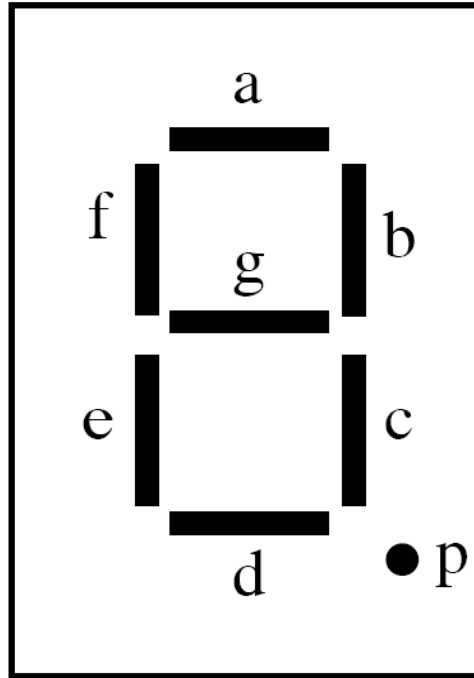


Chương 4: Giao tiếp hiển thị

4.1 LED 7 đoạn

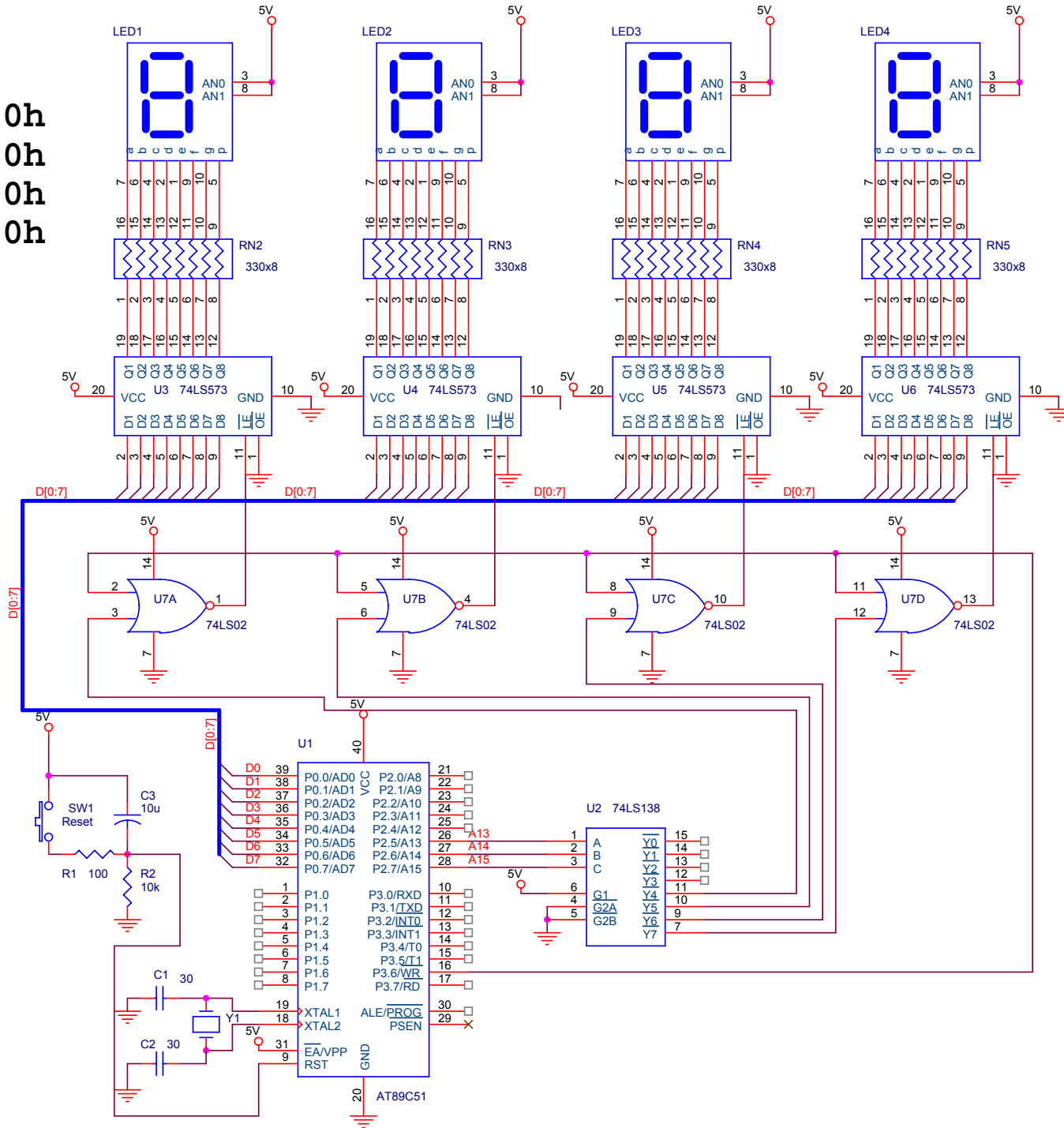


Đoạn:	p	g	f	e	d	c	b	a
	↑	↑	↑	↑	↑	↑	↑	↑
Bit:	D7	D6	D5	D4	D3	D2	D1	D0

Hiển thị	Anod chung	Cathode chung
0	C0h	3Fh
1	F9h	06h
2	A4h	5Bh
3	B0h	4Fh
4	99h	66h
5	92h	6Dh
6	82h	7Dh
7	F8h	07h
8	80h	7Fh
9	98h	67h
A	88h	77h
B	C6h	39h
C	86h	79h
D	8Eh	71h
E	82h	70h
F	89h	76h
.	7Fh	80h
[trắng]	FFh	00h

4.2 Phương pháp chốt

LED 1: 8000h
LED 2: A000h
LED 3: C000h
LED 4: E000h



Ví dụ 01

- Lập trình cho VXL 89c51 hiển thị lên LED 7 đoạn một số có 4 chữ số:

1486

2357

3769

1087

2789

```
#include "regx51.h" #include "absacc.h"
unsigned char x;    unsigned char y;
void main(void)
    {    while (1)
        {
        x=0x82;
        XBYTE[0XE000]=x; // xuat LED 4 ra gia tri 6
        x=0x80;
        XBYTE[0XC000]=x; // xuat LED 3 ra gia tri 8
        x=0x99;
        XBYTE[0XA000]=x; // xuat LED 2 ra gia tri 4
        x=0xF9;
        XBYTE[0X8000]=x; // xuat LED 1 ra gia tri 1
        }}
}
```

```
#include "regx51.h" #include "absacc.h"
unsigned char x;    unsigned char y;
unsigned char z[10]={0xC0, 0xF9, 0xA4, 0xB0, 0x99, 0x92,
0x82, 0xF8, 0x80, 0x90 };
void main(void)
    {    while (1)
        {
        x=6;
        XBYTE[0XE000]=z[x];
        x=8;
        XBYTE[0XC000]=z[x];
        x=4;
        XBYTE[0XA000]=z[x];
        x=1;
        XBYTE[0X8000]=z[x];
        }}
}
```


Ví dụ 02

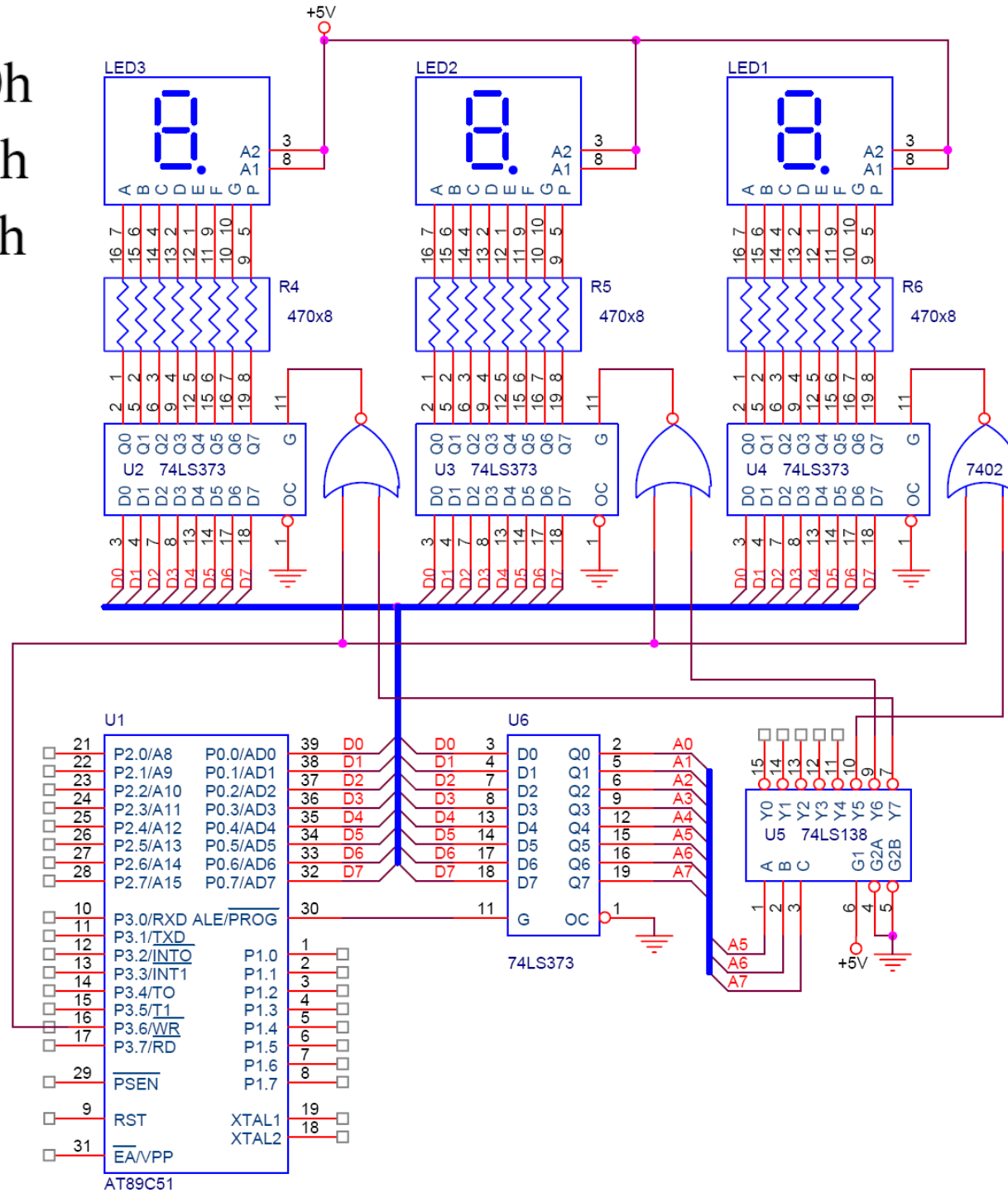
- Trong ô nhớ địa chỉ 31H của RAM chứa số hàng ngàn và hàng trăm, còn ô nhớ 30H chứa số hàng chục và hàng đơn vị.
- Lập trình cho VXL 89c51 hiển thị lên LED 7 đoạn!

```

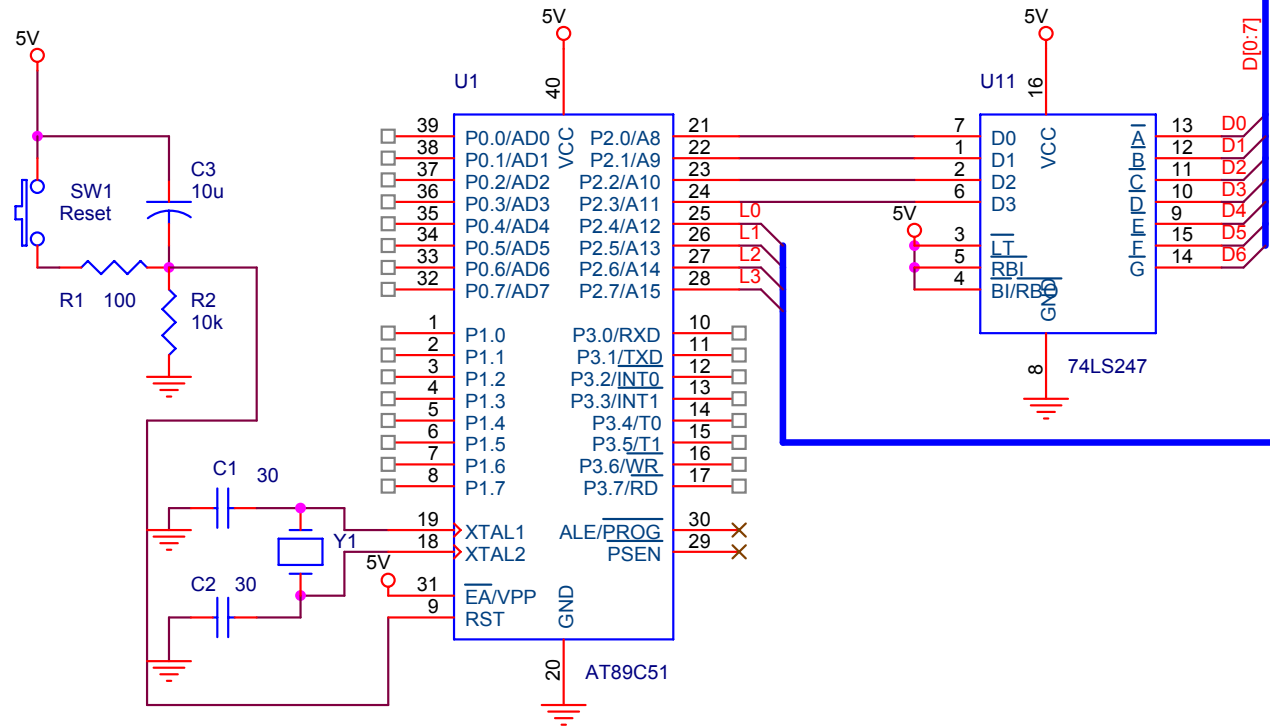
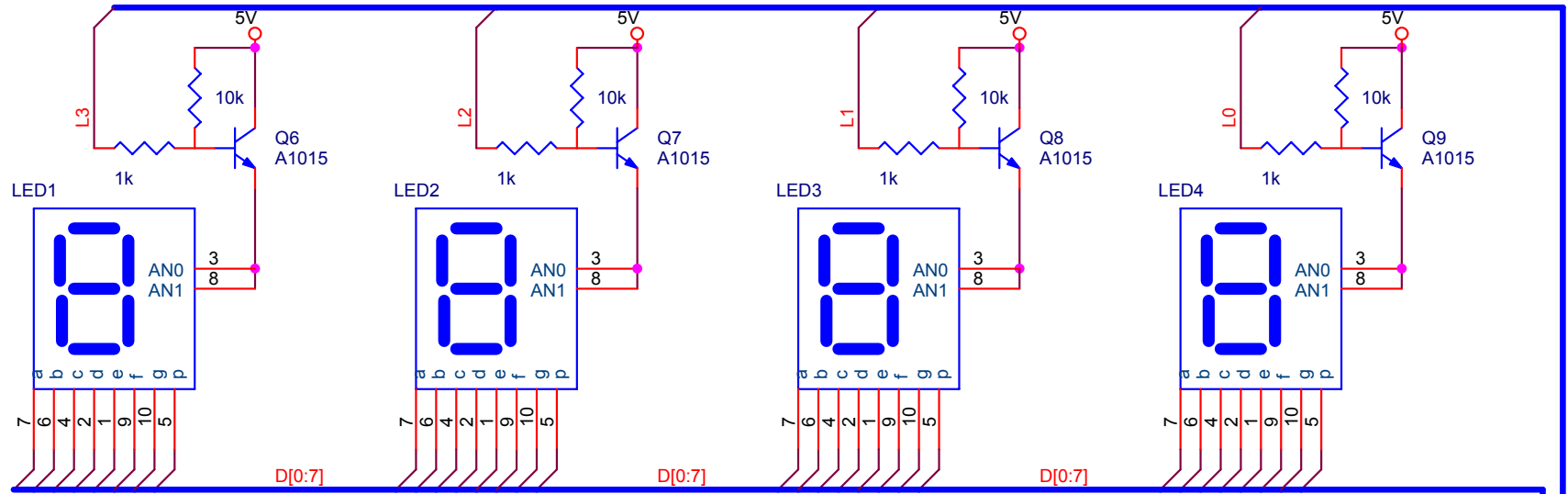
#include "regx51.h"  #include "absacc.h"  unsigned char x,y;
unsigned char z[10]={0xC0, 0xF9, 0xA4, 0xB0, 0x99, 0x92, 0x82,
0xF8, 0x80, 0x90 };
void main(void)
    {
        while (1)
            {
                y=CBYTE[0x0030];          x= y & 0x0F;
                XBYTE[0XE000]=z[x];      x=y & 0xF0;
                x>>=4;
                XBYTE[0XC000]=z[x];      y=CBYTE[0x0031];
                x=y & 0x0F;
                XBYTE[0XA000]=z[x];      x=y & 0xF0;
                x>>=4;
                XBYTE[0X8000]=z[x]; // xuat LED 1 ra gia tri
hang ngan
            }
    }

```

LED1: A0h
 LED2: C0h
 LED3: E0h

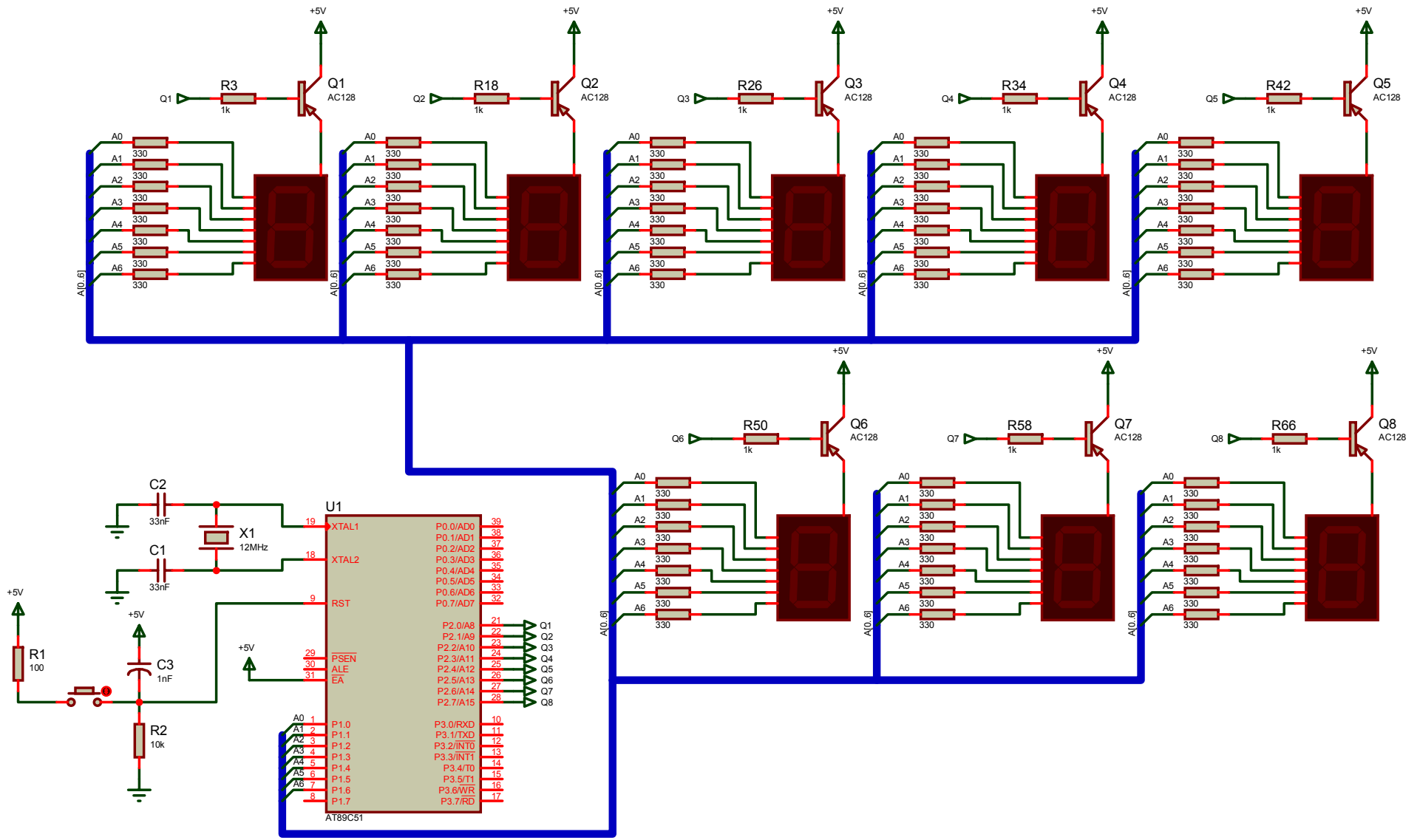


4.3 Phương pháp quét



Mã nhị phân trên port P1

P1.7 LED1	P1.6 LED2	P1.5 LED3	P1.4 LED4	P1.3	P1.2	P1.1	P1.0
1	1	1	0	4 bit BCD hàng đơn vị			
1	1	0	1	4 bit BCD hàng đơn chục			
1	0	1	1	4 bit BCD hàng trăm			
0	1	1	1	4 bit BCD hàng ngàn			



Ví dụ 03

- Xuất ra LED 7 đoạn hiển thị số 12345678.
Chú ý: 8051 sử dụng thạch anh 12 MHz.


```

#include "regx51.h"  unsigned char x;
unsigned char z[10]={0xC0, 0xF9, 0xA4, 0xB0, 0x99, 0x92,
0x82, 0xF8, 0x80, 0x90 };
unsigned char y[8] = {254, 253, 251, 247, 239, 223, 191, 131};
void Delay();
void main(void)
{
    P2=0x00;    P1=0x00;

while (1)
    {
        P2=0;
        for (x=1;x<8;x++)
            {P1=z[x];    P2=y[x-1];    Delay(); }// Delay 10mS
            P2=0; } }

void Delay()
{TMOD &= 0xF0; TMOD |= 0x00; ET0 = 0;
TH0 = 0xD8; TL0 = 0xEF; TF0 = 0;    TR0 = 1;
while (TF0 == 0)  {} TR0 = 0; /* Stop Timer 0 */ }

```

4.4 Giao tiếp với LCD

- LCD viết tắt của từ Liquid Crystal Display (Màn hình tinh thể lỏng).
- LCD có nhiều loại 1x16, 2x16, 4x16, 2x40, 4x40...



Sơ đồ chân và chức năng của LCD HD44780

Chân số	Ký hiệu	Mô tả
1	Vss	Nối mass hoặc đất
2	Vcc	Nguồn nuôi +5V
3	Vee	Chỉnh độ tương phản (0 đến +5V)
4	RS	Lựa chọn thanh ghi lệnh/dữ liệu
5	RW	Đọc/ghi
6	EN	Cho phép
7	DB0	Đường dẫn dữ liệu 0
8	DB1	Đường dẫn dữ liệu 1
9	DB2	Đường dẫn dữ liệu 2
10	DB3	Đường dẫn dữ liệu 3
11	DB4	Đường dẫn dữ liệu 4
12	DB5	Đường dẫn dữ liệu 5
13	DB6	Đường dẫn dữ liệu 6
14	DB7	Đường dẫn dữ liệu 7

- RS (Register Select): Dùng để điều khiển quá trình gửi một byte lệnh hay 1 byte vào LCD. RS ở mức thấp (0) thì byte gửi tới là lệnh, RS ở mức cao (1) thì byte gửi tới là dữ liệu và được hiển thị trên LCD.
- RW (Read/Write): dùng để khẳng định quá trình ghi vào hay đọc ra. Khi RW ở mức thấp (0) thì thông tin trên bus dữ liệu sẽ ghi vào LCD, RW ở mức cao là đọc trạng thái của LCD.
- EN (Enable): Đây là chân điều khiển, dùng để cho LCD biết rằng LCD đang được nhận dữ liệu vào. Khi EN ở mức thấp (0) thì LCD cho phép nhận dữ liệu, khi EN ở mức cao thì LCD sẽ cho phép truyền thông tin trạng thái.
- Bus dữ liệu 4 bit D4-D7 hay 8 bit D0-D7 phụ thuộc vào chọn chế độ hoạt động của LCD.

Bảng mã ký tự của LCD

Upper 4bit Lower 4bit	LLLL	LLHL	LLHH	LHLL	LHLH	LHHL	LHHH	HLLL	HLLH	HLHL	HLHH	HHLL	HHLH	HHHL	HHHH
LLLL	CG RAM (1)														
LLLH	(2)														
LLHL	(3)														
LLHH	(4)														
LHLL	(5)														
LHLH	(6)														
LHHL	(7)														
LHHH	(8)														
HLLL	(1)														
HLLH	(2)														
HHL	(3)														
HHLH	(4)														
HHL	(5)														
HHLH	(6)														
HHL	(7)														
HHLH	(8)														

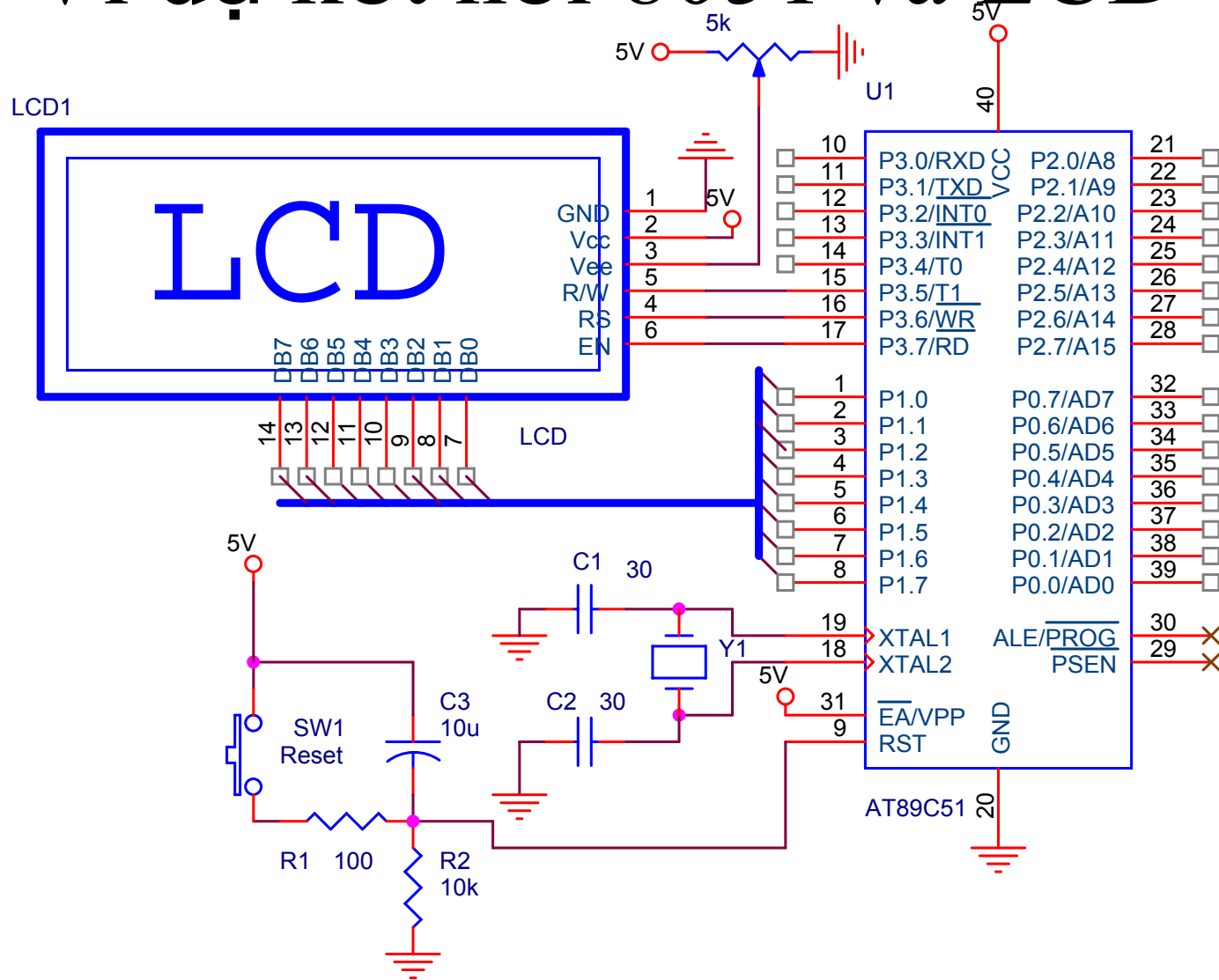
Lệnh của LCD

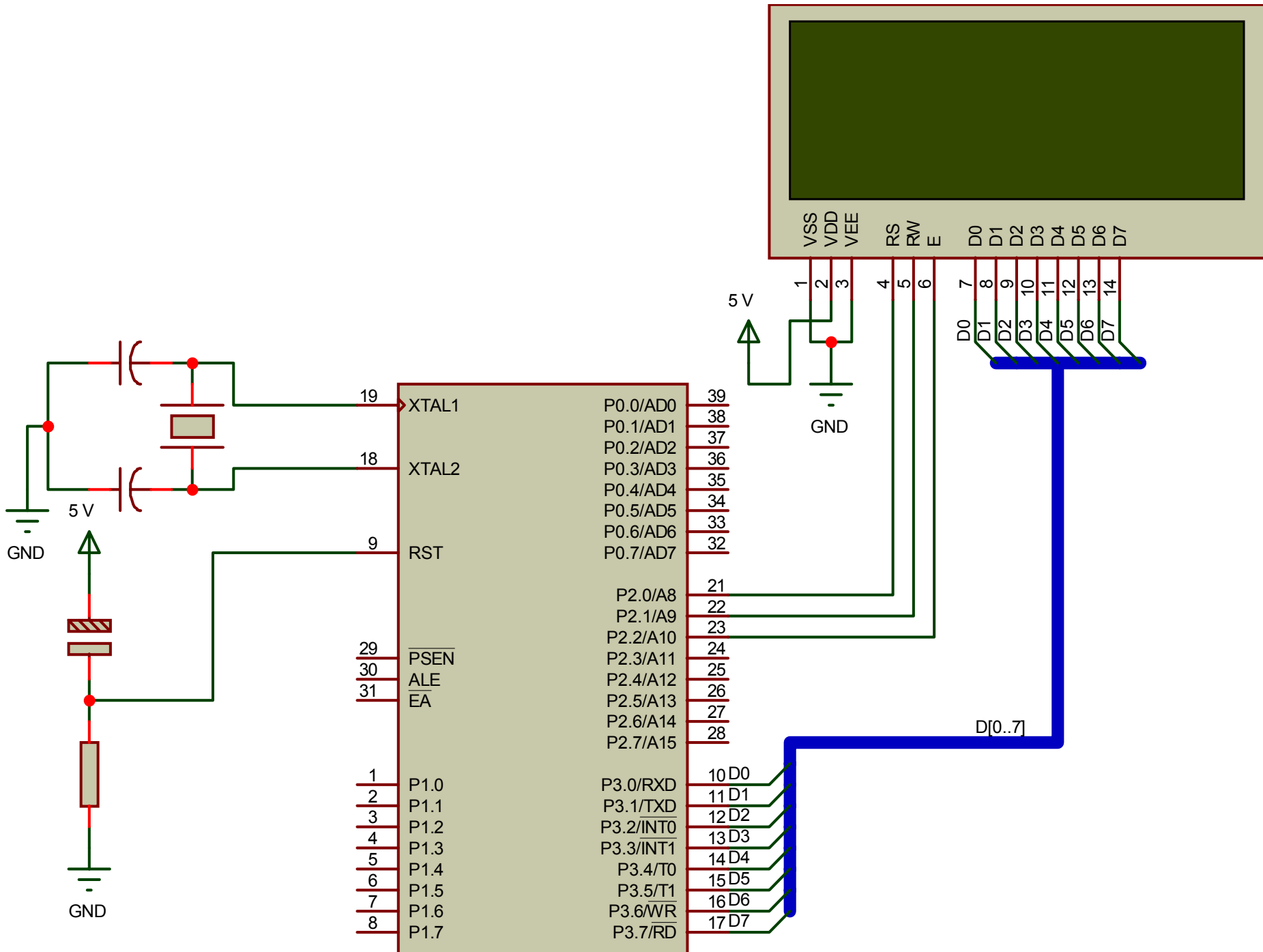
Chức năng	Lệnh
8-bit interface, 2 lines, 5*7 Pixels	38h
8-bit interface, 1 line, 5*7 Pixels	30h
4-bit interface, 2 lines, 5*7 Pixels	28h
4-bit interface, 1 line, 5*7 Pixels	20h
Dịch tất cả các dòng sang bên phải 1 ký tự	1Eh
Dịch tất cả các dòng sang bên trái 1 ký tự	18h
Đưa con trỏ về vị trí top/left	2h
Đưa con trỏ sang trái 1 ký tự	10h
Đưa con trỏ sang phải 1 ký tự	14h
Bật mở con trỏ dạng gạch chân ký tự	0Eh

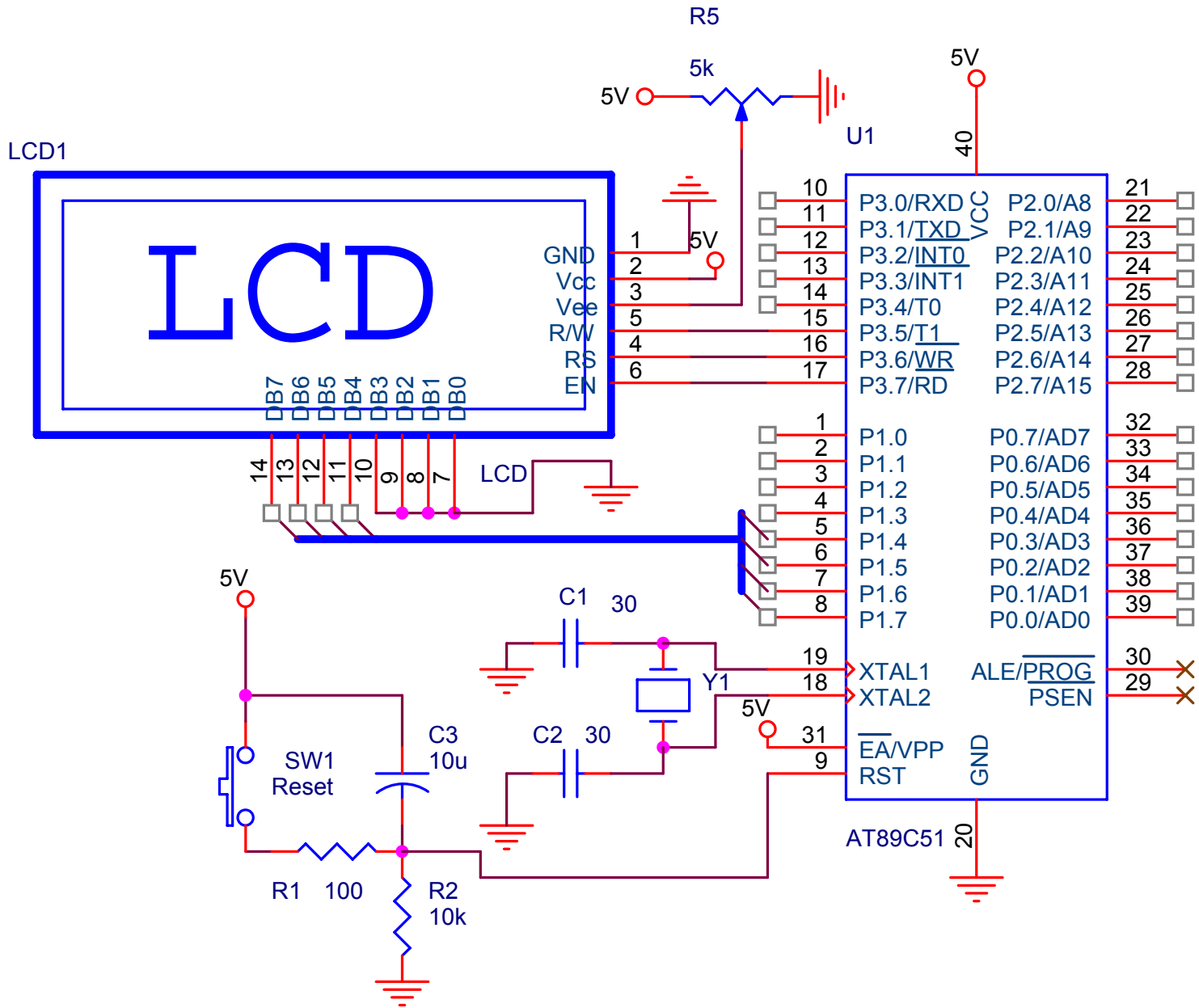
Lệnh của LCD

Chức năng	Lệnh
Bật mở con trỏ dạng nhấp nháy	0Fh
Tắt con trỏ	0Ch
Đề trống hiển thị (không xóa)	08h
Xóa màn hình hiển thị	01h
Xác định vị trí con trỏ	80+địa chỉ

Ví dụ kết nối 8051 và LCD







Định nghĩa các dòng lệnh cho LCD

```
typedef unsigned char bit8;    // dữ liệu 8 bit
/* các dòng lệnh cho LCD */
const bit8 spec = 0x38;        // ô hiển thị có 5x8 điểm ảnh
const bit8 LcdCur_ON = 0x0E;   // bật tắt con trỏ
const bit8 Lcd_Clr = 0x01;     // xoá hết LCD
const bit8 Curser_Right = 0x06; // di dời con trỏ sang phải
const bit8 Curser_HOME = 0x02; // di dời con trỏ về vị trí đầu
```

```
/* Định nghĩa chân tín hiệu */  
sbit RS = P2^0;    // Chân reset  
sbit RW = P2^1; //Chân đọc viết Read/write  
sbit E = P2^2;    // Chân enable/disable  
sbit busyCheck = P3^7; //Chân kiểm tra bận
```

```
/* Chương trình con kiểm tra trạng thái bận của LCD */  
void getReady()  
{  
    E = 0;          /* Cho LCD hoạt động */  
    busyCheck = 1; /* Bit bận = 1 */  
    RS = 0;        /* Ra lệnh cho LCD */  
    RW = 1;        /* LCD đọc lệnh */  
    while (busyCheck != 0) /* Đợi cho đến khi LCD hết bận */  
    {  
        E = 0;  
        E = 1;  
    }  
    E = 1;  
}
```

```
/* chương trình con xuất mã lệnh cho LCD */  
void writeCmd(bit8 cmd)  
{  
    getReady();           /* kiểm tra trạng thái bận ?*/  
    E = 0;                /* Cho phép LCD hoạt động*/  
    P3 = cmd;             /* mã lệnh gửi tới Port3 */  
    RS = 0;              /*Chọn thanh ghi*/  
    RW = 0;              /* Cho phép ghi vào LCD */  
    E = 1;  
    E = 0;  
}
```

```
/* chương trình con cài đặt ban đầu cho  
LCD */  
void init()  
{  
writeCmd(spec);  
writeCmd(LcdCur_ON);  
writeCmd(Lcd_Clr);  
writeCmd(Curser_Right);  
writeCmd(Curser_HOME);  
}
```

vị trí con trỏ LCD

Display	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16						
Line 1	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	...
Line 2	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	54	55	...

Lệnh xác định vị trí con trỏ LCD: $80h + \text{địa chỉ con trỏ}$

```
/*Chương trình con di dời vị trí con trỏ*/void  
LcdMoveCursor(unsigned char row, unsigned char col)  
{  
    writeCmd((0x80) + (row << 6) + col);  
}
```



```
/* Chương trình con hiển thị một đoạn chữ lên LCD*/  
/* Khai báo con trỏ 8 bit ban đầu*/  
bit8 code *text;  
/* Chương trình con*/  
void writeString(bit8 *str)  
{  
    bit8 i;  
        unsigned int j;  
        getReady();  
        E = 0;  
        for(i=0;str[i]!='\0';i++)  
            {  
                P3=str[i];  
                RS = 1;  
                RW = 0;  
                E = 1;  
                E = 0;  
                for(j = 0; j<25000; j++);  
            }  
}
```

/* Chương trình con hiển thị một ký tự (char) lên LCD */

void writeChar(unsigned char x)

{

unsigned int j;

unsigned char i;

i=0;

i=x+0x30; */* Chuyển đổi Char qua mã ASCII*/*

getReady();

E = 0;

P3=i; */* Xuất mã ASCII của ký tự ra LCD*/*

RS = 1;

RW = 0;

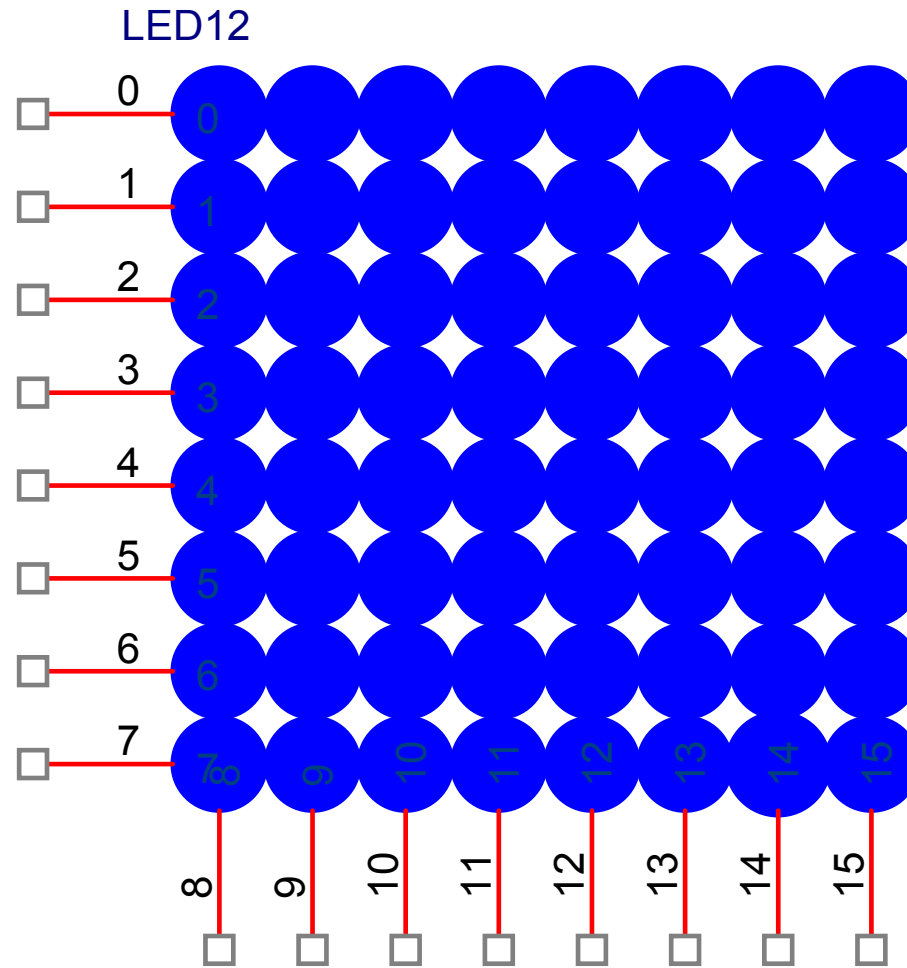
E = 1;

E = 0;

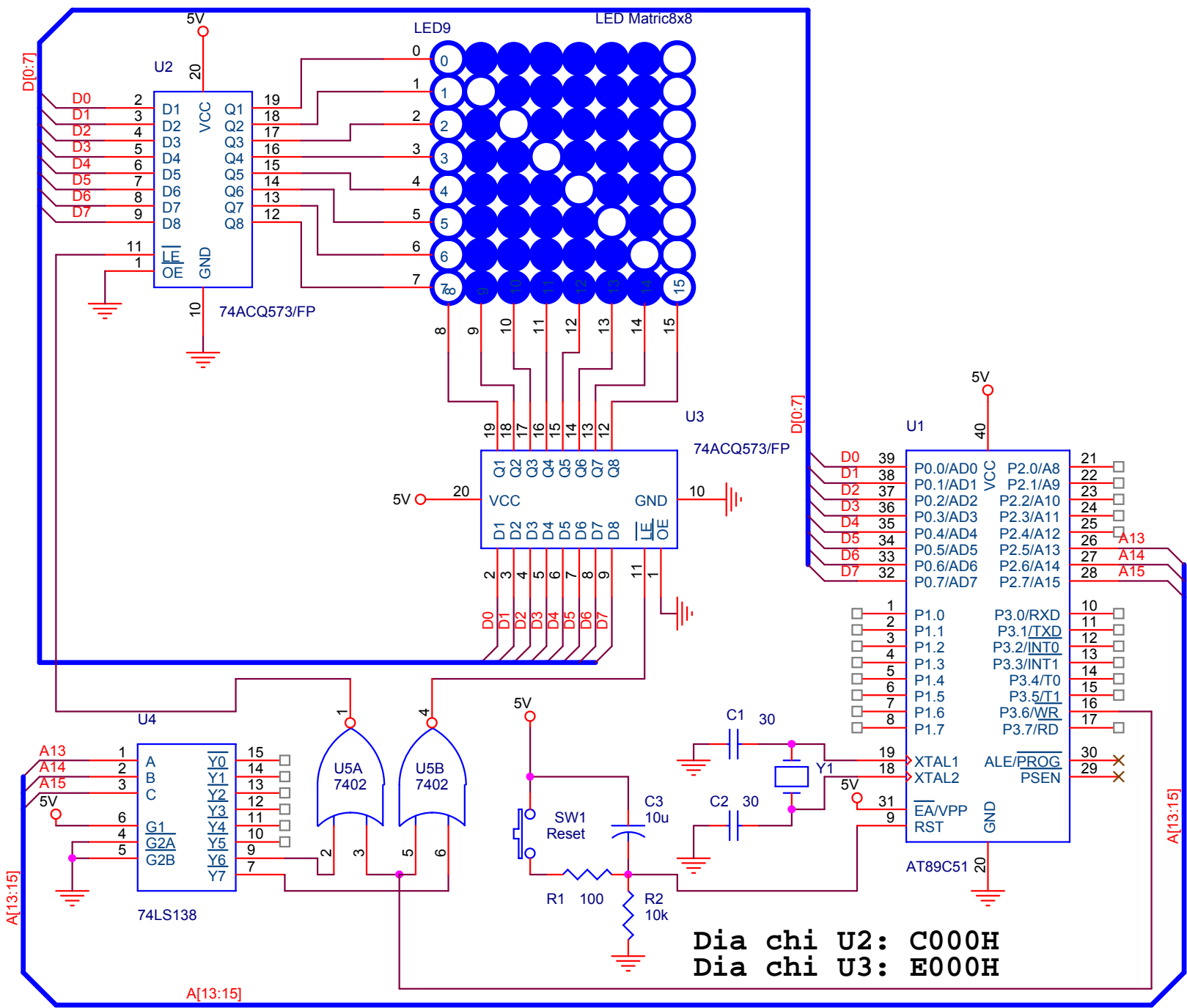
for(j = 0; j<25000; j++);

}

4.4 LED ma trận



LED Matrix 8x8



Dia chi U2: C000H
 Dia chi U3: E000H