## QSPI(QUEUED\_SPI)协议

### 1 介绍

#### 1.1 概述

QSPI完全兼容SPI通信协议并且具有SPI协议的所有功能以及一些新特性。

#### 1.2 QSPI协议特性

QSPI具有新特性：

* 全双工，三线同步传输；
* 半双工，双线同步传输；
* 主模式和从模式；
* 可编程比特率；
* 可编程时钟极性和频率；
* 传输结束产生中断；
* 可连接多种外部设备；

QSPI加强的特性：

* 可编程队列；
* 可编程外芯片选择信号；
* 回环传输模式，自动扫描模数转换器等外设，没有CPU开销；
* 可编程传输位宽；
* 可编程传输延时；
* 可编程队列指针；
* 连续传输模式；

#### 1.3 组成部分

* 可编程队列：允许QSPI执行多达16个串行传输且无需CPU参与；每个传输都对应于一个队列条目，其中包含QSPI独立完成一个串行传输所需的所有信息；极大地减少了CPU与QSPI模块的交互，增加了CPU和系统吞吐量；
* 可编程外设芯片选择信号：通过四个芯片选择引脚允许QSPI访问多达16个独立的外设；直接连接到芯片选择引脚则最多可以选择四个独立的外设；
* 回环传输模式：允许自动、连续地重新执行预先编程的队列。新传输的数据替换以前传输的数据；简化了与模数转换器通信的接口；
* 可编程传输长度：串行传输的比特数是可编程的(8到16位)；可编程长度简化了与需要不同数据长度的串行外设的接口；
* 可编程传输延迟：可编程传输间延时大约为1到500µs(使用16.78MHZ系统时钟)。可编程的延时简化了与串行外设在数据传输之间需要延迟时间；
* 可编程队列指针：QSPI有一个指针，用于标识包含下一个串行传输数据的队列位置。CPU可以通过写入队列指针来切换传输，从而改变接下来要传输的队列中的位置；否则，指针在每次串行传输后递增；
* 连续传输模式：连续传输模式允许用户使用外设发送和接收不间断的比特流。在没有CPU干预的情况下，单次可以传输最少8位，最多256位。然而，在每个队列条目传输之间需要插入1µs的延时来防止数据丢失(使用16.78 mhz系统时钟)；

### 2 QSPI信号

#### 2.1 PCS[3:0]

提供了QSPI外部设备芯片选择信号。

#### 2.2 SS

从设备选择信号；当QSPI处于从模式时，此双向信号的断言选择QSPI，复用PCS0引脚。

#### 2.3 SCK

此引脚用于输出或接收时钟信号(从模块）。

#### 2.4 MOSI

当QSPI模块被配置为主模式时，该引脚用于将数据传出；被配置为从模式时，该引脚用于接收数据。

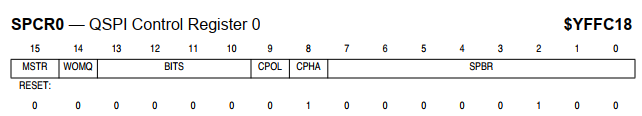
#### 2.5 MISO

当SPI模块被配置为从模式时，该引脚用于将数据传出；当被配置为主模式时，该引脚用于接收数据。

### 3 寄存器

#### 3.1 QSPI控制寄存器0

SPCR0在启用QSPI之前配置，CPU可以读写这个寄存器。

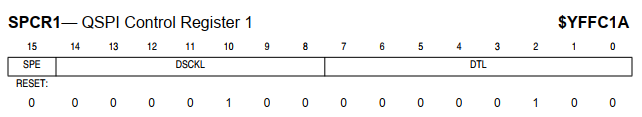


* MSTR：主从模式选择；复位时被清除并且只能由CPU写入；
* WOMQ：QSPI引脚输出模式(是否为开漏输出)；无论是否启用QSPI，WOMQ都会影响QSPI引脚；
* BITS：每次传输数据位数；在主模式下，决定队列中每个串行传输的数据位数；如果某个命令的BITSE为0，则不管该命令的BITS值是多少，该命令都会传输8位数据；从模式下不使用BITSE，所有传输位宽都由BITS指定；
* CPOL：时钟极性；指示时钟高有效还是低有效；
* CPHA:时钟相位；
* SPBR：串行时钟波特率；

SCK波特率：波特率=系统时钟/(2 \* SPBR)；

#### 3.2 QSPI控制寄存器1

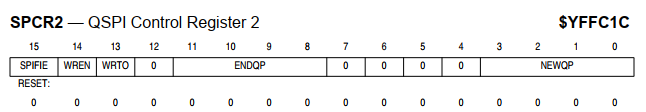
SPCR1在启用QSPI之前配置。CPU可以读写这个寄存器。在完成所有串行传输或发生模式故障时，QSPI会自动清除SPE位。



* SPE：QSPI使能；设置SPE会导致QSPI进入工作状态；如果QSPI为主模式则设置SPE会导致QSPI开始启动串行传输；为从模式则等待响应外部发起的串行传输；当QSPI被禁用时，CPU可以使用QSPI RAM。当QSPI被启用时，QSPI和CPU都可以访问QSPI RAM。CPU对QSPI RAM的所有80个字节都具有读写访问能力。QSPI只能读取QSPI RAM的发送数据段和命令控制段，也只能写入QSPI RAM的接收数据段；当完成传输时，QSPI自动清除SPE；通过设置SPCR3中的HALT位并等待SPSR中的HALTA位设置完成，然后SPE可以被安全地清除为零；CPU可以通过清除SPE立即禁用QSPI，但是这会导致当前串行传输中的数据丢失；
* DSCKL：这个位决定了QSPI选择信号 PCS有效相对SCK的延时，即SCK传输多长时间后PCS有效；实际延时为DSCKL/系统时钟频率；如果DSCK等于零，则不使用DSCKL，此时PCS相对 SCK延迟半个SCK周期；
* DTL：传输后延时大小；决定了QSPI在每次串行传输后延时；时延为(32 \* DTL)/系统时钟频率；

#### 3.3 QSPI控制寄存器2

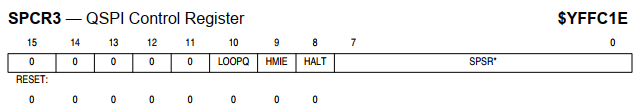
SPCR2用于配置QSPI；CPU可以读写这个寄存器；对该寄存器的写入将被缓冲。在数据传输中更改SPCR2(在QSPI运行时)无效，但在下一个串行传输中生效。SPCR2的读取将返回寄存器的实际当前值，而不是缓冲的值。



* SPIFIE：QSPI完成中断使能；所以写入SPIFIE的值仅在队列完成时应用(由ENDQP指示的条目的传输)。如果要传输单个队列条目序列(即没有WRAP)，则应该在第一次传输之前将SPIFIE设置为所需的状态。如果要使用子队列(参见位NEWQP)，导致子队列分支的同一CPU写操作可以启用或禁用子队列的SPIF中断。主队列保留自己选择的中断模式，可以启用，也可以禁用；SPIF中断必须通过清除SPIF来清除。将SPIFIE清除为零，可以禁止以后的中断；清除SPIFIE不会立即清除已经由SPIF引起的中断；
* WREN：回环传输使能；如果启用，QSPI将通过ENDQP中包含的命令执行队列中的命令。根据WRTO的状态，在地址$0或NEWQP中找到的地址继续执行。QSPI继续循环，直到WREN设置为0、HALT设置为1或SPE设置为0。一旦WREN被设置为0，QSPI通过ENDQP中包含的地址的命令完成执行命令，设置SPIF标志，然后停止。当设置WREN时，每次QSPI传输由ENDQP指示的条目时，都会设置SPIF；
* WRTO：当启用回环传输模式并且到达队列的末尾后，WRTO决定接下来QSPI从哪个地址执行。队列的结束由与ENDQP匹配的地址决定。如果没有设置WRTO，执行将包装为地址$0，如果设置了WRTO，则设置为在NEWQP的地址；
* ENDQP：结束队列指针；在队列中补全的最后一个绝对地址QSPI。在完成每个命令之后，QSPI将刚刚完成的命令的队列指针值与ENDQP的值进行比较。如果两个值匹配，则QSPI已经到达了队列的末端并将SPIF标志设置为1；可以在任何时候写入NEWQP值，从而改变执行流程。ENDQP也可以在任何时候写入，从而改变队列的长度；
* NEWQP：新队列指针值；确定QSPI首先传输哪个队列条目。NEWQP应该在使用SPE启用QSPI之前初始化。NEWQP也可以在QSPI正在运行。当发生这种情况时，QSPI完成正在进行的队列项的传输，然后立即开始从NEWQP指示的地址开始执行命令。通过这种方式，NEWQP提供了一种在QSPI RAM中支持多个队列或子队列的机制，从而为QSPI提供了额外的功能。通过更改NEWQP中的值，通过更改NEWQP中的值，可以使QSPI从队列中的任何位置开始执行一系列QSPI命令。可以预先为QSPI RAM中的不同任务设置单独的子队列。通过写入NEWQP，可以完成QSPI RAM中不同子队列之间的选择；

#### 3.4 QSPI控制寄存器3

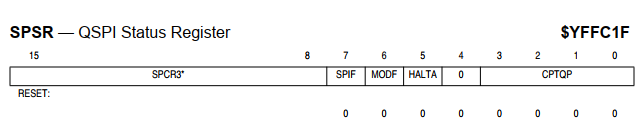
SPCR3配置QSPI；CPU可以读写这个寄存器。



* LOOPQ：QSPI循环状态；LOOPQ启用或禁用测试的数据串行器上的反馈路径。如果启用，LOOPQ将串行输出数据路由回数据序列化器，而不是接收到的数据。如果禁用，LOOPQ允许常规接收的数据进入数据序列化器。LOOPQ不影响QSPI输出引脚；
* HMIE：HALTA和MODF中断使能；
* HALT：CPU使用这个位来停止队列边界上的QSPI。QSPI在已知状态下停止，然后可以重新启动。当设置HALT时，QSPI完成当前串行传输(最多16位)后停止。在停止时，如果设置了最后一个命令的CONT位，那么QSPI将继续使用停止前最后一个命令指定的值驱动外围芯片选择引脚；如果在队列的最后一个命令期间设置HALT，那么QSPI将在完成最后一个命令后设置HALTA和SPIF并清除SPE。如果没有执行最后一个队列命令，断言HALT既不会设置SPIF也不会清除SPE。当CPU清除HALT时，QSPI继续执行；

#### 3.5 QSPI状态寄存器

SPSR包含QSPI状态信息。只有QSPI可以设置这个寄存器中的位。CPU读取这个寄存器来获取状态信息并写入这个寄存器来清除状态标志。CPU写入CPTQP没有影响。

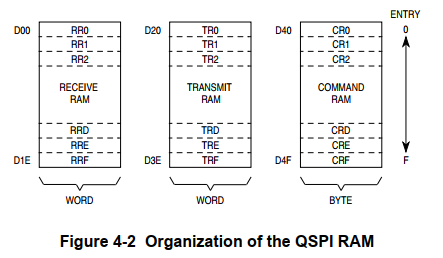


* SPIF：QSPI完成标志；SPIF是在QSPI执行完SPCR2中ENDQP中包含的地址所决定的最后一个命令时设置的。当所执行命令的地址与ENDQP匹配时，在完成串行传输后设置SPIF标志；
* MODF：模式错误标位；
* HALTA：停止确认标志位；
* CPTQP：完成的队列指针；队列中已完成的最后一个命令的队列指针值。CPTQP的值直到命令完全完成才会更新。当队列中的第一个命令正在执行时，CPTQP为重置值($0)或指向前一个队列中完成的最后一个命令的指针。如果QSPI停止，可以使用CPTQP来确定哪些命令没有执行；

#### 3.6 RAM

QSPI使用一个80字节的静态RAM块，QSPI和CPU都可以访问。当QSPI启用时，CPU访问QSPI RAM的时间可能比QSPI被禁用时长。QSPI在读写过程中可以插入1到4个CPU等待状态。

CPU对QSPI RAM的访问大小和类型会影响QSPI访问时间。QSPI是字节、字和长字可寻址的。当对QSPI RAM的访问是一个不可分割便称为CPU对RAM的访问是一致的。如果CPU对QSPI RAM进行了一致的访问，那么在CPU完成之前，QSPI不能访问QSPI RAM。长字或未对齐的字访问是不一致的，因为CPU必须将其对QSPI RAM的访问分为两部分，这允许QSPI在CPU的两次访问之间访问QSPI RAM。



如上图所示，RAM分为接收数据RAM、发送数据RAM和命令控制RAM。接收数据是从外部的串行设备接收的数据。传输数据是CPU存储的数据，用于传输到外部的外围芯片。命令ram包含QSPI执行传输所需的所有信息。

##### 3.6.1 接收RAM

RAM的这一段存储QSPI从外设、SPI总线主机或其他外部设备接收到的数据。CPU读取这段RAM以从QSPI检索数据。存储在接收RAM中的数据是右对齐的，即不管串行传输长度如何，最低有效位总是在字(第0位)中最右的位置。CPU可以使用字节、字或长字寻址访问数据。

SPSR中的CPTQP值显示已执行的队列条目。CPU在读取数据之前使用这些信息来确定接收RAM中的哪些位置包含有效数据。

##### 3.6.2 发送RAM

RAM的这一段存储由QSPI传输到外设的数据。CPU通常为要执行的每个队列命令写入一个字的数据到这个段中。如果相应的外设(如串行输入端口)仅用于输入数据，则不需要初始化此段。

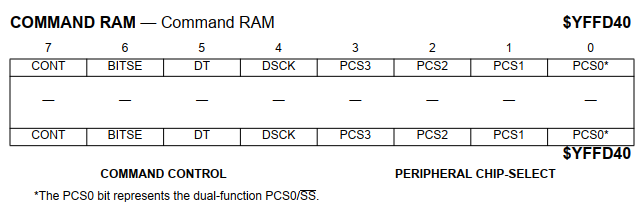
QSPI要传输的信息应该由CPU以右对齐的方式写入传输数据段。RAM的传输数据段中的信息不能被QSPI修改。QSPI只是将信息复制到串行转换器以便传输到外设。传输RAM中的信息一直保留在那里，直到CPU重新写入。

##### 3.6.3 命令RAM

QSPI RAM的命令段仅在QSPI处于主模式时使用。对于要执行的每个QSPI命令，CPU将一个字节的控制信息写入这个段。命令RAM中的信息不能被QSPI修改。它只是使用这些信息来执行串行传输。

命令RAM由16个字节组成。每个字节分成两个字段。第一个是外设芯片选择字段，在传输过程中激活正确的串行外设。第二个是命令控制字段，专门为命令/串行传输提供传输选项。该特性为用户提供了对每次传输的更多控制，提供了与不同要求的外部SPI芯片接口的灵活性。

队列命令控制字节中最多可包含16条命令。这些字节被分配一个从$0到$F的地址。QSPI的队列执行从NEWQP中包含的地址开始，一直到ENDQP中包含的地址。这两个字段都包含在SPCR2中。

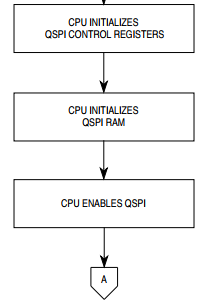


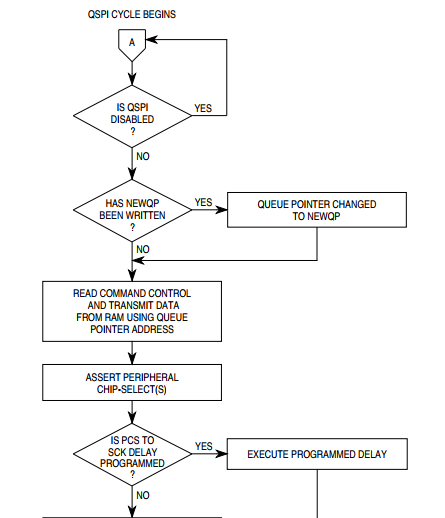
如上图所示，命令格式：

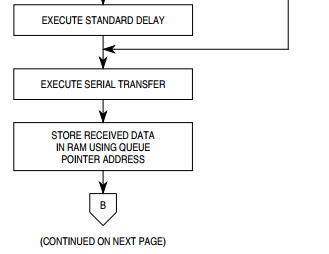
* PCS[3:0]：外设芯片选择信号；可直接连接或也可通过外部硬件解码。一次可以激活多个外设芯片选择，这对于多节点SPI系统中的广播消息非常有用。每个PCS引脚可以连接多个芯片。必须注意不要超过引脚的最大驱动能力；
* CONT：在每次QSPI传输之间必须取消某些外部芯片的选择。其他芯片必须在几个连续的串行传输之间保持选择。CONT旨在提供处理这两种情况。如果CONT设置为1并且下一个命令的外设芯片选择模式与当前命令相同，那么在两次串行传输中，QSPI将PCS引脚连续驱动到相同的值。如果CONT设置为1并且下一个命令的外围芯片选择模式与当前命令不同，QSPI将PCS引脚驱动到新值进行第二次串行传输；当CONT设置为0时，QSPI驱动PCS引脚新值；
* BITSE：使能比特传输位数；
* DT：使用DT字段。用户可以通过在与需要延迟的外围设备的传输之间执行与其他外围设备的传输来避免使用此延迟选项。这种交错操作提高了有效的串行传输速率；
* DSCK：PCS到SCK的延迟；设置为0时PCS延迟半个SCK周期；

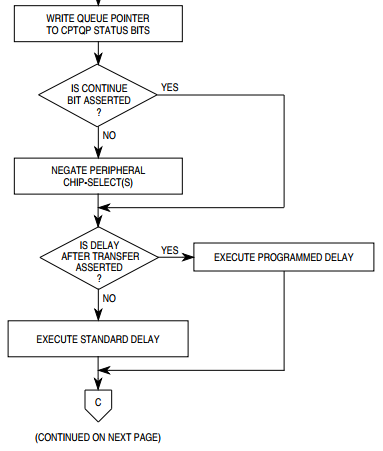
### 4 功能描述

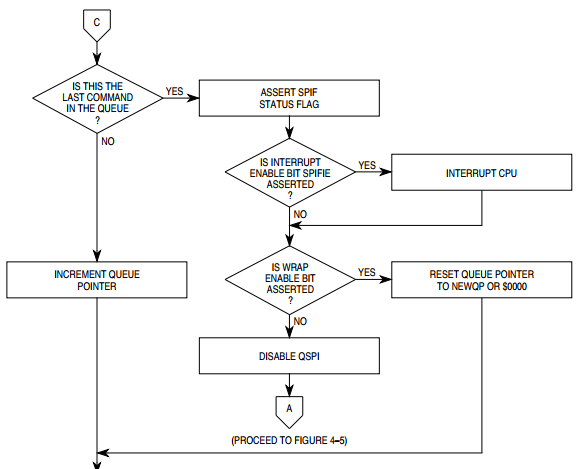
#### 4.1 主模式

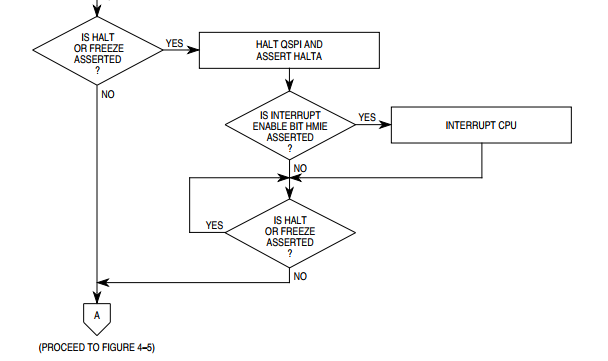












虽然QSPI支持多master操作，但没有提供特殊的仲裁机制。使用MODF来指示SPI仲裁的请求；仲裁必须由系统软件实现。注意，MSTR不会被模式错误清除，也不会禁用QSPI；当软件清除QSPI寄存器SPCR1中的SPE时，QSPI将被禁用。

通常，SPI总线执行同步的双向同步传输。SPI总线master提供时钟信号(SCK)来为比特的传输计时。时钟相位和极性可以采用四种可能的组合。数据首先以最高位传输。

通常，SPI总线的输出引脚不是开漏的，除非系统中有多个SPI主节点。如果需要，可以将SPCR0中的WOMQ设置为开漏输出。WOMQ影响所有QSPI引脚，无论分配给QSPI还是用作通用I/O。

当在主模式下操作时，QSPI可以启动串行传输。QSPI无法响应任何外部发起的串行传输。SCK引脚应该配置为输出。引脚MOSI和PCS3-PCS0/SS应根据需要配置为输出。如果需要，应该将MISO配置为输入。

主模式操作所需的引脚有MISO、MOSI,、SCK以及一个或多个PCS引脚，具体取决于要选择的外部外围芯片数量。主模式使用MISO作为数据输入引脚，主模式使用MOSI作为数据输出引脚。根据特定的应用程序，其中一种或两种都可能是必需的。SCK是主模式下的串行时钟输出。

PCS[3:0]/SS是用于为QSPI发起的串行传输选择外部SPI外围芯片的选择引脚。

##### 4.1.1 主模式操作

复位后，QSPI控制寄存器必须按照上面所述进行初始化。除了命令控制段之外，根据应用程序的不同，可能还需要初始化传输数据段。如果要从QSPI发送数据，用户应该在使能QSPI之前将数据写入传输数据段。

设置SPE后不久，QSPI将根据NEWQP从指定的地址处开始操作。QSPI将发送数据段中的数据按NEWQP指定的地址发送，QSPI将接收数据按NEWQP指定的地址存储在接收数据段中。数据与内部生成的SCK同步传输。

传输数据加载到数据串行器中。QSPI使用控制位CPHA和CPOL来确定MISO引脚使用哪条SCK边来锁住传入数据，而MOSI引脚使用哪条边来开始驱动传出数据。SCK的波特率由SPCR0的SPBR决定。DSCK和DSCKL确定片选信号对SCK的延迟。传输的比特数由BITSE和BITS字段决定。用户可以使用默认值8位，或者可以编程8 - 16位的长度。

一旦传输了适当的比特数，QSPI将接收到的数据存储在接收数据段中，将内部工作队列指针的值存储在CPTQP中，增加内部工作队列指针，并从队列中加载传输所需的下一个数据。内部工作队列指针地址是下一个执行的命令，除非CPU先写一个新值。

如果设置了CONT，并且外设芯片选择信号在上一个和当前传输之间没有改变，那么在两个串行传输期间，PCS引脚将保持不变。如果片选信号改变，则在第一次传输执行期间将取消片选，然后在第二次传输开始执行时QSPI切换到第二次传输的片选信号。如果CONT被清除，则在传输之间取消未被选择的片选信号 (在寄存器PORTQS中找到)。

DT会在指定的串行传输完成后进行延时。延时的长度由DTL决定。当DT被清除，使用标准延时。

##### 4.1.2 回环传输

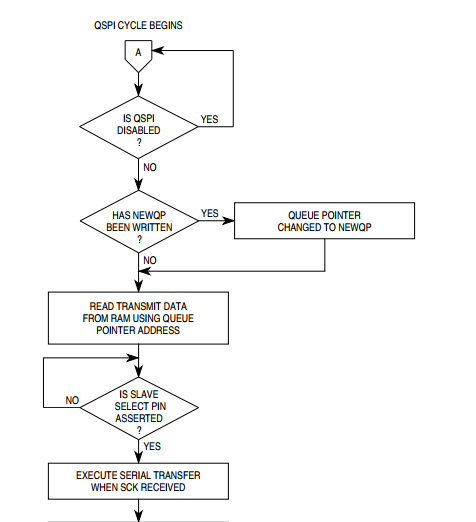
当QSPI到达队列的末尾时，总是设置SPIF标志，无论回环传输是启用还是禁用。当设置SPIF时，会生成一个可选的CPU中断。此时，QSPI清除SPE并停止，除非启用了回环传输。

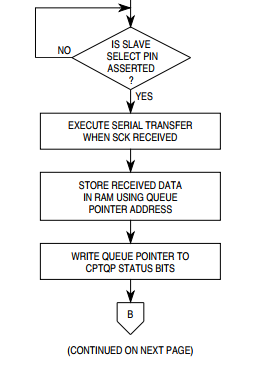
在回环传输下，QSPI连续地在队列中循环。每次到达队列末尾时，都会设置SPIF标志。如果CPU未能清除SPIF，将保持设置，QSPI继续向CPU发送中断请求(设置SPIFIE)。用户可以通过清除SPIFIE来避免造成CPU中断。由于SPIFIE是缓冲的，设置SPIF标志后清除不会立即停止CPU中断，而只是防止来自此源的下个中断。为了清除当前中断，CPU必须读取QSPI寄存器SPSR并清除SPIF。

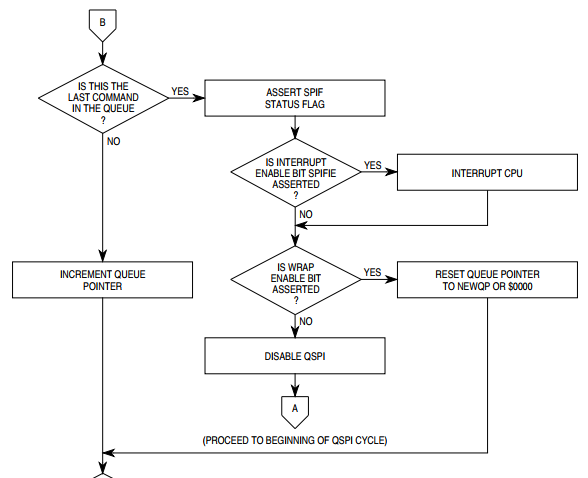
即使QSPI正在从CPU请求中断服务，执行也会以回环传输模式继续进行。内部工作队列指针增加到下一个地址，并再次执行命令。QSPI不清除SPE。新的接收数据将覆盖接收数据段中的先前接收数据。

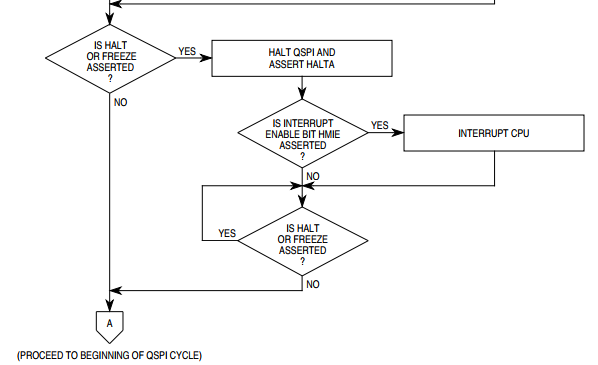
CPU可能会通过清除WREN禁用环绕模式或设置HALT。第一种下一次到达队列末端时，QSPI设置SPIF清除SPE并停止。第二种方法在当前传输完成后停止QSPI，允许CPU清除SPE。CPU可以通过清除SPE立即停止QSPI；不建议使用此方法，因为会导致QSPI终止正在进行的串行传输。

#### 4.2 从模式









在从模式下工作时，QSPI可以响应外部发起的串行传输。QSPI无法启动任何串行传输。当SPI总线网络中有多个mcu时，通常使用从模式，因为在任何给定时间只有一个设备可以是SPI主设备(在主模式下)。如果需要，MISO和MOSI引脚应分别配置为输出和输入。引脚SCK和PCS0/SS应配置为输入。

SCK是从模式下的从时钟输入。PCS0/SS是选择引脚；当QSPI处于从模式时，用于从外部SPI总线选择QSPI进行串行传输。

命令控制段不在从模式下实现，因此不需要初始化。这段QSPI RAM和任何其他未使用的段可以被CPU用作通用RAM。

##### 4.2.1 从模式操作

复位后，QSPI控制寄存器必须按照上面所述进行初始化。虽然没有使用命令控制段，但根据应用程序的不同，可能需要初始化发送和接收数据段。用户应该在使能QSPI之前将数据写入传输数据段。如果设置了SPE而没有设置MSTR，从选(PCS0/SS)引脚上为低电平时从NEWQP指定的地址处开始操作。QSPI将发送数据段中的数据按NEWQP指定的地址发送，QSPI将接收数据按NEWQP指定的地址存储在接收数据段中。

由于没有使用命令控制段，命令控制位和外围芯片选择码在从模式操作中没有作用。QSPI不驱动任何片选信号作为输出。PCS0/SS作为输入。虽然CONT不能在从模式下使用，但有规定允许接收超过16位的数据。在保持选定的QSPI (PCS0/SS保持低位)的同时，QSPI在当前接收数据段地址中存储由BITS指定的位数，对NEWQP进行递增，并在下一个接收数据段地址中继续存储剩余的位。

只要PCS0/SS保持在低水平，QSPI就会继续在连续接收数据段地址中存储传入的位流，直到达到BITS中的值或禁用回环传输使用队列结束地址。当到达队列末尾时，将设置SPIF标志，会产生可选的cpu中断。如果回环传输被禁用，任何额外的输入位将被忽略。如果启用了回环传输，则根据WRTO值继续存储到地址$0或NEWQP的地址。

当使用此功能接收较长的传入数据流时，必须使用适当的传输间隔。QSPI需要时间，在16.78 mhz系统时钟下大约1µs；为下一次传输预取下一个传输RAM数据。因此，用户可以选择在连续传输之间提供至少1µs延迟的波特率，以确保不丢失传入数据。如果系统时钟以较慢的速度运行，则传输之间的延迟必须成比例地增加。

因为命令控制段中的BITSE选项不再可用，BITS设置队列中所有传输要传输的比特数，直到CPU更改BITS值。如果PCS0/SS在接收到适当的比特数之前被取消，那么下一次选择QSPI时，将在停止的同一接收数据段地址中存储比特。如果在对PCS0/ SS取消之前传输了超过16位，QSPI将在当前接收数据段地址中存储由BITS表示的位数，然后增加地址并按上述方式继续存储。

一旦传输了适当的比特数(由BITS指定)，QSPI将接收到的数据存储在接收数据段中，存储CPTQP，增加内部工作队列指针，并将新的传输数据从传输数据段加载到数据串行器中。在下一次断言PCS0/SS时使用内部工作队列指针地址，除非CPU写入NEWQP。

从模式下不使用DT和DSCK命令控制位。作为slave，QSPI不驱动时钟线或片选线，因此不会产生延迟。在从模式下，QSPI移出传输数据段中的数据。传输数据被加载到数据序列化器中进行传输。当PCS0/SS引脚拉低时，MISO引脚激活，然后序列化器按顺序将16位数据移出，最重要的位先移出，由输入SCK信号计时。QSPI使用CPHA和CPOL来确定MOSI引脚使用哪个SCK边缘来锁存传入的数据，以及MISO引脚使用哪个边缘来驱动数据输出。

QSPI传输和接收数据，直到到达队列的末尾(ENDQP)，而不管PCS0/SS是否保持选中或在串行传输之间切换。接收到适当数量的位将会存储接收到的数据。QSPI总是在每个队列地址上发送和接收到的一样多的位，直到达到BITS值或PCS0/SS为负。

##### 4.2.2 回环传输

当QSPI到达队列的末尾时，总是设置SPIF标志。当设置SPIF时，会生成一个的CPU的可选中断。此时，QSPI清除SPE并停止，除非启用回环传输。与主模式下的回环传输基本相同。