

**www.mientayvn.com**

Khi đọc qua tài liệu này, nếu phát hiện sai sót hoặc nội dung kém chất lượng xin hãy thông báo để chúng tôi sửa chữa hoặc thay thế bằng một tài liệu cùng chủ đề của tác giả khác.

Bạn có thể tham khảo nguồn tài liệu được dịch từ tiếng Anh tại đây:

[http://mientayvn.com/Tai\\_lieu\\_da\\_dich.html](http://mientayvn.com/Tai_lieu_da_dich.html)

Thông tin liên hệ:

Yahoo mail: [thanhlam1910\\_2006@yahoo.com](mailto:thanhlam1910_2006@yahoo.com)

Gmail: [frbwrthes@gmail.com](mailto:frbwrthes@gmail.com)

**Theo yêu cầu của khách hàng, trong một năm qua, chúng tôi đã dịch qua 16 môn học, 34 cuốn sách, 43 bài báo, 5 sổ tay (chưa tính các tài liệu từ năm 2010 trở về trước) Xem ở đây**

**DỊCH VỤ  
DỊCH  
TIẾNG  
ANH  
CHUYÊN  
NGÀNH  
NHANH  
NHẤT VÀ  
CHÍNH  
XÁC  
NHẤT**

Chỉ sau một lần liên lạc, việc dịch được tiến hành

Giá cả: có thể giảm đến 10 nghìn/1 trang

Chất lượng: Tạo dựng niềm tin cho khách hàng bằng công nghệ 1. Bạn thấy được toàn bộ bản dịch; 2. Bạn đánh giá chất lượng. 3. Bạn quyết định thanh toán.

**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**  
**KHOA ĐIỆN TỬ VIỄN THÔNG**



**BÁO CÁO BÀI TẬP LỚN**  
**MÔN KỸ THUẬT VI XỬ LÝ**

**Đề tài: THIẾT KẾ MẠCH QUANG BÁO**

**Giáo viên hướng dẫn: Phạm Ngọc Nam**

**Sinh viên thực hiện: Ngô Hoàng Anh**

**Dương Trung Huyền**

**Nguyễn Xuân Tiến**

**Đặng Hữu Tùng (TN)**

**Nguyễn Trung Thu**

**Lớp:**

**ĐT9 – K47**



**11/2005**

## **Lời nói đầu**

Song hành với sự phát triển của ngành khoa học máy tính trong hơn 60 năm qua, công nghệ điện tử đã có những bước tiến vượt bậc với khả năng tích hợp ngày càng cao của vi mạch từ đó giúp tăng tốc tốc độ xử lý, nâng cao độ tin cậy và giảm giá thành sản phẩm. Từ những ứng dụng ban đầu chủ yếu trong lĩnh vực quân sự và máy tính của các bộ vi xử lý, ngày nay, sự ra đời của các họ vi điều khiển với việc tích hợp các khối chức năng trên một IC, các vi xử lý chuyên dụng, cùng với thể mạnh vốn có của các bộ vi xử lý đa năng đã giúp cho việc ứng dụng kỹ thuật vi xử lý vào trong các hệ thống phi máy tính trở nên đơn giản hơn, mở rộng đối tượng ứng dụng các thành quả của ngành công nghiệp điện tử hiện đại này. Ta có thể thấy ứng dụng của chúng trong các hệ thống máy tính lớn, các hệ thống viễn thông cho đến các sản phẩm quen thuộc như máy giặt, điều hòa, đèn giao thông,...

Trong khuôn khổ bài tập lớn này, với mục đích tìm hiểu ứng dụng thực tế của kỹ thuật vi xử lý, nhóm chúng em lựa chọn đề tài thiết kế mạch quang báo ứng dụng kỹ thuật vi xử lý.

## MỤC LỤC

|  |           |
|--|-----------|
| Lời nói đầu  |           |
| <b>I. Tổng quan</b>                                  | <b>4</b>  |
| <b>II. Sơ đồ khối</b>                                | <b>5</b>  |
| <b>III. Sơ đồ nguyên lý</b>                          | <b>5</b>  |
| <b>IV. Các IC và linh kiện sử dụng trong mạch</b>    | <b>5</b>  |
| <b>1. AT89C51</b>                                    | <b>5</b>  |
| <b>2. Thanh ghi dịch 74HC595</b>                     | <b>21</b> |
| <b>3. ULN2803</b>                                    | <b>23</b> |
| <b>4. LED ma trận 8x8</b>                            | <b>24</b> |
| <b>V. Nguyên lý và tác dụng linh kiện</b>            | <b>25</b> |
| <b>VI. Chương trình</b>                              | <b>26</b> |
| <b>VII. Nhận xét, kết luận, hướng mở rộng đề tài</b> | <b>29</b> |
| <b>Tài liệu tham khảo</b>                            | <b>31</b> |

## **I. Tổng quan**

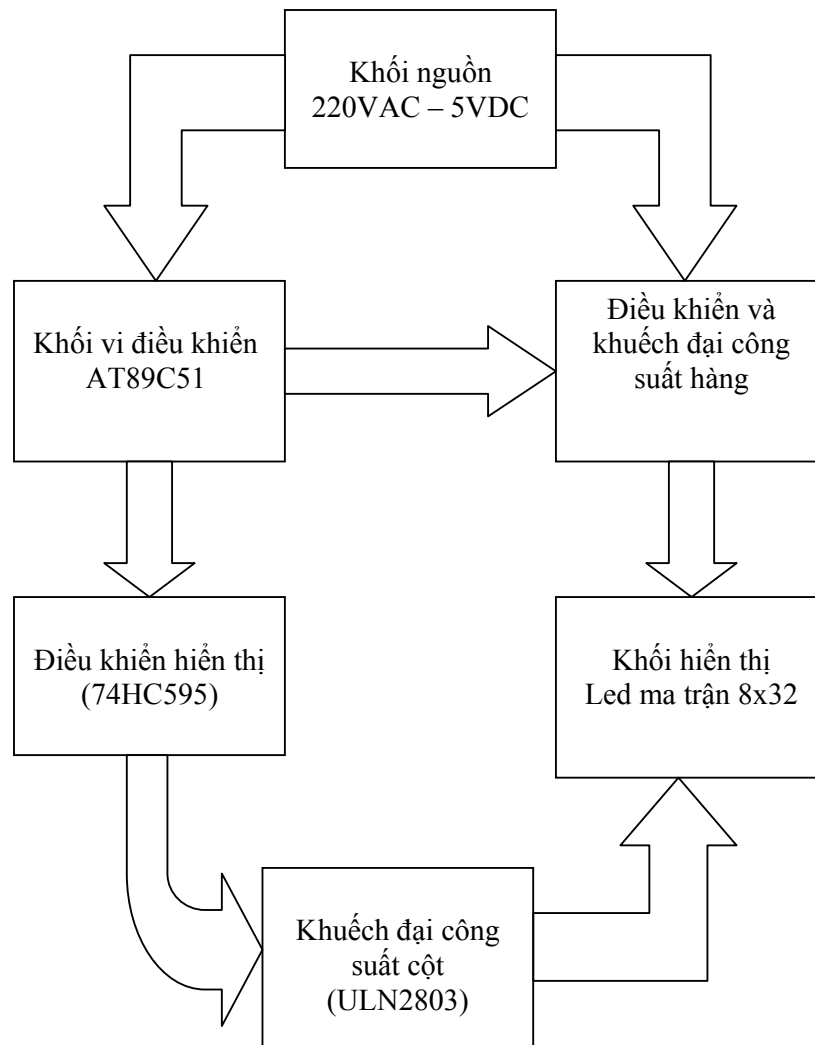
### ♦ Giới thiệu sản phẩm:

Mạch quang báo nhóm thực hiện là mạch có chức năng hiển thị nội dung trên ma trận điểm. Nội dung này có thể dịch chuyển từ phải sang trái. Nội dung cần hiển thị được nạp trước vào trong bộ nhớ của vi điều khiển trong quá trình nạp chương trình cho vi điều khiển. Mỗi khi cần thay đổi nội dung hiển thị cần nạp lại chương trình cho vi điều khiển.

### ♦ Lựa chọn các linh kiện:

Để thực hiện một sản phẩm như trên ở quy mô nhỏ ta có thể sử dụng các họ vi điều khiển khác nhau như AVR, PIC, 8051 hay vi xử lý đa năng như 8086. Tuy nhiên, để tiết kiệm chi phí cũng như xét trên khả năng mua các chip trên thị trường, tài liệu nghiên cứu về chúng, bộ Kit phát triển, nhóm em đã lựa chọn AT89C51 làm vi điều khiển cho mạch quang báo này. Ngoài ra, các linh kiện khác hoàn toàn dễ kiếm trên thị trường hiện nay.

## II. Sơ đồ khối



## III. Sơ đồ nguyên lý (kèm theo): file mach quang bao.pdf

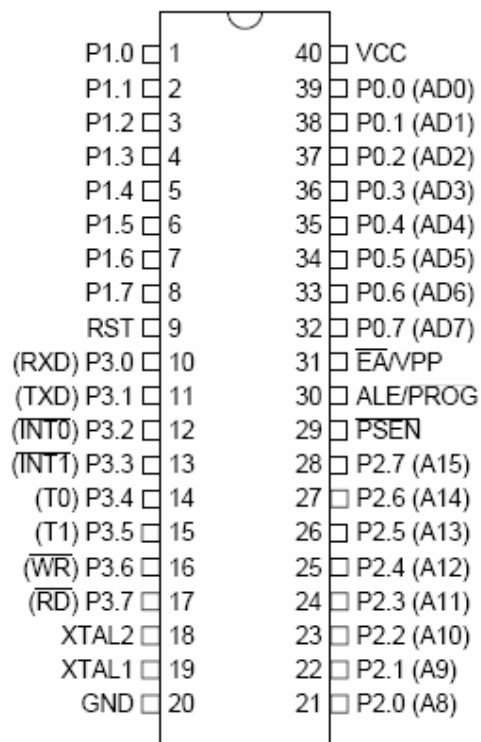
## IV. Các IC và các linh kiện sử dụng trong mạch:

### 1. AT89C51

- Tương thích với các sản phẩm thuộc họ vi điều khiển MCS-51
- Có 4 Kbyte bộ nhớ flash, khả năng ghi/xóa 1000 lần

- Làm việc với tần số 0Hz – 24MHz
- Khóa bộ nhớ chương trình 3 mức
- 128 x 8 bit RAM nội
- 32 đường xuất/nhập lập trình được
- 2 bộ định thời/đếm 16 bit
- 6 nguồn ngắt
- Kênh nối tiếp lập trình được
- Chế độ tiêu thụ ít năng lượng

#### a/ Các chân của IC 89C51



- **Vcc:** nối với điện áp nguồn
- **GND:** nối đất
- **Port 0:** cổng xuất/nhập 8 bit. Khi làm cổng xuất, mỗi chân có thể ghép nối với 8 đầu vào TTL. Khi các chân ở mức 1, các chân này có thể được dùng làm đầu vào trở kháng cao. Ngoài ra, khi truy cập tới chương trình và dữ liệu bên ngoài, port 0 có thể được sử dụng làm bus địa chỉ thấp/ dữ

liệu đa hợp. Port này có thể dùng để nhận chương trình nạp vào Flash hoặc kiểm tra

- **Port 1:** cổng xuất/nhập 8 bit, các bộ đệm ra có thể ghép nối với 4 đầu vào TTL. Các chân này có thể làm đầu vào khi tất cả được thiết lập ở mức 1. Port 1 nhận các byte địa chỉ thấp trong quá trình ghi chương trình và kiểm tra.

- **Port 2:** cổng xuất/nhập 8 bit, các bộ đệm ra có thể ghép nối với 4 đầu vào TTL. Các chân này có thể làm đầu vào khi tất cả được thiết lập ở mức 1. Port 2 truyền byte địa chỉ cao của bus địa chỉ với các thiết kế có bộ nhớ chương trình ngoài hoặc các thiết kế có nhiều hơn 256 byte bộ nhớ dữ liệu ngoài. Port 2 cũng nhận các bit địa chỉ cao và một vài tín hiệu điều khiển trong quá trình nạp chương trình và kiểm tra.

- **Port 3:** cổng xuất/nhập 8 bit, các bộ đệm ra có thể ghép nối với 4 đầu vào TTL. Khi tất cả các chân ở mức 1, Port 3 thực hiện nhận dữ liệu. Ngoài ra, Port còn phục vụ một số chức năng đặc biệt của AT89C51 như:

| Bit  | Tên  | Địa chỉ bit | Chức năng                           |
|------|------|-------------|-------------------------------------|
| P3.0 | RXD  | B0H         | Chân nhận dữ liệu của port nối tiếp |
| P3.1 | RXT  | B1H         | Chân phát dữ liệu của port nối tiếp |
| P3.2 | INT0 | B2H         | Ngõ vào ngắt ngoài 0                |
| P3.3 | INT1 | B3H         | Ngõ vào ngắt ngoài 1                |
| P3.4 | T0   | B4H         | Ngõ vào của bộ định thời/đếm 0      |
| P3.5 | T1   | B5H         | Ngõ vào của bộ định thời/đếm 1      |
| P3.6 | WR   | B6H         | Điều khiển ghi bộ nhớ ngoài         |
| P3.7 | RD   | B7H         | Điều khiển đọc bộ nhớ ngoài         |

Port 3 cũng nhận một số tín hiệu điều khiển trong quá trình nạp chương trình và kiểm tra

- **RST:** đầu vào reset. Khi chân này ở mức cao trong 2 chu kỳ máy khi oscillator đang hoạt động thì IC sẽ được reset



- **$\overline{\text{ALE}}/\overline{\text{PROG}}$** : chân cho phép chốt địa chỉ đưa ra xung để chốt byte địa chỉ thấp trong quá trình truy cập bộ nhớ ngoài. Chân này cũng đóng vai trò đầu vào xung chương trình  $\overline{\text{PROG}}$  trong quá trình nạp chương trình. Ở điều kiện bình thường, tín hiệu phát ra từ chân này có tần số bằng 1/6 tần số của mạch dao động trong chip và có thể được sử dụng làm xung clock
- **$\overline{\text{PSEN}}$** : chân cho phép bộ nhớ chương trình. Khi AT89C51 thực thi các lệnh từ bộ nhớ chương trình ngoài, chân này được tích cực 2 lần trong mỗi chu kỳ máy
- **$\overline{\text{EA}}/\text{Vpp}$** : chân này phải nối đất để IC có thể tìm mã từ các ô nhớ chương trình ngoài bắt đầu từ địa chỉ 0000H đến FFFFH (64Kbyte). Để IC tìm và thực thi các lệnh của chương trình trong bộ nhớ nội, chân này cần nối với Vcc. Chân này cũng nhận điện áp cho phép ghi chương trình 12V trong quá trình nạp chương trình.
- **XTAL1**: đầu vào của bộ khuếch đại dao động đảo
- **XTAL2**: đầu ra của bộ khuếch đại dao động đảo

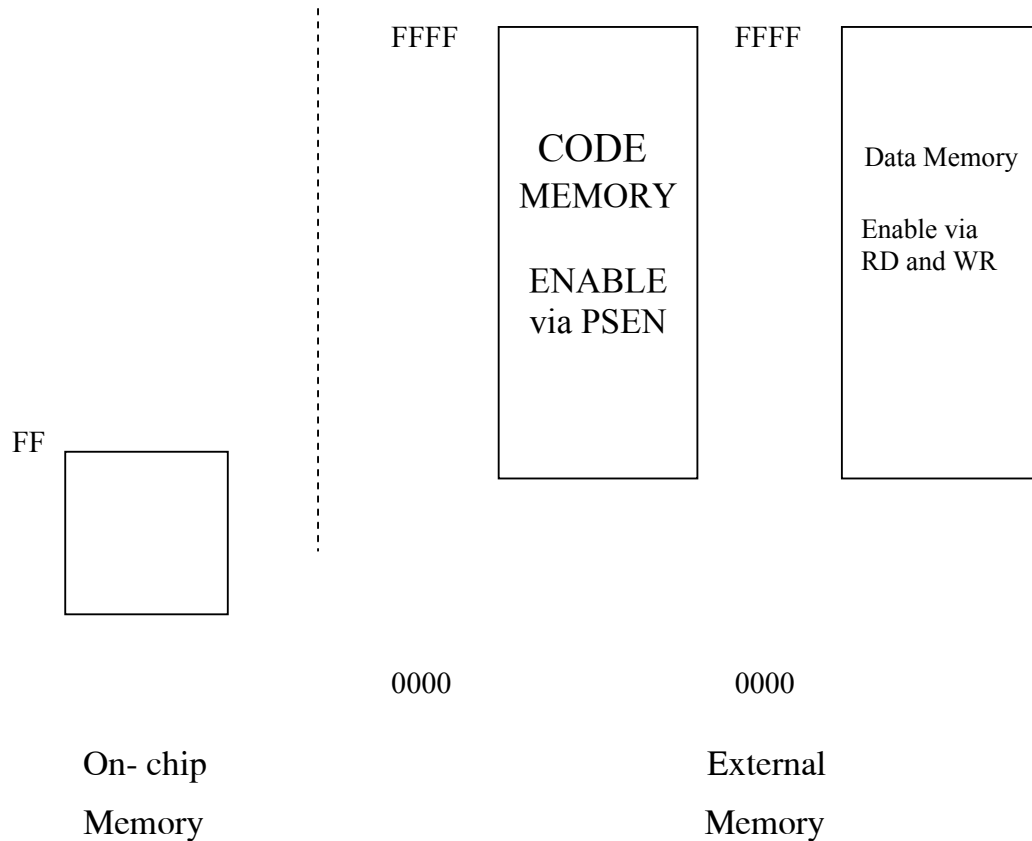
#### **b/ Tổ chức bộ nhớ**

Bộ nhớ bên trong chip bao gồm ROM, RAM và EPROM. RAM trên chip bao gồm vùng RAM đa chức năng, vùng RAM với từng bit được định địa chỉ, các dây thanh ghi (bank) và các thanh ghi chức năng đặc biệt.

Có 2 đặc tính đáng lưu ý:

- + Các thanh ghi và các port I/O được định địa chỉ theo kiểu ánh xạ bộ nhớ và được truy xuất như một vị trí nhớ trong bộ nhớ.
- + Vùng track thường trú trong RAM trên chip thay vì ở trong RAM ngoài như đối với các bộ vi xử lý.

Chi tiết bộ nhớ dữ liệu trên chip:



### ***Tóm tắt không gian nhớ của chip***

**\* Vùng RAM đa mục đích: Có 80 byte, địa chỉ từ 30H đến 7FH**

Bất cứ vị trí nào trong vùng RAM ta đều có thể truy xuất tự do bằng cách sử dụng định địa chỉ trực tiếp hoặc gián tiếp.

Ví dụ:

+ Kiểu định địa chỉ trực tiếp:

MOV A, 5FH ;Đọc nội dung tại địa chỉ 5FH của RAM  
;vào thanh chứa A.

+ Kiểu định địa chỉ gián tiếp: (Qua các thanh ghi R0,R1)

MOV R0, #5FH ; Di chuyển giá trị 5FH vào thanh ghi R0  
MOV A, @R0 ; Di chuyển dữ liệu trữ tới R0 vào thanh chứa A

**\* Vùng RAM định địa chỉ**

Chip 89C51 chứa 210 vị trí định địa chỉ trong đó có 128 byte chứa trong các byte ở địa chỉ 20H đến 2FH (16 byte x 8 = 128 bits), phần còn lại chứa trong các thanh ghi chức năng đặc biệt.

Công dụng: + Truy xuất các bit riêng rẽ thông qua các phần mềm.  
+ Các port có thể định địa chỉ từng bit, làm đơn giản việc giao tiếp bằng phần mềm với các thiết bị xuất nhập đơn bit.

Ví dụ: + Set bit trực tiếp:

SETB 67H; lệnh làm nhiệm vụ set bit 67H bằng 1

+ Hoặc ta có thể sử dụng lệnh sau để set bit 67H là bit lớn nhất của byte 2CH:

MOV A,2CH ; Đọc cả byte

ORL A,#10000000B ;Tác dụng set bit

MOV 2CH,A ; Ghi trở lại cả byte

|    |                     |    |    |    |    |    |    |    |
|----|---------------------|----|----|----|----|----|----|----|
| FF | General purpose RAM |    |    |    |    |    |    |    |
|    | 7F                  | 7E | 7D | 7C | 7B | 7A | 79 | 78 |
| 30 | 77                  | 76 | 75 | 74 | 73 | 72 | 71 | 70 |
| 2F | 6F                  | 6E | 6D | 6C | 6B | 6A | 69 | 68 |
|    | 67                  | 66 | 65 | 64 | 63 | 62 | 61 | 60 |
|    | 5F                  | 5E | 5D | 5C | 5B | 5A | 59 | 58 |
|    | 57                  | 56 | 55 | 54 | 53 | 52 | 51 | 50 |
|    | 4F                  | 4E | 4D | 4C | 4B | 4A | 49 | 48 |
|    | 47                  | 46 | 45 | 44 | 43 | 42 | 41 | 40 |
|    | 3F                  | 3E | 3D | 3C | 3B | 3A | 39 | 38 |
|    | 37                  | 36 | 35 | 34 | 33 | 32 | 31 | 30 |
|    | 2F                  | 2E | 2D | 2C | 2B | 2A | 29 | 28 |
|    | 27                  | 26 | 25 | 24 | 23 | 22 | 21 | 20 |
|    | 1F                  | 1E | 1D | 1C | 1B | 1A | 19 | 18 |
|    | 17                  | 16 | 15 | 14 | 13 | 12 | 11 | 10 |

|          |                                 |    |    |    |    |    |    |    |
|----------|---------------------------------|----|----|----|----|----|----|----|
|          | 0F                              | 0E | 0D | 0C | 0B | 0A | 09 | 08 |
|          | 07                              | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|          | BANK 3                          |    |    |    |    |    |    |    |
|          | BANK 2                          |    |    |    |    |    |    |    |
|          | BANK 1                          |    |    |    |    |    |    |    |
| 20<br>1F | Default register bank for R0-R7 |    |    |    |    |    |    |    |
| 00       |                                 |    |    |    |    |    |    |    |

### \* Các dãy thanh ghi:

Có địa chỉ từ 00H đến 1FH, 32 vị trí thấp nhất của bộ nhớ nội chứa các dãy thanh ghi. Các lệnh của 89C51 hỗ trợ 8 thanh ghi từ R0 đến R7 (mặc định thuộc bank 0 sau khi reset hệ thống)

|    |    |    |    |    |    |    |    |    |     |
|----|----|----|----|----|----|----|----|----|-----|
| 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 | P3  |
|    |    |    |    |    |    |    |    |    |     |
| A8 | AF | -  | -  | AC | AB | AA | A9 | A8 | IE  |
|    |    |    |    |    |    |    |    |    |     |
| A0 |    |    |    |    |    |    |    |    |     |

|    |                     |    |    |    |    |    |    |    |      |
|----|---------------------|----|----|----|----|----|----|----|------|
| A0 | A7                  | A6 | A5 | A4 | A3 | A2 | A1 | A0 | P2   |
|    |                     |    |    |    |    |    |    |    |      |
|    | Not bit addressable |    |    |    |    |    |    |    | SBUF |
| 99 | 9F                  | 9E | 9D | 9C | 9B | 9A | 99 | 98 | SCON |
| 98 |                     |    |    |    |    |    |    |    |      |
|    | 97                  | 96 | 95 | 94 | 93 | 92 | 91 | 90 | P1   |
| 90 |                     |    |    |    |    |    |    |    |      |
|    | Not bit addressable |    |    |    |    |    |    |    | TH1  |
|    | Not bit addressable |    |    |    |    |    |    |    | TH0  |
| 8D | Not bit addressable |    |    |    |    |    |    |    |      |
| 8C | Not bit addressable |    |    |    |    |    |    |    | TL1  |
| 8B | Not bit addressable |    |    |    |    |    |    |    | TL0  |
| 8A | Not bit addressable |    |    |    |    |    |    |    | TMOD |
| 89 |                     |    |    |    |    |    |    |    |      |
| 88 | 8F                  | 8E | 8D | 8C | 8B | 8A | 89 | 88 | TCON |
| 87 | Not bit addressable |    |    |    |    |    |    |    | PCON |
|    |                     |    |    |    |    |    |    |    |      |
|    | Not bit addressable |    |    |    |    |    |    |    | DPH  |
| 83 | Not bit addressable |    |    |    |    |    |    |    | DPL  |
| 82 | Not bit addressable |    |    |    |    |    |    |    | SP   |
| 81 | 87                  | 86 | 85 | 84 | 83 | 82 | 81 | 80 | PO   |
| 80 |                     |    |    |    |    |    |    |    |      |

Bộ nhớ dữ liệu trên chip

### \* Các thanh ghi chức năng đặc biệt (SFR)

Không phải tất cả 128 địa chỉ từ 80H đến FFH đều được định nghĩa mà chỉ có 21 địa chỉ được định nghĩa.

Các thanh ghi chức năng đặc biệt bao gồm:

- + Tử trạng thái chương trình PSW: có địa chỉ là D0H

| Bit   | Kí hiệu | Địa chỉ | Mô tả                               |
|-------|---------|---------|-------------------------------------|
| PSW.7 | CY      | D7H     | Cờ Nhớ                              |
| PSW.6 | AC      | D6H     | Cờ nhớ phụ                          |
| PSW.5 | F0      | D5H     | Cờ 0                                |
| PSW.4 | RS1     | D4H     | Chọn dây thanh ghi 1                |
| PSW.3 | RS0     | D3H     | Chọn dây thanh ghi 0                |
|       |         |         | 00 = bank 0: địa chỉ từ 00F đến 07H |
|       |         |         | 01 = bank 1: địa chỉ từ 08F đến 0FH |
|       |         |         | 10 = bank 2: địa chỉ từ 00F đến 07H |
|       |         |         | 11 = bank 3: địa chỉ từ 00F đến 07H |
| PSW.2 | OV      | D2H     | Cờ tràn                             |
| PSW.1 | -       | D1H     | Dự trữ                              |
| PSW.0 | P       | D0H     | Cờ kiểm tra chẵn lẻ                 |

+ Thanh ghi B: Có địa chỉ F0H được dùng chung với thanh chứa A trong các phép toán nhân và chia.

+ Con trỏ Stack (SP) : là thanh ghi 8 bit ở địa chỉ 81H, nó chứa địa chỉ của dữ liệu hiện đang ở đỉnh của stack.

+ Con trỏ dữ liệu DPTR:

- Dùng để truy xuất bộ nhớ chương trình ngoài hoặc bộ nhớ dữ liệu ngoài.

- DPTR là thanh ghi 16 bit có địa chỉ 82H (byte thấp ) và 83H (byte cao).

Ví dụ:

MOV A, #55H ;Nạp hằng dữ liệu 55H và thanh chứa A

MOV DPTR, %1000 ;Nạp hằng địa chỉ 16 bit 1000H cho

; con trỏ DPTR

MOV @DPTR, A ; Chuyển dữ liệu từ A vào RAM ngoài

; tại địa chỉ DPTR trỏ tới.

+ Các thanh ghi port:

- Port 0 : địa chỉ 80H

- Port 1 : địa chỉ 90H
- Port 2 : địa chỉ A0H
- Port 3 : địa chỉ B0H

+ Các thanh ghi định thời:

8951 có 2 bộ định thời/đếm dùng để định khoảng thời gian hoặc đếm các sự kiện.

- Bộ định thời 0: địa chỉ 8AH (TL0 ) và 8CH (TH0)
- Bộ định thời 1: địa chỉ 8bH (TL1 ) và 8DH (TH1)

Hoạt động của bộ định thời được thiết lập bởi thanh ghi chế độ định thời TMOD ở địa chỉ 89H và thanh ghi điều khiển bộ định thời TCON ở địa chỉ 88H (chỉ có TCON được định địa chỉ từng bit)

+ Các thanh ghi của port nối tiếp: Chip 8951 có 1 port nối tiếp để truyền thông với các thiết bị như các thiết bị đầu cuối hoặc modem...

+ Các thanh ghi ngắt: có một cấu trúc ngắt với 2 mức ưu tiên và 5 nguyên nhân ngắt. Các ngắt bị vô hiệu hoá sau khi Reset hệ thống và được phép bằng cách vào thanh ghi IE ở địa chỉ A8H.

Mức ưu tiên ngắt được thiết lập bởi thanh ghi IP ở địa chỉ B8H.

+ Thanh ghi điều khiển nguồn: PCON có địa chỉ 87H.

### **c/ Tóm tắt tập lệnh**

Thông qua việc khảo sát các kiểu định địa chỉ và các ví dụ trên các tình huống lập trình điển hình để chúng ta tiếp cận tập lệnh của họ MCS-51.

✦ Các kiểu định địa chỉ.

Có 8 kiểu định địa chỉ :

+ Thanh ghi.:

VD: MOV PSW,#00011000B

+ Trực tiếp :

VD: MOV P1, A

+ Gián tiếp:

VD: MOV A,@R0

+ Tức thời:

VD: MOV A, #54

+ Tương đối:

VD: SJMP THREE :Nhảy đến nhãn THREE

+ Tuyệt đối:

VD: AJMP THREE

+ Dài:

+ Chỉ số.

VD: JMP @A+DPTR

♦ Các loại lệnh:

Có 5 nhóm lệnh:

+ Nhóm lệnh số học.

ADD A , nguồn: Cộng toán hạng nguồn vào A

ADD A, #data : Cộng dữ liệu data với A

ADDC A, nguồn: Cộng nguồn với A và cờ nhớ.

ADDC A, #data : Cộng dữ liệu data với A và cờ nhớ.

SUBB A, nguồn: Trừ A với nguồn

SUBB A, #data : Trừ A với data

INC A : Tăng nội dung thanh ghi A lên 1

DEC A : Giảm nội dung thanh ghi A lên 1

INC DPTR : Tăng DPTR

MUL AB : Nhân nội dung thanh ghi A và B

DIV AB : Chia A cho B

DA A : Hiệu chỉnh thập phân thanh ghi A

+ Nhóm lệnh Logic.

ANL A, nguồn AND

ANL A, #data

ANL direct, A



|                    |                      |
|--------------------|----------------------|
| ANL direct , #data |                      |
| ORL A, nguồn       | OR                   |
| ORL A, #data       |                      |
| ORL direct, A      |                      |
| ORL direct , #data |                      |
| XRL A, nguồn       | OR                   |
| XRL A, #data       |                      |
| XRL direct, A      |                      |
| XRL direct , #data |                      |
| CLR A              | Xoá A                |
| CPL A              | Lấy bù A             |
| RL A               | Quay trái A          |
| RLC A              | Kể cả cờ nhớ         |
| RR A               | Quay phải A          |
| RRC A              | Kể cả cờ nhớ         |
| SWAP A             | Hoán đổi 2 nửa 4 bit |

#### + Nhóm lệnh di chuyển dữ liệu

|                   |                                     |
|-------------------|-------------------------------------|
| MOV A, nguồn      | Di chuyển toán hạng nguông đến đích |
| MOV A, #data      |                                     |
| MOV dest , A      |                                     |
| MOV dest , source |                                     |
| MOV dest, #data   |                                     |
| MOV DPTR, #data16 |                                     |
| MOVC A, @A+DPTR   | Di chuyển từ bộ nhớ chương trình    |
| MOVC A, #A+PC     |                                     |
| MOVX A, @Ri       |                                     |
| MOVX A, @DPTR     |                                     |

MOVX @Ri, A

MOVX @DPTR, A

PUSH direct                      Cất vào Stack

POP direct                      Lấy ra từ Stack

XCH A, source                      Trao đổi các byte

XCHD A, @Ri                      Trao đổi các digit thấp

+ Nhóm lệnh xử lí bit.

CLR C    xoá bit

CLR bit

SETB C

SETB bit

CPL C

CPL bit

ANL C, bit      AND

ANL C, /bit    AND NOT bit với C

ORL C, bit

ORL C, /bit

MOV C, bit

MOV bit, C

JC rel    Nhảy đến Rel nếu C=1

JNC rel    Nhảy đến Rel nếu C=0

JB bit, rel    Nhảy nếu bit bằng 1

JNB bit, rel    Nhảy nếu bit =0

JBC bit, rel    Nhảy nếu bit =1 rồi xoá bit

+Nhóm lệnh rẽ nhánh.

ACALL addr11    Gọi chương trình con

LCALL addr16

|                       |                              |
|-----------------------|------------------------------|
| RET                   | Quay về từ chương trình con  |
| RETI                  | Quay về từ chương trình ngắt |
| AJMP addr11           | Nhảy                         |
| LJMP addr16           |                              |
| SJMP rel              |                              |
| JMP @ A+DPTR          |                              |
| JZ rel                | Nhảy nếu A=0                 |
| JNZ rel               | Nhảy nếu A $\neq$ 0          |
| CJNE A,direct, rel    | So sánh và nhảy              |
| CJNE #data, rel       |                              |
| CJNE Rn,#data, rel    |                              |
| CJNE @ Ri,# data, rel |                              |
| DJNZ Rn, rel          | Giảm và nhảy nếu khác 0      |
| DJNZ direct, rel      |                              |
| NOP                   | Không làm gì                 |

#### **d/ Hoạt động định thời**

Là một chuỗi các Flip-Flop nối tiếp nhau và nhận tín hiệu từ nguồn xung nhịp. Bộ định thời được lập trình sao cho sẽ tràn trong một khoảng thời gian nhất định và set cờ tràn của bộ định thời = 1.

Các bộ định thời sử dụng để:

- Định thời trong một khoảng thời gian.
- Đếm sự kiện.
- Tạo tốc độ baud cho các port nối tiếp.

Các bộ định thời được truy xuất bằng cách sử dụng 6 thanh ghi chức năng đặc biệt.

| SFR của bộ định thời | Mục đích | Địa chỉ | Địa chỉ của Bit |
|----------------------|----------|---------|-----------------|
|                      |          |         |                 |

|      |                              |     |       |
|------|------------------------------|-----|-------|
| TCON | Điều khiển                   | 88H | Có    |
| TMOD | Chọn chế độ                  | 89H | Không |
| TL0  | Byte thấp của bộ định thời 0 | 8AH | Không |
| TL1  | Byte thấp của bộ định thời 1 | 8BH | Không |
| TH0  | Byte cao của bộ định thời 0  | 8CH | Không |
| TH1  | Byte cao của bộ định thời 1  | 8DH | Không |

#### e/ Hoạt động ngắt

- Là sự xảy ra một sự kiện, một điều kiện làm cho chương trình hiện hành tạm dừng trong khi điều kiện được phục vụ bởi một chương trình khác. Ngắt đóng vai trò quan trọng trong việc thiết kế, thực hiện ứng dụng các bộ vi điều khiển.

- Tổ chức ngắt của 8951:

+ Có 5 nguyên nhân tạo ra ngắt: 2 ngắt do bên ngoài, 2 ngắt do bộ định thời, một ngắt do port nối tiếp.

+ Khi xảy ra 2 hay nhiều ngắt đồng thời hoặc một ngắt trong khi đang thực hiện một ngắt khác thì ta sẽ có 2 sơ đồ xử lý các ngắt: Sơ đồ chuỗi vòng và sơ đồ 2 mức ưu tiên.

♦ Các cách thức cho phép ngắt và không cho phép ngắt:

Một số nguyên nhân ngắt được cho phép và không được cho phép.

| Bit  | Kí hiệu | Địa chỉ bit | Mô tả<br>(0= Không cho phép; 1= Cho phép) |
|------|---------|-------------|---|
| IE.7 | EA      | AFH         | Cho phép/ không cho phép toàn cục         |
| IE.6 | -       | AEH         | Không sử dụng                             |
| IE.5 | ET2     | ADH         | Cho phép ngắt do bộ định thời 2           |
| IE.4 | ES      | ACH         | Cho phép ngắt do port nối tiếp            |
| IE.3 | ET1     | ABH         | Cho phép ngắt do bộ định thời 1           |
| IE.2 | EX1     | AAH         | Cho phép ngắt từ bên ngoài (Ngắt ngoài 1) |

|      |     |     |  |
|------|-----|-----|--|
| IE.1 | ET0 | A9H | Cho phép ngắt do bộ định thời 0            |
| IE.0 | EX0 | A8H | Cho phép ngắt từ bên ngoài (Ngắt ngoài 0 ) |

♦ Ưu tiên ngắt:

Các ưu tiên ngắt được lập trình riêng rẽ để có một trong 2 mức ưu tiên thông qua các thanh ghi chức năng đặc biệt được định địa chỉ bit.

| Bit  | Kí hiệu | Địa chỉ bit | Mô tả<br>(0= Không cho phép; 1= Cho phép) |
|------|---------|-------------|---|
| IP.7 | -       | -           | Không sử dụng                             |
| IP.6 | -       | -           | Không sử dụng                             |
| IP.5 | PT2     | BDH         | Ưu tiên ngắt do bộ định thời 2            |
| IP.4 | PS      | BCH         | Ưu tiên ngắt do port nối tiếp             |
| IP.3 | PT1     | BBH         | Ưu tiên ngắt do bộ định thời 1            |
| IP.2 | PX1     | BAH         | Ưu tiên ngắt từ bên ngoài (Ngắt ngoài 1)  |
| IP.1 | PT0     | B9H         | Ưu tiên ngắt do bộ định thời 0            |
| IP.0 | PX0     | B8H         | Ưu tiên ngắt từ bên ngoài (Ngắt ngoài 0 ) |

♦ Xử lý ngắt:

Các thao tác xảy ra sau khi ngắt:

- Hoàn tất việc thực thi lệnh hiện hành.
- Bộ đếm chương trình PC được hoàn tất.
- Trạng thái của ngắt hiện hành được lưu giữ lại.
- Các ngắt được chặn lại ở mức ngắt.
- Bộ đếm chương trình PC được nạp địa chỉ véctơ của trình phục vụ ngắt ISR

- ISR được thực thi.

Các cờ ngắt:

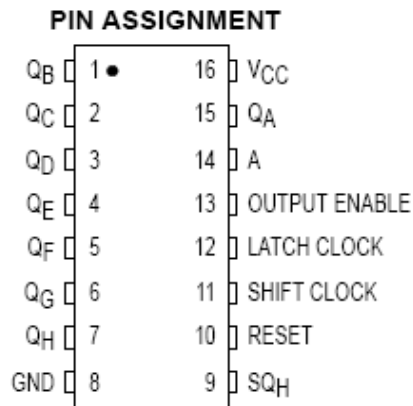
| Ngắt                        | Cờ  | Thanh ghi SFR và vị trí bit |
|-----------------------------|-----|-----------------------------|
| Do bên ngoài (ngắt ngoài 0) | IE0 | TCON.1                      |
| Do bên ngoài (ngắt ngoài 1) | IE1 | TCON.3                      |

|                   |     |        |
|-------------------|-----|--------|
| Do bộ định thời 1 | TF1 | TCON.7 |
| Do bộ định thời 0 | TF0 | TCON.5 |
| Do port nối tiếp  | TI  | SCON.1 |
| Do port nối tiếp  | RI  | SCON.0 |

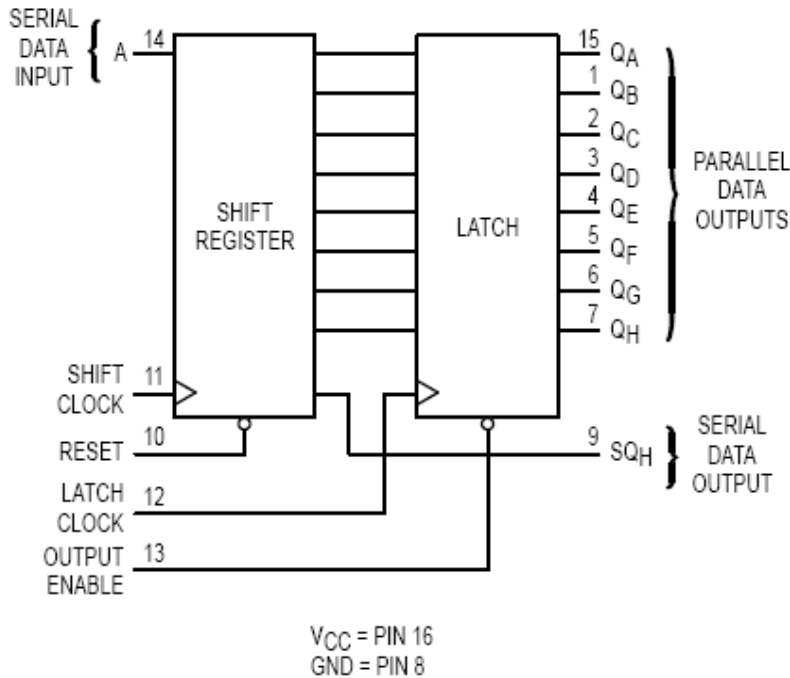
## 2. Thanh ghi dịch 74HC595

74HC595 là một thanh ghi dịch 8 bit đầu vào nối tiếp, có các đầu ra song song và nối tiếp, ngõ ra có bộ đệm 3 trạng thái.

Sơ đồ chân của 74HC595 như sau:



Sơ đồ chân 74HC595



Sơ đồ logic của 74HC595

+ QA, QB, QC, QD, QE, QF, QG, QH: là các ngõ ra song song của 74HC595

+ Chân 14 (A): đầu vào nối tiếp

+ Chân 9 (SQ<sub>H</sub>) : đầu ra nối tiếp

+ Chân 13 (Output Enable): Tích cực mức thấp, khi chân này ở mức thấp thì tín hiệu từ bộ chốt được đưa ra đầu ra. Khi nó ở mức cao thì các đầu ra song song ở trạng thái trở kháng cao. Đầu ra nối tiếp không bị ảnh hưởng bởi chân này.

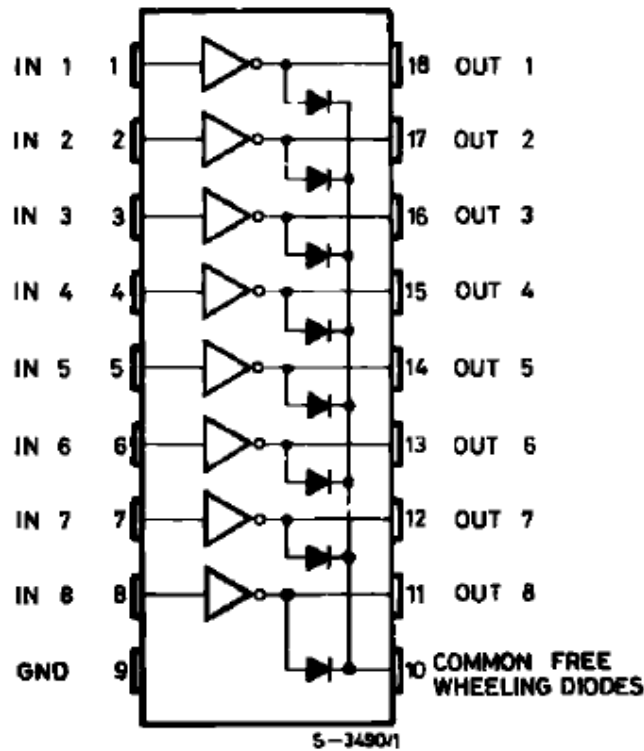
+ Chân 12 (Latch clock): Quá trình chuyển từ mức thấp sang mức cao ở Latch clock sẽ chốt dữ liệu được dịch trong thanh ghi dịch vào bộ chốt

+ Chân 11 (Shift Clock) : đầu vào xung nhịp, một quá trình chuyển từ mức thấp đến mức cao ở chân này sẽ dịch dữ liệu trong thanh ghi dịch một nhịp

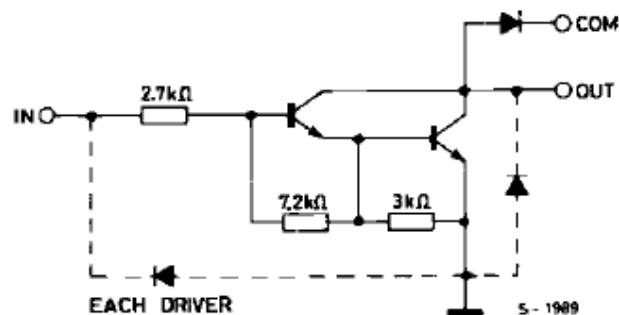
+ Chân 10 (Reset) : reset không đồng bộ, tích cực mức thấp. Mức thấp ở chân này sẽ reset thanh ghi dịch nhưng không reset bộ chốt lỗi ra.

### 3. ULN2803

+ Gồm 8 cặp transistor mắc kiểu darlington, có E chung



Sơ đồ chân của ULN2803



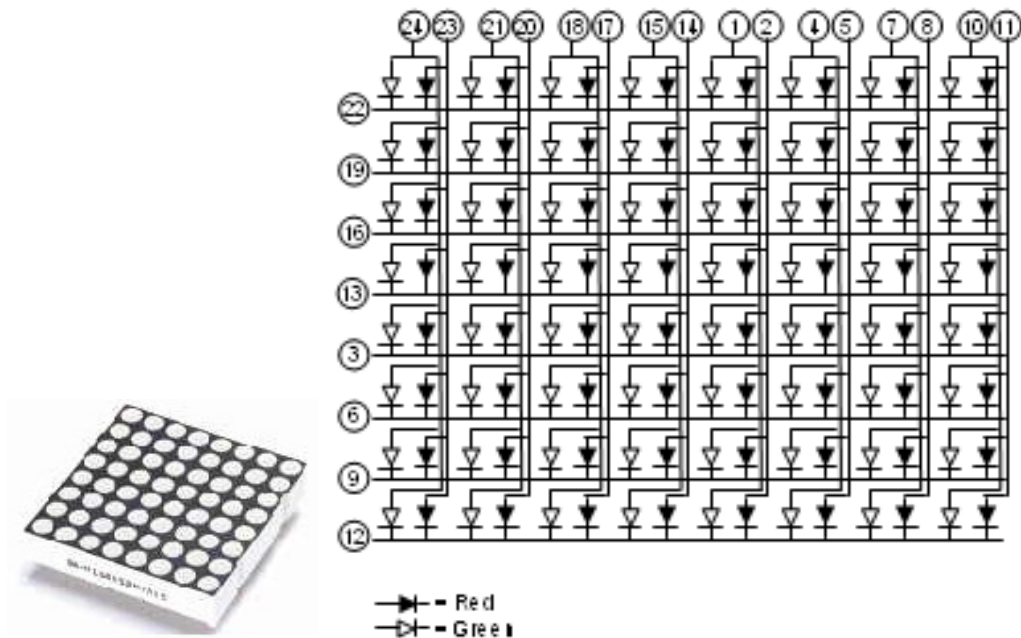
Sơ đồ mắc darlington vào ra của ULN2803



- + Chân 1 -> chân 8 (In 1 -> In 8): 8 đầu vào
- + Chân 11-> chân 16 : 8 đầu ra .
- + Chân 9 : đất chung của các cực E của Darlington
- + Chân 10 : Cực C chung

#### 4. Led ma trận 8x8

Loại led 2 màu, 24 chân, chung Anode theo hàng



## V. Nguyên lý và tác dụng linh kiện:

Đây là loại led Anode chung ở hàng nên để một led sáng thì dữ liệu ở hàng phải ở mức cao và xuất ra ở cột phải ở mức thấp. Vì điều khiển điều khiển thanh ghi dịch để lựa chọn cột sáng

Tại mỗi thời điểm chỉ có một cột được sáng, nhưng ta lợi dụng tính chất lưu ảnh trên võng mạc để có thể hiển thị được đồng thời cả 32 cột.

Tần số quét lúc này phải đảm bảo  $\geq 24$  hình/s

Để led sáng đẹp và ổn định thì dòng qua led khoảng  $I_{tb} = 10\text{mA}$

Một cột có 8 led, nghĩa là tối đa một thời điểm có 8 led sáng

Vậy dòng cần thiết cho một cột led là

$$\begin{aligned} I_{\text{cột}} &= I_{tb} \times 8 \times \text{Hệ số an toàn} \\ &= 10 \times 8 \times 2 \\ &= 160\text{mA} \end{aligned}$$

Ở cột ta dùng ULN2803 để hút dòng, mà khả năng hút dòng của ULN2803 lên tới 500mA nên hoàn toàn phù hợp trong trường hợp này

Về nguyên lý tại một thời điểm chỉ có một cột được phép sáng, nhưng do ta quét nhanh nên sẽ cảm thấy tất cả các cột đều sáng, thời gian sáng 1 cột chỉ là  $1/32$  chu kỳ quét. Để đảm bảo mắt người nhìn thấy các led sáng không bị nhấp nháy thì dòng cung cấp cho một led cần gấp 32 lần dòng trung bình để chia đều khoảng thời gian nó không được chiếu sáng

Dòng cần thiết cung cấp cho các hàng là:

$$\begin{aligned} I_{\text{hàng}} &= 10 \times 32 \times 2 \\ &= 640\text{mA} \end{aligned}$$

Với A1015 loại GR có hệ số khuếch đại là 200-400. Qua đo thực tế hệ số khuếch đại của các Transistor là khoảng 250.

Khi hoạt động để dòng cung cấp cho ma trận ổn định Tran hoạt động ở trạng thái bão hoà

$$I_{cbh} = I_{\text{hàng}} = 640\text{mA}$$

$$\beta * I_b \geq I_{cbh}$$

$$I_b \geq 640/250 = 2.56 \text{ mA}$$

$$U_b/R_b \geq 2.56$$

$$R_b \leq (5-0.7)/2.56 = 1.68 \text{ K}$$

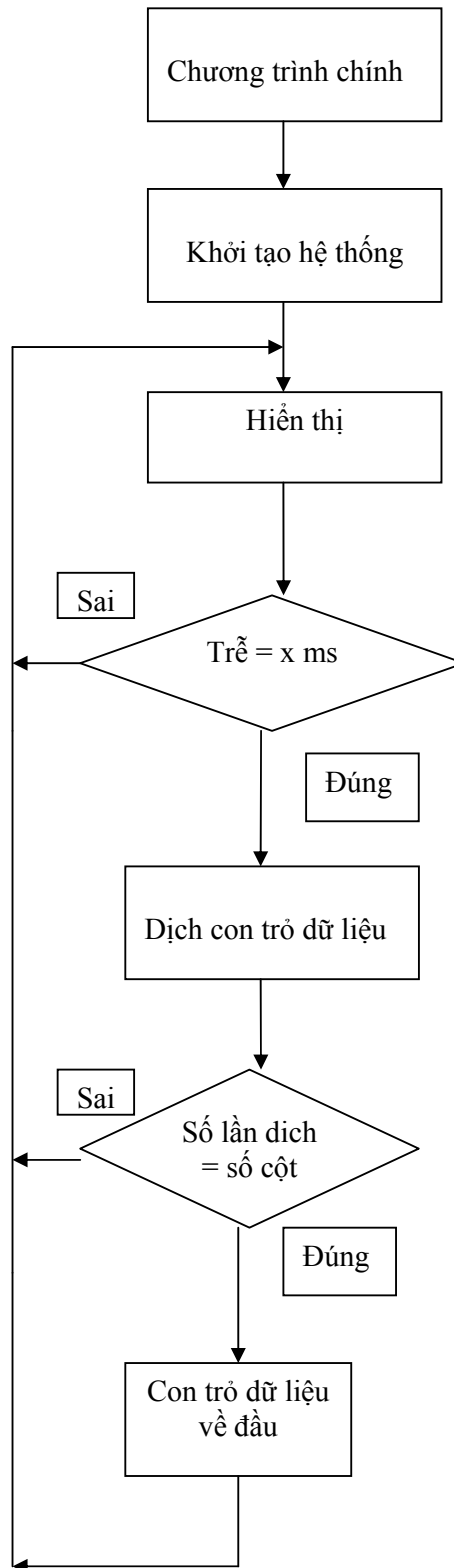
Chọn trở 1K

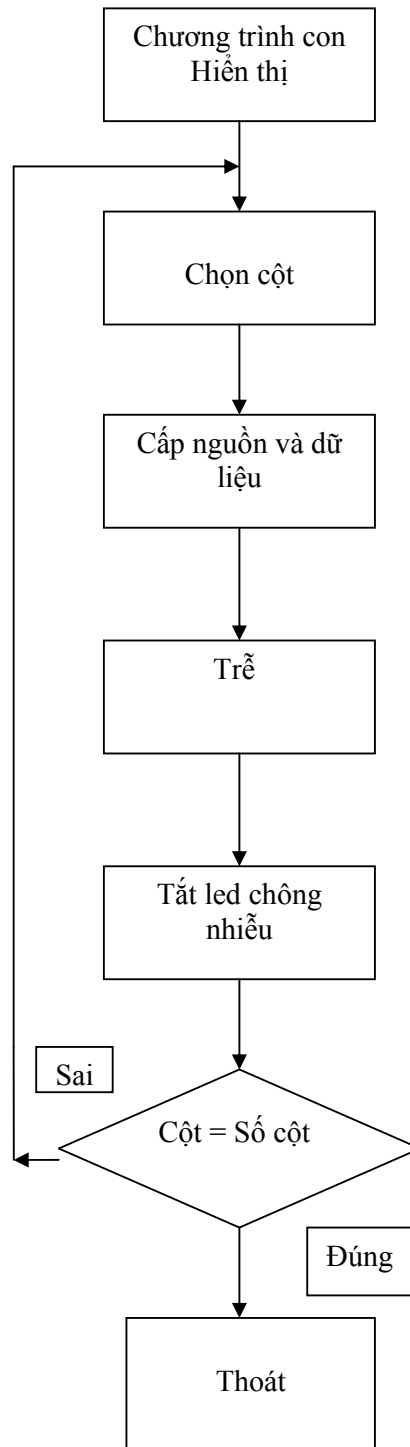
## **VI. Lưu đồ thuật toán và chương trình**

Sử dụng kiểu tra bảng dữ liệu, thực hiện quét theo cột

**Mã nguồn: file BYG.ASM kèm theo**

Lưu đồ thuật toán:





## **VII. Nhận xét, kết luận và hướng mở rộng đề tài**

Nhu cầu trang bị bảng điện tử ở cửa hàng, tòa nhà, sân bay, nhà ga, công ty chứng khoán hiện nay rất lớn. Đã có những doanh nghiệp chuyên sản xuất những sản phẩm như thế này như Phú Thành, tuy nhiên các sản phẩm đó mới chỉ được ứng dụng ở các cửa hàng, siêu thị với yêu cầu chất lượng chưa cao và ít tính năng. Từ thành công bước đầu trong dự án này, chúng em đã có điều kiện tìm hiểu về cách thức hoạt động của bảng điện tử, quan trọng hơn là biết ứng dụng kiến thức vi xử lý đã học vào trong thực tế. Đây là cơ sở quan trọng để nhóm có thể tiếp tục phát triển đề tài lên cao hơn.

Một số hạn chế của mạch:

- Phương pháp hiển thị trong mạch quang báo này là quét theo cột, do khi kích thước bảng tăng lên về số cột thời gian quét để hiển thị đầy đủ một khung hình sẽ tăng, đến một giá trị nào đó sẽ làm mất hiệu ứng lưu ảnh trên võng mạc và ta sẽ nhận thấy sự gián đoạn trong quá trình hiển thị. Ngoài ra, khi số hàng tăng lên thì cần thêm từng ấy Transistor để điều khiển hàng làm tăng kích thước của mạch. Nếu sử dụng phương pháp quét theo hàng thì mỗi khi quét một hàng ta sẽ xuất dữ liệu điều khiển các cột và lựa chọn hàng bằng cách sử dụng một IC giải mã. Trong bài này có thể sử dụng 74LS138. Tuy nhiên khi sử dụng quét hàng thì tại một thời điểm số LED cần hiển thị tối đa bằng số cột của bảng hiển thị, do đó yêu cầu cường độ dòng điện đủ lớn để cấp cho tất cả các LED sáng bình thường. Rõ ràng với mỗi phương pháp thì đều có ưu và nhược điểm riêng.

- Nội dung hiển thị được nạp vào trong ROM nên nếu nội dung hiển thị mà lớn thì đòi hỏi tăng kích thước bộ nhớ ROM bằng cách ghép nối thêm ROM. Kể cả như vậy thì bộ nhớ ghép nối chỉ được phép tối đa 64Kbyte (với IC AT89C51). Do vậy đây vẫn là một phương án chưa “tối ưu”. Để khắc phục điều này, các ký hiệu hiển thị cần được mã hóa và lưu

trong bộ nhớ. Mỗi khi một ký hiệu nào được gọi tới hiển thị thì chương trình sẽ tự động truy cập tới ô nhớ chứa mã điều khiển các LED tương ứng với ký hiệu đó. Phương án này đòi hỏi lập trình phức tạp hơn và có ghép nối với máy tính trong các ứng dụng cao cấp. Tuy nhiên với mục đích tìm hiểu ứng dụng của kỹ thuật vi xử lý và yêu cầu bài toán đơn giản cũng như thời gian giới hạn, nhóm chúng em mới hoàn thành sản phẩm ở mức độ hiện nay.

Trong quá trình thực hiện dự án này, chúng em đã học tập được thêm nhiều kiến thức thực tế, trao đổi thêm giữa các thành viên, làm quen với tác phong làm việc theo nhóm và cách thức xử lý các khó khăn khi gặp phải. Kỹ thuật vi xử lý là một môn học hay nhưng khó, đây cũng là lần đầu tiên nhóm thực hiện một dự án như thế này, do đó bên cạnh việc tự học thì sự hướng dẫn và các kinh nghiệm của thầy đã giúp đỡ chúng em rất nhiều. Chúng em cảm ơn thầy Phạm Ngọc Nam rất nhiều về những chỉ bảo tận tình trong thời gian qua và mong sẽ tiếp tục được thầy giúp đỡ trong quá trình ôn thi cho kỳ thi sắp tới!

## TÀI LIỆU THAM KHẢO

1. <http://home.wanadoo.nl/electro1/avr/dotmatrix.htm>
2. Tổng Văn On, Hoàng Đức Hải – Họ Vi điều khiển 8051, NXB Lao Động – Xã hội
3. Barry B. Brey - The Intel Microprocessors 8086/8088, 80186/80188. 80286, 80386, 80486, Pentium, and Pentium Pro Processor Architecture, Programming, and Interfacing, Fourth Edition, Prentice – Hall International, inc
4. Nguyễn Tăng Cường – Lập trình cho họ vi điều khiển 8051, NXB Khoa học Kỹ thuật
5. Ngô Diên Tập - Lập trình ghép nối máy tính trong Windows, NXB Khoa học Kỹ thuật
6. Huỳnh Đắc Thắng - Cẩm nang thực hành vi mạch tuyến tính, TTL/LS, CMOS, NXB Khoa học và Kỹ thuật, Hà Nội, 1994
7. Phạm Minh Hà, Kỹ thuật mạch điện tử, NXB KHKT, 1998
8. Robert Boylestad, Louis Nashelsky - Electronic Device and Circuit Theory Sixth Edition, Prentice - Hall International, Inc
9. Victor P. Nelson, H. Troy Nagle, Bill D. Carroll, J. David Irwin - Digital Logic Circuit Analysis and Design, Prentice - Hall International, Inc