

案例3

基于单片机控制液晶屏的设计和实现

3.1 设计任务与要求

1. 任务

由单片机控制点阵式液晶显示模块显示汉字,完成电路设计及实物制作。

2. 要求

熟悉点阵式液晶显示模块原理,设计硬件电路原理图以及 PCB 板图,软件使用汇编语言实现。

3.2 总体设计

本设计主要阐述点阵式液晶显示模块 WGM12232 与单片机 80C51 的接口及编程方法,用于单片机控制液晶屏显示汉字。其硬件设计原理图如图 3-1 所示。

设计中用外接石英晶体振荡器和微调电容与 80C51 片内的振荡器定时回路连接,石英晶振起振后,片内的振荡器电路将按相同频率自激振荡,再加上由 USB 口提供的+5V 的电压供电就可以使单片机工作了。80C51 单片机通过自身的并行接口与液晶显示模块相连接,单片机通过对该并行接口输出状态的编程操作,完成对液晶显示模块所需时序的操作和数据的传输。80C51 单片机的 P0 口与液晶显示模块通过总线连接用来驱动液晶屏,P2 口中的 P2.0~P2.3 与液晶显示模块连接控制液晶屏。液晶显示模块中最重要的就是显示存储器,因为液晶屏上会直接显示存储器中所存储的内容。于是利用汉字库内码转换程序把汉字的二进制代码存储于液晶显示模块的显示存储器中,再通过单片机指令程序控制液晶屏上汉字的逐步显示和清除。

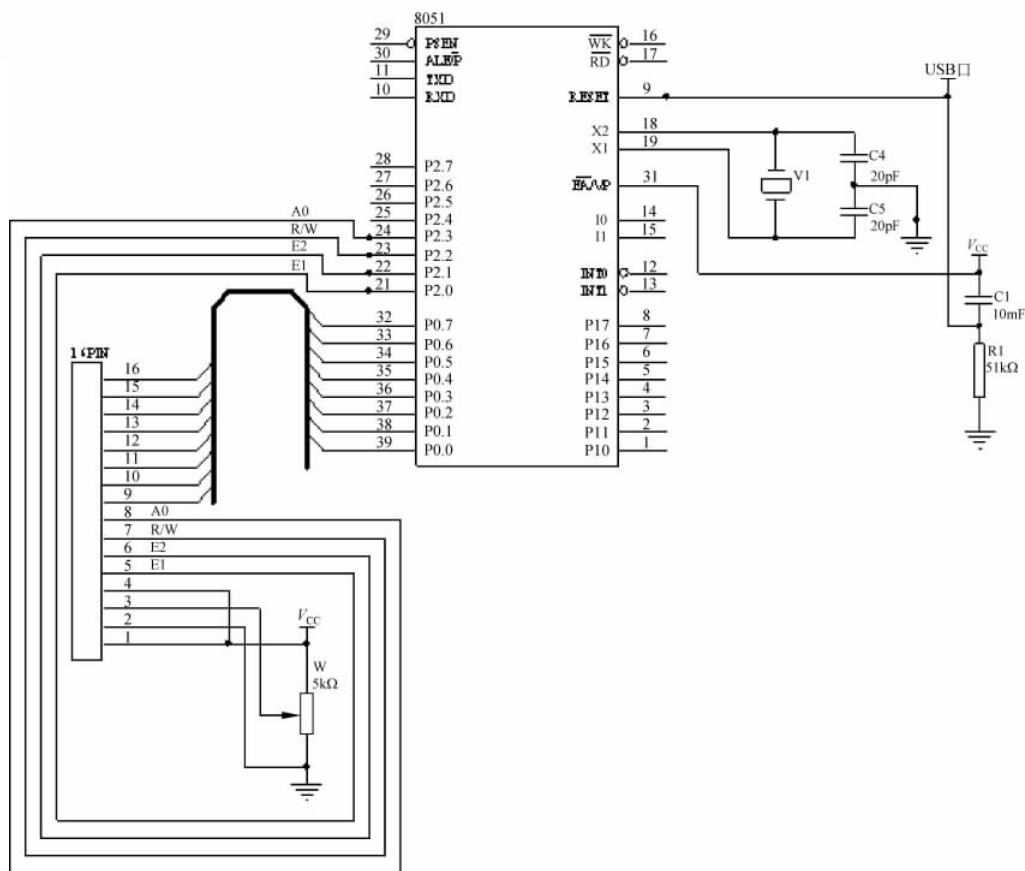


图 3-1 硬件设计原理图

3.3 80C51 单片机的硬件设计

3.3.1 总线型 DIP40 引脚封装

并行 I/O 引脚(32 个,分成 4 个 8 位口)。

- P0.0~P0.7: 一般 I/O 口引脚或数据/低位地址总线复用引脚;
- P1.0~P1.7: 一般 I/O 口引脚;
- P2.0~P2.7: 一般 I/O 口引脚或高位地址总线引脚;
- P3.0~P3.7: 一般 I/O 口引脚或第二功能引脚。

P3 的第二功能使用。

- P3.0: RXD(串行接口输入);
- P3.1: TXD(串行接口输出);
- P3.2: INT0(外部中断 0 输入);
- P3.3: INT1(外部中断 1 输入);
- P3.4: T0(定时/计数器 0 的外部输入);
- P3.5: T1(定时/计数器 1 的外部输入);

- P3.6: WR(片外数据存储器“写”选通控制输出);
- P3.7: RD(片外数据存储器“读”选通控制输出)。

3.3.2 80C51 的数据传送指令

- 算术运算类指令(24 条);
- 逻辑运算与循环类指令(24 条);
- 控制转移类指令(17 条);
- 位操作类指令(17 条)。

3.4 内置 SED1520 图形液晶显示模块

3.4.1 显示模块的电路特性

内置 SED1520 图形液晶显示模块的驱动控制系统是由两片 SED1520 组成,所以其电路特性实际上就是 SED1520 的电路特性。应用在液晶显示模块上最广的是 SED1520F0A,它的特性如下:

- (1) 具有 16 个行驱动输出和 61 个列驱动输出。
- (2) 可直接与 80 系列微处理器相连亦可直接与 68 系列微处理器相连。
- (3) 可以与 SED1520 级联使用以便扩展行列驱动能力,SED1520 按结构可分为控制部、驱动部和接口部。

3.4.2 控制部

控制部是由显示时序发生器和显示存储器以及管理电路两大部分组成。

1. SED1520 的引脚功能介绍

SED1520 功能原理图如图 3-2 所示。

2. 显示存储器部分

显示存储器部分是 SED1520 的核心部分。SED1520 具有 2560 位显示存储器,结构是一个 80×32 位的方阵形排布。显示存储器为双端口存储器结构。SED1520 列驱动仅有 61 路驱动输出,所以在 一行 80 位中仅有 61 位传输给驱动器使用。显示行地址指针是用来选择当前要传输的数据行。SED1520 把显示存储器列地址计数器设计为双向加一功能,既可以设置为自左向右加一,地址 0 对应驱动输出 SEG0,也可以设置成自右向左加一,地址 79(4FH)对应驱动输出 SEG0。

显示存储器的管理电路是由页地址指针、列地址指针、数据输入输出缓冲器和显示行地址指针等组成。

SED1520 中的显示 RAM 共有 32 行 80 列,其结构如图 3-3 所示。

3.4.3 驱动部

驱动部具有 16 路行驱动输出和 61 路列驱动输出。单片 SED1520 可以驱动 61×16 点阵的液晶显示器件。级联两片 SED1520 可以驱动 122×32 点阵的液晶显示器件。驱动部

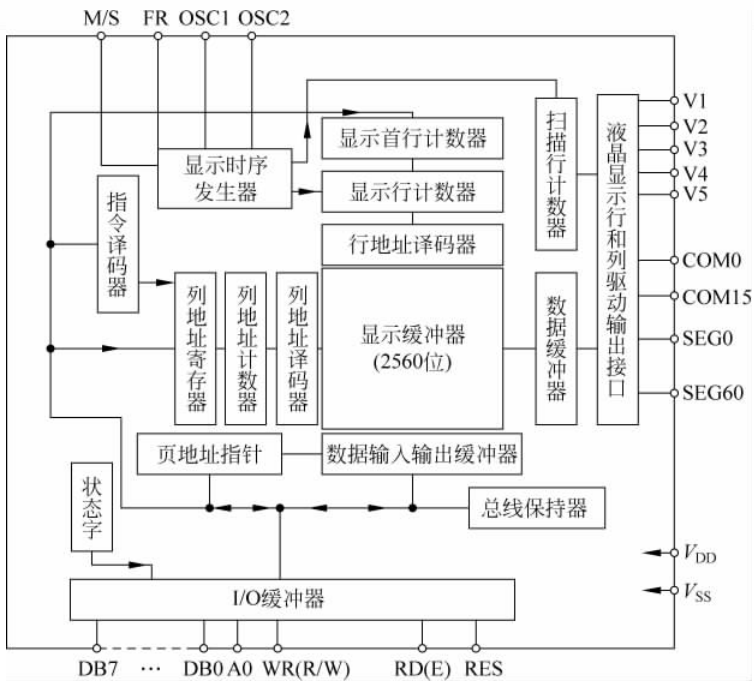


图 3-2 SED1520F0A 原理框图

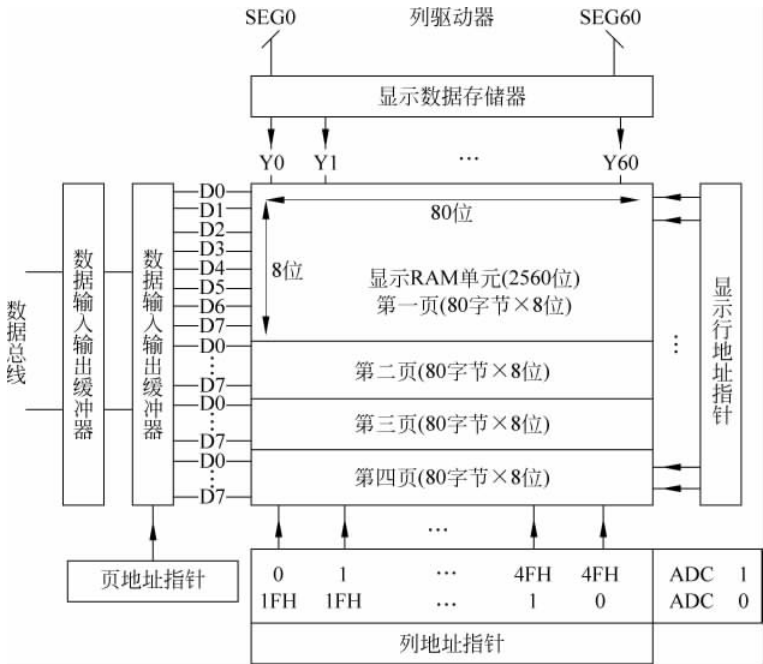


图 3-3 SED1520 显示存储器的结构原理

是由扫描行计数器、显示数据锁存器、液晶显示行驱动电路、液晶显示列驱动电路以及相关的外引脚组成。显示数据锁存器是用于锁存当前显示行的显示数据的。它只能实现显示存储器驱动数据传输所提供的 80 位驱动显示数据的 61 位的显示状态。

列驱动输出与列地址计数器输出的对应关系为：

当 $ADC=0$ 时，表示正向，\$ 0 对应 SEG0，\$ 60 对应 SEG60。

当 $ADC=1$ 时，表示逆向，\$ 13 对应 SEG60，\$ 79 对应 SEG0。

驱动部各引脚功能如下：

COM0~COM15：16 路行驱动输出。

SEG0~SEG60：61 路列驱动输出。

$V_1 \sim V_5, V_{DD}$ ：液晶显示驱动电源 ($V_{DD} \geq V_1 \geq V_2 \geq V_3 \geq V_4 \geq V_5$)。

V_{DD}, V_5 ：行、列选择电源的高、低电平。

V_1, V_4 ：行未选电压的高、低电平。

V_2, V_3 ：列未选电压的高、低电平。

液晶显示器件驱动结构示意图如图 3-4 所示。



图 3-4 液晶显示器件驱动结构示意图

3.4.4 接口部

SED1520 与计算机的接口信号如下：

DB0~DB7：三态，数据总线。

A0：输入，通道选择信号， $A0=0$ 为数据通道； $A0=1$ 为指令通道。

RES：输入，复位端和接口是时序类型设置端， $RES=0$ 表示接口适配 Intel8080， $RES=1$ 表示接口适配 M6800。

在 Intel8080 时序下，SED1520 的接口操作信号为读操作信号 RD，写操作信号 WR，接口时序表如表 3-1 所示。

表 3-1 在 Intel8080 时序下接口时序

A0	RD	WR	DB7~DB0	功 能
X	1	1	高阻	数据总线释放
0	0	1	输出	读状态字
0	1	0	输入	写指令代码
1	0	1	输出	读显示数据
1	1	0	输入	写显示数据

在 M6800 时序下，SED1520 的接口操作信号为读/写选择信号 R/W，使能信号 E。接口时序表如表 3-2 所示。

在 E 的下降沿处，数据或指令被锁存进 SED1520 内部的数据或指令寄存器。

在 E 高电平期间，数据或状态被读到数据总线上。有些模块提供了时序选择引脚 RES，用户可以自由选择以上两种时序。

A0	R/W	E	DB7~DB0	功 能
X	X	0	高阻	总线释放
0	0		输入	写指令代码
0	1		输出	读状态字
1	0		输入	写显示数据
1	1		输出	读显示数据

M/S 端接电源 V_{ss} , SED1520F0A 工作在主工作方式下; M/S 端接电源地, SED1520F0A 工作在从工作方式下。主工作方式下, SED1520F0A 负责上半屏 16 行的驱动和左半屏的列驱动, 从工作方式下, SED1520F0A 负责下半屏 16 行的驱动和右半屏的列驱动。

SED1520 液晶显示驱动器共有 13 种显示指令,可分为两大类:一类为显示方式的设置指令,表 3-3 前六条为这类指令,它们只需在初始化程序中写入一次就可以了;另一类为显示数据读/写操作指令,从表 3-3 第 7 条往下(包括读状态字)都是这类指令,它们需要经常使用。

[illegible]

3.4.6 内置 SED1520 图形液晶显示模块的接口技术

计算机控制 SED1520 图形液晶显示模块有直接访问形式和间接访问形式两种连接。本设计使用的是间接控制方式,这种间接控制方式是计算机通过自身的或系统的并行接口与液晶显示模块连接,如 8031 的 P0 口和 P2 口。电路中模块的数据总线接至 8031 的 P0 口,控制信号 A0 接至 P2.3, R/W 接至 P2.2, E1 接至 P2.0, E2 接至 P2.1。这种间接控制方式的电路简单,时序通过编程实现,这对于高速计算机控制液晶显示模块是非常有效的。

下面以表 3-4 和时序图 3-5 介绍间接控制方式驱动程序。由于 E1 和 E2 的写指令代码子程序、写显示数据子程序、读显示数据子程序都相同,在此只介绍 E1 的写指令代码子程序,写显示数据子程序,读显示数据子程序。

表 3-4

A0	R/W	E	DB7~DB0	功 能
X	X	0	高阻	总线释放
0	0		输入	写指令代码
0	1		输出	读状态字
1	0		输入	写显示数据
1	1		输出	读显示数据

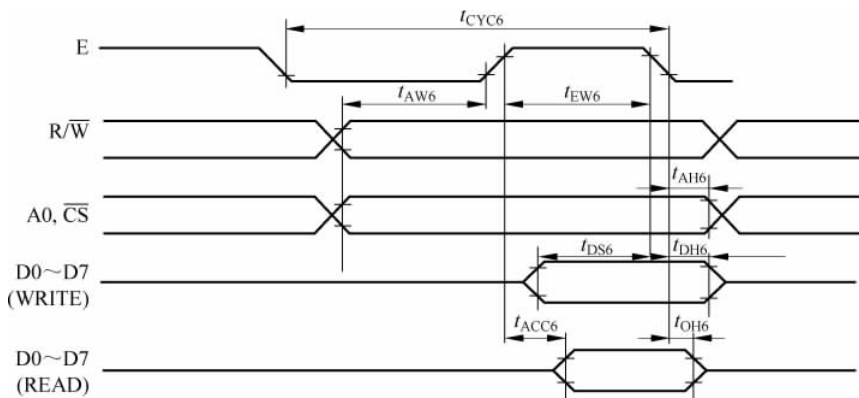


图 3-5 SED1520 M6800 读/写时序图

3.5 软件编程部分

流程图如图 3-6 所示。

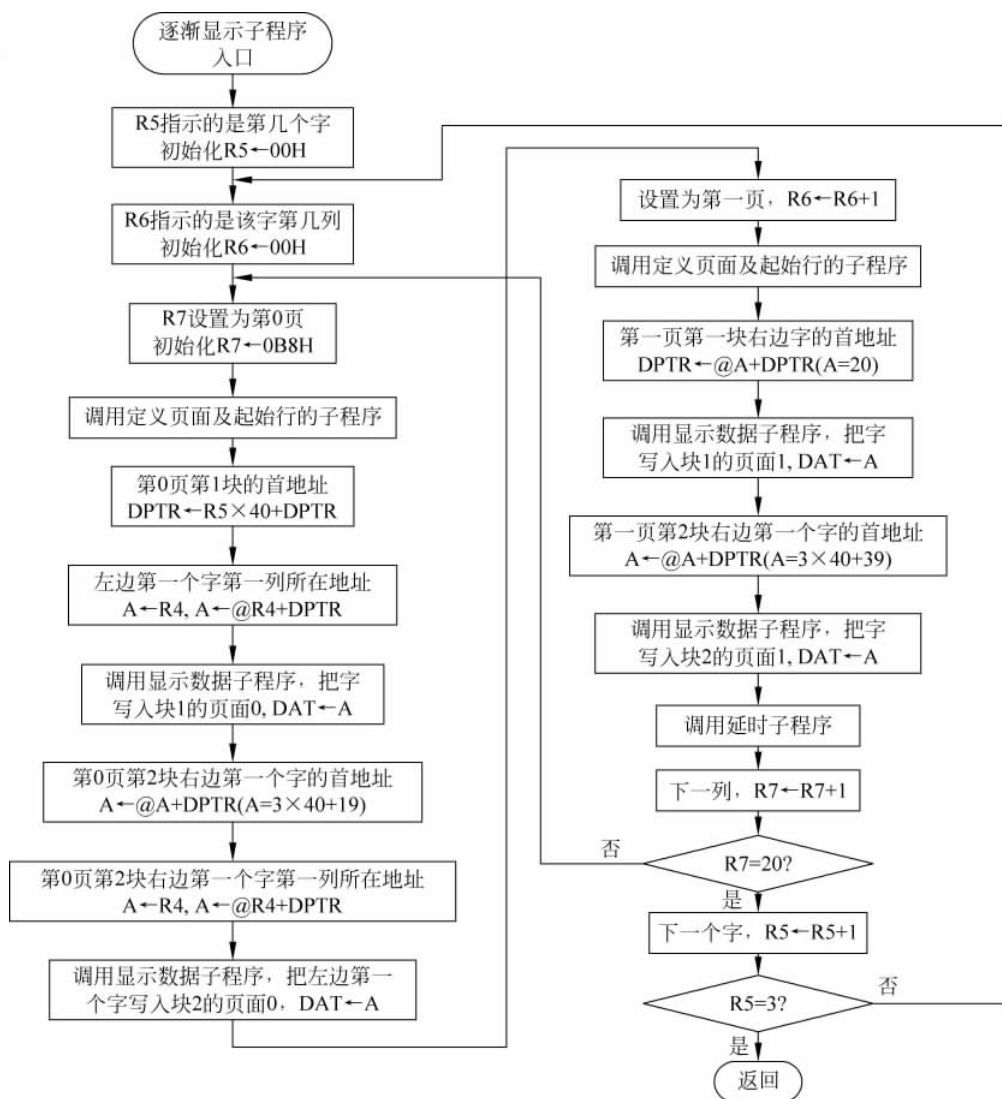


图 3-6 流程图

程序如下：

```

ORG 0000H           ;程序入口,PC 为 0000H
LJMP MAIN
ORG 0003H           ;设置外中断入口地址
LJMP INTO           ;外中断 0 入口
ORG 000BH           ;定时中断 0 入口
LJMP T0
ORG 0013H           ;外中断 1 入口
LJMP INT1
ORG 001BH           ;定时中断 0 入口
LJMP T1
ORG 0023H           ;串行中断入口
LJMP SS
  
```



```

ORG 0100H
COM    EQU 33H           ;命令字
DAT    EQU 34H           ;显示数据
A0     EQU P2.3
RW     EQU P2.2
E1     EQU P2.0           ;驱动片 1 使能信号
E2     EQU P2.1           ;驱动片 2 使能信号
MAIN:  MOV    SP, #60H
CLR    EA
;写指令代码字子程序(E1)
PR0:   CLR    A0
SETB   RW
PR01:  MOV    P0, #0FFH
SETB   E1               ;读操作要求 E1 为高电平
MOV    A,P0             ;读状态字
CLR    E1
JB     ACC.7, PR01       ;判"忙"则再读
CLR    RW
MOV    P0, COM           ;写指令字
SETB   E1               ;给一个下降沿,写操作发生在 E1 信号的下降沿
CLR    E1
RET
;写显示数据子程序(E1)
PR1:   CLR    A0
SETB   RW
PR11:  MOV    P0, #0FFH
SETB   E1               ;读操作要求 E1 为高电平
MOV    A,P0             ;读状态字
CLR    E1
JB     ACC.7, PR11       ;判"忙"则再读
SETB   A0
CLR    RW
MOV    P0, DAT           ;写数据
SETB   E1               ;给一个下降沿,写操作发生在 E1 信号的下降沿
CLR    E1
RET
;读显示数据子程序(E1)
PR2:   CLR    A0
SETB   RW
PR21:  MOV    P0, #0FFH
SETB   E1               ;读操作要求 E1 为高电平
MOV    A,P0             ;读状态字
CLR    E1
JB     ACC.7, PR21       ;判"忙"则再读
SETB   A0
MOV    P0, #0FFH
SETB   E1
MOV    DAT, P0           ;读数据
CLR    E1
RET
;初始化液晶

```

```

START:  MOV COM, # 0E2H      ;复位
        LCALL PR0            ;PR0 为驱动块 1 的写命令字子程序
        LCALL PR3            ;PR3 为驱动块 2 的写命令字子程序
        MOV COM, # 0A4H      ;关闭休眠状态
        LCALL PR0
        LCALL PR3
        MOV COM, # 0A9H      ;设置 1/32 占空比
        LCALL PR0
        LCALL PR3
        MOV COM, # 0A0H      ;正向扫描
        LCALL PR0
        LCALL PR3
        MOV COM, # 0C0H      ;设置显示起始行为第一行
        LCALL PR0
        LCALL PR3
        MOV COM, # 0AFH      ;开显示设置
        LCALL PR0
        LCALL PR3
        ;清屏
CLEAR:   MOV R4, # 00H      ;页面地址
CLEAR1:  MOV A, R4
        ORL A, # 0B8H
        MOV COM, A          ;设置页面
        LCALL PR0
        LCALL PR3
        MOV COM, # 00H      ;设置列地址为 0
        LCALL PR0
        LCALL PR3
        MOV R3, # 50H      ;一页清 80 个字节
CLEAR2:  MOV DAT, # 00H    ;显示数据为"0"
        LCALL PR1          ;PR1 是块 1 的写显示数据子程序
        LCALL PR4          ;PR4 是块 2 的写显示数据子程序
        DJNZ R3, CLEAR2    ;页内字节清零循环
        INC R4              ;下页
        CJNE R4, # 04H, CLEAR1 ;RAM 区清零循环
        ;开始显示
INT0:
MAIN12: LCALL S_DISP        ;调用逐渐显示子程序
        LCALL S_CLR         ;调用逐渐清除子程序
        LJMP MAIN12         ;显示停电
        RETI                ;中断返回
        ;延时子程序
DELAY:   MOV R1, # 00H
        MOV R2, # 00H
        DELAY1:  NOP
        DJNZ R2, DELAY1
        DJNZ R1, DELAY1
        RET
        ;逐渐显示子程序
S_DISP:  MOV R5, # 00H      ;指示显示的是第几个字
DISP1:  MOV R7, # 00H      ;指示该字的第几列

```



```

DISP2:    MOV R6, # 0B8H    ;设置为第 0 页
LCALL D_SET    ;D_SET 为定义页面及起始列的子程序
MOV DPTR, # TAB    ;取显示字地址
MOV A, R5
MOV B, # 40
MUL AB
ADD A, DPL
MOV DPL, A
MOV A, B
ADDC A, DPH
MOV DPH, A    ;此时 DPTR 指向该字首地址
MOV A, R7
MOVC A, @A + DPTR
MOV DAT,
LCALL PR1    ;显示数据写入块 1 的页面 0
MOV A, # 19
CLR C
SUBB A, R7
ADD A, # 120
MOVC A, @A + DPTR
MOV DAT, A
LCALL PR4    ;显示数据写入块 2 的页面 0
INC R6    ;设置为第一页
LCALL D_SET
MOV A, R7
ADD A, # 20
MOVC A, @A + DPTR
MOV DAT, A
LCALL PR1    ;显示数据写入块 1 的页面 1
MOV A, # 39
CLR C
SUBB A, R7
ADD A, # 120
MOVC A, @A + DPTR
MOV DAT, A
LCALL PR4    ;显示数据写入块 2 的页面 1
LCALL DELAY    ;延时
INC R7    ;下一列
CJNE R7, # 20, DISP2    ;一直循环, 写完一个字
INC R5    ;下一个字
CJNE R5, # 3, DISP1    ;一直写完三个字
RET

D_SET:    MOV COM, R6
LCALL PR0
LCALL PR3
MOV A, R5
MOV B, # 20
MUL AB
ADD A, R7
MOV R4, A    ;R4 中放列号
MOV COM, R4

```

```

LCALL PR0                ;设置 E1 的列
MOV A, #60
CLR C
SUBB A, R4
MOV COM, A
LCALL PR3                ;设置 E2 的列
RET
;逐渐清除子程序
S_CLR:    MOV R7, #61      ;每块的列数
CLR0:     MOV R6, #0B8H    ;第 0 页
CLR1:     MOV COM, R6
LCALL PR0
LCALL PR3
MOV A, R7
DEC A
MOV COM, A
LCALL PR0                ;设置 E1 的列
MOV DAT, #00H
LCALL PR1                ;清零
CLR C
MOV A, #61
SUBB A, R7
MOV COM, A
LCALL PR3                ;设置 E2 的列
MOV DAT, #00H
LCALL PR4                ;清零
INC R6                    ;下一页
CJNE R6, #0BAH, CLR1
LCALL DELAY
DEC R7
CJNE R7, #00H, CLR0
RET
T0:       RETI
INT1:     RETI
T1:       RETI
SS:       RETI

TAB: DB 00H, 00H, 40H, 42H, 44H, 0CCH, 00H, 00H, 0F1H, 91H;通
      DB 95H, 0F9H, 95H, 93H, 0F9H, 10H, 00H, 00H, 00H, 00H;
      DB 00H, 00H, 00H, 40H, 20H, 1FH, 20H, 40H, 0BFH, 84H;
      DB 84H, 0BFH, 94H, 0A4H, 9FH, 0C0H, 40H, 00H, 00H, 00H;
      DB 00H, 00H, 80H, 40H, 20H, 0F8H, 07H, 24H, 24H, 24H;信
      DB 25H, 26H, 24H, 24H, 0B4H, 26H, 04H, 00H, 00H, 00H;
      DB 00H, 00H, 00H, 00H, 00H, 0FFH, 00H, 01H, 0FDH, 45H;
      DB 45H, 45H, 45H, 45H, 0FDH, 01H, 00H, 00H, 00H, 00H;
      DB 00H, 00H, 00H, 04H, 04H, 04H, 04H, 04H, 04H, 0FCH;工
      DB 04H, 04H, 04H, 04H, 06H, 04H, 00H, 00H, 00H, 00H;
      DB 00H, 00H, 20H, 20H, 20H, 20H, 20H, 20H, 20H, 3FH;
      DB 20H, 20H, 20H, 20H, 20H, 30H, 20H, 00H, 00H, 00H;
      DB 00H, 00H, 00H, 10H, 60H, 80H, 00H, 0FFH, 00H, 00H;业
      DB 00H, 0FFH, 00H, 00H, 0C0H, 30H, 00H, 00H, 00H, 00H;

```

```

DB 00H, 00H, 40H, 40H, 40H, 47H, 40H, 7FH, 40H, 40H;
DB 40H, 7FH, 44H, 43H, 40H, 60H, 40H, 00H, 00H, 00H;
DB 00H, 00H, 40H, 40H, 48H, 48H, 48H, 0C8H, 78H, 4FH; 专
DB 48H, 48H, 48H, 4CH, 48H, 60H, 40H, 00H, 00H, 00H;
DB 00H, 00H, 00H, 00H, 00H, 00H, 02H, 23H, 22H, 42H;
DB 62H, 92H, 0AH, 06H, 02H, 00H, 00H, 00H, 00H, 00H;
DB 00H, 00H, 24H, 24H, 0A4H, 0FEH, 0A3H, 22H, 20H, 71H; 程
DB 42H, 42H, 42H, 42H, 42H, 7FH, 02H, 00H, 00H, 00H;
DB 00H, 00H, 08H, 06H, 01H, 0FFH, 00H, 43H, 41H, 49H;
DB 49H, 49H, 7FH, 49H, 4DH, 69H, 41H, 00H, 00H, 00H;

```

END

3.6 汉字库内码转换

所显示字符的点阵数据可从标准字库——汉字库内码转换中提取,需要正确掌握字库的编码方法和字符定位的计算。具体使用方法是运行上述可执行程序,根据提示输入所需字符。程序将该字符的码表提取出来,用十六进制数表示出来并存储在该字符的数据存储区。但需要注意字节排列顺序、字节中每一位与具体显示点的一一对应关系,如图 3-7 所示。



图 3-7 汉字字库提取程序

3.7 编程调试及仿真

3.7.1 编程调试及仿真软件介绍

程序的调试及仿真部分使用的是伟福仿真调试软件。其中仿真头使用的是 POD8051 仿真头,它用于仿真 P0、P2 口作为总线工作方式的 8031/32、8051/52 系列及兼容单片机;仿真器使用的是 P51 型仿真器,即 Philips 通用仿真器。

3.7.2 编译器设置及使用

打开仿真器设置,选择所使用的仿真器和仿真头类型以及所仿真 CPU 的类型。

将编译好的源程序设置为. ASM 文件,然后再新建一个项目,在该项目中打开源程序进行编译,在编译没有错误后就可以调试程序了。将仿真器连接到计算机上,将仿真头接到仿真器,检查连线无误后接上电源,打开仿真器的电源开关,运行程序并观察结果。

3.8 总结

本设计采用 16×16 点阵汉字,液晶显示器显示一个 16×16 点阵需要送进 32 个字节,因此必须将其送入两缓冲区中的相对列中。该系统将汉字显示设置成两页,每一页的字符块写入字模数为 16,当 16 个字模写完后,再转入下一页写入剩下的 16 个字模。汉字显示也存在中间结合部的处理问题,其处理方法与字符型的基本相同。即把 MGLS12032 屏分成 7×2 个汉字块。其中每行第四个汉字在 SED1520(1)占 12 个字节,在 SED1520(2)占四个字节,在编程时要转换两个芯片的 RAM 区。可以从计算机汉字字库内提取的汉字数组格式正好与 SED1520 的 RAM 格式相差 90° 。所以在提取汉字字模时要先旋转 90° ,才能在显示器上显示出正常格式。由于 MGLS12032 显示屏分为两页 SED1520(1)和 SED1520(2),所以显示程序应有两种,这两种程序除相关的地址不同外,其余均相同。本设计以 FPGA 为控制核心实现了对字符点阵液晶的控制,该方案硬件电路简单,软件程序简洁,对液晶的控制简单而且稳定,且可灵活改动,若要改变液晶显示的汉字,只需改变 ROM 表中的字模即可,实践证明,该设计是对液晶应用的一种行之有效的方法。