

微型计算机系统 原理及应用

(下册)

许 骏 编著



前 言

《计算机组成原理》、《汇编语言程序设计》和《微型计算机接口技术》是计算机应用专业三门重要的专业基础课，但内容重复和衔接不佳的现象比较严重。事实上，随着超大规模集成电路技术的进步，一度曾为大型机、中型机所特有的计算机系统结构技术正在迅速下移到高档微型计算机中，成为计算机专业人员必须掌握的基本知识。另一方面，随着C语言的普及，许多原来用汇编语言写的软件现在可以用C语言来编写，汇编语言通常仅用在能体现其速度和与硬件打交道能力的场合，例如接口驱动程序和实时控制软件。至于计算机接口设计，既要用到《计算机组成原理》的基本概念，也需要程序设计方法与技巧，显然是综合性的。本书尝试突破《计算机组成原理》、《汇编语言程序设计》和《微型计算机接口技术》三门课程的界限，使其内容有机结合、相互渗透，组成一门体系完整、结构合理、内容新颖的课程，取名为《微型计算机系统原理与应用》，全书共十二章，分上、下册出版。本书以与PC/AT兼容的386/486计算机系统为背景，全面介绍高档微型计算机组成原理及系统设计方法、汇编语言程序设计和接口技术。全书强调“系统”的概念，硬件与软件的紧密结合是贯穿全书的主线。本书的显著特色是注重能力的培养，这包括两个方面：一是对主流机型PC 386/486计算机系统应用开发的能力，即充分利用微机系统提供的软硬件资源，设计接口卡(板)及应用软件以组成满足应用要求的实际系统；二是以微处理器及其他芯片构成系统的能力，为此本书安排了单片计算机及其应用一章的内容。考虑到多媒体配置已逐步成为微型计算机系统的基本配置，本书增加了多媒体计算机系统一章。希望本书对促进计算机应用专业课程改革以及提高工科电子类专业微计算机技术教学水平有所裨益。

计算机技术属高新技术，其应用范围又触及社会的各个角落，作者在组织本书内容时，除注意选材的先进性、系统性和完整性外，还力求尽量缩短与实际工作的距离，做到既方便组织教学，又能直接指导应用。在具体写法上，力求做到内容精练、重点突出、概念清晰，特别注意避免手册式的繁琐叙述。书中给出的程序均全部在PC386/486计算机上调试通过，其中很多例子可供读者直接引用。

本书可用作计算机应用专业本、专科教材，也可作为通信、电子、自动化等专业本科生或研究生微计算机技术课的教材。对于从事计算机应用系统开发的科技人员也是一本合适的参考书。

本书是在作者编写的同名讲义的基础上，结合自己近年来的科研实践以及讲授该课程的经验，进行反复修改、更新和充实而写成的。在编写过程中，参考了大量国内近年出版的专著和教材，对这些文献的作者，在此一并表示感谢。

0/5 20/02

感谢促成本书出版的华南理工大学出版社。在本书的编写过程中，得到许多同事的热心帮助，黄惠青、谢敏菁完成了书稿的电脑录入和排版工作，陈力协助调试了书中的部分程序，他们为本书的出版付出了辛勤的劳动，在此谨表示最诚挚的谢意！

由于作者水平有限，加上时间仓促，不妥之处或错误在所难免，恳请同行和读者批评指正，谢谢。

编著者

1996年2月

目 录

第七章 输入输出系统(I/O).....	1
7.1 I/O 系统概述.....	1
一、主机和外设的连接方式	1
二、接口的概念.....	2
三、接口的分类.....	3
7.2 I/O 端口寻址与地址译码方法.....	5
一、I/O 端口寻址	5
二、I/O 端口的地址译码方法	7
7.3 程序查询方式及其接口	10
一、程序查询方式	10
二、查询式输入接口	11
三、查询式输出接口	12
7.4 中断系统	14
一、中断的基本概念	14
二、中断请求	16
三、中断源识别与中断优先级	18
四、PC/AT 的外中断与 8259A	20
五、中断响应	35
六、中断处理	39
7.5 DMA 传送方式及其接口.....	42
一、基本原理	42
二、DMA 控制器 8237A-5	44
三、应用举例	51
7.6 A/D 和 D/A 转换器与 CPU 接口技术.....	53
一、D/A 转换器与 CPU 的接口技术	54
二、A/D 转换器与 CPU 的接口技术	61
7.7 可编程并行接口芯片 8255A 及其应用	70
一、可编程并行接口芯片 8255A 的组成与功能.....	70

二、8255A 的工作方式及应用举例	72
7.8 总线技术	84
一、总线的一般概念	84
二、微计算机系统常用总线	86
第八章 串行通信接口	94
8.1 串行通信的基本概念	94
一、数据的串行传输	94
二、异步传输	97
三、同步传输	100
四、串行通信的差错控制	102
五、信号的调制与解调	104
六、异步串行通信标准接口 RS-232C	107
8.2 PC 386/486 异步通信接口	110
一、异步串行通信接口逻辑	111
二、INS 8250/NS16450 内部寄存器的编程方法	114
三、PC 386/486 计算机之间通信程序设计举例	124
四、PC 386/486 的串行异步通信程序简介	132
第九章 人机交互设备接口	136
9.1 概述	136
9.2 键盘及其接口	137
一、非编码键盘设计	138
二、IBM PC 计算机键盘及接口	143
9.3 LED 显示器接口	159
一、LED 显示器及显示原理	159
二、LED 显示器与 CPU 接口	160
9.4 CRT 显示及其接口	163
一、CRT 显示原理	163
二、VGA/TVGA 显示系统原理	166
三、VGA/TVGA 硬件直接程序设计	187
四、VGA/TVGA BIOS INT 10H 调用程序设计	198
五、以 PC 386/486 计算机为核心的数据采集系统	205

第十章 定时/计数技术	212
10.1 可编程定时/计数器 8253-5(8254)	212
10.2 8253-5 在 PC/AT 计算机中的应用	218
10.3 8253-5 在数据采集和实时控制系统中的应用	226
第十一章 单片计算机及其应用	230
11.1 概述	230
11.2 MCS-51 单片计算机体系结构	231
一、CPU	232
二、片内数据存储器	233
三、并行 I/O 接口	239
四、复位状态及复位电路	243
五、程序存储器结构及运行操作	243
11.3 片外 RAM 及并行 I/O 口扩展	245
一、片外数据存储器及其数据操作	245
二、并行 I/O 口扩展	247
11.4 MCS-51 单片计算机的中断系统	252
一、中断源	253
二、中断系统的控制	253
三、中断的响应过程	255
四、应用举例	256
11.5 MCS-51 单片计算机的定时/计数器	261
一、定时/计数器的结构	261
二、定时/计数器的方式和控制寄存器	262
三、定时/计数器的工作方式	263
四、定时/计数器的编程和应用举例	265
11.6 MCS-51 单片计算机串行通信接口	269
一、串行口控制寄存器	270
二、串行口的工作方式	271
11.7 单片计算机应用系统设计	274
一、应用系统设计的一般问题	274
二、应用系统设计举例	274

11.8 PC486 计算机与 MCS-51 单片机的数据通信	280
一、系统结构与功能	280
二、通信协议	282
三、程序设计	283
四、PC486 计算机通信程序清单	286
五、单片计算机通信程序清单	294
 第十二章 多媒体计算机系统	299
12.1 多媒体技术概论	299
一、引言	299
二、音频/视频信息数字化	300
12.2 PC 升级到 MPC	304
一、CD-ROM 驱动器	304
二、加入音频功能	306
三、在 PC 机上播放影视节目	308
四、处理视频	309
五、触摸屏技术	310
附录 MCS-51 指令表	313
参考文献	318

第七章 输入输出系统(I/O)

现代计算机的外部设备种类越来越多, 各类设备的组成和工作原理差异很大, 与计算机之间的连接和传输数据的方式也很不相同, 因此, 计算机的输入输出控制和输入输出子系统(简称I/O系统)就成为整个计算机系统中最具有多样性和复杂性的部分。

7.1 I/O 系统概述

一、主机和外设的连接方式

主机和外设的连接方式大致可以分为三类, 如图7-1所示。

图7-1a中, 主机和各外设之间有单独的数据通路, 称为辐射型连接。它的优点是控制简单, 但结构复杂、连线多, 尤其是扩充外设很麻烦。

图7-1b中, 主机通过一组总线与外设连接, 各外设通过集电极开路门或三态门挂载在总线上。这种连接方式结构简单且易于扩展, 各外设之间也有可能通过总线直接通信。其缺点是所有的外设都通过同一组总线分时工作, 由于信息吞吐量有限, 将影响信息交换速度。这种结构广泛用于微、小型计算机中。

图7-1c的连接方式可看成上述两种方式的结合型。主机通过“通道”管理输入/输出操作, 主机与通道间采用辐射型连接, 通道和外设之间则通过总线相连。

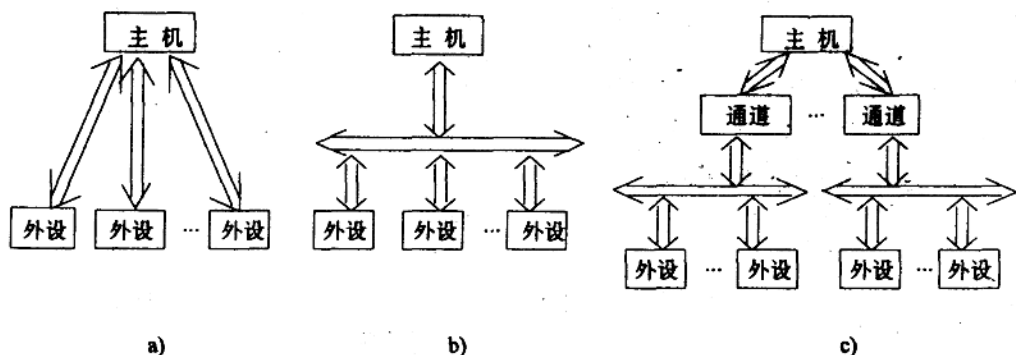


图7-1 主机与外设的连接方式

二、接口的概念

不论何种设备以何种方式与主机相连,其功能都是要实现计算机与外部世界的联系,其关键是两者之间的信息交换。然而,外部设备是多种多样的,常用的就有键盘、鼠标、触摸屏、CRT显示器、磁盘机、打印机、绘图仪、模/数(A/D)和数/模(D/A)转换器、扫描仪以及调制解调器(MODEM)等等,这些设备的结构与工作原理各不相同,涉及光、电、磁及机械传动等等。它们在工作速度、使用的数据格式等方面与主机不同,一般不能直接与主机相连。主机与外设的连接通常都要通过接口(Interface)实现,这便是I/O接口,如图7-2所示。I/O接口是主机和外设之间的交接界面,是CPU与外界各种检测、控制对象联络的纽带和桥梁。在下面的讨论中,不严格区分主机与CPU的概念,输入或输出是以CPU为中心确定的。

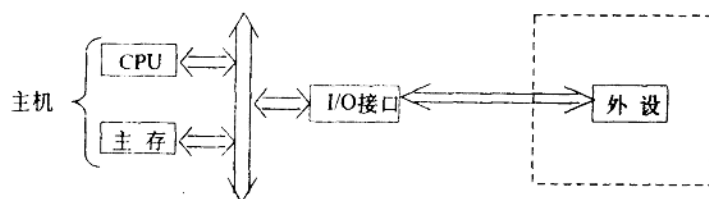


图7-2 I/O系统组成

通常, I/O接口应解决如下几个问题。

(1) 协调CPU与外部设备之间在数据传送速度上的差异。为此,接口部件的输出端口一般包含数据锁存器,而输入端口一般为三态缓冲器,如图7-3所示。CPU向外设输出数据时,由于数据在总线上停留的时间很短,必须采用数据锁存技术,利用I/O译码信号及I/O写控制信号 $\overline{IOW}$ 作为触发脉冲,把数据线上的状态及时地锁存到锁存器中,CPU写完此数据后便可以继续进行其他操作,慢速外设在需要时再把锁存器中的数据取走。外设向CPU输入数据时,外设把已准备好的数据送到三态缓冲器的输入端,再通知CPU读取。事实上,采用三态门也是多个外设分时使用数据总线的需要。

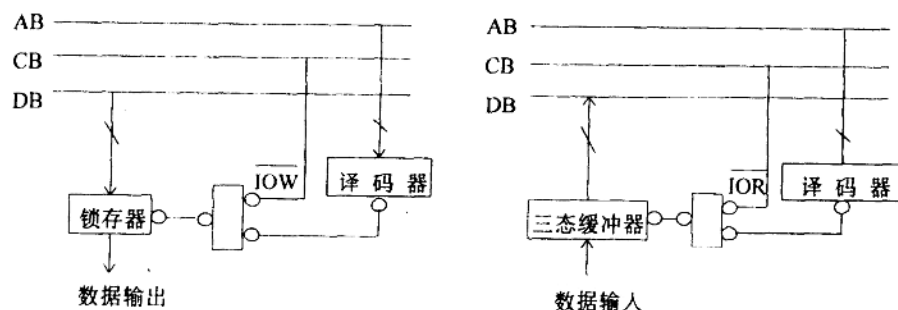


图7-3 数据缓冲原理

(2) CPU与外设是异步工作的,它们有各自独立的定时系统,在时间上是不同步的。对于一个要向计算机输入数据的输入设备(例如A/D转换器),CPU不能准确知道它什么时候已把要输入的数据准备好了,这个输入设备也不知道什么时候CPU已取走了该数据而应该开始准备下一个输入数据。对于一个接收CPU输出数据的输出设备(如打印机),CPU不能准确知道什么时候这个输出设备已经做好了接收数据的准备,反之,输出设备也不能准确知道什么时候CPU输出了新的数据。因此,为了可靠而有效地传输数据,CPU与外设之间需要互相提供联络信息。联络信息可以理解为状态,如向接收方提供“数据已准备好”的状态信息,请对方取走数据。联络信息也可以理解为命令,如“启动一次A/D转换”。实际上,理解为状态或命令并没有严格的区别。如何向对方提供联络信息、以何种方式响应和处理对方提供的联络信息,是I/O接口设计要解决的最基本问题之一。此外,计算机系统中往往有多个外设,这就有可能出现两个以上外设同时要求与CPU交换信息的情况,此时要根据某种策略决定首先为哪个外设服务。这是I/O接口设计要解决的另一个基本问题。

在微型计算机系统中,通常有三种基本的输入输出方式,即程序查询方式、程序中断方式和DMA传送方式。不同的输入输出方式在解决上述两个基本问题时各具特点,这也是本章要重点讨论的内容。

(3) 提供I/O设备寻址信息,使CPU能从众多的外设中选择其中之一与之进行数据交换。

(4) 进行数据格式的变换。模拟接口把外设送来的模拟信号变换成CPU能接受的数字信号(A/D接口),CPU向模拟设备输出信息时也必须进行变换,只是变换的方向相反而已(D/A接口);串行和并行数据格式的变换也是常见的(如串行通信接口);在LED或CRT显示接口中要进行字符码到字形码的代码变换。进行信号电平变换有时也是需要的。

接口的输入输出操作必须要有软件的支持,因此,接口还包括软件。用微计算机组成一个应用系统的关键技术是接口技术,而接口的研制与设计既有硬件电路设计的内容,也要为接口编写相应的驱动程序。随着各种大规模可编程接口芯片的出现与普及,I/O接口软件的地位愈加重要。

三、接口的分类

接口通常是以外设设备输入输出的信息特征进行分类。按照信息的形式分为数字接口和模拟接口,本章7.6节将专门讨论模拟接口。对数字接口,可按数据传送宽度的不同分为串行接口和并行接口。这里所说的数据传送宽度指外设和接口之间的传送宽度,而在主机(CPU)和接口一侧,数据总是并行传送的。并行接口中外设和接口之间一次同时传送一个字节(字)的所有位,传送速度快,但传输线的数目随着一个字的位数增多而增加。串行接口中外设和接口之间的数据是逐位串行传送的,第八章将专门讨论这种接口。此外,根据接口的使用范围,可分为通用接口和专用接口,通用接口是指功能上通用的一类接口,如并行I/O接口、串行I/O接口等;专用接口通常是为一特定外部设备专用的,如软盘控制器、CRT控制器、键盘接口控制器等。

由于接口电路的设计比较复杂,同时考虑到同一类接口有许多要求是相似的,所以在微处理机一开始出现,就有大规模集成接口电路芯片与之配套。绝大部分接口芯片都可以由CPU写入适当的控制字来改变其工作方式或工作参数,这称为可编程。这一特性使接口

芯片的适用范围得到扩充。使用这类芯片时，首先要用程序设定接口芯片的工作方式和参数，这就是所谓的初始化编程。不同功能的接口芯片，其结构虽各有不同，但都是由寄存器和控制逻辑两大部分组成的，如图7-4所示。

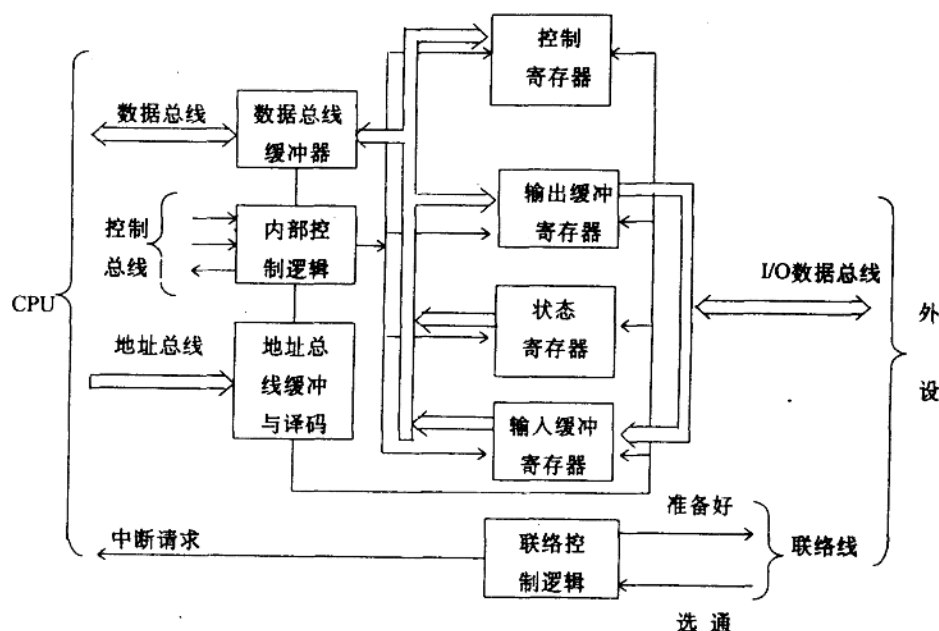


图7-4 接口芯片基本结构框图

(1) 数据缓冲寄存器 包括输入缓冲寄存器和输出缓冲寄存器，前者应具有三态功能，当它没有被选中时处于高阻态，后者则要有数据锁存能力，这是I/O接口的基本结构特点之一。

(2) 控制寄存器 用来存放CPU发来的控制命令和其他信息，以确定接口的工作方式和功能，这是可编程接口芯片所必需的。

(3) 状态寄存器 用于保存外设当前的各种状态信息。当CPU以程序查询方式同外设交换信息时，该寄存器是必不可少的。

以上三种寄存器构成接口芯片的核心部分。

(4) 端口地址译码器 用来正确选择接口芯片内部各端口寄存器。

(5) 对外联络控制逻辑 主要有面向CPU一侧的中断请求和响应信号，面向外设一侧的准备就绪和选通等控制与应答信号。此外，还有数据总线和地址总线缓冲器以及内部控制逻辑等，它们分别用于接口芯片与CPU总线(DB、AB、CB)的连接。

并非所有接口芯片都具备上述全部内容。

7.2 I/O 端口寻址与地址译码方法

一、I/O端口寻址

所谓端口(Port)是指I/O接口(接口芯片或接口卡)中供CPU直接存取访问的寄存器或某些特定的硬件电路。一个I/O接口总包括若干个端口,除常见的数据端口、控制命令端口和状态端口外,还有特殊用途的端口如方式控制端口、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,这将在以后各章节中详尽叙述。但应该指出,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)、只写(读出无意义,如方式控制)等属性,这也是在设计I/O接口功能时决定的。

既然端口可被CPU访问,这就有一个寻址的问题。在当今流行的各类微型计算机中,对端口有两种编址方法——独立的I/O编址和存储器映象编址。

1. 独立的I/O编址方式

这种方式有一个与存储器空间完全独立的I/O地址空间, I/O端口与存储器各自编址。对同一地址编号, CPU是寻址I/O端口还是寻址存储器呢?为此,要作以下约定:

(1) CPU设置两组读/写控制信号——存储器读/写信号 $\overline{\text{MEMR}}/\overline{\text{MEMW}}$ 和I/O读写信号 $\overline{\text{IOR}}/\overline{\text{IOW}}$, 如图7-5所示。

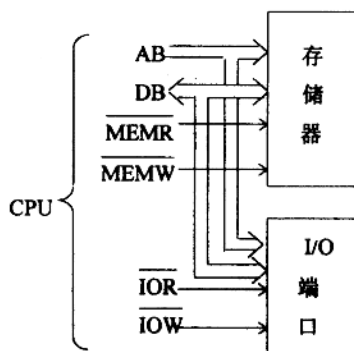


图7-5 CPU设置的两组读/写控制信号

(2) 在指令系统中设置专用I/O指令——输入指令IN和输出指令OUT。

I/O独立编址方式具有以下优点:

- 1) I/O端口不占用存储器空间。
- 2) 程序设计时,容易把I/O指令与存储器访问指令区分开。
- 3) I/O指令通常只需2个字节,寻址速度较快。

PC系列计算机由1024个I/O端口地址组成一个独立的I/O空间(000H~3FFH)。其中低端512个端口地址(000H~1FFH)供系统板电路使用,高端512个端口地址(200H~3FFH)供扩

展插槽使用,一般用户可以使用300H~31FH地址。所有连接在系统地址总线上的I/O设备使用低10位地址线($A_9 \sim A_0$),用户在设计I/O接口卡时,一定要使端口地址译码电路的 $A_9=1$,以免发生I/O端口地址重叠和冲突。I/O设备的读/写控制使用 $\overline{IOR}$ 和 $\overline{IOW}$ 信号(由总线控制部件或DMA控制器提供)。

8086/8088 CPU的I/O指令仅支持I/O设备与累加器(AL/AX)之间的数据交换,其中AX对应地址相邻的两个8位I/O地址端口。

输入指令: IN AL/AX, PORT ; $AL \leftarrow (PORT) / AX \leftarrow (PORT+1: PORT)$

或 MOV DX, PORT

IN AL/AX, DX

输出指令: OUT PORT, AL/AX

或 MOV DX, PORT

OUT DX, AL/AX ; $(PORT) \leftarrow AL / (PORT+1: PORT) \leftarrow AX$

I/O端口寻址可以是直接寻址或寄存器间接寻址。前者在指令中直接写上8位的端口地址PORT,这种方法只能寻址0~255(0FFH);后者使用16位寄存器DX放端口地址,寻址范围可达64KB。

80386/486 CPU允许I/O设备与内存之间直接进行数据传输。

输入指令: MOV DX, PORT

LEA DI, BUFFER_IN

INSB/INSW

输出指令: MOV DX, PORT

LEA SI, BUFFER_OUT

OUTSB/OUTSW

如果在INS和OUTS指令前使用重复前缀REP,可实现I/O设备与内存RAM之间的成批数据交换。PC/AT计算机系统的硬盘I/O控制程序(INT 13H)就是用这种方式实现硬盘扇区的读写操作。

2. 存储器映象方式

存储器映象方式将I/O端口和存储单元同等看待,一起编址,即I/O端口地址空间是存储器地址空间的一部分,CPU用对存储单元的读/写操作来完成I/O端口的输入/输出。CPU访问存储单元和I/O端口(实质都是存储单元)统一使用存储器读/写信号 $\overline{MEMR} / \overline{MEMW}$,两者的区分是通过地址译码实现的。Motorola公司生产的MC6800/68000系列、6502系列CPU以及INTEL公司的MCS-51系列单片计算机就是采用这种I/O编址方式。图7-6是存储器映象方式I/O编址的示意图。 $A_{15}=0$ 时, $A_{14} \sim A_0$ 用于寻址存储单元, $A_{15}=1$ 时由 $A_{14} \sim A_0$ 寻址I/O端口,显然,整个存储空间被分为两半。

存储器映象方式编址的主要优点是:①对I/O端口与对存储器的操作完全相同,由于存储器操作指令较丰富,大大增强了系统的I/O功能,使得访问外设I/O端口的操作更方便灵活;②扩大了I/O端口的寻址范围(I/O端口数目只受总存储器容量的限制),其主要缺点是:占用了存储器的部分地址空间,使可用的内存空间减少了。

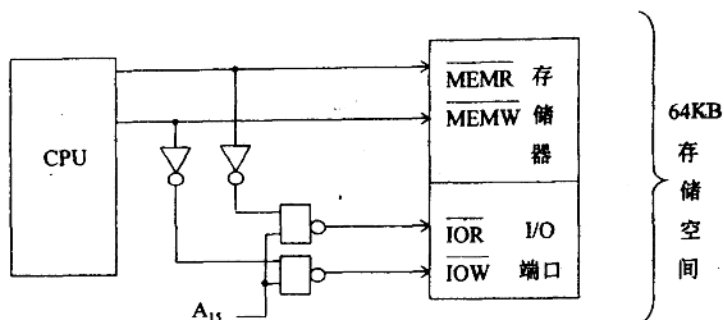


图7-6 存储器统一的I/O编址连接

二、I/O端口的地址译码方法

CPU要对I/O端口进行读/写操作，首先要确定与自己交换信息的端口地址。如何实现由CPU发来的地址码识别和确认I/O端口，这就是所谓I/O地址译码问题。译码方法灵活多样，一般由地址和控制信号($\overline{\text{IOR}}$ / $\overline{\text{IOW}}$ ，AEN等)的不同组合实现。译码电路采用的元器件通常有门电路和译码器，也有采用GAL或PAL器件进行译码的。

当仅需一个口地址时，采用门电路构成译码电路很简便。图7-7所示的电路，能译出进行读/写操作的端口地址2C7H(低电平有效)。图中用AEN参加译码，只有当AEN=0时，即不是DMA操作时译码才有效，从而避免在DMA周期由DMA控制器对该I/O口地址选中的外部设备进行读/写操作。

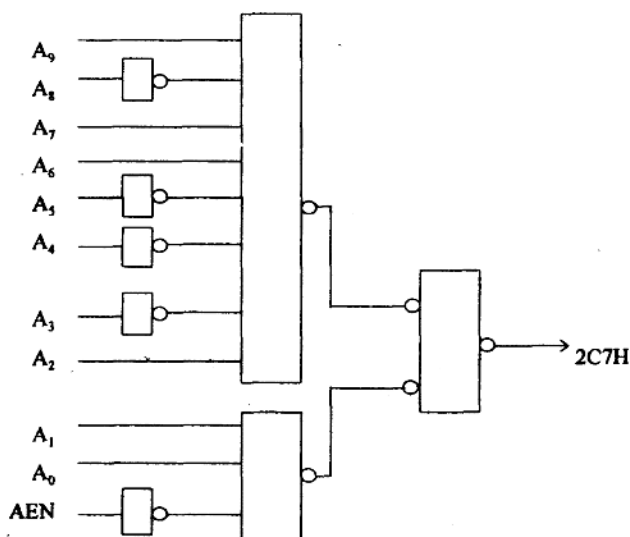


图7-7 由门电路构成的译码电路

微计算机系统通常使用多个接口芯片，每个芯片都有片选信号端 $\overline{CS}$ ，且内部含有多个寄存器。这时I/O端口译码应包括两个方面：

(1) 外部译码，产生片选信号。一般采用译码器芯片实现。图7-8是PC机系统板上接口芯片的端口地址译码电路，图中由地址线 $A_9 \sim A_5$ 译码，产生 $\overline{DMACS}$ (8237)、 $\overline{INTRCS}$ (8259)、 $\overline{T/C CS}$ (8253)、 $\overline{PPICS}$ (8255A)等片选信号，地址总线的低5位 $A_4 \sim A_0$ 在芯片内部进行局部译码以选择片内寄存器。很明显，8237A芯片的端口地址范围是000 ~ 01FH，8259A芯片的端口地址范围是020H ~ 03FH等等。这些端口地址我们以后会经常接触到。

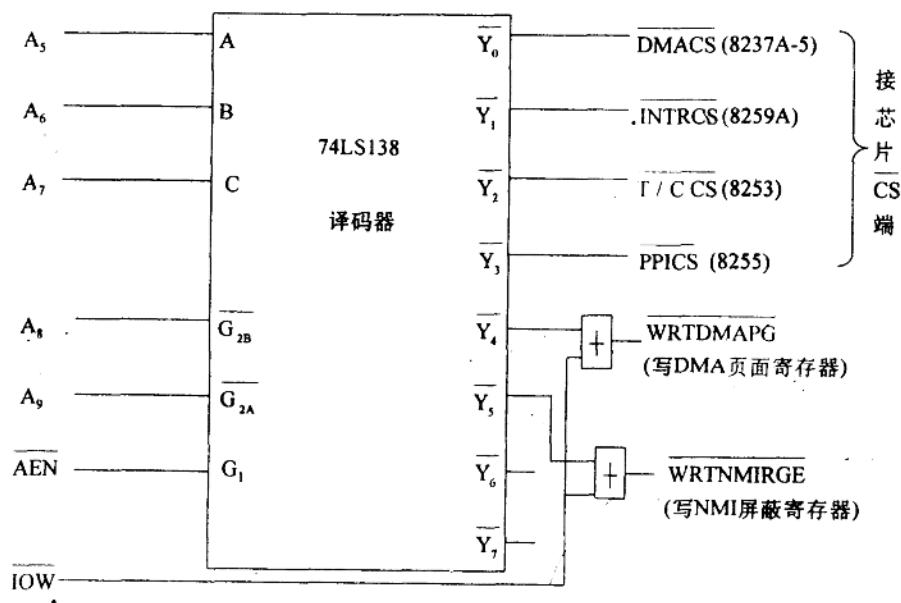


图7-8 系统板上接口芯片的片选译码电路

(2) 片选信号 $\overline{CS}$ 确定了一个接口芯片的I/O端口地址范围，芯片内各个寄存器的寻址是由芯片内部的逻辑电路完成的。

常用的内部译码有以下几种形式：

1. 一般全译码电路

通常，一个芯片内各寄存器的端口地址采用连续编址的方式，因此用若干条最低地址线就可以区分它们。比如，对一个包含四个端口地址的接口芯片，使用 A_1A_0 两条最低位地址线译码即可。

2. 计数译码电路

对接口芯片内各寄存器的寻址规定严格的先后次序，每执行一次I/O写入操作都同时对一个计数器计数，然后对计数器的计数值译码来保证正确的写入顺序。例如：假定芯片内共有四个寄存器 $R_0 \sim R_3$ ，设置一个两位的二进制计数器，系统复位后计数器的状态为00，经译码后使 R_0 的控制门打开，如图7-9所示。当CPU对选中的接口芯片(片选 $\overline{CS}$ 信号有效)

写入数据时，首先写入到 R_0 中，同时使计数器加1。下一次CPU对同一地址进行的写入操作，则使数据进入 R_1 ，其他寄存器依次类推。

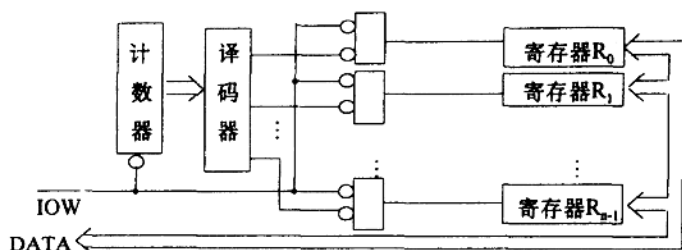


图7-9 计数译码器

这种方式工作的接口芯片只占用一个I/O地址，而不管它内部有几个寄存器。在对这种接口芯片进行初始化编程时，一定要注意写入的先后顺序问题。

3. 命令字中携带地址信息

在一些较复杂的接口电路中，可能有很多种工作方式供选择，因此CPU要把各种工作方式的代码放到不同的寄存器中。为了节省地址资源，有的接口芯片利用命令字中的一些特定位携带地址信息，图7-10给出一个可能的例子。CPU写入的命令字中，规定八位数据中的第4、5位为地址信息，用以指定该命令字写入的寄存器；片内电路从命令字中引出第4、5位，作为译码器的输入信号，而译码输出作为片内寄存器的选择控制信号。

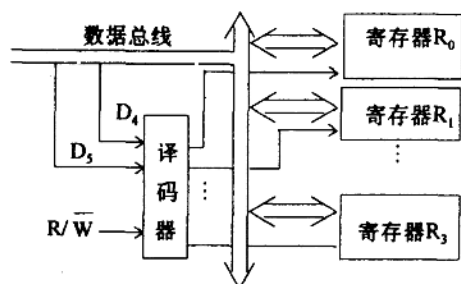


图7-10 命令字包含地址信息的片内寻址

在对这类接口芯片进行初始化编程时，写入的控制字的某些位要被指定为0或1。

4. 用专用数据字节指定寄存器地址

前面的三种方法在寄存器数量比较少的情况下是适用的，如果寄存器的数量比较多，就不太合适了。于是提出用一个数据字节来指定内部寄存器地址的方法，图7-11是这种方法的原理图。图中设置一个触发器FF，它的初始态使地址寄存器打开，可以接收数据线上的数据；当CPU第一次往接口芯片写数据时，一方面将这个数据送入地址寄存器，另一方面使触发器FF的状态翻转，让寄存器 $R_0 \sim R_{n-1}$ 处于允许接收数据的状态；CPU第二次对接口芯片进行写操作时，数据写入到由地址寄存器内容经译码后指定的寄存器中，写入结束时，触发器状态又一次翻转。

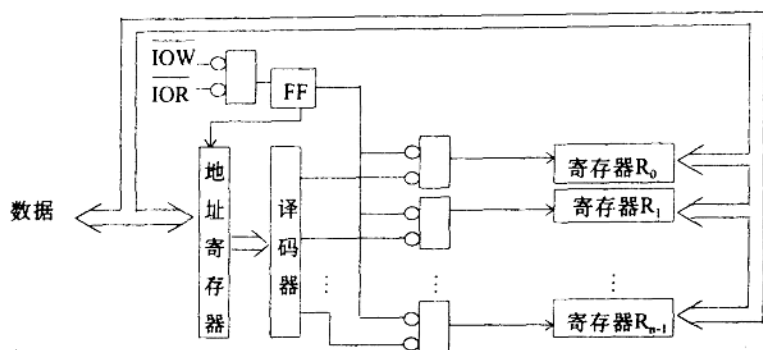


图7-11 用数据字节指定地址的片内寻址

以这种方式工作的接口芯片，虽然片内寄存器数量很多，但只需占用一个I/O口地址。它以先写地址后写数据的方式有效地节省了地址资源。当然也可以让地址寄存器单独占用一个地址，全部寄存器 $R_0 \sim R_{n-1}$ 共用一个地址，如VGA/TVGA视频系统的寄存器组就采用这种方法寻址。

本书后面各章在介绍微型计算机系统设计方法时，要用到各种接口芯片，读者可以从对这些接口芯片的操作，特别是对它的初始化编程中体会这4种内部译码方式的特点及使用方法。有的接口芯片的内部译码逻辑还可能同时采用几种方法，以达到既节省地址资源又便于使用的目的。

7.3 程序查询方式及其接口

一、程序查询方式

程序查询方式是通过利用程序检查外设的状态来协调CPU与外设之间的速度差异，实现CPU与外设之间可靠传送数据的一种方法。其工作过程可用图7-12表示，程序查询的核心是图中虚线框出部分。一个数据传送过程由三个环节组成。

①CPU从接口读取状态字；

②CPU检测状态字的对应位是否满足“就绪”条件，如果不满足，则返回第一步继续读状态字，以等待条件的满足；

③如果状态字表明外设已处于“就绪”状态，则传送数据。

图7-13给出了程序查询方式接口的示意图。

查询式接口至少要有两个寄存器，一个是数据缓冲寄存器，用来存放与CPU进行传送的数据信息；另一个是供CPU查询的I/O设备状态寄存器，这个寄存器可以由多个标志位组成，但最重要的是外设准备就绪标志(输入和输出设备的准备就绪位可以不是同一位)，当CPU得到该标志状态后就进行判断，以决定是继续等待还是进行一次I/O传送操作。在一些简单的接口电路中，可能只设置状态标志触发器来反映外设是否已准备就绪。