

Communication Between the TM1300 Synchronous Serial Interface and Modem Analog Front-End

TM1300 同步串行接口与 Modem 模拟前端之间的通信



1 TM1300 同步串行接口 (SSI)

TM1300 同步串行接口单元与 Modem 模拟前端 (MAFE) 子系统、网络终端、ADC/DAC 或多媒体数字信号编解码器可以进行灵活的位串连接, 硬件对位流执行全双工的串行化和解串化操作, 任何与其连接的前端设备都必须能够传送和接收数据并且能够由一个同步串行接口进行初始化。

因为通信算法是由软件来实现 (通过 DSPCPU 执行) 而且模拟的前端设备是脱离芯片的, 所以支持各种各样的 Modem 和网络协议。

1.1 接口

外部接口包括下面的六个引脚(见表 1):

1.2 功能模块图

SSI 单元的主要模块图 (见图 1)。

1.2.1 I/O 控制模块

I/O 控制模块是用来控制 I/O 引脚和选择传送时钟和传送帧同步信号的, SSI-IO1 和 SSI-IO2 外部引脚通过正确的设置 SSI-CTL 控制寄存器的值可以用作通用目的的 I/O, 也可以作为传送时钟信号输入和传送帧同步信号的输入和输出。SSI-CTL.IO1 和 SSI-CTL.IO2 的模式选择域可以控制这两个引脚的方向和功能。

摘要: 本文介绍了 Philip 公司的 Trimedia 系列多媒体 DSP 芯片 TM1300 的同步串行接口 (synchronous serial interface) 与 Modem 模拟前端 STLC7545 的结构、功能及特性, 分析了两者的通信过程, 并且介绍了通过 API 函数来实现两者之间通信。

关键词: TM1300 同步串行接口 Modem 模拟前端 通信 SSI STLC7545 API

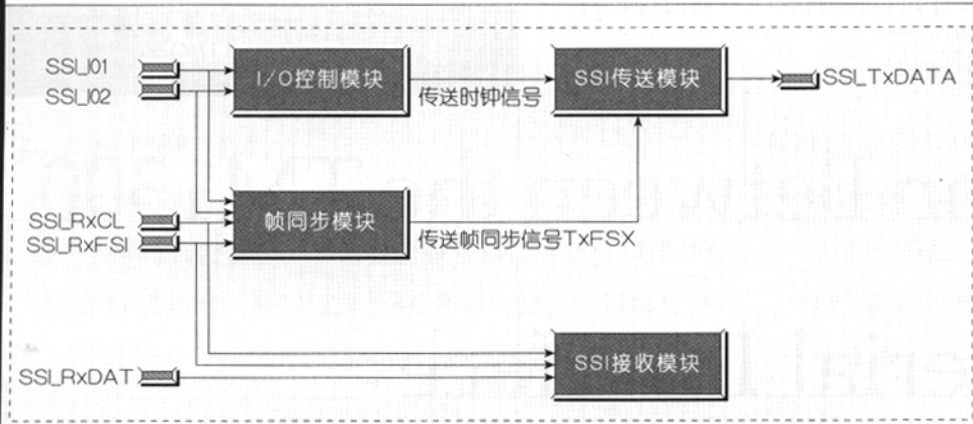


图 1 SSI 功能模块图

特别值得注意的是: 如果 SSI-I01 没有被选为传送位时钟输入, 那么传送时钟信号将从接收时钟信号取得。如果 SSI-I02 没有被选为传送帧同步信号输入和输出, 那么传送帧同步信号将从接收帧同步信号中取得。

1.2.2 帧同步模块

帧同步产生逻辑在图 2 里说明。一个传送帧同步信号 (TxFSX) 有三个来源, 选择哪一个由 SSI-CTL.I02 [0:1] 的取值来决定:

- (1) 当其值为 00 和 01 时, 从接收帧同步信号 (RxFSX) 取得。
- (2) 值为 10 时, 从内部帧同步信号 (Internal TxFSX) 中取得。
- (3) 值为 11 时, 从 SSI-I02 引脚取得。

1.2.3 SSI 传送模块

传送器的时钟有两个来源可选择, 从 SSI-I01 或 SSI-RxCLK 取得, 可通过对 SSI-CTL 寄存器中的 I01 [0:1] 位编程实现。一个传送可以发生在时钟的上升沿和下降沿, 这可以通过设置

SSI-CTL.TCP 的值来实现。

传送器有一个 30x16 位先进先出缓冲区 (TxFIFO), 它在 32 位传送数据寄存器 (SSI-TxDR) 和 16 位的传送移位寄存器 (TxSR) 之间提供数据缓冲。

TxSR 寄存器是一个 16 位的传送移位寄存器。

SSI-TxDR 是一个 32 位的 MMIO 传送寄存器。

1.2.4 SSI 接收模块

接收时钟信号, 接收帧同步信号和数据信号通常是由外部引脚引入得到。

接收器有一个 32x16 位的先进先出缓冲区 (RxFIFO), 它在 16 位的接收移位寄存器 (RxSR) 和 32 位接受数据寄存器 (SSI-TxDR) 之间提供数据缓冲。

输入引脚 SSI-RxDATA 提供串行移位的数到 RxSR, RxSR 是 16 位的串行移位寄存器。一个传送发生在接收时钟的上升边沿或下降边

沿, 这是通过设置 SSI-CTL.RCP 位来控制。

2 Modem 模拟前端 STLC7545

Modem 模拟前端芯片 STLC7545 的设计可以实现位率达到 38400bps 的高速率调制解调器。当与一个或多个数字信号处理器 (DSP) 连接时, 它可以实现符合 CCITT (v.17, v.21, v.22 V.22 bis, V.23, V.26, V.27, V.29, V.32, V.32 bis, V.33, v34 和 v34bis) 和贝尔 (103,202,212A) 推荐的多工作模式的 Modem。它的传送部分包括带有插值滤波器的 16 位重复取样数模转换器, 接收部分包括带有两个可编程滤波器 (一个用来抽选, 一个用来重构) 的 16 位重复取样模数转换器。重复取样率是可以在 128, 160, 192 赫兹间选择。提供两个独立的时钟产生系统, 一个用于 Tx 上速率的同步, 另一个用于 Rx 上速率的同步。在外部时钟模式下, 外部的重复采样时钟可以提供给这个芯片。有两个独立的同步串行接口 (SSI) 允许与几个标准 DSPs 进行通信。

2.1 STLC7545 几个重要引脚的功能描述

(见表 2)

2.2 STLC7545 的简要模块图 (见图 3)

注意: STLC7545 的内部结构非常复杂, 功能十分强大, 但是在与 SSI 通信的过程中很多引脚和功能实际上没有被用到, 因此文中的模块图实际上只是经过简化了的模块图。

3 TM1300 与 Modem 模拟前端 STLC7545 的连接

在我们的系统中, 同步串行接口实际上只有 4 个引脚和 Modem 接口芯片 STLC7545 相连接: SSI-RxCLK-BCLKX, SSI-RxFSX--FSX, SSI-RxDATA--TXDO, SSI-TxDATA--TXDI, 而 SSI 中的 SSI-I01 和 SSI-I02 两个引脚没有用到, 这主要是因为 TM1300 同步串行接口所使用的传送位

表 1 同步串行接口引脚

名字	类型	描述
SSI-RxCLK	IN-5	串行接口时钟信号, 由外部通信设备提供。
SSI-RxFSX	IN-5	帧同步参考信号, 由外部通信设备提供。
SSI-RxDATA	IN-5	接收串行数据信号。
SSI-TxDATA	OUT	传送串行数据输出。
SSI-I01	I/O-5	这个引脚可作为传送时钟输入或通用目的的 I/O 引脚。
SSI-I02	I/O-5	这个引脚可作为传送帧同步信号输入或输出, 或作为通用的 I/O 引脚。

时钟脉冲和传送帧同步时钟脉冲,都是直接使用的接收位时钟脉冲(从 SSI-RxCLK 得来)和接收帧同步时钟脉冲(从 SSI-RxFSX 得来)。

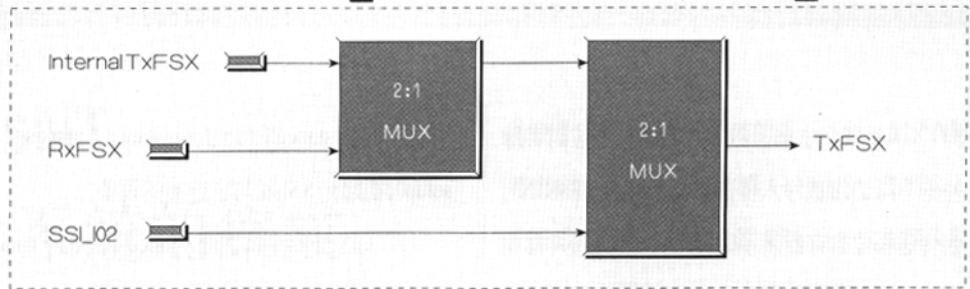


图 2 帧同步信号产生模块图

4 TM1300 与 Modem 模拟前端芯片 STLC7545 之间的通信过程

4.1 同步串行接口接受数据过程

首先, STLC7545 芯片从外部的电话线接收到发送数据终端发送来的数据(模拟信号), 然后进行模数转换, 并从其中提取传送位时钟信号(BCLKX)和传送帧同步信号(FSX)。经模数转换后的数据是 16 位数据字形式, 16 位数据字被装载入它的传送缓冲寄存器(TBROUT)。然后, 传送缓冲寄存器在 BCLKX 脉冲的上升边沿将其移入其传送移位寄存器(TSROUT)中。传送移位寄存器中的数据再通过 TxDO 引脚输入到串行接口的 SSI-RxDATA 引脚, 同时, 传送位时钟脉冲由 BCLKX 引脚传入 SSI-CLK 引脚, 帧同步脉冲由 FSX 引入 SSI-RxFSX 引脚, 此时, 在 TM1300 同步串行接口中, 接收器被激活(注意: 接收器和发送器必须被同时激活)接收状态机开始在第一个有效的 SSI-RxCLK 信号边沿(上升或下降边沿, 由 SSI-CTL.RCP 位控制)将 SSI-RxDATA 中的数据串行移入 RxSR (接收移位寄存器)。当 RxSR 满了时, 16 位的数据将并行的传入第一个有效的 RxFIFO 缓冲区, 然后再传入可用的 SSI-RxDR (接收数据寄存器)。接收过程重复进行下去, 当 RxSR 再一次充满了的时候, 16 位数据并行的传入下一个有效的 RxFIFO 缓冲区, 如果

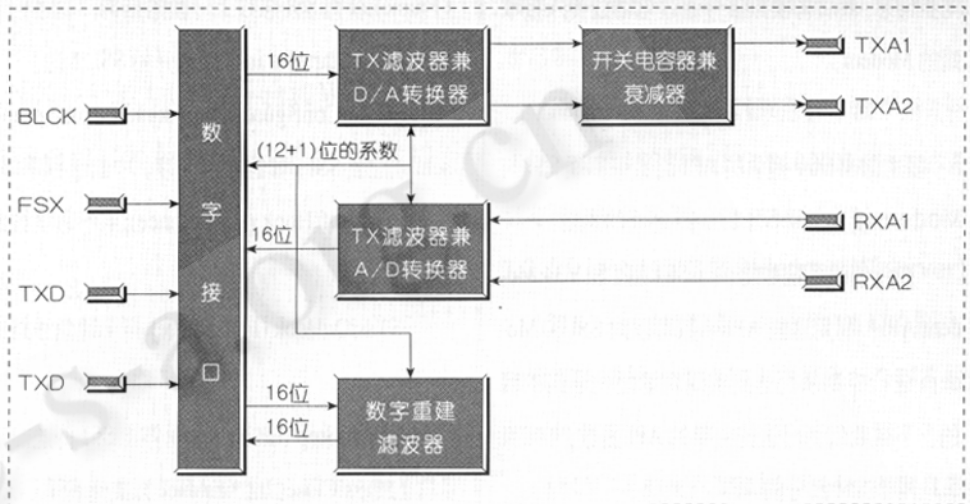


图 3 STLC7545 的简要模块图

SSI-RxDR 寄存器此时可用, 将数据再并行的传入 SSI-RxDR 寄存器, 整个接收过程会一直进行下去直到接收状态机无效或复位。如果接收状态机必须将 RxSR 的数据并行传入一个 RxFIFO 缓冲区, 而此时又没有有效的 RxFIFO 缓冲区, 那么这个再 RxSR 中的数据将会丢失, 接收状态机将会给出接收溢出状态标志并产生一个中断。

4.2 同步串行接口的传送数据过程

传送器被激活(传送器和接收器必须被同时激活), 整个传送过程的传送位时钟信号和传送帧同步信号都使用从 SSI-RxCLK 和 SSI-RxFSX 引脚接受到的来自 STLC7545 的传送位时钟脉冲

(BCLKX)和传送帧同步脉冲(FSX)来作为同步信号。传送状态机等待要传送的数据写入 SSI-TxDR 寄存器(传送数据寄存器)。只要 SSI-TxDR (32位)被写入数据, 就通过传送位时钟脉冲和传送帧同步脉冲同步, 将它推入两个有效的 16 位 TxFIFO 缓冲区, 然后串行的移入 TxSR 寄存器(传送移位寄存器)。接着数据开始从 TxSR 寄存器中移出, 在每一个有效的传送位时钟边沿(上升或下降边沿, 可由 SSI-CTL.TCP 位控制)传送一位, 直到 TxSR 为空。当 TxSR 为空时, 传送状态机将从下一个有效的 TxFIFO 缓冲区装载数据并开始串行移出那个数据。这个传送过程一直继续直到传送状态机无效或复位。如果最后一个有效的 TxFIFO 缓冲区的数据在要装载入 TxSR 寄存器时没有被及时的更新, 那么最后一个传送的帧将被重发。传送状态机将会置 SSI-CTL.TUE 位, 表示一个传送溢出错误并产生一个中断。接下来, 我们看看 STLC7545 这部分的情况: 从引

表 2 STLC7545 几个重要引脚的功能描述 (STLC7545 有 30 多个引脚)

引脚号	名字	功能描述	引脚号	名字	功能描述
2	TxDO	传送串行数据输出	29	RXA1	模拟信号的正极输入
1	TxDI	传送串行数据输入	30	RXA2	模拟信号的负极输入
3	FSX	传送帧同步输出	34	TXA1	平滑过滤器正极输出
4	BCLKX	传送位时钟输出	35	TXA2	平滑过滤器负极输出

脚TXDI接收到的串行数字数据在传送位时钟脉冲的下降边沿被移入传送移位寄存器 (TSRIN)。当传送移位寄存器满了时,则传入传送缓冲器中 (TBRIN)。再将数字数据信号转换为模拟信号,且要在模拟信号中加载传送位时钟信息和传送帧同步信息。最后将模拟信号通过电话线传入接收端的 Modem。

5 使用 TM1300 提供的 API 函数来实现 SSI 与 Modem 模拟前端 STLC7545 之间的通信

在我们的系统中提供了 SSI API 和 V34 Modem API, 利用这些 API 函数可以对 SSI 和 Modem 进行控制从而达到实现两者之间通信的目的。下面我们列出这些主要的 API 函数。

5.1 用于 SSI 部分的 API

(1) ssiGetCapabilities(pssiCapabilities-t *pcap);在安装 SSI 设备库之前系统资源控制器通过此函数来找到 SSI 设备库。

(2) ssiOpen(Int *instance); 打开 SSI。

(3) ssiInstanceSetup(Int instance, pssisetup-t setup);初始化 SSI 和与它连接的设备。

(4) ssiSetFraming(Int instance, pssiSetFrame-t Frame);控制 SSI 总线上一帧的结构。

(5) ssiGetFraming(Int instance, pssiSetFrame-t Frame);获得 SSI 总线上一帧的结构。

(6) ssiStart(Int instance);开始 SSI 传送。

(7) ssiConfigure(Int instance, pssisetup-t setup);配置 SSI, 如缓冲区长度, 中断级别等等。

(8) ssiOffHook(Int instance);用于通信线路的控制。

(9) ssiOnHook(Int instance);用于通信线路的控制。

(10) ssiStop(Int instance);停止 SSI 传送。

(11) ssiClose(Int *instance); 关闭 SSI。

5.2 用于 Modem 部分的 API

(1) tmModemAfeInit();初始化支持平台。

(2) tmModemGetCapabilities(&cap);获得 Modem 的属性的数据结构。

(3) tmModemOpen(&instance);打开一个 Modem 实例。

(4) tmModemInstanceSetup(instance, &mdmSetup);初始化 Modem 接口。

(5) tmModemStart(instance);Modem 开始工作。

(6) tmModemStop(instance);Modem 停止工作。

(7) tmModemClose(instance);释放 Modem 的实例以及为其内部变量所分配的内存空间。

6 结束语

如前文所述, TM1300 的同步串行接口在与 Modem 模拟前端 STLC7545 通信时, 所有的 Modem 信号处理过程都是由 TM1300 的 DSPCPU 来完成。也就是说整个通信过程的控制都通过软件来实现。

这就使得 SSI 可以灵活的接入符合各种通信协议的网络通信设备。但 SSI 的这种灵活性对 TM1300 的开发人员提出了更高的要求, 不仅要了解 SSI 和 Modem 接口芯片的地层的硬件工作原理十分熟悉, 还要对各种通信协议有深入的了解。 ■

参考文献

- 1 TM1300 Preliminary Data Book. Philips Semiconductors 1999.
- 2 Software Development Environment Disc Version 2.1.
- 3 苏涛, 吴顺君, 廖晓群编著. 高性能数字信号处理器与高速实时信号处理. 西安电子科技大学出版社.
- 4 STLC7545 ENHANCED V.34 BIS ANALOG FRONT-END PRELIMINARY DATASGS-THOMSON MicroElectronics.
- 5 <http://www.mds.com>