

Lời mở đầu:

Ngày nay khi nhu cầu về thông tin quảng cáo rất lớn , việc áp dụng các phương tiện kĩ thuật mới vào các lĩnh vực trên là rất cần thiết .

Khi bạn đến các nơi công cộng, bạn dễ dàng bắt gặp những áp phích quảng cáo điện tử chạy theo các hướng khác nhau với nhiều hình ảnh và màu sắc rất ấn tượng.

Từ yêu cầu của môn học kĩ thuật vi xử lý và thực tiễn như trên, chúng em quyết định chọn đề tài cho bài tập lớn môn học là:

Thiết kế mạch hiển thị dùng ma trận LED.

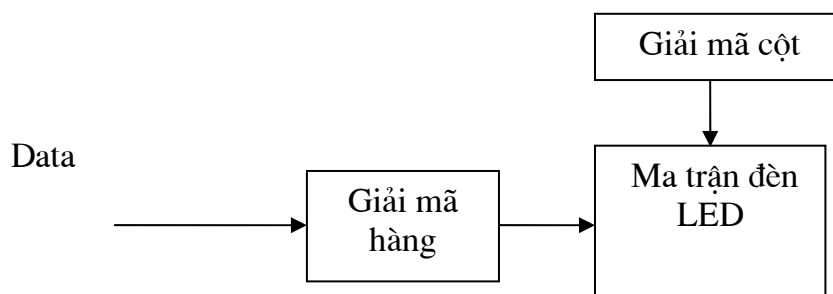
Khi đề tài được mở rộng thì sẽ có khả năng ứng dụng thực tiễn rất lớn .Nói tóm lại, trong thời đại bùng nổ thông tin hiện nay, khả năng ứng dụng và tiềm lực phát triển của hệ thống này là rất lớn, đặc biệt ở Việt Nam, các hệ thống như vậy còn rất ít, hầu hết đều được nhập từ nước ngoài với giá thành khá cao.

I) Cơ sở lý thuyết:

Dựa trên nguyên tắc như quét màn hình, ta có thể thực hiện việc hiển thị ma trận đèn bằng cách quét theo hàng và quét theo cột. Mỗi Led trên ma trận LED có thể coi như một điểm ảnh. Địa chỉ của mỗi điểm ảnh này được xác định đồng thời bởi mạch giải mã hàng và giải mã cột, điểm ảnh này sẽ được xác định trạng thái nhờ dữ liệu đưa ra từ bộ vi điều khiển 8951.

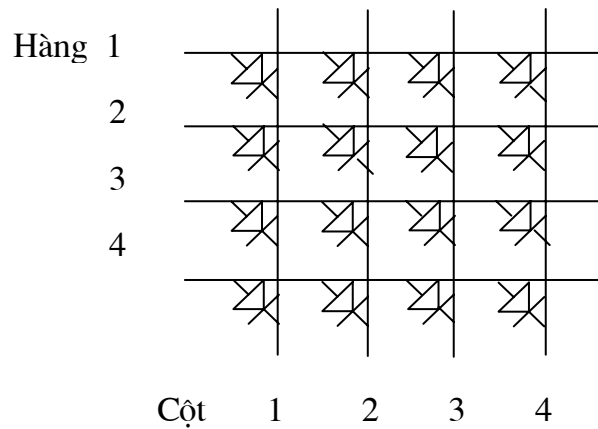
Như vậy tại mỗi thời điểm chỉ có trạng thái của một điểm ảnh được xác định. Tuy nhiên khi xác định địa chỉ và trạng thái của điểm ảnh tiếp theo thì các điểm ảnh còn lại sẽ chuyển về trạng thái tắt (nếu LED đang sáng thì sẽ tắt dần). Vì thế để hiển thị được toàn bộ hình ảnh của ma trận đèn, ta có thể quét ma trận nhiều lần với tốc độ quét rất lớn, lớn hơn nhiều lần thời gian kịp tắt của đèn. Mắt người chỉ nhận biết được tối đa 24 hình/s do đó nếu tốc độ quét rất lớn thì sẽ không nhận ra được sự thay đổi nhỏ của đèn mà sẽ thấy được toàn bộ hình ảnh cần hiển thị.

Sơ đồ khối:



Để thực hiện được quét hàng và quét cột thì ma trận LED được thiết kế như sau:

- ✓ Các LED trên cùng một hàng sẽ được nối các chân dương với nhau.
 - ✓ Các LED trên cùng một cột sẽ được nối các chân âm với nhau như hình vẽ
- Ta có thể mô phỏng một ma trận Led đơn giản 4x4 như sau:

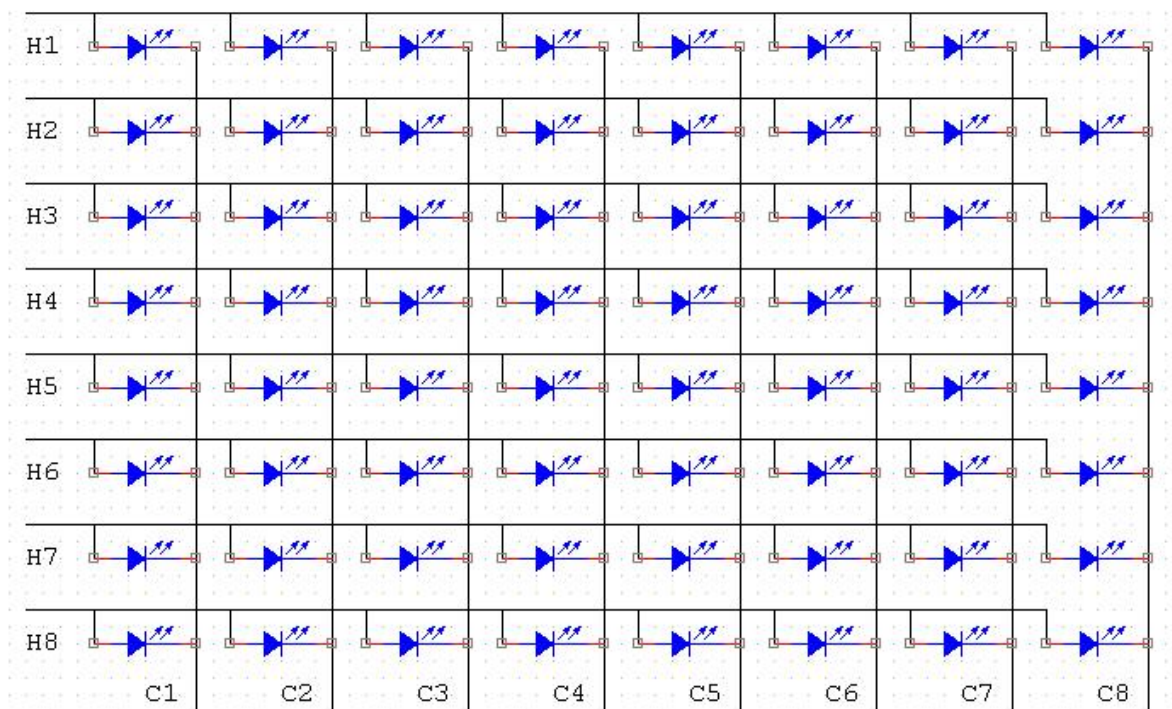


Sơ đồ thiết kế ma trận LED

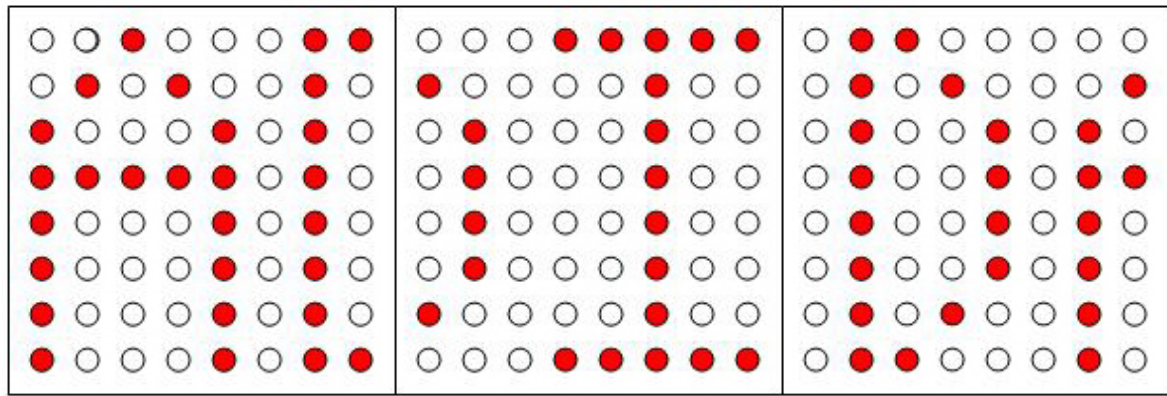
Trạng thái của một LED sẽ được quyết định bởi tín hiệu điện áp đi vào đồng thời cả 2 chân. Ví dụ để LED sáng thì điện áp 5V phải đưa vào chân dương và chân âm phải được nối đất, LED sẽ tắt khi không có điện áp đưa vào chân dương.

Với đề tài này, chúng em chọn loại ma trận LED 8x8 để hiển thị.

Ta có sơ đồ nguyên lý của Ma trận LED 8x8:



Để ma trận có thể sáng như hình vẽ (hiển thị một phần của chữ **ADIDAS**):



Đèn LED thứ nhất

Đèn LED thứ hai

Đèn Led thứ ba

Thực hiện quét dòng và cột:

- Chọn cột 1, đưa điện áp cột 1 về 0.
- Sau đó chọn và quét lần lượt các hàng 1,2,3,4,5,6,7,8 như sau:

- + Đèn 1 tắt → Điện áp đưa vào hàng 1 là 0V.
- + Đèn 2 tắt → Điện áp đưa vào hàng 2 là 0V.
- + Đèn 3 sáng → Điện áp đưa vào hàng 3 là 5V.
- + Đèn 4 sáng → Điện áp đưa vào hàng 4 là 5V.
- + Đèn 5 sáng → Điện áp đưa vào hàng 5 là 5V.
- + Đèn 6 sáng → Điện áp đưa vào hàng 6 là 5V.
- + Đèn 7 sáng → Điện áp đưa vào hàng 7 là 5V.
- + Đèn 8 sáng → Điện áp đưa vào hàng 8 là 5V.

Chọn cột 2, nối đất. Sau đó quét lần lượt các hàng 1,2,3,4,5,6,7,8.

- + Đèn 1 tắt → Điện áp đưa vào hàng 1 là 0V.
- + Đèn 2 sáng → Điện áp đưa vào hàng 2 là 5V
- + Đèn 3 tắt → Điện áp đưa vào hàng 3 là 0V
- + Đèn 4 sáng → Điện áp đưa vào hàng 4 là 5V
- + Đèn 5 tắt → Điện áp đưa vào hàng 5 là 0V
- + Đèn 6 tắt → Điện áp đưa vào hàng 6 là 0V
- + Đèn 7 tắt → Điện áp đưa vào hàng 7 là 0V
- + Đèn 8 tắt → Điện áp đưa vào hàng 8 là 0V

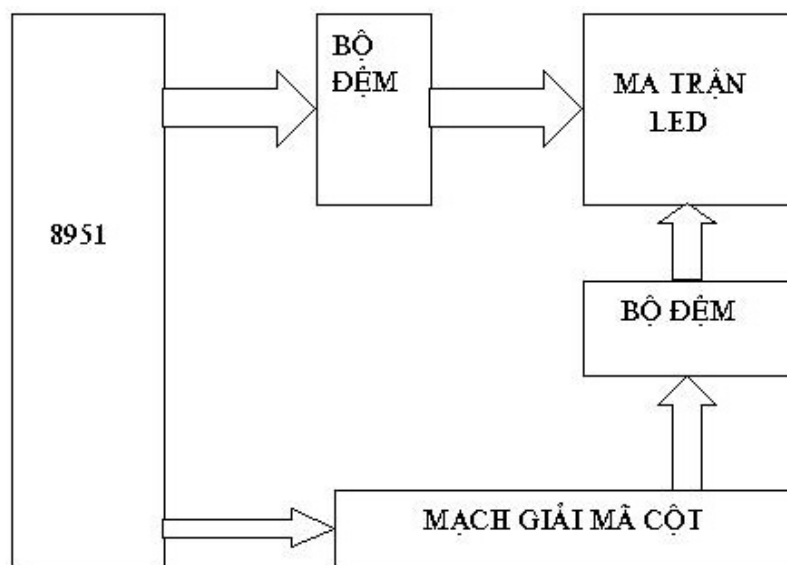
- Tiếp tục quét với các cột từ 3 đến 8 bằng cách như trên, sau đó chuyển sang quét đèn LED thứ hai và thứ ba một cách tương tự.
- Để mắt người nhận biết được toàn bộ hình ảnh của ma trận ta phải tiến hành quét nhiều lần. Do mắt người không phân biệt được sự thay đổi ảnh nếu ảnh đó được quét với tốc độ 24 hình/s nên nếu ta quét ảnh với tốc độ lớn hơn hoặc bằng 24 hình/s thì ảnh sẽ chạy liên tục và không bị giật

II) Thiết kế:

Để thực hiện được việc quét theo hàng và quét theo cột ta cần phải làm những công việc sau:

- Thiết kế ma trận LED theo mô hình như đã nói ở trên.
- Thiết kế bộ phân kênh đưa vào bộ giải mã địa chỉ cột cho tất cả các điểm ảnh.
- Thiết kế bộ đếm để ổn định dữ liệu (gồm bộ đếm hàng và bộ đếm cột).

1) Sơ đồ khối của mạch hiển thị dùng ma trận LED :



Sơ đồ khối cho mạch điều khiển ma trận LED 8 x 8.

2) Nhiệm vụ của các khối:

i.

Bộ vi điều khiển 8951:

Đây là nơi lưu giữ chương trình điều khiển chính và dữ liệu cho các mạch giải mã hàng và cột.

AT89C51 là một hệ vi tính 8 bit đơn chip CMOS có hiệu suất cao, công suất nguồn tiêu thụ thấp và có 4Kbyte bộ nhớ ROM Flash xóa được/lập trình được. Chip này được sản xuất dựa vào công nghệ bộ nhớ không mất nội dung có độ tích hợp cao.

AT89C51 có các đặc trưng chuẩn sau: 4Kbyte Flash, 128 byte RAM, 32 đường xuất nhập, hai bộ định thời/đếm 16 bit, một cấu trúc ngắt hai mức ưu tiên và 5 nguyên nhân ngắt, một port nối tiếp song công, mạch tạo dao động và tạo xung Clock trên Chip.

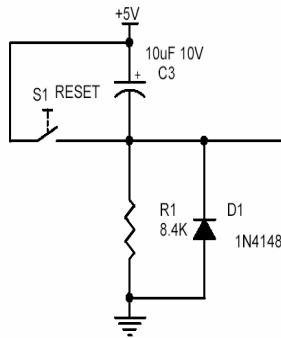
Chế độ nghỉ dừng CPU trong khi vẫn cho phép RAM, các bộ định thời/đếm, Port nối tiếp và hệ thống ngắt hoạt động.

Mô tả các chân của **IC 8951**:

- ✓ **Vcc**: Chân cung cấp điện.
- ✓ **GND**: Chân nối đất.
- ✓ **Port 0**: Port 0 là port xuất nhập 8 bit 2 chiều cực D hở. Port 0 còn được cấu hình làm bus địa chỉ (byte thấp) và làm bus dữ liệu đa hợp trong khi truy xuất bộ nhớ dữ liệu ngoài và bộ nhớ chương trình ngoài. Port 0 cũng nhận các byte mã trong khi lập trình cho Flash và xuất các byte mã trong khi kiểm tra chương trình.
- ✓ **Port 1**: Port 1 là port xuất nhập 8 bit hai chiều có các điện trở kéo lên bên trong. Khi các logic 1 được ghi lên các chân của port 1, các chân này được kéo lên mức cao bởi các điện trở kéo lên bên trong và có thể được sử dụng như là các ngõ vào. Khi làm nhiệm vụ là các port nhập, các chân của port 1 đang được kéo xuống mức thấp do tác động bên ngoài sẽ cấp dòng do có các điện trở kéo lên bên trong.
- ✓ **Port 2**: Port 2 là port xuất nhập 8 bit hai chiều có các điện trở kéo lên bên trong. Khi các logic 1 được ghi lên các chân của port 2, các chân này được sử dụng như là các ngõ vào. Khi làm nhiệm vụ port nhập, các chân của port 2 đang được kéo xuống mức thấp do tác động bên ngoài sẽ cấp dòng do có các điện trở kéo lên bên trong. Port 2 tạo ra byte cao của bus địa chỉ trong thời gian tìm nạp lệnh từ bộ nhớ chương trình ngoài, và trong thời gian truy xuất bộ nhớ dữ liệu ngoài sử dụng các địa chỉ 16 bit.
- ✓ **Port 3**: Là port xuất nhập 8 bit hai chiều có các điện trở kéo lên bên trong. Khi các logic 1 được ghi lên các chân của port 3, các chân này được kéo lên mức cao bởi các điện trở kéo lên bên trong và có thể được sử dụng như là các ngõ vào. Khi làm nhiệm vụ port nhập, các chân của port 3 đang được kéo xuống mức thấp do tác động bên ngoài sẽ cấp dòng do có các điện trở kéo lên bên trong. Port 3 còn được sử dụng làm các chức năng khác của **AT89C51**:

Chân	Chức năng
P3.0	ngõ vào Port nối tiếp
P3.1	ngõ ra Port nối tiếp
P3.2	ngõ vào ngắt ngoài 0
P3.3	ngõ vào ngắt ngoài 1
P3.4	ngõ vào bên ngoài của bộ định thời 1
P3.5	ngõ vào bên ngoài của bộ định thời 0
P3.6	điều khiển ghi bộ nhớ dữ liệu ngoài
P3.7	điều khiển đọc bộ nhớ dữ liệu ngoài

- ✓ **RST**: Ngõ vào Reset. Mức cao trên chân này trong 2 chu kỳ máy trong khi bộ dao động đang hoạt động sẽ Reset **AT89C51**.



Mạch Reset tự động khi khởi động máy

“Với bài này chúng em thực hiện Reset bằng cách nối chân 9 của 8951 với nguồn 5V”.

- ✓ **ALE:** ALE là một xung ngõ ra cho phép chốt địa chỉ ALE (Address Latch Enable) cho phép chốt byte thấp của địa chỉ trong thời gian truy xuất bộ nhớ ngoài. Chân này cũng được dùng làm ngõ vào xung lập trình (**PROG**) trong thời gian lập trình cho Flash.

Khi hoạt động bình thường, xung của ngõ ra ALE luôn luôn có tần số bằng 1/6 tần số của mạch dao động trên *chip*, có thể được sử dụng cho các mục đích định thời từ bên ngoài và tạo xung *Clock*. Tuy nhiên cần lưu ý là một xung ALE sẽ bị bỏ qua trong mỗi chu kỳ truy xuất của bộ nhớ dữ liệu ngoài.

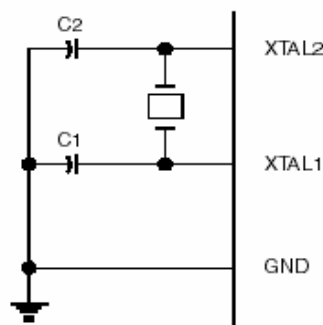
Khi cần, hoạt động cho phép chốt byte thấp của địa chỉ sẽ được vô hiệu hoá bằng cách *set* bit 0 của thanh ghi chức năng đặc biệt có địa chỉ byte là 8E(h). Khi bit này được *set*, **ALE** chỉ tích cực trong thời gian thực thi lệnh MOVX hoặc MOVC. Ngược lại chân này sẽ được kéo lên mức cao. Việc *set* bit không cho phép hoạt động chốt byte thấp của địa chỉ sẽ không có tác dụng nếu bộ vi điều khiển đang ở chế độ thực thi chương trình ngoài.

- ✓ **XTAL1:** Ngõ vào đến mạch khuếch đại đảo của mạch dao động và ngõ vào đến mạch tạo xung Clock bên trong chip.

- ✓ **XTAL2:** Ngõ ra từ mạch khuếch đại đảo của mạch dao động.

Để tạo mạch dao động cho vi điều khiển 8951 hoạt động, chúng em chọn mạch tạo dao động như hình vẽ dưới đây, với các giá trị của linh kiện là: $C1 = C2 = 30\text{pF}$;

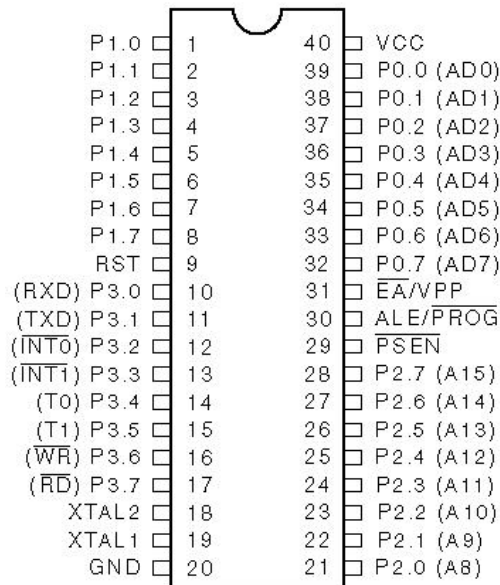
Thạch anh dao động có tần số 12MHz.



Note: $C1, C2 = 30\text{ pF} \pm 10\text{ pF}$ for Crystals
 $= 40\text{ pF} \pm 10\text{ pF}$ for Ceramic Resonators

Kết nối của mạch dao động

Sơ đồ chân của IC **AT89C51**:

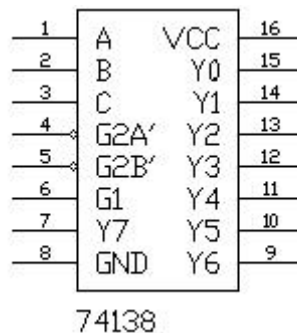


ii.

Mạch giải mã cột:

Dùng 3 IC **74LS138** (3 đầu vào, 8 đầu ra) để giải mã cho các cột của các ma trận LED. Do mỗi thời điểm chỉ có một IC giải mã được làm việc nên ta phải thiết kế mạch phân kênh để chọn IC làm việc. Dùng chân **P2.3 & P2.4** của vi điều khiển **8951** để làm đầu vào cho mạch phân kênh.

Sơ đồ chân IC **74LS138**:



Ta có bảng trạng thái cho mạch phân kênh như sau :

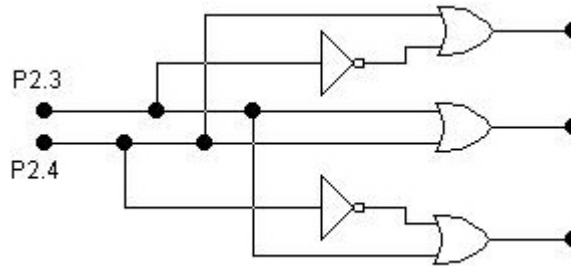
Từ đó ta có các hàm logic cho mạch phân kênh như sau :

$$\overline{Y1} = P2.3 + P2.4$$

$$\overline{Y2} = P2.3 + \overline{P2.4}$$

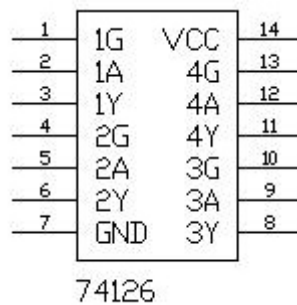
$$\overline{Y3} = \overline{P2.3} + P2.4$$

Ta có sơ đồ mạch phân kênh:



iii. Đối với hàng của ma trận LED :

Ta dùng 8 chân của **Port 1** nối trực tiếp với 8 hàng của ma trận LED thông qua bộ đệm gồm 2 IC **74LS126**.



iv. Bộ đệm :

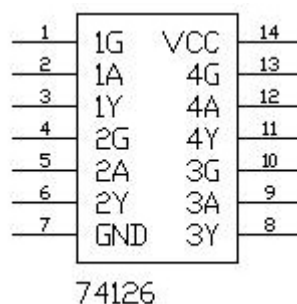
Điện áp có thể đưa trực tiếp từ bộ giải mã hàng qua các cổng NOT vào các hàng của ma trận đèn. Tuy nhiên điện áp này có thể không ổn định hoặc có thể xảy ra hiện tượng ảnh hưởng lẫn nhau giữa các chân tín hiệu đưa vào các hàng, dẫn đến độ sáng hoặc tắt của đèn không chính xác. Để tránh hiện tượng này đồng thời tránh phải dùng các cổng NOT, tại mỗi chân ra của bộ giải mã ta có thể dùng bộ đệm.

Bộ đệm thực chất là các mạch 3 trạng thái (tri-state logic), gồm có 1 đầu vào A, 1 đầu ra Y và 1 đầu điều khiển G. Trạng thái của đầu ra Y do đầu điều khiển G và đầu vào dữ liệu A quyết định. Khi đầu điều khiển chọn thì tín hiệu tại đầu ra sẽ thay đổi theo tín hiệu đầu vào; khi không được chọn thì ở đầu ra sẽ là trạng thái trở kháng cao, tín hiệu tại đầu vào không được đưa đến đầu ra.

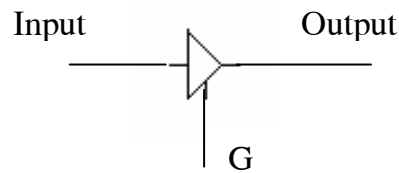
Trong bài này ta dùng 2 loại bộ đệm.

➤ Với các dữ liệu đưa đến hàng của ma trận LED ta dùng bộ đệm sử dụng IC **74LS126**.

Ta có sơ đồ IC **74LS126**:



- Sơ đồ mạch 3 trạng thái :



- Bảng trạng thái của IC **74LS126** (tích cực mức “1”):

Input	G	Output
0	0	high Z
1	0	high Z
0	1	0
1	1	1
0	1	high Z
1	1	high Z

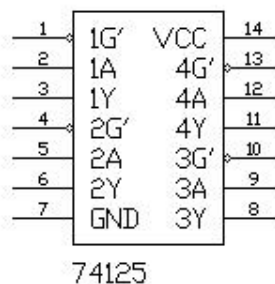
Khi đầu vào điều khiển $G = 0$ thì đầu ra ở trạng thái trở kháng cao không cho tín hiệu đi qua.

Khi đầu vào điều khiển $G = 1$ thì tín hiệu điện áp tại đầu ra sẽ giữ nguyên so với tín hiệu điện áp tại đầu vào (do đó không phải dùng các cổng NOT tại các đầu ra của bộ giải mã).

➤ Với các dữ liệu đưa đến cột của ma trận LED ta dùng bộ đệm sử dụng IC

74LS125

Ta có sơ đồ IC **74LS125**:



- Bảng trạng thái của IC **74LS125** (tích cực mức “0”):

Khi đầu vào điều khiển $G = 1$ thì đầu ra ở trạng thái trở kháng cao không cho tín hiệu đi qua.

Khi đầu vào điều khiển $G = 0$ thì tín hiệu điện áp tại đầu ra sẽ giữ nguyên so với tín hiệu điện áp tại đầu vào (do đó không phải dùng các cổng NOT tại các đầu ra của bộ giải mã).

Ma trận LED: Dùng 3 ma trận LED 8x8 để hiển thị nội dung .

3) Thiết kế:

- **Đối với hàng của bảng hiển thị:** dùng các chân của **Port 1** (từ chân **P1.0** đến chân **P1.7**) của vi điều khiển **8951** làm đầu vào điều khiển. Trước khi đưa vào các chân điều khiển hàng của ma trận, các chân của **Port 1** được đưa vào các chân điều khiển G của bộ đệm gồm 2 IC **74LS125**(có tất cả là 8 đầu vào điều khiển). Đầu vào của các IC này luôn được nối với nguồn 5V, 8 đầu ra của 2 IC này được đưa đến 8 hàng của ma trận LED.
- **Đối với giải mã cột của bảng hiển thị:** Dùng các chân **Port 2** của vi điều khiển **8951** làm đầu vào cho bộ giải mã cột.

Cu thể:

Các chân **P2.0, P2.1, P2.2** làm ba đầu vào cho từng IC giải mã **74LS138** (các chân A1, A2, A3 của **74LS138**). Các chân G2A, G2B của IC **74LS138** được nối với nhau và được nối với 3 đầu ra của mạch phân kênh.

Chân **P2.3, P2.4** làm đầu vào cho mạch phân kênh chọn IC giải mã làm việc tại từng thời điểm làm việc.

24 đầu ra của bộ giải mã (gồm 3 IC **74LS138**) được đưa đến các đầu vào điều khiển G của bộ đệm (gồm 6 IC **74LS125**). Đầu vào A của 6 IC này luôn được nối với đất. Đầu ra Y của 6 IC này (24 đầu ra) được nối với 24 cột của bảng hiển thị gồm 3 ma trận LED 8x8.

4) Nguyên lý hoạt động:

- Trong khi bộ giải mã cột chọn cột thứ nhất, bộ giải mã hàng sẽ quét lần lượt hết 8 hàng, dữ liệu được đưa vào sẽ xác định trạng thái của tất cả các đèn tại cột 1.
- Tiếp tục quét cột thứ hai, bộ giải mã hàng lại đưa dữ liệu vào lần lượt hết 8 hàng và quyết định trạng thái của tất cả các đèn ở cột 2.

Quá trình cứ thế tiếp tục cho đến khi quét hết 24 cột (do ta dùng 3 ma trận LED 8x8). Với đề tài này, chúng em chọn tốc độ quét là 60 lần và quét nhiều lần với thời gian trễ của đèn là 255 us, ta sẽ nhận biết được hình ảnh trên ma trận là dòng chữ : '**ADIDAS** ' chạy từ phải qua trái.

III) Các linh kiện dùng trong mạch:

- 1 IC **AT89C51**.
- 3 ma trận LED loại 8x8.
- 3 IC giải mã **74LS138**.
- 2 IC **74LS126**.
- 6 IC **74LS125**.
- 1 IC **7432** (cổng OR).
- 1 IC **7404** (cổng NOT).
- 2 tụ điện giá trị 33 pF.
- 1 bộ dao động thạch anh tần số 12 MHz.

IV) Tài liệu tham khảo:

- Kỹ thuật Vi xử lý -- Tác giả Văn Thế Minh--.
- Cấu trúc và lập trình họ vi điều khiển 8051. -- Tác giả Nguyễn Tăng Cường, Phan Quốc Thắng --

Mã nguồn của chương trình điều khiển mạch hiển thị dùng ma trận LED

(Sử dụng ngôn ngữ lập trình Assembly):

```
INCLUDE 89C51.MC
```

ORG 0H

JMP MAIN

DB

[illegible]

ORG 100H

```
BEGIN: MOV DPTR,#26
```

CLR A

CLR P3.1

MOV R7,#77

LOOP: CJNE R7,#0,QANH

ACALL DELAY

ACALL BEGIN

QANH: MOV R6,#60

MOV A,DPL

ADD A,#1

MOV DPL,A

```
CLR A
DEC R7
LOOP1: CJNE R6,#0,IMAGE
LJMP LOOP
IMAGE: MOV R5,#24
MOV R0,#0
MOV A,DPL
SUBB A,#24
MOV DPL,A
CLR A
DEC R6
LOOP2: CJNE R5,#0,COLUMN
LJMP LOOP1
COLUMN: MOVC A,@A+DPTR
MOV P1,#0
MOV P2,R0
MOV P1,A
ACALL DELAY
INC R0
DEC R5
CLR A
INC DPTR
LJMP LOOP2
DELAY:MOV TMOD,#02H
```

```
MOV TH0,#0H  
SETB TR0  
LOOP3:JNB TF0,LOOP3  
CLR TF0  
RET  
MAIN: ACALL BEGIN  
END
```

Hướng phát triển của chương trình

- Thực tế hình ảnh hiển thị lớn hoặc cần có độ phân giải tốt đòi hỏi số lượng đèn trong ma trận lớn hơn nhiều lần. Từ đó đặt ra một yêu cầu là mở rộng ma trận đèn.
- Dựa trên cơ sở là ma trận LED 8x8 như đã nói ở trên, có thể thiết kế mở rộng (bằng cách tăng thêm số hàng và số cột) theo nguyên lý tương tự (quét hàng và quét cột) tuy nhiên phải thêm các bộ đếm, các bộ giải mã địa chỉ và bộ đệm.
- Do các bộ đệm được đặt ở đầu ra của bộ giải mã cột nên để giảm số lượng bộ đệm phải sử dụng có thể mở rộng ma trận LED bằng cách tăng thêm số cột.