

32 位微处理器下的 SPI 技术的研究与实现

北京机械工业学院计算中心(100085) 彭双和 杨根兴

摘要: SPI 是一种高效的串行双向同步通信接口,适合于主机与外围设备进行通信,但有些 MCU 不带 SPI。为实现这一类 MCU 与带 SPI 的外围设备进行通信,本文介绍了一种软硬件结合的技术,并以伪 SPI 命名,模拟 SPI 的工作。

关键词: 串行外围接口 全双工同步通信 主模式 从模式

1 SPI 技术概述^{[1][2]}

同步串行外围接口(Serial Peripheral Interface, SPI)用于与各种外围器件进行通信。这些外围器件可以是简单的 TTL 移位寄存器、复杂的 LCD 显示驱动器或 A/D 转换器系统。SPI 系统可以容易地与许多厂家的各种标准外围器件直接连接。在多主机系统中 SPI 还可用于 MCU 之间的通信。

当 MCU 片内 I/O 功能或存储器不能满足需要时,可用 SPI 与各种外围器件相连,扩展 I/O 功能。这也是扩展 I/O 功能最方便、最简单的方法,只需 3~4 根线就可实现 I/O 功能扩展。

8 位 SPI 的基本结构如图 1 所示。

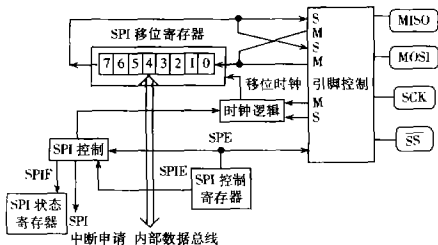


图 1 SPI 基本结构

其中与 SPI 数据传输有关的 4 个 I/O 引脚是:主机输入/从机输出端 MISO (Master In Slave Out)、主机输出/从机输入端 MOSI (Master Out Slave In)、串行时钟端 SCK 和从机选择端 SS。

SCK 引脚是主机模式的时钟输出和从机模式的时钟输入。把数据写入主 CPU 的 SPI 数据寄存器会启动 SPI 的时钟发生器,数据将从 MOSI 引脚移出和移入。在 1 个字节移出后, SPI 时钟发生器停止,并设置传送停止标志位 SPIE (SPI Flag)。如果 SPCR 寄存器 (SPI Control Register) 中的中断触发位 SPIE (SPI Enable) 被设置 (即允许 SPI 中断请求), 则生成一个中断请求。从机选择输入 (SS) 被设置为低, 用来选择单独的 SPI 器件作为从机。主机和从机的 2 个移位寄存器可以被认为是一个分开的 16 位环形移位寄存器, 如图 2 所示。当数据从主机移向从机, 同时数据也将移向相反的方向。这意味着在 1 个移位周期内, 主机和从机的数据交换。

主 从

图 2 主/从 GPU 之间的 SPI 内部连接

SPI 可用于 MCU 与外围器件 (包括其它 MCU) 进行全双工、同步串行通信。SPI 可以同时发出和接收串行数据。当 SPI 工作时, 在 8 位移位寄存器中的数据逐位从输出引脚输出 (高位在前), 同时从输入引脚接收的数据逐位移到 8 位移位寄存器 (高位在前)。发出 1 个字节后, 从另一个外围器件接收的字节数据进入 8 位移位寄存器。主 SPI 的时钟信号使传输过程同步。

在开发“多频漫游 PDA”项目的过程中, MCU 选用的是 Seiko Epson 公司的 32 位微处理器 E0C33208。在设计的过程中发现, 尽管 E0C33208 具有很强的 I/O 功能, 但因其所要实现的功能较多, E0C33208 片内 I/O 不能满足要求。因此, 为了能使其与具有 SPI 接口的器件通信正常进行, 本文介绍了采用一种软硬件结合的技术, 以伪 SPI 命名, 模拟 SPI 工作。

2 伪 SPI 技术的软件实现^[3]

SPI 的特征如下:

(1) 全双工, 3 线同步数据传送。

硬件纵横

- (2)主从操作。
- (3)LSB 或 MSB 在先。
- (4)可编程的位速率。
- (5)传送停止的中断标志。
- (6)从闲置模式下唤醒(仅在从模式下)

下面以 32 位微处理器 EOC33208 为 MCU, 介绍采用软硬件相结合的方式实现 SPI 功能。

EOC33208 与具有 SPI 接口的外围器件的

硬件连接如图 3 所示。利用 EOC33208 的输入口(K 口)K5.1、K5.2 和 I/O 口(P 口)P2.3、P2.6、P2.7 共 5 根线作为 SPI 的传输信号线。其中 EOC33208 的 P2.6 和 K5.1 分别与外围器件的 \overline{SS} 和 \overline{READY} 信号

图 3 具有 SPI 接口的外围器件端与 EOC33208 之间的信号连接

相连。作为二者通信的握手信号;EOC33208 的 K5.2 和 P2.3 分别与外围器件的 MOSI 和 MISO 相连,作为数据传输信号线;外围器件与 EOC33208 之间进行同步信息传输的时钟由 EOC33208 提供。这里使用的是 EOC33208 的 16 位定时器 5(P2.7 口的第 2 功能)。

如图 4 所示,由外围器件端发起的 SPI 数据包传输步骤如下:

- (1)外围器件驱动 \overline{READY} 信号变低,表示数据传输已准备好。
- (2)如果当前 MCU 仍未选通外围器件作为通信对象,则 MCU 通过驱动 \overline{SS} 信号为低电平选通外围器件。
- (3)MCU 接收或发送 32 位 SPI 数据包。
- (4)MCU 驱动 \overline{SS} 信号变高,使外围器件处于未选通状态,MCU 与外围器件之间的通信结束。

当 MCU 准备向外围器件端发送数据的时候,时序图如图 5 所示,其初始化步骤如下。

- (1)MCU 驱动 \overline{SS} 信号为低,选通外围器件作为当前的通信对象。
- (2)等待外围器件驱动 \overline{READY} 引脚为低,表示外围器件已准备好通信。

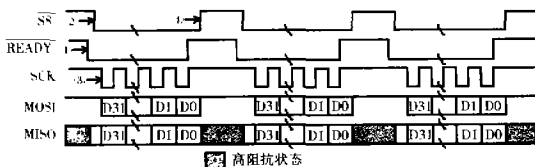


图 4 由外围器件发起的多 SPI 数据包传输的典型时序图

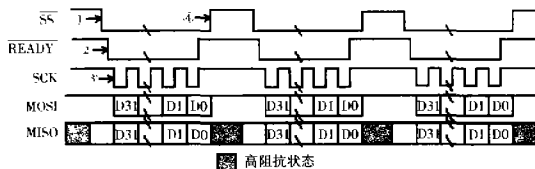


图 5 由主机发起的多 SPI 数据包传输典型时序图

- (3)MCU 发送 32 位 SPI 数据包。

(4)MCU 驱动 \overline{SS} 信号为高,使外围器件处于不工作状态,MCU 与外围器件之间的通信结束。

(5)若要继续发送 SPI 数据包,则重复执行步骤(1)~(4)。外围器件端与 EOC33208 之间的伪 SPI 实现的原理框图如图 6 所示。图中伪 SPI 中断申请以外部的键盘中断模拟。当外围器件需要进行 SPI 传输时,可按下按钮,申请 SPI 中断。

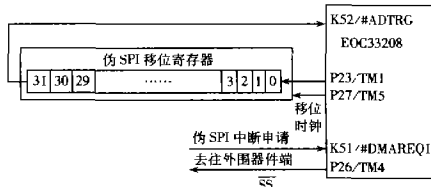


图 6 32 位微处理器下伪 SPI 原理框图

在伪 SPI 中断(即 K51 口中中断)中需要完成:

- (1)将 32 位 SPI 数据包包到伪 SPI 移位寄存器中。
- (2)启动 16 位定时器 5 工作。

K51 口的中断处理流程如图 7 所示。

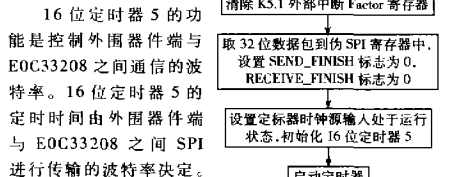


图 7 K51 口的中断处理流程

根据不同的传输波特率,设置不同的定时参数初始值,并提供伪 SPI 移位寄存器的移位时钟信号。当 32 位伪 SPI 数据包传输完成后,若外围器件端另有 SPI 数据包要传输,则重新向 EOC33208 提出申请,进行新一轮的伪 SPI 传输。其处理流程如图 8 所示。

硬件纵横

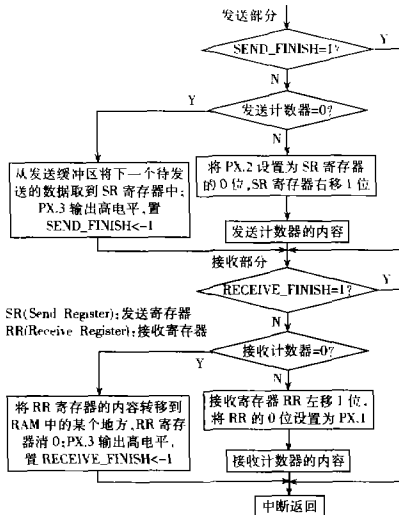


图8 16 位定时器5的中断处理流程

(上接第15页)

(4)编写用户代码

定义完用户接口和Home接口后,可以编写用户代码。

```

package com.wiley.compBook.roman.session.helloworld;
import javax.ejb.*;
import javax.naming.*;
import java.rmi.*;
import java.util.Properties;
/*
 * This class is an example of client code that invokes
 * methods on a simple stateless session bean.
 */
public class HelloClient {
    public static void main(String[] args) {
        try {
            // Get System properties for JNDI initialization
            Properties props=System.getProperties();
            //
            * Form an initial context
            Context ctx=new InitialContext(props);
            //
            * Get a reference to the home object
            * (the factory for EJB objects)
            //

```

《微型机与应用》2002年第2期

3 伪SPI的优缺点

采用伪SPI的优点是:可以灵活利用MCU的现有资源,扩充MCU对外的功能,使之具备与具有SPI接口的外围器件通信的功能。其缺点是:占用MCU的定时器和I/O口资源,并且占用一定的CPU时间,使传输速率不高。

4 结束语

在将源数据传输到目的地的过程中,除用一般的数据传输指令外,还可利用EOC33208的HSDMA(High Speed DMA)中断功能,进行快速、有效的数据传输。

参考文献

- 1 SEIKO EPSON CORPORATION.EOC33208/204/202 TECHNICAL MANUAL.1999
 - 2 邹宽明.单片机外围器件实用手册数据传输接口器件分册(第1版).北京:北京航空航天大学出版社,1998
 - 3 耿德根,宋建国.AVR高速嵌入式单片机原理与应用(第1版).北京:北京航空航天大学出版社,2001
- (收稿日期:2001-09-21)

用户程序执行了以下任务:

- (1)检查一个Home Object;
- (2)使用Home Object创建一个EJB Object;
- (3)在EJB Object上调用Hello();
- (4)删除EJB Object。

3 结论

EJB技术代表的是一种前沿技术,它使得复杂的多层结构应用系统开发变得容易。EJB技术建立在已取得的规范之上,采用EJB技术能有效地进行软件重用、降低软件的开发和维护成本。EJB技术使Java在企业计算中的地位得到加强,为基于Java的应用系统提供了一个框架。和目前系统和模型相比,EJB具有许多优越性,它有可能成为多层结构分布式应用系统的服务器端构件模型的首要选择。

参考文献

- 1 John W.Mastering Enterprise JavaBeans and the Java 2Platform,Enterprise Edition.Wiley Computer Publishing,1999
- 2 仲莘豪.中间件构筑复杂分布式应用的关键技术.http://www.apusic.com/tech/articles/middleware/middleware.html
- 3 王少锋,王宏宏.Java EJB技术研究.http://teamsun.com.cn/usseer/ejb.htm
- 4 全面研读EJB 2.0.http://www.umlchina.com/CBD/Newindex1.htm
- 5 宋解.江峰.Java服务器程序设计.北京:清华大学出版社,2000

(收稿日期:2001-10-08)