

送端开始发送包含信息的数据包或表明没有数据传送。接收端也要相应发送一个握手的数据包表明是否传送成功。发送端和接收端之间的 USB 数据传输，在主机和设备的端口之间，可视为一个通道。存在两种类型的通道：流和消息。流的数据不像消息的数据，它没有 USB 所定义的结构，而且通道与数据带宽、传送服务类型，端口特性（如方向和缓冲区大小）有关。多数通道在 USB 设备设置完成后即存在。USB 中有一个特殊的通道——缺省控制通道，它属于消息通道，当设备一启动即存在，从而为设备的设置、查询状况和输入控制信息提供一个入口。

事务预处理允许对一些数据流的通道进行控制，从而在硬件级上防止了对缓冲区的高估或低估，通过发送不确认握手信号从而阻塞了数据的传输速度。当不确认信号发过后，若总线有空闲，数据传输将再做一次。这种流控制机制允许灵活的任务安排，可使不同性质的流通道同时正常工作，这样多种流通常可在不同间隔进行工作，传送不同大小的数据包。

2.3.2 USB 数据流种类

USB 的结构包含四种基本的数据传输类型：

- 控制数据传送：在设备连接时用来对设备进行设置，还可对指定设备进行控制，如通道控制；
- 批量数据传送：大批量产生并使用的数据，在传输约束下，具有很广的动态范围；
- 中断数据的传送：用来描述或匹配人的感觉或对特征反应的回馈。
- 同步数据的传送：由预先确定的传送延迟来填满预定的 USB 带宽。

控制数据传送

当 USB 设备初次安装时，USB 系统软件使用控制数据对设备进行设置，设备驱动程序通过特定的方式使用控制数据来传送，数据传送是无损性的。

批量数据传送

批量数据是由大量的数据组成，如使用打印机和扫描仪时，批量数据是连续的。在硬件级上可使用错误检测可以保证可靠的数据传输，并在硬件级上引入了数据的多次传送。此外根据其它一些总线动作，被大量数据占用的带宽可以相应的进行改变。

中断数据传输

中断数据是少量的，且其数据延迟时间也是有限范围的。这种数据可由设备在任何时刻发送，并且以不慢于设备指定的速度在 USB 上传送。

中断数据一般由事件通告，特征及座标号组成，只有一个或几个字节。匹配定点设备的座标即为一例，虽然精确指定的传输率不必要，但 USB 必须对交互数据提供一个反应时间的最低界限。

同步传输

同步数据的建立、传送和使用时是连续且实时的，同步数据是以稳定的速率发送和接收实时的信息，同步数据要使接收者与发送者保持相同的时间安排，除了传输速率，同步数据对传送延迟非常敏感。所以同步通道的带宽的确定，必须满足对相关功能部件的取样特性。不可避免的延迟与每个端口的可用缓冲区数有关。

一个典型的同步数据的例子是语音，如果数据流的传送率不能保持，数据流是否丢失将取决于缓冲区的大小和损坏的程度。即使数据在 USB 硬件上以合适的速率传送，软件造成的传送延迟将对那些如电话会议等实时系统的应用造成损害。

实时的传送同步数据肯定会发生潜在瞬时的数据流丢失现象，换句话说，即使许多硬件机制，如重传的引入也不能避免错误的产生。实际应用中，USB 的数据出错率小到几乎可以忽略不计。从 USB 的带宽中，给 USB 同步数据流分配了专有的一部分以满足所想得到的速率，USB 还为同步数据的传送设计了最少延迟时间。

USB 通信流

USB 接口是一个基于令牌包 (token-based) 的总线协议，对于 USB 的通信，USB 规范引入了一种管道的概念。在整个 USB 的通信中包含了一个大的管道 (12Mbps)，这个大管道可以分为高达 127 个小的管道，每个小管道连接到一个 USB 的设备上。由于在 USB 令牌包中都含有 7 个用来寻址的位 (位于地址数据域，ADDR)，因此最多可寻址到 128 个设备，但是由于 00000000B 地址是默认地址，且用来指定给所有刚连上的设备。这也就是为什么 USB 最多能连接到 127 个设备的原因，而每一个连接到设备的小管道又可再细分为许多的微管道。由于在令牌包中包含了 4 位的端点 (Endpoint) 地址 (位于端点数据域，ENDP) 以

及一个输入/输出位，所以在一个小管道内最多可再分割成 16 组微管道，也就是可对 16 个输出/输入点的端点（共 32 个端点）寻址。

下面对端点和管道的概念进行一些解释。

1. 设备端点 (End Point)

端点是一个 USB 设备唯一可以确认的部分，它是主机和设备之间通信流的终点。USB 为主机上的客户软件与 USB 功能模块之间的通信提供了服务。端点可以决定端点和客户软件之间通信所需要的传输服务类型。一个端点具有以下一些属性：

- (1) 端点号
- (2) 总线频率/延时要求
- (3) 带宽要求
- (4) 差错控制要求
- (5) 端点可以接收或传递的最大分组
- (6) 端点的传送类型
- (7) 端点和主机之间的数据传送方向

(8) 端点 0 是 USB 设备的缺省端点，所有 USB 设备都必须拥有端点 0，该端点用于对 USB 设备进行配置（初始化）。端点 0 提供了对设备配置信息的访问权，通过它还允许访问 USB 状态和控制操作。端点 0 总是在设备接入和上电时就立即进行配置。

(9) 除端点 0 外，功能设备还具有其他的端点。低速功能设备有两个端点可供选择。而对于全速率设备来说，它的附加的端点数仅受到协议的限制，最多可有 16 个输入端点和 16 个输出端点。

在对端点进行配置之前，端点处于一种不确定的状态。所以一个端点只在对其进行配置之后，主机才能访问它。包括端点 0 在内的所有端点，都作为设备配置过程中的普通对象来对其进行配置。

2. 管道

一个 USB 管道是设备上的一个端点和主机上的软件的联合体。管道表示经过一个存储器缓冲区和一个设备上的端点，可以在主机上的软件之间传送数据的能力。

USB 并不对管道中传递的数据内容进行翻译,即使是消息管道要求根据 USB 的规定对数据进行打包,USB 也不会翻译这些数据的内容。

对一个 USB 设备进行配置之后,就会形成管道。由于一个 USB 设备上电后总要对端点 0 进行配置,所以端点 0 总是拥有一个管道。该管道称为缺省管道。系统软件利用该管道来识别设备和确定配置要求,并对设备进行配置。当设备配置完毕后,该设备的专用软件也可以使用缺省管道。同时 USB 系统软件也保留对缺省管道的“所有权”,并由它来协调其他客户软件对该管道的使用。

数据的编码与解码

在包传送时,USB 使用一种 NRZI(None Return Zero Invert,即无回零反向码)编码方案。在该编码方案中,“1”表示电平不变,“0”表示电平改变。图 2-7 列出了一个数据流及其它的 NRZI 编码,在该图的第二个波形图中,一开始的高电平表示数据线上的 J 态,后面就是 NRZI 编码。

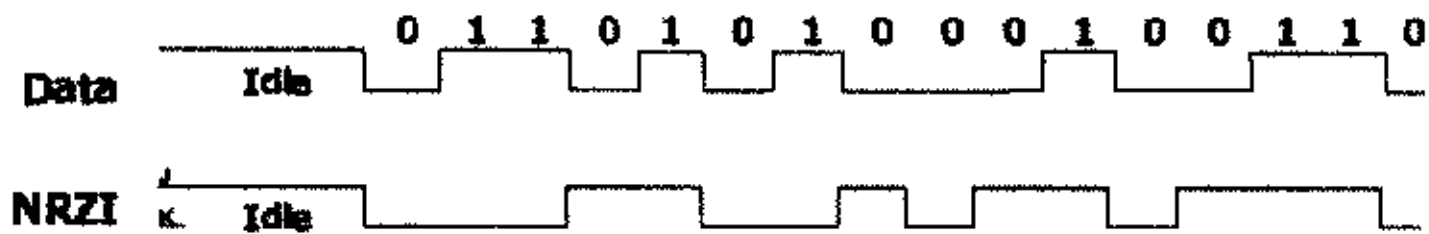


图 2-7 NRZI 数据编码

为了确集信号发送的准确性,当在 USB 上发送一个包时,传送设备就要进行位插入操作。所谓位插入操作是指在数据被编码前,在数据流中每六个连续的‘1’后插入一个‘0’,从而强迫 NRZI 码发生变化,如图 2-8 所示。

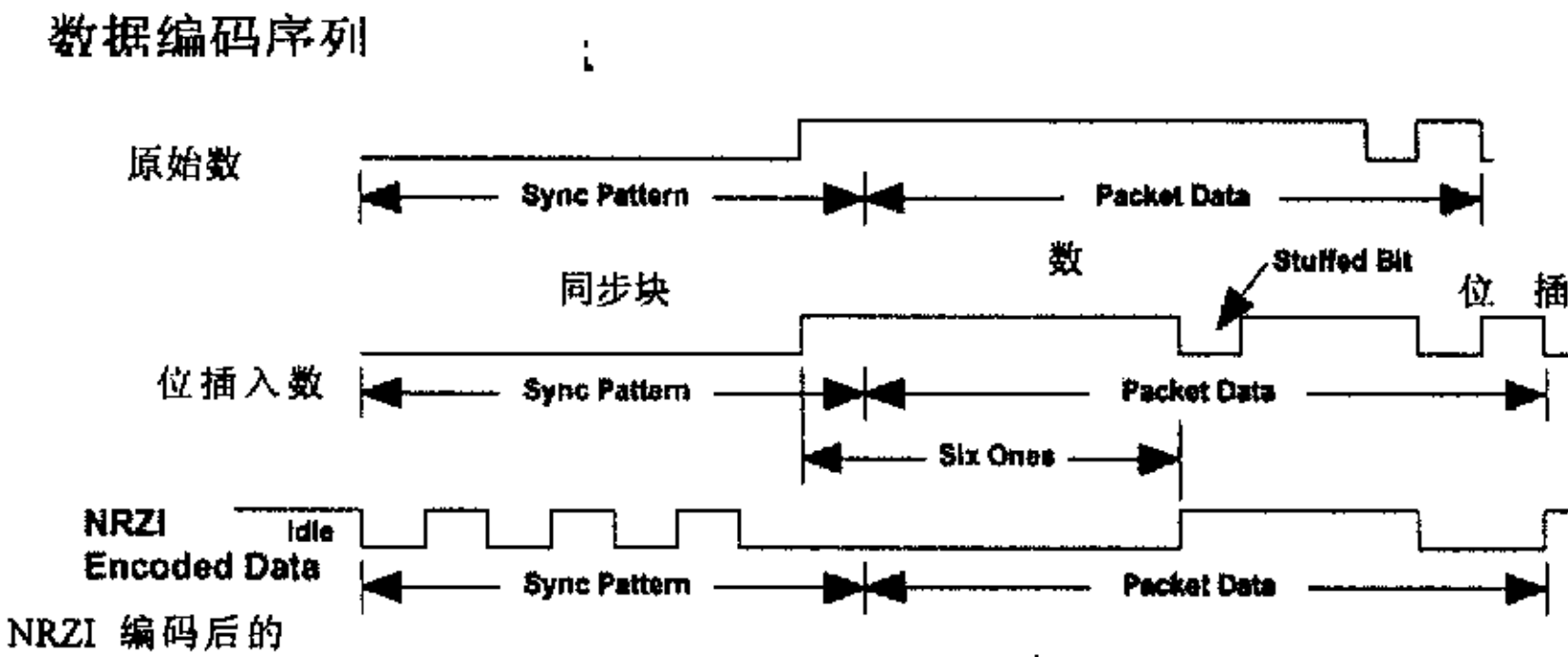
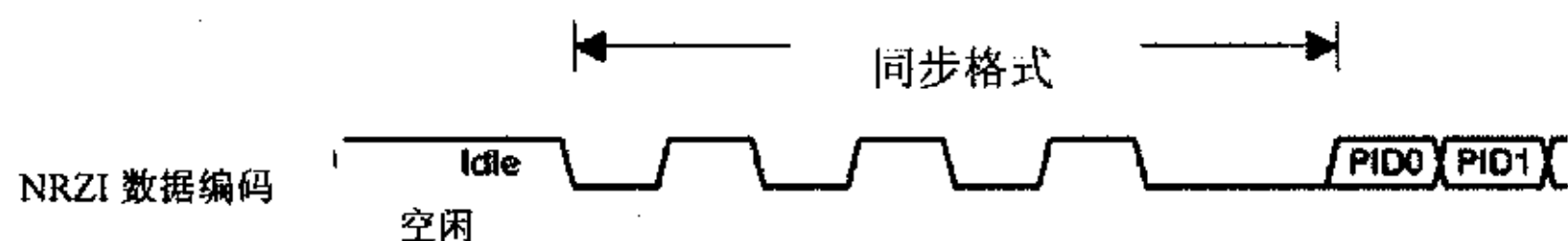


图 2-8 位插入

位插入操作从同步格式(如图 2-9 所示)开始,贯穿于整个传送过程,在同步

格式端的数据‘1’作为真正数据流的第一位。位插入操作是由传送端强制执行的，是没有例外的。如果严格遵守位插入规则，甚至在 EOP 信号结束前也要插入一位‘0’位。接收端必须能对 NRZI 数据进行解码，识别插入位并去掉它们。如果接收端发现包中任一处有七个连续的“1”，则将会产生一个位插入错误，该数据包将被忽略。



关于位的插入有一个特例，那就是刚好在 EOP 前的时间间隔，EOP 前的最后一个数据位可能被集线器的转换偏移而拉长，这种情况如图 2-10 所示。

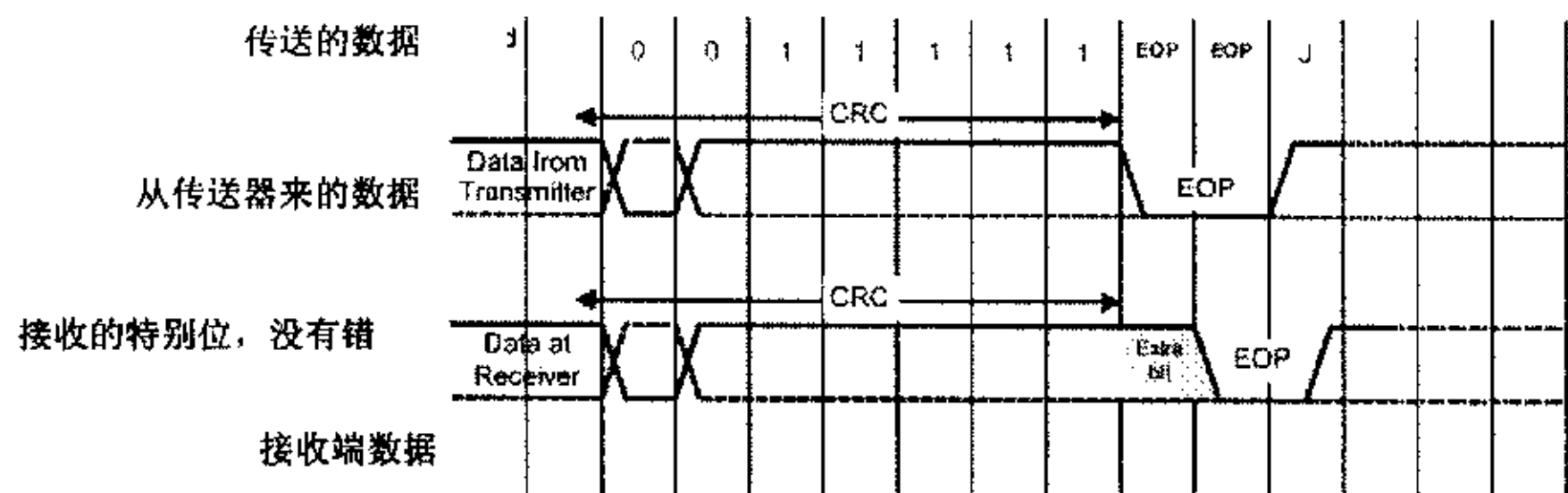


图 2-10 对 EOP 前的特别位的说明

2.4 USB 设备与枚举过程

2.4.1 USB 设备

USB 设备分为诸如集线器、分配器或文本设备等种类。集线器类指的是一种提供 USB 连接点的设备，USB 设备需要提供自检和属性设置的信息，USB 设备必须在任何时刻执行与所定义的 USB 设备的状态相一致的动态。

当设备被连接、编号后，该设备就拥有一个唯一的 USB 地址。设备就是通过该 USB 地址被操作的，每一个 USB 设备通过一个或多个通道与主机通讯。所有 USB 设备必须在零号端口上有一指定的通道，每个 USB 设备的 USB 控制通道将与之相连。通过此控制通道，所有的 USB 设备都列入一个共同的准入机制，以获得控制操作的信息。