

基于SPI协议的SD卡读写机制与实现方法

张洪涛, 莫文承, 李兵兵

(西安电子科技大学ISN国家重点实验室, 陕西 西安 710071)

摘要: 介绍了基于SPI协议的SD卡与单片机的连接方法, 分析了SPI总线模式, 并用软件模拟SPI总线时序的方法实现了单片机与SD卡的SPI总线接口通信。

关键词: SD卡; 串行外设协议; 时序; 读写机制

0 引言

SD卡是一种多功能存储卡, 具有较快的传输速度和较大的存储容量, 同时, SD存储卡还向上兼容MMC卡。SD卡内置控制电路, 可用于手机、数码相机、MP3、PDA等多种数字设备。为此, 本文首先介绍了单片机在SPI协议下与SD卡硬件接口电路, 然后简要阐述了SPI总线模式以及软件模拟的SPI总线字节传输, 最后给出了具体实现SD卡的初始化和读写的软件流程。

1 硬件电路设计

SD卡有两种总线模式, 即SD总线模式和SPI总线模式。其中SD总线模式采用四条数据线并行传输数据, 数据传输速率高, 但是传输协议复杂, 只有少数单片机才提供有此接口, 而用软件方法模拟SD总线又比较繁琐, 并会降低SD卡的数据传输速率; 而SPI总线模式只有一条数据传输线, 数据传输速率较低, 但绝大多数中高档单片机都提供SPI总线, 也易于用软件方法来模拟, 此外, SPI总线模式的传输协议简单, 易于实现。为此, 本设计采用SPI总线模式。

本系统中的接口电路采用的是W78LE58单片机。该单片机不具有SPI模式, 但可通过软件编程实现SPI模式的数据传输(包括串行时钟、数据的输入和输出)。在SPI模式下, 单片机与SD卡的连接主要有四根线(包括时钟线, 两根数据传输线和一根片选线)。其硬件连接如图1所示。

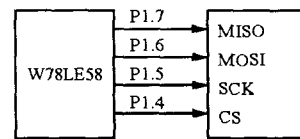


图1 单片机与SD卡的硬件连接图

2 SPI总线模式

SPI (Serial Peripheral Interface, 串行外围设备接口总线) 总线技术是MOTOROLA公司推出的一种同步串行总线接口, 它是目前单片机应用系统中最常用的几种串行扩展接口之一。SPI总线主要通过三根线进行数据传输: 同步时钟线SCK, 主机输入/从机输出数据线MISO、主机输出/从机输入数据线MOSI, 另外还有一条低电平有效的从机片选信号CS。SPI系统的片选信号以及同步时钟脉冲由主机提供。

SPI总线模式的数据是以字节为单位进行传输的, 每字节为8位, 每个命令或者数据块都是字节对齐的(8个时钟的整数倍)。主机与SD卡的各种通信都由主机控制, 主机在对SD卡进行任何操作前都必须先要拉低SD卡的片选信号CS (card select), 然后由主机向SD卡发送命令, SD卡对主机发送的任何命令都要进行响应, 不同的命令会有不同的响应格式(1字节或2字节响应)。SD卡除了对命令响应外, 在执行写操作时, 还要对主机发送的每个数据块进行响应(向主机发送一个特殊的数据响应标志)。

SPI总线模式下的所有命令都是由6个字节构成, 且发送时高位在前, 其命令格式如表1所列。其中, 7位CRC (Cyclic Redundancy Check——循

收稿日期: 2007-07-12

表1 SPI总线模式下的命令格式

Byte1		Byte2-5			Byte6	
7	6	5-0			7	0
0	1	Command	Command Argument	CRC	1	

环冗余校验) 校验位可以全部写入0, 因为默认情况下, SPI总线模式无需CRC校验。

3 软件模拟SPI总线模式下的字节传输

由于W78LE58单片机没有SPI接口, 因此, 应使用软件来模拟SPI的操作, 包括串行时钟和数据输入、输出。图1中的P1.5就是模拟SPI的时钟(SCK) 输出端, P1.4模拟SPI的从机选择端, P1.6模拟单片机的数据输出, P1.7模拟SD卡的数据输出。在将一个字节的数据(累加器ACC中的数据) 送往SD卡时, 先置时钟SCK为0, 以将累加器ACC中的最高位左移至进位C, 然后将进位C中的数据送至单片机数据输出口P1.6, 并置时钟SCK为1, 至此, 模拟1位数据输出即算完成。依此循环8次, 就可完成1次通过SPI传输8位数据的操作。从SD卡读取一个字节的数据至单片机的累加器ACC与写SD卡类似。

单片机写1字节数据(累加器ACC中的数据) 至SD卡的子程序如下:

WR_SD:

```

MOV R2, #8 ; 置循环次数
WR: CLR SCK ; 使时钟输出为0
NOP ; 延时
RLC A ; 左移累加器ACC最高位至C
MOV P1.6, C ; 进位C送入数据输出端
SETB SCK ; 使时钟输出为1
DJNZ R2, WR ; 判断是否循环8次(8位

```

数据)

RET

单片机从SD卡读1字节数据(至累加器ACC中) 的子程序如下:

RE_SD:

```

MOV R2, #8
RE: CLR SCK
NOP
MOV C, P1.7
RLC A

```

```

SETB SCK
DJNZ R2, RE
RET

```

4 软件设计

SD卡的软件设计主要包括两部分内容: SD卡的上电初始化过程和对SD卡的读写操作。

4.1 SD卡的初始化

SD卡从上电到对SD卡进行正确的读写操作需要一个上电初始化的过程。上电初始化的流程图如图2所示。

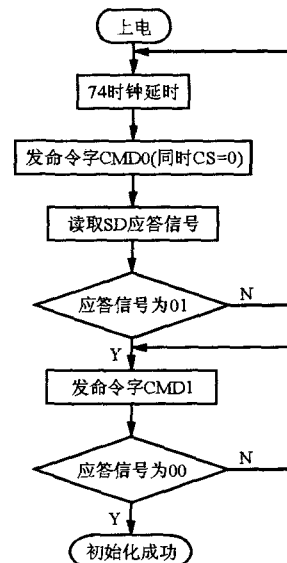


图2 SD卡的上电初始化流程

SD卡上电后, 主机必须先向SD卡发送74个时钟周期, 以完成SD卡上电过程。SD卡上电后会进入SD总线模式, 并在SD总线模式下向SD卡发送复位命令(CMD0), 若此时片选信号CS处于低电平, 则SD卡进入SPI总线模式, 否则SD卡工作在SD总线模式。SD卡进入SPI工作模式会发出应答信号, 若主机读到的应答信号为01, 即表明SD卡已进入SPI模式, 此时主机即可不断地向SD卡发送命令字(CMD1) 并读取SD卡的应答信号, 直到应答信号为00, 以表明SD卡已完成初始化过程, 准备好接受下一命令。此后, 系统便可读取SD卡的各寄存器, 并进行读写等操作。应当注意的是: 主机在向SD卡发送命令字CMD0时, SD卡是处于SD总线模式的, 此时要求每一个命令都要有合法的CRC校验位, (下转第47页)

储序列头信息 (sequence_header)、I帧或PB帧图像头信息 (picture_header)、片层头信息 (slice_header) 和宏块信息 (macroblock)。RAM2可存储熵解码器解码后的量化系数和量化系数游程。

3.5 指数哥伦布码表

在指数哥伦布解码时, me (v) 和ce (v) 两种解码方式在解码时都要用码表来完成对应数值的映射变换。由于me (v) 的码表比较小, 所以, 它可在变长码解码器模块内部定义。AVS标准中定义了与ce (v) 相关的19个码表, 这些量化码表可根据编码对象和编码方式的不同通过ROM其将分为亮度-帧内、亮度-帧间和色度三类。

4 仿真与综合结果

可使用Verilog HDL语言对本文所述结构进行实现, 并可使用Modelsim对其进行前仿真。在使用相同的输入视频测试码流文件的前提下, 前仿真的输出结果与AVS参考软件rm52c所输出的测

表2 综合结果列表

Total ALUTs	2,665/48352 (5%)
Total registers	1220
Total momery bits	298,182/2,544,192 (11%)

试结果完全吻合。仿真结果表明, 该实现方法完全符合AVS视频标准解码的要求。

在该实现通过功能仿真后, 便可使用Quartus II对该实现进行综合。选择器件类型为Stratix II系列的EP2S60F672C5ES时, 综合结果如表2所列。综合后的时钟可以达到81 MHz, 可以满足解码要求。

5 结束语

本文提出了一种适用于AVS熵解码器的VLSI实现方法。考虑到AVS硬件解码时的流水线操作, 本方法设计了高效的数据存储结构来方便后续模块的数据读取操作, 同时也满足了设计的资源与速度的要求, 完全可以应用于AVS视频解码芯片之中。

(上接第43页)

所以, 此时的命令字CMD0必须有正确的CRC校验位 (其校验位为95H)。而在发送命令字CMD1时, SD卡已处于SPI模式, 而默认的SPI模式无需CRC校验, 此时的CRC校验位可直接写入0。

4.2 SD卡的读写流程

SPI模式支持单块 (命令字为CMD24) 和多块 (命令字为CMD25) 写操作。多块写操作指的是从制定的位置开始写, 直到SD卡接收到一个停止命令 (命令字为CMD12) 才停止写操作。单块的写操作数据块的长度只能是512字节, 本设计采用的是单块写操作, 其流程图如图3所示。操作时, 可首先向SD卡发送写数据块的命令字CMD24, 在接收到SD卡的响应信号 (00) 后, 再发送数据起始标志 (0xFE), 然后发送512字节的数据, 并后跟两字节的CRC校验。当SD卡的回应信号为E5h时, 即表明SD卡可正确接收数据, 之后, SD卡的输出口变为低电平, 表明正在写SD卡, 当输出口变为高电平时表明写操作完成。SD卡的读操作与写操作相同, 故此不在赘述。

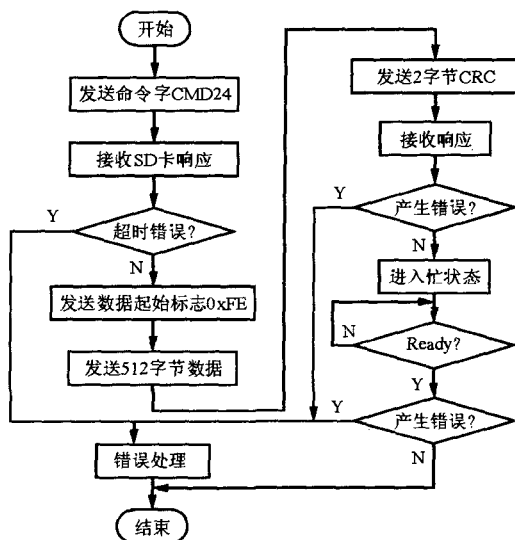


图3 单块数据写操作流程

5 结束语

该SD卡与W78LE58单片机的接口电路已在实际的试验板上成功验证, 并已建立了符合Windows标准文件格式的FAT16文件系统。实际使用证明。本设计可对SD卡上的文件进行各种操作。