

22 SPI

22.1 概述

串行外设接口（Serial Peripheral Interface，SPI）是外部设备通过4线交换数据的串行同步通讯手段。芯片提供了2个SPI接口模块，可配置为主设备或从设备，实现与外部的SPI通信。

特点：

- 全双工4线串行同步收发（SCLK, MOSI, MISO, SSN）
- MISO和MOSI可交换引脚顺序
- 半双工4线串行同步收发（SCLK, SDATA, SSN, DCN）
- 2路独立通道
- 主从模式
- 可编程时钟极性和相位
- 可编程比特速率
- 可编程数据字长（8/16/24/32bits）
- 最大波特率为 $F_{APBCLK}/2$
- 传输结束中断标志
- 写冲突错标志
- 主模式错误检测、保护和中断标志
- 支持DMA

22.2 结构框图

下图为SPI模块的结构示意图。

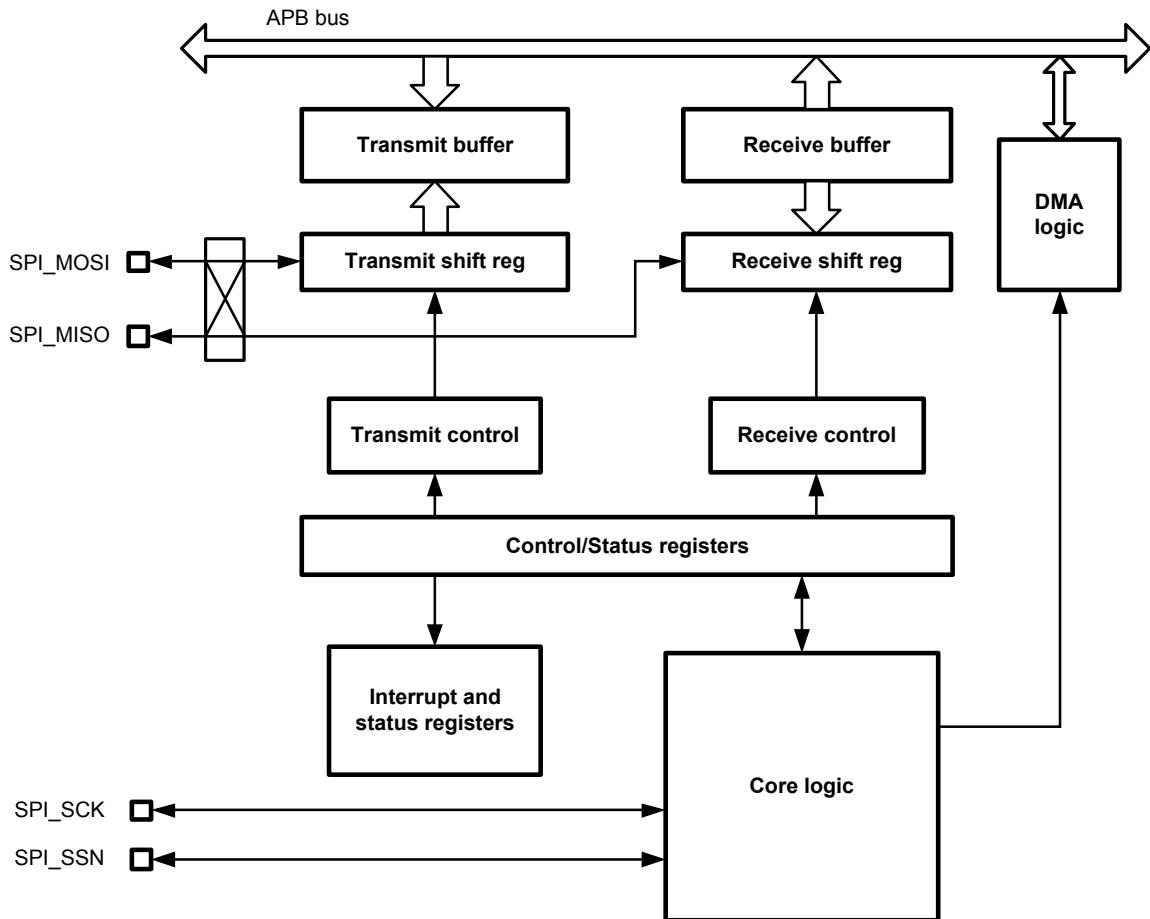


图 22-1 SPI 结构框图

22.3 引脚定义

SPI 模块使用 4 个引脚与外部器件通信，在全双工和半双工模式下，这四个引脚的功能定义有所不同，如下表所示：

引脚	SPIx	全双工	功能	半双工	功能
PB8/PD2	SPI1	SSN	片选信号	SSN	片选信号
PB9/PD3		SCLK	时钟	SCLK	时钟
PB10/PD4		MISO	主机输入从机输出	DCN	命令/数据标识
PB11/PD5		MOSI	主机输出从机输入	SDATA	双向数据
PC7	SPI2	SSN	片选信号	SSN	片选信号
PC8		SCLK	时钟	SCLK	时钟
PC9		MISO	主机输入从机输出	DCN	命令/数据标识
PC10		MOSI	主机输出从机输入	SDATA	双向数据

22.4 接口时序

为了兼容不同的 SPI 外设，SPI 串行时钟的时序可以通过时钟相位选择位（CPHA）和时钟极性选择位（CPOL）设置产生 4 种不同组合。为保证数据正确传输，主从器件的时序配置必需一致。

当处于从器件模式或 SPI 系统使能位（SPE）位为 0 时，SPI 的 SCK 引脚无串行时钟输出。

22.4.1 CPHA=0

CPHA=0 时，SPI 模块在串行时钟的第一个跳变沿采样数据，即：

若 CPOL=1，总线 IDLE 时 SCK 停留在高电平，SPI 在串行时钟的下降沿采样数据，在串行时钟上升沿发送数据；

若 CPOL=0，总线 IDLE 时 SCK 停留在低电平，SPI 在串行时钟的上升沿采样数据，在串行时钟的下降沿发送数据。

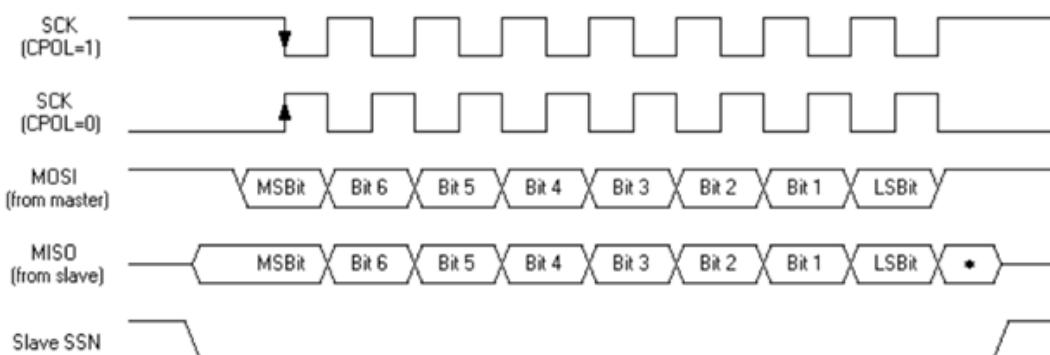


图 22-2 SPI 数据/时钟时序图 (CPHA=0)

22.4.2 CPHA=1

CPHA=1 时，SPI 模块在串行时钟的第二个跳变沿采样数据，即：

若 CPOL=1，总线 IDLE 时 SCK 停留在高电平，在串行时钟的上升沿采样数据，在串行时钟的下降沿发送数据；

若 CPOL=0，总线 IDLE 时 SCK 停留在低电平，在串行时钟的下降沿采样数据，在串行时钟上升沿发送数据。

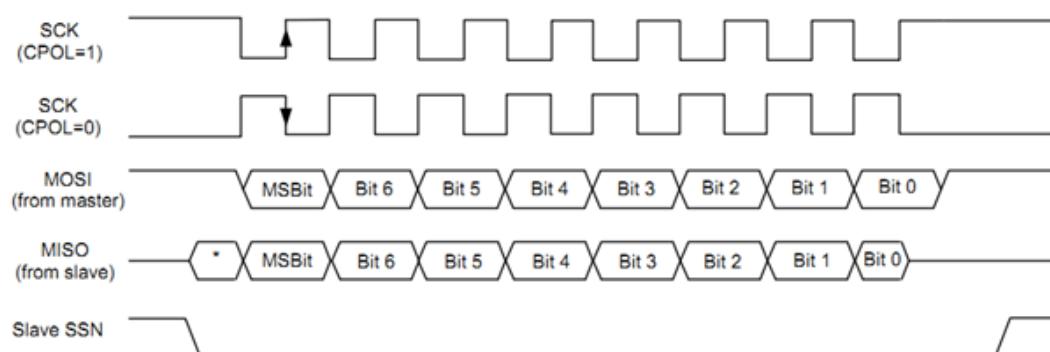


图 22-3 SPI 数据/时钟时序图 (CPHA=1)

22.4.1 4线半双工模式（主机）

4线半双工模式可以支持与点阵液晶或TFT屏的交互通信。在这种模式下，通过DCN信号的高低来区分当前发送的是命令帧还是数据帧。双向数据都通过SDATA（MOSI）引脚收发，由硬件自动完成数据方向切换。FM33LC0XX的SPI仅支持4线半双工主机模式，不支持从机模式。

所有通信都由主机发起，主机首先发送命令帧，然后再进行数据帧传输。命令帧和数据帧通过DCN信号线区分。主机可以通过4线半双工接口向从机写入数据，或从从机读取数据。

4线半双工写操作

软件通过清零HD_RW寄存器，表示当前主机要发起写操作。

主机发起写操作前，首先发送写命令帧。当写命令帧发送完毕后，如果发送缓冲区为空，则硬件将拉高SSN并停止SCLK发送；如果发送缓冲区已经写入了新的数据，则硬件会连续发送后续的数据帧。

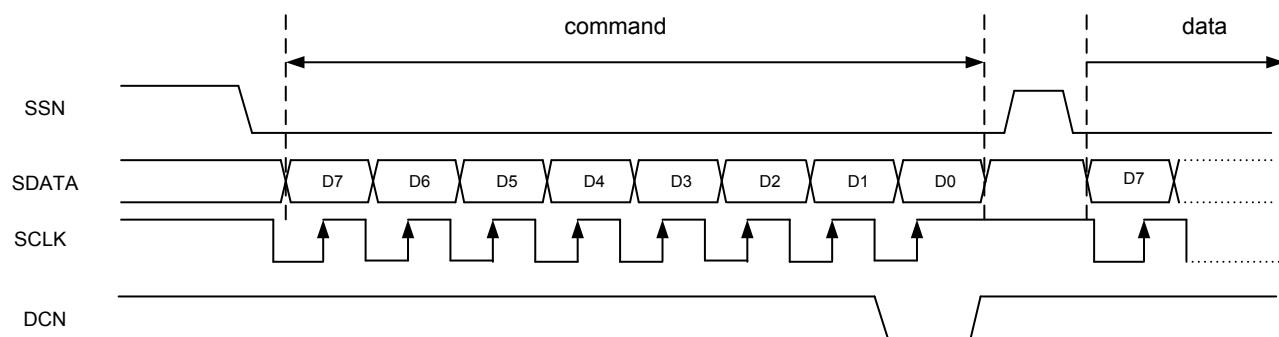


图 22-4 4 线半双工写操作

DCN在第8个时钟上升沿采样判决，如果为0，表示当前帧是命令帧。发送命令帧前，软件需要将DCN寄存器写0，命令帧发送完成后硬件自动将DCN寄存器置位。

4线半双工读操作

软件通过置位HD_RW寄存器，表示当前主机要发起读操作。

4线半双工读操作支持8位、24位和32位读取。主机发起读操作时，首先发送读命令帧。当读命令帧发送完毕后，可以根据寄存器配置发送1个dummy cycle，在dummy cycle期间，SCLK时钟正常发送，但是主机不驱动SDATA，也不接受SDATA输入。

完成命令帧和dummy cycle（可选）后，4线半双工SPI自动进入接收状态，SDATA信号改由从机驱动，主机收到的数据帧将被写入接收缓冲区。每个数据帧接收完成后，将置位RXBF中断标志寄存器。软件应及时读取接收缓冲区中的数据，如果接收缓冲区和接收移位寄存器都处于满状态，硬件会停止SCLK发送，暂停从从机读取数据，直到软件或DMA读取了接收缓冲区。

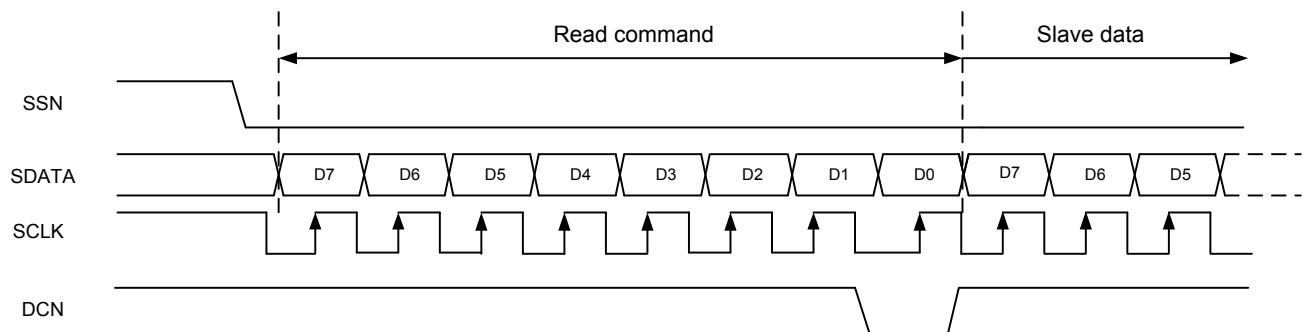


图 22-5 4 线半双工读操作（无 dummy cycle）

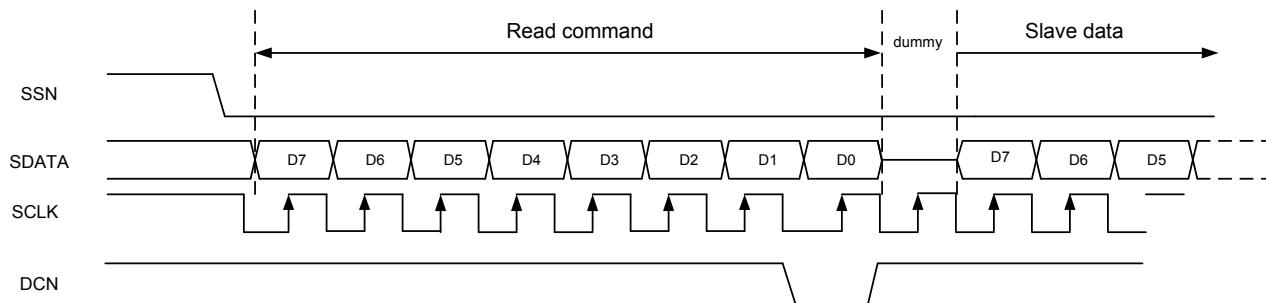


图 22-6 4 线半双工读操作（有 dummy cycle）

22.5 功能描述

22.5.1 I/O 配置

主输出、从输入（MOSI）

主出从入（MOSI）引脚是主器件的输出和从器件的输入，用于主器件到从器件的串行数据传输。当 SPI 配置为主器件时，该引脚为输出，当 SPI 配置为从器件时，该引脚为输入。数据传输时 MSB 在前。

主输入、从输出（MISO）

主入从出（MISO）引脚是从器件的输出和主器件的输入，用于从器件到主器件的串行数据传输。当 SPI 配置为主器件时，该引脚为输入，当 SPI 配置为从器件时，该引脚为输出。数据传输时 MSB 在前。

串行时钟（SCK）

串行时钟（SCK）引脚是主器件的输出和从器件的输入，用于同步主器件和从器件之间在 MOSI 和 MISO 线上的串行数据传输。当 SPI 配置为主器件时，该引脚输出时钟，当 SPI 配置为从器件时，该引脚为输入。

从选择（SSN）

从选择（SSN）引脚用来控制从器件选中，如图 22-2 所示，当 SPI 配置为主器件时，SSN 引脚必须接

高电平，当 SPI 配置为从器件时，SSN 引脚必须接低电平。

SPI 主从器件的连接如错误!未找到引用源。所示：

主从器件的 MOSI、MISO 和 SCK 分别连在一起，主器件的 SSN 必须接高电平，从器件的 SSN 必须接低电平。主从器件通过 MOSI、MISO 连成一个环路，主器件输出时钟，数据传输时，主器件通过 MOSI 输出数据，从器件通过 MISO 输出数据。一字节数据传输完毕，主从器件将交换 8 位移位寄存器数值。

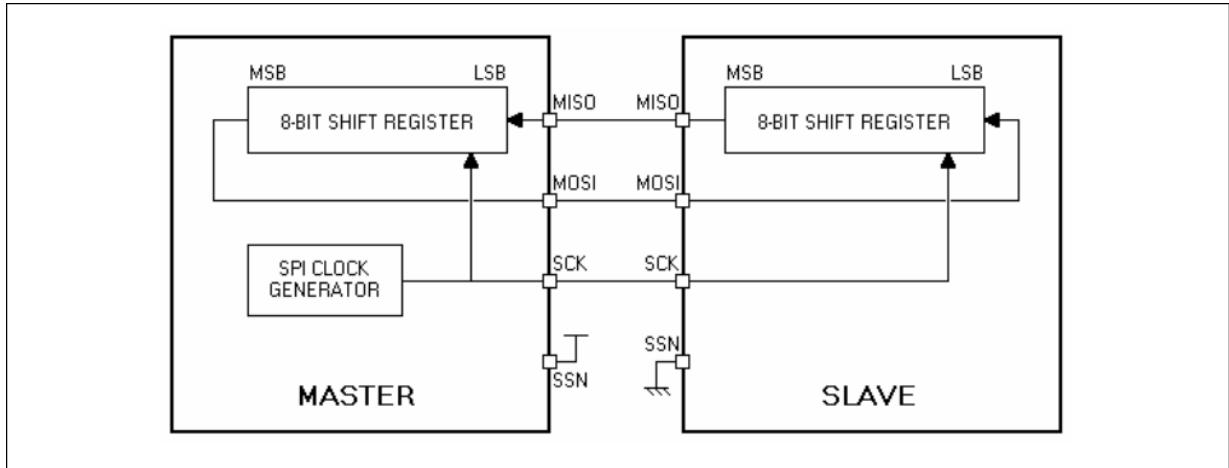


图 22-7 SPI Master/SPI Slave 互连

22.5.2 全双工数据通信

SPI 模块默认为全双工通信，如果需要实现连续不间断的数据通信，软件需要确保 TX BUFFER 非空。即使软件只用 SPI 进行数据接收，由于 SPI 的全双工属性，软件仍需要对 TX BUFFER 进行写操作，此时写入的是无效数据，可根据 MOSI 无效状态配置写入全 0 或全 F。

发送缓冲区

软件或 DMA 将待发送数据写入发送缓冲区（**SPIxTXBUF** 寄存器），当发送开始时，硬件将数据从发送缓冲区拷贝到移位寄存器并开始发送。数据从发送缓冲区转移至移位寄存器后，发送缓存空标志（**TXBE**）被置位，表示可以向 **TXBUF** 写入新数据；如果 **TXIE** 寄存器置位，则产生中断。通过向 **TXBUF** 写入数据，可以清零 **TXBE** 寄存器。

如果在移位寄存器移位完成前，新的数据被写入发送缓冲区，则可以保证连续不断的的数据发送。在 **TXBE** 为 0 的情况下写 **TXBUF**，则会产生数据冲突，参见 22.5.6 数据冲突。

接收缓冲区

当 SPI 完成一帧数据接收后，收到的数据将从移位寄存器拷贝到接收缓冲区（**SPIxRXBUF** 寄存器），同时 **RXBF** 标志被置位，表示 **RXBUF** 中已有数据待处理。如果 **RXIE** 寄存器置位，则产生中断。通过读取 **RXBUF** 可以清零 **RXBF** 标志。

在RXBF没有置位的情况下读RXBUF，将返回上一次接收到的数据；如果应用没有及时处理RXBF，新的数据在RXBF置位的情况下完成接收，则产生数据冲突，参见22.5.6数据冲突。

BUSY标志

当SPI正在进行数据收发时，BUSY寄存器置位。此寄存器在某些场景下可以用来判断最后一帧数据是否传输完毕。比如TXBE只是表示数据已经进入移位发送，但是真正发送完成，需要等待BUSY标志清零。

如何启动SPI通信

主机模式下，建议遵循以下步骤启动SPI通信：

- 应用配置SPI模块
- 置位SPIEN
- 向TXBUF写入数据，SPI模块自动开始发送SCK并进行数据收发

从机模式下，建议应用在主机开始发送SCK之前完成配置和使能，并将第一帧待发送数据写入TXBUF，等待主机发送SCK开始通信。

如何结束SPI通信

主机模式下，建议遵循以下步骤结束SPI通信：

- 等待RXBF和TXBE标志置位，此时移位寄存器中还有最后一帧数据正在发送
- 查询BUSY标志，直到BUSY为0，最后一帧数据收发完成
- 关闭SPI模块，如果需要，读取最后一帧接收数据

从机模式下，应用可以在读取任意一帧数据后关闭SPI模块，关闭前已经被移入移位寄存器的数据将被忽略。

22.5.3 TX-ONLY 模式

某些时候SPI通信是半双工的，在主机仅需进行发送的情况下，通过置位TXO寄存器进入TX-ONLY模式，此时MISO收到的数据不会被写入RX Buffer中，相应的也不会置位RXBF中断标志。

通过置位TXO_AC，可以实现TXO自动清零功能。在TX-ONLY模式下，如果TX buffer空（TXBE置位）并且发送移位寄存器空，则TXO寄存器自动清零，退出TX-ONLY状态。

22.5.4 RX-ONLY 模式

SPI主机仅需进行接收的情况下，通过置位RXO寄存器进入RX-ONLY模式，此时SPI模块无需软件

对TX Buffer进行写操作，即可进行连续不断的数据接收，此时MOSI将保持IDLE电平，并且不会置位TXBE中断标志寄存器。

22.5.5 主机 SSN 控制

SPI模块主机支持硬件或软件控制SSN信号。

当SSNSEN寄存器清零时，SSN由硬件电路控制；如果SSNM寄存器置位，则SPI每发完一帧数据后，将拉高SSN，SSN高电平时间由WAIT寄存器配置（若干个SCK时钟周期）；

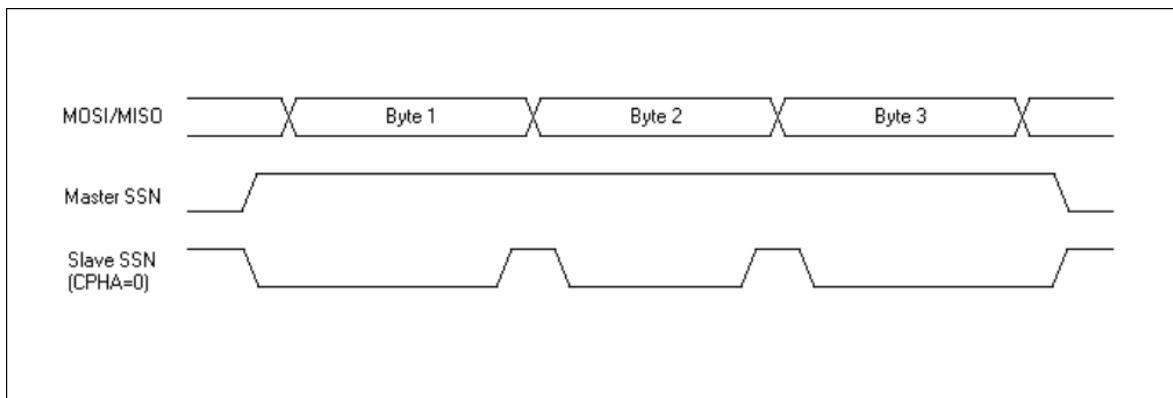


图 22-8 SPI SSN 时序图 (SSNM=1, CPHA=0)

如果SSNM寄存器复位，则SPI每发完一帧数据后不会拉高SSN，而是直接进入下一帧数据发送。

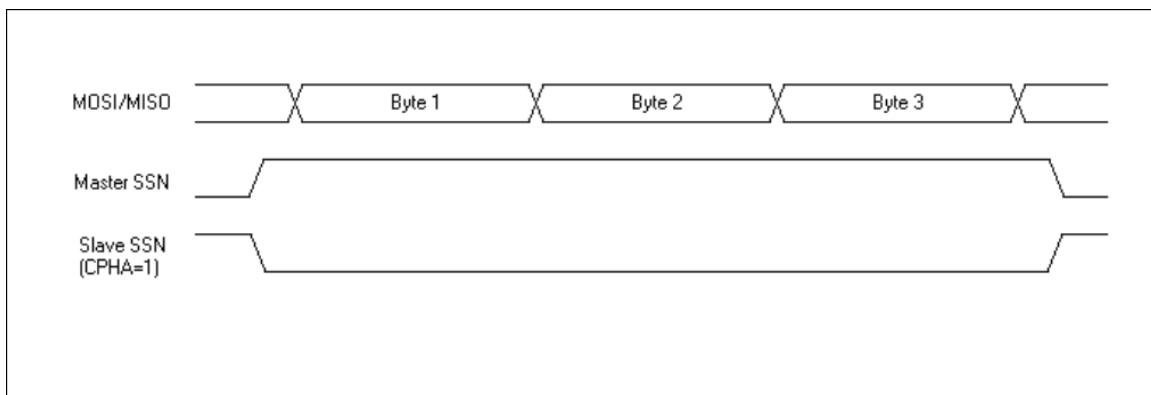


图 22-9 SPI SSN 时序图 (SSNM=0)

当SSNSEN寄存器置位时，SSN直接由软件控制。软件通过写SPIxCR2.SSN寄存器位，可以直接操作SPI主机发送的SSN电平。

22.5.6 数据冲突

当 SPI 的 TX Buffer 数据尚未被读进移位寄存器，或者 SPI 的 RX Buffer 中的数据未被软件或 DMA

读取时，对 TX Buffer 或 RX Buffer 的写操作会产生对应的冲突错误，TXCOL/RXCOL 位会置起，产生中断。导致冲突的写入数据将被忽略。数据冲突错误在主从模式下都会产生。

对 TX Buffer 的写操作，由芯片内部的 Master 模块发起，包括 CPU、DMA 等等。对 RX Buffer 的写操作，则由外部 SPI 器件发起。

当数据冲突发生时，TX Buffer 和 RX Buffer 内原有数据不会被刷新，新写入的数据丢失。

22.5.7 使用 DMA 进行 SPI 收发

当 SPI 模块被使能后，SPI 模块在发送缓冲区空和接收缓冲区满时都会自动产生相应的 DMA 请求。应用软件需要事先配置 DMA 通道连接，将特定通道指向 SPI 外设，设置 RAM 访问的指针地址，并使能 DMA 通道。此后 DMA 会自动响应 SPI 请求，并完成 RAM 和 SPI 之间的数据搬运。

注意：如果使用 DMA 进行收发全双工通信，软件应先使能 DMA 发送通道，再使能 DMA 的接收通道；反之可能会导致 SPI 额外发送一个字节的 **dummy** 数据。

使用 DMA 进行 SPI 接收

- 将 DMA 通道 3 或 5 配置为 SPI_RX
- 设置 RAM 指针地址、地址递增递减、通道优先级、传输长度和中断设置等
- 使能对应 DMA 通道
- 配置 SPI 模块参数
- 使能 SPI 模块，等待数据接收
- 收到数据后 SPI 自动产生 DMA 请求
- DMA 响应请求，读取 SPI 接收缓存寄存器，写入指定 RAM 地址
- 当指定长度的 DMA 传输结束后，DMA 将忽略后续请求并产生传输完成中断，软件应处理中断并关闭 SPI
- 如果关闭 SPI 前又有数据被接收，软件可以通过写 RXBFC 清除 RXBUF

使用 DMA 进行 SPI 发送

DMA发送过程与上述接收过程类似，主要差别是，当指定长度的DMA传输结束后，软件不能立即关闭SPI，因为此时最后一帧数据还在移位发送中，因此软件需要查询BUSY标志直到移位发送结束，再关闭SPI模块。

数据帧长度与 RAM 数据组织方式

SPI 传输帧长度可以配置为 8、16、24、32bit。

当数据帧长度为 8bit 时， DMA 每次搬运 1byte， 4 次搬运填满 RAM 一个地址，字内采用小端存储：
RAM word: { data3, data2, data1, data0 }

当数据帧长度为 16bit 时， DMA 每次搬运 2bytes， 2 次搬运填满 RAM 一个地址，字内采用小端存储：
RAM word: { data1, data0 }

当数据帧长度为 24bit 时， DMA 每次搬运 1 word， 1 次搬运填满 RAM 一个地址，但是有效数据仅占用 RAM 字内低 24bit：

RAM word: { 8'h0, data0 }

当数据帧长度为 32bit 时， DMA 每次搬运 1 word， 1 次搬运填满 RAM 一个地址：

RAM word: { data0 }

22.6 寄存器

offset 地址	名称	符号
SPI1 寄存器(模块起始地址:0x40018C00)		
0x00000000	SPI1 控制寄存器 1 (SPI1 Control Register1)	SPI1_CR1
0x00000004	SPI1 控制寄存器 2 (SPI1 Control Register2)	SPI1_CR2
0x00000008	SPI1 控制寄存器 3 (SPI1 Control Register3)	SPI1_CR3
0x0000000C	SPI1 中断使能寄存器 (SPI1 Interrupt Enable Register)	SPI1_IER
0x00000010	SPI1 中断状态寄存器 (SPI1 Status Register)	SPI1_ISR
0x00000014	SPI1 发送数据缓冲寄存器 (SPI1 Transmit Buffer)	SPI1_TXBUF
0x00000018	SPI1 接收数据缓冲寄存器 (SPI1 Receive Buffer)	SPI1_RXBUF
SPI2 寄存器(模块起始地址:0x40010800)		
0x00000000	SPI2 控制寄存器 1 (SPI2 Control Register1)	SPI2_CR1
0x00000004	SPI2 控制寄存器 2 (SPI2 Control Register2)	SPI2_CR2
0x00000008	SPI2 控制寄存器 3 (SPI2 Control Register3)	SPI2_CR3
0x0000000C	SPI2 中断使能寄存器 (SPI2 Interrupt Enable Register)	SPI2_IER
0x00000010	SPI2 中断状态寄存器 (SPI2 Status Register)	SPI2_ISR
0x00000014	SPI2 发送数据缓冲寄存器 (SPI2 Transmit Buffer)	SPI2_TXBUF
0x00000018	SPI2 接收数据缓冲寄存器 (SPI2 Receive Buffer)	SPI2_RXBUF

22.6.1 SPIx 控制寄存器 1 (SPIx_CR1)

名称	SPIx_CR1(x=1,2)							
Offset	0x00000000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			IOSWAP	MSPA	SSPA	MM	
位权限	U-0			R/W-0	R/W-0	R/W-0	R/W-1	
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	WAIT		BAUD			LSBF	CPOL	CPHA
位权限	R/W-00		R/W-000			R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:12	-	RFU: 未实现, 读为 0
11	IOSWAP	MOSI 和 MISO 引脚交换 (IO swapping) 0: 默认引脚顺序 1: 交换引脚顺序
10	MSPA	Master Sampling Position Adjustment, Master 对 MISO 信号的采样位置调整, 用于高速通信时补偿 PCB 走线延迟 1: 采样点延迟半个 SCK 周期 0: 不调整
9	SSPA	Slave Sending Position Adjustment, Slave MISO 发送位置调整 1: 提前半个 SCK 周期发送 0: 不调整
8	MM	Master/Slave 模式选择。 1: Master 模式 0: Slave 模式
7:6	WAIT	Master 模式下, 每发送完一帧后加入至少(1+WAIT)个 SCK cycle 等待时间, 再传输下一帧的数据。如果 SSN 由硬件控制, 并且 SSNM=1, 则硬件会自动拉高 SSN。
5:3	BAUD	Master 模式波特率配置位: 000: $f_{APBCLK}/2$ 001: $f_{APBCLK}/4$ 010: $f_{APBCLK}/8$ 011: $f_{APBCLK}/16$ 100: $f_{APBCLK}/32$ 101: $f_{APBCLK}/64$ 110: $f_{APBCLK}/128$ 111: $f_{APBCLK}/256$ 当通信正在进行的时候, 不能修改这些位。
2	LSBF	帧格式 (LSB First) 0: 先发送 MSB 1: 先发送 LSB 注: 当通信在进行时不能改变该位的值。
1	CPOL	时钟极性选择 (Clock Polarity) 1: 串行时钟停止在高电平 0: 串行时钟停止在低电平 注: 当通信在进行时不能改变该位的值 注: 当 SSN 为低时不能改变该位的值
0	CPHA	时钟相位选择 (Clock Phase) 1: 第二个时钟边沿是第一个捕捉边沿 0: 第一个时钟边沿是第一个捕捉边沿 注: 当通信在进行时不能改变该位的值。

22.6.2 SPIx 控制寄存器 2 (SPIx_CR2)

名称	SPIx_CR2(x=1,2)							
Offset	0x00000004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							

位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名					-			
位权限					U-0			
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	DUMMY_EN		-		RXO		DLEN	HALFDUPLEX
位权限	R/W-0		U-0		R/W-0		R/W-00	R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	HD_RW	CMD8b	SSNM	TXO_AC	TXO	SSN	SSNSEN	SPIEN
位权限	R/W-0	R/W-1	R/W-0	R/W-1	R/W-0	R/W-1	R/W-0	R/W-0

位号	助记符	功能描述
31:16	-	RFU: 未实现, 读为 0
15	DUMMY_EN	4 线半双工协议下是否在读操作中插入 dummy cycle (Dummy cycle Enable) 0: 不插入 dummy cycle 1: 在读命令之后插入一个 dummy cycle
14:12	-	RFU: 未实现, 读为 0
11	RXO	RXONLY 控制位, 此寄存器置位时, SPI 可以连续接收, 无需软件写 TXBUF (Receive Only mode) 1: 启动 Master 的单接收模式 0: 关闭单接收模式 (收发全双工)
10:9	DLEN	通信数据字长配置 (Data Length) 00: 8bit 01: 16bit 10: 24bit 11: 32bit
8	HALFDUPLEX	通信模式选择 (Half-Duplex mode) 0: 标准 SPI 模式, 4 线全双工 1: DCN 模式, 4 线半双工
7	HD_RW	半双工模式下主机读写操作配置 (Read/Write config for Half-Duplex mode) 0: 4 线半双工协议下主机写入从机 1: 4 线半双工协议下主机读取从机
6	CMD8b	半双工模式下定义 command 帧长度 (Command 8 bits) 1: command 帧固定为 8bit 0: command 帧长度由 DLEN 定义
5	SSNM	Master 模式下 SSN 控制模式选择 (SSN mode) 1: 每发送完一帧后 Master 拉高 SSN, 维持高电平时间由 WAIT 寄存器控制 0: 每发送完一帧后 Master 保持 SSN 为低
4	TXO_AC	TXONLY 硬件自动清空的使能 (TXONLY auto-clear enable) 1: TXONLY 硬件自动清零有效, 软件使能 TXO 后, 等待发送完毕后, 硬件清零 0: 关闭 TXONLY 硬件自动清零
3	TXO	TXONLY 控制位 (Transmit Only mode enable) 1: 启动 Master 的单发送模式 0: 关闭单发送模式 (收发全双工)
2	SSN	Master 模式下, 如果 SSNSEN 为 1, 软件可以通过此位控制 SSN 输出电平

位号	助记符	功能描述
		1: SSN 输出高电平 0: SSN 输出低电平
1	SSNSEN	Master 模式下, 软件控制 SSN 使能 (SSN Software Enable) 1: Master 模式下 SSN 输出由软件控制 0: Master 模式下 SSN 输出由硬件自动控制
0	SPIEN	SPI 使能 (SPI enable) 1: 使能 SPI 0: 关闭 SPI, 清空发送接收缓存

22.6.3 SPIx 控制寄存器 3 (SPIx_CR3)

名称	SPIx_CR3(x=1,2)							
Offset	0x00000008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名					-			
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名					-			
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名					-			
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名					TXBFC	RXBFC	MERRC	SERRC
位权限	U-0				W-0	W-0	W-0	W-0

位号	助记符	功能描述
31:4	-	RFU: 未实现, 读为 0
3	TXBFC	Transmit Buffer Clear, 软件写 1 清除发送缓存, 写 0 无效
2	RXBFC	Receive Buffer Clear, 软件写 1 清除发送缓存, 写 0 无效
1	MERRC	Master Error Clear, 软件写 1 清除 HSPISTA.MERR 寄存器
0	SERRC	Slave Error Clear, 软件写 1 清除 HSPISTA.SERR 寄存器

22.6.4 SPIx 中断控制寄存器 (SPIx_IER)

名称	SPIx_IER(x=1,2)							
Offset	0x0000000C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名					-			
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名					-			
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名					-			
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位名	-	ERRIE	TXIE	RXIE
位权限	U-0	R/W-0	R/W-0	R/W-0

位号	助记符	功能描述
31:3	-	RFU: 未实现, 读为 0
2	ERRIE	SPI 错误中断使能 (Error Interrupt Enable)
1	TXIE	发送完成中断使能 (Transmit Interrupt Enable)
0	RXIE	接收完成中断使能 (Receive Interrupt Enable)

22.6.5 SPIx 中断标志寄存器 (SPIx_ISR)

名称	SPIx_ISR(x=1,2)							
Offset	0x000000010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			DCN_TX	-	RXCOL	TXCOL	BUSY
位权限	U-0			R/W-1	U-0	R/W-0	R/W-0	R-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	MERR	SERR	-			TXBE	RXBF
位权限	U-0	R-0	R-0	U-0			R-1	R-0

位号	助记符	功能描述
31:13	-	RFU: 未实现, 读为 0
12	DCN_TX	半双工模式下 (HALFDUPLEX=1), 配置在每个数据帧的最后 bit 发送的 DCN 信号电平 (Data/Command transmit config) 0: DCN=0, 表示命令帧 1: DCN=1, 表示数据帧 软件应在发送前设置 DCN_TX 寄存器, 如果 DCN_TX=0, 硬件在完成一帧发送后, 自动将 DCN_TX 置 1, 即默认只会发送一个命令帧, 后续都是数据帧。
11	-	RFU: 未实现, 读为 0
10	RXCOL	接收缓存溢出, 软件写 1 清零 (Receive Collision flag, write 1 to flag)
9	TXCOL	发送缓存溢出, 软件写 1 清零 (Transmit Collision flag, write 1 to clear)
8	BUSY	SPI 空闲标志, 只读(busy flag) 1: SPI 传输进行中 0: SPI 传输空闲
7	-	RFU: 未实现, 读为 0
6	MERR	Master Error 标志(Master Error flag) 当 Master 下传输未满 8 位 SSN 就被拉高时, MERR 置位
5	SERR	Slave Error 标志(Slave Error flag) 当 Slave 下传输未满 8 位 SSN 就被拉高时, SERR 置位
4:2	-	RFU: 未实现, 读为 0

位号	助记符	功能描述
1	TXBE	TX Buffer Empty 标志位(TX Buffer Empty flag) 1: 发送缓存空, 软件写 TXBUF 清零 0: 发送缓存满
0	RXBF	RX Buffer Full 标志位(RX Buffer Full flag) 1: 接收缓存满, 软件读 RXBUF 清零 0: 接收缓存空

22.6.6 SPIx 发送缓存寄存器 (SPIx_TXBUF)

名称	SPIx_TXBUF(x=1,2)							
Offset	0x00000014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	TXBUF[31:24]							
位权限	W-0000 0000							
位	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23
位名	TXBUF[23:16]							
位权限	W-0000 0000							
位	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15
位名	TXBUF[15:8]							
位权限	W-0000 0000							
位	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7
位名	TXBUF[7:0]							
位权限	W-0000 0000							

位号	助记符	功能描述
31:0	TXBUF	SPI 发送缓存 (Transmit Buffer)

22.6.7 SPIx 接收缓存寄存器 (SPIx_RXBUF)

名称	SPIx_RXBUF(x=1,2)							
Offset	0x00000018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RXBUF[31:24]							
位权限	R-0000 0000							
位	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23
位名	RXBUF[23:16]							
位权限	R-0000 0000							
位	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15
位名	RXBUF[15:8]							
位权限	R-0000 0000							
位	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7
位名	RXBUF[7:0]							
位权限	R-0000 0000							

位号	助记符	功能描述
31:0	RXBUF	SPI 接收缓存 (Receive Buffer)