

串行外设接口SPI





串行外设接口SPI

在开发F28335时，有时候可能需要扩展一些外围设备，比如觉得F28335内部12位的ADC精度不够，想要外扩一个串行高精度的ADC，或者比如想外扩EEPROM、LCD显示驱动器、网络控制器、DAC等，就需要用到F28335的串行外围设备接口SPI了。SPI是一种高速的同步串行输入输出接口，允许1~16位的数据流在设备与设备之间进行交换，通常用于DSP与外围设备或者DSP与其他控制器之间进行通信。本章首先会介绍SPI接口通用的一些基本知识，然后将详细介绍F28335内部SPI的结构、特点、中断、工作方式等内容。



SPI模块的通用知识

SPI是Serial Peripheral Interface的缩写，翻译成中文就是串行外围设备接口。SPI最早是由Motorola公司在其MC68HCXX系列处理器上定义的一种高速同步串行通信接口。而前一章中所介绍的SCI是一种低速异步串行通信接口，从这一点上就能看出SPI和SCI的区别，SPI是同步通信，SCI是异步通信，那同步通信和异步通信有什么区别呢？最简单的来讲，同步通信时，通信双方的设备必须拥有相同的时钟脉冲，以相同的步调进行数据传输，就像国庆阅兵时，队伍中的官兵在统一的口令下齐步前进，整齐划一。而异步通信时，通信双方的设备可以拥有各自独立的时钟脉冲，可以独自进行数据传输，就像是两个人在散步，可以各走各的。



SPI模块的通用知识

SPI的总线系统可以直接与各个厂家生产的多标准外围器件直接接口，SPI接口一般使用四条线，如表16-1所示。当然，并不是所有的SPI接口都是采用四线制的，有的SPI接口带有中断信号线INT，而有的SPI接口没有主机输出/从机输入线MOSI。在F28335中SPI接口采用的是四线制。

线路名称	线路作用
SCK	串行时钟线
MISO	主机输入/从机输出线
MOSI	主机输出/从机输入线
\overline{CS}	低电平有效的从机选择线

表16-1 SPI接口通用的四条线



SPI模块的通用知识

SPI接口的通信原理很简单，它以主从方式进行工作，这种模式的通信系统中通常有一个主设备和多个从设备。其中，CS信号是用来控制从机的芯片是否被选中的。如图16-1所示，系统内有一个主设备M1和两个从设备S1和S2。当S1的片选信号为低电平时，S1被选中，M1通过MOSI引脚发送数据，S1通过MOSI引脚接收数据，或者S1通过MISO引脚发送数据，而M1通过MISO引脚接收数据。同样的，当S2的片选信号CS为低电平时，S2被选中，M1通过MOSI引脚发送数据，S2通过MOSI引脚接收数据，或者S2通过MISO引脚发送数据，而M1通过MISO引脚接收数据。从机只有通过CS信号被选中之后，对此从机的操作才会有效，可见片选信号的存在使得允许在同一总线上连接多个SPI设备成为可能。



SPI模块的通用知识

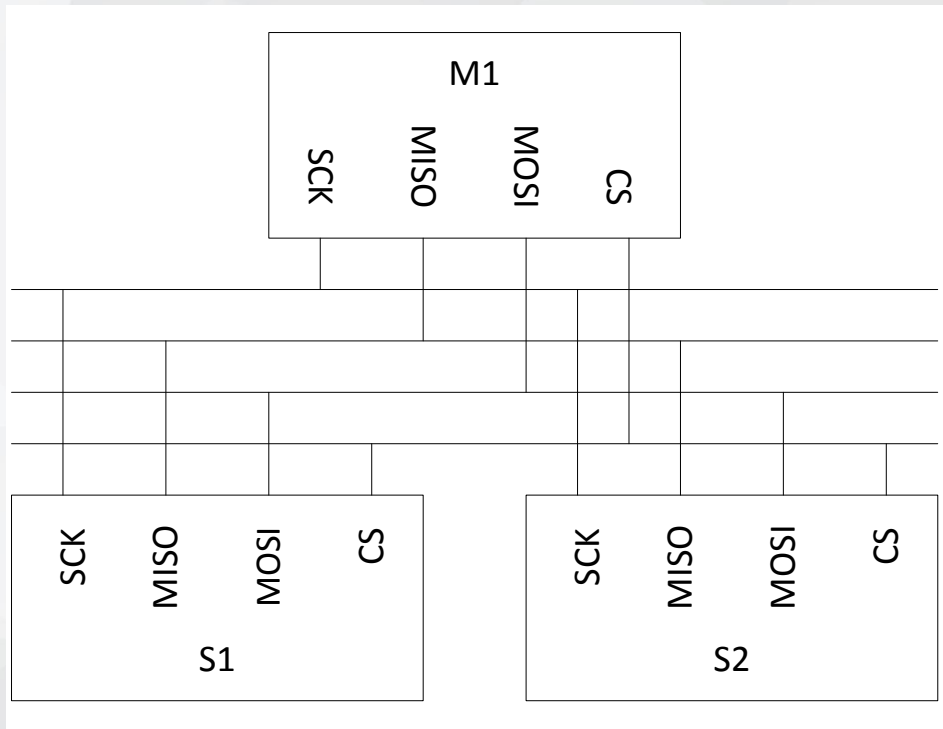


图16-1 SPI主从工作方式示意图



SPI模块的通用知识

当从机被选中，和主机建立连接之后，接下来起作用的就是负责通信的3根线了。通信时通过进行数据交换来完成的，这里首先要知道SPI采用的是串行通信协议，也就是说通信时数据是一位一位进行传输的。这也是SCK时钟信号存在的原因，传输时，由SCK提供时钟脉冲，MOSI和MISO引脚则是基于此脉冲完成数据的发送或者接收。如图16-1所示，当M1给S1发送数据时，数据在时钟脉冲的上升沿或者下降沿时通过M1的MOSI引脚发送，在紧接着的下降沿或者上升沿时通过S1的MOSI引脚接收。当S1给M1发送数据时，原理是一样的，只不过通过MISO引脚来完成。



SPI模块的通用知识

值得注意的是，SCK信号只由主设备控制，从设备不能控制时钟信号线，因此，在一个基于SPI的系统中，必须至少有一个主控设备，其向整个SPI系统提供时钟信号，系统内所有的设备都基于这个时钟脉冲进行数据的接收或者发送，所以SPI是同步串行通信接口。在点对点的通信中，SPI接口不需要寻址操作，且为全双工通信，因此显得简单高效。在多个从设备的系统中，每个从设备都需要独立的使能信号，硬件上比I2C系统要稍微复杂一些。



SPI模块的通用知识

SPI是一个环形总线结构，其时序其实比较简单，主要是在时钟脉冲SCK的控制下，两个双向移位寄存器SPIDAT进行数据交换。假设主机M1和从机S1进行通信，主机的8位寄存器SPIDAT1内的数据是10101010，而从机的8位寄存器SPIDAT2内的数据是01010101，在时钟脉冲上升沿的时候发送数据，在下降沿的时候接收数据，最高位的数据先发送，主机和从机之间进行全双工通信，也就是说两个SPI接口同时发送和接收数据，如图16-2所示。从图16-2可以看到，SPIDAT移位寄存器总是将最高位的数据移出，接着将剩余的数据分别左移一位，然后将接收到的数据移入其最低位。



SPI模块的通用知识

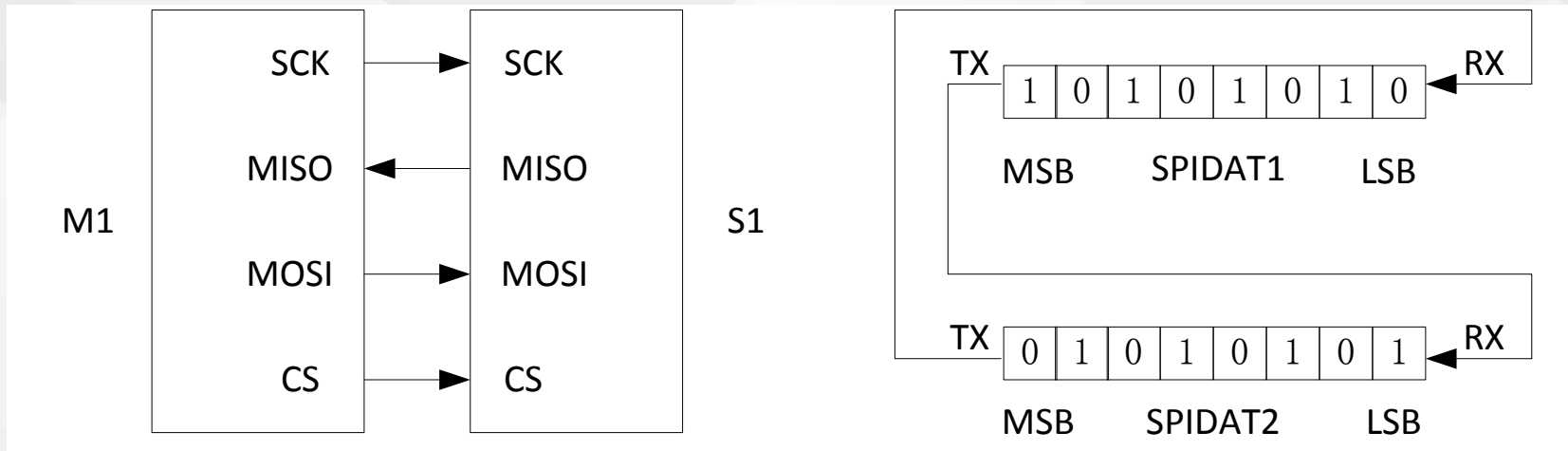


图16-2 SPI的环形总线结构



SPI模块的通用知识

如图16-3所示，当时钟脉冲第一个上升沿来的时候，SPIDAT1将最高位1移出，并将剩余所有的数据左移1位，这时主机的MOSI引脚为高电平，而SPIDAT2将最高位0移出，并将剩余所有的数据左移1位，这时从机的MOSI引脚为低电平。然后，当时钟脉冲下降沿到来的时候，SPIDAT1将锁存主机MISO引脚上的电平，也就是从机发出的低电平，并将数值0移入其最低位，同样的，SPIDAT2将锁存从机MISO引脚上的电平，也就是主机发出的高电平，并将数值1移入其最低位。经过8个时钟脉冲后，两个移位寄存器就实现了数据的交换，也就是完成了一次SPI的时序。



SPI模块的通用知识

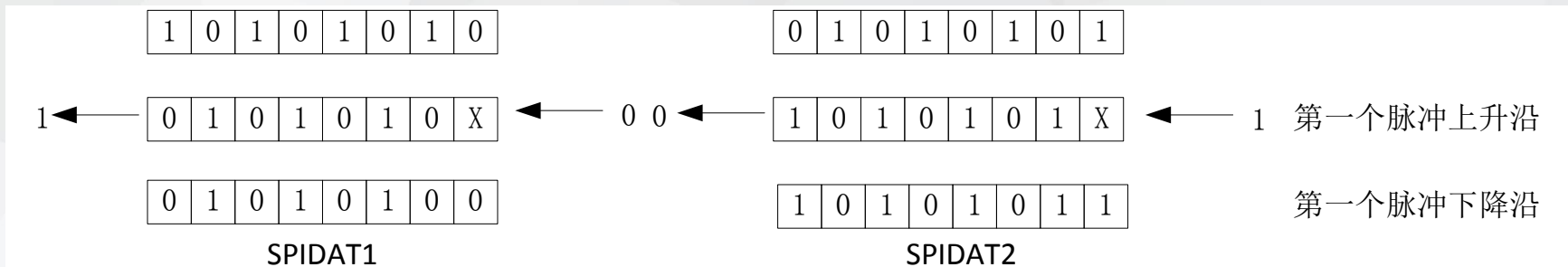


图16-3 SPIDAT数据传输示例

前面只是对SPI接口的基本情况作了介绍，分别讲述了SPI与SCI接口的区别，应用的范围，通信原理等方面的内容，目的是希望能够对SPI接口本身有所了解，因为不仅仅只有DSP才具有SPI接口，很多外围设备同样具有SPI接口。接下来，就要回归到本章的主体部分，开始F28335中SPI接口的介绍。通过后面的学习就会发现，其实SPI核心的知识这里已经提出来了，可以前后对照着学习。



F28335 SPI模块的概述

图16-4是F28335 SPI的CPU接口。

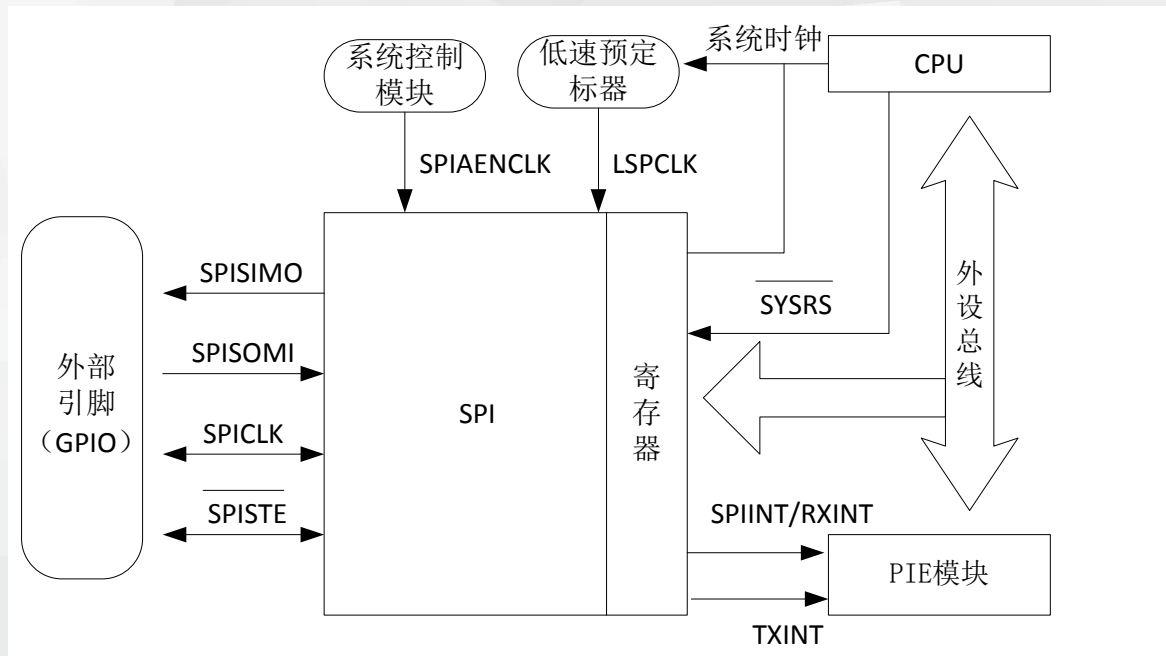


图16-4 F28335 SPI的CPU接口



F28335 SPI模块的概述·SPI模块的特点

图16-3所示的是F28335 SPI模块的接口图，其具有的特点如下：

1.具有4个外部引脚，如表16-2所示。从表16-2可见，F28335采用的是四线制的SPI接口。

引脚	功能说明
SPISOMI	SPI从模式输出/主模式输入引脚
SPISIMO	SPI从模式输入/主模式输出引脚
SPICLK	SPI串行时钟引脚
	SPI从模式发送使能引脚

表16-2 F28335 SPI接口的引脚



F28335 SPI模块的概述·SPI模块的特点

- 2.有两种工作模式可以选择：主工作模式和从工作模式。
- 3.波特率：具有125种可编程的波特率。能够使用的最大波特率受到I/O缓冲器最大缓存速度的限制，这些缓冲器是使用在SPI引脚上的I/O缓冲器，而最高的波特率不能超过LSPCLK/4。
- 4.单次发送的数据字的长度为1~16位，可以通过寄存器设定。
- 5.可选择四种脉冲时钟配置方案，具体的将在后面进行介绍。
- 6.接收和发送可以同步操作，也就是说可以实现全双工通信。当然，发送功能可以通过SPICTL寄存器的TALK位禁止或者使能。
- 7.和SCI相同，发送和接收都能通过查询或者中断方式来实现。



F28335 SPI模块的概述·SPI模块的特点

8.具有6个控制寄存器、3个数据寄存器和3个FIFO寄存器。值得注意的是，SPI所有的控制寄存器都是8位的，当寄存器被访问时，数据位于低8位，而高8位为0，因此把数据写入SPI这6个控制寄存器的高8位是无效的。但是，3个数据寄存器SPIRXBUF、SPITXBUF和SPIDAT都是16位的。3个FIFO寄存器也是16位的。

9.F28335的SPI也具有2个16级的FIFO，一个用于发送数据，一个用于接收数据。发送数据的时候，数据与数据之间的延时可以通过编程进行控制。

10.在标准的SPI模式(非FIFO模式)下，发送中断和接收中断都使用SPIINT/RXINT。在FIFO模式中，接收中断使用SPIINT/RXINT，而发送中断使用的是SPITXINT。



F28335 SPI模块的概述·SPI的信号总结

表16-3为SPI模块信号的功能描述。

信号名称	功能描述
外部引脚	
SPISOMI	SPI从模式输出/主模式输入引脚
SPISIMO	SPI从模式输入/主模式输出引脚
SPICLK	SPI串行时钟引脚
	SPI从模式发送使能引脚
控制信号	
SPI时钟速率	LSPCLK
中断信号	
SPIINT/RXINT	发送中断/接收中断(不使用FIFO情况下)
SPITXINT	发送中断(使用FIFO情况下)

表16-3 SPI信号功能描述



SPI模块的工作原理

图16-5为SPI模块的结构框图。从图中可以看出，SPI能够完成数据的交换主要依赖于3个数据寄存器，接收数据缓冲寄存器SCIRXBUF、发送数据缓冲寄存器SCITXBUF和数据移位寄存器SPIDAT，这3个寄存器均为16位数据寄存器。

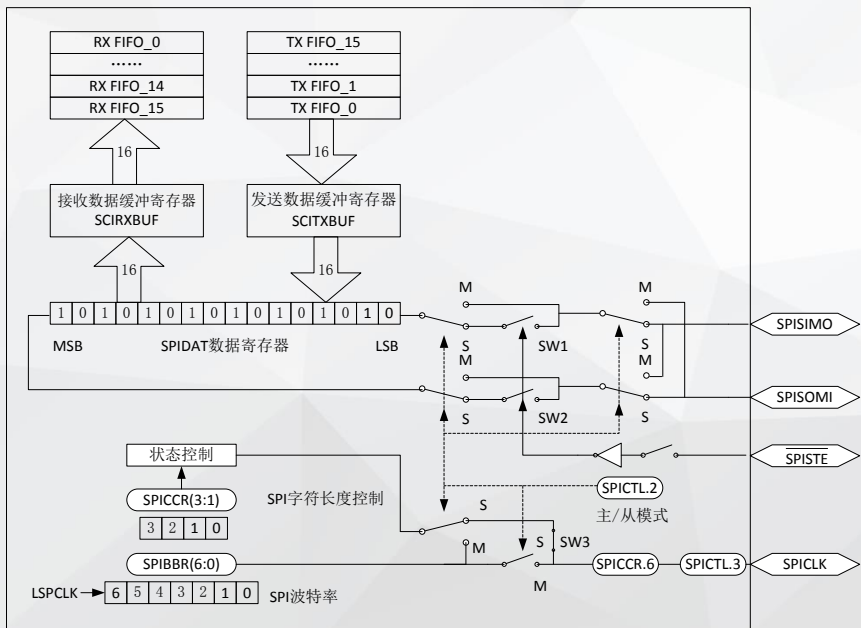


图16-5 SPI模块的结构框图



SPI模块的工作原理

SPI模块可以通过移位寄存器实现数据的交换，即通过SPIDAT寄存器移入或者移出数据，下面简单的来了解一下SPI工作在标准SPI模式下(FIFO未使能)时，数据交换的过程。首先，通过程序向发送缓冲寄存器 SCITXBUF 写入数据，如果此时 SPIDAT 寄存器为空，则 SCITXBUF 将需要发送的完整数据传输给 SPIDAT，数据在 SCITXBUF 寄存器和 SCIDAT 寄存器内存放都是左对齐的，也就是从高位开始存储的。SPIDAT 经过每一个时钟脉冲，完成一位数据的发送或者接收。假设在时钟脉冲的上升沿时，SPIDAT 将数据的最高位发送出去，然后将剩余的所有数据左移 1 位，接下来，在时钟脉冲的下降沿时，SPIDAT 锁存一位数据，并保存至其最低位。当发送完指定位数的数据后，SPIDAT 寄存器将其内部的数据发送给接收缓冲寄存器 SPIRXBUF，等待 CPU 来读取。数据在 SPIRXBUF 中存放是右对齐的，也就是从低位开始存储的。



SPI模块的工作原理

在标准SPI模式下，接收操作支持双缓冲，也就是在新的接收操作启动时，CPU可以暂时不读取SPIRXBUF中接收到的数据，但是在新的接收操作完成之前必须读取SPIRXBUF，否则将会覆盖原来接收到的数据。相同的，发送操作也支持双缓冲功能。



SPI模块的工作原理·SPI主从工作方式

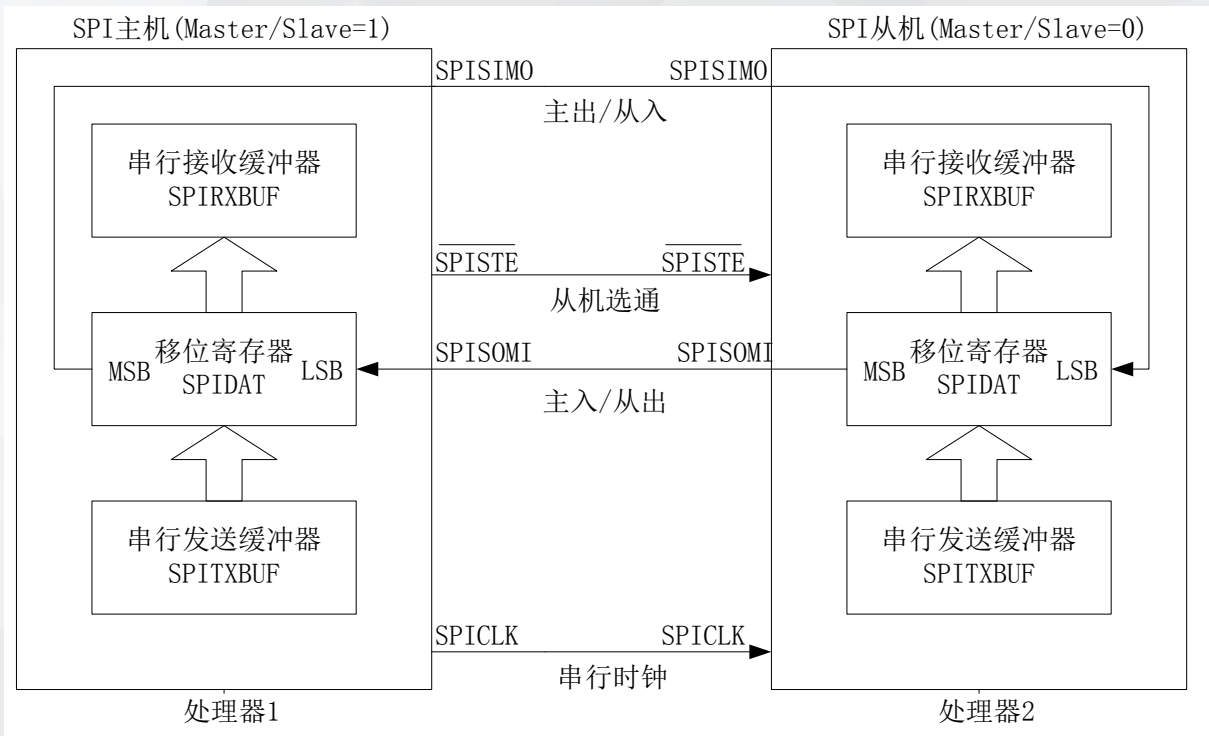


图16-6 SPI主/从模式连接图



SPI模块的工作原理·SPI主从工作方式

图16-6所示的是典型的SPI主/从模式的连接图，系统中有两个处理器，处理器1的SPI工作于主机模式，而处理器2的SPI工作于从机模式。SPI工作控制寄存器SPICTL的MASTER/SLAVE位决定了SPI工作于何种模式，当MASTER/SLAVE=1时，SPI工作于主机模式，而当MASTER/SLAVE=0时，SPI工作于从机模式。从图上也可以看到，时钟信号SPICLK是由主机提供给从机的，主机和从机在SPICLK的协调下同步进行数据的发送或者接收，数据在时钟脉冲信号的上升沿或者下降沿进行发送或者读取。当然，主机和从机之间进行通信的前提是从机片选信号 $\overline{\text{SPISTE}}$ 为低电平，将SPI从机选中，也就是将处理器2选中。主机和从机之间可以同时实现数据的发送和接收，也就是说可以工作于全双工模式。下面将分别详细探讨SPI工作于主机模式和从机模式时的特点。为了能够突出知识点，将采用问答的方式来表达，希望能够帮助对这部分内容的理解。



SPI模块的工作原理·SPI主从工作方式

1.主机模式

(1)问：如何设置SPI成为主机，就像图16-6中的处理器1？

答：通过设置SPI工作控制寄存器SPICTL的MASTER/SLAVE位为1来使得SPI工作于主机模式。编程的语句为：

`SpiaRegs.SPICTL.bit.MASTER_SLAVE=1。`

(2)问：整个SPI的通信网络中的时钟和波特率是由主机来提供的吗？

答：是的。从字面上理解，主机就是在系统中占主导地位的设备，关乎到整个系统的运行。主机通过SPICLK引脚为整个通信网络提供时钟脉冲信号。由于每经过一个时钟脉冲，SPI就完成一位数据的发送，因此时钟脉冲的频率就是通常所说的波特率，其值由主机的SPIBBR寄存器来决定。通过对SPIBBR寄存器的编程，SPI能够实现125种不同的波特率，最大波特率为LSPCLK/4。



SPI模块的工作原理·SPI主从工作方式

(3)问：主机的数据是如何发送和接收的呢？

答：主机通过SPISIMO引脚来发送数据，而通过SPISOMI引脚输入数据。如图16-6所示，当数据写到移位寄存器SPIDAT或者写到串行发送缓冲器SPITXBUF的时候，就会启动SPISIMO引脚开始发送数据，首先发送的是SPIDAT的最高位，接着将剩余的数据左移1位，然后将接收到得数据通过SPISOMI引脚移入SPIDAT的最低有效位。如此重复，当SPIDAT中所要发送的数据都发送出去之后，SPIDAT中接收到得数据被写到SPI的接收缓冲器SPIRXBUF中，等待CPU来读取。从上面的描述，不难理解，为了保证首先发送的是最高位，则发送缓冲器SPITXBUF和移位寄存器SPIDAT中的数据是左对齐的，而由于每次接收到得数据始终是写在最低位，所以接收缓冲寄存器SPIRXBUF中的数据是右对齐的。SPIRXBUF、SPITXBUF、SPIDAT这三个数据寄存器都是16位的。



SPI模块的工作原理·SPI主从工作方式

(4)问：当规定数目的数据通过移位寄存器SPIDAT完成发送时，会产生哪些事件？

答：

·发送了多少位的同时，也相当于接收了多少位的数据，因此当SPIDAT发送完规定数目中的数据时，SPIDAT中也存放了接收到的相同数目的数据，这时候，SPIDAT中接收到得数据会被写到SPIRXBUF中。

·bSPI的中断标志位SPI INT FLAG就会被置位，这时候如果SPIINT/RXINT中断已经被使能，从三级中断的角度来看，也就是SPICTL寄存器的SPINT ENA位被置位，相应的PIE中断被使能，相应的CPU中断已开启，则就会产生SPIINT/RXINT中断。由于SPI的发送和接收时一起完成的，所以这也就是为什么在非FIFO模式下，SPI的发送中断和接收中断使用的是同一个SPIINT/RXINT了。

·c当SPIDAT完成数据发送时，如果SPITXBUF中还有数据，则这些数据将被写入SPIDAT，继续发送。当SPIDAT中所有的数据都发送完成后，时钟脉冲SPICLK将会停止，直到有新的数据写入SPIDAT寄存器进行发送。



SPI模块的工作原理·SPI主从工作方式

(5)问：在数据传输过程和传输完成两种状态时，主机的SPIS_TE引脚有何变化？

答：从前面的学习已经知道， \overline{SPIS} 引脚是从机使能信号，这是一个低电平有效的信号，也就是说当主机需要给从机发送数据的时候， \overline{SPIS} 引脚就被置为低电平，当主机发送完需要发送的数据后， \overline{SPIS} 引脚重新被置为高电平。片选信号的存在使得系统能够同时拥有多个从机，但是在同一时刻，只能有一个从机起作用。



SPI模块的工作原理·SPI主从工作方式

2.从机模式

(1)问：如何设置SPI成为从机，就像图16-6中的处理器2？

答：通过设置SPI工作控制寄存器SPICTL的MASTER/SLAVE位为0来使得SPI工作于从机模式。编程的语句为：

```
SpiaRegs.SPICTL.bit.MASTER_SLAVE=0。
```

(2)问：SPI从机的时钟是由谁决定的？

答：前面已经讲到过，SPI系统通信的时钟是由主机来决定的，也就是说从机通过SPICLK引脚来接受主机提供的串行移位时钟。从机SPICLK引脚的输入频率应不大于LSPCLK/4。



SPI模块的工作原理·SPI主从工作方式

(3)问：从机的数据是如何接收和发送的呢？

答：这个和主机的数据传输机制其实是类似的，首先，从机数据是通过SPISOMI引脚来发送的，而通过SPISIMO引脚来接收的。当从机接收到来自于主机脉冲信号的边沿时，就可以启动数据的发送和接收了。当数据写入SPIDAT或者SPITXBUF后，SPIDAT就开始将数据的最高位移出，同时左移剩下的数据，然后将接收到的数据移入SPIDAT的最低位。在这里还需要探讨一下数据写入到SCITXBUF时的情况，如果数据写到SCITXBUF时，SPIDAT内有数据正在发送，这时SPITXBUF就得等待，等到SPIDAT中数据发送完成后再把SPITXBUF中的数据写入SPIDAT，而如果数据写到SCITXBUF时，SPIDAT没有数据在发送，则这些数据会被立马写入SPIDAT寄存器。



SPI模块的工作原理·SPI主从工作方式

(4)问：由于从机通常是接收功能用的比较多，那如何禁止SPI的发送功能？

答：可以通过设置SPICTL寄存器的TALK位来禁止SPI的发送功能，编程语句为：

```
SpiaRegs.SPICTL.bit.TALK=0。
```

当发送功能被禁止后，发送引脚SPISOMI就会被置为高阻态。如果在禁止发送功能的时候，还有数据正在被发送，则得等到数据被发送完成之后，SPISOMI引脚才会被置为高阻态，这样可以保证SPI能够正确的接收数据。

通过前面的介绍，应该对标准SPI模式下SPI模块的工作原理和运行情况有所了解了，值得提醒的是，请千万不要在通信期间去改变SPI的配置。



SPI模块的工作原理·SPI数据格式

F28335的SPI通过对配置控制寄存器SPICCR的第3位至第0位的选择，可以实现1~16位数据的传输。当每次传输的数据少于16位时，需要注意以下几点：

- (1) 当数据写入SPITXBUF和SPIDAT寄存器时，必须左对齐；
- (2) 当数据从SPIRXBUF寄存器读取时，必须右对齐；
- (3) SPIRXBUF寄存器中存放的是最新接收到的数据，数据采用右对齐方式，再加上前面移位到左边后留下的位。

假设SPIDAT寄存器当前的值为737BH，发送数据的长度为1位，则SPIDATA和SPIRXBUF在发送前后的状态如图16-7所示。



SPI模块的工作原理·SPI数据格式



图16-7 SPIDATA和SPIRXBUF寄存器数据移动方式



SPI模块的工作原理·SPI波特率

SPI通过对寄存器SPIBRR的配置，可以实现125种不同的波特率，计算公式如下：

当SPIBRR=0，1，2时：

$$\text{SPIBaudRate} = \frac{\text{LSPCLK}}{4} \quad (16-1)$$

当SPIBRR=3~127时：

$$\text{SPIBaudRate} = \frac{\text{LSPCLK}}{\text{SPIBRR} + 1} \quad (16-2)$$



SPI模块的工作原理·SPI波特率

式(16-1)和式(16-2)中的LSPCLK为DSP的低速外设时钟频率。从上面的波特率计算公式可以看出，SPI模块最大的波特率为LSPCLK/4。从式(16-2)可以看出，当SBPIBRR为奇数时，(SPIBRR+1)为偶数，SPICLK信号高电平与低电平在一个周期内保持对称。当SPIBRR为偶数时，(SPIBRR+1)为奇数，SPICLK信号高电平和低电平在一个周期内不对称。当时钟极性位被清零时，SPICLK的低电平比高电平多一个系统时钟周期；当时钟极性被置位时，SPICLK的高电平比低电平多一个系统时钟周期。当SPIBRR=0, 1, 2, 3时，SPICLK如图16-8所示。当SPIBRR=4，时钟极性被置位时，SPICLK如图16-9所示。



SPI模块的工作原理·SPI波特率

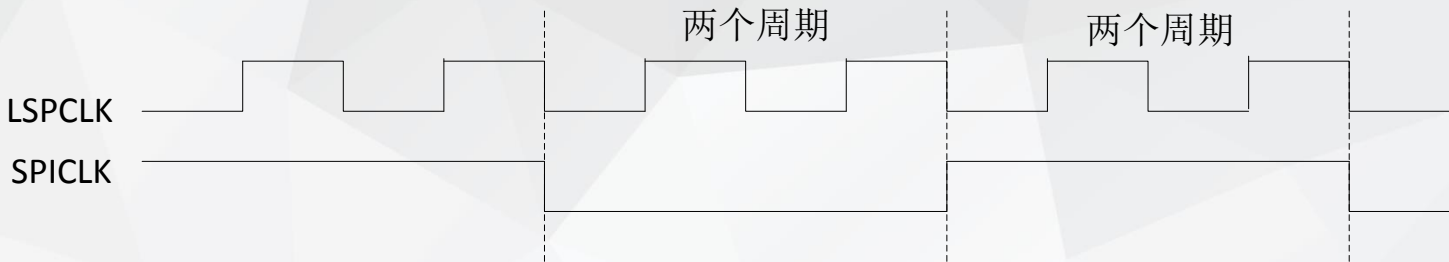


图16-8 当SPIBRR=0 , 1 , 2 , 3时 , SPICLK特性图

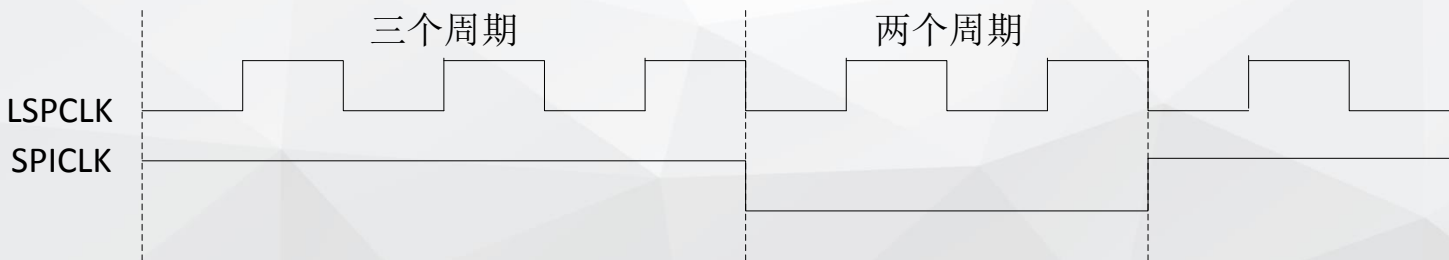


图16-9 当SPIBRR=4 , 时钟极性被置位时 , SPICLK特性图



SPI模块的工作原理·SPI时钟配置

SPI时钟配置方案是指SPI在时钟脉冲的什么时刻去发送或者接收数据。寄存器SPICCR的CLOCK POLARITY位和寄存器SPICTL的CLOCK PHASE位决定了SPI的时钟特性，前面的CLOCK POLARITY决定了时钟的极性，而后面的CLOCK PHASE决定了时钟的相位。两个参数不同取值的组合可以构成4种不同的时钟方案，如图16-10所示，每一种时钟方案都会对数据传输带来影响。



SPI模块的工作原理·SPI时钟配置

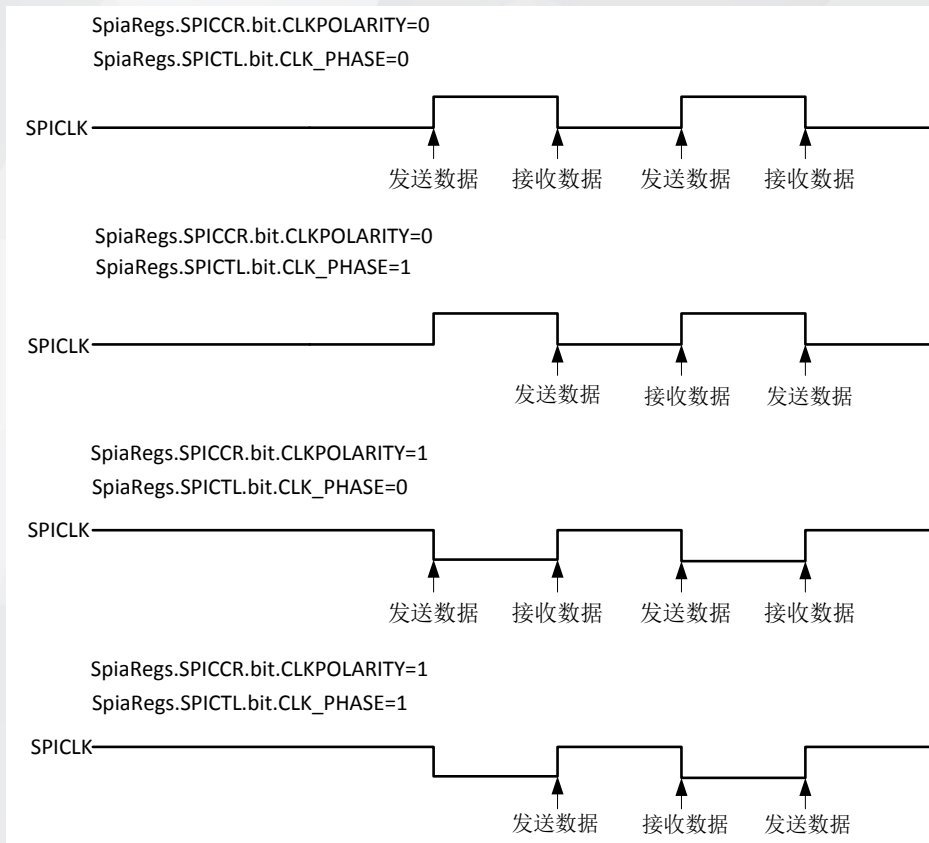


图16-10 SPI时钟配置方案



SPI模块的工作原理·SPI时钟配置

1.当CLOCK POLARITY=0时，当SPICLK没有数据发送时，SPICLK处于低电平，这时候：

(1)当CLOCK PHASE=0时，SPI在SPICLK信号的上升沿发送数据，在SPICLK信号的下降沿接收数据；

(2)当CLOCK PHASE=1时，SPI在SPICLK信号的上升沿延时了半个周期后发送，在随后的上升沿处接收数据。

2.当CLOCK POLARITY=1时，当SPICLK没有数据发送时，SPICLK处于高电平，这时候：

(1)当CLOCK PHASE=0时，SPI在SPICLK信号的下降沿发送数据，在SPICLK信号的上升沿接收数据；

(2)当CLOCK PHASE=1时，SPI在SPICLK信号的下降沿延时了半个周期后发送，在随后的下降沿处接收数据。



SPI模块的工作原理·SPI的FIFO队列

和SCI一样，F28335的SPI也具有16级深度的发送FIFO和接收FIFO。当FIFO功能未被使能时，SPI工作于标准SPI模式；当FIFO功能被使能时，SPI工作于增强的FIFO模式。FIFO的功能由3个寄存器来设置，它们分别是：SPI FIFO发送寄存器SPIFFTX、SPI FIFO接收寄存器SPIFFRX、SPI FIFO控制寄存器SPIFFCT。

当DSP复位时，SPI工作在标准SPI模式下，FIFO功能被禁止。通过将SPIFFTX寄存器中的SPIFFEN位置位来启动SPI的FIFO功能。将SPIFFTX的位SPIRST置1，可以在任何状态下复位FIFO模式，SPI FIFO将重新开始发送和接收数据。

SPI具有1个16*16b的发送缓冲器和1个16*16b的接收缓冲器，标准SPI模式下的发送缓冲器SPITXBUF将作为发送FIFO和移位寄存器SPIDAT之间的一个发送缓冲器。当最后一位数据从移位寄存器SPIDAT移出后，SPITXBUF将重新从FIFO装载数据。



SPI模块的工作原理·SPI的FIFO队列

数据从FIFO转移到移位寄存器的速度是可编程的。SPIFFCT寄存器的第7位到第0位，即FFTXDLY定义了两个数据发送间的延时。这个延时是以SPI串行时钟周期SPICLK为基准的。这个8位寄存器定义可以定义最小0个时钟周期的延时和最大256个时钟周期的延时。当延时为0个时钟周期时，SPI模块能够连续发送数据。当延时为256个时钟周期时，SPI模块发送数据将产生最大延时。这种可编程的特点，使得SPI接口可以更方便的与许多传输速率较慢的外设如EEPROM、ADC、DAC等之间进行通信。



SPI模块的工作原理·SPI的FIFO队列

发送和接收FIFO都有状态位TXFFST和RXFFST。TXFFST位于寄存器SPIFFTX[12:8]，共5位。RXFFST位于寄存器SPIFFRX[12:8]，共5位。这两位的作用是在任何时间可以标识FIFO队列中有效数据的个数。当TXFFST被清零时，发送FIFO队列的复位位TXFIFO RESET也被清零，发送FIFO的指针复位为0，可以通过将TXFIFO RESET置位来重新启动FIFO队列的发送操作。同样的，当RXFFST被清零时，接收FIFO队列的复位位RXFIFO RESET也被清零，接收FIFO的指针复位为0，可以通过将RXFIFO RESET置位来重新启动FIFO队列的接收操作。



SPI模块的工作原理·SPI的中断

图16-11是SPI中断标志和中断使能逻辑汇总。从图16-11可以看到，SPI工作于标准SPI模式下时，能够产生接收溢出中断RX_OVRN INT和发送或接收操作的中断SPIINT，这两个中断共用中断线SPIRXINT。SPI工作于FIFO模式下时，能够产生接收中断SPIRXINT和发送中断SPITXINT。下面进行一一介绍。

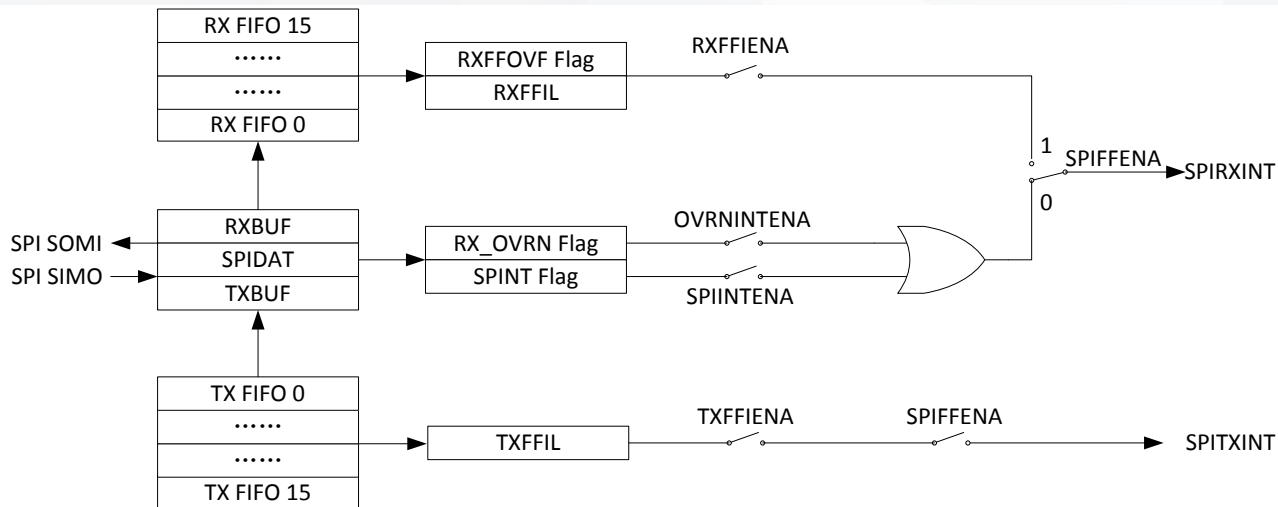


图16-11 SPI中断标志和中断使能逻辑汇总



SPI模块的工作原理·SPI的中断

1.在标准SPI模式下

如图16-11所示，当SPIFFTX寄存器的SPIFFENA位为0时，也就是FIFO功能未使能时，SPI工作于标准SPI模式。当一个完整的字符移入或者移出SPIDAT时，SPIRXINT的中断标志位SPIINT FLAG被置位，此时，SPIDAT中接收到的数据就会被写入SPIRXBUF缓冲寄存器，等待CPU读取。如果SPI工作控制寄存器的位SPI INT ENA被置位，也就是SPIRXINT中断被使能，则SPI将向PIE控制寄存器提出中断请求。



SPI模块的工作原理·SPI的中断

SPIRXINT也是一种复用的中断，当SPI接收数据产生溢出时，也会产生SPIRXINT的中断请求信号。如果新的接收数据写入SPIRXBUF寄存器之前，旧的数据CPU还尚未读取，那么新的数据写入之后就丢失了旧的数据，这时候接收溢出标志位RX_OVRN FLAG被置位，如果SPICTL寄存器的OVERRUN INT ENA位被置位，也就是接收溢出中断被使能，则SPI也将向PIE控制寄存器提出中断请求。

无论是接收溢出，还是接收完成或者发送完成，所产生的中断都使用SPIRXINT中断线。当CPU读取SPIRXBUF寄存器中的数据后，中断标志位SPI INT FLAG会自动被清除。



SPI模块的工作原理·SPI的中断

2.在FIFO模式下

如图16-11所示，当SPIFFTX寄存器的SPIFFENA位为1时，也就是FIFO功能被使能时，SPI工作于增强的FIFO模式。对于接收操作，前面已经介绍过，接收FIFO队列有状态位RXFFST，表示接收FIFO中有多少个接收到的数据。同时，SPI FIFO接收寄存器SPIFFRX还有一个可编程的中断触发级位RXFFIL。当RXFFST的值与预设好的RXFFIL相等时，接收FIFO就会产生接收中断SPIRXINT信号，如果SPIFFRX寄存器的位RXFFIENA为1，也就是FIFO接收中断已经使能，那么SPI将向PIE控制器提出中断请求。比如，假设通过编程，将RXFFIL位设置为8，那么当FIFO队列中接收到8个数据时，RXFFST的值也为8，正好和RXFFIL的值相等，这时候接收FIFO就产生了接收中断匹配事件。复位后，接收FIFO的中断触发级位RXFFIL默认的值为0x1111，即16，也就是说FIFO队列中接收到16个数据的时候产生接收中断请求。



SPI模块的工作原理·SPI的中断

对于发送操作，发送FIFO队列有状态位TXFFST，表示发送FIFO中有多少个数据需要发送。同时SPI FIFO发送寄存器SPIFFTX也有一个可编程的中断触发级位TXFFIL。当TXFFST的值与预设好的TXFFIL相等时，发送FIFO就会产生发送中断SPITXINT信号，如果SPIFFTX寄存器的位TXFFIENA为1，也就是FIFO发送中断已经使能，那么SPI将向PIE控制器提出中断请求。比如，假设通过编程，将TXFFIL位设置为8，那么当FIFO队列中还剩8个数据需要发送时，TXFFST的值也为8，正好和TXFFIL的值相等，这时候发送FIFO就产生了发送中断匹配事件。复位后，发送FIFO的中断触发级位TXFFIL默认值为0x0000，即0，也就是说FIFO队列中数据全部发送完毕后产生发送中断请求。

综上所述，SPI的中断如表16-4所示。



SPI模块的工作原理·SPI的中断

工作模式	SPI中断源	中断标志位	中断使能位	SPIFFENA	中断线
标准SPI模式	接收溢出	RX_OVRN	OVRNINTENA	0	SPIRXINT
	接收数据	SPIINT	SPIINTENA	0	SPIRXINT
	发送空	SPIINT	SPIINTENA	0	SPIRXINT
	FIFO接收中断	RXFFIL	RXFFIENA	1	SPIRXINT
	FIFO发送中断	TXFFIL	TXFFIENA	1	SPITXINT

表16-4 SPI的中断



SPI模块的工作原理·SPI模块的寄存器

SPI模块具有6个控制寄存器、3个数据寄存器、3个FIFO寄存器，如表16-5所示。

寄存器名	地址范围	尺寸(*16)	说明
SPICCR	0x0000 7040	1	SPI配置控制寄存器
SPICTL1	0x0000 7041	1	SPI工作控制寄存器
SPIST	0x0000 7042	1	SPI状态寄存器
SPIBRR	0x0000 7044	1	SPI波特率寄存器
SPIEMU	0x0000 7046	1	SPI仿真缓冲寄存器
SPIRXBUF	0x0000 7047	1	SPI接收数据缓冲寄存器
SPI TXBUF	0x0000 7048	1	SPI发送数据缓冲寄存器
SPIDAT	0x0000 7049	1	SPI数据移位寄存器
SPIFFTX	0x0000 704A	1	SPI FIFO发送寄存器
SPIFFRX	0x0000 704B	1	SPI FIFO接收寄存器
SPIFFCT	0x0000 704C	1	SPI FIFO控制寄存器
SPIPRI	0x0000 704F	1	SPI优先级控制寄存器

SPI寄存器的具体定义可见C2000助手。



SPI模块的工作原理·SPI模块的寄存器

本章首先介绍了SPI接口通用的一些知识，了解了SPI接口的基本工作原理，然后以F28335内部SPI模块为核心，详细介绍了SPI接口的结构、特点、工作方式、数据格式、波特率设置、时钟方案、FIFO队列、中断等内容。下一章，将详细介绍F28335增强型的CAN总线。