

引言

本应用笔记介绍怎样使用 Altera® MAX® II CPLD 来实现协议转换, 通过串行外设接口 (SPI) 控制 inter-IC 声音 (I²S) 总线上的音频设备数据流。

I²S 和 SPI 接口

I²S 是 3 线半双工串行接口, 常用于通过被称为 I²S 总线的 3 线总线来连接系统中的音频设备。I²S 设备和总线使用 3 条线: 串行数据 (SDA), 承载了对应于每一个音频通道的两路时分复用音频数据; 串行时钟 (SCK) 和字选择 (WS), 控制数字音频数据在 I²S 总线上不同设备之间的传输。I²S 系统能够处理和串行时钟分开的串行音频数据, 以消除抖动。

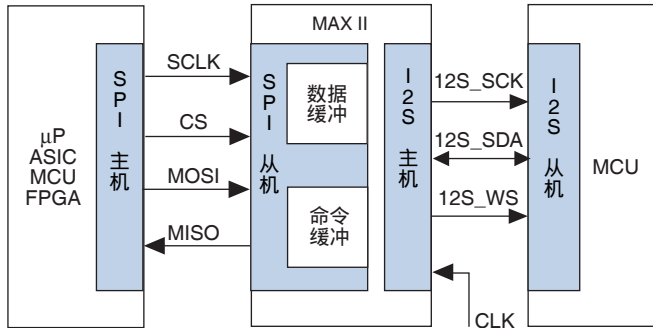
SPI 是 4 线全双工串行接口, 常用于连接系统内部和外部的系统处理器 (主机) 和外设 (从机)。SPI 利用单独的数据输出信号线 (主机输出从机输入, 即 MOSI) 和数据输入信号线 (主机输入从机输出, 即 MISO) 以及时钟 (SCLK) 和从机选择或者片选信号 (CS) 来进行通信。

本设计支持 SPI 接口主机对其他设备的数据流控制, 包括 A/D 转换器、数字信号处理器、数字滤波器、音频处理器、PC 多媒体音频转换器等 I²S 总线设备。有的嵌入式系统并没有为音频数据通信提供 I²S 接口, 但是有 SPI 接口。您可以在和这些接口相似的环境中有效地使用本设计。

利用 MAX II CPLD 实现 SPI 至 I²S 的接口

SPI 主机的桥接接口是 SPI 从机, 它有四条信号线 (CS, SCLK, MISO 和 MOSI)。I²S 总线一侧接口是 I²S 主机, 它有三条信号线 (I²S_ SCK, I²S_ SDA 和 I²S_ WS)。CLK 是主机时钟, 位于 I²S 主机外部。图 1 所示为采用 MAX II CPLD 来实现 SPI 至 I²S 接口。

图 1. 利用 MAX II CPLD 实现 SPI 至 I2S 接口

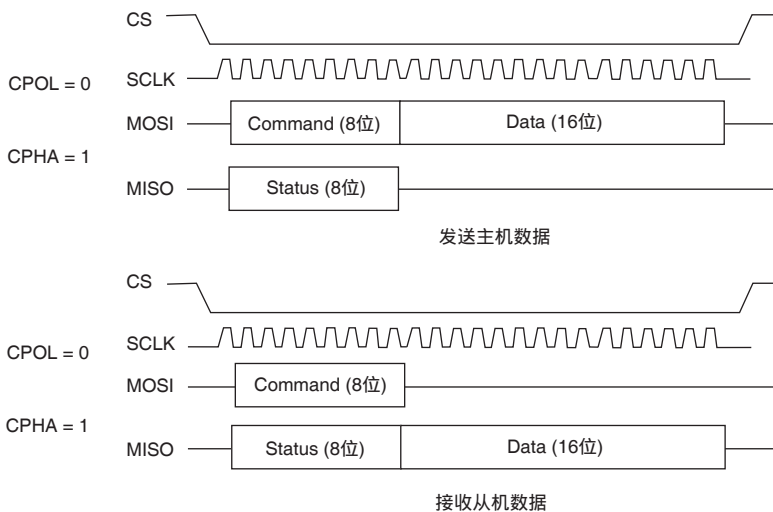


本设计提供协议转换功能，通过 SPI 接口（更常用的 4 线串行接口，用于连接串行外设接口和微处理器或者主机）来控制 I²S 总线（带宽相对较窄的协议，用于连接系统中的数字音频设备上音频设备的数据流。本设计使用了 MAX II CPLD 来桥接 SPI 主机，和 I²S 总线上的设备进行通信。在图 1 中，对于 SPI 主机（主处理器），MAX II CPLD 是 SPI 从机，反之，对于 I²S 总线，则起到了 I²S 主机的功能来连接主系统。

在传输过程中，SPI 数据的前 8 位是命令字节，然后是 16 位数据。只有当需要把数据传送到 I²S 上时，数据位才有效。在接收过程中，首先是前 8 位，即状态字节，然后是一个 16 位数据字。数据字是否有效由状态字节中的数据有效位进行规定。

图 2 为 SPI 时序图。

图 2. SPI 时序图



SPI 接口

正常情况下，SPI 总线只有一个主机以及和它相连的多个从机。CPLD 作为从机之一和 SPI 主机设备相连，实现桥接功能。表 1 列出了图 1 中的 SPI 从机线。

引脚名称	方向	说明
CS	输入 (低电平有效)	芯片 / 从机选择
SCLK	输入	串行时钟
MISO	输出	主机输入从机输出
MOSI	输入	主机输出从机输入

如图 2 所示，SPI 主机发送前 8 位命令字节，随后是 16 位数据。这一数据只有在传送时才有效。在接收数据时，或者 PS 主机用作控制器时，这些 16 位数据无效。

图 3 所示为命令字节格式。下一节对每一命令字节进行了说明。

图 3. 命令字节格式

Check	RST	-----	Controller	WS	RC	Clock	TR
-------	-----	-------	------------	----	----	-------	----

Check

这是首先发送的比特，说明了 SPI 主机发送的字的的有效性。如果该比特是 0，其后的所有比特均无效。如果该比特是 1，其后的比特均有效。从机读取状态时才用到该比特。

RST

这是 I²S 主机的复位信号。如果该比特是 1，I²S 主机被复位。如果该命令位指示 I²S 主机进行复位，主机在一个时钟周期中复位，在另一个时钟周期中准备执行新的命令。因此，总共需要两个时钟周期来完成复位过程。



每次需要改变主机功能时，发送一个 RST 命令来复位 I²S 主机。完成某一功能后，最后一个命令再次复位 I²S 主机。注意，无论什么时候发送复位命令，命令字节的其他比特都应含有前一命令，同时 RST 被设置为 1。之所以需要这样，是因为在完成整个数据处理周期 (16 个时钟周期中的第 15 个) 前，从命令缓冲中读取命令。

Controller

当 I²S 主机用作控制器时，该比特为高电平。



当 I²S 主机起到控制器的作用时，RC 和控制器位都应该是高电平。当 I²S 主机起到接收器的作用时，只有 RC 位是高电平。

WS

该比特说明了接收或者发送的每一数据字的字通道。

RC

如果 I²S 主机被用作接收器，该比特为高电平。

Clock

这是 I2S 主机的时钟使能信号。

TR

如果 I2S 主机被用作发送器, 该比特为高电平。

I2S 状态字节格式

每当 CS 为低电平时, SPI 主机接收或者发送 24 比特的数据。24 比特的前 8 位指示状态, 后 16 位对应于接收到的数据。

图 4 所示为 8 位状态字节格式。下一节介绍了每一命令字节。

图 4. I2S 状态字节格式

Current WS	TR/RC	WS/RC	Data Valid	Data Buff	Data Buff	Command Buff	Command Buff
------------	-------	-------	------------	-----------	-----------	--------------	--------------

Current WS

该比特说明了主机所选择的当前通道。

TR/RC

该比特说明了 I2S 主机的状态。如果该比特被设置为 1, I2S 主机当前为发送器。如果该比特被设置为 0, 主机为接收器。

WS/RC

该比特说明了接收数据的通道 (当 I2S 主机是接收器时)。

Data Valid

该比特说明了 16 位数据的有效性。当 I2S 主机是接收器, 该比特被设置为 1 时, 接收到的 16 位数据有效。当 I2S 主机是接收器, 该比特被设置为 0 时, 接收到的 16 位数据无效。

Data Buff

这两个比特说明了数据缓冲的当前状态 (有三个缓冲, 每个都是两个字节, 缓冲要发送的数据)。下面列出了这些缓冲的四种状态:

- 00: 所有缓冲空
- 01: 缓冲 1 和缓冲 2 空, 缓冲 3 满。
- 10: 缓冲 1 空, 缓冲 2 和缓冲 3 满。
- 11: 所有缓冲满

Command Buff

这两个比特说明命令缓冲的当前状态, 其格式与前面的数据缓冲相似。



在 SDA 线上, 数据处理的第 15 个时钟周期时, 命令缓冲的状态发生改变; 而数据缓冲的状态在第 16 个周期时改变。

当桥接 CPLD 接收到命令字节时, 检查最高有效位 (MSB) 是否为 1。只有是 1 时, 将命令字节发送给命令缓冲。跟随在这 8 个比特之后的 16 位数据被存储在数据缓冲中。命令缓冲和数据缓冲可存储三组命令字节和三组数据字。主机每次检查状态, 获得缓冲的状态信息, 决定是否发送新命令字节和数据字。

I²S 主机完成了数据接收后, 状态寄存器的数据有效位被设置为高电平, I²S 数据开始传送给 SPI 从机。可以通过 MISO 引脚将这些数据传送给 SPI 主机。当 MISO 处理完成后, 数据有效位被置为 0。

I²S 接口

MAX II CPLD 的 I²S 端口起到了 I²S 主机的作用, 它和 I²S 总线上的 I²S 从机相连。表 2 列出了 SPI 至 I²S 桥接的 I²S 线。

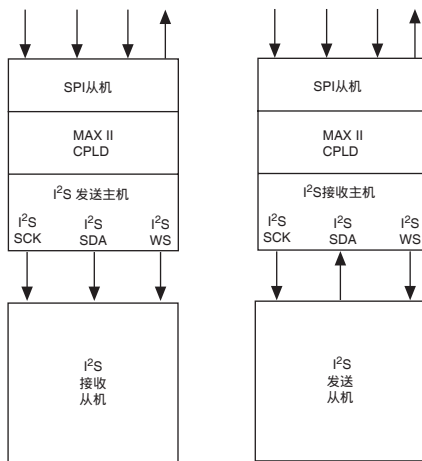
表 2. SPI 至 I²S 桥接的 I²S 线

引脚名称	方向	说明
I2S_SCK	输出	I ² S 串行时钟
I2S_SDA	双向	I ² S 串行时钟
I2S_WS	输出	I ² S 字选择
CLK	输入	输入主机时钟输入

I²S 主机首先开始从 SPI 从机设置的命令缓冲和数据缓冲读取数据。如果对应的命令字节指定了数据传送, 相应的数据缓冲中的数据在 I2S_SDA 线上被串行发送。发送了第 15 个数据比特后, I²S 主机从命令缓冲读取一个新命令, 准备新命令字节中指定的下一操作。这对于 ws 位非常重要,

在向不同的通道传送新数据的下一数据操作之前，它会改变一个时钟周期。I2S 主机现在的作用是发送器，向 I2S 从机接收器发送数据，如图 5 所示。

图 5. I2S 主机的一种实现方式



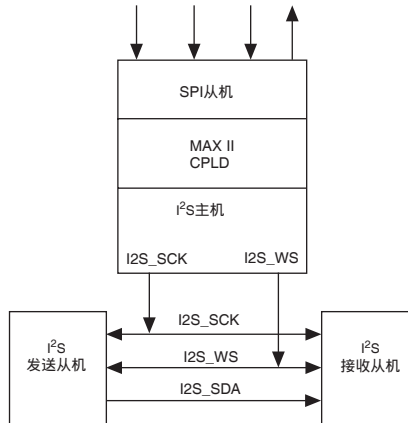
如果 I2S 主机从命令缓冲中读取的命令字节规定了数据接收，那么从 I2S-SDA 线接收数据。从该线上接收了 16 比特数据后，将其发送给 SPI 从机，同时将状态寄存器的数据有效位设置为 1。这样，SPI 从机通过其 MISO 线向 SPI 主机发送数据。状态字节的 WS 位也被更新，指明由哪一通道接收数据。

I2S 主机用作控制器

如果 SPI 主机需要 I2S 主机作为 I2S 控制器使用，命令字节中相应的比特 (controller、RC 和 clock) 被置为高电平。I2S 主机然后使能 I2S_SCK 和 I2S_WS 信号，使 I2S_SDA 线进入三态。

图 6 所示为处于控制器模式下的 I²S 主机。

图 6. 处于控制器模式下的 I²S 主机



设计实现

本设计可以采用 EPM570 或者 EPM1270 来实现。对设计源代码进行编译后，编程到 MAX II CPLD 映射中。SPI 接口和 I²S 总线连接到对应的通用 I/O (GPIO)，如图 1 所示。

源代码

本设计采用了 Verilog 来实现。下面的链接提供了源代码、测试台文件以及完整的 Quartus® II 工程。

www.altera.com/literature/an/an487_design_example.zip

结论

正如本设计所示，MAX II CPLD 是实现 SPI 至 I²S 等接口协议转换非常好的选择。其低功耗、低成本以及易于上电等特性使其成为接口协议转换应用理想的可编程逻辑器件选择。

其他资源

下面列出了和本应用笔记相关的其他资源：

- MAX II CPLD 主页：
<http://www.altera.com/products/devices/cpld/max2/mx2-index.jsp>
- MAX II 器件资料：
<http://www.altera.com/literature/lit-max2.jsp>

- MAX II 关断设计：
<http://www.altera.com/support/examples/max/exm-power-down.html>
- MAX II 应用笔记：
AN 428：MAX II CPLD 设计指南
AN 422：利用 MAX II CPLD 实现便携式系统的功耗管理

文档版本历史

表 3 列出了本应用笔记的版本历史。

日期和文档版本	进行的改动	对改动的总结
2007 年 12 月, 1.0 版	初次发布	—



101 Innovation Drive
San Jose, CA 95134
www.altera.com
Technical Support:
www.altera.com/support
Literature Services:
literature@altera.com

版权 © 2007 Altera 公司。保留所有版权。Altera、可编程解决方案公司、程式化 Altera 标识、专用器件名称和所有其他专有商标或者服务标记，除非特别声明，均为 Altera 公司在美国和其他国家的商标和服务标记。所有其他产品或者服务名称的所有权属于其各自持有人。Altera 产品受美国和其他国家多种专利、未决应用、模板著作权和版权的保护。Altera 保证当前规范下的半导体产品性能与 Altera 标准质保一致，但是保留对产品和服务在没有事先通知时的升级变更权利。除非与 Altera 公司的书面条款完全一致，否则 Altera 不承担由此处所述信息、产品或者服务导致的责任。Altera 建议客户在决定购买产品或者服务，以及确信任何公开信息之前，阅读 Altera 最新版的器件规范说明。



I.S. EN ISO 9001