

微 处 理 机 原 理 及 应 用

陆忠华 编著

下 册

武汉钢铁学院计算机教研室

TP 31
82

下 册 目 录

第四章 通用外设接口技术举例	(1)
§4.1 发光二极管字符显示器与显示程序	(1)
一、显示一位16进制数的显示器接口电路与显示程序	(1)
二、同时显示六位16进制数的显示接口电路与显示程序	(3)
例: TP801A单板机监控程序中的显示程序	(3)
三、硬件扫描显示器接口电路及其基本工作原理	(5)
四、能显示34种字符的16段LED显示器及其接口电路基本工作原理	(9)
§4.2 键盘输入接口电路与键扫描电路	(7)
一、非编码键盘	(8)
例1 8×8非编码键盘接口电路与键扫描程序	(8)
例2 TP801A单板机键扫描与键分析程序	(11)
二、编码键盘	(17)
§4.3 数模模数转换技术举例	(19)
例1 三角波发生程序	(19)
例2 数据采集程序	(19)
§4.4 打印机接口电路与打印程序	(20)
§4.5 电传打字机键盘接口电路与读数程序	(22)
一、采用SIO接口的方案	(22)
二、采用PIO接口的方案	(24)
§4.6 盒式磁带机接口电路与程序	(27)
一、磁带的记录格式	(27)
二、盒式磁带机的接口电路	(28)
三、写带与读带的操作过程	(30)
四、TP801A单板机的写带程序分析	(31)
五、TP801A单板机的读带程序分析	(35)
§4.7 EPROM写入接口电路与程序	(39)
一、TP801A EPROM写入接口电路	(39)
二、TP801A单板机写EPROM操作步骤	(41)
三、EPROM写入程序(PROM键处管程序)	(41)
§4.8 CRT显示器接口电路与程序	(43)
一、CRT字符显示原理	(44)
二、TMC-30 CRT显示器接口电路分析	(45)
三、CRT字符显示程序	(49)
习题四	(50)
第五章 微处理器应用简单实例	(52)
§5.1 顺序控制系统	(52)

§5.2	小功率直流电机调速系统	(56)
§5.3	可编音乐程序系统	(59)
§5.4	实时计时系统	(65)
§5.5	交通灯管理系统	(67)
§5.6	双座标图形发生器系统	(69)
§5.7	数据采集与报警系统	(72)
	习题五	(79)
第六章	16位微处理器及单片微处理机简介	(78)
§6.1	INTEL8086微处理器简介	(78)
一、	INTEL8086微处理器的外部引脚及内部结构框图	(78)
二、	8086指令系统的特点	(80)
三、	8086微处理机的最小系统	(81)
四、	INTEL8088微处理器	(81)
§6.2	Z8000微处理器	(82)
一、	Z8000微处理器的外部引脚及内部结构框图	(82)
二、	Z8000指令的特点	(85)
三、	Z800微处理器	(86)
§6.3	M68000微处理器简介	(86)
一、	M68000外部引脚	(86)
二、	M68000的内部寄存器库	(88)
三、	M68000指令系统特点	(89)
§6.4	INTEL8048系列单片微处理机简介	(90)
一、	INTEL8048系列单片微处理机	(90)
二、	8048的外部引脚内部结构框图	(91)
三、	8048微处理机的指令系统的特点	(92)
第七章	TP801A监控程序分析	(93)
§7.1	TP801A单板机监控程序主流程与地址空间分配	(93)
一、	TP801A单板机监控程序主流程	(93)
二、	TP801A单板机存储空间分配	(95)
三、	用户栈单元分配	(95)
四、	接口地址分配表	(96)
§7.2	RESET按钮、MON及MON'键的处理程序	(96)
一、	RESET按钮处理程序	(96)
二、	MON及MON'键的处理程序	(98)
§7.3	EXEC键与STEP键处理程序	(100)
一、	EXEC键与STEP键的操作步骤	(100)
二、	EXEC键处理程序	(101)
三、	STEP键处理程序	(102)
§7.4	读写存储单元、接口数据寄存器及CPU内部寄存器的程序	(103)

一、操作步骤	(103)
二、MEM键、PORT键、REG键及REG'键处理程序	(104)
三、NEXT键与LAST键处理程序	(106)
四、改写存储单元、接口及CPU内部寄存器的程序	(107)
§7.5 断点的设置、装配与清除	(110)
一、断点设置与清除的操作步骤	(111)
二、B.P键处理程序	(112)
三、有断点情况下, EXEC键的执行过程	(112)
§7.6 TP801A监控程序中的通用子程序	(115)
习题七	(116)
主要参考资料	(117)
附录: TP801A监控程序清单	(118)

第四章 通用外设接口技术举例

§4.1 发光二极管字符显示器与显示程序

一、显示一位16进制数的显示器接口电路与显示程序

图4.1(a)所示为采用七段发光二极管(LED)显示器的接口电路。图中采用74L273型8D触发器作为输出接口电路,起显示模型锁存器的作用。显示模型从 $Q_0 \sim Q_6$ 端输出,经半导体三极管驱动电路与发光二极管a~g连接。a~g按图4.1(b)所示方式组成七段发光二极管显示器。

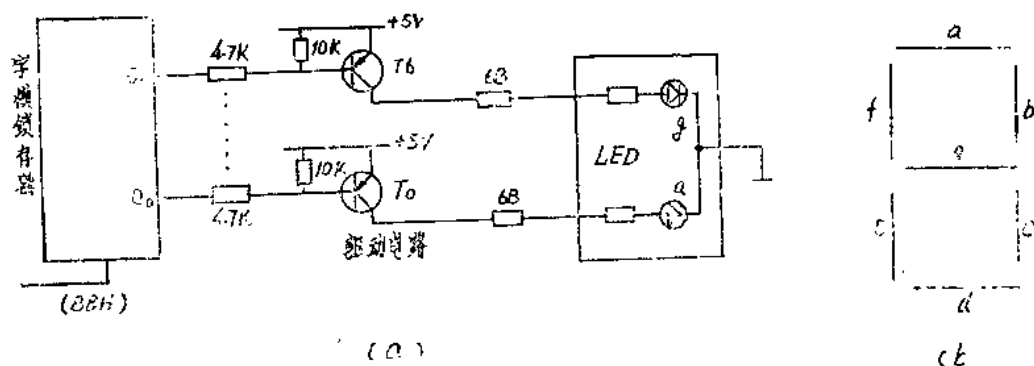


图4.1 数字显示接口电路

七段LED显示器只能显示0~F等16进制数码字符,而不能显示A~Z的任意英文字母及其它运算符号。又图4.1电路中只有一个LED显示器,故只能显示一位16进制数码字符。关于同时显示六位16进制数的显示器接口电路及显示其它字符的电路将在后面介绍。

下面举例说明图4.1的显示程序。

设已知在DATA单元中有一个一位16进制数。试编一程序。执行后,能将此数通过图4.1接口在LED显示器中显示出来。若DATA单元中的数>0FH则应拒绝显示。设字符锁存器的口地址为88H; DATA=2040H。

解:首先分析显示模型。

当字模锁存器输出数据 Q_0 位为另电平时, T_0 导通, 发光二极管a被点亮。同样, 当 $Q_1 \sim Q_6$ 为另电平时, 发光二极管b~g被点亮。

为了显示字符“0”, 应当点亮除g以外的全部LED。故应使 $Q_0 = 1$, 其余输出位均为零(Q_7 在任何情况下保持为零)。由此, 可求得字符“0”的显示模型为。

Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0
0	1	0	0	0	0	0	0

对应16进制数为40H。

相仿地, 为了显示字符“1”, 可令b、c点亮其它熄灭。故字符“1”的显示模型为:

Q ₇	Q ₆	Q ₅	Q ₄	Q ₃	Q ₂	Q ₁	Q ₀
0	1	1	1	1	0	0	1

对应16进制数为79H。

可将各种情况下的显示模型列表如下：

字 符	0	1	2	3	4	5	6	7	8
模 型	40H	79H	24H	30H	19H	12H	02H	78H	00H
字 符	9	A	b	c	d	E	F	熄灭	提示
模 型	18H	08H	03	46H	21H	06H	0EH	7FH	3FH

将以上显示模型存入以SEGPT为首地址的存储区。不难看出，所谓显示程序主要是一个查表程序。取得显示模型后，即送字模锁存器输出到显示电路。

可写出源程序如下：

标 号	操 作 码	操 作 数	注 解
DISPLAY	ORG	2000H	
	LD	B, 7FH	熄灭符送B寄存器
	LD	A, (DATA)	取 数 据
	CP	10H	
	JR	NC, DISPLAY-\$	} 若A>0FH则输出熄灭符
	LD	DE, SEGPT	
	LD	H, 0	} A送L, 00H送H
	LD	L, A	
	ADD	HL, DE	求待显示字符表地址
	LD	B, (HL)	取显示模型
	LD	A, B	
	OUT	(88H), A	从88H 口输出
	HALT		
SEGPT	DEFB	40H, 79H, 24H, 30H, 19H, 12H, 02H, 78H, 00H, 18H, 08H, 03H, 46H, 21H, 06H, 0EH, 7FH	
DATA	EQU	2040H	
	END		

二、同时显示六位 16 进制数的显示接口电路与显示程序

为了同时显示N位数字，可用二种方案来实现。1)静态显示法。2)扫描显示法。

采用第一种方案时需采用N个字模锁存器（对应N个接口地址），分别与N个数码管电路相连接。则可将各显示模型一次送出后，即可同时进行显示，占用CPU时间减少。若采用CL002型CMOS—LED组合显示器件则更为方便，因这种器件同时具有寄存、译码、驱动与显示的功能（且与TTL兼容），可直接连至系统数据总线。

采用扫描显示法时，可行方案之一是CPU以等间隔时间轮流接通各数码管电路。由于数码管的惯性与人的视觉暂留现象，可达到同时显示六位数字的效果。这种方案称为软件扫描法。采用这种方案，硬件比较节省。

在TP801单板机中即采用软件扫描显示方案，硬件接线方式如图4.2。在电路中有六个数码管，可同时显示六位16进制数。

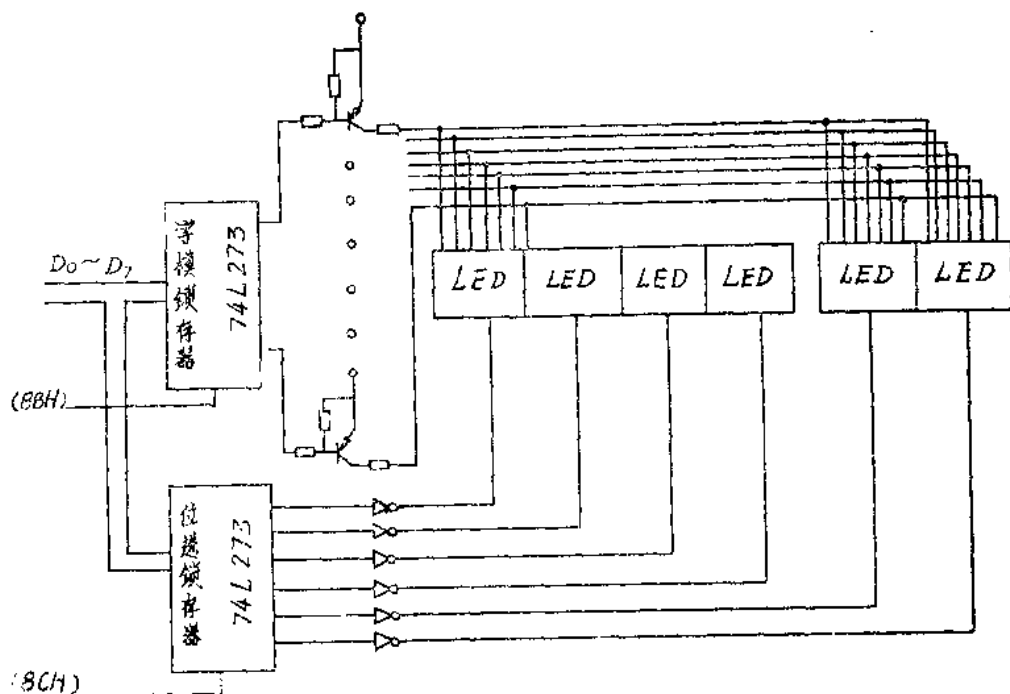


图4.2 数字显示接口电路

下面介绍TP801单板机的显示程序

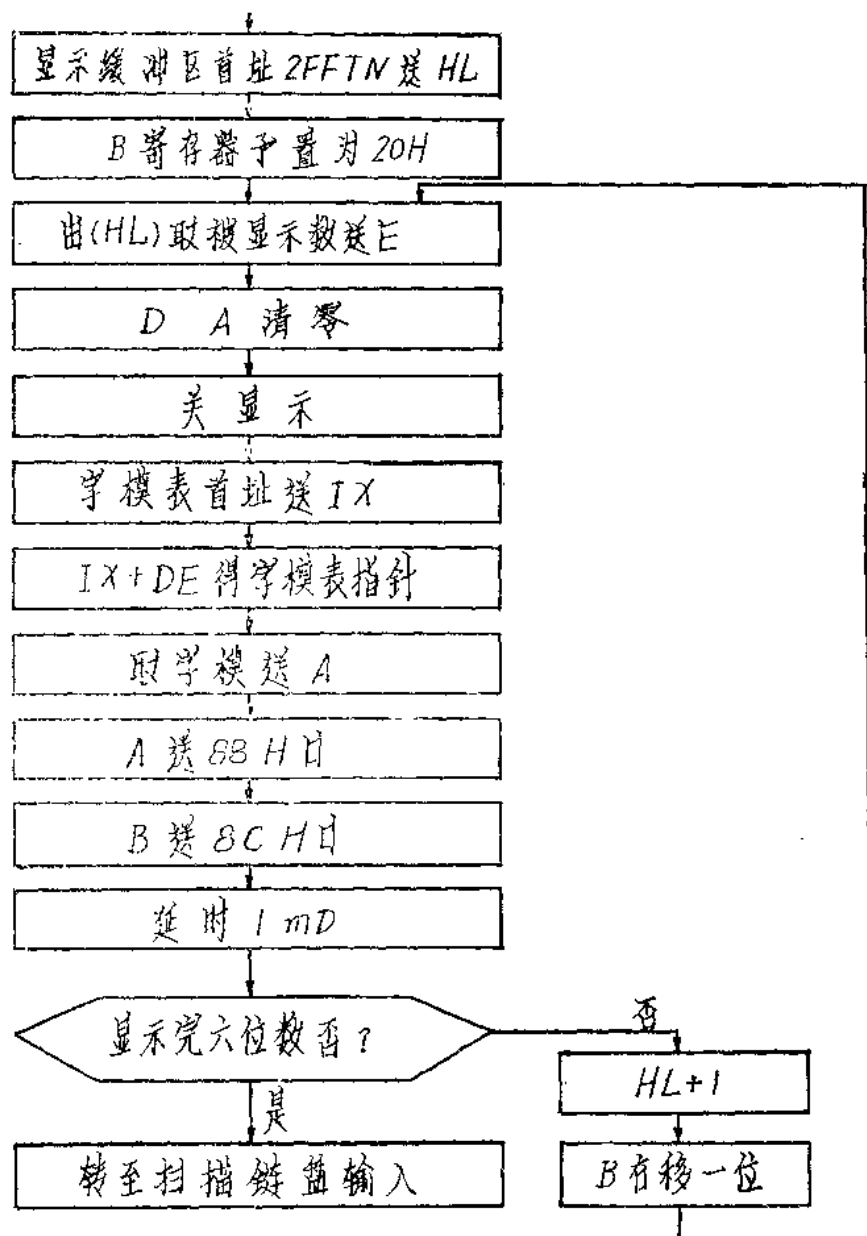
TP801单板机监控程序的入口条件、出口条件及程序清单见本书附录TP801监控程序中更新显示程序段（起始地址00F4H，末地址0122H）。试对程序进行分析讨论。

解：待显示的数码事先被予置在以2FF7H（标号DISMEN）为首地址的六个显示缓冲区中。为了轮流显示，首先将显示缓冲区指针指向DISMEN，并将存放数位选择模型的B寄存器内容予置为20H，利用点表法先将DISMEN单元内容变换为显示模型从88H口输出。20H经8CH口输出并经反相器反相后，加于六个数码管阴极的电平从左至右依次为：

LED	LED	LED	LED	LED	LED
0	1	1	1	1	1

于是最高位的数码管电路被接通而显示最高位的待显示数码。延迟 1ms 后，更新显示缓冲区地址及 B 寄存器内容，使次高位数码管显示。如此从左到右以 1ms 的间隔进行扫描。当显示完 6 位数后，转至键盘扫描程序。若有键则作键分析及键处理，无键则返回，重复以上扫描显示过程。

可画出 TP801 显示程序流程框图如图 4.3 读者可对照流程框图对显示程序清单具体分析。



IX——字模表指针

HL——显示缓冲区当前地址

B——当前位数指示器

E——保存被显示位的数据

图4.3 TP801A显示程序框图

三、硬件扫描显示器接口电路及其基本工作原理

采用下面所示的硬件扫描显示方法，可以减少CPU为进行软件显示所花费的时间，它只需一套驱动电路，即可实现N位显示。

接口电路接线方式如图4.4所示：

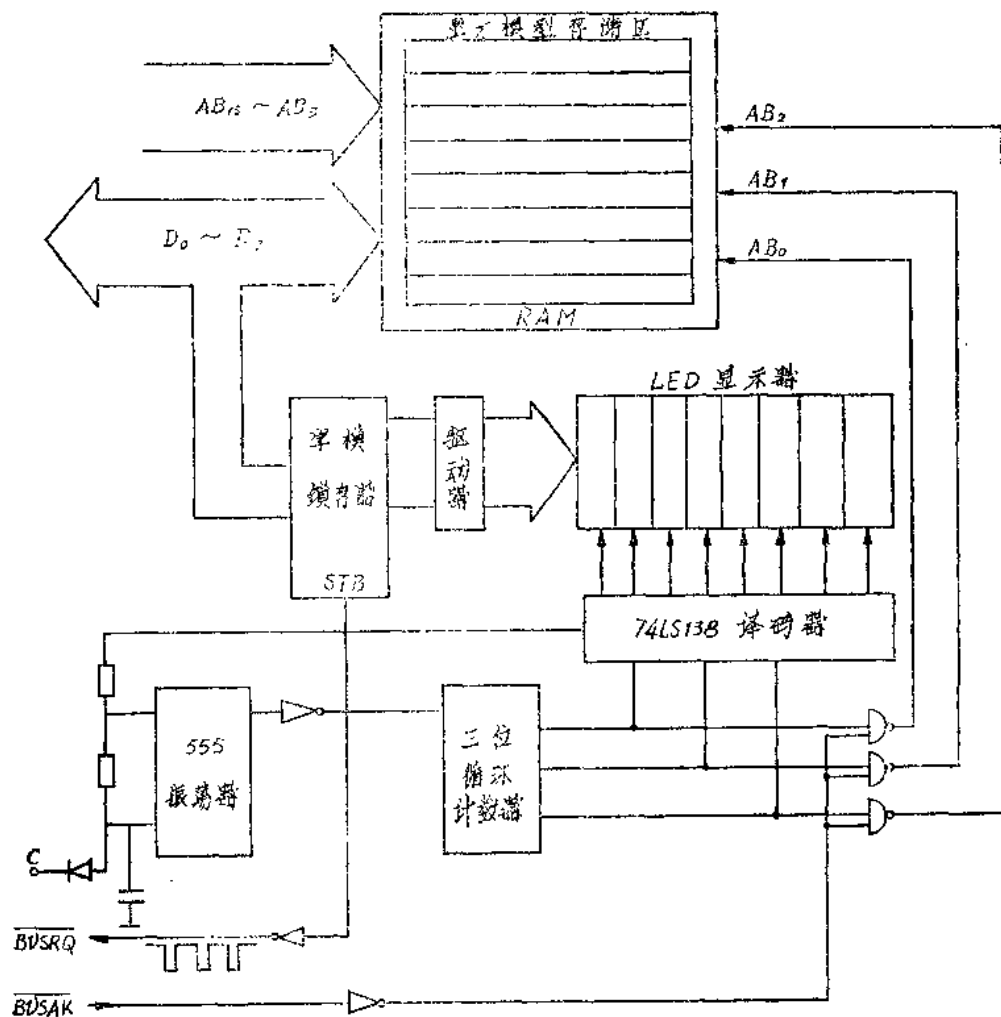


图4.4 硬件扫描显示接口电路图

在图4.4中采用具有控制端C的555振荡器作为硬件扫描的定时脉冲源。当其控制端为高电平时，将输出一连串的负脉冲，其脉宽为3—5 μ s，周期为1—2ms。555振荡器的输出，经过三位循环计数器连到三进八出译码器（74LS138）的输入端，使其8个输出端轮流为低电平，从而使8个数码管顺序依次被接通。

为了不占用CPU时间，显示模型的传送采用DMA方式。在图中用硬件组成了一个简

易的DMA控制器。首先从555振荡器输出端经缓冲器将负脉冲引出，送到CPU的总线申请端，作为总线申请信号。在三位循环计数器的输出端，连接三个与非门，用来在得到来自CPU总线响应信号后，输出显示模型单元的地址码的低三位。当CPU地址总线处于悬空状态时，存储器译码电路的地址码输入端全部相当于高电平。故显示模型存储单元在存储区中的位置，应根据地址总线悬空情况下，地址译码电路输出的地址信号来锁定，而不能任意指定。这是这一方案的缺点。当地址码低三位 $AB_0 \sim AB_2$ 随着三位循环计数器输出的变化而变化时，将依次选中对应的8个显示模型存储单元中的一个。必须注意的是，这里显示模型单元存放的不是待显示字符，而是其显示模型。因此，事先应通过软件，从显示缓冲区中取出待显示字符。利用查表法求出显示模型，存入8个显示模型单元。

此外，为了代替CPU在DMA期间发出读存储单元命令，应在电路中使用总线响应信号 $\overline{BUS\overline{AK}}$ 与 $\overline{RD}$ 及 $\overline{MREQ}$ 进行以下逻辑运算： $\overline{MEMR} = \overline{BUS\overline{AK}} \cdot (\overline{RD} + \overline{MREQ})$

当 $\overline{BUS\overline{AK}} = 0$ 时，即可发出读存储器命令 $\overline{MEMR}$ 。

从显示模型存储单元读出的显示模型将被锁存在字模锁存器中，并经过驱动电路同时加到8个显示管的输入端（但每次只接通一个）。关于电路的其它细节问题就不在此讨论了。

四、能显示64种字符的16段LED显示器及其接口电路基本工作原理

在ROCKWELL公司的AIM—65单板机中采用了能显示64种字符的16段LED显示器。这种显示器除能显示0~9十进制数符外，还能显示A~Z等26个英文字母，以及本书第一章表1.1 ASCII字符表中起始字符分别为SP、0、@及P等4列中的其它字符。这里我们简单介绍这类显示器的接口电路与基本工作原理。

显示器的接口电路如图4.5所示。

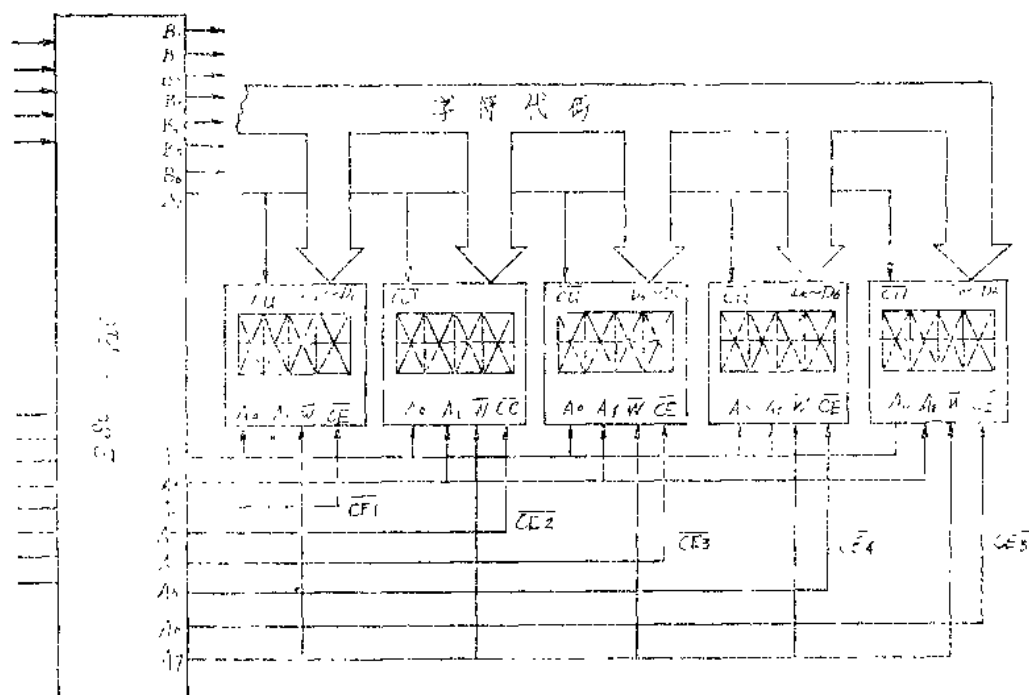


图4.5 LED字符显示器接口电路图

为了与本书其它内容统一起见,我们假定输出接口芯片为Z80—PIO。
在图中显示元件包括五个模块(DL—1416A型),每个模块由四个显示器组成。
16段显示器(见图4.6。)需采用双字节显示模型。

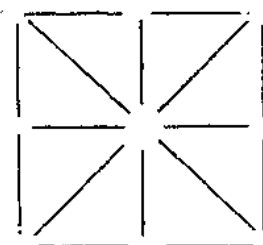


图4.6 16段LED显示器

根据字符的字形不难求出各自的显示模型。

显示电路的工作过程可简述如下:

从PIOA口送出片选与位选信号, $A_0 \sim A_2$ 用来对5个模块进行片选。当 $CE1 = 0$ 时选中第一模块; $CE2 = 0$ 时选中第二模块, 以此类推。 A_3 及 A_4 用来选择每一模块中的四个显示器, A_7 给出写命令

从PIOB口的 $B_0 \sim B_6$ 送出待显示字符的ASCII码, 加到各显示模块的 $D_0 \sim D_6$ 输入端。在DL—1416A模块内部有译码电路, 能自动完成ASCII码到显示模型的变换。

显示器提供了16段全亮时的图形作为光标符号, 当显示光标时令 $D_7 = B_7 = 0$ 。其它情况下 $D_7 = B_7 = 1$ 。

通过软件扫描, 可实现20个ASCII字符的同时显示。

§4.2 键盘输入接口电路与键扫描程序

键盘是微型计算机中实现人机对话的一种重要手段。它常与CRT显示器或电传打字机组合成为独立的终端设备。在TP801等单板计算机中, 键盘由0~F等16进制数码键与一些功能键组成。在具有汇编语言或高级语言软件的系统中则采用ASCII全键盘。

键盘实质上是一组按键矩阵。每个按键是一个机械开关。按动时前后沿均有弹跳过程, 通常这种暂态过程约10~20ms。克服弹跳的办法, 一种是硬件“滤波”式, 另一种是软件加延时程序。此外, 任何实用的键盘还必须考虑二个或二个以上按键同时按下兼并问题。

键盘有两种基本结构形式: 非编码键盘与编码键盘。前者靠CPU利用软件来完成键识别和产生代码。后者则利用专用硬件来完成这些功能。

下面我们先介绍非编码键盘的接口电路与利用软件产生代码的方法, 然后再介绍编码键盘。

一、非编码键盘

例1 8×8非编码键盘接口电路与键扫描程序

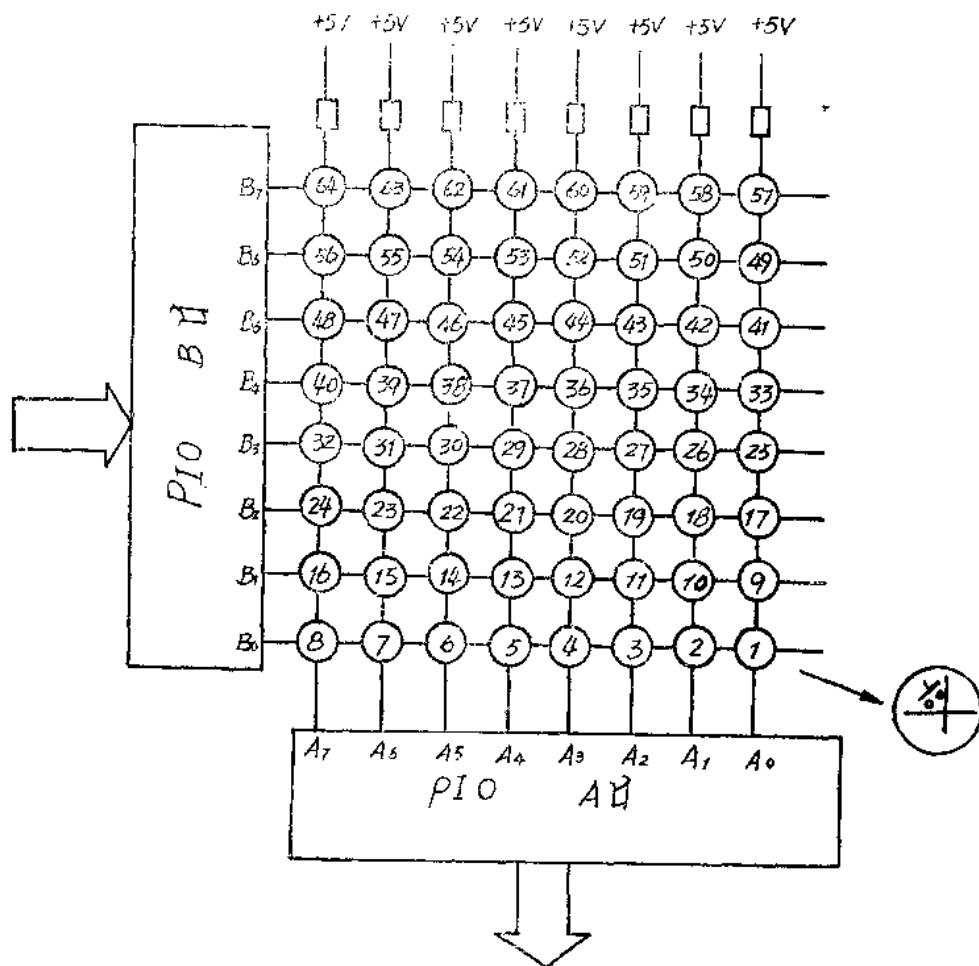


图 4.7 8×8非编码键盘接口电路图

图4.7所示为一8×8非编码键盘的接口电路。设PIO A口及B口地址分别为80H及81H。试编写一键输入程序。要求当按下某一按键时，CPU能将键号检测出来送存DATA单元。

解：解题思路如下：我们先置A口为输入口，B口为输出口，接着调用键扫描子程序来判断有键还是无键。在键扫描子程序中，先使B口所有行输出另电平，然后从A口读入列数据。若没有键按下，则所有列均应为高电平，即列数据为FFH。若列数据为FFH，则程序在循环中查询等待，直到列数据不为FFH为止，当判明有键输入后，本来可以接下来调用键分析程序以确定键号。但为了防止偶然性干扰，故采用二次扫描法。即延时20ms后，再重判一次。若无键，则说明为偶然性干扰，可返回循环。若仍为有键，则调用键分析子程序来确定键号。

为了取得键号，先在L寄存器中置键号初值为0，然后从B口输出分行扫描模型，先使第0行为低电平。从A口读入列数据，与FFH比较，若不相等则说明第0行有键输入。为了判别按键在那一列，先令键号加1。然后用RRA指令将列数据的最低位移入CF标志位。如CF=0，则键号即为1。否则，继续循环判键号。若第0列无键输入，则令键号加

8, 再扫描第1行, 以此类推。

下面画出主程序、键扫描子程序及键分析子程序流程图分别如图4.8、图4.9及图4.10。

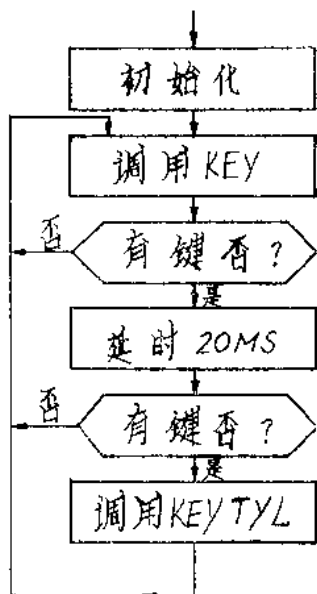


图4.8 主程序流程图

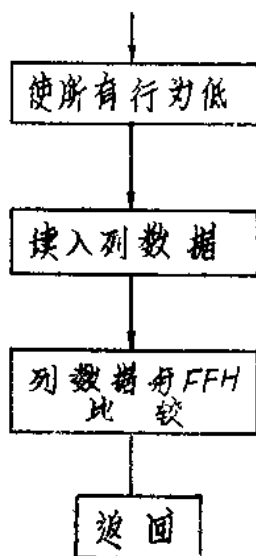


图4.9 键扫描子程序流程图

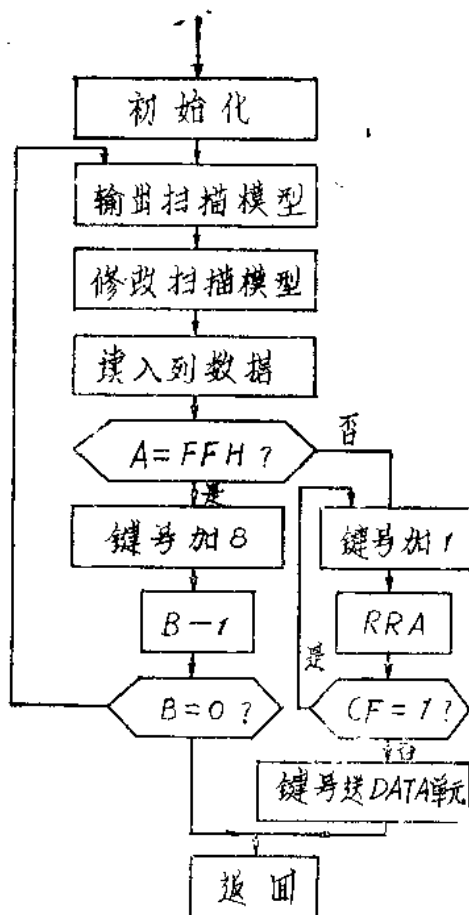


图4.10 键分析子程序流程图

根据程序框图不难写出Z80汇编语言程序如下:

标 号	操 作 码	操 作 数	注 解
LOOP	ORG	2000H	
	LD	A, 4FH	} 置A口为输入口
	OUT	(82H), A	
	LD	A, 0FH	} 置B口为输出口
	OUT	(83H), A	
	CALL	KEY	调用键扫描子程序
	JR	Z, LOOP-\$	无键, 继续查询
	CALL	D20MS	有键, 延时20ms
	CALL	KEY	再次调用键扫描子程序
	JR	Z, LOOP-\$	无键, 返回
	CALL	KEYTYL	调用键处理程序
	JR	LOOP-\$	返回
	END		

键扫描子程序KEY

标 号	操 作 码	操 作 数	注 解
KEY	XOR	A	} 使所有行为低
	OUT	(81H), A	
	IN	A, (80H)	读入列数据 与FFH比较
	CP	FFH	
	RET		

键分析子程序KEYTYL

标 号	操 作 码	操 作 数	注 解
KEYTYL	LD	B, 8	置循环次数 8
	LD	C, 81H	B口地址送C
	LD	H, FEH	扫描模型初始化
ROW	LD	E, 0	键号初始化
	OUT	(C), H	输出扫描模型
	RLC	H	修改扫描模型
	IN	A, (80H)	读入列数据
	CP	FFH	与FFH比较
	JR	NZ, COLUMN-\$	有键, 转COLUMN
	LD	A, E	} 键号加 8
	ADD	A, 08H	
	DAA		
	LD	E, A	
COLUMN	DJNZ	ROW-\$	转下一行扫描
	RET		
	INC	E	键号加 1
	DAA		
	RRA		将列数据最低位移入CF
	JR	C, COLUMN-\$	CF=1, 检查一下列
	LD	A, E	} 键号送DATA单元
	LD	(DATA), A	
	RET		

例 2. TP801A 单板机键扫描与键分析程序

在TP801A单板机中, 采用 6×5 非编码键盘。LED显示电路的位选择模型锁存器(8CH11), 在键盘接口电路中兼作行扫描模型锁存器。列数据则由地址为90H的三态门接口电路读入CPU。包括LED显示及键盘输入的接口电路如图4.11所示

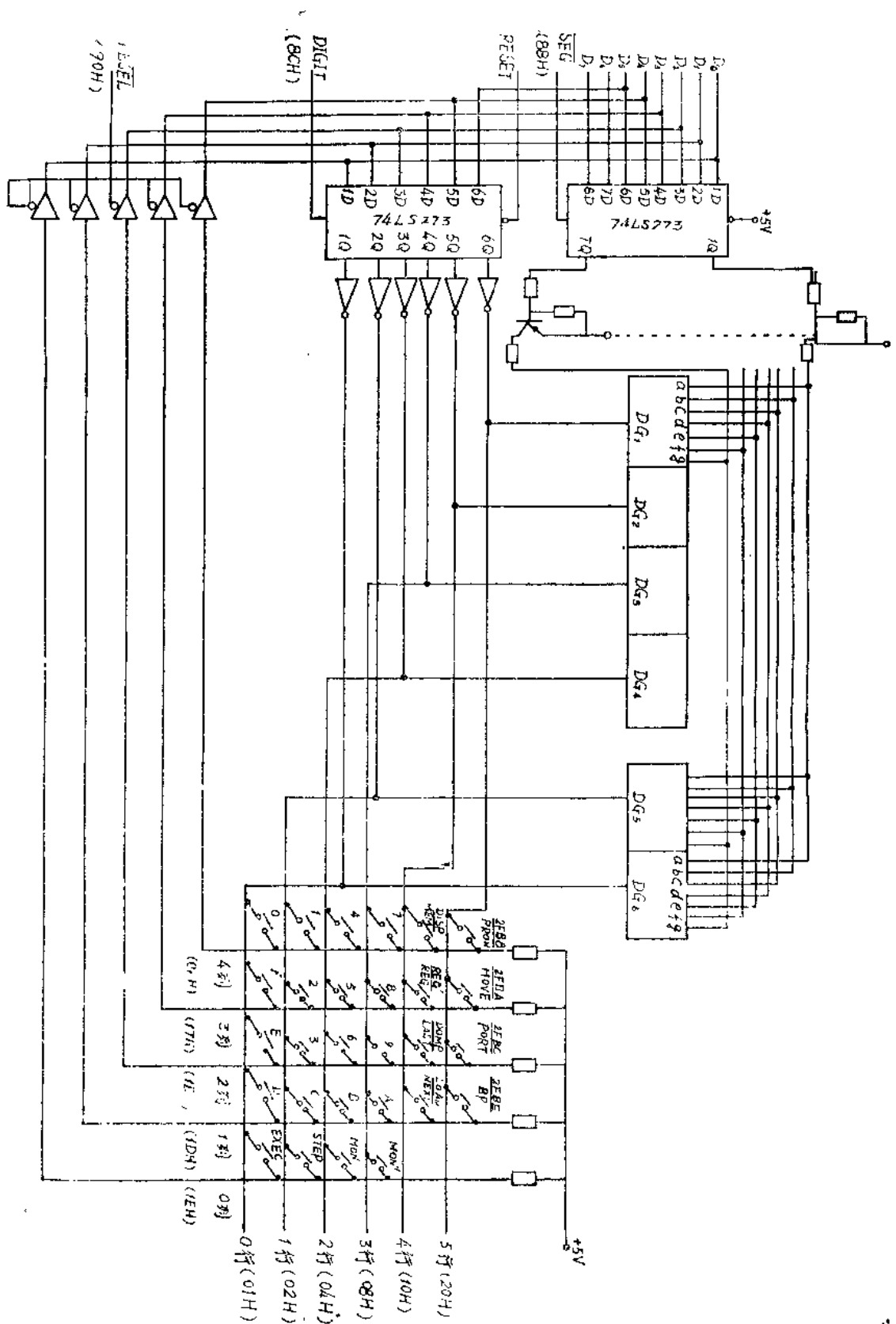


图4.11 TP801A显示与键盘接口电路图

在TP801A单板机的监控程序中，0123H至014EH程序段为键扫描程序段。014FH至021CH程序段为键分析程序段。前者的任务是通过扫描确定是否有按键按下。若无键，则转回00F4H显示程序；若有键，则转入014FH（标号KEYD2）键分析程序。键分析程序段的任务，首先是根据键扫描所得到的行数据及列数据来确定键号。在TP801A单板机中，28个键的键号与键名的对应关系如下：

键 号	00H	01H	02H	03H	04H	05H	06H	07H	08H	09H
键 名	0	1	2	3	4	5	6	7	8	9

0AH	0BH	0CH	0DH	0EH	0FH	10H	11H	12H	13H	14H
A	B	C	D	E	F	EXEO	STEP	MON	MON'	LOAD NEXT

15H	16H	17H	18H	19H	1AH	1BH
DUMP LAST	REG REG	DISP MEM	2FBE BP	2FBC PORT	2FBA MOVE	2FB8 PROM

由上表可知，当键号大于0FH时为功能键。因此共有12个功能键，其中从14H到1BH等8个功能键为双功能键。因此确定键号后，下一部要求区别是数字键还是功能键。若为数字键，则送至显示缓冲区进行显示；若为功能键，则分别转移到各自的键处理程序的入口地址去，以便执行该键所要求的处理程序。

下面要求结合程序清单对TP801A键扫描及键分析程序进行具体分析讨论。

解：首先画出TP801A键扫描程序流程图，如图4.12所示。

由流程图可以看出，TP801A的逐行扫描过程与例3相仿，只是键分析的方法不同。当扫描到某一行列数据不等于1FH时，说明该行有键按下。此时行数据（即扫描模型）存在B寄存器内，列数据（已用1FH屏蔽过）存在A寄存器内。进入键分析程序后要求根据这二个数据来确定键号，这一功能由014FH至0618H程序段来完成。下面对其实现方法进行讨论。

由扫描过程可知，当扫描第0行时B=01H；扫描第1行时B=02H；以此类推，扫描第2~5行时，B分别为04H、08H、10H及20H。

由键阵的连接方式可知，当第0列有键按下时，列数据A=1EH；第1列有键按下时A=1DH；以此类推，第2~4列数据分别为1BH、17H、及0FH。

进一步的问题是如何由行数据及列数据得到28个不同的键号。若采用简单的加法，令键号C=A+B，则键号与上表不相符合且将出现很多相重的键号。例如，1DH+02H=1EH+01H=1FH。为了解决这一问题，可将行数据作适当变换，使二相邻行数据的差为10H。而列数据的最大值与最小值之差为1EH-0FH=10H，这样A与B再相加后就得到28个不同的数码。我们称之为键码。要求行数据变换关系如下表：