

## 29 串行外设接口 (SPI)

### 29.1 描述

串行外设接口 (SPI) 电路提供一个可在主/从模式下与外部设备通信的同步数据链路。如果系统连接了一个外部处理器则还可以实现处理器间的通信。

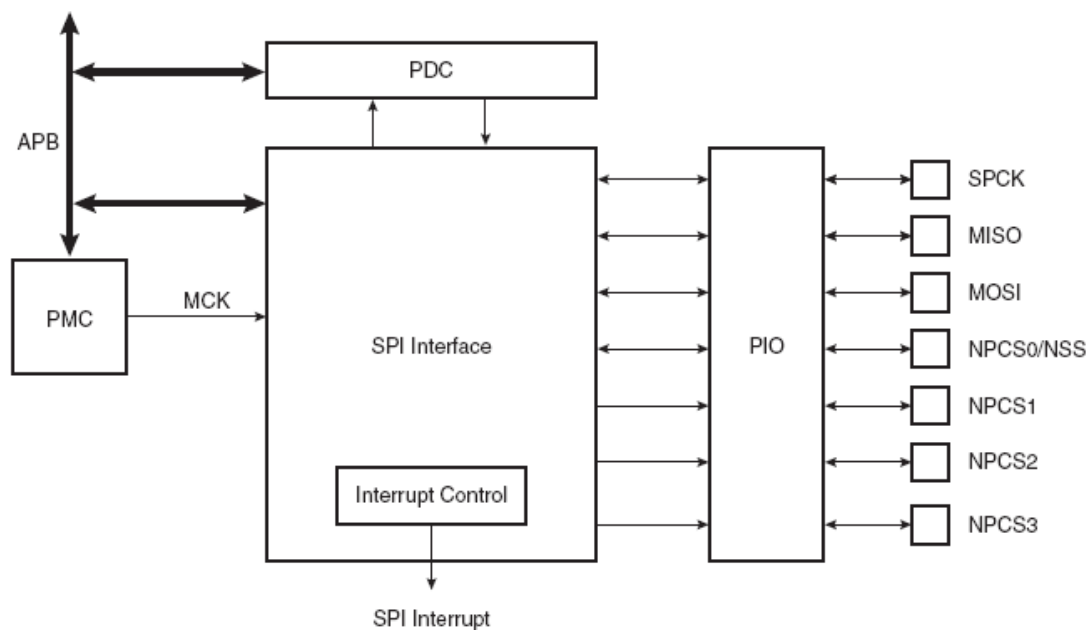
串行外设接口 (SPI) 本质上是一个串行传送数据位到其他 SPI 的移位寄存器。在一个数据传送期间，一个 SPI 系统作为控制数据流的“主控”运行，而其他设备作为通过主控将数据移入或移出的“从设备”在运行。不同的 CPU 可以轮流作为主控（多主控协议和单主控协议相反，单主控协议中一个 CPU 始终是主控而其他的始终是从设备）并且一个主控可同时将数据移入多个从设备。然而，仅单个从设备可以在任何给定时间驱动其输出来将数据写回到主控。

SPI 系统由两个数据口线和两个控制口线组成：

- 主控输出从设备输入 (MOSI)：此数据口线用来将主控输出的数据移入从设备。
- 主控输入从设备输出 (MISO)：此数据口线用来将从设备输出的数据移入主控。在任何特定的传送期间只能有单个从设备在发送数据。
- 串行时钟 (SPCK)：此控制口线由主控驱动，校验数据位流。主控可以以一种波特率发送数据；SPCK 口线对已发送的每位循环一次。
- 从设备选择 (NSS)：此控制口线允许由硬件开启和关断

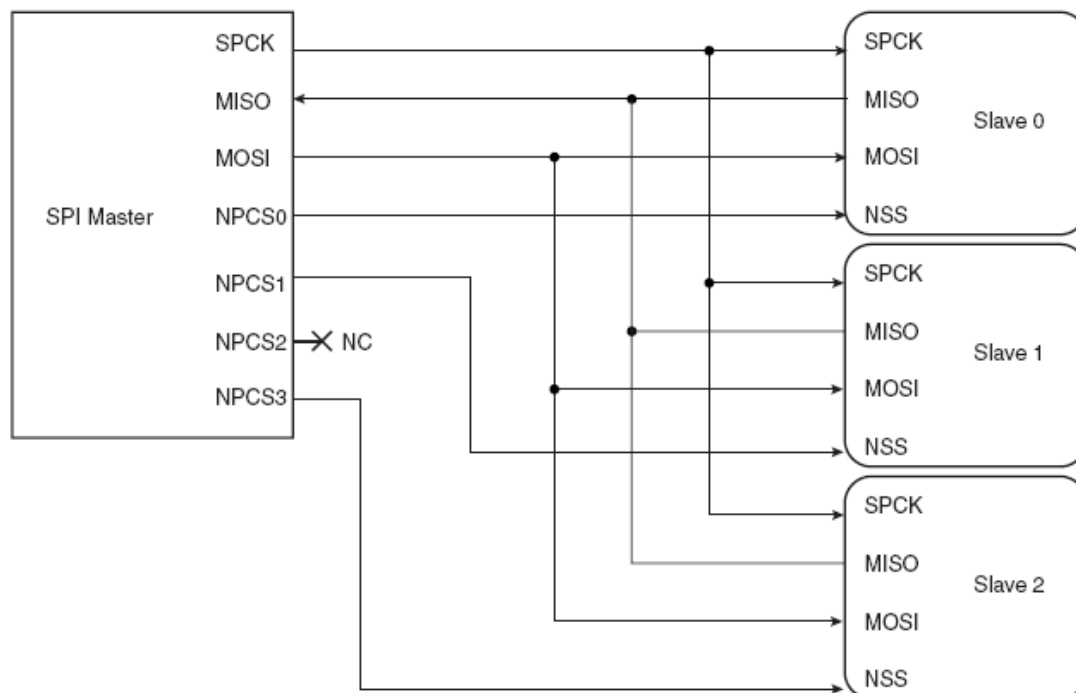
### 29.2 方块图

图 29-1 方块图



### 29.3 应用方块图

图 29-2 应用方块图：单主控/多从设备的实现



## 29.4 信号描述

引脚名称	引脚描述	类型	
		主控	从设备
MISO	主控输入从设备输出	输入	输出
MOSI	主控输出从设备输入	输出	输入
SPCK	串行时钟	输出	输入
NPCS1-NPCS3	外设芯片选择	输出	未用
NPCS0/NSS	外设芯片选择/从设备选择	输出	输入

## 29.5 产品依赖性

### 29.5.1 I/O 口线

用于连接相应外部设备的引脚可与 **PIO** 口线复用。编程者必须首先编程 **PIO** 控制器来分配 **SPI** 引脚为其外设功能。

### 29.5.2 电源管理

**SPI** 可通过电源管理控制器 (PMC) 同步，因此编程者必须首先配置 **PMC** 来使能 **SPI** 时钟。

### 29.5.3 中断

**SPI** 接口有一个连接于高级中断控制器 (AIC) 的中断口线。处理 **SPI** 中断需要在配置 **SPI** 前编程 **SIC**。

## 29.6 功能描述

### 29.6.1 操作模式

**SPI** 操作在主/从模式下。

通过向模式寄存器中的 **MSTR** 位写 1 来编程主控模式下的操作。配置引脚 **NPCS0** 到 **NPCS3** 为输出，驱动 **SPCK** 引脚，**MISO** 口线连接于接收器输入并且被发送器驱动为输出。如果对 **MSTR** 位写 0，则 **SPI** 操作在从模式。发送器输出驱动 **MISO** 口线，**MOSI** 口线连接于接收器输入，发送器驱动 **SPCK** 引脚来同步接收器。**NPCS0** 变为输入，并被用作从设备选择信号 (**NSS**)。引脚 **NPCS1** 到

NPCS3 未被驱动而可被用作其他用途。

数据传输同样可对两种操作模式编程。仅在主控模式激活波特率发生器。

### 29.6.2 数据传输

对数据传输有四个极性和相位的组合可用。用片选寄存器中的 CPOL 位编程时钟极性。用 NCPHA 位编程时钟相位。这两个参数决定已驱动和采样数据的时钟信号边沿。每两个参数有两个可能的状态，产生四个互相兼容的可能的组合。因此，一个主/从对必须用同样的参数对值来通信。若使用多个从设备并且固定为不同的配置，则主控每次需要和不同从设备通信时必须重新配置主控自身。

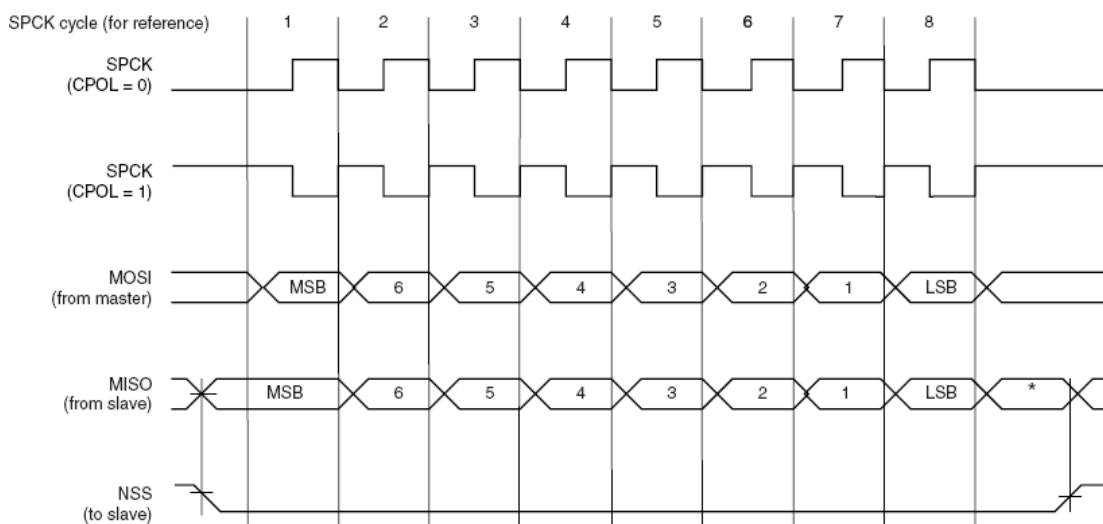
表 29-2 展示了四个模式和对应参数设置。

表 29-2 SPI 总线协议模式

SPI 模式	CPOL	NCPHA
0	0	1
1	0	0
2	1	1
3	1	0

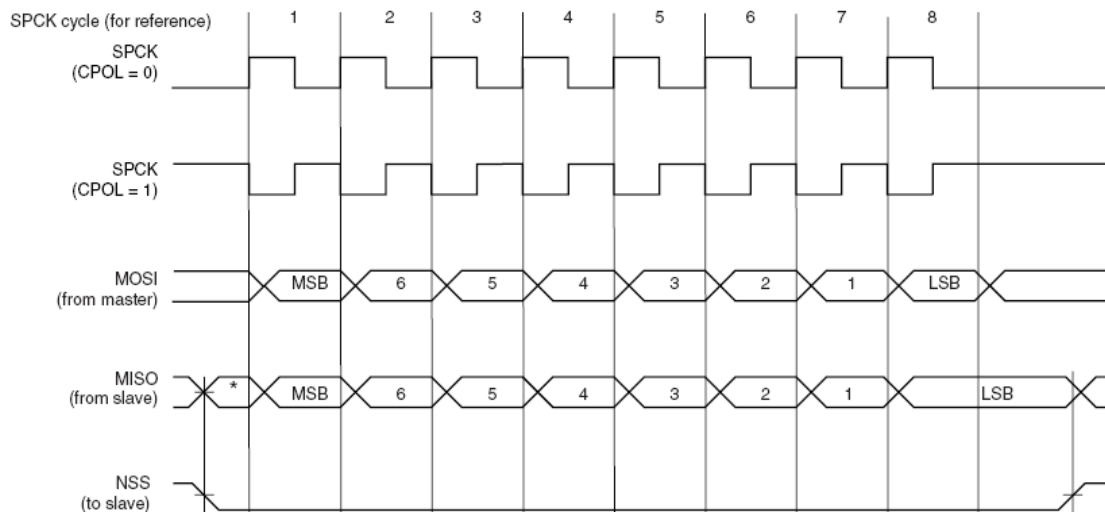
图 29-3 和图 29-4 为数据传输举例

图 29-3 SPI 传输格式 (NCPHA = 1, 每次传送 8 位)



\*未定义，但通常已接收前一字符的 MSB

图 29-4 SPI 传输格式 (NCPHA = 0, 每次传送 8 位)



\*未定义，但通常已发送前一字符的 LSB

### 29.6.3 主控模式操作

当配置为主控模式，SPI 运行于由内部可编程波特率发生器产生的时钟。完全控制来自和到达连接于 SPI 总线的从设备的数据传输。SPI 驱动片选口线到从设备和串行时钟信号 (SPCK)。

SPI 有两个保持寄存器，发送数据寄存器和接收数据寄存器还有一个单移位寄存器。保持寄存器以一个固定速率维持数据流。

使能 SPI 后，当处理器写入 SPI\_TDR，开始传输数据。已写数据立即被传输到移位寄存器并且启动 SPI 总线传送。当移位寄存器中的数据被转移到 MOSI 口线上，则 MISO 口线被采样和转移到移位寄存器。无接收则不能产生发送。

写 TDR 前，必须置位 PCS 域来选择一个从设备。

若在传输过程中有新数据写入 SPI\_TDR，继续保存直到当前传输完成。接着，接收到的数据从移位寄存器传输到 SPI\_RDR，SPI\_TDR 中的数据被装载到移位寄存器并且一次新的传输开始。

通过状态寄存器中 (SPI\_SR) 的 TDRE 位 (发送数据寄存器空) 指示移位寄存器中写入 SPI\_TDR 的数据的传输。当新数据被写入 SPI\_TDR，此位将被清零。TDRE 位被用来触发发送 PDC 通道。

通过 SPI\_SR 寄存器中的 TXEMPTY 标志指示传输结束。若对传输延迟比上次 (DLYBCT)，则在上述延迟完成后置位 TXEMPTY。此时可关断主控时钟 (MCK)。

通过状态寄存器 (SPI\_SR) 中的 RDRF 位 (接收数据寄存器满) 指示从 SPI\_RDR 中移位寄存器接收到的数据的传输。当读取接收到的数据时，清零 RDRF 位。

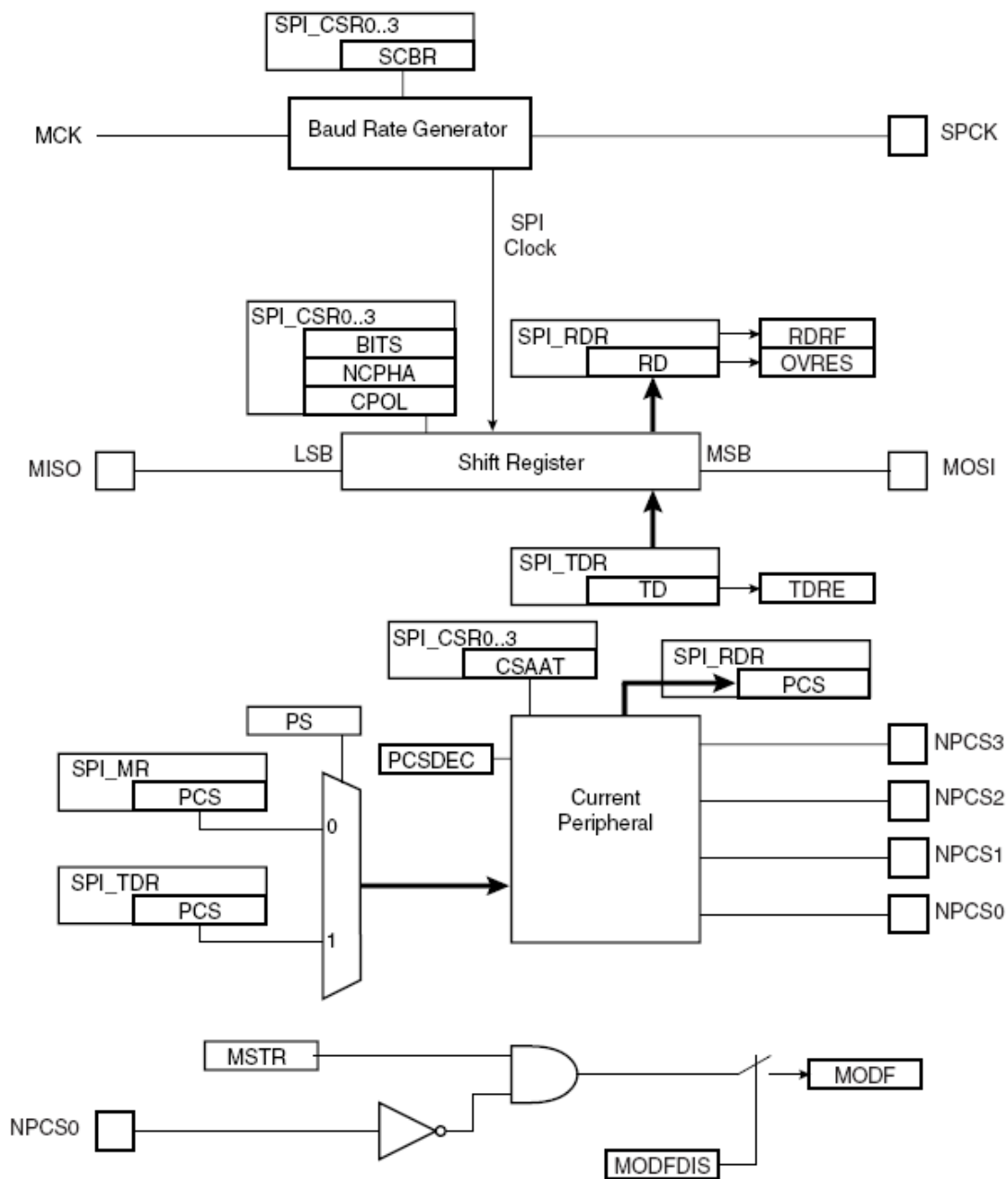
若 SPI\_RDR (接收数据寄存器) 在新数据被接收到前未被读取，则置位 SPI\_SR 中的超限错误位 (OVRES)。只要置位此标志位，则将数据装载到 SPI\_RDR 中。用户必须读取状态寄存器来清零 OVRES 位。

362 页图 29-6 显示了当运行于主控模式时 SPI 的方块图

362 页图 29-6 显示了描述怎样处理传输的流程图

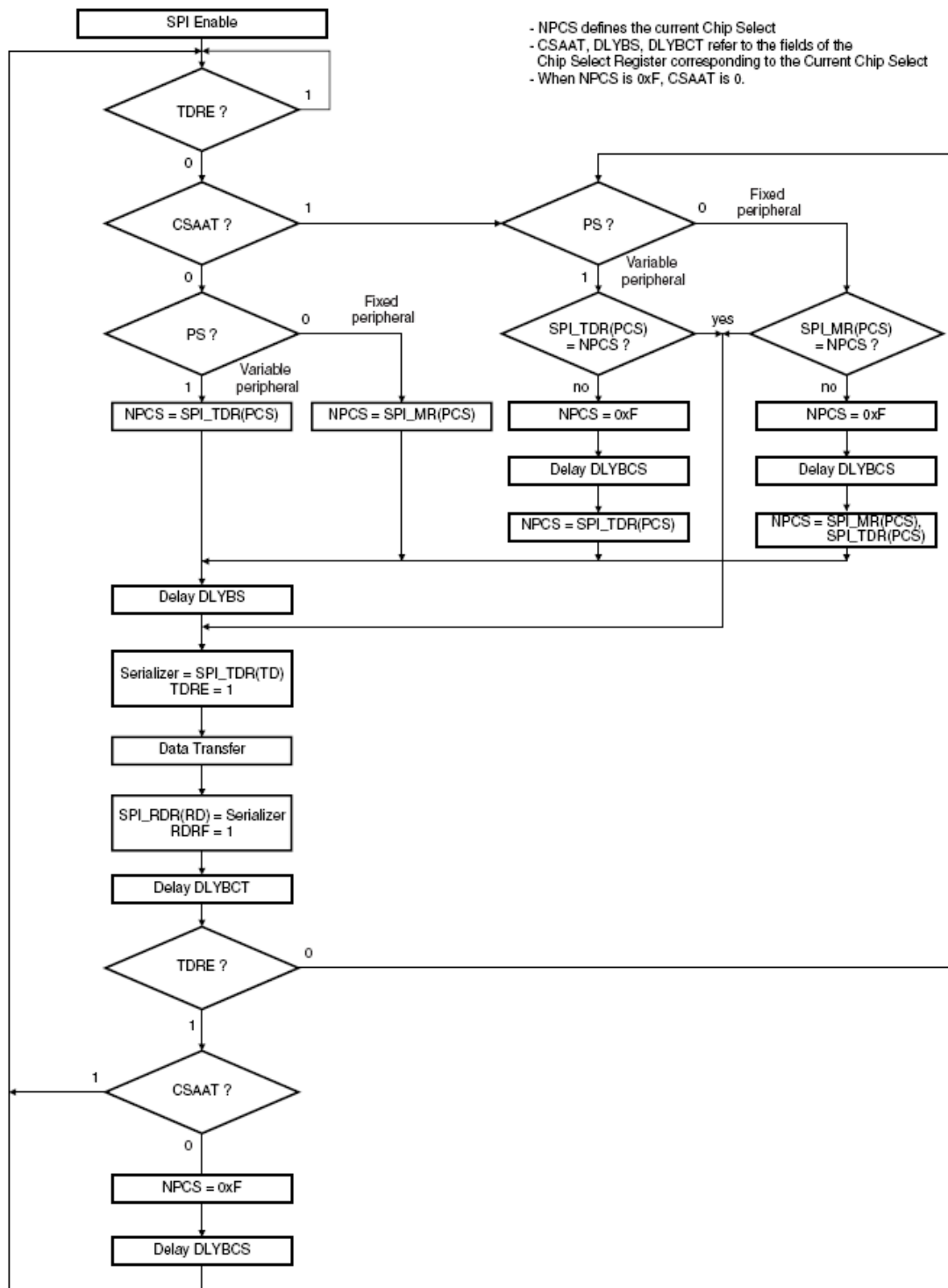
#### 29.6.3.1 主控模式方块图

图 29-5 主控模式方块图



29.6.3.2 主控模式流程图

图 29-6 主控模式流程图



### 29.6.3.3 时钟的产生

SPI 波特率时钟由主控时钟 (MCK) 分频产生，被 1 到 255 之间的值分频。这允许以最大可达主控时钟的波特率运行，以及被 255 分频的 MCK 的最小波特率运行。

禁止编程 SCBR 域为 0。当 SCBR 为 0 时触发传输可导致不可预期的结果。复位时，SCBR 为 0，用户必须在执行首次传输前以一个有效值编程它。

对每个片选可设置独立的分频器，并必须在片选寄存器的 SCBR 域中编程。这允许 SPI 在未编程的情形下对每个连接的外设自动适应波特率。

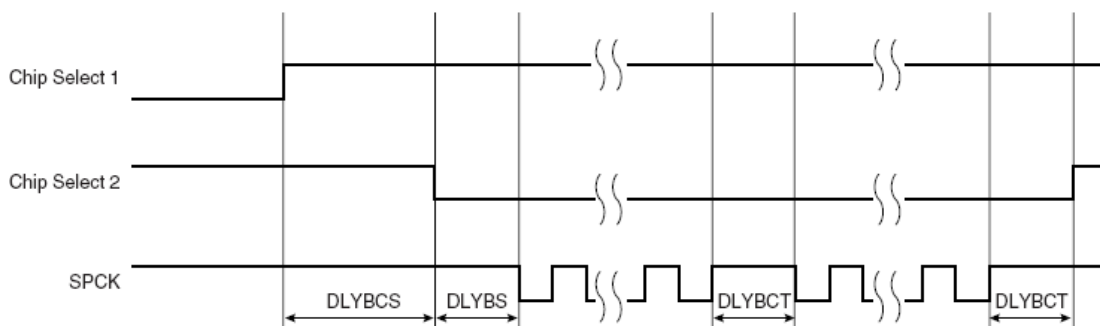
#### 29.6.3.4 传输延迟

图 29-7 显示了片选传输变化和相同片选上的连续传输。可编程三个延迟来修改传输波形：

- 片选间的延迟，仅可对所有片选通过写模式寄存器中的 DLYBCS 域编程一次。允许在一个片选的释放和一个新片选的激活之间插入一个延迟。
- SPCK 前的延迟，可对每个片选通过写 DLYBS 域独立编程。允许在片选激活后启动 SPCK 延迟
- 连续传输间的延迟，可对每个片选通过写 DLYBCT 域独立编程。允许在同一片选上的产生的两次传输间插入延迟。

这些延迟允许 SPI 适应已连接的外设和其速率还有总线释放时间。

图 29-7 可编程延迟



#### 29.6.3.5 外设选择

通过激活 NPCS0 到 NPCS3 信号选择串行外设。通过缺省，每次传输前后所有的 NPCS 信号都是高电平。

可用两种不同的方式执行外设选择：

- 固定外设选择：SPI 仅用一个外设交换数据
- 可变外设选择：可用多个外设交换数据

通过在 SPI\_MR (模式寄存器) 中对 PS 位写 0 来激活固定外设选择。此情形下，通过 SPI\_MR 中的 PCS 域定义当前外设而 SPI\_TDR 中的 PCS 域无效。

通过置位 PS 位激活可变外设选择。SPI\_TDR 中的 PCS 域用来选择当前外设。这就是说可对每个新数据定义外设选择。

固定外设选择允许缓冲器用一个单外设传输。使用 PDC 是最佳的方法，因为存储器和 SPI 间的数据传输的大小为 8 位或 16 位。然而，改变外设选择需要编程模式寄存器。

可变外设选择在未编程模式寄存器情形下允许缓冲器用多个外设传输。写入 SPI\_TDR 中的数据是 32 位宽并定义待发送的实数据和已定义的外设。此模式下使用 PDC 需要 32 位宽的缓冲器，LSB 中的数据，MSB 中的 PCS 和 LASTXFER 域，然而 SPI 仍然用片选配置寄存器通过 MISO 和 MOSI 口线控制传输位数 (8 到 16)。这对缓冲器的存储容量来说不是最佳方法，但是它提供一个非常有效的方法在无何处理器参与的情况下同若干外设交换数据。

#### 29.6.3.6 外设片选解码

用户可通过用外部逻辑解码四个片选口线，NPCS0 到 NPCS3 编程 SPI 运行多达 15 个外设。可通过对模式寄存器 (SPI\_MR) 中的 PCSDEC 位写 1 使能此功能。

当未解码运行时，SPI 确保在任何情况下仅一个片选口线被激活，即每次

驱动为低电平。

当解码运行时，SPI 直接输出由模式寄存器或发送数据寄存器（取决于 PS）中的 PCS 域定义的值。

因为 SPI 在片选口线上设置一个缺省的值 0xF（即所有片选口线为 1）

当未处理任何传输时，仅可解码 15 个外设。

SPI 仅有四个片选寄存器，不是 15 个。所以，当激活解码时，每个片选定义达四个外设的特性。例如，SPI\_CR0 定义外部解码的外设 0-3 的特性，对应 PCS 值 0x0 到 0x3。因此，用户必须确保在解码的片选口线 0 到 3，4 到 7，8 到 11 和 12 到 14 连接兼容的外设。

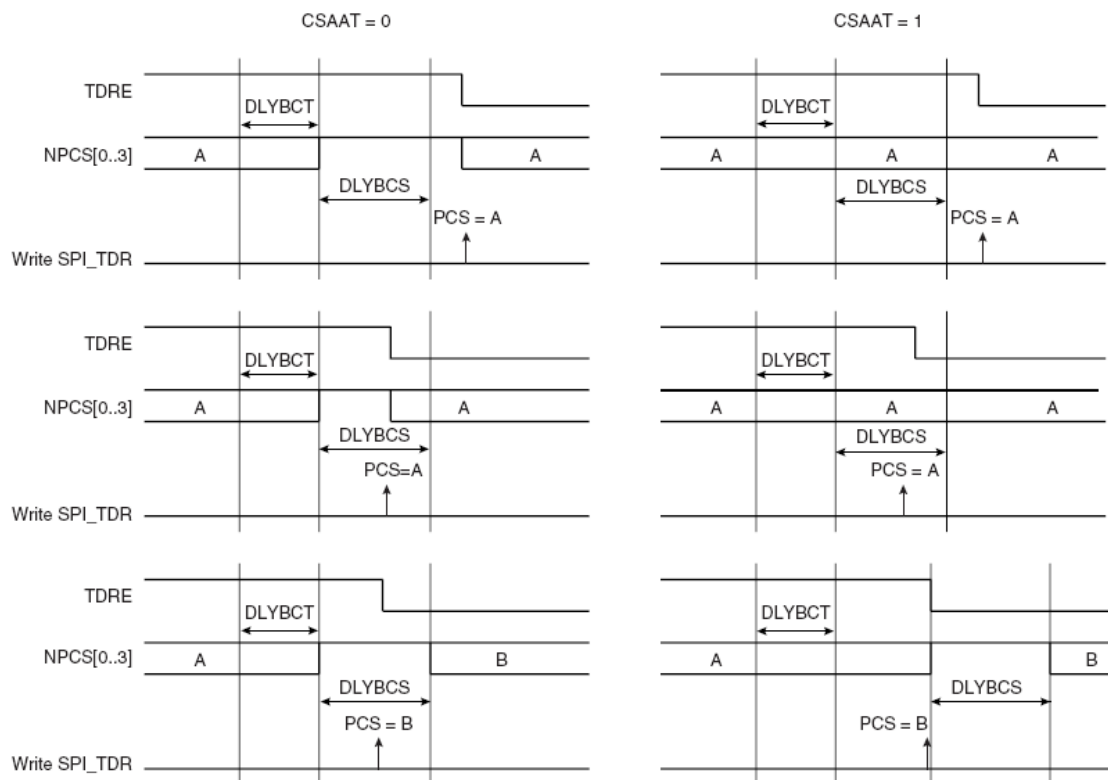
### 29.6.3.7 外设取消选择

当正常运行时，只要上次写入 SPI\_TDR 的数据的传输已完成，则 NPCS 口线都升高。若处理器响应中断太长则可能导致例程错误，并因此在满组传输期间可能导致连接若干串行外设困难，需要片选口线维持有效。

为方便和这样的设备的连接，可用 CSAAT 位（传输后片选有效）置 1 编程片选寄存器。这允许片选口线维持其当前状态（低电平有效）直到需要传输到另一个外设。

图 29-8 显示了不同外设取消选定的情形和 CSAAT 位的结果

图 29-8 外设取消选定



### 29.6.3.8 模式错误检测

当 SPI 被编程为主控模式并且 NPCS0/NSS 信号上被外部主控拉为低电平时检测到模式错误。NPCS0, MOSI, MISO 和 SPCK 必须通过 PIO 控制器配置为开漏，所以需要保证外部上拉电阻外高电平。

当检测到一个模式错误，置位 SPI\_SR 中的 MODF 位直到读取 SPI\_SR，自动禁用 SPI 直到通过对 SPI\_CR 中的 SPIEN 位写 1 重新被使能。

通过缺省，使能模式错误检测电路。用户可通过置位 SPI 模式寄存器

(SPI\_MR) 中的 MODFDIS 位禁用模式错误检测。

#### 29.6.4 SPI 从模式

当运行在从模式下，SPI 处理提供给 SPI 时钟引脚 (SPCK) 上时钟上的数据位。

SPI 等待 NSS 在从外部主控接收串行时钟前激活。当 NSS 下降，时钟在并串行转换器上生效，并串行转换器处理由片选寄存器 0 (SPI\_CSR0) 的 BITS 域定义的位数。处理了这些位紧接着处理由 SPI\_CSR0 的 NCPHA 和 CPOL 位分别定义的一个相位和极性。注意当编程 SPI 为从模式时，其他片选寄存器的 BITS, CPOL 和 NCPHA 无效。

MISO 口线上移出这些位并在 MOSI 口线上采样。

当所有位被处理，传输接收到的数据到接收数据寄存器并且 RDRF 位升高。如果当数据移传输时 RDRF 已经是高电平，则超限位升高并且中止到 SPI\_RDR 数据传输。

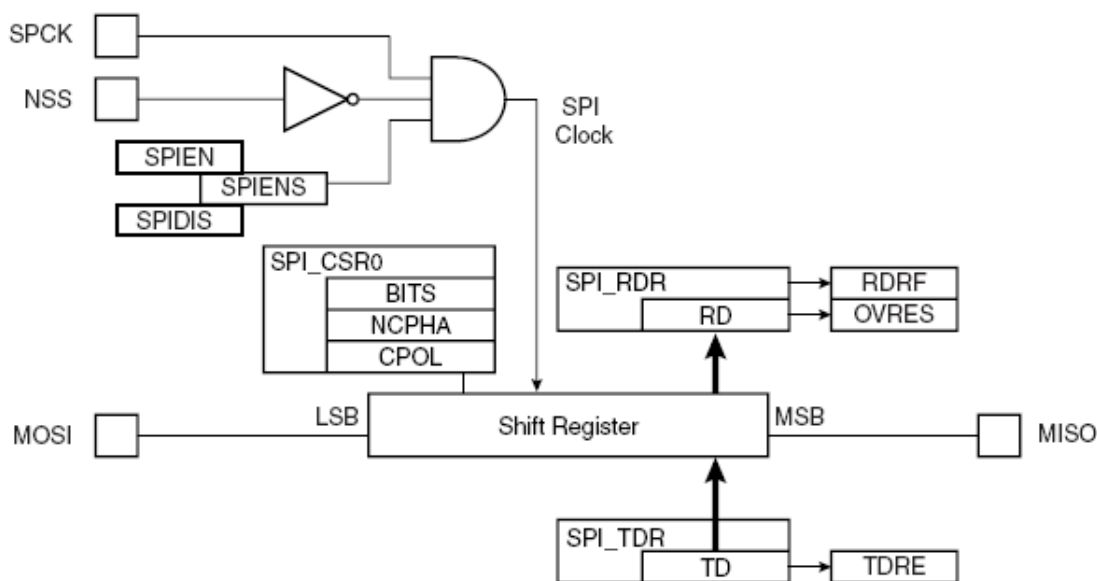
当启动一次传输时，移出的数据是当前移位寄存器中的数据。若无数据被写入发送数据寄存器中 (SPI\_TDR)，则已传输上次接收到的数据。若自上次复位未接收到数据，则所有位发送位低电平，因为移位寄存器复位值是 0。

当首次数据被写入 SPI\_TDR，则立即被传输到移位寄存器中并且 TDRE 位升高。若写入新数据，则保存在 SPI\_TDR 中直到开始传输，即，NSS 下降并且在 SPCK 引脚上有一个有效时钟。当传输开始，上次写入 SPI\_TDR 的数据被传输到移位寄存器并且 TDRE 位升高。这使得单传输关键变量频率更新可行。

接下来，从发送数据寄存器装载一个新数据到移位寄存器。万一无就绪的字符待发送，即，自上次从 SPI\_TDR 装载到移位寄存器，未向 SPI\_TDR 写入字符，未修改移位寄存器并且重发送上次接收到的字符。

图 29-9 显示了当 SPI 运行在从模式下时的方框图

图 29-9 从模式功能方框图



## 29.7 串行外设接口 (SPI) 用户接口

表 29-3 SPI 寄存器映射

偏移量	寄存器	寄存器名称	访问类型	复位值
0x00	控制寄存器	SPI_CR	只写	---
0x04	模式寄存器	SPI_MR	读/写	0x0
0x08	接收数据寄存器	SPI_RDR	只读	0x0
0x0C	发送数据寄存器	SPI_TDR	只写	---
0x10	状态寄存器	SPI_SR	只读	0x000000F0
0x14	中断使能寄存器	SPI_IER	只写	---
0x18	中断禁用寄存器	SPI_IDR	只写	---
0x1C	中断屏蔽寄存器	SPI_IMR	只读	0x0
0x20-0x2C	保留			
0x30	片选寄存器 0	SPI_CSR0	读/写	0x0
0x34	片选寄存器 1	SPI_CSR1	读/写	0x0
0x38	片选寄存器 2	SPI_CSR2	读/写	0x0
0x3C	片选寄存器 3	SPI_CSR3	读/写	0x0
0x004C-0x00F8	保留	-	-	-
0x004C-0x00FC	保留	-	-	-
0x100-0x124	为 PDC 保留			

### 29.7.1 SPI 控制寄存器

名称: SPI\_CR

访问类型: 只写

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	LASTXFER
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
SWRST	-	-	-	-	-	SPIDIS	SPIEN

● SPIEN: 使能 SPI

0=无效

1=使能 SPI 传输和接收数据

● SPIDIS: 禁用 SPI

0=无效

1=禁用 SPI

只要 SPIDIS 被置位, 则 SPI 就完成其传输

所有引脚在输入模式被置位并且不接收或发送数据。

若正在传输,则在禁用 SPI 前要完成传输。

若当写入控制寄存器时 SPIEN 和 SPIDIS 都是 1, 则 SPI 被禁用。

● SWRST: 软件复位 SPI

0=无效

1=复位 SPI。执行一个软件触发的 SPI 的硬件复位。

软件复位后, SPI 为从模式。

软件复位不影响 PDC 通道。

- LASTXFER: 上次传输

0=无效

1=传输完写入 TD 的字符后激活当前 NPCS。当 CSAAT 置位，就允许用当前串行外设通过拉高对应 NPCS 口线关闭通信，只要 TD 传输已完成。

### 29.7.2 SPI 模式寄存器

名称: SPI\_MR

访问类型: 读/写

31	30	29	28	27	26	25	24
DLYBCS							
23	22	21	20	19	18	17	16
-	-	-	-	PCS			
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
LLB	-	-	MODFDIS		PCSDEC	PS	MSTR

- MSTR: 主/从模式

0=SPI 在从模式

1=SPI 在主模式

- PS: 外设选择

0=固定外设选择

1=可变外设选择

- PCSDEC: 片选解码

0=将片选直接连接到外围设备

1=将四个片选口线连接到一个 4 到 16 位的解码器

当 PCSDEC 为 1 时，则使用一个外部 4 到 16 位的解码器用四个口线产生多达 15 个片选信号。

片选寄存器根据如下规则定义 15 个片选的特性：

SPI\_CSR0 定义外设片选信号 0-3

SPI\_CSR1 定义外设片选信号 4-7

SPI\_CSR2 定义外设片选信号 8-11

SPI\_CSRO 定义外设片选信号 12-14

- MODFDIS: 模式错误检测

0=使能模式错误检测

1=禁用模式错误检测

- LLB: 使能本地回送

0=禁用本地回送路径

1=使能本地回送路径

LLB 控制数据并串行转换器上本地回送，仅为测试主控模式。（MISO 内部的连接于 MOSI）

- PCS: 外设片选

此域仅在固定外设选择有效（PS=0）时使用。

若 PCSDEC = 0:

PCS = xx01 NPCS[3:0] = 1101

PCS = xxx0 NPCS[3:0] = 1110

PCS = x011 NPCS[3:0] = 1011  
 PCS = 0111 NPCS[3:0] = 0111  
 PCS = 1111 禁用（未选择外设）  
 （x=无关）

若 PCSDEC=1:

NPCS[3:0]输出信号= PCS。

● DLYBCS: 片选间的延迟

此域定义从无效的 NPCS 到激活另一个 NPCS 的延迟。DLYBCS 时间确保非重叠片选并且万一外设出现很长的数据浮动时间时解决总线冲突。

若 DLYBCS 小于等于 6 个，通过缺省插入 6 个 MCK 周期。

否则，一下等式决定延迟:

$$\text{片选间延迟} = \frac{DLYBCS}{MCK}$$

### 29.7.3 SPI 接收数据寄存器

名称: SPI\_RDR

访问类型: 只读

31	30	29	28	27	26	25	24
—	—	—	—	—	—	—	—
23	22	21	20	19	18	17	16
—	—	—	—	PCS			
15	14	13	12	11	10	9	8
RD							
7	6	5	4	3	2	1	0
RD							

● RD: 接收数据

由 SPI 接口接收的数据被右对齐的存储到此寄存器，未使用位读取值为 0

● PCS: 外设片选

仅在主控模式下，在一次传输的结尾这些位指示 NPCS 引脚上的值。否则，这些位读取值为 0

### 29.7.4 SPI 发送数据寄存器

名称: SPI\_TDR

访问类型: 只写

31	30	29	28	27	26	25	24
—	—	—	—	—	—	—	LASTXFER
23	22	21	20	19	18	17	16
—	—	—	—	PCS			
15	14	13	12	11	10	9	8
TD							
7	6	5	4	3	2	1	0
TD							

● TD: 发送数据

要由 SPI 接口发送的数据被存储到此寄存器中。必须将要发送的信息以右对齐格式写入发送数据寄存器。

**PCS: 外设片选**

此域仅在可变外设选择有效 (PS = 1) 时使用。

若 PCSDEC = 0:

PCS = xxx0 NPCS[3:0] = 1110

PCS = xx01 NPCS[3:0] = 1101

PCS = x011 NPCS[3:0] = 1011

PCS = 0111 NPCS[3:0] = 0111

PCS = 1111 禁用 (未选择外设)

(x=无关)

若 PCSDEC = 1:

NPCS[3:0]输出信号=PCS

● **LASTXFER: 上次传输**

0=无效

1=传输完写入 TD 的字符后激活当前 NPCS。当 CSAAT 被置位，则允许用当前串行外设通过拉高对应 NPCS 口线关闭通信，只要 TD 传输已完成。

仅在可变外设选择有效 (PS = 1) 时使用此域。

**29.7.5 SPI 状态寄存器**

名称: SPI\_SR

访问类型: 只读

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	SPIENS
15	14	13	12	11	10	9	8
-	-	-	-	-	-	TXEMPTY	NSSR
7	6	5	4	3	2	1	0
TXBUFE	RXBUFF	ENDTX	ENDRX	OVRES	MODF	TDRE	RDRF

● **RDRF: 接收数据寄存器满**

0=自上次读取 SPI\_RDR 后，未接收数据

1=自上次读取 SPI\_RDR 后，已接收数据并且已将接收到的数据从并串行转换器传输到了 SPI\_RDR。

● **TDRE: 发送数据寄存器空**

0=数据已写入 SPI\_RDR 但还未传输到并串行转换器

1=上次写入发送数据寄存器的数据已传输到并串行转换器

当禁用 SPI 或在复位时，SPITDRE 为 0。SPI 使能命令置此位为 1

● **MODF: 模式错误出错**

0=自上次读取 SPI\_SR，未检测模式错误

1=自上次读取 SPI\_SR，检测到一个模式错误

● **OVRES: 超限出错状态**

0=自上次读取 SPI\_SR，未检测到超限

1=自上次读取 SPI\_SR，产生一次超限

自上次读取 SPI\_RDR，当从并串行转换器至少两次装载 SPI\_RDR 时产生一次超限。

● **ENDRX: RX 缓冲器结束**

0=自上次写入 SPI\_RCR(1) 或 SPI\_RNCR(1)，接收计数器寄存器未达到 0

1=自上次写入 SPI\_RCR(1) 或 SPI\_RNCR(1)，接收计数器寄存器已达到 0

● ENDTX: TX 缓冲器结束

0=自上次写入 SPI\_TCR(1) 或 SPI\_TNCR(1)，发送计数器寄存器未达到 0

1=自上次写入 SPI\_TCR(1) 或 SPI\_TNCR(1)，发送计数器寄存器已达到 0

● RXBUFF: RX 缓冲器满

0=SPI\_RCR(1) 或 SPI\_RNCR(1)除了 0 还有其它值

1=SPI\_RCR(1) 和 SPI\_RNCR(1)都有一个 0 值

● TXBUFE: TX 缓冲器空

0 = SPI\_TCR(1) 或 SPI\_TNCR(1) 除了 0 还有其它值

1=SPI\_TCR(1) 和 SPI\_TNCR(1)有一个 0 值

● NSSR: NSS 上升沿

0=自上次读取，未检测到上升沿

1=自上次读取，产生一次上升沿

● TXEMPTY: 发送寄存器空

0=只要将数据写入 SPI\_TDR

1=SPI\_TDR 和内部移位器空。若已定义传输延迟，在这样的延迟完成后置位 TXEMPTY。

● SPIENS: SPI 使能状态

0=禁用 SPI

1=使能 SPI

注意：1. 物理分配 SPI\_RCR, SPI\_RNCR, SPI\_TCR, SPI\_TNCR 到 PDC。

### 29.7.6 SPI 中断使能寄存器

名称: SPI\_IER

访问类型: 只写

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	TXEMPTY	NSSR
7	6	5	4	3	2	1	0
TXBUFE	RXBUFF	ENDTX	ENDRX	OVRES	MODF	TDRE	RDRF

● RDRF: 接收数据寄存器满中断使能

● TDRE: SPI 发送数据寄存器空中断使能

● MODF: 模式错误出错中断使能

● OVRES: 超限出错中断使能

● ENDRX: 接收缓冲器结束中断使能

● ENDTX: 发送缓冲器结束中断使能

● RXBUFF: 接收缓冲器满中断使能

● TXBUFE: 发送缓冲器空中断使能

● TXEMPTY: 发送寄存器空使能

● NSSR: NSS 上升中断使能

0=无效

1=使能对应中断

### 29.7.7 SPI 中断禁用寄存器

名称：SPI\_IDR

访问类型：只写

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	TXEMPTY	NSSR
7	6	5	4	3	2	1	0
TXBUFE	RXBUFF	ENDTX	ENDRX	OVRES	MODF	TDRE	RDRF

- RDRF: 接收数据寄存器满中断禁用
- TDRE: SPI 发送数据寄存器空中断禁用
- MODF: 模式错误出错中断禁用
- OVRES: 超限出错中断禁用
- ENDRX: 接收缓冲器结束中断禁用
- ENDTX: 发送缓冲器结束中断禁用
- RXBUFF: 接收缓冲器满中断禁用
- TXBUFE: 发送缓冲器空中断禁用
- TXEMPTY: 发送寄存器空禁用
- NSSR: NSS 上升中断禁用

0=无效

1=禁用对应中断

### 29.7.8 SPI 中断屏蔽寄存器

名称：SPI\_IMR

访问类型：只读

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	TXEMPTY	NSSR
7	6	5	4	3	2	1	0
TXBUFE	RXBUFF	ENDTX	ENDRX	OVRES	MODF	TDRE	RDRF

- RDRF: 接收数据寄存器满中断屏蔽
- TDRE: SPI 发送数据寄存器空中断屏蔽
- MODF: 模式错误出错中断屏蔽
- OVRES: 超限出错中断屏蔽
- ENDRX: 接收缓冲器结束中断屏蔽
- ENDTX: 发送缓冲器结束中断屏蔽
- RXBUFF: 接收缓冲器满中断屏蔽

- TXBUFE: 发送缓冲器空中断屏蔽
- TXEMPTY: 发送寄存器空屏蔽
- NSSR: NSS 上升中断屏蔽

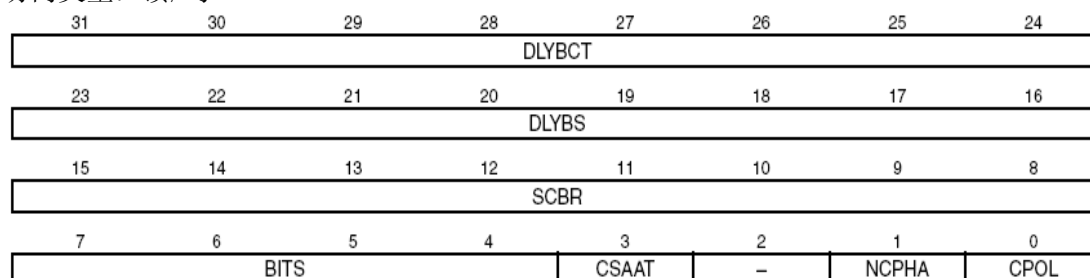
0=对应中断未使能

1=对应中断已使能

### 29.7.9 SPI 片选寄存器

名称: SPI\_CSR0... SPI\_CSR3

访问类型: 读/写



- CPOL: 时钟极性

0=SPCK 的待用状态值是逻辑电平 0

1=SPCK 的待用状态值是逻辑电平 1

使用 CPOL 确定串行时钟 (SPCK) 的待用状态值。用 CPOL 和 NCPHA 在主控和从设备之间引入需要的时钟/数据关系。

- NCPHA: 时钟相位

0=SPCK 前沿上改变数据并且在 SPCK 后沿捕捉到

1=SPCK 前沿上捕捉到数据并且在 SPCK 后沿改变

NCPHA 决定 SPCK 的哪个边沿导致数据改变，哪个边沿导致数据被捕捉到。用 NCPHA 和 CPOL 在 主控和从设备之间引入需要的时钟/数据关系。

- CSAAT: 传输后片选有效

0=只要上次传输到达则外设片选口线升高

1=上次传输到达后外设片选口线未升高。仍然维持有效直到其它的片选上请求新的传输。

- BITS: 每次传输位数

BITS 域决定传输数据的位数，不应使用保留值。

BITS	每次传输的位数
0000	8
0001	9
0010	10
0011	11
0100	12
0101	13
0110	14
0111	15
1000	16
1001	保留
1010	保留
1011	保留

1100	保留
1101	保留
1110	保留
1111	保留

● **SCBR: 串行时钟波特率**

主控模式下，SPI 接口使用模数计数器从主控时钟 MCK 取得 SPCK 波特率。通过向 SCBR 写入一个 1 到 255 之间的值来选择波特率。一下方程决定 SPCK 波特率：

$$\text{SPCK 波特率} = \frac{MCK}{SCBR}$$

禁止将 SCBR 域编程为 0。当 SCBR 为 0 时触发一此传输可能导致不可预期的结果。复位后，SCBR 为 0 并且用户比素在执行首次传输前以一个有效值编程 SCBR。

● **DLYBS: SPCK 前延迟**

此域定义了从 NPCS 有效到首次有效的 SPCK 传输的延迟

当 DLYBS 等于 0 时，从 NPCS 有效到首次有效的 SPCK 传输为 1/2 个 SPCK 时钟周期。

否则，以下方程决定延迟时间：

$$\text{Delay Before SPCK} = \frac{DLYBS}{MCK}$$

● **DLYBCT: 连续传输间的延迟**

此域定义未调动片选的情形下同一外设两次连续传输间的延迟。

通常在每次传输后和调动片选（若需要）前插入延迟。

当 DLYBCT 为 0，则在连续传输间不插入延迟并且时钟在整个字符传输期间保持其本来的周期。

否则，下式决定此延迟：

$$\text{连续传输间的延迟} = 32 * DLYBCT / MCK$$



**Educate Different** 

Powered by Team Mcuzone

QQ:8204136

Website: [www.mcuzone.com](http://www.mcuzone.com)

2008



[www.mcuzone.com](http://www.mcuzone.com)