

SPI インターフェーススタンドアロン CAN コントローラ

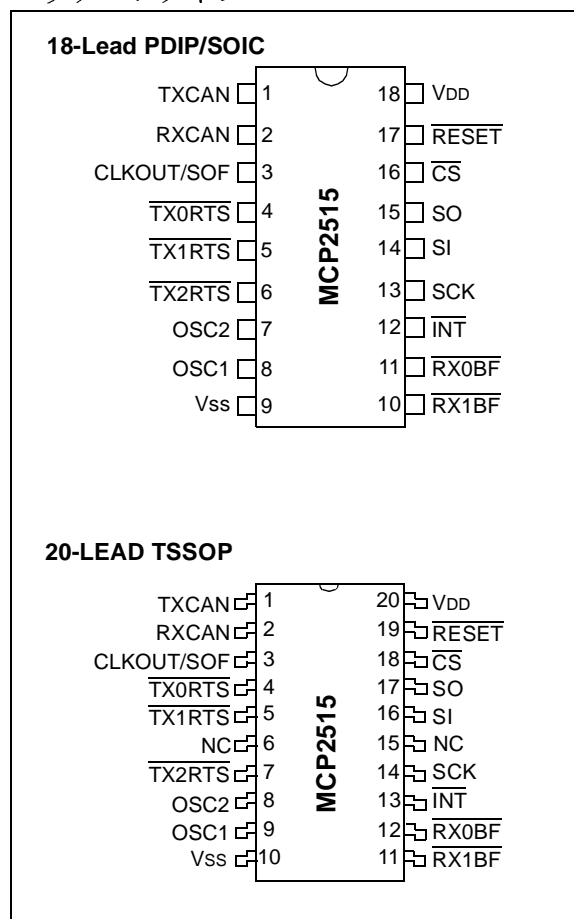
特徴

- 1 Mb/s の CAN V2.0B を内蔵:
 - 0-8 バイト長のデータフィールド
 - 標準および拡張のデータ、リモートフレーム
- 受信バッファ、マスクとフィルタ付き:
 - 2 個の受信バッファで、優先メッセージ格納
 - 6 組の 29 ビットフィルタ
 - 2 組の 29 ビットマスク
- 最初の 2 データバイトにデータバイトフィルタを適用 (標準データフレーム対応)
- 3 個の送信バッファで、優先、アボート機能付き
- 高速度 SPI™ インターフェース (10 MHz):
 - SPI モードは 0,0 または 1,1
- ワンショットモードによりメッセージ送信を確実に 1 回だけとします。
- 設定可能な分周比のクロック入力ピン:
 - 他のデバイスのクロック源として使用可能
- Start-of-Frame (SOF) 信号により SOF 信号のモニタが可能:
 - タイムスロットベースのプロトコルか、あるいは早期にバス悪化を検出しバス診断を行うのに使えます。
- 選べるイネーブル付きの割り込み出力ピン
- バッファフル出力ピンは下記のように構成可能:
 - 各受信バッファ対応の割り込み出力
 - 汎用目的の出力
- Request-to-Send (RTS) 入力ピンは個別に構成可能:
 - 各送信バッファ対応の送信要求の制御ピン
 - 汎用目的の入力
- 低消費電力 CMOS テクノロジ:
 - 動作電圧 2.7V ~ 5.5V
 - 動作電流 5 mA (typical)
 - スタンバイ電流 1 μ A (typical) (スリープモード)
- 下記の温度範囲をサポート:
 - 工業品 (I): -40°C ~ +85°C
 - 拡張品 (E): -40°C ~ +125°C

概要

マイクロチップテクノロジー社 MCP2515 は、CAN 仕様、バージョン 2.0B を満足するスタンドアロンのコントロールエリアネットワーク (CAN) コントローラです。そして標準と拡張データおよびリモートフレームの送受信を行うことができます。MCP2515 は、2 個の受信マスクと 6 個の受信フィルタを持っていて、不要なメッセージをふるい落とすのに使われます。これによりホスト MCU のオーバーヘッド負荷を減らします。MCP2515 は工業標準のシリアル周辺機器インターフェース (SPI) によって、マイコン (MCU) とインターフェースします。

パッケージタイプ



MCP2515

NOTES:

1.0 デバイス概要

MCP2515 はスタンドアロン CAN コントローラで、CAN バスとインターフェースする必要のある応用を簡単にするために開発されました。MCP2515 の簡単な内部ブロック図を図 1-1 に示します。デバイスは 3 つの主要ブロックで構成されています。:

1. CAN モジュールには、CAN プロトコルエンジン、マスク、フィルタ、送信、受信バッファが含まれています。
2. 制御ロジックとレジスタは、デバイスの構成と動作モードを決めるのに使われます。
3. SPI プロトコルブロック

このデバイスを使った例題システムの実装を図 1-2 に示します。

1.1 CAN モジュール

CANモジュールは、CANバスでメッセージを送受信するためのすべての機能を処理します。メッセージは、まず最初に適切なメッセージバッファと制御レジスタにロードされることで送信されます。SPI インターフェース経由で制御レジスタビットを制御するか、または送信イネーブルピンを使うことで送信が起動されます。状態やエラーは適切なレジスタを読み出すことでチェックすることが可能になります。CAN バス上に検出されたすべてのメッセージは、エラーチェックされてから、2つの受信レジスタのひとつに取り込むかどうかを判定するため、ユーザー定義のフィルタとの照合をします。

1.2 制御ロジック

制御ロジックブロックは、情報と制御を受け渡すために他のブロックとインターフェースすることで、MCP2515 のセットアップと動作を制御します。

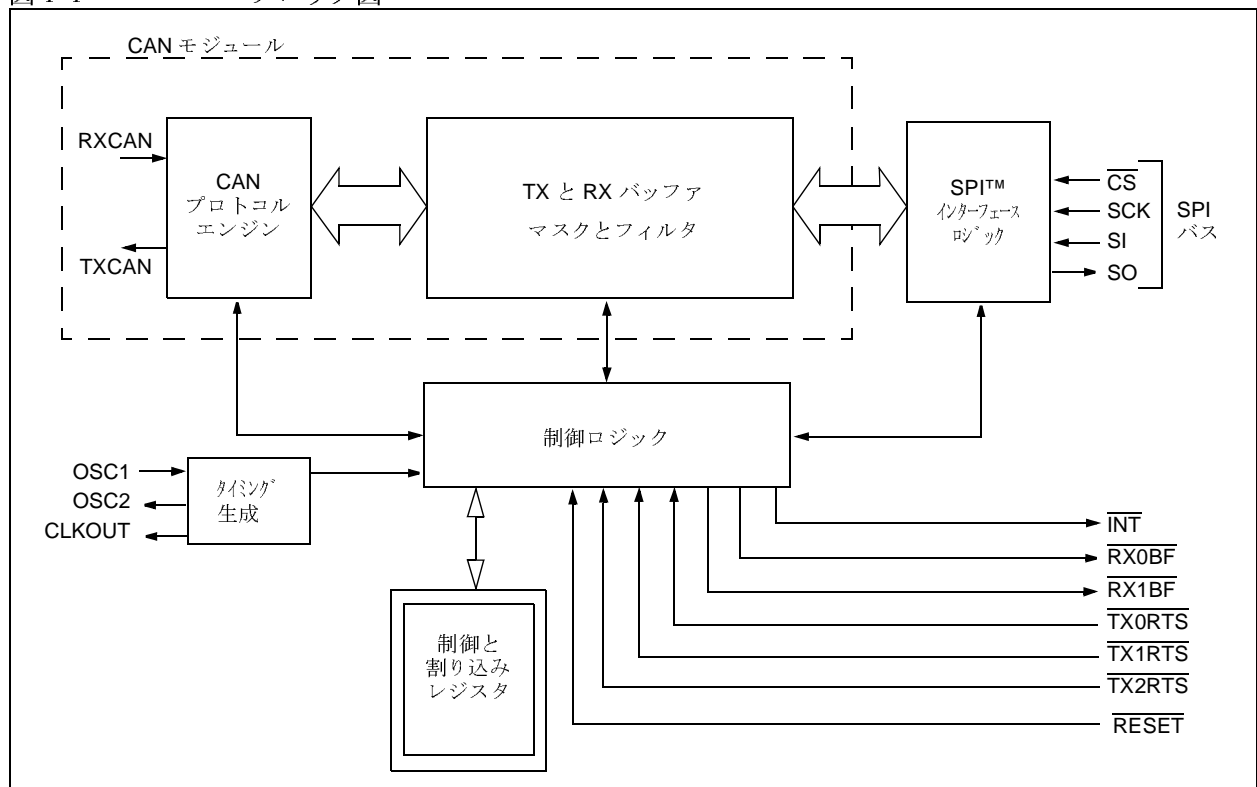
割り込みピンは、大きなシステム柔軟性を提供します。各受信レジスタには多目的の割り込みピン（専用割り込みピンと同様に）が用意されています。これにより、有効メッセージが受信され、受信バッファのひとつに格納されたことを表します。専用ピンはオプションで使われます。汎用割り込みピンは、(SPI インターフェース経由でアクセスされる) ステータスレジスタと同様に、有効メッセージが受信されたことを決めるために使うことができます。

さらに、3 個の送信レジスタのいずれかにロードされたメッセージの即時送信を起動するために 3 ピンが用意されています。これらのピンは、メッセージ送信を起動するためのオプションとしても使われます。もともとこの起動は SPI インターフェース経由で制御レジスタをアクセスすることも行うことができます。

1.3 SPI プロトコルブロック

デバイスとMCUのインターフェースはSPIインターフェースとなっています。標準のSPIリードライトコマンドに専用SPIコマンドを加えて使うことで、すべてのレジスタの読み書きをすることができます。

図 1-1: ブロック図



MCP2515

図 1-2: 例題システムの実装

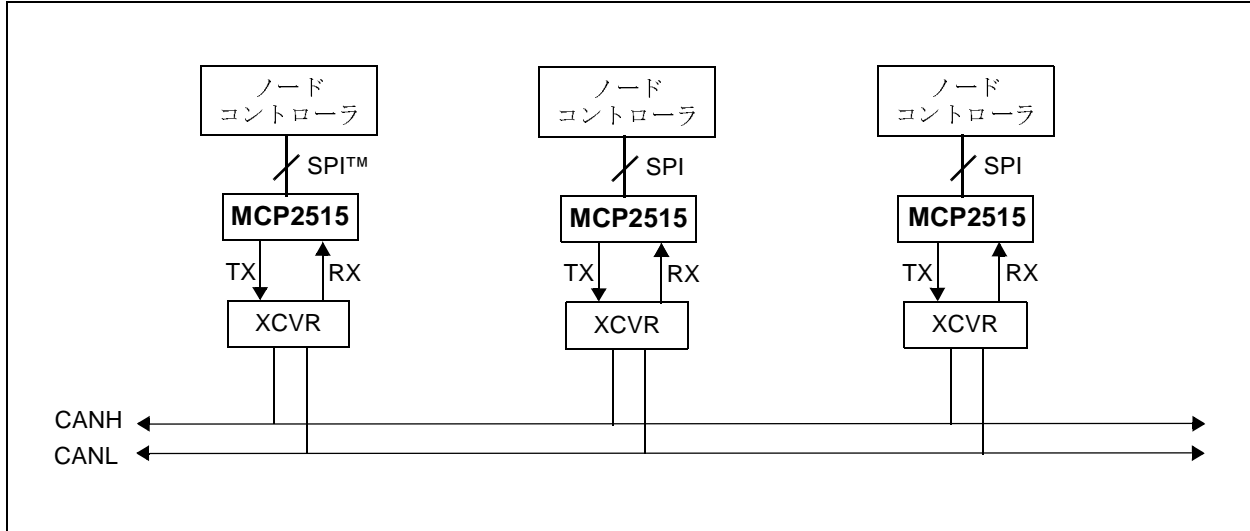


表 1-1: ピンの説明

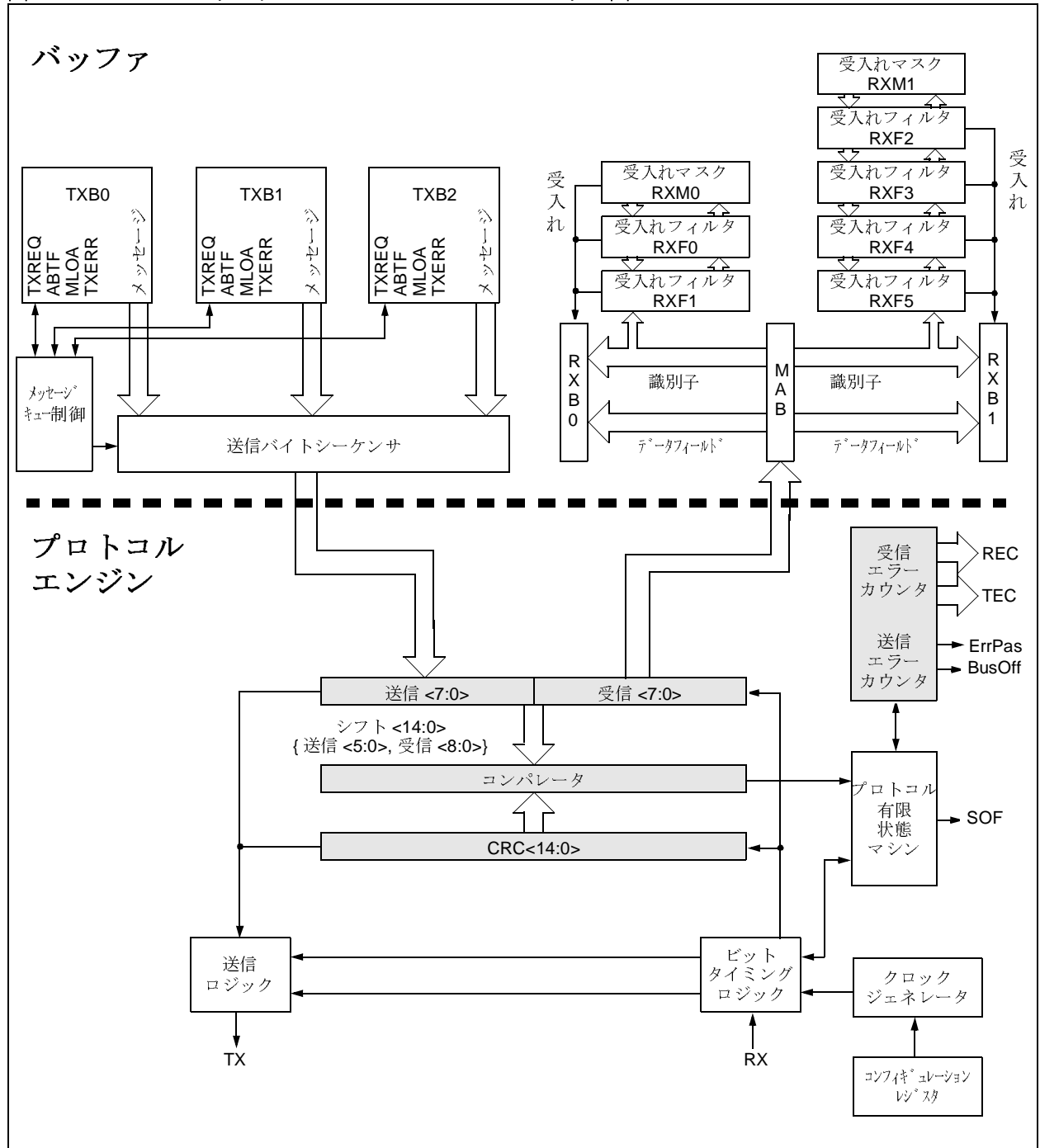
名称	PDIP/SOIC ピン#	TSSOP ピン#	I/O/P タイプ	説明	ピン代替機能
TXCAN	1	1	O	CAN バスへの送信出力ピン	—
RXCAN	2	2	I	CAN バスからの受信入力ピン	—
CLKOUT	3	3	O	プログラマブル分周付きクロック出力ピン	Start-of-Frame (SOF) 信号
TX0RTS	4	4	I	送信バッファ TXB0 用 送信要求。 100 kΩ で VDD にプルアップ	汎用のデジタル入力 100 kΩ で VDD にプルアップ
TX1RTS	5	5	I	送信バッファ TXB1 用 送信要求。 100 kΩ で VDD にプルアップ	汎用のデジタル入力 100 kΩ で VDD にプルアップ
TX2RTS	6	7	I	送信バッファ TXB2 用 送信要求。 100 kΩ で VDD にプルアップ	汎用のデジタル入力 100 kΩ で VDD にプルアップ
OSC2	7	8	O	クロック発振出力	—
OSC1	8	9	I	クロック発振入力	外部クロック入力
Vss	9	10	P	ロジックと I/O ピン用グラウンドリファレンス	—
RX1BF	10	11	O	受信バッファ RXB1 用 割り込みピン 汎用デジタル出力	汎用デジタル出力
RX0BF	11	12	O	受信バッファ RXB0 用 割り込みピン 汎用デジタル出力	汎用デジタル出力
INT	12	13	O	割り込み出力ピン	—
SCK	13	14	I	SPI™ インターフェース用クロック入力ピン	—
SI	14	16	I	SPI インターフェース用データ入力ピン	—
SO	15	17	O	SPI インターフェース用データ出力ピン	—
CS	16	18	I	SPI インターフェース用チップ選択ピン	—
RESET	17	19	I	負論理デバイスリセット入力	—
VDD	18	20	P	ロジックと I/O ピン用正電源	—
NC	—	6,15	—	未接続	

注: タイプ区別: I = 入力; O = 出力; P = 電源

1.4 送信/受信バッファ/マスク/フィルタ

MCP2515 は 3 個の送信バッファと 2 個の受信バッファ、2 個の受入れマスク (各受信バッファごとにひとつ)、6 個の受入れフィルタを持っています。図 1-3 は、これらのバッファとプロトコルエンジンとの接続構成のブロック図を示しています。

図 1-3: CAN バッファとプロトコルエンジンのブロック図



MCP2515

1.5 CAN プロトコルエンジン

CAN プロトコルエンジンは、図 1-4 に示したいくつかの機能ブロックを結合しています。それぞれ下記に説明します。

1.5.1 プロトコル有限状態マシン

エンジンの心臓部は、有限状態マシン (FSM) です。FSM は、TX/RX シフトレジスタ、CRC レジスタ、バスライン間のデータの流れを制御するシーケンサです。さらに、FSM は誤り制御ロジック (EML) と、TX/RX シフトレジスタとバッファ間の並列データの流れの制御も行っています。この FSM が、受信、アービトレーション、送信、誤りチェックが CAN プロトコルに従ったプロセスになるようにしています。バスライン上での自動メッセージ再送も FSM によって処理されます。

1.5.2 巡回冗長チェック

巡回冗長チェックレジスタが、制御フィールド後 (メッセージが 0 バイトの場合) とデータフィールド後に送信する巡回チェックコード (CRC) を生成し、さらに受信したメッセージの CRC フィールドのチェックを行うためにも使われます。

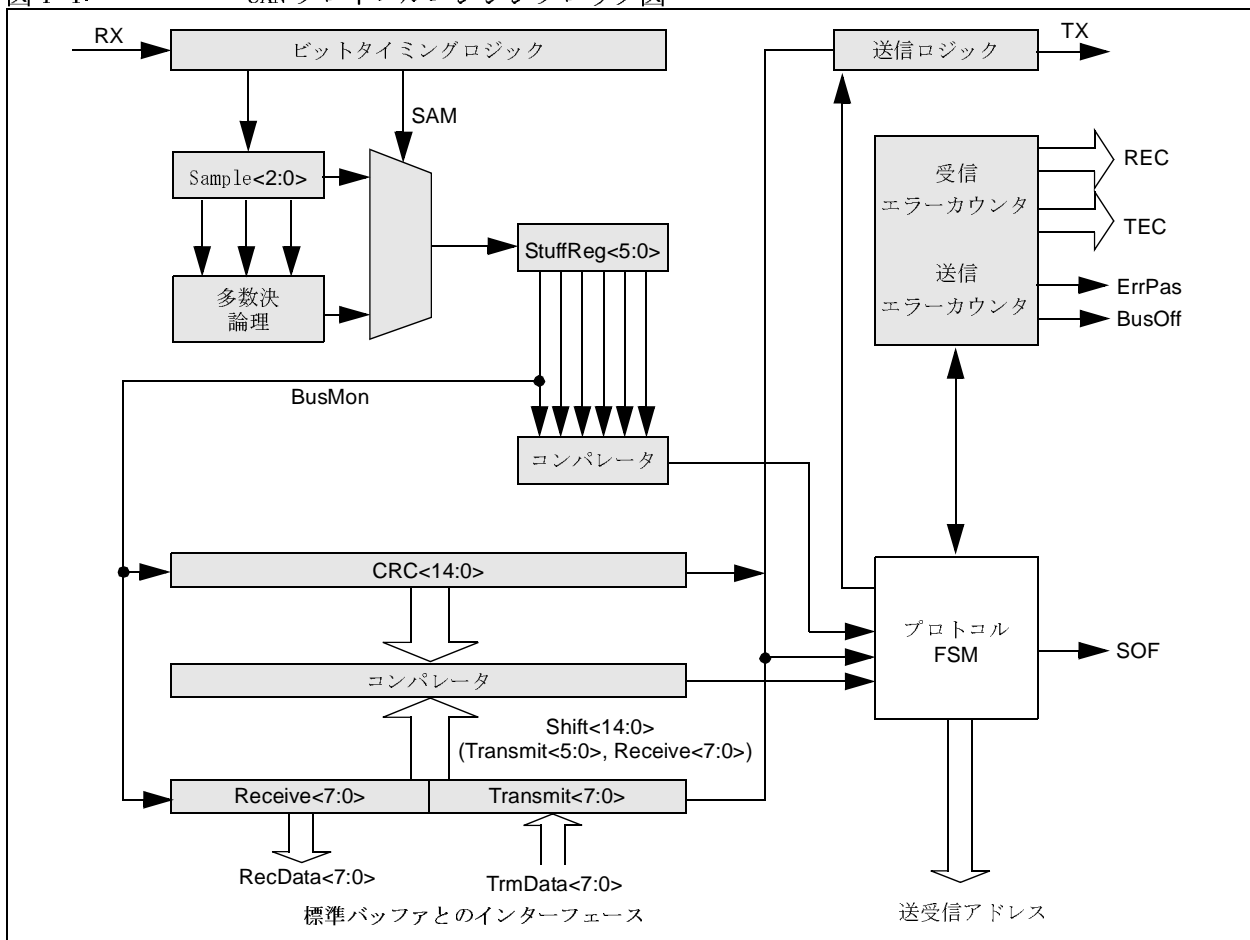
1.5.3 誤り制御ロジック

誤り制御ロジック (EML) は、CAN デバイスの誤り確認を司ります。受信エラーカウンタ (REC) と送信エラーカウンタ (TEC) の 2 つのカウンタを持っていて、ビット流れ処理部からのコマンドで増減されます。エラーカウンタの値に基づいて、CAN 制御部の状態が、エラーアクティブ、エラーパッシブ、バスオフのいずれかに設定されます。

1.5.4 ビットタイミングロジック

ビットタイミングロジック (BTL) は、バスラインの入力をモニタしていて、バス関連ビットタイミングを CAN プロトコルにしたがって処理します。BTL は Start-of-Frame (ハード同期) でリセッショからドミナントへのバス遷移に同期させます。さらに CAN 制御部自身がドミナントビット (再同期) を送信しない場合には、その後もリセッショからドミナントバス遷移に同期させます。また、BTL は伝播遅延時間や、位相シフトを補償したり、ビットタイム内のサンプル点を定めるために、プログラマブルな時間単位も提供します。BTL のプログラミングは、ボーレートと外付けの物理的な遅延時間に依存します。

図 1-4: CAN プロトコルエンジンブロック図



2.0 CAN メッセージフレーム

MCP2515は、CAN 2.0B仕様で定義されている標準データフレーム、拡張データフレーム、リモートフレーム（標準と拡張）をサポートしています。

2.1 標準データフレーム

CAN 標準データフレームは 図 2-1 で表されます。他のフレームと同じように、フレームは全てのノードのハード同期を取るためにドミナント状態である Start-Of-Frame(SOF) から始まります。

SOF の後に 12 ビットからなるアービトレーションフィールドが続きますが、それは 11 ビットの識別子とリモート送信要求ビット (RTR) から構成されます。RTR ビットは、データフレーム (RTR ビットドミナント) とリモートフレーム (RTR ビットリセッパ) とを区別するために用いられます。

アービトレーションフィールドの次は制御フィールドで 6 ビットで構成されます。このフィールドの最初のビットは識別子拡張 (IDE) と呼ばれ、これがドミナント状態のとき標準フレームであることを表します。引続くビットは CAN プロトコルの予約ビット 0 (RBO) で、ドミナントビットとして規定されています。制御フィールドの残りの 4 ビットはデータ長コード (DLC) で、メッセージに含まれるデータのバイト数 (0 ~ 8 バイト) を規定します。

制御フィールドの後はデータフィールドで、DLC により規定される長さ (0 ~ 8 バイト) の送信データバイトを含んでいます。

データフィールドの後に巡回冗長チェック (CRC) フィールドが続き、送信エラーの検出に用いられます。CRC フィールドは 15 ビットの CRC シーケンスとリセッパ CRC デリミタビットで構成されます。

最後のフィールドは 2 ビットのアクノレッジフィールド (ACK) です。ACK スロットビットの間に、送信ノードがリセッパビットを送信します。エラーの無いフレームを受信したどのノードも、(ノードが特定メッセージを受け付けるように構成されているかどうかに関わらず) ドミナントビットを返信することにより正しいフレームを受信したことを認識できるようにします。リセッパのアクノレッジデリミタでアクノレッジスロットを終了させ、ドミナントビットにより上書きされないようにします。

2.2 拡張データフレーム

拡張 CAN データフレームでは、図 2-2 に示すように、SOF に 32 ビットで構成されるアービトレーションフィールドが続きます。最初の 11 ビットは 29 ビットの識別子 (Base-ID) の上位 11 ビット (MSB) です。これらの 11 ビットに続き、リセッパとして送信される代替リモート要求ビット (SRR) がきます。SRR に続き、フレームが拡張 CAN フレームであるこ

とを示すリセッパの IDE ビットがきます。こうすると、識別子の最初の 11 ビットの送信後にアービトレーションが未解決の場合、アービトレーションにからんでいるノードの 1 つが標準 CAN フレーム (11 ビット識別子) を送信していると、標準 CAN フレームは、ドミナント IDE ビットであるためアービトレーションを獲得します。また、拡張 CAN フレーム内の SRR ビットは、標準 CAN リモートフレームを送信しているノードによりドミナント RTR が発行できるようにリセッパでなければなりません。

SRR と IDE ビットに引続き、識別子 (Extended ID) の残りの 18 ビットとドミナントリモート送信要求ビットが続きます。

共有ネットワーク間で、標準と拡張フレームが送信されることを有効にするために、29 ビットの拡張メッセージ識別子を 11 ビット (上位) と 18 ビット (下位) セクションに分割する必要があります。この分割により、IDE ビットが標準と拡張の両フレーム内で同じ位置を維持することができます。

アービトレーションフィールドの次は 6 ビット制御フィールドです。このフィールドの最初の 2 ビットは予約され、ドミナント状態でなければなりません。制御フィールドの残りの 4 ビットは DLC で、メッセージに含まれるデータバイト数を規定します。

フレームの残りの部分 (データフィールド、CRC フィールド、通知フィールド、フレーム終了および中断) は、標準データフレームと同様に構成されます。(第 2.1 項 “標準データフレーム” 参照)

2.3 リモートフレーム

データ送信は、通常データソースノードを使って自主的に動作します (例えば、センサはデータフレームを送出します)。しかし、宛先ノードがソースからデータを要求することができます。この目的のために、宛先ノードは、要求されたデータフレームの識別子と同じ識別子のリモートフレームを送信します。対応するデータソースノードは、このリモート要求の応答としてデータフレームを送信します。

リモートフレームとデータフレームとは 2 つの違いがあります。(図 2-3 に示す) 第 1 に RTR ビットがリセッパ状態であり、第 2 にデータフィールドが無いことです。同じ識別子を持ったデータフレームとリモートデータフレームが、同時に送信される場合、識別子に続くドミナント RTR のために、データフレームがアービトレーション権を獲得します。このようにして、リモートフレームを送信したノードは所望のデータを直ぐに受信します。

2.4 エラーフレーム

エラーフレームは、バスエラーを検出したノードによって生成されます。図 2-4 に示すように、エラーフレームは、エラーフラグフィールドとそれに続くエ

ラーデリミタフィールドの2つのフィールドにより構成されます。エラーフラグフィールドには2つの形式があります。エラーフラグフィールドの形式は、エラーを検出しエラーフラグフィールドを生成したノードのエラー状態に依存します。

2.4.1 アクティブエラー

エラーアクティブノードがバスエラーを検出した場合、そのノードはアクティブエラーフラグを生成して現メッセージの送信に割り込みをかけます。アクティブエラーフラグは、6個の連続するドミナントビットで構成されます。このビットシーケンスは、ビットスタッフルールに敢えて違反します。その他の全てのノードがビットスタッフルールを認識し、今度は自分自身で、エラーエコーフラグと呼ばれるエラーフレームを生成します。

従って、エラーフラグフィールドは6から12の連続するドミナントビット（1つもしくは2つのノードにより生成されます）で構成されます。エラーデリミタ（8ビットのリセシブビット）によりエラーフレームが完了します。エラーフレーム完了後、バスは通常状態に戻り、割り込みをかけられたノードは中断されたメッセージを再送信します。

注： エラーエコーフラグは、通常エラーフラグを送信しようとした1つか2つのノード（全部ではない）の局所的な障害で発生します。残りのノードは、最初のエラーフラグに対応した応答（エコー）を生成します。

2.4.2 パッシブエラー

エラーパッシブノードがバスエラーを検出した場合、そのノードはエラーパッシブフラグを送信し、エラーデリミタフィールドが続きます。エラーパッシブフラグは6個の連続するリセシブビットで構成されます。エラーパッシブノードのエラーフレームは、14ビットのリセシブビットで構成されます。したがって、バスエラーがエラーアクティブノード、または送信ノードで検出されなければ、エラーパッシブフラグがバスを妨害しないため、メッセージは正常に送信続きます。

送信ノードがエラーパッシブフラグを生成した場合は、その結果としてビットスタッフルールに違反するため、その他のノードがエラーフレームを生成します。エラーフレーム送信後、エラーパッシブノードは、バス通信に再参加しようとする前に、バス上の6個の連続するリセシブビットを待たねばなりません。

エラーデリミタは8ビットのリセシブビットで構成されており、バスノードがエラー発生後きれいにバス通信を再開できるようにします。

2.5 オーバーロードフレーム

オーバーロードフレームは、図 2-5 に示すように、アクティブエラーフレームと同じフォーマットを持ちます。しかし、オーバーロードフレームは、インターフレーム空間の間にのみ生成できます。このようにして、オーバーロードフレームは、エラーフレームと区別されます（エラーフレームは、メッセージの送信の間に送信されます）。オーバーロードフレームは、オーバーロードフラグとそれに続くオーバーロードデリミタの2フィールドで構成されます。オーバーロードフラグは6ビットのドミナントビットで構成され、それに続き他のノードで生成されたオーバーロードフラグが続きます（アクティブエラーフラグとして最大12ビットのドミナントビットを与えます）。オーバーロードデリミタは8ビットのリセシブビットで構成されます。オーバーロードフレームは、次の2つの場合にノードで生成されます。

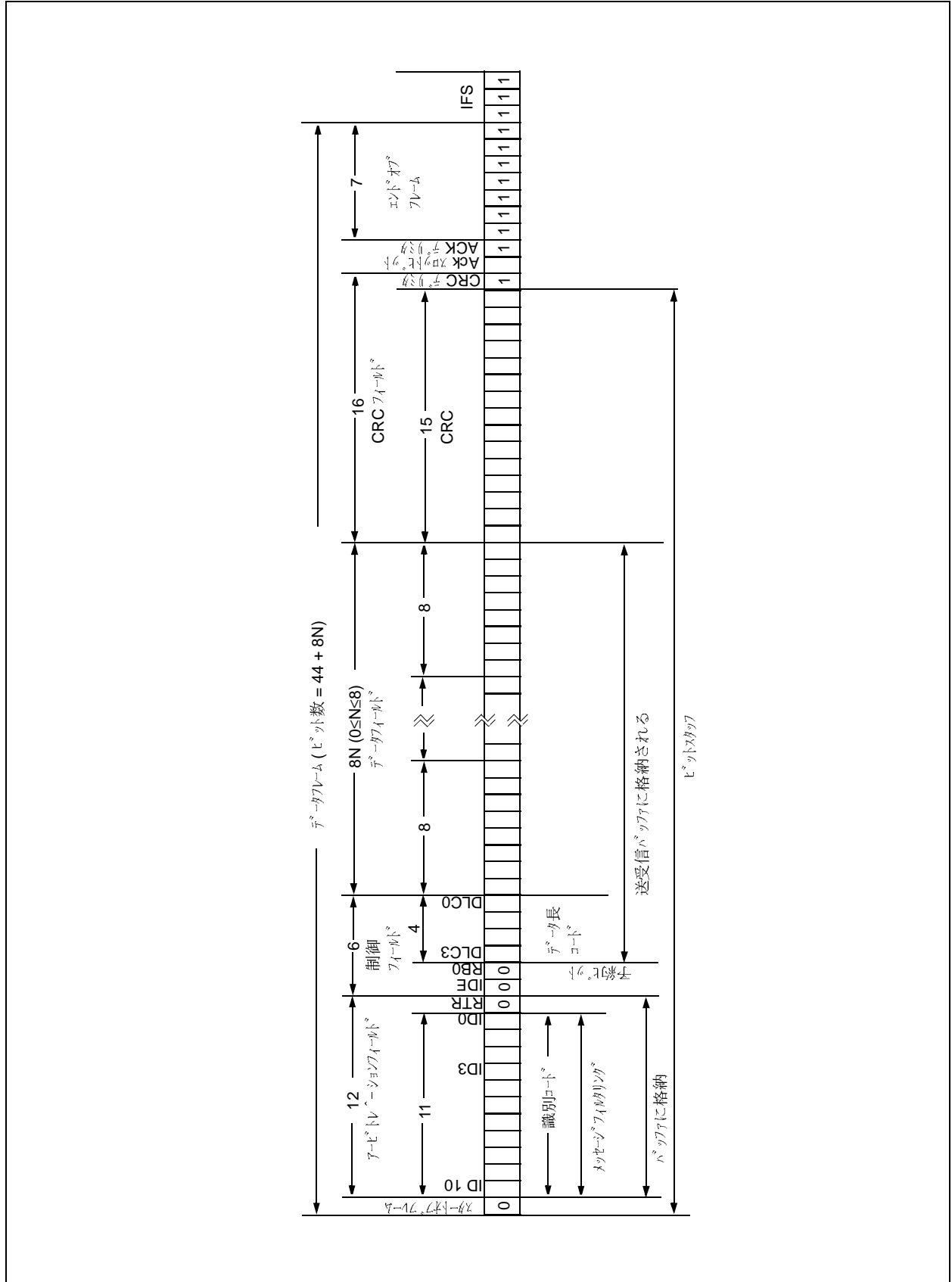
1. ノードが、インターフレーム空間の間にドミナントビットを検出したとき、これは不法な条件です。
例外：ドミナントビットはIFSの3ビット目に検出された場合、レシーバはこれをSOFとして解釈します。
2. 内部状態により、ノードが、まだ次のメッセージの受信開始ができないとき、次のメッセージの開始を遅らせるために、ノードは最大2つのシーケンシャルなオーバーロードフレームを生成することがあります。

注： 2のケースは、MCP2515では内部遅延が非常に小さいので起きることはありません。

2.6 インターフレーム空間 (IFS)

インターフレーム空間は先行する（どんな形式の）フレームと、引き続きデータもしくはリモートエラーフレームとを分けます。インターフレーム空間は、少なくとも3ビットのリセシブビットから構成され、インターミッションと呼ばれます。これは、次のメッセージフレームの開始以前に受信ノードによるメッセージの内部処理を行う時間を与えるために用意されています。インターミッション後、バスラインは、次の送信開始までリセシブ状態（バスアイドル）を保持します。

図 2-1: 標準データフレーム



MCP2515

図 2-2: 拡張データフレーム

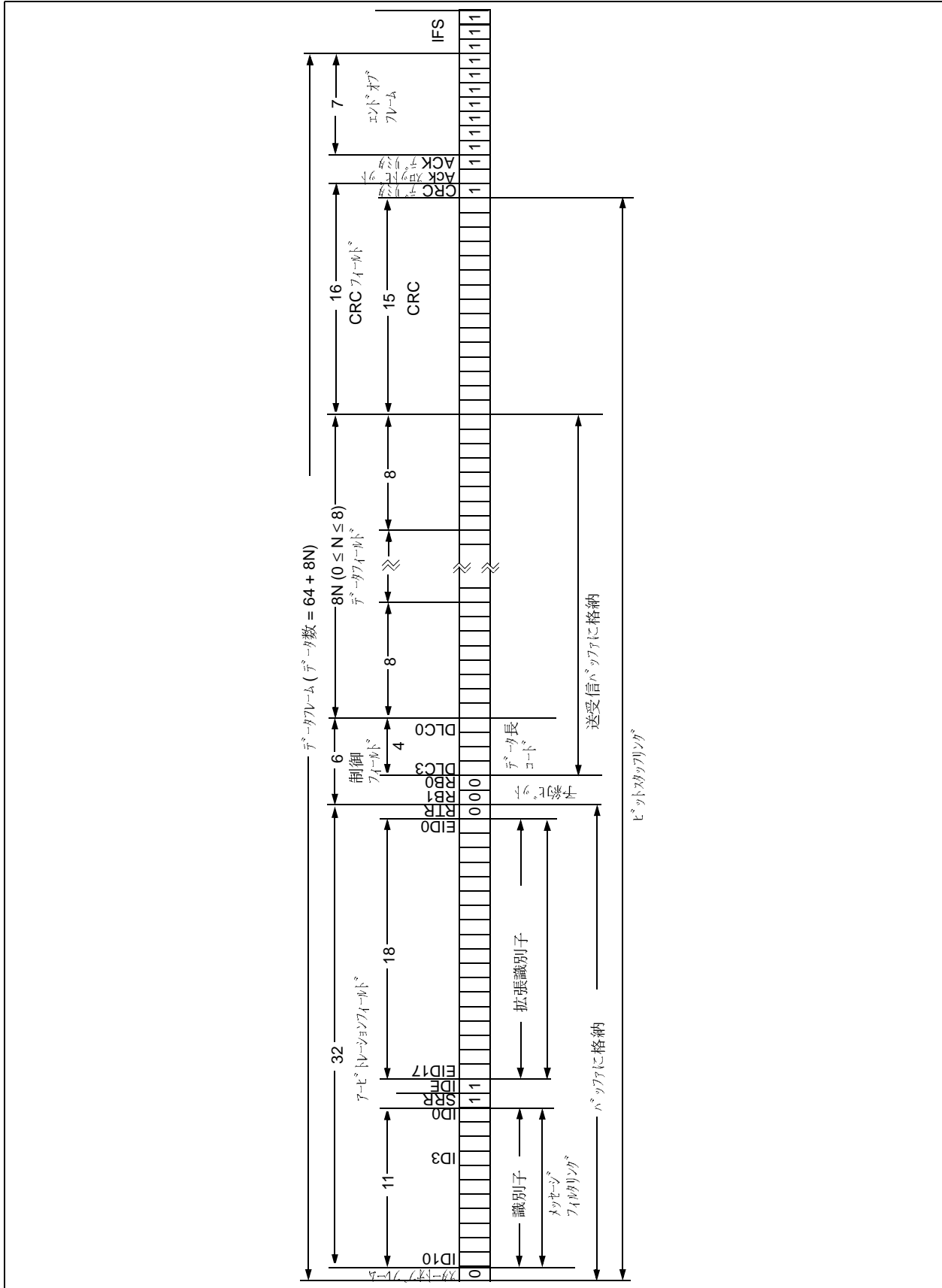
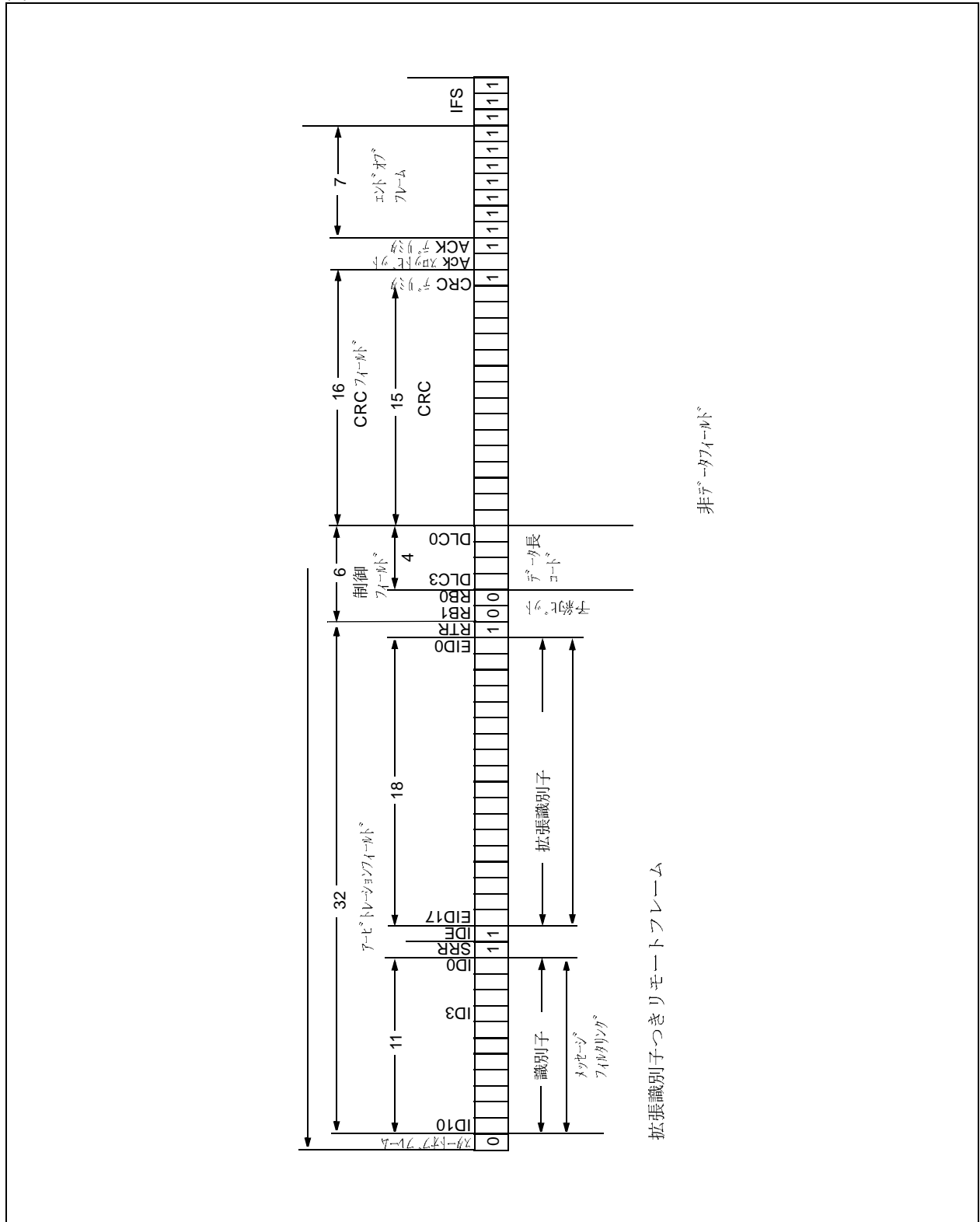
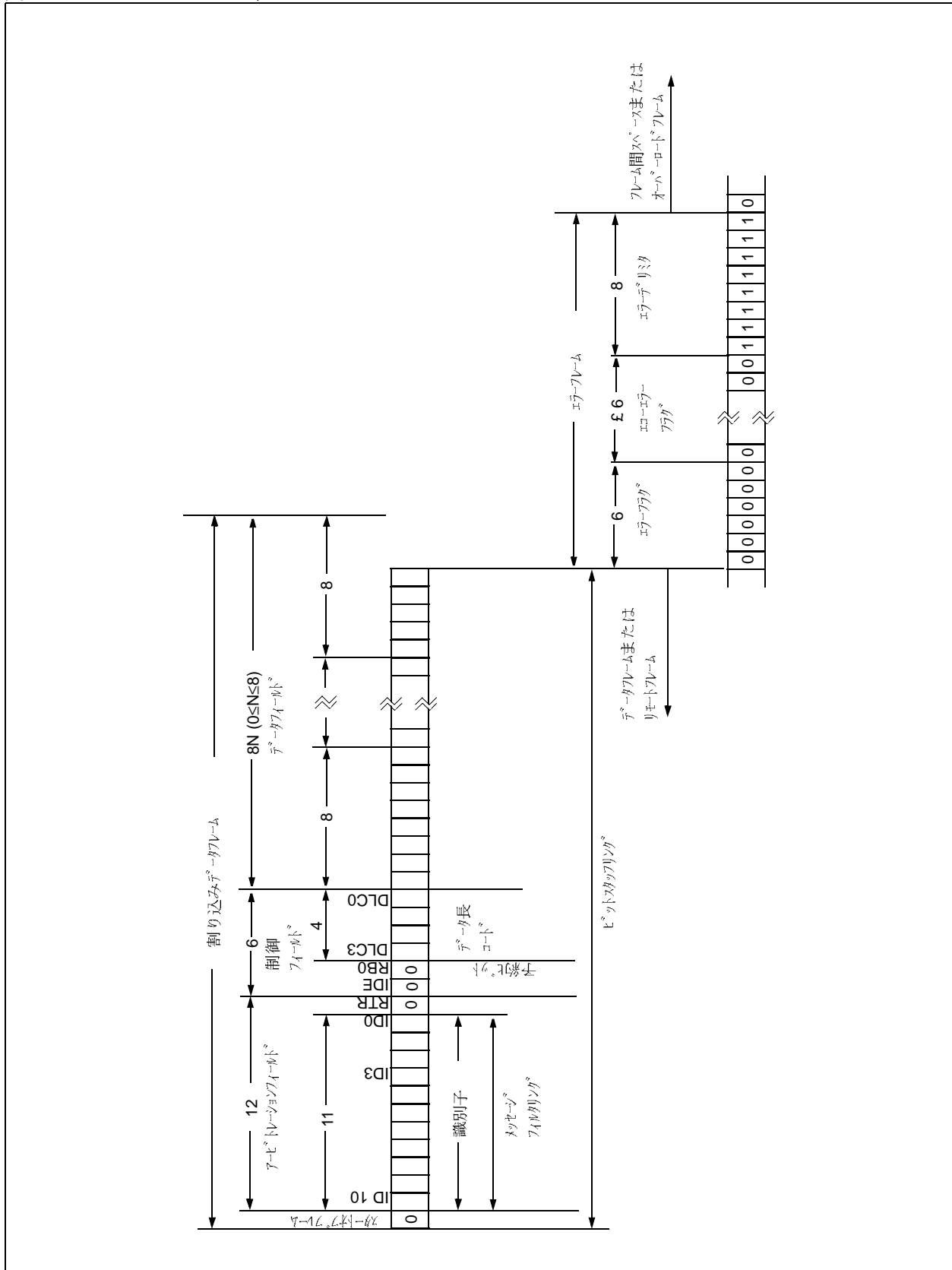


図 2-3: リモートフレーム



MCP2515

図 2-4: アクティブエラーフレーム



MCP2515

NOTES:

3.0 メッセージ送信

3.1 送信バッファ

MCP2515 は 3 つの送信バッファを持っています。バッファごとに 14 バイトの SRAM を占有しデバイスメモリマップ内にマッピングされます。

最初のバイト TXBnCTRL は、メッセージバッファに関する制御レジスタです。このレジスタ内の情報によってメッセージを送信するかどうかを決めたり、メッセージ送信状態を表しています。(レジスタ 3-1 参照)

5 バイトが標準および拡張識別子とその他のメッセージアービトレーション情報を保持します。(レジスタ 3-3 からレジスタ 3-7 参照)

最後の 8 バイトは、送信データとして使える 8 ビットデータバイトとなります。(レジスタ 3-8 参照)

少なくとも TXBnSIDH, TXBnSIDL, TXBnDLC レジスタには設定が必須です。もしメッセージにデータバイトがあるときは、TXBnDm レジスタにも設定が必要です。メッセージが拡張識別子として使われる場合には、TXBnEIDm レジスタも設定が必須で、TXBnSIDL.EXIDE ビットのセットが必須です。

メッセージを送る前に、MCU は CANINTE.TXInE ビットによってメッセージ送信後の割り込みを生成するかどうかを初期設定する必要があります。

注： TXBnCTRL.TXREQ ビット (送信バッファが送信待ちではないことを表す) は、送信バッファに書き込む前にクリアしなければなりません。

3.2 送信優先度

送信優先度は、送信待ちのメッセージを持つ MCP2515 間で優先付けします。この優先順位は、CAN プロトコルに組み込まれているメッセージアービトレーションによる優先度とは独立で必須の関係はありません。

SOF の送信前に、送信キューにあるすべてのバッファの優先度が比較されます。一番高い優先度を持った送信バッファが最初に送信されます。例えば、送信バッファ 0 が送信バッファ 1 よりも高い優先度を設定されている場合、バッファ 0 が最初に送信されます。2 つのバッファが同じ優先度設定されている場合、一番高いアドレスを持ったバッファが送信されます。例えば、送信バッファ 1 が送信バッファ 0 と同じ優先度設定の場合、バッファ 1 が最初に送信されます。

4 つの送信優先度レベルがあります。特定のメッセージバッファの TXBnCTRL.TXP<1:0> が '11' に設定されている場合、そのバッファが最高の優先度を持ちます。特定のメッセージバッファの TXBnCTRL.TXP<1:0> が '00' に設定されている場合、そのバッファは最低の優先度を持ちます。

3.3 送信の起動

メッセージ送信を起動するには、送信する各バッファの TXBnCTRL.TXREQ ビットをセットする必要があります。これは下記によって行うことができます。

- SPI 書き込みコマンドによってレジスタに書く
- SPI RTS コマンドを送る
- 送信する特定の送信バッファの TXnRTS ピンを Low にする。

SPI インターフェース経由で送信を起動する場合には、TXP 優先ビットとして TXREQ ビットも同時に設定することができます。

TXBnCTRL.TXREQ ビットがセットされると、TXBnCTRL.ABTF、TXBnCTRL.MLOA、TXBnCTRL.TXERR ビットは自動的にクリアされます。

注： TXBnCTRL.TXREQ ビットのセットではメッセージ送信は起動されません。それはメッセージバッファが送信レディになったというフラグでしかありません。送信は、デバイスがバスが有効であること検知したとき開始されます。

送信が正常に完了すると TXBnCTRL.TXREQ ビットがクリアされ、CANINTF.TXnIF ビットがセットされます。これで CANINTE.TXnIE ビットがセットされていれば割り込みを発生します。

メッセージ送信が失敗した場合には、TXBnCTRL.TXREQ はセットされたままです。これはメッセージがまだ送信待ちであることを示し、下記のいずれかの状態フラグとなります。

- メッセージ送信を開始したときエラー状態に出くわした場合には、TXBnCTRL.TXERR と CANINTF.MERRF ビットがセットされ、CANINTE.MERRE ビットがセットされていれば、INT ピンに割り込みを発生します。
- メッセージが失われた場合には TXBnCTRL.MLOA ビットのアービトレーションがセットされます。

注： ワンショットモードがイネーブル (CANCTRL.OSM) のときは、上記条件は、依然存在しますが、TXREQ ビットはクリアされ、メッセージの 2 回目の送信はしません。

3.4 ワンショットモード

ワンショットモードでは、メッセージは 1 回だけしか送信しません。通常は、CAN メッセージがアービトレーションを失うか、エラーフレームによって壊れたときには、メッセージは再送されます。ワンショットモードイネーブルのときは、アービトレーションを失っても、エラーフレームでも、メッセージは 1 度だけしか送信されません。

ワンショットモードには、TTCAN のような決まったスロットタイムを持っている必要があります。

3.5 TXnRTS ピン

$\overline{\text{TXnRTS}}$ ピンは、下記のように構成することができます。

- 送信要求入力として、任意の送信バッファからメッセージを送信するための別の起動手段を提供します。
- 標準デジタル入力

これらのピンの構成と制御は TXRTSCTRL レジスタで行います。(レジスタ 3-2 参照)

TXRTSCTRL レジスタは、MCP2515 がコンフィギュレーションモードのときだけ変更ができます。(第 10.0 項 “動作モード” 参照) 送信要求ピンとして動作するよう構成された場合には、ピンは各送信バッファに対応する TXBnCTRL.TXREQ ビットにマッピングされます。TXREQ ビットは $\overline{\text{TXnRTS}}$ ピンの立下りでラッチされます。 $\overline{\text{TXnRTS}}$ ピンは $\overline{\text{RXnBF}}$ ピンに直結するように設計されていて、 $\overline{\text{RXnBF}}$ ピンが Low になったときにメッセージ送信が自動的に行われるようになります。 $\overline{\text{TXnRTS}}$ ピンは 100 k Ω (標準) の内蔵プルアップ抵抗を持っています。

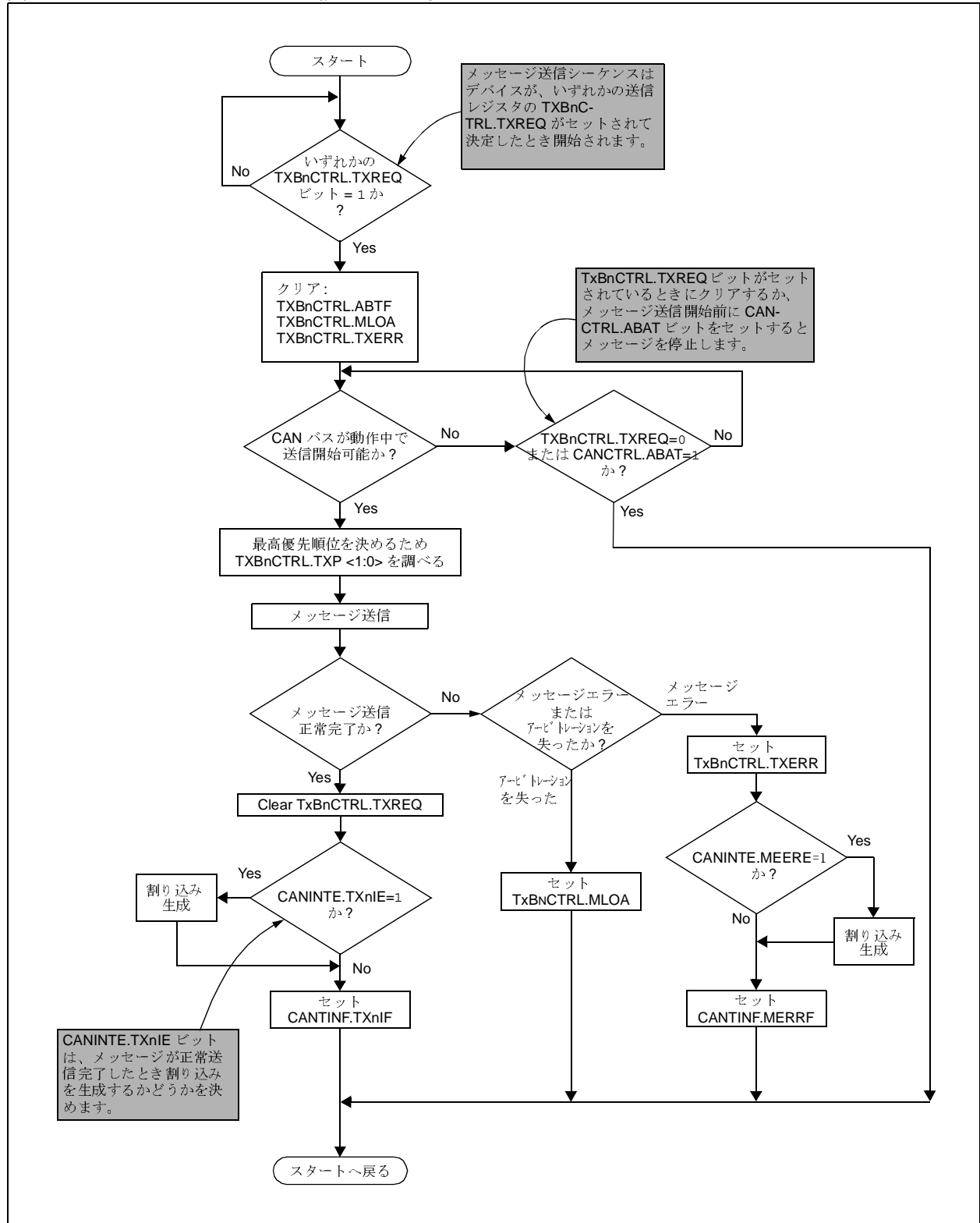
3.6 送信停止

MCU はメッセージバッファに対応した TXBnCTRL.TXREQ ビットをクリアすることでメッセージ停止を要求することができます。

さらに、すべての送信待ちメッセージを CANCTRL.ABAT ビットをセットすることで停止させることができます。このビットは、(通常、TXREQ ビットが参照されてクリアされた後) メッセージ送信を続けている間はリセットされていなければなりません。CANCTRL.ABTF フラグは、CANCTRL.ABAT ビットによって停止要求されたときだけセットされます。TXREQ ビットをリセットしてメッセージを停止させると、ABTF ビットをセットする要因にはなりません。

注：	停止要求されたとき、すでに送信中のメッセージはそのまま送信を継続します。メッセージ送信が正常に完了しなかったら (例えばアービトレーションを失ったり、エラーフレームに割り込まれたとき)、停止されます。
----	--

図 3-1: メッセージ送信フローチャート



MCP2515

レジスタ 3-1: TXBnCTRL : 送信バッファ n 制御レジスタ
(アドレス: 30h, 40h, 50h)

U-0	R-0	R-0	R-0	R/W-0	U-0	R/W-0	R/W-0
—	ABTF	MLOA	TXERR	TXREQ	—	TXP1	TXP0

bit 7

bit 0

- bit 7 未実装: 読むと '0'
- bit 6 **ABTF**: メッセージ停止フラグ
1 = メッセージは停止された
0 = メッセージ送信正常完了
- bit 5 **MLOA**: メッセージはアービトレーションビットを失った
1 = 送信中にメッセージがアービトレーションを失った
0 = 送信中メッセージはアービトレーションを失っていない
- bit 4 **TXERR**: 送信エラー検出ビット
1 = メッセージ送信中にバスエラーが発生した
0 = メッセージ送信中にバスエラーが発生しなかった
- bit 3 **TXREQ**: メッセージ送信要求ビット
1 = バッファは現在送信待ち中
(MCU はメッセージ送信要求で本ビットをセット、送信完了で自動的にクリア)
0 = バッファは現在送信待ち中ではない
(MCU はメッセージを中止するため本ビットをクリアできる)
- bit 2 未実装: 読むと '0'
- bit 1-0 **TXP**: 送信バッファ優先順位 <1:0>
11 = 最高メッセージ優先順位
10 = 中高位メッセージ優先順位
01 = 中低位メッセージ優先順位
00 = 最低位メッセージ優先順位

凡例:

R = 読み込み可能	W = 書き込み可能	U = 未実装, 読むと '0'
-n = POR 時の値	'1' = セット	'0' = クリア x = 不定

レジスタ 3-2: TXRTSCTRL : TXnRTS ピン制御と状態レジスタ
(アドレス: 0Dh)

U-0	U-0	R-x	R-x	R-x	R/W-0	R/W-0	R/W-0
—	—	B2RTS	B1RTS	B0RTS	B2RTSM	B1RTSM	B0RTSM
bit 7				bit 0			

- bit 7 未実装: 読むと '0'
- bit 6 未実装: 読むと '0'
- bit 5 **B2RTS:** $\overline{\text{TX2RTS}}$ ピン状態ビット
 - デジタル入力モードのとき $\overline{\text{TX2RTS}}$ ピンの状態を読む
 - ピンが送信要求モードのとき読むと '0'
- bit 4 **B1RTS:** $\overline{\text{TX1RTX}}$ ピン状態ビット
 - デジタル入力モードのときは、 $\overline{\text{TX1RTS}}$ ピンの状態を読む
 - ピンを送信要求モードで使うときには読むと '0'
- bit 3 **B0RTS:** $\overline{\text{TX0RTS}}$ ピンの状態ビット
 - デジタル入力モードのときは $\overline{\text{TX0RTS}}$ ピンの状態を読む
 - ピンを送信要求モードで使うときには読むと '0'
- bit 2 **B2RTSM:** $\overline{\text{TX2RTS}}$ ピンモードビット
 1 = TXB2 バッファのメッセージ送信要求のピンとして使う。(立下りエッジで)
 0 = デジタル入力
- bit 1 **B1RTSM:** $\overline{\text{TX1RTS}}$ ピンモードビット
 1 = TXB1 バッファのメッセージ送信要求のピンとして使う。(立下りエッジで)
 0 = デジタル入力
- bit 0 **B0RTSM:** $\overline{\text{TX0RTS}}$ ピンモードビット
 1 = TXB0 バッファのメッセージ送信要求のピンとして使う。(立下りエッジで)
 0 = デジタル入力

凡例:

R = 読み込み可能	W = 書き込み可能	U = 未実装, 読むと '0'
-n = POR 時の値	'1' = セット	'0' = クリア x = 不定

MCP2515

レジスタ 3-3: TXBnSIDH : 送信バッファ n 用の標準識別子 上位
(アドレス: 31h, 41h, 51h)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3

bit 7

bit 0

bit 7-0 **SID:** 標準識別子ビット <10:3>

凡例:

R = 読み込み可能

W = 書き込み可能

U = 未実装, 読むと '0'

-n = POR 時の値

'1' = セット

'0' = クリア

x = 不定

レジスタ 3-4: TXBnSIDL : 送信バッファ n用の標準識別子 下位
(アドレス: 32h, 42h, 52h)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
SID2	SID1	SID0	—	EXIDE	—	EID17	EID16
bit 7				bit 0			

- bit 7-5 **SID:** 標準識別子ビット <2:0>
- bit 4 未実装: 読むと '0'
- bit 3 **EXIDE:** 拡張識別子イネーブルビット
1 = メッセージを拡張識別子として送信する
0 = メッセージを標準識別子として送信する
- bit 2 未実装: 読むと '0'
- bit 1-0 **EID:** 拡張識別子ビット <17:16>

凡例:			
R = 読み込み可能	W = 書き込み可能	U = 未実装, 読むと '0'	
-n = POR 時の値	'1' = セット	'0' = クリア	x = 不定

MCP2515

レジスタ 3-5: TXBnEID8 : 送信バッファ n 用の拡張識別子 上位
(アドレス: 33h, 43h, 53h)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8

bit 7

bit 0

bit 7-0 **EID:** 拡張識別子ビット <15:8>

凡例:

R = 読み込み可能

W = 書き込み可能

U = 未実装, 読むと '0'

-n = POR 時の値

'1' = セット

'0' = クリア

x = 不定

レジスタ 3-6: TXBnEID0 : 送信バッファ n 用拡張識別子下位
(アドレス: 34h, 44h, 54h)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	
EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	
bit 7								bit 0

bit 7-0 **EID**: 拡張識別子ビット <7:0>

凡例:

R = 読み込み可能	W = 書き込み可能	U = 未実装, 読むと '0'
-n = POR 時の値	'1' = セット	'0' = クリア x = 不定

MCP2515

レジスタ 3-7: TXBnDLC - 送信バッファ n 用のデータ長コード
(アドレス: 35h, 45h, 55h)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	RTR	—	—	DLC3	DLC2	DLC1	DLC0

bit 7 bit 0

bit 7 未実装: 読むと '0'

bit 6 **RTR**: リモート送信要求ビット

1 = メッセージをリモート送信要求として送信する
0 = メッセージをデータフレームとして送信する

bit 5-4 未実装: 読むと '0'

bit 3-0 **DLC**: データ長コード <3:0>

送信されるデータバイト数をセットする (0 ~ 8 bytes)

注: DLC には 8 以上の値を設定することができますが、送信されるのは 8 バイトだけです。

凡例:

R = 読み込み可能

W = 書き込み可能

U = 未実装, 読むと '0'

-n = POR 時の値

'1' = セット

'0' = クリア

x = 不定

レジスタ 3-8: TXBnDm : 送信バッファ n 用のデータバイト数 m
 (アドレス: 36h - 3Dh, 46h - 4Dh, 56h - 5Dh)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
TXBnDm	TXBnDm	TXBnDm	TXBnDm	TXBnDm	TXBnDm	TXBnDm	TXBnDm
7	6	5	4	3	2	1	0

bit 7 bit 0

bit 7-0 **TXBnDM7:TXBnDM0:** 送信バッファ n 用データフィールドバイト m

凡例:			
R = 読み込み可能	W = 書き込み可能	U = 未実装, 読むと '0'	
-n = POR 時の値	'1' = セット	'0' = クリア	x = 不定

MCP2515

NOTES:

4.0 メッセージ受信

4.1 受信メッセージのバッファ

MCP2515 は2つのマルチ受入フィルタ付きの完全な受信バッファをもっています。さらに3つめの受信バッファとして働くメッセージアセンブリバッファ (MAB) を別にもっています。(図 4-2 参照)

4.1.1 メッセージアセンブリバッファ

3つの受信バッファのうち、MAB は常にバスからの次のメッセージを受信可能となっています。MAB はすべての受信メッセージを組み立てます。これらのメッセージは、受入フィルタに一致したものが RXBn バッファ(レジスタ 4-4 から レジスタ 4-9 参照)に転送されます。

4.1.2 RXB0 と RXB1

残りの2つの受信バッファは、RXB0 と RXB1 と呼ばれ、MAB 経由でプロトコルエンジンからの完全なメッセージを受信することができます。MPU が1つのバッファを処理している間、もう1つのバッファは受信もしくは前に受信されたデータを保持するために使用されます

注： いったんメッセージが受け入れられると、MAB の内容全体が受信バッファに移動されます。つまり、識別子 (標準か拡張) や受信データバイト数に関わらず、受信バッファ全体がMAB の内容で上書きされてしまいます。したがって、バッファ内の全レジスタの内容は、何らかのメッセージを受信したら書き換わると仮定しなければなりません。

4.1.3 受信フラグ / 割り込み

メッセージがいずれかの受信バッファに移されると、対応する CANINTF.RXnIF ビットがセットされます。新しいメッセージをバッファに受信するためには、MCU によりこのビットをクリアしなければなりません。このビットは、MCU がメッセージを処理完了するまで、MCP2515 が受信バッファに新たなメッセージをロードするのを確実にロックアウトします。

もし、CANINTE.RXnIE ビットがセットされていると、割り込みが INT ピンに生成され、有効なメッセージが受信されたことを通知します。さらに、対応する RXnBF ピンが受信バッファフルピンとして構成されていれば、ピンは Low にドライブされます。詳細は第 4.4 項 “RX0BF と RX1BF ピン” を参照してください。

4.2 受信優先度

RXB0 は最優先バッファで、1つのマスクと2つのメッセージ受入フィルタを持っています。受信したメッセージには、まずこの RXB0 のマスクとフィルタが適用されます。

RXB1 は低優先度のバッファで、1つのマスクと4つの受入フィルタを持っています。

まず RXB0 のマスクとフィルタが適用されたメッセージは、さらに、低い番号の受入フィルタが適用され、より厳しく RXB0 に一致するかが試されるので、バッファに対しより優先度が高くなることとなります。

メッセージが受信されると、RXBnCTRL レジスタのビット <3:0> が受入れた受入フィルタの番号と、受信メッセージがリモート送信要求かどうかを示しています。

4.2.1 ロールオーバー

これに加えて、RXB0 が有効なメッセージを含んでいるときに他の有効なメッセージを受信したら、オーバーフローを起こさず、RXB1 の受入基準にかかわらず、新しいメッセージを RXB1 に移すように RXB0CTRL レジスタを構成することができます。

4.2.2 RXM ビット

RXBnCTRL.RXM ビットは、特殊受信モードをセットします。通常は、これらのビットは 00 にクリアされ、対応する受入フィルタで決定されたすべての有効なメッセージが受信できるようにします。この場合には、標準または拡張メッセージを受け入れるかどうかを決めるのは、受入フィルタレジスタ内の RFXnSIDL.EXIDE ビットで決められます。

RXBnCTRL.RXM ビットが 01 か 10 にセットされると、受信機は、設定に応じて標準か拡張メッセージどちらかしか受信しないようになります。

受入フィルタの RFXnSIDL.EXIDE ビットがセットされていると、RXBnCTRL.RXM のモードと一致せず、受入フィルタは無効となります。この RXBnCTRL.RXM ビットによる2つのモードは、バスに標準か拡張メッセージのいずれかしかいないシステムで使われます。

RXBnCTRL.RXM ビットが 11 にセットされると、受入フィルタの値に関わらず、バッファはすべてのメッセージを受信します。また、メッセージが EOF より前にエラーとなったときには、エラーフレーム前までに MAB に組み立てられた分のメッセージがバッファにロードされます。このモードは CAN システムのデバッグ用で、実システム環境では使われません。

4.3 スタートオブフレーム信号

RXCAN ピンに受信された CAN メッセージの最初で、スタートオブフレーム信号が許可されていれば、SOF ピンに出力されます。

RXCAN ピンはアイドルバスをモニタしていて、リセッспからドミナントのエッジ変化を見つけるようにします。サンプル点までドミナント状態のままなら、MCP2515 はこれを SOF と解釈し、SOF パルスを生成します。サンプル点までドミナント状態のままであれば、MCP2515 はこれをバス上のグリッチと解釈し、SOF 信号は出力しません。図 4-1 に SOF 信号とグリッチフィルタについて説明しています。

TTCAN タイプのシステムでは、ワンショットモードとして SOF 信号が使われます。さらに、RXCAN ピンと SOF ピンを一緒にモニタすると、わずかなグリッチも検出することができますので、CAN 通信に影響を与える前に、MCU が早期にバスの物理的問題を検出することができます。

4.4 RX0BF と RX1BF ピン

MCU に多くの異なる条件で割り込み信号が提供できるように、INT ピンに加えて受信バッファフルピン (RX0BF と RX1BF) が、それぞれ RXB0 か RXB1 に新たなメッセージがロードされたことを表すのに使えます。これらのピンは下記の 3 つの異なる構成が取れます。(レジスタ 4-1)

1. 禁止
2. バッファフル割り込み

3. デジタル出力

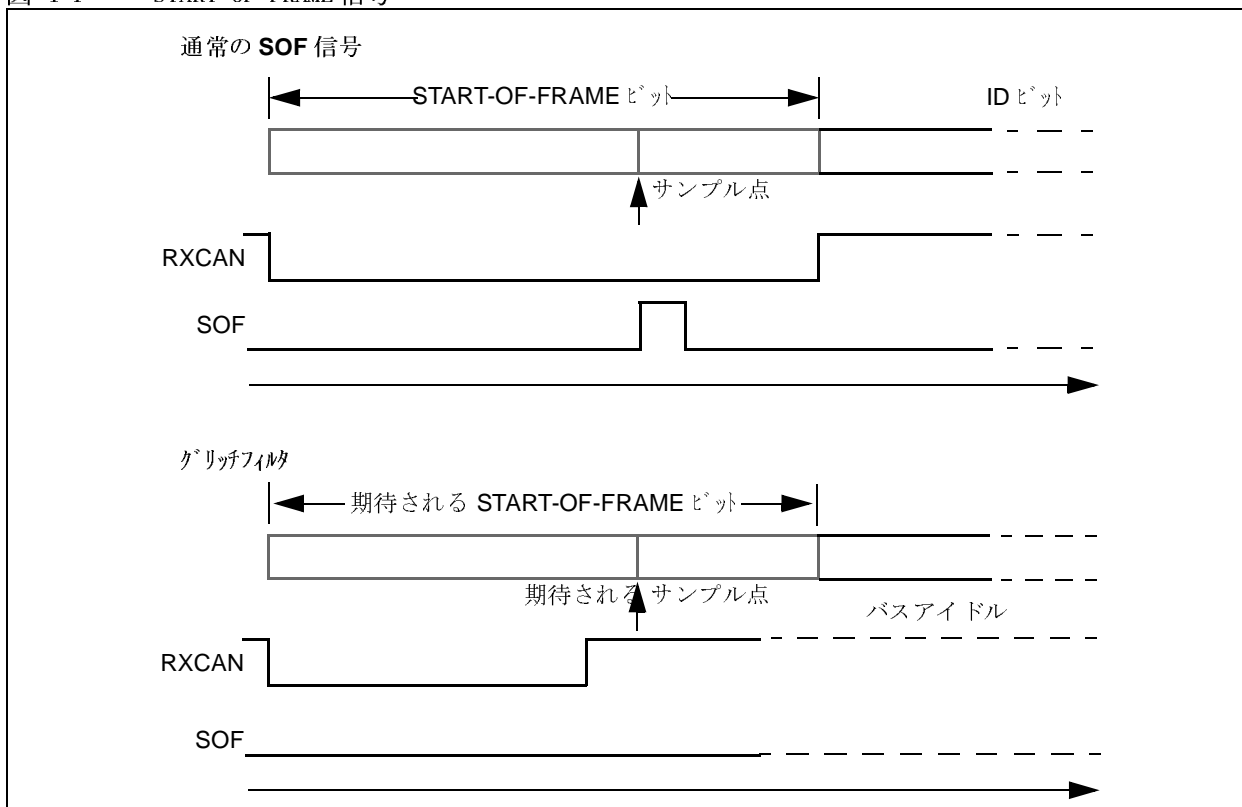
4.4.1 禁止

RXBnBF ピンは禁止されると BFPCTRL.BnBFE をクリアし、ハイインピーダンス状態となります。

4.4.2 バッファフルとして構成

RXBnBF ピンはバッファフル割り込みピンか、標準出力ピンとして構成することができます。これらのピンの構成制御と状態は BFPCTRL レジスタでできます。(レジスタ 4-3) (BFPCTRL.BxBFE と BFPCTRL.BxBFM ビットをセットすることで)割り込み動作モードとして設定されると、これらのビットはアクティブ Low となり、各受信バッファの CANINTF.RXnIF ビットにマッピングされます。受信バッファに対応するこのビットが High になると、(バッファに新たなメッセージが受信されたことを表す) 対応する RXBnBF ピンが Low となります。MCU により CANINTF.RXnIF ビットがクリアされると、対応する割り込みピンが論理 High 状態となり、次のメッセージが受信バッファにロードされるまで続きます。

図 4-1: START-OF-FRAME 信号



4.4.3 デジタル出力として構成

デジタル出力として使う場合には、関連するバッファの **BFPCTRL.BxBFM** をクリアし、**BFPCTRL.BnBFE** ビットをセットします。このモードでは、ピンの状態は、**BFPCTRL.BnBFS** ビットで制御されます。**BnBFS** ビットに '1' を書くと、対応するバッファフルピンが High レベルにドライブされ、'0' を書くと Low にドライブされます。このモードでピンを使うときには、バッファフルピンに起きうるグリッチを防止するため、ピンの状態はビット変更の **SPI** コマンドを使うことでしか変更できません。

表 4-1: RXNBF ピンの構成

BnBFE	BnBFM	BnBFS	ピン状態
0	X	X	禁止、ハイ impedance
1	1	X	受信バッファ割り込み
1	0	0	デジタル出力 = 0
1	0	1	デジタル出力 = 1

図 4-2: 受信バッファブロック図

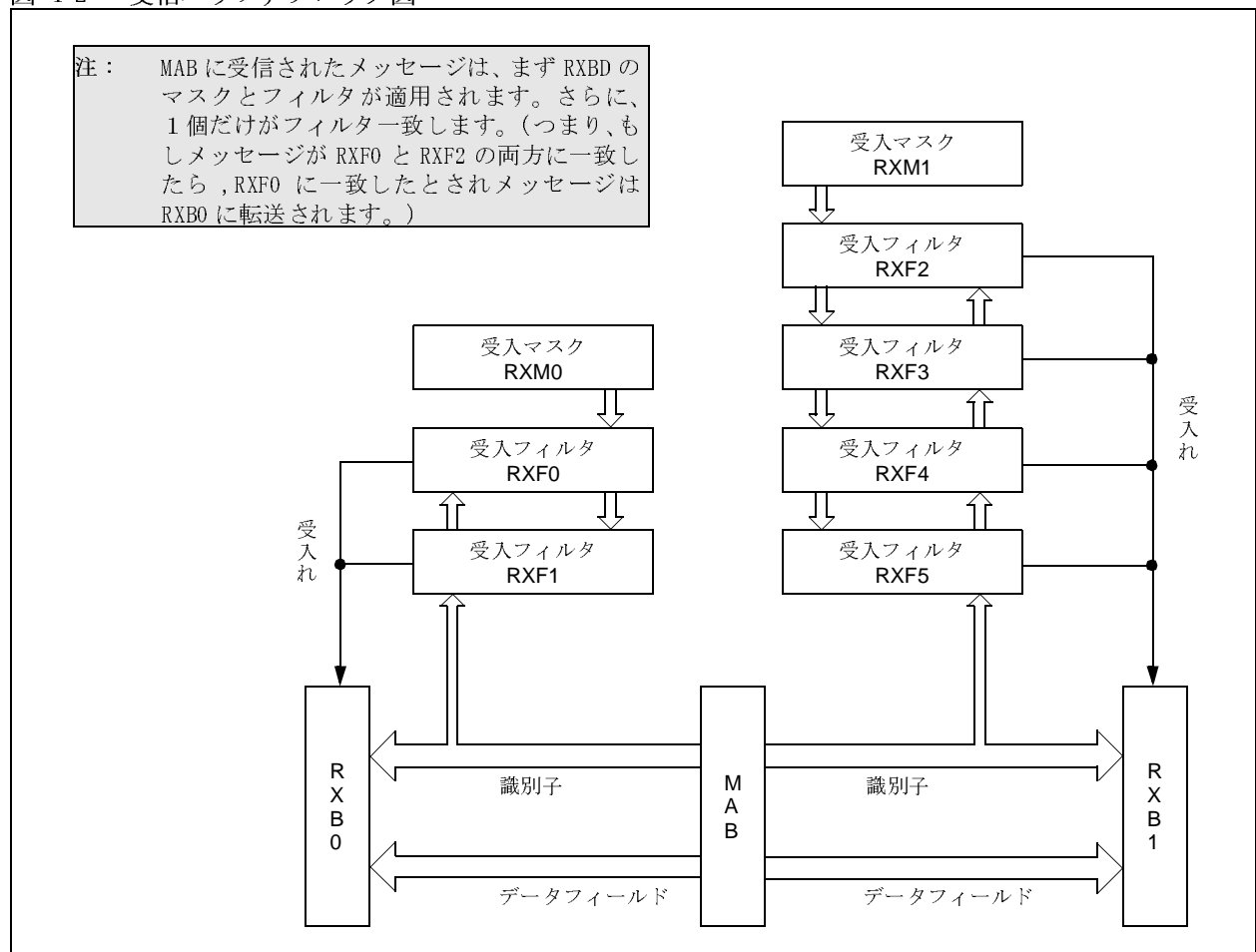
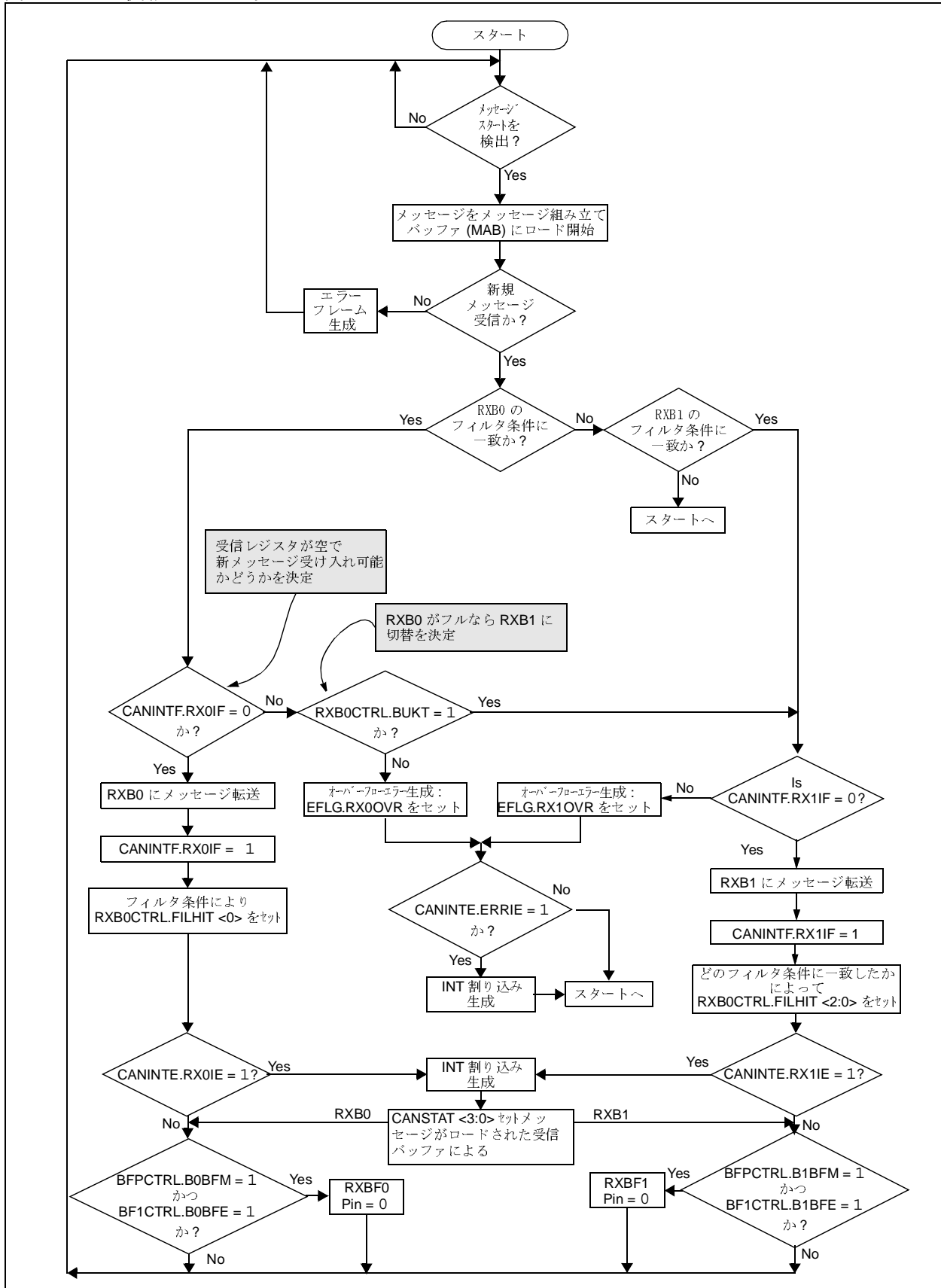


図 4-3: 受信フローチャート



レジスタ 4-1: RXB0CTRL - 受信バッファ 0 制御
(アドレス: 60h)

U-0	R/W-0	R/W-0	U-0	R-0	R/W-0	R-0	R-0
—	RXM1	RXM0	—	RXRTR	BUKT	BUKT1	FILHIT0

bit 7

bit 0

bit 7 未実装: 読むと '0'

bit 6-5 **RXM**: 受信バッファ動作モードビット

11 = マスク / フィルタがオフで、すべてのメッセージを受信

10 = 拡張識別子でフィルタに一致したメッセージだけ受信

01 = 標準識別子でフィルタに一致したメッセージだけ受信

00 = 標準識別子か拡張識別子を使っていて、フィルタに一致したすべてのメッセージを受信

bit 4 未実装: 読むと '0'

bit 3 **RXRTR**: リモート送信要求ビットを受信した

1 = リモート送信要求を受信した

0 = リモート送信要求は受信していない

bit 2 **BUKT**: 切替許可ビット

1 = RXB0 がフルなら RXB0 メッセージを RXB1 に切り替えて書き込む

0 = 切替禁止

bit 1 **BUKT1**: 読み取り専用 BUKT ビットと同じ (MCP2515 内部で使われる)

bit 0 **FILHIT**: フィルタ一致ビット - メッセージ受信がどの受入フィルタかを表す

1 = 受入フィルタ 1 (RXF1)

0 = 受入フィルタ 0 (RXF0)

注: RXB0 から RXB1 への切替が起きると、FILHIT ビットは切り替えられたメッセージを受け入れた側のフィルタを反映する。

凡例:

R = 読み出し可

W = 書き込み可

U = 未実装、読むと '0'

-n = POR 後の値

'1' = セット

'0' = クリア

x = 不定

MCP2515

レジスタ 4-2: RXB1CTRL – 受信バッファ 1 制御
(アドレス: 70h)

U-0	R/W-0	R/W-0	U-0	R-0	R-0	R-0	R-0
—	RXM1	RXM0	—	RXRTR	FILHIT2	FILHIT1	FILHIT0

bit 7

bit 0

bit 7 未実装: 読むと '0'

bit 6-5 **RXM:** 受信バッファ動作モードビット

11 = マスク / フィルタをオフ ; すべてのメッセージを受信

10 = フィルタと一致する拡張識別子をもつ有効メッセージだけ受信

01 = フィルタと一致する標準識別子をもつメッセージだけ受信

00 = フィルタと一致する標準または拡張識別子をもつすべてのメッセージを受信

bit 4 未実装: 読むと '0'

bit 3 **RXRTR:** リモート送信要求ビットの受信

1 = リモート送信要求を受信した

0 = リモート送信要求は受信していない

bit 2-0 **FILHIT:** フィルター一致ビット - メッセージ受信がどの受入フィルタかを示す

101 = 受入フィルタ 5 (RXF5)

100 = 受入フィルタ 4 (RXF4)

011 = 受入フィルタ 3 (RXF3)

010 = 受入フィルタ 2 (RXF2)

001 = 受入フィルタ 1 (RXF1) (RXB0CTRL 内の BUKT ビットがセットされるときのみ)

000 = 受入フィルタ 0 (RXF0) (RXB0CTRL 内の BUKT ビットがセットされているときのみ)

凡例:

R = 読み出し可

W = 書き込み可

U = 未実装、読むと '0'

-n = POR 後の値

'1' = セット

'0' = クリア

x = 不定

レジスタ 4-3: BFPCTRL -RXnBF ピン制御と状態
(アドレス: 0Ch)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	B1BFS	B0BFS	B1BFE	B0BFE	B1BFM	B0BFM

bit 7

bit 0

- bit 7 未実装: 読むと '0'
- bit 6 未実装: 読むと '0'
- bit 5 **B1BFS:** $\overline{RX1BF}$ ピン状態ビット (デジタル出力モードのときのみ)
- $\overline{RX1BF}$ が割り込みピンとして構成されているときは読むと '0'
- bit 4 **B0BFS:** $\overline{RX0BF}$ ピン状態ビット (デジタル出力モードのときのみ)
- $\overline{RX0BF}$ が割り込みピンとして構成されているときは読むと '0'
- bit 3 **B1BFE:** $\overline{RX1BF}$ ピン機能有効化ビット
1 = ピン機能有効、動作モードは **B1BFM** により決定する
0 = ピン機能無効、ピンはハイインピーダンス状態となる
- bit 2 **B0BFE:** $\overline{RX0BF}$ ピン機能有効化ビット
1 = ピン機能有効、動作モードは **B0BFM** により決定される
0 = ピン機能無効、ピンはハイインピーダンス状態となる
- bit 1 **B1BFM:** $\overline{RX1BF}$ ピン動作モードビット
1 = ピンを **RXB1** に有効メッセージ受信したときの割り込みピンとする
0 = デジタル出力モードとする
- bit 0 **B0BFM:** $\overline{RX0BF}$ ピン動作モードビット
1 = ピンを **RXB0** に有効メッセージ受信したときの割り込みピンとする
0 = デジタル出力モードとする

凡例:

R = 読み出し可	W = 書き込み可	U = 未実装、読むと '0'
-n =POR 後の値	'1' = セット	'0' = クリア x = 不定

レジスタ 4-4: RXBnSIDH - 受信バッファ n の標準識別子上位
(アドレス: 61h, 71h)

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3

bit 7

bit 0

- bit 7-0 **SID:** 標準識別子ビット <10:3>
これらのビットは、受信メッセージ用の標準識別子の上位8ビットを含む

凡例:

R = 読み出し可	W = 書き込み可	U = 未実装、読むと '0'
-n =POR 後の値	'1' = セット	'0' = クリア x = 不定

MCP2515

レジスタ 4-5: RXBnSIDL - 受信バッファ n の標準識別子 下位
(アドレス: 62h, 72h)

R-x	R-x	R-x	R-x	R-x	U-0	R-x	R-x
SID2	SID1	SID0	SRR	IDE	—	EID17	EID16
bit 7					bit 0		

bit 7-5 **SID:** 標準識別子ビット <2:0>

これらのビットは、受信メッセージ用の標準識別子の下位 3 ビットを含む

bit 4 **SRR:** 標準フレームのリモート送信要求ビット (IDE ビット = '0' のときだけ有効)

1 = 標準フレームのリモート送信要求を受信した

0 = 標準データフレームを受信した

bit 3 **IDE:** 拡張識別子フラグビット

このビットは、受信メッセージが標準フレームか拡張フレームかを示す

1 = 受信メッセージは拡張フレーム

0 = 受信メッセージは標準フレーム

bit 2 未実装: 読むと '0'

bit 1-0 **EID:** 拡張識別子ビット <17:16>

これらのビットは、受信メッセージの拡張識別子の上位 2 ビットを含む

凡例:

R = 読み出し可

W = 書き込み可

U = 未実装、読むと '0'

-n = POR 後の値

'1' = セット

'0' = クリア

x = 不定

レジスタ 4-6: RXBnEID8 - 受信バッファ n の拡張識別子 上位
(アドレス: 63h, 73h)

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8
bit 7					bit 0		

bit 7-0 **EID:** 拡張識別子ビット <15:8>

これらのビットは受信メッセージの拡張識別子の 15 から 8 ビットを保持

凡例:

R = 読み出し可

W = 書き込み可

U = 未実装、読むと '0'

-n = POR 後の値

'1' = セット

'0' = クリア

x = 不定

レジスタ 4-7: RXBnEID0 - 受信バッファ n の拡張識別子 下位
(アドレス: 64h, 74h)

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0
bit 7				bit 0			

bit 7-0 **EID**: 拡張識別子ビット <7:0>
これらのビットは受信メッセージの拡張識別子の下位 8 ビットを含む

凡例:			
R = 読み出し可	W = 書き込み可	U = 未実装、読むと '0'	
-n =POR 後の値	'1' = セット	'0' = クリア	x = 不定

レジスタ 4-8: RXBnDLC - 受信バッファ n のデータ長コード
(アドレス: 65h, 75h)

U-0	R-x	R-x	R-x	R-x	R-x	R-x	R-x
—	RTR	RB1	RB0	DLC3	DLC2	DLC1	DLC0
bit 7				bit 0			

bit 7 未実装: 読むと '0'

bit 6 **RTR**: 拡張フレームリモート送信要求ビット
(RXBnSIDL.IDE = '1' のときのみ有効)
1 = 拡張フレームリモート送信要求ビットを受信した
0 = 拡張データフレームを受信した

bit 5 **RB1**: 予約ビット 1

bit 4 **RB0**: 予約ビット 0

bit 3-0 **DLC**: データ長コードビット <3:0>
受信したデータのバイト数を示す

凡例:			
R = 読み出し可	W = 書き込み可	U = 未実装、読むと '0'	
-n =POR 後の値	'1' = セット	'0' = クリア	x = 不定

レジスタ 4-9: RXBnDM - 受信バッファ n のデータバイト M
(アドレス: 66h - 6Dh, 76h - 7Dh)

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
RBnDm7	RBnDm6	RBnDm5	RBnDm4	RBnDm3	RBnDm2	RBnDm1	RBnDm0
bit 7				bit 0			

bit 7-0 **RBnDm7:RBnDm0**: 受信バッファ n のデータフィールド m バイト目
受信メッセージのデータは 8 バイトを含む

凡例:			
R = 読み出し可	W = 書き込み可	U = 未実装、読むと '0'	
-n =POR 後の値	'1' = セット	'0' = クリア	x = 不定

4.5 メッセージ受入フィルタとマスク

メッセージ受入フィルタとマスクは、メッセージ組み立てバッファ内のメッセージがどちらかの受信バッファに転送すべきかどうかを決定するために使用されます。(図 4-5 参照) 有効なメッセージが MAB に受信されると、メッセージの識別子フィールドがフィルタ値と比較されます。一致すると、そのメッセージは対応する受信バッファに転送されます。

4.5.1 データバイトフィルタ

標準データフレーム (11 ビットの識別子) を受信したときは、MCP2515 は自動的にデータフィールドの最初の 16 ビットの拡張識別子部分に対して、通常の 16 ビットのマスクとフィルタを適用します。(データバイト 0 と 1 として) 図 4-4 にマスクとフィルタが標準データフレームを拡張データフレームに対しどのように適応されるかを示しています。

データバイトをフィルタリングすることで、最初のデータバイトをフィルタする上位層プロトコル (HLP)(例えば DeviceNet™) が実装された場合の MCU の負荷を減らします。

4.5.2 フィルタ一致

フィルタとマスク (レジスタ 4-14 からレジスタ 4-17 を参照) は、識別子のどのビットにフィルタを適用するかを決定します。表 4-2 に示す真理値表は、識別子の各ビットが、どのようにマスクされフィルタと比較されて、受信バッファに格納されるかを示しています。マスクは本来どのビットにフィルタを適用するかを決定します。マスクビットがゼロならそのビットはフィルタビットに関わらず、自動的に受入れます

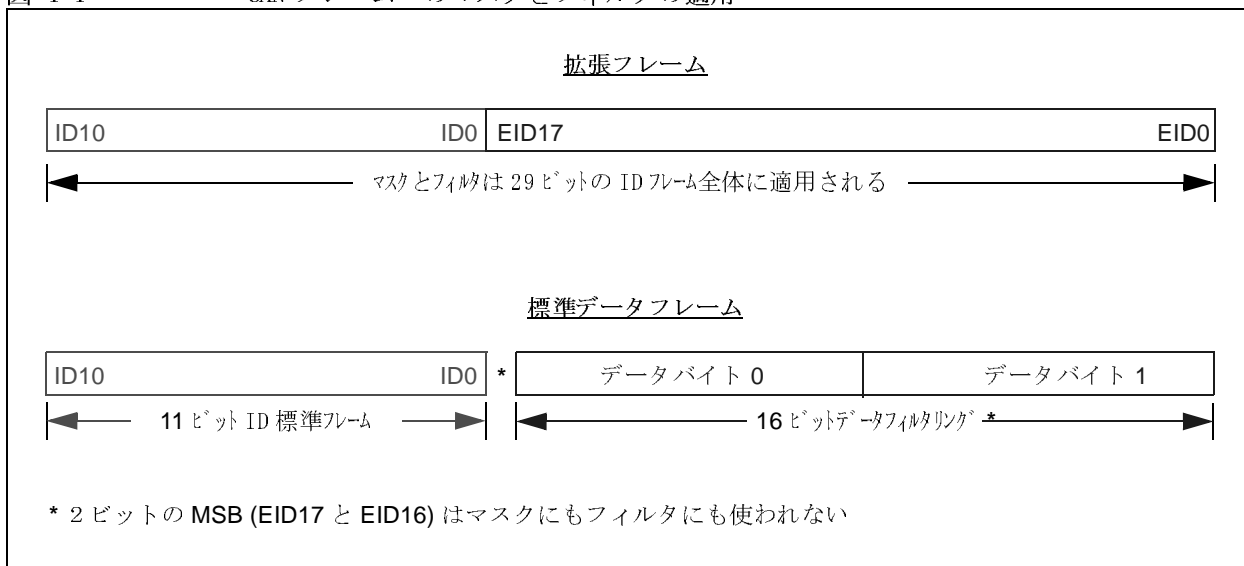
表 4-2: フィルタ / マスクの真理値表

マスクビット n	フィルタビット n	メッセージ識別子ビット	受入, 拒否 n
0	X	X	受入
1	0	0	受入
1	0	1	拒否
1	1	0	拒否
1	1	1	受入

注: X = 無関係

受信バッファブロック図 (図 4-2) に示したように、受入フィルタ RXF0 と RXF1(そしてフィルタマスク RXM0) は、RXB0 に関連します。RXF2, RXF3, RXF4, RXF5 とマスク RXM1 は RXB1 に関連します。

図 4-4: CAN フレームへのマスクとフィルタの適用



4.5.3 フィルタ一致ビット

受信メッセージのフィルタ一致がどれのフィルタかは、対応する $RXBnCTRL$ レジスタの $FILHIT$ ビットで求められます。 $RXB0CTRL.FILHIT0$ はバッファ 0 に、 $RXB1CTRL.FILHIT<2:0>$ はバッファ 1 に対応します。

受信バッファ 1 ($RXB1$) の 3 ビットの $FILHIT$ ビットは、下記のようにコード化されます。:

- 101 = 受入フィルタ 5 ($RXF5$)
- 100 = 受入フィルタ 4 ($RXF4$)
- 011 = 受入フィルタ 3 ($RXF3$)
- 010 = 受入フィルタ 2 ($RXF2$)
- 001 = 受入フィルタ 1 ($RXF1$)
- 000 = 受入フィルタ 0 ($RXF0$)

注: 000 と 001 は $RXB0CTRL$ の $BUKT$ ビットがセットされて、 $RXB0$ のメッセージが $RXB1$ に転送することが許可されているときだけ発生します。

$RXB0CTRL$ には 2 つの $BUKT$ ビットと $FILHIT<0>$ ビットが含まれています。

$BUKT$ ビットのコード化は $RXB1CTRL.FILHIT$ と同じ 3 ビットで行われ、 $RXF0$ と $RXF1$ フィルタの一致が、 $RXB0$ か $RXB1$ に転送された後かを区別します。

- 111 = 受入フィルタ 1 ($RXB1$)
- 110 = 受入フィルタ 0 ($RXB1$)
- 001 = 受入フィルタ 1 ($RXB0$)
- 000 = 受入フィルタ 0 ($RXB0$)

$BUKT$ がクリアされているときは、6 個のフィルタに対応する 6 通りのコードとなります。 $BUKT$ がセットされているときは、6 個のフィルタに対応する 6 個のコードに加えて $RXB1$ に転送後の $RXF0$ と $RXF1$ という 2 通りのコードが追加されます。

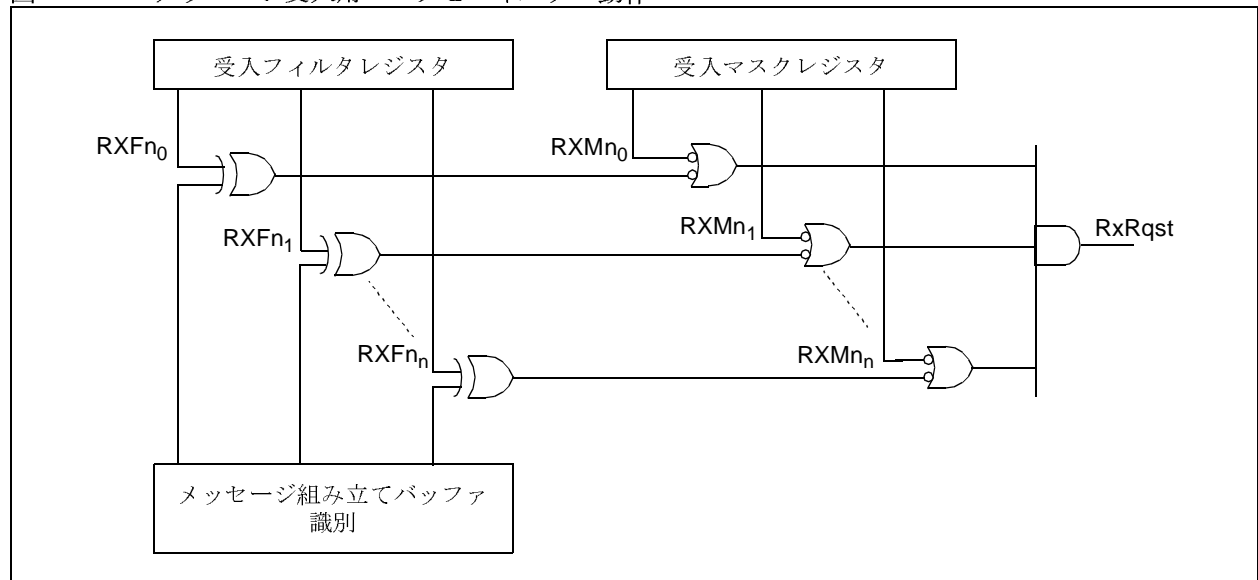
4.5.4 複数フィルタ一致

1 個以上の受入フィルタで一致したときは、 $FILHIT$ ビットが一致した中の最小の番号のフィルタのバイナリ値にコード化されます。例えば、 $RXF2$ フィルタと $RXF4$ フィルタで一致したとすると、 $FILHIT$ ビットには、 $RXF2$ の値がロードされます。つまり受入フィルタはもともと下位番号が高い優先順位をもつように順位づけされていることとなります。メッセージはフィルタ番号順のフィルタと比較されることとなります。これはまた、メッセージがただひとつのバッファにしか受信されないことを保証します。つまり $RXB0$ は $RXB1$ より高い優先順位を持っていることとなります。

4.5.5 マスクとフィルタの構成

マスクとフィルタレジスタは、 $MCP2515$ がコンフィギュレーションモードのときだけ変更が可能です。(第 10.0 項 “動作モード” を参照)

図 4-5: メッセージ受入用マスクとフィルタの動作



MCP2515

レジスタ 4-10: RXFnSIDH – フィルタ n の標準識別子 上位
(アドレス: 00h, 04h, 08h, 10h, 14h, 18h)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3
bit 7				bit 0			

bit 7-0 **SID:** 標準識別子フィルタビット <10:3>
これらのビットには受信メッセージの標準識別子部分のビット <10:3> に適用するフィルタビットを保持しています。

凡例:			
R = 読み出し可	W = 書き込み可	U = 未実装、読むと '0'	
-n = POR 後の値	'1' = セット	'0' = クリア	x = 不定

レジスタ 4-11: RXFnSIDL – フィルタ n の標準識別子 下位
(アドレス: 01h, 05h, 09h, 11h, 15h, 19h)

R/W-x	R/W-x	R/W-x	U-0	R/W-x	U-0	R/W-x	R/W-x
SID2	SID1	SID0	—	EXIDE	—	EID17	EID16
bit 7				bit 0			

bit 7-5 **SID:** 標準識別子フィルタビット <2:0>
これらのビットには受信メッセージの標準識別子部分のビット <2:0> に適用するフィルタビットを保持しています。

bit 4 未実装: 読むと '0'

bit 3 **EXIDE:** 拡張識別子有効化ビット
1 = フィルタは拡張フレームにだけ適用される
0 = フィルタは標準フレームにだけ適用される

bit 2 未実装: 読むと '0'

bit 1-0 **EID:** 拡張識別子ビット <17:16>
これらのビットには受信メッセージの拡張識別子部分のビット <17:16> に適用するフィルタビットを保持しています。

凡例:			
R = 読み出し可	W = 書き込み可	U = 未実装、読むと '0'	
-n = POR 後の値	'1' = セット	'0' = クリア	x = 不定

レジスタ 4-12: RXFnEID8 – フィルタ n の拡張識別子 上位
(アドレス: 02h, 06h, 0Ah, 12h, 16h, 1Ah)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8

bit 7

bit 0

bit 7-0 **EID:** 拡張識別子ビット <15:8>

これらのビットには受信メッセージの拡張識別子部分のビット <15:8> に適用するフィルタビットを保持しています。

凡例:

R = 読み出し可 W = 書き込み可 U = 未実装、読むと '0'
-n =POR 後の値 '1' = セット '0' = クリア x = 不定

レジスタ 4-13: RXFnEID0 – フィルタ n の拡張識別子 下位
(アドレス: 03h, 07h, 0Bh, 13h, 17h, 1Bh)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0

bit 7

bit 0

bit 7-0 **EID:** 拡張識別子ビット <7:0>

これらのビットには受信メッセージの拡張識別子部分のビット <7:0> に適用するフィルタビットを保持しています。

凡例:

R = 読み出し可 W = 書き込み可 U = 未実装、読むと '0'
-n =POR 後の値 '1' = セット '0' = クリア x = 不定

レジスタ 4-14: RXMnSIDH – マスク n の標準識別子 上位
(アドレス: 20h, 24h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3

bit 7

bit 0

bit 7-0 **SID:** 標準識別子マスクビット <10:3>

これらのビットには受信メッセージの標準識別子部分のビット <10:3> に適用するマスクビットを保持しています。

凡例:

R = 読み出し可 W = 書き込み可 U = 未実装、読むと '0'
-n =POR 後の値 '1' = セット '0' = クリア x = 不定

MCP2515

レジスタ 4-15: RXMnSIDL – マスク n の標準識別子 下位
(アドレス: 21h, 25h)

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
SID2	SID1	SID0	—	—	—	EID17	EID16
bit 7						bit 0	

bit 7-5 **SID:** 標準識別子マスクビット <2:0>

これらのビットには受信メッセージの標準識別子部分のビット <2:0> に適用するマスクビットを保持しています。

bit 4-2 未実装: 読むと '0'

bit 1-0 **EID:** 拡張識別子マスクビット <17:16>

これらのビットには受信メッセージの拡張識別子部分のビット <17:16> に適用するマスクビットを保持しています。

凡例:

R = 読み出し可	W = 書き込み可	U = 未実装、読むと '0'
-n =POR 後の値	'1' = セット	'0' = クリア x = 不定

レジスタ 4-16: RXMnEID8 – マスク n の拡張識別子 上位
(アドレス: 22h, 26h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8
bit 7						bit 0	

bit 7-0 **EID:** 拡張識別子マスクビット <15:8>

これらのビットには受信メッセージの拡張識別子部分のビット <15:8> に適用するマスクビットを保持しています。

凡例:

R = 読み出し可	W = 書き込み可	U = 未実装、読むと '0'
-n =POR 後の値	'1' = セット	'0' = クリア x = 不定

レジスタ 4-17: RXMnEID0 – マスク n 拡張識別子 下位
(ADDRESS: 23h, 27h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0
bit 7						bit 0	

bit 7-0 **EID:** 拡張識別子マスクビット <7:0>

これらのビットには受信メッセージの拡張識別子部分のビット <7:0> に適用するマスクビットを保持しています。

凡例:

R = 読み出し可	W = 書き込み可	U = 未実装、読むと '0'
-n =POR 後の値	'1' = セット	'0' = クリア x = 不定

5.0 ビットタイミング

ある CAN バス上のすべてのノードは同じ公称ビットレートを持つ必要があります。CAN プロトコルは、データストリーム中でクロックがエンコードできない非ゼロ復帰 (NRZ) コーディングを使用します。従って、受信クロックは受信側ノードにより回復され、送信クロックに同期させる必要があります。

発振器と送信時間はノード毎に変動するので、受信側はある種の位相ロックループ (PLL) を持ち、データ送信エッジと同期を取って受信クロックを同期化、保持する必要があります。データは NRZ コードなので、エッジが少なくとも 6 ビット回数発生するごとに、デジタルフェーズロックループ (DPLL) の同期を保持することを確実にするためにビットスタフィングを含める必要があります。

MCP2515 のビットタイミングには、到着データに同期を取る構成の DPLL が実装されていて、送信データが提供する公称タイミングと同じになるようになっています。DPLL は各ビットタイムを、時間単位 (TQ) と呼ばれる最小の周期からなる複数のセグメントに分割します。

ビットタイミングフレームの中で実行されるバスタイミング機能 (自己の発振器への同期とか、ネットワーク遅延補償とか、サンプル点の設定など) は、プログラマブルな DPLL のビットタイミングロジックによる定義されます。

5.1 CAN のビットタイム

CAN バス上のすべてのデバイスが同じビットレートでなければなりません。しかし、すべてのデバイスが同じマスタ発振クロック周波数から供給される必要はありません。個々のデバイスごとのクロック周波数の差異のため、ボーレート用分周器の設定と各セグメントの時間単位の数を調整することでビットレートを合わせなければなりません。

CAN のビットタイムは、重複しないセグメントで作られます。この各セグメントは、本データシートの後で解説される時間単位 (TQ) と呼ばれる積算単位から作られます。公称ビットレート (NBR) は、CAN の仕様で、再同期のない理想的な送信器によって 1 秒間に送信されるビット数として定義されています。これは下記式で表されます。:

式 5-1:

$$NBR = f_{bit} = \frac{1}{t_{bit}}$$

公称ビットタイム

公称ビットタイム (NBT) (t_{bit}) は、重複のないセグメントから作られます。(図 5-1) したがって NBT は、下記セグメントの集合となります。:

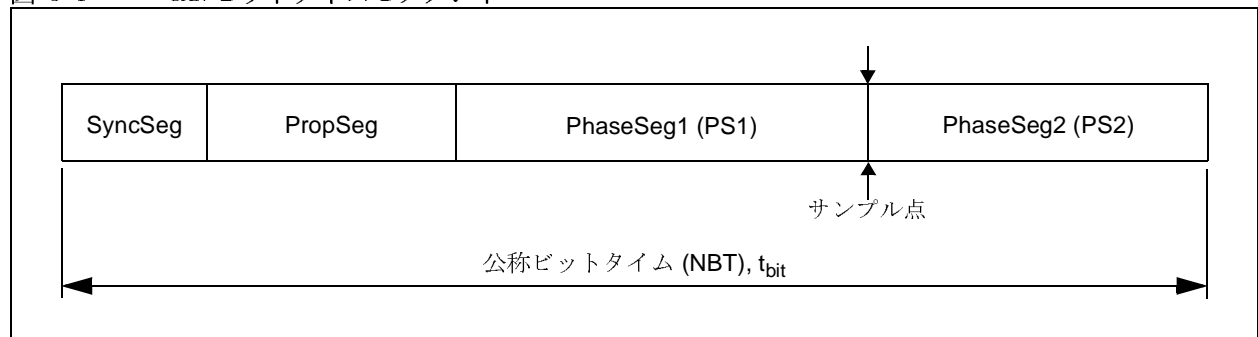
$$t_{bit} = t_{SyncSeg} + t_{PropSeg} + t_{PS1} + t_{PS2}$$

NBT に関連してサンプル点があります。同期ジャンプ幅 (SJW) と情報処理時間 (IPT) は後で説明します。

同期セグメント

同期セグメント (SyncSeg) は、NBT の最初のセグメントで、バス上のノードの同期を取るために使われず。ビットエッジが SyncSeg 内で現れることが期待されます。このセグメントは 1 TQ で固定です。

図 5-1: CAN ビットタイムセグメント



MCP2515

伝播セグメント

伝播セグメント (PropSeg) は、ネットワーク内の物理遅延時間を補正するためにあります。これらの伝播遅延は、対応するバスドライバの遅延を含めたバスライン上の合計信号伝播時間の2倍として定義されます。PropSeg は 1 ~ 8 TQ の間でプログラマブルです。

フェーズセグメント 1 (PS1) とフェーズセグメント 2 (PS2)

2つのフェーズセグメント PS1 と PS2 は、バス上のエッジ位相誤差を補正するために使われます。PS1 は再同期によって伸ばす (または PS2 を縮める) ことができます。PS1 は 1 ~ 8 TQ の範囲でプログラマブルで、PS2 は、2 ~ 8 TQ でプログラマブルです。

サンプル点

サンプル点は、ビットタイム内でロジックレベルが読み込まれ解釈される位置です。サンプル点は、PS1 の終わりに置かれます。この規則の例外は、サンプルモードがビット当たり 3 回のサンプリングと構成されている場合です。この場合には、ビットはやはり PS1 の終わりでサンプリングされますが、さらに追加の 2 回のサンプリングが PS1 の終わりから 1.5TQ 周期後に行われます。そしてビット値は多数決で決定されます。

情報処理時間

情報処理時間 (IPT) は、サンプルしたビットのビットレベルを決めるのに必要な時間です。IPT はサンプル点で開始され、TQ で計測され、マイクロチップの CAN モジュールでは 2 TQ で固定です。また、PS2 がサ

ンプル点で開始され、ビットタイムの最後のセグメントまでとなるので、PS2 の最小値は、IPT より短くなくてはなりません。

したがって:

$$PS2_{min} = IPT = 2TQ$$

同期ジャンプ幅

同期ジャンプ幅 (SJW) により、送信メッセージとの同期を保つため、1 ~ 4 TQ 単位 (設定可能) でビットクロックを調整します。同期についてのさらなる詳細は本データシートで後述します。

時間単位

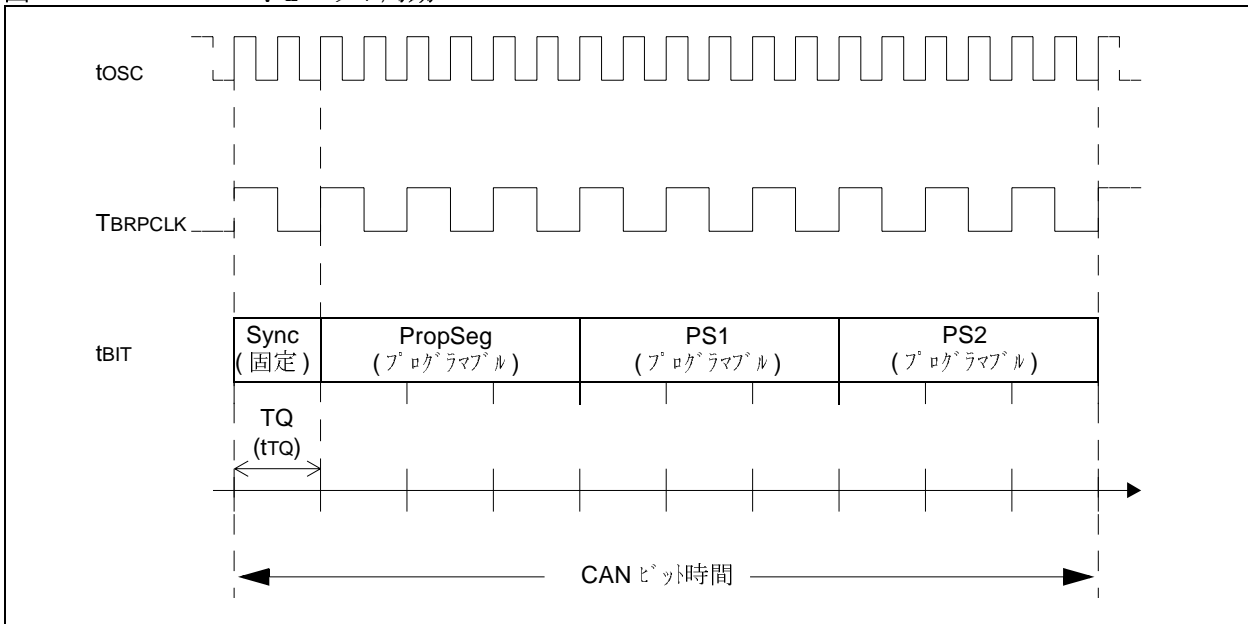
ビットタイムを作るセグメントのそれぞれは、時間単位 (TQ) と呼ばれる積算単位で作られています。各時間単位の長さは発振器周期 (t_{OSC}) に基づいています。基本の TQ は、発振器周期の 2 倍です。図 5-2 に T_{OSC} と TQ から導かれるビット周期を示しています。TQ の長さは、TQ の 1 クロック周期 (t_{BRPCLK}) に等しくなっていて、ポーレート分周器 (BRP) と呼ばれる分周器によりプログラマブルとなっています。これは下記式で表されます。:

式 5-2:

$$TQ = 2 \cdot BRP \cdot T_{OSC} = \frac{2 \cdot BRP}{F_{OSC}}$$

ここで: BRP は レジスタ 5-1 に示された構成と同じとします。

図 5-2: TQ とビット周期



5.2 同期化

バス上の各ノードの発振周波数間の位相ずれを補正するために、CAN コントローラーは入力信号の適切な信号エッジに同期させる必要があります。同期化は DPLL 機能が実装されていることにより可能なプロセスです。

送信データ内のエッジが検出されると、ロジックは期待される時間 (SyncSeg) のエッジの位置との比較をします。それから回路は PS1 と PS2 の値を必要であれば調整します。同期を取るために使用される 2 つの機構があります。:

1. ハード同期
2. 再同期

5.2.1 ハード同期

ハード同期は、BUS IDLE 状態中にリセッシブからドミナントのエッジがあるとき、これはメッセージの開始を示しますが、このときだけ行われます。ハード同期の後、ビットタイムカウンタは、SyncSeg とともに再スタートされます。

ハード同期は、再スタートビットタイムの同期化セグメント内にエッジがあるようにします。同期化のルールにより、ハード同期が取られた場合、そのビットタイム内では、再同期は行われません。

5.2.2 再同期

再同期の結果、PS1 が伸びたり、もしくは PS2 が縮んだりするかもしれませんが、フェーズバッファセグメントを伸び縮みさせる量は、再同期ジャンプ幅ビット (SJW) が上限として与えられます。

SJW の値は PS1 に追加されるかもしくは PS2 から差し引かれます。(図 5-3 参照) SJW の値は DPLL のループフィルタとして現れます。SJW は 1 TQ から 4 TQ の間でプログラマブルです。

5.2.2.1 位相誤差

NRZ ビットコーディング方式では、メッセージ中にはクロックがエンコードされません。クロック情報はリセッシブからドミナントへの遷移だけから引き出せます。連続するビットが同じ値を持つのは固定数のみという性質 (ビットスタッフィング) により、フレーム期間中に確実にビットストリームに再同期化させます。

エッジのフェーズエラーは、SyncSeg に対するエッジの位置により与えられ、時間単位で計測されます。フェーズエラーは、以下のように TQ の大きさを定義されます。:

- $e = 0$ エッジが同期セグメント以内にある場合
- $e > 0$ がサンプル点より前にある場合 (TQ が PS1 に加算される)
- $e < 0$ エッジが、前ビットのサンプル点より後にある場合 (TQ が PS2 から減算される)

5.2.2.2 フェーズエラーなし ($e = 0$)

フェーズエラーが SJW としてプログラムされた値以下の場合、再同期の効果はハード同期の効果と同じです。

5.2.2.3 正のフェーズエラー ($e > 0$)

フェーズエラーが SJW より大きい場合およびフェーズエラーが正の場合、PS1 は、SJW と同じ量だけ長くなります

5.2.2.4 負のフェーズエラー ($e < 0$)

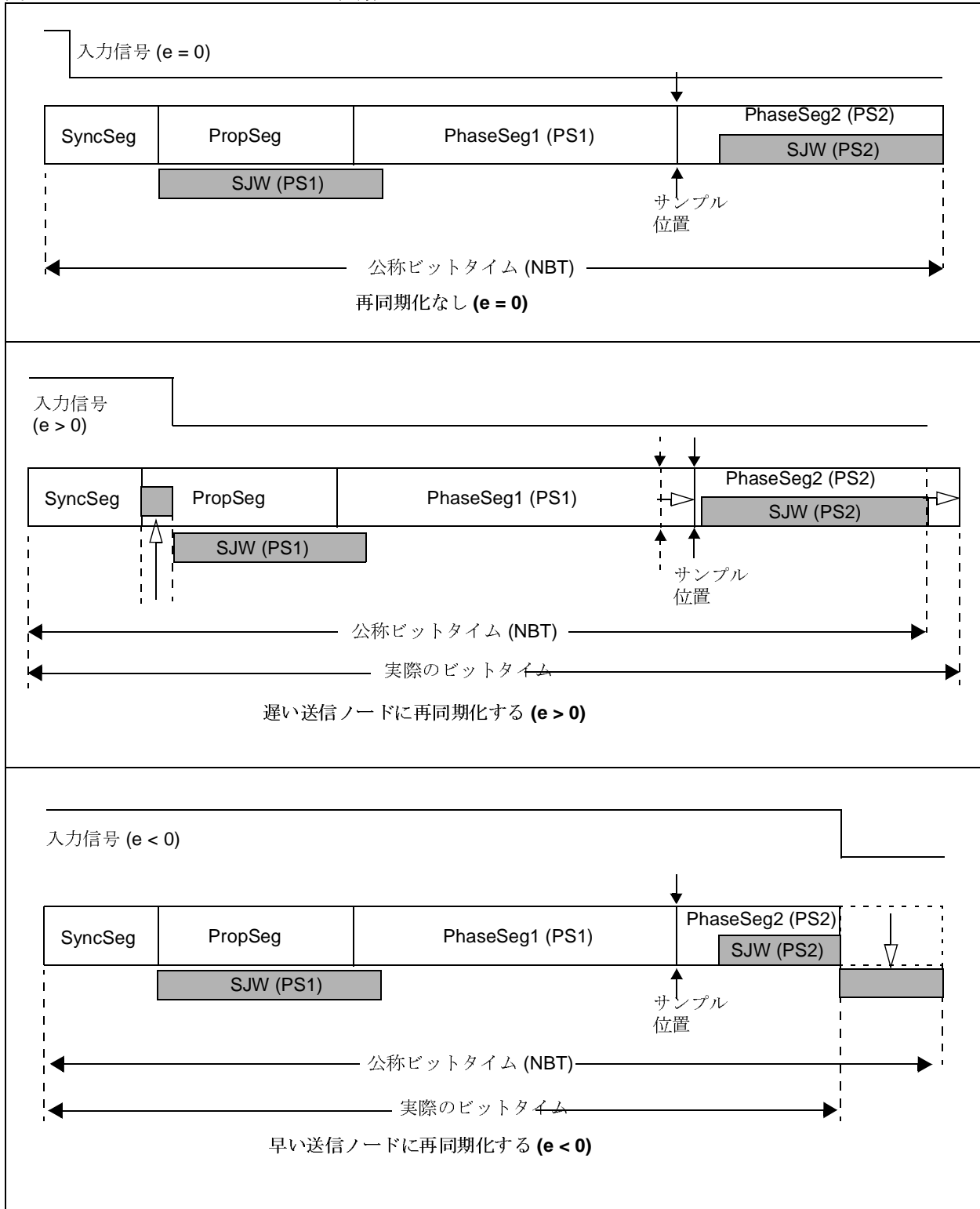
フェーズエラーが再同期ジャンプ幅より大きい場合およびフェーズエラーが負の場合、PS2 は、SJW と同じ量だけ短くなります。

5.2.3 同期化の規則

1. リセッシブからドミナントへのエッジだけが同期に使われる。
2. 1 ビットタイム中には 1 回だけの同期化しかできない。
3. エッジが同期化に使われるのは、前回のサンプル点で検出された値 (前回のバス読み込み値) が、エッジ直後のバスの値と異なっているときだけです。
4. 送信ノードは、正のフェーズエラーのとき ($e > 0$) は、再同期化はしません。
5. フェーズエラーの絶対値が SJW より大きいときは、適当なフェーズセグメントで SJW と同じ大きさになるように調整します。

MCP2515

図 5-3: ビットタイムの同期



5.3 タイムセグメントのプログラミング

タイムセグメントのプログラミングには、以下のよう
にいくつかの要求事項があります。:

- PropSeg + PS1 >= PS2
- PropSeg + PS1 >= TDELAY
- PS2 > SJW

例えば、CAN ボーレートが 125 kHz で Fosc = 20 MHz
としたいとすると:

Tosc = 50 ns なので BRP<5:0> = 04h を選択すれば
Tq = 500 ns です。125 kHzを得るためにはビットタイ
ムは 16 Tq としなければなりません。

通常、ビットのサンプリングは、システムパラメー
タによりますが、ビットタイムの約 60-70% のところ
で行われます。また、通常 TDELAY は 1 ~ 2 Tq です。

SyncSeg = 1 Tq で PropSeg = 2 Tq です。これで
PS1 = 7 Tq と設定すれば、遷移から 10 Tq 後にサンプ
ル点が来ます。ここは PS2 から 6 Tq 離れています。

PS2 が 6 では、規則にしたがって、SJW の最大値は 4
Tq となります。しかし、大きな SJW は通常異なるノー
ドのクロックがセラミック発振子などの不正確で不安
定な時しか必要としません。そこで SJW は通常 1 とす
れば十分です。

5.4 発振器の許容度

ビットタイミング要求から、セラミック発振子は送
信レートが大雑把に 125 kbit/sec までの用途のときに
許容されます。CAN プロトコルのフルバス速度範囲で
は、水晶発振器が必要です。最大のノードとノード間
の発振ずれは 1.7% が許容範囲です。

5.5 ビットタイミング設定レジスタ

コンフィギュレーションレジスタ (CNF1, CNF2,
CNF3) が CAN バスインターフェースのビットタイミ
ングを制御します。これらのレジスタは MCP2515 が
コンフィギュレーションモードのときだけ変更が可能
です。(第 10.0 項 “動作モード” を参照)

5.5.1 CNF1

BRP<5:0> ビットは、ボーレート分周器の制御をし
ます。これらのビットにより OSC1 の入力周波数の相
対値で、Tq の最小値を 2 Tosc (BRP<5:0> = 'b000000'
のとき) として Tq の長さを設定します。SJW<1:0>
ビットは Tq の数によって SJW を選択します。.

5.5.2 CNF2

PRSEG<2:0> ビットは伝播セグメントの長さを (Tq
単位で) 設定します。PHSEG1<2:0> ビットは PS1 の
長さを (Tq 単位で) 設定します。

SAM ビットは、RXCAN ピンを何回サンプリングす
るかを制御します。このビットを '1' にするとバスを
3 回サンプリングします。: サンプル点の前に Tq/2 ご
とに 2 回、そして公称サンプル点 (ここは PS1 の終わ
り) で 1 回サンプリングします。バスの値は、サンプ
リングの多数決で決定されます。SAM ビットを '0' に
設定すると、RXCAN ピンはサンプル点で 1 回だけの
サンプリングとなります。

BTLMODE ビットは PS2 の長さを決めます。この
ビットが '1' に設定されると、PS2 の長さは CNF3 の
PHSEG2<2:0> ビットで決定されます。(第 5.5.3 項
“CNF3” を参照) BTLMODE ビットが '0' に設定され
ると、PS2 の長さは PS1 や情報処理時間 (MCP2515
では、これは 2 Tq に固定となっています。) のそれよ
り長くなります。

5.5.3 CNF3

CNF2.BTLMODE ビットが '1' に設定されると、
PHSEG2<2:0> ビットで PS2 の長さを (Tq 単位で) 設定
します。BTLMODE ビットが '0' に設定されると
PHSEG2<2:0> ビットは何の影響も与えません。

MCP2515

レジスタ 5-1: CNF1 - コンフィギュレーション1 (アドレス: 2Ah)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SJW1	SJW0	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0

bit 7 bit 0

bit 7-6 **SJW**: 再同期ジャンプ幅長ビット <1:0>

11 = 長さ = 4 x TQ

10 = 長さ = 3 x TQ

01 = 長さ = 2 x TQ

00 = 長さ = 1 x TQ

bit 5-0 **BRP**: ポーレート分周器ビット <5:0>

$TQ = 2 \times (BRP + 1) / FOSC$

凡例:

R = 読み出し可

W = 書き込み可

U = 未実装、読むと '0'

-n = POR 後の値

'1' = セット

'0' = クリア

x = 不定

レジスタ 5-2: CNF2 - コンフィギュレーション 1 (アドレス: 29h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BTLMODE	SAM	PHSEG12	PHSEG11	PHSEG10	PRSEG2	PRSEG1	PRSEG0
bit 7						bit 0	

- bit 7 **BTLMODE:** PS2 ビットタイム長ビット
 1 = PS2 の長さは CNF3 の PHSEG22:PHSEG20 ビットで決定される
 0 = PS2 の長さは PS1 および IPT (2 Tq) より大きくする
- bit 6 **SAM:** サンプル点コンフィギュレーションビット
 1 = バスラインはサンプル点で 3 回サンプリングされる
 0 = バスラインはサンプル点で 1 回だけサンプリングされる
- bit 5-3 **PHSEG1:** PS1 長さビット <2:0>
 (PHSEG1 + 1) x Tq
- bit 2-0 **PRSEG:** 伝播セグメント長さビット <2:0>
 (PRSEG + 1) x Tq

凡例:			
R = 読み出し可	W = 書き込み可	U = 未実装、読むと '0'	
-n = POR 後の値	'1' = セット	'0' = クリア	x = 不定

MCP2515

レジスタ 5-3: CNF3 - コンフィギュレーション1 (アドレス: 28h)

R/W-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
SOF	WAKFIL	—	—	—	PHSEG22	PHSEG21	PHSEG20
bit 7					bit 0		

bit 7 **SOF:** スタートオブフレーム信号ビット
 CANCTRL.CLKEN = 1 のとき:
 1 = CLKOUT ピンは SOF 信号として有効化される
 0 = CLKOUT ピンはクロック出力機能として有効化される
 CANCTRL.CLKEN = 0 のとき、ビットは無影響

bit 6 **WAKFIL:** ウェイクアップフィルタビット
 1 = ウェイクアップフィルタ有効
 0 = ウェイクアップフィルタ無効

bit 5-3 **未実装: 読むと '0'**

bit 2-0 **PHSEG2:** PS2 長さビット <2:0>
 (PHSEG2 + 1) x Tq

Note: PS2 の設定最小値は 2 Tq

凡例:

R = 読み出し可	W = 書き込み可	U = 未実装、読むと '0'
-n = POR 後の値	'1' = セット	'0' = クリア x = 不定

NOTES:

6.0 エラー検出

CAN プロトコルは洗練されたエラー検出メカニズムを提供しています。下記のようなエラーが検出できます。

6.1 CRC エラー

巡回冗長チェック (CRC) により、送信器は、フレームの開始からデータフィールドの終わりまでのビットシーケンスに対して特別なチェックビットを計算します。この CRC シーケンスは、CRC フィールドで送信されます。受信ノードでも、同じ式を用いて CRC シーケンスを計算し、受信シーケンスとの比較を行います。不一致が検出されると、CRC エラーが発生しエラーフレームが生成されます。メッセージは再送されます。

6.2 アクノレッジエラー

メッセージのアクノレッジフィールドでは、送信器がアクノレッジスロット (受信器がリセッパビットを送っているはず) がドミナントでないかをチェックします。もしドミナントであれば、どのノードもフレームを正常に受信していないこととなります。アクノレッジエラーが起きると、エラーフレームが生成され、メッセージが再送されます。

6.3 形式エラー

ノードが 4 セグメント内 (エンドオブフレーム、フレーム間の空き、アクノレッジデリミタ、CRC デリミタを含む) にドミナントビットを 1 つでも検出すると、形式エラー発生となり、エラーフレームが生成されます。メッセージは再送されます。

6.4 ビットエラー

送信器が送信したビットレベルと反対のレベル (つまり、ドミナントビット送信したときリセッパを検出、あるいはリセッパを送信したときドミナントを検出した) を検出するとビットエラー発生となります。

例外: アービトレーションフィールドのアクノレッジスロットの間に、送信器がリセッパを送信してドミナントを検出したときは、ビットエラーにはなりません。なぜなら正常なアービトレーションが行われているからです。

6.5 スタップエラー

フレームの開始から CRC デリミッタ間で、同じ極性を持つ 6 つの連続するビットが検出された場合、ビットスタッピングルールに違反することになります。ビットスタッピングエラーが発生しエラーフレームが生成されます。メッセージは再送されます。

6.6 エラー状態

検出されたエラーはエラーフレームのよってすべてのノードに通知されます。エラーのメッセージの送信はアボートされ、フレームはできるだけ早期に再送されます。さらに、各 CAN ノードは内部エラーカウンタの値により 3 つの状態のいずれかとなります。:

1. エラーアクティブ
2. エラーパッシブ
3. バスオフ (送信器のみ)

エラーアクティブ状態は通常の状態、ノードがメッセージとアクティブエラーフレーム (ドミナントビットにより構成されます) を何ら制約無しに送信できます。

エラーパッシブ状態では、メッセージとパッシブエラーフレーム (リセッパビットにより構成されます) が送信されることがあります。

バスオフ状態では、ステーションは一時的にバス通信に参加できなくなります。この状態の間はメッセージは受信も送信もできません。送信側だけがバスオフとなります。

6.7 エラーモードとエラーカウンタ

MCP2515 には 2 つのエラーカウンタがあります: すなわちリセッパエラーカウンタ (REC) (レジスタ 6-2 参照) と送信エラーカウンタ (TEC) (レジスタ 6-1 参照) です。両カウンタの値は、MCU から読み出し可能です。これらのカウンタは CAN バス仕様にしたがって増減されます。

MCP2515 は両方のカウンタ値が 128 というエラーパッシブ限界より小さければエラーアクティブとなります。

エラーカウンタの少なくとも片方が 128 以上となるとエラーパッシブとなります。

TEC がバスオフ限界の 255 を超えるとバスオフとなります。この状態でもデバイスはバスオフ回復メッセージを受信します。バスオフ回復シーケンスは 11 個の連続リセッパビットが 128 回で構成されます。(図 6-1 参照)

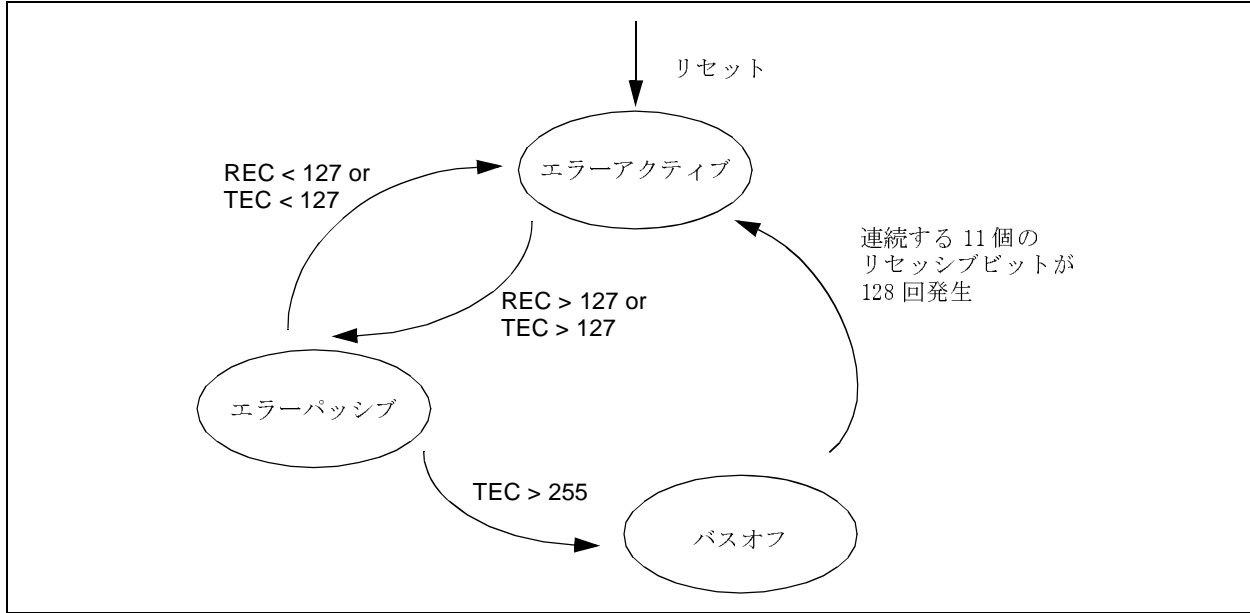
注: MCP2515 は、バスオフになった後、バスが 128 x 11 ビット時間だけアイドルとなると MCU による介入無しにエラーアクティブに復帰します。そうしたくないときは、エラー割り込み処理でこれを処理します。

MCP2515 の現状のエラー状態は EFLG レジスタによって MCU が読み出せます。(レジスタ 6-3 参照)。

さらにエラー状態警告フラグビット (EFLG:EWARN) があり、エラーカウンタの 1 つでもエラー警告限界の 96 以上になるとセットされます。EWARN は、両カウンタがエラー警告限界以下になるとリセットされます。

MCP2515

図 6-1: エラーモード状態遷移図



レジスタ 6-1: TEC - 送信エラーカウンタ
(アドレス: 1Ch)

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0
bit 7				bit 0			

bit 7-0 **TEC:** 送信エラーカウントビット <7:0>

凡例:

R = 読み出し可 W = 書き込み可 U = 未実装、読むと '0'
-n = POR 後の値 '1' = セット '0' = クリア x = 不定

レジスタ 6-2: REC - 受信エラーカウンタ
(アドレス: 1Dh)

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0
bit 7				bit 0			

bit 7-0 **REC:** 受信エラーカウントビット <7:0>

凡例:

R = 読み出し可 W = 書き込み可 U = 未実装、読むと '0'
-n = POR 後の値 '1' = セット '0' = クリア x = 不定

レジスタ 6-3: EFLG - エラーフラグ
(アドレス: 2Dh)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
RX1OVR	RX0OVR	TXBO	TXEP	RXEP	TXWAR	RXWAR	EWARN

bit 7

bit 0

- bit 7 **RX1OVR:** 受信バッファ 1 オーバーフローフラグビット
 - RXB1 かつ CANINTF.RX1IF = 1 のとき有効なメッセージが受信されるとセットされる
 - MCU からリセットする
- bit 6 **RX0OVR:** 受信バッファ 0 オーバーフローフラグビット
 - RXB0 かつ CANINTF.RX0IF = 1 のとき有効なメッセージが受信されるとセットされる
 - MCU からリセットする
- bit 5 **TXBO:** バスオフエラーフラグビット
 - TEC が 255 になるとセットされる
 - バス回復シーケンスの正常受信後にリセット
- bit 4 **TXEP:** 送信エラー - パッシブエラーフラグビット
 - TEC が 128 以上になるとセットされる
 - TEC が 128 より小さくなるとリセットされる
- bit 3 **RXEP:** 受信エラー - パッシブフラグビット
 - REC が 128 以上となるとセットされる
 - REC が 128 より小さくなるとリセットされる
- bit 2 **TXWAR:** 送信エラー警告フラグビット
 - TEC が 96 以上となるとセットされる
 - TEC が 96 より小さくなるとリセットされる
- bit 1 **RXWAR:** 受信エラー警告フラグビット
 - REC が 96 以上になるとセットされる
 - REC が 96 より小さくなるとリセットされる
- bit 0 **EWARN:** エラー警告フラグビット
 - TEC または REC が 96 以上となるとセットされる (TXWAR または RXWAR = 1)
 - REC と TEC 両方が 96 より小さくなるとリセットされる

凡例:

R = 読み出し可	W = 書き込み可	U = 未実装、読むと '0'
-n = POR 後の値	'1' = セット	'0' = クリア x = 不定

MCP2515

NOTES:

7.0 割り込み

MCP2515 は 8 個の割り込み要因を持っています。CANINTE レジスタが個々の割り込み要因の割り込みフラグビットとなっています。割り込みが発生すると、 $\overline{\text{INT}}$ ピンが MCP2515 により Low にドライブされ、MCU により割り込みがクリアされるまで保持します。割り込みは対応する条件がある限りクリアすることはできません。

CANINTF レジスタのフラグビットのクリアには、通常の書き込み動作よりビット変更コマンドを使うことをお勧めします。これは書き込みコマンド中に意図しないフラグの変更が起きて誤った割り込みが発生するのを防止するためです。

CANINTF フラグは読み書きができ、MCU がこれらのビットをセットしてしまうと、CANINTE ビットがセットされていると割り込みが発生することに注意が必要です。

7.1 割り込みコードビット

割り込み待ちの要因はレジスタ 10-2 に示した CANSTAT.ICOD (割り込みコード) ビットで示されます。複数割り込みイベントが起きた場合には、MCU により割り込みがリセットされるまで $\overline{\text{INT}}$ ピンが Low のままとなります。CANSTAT.ICOD ビットは、現在割り込み待ち中の最も高い優先度の割り込みのコードを反映します。割り込みは内部で ICOD 値の小さいものが高い優先度に優先順位づけされています。最も高い優先度の割り込みがクリアされると、(もしあれば) 割り込み待ち中の割り込みの中で次に優先度の高い割り込みのコードが ICOD ビットに反映されます。(表 7-1 参照) 割り込み要因の中で対応する CANINTE 許可ビットがセットされているものだけが ICOD ビットに反映されます。

表 7-1: ICOD<2:0> デコード

ICOD<2:0>	ブール代数表記
000	$\overline{\text{ERR}} \cdot \overline{\text{WAK}} \cdot \overline{\text{TX0}} \cdot \overline{\text{TX1}} \cdot \overline{\text{TX2}} \cdot \overline{\text{RX0}} \cdot \overline{\text{RX1}}$
001	ERR
010	$\overline{\text{ERR}} \cdot \overline{\text{WAK}}$
011	$\overline{\text{ERR}} \cdot \overline{\text{WAK}} \cdot \overline{\text{TX0}}$
100	$\overline{\text{ERR}} \cdot \overline{\text{WAK}} \cdot \overline{\text{TX0}} \cdot \overline{\text{TX1}}$
101	$\overline{\text{ERR}} \cdot \overline{\text{WAK}} \cdot \overline{\text{TX0}} \cdot \overline{\text{TX1}} \cdot \overline{\text{TX2}}$
110	$\overline{\text{ERR}} \cdot \overline{\text{WAK}} \cdot \overline{\text{TX0}} \cdot \overline{\text{TX1}} \cdot \overline{\text{TX2}} \cdot \overline{\text{RX0}}$
111	$\overline{\text{ERR}} \cdot \overline{\text{WAK}} \cdot \overline{\text{TX0}} \cdot \overline{\text{TX1}} \cdot \overline{\text{TX2}} \cdot \overline{\text{RX0}} \cdot \overline{\text{RX1}}$

注: $\overline{\text{ERR}}$ は CANINTE, ERRIE に対応

7.2 送信割り込み

送信割り込みが許可 (CANINTE.TXnIE = 1) されていると、対応するバッファが空で新メッセージがロードできる状態になると $\overline{\text{INT}}$ ピンに割り込みが発生します。CANINTF.TXnIF ビットがセットされて割り込み要因を示します。割り込みは TXnIF ビットをクリアすることでクリアされます。

7.3 受信割り込み

受信割り込みが許可 (CANINTE.RXnIE = 1) されていると、メッセージが正常に受信され、対応する受信バッファにロードされると $\overline{\text{INT}}$ ピンに割り込みが発生します。この割り込みは EOF フィールド受信後直ぐ発生します。CANINTF.RXnIF ビットがセットされて割り込み要因を示します。割り込みは RXnIF ビットをクリアすることでクリアされます。

7.4 メッセージエラー割り込み

メッセージの送受信中にエラーが発生すると、メッセージエラーフラグ (CANINTF.MERRF) がセットされ、CANINTE.MERRE がセットされていれば、割り込みが $\overline{\text{INT}}$ ピンに生成されます。これはリスンオンリーモードで使われたときには、ポーレート決定のために使われます。

7.5 バスアクティビティウエイクアップ割り込み

MCP2515 がスリープモードでバスアクティビティ割り込みが許可 (CANINTE.WAKIE = 1) されているときは、CAN バスでアクティビティが検出されると CANINTF.WAKIF ビットがセットされ、 $\overline{\text{INT}}$ ピンに割り込みが生成されます。この割り込みは MCP2515 をスリープモードから抜け出させます。割り込みは WAKIF ビットをクリアするとリセットされます。

注: MCP2515 はウエイクアップしてリスンオンリーモードになります。

7.6 エラー割り込み

エラー割り込みが許可 (CANINTE.ERRIE = 1) されていると、オーバーフローが起きたときか送受信器のエラー状態が変わったときに $\overline{\text{INT}}$ ピンに割り込みを生成します。エラーフラグレジスタ (EFLG) が下記の状態のひとつを表します。

7.6.1 受信オーバーフロー

MAB が有効なメッセージ (メッセージが受入フィルタに合致した) を組み立てたとき、対応するフィルタの受信バッファが新たなメッセージをロードできないと、オーバーフロー状態が発生します。対応する EFLG.RXnOVR ビットがセットされてオーバーフローを表します。このビットは MCU からクリアしなければなりません。

MCP2515

7.6.2 受信警告

REC が MCU 警告限界の 96 に達したとき。

7.6.3 送信警告

TEC が MCU 警告限界の 96 に達した。

7.6.4 受信エラーパッシブ

REC がエラーパッシブ限界 127 を超えてデバイスがエラーパッシブ状態になった。

7.6.5 送信エラーパッシブ

TEC がエラーパッシブ限界の 127 を超えてデバイスがエラーパッシブ状態になった。

7.6.6 バスオフ

TEC が 255 を超えるとデバイスはバスオフ状態となり切り離されます。

7.7 割り込みアクノレッジ

割り込みは CANINTF レジスタの 1 つ以上の状態フラグに直接関連します。フラグが 1 つでもセットされると割り込み待ち中となります。いったん割り込みフラグがデバイスでセットされると、割り込み要因がなくなって MCU がフラグをリセットするまで継続しません。

レジスタ 7-1: CANINTE - 割り込み許可
(アドレス: 2Bh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
MERRE	WAKIE	ERRIE	TX2IE	TX1IE	TX0IE	RX1IE	RX0IE

bit 7 bit 0

- bit 7 **MERRE:** メッセージエラー割り込み許可ビット
1 = メッセージ送受信中にエラー発生で割り込む
0 = 禁止
- bit 6 **WAKIE:** ウェイクアップ割り込み許可ビット
1 = CAN バスアクティビティで割り込む
0 = 禁止
- bit 5 **ERRIE:** エラー割り込み許可ビット (EFLG レジスタの複数要因)
1 = EFLG エラー状態が変化したとき割り込む
0 = 禁止
- bit 4 **TX2IE:** 送信バッファ 2 空割り込み許可ビット
1 = TXB2 が空になると割り込む
0 = 禁止
- bit 3 **TX1IE:** 送信バッファ 1 空割り込み許可ビット
1 = TXB1 が空になると割り込む
0 = 禁止
- bit 2 **TX0IE:** 送信バッファ 0 空割り込み許可ビット
1 = TXB0 が空になると割り込む
0 = 禁止
- bit 1 **RX1IE:** 受信バッファ 1 フル割り込み許可ビット
1 = RXB1 にメッセージが受信されると割り込む
0 = 禁止
- bit 0 **RX0IE:** 受信バッファ 0 フル割り込み許可ビット
1 = RXB0 にメッセージが受信されると割り込む
0 = 禁止

凡例:

R = 読み出し可	W = 書き込み可	U = 未実装、読むと '0'
-n = POR 後の値	'1' = セット	'0' = クリア x = 不定

レジスタ 7-2: CANINTF - 割り込みフラグ
(アドレス: 2Ch)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
MERRF	WAKIF	ERRIF	TX2IF	TX1IF	TX0IF	RX1IF	RX0IF

bit 7

bit 0

- bit 7 **MERRF:** メッセージエラー割り込みフラグビット
 1 = 割り込み待ち中 (割り込み条件リセットには MCU からクリアが必要)
 0 = 割り込み待ちなし
- bit 6 **WAKIF:** ウェイクアップ割り込みフラグビット
 1 = 割り込み待ち中 (割り込み条件リセットには MCU からクリアが必要)
 0 = 割り込み待ちなし
- bit 5 **ERRIF:** エラー割り込みフラグビット (EFLG レジスタの複数要因)
 1 = 割り込み待ち中 (割り込み条件リセットには MCU からクリアが必要)
 0 = 割り込み待ちなし
- bit 4 **TX2IF:** 送信バッファ 2 空割り込みフラグビット
 1 = 割り込み待ち中 (割り込み条件リセットには MCU からクリアが必要))
 0 = 割り込み待ちなし
- bit 3 **TX1IF:** 送信バッファ 1 空割り込みフラグビット
 1 = 割り込み待ち中 (割り込み条件リセットには MCU からクリアが必要)
 0 = 割り込み待ちなし
- bit 2 **TX0IF:** 送信バッファ 0 空割り込みフラグビット
 1 = 割り込み待ち中 (割り込み条件リセットには MCU からクリアが必要)
 0 = 割り込み待ちなし
- bit 1 **RX1IF:** 受信バッファ 1 フル割り込みフラグビット
 1 = 割り込み待ち中 (割り込み条件リセットには MCU からクリアが必要)
 0 = 割り込み待ちなし
- bit 0 **RX0IF:** 受信バッファ 0 フル割り込みフラグビット
 1 = 割り込み待ち中 (割り込み条件リセットには MCU からクリアが必要)
 0 = 割り込み待ちなし

凡例:

R = 読み出し可	W = 書き込み可	U = 未実装、読むと '0'
-n = POR 後の値	'1' = セット	'0' = クリア x = 不定

MCP2515

NOTES:

8.0 発振器

MCP2515 は水晶かセラミック発振子を OSC1 と OSC2 ピンに接続することで動作するように設計されています。MCP2515 発振器は並列カットの水晶を使うことで設計されています。直列カットの水晶を使うと水晶製造仕様からはずれた周波数で発振することがあります。標準的な発振回路を図 8-1 に示します。MCP2515 は、図 8-2 と 図 8-3 に示したように、外部クロック源を OSC1 ピンに接続してドライブすることもできます。

8.1 発振スタートアップタイム

MCP2515 は発振スタートアップタイム (OST) でリセット状態を保持します。これにより発振が安定してから内部状態マシンの動作を開始します。電源オンまたはスリープモードからのウェイクアップ後の最初の 128 OSC1 クロックサイクルの間は、OST がリセット状態を保持します。これにより OST が終了するまで、SPI プロトコルも動作できないことに注意してください。

8.2 クロック出力ピン

CLKOUT ピンは、システム中の他のデバイスへのメインクロックやクロック入力として使えるようシステム設計者に提供されています。CLKOUT は内部に分周器を持っていて、 F_{OSC} を 1, 2, 4, 8 で分周できます。CLKOUT 機能の許可と分周器の選択は、CANCNTRL レジスタで行います。(レジスタ 10-1 参照)

注： CLKOUT の最高周波数は 25 MHz として規定されています。(表 13-5 参照)

CLKOUT ピンはシステムリセットでアクティブになり、デフォルトでは最も遅い速度 (8 分周) となり MCU のクロックとして使えるようになっています。

スリープモードが要求されたときは、MCP2515 はスリープモードに入る前に、16 個の追加クロックを CLKOUT ピンに出力します。CLKOUT 機能が禁止 (CANCNTRL.CLKEN = '0') されると、CLKOUT ピンはハイインピーダンス状態となります。

CLKOUT 機能は、CLKOUT ピン機能が許可、禁止されたとき、および分周器が変更されたときには、 $t_{hCLKOUT}$ と $t_{lCLKOUT}$ のタイミングを保つように設計されています。

図 8-1: 水晶/セラミック発振子による場合

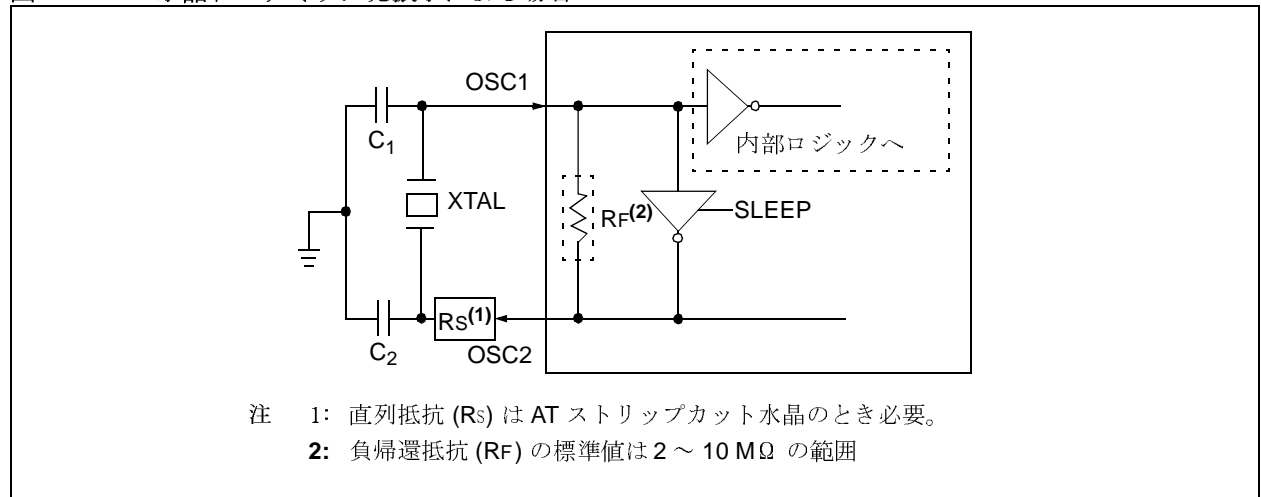
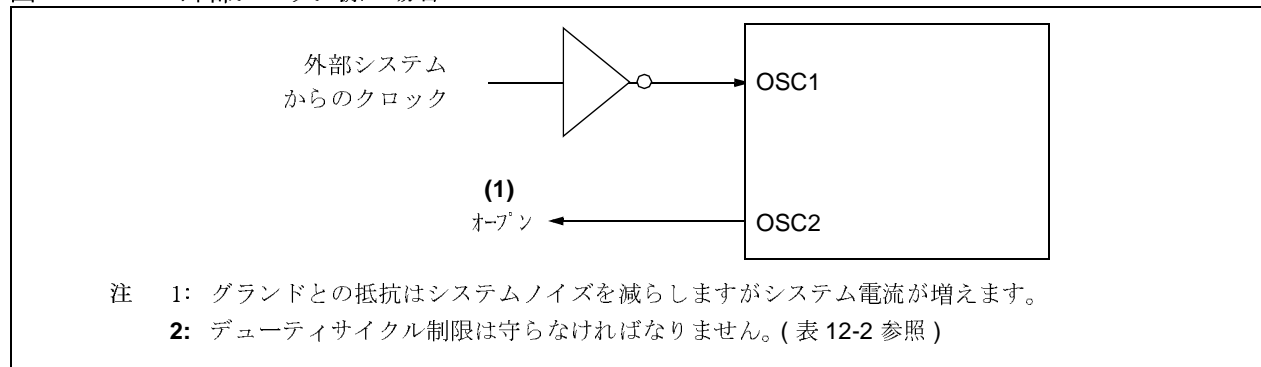


図 8-2: 外部クロック源の場合



MCP2515

図 8-3: 外付け直列水晶発振子の発振回路 (1)

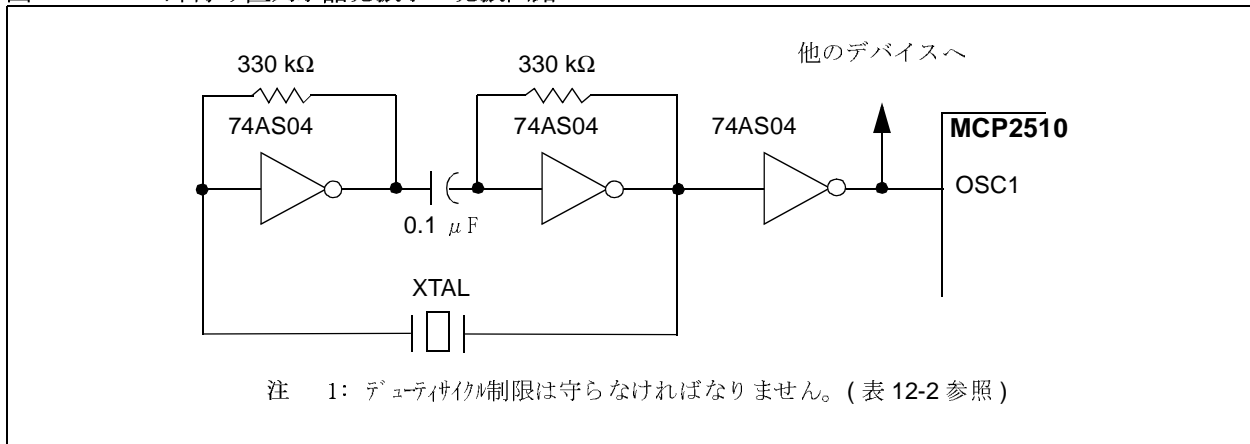


表 8-1: セラミック発振子の場合のコンデンサの選択

使用できる容量の標準値:			
モード	周波数	OSC1	OSC2
HS	8.0 MHz	27 pF	27 pF
	16.0 MHz	22 pF	22 pF

容量値は設計ガイド用のみ:
これらの容量は下記発振子で基本的なスタートアップと動作をテストしたもの。これらの値は最適化はされていません。
異なる容量値が許容できる発振動作のために必要とされるかも知れません。ユーザーの使用方法で期待する VDD と温度範囲全体で発振性能をテストすることを推奨します。
さらなる追加情報は下記表 8-2 の注を参照

使用発振子:
4.0 MHz
8.0 MHz
16.0 MHz

表 8-2: 水晶発振子の場合のコンデンサの選択

Osc タイプ (1)(4)	水晶周波数 (2)	テストした標準容量値:	
		C1	C2
HS	4 MHz	27 pF	27 pF
	8 MHz	22 pF	22 pF
	20 MHz	15 pF	15 pF

容量値は設計ガイド用のみ:
これらの容量は下記水晶で基本的なスタートアップと動作をテストしたもの。これらの値は最適化はされていません。
異なる容量値が許容できる発振動作のために必要とされるかも知れません。ユーザーの使用方法で期待する VDD と温度範囲全体で発振性能をテストすることを推奨します。
さらなる追加情報については本表の注を参照

使用水晶 (3):
4.0 MHz
8.0 MHz
20.0 MHz

- 注
- 容量を大きくすると発振の安定度は向上しますが、スタートアップ時間も増えます。
 - いずれの発振子 / 水晶でも個々の特性を持っていますので、ユーザーは外付けの部品の最適値については発振子 / 水晶のメーカーに相談すべきです。
 - RS は低負荷ドライブ仕様の水晶の過負荷を避けるために必要とされます。
 - 使用方法で期待される VDD と温度の全範囲で発振性能の確認を常にしてください。

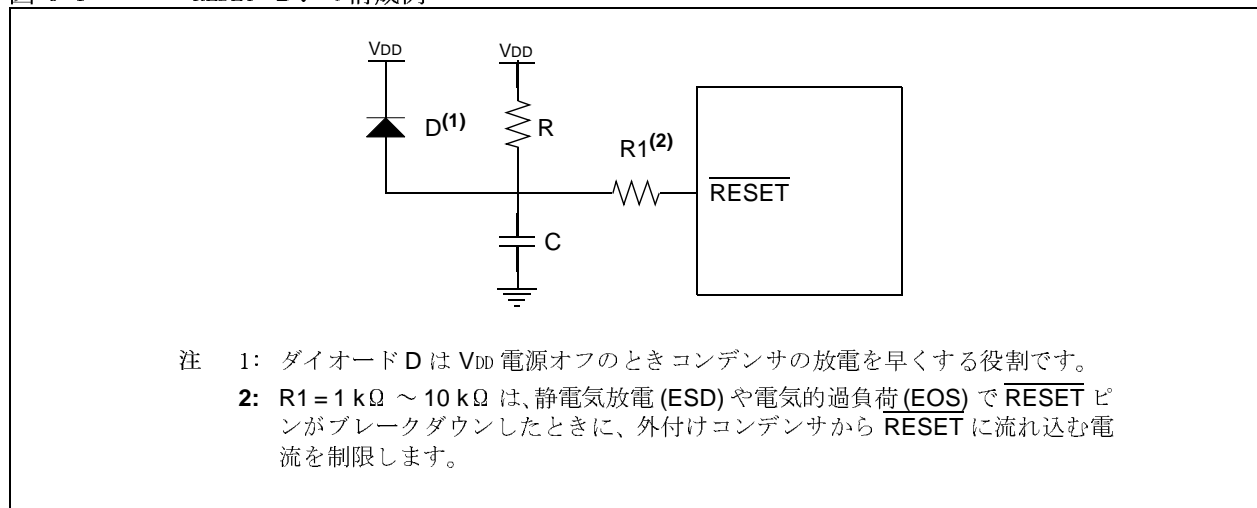
9.0 リセット

MCP2515 は2つのリセットを識別します。:

1. ハードウェアリセット – $\overline{\text{RESET}}$ ピンを Low
2. SPI リセット – SPI コマンドでリセット

両方のリセットとも機能は同じです。重要なことは、電源オン後に2つのリセットのいずれかを実行して、ロジックとレジスタを確実にデフォルト状態にすることです。ハードウェアリセットなら $\overline{\text{RESET}}$ ピンに RC を付けることで自動的にできるようになります。(図 9-1 参照) この値は、電氣的仕様で示されている (trL) とおり、VDD が動作電圧になってから、デバイスが最小2 μs の間リセットが保持されるようにしなければなりません。

図 9-1: $\overline{\text{RESET}}$ ピンの構成例



MCP2515

NOTES:

10.0 動作モード

MCP2515 は5つの動作モードを持っています。これらのモードとは下記となります。:

1. コンフィギュレーションモード
2. 通常モード
3. スリープモード
4. リスンオンリーモード
5. ループバックモード

動作モードは **CANCTRL.REQOP** ビット (レジスタ 10-1 参照) によって選択されます。

モードを変更すると、送信中のメッセージが完了するまでモードの実際の変更は行われません。要求されたモードは **CANSTAT.OPMODE** ビット (レジスタ 10-2 参照) を読むことで参照できます。

10.1 コンフィギュレーションモード

MCP2515 は動作前に初期化されなければなりません。これは唯一デバイスがコンフィギュレーションモードにあるときだけできます。コンフィギュレーションモードは、電源オンリセットでは自動的に選択され、**CANCTRL.REQOP** ビットを '100' にすることで、どんなモードからも入ることができます。コンフィギュレーションモードに入ると、すべてのエラーカウンタがクリアされます。コンフィギュレーションモードは下記のレジスタが設定できる唯一のモードです。:

- CNF1, CNF2, CNF3
- TXRTSCTRL
- フィルタレジスタ
- マスクレジスタ

10.2 スリープモード

MCP2515 は内部スリープモードを持っていて、デバイスの電流消費を最小にするのに使います。SPI インターフェースは、MCP2515 がスリープモードになっても読み出しがアクティブのまま、すべてのレジスタにアクセスできます。

スリープモードにするには、**CANCTRL** レジスタのモード要求ビット (**REQOP<2:0>**) を設定します。**CANSTAT.OPMODE** ビットは動作モードを表しています。このビットは MCP2515 にスリープコマンドを送った後で読み出します。このビットがスリープモードに入ったことを表すまで、MCP2515 がアクティブでまだスリープモードに入っていないこととなります。

内部スリープモードのときは、ウェイクアップ割り込みは (許可されていれば) アクティブのままです。これにより MCU もスリープモードにでき、MCP2515 のバスのアクティビティ検出をウェイクアップに使うことができます

スリープモードのときは、MCP2515 は内部発振を停止します。MCP2515 はバスアクティビティが起きたとき、または MCU が SPI インターフェース経由で **CANINTE.WAKIF** ビットをセットしてウェイクアップ要求を生成したときウェイクアップします。(CANINTE.WAKIE ビットは、ウェイクアップ割り込みを起すためにはセットしなければなりません。)

TXCAN ピンは、MCP2515 がスリープモード中でも、リセシブ状態を維持します。

10.2.1 ウェイクアップ機能

デバイスはスリープ中は、アクティビティのために **RXCAN** ピンをモニタしています。**CANINTE.WAKIE** ビットがセットされると、デバイスはウェイクアップして割り込みを発生します。内部発振器はスリープ中はシャットダウンしていますので、発振器が発振開始してデバイス自身がメッセージを受信可能になるまでにはちょっと時間がかかります。この発振スタートアップタイマー (OST) は 128 T_{osc} として定義されています。

デバイスがスリープモードからウェイクアップするときのメッセージは、デバイスがウェイクアップ中のメッセージと同様に無視します。デバイスはウェイクアップするとリスンオンリーモードとなります。MCP2515 がバス上で通信ができるようにするためには、MCU が通常モードに設定しなければなりません。

デバイスは内部スリープ中の間、**RXCAN** 入力ピンにローパスフィルタ機能を適用するようプログラムできます。この特性は、デバイスが CAN バスライン上の短時間のグリッチでウェイクアップしてしまうのを防ぐために使うことができます。**CNF3.WAKFIL** ビットがフィルタの許可禁止を行います。

10.3 リスンオンリーモード

リスンオンリーモードは、**RXBnCTRL.RXM<1:0>** ビットを設定することで、MCP2515 がすべてのメッセージ (エラーメッセージも含めて) を受信する手段を提供します。このモードは、バスモニタ用途や、ホットプラグ状況でポーレート検出するときに使えます。

自動ポーレート検出は、通信する相手が2つのノードだけのとき必要となります。ポーレートは有効なメッセージを受信できるまで異なる値でテストすることで実験的に決定されます。

リスンオンリーモードは、無言のモードです。つまりこのモードの間は何のメッセージも (エラーフラグやアクノレッジ信号を含めて) 送信しません。フィルタやマスクは、特別なメッセージだけを受信バッファに格納することを許可するためや、マスクをすべてゼロにしてどんな識別子のメッセージでも通すように使えます。エラーカウンタはリセットされ、この状態では使われません。リスンオンリーモードは **CANCTRL** レジスタのモード要求ビットを設定することでアクティブにできます。

MCP2515

10.4 ループバックモード

ループバックモードは、CAN バスに実際にメッセージ送信を行わずに、内部で送信バッファから受信バッファにメッセージ転送するようにします。このモードはシステム開発やテストに使うことができます。

このモードでは、ACK ビットは無視され、デバイスが自分自身から来る入力メッセージを他のノードから来たかのように扱えます。ループバックモードは、無言のモードです。つまりこのモードの間は何のメッセージも(エラーフラグやアクノレッジ信号を含めて)送信しません。TXCAN ピンはリセット状態となります。

フィルタやマスクは、特別なメッセージだけを受信バッファに格納することを許可することができます。マスクをすべてゼロにしてどんな識別子のメッセージ

でも通すように使えます。ループバックモードは CANCTRL レジスタのモード要求ビットを設定することでアクティブにできます。

10.5 通常モード

通常モードは、MCP2515 の標準の動作モードです。このモードでは、デバイスは積極的に全バスメッセージをモニタし、アクノレッジビットやエラーフレームなどを生成します。これは、MCP2515 がメッセージを CAN バスに送信できる唯一のモードです。

レジスタ 10-1: CANCTRL - CAN 制御レジスタ (アドレス: XFh)

R/W-1	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1	R/W-1	
REQOP2	REQOP1	REQOP0	ABAT	OSM	CLKEN	CLKPRE1	CLKPRE0	
bit 7								bit 0

bit 7-5 **REQOP:** 動作モード要求ビット <2:0>
 000 = 通常動作モードに設定
 001 = スリープモードに設定
 010 = ループバックモードに設定
 011 = リスンオンリーモードに設定
 100 = コンフィギュレーションモードに設定
 REQOP ビットの他の値は不正で使われません。

注: 電源オン時は REQOP = b'111'

bit 4 **ABAT:** すべての送信の停止ビット
 1 = すべての送信バッファの送信停止要求
 0 = 全送信停止の終了要求

bit 3 **OSM:** ワンショットモードビット
 1 = 許可、メッセージは 1 回だけ送信される
 0 = 禁止、メッセージは要求のつど送信される

bit 2 **CLKEN:** CLKOUT ピン有効化ビット
 1 = CLKOUT ピン有効
 0 = CLKOUT ピン無効 (ピンはハイインピーダンス状態となる)

bit 1-0 **CLKPRE:** CLKOUT ピン分周設定ビット <1:0>
 00 = FCLKOUT = システムクロック /1
 01 = FCLKOUT = システムクロック /2
 10 = FCLKOUT = システムクロック /4
 11 = FCLKOUT = システムクロック /8

凡例:

R = 読み出し可	W = 書き込み可	U = 未実装、読むと '0'
-n = POR 後の値	'1' = セット	'0' = クリア
		x = 不定

レジスタ 10-2: CANSTAT - CAN 状態レジスタ
(アドレス: XEh)

R-1	R-0	R-0	U-0	R-0	R-0	R-0	U-0
OPMOD2	OPMOD1	OPMOD0	—	ICOD2	ICOD1	ICOD0	—
bit 7				bit 0			

- bit 7-5 **OPMOD**: 動作モードビット <2:0>
 000 = デバイスは通常動作モード中
 001 = デバイスはスリープモード中
 010 = デバイスはループバックモード中
 011 = デバイスはリスンオンリーモード中
 100 = デバイスはコンフィギュレーションモード中
- bit 4 未実装: 読むと '0'
- bit 3-1 **ICOD**: 割り込みフラグコードビット <2:0>
 000 = 割り込みなし
 001 = エラー割り込み
 010 = ウェイクアップ割り込み
 011 = TXB0 割り込み
 100 = TXB1 割り込み
 101 = TXB2 割り込み
 110 = RXB0 割り込み
 111 = RXB1 割り込み
- bit 0 未実装: 読むと '0'

凡例:

R = 読み出し可	W = 書き込み可	U = 未実装、読むと '0'
-n = POR 後の値	'1' = セット	'0' = クリア x = 不定

MCP2515

NOTES:

11.0 レジスタマップ

MCP2515のレジスタマップを表 11-1に示します。各レジスタのアドレス位置は列（高位4ビット）と行（下位ビット）の値で決まります。レジスタはシーケンシャルなデータ読み書きに最適化した配置となっ

ています。ある特定の制御や状態レジスタはSPIビット変更コマンドでビットごとに変更ができます。このコマンドが許可されているレジスタを影付きで表 11-1に示します。MCP2515の制御レジスタのサマリを表 11-2に示します。

表 11-1: CAN 制御レジスタマップ

低位 アドレス ビット	高位アドレスビット							
	0000 xxxx	0001 xxxx	0010 xxxx	0011 xxxx	0100 xxxx	0101 xxxx	0110 xxxx	0111 xxxx
0000	RXF0SIDH	RXF3SIDH	RXM0SIDH	TXB0CTRL	TXB1CTRL	TXB2CTRL	RXB0CTRL	RXB1CTRL
0001	RXF0SIDL	RXF3SIDL	RXM0SIDL	TXB0SIDH	TXB1SIDH	TXB2SIDH	RXB0SIDH	RXB1SIDH
0010	RXF0EID8	RXF3EID8	RXM0EID8	TXB0SIDL	TXB1SIDL	TXB2SIDL	RXB0SIDL	RXB1SIDL
0011	RXF0EID0	RXF3EID0	RXM0EID0	TXB0EID8	TXB1EID8	TXB2EID8	RXB0EID8	RXB1EID8
0100	RXF1SIDH	RXF4SIDH	RXM1SIDH	TXB0EID0	TXB1EID0	TXB2EID0	RXB0EID0	RXB1EID0
0101	RXF1SIDL	RXF4SIDL	RXM1SIDL	TXB0DLC	TXB1DLC	TXB2DLC	RXB0DLC	RXB1DLC
0110	RXF1EID8	RXF4EID8	RXM1EID8	TXB0D0	TXB1D0	TXB2D0	RXB0D0	RXB1D0
0111	RXF1EID0	RXF4EID0	RXM1EID0	TXB0D1	TXB1D1	TXB2D1	RXB0D1	RXB1D1
1000	RXF2SIDH	RXF5SIDH	CNF3	TXB0D2	TXB1D2	TXB2D2	RXB0D2	RXB1D2
1001	RXF2SIDL	RXF5SIDL	CNF2	TXB0D3	TXB1D3	TXB2D3	RXB0D3	RXB1D3
1010	RXF2EID8	RXF5EID8	CNF1	TXB0D4	TXB1D4	TXB2D4	RXB0D4	RXB1D4
1011	RXF2EID0	RXF5EID0	CANINTE	TXB0D5	TXB1D5	TXB2D5	RXB0D5	RXB1D5
1100	BFPCTRL	TEC	CANINTF	TXB0D6	TXB1D6	TXB2D6	RXB0D6	RXB1D6
1101	TXRTSCTRL	REC	EFLG	TXB0D7	TXB1D7	TXB2D7	RXB0D7	RXB1D7
1110	CANSTAT	CANSTAT	CANSTAT	CANSTAT	CANSTAT	CANSTAT	CANSTAT	CANSTAT
1111	CANCTRL	CANCTRL	CANCTRL	CANCTRL	CANCTRL	CANCTRL	CANCTRL	CANCTRL

注： 影付きレジスタの番地は、ユーザーがビット変更コマンドで、個々のビットを扱うことができることを示しています。

表 11-2: 制御レジスタのサマリ

レジスタ 名称	アドレス (Hex)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR/RST Value
BFPCTRL	0C	—	—	B1BFS	B0BFS	B1BFE	B0BFE	B1BFM	B0BFM	--00 0000
TXRTSCTRL	0D	—	—	B2RTS	B1RTS	B0RTS	B2RTSM	B1RTSM	B0RTSM	--xx x000
CANSTAT	xE	OPMOD2	OPMOD1	OPMOD0	—	ICOD2	ICOD1	ICOD0	—	100- 000-
CANCTRL	xF	REQOP2	REQOP1	REQOP0	ABAT	OSM	CLKEN	CLKPRE1	CLKPRE0	1110 0111
TEC	1C	Transmit Error Counter (TEC)								0000 0000
REC	1D	Receive Error Counter (REC)								0000 0000
CNF3	28	SOF	WAKFIL	—	—	—	PHSEG22	PHSEG21	PHSEG20	00-- -000
CNF2	29	BTLMODE	SAM	PHSEG12	PHSEG11	PHSEG10	PRSEG2	PRSEG1	PRSEG0	0000 0000
CNF1	2A	SJW1	SJW0	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0	0000 0000
CANINTE	2B	MERRE	WAKIE	ERRIE	TX2IE	TX1IE	TX0IE	RX1IE	RX0IE	0000 0000
CANINTF	2C	MERRF	WAKIF	ERRIF	TX2IF	TX1IF	TX0IF	RX1IF	RX0IF	0000 0000
EFLG	2D	RX1OVR	RX0OVR	TXBO	TXEP	RXEP	TXWAR	RXWAR	EWARN	0000 0000
TXB0CTRL	30	—	ABTF	MLOA	TXERR	TXREQ	—	TXP1	TXP0	-000 0-00
TXB1CTRL	40	—	ABTF	MLOA	TXERR	TXREQ	—	TXP1	TXP0	-000 0-00
TXB2CTRL	50	—	ABTF	MLOA	TXERR	TXREQ	—	TXP1	TXP0	-000 0-00
RXB0CTRL	60	—	RXM1	RXM0	—	RXRTR	BUKT	BUKT	FILHIT0	-00- 0000
RXB1CTRL	70	—	RSM1	RXM0	—	RXRTR	FILHIT2	FILHIT1	FILHIT0	-00- 0000

MCP2515

NOTES:

12.0 SPI インターフェース

12.1 概要

MCP2515 は多くのマイクロコントローラで使えるシリアル周辺インターフェース (SPI) ポートと直接インターフェースするように設計されていて、モード 0,0 とモード 1,1 をサポートしています。コマンドとデータが SCK の立上りエッジでクロッキングされて、デバイスに SI ピン経由で送られます。MCP2515 からの出力データは SCK の立下りで (SO ライン上に) 出力されます。CS ピンはすべての動作中で Low に保持されていなければなりません。表 12-1 は全動作での命令バイトを示しています。モード 0,0 とモード 1,1 の両方の動作での入出力タイミングの詳細については図 12-10 と図 12-11 を参照してください。

注： MCP2515 は CS が Low になった直後のバイトは命令 / コマンドバイトとみなします。これはつまり、別のコマンドを起動すること、CS をいったん High にしてから Low にしなければならないということです。

12.2 リセット命令

リセット命令は MCP2515 の内部レジスタを再初期化し、コンフィギュレーションモードにするのに使うことができます。このコマンドは SPI インターフェースを介して、RESET ピンと同じ機能を提供します。

リセット命令は、1 バイト命令で CS を Low にすることでデバイスを選択し、命令バイトを送ったら CS を立ち上げます。リセットコマンド (または RESET ピンを Low にする) は、電源オン時の初期化シーケンスの一部として必ず実行するようお勧めします。

12.3 読み込み命令

読み込み命令は CS ピンを Low にすることから始まります。次に、読み込み命令と 8 ビットのアドレス (A7 から A0) が続けて MCP2515 に送られます。その次に、選択されたアドレスのレジスタに格納されていたデータが SO ピンにシフト出力されます。

内部アドレスポインタは、各 1 バイトシフト出力される都度、自動的に次のアドレスに増し分されます。したがって、次の連続するレジスタアドレスはクロックパルスを送るだけで読み込むことが可能です。この方法を使えば、連続するレジスタ位置ならいくつでもシーケンシャルに読み込むことができます。読み込み動作は、CS ピンを立ち上げることで終了します。(図 12-2)

12.4 RX バッファ読み込み命令

RX バッファ読み込み命令 (図 12-3) は、受信バッファを読み込むためのすばやいアドレス指定方法を提供します。この命令は 1 バイトのアドレスバイトにより SPI のオーバーヘッドを減らします。実際のコマンドバイトはアドレスポインタ指定のため 4 つの値をとります。一度コマンドバイトが送られると、コントローラは読み込み命令と同じアドレス位置のデータをクロックアウトします。(またシーケンシャル読み込

みも可能です) この命令は、コマンドの最後で CS が立ち上がったとき、関連する受信フラグ (CAN-INTF.RXnIF) を自動的にクリアすることで、SPI のオーバーヘッドをずっと少なくしてくれます。

12.5 書き込み命令

書き込み命令は CS ピンを Low にすることで始まります。次に書き込み命令にアドレスと最低 1 バイトのデータが続けて MCP2515 に送られます。

CS を Low にしている限り、データバイトとクロックを続ければシーケンシャルにレジスタに書き込むことができます。実際のデータのレジスタへの書き込みは、D0 ビットのときの SCK ラインの立上りエッジで実行されます。CS ラインが 8 ビットロードされるより前に High になると、そのデータバイトの書き込みは中止され、そのコマンドの直前のデータまでが書き込まれます。バイト書き込みシーケンスの詳細説明は図 12-4 のタイミング図を参照してください。

12.6 TX バッファへのロード命令

TX バッファロード命令 (図 12-5) は、通常書き込みコマンドに必要な 8 ビットのアドレスが要りません。8 ビット命令が、送信バッファに高速で書き込めるようアドレスポインタに 6 個のアドレスのうちの 1 つをセットします。このポインタが 3 つの送信バッファのどれかの ID かデータアドレスを指し示します。

12.7 送信要求 (RTS) 命令

RTS コマンドは、1 つ以上の送信バッファのメッセージの送信を起動するために使うコマンドです。

CS を Low にすることで MCP2515 を選択します。次に RTS コマンドバイトを送ります。図 12-6 に示したように、このコマンドの最後の 3 ビットがどの送信バッファを送信許可するかを示しています。

このコマンドは対応するバッファの TxBnC-TRL.TXREQ ビットをセットします。最後の 3 ビットの内、どれかあるいはすべてが 1 つのコマンドでセットできます。RTS コマンドが nnn = 000 で送られるとそのコマンドは無視されます。

12.8 状態読み込み命令

状態読み込み命令は、メッセージ送受信の状態ビットとして使われているどれかに 1 命令でアクセスできるようにします。

CS を Low にして MCP2515 を選択し、次に図 12-8 に示される状態読み込みコマンドバイトを MCP2515 に送ります。一度コマンドバイトが送信されると、MCP2515 は状態を含む 8 ビットのデータを返送します。

最初の 8 ビット送信に続いて追加クロックが送られれば、MCP2515 は CS ピンが Low に保持されクロックが SCK に供給される限り、状態ビットを出力し続けます。このコマンドで返送される各状態ビットは、標準の読み込みコマンドを使っても、適切にレジスタアドレスを指定すれば、読み込めます。

図 12-2: 読み込み命令

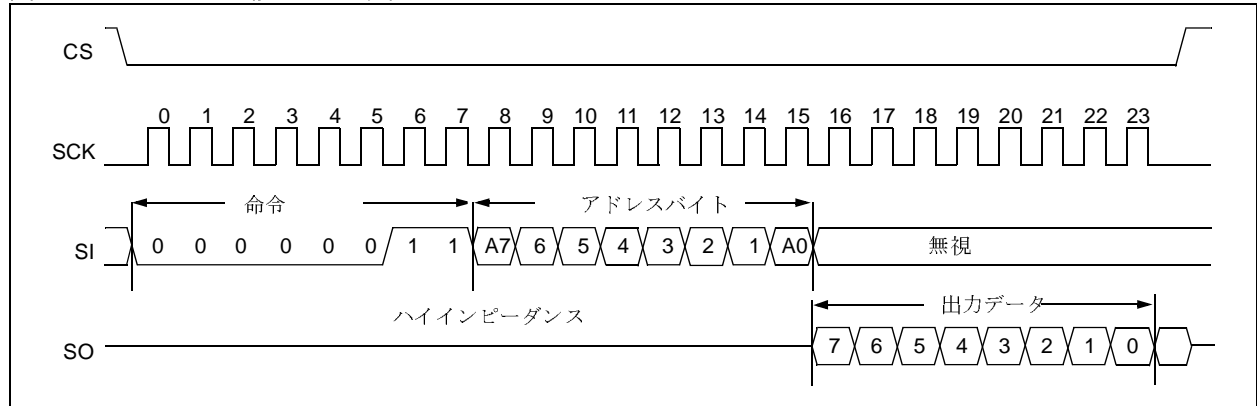


図 12-3: RX バッファ読み込み命令

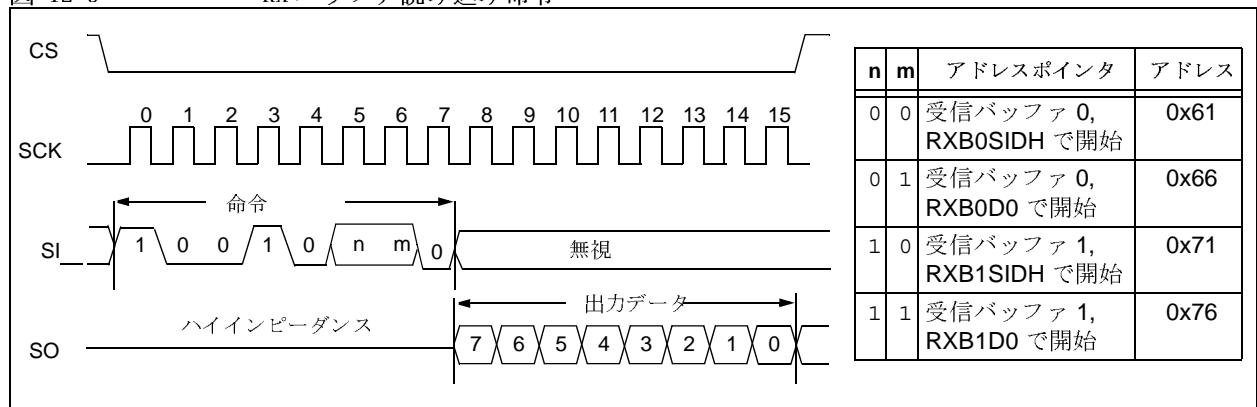
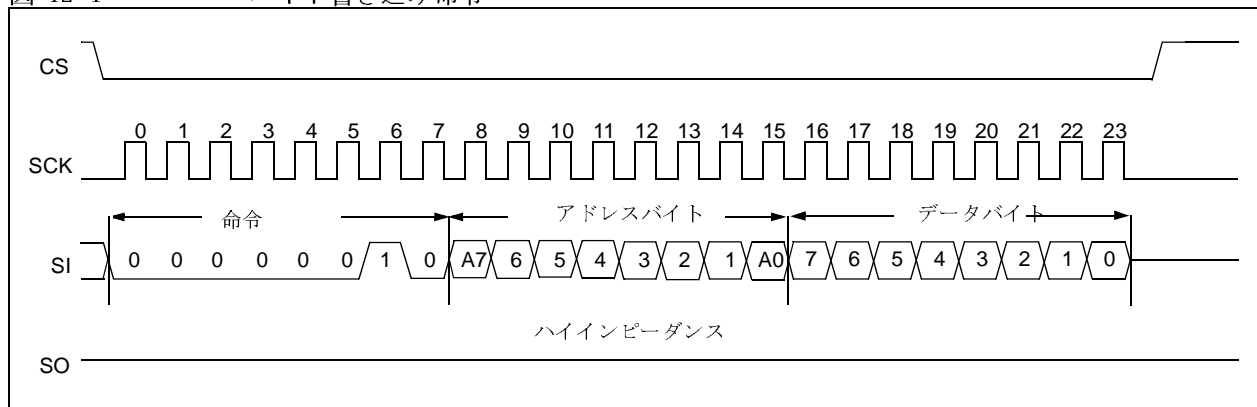


図 12-4: バイト書き込み命令



MCP2515

図 12-5: TX バッファロード

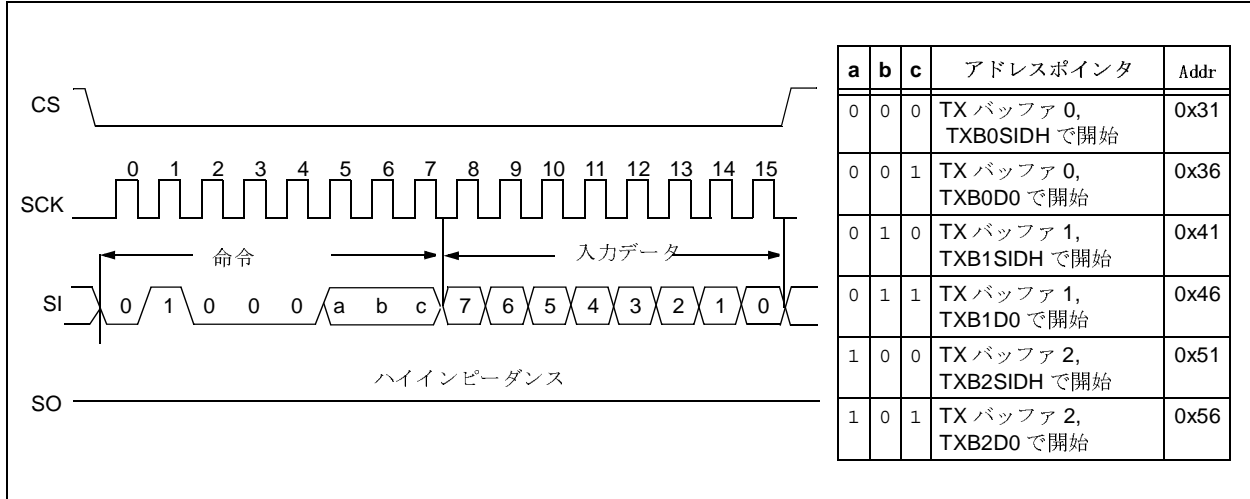


図 12-6: 送信要求 (RTS) 命令

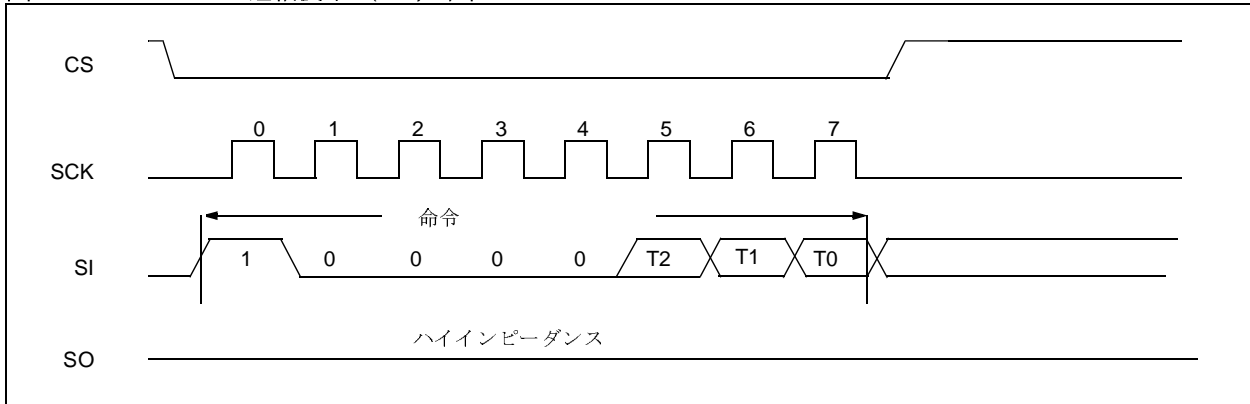


図 12-7: ビット変更命令

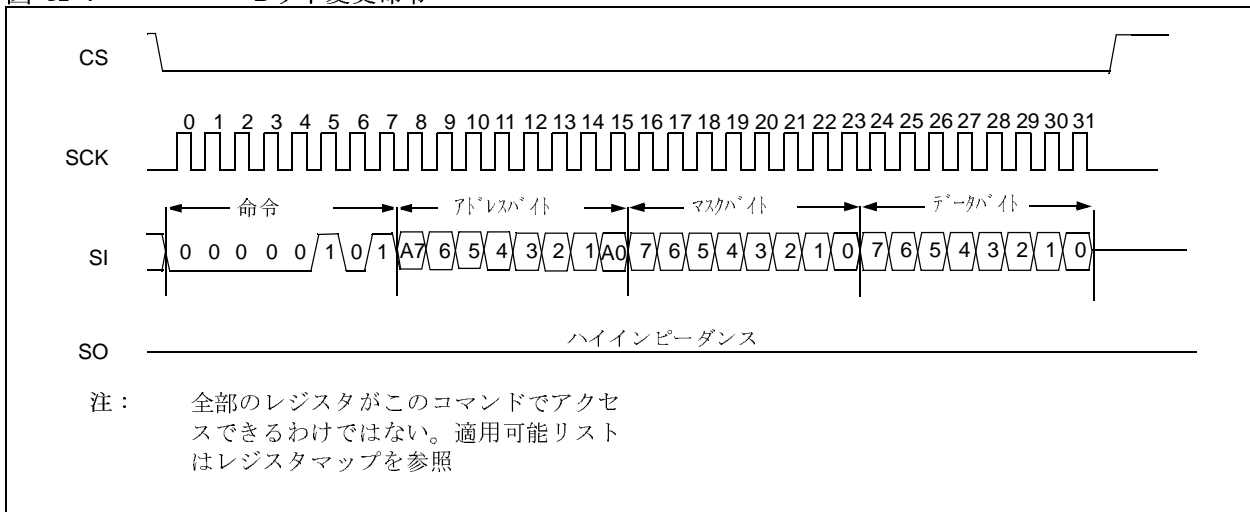


図 12-8: 状態読み込み命令

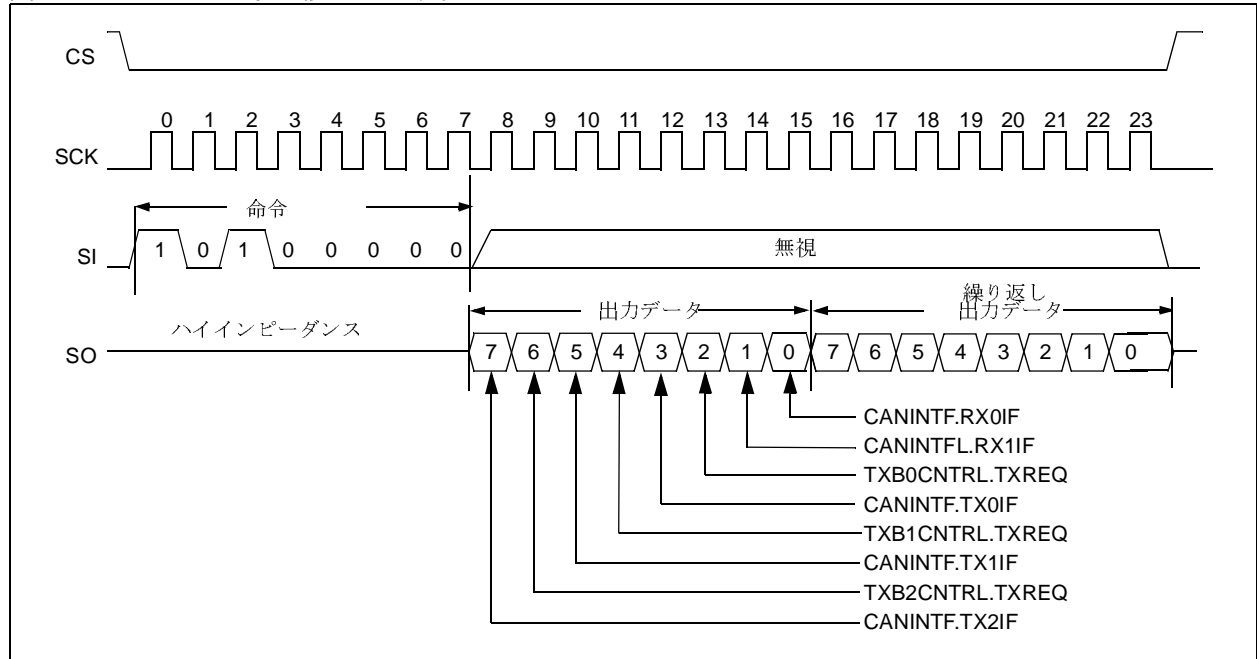
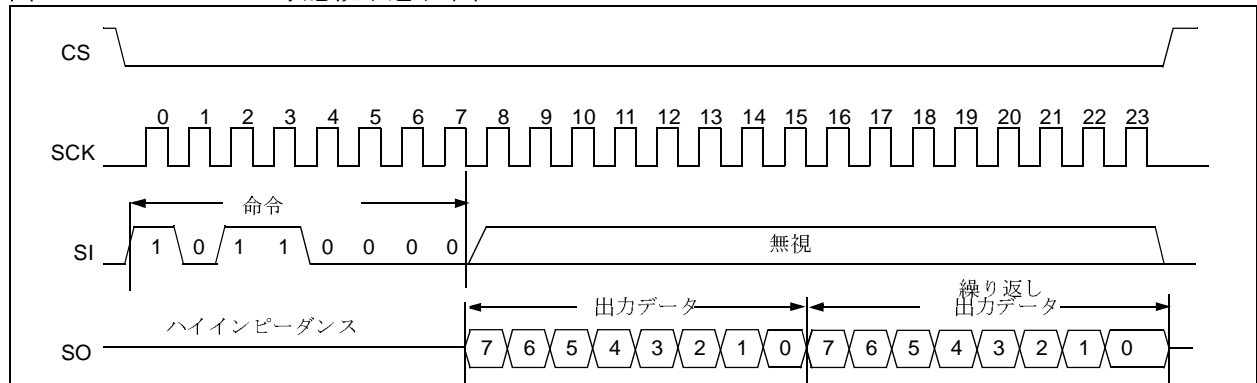


図 12-9: RX 状態読み込み命令



7	6	受信メッセージ
0	0	RX メッセージなし
0	1	RXB0 にメッセージあり
1	0	RXB1 にメッセージあり
1	1	両方にメッセージあり *

CANINTF.RXnIF ビットはビット 7 と 6 にマップされている。

* バッファ 0 が高い優先度をもつ、したがって RXB0 の状態がビット 4:0 に反映される。

4	3	受信メッセージタイプ
0	0	標準データフレーム
0	1	標準リモートフレーム
1	0	拡張データフレーム
1	1	拡張リモートフレーム

拡張 ID ビットはビット 4 に、RTR ビットはビット 3 にマップされている。

2	1	0	一致フィルタ
0	0	0	RXF0
0	0	1	RXF1
0	1	0	RXF2
0	1	1	RXF3
1	0	0	RXF4
1	0	1	RXF5
1	1	0	RXF0 (RXB1 に転送)
1	1	1	RXF1 (RXB1 に転送)

MCP2515

図 12-10: SPI 入力タイミング

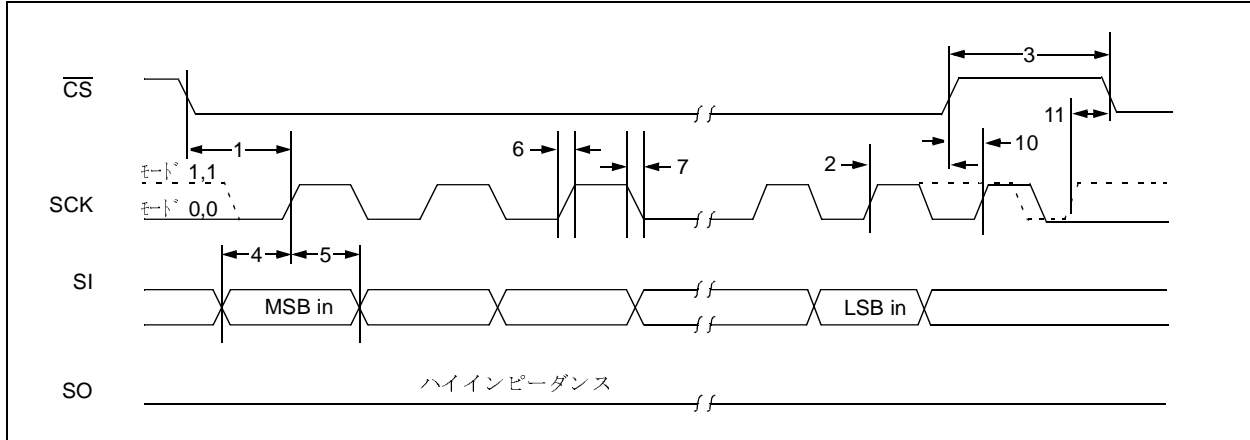
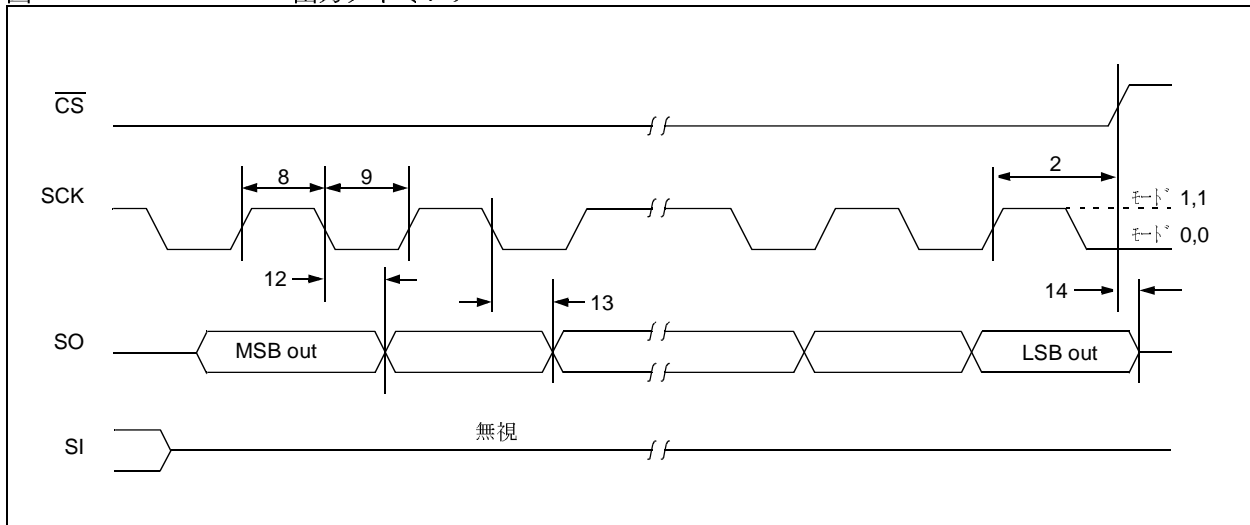


図 12-11: SPI 出力タイミング



13.0 電気的特性

13.1 絶対最大定格・

VDD.....	7.0V
全入出力ピン VSS に対して.....	-0.6V ~ VDD +1.0V
保存温度.....	-65°C ~ +150°C
動作時周囲温度.....	-65°C ~ +125°C
リードはんだ温度 (10 秒).....	+300°C

†注意: 上記の「最大定格」を超えるストレスを加えると、デバイスに恒久的な損傷を与えることがあります。この規定はストレス定格のみを規定するものであり、この仕様の動作条件に記載する規定値以上でのデバイス動作を定めたものではありません。長時間デバイスを最大定格状態にすると、デバイスの信頼性に影響を与えることがあります。

MCP2515

表 13-1: DC 特性

DC 特性			工業用 (I): 拡張品 (E):		TAMB = -40°C ~ +85°C TAMB = -40°C ~ +125°C		VDD = 2.7V ~ 5.5V VDD = 4.5V ~ 5.5V	
パラメータ No.	記号	特性	Min	Max	単位	条 件		
	VDD	電源電圧	2.7	5.5	V			
	VRET	レジスタ保持電圧	2.4	—	V			
	VIH	High レベル入力電圧						
		RXCAN	2	VDD + 1	V			
		SCK, $\overline{\text{CS}}$, SI, $\overline{\text{TXnRTS}}$ ピン	0.7 VDD	VDD + 1	V			
		OSC1	0.85 VDD	VDD	V			
		RESET	0.85 VDD	VDD	V			
	VIL	Low レベル入力電圧						
		RXCAN, $\overline{\text{TXnRTS}}$ ピン	-0.3	.15 VDD	V			
		SCK, $\overline{\text{CS}}$, SI	-0.3	0.4	V			
		OSC1	VSS	.3 VDD	V			
		RESET	VSS	.15 VDD	V			
	VOL	Low レベル出力電圧						
		TXCAN	—	0.6	V	IOL = +6.0 mA, VDD = 4.5V		
		$\overline{\text{RXnBF}}$ ピン	—	0.6	V	IOL = +8.5 mA, VDD = 4.5V		
		SO, CLKOUT	—	0.6	V	IOL = +2.1 mA, VDD = 4.5V		
		$\overline{\text{INT}}$	—	0.6	V	IOL = +1.6 mA, VDD = 4.5V		
	VOH	High レベル出力電圧						
		TXCAN, $\overline{\text{RXnBF}}$ ピン	VDD - 0.7	—	V	IOH = -3.0 mA, VDD = 4.5V		
		SO, CLKOUT	VDD - 0.5	—	V	IOH = -400 μ A, VDD = 4.5V		
		$\overline{\text{INT}}$	VDD - 0.7	—	V	IOH = -1.0 mA, VDD = 4.5V		
	ILI	入力リーク電流						
		全 I/O, OSC1 と TXnRTS ピンを除く	-1	+1	μ A	$\overline{\text{CS}} = \overline{\text{RESET}} = \text{VDD}$, VIN = VSS ~ VDD		
		OSC1 ピン	-5	+5	μ A			
	CINT	内部容量 (全入力と出力)	—	7	pF	TAMB = 25°C, fC = 1.0 MHz, VDD = 0V (注 1)		
	IDD	動作電流	—	10	mA	VDD = 5.5V, FOSC = 25 MHz, FCLK = 1 MHz, SO = オープン		
	IDDS	スタンバイ電流 (スリープモード)	—	5	μ A	$\overline{\text{CS}}$, $\overline{\text{TXnRTS}} = \text{VDD}$, 入力は VDD か VSS に接続, -40°C ~ +85°C		
			—	8	μ A	$\overline{\text{CS}}$, $\overline{\text{TXnRTS}} = \text{VDD}$, 入力は VDD か VSS に接続, -40°C ~ +125°C		

注 1: このパラメータは周期的なサンプル値で 100% はテストされていない。

表 13-2: 発振器タイミング特性

発振器タイミング特性 (注)			工業用 (I):		T _{AMB} = -40°C ~ +85°C		V _{DD} = 2.7V ~ 5.5V	
			拡張品 (E):		T _{AMB} = -40°C ~ +125°C		V _{DD} = 4.5V ~ 5.5V	
パラメータ No.	記号	特性	Min	Max	単位	条件		
	FOSC	クロック入力周波数	1	40	MHz	4.5V ~ 5.5V		
			1	25	MHz	2.7V ~ 5.5V		
	TOSC	クロック入力周期	25	1000	ns	4.5V ~ 5.5V		
			40	1000	ns	2.7V ~ 5.5V		
	TDUTY	デューティサイクル (拡張クロック入力)	0.45	0.55	—	TOSH/(TOSH + TosL)		

注: このパラメータは周期的なサンプル値で 100% はテストされていない。

表 13-3: CAN インターフェース AC 特性

CAN インターフェース AC 特性			工業用 (I):		T _{AMB} = -40°C ~ +85°C		V _{DD} = 2.7V ~ 5.5V	
			拡張品 (E):		T _{AMB} = -40°C ~ +125°C		V _{DD} = 4.5V ~ 5.5V	
パラメータ No.	記号	特性	Min	Max	単位	条件		
	TWF	ウェイクアップノイズフィルタ	100	—	ns			

表 13-4: リセット AC 特性

リセット AC 特性			工業用 (I):		T _{AMB} = -40°C ~ +85°C		V _{DD} = 2.7V ~ 5.5V	
			拡張品 (E):		T _{AMB} = -40°C ~ +125°C		V _{DD} = 4.5V ~ 5.5V	
パラメータ No.	記号	特性	Min	Max	単位	条件		
	trl	RESET ピン Low 時間	2	—	μs			

MCP2515

表 13-5: CLKOUT ピン AC 特性

CLKOUT ピン AC/DC 特性			工業用 (I):	T _{AMB} = -40°C ~ +85°C	V _{DD} = 2.7V ~ 5.5V	
			拡張品 (E):	T _{AMB} = -40°C ~ +125°C	V _{DD} = 4.5V ~ 5.5V	
パラメータ No.	記号	特性	Min	Max	単位	条件
	t _h CLKOUT	CLKOUT ピン High 時間	15	—	ns	T _{OSC} = 40 ns (注 1)
	t _l CLKOUT	CLKOUT ピン Low 時間	15	—	ns	T _{OSC} = 40 ns (注 1)
	t _r CLKOUT	CLKOUT ピン立上り時間	—	5	ns	0.3 V _{DD} ~ 0.7 V _{DD} で測定 (注 1)
	t _f CLKOUT	CLKOUT ピン立下り時間	—	5	ns	0.7 V _{DD} ~ 0.3 V _{DD} で測定 (注 1)
	t _d CLKOUT	CLOCKOUT 伝播遅延	—	100	ns	注 1
15	t _h SOF	Start-Of-Frame の High 時間	—	2 T _{OSC}	ns	注 1
16	t _d SOF	Start-Of-Frame の伝播遅延	—	2 T _{OSC} + 0.5 T _Q	ns	CAN ビットサンプル点から測定 デバイスは受信器 CNF1.BRP<5:0> = 0 (注 2)

- 図 13-1: デバイス周波数限界の全 CLKOUT モード機能と出力周波数でテスト済み。ただし CLKOUT 分周は 1/1。このパラメータは周期的なサンプル値で 100% はテストされていない。
- 2: 設計ガイド用のみ、未テスト

図 13-1: START-OF-FRAME ピン AC 特性

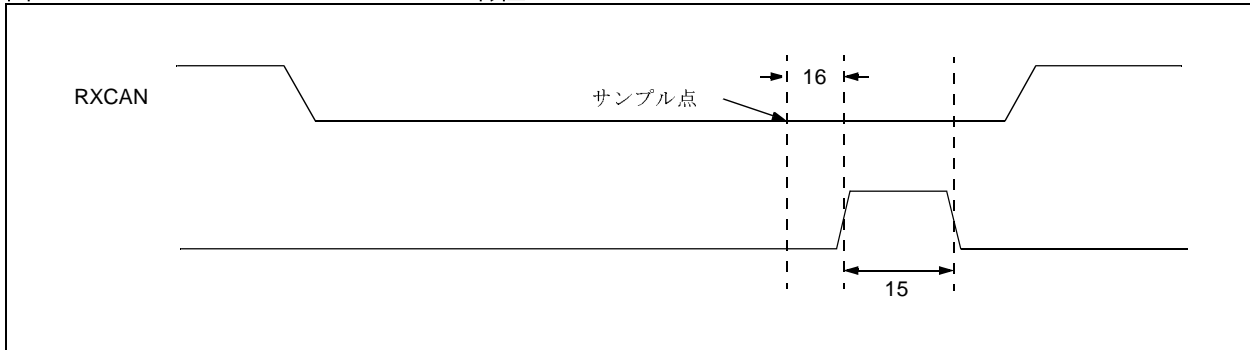


表 13-6: SPI インターフェース AC 特性

SPI™ インターフェース AC 特性			工業用 (I): 拡張品 (E):		T _{AMB} = -40°C ~ +85°C T _{AMB} = -40°C ~ +125°C	V _{DD} = 2.7V ~ 5.5V V _{DD} = 4.5V ~ 5.5V
パラメータ No.	記号	特性	Min	Max	単位	条件
	FCLK	クロック周波数	—	10	MHz	
1	T _{CSS}	\overline{CS} セットアップ時間	50	—	ns	
2	T _{CSH}	\overline{CS} 保持時間	50	—	ns	
3	T _{CSD}	\overline{CS} 禁止時間	50	—	ns	
4	T _{SU}	データセットアップ時間	10	—	ns	
5	T _{HD}	データ保持時間	10	—	ns	
6	T _R	CLK 立上り時間	—	2	μs	注 1
7	T _F	CLK 立下り時間	—	2	μs	注 1
8	T _{HI}	Clock High 時間	45	—	ns	
9	T _{LO}	Clock Low 時間	45	—	ns ns	
10	T _{CLD}	Clock 遅延時間	50	—	ns	
11	T _{CLE}	Clock 許可時間	50	—	ns	
12	T _V	Clock Low からの出力遷移時間	—	45	ns	
13	T _{HO}	出力保持時間	0	—	ns	
14	T _{DIS}	出力禁止時間	—	100	ns	

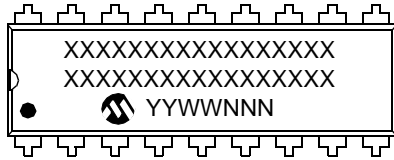
注 1: このパラメータは 100% はテストされていない

MCP2515

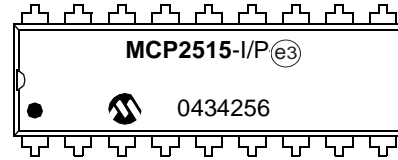
14.0 パッケージ情報

14.1 パッケージマーキング情報

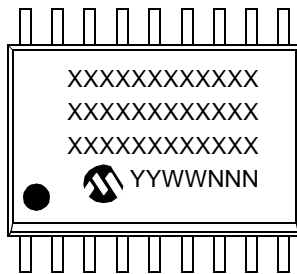
18-Lead PDIP (300 mil)



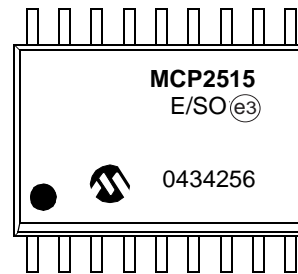
例:



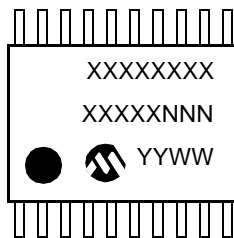
18-Lead SOIC (300 mil)



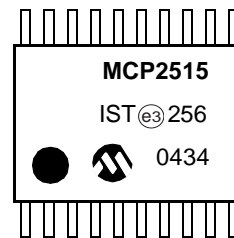
例:



20-Lead TSSOP (4.4 mm)



例:

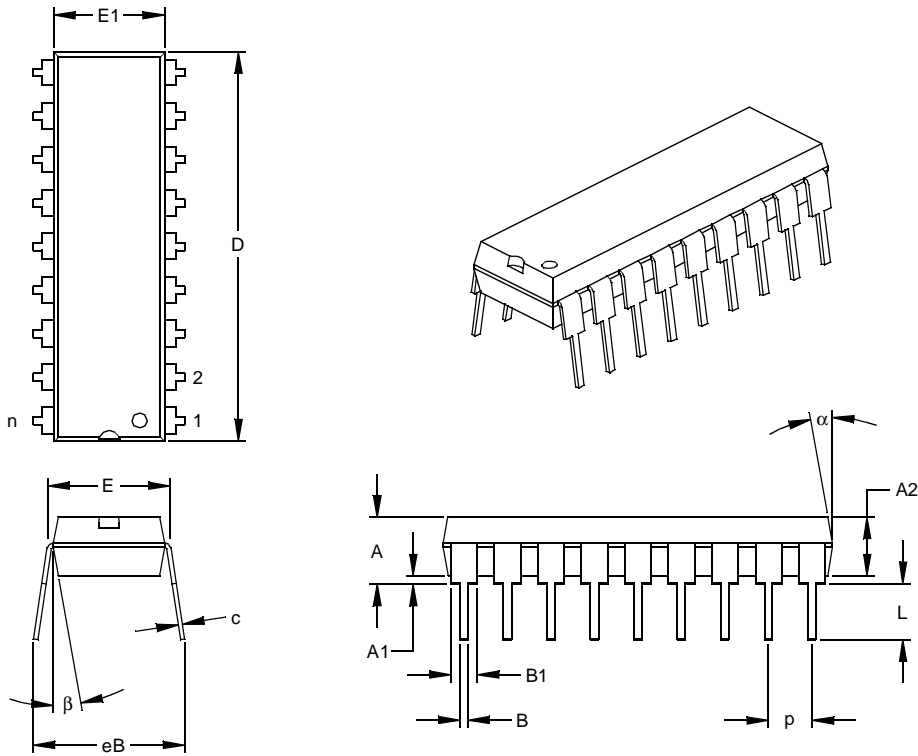


凡例:	XX...X	カスタマ仕様情報
	Y	年コード (カレンダー年の下位 1 桁目)
	YY	年コード (カレンダー年の下位 2 桁目)
	WW	週コード (1 月 1 日を週 '01' とする)
	NNN	英数字のトレース用コード
	(e3)	錫メッキ (Sn) に関する鉛フリー JEDEC 区別コード
	*	本パッケージは鉛フリーです。鉛フリー JEDEC 区別 (e3) はパッケージの外観から見えるようにしています。

注: マイクロチップのパーツ番号全体が 1 行で入らないときは、次の行にはみ出ます。このためカスタマ仕様情報用の文字数が制限されます。

MCP2515

18-Lead Plastic Dual In-line (P) – 300 mil (PDIP)



寸法限界	単位	インチ*			ミリメートル		
		MIN	NOM	MAX	MIN	NOM	MAX
ピン数	n		18			18	
ピッチ	p		.100			2.54	
頂部から実装面	A	.140	.155	.170	3.56	3.94	4.32
モールドパッケージ厚	A2	.115	.130	.145	2.92	3.30	3.68
底面から実装面	A1	.015			0.38		
肩間幅	E	.300	.313	.325	7.62	7.94	8.26
モールドパッケージ幅	E1	.240	.250	.260	6.10	6.35	6.60
全長	D	.890	.898	.905	22.61	22.80	22.99
先端から実装面	L	.125	.130	.135	3.18	3.30	3.43
リード厚	c	.008	.012	.015	0.20	0.29	0.38
リード上部幅	B1	.045	.058	.070	1.14	1.46	1.78
リード下部幅	B	.014	.018	.022	0.36	0.46	0.56
全間隔 §	eB	.310	.370	.430	7.87	9.40	10.92
モールド抜き角頂部	α	5	10	15	5	10	15
モールド抜き角底部	β	5	10	15	5	10	15

* 制御パラメータ

§ 有意特性

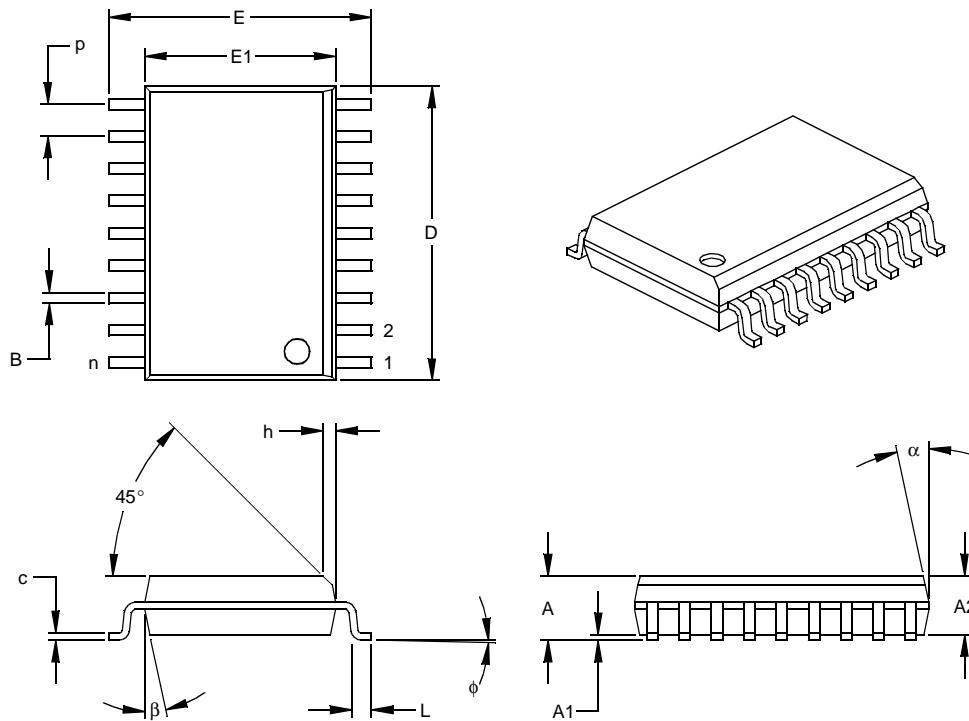
注:

D と E1 の寸法はモールドのはみ出しや突出部を含みません。モールドのはみ出しや突出部は側面から .010" (0.254mm) 以上はありません。

JEDEC 準拠: MS-001

Drawing No. C04-007

18-Lead Plastic Small Outline (SO) – Wide, 300 mil (SOIC)



単位		インチ*			ミリメートル		
寸法限界		MIN	NOM	MAX	MIN	NOM	MAX
ピン数	n		18			18	
ピッチ	p		.050			1.27	
全高	A	.093	.099	.104	2.36	2.50	2.64
モールドパッケージ厚	A2	.088	.091	.094	2.24	2.31	2.39
スタンドオフ §	A1	.004	.008	.012	0.10	0.20	0.30
全幅	E	.394	.407	.420	10.01	10.34	10.67
モールドパッケージ幅	E1	.291	.295	.299	7.39	7.49	7.59
全長	D	.446	.454	.462	11.33	11.53	11.73
面取り長	h	.010	.020	.029	0.25	0.50	0.74
足長	L	.016	.033	.050	0.41	0.84	1.27
足角	φ	0	4	8	0	4	8
リード厚	c	.009	.011	.012	0.23	0.27	0.30
リード幅	B	.014	.017	.020	0.36	0.42	0.51
モールド抜き角頂部	α	0	12	15	0	12	15
モールド抜き角底部	β	0	12	15	0	12	15

* 制御パラメータ

§ 有意特性

注:

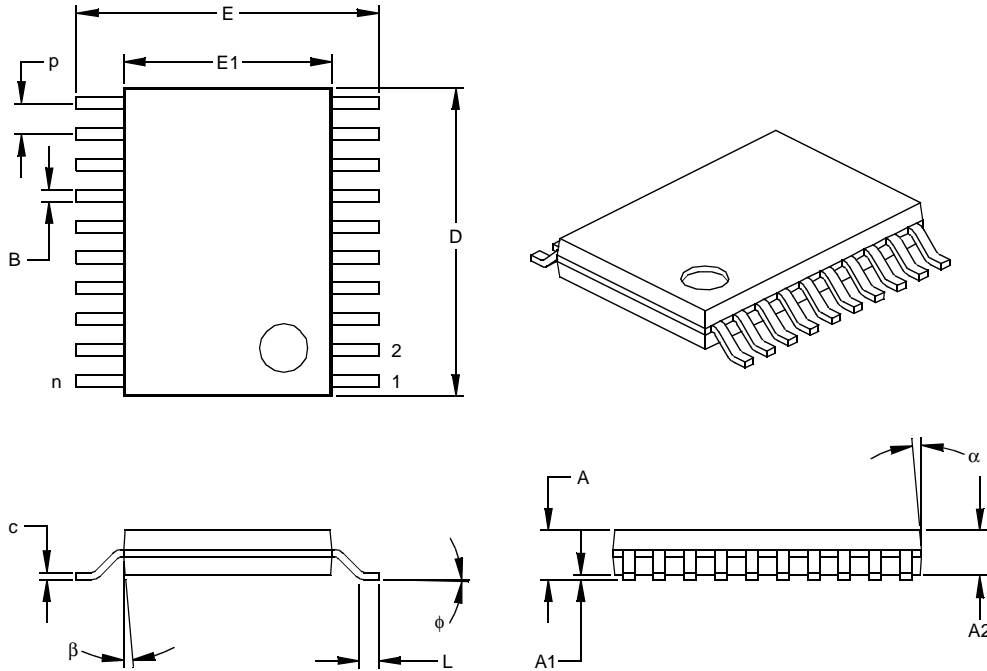
D と E1 の寸法はモールドのはみ出しや突出部を含みません。モールドのはみ出しや突出部は側面から .010" (0.254mm) 以上はありません。

JEDEC 準拠: MS-013

Drawing No. C04-051

MCP2515

20-Lead Plastic Thin Shrink Small Outline (ST) – 4.4 mm (TSSOP)



単位		インチ			ミリメートル*		
寸法限界		MIN	NOM	MAX	MIN	NOM	MAX
ピン数	n		20			20	
ピッチ	p		.026			0.65	
全高	A			.043			1.10
モールドパッケージ厚	A2	.033	.035	.037	0.85	0.90	0.95
スタンドオフ §	A1	.002	.004	.006	0.05	0.10	0.15
全幅	E	.246	.251	.256	6.25	6.38	6.50
モールドパッケージ幅	E1	.169	.173	.177	4.30	4.40	4.50
モールドパッケージ長	D	.252	.256	.260	6.40	6.50	6.60
足長	L	.020	.024	.028	0.50	0.60	0.70
足角	φ	0	4	8	0	4	8
リード厚	c	.004	.006	.008	0.09	0.15	0.20
リード幅	B	.007	.010	.012	0.19	0.25	0.30
モールド抜き角頂部	α	0	5	10	0	5	10
モールド抜き角底部	β	0	5	10	0	5	10

*制御パラメータ

§ 有意特性

注:

D と E1 の寸法はモールドのはみ出しや突出部を含みません。モールドのはみ出しや突出部は側面から .005" (0.127mm) 以上はありません。

JEDEC 準拠: MO-153

Drawing No. C04-088

付録 A: 改版履歴

レビジョン D (2005 年 4 月)

下記リスト部修正:

1. 第 8.0 章、表 8-1 と表 8-2 追加、表下に注を追加
2. 第 11.0 項、表 11-1. コラム見出しのアドレス変更
3. 第 14.0 項パッケージ情報に Pb フリーデバイスマーキング反映
4. 付録レビジョン A 改版履歴追加: 重要度順に並び替え

レビジョン C (2004 年 11 月)

下記リスト部修正:

1. 新たに第 9.0 章追加
2. 第 12 章, 見出し 12.1: 注を追加
見出し 12.6: 段落内の冗長さ変更
3. レビジョン A 追加: 改版履歴

レビジョン B (2003 年 9 月)

下記リスト部修正:

1. 前段ページまとめ: スタンバイ電流 (標準) (スリープモード) を $10\ \mu\text{A}$ から $1\ \mu\text{A}$ に変更
2. 第 8.2 項 CLKOUT ピン: 注を追加 CLKOUT の最高周波数について
3. 第 12.0 章, 表 12-1:
 - 電源電圧最小値変更 2.7V に
 - 内部容量: 条件 V_{DD} を 0V に変更
 - スタンバイ電流 (スリープモード): 仕様を分離 $-40^{\circ}\text{C} \sim +85^{\circ}\text{C}$ と $-40^{\circ}\text{C} \sim +125^{\circ}\text{C}$ に

レビジョン A (2003 年 5 月)

- 初版発行

MCP2515

NOTES:

製品識別システム

注文や資料請求、または価格や納期などの照会は工場もしくは後述のセールスオフィスへお問い合わせください。

PART NO.	-	X	/XX
デバイス		温度範囲	パッケージ
<p>デバイス</p> <p>MCP2515: CAN Controller w/ SPI™ Interface MCP2515T: CAN Controller w/SPI Interface (テープでリール)</p> <p>温度範囲</p> <p>I = -40°C ~ +85°C (工業用) E = -40°C ~ +125°C (拡張品)</p> <p>パッケージ</p> <p>P = Plastic DIP (300 mil Body), 18-Lead SO = Plastic SOIC (300 mil Body), 18-Lead ST = TSSOP, (4.4 mm Body), 20-Lead</p>			<p>Examples:</p> <p>a) MCP2515-E/P: 拡張温度品, 18LD PDIP パッケージ</p> <p>b) MCP2515-I/P: 工業温度品, 18LD PDIP パッケージ</p> <p>c) MCP2515-E/SO: 拡張温度品, 18LD SOIC パッケージ</p> <p>d) MCP2515-I/SO: I 工業温度品, 18LD SOIC パッケージ</p> <p>e) MCP2515T-I/SO: テープでリール, 工業温度品, 18LD SOIC パッケージ</p> <p>f) MCP2515-I/ST: 工業温度品, 20LD TSSOP パッケージ</p> <p>g) MCP2515T-I/ST: T テープでリール, 工業温度品, 20LD TSSOP パッケージ</p>

MCP2515

NOTES:

マイクロチップデバイスのコード保護機能に関する以下の点に留意ください。

- マイクロチップの製品は各製品独自のマイクロチップデータシートにある仕様を満たしています。
- 各製品ファミリーは、通常の状態ですべての方法で利用いただければ市場にある類似製品の中で最も安全なファミリーの一つとマイクロチップは信じております。
- 不正かつ非合法な方法を使ったコード保護機能の侵害があります。弊社の理解ではこうした手法は、マイクロチップデータシートにある動作仕様書以外の方法でマイクロチップ製品を使用することになります。こうした手法を使用した人は、ほとんどの場合、知的財産権の侵害となります。
- マイクロチップはコードの統合性に関心をお持ちの顧客とは協働させていただきます。
- マイクロチップまたは他のセミコンダクターメーカーがコードの安全性を保証したものではありません。コード保護は製品保護が「破られない」ということを保証するものではありません。

コード保護は常に進化します。マイクロチップは、当社製品のコード保護機能を継続的に改善することをお約束いたします。マイクロチップのコード保護機能を破ることは、デジタル・ミレニアム著作権法に違反します。こうした行為によるソフトウェアや著作権に関わる作品への不正アクセスがあった場合、同法に基づき賠償請求する権利があります。

本書の日本語版はユーザーの使用のために提供されます。**Microchip Technology Inc.** とその子会社、関連会社、すべての取締役、役員、職員、代理人は翻訳の間違いにより起こるいかなる責も負わないものとします。間違いが疑われる箇所については、**Microchip Technology Inc.** 発行のオリジナル文書を参照いただくようお願いいたします。

本書に書かれているデバイスアプリケーション等に関する内容は、参考情報に過ぎません。ご利用のアプリケーションが仕様を満たしているかどうかについては、お客様の責任において確認をお願いします。これらの情報の正確さ、またはこれらの情報の使用に関し、マイクロチップテクノロジーインクはいかなる表明と保証を行うものではなく、また、一切の責任を負うものではありません。マイクロチップの明示的な書面による承認なしに、生命維持装置あるいは生命安全用途にマイクロチップの製品を使用することはすべて購入者のリスクとし、また購入者はこれによって起きたあらゆる損害、クレーム、訴訟、費用に関して、マイクロチップは擁護され、免責され、損害をうけないことに同意するものとします。知的財産権に基づくライセンスを暗示的に与えたものではありません。

商標

マイクロチップの名称とロゴ、マイクロチップのロゴ、Accuron、dsPIC、KEELOQ、microID、MPLAB、PIC、PICmicro、PICSTART、PRO MATE、PowerSmart、rfPIC、SmartShunt は米国及び他の国々において、マイクロチップテクノロジーインク の登録商標です。

AmpLab、FilterLab、Migratable Memory、MXDEV、MXLAB、PICMASTER、SEEVAL、SmartSensor、The Embedded Control Solutions Company は、米国においてマイクロチップテクノロジーインク の登録商標です。

Analog-for-the-Digital Age、Application Maestro、dsPICDEM、dsPICDEM.net、dsPICworks、ECAN、ECONOMONITOR、FanSense、FlexROM、fuzzylab、In-Circuit Serial Programming、ICSP、ICEPIC、Linear Active Thermistor、MPASM、MPLIB、MPLINK、MPSIM、PICkit、PICDEM、PICDEM.net、PICLAB、PICtail、PowerCal、PowerInfo、PowerMate、PowerTool、rfLAB、rfPICDEM、Select Mode、Smart Serial、SmartTel、Total Endurance、UNI/O、WiperLock、及び Zena は、米国及び他の国々において、マイクロチップテクノロジーインク の登録商標とです。

SQTP は米国においてマイクロチップテクノロジーインク のサービスマークです。

本書に記載された上記以外の商標は、それぞれの会社の財産です。

著作権。© 2005 年マイクロチップテクノロジーインク、米国で印刷。無断複写・転載を禁じます。

 再生紙を使用。

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
== ISO/TS 16949:2002 ==

マイクロチップは、10S/TS-16949 を受けました。本社、アリゾナ州チャンドラーとテンペとカリフォルニア州マウンテンビューにあるデザイン及びウエハー施設に対する 2003 年 10 月品質システム認証です。弊社の品質システムプロセスと手続きは、PICmicro® 8-bit MCUs、KEELOQ® コードホッピングデバイス、シリアル EEPROMs、マイクロペリフェラル、非揮発性メモリーとアナログ製品を対象としています。更に、開発システムの設計及び製造に関するマイクロチップの品質システムは、2000 年に ISO9001 の認証を受けています。



MICROCHIP

全世界の販売及びサービス拠点

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
<http://support.microchip.com>
Web Address:
www.microchip.com

Atlanta

Alpharetta, GA
Tel: 770-640-0034
Fax: 770-640-0307

Boston

Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago

Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Dallas

Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit

Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Kokomo

Kokomo, IN
Tel: 765-864-8360
Fax: 765-864-8387

Los Angeles

Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

San Jose

Mountain View, CA
Tel: 650-215-1444
Fax: 650-961-0286

Toronto

Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8676-6200
Fax: 86-28-8676-6599

China - Fuzhou
Tel: 86-591-8750-3506
Fax: 86-591-8750-3521

China - Hong Kong SAR
Tel: 852-2401-1200
Fax: 852-2401-3431

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

China - Shunde
Tel: 86-757-2839-5507
Fax: 86-757-2839-5571

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7250
Fax: 86-29-8833-7256

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-4182-8400
Fax: 91-80-4182-8422

India - New Delhi
Tel: 91-11-5160-8631
Fax: 91-11-5160-8632

India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

Japan - Yokohama
Tel: 81-45-471-6166
Fax: 81-45-471-6122

Korea - Gumi
Tel: 82-54-473-4301
Fax: 82-54-473-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Penang
Tel: 60-4-646-8870
Fax: 60-4-646-5086

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-572-9526
Fax: 886-3-572-6459

Taiwan - Kaohsiung
Tel: 886-7-536-4818
Fax: 886-7-536-4803

Taiwan - Taipei
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-399
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands - Druenen
Tel: 31-416-690399
Fax: 31-416-690340

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

03/01/05