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

被 FPGA 驱动成高电平[22]。当 CPLD 检测到 DONE 拉高之后，它控制 Flash RAM 的控制信号，让 Flash RAM 输出高阻态，同时将外部晶振时钟直接输出作为 FPGA 的全局工作时钟。如果在配置的过程中出错，当PROGRAM被拉低时，CPLD 将使 Flash RAM 的地址计数器恢复为零，从而重新开始一轮配置过程。

5.3.2 FPGA 片内数据接收和处理模块的设计

FPGA 片内程序是此系统数据处理的主要模块，不仅实现接收外部的数据并将它们送入外部 SDRAM，而且可以提供一个接口给用户的算法处理模块连接，同时可以实现在数据处理完之后，再将数据从 SDRAM 中读取出来送给外部连接的电路。片内程序包括了三个主要电路模块，FIFO 模块、SDRAM 控制器和主控制模块，还有一个数据与信号选择模块。由于其中的数据与信号选择模块比较简单，它只实现数据和信号的选择，所以下面就对前三个主要模块分别进行详细介绍，然后再对整体的连接框架进行介绍[23]。

5.3.2.1 FIFO 模块的实现

在本系统设计中，数据信息的传递和交互是非常频繁的，而且存在着不同工作速率的器件，因此，在外部数据与 FPGA 通信之间都加入 FIFO，这样可以使数据进行一定的缓冲[24]。在 ISE 软件里面集成有许多 FIFO 的 IP 核资源，因此我们就直接采用里面的 IP 核来设计。由于系统中的每一个 USB 模块的数据包的大小都是 64 字节，而且为了数据传输的正确性，我们选择了一个具有 64 位深度和 8 位数据宽度的同步 IP 核进行设计，这样可以保证在每个数据包被取走之后另一个数据包才可以写入。

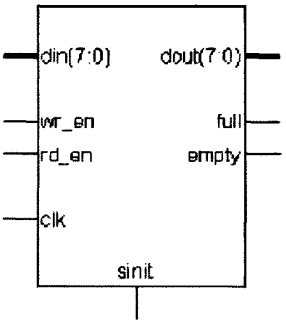

图 5-11 FIFO 的综合的电路框图

此FIFO的综合的电路框图如图5-11所示：

可以看出，它带有 wr_en 写使能引脚和 rd_en 读使能引脚，但只有一个 clk 时钟引脚。因此在写有效的时候，clk 就做为数据写入控制信号，在每个时钟上升沿写入在 din 引脚上面的数据。当读有效的时候，clk 作为数据读出控制信号，同样在每个时钟上升沿从 dout 引脚送出数据。当数据写满 FIFO 的时候，full 引脚就输出高电平。当 FIFO 里面的数据为空是，empty 引脚就输出高电平[25]。

5.3.2.2 SDRAM 控制器的设计

SDRAM 的控制逻辑比较麻烦，对时序要求也十分严格，使用很不方便，这就要求有一个专门的控制器，使得在电路中可以对 SDRAM 操作变得简单。因此，在