6.2.2 数码管的静态驱动和动态驱动

1、数码管的静态驱动

- 所谓静态显示，就是数码管的各笔划段都由具有锁存能力的I/O端口引脚驱动，CPU将段码写入锁存器后，每个数码管都由锁存器的输出信号持续驱动。直到下一次CPU更新锁存器存储的段码之前，数码管的显示不会改变;
- 当需要用静态显示的方法驱动多个数码管时，就需要使用多个具有锁存能力的I/O端口，每个端口驱动一个数码管的显示。

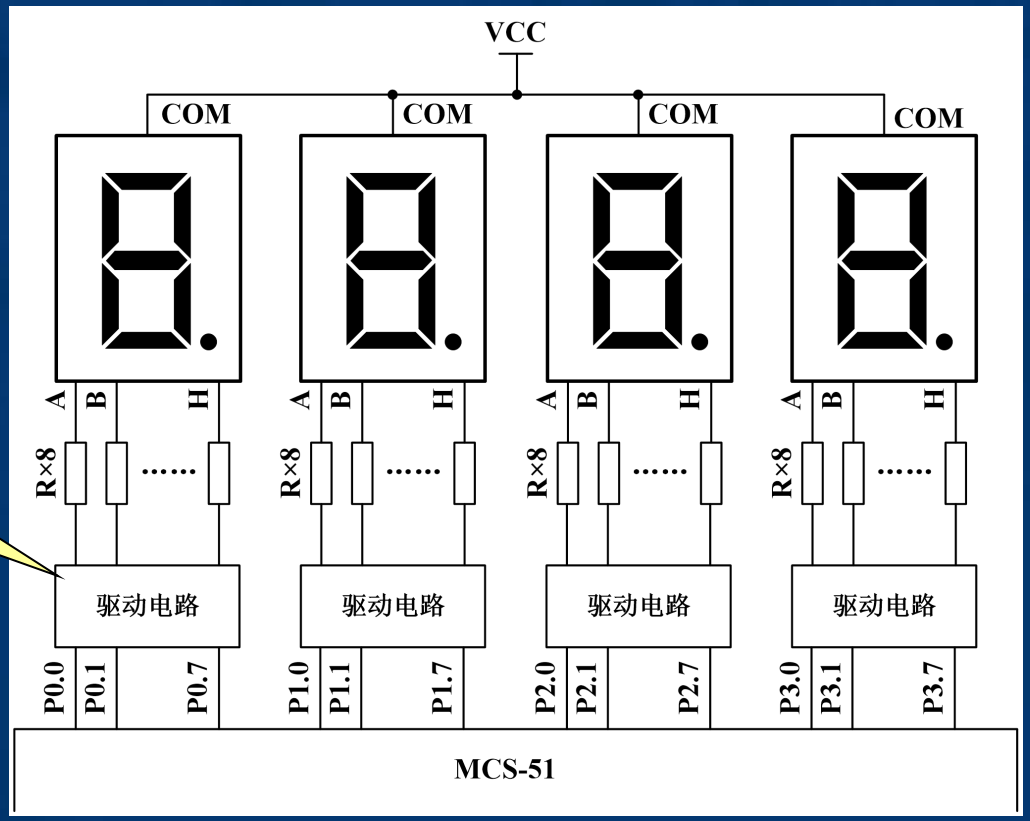
南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.2.2 数码管的静态驱动和动态驱动

1、数码管的静态驱动

- 多个共阳型数码管的静态驱动电路

单片机端口的总输入电流有限制，因此当数码管数目较多时，必须通过驱动电路驱动数码管



南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.2.2 数码管的静态驱动和动态驱动

2、数码管的动态驱动

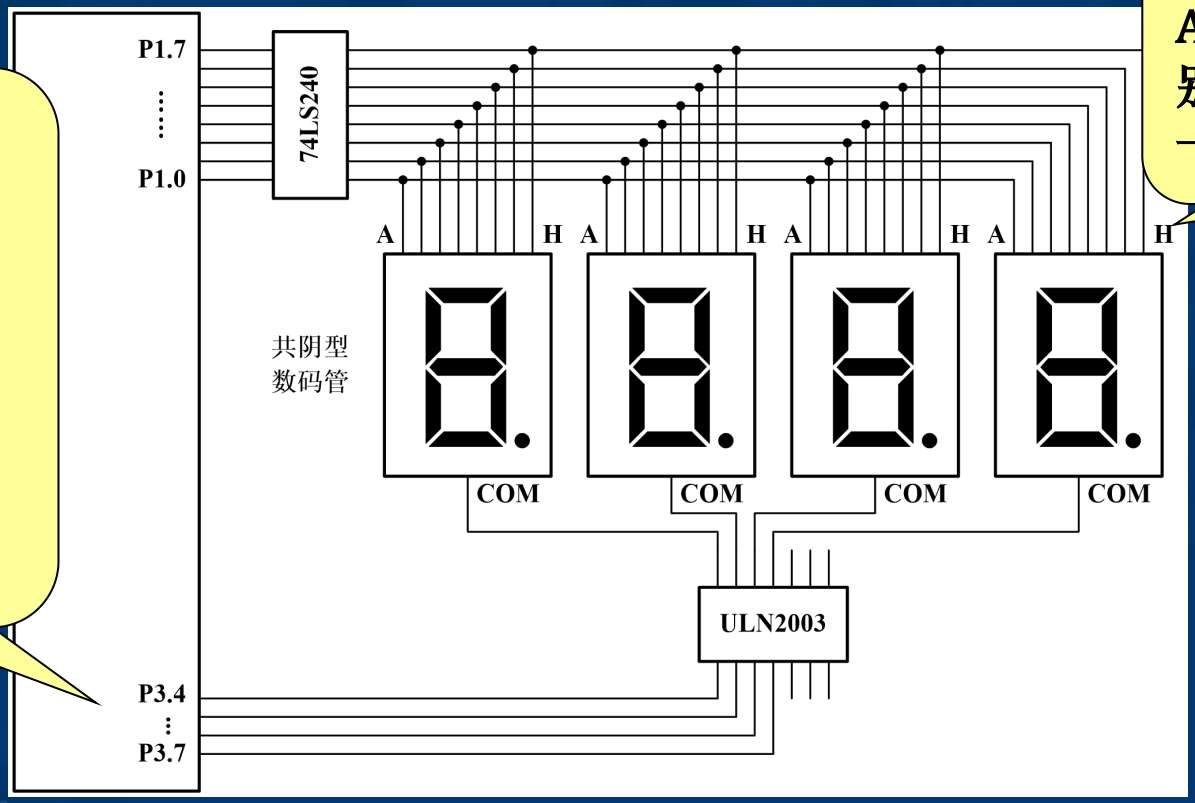
- 所谓动态显示驱动，就是通过软件，间隔固定时间对每一位数码管轮流驱动，使其交替点亮；
- 动态显示驱动利用了人眼的“视觉暂留”现象，只要数码管点亮的间隔小于人眼的视觉暂留时间(约40ms)，人们就会认为数码管是一直点亮的；
- 由于每次驱动只点亮一个数码管，因此驱动电路可大大简化——所有笔划驱动可以同名复接在一起。为了控制数码管轮流点亮，应增加位显示驱动控制线。具体电路如下页图所示：

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.2.2 数码管的静态驱动和动态驱动

2、数码管的动态驱动

- 多个共阴型数码管动态显示驱动电路



P3.4～P3.7轮流输出高，分别点亮1～4号数码管。每次点亮某个数码管前，应先通过P1端口输出该位置数码管的字形码

所有数码管的A、B、...H分别同名复接在一起

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.2.2 数码管的静态驱动和动态驱动

- 多个共阴型数码管动态显示驱动程序设计

```
#include <reg51.h>
#define SegCode P1
#define Position    P3
#define Dly         100
// 定义1234的字形码，存放于程序存储器空间
unsigned char code Seg[] = {0xF9,0xA4,0xB0,0x99};
unsigned char PosCtrl;                    // 定义位置控制码

void Delay(int i)                         // 循环延时函数
{
    int j;
    while(i--) for(j=0;j<Dly;j++);        // 循环延时，改变Dly的定义可调整延时时间
}

void main(void)
{
    unsigned char i;
    while(1)
    {
        for(i=0;i<4;i++)
        {
            Delay(10);
            Position &= 0x0F;             // P3高4位清0，关显示
            PosCtrl = 0x10;               // 显示位置控制码复原
            PosCtrl <<= i;                // 显示位置控制码左移i位
            SegCode = Seg[i];             // 送字形码
            Position |= PosCtrl;          // 当前显示位驱动有效，开显示
        }
    }
}
```


南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.2.2 数码管的静态驱动和动态驱动

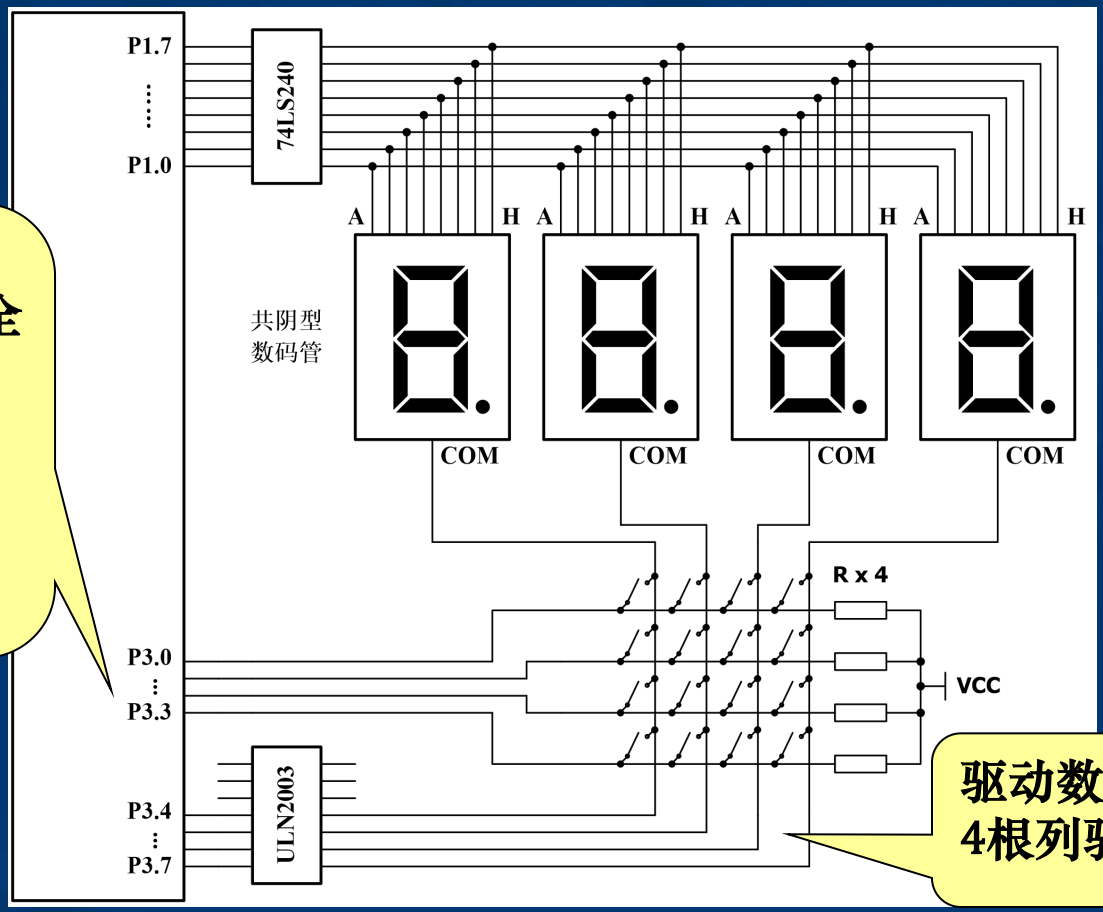
3、数码管动态显示驱动和按键扫描相结合

- 在进行数码管显示驱动时，位驱动线每次只有一根输出低电平，每隔固定时间（如10ms）移位一次，正好和行列式键盘扫描的逻辑相符；
- 将位驱动线复用为行列式键盘中的输出线，另外再设置和这些输出线相交叉的输入线，即可构成一个行列式键盘，电路如下图所示：

6.2.2 数码管的静态驱动和动态驱动

3、数码管动态显示驱动和按键扫描相结合

只要CPU读入P3.0～P3.3不为全高，则表示有键按下，结合当前为低的列驱动线即可判断出具体按键。



驱动数码管显示时，这4根列驱动线轮流变低

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

第六章 单片机输入输出接口及系统扩展设计

6.3 字符点阵LCD显示模块的控制——模拟总线时序驱动

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

LCD显示器简介

- LCD显示器是一种用液晶材料制成的液晶显示器，它具有体积小、功耗低、字迹清晰、无电磁辐射、使用寿命长等优点，因此广泛应用于各种手持式仪器仪表及消费类电子产品等低功耗应用场合中；
- LCD显示器的显示方式可分为字符点阵显示和图形点阵显示两种；
  - 字符点阵显示：字符由固定的5×7或5×10点阵构成，输入字符的ASCII码即可显示；
  - 图形点阵显示：没有固定的字符点阵，所有显示内容通过将点阵数据写入显示RAM构成，点阵数据中的“1”控制LCD显示一个点，“0”不显示，通过点阵的方式描画出字符、图形。

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

LCD显示器简介

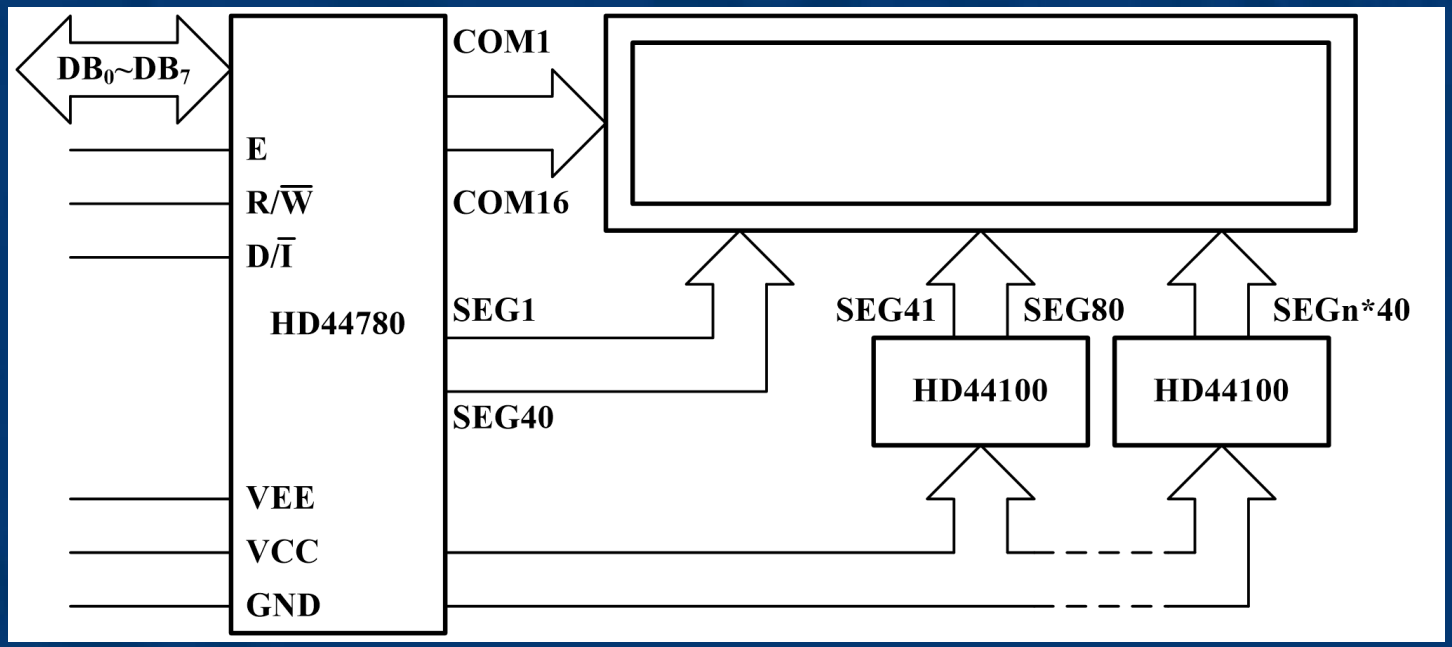
- 将LCD显示器、显示控制电路、CPU接口电路以及背光控制电路等装配在一起，就构成了液晶显示模块(LCM)；
- LCM具有CPU总线接口，可以连接在CPU总线上接受CPU的控制，完成显示工作；
- 对于没有外部总线扩展的CPU，也可以通过CPU的I/O端口引脚模拟CPU总线时序来操作LCM。

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.3.1 1602字符点阵LCM简介

1、LCM 1602的内部结构

- 1602字符点阵LCM有2个显示行，每行显示16个字符，显示控制器为HD44780，通过HD44100进行显示规模的扩展：



南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.3.1 1602字符点阵LCM简介

1、LCM 1602的内部结构

- LCM 1602引脚功能：
  - DB0～DB7为双向数据总线，用于和CPU交换数据；
  - E为总线周期有效指示，高电平有效；
  - R/$\overline{W}$为读写选择线，CPU送高电平表示对LCM显示控制器进行读操作，送低电平表示对LCM显示控制器进行写操作；
  - D/$\overline{I}$为寄存器选择线，CPU送高电平表示选择LCM的数据寄存器进行操作，送低电平表示对LCM的指令寄存器进行操作；
  - VEE为对比度调节控制信号，通过改变该引脚上的电压值可控制LCD显示内容的对比度；
  - VCC和GND为LCM的电源和地。

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.3.1 1602字符点阵LCM简介

1、LCM 1602的内部结构

- LCM 1602的内部寄存器：
  - 指令寄存器（IR）：用于存储CPU送达的指令代码，主要包括写入控制器的清屏及移动光标指令、设置显示地址指令、设置字形码指令等。IR是一个只写寄存器；
  - 数据寄存器（DR）：用于暂存CPU对控制器内部的DDRAM和CGRAM进行读写的数据；
  - 忙标志（BF）：当BF=1时，表示HD44780处于内部操作阶段，除了读忙标志指令之外，不接受任何其它指令。CPU在向HD44780发送指令前一定要先判断BF=0才可以继续写入指令；

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.3.1 1602字符点阵LCM简介

1、LCM 1602的内部结构

- LCM 1602的内部寄存器：
  - 地址计数器（AC）：指定被操作的DDRAM或CGRAM的地址。地址包含在指令里，首先由CPU写入指令寄存器IR，然后转存到地址计数器AC。指令写入控制器时将指明该地址到底是用于DDRAM还是CGRAM；
  - 当CPU对DDRAM或CGRAM的读写操作完成后，AC将根据自动增量或减量设置自动加1或减1。

6.3.1 1602字符点阵LCM简介

1、LCM 1602的内部结构

- LCM 1602的内部寄存器的控制：
  - CPU通过控制D/$\bar{I}$和R/$\bar{W}$选择要操作的寄存器：

D/$\bar{I}$	R/$\bar{W}$	操作对象
0	0	选择指令寄存器(IR)，进行写入操作
0	1	读出忙标志和地址计数器
1	0	选择数据寄存器(DR)，进行写入操作
1	1	选择数据寄存器(DR)，进行读出操作

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.3.1 1602字符点阵LCM简介

1、LCM 1602的内部结构

- LCM 1602的内部存储器：
  - 显示数据存储器（DDRAM）：存放对应位置要显示的数据的ASCII码或字形码。对于1602LCD显示模块，第一行16个显示字符的ASCII码或字形码存放在DDRAM中地址0x00开始的16个单元中，第二行16个显示字符的ASCII码和字形码则存放在DDRAM中地址0x40开始的16个单元中。

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.3.1 1602字符点阵LCM简介

1、LCM 1602的内部结构

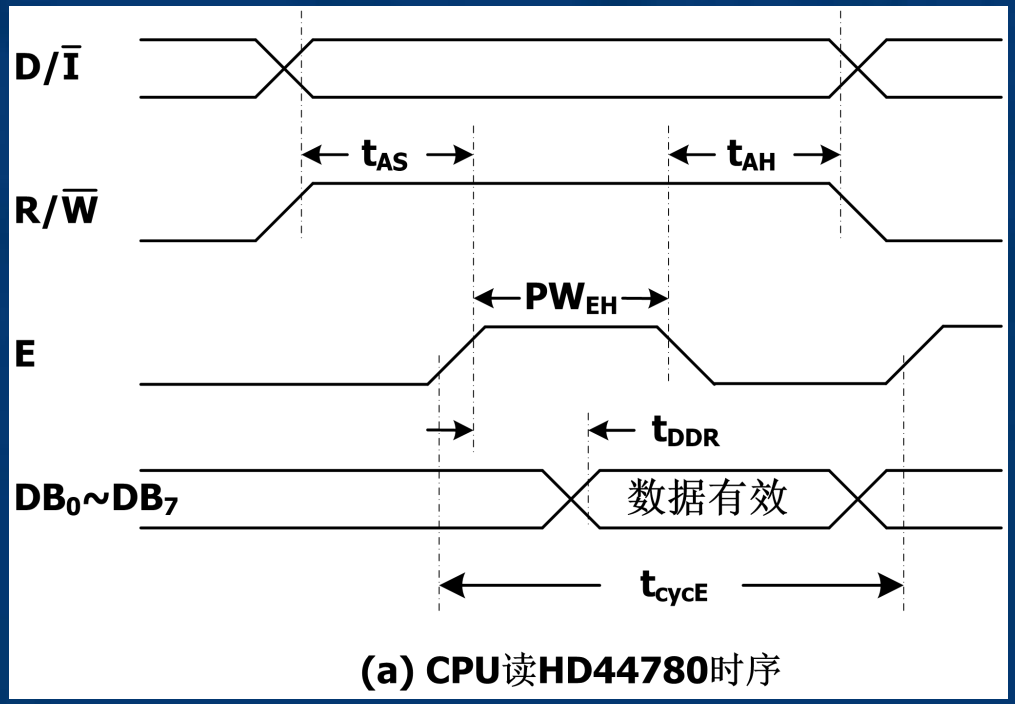
- LCM 1602的内部存储器：
  - 存放用户自行设计的字符点阵数据。CGRAM可存储8个5×8点阵字符的字形点阵数据，或者4个5×10点阵字符的字形点阵数据。这些字符在HD44780内部的编码从0x00开始排列；
  - CGRAM中的点阵数据写入后将一直保持，显示自定义字符时只要将相应的编码写入显示位置对应的DDRAM中即可；
  - LCM掉电后CGRAM中的数据不保存，系统断电再开机后必须重新写入。

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.3.1 1602字符点阵LCM简介

2、LCM 1602的操作时序

- CPU对HD44780进行一次总线读操作的时序为：



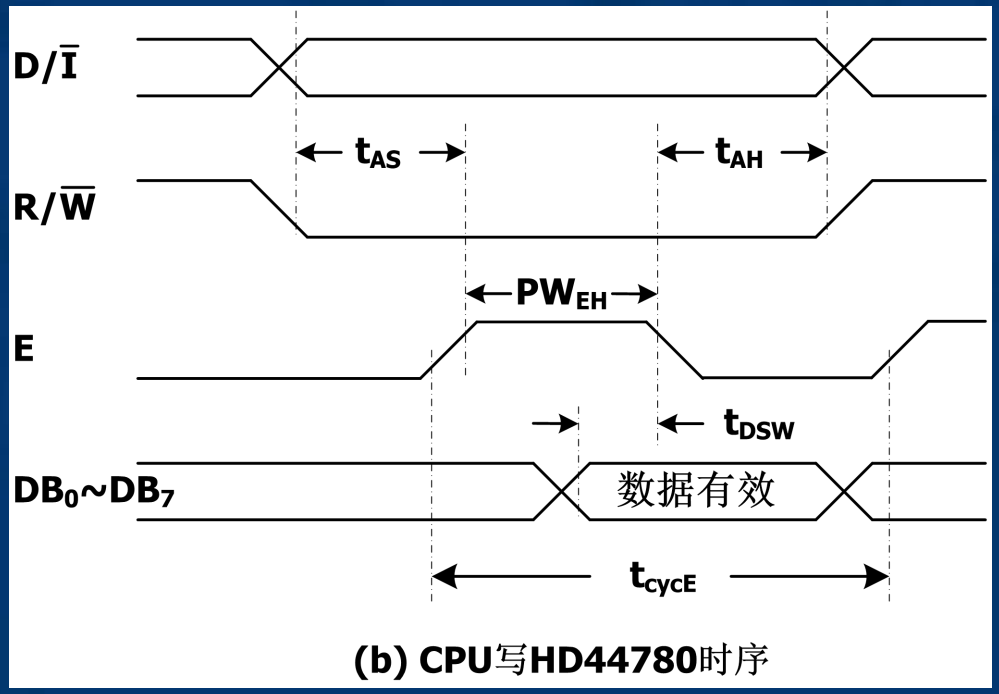
(a) CPU读HD44780时序

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.3.1 1602字符点阵LCM简介

2、 LCM 1602的操作时序

- CPU对HD44780进行一次总线写操作的时序为：



(b) CPU写HD44780时序

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.3.1 1602字符点阵LCM简介

2、LCM 1602的操作时序

- CPU对HD44780进行总线读写操作时的各种时间参数的大小及含义分别为：

时间段	最小值	典型值	最大值	单位	信号含义
t_{cycE}	1000	-	-	ns	E信号周期
PW_{EH}	450	-	-		E信号周期中高电平保持时间
t_{AS}	60	-	-		地址建立时间（D/$\bar{I}$及R/$\bar{W}$有效到E有效之间）
t_{AH}	20	-	-		地址保持时间
t_{DDR}	-	-	360		数据延迟时间（控制有效到读数据有效之间）
t_{DSW}	195	-	-		数据准备时间（数据有效到E无效之间）

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.3.1 1602字符点阵LCM简介

3、LCM 1602的控制指令

- CPU通过HD44780的R/$\overline{W}$和D/$\overline{I}$引脚以及数据总线DB0～DB7来控制LCM的显示。具体控制指令如下：

指令说明	指令代码										说明
	D/$\overline{I}$	R/$\overline{W}$	D7	D6	D5	D4	D3	D2	D1	D0	
清屏	L	L	0	0	0	0	0	0	0	1	清除屏幕,置AC为0
返回	L	L	0	0	0	0	0	0	1	x	设DDRAM地址为0，显示回原位，DDRAM内容不变
输入方式设置	L	L	0	0	0	0	0	1	D/I	S	设光标移动方向，并指定整体是否移动

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.3.1 1602字符点阵LCM简介

3、LCM 1602的控制指令

指令说明	指令代码										说明
	D/$\overline{I}$	R/$\overline{W}$	D7	D6	D5	D4	D3	D2	D1	D0	
显示开关控制	L	L	0	0	0	0	1	D	C	B	设置整体显示的开关(D),光标的开关(C),光标位置的字符是否闪烁(B)
移位	L	L	0	0	0	1	S/C	R/L	x	x	移动光标或显示区,不改变DDRAM内容
功能设置	L	L	0	0	1	DL	N	F	x	x	设接口数据位数(DL),显示行数(N)及字形(F)
CGRAM地址设置	L	L	0	1	6位CGRAM地址						设置CGRAM地址，此后CPU对DR的读写将影响CGRAM
DDRAM地址设置	L	L	1	7位DDRAM地址							设置DDRAM地址，此后CPU对DR的读写将影响DDRAM

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.3.1 1602字符点阵LCM简介

3、LCM 1602的控制指令

指令说明	指令代码										说明
	D/$\bar{I}$	R/$\bar{W}$	D7	D6	D5	D4	D3	D2	D1	D0	
读忙信号及地址计数器	L	H	BF	7位AC							判断忙信号位（BF）,并读地址计数器AC的内容
写数据:	H	L	数据								向CGRAM或DDRAM写数据
读数据:	H	H	数据								从CGRAM或DDRAM读数据
	I/D 1: 增量方式 0: 减量方式 S 1: 移位 S/C 1: 显示移位 0: 光标移位 R/L 1: 右移 0: 左移 DL 1: 8位数据总线 0: 4位数据总线 N 1: 2行 0: 1行										DDRAM :显示数据RAM CGRAM :字符生成RAM AC :用于DD/CGRAM的地址计数器 F 1: 5x10点阵 0: 5x7点阵 BF 1: 内部操作(忙) 0: 可接收命令(闲)

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

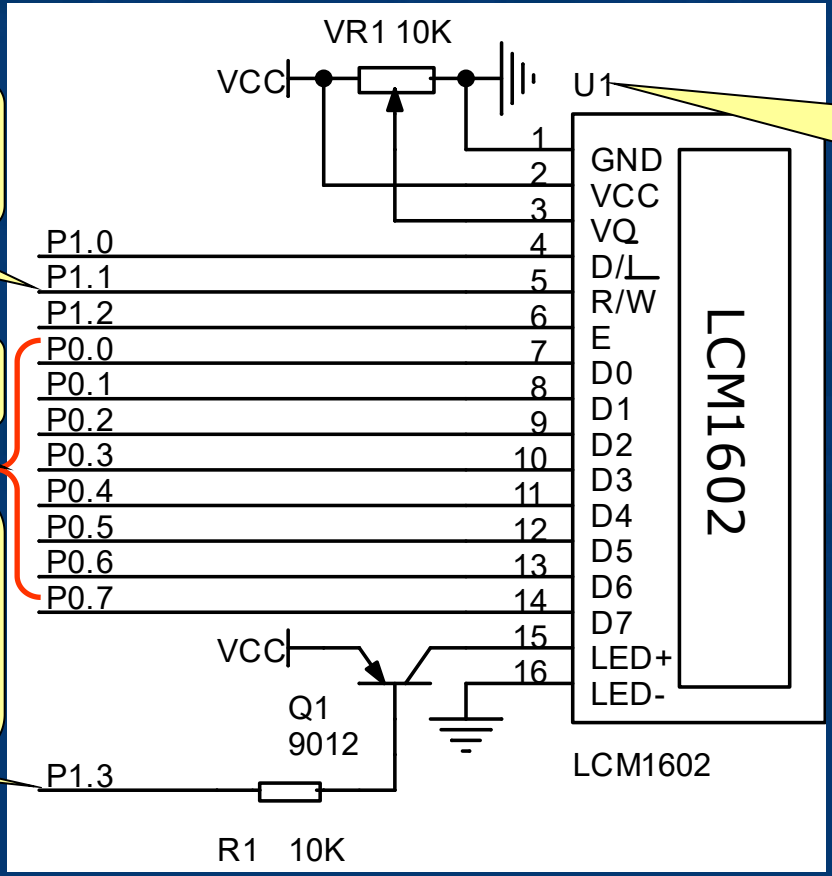
6.3.2 1602字符点阵LCM与单片机的接口

1602字符点阵LCM与单片机的接口原理图

P1口的低3位作为三根控制总线

P0口作为数据总线

P1.3为LCM的背光控制：
P1.3输出1时熄灭背光
P1.3输出0时点亮背光



可变电阻VR1控制LCM的显示对比度

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.3.2 1602字符点阵LCM与单片机的接口

1602字符点阵LCM与单片机的接口程序示例

```
#include <reg51.h>
#include <intrins.h>
#define LCM_DI      P10
#define LCM_RW      P11
#define LCM_E       P12
#define LCM_DB      P0

unsigned char LCMReadState(void)
{
    unsigned char state;

    LCM_DB = 0xff;          // 并行端口必须先置1再读
    LCM_E = 0;              // 开始模拟读IR的总线时序
    LCM_DI = 0;
    LCM_RW = 1;
    LCM_E = 1;              // E=1，有效总线周期开始
    _nop_();                // 延时一个机器周期等待
                            // LCM准备好数据
    state = LCM_DB;         // 读数据总线
    LCM_E = 0;              // E=0，结束总线周期
    return state;
}
```


南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

第六章 单片机输入输出接口及系统扩展设计

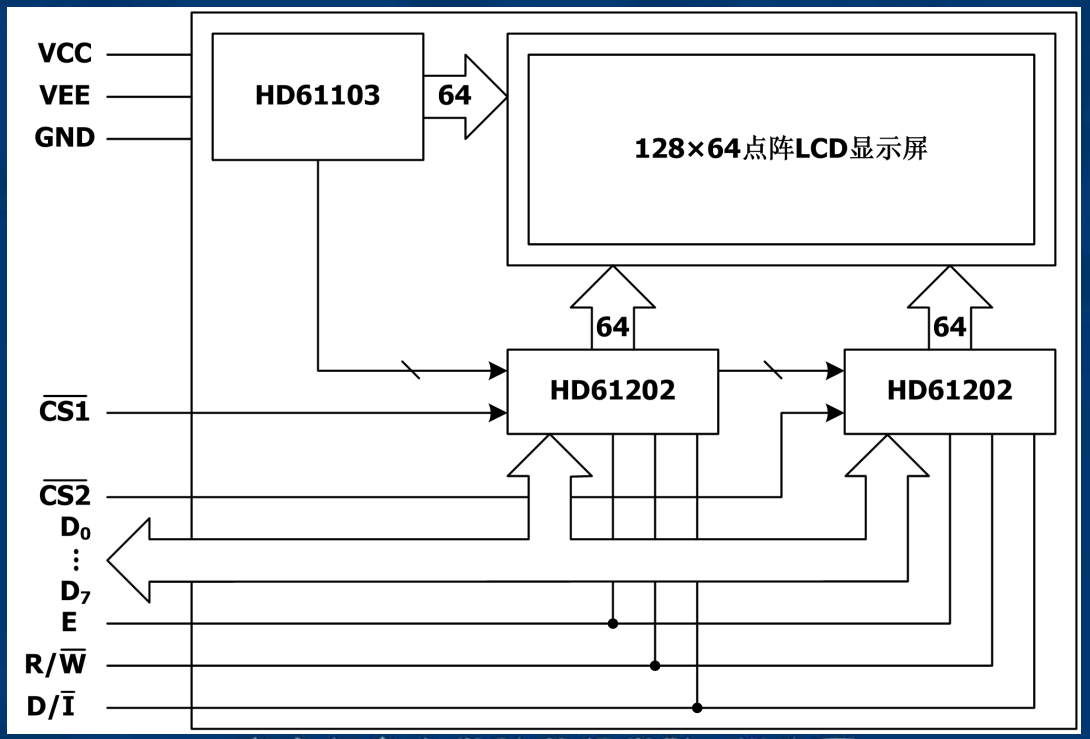
6.4 图形点阵LCD显示模块的控制——扩展总线时序驱动

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.4.1 128×64图形点阵LCM简介

1、LCM 12864的内部结构

- 12864图形点阵LCM采用两片HD61202U和一片HD61203U相配合驱动128×64点阵的LCD显示器。其内部结构为：



南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.4.1 128×64图形点阵LCM简介

1、LCM 12864的内部结构

- LCM 12864的引脚功能
  - DB0～DB7为双向数据总线，用于和CPU交换数据；
  - E为总线周期有效指示，高电平有效；
  - R/$\overline{W}$为读写选择线，CPU送高电平表示对LCM进行读操作，送低电平表示对LCM进行写操作；
  - D/$\overline{I}$为寄存器选择线，CPU送高电平表示选择LCM的数据寄存器进行操作，送低电平表示对LCM的指令寄存器进行操作；
  - VEE为对比度调节端，可控制LCD显示内容的对比度；
  - $\overline{CS1}$和$\overline{CS2}$：低有效，对LCM内部的两片HD61202U进行片选，$\overline{CS1}$有效选择左半屏控制器，$\overline{CS2}$有效选择右半屏控制器；
  - VCC和GND为LCM的电源和地。

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.4.1 128×64图形点阵LCM简介

1、LCM 12864的内部结构

- LCM 12864的内部寄存器及控制
  - 输入/输出寄存器：为CPU和LCM内部显示RAM之间数据传送的暂存器。CPU写入显示RAM的数据首先送到输入寄存器，然后由芯片内部操作过程自动送入显示RAM；
  - 当满足$\overline{CS1}$或$\overline{CS2}$有效并且R/$\overline{W}$、D/$\overline{I}$的组合选择了输入寄存器时，数据总线上的数据将在E信号的下降沿锁存；
  - 输出寄存器用来暂存从显示RAM中读出的数据；
  - 当满足$\overline{CS1}$或$\overline{CS2}$有效并且R/$\overline{W}$、D/$\overline{I}$均为1的条件时，存放在输出寄存器中的数据在E为高电平时输出。在E的下降沿，指定地址的数据被锁存在输出寄存器中，同时地址加1。

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.4.1 128×64图形点阵LCM简介

1、LCM 12864的内部结构

- LCM 12864的内部寄存器及控制
  - 忙标志：为1时表示HD61202U正在进行内部操作，除了读状态指令外，其余指令都不接受。忙标志在读出数据的D7表示。在输入任何指令前，应确认此位为0。

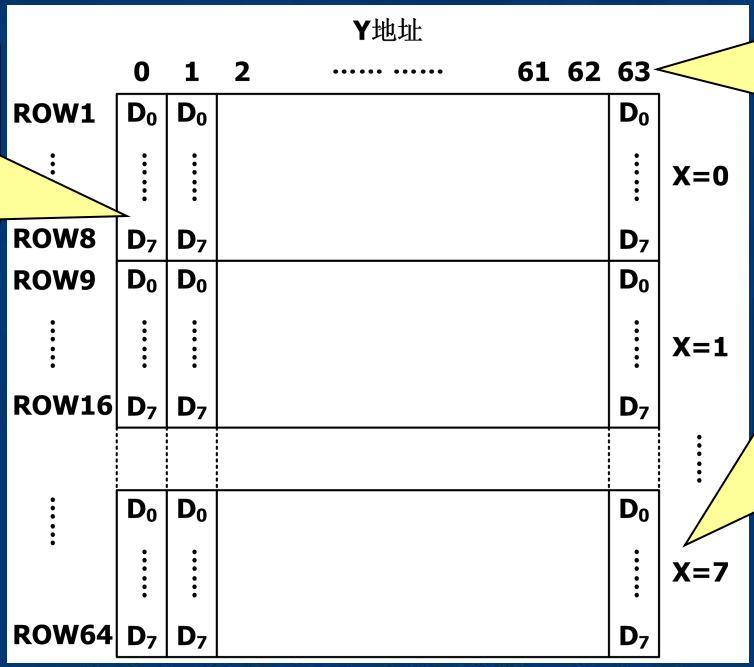
南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.4.1 128×64图形点阵LCM简介

1、LCM 12864的内部结构

- LCM 12864的内部寄存器及控制
  - 显示存储器：每一片HD61202U负责驱动64×64点阵的LCD显示器，其内部显示RAM的地址结构如下图所示：

某列的8行（位）数据对应一个字节，该字节中为1的位显示点，为0的位不显示



共64列，每列分成8个大行，每行对应显示RAM中的一个字节

一片HD61202U内部的RAM容量为512字节,按行的方向（横向）分为8页（页地址X=0~7），每页64字节（和列一一对应）

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.4.1 128×64图形点阵LCM简介

1、LCM 12864的内部结构

- LCM 12864的内部寄存器及控制
  - X、Y地址计数器：X，Y地址计数器是和内部512字节的显示RAM相对应的9bit的计数器。高3位为X地址计数器，低6位为Y地址计数器，根据指令的不同可以分别设置成X地址（页地址）计数器和Y地址（列地址）计数器。X地址计数器只能作为普通的无计数功能的寄存器使用，而Y地址则在CPU对显示数据进行读写操作后自动加1，并在0~63范围内循环。

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.4.1 128×64图形点阵LCM简介

1、LCM 12864的内部结构

- LCM 12864的内部寄存器及控制
  - 起始显示行寄存器：指定和LCD的第一行相对应的显示RAM的行号。此寄存器用于显示器的卷屏操作。通过设置起始显示行指令可以将6bit的起始显示行信息写入此寄存器；

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.4.1 128×64图形点阵LCM简介

1、LCM 12864的内部结构

- LCM 12864的内部寄存器及控制
  - 显示开关及翻转：通过选择Y1~Y64的开关状态来实现对LCM显示的控制；
  - 在开状态，显示RAM中为1的位会在点阵的对应位置上显示一个点；
  - 在关状态，所有的点都不显示；
  - 此功能通过显示器开/关指令控制。RST=0将显示器设置为关状态；
  - 当前显示状态通过读指令在D5位输出。显示开/关指令不影响显示RAM中的数据。

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.4.1 128×64图形点阵LCM简介

1、LCM 12864的内部结构

- LCM 12864的内部寄存器及控制
  - 复位：在LCM加电时将复位引脚（RST）接低将初始化LCM；
  - 所谓初始化是指关闭LCD的显示并将起始显示地址寄存器置0；
  - 当RST=0时，只能对LCM进行读状态操作；
  - 正常工作状态下，只有判断状态字的D4=0（RESET完成）和D7=0（就绪）时，程序方可输出其它的指令。

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.4.1 128×64图形点阵LCM简介

1、LCM 12864的内部结构

- LCM 12864内部寄存器的控制
  - CPU通过D/$\bar{I}$和R/$\bar{W}$操作LCM 12864的内部寄存器：

D/$\bar{I}$	R/$\bar{W}$	操作对象
0	0	选择指令寄存器（IR），进行写入操作
0	1	读出忙标志
1	0	选择输入寄存器，写数据到内部显示RAM
1	1	选择输出寄存器，从内部显示RAM读出数据

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.4.1 128×64图形点阵LCM简介

2、LCM 12864的控制指令

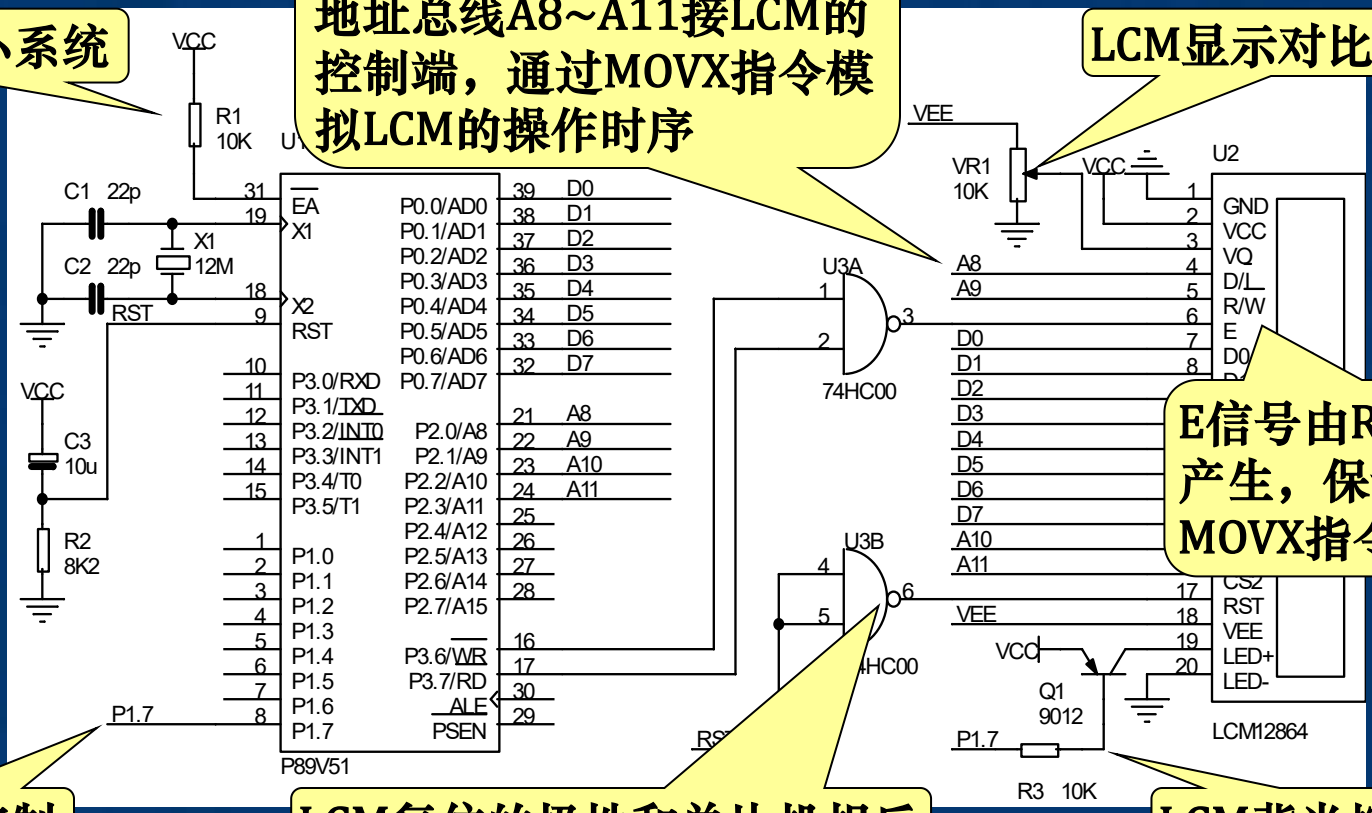
- CPU对HD61202的控制指令如下：

指令	指令代码										功能描述
	R/$\overline{W}$	D/$\overline{I}$	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
显示开/关	0	0	0	0	1	1	1	1	1	1/0	1:显示开 0:显示关
设置起始显示行	0	0	1	1	起始显示行(0~63)						设置起始显示行寄存器
设置页(X)地址	0	0	1	0	1	1	1	页号(0~7)			设置页寄存器
设置Y地址	0	0	0	1	Y地址(0~63)						设置Y地址计数器
读状态	1	0	Busy	0	On/Off	RST	0	0	0	0	状态解释: RST 1:复位，0:正常 On/Off 1:显示关闭，0:显示打开 Busy 1:内部操作，0:就绪
写显示数据	0	1	写入数据								
读显示数据	1	1	读出数据								

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.4.2 128×64图形点阵LCM和单片机的接口

- LCM 12864



南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.4.2 128×64图形点阵LCM和单片机的接口

- LCM 12864和单片机接口电路的工作过程
  - 单片机采用存储器映像方式通过扩展总线操作的方式对其进行控制；
  - 单片机的数据总线（P0.0～P0.7）和LCM的数据总线（D0~D7）直接相连，即直接入单片机的数据总线；
  - LCM的控制信号R/$\overline{W}$、D/$\overline{I}$、$\overline{CS1}$和$\overline{CS2}$直接和单片机地址总线中的A8、A9、A10和A11相连，在执行MOVX指令时，设置合适的地址将在A8~A11上产生符合LCM操作要求的控制信号时序；
  - 单片机的$\overline{RD}$、$\overline{WR}$经与非门U3A后和LCM12864的E相连，保证只有在单片机进行外部RAM读写操作时E才输出高电平，选通LCM。

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.4.2 128×64图形点阵LCM和单片机的接口

- LCM 12864内部寄存器的编址
  - 综上所述，符合LCM控制信号时序要求的编址方案如下表所示：

A_{15} ~ A_{12}	A_{11}	A_{10}	A_9	A_8	A_7 ~ A_0	操作地址	操作功能描述
	$\overline{CS2}$	$\overline{CS1}$	R/$\overline{W}$	D/$\overline{I}$			
未参与控制，0/1均可，假设全为0	1	0	0	0	未参与控制，0/1均可，假设全为0	0x0800	写LCM左半屏指令寄存器
	1	0	0	1		0x0900	写LCM左半屏显示存储器
	1	0	1	0		0x0A00	读LCM左半屏状态寄存器
	1	0	1	1		0x0B00	读LCM左半屏显示存储器
	0	1	0	0		0x0400	写LCM右半屏指令寄存器
	0	1	0	1		0x0500	写LCM右半屏显示存储器
	0	1	1	0		0x0600	读LCM右半屏状态寄存器
	0	1	1	1		0x0700	读LCM右半屏显示存储器

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.4.2 128×64图形点阵LCM和单片机的接口

128×64图形点阵LCM与单片机的接口程序示例

```
#include <reg51.h>
#include <intrins.h>

#define CWADD1  XBYTE[0x0800]
#define CRADD1  XBYTE[0x0A00]
#define DWADD1 XBYTE[0x0900]
#define DRADD1  XBYTE[0x0B00]
#define CWADD2  XBYTE[0x0400]
#define CRADD2  XBYTE[0x0600]
#define DWADD2 XBYTE[0x0500]
#define DRADD2  XBYTE[0x0700]

#define NOP        _nop_();_nop_()
```

```
unsigned char LCD_Status(unsigned char chip)
{
    unsigned char c;
    if(chip == 0)      // 如果读第一个控制寄存器
        c = CRADD1;    // 读第一个控制寄存器内容
    else
        c = CRADD2;    // 否则读第二个控制寄存
                       // 器内容
    NOP;               // 延时两个机器周期
    return c;          // 返回读出的内容
}
```


南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269