

微型计算机系列培训教程（第三册）

张载鸿 主编

微型机(PC 系列)接口控制教程

张载鸿 编

清华大学出版社

内 容 提 要

本书以IBM-PC系列及兼容机为背景,阐述微型计算机的I/O接口控制技术。全书共九章,内容分别为: I/O 接口概述、中断系统及编程、DMA 系统及编程、定时及发声系统的应用、键盘接口控制及应用、打印机接口控制及应用、异步串行接口控制及应用、视频显示接口控制及应用以及软盘、硬盘子系统接口控制及应用。

本书以“系统和应用相结合,硬件和软件相结合”为编写原则,叙述的概念清晰,提供的I/O接口技术全面。适合作为微型计算机系列培训的教材,也可作为高等院校从事计算机教学和科研的师生参考书。

(京)新登字 158 号

微型机(PC 系列)接口控制教程

张载鸿 编

☆

清华大学出版社

北京 清华园

门头沟胶印厂印刷

新华书店总店科技发行所发行

☆

开本: 787×1092 1/16 印张: 28.5 字数: 700 千字

1992年10月第1版 1994年1月第2次印刷

印数: 10001-18000

ISBN 7-302-01152-4/TP·427

定价: 22.00 元

编者的话

本书是由北京科海培训中心组织编写的《微型计算机系列培训教程》的第三册。在第一、二册的基础上，读者通过本教程的学习，可深入了解微型计算机内部的工作原理，尤其对系统如何控制各个外部设备的全过程有一透彻的理解。全书分成九章，由下述三部分组成：

第一部分包括第一章。它介绍微型机 I/O 接口控制的基本概念，对 I/O 接口的控制方式、硬件配置、端口布局及通道信号的作用都作了概要性的阐述，本章是学习以后各章的基础。

第二部分包括第二、三、四章。它们在分别叙述系统主要的 I/O 接口芯片，如中断控制器、DMA 控制器和定时器的基础上，介绍 PC 系列的中断控制系统、DMA 控制系统和定时与发声系统的工作原理和应用。通过这几章的学习可从总体上把握 PC 系列 I/O 机的接口控制技术。

第三部分包括第五、六、七、八、九章。它们分别叙述 PC 系列的键盘接口控制、打印机接口控制、异步串行通信接口控制、视频显示接口控制以及软、硬磁盘接口控制诸技术。各章内容都从基本工作原理入手，结合硬件逻辑和软件编程两方面，详尽分析接口控制的具体功能和方法，并通过实例指出接口应用及注意事项。对于本系列教材第二册中使用的各个 BIOS 中断功能（与 I/O 接口有关的），在各章中都进行了相应的剖析，有助于读者自行开发实用的 I/O 接口应用程序。

各章后面都附有习题。一类是答题，帮助读者熟悉每部分的内容，另一类是选择题，具有思考价值，可开拓读者的思路。

因本人水平有限，书中提出的观点和所举的实例难免有不当之处，敬请读者批评指正。

本书能顺利出版得到北京科海培训中心主任华根娣和编辑夏非彼的热心帮助和支持，在此表示诚挚的谢意。

编 者

序

《微型计算机系列教程》和广大读者见面了。

本套系列教程的出版顺应了当今微型计算机发展的总趋势，可满足各行各业微型计算机应用人员学习、掌握 PC 机系统的迫切需求，并对进一步开发微型机技术向深层次开拓发挥强大的推动作用。

微型计算机在 70 年代末期由 8 位机开始走向 16 位机的发展阶段，经过 80 年代整整十年的普及、推广和应用，使微型计算机这一高科技领域的“宠儿”触及了社会的各个角落，它为计算机发展历史作出了不可磨灭的功绩。在众多型号的微型计算机家族中，尤以 IBM PC 系列微型机，因其先进的结构设计和丰富的系统软件和实用软件能适应各种层次应用人员的要求，而成为过去十年间微型计算机工业事实上的生产标准和市场导向。

自 80 年代后期开始，微型计算机进入 32 位机的发展阶段，各类 386 机、486 机相继占领个人计算机的市场。根据我国经济实力和发展趋势推测，在二十世纪的最后一个十年间，国内的微型计算机应用技术仍将与 IBM PC 系列兼容的各类 PC 机作为主要开发对象。

基于此，一套以基础性、实用性和系统性为编写宗旨的《微型计算机系列教程》在九十年代的今天问世，乃是时代发展的要求。

本套系列教程以 IBM PC 系列及兼容机为展开讨论的模式，本着由浅入深、由外到里的原则，按 PC 机层次结构一步步地引导读者了解、使用和掌握微型机系统的结构原理、操作和维护方法，以及软件编程技术和硬件维护手段，最终达到 PC 系列开发应用的目的。

本套系列教程的主编张载鸿教授制定了编写大纲，确立其编写风格，并对全套各册教程内容作了全面的审阅和修改。参予各册具体编写的作者都是工作在微型计算机教学、科研、培训和维修第一线的教师和工程师，他们积累了丰富的实践经验。在各册编写中始终贯彻“理论与实践相结合、系统与应用相结合、软件与硬件相结合”的方针，使本套教程具有“内容简炼、编排新颖、叙述清晰、实例丰富”的特色。各册每章后面按内容介绍顺序附有习题，有的检验读者对内容理解的程度，有的可增长读者更多的知识。

愿所有的读者从中获得启迪和乐趣！

《微型计算机系列教程》分两批出版。首批出版的是前三册，其内容分别如下：

第一册书名是《微型计算机应用基础教程》，由王路敬执笔。

该册共分九章。从计算机基础知识入手，叙述微型机的结构特点、安装检测和日常维护；着重讲解中西文操作系统（PC-DOS 和 CC-DOS）的命令使用和上机操作，强调软盘和硬盘的使用技术；对汉化 DBASE III 和常用的汉字输入软件、编辑软件、制表软件和 WPS 排版软件作了实用性的阐述，并具体指出数据库编程方法和技巧。

第二册书名是《微型计算机系统功能教程》，由张昆藏执笔。

该册共分九章。从计算机核心 CPU 结构入手，在叙述指令系统的基础上，具体讲解汇编语言编程方法；PC 系统提供的功能有数百个，包括 DOS 功能和 BIOS 功能两大类，通过数十个实例详细描述字符设备 I/O、磁盘 I/O、时钟设备 I/O 以及文件管理等功能，最后讨论各种高级语言程序调用汇编子程序的编程技术。

第三册书名是《微型计算机接口控制教程》，由张载鸿执笔。

该册共分九章。从计算机接口控制方式入手，在叙述微型机常用的程序查询、中断控制和 DMA 传输三种控制原理的基础上，讲解 PC 系列各个接口控制卡的硬件逻辑和软件编程的相互关系；具体指出对 I/O 接口芯片、键盘控制器、日时钟和实时钟、打印机控制卡、串行通信控制卡、彩色图形控制卡、软盘控制卡和硬盘控制卡的编程技术。

计划第二批出版的是后二册，其内容安排如下：

第四册书名是《微型计算机硬件维修教程》。

该册从微型机常用的逻辑电路入手，在叙述微型机硬件的基础上，讲解系统板、显示器、打印机、内存存储器、软盘子系统和硬盘子系统等部件的维修手段和方法，并通过实例分析常见故障的处置技术。

第五册书名是《微型计算机高级技术教程》。

该册从操作系统 DOS 结构剖析入手，公开 DOS 内核的全部功能（包括 DOS 保留功能），通过具体实例讲解 DOS 不可重入性及其对策、内存驻留程序 TSR 编程技术、进程管理和多任务接口，以及网络重定向技术等。

本套系列教程可用作各类微型机技术培训班的教材。也可作为大专院校师生和各行业人员学习微型计算机的参考用书。

本套系列教程是由中国科学院北京科海中心组织编写的，自始至终得到中心主任华根娣、编辑夏非彼热情的支持和帮助，在此表示衷心感谢。

本人水平有限，主编一套系列教程是初次尝试，难免有不当之处，谨请读者批评指正。

主编 张载鸿
于北京计算机学院
1992年2月

目 录

序

第一章 PC 系列接口概述	1
1.1 I/O 接口概念	1
1.1.1 I/O 接口的硬件分类	1
1.1.2 I/O 接口与总线的连接	2
1.1.3 I/O 接口的功能	2
1.1.4 I/O 接口的端口寻址	3
1.1.5 I/O 接口的控制方式	4
1.2 PC 系列 I/O 接口配置	10
1.2.1 I/O 接口芯片的配置	11
1.2.2 I/O 接口控制卡的配置	13
1.2.3 286/386 系统 I/O 接口的配置	17
1.3 PC 系列 I/O 端口布局	20
1.3.1 I/O 端口的寻址特点	20
1.3.2 I/O 端口地址分配	21
1.4 PC 系列 I/O 通道	23
1.4.1 I/O 通道的作用	24
1.4.2 I/O 通道的局限性	26
1.4.3 I/O 通道信号分配	28
习题一	35
第二章 PC 系列中断系统	37
2.1 中断源类型	37
2.1.1 内中断源	37
2.1.2 外中断源	40
2.1.3 软中断源	42
2.2 中断向量表	45
2.2.1 中断向量表作用	45
2.2.2 中断向量表初始化	47
2.2.3 中断向量表内容	51
2.3 中断控制编程	56
2.3.1 中断控制器功能	56
2.3.2 中断控制器初始化编程	59
2.3.3 中断控制器操作编辑	67
2.3.4 中断控制编程原则	71

习题二	75
第三章 PC 系列 DMA 系统	77
3.1 DMA 系统的功能	77
3.1.1 DMA 控制器外特性	77
3.1.2 DMA 控制器内部功能	80
3.2 DMA 系统编程	82
3.2.1 8237A-5 芯片端口	82
3.2.2 8237A-5 编程应用	92
习题三	102
第四章 PC 系列定时与发声系统	104
4.1 PC 系列定时系统	104
4.1.1 PC 系列定时器	104
4.1.2 PC 系列定时器使用	111
4.1.3 PC 系列日时钟及应用	115
4.1.4 PC 系列实时钟及应用	123
4.2 PC 系列发声系统	136
4.2.1 PC 系列两种发声源	136
4.2.2 PC 系列发声应用	139
习题四	146
第五章 PC 系列键盘接口控制	148
5.1 PC 系列键盘工作原理	148
5.1.1 PC 系列键盘的特点	148
5.1.2 PC 系列键盘扫视法	148
5.1.3 键盘扫描码的发送	150
5.2 PC 标准键盘接口逻辑	154
5.2.1 标准键盘接口功能	154
5.2.2 标准键盘接口逻辑	154
5.3 PC 扩展键盘接口逻辑	156
5.3.1 扩展键盘接口功能	156
5.3.2 扩展键盘接口编程	160
5.4 键盘中断与键盘 I/O	173
5.4.1 BIOS 键盘缓冲区	173
5.4.2 键盘中断处理功能	176
5.4.3 键盘 I/O 处理功能	179
5.4.4 键盘中断与键盘 I/O 应用	181
习题五	185
第六章 PC 系列打印机接口控制	188
6.1 打印机并行接口逻辑及编程	188
6.1.1 并行接口逻辑结构	188

6.1.2	打印机时序及编程	192
6.2	打印机 I/O 程序及应用	198
6.3	针式打印机工作原理	201
6.3.1	针式打印机逻辑结构	202
6.3.2	针式打印机数据处理过程	207
6.3.3	针式打印机功能码	211
6.4	汉字打印一般技术	216
6.4.1	点图形打印方式的应用	217
6.4.2	汉字打印驱动程序	223
	习题六	229
第七章	PC 系列异步串行接口控制	232
7.1	串行通信接口标准	232
7.1.1	串行通信类型	232
7.1.2	串行通信传输方式	236
7.1.3	串行接口 RS-232C 标准	240
7.2	异步串行通信接口	245
7.2.1	调制解调器外特性	246
7.2.2	RS-232C 异步串行口连接方式	248
7.2.3	异步串行通信接口逻辑	254
7.3	异步串行通信编程应用	264
7.3.1	PC 系列串行口差异	264
7.3.2	查询 I/O 异步通信编程	266
7.3.3	BIOS 异步通信程序	274
7.3.4	中断 I/O 异步通信编程	276
	习题七	280
第八章	PC 系列视频显示接口控制	283
8.1	视频显示标准	283
8.2	视频显示的工作原理	287
8.2.1	视频扫描操作	287
8.2.2	视频显示一般原理	291
8.3	视频显示接口逻辑	294
8.3.1	MC6845 控制器	295
8.3.2	CGA 控制卡逻辑	298
8.3.3	CGA I/O 端口及编程	301
8.4	CGA 显示编程	308
8.4.1	CGA 字符显示方式特点	308
8.4.2	CGA 字符显示方式编程	311
8.4.3	CGA 图形显示方式特点	329
8.4.4	CGA 图形显示方式编程	335

习题八	338
第九章 PC 系列磁盘子系统接口控制	342
9.1 磁盘数据存储原理	342
9.1.1 数字磁记录原理	342
9.1.2 数字磁记录编码方式	346
9.1.3 磁盘格式化——磁道格式	352
9.2 PC 系列软盘接口控制	356
9.2.1 360KB 软盘驱动器结构逻辑	356
9.2.2 软盘控制卡结构逻辑	362
9.2.3 软盘控制器编程方法	370
9.2.4 软盘控制器命令及执行	376
9.2.5 软盘控制器编程举例	387
9.2.6 1.2MB 软盘接口控制	392
9.2.7 软盘 I/O 程序 INT 13H	399
9.3 PC 系列硬盘接口控制	402
9.3.1 硬盘驱动器特点	405
9.3.2 XT 硬盘接口控制	412
9.3.3 AT 硬盘接口控制	418
9.3.4 硬盘 I/O 程序 INT 13H	429
习题九	438

第一章 PC 系列接口概述

一个完整的微型计算机系统由硬件系统和软件系统组成。硬件系统的核心是微计算机 (Microcomputer)，此外，还包括外围设备和电源部件。微计算机的核心是微处理机或称微处理器 (Microprocessor)。它只包含通常所说的控制器、运算器和数据通路，执行机器语言描述的一条条指令，完成对信息的控制 and 处理。若要对信息进行存储、输入或输出，则需为微处理机配备存储器 (RAM 和 ROM) 和输入 / 输出电路，并用系统总线将它们连在一起。这样就构成了一台微计算机。

在这里，整机控制的中心是微计算机 (故称中央处理器，简称 CPU)，而它与存储器及输入 / 输出电路部件的相连都需要一个接口来实现。前者称为存储器接口，后者称为 I/O 接口。但存储器通常是在 CPU 的同步控制下工作的，其接口电路及相应的控制比较单一、简单，而 I/O 设备品种繁多，其相应的 I/O 电路逻辑各异，控制复杂，故习惯上把 CPU 和存储器合称为主机系统，凡说到“接口”两字，一般专指 I/O 接口而言。

本章按此定义叙述 PC 系列接口的基本概况，包括 I/O 接口的配置和控制方式，以及 I/O 端口地址的分配。

1.1 I/O 接口概念

1.1.1 I/O 接口的硬件分类

既然，I/O 接口的功能是负责实现 CPU 通过系统总线把 I/O 电路和外围设备互连在一起，那么，按照电路和设备的复杂程度，I/O 接口的硬件主要分成两大类：

1. I/O 接口芯片

这些芯片大多是可编程的大规模集成电路。它们可通过 CPU 输出不同的命令和参数，灵活地控制互连的 I/O 电路或某些简单的外围设备进行相应的操作，如定时 / 计数器、中断控制器、DMA 控制器、并行接口和用单片机构成的键盘控制器等。

2. I/O 接口控制卡

这些接口控制卡由若干个集成电路 (大、中、小规模都可能) 按一定的逻辑构成，组装为一个部件。它或直接与 CPU 拼接在一个系统板上，或干脆制成一个插件插在系统总线槽上。按照所连接的外部设备 (一般都很复杂) 控制的难易程度，该控制卡的核心器件或为一般的接口芯片或为微处理器。凡安装微处理器的接口控制卡通常称为智能接口，这种卡上必有一片 EPROM 芯片，芯片内固化了控制程序，如 PC/XT 的硬盘驱动器接口控制卡。

在此说明一点，在 IBM 公司出版的 PC 系列技术资料上，是使用 Adapter (中译名“适配器”) 一词描述上述插件板的。本书按国内习惯叫法并考虑与接口概念相统一，故改用“接口控制卡”似更为贴切。

1.1.2 I/O 接口与总线的连接

在微计算机中，中央处理机是通过总线（含地址、数据和控制信息）与存储器、I/O 接口相连的。按 I/O 电路和外围设备不同，I/O 接口电路又分为 I/O 接口芯片和 I/O 控制卡。典型 PC 系列机 I/O 接口与总线的连接示于图 1.1。

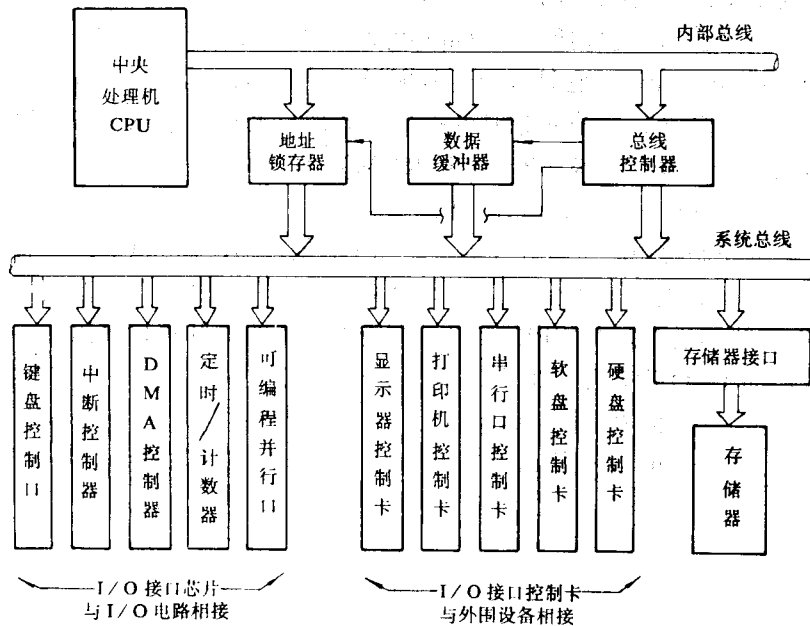


图 1.1 I/O 接口与总线的连接示意

由图 1.1 可知，不管是 PC 系列的哪个机种，就 I/O 接口的逻辑功能而言，它总是两类 I/O 接口硬件分别通过系统总线与中央处理机相连，以构成一台微计算机的硬件系统的。

1.1.3 I/O 接口的功能

前已指出，微处理机与 I/O 设备实现数据传输远不如与存储器那样简单。因为后者由半导体电路芯片堆积而成，而且，系统选用的存储器芯片的访问周期通常与时钟周期是同一个数量级，而存储器本身又提供数据缓冲的能力，所以微处理机一般只需要 2 个时钟周期左右的时间，便能在总线上存取数据。然而，I/O 设备品种繁多，几乎无例外地都离不开机电传动装置，这就给微处理机在与 I/O 设备进行数据交换时带来以下 4 个问题：

1. 速度的不匹配

I/O 设备的工作速度比微处理机慢得多，而且因设备种类不同，其速度差异甚大。有每秒能传输几兆位的硬盘驱动器，但也有每秒仅能输出 100 多个字符的串行打印机。

2. 时序的不匹配

一般，I/O 设备上的信息处理都有自身的定时控制电路，以自己的速率传输数据，无法与微处理机的时序取得同步。

3. 信息格式的不匹配

在不同的 I/O 设备上，存储和处理信息的格式不同。有并行或串行方式之分；也有二进制位或 ASCII 编码或 BCD 编码之分；若在磁盘上存储信息，更有按不同的磁道记录格式将选定的编码信息进行存取。这些信息格式都比微处理机与存储器间的数据交换格式复杂得多。

4. 信息类型与电平的不匹配

不同 I/O 设备处理的信号类型不同。它们或是数字电压，或是连续电流或是其它模拟量，而且，信号电平的幅值不一。

鉴于这些，微处理机与 I/O 设备间信息的交换必须通过 I/O 接口的控制来解决这 4 个问题。于是，设计一个 I/O 接口要完成如下功能：

1) 设置数据的寄存、缓冲逻辑，以适应两者速度上的差异。它们通常由若干个寄存器或 RAM 芯片组成。若 RAM 容量足够大，在某些接口上可实现批量数据的传输，如硬盘驱动器接口控制卡。

2) 设置信息格式相容性转换逻辑，以满足两者各自对数据格式的要求。如串行转换为并行，或反之。

3) 设置电气性能的适配逻辑，以确保两者的电路按各自要求的信号类型和电平工作。如电平转换驱动器，或数/模、模/数转换器等。

4) 设置时序控制异步逻辑，以实现两者间异步传输的规程，满足双方各自对时序的要求。通常，用“握手应答”方式确保两者操作同步。

此外，为正确完成微处理机与 I/O 设备间的数据传输全过程，I/O 接口还应具有：

5) 设置地址译码和设备选择逻辑，以保证微处理机按照预定的路径访问选定的 I/O 设备。

6) 设置设备控制及监测逻辑，以保证接受微处理机输出的命令和参数，按指定的命令控制设备完成相应的操作，并把指定设备的工作状态返回给微处理机。

7) 设置中断和（或）DMA 控制逻辑，以保证在中断和（或）DMA 允许的情况下，产生中断和（或）DMA 请求信号，并在接受到中断和（或）DMA 应答之后，实施对设备中断处理和（或）DMA 的传输。

需指出的是，由于设备的功能不同，微处理机对设备的控制方式也可随之而变，因此，并非所有的 I/O 接口都应具备上述 7 个功能。不过，最基本的第 1 条、第 5 条和第 6 条是每个 I/O 接口所必备的功能，即通常所说的数据端口、命令端口和状态端口，这三个端口是实现数据输入/输出操作的充要条件。

1.1.4 I/O 接口的端口寻址

所谓端口是指 I/O 接口（包括芯片和控制卡）中供 CPU 直接存取访问的那些寄存器或某些特定硬件电路。一个 I/O 接口总要包括若干个端口，除常见的数据端口、命令端口和状态端口外，还有特殊用途的端口：方式控制端口、操作结果端口和地址索引端口

等等。端口的多少及其相应的功能完全取决于一个 I/O 接口所关联的 I/O 设备。这将在以后各章详尽叙述。但需指出，一个端口可设定为只读（一般为状态或结果信息）、读写（一般为数据或命令信息）、或只写（读出无意义，如方式控制、命令参数信息）的。这种属性也是在设计 I/O 接口功能时决定的，用户不能变更。

既然端口可被微处理机访问，如同存储器的单元，那么每个端口也存在着编址的方式问题。在当今流行的各类微型机中，对 I/O 接口的端口编址有两种办法：端口统一编址和端口独立编址。用微处理器 6800、6502、68000 系列构成的微型机采用前一种方法；而用微处理器 Z-80、Z-8000 或 8085、8086、8088、80286、80386 系列构成的微型机都采用后一种方法。

一、端口统一编址

这种编址的出发点是把每一个端口视为一个存储器单元，并赋以相应的存储器地址。微处理机访问端口，如同访问存储器（只是地址值不同），所有访内指令同样适合于 I/O 端口。

由于端口地址被映象到存储空间，作为存储空间的一小部分，因此，这种统一编址亦称为“存储器映象编址”。

端口和存储单元统一编址的最大优点是无需专门的 I/O 指令，因而简化了指令系统，并可通过功能强的访内指令直接对 I/O 数据进行算术或逻辑运算。

当然，这种编址有它的不足之处：除了要占用原本就有限的一部分存储空间外，最主要的是因访内指令一般都需三、四个字节，使原来极简单的 I/O 数据传输时间加长了。

二、端口独立编址

这种编址的出发点是把所有 I/O 接口看作一个独立于存储器空间的 I/O 空间。在这个 I/O 空间内，每个端口都被分配一个地址与之对应。

显然，要访问独立于存储器空间的端口，必须用专门的 I/O 指令。为加快 I/O 数据的传输速度，设计的这种 I/O 指令均为单字节或多字节（指令直接带端口地址）。不过，这种 I/O 指令仅作数据传送而无算术或逻辑运算功能。

1.1.5 I/O 接口的控制方式

从整个计算机的发展来看，中央处理机对输入输出设备的控制方式大致有程序查询方式、中断处理方式、DMA 传输方式、通道控制方式和 I/O 处理机方式。前 3 种通常用于微型机和小型机，而后 2 种主要由中型机和大型机使用。

对于 PC 系列微型机，CPU 对不同 I/O 接口采取不同的控制方式，但都离不开前三种。

一、程序查询方式

程序查询的基本思想是由 CPU 主动通过 I/O 指令询问指定设备的当前状态。若设备就绪，则立即与设备进行数据交换，否则，CPU 处于循环查询状态。

在实际应用中，为防止设备因某种原因发生故障而无法就绪，从而导致 CPU 处于无

限循环查询之中，通常在查询流程中设计一个等待超时值（具体值随设备而变）进行检测，称为超时检测代码段。如果在规定时间内设备一直未能就绪，则迫使 CPU 从循环查询中跳出不再等待，并置超时错标志。

图 1.2 为程序查询方式的实际流程。

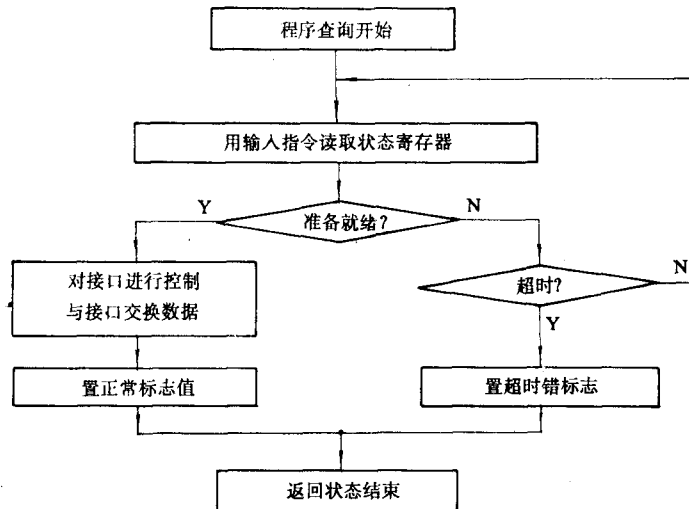


图 1.2 程序查询方式的实际流程

从图 1.2 可知，为确保程序查询方式的正确执行，I/O 接口应具备以下条件：

1. 设置端口地址符合逻辑

该逻辑的设置应能正确识别 CPU 本次输出的端口地址（所有接口都会接收到）是否对自己“呼叫”。若与本端口地址符合，则产生相应的控制信号使选中的端口（即某个寄存器）进入工作状态，否则不予理睬。

在实际的 I/O 接口设计时，该逻辑究竟放在接口一侧还是微处理机一侧，取决于接口的具体特性。一般而言，微处理机访问 I/O 接口芯片，往往通过一个总的地址符合逻辑产生不同的选中信号直连到芯片的片选端 CS，而对 I/O 接口控制卡访问时，则每块卡上都有地址符合逻辑。

2. 设置状态寄存器

该寄存器反映当前设备的各种状态信息，包括准备就绪位、忙碌位和错误位等。有的设备只用于输出，如打印机、绘图仪等，此时准备就绪位和忙碌位合二为一。若设备就绪或不忙碌，即可开始数据传输。有的设备既可输入又能输出，如串行、并行接口等，准备就绪位和忙碌位的含义就不同了。前者表示数据已准备好，可用于 CPU 输入；后者表示设备处于空闲（不忙碌）状态，可用于 CPU 输出。

另外，错误位的长度及其含义更与设备的特性密切相关，有的只需一、二位即可（如打印机），有的可能需要几个字节才能反映各种错误类型（如磁盘驱动器）。

总之，状态寄存器各位的含义没有统一的标准，完全取决于具体的设备。

3. 设置数据寄存器

该寄存器存放供输入输出使用的数据，其中可为数据、命令参数等。通常寄存器的长度为 8 位或 16 位。

4. 设置控制寄存器

该寄存器随设备种类不同而相差甚大。通常，存放 CPU 通过输出指令传送的控制命令，以控制设备完成相应的操作。

控制寄存器（或称命令寄存器）各位可包括启动、初始化、控制方式、数据选通、中断允许/禁止、DMA 允许/禁止和内部通道选择，等等。由于该寄存器的含义随设备的具体特征而变，因此，寄存器的长度由单字节到多字节不等。

基于此，程序查询方式最大的优点是简单、经济，只需少量的硬件（如上述的几个端口寄存器）即可实现处理机与 I/O 设备的数据传送。然而，这种控制方式使 CPU 效率降低，尤其当处理器之高速与 I/O 设备之低速的悬殊愈来愈大时，情况会更加严重。

二、中断处理方式

中断处理方式基于这样一种思想：由 I/O 设备主动地提出服务请求（当自身准备就绪或空闲时），一旦 CPU 响应此请求，便暂停正在执行的某个程序，转而执行为该请求服务的中断子程序。当该中断子程序完成指定的服务后，又恢复执行原被中断了的程序。

显然，这种控制方式使 CPU 省去主动查询设备状态及避免为等待设备就绪而浪费的大量时间，不仅克服了程序查询方式导致 CPU 使用效率过低的缺点，而且，又能满足 I/O 设备的实时要求。

微计算机中配备着多个 I/O 设备，每一种设备提供的服务请求（在此称中断请求）都是异步实时性的。所以，为使各个中断请求与微处理机协调同步，系统除了要给每个 I/O 设备分配一个中断请求号和设置一段相应的中断服务程序外，还要配置一个硬件电路——中断控制器（I/O 接口芯片）——专门管理多个 I/O 设备提出的中断请求。

当前，微型机处理中断通常采用“向量中断”方法。所谓向量即指中断服务程序的入口地址。由于存在多个中断请求，相应就有多个中断服务程序，即多个中断向量。为此，系统在内存特定区域安排一张中断向量表，专门存放所有的中断向量。换言之，一个中断请求（由中断号提供给 CPU）与该表内的一个中断向量建立一一对应关系。如在 PC 系列机中，一个向量代表的中断服务程序入口地址为 4 个字节（表示段地址和位移地址之形式），故而，中断号与中断向量有如下关系：

$$\text{中断向量} = [\text{中断号} \times 4]$$

方括号的含义是内存单元的内容（中断向量表刚好放在内存绝对地址 0 开始的位置）。

为有效地管理多个中断请求，在设计系统时，事先应按一定的原则将所有中断请求以优先级排队，并为每个中断请求级赋给相应的中断号。于是，当多个中断源同时提出中断请求时，应由中断判优及屏蔽逻辑选中当前最高优先级的中断请求，并屏蔽掉其余同级或优先级较低的中断请求，随后送出被选中的那个中断请求对应的中断号。上述接受中断请求及判优屏蔽的功能实际上是由中断控制器完成的。

一个完整的向量中断过程分成中断请求、中断判优及屏蔽、中断响应、中断处理和中断返回五个部分。图 1.3 描述了向量中断全过程。

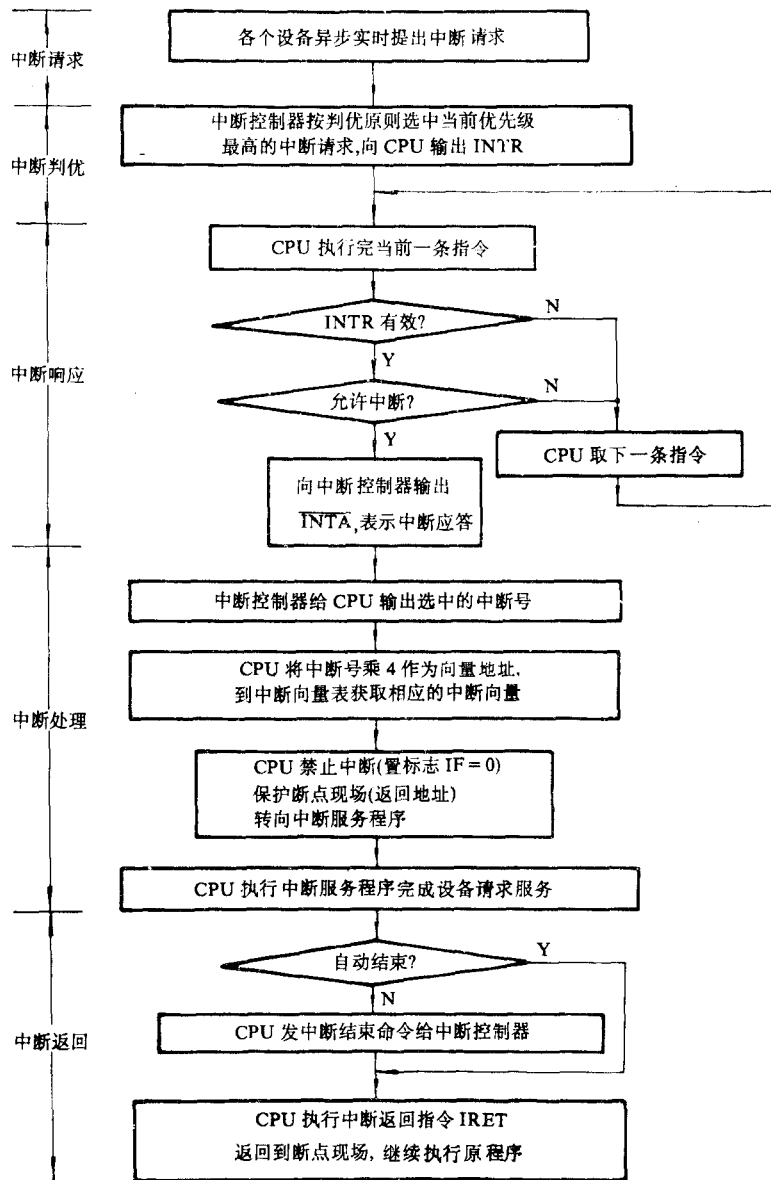


图 1.3 一个向量中断的全过程

根据图 1.3 流程，在中断处理中应注意下述几个问题：

1. 中断请求优先级的确定

在微型机中，通常按各个 I/O 设备实时处理所要求的轻重缓急程度确定中断请求优先级别的高低。如 PC/XT 机可处理 8 级中断请求，规定 0 级最高，1 级次之，7 级最低。它们分别对应于日时钟、键盘、保留、串行通信口 2、串行通信口 1、硬盘、软盘和打印机。

由于中断处理实际上是执行一段中断服务程序，所以中断程序同样可被其它中断请求所中断。这便是多重中断。一般，中断控制器内部都有一个中断服务寄存器（ISR）随时记载当前服务的中断请求级，故而，当出现多重中断时，通常要禁止再发生同级的或较低级别的中断（即屏蔽），但仍允许较高级别的中断响应。基于此，为保证每一个中断请求能较快地获得 CPU 的响应，要求中断服务程序尽可能短，以免干扰其它同级和低级中断设备的工作，甚至影响系统时钟的准确性。

2. 中断服务程序的编程

虽然中断全过程的前 3 个阶段是通过硬件实现的，仅使用几个机器周期，但是，中断过程的后两个阶段却要依靠程序执行，因而费时。对此，中断服务程序的编程要讲究效率。从图 1.3 可知，在执行中断服务程序之前，CPU 为保护好现场要先关闭中断，以允许高一级的中断产生，并保存所要使用到的寄存器（并非全部）；然后以尽可能快的速度处理完该中断（其因第 1 点中已叙述过）；最后，在程序结束执行中断返回指令 IRET 之前，若是非自动结束中断方式，则由 CPU 向中断控制器发出中断结束命令，即清除中断服务寄存器（ISR）当前正在服务的中断级，否则，将永远屏蔽掉与该级同级的和较低级的中断请求。

由以上分析可知，与程序查询方式相比，尽管中断处理方式使 CPU 的效率有很大提高，但也存在不足。这就是：为传送一个字符要执行一次中断服务程序。这不仅要启用一次中断控制器，而且，必须要完成保留现场和恢复现场的一套辅助性操作，但真正实现数据传送的指令可能只需少量几条。尤其当数据以成批方式与 I/O 设备交换时，要频繁地中断主程序，使大量的时间耗费在次要动作上，从而降低了系统的运行性能。

三、DMA 传输方式

DMA 是直接存储器存取的简称。它可满足高速 I/O 设备与 RAM 进行批量数据传送的需要。其特点是通过一个专门的硬件装置直接控制 I/O 设备与 RAM 的数据传输，而无需 CPU 介入。

由此可见，DMA 传输方式与程序查询和中断处理两种方式有本质上的不同，即用硬件代替软件实现数据的传输。这个专门的硬件电路称为 DMA 控制器。

在实现 DMA 传输时，是由 DMA 控制器直接掌管总线，因此，存在着一个总线控制权转移的问题，即 DMA 传输前，CPU 要把总线控制权交给 DMA 控制器，而在结束 DMA 传输后，DMA 控制器应立即把总线控制权再交回给 CPU。对此，一个完整的 DMA 传输过程必须经过下述 4 个步骤。

1. DMA 请求

CPU 对 DMA 控制器初始化，并向 I/O 接口发出操作命令，I/O 接口提出 DMA 请求。

2. DMA 响应

DMA 控制器对 DMA 请求判优及屏蔽，向总线裁决逻辑提出总线请求。当 CPU 执行完当前总线周期即可释放总线控制权。此时，总线裁决逻辑输出总线应答，表示 DMA 已经响应，通过 DMA 控制器通知 I/O 接口开始 DMA 传输。

3. DMA 传输