

セクション 18. シリアル ペリフェラル インターフェイス (SPI)

ハイライト

本セクションには以下の主要項目を記載しています。

18.1	はじめに	18-2
18.2	SPI レジスタ	18-3
18.3	動作モード	18-7
18.4	マスタモードのクロック周波数	18-20
18.5	DMA を使用する SPI 動作	18-21
18.6	省電力モード時の動作	18-24
18.7	SPI モジュール関連の特殊機能レジスタ	18-25
18.8	関連アプリケーション ノート	18-26
18.9	改訂履歴	18-27

Note: ファミリ リファレンス マニュアルの本セクションは、デバイス データシートの補足を目的としています。本セクションの内容は、dsPIC33F/PIC24H ファミリの一部のデバイスには対応していません。

本書の内容がお客様のご使用になるデバイスに対応しているかどうかは、最新デバイス データシート内の「シリアル ペリフェラル インターフェイス (SPI)」の冒頭に記載している注意書きでご確認ください。

デバイス データシートとファミリ リファレンス マニュアルの各セクションは、マイクロチップ社のウェブサイト (<http://www.microchip.com>) からダウンロードできます。

18.1 はじめに

シリアル ペリフェラル インターフェイス (SPI) モジュールは、他の周辺モジュールまたはマイクロコントローラ デバイスとの通信に使用できる同期シリアル インターフェイスです。通信相手の周辺モジュールには EEPROM、シフトレジスタ、ディスプレイ ドライバ、A/D コンバータ等が含まれます。SPI モジュールは Motorola 社製の SPI および SIOP インターフェイスと互換性を持ちます。

dsPIC33F/PIC24H ファミリには、SPI モジュールを 1 つだけ備えるデバイスと 2 つ備えるデバイスが存在します。2 つのモジュール (SPI1 と SPI2) は機能的に同じです。全てのデバイスが SPI1 モジュールを実装し、多ピンパッケージの多くのデバイスが SPI2 を追加で実装しています。

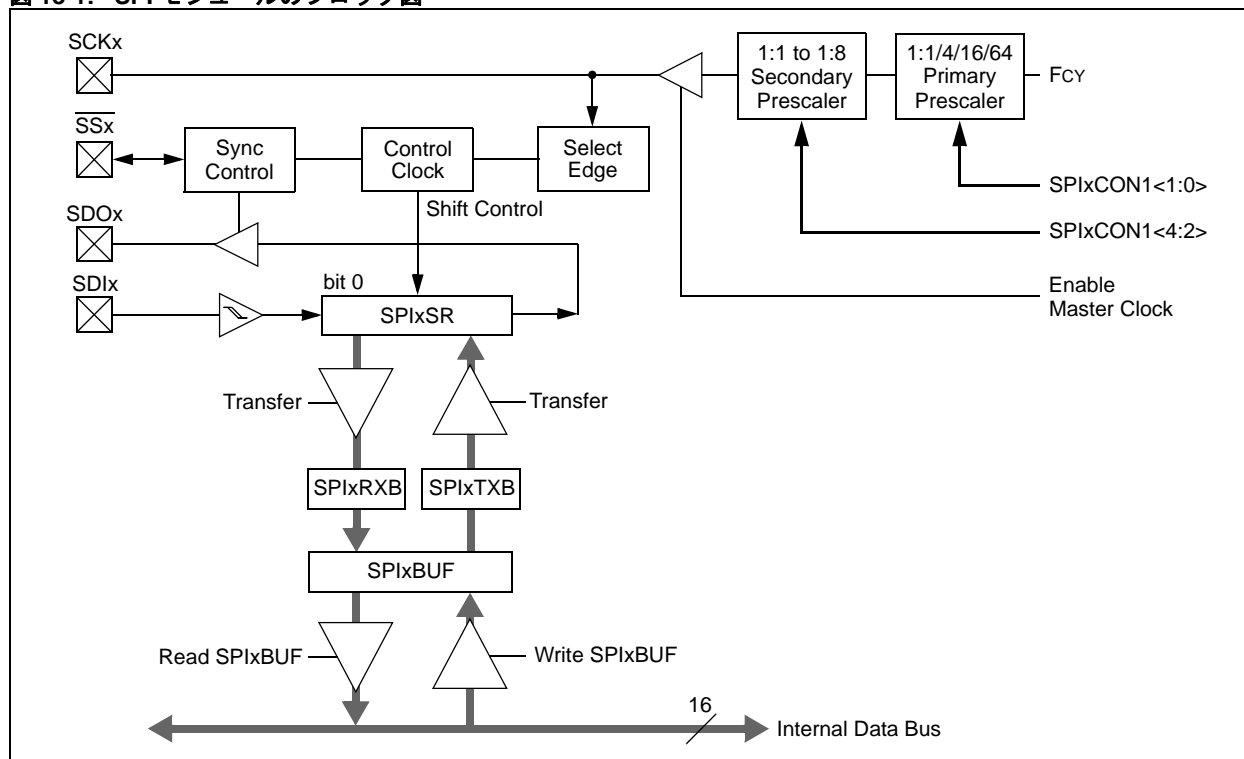
Note: 本書では、両モジュールを併せて SPIx と表記するか、個別に SPI1 および SPI2 と表記します。これらのモジュールに関連する特殊機能レジスタ (SFR) の表記もこれに従います。例えば SPIxCON は、SPI1 または SPI2 モジュール用の制御レジスタです (x は 1 または 2)。

SPIx シリアル インターフェイスは下記の 4 本のピンを備えます。

- SDIx: シリアルデータ入力
- SDOx: シリアルデータ出力
- SCKx: シフトクロック入力または出力
- SSx/FSYNCx: アクティブ LOW スレーブ選択またはフレーム同期 I/O パルス

図 18-1 に SPI モジュールのブロック図を示します。

図 18-1: SPI モジュールのブロック図



18.2 SPI レジスタ

SPIxSTAT: SPIx ステータス / 制御レジスタ

SPIx ステータス / 制御レジスタ (SPIxSTAT) は、受信オーバーフロー、送信バッファフル、受信バッファフル等の各種ステータスを示します。このレジスタは、アイドルモード時のモジュールの動作を指定します。また、モジュールを有効化 / 無効化するためのビットもこのレジスタに格納されています。

SPIxCON1: SPIx 制御レジスタ 1

SPIx 制御レジスタ 1 (SPIxCON1) はクロック プリスケアラ、マスタ/スレーブモード、ワード/バイト通信、クロック極性、クロック/データピン動作を設定します。

SPIxCON2: SPIx 制御レジスタ 2

SPIx 制御レジスタ 2 (SPIxCON2) は、フレーム化された SPI 動作を有効化 / 無効化します。このレジスタは、フレーム同期パルスの方向、極性、エッジの選択も定義します。

SPIxBUF: SPIx データ受信 / 送信バッファレジスタ

SPIx データ受信 / 送信バッファレジスタ (SPIxBUF) は、実際には別々の 2 つのバッファ (送信バッファ (SPIxTXB) と受信バッファ (SPIxRXB)) です。これら 2 つのバッファは SPIxBUF の SFR アドレスを共有する片方向の 16 ビットレジスタです。ユーザ アプリケーションが送信データを SPIxBUF アドレスに書き込む場合、データは内部的に SPIxTXB レジスタに書き込まれます。

同様にユーザ アプリケーションが受信データを SPIxBUF から読み出す場合、データは内部的に SPIxRXB レジスタから読み出されます。

この方式は送信 / 受信動作を二重バッファリングし、バックグラウンドでの連続的なデータ転送を可能にします。送信と受信は同時に発生します。

加えて、SPI モジュールはメモリには配置されない内部 16 ビットシフトレジスタ (SPIxSR) も備えます。このレジスタは SPI ポートのデータをシフトイン / シフトアウトします。

dsPIC33F/PIC24H ファミリ リファレンス マニュアル

レジスタ 18-1: SPIxSTAT: SPIx ステータス / 制御レジスタ

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
SPIEN	—	SPISIDL	—	—	—	—	—
bit 15							bit 8

U-0	R/C-0	U-0	U-0	U-0	U-0	R-0	R-0
—	SPIROV	—	—	—	—	SPITBF	SPIRBF
bit 7							bit 0

凡例:	C = クリア可能ビット		
R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し	
-n = POR 時の値	1 = ビットをセット	0 = ビットをクリア	x = ビットは未知

- bit 15 **SPIEN:** SPIx イネーブルビット
1 = モジュールを有効にし、SCKx、SDOx、SDIx、 \overline{SSx} をシリアルポート ピンとして設定する
0 = モジュールを無効にする
- bit 14 **未実装:** 「0」として読み出し
- bit 13 **SPISIDL:** アイドルモード時停止ビット
1 = デバイスがアイドルモードに移行した時にモジュールの動作を停止する
0 = アイドルモード中もモジュールの動作を継続する
- bit 12-7 **未実装:** 「0」として読み出し
- bit 6 **SPIROV:** 受信オーバーフロー フラグビット
1 = 新たなバイト/ワードを完全に受信したが破棄した(ユーザ ソフトウェアはSPIxBUF レジスタ内の既存データをまだ読み出していない)
0 = オーバーフローは発生していない
- bit 5-2 **未実装:** 「0」として読み出し
- bit 1 **SPITBF:** SPIx 送信バッファフル ステータスビット
1 = 送信をまだ開始していない (SPIxTXB はフル)
0 = 送信を開始した (SPIxTXB はエンプティ)
このビットは、CPU が SPIxBUF に書き込んだ時 (SPIxTXB に書き込んだ時) に、ハードウェアで自動的にセットされます。
このビットは、SPIx モジュールがデータを SPIxTXB から SPIxSR へ転送した時に、自動的にクリアされます。
- bit 0 **SPIRBF:** SPIx 受信バッファフル ステータスビット
1 = 受信を完了した (SPIxRXB はフル)
0 = 受信を完了していない (SPIxRXB はエンプティ)
このビットは、SPIx がデータを SPIxSR から SPIxRXB へ転送した時に、自動的にセットされます。
このビットは、CPU が SPIxBUF を読み出した時 (SPIxRXB を読み出した時) に、ハードウェアで自動的にセットされます。

セクション 18. シリアル ペリフェラル インターフェイス (SPI)

レジスタ 18-2: SPIxCON1: SPIx 制御レジスタ 1

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	DISSCK	DISSDO	MODE16	SMP ⁽¹⁾	CKE ⁽²⁾
bit 15							bit 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
SSEN	CKP	MSTEN	SPRE<2:0> ⁽³⁾			PPRE<1:0> ⁽³⁾	
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
 -n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

- bit 15-13 **未実装:** 「0」として読み出し
- bit 12 **DISSCK:** SCKx ピン ディセーブル ビット (SPI マスタモード専用)
 1 = 内部 SPI クロックを無効にする (ピンは I/O として機能する)
 0 = 内部 SPI クロックを有効にする
- bit 11 **DISSDO:** SDOx ピン ディセーブル ビット
 1 = モジュールは SDOx ピンを使用しない (ピンは I/O として機能する)
 0 = モジュールは SDOx ピンを制御する
- bit 10 **MODE16:** ワード/バイト通信選択ビット
 1 = ワード幅 (16 ビット) で通信する
 0 = バイト幅 (8 ビット) で通信する
- bit 9 **SMP:** SPIx データ入力サンプルフェイズ ビット⁽¹⁾
マスタモードの場合:
 1 = データ出力時間の最後に入力データをサンプリングする
 0 = データ出力時間の途中で入力データをサンプリングする
スレーブモードの場合:
 SPIx モジュールをスレーブモードで使用する場合、SMP ビットをクリアする必要があります。
- bit 8 **CKE:** SPIx クロックエッジ選択ビット⁽²⁾
 1 = シリアル出力データはクロックがアクティブからアイドルへ遷移した時に変化する (bit 6 参照)
 0 = シリアル出力データはクロックがアイドルからアクティブへ遷移した時に変化する (bit 6 参照)
- bit 7 **SSEN:** スレーブ選択イネーブルビット (スレーブモード)
 1 = SSx ピンをスレーブモードで使用する
 0 = SSx ピンをスレーブモードで使わない (ポート機能がピンを制御する)
- bit 6 **CKP:** クロック極性選択ビット
 1 = クロックは HIGH レベルをアイドル状態とする (LOW レベルがアクティブ状態)
 0 = クロックは LOW レベルをアイドル状態とする (HIGH レベルがアクティブ状態)
- bit 5 **MSTEN:** マスタモード イネーブルビット
 1 = マスタモード
 0 = スレーブモード
- bit 4-2 **SPRE<2:0>:** セカンダリ プリスケール ビット (マスタモード)⁽³⁾
 111 = セカンダリ プリスケールは 1:1
 110 = セカンダリ プリスケールは 2:1
 •
 •
 •
 000 = セカンダリ プリスケールは 8:1
- bit 1-0 **PPRE<1:0>:** プライマリ プリスケール ビット (マスタモード)⁽³⁾
 11 = プライマリ プリスケールは 1:1
 10 = プライマリ プリスケールは 4:1
 01 = プライマリ プリスケールは 16:1
 00 = プライマリ プリスケールは 64:1

Note 1: SMP ビットは、MSTEN ビットがセットされる場合のみセットできます。MSTEN = 0 の場合、SMP ビットをクリアする必要があります。

2: フレーム化 SPI モードでは CKE ビットを使用しません。フレーム化 SPI モード (FRMEN = 1) を使用する場合、このビットを「0」に設定する必要があります。

3: プライマリおよびセカンダリ プリスケール ビット値を 1:1 に設定しないでください。

dsPIC33F/PIC24H ファミリ リファレンス マニュアル

レジスタ 18-3: SPIxCON2: SPIx 制御レジスタ 2

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0
FRMEN	SPIFSD	FRMPOL	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	U-0
—	—	—	—	—	—	FRMDLY	—
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
 -n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

- bit 15 **FRMEN:** フレーム化 SPIx サポートビット
 1 = フレーム化 SPIx のサポートを有効にする ($\overline{\text{SSx}}$ ピンをフレーム同期パルスの入出力に使用する)
 0 = フレーム化 SPIx のサポートを無効にする
- bit 14 **SPIFSD:** フレーム同期パルス方向制御ビット
 1 = フレーム同期パルス入力 (スレーブ)
 0 = フレーム同期パルス出力 (マスタ)
- bit 13 **FRMPOL:** フレーム同期パルス極性ビット
 1 = フレーム同期パルスの極性はアクティブ HIGH
 0 = フレーム同期パルスの極性はアクティブ LOW
- bit 12-2 **未実装:** 「0」として読み出し
- bit 1 **FRMDLY:** フレーム同期パルスエッジ選択ビット
 1 = フレーム同期パルスは先頭ビットクロックと同時に発生する
 0 = フレーム同期パルスは先頭ビットクロックに先行して発生する
- bit 0 **未実装:** このビットはユーザ アプリケーションによって「1」にセットされない必要があります。

レジスタ 18-4: SPIxBUF: SPIx データ受信 / 送信バッファレジスタ

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SPIx 送信 / 受信バッファレジスタ							
bit 15							bit 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
SPIx 送信 / 受信バッファレジスタ							
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
 -n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

- bit 15-0 送信 / 受信バッファビット

18.3 動作モード

SPI モジュールは下記の動作モードを使用します。

- 8 ビットと 16 ビットのデータ送受信
- マスタモードとスレーブモード
- フレーム化 SPIx モード
- SPIx 受信専用動作
- SPIx エラー対応

18.3.1 8 ビット動作と 16 ビット動作

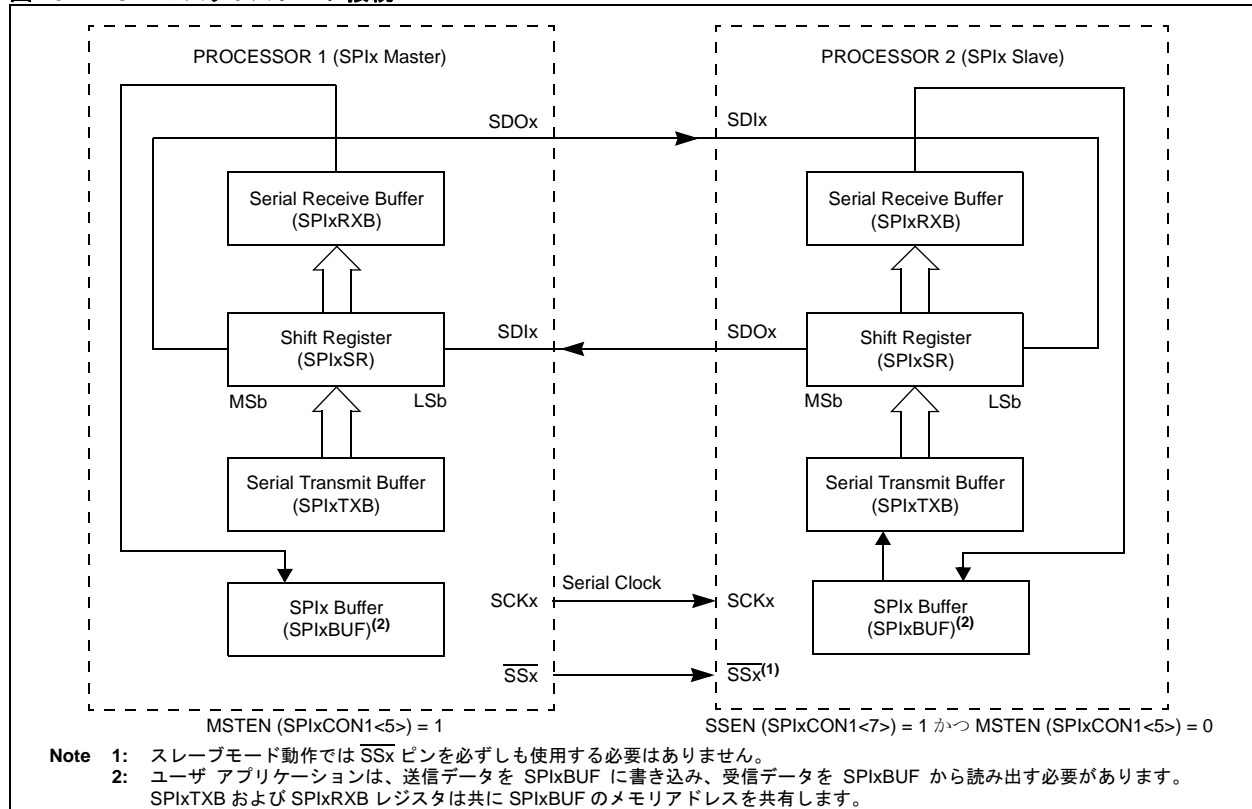
SPIx 制御レジスタ 1 (SPIxCON1<10>) のワード / バイト通信選択制御ビット (MODE16) の設定により、モジュールは 8 ビットモードまたは 16 ビットモードのいずれかで通信できます。両モードの機能は基本的に同じですが、送受信するデータのビット数が異なります。

- MODE16 ビットの値を変更するとモジュールはリセットされます。従って、通常動作中にこのビットを変更しない必要があります。
- 8 ビット動作の場合、データは SPIx シフトレジスタ (SPIxSR) の bit 7 から転送されます。16 ビット動作の場合、bit 15 (SPIxSR<15>) から転送されます。どちらのモードでも、データは bit 0 (SPIxSR<0>) にシフトインされます。
- 8 ビットモードでデータを送受信する場合、データのシフトイン / シフトアウトには SCKx ピンで 8 クロックパルスが必要です。同様に 16 ビットモードでは、SCKx ピンで 16 クロックパルスが必要です。

18.3.2 マスタモードとスレーブモード

データは、一方のモジュールのシフトレジスタ最上位ビット (MSb) と他方のモジュールのシフトレジスタ最下位ビット (LSb) の間のダイレクトパスを介して転送され、その後適切な送信バッファまたは受信バッファに転送されると考える事ができます。マスタモジュールとして設定されたモジュールは、シリアルクロックと同期信号 (必要な場合) をスレーブデバイスに供給します。図 18-2 に、マスタモジュールとスレーブモジュールの接続を示します。

図 18-2: SPI マスタ / スレーブ接続



18.3.2.1 マスタモード

マスタモードでは、プリスケールしたシステムクロックをシリアルクロックとして使用します。プリスケールは、SPIx 制御レジスタ 1 (SPIxCON1<1:0>) のプライマリ プリスケール ビット (PPRE<1:0>) と、同じく SPIx 制御レジスタ 1 (SPIxCON1<4:2>) のセカンダリ リスケール ビット (SPRE<2:0>) の設定に基づきます。シリアルクロックは SCKx ピンを介してスレーブデバイスへ出力されます。クロックパルスは、送信すべきデータが存在する時にだけ生成されます。詳細は 18.4「マスタモードのクロック周波数」を参照してください。CKP および CKE ビットは、クロックパルスのどちらのエッジでデータ送信が発生するのかを指定します。

送信データは SPIxBUF レジスタに書き込まれ、受信データは SPIxBUF レジスタから読み出されます。

マスタモードにおける SPIx モジュールの動作は下記の通りです。

1. モジュールをマスタモードに設定して有効化すると、送信データが SPIxBUF レジスタに書き込まれ、SPIx ステータス / 制御レジスタ (SPIxSTAT<1>) の SPIx 送信バッファフルステータスビット (SPITBF) がセットされます。
2. SPIx 送信バッファレジスタ (SPIxTXB) の内容が SPIx シフトレジスタ (SPIxSR) に転送され、モジュールは SPITBF ビット (SPIxSTAT<1>) をクリアします。
3. 連続した 8 または 16 回のクロックパルスにより、8 または 16 ビットの送信データがシフトレジスタ (SPIxSR) から SDOx ピンへシフトアウトされ、同時に SDIx ピン上のデータが SPIxSR ヘシフトインされます。
4. 転送が完了すると、割り込みコントローラでは下記のイベントが発生します。
 - a) 割り込みコントローラ内の対応する割り込みフラグビットがセットされます。
 - 割り込みフラグステータス レジスタ 0 (IFS0<10>) の SPI1IF ビットがセットされる
 - 割り込みフラグステータス レジスタ 2 (IFS2) の SPI2IF ビットがセットされるこれらの割り込みは、対応する割り込みイネーブルビットをセットする事により有効化されます。
 - SPI1IE は割り込みイネーブル制御レジスタ 0 (IEC0<10>) で有効化される
 - SPI12E は割り込みイネーブル制御レジスタ 2 (IEC0<2>) で有効化されるハードウェアは SPIxIF フラグを自動的にクリアしません。
 - b) 送受信動作が完了すると、SPIx シフトレジスタ (SPIxSR) の内容が SPIx 受信バッファレジスタ (SPIxRXB) に転送されます。
 - c) モジュールは SPIx ステータス / 制御レジスタ (SPIxSTAT<0>) の SPIx 受信バッファフルステータスビット (SPIRBF) をセットして、受信バッファがフルであることを示します。ユーザコードが SPIxBUF レジスタを読み出すと、ハードウェアが SPIRBF ビットをクリアします。
5. SPIx モジュールが SPIxSR から SPIxRXB へデータを転送する必要がある時に SPIRBF がセットされている (受信バッファがフルである) 場合、モジュールは SPIx ステータス / 制御レジスタ (SPIxSTAT<6>) の受信オーバーフロー フラグビット (SPIROV) をセットしてオーバーフローが発生した事を示します。
6. SPITBF ビット (SPIxSTAT<1>) がクリアされていれば、ユーザ ソフトウェアはいつでも送信データを SPIxBUF に書き込む事ができます。SPIxSR が先に書き込まれたデータをシフトアウトしている間に SPIxBUF へ書き込めるため、連続的な送信が可能です。

Note: ユーザ アプリケーションは SPIxSR レジスタに直接書き込む事はできません。SPIxSR レジスタへの書き込みは、全て SPIxBUF レジスタ経由で行われます。

セクション 18. シリアル ペリフェラル インターフェイス (SPI)

18.3.2.1.1 マスタモードのセットアップ手順

SPIx モジュールをマスタモード動作向けにセットアップする手順は下記の通りです。

1. 割り込みを使用する場合、割り込みコントローラを下記のように設定します。
 - a) 割り込みコントローラ内の各割り込みフラグステータス レジスタ (IFS0<10> または IFS2<1>) の SPIx イベント割り込みフラグステータス ビット (SPIxIF) をクリアする
 - b) 割り込みコントローラ内の各割り込みイベント制御レジスタ (IEC0<10> または IEC2<1>) の SPIx イベント割り込みイネーブルビット (SPIxIE) をセットする
 - c) 割り込みコントローラ内の各割り込み優先度制御レジスタ (IPC2<10-8> または IPC8<6-4>) の SPIx イベント割り込み優先度ビット (SPIxIP) で割り込み優先度を設定する
2. SPIxCON1 レジスタのマスタモードイネーブルビット (MSTEN) をセットします (SPIxCON1<5> = 1)。
3. SPIxSTAT レジスタの受信オーバーフロー フラグビット (SPIROV) をクリアします (SPIxSTAT<6> = 0)。
4. SPIxSTAT レジスタの SPIx イネーブルビット (SPIEN) をセットして SPIx の動作を有効にします (SPIxSTAT<15> = 1)。
5. 送信するデータを SPIxBUF レジスタに書き込みます。この書き込み後すぐに送信 (および受信) が開始されます。

例 18-1 に、SPI レジスタをマスタモード向けに設定するサンプルコードを示します。

例 18-1: SPI のコンフィグレーション - マスタモード

```
/* The following code shows the SPI register configuration for Master mode */

IFS0bits.T1IF = 0;      // Clear Timer1 Interrupt Flag
IEC0bits.SPI1IE = 0;    // Disable the Interrupt

                        // SPI1CON1 Register Settings
SPI1CON1bits.DISSCK = 0; // Internal Serial Clock is Enabled
SPI1CON1bits.DISSDO = 0; // SDOx pin is controlled by the module
SPI1CON1bits.MODE16 = 1; // Communication is word-wide (16 bits)
SPI1CON1bits.SMP = 0;   // Input data is sampled at the middle of data
                        // output time
SPI1CON1bits.CKE = 0;   // Serial output data changes on transition
                        // from Idle clock state to active clock state
SPI1CON1bits.CKP = 0;   // Idle state for clock is a low level;
                        // active state is a high level
SPI1CON1bits.MSTEN = 1; // Master mode Enabled
SPI1STATbits.SPIEN = 1; // Enable SPI module
SPI1BUF = 0x0000;      // Write data to be transmitted

                        // Interrupt Controller Settings
IFS0bits.SPI1IF = 0;    // Clear the Interrupt Flag
IEC0bits.SPI1IE = 1;    // Enable the Interrupt
```

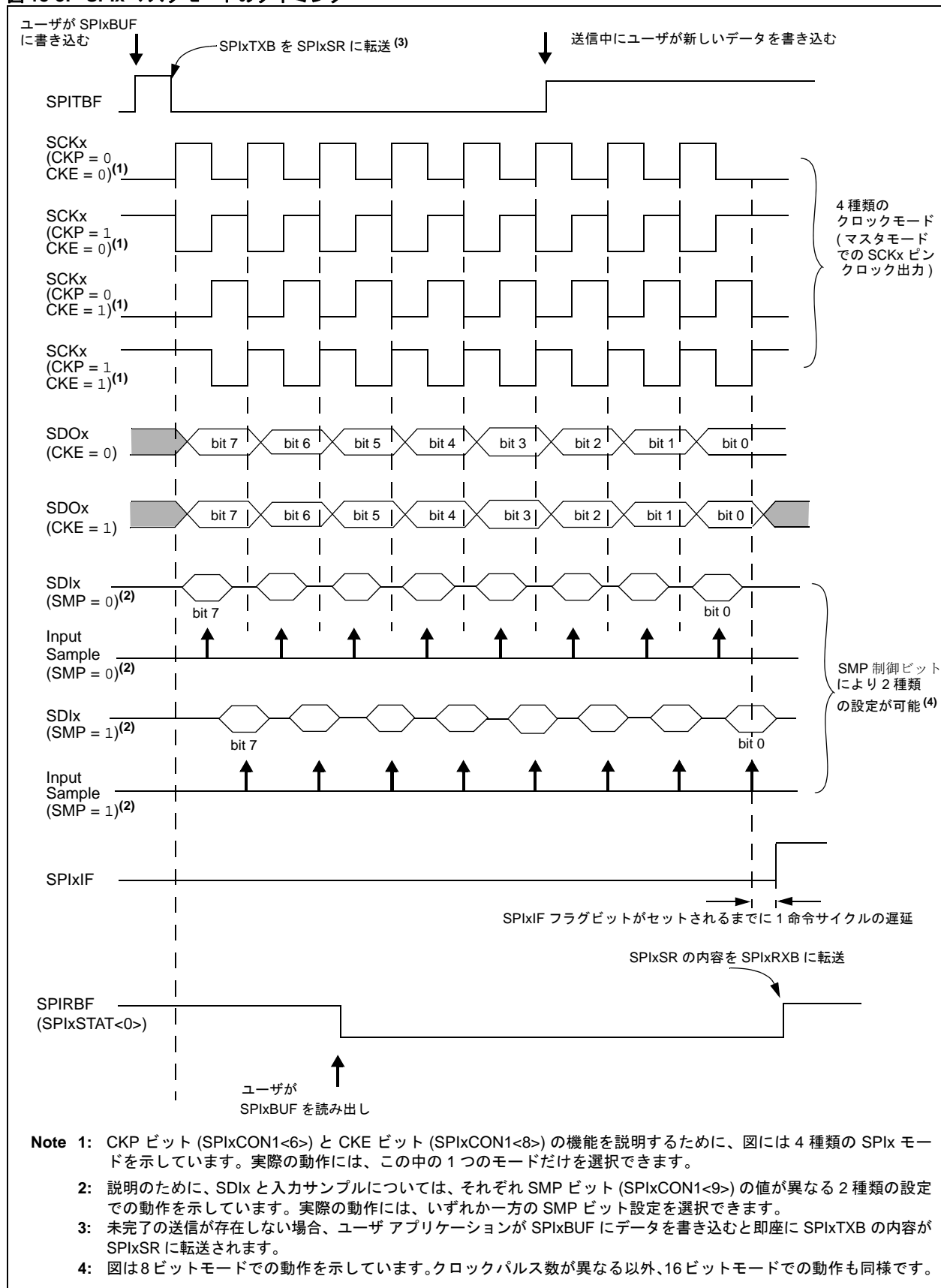
18.3.2.1.2 マスタモードにおける外部クロックの使用

マスタモードでは、外部データクロックを使用して動作するようにモジュールを設定する事もできます。SPIx クロック動作は、SPIx 制御レジスタ 1 (SPIxCON1<12>) の SCKx ピン無効化ビット (DISSCK) によって制御されます。

このビットをセットすると、内部データクロックは無効化されます。この場合、SCKx ピンに供給される外部クロックパルスに従ってデータが転送されます。これ以外は、内部クロックによるマスタモード動作と同じです。

Note: DISSCK ビットは SPI マスタモードでのみ利用できます。

図 18-3: SPIx マスタモードのタイミング



18.3.2.2 スレーブモード

スレーブモードでは、SCKx ピンに供給される外部クロックパルスに従ってデータを送受信します。SPIx 制御レジスタ 1 (SPIxCON1) の SPIx クロック極性選択ビット (CKP) と SPIx クロックエッジ選択ビットの設定によって、クロックパルスのどのエッジでデータ送信が発生するのかが決まります。送信するデータは SPIxBUF レジスタに書き込まれ、受信したデータは SPIxBUF レジスタから読み出されます。その他のモジュール動作はマスタモードと同じです。

18.3.2.2.1 スレーブモードのセットアップ手順

SPIx モジュールをスレーブモード動作向けにセットアップする手順は下記の通りです。

1. SPIxBUF レジスタをクリアします。
2. 割り込みを使用する場合、割り込みコントローラを下記のように設定します。
 - a) 割り込みコントローラ内の各割り込みフラグステータス レジスタ (IFS0<10> または IFS2<1>) の SPIx イベント割り込みフラグステータス ビット (SPIxIF) をクリアします。
 - b) 各 IECn レジスタの SPIx イベント割り込みイネーブルビット (SPIxIE) をセットします。
 - c) 各 IPCn レジスタの SPIx イベント割り込み優先度ビット (SPIxIP) で割り込み優先度を設定します。
3. SPIxCON1 レジスタを下記のように設定します。
 - a) SPIxCON1 レジスタのマスタモード イネーブルビット (MSTEN) をクリアする (SPIxCON1<5> = 0)
 - b) SPIxCON1 レジスタの SPIx データ入力サンプルフェイズ ビット (MSTEN) をクリアする (SPIxCON1<9> = 0)
 - c) SPIx クロックエッジ選択ビット (CKE) をセットした場合、スレーブ選択イネーブルビット (SSx) をセットして SSx ピンを有効化する (SPIxCON1<7> = 1)
4. SPIxSTAT レジスタを下記のように設定します。
 - a) SPIxSTAT レジスタの受信オーバーフロー フラグビット (SPIROV) をクリアする (SPIxSTAT<6> = 0)
 - b) SPIxSTAT レジスタの SPIx イネーブルビット (SPIEN) をセットして SPIx の動作を有効化する (SPIxSTAT<15> = 1)

例 18-2 に、SPI レジスタをスレーブモード向けに設定するサンプルコードを示します。

例 18-2: SPI のコンフィグレーション - スレーブモード

```
/* The following code shows the SPI register configuration for Slave mode */

SPI1BUF = 0;
IFS0bits.SPI1IF = 0; // Clear the Interrupt Flag
IEC0bits.SPI1IE = 0; // Disable the Interrupt

// SPI1CON1 Register Settings
SPI1CON1bits.DISSCK = 0; // Internal Serial Clock is Enabled
SPI1CON1bits.DISSDO = 0; // SDOx pin is controlled by the module
SPI1CON1bits.MODE16 = 1; // Communication is word-wide (16 bits)
SPI1CON1bits.SMP = 0; // Input data is sampled at the middle of data
// output time
SPI1CON1bits.CKE = 0; // Serial output data changes on transition
// from Idle clock state to active clock state
SPI1CON1bits.CKP = 0; // Idle state for clock is a low level; active
// state is a high level
SPI1CON1bits.MSTEN = 0; // Master mode Disabled
SPI1STATbits.SPIROV = 0; // No Receive Overflow has occurred
SPI1STATbits.SPIEN = 1; // Enable SPI module

// Interrupt Controller Settings
IFS0bits.SPI1IF = 0; // Clear the Interrupt Flag
IEC0bits.SPI1IE = 1; // Enable the Interrupt
```

18.3.2.2.2 スレーブ選択の同期

\overline{SSx} ピンは同期スレーブモードを有効にします。スレーブ選択イネーブルビット (SSEN) を「1」にセットした場合、 \overline{SSx} ピンが LOW 状態に駆動されている時にのみ送受信が有効化されます (図 18-5 参照)。 \overline{SSx} ピンを入力として使用するために、このピンでポート出力または他の周辺モジュール出力を駆動しない必要があります。SSEN ビットをセットした状態で \overline{SSx} ピンが HIGH に駆動されると、モジュールが送信の途中であっても SDOx ピンは駆動されなくなり、3 ステートになります。

中止された送信は、次に \overline{SSx} ピンが LOW に駆動された時に、SPIxTXB レジスタで保持されていた値を使用して再試行されます。SSEN ビットをセットしない場合、 \overline{SSx} ピンはスレーブモード時のモジュール動作に影響しません。

Note: CKE = 1 の場合、モジュールのタイミング要件を満たすには \overline{SSx} ピンをスレーブモードで有効にする必要があります (詳細は図 18-6 参照)。

18.3.2.2.3 SPITBF ステータスフラグの動作

スレーブモードでは、SPIxSTAT レジスタの送信バッファフル ステータスビット (SPITBF) の機能がマスタモードと異なります。

SSEN をクリア (= 0) した場合、ユーザアプリケーションが SPIxBUF に書き込んだ時に SPITBF ビットがセットされます。このビットは、モジュールが SPIxTXB の内容を SPIxSR へ転送した時にクリアされます。これは、マスタモードにおける SPITBF ビットの機能と同じです。

SSEN をセット (= 1) した場合も、ユーザアプリケーションが SPIxBUF に書き込んだ時に SPITBF ビットがセットされます。ただし、このビットは SPIx モジュールがデータ送信を完了した時にだけクリアされます。送信は \overline{SSx} ピンが HIGH に遷移すると中止され、後で再試行可能です。各データワードは、全てのビットがレシーバへ送信されるまで SPIxTXB で保持されます。

セクション 18. シリアル ペリフェラル インターフェイス (SPI)

図 18-4: SPIx スレーブモードのタイミング図 (スレーブ選択ピンを無効化した場合)⁽³⁾

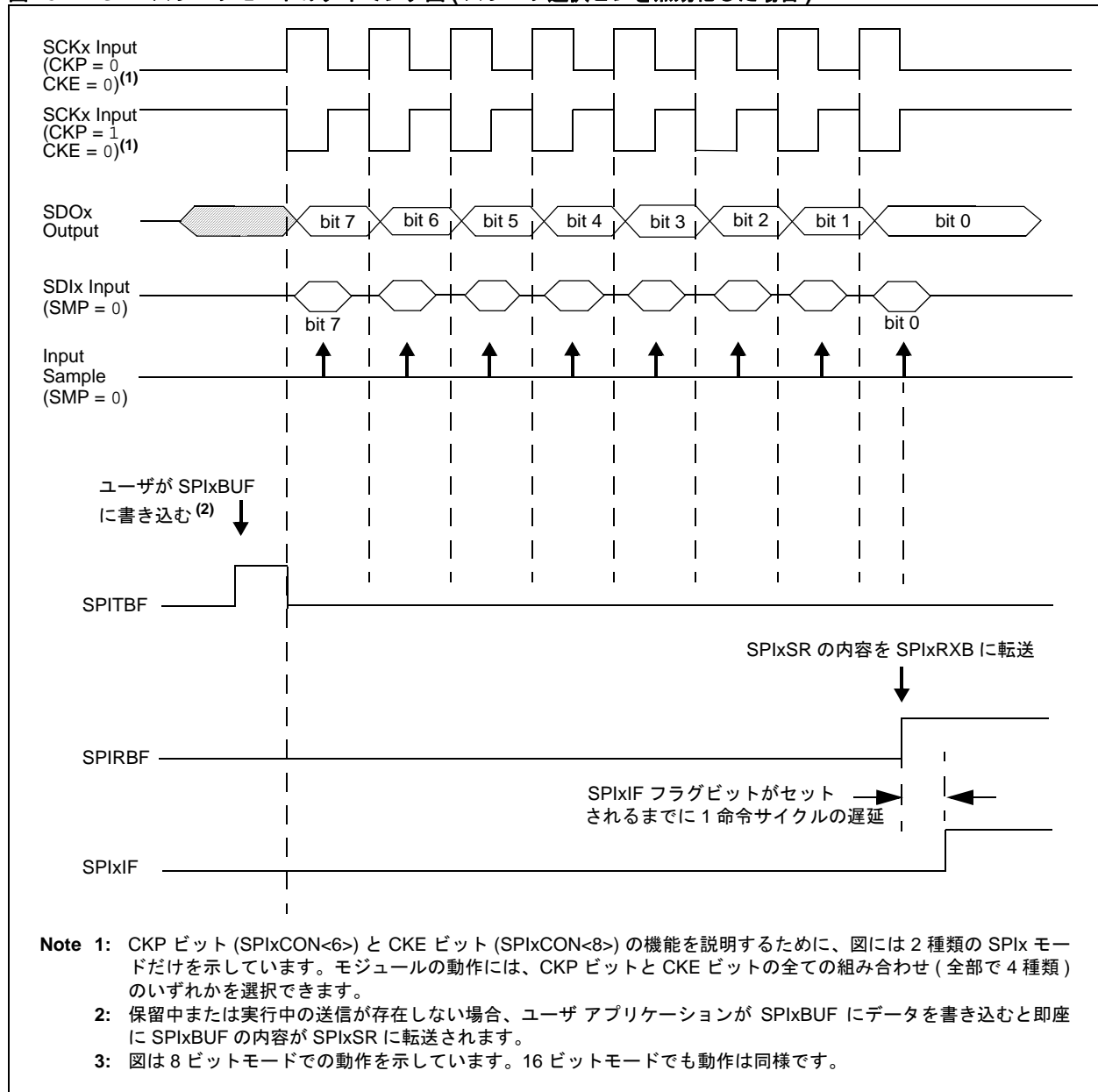
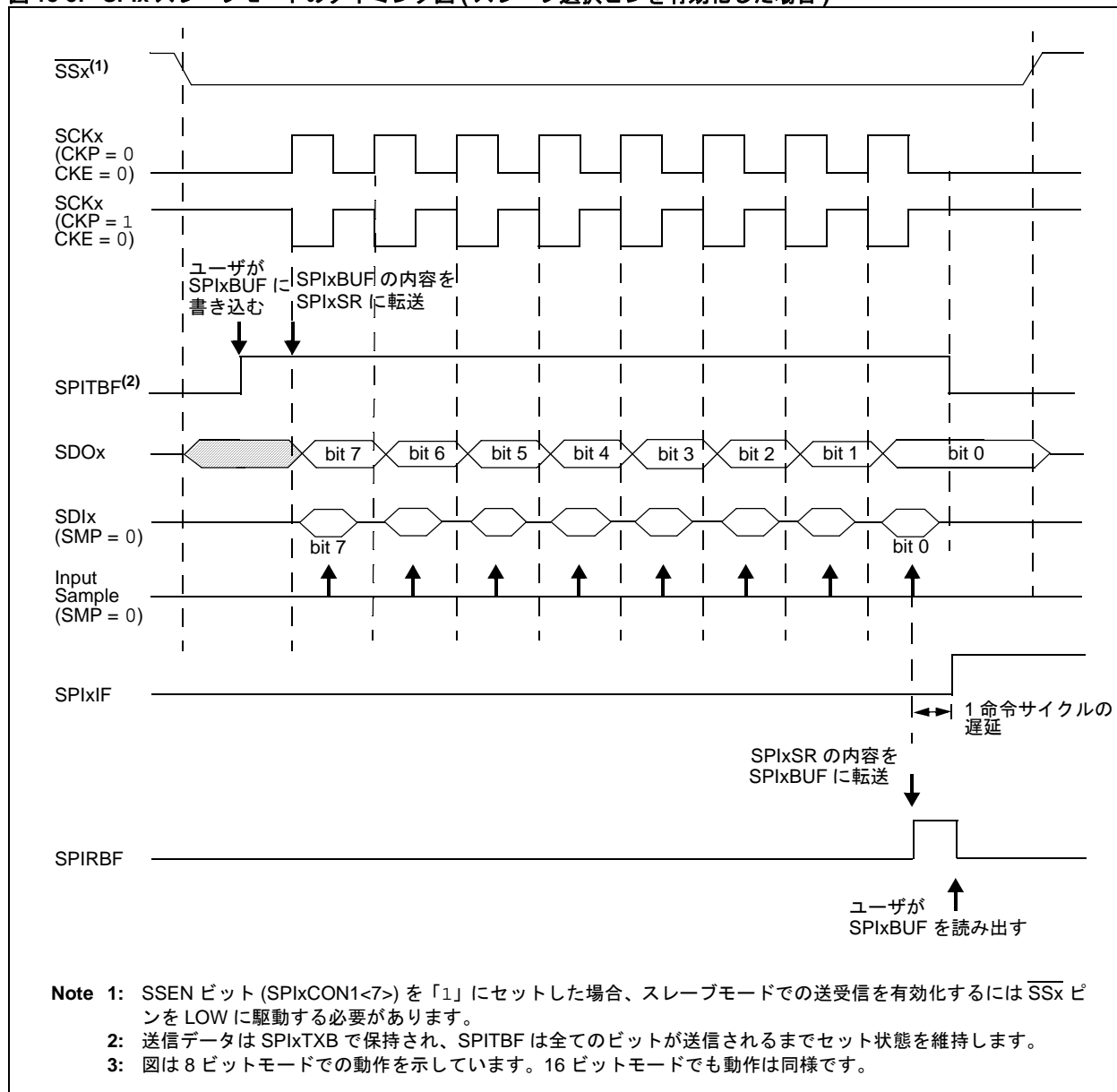
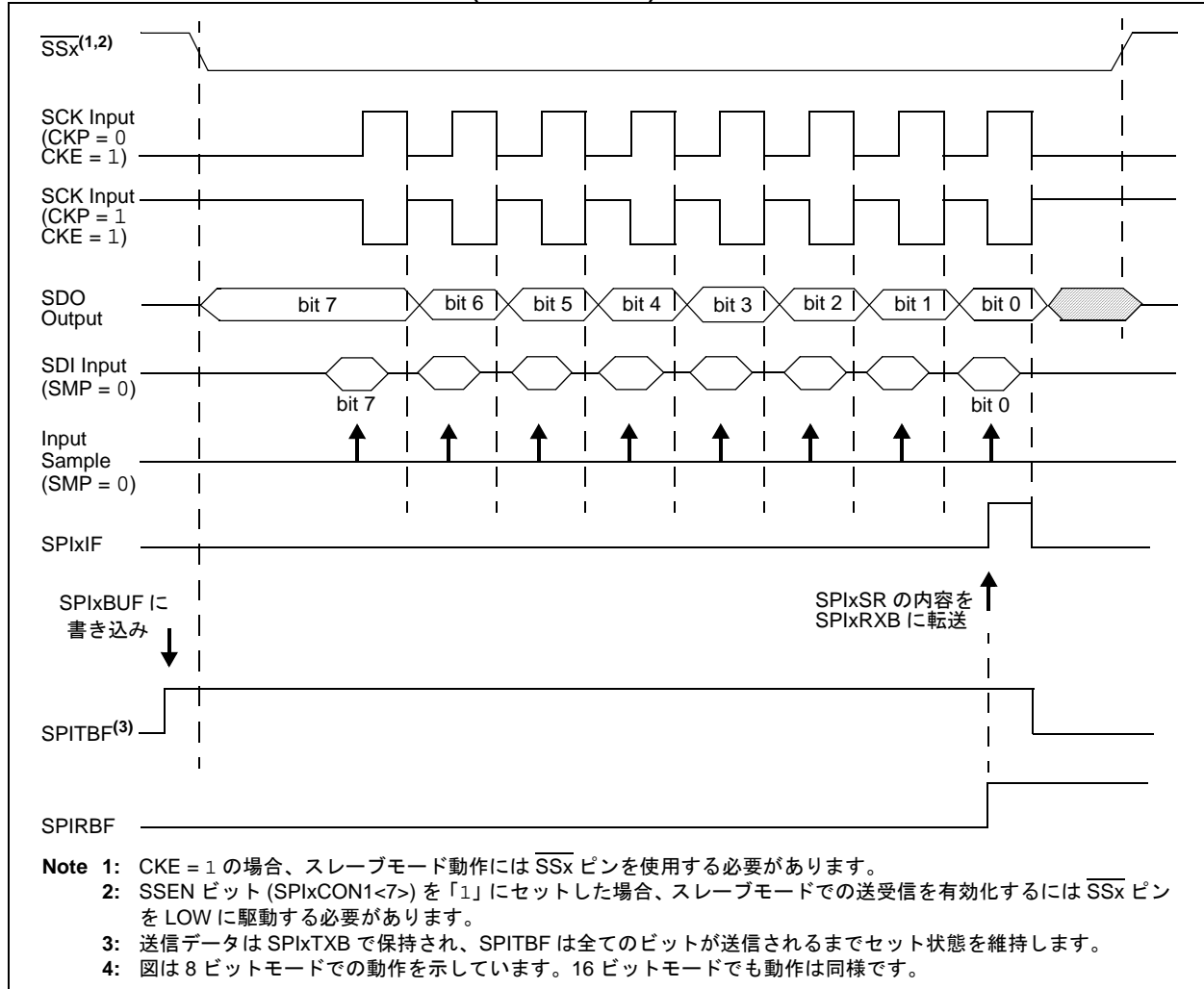


図 18-5: SPIx スレーブモードのタイミング図 (スレーブ選択ピンを有効化した場合)⁽³⁾



セクション 18. シリアル ペリフェラル インターフェイス (SPI)

図 18-6: SPIx スレーブモードのタイミング (CKE = 1 の場合)⁽⁴⁾



18

シリアルペリフェラル
インターフェイス
(SPI)

18.3.3 フレーム化 SPIx モード

SPI モジュールは、マスターモードまたはスレーブモードでの基本的なフレーム化 SPIx プロトコルをサポートします。フレーム化 SPIx 動作の設定には下記の 4 つの制御ビットを使用します。

- フレーム化 SPIx サポートビット (FRMEN)
FRMEN ビット (SPIxCON2<15>) はフレーム化 SPIx モードを有効化します。これにより \overline{SSx} ピンは、フレーム同期パルスの入力または出力ピンとして使用されます。SSEN ビット (SPIxCON1<7>) の状態は無視されます。
- フレーム同期パルス方向制御ビット (SPIFSD)
SPIFSD ビット (SPIxCON2<14>) は、 \overline{SSx} ピンの方向 (入力または出力) を指定します。この設定により、モジュールがフレーム同期パルスを受信するのか生成するのかが決まります。
- フレーム同期パルス極性ビット (FRMPOL)
FRMPOL ビット (SPIxCON2<13>) は、各 SPIx データフレームに対して生成されるフレーム同期パルスの極性 (アクティブ HIGH またはアクティブ LOW) を選択します。
- フレーム同期パルスエッジ選択ビット (FRMDLY)
FRMDLY ビット (SPIxCON2<1>) は、同期パルスのタイミングを先頭のシリアルクロックパルスと同時にするのか、それとも先行させるのかを選択します。

フレーム化マスターモードでは、SPIx モジュールはフレーム同期パルスを生成し、 \overline{SSx} ピンを介してこのパルスを他のデバイスへ供給します。

フレーム化スレーブモードでは、SIPx モジュールは \overline{SSx} ピンを介して他から供給されるフレーム同期パルスを使用します。

Note: 全てのフレーム化 SPIx モードでは、 \overline{SSx} および SCKx ピンを使用する必要があります。

フレーム化 SPIx モードは、フレーム化しない通常のマスタおよびスレーブモードとの組み合わせで使用できます。このため下記の 4 種類の組み合わせによるフレーム化 SPIx コンフィグレーションが可能です。

- SPIx マスタモードとフレーム化マスタモード
- SPIx マスタモードとフレーム化スレーブモード
- SPIx スレーブモードとフレーム化マスタモード
- SPIx スレーブモードとフレーム化スレーブモード

これらのモードによって、SPIx モジュールがシリアルクロックとフレーム化同期パルスを生じ生成するかどうかが決まります。

FRMEN = 1 かつ MSTEN = 1 の場合、SCKx ピンは出力として機能し、このピンから出力される SPI クロックはフリーランニング クロックとなります。

FRMEN = 1 かつ MSTEN = 0 の場合、SCKx ピンは入力として機能します。SCKx ピンに供給される源クロックはフリーランニング クロックであるとみなされます。

クロックの極性は CKP ビットによって決まります。フレーム化 SPI モードには CKE ビットを使用しません。ユーザ ソフトウェアはこのビットを「0」に設定する必要があります。

CKP = 0 の場合、フレーム同期パルス出力と SDOx データ出力は、SCKx ピンにおけるクロックパルス立ち上がりエッジで状態が変化します。入力データは、SDIx 入力ピンにおけるシリアルクロック立ち下がりエッジでサンプリングされます。

CKP = 1 の場合、フレーム同期パルス出力と SDOx データ出力は、SCKx ピンにおけるクロックパルス立ち下がりエッジで状態が変化します。入力データは、SDIx 入力ピンにおけるシリアルクロック立ち上がりエッジでサンプリングされます。

18.3.3.1 フレーム化マスタモードとフレーム化スレーブモード

SPIx 制御レジスタ 2 (SPIxCON2<14>) で SPIFSD = 0 の場合、SPIx モジュールはフレーム化マスタモードで動作します。このモードでは、ユーザ ソフトウェアが送信データを SPIxBUF に書き込んだ時 (すなわち SPIxTXB レジスタに送信データが書き込まれた時) に、モジュールがフレーム同期パルスの生成を開始します。フレーム同期パルスの終端で、SPIxTXB が SPIxSR へ転送され、データの送受信が開始されます。

SPIFSD = 1 の場合、モジュールはフレーム化スレーブモードで動作します。このモードでは、外部ソースがフレーム同期パルスを生じ生成します。モジュールがフレーム同期パルスを検出した時に、SPIxTXB レジスタの内容が SPIxSR に転送され、データの送受信が開始されます。ユーザ アプリケーションは、フレーム同期パルスを受信する前に、正しい送信データを SPIxBUF に書き込む必要があります。

Note: フレーム同期パルスを受信すると、データが SPIxBUF に書き込まれたかどうかに関係なく、送信が開始されます。SPIxBUF にデータを書き込まなかった場合、SPIxTXB の既存の内容が送信されます。

18.3.3.2 SPIx マスタ / フレーム化マスタモード

SPIx マスタ / フレーム化マスタモードでは、SPIx モジュールはクロックとフレーム同期信号の両方を生成します (図 18-7 参照)。このコンフィグレーションを有効にするには、MSTEN および FRMEN ビットを「1」、SPIFSD ビットを「0」に設定します。

このモードは、モジュールが送信中であるかどうかに関係なく、SCKx ピンでシリアルクロックを常時出力します。SPIxBUF が書き込まれた時、 \overline{SSx} ピンは SCKx クロックの適切な送信エッジでアクティブ状態 (FRMPOL ビットで定義) に駆動され、1 クロック間アクティブ状態を維持します。FRMDLY 制御ビット (SPIxCON2<1>) をクリアした場合、フレーム同期パルスはデータ送信に先行して生成されます (図 18-8 参照)。FRMDLY ビットをセットした場合、フレーム同期パルスはデータ送信の開始と同時に生成されます (図 18-9 参照)。モジュールは、SCKx ピンの次の送信エッジでデータの送信を開始します。

セクション 18. シリアル ペリフェラル インターフェイス (SPI)

図 18-7: SPIx マスタ / フレーム化マスタの接続図

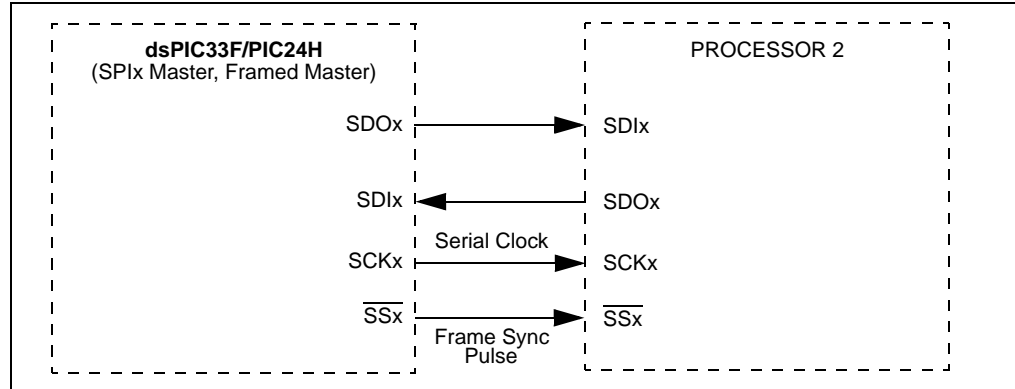


図 18-8: SPIx マスタ / フレーム化マスタのタイミング (FRMDLY = 0)

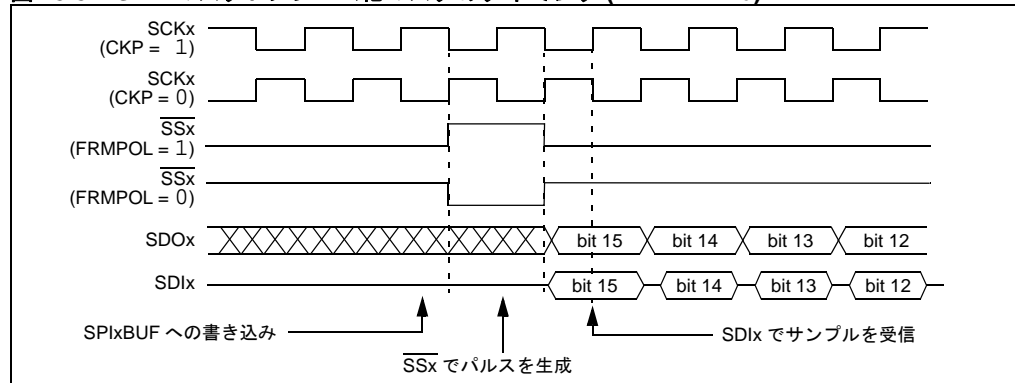
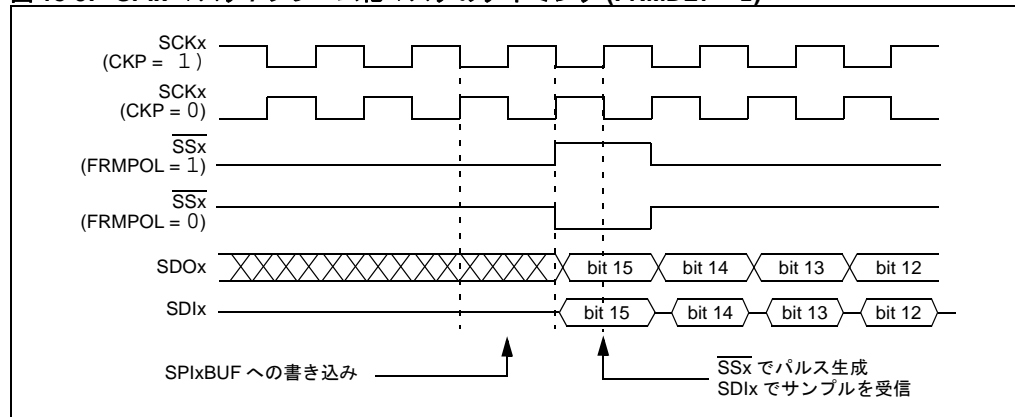


図 18-9: SPIx マスタ / フレーム化マスタのタイミング (FRMDLY = 1)



18.3.3.3 SPIx マスタ / フレーム化スレーブモード

SPIx マスタ / フレーム化スレーブモードでは、モジュールはクロック信号を生成しますが、データ送信にはスレーブモジュール フレーム同期信号を使用します (図 18-10 参照)。このモードを有効にするには、MSTEN、FRMEN、SPIFSD ビットを「1」にセットします。

このモードでは SSx ピンは入力として機能します。この入力は、SPIx クロックのサンプルエッジでサンプリングされます。この入力アクティブ状態としてサンプリングされると、SPIx クロックの次の送信エッジでデータが送信されます。送信が完了すると割り込みフラグ SPIxIF がセットされます。ユーザ アプリケーションは、SSx ピンでアクティブ信号を受信する前に、正しいデータを SPIxBUF に書き込む必要があります。

図 18-10: SPIx マスタ / フレーム化スレーブの接続図

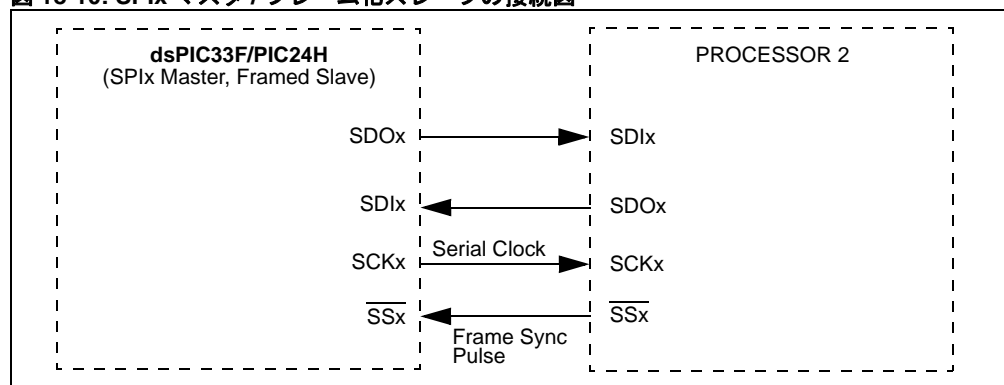


図 18-11: SPIx マスタ / フレーム化スレーブのタイミング (FRMDLY = 0)

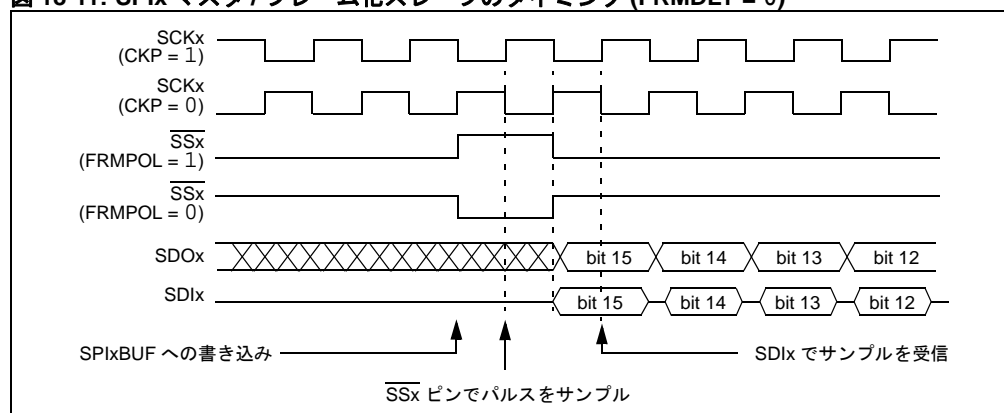
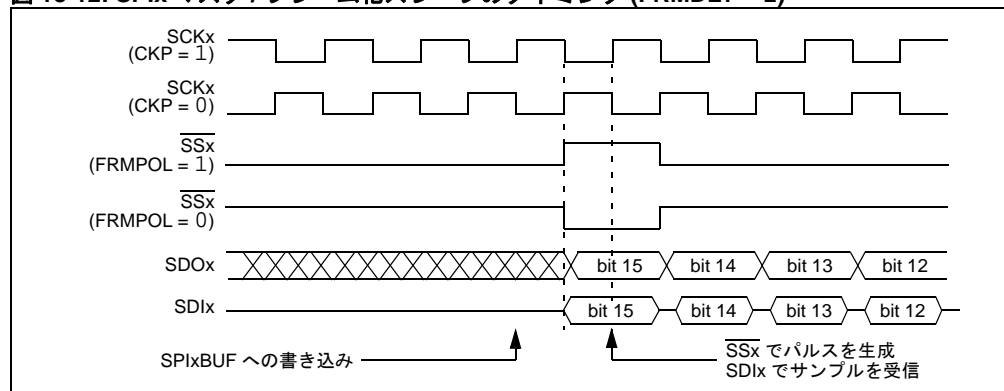


図 18-12: SPIx マスタ / フレーム化スレーブのタイミング (FRMDLY = 1)

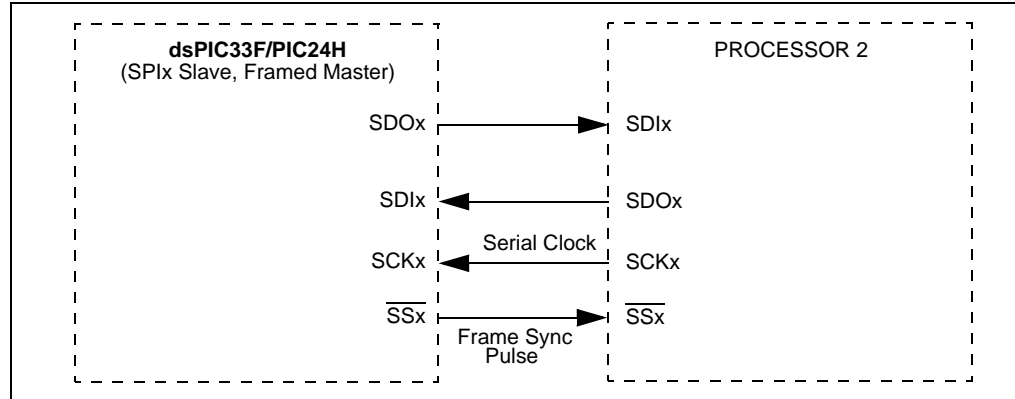


18.3.3.4 SPIx スレーブ / フレーム化マスタモード

SPIx スレーブ / フレーム化マスタモードではモジュールは SPIx スレーブとして動作し、クロックは他の SPIx モジュールから供給されますが、データ送信を制御するためにフレーム同期信号を生成します (図 18-13 参照)。このモードを有効にするには、MSTEN ビットを「0」、FRMEN ビットを「1」、SPIFSD ビットを「0」に設定します。

スレーブモードでは入力 SPIx クロックは常時供給されます。SPIFSD ビットが LOW の時、SSx ピンは出力として機能します。従って SPIx 書き込み時に、モジュールは SPIx クロックの適切な送信エッジで SSx ピンをアクティブ状態に駆動し、SPIx クロックの 1 サイクル間アクティブ状態を維持します。適切な SPIx クロック送信エッジでデータの送信が開始されます。

図 18-13: SPIx スレーブ / フレーム化マスタの接続図

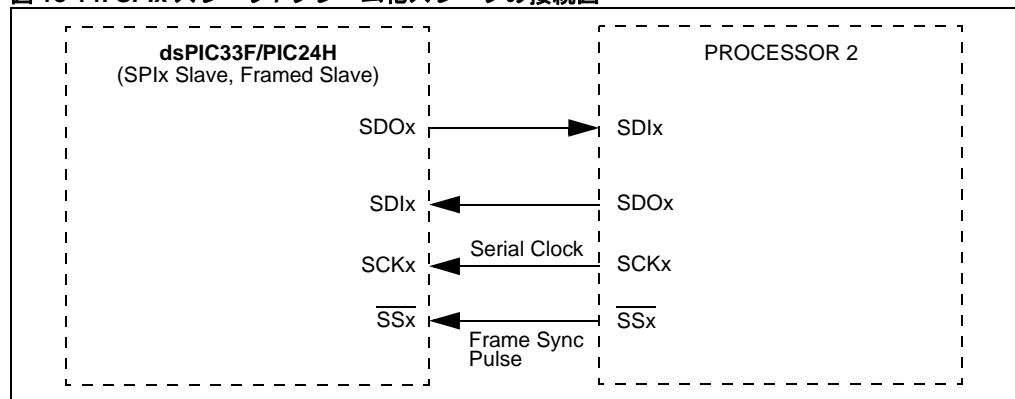


18.3.3.5 SPIx スレーブ / フレーム化スレーブモード

SPIx スレーブ / フレーム化スレーブモードでは、モジュールのクロックとフレーム同期信号は、両方ともマスタモジュールから供給されます (図 18-14 参照)。このモードを有効にするには、MSTEN ビットを「0」、FRMEN ビットを「1」、SPIFSD ビットを「1」に設定します。

このモードでは、SCKx ピンと $\overline{\text{SSx}}$ ピンの両方が入力として機能します。 $\overline{\text{SSx}}$ ピンは SPIx クロックのサンプルエッジでサンプリングされます。SSx がアクティブ状態としてサンプリングされると、SCKx の適切な送信エッジでデータが送信されます。

図 18-14: SPIx スレーブ / フレーム化スレーブの接続図



18.3.4 SPIx の受信専用動作

DISSDO 制御ビット (SPIxCON1<11>) をセットすると、SDOx ピンでの送信が無効化されます。これにより、SPIx モジュールを受信専用動作モードに設定できます。DISSDO ビットをセットした場合、SDOx ピンは対応するポート機能によって制御されます。DISSDO 機能は全ての SPIx 動作モードで使用できます。

18.3.5 SPIx のエラー対応

新しいデータが SPIxSR にシフトインされ、かつ SPIxBUF の既存内容が読み出されていない場合、SPIROV ビット (SPIxSTAT<6>) がセットされます。SPIxSR 内の受信データは転送されず、SPIROV ビットがクリアされるまで以降のデータ受信は無効化されます。モジュールは SPIROV ビットを自動的にクリアしないため、ユーザ アプリケーションでこのビットをクリアする必要があります。

SPIxSTAT レジスタの SPIx 受信バッファフル ステータスビット (SPIRBF) または SPIx 送信バッファフル ステータスビット (SPITBF) がセットされた時、SPIx イベント割り込みフラグ ステータスビット (SPIxIF) がセットされます。割り込みフラグはハードウェアでクリアされないため、ソフトウェアでリセットする必要があります。実際の SPIx 割り込みは、IECn レジスタの対応する SPIxIE ビットがセットされている場合にのみ生成されます。

加えて、SPIROV ビットがセットされた時に SPIx エラー割り込みフラグビット (SPIxEIF) がセットされます。この割り込みフラグはソフトウェアでクリアする必要があります。実際の SPIx エラー割り込みは、IECn レジスタの対応する SPIxEIE ビットがセットされている場合にのみ生成されます。

18.4 マスタモードのクロック周波数

マスタモードでは、命令サイクルクロック (Tcy) が SPIx モジュールに供給されます。このクロックは、プライマリ プリスケアラ (SPIxCON1 レジスタのプライマリ プリスケールビット (PPRE<1:0>) で指定) とセカンダリ プリスケアラ (SPIxCON1 レジスタのセカンダリ プリスケールビット (SPRE<2:0>) で指定) によってプリスケールされます。プリスケールされた命令クロックは、シリアルクロックとして SCKx ピンから外部デバイスへ供給されます。

Note: 通常の SPI モードでは、SCKx 信号クロックはフリーランニングではありません。通常モードでは、SCKx は SPIxBUF にデータが書き込まれた時にのみパルスを 8 または 16 回生成しますが、フレーム化モードではパルスを常時生成します。

式 18-1 に、プライマリおよびセカンダリ プリスケアラの設定値から SCKx クロック周波数を求める計算式を示します。

式 18-1: SPI クロックの周波数

$$F_{SCK} = \frac{F_{CY}}{\text{プライマリ プリスケアラ} \times \text{セカンダリ プリスケアラ}}$$

SPIx クロック周波数 (kHz) の例を表 18-1 に示します。

Note: 一部のクロック周波数はサポートされません。詳細は、各デバイスのデータシートに記載されている SPIx タイミング仕様を参照してください。

表 18-1: SCKx 周波数の例 ⁽¹⁾

Fcy = 40 MHz		セカンダリ プリスケアラ設定				
		1:1	2:1	4:1	6:1	8:1
プライマリ プリスケアラ設定	1:1	無効	無効	10000	6666.67	5000
	4:1	10000	5000	2500	1666.67	1250
	16:1	2500	1250	625	416.67	312.50
	64:1	625	312.5	156.25	104.17	78.125
Fcy = 5 MHz						
プライマリ プリスケアラ設定	1:1	5000	2500	1250	833	625
	4:1	1250	625	313	208	156
	16:1	313	156	78	52	39
	64:1	78	39	20	13	10

Note 1: SCKx 周波数の単位は kHz です。

18.5 DMA を使用する SPI 動作

dsPIC33F/PIC24H の一部のデバイスでは、ダイレクト メモリアクセス (DMA) モジュールを使用して、CPU に負荷をかけずに CPU と SPI モジュール間でデータを転送できます。デバイスが DMA を内蔵するかどうかは、各デバイスのデータシートを参照してください。詳細はセクション 22. 「ダイレクト メモリアクセス (DMA)」 (DS70182) を参照してください。

DMA チャンネルを SPI レシーバに割り当てた場合、SPI モジュールはデータが SPI から RAM へ転送可能となるたびに DMA 要求を発行します。DMA は SPIxBUF レジスタから RAM へデータを転送し、あらかじめ指定された数のデータを転送した後に CPU 割り込みを生成します。

同様に、DMA チャンネルを SPI トランスミッタに割り当てた場合、SPI モジュールは送信に成功するたびに DMA 要求を発行します。DMA は各 DMA 要求の後に新たなデータを SPIxBUF レジスタに転送し、事前に決められた数のデータを転送した後に CPU 割り込みを生成します。DMA チャンネルは片方向であるため、SPI を受信と送信の双方向で使用するには 2 つの DMA チャンネルが必要です。各 DMA チャンネルは、表 18-2 のように初期化する必要があります。

表 18-2: SPI を DMA に割り当てるための DMA チャンネル レジスタの初期化

周辺モジュールから DMA への割り当て	DMAxREQ レジスタの IRQSEL<6:0> ビット	周辺モジュールから読み込み / 周辺モジュールへ書き込む DMAxPAD レジスタ値
SPI1TX/RX – SPI1 送信 / 受信	0001010	0x0248 (SPI1BUF)
SPI2TX/RX – SPI2 送信 / 受信	0100001	0x0268 (SPI2BUF)

SPI 周辺モジュールの DMA 送受信の開始方法は、SPI データの転送方向とスレーブ / マスタのどちらのモードで転送するかによって異なります。

- マスタモードで送信のみ行う場合：**このコンフィグレーションは SPI データの先頭ブロックを送信するまで DMA 要求を発行しません。DMA 転送を開始するには、ユーザアプリケーションで最初に DMA 手動送信モードを使用してデータを送信するか、あるいは最初に DMA を使用せずに SPI バッファ (SPIxBUF) にデータを書き込む必要があります。
- マスタモードで受信のみ行う場合：**このコンフィグレーションは SPI データの先頭ブロックを受信するまで DMA 要求を発行しません。しかしマスタモードでは、SPI が先に送信しないとデータを受信しません。DMA 転送を開始するには、ユーザアプリケーションで DMA Null データ書き込みモードを使用して DMA 手動送信モードを開始する必要があります。
- マスタモードで受信と送信を行う場合：**このコンフィグレーションは SPI データの先頭ブロックを受信するまで DMA 要求を発行しません。しかしマスタモードでは、SPI が先に送信しないとデータを受信しません。DMA 転送を開始するには、ユーザアプリケーションで最初に DMA 手動送信モードを使用してデータを送信するか、あるいは最初に DMA を使用せずに SPI バッファ (SPIxBUF) にデータを書き込む必要があります。
- スレーブモードで送信のみ行う場合：**このコンフィグレーションは SPI データの先頭ブロックを受信するまで DMA 要求を発行しません。DMA 転送を開始するには、ユーザアプリケーションで最初に DMA 手動送信モードを使用してデータを送信するか、あるいは最初に DMA を使用せずに SPI バッファ (SPIxBUF) にデータを書き込む必要があります。
- スレーブモードで受信のみ行う場合：**このコンフィグレーションは、最初の SPI データを受信すると即座に DMA 要求を発行します。DMA 転送を開始するためにユーザアプリケーションによる特別な手順は不要です。
- スレーブモードで受信と送信を行う場合：**このコンフィグレーションは先頭の SPI データブロックを受信するまで DMA 要求を発行しません。DMA 転送を開始するには、ユーザアプリケーションで最初に DMA 手動送信モードを使用してデータを送信するか、あるいは最初に DMA を使用せずに SPI バッファ (SPIxBUF) にデータを書き込む必要があります。

18.5.1 DMA を使用する SPI 送受信

例 18-3 に DMA を使用する SPI 送受信のサンプルコードを示します。このコードは SPI モジュールをマスタモードに設定します。2 つの DMA チャンネルを使用し、チャンネル 0 をデータ送信用、チャンネル 1 をデータ受信用に使用します。

下記のパラメータにより DMA チャンネル 0 を SPI 送信用に設定します。

- RAM から SPI への連続的なデータ転送
- ポスト インクリメントによるレジスタ間接
- 2 つのピンポンバッファを使用
- バッファあたり 16 転送

下記のパラメータにより DMA チャンネル 1 を SPI 受信用に設定します。

- SPI から RAM への連続的なデータ転送
- ポスト インクリメントによるレジスタ間接
- 2 つのピンポンバッファを使用
- バッファあたり 16 転送

例 18-3: DMA を使用する SPI 送受信

SPI1 をマスタモードにセットアップする :

```
// Interrupt Controller Settings
IFS0bits.SPI1IF = 0;

// SPI1CON1 Register Settings
SPI1CON1bits.MODE16 = 1; // Communication is word-wide (16 bits)
SPI1CON1bits.MSTEN = 1; // Master mode Enabled

// SPI1CON2 Register Settings
SPI1CON2bits.FRMEN = 0; // Framed mode Disabled

// SPI1STAT Register Settings
SPI1STATbits.SPIIDL = 0; // Continue module operation in Idle mode
SPI1STATbits.BUFE = 0; // Buffer Length = 1 Word
SPI1STATbits.SPIROV = 0; // No Receive Overflow has occurred
SPI1STATbits.SPIEN = 1; // Enable SPI module

// Force First Word after Enabling SPI
DMA0REQbits.FORCE = 1;
while (DMA0REQbits.FORCE == 1)

IEC0bits.SPI1IE = 1;
```

DMA チャンネル 0 を連続ピンポンモードによる送信用にセットアップする :

```
unsigned int TxBufferA[16] __attribute__((space(dma)));
unsigned int TxBufferB[16] __attribute__((space(dma)));

IFS0bits.DMA0IF = 0;
IEC0bits.DMA0IE = 1;
DMAC0 = 0;
DMA0CON = 0x2002;
DMA0STA = __builtin_dmaoffset(TxBufferA);
DMA0STB = __builtin_dmaoffset(TxBufferB);
DMA0PAD = (volatile unsigned int) &SPI1BUF;
DMA0CNT = 15;
DMA0REQ = 0x000A;
DMA0CONbits.CHEN = 1;
```

例 18-3: DMA を使用する SPI 送受信 (続き)

DMA チャンネル 1 を連続ピンポンモードによる受信用にセットアップする:

```
unsigned int RxBufferA[16] __attribute__((space(dma)));
unsigned int RxBufferB[16] __attribute__((space(dma)));

IFS0bits.DMA1IF = 0;
IEC0bits.DMA1IE = 1;
DMA1CON = 0x0002;
DMA1STA = __builtin_dmaoffset(RxBufferA);
DMA1STB = __builtin_dmaoffset(RxBufferB);
DMA1PAD = (volatile unsigned int) &SPI1BUF;
DMA1CNT = 15;
DMA1REQ = 0x000A;
DMA1CONbits.CHEN = 1;
```

SPI および DMA 割り込みハンドラ:

```
void __attribute__((__interrupt__)) _SPI1Interrupt(void)
{
    IFS0bits.SPI1IF = 0;
}

void __attribute__((__interrupt__)) _DMA0Interrupt(void)
{
    static unsigned int BufferCount = 0; // Keep record of which buffer
    // contains TX Data
    if(BufferCount == 0)
    {
        TxData(BufferA); // Transmit SPI data in
        // DMA RAM Primary buffer
    }
    else
    {
        TxData(BufferB); // Transmit SPI data in
        // DMA RAM Secondary buffer
    }
    BufferCount ^= 1;
    IFS0bits.DMA0IF = 0; // Clear the DMA0 Interrupt Flag
}

void __attribute__((__interrupt__)) _DMA1Interrupt(void)
{
    static unsigned int BufferCount = 0; // Keep record of which buffer
    // contains RX Data
    if(BufferCount == 0)
    {
        ProcessRxData(BufferA); // Process received SPI data in
        // DMA RAM Primary buffer
    }
    else
    {
        ProcessRxData(BufferB); // Process received SPI data in
        // DMA RAM Secondary buffer
    }
    BufferCount ^= 1;
    IFS0bits.DMA1IF = 0; // Clear the DMA1 Interrupt Flag
}
```

18.5.2 SPI と DMA の Null データ書き込みモード

マスタモードに設定した SPI でデータの受信だけが必要な場合、SPI クロックを起動して外部データを受信するには、何らかのデータを SPI 送信バッファに書き込む必要があります。このような場合、DMA の Null データ書き込みモードを使用します。DMA Null データ書き込みモードの詳細はセクション22.「ダイレクトメモリアクセス(DMA)」(DS70182)を参照してください。

18.6 省電力モード時の動作

dsPIC33F/PIC24H ファミリのデバイスは、通常 (フルパワー) モード以外に PWRSAV 命令で呼び出す 2 つの省電力モードを備えます。選択した SPIx のモードによっては、省電力モードへの移行がモジュールの動作にも影響します。

18.6.1 スリープモード

デバイスがスリープモードに移行するとシステムクロックが停止します。スリープモードへの移行による影響は、設定されていたスリープモード時のモジュールの動作モード (マスタまたはスレーブ) によって異なります。

18.6.1.1 マスタモードの場合

SPIx モジュールがマスタ動作に設定されていた時にスリープモードへ移行すると、モジュールは下記の影響を受けます。

- SPIx モジュールの baud レート ジェネレータ (BRG) が停止してリセットされる
- スリープモード中にトランスミッタとレシーバは停止する (トランスミッタまたはレシーバは、移行時に中断した転送をウェイクアップ時に再開しません)
- スリープモードに移行する時に SPIx モジュールが実行中であった送受信は中止される (送信または受信の途中でスリープモードに移行する事を回避するための自動的な方法は用意されていません。このためユーザ ソフトウェアはスリープへの移行を SPIx の動作に同期する事によって転送の中止を回避する必要があります)

18.6.1.2 スレーブモードの場合

スレーブモードの場合、SCKx のクロックパルスは外部から供給されるため、モジュールはスリープモードでも機能し続けます。スリープへの移行中のトランザクションは全て完全に実行されます。トランザクションが完了すると SPIRBF フラグがセットされます。この結果、SPIxIF ビットがセットされます。

SPIx 割り込みが有効 (SPIxIE = 1) であれば、デバイスはスリープからウェイクアップします。SPIx 割り込みの優先度が CPU 優先度よりも高い場合、コード実行は SPIx 割り込みベクタ位置から再開されます。これ以外の場合、コード実行は直前のスリープモードを呼び出した PWRSAV 命令の次の命令から再開されます。モジュールがスレーブデバイスとして動作している場合、スリープモードへの移行時にモジュールはリセットされません。

SPIx モジュールがスリープモードへ移行する時またはスリープモードから復帰する時にレジスタの内容は影響を受けません。

18.6.2 アイドルモード

デバイスがアイドルモードに移行しても、システムクロック源は機能し続けます。アイドルモード時にモジュールの機能を停止するかどうかは、SPISIDL ビット (SPIxSTAT<13>) で選択します。

SPISIDL = 1 の場合、SPIx モジュールはアイドルモードに移行した時に通信を停止します。この場合、モジュールはスリープモード時と同様に動作します。PSIDL = 0 (既定値) の場合、モジュールはアイドルモード中も動作を続けます。

18.8 関連アプリケーション ノート

本セクションに関連するアプリケーション ノートの一覧を下に記載します。これらのアプリケーション ノートは dsPIC33F/PIC24H デバイスファミリ向けではありません。ただし概念は共通しており、変更が必要であったり制限事項が存在するものの利用が可能です。シリアル ペリフェラル インターフェイス (SPI) モジュールに関連する最新のアプリケーション ノートは以下の通りです。

タイトル	アプリケーション ノート番号
マイクロチップ社製 MCP41XXX および MCP42XXX デジタル ポテンショメータと PIC® マイクロコントローラ間のインターフェイス	AN746
マイクロチップ社製 MCP3201 アナログ / デジタル コンバータと PIC® マイクロコントローラ間のインターフェイス	AN719

Note: dsPIC33F/PIC24H ファミリ関連のアプリケーション ノートとサンプルコードはマイクロチップ社のウェブサイト (www.microchip.com) でご覧になれます。

18.9 改訂履歴

リビジョン A (2007 年 4 月)

本書の初版

リビジョン B (2008 年 1 月)

このリビジョンでの変更内容は以下の通りです。

- レジスタ :
 - SPIxCON1:SPIx 制御レジスタ 1 (レジスタ 18-2 参照) に Note 1 を追加
- 上記に加えて、表現および体裁の変更等、本書全体の細部を修正

リビジョン C (2010 年 1 月)

このリビジョンでの変更内容は以下の通りです。

- 補足文書に関する情報を記載した網掛け注釈ボックスを本セクションの冒頭に追加
- ファミリ リファレンス マニュアルのタイトルを dsPIC33F から dsPIC33F/PIC24H に変更
- 本書内の「dsPIC33F」を全て「dsPIC33F/PIC24H」に変更
- 図 :
 - 図 18-3 内の 4 番目のクロックモードの CKE ビット値を「0」から「1」に変更
 - 図 18-7 内のテキスト「SPIx スレーブ、フレーム化スレーブ」を「SPIx マスタ、フレーム化マスタ」に変更
 - 図 18-13 内のテキスト「SPIx スレーブ / フレーム化スレーブ」を「SPIx スレーブ、フレーム化マスタ」に変更
 - 図 18-14 内のテキスト「SPIx マスタ / フレーム化スレーブ」を「SPIx スレーブ、フレーム化スレーブ」に変更
- レジスタ 18-2 に Note 3 を追加
- 上記に加えて、表現および体裁の変更等、本書全体の細部を修正

ISBN: 978-1-60932-867-2

NOTE: