

HUAZHONG UNIVERSITY OF SCIENCE AND TECHNOLOGY PRESS



微型计算机 接口技术及应用

WEIXINGJISUANJI JIEKOUJISHU JI YINGYONG

刘乐善 叶济忠 胡盛斌 编著

华中理工大学出版社

前 言

随着工业生产的现代化和人民生活水平的不断提高,对于工业与民用产品的要求也越来越高,尤其对多功能化、智能化方面的要求更为突出。应用单片机是满足与实现这一要求的最佳方案。在国外,目前单片机的年产量数以亿计,其数量之大,应用之广,是任何其它类型计算机所不可比拟的。

单片机进入我国市场的时间还不足十年,但已被各个领域的工程技术人员所接受,并应用于各个领域之中。尤其在智能仪表、工业控制器、家用电器领域中,更受到青睐。

人们最初接触到的是少数几个品种规格的单片机芯片,它们具有 4~8 位 CPU,少量存储器和输出输入接口。随着单片机技术的发展,越来越多的单片机芯片增强了通讯接口的功能,从而使单片机的最佳工作方式不仅仅限于单片或最小扩展方式,还具有总线通讯扩展方式。单片机已成为系统控制家族中的一员。

当前,各大公司生产的单片机芯片规格各异,总线接口硬、软件的联接与约定也不尽相同。本书的主要目的在于把当前常见、应用较广的几种单片机总线型式介绍给读者,并对其应用作典型说明。如飞利浦公司的 I²C 总线,英特尔公司的位总线,美国国家半导体公司的 MICROWIRE 总线和莫托罗拉公司的 68 系列单片机 SPI 总线。

通过总线通讯扩展,使分散的单片机控制设备非常方便地构成网络 and 控制系统,而且还可以和其它微机系统构成多级控制系统及管理与控制相结合的综合系统。由此更进一步扩大了单片机的应用范围。

另一个方面,由于用户系统往往需要扩充或变更部分功能以形成新的系统。在这种情况下,选择相适应的单片机品种、规格来变更系统是最佳的技术方案,但往往受条件、资金等方面的制约而不能实现。为此,通过总线扩展来增强部分功能是当前国内广大单片机用户需要采用的必然手段;而当单片机用于系统控制之中时,采用标准的总线接口则给进一步扩展带来许多方便。据此,本书也以一定的篇幅介绍了 STD 总线、PC 总线、FD 总线和常用的并行与串行标准接口。

由于作者水平与能力所限,错误与疏漏在所难免,谨希广大读者见谅,并望提出宝贵意见。

本书由天津纺织工学院尤一鸣、傅景义、王俊省同志编写,北京航空航天大学何立民同志审阅了全书,天津纺织工学院的赵可萍、王金海同志参加了书中应用实例的研制与调试工作。

编 者

一九九二年十月

目 录

第一章 微型计算机接口技术概述	(1)
1.1 接口及接口的功能	(1)
1.2 接口技术的发展简述	(3)
1.3 CPU 与接口之间传送信息的方式	(3)
1.4 设计与分析接口电路的基本方法	(4)
1.5 外围接口芯片	(5)
习题一	(6)
第二章 I/O 接口地址译码技术	(7)
2.1 I/O 接口的寻址方式	(7)
2.2 系统对 I/O 接口地址的分配	(8)
2.3 I/O 接口地址译码方法	(8)
习题二	(11)
第三章 总线技术	(13)
3.1 概 述	(13)
3.1.1 总线的分类	(14)
3.1.2 总线的数据传输	(14)
3.2 IBM-PC 总线	(17)
3.2.1 IBM PC 总线标准	(17)
3.2.2 系统总线周期时序	(21)
3.3 STD 总线	(21)
3.3.1 STD 总线的特点	(21)
3.3.2 STD 总线的规范	(22)
3.3.3 STD 应用实例	(25)
3.4 几种常用的总线标准	(28)
3.4.1 MULTIBUS 总线	(28)
3.4.2 VME 总线	(31)
3.4.3 RS-232C 总线	(33)
3.4.4 IEEE488 总线接口(即 GP-IB 总线)	(34)
习题三	(35)
第四章 中断技术	(36)
4.1 中断的基本概念	(36)
4.1.1 中断	(36)
4.1.2 中断过程	(36)
4.1.3 中断源及其优先级	(37)
4.1.4 中断处理的隐操作及堆栈的使用	(37)
4.1.5 中断向量	(38)

4.2	8088/8086 的中断系统	(40)
4.2.1	外部中断	(41)
4.2.2	内部中断	(42)
4.3	8259A 可编程中断控制器	(43)
4.3.1	8259A 的外部特性和内部结构	(43)
4.3.2	8259A 的中断操作功能及其实现方法	(45)
4.4	8259A 在微机系统中的使用	(57)
4.4.1	8259A 与 CPU 的硬件连接	(57)
4.4.2	8259A 的软件编程	(57)
4.5	可屏蔽硬件中断请求线的扩充	(59)
4.5.1	中断请求线的扩充方法	(59)
4.5.2	扩充的中断系统请求中断的过程	(60)
4.5.3	有扩充中断请求线的中断程序举例	(60)
	习题四	(62)
第五章	DMA 技术	(63)
5.1	DMA 控制器	(63)
5.1.1	8237A 的外部特性	(63)
5.1.2	8237A 内部寄存器及编程方法	(64)
5.1.3	DMA 控制器的工作时序	(69)
5.2	DMA 控制器在系统中的使用	(71)
5.2.1	DMA 控制器在系统中的地位	(72)
5.2.2	DMA 控制器在微机系统中有效地址的生成	(72)
5.2.3	DMA 控制器 8237A 的初始化	(75)
5.3	DMA 控制器的应用举例	(76)
5.3.1	DMA 控制器与 I/O 设备的连接	(76)
5.3.2	DMA 控制器的软件编程	(77)
	习题五	(79)
第六章	定时/计数技术	(80)
6.1	概 述	(80)
6.2	可编程定时/计数器 8253(8254)	(81)
6.2.1	8253 的工作原理	(81)
6.2.2	8253 的工作方式	(83)
6.2.3	8253(8254)的编程	(88)
6.3	8253 在系统中的应用	(90)
	习题六	(92)
第七章	存储器接口	(93)
7.1	存储器和存储器组织	(93)
7.1.1	半导体存储器的分类	(93)
7.1.2	存储器组织	(94)
7.2	存储器地址空间寻址方法	(94)
7.2.1	RAM 与 CPU 的连接	(94)
7.2.2	存储器地址译码	(95)

7.3 存储器接口	(97)
7.3.1 存储器子系统的基本器件	(98)
7.3.2 RAM 子系统	(103)
习题七	(107)
第八章 并行接口	(108)
8.1 并行接口概念	(108)
8.2 硬线连接并行接口	(108)
8.2.1 简单的并行输入/输出接口	(108)
8.2.2 带有应答信号的输入/输出接口	(110)
8.2.3 采用中断方式的输入/输出接口	(111)
8.3 可编程并行接口	(112)
8.3.1 8255A 工作原理及编程命令	(112)
8.3.2 8255A 的 0 方式及连接方法	(117)
8.3.3 8255A 的 1 方式及连接方法	(120)
8.3.4 8255A 的 2 方式及连接方法	(125)
8.4 利用 PC/XT、AT 的打印机适配器作非打印机并行设备接口	(129)
8.4.1 打印机适配器	(129)
8.4.2 利用打印机并行接口进行 PC/XT 机与 TP86 单板机的并行通信	(131)
习题八	(134)
第九章 人-机接口	(135)
9.1 概 述	(135)
9.2 键盘及接口电路	(135)
9.2.1 微型机与非编码键盘的接口	(135)
9.2.2 8279 键盘接口芯片	(137)
9.2.3 PC 机的键盘接口	(142)
9.3 LED 显示器及接口	(145)
9.4 CRT 显示器接口	(147)
9.4.1 CRT 显示器控制器电路	(147)
9.4.2 CRT 控制器接口芯片	(148)
9.5 计算机的图形接口	(148)
9.5.1 彩色适配器(CGA)电路	(148)
9.5.2 PC 机彩色适配器 CGA 的编程及功能调用	(153)
9.6 几种常见的视屏标准	(158)
9.7 交互式人-机接口	(159)
9.7.1 鼠标器	(159)
9.7.2 光 笔	(162)
9.7.3 操纵杆	(163)
9.8 汉字输入与输出	(164)
9.8.1 汉字代码	(165)
9.8.2 汉字输入	(165)
9.8.3 汉字输出	(165)
习题九	(166)

第十章 串行通信接口	(167)
10.1 串行通信的基本概念	(167)
10.1.1 串行通信的特点及通信方式	(167)
10.1.2 数据传送模式	(169)
10.1.3 信号的调制与解调	(170)
10.1.4 信息的检错与纠错	(171)
10.1.5 传输速率与传送距离	(171)
10.2 串行接口标准	(172)
10.2.1 RS-232C 接口标准	(172)
10.2.2 20mA 电流环标准	(176)
10.3 串行通信接口	(178)
10.3.1 串行通信接口的基本任务	(178)
10.3.2 用 8251 组成的串行接口电路	(178)
10.3.3 8251 的内部逻辑与外部信号	(178)
10.3.4 8251 的控制字和状态字	(181)
10.3.5 8251 应用举例	(184)
10.4 IBM-PC/XT 异步通信适配器的分析和使用	(187)
10.4.1 异步通信适配器的组成	(187)
10.4.2 通用异步通信收发器 INS8250 的外部特性	(188)
10.4.3 8250 内部寄存器及其编程方法	(189)
10.4.4 两台 IBM-PC 微机之间的串行通信	(194)
10.5 异步串行通信 I/O 功能调用	(197)
10.5.1 DOS 串行通信口功能调用	(197)
10.5.2 BIOS 串行通信口功能调用	(198)
10.5.3 异步通信 I/O 功能调用举例	(200)
习题十	(204)
第十一章 D/A 和 A/D 转换器接口	(206)
11.1 D/A 转换器接口	(206)
11.1.1 D/A 转换器及其连接特性	(206)
11.1.2 D/A 转换器与微处理器的接口方法	(207)
11.1.3 D/A 转换器接口电路举例	(208)
11.1.4 D/A 转换器应用的微机接口电路	(211)
11.2 A/D 转换器接口	(213)
11.2.1 A/D 转换器及其连接特性	(213)
11.2.2 A/D 转换器与微处理器的接口方法	(214)
11.2.3 A/D 转换器接口电路举例	(215)
11.3 微型计算机系统的 A/D 和 D/A 通道	(221)
11.3.1 多通道模拟开关	(221)
11.3.2 采样保持器	(222)
11.3.3 A/D 通道的结构形式	(222)
11.3.4 D/A 通道的结构形式	(223)
11.3.5 数据采集与数据分配系统的 A/D 和 D/A 通道设计	(225)
11.4 高速微机数据采集系统	(227)

11.4.1 采用DMA方式的A/D转换器接口电路	(228)
11.4.2 初始化编程	(228)
习题十一	(229)
第十二章 软磁盘机接口	(230)
12.1 软磁盘机	(230)
12.1.1 软磁盘机的基本结构	(230)
12.1.2 软磁盘机的种类	(231)
12.1.3 软磁盘机的记录方式	(231)
12.1.4 软磁盘机的信号接口	(231)
12.1.5 软磁盘机的格式化	(232)
12.2 软磁盘机 BIOS 驱动程序	(234)
12.2.1 软磁盘 BIOS 的调用	(235)
12.2.2 软磁盘 BIOS 的结构与功能	(235)
12.3 软盘控制器(FDC)专用集成电路	(238)
12.4 IBM-PC 软盘控制器	(243)
12.4.1 电路原理图及主机侧接口电路	(243)
12.4.2 驱动器侧的接口电路	(243)
12.4.3 软盘控制器的应用	(246)
习题十二	(247)
第十三章 温盘机接口技术	(248)
13.1 温盘机概述	(248)
13.2 温盘机信号接口	(250)
13.2.1 ST506/412 接口	(250)
13.2.2 ESDI 接口	(252)
13.2.3 SMD 接口	(254)
13.2.4 SCSI 接口	(256)
13.2.5 IPI 接口	(259)
13.2.6 IDE 接口	(259)
13.3 硬盘 BIOS 驱动程序	(260)
13.4 温盘控制器(WDC)专用集成电路	(263)
13.5 WD1002 温盘控制器	(267)
13.5.1 主机侧接口逻辑	(267)
13.5.2 主控电路与驱动器接口电路	(269)
习题十三	(271)
参考文献	(273)

第一章 微型计算机接口技术概述

1.1 接口及接口的功能

要了解接口,首先要了解微型计算机系统的组成。一个微机系统除了 CPU 之外,还必须要存储器和接口电路、外部设备、总线(BUS)以及时钟、电源等部分,如图 1.1 所示。从图中可以看出,各类外部设备和存储器,都是通过各自的接口电路连到微机系统的总线上去的,因此用户可以根据自己的要求,选用不同类型的外设,设置相应的接口电路,把它们挂到系统总线上,构成不同用途、不同规模的应用系统。

所谓接口就是 CPU 与外界的连接部件(电路),是 CPU 与外界进行信息交换的中转站。例如源程序或原始数据要通过接口从输入设备送进去,运算结果要通过接口向输出设备送出来;控制命令通过接口发出去,现场状态通过接口取进来,这些来往信息都要通过接口进行变换与中转。微机接口技术是采用硬件与软件相结合的方法,研究微处理器如何与外部世界进行最佳耦合与匹配,以在 CPU 与外界之间实现高效、可靠的信息交换的一门技术。这里所说的“外部世界”,是指除 CPU 本身以外的所有设备或电路,包括存储器、I/O 设备、控制设备、测量设备、通信设备、A/D 和 D/A 转换器等。从解决 CPU 与外设在连接时存在的矛盾的观点来看,接口一般有如下功能:

1. 数据缓冲功能

接口中一般都设置数据寄存器或锁存器,以解决主机高速与外设低速的矛盾,避免因速度不一致而丢失数据。为了联络,接口电路还要提供寄存器“空”、“满”或“准备好”、“忙”、“闲”等状态信号,向 CPU 报告寄存器的工作情况。

2. 设备选择功能

微机系统中一般带有多种外设,同一种外设中也可能有多台,而 CPU 在同一时间里只能与一台外设交换信息,这就要借助于接口中的地址译码电路对外设进行寻址。高位地址用于芯片选择,低位地址用于芯片内部寄存器或锁存器的选择,以选定需要与自己交换信息的设备,只有被选中的设备才能与 CPU 进行数据交换或通信。

3. 信号转换功能

由于外设所能提供的状态信号和它所需的控制信号往往同微机的总线信号不兼容,尤其是连接不同公司生产的芯片时,信号转换就不可避免,因此,信号转换(包括 CPU 的信号与外设的信号的逻辑关系上、时序配合上以及电平匹配上的转换)就成为接口设计中的一个重要任务。

4. 中断管理功能

当外设需要及时得到 CPU 的服务,特别是在出现故障需要 CPU 进行刻不容缓地处理时,就要求在接口中设置中断控制器,以便为 CPU 处理有关中断事务(提出中断请求,进行中断优先级排队,提供中断向量等)。这样既提高了微机系统对外界的响应速度,又使 CPU 与外设并行工作,提高 CPU 的效率。

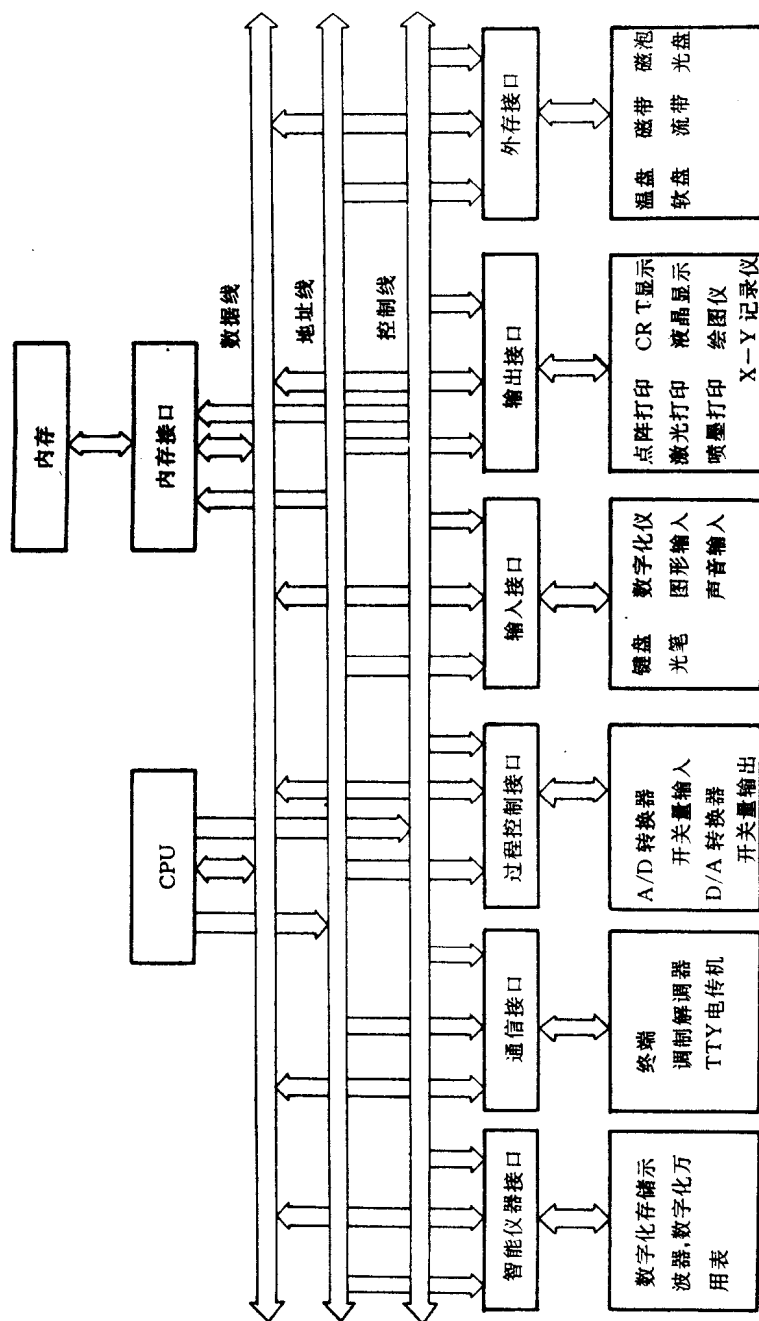


图 1.1 微机系统组成框图

5. 数据宽度变换的功能

CPU 所处理的是并行数据(8 位、16 位或 32 位),而有的外设(如串行通信设备,磁盘驱动器等)只能处理串行数据。在这种情况下,接口就应具有数据“并→串和串→并”的变换能力。为此,在接口中设置移位寄存器。

6. 可编程功能

现在的接口芯片基本上都是可编程的,这样在不改动硬件的情况下,只修改驱动程序就可以改变接口的工作方式,大大增加了接口的灵活性和可扩充性,使接口向智能化方向发展。

上述功能并非是每种接口都要求具备,对不同配置和不同用途的微机系统,其接口功能不同,接口电路的复杂程度大不一样。但前三种功能和可编程能力是一般接口都需要的。

1.2 接口技术的发展简述

早期的计算机系统中并没有设置独立的接口部件,对外设的控制与管理全由 CPU 直接承担。这在当时外设品种少、操作简单的情况下,不设置接口可以勉强由 CPU 承担。然而,由于微机技术不断发展,其应用越来越广泛,外设门类品种大大增加,且性能各异,操作复杂,因此,不设置接口是不行的。首先,如果仍由 CPU 直接管理外设,则会使主机完全陷入与外设打交道的沉重负担之中。因为 CPU 要向外设发控制信号,包括设备的选定,设备启动,信息的转换,数据的装配与拆卸,地址的改变以及检测和判断信息是否结束等,这些操作都由主机按程序进行,而且每交换一次信息就需要按上述过程循环一次,直到所交换的信息完成之后,主机才能做下一步的工作,效率非常之低。其次,是外设种类繁多,且每种外设提供的信息格式、电平高低和逻辑关系各不相同,因此,主机对每一种外设就要配置一套相应的控制和逻辑电路,使得主机对外设的控制电路非常复杂,而且是固定的联接,不易扩充和改变,这种结构极大地阻碍了计算机的发展。

为了解决以上矛盾,起初在 CPU 与外设之间设置了简单的接口电路,后来逐步发展成为独立的接口和设备控制器,把对外设的控制任务交给接口去完成,这样大大地减轻了主机的负担,简化了 CPU 对外设的控制和管理。同时,有了接口之后,研制 CPU 时无须考虑外设的结构特性如何,反之,研制外设时也不需考虑它是同哪种 CPU 联接,处理器与外设按各自的规律更新,形成微机本身和外设产品的标准化和系列化,促进了微机系统的发展。并且,由于超大规模接口集成芯片的出现,使得微机接口向智能化、标准化、单一芯片化方向发展。

1.3 CPU 与接口之间传送信息的方式

外设与微机之间的信息传送实际上是 CPU 与接口之间的信息传送,传送的方式不同,CPU 对外设的控制方式也不同,从而使接口电路的结构及功能也不同,所以接口电路设计者对 CPU 与外设之间采用什么方式传送信息颇为关心。一般有以下三种方式:

1. 查询方式

查询方式是主机在传送数据(包括读入和写出)之前,要检查外设是否“准备好”,若没有准备好,则继续查其状态,直至外设准备好了,即确认外设已具备传送条件之后,才能进行数据传送。显然,在这种方式下,CPU 要传送一个数据,需花费很多时间来等待外设进行数据传送的准备,且 CPU 与外设不能同时工作,各种外设也不能同时工作,因此,信息传送的效率非常之

低。但实现这种方式的接口电路简单,硬件开销小,在 CPU 不太忙且传送速度不高的情况下,可以采用。

2. 中断方式

采用中断方式传送信息时,无需反复测试外设是否准备好的状态。在外设没有作好数据传送准备时,CPU 可以运行与传送数据无关的其它指令。外设作好传送准备后,主动向 CPU 请求中断。若 CPU 响应这一请求,则暂停正在运行的程序,转入用来进行数据传送的中断服务子程序,完成中断服务子程序(即完成数据传送)后,自动返回原来运行的程序。这样,虽然外设工作速度比较低,但 CPU 在外设工作时,仍然可以运行与外设传送无关的其它程序,使外设与 CPU 并行工作,提高了 CPU 的效率。为了实现中断传送,要求在 CPU 与外设之间设置中断控制器。中断方式用于 CPU 的任务比较忙,传送速度不太高的系统中,尤其适合实时控制及紧急事件的处理。

3. 直接存储器存取(DMA)方式

虽然中断传送方式可以在一定程度上实现 CPU 与外设并行工作,但是在外设与内存之间,或在外设与外设之间进行数据传送时,还是要经过 CPU 中转(即经过 CPU 的累加器读进和送出),这对高速外设(如磁盘)在进行大批量数据传送时,会造成中断次数过于频繁,不仅传送速度上不去,而且耗费大量 CPU 的时间。为此,采用直接存储器存取方式,使 CPU 不参加数据的传送工作,由 DMA 控制器来实现内存与外设,或外设与外设之间的直接快速传送,从而也减轻了 CPU 的负担。这种方式使计算机的硬件结构发生了变化,信息传送从以 CPU 为中心变为以内存为中心。若采用高速存储器,就使外设与 CPU 分时访问内存得以实现。

DMA 方式实际上是把输入输出过程中外设与内存交换信息的那部分操作与控制交给了 DMA 控制器,简化了 CPU 对输入输出的控制,这对高速度大批量数据传送特别有用。但这种方式要求设置 DMA 控制器,电路结构复杂,硬件开销大。

1.4 设计与分析接口电路的基本方法

对一个已有接口电路进行分析解剖,或要求设计一个新的接口电路,其基本方法是:

1. 分析接口两侧的情况

一侧是 CPU,另一侧是外设。对 CPU 一侧,要搞清是什么类型的 CPU,以及它提供的数据线的宽度(8 位、16 位、32 位等),地址线的宽度(16 位、20 位、24 位)和控制线的逻辑定义(高电平有效、低电平有效、脉冲跳变),时序关系有什么特点。其中数据与地址线比较规整,不同的 CPU 其变化不大,而控制线往往因 CPU 不同其定义与时序配合差别较大,故重点要放在控制线的分析上。外设一侧的情况很复杂,这是因为外设种类繁多,型号不一,所提供的信号线五花八门;其逻辑定义、时序关系,电平高低差异甚大。对这一侧的分析重点放在搞清被连外设的工作原理与特点上,找出需要接口为它提供哪些信号才能正常工作,它能反馈给接口哪些状态信号报告工作过程,以达到与 CPU 交换数据的目的。外设的种类甚多,从高容量快速磁存储器到指示灯和扬声器,不管其复杂程度如何,只要将它们的工作原理及各自原始的(本身所固有的)来去信号线的特性分析清楚了,对接口电路的剖析也就不难。

2. 进行信号转换

由上述可知,要把 CPU 与外设两侧的信号线,不加处理(改造)就直接连接是不行的。因此,经过对接口两侧信号的分析,找出两侧信号的差别之后,设法进行信号转换与改造,使之协

调。这可以从 CPU 一侧做起,将 CPU 的信号进行转换以达到外设的要求,也可以从外设一侧做起,将外设的信号进行改造(逻辑处理)以达到 CPU 的要求。经过改造的信号线,在功能定义、逻辑关系和时序配合上,能同时满足两侧的要求,故可以协调工作。因此,在分析已有接口电路时,可以从两侧的原始信号出发查查它们,通过哪些元器件进行了改造与转换,最后送到什么地方去了。顺藤摸瓜,追根求源,搞清来龙去脉。

3. 接口驱动程序分析

接口的硬件电路只提供了接口工作的条件,要使接口真正发挥作用就要配备相应的驱动程序。对于微机系统中的标准设备(如 CRT、KB、PRINTER、HD、communication 等),在 ROM-BIOS 中都有相应的功能块子程序供用户调用。但是对于接口设计者来说,常常碰到的是一些非标准设备,况且在微机控制应用中,往往采用单板机或单片机,此时没有配置 BIOS,故无功能块子程序可供调用,所以需要自己动手编制接口驱动程序。为此,必须了解外设的工作原理和接口电路的硬件结构,否则,无法编程。接口驱动程序是模块化和结构化的,一般由初始化模块和功能模块等组成。

4. 在对具体接口电路进行设计与分析时应注意事项

一是尽量采用 LSI、VLSI 集成接口芯片,这样不仅开发周期短,而且可靠性高,特别是选用与 CPU 同一系列的接口芯片,就更为方便。二是软硬结合,综合考虑。在不同场合,实现同一种接口功能,可以采用硬件软化,也可以采用软件硬化的方式,根据需要与可能具体进行优化选择。不过一旦硬件结构定下来,相应的软件编程也就跟着定下来了。

总之,分析接口问题的基本方法可归纳为:首先分析接口两侧的信号及其特点,找出两侧进行连接时存在的差异;然后针对要消除两侧的这些差异,来确定接口应完成的任务;为了实现接口的任务,要考虑作哪些信号变换,选择什么样的元器件来进行这些变换,并据此进行接口电路功能模块化总体结构设计,这样就完成了对接口硬件的分析。对接口问题,仅有硬件分析,还不能真正了解,还必须对接口的软件编程进行分析。而软件编程是与硬件结构紧密相关的,硬件发生变化,接口的驱动程序也就随之改变。

1.5 外围接口芯片

由于现代微电子技术的成就和集成电路的发展,目前,各种功能接口电路都已做成集成电路,即已由中规模或大规模集成接口芯片代替过去的数字电路。因此,在接口设计中,通常不需要繁杂的电路参数计算,而需要熟练地掌握和深入了解各类芯片的功能、特点、工作原理、使用方法及编程技巧,以便根据设计要求和经济准则,合理选择芯片,把它们与微处理器正确地连接起来,并编写相应的驱动程序。采用集成接口芯片不仅使接口体积小,功能完善,可靠性高,易于扩充,应用极其灵活、方便,而且推动接口向智能化方向发展。所以,接口芯片在微机接口技术中,起着很重要的作用,应给以足够的重视。

外围接口芯片种类繁多,有中规模与大规模、可编程与不可编程、通用与专用之分。常用的有:并行口芯片,如 8255、PIO、6821;串行口芯片,如 8251、SIO、8250;定时/计数器,如 8253(8254)、CTC;中断控制器,如 8259;DMA 控制器,如 8237、Z80DMA;LED 显示/键盘控制器,如 8279;CRT 控制器,如 8275、6845;软盘控制器,如 8272、6843、 μ PD765;GPIB 控制器,如 8291、8292、68488 等等。另外,在接口电路中还经常使用中规模器件(如锁存器、译码器、总线驱动器和各种门电路)作为辅助元件。在模拟接口中,还要与 A/D、D/A 转换器芯片及 OP 芯

片打交道。这些将在本书各章节的接口电路中进行详细讨论或介绍。

随着集成电路集成度的增高,电子计算机向微型化和超微型化方向发展,微型计算机已成为导弹、智能机器人、人类宇航和探测太空奥秘复杂系统必不可少的智能部件。目前,微机不仅作为科学计算、实时控制、现代化通信和管理的手段,而且也成为人类进行学习、看病、咨询、智力游戏等生活服务和娱乐的工具。然而,在微机系统中,微处理器的这种神通广大的功能必须通过外部设备才能实现,而外设与微处理器之间的信息交换及通信又是靠接口来实现的,因此,接口技术就成为直接影响微机系统的处理能力和微机推广应用的关键。可以这样说,微机的应用是随着外部设备不断更新和接口技术的发展而深入到各个领域的。所以,掌握微机接口技术就成为当代的科技和工程技术人员应用微型计算机必不可少的基本技能。

习 题 一

1. 微型计算机的接口一般具备哪些功能?
2. 为什么要在 CPU 与外设之间设置接口?
3. CPU 与接口之间有哪几种传送信息的方式? 它们各应用在什么场合?
4. 设计与分析接口电路的基本方法是什么?
5. 外围接口芯片在微机接口技术中的作用如何? 你所知道的外围接口芯片有哪些?
6. 接口技术在微机应用中起什么作用?

第二章 I/O 接口地址译码技术

在分析和设计微机系统各类接口电路时,除了要考虑 CPU 与 I/O 设备之间的信号转换及其连接之外,往往还要涉及到对接口电路中寄存器的访问,即 I/O 接口的寻址问题。本章将对此进行讨论。

2.1 I/O 接口的寻址方式

外部设备接口中能被 CPU 直接访问的寄存器通常称之为端口(PORT)。CPU 通过这些端口发送命令,读取状态和传送数据,因此,一个接口可有几个端口,如命令口、状态口、数据口等。如何实现对这些端口的访问,就是所谓 I/O 接口寻址问题。有两种寻址方式。一种是端口地址与存储器地址统一编址,即存储器映射方式;另一种是 I/O 端口地址和存储器地址分开独立编址,即 I/O 映射方式。

1. 统一编址方式

这种方式,是从存储空间划出一部分地址空间给 I/O 设备,把 I/O 端口当作存储器单元一样进行访问,不设置专门的 I/O 指令。凡对存储器可以使用的指令均可用于端口。6800 系列,6502 系列微型机和 PDP-11 小型机就是采用这种方式。

统一编址方式由于对 I/O 设备的访问是使用访问存储器的指令,指令类型多,功能齐全,这不仅使访问 I/O 设备端口进行输入/输出操作灵活、方便,而且还可对端口内容进行算术逻辑运算,移位等等。另外,能给端口有较大的编址空间,这对大型控制系统和数据通信系统是很有意义的。这种方式的主要缺点是端口占用了存储器的地址,使存储器容量变小,还有指令长度比起专门 I/O 指令要长,因而执行时间较长。

2. 独立编址方式

这种方式是接口中的端口地址单独编址而不占用存储空间,访问端口时要求使用专门的 I/O 指令,可访问的端口地址一般为 256~1024 个。大型计算机通常采用这种方式,有些微机,如 8088/8086 和 Z-80 系列机也采用这种方式。

这种方式的优点是不占用存储器地址,故不会缩小存储器容量,单独用于端口的地址线少,地址译码方便,I/O 指令短,执行速度快,特别是由于专门 I/O 指令与存储器访问指令有明显的差别,使程序中 I/O 操作和其它操作层次清晰,便于理解。这种方式的缺点是指令类型少,功能单一,只能对端口进行输入输出操作。这种方式要求 CPU 设置两组读/写控制信号(存储器读/写,I/O 读/写)。例如,8086/8088 最小模式下要用 $\text{IO}/\overline{\text{M}}$ 引脚和 $\overline{\text{RD}}$ 、 $\overline{\text{WR}}$ 构成两组控制信号;而在最大模式下,由于引脚不够用,没有直接输出 $\text{IO}/\overline{\text{M}}$ 、 $\overline{\text{RD}}$ 和 $\overline{\text{WR}}$,故这些对外设和存储器进行读写操作的控制信号,由 S_2 、 S_1 、 S_0 输出三个总线周期状态信号编码送至总线控制器 8288,以取代 $\text{IO}/\overline{\text{M}}$ 、 $\overline{\text{RD}}$ 和 $\overline{\text{WR}}$ 等,经 8288 解读后,再生成存储器读/写和 I/O 读/写两组控制信号。

分为门电路译码和译码器译码。目前,不少系统中采用 GAL 或 PAL 器件进行译码,这种器件使用起来非常方便。

表 2.2 插槽 I/O 口地址使用表

地址空间	器 件/设 备	地址空间	器 件/设 备
200~20F	游戏卡	380~38F	SDLC 通信卡
210~217	扩充部件	390~3AF	未用
218~2F7	未用	3B0~3BF	单显/打印卡
2F8~2FF	异步通信卡(二)	3C0~3CF	未用
300~31F	未用	3D0~3DF	彩显/图形卡
320~32F	硬盘卡	3E0~3EF	未用
330~377	未用	3F0~3F7	软盘卡
378~37F	打印卡	3F8~3FF	异步通信卡(一)

1. 固定式接口地址译码

所谓固定式是指接口卡中用到的口地址不能更改,目前,接口卡中大部分都采用固定式译码。在固定式译码方式中,若仅需一个口地址时,则采用门电路构成译码电路很简便。如图 2.1(a)所示电路,可译出 2F8H 读操作口地址。图 2.1(b)所示电路,能译出进行读/写操作的 2E2H 口地址。图中 AEN 参加译码,它对口地址译码进行控制,只有当 AEN=0 时,即不是 DMA 操作时译码才有效;当 AEN=1 时,即是 DMA 操作时,使译码无效。避免在 DMA 周期,由 DMA 控制器对这些 I/O 口地址的外部设备进行读/写操作。

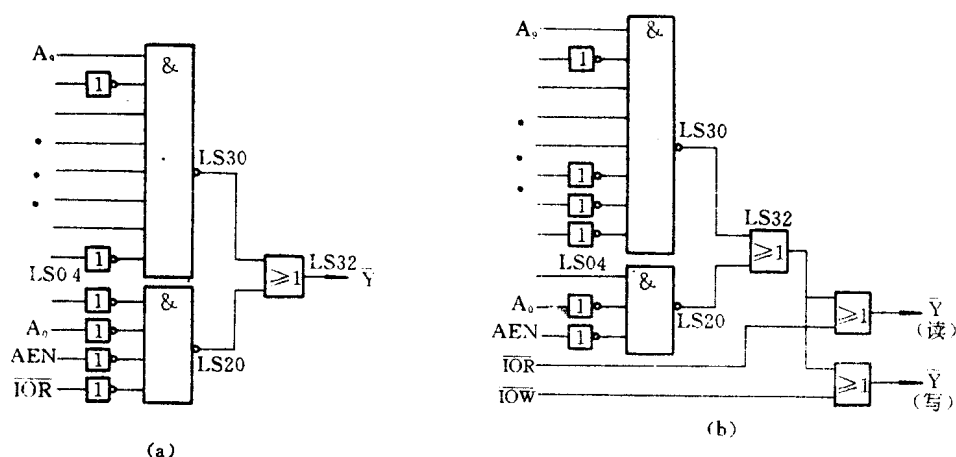


图 2.1 门电路译码电路

若接口电路中需使用多个口地址时,则采用译码器译码比较方便。译码器的型号很多,如 3-8 译码器 74LS138, 4-16 译码器 74LS154, 双 2-4 译码器 74LS139 和 74LS155 等。采用 3-8 译码器 74LS138, 可从它输入的 3 个代码中(A,B,C)解出 8 个输出来($Y_0 \sim Y_7$), 它的 3 个输入控制端是 $G_1, \overline{G_2A}, \overline{G_2B}$, 只有当 $G_1=1, G_2A=G_2B=0$ 时,才允许对输入 A,B,C 进行译码。若把

G_1 接高电平, $\overline{G_2A}$ 接 AEN, $\overline{G_2B}$ 接 $\overline{IOW}$ (或 $\overline{IOR}$), 输入端 A, B, C 分别接 A_0, A_1, A_2 , 则当不是 DMA 操作时, 由输入端 A, B, C 的编码来决定输出: CBA = 000, 则 $Y_0 = 0$; CBA = 001, $Y_1 = 0 \dots$; CBA = 111, $Y_7 = 0$ 。例如 PC 机系统板上的口地址译码, 采用 74LS138 译码器, 如图 2.2 所示。图中地址的高 6 位 $A_9 \sim A_5$ 参加译码, 分别产生 DMAC (8237), PIC (8259), T/C (8253), PPI (8255A) 的片选 $\overline{CS}$ 信号, 而地址的低 4 位作芯片内部寄存器的访问地址。很明显, 8237A 的口地址范围是 000~01FH, 8259A 的口地址范围是 020~03FH 等等, 正好和前面表 2.1 所列出的口地址分配表一致。

2. 开关式可选接口地址译码

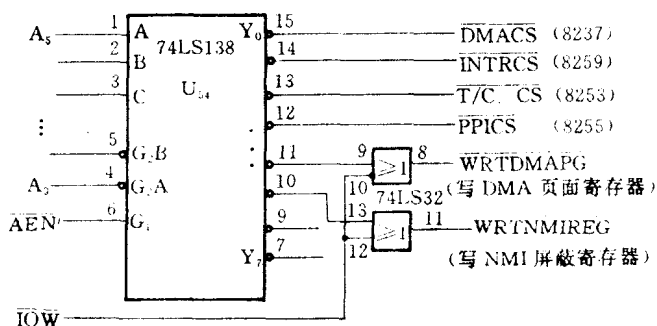


图 2.2 系统板口地址译码电路

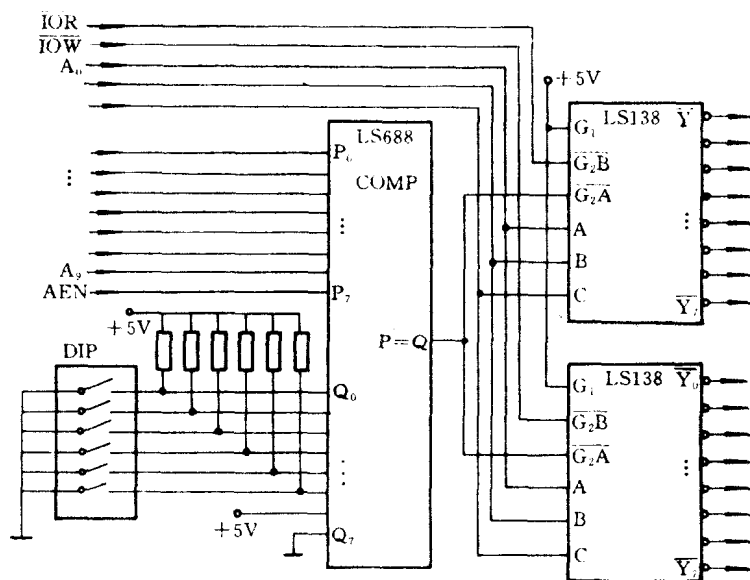


图 2.3 开关式可选译码电路

如果用户要求接口卡的口地址能适应不同的地址分配场合, 或为系统以后扩充留有余地, 则采用开关式口地址译码。这种译码方式使接口卡的 I/O 口地址可根据要求加以改变而无需改动线路, 如图 2.3 所示。图中 DIP 开关状态的设置, 就决定了译码电路的输出, 若改变开关状态, 就改变了 I/O 口地址。电路中使用了一片 8 位比较器 74LS688, 当输入端 $P_0 \sim P_7$ 的地址与设置端 $Q_0 \sim Q_7$ 的开关状态一致时, 输出为低。使用时可预置 DIP 开关为某一值, 得到一组所要求的口地址。若考虑到读写分别控制, 让 $\overline{IOR}$ 和 $\overline{IOW}$ 信号参加译码, 使 8 个口地址作 16 个使用。图中保证 $A_9 = 1, AEN = 0$ 才使译码有效。如果采用跳线连接代替 DIP 开关, 则可得到如