```
unsigned char len);                                    //写端点
```

以上所有命令都是通过调用底层函数实现的。

5.1.3 中断服务程序

5.1.3.1 FPGA 模块中断服务程序

这部分程序在 USB 上传数据模块中实现，它对 FPGA 模块上传数据时产生的中断进行处理。在中断处理程序中，单片机循环向 FPGA 模块发送读脉冲信号，然后接收 64 字节的数据到内部缓冲区并设立标志位。在主程序中，通过主端点将接收到的数据发送到 D12 的数据缓冲区里面。在数据发送完成后，主程序清除该标志位以便接收新的数据包[17]。在这段代码中，采用了汇编语言编写以节省时间，保证接收效率。流程图如图 5-2 所示：

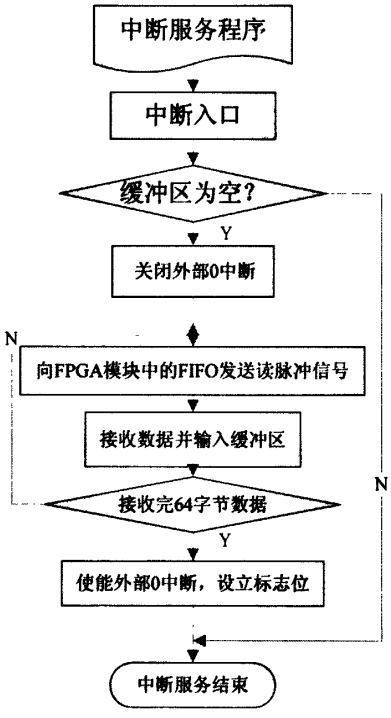

图 5-2 FPGA 模块中断服务程序流程图

5.1.3.2 D12 中断服务程序

这部分程序对 D12 产生的中断进行处理。它将数据从 D12 的内部 FIFO 取回到 CPU 存储器，并建立正确的事件标志以通知主循环程序进行处理[18]。在 ISR 函数的入口使用 Read Interrupt Register()来决定中断源，根据中断源进入相应的子程序进行处理。ISR 从 D12 收集数据，而主循环对数据进行处理。当 ISR 收集了足够的数据时，它只通知主循环已经准备好等待处理，这将减少主循环不必要的服务时间并且简化了主循环的编程。中断服务程序流程如图 5-3 所示：

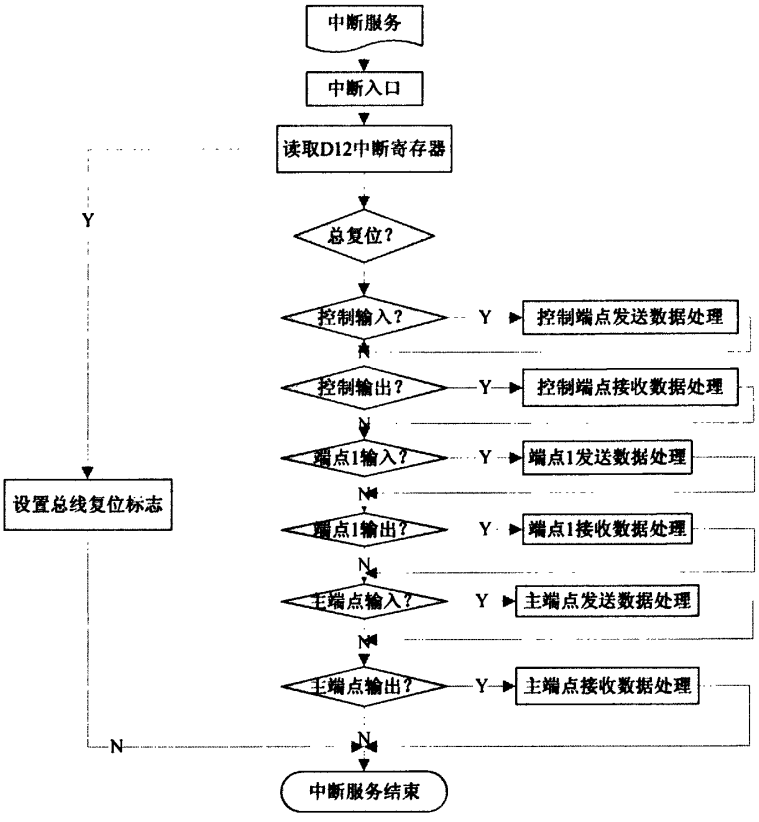

图 5-3 D12 中断服务程序流程图

1）控制输出端点处理程序

控制传输总是在 SETUP 阶段开始，之后为可选的 DATA 阶段。然后在 STATUS 阶段结束。因此主机下放的所有指令，都应以控制输出作为传输的起始。控制输出模块的流程如图 5-4 所示：

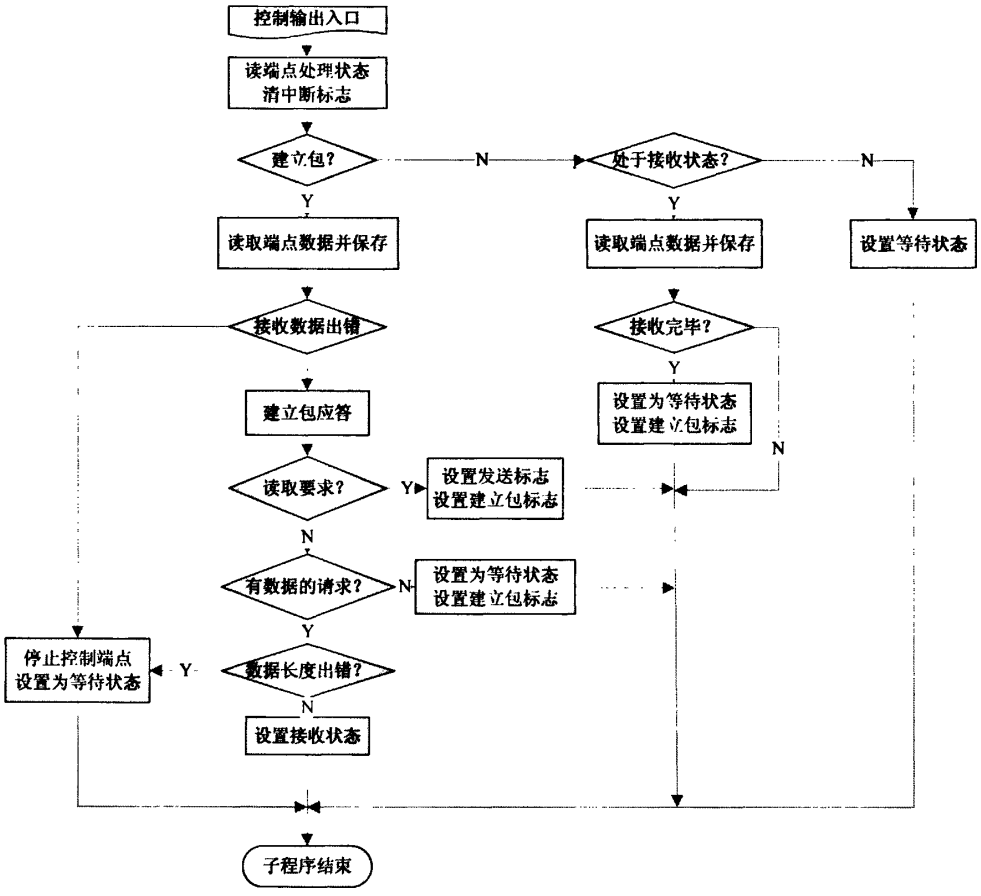

图 5-4 控制输出端点处理程序

当 D12 接收到建立包，将产生一个中断通知单片机使它响应中断，通过读 D12 中断寄存器决定包是发到控制端点还是普通端点。如果包是送往控制端点，MCU 需要通过读 D12 的最后处理状态寄存器，进一步确定数据是否是一个建立包。第一个包必须是建立包。

从图 5-4 流程图中，单片机需要通过选择控制输出端点，来提取建立包的内容，进而决定端点是为满，还是空。如果控制端点为满，单片机将从缓冲区读出内容并将其存入存储区。之后它将从存储区读出，并使主设备请求生效。如果是一个有效的请求，单片机必须向控制输出端点发送应答建立命令，重新使能下一个建立阶段。接下来单片机需要证实控制传输是控制读还是写。这可以通过读建立包中 bmRequestType 的第 8 位来实现，如果控制传输是一个控制读类型，那就是说器件需要在下一个数据阶段向主机发回数据包，单片机需要设置一个标志变量来指示 USB 设备现在正处于传输模式，即准备在主机发送请求时发送数据[19]。

2）控制输入端点处理程序

建立阶段结束之后，主机会执行数据阶段。D12 等待接收 Control In 包过程如图 5-5 控制输入模块流程图所示。单片机首先需要通过读 D12 的最后处理状

态寄存器，来清零 Control In 中断标志位。接着单片机在确认 D12 处于传输模式后进行数据包的发送。

由于 D12 的控制端点只有 16 字节 FIFO，如果传输的长度大于 16 字节，单片机在传输阶段就必须控制数据的数量。如流程图 5-5 所示，单片机必须检查要发送到主机的当前和剩余的数据大小。如果剩下的字节数大于 16，单片机将先发送 16 个字节并用减去 16。当下一个 Control In 标志来到时，单片机将确定剩余的字节是否为零。如果已经没有数据要发送，单片机需要发送一个空的包以指示主机数据已经发送完毕。

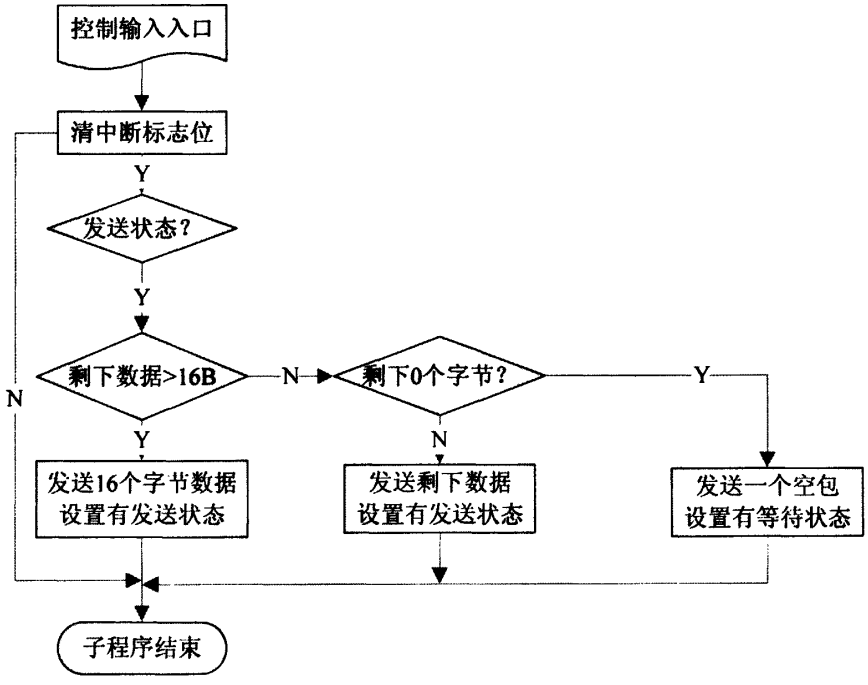

图 5-5 控制输入端点处理程序流程图

3）普通端点输入输出处理程序

端点 1 和端点 2 输出端点配置为从主机接收数据包，当单片机通过读取中断寄存器识别到从 PC 主机接收到输出中断标志，并清零 D12 的中断位。选择将清零缓冲区，然后单片机确认数据长度并把数据读出，然后设立收到数据标志位；对于输入端点的中断，只需要读取端点号的最后处理状态寄存器来判断中断寄存器的相应位，并把状态清零[20]。输入端点中断是在发送完数据后才产生，当还有数据要发送时，可以在中断程序中接着发送，数据发送完成标志是在主机的一个有效的 IN 事务后发生的，当主机发出 IN 事务而设备没有发出数据时是不会产生中断的。程序流程图如图 5-6 所示：

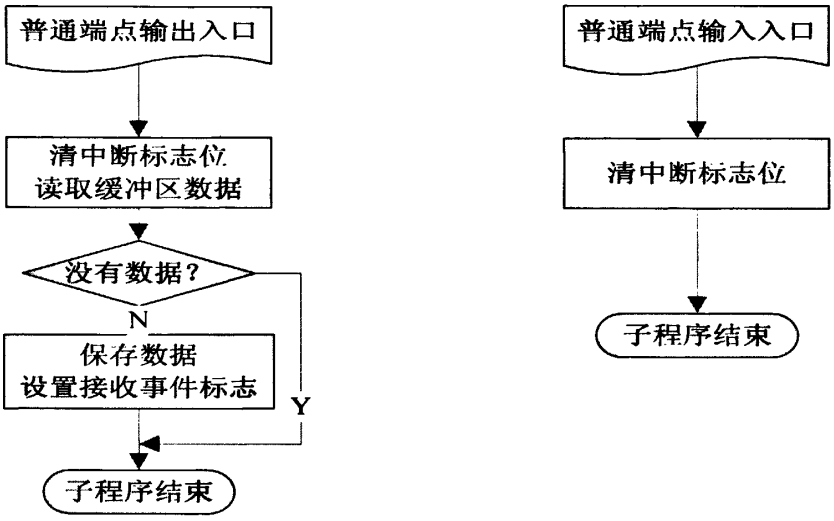

图 5-6 普通端点处理程序流程图

5.1.4 设备标准请求与厂商请求

在主循环中，当建立包标志为真时，利用 control handlero 函数进入然后根据不同的 bmRequestType 和 bRequest 值，进入不同的程序。若 bmRequestType 值是 Standard Request，那么主机发出的是 USB 的标准请求。通过(*StandardDeviceRequest[req])()散转程序来分别进入标准请求相应的处理函数中；如果 bmRequestType 值是 USB Vendor Request，则通过(*VendorDeviceRequest[req])()进入厂商请求相应的处理函数中。这里涉及的函数比较多，但是结构都比较简单明了，就不一一介绍。

5.1.5 主程序

主循环检查事件标志并进入对应的子程序进行进一步的处理。在主循环中，单片机首先对其所有端口、存储区、定时器和中断服务程序进行初始化，之后单片机将重新连接 USB，包括将 SoftConnect 寄存器设置为 ON。这些过程是很重要的，它确保了在单片机准备好服务 D12 之前，D12 不会进行操作。初始化后，进入循环，查询各种状态。

在下载数据时，PC 机通过端点 1 把要发送的数据包的数量大小发给此 USB 数据下载模块，同时通过主端点发送数据给此模块。然后 USB 下载模块接收主端点数据包并发送给 FPGA，同时使能它与 FPGA 之间的写信号，并维持到数据包发送完毕。数据下载主程序流程图如图 5-7 所示：