

Interface Design between the DSP Chip TMS320VC5509 and Serial Flash Based on Mcbsp

MENG Bo^{1,2}, ZHU Ming²(1. Changchun Institute of Optics, Fine Mechanics and Physics, The Chinese Academy of Sciences, Changchun 130033, China;
2. Graduate School of The Chinese Academy of Sciences, Beijing 100039, China)

Abstract: In order to realize the SPI interface protocol between DSP chip TMS320VC5509 and the external serial Flash, the Multichannel Buffered Serial Port (McBSP) was adopted. It used the McBSP of TMS320VC5509 as the master of the SPI protocol, and chose the serial Flash AT25F1024 as the slave. The experiment results show that the system achieved the serial communication between DSP and Flash and also reduce its complexity effectively. The configuration of the McBSP with the CSL library functions routines was also given.

Key words: McBSP; DSP; serial flash; SPI protocol; CSL library function routines.

EEACC: 6150

基于 McBSP 实现 DSP 与串行 Flash 之间的接口通讯

孟 勃^{1,2}, 朱 明¹(1. 中国科学院 长春光学精密机械与物理研究所, 长春 130033;
2. 中国科学院 研究生院, 北京 100039)

摘 要: 利用多通道缓冲串口 McBSP0 实现了 TMS320VC5509 DSP 与外部串行 Flash 之间的 SPI 通讯。完成了用 DSP 控制 Flash 进行在线系统编程。有效地降低了系统设计的复杂性, 节省了空间。重点介绍了 McBSP 与 Flash 之间串行接口的设计, 并介绍了 TMS320VC5509 串行 8 位引导装载的实现方法。同时给出用 CSL 库函数实现 McBSP 配置的程序。

关键词: McBSP; 串行 Flash; SPI 协议; CSL 库函数

中图分类号: TP311

文献标识码: A

文章编号: 1005-9490(2006)03-0920-05

现代的定点和浮点 DSP 都具有一个串行或并行存储器接口。串行接口与并行接口相比的主要优点之一是通过采用 3 线接口来减少引脚数, 以便连接微控制器与外设。此方法是有效的, 特别是当串行传送速率大于变换器或外设所需的数据带宽时更是如此。串行传送在一个周期内被传送的数据是以位串行方式传送, 首先传送的是变换数据的最高有效位(MSB)或最低有效位(LSB)。在每一个串行时钟周期一位一位的传送下去, 直到 N 位变换数据的

所有位传送完为止。这样大大减少了使用的空间, 降低了成本^[1]。

串行 Flash 存储器大多采用 I2C 接口或 SPI 接口进行读写; 与并行 Flash 存储器相比, 所需引脚少、体积小、易于扩展、与单片机或控制器连接简单、工作可靠, 所以串行 Flash 存储器越来越多地用在各类电子产品和工业测控系统中^[2]。

本系统利用 TMS320VC5509 内部的多通道缓冲串口(McBSP)来完成 DSP 与外部串行 Flash 之间的 SPI 通讯。其中 McBSP0 作为 SPI 协议的主设备, 外

收稿日期: 2005-10-17

作者简介: 孟 勃(1980-), 女, 吉林省吉林市人, 博士研究生 主要研究方向为数字图像的采集与处理, mengbo_nannan@sohu.com;
朱 明(1964-), 男, 江西南昌人, 研究员, 从事数字图像处理、电视跟踪和自动目标识别技术的研究。

部 Flash 作为 SPI 协议的从设备。有效的降低了系统设计的复杂性, 减少了成本, 节省了空间。

1 串口设计

1.1 McBSP

McBSP 的结构框图如图 1 所示。可以分为数据通道和控制通道两部分^[3]。数据发送引脚 DX 负责数据的发送, 数据接收引脚 DR 负责数据的接收, 发送时钟引脚 CLKX, 接收时钟引脚 CLKR, 发送帧同步引脚 FSX 和接收帧同步引脚 FSR 提供串行时钟和控制信号。CPU 和 DMA 控制器通过外设总线与 McBSP 进行通信。当发送数据时, CPU 和 DMA 将数据写入数据发送寄存器(DXR1, DXR2), 接着复制到发送移位寄存器(XSR1, XSR2), 通过发送移位寄存器输出至 DX 引脚。同样, 当接收数据时, DR 引脚上接收到的数据先移位到接收移位寄存器(RSR1, RSR2), 接着复制到接收缓冲寄存器(RBR1, RBR2), RBR 再将数据复制到数据接收寄存器(DRR1, DRR2)中, 并通过串口事件通知 CPU 或 DMA 读取数据。这种多极缓冲方式使得片内数据通信和串行数据通信能够同时进行。

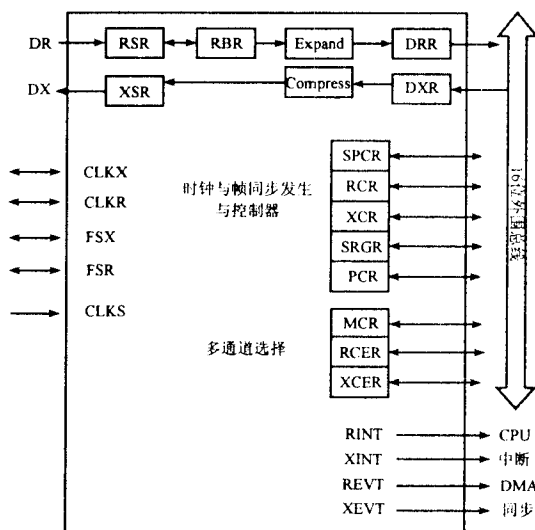


图 1 McBSP 的结构框图

1.2 SPI 协议

串行外设接口 SPI 是 Motorola 公司推出的一种同步串行接口。是目前使用比较多的串行总线接口标准。它是一种通用串行接口总线, 字长为 8bit, 用来与外部设备(例如 EEPROM、A/D 转换器等)进行通信。SPI 接口利用 SCK、SI 和 SO 三根线进行数据的读/写。其中, SCK 为时钟信号, SI 和 SO 为数据输入和输出线^[2]。

SPI 协议是一种主从式配置^[4]。包括一个主设备和一个或多个从设备。对 McBSP 进行适当的配

置, 可以使其工作于 SPI 方式。SPI 接口有四个信号:

串行数据主入从出信号: MISO (Master-In, Slave-Out);

串行数据主出从入信号: MOSI (Master-Out, Slave-In);

串行时钟信号: SCK;

从设备使能信号: /SS。

1.3 McBSP 与串行 Flash 之间的 SPI 接口设计

SPI 接口共有四种操作模式。SPI 操作模式决定了设备接收和发送数据时的时钟相位和极性, 即决定了时钟信号的上升和下降沿与数据流行方向之间的关系。在这里, 将 TMS320VC5509 的 McBSP0 作为 SPI 协议的主设备。McBSP 的时钟停止模式与 SPI 协议兼容^[5]。SPI 协议的从设备为 AT25F1024。AT25F1024 是 ATMEL 公司的一种 SPI 接口的 8 位串行 Flash。容量为 1 Mbit, 按照字节寻址读写。它支持 SPI 协议中的模式 0(0, 0) 和模式 3(1, 1)。在这两种模式下, SCK 信号的上升沿触发数据输入, 下降沿触发数据输出。二者的区别是 SCK 信号的起始电平不同。具体连接如图 2。

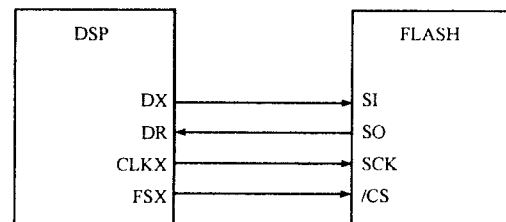


图 2 DSP 与 Flash 连接框图

这里采用的是时钟停止模式 3。即将 McBSP 的 Serial Port Control Register1 (SPCR1) 中的 CLKSTP 位设置成 11b, 将 Pin Control Register (PCR) 中的 CLKXP 位设置为 1b 的模式。也即时钟开始于下降沿, 有延时。当作为 SPI 主设备时, McBSP 内部的采样率发生器驱动时钟 CLKX 输出, FSX 输出作为从设备使能信号。实际上, 在 McBSP 内部会产生连续的 CLKX 时钟, 然后由使能信号选通后输出, 从而实现接口需要的时钟停止模式。因此在 McBSP 一端, 对收/发的内部操作而言, 时钟信号是连续的, 需要设置 SRGR 寄存器的 CLKSM 位选择采样率发生器的时钟源, 由 CLKGDV 位设置需要的 SPI 数据传输速率。选择时钟停止模式时, SRGR 寄存器中的帧设置位(FPER 和 FWID)无意义。由于 McBSP 产生 CLKX 和 FSX 信号输出。因此需设置 CLKXM = FSXM = 1。此时, 还需要设置 SRGR 寄存器中的 FSMG = 0, 即每次进行 DXR 到 XSR 的拷贝操作时要产生 FSX 信号。McBSP 在 CLKX 下降沿的半个时钟

周期之前发送数据,在 CLKR 的下降沿读取数据。时序如图 3:

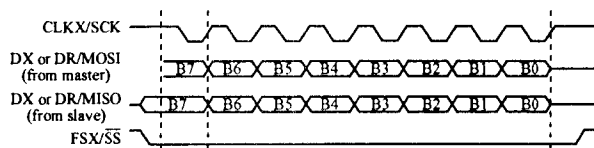


图 3 SPI 操作模式 3 时序图

将 McBSP 配置成 SPI 接口需要遵循以下步骤:

- ① 将接收器和发送器设置为复位状态。将 Serial Port Control Register2 (SPCR2) 中的发送器复位位置 0 (XRST=0) 以便复位发送器。将 SPCR1 中的接收器复位位置 0 (RRST=0), 以复位接收器。
- ② 将采样率发生器置位。清除 SPCR2 中的采样率发生器复位位。(GRST=0)。以复位采样率发生器。
- ③ 对各寄存器进行设置以使 SPI 协议有效。对 McBSP 的各寄存器进行配置, 以便使 SPI 的各设备进行有效的操作。
- ④ 使能采样率发生器。释放采样率发生器, 将 SPCR2 中的 GRST 置 1 (GRST=1)。
- ⑤ 使能接收器和发送器。采样率发生器被释放后, 等待两个时钟周期, 以保证适当的同步。由 CPU 来访问 McBSP, 置/XRST=/RRST=1, 使能收发, SPCR 寄存器中其他的设置值不变。
- ⑥ 等待两个时钟周期, 使内部能够正确地同步。
- ⑦ 如果要求内部帧同步信号使能采样率发生器的帧同步逻辑, 即使 FRST=1。

1.4 对串行 Flash 的读写操作

AT25F1024 是 Atmel 公司生产的一款应用广泛的高性能串行 Flash。存储容量为 1 Mbit (131, 072 * 8bit)。分为 4 个扇区, 每个扇区容量为 32kbit。支持扇区擦除和整片擦除^[6]。它的指令格式如表 1。

表 1 AT25F1024 指令格式

指令名称	指令格式	说 明
WREN	0000 x110	设置写使能
WRDI	0000 x100	清除写使能
RDSR	0000 x101	读状态寄存器
WRSR	0000 x001	写状态寄存器
READ	0000 x011	读数据
PROGRAM	0000 x010	写数据
SECTION ERASE	0101 x010	整段擦除
CHIP ERASE	0110 x010	全片擦除
RDID	0001 x101	读厂商和器件编号

当从 AT25F1024 读取一个字节时, 首先将 /CS 片选信号置低, 然后读指令 (READ) 和要读的地址经由 SI 管脚传入, 在指定地址的数据 (D7-D0) 经

由 SO 管脚传出。如果只读取一个字节, 当数据被送出后, /CS 片选信号将被拉高。当字节地址自动增加时, 读指令 (READ) 将继续, 数据应将继续被送出, 当到达最高地址 (00FFFF) 时, 读指令将停止。当将一个字节写入 AT25F1024 时, 要执行两条独立的命令。首先, 通过写使能指令 (WREN) 使 Flash 写使能。然后, 执行编程指令 (PROGRAM) 来对 Flash 进行写操作。在对 Flash 编程的过程中, 首先是 /CS 片选信号有效 (低电平)。然后编程指令 (PROGRAM), 地址和数据通过 SI 引脚传送进来。最后, 当 /CS 片选信号抬高 (高电平) 后, 芯片开始编程。/CS 片选信号由低到高的跳变要求必须在最后一个数据比特 D0 (LSB) 传送完成后, 紧跟着的 SCK 移位时钟为低时产生。编程指令只能对没有被块写保护指令保护的空間进行写操作。由于写命令只能将内部数据位由 1 写成 0, 反之则不行, 因此, 在写入数据之前一定要先对内部空间进行擦除操作 (SECTION ERASE OR CHIP ERASE)。读写时序如图 4 和图 5 所示。

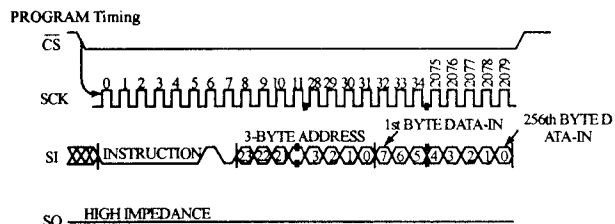


图 4 写时序

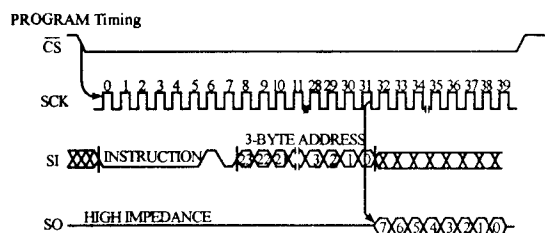


图 5 读时序

AT25F1024 支持单字节编程模式和 256 byte 的页编程模式。芯片内部每页 256 byte 的编程模式。芯片内部每页 256 byte, 一条编程指令可以在一页内烧写 1 到 256 个连续的字节, 烧写的起始地址可以在一页内的任意位置。连续烧写时地址自动增加, 但是当地址到达本页的最后时会自动跳到本页首地址。在内部写操作周期内, 其他命令都将被忽略, 但是读状态寄存器指令 (RDSR) 除外。

2 对 Flash 的串行引导

TMS320VC5509 芯片的 Bootloader 程序支持五种引导模式。本系统采用的是支持 24 位地址的

SPI 串行引导模式。VC5509 引导模式的选择是通过四个模式选择引脚 BOOTM[3:0]完成的。BOOTM0~3 引脚与芯片的通用输入输出引脚 GPIO1,2,3,0 相对应。选择的是 SPI 的引导模式,则 BOOTM[3:0]=0001。所谓引导表,就是在 DSP 芯片上电复位后由 Bootloader 从外部存储器装入片内 RAM 的一个数据块,这个数据块包括用户程序的数据段和代码段,还包括程序入口地址、寄存器设置、可编程延时等信息^[4,7]。

DSP 芯片在上电复位后会自动运行 ROM 中固化的 Bootloader 引导程序。引导程序会根据片外存储器中存放的引导表的信息,将用户程序的各个段引导到片内 RAM 的相应位置。然后从入口地址开始执行。这样就完成了整个装载引导的过程^[11]。

烧写 Flash 的步骤如下:

(1) 用 CCS 将最终程序编译生成 .out 格式文件。

(2) 将 .out 文件转化为 .hex 文件。

(3) 在生成 .hex 文件之前首先用文本编译器编写 .cmd 文件,文件格式如下:

```
-boot;说明创建 boot 文件
-v5510;2;生成 55xboot 文件格式
-serial 8;使用串行加载方式
-a;ASCII 格式
-reg_config 0x1c00, 0x02b7;在 0x1c00 寄存器写 0x02b7
```

```
-delay 0x100;延迟 0x100 个 CPU 时钟周期
```

```
-o flashtest.hex;输出 .hex 文件
```

```
flashtest.out;输入的 .out 文件
```

(4) 将生成的 .hex 文件转换为 .dat 文件。

(5) 将生成的数据文件烧写进 Flash。

3 用 CSL 库函数实现 McBSP 的配置

在 DSP 应用系统中,一般会涉及到大量对 DSP 器件外设特别是片上外设的编程处理工作,在开发初期消耗了大量的精力。在 TI 公司 CCS 开发环境中,提供了 DSP 片级支持库 CSL 作为一个自包容组件被归于 DSP/BIOS 中,多数 CSL 模块都由对应函数、宏、类和表示符号组合构成,在不调用其他 DSP/BIOS 组件的情况下也可以简单方便地完成对 DSP 器件片上外设配置和控制的编程工作,从而简化了 DSP 片上外设开发工作,缩短了开发周期^[9]。

对于 CSL 的使用有两种方法,一种是 CSL 的 GUI 调用,另一种是 CSL 库的直接调用。这里采用的是对 CSL 库的直接调用。具体的对 McBSP 的配置程序如下:

```
MCBSP_HANDLE hMcbbsp0; //句柄说明
MCBSP_Config myhMcbbsp={
    0x1800, //spcr1, CLKSTP=11
    0x0000, //spcr2, /FRST=/GRST=0
    0x0700, //rcr1, 接收帧长度为每帧 8 个字,接收字长为 8 位
    0x0005, //rcr2, 单相帧,无扩展,第一个帧同步脉冲之后,忽略帧同步信号,1 位数据延时
    0x0700, //xcr1, 发送帧长度为每帧 8 个字,发送字长为 8 位
    0x0005, //xcr2, 无扩展,数据转换开始于最高位 MSB
    0x0063, //srgr1, CLKMV=63, 100 分频
    0x2000, //srgr2
    0x0000, //mcr1
    0x0000, //mcr2
    0x0a0d, //pcr, 采样率发生器中的帧同步位 FSGM 决定帧同步信号。帧同步脉冲下降沿触发。发送数据在 CLKX 的上升沿采样。接受数据在 CLKR 的上升沿采样。
}; //与 SPI 模式 3 设置相关的一些相应的寄存器位配置[10]
Uint16 MCBSP_read8(MCBSP_Handle_MCBSP){
    Uint16 val;
    MCBSP _ ASSERT _ HANDLE ( _ MCBSP, return 0Xffffff);
    val=_PREG_GET(((MCBSP_PrivateObi*)_MCBSP)->Drr1Addr);
    return(val&0x00ff);
} //对 Flash 进行读操作的子程序
void MCBSP_write8(MCBSP_Handle_MCBSP, Uint16 val)
{
    MCBSP_ASSERT_HANDLE(_MCBSP, return);
    _PREG_SET((((MCBSP_PrivateObj*)_MCBSP)->Dxr1Addr),(val&0xff));
} //对 Flash 进行写操作的子程序
main() //对 Flash 的操作程序
{
    CSL_init();
    mcbbsp_init();
    clk_init();
    EBCR=(int *)0x6c00;
    *EBCR=0x01;
    *(int *)0x600001=0x1; //LED1 ON,擦除 Flash 时,灯 1 亮
    flash_erase(); //对 Flash 进行擦除操作
    for(i=0;i<60000000;i++);
    *(int *)0x600001=0x0; //LED1 OFF,擦除完毕后,灯 1 灭
    *(int *)0x600002=0x1; //LED2 ON,写 Flash 时,2 亮
    code_low=(int *)0x800b;
    code_high=(int *)0x800c;
    code_length=((long)code_high<<16)|code_low;
    flash_program(code_length); //对 Flash 进行写操作
```

```

for(i=0;i<6000000;i++);
*(int *)0x600002=0x0; //LED2 OFF,写完后,灯 2 灭
}
int mcbsp_init() //用 CSL 库函数完成对 McBSP 口的初始化操作
{
  Uint16 xmt;
  Uint16 rcv;
  old_int=IRQ_globalDisable();
  hMcbSP0=MCBSP_open(MCBSP_PORT0, MCBSP_OPEN_RESET); //打开串口 McBSP0
  rcvEventId=MCBSP_getRcvEventId(hMcbSP0); //设置接收中断
  xmtEventId=MCBSP_getXmtEventId(hMcbSP0); //设置发送中断
  IRQ_clear(rcvEventId); //清中断
  IRQ_clear(xmtEventId);
  MCBSP_config(hMcbSP0,&myhMcbSP); //调用配置函数,进行串口配置
  IRQ_enable(rcvEventId); //中断使能
  IRQ_enable(xmtEventId);
  IRQ_globalRestore(old_int);
  MCBSP_write8(hMcbSP0,xmt);
  MCBSP_start(hMcbSP0, MCBSP_XMIT_START|MCBSP_RCV_START|MCBSP_SRGR_START|MCBSP_SRGR_FRAMESYNC, 0x300u); //串口开始运行
  rcv=MCBSP_read8(hMcbSP0);
  MCBSP_ASSERT_HANDLE(hMcbSP0, return 11);
}

```

用一个简单的测试程序来检测一下对串口的设计是否实现。要完成的功能是使系统上电后,三个测试灯的自动循环亮灭。将编译生成后的 fleshtest.out 和 ledtest.out 文件加载到 DSP 的程序 RAM 后,执行

(上接第 919 页)

3 结束语

文章以 OMAP5912 为嵌入式系统平台,采用 SIP 在 oSIP 协议栈的基础上设计并实现了 IP 电话终端,经过测试,结果符合 RFC3261 规范。不过,随着 SIP 协议的更新,文章所设计的 IP 电话在软、硬件方面均需进一步完善。下一步将考虑采用无线的方式进行 IP 电话终端的设计,结合 3G 的 CDMA 无线分组网络技术,使该 IP 电话终端满足真正的车载汽车电子的应用。

参考文献:

[1] RFC 3261, SIP; Session Initiation Protocol[S], IETF, 2002.

FLASH 程序,对 DSP 硬件复位后或重启 CCS 后,可以发现三个测试灯的循环亮灭。证明对串口的设计以及对 Flash 的串行引导装载成功。

4 结束语

本文在实际应用的基础上,提出了一种基于 MCBSP 的 TMS320VC5509 与外部串行 Flash 的 SPI 协议的设计。采用串行 Flash 代替常规的并行 EPROM,大大节省了电路空间和功耗,降低了系统设计的复杂性,减少了成本,节省了空间,具有很好的实用性。

参考文献:

- [1] 周霖等. DSP 算法设计与系统方案[M]. 国防工业出版社 2004. 7.
- [2] 王文兴, 颜国正. 串行 DataFlash 存储器及其与单片机的接口[M]. 单片机与嵌入式系统应用.
- [3] 汪春梅, 孙洪波等. TMS320C5000 系列 DSP 系统设计与开发实例[M]. 电子工业出版社 北京 2004. 7.
- [4] 褚超, 杜栓义. TMS320VC5509 数传平台中串行引导的研究与实现[J]. 应用天地.
- [5] TMS320V5501/5502/5503/5507/5509/5510 DSP (McBSP) Reference Guide [S]. Texas Instruments, Incorporated SPRU592, PDF.
- [6] SPI Serial Memory AT25F1024 Datasheet Atmel[S].
- [7] Using the TMS320VC5503/VC5507/VC5509/VC5509A Bootloader[S]. Texas Instruments, SPRA375E.
- [8] TMS320C55XAPI[S] Texas Instruments.
- [9] 彭启宗, 管庆等 DSP 集成开发环境——CCS 及 DSP/BIOS 的原理与应用[M]. 北京电子工业出版社, 2004. 7.
- [10] 吴开杰, 邵蔚, 李刚. TMS320VC5410 的 McBSP 串行接口技术与程序设计[J]. 新器件新技术.
- [11] 徐胜, 管庆. TMS320C5000 的 Bootloader 技术. IC 与元器件.

[2] OMAP5912 TM HW Design Specification.

[3] 糜正琨编著. IP 网络电话技术[M]. 北京:人民邮电出版社, 2000. 6.

[4] Gonzalo Camarillo 著. SIP 揭密[M]. 北京:人民邮电出版社, 2003. 6.

[5] 桂海源编著. IP 电话技术与软交换[M]. 北京:北京邮电大学出版社, 2001. 3.

[6] Daniel Collins 著. VoIP 技术与应用[M]. 北京:人民邮电出版社, 2003. 6.

[7] Richard Petersen 著. Linux 编程命令详解[M]. 北京:电子工业出版社, 2004. 5.

[8] 蔡闵光编著. Linux 应用程序大全[M]. 北京:科学出版社, 2001. 3.