COG19264-32使用说明书

目　　录

1．概述

科飞研科技专注于液晶屏及液晶模块的研发、制造。所生产19264-32型液晶模块由于使用方便、显示清晰，广泛应用于各种人机交流面板。

19264-32可以显示192列*64行点阵单色图片，或显示12个/行*4行16*16点阵的汉字，或显示 24 个/行*8 行 8*8 点阵的英文、数字、符号。

2．19264-32图像型点阵液晶模块的特性

2.1 结构牢。

2.2IC 采用 IST3020,功能强大，稳定性好

2.3 功耗低:2-200mW（不带背光<2mW,带背光<200mW）；

2.4 显示内容:

●192*64 点阵单色图片;

●可选用 16*16 点阵或其他点阵的图片来自编汉字，按照 16*16 点阵汉字来计算可显示 12 字/行*4 行。按照 12*12 点阵汉字来计算可显示 16 字/行*4 行。

2.5 指令功能强:可软件调对比度、正显/反显转换、行列扫描方向可改（可旋转 180 度使用）。并口时：可以“读-改-写”；

2.6 接口简单方便:可采用 4 线 SPI 串行接口，或选择并行接口（6800 时序和 8080 时序可选）。

2.7 工作温度宽:-20℃ - 70℃;

3. 外形尺寸及接口引脚功能

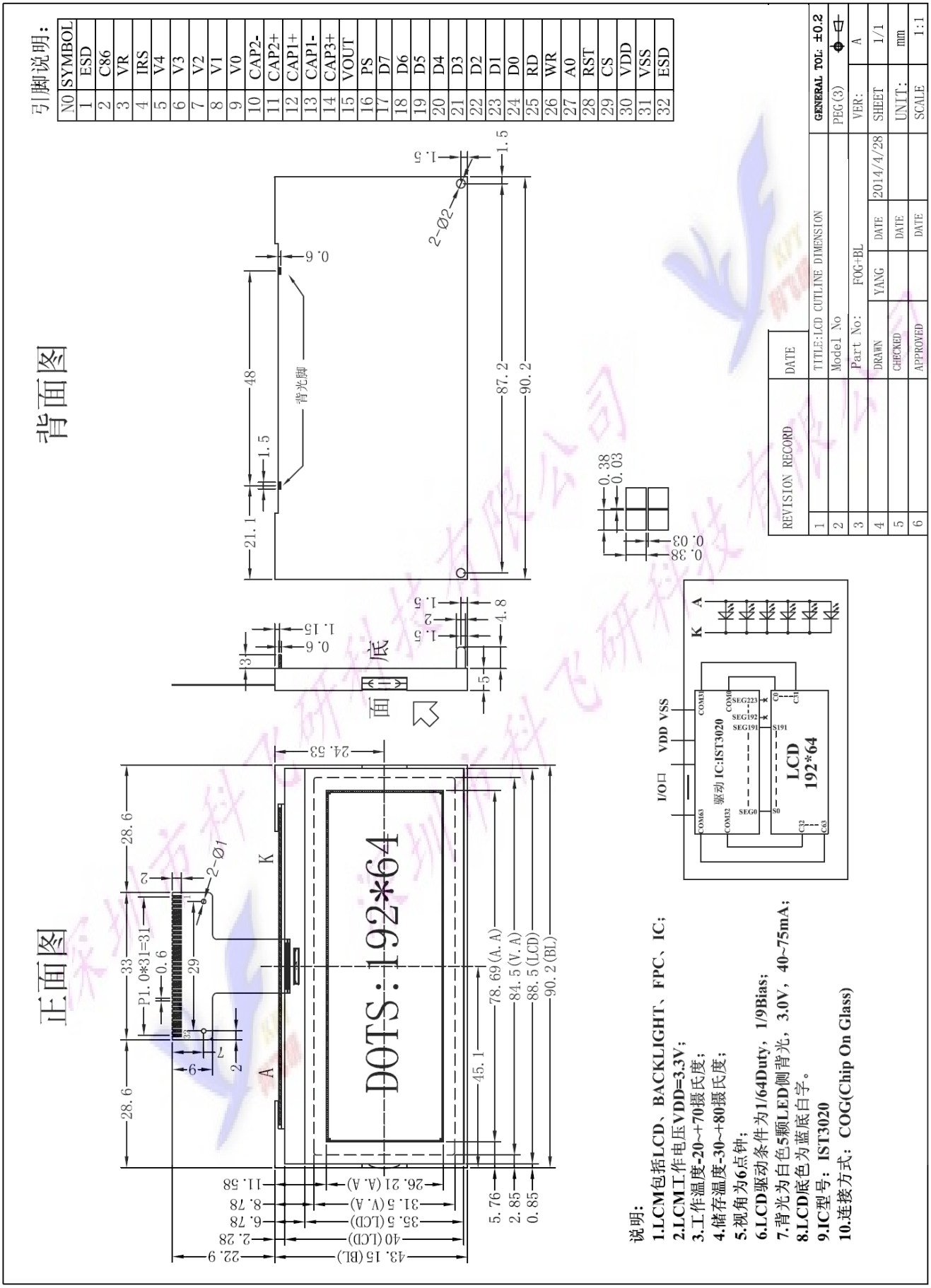
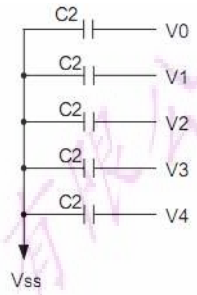
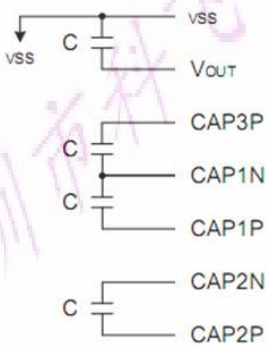


图 1. 液晶模块外形尺寸

模块的接口引脚功能

引 线 号	符 号	名 称	功 能
1	ESD	空脚	空脚
2	C86	选择 6800 或 8080	**并行接口时：H:6800 系统,L:8080 系统。** **串行接口时：接 VDD**
3	VR	输出电压调整	输出电压调整，通过外部电阻来调整电压。调整电阻值会改变 V0 输出电压，从而改变液晶屏对比度。详细方法见“7.指令功能及硬件接口与编程案例”。 **当 IRS 脚接 VDD 时，此引脚“VR”失效。**
4	IRS	内/外电阻	**接 VSS:使用 IC 外部电阻，此时“VR”引脚起作用。** **接 VDD:使用 IC 内部电阻进行调对比度，此时“VR”引脚失效**
5	V4	偏置电压	LCD 驱动偏置电压。各与 VSS 之间接电容。  **电压关系：VOUT>V0>V1>V2>V3>V4>VSS。**
6	V3	偏置电压	
7	V2	偏置电压	
8	V1	偏置电压	
9	V0	偏置电压	
10	CAP2-	倍压电路	外接升压电容，如下图： 
11	CAP2+	倍压电路	
12	CAP1+	倍压电路	
13	CAP1-	倍压电路	
14	CAP3+	倍压电路	
15	VOUT	LCD 倍压输出	
16	PS	选串并控制接口	**接 VDD:选择并行接口， 接 VSS:选择串行接口**
17	D7(SDA)	I/O	并行接口时：数据总线 DB7 串行接口时：串行数据(SDA)
18	D6(SCK)	I/O	并行接口时：数据总线 DB6 串行接口时：串行时钟(SCLK)
19-24	D5-D0	I/O	数据总线 DB0-DB5 串行接口时：空脚
25	E(RD)	**6800 时序:使能** 8080 时序：读	**并行接口时并且选择 6800 时序时：使能信号,高电平有效.** 并行接口时并且选择 8080 时序时：读数据,低电平有效. 串行接口时：接 VDD 或悬空
26	R/W(WR)	**6800 时序:读/写** 8080 时序：写	**并行接口时并且选择 6800 时序时：H:读数据 L:写数据** 并行接口时并且选择 8080 时序时：写数据,低电平有效. 串行接口时：接 VDD 或悬空
27	RS(A0)	寄存器选择信号	H:数据寄存器 0:指令寄存器
28	RST	复位	低电平复位，复位完成后，回到高电平，液晶模块开始工作

29	CS	片选	低电平片选
30	VDD	电路电源	3.3V
31	VSS	接地	0V
32	ESD	空脚	空脚

表 1：模块的接口引脚功能

4.基本原理

4.1 液晶屏（LCD）

在 LCD 上排列着 192×64 点阵,192 个列信号与驱动 IC 相连，64 个行信号也与驱动 IC 相连，IC 邦定在 LCD 玻璃上（这种加工工艺叫 COG）.

4.2 电路内部框图。

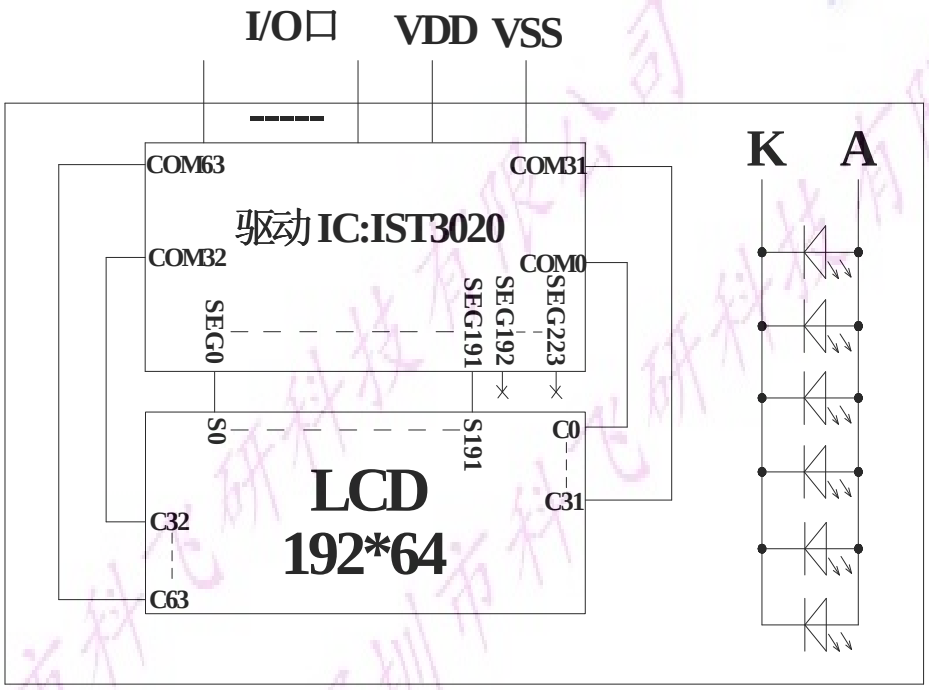


图 2: 图像点阵型液晶模 块的电路框图

4.3 背光参数

该型号液晶模块带 LED 背光源。它的性能参数如下:
工作温度:-20∽+70° C;
存储温度:-30∽+80° C;
背光板白色。
正常工作电流为：(8∽15)*5=40~75mA（LED 灯数共 5 颗);
工作电压：3.0V;

5.技术参数

5.1 最大极限参数（超过极限参数则会损坏液晶模块）

名称	符号	标准值			单位
		最小	典型	最大	
电路电源	VDD - VSS	-0.3		7.0	V
LCD 驱动电压	V0、VOUT	-0.3		13.5V	V
LCD 驱动电压	V1\V2\V3\V4	-0.3		V0	
静电电压		-	-	100	V
工作温度		-20		+70	℃
储存温度		-30		+80	℃

表 2：最大极限参数

5.2 直流（DC）参数

名 称	符 号	测 试 条 件	标 准 值			单位
			MIN	TYPE	MAX	
工作电压（当 3.3V 供电时）	VDD		2.4	3.3	3.6	V
工作电压（当 5.0V 供电时）			4.8	5.0	5.2	V
背光工作电压	VLED		2.9	3.0	3.1	V
输入高电平	VIH	-	2.2		VDD	V
输入低电平	VIO	-	-0.3		0.6	V
输出高电平	VOH	IOH = 0.2mA	2.4		-	V
输出低电平	VOO	IOO = 1.2mA	-		0.4	V
模块工作电流	IDD	VDD = 3.3V	-		0.3	mA
背光工作电流	ILED	VLED=3.0V	40	60	100	mA

表 3：直流（DC）参数

6. 读写时序特性

6.1 串行接口：

从 CPU 写到 IST3020（Writing Data from CPU to IST3020）

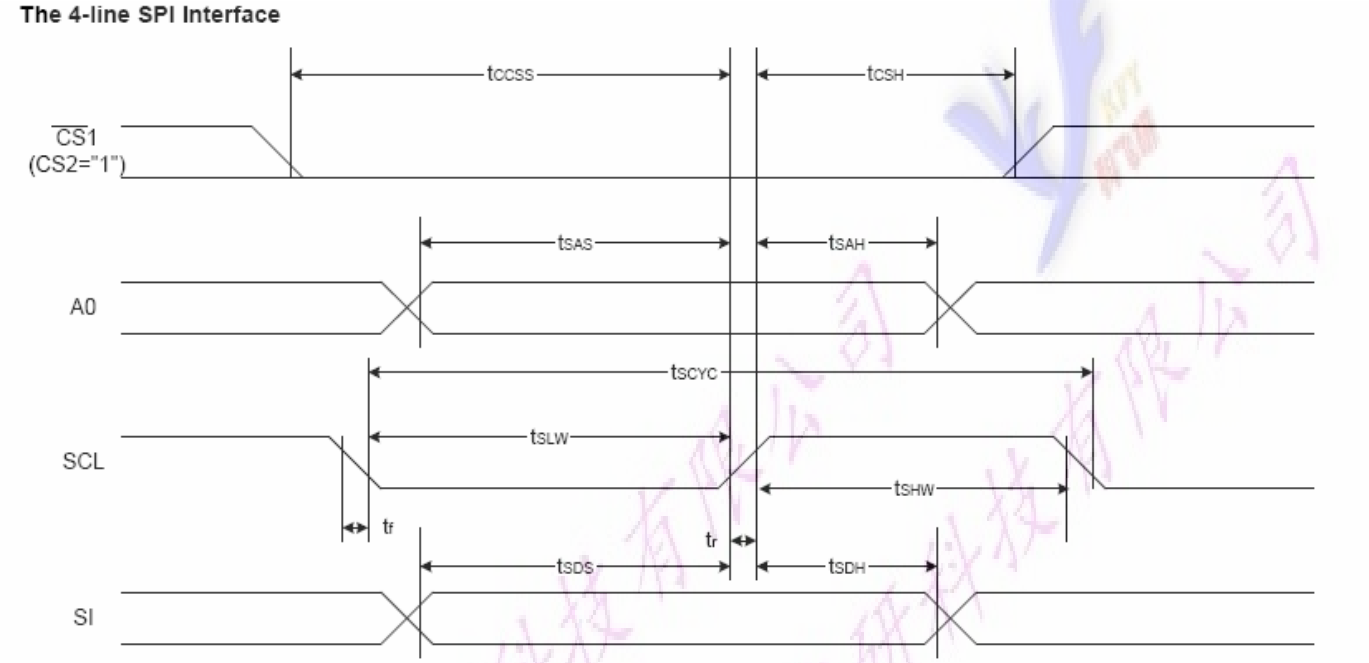


图 4. 从 CPU 写到 IST3020（Writing Data from CPU to IST3020）

6.2 串行接口：时序要求（AC 参数）：

写数据到 IST3020 的时序要求： 表 4.

项 目	符 号	测试条件	极限值			单位
			MIN	TYPE	MAX	
4线 SPI串口时钟周期 （4-line SPI Clock Period）	Tscyc	引脚：SCK	250	--	50	ns
保持SCK高电平脉宽 （SCK “H” pulse width）	Tshw	引脚：SCK	100			ns
保持SCK低电平脉宽 （SCK “L” pulse width）	TSLW	引脚：SCK	100			ns
地址建立时间 (Address setup time)	TSAS	引脚： RS	150	--	--	ns
地址保持时间 (Address hold time)	Tsah	引脚：RS	150	--	--	ns
数据建立时间 (Data setup time)	Tsds	引脚：SID	100	--	--	ns
数据保持时间 (Data hold time)	TSDH	引脚：SID	100	--	--	ns
片选信号建立时间 （CS-SCL time）	Tcss	引脚：CS	150			ns

片选信号保持时间 （CS-SCL time）	T_{csh}	引脚：CS	150			ns

VDD =3.0V±5%, Ta = 25℃

6.3 并行接口：

从 CPU 写到 IST3020（Writing Data from CPU to IST3020）

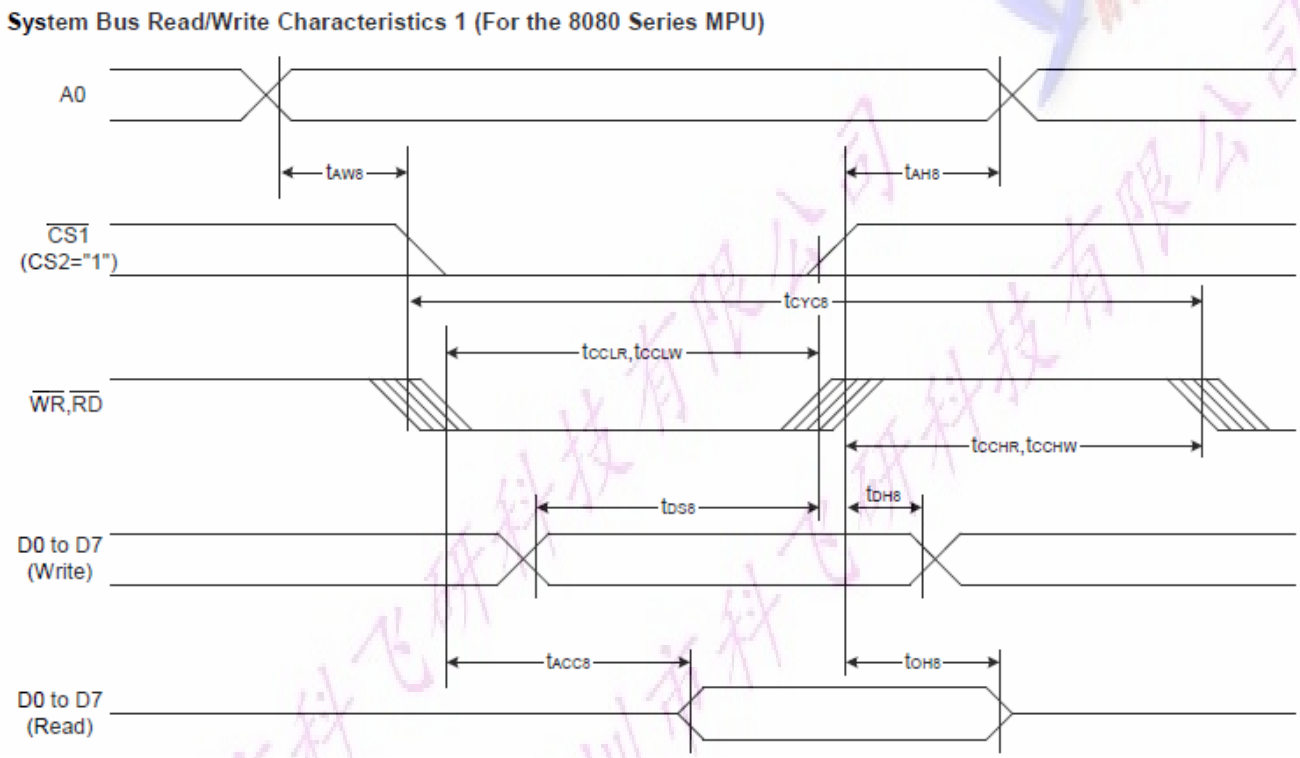


图 5. 从 CPU 写到 IST3020（Writing Data from CPU to IST3020）

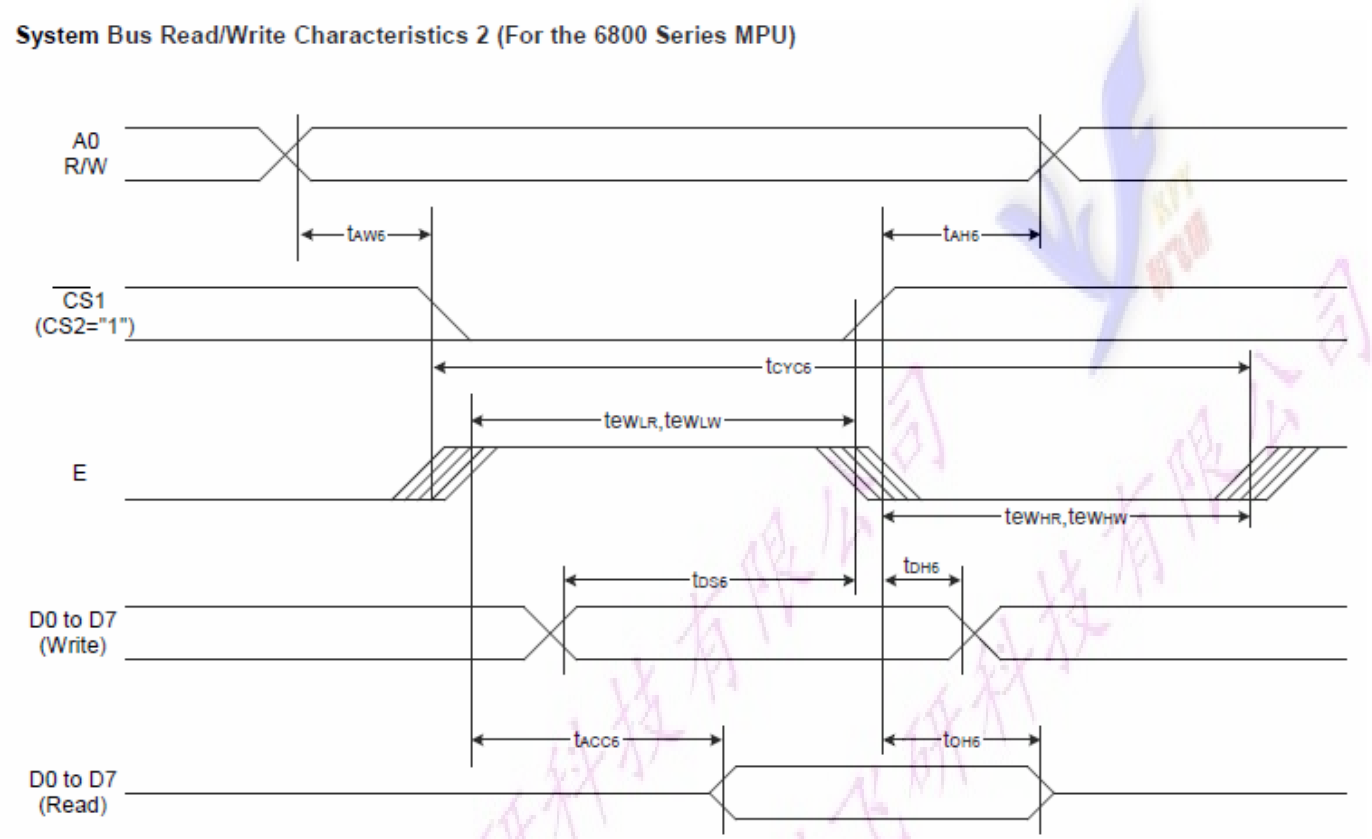


图 6. 从 CPU 写到 IST3020（Writing Data from CPU to IST3020）

6.4 并行接口：时序要求（AC 参数）：

写数据到 IST3020 的时序要求：（8080 系列 MPU）

项 目	符 号	测试条件	极限值			单位
			MIN	TYPE	MAX	
地址保持时间	A0	tAH8	0	--	--	ns
地址建立时间		tAW8	0		--	ns
系统循环时间		tCYC8	300		--	ns
使能“低”脉冲（写）	WR	tCCLW	60	--	--	ns
使能“高”脉冲（写）		tCCHW	60	--	--	ns
使能“低”脉冲（读）	RD	tCCLR	60	--	--	ns
使能“高”脉冲（读）		tCCHR	60	--		ns
写数据建立时间	D0-D7	tDS8	40		--	ns
写数据保持时间		tDH8	15		--	
读时间		tACC8	--		140	
读输出来允许时间		tOH8	10		100	ns

写数据到 IST3020 的时序要求：（6800 系列 MPU）

项 目	符 号	测试条件	极限值			单位
			MIN	TYPE	MAX	
地址保持时间	A0	tAH6	0	--	--	ns
地址建立时间		tAW6	0		--	ns
系统循环时间		tCYC6	300		--	ns
使能“低”脉冲（写）	WR	tEWLW	60	--	--	ns
使能“高”脉冲（写）		tEWHW	60	--	--	ns
使能“低”脉冲（读）	RD	tEWLR	60	--	--	ns
使能“高”脉冲（读）		tEWHR	60	--		ns
写数据建立时间	D0-D7	tDS6	40		--	ns
写数据保持时间		tDH6	15		--	
读时间		tACC6	--			
读输出来允许时间		tOH6	60			ns

6.5 电源启动后复位的时序要求（RESET CONDITION AFTER POWER UP）：

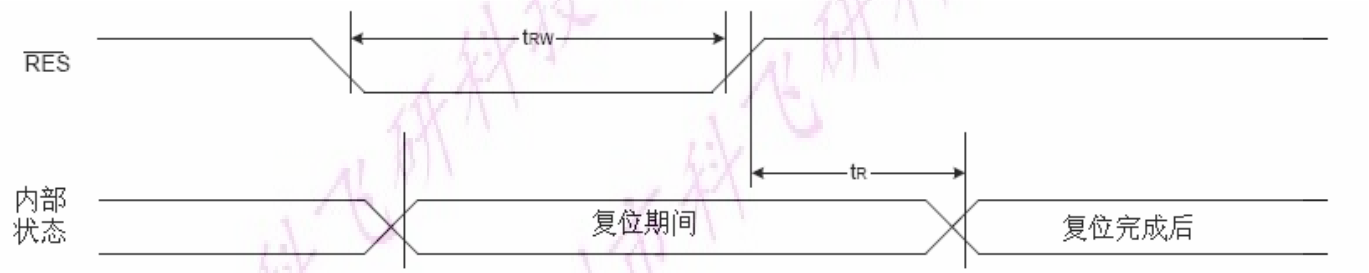


图 7：**电源启动后复位的时序**

表 6：电源启动后复位的时序要求

项 目	符 号	测试条件	极限值			单位
			MIN	TYPE	MAX	
复位时间	tR		--	--	1.0	us
复位保持低电平的时间	tRW	引脚：RES	1.0	--	--	us

7．指令功能：

7.1 指令表

指 令 表 表 8.

指令名称	指令码									说明
	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
(1)显示开/关（display on/off）	0	1	0	1	0	1	1	1	0 1	显示开/关: 0XAE:关，0XAF：开
(2)显示初始行设置（Display start line set）	0	0	1	显示初始行地址，共 6 位						设置显示存储器的显示初始行,可设置值为 0X40~0X7F,分别代表第 0～63 行，针对该液晶屏一般设置为 0x60
(3)页地址设置 (Page address set)	0	1	0	1	1	显示页地址，共 4 位				设置页地址。每 8 行为一个页，64 行分为 8 个页，可设置值为：0XB0～0XB8 分别对应第一页到第九页，第九页是一个单独的一行图标，本液晶屏没有这一行图标，所以设置值为 0XB0～0XB7 分别对应第一页～第八页。
(4) 列地址高4位设置	0	0	0	0	1	列地址的高 4 位				高 4 位与低 4 位共同组成列地址，指定 128 列中的其中一列。比如液晶模块的第 100 列地址十六进制为 0x64，那么此指令由 2 个字节来表达：0x16, 0x04
(4) 列地址低4位设置	0	0	0	0	0	列地址的低 4 位				
(5) 读状态 (Status read)	0	状态				0	0	0	0	并口时：读驱动 IC 的当前状态,串口时不能用此指令
(6)写显示数据到液晶屏 (Display data write)	1	8 位显示数据								从 CPU 写数据到液晶屏，每一位对应一个点阵，1 个字节对应 8 个竖置的点阵
(7)读液晶屏的显示数据 (Display data read)	1	8 位显示数据								并口时：读已经显示到液晶屏上的点阵数据。串口时不能用此指令
(8) 显示列地址增减 (ADC select)		1	0	1	0	0	0	0	0 1	显示列地址增减： 0xA0：常规：列地址从左到右， 0xA1：反转：列地址从右到左
(9)显示正显/反显 (Display normal/reverse)	0	1	0	1	0	0	1	1	0 1	显示正显/反显: 0xA6：常规：正显 0xA7：反显
(10)显示全部点阵 (Display all points)	0	1	0	1	0	0	1	0	0 1	显示全部点阵: 0xA4：常规 0xA5：显示全部点阵
(11)LCD 偏压比设置 (LCD bias set)	0	1	0	1	0	0	0	1	0 1	设置偏压比： 0XA2：BIAS=1/9 (常用) 0XA3：BIAS=1/7
(12) 读-改-写 (Read-modify-write)	0	1	1	1	0	0	0	0	0	0XE0: “读-改-写” 开始。 列地址的增加： 写入时：列地址+1 读出时：列地址不加 详情请参考IC资料第43-44页
(13) 退出上述“读-改-写”指令(End)	0	1	1	1	0	1	1	1	0	0XEE :上述“读-改-写”指令结束 详情请参考 IC 资料第 43-44 页
(14) 软件复位（Reset）	0	1	1	1	0	0	0	1	0	0XE2 :软件复位。

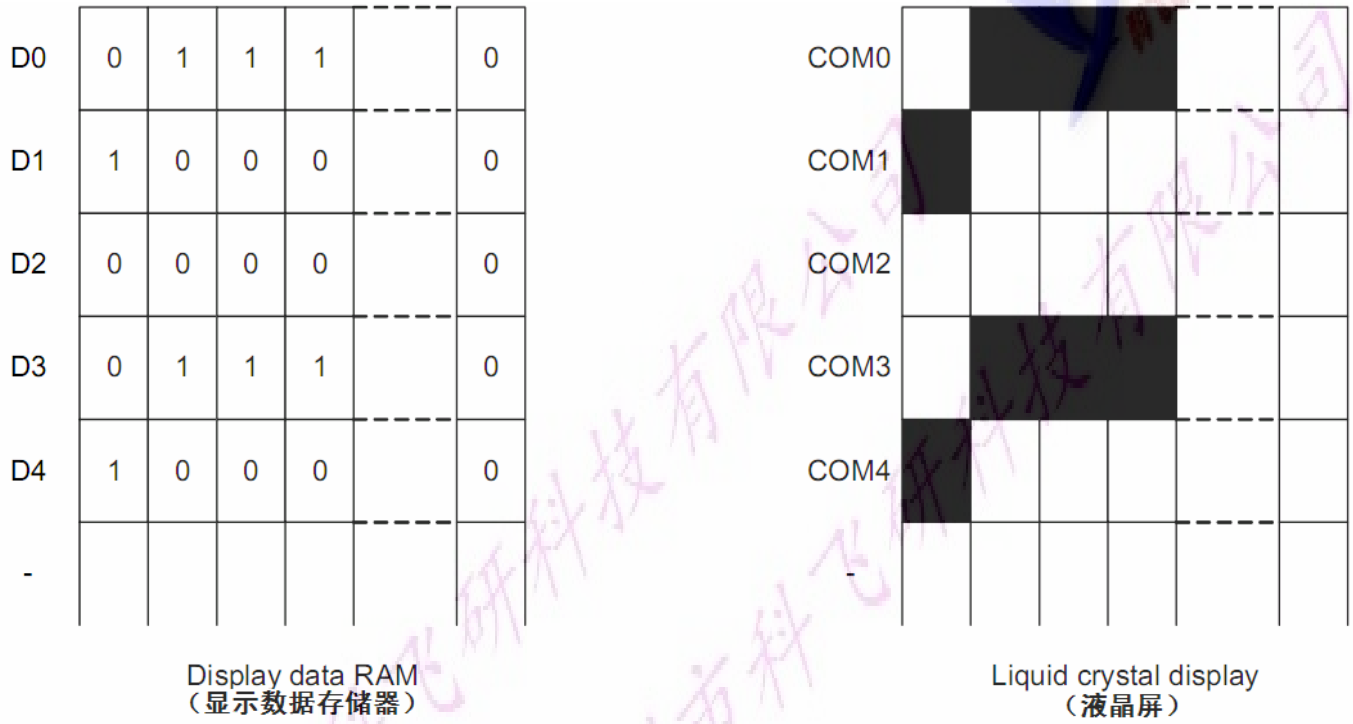
(15) 行扫描顺序选择 (Common output mode select)			1	1	0	0	0 1	0	0	**0**	**行扫描顺序选择：** **0XC0:普通扫描顺序：从上到下** **0XC8:反转扫描顺序：从下到上**
(16) 电源控制 (Power control set)			0	0	1	0	1	**电压操作模式选择，共 3 位**			选择内部电压供应操作模式： D2、D1、D0 位分别对应内部升压是否打开(1 为打开，0 为不打开)，电压调整电路是否打开(1 为打开，0 为不打开)，电压跟随器是否打开(1 为打开，0 为不打开)。 通常是 **0x2C,0x2E,0x2F** 三条指令按顺序紧接着写，表示依次打开内部升压、电压调整电路、电压跟随器。也可以单单写 **0x2F**，一次性打开三部分电路。
(17) 选择内部电阻比例		0	0	0	1	0	0	**内部电压值电阻设置**			选择内部电阻比例（Rb/Ra）:可以理解为**粗调**对比度值。可设置范围为：**0x20～0x27，**数值越大对比度越浓，越小越淡
(18)	内部设置液晶电压模式	0	1	0	0	0	0	0	0	1	设置内部电阻微调，可以理解为**微调**对比度值，此两个指令需紧接着使用。上面一条指令 0x81 是不改的，下面一条指令可设置范围为：**0x00～0x3F,**数值越大对比度越浓，越小越淡
	设置的电压值		0	0	**6 位电压值数据，0～63 共 64 级**						
(19)静态图标显示：开/关		0	1	0	1	0	1	1	0	**0** **1**	静态图标的开关设置： **0xAC**: 关, **0xAD**: 开。 此指令在进入及退出睡眠模式时起作用
(20) 升压倍数选择 (Booster ratio set)		0	1	1	1	1	1	0	0	0	选择升压倍数： 00：2 倍，3 倍，4 倍 01：5 倍 11：6 倍。本模块外部已设置升压倍数为 4 倍，不必使用此指令
			0	0	0	0	0	0	2 位数设置升压倍数		
(21) 省电模式（Power save）											省电模式，此非一条指令，是由“(10)显示全部点阵”、 (19)静态图标显示：开/关等指令合成一个“省电功能”。详细看 IC 规格书第 47 页“POWER SAVE”
(22)空指令（ NOP）		0	1	1	1	0	0	0	1	1	空操作
(23) 测试（Test）		0	1	1	1	1	*	*	*	*	内部测试用，千万别用！

请详细参考 IC 资料”IST3020.PDF”的第 32～46 页。

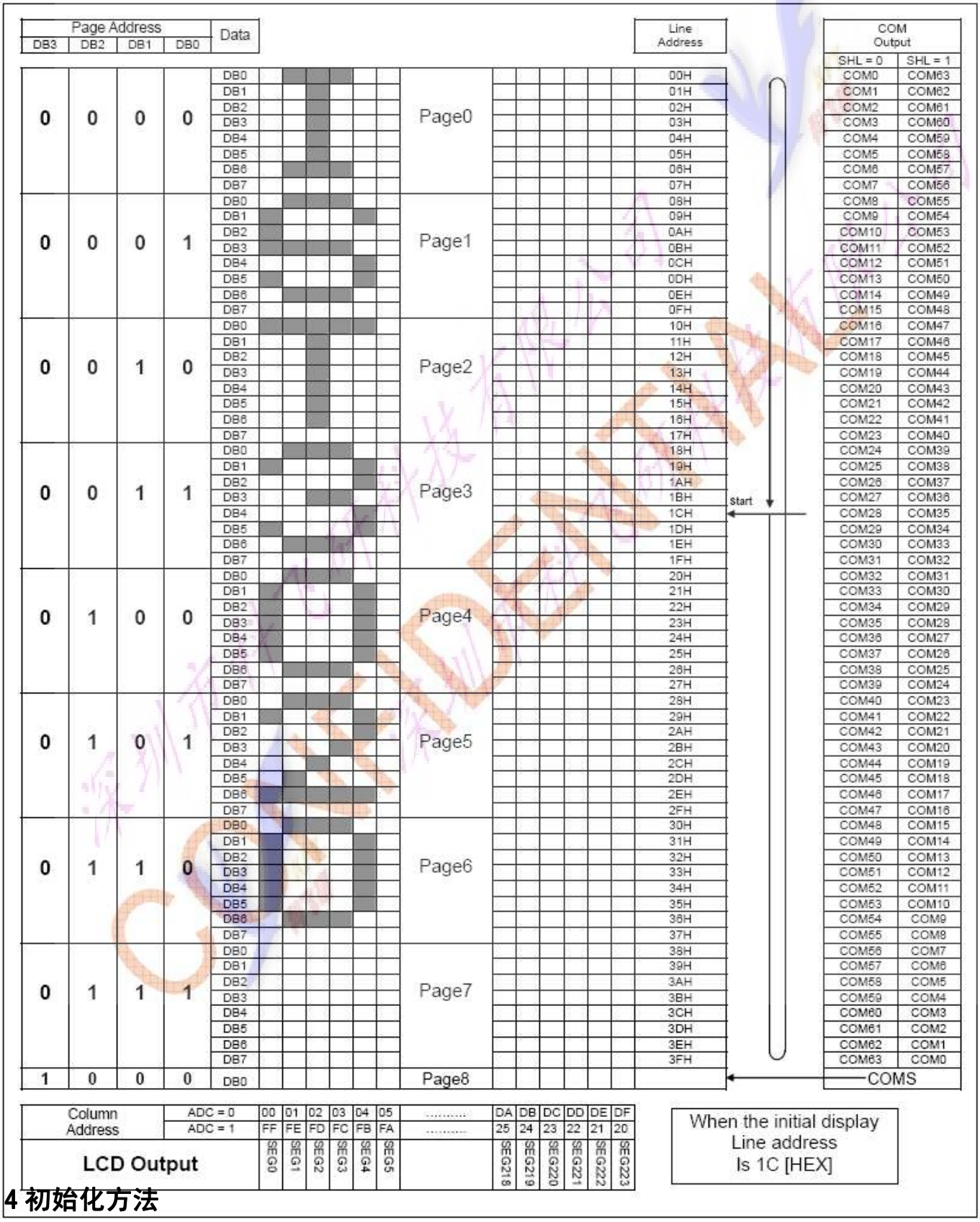
7.3 点阵与 DD RAM 地址的对应关系

请留意页的定义：PAGE,与平时所讲的“页”并不是一个意思，在此表示 8 个行就是一个“页”，一个 128*64 点阵的屏分为 8 个“页”，从第 0“页”到第 7“页”。

DB7--DB0 的排列方向：数据是从下向上排列的。最低位 D0 是在最上面，最高位 D7 是在最下面。每一位（bit）数据对应一个点阵，通常“1”代表点亮该点阵，“0”代表关掉该点阵。如下图所示：

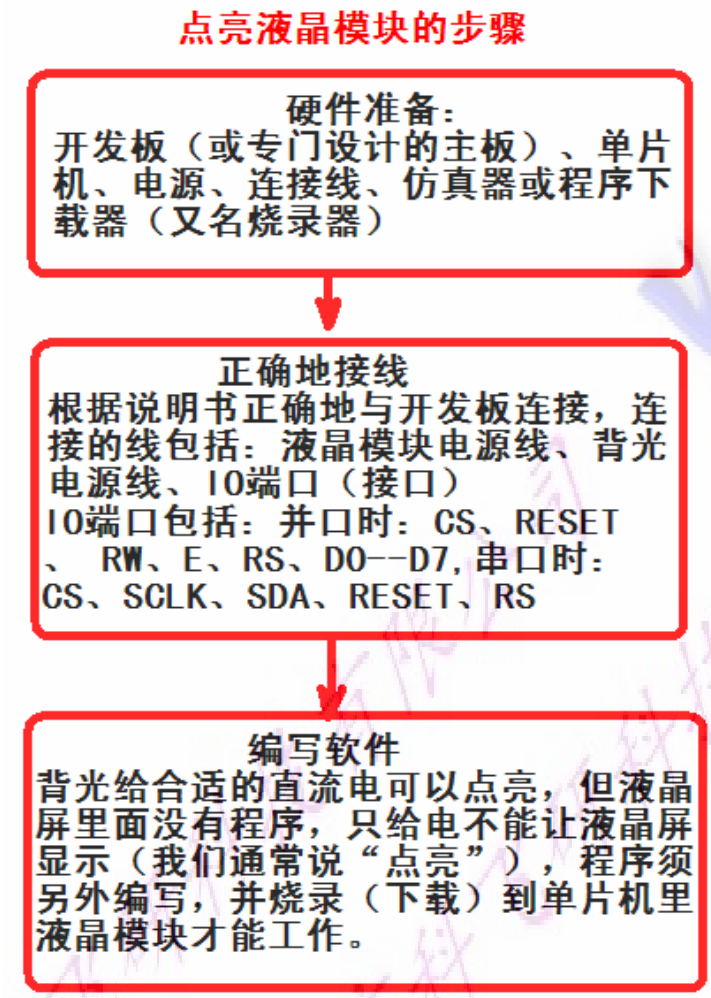


下图摘自 IST3020 IC 资料，可通过“IST3020.PDF”之第 18 页获取最佳效果。



7.4 初始化方法

用户所编的显示程序, 开始必须进行初始化, 否则模块无法正常显示, 过程请参考程序



7.5 程序举例:

液晶模块与 MPU(以 8051 系列单片机为例)接口图如下:

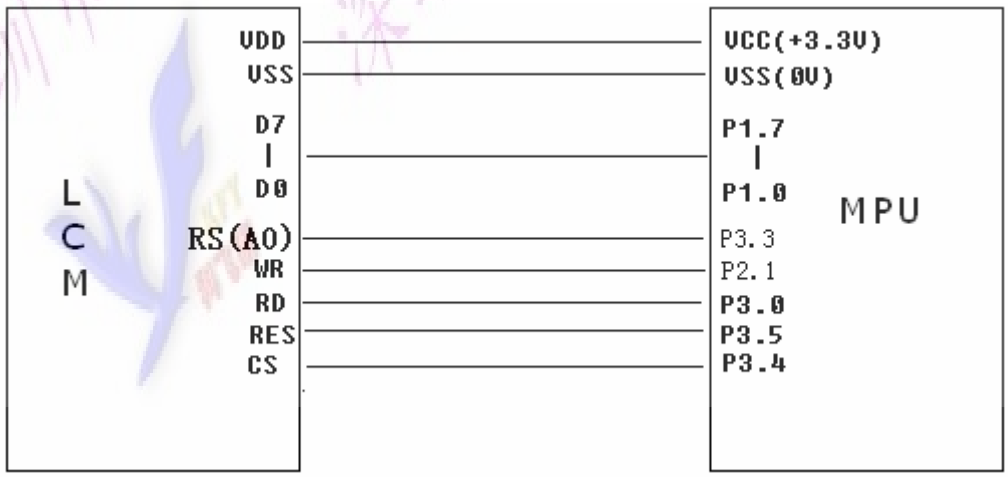
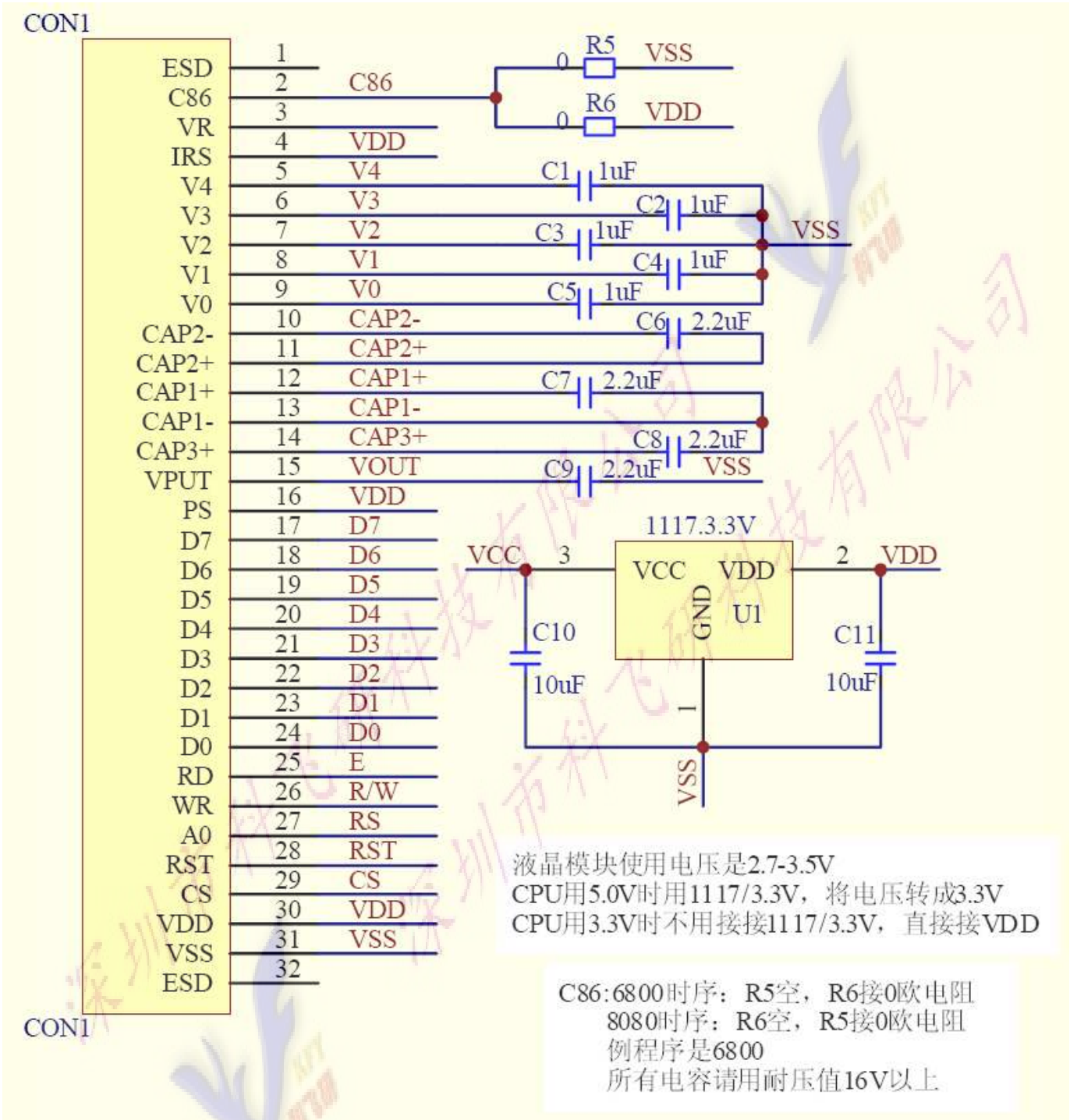


图 8. 并行接口

并行电路图


CON1
ESD 1
C86 2
VR 3
IRS 4
V4 5
V3 6
V2 7
V1 8
V0 9
CAP2- 10
CAP2+ 11
CAP1+ 12
CAP1- 13
CAP3+ 14
VPUT 15
PS 16
D7 17
D6 18
D5 19
D4 20
D3 21
D2 22
D1 23
D0 24
RD 25
WR 26
A0 27
RST 28
CS 29
VDD 30
VSS 31
ESD 32
CON1
C86
VDD
V4
V3
V2
V1
V0
CAP2-
CAP2+
CAP1+
CAP1-
CAP3+
VOUT
VDD
D7
D6
D5
D4
D3
D2
D1
D0
E
R/W
RS
RST
CS
VDD
VSS
R5 0 VSS
R6 0 VDD
C1 1uF
C2 1uF
C3 1uF
C4 1uF
C5 1uF
C6 2.2uF
C7 2.2uF
C8 2.2uF
C9 2.2uF
VSS
1117.3.3V
VCC 3
VCC VDD
GND U1
2 VDD
C10 10uF
C11 10uF
1
VSS
液晶模块使用电压是2.7-3.5V
CPU用5.0V时用1117/3.3V，将电压转成3.3V
CPU用3.3V时不用接接1117/3.3V，直接接VDD
C86:6800时序：R5空，R6接0欧电阻
8080时序：R6空，R5接0欧电阻
例程序是6800
所有电容请用耐压值16V以上

7.51 程序：

以下为并行方式的范例程序：

```
/* Test program for 19264-32 ，并行接口
   驱动 IC 是:IST3020(or compatible)

*/
#include <reg51.H>
#include <intrins.h>
#include <Ctype.h>

sbit cs1=P3^4;     /*接口定义*/
sbit reset=P3^5;   /*接口定义*/
sbit rs=P3^3;      /*接口定义*/
sbit rd=P3^0;      /*接口定义*/
sbit wr=P2^1;      /*接口定义。另外 P1.0~1.7 对应 DB0~DB7*/
sbit key=P2^0;     /*按键接口，P2.0 口与 GND 之间接一个按键*/
```

```
#define uchar unsigned char
#define uint unsigned int
#define ulong unsigned long
uchar code bmp_19264[];

uchar code zhuang1[];
uchar code tai1[];
uchar code shi1[];
uchar code yong1[];

void delay(int i);
void delay_us(int i) ;

//=============transfer command to LCM==============
void transfer_command(int data1)
{
     cs1=0;
     rs=0;
     wr=0;
     rd=0;
    delay_us(2);
     P1=data1;
     rd=1;
    delay_us(2);
     cs1=1;
     rd=0;
}

//-----------transfer data to LCM---------------
void transfer_data(int data1)
{
     cs1=0;
     rs=1;
     wr=0;
     rd=0;
     P1=data1;
     rd=1;
     cs1=1;
     rd=0;
}

/*延时*/
void delay(int i)
{
     int j,k;
     for(j=0;j<i;j++)
     {
          for(k=0;k<109;k++);   //1ms
     }
}

/*延时*/
void delay_us(int i)
{
     int j,k;
     for(j=0;j<i;j++)
     {
          for(k=0;k<1;k++);
     }
```

```
}

void waitkey()
{
repeat:  if(key==1)
              goto repeat;
              else
              delay(1000);
}

/*LCD 模块初始化*/
void initial_lcd()
{
     rd=0;
     reset=0;           /*低电平复位*/
     delay(20);
     reset=1;          /*复位完毕*/
     delay(10);
     transfer_command(0xAB);//开启内部晶振电路
     delay(10);
     transfer_command(0x2C);//升压步骤 1
     delay(20);
     transfer_command(0x2E);//升压步骤 2
     delay(20);
     transfer_command(0x2F);//升压步骤 3
     delay(20);
     transfer_command(0xA2); //BIAS 设置
     transfer_command(0x24); //粗调对比度
     transfer_command(0x81); //微调对比度
     transfer_command(0x14); //微调对比度的值：从 0x00 到 0x3f
     transfer_command(0x40);
     transfer_command(0xA1); //列扫描顺序：从左到右
     transfer_command(0xC0); //行扫描顺序：从左到右
     transfer_command(0xAF);
}


void lcd_address(uchar page,uchar column)
{
     column=column+31;                                    //
     page=page-1;
     transfer_command(0xb0+page);                  //设置页地址。每页是 8 行。一个画面的 64 行被分成 8 个页。我们平常
                                                          所说的第 1 页，在 LCD 驱动 IC 里是第 0 页，所以在这里减去 1*/
     transfer_command(((column>>4)&0x0f)+0x10);    //设置列地址的高 4 位
     transfer_command(column&0x0f);                //设置列地址的低 4 位
}

/*全屏清屏*/
void clear_screen()
{
     unsigned char i,j;
     for(i=0;i<9;i++)
     {
          lcd_address(1+i,1);
          for(j=0;j<224;j++)
          {
               transfer_data(0x00);
          }
     }
}
```

```
//===显示测试画面：例如全显示，隔行显示，隔列显示，雪花显示=====
void test_display(uchar data1,uchar data2)
{
    int i,j;
    for(j=0;j<8;j++)
    {
        cs1=0;
        lcd_address(j+1,1);
        for(i=0;i<192;i++)
        {
            transfer_data(data1);
            transfer_data(data2);
        }
    }
}

void display_graphic_16x192(uchar page,uchar column,uchar data1)
{
    int i,j;
    for(j=0;j<2;j++)
    {
        lcd_address(page+j,column);
        for(i=0;i<192;i++)
        {
            transfer_data(data1);
        }
    }
}

void display_graphic_16x16(uchar reverse,uchar page,uchar column,uchar *dp)
{
    int i,j;
    for(j=0;j<2;j++)
    {
        lcd_address(page+j,column);
        for(i=0;i<16;i++)
        {
            if(reverse==1)
                transfer_data(*dp);
            else
                transfer_data(~*dp);
            dp++;
        }
    }
}

void display_graphic_19264(uchar page,uchar column,uchar *dp)
{
    int i,j;
    for(j=0;j<8;j++)
    {
        lcd_address(page+j,column);
        for(i=0;i<192;i++)
        {
            transfer_data(*dp);
            dp++;
        }
    }
```

```
}

void main(void)
 {
    while(1)
    {
        initial_lcd();
        clear_screen();                                    //clear all dots
        display_graphic_19264(1,1,bmp_19264);
        waitkey();
        clear_screen();
        display_graphic_16x192(1,1,0xff);
        display_graphic_16x16(0,1,(16*3),zhuang1);
        display_graphic_16x16(0,1,(16*4),tai1);
        waitkey();
        test_display(0xff,0xff);                          //全显示
        waitkey();
        test_display(0xff,0x00);                          //横线 1
        waitkey();
        test_display(0x00,0xff);                          //横线 2
        waitkey();
        test_display(0xaa,0xaa);                          //竖线 1
        waitkey();
        test_display(0x55,0x55);                          //竖线 2
        waitkey();
    }
}


uchar code zhuang1[]={
/*--  文字:  状  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=16x16   --*/
0x08,0x30,0x00,0xFF,0x20,0x20,0x20,0x20,0xFF,0x20,0xE1,0x26,0x2C,0x20,0x20,0x00,
0x04,0x02,0x01,0xFF,0x40,0x20,0x18,0x07,0x00,0x00,0x03,0x0C,0x30,0x60,0x20,0x00};

uchar code tai1[]={
/*--  文字:  态  --*/
/*--  宋体 12;  此字体下对应的点阵为：宽 x 高=16x16   --*/
0x00,0x04,0x04,0x04,0x84,0x44,0x34,0x4F,0x94,0x24,0x44,0x84,0x84,0x04,0x00,0x00,
0x00,0x60,0x39,0x01,0x00,0x3C,0x40,0x42,0x4C,0x40,0x40,0x70,0x04,0x09,0x31,0x00};

uchar code bmp_19264[]={
/*--  调入了一幅图像：      --*/
/*--  宽度 x 高度=192x64  --*/
                        };
```

串行接口

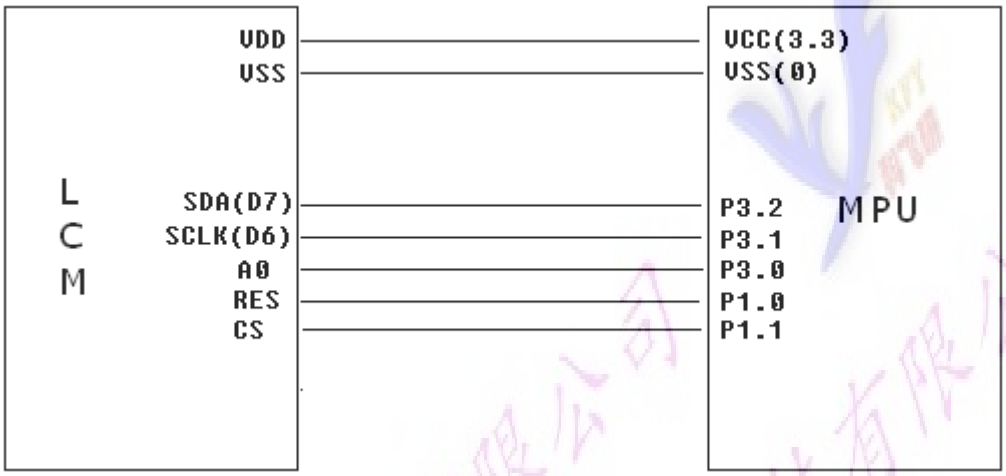
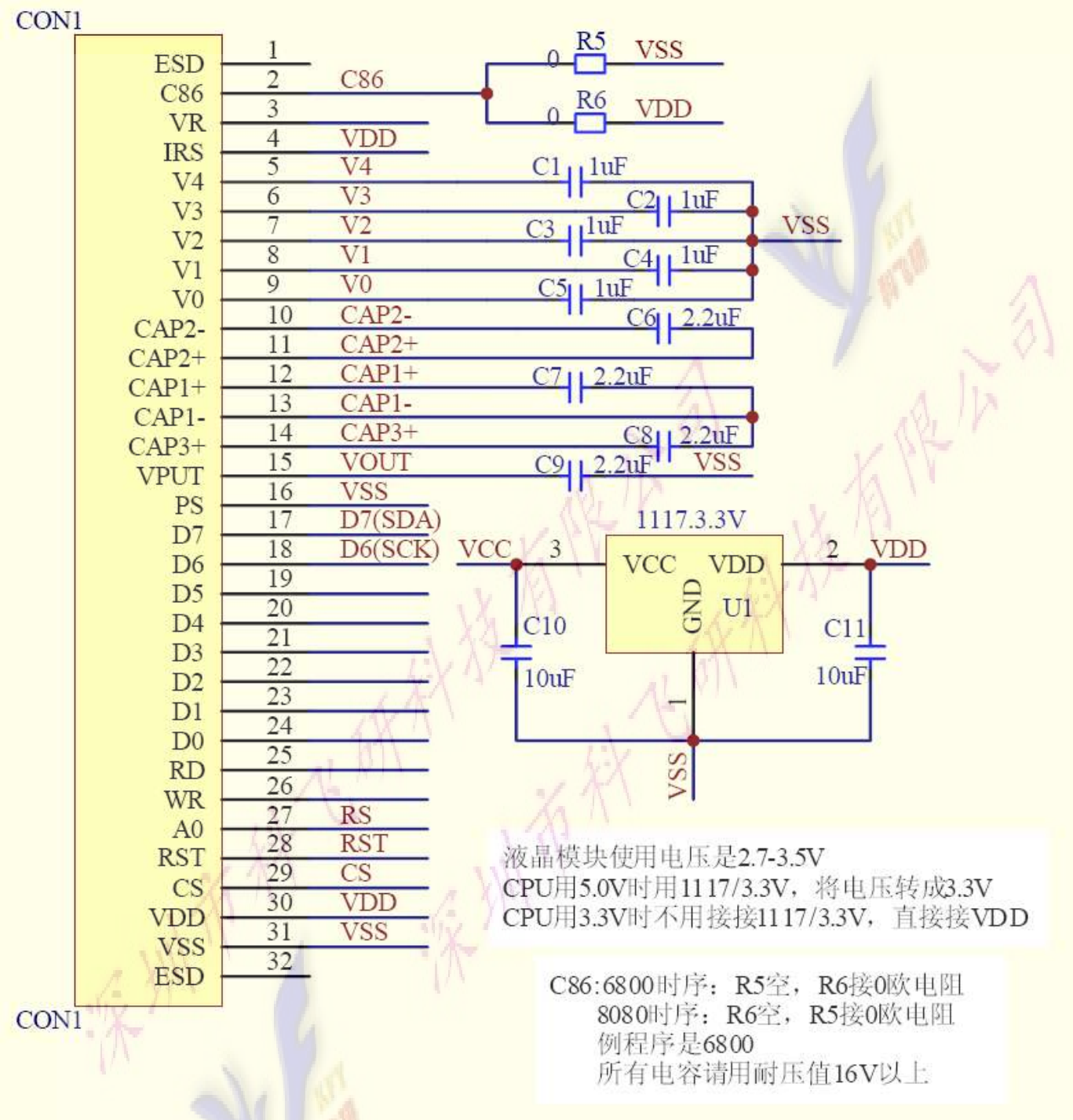


图 9. 串行接口

与并行方式相比较，只需改变接口顺序以及传送数据、传送命令这两个函数即可：

串行电路图



串行程序：

```
#include <reg51.H>

sbit lcd_rs=P3^3;     /*接口定义:lcd_rs 就是 LCD 的 rs*/
sbit lcd_sclk=P1^6;   /*接口定义:lcd_sclk 就是 LCD 的 sclk*/
sbit lcd_sid=P1^7;    /*接口定义:lcd_sid 就是 LCD 的 sid*/
sbit reset=P3^5;      /*接口定义:lcd_reset 就是 LCD 的 reset*/
sbit cs1=P3^4;    /*接口定义:lcd_cs1 就是 LCD 的 cs1*/
void transfer_command(int data1)
{
    char i;
    lcd_rs=0;
```

```
     for(i=0;i<8;i++)
   {
         lcd_sclk=0;
         if(data1&0x80) lcd_sid=1;
         else lcd_sid=0;
         lcd_sclk=1;
         data1=data1<<=1;
   }
}

/*写数据到 LCD 模块*/
void transfer_data(int data1)
{
     char i;
     lcd_rs=1;
     for(i=0;i<8;i++)
   {
         lcd_sclk=0;
         if(data1&0x80) lcd_sid=1;
         else lcd_sid=0;
         lcd_sclk=1;
         data1=data1<<=1;
    }
}
```