第 28 卷第 2 期　　佛山科学技术学院学报(自然科学版)　　Vol. 28 No. 2
2010 年 3 月　　Journal of Foshan University (Natural Science Edition)　　Mar. 2010

文章编号: 1008-0171 (2010) 02-0034-05

电能质量监测系统中的 SPI 串行通信

熊　柯[1,2], 陈立定[1], 杨兆华[2*]

(1. 华南理工大学 自动化科学与工程学院, 广东 广州 510640;
2. 佛山科学技术学院 自动化系, 广东 佛山 528000)

摘要: 以电能质量监测系统为研究背景, 提出了一种可广泛应用于现代测量仪器中的 ARM 加 DSP 的分布式控制结构, 着重分析了不同厂家的 SPI 接口在技术上的差异, 同时根据电能质量监测系统的功能需求给出了一个 DSP 与 ARM 进行 SPI 接口通信的解决方案, 并在此基础之上编写了实现基于 SPI 接口的 ARM 与 DSP 之间的串行通信。

关键词: 电能质量; SPI 通信; ARM; DSP
中图分类号: TM 73　　文献标志码: A

电能质量监测是发现和分析各种电能质量问题的主要手段, 为电能质量的改善和电力系统故障的诊断提供决策依据。电能质量监测系统作为一种典型的现代测量仪器, 具有数据采集量和处理量大、实时性要求高的特点, 且具有较强的网络通信能力和友好的人机交互能力。本文提出了一种 ARM + DSP 的分布式方案, ARM 处理器可以移植 Linux 操作系统, 易于实现网络通信、人机交互等功能; DSP 可以实现数据采集、传输的控制, 避免了因 ARM 处理器频繁 IO 操作造成的系统性能下降, 增强了系统实时性, 而 DSP 与 ARM 的通信又是整个系统的一个关键环节, 并着重介绍了三星 S3C2440 处理器和 TI 的 TMS320LF2407 的 SPI 接口串行通信。

1　SPI 接口的定义

SPI (Serial Peripheral Interface, 串行外设接口) 总线系统是一种同步串行外设接口, 它可以使 MCU 与各种外围设备以串行方式进行通信以交换信息, Motorola 首先在其 MC68HCXX 系列处理器上定义的。SPI 主要具有以下特点: 1) 占用接口线少, 利于系统的小型化, SPI 接口通常包含 4 条线, 即串行时钟线 (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 和低电平有效的从机选择线 SS; 2) 传输速率高, 可以达到几 Mbps。

由于 SPI 接口简单实用, 因此许多厂家的设备都支持该接口, 广泛应用于外设控制领域, 没有标准协议, 它现成为一种事实标准, 大部分厂家都是参照 Motorola 的 SPI 接口定义来设计的。因为没有确切的版本协议, 不同家产品的 SPI 接口在技术上存在一定的差别, 容易引起歧义, 所以本文所采用的 DSP 和 ARM 处理器的 SPI 接口存在很大的差异, 主要体现在接口寄存器和相关引脚功能的定义上。

图 1 是 SPI 接口的时序图。一般情况下 SPI 接口有 4 种时序[1], 可以配置相关控制寄存器选择时钟的相位以及极性, 综合考虑选择无延时的下降模式。同步时钟信号是由主机提供, 可以设置波特率以确定其频率, 在时钟的同步下, 数据由 MOSI 和 MISO 线在主机和从机之间传输。本文所采用的 DSP 和

收稿日期: 2009-11-11
作者简介: 熊　柯 (1984-), 男, 布依族, 贵州贵阳人, 华南理工大学与佛山科学技术学院联合培养硕士研究生。
* 通信作者: 杨兆华 (1964-), 男, 湖北荆门人, 佛山科学技术学院教授, 硕士生导师。

ARM 处理器的 SPI 接口在定义上差异较大, 其中 DSP 的 SPI 数据寄存器包括接收缓存寄存器 SPIRXBUF、发送缓存寄存器 SPITXBUF 和发送 接收缓冲寄存器 SPIDAT, 其在发送 接收数据的时候要经过 SPITXBUF SPIRXBUF、SPIDAT 两个数据寄存器, 并引起 SPI 接口状态寄存器相应位的置位以及中断产生。而ARM 处理器的SPI接口却没有SPIDAT 寄存器, 在接收到数据时也不会产生相应的中断并通知CPU 读取数据, 其 SPI 中断依赖于 SS 引脚上的低电平, 当DSP 和ARM 处理器同时设置为主机时, 如果ARM 的SS 引脚检测到低电平, 则会在 SPI 状态寄存器相应位上产生多主机错误指示, 同时产生 SPI 中断, 并自动设置自己为从机。另外, ARM 处理器的 SPI 接口数据寄存器是 8 位, DSP 的 SPI 数据寄存器是 16 位的, 而本文中的采集数据达到 12 位, 因此在DSP 与ARM 之间传输数据时, 还要进行一定的转换。

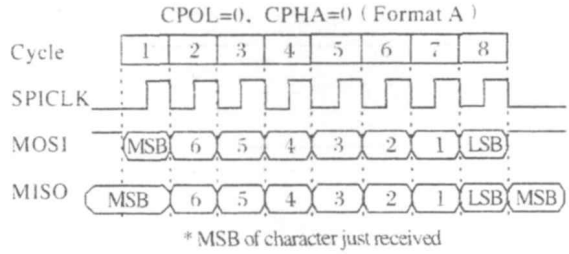

图 1　SPI 接口的时序

2　ARM 与DSP 的 SPI 串行通信解决方案

SPI 接口没有握手机制, 数据传输效率高, 速率也更快, 通常应用中可达几Mbps。此外 SPI 是全双工通信, 可同时发送和接收数据, 因此, SPI 比较适合用于数据传输的场合。比如需要较大批量数据传输的场合, 但这同时也是 SPI 接口的一个缺点: 没有指定的流控制、没有应答机制确认是否接收到数据。

基于上面的分析, 本文提出了一个较好的解决方案, 既能发挥 SPI 口的高传输效率, 又能避免由于没有应答机制造成的数据丢失。三星 S3C2440ARM 处理器的 SPI 接口支持查询、DMA 与中断 3 种传输模式。假设以DSP 为主机, 采集数据后以查询的方式发送给从机(ARM 处理器), 参考三星公司提供的 S3C2440 芯片资料, 如果将 SPI 设置为查询传输模式, 那么在每接收一个字节的数据前, ARM 处理器必然要先发送伪数据 0xFF 给DSP, 由于是双工通信, 一个字节的伪数据发送完毕时, 从DSP 接收来的一个字节的数据也传输完毕, 这时ARM 处理器的SPI 状态寄存器 SPSTA 0 的 Transfer Ready Flag 位也自动置位, 通过查询这一位的情况可以知道一个字节数据是否传输完毕。由于应答机制的缺失, 如果仅仅采用查询的方式, 为了避免数据的丢失, ARM 处理器必然不停地发送伪数据并查询SPI 状态寄存器, 这样就会造成ARM 处理器频繁 IO 操作而负担过重, 系统实时性随之降低, 也就失去了 ARM + DSP 双 CPU 分布式控制的意义。因此提出一种新的方案, 其硬件连接示意如图 2 所示。

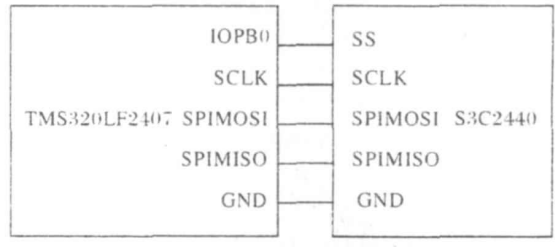

图 2　ARM 与DSP 的 SPI 接口连接

DSP 采集的数据并不直接传输给ARM 处理器, 而是达到一定的量之后再一次性传输, 这就起到了一定的缓冲, DSP 周期性地给ARM 发送一帧数据, 每帧数据的格式在下面将详细讨论。在DSP 与ARM 通信之前, 将ARM 处理器设置为中断工作模式, 把DSP 的一个 IO 口与ARM 的 SPI 接口 SS 引脚相连。DSP 在传输数据之前, 先向该口输出一个低电平, 这就产生了SPI 中断, 可以在中断处理程序开始设置ARM 处理器工作时查询状态, 在查询状态下完成一帧数据的传输, 然后重新设置ARM 处理器为中断状态, 中断返回。在没有数据传输的情况下, ARM 处理器可以完成数据处理、人机交互、网络通信等功能, 这样解决了DSP 与ARM 的通信中既要利用 SPI 没有应答机制、数据传输效率高的特性, 又要避免数据丢失的问题。

3　系统软件的设计

DSP 主要负责数据的实时采集, 并起到一定的数据缓冲功能, 移植了Linux 操作系统的ARM 主要负责人机交互和数据处理, DSP 采集到 5 个周期的数据再通过 SPI 接口传输数据到的ARM 处理器, 每次通信, DSP 将发送一帧数据, 具体的数据定义如表 1 所示。

表 1 每帧数据格式

第 1 周期		第 2、3、4 周期		第 5 周期		
时间	128 点采样数据	时间	128 点采样数据	时间	128 点采样数据	结束字符
8 个字	128 个字	同第 1 周期	同第 1 周期	8 个字	128 个字	1 个字

由于该系统实时监控的过程中需在采样每个周期的数据前都要读取 8 个字的时间, 采样按每周期 128 点进行, 每周期就包含 128 字的数据, 第 5 周期最后是 1 个结束字符, 占 1 个字空间, 这样就构成了一帧数据。

3.1 DSP 发送数据的程序设计

DSP 的 SPI 数据发送程序, 需要将 DSP 的 SPI 模块配置为主动方式使用 IOPB4 引脚作为 IO 口输出一个低电平[2], 使 ARM 处理器的 SPI 模块的 SS 引脚出现低电平, 引起中断。每次发送一帧数据, 其发送数据流程如图 3 所示。

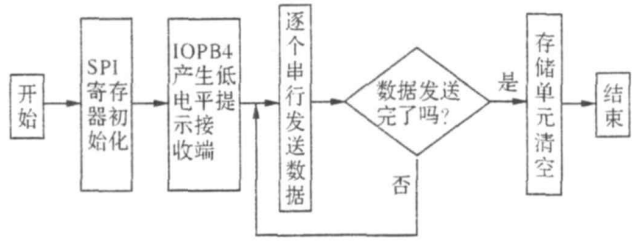

图 3 DSP 端 SPI 发送数据流程图

流程的关键代码如下:

```
Send ()
{ * PBDATDIR = * PBDATDIR  0x0010;     产生一个下降沿脉冲
                                  for(i= 0; i< 100; i+ + ) i= i;
 * PBDATDIR = * PBDATDIR &0xff00;
…
if(spiTxReady()= = 1)
  {     if( send  num= = 680)
    { * SPITXBUF = À ;      一帧的结束
    }
    else * SPITXBUF= data  int[send  num];
      send  num+ + ;
      if(send  num= = 681)
        {send  num= 0;
          }
      }
}
```

3.2 ARM 端的底层驱动程序

ARM 处理器的特点在于可以移植操作系统以完成人机交互、网络通信等功能, 在移植了 Linux 的 ARM 处理器平台上进行嵌入式开发可以分为 2 个部分, 一部分与底层硬件设备相关, 称为驱动程序; 另一部分是上层的应用程序。

在分析 ARM 的 SPI 接口驱动程序的结构的基础上, 给出其关键代码如下[3]:

```
static int spi  open(struct inode * inode, struct file * filp)
{…
    disable  irq(IRQ  SPI0);
    …
    spi  sppre0= 0xFF;    SPI 波特率设置
    spi  spcon0= (1< < 2)  (1< < 1)  (0< < 0);
      MultiMaster error detect enable, release
    …
```

```
    enable_irq(IRQ_SPI0);
    }
```

spi_open 函数主要是完成 SPI 接口的配置, 设置 SPI 接口 SPICLK 的相位和极性, 注意保持和 DSP 端的一致, 将 ARM 处理器设置为中断工作模式, 打开多主机检测, 此时 ARM 处理器也应设置为主机。

```
static ssize_t spi_read(struct file * filp, char __user * buf, size_t count, loff_t * f_ops)
{ copy_to_user(buf, dataRx, count);
  return 0;
}
```

Linux 操作系统的内存空间分为用户空间和内核空间, spi_read 函数中的 copy_to_user() 实现了将内核空间的数据拷贝到用户空间, 以供应用程序调用该数据。

如上文所述, 本文提出一种中断加查询的方案来解决 SPI 接口通信的问题, SPI 接口驱动的中断函数实现如下[4]:

```
static int __init spi_init(void)
{ …
  request_irq(IRQ_SPI0, spi_interrupt, SA_INTERRUPT, DEVICE_NAME, NULL);
  …
}
```

首先要在 spi_init 中调用 request_irq() 函数, 将中断号 IRQ_SPI0 与 spi_interrupt() 绑定, 一旦发生 SPI0 中断, 则系统会自动跳转到 spi_interrupt() 并以查询方式完成一帧数据的传输。

```
static void spi_interrupt(int irq, void * dev_id, struct pt_regs * reg)
{ …
  disable_irq(IRQ_SPI0);
  spi_spcon0= 0x15;
    设置为查询模式
    int i= 0;
    volatile int endSpiRx= 0;
    while(endSpiRx= = 0)
    {
      while(! (spi_spsta0&0x01));
      spi_sptdat0= 0xFF;
      while(! (spi_spsta0&0x01));
      dataRx[i]= spi_sprdat0;
      if((dataRx[i]) = = A )
        endSpiRx= 1;
    }
    spi_spcon0= (0≪6) (1≪5) (0≪4) (1≪3) (1≪2) (0≪1) (0≪0);
      一帧数据发送完毕恢复中断模式, 并设置ARM 处理器为主机
    enable_irq(IRQ_SPI0);
    kill_fasync(&async_queue, SIGUSR1, POLL_IN);    向应用程序发出异步通知信号
    }
```

进入 spi_interrupt 中断函数需要重新配置 SPI 接口以查询方式传输数据, 当数据传输完成, 恢复中断模式, 并设置 ARM 处理器为主机, 等待下一次数据传输, kill_fasync 发出异步通知信号使得上层应用程序调用 read() 系统调用将内核空间的数据复制到用户空间进行处理。

3.3 ARM 端的上层应用程序

上层的应用程序需实现人机交互、数据处理等功能, 本文采用 Trolltech 公司的 qt3 进行开发, 其中需要用到 open 系统调用打开 SPI 设备, 当驱动程序完成一帧数据传输并发出异步通知信号时, 就可以利用 read 系统调用读取数据进行处理了。

4 小结

本文以DSP 结合ARM 的分布式电能质量监测系统为研究背景, 提出一种中断加查询的新方案, 能较好地解决DSP 与ARM 的 SPI 接口通信中的问题, 由于现代测量仪器功能复杂, 多采用分布式结构。而 SPI 接口的通信方式具有效率高、占用资源少的特点, 因此该 SPI 接口通信方案对于设计分布式结构的测量仪器有较高的实用价值。

参考文献:

[1] 孙秋野, 孙凯, 冯健. ARM 嵌入式系统开发典型模块[M]. 北京: 人民邮电出版社, 2007.

[2] 刘和平, 王维俊, 江渝, 等. TMS320LF240xDSP[M]. 北京: 北京航空航天大学出版社, 2002.

[3] 张岩马, 旭东, 张云帆. ARM 与DSP 的 SPI 通信设计实现[J]. 工业控制计算机, 2009, 21(9): 56-57.

[4] 宋宝华. LINUX 设备驱动开发详解[M]. 北京: 人民邮电出版社, 2008.

【责任编辑: 任小平】

SPI communication in the monitor system of power quality

XIONG Ke[1,2], CHEN Li-ding[1], YANG Zhao-hua[2]

(1. College of Automation Science and Engineering, South China University of Technology, Guangzhou 510640, China;
2. Department of Automation Engineering, Foshan University, Foshan 528000, China)

Abstract: Based on the study of the monitor system of power quality, the distributed control structure which combines ARM and DSP is proposed to be used widely for the modern measuring instruments. This paper highlights the analysis of the technical difference of SPI interface made by different manufactory. With the consideration of the need of the monitor system of power quality, a solution to the SPI communication between ARM and DSP is proposed.

Key words: power quality; SPI communication; ARM; DSP