

セクション 19. I²C™ (Inter-Integrated Circuit™)

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。

19.1 概要	19-2
19.2 I ² C バスの特性	19-4
19.3 制御 / ステータス レジスタ	19-7
19.4 I ² C 動作の有効化	19-13
19.5 シングルマスタ環境でマスタとして通信する	19-15
19.6 マルチマスタ環境でマスタとして通信する	19-28
19.7 スレーブとしての通信	19-31
19.8 I ² C バスの接続に関する注意事項	19-46
19.9 PWSAV 命令実行時のモジュール動作	19-48
19.10 周辺モジュール ディセーブル (PMD) レジスタ	19-48
19.11 リセットの影響	19-48
19.12 レジスタマップ	19-49
19.13 設計のヒント	19-50
19.14 関連アプリケーション ノート	19-51
19.15 改訂履歴	19-52

19.1 概要

I²C (Inter-Integrated Circuit) モジュールは、周辺モジュール (EEPROM、ディスプレイ ドライバ、A/D コンバータ等) または他のマイクロコントローラ デバイスとの通信に使用するシリアル インターフェイスです。

I²C モジュールは下記の I²C システム内で動作可能です。

- スレーブデバイスとして
- シングルマスタ システム内のマスタデバイスとして (スレーブデバイスとして動作する場合もあり)
- マルチマスタ システム内のマスタ / スレーブデバイスとして (バス衝突検出とバス調停が可能)

I²C モジュールは、独立したマスタ I²C ロジックと I²C スレーブロジックを備え、各ロジックはそれぞれのイベントに基づいて割り込みを生成します。マルチマスタ システムでは、ソフトウェアは単純にマスタ コントローラとスレーブ コントローラに分割されます。

I²C マスタロジックがアクティブな時、スレーブロジックもアクティブ状態を維持してバスの状態を監視し、シングルマスタまたはマルチマスタから受信する可能性のあるメッセージを監視します。マルチマスタ システムでバス調停が発生してもメッセージは失われません。

マルチマスタ システムでは、システム内の他のマスタとのバス衝突が検出されます。モジュールはバス衝突によって中止されたメッセージを後で再開する方法を提供します。

I²C モジュールは baud レート ジェネレータ (BRG) を備えます。I²C の baud レート ジェネレータは、デバイス内の他のタイマリソースを消費しません。

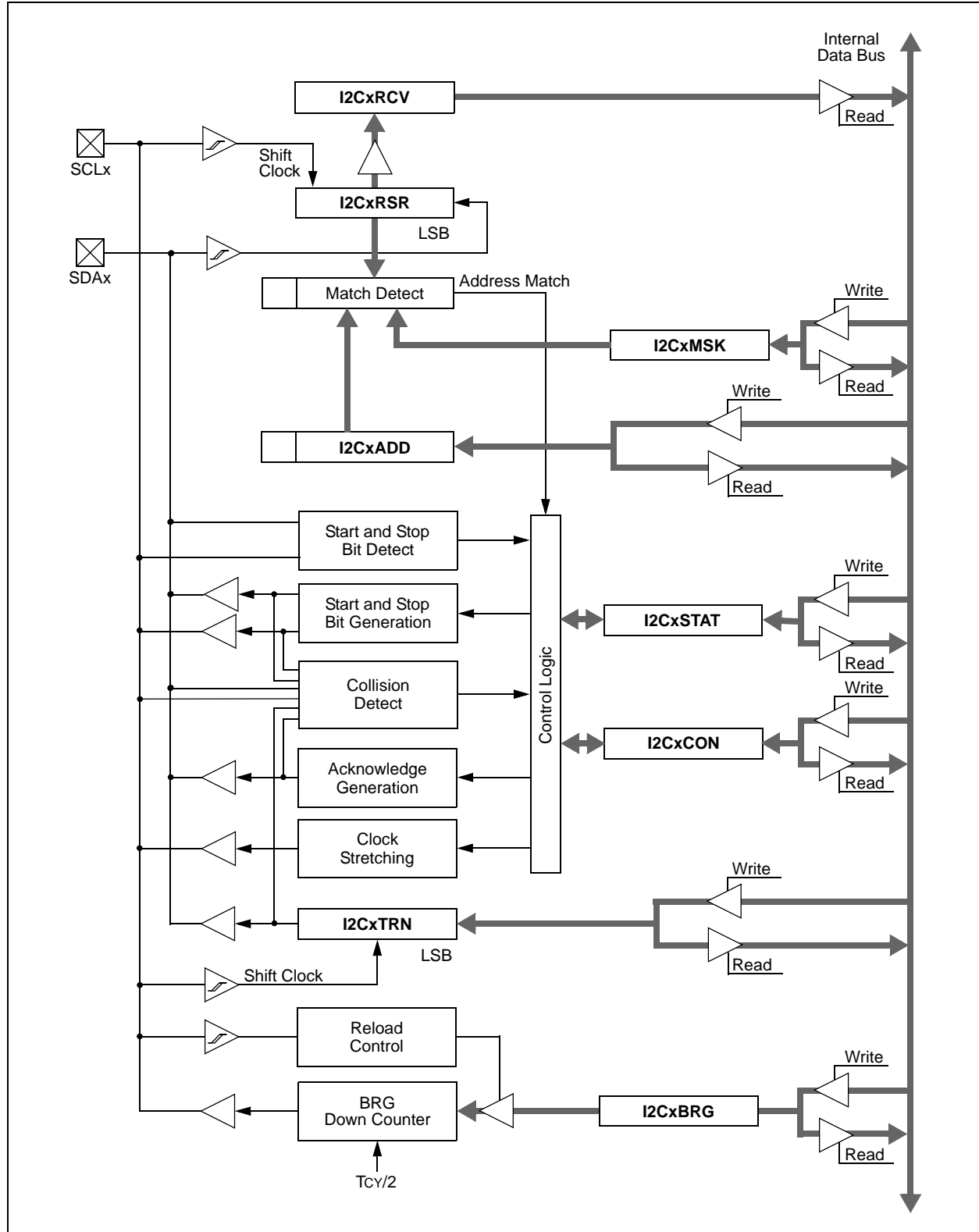
I²C の主な特長は以下の通りです。

- 独立したマスタロジックとスレーブロジック
- マルチマスタをサポート (調停によるメッセージの喪失を回避)
- スレーブモードでは、7 ビットおよび 10 ビット デバイスアドレスの検出用にアドレス マスキングを使用可能 (マスキングするビットを任意に設定可能)
- I²C プロトコルで定義されているゼネラルコール アドレスを検出
- バスリピータ モードにより、モジュールをスレーブとして動作させてアドレスとは無関係に全てのメッセージに応答させる事が可能
- 自動SCL x クロックストレッチングがスレーブからの要求に対するプロセッサの遅延を生成する
- 100 kHz と 400 kHz のバス仕様をサポート
- IPMI (Intelligent Platform Management Interface) 規格をサポート

図 19-1 に I²C モジュールのブロック図を示します。

セクション 19. I²C™ (Inter-Integrated Circuit™)

図 19-1: I²C™ のブロック図



19.2 I²C バスの特性

I²C バスは 2 線式シリアル インターフェイスです。図 19-2 に、I²C インターフェイスの代表例として、dsPIC33F/PIC24H デバイスと 24LC256 I²C シリアル EEPROM 間の I²C 接続回路図を示します。

このインターフェイスは、信頼性の高いデータ送受信を保証する包括的なプロトコルを使用します。通信中に、一方のデバイスは「マスタ」として動作してバス上で転送を開始し、その転送を可能にするためにクロック信号を生成します。他方のデバイスは転送に応答する「スレーブ」として動作します。クロックライン SCLx はマスタからスレーブに向けて出力されますが、スレーブが SCLx ラインを駆動する場合もあります。データライン SDAx は、マスタとスレーブの双方からの出力および入力として機能します。

SDAx および SCLx ラインは双方向であるため、これらのラインを駆動するデバイス出力段は、バスのワイヤード AND 機能を実行するためにオープンドレイン出力である必要があります。いずれのデバイスもラインをプルダウンしていない時に HIGH レベルを確保するために、外付けプルアップ抵抗を使用します。

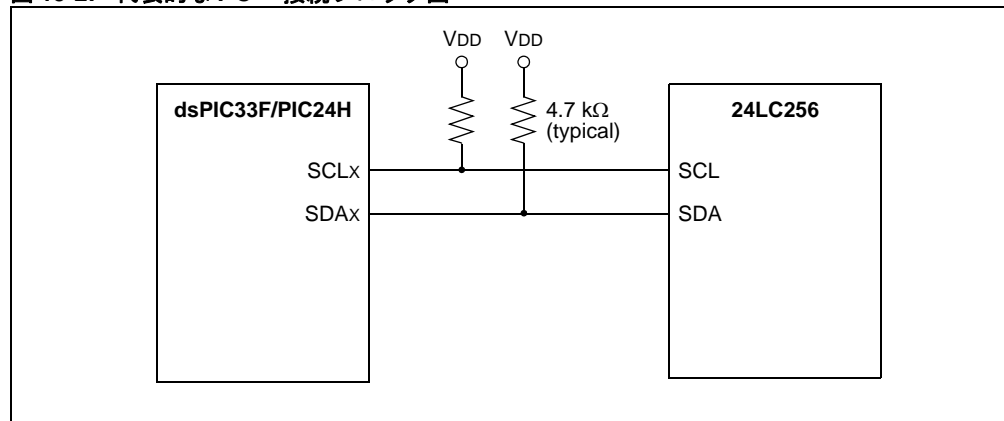
I²C インターフェイス プロトコルでは、各デバイスが 1 つのアドレスを持ちます。マスタは、データ転送を開始する際にまず転送先デバイスのアドレスを送信します。全てのデバイスは、そのアドレスが自分のアドレスに一致するかどうかを「listen」します。このアドレスの bit 0 は、マスタがスレーブデバイスに対して「読み出し」と「書き込み」のどちらを行うのかを指定します。データ転送中、マスタとスレーブは常に相対するモード（トランスミッタ / レシーバ）で動作します。つまり、マスタとスレーブは下記のいずれかの関係で動作します。

- マスタがトランスミッタ、スレーブがレシーバ
- スレーブがトランスミッタ、マスタがレシーバ

いずれの場合も、マスタが SCLx クロック信号を生成します。

Note: I²C ピン (SDA ピンと SCL ピン) は 5 V を許容しません。しかし、これらの I²C ピンに多重化された他のピン機能 (I/O ポート等) は 5 V を許容する場合があります。詳細は各デバイスのデータシートを参照してください。

図 19-2: 代表的な I²C™ 接続ブロック図



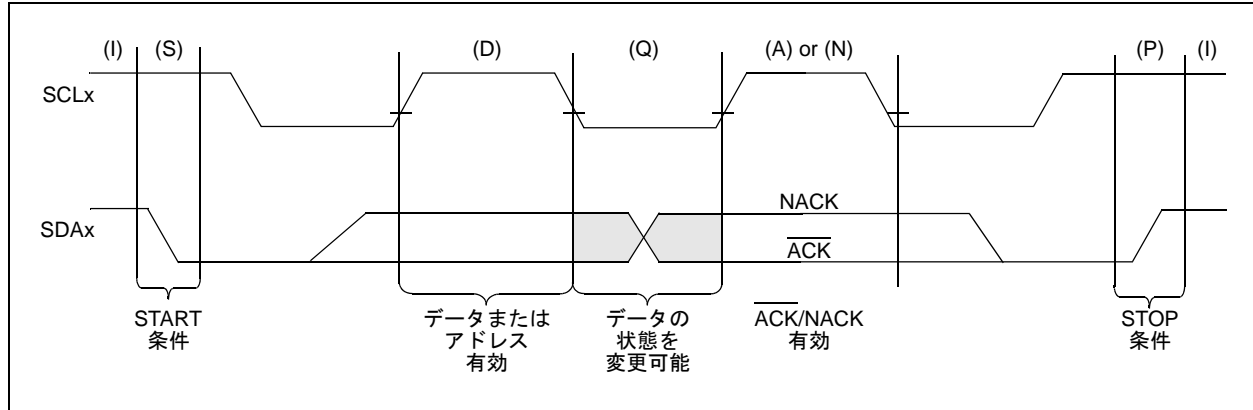
19.2.1 バスプロトコル

下記の I²C バスプロトコルが定義されています。

- データ転送はバスがビジー状態ではない時にのみ開始できる
- データ転送中、SCLx クロックラインが HIGH の時にデータラインの状態は変化してはならない (SCLx クロックラインが HIGH の時にデータラインの状態が変化した場合、START 条件または STOP 条件とみなす)

これに従い、図 19-3 に示すバス条件が定義されています。

図 19-3: I²C™ バスプロトコルに従うバスの状態



19.2.1.1 データ転送の開始 (B)

バスがアイドル状態の時に、クロック (SCLx) が HIGH を維持したまま SDAx ラインが HIGH から LOW へ変化すると、START 条件が発生します。START 条件は全てのデータ転送に先行する必要があります。

19.2.1.2 データ転送の停止 (C)

クロック (SCLx) が HIGH の時に SDAx ラインが LOW から HIGH へ変化すると STOP 条件が発生します。全てのデータ転送は STOP 条件によって終了します。

19.2.1.3 リピートスタート (R)

待機状態の後、クロック (SCLx) が HIGH の時に SDAx ラインが HIGH から LOW へ変化するとリピートスタート条件が発生します。リピートスタートにより、マスタはバスの制御を中止する事なくバスの方向またはアドレス先スレーブデバイスを変更できます。

19.2.1.4 データ有効 (D)

START 条件の後、クロック信号の HIGH 期間中に SDAx ラインが変化しなければ、その時の SDAx ラインの状態が有効データを表します。SCLx クロック 1 周期あたり 1 ビットのデータが転送されます。

19.2.1.5 肯定応答 (A) または否定応答 (N)

全てのデータバイト送信には、レシーバからの肯定応答 ($\overline{\text{ACK}}$) または否定応答 (NACK) が必要です。レシーバは SDAx ラインを LOW にする事によって $\overline{\text{ACK}}$ を返し、SDAx ラインを HIGH にする事によって NACK を返します。肯定応答は SCLx クロックの 1 周期を使用する 1 ビットのデータ転送です。

19.2.1.6 待機 / データ無効 (Q)

データラインの状態は、クロック信号の LOW 期間中に変化する必要があります。デバイスは、SCLx ラインを LOW にする事によってクロック LOW 期間を延長する (バスを待機させる) 事ができます。

19.2.1.7 バスアイドル (I)

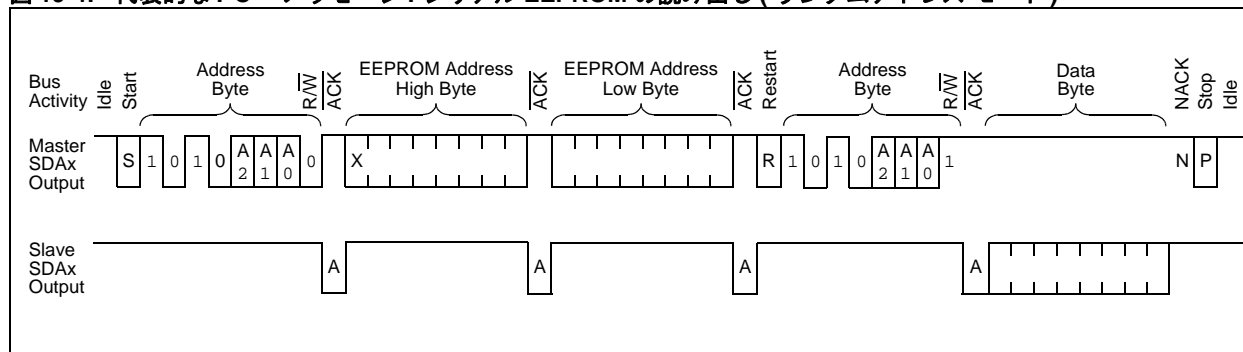
STOP 条件の後と START 条件の前では、データラインとクロックラインの両方が HIGH 状態を維持します。

19.2.2 メッセージ プロトコル

代表的な I²C メッセージを図 19-4 に示します。この例では、メッセージは 24LC256 I²C シリアル EEPROM から指定バイトを読み出します。dsPIC33F/PIC24H デバイスはマスタとして動作し、24LC256 デバイスはスレーブとして動作します。

図 19-4 には、マスタデバイスが駆動するデータとスレーブデバイスが駆動するデータを別々に示しています。ここで、複合 SDAx ラインがマスタとスレーブデータのワイヤード AND である事を思い出してください。マスタデバイスはプロトコルを制御およびシーケンシングします。スレーブデバイスは特別に決められたタイミングでのみバスを駆動します。

図 19-4: 代表的な I²C™ メッセージ: シリアル EEPROM の読み出し (ランダムアドレス モード)



19.2.2.1 メッセージの開始

各メッセージは START 条件で始まり、STOP 条件で終了します。START 条件から STOP 条件までの間に転送するデータのバイト数は、マスタデバイスが決定します。システム プロトコルの定義に従い、メッセージの各バイトは特別な意味 (デバイスアドレス、データバイト等) を持ちます。

19.2.2.2 スレーブのアドレス指定

図 19-4 に示すように、先頭バイトはデバイスアドレス バイトです。I²C メッセージは常にアドレスバイトで始まる必要があります。アドレスバイトはデバイスアドレスと R/W ステータスビットを格納します。アドレスバイト フォーマットの詳細は dsPIC30F ファミリ リファレンス マニュアルの補遺 A:「I²C™ の概要」(DS70074) を参照してください (マイクロチップ社のウェブサイト: www.microchip.com をご覧ください)。この先頭バイトの R/W ステータスビットが「0」の場合、マスタはトランスミッタとして動作し、スレーブはレシーバとして動作する事を示します。

19.2.2.3 スレーブの肯定応答

レシーバとして動作するデバイスは、各バイトを受信した後に肯定応答信号「ACK」を生成する必要があります。マスタデバイスは、この肯定応答ビット用に 1 周期の SCLx クロックを生成する必要があります。

19.2.2.4 マスタの送信

アドレスバイトの後にマスタがスレーブへ送信する 2 つのバイトは、EEPROM データバイトの読み出し位置を格納します。スレーブは、各データバイトに対して肯定応答を返す必要があります。

19.2.2.5 リピートスタート

この時点で、スレーブの EEPROM は、要求されたデータバイトをマスタへ返送するために必要なアドレス情報を既に取得しています。しかし、先頭アドレスバイトの R/W ステータスビットにより、マスタがトランスミッタ、スレーブがレシーバとして指定されています。このため、スレーブがマスタへデータを送信するには、バスの方向を反転する必要があります。

メッセージを終了する事なくバスの方向を変更するために、マスタは「リピートスタート」を送信します。これに続き、最初に送信したのと同じデバイスアドレスと、「1」にセットした R/W ステータスビットを格納したアドレスバイトを送信し、スレーブがトランスミッタ、マスタがスレーブとして動作する事を指定します。

19.2.2.6 スレーブの応答

マスタはバスへクロックを供給し続けますが、SDAx ラインの駆動をリリースします。この間にスレーブが SDAx ラインを駆動してデータバイトを送信します。

19.2.2.7 マスタの肯定応答

マスタからの読み出し動作の場合、マスタはメッセージの最終バイトで否定応答を返す (「NACK」を生成する) 事によって、スレーブに対するデータ要求を終了する必要があります。

19.2.2.8 メッセージの停止

マスタは STOP 条件を送信してメッセージを終了し、バスをアイドル状態に戻します。

19.3 制御 / ステータス レジスタ

I²C モジュールは動作に関連する 7 個のレジスタを備えます。ユーザ アプリケーションはこれらのレジスタにアクセスできます。全てのレジスタには、バイトモードまたはワードモードのどちらでもアクセスできます。これらのレジスタを以下に記載します。

- **I2CxCON: I2Cx 制御レジスタ**

このレジスタ (レジスタ 19-1 参照) はモジュールの動作を制御します。

- **I2CxSTAT: I2Cx ステータス レジスタ**

このレジスタ (レジスタ 19-2 参照) は、動作中のモジュールの状態を示す各種ステータスフラグを格納します。

- **I2CxMSK: I2Cx スレーブモード アドレスマスク レジスタ**

このレジスタ (レジスタ 19-3 参照) は、アドレス検出時に I2CxADD レジスタ内で無視する (マスクする) ビット位置を指定します。これにより、スレーブを複数のアドレスに応答させる事ができます。

- **I2CxRCV: I2Cx 受信バッファレジスタ**

このバッファレジスタからデータバイトを読み出す事ができます。I2CxRCV レジスタは読み出し専用レジスタです。

- **I2CxTRN: I2Cx 送信レジスタ**

これは送信用レジスタです。送信動作中に送信バイトがこのレジスタに書き込まれます。I2CxTRN レジスタは読み書き可能レジスタです。

- **I2CxADD: I2Cx アドレスレジスタ**

このレジスタは、スレーブデバイスのアドレスを保持します。

- **I2CxBRG: I2Cx baud レート ジェネレータ リロードレジスタ**

このレジスタは、I²C モジュールの baud レート ジェネレータ (BRG) 向けのリロード値を保持します。

I2CxTRN レジスタには送信データが書き込まれます。このレジスタは、モジュールがマスタとしてスレーブヘデータを送信する時、またはモジュールがスレーブとして応答データをマスタへ送信する際に使用されます。メッセージの送信中に、I2CxTRN レジスタは各ビットを順次シフトアウトします。このため、バスがアイドル状態ではない時に I2CxTRN レジスタに書き込む事はできません。I2CxTRN レジスタは、既存データの送信中にリロードできます。

マスタまたはスレーブが受信したデータは、シフトレジスタ I2CxRSR にシフトインされます。ユーザ アプリケーションはこのシフトレジスタにアクセスできません。1 バイトを完全に受信すると、そのバイトは I2CxRCV レジスタへ転送されます。受信動作では、I2CxRSR レジスタと I2CxRCV レジスタによって二重バッファリング レシーバが構成されます。これにより、受信済みデータバイトの読み出しを開始する前に次のバイトを受信できます。

ソフトウェアが受信済みバイトを I2CxRCV レジスタから読み出す前に、モジュールが次のバイトを完全に受信した場合、レシーバ オーバーフローが発生して I2COV ビット (I2CxSTAT<6>) がセットされます。I2CxRSR レジスタ内のバイトは失われます。モジュールがバス上で START/リピートスタート、START/STOP 条件を検出するまで、以降の受信とクロック ストレッチングは無効化されます。I2COV フラグがクリアされていれば正常に受信できます。I2COV フラグがクリアされていない場合、モジュールは次のバイトを正しく受信しても NACK を返します。その後 START/リピートスタート、START/STOP 条件を検出するまで、モジュールは受信する事もクロックをストレッチングする事もできません。

I2CxADD レジスタはスレーブデバイスのアドレスを保持します。10 ビット アドレッシングモードでは、レジスタ内の全てのビットを使用します。7 ビット アドレッシングモードでは、I2CxADD<6:0> ビットだけを使用します。I2CxADD<6:0> ビットはアドレスバイトの上位 7 ビットに対応し、R/W ビットはこのレジスタの値に含まれない事に注意が必要です。A10M ビット (I2CxCON<10>) は、スレーブアドレスのモード (10 ビットまたは 7 ビット) を指定します。いずれのスレーブ アドレッシングモードでも、I2CxMSK レジスタを I2CxADD レジスタに組み合わせる事により、完全なアドレス一致から 1 つまたは複数のビット位置を除外できます。これにより、スレーブモードで動作中のモジュールを複数のアドレスに対して応答させる事ができます。

dsPIC33F/PIC24H ファミリ リファレンス マニュアル

レジスタ 19-1: I2CxCON: I2Cx 制御レジスタ

R/W-0	U-0	R/W-0	R/W-1, HC	R/W-0	R/W-0	R/W-0	R/W-0
I2CEN	—	I2CSIDL	SCLREL	IPMIEN ⁽¹⁾	A10M	DISSLW	SMEN
bit 15				bit 8			

R/W-0	R/W-0	R/W-0	R/W-0, HC	R/W-0, HC	R/W-0, HC	R/W-0, HC	R/W-0, HC
GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
bit 7				bit 0			

凡例:

U = 未実装ビット、「0」として読み出し

R = 読み出し可能ビット W = 書き込み可能ビット HS = ハードウェアでセット HC = ハードウェアでクリア

-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 **I2CEN:** I2Cx イネーブルビット
 1 = I2Cx モジュールを有効にし、SDAx および SCLx ピンをシリアルポート ピンとして設定する
 0 = I2Cx モジュールを無効にする (ポート機能が全ての I²C ピンを制御する)
- bit 14 **未実装:** 「0」として読み出し
- bit 13 **I2CSIDL:** 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 = デバイスがアイドルモード中でもモジュールの動作を継続する
- bit 12 **SCLREL:** SCLx リリース制御ビット (I²C スレーブとして動作している場合)
 1 = SCLx クロックをリリースする
 0 = SCLx クロックを LOW に保持する (クロック ストレッチング)
STREN = 1 の場合:
 このビットは読み書き可能です (ソフトウェアは「0」を書き込んでクロック ストレッチングを開始し、「1」を書き込んでクロックをリリースできます)。ハードウェアはスレーブ送信の開始時とスレーブ受信の終了時にこのビットをクリアします。
STREN = 0 の場合:
 このビットは読み出しとリセットのみ可能です (ソフトウェアは「1」を書き込んでクロックをリリースする事しかできません)。ハードウェアはスレーブ送信の開始時にこのビットをクリアします。
- bit 11 **IPMIEN:** IPMI (Intelligent Platform Management Interface) イネーブルビット ⁽¹⁾
 1 = IPMI サポートモードを有効にする (全てのアドレスに応答する)
 0 = IPMI サポートモードを無効にする
- bit 10 **A10M:** 10 ビット スレーブアドレス ビット
 1 = I2CxADD レジスタは 10 ビット スレーブアドレスを格納する
 0 = I2CxADD レジスタは 7 ビット スレーブアドレスを格納する
- bit 9 **DISSLW:** スルーレート制御ディセーブル ビット
 1 = スルーレート制御を無効にする
 0 = スルーレート制御を有効にする
- bit 8 **SMEN:** SMBus 入力レベルビット
 1 = SMBus 仕様に従う I/O ピンしきい値を有効にする
 0 = SMBus 用入力しきい値を無効にする
- bit 7 **GCEN:** ゼネラルコール イネーブルビット (I²C スレーブとして動作する場合)
 1 = I2CxRSR レジスタにゼネラルコールアドレスを受信した時に割り込みを有効にする (モジュールを受信用に有効にする)
 0 = ゼネラルコールアドレスを無効にする
- bit 6 **STREN:** SCLx クロック ストレッチング イネーブルビット (I²C スレーブモードのみ、SCLREL ビットとの組み合わせで使用)
 1 = ソフトウェアまたは受信クロック ストレッチングを有効にする
 0 = ソフトウェアまたは受信クロック ストレッチングを無効にする

Note 1: マスタとして動作中に IPMIEN ビットをセットしない必要があります。

レジスタ 19-1: I2CxCON: I2Cx 制御レジスタ (続き)

- bit 5 **ACKDT:** 肯定応答データビット (I²C マスタモード、受信動作のみ)
ソフトウェアが肯定応答シーケンスを開始した時に送信する値
1 = 肯定応答時に NACK を送信する
0 = 肯定応答時に ACK を送信する
- bit 4 **ACKEN:** 肯定応答シーケンス イネーブルビット (I²C マスタモード受信動作)
1 = SDAxおよびSCLxピンで肯定応答シーケンスを開始し、ACKDTデータビットを送信する(マスタ肯定応答シーケンスの終了時にハードウェアがこのビットクリアします)
0 = 肯定応答シーケンスを実行していない
- bit 3 **RCEN:** 受信イネーブルビット (I²C マスタモード)
1 = I²Cの受信モードを有効にする(マスタ受信データバイトの第8ビット受信終了時にハードウェアがこのビットをクリアします)
0 = 受信シーケンスを実行していない
- bit 2 **PEN:** STOP 条件イネーブルビット (I²C マスタモード)
1 = SDAxおよびSCLxピンでSTOP条件を開始する(マスタSTOPシーケンス終了時にハードウェアがこのビットをクリアします)
0 = STOP条件を実行していない
- bit 1 **RSEN:** リピートスタート条件イネーブルビット (I²C マスタモード)
1 = SDAxおよびSCLxピンでリピートスタート条件を開始する(マスタリピートスタートシーケンス終了時にハードウェアがこのビットをクリアします)
0 = リピートスタート条件を実行していない
- bit 0 **SEN:** START 条件イネーブルビット (I²C マスタモード)
1 = SDAxおよびSCLxピンでSTART条件を開始する(マスタSTARTシーケンス終了時にハードウェアがこのビットをクリアします)
0 = START条件を実行していない

Note 1: マスタとして動作中に IPMIEN ビットをセットしない必要があります。

dsPIC33F/PIC24H ファミリ リファレンス マニュアル

レジスタ 19-2: I2CxSTAT: I2Cx ステータス レジスタ

R-0, HSC	R-0, HSC	U-0	U-0	U-0	R/C-0, HS	R-0, HSC	R-0, HSC
ACKSTAT	TRSTAT	—	—	—	BCL	GCSTAT	ADD10
bit 15						bit 8	

R/C-0, HS	R/C-0, HS	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC
IWCOL	I2COV	D/A	P	S	R/W	RBF	TBF
bit 7							bit 0

凡例:				U = 未実装ビット、「0」として読み出し			
R = 読み出し可能ビット		C = クリア可能ビット		HS = ハードウェアでセット		HSC = ハードウェアでセット / クリア	
-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	ACKSTAT: 肯定応答ステータスビット 1 = スレーブから NACK を受信した 0 = スレーブから ACK を受信した スレーブまたはマスタ肯定応答終了時にハードウェアがこのビットをセットまたはクリアします。
bit 14	TRSTAT: 送信ステータスビット (I ² C マスタモード送信動作) 1 = マスタ送信を実行中 (8 ビット + ACK) 0 = マスタ送信を実行中ではない ハードウェアは、マスタ送信開始時にこのビットをセットし、スレーブ肯定応答終了時にこのビットをクリアします。
bit 13-11	未実装: 「0」として読み出し
bit 10	BCL: マスタ バス衝突検出ビット 1 = マスタ動作中にバス衝突を検出した 0 = 衝突を検出していない バス衝突検出時にハードウェアがこのビットをセットします。
bit 9	GCSTAT: ゼネラルコール ステータスビット 1 = ゼネラルコールアドレスを受信した 0 = ゼネラルコールアドレスを受信しなかった ハードウェアはアドレスがゼネラルコールアドレスに一致した時にこのビットをセットし、STOP 検出時にこのビットをクリアします。
bit 8	ADD10: 10 ビットアドレス ステータスビット 1 = 10 ビットアドレスが一致した 0 = 10 ビットアドレスは一致しなかった ハードウェアは一致した10ビットアドレスの第2バイトが一致した時にこのビットをセットし、STOP 条件でこのビットをクリアします。
bit 7	IWCOL: 書き込み衝突検出ビット 1 = I ² C モジュールがビジーであるため、I2CxTRN レジスタの書き込みに失敗した 0 = 書き込み衝突は発生していない ビジー中に I2CxTRN レジスタへの書き込みが発生すると、ハードウェアがこのビットをセットします (クリアはソフトウェアによる)。
bit 6	I2COV: 受信オーバーフロー フラグビット 1 = I2CxRCV レジスタが先に受信したバイトをまだ保持している時に次のバイトを受信した 0 = オーバーフローは発生していない I2CxRSR レジスタから I2CxRCV レジスタへの転送が発生した時に、ハードウェアがこのビットをセットします (クリアはソフトウェアによる)。
bit 5	D/A: データ / アドレスビット (I ² C スレーブモード) 1 = 直前に受信したバイトはデータであった事を示します 0 = 直前に受信したバイトはデバイスアドレスであった事を示します ハードウェアはデバイスアドレス一致時にこのビットをクリアし、スレーブバイト受信時にこのバイトをセットします。あるいは、このビットは送信が完了した後に TBF フラグがクリアされた時にセットされます。
bit 4	P: STOP ビット 1 = STOP ビットを検出した 0 = STOP ビットを検出しなかった START、リピートスタート、STOP を検出した時に、ハードウェアがこのビットをセットまたはクリアします。

セクション 19. I²C™ (Inter-Integrated Circuit™)

レジスタ 19-2: I2CxSTAT: I2Cx ステータス レジスタ (続き)

bit 3	S: START ビット 1 = START (またはリピートスタート) ビットを検出した 0 = START ビットを検出しなかった START、リピートスタート、STOP を検出した時に、ハードウェアがこのビットをセットまたはクリアします。
bit 2	R/W: 読み / 書き情報ビット (I ² C スレーブとして動作している場合) 1 = 読み出し : データをスレーブから転送する 0 = 書き込み : データをスレーブへ転送する I ² C デバイスアドレス バイトを受信した後に、ハードウェアがこのビットをセットまたはクリアします。
bit 1	RBF: 受信バッファフル ステータスビット 1 = 受信を完了した (I2CxRCV レジスタはフル) 0 = 受信を完了していない (I2CxRCV レジスタはエンプティ) ハードウェアは I2CxRCV レジスタに受信バイトが書き込まれた時にこのビットをセットし、ソフトウェアが I2CxRCV レジスタを読み出した時にこのビットをクリアします。
bit 0	TBF: 送信バッファフル ステータスビット 1 = 送信を完了していない (I2CxTRN レジスタはフル) 0 = 送信を完了した (I2CxTRN レジスタはエンプティ) ハードウェアはソフトウェアが I2CxTRN レジスタに書き込んだ時にこのビットをセットし、データ送信完了時にこのビットをクリアします。

dsPIC33F/PIC24H ファミリ リファレンス マニュアル

レジスタ 19-3: I2CxMSK: I2Cx スレーブモード アドレスマスク レジスタ

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	AMSK9	AMSK8
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
AMSK7	AMSK6	AMSK5	AMSK4	AMSK3	AMSK2	AMSK1	AMSK0
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 15-10 **未実装:** 「0」として読み出し

bit 9-0 **AMSKx:** アドレスビット x のマスクング選択ビット

10 ビットアドレスの場合:

1 = 受信するメッセージアドレスの bit Ax のマスクングを有効にする (このビット位置のビット一致を要求しない)

0 = bit Ax のマスクングを無効にする (このビット位置のビット一致を要求する)

7 ビットアドレスの場合 (I2CxMSK<6:0> のみ使用):

1 = 受信するメッセージアドレスの bit Ax + 1 のマスクングを有効にする (このビット位置のビット一致を要求しない)

0 = bit Ax + 1 のマスクングを無効にする (このビット位置のビット一致を要求する)

19.4 I²C 動作の有効化

I²C モジュールは、I2CEN (I2CxCON<15>) ビットをセットする事によって有効化されます。

I²C モジュールは全てのマスタ機能とスレーブ機能を実装しています。モジュールが有効な時、マスタ機能とスレーブ機能は同時に動作し、ソフトウェアまたはバスのイベントに従って応答します。

有効化した直後の初期状態では、モジュールは SDAx ピンと SCLx ピンをリリースし、バスをアイドル状態にします。ソフトウェアが SEN 制御ビットをセットし、かつデータを I2CxTRN レジスタに書き込むまで、マスタ機能はアイドル状態を維持します。上記の 2 つの動作が発生するとマスタイベントが開始されます。

マスタロジックがアクティブになった時、スレーブロジックもアクティブ状態を維持してバスの監視を開始します。スレーブロジックは、バス上で START イベントと有効なアドレスを検出すると、スレーブ トランザクションを開始します。

19.4.1 I²C I/O の有効化

バス動作には 2 本のピンを使用します。1 本はクロック用の SCLx ピン、もう 1 本はデータ用の SDAx ピンです。モジュールを有効にした時、そのモジュールよりも高い優先度を持つ他のモジュールがバスを制御していなければ、そのモジュールが SDAx ピンと SCLx ピンの制御を引き継ぎます。この時、モジュールがこれらのピンの状態と方向を変更するため、モジュールソフトウェアはこれらのピンのポート I/O 状態に関与する必要はありません。これらのピンは、初期化時に 3 ステートにされます (リリースされます)。

19.4.2 I²C 割り込み

I²C モジュールは 2 種類の割り込みを生成します。1 つはマスタイベントに割り当てられた割り込み (MI2CxIF)、もう 1 つはスレーブイベントに割り当てられた割り込み (SI2CxIF) です。これらの割り込みは対応する割り込みフラグビットをセットします。対応する割り込みイネーブルビットが事前にセットされ、なおかつ割り込み優先度が十分に高ければ、これらの割り込みはソフトウェア処理に割り込みます。

MI2CxIF 割り込みは、下記のマスタメッセージ イベントが完了した時に生成されます。

- START 条件
- STOP 条件
- データ転送バイトの送受信
- 肯定応答の送信
- リピートスタート
- バス衝突 イベントの検出

SI2CxIF 割り込みは、そのスレーブに向けられたメッセージ (下記を含む) を検出した時に生成されます。

- 有効なデバイスアドレス (ゼネラルコールを含む)
- データ送信の要求 (ACK) または拒否 (NACK)
- 受信データ

19.4.3 バスマスタとして動作時の baud レートの設定

I²C マスタとして動作するモジュールはシステム SCLx クロックを生成する必要があります。一般的に、I²C システムクロックは 100 kHz、400 kHz、1 MHz のいずれかに指定されます。システムクロック レートは、SCLx の最小 LOW 時間と最小 HIGH 時間の合計として指定されます。ほとんどの場合、これは 2 TBRG インターバルによって定義されます。

baud レート ジェネレータ (BRG) には、図 19-5 に示すように I2CxBRG レジスタ値がリロードされます。baud レート ジェネレータにこの値がリロードされると、ジェネレータは 0 までカウントダウンし、次にリロードされるまで停止します。このジェネレータ カウントは、1 命令サイクル (Tcy) あたり 2 回デクリメントされます。baud レート ジェネレータは、baud レート再開時に自動的にリロードされます。例えばクロック同期中である場合、SCLx ピンで HIGH が検出された時に baud レート ジェネレータがリロードされます。

Note: 2 未満の I2CxBRG レジスタ値はサポートされません。

baud レート ジェネレータのリロード値は下式により計算できます。

式 19-1: BRG リロード値の計算

$$I2C_{BRG} = \left[\left(\frac{1}{F_{SCL}} - PGD \right) \cdot F_{CY} \right] - 2$$

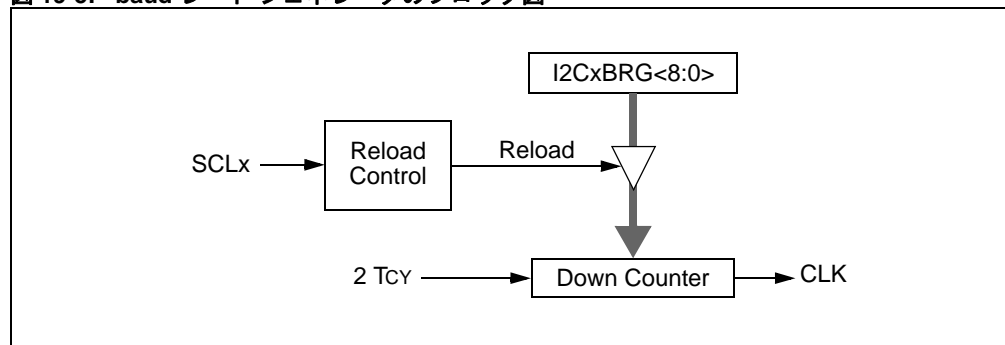
表 19-1: I²C™ のクロックレート

要求 システム F _{SCL}	F _{CY}	PGD ⁽¹⁾	I2C _{BRG} 10 進数	I2C _{BRG} 16 進数
100 kHz	40 MHz	130 ns	392.8	0x188
100 kHz	20 MHz	130 ns	195.4	0x0C3
100 kHz	10 MHz	130 ns	96.7	0x060
400 kHz	20 MHz	130 ns	45.4	0x02D
400 kHz	10 MHz	130 ns	21.7	0x015
400 kHz	5 MHz	130 ns	9.85	0x009
1 MHz	10 MHz	130 ns	6.7	0x006

Note 1: パルスゴブラー遅延 (PGD) は 130 ns (typical) です。詳細は各デバイスのデータシートを参照してください。

Note: 式 19-1 と表 19-1 はあくまでもガイドラインです。これらはシステム依存パラメータであるため、実際の baud レートは若干異なる場合があります。実際の baud レートがシステム要件を満たす事を確認するにはテストが必要です。システム要件を満たさない場合、I2CxBRG の調整が必要かもしれません。

図 19-5: baud レート ジェネレータのブロック図



19.5 シングルマスタ環境でマスタとして通信する

I²C モジュールを使用する一般的なシステムは、I²C シリアルメモリ等の I²C 周辺モジュールとの通信に I²C を使用します。I²C システムでは、マスタがバス上の全てのデータ通信シーケンスを制御します。ここで取り上げる例では、dsPIC33F/PIC24H とその I²C モジュールがシステム内で唯一のマスタとして動作します。このシングルマスタは SCLx クロックを生成し、メッセージ プロトコルを制御します。

Note: マスタとして動作している時に IPMIEN (I2CxCON<11>) ビットをセットしない必要があります。

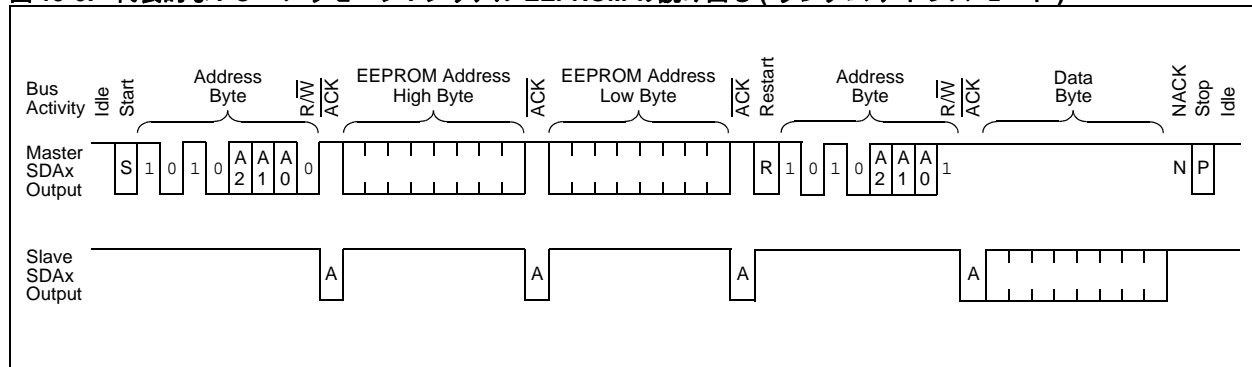
I²C モジュールは I²C メッセージ プロトコルの各部分を制御しますが、プロトコルの各部分をシーケンシングして完全なメッセージを構成するのはソフトウェアの役割です。

シングルマスタ環境での代表的動作として、I²C シリアル EEPROM から 1 バイトを読み出すメッセージ例を図 19-6 に示します。

このメッセージを完了するために、ソフトウェアは下記の手順でプロトコルをシーケンシングします。

1. SDAx と SCLx で START 条件を生成する
2. 書き込み指示を含む I²C デバイスアドレス バイトをスレーブへ送信する
3. スレーブからの肯定応答を待機 / 確認する
4. シリアルメモリ アドレスの上位バイトをスレーブへ送信する
5. スレーブからの肯定応答を待機 / 確認する
6. シリアルメモリ アドレスの下位バイトをスレーブへ送信する
7. スレーブからの肯定応答を待機 / 確認する
8. SDAx と SCLx でリピートスタート条件を生成する
9. 読み込み指示を含むデバイスアドレス バイトをスレーブへ送信する
10. スレーブからの肯定応答を待機 / 確認する
11. マスタ受信を有効にしてシリアルメモリ データを受信する
12. データバイトの受信終了時に ACK または NACK 条件を生成する
13. SDAx と SCLx で STOP 条件を生成する

図 19-6: 代表的な I²C™ メッセージ: シリアル EEPROM の読み出し (ランダム アドレス モード)



I²C モジュールは、START および STOP ジェネレータ、データバイト送信、データバイト受信、肯定応答ジェネレータ、baud レート ジェネレータを含むマスタモード通信をサポートします。一般的に、ソフトウェアは制御レジスタに書き込む事によって特定のステップを開始し、割り込みを待機するか状態をポーリングする事によってステップの完了を待機します。

以下では、これらの各動作について詳細に説明します。

Note: I²C モジュールはイベントのキューを許容しません。例えば、ソフトウェアは START 条件を開始した後に START 条件の完了を待たず即座に I2CxTRN レジスタに書き込む事はできません。この場合、I2CxTRN レジスタには書き込まれず、この書き込みが発生しなかった事を示すために IWCOL ステータスビットがセットされます。

19.5.1 START バスイベントの生成

START イベントを開始するために、ソフトウェアは START イネーブルビット (SEN (I2CxCON<0>)) をセットします。START ビットをセットする前に、ソフトウェアは P ステータスビット (I2CxSTAT<4>) をチェックする事によってバスがアイドル状態であるかどうかを確認できます。

図 19-7 に START 条件のタイミングを示します。

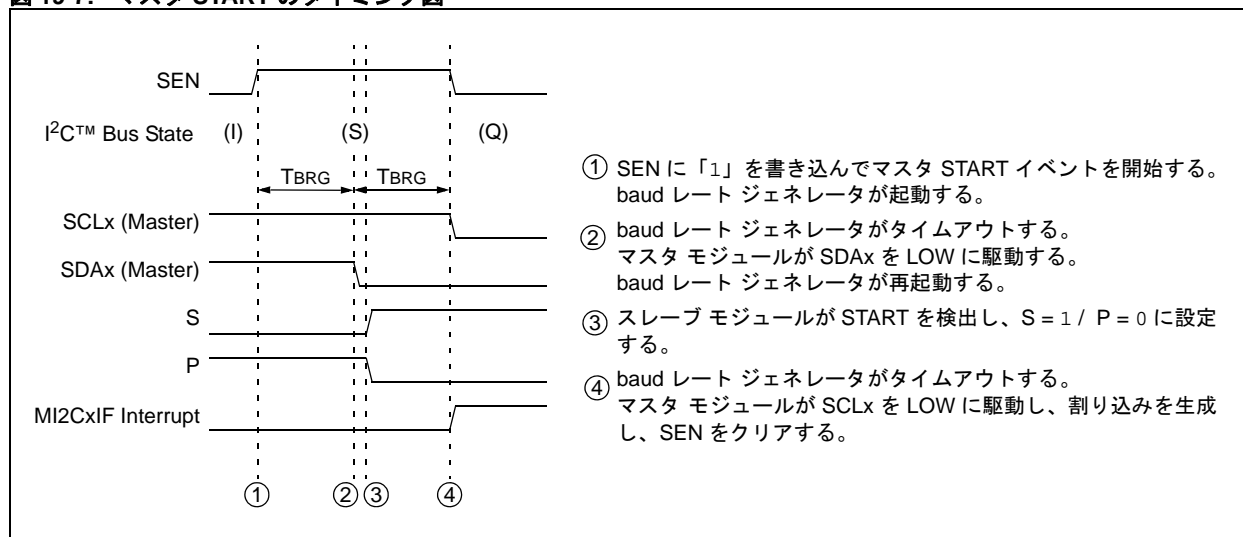
- スレーブロジックは START 条件を検出した後に S ステータスビット (I2CxSTAT<3>) をセットし、P ステータスビット (I2CxSTAT<4>) をクリアします。
- START 条件が完了した時に SEN ビットが自動的にクリアされます。
- STAR 条件が完了した時に MI2CxIF 割り込みが生成されます。
- START 条件終了後、SDAx ラインと SCLx ラインは LOW を維持します (Q 状態)。

19.5.1.1 IWCOL ステータスフラグ

START シーケンスの途中でソフトウェアが I2CxTRN レジスタに書き込むと、IWCOL ステータスビットがセットされます。書き込みは発生せず、送信バッファの内容は変更されません。

Note: イベントのキューは許容されないため、I2CxCON レジスタの下位 5 ビットへの書き込みは START 条件が完了するまで無効化されます。

図 19-7: マスタ START のタイミング図



19.5.2 スレーブデバイスへのデータ送信

データバイト、7 ビット デバイスアドレス バイト、10 ビットアドレスの第 2 バイトは、適切な値を I2CxTRN レジスタに書き込む事によって単純に送信できます。このレジスタに書き込むと、下記のプロセスが開始されます。

- ソフトウェアが送信データバイトを I2CxTRN レジスタに書き込む
 - I2CxTRN レジスタへの書き込みによってバッファフルフラグビット (TBF (I2CxSTAT<0>)) がセットされる
 - SDAx ピンでデータバイトをシフトアウトし、8 ビットの全てを送信する (SCLx の各立ち上がりエッジの後でアドレス / データの各ビットを SDAx ピンにシフトアウトする)
 - 9 番目の SCLx クロックで、モジュールがスレーブデバイスからの ACK ビットをシフトインし、その値を ACKSTAT ステータスビット (I2CxSTAT<15>) に書き込む
 - 9 番目の SCLx クロックサイクルの終了時に、モジュールが MI2CxIF 割り込みを生成する
- モジュール自体はデータバイトの生成も検証も行いません。各バイトの内容と使用方法は、ソフトウェアが管理するメッセージ プロトコルの状態によって決まります。

19.5.2.1 7 ビットアドレスをスレーブへ送信する

7 ビット デバイスアドレスの送信では、1 バイトをスレーブへ送信します。7 ビット アドレス バイトは、7 ビットの I²C デバイスアドレスと R/W ステータスビットを格納する必要があります。R/W ステータスビットは、メッセージをスレーブへ書き込む (マスタ送信 / スレーブ受信) のか、それともスレーブから読み出す (スレーブ送信 / マスタ受信) のかを指定します。

Note: 7 ビット アドレッシング モードでは、I²C プロトコルを使用する各ノードに一意のアドレス (I2CxADD レジスタに保存) を割り当てる必要があります。

19.5.2.2 10 ビットアドレスをスレーブへ送信する

10 ビット デバイスアドレスの送信では、2 バイトをスレーブへ送信します。第 1 バイトは、10 ビット アドレッシング モード用に予約された 5 ビットの I²C デバイスアドレスと、10 ビット アドレスの上位 2 ビットを格納します。スレーブは 10 ビットアドレスの残りの 8 ビットを格納した第 2 バイトを受信する必要があるため、第 1 バイトの R/W ステータスビットは「0」(マスタ送信 / スレーブ受信) である必要があります。引き続きメッセージデータをスレーブへ向けて送信する場合、マスタはそのままデータ送信を続ける事ができます。しかし、マスタがスレーブからの応答を要求する場合、R/W ステータスビットを「1」にセットしたりピートスタートシーケンスを使用してメッセージの R/W ステータスをスレーブからの読み出し用に変更する必要があります。

Note: 10 ビット アドレッシング モードでは、I²C プロトコルを使用する各ノードに一意のアドレス (I2CxADD レジスタに保存) を割り当てる必要があります。

19.5.2.3 スレーブから肯定応答を受信する

SCLx クロックの 8 番目の立ち下がりエッジで、TBF ステータスビットがクリアされ、マスタが SDAx ピンをリリースします。これにより、スレーブからの肯定応答の送信が可能になります。その後マスタが 9 番目の SCLx クロックを生成します。

アドレスが一致した場合、またはデータを正しく受信した場合、スレーブデバイスは 9 番目のビット時間中に ACK ビットを返す事ができます。スレーブは、デバイスアドレスの一致を認識した時 (ゼネラルコールを含む)、または正しくデータを受信した時に、肯定応答を送信します。

ACK のステータスは、SCLx クロックの 9 番目の立ち下がりエッジで、肯定応答ステータスビット (ACKSTAT (I2CxSTAT<15>)) に書き込まれます。その後モジュールは MI2CxIF 割り込みを生成し、次のデータバイトが I2CxTRN レジスタに書き込まれるまでアイドル状態に移行します。

19.5.2.4 ACKSTAT ステータスフラグ

ACKSTAT ステータスビット (I2CxSTAT<15>) は、スレーブが肯定応答 (ACK = 0) を送信した時にクリアされ、肯定応答しない (ACK = 1) 時にセットされます。

19.5.2.5 TBF ステータスフラグ

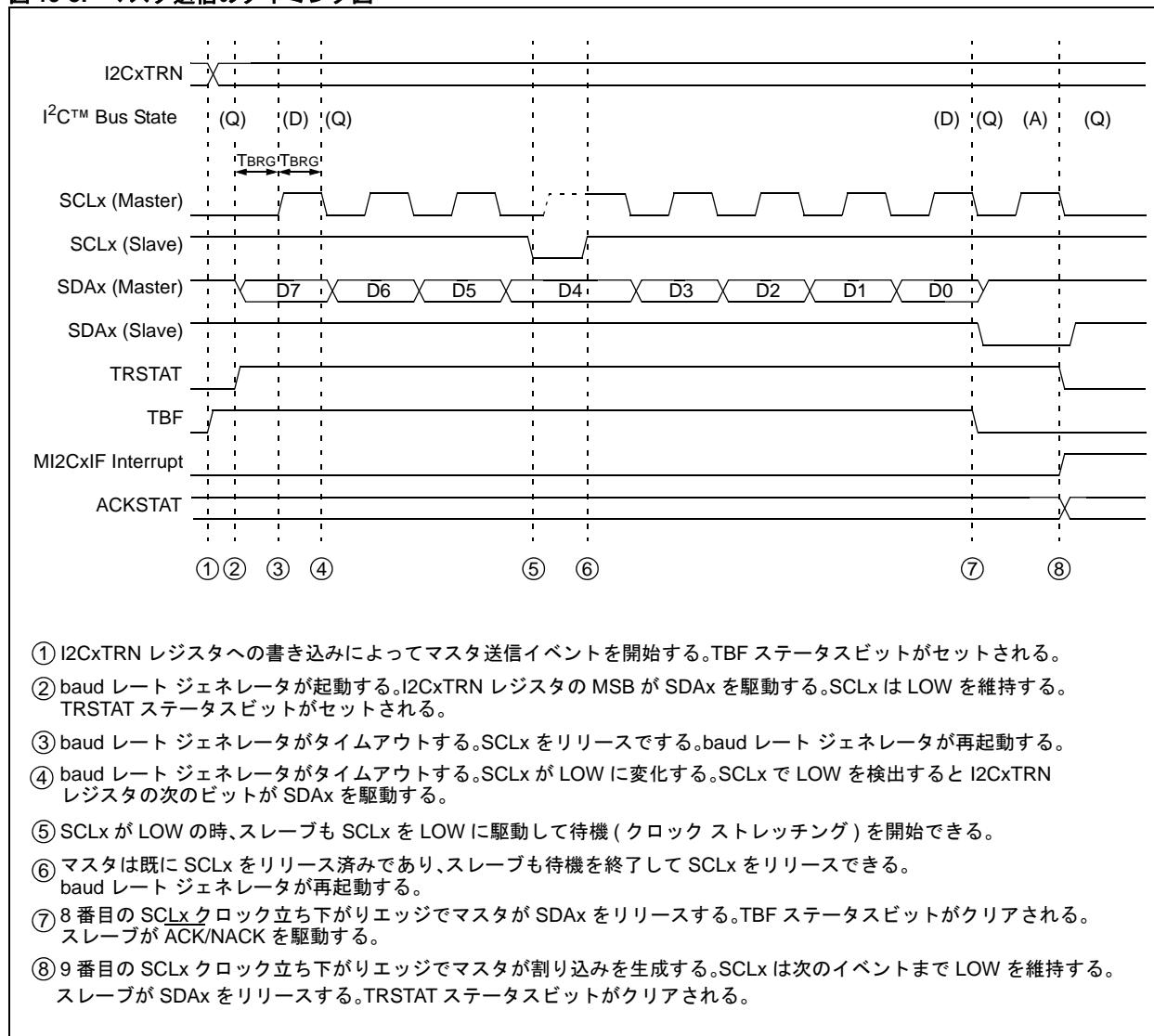
送信中、TBF ステータスビット (I2CxSTAT<0>) は、CPU が I2CxTRN レジスタに書き込むとセットされ、8 ビットの全てがシフトアウトされるとクリアされます。

19.5.2.6 IWCOL ステータスフラグ

送信を実行中である時 (モジュールがデータバイトをまだシフトアウト中である時) にソフトウェアが I2CxTRN レジスタへの書き込みを試みた場合、IWCOL ステータスビットがセットされ、バッファの内容は変更されません (書き込みは発生しません)。IWCOL ステータスビットはソフトウェアでクリアする必要があります。

Note: イベントのキューは許容されないため、I2CxCON レジスタの下位 5 ビットへの書き込みは送信状態が完了するまで無効化されます。

図 19-8: マスタ送信のタイミング図



19.5.3 スレーブデバイスからのデータ受信

マスタは、 $\overline{R/W}$ ステータスビットを「1」にセットしてスレーブアドレスを送信した後に、スレーブデバイスからデータを受信できるようになります。これは受信イネーブルビット (RCEN (I2CxCON<3>)) をセットする事により有効化されます。マスタロジックはクロックの生成を開始し、SCLx クロックの各立ち下がりエッジの前で SDAx ラインをサンプリングし、データを I2CxRSR レジスタにシフトインします。

Note: RCEN ビットをセットする前に、I2CxCON レジスタの下位 5 ビットが「0」である必要があります。これはマスタロジックが非アクティブである事を保証します。

8 番目の SCLx クロック立ち下がりエッジの後に下記のイベントが発生します。

- RCEN ビットが自動的にクリアされる
- I2CxRSR レジスタの内容が I2CxRCV レジスタに転送される
- RBF ステータスビットがセットされる
- モジュールが MI2CxIF 割り込みを生成する

RBF ステータスビットは、CPU がバッファを読み出した時に自動的にクリアされます。ソフトウェアはデータを処理した後に肯定応答シーケンスを実行できます。

19.5.3.1 RBF ステータスフラグ

データ受信中、デバイスアドレスまたはデータバイトが I2CxRSR レジスタから I2CxRCV レジスタへ転送された時に RBF ステータスビットがセットされます。このビットはソフトウェアが I2CxRCV レジスタを読み出した時にクリアされます。

19.5.3.2 I2COV ステータスフラグ

RBF ステータスビットがセットされた状態で、かつ以前に受信したバイトが I2CxRCV レジスタに残ったままである時に次のバイトを I2CxRSR レジスタで受信すると、I2 COV ステータスビットがセットされ、I2CxRSR レジスタ内のデータは失われます。

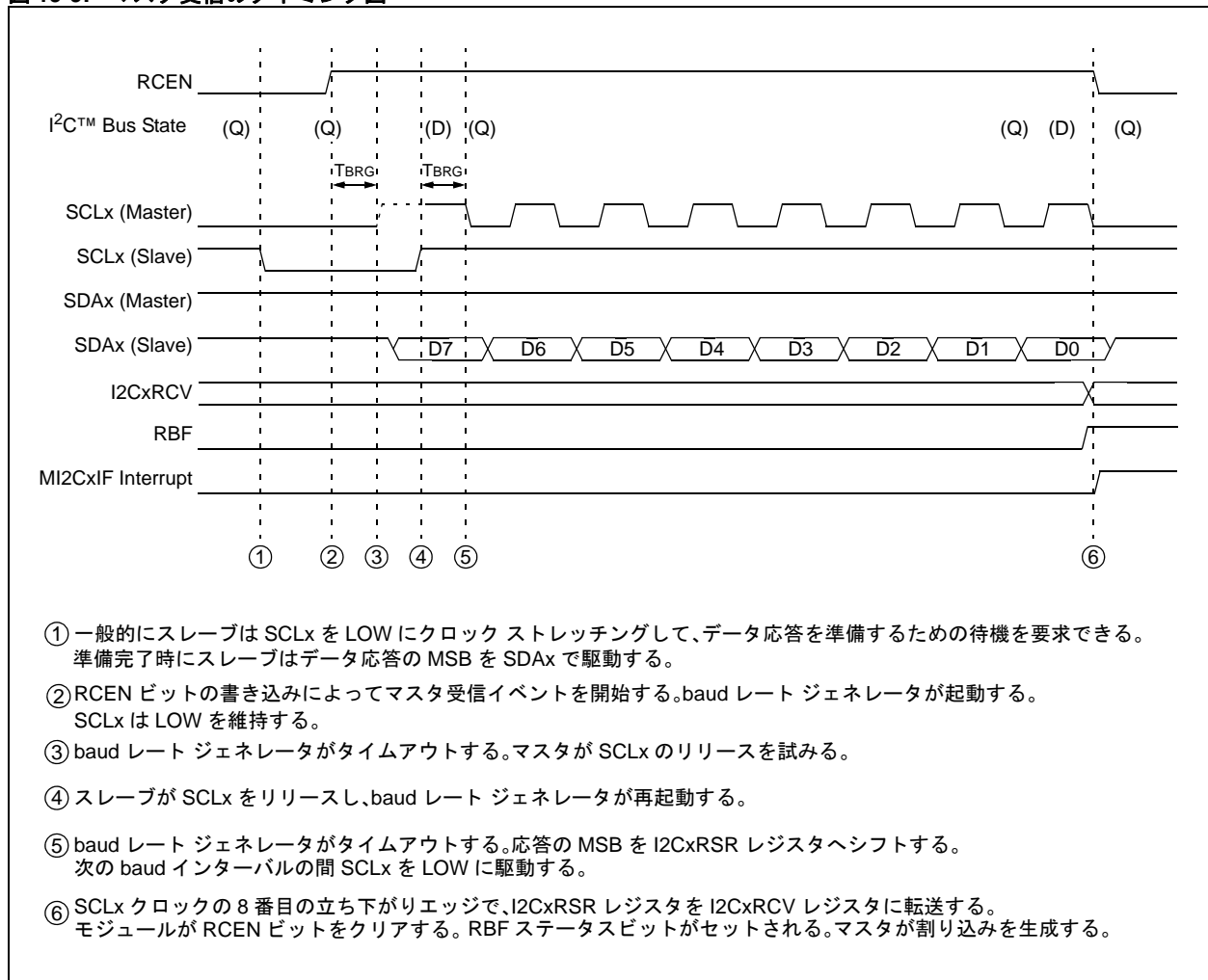
I2COV ステータスビットがセットされたままであっても、後続の受信は禁止されません。I2CxRCV レジスタを読み出して RBF ステータスビットがクリアされた後に、I2CxRSR レジスタで次のバイトを受信すると、そのバイトは I2CxRCV レジスタへ転送されます。

19.5.3.3 IWCOL ステータスフラグ

受信が実行中である時 (I2CxRSR レジスタがデータバイトをまだシフトイン中である時) にソフトウェアが I2CXTRN レジスタに書き込みを試みた場合、IWCOL ステータスビットがセットされ、バッファの内容は変更されません (書き込みは発生しません)。

Note: イベントのキューは許容されないため、I2CxCON レジスタの下位 5 ビットへの書き込みは受信状態が完了するまで無効化されます。

図 19-9: マスタ受信のタイミング図



19.5.4 肯定応答の生成

肯定応答イネーブルビット (ACKEN (I2CxCON<4>)) をセットすると、マスタ肯定応答シーケンスの生成が有効化されます。

Note: ACKEN ビットをセットする前に、I2CxCON レジスタの下位 5 ビットが「0」(マスタロジックが無効) である必要があります。

図 19-10 に、 $\overline{\text{ACK}}$ シーケンス、図 19-11 に NACK シーケンスを示します。肯定応答データビット (ACKDT (I2CxCON<5>)) は ACK または NACK のいずれかを指定します。

ACKEN ビットは 2 baud 周期の後に自動的にクリアされ、モジュールが MI2CxIF 割り込みを生成します。

19.5.4.1 IWCOL ステータスフラグ

肯定応答シーケンスの途中でソフトウェアが I2CxTRN レジスタへの書き込みを試みた場合、IWCOL ステータスビットがセットされ、バッファの内容は変更されません (書き込みは発生しません)。

Note: イベントのキューは許容されないため、I2CxCON レジスタの下位 5 ビットへの書き込みは肯定応答条件が完了するまで無効化されます。

図 19-10: マスタ肯定応答 ($\overline{\text{ACK}}$) のタイミング図

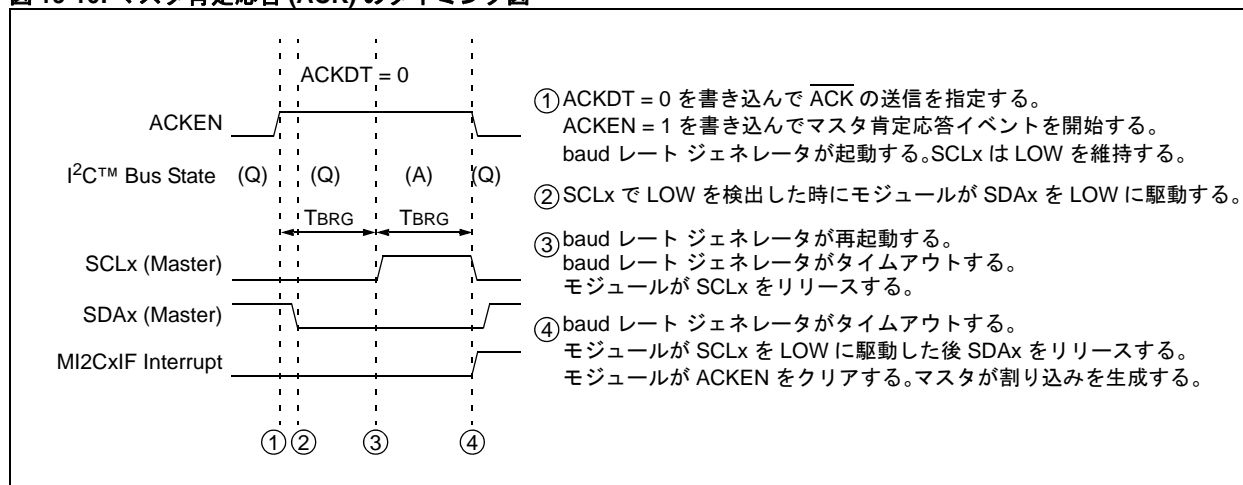
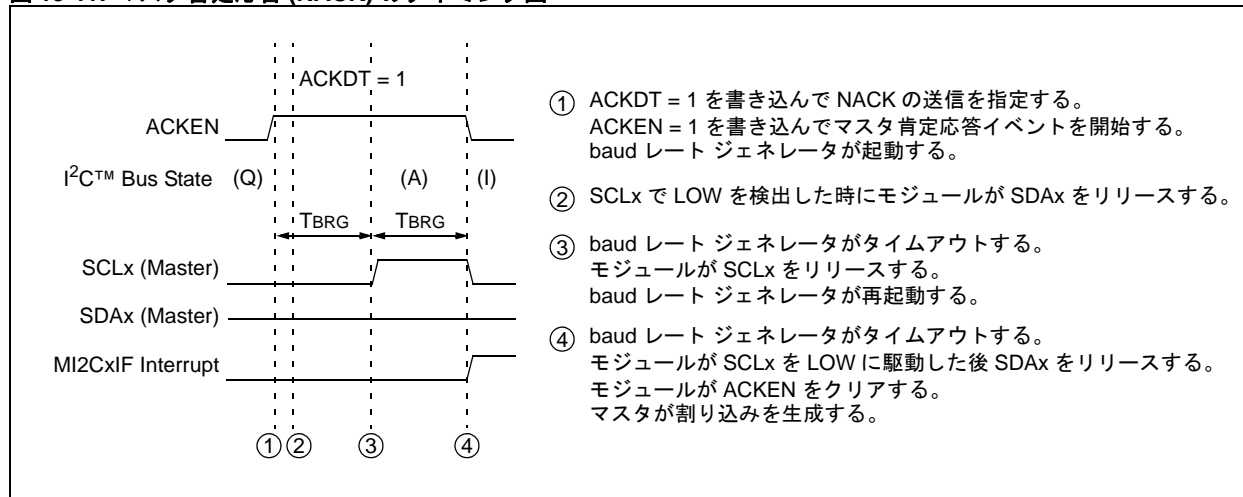


図 19-11: マスタ否定応答 (NACK) のタイミング図



19.5.5 STOP バスイベントの生成

STOP イネーブルビット (PEN (I2CxCON<2>)) をセットすると、マスタ STOP シーケンスの生成が有効化されます。

Note: PEN ビットをセットする前に、I2CxCON レジスタの下位 5 ビットが「0」(マスタロジックが無効)である必要があります。

PEN ビットをセットすると、マスタは図 19-12 のように STOP シーケンスを生成します。

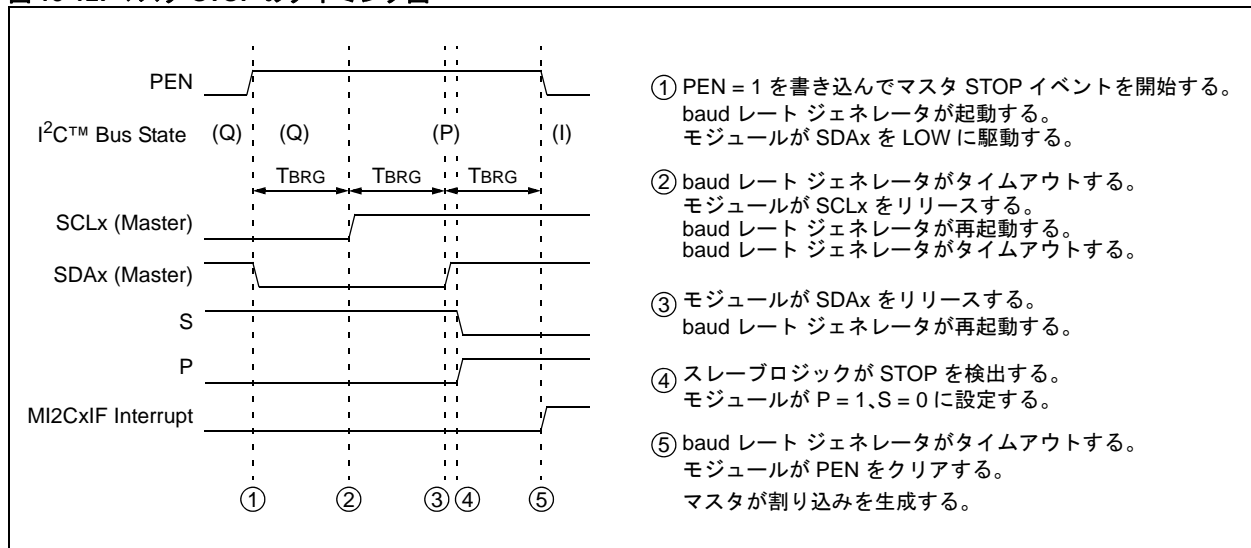
- スレーブは STOP 条件を検出した時に、P ステータスビット (I2CxSTAT<4>) をセットし、S ステータスビット (I2CxSTAT<3>) をクリアする
- PEN ビットが自動的にクリアされる
- モジュールが MI2CxIF 割り込みを生成する

19.5.5.1 IWCOL ステータスフラグ

STOP シーケンスの実行中にソフトウェアが I2CxTRN レジスタへの書き込みを試みた場合、IWCOL ステータスビットがセットされ、バッファの内容は変更されません (書き込みは発生しません)。

Note: イベントのキューは許容されないため、I2CxCON レジスタの下位 5 ビットへの書き込みは STOP 条件が完了するまで無効化されます。

図 19-12: マスタ STOP のタイミング図



19.5.6 リピートスタート バスイベントの生成

リピートスタート イネーブルビット (RSEN (I2CxCON<1>)) をセットすると、マスタ START シーケンスの生成が有効化されます (図 19-13 参照)。

Note: RSEN ビットをセットする前に、I2CxCON レジスタの下位 5 ビットが「0」(マスタ ロジックが無効)である必要があります。

リピートスタート条件を生成するために、ソフトウェアはRSEN ビット (I2CxCON<1>) をセットします。モジュールは SCLx ピンを LOW に駆動します。モジュールが SCLx ピンで LOW を検出した時、モジュールは baud レート ジェネレータの 1 カウント (TBRG) 間 SDAx ピンをリリースします。モジュールは、baud レート ジェネレータがタイムアウトしてモジュールが SDAx で HIGH を検出した時に SCLx ピンをリリースします。モジュールが SCLx ピンで HIGH を検出した時、baud レート ジェネレータはリロードしてカウントを開始します。SDAx と SCLx は 1 TBRG の間 HIGH を維持する必要があります。この動作の後に、SDAx ピンが 1 TBRG の間 LOW に駆動され、SCLx は HIGH を維持します。

リピートスタート シーケンスは下記の通りです。

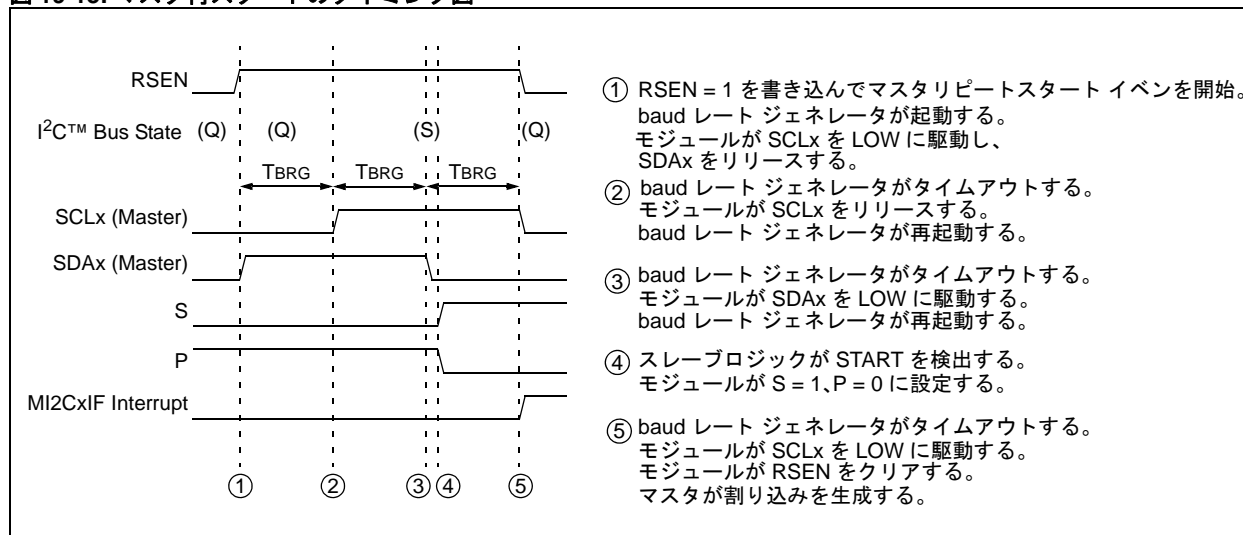
- スレーブは START 条件を検出した時に、S ステータスビット (I2CxSTAT<3>) をセットし、P ステータスビット (I2CxSTAT<4>) をクリアする
- RSEN ビットが自動的にクリアされる
- モジュールが MI2CxIF 割り込みを生成する

19.5.6.1 IWCOL ステータスフラグ

リピートスタート シーケンスの実行中にソフトウェアが I2CxTRN レジスタへの書き込みを試みた場合、IWCOL ステータスビットがセットされ、バッファの内容は変更されません (書き込みは発生しません)。

Note: イベントのキューは許容されないため、I2CxCON レジスタの下位 5 ビットへの書き込みはリピートスタート条件が完了するまで無効化されます。

図 19-13: マスタ再スタートのタイミング図



19.5.7 完全なマスタメッセージの作成

19.5「シングルマスタ環境でマスタとして通信する」の冒頭で述べたように、ソフトウェアがメッセージ プロトコルに従ってメッセージを構成します。モジュールは I²C メッセージ プロトコルの各部分を制御しますが、プロトコルの各部分をシーケンシングして完全なメッセージを構成するのはソフトウェアの役割です。

ソフトウェアは、モジュールを使用している時にポーリングまたは割り込みを使用できます。ここでは割り込みの使用例を説明します。

ソフトウェアはメッセージを処理する際に SEN、RSEN、PEN、RCEN、ACKEN ビット (I2CxCON レジスタの下位 5 ビット) と TRSTAT ステータスビットを「ステート」フラグとして使用できます。ステート番号とバス状態の割り当て例を表 19-2 に示します。

表 19-2: マスタメッセージのプロトコル ステート

ステート番号 (参考例)	I2CxCON<4:0>	TRSTAT (I2CxSTAT<14>)	状態
0	00000	0	バスはアイドルまたは待機中
1	00001	N/A	START イベントを送信中
2	00000	1	マスタが送信中
3	00010	N/A	リピータースタート イベントを送信中
4	00100	N/A	STOP イベントを送信中
5	01000	N/A	マスタが受信済
6	10000	N/A	マスタ肯定応答

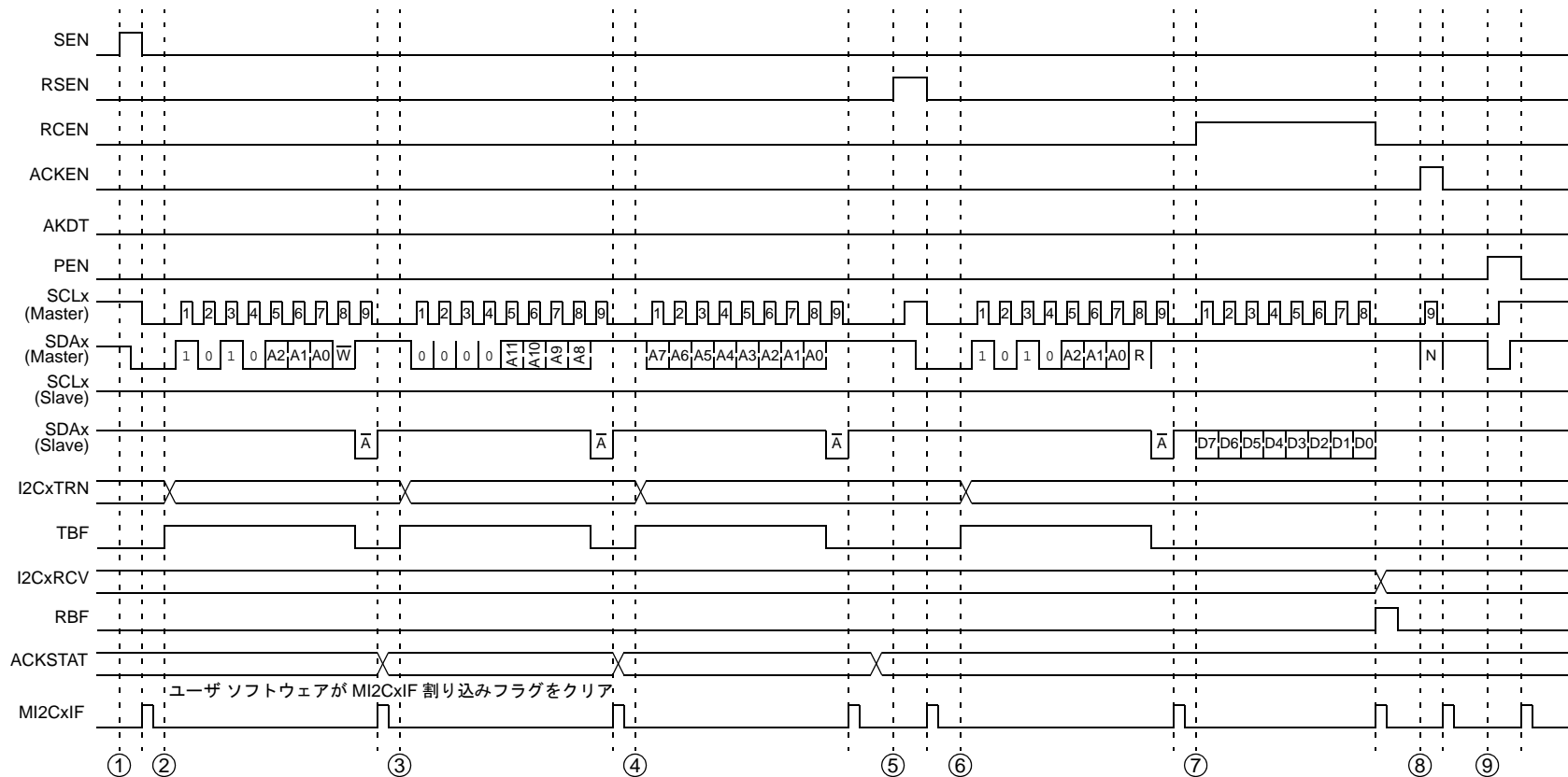
Note: 表のステート番号の割り当てはあくまでも参考例です。ユーザ ソフトウェアはステート番号を自由に割り当てる事ができます。

ソフトウェアは START 命令を発行する事によってメッセージを開始し、START に対応するステート番号を記録します。

各イベント完了時の割り込みが発生するたびに、割り込みハンドラはステート番号をチェックできます。START ステートの場合、割り込みハンドラは START シーケンスの実行を確認した後、I²C アドレスを送信してマスタ送信イベントを開始し、ステート番号をマスタ送信に対応した番号に変更します。

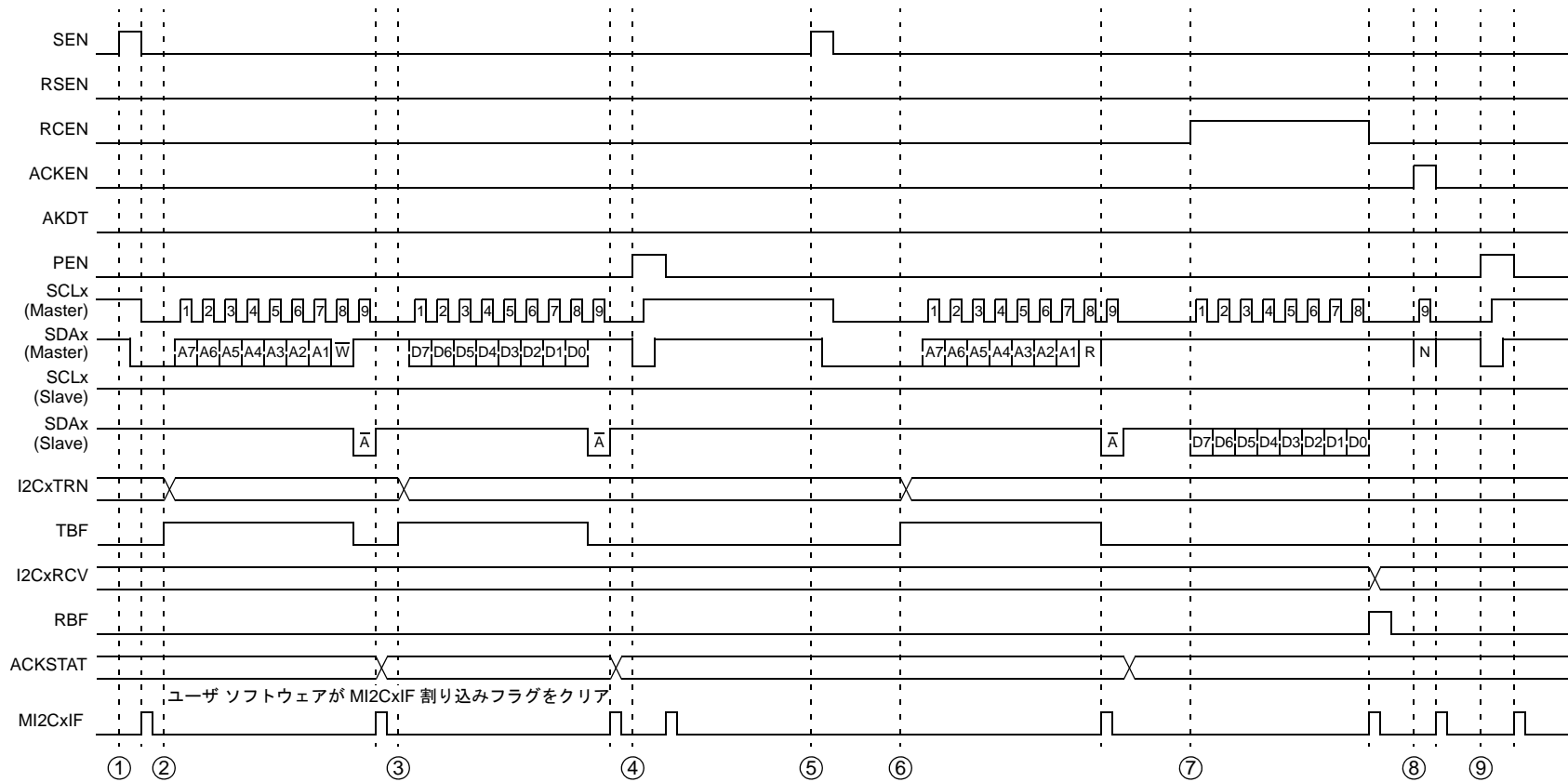
次の割り込み時に、割り込みハンドラはそのステートを再びチェックし、マスタ送信が完了したかどうかを判定します。割り込みハンドラは、データ送信に成功した事を確認した後、メッセージの内容に応じて次のイベントへ移行します。このようにして、メッセージの送信が完了するまで、割り込みハンドラが割り込みのたびにメッセージ プロトコルを処理します。

図 19-6 に示したメッセージ シーケンスのより詳細な解説を図 19-14 に示します。7 ビット アドレッシング フォーマットを使用するメッセージの簡単な例を図 19-15 に示します。10 ビット アドレッシング フォーマットを使用してスレーブヘータを送信するメッセージの例を図 19-16 に示します。10 ビット アドレッシング フォーマットを使用してメッセージをスレーブから受信する例を図 19-17 に示します。

図 19-14: マスタメッセージ (代表的な I²C™ メッセージ: シリアル EEPROM の読み出し)

- ① SEN ビットをセットして START イベントを開始する。
- ② I2CxTRN レジスタへの書き込みによってマスタ送信イベントを開始する。
このデータはシリアル EEPROM のデバイスアドレス バイトと R/W ステータス ビット (クリア、書き込み指定) を格納する。
- ③ I2CxTRN レジスタへの書き込みによってマスタ送信イベントを開始する。
このデータは EEPROM データアドレスの先頭バイトを格納する。
- ④ I2CxTRN レジスタへの書き込みによってマスタ送信イベントを開始する。
このデータは EEPROM データアドレスの第 2 バイトを格納する。
- ⑤ RSEN ビットをセットしてリピートスタート イベントを開始する。
- ⑥ I2CxTRN レジスタへの書き込みによってマスタ送信イベントを開始する。
R/W ステータスビットをセットした読み出し指定を含むシリアル EEPROM の
デバイスアドレス バイトを再送信する。
- ⑦ RCEN ビットをセットしてマスタ受信を開始する。
割り込み発生時にソフトウェアが I2CxRCV レジスタを読み出し、
RBF ステータスビットがクリアされる。
- ⑧ ACKEN ビットをセットして START イベントを開始する。
ACKDT = 1 に設定して NACK を送信する。
- ⑨ PEN ビットをセットしてマスタ STOP イベントを開始する。

図 19-15: マスタメッセージ (7 ビットアドレス : 送受信)



- ① SEN ビットをセットして START イベントを開始する。
- ② I2CxTRN レジスタへの書き込みによってマスタ送信イベントを開始する。
このデータはアドレスバイトとクリアされた R/W ステータスビットを格納する。
- ③ I2CxTRN レジスタへの書き込みによってマスタ送信イベントを開始する。
このデータはメッセージバイトを格納する。
- ④ PEN ビットをセットしてマスタ STOP イベントを開始する。
- ⑤ SEN ビットをセットして START イベントを開始する。
- ⑥ I2CxTRN レジスタへの書き込みによってマスタ受信イベントを開始する。
このデータはアドレスバイトとセットされた R/W ステータスビットを格納する。
- ⑦ RCEN ビットをセットしてマスタ受信を開始する。
- ⑧ ACKEN ビットをセットして肯定応答イベントを開始する。
ACKDT = 1 に設定して NACK を送信する。
- ⑨ PEN ビットをセットしてマスタ STOP イベントを開始する。

図 19-16: マスタメッセージ (10 ビット送信)

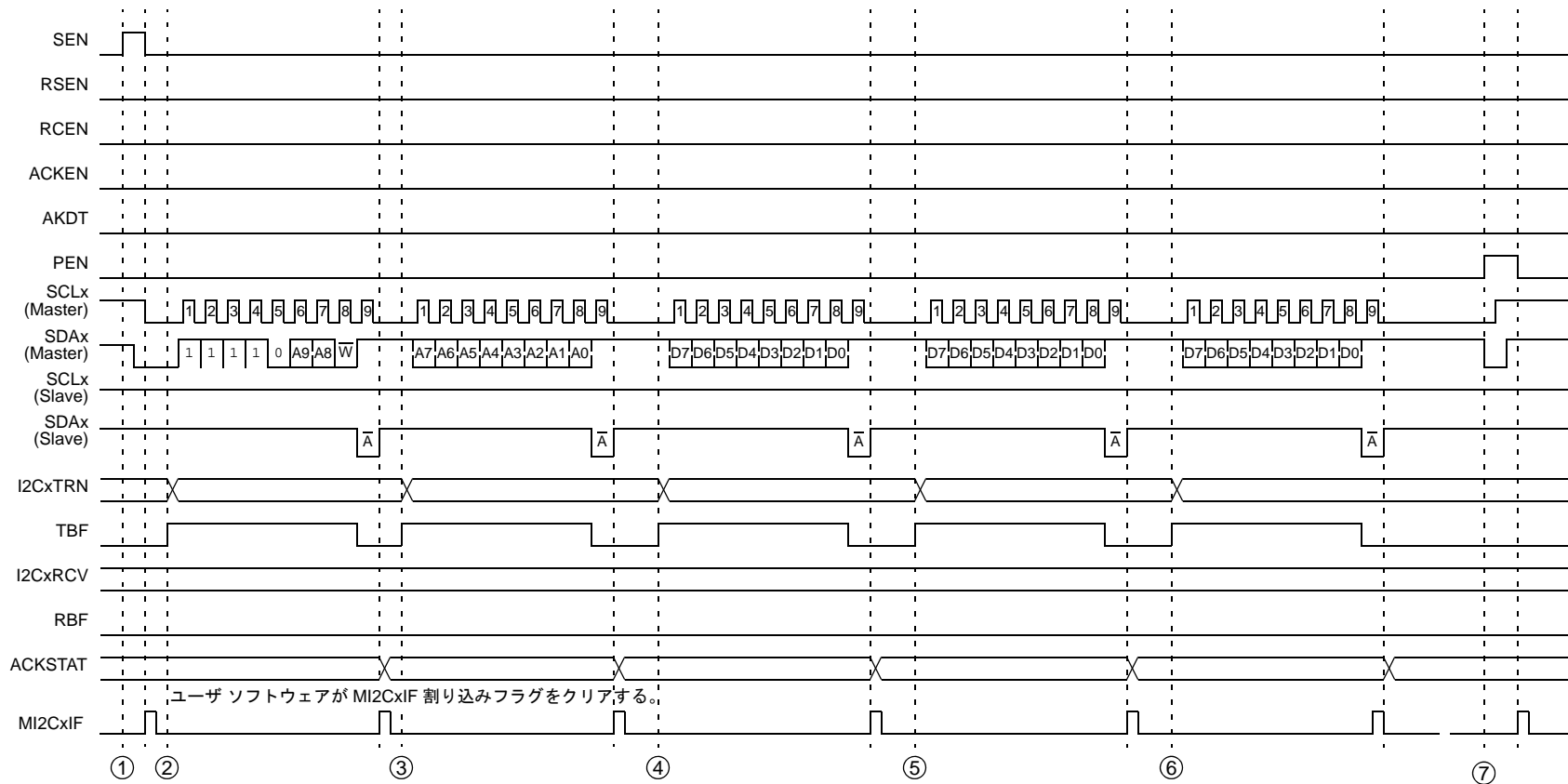
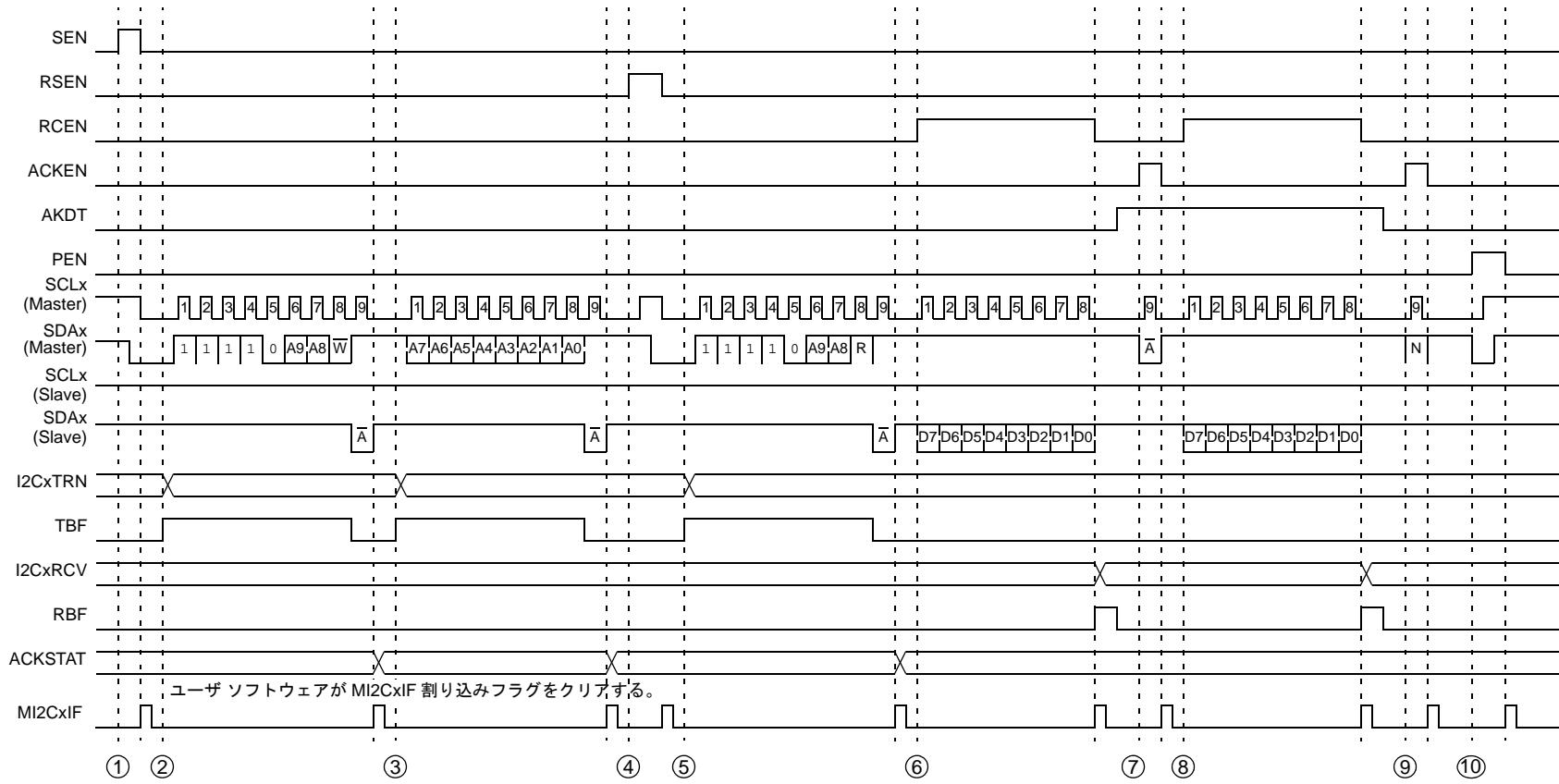


図 19-17: マスタメッセージ (10 ビット受信)



- ① SEN ビットをセットして START イベントを開始する。
- ② I2CxTRN レジスタへの書き込みによってマスタ送信イベントを開始する。
このデータは R/W ステータスビットがクリアされたアドレス先頭バイトを格納する。
- ③ I2CxTRN レジスタへの書き込みによってマスタ送信イベントを開始する。
このデータはアドレスの第 2 バイトを格納する。
- ④ RSEN ビットをセットしてマスタリポートスタート イベントを開始する。
- ⑤ I2CxTRN レジスタへの書き込みによってマスタ送信イベントを開始する。
このデータは R/W ステータスビットをセットしたアドレス先頭バイト (再送信) を格納する。
- ⑥ RCEN ビットをセットしてマスタ受信を開始する。
割り込み発生時にソフトウェアが I2CxRCV レジスタを読み出し、
これにより RBF ステータスビットをクリアされる。
- ⑦ ACKEN ビットをセットして肯定応答イベントを開始する。
ACKDT = 0 に設定して ACK を送信する。
- ⑧ RCEN ビットをセットしてマスタ受信を開始する。
- ⑨ ACKEN ビットをセットして肯定応答イベントを開始する。
ACKDT = 1 に設定して NACK を送信する。
- ⑩ PEN ビットをセットしてマスタ STOP イベントを開始する。

19.6 マルチマスタ環境でマスタとして通信する

I²C プロトコルでは、単一システムバスに複数のマスタを接続できます。各マスタはメッセージのトランザクションを開始してバスにクロックを供給するため、プロトコルは複数のマスタがバス制御を同時に試みた場合の対処方法を備えます。クロック同期機能により、複数ノードの各 SCLx クロックを SCLx ライン上の共通クロックに同期させる事ができます。複数ノードがメッセージのトランザクションを試みた場合、バス調停によって 1 つのノードだけがメッセージを完了できます。他のノードはバス調停に敗れてバス衝突状態に置かれます。

Note: マスタとして動作している時に IPMIEN (I2CxCON<11>) ビットをセットしない必要があります。

19.6.1 マルチマスタの動作

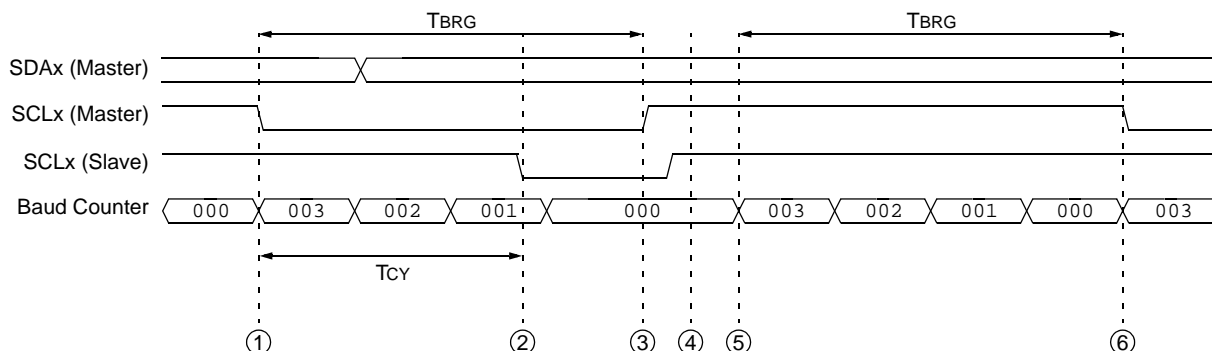
マスタモジュールはマルチマスタ動作を有効にするための特別な設定を持ちません。モジュールはクロック同期とバス調停を常時実行します。モジュールをシングルマスタ環境で使用する場合、クロック同期はそのマスタと複数スレーブ間でのみ発生し、バス調停は発生しません。

19.6.2 マスタクロックの同期

マルチマスタ システムでは、マスタ間で baud レートが異なる可能性があります。クロック同期は、これらのマスタがバス調停を試みた時に、それらのクロックを調和させます。

クロック同期はマスタが SCLx ピンをリリース (SCLx を HIGH にフロート) した時に発生します。SCLx ピンがリリースされると、baud レート ジェネレータ (BRG) のカウンタは SCLx ピンで実際に HIGH を検出するまで一時停止します。SCLx ピンで HIGH を検出すると、baud レート ジェネレータに I2CxBRG<8:0> の内容がリロードされ、カウンタが開始されます。これにより、他のデバイスがクロックを LOW に保持する場合、SCLx HIGH 時間は常に 1 回の BRG ロールオーバー カウント時間以上となります (図 19-18 参照)。

図 19-18: クロック同期による baud レート ジェネレータのタイミング



- ① baud カウンタは Tcy あたり 2 回デクリメントする。ロールオーバー時にマスタ SCLx の状態が変化する。
- ② スレーブが SCLx を LOW にして待機を開始する。
- ③ マスタ baud カウンタがロールオーバーする時、SCLx で LOW を検出するとカウンタが一時停止する。
- ④ ロジックは Tcy あたり SCLx を 1 回サンプリングする。ロジックが SCLx HIGH を検出する。
- ⑤ baud カウンタのロールオーバーが次のサイクルで発生する。
- ⑥ 次のロールオーバーでマスタ SCLx の状態が変化する。

19.6.3 バス調停とバス衝突

バス調停はマルチマスタ システムの動作をサポートします。ワイヤード AND 機能によりバス調停が可能になります。最初のマスタが SDAx を HIGH にフロートして SDAx で「1」を出力すると同時に 2 番目のマスタが SDAx を LOW にして SDAx で「0」を出力すると、バス調停が発生します。SDAx 信号は LOW に変化します。この場合、2 番目のマスタがバス調停に勝ちました。最初のマスタはバス調停に敗れたため、バス衝突が発生します。

最初のマスタは SDAx 上のデータを「1」にしようと試みましたが、SDAx で検出されるデータは「0」のままです。このような状態をバス衝突と定義します。

最初のマスタはマスタ バス衝突 ビット (BCL (I2CxSTAT<10>)) をセットし、マスタ割り込みを生成します。このマスタモジュールは I²C ポートをアイドル状態にリセットします。

マルチマスタ動作ではバス調停が発生する可能性があるため、実際の信号レベルが要求した出力レベルに一致するかどうかを確認するために、SDAx ラインを監視する必要があります。マスタモジュールはこの確認を実行し、その結果を BCL ステータスビットに反映します。

下記のステートではバス調停に敗れる可能性があります。

- START 条件
- リピートスタート条件
- アドレス、データ、肯定応答ビット
- STOP 条件

19.6.4 バス衝突の検出とメッセージの再送

バス衝突が発生すると、モジュールは BCL ステータスビットをセットし、マスタ割り込みを生成します。バイト送信中にバス衝突が発生した場合、その送信は中止され、TBF ステータスビットがクリアされ、SDAx および SCLx ピンがリリリースされます。START、リピートスタート、STOP、肯定応答のいずれかの条件でバス衝突が発生した場合、I2CxCON レジスタの対応する制御ビットがクリアされ、SDAx および SCLx ラインがリリリースされます。

ソフトウェアはマスタイベント完了時の割り込みを待機します。ソフトウェアは、BCL ステータスビットをチェックする事によって、マスタイベントが正常に完了したのか、それともバス衝突が発生したのを判定できます。バス衝突が発生した場合、ソフトウェアは保留中メッセージの以降の送信を中止し、バスがアイドル状態に戻った時に START 条件から始まる完全なメッセージを再送するために準備する必要があります。ソフトウェアは S および P ステータスビットを監視する事によってバスのアイドル状態を待機できます。ソフトウェアがマスタ割り込みサービスルーチンを処理した時に I²C バスがフリーであれば、ソフトウェアは START 条件を生成する事によって通信を再開できます。

19.6.5 START 条件中のバス衝突

ソフトウェアは、START 命令を発行する前に、S および P ステータスビットを使用してバスがアイドル状態である事を確認する必要があります。しかし、2 つのマスタが同時にメッセージの開始を試みる可能性があります。一般的に、両マスタはクロックを同期し、一方が調停に敗れるまでメッセージへの調停を続けます。しかし下記の条件では、START 中にバス衝突が発生する可能性があります。

- START 条件の開始時に SDA および SCL ピンが論理 LOW 状態であった場合、または、
- SDA ラインが LOW に駆動される前に SCL ラインが論理 LOW 状態であった場合

いずれの場合でも、START ビット送信時にバス調停に敗れたマスタは、バス衝突割り込みを生成します。

19.6.6 リピートスタート条件中のバス衝突

2 つのマスタがアドレスバイトの送信で衝突しなかった場合、一方のマスタがリピートスタートの生成を試み、他方がデータの送信を試みた時にバス衝突が発生する可能性があります。この場合、リピートスタートの生成を試みたマスタが調停に敗れてバス衝突割り込みを生成します。

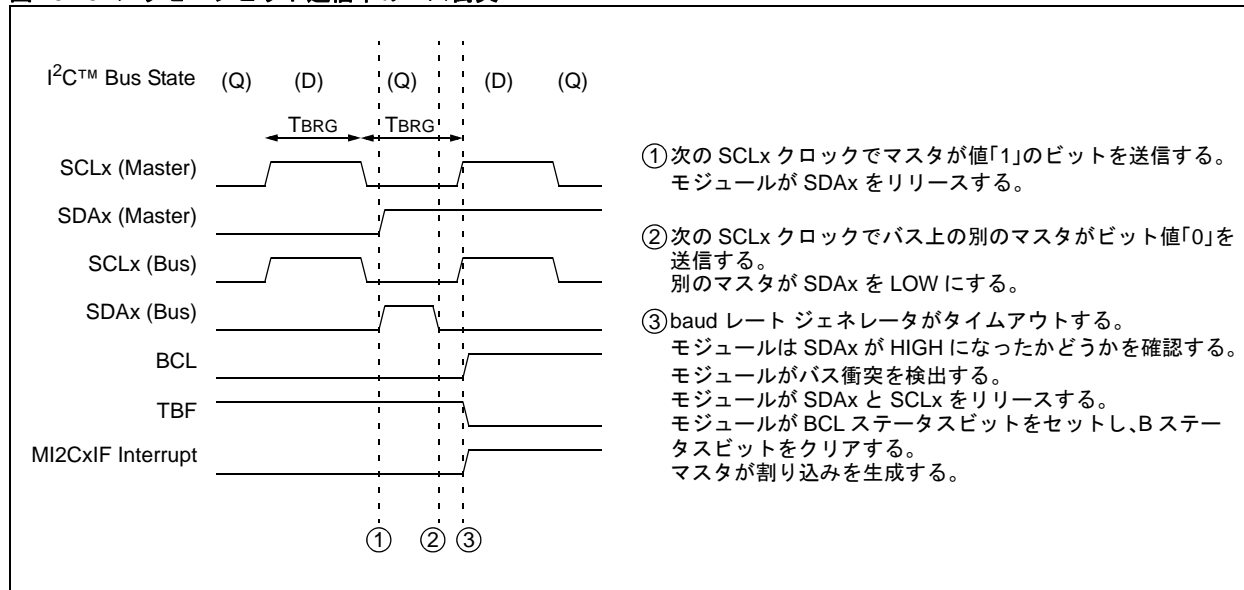
19.6.7 メッセージビット送信中のバス衝突

データ衝突が発生する最も典型的な状況は、マスタがデバイスアドレス バイト、データバイト、肯定応答ビットの送信を試みる時です。

ソフトウェアがバスの状態を正しく監視している限り、START 条件中のバス衝突はほとんど発生しません。しかし、別のマスタが同時にバスをチェックして START 条件の開始を試みる可能性もあり、このような場合 SDAx 調停が発生して 2 つのマスタの START を同期する可能性があります。このような条件では、両方のマスタがメッセージの送信を開始し、メッセージビットの調停で一方のマスタが敗れるまで送信を続けます。SCLx クロック同期は、一方のマスタが調停に敗れるまで 2 つのマスタ間の同期を維持する事に注意してください。

図 19-19 に、メッセージビット調停の例を示します。

図 19-19: メッセージビット送信中のバス衝突



19.6.8 STOP 条件中のバス衝突

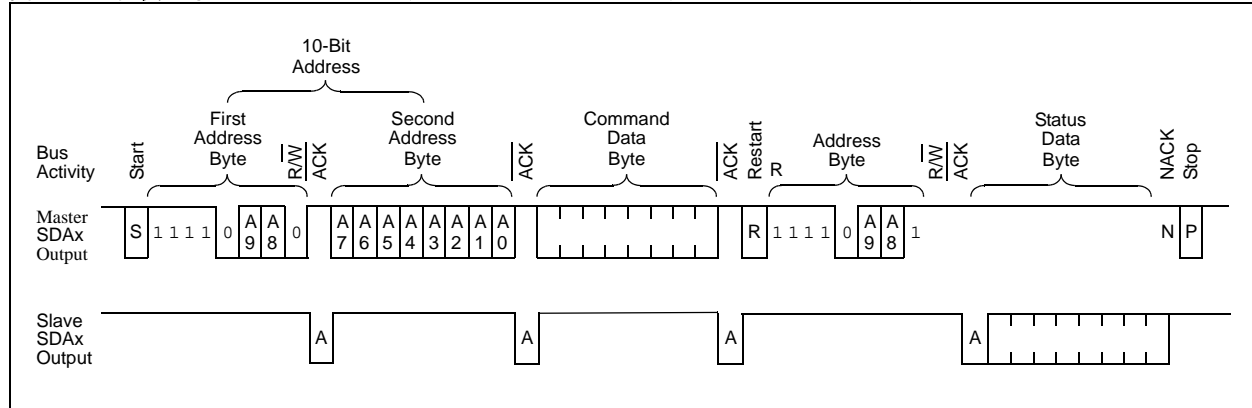
マスタ ソフトウェアが I²C バスの状態を見失った場合、STOP 条件中にバス衝突が発生し得る状況は極めて多数存在するため、ここにそれらを列挙する事はできません。この場合、STOP 条件の生成を試みたマスタが調停に敗れてバス衝突割り込みを生成します。

19.7 スレーブとしての通信

一部のシステム (特に複数のプロセッサが互いに通信するシステム) では、dsPIC33F/PIC24H はスレーブとして通信できます (図 19-20 参照)。モジュールを有効にすると、スレーブ モジュールが起動します。スレーブはメッセージを開始できず、マスタが開始したメッセージ シーケンスに応答するしかできません。マスタは、I²C プロトコル内のデバイスアドレス バイトが指定する特定スレーブからの応答を要求します。スレーブ モジュールは、プロトコルが定める適切なタイミングでマスタに応答します。

マスタ モジュールと同様に、応答用プロトコルの各部をシーケンシングするのはソフトウェア の役割です。アドレス一致 (ソフトウェアがそのスレーブに対して指定したアドレスとデバイス アドレスの一致) の検出は、スレーブ モジュールが行います。

図 19-20: 代表的なスレーブ I²C™ メッセージ: マルチプロセッサ コマンド / ステータス



START 条件の後に、スレーブ モジュールはデバイスアドレスを受信してアドレス一致をチェックします。スレーブは 7 ビットアドレスか 10 ビットアドレスのいずれかを指定できます。デバイスアドレスが一致すると、モジュールは割り込みを生成してこのデバイスがマスタによって選択されている事をソフトウェアに知らせます。スレーブは、マスタが送信した R/W ステータスビットの状態に基づいてデータを受信または送信します。スレーブがデータを受信する場合、スレーブ モジュールは自動的に肯定応答 (ACK) を生成し、I2CxRSR レジスタに現在格納されている受信値を I2CxRCV レジスタに転送し、割り込みを生成してソフトウェアに知らせます。スレーブがデータを送信する場合、ソフトウェアは I2CxTRN レジスタにデータを書き込む必要があります。

19.7.1 受信データのサンプリング

全ての受信ビットはクロック (SCLx) ラインの立ち上がりエッジでサンプリングされます。

19.7.2 START および STOP 条件の検出

スレーブ モジュールは、バス上の START および STOP 条件を検出し、そのステータスを S ステータスビット (I2CxSTAT<3>) と P ステータスビット (I2CxSTAT<4>) で示します。START (S) および STOP (P) ステータスビットは、リセット発生時またはモジュールが無効化された時にクリアされます。START またはリピートスタート イベントを検出した場合、S ステータスビットがセットされ、P ステータスビットがクリアされます。STOP イベントを検出した場合、P ステータスビットがセットされ、S ステータスビットがクリアされます。

19.7.3 アドレスの検出

モジュールが有効化されると、スレーブ モジュールは START 条件の発生を待機します。START 発生後、スレーブは A10M ビット (I2CxCON<10>) の状態に応じて 7 ビットまたは 10 ビットアドレスの検出を試みます。スレーブ モジュールは、1 バイト (7 ビットアドレスの場合) または 2 バイト (10 ビットアドレスの場合) の受信バイトを比較します。7 ビットアドレスのアドレスバイトは、アドレス値の後にデータ転送の方向 (読み / 書き) を指定する R/W ステータスビットも格納します。R/W = 0 は書き込みを指定し、スレーブはマスタからデータを受信します。R/W = 1 は読み出しを指定し、スレーブはマスタへデータを送信します。10 ビットアドレスの先頭アドレスバイトも R/W ステータスビットを含みますが、規則により R/W は常に「0」です (スレーブは 10 ビットアドレスの第 2 バイトを次に受信する必要があるため)。

19.7.3.1 スレーブアドレスのマスキング

I2CxMSK レジスタは、アドレスバイトの各ビット位置を「ドントケア」ビットとして指定する事によってアドレスをマスキングします。この機能は 10 ビットおよび 7 ビットのどちらのアドレッシング モードでも使用できます。I2CxMSK レジスタ内のいずれかのビットをセット (= 1) すると、スレーブ モジュールはアドレス内の対応するビット位置が「0」であっても「1」であっても応答します。例えば、7 ビット スレーブモードで I2CxMSK = 0100000 の場合、モジュールはアドレス「0000000」と「0100000」に応答します。

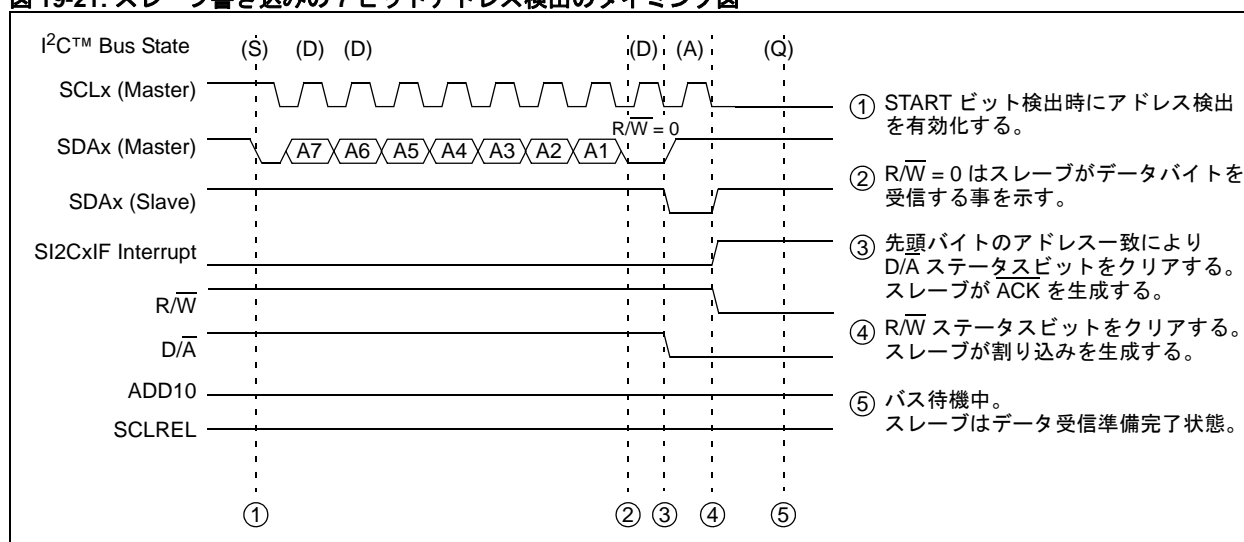
アドレス マスキングを有効化する場合、IPMIEN ビット (I2CxCON<11>) をクリアする事によって IPMI (Intelligent Platform Management Interface) を無効化する必要があります。

19.7.3.2 7 ビットアドレスとスレーブ書き込み

START 条件の後に、モジュールは 8 ビットを I2CxRSR レジスタにシフトインします (図 19-21 参照)。レジスタ I2CxRSR<7:1> の値は、クロック (SCLx) の 8 番目の立ち下がりエッジで、I2CxADD<6:0> および I2CxMSK<6:0> レジスタの値と比較されます。アドレスが有効な場合 (マスキングされていない全てのビット位置が一致した場合)、下記のイベントが発生します。

1. $\overline{\text{ACK}}$ を生成する
2. $\text{D}/\overline{\text{A}}$ および $\text{R}/\overline{\text{W}}$ ステータスビットをクリアする
3. 9 番目の SCLx クロック立ち下がりエッジで SI2CxIF 割り込みを生成する
4. マスタからのデータ送信を待機する

図 19-21: スレーブ書き込みの 7 ビットアドレス検出のタイミング図



19.7.3.3 7 ビットアドレスによるスレーブ読み出し

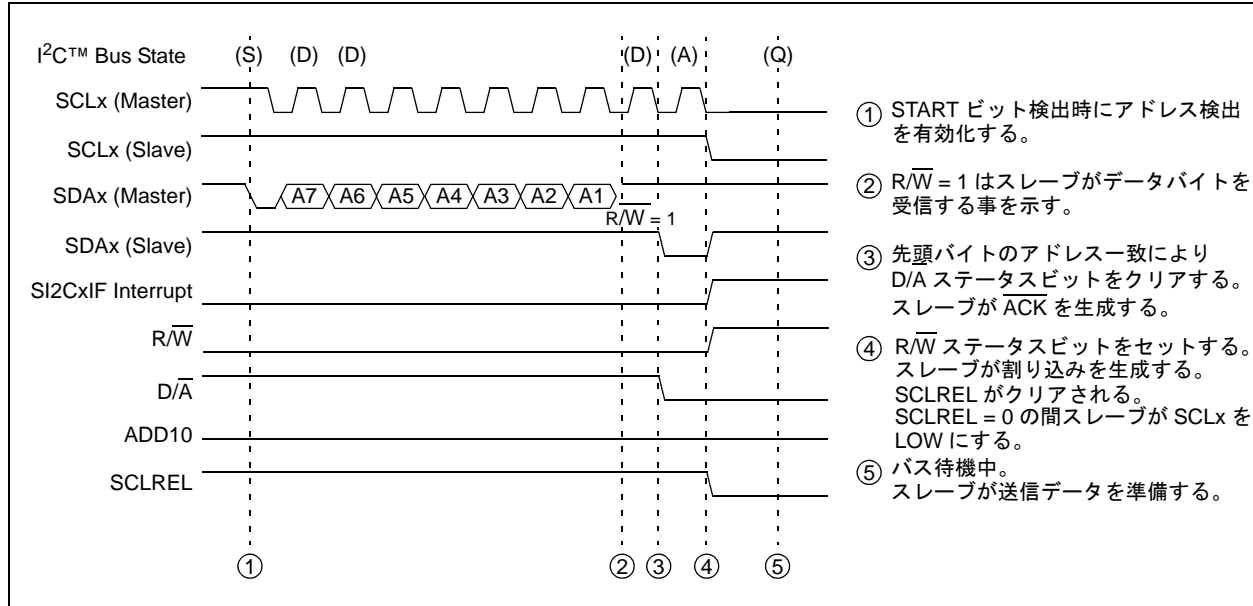
7 ビットアドレスバイトの $\text{R}/\overline{\text{W}} = 1$ によってスレーブ読み出しが指定された場合のデバイスアドレス検出プロセスは、スレーブ書き込みの場合と同様です (図 19-22 参照)。アドレスが一致すると下記のイベントが発生します。

- $\overline{\text{ACK}}$ を生成する
- $\text{D}/\overline{\text{A}}$ ステータスビットをクリアし、 $\text{R}/\overline{\text{W}}$ ステータスビットをセットする
- 9 番目の SCLx クロックサイクル立ち下がりエッジで SI2CxIF 割り込みを生成する

この時点でスレーブ モジュールにはデータを返送する事が期待されるため、モジュールはソフトウェアが応答を準備できるまで I2C バスの動作を一時停止します。これは、モジュールが SCLREL ビットをクリアした時に自動的に行われます。SCLREL が LOW の場合、スレーブ モジュールは SCLx クロックラインを LOW にして、I2C バスを待機状態にします。スレーブ モジュールと I2C バスは、ソフトウェアが I2CxTRN レジスタに応答データを書き込んで SCLREL ビットをセットするまで、この状態を維持します。

Note: SCLREL は、STREN ビットの状態に関係なく、スレーブ読み出しアドレスの検出後に自動的にクリアされます。

図 19-22: スレーブ読み出しの 7 ビットアドレス検出のタイミング図



19.7.3.4 10 ビット アドレッシング モード

10 ビット アドレッシング モードでは、スレーブは 2 つのデバイスアドレス バイトを受信する必要があります (図 19-23 参照)。先頭アドレスバイトの上位 5 ビットは、アドレッシング モードが 10 ビットである事を指定します。スレーブデバイスは第 2 アドレスバイトも受信する必要がありますため、先頭アドレスバイトの R/\overline{W} ステータスビットは書き込みを指定する必要があります。10 ビットアドレスの先頭バイトは「11110 A9 A8 0」(A9 と A8 はアドレスの上位 2 ビット) となります。

I2CxMSK レジスタは、10 ビットアドレスの任意のビット位置をマスキングできます。I2CxMSK レジスタの上位 2 ビットは、先頭バイトで受信する 10 ビットアドレスの上位 2 ビットをマスキングします。I2CxMSK レジスタの下位バイトは、第 2 バイトで受信する 10 ビットアドレスの下位バイトをマスキングします。

START 条件の後に、モジュールは 8 ビットを I2CxRSR レジスタにシフトインします。I2CxRSR<2:1> ビットの値は、I2CxADD<9:8> および I2CxMSK<9:8> ビットの値と比較され、I2CxRSR<7:3> ビットの値は「11110」と比較されます。アドレスの比較はクロック (SCLx) の 8 番目の立ち下がりエッジで発生します。アドレスが有効となるには、I2CxRSR<7:3> が「11110」に一致し、I2CxRSR<2:1> が I2CxADD<9:8> 内のマスクされていない全てのビット位置で一致する必要があります (2 つのビットが両方ともマスクされている場合、一致は不要)。アドレスが一致すると下記のイベントが発生します。

1. \overline{ACK} を生成する
2. D/\overline{A} および R/\overline{W} ステータスビットをクリアする
3. 9 番目の SCLx クロックサイクル立ち下がりエッジで SI2CxIF 割り込みを生成する

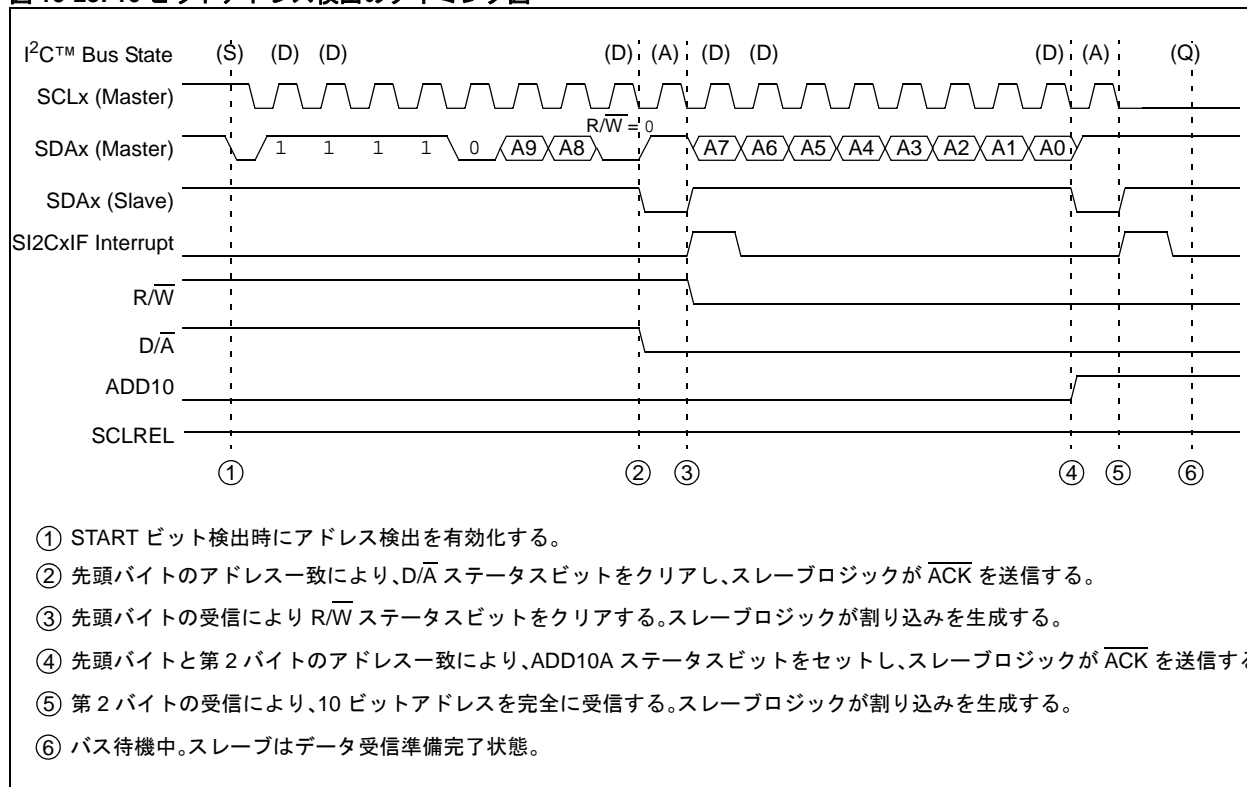
モジュールは 10 ビットアドレスの先頭バイトを受信した後に割り込みを生成しますが、この割り込みは特に効果を持ちません。

モジュールは続けて第 2 バイトを I2CxRSR レジスタに受信します。この時に I2CxRSR<7:0> ビットが I2CxADD<7:0> および I2CxMSK<7:0> ビットと比較されます。上位 2 ビットと同様にアドレスの下位バイトも有効であった場合、下記のイベントが発生します。

1. \overline{ACK} を生成する
2. ADD10 ステータスビットをセットする
3. 9 番目の SCLx クロックサイクル立ち下がりエッジで SI2CxIF 割り込みを生成する
4. モジュールはマスタからのデータ送信またはリピートスタート条件の開始を待機する

Note: 10 ビット アドレッシング モードでは、リピートスタート条件後に、スレーブ モジュールは先頭の 7 ビットのアドレス「11110 A9 A8 0」の一致だけを検出します。

図 19-23: 10 ビットアドレス検出のタイミング図



19.7.3.5 ゼネラルコール動作

通常の I²C バスのアドレッシング プロセスでは、START 条件後の先頭バイトによってマスタがアドレッシングするスレーブデバイスが指定されます。例外として、ゼネラルコール アドレスは全てのデバイスをアドレッシング可能です。このアドレスを使用すると、有効化された全てのデバイスが肯定応答を返します。ゼネラルコール アドレスは、I²C プロトコルが特殊な用途向けに予約している 8 つのアドレスの 1 つです。このアドレスは全て「0」のビットで構成され、R/W ビットも「0」です。ゼネラルコールは常にスレーブ書き込み動作です。

ゼネラルコール アドレスは、ゼネラルコール イネーブルビット (GCEN (I2CxCON<7>)) がセットされている時に認識されます (図 19-24 参照)。START ビットの検出後、8 ビットが I2CxRSR レジスタにシフトインされ、そのアドレスが I2CxADD レジスタおよびゼネラルコール アドレスと比較されます。

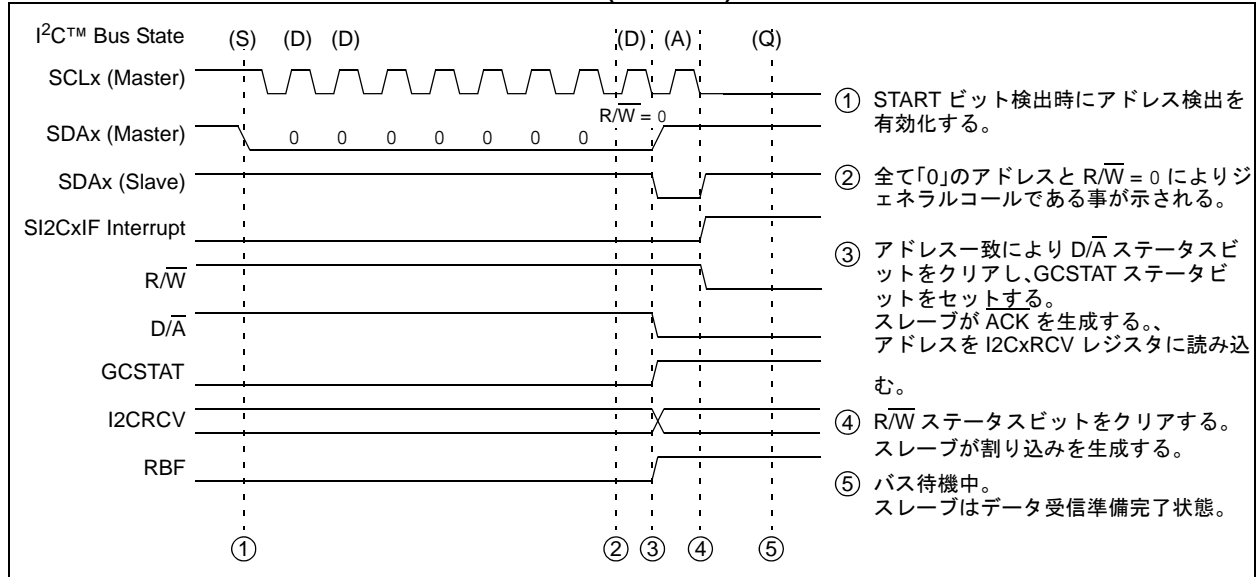
ゼネラルコール アドレスが一致すると、下記のイベントが発生します。

1. ACK を生成する
2. スレーブ モジュールが GCSTAT ステータスビット (I2CxSTAT<9>) をセットする
3. D/A および R/W ステータスビットをクリアする
4. 9 番目の SCLx クロックサイクル立ち下がりで SI2CxIF 割り込みを生成する
5. I2CxRSR レジスタの内容を I2CxRCV レジスタに転送し、RBF ステータスビットをセットする (第 8 ビット転送時)
6. マスタからのデータ送信を待機する

割り込みがサービスされた時に GCSTAT ステータスビットの内容を読み出す事によって、デバイスアドレスとゼネラルコール アドレスのどちらが一致したのかを判別できます。

ゼネラルコール アドレスは 7 ビットアドレスです。A10M ビットをセットしてスレーブ モジュールを 10 ビットアドレス向けに設定している場合、GCEN ビットをセットするとスレーブ モジュールは 7 ビットのゼネラルコール アドレスを検出し続けます。

図 19-24: ゼネラルコール アドレス検出のタイミング図 (GCEN = 1)

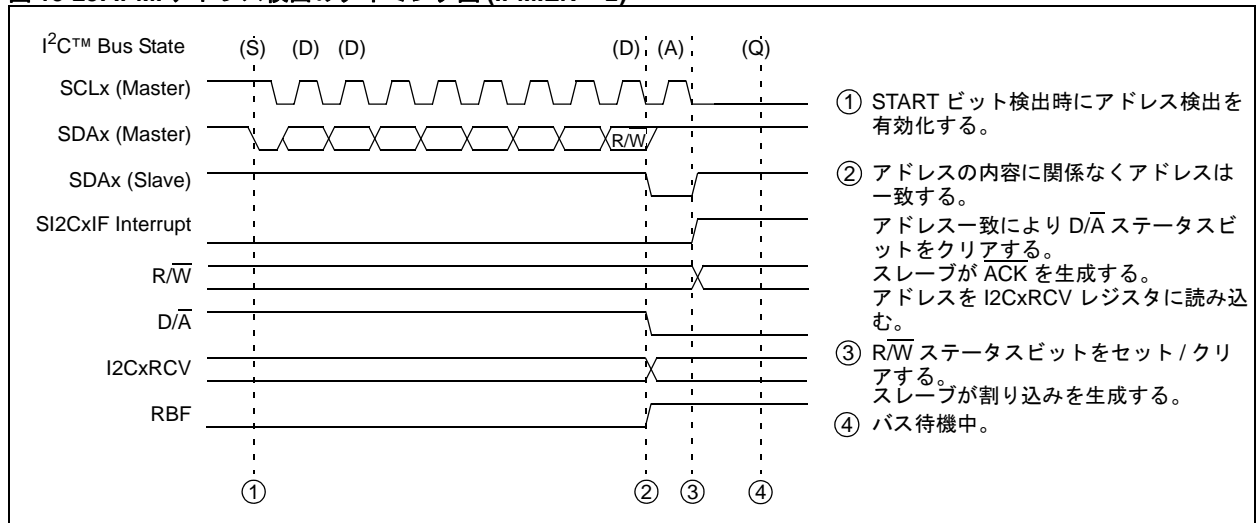


19.7.3.6 全てのアドレスを受信する (IPMI 動作)

一部の I²C システム プロトコルは、スレーブに対してバス上の全てのメッセージに応答する事を要求します。例えば IPMI (Intelligent Platform Management Interface) バスは、分散型ネットワーク内で I²C ノードをメッセージ リピータとして使用します。ノードが全てのメッセージをリピートするには、スレーブ モジュールがデバイスアドレスに関係なく全てのメッセージを受け入れる必要があります。

IPMI モードを有効にするには、IPMIEN ビット (I2CxCON<11>) をセットします (図 19-25 参照)。A10M および GCEN ビットの状態または I2CxADD レジスタに書き込まれた値に関係なく、全てのアドレスが受け入れられます。これには有効な 7 ビットアドレス、ゼネラルコール、START バイト、CBUS、予約および HS モード、さらに 10 ビットアドレスのプリアンブルが全て含まれます。

図 19-25: IPMI アドレス検出のタイミング図 (IPMIEN = 1)



Note: ユーザ アプリケーションは I²C マスタとして動作する時に IPMIEN (I2CxCON<11>) ビットをクリアし、IPMI スレーブとして動作する時にこのビットをセットする必要があります。

19.7.3.7 アドレスが無効な時

7 ビットアドレスが I2CxADD<6:0> の内容と一致しない場合、スレーブ モジュールはアイドル状態に戻り、STOP 条件後の状態になるまで全てのバス動作を無視します。

10 ビットアドレスの先頭バイトが I2CxADD<9:8> の内容と一致しない場合、スレーブ モジュールはアイドル状態に戻り、STOP 条件後の状態になるまで全てのバス動作を無視します。

10 ビットアドレスの先頭バイトは I2CxADD<9:8> の内容と一致したが、第 2 バイトが I2CxADD<7:0> の内容に一致しない場合、スレーブ モジュールはアイドル状態に戻り、STOP 条件後の状態になるまで全てのバス動作を無視します。

19.7.3.8 マスキングから除外される予約アドレス

マスキングを有効にしても、ハードウェアでマスキングから除外される各種のアドレスが存在します。これらのアドレスに対しては、マスキング設定に関係なく肯定応答は生成されません。これらのアドレスを表 19-3 に示します。

表 19-3: 予約済み I²C バスアドレス ⁽¹⁾

7 ビットアドレス モード		
スレーブアドレス	R/W ビット	内容
0000 000	0	ゼネラルコール アドレス ⁽¹⁾
0000 000	1	START バイト
0000 001	x	CBUS アドレス
0000 010	x	予約
0000 011	x	予約
0000 1xx	x	HS モード マスタコード
1111 1xx	x	予約
1111 0xx	x	10 ビットスレーブ上位バイト ⁽²⁾

Note 1: アドレスは GCEN = 1 の場合にのみ肯定応答されます。

2: このアドレスとの一致は、10 ビット アドレッシング モードの上位バイトとしてのみ発生可能です。

19.7.4 マスタデバイスからデータを受信する

R/W ステータスビットがゼロに設定されたデバイスアドレス バイトとアドレスが一致すると、R/W ステータスビット (I2CxSTAT<2>) がクリアされます。スレーブ モジュールはマスタが送信するデータを待機します。デバイスアドレス バイトに続くデータバイトの内容は、システムプロトコルによって定義され、当該スレーブだけがそのデータを受信します。

スレーブ モジュールは 8 ビットを I2CxRSR レジスタにシフトインします。クロック (SCLx) の 8 番目の立ち下がりがエッジで下記のイベントが発生します。

1. モジュールが ACK または NACK の生成を開始する
2. RBF ステータスビットをセットしてデータを受信した事を示す
3. I2CxRSR レジスタバイトの内容を、ソフトウェアからアクセス可能な I2CxRCV レジスタに転送する
4. D/A ステータスビットをセットする
5. スレーブ割り込みを生成する (ソフトウェアは、I2CxSTAT レジスタのステータスをチェックする事によってイベントの発生原因を特定した後に、SI2CxIF 割り込みフラグをクリアできす)
6. モジュールは次のデータバイトを待機する

19.7.4.1 肯定応答の生成

通常、スレーブ モジュールは受信した全てのバイトに対して、9 番目の SCLx クロックで $\overline{\text{ACK}}$ を送信する事によって肯定応答を返します。受信バッファがオーバーランした場合、スレーブ モジュールは $\overline{\text{ACK}}$ を生成しません。オーバーランは下記のいずれかまたは両が発生した場合に示されます。

1. 転送を受信する前にバッファフル ビット (RBF (I2CxSTAT<1>)) がセットされた
2. 転送を受信する前にオーバーフロー ビット (I2COV (I2CxSTAT<6>)) がセットされた

表 19-4 に、データ転送バイトを受信した時の RBF および I2COV ステータスビットの状態と、その時に発生する動作を示します。スレーブ モジュールが I2CxRCV レジスタへの転送を試みた時に RBF ステータスが既にセットされていた場合、転送は発生せずに割り込みが生成され、I2COV ステータスビットがセットされます。RBF および I2COV ステータスビットが両方ともセットされていた場合も、スレーブ モジュールは上記と同様に動作します。テーブル内の網掛け部分は、ソフトウェアがオーバーフロー条件を正しくクリアしなかった状態を示します。

I2CxRCV レジスタを読み出すと、RBF ステータスビットはクリアされます。I2COV ステータスビットは、ソフトウェアで「0」を書き込む事によってクリアされます。

表 19-4: データ転送バイト受信時の動作

データバイト 受信時の ステータスビット		I2CxRSR から I2CxRCV への 転送	$\overline{\text{ACK}}$ の生成	SI2CxIF 割り込み の生成 (割り込み有効時)	RBF の セット	I2COV の セット
RBF	I2COV					
0	0	発生する	発生する	発生する	発生する	変化しない
1	0	発生しない	発生しない	発生する	変化しない	発生する
1	1	発生しない	発生しない	発生する	変化しない	発生する
0	1	発生する	発生しない	発生する	発生する	変化しない

凡例: 網掛け部分は、ソフトウェアがオーバーフロー条件を正しくクリアしなかった状態を示します。

19.7.4.2 スレーブ受信中の待機ステート

スレーブ モジュールがデータバイトを受信すると、マスタは即座に次のバイトの送信を開始できます。この場合、スレーブ モジュールを制御するソフトウェアは、先に受信したバイトを 9 SCLx クロック期間内に処理する必要があります。この時間が十分ではない場合、スレーブ ソフトウェアはバス待機期間を要求できます。

STREN ビット (I2CxCON<6>) は、スレーブ受信時のバス待機を有効にします。受信バイトの 9 番目の SCLx クロック立ち上がりエッジで STREN が「1」の場合、スレーブ モジュールは SCLREL ビットをクリアします。SCLREL ビットをクリアすると、スレーブ モジュールは SCLx ラインを LOW にして待機を開始します。マスタとスレーブの SCLx クロックは、19.6.2「マスタクロックの同期」のように同期します。

ソフトウェアは、次の受信を開始できる状態になった時に SCLREL をセットします。これにより、スレーブ モジュールは SCLx ラインをリリースし、マスタがクロックの駆動を再開します。

19.7.4.3 スレーブ受信のメッセージ例

スレーブ メッセージの受信は自動処理されます。スレーブ プロトコルを処理するソフトウェアは、イベントに同期するためにスレーブ割り込みを使用します。

スレーブは有効なアドレスを検出した時に対応する割り込みを生成し、ソフトウェアに後続のメッセージを待機するように伝えます。データ受信時、スレーブは各データバイトを I2CxRCV レジスタに転送するたびに割り込みを生成し、ソフトウェアにバッファを読み出すように伝えます。

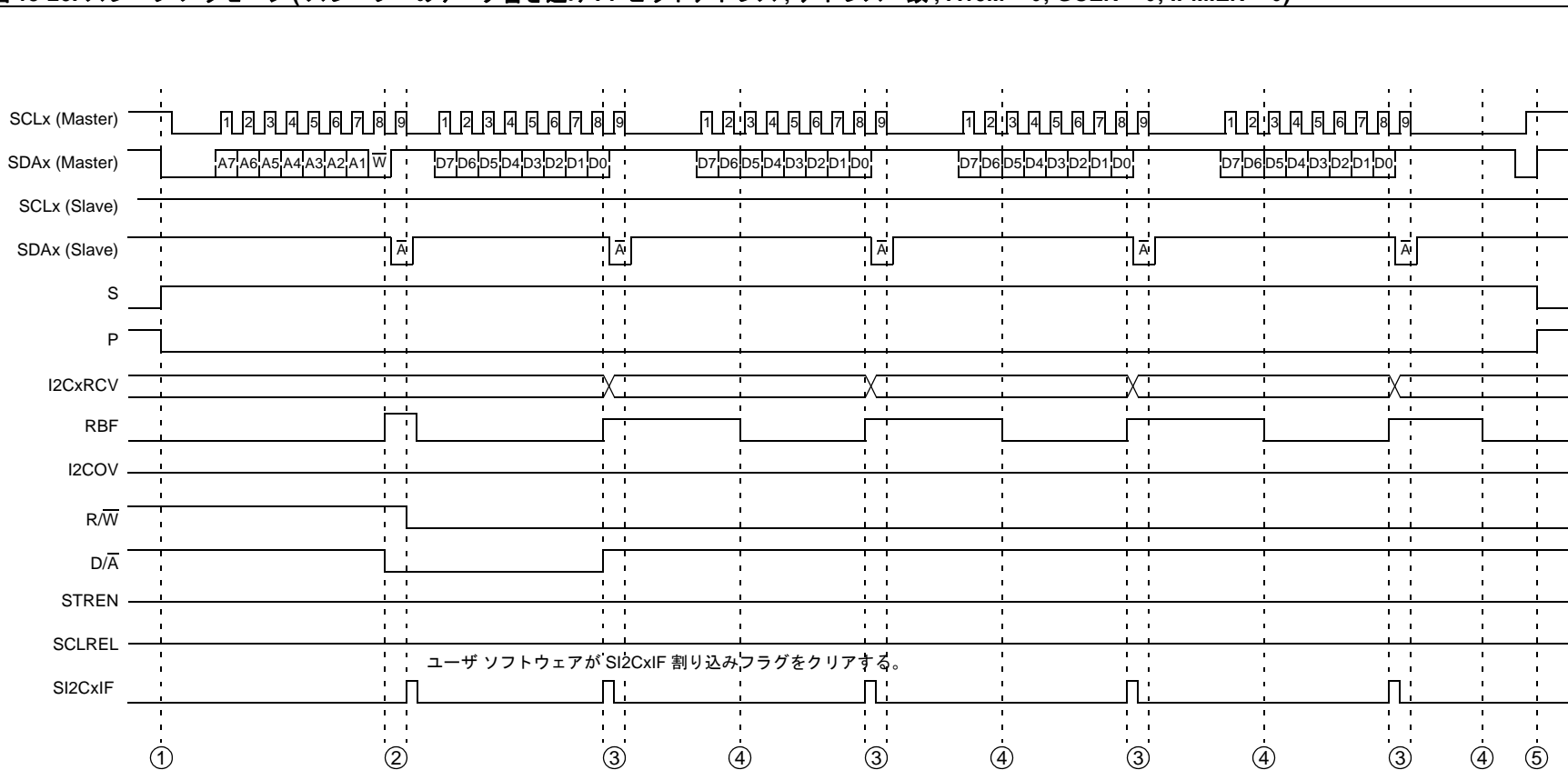
図 19-26 に、単純な受信メッセージを示します。このメッセージは 7 ビットアドレス メッセージであるため、アドレスバイトに対して 1 回の割り込みが発生します。次に 4 つのデータバイトの各々で割り込みが発生します。割り込み発生時に、ソフトウェアは RBF、D/Ā、R/W ステータスビットを監視する事によって、受信したバイトの条件を判別できます。

図 19-27 に、10 ビットアドレスを使用する同様のメッセージを示します。この場合、アドレスには 2 バイトが必要です。

図 19-28 に、ソフトウェアが受信バイトに応答せずにバッファ オーバーランが発生する事例を示します。第 2 バイトの受信時に、モジュールはマスタ送信に対して自動的に NACK を返します。この場合、通常マスタは同じバイトを再送信します。I2COV ステータスビットは、バッファがオーバーランした事を示します。I2CxRCV レジスタバッファは、最初に受信したバイトの内容を保持します。3 回目のバイト受信でもバッファはフルのままであるため、モジュールはマスタに NACK を返します。この後にソフトウェアがバッファを読み出します。バッファの読み出しによって RBF ステータスビットはクリアされますが、I2COV ステータスビットはセットされたままです。ソフトウェアは I2COV ステータスビットをクリアする必要があります。次に受信したバイトは I2CxRCV レジスタバッファに転送され、モジュールは ACK で応答します。

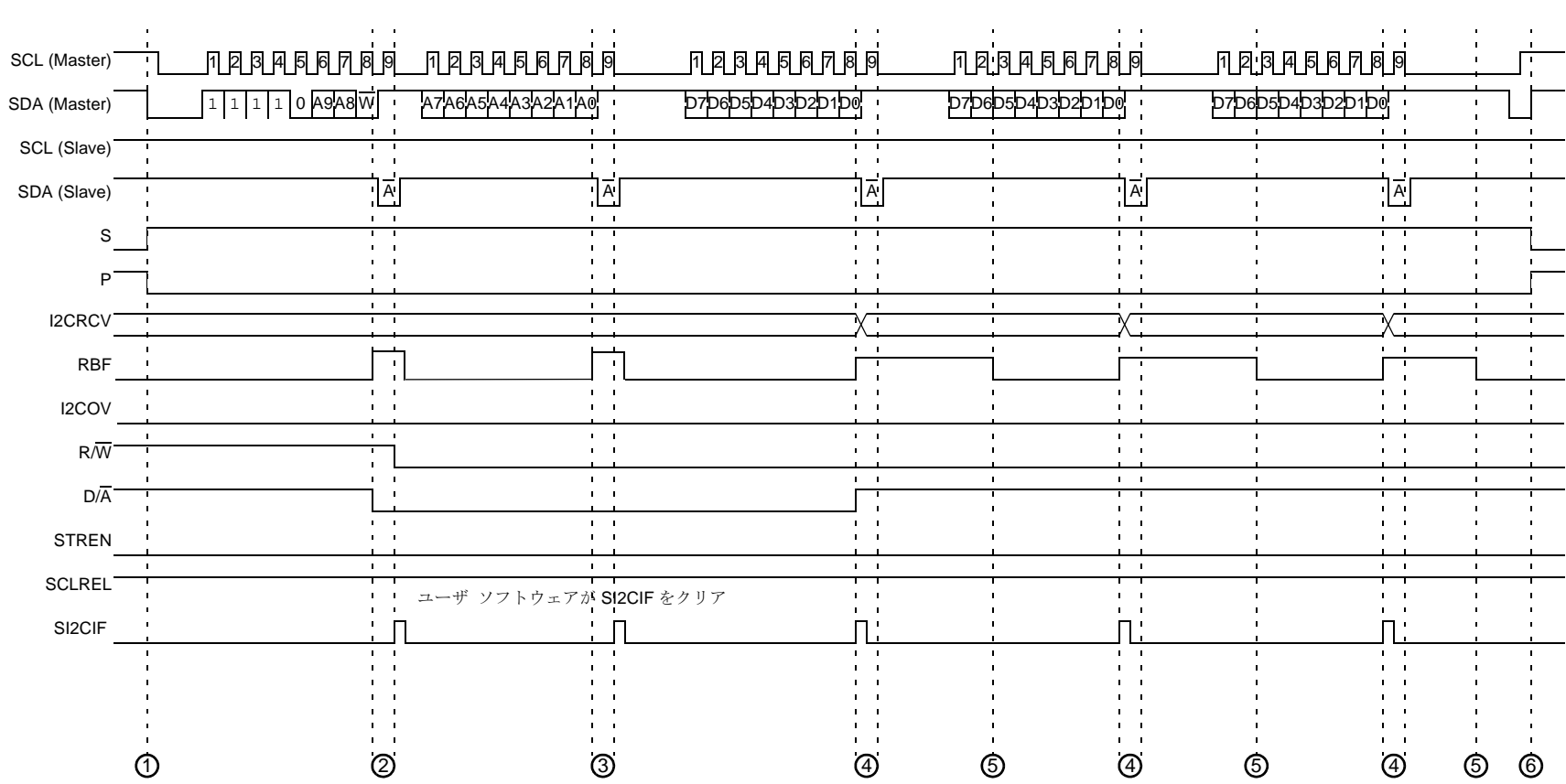
図 19-29 に、データ受信時のクロック ストレッチングを示します。前記の例では、STREN = 0 によってメッセージ受信時のクロック ストレッチングは無効化されていました。この例では、ソフトウェアが STREN をセットしてクロック ストレッチングを有効にします。STREN = 1 の場合、モジュールは各データバイトを受信した後に自動的にクロックをストレッチします。これによりソフトウェアは、バッファからのデータの移動により長い時間を使用できます。9 番目のクロック立ち下がりエッジで RBF = 1 の場合、モジュールは自動的に SCLREL ビットをクリアして SCLx バスラインを LOW にします。2 番目のデータバイトの受信に示すように、9 番目のクロック立ち下がりエッジの前にソフトウェアがバッファを読み出して RBF ステータスビットをクリアする事ができた場合、クロック ストレッチングは発生しません。ソフトウェアはいつでもバスを待機させる事ができます。SCLREL ビットをクリアすると、モジュールはバス上の SCLx クロックが LOW である事を検出した後に SCLx クロックを LOW に駆動します。SCLREL ビットが再びセットされるまで、SCLx ラインは LOW を維持してトランザクションを一時停止します。

図 19-26: スレーブ メッセージ (スレーブ へのデータ書き込み: 7 ビットアドレス; アドレス一致; A10M = 0; GCEN = 0; IPMIEN = 0)



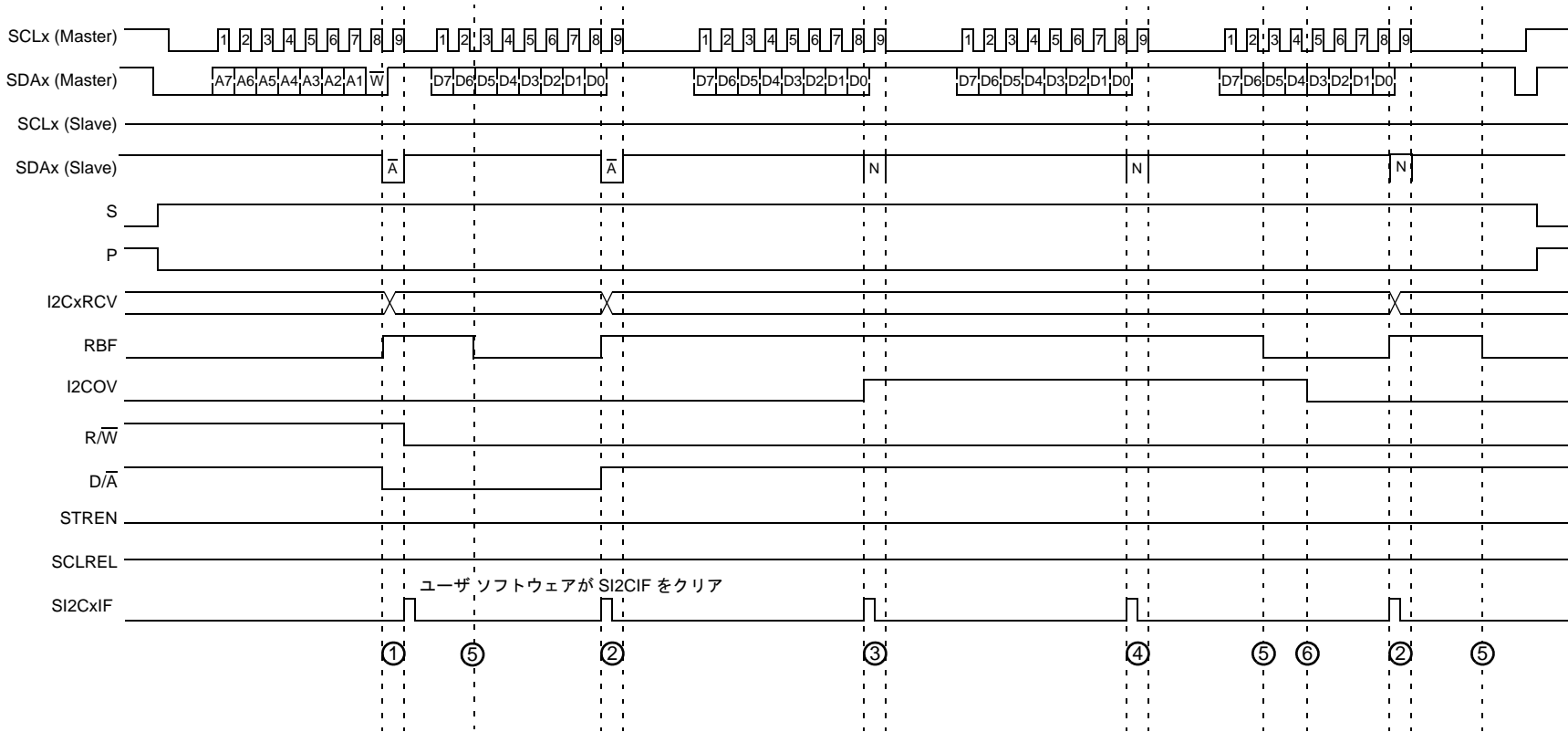
- ① スレーブが START イベントを認識し、S および P ステータスビットを状況に応じてセット / クリアする。
- ② スレーブがアドレスバイトを受信する。アドレスが一致する。スレーブが肯定応答を返して割り込みを生成する。
アドレスバイトを I2CxRCV レジスタに転送する。バッファ オーバーフローを回避するためにユーザ ソフトウェアはこれを読み出す必要がある。
- ③ 次の受信バイトはメッセージデータを格納する。バイトを I2CxRCV レジスタに転送して RBF ステータスビットをセットする。
スレーブが割り込みを生成する。スレーブが肯定応答を返す。
- ④ ソフトウェアが I2CxRCV レジスタを読み出す。RBF ステータスビットをクリアする。
- ⑤ スレーブが STOP イベントを認識し、S および P ステータスビットを状況に応じてセット / クリアする。

図 19-27: スレーブ メッセージ (スレーブへのデータ書き込み:10 ビットアドレス; アドレス一致; A10M = 1; GCEN = 0; IPMIEN = 0)



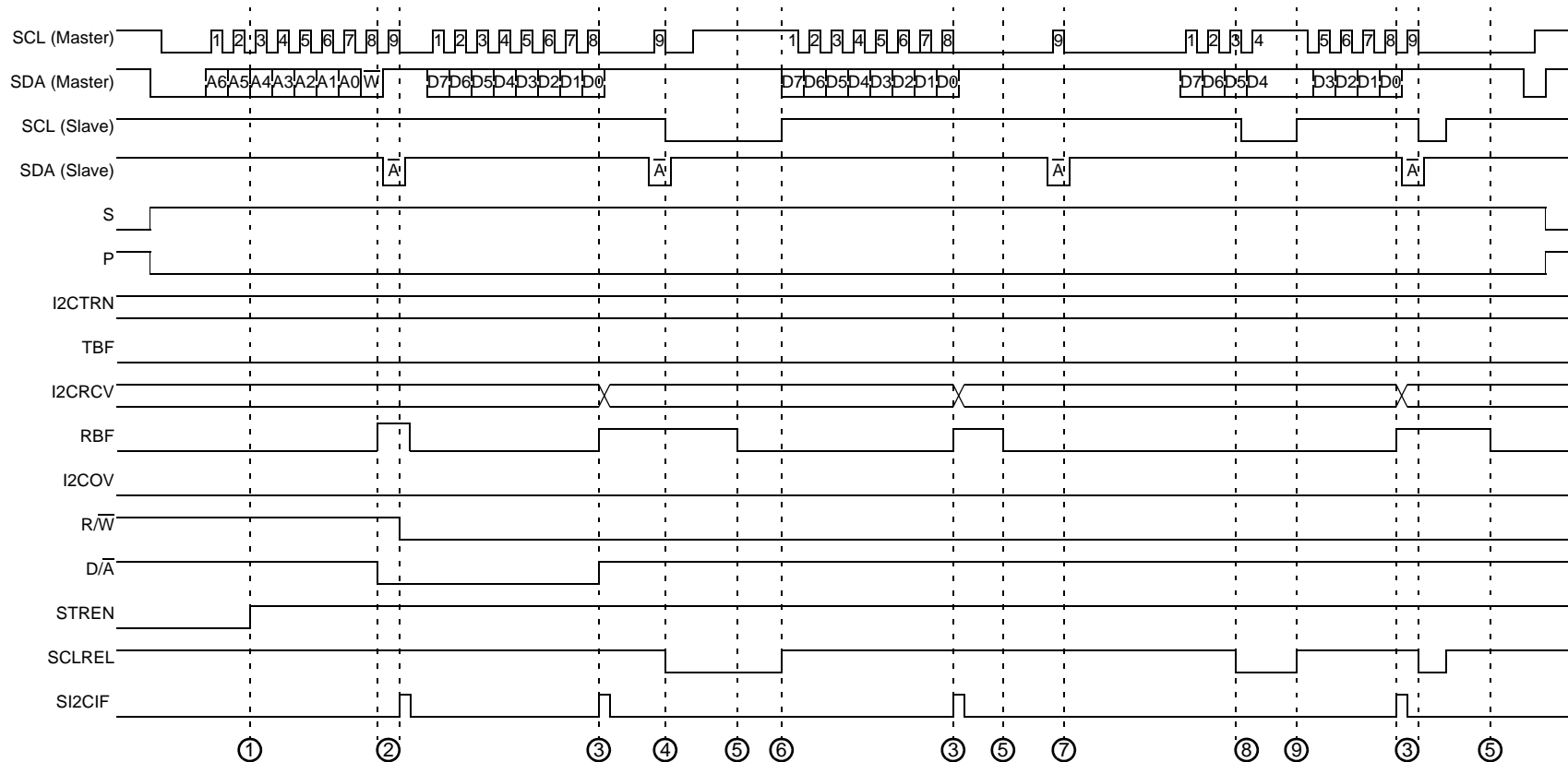
- ① - スレーブが START イベントを認識し、S および P ビットを状況に応じてセット/クリアする。
- ② - スレーブがアドレスバイトを受信する。上位アドレスが一致する。スレーブが肯定応答を返して割り込みを生成する。アドレスバイトを I2CRCV レジスタへ転送する。ユーザ ソフトウェアはこれを読み出してバッファ オーバフローを回避する。
- ③ - スレーブがアドレスバイトを受信する。下位アドレスが一致する。スレーブが肯定応答を返して割り込みを生成する。アドレスバイトを I2CRCV レジスタへ転送する。ユーザ ソフトウェアはこれを読み出してバッファ オーバフローを回避する。
- ④ - 次の受信バイトはメッセージデータを格納する。バイトを I2CRCV レジスタに転送して RBF をセットする。スレーブが肯定応答を返して割り込みを生成する。
- ⑤ - ソフトウェアが I2CRCV レジスタを読み出す。RBF ビットをクリアする。
- ⑥ - スレーブが STOP イベントを認識し、S および P ビットを状況に応じてセット/クリアする。

図 19-28: スレーブ メッセージ (スレーブへのデータ書き込み:7 ビットアドレス; パッファ オーバーラン; A10M = 0; GCEN = 0; IPMIEN = 0)



- ① - スレーブがアドレスバイトを受信する。アドレスが一致する。スレーブが割り込みを生成する。アドレスバイトを I2CxRCV レジスタに転送する。パッファ オーバーフローを回避するためにユーザ ソフトウェアはこれを読み出す必要がある。
- ② - 次の受信バイトはメッセージデータを格納する。バイトを I2CxRCV レジスタに転送して RBF をセットする。スレーブが割り込みを生成する。スレーブが肯定応答を返す。
- ③ - ソフトウェアが I2CxRCV を読み出す前に次のバイトを受信する。I2CxRCV レジスタは変更されない。I2COV オーバーフロー ビットをセットする。スレーブが割り込みを生成する。スレーブが受信に対して NACK を送信する。
- ④ - ソフトウェアが I2CxRCV を読み出す前に次のバイトを受信する。I2CxRV レジスタは変更されない。スレーブが割り込みを生成する。スレーブが受信に対して NACK を送信する。このように NACK を受信した後に次のバイトを送信しないようにマスター状態マシンをプログラミングする必要があります。ステートマシンは次のバイトを送信するのではなく、STOP 条件を送信するか、またはリビートスタート条件を送信してデータの再送信を試みる必要があります。
- ⑤ - ソフトウェアが I2CxRCV レジスタを読み出す。RBF ビットをクリアする。
- ⑥ - ソフトウェアが I2COV ビットをクリアする。モジュールが STOP/リビートスタート ビットを検出するまで正常な受信はできない。これらの条件のいずれかを検出するまで、次の送信を正常に受信するが、再び NACK を返して I2COV をセットする。

図 19-29: スレーブ メッセージ (スレーブへのデータ書き込み: 7 ビットアドレス; クロック ストレッチング; A10M = 0; GCEN = 0; IPMIEN = 0)



- ① - ソフトウェアが STREN ビットをセットしてクロック ストレッチングを有効にする。
- ② - スレーブがアドレスバイトを受信する。
ユーザ ソフトウェアはこれを読み出してバフ オーバーフローを回避する。
- ③ - 次の受信バイトはメッセージデータを格納する。
バイトを I2CRCV レジスタに転送して RBF をセットする。
- ④ - 9 番目のクロックで RBF = 1 であるためクロック ストレッチングを自動的に開始。
スレーブが SCLREL ビットをクリアする。
スレーブが SCL ラインを LOW にしてクロックをストレッチングする。
- ⑤ - ソフトウェアが I2CRCV レジスタを読み出す。RBF ビットをクリアする。
- ⑥ - ソフトウェアが SCLREL ビットをセットしてクロックをリリースする。
- ⑦ - スレーブは SCLREL をクリアしない (この時点では RBF = 0 であるため)
- ⑧ - ソフトウェアは SCLREL をクリアしてクロックをストレッチングできる。
モジュールは SCL を LOW に駆動する前に SCL LOW を検出する必要がある。
- ⑨ - ソフトウェアは SCLREL をセットしてクロック ストレッチングを解除できる。

19.7.5 マスタデバイスヘータを送信する

受信したデバイスアドレス バイトの R/W ステータスビットが「1」の時にアドレスが一致すると、R/W ステータスビット (I2CxSTAT<2>) がセットされます。この時点でマスタデバイスは、スレーブが1バイトのデータを送信して応答する事を期待します。システム プロトコルがバイトの内容を定義し、当該スレーブだけがデータを送信します。

Note: IPMIEN = 1 (IPMI モード) の場合、I²C モジュールは R/W ビットが「0」であるとみなします。従ってスレーブ送信は無効化されます。R/W ビットが「1」の場合、I²C モジュールは割り込みをトリガしますが、この割り込みは無視され (I²C 割り込みフラグ がクリアされ)、I²C スレーブ送信イベントは中止されます。

アドレス検出から割り込みが発生すると、ソフトウェアは 1 バイトを I2CxTRN レジスタに書き込んでデータ送信を開始できます。

スレーブ モジュールは TBF ステータスビットをセットします。SCLx 入力の各立ち下がりエッジで、8 つのデータビットがシフトアウトされます。これにより、SCLx の HIGH 時間中の SDAx 信号が有効データとしてサンプリングされます。8 ビットの全てをシフトアウトすると、TBF ステータスビットがクリアされます。

スレーブ モジュールは、9 番目の SCLx クロック立ち上がりエッジで、マスタレシーバからの肯定応答を検出します。

SDAx ラインが肯定応答 (ACK) を示す LOW であれば、マスタは後続のデータを待機し、メッセージは終了しません。モジュールはスレーブ割り込みを生成して ACKSTAT ステータスビットをチェックする事によって、次のデータが要求されているかどうかを判別できます。

スレーブ割り込みは SCLx クロックの 9 番目の立ち下がりエッジで生成されます。ソフトウェアは I2CxSTAT レジスタのステータスをチェックし、SI2CxIF 割り込みフラグをクリアする必要があります。

SDAx ラインが否定応答 (NACK) を示す HIGH であれば、データ転送は終了します。スレーブ モジュールはリセットして割り込みを生成し、次の START ビットの検出を待機します。

19.7.5.1 スレーブ送信中の待機ステート

スレーブがメッセージを送信する場合、マスタはスレーブが R/W = 1 を持つ有効アドレスを検出した直後にデータを返送する事を期待します。このためスレーブ モジュールは、データを返送する時に必ずバス待機を自動生成します。

自動待機は、有効なデバイスアドレス バイトまたはマスタが後続の送信データを期待する事を示すために肯定応答を返した送信バイトの 9 番目の SCLx クロック立ち下がりエッジで発生します。

スレーブ モジュールは SCLREL ビットをクリアします。SCLREL ビットをクリアすると、スレーブ モジュールは SCLx ラインを LOW にして待機を開始します。マスタとスレーブの SCLx クロックは、19.6.2 「マスタクロックの同期」のように同期します。

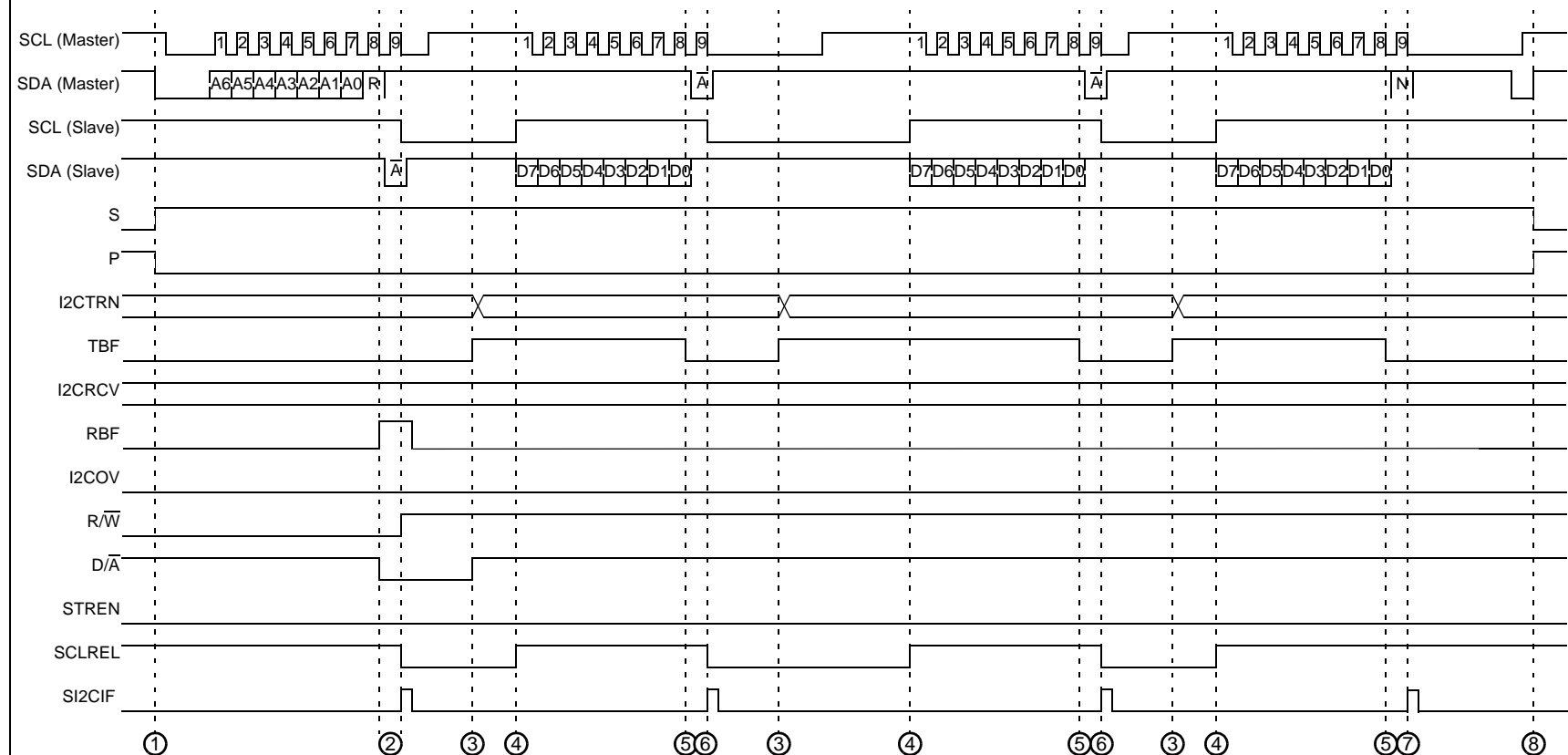
ソフトウェアは I2CxTRN レジスタにデータを書き込み、送信の再開が可能になると SCLREL をセットします。これによりスレーブ モジュールは SCLx ラインをリリースし、マスタがクロックの駆動を再開します。

19.7.5.2 スレーブ送信のメッセージ例

7 ビットアドレス メッセージのスレーブ送信を図 19-30 に示します。アドレスが一致し、そのアドレスの R/W ステータスビットがスレーブ送信を指定している場合、モジュールは SCLREL ビットをクリアする事によって自動的にクロック ストレッチングを開始し、割り込みを生成して応答バイトが必要である事を示します。ソフトウェアは応答バイトを I2CxTRN レジスタに書き込みます。送信が完了すると、マスタは肯定応答を返します。マスタが ACK を返して次のデータを期待している事を示した場合、モジュールは再び SCLREL ビットをクリアして割り込みを生成します。マスタが NACK を返して次のデータを要求していない事を示した場合、モジュールはクロックをストレッチングせずに割り込みを生成します。

10 ビットアドレス メッセージのスレーブ送信では、スレーブは最初に 10 ビットアドレス モードである事を認識する必要があります。マスタはアドレス用に 2 バイトを送信するため、先頭バイトの R/W ステータスビットは必ず「書き込み」を指定します。メッセージを「読み出し」に変更するために、マスタはリピータースタートを送信してから、R/W ステータスビットを「読み出し」に設定したアドレス先頭バイトを再度送信する必要があります。この時点でスレーブ送信が開始されます (図 19-31 参照)。

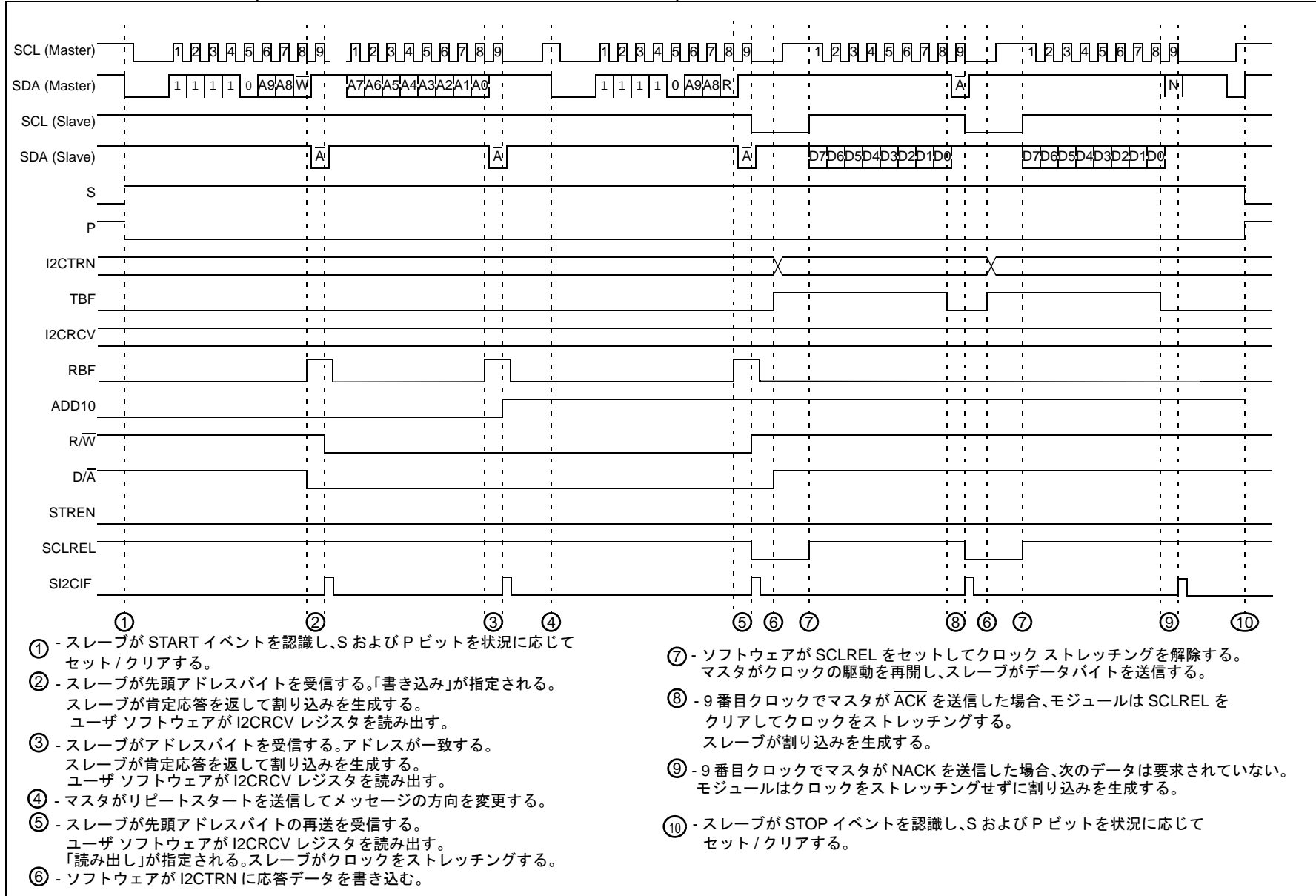
図 19-30: スレーブ メッセージ (スレーブからのデータ読み出し: 7 ビットアドレス)



- ① - スレーブが START イベントを認識し、S および P ビットを状況に応じてセット/クリアする。
- ② - スレーブがアドレスバイトを受信する。アドレスが一致する。スレーブが割り込みを生成する。アドレスバイトを I2CRCV レジスタへ転送する。ユーザ ソフトウェアがこれを読み出してバッファ オバフローを回避する。R/W = 1 がスレーブからの読み出しを指定する。SCLREL = 0 によってマスタクロックをストレッチングする。
- ③ - ソフトウェアが I2CTRN に応答データを書き込む。TBF = 1 がバッファフルを示す。I2CTRN への書き込みによって D/A がセットされ、データバイトである事を示す。
- ④ - ソフトウェアが SCLREL をセットしてクロック ストレッチングを解除する。マスタがクロックの駆動を再開し、スレーブがデータバイトを送信する。

- ⑤ - 最終ビットの後にモジュールが TBF ビットをクリアし、次のバイト用にバッファが空いている事を示す。
- ⑥ - 9 番目クロックでマスタが ACK を送信した場合、モジュールは SCLREL をクリアしてクロックをストレッチングせず割り込みを生成する。
- ⑦ - 9 番目クロックでマスタが NACK を送信した場合、次のデータは要求されていない。モジュールはクロックをストレッチングせずに割り込みを生成する。
- ⑧ - スレーブが STOP イベントを認識し、S および P ビットを状況に応じてセット/クリアする。

図 19-31: スレーブ メッセージ (スレーブからのデータ読み出し:10 ビットアドレス)



19.8 I²C バスの接続に関する注意事項

I²C バスはワイヤード AND 接続であるため、バスにはプルアップ抵抗が必要です (図 19-32 内の R_P 参照)。直列抵抗 (図中の R_S) は必ずしも必要ではありませんが、ESD 耐性を改善します。R_P と R_S の抵抗値は下記のパラメータによって決まります。

- 電源電圧
- バス静電容量
- 接続するデバイス数 (入力電流 + リーク電流)
- 入力レベルの選択 (I²C または SMBus)

デバイスは R_P に対抗してバスを LOW にする必要があるため、R_P による引き込み電流はデバイス出力段の I/O ピン最小シンク電流 (I_{OL} = 6.6 mA at V_{OLMAX} = 0.4 V) を上回る必要があります。例として、V_{DD} = 3 V + 10% の場合を下式に示します。

式 19-2:

$$R_{P\min} = (V_{DD\max} - V_{OL\max})/I_{OL} = (3.3\text{V} - 0.6\text{V})/8.5\text{ mA} = 439\Omega$$

最小立ち上がり時間の仕様値は、400 kHz システムで 300 ns、100 kHz システムで 1000 ns です。R_P は、総静電容量 (C_B) に対して、バスを最大立ち上がり時間 300 ns で (V_{DD} - 0.7 V) までプルアップする必要があるため、プルアップの最大抵抗値 (R_{PMAX}) を下式より小さくする必要があります。

式 19-3:

$$-tR/(C_B * (\ln(1 - (V_{DD\max} - V_{IL\max})))) = -300\text{ ns}/(100\text{pF} * \ln(1 - (0.99 - 3.3))), \text{ または } 2.5\text{k}\Omega$$

R_S の最大値は、LOW レベルに対する要求ノイズマージンによって決まります。R_S は、(デバイス V_{OL} + R_S 両端電位差) が最大 V_{IL} を超えられるほど十分には電圧を降下できません (下式)。

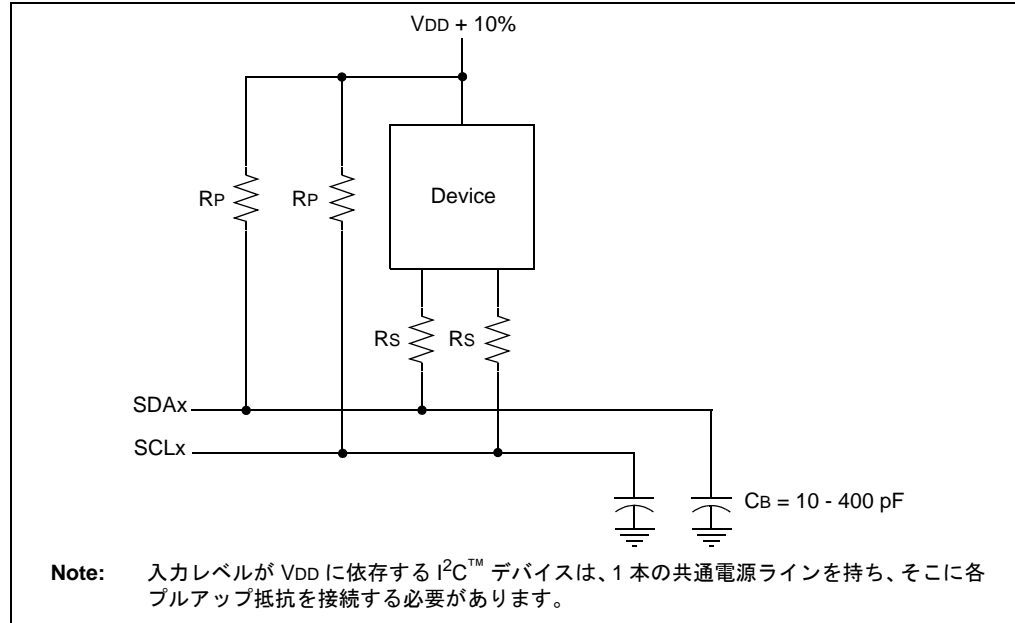
式 19-4:

$$R_{S\max} = (V_{IL\max} - V_{OL\max})/I_{OL\max} = (0.3\text{ V}_{DD} - 0.4)/6.6\text{mA} = 89\Omega$$

適正動作を保証するために、SCLx クロック入力には最小 HIGH および LOW 時間が指定されています。I²C バスの HIGH および LOW 時間の仕様値と、I²C モジュールの要件については、各データシートの「電気的特性」を参照してください。

Note: I²C ピン (SDA および SCL ピン) は 5 V を許容しません。しかし、これらの I²C ピンに多重化された他のピン機能 (I/O ポート等) は 5 V を許容する場合があります。詳細は各デバイスのデータシートを参照してください。

図 19-32: I²C™ バスのデバイス コンフィグレーション例



19.8.1 内蔵信号コンディショニング

SCLxおよびSDAxピンは入力グリッチフィルタを備えます。I²Cバスは100 kHzおよび400 kHzシステムの両方でこのフィルタを必要とします。

400 kHz バスで動作する場合、I²C バス仕様はデバイスピン出力のスルーレート制御を要求します。このスルーレート制御はデバイスに内蔵されています。DISSLW ビット (I2CxCON<9>) をクリアすると、スルーレート制御が有効化されます。他のバス速度の場合、I²C バス仕様はスルーレート制御を要求しないため、DISSLW ビットをセットする必要があります。

I²C バスのシステム形態によって V_{ILMAX} と V_{IHMIN} の要求入力レベルが異なります。通常の I²C システムでは、 V_{ILMAX} は $0.3 V_{DD}$ 、 V_{IHMIN} は $0.7 V_{DD}$ です。これに対して SMBus (System Management Bus) システムの場合、 V_{ILMAX} は $0.8 V$ に設定され、 V_{IHMIN} は $2.1 V$ に設定されます。

SMEN ビット (I2CxCON<8>) は入力レベルを制御します。SMEN ビットをセットする (= 1) と、入力レベルは SMBus 仕様向けに変更されます。

19.9 PWRSAV 命令実行時のモジュール動作

19.9.1 スレーブモード中のスリープモード

I²C モジュールは、有効なアドレス一致を検出した時にスリープモードからウェイクアップできます。全てのビットシフトは、I²C マスタが生成する外部 SCL 信号を基準として行われるため、スリープモード中でも送受信を続ける事ができます。

Note: スレーブ I²C の挙動に従い、スレーブ割り込みはアドレス一致時にのみ生成されます。このため、I²C スレーブがスリープモード中にマスタからのメッセージを受信すると、受信したアドレスの一致検出に必要なクロックはマスタから供給されます。割り込みがあらかじめ有効化され、ISR が定義済みである場合にのみ、アドレス一致時に割り込みが発生してデバイスがスリープからウェイクアップします。

19.9.2 マスタモード中のスリープモード

マスタ送信中にスリープへの移行が発生し、クロックが停止した時にステートマシンが送信の途中であった場合、モジュールの挙動は未確定です。同様にマスタ受信中にスリープへの移行が発生した場合も、モジュールの挙動は未確定です。マスタモードで動作中のトランスミッタとレシーバはスリープモード時に停止します。レジスタの内容は、スリープモードへの移行またはスリープモードからの復帰によって影響を受けません。送信または受信動作の途中でのスリープモードへの移行を自動的に回避する方法は存在しません。未確定のモジュール挙動を回避するために、ユーザ ソフトウェアはスリープモードへの移行を I²C の動作に同期させる必要があります。

19.9.3 デバイスがアイドルモードに移行した場合

デバイスは PWRSAV 1 命令を実行する事によってアイドルモードに移行します。アイドルモード時に、モジュールは I2CSIDL ビット (I2CxCON<13>) の状態に応じて省電力モードに移行します。I2CSIDL = 1 の場合、モジュールは省電力モード (スリープモード時と同様の挙動) に移行します。I2CSIDL = 0 の場合、モジュールは省電力モードへ移行せず、通常の動作を続けます。

19.10 周辺モジュール ディセーブル (PMD) レジスタ

周辺モジュール ディセーブル (PMDx) レジスタを使用すると、I²C モジュールへのクロック供給源を全て停止する事によってモジュールを無効化できます。対応する PMDx 制御ビットで周辺モジュールを無効化すると、そのモジュールは最小消費電力状態となります。その周辺モジュールに関連する制御およびステータス レジスタも無効化されるため、それらのレジスタへの書き込みは効果を持たず、読み出し値は無効です。I²C モジュールは、PMDx レジスタの I2CxMD ビットがクリアされている場合にのみ有効化されます。

19.11 リセットの影響

リセットは I²C モジュールを無効化し、実行中または保留中の全てのメッセージ動作を終了します。I2CxCON および I2CxSTAT レジスタのリセット状態については、それらのレジスタの定義を参照してください。

Note: ここでの「アイドル」とは CPU の省電力状態を意味します。I²C モジュールがバス上でデータを転送中ではないという意味の「アイドル」とは異なりますので、注意してください。

19.12 レジスタマップ

dsPIC33F/PIC24H I²C モジュールに関連するレジスタの要約を表 19-5 に示します。

表 19-5: I²Cx 関連のレジスタマップ

レジスタ名	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	全 リセット
I2CxRCV	—	—	—	—	—	—	—	—	受信レジスタ								0000
I2CxTRN	—	—	—	—	—	—	—	—	送信レジスタ								00FF
I2CxBRG	—	—	—	—	—	—	—	baud レート ジェネレータ								0000	
I2CxCON	I2CEN	—	I2CSIDL	SCLREL	IPMIEN	A10M	DISSLW	SMEN	GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	1000
I2CxSTAT	ACKSTAT	TRSTAT	—	—	—	BCL	GCSTAT	ADD10	IWCOL	I2COV	D/A	P	S	R/W	RBF	TBF	0000
I2CxADD	—	—	—	—	—	—	アドレスレジスタ										0000
I2CxMSK	—	—	—	—	—	—	アドレスマスク										0000

凡例: — = 未実装、「0」として読み出し。リセット値は 16 進数で表記

19.13 設計のヒント

- 質問 1:** バスマスタとして動作中にデータを送信するとスレーブ割り込みと受信割り込みが同時に発生し続けます。なぜですか。
- 回答:** マスタとスレーブの回路は独立しています。スレーブ モジュールはマスタが送信したバスからイベントを受信します。
- 質問 2:** スレーブとして動作中にデータを I2CxTRN レジスタに書き込みましたが、データが送信されません。原因は何ですか。
- 回答:** スレーブは送信の準備中に自動的に待機状態に移行します。SCLREL ビットをセットして I²C クロックをリリースする必要があります。
- 質問 3:** マスタ モジュールの状態を知る方法を教えてください。
- 回答:** SEN、RSEN、PEN、RCEN、ACKEN、TRSTAT ビットからマスタ モジュールの状態を知ることができます。全てのビットが「0」であれば、モジュールはアイドル中です。
- 質問 4:** スレーブとして動作中に STREN = 0 の状態でバイトを受信します。次のバイトを受信する前に受信済みバイトをソフトウェアが処理できない場合、ソフトウェアはどのように動作すべきですか。
- 回答:** STREN が「0」であったため、モジュールは受信バイトで自動待機を生成しませんでした。しかし、ソフトウェアはメッセージの途中でいつでも STREN をセットして SCLREL をクリアできます。これにより、次の機会で待機を生成して SCLx クロックに同期する事ができます。
- 質問 5:** I²C モジュールをマルチマスタ システムで使用しています。メッセージの送信を試みると、メッセージが破損します。原因は何ですか。
- 回答:** マルチマスタ システムでは、他のマスタがバス衝突を生じる可能性があります。マスタの割り込みサービスルーチンで BCL ステータスビットをチェックして、動作が衝突の発生なく完了したかどうかを確認してください。衝突を検出した場合、メッセージを最初から再送信する必要があります。
- 質問 6:** I²C モジュールをマルチマスタ システムで使用しています。メッセージを開始できるかどうかは、どのように判断すれば良いのですか。
- 回答:** S ステータスビットをチェックしてください。S = 0 であれば、バスはアイドル中です。
- 質問 7:** バスに START 条件を送信してから I2CxTRN レジスタにデータを書き込んでバイトの送信を試みましたが、そのバイトは送信されませんでした。原因は何ですか。
- 回答:** I²C バスでは、1 つのイベントが完了してから次のイベントを開始する必要があります。この場合、I2CxTRN レジスタに書き込む前に、SEN ビットをポーリングする事によって START イベントの完了を検出するか、マスタ I²C 割り込みを待機する必要があります。

19.14 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 ノートの一覧を以下に記載します。一部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。I²C™ (Inter-Integrated Circuit™) 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
I ² C™ マルチマスタ環境における SSP モジュールの使用	AN578
スレーブ I ² C™ 通信向け PIC® デバイスの SSP および MSSP モジュールの使用	AN734
マスタ I ² C™ 通信向け PICmicro® MSSP モジュールの使用	AN735
環境監視向け I ² C™ ネットワーク プロトコル	AN736

Note: dsPIC33F/PIC24H デバイス 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。

19.15 改訂履歴

リビジョン A (2007 年 2 月)

本書の初版

リビジョン B (2008 年 8 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 19-2 内の ACKSTAT ビット (I2CxSTAT<15>) と D/A ビット (I2CxSTAT<5>) のビット定義を更新
- 式 19-1 内の I2CBRG 計算式の分母を 1,111,111 から 10,000,000 に変更
- 表 19-1 内の I²C クロックレート値を更新、表の注釈を削除、表の直後に全般的な注釈を追加
- 19.3 「制御 / ステータス レジスタ」の末尾 2 段落を更新し、I2CxRSR レジスタ内の一致アドレスバイトの I2CxRCV レジスタへのシフトについて明確化
- 19.4 「I²C 動作の有効化」を更新し、マスタ機能は SEN ビットがセットされた状態で I2CxTRN レジスタにデータが書き込まれた時に有効化される事を明確化
- スレーブモードでの NACK ステータスを明確化するために下記のセクションを更新
 - 19.4.2 「I²C 割り込み」
 - 19.7.5 「マスタデバイスヘータを送信する」
 - 図 19-28 ~ 図 19-31
- IPMIEN ビットが「Intelligent Peripheral Management Interface Enable bit」として誤記されていたため、全ての箇所を「Intelligent Platform Management Interface bit」に訂正
- 19.9.2 「マスタモード中のスリープモード」を更新し、送信中にスリープモードに移行した時に発生する挙動について明確化
- 図 19-26 ~ 図 19-31 内のスレーブ メッセージ RBF ステータスビットに関する情報を更新
- 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 (2011 年 7 月)

このリビジョンでの変更内容は以下の通りです。

- 本書を PIC24H ファミリ デバイスを含めた内容に更新
- I2CxCON レジスタ (レジスタ 19-1 参照) に Note 1 を追加
- I2CxSTAT レジスタ (レジスタ 19-2 参照) 内の P および S ビットの HSC (ハードウェア セット / クリア) に関するビットステータスを更新
- BRG リロード値の計算式 (式 19-1 参照) を更新
- 19.2 「I²C バスの特性」と 19.8 「I²C バスの接続に関する注意事項」に、SDA および SCL ピンに関する網掛け注釈を追加
- 19.5 「シングルマスタ環境でマスタとして通信する」と 19.6 「マルチマスタ環境でマスタとして通信する」の第一段落に網掛け注釈を追加
- I²C クロックレート表 (表 19-1) 内の「Actual F_{SCL}」列を削除、「PGD」列を追加、10 進値と 16 進値を更新
- 19.7.3.6 「全てのアドレスを受信する ((IPMI 動作))」の第 2 段落の末尾に 1 文章を追加し、F 網掛け注釈を追加
- 19.7.5 「マスタデバイスヘータを送信する」の第 1 段落に 1 文章を追加
- 19.7.5.2 「スレーブ送信のメッセージ例」の第 1 段落の末尾の文章を更新し、割り込みが発生する事を明確化
- 19.9.1 「スレーブモード中のスリープモード」に網掛け注釈ボックスを追加

ISBN: 978-1-60932-868-9