

## **Mục lục**

	<b>Trang</b>
<b>I. ĐẶT VẤN ĐỀ</b>	<b>2</b>
<b>II. PHƯƠNG ÁN THIẾT KẾ</b>	<b>3</b>
1. Mô tả hoạt động của hệ thống	3
2. Phương án thiết kế	3
3. Sơ đồ khối hệ thống	4
<b>III. THIẾT KẾ</b>	<b>5</b>
1. Khối xử lý	5
1.1. Giới thiệu chung về vi điều khiển 8051	5
1.2. Chức năng của các chân tín hiệu	6
1.3. Bộ nhớ trên chip	10
1.4. Các Bộ định thời/Bộ đếm	11
1.5. Điều khiển ngắt	12
2. Khối giải mã	13
3. Khối hiển thị	15
<b>IV. NGUYÊN LÝ HOẠT ĐỘNG</b>	<b>16</b>
1. Nguyên lý hoạt động	16
2. Sơ đồ nguyên lý mạch đèn giao thông	18
3. Chương trình điều khiển	18
<b>V. MỞ RỘNG</b>	<b>23</b>
<b>Tài liệu tham khảo</b>	<b>25</b>

## **I. ĐẶT VẤN ĐỀ**

Trong hệ thống giao thông hiện nay ở nước ta, vấn đề về an toàn giao thông và tránh ùn tắc tại các đô thị và thành phố lớn là một trong những vấn đề hết sức cấp bách và được toàn xã hội quan tâm. Vì vậy các phương tiện hướng dẫn giao thông đóng vai trò rất quan trọng, nó góp phần hạn chế những xung đột xảy ra khi tham gia giao thông. Tại các đô thị thì hệ thống đèn điều khiển giao thông là rất quan trọng. Hệ thống đèn điều khiển giao thông không những có tác dụng hạn chế những xung đột trong giao thông thành phố mà còn là công cụ điều khiển các luồng giao thông nhằm hạn chế ùn tắc - một vấn đề nan giải tại các thành phố lớn. Vì lí do trên, chúng em quyết định chọn đề tài **“Thiết kế hệ thống đèn điều khiển giao thông”** cho bài tập lớn môn học **“Vi xử lý”**.

## II. PHƯƠNG ÁN THIẾT KẾ

### 1. Mô tả hoạt động của hệ thống

Mật độ giao thông phát triển mạnh mẽ ở các đô thị lớn đòi hỏi hệ thống điều khiển giao thông phải chính xác, linh hoạt, đơn giản nhưng hiệu quả cao. Hệ thống đèn giao thông hai pha phần nào đã đáp ứng được những yêu cầu trên.

Mỗi pha gồm các đèn: Xanh - Đỏ - Vàng điều khiển các phương tiện cơ giới.

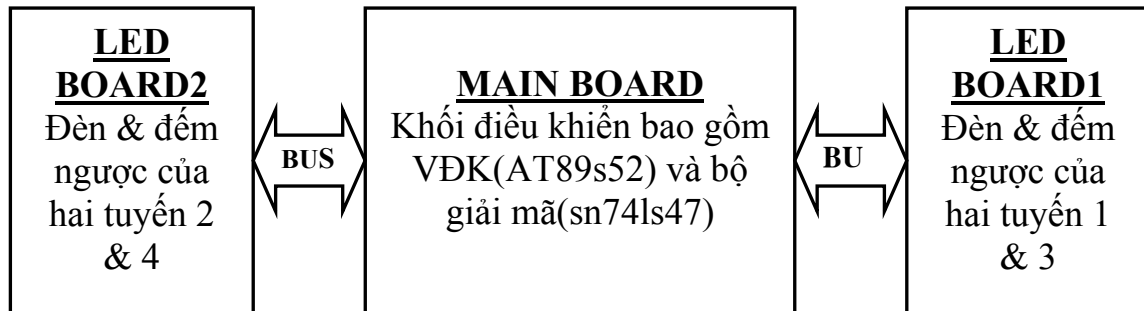
Hoạt động của từng pha ở chế độ điều khiển tự động như sau:

- ✓ **Đèn xanh:** trong 20 giây, hướng đi ứng với pha này được phép đi.
- ✓ **Đèn vàng:** trong 3 giây, thông báo cho các phương tiện ứng với pha này giảm tốc độ, chuẩn bị dừng lại.
- ✓ **Đèn đỏ:** trong 27 giây, các phương tiện dừng lại.
- ✓ **Khi đèn xanh của làn đường 1 sáng thì đèn xanh của người đi bộ ở làn đường kia cũng sang và ngược lại.**

### 2. Phương án thiết kế

Hiện nay việc sử dụng các mạch số kết hợp với chip vi điều khiển trong các hệ thống điều khiển tự động đã trở nên rất phổ biến vì những ưu việt của nó như: độ chính xác, khả năng lập trình được, tốc độ điều khiển nhanh, sử dụng đơn giản,... Mặt khác kỹ thuật số, vi xử lý, vi điều khiển là lĩnh vực đang phát triển mạnh mẽ và có ứng dụng trong rất nhiều ngành sản xuất. Vì vậy, ta sẽ thiết kế một hệ thống điều khiển giao thông đơn giản, chỉ sử dụng bộ vi điều khiển. Tất cả các tín hiệu điều khiển đều được đưa đến khối hiển thị trực tiếp từ các cổng của bộ vi điều khiển. Phương án này có đặc điểm là mạch gọn nhẹ, không quá phức tạp, cách thức bố trí linh kiện dễ dàng, lập trình đơn giản, dễ chỉnh sửa.

### 3. Sơ đồ khối hệ thống



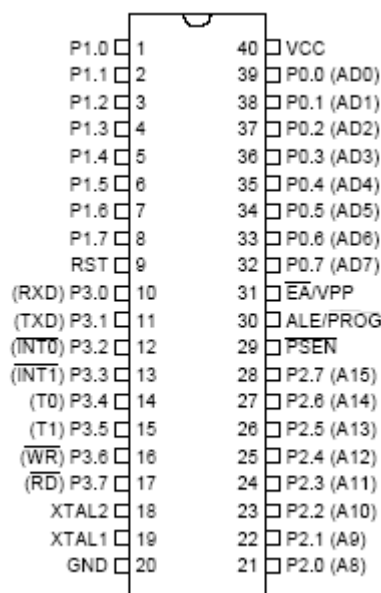
Hình 1: Sơ đồ khối hệ thống đèn giao thông

### III. THIẾT KẾ

#### 1. Khối xử lý

##### 1.1. Giới thiệu chung về vi điều khiển 8051

Vi điều khiển AT89C51 là một vi điều khiển thuộc họ 8051, loại CMOS, có tốc độ cao và công suất thấp với bộ nhớ Flash có thể lập trình được. Nó được sản xuất với công nghệ bộ nhớ không bay hơi mật độ cao của hãng Atmel. AT89C51 có 40 chân, được đóng gói theo tiêu chuẩn PDIP. Hình 2 biểu diễn sơ đồ chân ra và hình 3 biểu diễn sơ đồ khối chức năng của bộ vi điều khiển này.



Hình 2: Sơ đồ chân ra của vi điều khiển 8051

Các đặc điểm tiêu chuẩn (của họ vi điều khiển 8051):

- ✓ 4KB Flash ROM.
- ✓ 128 byte RAM.
- ✓ 4 cổng vào/ra song song 8 bit.
- ✓ 2 bộ định thời/đếm 16 bit.
- ✓ Kiến trúc 5 vectơ ngắt 2 mức (five vector two-level interrupt architecture).
- ✓ 1 cổng nối tiếp song công (full-duplex).

✓ Mạch tạo dao động trên chip và mạch đồng hồ.  
AT89C51 được thiết kế với logic tĩnh cho hoạt động có tần số giảm xuống 0 và hỗ trợ hai chế độ tiết kiệm năng lượng được lựa chọn bằng phần mềm. Chế độ nghỉ dừng CPU trong khi vẫn cho phép RAM, các bộ định thời/đếm, cổng nối tiếp và hệ thống ngắt tiếp tục hoạt động. Chế độ nguồn giảm duy trì nội dung của RAM nhưng không cho mạch dao động cung cấp xung clock nhằm vô hiệu hóa các hoạt động khác của chip cho đến khi có reset cứng tiếp theo.

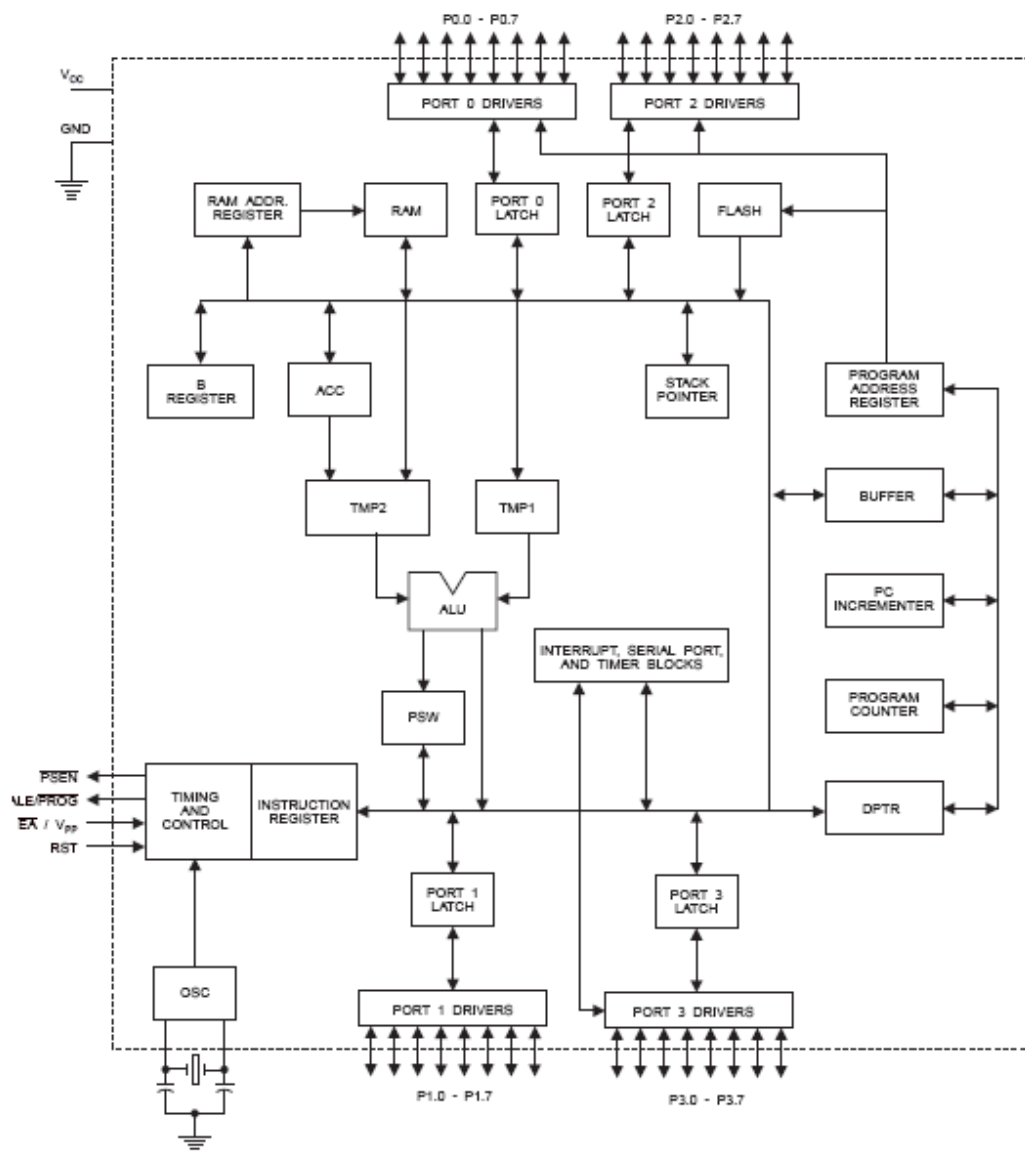
## **1.2. Chức năng của các chân tín hiệu**

### **1.2.1. Các cổng vào/ra song song**

8051 có 4 cổng vào/ra song song 8 bit là Port0, Port1, Port2, Port3. Các cổng này có thể sử dụng như là cổng vào hoặc cổng ra.

+) Cổng Port0 (các chân 32÷39) : là cổng vào/ra song song có hai chức năng. Trong các thiết kế cỡ nhỏ không dùng bộ nhớ mở rộng, nó có chức năng như các đường vào/ra. Trong các thiết kế cỡ lớn có bộ nhớ mở rộng nó trở thành bus địa chỉ và bus dữ liệu đa hợp.

+) Cổng Port1 (các chân 1÷8): là cổng vào/ra song song. Các chân được ký hiệu P1.0, P1.1, P1.2, ... có thể dùng cho giao tiếp với các thiết bị ngoại vi nếu cần. Cổng Port1 không có các chức năng khác, vì vậy chúng chỉ được dùng cho giao tiếp với các thiết bị ngoại vi.



Hình 3: Sơ đồ khối chức năng của vi điều khiển 8051

+) Cổng Port2 (các chân 21÷28): là một cổng vào/ra song song có tác dụng kép, được dùng như các đường xuất nhập hoặc là byte của bus địa chỉ 16 bit đối với các thiết bị dùng bộ nhớ mở rộng.

+) Cổng Port3 (các chân 10÷17): là cổng vào/ra song song có tác dụng kép. Khi không hoạt động xuất nhập các chân của cổng này có nhiều chức năng riêng. Bảng 1 cho ta chức năng của các chân cổng Port3

Bit	Tên	Chức năng chuyển đổi
P3.0	RxT	Đường vào dữ liệu cổng nối tiếp
P3.1	TxD	Đường xuất dữ liệu cổng nối tiếp
P3.2	INT0	Đường vào ngắt ngoài 0
P3.3	INT1	Đường vào ngắt ngoài 1
P3.4	T0	Đường vào của Bộ định thời/Bộ đếm thứ 0
P3.5	T1	Đường vào của Bộ định thời/Bộ đếm thứ 1
P3.6	WR	Tín hiệu ghi dữ liệu bộ nhớ ngoài
P3.7	RD	Tín hiệu đọc dữ liệu bộ nhớ ngoài

**Bảng 1: Các chức năng riêng của các chân cổng Port3**

### 1.2.2. Các chân tín hiệu điều khiển

Chân cho phép bộ nhớ chương trình PSEN (Program Storage Enable):

- ✓ Tín hiệu PSEN là tín hiệu ra ở chân 29 có tác dụng kép.
- ✓ Cho phép đọc bộ nhớ chương trình ngoài, thường được nối đến chân OE (Output Enable) của EPROM cho phép đọc các byte mã lệnh. Tín hiệu PSEN ở logic 0 trong thời gian vi điều khiển tìm nạp lệnh. Các mã lệnh được đọc từ EPROM qua bus dữ liệu và được chốt vào thanh ghi lệnh IR của vi điều khiển để giải mã.
- ✓ Khi vi điều khiển thi hành chương trình trong ROM nội PSEN sẽ ở mức logic 1.

Chân cho phép chốt địa chỉ ALE/PROG (Address Latch Enable):

- ✓ Chân tín hiệu ALE (chân 30) đưa ra xung điều khiển cho phép chốt byte thấp của địa chỉ khi vi điều khiển truy xuất bộ nhớ ngoài. Chân này cũng là đầu vào của xung lập trình khi lập trình cho FLASH, khi đó chân tín hiệu ở mức 0.
- ✓ Khi hoạt động bình thường, tín hiệu ALE được phát ra với tần số không đổi bằng 1/6 tần số của bộ tạo dao động trên chip, và có thể sử dụng cho mục đích định thời. Tuy nhiên, sẽ có một xung ALE bị bỏ qua mỗi khi vi điều khiển truy xuất bộ nhớ ngoài.

Chân tín hiệu truy xuất ngoài EA (External Access):

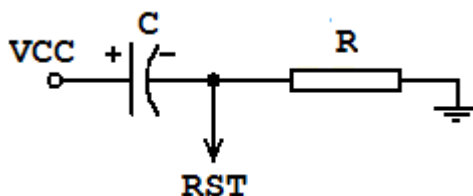
- ✓ Tín hiệu vào EA (chân 31) được nối với 5V (mức logic 1) hoặc với GND (mức 0). Nếu ở mức 1, vi điều khiển thi hành chương trình từ ROM nội. Nếu ở mức 0, vi điều khiển sẽ thi hành chương trình ở bộ nhớ mở rộng.



- ✓ Chân EA được lấy làm chân cấp nguồn 12V khi lập trình cho FLASH trong vi điều khiển.

Chân thiết lập lại RST (Reset):

- ✓ Chân RST (chân 9) là đường vào xóa chính của vi điều khiển dùng để thiết lập lại hệ thống. Khi chân tín hiệu này đưa lên mức 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.
- ✓ RST có thể được kích khi cấp điện dùng một mạch R-C. Mạch này như sau:



Hình 4: Mạch thiết lập lại cho 8051

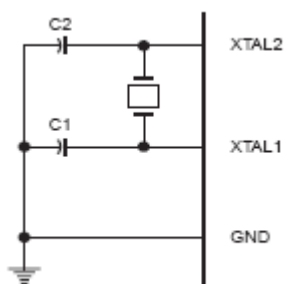
- ✓ Trạng thái các thanh ghi của vi điều khiển được tóm tắt trong bảng 2. Quan trọng nhất trong các thanh ghi trên là thanh ghi bộ đếm chương trình (PC – Program Counter). Sau khi thiết lập lại (RST trở về mức thấp), thanh ghi PC có giá trị 0000H, tức là chương trình luôn bắt đầu tại địa chỉ đầu tiên trong bộ nhớ chương trình. Nội dung của RAM trên chip không bị thay đổi khi thiết lập lại.

Thanh ghi	Nội dung
Bộ đếm chương trình	0000H
Thanh chứa A	00H
Thanh chứa B	00H
PSW	00H
SP	07H
DPTR	0000H
Port 0 – 3	FFH
IP	xxx00000B
IE	0xx00000B
Các thanh ghi định thời	00H
SCON	00H
SBUF	00H
PCON(HMOS)	0xxxxxxxB
PCON(CMOS)	0xxx0000B

Bảng 2: Trạng thái các thanh ghi sau khi Reset

Các chân XTAL1, XTAL2:

- ✓ Các chân này (chân 18, 19) nối với bộ tạo dao động trên chip. Mạch tạo dao động như sau:



Hình 5: Mạch tạo dao động cho bộ tạo dao động trên chip của AT89C51

- ✓ Tần số của dao động thường là 12MHz. Khi đó tụ có giá trị 33pF. Chân VCC nối đến +5V của nguồn cấp, chân GND nối đất.

### 1.3. Bộ nhớ trên chip

RAM trong:

- ✓ Bộ vi điều khiển 8051 có 128 byte RAM trong bao gồm 32 byte đầu tiên (00H đến 1FH) dành cho các thanh ghi, 16 byte tiếp theo (20H đến 2FH) là vùng RAM định địa chỉ theo bit, sau đó là 80 byte RAM nháp.
- ✓ Vùng thanh ghi có 32 byte, chia thành 4 khối (bank 0 đến bank 3), mỗi khối có 8 thanh ghi (từ R0 đến R7).
- ✓ Ở vùng RAM định địa chỉ theo bit, các bit được đánh địa chỉ từ 00H đến 7FH.

Các thanh ghi chuyên dụng (SFRs – Special Function Registers):

- ✓ Các thanh ghi này có địa chỉ từ 80H đến FFH. Chúng chứa nội dung của các thanh ghi điều khiển.
- ✓ Sau đây là một số thanh ghi chuyên dụng:

Thanh ghi	Mã gọi nhớ	Địa chỉ
Chốt cổng Port0	P0	80H
Chốt cổng Port1	P1	90H
Chốt cổng Port2	P2	A0H
Chốt cổng Port3	P3	B0H

Điều khiển Bộ định thời/Bộ đếm	TCON	88H
Điều khiển chế độ Bộ định thời/Bộ đếm	TMOD	89H
Byte thấp Bộ định thời/Bộ đếm 0	TL0	8AH
Byte cao Bộ định thời/Bộ đếm 0	TH0	8BH
Byte thấp Bộ định thời/Bộ đếm 1	TL1	8CH
Byte cao Bộ định thời/Bộ đếm 1	TH1	8DH
Cho phép ngắt	IE	A8H
Điều khiển ưu tiên ngắt	IP	B8H
Từ trạng thái chương trình	PSW	D0H
Thanh ghi tích lũy	ACC hoặc A	E0H
Thanh ghi B	B	F0H

**Bảng 3: Một số thanh ghi chuyên dụng của vi điều khiển 8051**

ROM:

- ✓ Bộ vi điều khiển AT89C51 có 4KB FLASH lập trình được.
- ✓ ROM luôn chiếm vùng địa chỉ thấp nhất trong bộ nhớ chương trình.

#### 1.4. Các Bộ định thời/Bộ đếm

Bộ vi điều khiển 8051 có 2 Bộ định thời/Bộ đếm là Bộ định thời/Bộ đếm 0 và Bộ định thời/Bộ đếm 1. Chúng có thể hoạt động như là bộ định thời hoặc bộ đếm.

Chế độ hoạt động của các Bộ định thời/Bộ đếm được cất trong thanh ghi TMOD:

GATE1	C/T1	M1(1)	M0(1)	GATE0	C/T0	M1(0)	M0(0)
-------	------	-------	-------	-------	------	-------	-------

- ✓ Nếu bit GATE xóa, các Bộ định thời/Bộ đếm được phép hoạt động khi bit TR# tương ứng trong thanh ghi TCON thiết lập. Ngược lại, nếu GATE thiết lập thì các Bộ định thời/Bộ đếm chỉ hoạt động khi các chân INT# tương ứng tích cực (mức thấp).
- ✓ Bit C/T# dùng để lựa chọn chế độ hoạt động bộ đếm hay bộ định thời. Nếu được thiết lập thì nó hoạt động theo chế độ đếm sự kiện, lúc này nguồn xung cho bộ đếm là xung ngoài đưa vào từ chân T# tương ứng (chân 14, 15). Nếu bị xóa, thì nó hoạt động theo chế độ định thời với nguồn xung là xung tạo ra từ bộ tạo dao động trên chip sau khi chia 12.

- ✓ Các bit M0, M1 dùng để xác định chế độ đếm cho các bộ đếm:

M1	M0	Chế độ
0	0	Đếm 13 bit
0	1	Đếm 16 bit
1	0	Đếm 8 bit và tự động nạp lại
1	1	Bộ đếm 0 đếm 8 bit riêng rẽ Bộ đếm 1 dùng hoạt động

**Bảng 4: Các chế độ hoạt động của các bộ đếm**

Sự hoạt động của các Bộ định thời/Bộ đếm được điều khiển bởi thanh ghi TCON:

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
-----	-----	-----	-----	-----	-----	-----	-----

- ✓ Các bit TR# cho phép Bộ định thời/Bộ đếm hoạt động (nếu được thiết lập) hoặc không cho phép chúng hoạt động (nếu bị xóa).
- ✓ Các bit TF# là các cờ tràn tương ứng với các Bộ định thời/Bộ đếm. Chúng được thiết lập khi xảy ra tràn và được xóa bằng phần cứng nếu khi đó bộ xử lý rẽ nhánh đến chương trình phục vụ ngắt tương ứng.
- ✓ Các bit IT# là các bit ngắt. Thiết lập IT# tạo ra chế độ ngắt cạnh, chân INT# nhận ra một ngắt khi nhận ra có một sườn âm (↓) của tín hiệu vào. Xóa IT# tạo ra chế độ ngắt mức, tức là ngắt được tạo ra khi tín hiệu vào ở mức thấp (0). Ở chế độ ngắt mức, khi tín hiệu vào còn ở mức thấp thì ngắt được tạo ra liên tục cho đến khi tín hiệu vào chuyển lên mức cao hoặc thiết lập IT#.
- ✓ Các bit IE# là các cờ ngắt cạnh, được thiết lập khi dò thấy ngắt cạnh.

## 1.5. Điều khiển ngắt

Bộ vi điều khiển 8051 có 5 nguồn ngắt: TF0, TF1, INT0, INT1 và ngắt do cổng nối tiếp. Sự điều khiển hoạt động ngắt được cất trong 2 thanh ghi là thanh ghi cho phép ngắt IE (Interrupt Enable) và thanh ghi xác định thứ tự ưu tiên ngắt IP (Interrupt Priority).

Các bit và chức năng của chúng trong thanh ghi IE như sau (thiết lập là cho phép, xóa là cấm):

Bit	Mã gọi nhớ	Chức năng
7	EA	Cho ngắt toàn cục
6	--	Không dùng
5	ET2	Cho phép ngắt do bộ định thời 2
4	ES	Cho phép ngắt do cổng nói tiếp
3	ET1	Cho phép ngắt do bộ đếm 1
2	EX1	Cho phép ngắt từ bên ngoài 1
1	ET0	Cho phép ngắt do bộ đếm 0
0	EX0	Cho phép ngắt từ bên ngoài 0

**Bảng 5:** Các bit và chức năng của nó trong thanh ghi IE

Với thanh ghi IP:

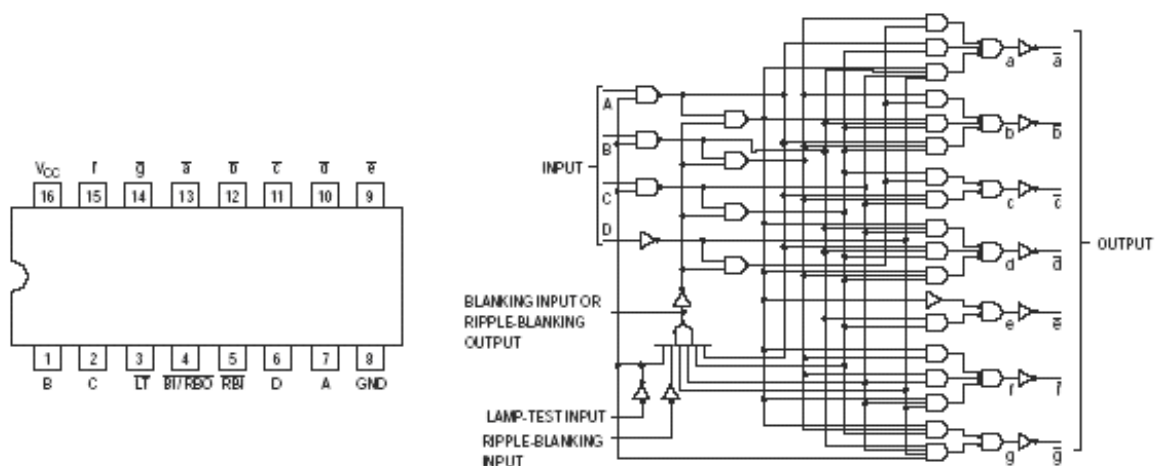
Bit	Mã gọi nhớ	Chức năng
7	--	Không dùng
6	--	Không dùng
5	PT2	Ưu tiên ngắt do bộ định thời 2
4	PS	Ưu tiên ngắt do cổng nói tiếp
3	PT1	Ưu tiên ngắt do bộ đếm 1
2	PX1	Ưu tiên ngắt từ bên ngoài 1
1	PT0	Ưu tiên ngắt do bộ đếm 0
0	PX0	Ưu tiên ngắt từ bên ngoài 0

**Bảng 6:** Các bit và chức năng của chúng trong thanh ghi IP

## 2. Khối giải mã

Khối giải mã nhận tín hiệu điều khiển từ khối xử lý, sau đó giải mã để đưa đến hiển thị trên các đồng hồ đếm ngược. Với chức năng trên thì khối này chính là khối giải mã cho đèn LED 7 thanh.

Vì các đồng hồ được hiển thị bằng các đèn LED 7 thanh nên ta sẽ sử dụng bộ giải mã là các IC 74LS47. Sơ đồ chân ra và sơ đồ khối chức năng như sau:



**Hình 6: Sơ đồ chân ra và sơ đồ khối chức năng của IC 74LS47**

Các đầu ra (từ a đến f) nối đến các chân tương ứng của LED 7 thanh. Ta thấy các đầu ra đều có mức tích cực là mức thấp. Do đó, loại LED 7 thanh cần sử dụng là loại Anode chung. LED 7 thanh sẽ ghép nối với IC này theo bảng chân lý sau:

TRUTH TABLE

INPUTS							OUTPUTS								
DECIMAL OR FUNCTION	LT	RBI	D	C	B	A	BI/RBO	$\bar{a}$	$\bar{b}$	$\bar{c}$	$\bar{d}$	$\bar{e}$	$\bar{f}$	$\bar{g}$	NOTE
0	H	H	L	L	L	L	H	L	L	L	L	L	L	H	A
1	H	X	L	L	L	H	H	H	L	L	H	H	H	H	A
2	H	X	L	L	H	L	H	L	L	H	L	L	H	L	
3	H	X	L	L	H	H	H	L	L	L	L	H	H	L	
4	H	X	L	H	L	L	H	H	L	L	H	H	L	L	
5	H	X	L	H	L	H	H	L	H	L	L	H	L	L	
6	H	X	L	H	H	L	H	H	H	L	L	L	L	L	
7	H	X	L	H	H	H	H	L	L	L	H	H	H	H	
8	H	X	H	L	L	L	H	L	L	L	L	L	L	L	
9	H	X	H	L	L	H	H	L	L	L	H	H	L	L	
10	H	X	H	L	H	L	H	H	H	H	L	L	H	L	
11	H	X	H	L	H	H	H	H	H	L	L	H	H	L	
12	H	X	H	H	L	L	H	H	L	H	H	H	L	L	
13	H	X	H	H	L	H	H	L	H	H	L	H	L	L	
14	H	X	H	H	H	L	H	H	H	H	L	L	L	L	
15	H	X	H	H	H	H	H	H	H	H	H	H	H	H	
BI	X	X	X	X	X	X	L	H	H	H	H	H	H	H	B
RBI	H	L	L	L	L	L	L	H	H	H	H	H	H	H	C
LT	L	X	X	X	X	X	H	L	L	L	L	L	L	L	D

H = HIGH Voltage Level    L = LOW Voltage Level    X = Immaterial

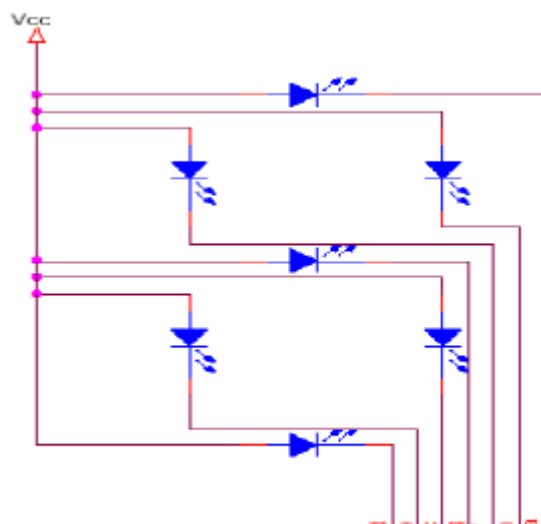
Bảng 7: Bảng chân lý của IC 74LS47

### 3. Khối hiển thị

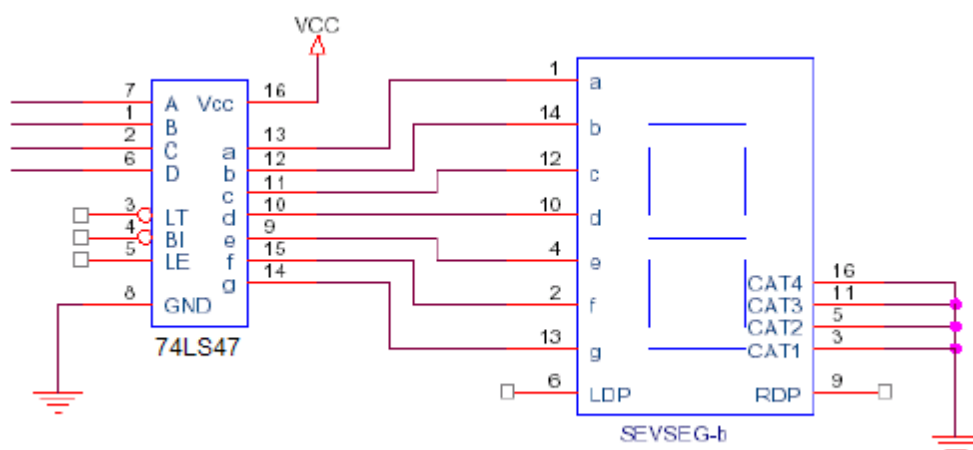
Khối hiển thị có chức năng đưa ra thông tin điều khiển giao thông tương ứng với trạng thái hiện thời của hệ thống. Khối này gồm 2 phần: đèn điều khiển và đồng hồ.

Đèn điều khiển bao gồm: Đèn dành cho các phương tiện tham gia giao thông: xanh, đỏ, vàng.

Đồng hồ dùng các LED 7 thanh để tạo thành các bộ hiển thị từ 00 đến 99. Ở đây ta dùng loại Anode chung (phù hợp với IC 74LS47), cấu trúc của nó như sau:



Hình 7: Cấu trúc bên trong của LED 7 thanh Anode chung



Hình 8: Ghép nối 74LS47 đến LED 7 thanh

## IV. NGUYÊN LÝ HOẠT ĐỘNG

### 1. Nguyên lý hoạt động

Các mạch hiển thị hoạt động trên nguyên tắc điều khiển bit vào/ra của các cổng trên IC 89C51.



Cụ thể như sau: Vi điều khiển được lập trình để điều khiển 2 công việc chính:

**a) Chuyển mức của các đèn tín hiệu trên hai làn đường(Xanh,đỏ,vàng)**

Việc chuyển mức này sẽ được thực hiện bằng 6 bit truyền tín hiệu:

Làn đường thứ nhất:

P3.5: đèn xanh.

P3.6: đèn vàng.

P3.7: đèn đỏ.

Làn đường thứ hai:

P0.2: đèn xanh.

P0.1: đèn vàng.

P0.0: đèn đỏ.

Các đèn hiển thị là các LED đơn nối chung nhau anốt. Đèn sáng sẽ tương ứng với mức logic thấp. Giữa các lần hiển thị sẽ kết hợp đèn của hai làn đường. Đèn xanh ở làn 1 thì tương ứng sẽ là đèn đỏ ở làn 2 và ngược lại. Và chuyển mức giữa các đèn sẽ xen kẽ những bộ đếm,kết thúc mỗi bộ đếm sẽ tương ứng thiết lập lại các bit để hiển thị các trạng thái đèn tiếp theo.

Giữa đèn xanh và đèn vàng là 20s

Giữa đèn vàng và đèn đỏ là trễ một thời gian

Giữa đèn đỏ và đèn xanh là 27s.

Có đèn báo hiệu cho người đi bộ của hai làn đường.

**b) Hiển thị bộ đếm tương ứng với mỗi trạng thái đèn:**

- Ta sử dụng IC 7447 để thực hiện việc giải mã BCD sang thập phân và hiển thị các trạng thái thập phân đó trên LED 7 thanh.

- Mã BCD là mã 4 bit(tương ứng sẽ hiển thị được một LED 7 thanh).

Ta sử dụng số có hai chữ số để đếm cho hai làn đường.Vì thế 8 bit của cổng P2 sẽ được dùng cho việc hiển thị hai LED trên làn đường thứ nhất và 8 bit của cổng P1 sẽ được sử dụng cho hiển thị LED tại làn đường thứ hai.

Các cổng 8 bit này sẽ được nối tới 2 IC 7447 tương ứng(4 bit cho một IC) và ở đầu ra(trên LED 7 thanh) sẽ hiển thị trạng thái thập phân của số Hexa đang gửi tới cổng đó.

- Việc tạo ra số thập phân được thực hiện bằng cách ghi dãy số Hexa 27 giảm dần vào vị trí ROM nội bắt đầu từ địa chỉ 40H.(Lúc này ta sử dụng chân số 31(EA/VPP) và cho nó mức điện áp cao để đọc dữ liệu từ Rom nội).

Khi ghi dãy số Hexa 27-0 ta sẽ bỏ qua các trạng thái 1F, 1E...1A, F, E,..., A để tương ứng ta thu được mã BCD hiển thị ra dãy số thập phân 27-0. Khi đó ta sẽ tách làm 3 vòng lặp:

- ✓ Vòng lặp thứ nhất sẽ ghi các số từ 27-20. (tương ứng là các địa chỉ 40h-47h).
- ✓ Vòng lặp thứ hai sẽ ghi các số từ 19-10. (tương ứng là địa chỉ 48h-51h).
- ✓ Vòng lặp cuối cùng là các số từ 9-0. (ứng với các địa chỉ 52h-5Bh).

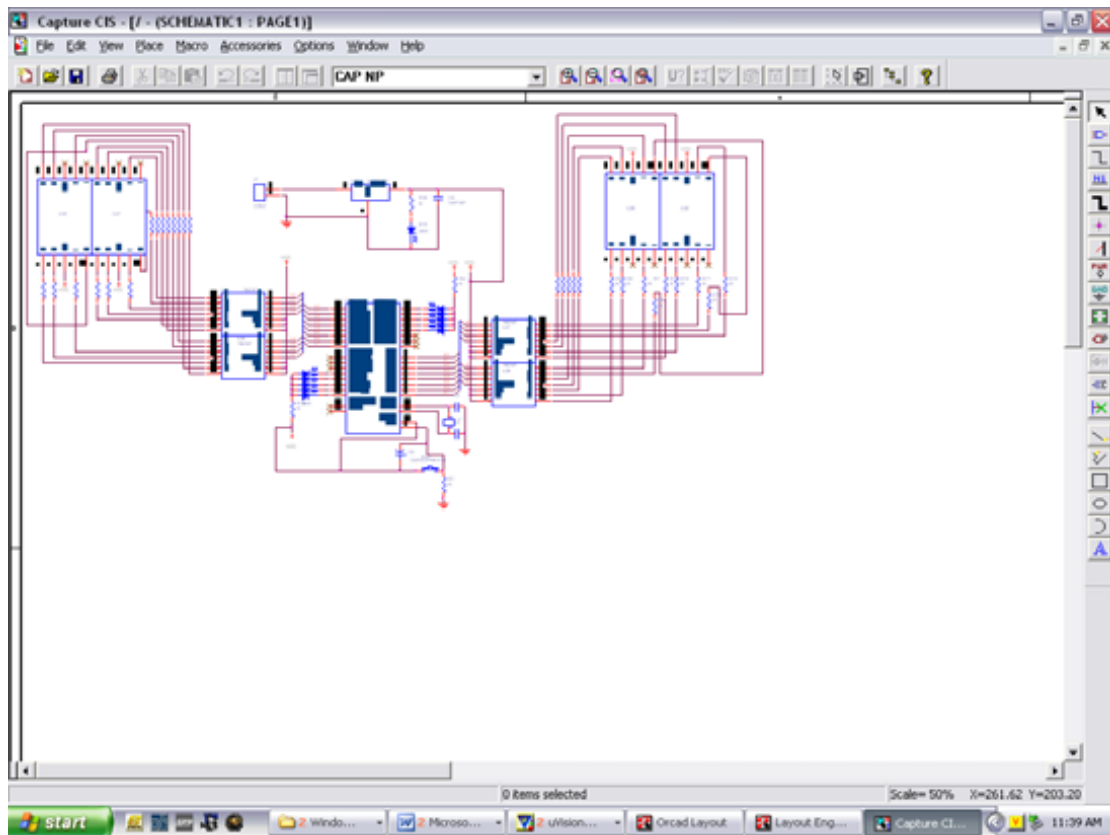
Như vậy từ địa chỉ 40h-5Bh ta sẽ thu được dãy số 27-0.

- Việc hiển thị cùng các đèn ta sẽ thực hiện tương ứng như sau:

- ✓ Đèn đỏ sáng kết hợp hiển thị trạng thái từ địa chỉ 40h-5Bh ta sẽ thu được bộ đếm 27s-0s.
- ✓ Đèn vàng sáng trễ một thời gian .
- ✓ Đèn xanh sáng kết hợp cùng hiển thị trạng thái từ địa chỉ 47h-5Bh ta sẽ thu được bộ đếm 20s-0s.

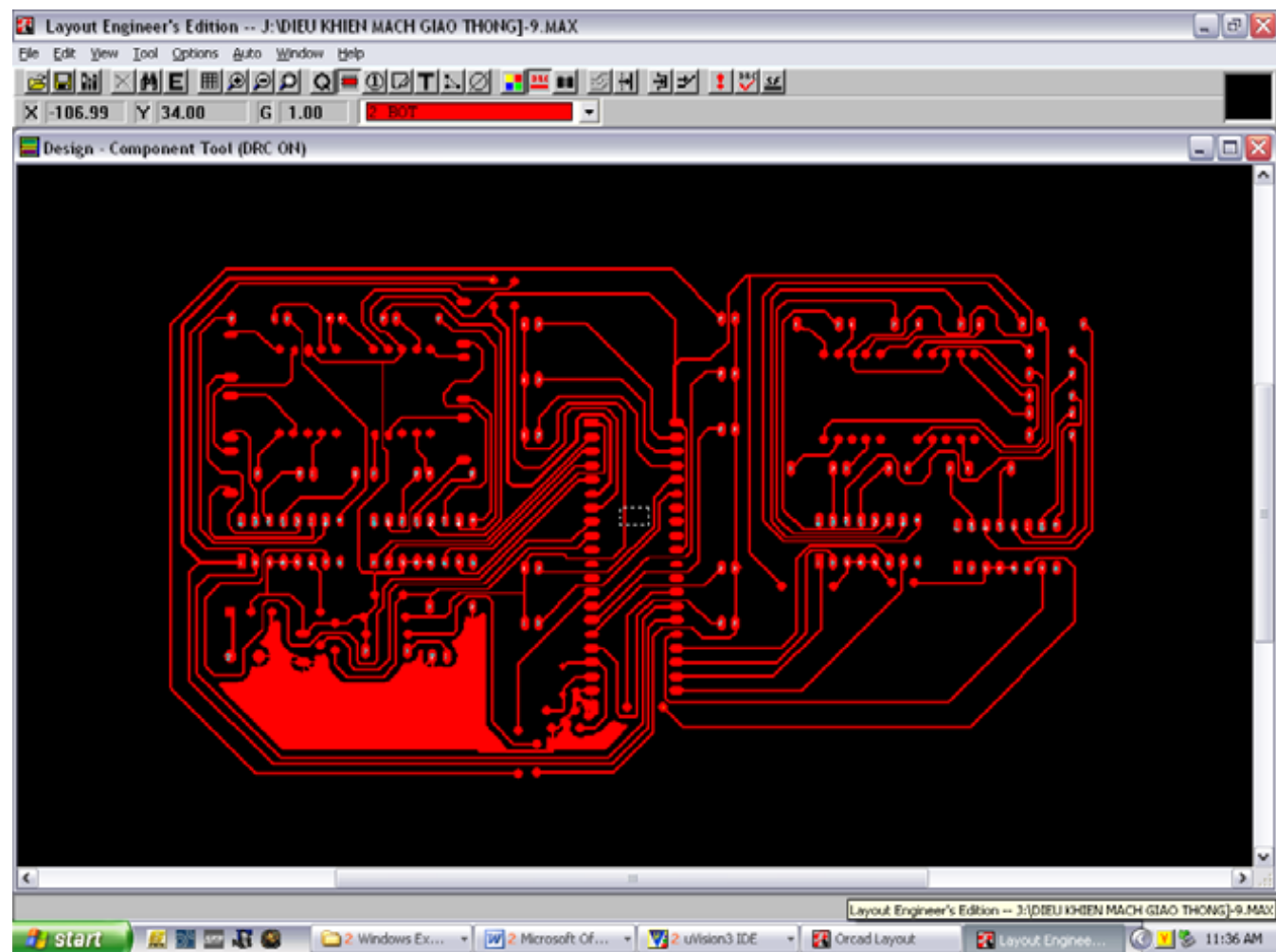
==> Cuối cùng ta được một hệ thống đèn giao thông hoàn chỉnh kết hợp giữa tạo trễ và hiển thị.

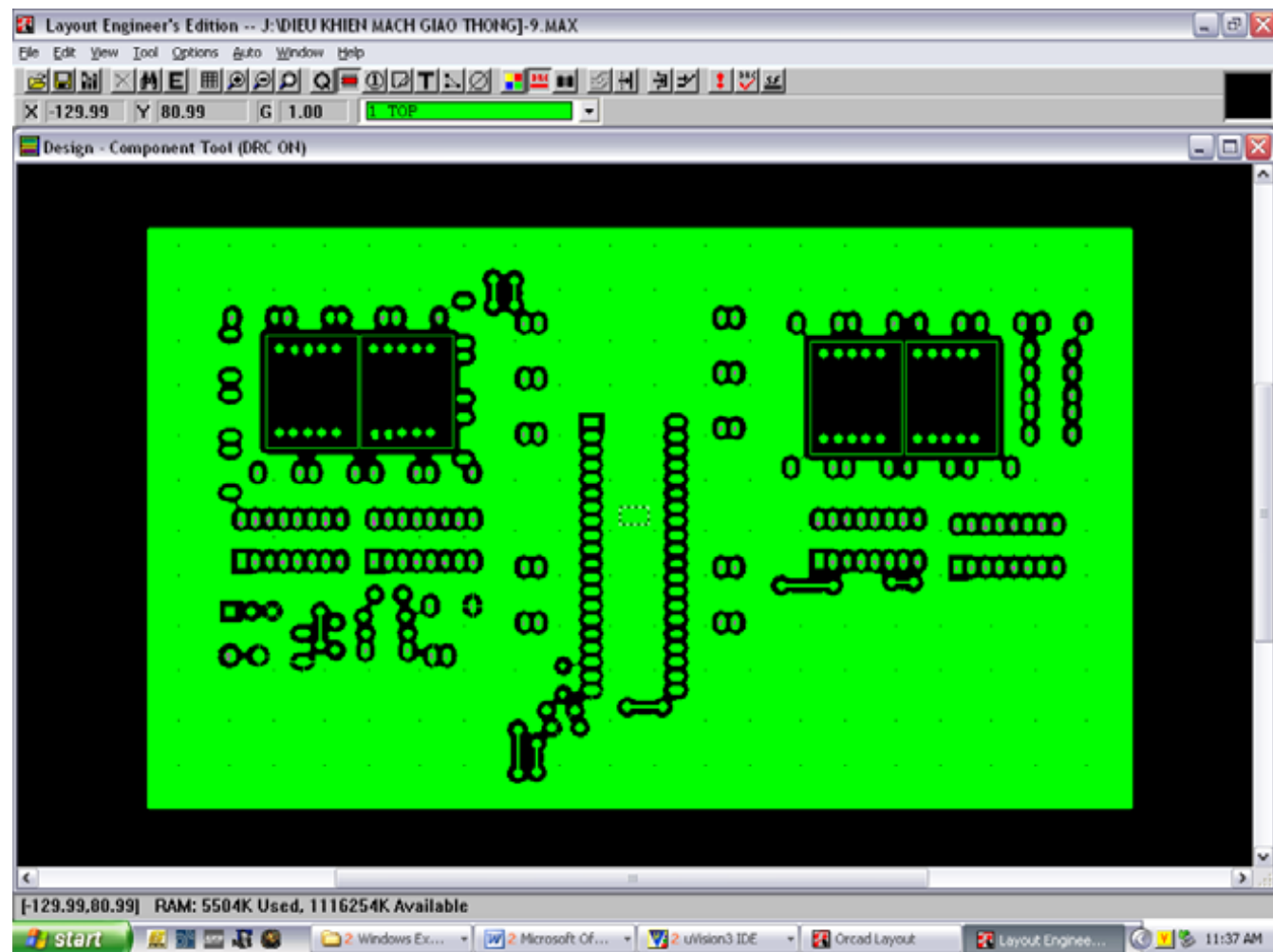
## 2. Sơ đồ nguyên lý mạch đèn giao thông:



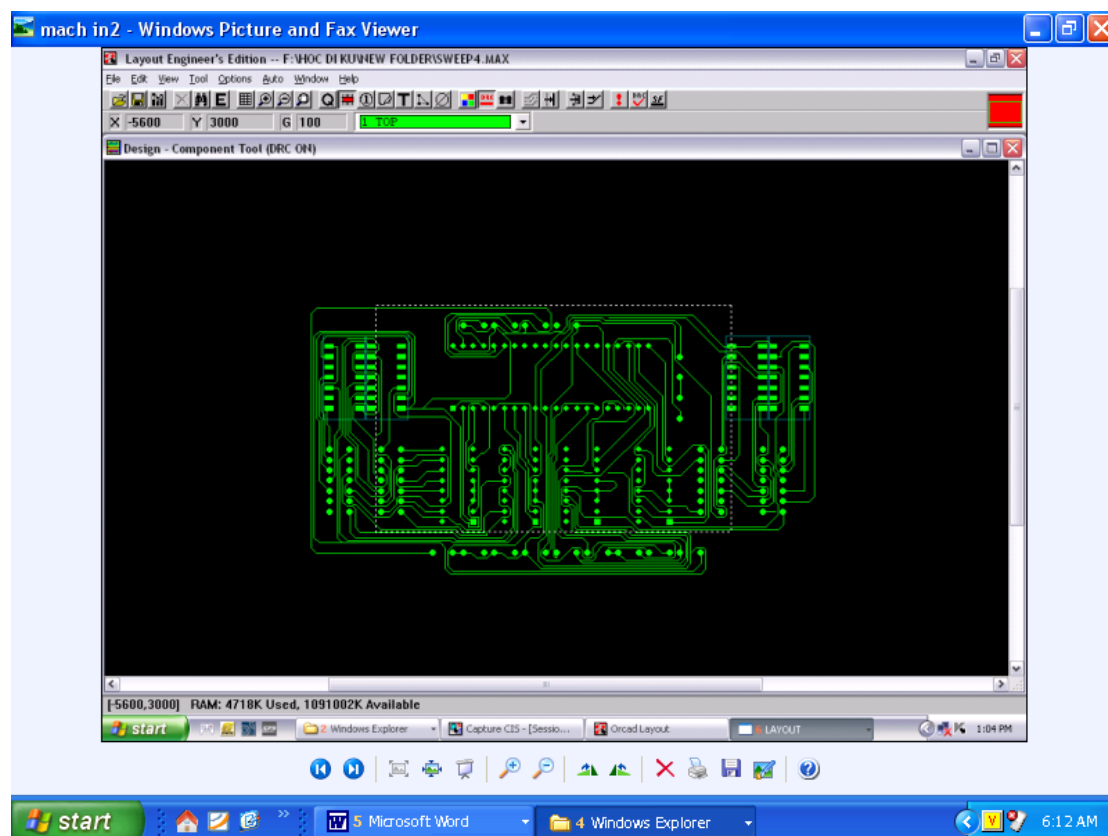
Hình 9.1: Sơ đồ nguyên lý mạch đèn giao thông







Hình 10.2: Sơ đồ mạch in



### 3.Chương trình điều khiển

```

=====
;
=====
;***** CHUONG TRINH DIEU KHIEN TIN HIEU DEN GIAO
THONG *****
;
=====
;
=====
;-----
;Phan khai bao bien ban dau
;-----
;
DO1          BIT          P0.0
;Den do huong di so 1
DO2          BIT          P3.5      ;Den do
huong di so 2
XANH1        BIT          P0.2      ;Den xanh
huong di so 1
XANH2        BIT          P3.7      ;Den xanh huong
di so 2
VANG1        BIT          P0.1      ;Den vang
huong di so 1
    
```

```

VANG2          BIT          P3.6          ;Den vang
huong di so 2

XANH_DIBO1     BIT          P0.4          ;Den xanh
cho nguoi di bo

DO_DIBO1       BIT          P0.3          ;Den do cho
nguoi di bo

XANH_DIBO2     BIT          P3.3          ;Den xanh
cho nguoi di bo

DO_DIBO2       BIT          P3.4          ;Den do cho
nguoi di bo

SET_TIME       BIT          73H           ;Dung de
thoat khoi chuong trinh DELAY

SAVE_TIME      EQU          1CH          ;Dung luu
gia tri delay

```

;\*\*\*\*\* CHUONG TRINH CHINH \*\*\*\*\*

```

ORG            0000H
LJMP MAIN                      ;Nhay den chuong trinh chinh

ORG            001BH
LJMP T1ISR                    ;Nhay den chuong trinh phuc
vu ngat 1

ORG            0030H

```

MAIN:

```

MOV            TMOD,#00010000B      ;Che do 16B Timer1 va
8B Timer2

MOV            IE,#10001000B        ;Cho phep ngat do T1

CALL KHOI_TAO

LOOP:
CALL XANH1_DO2                    ;Cho di huong di so 1,dung
huong di so 2

CALL XANH2_DO1                    ;Cho di huong di so 2,dung
huong di so 1

SJMP LOOP

```

```

;*****Chuong          trinh          ngat          dinh
thoi*****

```



**T1ISR:**

```

        CLR          TR1
    MOV  TH1,#HIGH(-1000)
    MOV  TL1,#LOW(-1000)
    SETB TR1
    CJNE R7,#00,DECDL
    MOV          R7,SAVE_TIME
    DJNZ R6,ESCDL
    CLR          TR1
    SETB SET_TIME
    SJMP ESCDL

```

**DECDL:**

```

    DEC          R7

```

**ESCDL:**

```

    RETI

```

**;\*\*\*\*\*Chương trình khai báo các biến ban đầu\*\*\*\*\***

**KHOI\_TAO:**

```

    MOV          P1,#00H           ;Hien thi so 0
    MOV          P2,#00H           ;Hien thi so 0
    MOV          P0,#00H
    MOV          P3,#00H
    MOV          DPTR,#TABLE
    RET

```

**;\*\*\*\*\*Chương trình hien thi dem lui 27S\*\*\*\*\***

**HIEN\_THI1:**

```

    CJNE R1,#00,TIEP1
    MOV          P1,#00
    SJMP ESC1

```

**TIEP1:**

```

    INC          R0
    MOV          A,R0
    MOVC         A,@A+DPTR

```

MOV P1,A

**ESC1:**

RET

\*\*\*\*\*Chuong trinh hien thi dem lui  
20S\*\*\*\*\*

**HIEN THI2:**

**CJNE R4,#00,TIEP2**

MOV P2,#00

**SJMP ESC2**

**TIEP2:**

INC R3

MOV A,R3

**MOVC**      **A,@A+DPTR**

MOV P2,A

**ESC2:**

RET

```
;*****Chuong trinh nap gia tri cho cac con
tro*****
```

**HUONG\_DI1;**Nap cac gia tri cho cac thanh ghi khi huong di so 1 duoc phep di

**MOV R0,#00 ;Con tro chuyen DPTR den vi tri so 20**

**MOV R3,#07 ;Con tro chuyen DPTR den vi tri so 27**

**MOV R1,#07 ;Con tro han che hien thi**

**MOV R4,#20 ;Con tro han che hien thi**

MOV R2,#02

RET

**HUONG\_DI2;:Nap cac gia tri cho cac thanh ghi khi huong di so 2 duoc phep di**

**MOV R0,#07 ;Con tro chuyen DPTR den vi tri so 20**

```

MOV      R3,#00          ;Con tro chuyen DPTR den vi
tri so 27
MOV      R1,#20          ;Con tro han che hien thi
MOV      R4,#7           ;Con tro han che hien thi
MOV      R2,#02
RET

```

**;\*\*\*\*\*Chương trình điều khiển đèn xanh do\*\*\*\*\***

**XANH1\_DO2:;Hien thi den xanh ben huong di so 1 va hien thi den do o huong di so 2**

```

CALL HUONG_DI1
CLR      XANH1           ;Hien thi den xanh
huong di so 1
CLR      DO2            ;Hien thi den do huong
di so 2
CLR      DO_DIBO1       ;Hien thi den do cua nguoi di
bo
CLR      XANH_DIBO2     ;Hien thi den xanh cua
nguoi di bo

```

**BACK1:**

```

CALL HIEN_THI1          ;Cho hien thi thoi gian giam
CALL HIEN_THI2          ;Cho hien thi thoi gian giam
CALL DELAY1S
CJNE R4,#00,DEC11       ;Khi hien thi du so thi khong giam nua
CLR      VANG1          ;Hien thi den vang
SETB XANH1              ;Tat den xanh
CJNE R2,#00,DEC31       ;Doi them 4S nua thi nhap nhay den

```

```

CPL      XANH_DIBO2     ;Nhap nhay den xanh
CPL      DO_DIBO1       ;Nhap nhay den do

```

**SKIP1:**

```

CJNE R1,#00,DEC21       ;Khi hien thi du so thi khong giam nua
SETB XANH1              ;Tat den xanh huong di so 1
SETB DO2                ;Tat den do huong di so 2
SETB DO_DIBO1           ;Tat den do cua nguoi di bo
SETB VANG1              ;Tat den vang 1
SETB XANH_DIBO2         ;Tat den xanh cua nguoi di bo

```

```

RET
DEC11:
    DEC        R4
    SJMP BACK1
DEC21:
    DEC        R1
    SJMP BACK1
DEC31:
    DEC        R2
    SJMP SKIP1

XANH2_DO1;;Hien thi den xanh ben huong di so 2 va hien thi den do o
huong di so 1
    CALL HUONG_DI2
    CLR        XANH2                ;Hien thi den xanh
huong di so 2
    CLR        DO1                  ;Hien thi den do huong
di so 1
    CLR        DO_DIBO2            ;Hien thi den do cua nguoi di
bo
    CLR        XANH_DIBO1          ;Hien thi den xanh cua
nguoi di bo
BACK2:
    CALL HIEN_THI1                ;Cho hien thi thoi gian giam
    CALL HIEN_THI2                ;Cho hien thi thoi gian giam
    CALL DELAY1S
    CJNE R1,#00,DEC22              ;Khi hien thi du so thi khong giam nua

    CLR        VANG2                ;Hien thi den vang
    SETB XANH2                    ;Tat den xanh
    CJNE R2,#00,DEC32              ;Doi them 4S nua thi nhap nhay den
    CPL        XANH_DIBO1          ;Nhap nhay den xanh
    CPL        DO_DIBO2            ;Nhap nhay den do
SKIP2:
    CJNE R4,#00,DEC12              ;Khi hien thi du so thi khong giam nua
    SETB XANH2                    ;Tat den xanh huong di so 1
    SETB DO1                      ;Tat den do huong di so 2
    SETB VANG2                    ;Tat den vang

```

```

SETB DO_DIBO2                ;Tat den do cua nguoi di bo
SETB XANH_DIBO1              ;Tat den xanh cua nguoi di bo
RET
DEC12:
    DEC        R4
    SJMP BACK2
DEC22:
    DEC        R1
    SJMP BACK2
DEC32:
    DEC        R2
    SJMP SKIP2
    
```

```

;-----
;***** CAC KHOANG THOI GIAN TRE
;*****
;-----
    
```

```

;***** CHUONG TRINH TRE 1S *****
    
```

```

DELAY1s:
    PUSH 07H
    PUSH 06H
    CLR        SET_TIME
    MOV        SAVE_TIME,#100
    MOV R7,#100
    MOV        R6,#10
    SETB TR1
    SETB TF1
    JNB        SET_TIME,$
    POP        06
    POP        07
    RET
    
```

**TABLE:;Hien thi tu 27-0**

```

DB        000H,0027H,0026H,0025H,0024H,0023H,0022H,0021H
    
```

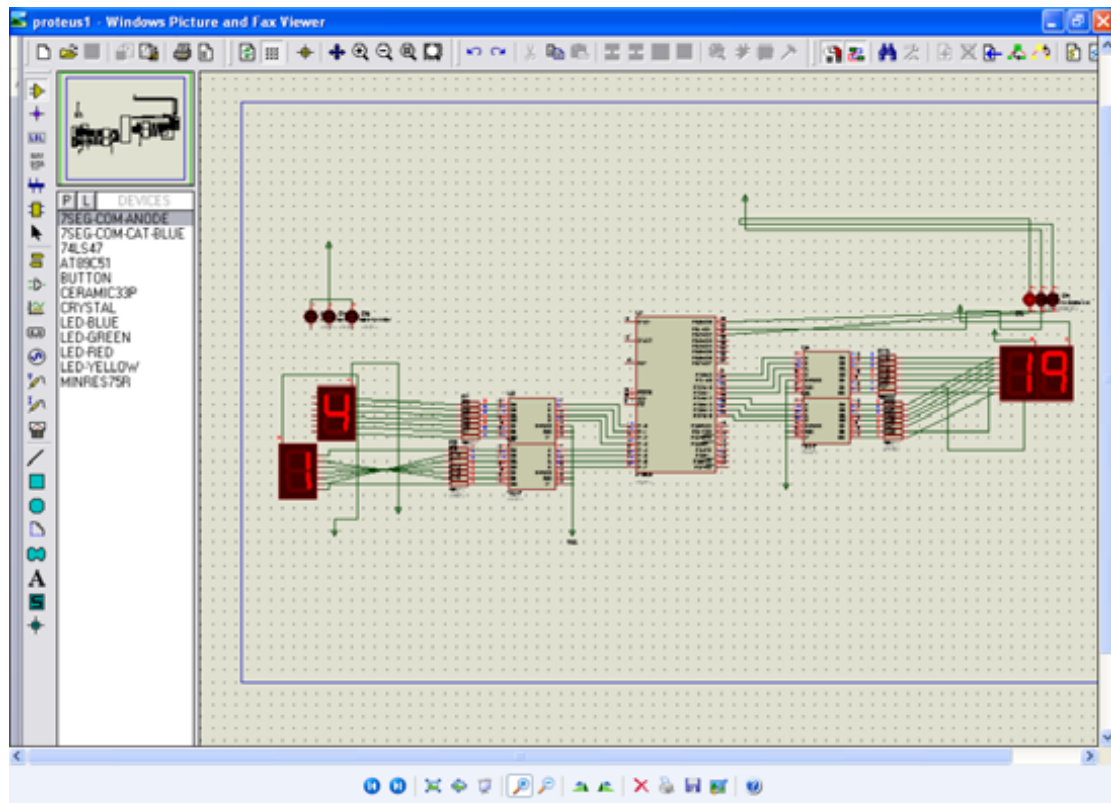
DB                    020H,019H,018H,017H,016H,015H,014H,013H,012H,011H

DB

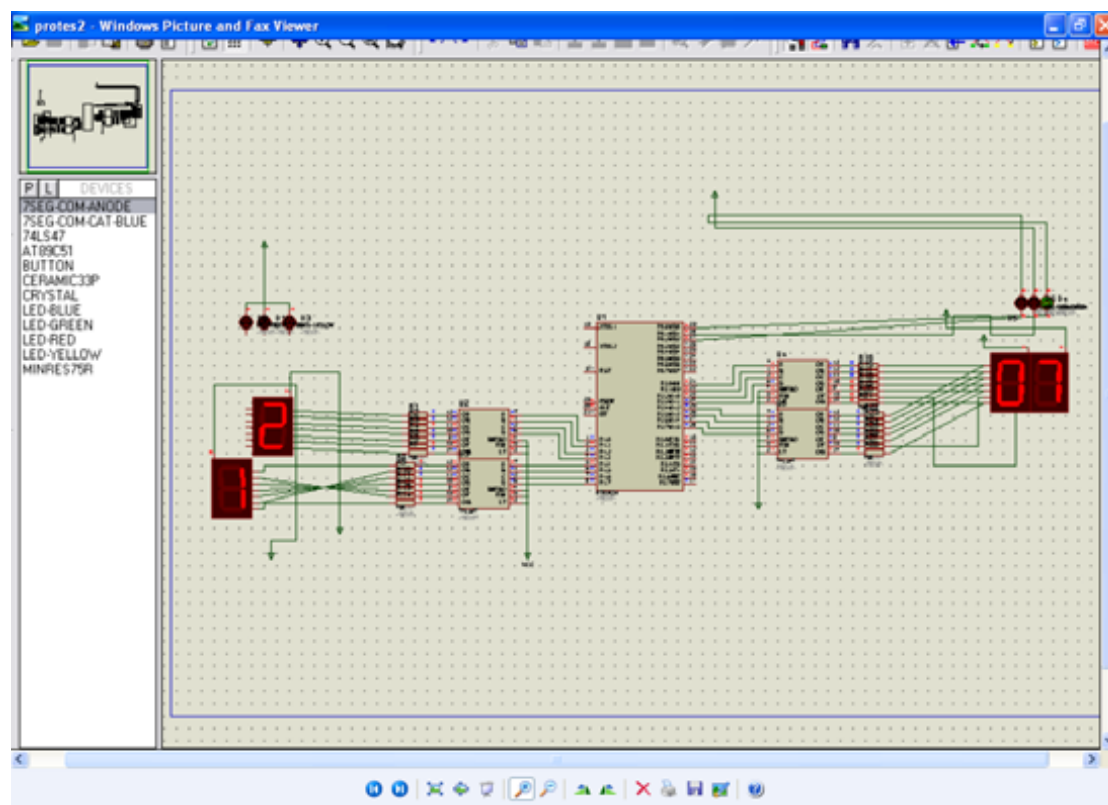
010H,009H,008H,007H,006H,005H,004H,003H,002H,001H,000H

RET

END.



Hình 11.1: Mạch chạy mô phỏng đèn giao thông



Hình 11.2: Mạch chạy mô phỏng đèn giao thông

## V. MỞ RỘNG

Ngoài hệ thống điều khiển đèn trên hai tuyến:

-Và ở những thành phố lớn, trên những nút giao thông trọng điểm lúc này không chỉ còn là sự giao nhau của hai làn đường mà là rất nhiều làn đường khác nhau, cùng với số phương tiện lưu thông lớn. Chính vì thế mà ở đó ngoài hệ thống đèn điều khiển chính còn có xen kẽ thêm một hệ thống đèn dùng để phân làn đường cho phép các phương tiện đang dừng đèn đỏ được phép rẽ theo một hướng ưu tiên nào đó.

1). Về hệ thống phân làn đường, và chuyển pha (Một tuyến có thể có nhiều pha khác nhau):

-Khi tuyến một đang lưu thông, trên tuyến còn lại đèn đỏ sẽ hiển thị và buộc các phương tiện phải dừng.

-Tuy nhiên, ta có thể thiết kế thêm một đèn báo hiệu cho phép các phương tiện trên tuyến đang dừng có thể rẽ cùng với chiều của các phương tiện đang lưu thông.

-Điều này sẽ giúp giảm bớt số lượng các phương tiện phải dừng chờ, góp phần giải tỏa ách tắc giao thông (nhất là đối với các thành phố trọng điểm, ở đó ách tắc giao thông thường xuyên xảy ra do sự lưu thông quá lớn của các phương tiện trong giờ cao điểm)

-Đèn báo hiệu này sẽ là đèn xanh có hình mũi tên chỉ hướng được phép rẽ. Giả sử đèn báo hiệu rẽ này là của tuyến 2 thì nó sẽ sáng sau đèn xanh của tuyến 1 khoảng 10s. Lúc này đèn đỏ của tuyến 2 vẫn sáng và các phương tiện không được phép đi thẳng mà chỉ được phép rẽ.

Trên tuyến còn lại đèn báo hiệu rẽ cũng hoạt động tương tự.

2). Trong ngày có nhiều khoảng thời gian (Các bộ điều khiển hiện nay thường chọn 10 khoảng), mỗi khoảng có chiến lược riêng. Ví dụ: Ban đêm, nhấp nháy đèn vàng 2 s: 0-0-2-0-0-0-2-0. Buổi sáng sớm có 1 chiến lược, lúc cao điểm có một chiến lược. Người ta có thể gán cho 10 khoảng thời gian các chiến lược khác nhau, được đánh số. Ví dụ: Thứ Hai: 1,1,2,3,2,3,3,2,2,1.

3). Trong một tuần, có 7 ngày có thể có 7 tập hợp các chiến lược khác nhau. Ví dụ: Đầu tuần, người đi vào thành phố nhiều, cuối tuần ...

4). Một năm có 52 tuần, có thể khác nhau theo thống kê. Ví dụ: Mùa hè xe chạy nhiều hơn mùa đông.

5). Nếu không có thay đổi, mặc nhiên các ngày sẽ giống nhau.

6). Như vậy, bộ xử lý phải có đồng hồ, xác định được thời điểm hoạt động (giờ, phút, ngày tháng) và nạp bộ thông số chiến lược tương ứng. Ví dụ từ 22 h - 24 h, nạp nhấp nháy đèn vàng 1 s, 0h - 5h30, nhấp nháy đèn vàng 2 s, 5h30 - 6h30, chiến lược 3, .v.v...

7). Với ngã tư hiện đại, sẽ có thiết bị xác định lưu lượng xe để điều chỉnh thời gian thông xe (đèn xanh). Thông thường xử dụng các vòng từ chôn dưới đường. Nếu trong chiến lược là 25 s Xanh, vòng từ xác định chỉ 15 s có xe, thì chu kì tiếp theo (Hoặc theo dõi trong 2,3 chu kì), thời gian Xanh được giảm xuống theo bước xác định để tăng thời gian cho tuyến kia...

8). Các tủ điều khiển có công tắc chuyển sang chế độ bằng tay.

9). Nếu tạo làn xanh, các tủ phải có khả năng kết nối (cáp quang...)

Trên đây là các vấn đề chính để thiết kế bộ điều khiển đèn. Các vấn đề



khác trong tầm tay dân điện tử. (Ví dụ như đếm lùi, bật tắt đèn). Hiện nay, các đèn đếm lùi thường chế tạo từ các LED rời, ít sử dụng LED 7 thanh, vì yêu cầu kích cỡ chữ số và độ sáng.