

《现代电子技术》增刊

C语言基本编程教程

C 语言基本编程教程

梅宏斌 编
阎明印

梅宏斌 阎明印 编

TP312
社

陕西电子杂志社

TP312
M470

392266

微机数据采集与处理

C 语言基本编程教程

梅宏斌 阎明印 编

JS176/11
内容简介

以 IBM PC 系列微机为中心,加上适当的 A/D—D/A 接口板及相应的应用软件构成的微机数据采集与处理系统在故障监测与诊断、工业自动控制、生产过程监控等各个领域得到了极为广泛的应用。本书以目前国内外比较流行的 C 语言作为编程语言,比较系统地讨论了微机数据采集与处理系统软件开发中牵涉到的几个方面的内容:数据采集、数据处理和用户界面。其中数据采集中牵涉到的定时、中断、DMA 等技术的编程全部由 C 完成,同传统的汇编语言相比,代码简单、清晰,便于理解与掌握。数据处理部分包括如何用幅域参数指标进行状态监视、如何用 FFT 进行频谱分析、包络分析及如何根据模拟低通滤波器设计数字低、带、高通滤波器。用户界面包括汉化技术、各种菜单设计技术及串行通讯等内容。本书既不同于一本程序集,因为它对数据采集与处理方面的基本概念与内容作了比较系统、必要的论述;又不同于一本数据采集与处理的教科书,因为它包含了一些实用性很强的程序。

本书适用于大专院校、科研院所和工厂中从事 C 语言软件开发、微机接口及应用和数据采集与处理及其相关领域工作的广大科技人员阅读和参考。

前 言

数据采集与处理是计算机应用的一个重要分支,是获取信息的主要与最新手段之一。它广泛应用于智能仪器仪表、故障监测与诊断、工业自动控制、生产过程监控等各个领域。顾名思义,数据采集与处理研究数据的“采集”和“处理”两个方面的内容,采集主要研究计算机如何将外部客观世界存在的各种模拟物理量(如温度、压力、振动等)转换为适合于计算机处理的数字量(这称为模数转换—A/D)以便对这些物理量进行精确的测量与监视,以及如何将计算机处理过的数字量转换为模拟电量(这称为数模转换—D/A)以便控制外部对象,因此,它牵涉到计算机与外部的接口问题;而处理主要研究的是计算机内部对数据进行各种数学上、逻辑上的运算、分析和加工,从而对外部对象的状态有一个准确的认识,并为进行各种控制作准备。由此可以看出,计算机数据采集与处理既要考虑计算机与外部的接口(主要是A/D—D/A接口),又要考虑计算机内部的数据处理,所以在软件编程时要兼顾这两方面的要求。

在我国,目前IBM PC系列微机及其兼容机仍是通用微机的主流产品,以这种微机为中心加上合适的A/D—D/A接口板,并开发相应的软件构成的微机数据采集与处理系统得到了极为广泛的应用并取得了良好的成效。另一方面,目前国内外流行的C语言同时具有较强的硬件接口能力和数学运算能力,所以既可以取代传统的汇编语言用于A/D—D/A接口编程,又可以用于数据处理的编程,用它作为编程语言开发数据采集与处理系统应用软件是一种明智的选择。现在,讲解C语言及其编程的书很多,讲解IBM PC系列微机接口的书也不少,数据采集与处理方面的书也有一些,但讨论C语言与PC系列机接口实现数据采集与处理的书几乎见不到。编者根据近几年来从事这方面研究开发工作的经验和体会编写这本书,期望对从事微机C语言数据采集与处理的读者有所启发和帮助。

本书共分九章。第一章概要介绍了数据采集与处理、微机接口及C语言的一些基本知识;第二章和第三章讨论了A/D—D/A接口原理及编程方法,其中包括A/D采样时如何准确定时、中断技术应用、DMA高速数据采集等内容;第四章、第五章和第六章讨论数据处理的基本编程问题,其中第四章介绍利用数据的幅域参数指标实现状态监视的编程方法,第五章讨论了快速富立叶变换(FFT)的算法及其频谱分析的编程方法,第六章讨论了如何根据模拟低通滤波器设计数字低、带、高通滤波器从而实现数字滤波;第七章讨论数据采集与处理系统应用软件的用户界面设计问题;第八章介绍如何用C语言控制一些常用的外设,主要讨论了屏幕打印,TAGA卡编程,鼠标控制及串行通讯等。

由于编者水平所限,谬误之处在所难免,敬请广大读者批评指正,编者不胜感激。

编 者

一九九四年九月

目 录

前言

第一章 概述	(1)
1.1 微机数据采集与处理系统	(1)
1.2 微机接口初步	(2)
1.2.1 接口基本概念	(3)
1.2.2 接口寻址方法	(3)
1.2.3 可编程并行接口 8255	(5)
1.2.4 可编程计数器/定时器 8253	(7)
1.2.5 可编程中断控制器 8259	(10)
1.2.6 DMA 方式及 DMA 控制器 8237	(14)
1.3 C 语言在微机数据采集与处理中的应用概述	(16)
1.3.1 C 语言直接端口操作函数	(17)
1.3.2 中断处理	(17)
1.3.3 直接内存操作函数	(19)
1.3.4 文件操作	(19)
第二章 A/D—D/A 接口基本原理	(21)
2.1 基本概念	(21)
2.2 D/A 转换的基本原理	(22)
2.2.1 D/A 转换器工作原理	(23)
2.2.2 D/A 转换器输出	(24)
2.3 A/D 转换的基本原理	(25)
2.3.1 多路模拟开关	(25)
2.3.2 采样保持(S—H)电路	(25)
2.3.3 A/D 转换器(ADC)	(25)
2.4 与 IBM PC 兼容的板级数据采集产品	(27)
2.4.1 MS1215 A/D—D/A 接口板	(27)
2.4.2 AX5412 高速数据采集板	(29)
第三章 A/D—D/A 接口编程	(35)
3.1 用纯软件方式进行数据采集	(35)
3.1.1 A/D 转换的启动方式和管理方式	(35)
3.1.2 软件启动、查询管理的数据采集	(36)
3.2 利用中断进行数据采集	(44)
3.2.1 数据采集的定时及中断的应用	(44)
3.2.2 定时中断启动、查询管理方式进行数据采集	(45)
3.2.3 定时启动、中断管理方式进行数据采集	(46)

3.2.4 应用举例	(48)
3.3 用 DMA 方式进行数据采集	(50)
3.3.1 概述	(50)
3.3.2 DMA 的初始化编程	(51)
3.3.3 定时启动、DMA 传送、中断管理进行数据采集	(53)
3.4 本章小结	(54)
第四章 数据的幅域处理	(55)
4.1 数据存贮	(55)
4.2 幅域参数指标	(60)
4.2.1 有量纲指标	(60)
4.2.2 无量纲参数指标	(60)
4.3 直方图	(61)
4.4 Turbo C 语言程序	(61)
第五章 频谱分析	(69)
5.1 频谱分析在数据处理中的作用	(69)
5.2 快速富立叶变换原理及程序	(69)
5.2.1 基 2 时间抽选算法原理	(69)
5.2.2 时间抽选过程的流图表示	(71)
5.2.3 数据重排与二进制反序	(71)
5.2.4 基 2 时间抽选 FFT 算法 Turbo C 源程序	(73)
5.3 计算功率谱的 Turbo C 程序	(77)
第六章 数字滤波和检波	(83)
6.1 数字滤波的基本原理	(83)
6.2 从模拟低通滤波器设计数字低、高、带通滤波器	(84)
6.2.1 模拟低通巴特沃思滤波器设计	(84)
6.2.2 从模拟低通滤波器设计数字低通滤波器	(84)
6.2.3 从模拟低通滤波器设计数字高通滤波器	(89)
6.2.4 从模拟低通滤波器设计数字带通滤波器	(90)
6.3 低、高、带通数字滤波的 Turbo C 程序	(92)
6.3.1 从输入数据和滤波系数求输出数据	(92)
6.3.2 滤波程序	(93)
6.3.3 小结	(97)
6.4 数字包络检波	(98)
第七章 用户界面	(100)
7.1 文本窗口及菜单技术	(100)
7.1.1 文本窗口边框	(100)
7.1.2 弹出窗口	(101)
7.1.3 文本光条技术	(102)

7.1.4 文本窗口及菜单举例—文本编辑器	(104)
7.2 图形窗口及菜单技术	(110)
7.2.1 图形窗口操作	(110)
7.2.2 图形光条技术	(111)
7.2.3 图形状态下输出数据及字符	(113)
7.3 西文 DOS 下的汉字显示	(114)
7.3.1 汉字显示的基本原理	(115)
7.3.2 汉字显示程序	(116)
7.3.3 放大或缩小显示汉字程序	(117)
7.3.4 带小字库的汉字显示程序	(118)
7.3.5 各种字体多点阵大汉字显示	(121)
第八章 外设控制与通讯	(125)
8.1 中断技术	(125)
8.1.1 中断服务程序设计	(125)
8.1.2 内存驻留程序设计	(127)
8.2 屏幕图形的打印机拷贝	(129)
8.2.1 引言	(129)
8.2.2 图形打印基本原理	(130)
8.2.3 图形硬拷贝程序	(131)
8.3 鼠标控制程序设计	(134)
8.4 TVGA 编程初步	(137)
8.5 串行通讯	(139)
8.5.1 数据的异步串行发送和接收	(140)
8.5.2 RS-232 标准	(140)
8.5.3 通过 BIOS 调用进行串行通讯	(141)
8.5.4 串行通讯举例—计算机之间传送文件	(144)

第一章 概述

1.1 微机数据采集与处理系统

数据采集与处理是计算机应用的一个重要分支,是获取信息的主要与最新手段之一。它研究信息数据的采集、存贮、处理及控制等内容。在智能仪器仪表、故障检测与诊断、工业自动控制、生产过程监控等领域,都存在着模拟量的测量和控制问题。为了对温度、压力、流量、速度、位移等等物理量进行测量和控制,都是通过传感器把上述物理量转换成能模拟物理量的电信号,即模拟电信号。将模拟电信号经过处理并转换成计算机能识别的数字量,送进计算机,这就是数据采集。计算机将采集来的数字量根据需要进行不同的判别、分析、运算,得出所需要的结果,这就是数据处理。数据处理结果显示于显示器屏幕或由打印机打印在纸上,以便对某些物理量进行监视,再将数据处理结果的数字量转换成模拟信号去控制某些物理量,这就是监控。由上述可以看出,数据采集的核心实际上是模拟量到数字量的转换(简称 A/D 转换),它牵涉到硬件电路和计算机软件编程两部分内容;数据处理主要是根据被测量或被控制对象的需要,对采集到的数据进行多种数学上的运算或分析,这主要牵涉到软件编程问题;第三步监控中的“监”实际上是显示与输出数据处理结果的问题,而“控”实际上是一个数字量到模拟量的转换问题,这也涉及到硬件电路和软件编程问题。因此,具体说来,数据采集与处理技术研究的主要是模数转换(A/D)、数据处理和数模转换(D/A)三个方面的问题。其中 A/D、D/A 负责的是计算机同外界的联系问题,而数据处理是计算机内部的问题。

作为一种获取信息的主要手段,数据采集与处理在工程各个领域获得了极为广泛的应用。就拿设备工况监视与故障诊断来说吧。设备在工作过程中状态的变化往往有许多物理量来反映,如温度、振动、噪声等,测量这些物理量时必须先用各种传感器将这些量转变成模拟电量,再由数据采集器将这些模拟电量转换成时间和数值上都离散且量化的数字量,然后送入计算机,由计算机对这些量进行加工、处理,判断设备工况是否正常。如不正常,再采集更多的信息并进行更进一步的分析处理以判别出现了什么类型的故障,以及故障的部位、原因、趋势等。如果有可能,根据数据处理结果,由计算机通过 D/A 转换发出模拟控制量控制设备的某一参数(如载荷、速度等),从而抑制故障发展或使设备由异常转入正常。由此可以看出,离开了数据采集与处理,故障诊断简直无法进行。

70 年代以来随着微机技术突飞猛进的发展和微机在工程领域的普遍应用,以微机为核心的可编程的数据采集及处理系统迅速发展起来。特别是适合于通用微机(如 IBM PC 系列)使用的板级数据采集产品大量涌现,使得通用微机的应用从原有的科学计算领域迅速扩展到信息处理及实时控制。现时的通用微机,只要在其扩展插槽内插上一块数据采集板,并开发一套应用软件,就可以很方便地组成一微机数据采集与处理系统用于各种信息处理及实时控制目的。

一般说来,微机数据采集与处理系统应具有如下功能:

- (1) 程序控制多路检测与控制;
- (2) 程序控制利用 DMA 方式实现高速数据采集;
- (3) 存储器存贮和以数据文件形式磁盘永久存贮大量采集到的数据;
- (4) 具有常规的幅域、频域等数据处理功能;根据特别用途,应有一些特别的数据处理功能;
- (5) 一般说来,应具有对被测物理量的在线实时监测及对被控物理量的在线实时控制功能;
- (6) 具有显示、打印、报警(如声响报警和红黄色方块闪烁显示报警)和绘图等功能。

微机数据采集与处理系统由硬件和软件两部分组成。其中硬件的组成如图 1.1 所示,主要由被测控对象(设备)、A/D 和 D/A 通道、开关量输入输出通道、微机、人机联系设备(显示器、键盘、打印机等)等组成。复杂一些的系统根据需要还可使用多台同类型或不同类型的微机组成的多机系统(如一台通用 IBM PC 系列微机加数台单片机组成的多机系统,以及数台 IBM PC 微机构成的多机系统等)。

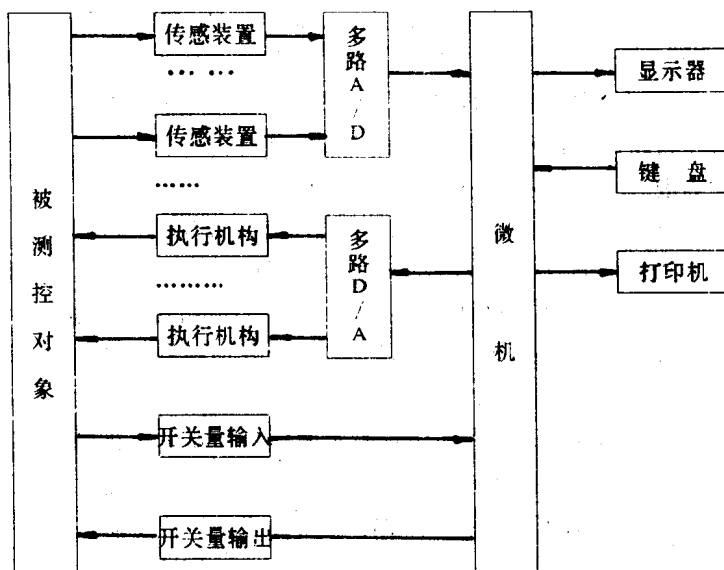


图 1.1 微机数据采集与处理系统硬件组成

软件一般是指用户根据实际需要开发的应用软件。由于微机既要与被控对象通过 D/A、A/D 或开关量发生联系,内部又要进行数据处理,所以应用软件编制不但牵涉到微机接口方面的内容,而且牵涉到大量数据处理需要的工程及科学计算方法,所以,选择编程语言时要兼顾这两方面的要求。基于上述考虑,我们以 C 语言作为微机数据采集与处理系统的编程语言。

1.2 微机接口初步

在微机数据采集与处理系统中,在时域连续变化的模拟信号要经过 A/D 接口转换为离散数字信号进入微机,微机发出的控制量也要经过 D/A 接口转换为连续变化的模拟控制信号。在 A/D 采样时,不但牵涉到 A/D 本身的接口问题,而且 A/D 转换要用到计时、中断和

DMA 等技术,这牵涉到计数器接口芯片、中断控制器接口芯片和 DMA 控制器接口芯片等,为了以后讨论问题方便起见,本节先把微机接口方面的知识简单介绍一下。

1.2.1 接口基本概念

要利用微机实现数据采集与处理,必须通过各种输入输出接口(简称 I/O 接口)把外部设备与微机连起来。在英文中,接口叫“interface”,是由前缀“inter”(意为“相关的”)加上“face”(意为“面”,“界面”)构成的,所以,接口就是两个不同对象的交接部分。因此,软件有软件接口,硬件有硬件接口,这里指的是硬件接口。正规一点讲,接口是一组电路,是 CPU 与存贮器、输入输出设备等外设之间协调动作的控制电路。接口电路的作用就是将来自外部设备的数据信号传送给 CPU,CPU 对数据进行适当的加工后再通过接口传回外部设备。所以,接口的基本功能就是对数据传送控制。

接口电路往往集成在一块芯片上,接口芯片一般由控制命令逻辑电路、状态设置和存贮电路、数据存贮与缓冲电路三部分组成。

接口按功能特点分为通用并行输入输出接口,通用串行通讯接口、DMA 控制器和专用外设控制器等。

接口是用来控制数据传送的,就数据传送方式而言只有两种:串行传送(一次一位的传送)和并行传送(一次数位);就传送控制方式而言,有查询方式、中断方式和 DMA 方式三种。

1.2.2 接口寻址方法

为了同不同的外设打交道,要使用不同的接口电路。为区别不同的接口电路,一般赋予它们一个接口地址,这样微机就可以象访问存贮单元一样按地址访问这些接口电路,从而与外设发生联系。一个接口电路中根据需要可能有多个寄存器,如数据寄存器、状态寄存器和命令寄存器等,为了区别它们,也给予不同的地址,以便微机能正确找到它们。接口中除上述的寄存器外,也许还有其它不同电路,为区别它们,启动需要的电路工作,也可给它们赋以不同的地址。为了将这些地址和存贮器地址相区别,称它们为接口地址。有时也将上述提到的接口中可被主机直接访问的一些寄存器称为端口,一个接口常常有几个端口,如数据端口、状态端口、命令端口等,每个端口的地址叫端口地址,如何实现对这些接口地址、端口地址的访问,就是 I/O 地址的寻址问题。

I/O 接口的寻址方式一般有两种:一种是将接口地址和存贮器地址统一编址;一种是接口地址和存贮器地址独立分别编址。IBM PC 系列微机一般采用 I/O 独立编址方式,并采用专门的 I/O 指令(无论是汇编语言还是高级语言,都有专门访问 I/O 端口的指令)来对接口地址进行操作,这样存贮器地址和 I/O 接口地址可以重迭(所以有时也称为复盖编址),然而由于两者采用不同的指令进行读写操作,所以不会由于地址相同而混淆。

IBM PC 有很强的输入输出能力,主要表现在输入输出(I/O)地址分配和 I/O 通道设计上。虽然 8086/8088 的 I/O 指令可使用 16 位有效地址(A15~A0)而使 I/O 地址空间达 64K,但在实际 PC 系统中,只能寻址 1K 范围内的外设寄存器,这是因为 IBM PC 仅使用地址总线的低 10 位,即 A9~A0。在这 1K I/O 空间中又划分为两部分,一部分用于系统板上

接口芯片的寻址,一部分用于 I/O 通道上各插件板(即扩充插槽连接的插件板)的寻址。系统板上的接口芯片(包括 8255、8259、8253 和 8237 等)所占地址空间从 0000H 到 01FFH,共 512 个单元,当 I/O 地址 A9 位为 0 时,寻址系统板上的接口芯片,系统总线同系统板交换数据,图 1.2 为系统板 I/O 地址使用图,每个接口芯片占用 32 个单元的空间,但不一定全部占用。当 I/O 地址的 A9 位为 1 时,系统总线与 I/O 通道上的插件板交换数据信号,这部分空间从 0200H~03FFH,共 512 单元,图 1.3 为 I/O 通道插件板的使用图,在各部分地址之间尚有空闲空间,用户自行开发的插件板(如 A/D 板)可使用这些单元。

译码范围		使用地址		对应接口
0000H—001FH	32	0000H—000FH	16	DMA 芯片(8237)
0020H—003FH	32	0020H—0021H	2	中断芯片(8259A)
0040H—005FH	32	0040H—0043H	4	定时计数芯片(8253)
0060H—007FH	32	0060H—0063H	4	并行接口芯片(8255)
0080H—009FH	32	0080H—0083H	4	DMA 页面寄存器
00A0H—00BFH	32	00A0H	1	NMI 屏蔽位
00C0H—01FFH	320	系统板上不译码 或不用		

图 1.2 系统板 I/O 地址使用图

0200H	1		0200H	(不用)
0201H	1		0201H	游戏控制适配器
0202H~0277H	118		0202H~0277H	(不用)
0278H~027FH	8		0278H~027FH	第二打印机适配器
0280H~02F7H	120		0280H~02F7H	(不用)
02F8H~02FFH	8		02F8H~02FFH	第二串行口适配器
0300H~0377H	120		0300H~0377H	(不用)
0378H~037FH	8		0378H~037FH	打印机口适配器
0380H~03AFH	48		0380H~03AFH	(不用)
03B0H~03BFH	16		03B0H~03BFH	单色显示器和打印机适配器
03C0H~03CFH	16		03C0H~03CFH	(不用)
03D0H~03DFH	16		03D0H~03DFH	彩色/图形显示适配器
03E0H~03EFH	16		03E0H~03EFH	(不用)
03F0H~03F7H	8		03F0H~03F7H	磁盘驱动器适配器
03F8H~03FFH	8		03F8H~03FFH	串行口适配器

图 1.3 扩充插槽地址使用图

IBM PC 系列微机的设计采用了适应未来发展的开放式结构。系统中除键盘接口包括在系统板上外,其余外设一般通过适配器插件板与 CPU 连接,实现连接的电路即 I/O 通道。I/O 通道以插座的形式固定在系统板上,外设适配器插在通道插座上,外设与 CPU 之间的信息交换通过分布在 I/O 通道上的总线信号实现。总线信号共 62 个(286、386 可扩展为 98 个),与 I/O 通道插座引脚一一对应。

1.2.3 可编程并行接口 8255

8255 是一种可编程的外部接口芯片,通常作为微机系统总线与外设的接口控制部件,其工作方式可由编程灵活设定。IBM PC 系统板上就使用一片 8255 与键盘、扬声器等其它外设进行接口。有的数据采集插件板也用 8255 作为数字输入输出接口芯片。

1 8255 的组成

图 1.4 为 8255 的逻辑框图。这个芯片内部有 3 个 8 位的 I/O 端口, A 口、B 口和 C 口;也可分为各有 12 位的两组, A 组和 B 组。A 组包括 A 口 8 位和 C 口的高 4 位; B 组包括 B 口的 8 位和 C 口的低 4 位。A 组控制和 B 组控制用于实现方式选择等操作。读/写控制逻辑用于控制对芯片内寄存器的数据和控制字经数据总线缓冲器送入各组接口寄存器中。数据总线缓冲器直接和系统数据总线($D_7 \sim D_0$)相连,实现 CPU 与接口之间的信息交换。

8255 共有 40 个引脚,分别为图 1.4 的 $PA_7 \sim PA_0$ (A 口), $PB_7 \sim PB_0$ (B 口), $PC_7 \sim PC_0$ (C 口), $D_7 \sim D_0$ (数据总线), RD (读信号线), WR (写信号线), A_1 、 A_0 (口地址线), CS (片选线), $RESET$ (复位线), 外加 GND (地) 和 V_{cc} (电源)。

2 8255 控制字与工作方式

8255 有三种工作方式,即方式 0、1 和 2,这可以通过对芯片的编程,也就是对 8255 内的控制寄存器装入不同的控制字来决定其三种不同的工作方式。

控制字的最高位 D_7 叫工作方式标志位,若 $D_7 = 1$,则表示此控制字是规定接口作输入输出用的,并且由其余各位是 0 或 1 来决定 A 组和 B 组的工作方式。此时控制字的格式和功能如图 1.5 所示。8255 的三种工作方式简述如下。

(1) 方式 0—基本 I/O 方式

这是一种在 8255 和外设之间没有控制信号的输入输出方式,此时, A 口和 B 口可以由

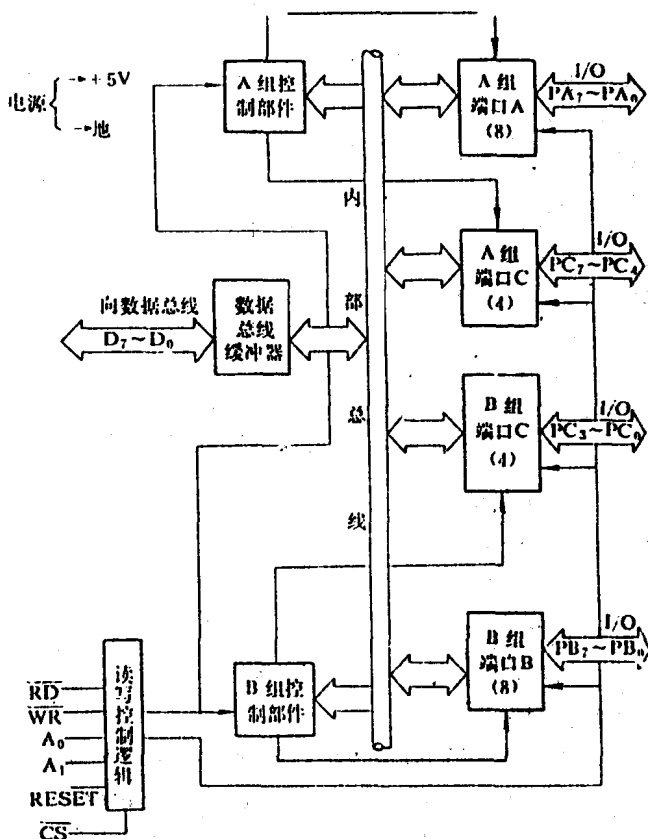


图 1.4 8255 组成

输入的控制字规定为输入或输出, C 口分为高 4 位(PC₇~PC₄)和低 4 位(PC₃~PC₀)两组, 均可分别由控制字规定为输入或输出。由于没有控制信号, 芯片各口的信息只能按 4 位或 8 位一组进行工作。C 口在其它方式下可输出控制信号, 但在方式 0 下它分成两组, 每组 4 位, 工作时只能将其其中一组的 4 位全部置为输入或输出, 但两组方向可以不同。

(2) 方式 1—单向 I/O 方式

这种方式具有选通控制输入/输出的能力。它分为 A、B 两组, A 组由数据口 A 和控制口 C 的高几位组成, B 组由数据口 B

和控制口 C 的低 3 位组成。数据口的输入输出都是锁存的, 与方式 0 相同, 通过 CPU 写入的控制字来规定它用作输入或是输出, C 口中的相应位用来寄存数据传送中所需的状态信息和控制信息, 例如数据就绪、应答、中断请求等。此时, C 口作为控制口, 不受控制字中 D₀ 位和 D₃ 位的影响。

(3) 方式 2—双向 I/O 方式

只有 A 口可使用方式 2, 此时, A 口为输入输出双向口, C 口中的 5 位(PC₃~PC₇)作为 A 口的控制位。B 口只能在方式 0 和方式 1 下工作, C 口的 3 位(PC₀~PC₂)可用作输入输出线, 也可作 B 口的控制位。

3 系统板上的 8255

在 IBM PC 系统板上, 装有一片 8255, 其中的 A 口、B 口、C 口均被 BIOS 编程在方式 0 下工作。A 口作为输入数据, 用于键盘中断处理程序中读取已串一并转换完毕的键盘扫描码, B 口作为输出控制, 用于对键盘串一并转换、RAM 和 I/O 通道检验以及扬声器等的启动控制。C 口亦作为输入数据, 高 4 位为一些状态测试位, 低 4 位用来读取系统板上系统配置

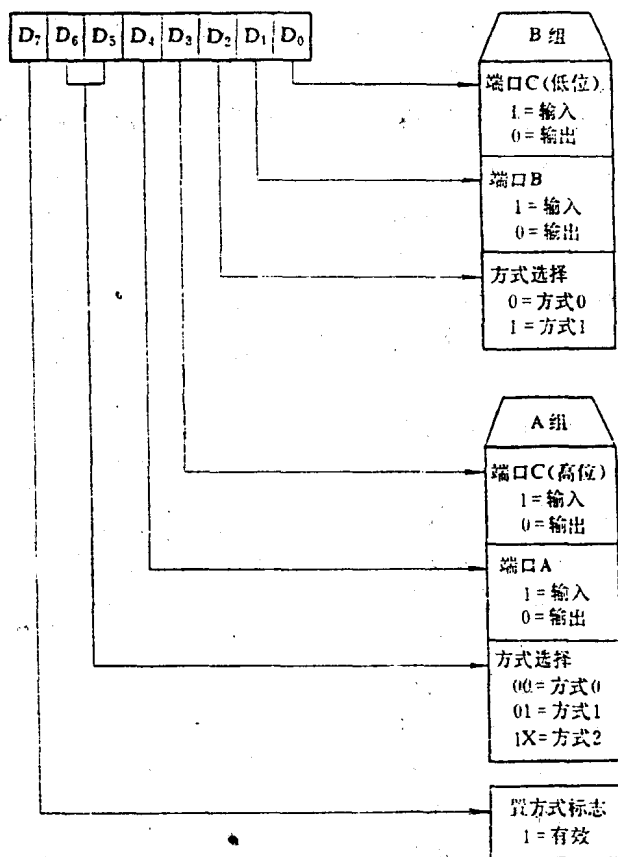


图 1.5 方式命令字格式

开关 DIP 的状态以确定系统的配置情况。

系统分配给 8255 的口地址分别为 60H、61H、62H 和 63H,其中 A 口对应 60H,B 口对应 61H,C 口对应 62H,控制寄存器对应 63H。系统初启时,BIOS 将 8255 的 A、C 口设为输入口,B 口设为输出口,工作于方式 0,对应的控制字为 99H,即(10011001)_B,对 8255 的控制是通过向控制寄存器(端口地址 63H)发出控制字来规定工作方式的。

1.2.4 可编程计数器/定时器 8253

8253 是一种通用的可编程计数/定时器,它具有三个功能完全相同的 16 位减法计数器,分别称为计数器 0、计数器 1 和计数器 2,每个计数器的工作方式都可以单独编程设定,因而能以不同的工作方式满足不同的接口要求。IBM PC 机系统板上就装有一片 8253,有的 A/D 板上也自带一片 8253,作为数据采集定时之用。

1 8253 的组成

如图 1.6 所示,8253 由三个计数器、控制寄存器、读写逻辑和总线缓冲器等四部分组成。数据总线缓冲器和系统数据总线(D₇~D₀)连接,CPU 通过它将控制命令字和计数值写入 8253,或者从 8253 读取当前计数值。读写逻辑的任务是接收来自 CPU 的控制信号,完成对 8253 各计数器的读/写操作,这些控制信号包括读信号 RD,写信号 WR,片选信号 CS 和芯片内部寄存器寻址信号 A₀~A₁。控制命令寄存器用来保存来自 CPU 的命令字,每个计数器都各有一个控制命令寄存器,用于保存该计数器的控制信息,命令字只有一种格式,在控制命令字中有两位专门用来指定当前命令字是发给哪个计数器的,命令寄存器只能写入,不能读出。8253 每个计数器都由一个 16 位锁定寄存器和一个 16 位计数寄存器组成,每个计数器有两个输入信号,时钟信号 CLK 和门控信号 GATE,以及一个输出信号 OUT。对 8253 编程时,送入每个计数器的计数值经锁定寄存器传给计数寄存器。8253 采用递减方式计数,即每当时钟输入端 CLK 出现一个脉冲时,计数寄存器保存的计数值减 1,直至为 0,此时 8253 在输出端 OUT 线上产生一个输出信号电平。

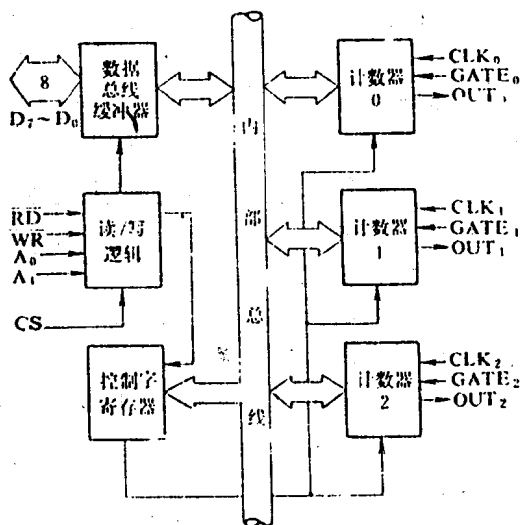


图 1.6 8253 组成

8253 每个计数器都由一个 16 位锁定寄存器和一个 16 位计数寄存器组成,每个计数器有两个输入信号,时钟信号 CLK 和门控信号 GATE,以及一个输出信号 OUT。对 8253 编程时,送入每个计数器的计数值经锁定寄存器传给计数寄存器。8253 采用递减方式计数,即每当时钟输入端 CLK 出现一个脉冲时,计数寄存器保存的计数值减 1,直至为 0,此时 8253 在输出端 OUT 线上产生一个输出信号电平。

8253 是一个 26 引脚芯片,除图 1.6 中的 D₇~D₀,RD,WR,A₀,A₁,CS,CLK₀~CLK₂,GATE₀~GATE₂,OUT₀~OUT₂ 外,还有未画的 GND(地)和 V_{cc}(电源)。

2 8253 的命令字和读/写操作

8253 的命令字有 8 位,保存在单字节命令寄存器中,图 1.7 给出其格式及各对应位意义。第 0 位定义计数值表示形式,可以是二进制 BINARY 形式,也可以是二一十进制 BCD 表

示形式;第 1 位~第 3 位设定计数器的工作方式,共 6 种方式;第 4、5 位用于设定计数器的读/写方式,这是因为计数器是 16 位,而 8253 以 8 位数据线与系统总线连接;第 6、7 位用来寻址 3 个计数器。

8253 的读/写操作如下:

(1) 读操作

读操作有两种方法。第一种方法利用门控 GATE 信号阻止时钟输入,在暂停计数操作时,读出确定的计数值。第二种方法是在计数过程中读出计数值,但不影响计数操作,在读当前计数值时,向端口发出位 4 位 5 都为 0 的锁存命令字,把当前的计数值锁存到寄存器中,供 CPU 读取。

(2) 写操作

写操作包括命令字的写入和计数值的写入。写入命令字时,只要给定写入端口的地址(命令字的最高二位),即可在任何适当时刻由 CPU 发出输出命令字指令,而计数值的写入要紧跟命令字之后。由于 16 位计数值要分两次写出,所以要按命令字第 4 和第 5 位规定的方式进行,注意高低字节顺序。由计数器递减计数,所以写入 0000H 计数值,则产生最大计数。

3 8253 的工作方式

8253 计数器的每个通道都有 6 种可编程选择的工作方式。对于每一种工作方式,由时钟输入信号 CLK 确定计数器递减速率(因此可起到计时作用),门控信号 GATE 用于允许或禁止 CLK 信号进入计数器,或者根据工作方式用作计数器的启停信号,计数结束时在输出线上产生一个 OUT 信号,可定义为脉冲、常数电位或周期信号。

8253 的 6 种工作方式简述如下。

(1) 方式 0—计数结束产生中断方式

这种方式下,启动计数有两种情况:WR 启动计数和 GATE 启动计数。无论怎样启动,

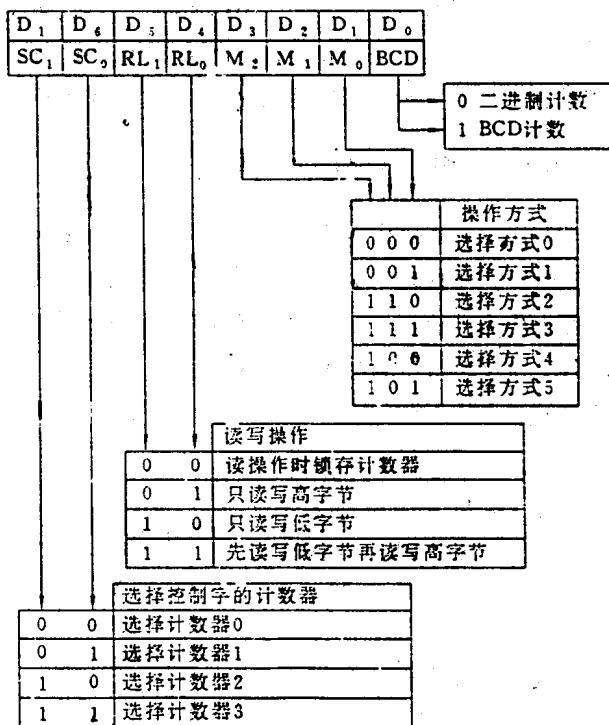


图 1.7 8253 命令字格式

都在计数到 0 时通过 OUT 线产生中断信号电平。(a)WR 启动计数:当方式控制字写入时,OUT 最初输出低电平,在计数值被写入所选的计数器之后,计数马上开始,此时输出仍维持原状(低电平),直到最终一次计数进入时,输出便成为高电平,并一直保持到重新写入控制字或重新写入新的计数值为止。OUT 输出信号作为计数结束的状态信号向 CPU 请求中断或供 CPU 查询。(b)GATE 启动计数:在计数过程中,若门控信号 GATE 变为低电平,则计数将暂时停止,待 GATE 变为高电平时,继续计数。

若在计数过程中对计数器进行写入,则写入第一字节时,中止现在的计数,写入第二字节时,开始新的计数。

(2)方式 1—可编程单脉冲输出方式

这种方式产生单拍负脉冲信号输出,脉冲宽度由计数值决定。在设定工作方式和写入计数值后,OUT 输出高电平。在触发信号(GATE)上升为高电平时,OUT 输出低电平,并开始计数。最终一次计数时,OUT 输出高电平。

在 OUT 输出保持低电平期间,若输入一个新的计数值,则低电平的持续时间不受影响,只有当下一触发脉冲(GATE)到来时,才使用新的计数值。如果计数尚未结束时,又出现新的触发脉冲,则从新的触发脉冲上升沿之后开始重新计数。CPU 任何时候都可以读出计数器的值,而且对单拍脉冲无影响。

(3)方式 2—脉冲信号发生器方式

在该方式下,计数器装入初始值开始工作后,输出端将不断输出负脉冲,其宽度等于一个时钟周期,两脉冲间的时钟个数等于计数器装入的初始值。在这种方式下,门控信号 GATE 相当于复位信号。当 $GATE=0$ 时,立即停止计数强迫输出为高电平;当 $GATE=1$ 时便启动一次新的计数的周期,这样可以用一个外部控制逻辑来控制 GATE,从而到达同步计数的作用。当然也可以用软件控制 GATE 达到同步控制目的。

(4)方式 3—方波频率发生器方式

与方式 2 类似,当装入一个计数器初值 n 后,在 GATE 信号的上升沿启动计数,在完成前一半计数时,输出一直保持高电平,而在进行后一半计数时,输出又成为低电平。若装入的计数值 n 为奇数,则在 $(n+1)/2$ 个计数期间,输出高电平。

(5)方式 4—软件触发选通方式

该方式下的计数操作是从写入计数值开始的,所以称为软件触发选通。采用这种方式时,OUT 在方式控制置位后输出高电平。当写入计数值时,计数器立即开始计数,在计数到最后一个数时,OUT 输出一个负脉冲选通信号,其脉冲宽度等于一个输入时钟周期。然后 OUT 又变为高电平。如果在计数期间写入新的计数值,则当前周期不受影响,但随后的周期将按新数值计数。当门控 GATE 变为低电平时,禁止计数;GATE 变为高电平时,重新开始计数。

(6)方式 5—硬件触发选通方式

所谓硬件触发选通,即在该方式下写入计数值命令字并不启动计数,而只有门控信号 GATE 的上升沿到来之后才开始计数。OUT 信号在 GATE 输入之前一直保持高电平,直至计数结束,才输出一个选通负脉冲,其宽度等于一个输入时钟。在该方式下,GATE 是高电平或低电平都不再影响计数器工作,但计数操作可用 GATE 的上升沿重新触发,当正在计数期间重新触发,便又从原来的计数值开始计数。

最后需要说明:当 8253 用于计数和产生设定频率的脉冲输出时,一定要弄清楚计数器何时开始计数,又何时结束计数,这样才能使理论预算的和实际测定的相符。

4 系统板上的 8253

IBM PC 系统板上用了一片 8253 芯片。8283 时钟发生器输出的 2.38MHz 的信号在二分频后(成为频率为 1.19MHz 的信号)作为 8253 三个计数/定时器的时钟输入。计数器 0 被编程为方式 3(方波发生器), $GATE_0$ 固定为高电平, OUT_0 作为中断请求接至 8259 中断控制器的 IRQ_0 ,此定时中断($54.936ms \approx 55ms$ 产生一次)用于报时时钟和磁盘驱动器的马达定时。计数器 1 被编程为方式 2(频率发生器), $GATE_1$ 也固定为高电平, OUT_1 的输出经一 D 型触发器后作为对 8237DMA 控制器通道 0 的 DMA 服务请求 $DREQ_0$,用于定时(约 $15\mu s$)启动刷新动态 RAM。计数器 2 编程为方式 3,1KHz 的方波输出滤掉高频分量后送到扬声器,它的门控信号 $GATE_2$ 是 8255 的 PB_0 , OUT_2 输出亦经一与门控制,控制信号为 PB_1 ,于是,可用 PB_1 、 PB_0 同时为“高”的时间来控制发长音(3 秒)还是发短音(0.5 秒)。

系统板上 8253 的计数器 0、1、2 和命令寄存器的口地址依次分别为 40H、41H、42H 和 43H。

1.2.5 可编程中断控制器 8259

1 中断概述

所谓中断,是指 CPU 在正常运行程序时,由于程序的预先安排的内外事件,引起了 CPU 中断正在运行的程序,而转到预先安排的内部事件或外部事件的服务程序中去。这些引起程序中断的事件称为中断源。预先安排的事件是指 PC 机的中断指令,执行至此,立即转向对应的服务程序去执行。内部事件是指系统板上出现的一些事件信号,外部事件是指某些接口设备所发出的请求中断程序执行的信号,这些信号称为中断请求信号。中断请求信号何时发生是不能预知的,然而,它们一旦请求中断,则会向 CPU 发出信号,因此这些信号 CPU 是可以测得的,这样 CPU 就无须花大量的时间去查询这些信号是否产生,因为中断请求信号一旦产生,便会马上通知 CPU,如键盘,何时键按下是随机的,因而 CPU 可以对键盘不加等待,而去执行其他程序,一旦有键按下,键盘马上产生中断请求信号,CPU 得知这一信号后,便立即去执行为键盘服务的中断程序,服务完后,CPU 又恢复执行被中断的程序。所以,使用中断技术后,可以使 CPU 的利用率大大提高。有时中断请求信号(中断源)可能有好几个,因此 CPU 响应这些中断就得有先后次序,这称为中断的优先级。CPU 响应中断按优先级的高低次序顺序进行。有些中断源产生的中断,可以用编程的方法,使 CPU 不予理睬,这叫中断的屏蔽。CPU 响应中断,由中断源提供地址信息,引导程序转移并执行中断服务程序,这个中断服务程序存放地址称为中断向量,它一般是和中断源相对应的。PC 机采用类型码来标识中断源。PC 机中断有两类:由执行某些指令引起的软中断和接口设备引起的硬中断。软中断包括除数为 0 商溢出中断(0 型中断),溢出标志位置位中断(4 型中断),单步中断(1 型中断),执行中断指令 $INT\ n$ 产生的 n 型中断等。8088CPU 引脚有两条线:非屏蔽中断 NMI 和可屏蔽中断 $INTR$ 线,由这两条线上产生中断请求信号而引起的中断称为硬中断。 NMI 中断有三种来源:协处理器 8087 中断请求、RAM 产生奇偶错和 I/O 通道检查出错。 $INTR$ 中断请求线接系统板上的 8259A 中断控制器,该中断控制器有可进行优先级排