

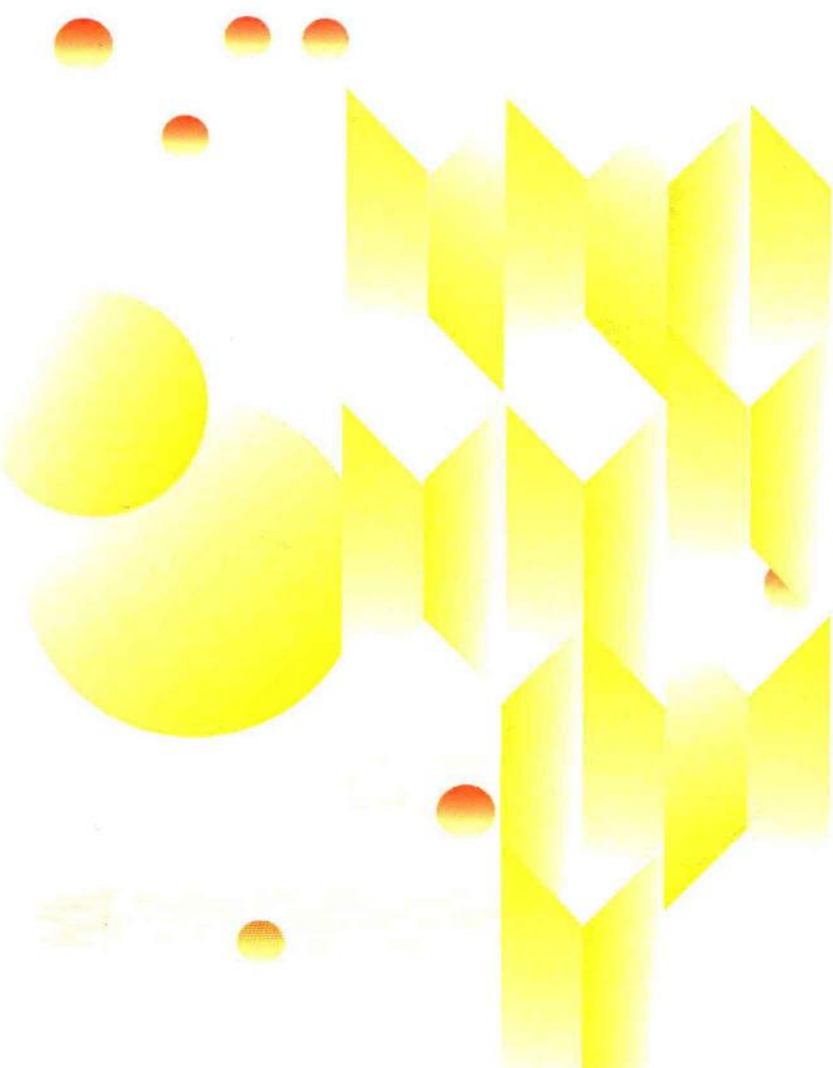


教育部人才培养模式改革和开放教育试点教材

计算机应用专业系列教材

微机接口技术实验指导

江琪 编



中央广播电视大学出版社

计算机应用专业系列教材

微机接口技术实验指导

江 琪 编

中央广播电视大学出版社

图书在版编目(CIP)数据

微机接口技术实验指导/江琪编. —北京: 中央广播电视大学出版社, 2001. 1

ISBN 7-304-01975-1

I. 微… II. 江… III. 微型计算机-接口-实验-高等学校-自学参考资料 IV. TP364.7

中国版本图书馆 CIP 数据核字(2001)第 04282 号

版权所有, 翻印必究。

计算机应用专业系列教材

微机接口技术实验指导

江 琪 编

出版·发行: 中央广播电视大学出版社

电话: 发行部: 010-68519502 总编室: 010-68182524

网址: <http://www.crtvup.com.cn>

地址: 北京市海淀区西四环中路 45 号

邮编: 100039

经销: 新华书店北京发行所

印刷: 北京宏伟双华印刷有限公司 印数: 32501~37500

版本: 2001 年 1 月第 1 版 2004 年 6 月第 4 次印刷

开本: 787×1092 1/16 印张: 5.25 字数: 111 千字

书号: ISBN 7-304-01975-1/TP·146

定价: 10.00 元

(如有缺页或倒装, 本社负责退换)

内 容 简 介

本书为孙琦主编的《微机接口技术》一书的配套教材。本书是根据多种媒体教材一体化总体设计方案的要求编写，全书共安排了11个实验，其中含有2个大型作业，共用27学时。

本书与主教材紧密配合，并利用专门设计的接口实验板完成所有实验，具有通俗、实验简练的特点，可作为计算机专业本、专科生的计算机接口技术实验指导书，也可作为从事计算机硬件开发的工程技术人员参考书。

本书中的实验报告全部集中在附录前，实验报告（一）表示是实验一的实验报告。

编者的话

中央广播电视大学于1999年10月审定通过了高等专科计算机应用专业“微机接口技术”课程的教学大纲，大纲把实践教学分为两部分，一是实验部分，二是大作业部分，两部分共安排27学时。我们依照大纲，对整个实验作如下的安排：人机接口、8255A并行接口芯片安排三次共6学时，8253定时器/计数器芯片安排两次共4学时，8259中断控制器安排一次2学时，一次电子钟大作业4学时；DAC0832数模转换接口芯片安排一次2学时，ADC0809模数转换接口芯片安排一次2学时，一次波形发生器大作业3学时；8250串行接口芯片安排一次两个实验共4学时。

为了使实验能够较顺利地进行，我们建议任课教师在教学时把第七章人机接口的内容提前到第五章之后进行，在完成了前五章及第七章的教学任务后，可安排完成电子钟大作业。另外为了避免在计算机实验室内使用示波器，我们建议先进行模数转换实验，后做数模转换的实验。

完成接口实验就要涉及到实验板，鉴于目前市场上接口实验板价格昂贵，且不能很好地完成我们的实验的要求，我们与北京清大新洋信息技术有限公司合作专门开发了适于本书内容的接口实验板INFOSEA16-1。由于设计的目的明确，使得接口实验板非常适合本书的接口实验，且具有操作简单、体积小、稳定性强、成本低廉等特点。建议读者采用该种实验板。

本教材中使用图标说明：



本部分内容有录像讲解



要求学生复习以前内容



重点、难点

由于时间仓促及作者水平有限，书中谬误及不当之处在所难免，企盼读者批评指正。

作者

2000年10月27日

目 录

第一部分 实验准备	[1]
第一章 汇编语言程序的上机过程	[1]
第二章 DEBUG 调试程序	[3]
第三章 接口实验板工作原理	[7]
3.1 接口卡的工作原理	[7]
3.2 实验板的设计	[9]
第二部分 实验部分	[15]
实验一 “8” 字循环显示实验	[15]
实验二 数码管显示控制实验	[18]
实验三 键 盘 识 别	[20]
实验四 方波显示实验	[22]
实验五 8259A 中断控制器实验	[25]
实验六 时钟程序实验	[27]
实验七 电子钟程序（大作业）	[28]
实验八 ADC0809 模数转换接口实验	[30]
实验九 DAC0832 数模转换接口实验	[32]
实验十 波形发生器（大作业）	[34]
实验十一 串行通信实验	[35]
附录 1 ASCII（美国标准信息交换码）表	[63]
附录 2 DOS 功能调用	[64]
附录 3 BIOS 中断	[69]
附录 4 集成电路引脚图	[74]
参考文献	[77]

第一部分 实验准备

第一章 汇编语言程序的上机过程

汇编语言程序的上机过程与编译语言程序的上机过程相类似，必须经过以下几个步骤：

1. 用 EDIT 文本编辑程序建立和修改以 .ASM 为扩展名的源程序文件。
2. 用汇编程序 MASM 对源程序文件进行汇编，生成以 .OBJ 为扩展名的目标码文件。
3. 用连接程序 LINK 对目标文件进行连接，形成一个可执行的浮动代码文件，文件的扩展名为 .EXE。

经过以上的三个步骤以后，在盘上已经建立了可执行文件，则可在 DOS 的提示符下，直接键入文件名（可不要扩展名），就可把执行文件从盘上装入内存，并立即执行此程序。通常一个较复杂、较长的汇编语言源程序一次通过的可能性很小，这样，就需反复执行以上步骤，对汇编语言源程序进行修改，直至程序运行正确为止。汇编语言上机过程如图 1-1 所示。

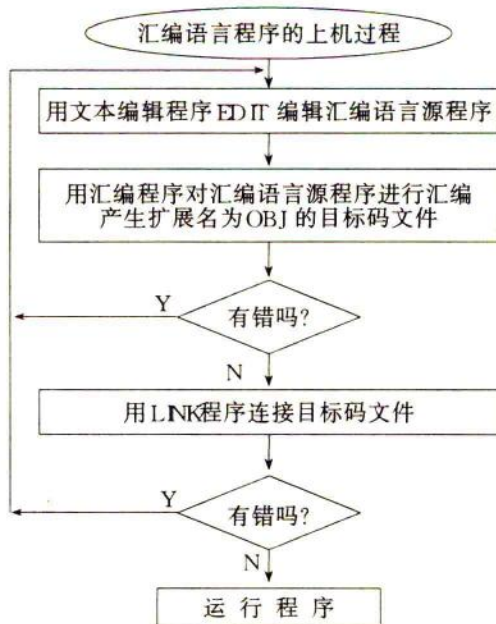


图 1-1 汇编语言上机过程示意图

另外，DOS 还为我们提供了一个用于动态调试程序的 DEBUG 调试程序。用它可以单步执行程序，断点执行程序，可以显示修改 CPU 内部寄存器的内容、标志位的内容以及存

贮器单元的内容等，这样就便于查找程序中的错误了。

可见，要想建立、汇编、连接、调试和运行汇编语言源程序，应该首先建立一张系统盘，在这张盘上除了要具备启动 DOS 所必需的系统文件之外，至少还需有如下文件：

```
edit.com  
masm.exe  
link.exe  
debug.com
```


第二章 DEBUG 调试程序

大部分的程序必须经过调试才能纠正程序设计中的错误，从而得到正确的结果。DEBUG 是专为汇编语言设计的一种调试工具，是汇编语言程序员必须掌握的调试手段。启动 DEBUG 程序的过程如下：

```
C>DEBUG <CR>
```

DEBUG 程序调入内存后出现提示符“-”，以后便可以使用 DEBUG 的所有命令了。

1. 显示内存单元内容命令 D

格式为：-D 地址

或

-D 范围

例如按指定范围显示存储单元内容的方法为：

```
-D 100 126
```

```
18E4:0100  E9 84 3D 43 6F 6E 76 65 - 72 74 65 64 00 00 00 00  .. = Converted....
```

```
18E4:0110  4D 5A 6F 01 1F 00 03 00 - 20 00 00 00 FF FF 62 03  MZo.....b.
```

```
18E4:0120  6A 01 81 5E 00 01 28                                     j..^..(
```

其中 0100H 至 0126H 是 DEBUG 显示的地址范围。左边用十六进制表示每个字节，第 8、9 字节间有一连接字符“-”；右边用 ASCII 码字符表示每个字节，“.”表示不可显示字符。这里没有指定段地址，但 D 命令会自动显示 DS 段的内容。如果 D 命令后面只给出一个地址，则显示从此地址开始的 80 个字节的内容。如果没有指定地址，则接着上一个 D 命令显示以后的 80 个字节的内容。

2. 修改存储单元内容的命令 E

它有两种基本格式：

(1) 用命令给定的内容表去代替指定范围的内存单元的内容，命令格式为：

```
-E 地址 内容表
```

例如：

```
-E DS: 120 00 11 'ABC'22
```

内存单元 DS: 120 到 DS: 125 这 6 个单元的内容由表中给定的 6 个字节内容（其中 3 个字节用 16 进制数表示，即 00，11，22；另三个用字符表示，则把它们的 ASCII 码值代入）所代替。

(2) 逐个单元相继修改，命令格式如下：

```
-E 地址
```

例如：

```
-E 100
```

在输入了上述命令后，屏幕上显示 100H 单元的地址和原有的内容：

18E4: 0100 12. _

如果需要把该单元的内容修改为 78, 则可以直接输入 78, 然后再按空格键, 则完成修改, 并显示下一个单元的地址和原有的内容。显示如下:

18E4: 0100 12.78 1B. _

这样, 可以不断修改相继单元的内容直到用回车键结束该命令为止。

3. 检查和修改寄存器的命令 R

它有三种命令格式:

(1) 显示 CPU 内所有寄存器内容和标志位状态, 其格式为:

- R

显示格式如下:

AX = 0000 BX = 0000 CX = 0000 DX = 0000 SP = FFEE BP = 0000 SI = 0000

DI = 0000

DS = 3F65 ES = 3F65 SS = 3F65 CS = 3F65 IP = 0100 NV UP EI PL NZ NA PO NC

3F65: 0100 3E

DS:

3F65: 0101 B702

MOV BH, 02

(2) 显示和修改某个寄存器内容, 其格式为:

- R 寄存器名

例如, 输入

- R AX

系统将显示如下:

AX 0000

:

即 AX 寄存器的当前内容为 0000H, 如不修改其内容则按回车键, 否则, 可输入欲修改的内容。

(3) 显示和修改标志位状态

其格式为:

- RF

系统将显示如下:

NV UP EI PL NZ NA PO NC -

此时, 如不修改其内容可按回车键, 否则, 可键入欲修改的一个或多个标志位内容。

例如:

NV UP EI PL NZ NA PO NC - CY DI NG ZR <CR>

可见, 键入的顺序可以是任意的。

对 8 个标志位的操作如表 1-1 所示。

表 1-1 标志位符号表

标志位名	置 位	复 位
溢出 Overflow	OV	NV
方向 Direction	DN	UP
中断 Interrupt	EI	DI
符号 Sign	NG	PL
零 Zero	ZR	NZ
辅助进位 Auxiliary Carry	AC	NA
奇偶 Parity	PE	PO
进位 Carry	CY	NC

4. 运行命令 G

为了检查程序运行是否正确，希望在运行中能设置断点以便逐段调试程序。

G 命令的格式为：

G [= 地址 1] [地址 2 [地址 3...]]

其中地址 1 指定运行的起始地址，如省略，则从当前的 CS: IP 处开始运行。地址 2，地址 3……均为断点地址，DEBUG 程序最多允许设置 10 个断点。设置断点的工作原理是：它用中断类型 3 指令（操作码 CCH）来代替被调试程序在断点地址处的指令操作码，当程序执行到断点时，就停止执行并显示当前所有寄存器及标志位的内容和下一条将要执行的指令，且全部断点被取消，断点处的指令码被恢复（若程序未遇到断点，则不恢复）。

注意，在命令格式中，各地址之间用空格隔开。

5. 跟踪命令 T

它有两种命令格式：

(1) 逐条指令跟踪，其格式如下：

- T [= 地址]

程序执行一条指定地址处的指令后停下来，显示 CPU 当前所有寄存器及标志位的值以及下一条指令的地址及内容。若在指令中没有指定地址，则从 CS: IP 的现行地址处单步执行指令。显然反复执行 T 命令便可起到单步执行指令的目的。

(2) 多条指令跟踪命令，其命令格式如下：

- T [= 地址] [值]

此命令从指定的地址开始，执行由命令中的值所决定的若干条指令，然后停下来显示 CPU 当前所有寄存器的内容及全部标志位的值。

6. 汇编命令 A

若在调试程序时发现程序中的某一部分要改写，或要增补部分指令时，则可用 A 命令及其它一些命令来输入，汇编，运行，调试这一段程序。另外，调试简单的程序甚至可以直接利用 A 命令来完成。A 命令的格式为：

- A [地址]

它是从指定的地址（缺省时从 CS: IP 处）开始输入汇编语言的语句。A 命令把它们汇编成机器代码相继地存放到从指定地址开始的存贮区中。若输入的语句有语法错误，则 DEBUG 显示：

^ERROR

且重新显示现行的汇编地址等待新的输入。程序输入完毕后，最后一行不输入内容，直接按回车即可。DEBUG 支持 8086 汇编语言语法和 MASM 宏汇编语言。但是程序所支持的语法稍有不同。简述如下：

- ①所有输入的数字值全为 16 进制数，每个数字后面不能使用 H。
- ②前缀助记符必须在相关的指令之前输入。
- ③支持两个最常用的 DB, DW 伪指令，用来把字和字节的值直接送入相应的存贮单元。

7. 反汇编命令 U

若内存某一区域中，已经有了某一个程序的目标码，为了能清楚了解此程序的内容，就希望能把目标程序反汇编为源程序，这就需要用到 U 命令。命令格式如下：

- U [范围]

在这里范围是可选项，若没有此项，则以上一条 U 命令的最后一条地址的下一单元作为起始地址开始反汇编 32 个字节。若没有使用过 U 命令，则从当前段的 100H 地址单元开始反汇编 32 个字节。若范围可选项中仅有一个地址，则从这个地址开始反汇编 32 个字节。若范围可选项中既有起始地址又有结束地址则反汇编此范围中的目标码。

8. 命名命令 N

不管是执行读盘命令 R 和写盘命令 W，一般都先要给文件命名，命令格式如下：

- N 文件名

这里，文件名可以有驱动器名和扩展名。例如：

- N A: EXAM.COM

9. 读盘命令 L

L 命令完成把指定文件读入到内存中的操作。若命令中规定了地址，则读入到指定区域中。否则，读入到 CS: 0100 开始的内存区域中。但是若具有扩展名 .COM 的文件，则始终是读入到 CS: 0100 开始的区域中，而把具有 .EXE 扩展名的文件读入到 CS: 0000 开始的区域中，命令格式如下：

- L [地址]

在 BX 和 CX 中包含着读入文件的字节数，其中 BX 中放的是高位数，CX 中放的是低位数。

10. 写盘命令 W

W 命令完成将内存区域中的程序和数据存盘的操作。命令格式如下：

- W [地址]

在 W 命令执行之前，必须由 N 命令指定文件名，由 BX 和 CX 指定要写入文件的字节数，其中 BX 中放的是高位数，CX 中放的是低位数。命令中若没有指定地址，则内存从 CS: 100H 开始，否则从指定地址开始。

11. 退出命令 Q

当把程序调试完毕后，使用 Q 命令退出 DEBUG，而返回到 DOS。

命令格式如下：

- Q

但是 Q 命令并不执行存盘操作。故退出前，如需存盘的话，必须先使用 W 命令存盘后再退出。



第三章 接口实验板工作原理

完成接口实验需要使用接口实验板，为此我们首先进行接口实验板的设计。可能你所在的学校根本没有接口实验板，那么可以按照我们的设计自己完成接口实验板的制作，或者与我们联系购买事宜*。本书所有的实验均假设使用我们设计的实验板。

我们先对接口实验板的硬件工作原理做一个简单介绍。

 接口实验板由接口卡和实验板这两大部分组成，见图 1-2。其中，接口卡插在 PC 机的插槽上，由地址译码器电路、数据总线缓冲器和时钟发生器等电路组成，它完成 PC 机与实验板的硬件连接。实验板由 8253、8255A、ADC0809、DAC0832 等芯片组成，通过 8255A 芯片还扩展了 8 位 LED 显示器和 8 只按键。另外，实验板上还留出了大量的插座，以便于完成中断及其它实验。

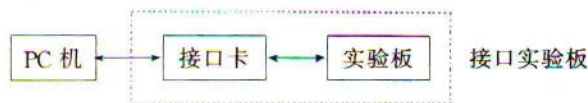


图 1-2 接口实验板工作原理图

3.1 接口卡的工作原理

接口卡用于完成实验板与 PC 机的连接，它被插于 PC 机的 PC 总线插槽上，它与实验板之间用 26 芯的扁平电缆连接。接口卡的工作原理如图 1-3 所示。

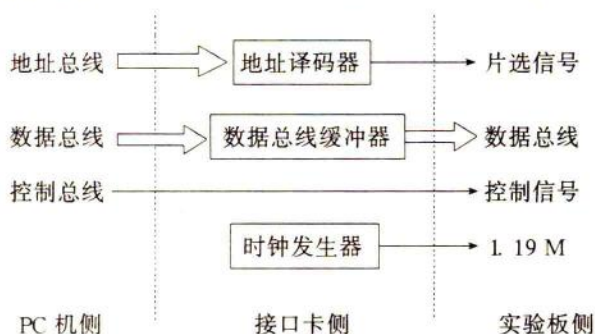


图 1-3 接口卡工作原理图

一、地址译码器电路

实验板上各芯片的端口地址均在地址 300H~31FH 范围内。由于实验板上的各个可编

* 实验板的购买事宜可与徐汝鑫联系（免费长途电话为 8008100918）。

程的接口芯片均需要片选信号，故我们设计了4个片选信号供实验板选用。这部分的电路原理如图1-4所示。

由图1-4可见，接口卡提供了 $\overline{Y_0}$ 、 $\overline{Y_1}$ 、 $\overline{Y_2}$ 和 $\overline{Y_3}$ 这4个片选信号，这4个片选信号经过与门后，再与8086CPU的 A_2 引脚过或门后，接往74LS245的 $\overline{E}$ 端，这样保证了只有当 $\overline{Y_0}$ 、 $\overline{Y_1}$ 、 $\overline{Y_2}$ 和 $\overline{Y_3}$ 中的一个有效且CPU的 A_2 引脚为低电平时，双向缓冲驱动器74LS245才能正常工作。 A_9 、 A_8 、 $\dots$ 、 A_3 和 $\overline{IOR}$ 、 $\overline{IOW}$ 、 AEN 与译码器输出端 $\overline{Y_0}$ 、 $\overline{Y_1}$ 、 $\overline{Y_2}$ 和 $\overline{Y_3}$ 的对应关系如表1-2所示。

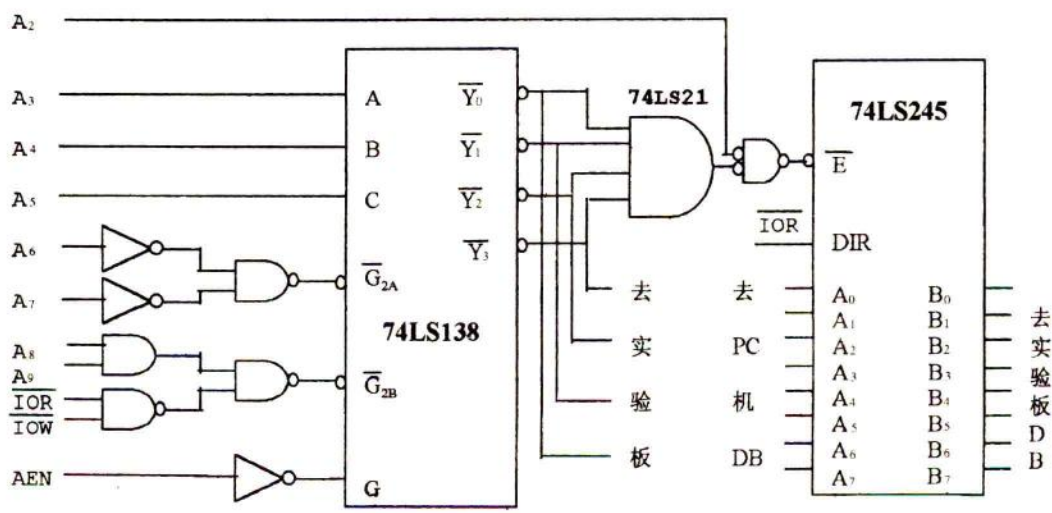


图1-4 接口卡原理图

表1-2 接口卡逻辑关系表

AEN	$\overline{IOR} \wedge \overline{IOW}$	A_9	A_8	A_7	A_6	A_5	A_4	A_3	译码输出端
0	0	1	1	0	0	0	0	0	$\overline{Y_0}$
0	0	1	1	0	0	0	0	1	$\overline{Y_1}$
0	0	1	1	0	0	0	1	0	$\overline{Y_2}$
0	0	1	1	0	0	0	1	1	$\overline{Y_3}$
1	×	×	×	×	×	×	×	×	无输出
×	1	×	×	×	×	×	×	×	无输出

由表1-1可见：

- $\overline{Y_0}$ 的地址范围是： 300H~307H
- $\overline{Y_1}$ 的地址范围是： 308H~30FH
- $\overline{Y_2}$ 的地址范围是： 310H~317H
- $\overline{Y_3}$ 的地址范围是： 318H~31FH

 如果考虑到只有当CPU的 A_2 引脚为低电平时，74LS245才能工作，则实际上片选信号的地址范围是：

- $\overline{Y_0}$ 的地址范围是： 300H~303H

$\overline{Y}_1$ 的地址范围是: 308H~30BH

$\overline{Y}_2$ 的地址范围是: 310H~313H

$\overline{Y}_3$ 的地址范围是: 318H~31BH

二、总线驱动器

74LS245 芯片用作系统机与实验板之间双向总线缓冲驱动器。其中的 A 端接往 PC 机的数据总线, B 端接往实验板。方向控制引脚 DIR 接往 PC 机的 $\overline{IOR}$, 当 $\overline{E}$ 端有效且 $\overline{IOR}$ 有效时, 74LS245 完成 B 端向 A 端的数据传送; 否则, 当 $\overline{E}$ 端有效 ($\overline{IOR}$ 与 $\overline{IOW}$ 一定有一个有效) 及 $\overline{IOR}$ 无效 (意味着此时 $\overline{IOW}$ 有效) 时, 74LS245 完成 A 端向 B 端的数据传送。

三、CLK 时钟发生电路 1.19M

在接口卡上有一个 CLK 时钟发生电路, 它用于产生实验板上需要的 1.19M 的方波。在电路组成上, 我们使用 8284 为主芯片, 在 8284 的 X_1 , X_2 端接有一个 14.318M 的晶体, 它的 6 分频输出端 PCLK 输出 2.386MHz 的方波, 再经过 74LS175 的 2 分频最后产生 1.19MHz 的方波供给实验板上的 8253A 芯片使用。

3.2 实验板的设计

一、实验板上的 8253 芯片设计

8253 定时计数器芯片共有 24 条引脚。在实验板上, 它的 8 根数据线接往接口卡上 74LS245 的 B 端。 $\overline{RD}$ 和 $\overline{WR}$ 分别接往 PC 机的 $\overline{IOR}$ 和 $\overline{IOW}$ 。8253 三个通道的 CLK 引脚均接至接口卡的 1.19MHz 频率端, 而 8253 三个通道的 OUT 输出引脚则都被悬空, 另分别被接至实验板插子 J_4 的第 3, 4, 5 脚, 由学生自己连线使用。对于 GATE 引脚的处理如图 1-5 所示:

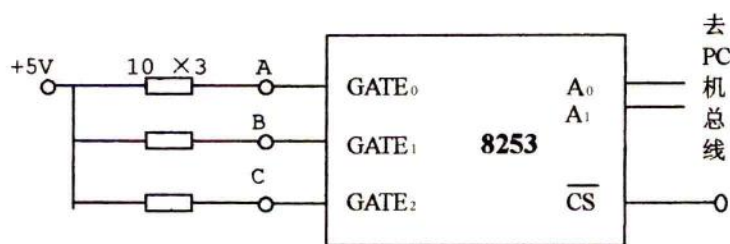


图 1-5 8253 GATE 引脚处理图

可见 $GATE_0$ 、 $GATE_1$ 和 $GATE_2$ 均接往高电平, 即全部打开计数/定时器。考虑到有些实验的实际需要, 我们分别从这三个引脚引出了插孔可供使用。如图中所给出的 A, B, C 点。这样, 如需要把 $GATE_0$ 由其它电路控制, 可将控制引线接往插孔 A。在实验板上, 8253 的 $\overline{CS}$ 引脚悬空, 实验时可用连线由 $\overline{Y}_0$, $\overline{Y}_1$, $\overline{Y}_2$ 或 $\overline{Y}_3$ 接来。

对于 8253 的 A_1 、 A_0 这两个引脚, 我们把它们直接接往 PC 机 ISA 总线的 A_1 、 A_0 引脚, 这时 8253 的各端口地址如表 1-3 所示 (假设 $\overline{CS}$ 引脚接往 $\overline{Y}_1$)。

表 1-3 8253 各端口地址表 (1)

A ₉	A ₈	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀	端口地址	端口地址
1	1	0	0	0	0	1	0	0	0	308H	计数器 0
1	1	0	0	0	0	1	0	0	1	309H	计数器 1
1	1	0	0	0	0	1	0	1	0	30AH	计数器 2
1	1	0	0	0	0	1	0	1	1	30BH	控制端口

当 $\overline{\text{CS}}$ 引脚接往片选信号 $\overline{\text{Y}}_0$ 时, 8253 的各端口地址如表 1-4 所示。

表 1-4 8253 各端口地址表 (2)

A ₉	A ₈	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀	端口地址	端口地址
1	1	0	0	0	0	0	0	0	0	300H	计数器 0
1	1	0	0	0	0	0	0	0	1	301H	计数器 1
1	1	0	0	0	0	0	0	1	0	302H	计数器 2
1	1	0	0	0	0	0	0	1	1	303H	控制端口

二、8255A 芯片及其外设

 8255A 具有三个可编程的并行口。这三个口可分别安排输入和输出等几种工作方式。在实验板的设计上, 我们安排 PA, PB 为输出口, 用来控制 8 位七段显示器 LED 的显示; 安排 PC 口为输入口用来监视按键的工作, 而上面这种安排是靠软件来支持的 (由学生自己编制), 也可以由学生改变这种安排 (通过拔下实验板上的部分短路子来完成)。下面分别介绍:

1.8255A 的端口地址

8255A 的 $\overline{\text{CS}}$ 信号悬空, A₁, A₀ 直接接往了 PC 机 ISA 总线的 A₁、A₀ 引脚。这样当把 8255A 的 $\overline{\text{CS}}$ 引脚接到 $\overline{\text{Y}}_2$ 时, 8255A 各端口地址如表 1-5 所示。

表 1-5 8253A 端口地址表

A ₉	A ₈	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀	端口地址	端口名称
1	1	0	0	0	1	0	0	0	0	310H	端口 A
1	1	0	0	0	1	0	0	0	1	311H	端口 B
1	1	0	0	0	1	0	0	1	0	312H	端口 C
1	1	0	0	0	1	0	0	1	1	313H	控制端口

2.LED 显示电路

实验板上采用 8 个共阴极的 LED, 它的各段选位均连接在一起由 PA 口来控制。考虑到驱动问题, 不是将 PA 口的输出直接接往段选端, 而是经过了两片 74LS06 芯片 (集电极开路非门) 以后接往段选端的。如图 1-6 所示。

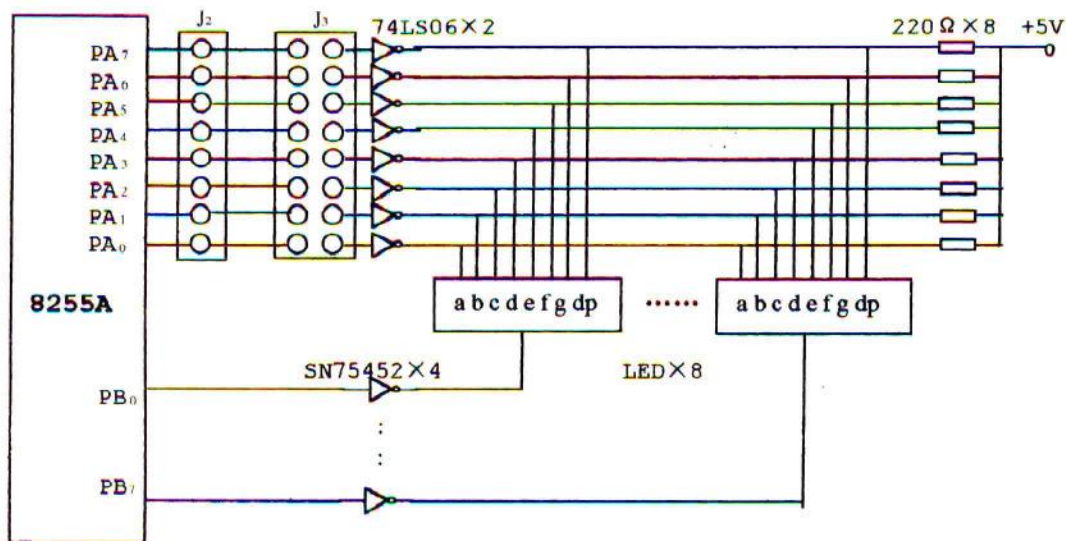


图 1-6 LED 连接原理图

现若想要某数码管显示如图 1-7 所示的数字 5，则由 PA 口应送出 92H，即 10010010B，经过非门后变为：

dp g f e d c b a

0 1 1 0 1 1 0 1

这样可得到数字 0~F 的字段码值，见表 1-6。

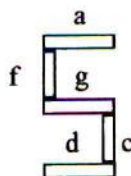


图 1-7 数字 5 的字体图

表 1-6 LED 字段码表

字型	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
码值	c0	f9	A4	b0	99	92	82	f8	80	98	88	83	c6	a1	86	8e

图 1-6 中的 J₃ 为八个短路块。当插上这八个短路块时，PA₇~PA₀ 就被接往段选端。当拨下这八个短路块时，则 PA₇~PA₀ 就处于悬空状态，学生可通过插孔排 J₂ 而使用 PA 口。

8255A 的 PB 口用来控制 LED 的位选端。同样考虑到驱动问题，PB₇~PB₀ 也不是直接接到各个数码管的位选端。而是经过了 SN75452 驱动电路（非门）。在这里，我们也同样设计了如上述的 J₂ 和 J₃，当拨下这八块短路块时，PB 口悬空，如果插上 J₃，则 PB₀ 选择最高位数码管，PB₁ 选择次高位数码管，……，PB₇ 选择最低位数码管。

3. 键盘电路

在 PC 口的设计上我们同样使用了 J₂, J₃。如果插上 J₃，则 PC 口的八个引脚就连到了八个按键。我们安排 PC₇ 连到键 7，PC₆ 连到键 6，……，PC₀ 连到键 0。这样当无键按下时，PC₇~PC₀ 均为高电平输入；当有键按下时，则相应位为低电平输入，其余位为高电平输入。工作原理见图 1-8。