 'gio vao  
Text17.Text = Form2.Text28.Text 'ngay vao  
Text7.Text = Form2.Text13.Text 'mat ma3  
Text8.Text = Form2.Text18.Text 'ma code3  
End If  
End If
```

```
If Text2.Text = Form2.Text4.Text Then  
If Text3.Text = Form2.Text9.Text Then  
Text4.Text = Form2.Text24.Text 'gio vao  
Text17.Text = Form2.Text29.Text 'ngay vao  
Text7.Text = Form2.Text14.Text 'mat ma4  
Text8.Text = Form2.Text19.Text 'ma code4  
End If  
End If
```

```
If Text2.Text = Form2.Text5.Text Then  
If Text3.Text = Form2.Text10.Text Then  
Text4.Text = Form2.Text25.Text 'gio vao  
Text17.Text = Form2.Text30.Text 'ngay vao  
Text7.Text = Form2.Text15.Text 'mat ma5  
Text8.Text = Form2.Text20.Text 'ma code5  
End If  
End If
```

```
giovao = Mid(Text4, 1, 2)  
giora = Mid(Text5, 1, 2)  
ngayvao = Mid(Text17, 3, 2)  
ngayra = Mid(Text16, 3, 2)
```

ngay = ngayra - ngayvao

```
If ngay = "" Then
gio = giora - giovao
Else
gio = ((ngay * 24) + giora) - giovao
Text6.Text = gio
Text10.Text = Text6 * Text9
End If
End Sub
```

```
Private Sub Command3_Click()
If Text8.Text = "01" Then
code1 = "da thu"
End If
```

```
If Text8.Text = "02" Then
code2 = "da thu"
End If
```

```
If Text8.Text = "03" Then
code3 = "da thu"
End If
```

```
If Text8.Text = "04" Then
code4 = "da thu"
End If
```

```
If Text8.Text = "05" Then
code5 = "da thu"
End If
```

End Sub

```
Private Sub Command4_Click()
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
Text17.Text = ""
Text7.Text = ""
Text8.Text = ""
Text6.Text = ""
Text10.Text = ""
End Sub
```

```
Private Sub Form_Load()
```

```

MSComm1.CommPort = 1
MSComm1.Settings = "1200,n,8,1"
MSComm1.InputLen = 0
MSComm1.PortOpen = True

```

```

Text1.Text = "0" 'so xe trong bai
k = 1
n = ""

```

```

ma1 = "0"
ma2 = "0"
ma3 = "0"
ma4 = "0"
ma5 = "0"
End Sub

```

```

Private Sub Timer1_Timer()
Text5.Text = Time$
Text16.Text = Date
m = MSComm1.Input

```

```

If m = "a" Then
If Form2.Text21.Text = "" Then
Form2.Text21.Text = Text5 'nhap gio vao cho code1
Form2.Text26.Text = Text16 'nhap ngay vao cho code1
MSComm1.Output = "A" 'mo cua cho xe vao bai
Text1.Text = (Text1) + 1 'tang so xe trong bai
Else
If code1 = "da thu" Then
ma1 = Form2.Text11.Text
MSComm1.Output = "B" 'yeu cau nhap mat ma
Else
MSComm1.Output = "C" 'yeu cau den tram dieu hanh dong tien
End If
End If
End If

```

```

If m = "b" Then
If Form2.Text22.Text = "" Then
Form2.Text22.Text = Text5 'nhap gio vao cho code2
Form2.Text27.Text = Text16 'nhap ngay vao cho code2
MSComm1.Output = "A" 'mo cua cho xe vao bai
Text1.Text = (Text1) + 1 'tang so xe trong bai
Else
If code2 = "da thu" Then
ma2 = Form2.Text12.Text

```

```

MSComm1.Output = "B"      'yeu cau nhap mat ma
Else
MSComm1.Output = "C"      'yeu cau den tram dieu hanh dong tien
End If
End If
End If

```

```

If m = "c" Then
If Form2.Text23.Text = "" Then
Form2.Text23.Text = Text5      'nhap gio vao cho code3
Form2.Text28.Text = Text16     'nhap ngay vao cho code3
MSComm1.Output = "A"          'mo cua cho xe vao bai
Text1.Text = (Text1) + 1      'tang so xe trong bai
Else
If code3 = "da thu" Then
ma3 = Form2.Text13.Text
MSComm1.Output = "B"          'yeu cau nhap mat ma
Else
MSComm1.Output = "C"          'yeu cau den tram dieu hanh dong tien
End If
End If
End If

```

```

If m = "d" Then
If Form2.Text24.Text = "" Then
Form2.Text24.Text = Text5      'nhap gio vao cho code4
Form2.Text29.Text = Text16     'nhap ngay vao cho code4
MSComm1.Output = "A"          'mo cua cho xe vao bai
Text1.Text = (Text1) + 1      'tang so xe trong bai
Else
If code4 = "da thu" Then
ma4 = Form2.Text14.Text
MSComm1.Output = "B"          'yeu cau nhap mat ma
Else
MSComm1.Output = "C"          'yeu cau den tram dieu hanh dong tien
End If
End If
End If

```

```

If m = "e" Then
If Form2.Text25.Text = "" Then
Form2.Text25.Text = Text5      'nhap gio vao cho code5
Form2.Text30.Text = Text16     'nhap ngay vao cho code5
MSComm1.Output = "A"          'mo cua cho xe vao bai
Text1.Text = (Text1) + 1      'tang so xe trong bai
Else

```

```

If code5 = "da thu" Then
ma5 = Form2.Text15.Text
MSComm1.Output = "B"           'yeu cau nhap mat ma
Else
MSComm1.Output = "C"           'yeu cau den tram dieu hanh dong tien
End If
End If
End If

```

```

If m = "0" Or m = "1" Or m = "2" Or m = "3" Or m = "4" Or m = "5" Or m = "6" Or m
= "7" Or m = "8" Or m = "9" Then

```

```

If k = 1 Then
n1 = m
MSComm1.Output = "F"
End If

```

```

If k = 2 Then
n2 = m
MSComm1.Output = "F"
End If

```

```

If k = 3 Then
n3 = m
MSComm1.Output = "F"
End If

```

```

If k = 4 Then
n4 = m
k = 0
n = n1 & n2 & n3 & n4

```

```

If n = ma1 Then
Form2.Text21.Text = ""
Form2.Text26.Text = ""
code1 = ""
ma1 = "0"
Text1.Text = (Text1) - 1           'giam so xe trong bai
MSComm1.Output = "D"             'cho xe ra bai
Else

```

```

If n = ma2 Then
Form2.Text22.Text = ""
Form2.Text27.Text = ""
code2 = ""
ma2 = "0"

```



```

Text1.Text = (Text1) - 1      'giam so xe trong bai
MSComm1.Output = "D"      'cho xe ra bai
Else

If n = ma3 Then
Form2.Text23.Text = ""
Form2.Text28.Text = ""
code3 = ""
ma3 = "0"
Text1.Text = (Text1) - 1    'giam so xe trong bai
MSComm1.Output = "D"      'cho xe ra bai
Else

If n = ma4 Then
Form2.Text24.Text = ""
Form2.Text29.Text = ""
code4 = ""
ma4 = "0"
Text1.Text = (Text1) - 1    'giam so xe trong bai
MSComm1.Output = "D"      'cho xe ra bai
Else

If n = ma5 Then
Form2.Text25.Text = ""
Form2.Text30.Text = ""
code5 = ""
ma5 = "0"
Text1.Text = (Text1) - 1    'giam so xe trong bai
MSComm1.Output = "D"      'cho xe ra bai
Else
MSComm1.Output = "E"      'yeu cau nhap lai mat ma
n = ""
End If
End If
End If
End If
End If

End If
m = ""
k = k + 1
End If

Text11.Text = n1
Text12.Text = n2
Text13.Text = n3

```

Text14.Text = n4

Text15.Text = n

End Sub

**Form 2 :**

Form2

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM  
KHOA ĐIỆN TỬ  
LỚP: 032011

**MÔ HÌNH BÃI ĐỂ XE TỰ ĐỘNG**

CODE	HỌ VÀ TÊN	CMND	MẬT MÃ	NGÀY	GIỜ VÀO
01	NGUYEN VAN A	1111	1111		
02	NGUYEN VAN B	2222	2222		
03	NGUYEN VAN C	3333	3333		
04	NGUYEN VAN D	4444	4444		
05	NGUYEN VAN E	5555	5555		

**THOÁT**

**II. CHƯƠNG TRÌNH FORM 2:**

```
Private Sub Command1_Click()  
'Unload Me  
Form2.Visible = False  
Form1.Visible = True  
End Sub  
Private Sub Form_Unload(Cancel As Integer)  
'Unload Me  
Form2.Visible = False  
Form1.Visible = True  
End Sub
```

**PHẦN E :**

**KẾT LUẬN VÀ HƯỚNG**

**PHÁT TRIỂN CỦA ĐỀ TÀI**

**I.TÓM TẮT NỘI DUNG ĐỀ TÀI :**

Hệ thống bãi giữ xe tự động là một hệ thống khá hoàn chỉnh.Hệ thống này có hai chức năng:

**1.Điều khiển các thiết bị điện từ xa với hệ thống sóng vô tuyến :**

Các thiết bị được điều khiển bởi hệ thống thu phát sóng vô tuyến thông qua khối xử lý vi điều khiển.Chức năng này đã được thể hiện qua nhiệm vụ của từng khối từ sơ đồ tổng quan của đề tài:

- *Khối phát xe 1,xe 2,...*:có nhiệm vụ phát tín hiệu có mã code vừa được cấp ở trạm điều hành đến khối thu giao tiếp với máy tính.
- *Khối thu giao tiếp PC1*: sẽ thu tín hiệu vô tuyến mang mã code từ khối phát bao gồm khối phát của xe và khối nhập mật mã. Các khối này được set sẵn mã địa chỉ là 00H .
- *Khối giao tiếp máy tính*: khối này nhận tín hiệu dữ liệu từ khối thu giao tiếp 1 sau đó truyền về máy tính . Sau đó chờ lệnh của PC để đưa lệnh này tới khối phát giao tiếp 2.
- *Khối phát giao tiếp PC2*: nhận dữ liệu từ khối giao tiếp và truyền dữ liệu này tới khối thu giao tiếp VĐK 2 . Cả hai khối này được set chung mã địa chỉ là 01H
- *Khối thu giao tiếp PC2*: nhận dữ liệu từ khối phát giao tiếp PC 2 và truyền dữ liệu này tới khối xử lý trên mô hình VĐK 2 .
- *Khối xử lý VĐK2* : nhận dữ liệu từ khối thu giao tiếp VĐK 2 , xử lý và điều khiển các thiết bị trên bãi giữ xe .

**2. Tự động báo động bảo vệ khi có sự cố :**

Đây là chức năng mang tính bảo vệ của hệ thống khi có sự cố xảy ra.Khi khách hàng đưa xe vào hoặc ra khỏi bãi đều được thông báo bằng âm thanh.Khi lấy xe ra chủ xe liên hệ với trạm điều hành để cung cấp mã code(của xe) cho trạm và thanh toán tiền ,đồng thời tại cửa ra của bãi chủ xe sẽ nhập mật mã(của chủ xe).Sau đó trạm điều sẽ mở mã để chủ xe được phép đưa xe ra khỏi bãi.Nếu như chủ xe nhập mật mã sai thì sẽ nhận được những thông báo bằng âm thanh.

**II.KẾT QUẢ ĐẠT ĐƯỢC :****1.Tính khoa học :**

Thế kỷ 21 là thế kỷ của Thông Tin Điện Tử-Viễn Thông,là nền tảng quan trọng trong việc hiện đại hóa toàn cầu,phù hợp với xu hướng phát triển của khoa học kỹ thuật.

Đề tài sử dụng kỹ thuật tiên tiến trong quá trình sử dụng sóng vô tuyến để điều khiển các thiết bị. Ngoài ra đề tài đã thể hiện tính ứng dụng đa năng trong kỹ thuật Vi Điều Khiển.

**2.Khả năng triển khai ứng dụng vào thực tiễn :**

- Mục tiêu của đề tài là thiết kế và thi công hệ thống mô hình bãi giữ xe tự động.Hệ thống bãi giữ xe tự động góp phần giải quyết nhu cầu trước hết là có nơi để xe bốn bánh do việc đi lại bằng xe bốn bánh ngày càng tăng,đồng thời cũng mang tính chuyên nghiệp đem lại cho khách hàng sự tiện ích,nhanh chóng.

- Ngoài ra hệ thống còn tạo cho khách hàng sự yên tâm bởi tính an toàn cao đối với tài sản của họ.

Với những đặc điểm và tính năng trên đề tài có khả năng triển khai và ứng dụng rộng rãi trong thực tế và mang tính hiện đại thực tiễn cao.

**3.Hiệu quả kinh tế :**

- Nhờ có sự tự động hóa con người sẽ tiết kiệm được thời gian cũng như số người phải tham gia vào công việc,với điều kiện thi công của đề tài rất khả thi,gia thành thấp,phù hợp với điều kiện hiện tại tại nước nhà.Đề tài sẽ mang tính tiện ích cao khi ứng dụng ở các thành phố,khu đô thị lớn.

- Hơn nữa về mang tính khả thi trong tương lai,đề tài mang tính kích thích thúc đẩy sự phát triển của ngành công nghệ điện tử Việt Nam.Từ đó góp phần xây dựng đất nước ngày càng phồn vinh,vì ngành điện tử là ngành đóng vai trò kinh tế mũi nhọn.

**III.KẾT LUẬN :**

- Quá trình thi công đề tài với sự hạn chế về thời gian kiến thức ,tài liệu cũng như về tài chính vì vậy đòi hỏi bản thân sinh viên thực hiện phải cố gắng tìm tòi và nhiệt tình trong công việc nghiên cứu đề tài và cuối cùng đề tài cũng đã hoàn thành một cách trọn vẹn, đúng thời hạn . Đó là kết quả của một thời gian dài nỗ lực và dưới sự tận tình hướng dẫn của giáo viên hướng dẫn cùng với sự giúp đỡ của nhà trường.

- Với đề tài này,yêu cầu có kiến thức tổng quát về nhiều mặt :kiến thức về kỹ thuật số,tương tự,xử lý tín hiệu ,xử lý âm thanh,lý thuyết mạch,kiến thức phần cứng,phần mềm của vi điều khiển....Sinh viên thực hiện đã cố gắng áp dụng nhuần nhuyễn giữa lý thuyết và thực hành.

- Đề tài “Bãi giữ xe tự động ”là sự kết hợp giữa các khối thành hệ thống hoàn chỉnh.Trong đề tài nghiên đã trình bày khá sâu sắc về cấu trúc và chức năng từng khối của phần cứng,phần mềm giúp cho người đọc dễ dàng nắm được cấu trúc và chức

năng của mạch. Nội dung của đề tài được trình bày đầy đủ, hình thức rõ ràng, từ ngữ thông dụng và dễ giúp cho người đọc hiểu được một cách dễ dàng để vận dụng đúng đắn và chính xác.

#### **IV. NHỮNG HẠN CHẾ VÀ HƯỚNG PHÁT TRIỂN CỦA ĐỀ TÀI :**

- Như đã trình bày do hạn chế thực tế cũng như tài chính sinh viên chưa thực hiện hết các tính năng của đề tài , nó bao gồm :

\* Phân thanh toán tiền còn chưa được tự động hóa , hướng giải quyết là thanh toán tiền qua mạng hay thẻ ATM ngoài ra còn có thể thanh toán hàng tháng hàng quý điều này đòi hỏi sinh viên phải am hiểu lĩnh vực mạng cũng như tài chính .

\* Về phần kỹ thuật còn hạn chế tính năng bảo vệ xe vì việc quản lý mã chưa là phương pháp tối ưu , ở đây sinh viên đưa ra hướng khắc phục là sử dụng phương pháp sử lý điểm ảnh để quản lý về biển số của xe. Tốc độ xử lý của VĐK còn hạn chế do không thể thực hiện nhiều việc trong cùng một lúc.



## TÀI LIỆU THAM KHẢO

### 1.SÁCH THAM KHẢO :

- 1.Tài liệu thực hành VI ĐIỀU KHIỂN – Nguyễn Đình Phú.
- 2.Tài liệu thực hành VI MẠCH – Nguyễn Đình Phú.
- 3.Tài liệu MẠCH SỐ – Nguyễn Hữu Phương.
- 4.Tài liệu bài tập thực hành LẬP TRÌNH VISUAL BASIC – Võ Văn Biện.
- 5.Tài liệu Vi xử lý – Hồ Trung Mỹ,NXB ĐHQGTPHCM 2003.
- 6.Tài liệu Kỹ thuật Điện tử – Lê Phi Yến,Lưu Phú,Nguyễn Như Anh.
7. Các đề tài tốt nghiệp của các sinh viên khóa trước.

### 2. CÁC WEB SITE THAM KHẢO :

- <http://www.Datasheet.com>
- <http://www.Google.Com.Vn>
- <http://www.Hiendaihoa.Com.vn>
- <http://home.Cogeco.ca/rpaisley4/circuitIndex.html>