4. FAT 文件系统函数集—Fat.c 和 Fat.h

实现了精简的 FAT 文件系统，以此来访问和管理 U 盘中的文件，操作 FAT 文件时对 FAT 表进行定位、搜索和读写等操作。

5. USB 协议层—SL811.e 和 SL811.h

除了包括 USB 主机接口芯片 SL811 的初始化程序 SL811_Init()，该部分源代码文件中还实现整个 USB 主机协议栈。USB 最底层的数据结构是包，函数 usbXfer0 实现 USB 的设置、输入和输出事务功能；由于此嵌入式主机系统仅支持 USB Mass Storage 类，只需支持 USB 的控制传输和批量传输，因此包括 USB 控制传输函数 ep0Xfer()和批量传输函数 epBulkSend()、epBulkRcv()；主机系统要枚举设备需实现标准的 USB 请求命令，这些命令是基于控制传输实现的，包括获取描述符请求命令 GetDesc()、设置设备地址请求命令 SetAddress()、枚举设备函数 EnumUsbDev()、设置配置函数 Set_Configration()。

6. 硬件提取层—Halfat.c 和 Halfat.h

最底层的读写操作函数，包括读写数据函数 SL811Read0、SL811Write()和读写批量数据函数 SL811BufRead()、SL811BunWrite()。单片机通过数据线、片选线及读写控制线完成对 SL811 的读写操作。

5.3 FPGA 模块程序的编写

FPGA 模块包含了两部分的功能电路，一个是 CPLD 实现的 FPGA 配置电路的控制器，一个是 FPGA 内部实现数据处理程序。这两部分程序都是采用 VHDL 高速硬件描述语言实现的，通过 Xilinx 公司提供的 ISE 平台来实现对设计的输入、逻辑综合、布局布线以及生成芯片的配置文件，采用 Mentor Graphics 公司的 ModelSim 软件进行程序仿真。

5.3.1 CPLD 配置控制电路的工作原理

CPLD 控制配置电路工作如下：首先，由于 AT49F002 的快速读取时间大约为 55ns，所以将板上的时钟进行 4 分频作为编程时钟。在每次上电后，CPLD 先拉低PROGRAM引脚一定的时间。当PROGRAM被拉低后，FPGA 向INIT和 DONE 引脚输出低电平。当 CPLD 检测到 DONE 低电平后，CPLD 使能 Flash RAM 的读信号、片选信号同时屏蔽其复位信号和写信号。然后，等待PROGRAM和INIT引脚由低变高后，CPLD 使能 FPGA 的片选线CE和写信号 WE，之后 FPGA 在编程时钟的上升沿接收配置数据[21]。CPLD 内部计数器依次增加 Flash RAM 的地址，FPGA 将接受相应单元的配置数据写入。

在 FPGA 处于编程状态时 DONE 引脚一直处于低电平，编程完成后 DONE 管脚