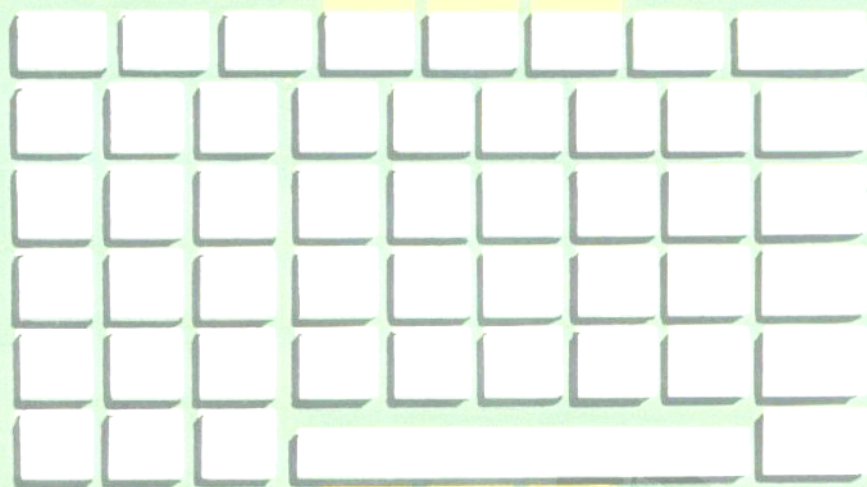


微处理器与微计算机 在交换技术中的应用

孔俊宝 陈振中 编



国防工业出版社

研
究
生
學
習

PDG

前 言

近几年来, 电话交换机已从纵横制逐渐转入程控数字电话交换制, 也就是采用电子计算机作为中央控制设备, 由计算机中存储程序, 用软件来控制交换接续的交换系统。目前, 我国正在大力发展程控数字电话交换技术。编写本书的目的, 就是介绍计算机在交换技术中的具体应用情况及其基本原理, 以使有关的工程技术人员和高等院校的师生作为进修学习和参考用书。

本书共分八章。第一章是概述了交换技术中微计算机系统的组成及其接口的基本类型。第二章是从通道寻址、数据传送及其控制方法以及微计算机系统的输入、输出技术, 进行了深入的分析讨论和在交换技术中的应用。第三章较全面地分析了各种总线技术及其标准。第四章介绍了在微计算机系统之间实现串、并行通信的硬件连接和软件编程。第五章是概述了交换技术的发展及趋势。第六章是微计算机在交换技术中的控制系统。第七章是软件在交换技术中的应用。第八章是微计算机在电话交换中的应用, 并简介了两种典型交换机。

本书第一章至第四章由陈振中同志编写, 第五章至第八章由孔俊宝同志编写。全书由孔俊宝同志统编。

编者水平有限, 书中如有错误或不妥之处, 欢迎读者批评指正。

目 录

第一章 微计算机系统的组成及其接口的基本类型	1
§ 1-1 微计算机系统的组成	1
§ 1-2 微计算机系统的接口	3
第二章 微计算机的输入、输出方法	5
§ 2-1 输入/输出的寻址方式	5
§ 2-2 I/O的数据传送方式	8
§ 2-3 I/O数据传送的控制方法	15
第三章 总线技术及标准	24
§ 3-1 内部总线	25
§ 3-2 外部总线	28
第四章 微计算机系统间串并行通信的实现	55
§ 4-1 CPU之间的并行通信	55
§ 4-2 CPU之间的串行通信	75
第五章 交换技术基础	103
§ 5-1 引言	103
§ 5-2 信号、交换及网的发展趋势	103
第六章 交换技术中的控制系统	107
§ 6-1 概述	107
§ 6-2 交换网的控制	107
§ 6-3 控制结构及设备	109
§ 6-4 呼叫处理	118
§ 6-5 新电话业务	124
§ 6-6 市内电话的自动计费	127
第七章 交换系统中的软件	132
§ 7-1 概述	132
§ 7-2 软件技术基础	132
§ 7-3 基本软件	134
§ 7-4 交换机的新模型	135
§ 7-5 辅助软件	136
§ 7-6 数据库	137
第八章 微计算机在电话交换中的应用实例	139
§ 8-1 FETEX-150数字电话交换机简介	139
§ 8-2 ITT-1240数字电话交换机简介	142
参考文献	146

第一章 微计算机系统的组成 及其接口的基本类型

§ 1-1 微计算机系统的组成

计算机系统由硬件和软件两大部分组成。硬件是指组成计算机的任何机械的、磁性的、电子的装置和部件，又称机器系统，是组成计算机的物质基础；而软件是指为了方便用户和充分发挥计算机效能的各种程序的总称，又叫程序系统，它属于信息性的东西，是组成计算机的上层建筑。

微计算机是在小型计算机的基础上，借助于大规模集成电路技术而发展起来的。它也是由硬件和软件两大部分组成的。在微计算机中，硬件和软件更加密不可分，发展带有软件硬化的微计算机系统已成为一个重要的发展方向。所谓软件硬化就是把软件功能固化于硬件中，所以又称固件。

微计算机系统硬件的组成如图 1-1 所示，它由中央处理单元 CPU（或 MPU）、存储器 RAM 和 ROM 及输入/输出（又叫 I/O）接口电路组成，而 I/O 接口电路又和 I/O 设备相连。我们从图 1-1 中可见，在微计算机系统中各硬件之间是通过三组总线，地址总线 AB、数据总线 DB 和控制总线 CB 来连接的。

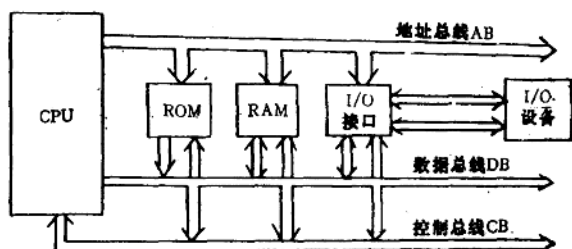


图 1-1 微计算机系统硬件组成

一、CPU

它是由大规模集成电路工艺技术制成的具有运算和控制功能的器件，又叫微处理器。

从概念上来说，CPU 是一种数字器件，它接收二进制数字输入信息，并能根据存放于存储器中的指令，对数据进行处理，产生二进制数字的输出信息。也就是说，在任何一个给定的时刻，CPU 的输出信息取决于下面两个因素：

1. 在该时刻之前输入到 CPU 全部信息的情况；
2. 存于存储器内部的指令。

第一个因素是时序电路共有的特点；而第二个因素则表明 CPU 具有程序控制的功能。

每一种 CPU 都有一组自己所能识别和执行的机器语言，称为该 CPU 的指令系统。在应用时，使用者必须根据所要解决的实际问题，应用 CPU 指令系统中的各种指令编出相应的应用程序；通过程序告诉 CPU 应当做些什么，先做什么，后做什么以及如何去做等等。因此程序控制是 CPU 功能性、适应性广的关键所在。

二、总线 (BUS)

单片的 CPU 是不可能发挥任何作用的。它必须通过引出线（也称引脚）和各种外电路相连，通常把这些和外电路的连线称为总线，按其功能可分为如下三种类型：

1. 数据总线 (DATA BUS)

所有与 CPU 传送数据的线的总体称为数据总线。例如八位的 CPU 就具有八位数据总线，每次传送八位二进制数据。通常八位二进制数称为一个“字节”，而四位二进制数称为“半字节”。八位数据用符号 D_0 到 D_7 来表示。其中 D_0 称为最低有效位 (LSB)，而 D_7 称为最高有效位 (MSB)。

CPU 的数据总线都有双向性，即每一根数据线既能输入数据，又能输出数据，当然同一时刻只能是其中之一。为完成这一功能，通常都采用“三态门”。

2. 地址总线 (ADDRESS BUS)

为实现运算和控制功能，CPU 必须与存储器交换信息，因为运算所需要的数据和控制运算的程序都存于存储器内。存储器有许多单元，每一个单元都有一个用以表示其位置的“地址”，CPU 正是通过对地址线发出相应的信息来选中所要求通信的存储单元。专门用来传送地址信息的全部传输线称为地址总线。

通常我们将从存储单元取出数据，并通过数据总线将其送入 CPU 的过程称为读存储器；而把 CPU 的数据通过数据总线存入存储器的过程称为写存储器。无论是读或写存储器，CPU 都必须首先给出所需要的存储单元地址，也就是通过地址总线送出二进制数的地址。由于地址总线上的每一条线只能有 1 或 0 两个状态，因此， P 条地址总线，就能对 2^P 个存储单元编排各不相同的地址。

3. 控制总线 (CONTROL BUS)

除了数据总线和地址总线之外，CPU 还有一组控制总线——输入控制线和输出控制线。通过它们使 CPU 的工作和外部电路的工作同步。控制总线上的控制信号，有的是高电平起控制作用，也有的是低电平起控制作用。

三、存储器

CPU 要充分发挥其信息处理的功能，必须有记忆程序和数据部件，这个部件就是存储器。

在一般情况下，存储器都与 CPU 分开，以便于不断扩充存储信息的容量。只有单片微计算机才在同一芯片上集成有 CPU、存储器和输入输出接口部件。

CPU 所使用的存储器大都属于半导体存储器。按照信息存储的功能又可将半导体存储器分为只读存储器 (ROM)，以及读写存储器，也叫随机存储器 (RAM) 两大类。它们之间的主要区别是：

1. 随机存储器工作时，既能读出已存入的信息，又能用新的信息取代原有的信息。也就是说读或写操作可随时按需要进行。而只读存储器则只能读出原来存入的信息而不能随机地写入。

2. 随机存储器的信息是用电写入的, 一旦断电, 所存信息就会消失, 因而属于易失性存储器。而只读存储器中的信息是预先用特殊方法写入的, 一旦写入信息就能可靠地存储下来, 不会因断电而消失, 因而属于非易失性存储器。

此外, 随机存储器又有双极型与 MOS 型两类, 一般都采用 MOS 型。而 MOS 型又分为静态存储器 (SRAM) 和动态存储器 (DRAM) 两种。SRAM 的基本单元是双稳态电路, 而 DRAM 的基本单元则依靠栅源电容的充放电。如果采用的存储器容量较小, 要求外加电路尽可能简单时, 宜采用静态存储器; 而在存储容量较大时, 则宜采用动态存储器。

ROM 分为由工厂编程或由用户编程两种。工厂编程的 ROM 是在制造时, 通过集成电路的金属化掩膜来确定 ROM 存储单元的状态。当这种工艺大量制造时, ROM 的成本是最低的。用户可编程的只读存储器 (PROM) 有两种类型, 用熔丝连接的 PROM 只能编程一次; 而可擦除的 PROM 能够多次擦除和重新写入。其中一种称作紫外线擦除写入的 PROM (UVEPROM), 它是用强烈的紫外线透过石英玻璃照射芯片来擦除信息的; 另一种是 EAEPROM, 它用电来擦除信息, 使用方便, 但其价格远比 UVEPROM 贵, 所以尚未普遍使用。

§1-2 微计算机系统的接口

在微计算机系统中, 一个部件与另一些部件之间的联系, 既需要硬件的连接又需要软件的控制。完成这一任务的硬、软件的综合称为接口。对这些硬、软件的设计称为接口技术。接口通常分为四种类型: 用户交换接口 (又叫用户通信接口), 辅助操作接口 (又叫内务操作接口), 传感器接口和控制接口。

用户交换接口的主要功能是将来自用户的数据、信息传送给微计算机, 或将来自微计算机的数据、信息传送给外部设备。常见的打印机接口、键盘接口和终端显示接口均属于这一类。

辅助操作接口包括总线驱动器、总线接收器、时钟电路、磁带和磁盘系统接口。一般情况下, CPU 的扇出仅 2~4 个 TTL 负载。如果 CPU 所带的存储器或接口较多时, 则 CPU 的驱动能力将不足。为使信息能有效地传送, 需要加接总线驱动器和总线接收器。

传感器接口通常用于微计算机控制和检测系统中。由于客观世界是一个模拟世界, 微计算机要控制和检测的外部信息例如压力、温度、流量和形变等一般均是模拟量, 而微计算机所能接受和处理的都是数字量, 因此常常通过传感器接口去监视和感受外部世界被检测或控制对象的变化, 首先将这种变化转换成电压或电流的变化, 然后再进一步转换成微计算机所能接受的数字量。

控制接口是当微计算机进行处理和运算后, 需要控制外部执行机构动作时 (例如驱动马达, 启动阀门或点亮指示灯等), 用来把微计算机输出的数字量转换成模拟量, 以实现对外部世界的控制。

无论是哪一种类型的接口, 从 CPU 的立场上分析, 不是输入, 就是输出, 或两者兼有; 而且主要解决以下几方面的问题:

1. 输入 把外部设备送往微计算机的信息转换成与微计算机相容的格式。

2. 输出 把微计算机送往外部设备的信息转换成与外部设备相容的格式。

这里的相容是指在信号类型、信号电平和信号传送方式等方面,使得外部设备和微计算机之间取得协调。

3. 同步脉冲的产生 通常外部设备不止一个,微计算机在不同时刻需要和不同的外部设备取得联系,为此必须对不同的外部设备进行寻址。这就需要产生设备选择脉冲,以使相应的外部设备与微计算机同步工作。

4. 中断处理 检测并处理外部设备发往 CPU 的中断请求信号。

实际上一个完整的接口设计,包括机械、电气和功能几方面要素。机械要素包括接插件、电缆等;电气的包括发送和接收电路、信号形式和电平等;功能的要素包括接口的管理能力、接收、发送和控制等逻辑以及信息系统的编程等。

综上所述,为使微计算机有效地与客观世界或用户进行联系,充分发挥其效能,接口设计是一项重要的工作。因而掌握接口技术,就成了推广应用微计算机的关键。

第二章 微计算机的输入、输出方法

在微计算机系统的各种应用场合中, CPU 除了与存储器之间交换信息, 进行存储器读、写操作外, 还必然要经常和各种外部设备交换信息。例如把计算机的输出信息通过打印机打印在记录纸上, 供使用者查看或保存。又如用 CRT 显示器将计算机输出数据变成各种直观的文字或图像; 通过键盘操作向计算机送入数据或控制命令; 通过纸带读入机或卡片输入机输入程序或命令; 通过 A/D 转换器把现场或控制对象的状态信息送入计算机或通过 D/A 转换器把计算机的输出信息转换成各种控制信号。还可以通过磁带机或磁盘在外部为计算机存储信息, 并供计算机需要时调用。

在各种外部设备中, 有一类是把程序、命令、原始数据和各种现场状态送入计算机的, 称为输入设备。键盘、纸带阅读机、卡片输入机和 A/D 转换器等属于这一类; 另一类是接受计算机送出的计算结果和控制信号等, 称为输出设备。打印机、CRT 显示器和 D/A 转换器等属于这一类。另外还有一些外部设备则兼有输入和输出二者的功能, 例如磁带机和软磁盘等。常把各种外部设备统称为输入/输出设备, 简称 I/O 设备。它们是计算机系统的重要组成部分。CPU 与 I/O 设备之间的信息交换称为输入或输出, 简称 I/O。

I/O 设备具有多样性, 这表现在 I/O 设备的型式是多样的, 有机械式、电子式、电动式及其它形式; 也表现在信息的类型上, 有模拟量、数字量或开关量。信息的传送速度可快达 1M bit/s , 也可以慢到秒的数量级。信息的传送方式可以是并行的 (若干位同时传送) 或者是串行的 (一位一位依次传送)。因此 CPU 与 I/O 设备之间的连接和信息交换比较复杂。下面分别讨论与输入、输出相关的几个重要问题。

§ 2-1 输入/输出的寻址方式

CPU 都使用同一条数据总线和存储器或输入/输出设备交换信息。如何在同一条总线上区分是 I/O 传送还是存储器传送? 通常有两种方法:

一、专用 I/O 通道寻址方式

这种寻址方式又叫端口寻址方式, 如图 2-1 所示。在这种方式中, 存储器的地址空间和 I/O 端口的地址空间是分开的, 所以这种 I/O 寻址方式不会影响存储器的地址空间。存储器用 16 条地址线 $A_{15} \sim A_0$ 寻址, 可寻址 64K 存储器空间 ($0000\text{H} \sim \text{FFFFH}$), I/O 端口用 8 条地址线 $A_7 \sim A_0$ 寻址, 可寻址 256 个 I/O 端口 ($00\text{H} \sim \text{FFH}$)。在读写控制命令方面也是分开的。对存储器的操作是将 $\overline{\text{MREQ}}$ 、 $\overline{\text{WR}}$ 和 $\overline{\text{RD}}$ 控制信号组合成存储器读、写命令 $\overline{\text{MEMR}}$ 和 $\overline{\text{MEMW}}$; 而对 I/O 设备的操作是将 $\overline{\text{IORQ}}$ 、 $\overline{\text{WR}}$ 和 $\overline{\text{RD}}$ 控制信号组合成 I/O 读写命令 $\overline{\text{IOR}}$ 和 $\overline{\text{IOW}}$ 。在操作指令方面, 存储器读写是用 LD 指令, 而 I/O 设备的读写有专用的 I/O 指令, 如 IN 和 OUT 指令以及相关的指令组。

图 2-2 是 I/O 端口寻址方式的一个具体例子。图中七段数字显示是输出设备, 它有七段选择和数位选择二个端口。这是因为每个显示器由七段组成, 共有六个显示器。

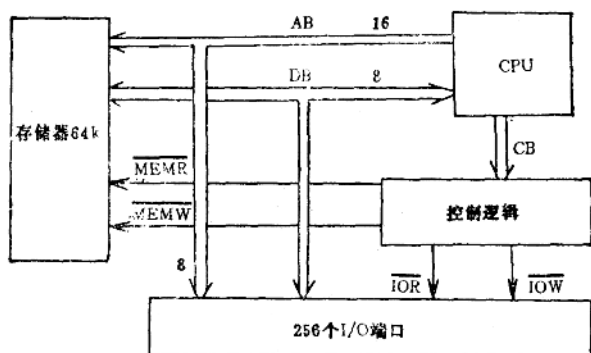


图2-1 专用 I/O 通道寻址方式结构

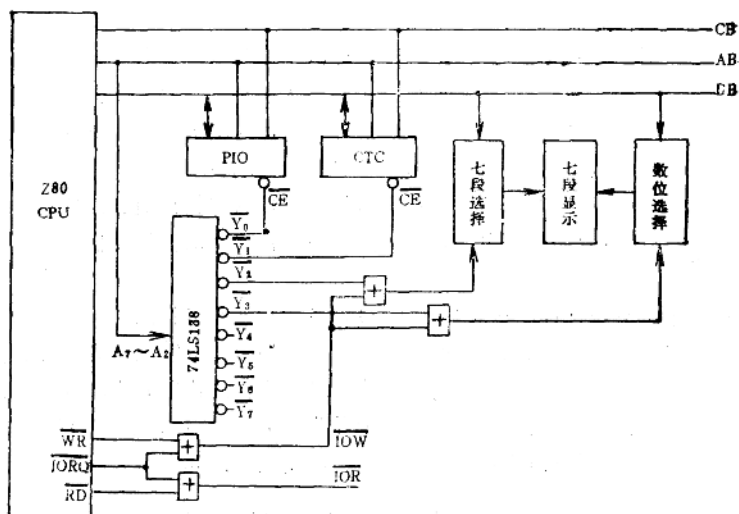


图2-2 I/O 端口寻址方式实例

某一时刻需要点亮某几位显示器，而对每一个显示器来说，又要决定点亮某几段，因此就需要设置两个端口。一个端口用来控制点亮的数位，而另一个端口用来控制点亮的段号。这两个端口相互配合，就可在六位显示器上显示出相应的字母、数字或符号。从图

表2-1 图2-2对应的端口地址分配

$A_7 \sim A_2$	译码有效输出	端 口	端 口 地 址
100000	$\bar{Y}_0$	Z80-PIO	80H~83H
100001	$\bar{Y}_1$	Z80-CTC	84H~87H
100010	$\bar{Y}_2$	七段字形选择	88H~8BH
100011	$\bar{Y}_3$	数位选择	8CH~8FH
100100	$\bar{Y}_4$	未 用	90H~93H
100101	$\bar{Y}_5$	未 用	94H~97H
100110	$\bar{Y}_6$	未 用	98H~9BH
100111	$\bar{Y}_7$	未 用	9CH~9FH

2-2 中可见, I/O端口的选择是用 3-8 译码器 74LS138 来实现的, 其输入是地址总线的 $A_7 \sim A_3$, 而 A_1 和 A_0 没有参加译码, 可为任意值。故译码器的每一个输出占有四个地址。表 2-1 列出了有关端口的地址分配。

二、存储器对应的 I/O 寻址方式

存储器对应的 I/O 寻址方式也称为存储器映像 I/O 寻址方式。这种寻址方式的结构如图 2-3 所示。在这种寻址方式中把 I/O 地址空间看成为存储器地址空间的一个组成部分, 在存储器中, 预先指定一部分地址空间作为 I/O 的地址空间。例如把 64 k 地址中的低 32 k 作为存储器地址, 高 32 k 作为 I/O 地址。从图 2-3 中可见, 不论对存储器还

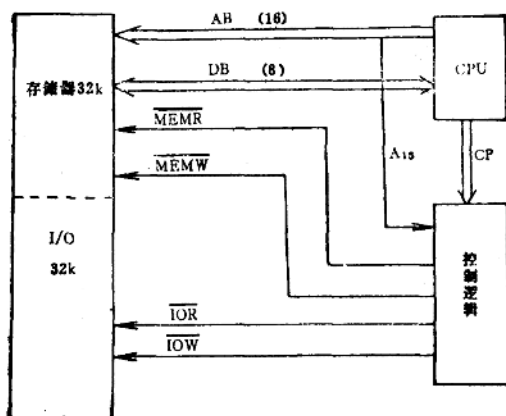


图 2-3 存储器对应 I/O 寻址方式结构

是对 I/O 寻址, 都由 16 条地址总线来完成。当地址总线的 $A_{15} = 0$ 时, 通过控制逻辑产生存储器读、写信号 ($\overline{\text{MEMR}}$ 和 $\overline{\text{MEMW}}$), 从而完成对存储器的存取操作; 而当 $A_{15} = 1$ 时, 通过控制逻辑形成 I/O 读、写命令 ($\overline{\text{IOR}}$ 和 $\overline{\text{IOW}}$), 以完成特定的 I/O 操作。对存储器和 I/O 的读、写指令都用 LD 指令, 而不必分别用两套指令。

存储器对应 I/O 寻址方式的主要优点是, 所有存储器访问指令都可用来处理 I/O 操作, 不需要专门的 I/O 指令。各种微计算机的指令系统中, 存储器访问指令花样较多, 所以编程比较灵活。另外, 由于在存储器中划出一个区域作为 I/O 的地址空间, 所以外设的数目可以几乎不受限制, 这种寻址的控制逻辑也比较简单。

存储器对应 I/O 寻址的主要缺点是: 存储器空间被外设占用了一部分, 存储器空间相对减少; 对 I/O 的访问和对存储器的访问一样, 需要全字长的地址译码, 这样执行 I/O 寻址的操作时间相对比较长; 由于用同样指令访问存储器和 I/O 设备, 因此在程序中较难区别这两种操作, 使程序比较难懂和难调试。

端口寻址方式的优点是: I/O 端口地址较短, 译码线路较简单; I/O 与存储器结构是彼此独立的, 两者的设计可分开进行, 比较方便, 程序也容易读懂。

端口寻址方式的缺点是: 需要专门的 I/O 指令, 这些指令一般不如存储器访问指令丰富, 所以程序设计的灵活性较差; 另外, 由于是 8 位地址译码, 外设数目受到限制, 控制逻辑也较复杂些。

对于一些专用的微计算机, 实际所需的 I/O 设备不多, 只有几个 I/O 端口。这时还可采用较简单的线性选址方法, 即把 8 位地址线的各位分别接到端口的选通端, 以每位为 1 (或 0) 时选通外设, 从而使译码线路得到简化, 译码时间也可以缩短。

§2-2 I/O 的数据传送方式

CPU 与 I/O 设备之间的信息交换是经 I/O 接口电路来实现的。按数据传送方式可分为并行传送和串行传送两种。任何数据传送必须包括源地址、目标地址和数据本身以及用来控制正确传送的相关装置, 我们可以将其归纳为三个关键要素: 地点、内容和时间。地点给出了数据的源地址和目标地址, 内容即数据本身, 时间是通过适当的控制, 使传送数据的过程获得正确地同步。数据从源地址传送到目标地址的过程中, 如果在一个脉冲信号上每次只传送一位, 那么这种传送方式就称为位串行或串行传送; 如果一次传送的是一组信息, 那么这种传送方式就称为并行传送。

一、并行数据传送

图2-4是一个典型的 8 位并行数据通道。SG 线为 8 位并行数据线的逻辑状态提供了公共地, DR (数据就绪) 线通知接收装置何时采样或者读入 8 位数据线上的数据。值得注意的是 8 位并行数据线是同时被采集的, 因此可以说 8 位数据总线是以字节为单位传送的, 所以有些计算机厂家常用字节串行传送来描述 8 位并行数据传送。

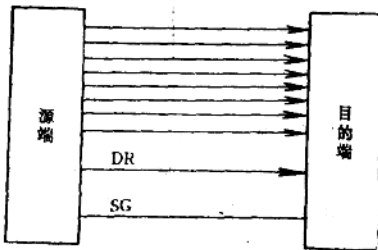


图2-4 简单的并行数据传送通路

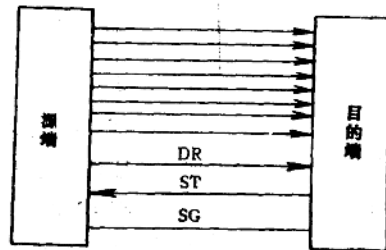


图2-5 两个握手信号的并行数据传送

在图2-4中, 数据就绪信号 DR 是由源端发出的同步信息, 源端完全控制着并行接口。如果目的端失灵, 则没有任何方法通知源端。因此大多数并行传送都采用多条握手联络信号, 以沟通源端和目的端之间的联系和同步。图2-5给出了具有两个握手信号线路的并行数据传送通路。源端将输出的 8 位数据送到 8 条数据线上, 发出 DR 信号, 通知目的端数据线上的数据已经就绪; 目的端收到 DR 信号后, 就把数据线上的数据取走, 同时发出选通信号 ST, 通知源端目的端已做好接收下一个字节的准备。这样当源端检测到 ST 信号时, 才开始发送下一个字节的信息。从而实现了收、发之间的正确同步。

二、串行数据传送

有些 I/O 设备, 在数据传送方面不具备并行能力, 而要求数据在一条传输线上一位一位地顺序传送。例如电传打字机和盒式磁带记录器等就属于这类设备。当它们和 CPU 之间传送数据时, 需经过串行接口。对 CPU 进行输入时, 数据应从串行变成并行; 从 CPU 输出数据时, 数据应从并行变成串行。另外当终端设备远离 CPU 时, 如果采

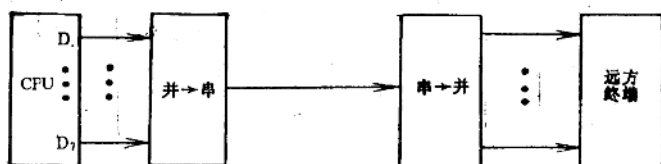


图2-6 CPU和远方终端的串行传送

用8条并行线传送数据，则用线量大，价格贵，这时也应采用串行传送方式，如图2-6所示。从而可以大大减少传送用线而降低成本。但串行传送的速度慢。为了确保正确和可靠的串行传送信息，必须妥善地解决以下一些问题。

(一) 串行数据的编码

在计算机和外界的通信中，一般需要传送英文大写字母(A~Z)、小写字母(a~z)、阿拉伯数字(0~9)以及大约25个数学运算符和语法符号等共87个符号。但是计算机只认识二进制码，因此必须规定编码方法，即以某种特定的二进制数串来代表上述需传送的87个符号。目前有两种常用的编码。一种是美国信息交换标准码ASCII(American Standard Code for Information Interchange)，这种编码已被广泛用于串行数据传送、字符显示和打印字符中。ASCII码采用七位二进制数，从00H~7FH总共有128个码。其中一部分如表2-2所示。ASCII码多用于小型及微计算机。另一种是扩展的BCD交换码EBCDIC(Extended Binary Coded Decimal Interchange Code)，它采用八位二进制数，从00H~FFH总共有256个码，多用于大型计算机。

表2-2 部分字符的ASCII码

字 符	ASCII码
0~9	30H~39H
A~Z	41H~5AH
a~z	61H~7AH

(二) 数据传送速率——波特率(Baud rate)

每秒钟串行发送或接收的二进制位数称波特率。例如每秒传送20个字符，每个字符由10位组成，那么传送速率是200波特。

(三) 串行传送(通信)的方式和规程

串行传送有两种基本方式，即异步通信和同步通信。由于在串行通信中没有配置专门的状态信号线来指示接收器或发送器在哪个时刻，怎样解释所传送的信号。所以不论是同步还是异步串行通信，都必须严格遵守约定的通信规程，才能正确地进行传送。

1. 异步传送规程

规定异步数据流中每个信息单元例如字符A或数字5等，必须由起始位(1位低电平)开始，而以停止位结束(1, 1.5, 2位高电平)，称为一帧。起始位和停止位称为帧位。

异步传送的格式如图2-7所示。在起始位和停止位之间的是N+1位二进制数和1位奇偶校验位(可有可无)。N的取值可为4、5、6或7，二进制数的各位可为0或

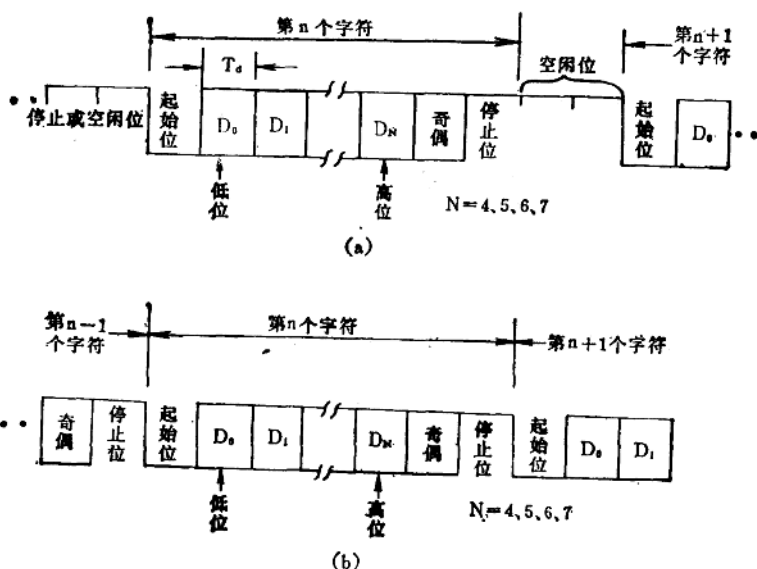


图2-7 异步串行传送的格式

(a) 小于最高数据传输率; (b) 最高数据传输率。

1, 由所传送的具体字符而定。如果有奇偶校验位, 若选用偶校验, 当 $N+1$ 位二进制数中“1”的个数为偶数时, 则奇偶校验位为 0, 否则为 1; 若选用奇校验, 当 $N+1$ 位二进制数中“1”的个数为偶数时, 则奇偶校验位为 1, 否则为 0。低电平的起始位又叫空号 (SPACE); 高电平的停止位又叫传号 (MARK)。停止位后面可能不紧跟下一个字符的起始位, 这时停止位后一直维持高电平状态 (“1”状态), 称这些位为“空闲位”。这个“1”状态, 保证当下一个字符的起始位到达时, 在通信线上能引起从“1”到“0”的跳变, 从而识别出新字符的到来。起始位和停止位不仅标志每个字符的开始和结束, 而更重要的是它们使得接收器能够把本地的时钟与每个新字符再同步。由于异步传送发送器和接收器的时钟不是同一个, 如图2-8所示。一般来说, 经过起始位同

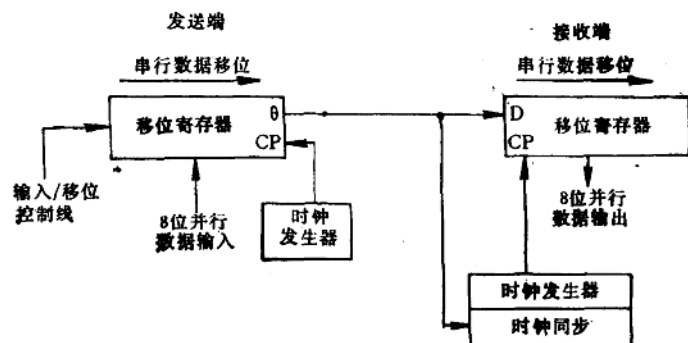


图2-8 异步串行传送框图

步以后,收发双方的时钟频率也不可能完全相同,如果接收端的时钟频率稍快,这种较小的误差,经过若干周期的积累,就会导致收发双方时钟严重失调,产生传送错误。如果接收端的时钟频率稍慢,同样由于误差的积累也会导致传送错误。为了减小这种积累的误差,保证数据的正确传送,异步串行传送,每传送一帧信息(一个字符),都利用帧位使收发双方的时钟再次获得同步。

异步传送的波特率一般在50~9600波特之间。设接收端的时钟周期为 T_c ,串行传送数据每一位(又称位单元)所占的时间为 T_d ,则有以下关系:

$$T_c = \frac{T_d}{K} \text{ 即 } T_d = kT_c$$

其中 $k=16$ 或 64 ,即每个位单元的时间是接收时钟的16倍或64倍。从图2-9中可见,在每个接收时钟的上升沿采样接收数据线,在找到连续九个“0”以后,方确认检测到起始位,从而建立起同步。这样就可以避免把瞬间的干扰误认为是起始位。并从此刻开始,每隔16个 T_c 读出字符的其它各位。从而保证接收端对每一位的采样时间,正好在每个位单元的中间,这样就提高了传送的可靠性。如每传送一个ASCII字符(7位),采用一位奇偶校验位,一位停止位,再加上一位停止位,共有10个位单元。显然在这10个位单元的时间内,接收时钟相对于发送时钟,其频率增加或减小所引起的时间偏差应小于 $1/2$ 个位单元的时间间隔。因此我们要求收发双方两个时钟的误差容限应在5%以内,目前的技术是很容易实现这个要求的。

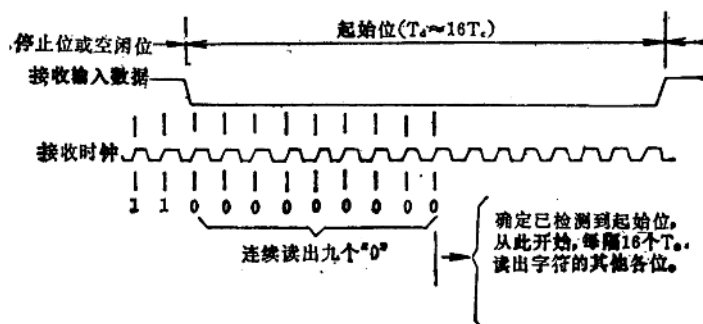


图2-9 接收端时钟和接收数据起始位之间的同步

2. 同步传送规程

在异步传送中,每一个字符都要用起始位和停止位作为字符的开始和结束标志,占用了时间。所以在数据块高速传送时,就去掉这些标志,采用同步传送。在同步通信规程中,以同步字符指示传送开始,采用一个同步字符的称单同步方式,采用两个同步字符的称双同步方式。同步字符后面跟的是要传送的所有数据信息,称数据场。数据场之后是两个循环冗余码校验(Cyclic Redundancy Check)字符。图2-10是同步通信信息格式。这个完整的通信格式称为一帧。采用单同步字符或双同步字符时,传送的同步是内部检测的。有时也可不用同步字符同步,而采用由外部输入一个同步信号来实现传送的同步,指示数据场的开始,这称外同步方式。

循环冗余校验字符用来检测同步传送过程的正确性。循环冗余字符是这样产生的,

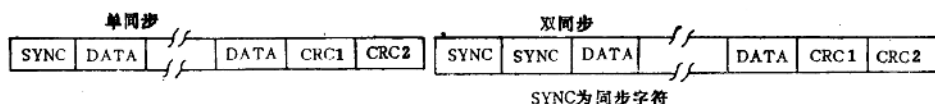


图2-10 同步传送的信息格式

把传送的数据块看作一组连续的二进制数并作为被除数，以一个二进制多项式作除数，通过模2除法运算，以所得余数作为发送数据时的循环冗余字符，附加在被发送的数据场之后。在接收方面，用与发送时同样的多项式作除数，对接收到的有CRC字符的二进制序列再作模2除法，若除的结果余数为0，则说明传送无误，否则就出错。在作模2除法时， $1+1=0$ ，不计及进位； $0-1=1$ ，不计及借位。

接收设备首先要搜索同步字符（SYNC），在得到同步字符后，才开始装配数据。在传送过程中，发送和接收设备必须保持完全同步，如果由于某种原因，例如噪声，使接收过程中漏掉了一位，则所有下面的数据字节的接收，都将是不正确的。

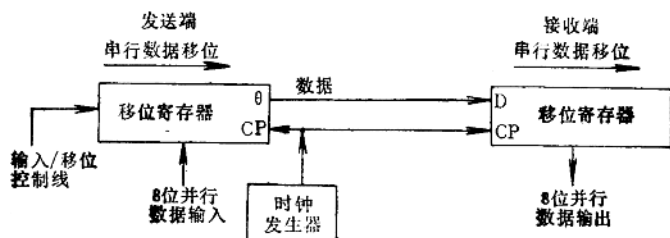


图2-11 同步串行数据传送

显然，在同步传送中要求发送和接收设备必须使用同一时钟，在近距离时这可以在传输线中增加一根时钟信号线，用同一时钟发生器来驱动发送和接收设备，如图2-11所示。在远距离传送时，这种方法是不行的。这时可由发送端用一定的编码方式，将同步信号和数据从同一根数据线上串行发送出去，接收端必须从接收到的信息中提取同步信号，用锁相技术可以得到和发送时钟频率完全相同的接收时钟信号。同步传送的速度高于异步，可达56千波特，但硬件电路较复杂。随着计算机通信网络技术的发展，同步通信将越来越受到重视。同步传送的HDLC(High-level Data-link Control)规程，目前已在许多数据通信系统和计算机网络中广泛使用，并且已有实现HDLC规程的LSI器件出售。HDLC是从两个较早的标准SDLC(Synchronous Data Link Control)和ADCCP(Advanced Data Communication Control Procedure)发展起来的，由于这三个规程很类似，我们只讨论HDLC规程。

所有HDLC通信网都是一个主站(Primary Station)和一个或多个次站(Secondary Station)组成。主站负有控制整个网络的责任，它向次站发送各种控制命令，而次站只能按照来自主站的命令动作，并作出响应回答。主站可以选择任意一个次站交换数据，构成通信的双方。次站一旦被选中，既可以接收来自主站的信息，也可以向主站发送，于是开始了信息传送阶段。而各次站之间不能直接通信。主站和次站的分配是固定的。

不能动态地改变。HDLC 网络可以具有如图2-12所示的几种形式。

主站和次站之间的信息传送格式采用帧结构, 并进行 CRC 校验。如果接收端收到的信息被检验有错, 则将其作废, 并请求发送端重发。若发送端逾期得不到接收端的认可响应, 也将自动进行重发。如果接收端已接收到正确的信息, 则应给予认可响应。

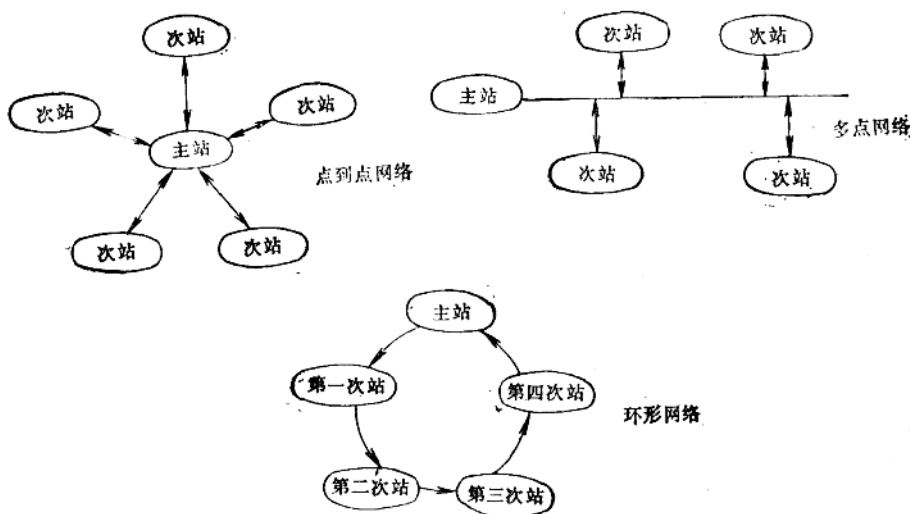


图2-12 HDLC网络的几种形式

每当发送端在帧结束之前需要中止发送时, 可以发出一个异常结束 (Abort) 字符, 接收端一旦收到 Abort 符号, 便把已经收到的信息作废。在 HDLC 规程中, 七个连续的“1”被作为失效字符。

帧内的信息不允许间断, 但帧与帧之间的间隔可以用标志符 (Flag 符) 或 Abort 符来填充。

当信息传送阶段结束时, 发送端可以用传送拆除命令的方法, 表明它希望结束传送阶段而转入空闲 (Idle)。当接收端收到连续15个“1”时, 也自动进入空闲状态。

图2-13为 HDLC 的帧结构。

开始标志 (F)	地址字段 (A)	控制字段 (C)	信息字段 (I)	帧校验序列 (FCS)	结束标志
01111110	8位或8位的整数倍	8位或16位	可变长度	16位CRC	01111110

图2-13 HDLC帧格式

(1) 开始标志 F (Flag)

开始标志共 8 位, 其具体编码为 01111110, 用来标明帧的开始。结束标志和开始标志相同, 因此 F 又称标志序列。两帧之间可用标志序列填充, 但帧内不允许填充。标志序列在接收过程中又作为同步字符使用。如果在数据信息中也有 01111110 的格式, 接收

器如何识别这究竟是数据信息还是标志序列呢? HDLC采用了一种特殊技术来保证标志01111110的唯一性。发送方在发送除标志序列以外的所有信息(包括地址字段、控制字段、信息字段和帧校验序列)时,只要遇到连续五个“1”,就自动插入一个“0”;反之接收方在接收除标志序列以外的所有信息时,如果连续接收到五个“1”,就自动将其后的“0”比特删除,以便恢复信息的原有形式。我们把这种方法称为“0”比特插入/删除技术,是由硬件完成的。“0”比特是由发送器插入而被接收器删除的,从而保证了只有标志序列才唯一具有两个“0”夹着六个“1”这种格式。按规程规定,插入的“0”不参加CRC校验。由于采用了“0”比特插入/删除技术,使HDLC具有较好的传输透明性,任何比特代码均可传送。

(2) 地址字段和控制字段

在标志F之后可以有一个地址字段(A)和一个控制字段(C)。HDLC规定地址字段可为8位或8的整倍数位,而控制字段为8位或16位。接收方必须检查每个地址字节的第一位,如果这一位为0,则后面跟着另一个地址字节;如果这一位为1,则该字节就是最后一个地址字节。同理,如果控制字段字节的第一位为0,则还有第二个控制字节,否则就只有8位控制字节。

(3) 信息字段

信息字段跟在控制字段之后,简称I字段。I字段包含有要传送的数据信息,但不是每一帧都必须有信息字段。HDLC传送数据是作为任意位长的串行位流传送的,其长度可以从0到存储器可以处理的最大位数。

(4) 帧校验字段

一个16位的帧校验序列(FCS)紧跟在信息字段之后。HDLC采用16位循环冗余校验码进行差错控制,其生成多项式为CCITT多项式:

$$X^{16} + X^{12} + X^5 + 1$$

除了标志序列和自动插入的“0”比特外,所有的信息都参加CRC计算。

(四) 串行数据传送的方向

在串行数据传送中,按信息流的方向,可分为三种方式。

1. 单工方式

单工方式,信息只能朝一个方向传送,即从发送端传送到接收端,如图2-14所示。

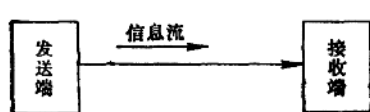


图2-14 单工方式



图2-15 半双工方式

2. 半双工方式

在半双工方式中,信息流可以在装置A和B之间交替传送,即信息可以从A端传送到B端,也可以从B端传送到A端,但这种传送不能同时进行。如图2-15所示。

在这种情况下,半双工连接的二个装置端口间必须互相配合,以控制数据的往返通道。