

第 23 章 串行外设接口（SPI）

目录

本章包括下列主题：

23.1	简介	23-2
23.2	状态和控制寄存器	23-7
23.3	工作模式	23-16
23.4	音频协议接口模式	23-30
23.5	中断	23-50
23.6	节能和调试模式下的操作	23-53
23.7	各种复位的影响	23-54
23.8	使用 SPI 模块的外设	23-54
23.9	相关应用笔记	23-55
23.10	版本历史	23-56

注： 本系列参考手册章节旨在用作对器件数据手册的补充。根据不同的器件型号，本手册章节可能并不适用于所有 PIC32 器件。

请参见当前器件数据手册中“**充电时间测量单元 (CTMU)**”章节开头部分的注释，以检查本文档是否支持您所使用的器件。

器件数据手册和系列参考手册章节可从 Microchip 网站下载：
<http://www.microchip.com>

23.1 简介

串行外设接口 (Serial Peripheral Interface, SPI) 模块是用于同外部外设和其他单片机器件进行通信的同步串行接口。这些外设可以是串行 EEPROM、移位寄存器、显示驱动器和模数转换器 (ADC) 或音频编解码器。PIC32 系列的 SPI 模块与 Motorola® 的 SPI 和 SIOP 接口兼容。图 23-1 显示了 SPI 模块的框图。

该模块的部分主要特性有：

- 支持主 / 从模式
- 4 种不同的时钟格式
- 支持帧 SPI 协议
- 标准和增强型缓冲模式（增强型缓冲模式并非在所有器件上都可用）
- 用户可配置的 8 位、16 位和 32 位数据宽度
- SPI 接收和发送缓冲区是 FIFO 缓冲区，在增强型缓冲模式下为 4/8/16 级深
- 针对每个 8 位、16 位和 32 位数据传输的可编程中断事件
- 音频协议接口模式

一些 PIC32 器件支持音频编解码器串行协议，如用于 16、24 和 32 位音频数据的 I²S、左对齐、右对齐和 PCM/DSP 模式。关于这些特性的可用性，请参见具体器件的数据手册。

SPIx 串行接口由以下四个引脚组成：

- SDIx：串行数据输入
- SDOx：串行数据输出
- SCKx：移位时钟输入或输出
- SSx：低动态功耗从选择或帧同步 I/O 脉冲

表 23-1: SPI 特性

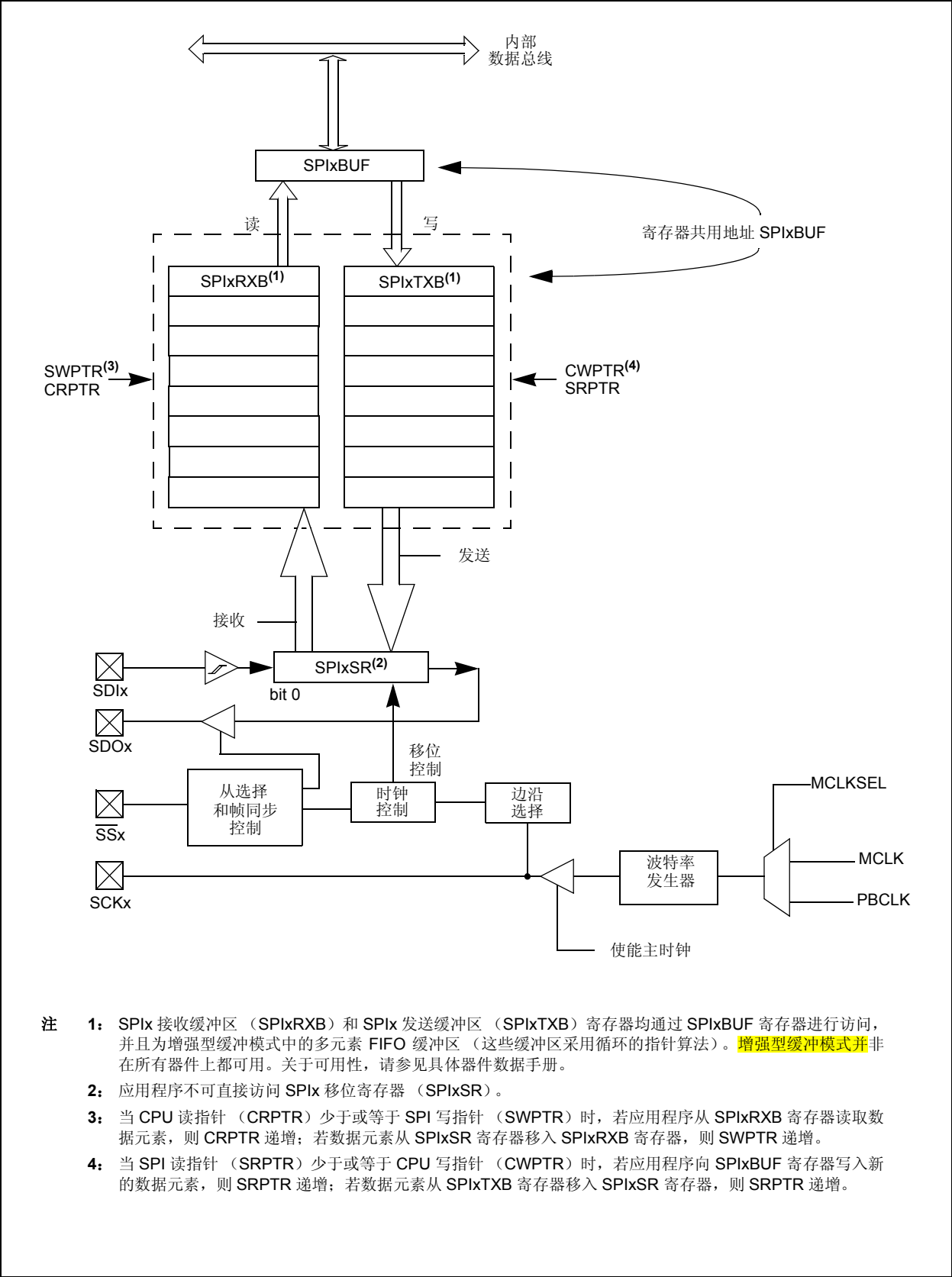
可用的 SPI 模式	SPI 主器件	SPI 从器件	帧主器件	帧从器件	8 位、16 位和 32 位模式	可选的时钟脉冲和边沿	可选的帧同步脉冲和边沿	从选择脉冲
正常模式	有	有	—	—	有	有	—	有
帧模式	有	有	有	有	有	有	有	—

表 23-2: 音频协议接口模式中的 SPI 特性

支持的音频协议	SPI 主器件	SPI 从器件	16/24/32 位数据格式	32/64 位帧	溢出 / 下溢检测	单声道 / 立体声模式	支持主时钟 (MCLK)
I ² S、左对齐、右对齐和 PCM/DSP	有	有	有	有	有	有	有

注 1： 不是所有器件都有该特性。关于可用性，请参见具体器件的数据手册。

图 23-1: SPI 模块框图



- 注
- 1: SPIx 接收缓冲区 (SPIxRXB) 和 SPIx 发送缓冲区 (SPIxTXB) 寄存器均通过 SPIxBUF 寄存器进行访问，并且为增强型缓冲模式中的多元素 FIFO 缓冲区 (这些缓冲区采用循环的指针算法)。增强型缓冲模式并非在所有器件上都可用。关于可用性，请参见具体器件数据手册。
 - 2: 应用程序不可直接访问 SPIx 移位寄存器 (SPIxSR)。
 - 3: 当 CPU 读指针 (CRPTR) 少于或等于 SPI 写指针 (SWPTR) 时，若应用程序从 SPIxRXB 寄存器读取数据元素，则 CRPTR 递增；若数据元素从 SPIxSR 寄存器移入 SPIxRXB 寄存器，则 SWPTR 递增。
 - 4: 当 SPI 读指针 (SRPTR) 少于或等于 CPU 写指针 (CWPTR) 时，若应用程序向 SPIxBUF 寄存器写入新的数据元素，则 SRPTR 递增；若数据元素从 SPIxTXB 寄存器移入 SPIxSR 寄存器，则 SRPTR 递增。

23.1.1 正常模式 SPI 操作

在正常模式下工作时，SPI 主器件控制串行时钟的产生。输出时钟脉冲的数量对应于传输数据宽度：8、16 或 32 位。图 23-2 和图 23-3 给出了 SPI 主 - 从器件和从 - 主器件连接图。

图 23-2: 典型的 SPI 主器件—从器件连接图

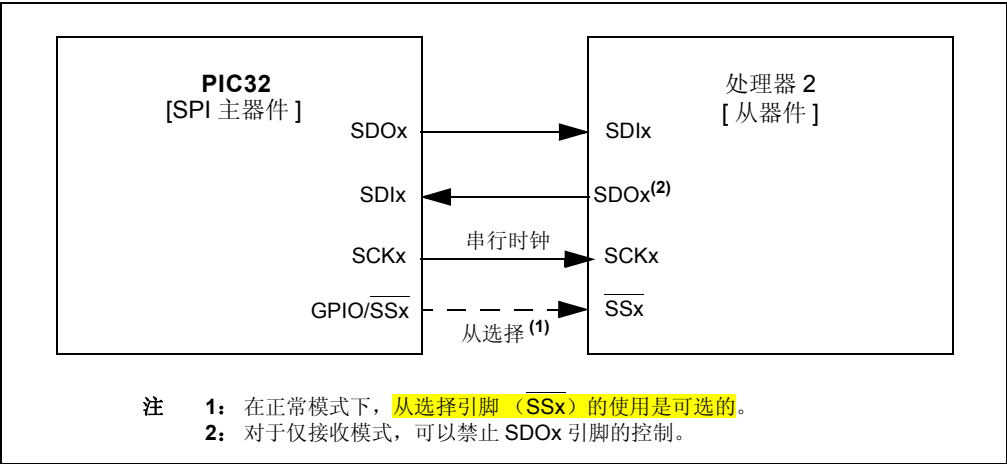
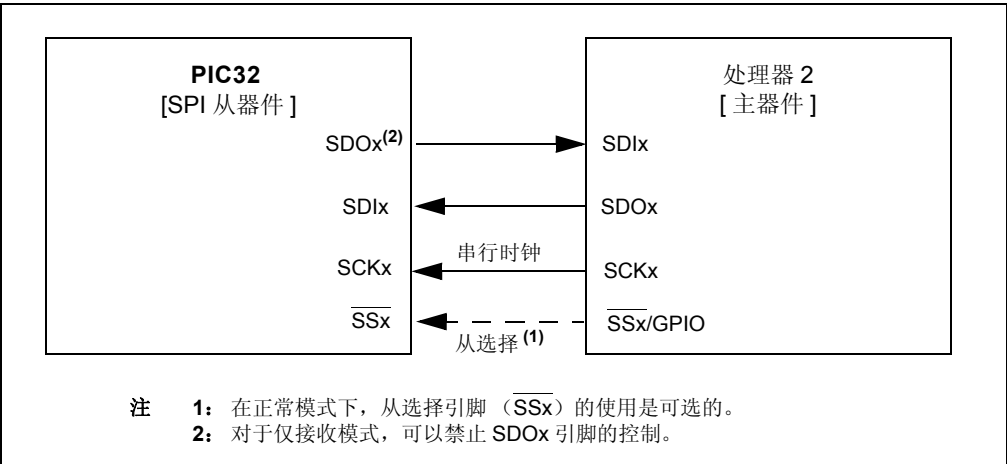


图 23-3: 典型的 SPI 从器件—主器件连接图



23.1.2 帧模式 SPI 操作

在帧模式下工作时，帧主器件控制帧同步脉冲的产生。SPI 时钟仍由 SPI 主器件产生，并连续运行。图 23-4 和图 23-5 给出了 SPI 帧主器件和帧从器件连接图。

图 23-4: 典型的 SPI 主器件—帧主器件连接图

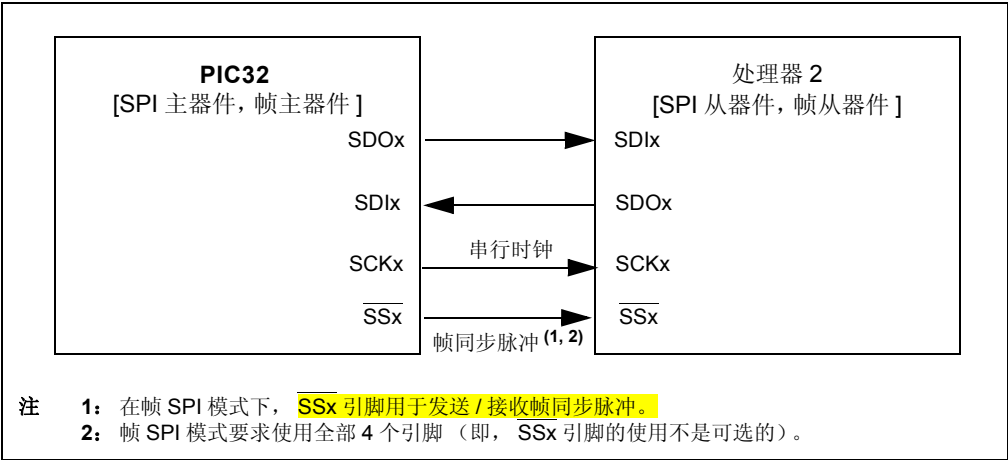
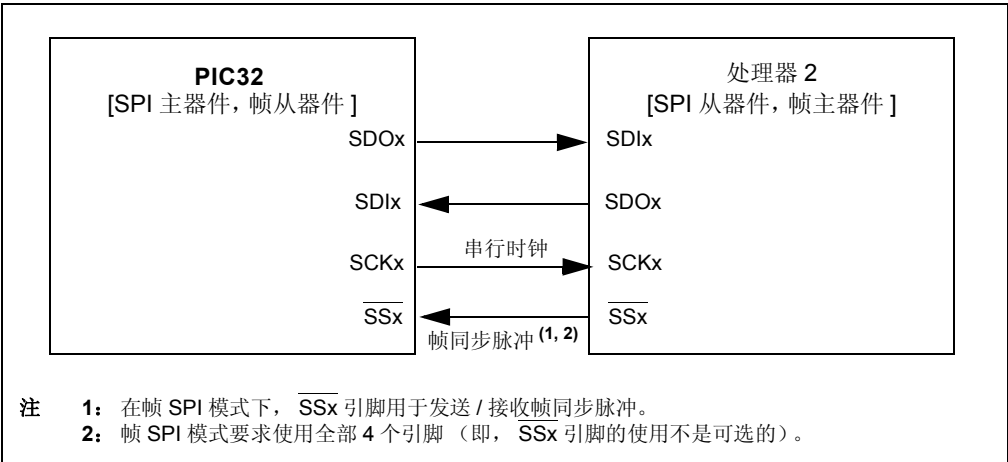


图 23-5: 典型的 SPI 主器件—帧从器件连接图

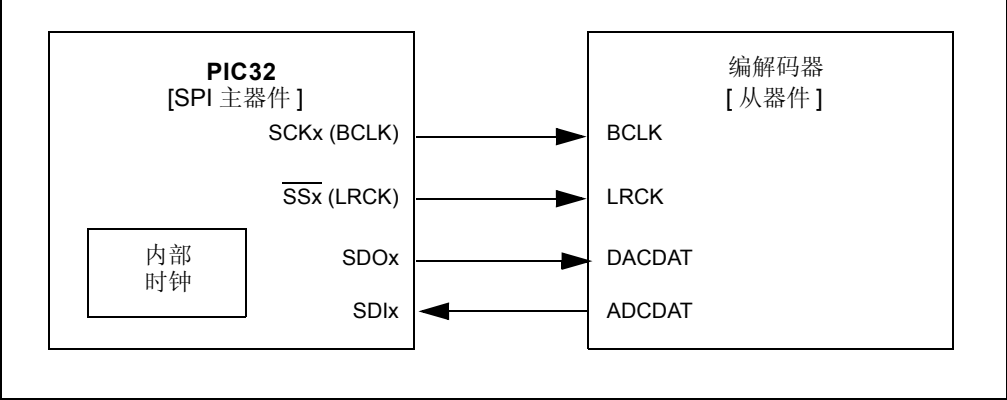


23.1.3 音频协议接口模式

23.1.3.1 SPI 音频主模式连接到编解码器从模式

图 23-6 显示了由 PIC32 SPI 模块产生的位时钟 (Bit Clock, BCLK) 和左/右通道时钟 (Left/Right Channel Clock, LRCK)。

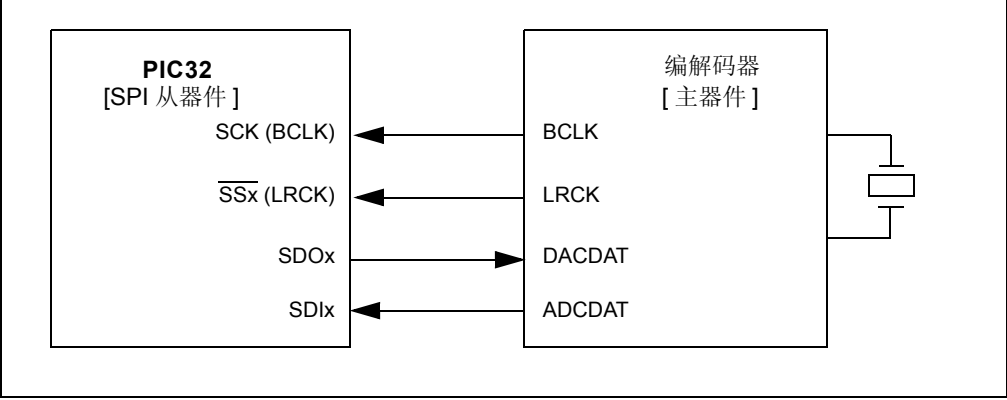
图 23-6: 主器件自行生成时钟——输出 BCLK 和 LRCK



23.1.3.2 SPI 音频从模式连接到编解码器主模式

图 23-7 显示了由编解码器主器件产生的 BCLK 和 LRCK。

图 23-7: 作为主器件的编解码器通过外部晶振产生所需时钟



23.2 状态和控制寄存器

注： 不同的 PIC32 系列器件型号可能具有一个或多个 SPI 模块。在引脚、控制 / 状态位和寄存器的名称中使用的“x”表示特定的模块。更多详细信息，请参见具体器件数据手册。

SPI 模块包含以下特殊功能寄存器（Special Function Register，SFR）：

- **SPIx CON：SPI 控制寄存器**
- **SPIxCON2：SPI 控制寄存器 2**
- **SPIxSTAT：SPI 状态寄存器**
- **SPIxBUF：SPI 缓冲寄存器**
- **SPIx BRG：SPI 波特率寄存器**

表 23-3 汇总了所有与 SPI 相关的寄存器。该汇总表之后列出了相应的寄存器，并且每个寄存器均附有详细的说明。

表 23-3: SPI SFR 汇总

名称	位 范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
SPIxCON ^(1,2,3)	31:24	FRMEN	FRMSYNC	FRMPOL	MSEN ⁽⁴⁾	FRMSYPW ⁽⁴⁾	FRMCNT<2:0> ⁽⁴⁾		
	23:16	MCLKSEL ⁽⁴⁾	—	—	—	—	—	SPIFE	ENHBUF ⁽⁴⁾
	15:8	ON	—	SIDL	DISSDO	MODE32	MODE16	SMP	CKE
	7:0	SSEN	CKP	MSTEN	DISSDI ⁽⁴⁾	STXISEL<1:0> ⁽⁴⁾		SRXISEL<1:0> ⁽⁴⁾	
SPIxCON2 ^(1,2,3,5)	31:24	—	—	—	—	—	—	—	—
	23:16	—	—	—	—	—	—	—	—
	15:8	SPIGNEXT	—	—	FRMERREN	SPIROVEN	SPITUREN	IGNROV	IGNTUR
	7:0	AUDEN	—	—	—	AUDMONO	—	AUDMOD<1:0>	
SPIxSTAT ^(1,2,3)	31:24	—	—	—	RXBUFELM<4:0> ⁽⁴⁾				
	23:16	—	—	—	TXBUFELM<4:0> ⁽⁴⁾				
	15:8	—	—	—	FRMERR ⁽⁴⁾	SPIBUSY	—	—	SPITUR
	7:0	SRMT ⁽⁴⁾	SPIROV	SPIRBE ⁽⁴⁾	—	SPITBE	—	SPITBF ⁽⁴⁾	SPIRBF
SPIxBUF	31:24	DATA<31:24>							
	23:16	DATA<23:16>							
	15:8	DATA<15:8>							
	7:0	DATA<7:0>							
SPIxBRG ^(1,2,3)	31:24	—	—	—	—	—	—	—	—
	23:16	—	—	—	—	—	—	—	—
	15:8	—	—	—	BRG<12:8> ⁽⁶⁾				
	7:0	BRG<7:0> ⁽⁶⁾							

图注： — = 未实现，读为 0。地址偏移值以十六进制显示。

注

- 1: 该寄存器具有关联的清零寄存器，位于 0x4 字节偏移处。这些清零寄存器的命名方式是在关联寄存器的名称末尾附加 CLR（例如，SPIxCONCLR）。向清零寄存器的任意位写入 1 时，会将关联寄存器中的有效位清零。对清零寄存器的读操作将被忽略。
- 2: 该寄存器具有关联的置 1 寄存器，位于 0x8 字节偏移处。这些置 1 寄存器的命名方式是在关联寄存器的名称末尾附加 SET（例如，SPIxCONSET）。向置 1 寄存器的任意位写入 1 时，会将关联寄存器中的有效位置 1。对置 1 寄存器的读操作将被忽略。
- 3: 该寄存器具有关联的取反寄存器，位于 0xC 字节偏移处。这些取反寄存器的命名方式是在关联寄存器的名称末尾附加 INV（例如，SPIxCONINV）。向取反寄存器的任意位写入 1 时，会将关联寄存器中的有效位取反。对取反寄存器的读操作将被忽略。
- 4: 该位并非在所有器件上都可用。详情请参见具体器件数据手册。
- 5: 该寄存器并非在所有器件上都可用。详情请参见具体器件数据手册。
- 6: 根据所选器件，BRG 最多可达 13 位。详情请参见具体器件数据手册。

寄存器 23-1: SPIxCON: SPI 控制寄存器

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	FRMEN	FRMSYNC	FRMPOL	MSEN ^(1,2)	FRMYPW ⁽¹⁾	FRMCNT<2:0> ⁽¹⁾		
23:16	R/W-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
	MCLKSEL	—	—	—	—	—	SPIFE	ENHBUF ⁽¹⁾
15:8	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	ON	—	SIDL	DISSDO	MODE32	MODE16	SMP	CKE
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	SSEN	CKP	MSTEN	DISSDI	STXISEL<1:0> ^(1,3)		SRXISEL<1:0> ^(1,3)	

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 31 **FRMEN:** 帧 SPI 支持位

1 = 使能帧 SPI 支持 (SSx 引脚用作 FSYNC 输入 / 输出)

0 = 禁止帧 SPI 支持

bit 30 **FRMSYNC:** SSx 引脚上的帧同步脉冲方向控制位 (仅适用于帧 SPI 模式)

1 = 帧同步脉冲输入 (从模式)

0 = 帧同步脉冲输出 (主模式)

bit 29 **FRMPOL:** 帧同步脉冲极性位 (仅适用于帧 SPI 模式)

1 = 帧同步脉冲为高电平有效

0 = 帧同步脉冲为低电平有效

bit 28 **MSEN:** 主模式从选择使能位 ^(1,2)

1 = 使能从选择 SPI 支持。在主模式下, 在发送期间会自动驱动 SS 引脚。极性由 FRMPOL 位决定。

0 = 禁止从选择 SPI 支持

bit 27 **FRMSYPW:** 帧同步脉冲宽度位 ⁽¹⁾

1 = 帧同步脉冲为一个字符宽, 由 MODE<32,16> 位 (SPIxCON<11:10>) 定义

0 = 帧同步脉冲为一个时钟宽

bit 26-24 **FRMCNT<2:0>:** 帧同步脉冲计数器位

该位控制每个脉冲发送的数据字符数 ⁽¹⁾。

111 = 保留; 不要使用

110 = 保留; 不要使用

101 = 对于每 32 个数据字符产生一个帧同步脉冲

100 = 对于每 16 个数据字符产生一个帧同步脉冲

011 = 对于每 8 个数据字符产生一个帧同步脉冲

010 = 对于每 4 个数据字符产生一个帧同步脉冲

001 = 对于每 2 个数据字符产生一个帧同步脉冲

000 = 对于每 1 个数据字符产生一个帧同步脉冲

该位仅在帧 SPI 模式下有效 (FRMEN = 1)。

bit 23 **MCLKSEL:** 主时钟选择位 ⁽²⁾

1 = 波特率发生器使用 MCLK

0 = 波特率发生器使用 PBCLK

bit 22-18 **未实现:** 写为 0, 忽略读操作

注 1: 这些位并非在所有器件上都可用。详情请参见具体器件数据手册。

2: 当 FRMEN = 1 时, 不使用 MSEN 位。

3: 仅当增强型缓冲区 (ENHBUF = 1) 使能时才有效。

寄存器 23-1: SPIxCON: SPI 控制寄存器 (续)

bit 17	SPIFE: 帧同步脉冲边沿选择位 (仅适用于帧 SPI 模式)	
	1 = 帧同步脉冲与第一个位时钟一致	
	0 = 帧同步脉冲比第一个位时钟提前	
bit 16	ENHBUF: 增强型缓冲区使能位 ⁽¹⁾	
	1 = 使能增强型缓冲模式	
	0 = 禁止增强型缓冲模式	
bit 15	ON: SPI 外设使能位	
	1 = 使能 SPI 外设	
	0 = 禁止 SPI 外设	
	当 ON = 1 时, DISSDO 和 DISSDI 是仅能修改的两个位。使用 1:1 PBCLK 分频比时, 在清零模块 ON 位的指令之后, 用户的软件不应立即在 SYSCLK 周期中读 / 写外设的 SFR。	
bit 14	未实现: 写为 0, 忽略读操作	
bit 13	SIDL: 空闲模式停止位	
	1 = 在 CPU 进入 Idle (空闲) 模式时停止工作	
	0 = 在 Idle (空闲) 模式下继续工作	
bit 12	DISSDO: 禁止 SDOx 引脚位	
	1 = 模块不使用 SDOx 引脚 (引脚由关联的 PORT 寄存器控制)	
	0 = SDOx 引脚由模块控制	
bit 11-10	MODE<32,16>: 32/16 位通信选择位	
	当 AUDEN = 1 时:	
	MODE32	MODE16 通信
	1	1 24 位数据、32 位 FIFO 和 32 位通道 /64 位帧
	1	0 32 位数据、32 位 FIFO 和 32 位通道 /64 位帧
	0	1 16 位数据、16 位 FIFO 和 32 位通道 /64 位帧
	0	0 16 位数据、16 位 FIFO 和 16 位通道 /32 位帧
	当 AUDEN = 0 时:	
	MODE32	MODE16 通信
	1	x 32 位
	0	1 16 位
	0	0 8 位
bit 9	SMP: SPI 数据输入采样阶段位	
	<u>主模式 (MSTEN = 1):</u>	
	1 = 在数据输出时间的末端采样输入数据	
	0 = 在数据输出时间的中间采样输入数据	
	<u>从模式 (MSTEN = 0):</u>	
	当在从模式下使用 SPI 时, 忽略 SMP 值。模块总是使用 SMP = 0。	
bit 8	CKE: SPI 时钟边沿选择位	
	1 = 串行输出数据在时钟从工作状态转变为空闲状态时变化 (见 CKP 位)	
	0 = 串行输出数据在时钟从空闲状态转变为工作状态时变化 (见 CKP 位)	
	在帧 SPI 模式下不使用 CKE 位。在帧 SPI 模式 (FRMEN = 1) 下, 用户应将该位编程为 0。	
bit 7	SSEN: 从选择使能 (从模式) 位	
	1 = SSx 引脚用于从模式	
	0 = 从模式不使用 SSx 引脚, 引脚由端口功能控制	
bit 6	CKP: 时钟极性选择位	
	1 = 空闲状态时时钟信号为高电平; 工作状态时为低电平	
	0 = 空闲状态时时钟信号为低电平; 工作状态时为高电平	
注	1: 这些位并非在所有器件上都可用。详情请参见具体器件数据手册。	
	2: 当 FRMEN = 1 时, 不使用 MSSEN 位。	
	3: 仅当增强型缓冲区 (ENHBUF = 1) 使能时才有效。	

寄存器 23-1: SPIxCON: SPI 控制寄存器 (续)

bit 5	MSTEN: 主模式使能位 1 = 主模式 0 = 从模式
bit 4	DISSDI: SDI 禁止位 1 = SPI 模块不使用 SDIx 引脚 (引脚由 PORT 功能控制) 0 = SDIx 引脚由 SPI 模块控制
bit 3-2	STXISEL<1:0>: SPI 发送缓冲区为空的中断模式位 (1,3) 11 = SPIxTXIF 在缓冲区未满 (具有一个或多个空元素) 时置 1 10 = SPIxTXIF 在缓冲区有一半或更多元素为空时置 1 01 = SPIxTXIF 在缓冲区全空时置 1 00 = SPIxTXIF 在最后一个传输数据移出 SPISR 并且发送操作完成时置 1
bit 1-0	RTXISEL<1:0>: SPI 接收缓冲区为满的中断模式位 (1,3) 11 = SPIxRXIF 在缓冲区全满时置 1 10 = SPIxRXIF 在缓冲区有一半或更多元素不为空时置 1 01 = SPIxRXIF 在缓冲区不为空时置 1 00 = SPIxRXIF 在接收缓冲区中的最后一个字被读取 (即, 缓冲区为空) 时置 1

- 注
- 1: 这些位并非在所有器件上都可用。详情请参见具体器件数据手册。
 - 2: 当 FRMEN = 1 时, 不使用 MSSEN 位。
 - 3: 仅当增强型缓冲区 (ENHBUF = 1) 使能时才有效。

寄存器 23-2: SPIxCON2: SPI 控制寄存器 2

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0
	SPISGNXT	—	—	FRMER- REN	SPIROVEN	SPIUREN	IGNROV	IGNTUR
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	AUDEN ^(1,3)	—	—	—	AUDONO ⁽²⁾	—	AUDMOD<1:0> ^(2,4)	

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 31-16 未实现: 写为 0, 忽略读操作

bit 15 **SPISGNEXT**: 从 RX FIFO 读取符号扩展数据位

1 = 来自 RX FIFO 的数据为符号扩展数据

0 = 来自 RX FIFO 的数据不为符号扩展数据

bit 14-13 未实现: 写为 0, 忽略读操作

bit 12 **FRMERREN**: 通过 FRMERR 允许中断事件位

1 = 帧错误溢出产生错误中断

0 = 帧错误溢出不产生错误中断

bit 11 **SPIROVEN**: 通过 SPIROV 允许中断事件位

1 = 接收溢出产生错误中断

0 = 接收溢出不产生错误中断

bit 10 **SPITUREN**: 通过 SPITUR 允许中断事件位

1 = 发送不足产生错误中断

0 = 发送不足不产生错误中断

bit 9 **IGNROV**: 忽略接收溢出位 (用于音频数据传输)

1 = ROV 不是关键性错误, 在 ROV 数据位于 FIFO 期间不被接收数据覆盖

0 = ROV 是关键性错误, 停止 SPI 操作

bit 8 **IGNTUR**: 忽略发送不足位 (用于音频数据传输)

1 = TUR 不是关键性错误, 发送零直至 SPIxTXB 不为空

0 = TUR 是关键性错误, 停止 SPI 操作

bit 7 **AUDEN**: 支持音频编解码器使能位 ^(1,3)

1 = 使能音频协议

0 = 禁止音频协议

bit 6-4 未实现: 写为 0, 忽略读操作

注 1: 当 ON 位 = 0 时, 该位为只写位。

2: 当 ON 位 = 0 时, 该位为只写位, 且仅在 AUDEN = 1 时有效。

3: 使能音频模式 (即 AUDEN = 1) 时, 以下 SPIxCON 寄存器中的位在模块内部配置为:

• 根据 MSTEN 位选择位时钟 (BCLK) 和左 / 右通道时钟 (LRCK) 的方向

• FRMEN = 1、FRMCNT = 1 且 SMP = 0

• 在从模式下, MSTN = 0 且 FRMSYNC = 1; 在主模式下, MSTN = 1 且 FRMSYNC = 0

4: FRMSYPW = 0 时, 在 I²S 模式下, SPIFE = 0; 在右或左对齐模式下, SPIFE = 1; DSP/PCM 模式除外。

5: 并非所有器件都有该特性。关于可用性, 请参见具体器件数据手册。

寄存器 23-2: SPIxCON2: SPI 控制寄存器 2 (续)

- bit 3 **AUDMONO:** 发送音频数据格式位 ⁽²⁾
1 = 音频数据为单声道 (每个数据字同时在左右通道上发送)
0 = 音频数据为立体声
- bit 2 **未实现:** 写为 0, 忽略读操作
- bit 1-0 **AUDMOD<1:0>:** 音频协议模式位 ^(2,4)
11 = PCM/DSP 模式
10 = 右对齐位
01 = 左对齐位
00 = I²S 模式 ⁽⁵⁾

- 注 **1:** 当 ON 位 = 0 时, 该位为只写位。
- 2:** 当 ON 位 = 0 时, 该位为只写位, 且仅在 AUDEN = 1 时有效。
- 3:** 使能音频模式 (即 AUDEN = 1) 时, 以下 SPIxCON 寄存器中的位在模块内部配置为:
- 根据 MSTEN 位选择位时钟 (BCLK) 和左 / 右通道时钟 (LRCK) 的方向
 - FRMEN = 1、FRMCNT = 1 且 SMP = 0
 - 在从模式下, MSTN = 0 且 FRMSYNC = 1; 在主模式下, MSTN = 1 且 FRMSYNC = 0
- 4:** FRMSYPW = 0 时, 在 I²S 模式下, SPIFE = 0; 在右或左对齐模式下, SPIFE = 1; DSP/PCM 模式除外。
- 5:** 并非所有器件都有该特性。关于可用性, 请参见具体器件数据手册。

寄存器 23-3: SPIxSTAT: SPI 状态寄存器

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	R-0	R-0	R-0	R-0	R-0
	—	—	—	RXBUFELM<4:0> ⁽¹⁾				
23:16	U-0	U-0	U-0	R-0	R-0	R-0	R-0	R-0
	—	—	—	TXBUFELM<4:0> ⁽¹⁾				
15:8	U-0	U-0	U-0	R/C-0, HS	R-0	U-0	U-0	R/C-0,HS
	—	—	—	FRMERR	SPIBUSY	—	—	SPITUR ⁽¹⁾
7:0	R-0	R/C-0,HS	R-0	U-0	R-1	U-0	R-0	R-0
	SRMT ⁽¹⁾	SPIROV	SPIRBE ⁽¹⁾	—	SPITBE	—	SPITBF ⁽¹⁾	SPIRBF

图注:

C = 可清零位

HS = 硬件置 1

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 31-29 **未实现**: 写为 0; 忽略读操作bit 28-24 **RXBUFELM<4:0>**: 接收缓冲区元素计数位 (仅在 ENHBUF = 1 时有效) ⁽¹⁾

接收采样的数量存放于 FIFO 中。

bit 23-21 **未实现**: 写为 0; 忽略读操作bit 20-16 **TXBUFELM<4:0>**: 发送缓冲区元素计数位 (仅在 ENHBUF = 1 时有效) ⁽¹⁾

接收采样的数量存放于 FIFO 中。

bit 15-13 **未实现**: 写为 0; 忽略读操作bit 12 **FRMERR**: SPI 帧错误状态位

1 = 检测到帧错误

0 = 未检测到帧错误

FRMERR 仅在 FRMEN = 1 时有效。该位只能由硬件置 1。可通过写入零来清零, 建议通过命令 SPIxSTATCLR = 1<<12 实现。也可以通过 SPIxCON 中的 ON 位禁止和重新使能模块来清零。

bit 11 **SPIBUSY**: SPI 活动状态位

1 = SPI 外设当前正忙于执行某些事务

0 = SPI 外设当前空闲

bit 10-9 **未实现**: 写为 0; 忽略读操作bit 8 **SPITUR**: 发送数据不足位 ⁽¹⁾

1 = 发送缓冲区遇到了数据不足条件

0 = 发送缓冲区未产生数据不足条件

该位仅在帧同步模式下有效。该位只能由硬件置 1。可通过写入零来清零, 建议通过命令 SPIxSTATCLR = 1<<12 实现。也可以通过 SPIxCON 中的 ON 位禁止和重新使能模块来清零。

bit 7 **SRMT**: 移位寄存器空位 (仅当 ENHBUF = 1 时有效) ⁽¹⁾

1 = 当 SPI 模块移位寄存器为空时

0 = 当 SPI 模块移位寄存器不为空时

bit 6 **SPIROV**: 接收溢出标志位

1 = 一个新数据已完全接收并丢弃。用户软件尚未读取 SPIxBUF 寄存器中的先前数据。

0 = 未发生溢出

该位只能由硬件置 1。可通过写入零来清零, 建议通过命令 SPIxSTATCLR = 1<<12 实现。也可以通过 SPIxCON 中的 ON 位禁止和重新使能模块来清零。

注 1: 这些位并非在所有器件上都可用。详情请参见具体器件数据手册。

寄存器 23-3: SPIxSTAT: SPI 状态寄存器 (续)

- bit 5 **SPIRBE**: 接收 FIFO 空位 (仅当 ENHBUF = 1 时有效)
1 = 接收 FIFO 为空 (CRPTR = SWPTR)
0 = 接收 FIFO 非空 (CRPTR < SWPTR)
- bit 4 **未实现**: 写为 0; 忽略读操作
- bit 3 **SPITBE**: SPI 发送缓冲区空状态位 ⁽¹⁾
1 = 发送缓冲区 SPIxTXB 为空
0 = 发送缓冲区 SPIxTXB 非空
当 SPI 将数据从 SPIxTXB 传输到 SPIxSR 时, 该位由硬件自动置 1。
当通过写 SPIxBUF 装入 SPIxTXB 时, 该位由硬件自动清零。
- bit 2 **未实现**: 写为 0; 忽略读操作
- bit 1 **SPITBF**: SPI 发送缓冲区满状态位 ⁽¹⁾
1 = 发送尚未开始, SPIxTXB 为满
0 = 发送缓冲区未满
标准缓冲模式:
当内核通过写 SPIxBUF 存储单元装入 SPIxTXB 时, 该位由硬件自动置 1。
当 SPI 模块将数据从 SPIxTXB 传输到 SPIxSR 时, 该位由硬件自动清零。
增强型缓冲模式:
当 FIFO 中没有可用空间时置 1。
- bit 0 **SPIRBF**: SPI 接收缓冲区满状态位
1 = 接收缓冲区 SPIxRXB 为满
0 = 接收缓冲区 SPIxRXB 未满
标准缓冲模式:
当 SPI 模块将数据从 SPIxSR 传输到 SPIxRXB 时, 该位由硬件自动置 1。
当通过读 SPIxBUF 进行 SPIxRXB 读取时, 该位由硬件自动清零。
增强型缓冲模式:
当 FIFO 中没有可用空间时置 1。

注 1: 这些位并非在所有器件上都可用。详情请参见具体器件数据手册。

寄存器 23-4: SPIxBUF: SPI 缓冲寄存器

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	DATA<31:24>							
23:16	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	DATA<23:16>							
15:8	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	DATA<15:8>							
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	DATA<7:0>							

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 31-0

DATA<31:0>: SPI 发送 / 接收缓冲寄存器

用作发送 (SPIxTXB) 和接收 (SPIxSR) 寄存器的存储器映射值。

当使能 32 位数据模式 (MODE<32,16> (SPIxCON<11:10>) = 1x) 时:

该寄存器的全部 32 位 (SPIxBUF<31:0>) 都用于构成 32 位字符。

当使能 16 位数据模式 (MODE<32,16> (SPIxCON<11:10>) = 01) 时:

只有该寄存器的低 16 位 (SPIxBUF<15:0>) 用于构成 16 位字符。

当使能 8 位数据模式 (MODE<32,16> (SPIxCON<11:10>) = 00) 时:

只有该寄存器的低 8 位 (SPIxBUF<7:0>) 用于构成 8 位字符。

寄存器 23-5: SPIxBRG: SPI 波特率寄存器

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	—	—	BRG<12:8>				
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	BRG<7:0>							

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 31-13

未实现: 写为 0; 忽略读操作

bit 12-0

BRG<12:0>: 波特率分频比位

23.3 工作模式

SPI 模块提供以下工作模式：

- 8 位、16 位和 32 位数据发送模式
- 8 位、16 位和 32 位数据接收模式
- 主 / 从模式
- 帧 SPI 模式
- 音频协议接口模式

注 1: 在帧 SPI 模式下，使用以下 4 个引脚：SDIx、SDOx、SCKx 和 SSx。
注 2: 如果使用从选择功能，则使用注 1 中列出的 4 个引脚。
注 3: 如果使用标准 SPI 模式，但 CKE = 1，同时强制使能 / 使用从选择功能，则使用注 1 中列出的 4 个引脚。
注 4: 如果使用标准 SPI 模式，但 DISSDO = 1，则只使用 2 个引脚：SDIx 和 SCKx；除非还使能了从选择功能。
注 5: 其他情况下，使用 3 个引脚：SDIx、SDOx 和 SCKx。

23.3.1 8 位、16 位和 32 位操作

在通过 SPI 总线发送和接收数据时，SPI 模块支持三种数据宽度。数据宽度的选择决定 SPI 数据的最小长度。例如，当选择的数据宽度为 32 时，所有发送和接收都按 32 位值执行。来自 CPU 的所有读写操作也都按 32 位值执行。相应地，应用程序软件应选择适当的数据宽度，以最大程度提高它的数据吞吐量。

两个控制位 MODE32 和 MODE16（SPIxCON<11:10>）（即 MODE<32,16>）用于定义工作模式。要动态更改工作模式，SPI 模块必须处于空闲状态（即，不在执行任何事务）。如果 SPI 模块被关闭（SPIxCON<15> = 0），则新模式将在模块再次开启时可用。

此外，还应注意以下几点：

- 在事务执行过程时，不应更改 MODE<32,16> 位
- 从 SPIxSR 移出的第一个位会因所选工作模式而不同：
 - 8 位模式，bit 7
 - 16 位模式，bit 15
 - 32 位模式，bit 31
- 在每种模式下，数据都会移入 SPIxSR 的 bit 0
- SCKx 引脚上的时钟脉冲数量也取决于所选工作模式：
 - 8 位模式，8 个时钟
 - 16 位模式，16 个时钟
 - 32 位模式，32 个时钟

23.3.2 缓冲模式

有两种 SPI 缓冲模式：标准和增强型。

注： 增强型缓冲模式并非在所有器件上都可用。详情请参见具体器件数据手册。

23.3.2.1 标准缓冲模式

SPI 数据接收 / 发送缓冲（SPIxBUF）寄存器实际上是两个独立的内部寄存器：发送缓冲区（SPIxTXB）和接收缓冲区（SPIxRXB）。这两个单向的寄存器共用 SPIxBUF 的 SFR 地址。

当接收到完整字节 / 字时，它会从 SPIxSR 传输到 SPIxRXB，并且 SPIxRBF 标志会置 1。如果软件读取 SPIxBUF 缓冲区，SPIxRBF 位会被清零。

在软件写入 SPIxBUF 时，数据会被装入 SPIxTXB，SPIxTBF 位会由硬件置 1。在数据从 SPIxSR 中发送出去时，SPIxTBF 标志会被清零。

SPI 模块对发送 / 接收操作进行了双重缓冲，允许在后台连续地传输数据。发送和接收在 SPIxSR 中同时进行。

23.3.2.2 增强型缓冲模式

可以通过将 SPI 控制寄存器中的增强型缓冲使能位 ENHBUF (SPIxCON<16>) 置 1 来使能增强型缓冲模式。

在增强型缓冲模式下，将使用 2 个 128 位的 FIFO 缓冲区来作为发送缓冲区 (SPIxTXB) 和接收缓冲区 (SPIxRXB)。SPIxBUF 可用于访问接收和发送 FIFO，并且在该模式下，SPIxSR 缓冲区中的数据发送和接收与标准缓冲模式下的相同。FIFO 深度取决于通过 SPI 控制寄存器中的字 / 半字节通信选择位 MODE<32,16> (SPIxCON<11:10>) 选择的数据宽度。如果 MODE 字段选择 32 位数据长度，则 FIFO 为 4 级深；如果 MODE 选择 16 位数据长度，则 FIFO 为 8 级深；或者如果 MODE 选择 8 位数据长度，则 FIFO 为 16 级深。

当发送 FIFO 缓冲区中的元素全满时，SPIxTBF 状态位会置 1，在其中有一个或多个元素为空时，该位会清零。当接收 FIFO 缓冲区中的元素全满时，SPIxRBF 状态位会置 1，在软件读取 SPIxBUF 缓冲区时，该位会清零。

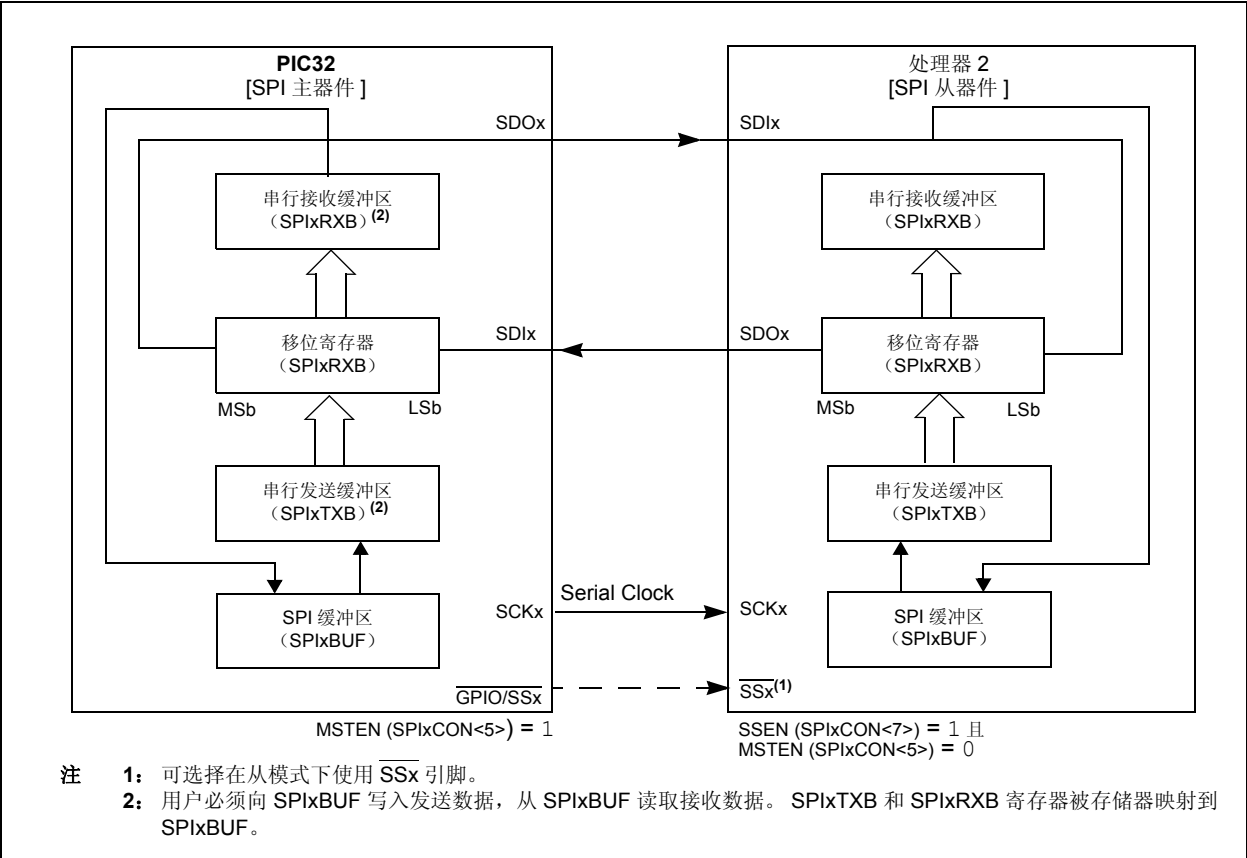
如果发送 FIFO 缓冲区中的元素全为空，SPIxTBE 状态位会置 1，否则该位会清零。如果接收 FIFO 缓冲区中的元素全为空，SPIxRBE 位会置 1，否则该位会清零。只有在增强型缓冲模式下，移位寄存器空 (SRMT) 位才有效，该位在移位寄存器为空时置 1，其他情况下则清零。

器件没有用于防止以下操作的数据不足或溢出保护：读取为空的接收 FIFO 元素或写入已满的发送 FIFO 元素。然而，SPIxSTAT 寄存器提供了发送数据不足状态位 (SPITUR) 和接收数据溢出状态位 (SPIROV)，它们可以与其他状态位一起用于进行监视。

SPI 状态寄存器中的接收缓冲区元素计数位 RXBUFELM<4:0> (SPIxSTAT<28:24>) 用于指示接收 FIFO 中未读元素的数量。SPI 状态寄存器中的发送缓冲区元素计数位 TXBUFELM<4:0> (SPIxSTAT<20:16>) 用于指示发送 FIFO 中未发送元素的数量。

23.3.3 主 / 从模式

图 23-8: SPI 主 / 从器件连接图



23.3.3.1 主模式工作

要将 SPI 模块设置为主模式工作，请执行以下步骤：

1. 禁止相应 IECx 寄存器中的 SPI 中断。
2. 通过清零 ON 位停止并复位 SPI 模块。
3. 清除接收缓冲区。
4. 如果使用标准缓冲模式，则将 ENHBUF 位 (SPIxCON<16>) 清零，如果使用增强型缓冲模式，则将它置 1。
5. 如果不希望使用 SPI 中断，则跳过此步骤并继续步骤 5。否则，需要执行以下附加步骤：
 - a) 清零相应 IFSx 寄存器中的 SPIx 中断标志 / 事件。
 - b) 在相应的 IPCx 寄存器中写入 SPIx 中断优先级位和子优先级位。
 - c) 将相应 IECx 寄存器中的 SPIx 中断允许位置 1。
6. 写入波特率发生器寄存器 SPIxBRG。
7. 清零 SPIROV 位 (SPIxSTAT<6>)。
8. 将所需的设置写入 SPIxCON 寄存器，且 MSTEN (SPIxCON<5>) = 1。
9. 通过将 ON 位 (SPIxCON<15>) 置 1 使能 SPI 工作。
10. 将待发送数据写入 SPIxBUF 寄存器。发送 (和接收) 在数据写入 SPIxBUF 寄存器时立即开始。

注： 在将模式从从模式更改为主模式之前，必须先关闭 SPI 器件。(使用从选择模式时，SSx 或另一个 GPIO 引脚用于控制从器件的 SSx 输入。引脚必须用软件进行控制。)

在主模式下，PBCLK 被分频，然后被用作串行时钟。分频基于 SPIxBRG 寄存器中的设置。串行时钟通过 SCKx 引脚输出到从器件。只有存在待发送数据时，才会产生时钟脉冲；处于帧模式时除外，此时会连续产生时钟。更多信息，请参见第 23.3.7 节“SPI 主模式时钟频率”。

在主模式下，可以通过将 SPI 控制寄存器中的主模式从选择使能位 MSSEN (SPIxCON<28>) 置 1 来自动驱动从选择信号 (SS)。清零该位会禁止主模式下的从选择信号支持。FRMPOL 位 (SPIxCON<29>) 决定主模式下从选择信号的极性。

注： MSSEN 位并非在所有器件上都可用。详情请参见具体器件数据手册。SPI 帧模式使能 (即 FRMEN = 1) 时不应将该位置 1。

对于没有 MSSEN 位的器件，必须通过 SSx 引脚或其他通用 I/O 在软件的控制下产生从选择信号 (在非帧 SPI 模式下)。

CKP 位 (SPIxCON<6>) 和 CKE 位 (SPIxCON<8>) 决定数据在时钟的哪个边沿发送。

注： 在更改 CKE 或 CKP 位之前，用户必须先关闭 SPI 器件。否则将无法保证器件的行为。

要发送的数据写入 SPIxBUF 寄存器，接收到的数据也是从 SPIxBUF 寄存器读取。

主模式下 SPI 模块的工作原理如下所述：

1. 一旦模块设置为主模式工作并使能，待发送数据就会写入 SPIxBUF 寄存器。SPITBE 位 (SPIxSTAT<3>) 清零。
2. SPIxTXB 的内容移入移位寄存器 SPIxSR (见图 23-8)，且模块将 SPITBE 位清零。
3. 一组 8/16/32 个时钟脉冲将 8/16/32 位发送数据从 SPIxSR 移出到 SDOx 引脚，同时将 SDOx 引脚的数据移入 SPIxSR。

4. 当传输结束时，将发生以下事件：
 - a) 中断标志位 **SPIxRXIF** 置 1。通过将中断允许位 **SPIxRXIE** 置 1 来允许 SPI 中断。**SPIxRXIF** 标志不会被硬件自动清零。
 - b) 此外，当正在进行的发送和接收操作结束时，**SPIxSR** 的内容将移入 **SPIxRXB**。
 - c) 模块将 **SPIRBF** 位 (**SPIxSTAT<0>**) 置 1，指示接收缓冲区已满。一旦用户代码读取了 **SPIxBUF**，硬件就会将 **SPIRBF** 位清零。在增强型缓冲模式下，**SPIRBE** 位 (**SPIxSTAT<5>**) 会在 **SPIxRXB** FIFO 缓冲区全空时置 1，在全空时清零。
5. 当 SPI 模块需要将数据从 **SPIxSR** 传输到 **SPIxRXB** 时，如果 **SPIRBF** 位置 1（接收缓冲区满），模块会将 **SPIROV** 位 (**SPIxSTAT<6>**) 置 1，指示产生了溢出条件。
6. 只要 **SPITBE** 位 (**SPIxSTAT<3>**) 置 1，用户软件就可以在任何时候将待发送数据写入 **SPIxBUF**。写操作可以与 **SPIxSR** 移出先前写入的数据同时发生，因此可以允许连续发送。在增强型缓冲模式下，**SPITBF** 位 (**SPIxSTAT<1>**) 会在 **SPIxTXB** FIFO 缓冲区全满时置 1，在全满时清零。

注： 用户不能直接写入 **SPIxSR** 寄存器。对 **SPIxSR** 寄存器的所有写操作均通过 **SPIxBUF** 寄存器执行。

例 23-1: 16 位 SPI 主模式的初始化代码

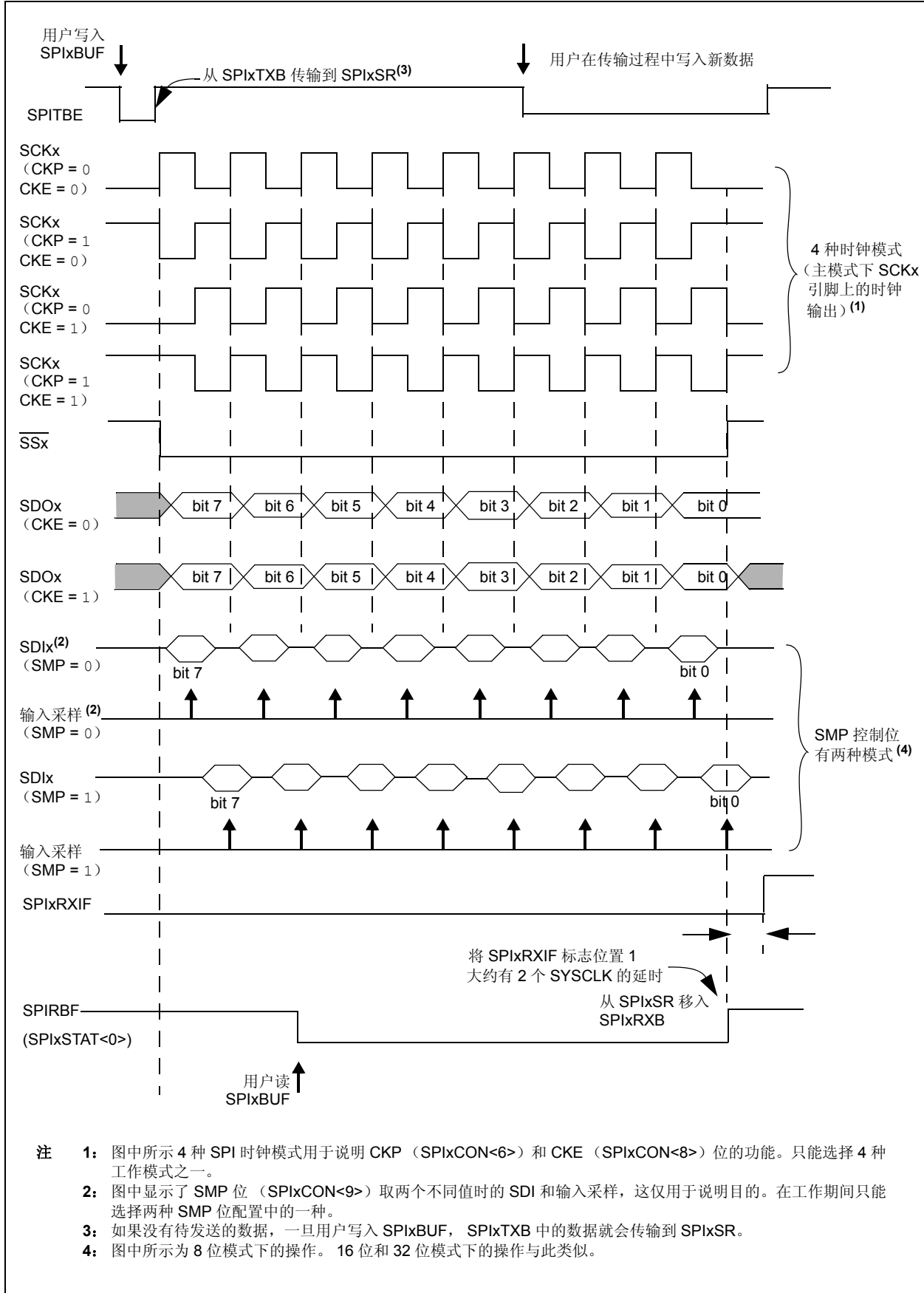
```
/*
The following code example will initialize the SPI1 in Master mode.
It assumes that none of the SPI1 input pins are shared with an analog input. If so, the
AD1PCFG and corresponding TRIS registers have to be properly configured.
*/
int rData;

IEC0CLR=0x03800000;           // disable all interrupts
SPI1CON = 0;                   // Stops and resets the SPI1.
rData=SPI1BUF;                 // clears the receive buffer
IFS0CLR=0x03800000;           // clear any existing event
IPC5CLR=0x1f000000;           // clear the priority
IPC5SET=0x0d000000;           // Set IPL=3, Subpriority 1
IEC0SET=0x03800000;           // Enable RX, TX and Error interrupts

SPI1BRG=0x1;                   // use Fpb/4 clock frequency
SPI1STATCLR=0x40;              // clear the Overflow
SPI1CON=0x8220;                // SPI ON, 8 bits transfer, SMP=1, Master mode

                                // from now on, the device is ready to transmit and receive
                                data
SPI1BUF='A';                    // transmit an A character
```

图 23-9: 8 位模式下的 SPI 主模式工作 (MODE32 = 0, MODE16 = 0)



23.3.3.2 从模式工作

要将 SPI 模块设置为从工作模式，请执行以下步骤：

1. 如果使用中断，禁止相应 IECx 寄存器中的 SPI 中断。
2. 通过清零 ON 位，停止并复位 SPI 模块。
3. 清除接收缓冲区。
4. 如果使用标准缓冲模式，则将 ENHBUF 位 (SPIxCON<16>) 清零，如果使用增强型缓冲模式，则将它置 1。
5. 如果要使用中断，则需要执行以下附加步骤：
 - a) 清零相应 IFSx 寄存器中的 SPIx 中断标志 / 事件。
 - b) 在相应的 IPCx 寄存器中写入 SPIx 中断优先级位和子优先级位。
 - c) 将相应 IECx 寄存器中的 SPIx 中断允许位置 1。
6. 清零 SPIROV 位 (SPIxSTAT<6>)。
7. 将所需的设置写入 SPIxCON 寄存器，且 MSTEN (SPIxCON<5>) = 0。
8. 通过将 ON 位 (SPIxCON<15>) 置 1 使能 SPI 工作。
9. 发送 (及接收) 会在主器件提供串行时钟时立即开始。

注： 在将模式从主模式更改为从模式之前，必须先关闭 SPI 模式。

在从模式下，当 SCKx 引脚上出现外部时钟脉冲时开始发送和接收数据。CKP 位 (SPIxCON<6>) 和 CKE 位 (SPIxCON<8>) 决定数据在时钟的哪个边沿发送。

要发送的数据写入 SPIxBUF 寄存器，接收到的数据也是从 SPIxBUF 寄存器读取。

从模式下模块的其余操作与上述主模式下的操作相同 (包括增强型缓冲模式)。

23.3.3.2.1 从模式附加特性

从模式下提供了以下附加特性：

- 从选择同步

$\overline{\text{SSx}}$ 引脚允许同步从模式。如果 SSEN 位 (SPIxCON<7>) 置 1，则仅当 $\overline{\text{SSx}}$ 引脚被驱动为低电平时，才能使能从模式下的发送和接收。要使 $\overline{\text{SSx}}$ 引脚用作输入，就不能驱动端口输出或其他外设输出。如果 SSEN 位置 1 且 $\overline{\text{SSx}}$ 引脚被驱动为高电平，SDOx 引脚将不再被驱动并且将变为三态，即使模块在发送过程中也是如此。下次 $\overline{\text{SSx}}$ 引脚被驱动为低电平时，将使用 SPIxTXB 寄存器中保存的数据重新尝试进行被中止的发送。如果 SSEN 位未置 1，则 $\overline{\text{SSx}}$ 引脚不影响从模式下模块的工作。

- SPITBE 状态标志的操作

SPITBE 位 (SPIxSTAT<3>) 的功能在从工作模式下有所不同。以下介绍了各种从工作模式设置下的 SPITBE 功能：

- 如果 SSEN (SPIxCON<7>) 清零，当用户代码将数据装入 SPIxBUF 时，SPITBE 清零。它在模块将 SPIxTXB 中的数据传送到 SPIxSR 时置 1。这与主模式下 SPITBE 位的功能类似。
- 如果 SSEN 置 1，当用户代码将数据装入 SPIxBUF 时，SPITBE 清零。但是，它只有在 SPIx 模块完成数据发送时才被置 1。当 $\overline{\text{SSx}}$ 引脚变为高电平时，发送将被中止，但是可能在稍后重新尝试发送。因此，每个数据字都保存在 SPIxTXB 中，直到所有位都被发送到接收器中。

注： 工作于帧模式时，不能使用从选择信号。

例 23-2: 16 位 SPI 从模式的初始化代码

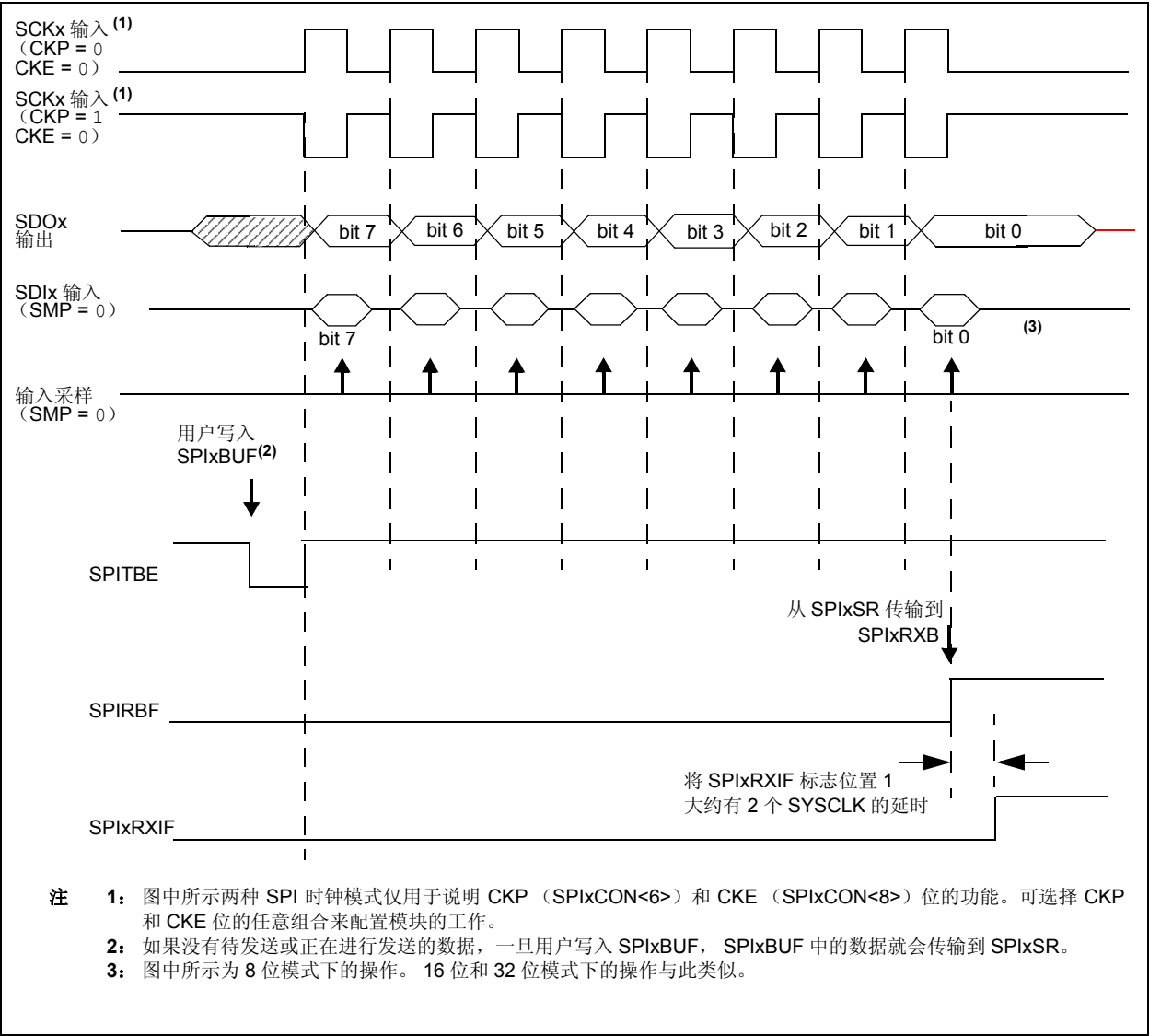
```
/*
The following code example will initialize the SPI1 in Slave mode.
It assumes that none of the SPI1 input pins are shared with an analog input.If so, the
AD1PCFG and corresponding TRIS registers have to be properly configured.
*/
int    rData;

IEC0CLR=0x03800000;      // disable all interrupts
SPI1CON = 0;              // Stops and resets the SPI1.
rData=SPI1BUF;            // clears the receive buffer
IFS0CLR=0x03800000;      // clear any existing event
IPC5CLR=0x1f000000;      // clear the priority
IPC5SET=0x0d000000;      // Set IPL=3, Subpriority 1
IEC0SET=0x03800000;      // Enable RX, TX and Error interrupts

SPI1STATCLR=0x40;        // clear the Overflow
SPI1CON=0x8000;          // SPI ON, 8 bits transfer, Slave mode

// from now on, the device is ready to receive and transmit data
```

图 23-10: 8 位模式下的 SPI 从模式工作（禁止从选择引脚）（MODE32 = 0, MODE16 = 0, SSEN = 0）



23.3.4 SPI 错误处理

如果一个新的数据字已经移入移位寄存器 $SPIxSR$ ，但是接收寄存器 $SPIxRXB$ 的先前内容尚未被用户软件读取，则 $SPIROV$ 位 ($SPIxSTAT<6>$) 将被置 1。此时模块不会将接收到的数据从 $SPIxSR$ 传输到 $SPIxRXB$ 中。禁止后续的数据接收，直到 $SPIROV$ 位被清零。模块不会自动清零 $SPIROV$ 位，它必须由用户软件清零。

23.3.5 SPI 只接收操作

将 $DISSDO$ 控制位 ($SPIxCON<12>$) 置 1 会禁止 $SDOx$ 引脚上的发送。这允许将 $SPIx$ 模块配置为只接收工作模式。如果 $DISSDO$ 位置 1， $SDOx$ 引脚将由对应的端口功能控制。

$DISSDO$ 功能可应用于所有 SPI 工作模式。

23.3.6 帧 SPI 模式

当工作于主模式或从模式时，模块支持一个基本的帧 SPI 协议。SPI 模块中提供了以下特性来支持帧 SPI 模式：

- $FRMEN$ 控制位 ($SPIxCON<31>$) 用于使能帧 SPI 模式并使 \overline{SSx} 引脚用作帧同步脉冲输入或输出引脚。 $SSEN$ ($SPIxCON<7>$) 的状态被忽略。
- $FRMSYNC$ 控制位 ($SPIxCON<30>$) 决定 \overline{SSx} 引脚是输入还是输出（即，模块是接收还是产生帧同步脉冲）。
- $FRMPOL$ 控制位 ($SPIxCON<29>$) 决定单个 SPI 时钟周期的帧同步脉冲的极性。
- 可通过将 $FRMSYPW$ 控制位 ($SPIxCON<27>$) 置 1，将帧同步脉冲宽度配置为一个字符宽。

注： $FRMSYPW$ 位并非在所有器件上都可用。详情请参见具体器件数据手册。

- 可通过设置 $FRMCNT<2:0>$ 控制位 ($SPIxCON<26:24>$) 来配置每个帧同步脉冲发送的数据字符数。

SPI 模块支持以下帧 SPI 模式：

- 帧主模式
SPI 模块产生帧同步脉冲并在 \overline{SSx} 引脚上为其他器件提供此脉冲。
- 帧从模式
SPI 模块使用 \overline{SSx} 引脚上接收到的帧同步脉冲。

可同时支持帧 SPI 模式和主 / 从模式。因此，可以使用以下帧 SPI 配置：

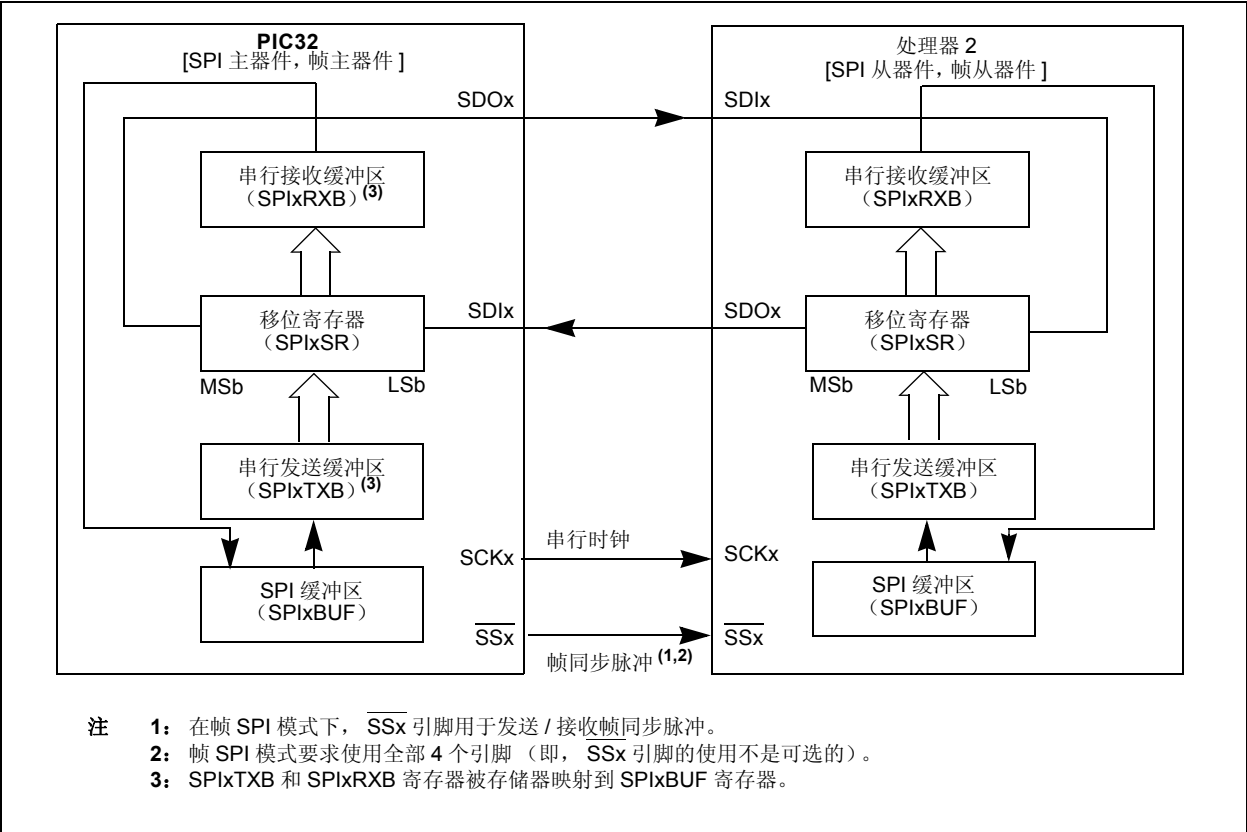
- SPI 主模式和帧主模式
- SPI 主模式和帧从模式
- SPI 从模式和帧主模式
- SPI 从模式和帧从模式

这 4 种模式决定 $SPIx$ 模块是否产生串行时钟和帧同步脉冲。

在帧 SPI 模式下，可通过配置 $ENHBUF$ 位 ($SPIxCON<16>$) 来使用标准缓冲模式或增强型缓冲模式。

此外，在帧 SPI 模式下可将 SPI 模块用于连接外部音频 DAC/ADC 和编解码器器件。

图 23-12: SPI 主器件—帧主器件连接图



23.3.6.1 帧 SPI 模式下的 SCKx

当 FRMEN (SPIxCON<31>) = 1 且 MSTEN (SPIxCON<5>) = 1 时，SCKx 引脚变为输出引脚，且 SCKx 上的 SPI 时钟变为自由运行的时钟。

当 FRMEN = 1 且 MSTEN = 0 时，SCKx 引脚变为输入引脚。假设提供给 SCKx 引脚的源时钟是自由运行的时钟。

时钟的极性由 CKP 位 (SPIxCON<6>) 选择。在帧 SPI 模式下不使用 CKE 位 (SPIxCON<8>)。

当 CKP xor CKE = 0 时，帧同步脉冲输出和 SDOx 数据输出在 SCKx 引脚的时钟脉冲上升沿改变。在串行时钟的下降沿，在 SDIx 输入引脚上采样输入数据。

当 CKP xor CKE = 1 时，帧同步脉冲输出和 SDOx 数据输出在 SCKx 引脚的时钟脉冲下降沿改变。在串行时钟的上升沿，在 SDIx 输入引脚上采样输入数据。

23.3.6.2 帧 SPI 模式下的 SPI 缓冲区

当 FRMSYNC (SPIxCON<30>) = 0 时，SPIx 模块处于帧主模式工作。在该模式下，当用户软件将发送数据写入 SPIxBUF 存储单元（从而将发送数据写入 SPIxTXB 寄存器）时，模块发出帧同步脉冲。在帧同步脉冲的末尾，SPIxTXB 中的数据被传输到 SPIxSR，同时开始发送/接收数据。

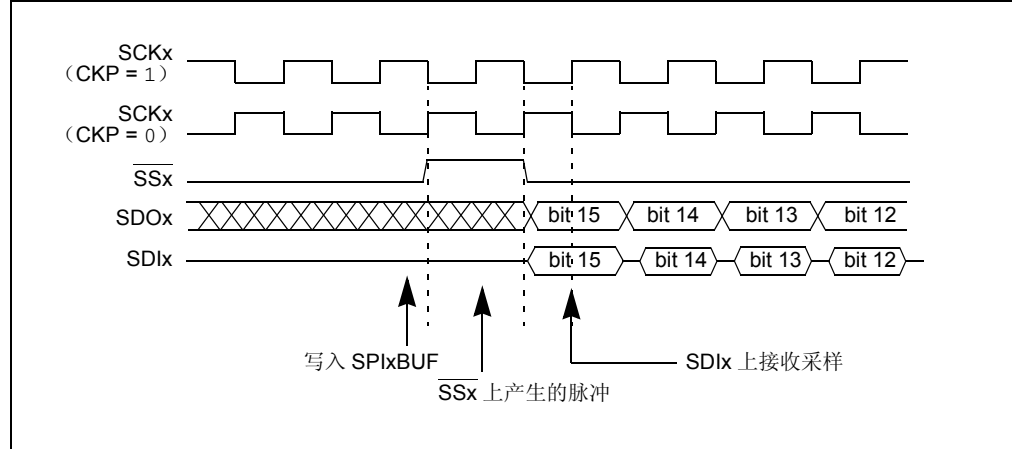
当 FRMSYNC = 1 时，模块处于帧从模式。在该模式下，帧同步脉冲由外部时钟源产生。当模块采样到帧同步脉冲时，它会将 SPIxTXB 寄存器的内容传输到 SPIxSR，同时开始发送/接收数据。在接收到帧同步脉冲之前，用户必须确保 SPIxBUF 中装入了正确的发送数据。

注： 无论数据是否写入 SPIxBUF，接收到帧同步脉冲时都将启动发送。如果未执行写操作，则会发送 0。

23.3.6.3 SPI 主模式和帧主模式

通过将 MSTEN 位 (SPIxCON<5>) 和 FRMEN 位 (SPIxCON<31>) 设置为 1, 并将 FRMSYNC 位 (SPIxCON<30>) 设置为 0 使能该帧 SPI 模式。在该模式下, 无论模块是否正在进行发送, 都将在 SCKx 引脚上连续输出串行时钟。当写入 SPIxBUF 时, SSx 引脚将在 SCKx 时钟的下一个发送边沿驱动为有效 (高电平或低电平, 取决于 FRMPOL 位 (SPIxCON<29>))。SSx 引脚将为高电平, 保持一个 SCKx 时钟周期。模块在 SCKx 的下一个发送边沿开始发送数据, 如图 23-13 所示。图 23-12 给出了指示该工作模式信号方向的连接图。

图 23-13: SPI 主模式和帧主模式 (MODE32 = 0, MODE16 = 1, SPIFE = 0, FRMPOL = 1)



23.3.6.4 SPI 主模式和帧从模式

通过将 MSTEN 位 (SPIxCON<5>)、FRMEN 位 (SPIxCON<31>) 和 FRMSYNC 位 (SPIxCON<30>) 设置为 1 使能该帧 SPI 模式。SSx 引脚为输入引脚, 并且在 SPI 时钟的采样边沿对其进行采样。当采样到它为有效时 (高电平或低电平, 取决于 FRMPOL 位 (SPIxCON<29>)), 数据将在 SPI 时钟的后续发送边沿发送, 如图 23-14 所示。当发送完成时, 中断标志 SPIxIF 置 1。在 SSx 引脚上接收到信号之前, 用户必须确保 SPIxBUF 中装入了正确的发送数据。图 23-15 给出了指示该工作模式信号方向的连接图。

图 23-14: SPI 主模式和帧从模式 (MODE32 = 0, MODE16 = 1, SPIFE = 0, FRMPOL = 1)

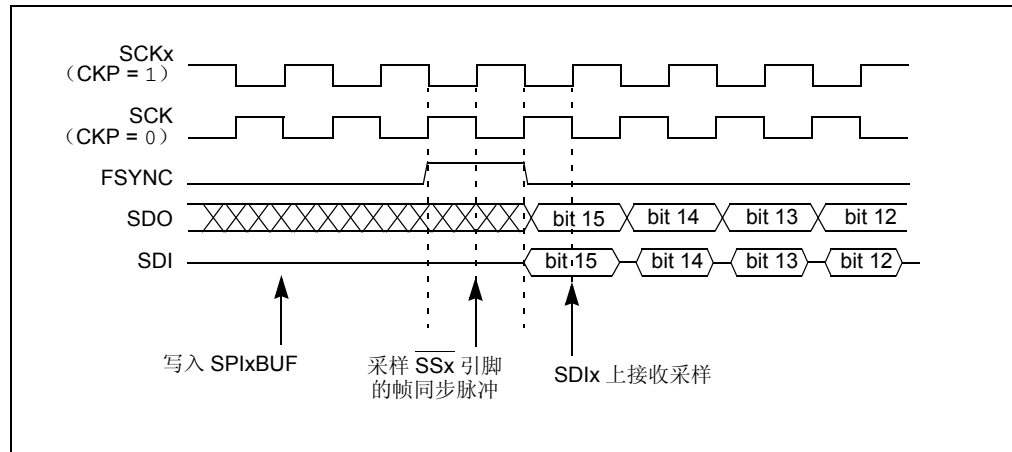
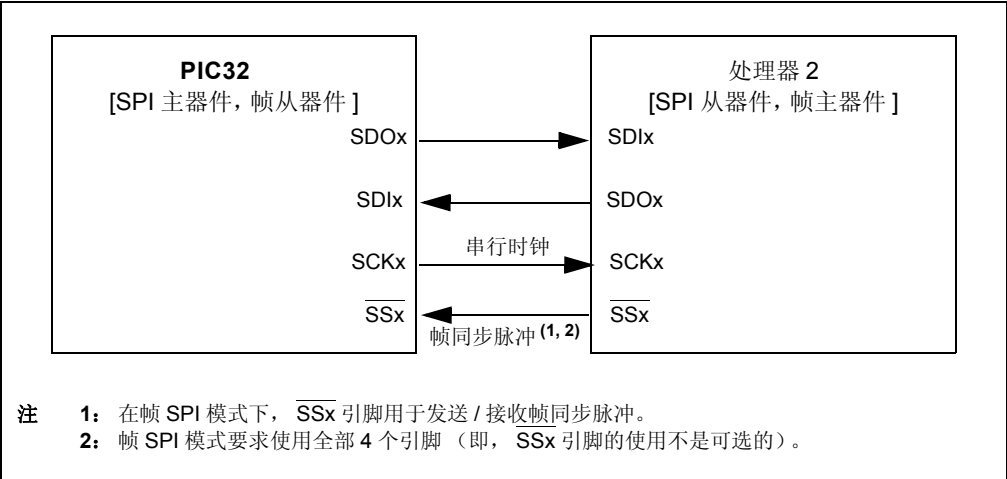


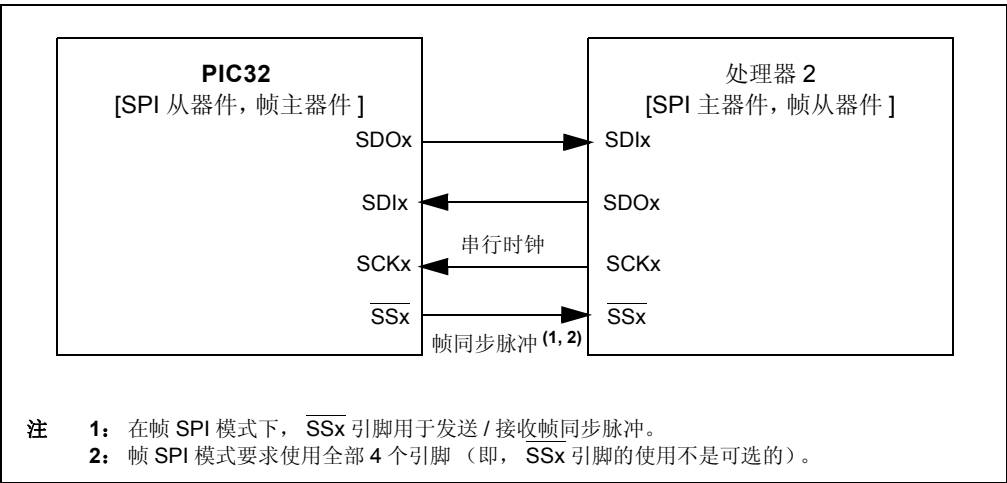
图 23-15: SPI 主器件—帧从器件连接图



23.3.6.5 SPI 从模式和帧主模式

通过将 **MSTEN** 位 (**SPIxCON<5>**) 设置为 0, 将 **FRMEN** 位 (**SPIxCON<31>**) 设置为 1, 并将 **FRMSYNC** 位 (**SPIxCON<30>**) 设置为 0 以使能该帧 SPI 模式。在从模式下, 输入 SPI 时钟是连续的。当 **FRMSYNC** 位为低电平时, \overline{SSx} 引脚是输出引脚。因此, 当写入 **SPIBUF** 时, 模块会在 SPI 时钟的下一个发送边沿将 \overline{SSx} 引脚驱动为有效 (高电平或低电平, 取决于 **FRMPOL** 位 (**SPIxCON<29>**))。 \overline{SSx} 引脚将驱动为高电平, 保持一个 SPI 时钟周期。在下一个 SPI 时钟发送边沿开始发送数据。图 23-16 给出了指示该工作模式信号方向的连接图。

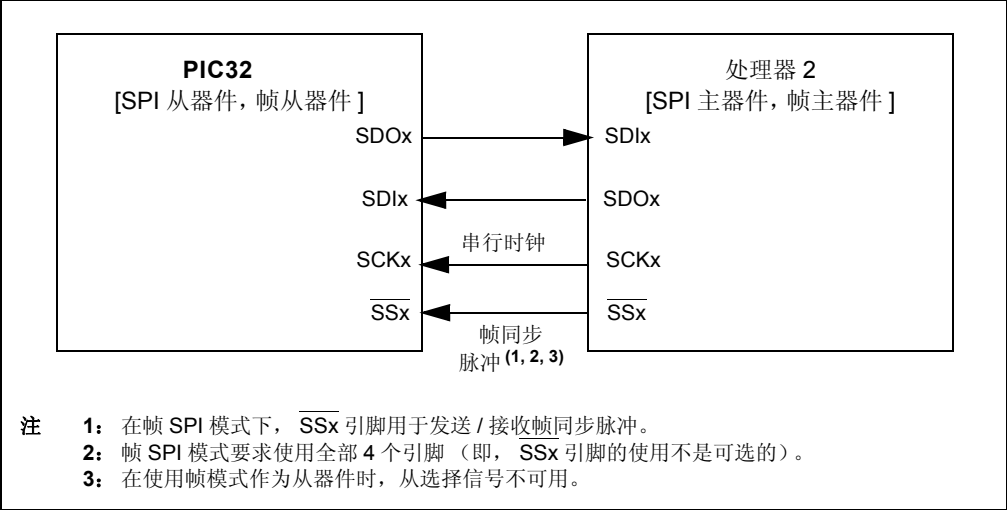
图 23-16: SPI 从器件—帧主器件连接图



23.3.6.6 SPI 从模式和帧从模式

通过将 MSTEN 位 (SPIxCON<5>) 设置为 0, 将 FRMEN 位 (SPIxCON<31>) 设置为 1, 并将 FRMSYNC 位 (SPIxCON<30>) 设置为 1 使能该帧 SPI 模式。因此, SCKx 和 SSx 引脚都是输入引脚。在 SPI 时钟的采样边沿对 SSx 引脚进行采样。当采样到 SSx 有效时 (高电平或低电平, 取决于 FRMPOL (SPIxCON<29>) 位), 数据将在 SCKx 的下一个发送边沿发送。图 23-17 给出了指示该工作模式信号方向的连接图。

图 23-17: SPI 从器件—帧从器件连接图



23.3.7 SPI 主模式时钟频率

SPI 模块通过 9 位 SPIxBRG 寄存器来支持灵活地产生波特率。SPIxBRG 是可读写寄存器，用于决定波特率。为 SPI 模块提供的外设时钟 PBCLK 是 CPU 内核时钟的分频函数。该时钟根据装入 SPIxBRG 的值进行分频。通过 PBCLK 分频获得的 SCKx 时钟的占空比为 50%，它通过 SCKx 引脚提供给外部器件。

注： SCKx 时钟在非帧 SPI 模式下不是自由运行的。它仅在 SPIxBUF 中装入数据时运行 8、16 或 32 个脉冲时间。但在帧模式下，它会连续运行。

公式 23-1 将 SCKx 时钟频率定义为 SPIxBRG 设置的函数。

公式 23-1:

$$F_{SCK} = \frac{F_{PB}}{2 \cdot (SPIxBRG + 1)}$$

因此，可能的最大波特率为 F_{PB}/2（SPIxBRG = 0），可能的最小波特率为 F_{PB}/1024。

表 23-4 列出了部分 SPI 时钟频率示例（单位为 kHz）：

表 23-4: SCKx 频率示例

SPIxBRG 设置	0	15	31	63	85	127	255	511
FPB = 80 MHz	40.00 MHz	2.5 MHz	1.25 kHz	625 kHz	465.11 kHz	312.5 kHz	156.25 kHz	78.13 kHz
FPB = 72 MHz	36.00 MHz	2.25 MHz	1.13 kHz	562.5 kHz	418.60 kHz	281.25 kHz	140.63 kHz	70.31 kHz
FPB = 60 MHz	30.00 MHz	1.88 MHz	937.5 kHz	468.75 kHz	348.83 kHz	234.38 kHz	117.19 kHz	58.59 kHz
FPB = 50 MHz	25.00 MHz	1.56 MHz	781.25 kHz	390.63 kHz	290.7 kHz	195.31 kHz	97.66 kHz	48.83 kHz
FPB = 40 MHz	20.00 MHz	1.25 MHz	625.00 kHz	312.50 kHz	232.56 kHz	156.25 kHz	78.13 kHz	39.06 kHz
FPB = 25 MHz	12.50 MHz	781.25 kHz	390.63 kHz	195.31 kHz	145.35 kHz	97.66 kHz	48.83 kHz	24.41 kHz
FPB = 20 MHz	10.00 MHz	625.00 kHz	312.50 kHz	156.25 kHz	116.28 kHz	78.13 kHz	39.06 kHz	19.53 kHz
FPB = 10 MHz	5.00 MHz	312.50 kHz	156.25 kHz	78.13 kHz	58.14 kHz	39.06 kHz	19.53 kHz	9.77 kHz

注： 并不支持所有时钟频率。更多信息，请参见具体器件数据手册的“电气特性”章节中的 SPI 时序规范。

23.4 音频协议接口模式

SPI 模块可以与目前大多数的编解码器器件连接以提供基于 PIC32 单片机的音频解决方案。SPI 模块通过四个标准 I/O 引脚提供对音频协议功能的支持。构成音频协议接口模式的四个引脚为：

- SDIx: 用于接收采样数字音频数据的串行数据输入 (ADC DAT)
- SDOx: 用于发送数字音频数据的串行数据输出 (DAC DAT)
- SCKx: 串行时钟, 也称为位时钟 (BCLK)
- SSx: 左 / 右通道时钟 (LRCK)

BCLK 提供将数据驱动出或驱动入模块所需的时钟, 而 LRCK 根据所选协议模式提供帧同步。

在某些编解码器中, 串行时钟 (SCK) 指的是波特 / 位时钟 (BCLK)。在本节中, 为了符合编解码器命名约定, 将信号 SSx 称为 LRCK。SPI 模块能够在音频协议主模式和音频协议从模式下工作。在主模式下, 该模块在 SCKx 引脚上生成 BCLK, 在 SSx 引脚上生成 LRCK。在某些器件中, 当处于从模式时, 模块从工作在主模式的 I²S 伙伴处接收这两个时钟。

当处于主模式时, SPI 模块能够通过来自各种内部源 (如基准时钟、PBCLK、USB 时钟、FRC 和其他内部源) 的主时钟 (MCLK) 在内部生成自己的时钟。此外, SPI 模块还能为编解码器器件提供 MCLK, 这是常见的要求。

要启动音频协议模式, 首先将 ON 位 (SPIxCON<15>) 置 0 来禁止外设。接下来, 将 AUDEN 位 (SPIxCON2<7>) 置 1, 然后将 ON 位置 1 来重新使能外设。

在主模式下配置时, SCK 和 LRCK 的前沿在启动音频协议后的一个 SCK 周期内出现。串行数据以通过 AUDMOD<1:0> 位 (SPIxCON2<1:0>) 设置的协议模式所确定的时序移入或移出。如果发送 FIFO 为空, 则发送零。

在从模式下, 外设在 SDO 上输出零, 但直到出现 LRCK 的前沿才会将发送 FIFO 的内容发送出去, 之后开始接收数据 (如果未禁止 SDI)。只要发送 FIFO 为空, 就将继续发送零。

无论在主模式还是从模式下, SPI 模块在启动后都不会发生 TX FIFO 数据不足的情况。这样软件就可以设置 SPI、设置 DMA、开启 SPI 音频协议, 然后无错开启 DMA。

在第一次写入 TX FIFO (SPIxBUF) 之后, SPI 将使能数据不足检测和生成。要在使能 DMA 之前使 RX FIFO 保持为空, 请将 DISSDI (SPIxCON<4>) 置 1。使能 DMA 之后, 将 DISSDI 置 0 以开始接收。

23.4.1 主模式

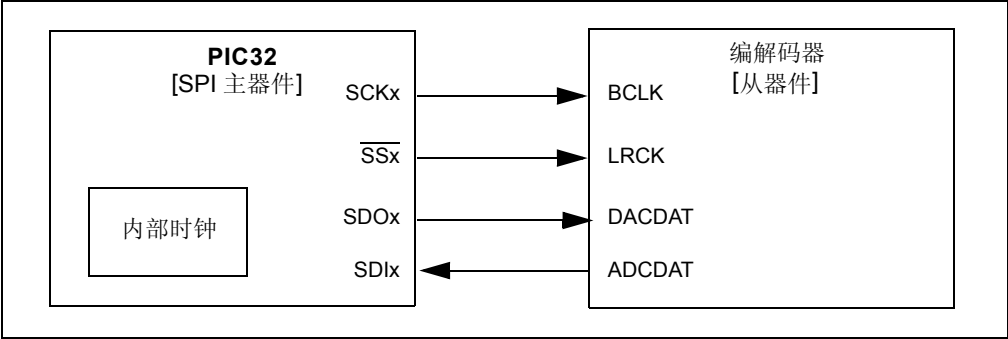
要在音频协议主模式下配置 PIC32 器件, 请将 MSTEN 位 (SPIxCON<5>) 和 AUDEN 位 (SPIxCON2<7>) 置 1。

主模式具有以下特性:

- 只要 ON 位 (SPIxCON<15>) = 1, 此模式就会使能器件生成 SCK 和 LRCK 脉冲
- 在主模式下, SPI 模块在任何情况下都将连续生成 LRCK 和 SCK, 无论是否发送数据
- SPI 模块在一个 SCK 周期内输出 LRCK 和 SCK 的前沿, 并且串行数据连续移入和移出, 即使 TX FIFO 为空时也是如此

图 23-18 显示了主模式下主器件和从器件之间的典型接口。

图 23-18: 主器件自行生成时钟——输出 BCLK 和 LRCK



23.4.2 从模式

通过将 MSTEN 位 (SPIxCON<5>) 置 0 和将 AUDEN 位 (SPIxCON2<7>) 置 1, 可在音频协议从模式下配置 SPI 模块

从模式具有以下特性:

- 只要 ON 位 (SPIxCON<15>) = 1, 此模式就会使能器件接收 SCK 和 LRCK 脉冲
- SPI 模块在 SDO 上输出零, 但直到模块接收到 LRCK (即左通道之前的边沿) 才会移出或移入数据 (SDI)
- 模块接收到 LRCK 的前沿之后, 如果 DISSDI (SPIxCON<4>) = 0, 则开始接收数据, 并且串行数据将连续移出, 即使 TX FIFO 为空时也是如此

图 23-19 显示了处于音频从接口模式的 SPI 模块与编解码器主器件之间的接口。

图 23-19: 作为主器件的编解码器器件通过外部晶振生成所需时钟

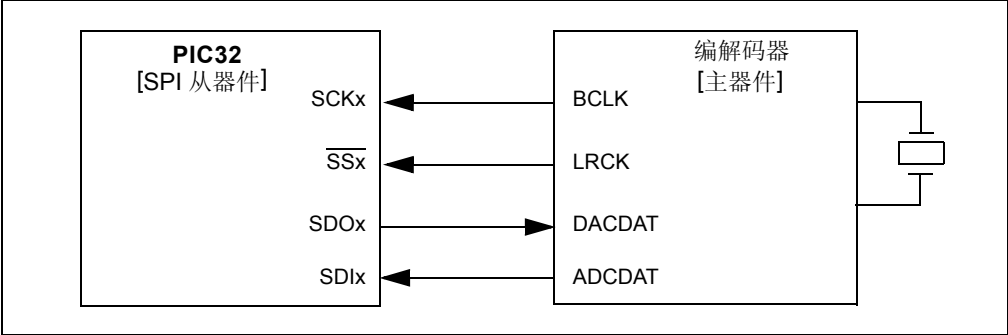
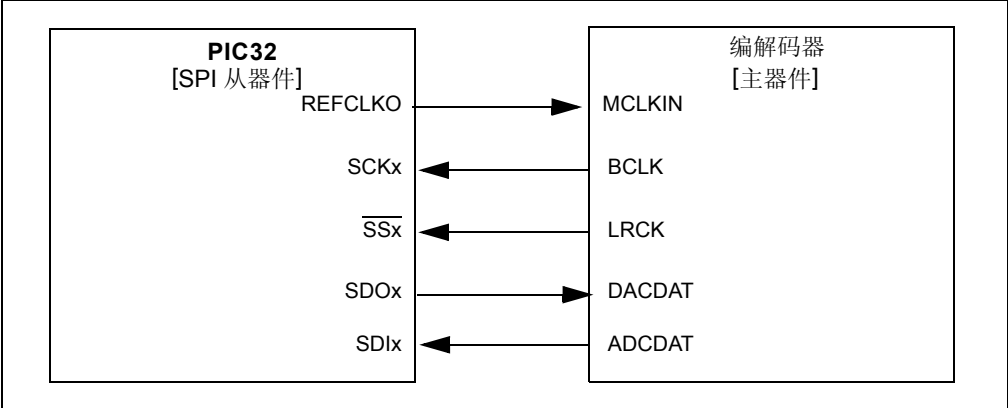


图 23-20 显示了处于音频从接口模式的 SPI 模块与编解码器主器件之间的接口, 其中主时钟来自 SPI 参考时钟输出功能。

图 23-20: 作为主器件的编解码器器件通过 PIC32 参考时钟输出得出 MCLK



23.4.3 音频数据长度和帧长度

尽管编解码器器件可生成字长度为 8/16/20/24/32 的音频数据采样，但 PIC32 SPI 模块只支持发送 / 接收长度为 16、24 和 32 的音频数据。

注： 实际采样数据可以为任意长度，最多 32 位，数据必须打包成 16/24/32 等三种格式之一。

表 23-5 说明了 MODE<32,16> 位（SPIxCON<11:10>）如何控制最大允许采样长度和帧长度（SSx 上的 LRCK 周期）。

表 23-5: 音频数据长度与 LRCK 周期

SPIxCON<11:10>		数据长度 (位)	FIFO 宽度 (位)	左 / 右通道 采样长度 (位)	增强型缓冲区 FIFO 深度 (采样)	LRCK 周期 帧长度 (位)
MODE32	MODE16					
0	0	16	16	≤ 16	8	32
0	1	16	16	≤ 32	8	64
1	1	24	32	≤ 32	4	64
1	0	32	32	≤ 32	4	64

MODE<32,16> 位（SPIxCON<11:10>）的参数具有以下特性：

- 控制左 / 右通道数据长度和帧长度
- 在 16 位采样模式下，支持 32/64 位帧长度
- 在 24/32 位采样模式下，支持 64 位帧长度
- 定义 FIFO 宽度和深度（例如，24 位数据具有 32 位宽和 4 个存储单元深的 FIFO）
- 如果写入数据的长度大于所选数据的长度，则忽略高字节
- 如果写入数据的长度小于所选数据的长度，则在写入所选长度的最高字节（Most Significant Byte, MSB）时，FIFO 指针将发生变化

如果分多次将此数据写入发送 FIFO，则写入顺序必须是从最低字节到最高字节。

例如，假设音频数据为每个采样 24 位，每次 8 位。根据表 23-5，FIFO 宽度为每个采样 32 位。因此，每个 FIFO 采样中的 8 个最高有效位（Most Significant bit, MSb），即位 31:24 将被忽略。

位 15:8 和 7:0 可按任意顺序写入 SPIxBUF 寄存器；但是，位 23:16 必须最后写入，因为写入最高有效位（MSb）位 24 时将触发发送缓冲区的指针发生变化。

写入未使用字节的数据将被忽略。同样，仅涉及未使用字节的事务也将被忽略。因此，对地址偏移 0x0023 的字节写操作将被完全忽略，并且如果数据宽度小于 32 位，则不会造成 FIFO 进栈。

23.4.4 帧错误 /LRCK 错误

SPI 模块为调试提供帧 /LRCK 错误检测。如果用于定义通道开始的 LRCK 边沿出现在正确的位数（由 MODE<32,16> 定义）之前，将会发生帧 /LRCK 错误。

SPI 模块会立即将 FRMERR 位（SPIxSTAT<12>）置 1，将数据从 SPIxSR 寄存器推入到 SPIxRXB 寄存器中，并将数据从 SPIxTXB 寄存器弹出到 SPIxSR 寄存器中。通过将 FRMERREN 位（SPIxCON2<12>）置 1 可将该模块配置为检测帧 /LRCK 相关的错误。

注： 在音频协议模式下，BCLK（SCKx 引脚上）和 LRCK（SSx 引脚上）是自由运行的，这意味着它们是连续的。通常，LRCK 的长度是固定数目的 BCLK。在所有情况下，SPI 模块都将与新帧的边沿重新对齐，并将 FRMERR 位置 1。如果在非 PCM 模式下工作，当帧过短时，SPI 模块还会将缩简的数据推入到 FIFO 中。

23.4.5 音频协议模式

SPI 模块支持四种音频协议模式，可工作在下列任何一种模式：

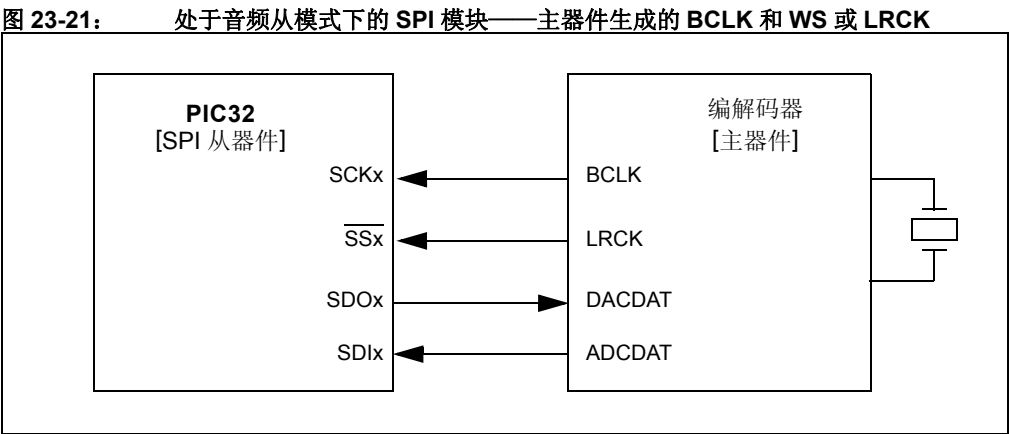
- I²S 模式（并非所有器件都可用；请参见具体器件的数据手册以查看是否可用）
- 左对齐模式
- 右对齐模式
- PCM/DSP 模式

可通过配置 AUDMOD<1:0> 位（SPIxCON2<1:0>）使能这些音频协议模式。这些模式使能与不同类型的编解码器之间的通信并控制 LRCK 和 SDI/SDO 相对于 SCK 的边沿关系。

至于数据发送，在所有协议模式中，都最先发送 MSB，然后发送 MSB-1，以此类推，直到发送完最低字节（Least Significant Byte, LSB）。第 23.4.3 节“音频数据长度和帧长度”中讨论了数据长度。如果在发送 LSB 之后存在剩余的 SCK 周期，则发送零填充帧。

与主模式相比，当处于从模式时，BCLK（SCKx 引脚上）与 LRCK（SSx 引脚上）的周期（或帧长度）之间的关系所受到的限制要小得多。在主模式下，帧长度等于 32 或 64 个 BCLK，具体取决于 MODE<32,16> 位（SPIxCON<11:10>）设置。然而在从模式下，帧长度可以大于或等于 32 或 64 个 BCLK，但如果帧 LRCK 的边沿提前到达，FRMERR 位（SPIxSTAT<12>）将会置 1。

图 23-21 说明了编解码器器件和处于音频模式下的 SPI 模块之间的一般接口。



23.4.5.1 I²S 模式

注： 并非所有器件均提供此功能。请参见具体器件的数据手册以查看是否提供此功能。

I²S 协议允许通过单个串行接口传输两个通道的数字音频数据。I²S 协议定义了通过 WS/LRCK 线处理立体声数据的 3 线接口。I²S 规范定义了支持发送或接收的半双工接口，但不能同时发送和接收。当 SDO 和 SDI 都可用时，此外设支持全双工操作，如图 23-22 所示。

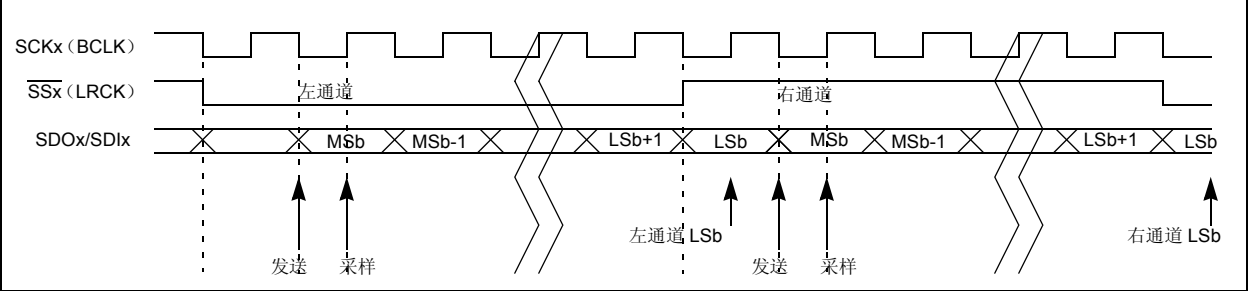
- 数据发送和时钟：
 - 在 LRCK 转变后，发送器在 SCK 的第一个下降沿对音频采样数据的 MSb 进行移位
 - 接收器在 SCK 的第二个上升沿采样 MSb
 - 当 LRCK 为低电平时，左通道数据移出；当 LRCK 为高电平时，右通道数据移出
 - 左通道和右通道中的数据由单个帧组成
- 所需的配置设置：

要将模块设置为 I²S 模式，必须设置以下位：

 - AUDMOD<1:0> (SPIxCON2<1:0>) = 00
 - FRMPOL (SPIxCON<29>) = 0
 - CKP (SPIxCON<6>) = 1

设置这些位将使 SDO 和 LRCK (\overline{SSx}) 在 SCK (BCLK) 的下降沿发生转变，并使 SDI 在 SCK 的上升沿进行采样。请参见图 23-22。

图 23-22: 处理 16 位数据 / 通道或 32 位数据 / 通道的 I²S



23.4.5.1.1 I²S 音频从工作模式

通过以下步骤将 SPI 模块设置为 I²S 音频从工作模式：

1. 如果使用中断，则禁止相应 IECx 寄存器中的 SPI 中断。
2. 通过清零 ON 位 (SPIxCON<15>) 停止并复位 SPI 模块。
3. 复位 SPI 音频配置寄存器 SPIxCON2。
4. 清除接收缓冲区。
5. 如果使用标准缓冲模式，则将 ENHBUF 位 (SPIxCON<16>) 清零，如果使用增强型缓冲模式，则将该位置 1。
6. 如果使用中断，则执行以下附加步骤：
 - a) 清除相应 IFSx 寄存器中的 SPIx 中断标志 / 事件。
 - b) 写入相应 IPCx 寄存器中的 SPIx 中断优先级位和子优先级位。
 - c) 将相应 IECx 寄存器中的 SPIx 中断使能位置 1。
7. 清零 SPIROV 位 (SPIxSTAT<6>)。
8. 将所需设置写入 SPIxCON2 寄存器。
 - a) AUDMOD<1:0> 位 (SPIxCON2<1:0>) = 00
 - b) AUDEN 位 (SPIxCON2<7>) = 1
9. 将所需设置写入 SPIxCON 寄存器：
 - a) MSTEN (SPIxCON<5>) = 0
 - b) CKP (SPIxCON<6>) = 1
 - c) 对于 16 位音频通道数据，MODE<32,16> (SPIxCON<11:10>) = 0。
 - d) 通过将 ON 位 (SPIxCON<15>) 置 1 使能 SPI 操作。
10. 主器件一提供 BCLK 和 LRCK，就开始发送 (和接收)。

例 23-3: I²S 从模式，16 位通道数据，32 位帧

```

/* The following code example will initialize the SPI1 Module in I2S Slave mode. */
/* It assumes that none of the SPI1 input pins are shared with an analog input. */

unsigned int rData;
IEC0CLR=0x03800000; // disable all interrupts
SPI1CON = 0;        // Stops and resets the SPI1.
SPI1CON2 = 0;       // Reset audio settings
rData=SPI1BUF;      // clears the receive buffer
IFS0CLR=0x03800000; // clear any existing event
IPC5CLR=0x1f000000; // clear the priority
IPC5SET=0x0d000000; // Set IPL = 3, Subpriority 1
IEC0SET=0x03800000; // Enable RX, TX and Error interrupts

SPI1STATCLR=0x40;   // clear the Overflow
SPI1CON2=0x00000080; // I2S Mode, AUDEN = 1, AUDMON = 0
SPI1CON =0x00008040; // Slave mode, SPI ON, CKP = 1, 16-bit audio data, 32 bits per frame
// from here, the device is ready to receive and transmit data

/* Note: A few of bits related to frame settings are not required to be set in the SPI1CON */
/* register during audio mode operation. Please refer to the notes in the SPIxCON2 register.*/

```

23.4.5.1.2 I²S 音频主工作模式

当与编解码器从器件连接时，典型应用可以是播放 PCM 数据（8 kHz 采样率，16 位数据，32 位帧大小）。在这种情况下，SPI 模块初始化为生成 256 kbps 的 BCLK。假设采用 40 MHz 的外设总线时钟， $F_{PB} = 40e6$ ，则使用公式 23-2 确定波特率。

公式 23-2:

$$\text{波特率} = \frac{F_{PB}}{2 \cdot (SPIxBRG + 1)}$$

公式 23-3 给出了求出 $SPIxBRG$ 值的方法。

公式 23-3:

$$SPIxBRG = \frac{F_{PB}}{2 \cdot (\text{波特率})} - 1$$

此时波特率等于 256e3。公式 23-4 给出了计算结果。

公式 23-4:

$$SPIxBRG = \frac{40e6}{2(256e3)} - 1 = 77.125$$

如果将公式 23-4 的结果舍入到最近的整数，则此时 $SPIxBRG$ 等于 77；因此，有效的波特率是公式 23-5 给出的结果。

公式 23-5:

$$\frac{40e6}{2 \cdot (77 + 1)} = \frac{40e6}{156} = 256410.25 \text{ 位/秒}$$

该结果快了 0.16%；但是，这完全在大多数系统的容差范围内（0.16% 等于 1/625）。在某些器件上，可使用 REFOTRIM 寄存器消除此误差，准确地以所需频率提供主时钟输出。

可使用以下步骤设置 SPI 模块，以在 I²S 音频主模式下工作：

1. 如果使用中断，则禁止相应 IECx 寄存器中的 SPI 中断。
2. 通过清零 ON 位（SPIxCON<15>）停止并复位 SPI 模块。
3. 复位 SPI 音频配置寄存器 SPIxCON2。
4. 复位波特率寄存器 SPIxBRG。
5. 清除接收缓冲区。
6. 如果使用标准缓冲模式，则将 ENHBUF 位（SPIxCON<16>）清零，如果使用增强型缓冲模式，则将该位置 1。
7. 如果使用中断，则执行以下附加步骤：
 - a) 清除相应 IFSx 寄存器中的 SPIx 中断标志 / 事件。
 - b) 写入相应 IPCx 寄存器中的 SPIx 中断优先级位和子优先级位。
 - c) 将相应 IECx 寄存器中的 SPIx 中断使能位置 1。
8. 清零 SPIROV 位（SPIxSTAT<6>）。
9. 将所需设置写入 SPIxCON2 寄存器。对于 I²S 模式，必须将 AUDMOD<1:0> 位（SPIxCON2<1:0>）设置为 00，并且必须将 AUDEN 位（SPIxCON2<7>）置 1 才能使能音频协议。
10. 将 SPIxBRG 波特率寄存器设置为 0x4D（以在 PBCLK 为 40 MHz 时产生约 256 kbps 的采样率）。

11. 将所需设置写入 SPIxCON 寄存器:
 - a) MSTEN (SPIxCON<5>) = 1。
 - b) CKP (SPIxCON<6>) = 1。
 - c) 对于 16 位音频通道数据, MODE<32,16> (SPIxCON<11:10>) = 0。
 - d) 通过将 ON 位 (SPIxCON<15>) 置 1 使能 SPI 操作。
12. ON 位置 1 后将立即开始发送 (和接收)。

例 23-4: I²S 主模式, 256 kbps BCLK, 16 位通道数据, 32 位帧

```

/* The following code example will initialize the SPI1 Module in I2S Master mode. */
/* It assumes that none of the SPI1 input pins are shared with an analog input. */

unsigned int rData;
IEC0CLR=0x03800000;    // disable all interrupts
SPI1CON = 0;           // Stops and resets the SPI1.
SPI1CON2 = 0;          // Reset audio settings
SPI1BRG=0;            // Reset Baud rate register
rData=SPI1BUF;         // clears the receive buffer
IFS0CLR=0x03800000;    // clear any existing event
IPC5CLR=0x1f000000;    // clear the priority
IPC5SET=0x0d000000;    // Set IPL = 3, Subpriority 1
IEC0SET=0x03800000;    // Enable RX, TX and Error interrupts

SPI1STATCLR=0x40;      // clear the Overflow
SPI1CON2=0x00000080;   // I2S Mode, AUDEN = 1, AUDMON = 0
SPI1BRG =0x4D;         // (to generate 256 kbps sample rate, PBCLK @ 40 MHz)
SPI1CON =0x00008060;   // Master mode, SPI ON, CKP = 1, 16-bit audio channel
                        // data, 32 bits per frame
// from here, the device is ready to receive and transmit data

/* Note: A few of bits related to frame settings are not required to be set in the SPI1CON */
/* register during audio mode operation. Please refer to the notes in the SPIxCON2 register.*/

```

23.4.5.2 左对齐模式

左对齐模式与 I²S 模式相似；但是，在此模式下，SPI 在第一个 SCK 边沿对音频数据的 MSb 移位，与 LRCK 的转变同时进行。在接收器侧，SPI 模块在下一个 SCK 边沿采样 MSb。

通常，编解码器默认使用对齐协议在 SCK 的上升沿发送数据，在 SCK 的下降沿接收数据。

- 所需的配置设置
要将模块设置为左对齐模式，必须设置以下位
 - AUDMOD<1:0> (SPIxCON2<1:0>) = 01
 - FRMPOL (SPIxCON<29>) = 1
 - CKP (SPIxCON<6>) = 0

这将使 SDO 和 LRCK 在 SCK 的上升沿发生转变。关于 16、24 和 32 位音频数据传输，请参见图 23-23 和图 23-24 中所示的采样波形图。

图 23-23: 16 位数据 / 通道或 32 位数据 / 通道的左对齐

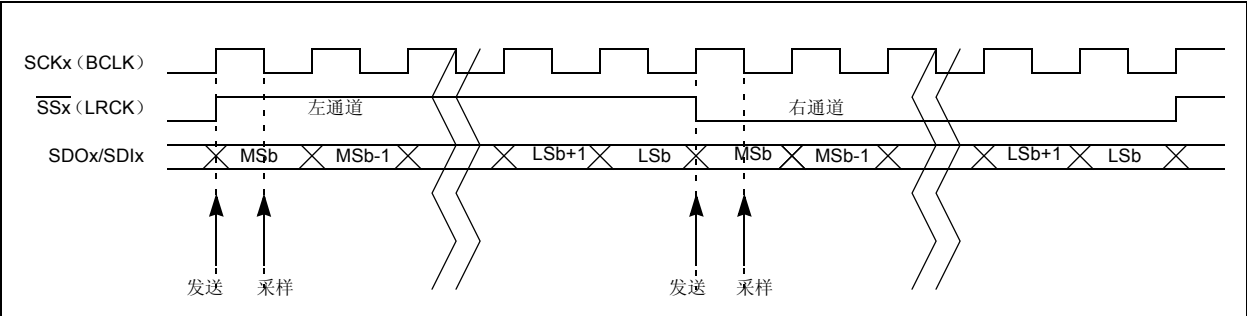
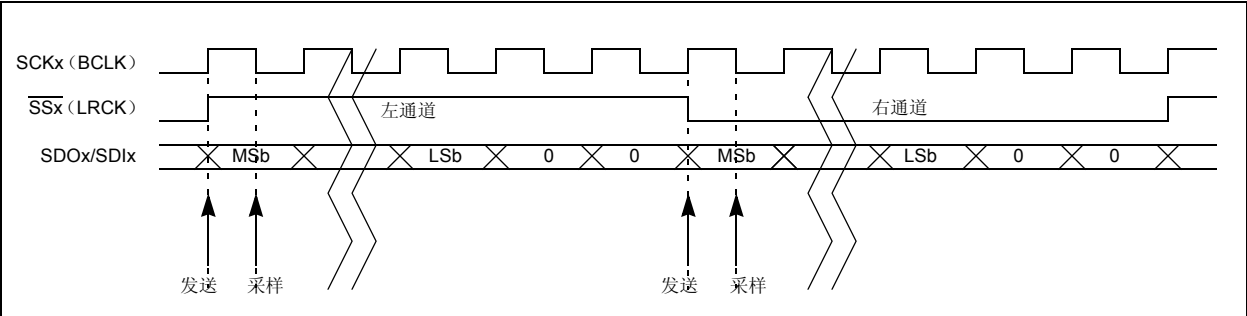


图 23-24: 16/24 位数据和 32 位通道的左对齐



23.4.5.2.1 左对齐音频从工作模式

通过以下步骤将 SPI 模块设置为左对齐音频从工作模式：

1. 如果使用中断，则禁止相应 IECx 寄存器中的 SPI 中断。
2. 通过清零 ON 位 (SPIxCON<15>) 停止并复位 SPI 模块。
3. 复位 SPI 音频配置寄存器 SPIxCON2。
4. 清除接收缓冲区。
5. 如果使用标准缓冲模式，则将 ENHBUF 位 (SPIxCON<16>) 清零，如果使用增强型缓冲模式，则将该位置 1。
6. 如果使用中断，则执行以下附加步骤：
 - a) 清除相应 IFSx 寄存器中的 SPIx 中断标志/事件。
 - b) 写入相应 IPCx 寄存器中的 SPIx 中断优先级位和子优先级位。
 - c) 将相应 IECx 寄存器中的 SPIx 中断使能位置 1。
7. 清零 SPIROV 位 (SPIxSTAT<6>)。
8. 将所需设置写入 SPIxCON2 寄存器。对于左对齐模式，必须将 AUDMOD<1:0> 位 (SPIxCON2<1:0>) 设置为 01，并且必须将 AUDEN 位 (SPIxCON2<7>) 置 1 才能使用音频协议。
9. 将所需设置写入 SPIxCON 寄存器：
 - a) 设置为从模式，MSTEN (SPIxCON<5>) = 0。
 - b) 设置时钟极性，CKP (SPIxCON<6>) = 0。
 - c) 设置帧极性，FRMPOL (SPIxCON<29>) = 1。
 - d) 对于 16 位音频通道数据，设置 MODE<32,16> (SPIxCON<11:10>) = 0
 - e) 通过将 ON 位 (SPIxCON<15>) 置 1 使能 SPI 操作。
10. 主器件一提供 BCLK 和 LRCK，就开始发送 (和接收)。

例 23-5: 左对齐从模式，16 位通道数据，32 位帧

```
/* The following code example will initialize the SPI1 Module in Left-Justified Slave mode. */
/* It assumes that none of the SPI1 input pins are shared with an analog input. */

unsigned int rData;
IEC0CLR=0x03800000; // disable all interrupts
SPI1CON = 0; // Stops and resets the SPI1.
SPI1CON2 = 0; // Reset audio settings
rData=SPI1BUF; // clears the receive buffer
IFS0CLR=0x03800000; // clear any existing event
IPC5CLR=0x1f000000; // clear the priority
IPC5SET=0x0d000000; // Set IPL = 3, Subpriority 1
IEC0SET=0x03800000; // Enable RX, TX and Error interrupts
SPI1STATCLR=0x40; // clear the Overflow
SPI1CON2=0x00000081; // Left-Justified Mode, AUDEN = 1, AUDMON = 0
SPI1CON =0x20008000; // Slave mode, SPI ON, CKP = 0, FRMPOL = 1,
// 16-bit audio data, 32 bits per frame
// from here, the device is ready to receive and transmit data

/* Note: A few of bits related to frame settings are not required to be set in the SPI1CON */
/* register during audio mode operation. Please refer to the notes in the SPIxCON2 register.*/
```

23.4.5.2.2 左对齐音频主工作模式

通过以下步骤将 SPI 模块设置为左对齐音频主工作模式：

1. 如果使用中断，则禁止相应 IECx 寄存器中的 SPI 中断。
2. 通过清零 ON 位（SPIxCON<15>）停止并复位 SPI 模块。
3. 复位 SPI 音频配置寄存器 SPIxCON2。
4. 复位波特率寄存器 SPIxBRG。
5. 清除接收缓冲区。
6. 如果使用标准缓冲模式，则将 ENHBUF 位（SPIxCON<16>）清零，如果使用增强型缓冲模式，则将该位置 1。
7. 如果使用中断，则执行以下附加步骤：
 - a) 清除相应 IFSx 寄存器中的 SPIx 中断标志/事件。
 - b) 写入相应 IPCx 寄存器中的 SPIx 中断优先级位和子优先级位。
 - c) 将相应 IECx 寄存器中的 SPIx 中断使能位置 1。
8. 清零 SPIROV 位（SPIxSTAT<6>）。
9. 将所需设置写入 SPIxCON2 寄存器。对于左对齐模式，必须将 AUDMOD<1:0> 位（SPIxCON2<1:0>）设置为 01，并且必须将 AUDEN 位（SPIxCON2<7>）置 1 才能使能音频协议。
10. 将 SPIxBRG 波特率寄存器设置为 0x4D（以在 PBCLK 为 40 MHz 时产生约 256 kbps 的采样率）。
11. 将所需设置写入 SPIxCON 寄存器：
 - a) 设置为主模式，MSTEN（SPIxCON<5>）= 1。
 - b) 设置时钟极性，CKP（SPIxCON<6>）= 0。
 - c) 设置帧极性，FRMPOL（SPIxCON<29>）= 1。
 - d) 对于 16 位音频通道数据，设置 MODE<32,16>（SPIxCON<11:10>）= 0。
 - e) 通过将 ON 位（SPIxCON<15>）置 1 使能 SPI 操作。
12. ON 位置 1 后将立即开始发送（和接收）。

例 23-6: 左对齐主模式，16 位通道数据，32 位帧

```
/* The following code example will initialize the SPI1 Module in Left-Justified Master mode. */
/* It assumes that none of the SPI1 input pins are shared with an analog input. */

unsigned int rData;
IEC0CLR=0x03800000;      // disable all interrupts
SPI1CON = 0;              // Stops and resets the SPI1
SPI1CON2 = 0;             // Reset audio settings
SPI1BRG = 0;
rData=SPI1BUF;            // clears the receive buffer
IFS0CLR=0x03800000;      // clear any existing event
IPC5CLR=0x1f000000;      // clear the priority
IPC5SET=0x0d000000;      // Set IPL = 3, Subpriority 1
IEC0SET=0x03800000;      // Enable RX, TX and Error interrupts
SPI1STATCLR=0x40;        // clear the Overflow
SPI1CON2=0x00000081;      // Left-Justified Mode, AUDEN = 1, AUDMON = 0
SPI1BRG =0x4D;           // (to generate 256 kbps sample rate, PBCLK @ 40 MHz)
SPI1CON =0x20008040;      // Master mode, SPI ON, CKP = 0, FRMPOL = 1, MSTEN = 1
                          // 16-bit audio data, 32 bits per frame

// from here, the device is ready to receive and transmit data

/* Note: A few of bits related to frame settings are not required to be set in the SPI1CON */
/* register during audio mode operation. Please refer to the notes in the SPIxCON2 register.*/
```


23.4.5.3 右对齐模式

在右对齐模式下，SPI 模块在将数据与最后一个时钟周期对齐后对音频采样数据的 MSb 进行移位。可通过将 DISSDO 位 (SPIxCON<12>) 置 0 来将音频采样数据之前的位驱动为逻辑电平 0。当 DISSDO = 0 时，模块将忽略未使用的位时隙。

- 所需配置：
要将 SPI 模块设置为右对齐模式，必须设置以下位：
 - AUDMOD<1:0> (SPIxCON2<1:0>) = 10
 - FRMPOL (SPIxCON<29>) = 1
 - CKP (SPIxCON<6>) = 0

在 LSb 与最后一个时钟周期对齐后，这将使 SDO 和 LRCK 在 SCK 的上升沿发生转变。关于 16、24 和 32 位音频数据传输，请参见图 23-25 和图 23-26 中所示的采样波形图。

图 23-25: 16 位数据 / 通道或 32 位数据 / 通道的右对齐

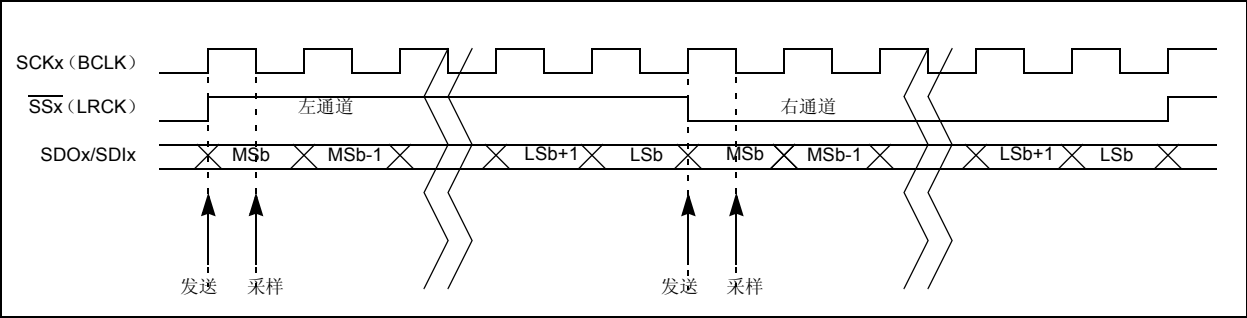
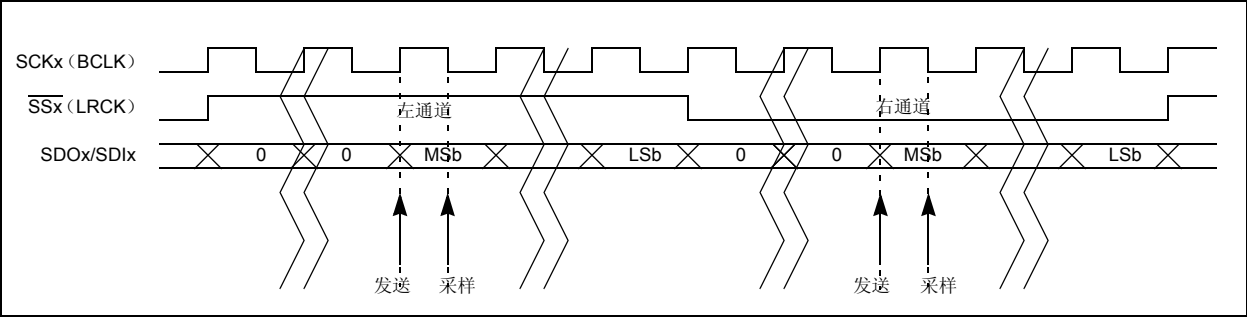


图 23-26: 16/24 位数据和 32 位通道的右对齐



23.4.5.3.1 右对齐音频从工作模式

通过以下步骤将 SPI 模块设置为右对齐音频从工作模式：

1. 如果使用中断，则禁止相应 IECx 寄存器中的 SPI 中断。
2. 通过清零 ON 位（SPIxCON<15>）停止并复位 SPI 模块。
3. 复位 SPI 音频配置寄存器 SPIxCON2。
4. 清除接收缓冲区。
5. 如果使用标准缓冲模式，则将 ENHBUF 位（SPIxCON<16>）清零，如果使用增强型缓冲模式，则将该位置 1。
6. 如果使用中断，则执行以下步骤：
 - a) 清除相应 IFSx 寄存器中的 SPIx 中断标志/事件。
 - b) 写入相应 IPCx 寄存器中的 SPIx 中断优先级位和子优先级位。
 - c) 将相应 IECx 寄存器中的 SPIx 中断使能位置 1。
7. 清零 SPIROV 位（SPIxSTAT<6>）。
8. 将所需设置写入 SPIxCON2 寄存器。对于右对齐模式，必须将 AUDMOD<1:0> 位（SPIxCON2<1:0>）设置为 10，并且必须将 AUDEN 位（SPIxCON2<7>）置 1 才能使能音频协议。
9. 将所需设置写入 SPIxCON 寄存器：
 - a) 设置为从模式，MSTEN（SPIxCON<5>）= 0。
 - b) 设置时钟极性，CKP（SPIxCON<6>）= 0。
 - c) 设置帧极性，FRMPOL（SPIxCON<29>）= 1。
 - d) 对于 16 位音频通道数据，设置 MODE<32,16>（SPIxCON<11:10>）= 0。
 - e) 通过将 ON 位（SPIxCON<15>）置 1 使能 SPI 操作。
10. 主器件一提供 BCLK 和 LRCK，就开始发送（和接收）。

例 23-7: 右对齐从模式，16 位通道数据，32 位帧

```
/* The following code example will initialize the SPI1 Module in Right-Justified Slave mode. */
/* It assumes that none of the SPI1 input pins are shared with an analog input. */

unsigned int rData;
IEC0CLR=0x03800000;    // disable all interrupts
SPI1CON=0;              // Stops and resets the SPI1.
SPI1CON2=0;            // Reset audio settings
rData=SPI1BUF;         // clears the receive buffer
IFS0CLR=0x03800000;    // clear any existing event
IPC5CLR=0x1f000000;    // clear the priority
IPC5SET=0x0d000000;    // Set IPL = 3, Subpriority 1
IEC0SET=0x03800000;    // Enable RX, TX and Error interrupts
SPI1STATCLR=0x40;      // clear the Overflow
SPI1CON2=0x00000082;    // Right-Justified Mode, AUDEN = 1, AUDMON = 0
SPI1CON=0x20008000;     // Slave mode, SPI ON, CKP = 0, FRMPOL = 1,
                        // 16-bit audio data, 32 bits per frame
                        // DISSDO = 0, transmit unused bit slots with logic level 0
                        // DISSDI = 0, receiver to ignore the unused bit slots

// from here, the device is ready to receive and transmit data

/* Note: A few of bits related to frame settings are not required to be set in the SPI1CON */
/* register during audio mode operation. Please refer to the notes in the SPIxCON2 register.*/
```

23.4.5.3.2 右对齐音频主工作模式

通过以下步骤将 SPI 模块设置为右对齐音频主工作模式：

1. 如果使用中断，则禁止相应 IECx 寄存器中的 SPI 中断。
2. 通过清零 ON 位 (SPIxCON<15>) 停止并复位 SPI 模块。
3. 复位 SPI 音频配置寄存器 SPIxCON2。
4. 复位波特率寄存器 SPIxBRG。
5. 清除接收缓冲区。
6. 如果使用标准缓冲模式，则将 ENHBUF 位 (SPIxCON<16>) 清零，如果使用增强型缓冲模式，则将该位置 1。
7. 如果使用中断，则执行以下附加步骤：
 - a) 清除相应 IFSx 寄存器中的 SPIx 中断标志 / 事件。
 - b) 写入相应 IPCx 寄存器中的 SPIx 中断优先级位和子优先级位。
 - c) 将相应 IECx 寄存器中的 SPIx 中断使能位置 1。
8. 清零 SPIROV 位 (SPIxSTAT<6>)。
9. 将所需设置写入 SPIxCON2 寄存器。对于右对齐模式，必须将 AUDMOD<1:0> 位 (SPIxCON2<1:0>) 设置为 10，并且必须将 AUDEN 位 (SPIxCON2<7>) 置 1 才能使能音频协议。
10. 将 SPIxBRG 波特率寄存器设置为 0x4D (以在 PBCLK 为 40 MHz 时产生约 256 kbps 的采样率)。
11. 将所需设置写入 SPIxCON 寄存器：
 - a) 设置为主模式，MSTEN (SPIxCON<5>) = 1。
 - b) 设置时钟极性，CKP (SPIxCON<6>) = 0。
 - c) 设置帧极性，FRMPOL (SPIxCON<29>) = 1。
 - d) 对于 16 位音频通道数据，设置 MODE<32,16> (SPIxCON<11:10>) = 0。
 - e) 通过将 ON 位 (SPIxCON<15>) 置 1 使能 SPI 操作。
12. ON 位置 1 后将立即开始发送 (和接收)。

例 23-8: 右对齐主模式，16 位通道数据，32 位帧

```

/* The following code example will initialize the SPI1 Module in Right-Justified Master mode. */
/* It assumes that none of the SPI1 input pins are shared with an analog input. */

unsigned int rData;
IEC0CLR=0x03800000; // disable all interrupts
SPI1CON = 0; // Stops and resets the SPI1.
SPI1CON2 = 0; // Reset audio settings
SPI1BRG=0;
rData=SPI1BUF; // clears the receive buffer
IFS0CLR=0x03800000; // clear any existing event
IPC5CLR=0x1f000000; // clear the priority
IPC5SET=0x0d000000; // Set IPL=3, Subpriority 1
IEC0SET=0x03800000; // Enable RX, TX and Error interrupts
SPI1STATCLR=0x40; // clear the Overflow
SPI1CON2=0x00000082; // Right-Justified Mode, AUDEN =1, AUDMON=0
SPI1BRG =0x4D; // (to generate 256 kbps sample rate, PBCLK @ 40 MHz)
SPI1CON =0x20008020; // Master mode, SPI ON, CKP = 0, FRMPOL = 1, MSTN = 1
// 16-bit audio data, 32 bits per frame
// DISSDO = 0, transmit unused bit slots with logic level 0
// DISSDI = 0, receiver to ignore the unused bit slots

// from here, the device is ready to receive and transmit data

/* Note: A few of bits related to frame settings are not required to be set in the SPI1CON */
/* register during audio mode operation. Please refer to the notes in the SPIxCON2 register.*/

```

23.4.5.4 PCM/DSP 模式

PCM/DSP 协议模式可用于与某些编解码器和特定 DSP 器件通信。此模式会修改 LRCK 的行为以及音频数据间隔。在 PCM/DSP 模式下，LRCK 的宽度可以是单个位（即 1 个 SCK），也可以与音频数据（16、24 和 32 位）的宽度相同。音频数据在帧中以左通道数据后面紧跟右通道数据的形式打包。当此器件是主器件时，帧长度仍为 32 或 64 个时钟。

在 PCM/DSP 模式下，发送器在 SCK 的第一个或第二个发送边沿（见 SPIFE 位（SPIxCON<17>））驱动音频数据的（左通道）MSb（在 LRCK 转变后）。紧随（左通道）LSb 之后，发送器驱动（右通道）MSb。

- 所需配置设置：

要将 SPI 模块设置为左对齐模式，必须设置以下位：

- AUDMOD<1:0> 位（SPIxCON2<1:0>）= 11

关于 16、24 和 32 位音频数据传输，请参见图 23-27 和图 23-28 中所示的采样波形图。

图 23-27: 16 位数据 / 通道或 32 位数据 / 通道的 PCM/DSP

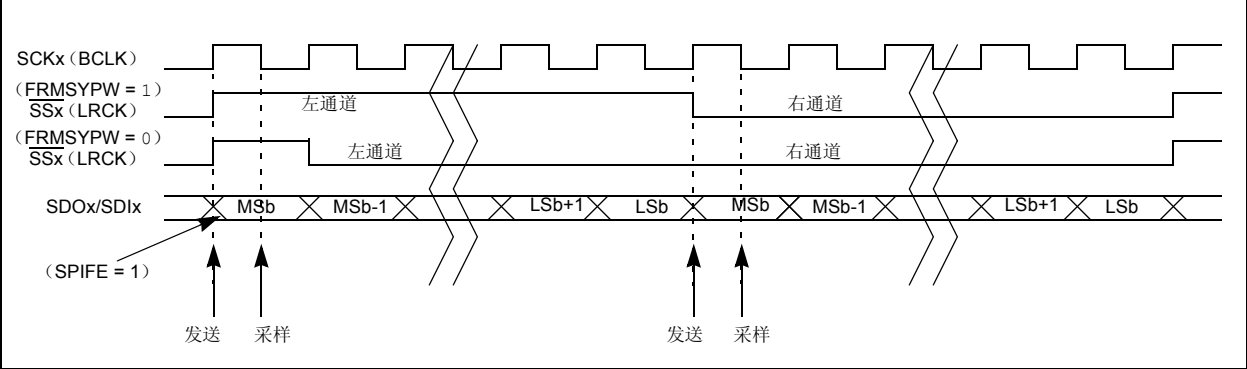
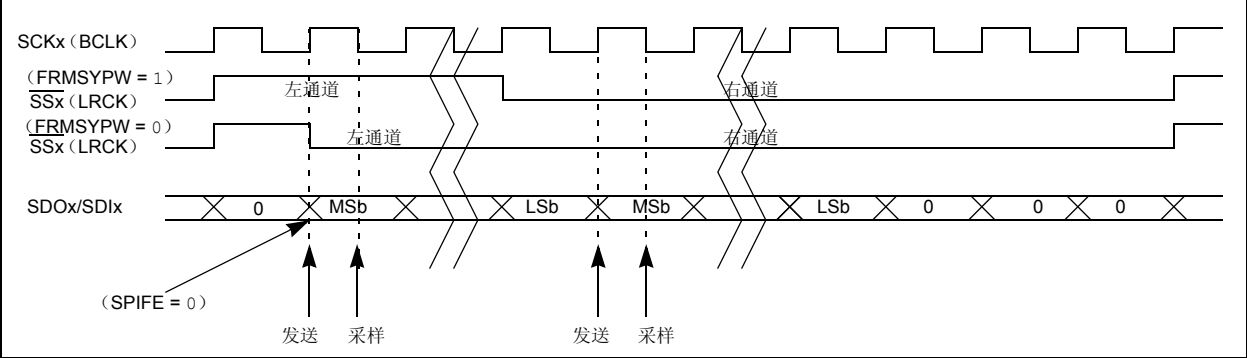


图 23-28: 16/24 位数据和 32 位通道的 PCM/DSP



23.4.5.4.1 PCM/DSP 音频从工作模式

通过以下步骤将 SPI 模块设置为 PCM/DSP 音频从工作模式：

1. 如果使用中断，则禁止相应 IECx 寄存器中的 SPI 中断。
2. 通过清零 ON 位 (SPIxCON<15>) 停止并复位 SPI 模块。
3. 复位 SPI 音频配置寄存器 SPIxCON2。
4. 清除接收缓冲区。
5. 如果使用标准缓冲模式，则将 ENHBUF 位 (SPIxCON<16>) 清零，如果使用增强型缓冲模式，则将该位置 1。
6. 如果使用中断，则执行以下附加步骤：
 - a) 清除相应 IFSx 寄存器中的 SPIx 中断标志/事件。
 - b) 写入相应 IPCx 寄存器中的 SPIx 中断优先级位和子优先级位。
 - c) 将相应 IECx 寄存器中的 SPIx 中断使能位置 1。
7. 清零 SPIROV 位 (SPIxSTAT<6>)。
8. 将所需设置写入 SPIxCON2 寄存器。对于 DSP/PCM 模式，必须将 AUDMOD<1:0> 位 (SPIxCON2<1:0>) 设置为 1，并且必须将 AUDEN 位 (SPIxCON2<7>) 置 1 才能使能音频协议。
9. 将所需设置写入 SPIxCON 寄存器：
 - a) 设置为从模式，MSTEN (SPIxCON<5>) = 0。
 - b) 对于 16 位音频通道数据，设置 MODE<32,16> (SPIxCON<11:10>) = 0。
 - c) 通过将 ON 位 (SPIxCON<15>) 置 1 使能 SPI 操作。
10. 主器件一提供 BCLK 和 LRCK，就开始发送 (和接收)。

例 23-9: PCM/DSP 从模式，16 位通道数据，32 位帧

```

/* The following code example will initialize the SPI1 Module in PCM/DSP Slave Mode. */
/* It assumes that none of the SPI1 input pins are shared with an analog input. */

unsigned int rData;
IEC0CLR=0x03800000; // disable all interrupts
SPI1CON = 0; // Stops and resets the SPI1.
SPI1CON2 = 0; // Reset audio settings
rData=SPI1BUF; // clears the receive buffer
IFS0CLR=0x03800000; // clear any existing event
IPC5CLR=0x1f000000; // clear the priority
IPC5SET=0x0d000000; // Set IPL = 3, Subpriority 1
IEC0SET=0x03800000; // Enable RX, TX and Error interrupts
SPI1STATCLR=0x40; // clear the Overflow
SPI1CON2=0x00000083; // PCM/DSP Slave Mode, AUDEN = 1, AUDMON = 0
SPI1CON =0x00008000; // Slave mode, SPI ON, FRMSYPW = 0
// 16-bit audio data, 32 bits per frame
// from here, the device is ready to receive and transmit data

/* Note: A few of bits related to frame settings are not required to be set in the SPI1CON */
/* register during audio mode operation. Please refer to the notes in the SPIxCON2 register.*/

```

23.4.5.4.2 PCM/DSP 音频主工作模式

通过以下步骤将 SPI 模块设置为 PCM/DSP 音频主工作模式：

1. 如果使用中断，则禁止相应 IECx 寄存器中的 SPI 中断。
2. 通过清零 ON 位（SPIxCON<15>）停止并复位 SPI 模块。
3. 复位 SPI 音频配置寄存器 SPIxCON2。
4. 复位波特率寄存器 SPIxBRG。
5. 清除接收缓冲区。
6. 如果使用标准缓冲模式，则将 ENHBUF 位（SPIxCON<16>）清零，如果使用增强型缓冲模式，则将该位置 1。
7. 如果使用中断，则执行以下步骤：
 - a) 清除相应 IFSx 寄存器中的 SPIx 中断标志 / 事件。
 - b) 写入相应 IPCx 寄存器中的 SPIx 中断优先级位和子优先级位。
 - c) 将相应 IECx 寄存器中的 SPIx 中断使能位置 1。
8. 清零 SPIROV 位（SPIxSTAT<6>）。
9. 将所需设置写入 SPIxCON2 寄存器。对于 DSP/PCM 模式，必须将 AUDMOD<1:0> 位（SPIxCON2<1:0>）设置为 11，并且必须将 AUDEN 位（SPIxCON<7>）置 1 才能使能音频协议。
10. 将 SPIxBRG 波特率寄存器设置为 0x4D（以在 PBCLK 为 40 MHz 时产生约 256 kbps 的采样率）。
11. 将所需设置写入 SPIxCON 寄存器：
 - a) 设置为主模式，MSTEN（SPIxCON<5>）= 1。
 - b) 对于 16 位音频通道数据，设置 MODE<32,16>（SPIxCON<11:10>）= 0。
 - c) 通过将 ON 位（SPIxCON<15>）置 1 使能 SPI 操作。
12. ON 位置 1 后将立即开始发送（和接收）。

例 23-10: PCM/DSP 主模式，16 位通道数据，32 位帧

```
/* The following code example will initialize the SPI1 Module in PCM/DSP Master Mode. */
/* It assumes that none of the SPI1 input pins are shared with an analog input. */

unsigned int rData;
IEC0CLR=0x03800000;    // disable all interrupts
SPI1CON = 0;           // Stops and resets the SPI1.
SPI1CON2 = 0;          // Reset audio settings
rData=SPI1BUF;         // clears the receive buffer
IFS0CLR=0x03800000;    // clear any existing event
IPC5CLR=0x1f000000;    // clear the priority
IPC5SET=0x0d000000;    // Set IPL=3, Subpriority 1
IEC0SET=0x03800000;    // Enable RX, TX and Error interrupts
SPI1STATCLR=0x40;      // clear the Overflow
SPI1CON2=0x00000083;    // PCM/DSP Master Mode, AUDEN =1, AUDMON=0
SPI1BRG =0x4D;         // (to generate 256 kbps sample rate, PBCLK @ 40 MHz)
SPI1CON =0x00008020;    // Master mode, SPI ON, FRMSYPW = 0
                        // 16-bit audio data, 32 bits per frame
// from here, the device is ready to receive and transmit data

/* Note: A few of bits related to frame settings are not required to be set in the SPI1CON */
/* register during audio mode operation. Please refer to the notes in the SPIxCON2 register.*/
```

23.4.6 音频协议模式特性

23.4.6.1 BCLK/SCK 和 LRCK 生成

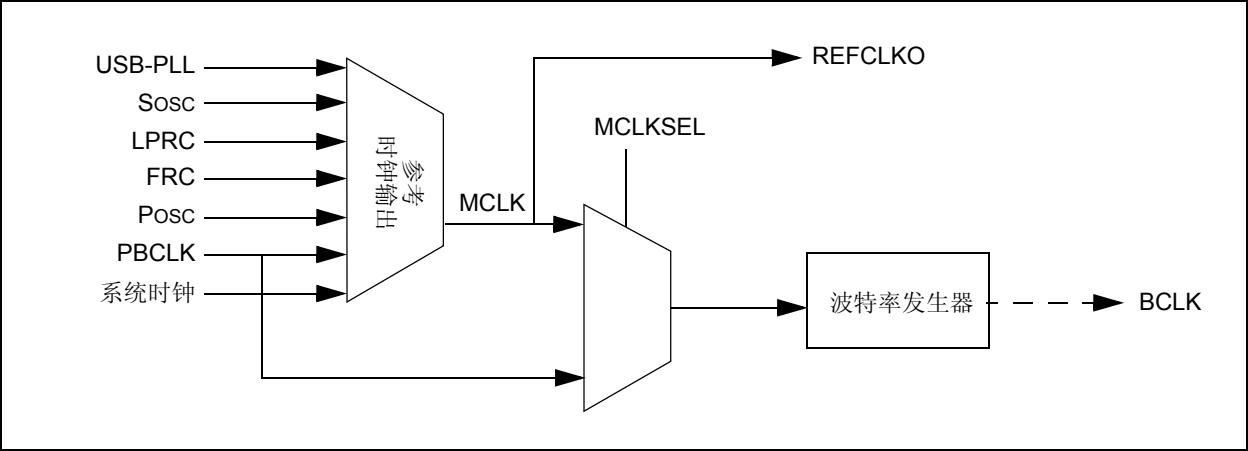
BCLK 和 LRCK 的生成是主模式下的关键要求。SCK 和 LRCK 的帧频率由 $MODE<32,16>$ 位 ($SPIxCON<11:10>$) 定义。当帧为 64 位时，SCK 的频率是 LRCK 频率的 64 倍。同样，当帧为 32 位时，SCK 的频率是 LRCK 频率的 32 倍。SCK 的频率必须通过 LRCK 的翻转率和帧大小得出。

例如，要在 $PBCLK = 36.864\text{ MHz}$ 时以 8 kHz 采样 16 位通道数据，应将 $SPIxBRG$ 寄存器设置为 “0x47” 以生成 8 kHz LRCK。

23.4.6.2 主模式时钟和 MCLK

作为主器件的 SPI 模块能通过 PBCLK ($MCLKSEL = 0$) 在内部生成 BCLK 和 LRCK。SPI 模块可通过参考输出 REFCLKO 功能为外部编解码器器件生成时钟（见图 23-29），尽管一些编解码器也可以通过晶振生成自己的 MCLK 以提供准确的音频采样率。图 23-30 显示了可被编解码器用作 MCLKIN 的 REFCLKO 时钟。

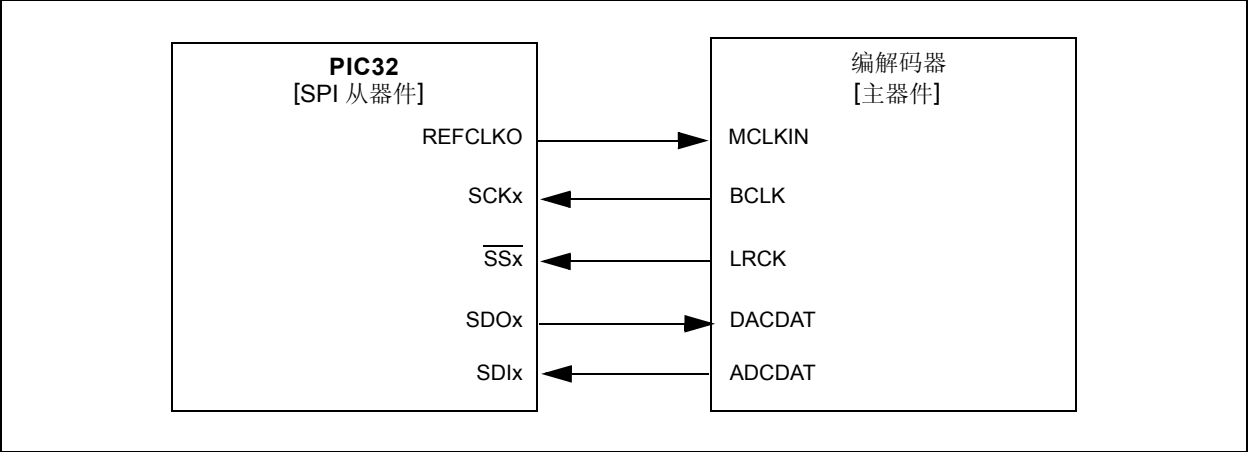
图 23-29: SPI 主器件时钟生成



有关参考时钟输出接口的更多信息，请参见具体器件的数据手册。

图 23-30 显示了 SPI 从器件和编解码器主器件之间的接口，时钟来自 MCLK 输入参考。

图 23-30: SPI 从器件和编解码器主器件——时钟来自 MCLK



23.4.6.2.1 使用 REFCLKO 的 I²S 音频主工作模式

可使用以下步骤将 SPI 模块设置为 I²S 音频主工作模式并使能 MCLK。SPI 模块初始化为生成 256 kbps 的 BCLK，并且通过参考振荡器输出配置寄存器从 PBCLK 得出 MCLK。当与编解码器从器件连接时，典型应用可以是播放 PCM 数据（8 kHz 采样率，16 位数据，32 位帧）。

1. 如果使用中断，则禁止相应 IECx 寄存器中的 SPI 中断。
2. 通过清零 ON 位（SPIxCON<15>）停止并复位 SPI 模块。
3. 复位 SPI 音频配置寄存器 SPIxCON2。
4. 复位参考振荡器控制器寄存器 REFOCON。
5. 复位波特率寄存器 SPIxBRG。
6. 清除接收缓冲区。
7. 如果使用标准缓冲模式，则将 ENHBUF 位（SPIxCON<16>）清零，如果使用增强型缓冲模式，则将该位置 1。
8. 如果使用中断，则执行以下附加步骤：
 - a) 清除相应 IFSx 寄存器中的 SPIx 中断标志 / 事件。
 - b) 写入相应 IPCx 寄存器中的 SPIx 中断优先级位和子优先级位。
 - c) 将相应 IECx 寄存器中的 SPIx 中断使能位置 1。
 - d) 清零 SPIROV 位（SPIxSTAT<6>）。
9. 将所需设置写入 SPIxCON2 寄存器。对于 I²S 模式，必须将 AUDMOD<1:0> 位（SPIxCON2<1:0>）设置为 00，并且必须将 AUDEN 位（SPIxCON2<7>）置 1 才能使能音频协议。
10. 设置参考振荡器控制器寄存器 REFOCON：
 - a) RODIV<14:0>（REFOCON<30:16>）= 0。
 - b) ON（REFOCON<15>）= 1，使能参考振荡器。
 - c) OE（REFOCON<4>）= 1，使能输出。
11. 将 SPIxBRG 波特率寄存器设置为 0x4D（以在 PBCLK 为 40 MHz 时产生约 256 kbps 的采样率）。
12. 将所需设置写入 SPIxCON 寄存器：
 - a) MSTEN（SPIxCON<5>）= 1。
 - b) CKP（SPIxCON<6>）= 1。
 - c) 对于 16 位音频通道数据，MODE<32,16>（SPIxCON<11:10>）= 0。
 - d) MCLKSEL（SPIxCON<23>）= 1，主模式。
 - e) 通过将 ON 位（SPIxCON<15>）置 1 使能 SPI 操作。
13. 主器件一提供 BCLK 和 LRCK，就开始发送（和接收）。

例 23-11: I²S 主模式, 256 kbps BCLK, 16 位通道数据, 32 位帧

```

/* The following code example will initialize the SPI1 Module in I2S Slave mode.
/* It assumes that none of the SPI1 input pins are shared with an analog input. */

unsigned int rData;
IEC0CLR=0x03800000; // disable all interrupts
SPI1CON = 0; // Stops and resets the SPI1.
SPI1CON2 = 0; // Reset audio settings
REFOCON = 0x0; // Reset reference oscillator register
SPI1BRG=0; // Reset Baud rate register
rData=SPI1BUF; // clears the receive buffer
IFS0CLR=0x03800000; // clear any existing event
IPC5CLR=0x1f000000; // clear the priority
IPC5SET=0x0d000000; // Set IPL = 3, Subpriority 1
IEC0SET=0x03800000; // Enable RX, TX and Error interrupts

SPI1STATCLR=0x40; // clear the Overflow
SPI1CON2=0x00000080; // I2S Mode, AUDEN =1, AUDMON=0
SPI1BRG =0x4D; // (to generate 256 kbps sample rate, PBCLK @ 40 MHz)
REFOCON = 0x8001; // ON = 1, ROSEL = 1 for PBCLK
SPI1CON =0x00808060; // MCLKSEL = 1, MSTEN = 1, ON = 1, CKP = 1, 16-bit audio channel
// data, 32-bits per frame
// from here, the device is ready to receive and transmit data

/* Note: A few of bits related to frame settings are not required to be set in the SPI1CON */
/* register during audio mode operation. Please refer to the notes in the SPIxCON2 register.*/

```

注: 使用参考时钟输出为编解码器生成 MCLK 可能并不是最佳选择。将时钟输出驱动到 I/O 焊盘会引起抖动, 这种抖动可能会降低编解码器的音频保真度。对于编解码器来说, 最佳解决方案是使用晶振且编解码器是主 I²S/音频器件。

23.4.7 单声道模式与立体声模式

SPI 模块通过设置 AUDMONO 位 (SPIxCON2<3>) 来使音频数据以单声道或立体声模式传输。当 AUDMONO 位设置为 0 (立体声模式) 时, 移位寄存器对每个 FIFO 存储单元使用一次, 这为每个通道提供了唯一的立体声数据流。当 AUDMONO 位设置为 1 (单声道模式) 时, 移位寄存器对每个 FIFO 存储单元使用两次, 这为每个通道提供了相同的单声道音频数据流。

注: 接收数据不受 AUDMONO 位设置的影响。

23.4.8 流数据支持和错误处理

大多数音频流应用会连续发送或接收数据。这是为了在工作期间使通道保持在活动状态并尽可能保证最佳精度。由于流音频的缘故, 数据源可能出现数据突发或者数据包丢失, 这会导致 SPI 模块遇到类似数据不足的情况。软件需要能够从数据不足的状态恢复。

当忽略发送数据不足 (IGNTUR) 位 (SPIxCON2<8>) 置 1 时, 将忽略数据不足的情况。当软件不理睬或不需要了解数据不足情况时, 这一设置是非常有用的。当遇到数据不足的情况时, 如果 SPITUREN (SPIxCON2<10>) = 1, 则 SPI 模块将 SPITUR 位 (SPIxSTAT<8>) 置 1, 并且 SPI 模块将保持在错误状态, 直到软件清除此状态或 ON 位 (SPIxCON<15>) = 0 为止。

在数据不足期间, SPI 模块在 SPIxSR 寄存器中装入零, 而不是 SPIxTXB 寄存器的数据, 并且 SPI 模块将继续发送零。当错误情况清除 (即, SPIxTXB 寄存器非空) 后, SPI 模块在下一个 LRCK 帧的边界将发送缓冲区中的音频数据装入 SPIxSR 寄存器, 软件必须确保左右音频数据始终成对传输到 FIFO。

当忽略接收溢出 (IGNROV) 位 (SPIxCON2<9>) 置 1 时, 将忽略接收溢出情况。当系统中存在软件必须适当处理的一般性能问题时, 这一设置是非常有用的。另一种处理接收溢出的方法是当系统不需要接收音频数据时将 DISSDI 位 (SPIxCON<4>) 置 1。在 LRCK 的前沿实时更改 DISSDI 位, 接收移位寄存器将开始接收。

23.5 中断

SPI 模块能够产生一些中断，以反映在数据通信期间发生的事件。它可以产生以下类型的中断：

- 接收数据可用中断通过 SPI1RXIF 和 SPI2RXIF 指示。在 SPIxBUF 接收缓冲区中汇集新数据时，将会发生该事件。
- 发送缓冲区为空中断通过 SPI1TXIF 和 SPI2TXIF 指示。在 SPIxBUF 发送缓冲区中有可用空间，可以写入新数据时，将会发生该事件。
- 错误中断通过 SPI1EIF 和 SPI2EIF 指示。当 SPIxBUF 接收缓冲区存在溢出条件（即，汇集了新的接收数据，但前一个数据尚未被读取）、当发送缓冲区数据不足或发生 FRMERR 事件时，将会发生该事件。

所有这些必须用软件清零的中断标志均位于 IFSx 寄存器中。关于更多信息，请参见具体器件的数据手册。

要允许 SPI 中断，请使用相关 IECx 寄存器中的相应 SPI 中断允许位：SPIxRXIE、SPIxTXIE 和 SPIxFIE。

此外，还必须在相关 IPCx 寄存器中通过 SPIxIP 和 SPIxIS 位来配置中断优先级位和中断子优先级位。

使用增强型缓冲模式时，SPI 控制寄存器（SPIxCON<3:2>）中的 SPI 发送缓冲区为空中断模式位（STXISEL<1:0>）可用于配置以下条件下发送缓冲区为空中断的操作：缓冲区未满、一半或更多元素为空、全空，或者最后一个传输数据移出。

类似地，使用增强型缓冲模式时，SPI 控制寄存器（SPIxCON<1:0>）中的 SPI 接收缓冲区为满中断模式位（SRXISEL<1:0>）可用于配置以下条件下接收缓冲区为满中断的产生：缓冲区为满、一半或更多元素为满、不为空，或者最后一个字被读取。

注： 增强型缓冲模式并非在所有器件上都可用。详情请参见具体器件数据手册。

更多详细信息，请参见第 8 章“中断”（DS61108）。

23.5.1 中断配置

每个 SPI 模块都有 3 个专用的中断标志位：SPIxEIF、SPIxRXIF 和 SPIxTXIF，以及相应的中断允许 / 屏蔽位 SPIxEIE、SPIxRXIE 和 SPIxTXIE。这些位用于决定中断源和使能 / 禁止各个中断源。请注意，特定 SPI 模块的所有中断源共用一个中断向量。每个 SPI 模块可以具有独立于其他 SPI 模块的优先级。

当 SPI 发送缓冲区为空，并且可以向 SPIxBUF 寄存器写入另一个字符时，SPIxTXIF 会置 1。当 SPIxBUF 中有接收字符可用时，SPIxRXIF 会置 1。当发生接收溢出条件时，SPIxEIF 会置 1。

请注意，SPIxTXIF、SPIxRXIF 和 SPIxEIF 位是否置 1 与相应允许位的状态无关。如果需要，可以用软件查询中断标志位。

SPIxEIE、SPIxTXIE 和 SPIxRXIE 位用于定义在相应的 SPIxEIF、SPIxTXIF 或 SPIxRXIF 位置 1 时，中断控制器的行为。当相应的中断允许位清零时，中断控制器不会为事件产生 CPU 中断。如果中断允许位置 1，则中断控制器会在相应的中断标志位置 1 时向 CPU 产生中断（受以下段落中概述的优先级和子优先级制约）。

处理特定中断的用户软件程序负责在服务程序完成之前清零相应的中断标志位。

每个 SPI 模块的优先级可以通过 SPIxIP<2:0> 位独立设置。该优先级定义了中断源将分配到的优先级组。优先级组值的范围为 7（最高优先级）到 0（不产生中断）。较高优先级组中的中断会抢占正在处理、但优先级较低的中断。每个错误中断源分别由 SPIxCON2 寄存器中的 FRMERREN、SPIROVEN 和 SPITUREN 位控制。

子优先级位用于设置中断源在优先级组中的优先级。子优先级 SPIxIS<1:0> 值的范围为 3（最高优先级）到 0（最低优先级）。处于相同优先级组，但具有更高子优先级值的中断不会抢占子优先级较低、但正在进行的中断。

优先级组和子优先级位让多个中断源可以共用相同的优先级和子优先级。如果在该配置下同时发生若干个中断，则中断源在优先级 / 子优先级组对中的自然顺序将决定所产生的中断。自然优先级基于中断源的向量编号。向量编号越小，中断的自然优先级就越高。在当前中断的中断标志清零之后，所有不按照自然顺序执行的中断会基于优先级、子优先级和自然顺序产生相应的中断。

产生允许的中断之后，CPU 将跳转到为该中断分配的向量处。该中断的向量编号与自然顺序编号相同。然后，CPU 将在向量地址处开始执行代码。该向量地址处的用户代码应执行所需的任何特定于应用程序的操作，清零中断标志 SPIxEIF、SPIxTXIF 或 SPIxRXIF，然后退出。关于中断的更多信息，请参见第 8 章 “中断”（DS61108）中的向量地址表详细信息。

例 23-12: 允许中断的 SPI 初始化代码示例

```
/*
The following code example illustrates an SPI1 interrupt configuration.
When the SPI1 interrupt is generated, the cpu will jump to the vector assigned to SPI1
interrupt.
It assumes that none of the SPI1 input pins are shared with an analog input.If so, the
AD1PCFG and corresponding TRIS registers have to be properly configured.
*/

int rData;

IEC0CLR=0x03800000;           // disable all SPI interrupts
SPI1CON = 0;                   // Stops and resets the SPI1.
rData=SPI1BUF;                 // clears the receive buffer
IFS0CLR=0x03800000;           // clear any existing event
IPC5CLR=0x1f000000;           // clear the priority
IPC5SET=0x0d000000;           // Set IPL=3, Subpriority 1
IEC0SET=0x03800000;           // Enable RX, TX and Error interrupts

SPI1BRG=0x1;                  // use FFB/4 clock frequency
SPI1STATCLR=0x40;             // clear the Overflow
SPI1CON=0x8220;               // SPI ON, 8 bits transfer, SMP=1, Master mode
```

例 23-13: SPI1 ISR 代码示例

```
/*
The following code example demonstrates a simple interrupt service routine for SPI1
interrupts.The user's code at this vector should perform any application specific operations
and must clear the SPI1 interrupt flags before exiting.
*/

void __ISR(_SPI_1_VECTOR, ip13)__SPI1Interrupt(void)
{
    // ... perform application specific operations in response to the
    // interrupt

    IFS0CLR = 0x03800000;       // Be sure to clear the SPI1 interrupt flags
                                // before exiting the service routine.
}
```

对于使用增强型缓冲模式的器件，用户应用程序应在处理中断条件之后清零中断请求标志。

如果发生了 SPI 中断，ISR 应读取 SPI 数据缓冲（SPIxBUF）寄存器，然后清零 SPI 中断标志，如例 23-14 所示。

例 23-14: SPI1 ISR 代码示例（对于使用增强型缓冲模式的器件）

```
/*
   The following code example demonstrates a simple interrupt service routine for SPI1
   interrupts. The user's code at this vector should perform any application specific operations
   and must clear the SPI1 interrupt flags before exiting.
*/

void __ISR(_SPI_1_VECTOR, IPL3) __SPI1Interrupt(void)
{
    int Data;                // Read SPI data buffer
    Data = SPI1BUF;

    // ... perform application specific operations in response to the
    // interrupt

    IFS0CLR = 0x03800000;    // Be sure to clear the SPI1 interrupt flags
    // before exiting the service routine.
}
```

注： SPI1 ISR 代码示例显示的是 MPLAB® C32 C 编译器的特定语法。关于对 ISR 的支持，请参见编译器手册。

23.6 节能和调试模式下的操作

23.6.1 休眠模式

当器件进入 Sleep（休眠）模式时，系统时钟被禁止。Sleep（休眠）模式期间的确切 SPI 模块操作取决于当前工作模式。以下小节介绍了特定于模式的行为。

23.6.1.1 休眠模式下的主模式

在 Sleep（休眠）模式下，应注意以下事项：

- 波特率发生器停止并可能会复位（请查看器件数据手册进行确认）。
- 正在进行的发送和接收序列会被中止。当退出 Sleep（休眠）模式时，模块可能不会继续执行被中止的序列。（同样，请查看数据手册进行确认。）
- 进入 Sleep（休眠）模式后，模块不会发送或接收任何新数据。

注： 为了防止发送和接收序列意外中止，请先等待当前发送完成，然后再激活 Sleep（休眠）模式。

23.6.1.2 休眠模式下的从模式

在从模式下，SPI 模块依靠外部 SPI 主器件提供的 SCK 工作。因为在从模式下，SCKx 上的时钟脉冲由外部提供，所以模块在 Sleep（休眠）模式下将继续工作。它将在进入 Sleep（休眠）模式的转变期间完成所有事务处理。完成事务处理后，SPIRBF 标志置 1。因此，SPIxRXIF 位将被置 1。如果允许了 SPI 中断（SPIxRXIE = 1）并且 SPI 中断的优先级大于当前的 CPU 优先级，将从 Sleep（休眠）模式唤醒器件，并且将从 SPIx 中断向量地址处恢复代码执行。如果 SPI 中断的优先级小于等于当前的 CPU 优先级，CPU 将保持在 Sleep（休眠）模式。

如果模块作为从器件工作，则在进入 Sleep（休眠）模式时它将不会复位。当 SPIx 模块进入或退出 Sleep（休眠）模式时，寄存器内容不受影响。

23.6.2 空闲模式

当器件进入 Idle（空闲）模式时，系统时钟源继续保持工作。

23.6.2.1 空闲模式下的主模式

SIDL 位（SPIxCON<13>）用于选择在 Idle（空闲）模式下模块是停止还是继续工作。

- 如果 SIDL = 1，则模块将在 Idle（空闲）模式下停止工作。模块在 Idle（空闲）模式下停止工作时将执行与在 Sleep（休眠）模式下相同的程序。
- 如果 SIDL = 0，则模块将在 Idle（空闲）模式下继续工作。

23.6.2.2 空闲模式下的从模式

无论 SIDL 设置如何，模块在 Idle（空闲）模式下都会继续工作。其行为与处于 Sleep（休眠）模式下的行为相同。

23.6.3 调试模式

23.6.3.1 SPIxBUF 的操作

23.6.3.1.1 调试模式期间的读操作

在 Debug（调试）模式期间，可以读取 SPIxBUF；但读操作不会影响任何状态位。例如，如果 SPIRBF 位（SPIxSTAT<0>）在进入 Debug（调试）模式时置 1，则即使在 Debug（调试）模式下读取了 SPIxBUF 寄存器，从 Debug（调试）模式退出时该位也会保持置 1。

23.6.3.1.2 调试模式期间的写操作

当 FRZ 置 1 时，写功能取决于 SPI 是处于主模式还是从模式。

在主模式下：写操作会将数据放入缓冲区，但只有在退出 Debug（调试）模式时，发送才会开始。

在从模式下：写操作会将数据放入缓冲区，并且数据将在每次主器件启动新事务时送出，即使器件仍然处于 Debug（调试）模式。

23.7 各种复位的影响

23.7.1 器件复位

在发生器件复位时，所有 SPI 寄存器会被强制设为它们的复位状态。当异步复位输入变为有效时，SPI 逻辑将：

- 将 SPIxCON 和 SPIxSTAT 中的所有位复位
- 将发送和接收缓冲区（SPIxBUF）复位为空状态
- 将波特率发生器复位

23.7.2 上电复位

在发生上电复位时，所有 SPI 寄存器会被强制设为它们的复位状态。

23.7.3 看门狗定时器复位

在发生看门狗定时器复位时，所有 SPI 寄存器会被强制设为它们的复位状态。

23.8 使用 SPI 模块的外设

没有其他外设使用 SPI 模块。

23.9 相关应用笔记

本节列出了与手册本章内容相关的应用笔记。这些应用笔记可能并不是专为 PIC32 系列器件而编写的，但其概念是相近的，通过适当修改并受到一定限制即可使用。当前与 SPI 模块相关的应用笔记有：

标题	应用笔记编号
Interfacing Microchip's MCP41XXX/MCP42XXX Digital Potentiometers to a PIC [®] Microcontroller	AN746
Interfacing Microchip's MCP3201 Analog-to-Digital Converter to the PIC [®] Microcontroller	AN719

注：如需获取更多 PIC32 系列器件的应用笔记和代码示例，请访问 Microchip 网站 (www.microchip.com)。

23.10 版本历史

版本 A（2007 年 7 月）

这是本文档的初始版本。

版本 B（2007 年 10 月）

修改了例 23-1、23-2 和 23-3；修改了表 23-5。

版本 C（2007 年 10 月）

更新了文档（删除了“机密”状态）。

版本 D（2008 年 4 月）

将状态修改为“初稿”；将 U-0 修改为 r-x。

版本 E（2008 年 6 月）

在寄存器 12-17 中增加了脚注编号；修改了例 23-4；修改了图 23-8；将保留位从“保持为”更改为“写入”；为 ON 位（SPIxCON 寄存器）增加了注释。

版本 F（2009 年 8 月）

该版本包括以下更改：

- 对整篇文档的文字和格式进行了少量更改
- 更新了第 23.2 节“状态和控制寄存器”中的寄存器介绍
- 寄存器汇总（表 23-3）
 - 删除了对清零、置 1、取反、IFS0、IFS1、IEC0、IEC1、IPC5 和 IPC7 寄存器的引用
 - 增加了“地址偏移”栏
 - 增加了介绍清零、置 1 和取反寄存器的“注 1”、“注 2”和“注 3”
 - 增加了以下位：MSEN、FRMSYPW、FRMCNT<2:0>、ENHBUF、STXISEL<1:0>、SRXISEL<1:0>、RXBUFELM<4:0>、SPITUR、SRMT、SPIRBE 和 SPITBF
- 删除了 IFS0、IFS1、IEC0、IEC1、IPC5 和 IPC7 寄存器
- 在以下寄存器中增加了介绍清零、置 1 和取反寄存器的注释：
 - SPIxCON
 - SPIxSTAT
 - SPIxBRG
- 在“SCKx 频率示例”表中增加了 60、72 和 80 MHz 的 SPIxBRG 设置（见表 23-4）
- 删除了“各种偏移量的 SPI 中断向量”表（表 23-4）
- 增加了第 23.3.2 节“缓冲模式”
- 在第 23.3.3.1 节“主模式工作”中增加了关于 MSEN 位的详细信息的段落
- 在第 23.3.6 节“帧 SPI 模式”中增加了关于 FRMSYPW 和 FRMCNT 位的两项详细信息
- 在第 23.4 节“中断”中增加了关于 STXISEL<1:0> 和 SRXISEL<1:0> 位的详细信息的两段
- 在第 23.4.1 节“中断配置”中例 23-4 之后增加了关于使用增强型缓冲模式器件的 SPI1 ISR 段落
- 增加了使用增强型缓冲模式的器件的 SPI1 ISR 代码示例（见例 23-14）
- 删除了第 23.8 节“I/O 引脚控制”

版本 G (2011 年 10 月)

该版本包括以下更改:

- 增加了参考相关文档的注释框, 并更新了 SPI 模块特性列表 (见第 23.1 节 “简介”)
- 在 SPI 特性中增加了音频协议接口模式 (见表 23-2)
- 更新了 SPI 模块框图 (见图 23-1)
- 对 SPI 控制寄存器进行如下更新 (见寄存器 23-1)
 - 增加了注 5
 - 更新了 Bit 26-24 (FRMCNT<2:0>) 的注释
 - 更新了 Bit 3-2 (STXISEL<2:0>) 的定义
 - 在所有出现 SPI_TBE_EVENT 和 SPI_RBF_EVENT 的地方, 分别用 SPIxTXIF 和 SPIxRXIF 替换
 - 更新了 MODE32 和 MODE16 位 (SPIxCON<11:10>) 的位值
- 增加了 SPIxCON2 寄存器和 MCLKSEL、DISSDI 及 FRMERR 位 (见表 23-3、寄存器 23-1 和寄存器 23-2)
- 向 CLR、SET 和 INV 寄存器, 以及 SPI 状态寄存器增加了注释 (见表 23-3 和寄存器 23-3)
- 更新了 SPI 状态寄存器中 SRMT 位的位值 (见寄存器 23-3 中的 Bit 7)
- 在第 23.3 节 “工作模式” 中增加了引脚用法的注释框
- 更新了 SCKx 采样频率 (见表 23-4)
- 交换了主模式操作顺序中的步骤 b) 和步骤 c) (见第 23.3.3.1 节 “主模式工作”)
- 在第 23.3.3.1 节 “主模式工作” 的第二个注释框后插入一个新段落。此外, 还更新了第二个注释框中的内容。
- 对 MSSEN 位增加了一个新段落。
- 在 SPI 主模式操作的 8 位模式时序图中增加了 SSx 引脚的时序 (见图 23-7)
- 交换了从模式操作顺序中的步骤 b) 和步骤 c) (见第 23.3.3.2 节 “从模式工作”)
- 在图 23-12 之前新增了一段关于其他接口选项的段落 (见第 23.3.6 节 “帧 SPI 模式”)
- 增加了章节第 23.4 节 “音频协议接口模式”
- 更新了第 23.5 节 “中断” 的项目列表中的位名称和寄存器参考
- 在第 23.6.3 节 “调试模式” 中, 删除了除第 23.6.3.1 节 “SPIxBUF 的操作” 以外的全部内容
- 删除了章节 23.9 “设计技巧”
- 通篇将提到 LRC 的地方更新为 LRCK
- 删除所有代码示例中的代码未经测试水印
- 对整篇文档的文字和格式进行了少量更改

注:

请注意以下有关 Microchip 器件代码保护功能的要点：

- Microchip 的产品均达到 Microchip 数据手册中所述的技术指标。
- Microchip 确信：在正常使用的情况下，Microchip 系列产品是当今市场上同类产品中最安全的产品之一。
- 目前，仍存在着恶意、甚至是非法破坏代码保护功能的行为。就我们所知，所有这些行为都不是以 Microchip 数据手册中规定的操作规范来使用 Microchip 产品的。这样做的人极可能侵犯了知识产权。
- Microchip 愿与那些注重代码完整性的客户合作。
- Microchip 或任何其他半导体厂商均无法保证其代码的安全性。代码保护并不意味着我们保证产品是“牢不可破”的。

代码保护功能处于持续发展中。Microchip 承诺将不断改进产品的代码保护功能。任何试图破坏 Microchip 代码保护功能的行为均可视为违反了《数字器件千年版权法案（Digital Millennium Copyright Act）》。如果这种行为导致他人在未经授权的情况下，能访问您的软件或其他受版权保护的成果，您有权依据该法案提起诉讼，从而制止这种行为。

提供本文档的中文版本仅为了便于理解。请勿忽视文档中包含的英文部分，因为其中提供了有关 Microchip 产品性能和使用情况的有用信息。Microchip Technology Inc. 及其分公司和相关公司、各级主管与员工及事务代理机构对译文中可能存在的任何差错不承担任何责任。建议参考 Microchip Technology Inc. 的英文原版文档。

本出版物中所述的器件应用信息及其他类似内容仅为您提供便利，它们可能由更新之信息所替代。确保应用符合技术规范，是您自身应负的责任。Microchip 对这些信息不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保，包括但不限于针对其使用情况、质量、性能、适销性或特定用途的适用性的声明或担保。Microchip 对因这些信息及使用这些信息而引起的后果不承担任何责任。如果将 Microchip 器件用于生命维持和/或生命安全应用，一切风险由买方自负。买方同意在由此引发任何一切伤害、索赔、诉讼或费用时，会维护和保障 Microchip 免于承担法律责任，并加以赔偿。在 Microchip 知识产权保护下，不得暗中以其他方式转让任何许可证。

商标

Microchip 的名称和徽标组合、Microchip 徽标、dsPIC、KEELOQ、KEELOQ 徽标、MPLAB、PIC、PICmicro、PICSTART、PIC³² 徽标、rfPIC 和 UNI/O 均为 Microchip Technology Inc. 在美国和其他国家或地区的注册商标。

FilterLab、Hampshire、HI-TECH C、Linear Active Thermistor、MXDEV、MXLAB、SEEVAL 和 The Embedded Control Solutions Company 均为 Microchip Technology Inc. 在美国的注册商标。

Analog-for-the-Digital Age、Application Maestro、chipKIT、chipKIT 徽标、CodeGuard、dsPICDEM、dsPICDEM.net、dsPICworks、dsSPEAK、ECAN、ECONOMONITOR、FanSense、HI-TIDE、In-Circuit Serial Programming、ICSP、Mindi、MiWi、MPASM、MPLAB Certified 徽标、MPLIB、MPLINK、mTouch、Omniscient Code Generation、PICC、PICC-18、PICDEM、PICDEM.net、PICkit、PICtail、REAL ICE、rfLAB、Select Mode、Total Endurance、TSHARC、UniWinDriver、WiperLock 和 ZENA 均为 Microchip Technology Inc. 在美国和其他国家或地区的商标。

SQTP 是 Microchip Technology Inc. 在美国的服务标记。

在此提及的所有其他商标均为各持有公司所有。

© 2010 - 2011, Microchip Technology Inc. 版权所有。

ISBN: 978-1-61341-849-9

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
== ISO/TS 16949:2002 ==

Microchip 位于美国亚利桑那州 Chandler 和 Tempe 与位于俄勒冈州 Gresham 的全球总部、设计和晶圆生产厂及位于美国加利福尼亚州和印度的设计中心均通过了 ISO/TS-16949:2009 认证。Microchip 的 PIC[®] MCU 与 dsPIC[®] DSC、KEELOQ[®] 跳码器件、串行 EEPROM、单片机外设、非易失性存储器和模拟产品严格遵守公司的质量体系流程。此外，Microchip 在开发系统的设计和生产方面的质量体系也已通过了 ISO 9001:2000 认证。

全球销售及服务中心

美洲

公司总部 **Corporate Office**
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 1-480-792-7200
Fax: 1-480-792-7277

技术支持:
<http://www.microchip.com/support>

网址: www.microchip.com

亚特兰大 Atlanta
Duluth, GA
Tel: 1-678-957-9614
Fax: 1-678-957-1455

波士顿 Boston
Westborough, MA
Tel: 1-774-760-0087
Fax: 1-774-760-0088

芝加哥 Chicago
Itasca, IL
Tel: 1-630-285-0071
Fax: 1-630-285-0075

克里夫兰 Cleveland
Independence, OH
Tel: 1-216-447-0464
Fax: 1-216-447-0643

达拉斯 Dallas
Addison, TX
Tel: 1-972-818-7423
Fax: 1-972-818-2924

底特律 Detroit
Farmington Hills, MI
Tel: 1-248-538-2250
Fax: 1-248-538-2260

印第安纳波利斯 Indianapolis
Noblesville, IN
Tel: 1-317-773-8323
Fax: 1-317-773-5453

洛杉矶 Los Angeles
Mission Viejo, CA
Tel: 1-949-462-9523
Fax: 1-949-462-9608

圣克拉拉 Santa Clara
Santa Clara, CA
Tel: 1-408-961-6444
Fax: 1-408-961-6445

加拿大多伦多 Toronto
Mississauga, Ontario, Canada
Tel: 1-905-673-0699
Fax: 1-905-673-6509

亚太地区

亚太总部 **Asia Pacific Office**
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

中国 - 北京
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

中国 - 成都
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

中国 - 重庆
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

中国 - 杭州
Tel: 86-571-2819-3187
Fax: 86-571-2819-3189

中国 - 香港特别行政区
Tel: 852-2401-1200
Fax: 852-2401-3431

中国 - 南京
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

中国 - 青岛
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

中国 - 上海
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

中国 - 沈阳
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

中国 - 深圳
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

中国 - 武汉
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

中国 - 西安
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

中国 - 厦门
Tel: 86-592-238-8138
Fax: 86-592-238-8130

中国 - 珠海
Tel: 86-756-321-0040
Fax: 86-756-321-0049

亚太地区

台湾地区 - 高雄
Tel: 886-7-536-4818
Fax: 886-7-330-9305

台湾地区 - 台北
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

台湾地区 - 新竹
Tel: 886-3-5778-366
Fax: 886-3-5770-955

澳大利亚 Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

印度 India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

印度 India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

印度 India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

日本 Japan - Osaka
Tel: 81-66-152-7160
Fax: 81-66-152-9310

日本 Japan - Yokohama
Tel: 81-45-471-6166
Fax: 81-45-471-6122

韩国 Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

韩国 Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 或
82-2-558-5934

马来西亚 Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

马来西亚 Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

菲律宾 Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

新加坡 Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

泰国 Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

欧洲

奥地利 Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

丹麦 Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

法国 France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

德国 Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

意大利 Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

荷兰 Netherlands - Druenen
Tel: 31-416-690399
Fax: 31-416-690340

西班牙 Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

英国 UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820