



**MICROCHIP**

---

**PIC18F97J60 ファミリ  
データシート**

**64/80/100 ピン高性能  
1M ビット フラッシュ マイクロ  
コントローラ Ethernet モジュール内蔵**

注意：この日本語版文書は参考資料としてご利用ください。最新情報は必ずオリジナルの英語版をご参照願います。

---

マイクロチップ社製デバイスのコード保護機能に関して以下の点にご注意ください。

- マイクロチップ社製品は、該当するマイクロチップ社データシートに記載の仕様を満たしています。
- マイクロチップ社では、通常の条件ならびに仕様に従って使用した場合、マイクロチップ社製品のセキュリティレベルは、現在市場に流通している同種製品の中でも最も高度であると考えています。
- しかし、コード保護機能を解除するための不正かつ違法な方法が存在する事もまた事実です。弊社の理解では、こうした手法はマイクロチップ社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社製品を使用する事になります。このような行為は知的所有権の侵害に該当する可能性が非常に高いと言えます。
- マイクロチップ社は、コードの保全性に懸念を抱いているお客様と連携し、対応策に取り組んでいきます。
- マイクロチップ社を含む全ての半導体メーカーで、自社のコードのセキュリティを完全に保証できる企業はありません。コード保護機能とは、マイクロチップ社が製品を「解読不能」として保証するものではありません。

コード保護機能は常に進歩しています。マイクロチップ社では、常に製品のコード保護機能の改善に取り組んでいます。マイクロチップ社のコード保護機能の侵害は、デジタル ミレニアム著作権法に違反します。そのような行為によってソフトウェアまたはその他の著作物に不正なアクセスを受けた場合、デジタル ミレニアム著作権法の定めるところにより損害賠償訴訟を起こす権利があります。

---

本書に記載されているデバイス アプリケーション等に関する情報は、ユーザの便宜のためにのみ提供されているものであり、更新によって無効とされる事があります。お客様のアプリケーションが仕様を満たす事を保証する責任は、お客様にあります。マイクロチップ社は、明示的、暗黙的、書面、口頭、法定のいずれであるかを問わず、本書に記載されている情報に関して、状態、品質、性能、商品性、特定目的への適合性をはじめとする、いかなる類の表明も保証も行いません。マイクロチップ社は、本書の情報およびその使用に起因する一切の責任を否認します。マイクロチップ社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社の製品を使用する事は全て購入者のリスクとし、また購入者はこれによって発生したあらゆる損害、クレーム、訴訟、費用に関して、マイクロチップ社は擁護され、免責され、損害を受けない事に同意するものとします。暗黙的あるいは明示的を問わず、マイクロチップ社が知的財産権を保有しているライセンスは一切譲渡されません。

#### 商標

マイクロチップ社の名称とロゴ、Microchip ロゴ、dsPIC、FlashFlex、KEELOQ、KEELOQ ロゴ、MPLAB、PIC、PICmicro、PICSTART、PIC<sup>32</sup> ロゴ、rfPIC、SST、SST ロゴ、SuperFlash、UNI/O は、米国およびその他の国におけるマイクロチップ・テクノロジー社の登録商標です。

FilterLab、Hampshire、HI-TECH C、Linear Active Thermistor、MTP、SEEVAL、Embedded Control Solutions Company は、米国におけるマイクロチップ・テクノロジー社の登録商標です。

Silicon Storage Technology は、他の国におけるマイクロチップ・テクノロジー社の登録商標です。

Analog-for-the-Digital Age、Application Maestro、BodyCom、chipKIT、chipKIT ロゴ、CodeGuard、dsPICDEM、dsPICDEM.net、dsPICworks、dsSPEAK、ECAN、ECONOMONITOR、FanSense、HI-TIDE、In-Circuit Serial Programming、ICSP、Mindi、MiWi、MPASM、MPF、MPLAB Certified ロゴ、MPLIB、MPLINK、mTouch、Omniscient Code Generation、PICC、PICC-18、PICDEM、PICDEM.net、PICkit、PICtail、REAL ICE、rfLAB、Select Mode、SQL、Serial Quad I/O、Total Endurance、TSHARC、UniWinDriver、WiperLock、ZENA および Z-Scale は、米国およびその他の国におけるマイクロチップ・テクノロジー社の商標です。

SQTP は、米国におけるマイクロチップ・テクノロジー社のサービスマークです。

GestIC および ULPP は、マイクロチップ・テクノロジー社の子会社である Microchip Technology Germany II GmbH & Co. & KG 社の他の国における登録商標です。

その他、本書に記載されている商標は各社に帰属します。

© 2013, Microchip Technology Incorporated, All Rights Reserved.

ISBN: 978-1-62076-282-0

**QUALITY MANAGEMENT SYSTEM**  
**CERTIFIED BY DNV**  
**== ISO/TS 16949 ==**

マイクロチップ社では、Chandler および Tempe (アリゾナ州)、Gresham (オレゴン州)の本拠地、設計部およびウェハー製造工場としてカリフォルニア州とインドのデザインセンターがISO/TS-16949:2009 認証を取得しています。マイクロチップ社の品質システム プロセスおよび手順は、PIC<sup>®</sup> MCU および dsPIC<sup>®</sup> DSC、KEELOQ<sup>®</sup> コードホッピングデバイス、シリアルEEPROM、マイクロベリフェラル、不揮発性メモリ、アナログ製品に採用されています。さらに、開発システムの設計と製造に関するマイクロチップ社の品質システムはISO 9001:2000 認証を取得しています。

## 64/80/100 ピン高性能 1M ビット フラッシュ マイクロコントローラ、Ethernet モジュール内蔵

### Ethernet 機能：

- IEEE 802.3™ 準拠の Ethernet コントローラ
- 10/100/1000Base-T ネットワークと完全互換
- MAC および 10Base-T PHY を内蔵
- 8 KB 送受信パケットバッファ SRAM
- 10Base-T ポートを 1 つサポート
- コリジョン発生時の自動再送がプログラマブル
- パディングと CRC 生成がプログラマブル
- エラーパケットの自動拒否がプログラマブル
- アクティビティを表す 2 つの LED インジケータ出力
- バッファ：
  - 送受信バッファサイズを設定可能
  - ハードウェア管理の環状 FIFO 受信バッファ
  - バイト単位のランダムおよび順次アクセス
  - 内部 DMA による高速メモリコピー
  - 各種プロトコル用のチェックサム計算をハードウェアで支援
- MAC:
  - ユニキャスト、マルチキャスト、ブロードキャストパケットをサポート
  - ユーザ定義オフセットでのパケット内の最大 64 バイトに対するパターンマッチングがプログラマブル
  - 複数のパケットフォーマットによる復帰がプログラマブル
- PHY:
  - 波形シェーピング出力フィルタ

### 柔軟なオシレータ構造：

- 25 MHz の単一外部クロック源から広範囲のシステムクロックを生成可能：
  - 2.778 ~ 41.667 MHz
- 内部 31 kHz オシレータ
- 32 kHz の Timer1 を使うセカンダリ オシレータ
- フェイルセーフクロックモニタ：
  - オシレータ停止時に安全にシャットダウン可能
- オシレータ 2 段階起動

### 外部メモリバス (100 ピンデバイスのみ)：

- 最大 2 MB のアドレス空間
- 8/16 ビット インターフェイス
- 12/16/20 ビット アドレッシング モード

### 周辺機能の特長：

- 大電流のシンク / ソース：  
PORTB と PORTC で 25 mA/25 mA

- 5 つのタイマモジュール (Timer0 ~ Timer4)
- 4 つの外部割り込みピン
- 2 つのキャプチャ / コンペア / PWM (CCP) モジュール
- 3 つの拡張キャプチャ / コンペア / PWM (ECCP) モジュール：
  - 1、2、4 ついずれかの PWM 出力
  - 極性が選択可能
  - デッドタイムがプログラマブル
  - 自動シャットダウンと自動再起動
- SPI (4 つのモード全て) と I<sup>2</sup>C™ マスタ / スレーブモードをサポートした最大 2 つのマスタ同期シリアルポート (MSSP) モジュール
- 最大 2 つの拡張 USART モジュール：
  - RS-485、RS-232、LIN/J2602 をサポート
  - Start ビットによる自動復帰
  - 自動 baud レート検出 (ABD)
- 10 ビット アナログ / デジタル (A/D) コンバータモジュール (最大 16 チャンネル)：
  - 自動アキュイジション機能
  - スリープ時も変換可能
- 入力マルチプレクサを持つ 2 つのアナログコンパレータ
- パラレルスレーブポート (PSP) モジュール (100 ピンデバイスのみ)

### その他の特長：

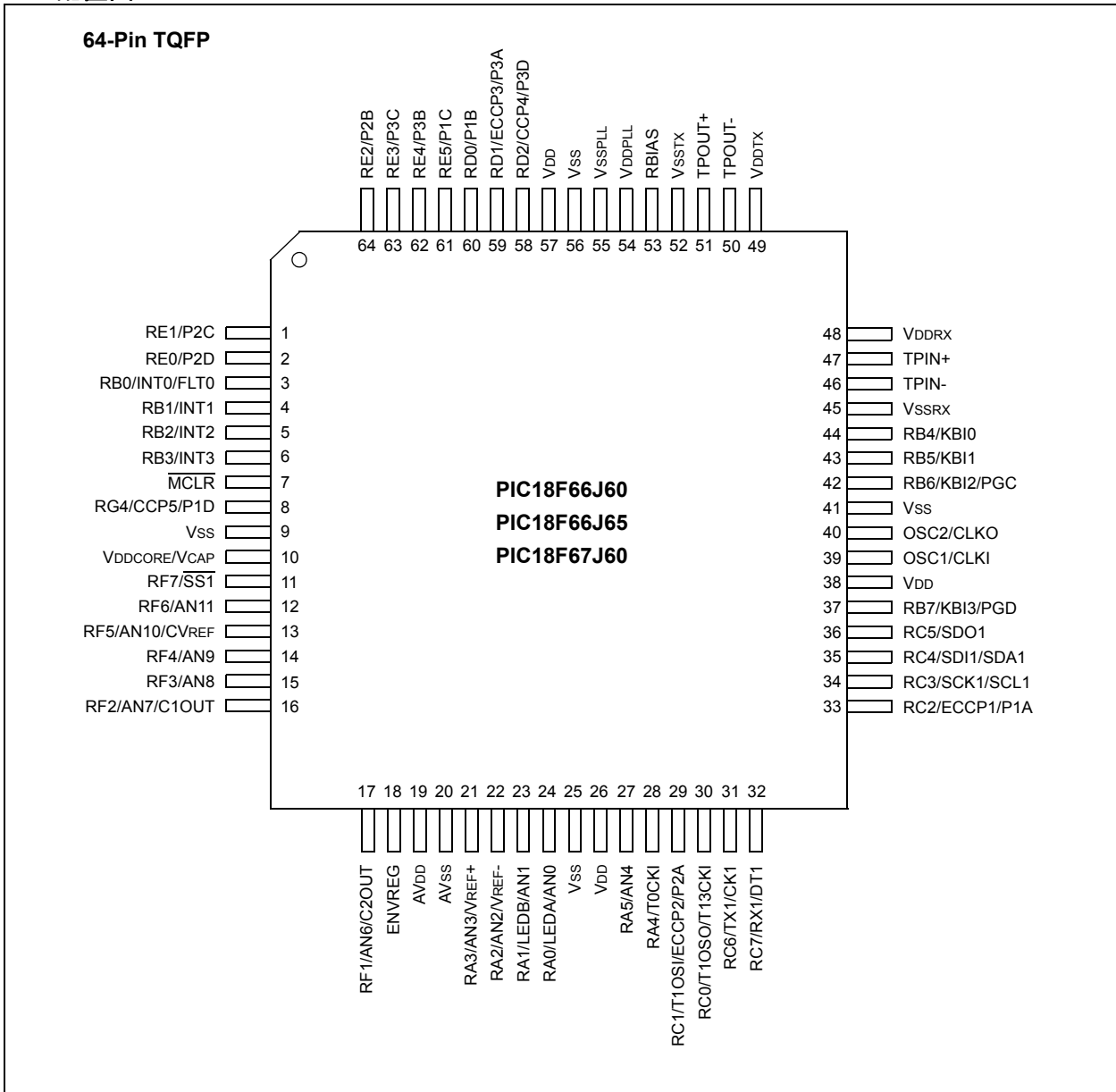
- 5.5 V 耐圧の入力ピン (デジタル専用ピン)
- 低消費電力、高速の CMOS フラッシュ技術：
  - ソフトウェアで自己プログラムが可能
- 再入可能なコード向け C コンパイラに最適化されたアーキテクチャ
- 省電力機能：
  - ランモード：CPU と周辺モジュールの両方が動作
  - アイドル：CPU のみ動作を停止し、周辺モジュールは動作
  - スリープ：CPU と周辺モジュールの両方が動作を停止
- 優先度付きの割り込み
- 8 x 8 シングルサイクルのハードウェア乗算器
- 拡張ウォッチドッグ タイマ (WDT)：
  - 周期は 4 ms ~ 134s の間でプログラミング可能
- 2 本のピンを使う 3.3 V 単電源インサーキットシリアルプログラミング™ (ICSP™)
- 2 本のピンを使い、3 つのブレイクポイントを備えたインサーキットデバッグ (ICD)
- 2.35 ~ 3.6 V の動作電圧レンジ (Ethernet モジュールを使う場合は 3.1 ~ 3.6 V)
- 2.5 V 内蔵電圧レギュレータ

# PIC18F97J60 ファミリ

デバイス名	フラッシュ プログラム メモリ (バイト)	SRAM データ メモリ (バイト)	Ethernet TX/RX バッファ (バイト)	I/O	10ビット A/D (ch)	CCP/ ECCP	MSSP		EUSART	コンパレータ	タイマ 8/16 ビット	PSP	外部 メモリバス	
							SPI	マスタ I <sup>2</sup> C™						
PIC18F66J60	64K	3808	8192	39	11	2/3	1	Yes	Yes	1	2	2/3	No	No
PIC18F66J65	96K	3808	8192	39	11	2/3	1	Yes	Yes	1	2	2/3	No	No
PIC18F67J60	128K	3808	8192	39	11	2/3	1	Yes	Yes	1	2	2/3	No	No
PIC18F86J60	64K	3808	8192	55	15	2/3	1	Yes	Yes	2	2	2/3	No	No
PIC18F86J65	96K	3808	8192	55	15	2/3	1	Yes	Yes	2	2	2/3	No	No
PIC18F87J60	128K	3808	8192	55	15	2/3	1	Yes	Yes	2	2	2/3	No	No
PIC18F96J60	64K	3808	8192	70	16	2/3	2	Yes	Yes	2	2	2/3	Yes	Yes
PIC18F96J65	96K	3808	8192	70	16	2/3	2	Yes	Yes	2	2	2/3	Yes	Yes
PIC18F97J60	128K	3808	8192	70	16	2/3	2	Yes	Yes	2	2	2/3	Yes	Yes

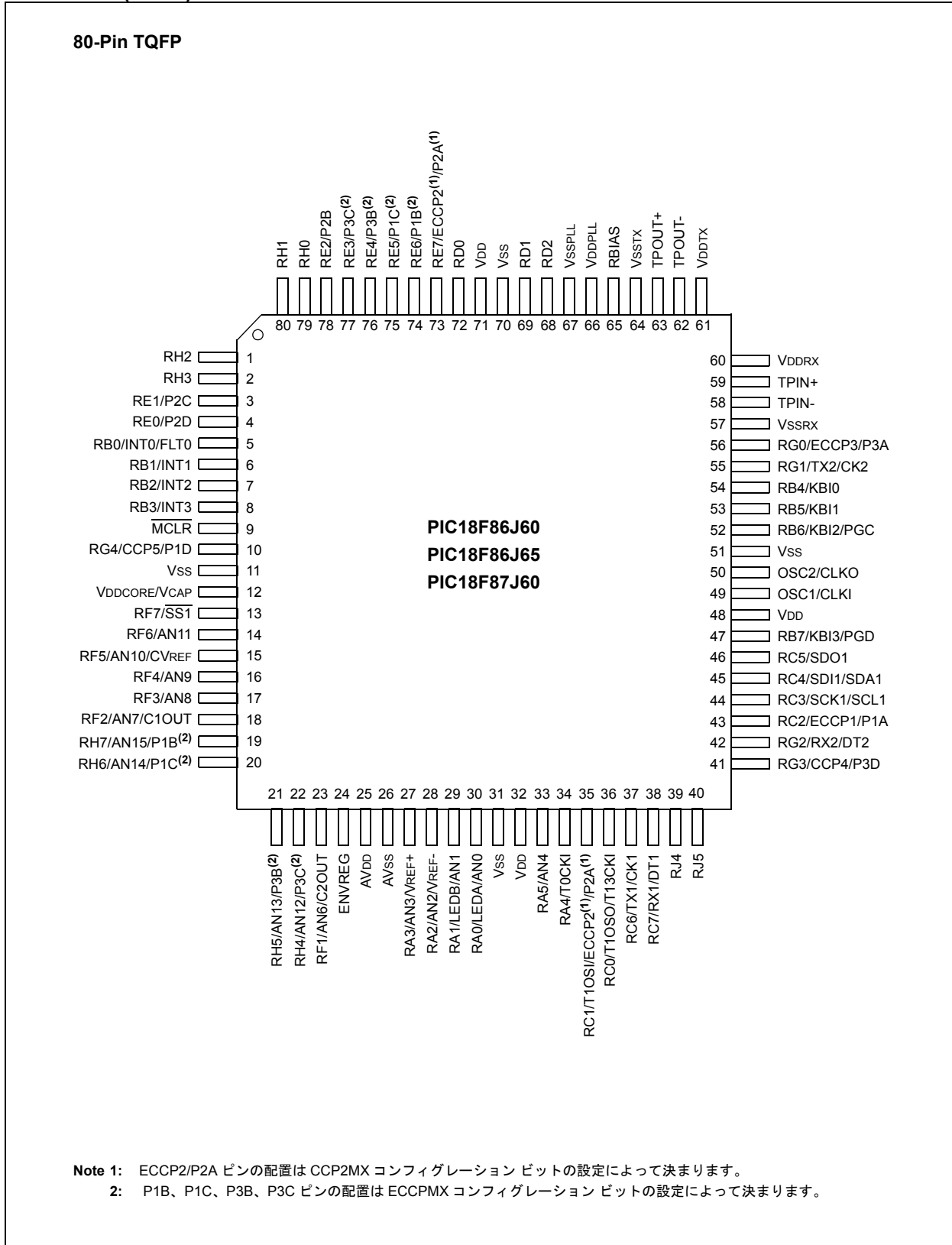
# PIC18F97J60 ファミリ

## ピン配置図



# PIC18F97J60 ファミリ

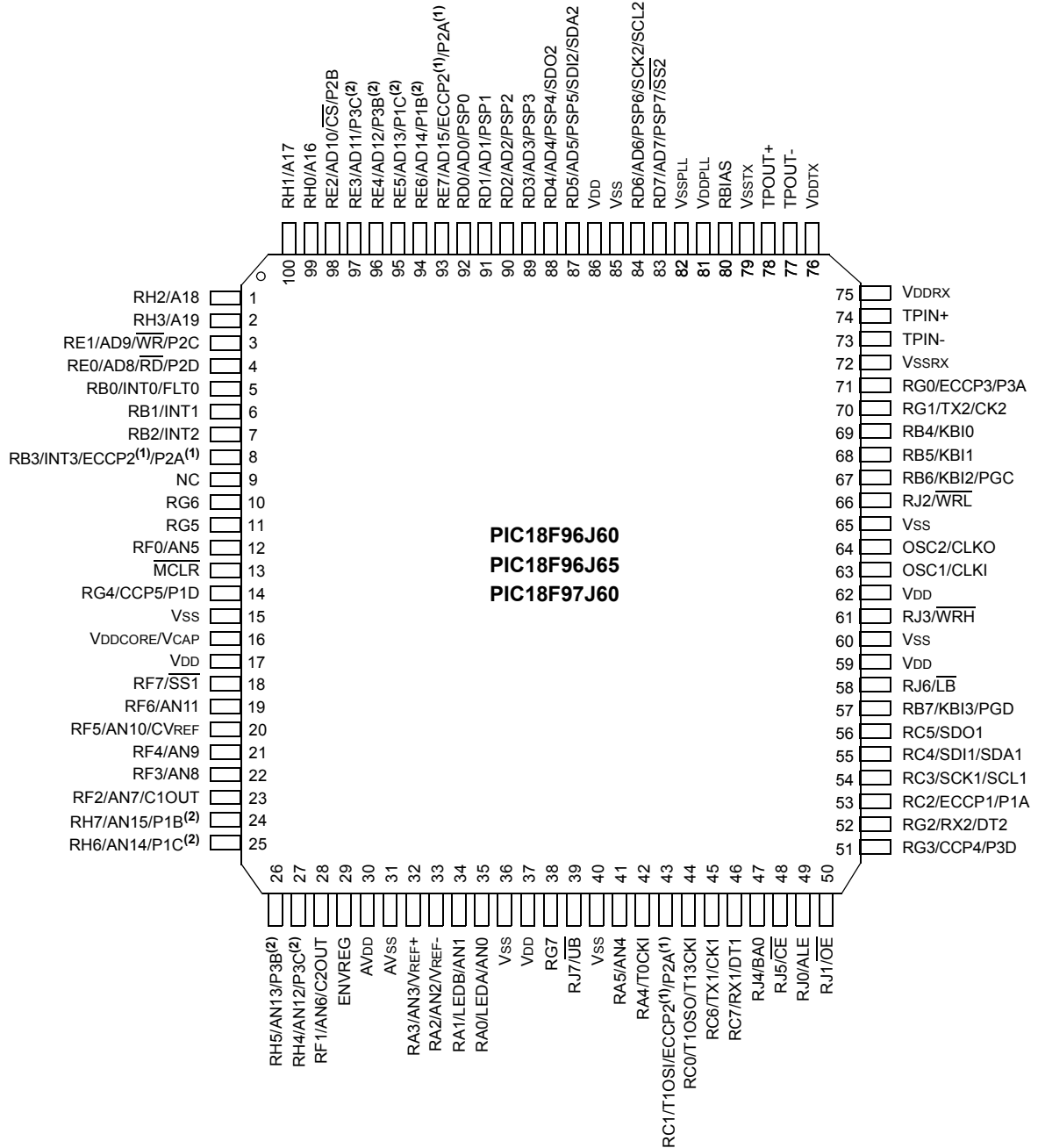
## ピン配置図 (続き)



# PIC18F97J60 ファミリ

## ピン配置図 (続き)

100-Pin TQFP



**Note 1:** ECCP2/P2A ピンの配置は CCP2MX コンフィグレーション ビットとプロセッサモードの設定によって決まります。

**2:** P1B、P1C、P3B、P3C ピンの配置は ECCPMX コンフィグレーション ビットの設定によって決まります。

# PIC18F97J60 ファミリ

---

## 目次

1.0	デバイス概要	11
2.0	PIC18FJ マイクロコントローラの使用にあたってのガイドライン	43
3.0	オシレータの設定	49
4.0	電力管理モード	55
5.0	リセット	63
6.0	メモリ構成	77
7.0	フラッシュ プログラムメモリ	105
8.0	外部メモリバス	115
9.0	8 x 8 ハードウェア乗算器	127
10.0	割り込み	129
11.0	I/O ポート	145
12.0	Timer0 モジュール	171
13.0	Timer1 モジュール	175
14.0	Timer2 モジュール	180
15.0	Timer3 モジュール	183
16.0	Timer4 モジュール	187
17.0	キャプチャ / コンペア / PWM (CCP) モジュール	189
18.0	拡張キャプチャ / コンペア / PWM (ECCP) モジュール	197
19.0	Ethernet モジュール	217
20.0	マスタ同期シリアルポート (MSSP) モジュール	269
21.0	EUSART (Enhanced Universal Synchronous Asynchronous Receiver Transmitter)	315
22.0	10 ビット A/D コンバータ (ADC) モジュール	339
23.0	コンパレータ モジュール	349
24.0	コンパレータ参照電圧モジュール	355
25.0	CPU の特殊機能	359
26.0	命令セットの概要	375
27.0	開発サポート	425
28.0	電気的特性	429
29.0	パッケージ情報	465
	補遺 A: 改訂履歴	475
	補遺 B: デバイス間の違い	476
	索引	477
	マイクロチップ社ウェブサイト	493
	お客様向け変更通知サービス	493
	カスタマサポート	493
	お客様アンケート	494
	製品識別システム	495



## 大切なお客様へ

マイクロチップ社は、弊社製品を存分にご活用頂くために、文書の作成に最善の努力を尽くしています。このため、弊社はおお客様のニーズにより的確にお応えできるように、継続的に文書の改善に努めて参ります。弊社は新刊、更新版をリリースする際に内容の見直しと充実を図って参ります。

本書に関してご質問またはご意見がございましたら、マーケティング コミュニケーション部宛てにメールまたは FAX でご連絡ください。メールの宛先は [docerrors@microchip.com](mailto:docerrors@microchip.com)、FAX 番号は (480) 792-4150 です。FAX の場合、本書の巻末にある「お客様アンケート」フォームをご利用ください。皆様からのご意見をお待ちしております。

## 最新のデータシート

本データシートの最新版を入手するには、下記のウェブサイト登録手続きを行ってください。

<http://www.microchip.com>

データシートのリビジョンは、各ページの欄外下隅に記載されている文書番号で確認できます。文書番号の末尾文字がバージョン番号を表します (例: DS30000A\_JP であれば文書 DS30000\_JP のバージョン A)。

## エラッタ

現行デバイスの動作とデータシート上の動作との間に微細な差違が生じた場合、その相違点と推奨対応策を記載したエラッタシートを発行する場合があります。弊社は、デバイスや文書に関する問題を認識した時点でエラッタを発行します。エラッタには、該当するシリコンと文書のリビジョンを明記しています。

ご使用のデバイス向けにエラッタシートが存在するかどうかは、以下の方法で確認できます。

- マイクロチップ社のウェブサイト: <http://www.microchip.com>
- 最寄のマイクロチップ社営業所 (本書の最終ページ参照)

お問い合わせの際は、ご使用のデバイス、シリコンとデータシートのリビジョン (文書番号含む) をお知らせください。

## お客様向け変更通知システム

弊社ウェブサイト ([www.microchip.com](http://www.microchip.com)) で登録頂くと、弊社全製品に関する最新情報をお受け取り頂けます。

# PIC18F97J60 ファミリ

---

NOTES:

## 1.0 デバイス概要

本データシートには、下記の各デバイスに関する情報を記載しています。

- PIC18F66J60
- PIC18F66J65
- PIC18F67J60
- PIC18F86J60
- PIC18F86J65
- PIC18F87J60
- PIC18F96J60
- PIC18F96J65
- PIC18F97J60

このファミリは、高い演算性能と豊富な機能をきわめて低価格で実現したPIC18マイクロコントローラの伝統的な利点を継承しつつ、新たに投入された低電圧デバイスファミリです。このような特長を持つPIC18F97J60ファミリは、高性能と低コストの両立が強く求められる多くのアプリケーションに最適です。

## 1.1 コアの特長

### 1.1.1 オシレータの選択肢と機能

PIC18F97J60ファミリでは以下の5つのオシレータが選択でき、アプリケーションハードウェアを開発する際に幅広い選択が可能です。

- 水晶またはセラミック振動子用の2種類の水晶振動子モード
- 4分周クロック出力を備えた2種類の外部クロックモード
- PLL (Phase Lock Loop) 周波数マルチプライヤ：外部オシレータモードでは、PLLを使って最大41.667 MHzのクロック速度を実現できます。
- 31 kHz 出力固定の内部 RC オシレータ：タイミング精度が要求されないアプリケーションで超低消費電力を実現できます。

内部オシレータブロックが安定した参照クロック源を提供し、このファミリの動作信頼性をさらに高める以下の機能を実現します。

- **フェイルセーフクロックモニタ**：この機能では、メインクロック源を、内部オシレータによる参照信号に対して常時監視します。クロック障害が発生するとコントローラが内部オシレータに切り換わり、低速クロックで動作を継続するかアプリケーションを安全にシャットダウンする事ができます。
- **2段階起動**：この機能では、パワーオンリセットまたはスリープからの復帰時に、プライマリクロック源が使用可能になるまで内部オシレータをクロック源として使います。

### 1.1.2 メモリ空間の拡張

PIC18F97J60ファミリは、アプリケーションコード格納用に64～128KBの広大なプログラムメモリを備えています。プログラムメモリに使っているフラッシュセルは、最低でも100サイクルの消去/書き込み耐性があります。リフレッシュなしのデータ保持期間は控えめに見積もっても20年以上です。

また、PIC18F97J60ファミリはアプリケーションデータの格納用として3808バイトの大容量データRAMも備えています。

### 1.1.3 外部メモリバス

128KBのメモリが不足した場合のために、100ピンのPIC18F97J60ファミリは外部メモリバス(EMB)を実装しています。EMBを使うと、コントローラの内部プログラムカウンタは最大2MBのメモリ空間をアドレス指定でき、8ビットデバイスとしては類を見ないほど多くのデータにアクセスできます。EMBを利用すると、以下のように多彩なメモリ使用法が可能です。

- 内蔵メモリと最大2MBの外部メモリを組み合わせて使う
- 外部フラッシュメモリを書き換え可能なアプリケーションコードまたは大容量のデータテーブル用として使う
- 外部RAMデバイスを大容量のデータ格納用として使う

### 1.1.4 拡張命令セット

PIC18F97J60ファミリでは、8つの新命令とインデックス付きアドレッシングモードを追加した拡張PIC18命令セットを使えます。これらの拡張命令は、もともとC言語等の高級言語向けに開発されたりエンタラントアプリケーションコードを最適化するために設計されており、デバイスコンフィグレーションとして有効にできます。

### 1.1.5 容易なデバイス移行

メモリサイズに関係なく、全てのデバイスが共通の充実した周辺モジュール機能セットを実装するため、アプリケーションの高性能化/高機能化に合わせて円滑な移行が可能です。

# PIC18F97J60 ファミリ

---

## 1.2 その他の特殊機能

- **通信**: PIC18F97J60 ファミリは、最大 2 つの独立した拡張 USART、SPI と I<sup>2</sup>C™ (マスタおよびスレーブ) の両方の動作モードをサポートする最大 2 つのマスタ SSP モジュール等、幅広いシリアル通信機能を備えています。また、汎用 I/O ポートの 1 つを 8 ビットパラレルスレーブポートとして設定すると、プロセッサ間の直接通信が可能です。
- **CCP モジュール**: 本ファミリの全デバイスは 2 つのキャプチャ/コンペア/PWM (CCP) モジュールと 3 つの拡張 CCP (ECCP) モジュールを内蔵しており、制御アプリケーションに最大限の柔軟性を提供します。最大 4 つの異なるタイムベースを使って、数種類の処理を同時に実行できます。3 つの ECCP モジュールはそれぞれ最大 4 つの PWM 出力を持ち、合計 12 の PWM をサポートできます。ECCP モジュールはこの他にも極性選択、プログラマブルデッドタイム、自動シャットダウン / リスタート、ハーフブリッジ/フルブリッジ出力モード等、多くの便利な機能を備えています。
- **10 ビット A/D コンバータ**: このモジュールにはアキュレーション時間をプログラムできる機能があり、サンプリング期間の終了を待たずにチャンネルを選択して変換を開始できるため、コードのオーバーヘッドが低減されます。
- **拡張ウォッチドッグ タイマ (WDT)**: 拡張 WDT には 16 ビットのプリスケアラがあり、タイムアウトレンジを広くとる事ができます。タイムアウト時間の詳細は [セクション 28.0「電気的特性」](#) を参照してください。

## 1.3 ファミリ各製品の詳細

PIC18F97J60 ファミリのデバイスは、64 ピン、80 ピン、100 ピンのパッケージで提供しています。[図 1-1](#)、[図 1-2](#)、[図 1-3](#) にこれら 3 つのグループのブロック図を示します。

デバイスによって下記の 4 項目が異なります。

1. フラッシュプログラムメモリ (64 KB の PIC18FX6J60 から 128 KB の PIC18FX7J60 まで 3 種類の容量をラインナップ)
2. A/D チャンネル数 (64 ピンデバイスは 11、80 ピンデバイスは 15、100 ピンデバイスは 16)
3. シリアル通信モジュール (64 ピンデバイスは EUSART モジュールと MSSP モジュール各 1、80 ピンデバイスは EUSART モジュール 2 と MSSP モジュール 1、100 ピンデバイスは EUSART モジュールと MSSP モジュール各 2)
4. I/O ピンの数 (64 ピンデバイスは 39、80 ピンデバイスは 55、100 ピンデバイスは 70)

その他の機能はファミリ内の全デバイスで共通です。[表 1-1](#)、[表 1-2](#)、[表 1-3](#) に、各デバイスの機能の一覧をまとめます。

[表 1-4](#)、[表 1-5](#)、[表 1-6](#) に、各デバイスのピン配置を示します。

# PIC18F97J60 ファミリ

表 1-1: PIC18F97J60 ファミリ (64 ピンデバイス) の機能

特長	PIC18F66J60	PIC18F66J65	PIC18F67J60
動作周波数	DC ~ 41.667 MHz	DC ~ 41.667 MHz	DC ~ 41.667 MHz
プログラムメモリ (バイト)	64K	96K	128K
プログラムメモリ (命令)	32764	49148	65532
データメモリ (バイト)	3808		
割り込み要因	26		
I/O ポート	ポート A、B、C、D、E、F、G		
I/O ピン数	39		
タイマ	5		
キャプチャ/コンペア/PWM モジュール	2		
拡張キャプチャ/コンペア/PWM モジュール	3		
シリアル通信	MSSP (1)、拡張 USART (1)		
Ethernet 通信 (10Base-T)	あり		
パラレルスレーブポート通信 (PSP)	なし		
外部メモリバス	なし		
10 ビット A/D コンバータ モジュール	11 入力チャンネル		
リセット (および遅延)	POR、BOR、RESET 命令、スタックフル、スタック アンダーフロー、MCLR、WDT (PWRT、OST)		
命令セット	75 命令、拡張命令セットを有効にした場合は 83 命令		
パッケージ	64 ピン TQFP		

表 1-2: PIC18F97J60 ファミリ (80 ピンデバイス) の機能

特長	PIC18F86J60	PIC18F86J65	PIC18F87J60
動作周波数	DC ~ 41.667 MHz	DC ~ 41.667 MHz	DC ~ 41.667 MHz
プログラムメモリ (バイト)	64K	96K	128K
プログラムメモリ (命令)	32764	49148	65532
データメモリ (バイト)	3808		
割り込み要因	27		
I/O ポート	ポート A、B、C、D、E、F、G、H、J		
I/O ピン数	55		
タイマ	5		
キャプチャ/コンペア/PWM モジュール	2		
拡張キャプチャ/コンペア/PWM モジュール	3		
シリアル通信	MSSP (1)、拡張 USART (2)		
Ethernet 通信 (10Base-T)	あり		
パラレルスレーブポート通信 (PSP)	なし		
外部メモリバス	なし		
10 ビット A/D コンバータ モジュール	15 入力チャンネル		
リセット (および遅延)	POR、BOR、RESET 命令、スタックフル、スタック アンダーフロー、MCLR、WDT (PWRT、OST)		
命令セット	75 命令、拡張命令セットを有効にした場合は 83 命令		
パッケージ	80 ピン TQFP		

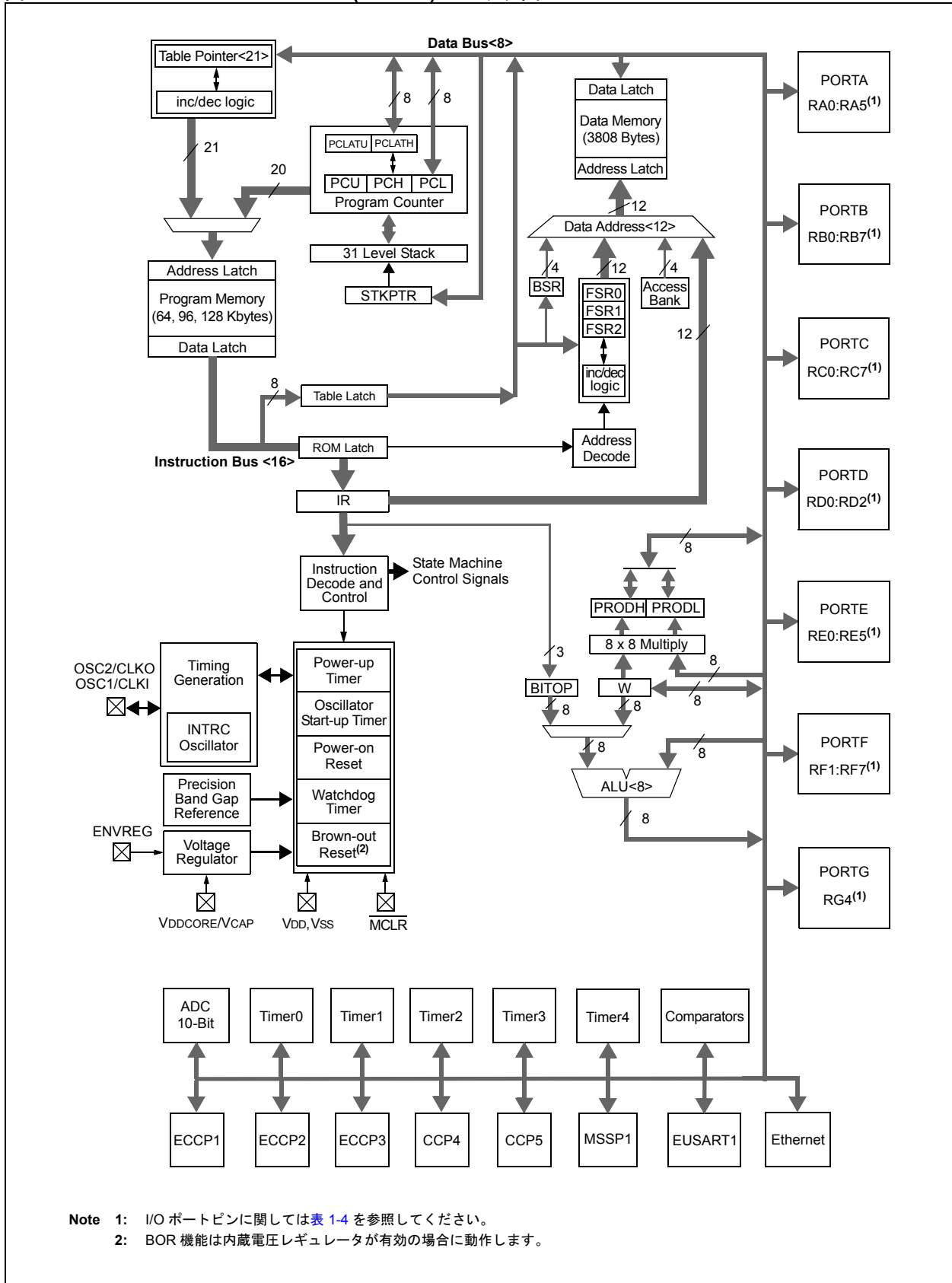
# PIC18F97J60 ファミリ

表 1-3: PIC18F97J60 ファミリ (100 ピンデバイス) の機能

特長	PIC18F96J60	PIC18F96J65	PIC18F97J60
動作周波数	DC ~ 41.667 MHz	DC ~ 41.667 MHz	DC ~ 41.667 MHz
プログラムメモリ (バイト)	64K	96K	128K
プログラムメモリ (命令)	32764	49148	65532
データメモリ (バイト)	3808		
割り込み要因	29		
I/O ポート	ポート A、B、C、D、E、F、G、H、J		
I/O ピン数	70		
タイマ	5		
キャプチャ/コンペア/PWM モジュール	2		
拡張キャプチャ/コンペア/PWM モジュール	3		
シリアル通信	MSSP (2)、拡張 USART (2)		
Ethernet 通信 (10Base-T)	あり		
パラレルスレーブポート通信 (PSP)	あり		
外部メモリバス	あり		
10 ビット A/D コンバータ モジュール	16 入力チャンネル		
リセット (および遅延)	POR、BOR、RESET 命令、スタックフル、スタックアンダーフロー、MCLR、WDT (PWRT、OST)		
命令セット	75 命令、拡張命令セットを有効にした場合は 83 命令		
パッケージ	100 ピン TQFP		

# PIC18F97J60 ファミリ

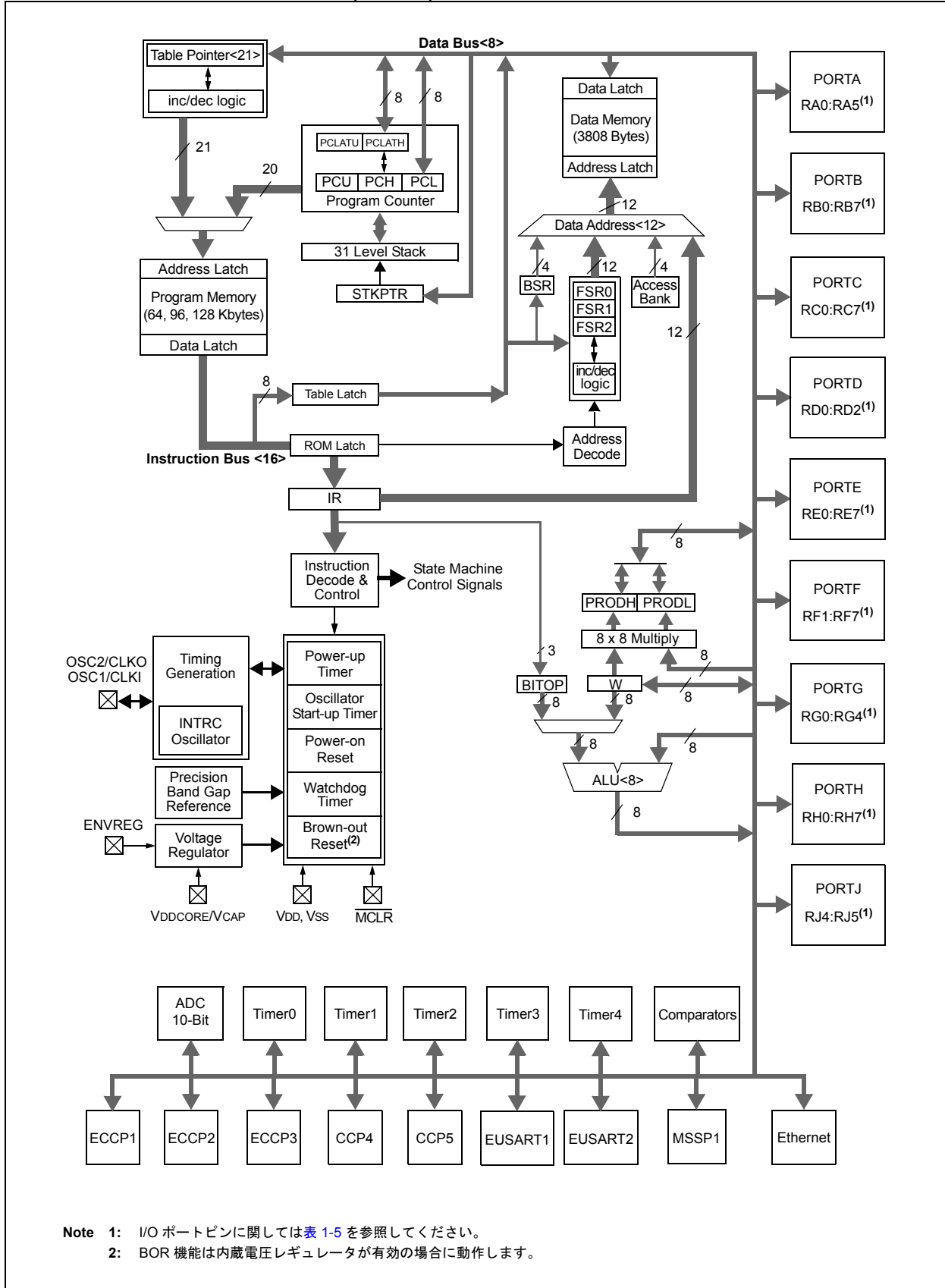
図 1-1: PIC18F66J60/66J65/67J60 (64 ピン) ブロック図



Note 1: I/O ポートピンに関しては表 1-4 を参照してください。  
 Note 2: BOR 機能は内蔵電圧レギュレータが有効の場合に動作します。

# PIC18F97J60 ファミリ

図 1-2: PIC18F86J60/86J65/87J60 (80 ピン) ブロック図

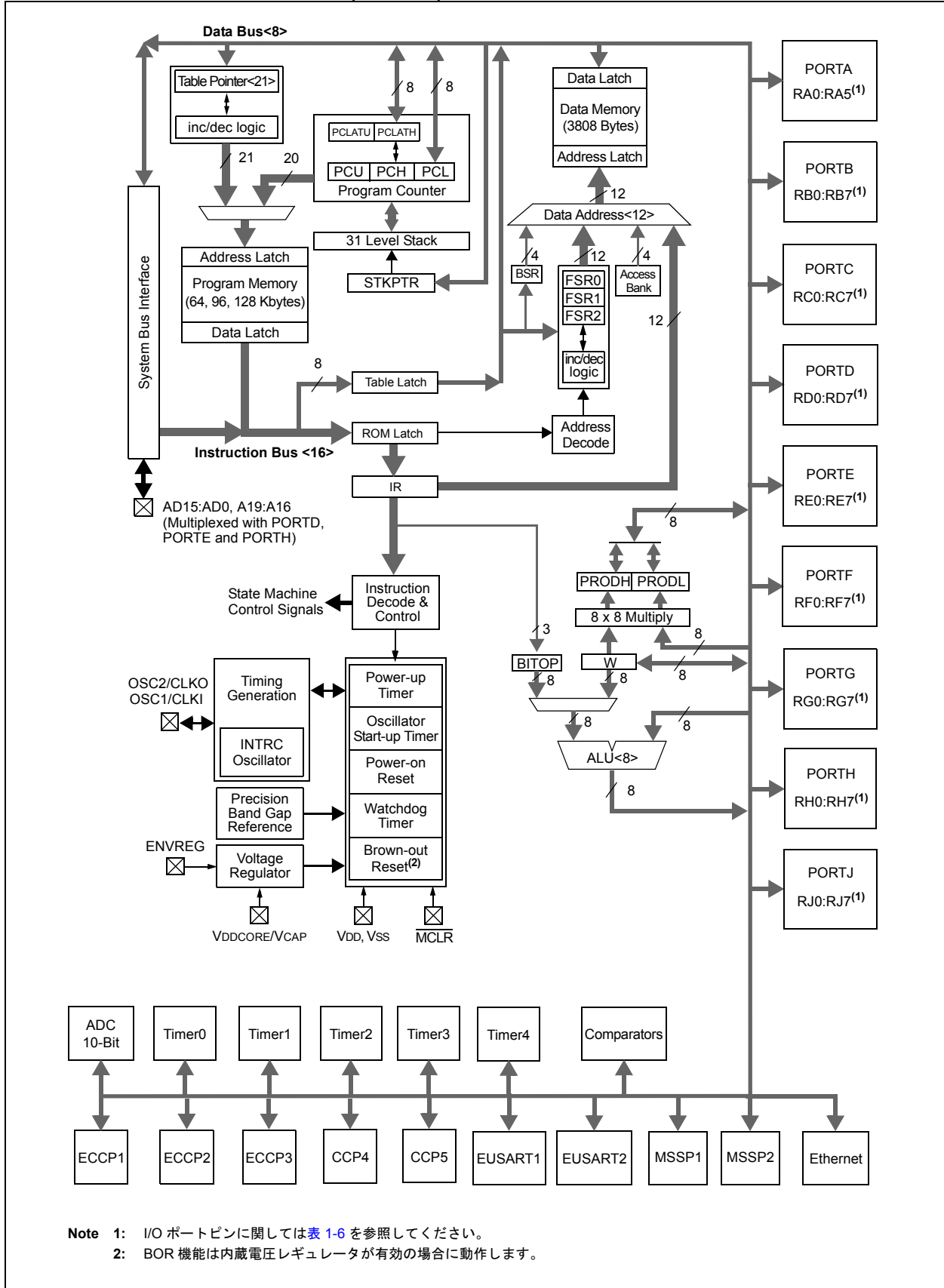


**Note 1:** I/O ポートピンに関しては表 1-5 を参照してください。  
**Note 2:** BOR 機能は内蔵電圧レギュレータが有効の場合に動作します。



# PIC18F97J60 ファミリ

図 1-3: PIC18F96J60/96J65/97J60 (100 ピン) ブロック図



# PIC18F97J60 ファミリ

表 1-4: PIC18F66J60/66J65/67J60 のピン配置と入出力の説明

ピン名	ピン番号	ピン タイプ	バッファ タイプ	説明
	TQFP			
MCLR	7	I	ST	マスタクリア (リセット) 入力。このピンはアクティブ Low でデバイスをリセット
OSC1/CLKI OSC1	39	I	ST	オシレータ水晶振動子または外部クロック入力 オシレータ水晶振動子入力または外部クロック源入力。 内部 RC モードに設定されている場合は ST バッファ、その他の場合は CMOS 外部クロック源入力。常に OSC1 ピン機能と関連付けられている (関連する OSC2/CLKO ピン参照)
CLKI		I	CMOS	
OSC2/CLKO OSC2	40	O	-	オシレータ水晶振動子またはクロック出力 オシレータ水晶振動子出力。水晶振動子または振動子に接続 (水晶振動子モードの場合) 内部 RC モードの場合、OSC2 ピンは CLKO を出力する。 その周波数は OSC1 の 1/4 で、これが命令サイクルの速度を示す
CLKO		O	-	
RA0/LEDA/AN0 RA0 LEDA AN0	24	I/O	TTL	PORTA は双方向 I/O ポート  デジタル I/O Ethernet LEDA インジケータ出力 アナログ入力 0
		O	-	
RA1/LEDB/AN1 RA1 LEDB AN1	23	I/O	TTL	デジタル I/O Ethernet LEDB インジケータ出力 アナログ入力 1
		O	-	
RA2/AN2/VREF- RA2 AN2 VREF-	22	I/O	TTL	デジタル I/O アナログ入力 2 A/D 参照電圧 (Low) 入力
		I	アナログ	
RA3/AN3/VREF+ RA3 AN3 VREF+	21	I/O	TTL	デジタル I/O アナログ入力 3 A/D 参照電圧 (High) 入力
		I	アナログ	
RA4/T0CKI RA4 T0CKI	28	I/O	ST	デジタル I/O Timer0 外部クロック入力
		I	ST	
RA5/AN4 RA5 AN4	27	I/O	TTL	デジタル I/O アナログ入力 4
		I	アナログ	

凡例: TTL = TTL 互換入力      CMOS = CMOS 互換入力または出力  
 ST = CMOS レベルのシュミットトリガ入力      アナログ = アナログ入力  
 I = 入力      O = 出力  
 P = 電源      OD = オープンドレイン (V<sub>DD</sub> への P ダイオードなし)

# PIC18F97J60 ファミリ

表 1-4: PIC18F66J60/66J65/67J60 のピン配置と入出力の説明 ( 続き )

ピン名	ピン番号	ピン タイプ	バッファ タイプ	説明
	TQFP			
RB0/INT0/FLT0 RB0 INT0 FLT0	3	I/O I I	TTL ST ST	PORTB は双方向 I/O ポート。PORTB は全ての入力に対し内部弱プルアップ有無をソフトウェアで設定可能  デジタル I/O 外部割り込み 0 拡張 PWM フォルト入力 (ECCP モジュール)。ソフトウェアで有効化
RB1/INT1 RB1 INT1	4	I/O I	TTL ST	デジタル I/O 外部割り込み 1
RB2/INT2 RB2 INT2	5	I/O I	TTL ST	デジタル I/O 外部割り込み 2
RB3/INT3 RB3 INT3	6	I/O I	TTL ST	デジタル I/O 外部割り込み 3
RB4/KBI0 RB4 KBI0	44	I/O I	TTL TTL	デジタル I/O 状態変化割り込みピン
RB5/KBI1 RB5 KBI1	43	I/O I	TTL TTL	デジタル I/O 状態変化割り込みピン
RB6/KBI2/PGC RB6 KBI2 PGC	42	I/O I I/O	TTL TTL ST	デジタル I/O 状態変化割り込みピン インサーキット デバッガ /ICSP™ プログラミング用クロックピン
RB7/KBI3/PGD RB7 KBI3 PGD	37	I/O I I/O	TTL TTL ST	デジタル I/O 状態変化割り込みピン インサーキット デバッガ /ICSP プログラミング用データピン

凡例: TTL = TTL 互換入力      CMOS = CMOS 互換入力または出力  
 ST = CMOS レベルのシュミットトリガ入力      アナログ = アナログ入力  
 I = 入力      O = 出力  
 P = 電源      OD = オープンドレイン (V<sub>DD</sub> への P ダイオードなし)

# PIC18F97J60 ファミリ

表 1-4: PIC18F66J60/66J65/67J60 のピン配置と入出力の説明 ( 続き )

ピン名	ピン番号	ピン タイプ	バッファ タイプ	説明
	TQFP			
RC0/T1OSO/T13CKI	30			PORTC は双方向 I/O ポート
RC0		I/O	ST	デジタル I/O
T1OSO		O	-	Timer1 オシレータ出力
T13CKI		I	ST	Timer1/Timer3 外部クロック入力
RC1/T1OSI/ECCP2/P2A	29			
RC1		I/O	ST	デジタル I/O
T1OSI		I	CMOS	Timer1 オシレータ入力
ECCP2		I/O	ST	キャプチャ 2 入力 / コンペア 2 出力 / PWM2 出力
P2A		O	-	ECCP2 PWM 出力 A
RC2/ECCP1/P1A	33			
RC2		I/O	ST	デジタル I/O
ECCP1		I/O	ST	キャプチャ 1 入力 / コンペア 1 出力 / PWM1 出力
P1A		O	-	ECCP1 PWM 出力 A
RC3/SCK1/SCL1	34			
RC3		I/O	ST	デジタル I/O
SCK1		I/O	ST	SPI モードの同期シリアルクロック入出力
SCL1		I/O	ST	I <sup>2</sup> C™ モードの同期シリアルクロック入出力
RC4/SDI1/SDA1	35			
RC4		I/O	ST	デジタル I/O
SDI1		I	ST	SPI データ入力
SDA1		I/O	ST	I <sup>2</sup> C データ入出力
RC5/SDO1	36			
RC5		I/O	ST	デジタル I/O
SDO1		O	-	SPI データ出力
RC6/TX1/CK1	31			
RC6		I/O	ST	デジタル I/O
TX1		O	-	EUSART1 非同期送信
CK1		I/O	ST	EUSART1 同期クロック ( 関連する RX1/DT1 ピン参照 )
RC7/RX1/DT1	32			
RC7		I/O	ST	デジタル I/O
RX1		I	ST	EUSART1 非同期受信
DT1		I/O	ST	EUSART1 同期データ ( 関連する TX1/CK1 ピン参照 )

**凡例:** TTL = TTL 互換入力                      CMOS                      = CMOS 互換入力または出力  
 ST = CMOS レベルのシュミットトリガ入力    アナログ= アナログ入力  
 I    = 入力    O                                      = 出力  
 P    = 電源    OD                                    = オープンドレイン (V<sub>DD</sub> への P ダイオードなし)

# PIC18F97J60 ファミリ

表 1-4: PIC18F66J60/66J65/67J60 のピン配置と入出力の説明 ( 続き )

ピン名	ピン番号	ピン タイプ	バッファ タイプ	説明
	TQFP			
RD0/P1B RD0 P1B	60	I/O O	ST -	PORTD は双方向 I/O ポート デジタル I/O ECCP1 PWM 出力 B
RD1/ECCP3/P3A RD1 ECCP3 P3A	59	I/O I/O O	ST ST -	デジタル I/O キャプチャ 3 入力 / コンペア 3 出力 / PWM3 出力 ECCP3 PWM 出力 A
RD2/CCP4/P3D RD2 CCP4 P3D	58	I/O I/O O	ST ST -	デジタル I/O キャプチャ 4 入力 / コンペア 4 出力 / PWM4 出力 CCP4 PWM 出力 D

凡例: TTL = TTL 互換入力      CMOS = CMOS 互換入力または出力  
 ST = CMOS レベルのシュミットトリガ入力 アナログ = アナログ入力  
 I = 入力      O = 出力  
 P = 電源      OD = オープンドレイン (VDD への P ダイオードなし)

# PIC18F97J60 ファミリ

表 1-4: PIC18F66J60/66J65/67J60 のピン配置と入出力の説明 ( 続き )

ピン名	ピン番号	ピン タイプ	バッファ タイプ	説明
	TQFP			
RE0/P2D RE0 P2D	2	I/O O	ST -	PORTE は双方向 I/O ポート デジタル I/O ECCP2 PWM 出力 D
RE1/P2C RE1 P2C	1	I/O O	ST -	デジタル I/O ECCP2 PWM 出力 C
RE2/P2B RE2 P2B	64	I/O O	ST -	デジタル I/O ECCP2 PWM 出力 B
RE3/P3C RE3 P3C	63	I/O O	ST -	デジタル I/O ECCP3 PWM 出力 C
RE4/P3B RE4 P3B	62	I/O O	ST -	デジタル I/O ECCP3 PWM 出力 B
RE5/P1C RE5 P1C	61	I/O O	ST -	デジタル I/O ECCP1 PWM 出力 C

**凡例:** TTL = TTL 互換入力      CMOS = CMOS 互換入力または出力  
 ST = CMOS レベルのシュミットトリガ入力      アナログ = アナログ入力  
 I = 入力      O = 出力  
 P = 電源      OD = オープンドレイン (V<sub>DD</sub> への P ダイオードなし)

# PIC18F97J60 ファミリ

表 1-4: PIC18F66J60/66J65/67J60 のピン配置と入出力の説明 ( 続き )

ピン名	ピン番号	ピン タイプ	バッファ タイプ	説明
	TQFP			
RF1/AN6/C2OUT	17	I/O	ST	PORTF は双方向 I/O ポート デジタル I/O アナログ入力 6 コンパレータ 2 出力
RF1		I	アナログ	
AN6		O	-	
C2OUT				
RF2/AN7/C1OUT	16	I/O	ST	デジタル I/O アナログ入力 7 コンパレータ 1 出力
RF2		I	アナログ	
AN7		O	-	
C1OUT				
RF3/AN8	15	I/O	ST	デジタル I/O アナログ入力 8
RF3		I	アナログ	
AN8				
RF4/AN9	14	I/O	ST	デジタル I/O アナログ入力 9
RF4		I	アナログ	
AN9				
RF5/AN10/CVREF	13	I/O	ST	デジタル I/O アナログ入力 10 コンパレータの参照電圧出力
RF5		I	アナログ	
AN10		O	-	
CVREF				
RF6/AN11	12	I/O	ST	デジタル I/O アナログ入力 11
RF6		I	アナログ	
AN11				
RF7/SS1	11	I/O	ST	デジタル I/O SPI スレーブ選択入力
RF7		I	TTL	
SS1				

**凡例:** TTL = TTL 互換入力      CMOS = CMOS 互換入力または出力  
 ST = CMOS レベルのシュミットトリガ入力      アナログ = アナログ入力  
 I = 入力      O = 出力  
 P = 電源      OD = オープンドレイン (V<sub>DD</sub> への P ダイオードなし)

# PIC18F97J60 ファミリ

表 1-4: PIC18F66J60/66J65/67J60 のピン配置と入出力の説明 ( 続き )

ピン名	ピン番号	ピン タイプ	バッファ タイプ	説明
	TQFP			
RG4/CCP5/P1D RG4 CCP5 P1D	8	I/O I/O O	ST ST -	PORTG は双方向 I/O ポート  デジタル I/O キャプチャ 5 入力 / コンペア 5 出力 / PWM5 出力 ECCP1 PWM 出力 D
VSS	9, 25, 41, 56	P	-	ロジックと I/O ピンのグランド参照電圧
VDD	26, 38, 57	P	-	周辺デジタルロジックおよび I/O ピン用の正電源
AVSS	20	P	-	アナログ モジュール用のグランド参照電圧
AVDD	19	P	-	アナログ モジュール用の正電源
ENVREG	18	I	ST	内蔵電圧レギュレータのイネーブル
VDDCORE/VCAP VDDCORE  VCAP	10	P  P	-  -	コアロジック電源または外部フィルタ コンデンサ接続 マイクロコントローラ コアロジックの正電源 ( レギュレータ 無効 ) 外部フィルタ コンデンサ接続 ( レギュレータ有効 )
VSSPLL	55	P	-	Ethernet PHY PLL のグランド参照電圧
VDDPLL	54	P	-	Ethernet PHY PLL の 3.3 V 正電源
VSSTX	52	P	-	Ethernet PHY 送信サブシステムのグランド参照電圧
VDDTX	49	P	-	Ethernet PHY 送信サブシステムの 3.3 V 正電源
VSSRX	45	P	-	Ethernet PHY 受信サブシステムのグランド参照電圧
VDDRX	48	P	-	Ethernet PHY 受信サブシステムの 3.3 V 正電源
RBIAS	53	I	アナログ	Ethernet PHY のバイアス電流。抵抗を介して VSS に接続する事。 仕様値については <a href="#">セクション 19.0 「Ethernet モジュール」</a> 参照
TPOUT+	51	O	-	Ethernet 差動信号出力
TPOUT-	50	O	-	Ethernet 差動信号出力
TPIN+	47	I	アナログ	Ethernet 差動信号入力
TPIN-	46	I	アナログ	Ethernet 差動信号入力

凡例: TTL = TTL 互換入力                      CMOS                      = CMOS 互換入力または出力  
 ST = CMOS レベルのシュミットトリガ入力 アナログ = アナログ入力  
 I    = 入力                                      O                                = 出力  
 P    = 電源                                      OD                             = オープンドレイン (VDD への P ダイオードなし)



# PIC18F97J60 ファミリ

表 1-5: PIC18F86J60/86J65/87J60 のピン配置と入出力の説明

ピン名	ピン番号	ピン タイプ	バッファ タイプ	説明
	TQFP			
MCLR	9	I	ST	マスタクリア (リセット) 入力。このピンはアクティブ Low でデバイスをリセット
OSC1/CLKI OSC1	49	I	ST	オシレータ水晶振動子または外部クロック入力 オシレータ水晶振動子入力または外部クロック源入力 内部 RC モードに設定されている場合は ST バッファ、 その他の場合は CMOS 外部クロック源入力。常に OSC1 ピン機能と関連付けら れている (関連する OSC2/CLKO ピン参照)
CLKI		I	CMOS	
OSC2/CLKO OSC2	50	O	-	オシレータ水晶振動子またはクロック出力 オシレータ水晶振動子出力。水晶振動子または振動子に 接続 (水晶振動子モードの場合) 内部 RC モードの場合、OSC2 ピンは CLKO を出力する。 その周波数は OSC1 の 1/4 で、これが命令サイクルの速 度を示す
CLKO		O	-	
RA0/LEDA/AN0 RA0 LEDA AN0	30	I/O	TTL	PORTA は双方向 I/O ポート  デジタル I/O Ethernet LEDA インジケータ出力 アナログ入力 0
RA1/LEDB/AN1 RA1 LEDB AN1		I/O	TTL	
RA2/AN2/VREF- RA2 AN2 VREF-		I/O	TTL	
RA3/AN3/VREF+ RA3 AN3 VREF+		I/O	TTL	
RA4/T0CKI RA4 T0CKI	34	I/O	ST	デジタル I/O Timer0 外部クロック入力
RA5/AN4 RA5 AN4		I/O	TTL	
	33	I	アナログ	アナログ入力 4

凡例: TTL = TTL 互換入力 CMOS = CMOS 互換入力または出力  
 ST = CMOS レベルのシュミットトリガ入力 アナログ = アナログ入力  
 I = 入力 O = 出力  
 P = 電源 OD = オープンドレイン (VDD への P ダイオードなし)

- Note 1:** CCP2MX コンフィグレーション ビットをセットした場合の ECCP2/P2A の既定値のピン割り当てです。  
**2:** ECCPMX コンフィグレーション ビットをセットした場合の P1B/P1C/P3B/P3C の既定値のピン割り当てです。  
**3:** CCP2MX コンフィグレーション ビットをクリアした場合の ECCP2/P2A の代替ピン割り当てです。  
**4:** ECCPMX コンフィグレーション ビットをクリアした場合の P1B/P1C/P3B/P3C の代替ピン割り当てです。

# PIC18F97J60 ファミリ

表 1-5: PIC18F86J60/86J65/87J60 のピン配置と入出力の説明 ( 続き )

ピン名	ピン番号	ピン タイプ	バッファ タイプ	説明
	TQFP			
RB0/INT0/FLT0 RB0 INT0 FLT0	5	I/O I I	TTL ST ST	PORTB は双方向 I/O ポート。PORTB は全ての入力に対し内部弱プルアップ有無をソフトウェアで設定可能  デジタル I/O 外部割り込み 0 拡張 PWM フォルト入力 (ECCP モジュール)。ソフトウェアで有効化
RB1/INT1 RB1 INT1	6	I/O I	TTL ST	デジタル I/O 外部割り込み 1
RB2/INT2 RB2 INT2	7	I/O I	TTL ST	デジタル I/O 外部割り込み 2
RB3/INT3 RB3 INT3	8	I/O I	TTL ST	デジタル I/O 外部割り込み 3
RB4/KBI0 RB4 KBI0	54	I/O I	TTL TTL	デジタル I/O 状態変化割り込みピン
RB5/KBI1 RB5 KBI1	53	I/O I	TTL TTL	デジタル I/O 状態変化割り込みピン
RB6/KBI2/PGC RB6 KBI2 PGC	52	I/O I I/O	TTL TTL ST	デジタル I/O 状態変化割り込みピン インサーキット デバッガ /ICSP™ プログラミング用 クロックピン
RB7/KBI3/PGD RB7 KBI3 PGD	47	I/O I I/O	TTL TTL ST	デジタル I/O 状態変化割り込みピン インサーキット デバッガ /ICSP プログラミング用データ ピン

凡例: TTL = TTL 互換入力      CMOS      = CMOS 互換入力または出力  
 ST = CMOS レベルのシュミットトリガ入力      アナログ = アナログ入力  
 I = 入力      O = 出力  
 P = 電源      OD = オープンドレイン (VDD への P ダイオードなし)

- Note** 1: CCP2MX コンフィグレーション ビットをセットした場合の ECCP2/P2A の既定値のピン割り当てです。  
 2: ECCPMX コンフィグレーション ビットをセットした場合の P1B/P1C/P3B/P3C の既定値のピン割り当てです。  
 3: CCP2MX コンフィグレーション ビットをクリアした場合の ECCP2/P2A の代替ピン割り当てです。  
 4: ECCPMX コンフィグレーション ビットをクリアした場合の P1B/P1C/P3B/P3C の代替ピン割り当てです。

# PIC18F97J60 ファミリ

表 1-5: PIC18F86J60/86J65/87J60 のピン配置と入出力の説明 ( 続き )

ピン名	ピン番号	ピン タイプ	バッファ タイプ	説明
	TQFP			
RC0/T1OSO/T13CKI	36			PORTC は双方向 I/O ポート
RC0		I/O	ST	デジタル I/O
T1OSO		O	-	Timer1 オシレータ出力
T13CKI		I	ST	Timer1/Timer3 外部クロック入力
RC1/T1OSI/ECCP2/P2A	35			
RC1		I/O	ST	デジタル I/O
T1OSI		I	CMOS	Timer1 オシレータ入力
ECCP2 <sup>(1)</sup>		I/O	ST	キャプチャ 2 入力 / コンペア 2 出力 / PWM2 出力
P2A <sup>(1)</sup>		O	-	ECCP2 PWM 出力 A
RC2/ECCP1/P1A	43			
RC2		I/O	ST	デジタル I/O
ECCP1		I/O	ST	キャプチャ 1 入力 / コンペア 1 出力 / PWM1 出力
P1A		O	-	ECCP1 PWM 出力 A
RC3/SCK1/SCL1	44			
RC3		I/O	ST	デジタル I/O
SCK1		I/O	ST	SPI モードの同期シリアルクロック入出力
SCL1		I/O	ST	I <sup>2</sup> C™ モードの同期シリアルクロック入出力
RC4/SDI1/SDA1	45			
RC4		I/O	ST	デジタル I/O
SDI1		I	ST	SPI データ入力
SDA1		I/O	ST	I <sup>2</sup> C データ入出力
RC5/SDO1	46			
RC5		I/O	ST	デジタル I/O
SDO1		O	-	SPI データ出力
RC6/TX1/CK1	37			
RC6		I/O	ST	デジタル I/O
TX1		O	-	EUSART1 非同期送信
CK1		I/O	ST	EUSART1 同期クロック ( 関連する RX1/DT1 ピン参照 )
RC7/RX1/DT1	38			
RC7		I/O	ST	デジタル I/O
RX1		I	ST	EUSART1 非同期受信
DT1		I/O	ST	EUSART1 同期データ ( 関連する TX1/CK1 ピン参照 )

凡例: TTL = TTL 互換入力      CMOS      = CMOS 互換入力または出力

ST = CMOS レベルのシュミットトリガ入力      アナログ = アナログ入力

I = 入力      O = 出力

P = 電源      OD = オープンドレイン (VDD への P ダイオードなし)

**Note 1:** CCP2MX コンフィグレーション ビットをセットした場合の ECCP2/P2A の既定値のピン割り当てです。

**2:** ECCPMX コンフィグレーション ビットをセットした場合の P1B/P1C/P3B/P3C の既定値のピン割り当てです。

**3:** CCP2MX コンフィグレーション ビットをクリアした場合の ECCP2/P2A の代替ピン割り当てです。

**4:** ECCPMX コンフィグレーション ビットをクリアした場合の P1B/P1C/P3B/P3C の代替ピン割り当てです。

# PIC18F97J60 ファミリ

表 1-5: PIC18F86J60/86J65/87J60 のピン配置と入出力の説明 ( 続き )

ピン名	ピン番号	ピン タイプ	バッファ タイプ	説明
	TQFP			
RD0	72	I/O	ST	PORTD は双方向 I/O ポート デジタル I/O
RD1	69	I/O	ST	
RD2	68	I/O	ST	
RE0/P2D	4	I/O	ST	PORTE は双方向 I/O ポート デジタル I/O ECCP2 PWM 出力 D
RE0 P2D		O	-	
RE1/P2C	3	I/O	ST	デジタル I/O ECCP2 PWM 出力 C
RE1 P2C		O	-	
RE2/P2B	78	I/O	ST	デジタル I/O ECCP2 PWM 出力 B
RE2 P2B		O	-	
RE3/P3C	77	I/O	ST	デジタル I/O ECCP3 PWM 出力 C
RE3 P3C <sup>(2)</sup>		O	-	
RE4/P3B	76	I/O	ST	デジタル I/O ECCP3 PWM 出力 B
RE4 P3B <sup>(2)</sup>		O	-	
RE5/P1C	75	I/O	ST	デジタル I/O ECCP1 PWM 出力 C
RE5 P1C <sup>(2)</sup>		O	-	
RE6/P1B	74	I/O	ST	デジタル I/O ECCP1 PWM 出力 B
RE6 P1B <sup>(2)</sup>		O	-	
RE7/ECCP2/P2A	73	I/O	ST	デジタル I/O キャプチャ 2 入力 / コンペア 2 出力 / PWM2 出力 ECCP2 PWM 出力 A
RE7		I/O	ST	
ECCP2 <sup>(3)</sup> P2A <sup>(3)</sup>		O	-	

凡例: TTL = TTL 互換入力      CMOS      = CMOS 互換入力または出力  
 ST = CMOS レベルのシュミットトリガ入力 アナログ= アナログ入力  
 I = 入力      O      = 出力  
 P = 電源      OD      = オープンドレイン (VDD への P ダイオードなし)

- Note** 1: CCP2MX コンフィグレーション ビットをセットした場合の ECCP2/P2A の既定値のピン割り当てです。  
 2: ECCPMX コンフィグレーション ビットをセットした場合の P1B/P1C/P3B/P3C の既定値のピン割り当てです。  
 3: CCP2MX コンフィグレーション ビットをクリアした場合の ECCP2/P2A の代替ピン割り当てです。  
 4: ECCPMX コンフィグレーション ビットをクリアした場合の P1B/P1C/P3B/P3C の代替ピン割り当てです。

# PIC18F97J60 ファミリ

表 1-5: PIC18F86J60/86J65/87J60 のピン配置と入出力の説明 ( 続き )

ピン名	ピン番号	ピン タイプ	バッファ タイプ	説明
	TQFP			
RF1/AN6/C2OUT	23	I/O	ST	PORTF は双方向 I/O ポート
RF1		I	アナログ	デジタル I/O
AN6		O	-	アナログ入力 6
C2OUT				コンパレータ 2 出力
RF2/AN7/C1OUT	18	I/O	ST	デジタル I/O
RF2		I	アナログ	アナログ入力 7
AN7		O	-	コンパレータ 1 出力
C1OUT				
RF3/AN8	17	I/O	ST	デジタル I/O
RF3		I	アナログ	アナログ入力 8
AN8				
RF4/AN9	16	I/O	ST	デジタル I/O
RF4		I	アナログ	アナログ入力 9
AN9				
RF5/AN10/CVREF	15	I/O	ST	デジタル I/O
RF5		I	アナログ	アナログ入力 10
AN10		O	-	コンパレータの参照電圧出力
CVREF				
RF6/AN11	14	I/O	ST	デジタル I/O
RF6		I	アナログ	アナログ入力 11
AN11				
RF7/SS1	13	I/O	ST	デジタル I/O
RF7		I	TTL	SPI スレーブ選択入力
SS1				

凡例: TTL = TTL 互換入力      CMOS = CMOS 互換入力または出力  
 ST = CMOS レベルのシュミットトリガ入力      アナログ = アナログ入力  
 I = 入力      O = 出力  
 P = 電源      OD = オープンドレイン (VDD への P ダイオードなし)

- Note 1:** CCP2MX コンフィグレーション ビットをセットした場合の ECCP2/P2A の既定値のピン割り当てです。  
**Note 2:** ECCPMX コンフィグレーション ビットをセットした場合の P1B/P1C/P3B/P3C の既定値のピン割り当てです。  
**Note 3:** CCP2MX コンフィグレーション ビットをクリアした場合の ECCP2/P2A の代替ピン割り当てです。  
**Note 4:** ECCPMX コンフィグレーション ビットをクリアした場合の P1B/P1C/P3B/P3C の代替ピン割り当てです。

# PIC18F97J60 ファミリ

表 1-5: PIC18F86J60/86J65/87J60 のピン配置と入出力の説明 ( 続き )

ピン名	ピン番号	ピン タイプ	バッファ タイプ	説明
	TQFP			
RG0/ECCP3/P3A RG0 ECCP3 P3A	56	I/O I/O O	ST ST -	PORTG は双方向 I/O ポート  デジタル I/O キャプチャ 3 入力 / コンペア 3 出力 / PWM3 出力 ECCP3 PWM 出力 A
RG1/TX2/CK2 RG1 TX2 CK2	55	I/O O I/O	ST - ST	デジタル I/O EUSART2 非同期送信 EUSART2 同期クロック ( 関連する RX2/DT2 ピン参照 )
RG2/RX2/DT2 RG2 RX2 DT2	42	I/O I I/O	ST ST ST	デジタル I/O EUSART2 非同期受信 EUSART2 同期データ ( 関連する TX2/CK2 ピン参照 )
RG3/CCP4/P3D RG3 CCP4 P3D	41	I/O I/O O	ST ST -	デジタル I/O キャプチャ 4 入力 / コンペア 4 出力 / PWM4 出力 ECCP3 PWM 出力 D
RG4/CCP5/P1D RG4 CCP5 P1D	10	I/O I/O O	ST ST -	デジタル I/O キャプチャ 5 入力 / コンペア 5 出力 / PWM5 出力 ECCP1 PWM 出力 D

凡例: TTL = TTL 互換入力                      CMOS                      = CMOS 互換入力または出力  
 ST = CMOS レベルのシュミットトリガ入力 アナログ= アナログ入力  
 I = 入力    O    = 出力  
 P = 電源    OD    = オープンドレイン (VDD への P ダイオードなし)

- Note 1:** CCP2MX コンフィグレーション ビットをセットした場合の ECCP2/P2A の既定値のピン割り当てです。  
**2:** ECCPMX コンフィグレーション ビットをセットした場合の P1B/P1C/P3B/P3C の既定値のピン割り当てです。  
**3:** CCP2MX コンフィグレーション ビットをクリアした場合の ECCP2/P2A の代替ピン割り当てです。  
**4:** ECCPMX コンフィグレーション ビットをクリアした場合の P1B/P1C/P3B/P3C の代替ピン割り当てです。

# PIC18F97J60 ファミリ

表 1-5: PIC18F86J60/86J65/87J60 のピン配置と入出力の説明 ( 続き )

ピン名	ピン番号	ピン タイプ	バッファ タイプ	説明
	TQFP			
RH0	79	I/O	ST	PORTH は双方向 I/O ポート デジタル I/O
RH1	80	I/O	ST	デジタル I/O
RH2	1	I/O	ST	デジタル I/O
RH3	2	I/O	ST	デジタル I/O
RH4/AN12/P3C	22	I/O	ST	デジタル I/O
RH4		I	アナログ	アナログ入力 12
AN12		O	-	ECCP3 PWM 出力 C
P3C <sup>(4)</sup>				
RH5/AN13/P3B	21	I/O	ST	デジタル I/O
RH5		I	アナログ	アナログ入力 13
AN13		O	-	ECCP3 PWM 出力 B
P3B <sup>(4)</sup>				
RH6/AN14/P1C	20	I/O	ST	デジタル I/O
RH6		I	アナログ	アナログ入力 14
AN14		O	-	ECCP1 PWM 出力 C
P1C <sup>(4)</sup>				
RH7/AN15/P1B	19	I/O	ST	デジタル I/O
RH7		I	アナログ	アナログ入力 15
AN15		O	-	ECCP1 PWM 出力 B
P1B <sup>(4)</sup>				

凡例: TTL = TTL 互換入力      CMOS = CMOS 互換入力または出力  
 ST = CMOS レベルのシュミットトリガ入力      アナログ = アナログ入力  
 I = 入力      O = 出力  
 P = 電源      OD = オープンドレイン (VDD への P ダイオードなし)

- Note 1:** CCP2MX コンフィグレーション ビットをセットした場合の ECCP2/P2A の既定値のピン割り当てです。  
**2:** ECCPMX コンフィグレーション ビットをセットした場合の P1B/P1C/P3B/P3C の既定値のピン割り当てです。  
**3:** CCP2MX コンフィグレーション ビットをクリアした場合の ECCP2/P2A の代替ピン割り当てです。  
**4:** ECCPMX コンフィグレーション ビットをクリアした場合の P1B/P1C/P3B/P3C の代替ピン割り当てです。

# PIC18F97J60 ファミリ

表 1-5: PIC18F86J60/86J65/87J60 のピン配置と入出力の説明 ( 続き )

ピン名	ピン番号	ピン タイプ	バッファ タイプ	説明
	TQFP			
RJ4	39	I/O	ST	PORTJ は双方向 I/O ポート デジタル I/O
RJ5	40	I/O	ST	デジタル I/O
Vss	11, 31, 51, 70	P	-	ロジックと I/O ピンのグラウンド参照電圧
VDD	32, 48, 71	P	-	周辺デジタルロジックおよび I/O ピン用の正電源
AVss	26	P	-	アナログ モジュール用のグラウンド参照電圧
AVDD	25	P	-	アナログ モジュール用の正電源
ENVREG	24	I	ST	内蔵電圧レギュレータのイネーブル
VDDCORE/VCAP VDDCORE	12	P	-	コアロジック電源または外部フィルタ コンデンサ接続 マイクロコントローラ コアロジックの正電源 (レギュレータ無効) 外部フィルタ コンデンサ接続 (レギュレータ有効)
VCAP		P	-	
VSSPLL	67	P	-	Ethernet PHY PLL のグラウンド参照電圧
VDDPLL	66	P	-	Ethernet PHY PLL の 3.3 V 正電源
VSSTX	64	P	-	Ethernet PHY 送信サブシステムのグラウンド参照電圧
VDDTX	61	P	-	Ethernet PHY 送信サブシステムの 3.3 V 正電源
VSSRX	57	P	-	Ethernet PHY 受信サブシステムのグラウンド参照電圧
VDDRX	60	P	-	Ethernet PHY 受信サブシステムの 3.3 V 正電源
RBIAS	65	I	アナログ	Ethernet PHY のバイアス電流。抵抗を介して Vss に接続する事。仕様値については <a href="#">セクション 19.0「Ethernet モジュール」</a> 参照
TPOUT+	63	O	-	Ethernet 差動信号出力
TPOUT-	62	O	-	Ethernet 差動信号出力
TPIN+	59	I	アナログ	Ethernet 差動信号入力
TPIN-	58	I	アナログ	Ethernet 差動信号入力

凡例: TTL = TTL 互換入力                      CMOS                      = CMOS 互換入力または出力  
 ST = CMOS レベルのシュミットトリガ入力 アナログ= アナログ入力  
 I    = 入力                                      O                                      = 出力  
 P    = 電源                                      OD                                    = オープンドレイン (VDD への P ダイオードなし)

- Note** 1: CCP2MX コンフィグレーション ビットをセットした場合の ECCP2/P2A の既定値のピン割り当てです。  
 2: ECCPMX コンフィグレーション ビットをセットした場合の P1B/P1C/P3B/P3C の既定値のピン割り当てです。  
 3: CCP2MX コンフィグレーション ビットをクリアした場合の ECCP2/P2A の代替ピン割り当てです。  
 4: ECCPMX コンフィグレーション ビットをクリアした場合の P1B/P1C/P3B/P3C の代替ピン割り当てです。



# PIC18F97J60 ファミリ

表 1-6: PIC18F96J60/96J65/97J60 のピン配置と入出力の説明

ピン名	ピン番号	ピン タイプ	バッファ タイプ	説明
	TQFP			
MCLR	13	I	ST	マスタクリア (リセット) 入力。このピンはアクティブ Low でデバイスをリセット
OSC1/CLKI OSC1	63	I	ST	オシレータ水晶振動子または外部クロック入力 オシレータ水晶振動子入力または外部クロック源入力 内部 RC モードに設定されている場合は ST バッファ、 その他の場合は CMOS 外部クロック源入力。常に OSC1 ピン機能と関連付けら れている (関連する OSC2/CLKO ピン参照)
CLKI		I	CMOS	
OSC2/CLKO OSC2	64	O	-	オシレータ水晶振動子またはクロック出力 オシレータ水晶振動子出力。水晶振動子または振動子に 接続 (水晶振動子モードの場合) 内部 RC モードの場合、OSC2 ピンは CLKO を出力する。 その周波数は OSC1 の 1/4 で、これが命令サイクルの速度 を示す
CLKO		O	-	
RA0/LEDA/AN0 RA0 LEDA AN0	35	I/O	TTL	PORTA は双方向 I/O ポート  デジタル I/O Ethernet LEDA インジケータ出力 アナログ入力 0
RA1/LEDB/AN1 RA1 LEDB AN1		I/O	TTL	
RA2/AN2/VREF- RA2 AN2 VREF-	33	I/O	TTL	デジタル I/O アナログ入力 2 A/D 参照電圧 (Low) 入力
RA3/AN3/VREF+ RA3 AN3 VREF+		I/O	TTL	
RA4/T0CKI RA4 T0CKI	42	I/O	ST	デジタル I/O Timer0 外部クロック入力
RA5/AN4 RA5 AN4		I/O	TTL	
RA5/AN4 RA5 AN4	41	I/O	TTL	デジタル I/O アナログ入力 4
RA5/AN4 RA5 AN4		I/O	アナログ	

凡例: TTL = TTL 互換入力 CMOS = CMOS 互換入力または出力

ST = CMOS レベルのシュミットトリガ入力 アナログ = アナログ入力

I = 入力 O = 出力

P = 電源 OD = オープンドレイン (V<sub>DD</sub> への P ダイオードなし)

- Note 1:** CCP2MX コンフィグレーション ビットをクリアした場合 (拡張マイクロコントローラ モード) の ECCP2/P2A の代替ピン割り当てです。
- 2:** 全ての動作モードにおける ECCP2/P2A の既定値のピン割り当てです (CCP2MX コンフィグレーション ビットをセットした場合)。
- 3:** P1B/P1C/P3B/P3C の既定値のピン割り当てです (ECCPMX コンフィグレーション ビットをセットした場合)。
- 4:** CCP2MX コンフィグレーション ビットをクリアした場合 (マイクロコントローラ モード) の ECCP2/P2A の代替ピン割り当てです。
- 5:** P1B/P1C/P3B/P3C の代替ピン割り当てです (ECCPMX コンフィグレーション ビットをクリアした場合)。

# PIC18F97J60 ファミリ

表 1-6: PIC18F96J60/96J65/97J60 のピン配置と入出力の説明 ( 続き )

ピン名	ピン番号	ピン タイプ	バッファ タイプ	説明
	TQFP			
RB0/INT0/FLT0 RB0 INT0 FLT0	5	I/O I I	TTL ST ST	PORTB は双方向 I/O ポート。PORTB は全ての入力に対し内部弱プルアップ有無をソフトウェアで設定可能  デジタル I/O 外部割り込み 0 拡張 PWM フォルト入力 (ECCP モジュール)。ソフトウェアで有効化
RB1/INT1 RB1 INT1	6	I/O I	TTL ST	デジタル I/O 外部割り込み 1
RB2/INT2 RB2 INT2	7	I/O I	TTL ST	デジタル I/O 外部割り込み 2
RB3/INT3/ECCP2/P2A RB3 INT3 ECCP2 <sup>(1)</sup> P2A <sup>(1)</sup>	8	I/O I I/O O	TTL ST ST -	デジタル I/O 外部割り込み 3 キャプチャ 2 入力 / コンペア 2 出力 / PWM2 出力 ECCP2 PWM 出力 A
RB4/KBI0 RB4 KBI0	69	I/O I	TTL TTL	デジタル I/O 状態変化割り込みピン
RB5/KBI1 RB5 KBI1	68	I/O I	TTL TTL	デジタル I/O 状態変化割り込みピン
RB6/KBI2/PGC RB6 KBI2 PGC	67	I/O I I/O	TTL TTL ST	デジタル I/O 状態変化割り込みピン インサーキット デバッガ / ICSP™ プログラミング用 クロックピン
RB7/KBI3/PGD RB7 KBI3 PGD	57	I/O I I/O	TTL TTL ST	デジタル I/O 状態変化割り込みピン インサーキット デバッガ / ICSP プログラミング用データ ピン

凡例: TTL = TTL 互換入力 CMOS = CMOS 互換入力または出力

ST = CMOS レベルのシュミットトリガ入力 アナログ = アナログ入力

I = 入力 O = 出力

P = 電源 OD = オープンドレイン (VDD への P ダイオードなし)

- Note 1:** CCP2MX コンフィグレーション ビットをクリアした場合 ( 拡張マイクロコントローラ モード ) の ECCP2/P2A の代替ピン割り当てです。
- 2:** 全ての動作モードにおける ECCP2/P2A の既定値のピン割り当てです (CCP2MX コンフィグレーション ビットをセットした場合)。
- 3:** P1B/P1C/P3B/P3C の既定値のピン割り当てです (ECCPMX コンフィグレーション ビットをセットした場合)。
- 4:** CCP2MX コンフィグレーション ビットをクリアした場合 ( マイクロコントローラ モード ) の ECCP2/P2A の代替ピン割り当てです。
- 5:** P1B/P1C/P3B/P3C の代替ピン割り当てです (ECCPMX コンフィグレーション ビットをクリアした場合)。

# PIC18F97J60 ファミリ

表 1-6: PIC18F96J60/96J65/97J60 のピン配置と入出力の説明 ( 続き )

ピン名	ピン番号	ピン タイプ	バッファ タイプ	説明
	TQFP			
RC0/T1OSO/T13CKI RC0 T1OSO T13CKI	44	I/O O I	ST - ST	PORTC は双方向 I/O ポート  デジタル I/O Timer1 オシレータ出力 Timer1/Timer3 外部クロック入力
RC1/T1OSI/ECCP2/P2A RC1 T1OSI ECCP2(2) P2A(2)	43	I/O I I/O O	ST CMOS ST -	デジタル I/O Timer1 オシレータ入力 キャプチャ 2 入力 / コンペア 2 出力 / PWM2 出力 ECCP2 PWM 出力 A
RC2/ECCP1/P1A RC2 ECCP1 P1A	53	I/O I/O O	ST ST -	デジタル I/O キャプチャ 1 入力 / コンペア 1 出力 / PWM1 出力 ECCP1 PWM 出力 A
RC3/SCK1/SCL1 RC3 SCK1 SCL1	54	I/O I/O I/O	ST ST ST	デジタル I/O SPI モードの同期シリアルクロック入出力 I <sup>2</sup> C™ モードの同期シリアルクロック入出力
RC4/SDI1/SDA1 RC4 SDI1 SDA1	55	I/O I I/O	ST ST ST	デジタル I/O SPI データ入力 I <sup>2</sup> C データ入出力
RC5/SDO1 RC5 SDO1	56	I/O O	ST -	デジタル I/O SPI データ出力
RC6/TX1/CK1 RC6 TX1 CK1	45	I/O O I/O	ST - ST	デジタル I/O EUSART1 非同期送信 EUSART1 同期クロック ( 関連する RX1/DT1 ピン参照 )
RC7/RX1/DT1 RC7 RX1 DT1	46	I/O I I/O	ST ST ST	デジタル I/O EUSART1 非同期受信 EUSART1 同期データ ( 関連する TX1/CK1 ピン参照 )

凡例: TTL = TTL 互換入力                      CMOS                      = CMOS 互換入力または出力  
ST = CMOS レベルのシュミットトリガ入力    アナログ = アナログ入力  
I   = 入力                                      O                                      = 出力  
P   = 電源                                      OD                                    = オープンドレイン (V<sub>DD</sub> への P ダイオードなし)

- Note 1:** CCP2MX コンフィグレーション ビットをクリアした場合 ( 拡張マイクロコントローラ モード ) の ECCP2/P2A の代替ピン割り当てです。
- 2:** 全ての動作モードにおける ECCP2/P2A の既定値のピン割り当てです (CCP2MX コンフィグレーション ビットをセットした場合)。
- 3:** P1B/P1C/P3B/P3C の既定値のピン割り当てです (ECCPMX コンフィグレーション ビットをセットした場合)。
- 4:** CCP2MX コンフィグレーション ビットをクリアした場合 ( マイクロコントローラ モード ) の ECCP2/P2A の代替ピン割り当てです。
- 5:** P1B/P1C/P3B/P3C の代替ピン割り当てです (ECCPMX コンフィグレーション ビットをクリアした場合)。

# PIC18F97J60 ファミリ

表 1-6: PIC18F96J60/96J65/97J60 のピン配置と入出力の説明 ( 続き )

ピン名	ピン番号	ピン タイプ	バッファ タイプ	説明
	TQFP			
RD0/AD0/PSP0 RD0 AD0 PSP0	92	I/O I/O I/O	ST TTL TTL	PORTD は双方向 I/O ポート デジタル I/O 外部メモリアドレス / データ 0 パラレルスレーブ ポートのデータ
RD1/AD1/PSP1 RD1 AD1 PSP1	91	I/O I/O I/O	ST TTL TTL	デジタル I/O 外部メモリアドレス / データ 1 パラレルスレーブ ポートのデータ
RD2/AD2/PSP2 RD2 AD2 PSP2	90	I/O I/O I/O	ST TTL TTL	デジタル I/O 外部メモリアドレス / データ 2 パラレルスレーブ ポートのデータ
RD3/AD3/PSP3 RD3 AD3 PSP3	89	I/O I/O I/O	ST TTL TTL	デジタル I/O 外部メモリアドレス / データ 3 パラレルスレーブ ポートのデータ
RD4/AD4/PSP4/SDO2 RD4 AD4 PSP4 SDO2	88	I/O I/O I/O O	ST TTL TTL -	デジタル I/O 外部メモリアドレス / データ 4 パラレルスレーブ ポートのデータ SPI データ出力
RD5/AD5/PSP5/ SDI2/SDA2 RD5 AD5 PSP5 SDI2 SDA2	87	I/O I/O I/O I I/O	ST TTL TTL ST ST	デジタル I/O 外部メモリアドレス / データ 5 パラレルスレーブ ポートのデータ SPI データ入力 I <sup>2</sup> C™ データ入出力
RD6/AD6/PSP6/ SCK2/SCL2 RD6 AD6 PSP6 SCK2 SCL2	84	I/O I/O I/O I/O I/O	ST TTL TTL ST ST	デジタル I/O 外部メモリアドレス / データ 6 パラレルスレーブ ポートのデータ SPI モードの同期シリアルクロック入出力 I <sup>2</sup> C™ モードの同期シリアルクロック入出力
RD7/AD7/PSP7/ <u>SS2</u> RD7 AD7 PSP7 <u>SS2</u>	83	I/O I/O I/O I	ST TTL TTL TTL	デジタル I/O 外部メモリアドレス / データ 7 パラレルスレーブ ポートのデータ SPI スレーブ 選択入力

凡例: TTL = TTL 互換入力      CMOS = CMOS 互換入力または出力  
 ST = CMOS レベルのシュミットトリガ入力      アナログ = アナログ入力  
 I = 入力      O = 出力  
 P = 電源      OD = オープンドレイン (V<sub>DD</sub> への P ダイオードなし)

- Note 1:** CCP2MX コンフィグレーション ビットをクリアした場合 ( 拡張マイクロコントローラ モード ) の ECCP2/P2A の代替ピン割り当てです。
- 2:** 全ての動作モードにおける ECCP2/P2A の既定値のピン割り当てです ( CCP2MX コンフィグレーション ビットをセットした場合 )。
- 3:** P1B/P1C/P3B/P3C の既定値のピン割り当てです ( ECCPMX コンフィグレーション ビットをセットした場合 )。
- 4:** CCP2MX コンフィグレーション ビットをクリアした場合 ( マイクロコントローラ モード ) の ECCP2/P2A の代替ピン割り当てです。
- 5:** P1B/P1C/P3B/P3C の代替ピン割り当てです ( ECCPMX コンフィグレーション ビットをクリアした場合 )。

# PIC18F97J60 ファミリ

表 1-6: PIC18F96J60/96J65/97J60 のピン配置と入出力の説明 ( 続き )

ピン名	ピン番号	ピン タイプ	バッファ タイプ	説明
	TQFP			
RE0/AD8/ $\overline{RD}$ /P2D	4			PORTE は双方向 I/O ポート
RE0		I/O	ST	デジタル I/O
AD8		I/O	TTL	外部メモリアドレス / データ 8
$\overline{RD}$		I	TTL	パラレルスレーブ ポートの読み出し制御
P2D	O	-	ECCP2 PWM 出力 D	
RE1/AD9/ $\overline{WR}$ /P2C	3			
RE1		I/O	ST	デジタル I/O
AD9		I/O	TTL	外部メモリアドレス / データ 9
$\overline{WR}$		I	TTL	パラレルスレーブ ポートの書き込み制御
P2C	O	-	ECCP2 PWM 出力 C	
RE2/AD10/ $\overline{CS}$ /P2B	98			
RE2		I/O	ST	デジタル I/O
AD10		I/O	TTL	外部メモリアドレス / データ 10
$\overline{CS}$		I	TTL	パラレルスレーブ ポートのチップセレクト制御
P2B	O	-	ECCP2 PWM 出力 B	
RE3/AD11/P3C	97			
RE3		I/O	ST	デジタル I/O
AD11		I/O	TTL	外部メモリアドレス / データ 11
P3C <sup>(3)</sup>	O	-	ECCP3 PWM 出力 C	
RE4/AD12/P3B	96			
RE4		I/O	ST	デジタル I/O
AD12		I/O	TTL	外部メモリアドレス / データ 12
P3B <sup>(3)</sup>	O	-	ECCP3 PWM 出力 B	
RE5/AD13/P1C	95			
RE5		I/O	ST	デジタル I/O
AD13		I/O	TTL	外部メモリアドレス / データ 13
P1C <sup>(3)</sup>	O	-	ECCP1 PWM 出力 C	
RE6/AD14/P1B	94			
RE6		I/O	ST	デジタル I/O
AD14		I/O	TTL	外部メモリアドレス / データ 14
P1B <sup>(3)</sup>	O	-	ECCP1 PWM 出力 B	
RE7/AD15/ECCP2/P2A	93			
RE7		I/O	ST	デジタル I/O
AD15		I/O	TTL	外部メモリアドレス / データ 15
ECCP2 <sup>(4)</sup>		I/O	ST	キャプチャ 2 入力 / コンペア 2 出力 / PWM2 出力
P2A <sup>(4)</sup>	O	-	ECCP2 PWM 出力 A	

凡例: TTL = TTL 互換入力 CMOS = CMOS 互換入力または出力

ST = CMOS レベルのシュミットトリガ入力 アナログ = アナログ入力

I = 入力 O = 出力

P = 電源 OD = オープンドレイン (V<sub>DD</sub> への P ダイオードなし)

- Note** 1: CCP2MX コンフィグレーション ビットをクリアした場合 ( 拡張マイクロコントローラ モード ) の ECCP2/P2A の代替ピン割り当てです。
- 2: 全ての動作モードにおける ECCP2/P2A の既定値のピン割り当てです (CCP2MX コンフィグレーション ビットをセットした場合)。
- 3: P1B/P1C/P3B/P3C の既定値のピン割り当てです (ECCPMX コンフィグレーション ビットをセットした場合)。
- 4: CCP2MX コンフィグレーション ビットをクリアした場合 ( マイクロコントローラ モード ) の ECCP2/P2A の代替ピン割り当てです。
- 5: P1B/P1C/P3B/P3C の代替ピン割り当てです (ECCPMX コンフィグレーション ビットをクリアした場合)。

# PIC18F97J60 ファミリ

表 1-6: PIC18F96J60/96J65/97J60 のピン配置と入出力の説明 ( 続き )

ピン名	ピン番号	ピン タイプ	バッファ タイプ	説明
	TQFP			
RF0/AN5 RF0 AN5	12	I/O I	ST アナログ	PORTF は双方向 I/O ポート デジタル I/O アナログ入力 5
RF1/AN6/C2OUT RF1 AN6 C2OUT	28	I/O I O	ST アナログ -	デジタル I/O アナログ入力 6 コンパレータ 2 出力
RF2/AN7/C1OUT RF2 AN7 C1OUT	23	I/O I O	ST アナログ -	デジタル I/O アナログ入力 7 コンパレータ 1 出力
RF3/AN8 RF3 AN8	22	I/O I	ST アナログ	デジタル I/O アナログ入力 8
RF4/AN9 RF4 AN9	21	I/O I	ST アナログ	デジタル I/O アナログ入力 9
RF5/AN10/CVREF RF5 AN10 CVREF	20	I/O I O	ST アナログ -	デジタル I/O アナログ入力 10 コンパレータの参照電圧出力
RF6/AN11 RF6 AN11	19	I/O I	ST アナログ	デジタル I/O アナログ入力 11
RF7/SS1 RF7 SS1	18	I/O I	ST TTL	デジタル I/O SPI スレーブ選択入力

凡例: TTL = TTL 互換入力                      CMOS                      = CMOS 互換入力または出力  
 ST = CMOS レベルのシュミットトリガ入力      アナログ = アナログ入力  
 I = 入力    O = 出力  
 P = 電源    OD = オープンドレイン (V<sub>DD</sub> への P ダイオードなし)

- Note 1:** CCP2MX コンフィグレーション ビットをクリアした場合 ( 拡張マイクロコントローラ モード ) の ECCP2/P2A の代替ピン割り当てです。
- 2:** 全ての動作モードにおける ECCP2/P2A の既定値のピン割り当てです (CCP2MX コンフィグレーション ビットをセットした場合)。
- 3:** P1B/P1C/P3B/P3C の既定値のピン割り当てです (ECCPMX コンフィグレーション ビットをセットした場合)。
- 4:** CCP2MX コンフィグレーション ビットをクリアした場合 ( マイクロコントローラ モード ) の ECCP2/P2A の代替ピン割り当てです。
- 5:** P1B/P1C/P3B/P3C の代替ピン割り当てです (ECCPMX コンフィグレーション ビットをクリアした場合)。

# PIC18F97J60 ファミリ

表 1-6: PIC18F96J60/96J65/97J60 のピン配置と入出力の説明 ( 続き )

ピン名	ピン番号	ピン タイプ	バッファ タイプ	説明
	TQFP			
RG0/ECCP3/P3A RG0 ECCP3 P3A	71	I/O I/O O	ST ST -	PORTG は双方向 I/O ポート  デジタル I/O キャプチャ 3 入力 / コンペア 3 出力 / PWM3 出力 ECCP3 PWM 出力 A
RG1/TX2/CK2 RG1 TX2 CK2	70	I/O O I/O	ST - ST	デジタル I/O EUSART2 非同期送信 EUSART2 同期クロック ( 関連する RX2/DT2 ピン参照 )
RG2/RX2/DT2 RG2 RX2 DT2	52	I/O I I/O	ST ST ST	デジタル I/O EUSART2 非同期受信 EUSART2 同期データ ( 関連する TX2/CK2 ピン参照 )
RG3/CCP4/P3D RG3 CCP4 P3D	51	I/O I/O O	ST ST -	デジタル I/O キャプチャ 4 入力 / コンペア 4 出力 / PWM4 出力 ECCP3 PWM 出力 D
RG4/CCP5/P1D RG4 CCP5 P1D	14	I/O I/O O	ST ST -	デジタル I/O キャプチャ 5 入力 / コンペア 5 出力 / PWM5 出力 ECCP1 PWM 出力 D
RG5	11	I/O	ST	デジタル I/O
RG6	10	I/O	ST	デジタル I/O
RG7	38	I/O	ST	デジタル I/O

凡例: TTL = TTL 互換入力                      CMOS                      = CMOS 互換入力または出力  
 ST = CMOS レベルのシュミットトリガ入力 アナログ = アナログ入力  
 I = 入力    O    = 出力  
 P = 電源    OD    = オープンドレイン (V<sub>DD</sub> への P ダイオードなし)

- Note 1:** CCP2MX コンフィグレーション ビットをクリアした場合 ( 拡張マイクロコントローラ モード ) の ECCP2/P2A の代替ピン割り当てです。
- 2:** 全ての動作モードにおける ECCP2/P2A の既定値のピン割り当てです (CCP2MX コンフィグレーション ビットをセットした場合)。
- 3:** P1B/P1C/P3B/P3C の既定値のピン割り当てです (ECCPMX コンフィグレーション ビットをセットした場合)。
- 4:** CCP2MX コンフィグレーション ビットをクリアした場合 ( マイクロコントローラ モード ) の ECCP2/P2A の代替ピン割り当てです。
- 5:** P1B/P1C/P3B/P3C の代替ピン割り当てです (ECCPMX コンフィグレーション ビットをクリアした場合)。

# PIC18F97J60 ファミリ

表 1-6: PIC18F96J60/96J65/97J60 のピン配置と入出力の説明 ( 続き )

ピン名	ピン番号	ピン タイプ	バッファ タイプ	説明
	TQFP			
RH0/A16 RH0 A16	99	I/O O	ST -	PORTH は双方向 I/O ポート デジタル I/O 外部メモリアドレス 16
RH1/A17 RH1 A17	100	I/O O	ST -	デジタル I/O 外部メモリアドレス 17
RH2/A18 RH2 A18	1	I/O O	ST -	デジタル I/O 外部メモリアドレス 18
RH3/A19 RH3 A19	2	I/O O	ST -	デジタル I/O 外部メモリアドレス 19
RH4/AN12/P3C RH4 AN12 P3C <sup>(5)</sup>	27	I/O I O	ST アナログ -	デジタル I/O アナログ入力 12 ECCP3 PWM 出力 C
RH5/AN13/P3B RH5 AN13 P3B <sup>(5)</sup>	26	I/O I O	ST アナログ -	デジタル I/O アナログ入力 13 ECCP3 PWM 出力 B
RH6/AN14/P1C RH6 AN14 P1C <sup>(5)</sup>	25	I/O I O	ST アナログ -	デジタル I/O アナログ入力 14 ECCP1 PWM 出力 C
RH7/AN15/P1B RH7 AN15 P1B <sup>(5)</sup>	24	I/O I O	ST アナログ -	デジタル I/O アナログ入力 15 ECCP1 PWM 出力 B

凡例: TTL = TTL 互換入力      CMOS = CMOS 互換入力または出力  
 ST = CMOS レベルのシュミットトリガ入力      アナログ = アナログ入力  
 I = 入力      O = 出力  
 P = 電源      OD = オープンドレイン (V<sub>DD</sub> への P ダイオードなし)

- Note 1:** CCP2MX コンフィグレーション ビットをクリアした場合 ( 拡張マイクロコントローラ モード ) の ECCP2/P2A の代替ピン割り当てです。
- 2:** 全ての動作モードにおける ECCP2/P2A の既定値のピン割り当てです (CCP2MX コンフィグレーション ビットをセットした場合)。
- 3:** P1B/P1C/P3B/P3C の既定値のピン割り当てです (ECCPMX コンフィグレーション ビットをセットした場合)。
- 4:** CCP2MX コンフィグレーション ビットをクリアした場合 ( マイクロコントローラ モード ) の ECCP2/P2A の代替ピン割り当てです。
- 5:** P1B/P1C/P3B/P3C の代替ピン割り当てです (ECCPMX コンフィグレーション ビットをクリアした場合)。



# PIC18F97J60 ファミリ

表 1-6: PIC18F96J60/96J65/97J60 のピン配置と入出力の説明 (続き)

ピン名	ピン番号	ピン タイプ	バッファ タイプ	説明
	TQFP			
RJ0/ALE RJ0 ALE	49	I/O O	ST -	PORTJ は双方向 I/O ポート  デジタル I/O 外部メモリ アドレスラッチ イネーブル
RJ1/ $\overline{OE}$ RJ1 OE	50	I/O O	ST -	デジタル I/O 外部メモリ 出カ イネーブル
RJ2/ $\overline{WRL}$ RJ2 WRL	66	I/O O	ST -	デジタル I/O 外部メモリ 書き込み Low 制御
RJ3/ $\overline{WRH}$ RJ3 WRH	61	I/O O	ST -	デジタル I/O 外部メモリ 書き込み High 制御
RJ4/BA0 RJ4 BA0	47	I/O O	ST -	デジタル I/O 外部メモリ バイトアドレス 0 制御
RJ5/ $\overline{CE}$ RJ5 CE	48	I/O O	ST -	デジタル I/O 外部メモリ チップイネーブル制御
RJ6/ $\overline{LB}$ RJ6 LB	58	I/O O	ST -	デジタル I/O 外部メモリ Low バイト制御
RJ7/ $\overline{UB}$ RJ7 UB	39	I/O O	ST -	デジタル I/O 外部メモリ High バイト制御

凡例: TTL = TTL 互換入力                      CMOS                      = CMOS 互換入力または出力  
 ST = CMOS レベルのシュミットトリガ入力    アナログ = アナログ入力  
 I = 入力    O = 出力  
 P = 電源    OD = オープンドレイン (V<sub>DD</sub> への P ダイオードなし)

- Note 1:** CCP2MX コンフィグレーション ビットをクリアした場合 (拡張マイクロコントローラ モード) の ECCP2/P2A の代替ピン割り当てです。
- 2:** 全ての動作モードにおける ECCP2/P2A の既定値のピン割り当てです (CCP2MX コンフィグレーション ビットをセットした場合)。
- 3:** P1B/P1C/P3B/P3C の既定値のピン割り当てです (ECCPMX コンフィグレーション ビットをセットした場合)。
- 4:** CCP2MX コンフィグレーション ビットをクリアした場合 (マイクロコントローラ モード) の ECCP2/P2A の代替ピン割り当てです。
- 5:** P1B/P1C/P3B/P3C の代替ピン割り当てです (ECCPMX コンフィグレーション ビットをクリアした場合)。

# PIC18F97J60 ファミリ

表 1-6: PIC18F96J60/96J65/97J60 のピン配置と入出力の説明 ( 続き )

ピン名	ピン番号	ピン タイプ	バッファ タイプ	説明
	TQFP			
NC	9	-	-	未接続
VSS	15, 36, 40, 60, 65, 85	P	-	ロジックと I/O ピンのグラウンド参照電圧
VDD	17, 37, 59, 62, 86	P	-	周辺デジタルロジックおよび I/O ピン用の正電源
AVSS	31	P	-	アナログ モジュール用のグラウンド参照電圧
AVDD	30	P	-	アナログ モジュール用の正電源
ENVREG	29	I	ST	内蔵電圧レギュレータのイネーブル
VDDCORE/VCAP VDDCORE	16	P	-	コアロジック電源または外部フィルタ コンデンサ接続 マイクロコントローラ コアロジックの正電源 (レギュレータ無効)
VCAP		P	-	外部フィルタ コンデンサ接続 (レギュレータ有効)
VSSPLL	82	P	-	Ethernet PHY PLL のグラウンド参照電圧
VDDPLL	81	P	-	Ethernet PHY PLL の 3.3 V 正電源
VSSTX	79	P	-	Ethernet PHY 送信サブシステムのグラウンド参照電圧
VDDTX	76	P	-	Ethernet PHY 送信サブシステムの 3.3 V 正電源
VSSRX	72	P	-	Ethernet PHY 受信サブシステムのグラウンド参照電圧
VDDRX	75	P	-	Ethernet PHY 受信サブシステムの 3.3 V 正電源
RBIAS	80	I	アナログ	Ethernet PHY のバイアス電流。抵抗を介して VSS に接続する事。仕様値については <a href="#">セクション 19.0「Ethernet モジュール」</a> 参照
TPOUT+	78	O	-	Ethernet 差動信号出力
TPOUT-	77	O	-	Ethernet 差動信号出力
TPIN+	74	I	アナログ	Ethernet 差動信号入力
TPIN-	73	I	アナログ	Ethernet 差動信号入力

凡例: TTL = TTL 互換入力 CMOS = CMOS 互換入力または出力  
 ST = CMOS レベルのシュミットトリガ入力 アナログ = アナログ入力  
 I = 入力 O = 出力  
 P = 電源 OD = オープンドレイン (VDD への P ダイオードなし)

- Note 1:** CCP2MX コンフィグレーション ビットをクリアした場合 (拡張マイクロコントローラ モード) の ECCP2/P2A の代替ピン割り当てです。
- 2:** 全ての動作モードにおける ECCP2/P2A の既定値のピン割り当てです (CCP2MX コンフィグレーション ビットをセットした場合)。
- 3:** P1B/P1C/P3B/P3C の既定値のピン割り当てです (ECCPMX コンフィグレーション ビットをセットした場合)。
- 4:** CCP2MX コンフィグレーション ビットをクリアした場合 (マイクロコントローラ モード) の ECCP2/P2A の代替ピン割り当てです。
- 5:** P1B/P1C/P3B/P3C の代替ピン割り当てです (ECCPMX コンフィグレーション ビットをクリアした場合)。

## 2.0 PIC18FJ マイクロコントローラの使用にあたってのガイドライン

### 2.1 基本的な接続要件

8 ビット マイクロコントローラの PIC18F97J60 ファミリを使って開発を始める前に、デバイスピンの最小接続要件に注意が必要です。

下記のピンは常に接続する必要があります。

- 全ての VDD および VSS ピン  
([セクション 2.2 「電源ピン」](#) 参照)
- 全ての AVDD および AVSS ピン (アナログデバイス機能未使用時も必須)  
([セクション 2.2 「電源ピン」](#) 参照)
- MCLR ピン  
([セクション 2.3 「マスタクリア \(MCLR\) ピン」](#) 参照)
- ENVREG (実装している場合のみ) および VCAP/VDDCORE ピン  
([セクション 2.4 「電圧レギュレータピン \(ENVREG と VCAP/VDDCORE\)」](#) 参照)

最終アプリケーションで使う場合、以下のピンも接続する必要があります。

- PGC/PGD ピン (In-Circuit Serial Programming™ (ICSP™) とデバッグに使用)  
([セクション 2.5 「ICSP ピン」](#) 参照)
- OSCI/OSCO ピン  
(外部オシレータ信号源を使う場合)  
([セクション 2.6 「外部オシレータピン」](#) 参照)

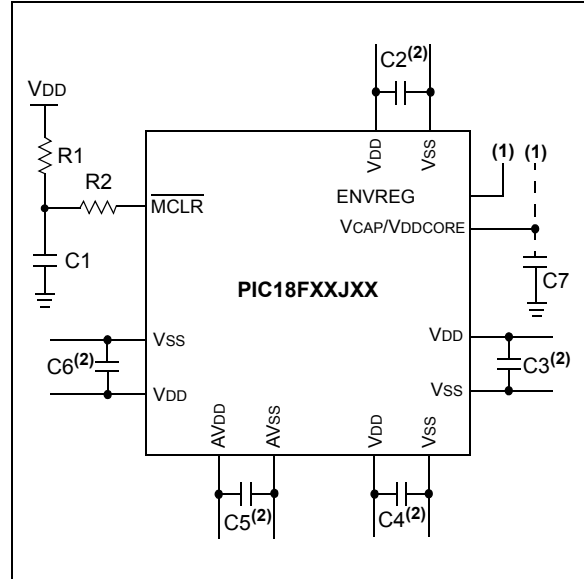
さらに、以下のピンの接続が必要な場合もあります。

- VREF+/VREF- ピン (アナログ モジュール用に外部参照電圧を使う場合に使用)

**Note:** AVDD および AVSS ピンは、アナログ モジュールを使わない場合でも、常に接続する必要があります。

図 2-1 に、最低限の必須ピン接続を示します。

図 2-1: 必須接続ピン



#### コンデンサと抵抗の推奨値:

C1 ~ C6: 0.1  $\mu$ F、20 V、セラミック

C7: 10  $\mu$ F、6.3 V 以上、タンタルまたはセラミック

R1: 10 k $\Omega$

R2: 100 ~ 470  $\Omega$

- Note 1:** ENVREG ピンの接続については[セクション 2.4 「電圧レギュレータピン \(ENVREG と VCAP/VDDCORE\)」](#)を参照してください。
- 2:** 上図は、5 組の VDD/VSS および AVDD/AVSS ピンペアを持つ PIC18F の例です。これらのピンペアの数はデバイスによって異なります。従って、必要なデカップリングコンデンサの数も異なります。

# PIC18F97J60 ファミリ

## 2.2 電源ピン

### 2.2.1 デカップリング コンデンサ

VDD/VSS、AVDD/AVSS 等、全ての電源ピンペアにはデカップリング コンデンサが必要です。

デカップリング コンデンサを使う場合、以下の基準に注意してください。

- **コンデンサの値とタイプ**: 0.1  $\mu\text{F}$  (100 nF)、10 ~ 20 V のコンデンサを推奨します。このコンデンサには、共振周波数が 200 MHz 以上の低 ESR 品を使う必要があります。セラミック コンデンサの使用を推奨します。
- **プリント基板の配置**: デカップリング コンデンサは、できるだけピンの近くに配置します。コンデンサとデバイスを基板の同一面上に配置する事を推奨します。スペースに制約がある場合、ビアを使ってコンデンサを基板の別の層に配置する事もできますが、ピンからコンデンサまでのトレース長を 0.25 インチ (6 mm) 以下にする必要があります。
- **高周波ノイズへの対応**: 基板が数十 MHz 以上の高周波ノイズの影響を受ける場合、上記のデカップリング コンデンサと並列に補助コンデンサ (セラミックタイプ) を追加します。0.01 ~ 0.001  $\mu\text{F}$  の補助コンデンサを、デカップリング コンデンサに近付けて配置します。高速回路の設計では、このようなコンデンサのペア (例: 0.1  $\mu\text{F}$  と 0.001  $\mu\text{F}$ ) を電源ピンとグランドピンのできるだけ近くに配置するよう配慮が必要です。
- **性能の最適化**: 電源回路からのボードレイアウトでは、最初にデカップリング コンデンサまでの電源の往復パターンを作ってからデバイスピンへのパターンを作ります。これにより、デカップリング コンデンサを電源システムの初段として、確実に配置します。コンデンサと電源ピン間のパターン長を最短にして PCB の配線インダクタンスを低減する事も重要です。

### 2.2.2 タンクコンデンサ

マイクロコントローラを含む IC 用の基板設計では、基板上の電源パターンの長さが 6 インチ (約 15 cm) を超える場合、タンクコンデンサを使ってローカル電源を提供する事を推奨します。タンクコンデンサの静電容量は、電源とデバイスを接続するパターンの抵抗と、そのアプリケーションでデバイスが消費する最大電流に基づいて決定する必要があります。つまり、デバイスにおける電圧低下が許容範囲に収まるようにタンクコンデンサの静電容量を選択します。タンクコンデンサの代表的な静電容量は 4.7 ~ 47  $\mu\text{F}$  です。

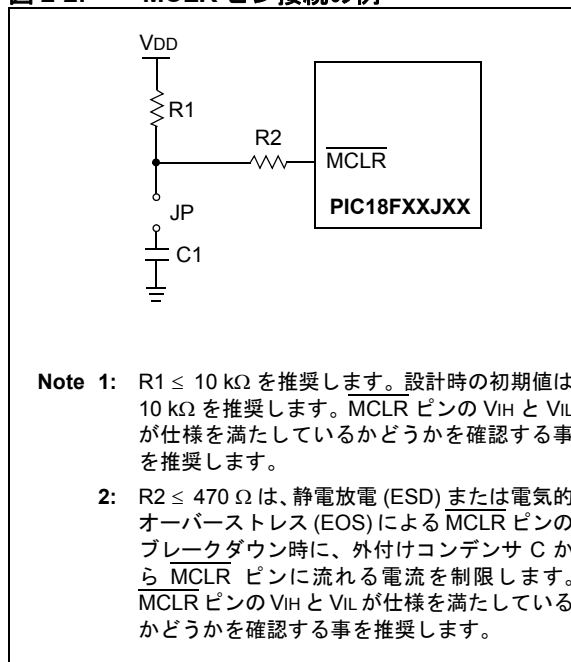
## 2.3 マスタクリア (MCLR) ピン

MCLR ピンは、2つのデバイス機能 (デバイスリセットとデバイス プログラミング/デバッグ) を提供します。最終アプリケーションでプログラミング/デバッグ機能を使わない場合、このピンは VDD に直接接続できます。ただし、このピンに別の部品を接続しておけば、瞬時電圧低下による不意のリセットを防ぐ事ができる場合があります。図 2-1 に代表的な回路を示します。アプリケーションの要件に応じて、これとは異なる回路も使えます。

プログラミング/デバッグ時には、このピンに付加される可能性のある抵抗と静電容量に注意する必要があります。デバイス プログラマ/デバッガは MCLR ピンを駆動します。この際に、電圧レベル ( $V_{IH}$ 、 $V_{IL}$ ) と高速信号挙動に対する悪影響を回避する必要があります。このため、アプリケーションと PCB の要件に基づいて R1 と C1 の値を調整する必要があります。例えば、プログラミング/デバッグ時にジャンパを使って、コンデンサ C1 を MCLR ピンから切り離す事を推奨します (図 2-2 参照)。通常動作時は、このジャンパを元に戻します。

MCLR ピンに関連する全ての部品は、ピンから 0.25 インチ (6 mm) 以内に配置する必要があります。

図 2-2: MCLR ピン接続の例



## 2.4 電圧レギュレータ ピン (ENVREG と VCAP/VDDCORE)

内蔵電圧レギュレータ イネーブルピンのENVREGは、必ず電源電圧またはグラウンドに直接接続する必要があります。ENVREG ピンをVDDに接続するとレギュレータが有効になり、グラウンドに接続するとレギュレータが無効になります。内蔵電圧レギュレータの接続および使用方法は、[セクション 25.3「内蔵電圧レギュレータ」](#)を参照してください。

レギュレータを有効にした場合、電圧レギュレータの出力電圧を安定させるために VCAP/VDDCORE ピンに低 ESR (< 5 Ω) のコンデンサが必要です。VCAP/VDDCORE ピンをVDDに直接接続しないでください。このピンは、10 μF のコンデンサを介してグラウンドに接続する必要があります。このコンデンサにはセラミックまたはタンタル型が使えます。[表 2-1](#) に、この用途に適したコンデンサの例を示します。これらと同等の仕様を持つコンデンサであれば使えます。

使用コンデンサの ESR 等価性を評価するために [図 2-3](#) が使えます。

トレース長は 0.25 インチ (6 mm) 以内とする事を推奨します。詳細は、[28.0「電気的特性」](#)を参照してください。

レギュレータを無効にした場合、VCAP/VDDCORE ピンを VDDCORE レベルの電源電圧に接続する必要があります。VDD と VDDCORE の詳細は、[28.0「電気的特性」](#)を参照してください。

なお、PIC18FJ ファミリのうち「LF」タイプの少ピンデバイス (PIC18LF45J10 等) には ENVREG ピンがありません。これらのデバイスに内蔵された電圧レギュレータは恒久的に無効にされており、VDDCORE ピンには常に電源電圧を供給する必要があります。

図 2-3: 周波数に対する推奨コンデンサの ESR 特性

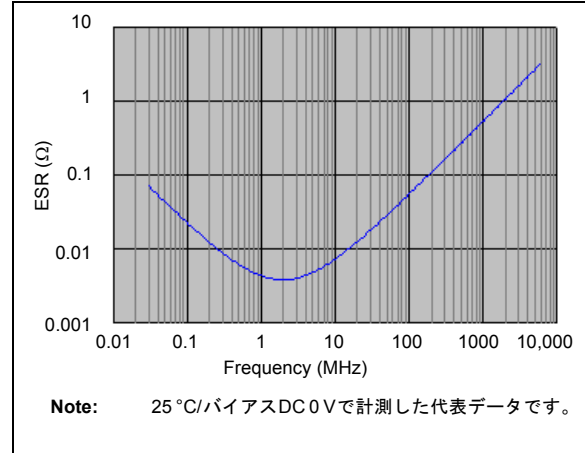


表 2-1: 推奨コンデンサの例

メーカー	製品番号	公称静電容量	基本公差	定格電圧	温度レンジ
TDK	C3216X7R1C106K	10 μF	±10%	16 V	-55 ~ 125 °C
TDK	C3216X5R1C106K	10 μF	±10%	16 V	-55 ~ 85 °C
Panasonic	ECJ-3YX1C106K	10 μF	±10%	16 V	-55 ~ 125 °C
Panasonic	ECJ-4YB1C106K	10 μF	±10%	16 V	-55 ~ 85 °C
Murata	GRM32DR71C106KA01L	10 μF	±10%	16 V	-55 ~ 125 °C
Murata	GRM31CR61C106KC31L	10 μF	±10%	16 V	-55 ~ 85 °C

# PIC18F97J60 ファミリ

## 2.4.1 セラミック コンデンサに関する注意点

近年、数十  $\mu\text{F}$  までの表面実装型大容量 / 低電圧セラミック コンデンサのコスト効率が大幅に改善されています。低 ESR でサイズが小さく、その他の特性にも優れるセラミック コンデンサは、多くのアプリケーションに適しています。

このマイクロコントローラの  $V_{\text{DDCORE}}$  電圧レギュレータにはセラミック コンデンサが適します。しかし、アプリケーションの動作レンジ全体で十分な静電容量を確保するために、コンデンサの選択には注意が必要です。

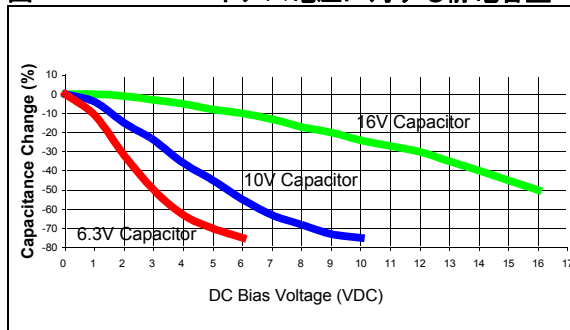
一般的に入手可能な低価格の 10  $\mu\text{F}$  セラミック コンデンサの温度特性は X5R、X7R、Y5V です (その他のタイプも入手可能ですが、一般的ではありません)。これらのコンデンサの初期公差は、多くの場合  $\pm 10\% \sim \pm 20\%$  (X5R、X7R) または  $-20\%/+80\%$  (Y5V) です。しかし、これらのコンデンサをアプリケーション回路で使った時の実効静電容量は、DC バイアス電圧や温度等の要因によって変化します。従って回路内での総公差は、仕様上の公差よりも大幅に大きくなります。

一般的に、X5R および X7R タイプのコンデンサは、十分に良好な温度特性を備えています (広い温度レンジで  $\pm 15\%$ )。正確な仕様値については、メーカーのデータシートを参照してください。しかし、Y5V タイプのコンデンサの温度特性仕様値は、これより大幅に劣ります ( $+22\%/-82\%$ )。このため、公称値 10  $\mu\text{F}$  の Y5V タイプ コンデンサでは、最低限の  $V_{\text{DDCORE}}$  電圧レギュレータ安定性と過渡応答要件を満たせる十分な総静電容量を確保できません。従って、アプリケーションの想定動作温度レンジが広い場合、 $V_{\text{DDCORE}}$  レギュレータ用に Y5V タイプのコンデンサは推奨しません。

温度の影響以外に、大容量セラミック コンデンサの実効静電容量は、コンデンサに印加される DC 電圧によっても大きく変化します。これは非常に大きく影響する可能性があります。見過ごされる事が多く、仕様書に明記されない事もあります。

図 2-4 に、X7R タイプおよび Y5V タイプのコンデンサの DC バイアス電圧と静電容量の関係を示す代表的なグラフを示します。

図 2-4: DC バイアス電圧に対する静電容量



$V_{\text{DDCORE}}$  電圧レギュレータ用にセラミック コンデンサを選ぶ際、動作電圧に対して最大定格電圧が十分に高いコンデンサの使用を推奨します。例えば、2.5 V  $V_{\text{DDCORE}}$  電圧に対しては、定格電圧 16 V のセラミック コンデンサを推奨します。推奨コンデンサは表 2-1 を参照してください。

## 2.5 ICSP ピン

PGC および PGD ピンは、In-Circuit Serial Programming™ (ICSP™) とデバッグ用に使います。ICSP コネクタとデバイスの ICSP ピン間のトレース長をできるだけ短くする事を推奨します。ICSP コネクタで ESD が予測される場合、数十  $\Omega$  の抵抗 (100  $\Omega$  以下) を直列に接続する事を推奨します。

PGC ピンと PGD ピンにプルアップ抵抗、直列ダイオード、コンデンサを接続する事は推奨しません (デバイスとプログラマ / デバッガ間の通信に干渉するため)。このようなディスクリート部品がアプリケーションに必要な場合、プログラミング / デバッグ時にこれらの部品を回路から取り外す必要があります。また、容量性負荷の制限、ピン入力電圧の HIGH ( $V_{\text{IH}}$ ) および LOW ( $V_{\text{IL}}$ ) 要件の詳細は、各デバイスのフラッシュ プログラミング仕様書に記載されている AC/DC 特性とタイミング要件を参照してください。

デバイス エミュレーションを行う場合、デバイスにプログラミングした「通信チャンネル選択」(すなわち PGCx/PGDx ピンの設定)が、ICSP とマイクロチップ社製デバッガ / エミュレータ ツール間の物理的接続に一致している事が必要です。

マイクロチップ社製開発ツールの接続に関する要件の詳細は、セクション 27.0「開発サポート」を参照してください。

## 2.6 外部オシレータピン

多くのマイクロコントローラでは、少なくとも2つのオシレータ（高速プライマリオシレータと低速セカンダリオシレータ）を選択できます（詳細は**セクション 3.0「オシレータの設定」**参照）。

オシレータ回路と MCU は、基板の同一面上に配置する必要があります。オシレータ回路は各オシレータピンの近くに配置する必要があります（回路部品とピン間のトレース長が0.5インチ（12mm）を超えない事）。負荷コンデンサは、オシレータ本体近くの同一基板面上に配置する必要があります。

オシレータ回路の周囲をグランドに接続した銅ベタで囲んで、周囲の回路から隔離します。この銅ベタは、MCU のグランドに直接接続する必要があります。銅ベタ内に信号トレースまたは電源トレースを配置しないでください。また、両面基板を使う場合、水晶振動子の裏側にはトレースを配置しないでください。

**図 2-5** に推奨レイアウトを示します。インラインパッケージの場合、片面基板でオシレータピンを完全に囲むことができます。ファインピッチパッケージでは、オシレータピンと部品を完全に囲む事ができません。この場合、囲みが途切れる箇所を、裏面のグランド層に接続する事により適切に対処できます。どのような場合も、ガードトレースをグランドに戻す必要があります。

アプリケーションの回路パターンと I/O 割り当てを検討する際に、オシレータに近い（オシレータピンと隣り合う）ポートピンと信号が影響を及ぼさないように配慮する必要があります（高周波、急峻な立ち上がり/立ち下がり、その他のノイズからの遮断）。

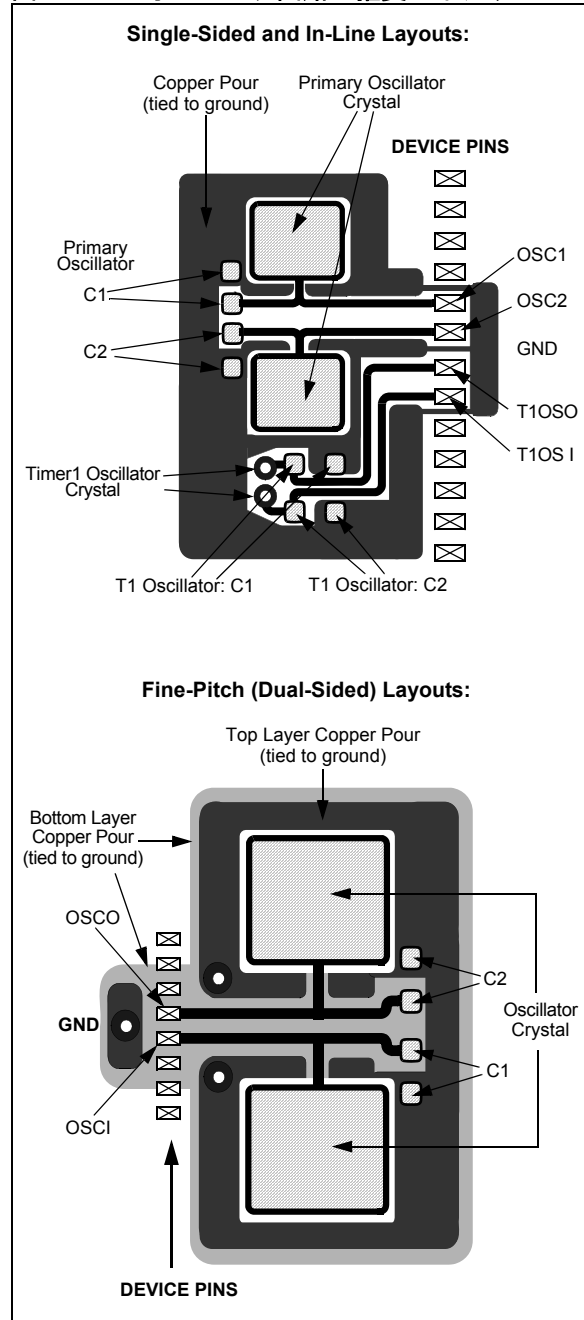
オシレータ回路の設計に関するその他の情報は、マイクロチップ社ウェブサイト ([www.microchip.com](http://www.microchip.com)) で下記のアプリケーションノートを参照してください。

- AN826 - 『Crystal Oscillator Basics and Crystal Selection for rPIC™ and PICmicro® Devices』
- AN849 - 『Basic PICmicro® Oscillator Design』
- AN943 - 『Practical PICmicro® Oscillator Analysis and Design』
- AN949 - 『Making Your Oscillator Work』

## 2.7 未使用 I/O

未使用の I/O ピンは、出力として設定し、論理 LOW 状態に駆動する必要があります。または、1 ~ 10 KΩ の抵抗を未使用ピンと V<sub>SS</sub> の間に接続して出力を論理 LOW に駆動します。

**図 2-5: オシレータ回路の推奨レイアウト**



# PIC18F97J60 ファミリ

---

NOTES:



## 3.0 オシレータの設定

### 3.1 概要

PIC18F97J60 ファミリが内蔵しているオシレータとクロックシステムは、標準の PIC18FXXJXX とは異なります。安定した 25 MHz のクロック源を必要とする Ethernet モジュールが追加されているため、プライマリ オシレータはマイクロコントローラの幅広いクロック速度に加えてこの周波数も供給する必要があります。図 3-1 に、オシレータの構成の概要を示します。

内部 RC オシレータやクロック切り換え等、PIC18FXXJXX 拡張マイクロコントローラで使われるその他のオシレータ機能は共通です。これらの機能については、本章の後半で説明します。

### 3.2 オシレータの種類

PIC18F97J60 ファミリは、以下の 5 つのオシレータモードで動作します。

1. HS 高速水晶振動子 / 振動子
2. HSPLL ソフトウェア PLL 制御付き  
高速水晶振動子 / 振動子
3. EC 1/4Fosc 出力の外部クロック
4. ECPLL ソフトウェア PLL 制御付き  
外部クロック
5. INTRC 内部 31 kHz オシレータ

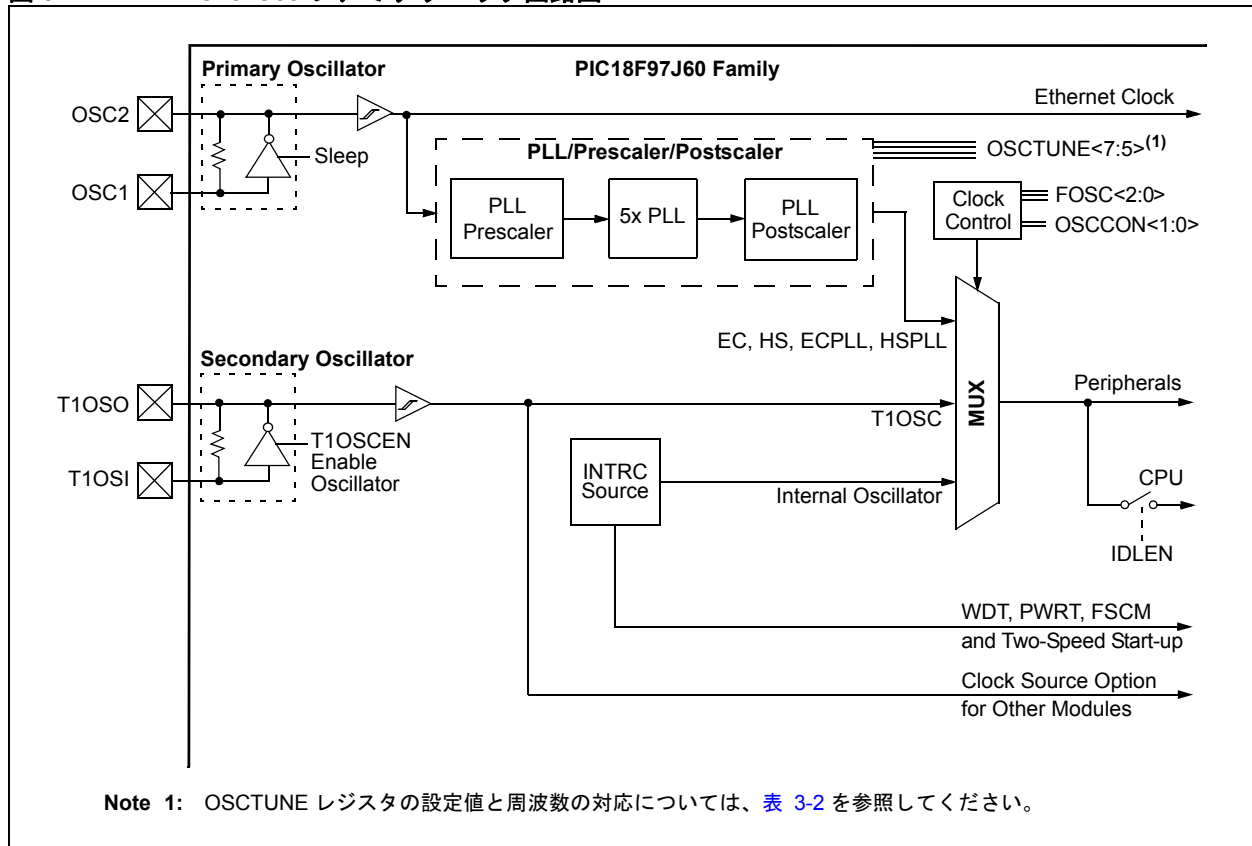
#### 3.2.1 オシレータ制御

オシレータモードは、FOSC<2:0> コンフィグレーションビットで選択します。FOSC<1:0> ビットで既定値のプライマリ オシレータモードを選択しますが、INTRC に切り換える場合は FOSC2 で選択します。

OSCCON レジスタ (レジスタ 3-2) はアクティブなクロックモードを選択します。これは、主に電力管理モードにおけるクロック切り換えに使用します。詳細は、[セクション 3.7.1「オシレータ制御レジスタ」](#)を参照してください。

OSCTUNE レジスタ (レジスタ 3-1) は、PLL の有効/無効とプリスケラ / ポストスケラの設定を行い、プライマリ オシレータ信号源からのシステムクロック周波数を選択するために使用します。詳細は、[セクション 3.6.1「PLL ブロック」](#)を参照してください。

図 3-1: PIC18F97J60 ファミリ クロック回路図



# PIC18F97J60 ファミリ

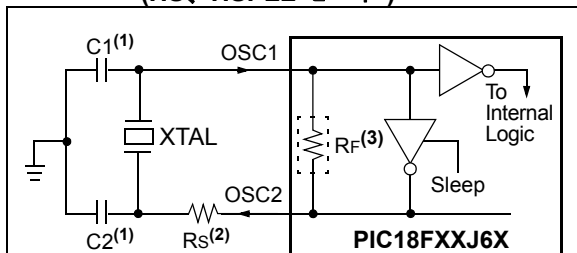
## 3.3 水晶振動子 / セラミック振動子 (HS モード)

HS または HSPLL オシレータモードでは、OSC1 および OSC2 ピンに水晶振動子を接続して発振回路を形成します。図 3-2 にピンの接続を示します。

このオシレータ回路では、並列共振動作の水晶振動子を使う必要があります。

**Note:** 直列共振動作水晶振動子を使うと、水晶振動子メーカーの仕様範囲外の周波数が発生する事があります。

図 3-2: 水晶発振回路の動作 (HS、HSPLL モード)



- Note 1:** C1とC2の初期値は表 3-1を参照してください。
- 2:** 水晶振動子によっては、オーバードライブを防ぐために直列抵抗 (RS) を必要とします。
- 3:** Rf は選択したオシレータモードによって変化します。

表 3-1: 水晶発振回路用コンデンサの選定

タイプ	水晶振動子周波数	検証済みのコンデンサの代表値	
		C1	C2
HS	25 MHz	33 pF	33 pF

ここに示すコンデンサの値は設計上の指針に過ぎません。

許容範囲内のオシレータ動作を得るためには、異なる値のコンデンサが必要となる事があります。アプリケーションで予測される VDD および温度レンジ全体でオシレータの性能を検証してください。オシレータに関する参考情報として、以下のアプリケーションノートも参照してください。

- AN588 - 『PIC® Microcontroller Oscillator Design Guide』
- AN826 - 『Crystal Oscillator Basics and Crystal Selection for rfPIC® and PIC® Devices』
- AN849 - 『Basic PIC® Oscillator Design』
- AN943 - 『Practical PIC® Oscillator Analysis and Design』
- AN949 - 『Making Your Oscillator Work』

詳細は、この表の Note を参照してください。

**Note 1:** 静電容量を大きくするとオシレータの安定性は増すものの、起動に要する時間も増大します。

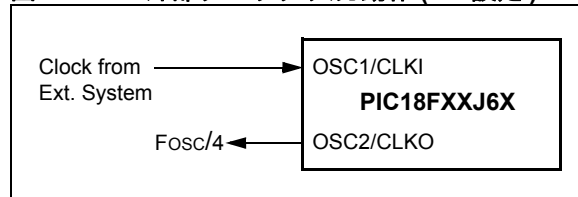
- 2:** 水晶振動子は製品ごとに固有の特性を持つため、外付け部品の適切な数値は水晶振動子のメーカーに問い合わせてください。
- 3:** 水晶振動子のオーバードライブを防ぐために Rs を必要とする場合があります。
- 4:** アプリケーションで予測される VDD および温度レンジ全体でオシレータの性能を検証してください。

## 3.4 外部クロック入力 (EC モード)

EC および ECPLL オシレータモードでは、外部クロック源を OSC1 ピンに接続する必要があります。パワーオンリセット (POR) 後またはスリープからの復帰後にオシレータ起動タイムの遅延は不要です。

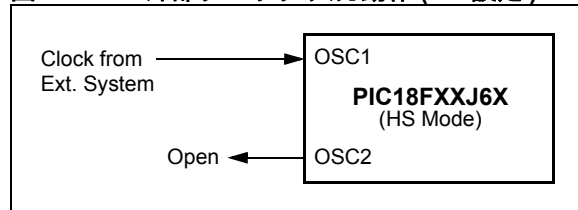
EC オシレータモードでは、オシレータの周波数を 4 分周した周波数が OSC2 ピンから出力されます。この信号はテスト目的または他のロジックの同期用として使えます。図 3-3 に、EC オシレータモードのピン接続を示します。

図 3-3: 外部クロック入力動作 (EC 設定)



HS モードで外部クロック源を OSC1 ピンに接続する事もできます (図 3-4)。この設定では、OSC2 ピンは開放のままです。この設定では内部オシレータのフィードバック回路が有効なため、EC モードよりも消費電流が若干大きくなります (EC モードではフィードバック回路が無効)。

図 3-4: 外部クロック入力動作 (HS 設定)



## 3.5 内部オシレータ ブロック

PIC18F97J60 ファミリの内部オシレータ信号源 (INTRC) は、公称 31 kHz のクロックを出力します。INTRC はデバイスの電源投入時に有効になり、デバイスがコンフィグレーションサイクルの間(すなわち動作モードに移行するまで)クロック源として使われます。この他、INTRC をデバイスのクロック源として選択した場合、または以下のいずれかを有効にした場合も INTRC が有効になります。

- フェイルセーフ クロックモニタ
- ウォッチドッグ タイマ
- 2 段階起動

これらの機能の詳細は、[セクション 25.0「CPU の特殊機能」](#) で説明します。

FOSC2 コンフィグレーション ビットの設定により、INTRC をデバイス起動時の既定値のクロック源として設定する事もできます。詳細は、[セクション 3.7.1「オシレータ制御レジスタ」](#) を参照してください。

## 3.6 Ethernet 動作とマイクロコントローラ クロック

PIC18F97J60 ファミリは幅広い種類の水晶振動子と外部オシレータ入力に対応していますが、Ethernet アプリケーションで使う場合は 25 MHz のクロック源が

必須です。Ethernet に必要なクロックを別の周波数のプライマリ オシレータ信号源から内部生成する仕組みは用意されていません。また、周波数の許容誤差も仕様で定められており、セラミック振動子の使用も適しません。詳細は[セクション 28.0「電気的特性」](#)、[表 28-6](#) のパラメータ 5 を参照してください。

### 3.6.1 PLL ブロック

幅広いアプリケーションとマイクロコントローラ クロック速度をサポートするため、クロックシステムには別途 PLL ブロックが組み込まれています。このブロックは以下の 3 つで構成されます。

- 設定可能なプリスケアラ (1:2 または 1:3)
- 5x PLL 周波数マルチプライヤ
- 設定可能なポストスケアラ (1:1、1:2、1:3)

これら PLL ブロック各部の動作は、OSCTUNE レジスタ ([レジスタ 3-1](#)) で設定します。PLL の有効 / 無効、PLL ブロックのプリスケアラとポストスケアラの設定を行う事で、プライマリ オシレータの 25 MHz をそのまま使う事も含め、幅広いシステムクロック周波数を生成できます。[表 3-2](#) に、Ethernet 動作と互換の全オシレータ設定を示します。

レジスタ 3-1: OSCTUNE: PLL ブロック制御レジスタ

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
PPST1	PLLEN <sup>(1)</sup>	PPST0	PPRE	—	—	—	—
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

- bit 7      **PPST1:** PLL ポストスケアラ コンフィグレーション ビット  
 1 = 2 分周  
 0 = 3 分周
- bit 6      **PLLEN:** 5x 周波数マルチプライヤ PLL イネーブルビット <sup>(1)</sup>  
 1 = PLL を有効にする  
 0 = PLL を無効にする
- bit 5      **PPST0:** PLL ポストスケアラ イネーブルビット  
 1 = ポストスケアラを有効にする  
 0 = ポストスケアラを無効にする
- bit 4      **PPRE:** PLL プリスケアラ コンフィグレーション ビット  
 1 = 2 分周  
 0 = 3 分周
- bit 3-0      **未実装:** 「0」として読み出し

**Note 1:** オシレータが ECPLL または HSPLL モードの場合のみです。それ以外のモードではこのビットは使えず、「0」として読み出されます。

# PIC18F97J60 ファミリ

表 3-2: PLL ブロックの各種設定とデバイスのクロック速度

5x PLL	PLL プリスケーラ	PLL ポストスケーラ	PLL ブロックの設定 (OSCTUNE<7:4>)	クロック周波数 (MHz)
有効	÷2	無効	×101	(Note 1)
		÷2	1111	31.2500
		÷3	0111	20.8333
	÷3	無効	×100	41.6667
		÷2	1110	20.8333
		÷3	0110	13.8889
無効	無効 <sup>(2)</sup>	無効	×00×	25 (既定値)
	÷2	÷2	1011	6.2500
		÷3	0011	4.1667
	÷3	÷2	1010	4.1667
		÷3	0010	2.7778

凡例: × = ドントケア

**Note 1:** 予約済みの設定です。この設定で生成されるクロック周波数はマイクロコントローラの動作レンジを超えています。

**2:** PLL とポストスケーラを両方無効にすると、プリスケーラは自動的に無効になります。

## 3.7 クロック源とオシレータの切り換え

PIC18F97J60 ファミリにはデバイスのクロック源をメインオシレータから代替クロック源へ切り換える機能があります。また、これらのデバイスには2つの代替クロック源があります。代替クロック源を有効にすると、各種の電力管理動作モードを使えます。

基本的に、これらのデバイスには3つのクロック源があります。

- プライマリ オシレータ
- セカンダリ オシレータ
- 内部オシレータ ブロック

**プライマリ オシレータ**には、外部水晶振動子 / 振動子モードと外部クロックモードがあります。オシレータモードは、FOSC<2:0> コンフィグレーション ビットで定義します。これらのモードの詳細は、本章の前半で説明しています。

**セカンダリ オシレータ**とは、OSC1 または OSC2 ピン以外に接続される外部信号源をいいます。これらの信号源は、コントローラが電力管理モードに移行した後も動作を継続できます。PIC18F97J60 ファミリには、セカンダリ オシレータとしてTimer1オシレータがあります。全ての電力管理モードで、このオシレータはリアルタイムクロック (RTC) 等の機能のタイムベースとしてよく使われます。

最も一般的な例として、32.768 kHz の時計用水晶振動子を RC0/T1OSO/T13CKI ピンと RC1/T1OSI ピンの間に接続します。また、各ピンからグランドへ負荷コンデンサも接続します。Timer1 オシレータについては、[セクション 13.3「Timer1 オシレータ」](#)で詳しく説明します。

**内部オシレータ**はプライマリ クロック源としてだけでなく、電力管理モード時のクロック源としても使えます。INTRC は、WDT やフェイルセーフ クロックモニタ (FSCM) 等いくつかの特殊機能のクロック源としても使います。

[図 3-1](#) に、PIC18F97J60 ファミリのクロック源を示します。コンフィグレーション レジスタの詳細は、[セクション 25.0「CPU の特殊機能」](#)を参照してください。

## 3.7.1 オシレータ制御レジスタ

OSCCON レジスタ (レジスタ 3-2) は、通常の動作モード時と電力管理モード時におけるデバイスクロック動作の一部を制御します。

システムクロック選択ビット (SCS<1:0>) でクロック源を選択します。ここで選択できるクロック源は、プライマリクロック (FOSC<2:0> コンフィグレーションビットで定義)、セカンダリクロック (Timer1 オシレータ)、内部オシレータのいずれかです。システムクロック選択ビットを変更すると、クロック切り換えのための短い遅延後、クロック源が切り換わります。

OSTS (OSCCON<3>) ビットと T1RUN (T1CON<6>) ビットは、デバイスの現在のクロック源を示します。セカンダリクロックモードで Timer1 オシレータがデバイスにクロックを供給している場合、T1RUN ビットに示されます。電力管理モードでは、これら 2 つのビットが同時にセットされる事はありません。どちらのビットもセットされていなければ、INTRC がクロック源であるか、または内部オシレータが起動したばかりでまだ安定していない事を示します。

IDLEN ビットは、SLEEP 命令を実行した場合にスリープとアイドルのどちらに移行するかを決定します。

OSCCON レジスタのフラグビットと制御ビットの使用法については、セクション 4.0「電力管理モード」で詳しく説明します。

**Note 1:** セカンダリ クロック源を選択するには、Timer1 オシレータを有効にする必要があります。Timer1 オシレータを有効にするには、Timer1 制御レジスタ (T1CON<3>) の T1OSCCEN ビットをセットします。Timer1 オシレータが無効の場合、セカンダリ クロック源を選択しても無視されます。

**2:** SLEEP 命令の実行前に Timer1 オシレータを動作させ、安定化させておく事を推奨します。そうしないと、Timer1 オシレータの起動中に非常に長い遅延が発生する場合があります。

レジスタ 3-2: OSCCON: オシレータ制御レジスタ

R/W-0	U-0	U-0	U-0	R-q	U-0	R/W-0	R/W-0
IDLEN	—	—	—	OSTS <sup>(1)</sup>	—	SCS1	SCS0
bit 7							bit 0

<b>凡例:</b>	q = 設定による
R = 読み出し可能ビット	W = 書き込み可能ビット
-n = POR 時の値	「1」= ビットはセット
	「0」= ビットはクリア
	x = ビットは未知

- bit 7 **IDLEN:** アイドル イネーブルビット
  - 1 = SLEEP 命令によってデバイスがアイドルに移行する
  - 0 = SLEEP 命令によってデバイスがスリープに移行する
- bit 6-4 **未実装:** 「0」として読み出し
- bit 3 **OSTS:** オシレータ ステータスビット<sup>(1)</sup>
  - 1 = SCS<1:0> = 00 の場合、定義されたオシレータ信号源でデバイスが動作している
  - 0 = SCS<1:0> = 01、10、11 の場合、定義されたオシレータ信号源でデバイスが動作している
- bit 2 **未実装:** 「0」として読み出し
- bit 1-0 **SCS<1:0>:** システムクロック選択ビット
  - 11 = 内部オシレータ
  - 10 = プライマリ オシレータ
  - 01 = Timer1 オシレータ

**FOSC2 = 1 の場合:**

  - 00 = プライマリ オシレータ

**FOSC2 = 0 の場合:**

  - 00 = 内部オシレータ

**Note 1:** リセット値は 2 段階起動が有効な場合が「0」で、無効な場合が「1」です。

# PIC18F97J60 ファミリ

## 3.7.1.1 システムクロック選択と FOSC2 コンフィグレーションビット

SCS ビットは全てのリセットでクリアされます。つまり、デバイスの既定値の設定では FOSC<1:0> で定義したプライマリ オシレータ (すなわち HC/EC モードのいずれか) がデバイスリセット時にプライマリ クロック源として使われます。

リセット時に既定値として使われるクロック源の設定は、FOSC2 コンフィグレーション ビットで変更できます。このビットは、SCS<1:0> = 00 の場合のクロック源を選択します。FOSC2 = 1 (既定値) の場合、SCS<1:0> = 00 の時のオシレータ信号源は FOSC<1:0> の設定によって決まります。FOSC2 = 0 の場合、SCS<1:0> = 00 の時のオシレータ信号源は INTRC です。SCS ビットはリセットでクリアされるため、FOSC2 の設定によってリセット時の既定値のオシレータモードも変更されます。

FOSC2 の設定にかかわらず、INTRC はデバイスの電源投入時に常に有効になります。INTRC は、デバイスがメモリからコンフィグレーションの値を読み込むまでクロック源として使われます。FOSC コンフィグレーション ビットを読み込んだ時点で、その設定に基づいたオシレータ信号源で動作が開始します。

FOSC2 の設定により、プライマリ クロックと内部オシレータのどちらか一方が 2 つのビット設定値を持ちます。

## 3.7.2 オシレータの移行

PIC18F97J60 ファミリは、クロック源の切り換え時に発生するクロックの「グリッチ」を防止する回路を内蔵しています。クロック切り換え時には、デバイスのクロックがわずかな間停止します。停止時間の長さは、切り換え前のクロック源 2 サイクル分と切り換え後のクロック源 3 ~ 4 サイクル分の合計です。この仕組みにより、切り換え後のクロック源の確実な安定をはかっています。

クロック移行の詳細は、[セクション 4.1.2 「電力管理モードへの移行」](#) で詳しく説明します。

## 3.8 各種クロック源に対する電力管理 モードの影響

PRI\_IDLE モードを選択すると、指定したプライマリ オシレータが常時動作を継続します。その他全ての電力管理モードでは、OSC1 ピンを使ったオシレータは無効になります。OSC1 ピン (およびオシレータで使っている場合は OSC2 ピン) は発振を停止します。

セカンダリクロックモード (SEC\_RUN と SEC\_IDLE) では、Timer1 オシレータが動作を継続し、デバイスにクロックを供給します。また、Timer1 オシレータが Timer1 または Timer3 にクロックを供給している場合、Timer1 オシレータは全ての電力管理モードで動作を継続します。

RC\_RUN および RC\_IDLE モードでは、内部オシレータがデバイスにクロックを供給します。31 kHz の INTRC 出力を直接クロック源として使うと、電力管理モードにかかわらず各種の特殊機能を有効にできます (WDT、フェイルセーフクロックモニタ、2 段階起動の詳細は、[セクション 25.2 「ウォッチドッグタイマ \(WDT\)」](#) ~ [セクション 25.5 「フェイルセーフクロックモニタ」](#) を参照してください)。

スリープを選択した場合、全てのクロック源が停止します。トランジスタのスイッチング電流が完全に停止するため、スリープではデバイスの消費電流が最小です (リーク電流のみ)。

スリープ中も動作を継続する機能を使うと、スリープ中の消費電流が増大します。WDT を動作させるには、INTRC が必要です。リアルタイム クロックをサポートするには Timer1 オシレータの動作を継続する必要があります。その他、デバイスのクロック源を必要としない機能 (MSSP スレープ、PSP、INTx ピン等) は動作を継続する事があります。消費電流が顕著なモジュールについては、[セクション 28.2 「DC 特性: パワーダウン電流と消費電流 PIC18F97J60 ファミリ \(産業用\)」](#) を参照してください。

## 3.9 パワーアップ遅延

パワーアップ遅延は 2 つのタイマで制御されるため、ほとんどのアプリケーションで外付けのリセット回路は必要ありません。この遅延によって、通常環境においてデバイスの電源が安定し、プライマリ クロックが安定動作するまでデバイスを確実にリセット状態に保持します。パワーアップ遅延の詳細は、[セクション 5.6 「パワーアップタイマ \(PWRT\)」](#) を参照してください。

最初のタイマはパワーアップ タイマ (PWRT) で、パワーアップ時に一定の遅延 ([表 28-12](#) のパラメータ 33) を発生させます。PWRT は常に有効です。

2 番目のタイマはオシレータ起動タイマ (OST) で、水晶振動子が安定するまで (HS モード時) デバイスをリセット状態に保持します。OST は 1024 オシレータ サイクルをカウントした後、オシレータからのクロックをデバイスに供給します。

POR の発生からコントローラが命令を実行できるようになるまでには、TcSD ([表 28-12](#) のパラメータ 38) の遅延があります。

表 3-3: スリープ時の OSC1 および OSC2 ピンの状態

オシレータモード	OSC1 ピン	OSC2 ピン
EC, ECPLL	フローティング、外部クロックによってプルアップ/ダウン	論理 Low (1/4 クロック出力)
HS, HSPLL	フィードバック インバータは静止電圧レベルでは無効	フィードバック インバータは静止電圧レベルでは無効

**Note:** スリープと MCLR リセットによるタイムアウトについては、[セクション 5.0 「リセット」](#) の [表 5-2](#) を参照してください。

## 4.0 電力管理モード

PIC18F97J60 ファミリは、CPU と周辺モジュールへのクロック供給を管理して簡単に消費電力を抑える事ができます。一般的に、クロック周波数が低いほど、また、クロックの供給を受ける回路の数が少ないほど、消費電力は低下します。アプリケーションの電力管理のために、以下の3つの動作モードを用意しています。

- ランモード
- アイドル
- スリープ

これらのモードは、デバイスのどの部分にどれだけの速度のクロックを供給するかを定義します。ランモードとアイドルでは3つのクロック源(プライマリ、セカンダリ、内部オシレータ ブロック)のいずれかが使えますが、スリープではクロック源をしません。

これらの電力管理モードは、従来の PIC<sup>®</sup> MCU が搭載していたいくつかの省電力機能を含んでいます。例えば、他の PIC18 と同様に、コントローラのクロック源をプライマリ オシレータから Timer1 オシレータに切り換える機能がその1つです。また、全ての PIC MCU が備えているスリープも搭載しています。このモードではデバイスクロックが全て停止します。

### 4.1 電力管理モードの選択

電力管理モードを選択するには、(1) CPU へのクロック供給を継続するかどうか、(2) どのクロック源を使うか、を決定する必要があります。CPU へのクロック供給は IDLEN ビット (OSCCON<7>)、クロック源は SCS<1:0> ビット (OSCCON<1:0>) で選択します。表 4-1 に、各モードのビット設定、クロック源、影響を受けるモジュールをまとめます。

#### 4.1.1 クロック源

電力管理モードで使うクロック源は、SCS<1:0> ビットを使って以下の3つから選びます。

- FOSC<2:0> コンフィグレーション ビットで定義したプライマリ クロック
- セカンダリ クロック (Timer1 オシレータ)
- 内部オシレータ

#### 4.1.2 電力管理モードへの移行

電力管理モードを切り換えるには、まず OSCCON レジスタに値を書き込みます。SCS<1:0> ビットでクロック源を選択し、どのランモード / アイドルを使うかを決定します。これらのビットを変更すると、新しいクロック源が既に動作している場合はただちに切り換わります。切り換え時にも、クロック遷移の遅延が発生します。詳細については、[セクション 4.1.3 「クロック切り換えとステータス インジケータ」](#)以降のセクションで説明します。

アイドル / スリープへは SLEEP 命令の実行によって移行します。どちらのモードに移行するかは、IDLEN ビットの設定によって決まります。

切り換え前後のモードの組み合わせによっては、これら全てのビットを変更しなくてもモードを切り換えられる場合があります。多くの場合、SLEEP 命令を発行する前にオシレータ選択ビットまたは IDLEN ビットを変更するだけでモードを切り換える事ができます。IDLEN ビットが既に正しく設定されていれば、SLEEP 命令を実行するだけで所定のモードに切り換える事ができます。

表 4-1: 電力管理モード

モード	OSCCON<7,1:0>		モジュールへのクロック供給		利用可能なクロック / オシレータ源
	IDLEN <sup>(1)</sup>	SCS<1:0>	CPU	周辺モジュール	
スリープ	0	N/A	なし	なし	なし - 全てのクロックが停止
PRI_RUN	N/A	10	あり	あり	プライマリ - HS、EC、HSPLL、ECPLL (通常の実行モードです。消費電力を低減する電力管理モードではありません)
SEC_RUN	N/A	01	あり	あり	セカンダリ - Timer1 オシレータ
RC_RUN	N/A	11	あり	あり	内部オシレータ
PRI_IDLE	1	10	なし	あり	プライマリ - HS、EC、HSPLL、ECPLL
SEC_IDLE	1	01	なし	あり	セカンダリ - Timer1 オシレータ
RC_IDLE	1	11	なし	あり	内部オシレータ

**Note 1:** IDLEN は、SLEEP 命令実行時の値です。

# PIC18F97J60 ファミリ

## 4.1.3 クロック切り換えとステータスインジケータ

クロック源の切り換えにかかる時間の長さは、切り換え前のクロック源2サイクル分と切り換え後のクロック源3～4サイクル分の合計です。この仕組みにより、切り換え後のクロック源の確実な安定をはかっています。

現在のクロック源とその状態は、OSTS (OSCCON<3>) ビットと T1RUN (T1CON<6>) ビットの2つで表されます。通常、電力管理モードではこれらビットのどちらか1つだけがセットされます。OSTS ビットがセットされている場合、プライマリクロックがデバイスのクロック源です。T1RUN ビットがセットされている場合、Timer1 オシレータが現在のクロック源です。どちらのビットもセットされていない場合、INTRC がデバイスのクロック源です。

**Note:** SLEEP 命令を実行しても、デバイスが必ずスリープに移行するとは限りません。この命令は、IDLEN ビットの設定に応じて、コントローラをスリープまたはいずれかのアイドルに移行させるトリガとして機能します。

## 4.1.4 複数の SLEEP 命令

SLEEP 命令によってどの電力管理モードになるかは、命令実行時の IDLEN ビットの設定によって決まります。もう一度 SLEEP 命令を実行すると、デバイスはその時点の IDLEN ビットの設定に従って電力管理モードに移行します。IDLEN の値が変更されていれば、その新しい設定に基づいた別の電力管理モードに移行します。

## 4.2 ランモード

ランモードでは、コアと周辺モジュールの両方にクロックを供給します。ランモードには、クロック源の違いによって3つの種類があります。

### 4.2.1 PRI\_RUN モード

PRI\_RUN モードは、省電力機能を使わない通常の実行モードです。デバイスのリセット時、2段階起動が有効でない限りこのモードが既定値です(詳細は、[セクション 25.4「2段階起動」](#)参照)。このモードでは、OSTS ビットがセットされます([セクション 3.7.1「オシレータ制御レジスタ」](#)参照)。

### 4.2.2 SEC\_RUN モード

SEC\_RUN モードは、他の PIC18 が持つ「クロック切り換え」機能と互換のモードです。このモードでは、Timer1 オシレータが CPU と周辺モジュールにクロックを供給します。このモードでは、高精度のクロック源を使いながら消費電力を低減できます。

SCS<1:0> ビットを「01」に設定すると SEC\_RUN モードへ移行します。この場合、デバイスのクロック源は Timer1 オシレータに切り換わり(図 4-1 参照)、プライマリ オシレータは動作を停止し、T1RUN ビット (T1CON<6>) がセットされ、OSTS ビットがクリアされます。

**Note:** SEC\_RUNモードに移行するには、Timer1 オシレータが動作を開始済みである事が必要です。SCS<1:0> ビットを「01」に設定しても、その時点で T1OSCEN ビットがセットされていないければ、SEC\_RUN モードには移行しません。Timer1 オシレータが有効な場合でも、まだ動作していないければ、Timer1 オシレータが動作を開始するまでデバイスのクロックは供給されません。このような場合、動作開始直後のオシレータは動作が安定していないため、予測不能な動作が生じる事があります。

SEC\_RUN モードから PRI\_RUN モードへ移行する場合、プライマリクロックが完全に起動するまでの間、周辺モジュールと CPU には Timer1 オシレータからクロックが供給されます。プライマリクロックの準備が完了したら、プライマリクロックへの切り換えが行われます(図 4-2 参照)。クロックの切り換えが完了すると、T1RUN ビットがクリア、OSTS ビットがセットされ、プライマリクロックがクロックを供給します。IDLEN ビットと SCS ビットは復帰の影響を受けず、Timer1 オシレータは動作を継続します。



図 4-1: SEC RUN モードへの移行時の遷移タイミング

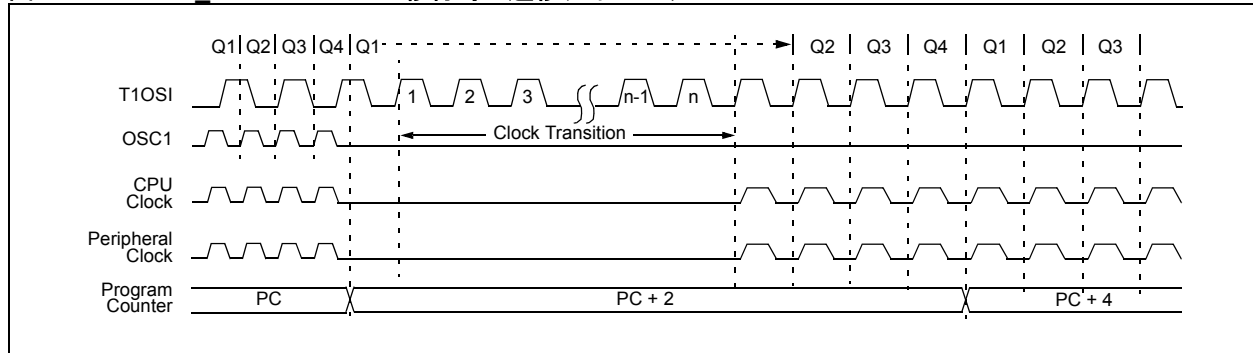
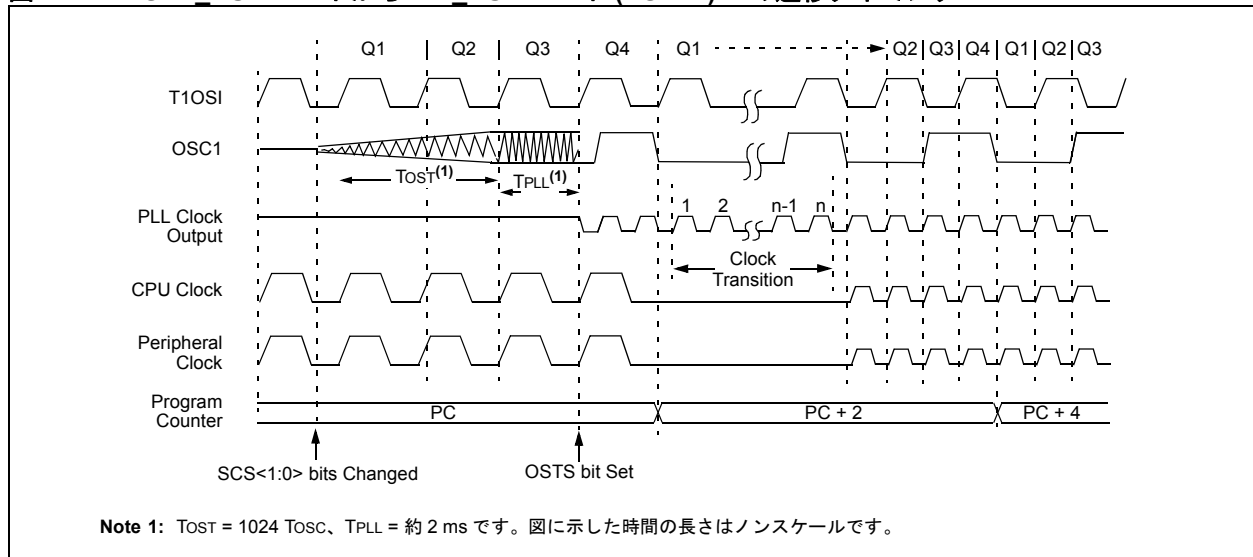


図 4-2: SEC RUN モードから PRI RUN モード (HSPLL) への遷移タイミング



Note 1: TOST = 1024 TOSC、TPLL = 約 2 ms です。図に示した時間の長さはノンスケールです。

# PIC18F97J60 ファミリ

## 4.2.3 RC\_RUN モード

RC\_RUN モードでは、内部オシレータが CPU と周辺モジュールにクロックを供給し、プライマリ クロックは停止します。このモードは、コード実行を継続できる3つのランモードの中で最も消費電力の少ないモードです。このモードは、それほど高いタイミング精度が要求されないユーザ アプリケーションまたは常時高速クロックで動作させる必要のないユーザ アプリケーションに適しています。

SCS<1:0> を「11」に設定すると、このモードに移行します。クロック源を INTRC に切り換えると (図 4-3 参照)、プライマリ オシレータは停止し、OSTS ビットはクリアされます。

RC\_RUN モードから PRI\_RUN モードへ移行する場合、プライマリ クロックが完全に起動するまでの間、デバイスには INTRC からのクロックが供給されます。プライマリ クロックの準備が完了したら、プライマリ クロックへの切り換えが行われます (図 4-4 参照)。クロックの切り換えが完了すると、OSTS ビットがセットされ、プライマリ クロックがデバイスにクロックを供給します。IDLEN ビットと SCS ビットは、クロック切り換えの影響を受けません。WDT またはフェイルセーフクロックモニタ (FSCM) のいずれかが有効な場合、INTRC は動作を継続します。

図 4-3: RC\_RUN モードへの遷移タイミング

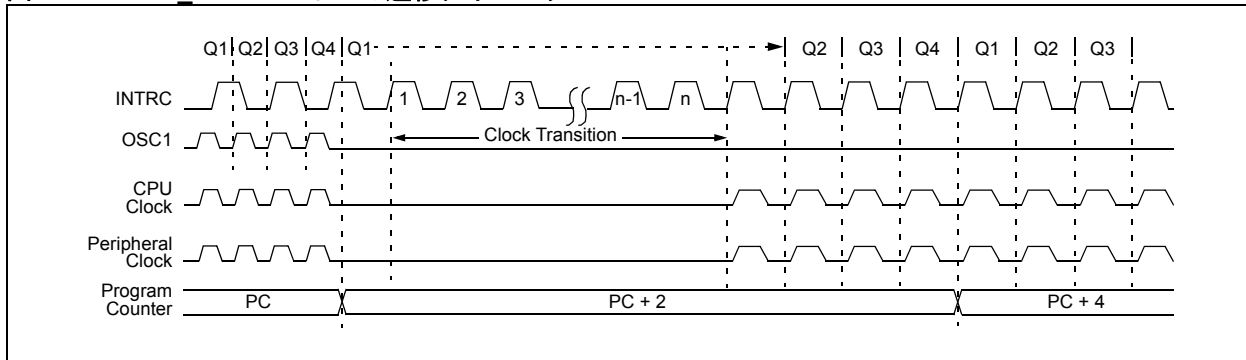
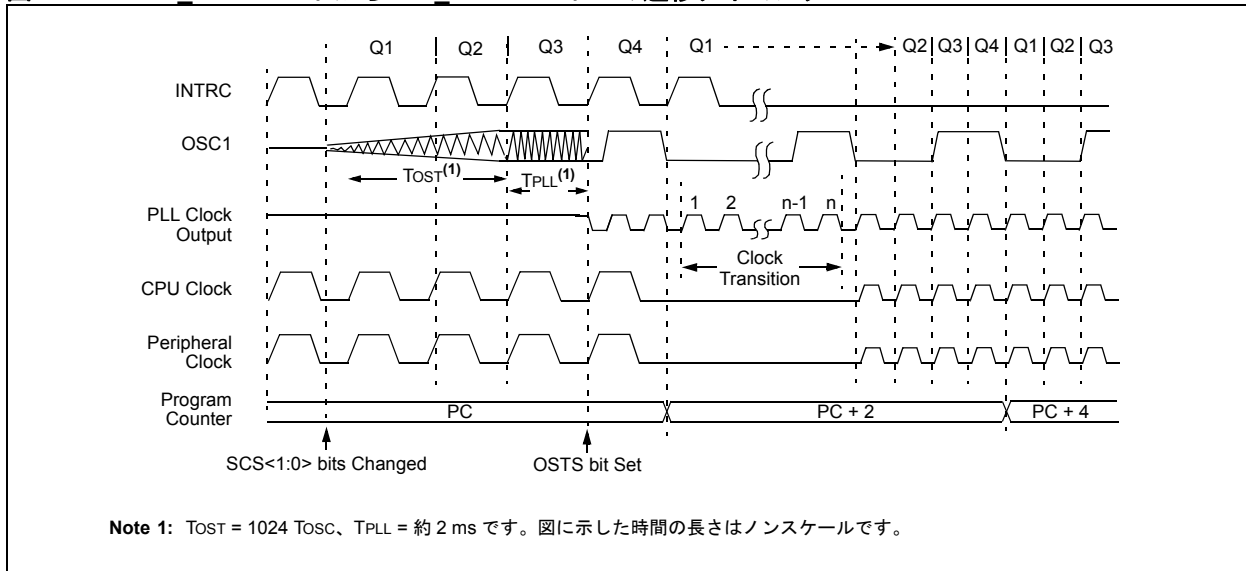


図 4-4: RC\_RUN モードから PRI\_RUN モードへの遷移タイミング



Note 1: TOST = 1024 TOSC, TPLL = 約 2 ms です。図に示した時間の長さはノンスケールです。

## 4.3 スリープ

電力管理モードとしてのスリープは、全ての PIC MCU が備える従来のスリープと同じものです。IDLEN ビットをクリアした状態（デバイスリセット時の既定値）で SLEEP 命令を実行すると、このモードに移行します。スリープに移行するとオシレータは停止します（図 4-5）。クロック源ステータスビットは全てクリアされます。

他のモードからスリープに移行する場合、クロックの切り換えは不要です。これは、コントローラがスリープに移行するとクロックを必要としないためです。WDT を有効にした場合、INTRC は動作を継続します。Timer1 オシレータを有効に設定している場合、Timer1 オシレータも動作を継続します。

スリープで（割り込み、リセット、WDT タイムアウト等の要因によって）復帰イベントが発生しても、SCS<1:0> ビットで選択したクロック源の準備が完了するまでデバイスにはクロックが供給されません（図 4-6）。ただし、2 段階起動またはフェイルセーフクロックモニタ (FSCM) のいずれか（[セクション 25.0 「CPU の特殊機能」](#) 参照）を有効にしている場合、内部オシレータからクロックが供給されます。どちらの場合も、プライマリクロックがデバイスにクロックを供給している場合は OSTS ビットがセットされます。復帰後も、IDLEN ビットと SCS ビットは変化しません。

## 4.4 アイドル

アイドルでは、CPU へのクロック供給のみが選択的に停止し、周辺モジュールは動作を継続します。3 種類のアイドルを使い分ける事によって、消費電力をさらに細かく管理する事ができます。

IDLEN ビットを「1」にセットした状態で SLEEP 命令を実行すると、周辺モジュールは SCS<1:0> ビットで選択したクロック源で動作しますが、CPU へのクロック供給は停止します。クロック源ステータスビットは影響を受けません。IDLEN ビットをセットして SLEEP 命令を実行すると、現在のランモードと同じクロック源を使ったアイドルへ簡単に切り換える事ができます。

WDT を有効にした場合、INTRC は動作を継続します。Timer1 オシレータを有効に設定している場合、Timer1 オシレータも動作を継続します。

CPU による命令実行は停止しているため、アイドルから復帰するには割り込み、WDT タイムアウト、リセットのいずれかを必要とします。これらの復帰イベントが発生すると、TcSD（表 28-12 のパラメータ 38）の遅延時間の後、CPU はコード実行を開始します。CPU は、現在のアイドルと同じクロック源でコード実行を再開します。例えば、RC\_IDLE モードから復帰すると RC\_RUN モードで、内部オシレータブロックが CPU と周辺モジュールにクロックを供給します。復帰後も、IDLEN ビットと SCS ビットは変化しません。

アイドルまたはスリープの時に WDT がタイムアウトになると WDT 復帰となり、SCS<1:0> ビットの現在の設定に基づいたランモードに復帰します。

図 4-5: スリープへの移行時の遷移タイミング

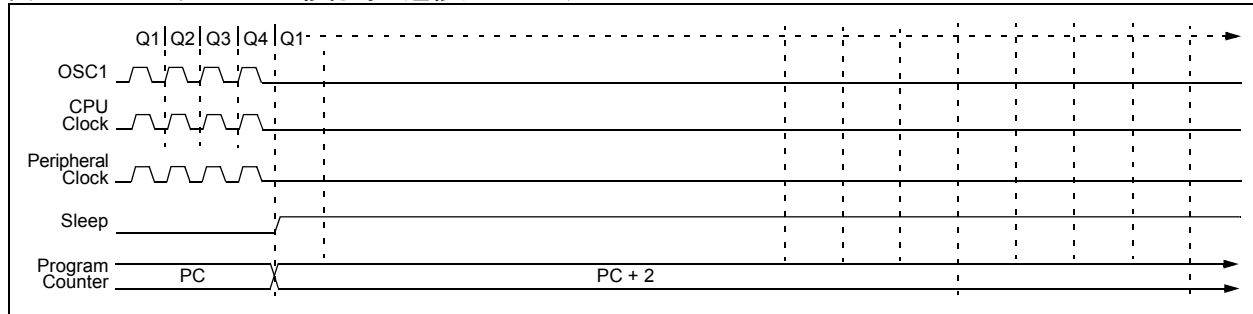
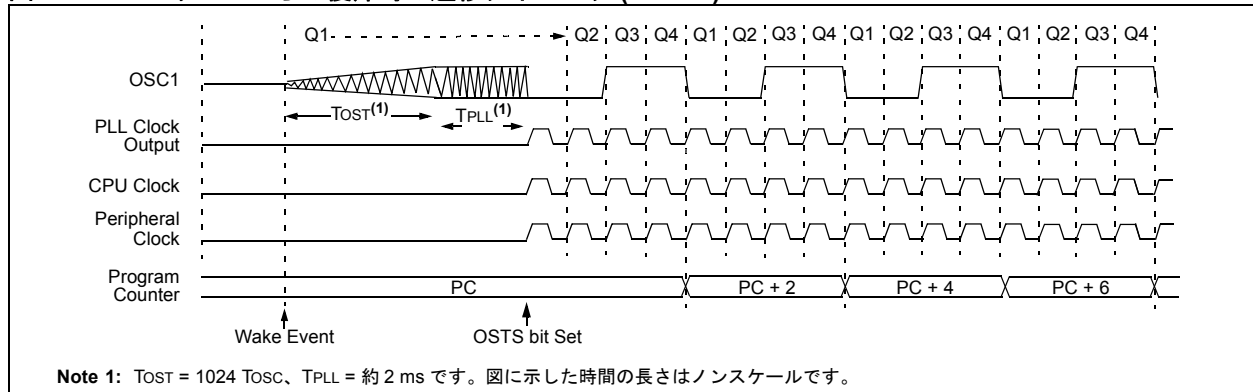


図 4-6: スリープからの復帰時の遷移タイミング (HSPLL)



# PIC18F97J60 ファミリ

## 4.4.1 PRI\_IDLE モード

3種類のアイドルのうち、このモードだけはプライマリクロックが停止しません。このため、クロック源の「ウォームアップ」または他のオシレータからの切り換えなしに、高精度なプライマリクロック源を使って高速にデバイスの動作を再開でき、高いタイミング精度が要求されるアプリケーションに適しています。

PRI\_RUN モードにおいて、IDLEN ビットをセットした状態で SLEEP 命令を実行すると PRI\_IDLE モードに移行します。その他のランモードで動作している場合、IDLEN ビットをセットした後で SCS<1:0> ビットを「10」に設定し、SLEEP 命令を実行します。このモードでは CPU は動作を停止しますが、周辺モジュールには FOSC<1:0> コンフィグレーションビットで指定したプライマリクロック源からのクロック供給が継続します。OSTS ビットはセットされたままです (図 4-7 参照)。

復帰イベントが発生すると、CPU はプライマリクロック源で動作を再開します。復帰イベントが発生してからコード実行が開始するまでの間には、TcSD の遅延時間が必要です。これは、CPU が命令を実行できるようになるまでに必要な時間です。復帰後も、OSTS ビットはセットされたままです。IDLEN ビットと SCS ビットは、復帰の影響を受けません (図 4-8 参照)。

## 4.4.2 SEC\_IDLE モード

SEC\_IDLE モードでは、CPU は動作を停止しますが、周辺モジュールには Timer1 オシレータからのクロック供給が継続します。SEC\_RUN モードにおいて、IDLEN ビットをセットした状態で SLEEP 命令を実行すると SEC\_IDLE モードに移行します。その他のランモードで動作している場合、IDLEN ビットをセットした後で SCS<1:0> ビットを「01」に設定し、SLEEP 命令を実行します。クロック源が Timer1 オシレータに切り換わるとプライマリオシレータは停止し、OSTS ビットがクリア、T1RUN ビットがセットされます。

復帰イベントが発生しても、周辺モジュールは Timer1 オシレータで動作を継続します。復帰イベントの発生から TcSD の遅延時間の後、CPU は Timer1 オシレータでコード実行を開始します。IDLEN ビットと SCS ビットは復帰の影響を受けず、Timer1 オシレータは動作を継続します (図 4-8 参照)。

**Note:** SEC\_IDLE モードに移行するには、Timer1 オシレータが動作を開始済みである必要があります。SLEEP 命令実行時に T1OSCEN ビットがセットされていなければ、SLEEP 命令は無視され、SEC\_IDLE モードへは移行しません。Timer1 オシレータが有効な場合でも、まだ動作していなければ、Timer1 オシレータが動作を開始するまで周辺モジュールのクロックは供給されません。このような場合、動作開始直後のオシレータは動作が安定していないため、予測不能な動作が生じる事があります。

図 4-7: アイドルへの移行時の遷移タイミング

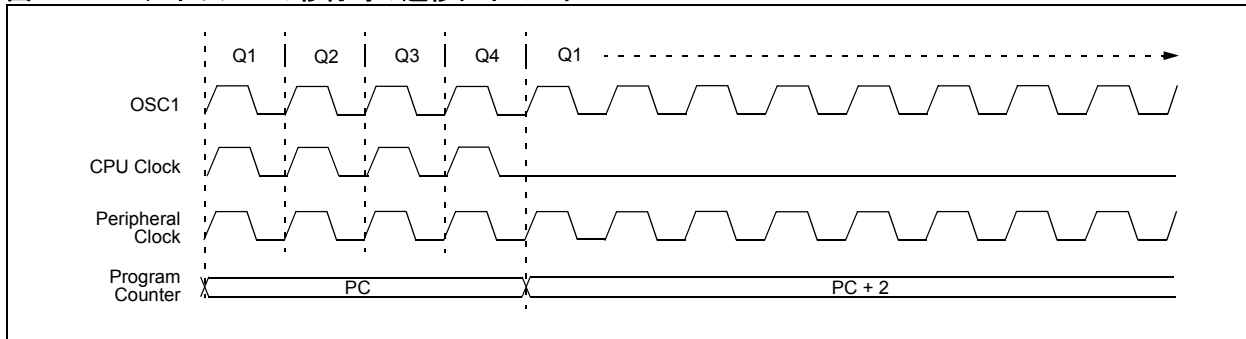
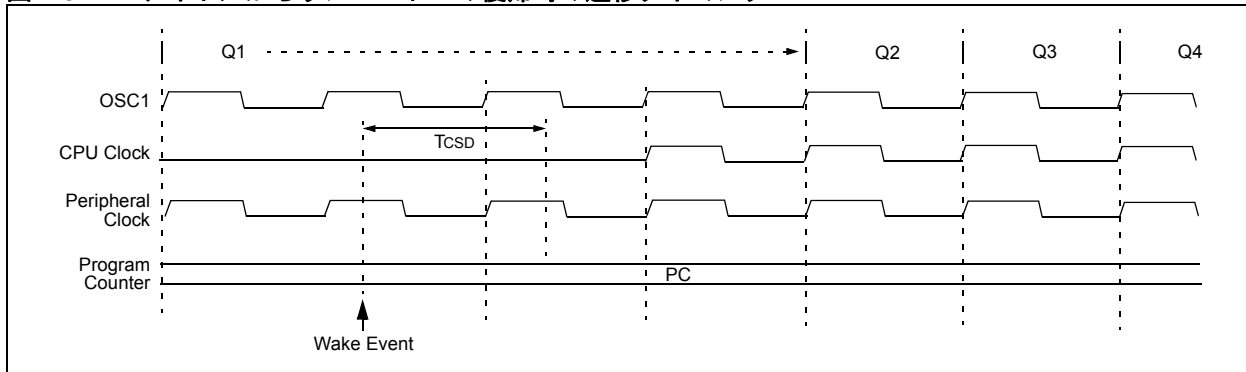


図 4-8: アイドルからランモードへの復帰時の遷移タイミング



## 4.4.3 RC\_IDLE モード

RC\_IDLE モードでは、CPU は動作を停止しますが、周辺モジュールには内部オシレータからのクロック供給が継続します。このモードでは、アイドル期間中の消費電力を制御できます。

RC\_RUN モードにおいて、IDLEN ビットをセットした状態で SLEEP 命令を実行すると RC\_IDLE モードに移行します。その他のランモードで動作している場合、IDLEN ビットをセットした後で SCS<1:0> ビットを「11」に設定し、SLEEP 命令を実行します。クロック源を INTRC に切り換えると、プライマリ オシレータは停止し、OSTS ビットがクリアされます。

復帰イベントが発生すると、周辺モジュールは INTRC からのクロックで動作を継続します。復帰イベントの発生から Tcsd の遅延時間の後、CPU は INTRC からのクロックでコード実行を開始します。復帰後も、IDLEN ビットと SCS ビットは変化しません。WDT またはフェイルセーフクロックモニタ (FSCM) のいずれかが有効な場合、INTRC は動作を継続します。

## 4.5 アイドル/スリープの終了

割り込み、リセット、WDT タイムアウトのいずれかが発生するとアイドル/スリープは終了します。ここでは、アイドル/スリープを終了させるトリガについて説明します。クロック供給サブシステムの動作については、各電力管理モードの説明 ( [セクション 4.2「ランモード」](#)、[セクション 4.3「スリープ」](#)、[セクション 4.4「アイドル」](#) ) を参照してください。

### 4.5.1 割り込みによる終了

利用可能な全ての割り込み要因は、デバイスをアイドルまたはスリープからランモードに移行させる事ができます。この機能を使うには、INTCON または PIE レジスタで割り込み要因のイネーブルビットをセットしておく必要があります。これで、対応する割り込みフラグビットがセットされるとアイドル/スリープ終了シーケンスが開始します。

GIE/GIEH ビット (INTCON<7>) がセットされている場合、割り込みによってアイドル/スリープを終了すると、コード実行は必ず割り込みベクタへ分岐します。それ以外の場合は分岐せず、そのままコード実行を継続または再開します ( [セクション 10.0「割り込み」](#) 参照 )。

アイドル/スリープを終了する際は、復帰イベントの発生から Tcsd の遅延時間が必要です。この遅延時間は、CPU が命令実行を開始できるようになるまでの時間を確保するために必要です。命令実行は、この遅延が経過した後の最初のクロックサイクルで再開されます。

### 4.5.2 WDT タイムアウトによる終了

WDT タイムアウト後の動作は、どの電力管理モードで WDT タイムアウトが発生したかによって異なります。

デバイスがコードを実行していない時 (アイドルまたはスリープ) に WDT タイムアウトになると、これらのモードが終了します ( [セクション 4.2「ランモード」](#)、[セクション 4.3「スリープ」](#) 参照 )。コード実行中 (ランモード) に WDT タイムアウトになると、WDT リセットとなります ( [セクション 25.2「ウォッチドッグタイマ \(WDT\)」](#) 参照 )。

WDT タイマとポストスケラは、以下のいずれかのイベントによってクリアされます。

- SLEEP または CLRWDT 命令を実行した場合
- フェイルセーフクロックモニタ (FSCM) 有効時に、現在選択中のクロック源が失われた場合

### 4.5.3 リセットによる終了

リセットによってアイドル/スリープが終了すると、デバイスのクロック源は自動的に INTRC に切り換わります。

### 4.5.4 オシレータ起動タイマ遅延を伴わない終了

OST を一切起動せずにアイドル/スリープを終了できる場合もあります。これには、以下の 2 つの場合があります。

- PRI\_IDLE モードの場合 (プライマリクロック源が停止していない)
- プライマリクロック源が EC または ECPLL モードの場合

これらの場合には、プライマリクロック源が既に動作している (PRI\_IDLE モード) か、または元々オシレータ起動遅延が必要ない (EC および ECPLL モード) ため、オシレータ起動遅延も発生しません。ただし、アイドル/スリープを終了する際は、復帰イベントが発生してから CPU がコード実行を開始できるようになるまでに Tcsd の遅延時間が必要である事には変わりありません。命令実行は、この遅延が経過した後の最初のクロックサイクルで再開されます。

# PIC18F97J60 ファミリ

---

NOTES:

## 5.0 リセット

PIC18F97J60 ファミリのリセットには、以下の種類があります。

- 通常動作中の  $\overline{\text{MCLR}}$  リセット
- 電力管理モード時の  $\overline{\text{MCLR}}$  リセット
- パワーオン リセット (POR)
- ブラウンアウト リセット (BOR)
- コンフィグレーション不一致 (CM)
- RESET 命令
- スタックフル リセット
- スタック アンダーフロー リセット
- コード実行中のウォッチドッグ タイマ (WDT) リセット

このセクションでは、ハードイベント ( $\overline{\text{MCLR}}$ )、電源イベント (POR, BOR)、コンフィグレーション不一致 (CM) によって発生するリセットについて説明します。また、各種起動タイマの動作についても説明します。スタックリセット イベントについては [セクション 6.1.6.4「スタックフルとスタック アンダーフロー](#)

[によるリセット](#)」で説明します。WDT リセットについては [セクション 25.2「ウォッチドッグ タイマ \(WDT\)」](#) で説明します。

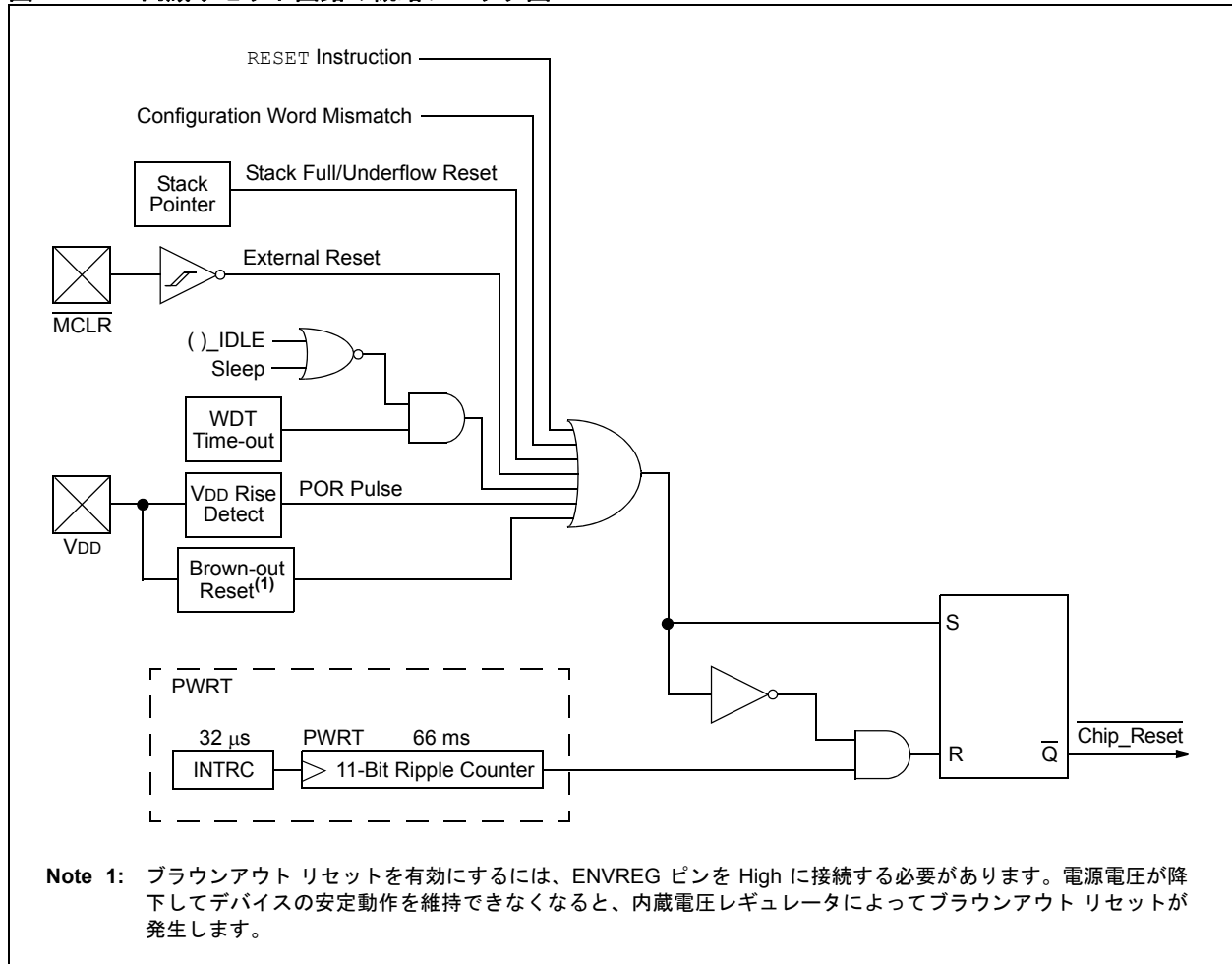
図 5-1 に、内蔵リセット回路の概略ブロック図を示します。

## 5.1 RCON レジスタ

デバイスリセット イベントは、RCON レジスタ ([レジスタ 5-1](#)) を使って監視します。このレジスタの下位 6 ビットは、発生したリセットイベントの種類を示します。ほとんどの場合、これらのビットはイベントによってのみセットされ、イベント発生後にアプリケーションでクリアする必要があります。これらのフラグビットの状態を全て読み出す事によって、発生したリセットの種類を特定できます。詳細は、[セクション 5.7「レジスタのリセット状態」](#) で説明します。

RCON レジスタには、割り込み優先度を設定する制御ビット (IPEN) もあります。割り込み優先度については [セクション 10.0「割り込み」](#) で説明します。

図 5-1: 内蔵リセット回路の概略ブロック図



# PIC18F97J60 ファミリ

## レジスタ 5-1: RCON: リセット制御レジスタ

R/W-0	U-0	R/W-1	R/W-1	R-1	R-1	R/W-0	R/W-0
IPEN	—	$\overline{\text{CM}}$	$\overline{\text{RI}}$	$\overline{\text{TO}}$	$\overline{\text{PD}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

- bit 7      **IPEN:** 割り込み優先度イネーブルビット  
 1 = 割り込みに対する優先度レベルの設定を有効にする  
 0 = 割り込みに対する優先度レベルの設定を無効にする (PIC16CXXX 互換モード)
- bit 6      **未実装:** 「0」として読み出し
- bit 5      **CM:** コンフィグレーション不一致フラグビット  
 1 = コンフィグレーション不一致リセットは発生していない  
 0 = コンフィグレーション不一致リセットが発生した (コンフィグレーション不一致リセット発生後にソフトウェアによるセットが必要である)
- bit 4      **RI:** RESET 命令フラグビット  
 1 = RESET 命令は実行されていない (ファームウェアのみセット可能である)  
 0 = RESET 命令が実行され、デバイスがリセットされた (ブラウンアウト リセット発生後はソフトウェアによるセットが必要である)
- bit 3      **TO:** ウォッチドッグ タイマ タイムアウト フラグビット  
 1 = 電源投入、CLRWDT 命令または SLEEP 命令によってセット  
 0 = WDT タイムアウトが発生した
- bit 2      **PD:** パワーダウン検出フラグビット  
 1 = 電源投入または CLRWDT 命令によってセット  
 0 = SLEEP 命令の実行によってセット
- bit 1      **POR:** パワーオン リセット ステータスビット  
 1 = パワーオン リセットは発生していない (ファームウェアのみセット可能である)  
 0 = パワーオン リセットが発生した (パワーオン リセット発生後はソフトウェアによるセットが必要である)
- bit 0      **BOR:** ブラウンアウト リセット ステータスビット  
 1 = ブラウンアウト リセットは発生していない (ファームウェアのみセット可能である)  
 0 = ブラウンアウト リセットが発生した (ブラウンアウト リセット発生後はソフトウェアによるセットが必要である)

**Note 1:** パワーオン リセットの検出後は、次のパワーオン リセットを検出できるように  $\overline{\text{POR}}$  ビットをセットする事を推奨します。

**2:** 内蔵電圧レギュレータが無効な場合、 $\overline{\text{BOR}}$  は常に「0」のままです。詳細は、[セクション 5.4.1「BOR の検出」](#)を参照してください。

**3:**  $\overline{\text{BOR}}$  が「0」かつ  $\overline{\text{POR}}$  が「1」の場合、ブラウンアウト リセットが発生したと考えられます (パワーオン リセット直後に  $\overline{\text{POR}}$  をソフトウェアで「1」にセットしている場合)。



## 5.2 マスタクリア ( $\overline{\text{MCLR}}$ )

$\overline{\text{MCLR}}$  ピンは、デバイスを外部からハードリセットする際に使います。このピンを Low に保持すると、リセットが発生します。PIC18 拡張マイクロコントローラの  $\overline{\text{MCLR}}$  リセット回路にはノイズフィルタがあり、微小パルスを検出しても無視します。

WDT を含めいづれの内部リセットも、 $\overline{\text{MCLR}}$  ピンを Low に駆動する事はありません。

## 5.3 パワーオンリセット (POR)

VDD が一定のしきい値を超えると、デバイス内部でパワーオンリセット条件が発生します。これにより、デバイスは VDD が十分に立ち上がってから初期化状態で起動します。

POR 回路を使うには、VDD と  $\overline{\text{MCLR}}$  ピンの間に抵抗 (1 ~ 10 k $\Omega$ ) を挟みます。これにより、外付けの RC 部品を使わずにパワーオンリセット遅延を確保できます。VDD の最小立ち上がりレートはパラメータ D004 に規定されています。立ち上がり時間が遅い場合、図 5-2 を参照してください。

デバイスがリセット条件を終了して通常動作を開始する際、デバイスの動作パラメータ (電圧、周波数、温度等) は正常動作の要件を満たす必要があります。これらの条件を満たさない場合、動作条件が満たされるまでデバイスをリセット条件に維持する必要があります。

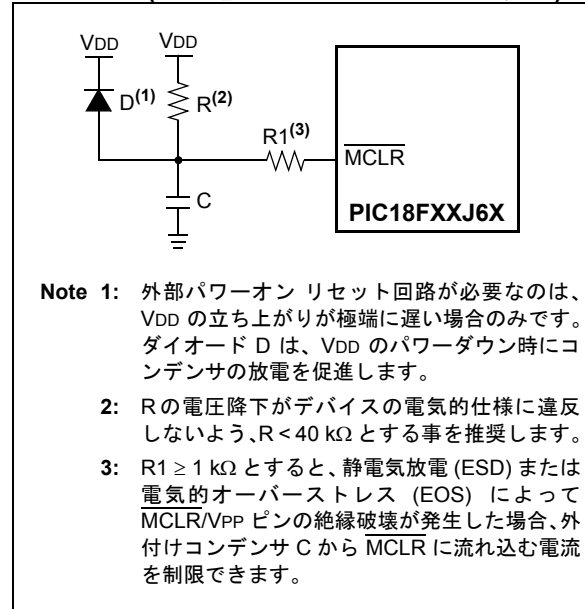
POR イベントが発生すると、 $\overline{\text{POR}}$  ビット (RCON<1>) にキャプチャされます。このビットは、POR イベントが発生すると「0」になりますが、他のリセットイベントでは変化しません。また、 $\overline{\text{POR}}$  ビットはいかなるハードウェア イベントによっても「1」になる事はありません。イベントを何度もキャプチャするには、パワーオンリセット後にソフトウェアでこのビットを「1」に戻しておく必要があります。

## 5.4 ブラウンアウトリセット (BOR)

PIC18F97J60 ファミリには、内蔵電圧レギュレータを有効 (ENVREG ピンを VDD に接続) にすると動作するシンプルな BOR 機能があります。VDD が  $V_{BOR}$  (パラメータ D005) を下回る時間が  $T_{BOR}$  (パラメータ 35) を超えると、デバイスはリセットされます。VDD が  $V_{BOR}$  を下回る時間が  $T_{BOR}$  より短い場合、リセットが発生するかどうかは確定しません。VDD が  $V_{BOR}$  を上回るまで、デバイスはブラウンアウトリセット状態のままです。

BOR が発生すると、パワーアップタイマによって TPWRT の期間 (パラメータ 33) だけデバイスがリセット状態に保持されます。パワーアップタイマの動作中に VDD が  $V_{BOR}$  を下回ると、デバイスは再びブラウンアウトリセット状態に戻り、パワーアップタイマは初期化されます。次に VDD が  $V_{BOR}$  を上回った時点からパワーアップタイマによる遅延が再び開始します。

図 5-2: 外部パワーオンリセット回路 (VDD 電源の立ち上がりが遅い場合)



### 5.4.1 BOR の検出

ブラウンアウトリセットまたはパワーオンリセットが発生すると、BOR ビットは必ず「0」にリセットされます。このため、BOR ビットを読み出すだけでは、ブラウンアウトリセットの発生を確実に検出する事はできません。 $\overline{\text{POR}}$  と BOR の状態を同時に確認の方がより確実です。これは、パワーオンリセットが発生した直後に  $\overline{\text{POR}}$  ビットをソフトウェアで「1」にリセットする事が前提です。BOR が「0」で  $\overline{\text{POR}}$  が「1」の場合、ブラウンアウトリセットが発生したと判断できます。

電圧レギュレータを無効にすると、ブラウンアウトリセット機能も無効になります。この場合、ブラウンアウトリセットが発生したかどうかを BOR ビットで判断する事はできません。この場合も、パワーオンリセットが発生すると BOR ビットはクリアされます。

## 5.5 コンフィグレーション不一致 (CM)

コンフィグレーション不一致 (CM) リセットは、メモリ内容のランダムな破損イベントを検出して回復を試みる機能です。例えば静電気放電 (ESD) イベントが発生するとデバイス全体にシングルビットの変化が広がり、致命的なエラーを引き起こす事があります。

PIC18FXXJ には、動作時にコンフィグレーションメモリ空間にあるデバイスコンフィグレーションレジスタを常時監視し、相補的なシャドーレジスタの内容と比較する機能があります。これらのレジスタの内容が一致しない場合、自動的に CM リセットが発生します。CM リセットが発生すると、そのイベントが  $\overline{\text{CM}}$  ビット (RCON<5>) にキャプチャされます。このビットは、CM リセットが発生すると「0」になりますが、他のリセットイベントでは変化しません。

# PIC18F97J60 ファミリ

CM リセットの挙動は、マスタクリアリセット、RESET 命令によるリセット、WDT タイムアウトリセット、スタックイベントリセットと同様です。ハードリセットとパワーリセットの発生時と同様、デバイスの再起動時にプログラムメモリ内のフラッシュ コンフィグレーションワードからデバイス コンフィグレーションワードに値が読み込まれます。

## 5.6 パワーアップ タイマ (PWRT)

PIC18F97J60 ファミリは、パワーオンリセットプロセスを安定化するためのパワーアップタイマ (PWRT) を内蔵しています。PWRT は常に有効で、その主な目的は、デバイスの電圧が安定してからコード実行を開始できるようにする事です。

PIC18F97J60 ファミリのパワーアップタイマ (PWRT) は、INTRC をクロック源とする 11 ビットのカウンタです。このタイマにより、約  $2048 \times 32 \mu\text{s} = 66 \text{ms}$  の遅延を確保します。PWRT がカウント中は、デバイスはリセット状態のままです。

PWRT による遅延時間は、INTRC のクロックによって変化する他、温度とプロセスばらつきの影響も受けるため、チップごとに異なります。詳細は、DC パラメータ 33 を参照してください。

### 5.6.1 タイムアウト シーケンス

POR パルスがクリアになると、PWRT タイムアウトが開始します。タイムアウトの合計時間は、PWRT の状態によって異なります。図 5-3、図 5-4、図 5-5、図 5-6 に、パワーアップ時のタイムアウト シーケンスを示します。

タイムアウトは POR パルスを起点に発生するため、MCLR を長時間 Low に維持すると、PWRT が先にタイムアウトします。その後で MCLR を High にすると、ただちに実行が開始します (図 5-5)。これは、テスト時または並列動作している複数の PIC18FXXJ6X の同期を取る場合に便利です。

図 5-3: パワーアップ後のタイムアウト シーケンス (MCLR を VDD に接続、VDD 立ち上がり時間 < TPWRT)

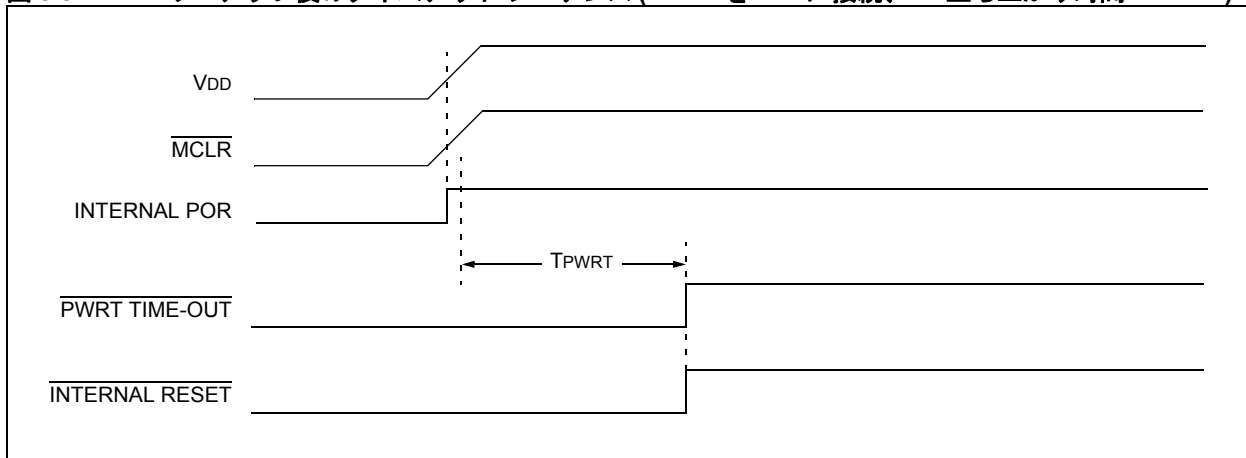
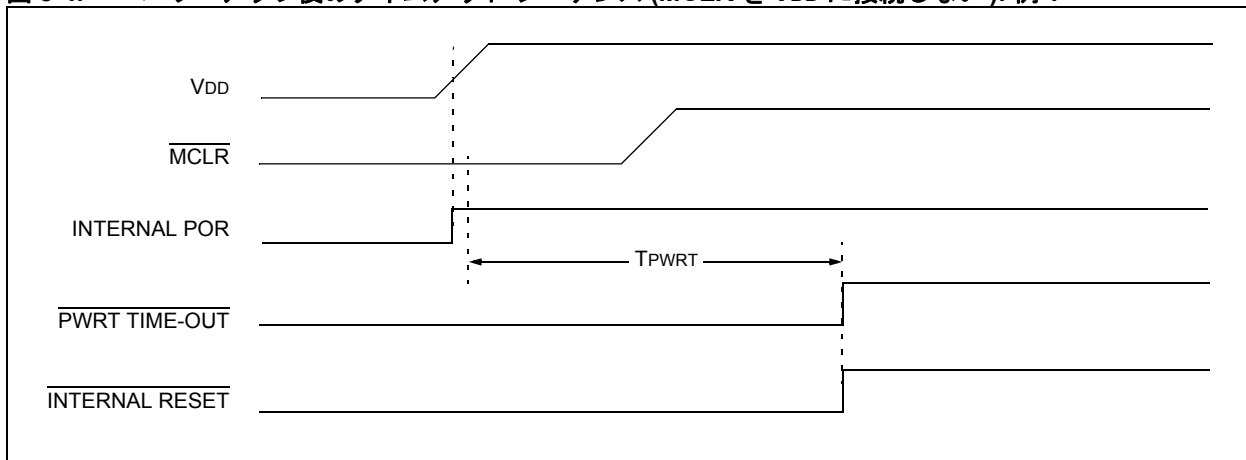


図 5-4: パワーアップ後のタイムアウト シーケンス (MCLR を VDD に接続しない): 例 1



# PIC18F97J60 ファミリ

図 5-5: パワーアップ後のタイムアウトシーケンス ( $\overline{\text{MCLR}}$  を  $\text{VDD}$  に接続しない): 例 2

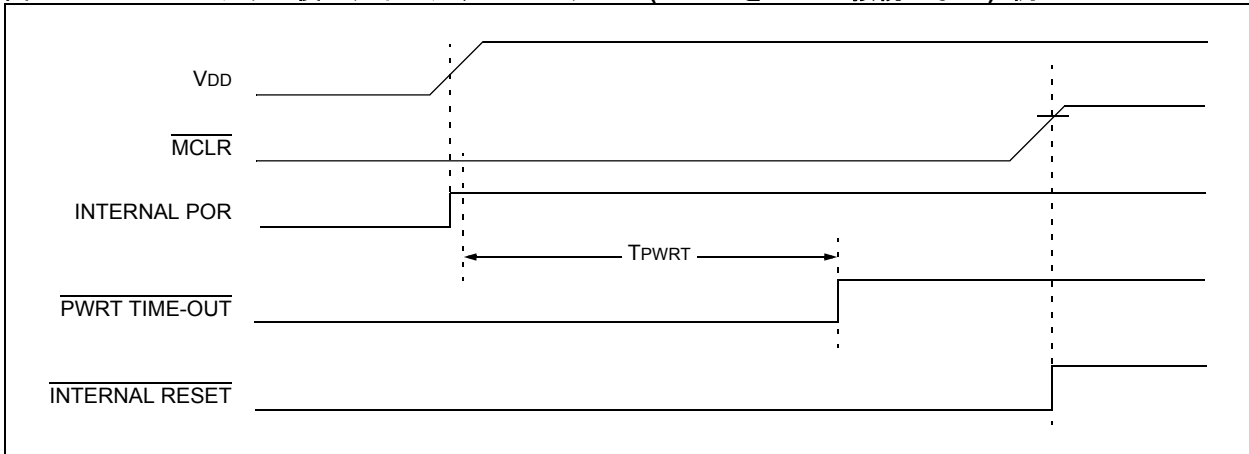
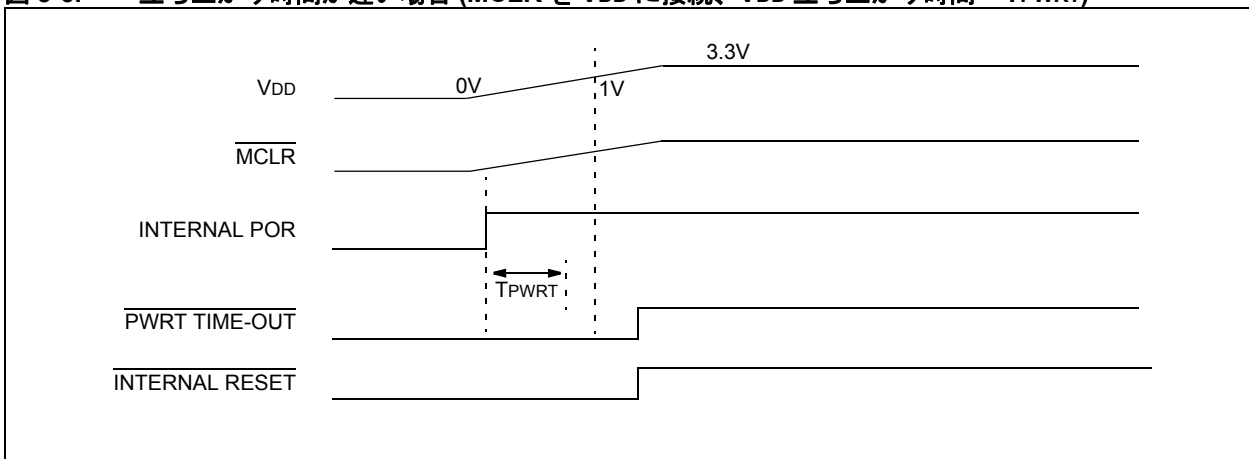


図 5-6: 立ち上がり時間が遅い場合 ( $\overline{\text{MCLR}}$  を  $\text{VDD}$  に接続、 $\text{VDD}$  立ち上がり時間  $>$   $T_{PWRT}$ )



# PIC18F97J60 ファミリ

## 5.7 レジスタのリセット状態

ほとんどのレジスタは、リセットによって値が変化しません。そのようなレジスタは POR 時の状態が不明であり、その他のリセットによって状態が変化する事はありません。ただし、発生したリセットの種類に応じて「リセット状態」に戻るレジスタもあります。

WDT による復帰は通常動作の再開と見なされるため、ほとんどのレジスタが影響を受けません。RCON レジスタのステータスビット ( $\overline{\text{CM}}$ 、 $\overline{\text{RI}}$ 、 $\overline{\text{TO}}$ 、 $\overline{\text{PD}}$ 、 $\overline{\text{POR}}$ 、

$\overline{\text{BOR}}$ ) がリセットによってセットされるかクリアされるかは、表 5-1 に示すようにリセットの種類によって異なります。これらのビットを使うと、リセットの種類をソフトウェアで判定できます。

表 5-2 に、全ての特殊機能レジスタのリセット状態を示します。この表では、「パワーオン リセット / ブラウンアウト リセット時」、「マスタクリア / WDT リセット時」、「WDT による復帰時」等に分けてリセット条件を示します。

表 5-1: RCON レジスタのステータスビットの意味と初期化条件

条件	プログラムカウンタ <sup>(1)</sup>	RCON レジスタ						STKPTR レジスタ	
		$\overline{\text{CM}}$	$\overline{\text{RI}}$	$\overline{\text{TO}}$	$\overline{\text{PD}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	STKFUL	STKUNF
パワーオン リセット	0000h	1	1	1	1	0	0	0	0
RESET 命令	0000h	u	0	u	u	u	u	u	u
ブラウンアウト リセット	0000h	1	1	1	1	u	0	u	u
コンフィグレーション不一致リセット	0000h	0	u	u	u	u	u	u	u
電力管理ランモード時の $\overline{\text{MCLR}}$	0000h	u	u	1	u	u	u	u	u
電力管理アイドル/スリープ時の $\overline{\text{MCLR}}$	0000h	u	u	1	0	u	u	u	u
フルパワー実行時の $\overline{\text{MCLR}}$	0000h	u	u	u	u	u	u	u	u
スタックフル リセット (STVREN = 1)	0000h	u	u	u	u	u	u	1	u
スタック アンダーフロー リセット (STVREN = 1)	0000h	u	u	u	u	u	u	u	1
スタック アンダーフロー エラー (STVREN = 0、実際にはリセットでない)	0000h	u	u	u	u	u	u	u	1
フルパワーまたは電力管理ランモード時の WDT タイムアウト	0000h	u	u	0	u	u	u	u	u
電力管理アイドル/スリープ時の WDT タイムアウト	PC + 2	u	u	0	0	u	u	u	u
割り込みによる電力管理モードの終了時	PC + 2	u	u	u	0	u	u	u	u

凡例: u = 不変

Note 1: 割り込みによって復帰し、GIEH または GIEL ビットがセットされている場合、PC には割り込みベクタ (0008h または 0018h) が読み込まれます。

# PIC18F97J60 ファミリ

表 5-2: 全レジスタの初期化条件

レジスタ	該当するデバイス			パワーオンリセット、 ブラウンアウトリセット	MCLR リセット、 WDT リセット、 RESET 命令、 スタックリセット、 CM リセット	WDT による復帰、 割り込みによる復帰
TOSU	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---0 0000	---0 0000	---0 uuuu <sup>(1)</sup>
TOSH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu <sup>(1)</sup>
TOSL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu <sup>(1)</sup>
STKPTR	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	00-0 0000	uu-0 0000	uu-u uuuu <sup>(1)</sup>
PCLATU	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---0 0000	---0 0000	---u uuuu
PCLATH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
PCL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	PC + 2 <sup>(2)</sup>
TBLPTRU	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	--00 0000	--00 0000	--uu uuuu
TBLPTRH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
TBLPTRL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
TABLAT	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
PRODH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
PRODL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
INTCON	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 000x	0000 000u	uuuu uuuu <sup>(3)</sup>
INTCON2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	1111 1111	1111 1111	uuuu uuuu <sup>(3)</sup>
INTCON3	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	1100 0000	1100 0000	uuuu uuuu <sup>(3)</sup>
INDF0	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
POSTINC0	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
POSTDEC0	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
PREINC0	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
PLUSW0	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
FSR0H	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---- xxxx	---- uuuu	---- uuuu
FSR0L	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
WREG	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF1	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
POSTINC1	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
POSTDEC1	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
PREINC1	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
PLUSW1	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
FSR1H	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---- xxxx	---- uuuu	---- uuuu
FSR1L	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
BSR	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---- 0000	---- 0000	---- uuuu
INDF2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
POSTINC2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
POSTDEC2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
PREINC2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
PLUSW2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
FSR2H	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---- xxxx	---- uuuu	---- uuuu
FSR2L	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu

凡例: u = 不変、x = 未知、- = 未実装、「0」として読み出し、q = 条件による  
網掛けの部分は、本デバイスにその条件が適用されない事を示します。

- Note 1:** 割り込みによって復帰し、GIEL または GIEH ビットがセットされている場合、TOSU、TOSH、TOSL には現在の PC の値が読み込まれます。STKPTR の参照先は、ハードウェア スタックの次のアドレスに変更されます。
- 2:** 割り込みによって復帰し、GIEL または GIEH ビットがセットされている場合、PC には割り込みベクタ (0008h または 0018h) が読み込まれます。
- 3:** INTCONx または PIRx レジスタの 1 つまたは複数のビットが影響を受けます (復帰動作を実行するため)。
- 4:** 各条件下でのリセット値は、表 5-1 を参照してください。

# PIC18F97J60 ファミリ

表 5-2: 全レジスタの初期化条件 ( 続き )

レジスタ	該当するデバイス			パワーオンリセット、 ブラウンアウトリセット	MCLR リセット、 WDT リセット、 RESET 命令、 スタックリセット、 CM リセット	WDT による復帰、 割り込みによる復帰
STATUS	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---x xxxx	---u uuuu	---u uuuu
TMR0H	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
TMR0L	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
T0CON	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	1111 1111	1111 1111	uuuu uuuu
OSCCON	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0--- q-00	0--- q-00	u--- q-uu
ECON1	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 00--	0000 00--	uuuu uu--
WDTCON	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---- --0	---- --0	---- --u
RCON <sup>(4)</sup>	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0-q1 1100	0-uq qquu	u-uu qquu
TMR1H	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR1L	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1CON	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	u0uu uuuu	uuuu uuuu
TMR2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
PR2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	1111 1111	1111 1111	1111 1111
T2CON	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	-000 0000	-000 0000	-uuu uuuu
SSP1BUF	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
SSP1ADD	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
SSP1STAT	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
SSP1CON1	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
SSP1CON2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
ADRESH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADRESL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON0	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0-00 0000	0-00 0000	u-uu uuuu
ADCON1	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	--00 0000	--00 0000	--uu uuuu
ADCON2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0-00 0000	0-00 0000	u-uu uuuu
CCPR1H	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR1L	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP1CON	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
CCPR2H	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR2L	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP2CON	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
CCPR3H	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR3L	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP3CON	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
ECCP1AS	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
CVRCON	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
CMCON	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0111	0000 0111	uuuu uuuu
TMR3H	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR3L	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu

凡例: u = 不変、x = 未知、- = 未実装、「0」として読み出し、q = 条件による  
網掛けの部分は、本デバイスにその条件が適用されない事を示します。

- Note 1:** 割り込みによって復帰し、GIEL または GIEH ビットがセットされている場合、TOSU、TOSH、TOSL には現在の PC の値が読み込まれます。STKPTR の参照先は、ハードウェア スタックの次のアドレスに変更されます。
- 2:** 割り込みによって復帰し、GIEL または GIEH ビットがセットされている場合、PC には割り込みベクタ (0008h または 0018h) が読み込まれます。
- 3:** INTCONx または PIRx レジスタの 1 つまたは複数のビットが影響を受けます ( 復帰動作を実行するため )。
- 4:** 各条件下でのリセット値は、表 5-1 を参照してください。

# PIC18F97J60 ファミリ

表 5-2: 全レジスタの初期化条件 ( 続き )

レジスタ	該当するデバイス			パワーオンリセット、 ブラウンアウトリセット	MCLR リセット、 WDT リセット、 RESET 命令、 スタックリセット、 CM リセット	WDT による復帰、 割り込みによる復帰
T3CON	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	uuuu uuuu	uuuu uuuu
PSPCON	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 ----	0000 ----	uuuu ----
SPBRG1	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
RCREG1	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
TXREG1	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXSTA1	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0010	0000 0010	uuuu uuuu
RCSTA1	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 000x	0000 000x	uuuu uuuu
EECON2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---- ----	---- ----	---- ----
EECON1	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---0 x00-	---0 x00-	---u uuu-
IPR3	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	1111 1111	1111 1111	uuuu uuuu
PIR3	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu <sup>(3)</sup>
PIE3	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
IPR2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	1111 1-11	1111 1-11	uuuu u-uu
PIR2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0-00	0000 0-00	uuuu u-uu <sup>(3)</sup>
PIE2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0-00	0000 0-00	uuuu u-uu
IPR1	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	1111 1111	1111 1111	uuuu uuuu
PIR1	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu <sup>(3)</sup>
PIE1	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
MEMCON	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0-00 --00	0-00 --00	u-uu --uu
OSCTUNE	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 ----	0000 ----	uuuu ----
TRISJ	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	--11 ----	--11 ----	--uu ----
	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	1111 1111	1111 1111	uuuu uuuu
TRISH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	1111 1111	1111 1111	uuuu uuuu
TRISG	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---1 ----	---1 ----	---u ----
	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---1 1111	---1 1111	---u uuuu
	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	1111 1111	1111 1111	uuuu uuuu
TRISF	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	1111 111-	1111 111-	uuuu uuu-
	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	1111 1111	1111 1111	uuuu uuuu
TRISE	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	--11 1111	--11 1111	--uu uuuu
	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	1111 1111	1111 1111	uuuu uuuu
TRISD	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---- -111	---- -111	---- -uuu
	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	1111 1111	1111 1111	uuuu uuuu
TRISC	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	1111 1111	1111 1111	uuuu uuuu
TRISB	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	1111 1111	1111 1111	uuuu uuuu
TRISA	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	--11 1111	--11 1111	--uu uuuu
LATJ	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	--xx ----	--uu ----	--uu ----
	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu

凡例: u = 不変、x = 未知、- = 未実装、「0」として読み出し、q = 条件による  
網掛けの部分は、本デバイスにその条件が適用されない事を示します。

- Note 1:** 割り込みによって復帰し、GIEL または GIEH ビットがセットされている場合、TOSU、TOSH、TOSL には現在の PC の値が読み込まれます。STKPTR の参照先は、ハードウェア スタックの次のアドレスに変更されます。
- 2:** 割り込みによって復帰し、GIEL または GIEH ビットがセットされている場合、PC には割り込みベクタ (0008h または 0018h) が読み込まれます。
- 3:** INTCONx または PIRx レジスタの 1 つまたは複数のビットが影響を受けます (復帰動作を実行するため)。
- 4:** 各条件下でのリセット値は、表 5-1 を参照してください。

# PIC18F97J60 ファミリ

表 5-2: 全レジスタの初期化条件 (続き)

レジスタ	該当するデバイス			パワーオンリセット、 ブラウンアウトリセット	MCLR リセット、 WDT リセット、 RESET 命令、 スタックリセット、 CM リセット	WDT による復帰、 割り込みによる復帰
LATG	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---x ----	---u ----	---u ----
	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---x xxxx	---u uuuu	---u uuuu
	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATF	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxx-	uuuu uu-	uuuu uu-
	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATE	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	--xx xxxx	--uu uuuu	--uu uuuu
	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATD	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---- -xxx	---- -uuu	---- -uuu
	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATC	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATB	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATA	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	00xx xxxx	00uu uuuu	uuuu uuuu
PORTJ	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	--xx ----	--uu ----	--uu ----
	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTG	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---x ----	---u ----	---u ----
	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---x xxxx	---u uuuu	---u uuuu
	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	111x xxxx	111u uuuu	uuuu uuuu
PORTF	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	x000 000-	x000 000-	uuuu uu-
	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	x000 000-	x000 000-	uuuu uu-
PORTE	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	--xx xxxx	--uu uuuu	--uu uuuu
	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTD	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---- -xxx	---- -uuu	---- -uuu
	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTC	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTB	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0-0x 0000	0-0u 0000	u-uu uuuu
SPBRGH1	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
BAUDCON1	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0100 0-00	0100 0-00	uuuu u-uu
SPBRGH2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
BAUDCON2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0100 0-00	0100 0-00	uuuu u-uu
ERDPTH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---0 1010	---0 1010	---u uuuu
ERDPTL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	1111 0101	1111 0101	uuuu uuuu
ECCP1DEL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
TMR4	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
PR4	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	1111 1111	1111 1111	1111 1111
T4CON	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	-000 0000	-000 0000	-uuu uuuu
CCPR4H	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR4L	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu

凡例: u = 不変、x = 未知、- = 未実装、「0」として読み出し、q = 条件による網掛けの部分は、本デバイスにその条件が適用されない事を示します。

- Note 1:** 割り込みによって復帰し、GIEL または GIEH ビットがセットされている場合、TOSU、TOSH、TOSL には現在の PC の値が読み込まれます。STKPTR の参照先は、ハードウェア スタックの次のアドレスに変更されます。
- 2:** 割り込みによって復帰し、GIEL または GIEH ビットがセットされている場合、PC には割り込みベクタ (0008h または 0018h) が読み込まれます。
- 3:** INTCONx または PIRx レジスタの 1 つまたは複数のビットが影響を受けます (復帰動作を実行するため)。
- 4:** 各条件下でのリセット値は、表 5-1 を参照してください。



# PIC18F97J60 ファミリ

表 5-2: 全レジスタの初期化条件 ( 続き )

レジスタ	該当するデバイス			パワーオンリセット、 ブラウンアウトリセット	MCLR リセット、 WDT リセット、 RESET 命令、 スタックリセット、 CM リセット	WDT による復帰、 割り込みによる復帰
CCP4CON	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	--00 0000	--00 0000	--uu uuuu
CCPR5H	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR5L	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP5CON	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	--00 0000	--00 0000	--uu uuuu
SPBRG2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
RCREG2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
TXREG2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
TXSTA2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0010	0000 0010	uuuu uuuu
RCSTA2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 000x	0000 000x	uuuu uuuu
ECCP3AS	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
ECCP3DEL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
ECCP2AS	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
ECCP2DEL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
SSP2BUF	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
SSP2ADD	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
SSP2STAT	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
SSP2CON1	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
SSP2CON2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
EDATA	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
EIR	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	-000 0-00	-000 0-00	-uuu u-uu
ECON2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	100- ----	100- ----	uuu- ----
ESTAT	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	-0-0 -000	-0-0 -000	-u-u -uuu
EIE	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	-000 0-00	-000 0-00	-uuu u-uu
EDMACSH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
EDMACSL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
EDMADSTH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---0 0000	---0 0000	---u uuuu
EDMADSTL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
EDMANDH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---0 0000	---0 0000	---u uuuu
EDMANDL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
EDMASTH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---0 0000	---0 0000	---u uuuu
EDMASTL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
ERXWRPTH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---0 0000	---0 0000	---u uuuu
ERXWRPTL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
ERXRDPH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---0 0101	---0 0101	---u uuuu
ERXRDPH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	1111 1010	1111 1010	uuuu uuuu
ERXNDH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---1 1111	---1 1111	---u uuuu
ERXNDL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	1111 1111	1111 1111	uuuu uuuu
ERXSTH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---0 0101	---0 0101	---u uuuu

凡例: u = 不変、x = 未知、- = 未実装、「0」として読み出し、q = 条件による  
網掛けの部分は、本デバイスにその条件が適用されない事を示します。

- Note 1:** 割り込みによって復帰し、GIEL または GIEH ビットがセットされている場合、TOSU、TOSH、TOSL には現在の PC の値が読み込まれます。STKPTR の参照先は、ハードウェア スタックの次のアドレスに変更されます。
- 2:** 割り込みによって復帰し、GIEL または GIEH ビットがセットされている場合、PC には割り込みベクタ (0008h または 0018h) が読み込まれます。
- 3:** INTCONx または PIRx レジスタの 1 つまたは複数のビットが影響を受けます ( 復帰動作を実行するため )。
- 4:** 各条件下でのリセット値は、表 5-1 を参照してください。

# PIC18F97J60 ファミリ

表 5-2: 全レジスタの初期化条件 ( 続き )

レジスタ	該当するデバイス			パワーオンリセット、 ブラウンアウトリセット	MCLR リセット、 WDT リセット、 RESET 命令、 スタックリセット、 CM リセット	WDT による復帰、 割り込みによる復帰
ERXSTL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	1111 1010	1111 1010	uuuu uuuu
ETXNDH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---0 0000	---0 0000	---u uuuu
ETXNDL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
ETXSTH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---0 0000	---0 0000	---u uuuu
ETXSTL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
EWRPTH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---0 0000	---0 0000	---u uuuu
EWRPTL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
EPKTCNT	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
ERXFCON	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	1010 0001	1010 0001	uuuu uuuu
EPMOH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---0 0000	---0 0000	---u uuuu
EPMOL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
EPMCSH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
EPMCSL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
EPMM7	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
EPMM6	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
EPMM5	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
EPMM4	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
EPMM3	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
EPMM2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
EPMM1	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
EPMM0	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
EHT7	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
EHT6	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
EHT5	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
EHT4	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
EHT3	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
EHT2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
EHT1	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
EHT0	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
MIRDH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
MIRDL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
MIWRH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
MIWRL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
MIREGADR	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---0 0000	---0 0000	---u uuuu
MICMD	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---- --00	---- --00	---- --uu
MAMXFLH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0110	0000 0110	uuuu uuuu
MAMXFLI	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
MAIPGH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	-000 0000	-000 0000	-uuu uuuu

凡例: u = 不変、x = 未知、- = 未実装、「0」として読み出し、q = 条件による  
網掛けの部分は、本デバイスにその条件が適用されない事を示します。

- Note 1:** 割り込みによって復帰し、GIEL または GIEH ビットがセットされている場合、TOSU、TOSH、TOSL には現在の PC の値が読み込まれます。STKPTR の参照先は、ハードウェア スタックの次のアドレスに変更されます。
- 2:** 割り込みによって復帰し、GIEL または GIEH ビットがセットされている場合、PC には割り込みベクタ (0008h または 0018h) が読み込まれます。
- 3:** INTCONx または PIRx レジスタの 1 つまたは複数のビットが影響を受けます ( 復帰動作を実行するため )。
- 4:** 各条件下でのリセット値は、表 5-1 を参照してください。

# PIC18F97J60 ファミリ

表 5-2: 全レジスタの初期化条件 ( 続き )

レジスタ	該当するデバイス			パワーオンリセット、 ブラウンアウトリセット	MCLR リセット、 WDT リセット、 RESET 命令、 スタックリセット、 CM リセット	WDT による復帰、 割り込みによる復帰
MAIPGL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	-000 0000	-000 0000	-uuu uuuu
MABBIPG	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	-000 0000	-000 0000	-uuu uuuu
MACON4	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	-000 --00	-000 --00	-uuu --uu
MACON3	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
MACON1	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---0 0000	---0 0000	---u uuuu
EPAUSH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0001 0000	0001 0000	000u uuuu
EPAUSL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
EFLOCON	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---- -000	---- -000	---- -uuu
MISTAT	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---- 0000	---- 0000	---- uuuu
MAADR2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
MAADR1	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
MAADR4	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
MAADR3	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
MAADR6	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
MAADR5	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu

凡例: u = 不変、x = 未知、- = 未実装、「0」として読み出し、q = 条件による  
網掛けの部分は、本デバイスにその条件が適用されない事を示します。

- Note 1:** 割り込みによって復帰し、GIEL または GIEH ビットがセットされている場合、TOSU、TOSH、TOSL には現在の PC の値が読み込まれます。STKPTR の参照先は、ハードウェア スタックの次のアドレスに変更されます。
- 2:** 割り込みによって復帰し、GIEL または GIEH ビットがセットされている場合、PC には割り込みベクタ (0008h または 0018h) が読み込まれます。
- 3:** INTCONx または PIRx レジスタの 1 つまたは複数のビットが影響を受けます (復帰動作を実行するため)。
- 4:** 各条件下でのリセット値は、表 5-1 を参照してください。

# PIC18F97J60 ファミリ

---

NOTES:

# PIC18F97J60 ファミリ

## 6.0 メモリ構成

PIC18 マイクロコントローラは、以下の 2 種類のメモリを搭載しています。

- プログラムメモリ
- データ RAM

このデバイスはハーバードアーキテクチャを採用しており、データメモリとプログラムメモリは別々のバスを使います。このため、これら 2 つのメモリ空間へ同時にアクセスできます。

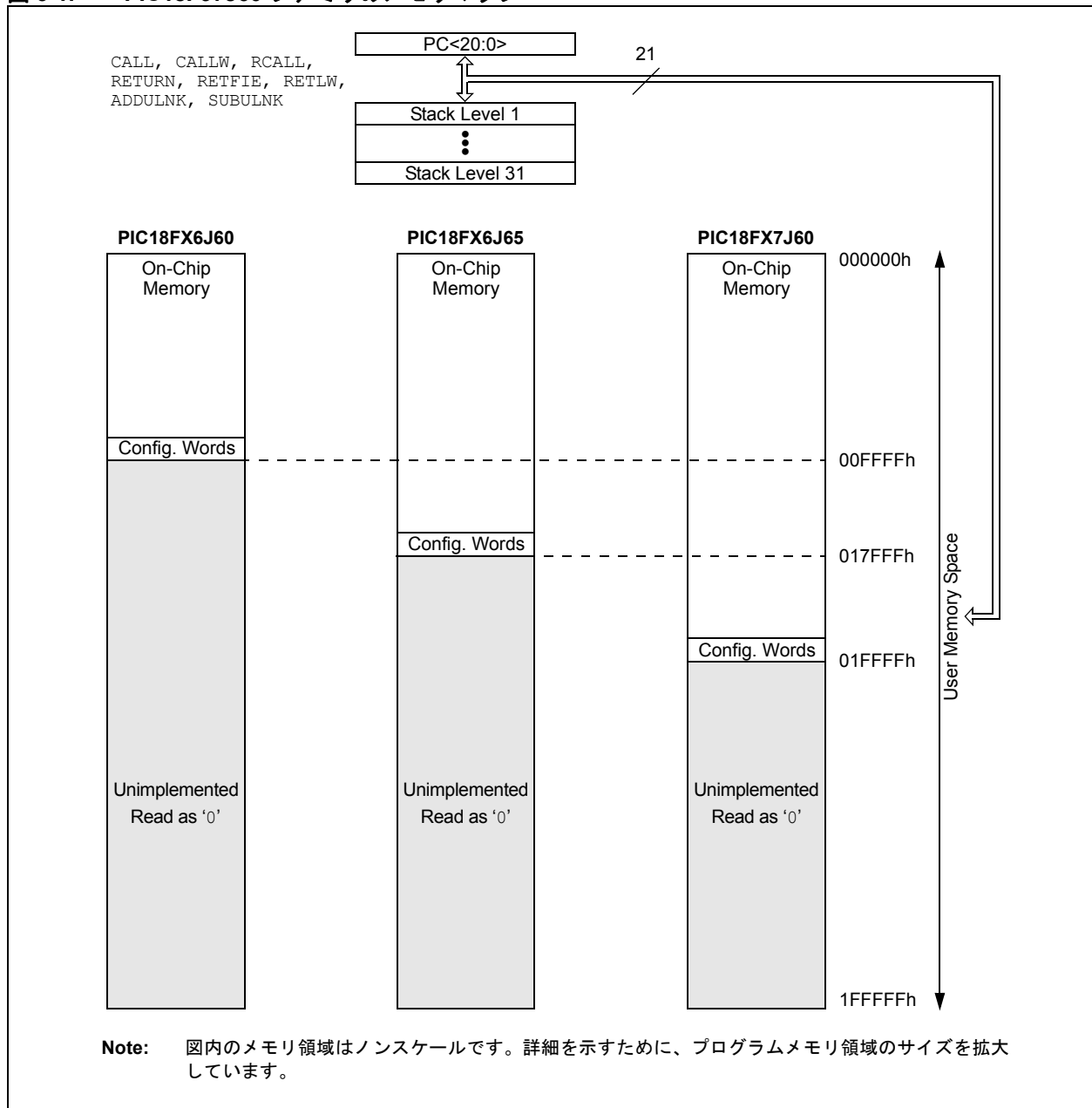
フラッシュ プログラムメモリの動作の詳細は [セクション 7.0「フラッシュ プログラムメモリ」](#) で説明します。

## 6.1 プログラムメモリの構成

PIC18 マイクロコントローラは 21 ビットのプログラムカウンタを実装し、2 MB のプログラムメモリ空間に対してアドレスを指定できます。物理的に実装されたメモリの上限と 2 MB のアドレスの間のメモリ位置にアクセスすると、常に「0」(NOP 命令)が返されます。

PIC18F97J60 ファミリの内蔵フラッシュ プログラムメモリの容量はデバイスによって異なり、64 KB (最大 32,764 個のシングルワード命令) から 128 KB (65,532 個のシングルワード命令) まで全部で 3 種類あります。図 6-1 に、これら 3 種類のデバイスのプログラムメモリのマップを示します。

図 6-1: PIC18F97J60 ファミリのメモリマップ



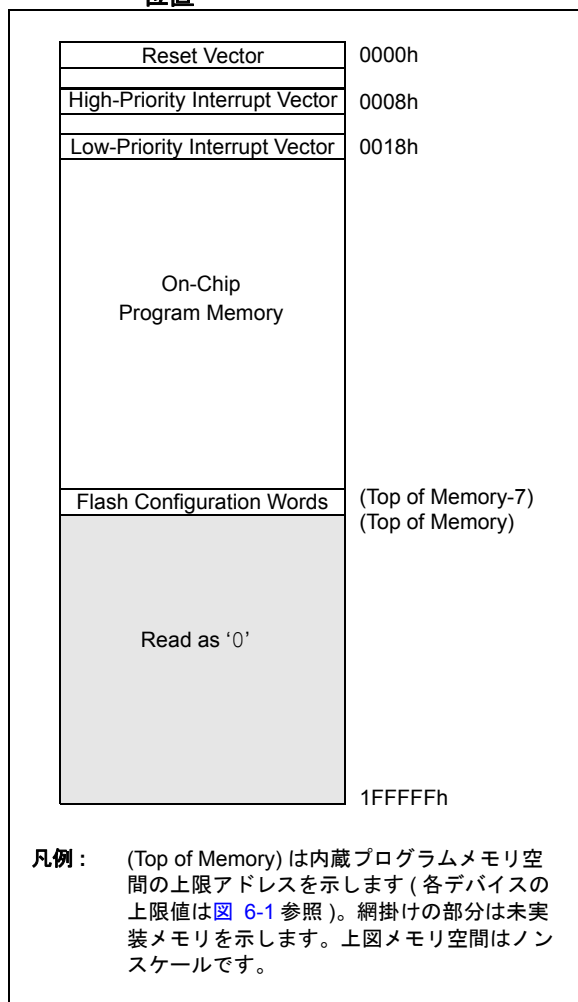
# PIC18F97J60 ファミリ

## 6.1.1 ハードメモリ ベクタ

全ての PIC18 には、プログラムメモリ空間に合計 3 つのハードコードされたリターンベクタがあります。リセットベクタ アドレスは、全てのデバイスリセットでプログラム カウンタが既定値として戻る値で、0000h にあります。

PIC18 には、高優先度と低優先度の割り込みを処理するための割り込みベクタアドレスも 2 つあります。高優先度割り込みのベクタは 0008h、低優先度割り込みのベクタは 0018h です。図 6-2 に、プログラムメモリ マップにおけるこれらのハードベクタの位置を示します。

**図 6-2: PIC18F97J60 ファミリのハードベクタとコンフィグレーションワードの位置**



**凡例:** (Top of Memory) は内蔵プログラムメモリ空間の上限アドレスを示します (各デバイスの上限値は図 6-1 参照)。網掛けの部分は未実装メモリを示します。上図メモリ空間はノンスケールです。

## 6.1.2 フラッシュ コンフィグレーションワード

PIC18F97J60 ファミリには不揮発性のコンフィグレーションメモリはなく、内蔵プログラムメモリの最上位 4 ワードがコンフィグレーション情報用に予約済みです。リセット時、このコンフィグレーション情報がコンフィグレーションレジスタにコピーされます。

コンフィグレーションワードはプログラムメモリの最上位 4 ワードに、番号の小さいものから順に配置されています (CONFIG1 の下位バイトが最下位アドレスに配置され、CONFIG4 の上位バイトが最上位アドレスに配置)。ただしこのデバイスで使うコンフィグレーションワードは CONFIG1 ~ CONFIG3 のみで、CONFIG4 は予約済みです。表 6-1 に、PIC18F97J60 ファミリのフラッシュコンフィグレーションワードの実際のアドレスを示します。図 6-2 には、フラッシュコンフィグレーションワードの位置もハードベクタと共にメモリマップとして示しています。

コンフィグレーションワードの詳細は、[セクション 25.1「コンフィグレーションビット」](#)で説明します。

**表 6-1: PIC18F97J60 ファミリのフラッシュコンフィグレーションワード**

デバイス名	プログラムメモリ (KB)	コンフィグレーションワードのアドレス
PIC18F66J60	64	FFF8h ~ FFFFh
PIC18F86J60		
PIC18F96J60		
PIC18F66J65	96	17FF8h ~ 17FFFh
PIC18F86J65		
PIC18F96J65	128	1FFF8h ~ 1FFFFh
PIC18F67J60		
PIC18F87J60		
PIC18F97J60		

# PIC18F97J60 ファミリ

## 6.1.3 PIC18F9XJ60/9XJ65 のプログラムメモリモード

このファミリの 100 ピンデバイスでは、最大で合計 2 MB までのプログラムメモリをアドレス指定できます。これには、外部メモリバスを使います。コントローラのプログラムメモリモードには以下の 2 種類があります。

- マイクロコントローラ (MC) モード
- 拡張マイクロコントローラ (EMC) モード

プログラムメモリのモードは、EMB コンフィグレーション ビット (CONFIG3L<5:4>) の設定によって決まります (レジスタ 6-1 参照)。(コンフィグレーション ビットの詳細は [セクション 25.1「コンフィグレーション ビット」](#) を参照してください)。

各プログラムメモリモードの動作は以下の通りです。

- **マイクロコントローラモード:** 内蔵フラッシュメモリのみアクセスします。内蔵メモリの上限を超えたアドレスを読み出そうとすると、全て「0」(NOP 命令)として読み出されます。

64 ピンおよび 80 ピンデバイスでは、マイクロコントローラモードしか使えません。

- **拡張マイクロコントローラモード:** 内蔵プログラムメモリと外部プログラムメモリの両方に 1 つのブロックとしてアクセスできます。デバイスからは、内蔵プログラムメモリ全体に加え、最大 2 MB のプログラムメモリ空間の範囲内で外部プログラムメモリにもアクセスできます。これら 2 つのメモリからの実行は、必要に応じて自動的に切り換わります。

外部メモリバスのアドレスバス幅も EMB コンフィグレーション ビットで設定します。これについては、[セクション 8.0「外部メモリバス」](#) で詳しく説明します。

どのモードでも、マイクロコントローラはデータ RAM に完全にアクセスできます。

[図 6-3](#) に、各プログラムメモリモード時のメモリマップを示します。[表 6-2](#) に、内蔵メモリと外部メモリへのアクセスの制約を一覧で示します。

**レジスタ 6-1: CONFIG3L: コンフィグレーション レジスタ 3 Low**

R/WO-1	R/WO-1	R/WO-1	R/WO-1	R/WO-1	U-0	U-0	U-0
WAIT <sup>(1)</sup>	BW <sup>(1)</sup>	EMB1 <sup>(1)</sup>	EMB0 <sup>(1)</sup>	EASHFT <sup>(1)</sup>	—	—	—
bit 7					bit 0		

### 凡例:

R = 読み出し可能ビット      WO = ライトワンス ビット      U = 未実装ビット、「0」として読み出し  
 -n = デバイス未プログラム時の値      「1」= ビットはセット      「0」= ビットはクリア

- bit 7      **WAIT:** 外部バスウェイト イネーブルビット <sup>(1)</sup>  
 1 = 外部メモリバスに対する操作のウェイトステートを無効にする  
 0 = 外部メモリバスに対する操作のウェイトステートを有効にし、MEMCON<5:4> で選択する
- bit 6      **BW:** データバス幅選択ビット <sup>(1)</sup>  
 1 = 16 ビット データ幅モード  
 0 = 8 ビット データ幅モード
- bit 5-4      **EMB<1:0>:** 外部メモリバス コンフィグレーション ビット <sup>(1)</sup>  
 11 = マイクロコントローラモード、外部バスを無効にする  
 10 = 拡張マイクロコントローラモード、12 ビット アドレッシングモード  
 01 = 拡張マイクロコントローラモード、16 ビット アドレッシングモード  
 00 = 拡張マイクロコントローラモード、20 ビット アドレッシングモード
- bit 3      **EASHFT:** 外部アドレスバス シフト イネーブルビット <sup>(1)</sup>  
 1 = アドレスシフトを有効にする。外部バスのアドレスが 000000h から開始するようにオフセットする  
 0 = アドレスシフトを無効にする。外部バスのアドレスをそのまま PC の値にする
- bit 2-0      **未実装:** 「0」として読み出し

**Note 1:** 100 ピンデバイスだけがこのビットを実装します。

# PIC18F97J60 ファミリ

## 6.1.4 拡張マイクロコントローラ モードと アドレスシフト

既定値では、デバイスを拡張マイクロコントローラモードに設定すると、拡張メモリ空間のレンジにあるアドレスについてはプログラムカウンタの値が直接外部アドレスバスに現れます。つまり、内蔵メモリの上限より下のアドレスの外部メモリデバイスにはアクセスできない事を意味します。

これを防ぐため、拡張マイクロコントローラモードにはアドレスを自動で変換できるアドレスシフト機能があります。このモードを選択すると、外部バスに現れるアドレスは内蔵プログラムメモリの容量分だけ下にシフトし、開始アドレスを 0000h として再マッピングされます。これにより、外部メモリデバイスのメモリ空間を完全に利用できます。

図 6-3: PIC18F97J60 ファミリの各プログラムメモリモードにおけるメモリマップ

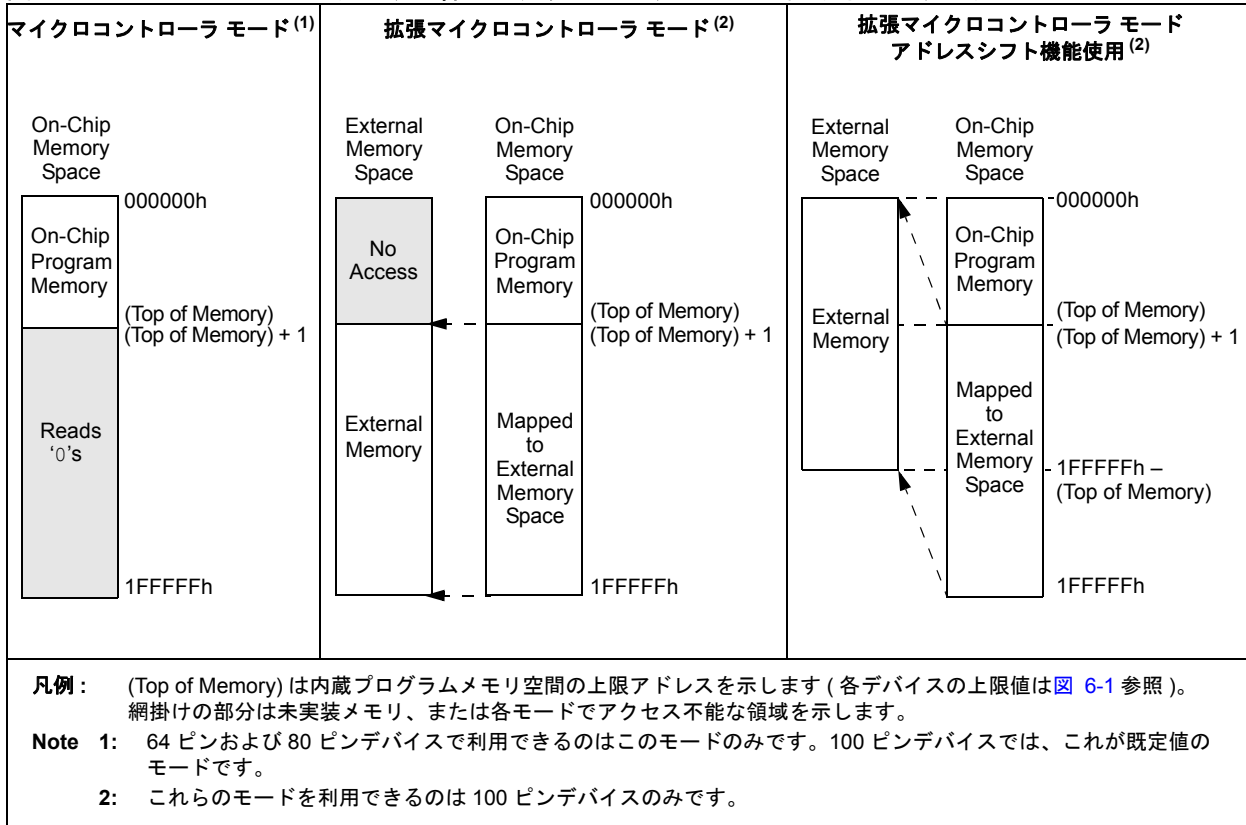


表 6-2: PIC18F9XJ60/9XJ65 の各プログラムメモリモード時のメモリアクセス

動作モード	内蔵プログラムメモリ			外部プログラムメモリ		
	命令実行	テーブル読み出し	テーブル書き込み	命令実行	テーブル読み出し	テーブル書き込み
マイクロコントローラ	可	可	可	アクセス不可	アクセス不可	アクセス不可
拡張マイクロコントローラ	可	可	可	可	可	可



## 6.1.5 プログラムカウンタ

プログラムカウンタ (PC) は、次にフェッチ / 実行する命令のアドレスを指定します。PC は 21 ビット幅で、3 つの独立した 8 ビットレジスタから構成されます。PCL レジスタと呼ばれる下位バイトは読み書き可能です。上位バイトの PCH レジスタには PC<15:8> のビットが含まれ、直接読み書きはできません。PCH レジスタは、PCLATH レジスタを介して更新します。最上位バイトは PCU と呼ばれます。このレジスタには、PC<20:16> のビットが含まれ、PCH と同様に直接読み書きすることはできません。PCU レジスタは、PCLATU レジスタを介して更新します。

PCL への書き込み動作を実行するたびに、PCLATH と PCLATU の内容がプログラムカウンタに転送されます。同様に、PCL からの読み出し動作を実行するたびに、プログラムカウンタの上位 2 バイトが PCLATH と PCLATU に転送されます。この機能は、計算によって PC をオフセットさせる場合に便利です ( [セクション 6.1.8.1 「計算型 GOTO」](#) 参照 )。

PC はバイト単位でプログラムメモリのアドレスを指定します。ワード単位の命令と PC の間で不整合が生じないように、PCL の最下位ビットは「0」に固定されています。従って、プログラムメモリ内の連続する命令に対して順にアドレスを指定していく場合、PC は 2 ずつインクリメントします。

CALL、RCALL、GOTO、プログラム分岐命令は、プログラムカウンタに直接書き込みます。これらの命令では、PCLATH と PCLATU の内容はプログラムカウンタに転送されません。

## 6.1.6 リターンアドレススタック

リターンアドレススタックは、任意の組み合わせのプログラムコールと割り込みに対して最大 31 個まで対応できます。CALL または RCALL 命令を実行するか、割り込みに対して肯定応答 (ACK) すると、スタックに PC の値がプッシュされます。RETURN、RETLW、RETFIE 命令 (および拡張命令セットが有効な場合は ADDULNK、SUBULNK 命令) を実行すると、PC の値がスタックからポップされます。PCLATU と PCLATH は、RETURN 系命令または CALL 系命令のいずれからも影響を受けません。

スタックは、31 ワード x 21 ビットの RAM、5 ビットのスタックポインタ STKPTR によって動作します。スタック空間は、プログラム空間とデータ空間から独立しています。スタックポインタは読み書き可能です。また、スタックの最上位 TOS (Top-of-Stack) に格納されているアドレスは、TOS 特殊機能レジスタを介して読み書きできます。スタックのデータは、これらのレジスタによってプッシュまたはポップできます。

CALL 系命令を実行すると、スタックへのプッシュが行われます。まずスタックポインタがインクリメントした後、スタックポインタの指し示す位置に PC (既に CALL の次の命令を指し示しています) の内容が書き込まれます。RETURN 系命令を実行すると、スタックからのポップが行われます。STKPTR が指し示す位置の内容が PC に転送されてから、スタックポインタがデクリメントします。

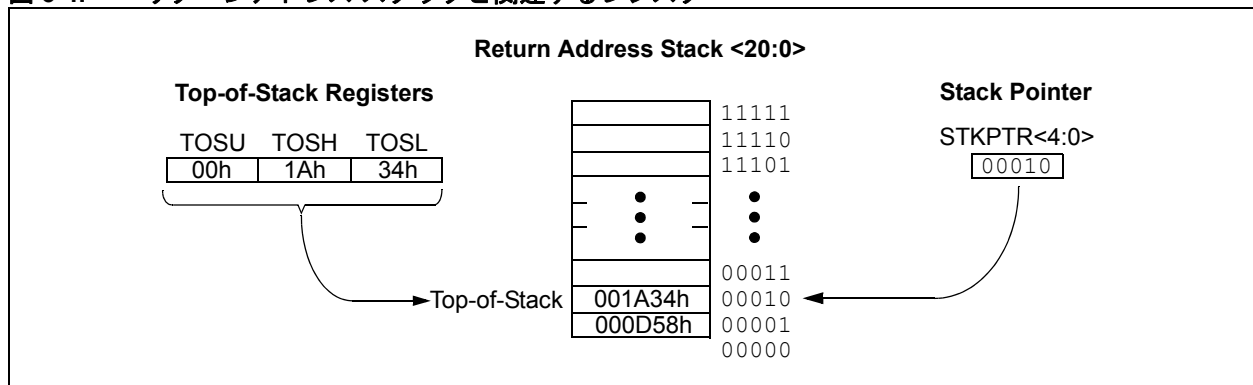
全てのリセット動作は、スタックポインタを「00000」に初期化します。RAM 上にスタックポインタの値「00000」に対応する位置は存在しません。これは単なるリセット値です。スタックフルまたはオーバーフロー / アンダーフローの状態は、ステータスビットによって示します。

### 6.1.6.1 Top-of-Stack へのアクセス

リターンアドレススタックの読み書きは最上位の Top-of-Stack (TOS) に対してのみ実行できます。STKPTR レジスタが指し示すスタック位置の内容は、3 つのレジスタセット TOSU:TOSH:TOSL に保持されます ( [図 6-4](#) )。これを利用して、ユーザが必要に応じてソフトウェアスタックを実装することができます。CALL、RCALL、割り込み、(および拡張命令セットが有効な場合は ADDULNK、SUBULNK 命令) の後、ソフトウェアで TOSU:TOSH:TOSL レジスタを読み出すと、プッシュされた値を取得できます。取得した値は、ユーザ定義のソフトウェアスタックに格納できます。リターン時には、ソフトウェアでこれらの値を TOSU:TOSH:TOSL に返し、リターンを実行します。

スタックにアクセスする場合はグローバル割り込みイネーブルビットを無効にして、意図しないスタック破損を防ぐ必要があります。

図 6-4: リターンアドレススタックと関連するレジスタ



# PIC18F97J60 ファミリ

## 6.1.6.2 リターンスタックポインタ (STKPTR)

STKPTR レジスタ (レジスタ 6-2) は、スタックポインタの値、STKFUL (スタックフル) ステータスビット、STKUNF (スタック アンダーフロー) ステータスビットを含みます。スタックポインタが取る事のできる値は 0 ~ 31 です。スタックポインタは、値をスタックにプッシュする前にインクリメントされ、スタックからポップした後にデクリメントされます。リセットするとスタックポインタの値はゼロになります。スタックポインタの値は、ユーザによる読み書きが可能です。この機能は、リアルタイム オペレーティングシステム (RTOS) におけるリターンスタックの管理に役立ちます。

スタックに PC が 31 回プッシュされると (その間、スタックから値を一切ポップしなかった場合)、STKFUL ビットがセットされます。STKFUL ビットは、ソフトウェアまたは POR によってクリアされます。

スタックフルが発生した場合の動作は、コンフィグレーションビット STVREN (スタック オーバーフロー リセット イネーブル) の状態によって決まります (デバイス コンフィグレーション ビットの詳細は [セクション 25.1「コンフィグレーション ビット」](#) を参照してください)。STVREN がセットされている場合 (既定値)、31 回目のプッシュが実行されると、(PC + 2) の値がスタックに格納され、STKFUL ビットのセット後、デバイスがリセットされます。STKFUL ビットはセットされたまま、スタックポインタはゼロにリセットされます。

一方、STVREN がクリアされている場合、31 回目のプッシュによって STKFUL ビットがセットされ、スタックポインタは 31 にインクリメントされます。その後は、プッシュ動作を実行しても 31 回目にプッシュされた値が上書きされる事はなく、STKPTR の値も 31 に保たれます。

スタックが完全に空になるまでポップすると、次のポップでは PC にゼロが返され、STKUNF ビットがセットされます。スタックポインタの値はゼロのままです。STKUNF ビットは、ソフトウェアによってクリアされるか POR が発生するまで、セットされた状態を維持します。

**Note:** アンダーフローが発生して PC に値ゼロが返されると、プログラムはリセットベクタに分岐します。分岐先でスタックの状態を確認して適切な措置を講ずる事ができます。この動作は特殊機能レジスタ (SFR) の内容には影響を与えない事から、リセットとは異なります。

## 6.1.6.3 PUSH および POP 命令

Top-of-Stack (TOS) は読み書き可能であるため、通常のプログラム実行を妨げる事なく、スタックとの間で値をプッシュ / ポップできる機能があると便利です。PIC18 の命令セットには、ソフトウェアによるこのような TOS 操作を可能とする 2 つの命令、PUSH と POP があります。これらの命令によって TOSU、TOSH、TOSL を変更して、スタックにデータまたはリターンアドレスを格納できます。

PUSH 命令は現在の PC の値をスタックにプッシュします。つまり、スタックポインタをインクリメントして、現在の PC の値をスタックに格納します。

POP 命令はスタックポインタをデクリメントして、現在の TOS の値を破棄します。そして、破棄された値より前にスタックにプッシュされた値が、新たな TOS の値になります。

レジスタ 6-2: STKPTR: スタックポインタ レジスタ

R/C-0	R/C-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
STKFUL <sup>(1)</sup>	STKUNF <sup>(1)</sup>	—	SP4	SP3	SP2	SP1	SP0
bit 7							bit 0

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

- bit 7      **STKFUL:** スタックフル フラグビット<sup>(1)</sup>  
1 = スタックがフルまたはオーバーフローした  
0 = スタックはフルではなく、オーバーフローもしていない
- bit 6      **STKUNF:** スタック アンダーフロー フラグビット<sup>(1)</sup>  
1 = スタック アンダーフローが発生した  
0 = スタック アンダーフローは発生していない
- bit 5      **未実装:** 「0」として読み出し
- bit 4-0      **SP<4:0>:** スタックポインタの位置を表すビット

**Note 1:** bit 7 と bit 6 は、ユーザ ソフトウェアまたは POR によってクリアされます。

## 6.1.6.4 スタックフルとスタック アンダーフローによるリセット

スタック オーバーフローおよびスタック アンダーフロー条件によるデバイスのリセットは、コンフィグレーションレジスタ CONFIG1L の STVREN ビットをセットすると有効になります。STVREN がセットされている場合、スタックフルまたはアンダーフロー条件によってそれぞれ STKFUL または STKUNF ビットがセットされたのち、デバイスがリセットされます。STVREN がクリアされている場合、スタックフルまたはアンダーフロー条件によってそれぞれ STKFUL または STKUNF ビットがセットされますが、デバイスはリセットされません。STKFUL および STKUNF ビットは、ユーザ ソフトウェアまたはパワーオンリセットによってクリアされます。

## 6.1.7 高速レジスタスタック

STATUS、WREG、BSR の各レジスタは高速レジスタスタック (FRS) を装備し、割り込み処理後の「高速リターン」機能を提供します。このスタックは 1 レベルの深さしかなく、読み出しも書き込みもできません。このスタックは、プロセッサが割り込みベクタに分岐すると対応するレジスタの現在の値を格納します (全ての割り込み要因でスタックレジスタに値がプッシュされます)。その後、RETFIE FAST 命令によって割り込み処理から戻る際に、スタックレジスタの値がワーキングレジスタに返されます。

低優先度と高優先度の割り込みを両方とも有効にしている場合、低優先度の割り込みからのリターンに対しては、スタックレジスタが意図した通りに動作しない事があります。これは、低優先度の割り込み処理中に高優先度の割り込みが発生した場合、低優先度割り込みがスタックレジスタに保存した値が上書きされてしまうためです。ユーザはこのような場合に備えて、低優先度割り込みが発生したら主要レジスタの値をソフトウェアで保存しておく必要があります。

割り込みの優先度を使わない場合、全ての割り込みからのリターンに高速レジスタスタックを使えます。また、割り込みを一切使わない場合でも、サブルーチンコールの終わりに高速レジスタスタックを使って STATUS、WREG、BSR 各レジスタの値を復元できます。サブルーチンコールで高速レジスタスタックを使うには、CALL label, FAST 命令を実行して STATUS、WREG、BSR レジスタの値を高速レジスタスタックに保存する必要があります。その後、RETURN, FAST 命令を実行して、これらのレジスタの値を高速レジスタスタックから復元します。

例 6-1 に、サブルーチンのコールおよびリターン時に高速レジスタスタックを使うサンプルコードを示します。

### 例 6-1: 高速レジスタスタックを使うサンプルコード

```
CALL SUB1, FAST ;STATUS, WREG, BSR
                  ;SAVED IN FAST REGISTER
                  ;STACK
.
.
SUB1 .
.
RETURN FAST ;RESTORE VALUES SAVED
          ;IN FAST REGISTER STACK
```

## 6.1.8 プログラムメモリ内のルックアップテーブル

プログラミングにおいて、プログラムメモリ内にデータ構造、つまりルックアップテーブルの作成が必要となる場合があります。PIC18 では、以下の 2 つの方法でルックアップテーブルを実装できます。

- 計算型 GOTO
- テーブル読み出し

### 6.1.8.1 計算型 GOTO

計算型 GOTO は、プログラムカウンタにオフセット値を加算する事で実現します。例 6-2 に、この例を示します。

ルックアップテーブルは ADDWF PCL 命令と、一連の RETLW nn 命令を使って作成できます。W レジスタ (WREG) には、そのテーブルルーチン呼び出す前に、テーブル内のオフセットを格納しておきます。呼び出されるルーチンの最初の命令は ADDWF PCL です。その次に実行される命令は、RETLW nn 命令の中の 1 つで、呼び出し元の関数に値「nn」を返します。

WREG 内のオフセット値は、プログラムカウンタを進めるバイト数を表し、2 の倍数である必要があります (LSb = 0)。

この方法では、1 命令分の位置に 1 バイトのデータしか保存できず、リターンアドレススタックに空きが必要です。

### 例 6-2: オフセット値を使う計算型 GOTO

```
MOVWF OFFSET, W
CALL TABLE
ORG nn00h
TABLE ADDWF PCL
      RETLW nnh
      RETLW nnh
      RETLW nnh
      .
      .
      .
```

### 6.1.8.2 テーブル読み出し

プログラムメモリ内にデータを保存するには、1 命令分の位置に 2 バイトのデータを保存できるこちらの方法がより適しています。

ルックアップテーブルのデータは、プログラミング時に 1 プログラムワード当たり 2 バイトを保存できます。テーブルポインタ (TBLPTR) でバイトアドレスを指定し、テーブルラッチ (TABLAT) にプログラムメモリから読み出したデータを格納します。データはプログラムメモリから一度に 1 バイトずつ転送されます。

テーブル読み出し動作の詳細は、[セクション 7.1「テーブル読み書き」](#)で説明します。

# PIC18F97J60 ファミリ

## 6.2 PIC18 の命令サイクル

### 6.2.1 クロック方式

マイクロコントローラへのクロック入力は、それが内部クロック源と外部クロック源のいずれから供給されたものであれ、内部で 4 分周されて 4 つのオーバーラップのない直角位相クロック (Q1、Q2、Q3、Q4) が生成されます。プログラムカウンタは、内部で Q1 ごとにインクリメントします。命令は、Q4 でプログラムメモリからフェッチされ、命令レジスタ (IR) にラッチされます。次の Q1 から Q4 の期間に命令がデコードされ実行されます。図 6-5 にクロックと命令実行のフローを示します。

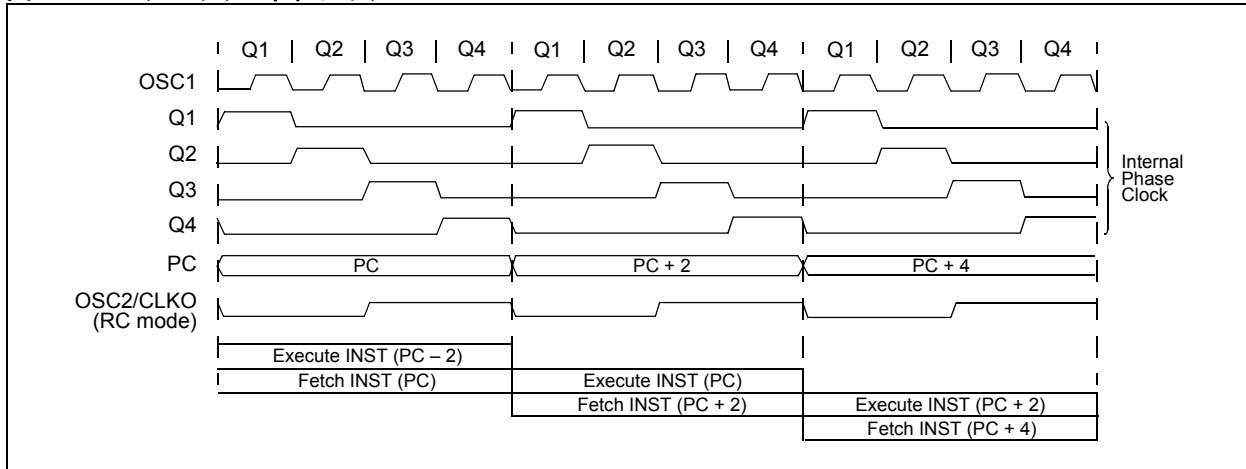
### 6.2.2 命令のフロー/パイプライン処理

「命令サイクル」は Q1 ~ Q4 の 4 つの Q サイクルで構成されます。命令のフェッチと実行には、フェッチに 1 命令サイクル、デコードと実行にもう 1 命令サイクルを使うパイプライン方式が採用されています。パイプライン方式により、各命令は実質的には 1 サイクルで処理されます。プログラムカウンタの変更を伴う命令の場合 (GOTO 等)、命令の完了には 2 サイクル必要です (例 6-3)。

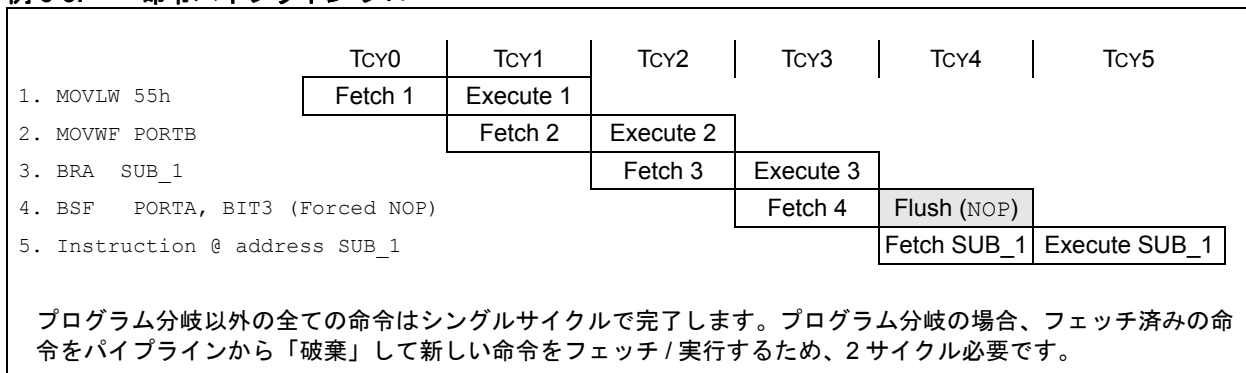
フェッチサイクルは、Q1 でプログラムカウンタ (PC) がインクリメントすると開始します。

実行サイクルでは、フェッチされた命令がサイクル Q1 で ROM ラッチにラッチされます。続くサイクル Q2、Q3、Q4 で命令がデコードされ実行されます。データメモリからの読み出し (オペランドの読み出し) は Q2 で、データメモリへの書き込み (結果格納先への書き込み) は Q4 で実行されます。

図 6-5: クロック/命令サイクル



例 6-3: 命令パイプラインフロー



## 6.2.3 プログラムメモリ内の命令

プログラムメモリのアドレスはバイト単位で指定します。命令は、2 バイトまたは 4 バイトのいずれかでプログラムメモリに格納されます。命令ワードの最下位バイト (LSB) は、必ずプログラムメモリ内の偶数アドレス位置 (LSb = 0) に保存されます。命令の境界に合わせて PC は 2 ずつインクリメントするため、最下位ビットは常に「0」として読み出されます ( [セクション 6.1.5「プログラムカウンタ」](#) 参照 )。

[図 6-6](#) に、プログラムメモリへの命令ワードの格納例を示します。

CALL および GOTO 命令には、プログラムメモリの絶対アドレスが埋め込まれています。命令は全てワード境界に沿って格納されるため、これらの命令に含まれるデータもワードのアドレスです。ワードアドレスを PC<20:1> に書き込み、プログラムメモリ内のバイトアドレスにアクセスします。[図 6-6](#) の Instruction 2 は、GOTO 0006h という命令がプログラムメモリ内のようにエンコードされるかを示します。相対アドレスオフセットをエンコードするプログラム分岐命令も同様に動作します。分岐命令に保存されるオフセット値は、PC をオフセットさせる量をシングルワード命令の数として表します。命令セットの詳細は、[セクション 26.0「命令セットの概要」](#) で説明します。

**図 6-6: プログラムメモリ内の命令**

Program Memory Byte Locations →			Word Address		
			LSB = 1	LSB = 0	↓
					000000h
					000002h
					000004h
					000006h
Instruction 1:	MOVLW	055h	0Fh	55h	000008h
Instruction 2:	GOTO	0006h	EFh	03h	00000Ah
			F0h	00h	00000Ch
			C1h	23h	00000Eh
			F4h	56h	000010h
					000012h
					000014h

## 6.2.4 2 ワード命令

標準の PIC18 命令セットには、2 ワード命令として CALL、MOVFF、GOTO、LSFR の 4 つがあります。いずれの命令も 2 ワード目の最上位 (MSb) 4 ビットは常に「1111」で、残りの 12 ビットはリテラルデータ (通常はデータメモリのアドレス) を収めます。

2 ワード目の最上位 4 ビットの値「1111」は、特殊な NOP 動作を規定します。2 ワード命令が正しいシーケンスで実行 (すなわち 1 ワード目の直後に 2 ワード目が実行) された場合、2 ワード目のデータにアクセス

して命令シーケンスで使います。しかし、何らかの理由で 1 ワード目がスキップされ、2 ワード目だけが実行された場合、NOP が実行されます。このような仕組みは、2 ワード命令の前に PC を変更する条件分岐命令が存在する場合に必要です。[例 6-4](#) に、動作例を示します。

**Note:** 拡張命令セットの 2 ワード命令については、[セクション 6.5「プログラムメモリと拡張命令セット」](#) を参照してください。

**例 6-4: 2 ワード命令**

例 1:	
オブジェクトコード	ソースコード
0110 0110 0000 0000	TSTFSZ REG1 ; is RAM location 0?
1100 0001 0010 0011	MOVFF REG1, REG2 ; No, skip this word
1111 0100 0101 0110	; Execute this word as a NOP
0010 0100 0000 0000	ADDWF REG3 ; continue code
例 2:	
オブジェクトコード	ソースコード
0110 0110 0000 0000	TSTFSZ REG1 ; is RAM location 0?
1100 0001 0010 0011	MOVFF REG1, REG2 ; Yes, execute this word
1111 0100 0101 0110	; 2nd word of instruction
0010 0100 0000 0000	ADDWF REG3 ; continue code

## 6.3 データメモリの構成

**Note:** PIC18 の拡張命令セットを有効にした場合、データメモリの動作の一部が変更されます。詳細は、[セクション 6.6「データメモリと拡張命令セット」](#)を参照してください。

PIC18 のデータメモリは SRAM として実装しています。データメモリの各レジスタは 12 ビットのアドレスを持ち、最大 4096 バイトのメモリをアドレス指定できます。メモリ空間は、16 個のバンク (1 バンクは 256 バイト) に分割されます。PIC18F97J60 ファミリの全デバイスは全てのバンクを実装しており、ユーザは 3808 バイトのデータメモリを利用できます。[図 6-7](#)に、これらのデバイスのデータメモリ構成を示します。

データメモリには特殊機能レジスタ (SFR) と汎用レジスタ (GPR) が含まれます。SFR はコントローラと周辺機能の制御またはステータス表示に、GPR はデータの格納とユーザアプリケーションの一時格納動作 (スクラッチパッド動作) に使います。未実装のメモリ位置を読み出すと、常に「0」が返されます。

命令セットとアーキテクチャは、全てのバンク範囲に対応できます。直接、間接、インデックス付きのアドレッシングモードによって、データメモリの全空間にアクセスできます。アドレッシングモードについては、このセクションの後半で説明します。

頻繁に使われるレジスタ (ほとんどの SFR と一部の GPR) にシングルサイクルでアクセスできるように、PIC18 はアクセスバンクを実装しています。これは、SFR の大部分と GPR バンク 0 の下位部分に、BSR を使わないで高速にアクセスできるようにする 256 バイトのメモリ空間です。アクセス RAM については、[セクション 6.3.2「アクセスバンク」](#)で詳しく説明します。

### 6.3.1 バンクセレクトレジスタ

大きなデータメモリの空間に対しては、全てのアドレスに高速にアクセスできるように、効率的なアドレッシング方式が必要です。このため、読み書き動作ごとにアドレス全体を指定する必要がない事が理想です。PIC18 では、RAM のバンク方式によってこれを実現しています。この方式は、メモリ空間をそれぞれ 256 バイトの連続する 16 個のバンクに分割します。メモリ位置は、命令に応じて 12 ビットのアドレス全体で指定するか、8 ビットの下位アドレスと 4 ビットのバンクポイントに分けて指定します。

PIC18 命令セットのほとんどの命令は、バンクセレクトレジスタ (BSR) と呼ばれるバンクポイントを使います。この SFR にメモリ位置アドレスの最上位 4 ビットが格納され、下位 8 ビットは命令自体に埋め込まれます。BSR は下位 4 ビット (BSR3:BSR0) のみが実装されています。上位 4 ビットは未使用です。常に「0」が読み出され、書き込む事はできません。BSR には、MOVLB 命令によって値を直接格納できます。

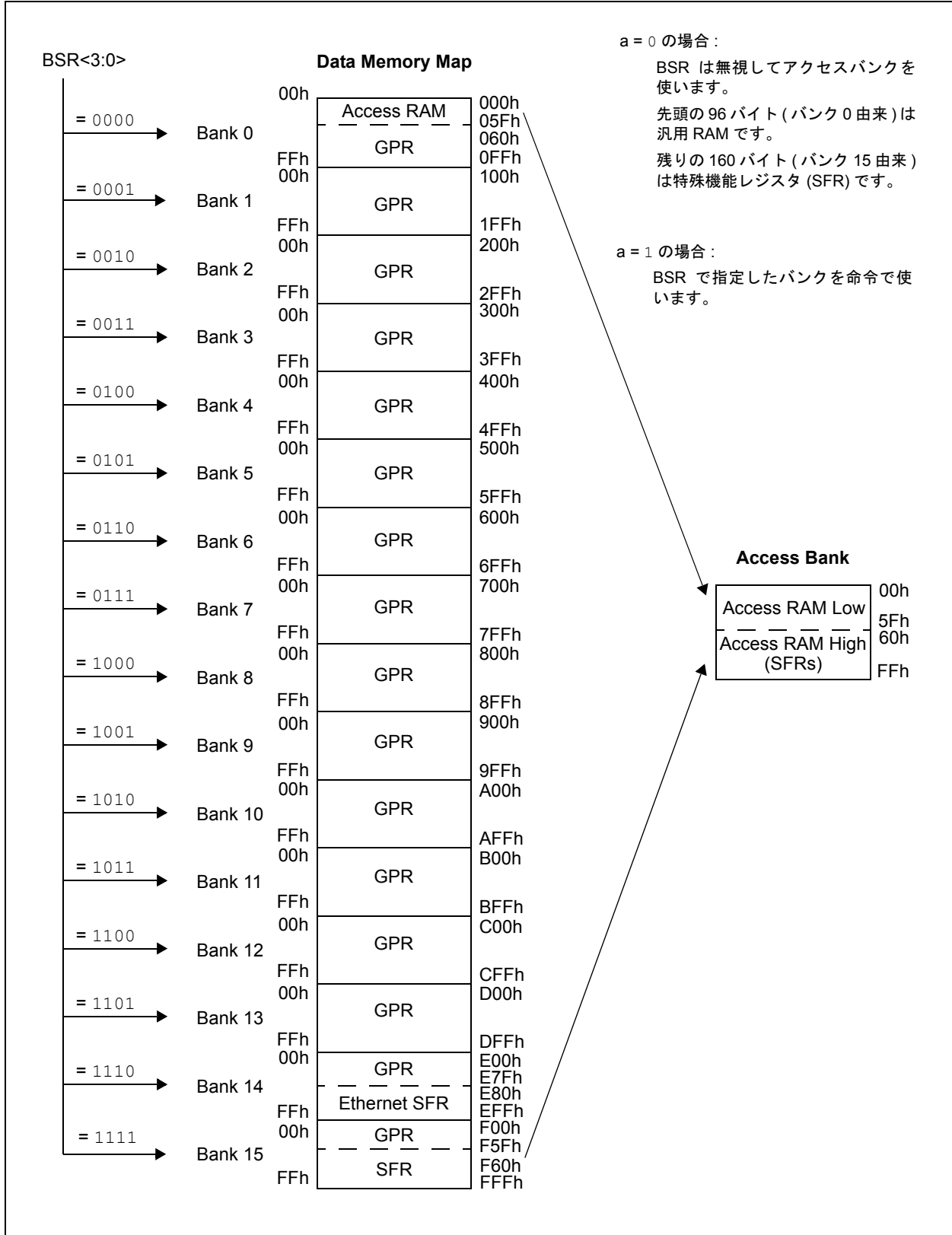
BSR の値はデータメモリ内のバンクを示します。命令内の 8 ビットはバンク内での位置を示しており、バンクの開始アドレスからのオフセットと見なす事ができます。[図 6-8](#)に、BSR の値とデータメモリのバンク分割の関係を示します。

最大 16 個のレジスタに同一の下位アドレスが存在するため、ユーザはデータの読み書きを実行する前に適切なバンクが選択されている事を確認する必要があります。例えば、プログラムデータを書き込むつもりで 8 ビットアドレスに F9h を指定した場合、BSR の値に 0Fh が指定されているとプログラムカウンタがリセットされてしまいます。

バンクは任意に指定できますが、読み書きを実行できるのは、実際に実装されているバンクに対してのみです。実装されていないバンクに対する書き込みは無視され、読み出しでは「0」が返されます。そのような場合でも、STATUS レジスタには動作が正常に実行されたかのような値が設定されます。[図 6-7](#)のデータメモリマップに、どのバンクが実装されているかを示します。

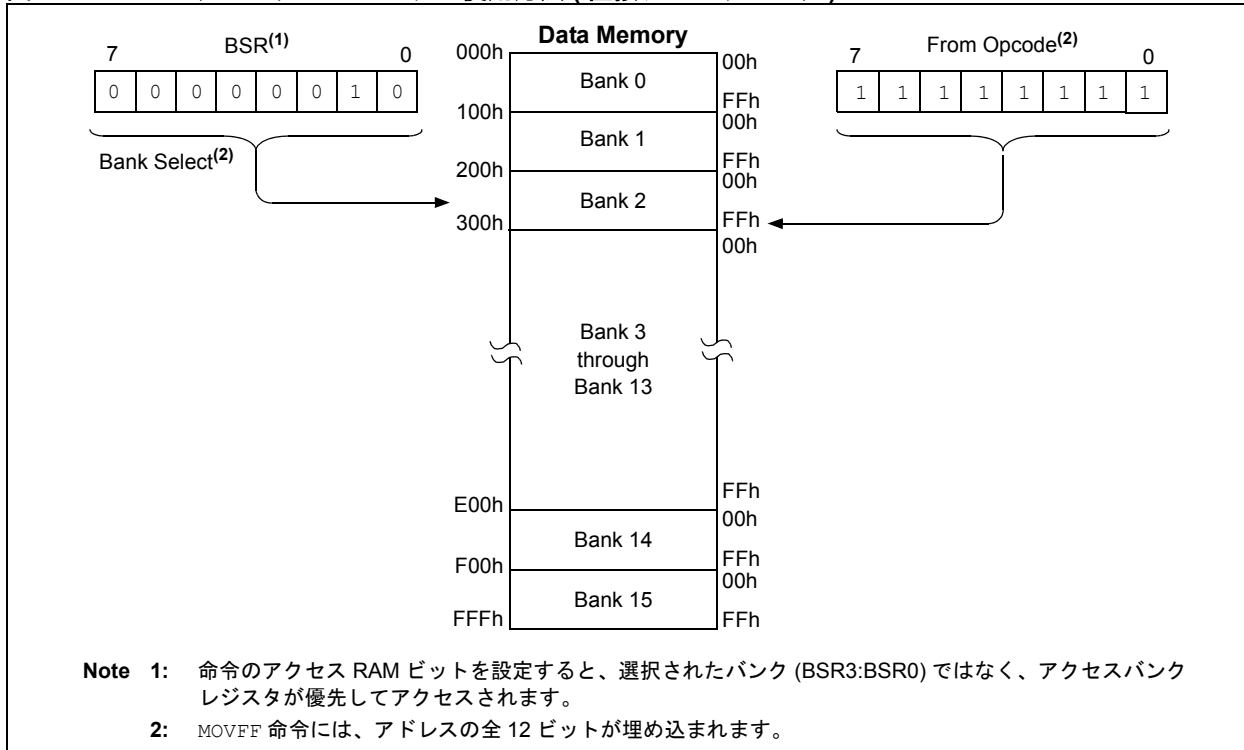
PIC18 のコア命令セットの中では、MOVFF 命令のみが、移動元および移動先レジスタのアドレス全 12 ビットを指定します。この命令は BSR を無視します。その他の命令は、いずれもオペランドとして下位アドレスのみを指定するため、対象のレジスタ位置を指定するには、BSR またはアクセスバンクを使う必要があります。

図 6-7: PIC18F97J60 ファミリのデータメモリマップ



# PIC18F97J60 ファミリ

図 6-8: バンクセレクトレジスタの使用法 (直接アドレッシング)



## 6.3.2 アクセスバンク

ユーザは、BSR と命令に埋め込まれた 8 ビットのアドレスによってデータメモリの全アドレス範囲を指定できますが、これは同時に、正しいバンクが選択されているかどうかユーザが常に確認する必要がある事を意味します。この確認を怠ると、誤ったメモリ位置に対する読み書きが実行される恐れがあります。例えば、演算結果を GPR ではなく SFR に書き込んでしまった場合の影響は深刻です。しかし、データメモリの読み書きごとに BSR を確認、変更する事は非常に非効率です。

最も頻繁に使われるデータメモリ位置へのアクセスを効率化するために、データメモリの一部に対してアクセスバンクが設定されています。これにより、BSR を指定せずにアクセスできます。アクセスバンクは、バンク 0 のメモリの先頭 96 バイト (00h-5Fh) とバンク 15 の末尾 160 バイト (60h-FFh) で構成されます。下位のブロックは「アクセス RAM」と呼ばれ、GPR で構成されます。上位のブロックにはデバイスの SFR がマッピングされます。これら 2 つの領域は、アクセスバンク内で連続してマッピングされ、8 ビットでリアルタイムにアドレスを指定できます (図 6-7)。

アクセスバンクは、PIC18 のコア命令セットのうち、アクセス RAM ビット (命令内の「a」パラメータ) を含むものが使います。「a」が「1」の場合、命令はデータメモリのアドレスとして、BSR とオペコードに含まれる 8 ビットのアドレスを使います。一方、「a」が「0」の場合、命令は BSR の現在値を無視し、アクセスバンクのアドレスマップを使います。

この「強制」アドレッシングを使う事で、命令は最初に BSR を更新する事なく、シングルサイクルでデータアドレスに対する演算を実行できます。これによって、アクセスバンク内の 60h 以上については、8 ビットアドレスによって SFR に対する評価と演算をより効率的に実行できます。60h よりも下位のアクセス RAM は、直近の計算結果や頻繁に使われるプログラム変数等、高速のアクセスが必要なデータ値の格納に適しています。また、アクセス RAM を使う事で、変数の内容に応じた保存と切り換えを、より高速で効率的なコードによって実装できます。

拡張命令セットが有効な場合 (コンフィグレーションビット XINST = 1)、アクセスバンクのマッピングは若干異なります。これについては、[セクション 6.6.3「インデックス付きリテラル オフセット モードにおけるアクセスバンクのマッピング」](#)で詳しく説明します。

## 6.3.3 汎用レジスタファイル

PIC18 は、GPR 領域内にバンク分けされたメモリを設定できます。これは、全命令で使えるデータ RAM です。GPR はバンク 0 の下端 (アドレス 000h) から始まり、上位にある SFR 領域の下端に向けて並びます。GPR はパワーオンリセットによって初期化されず、その他のいずれのリセットが発生しても変化しません。



# PIC18F97J60 ファミリ

## 6.3.4 特殊機能レジスタ

特殊機能レジスタ (SFR) は、CPU と周辺モジュールがデバイスの動作を制御するために使うレジスタです。これらのレジスタは SRAM として実装されています。

SFR の大部分はデータメモリの上端 (FFFh) から下位に向かって並び、バンク 15 の上半分以上 (F60h ~ FFFh) を占有します。SFR は、デバイスの「コア」機能 (ALU、リセット、割り込み) に関連するものと、周辺機能に関連するものの 2 つに分類できます。リセットと割り込みのレジスタは、それぞれ独立した章で説

明しますが、ALU の STATUS レジスタについては、このセクションの後半で説明します。また、周辺機能の動作に関連するレジスタについては、その周辺機能の章の中で説明します。

ほとんどの SFR は、制御対象の機能を持つ周辺モジュールごとにまとめて配置されています。未使用の SFR 位置にはメモリが実装されておらず、「0」が読み出されます。表 6-3 に SFR の一覧を示します。表 6-5 に詳しい説明を示します。

表 6-3: PIC18F97J60 ファミリの特殊機能レジスタマップ

アドレス	名前	アドレス	名前	アドレス	名前	アドレス	名前	アドレス	名前
FFFh	TOSU	FDFh	INDF2 <sup>(1)</sup>	FBFh	CCPR1H	F9Fh	IPR1	F7Fh	SPBRGH1
FFEh	TOSH	FDEh	POSTINC2 <sup>(1)</sup>	FBEh	CCPR1L	F9Eh	PIR1	F7Eh	BAUDCON1
FFDh	TOSL	FDDh	POSTDEC2 <sup>(1)</sup>	FBDh	CCP1CON	F9Dh	PIE1	F7Dh	SPBRGH2
FFCh	STKPTR	FDCh	PREINC2 <sup>(1)</sup>	FBCh	CCPR2H	F9Ch	MEMCON <sup>(4)</sup>	F7Ch	BAUDCON2
FFBh	PCLATU	FDBh	PLUSW2 <sup>(1)</sup>	FBBh	CCPR2L	F9Bh	OSCTUNE	F7Bh	ERDPTH
FFAh	PCLATH	FDAh	FSR2H	FBAh	CCP2CON	F9Ah	TRISJ <sup>(3)</sup>	F7Ah	ERDPTL
FF9h	PCL	FD9h	FSR2L	FB9h	CCPR3H	F99h	TRISH <sup>(3)</sup>	F79h	ECCP1DEL
FF8h	TBLPTRU	FD8h	STATUS	FB8h	CCPR3L	F98h	TRISG	F78h	TMR4
FF7h	TBLPTRH	FD7h	TMR0H	FB7h	CCP3CON	F97h	TRISF	F77h	PR4
FF6h	TBLPTRL	FD6h	TMR0L	FB6h	ECCP1AS	F96h	TRISE	F76h	T4CON
FF5h	TABLAT	FD5h	T0CON	FB5h	CVRCON	F95h	TRISD	F75h	CCPR4H
FF4h	PRODH	FD4h	._ <sup>(2)</sup>	FB4h	CMCON	F94h	TRISC	F74h	CCPR4L
FF3h	PRODL	FD3h	OSCCON	FB3h	TMR3H	F93h	TRISB	F73h	CCP4CON
FF2h	INTCON	FD2h	ECON1	FB2h	TMR3L	F92h	TRISA	F72h	CCPR5H
FF1h	INTCON2	FD1h	WDTCON	FB1h	T3CON	F91h	LATJ <sup>(3)</sup>	F71h	CCPR5L
FF0h	INTCON3	FD0h	RCON	FB0h	PSPCON	F90h	LATH <sup>(3)</sup>	F70h	CCP5CON
FEFh	INDF0 <sup>(1)</sup>	FCFh	TMR1H	FAFh	SPBRG1	F8Fh	LATG	F6Fh	SPBRG2
FEeh	POSTINC0 <sup>(1)</sup>	FCEh	TMR1L	FAEh	RCREG1	F8Eh	LATF	F6Eh	RCREG2
FEDh	POSTDEC0 <sup>(1)</sup>	FCDh	T1CON	FADh	TXREG1	F8Dh	LATE	F6Dh	TXREG2
FECh	PREINC0 <sup>(1)</sup>	FCCh	TMR2	FACH	TXSTA1	F8Ch	LATD	F6Ch	TXSTA2
FEbh	PLUSW0 <sup>(1)</sup>	FCBh	PR2	FABh	RCSTA1	F8Bh	LATC	F6Bh	RCSTA2
FEAh	FSR0H	FCAh	T2CON	FAAh	._ <sup>(2)</sup>	F8Ah	LATB	F6Ah	ECCP3AS
FE9h	FSR0L	FC9h	SSP1BUF	FA9h	._ <sup>(2)</sup>	F89h	LATA	F69h	ECCP3DEL
FE8h	WREG	FC8h	SSP1ADD	FA8h	._ <sup>(2)</sup>	F88h	PORTJ <sup>(3)</sup>	F68h	ECCP2AS
FE7h	INDF1 <sup>(1)</sup>	FC7h	SSP1STAT	FA7h	EECON2 <sup>(1)</sup>	F87h	PORTH <sup>(3)</sup>	F67h	ECCP2DEL
FE6h	POSTINC1 <sup>(1)</sup>	FC6h	SSP1CON1	FA6h	EECON1	F86h	PORTG	F66h	SSP2BUF
FE5h	POSTDEC1 <sup>(1)</sup>	FC5h	SSP1CON2	FA5h	IPR3	F85h	PORTF	F65h	SSP2ADD
FE4h	PREINC1 <sup>(1)</sup>	FC4h	ADRESH	FA4h	PIR3	F84h	PORTE	F64h	SSP2STAT
FE3h	PLUSW1 <sup>(1)</sup>	FC3h	ADRESL	FA3h	PIE3	F83h	PORTD	F63h	SSP2CON1
FE2h	FSR1H	FC2h	ADCON0	FA2h	IPR2	F82h	PORTC	F62h	SSP2CON2
FE1h	FSR1L	FC1h	ADCON1	FA1h	PIR2	F81h	PORTB	F61h	EDATA
FE0h	BSR	FC0h	ADCON2	FA0h	PIE2	F80h	PORTA	F60h	EIR

- Note**
- 1: これは物理レジスタではありません。
  - 2: 未実装のレジスタからは「0」が読み出されます。
  - 3: 64 ピンデバイスにはこのレジスタはありません。
  - 4: このレジスタは 64 ピンおよび 80 ピンのデバイスにはありません。

# PIC18F97J60 ファミリ

## 6.3.5 Ethernet SFR

PIC18F97J60 ファミリには、バンク 15 の標準 SFR セット以外にも SFR セットがあります。これらの SFR は全て Ethernet モジュールに関するもので、バンク 14 の上半分 (E80h ~ EFFh) を占めています。

**Note:** 性能向上のため、頻繁にアクセスする Ethernet 関連レジスタは標準 SFR バンク (F60h ~ FFFh) に配置されています。

表 6-4 に、Ethernet SFR の一覧を示します。全 SFR の詳細は表 6-5 を参照してください。

表 6-4: PIC18F97J60 ファミリの Ethernet SFR マップ

アドレス	名前	アドレス	名前	アドレス	名前	アドレス	名前
EFFh	_(1)	EDFh	_(1)	EBFh	_(1)	E9Fh	_(1)
EFEh	ECON2	EDEh	_(1)	EBEh	_(1)	E9Eh	_(1)
EFDh	ESTAT	EDDh	_(1)	EBDh	_(1)	E9Dh	_(1)
EFCh	_(1)	EDCh	_(1)	EBCh	_(1)	E9Ch	_(1)
EFBh	EIE	EDBh	_(1)	EBBh	_(1)	E9Bh	_(1)
EFAh	_(1)	EDAh	_(1)	EBAh	_(1)	E9Ah	_(1)
EF9h	_(2)	ED9h	EPKTCNT	EB9h	MIRRDH	E99h	EPAUSH
EF8h	_(2)	ED8h	ERXFCON	EB8h	MIRDL	E98h	EPAUSL
EF7h	EDMACSH	ED7h	_(1)	EB7h	MIWRH	E97h	EFLOCON
EF6h	EDMACSL	ED6h	_(1)	EB6h	MIWRL	E96h	_(2)
EF5h	EDMADSTH	ED5h	EPMOH	EB5h	_(1)	E95h	_(2)
EF4h	EDMADSTL	ED4h	EPMOL	EB4h	MIREGADR	E94h	_(2)
EF3h	EDMANDH	ED3h	_(2)	EB3h	_(2)	E93h	_(2)
EF2h	EDMANDL	ED2h	_(2)	EB2h	MICMD	E92h	_(2)
EF1h	EDMASTH	ED1h	EPMCSH	EB1h	_(1)	E91h	_(2)
EF0h	EDMASTL	ED0h	EPMCSL	EB0h	_(1)	E90h	_(2)
EEFh	ERXWRPTH	ECFh	EPMM7	EAFh	_(2)	E8Fh	_(2)
EEEh	ERXWRPTL	ECEh	EPMM6	EAEh	_(1)	E8Eh	_(2)
EEDh	ERXRDPH	ECDh	EPMM5	EADh	_(1)	E8Dh	_(2)
EECh	ERXRPTL	ECCh	EPMM4	EACH	_(1)	E8Ch	_(2)
EEBh	ERXNDH	ECBh	EPMM3	EABh	MAMXFLH	E8Bh	_(2)
EEAh	ERXNDL	ECAh	EPMM2	EAAh	MAMXFLH	E8Ah	MISTAT
EE9h	ERXSTH	EC9h	EPMM1	EA9h	_(1)	E89h	_(1)
EE8h	ERXSTL	EC8h	EPMM0	EA8h	_(1)	E88h	_(1)
EE7h	ETXNDH	EC7h	EHT7	EA7h	MAIPGH	E87h	_(1)
EE6h	ETXNDL	EC6h	EHT6	EA6h	MAIPGL	E86h	_(1)
EE5h	ETXSTH	EC5h	EHT5	EA5h	_(2)	E85h	MAADR2
EE4h	ETXSTL	EC4h	EHT4	EA4h	MABBIPG	E84h	MAADR1
EE3h	EWRPTH	EC3h	EHT3	EA3h	MACON4	E83h	MAADR4
EE2h	EWRPTL	EC2h	EHT2	EA2h	MACON3	E82h	MAADR3
EE1h	_(1)	EC1h	EHT1	EA1h	_(1)	E81h	MAADR6
EE0h	_(1)	EC0h	EHT0	EA0h	MACON1	E80h	MAADR5

**Note** 1: 予約済みのレジスタ位置のため書き換えはできません。  
 2: 未実装のレジスタからは「0」が読み出されます。

# PIC18F97J60 ファミリ

表 6-5: レジスタファイルのまとめ (PIC18F97J60 ファミリ)

SFR 名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR、BOR 時の値	詳細 ページ
TOSU	-	-	-	Top-of-Stack レジスタの最上位バイト (TOS<20:16>)					---0 0000	69, 81
TOSH	Top-of-Stack レジスタの上位バイト (TOS<15:8>)								0000 0000	69, 81
TOSL	Top-of-Stack レジスタの下位バイト (TOS<7:0>)								0000 0000	69, 81
STKPTR	STKFUL <sup>(1)</sup>	STKUNF <sup>(1)</sup>	-	SP4	SP3	SP2	SP1	SP0	00-0 0000	69, 82
PCLATU	-	-	bit 21 <sup>(2)</sup>	PC<20:16> の保持レジスタ					---0 0000	69, 81
PCLATH	PC<15:8> の保持レジスタ								0000 0000	69, 81
PCL	PC の下位バイト (PC<7:0>)								0000 0000	69, 81
TBLPTRU	-	-	bit 21	プログラムメモリ テーブルポインタの最上位バイト (TBLPTR<20:16>)					--00 0000	69, 108
TBLPTRH	プログラムメモリ テーブルポインタの上位バイト (TBLPTR<15:8>)								0000 0000	69, 108
TBLPTRL	プログラムメモリ テーブルポインタの下位バイト (TBLPTR<7:0>)								0000 0000	69, 108
TABLAT	プログラムメモリ テーブルラッチ								0000 0000	69, 108
PRODH	積レジスタ上位バイト								xxxx xxxx	69, 127
PRODL	積レジスタ下位バイト								xxxx xxxx	69, 127
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	69, 131
INTCON2	RBPÜ	INTEDG0	INTEDG1	INTEDG2	INTEDG3	TMR0IP	INT3IP	RBIP	1111 1111	69, 132
INTCON3	INT2IP	INT1IP	INT3IE	INT2IE	INT1IE	INT3IF	INT2IF	INT1IF	1100 0000	69, 133
INDF0	FSR0 の内容が指すデータメモリのアドレスにアクセスします。FSR0 の値は変化しません (物理レジスタではありません)								N/A	69, 99
POSTINC0	FSR0 の内容が指すデータメモリのアドレスにアクセスします。アクセス後、FSR0 の値がインクリメントされます (物理レジスタではありません)								N/A	69, 100
POSTDEC0	FSR0 の内容が指すデータメモリのアドレスにアクセスします。アクセス後、FSR0 の値がデクリメントされます (物理レジスタではありません)								N/A	69, 100
PREINC0	FSR0 の内容が指すデータメモリのアドレスにアクセスします。アクセス前に FSR0 の値がインクリメントされます (物理レジスタではありません)								N/A	69, 100
PLUSW0	FSR0 の内容が指すデータメモリのアドレスにアクセスします。アクセス前に FSR0 の値が W レジスタの値だけオフセットされます (物理レジスタではありません)								N/A	69, 100
FSR0H	-	-	-	-	間接データメモリ アドレスポインタ 0 上位バイト				---- xxxx	69, 99
FSR0L	間接データメモリ アドレスポインタ 0 下位バイト								xxxx xxxx	69, 100
WREG	ワーキング レジスタ								xxxx xxxx	69
INDF1	FSR1 の内容が指すデータメモリのアドレスにアクセスします。FSR1 の値は変化しません (物理レジスタではありません)								N/A	69, 99
POSTINC1	FSR1 の内容が指すデータメモリのアドレスにアクセスします。アクセス後、FSR1 の値がインクリメントされます (物理レジスタではありません)								N/A	69, 100
POSTDEC1	FSR1 の内容が指すデータメモリのアドレスにアクセスします。アクセス後、FSR1 の値がデクリメントされます (物理レジスタではありません)								N/A	69, 100
PREINC1	FSR1 の内容が指すデータメモリのアドレスにアクセスします。アクセス前に FSR1 の値がインクリメントされます (物理レジスタではありません)								N/A	69, 100
PLUSW1	FSR1 の内容が指すデータメモリのアドレスにアクセスします。アクセス前に FSR1 の値が W レジスタの値だけオフセットされます (物理レジスタではありません)								N/A	69, 100
FSR1H	-	-	-	-	間接データメモリ アドレスポインタ 1 上位バイト				---- xxxx	69, 99
FSR1L	間接データメモリ アドレスポインタ 1 下位バイト								xxxx xxxx	69, 99
BSR	-	-	-	-	バンクセレクト レジスタ				---- 0000	69, 99

凡例: x = 未知、u = 不変、- = 未実装、「0」として読み出し、q = 条件による、r = 予約済みビットのため変更不可、網掛けの部分は未実装、「0」として読み出し

Note 1: bit 7 と bit 6 は、ユーザ ソフトウェアまたは POR によってクリアされます。

2: PC の bit 21 はシリアル プログラミング モードでのみ使えます。

3: リセット値は、2 段階起動が有効な場合が「0」で、無効な場合が「1」です。

4: MSSP モジュールが I<sup>2</sup>C™ スレーブモードで動作時のビットの代替名と定義です。

5: これらのビット / レジスタを利用できるのは 100 ピンデバイスのみです。それ以外のデバイスには実装されておらず、「0」として読み出されます。表示してあるリセット値は、100 ピンデバイスの場合のみです。

6: これらのビット / レジスタを使うのは 80 ピンおよび 100 ピンのデバイスのみです。64 ピンデバイスでは未実装で、「0」として読み出されます。リセット時の値は 100 ピンデバイスの場合です。

7: マイクロコントローラ モードでは、このレジスタのビットは書き込み不可で「0」として読み出されます。

8: PLEN を使えるのはオシレータモードに ECPLL または HSPLL を選択した場合のみです。それ以外では「0」として読み出されます。

9: 100 ピンデバイスにのみ実装されており、使えるのはマイクロコントローラ モード時のみです。

# PIC18F97J60 ファミリ

表 6-5: レジスタファイルのまとめ (PIC18F97J60 ファミリ) (続き)

SFR 名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR、BOR 時の値	詳細 ページ	
INDF2	FSR2 の内容が指すデータメモリのアドレスにアクセスします。FSR2 の値は変化しません (物理レジスタではありません)								N/A	69, 99	
POSTINC2	FSR2 の内容が指すデータメモリのアドレスにアクセスします。アクセス後、FSR2 の値がインクリメントされます (物理レジスタではありません)								N/A	69, 100	
POSTDEC2	FSR2 の内容が指すデータメモリのアドレスにアクセスします。アクセス後、FSR2 の値がデクリメントされます (物理レジスタではありません)								N/A	69, 100	
PREINC2	FSR2 の内容が指すデータメモリのアドレスにアクセスします。アクセス前に FSR2 の値がインクリメントされます (物理レジスタではありません)								N/A	69, 100	
PLUSW2	FSR2 の内容が指すデータメモリのアドレスにアクセスします。アクセス前に FSR2 の値が W レジスタの値だけオフセットされます (物理レジスタではありません)								N/A	69, 100	
FSR2H	-	-	-	-	間接データメモリ アドレスポインタ 2 上位バイト					---- xxxx	69, 99
FSR2L	間接データメモリ アドレスポインタ 2 下位バイト								xxxx xxxx	69, 99	
STATUS	-	-	-	N	OV	Z	DC	C	---x xxxx	70, 97	
TMR0H	Timer0 レジスタ 上位バイト								0000 0000	70, 171	
TMR0L	Timer0 レジスタ 下位バイト								xxxx xxxx	70, 171	
T0CON	TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0	1111 1111	70, 171	
OSCCON	IDLEN	-	-	-	OSTS <sup>(3)</sup>	-	SCS1	SCS0	0--- q-00	70, 53	
ECON1	TXRST	RXRST	DMAST	CSUMEN	TXRTS	RXEN	-	-	0000 00--	70, 227	
WDTCON	-	-	-	-	-	-	-	SWDTEN	--- ---0	70, 368	
RCON	IPEN	-	$\overline{CM}$	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$	0-q1 1100	70, 64, 143	
TMR1H	Timer1 レジスタ 上位バイト								xxxx xxxx	70, 175	
TMR1L	Timer1 レジスタ 下位バイト								xxxx xxxx	70, 175	
T1CON	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	TMR1CS	TMR1ON	0000 0000	70, 175	
TMR2	Timer2 レジスタ								0000 0000	70, 180	
PR2	Timer2 周期レジスタ								1111 1111	70, 180	
T2CON	-	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	70, 180	
SSP1BUF	MSSP1 受信バッファ / 送信レジスタ								xxxx xxxx	70, 279	
SSP1ADD	MSSP1 アドレスレジスタ (I <sup>2</sup> C™ スレーブモード)、MSSP1 baud レート再読み込みレジスタ (I <sup>2</sup> C マスタモード)								0000 0000	70, 279	
SSP1STAT	SMP	CKE	$\overline{D/A}$	P	S	$\overline{R/W}$	UA	BF	0000 0000	70, 270, 280	
SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	70, 271, 281	
SSP1CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	70, 282	
	GCEN	ACKSTAT	ADMSK5 <sup>(4)</sup>	ADMSK4 <sup>(4)</sup>	ADMSK3 <sup>(4)</sup>	ADMSK2 <sup>(4)</sup>	ADMSK1 <sup>(4)</sup>	SEN			
ADRESH	A/D 変換結果レジスタ 上位バイト								xxxx xxxx	70, 347	
ADRESL	A/D 変換結果レジスタ 下位バイト								xxxx xxxx	70, 347	
ADCON0	ADCAL	-	CHS3	CHS2	CHS1	CHS0	$\overline{GO/DONE}$	ADON	0-00 0000	70, 339	
ADCON1	-	-	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	--00 0000	70, 340	
ADCON2	ADFM	-	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0	0-00 0000	70, 341	
CCPR1H	キャプチャ / コンペア / PWM レジスタ 1 上位バイト								xxxx xxxx	70, 193	
CCPR1L	キャプチャ / コンペア / PWM レジスタ 1 下位バイト								xxxx xxxx	70, 193	

凡例: x = 未知、u = 不変、- = 未実装、「0」として読み出し、q = 条件による、r = 予約済みビットのため変更不可、網掛けの部分は未実装、「0」として読み出し

- Note 1: bit 7 と bit 6 は、ユーザ ソフトウェアまたは POR によってクリアされます。
- 2: PC の bit 21 はシリアル プログラミング モードでのみ使えます。
  - 3: リセット値は、2 段階起動が有効な場合が「0」で、無効な場合が「1」です。
  - 4: MSSP モジュールが I<sup>2</sup>C™ スレーブモードで動作時のビットの代替名と定義です。
  - 5: これらのビット / レジスタを利用できるのは 100 ピンデバイスのみです。それ以外のデバイスには実装されておらず、「0」として読み出されます。表示してあるリセット値は、100 ピンデバイスの場合のみです。
  - 6: これらのビット / レジスタを使えるのは 80 ピンおよび 100 ピンのデバイスのみです。64 ピンデバイスでは未実装で、「0」として読み出されます。リセット時の値は 100 ピンデバイスの場合です。
  - 7: マイクロコントローラ モードでは、このレジスタのビットは書き込み不可で「0」として読み出されます。
  - 8: PLEN を使えるのはオシレータモードに ECPLL または HSPLL を選択した場合のみです。それ以外では「0」として読み出されます。
  - 9: 100 ピンデバイスにのみ実装されており、使えるのはマイクロコントローラ モード時のみです。

# PIC18F97J60 ファミリ

表 6-5: レジスタファイルのまとめ (PIC18F97J60 ファミリ) (続き)

SFR 名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR、BOR 時の値	詳細 ページ
CCP1CON	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	0000 0000	70, 198
CCPR2H	キャプチャ/コンペア/PWM レジスタ 2 上位バイト								xxxx xxxx	70, 193
CCPR2L	キャプチャ/コンペア/PWM レジスタ 2 下位バイト								xxxx xxxx	70, 193
CCP2CON	P2M1	P2M0	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	0000 0000	70, 198
CCPR3H	キャプチャ/コンペア/PWM レジスタ 3 上位バイト								xxxx xxxx	70, 193
CCPR3L	キャプチャ/コンペア/PWM レジスタ 3 下位バイト								xxxx xxxx	70, 193
CCP3CON	P3M1	P3M0	DC3B1	DC3B0	CCP3M3	CCP3M2	CCP3M1	CCP3M0	0000 0000	70, 198
ECCP1AS	ECCP1ASE	ECCP1AS2	ECCP1AS1	ECCP1AS0	PSS1AC1	PSS1AC0	PSS1BD1	PSS1BD0	0000 0000	70, 212
CVRCON	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	0000 0000	70, 355
CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0111	70, 349
TMR3H	Timer3 レジスタ上位バイト								xxxx xxxx	70, 183
TMR3L	Timer3 レジスタ下位バイト								xxxx xxxx	70, 183
T3CON	RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	T3SYN $\bar{C}$	TMR3CS	TMR3ON	0000 0000	71, 183
PSPCON <sup>(5)</sup>	IBF	OBF	IBOV	PSPMODE	-	-	-	-	0000 ----	71, 169
SPBRG1	EUSART1 baud レート ジェネレータ レジスタ下位バイト								0000 0000	71, 320
RCREG1	EUSART1 受信レジスタ								0000 0000	71, 327
TXREG1	EUSART1 送信レジスタ								xxxx xxxx	71, 329
TXSTA1	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	71, 320
RCSTA1	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	71, 320
EECON2	プログラムメモリ制御レジスタ (物理レジスタではありません)								---- ----	71, 106
EECON1	-	-	-	FREE	WRERR	WREN	WR	-	---0 x00-	71, 107
IPR3	SSP2IP <sup>(5)</sup>	BCL2IP <sup>(5)</sup>	RC2IP <sup>(6)</sup>	TX2IP <sup>(6)</sup>	TMR4IP	CCP5IP	CCP4IP	CCP3IP	1111 1111	71, 142
PIR3	SSP2IF <sup>(5)</sup>	BCL2IF <sup>(5)</sup>	RC2IF <sup>(6)</sup>	TX2IF <sup>(6)</sup>	TMR4IF	CCP5IF	CCP4IF	CCP3IF	0000 0000	71, 136
PIE3	SSP2IE <sup>(5)</sup>	BCL2IE <sup>(5)</sup>	RC2IE <sup>(6)</sup>	TX2IE <sup>(6)</sup>	TMR4IE	CCP5IE	CCP4IE	CCP3IE	0000 0000	71, 139
IPR2	OSCFIP	CMIP	ETHIP	r	BCL1IP	-	TMR3IP	CCP2IP	1111 1-11	71, 141
PIR2	OSCFIF	CMIF	ETHIF	r	BCL1IF	-	TMR3IF	CCP2IF	0000 0-00	71, 135
PIE2	OSCFIE	CMIE	ETHIE	r	BCL1IE	-	TMR3IE	CCP2IE	0000 0-00	71, 138
IPR1	PSPIP <sup>(9)</sup>	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	1111 1111	71, 140
PIR1	PSPIF <sup>(9)</sup>	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	0000 0000	71, 134
PIE1	PSPIE <sup>(9)</sup>	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	0000 0000	71, 137
MEMCON <sup>(5,7)</sup>	EBDIS	-	WAIT1	WAIT0	-	-	WM1	WM0	0-00 --00	71, 116
OSCTUNE	PPST1	PLLEN <sup>(8)</sup>	PPST0	PPRE	-	-	-	-	0000 ----	71, 51
TRISJ <sup>(6)</sup>	TRISJ7 <sup>(5)</sup>	TRISJ6 <sup>(5)</sup>	TRISJ5 <sup>(6)</sup>	TRISJ4 <sup>(6)</sup>	TRISJ3 <sup>(5)</sup>	TRISJ2 <sup>(5)</sup>	TRISJ1 <sup>(5)</sup>	TRISJ0 <sup>(5)</sup>	1111 1111	71, 167
TRISH <sup>(6)</sup>	TRISH7 <sup>(6)</sup>	TRISH6 <sup>(6)</sup>	TRISH5 <sup>(6)</sup>	TRISH4 <sup>(6)</sup>	TRISH3 <sup>(6)</sup>	TRISH2 <sup>(6)</sup>	TRISH1 <sup>(6)</sup>	TRISH0 <sup>(6)</sup>	1111 1111	71, 165
TRISG	TRISG7 <sup>(5)</sup>	TRISG6 <sup>(5)</sup>	TRISG5 <sup>(5)</sup>	TRISG4	TRISG3 <sup>(6)</sup>	TRISG2 <sup>(6)</sup>	TRISG1 <sup>(6)</sup>	TRISG0 <sup>(6)</sup>	1111 1111	71, 163
TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	TRISF0 <sup>(5)</sup>	1111 1111	71, 161
TRISE	TRISE7 <sup>(6)</sup>	TRISE6 <sup>(6)</sup>	TRISE5	TRISE4	TRISE3	TRISE2	TRISE1	TRISE0	1111 1111	71, 159
TRISD	TRISD7 <sup>(5)</sup>	TRISD6 <sup>(5)</sup>	TRISD5 <sup>(5)</sup>	TRISD4 <sup>(5)</sup>	TRISD3 <sup>(5)</sup>	TRISD2	TRISD1	TRISD0	1111 1111	71, 156
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	71, 153
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	71, 150
TRISA	-	-	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	71, 147

凡例: x = 未知、u = 不変、- = 未実装、「0」として読み出し、q = 条件による、r = 予約済みビットのため変更不可、網掛けの部分は未実装、「0」として読み出し

- Note 1: bit 7 と bit 6 は、ユーザ ソフトウェアまたは POR によってクリアされます。
- 2: PC の bit 21 はシリアル プログラミング モードでのみ使えます。
  - 3: リセット値は、2 段階起動が有効な場合が「0」で、無効な場合が「1」です。
  - 4: MSSP モジュールが I<sup>2</sup>C™ スレープモードで動作時のビットの代替名と定義です。
  - 5: これらのビット/レジスタを利用できるのは 100 ピンデバイスのみです。それ以外のデバイスには実装されておらず、「0」として読み出されます。表示してあるリセット値は、100 ピンデバイスの場合のみです。
  - 6: これらのビット/レジスタを使えるのは 80 ピンおよび 100 ピンのデバイスのみです。64 ピンデバイスでは未実装で、「0」として読み出されます。リセット時の値は 100 ピンデバイスの場合です。
  - 7: マイクロコントローラ モードでは、このレジスタのビットは書き込み不可で「0」として読み出されます。
  - 8: PLLEN を使えるのはオシレータモードに ECPLL または HSPLL を選択した場合のみです。それ以外では「0」として読み出されます。
  - 9: 100 ピンデバイスにのみ実装されており、使えるのはマイクロコントローラ モード時のみです。

# PIC18F97J60 ファミリ

表 6-5: レジスタファイルのまとめ (PIC18F97J60 ファミリ) (続き)

SFR 名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR、BOR 時の値	詳細 ページ
LATJ <sup>(6)</sup>	LATJ7 <sup>(5)</sup>	LATJ6 <sup>(5)</sup>	LATJ5 <sup>(6)</sup>	LATJ4 <sup>(6)</sup>	LATJ3 <sup>(5)</sup>	LATJ2 <sup>(5)</sup>	LATJ1 <sup>(5)</sup>	LATJ0 <sup>(5)</sup>	xxxx xxxx	71, 167
LATH <sup>(6)</sup>	LATH7 <sup>(6)</sup>	LATH6 <sup>(6)</sup>	LATH5 <sup>(6)</sup>	LATH4 <sup>(6)</sup>	LATH3 <sup>(6)</sup>	LATH2 <sup>(6)</sup>	LATH1 <sup>(6)</sup>	LATH0 <sup>(6)</sup>	xxxx xxxx	71, 165
LATG	LATG7 <sup>(5)</sup>	LATG6 <sup>(5)</sup>	LATG5 <sup>(5)</sup>	LATG4	LATG3 <sup>(6)</sup>	LATG2 <sup>(6)</sup>	LATG1 <sup>(6)</sup>	LATG0 <sup>(6)</sup>	xxxx xxxx	72, 163
LATF	LATF7	LATF6	LATF5	LATF4	LATF3	LATF2	LATF1	LATF0 <sup>(5)</sup>	xxxx xxxx	72, 161
LATE	LATE7 <sup>(6)</sup>	LATE6 <sup>(6)</sup>	LATE5	LATE4	LATE3	LATE2	LATE1	LATE0	xxxx xxxx	72, 159
LATD	LATD7 <sup>(5)</sup>	LATD6 <sup>(5)</sup>	LATD5 <sup>(5)</sup>	LATD4 <sup>(5)</sup>	LATD3 <sup>(5)</sup>	LATD2	LATD1	LATD0	xxxx xxxx	72, 156
LATC	LATC7	LATC6	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	xxxx xxxx	72, 153
LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	xxxx xxxx	72, 150
LATA	RDPU	REPU	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0	00xx xxxx	72, 147
PORTJ <sup>(6)</sup>	RJ7 <sup>(5)</sup>	RJ6 <sup>(5)</sup>	RJ5 <sup>(6)</sup>	RJ4 <sup>(6)</sup>	RJ3 <sup>(5)</sup>	RJ2 <sup>(5)</sup>	RJ1 <sup>(5)</sup>	RJ0 <sup>(5)</sup>	xxxx xxxx	72, 167
PORTH <sup>(6)</sup>	RH7 <sup>(6)</sup>	RH6 <sup>(6)</sup>	RH5 <sup>(6)</sup>	RH4 <sup>(6)</sup>	RH3 <sup>(6)</sup>	RH2 <sup>(6)</sup>	RH1 <sup>(6)</sup>	RH0 <sup>(6)</sup>	0000 xxxx	72, 165
PORTG	RG7 <sup>(5)</sup>	RG6 <sup>(5)</sup>	RG5 <sup>(5)</sup>	RG4	RG3 <sup>(6)</sup>	RG2 <sup>(6)</sup>	RG1 <sup>(6)</sup>	RG0 <sup>(6)</sup>	111x xxxx	72, 163
PORTF	RF7	RF6	RF5	RF4	RF3	RF2	RF1	RF0 <sup>(5)</sup>	0000 0000	72, 161
PORTE	RE7 <sup>(6)</sup>	RE6 <sup>(6)</sup>	RE5	RE4	RE3	RE2	RE1	RE0	xxxx xxxx	72, 159
PORTD	RD7 <sup>(5)</sup>	RD6 <sup>(5)</sup>	RD5 <sup>(5)</sup>	RD4 <sup>(5)</sup>	RD3 <sup>(5)</sup>	RD2	RD1	RD0	xxxx xxxx	72, 156
PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	72, 153
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	72, 150
PORTA	RJPU <sup>(6)</sup>	-	RA5	RA4	RA3	RA2	RA1	RA0	0-0x 0000	72, 147
SPBRGH1	EUSART1 baud レート ジェネレータ レジスタ上位バイト								0000 0000	72, 320
BAUDCON1	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	-	WUE	ABDEN	0100 0-00	72, 318
SPBRGH2	EUSART2 baud レート ジェネレータ レジスタ上位バイト								0000 0000	72, 320
BAUDCON2	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	-	WUE	ABDEN	0100 0-00	72, 318
ERDPTH	-	-	-	バッファ読み出しポインタ上位バイト					---0 0101	72, 223
ERDPTL	バッファ読み出しポインタ下位バイト								1111 1010	72, 223
ECCP1DEL	P1RSEN	P1DC6	P1DC5	P1DC4	P1DC3	P1DC2	P1DC1	P1DC0	0000 0000	72, 211
TMR4	Timer4 レジスタ								0000 0000	72, 187
PR4	Timer4 周期レジスタ								1111 1111	72, 187
T4CON	-	T4OUTPS3	T4OUTPS2	T4OUTPS1	T4OUTPS0	TMR4ON	T4CKPS1	T4CKPS0	-000 0000	72, 187
CCPR4H	キャプチャ / コンペア / PWM レジスタ 4 上位バイト								xxxx xxxx	72, 193
CCPR4L	キャプチャ / コンペア / PWM レジスタ 4 下位バイト								xxxx xxxx	72, 193
CCP4CON	-	-	DC4B1	DC4B0	CCP4M3	CCP4M2	CCP4M1	CCP4M0	--00 0000	73, 189
CCPR5H	キャプチャ / コンペア / PWM レジスタ 5 上位バイト								xxxx xxxx	73, 193
CCPR5L	キャプチャ / コンペア / PWM レジスタ 5 下位バイト								xxxx xxxx	73, 193
CCP5CON	-	-	DC5B1	DC5B0	CCP5M3	CCP5M2	CCP5M1	CCP5M0	--00 0000	73, 189
SPBRG2	EUSART2 baud レート ジェネレータ レジスタ下位バイト								0000 0000	73, 320
RCREG2	EUSART2 受信レジスタ								0000 0000	73, 327
TXREG2	EUSART2 送信レジスタ								0000 0000	73, 329
TXSTA2	CSRC	TX9	TXEN	SYNC	SEnDB	BRGH	TRMT	TX9D	0000 0010	73, 316
RCSTA2	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	73, 317
ECCP3AS	ECCP3ASE	ECCP3AS2	ECCP3AS1	ECCP3AS0	PSS3AC1	PSS3AC0	PSS3BD1	PSS3BD0	0000 0000	73, 212
ECCP3DEL	P3RSEN	P3DC6	P3DC5	P3DC4	P3DC3	P3DC2	P3DC1	P3DC0	0000 0000	73, 211

凡例: x = 未知、u = 不変、- = 未実装、「0」として読み出し、q = 条件による、r = 予約済みビットのため変更不可、網掛けの部分は未実装、「0」として読み出し

- Note 1: bit 7 と bit 6 は、ユーザ ソフトウェアまたは POR によってクリアされます。  
 2: PC の bit 21 はシリアル プログラミング モードでのみ使えます。  
 3: リセット値は、2 段階起動が有効な場合が「0」で、無効な場合が「1」です。  
 4: MSSP モジュールが I<sup>2</sup>C™ スレーブモードで動作時のビットの代替名と定義です。  
 5: これらのビット / レジスタを利用できるのは 100 ピンデバイスのみです。それ以外のデバイスには実装されておらず、「0」として読み出されます。表示してあるリセット値は、100 ピンデバイスの場合のみです。  
 6: これらのビット / レジスタを使えるのは 80 ピンおよび 100 ピンのデバイスのみです。64 ピンデバイスでは未実装で、「0」として読み出されます。リセット時の値は 100 ピンデバイスの場合です。  
 7: マイクロコントローラ モードでは、このレジスタのビットは書き込み不可で「0」として読み出されます。  
 8: PLLEN を使えるのはオシレータモードに ECPLL または HSPLL を選択した場合のみです。それ以外では「0」として読み出されます。  
 9: 100 ピンデバイスにのみ実装されており、使えるのはマイクロコントローラ モード時のみです。

# PIC18F97J60 ファミリ

表 6-5: レジスタファイルのまとめ (PIC18F97J60 ファミリ) (続き)

SFR 名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR、BOR 時の値	詳細 ページ
ECCP2AS	ECCP2ASE	ECCP2AS2	ECCP2AS1	ECCP2AS0	PSS2AC1	PSS2AC0	PSS2BD1	PSS2BD0	0000 0000	73, 212
ECCP2DEL	P2RSEN	P2DC6	P2DC5	P2DC4	P2DC3	P2DC2	P2DC1	P2DC0	0000 0000	73, 211
SSP2BUF	MSSP2 受信バッファ / 送信レジスタ								xxxx xxxx	73, 279
SSP2ADD	MSSP2 アドレスレジスタ (I <sup>2</sup> C™ スレーブモード)、MSSP2 baud レート再読み込みレジスタ (I <sup>2</sup> C マスタモード)								0000 0000	73, 279
SSP2STAT	SMP	CKE	D $\bar{A}$	P	S	R $\bar{W}$	UA	BF	0000 0000	73, 270
SSP2CON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	73, 271, 281
SSP2CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	73, 282
	GCEN	ACKSTAT	ADMSK5 <sup>(4)</sup>	ADMSK4 <sup>(4)</sup>	ADMSK3 <sup>(4)</sup>	ADMSK2 <sup>(4)</sup>	ADMSK1 <sup>(4)</sup>	SEN		
EDATA	Ethernet 送信 / 受信バッファレジスタ (EDATA<7:0>)								xxxx xxxx	73, 223
EIR	-	PKTIF	DMAIF	LINKIF	TXIF	-	TXERIF	RXERIF	-000 0-00	73, 241
ECON2	AUTOINC	PKTDEC	ETHEN	-	-	-	-	-	100- ----	73, 228
ESTAT	-	BUFER	-	r	-	RXBUSY	TXABRT	PHYRDY	-0-0 -000	73, 228
EIE	-	PKTIE	DMAIE	LINKIE	TXIE	-	TXERIE	RXERIE	-000 0-00	73, 240
EDMACSH	DMA チェックサム レジスタ上位バイト								0000 0000	73, 265
EDMACSL	DMA チェックサム レジスタ下位バイト								0000 0000	73, 265
EDMADSTH	-	-	-	DMA デスティネーション レジスタ上位バイト				---	0000	73, 265
EDMADSTL	DMA デスティネーション レジスタ下位バイト								0000 0000	73, 265
EDMANDH	-	-	-	DMA 終了レジスタ上位バイト				---	0000	73, 265
EDMANDL	DMA 終了レジスタ下位バイト								0000 0000	73, 265
EDMASTH	-	-	-	DMA 開始レジスタ上位バイト				---	0000	73, 265
EDMASTL	DMA 開始レジスタ下位バイト								0000 0000	73, 265
ERXWRPTH	-	-	-	受信バッファ書き込みポインタ上位バイト				---	0000	73, 225
ERXWRPTL	受信バッファ書き込みポインタ下位バイト								0000 0000	73, 225
ERXRDPH	-	-	-	受信バッファ読み出しポインタ上位バイト				---	0101	73, 225
ERXRPTL	受信バッファ読み出しポインタ下位バイト								1111 1010	73, 225
ERXNDH	-	-	-	受信終了レジスタ上位バイト				---	1111	73, 225
ERXNDL	受信終了レジスタ下位バイト								1111 1111	73, 225
ERXSTH	-	-	-	受信開始レジスタ上位バイト				---	0101	73, 225
ERXSTL	受信開始レジスタ下位バイト								1111 1010	74, 225
ETXNDH	-	-	-	送信終了レジスタ上位バイト				---	0000	74, 226
ETXNDL	送信終了レジスタ下位バイト								0000 0000	74, 226
ETXSTH	-	-	-	送信開始レジスタ上位バイト				---	0000	74, 226
ETXSTL	送信開始レジスタ下位バイト								0000 0000	74, 226
EWRPTH	-	-	-	バッファ書き込みポインタ上位バイト				---	0000	74, 223
EWRPTL	バッファ書き込みポインタ下位バイト								0000 0000	74, 223
EPKTCNT	Ethernet パケットカウント レジスタ								0000 0000	74, 252
ERXFCN	UCEN	ANDOR	CRGEN	PMEN	MPEN	HTEN	MCEN	BCEN	1010 0001	74, 260
EPMOH	-	-	-	パターンマッチ オフセット レジスタ上位バイト				---	0000	74, 263
EPMOL	パターンマッチ オフセット レジスタ下位バイト								0000 0000	74, 263
EPMCSH	パターンマッチ チェックサム レジスタ上位バイト								0000 0000	74, 263

凡例: x = 未知, u = 不変, - = 未実装、「0」として読み出し、q = 条件による、r = 予約済みビットのため変更不可、網掛けの部分は未実装、「0」として読み出し

- Note 1:** bit 7 と bit 6 は、ユーザ ソフトウェアまたは POR によってクリアされます。
- 2:** PC の bit 21 はシリアル プログラミング モードでのみ使えます。
- 3:** リセット値は、2 段階起動が有効な場合が「0」で、無効な場合が「1」です。
- 4:** MSSP モジュールが I<sup>2</sup>C™ スレーブモードで動作時のビットの代替名と定義です。
- 5:** これらのビット / レジスタを利用できるのは 100 ピンデバイスのみです。それ以外のデバイスには実装されておらず、「0」として読み出されます。表示してあるリセット値は、100 ピンデバイスの場合のみです。
- 6:** これらのビット / レジスタを使うのは 80 ピンおよび 100 ピンのデバイスのみです。64 ピンデバイスでは未実装で、「0」として読み出されます。リセット時の値は 100 ピンデバイスの場合です。
- 7:** マイクロコントローラ モードでは、このレジスタのビットは書き込み不可で「0」として読み出されます。
- 8:** PLEN を使えるのはオシレータモードに ECPLL または HSPLL を選択した場合のみです。それ以外では「0」として読み出されます。
- 9:** 100 ピンデバイスにのみ実装されており、使えるのはマイクロコントローラ モード時のみです。

# PIC18F97J60 ファミリ

表 6-5: レジスタファイルのまとめ (PIC18F97J60 ファミリ) (続き)

SFR 名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR、BOR 時の値	詳細 ページ
EPMCSL	パターンマッチ チェックサム レジスタ下位バイト								0000 0000	<a href="#">74, 263</a>
EPMM7	パターンマッチ マスクレジスタ バイト 7								0000 0000	<a href="#">74, 263</a>
EPMM6	パターンマッチ マスクレジスタ バイト 6								0000 0000	<a href="#">74, 263</a>
EPMM5	パターンマッチ マスクレジスタ バイト 5								0000 0000	<a href="#">74, 263</a>
EPMM4	パターンマッチ マスクレジスタ バイト 4								0000 0000	<a href="#">74, 263</a>
EPMM3	パターンマッチ マスクレジスタ バイト 3								0000 0000	<a href="#">74, 263</a>
EPMM2	パターンマッチ マスクレジスタ バイト 2								0000 0000	<a href="#">74, 263</a>
EPMM1	パターンマッチ マスクレジスタ バイト 1								0000 0000	<a href="#">74, 263</a>
EPMM0	パターンマッチ マスクレジスタ バイト 0								0000 0000	<a href="#">74, 263</a>
EHT7	ハッシュテーブル レジスタ バイト 7								0000 0000	<a href="#">74, 259</a>
EHT6	ハッシュテーブル レジスタ バイト 6								0000 0000	<a href="#">74, 259</a>
EHT5	ハッシュテーブル レジスタ バイト 5								0000 0000	<a href="#">74, 259</a>
EHT4	ハッシュテーブル レジスタ バイト 4								0000 0000	<a href="#">74, 259</a>
EHT3	ハッシュテーブル レジスタ バイト 3								0000 0000	<a href="#">74, 259</a>
EHT2	ハッシュテーブル レジスタ バイト 2								0000 0000	<a href="#">74, 259</a>
EHT1	ハッシュテーブル レジスタ バイト 1								0000 0000	<a href="#">74, 259</a>
EHT0	ハッシュテーブル レジスタ バイト 0								0000 0000	<a href="#">74, 259</a>
MIRDL	MII 読み出しデータレジスタ下位バイト								0000 0000	<a href="#">74, 232</a>
MIRDL	MII 読み出しデータレジスタ下位バイト								0000 0000	<a href="#">74, 232</a>
MIWRH	MII 書き込みデータレジスタ上位バイト								0000 0000	<a href="#">74, 232</a>
MIWRL	MII 書き込みデータレジスタ下位バイト								0000 0000	<a href="#">74, 232</a>
MIREGADR	-	-	-	MII アドレスレジスタ				-	---0 0000	<a href="#">74, 232</a>
MICMD	-	-	-	-	-	-	MIISCAN	MIIRD	---- --00	<a href="#">74, 231</a>
MAMXFLH	最大フレーム長レジスタ上位バイト								0000 0110	<a href="#">74, 245</a>
MAMXFLL	最大フレーム長レジスタ下位バイト								0000 0000	<a href="#">74, 245</a>
MAIPGH	-	MAC 不連続パケット間ギャップレジスタ上位バイト							-000 0000	<a href="#">74, 245</a>
MAIPGL	-	MAC 不連続パケット間ギャップレジスタ下位バイト							-000 0000	<a href="#">75, 245</a>
MABBIPG	-	BBIPG6	BBIPG5	BBIPG4	BBIPG3	BBIPG2	BBIPG1	BBIPG0	-000 0000	<a href="#">75, 246</a>
MACON4	-	DEFER	r	r	-	-	r	r	-000 --00	<a href="#">75, 231</a>
MACON3	PADCFG2	PADCFG1	PADCFG0	TXCRCEN	PHDREN	HRFMEN	FRMLNEN	FULDPX	0000 0000	<a href="#">75, 230</a>
MACON1	-	-	-	r	TXPAUS	RXPAUS	PASSALL	MARXEN	---0 0000	<a href="#">75, 229</a>
EPAUSH	一時停止タイム値レジスタ上位バイト								0001 0000	<a href="#">75, 258</a>
EPAUSL	一時停止タイム値レジスタ下位バイト								0000 0000	<a href="#">75, 258</a>
EFLOCON	-	-	-	-	-	r	FCEN1	FCEN0	---- -000	<a href="#">75, 258</a>
MISTAT	-	-	-	-	r	NVALID	SCAN	BUSY	---- 0000	<a href="#">75, 232</a>
MAADR2	MAC アドレスレジスタ バイト 2 (MAADR<39:32>), OUI バイト 2								0000 0000	<a href="#">75, 245</a>
MAADR1	MAC アドレスレジスタ バイト 1 (MAADR<47:40>), OUI バイト 1								0000 0000	<a href="#">75, 245</a>
MAADR4	MAC アドレスレジスタ バイト 4 (MAADR<23:16>)								0000 0000	<a href="#">75, 245</a>
MAADR3	MAC アドレスレジスタ バイト 3 (MAADR<31:24>), OUI バイト 3								0000 0000	<a href="#">75, 245</a>
MAADR6	MAC アドレスレジスタ バイト 6 (MAADR<7:0>)								0000 0000	<a href="#">75, 245</a>
MAADR5	MAC アドレスレジスタ バイト 5 (MAADR<15:8>)								0000 0000	<a href="#">75, 245</a>

凡例: x = 未知、u = 不変、- = 未実装、「0」として読み出し、q = 条件による、r = 予約済みビットのため変更不可、網掛けの部分は未実装、「0」として読み出し

- Note 1: bit 7 と bit 6 は、ユーザ ソフトウェアまたは POR によってクリアされます。
- 2: PC の bit 21 はシリアル プログラミング モードでのみ使えます。
- 3: リセット値は、2 段階起動が有効な場合が「0」で、無効な場合が「1」です。
- 4: MSSP モジュールが I<sup>2</sup>C™ スレーブモードで動作時のビットの代替名と定義です。
- 5: これらのビット/レジスタを利用できるのは 100 ピンデバイスのみです。それ以外のデバイスには実装されておらず、「0」として読み出されます。表示してあるリセット値は、100 ピンデバイスの場合のみです。
- 6: これらのビット/レジスタを使えるのは 80 ピンおよび 100 ピンのデバイスのみです。64 ピンデバイスでは未実装で、「0」として読み出されます。リセット時の値は 100 ピンデバイスの場合です。
- 7: マイクロコントローラ モードでは、このレジスタのビットは書き込み不可で「0」として読み出されます。
- 8: PLLLEN を使えるのはオシレータモードに ECPLL または HSPLL を選択した場合のみです。それ以外では「0」として読み出されます。
- 9: 100 ピンデバイスにのみ実装されており、使えるのはマイクロコントローラ モード時のみです。



## 6.3.6 STATUS レジスタ

レジスタ 6-3 に示す STATUS レジスタは、ALU による算術演算の状態を格納します。STATUS レジスタは、他の全てのレジスタと同様に、任意の命令のオペランドとすることができます。STATUS レジスタが Z、DC、C、OV、N のいずれかのビットに影響を及ぼす命令の結果格納先である場合、これら 5 つのビットへは書き込みできません。

これらのビットはデバイスのロジックに従ってセットまたはクリアされます。従って、STATUS レジスタを結果格納先とする命令を実行した場合、意図した結果とならない場合があります。例えば、CLRf STATUS

命令を実行すると Z ビットがセットされますが、他のビットは変化しません。この場合、STATUS レジスタは「000u u1uu」として読み出されます。従って、STATUS レジスタを変更する際は、Z、C、DC、OV、N ビットに影響を与えない BCF、BSF、SWAPF、MOVFF、MOVWF 命令のみを使う事を推奨します。

ステータスビットに影響を与えないその他の命令については、表 26-2 と表 26-3 の命令セットのまとめを参照してください。

**Note:** 減算では、C ビットがポロー、DC ビットがデジットポロービットとして動作します。

レジスタ 6-3: STATUS レジスタ

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	N	OV	Z	DC <sup>(1)</sup>	C <sup>(2)</sup>
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

bit 7-5      **未実装:** 「0」として読み出し

bit 4      **N:** 負号ビット

このビットは符号付き算術演算に使われます (2 の補数)。結果が負 (ALU MSb = 1) であるかどうかを示します。

1 = 結果が負  
 0 = 結果が正

bit 3      **OV:** オーバーフロー ビット

このビットは符号付き算術演算に使われます (2 の補数)。結果が 7 ビットで表せる値を超え、オーバーフローによって符号ビット (結果の bit 7) の状態が変化した事を示します。

1 = 符号付き算術演算においてオーバーフローが発生した (今回実行した算術演算)  
 0 = オーバーフローは発生していない

bit 2      **Z:** ゼロビット

1 = 算術演算または論理演算の結果がゼロである  
 0 = 算術演算または論理演算の結果はゼロではない

bit 1      **DC:** デジットキャリー / ポロービット <sup>(1)</sup>

ADDWF、ADDLW、SUBLW、SUBWF 命令の場合:

1 = 演算結果の下位 4 ビット目からキャリーが発生した  
 0 = 演算結果の下位 4 ビット目からキャリーは発生していない

bit 0      **C:** キャリー / ポロービット <sup>(2)</sup>

ADDWF、ADDLW、SUBLW、SUBWF 命令の場合:

1 = 演算結果の最上位ビットからキャリーが発生した  
 0 = 演算結果の最上位ビットからキャリーは発生していない

**Note 1:** ポローの場合、極性は逆です。減算は、2 番目のオペランドの 2 の補数を加算する事によって実行されます。ローテート (RRF、RLF) 命令の場合、このビットにはソースレジスタの bit 4 または bit 3 のいずれかが読み込まれます。

**2:** ポローの場合、極性は逆です。減算は、2 番目のオペランドの 2 の補数を加算する事によって実行されます。ローテート (RRF、RLF) 命令の場合、このビットにはソースレジスタの上位ビットまたは下位ビットのいずれかが読み込まれます。

# PIC18F97J60 ファミリ

## 6.4 データ アドレッシング モード

**Note:** PIC18 拡張命令セットを有効にした場合、一部の PIC18 コア命令セットの動作が変化します。詳細は、[セクション 6.6「データメモリと拡張命令セット」](#)を参照してください。

プログラムメモリのアドレスを指定する方法はプログラムカウンタを用いる以外にありませんが、データメモリ空間はいくつかの方法でアドレス指定できます。ほとんどの命令では、アドレッシングモードが固定されています。しかし一部の命令では、どのオペランドを使うか、拡張命令セットを使うかどうかに応じて、最大3つのモードでアドレスを指定できます。

アドレッシングモードには以下の種類があります。

- 含意
- リテラル
- 直接
- 間接

拡張命令セットを有効にした場合 ( コンフィグレーションビット XINST = 1)、さらにインデックス付きリテラルオフセットアドレッシングモードも使えます。その動作については、[セクション 6.6.1「リテラルオフセットを使うインデックス付きアドレッシング」](#)で詳しく説明します。

### 6.4.1 含意およびリテラル アドレッシング

PIC18 の制御命令の多くは、引数を全く必要としません。これらの命令はデバイス全体に影響する操作を実行するか、または暗黙的に1つのレジスタに対して操作を実行します。このようなアドレッシングモードを含意アドレッシングと呼びます。例えば、SLEEP、RESET、DAW等の命令です。

これ以外に、同様に動作しますが、オペコード内で明示的に引数を指定する必要のある命令があります。このように何らかのリテラル値を引数として必要とするものをリテラルアドレッシングモードと呼びます。例えば、Wレジスタ (WREG) に対してリテラル値を加算する ADDLW 命令、リテラル値を移動する MOVLW 命令がこれに該当します。また、20ビットのプログラムメモリアドレスを指定する CALL 命令、GOTO 命令もあります。

### 6.4.2 直接アドレッシング

直接アドレッシングモードでは、演算のソースアドレスと結果の格納先アドレス (またはそのいずれか一方) の、アドレス全体またはその一部をオペコード自体の中で指定します。オプションは命令の引数として指定します。

PIC18 コア命令セットでは、ビット/バイト指向命令が既定値で直接アドレッシングを使います。これらの命令は、いずれも最下位バイトに8ビットのリテラルアドレスを含みます。このアドレスは、データ RAM ([セクション 6.3.3「汎用レジスタファイル」](#)) のいず

れかのバンクにおけるレジスタアドレスか、アクセスバンク ([セクション 6.3.2「アクセスバンク」](#)) 内の位置を表し、命令のデータソースとして参照されます。

アクセス RAM ビット「a」の値に応じて、アドレスの解釈方法が異なります。「a」が「1」の場合、BSR ([セクション 6.3.1「バンクセレクトレジスタ」](#)) の内容とオペコード内のアドレスによって全12ビットのレジスタアドレスが決まります。「a」が「0」の場合、アクセスバンクレジスタを指定したものと見なされます。アクセスRAMを使うアドレッシングを、直接強制アドレッシングモードと呼ぶ事もあります。

MOVFF 等いくつかの命令は、オペコードに全12ビットのアドレス (ソースまたは結果格納先) を含みます。この場合、BSRは無視されます。

演算結果の格納先は、格納先ビット「d」によって決まります。「d」が「1」の場合、結果はソースレジスタに上書きの形で書き戻されます。「d」が「0」の場合、結果は W レジスタに格納されます。引数に「d」を持たない命令の結果格納先は命令ごとに暗黙的に決まっています。その場合の結果格納先は演算対象のレジスタか W レジスタのどちらかです。

### 6.4.3 間接アドレッシング

間接アドレッシングモードでは、命令内で固定アドレスを指定する事なく、データメモリ内の指定した位置にアクセスできます。このアクセスは、ファイルセレクトレジスタ (FSR) を、読み書き対象のメモリ位置に対するポインタとして使う事で実現します。FSR 自体も特殊機能レジスタとしてRAM内に存在するため、プログラム制御のもと直接操作できます。このような特長から FSR はデータメモリ内にテーブルや配列等のデータ構造を実装する時に非常に便利です。

間接アドレッシング用のレジスタとして間接ファイルオペランド (INDF) も実装されています。このオペランドを使って、ポインタ値の自動インクリメント、自動デクリメント、別の値によるオフセット等の操作を自動で行えます。この機能により、ループを使った効率の良いコードを作成できます。[例 6-5](#)に、RAM バンク全体をクリアするサンプルコードを示します。また、データメモリ内でプログラムメモリに対するインデックス付きアドレッシングまたはその他のスタックポインタ操作も実行できます。

#### 例 6-5: 間接アドレッシングによる RAM (バンク 1) のクリア

```
LFSRFSR0, 100h;
NEXT      CLRFPSTINC0; Clear INDF
          ; register then
          ; inc pointer
          BTFSSFSR0H, 1; All done with
          ; Bank1?
          BRA NEXT   ; NO, clear next
CONTINUE  ; YES, continue
```

## 6.4.3.1 FSR レジスタと INDF オペランド

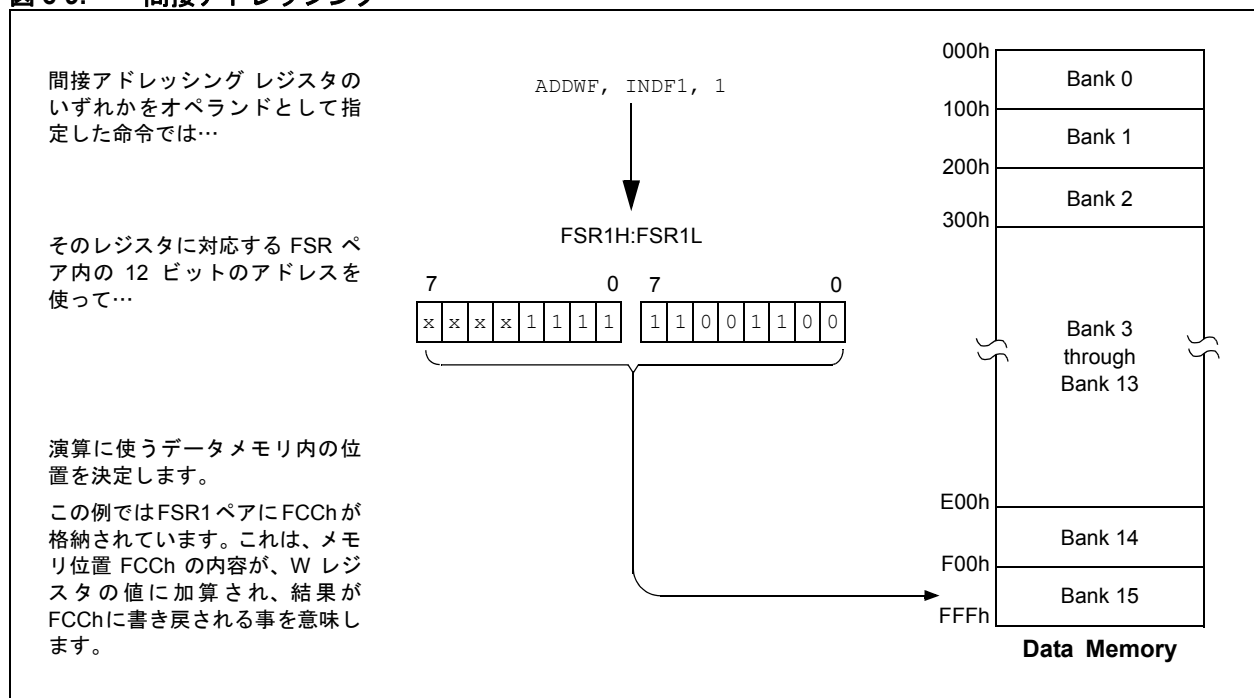
間接アドレッシングの中心となるのが、3つのレジスタセット、FSR0、FSR1、FSR2 です。各セットは、1対の8ビットレジスタ FSRnH と FSRnL から成ります。FSRnH レジスタの上位4ビットは使いません。このため、FSR の各ペアは 12 ビットの値を格納します。この 12 ビットの FSR 値で、データメモリのアドレス全範囲をリニアに指定します。これらの FSR レジスタペアは、データメモリ位置を示すポインタとして機能します。

間接アドレッシングには、間接ファイル オペランド INDF0 ~ INDF2 を使います。これらは「仮想」レジスタと考える事ができます。SFR 空間にマッピングされ

ているものの、物理的には実装されていないためです。特定の INDF レジスタに対する読み書きは、実際には対応する FSR レジスタペアへのアクセスです。例えば、INDF1 からの読み出しの場合、データは FSR1H:FSR1L が示すアドレスから読み出されます。オペランドとして INDF レジスタを指定した命令は、実際には対応する FSR の内容を命令対象へのポインタとして使います。INDF オペランドは、ポインタを使いやすくする機能です。

間接アドレッシングでは、アドレスの全 12 ビットを使うため、データ RAM のバンク指定は不要です。従って、BSR の内容とアクセス RAM ビットはアドレスの決定に一切影響しません。

図 6-9: 間接アドレッシング



# PIC18F97J60 ファミリ

## 6.4.3.2 FSR レジスタと POSTINC、POSTDEC、PREINC、PLUSW

INDF オペランドの他に、各 FSR レジスタペアには 4 つの間接オペランドがあります。これらは INDF と同じ「仮想」レジスタであるため、間接的に読み書きする事はできません。これらのレジスタへのアクセスは実際には関連する FSR レジスタペアへのアクセスですが、同時に、格納された値に対して以下のような操作を実行します。

- POSTDEC: FSR の値にアクセスした後、その値を自動的に「1」デクリメントする
- POSTINC: FSR の値にアクセスした後、その値を自動的に「1」インクリメントする
- PREINC: FSR の値を「1」インクリメントした値を演算で使う
- PLUSW: W レジスタの符号付きの値 (-128 ~ 127) を FSR に加算した値を演算で使う

INDF レジスタへのアクセスでは、FSR レジスタの値を使いますが、FSR レジスタの値自体は変更しません。同様に、PLUSW レジスタにアクセスした場合も FSR の値に W レジスタの値がオフセットとして加算されますが、どちらのレジスタの値も演算で変化する事はありません。しかし、これら以外の仮想レジスタへのアクセスは、FSR レジスタの値を変更します。

POSTDEC、POSTINC、PREINC を使った FSR への操作は、レジスタペア全体に作用します。つまり、FSRnL レジスタの FFh から 00h へのロールオーバーによって、FSRnH レジスタへのキャリーオーバーが発生するという事です。一方これらの操作の結果は、STATUS レジスタのいずれのフラグ (Z、N、OV 等) の値にも影響を与えません。

PLUSW レジスタは、データメモリ空間内にインデックス付きアドレッシングのデータ構造を実装する時に使えます。W レジスタの値を操作する事で、ポインタアドレスから一定の値だけオフセットしたアドレスにアクセスできます。アプリケーションによっては、この機能でデータメモリ内にソフトウェア スタック等の強力なプログラム制御構造を実装できます。

## 6.4.3.3 FSR による FSR に対する操作

他の FSR または仮想レジスタを対象として間接アドレッシングの動作を実行すると特殊な挙動を示します。例えば、FSR を使って仮想レジスタのアドレスを指定すると失敗します。例えば、INDF1 のアドレス FE7h が FSR0H:FSR0L ペアに格納されている場合を考えます。この場合、INDF0 をオペランドとして INDF1 の値を読み出そうとすると、00h が返されます。INDF0 をオペランドとして INDF1 に書き込もうとすると、NOP となります。

一方、仮想レジスタによって FSR ペアに書き込むと、意図した結果が得られない場合があります。このような場合、値は FSR ペアに書き込まれますが、インクリメント / デクリメントは実行されません。従って、INDF2 または POSTDEC2 に書き込むと、同じ値が FSR2H:FSR2L ペアに書き込まれます。

FSR は SFR 空間にマッピングされた物理レジスタであるため、全ての直接演算で操作できます。従って、ユーザはこれらのレジスタを使うにあたって、特にコードに間接アドレッシングを採用する場合、十分な注意を払う必要があります。

同様に、通常は全ての SFR を間接アドレッシングによって操作できます。このため、SFR による設定を誤って変更してデバイス動作に影響を与えないように注意する必要があります。

## 6.5 プログラムメモリと拡張命令セット

拡張命令セットを使ってもプログラムメモリの動作には影響しません。

拡張命令セットを有効にすると、既存の PIC18 命令セットに新たに 5 個の 2 ワード命令 (ADDFSR、CALLW、MOVSF、MOVSS、SUBFSR) が追加されます。これらの命令の実行については、[セクション 6.2.4「2 ワード命令」](#)で説明します。

## 6.6 データメモリと拡張命令セット

PIC18 拡張命令セットを有効にすると ( コンフィグレーション ビット XINST = 1)、データメモリとそのアドレス方法の一部が大きく変わります。具体的には、PIC18 コア命令の多くでアクセスバンクの使い方が変わります。これは、データメモリ空間に新しいアドレッシング モードが追加されるためです。このモードでは、FSR2 と関連するオペランドを使った間接アドレッシングの挙動も変わります。

変わらない点を把握しておく事も重要です。まず、データメモリ空間のサイズと、リニアアドレッシングである事は変わりません。SFR のマップも変わりません。PIC18 コア命令セットは、引き続き直接および間接アドレッシング モードの両方で動作します。含意およびリテラル命令には一切変更はありません。FSR0 と FSR1 による間接アドレッシングも以前のままです。

## 6.6.1 リテラル オフセットを使う インデックス付きアドレッシング

PIC18 拡張命令セットを有効にすると、FSR2 レジスタペアと関連するファイル オペランドを使った間接アドレッシングの挙動が変化します。適切な条件のもとでは、アクセスバンクを使う命令、つまりほとんどのビット / バイト指向命令において、オフセット値を命令に含めるインデックス付きアドレッシング方式を使えます。この特殊なアドレッシングモードは、リテラル オフセットを使うインデックス付きアドレッシング、またはインデックス付きリテラル オフセットモードと呼ばれます。

拡張命令セットを使う場合、このアドレッシングモードでは以下の2つの条件を満たす必要があります。

- 必ずアクセスバンクを使う事 (「a」= 0)
- ファイルアドレス引数が 5Fh 以下である事

これらの条件のもとでは、命令のファイルアドレス引数の解釈は、アドレスの下位バイト (直接アドレッシングにおいて BSR と共に使う値) でも、アクセスバンク内の 8 ビットアドレスでもありません。ファイルアドレスの値は、FSR2 が指定するアドレスポインタに対するオフセット値と解釈されます。つまり、このオフセット値と FSR2 の内容を加算したものが、演算の対象アドレスになります。

## 6.6.2 インデックス付きリテラルオフセット モードの影響を受ける命令

直接アドレッシングを使用可能な PIC18 コア命令は、いずれもインデックス付きリテラル オフセット アドレッシングモードの影響を受ける可能性があります。これには、全てのバイト / ビット指向の命令、つまり PIC18 標準命令セットのほぼ半数が含まれます。含意またはリテラル アドレッシングモードのみを使う命令は影響を受けません。

また、バイト / ビット指向命令であってもアクセスバンクを使わない場合 (アクセス RAM ビットが「1」)、または 60h 以上のファイルアドレスを含む場合、影響を受けません。これらの条件を満たす命令は、以前と同様に実行されます。図 6-10 に、拡張命令セットを有効にした場合に各アドレッシングモードがどのように変化するかを示します。

インデックス付きリテラル オフセット モードでバイト / ビット指向命令を使う場合、アセンブラ構文の変更にご注意してください。これについては、[セクション 26.2.1 「拡張命令の構文」](#)で詳しく説明します。

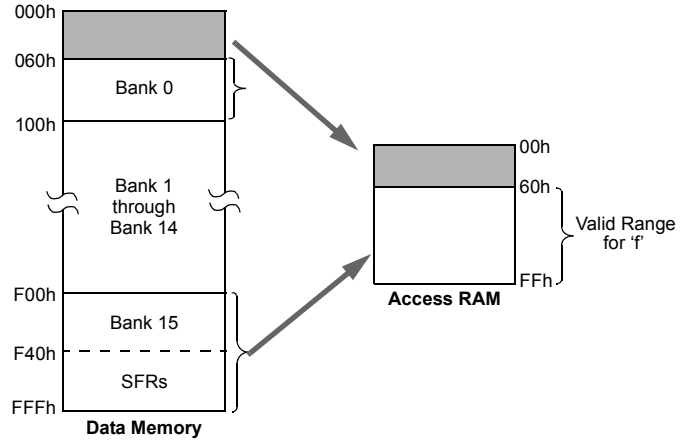
# PIC18F97J60 ファミリ

図 6-10: ビット/バイト指向命令に対するアドレッシングモードの比較 (拡張命令セットが有効な場合)

Example Instruction: `ADDWF, f, d, a` (Opcode: `0010 01da ffff ffff`)

**a = 0 かつ f ≥ 60h の場合 :**

命令は直接強制モードで実行されます。「f」はアクセス RAM 内の 060h ~ FFFh の間のメモリ位置と見なされます。これは、データメモリ内の F60h ~ FFFh (バンク 15) の位置を指定した場合と同じです。060h よりも下位のメモリ位置は、このアドレッシングモードでは使えません。



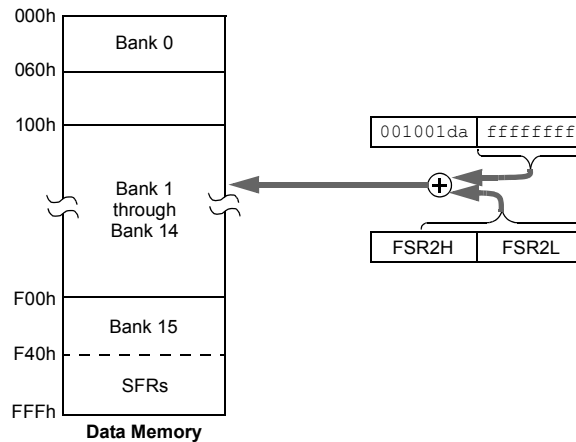
**a = 0 かつ f ≤ 5Fh の場合 :**

命令はインデックス付きリテラル オフセットモードで実行されます。「f」は FSR2 内のアドレス値に対するオフセットと見なされます。両者を加算する事で命令の対象レジスタのアドレスが得られます。アドレスとしては、データメモリ空間内の任意の位置を指定できます。

このモードにおける正しい構文は、以下になる事に注意してください。

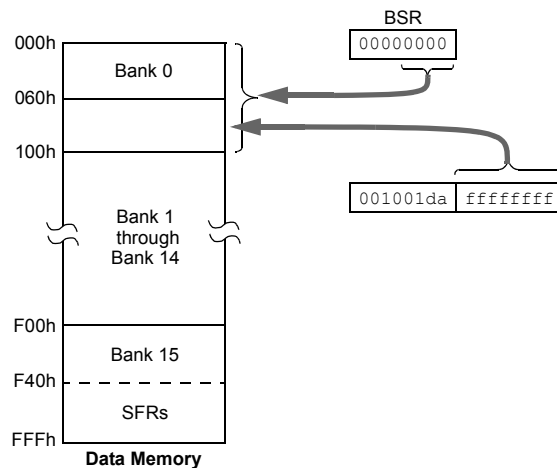
`ADDWF [k], d`

ただし「k」と「f」は同じです。



**a = 1 (f は任意の値の場合) :**

命令は直接モード (直接ロングモードとも呼ばれます) で実行されます。「f」は、データメモリ空間の 16 バンクのうちの1つにおける位置と見なされます。バンクは、バンクセレクトレジスタ (BSR) で指定します。データメモリ空間内に実装された全てのバンク内のアドレスを指定できます。



## 6.6.3 インデックス付きリテラルオフセットモードにおけるアクセスバンクのマッピング

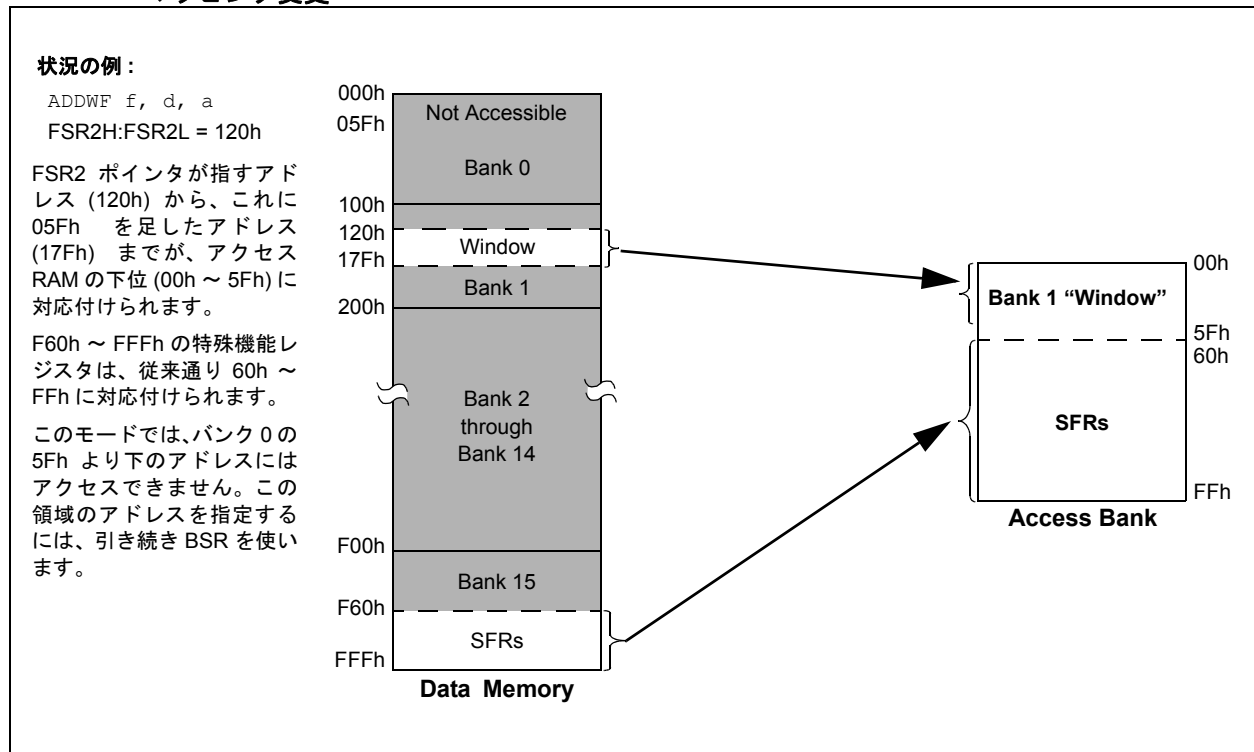
インデックス付きリテラル オフセット アドレッシングモードを使うと、アクセス RAM の下位部分 (00h ~ 5Fh) のマッピングが実質的に変更されます。このモードでは、バンク 0 の先頭部分の内容に限らず、バンク 0 と、データメモリ空間内にユーザが任意に定義した「ウィンドウ」の内容をマッピングします。このウィンドウにマッピングされるデータメモリの開始アドレスは FSR2 の値で定義され、終了アドレスは FSR2 に 95 (5Fh) を加えた値で定義されます。アクセス RAM 内の 5Fh以降のアドレスのマッピングは、**セクション 6.3.2 「アクセスバンク」** で説明した通りです。図 6-11 に、このアドレッシングモードでアクセスバンクのマッピングがどのように変わるかの例を示します。

アクセスバンクのマッピング変更は、インデックス付きリテラル オフセット モードを使う演算のみに適用されます。BSR を使う演算 (アクセス RAM ビットが「1」) には、通常の直接アドレッシングがそのまま適用されます。間接またはインデックス付き演算で明示的に間接ファイルオペランド (FSR2 を含む) を使うものも、通常の間接アドレッシングとしての動作のままです。アクセスバンクを使う命令でも、05Fh より大きいレジスタアドレスを含む場合は直接アドレッシングと通常のアksesバンク マップを使います。

## 6.6.4 インデックス付きリテラルオフセットモードにおける BSR

拡張命令セットを有効にするとアクセスバンクのマッピングが変わりますが、BSR の演算は変わりません。BSR を使ってデータメモリ バンクを切り換える直接アドレッシングは、先に説明した動作の通りです。

図 6-11: インデックス付きリテラル オフセット アドレッシングにおけるアクセスバンクのマッピング変更



# PIC18F97J60 ファミリ

---

NOTES:



## 7.0 フラッシュ プログラムメモリ

フラッシュ プログラムメモリは、通常動作中 VDD の全レンジで読み出し、書き込み、消去動作が可能です。プログラムメモリからの読み出しは、1 バイト単位で実行されます。プログラムメモリへの書き込みは、64 バイトのブロック単位で実行されます。プログラムメモリの消去は、1024 バイトのブロック単位で実行されます。バルク消去動作はユーザコードからは実行できません。

プログラムメモリに対する書き込みまたは消去動作が完了するまで、命令のフェッチは中止されます。書き込みまたは消去中はプログラムメモリにアクセスできないため、コードを実行できません。プログラムメモリの書き込みと消去は、内部のプログラミング タイマによって終了します。

プログラムメモリに書き込む値は、有効な命令である必要はありません。無効な命令を含むプログラムメモリ位置の実行は、NOP になります。

## 7.1 テーブル読み書き

プログラムメモリの読み書き時に、プロセッサがプログラムメモリ空間とデータRAM間でデータバイトを移動する方法には、以下の2つの動作があります。

- テーブル読み出し (TBLRD)
- テーブル書き込み (TBLWT)

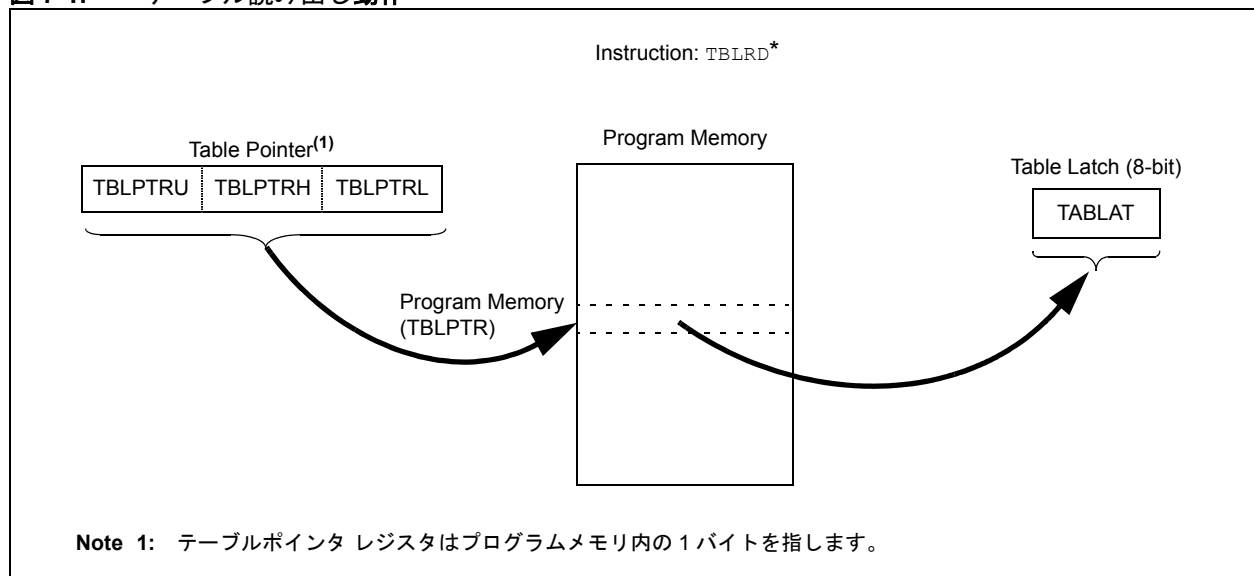
プログラムメモリ空間は 16 ビット幅であるのに対し、データ RAM 空間は 8 ビット幅です。テーブルに対する読み書きは、8 ビットのレジスタ (TABLAT) を介して、これら2つのメモリ空間の間でデータを移動します。

テーブル読み出し動作はプログラムメモリからデータを取得し、データ RAM 空間に格納します。図 7-1 に、プログラムメモリとデータ RAM 間のテーブル読み出し動作を示します。

テーブル書き込み動作は、データメモリ空間からのデータをプログラムメモリ内の保持レジスタに格納します。保持レジスタの内容をプログラムメモリに書き込む手順の詳細は、[セクション 7.5「フラッシュ プログラムメモリへの書き込み」](#)で説明します。図 7-2 に、プログラムメモリとデータRAM間のテーブル書き込み動作を示します。

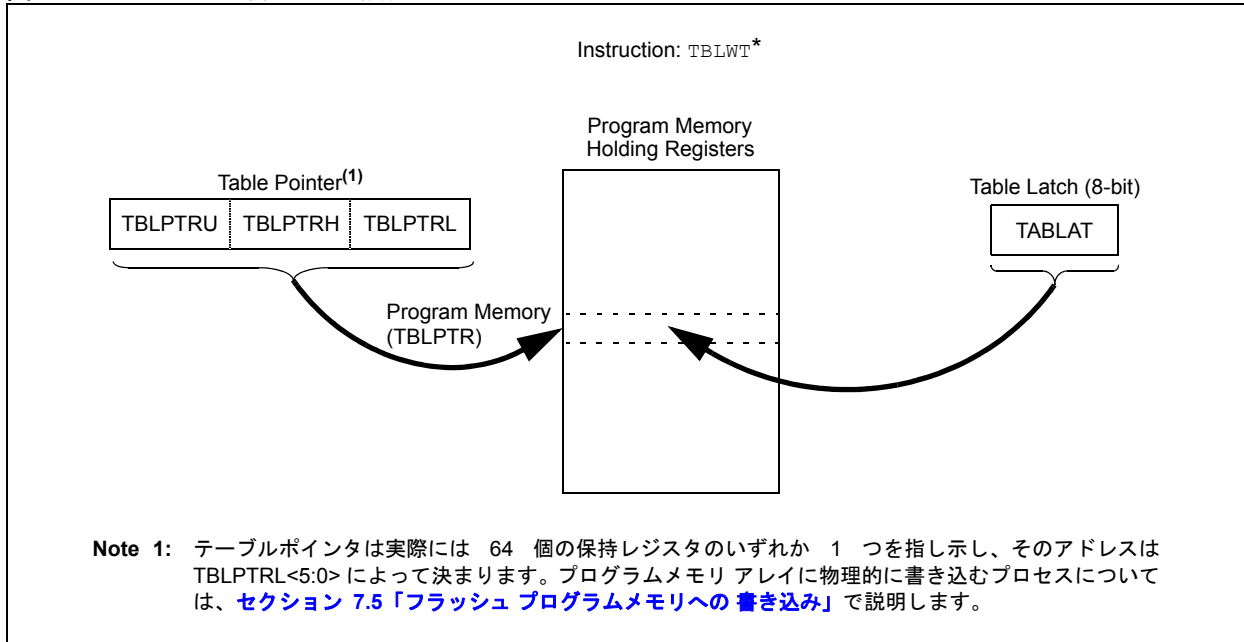
テーブル動作はバイト単位で実行されます。テーブルブロックにプログラム命令ではなくデータを格納する場合、ワード境界でアライメントする必要はありません。従って、テーブルブロックは任意のバイトアドレスから開始し、終了する事ができます。テーブル書き込みによって実行可能コードをプログラムメモリに書き込む場合、プログラム命令をワード境界でアライメントする必要があります。

図 7-1: テーブル読み出し動作



# PIC18F97J60 ファミリ

図 7-2: テーブル書き込み動作



## 7.2 制御レジスタ

TBLRD および TBLWT 命令では、以下を含む制御レジスタを使います。

- EECON1 レジスタ
- EECON2 レジスタ
- TABLAT レジスタ
- TBLPTR レジスタ

### 7.2.1 EECON1 および EECON2 レジスタ

EECON1 レジスタ ([レジスタ 7-1](#)) は、メモリアクセスを制御するレジスタです。EECON2 レジスタは物理レジスタではなく、メモリの書き込みおよび消去シーケンスでのみ使います。EECON2 を読み出すと、常に「0」が返されます。

FREE ビットをセットすると、プログラムメモリに対する消去動作が許可されます。FREE ビットをセットした後の最初の WR コマンドで消去動作が開始します。FREE をクリアすると、書き込みだけが有効になります。

WREN ビットをセットすると、書き込み動作が許可されます。WREN ビットは、電源投入時にクリアされます。WRERR ビットは WR ビットをセットするとハードウェアによってセットされます。その後、内部プログラミング タイマがタイムアウトになり書き込み動作が完了するとクリアされます。

**Note:** 通常動作中に WRERR から「1」が読み出された場合、書き込み動作がリセットによって不完全なまま中止されたか、不適切な方法で実行された事を示している可能性があります。

書き込み動作は、制御ビット WR によって開始します。このビットはソフトウェアではセットのみ可能でクリアはできません。書き込み動作が完了するとハードウェアによってクリアされます。

# PIC18F97J60 ファミリ

レジスタ 7-1: EECON1: EEPROM 制御レジスタ 1

U-0	U-0	U-0	R/W-0	R/W-x	R/W-0	R/S-0	U-0
—	—	—	FREE	WRERR	WREN	WR	—
bit 7							bit 0

<b>凡例:</b>	S = セット可能ビット		
R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し	
-n = POR 時の値	「1」= ビットはセット	「0」= ビットはクリア	x = ビットは未知

bit 7-5 **未実装:** 「0」として読み出し

bit 4 **FREE:** フラッシュ行消去イネーブルビット

- 1 = TBLPTRによってアドレス指定されたプログラムメモリ行を次の WR コマンドで消去する (消去動作が完了するとクリアされる)
- 0 = 書き込み動作のみ実行する

bit 3 **WRERR:** フラッシュ プログラムエラー フラグビット

- 1 = 書き込み動作が不完全なまま中止された (通常動作時の内部タイマによるプログラミング中にいずれかのリセット動作が実行されたか、不適切な書き込みを実行しようとした)
- 0 = 書き込み動作が完了した

bit 2 **WREN:** フラッシュ プログラム書き込みイネーブルビット

- 1 = フラッシュ プログラムメモリへの書き込みサイクルを許可する
- 0 = フラッシュ プログラムメモリへの書き込みサイクルを禁止する

bit 1 **WR:** 書き込み制御ビット

- 1 = プログラムメモリの消去サイクルまたは書き込みサイクルを開始する (動作は内部タイマによって制御され、ビットは書き込み完了時にハードウェアでクリアされる。WR ビットはソフトウェアではセットのみ可能で、クリアできない)
- 0 = 書き込みサイクルが完了した

bit 0 **未実装:** 「0」として読み出し

# PIC18F97J60 ファミリ

## 7.2.2 テーブルラッチ レジスタ (tblat)

テーブルラッチ (TABLAT) は、SFR 空間にマッピングされた 8 ビットのレジスタです。テーブルラッチ レジスタは、プログラムメモリとデータ RAM 間でデータを転送する時に 8 ビットのデータを保持します。

## 7.2.3 テーブルポインタ レジスタ (TBLPTR)

テーブルポインタ (TBLPTR) レジスタは、プログラムメモリ内の 1 バイトのアドレスを指します。TBLPTR は 3 つの SFR レジスタから構成されます。テーブルポインタ最上位バイト、テーブルポインタ上位バイト、テーブルポインタ下位バイト (TBLPTRU:TBLPTRH:TBLPTRL) の 3 つです。これら 3 つを結合して、22 ビット幅のポインタを構成します。下位 21 ビットによって、最大 2 MB のプログラムメモリ空間のアドレスを指定できます。22 番目のビットは、デバイス ID とコンフィグレーション ビットへのアクセスを可能にします。

テーブルポインタ レジスタ TBLPTR は、TBLRD および TBLWT 命令で使います。これらの命令は、テーブルポインタに対する操作に基づく 4 つの方法のいずれかによって TBLPTR を更新します。表 7-1 にこれらの操作を示します。TBLPTR に対するテーブル操作は、下位 21 ビットに対してのみ影響します。

## 7.2.4 テーブルポインタの境界

TBLPTR はフラッシュ プログラムメモリの読み出し、書き込み、消去に使います。

TBLRD を実行すると、TBLPTR の全 22 ビットに基づいてプログラムメモリのどのバイトを TABLAT に読み出すかを判断します。

TBLWT を実行すると、テーブルポインタ レジスタの下位 6 ビット (TBLPTR<5:0>) に基づいて 64 個のプログラムメモリ保持レジスタのどれに書き込むかを決定します。プログラムメモリに対して内部タイマを使った書き込みを (WR ビットによって) 開始した場合、TBLPTR の上位 15 ビット (TBLPTR<20:6>) に基づいて 64 バイトのプログラムメモリ ブロックのどれに書き込むかを決定します。詳細は、[セクション 7.5「フラッシュ プログラムメモリへの書き込み」](#)を参照してください。

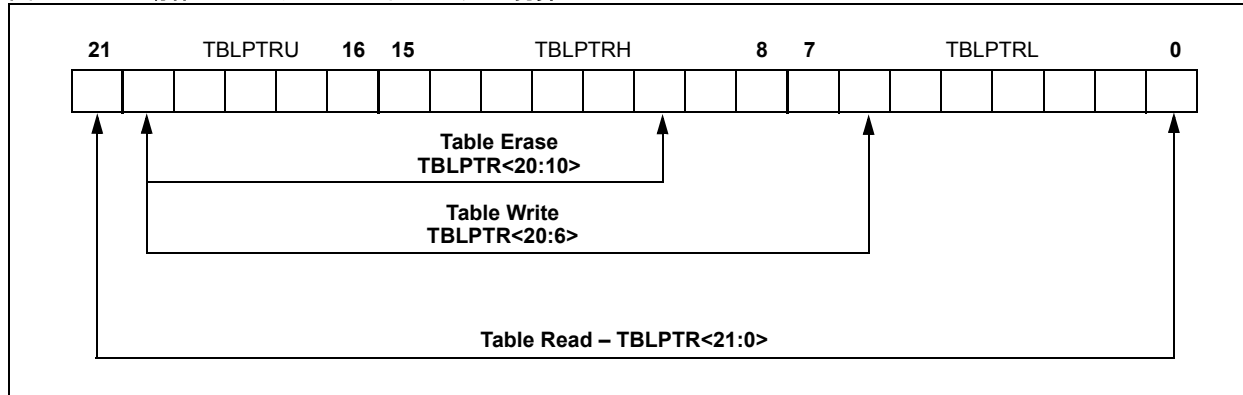
プログラムメモリ消去を実行する場合、消去対象となる 1024 バイトのブロックは、テーブルポインタ レジスタの上位 11 ビット (TBLPTR<20:10>) によって指定されます。下位 10 ビット (TBLPTR<9:0>) は無視されます。

図 7-3 に、フラッシュ プログラムメモリ動作に対応する TBLPTR の境界を示します。

表 7-1: TBLRD および TBLWT 命令に伴うテーブルポインタに対する操作

例	テーブルポインタに対する操作
TBLRD* TBLWT*	TBLPTR は変化しない
TBLRD*+ TBLWT*+	読み書きの後に TBLPTR がインクリメントされる
TBLRD*- TBLWT*-	読み書きの後に TBLPTR がデクリメントされる
TBLRD+* TBLWT+*	読み書きの前に TBLPTR がインクリメントされる

図 7-3: 動作ごとのテーブルポインタの境界



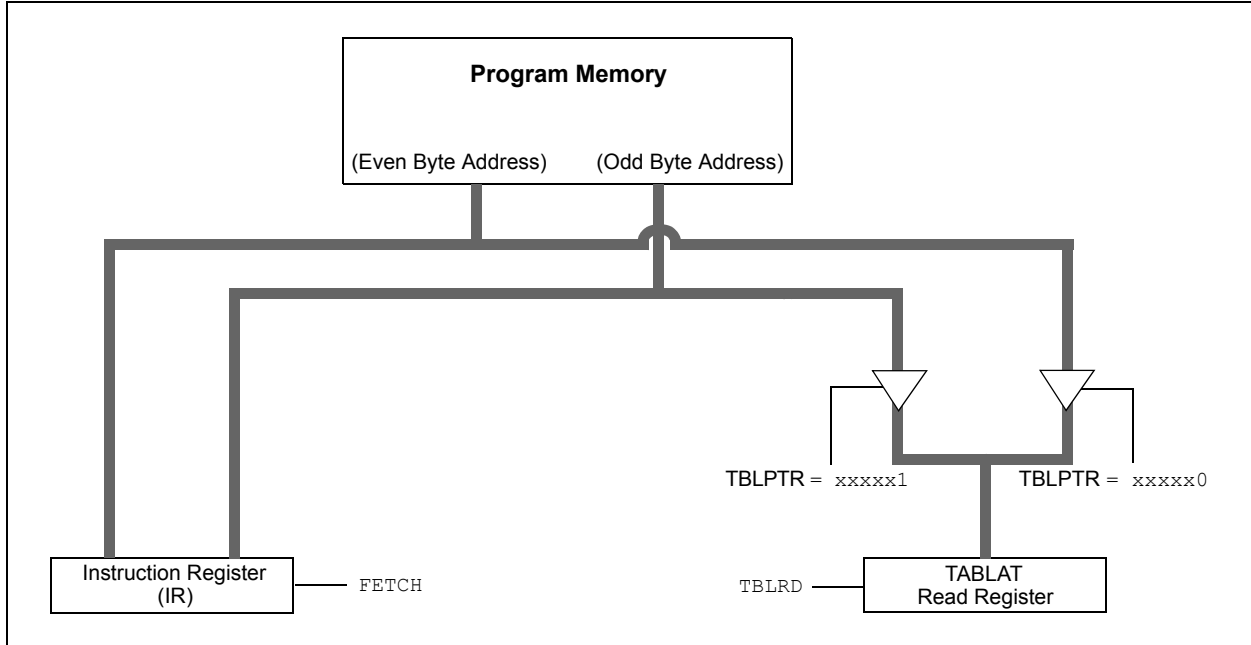
## 7.3 フラッシュ プログラムメモリの読み出し

TBLRD 命令は、プログラムメモリからデータを取得して、データ RAM に格納します。プログラムメモリからのテーブル読み出しは、一度に 1 バイトずつ実行されます。

TBLPTR はプログラム空間内の 1 バイトのアドレスを指します。TBLRD を実行すると、TBLPTR が指すバイトが TABLAT に格納されます。さらに、次のテーブル読み出し動作に備えて、TBLPTR の値を自動的に変更する事ができます。

内部プログラムメモリの内容は通常ワード単位で並んでいます。アドレスの LSb が、ワードの上位バイトと下位バイトを区別します。図 7-4 に、内部プログラムメモリと TABLAT 間のインターフェイスを示します。

図 7-4: フラッシュ プログラムメモリからの読み出し



例 7-1: フラッシュ プログラムメモリ ワードの読み出し

```

MOV LW CODE_ADDR_UPPER      ; Load TBLPTR with the base
MOV WF TBLPTR               ; address of the word
MOV LW CODE_ADDR_HIGH
MOV WF TBLPTRH
MOV LW CODE_ADDR_LOW
MOV WF TBLPTRL

READ_WORD
TBLRD*+                    ; read into TABLAT and increment
MOV F TABLAT, W            ; get data
MOV WF WORD_EVEN
TBLRD*+                    ; read into TABLAT and increment
MOV F TABLAT, W            ; get data
MOV F WORD_ODD
    
```

# PIC18F97J60 ファミリ

## 7.4 フラッシュ プログラムメモリの消去

消去ブロックの最小単位は 1024 バイトです。これよりも大きなプログラムメモリ ブロックをバルク消去するには、外部プログラマを使うか、ICSP 制御を利用する必要があります。フラッシュアレイは、ワード単位の消去に対応していません。

マイクロコントローラから消去シーケンスを開始する場合、プログラムメモリは 1024 バイトのブロック単位で消去されます。消去するブロックは、TBLPTR<20:10> の 11 ビットで指定します。TBLPTR<9:0> ビットは無視されます。

消去動作は EECON1 レジスタで制御します。書き込み動作を有効にするには、WREN ビットをセットする必要があります。消去動作を選択するには、FREE ビットをセットします。

誤操作を防止するため、EECON2 に対する書き込み開始シーケンスを実行する必要があります。

内部フラッシュを消去するには、ロングライトが必要です。ロングライトサイクル中は、命令の実行が一時停止します。ロングライトは、内部のプログラミングタイマによって終了します。消去時間は内蔵タイマで制御します。書き込み / 消去用の電圧は、デバイスの動作電圧レンジのほとんどで動作可能な内蔵チャージポンプによって生成されます。上限値と下限値については、パラメータ D132B (VPEW) を参照してください。

### 例 7-2: フラッシュ プログラムメモリの行消去

	MOVLW	CODE_ADDR_UPPER		; load TBLPTR with the base
	MOVWF	TBLPTRU		; address of the memory block
	MOVLW	CODE_ADDR_HIGH		
	MOVWF	TBLPTRH		
	MOVLW	CODE_ADDR_LOW		
	MOVWF	TBLPTRL		
	ERASE_ROW			
	BSF	EECON1, WREN		; enable write to memory
	BSF	EECON1, FREE		; enable Row Erase operation
	BCF	INTCON, GIE		; disable interrupts
<b>Required</b>	MOVLW	55h		
<b>Sequence</b>	MOVWF	EECON2		; write 55h
	MOVLW	0AAh		
	MOVWF	EECON2		; write 0AAh
	BSF	EECON1, WR		; start erase (CPU stall)
	BSF	INTCON, GIE		; re-enable interrupts

### 7.4.1 フラッシュ プログラムメモリの消去シーケンス

内部プログラムメモリのブロック消去動作のシーケンスは以下の通りです。

1. 消去する行のアドレスをテーブルポインタ レジスタに格納する。
2. EECON1 レジスタを以下の通り消去動作に設定する。
  - WREN ビットをセットして、書き込みを有効にする。
  - FREE ビットをセットして、消去を有効にする。
3. 割り込みを無効にする。
4. EECON2 に 55h を書き込む。
5. EECON2 に 0AAh を書き込む。
6. WR ビットをセットする。これで、ブロック消去サイクルが開始する。
7. 消去を実行中、CPU がストールする。
8. 割り込みを再び有効にする。

## 7.5 フラッシュ プログラムメモリへの書き込み

書き込みブロックの最小単位は32ワードつまり64バイトです。ワードまたはバイト単位のプログラミングには対応していません。

フラッシュメモリのプログラミングに必要な保持レジスタへのデータ読み込みには、デバイス内部でテーブル書き込みを使います。プログラミング時のテーブル書き込みでは、64個の保持レジスタを使います。

テーブルラッチ (TABLAT) には1バイトしか格納できないため、書き込み動作のたびに TBLWT 命令を64回実行する必要があります。これらのテーブル書き込みは保持レジスタに対する書き込みであるため、全てショートライトです。64個の保持レジスタへの書き込みが完了したら、EECON1 レジスタに書き込みを実行してロングライトの書き込みを開始する必要があります。

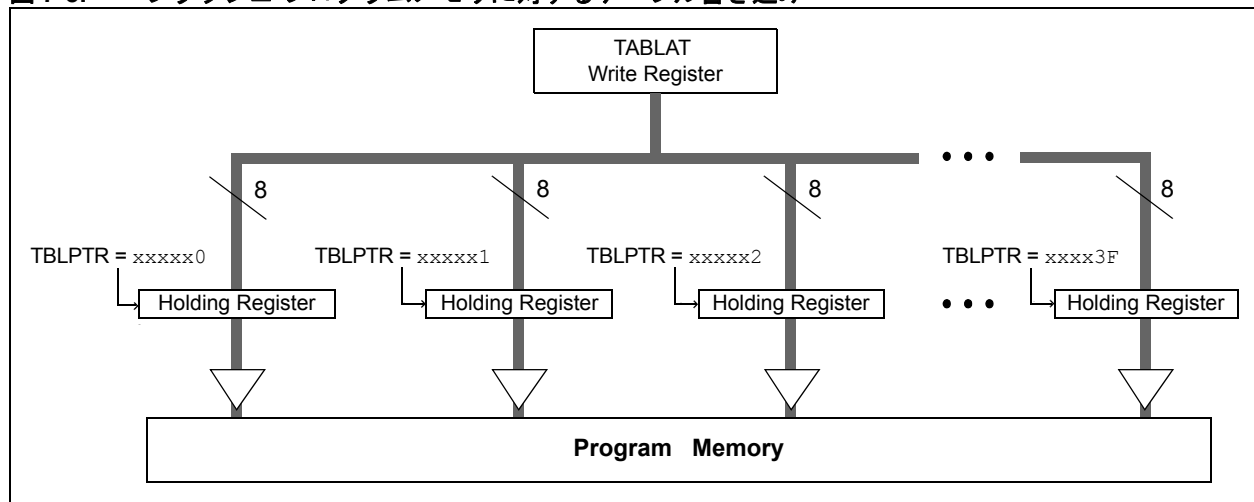
内部フラッシュのプログラミングには、ロングライトが必要です。ロングライトサイクル中は、命令の実行が一時停止されます。ロングライトは、内部のプログラミング タイマによって終了します。

書き込み時間は内蔵タイマで制御します。書き込み / 消去用の電圧は、デバイスの動作電圧レンジのほとんどで動作可能な内蔵チャージポンプによって生成されます。上限値と下限値については、パラメータ D132B (VPEW) を参照してください。

**Note 1:** 以前のPIC MCUとは異なり、PIC18F97J60 ファミリは書き込み完了後に保持レジスタをリセットしません。従って、書き込みシーケンスの前には保持レジスタをクリアまたは上書きする必要があります。

**2:** プログラムメモリ セルの書き込み耐性を維持するため、1つのフラッシュバイトに対して消去と消去の間に複数回の書き込みを実行しないでください。ターゲットセルの内容を書き換える前に、ターゲット行に対する行消去またはメモリ全体のバルク消去を実行する必要があります。

図 7-5: フラッシュ プログラムメモリに対するテーブル書き込み



### 7.5.1 フラッシュ プログラムメモリの書き込みシーケンス

内部プログラムメモリに対するプログラミング動作のシーケンスは以下の通りです。

1. 書き込もうとするプログラムメモリ位置が既にプログラミングされている場合、書き込みを実行する前にメモリを消去する必要がある ( [セクション 7.4.1「フラッシュ プログラムメモリの消去シーケンス」](#) 参照 )。
2. 自動インクリメントにより、保持レジスタに64バイトを書き込む。
3. WREN ビットをセットして、バイト書き込みを有効に設定する。
4. 割り込みを無効にする。

5. EECON2 に 55h を書き込む。
6. EECON2 に AAh を書き込む。
7. WR ビットをセットする。これで書き込みサイクルが開始する。
8. 書き込みを実行中、CPU がストールする。
9. 割り込みを再び有効にする。
10. メモリ内容をベリファイする(テーブル読み出し)。

例 7-3 に、必要なコードの例を示します。

**Note:** WR ビットをセットする前に、テーブルポインタのアドレスを保持レジスタ内の該当する64バイトのアドレス範囲内に設定しておく必要があります。

# PIC18F97J60 ファミリ

## 例 7-3: フラッシュ プログラムメモリへの書き込み

```

MOV LW CODE_ADDR_UPPER ; Load TBLPTR with the base
MOV WF TBLPTRU ; address of the memory block
MOV LW CODE_ADDR_HIGH
MOV WF TBLPTRH
MOV LW CODE_ADDR_LOW
MOV WF TBLPTRL

ERASE_BLOCK

BSF EECON1, WREN ; enable write to memory
BSF EECON1, FREE ; enable Row Erase operation
BCF INTCON, GIE ; disable interrupts
MOV LW 55h
MOV WF EECON2 ; write 55h
MOV LW 0AAh
MOV WF EECON2 ; write 0AAh
BSF EECON1, WR ; start erase (CPU stall)
BSF INTCON, GIE ; re-enable interrupts
MOV LW D'16'
MOV WF WRITE_COUNTER ; Need to write 16 blocks of 64 to write
; one erase block of 1024

RESTART_BUFFER

MOV LW D'64'
MOV WF COUNTER
MOV LW BUFFER_ADDR_HIGH ; point to buffer
MOV WF FSR0H
MOV LW BUFFER_ADDR_LOW
MOV WF FSR0L

FILL_BUFFER

... ; read the new data from I2C, SPI,
; PSP, USART, etc.

WRITE_BUFFER

MOV LW D'64 ; number of bytes in holding register
MOV WF COUNTER

WRITE_BYTE_TO_HREGS

MOV FF POSTINC0, WREG ; get low byte of buffer data
MOV WF TABLAT ; present data to table latch
TBLWT+* ; write data, perform a short write
; to internal TBLWT holding register.
DECFSZ COUNTER ; loop until buffers are full
BRA WRITE_BYTE_TO_HREGS

PROGRAM_MEMORY

BSF EECON1, WREN ; enable write to memory
BCF INTCON, GIE ; disable interrupts
MOV LW 55h
Required MOV WF EECON2 ; write 55h
Sequence MOV LW 0AAh
MOV WF EECON2 ; write 0AAh
BSF EECON1, WR ; start program (CPU stall)
BSF INTCON, GIE ; re-enable interrupts
BCF EECON1, WREN ; disable write to memory

DECFSZ WRITE_COUNTER ; done with one write cycle
BRA RESTART_BUFFER ; if not done replacing the erase block

```



# PIC18F97J60 ファミリ

## 7.5.2 書き込みのベリファイ

アプリケーションによっては、メモリに書き込まれた値を元の値と比較して検証する事が望ましい場合があります。例えば、仕様限界近くまで頻繁に書き込みを実行するようなアプリケーションです。

## 7.5.3 書き込み動作の予期せぬ中断

電源遮断や予期しないリセット等、不測のイベントによって書き込みが中止された場合、必要に応じてプログラミングしたメモリ位置のベリファイと再プログラミングを実行してください。MCLR リセット、または通常動作中の WDT タイムアウトによって書き込み動作が中断した場合、必要に応じて WRERR ビットをチェックして同じ位置に再書き込みを行う事ができます。

## 7.5.4 誤書き込み防止

フラッシュ プログラムメモリへの誤書き込みを防止するため、書き込み開始シーケンスを遵守する必要があります。詳細については[セクション 25.0「CPU の特殊機能」](#)を参照してください。

## 7.6 コード保護中のフラッシュ プログラムメモリの動作

フラッシュ プログラムメモリのコード保護に関する詳細は、[セクション 25.6「プログラムの検証とコード保護」](#)を参照してください。

表 7-2: プログラム フラッシュメモリに関連するレジスタ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	リセット値の記載ページ
TBLPTRU	-	-	bit 21	プログラムメモリ テーブルポインタの最上位バイト (TBLPTR<20:16>)					69
TBPLTRH	プログラムメモリ テーブルポインタの上位バイト (TBLPTR<15:8>)								69
TBLPTRL	プログラムメモリ テーブルポインタの下位バイト (TBLPTR<7:0>)								69
TABLAT	プログラムメモリ テーブルラッチ								69
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	69
EECON2	EEPROM 制御レジスタ 2 (物理レジスタではありません)								71
EECON1	-	-	-	FREE	WRERR	WREN	WR	-	71

凡例: - = 未実装ビット、「0」として読み出されます。網掛けのビットはフラッシュ /EEPROM へのアクセスには使いません。

# PIC18F97J60 ファミリ

---

NOTES:

## 8.0 外部メモリバス

**Note:** 64ピンと80ピンデバイスには外部メモリバスが実装されていません。

外部メモリバス (EMB) を使うと、外部メモリデバイス (フラッシュ、EPROM、SRAM 等) にプログラムメモリまたはデータメモリとしてアクセスできます。EMB は、8ビット/16ビットモードのデータ幅、最大20ビットを含む3つのモードのアドレス幅に対応します。

外部メモリバスは28ピンで実装され、4つのI/Oポートに多重化されています。3つのポート (PORTD、PORTE、PORTH) は、アドレス/データバスと多重化され合計20本のラインが使えます。PORTJはバス制御信号と多重化されています。

表 8-1 にピンと機能の一覧を示します。

表 8-1: PIC18F96J60/96J65/97J60 の外部メモリバス - I/O ポートの機能

バス名	ポート	Bit	外部メモリバスの機能
RD0/AD0	PORTD	0	アドレス Bit 0 またはデータ Bit 0
RD1/AD1	PORTD	1	アドレス Bit 1 またはデータ Bit 1
RD2/AD2	PORTD	2	アドレス Bit 2 またはデータ Bit 2
RD3/AD3	PORTD	3	アドレス Bit 3 またはデータ Bit 3
RD4/AD4	PORTD	4	アドレス Bit 4 またはデータ Bit 4
RD5/AD5	PORTD	5	アドレス Bit 5 またはデータ Bit 5
RD6/AD6	PORTD	6	アドレス Bit 6 またはデータ Bit 6
RD7/AD7	PORTD	7	アドレス Bit 7 またはデータ Bit 7
RE0/AD8	PORTE	0	アドレス Bit 8 またはデータ Bit 8
RE1/AD9	PORTE	1	アドレス Bit 9 またはデータ Bit 9
RE2/AD10	PORTE	2	アドレス Bit 10 またはデータ Bit 10
RE3/AD11	PORTE	3	アドレス Bit 11 またはデータ Bit 11
RE4/AD12	PORTE	4	アドレス Bit 12 またはデータ Bit 12
RE5/AD13	PORTE	5	アドレス Bit 13 またはデータ Bit 13
RE6/AD14	PORTE	6	アドレス Bit 14 またはデータ Bit 14
RE7/AD15	PORTE	7	アドレス Bit 15 またはデータ Bit 15
RH0/A16	PORTH	0	アドレス Bit 16
RH1/A17	PORTH	1	アドレス Bit 17
RH2/A18	PORTH	2	アドレス Bit 18
RH3/A19	PORTH	3	アドレス Bit 19
RJ0/ALE	PORTJ	0	アドレスラッチ イネーブル (ALE) 制御ビット
RJ1/OE	PORTJ	1	出カインーブル (OE) 制御ビット
RJ2/WRL	PORTJ	2	下位バイト書き込み (WRL) 制御ビット
RJ3/WRH	PORTJ	3	上位バイト書き込み (WRH) 制御ビット
RJ4/BA0	PORTJ	4	バイトアドレス (BA0) Bit 0
RJ5/CE	PORTJ	5	チップイネーブル (CE) 制御ビット
RJ6/LB	PORTJ	6	下位バイトイネーブル (LB) 制御ビット
RJ7/UB	PORTJ	7	上位バイトイネーブル (UB) 制御ビット

**Note:** この表では、I/Oポートと外部バスの割り当てだけを示しています。ピンによっては、この表に示した以外の機能が多重化されています。

# PIC18F97J60 ファミリ

## 8.1 外部メモリバスの制御

インターフェイスの動作は、MEMCON レジスタ (レジスタ 8-1) で制御します。このレジスタは、マイクロコントローラ モードを除く、全てのプログラムメモリ動作モードで使えます。マイクロコントローラ モードでは、このレジスタは無効になり書き込みができません。

EBDIS ビット (MEMCON<7>) がバスおよび関連ポート機能の動作を制御します。EBDIS をクリアすると、ポートの I/O 機能とこれらのピンに多重化された全機能が無効になり、インターフェイスが有効になります。このビットをセットすると、I/O ポートとその他の機能が有効になりますが、外部メモリ動作が必要な場合はインターフェイスがピンの他の機能を全てオーバーライドします。既定値では、外部バスが常に有効で他の I/O は全て無効です。

EBDIS ビットの動作は、プログラムメモリ モードにも影響されます。これについては、[セクション 8.5「プログラムメモリ モードと外部メモリバス」](#)で詳細に説明します。

WAIT ビットを使うと、外部メモリ動作にウェイトステートを追加できます。これらのビットの使い方は、[セクション 8.3「ウェイトステート」](#)で説明します。

WM ビットでは、バスが 16 ビットデータ幅モード時の動作モードを選択します。これらの動作モードの詳細は[セクション 8.6「16 ビットデータ幅モード」](#)で説明します。8 ビットデータ幅モード時は、WM ビットは動作に影響しません。

レジスタ 8-1: MEMCON: 外部メモリバス制御レジスタ

R/W-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
EBDIS	—	WAIT1	WAIT0	—	—	WM1	WM0
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

- bit 7      **EBDIS:** 外部バス ディセーブル ビット  
1 = マイクロコントローラが外部メモリにアクセスする時に外部バスを有効にする。それ以外の時は全ての外部バスドライバを I/O ポートに割り当てる  
0 = 常に外部バスを有効にして、I/O ポートを無効にする
- bit 6      **未実装:** 「0」として読み出し
- bit 5-4      **WAIT<1:0>:** テーブル読み書きバスサイクル ウェイト (待機) カウントビット  
11 = テーブル読み書きを 0 Tcy の間待機する  
10 = テーブル読み書きを 1 Tcy の間待機する  
01 = テーブル読み書きを 2 Tcy の間待機する  
00 = テーブル読み書きを 3 Tcy の間待機する
- bit 3-2      **未実装:** 「0」として読み出し
- bit 1-0      **WM<1:0>:** 16 ビットデータバス幅 TBLWT 動作選択ビット  
1x = ワード書き込みモード: TABLAT への書き込みで TBLPTR が奇数アドレスを格納している場合、WRH をアクティブにする。TBLPTR が偶数アドレスを格納している場合、TABLAT に書き込むとその値は保持ラッチに読み込まれる。  
01 = バイト選択モード: TABLAT のデータが MSB と LSB の両方にコピーされ、WRH と (UB または LB) がアクティブになる  
00 = バイト書き込みモード: TABLAT のデータが MSB と LSB の両方にコピーされ、WRH または WRL がアクティブになる

## 8.2 アドレス幅とデータ幅

PIC18F97J60 ファミリは、同じメモリバス上で異なるアドレス幅とデータ幅を使うように設定できます。アドレス幅とデータ幅は、いずれも CONFIG3L レジスタのコンフィグレーション ビットで設定します。コンフィグレーション ビットで設定するという事は、これらの設定がデバイスのプログラミングによってのみ可能であり、ソフトウェアでは制御できない事を意味します。

BW ビットは、8 ビットまたは 16 ビットのデータバス幅を選択します。このビットをセットすると (既定値)、16 ビットのデータ幅が選択されます。

EMB<1:0> ビットは、プログラムメモリの動作モードとアドレスバス幅の両方を決定します。選択可能なモードには、既定値であるマイクロコントローラ モード (外部バスは無効) に加えて、20/16/12 ビットモードがあります。16 ビットまたは 12 ビット幅を選択した場合、対応する本数の上位ラインは I/O 機能に使用されます。これらのピンは、EBDIS ビットの設定の影響を受けません。例えば、16 ビットのアドレッシング モードを選択した場合 (EMB<1:0> = 01)、A<19:16> は無効となり、バスからの割り込みなしに PORTH<3:0> ビットの機能を使用します。アドレス幅を狭くする事で、外部メモリ空間のサイズに合わせてメモリバスを調整すると共に、空いたピンを専用 I/O 動作作用として使えます。

EMB ビットにはメモリバス動作に応じてピンを無効にする効果があるため、必ずデータ幅以上のアドレス幅を選択する事が重要です。12 ビットアドレス幅を 16 ビットデータ幅と共に使う場合、このバスではデータの上位 4 ビットを使えません。

アドレス幅とデータ幅のいかなる組み合わせでも、アドレスとデータの情報を同じライン上で多重化する必要があります。表 8-2 に、アドレスおよびデータの多重化と、アドレス幅を狭くする事で使える I/O ポートを示します。

### 8.2.1 外部バス上のアドレスシフト

既定値では、外部バスに出力されるアドレスは PC の値です。これは事実上、外部メモリのアドレスのうち内蔵メモリの最上位より下位のアドレスをマイクロコントローラが使えない事を意味します。これらの物理位置にアクセスするには、マイクロコントローラと外部メモリ間のグルーロジックが何らかの方法でアドレスを変換する必要があります。

インターフェイスを簡素化するために、外部バスは自動的にアドレスシフトを実行する、拡張マイクロコントローラ モードと呼ばれる拡張機能を提供します。この機能は、EASHFT コンフィグレーション ビットによって制御されます。このビットをセットすると、バス上のアドレスがマイクロコントローラの内蔵プログラムメモリのサイズだけオフセットされ、最下位アドレスが 0000h に設定されます。これによって、デバイスは外部メモリの物理アドレスの全範囲を使えます。

### 8.2.2 21 ビット アドレッシング

20 ビット アドレッシングの拡張として、外部メモリバスでは 2 MB のメモリ空間全体をアドレス指定できます。それには、バスアドレス Bit 0 (BA0) 制御ラインをアドレスの最下位ビットとして使います。一部のメモリデバイスでは、UB および LB 制御信号を使って、16 ビット幅のデータワード内の上位バイトと下位バイトを選択する事もできます。

このアドレッシングモードは 8 ビットデータ幅モードと一部の 16 ビットデータ幅モードの両方で使えます。詳細は、[セクション 8.6.3 「16 ビットバイト選択モード」](#)と[セクション 8.7 「8 ビットデータ幅モード」](#)で説明します。

表 8-2: 各種のアドレス幅とデータ幅に対するアドレスラインとデータライン

データ幅	アドレス幅	多重化されるデータラインとアドレスライン (対応ポート)	アドレス専用ライン (対応ポート)	I/O に使えるポート
8 ビット	12 ビット	AD<7:0> (PORTD<7:0>)	AD<11:8> (PORTE<3:0>)	PORTE<7:4>、 PORTH 全体
	16 ビット		AD<15:8> (PORTE<7:0>)	PORTH 全体
	20 ビット		A<19:16>, AD<15:8> (PORTH<3:0>、 PORTE<7:0>)	-
16 ビット	16 ビット	AD<15:0> (PORTD<7:0>、 PORTE<7:0>)	-	PORTH 全体
	20 ビット		A<19:16> (PORTH<3:0>)	-

# PIC18F97J60 ファミリ

## 8.3 ウェイトステート

外部メモリデバイスはマイクロコントローラのクロックレートで動作するものと思われがちですが、そうでない場合も数多く存在します。実際多くのデバイスは、データの書き込みまたは読み取りに、テーブル読み書きに与えられるよりも長い時間を必要とします。

これを補うために、バスを使うテーブル動作ごとに一定の遅延を付加するように外部メモリバスを設定できます。ウェイトステートは、WAIT コンフィグレーションビットをセットする事で有効化します。有効時、遅延時間は WAIT<1:0> (MEMCON<5:4>) ビットで設定します。遅延は、マイクロコントローラの命令サイクル時間の倍数を基準とし、テーブル動作を実行した命令サイクルの後に加えられます。遅延時間の範囲は、遅延なしから 3 TCY (既定値) です。

## 8.4 ポートピンの弱プルアップ

上位アドレスラインの A<19:16> を除き、外部メモリバスに関連付けられているピンは弱プルアップを備えています。プルアップは、LATA<7:6> と PORTA<7> の位置にあるビットによって制御されます。これらのビットは RDPU、REPU、RJPU と呼ばれ、それぞれ PORTD、PORTE、PORTJ のプルアップを制御します。これらのビットをセットすると、そのポートに対応するプルアップが有効になります。既定値では、全てのデバイスリセットでプルアップは全て無効になります。

拡張マイクロコントローラ モードでバスが一時的に無効化されている (EBDIS = 1) 時に、外部バスのメモリ状態を保持するためにポートのプルアップが役に立つ場合があります。

## 8.5 プログラムメモリ モードと外部メモリバス

PIC18F97J60 ファミリのデバイスは、内蔵メモリと外部のプログラムメモリを組み合わせる事で、2 つのプログラムメモリ モードのいずれかによる動作が可能です。多重化されたポートピンの機能は、選択したプログラムメモリ モードと EBDIS ビットの設定によって決まります。

マイクロコントローラ モードの場合、バスは非アクティブでピンはポート機能だけを果たします。MEMCOM レジスタには書き込む事ができません。EBDIS のリセット値 (「0」) は無視され、EMB ピンは I/O ポートとして動作します。

拡張マイクロコントローラ モードでは、外部プログラムメモリバスがピンの I/O ポート機能を共有します。デバイスが外部プログラムメモリ空間上でフェッチ動作またはテーブル読み書きを実行している間のピンは、外部バス機能を持ちます。

デバイスが内部プログラムメモリ位置に対してのみフェッチまたはアクセスしている場合、EBDIS 制御ビットに基づいてピンの機能が外部メモリから I/O ポートに変更されます。EBDIS = 0 の場合、ピンは外部バスとして機能します。EBDIS = 1 の場合、ピンは I/O ポートとして機能します。

EBDIS = 1 の時にデバイスが外部メモリに対してフェッチまたはアクセスを行うと、ピンは外部バスに切り換わります。外部メモリから実行中のプログラムによって EBDIS ビットをセットする場合、その動作はプログラムが内部メモリに分岐するまで遅延されます。内部メモリに分岐した時点で、ピンは外部バスから I/O ポートに変わります。

EBDIS = 0 で内部メモリからプログラムを実行している場合、メモリバスのアドレス / データおよび制御ピンはアクティブになりません。アクティブなアドレス / データピンは 3 ステートに移行し、 $\overline{CE}$ 、 $\overline{OE}$ 、WRH、WRL、UB、LB 信号は「1」に、ALE と BA0 は「0」の状態に移行します。3 ステートに移行するのは現在のアドレス幅に対応するピンだけであり、その他のピンは引き続き I/O として機能する事に注意してください。例えばアドレス幅が 16 ビットの場合、AD<15:0> (PORTD と PORTE) だけが影響を受け、A<19:16> (PORTH<3:0>) は引き続き I/O として機能します。

いずれの外部メモリモードでも、バスは多重化された他の周辺機能よりも優先されます。ここでいう周辺機能には、パラレルスレーブポートとシリアル通信モジュールも含まれます。これらは、I/O ポートよりは優先されます。

## 8.6 16 ビットデータ幅モード

16 ビットデータ幅モードでは、外部メモリ インターフェイスを以下の 3 種類の設定で外部メモリに接続できます。

- 16 ビットバイト書き込み
- 16 ビットワード書き込み
- 16 ビットバイト選択

使う設定は、MEMCON レジスタの WM<1:0> ビット (MEMCON<1:0>) によって決まります。これら 3 つの設定を使う事で、8 ビットと 16 ビットのいずれのデバイスでも 16 ビットのデータを使い、最大限の柔軟性を得る事ができます。

16 ビットデータ幅のいずれのモードでも、アドレスラッチイネーブル (ALE) ピンは、外部メモリ インターフェイスバスのアドレスビット AD<15:0> が使える事を示します。アドレスラッチに続き、出力イネーブル信号 ( $\overline{OE}$ ) がプログラムメモリの両バイトを同時に有効化し、16 ビットの命令ワードが構成されます。チップイネーブル信号 (CE) は、マイクロコントローラが外部メモリを読み書きしている間は常にアクティブです。デバイスがスリープの間は、常に非アクティブ (High アサート) です。

バイト選択モードの場合、JEDEC 標準のフラッシュメモリはバイトアドレスライン用に BA0、バイトモードとワードモードの選択用に I/O ラインを 1 本必要とします。その他の 16 ビットデータ幅モードは BA0 を必要としません。JEDEC 標準の SRAM メモリは、バイト選択用に UB または LB 信号を使います。

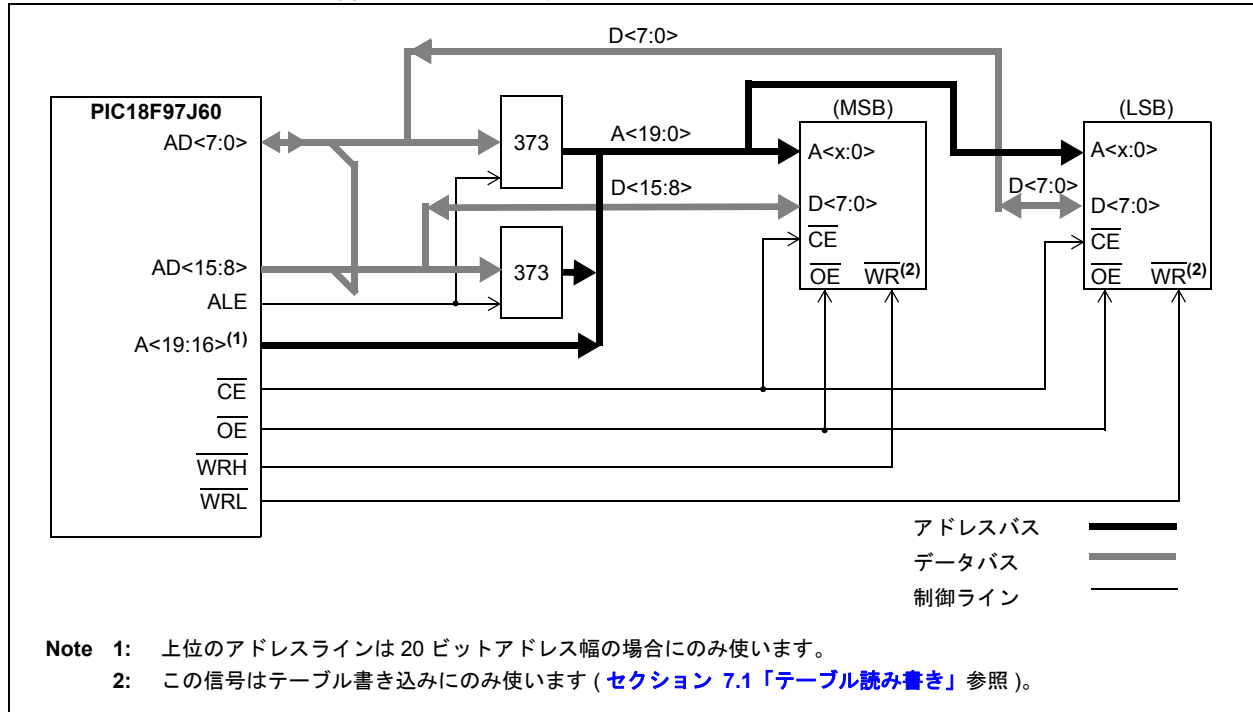
# PIC18F97J60 ファミリ

## 8.6.1 16 ビットバイト書き込みモード

図 8-1 に、PIC18F97J60 ファミリの 16 ビットバイト書き込みモードの例を示します。このモードを使って、独立した 2 つの 8 ビットメモリを連結して 16 ビット動作を実現します。この構成は通常、基本的な EPROM とフラッシュ デバイ스에適用できます。バイト幅の外部メモリに対するテーブル書き込みが可能です。

TBLWT 命令サイクルの間、AD<15:0> バスの上位および下位バイトには TABLAT のデータが出力されます。対応する WRH または WRL 制御ラインは、TBLPTR の Lsb でストローブされます。

図 8-1: 16 ビットバイト書き込みモードの例



# PIC18F97J60 ファミリ

## 8.6.2 16 ビットワード書き込みモード

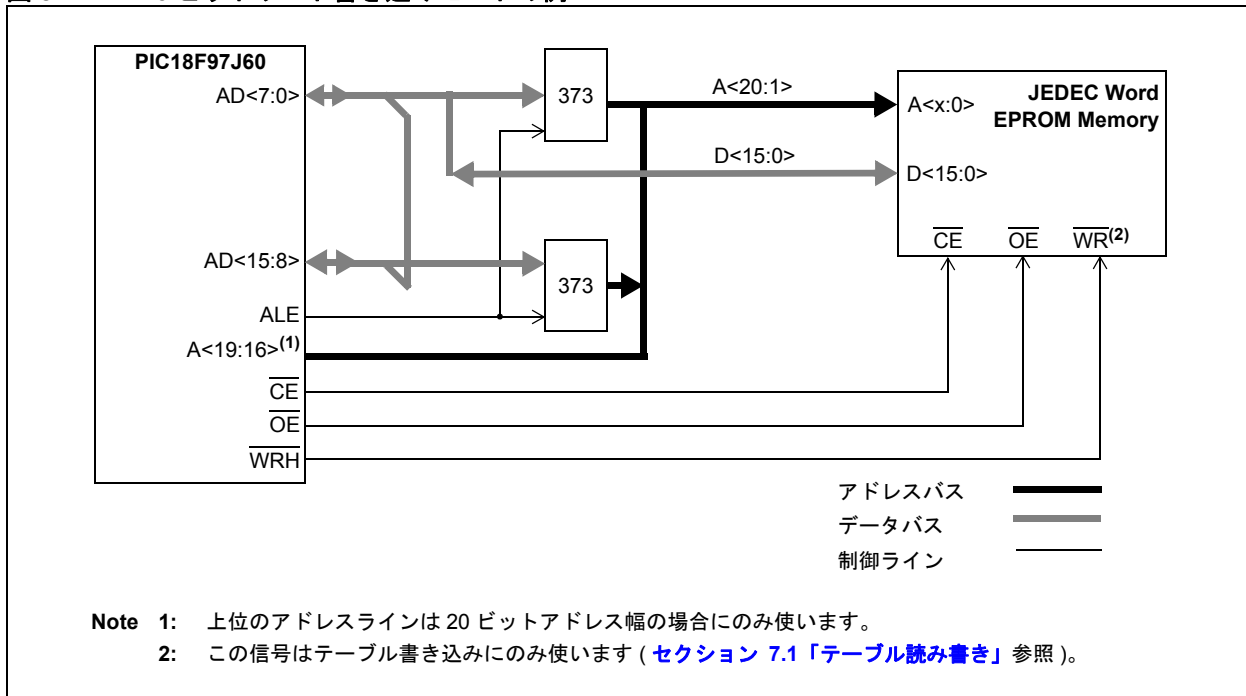
図 8-2 に、PIC18F97J60 ファミリの 16 ビットワード書き込みモードの例を示します。このモードはワード幅のメモリに対して使います。そのようなメモリには一部の EPROM やフラッシュタイプのメモリが含まれます。このモードでは、あらゆる 16 ビットメモリからのオペコードフェッチとテーブル読み出し、あらゆるワード幅外部メモリに対するテーブル書き込みが可能です。この方法では、偶数アドレスと奇数アドレスに対する TBLWT サイクルを区別します。

偶数アドレスに対する TBLWT サイクル (TBLPTR<0> = 0) 中、TABLAT のデータが保持ラッチに転送され、バスサイクルのデータ部分では外部アドレス データバスが 3 ステート状態です。書き込み信号は一切アクティブになりません。

奇数アドレスに対する TBLWT サイクル (TBLPTR<0> = 1) 中、TABLAT のデータが AD<15:0> バスの上位バイトに出力されます。保持ラッチの内容は AD<15:0> バスの下位バイトに出力されます。

書き込みサイクルごとに WRH 信号がストローブされます。WRL ピンは使いません。BA0 ピンの信号は TBLPTR の LSb を示しますが、未接続のままとします。代わりに UB および LB 信号がアクティブになり、両方のバイトを選択します。この方法の制約は、ワード位置に正確に書き込むにはテーブル書き込みを特定のワード境界に対してペアで実行する必要がある、という事です。

図 8-2: 16 ビットワード書き込みモードの例





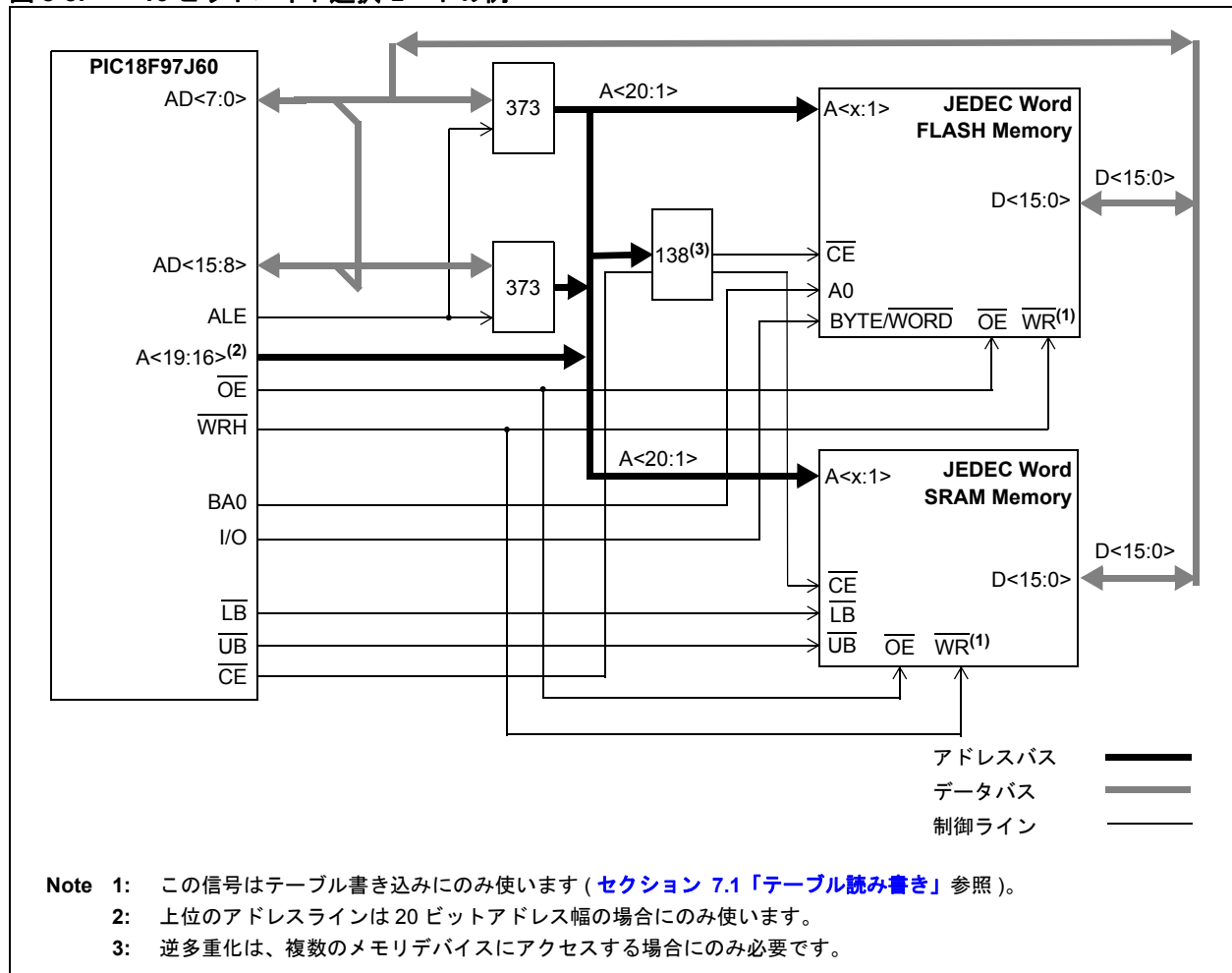
## 8.6.3 16 ビットバイト選択モード

図 8-3に、16ビットバイト選択モードの例を示します。このモードを使うと、バイト選択機能を持つワード幅の外部メモリに対するテーブル書き込みが可能です。このようなメモリにはワード幅のフラッシュと SRAM デバイスが含まれます。

TBLWT サイクル中、AD<15:0> バスの上位および下位バイトには TABLAT のデータが出力されます。書き込みサイクルごとに WRH 信号がストロブされます。WRL ピンは使いません。TBLPTR レジスタの最下位ビットに基づき、BA0 または UB/LB 信号によって書き込むバイトを選択します。

バイト選択モードの実装には、フラッシュと SRAM デバイスで異なる制御信号の組み合わせを使います。JEDEC 標準のフラッシュメモリでは、選択信号を提供するために、コントローラの I/O ポートピンをメモリの BYTE/WORD ピンに接続する必要があります。さらにコントローラからの BA0 信号もバイトアドレスとして使います。一方、JEDEC 標準の SRAM メモリは、バイト選択用に UB または LB 信号を使います。

図 8-3: 16 ビットバイト選択モードの例



# PIC18F97J60 ファミリ

## 8.6.4 16 ビットモードのタイミング

外部メモリバスに出力される制御信号は、各動作モードで異なります。図 8-4 と図 8-5 に、代表的な信号タイミング チャートを示します。

図 8-4: TBLRD の外部メモリバス タイミング (拡張マイクロコントローラ モード)

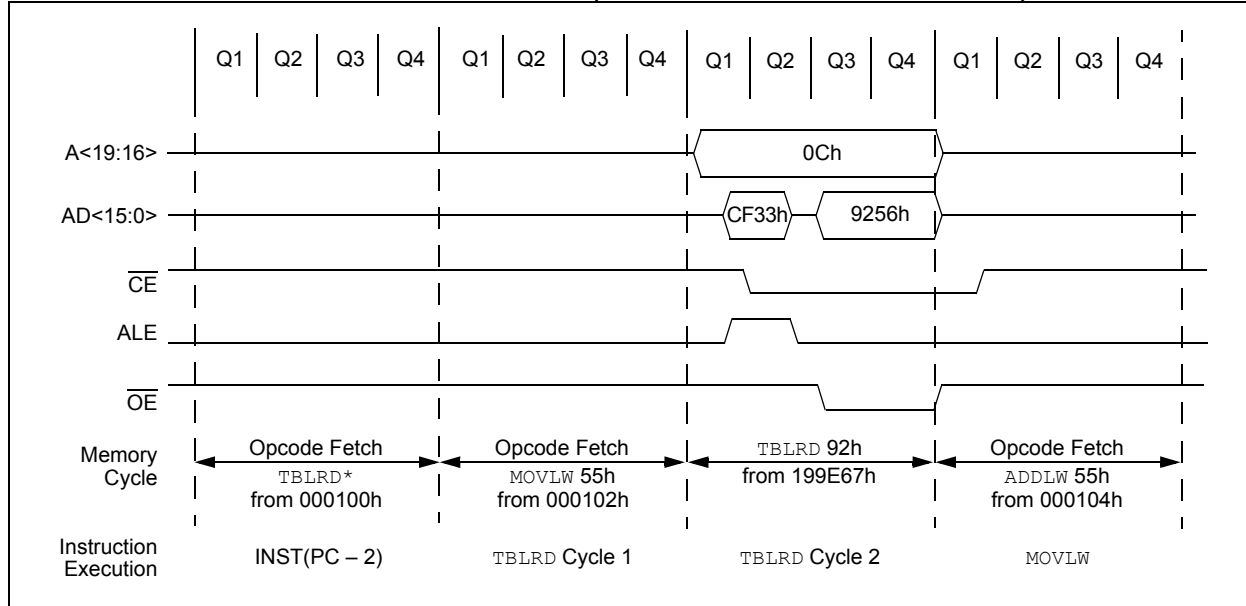
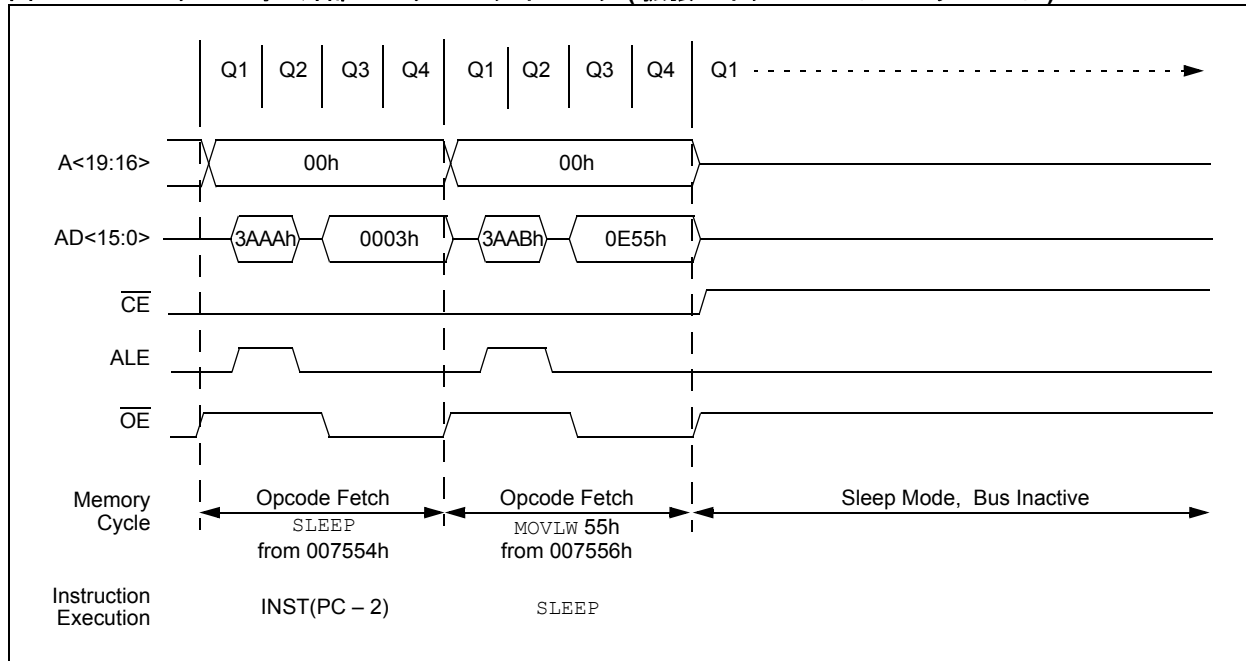


図 8-5: スリープ時の外部メモリバス タイミング (拡張マイクロコントローラ モード)



## 8.7 8ビットデータ幅モード

8ビットデータ幅モードの外部メモリバスは多重化モードでのみ動作します。すなわち、データはアドレスバスの下位8ビットを共有します。

図 8-6 に、100 ピンデバイスの 8 ビット多重化モードの例を示します。このモードを使って、16 ビット動作で単一の 8 ビットメモリを接続します。命令は、共有データ/アドレスバス上の 2 つの 8 ビットバイトとしてフェッチされます。この 2 バイトは、1 命令サイクル (Tcy) 内で連続してフェッチされます。従って、1/2 Tcy (命令レートの 2 倍) に基づくタイミング計算を基に外部メモリデバイスを選ぶ必要があります。メモリスピードを適切に選択するには、セットアップおよびホールド時間と共に、グルーロジック伝播遅延時間も考慮する必要があります。

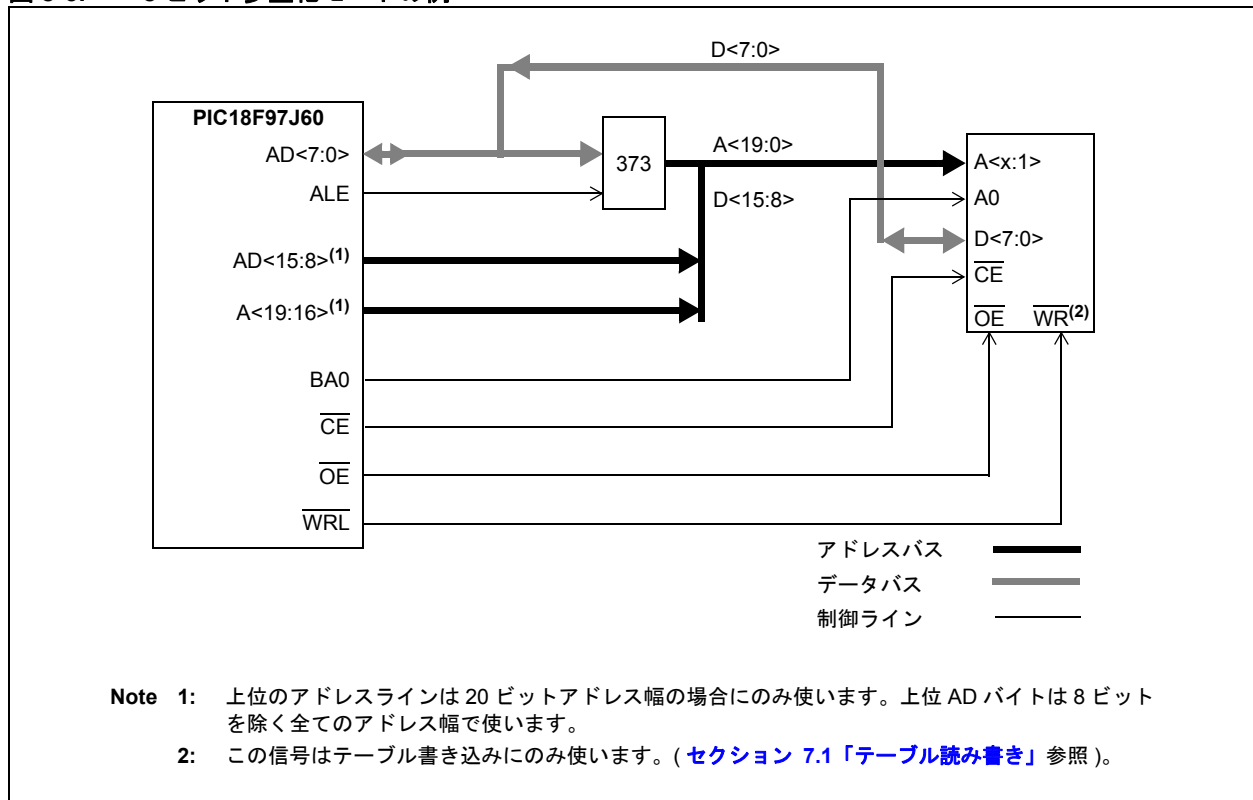
アドレスラッチ イネーブル (ALE) ピンは、外部メモリインターフェイスバスのアドレスビット AD<15:0> が使える事を示します。出力イネーブル信号 ( $\overline{OE}$ ) が、命

令サイクル内でプログラムメモリの 1 バイトを有効にした後、BA0 が変化して 2 番目のバイトが有効化される事で 16 ビットの命令ワードが形成されます。このモードでは、アドレスの最下位ビット BA0 をメモリデバイスに接続する必要があります。チップイネーブル信号 (CE) は、マイクロコントローラが外部メモリを読み書きしている間は常にアクティブです。デバイスがスリープの間は、常に非アクティブ (High アサート) です。

このプロセスには通常、基本的な EPROM とフラッシュ デバイスが含まれます。バイト幅の外部メモリに対するテーブル書き込みが可能です。

TBLWT 命令サイクルの間、AD<15:0> バスの上位および下位バイトには TABLAT のデータが出力されます。BA0 制御ラインの該当するレベルは、TBLPTR の LSB でストローブされます。

図 8-6: 8ビット多重化モードの例



# PIC18F97J60 ファミリ

## 8.7.1 8ビットモードのタイミング

外部メモリバスに出力される制御信号は、各種の動作モードで異なります。図 8-7 と 図 8-8 に、代表的な信号タイミングチャートを示します。

図 8-7: TBLRD の外部メモリバス タイミング (拡張マイクロコントローラモード)

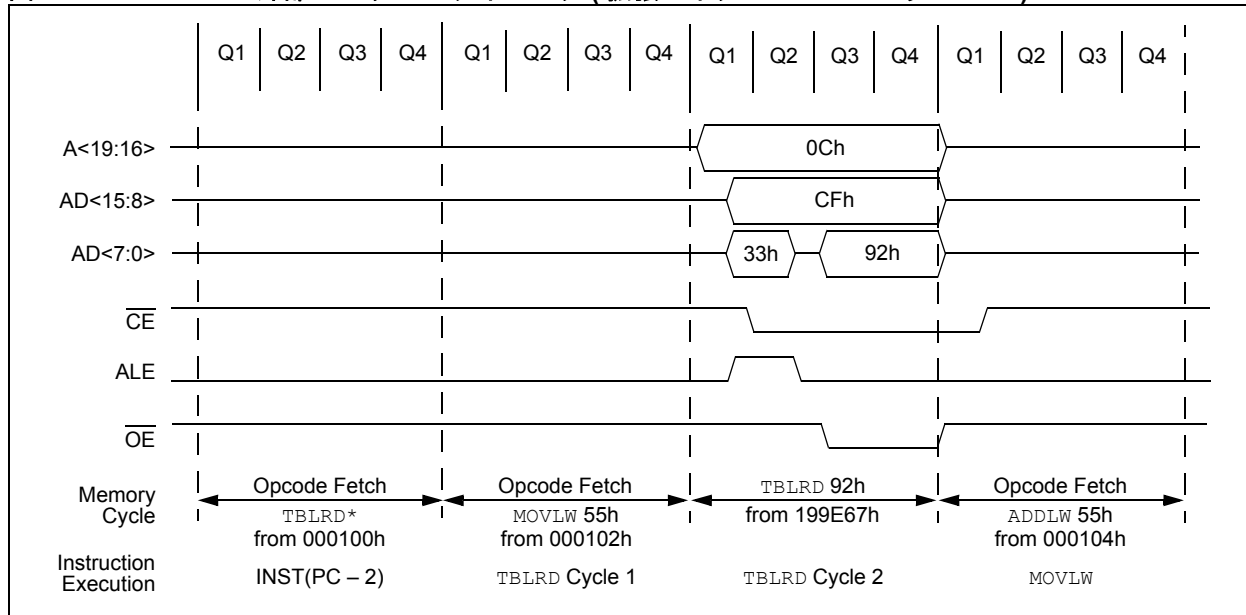
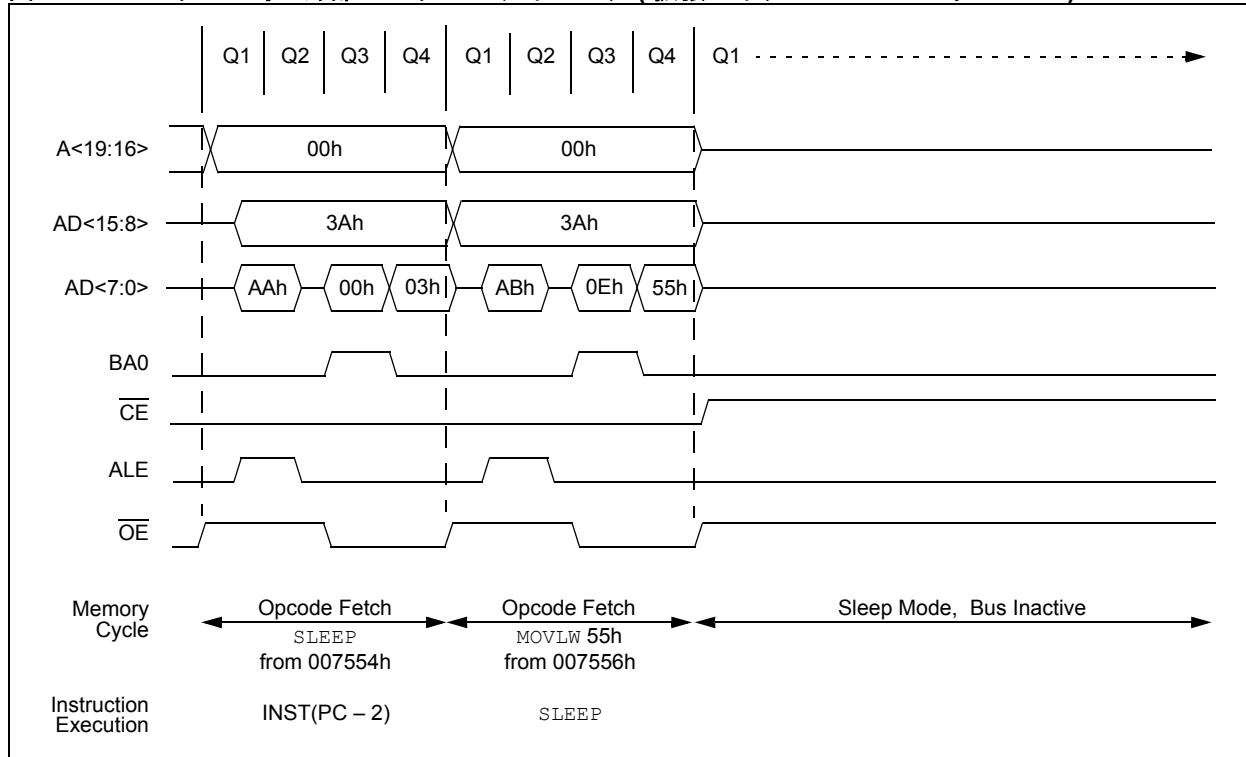


図 8-8: スリープ時の外部メモリバス タイミング (拡張マイクロコントローラモード)



## 8.8 電力管理モードにおける動作

電力管理ランモードでも、外部バスは通常動作を継続します。より低速のクロック源を選択した場合、バスはその速度で動作します。その場合、ウェイトステートを有効にして外部メモリ動作に適用していると、外部メモリに対するアクセス時間が過剰にかかる可能性があります。より消費電力の低いランモードの動作が予想される場合、低速のクロックに対してユーザアプリケーション内でメモリアクセス時間を調整する必要があります。

アイドル/スリープ中、マイクロコントローラ コアはデータにアクセスする必要がないため、バス動作は中断されます。外部バスの状態は固定され、アドレス / データピンと大部分の制御ピンはそのモードに移行した時点と同じ状態を保持します。変化する可能性があるピンは、CE、LB、UB だけです。これらは論理 High に保持されます。

# PIC18F97J60 ファミリ

---

NOTES:

## 9.0 8 x 8 ハードウェア乗算器

### 9.1 はじめに

全ての PIC18 は、ALU の一部として 8x8 のハードウェア乗算器を搭載しています。この乗算器は符号なし演算を実行し、16 ビットの演算結果を生成します。結果は積レジスタペア PRODH:PRODL に格納されます。乗算器による演算は、STATUS レジスタのフラグを一切変化させません。

ハードウェアによる乗算を利用する事で、1 命令サイクルで演算が完了します。これには、計算のスループットが向上すると同時に、乗算アルゴリズムのコードサイズを削減できるという利点があり、以前はデジタルシグナルプロセッサを使う必要があった多くのアプリケーションで PIC18 を使えるようになりました。表 9-1 に、各種のハードウェアとソフトウェアによる乗算を比較し、それぞれのメモリと実行時間を示します。

### 9.2 動作

例 9-1 に、8x8 符号なし乗算の命令シーケンスを示します。引数の一方が WREG レジスタにあらかじめ格納されていれば、乗算には 1 命令しか必要ありません。

例 9-2 に、8x8 符号付き乗算のシーケンスを示します。引数の符号ビットを適切に処理するために、各引数の最上位ビット (MSb) の値を調べ、必要な減算を実行しています。

#### 例 9-1: 8 x 8 符号なし乗算のルーチン

```
MOVWF ARG1, W ;
MULWF ARG2 ; ARG1 * ARG2 ->
; PRODH:PRODL
```

#### 例 9-2: 8 x 8 符号付き乗算のルーチン

```
MOVWF ARG1, W
MULWF ARG2 ; ARG1 * ARG2 ->
; PRODH:PRODL

BTFSC ARG2, SB ; Test Sign Bit
SUBWF PRODH, F ; PRODH = PRODH
; - ARG1

MOVWF ARG2, W
BTFSC ARG1, SB ; Test Sign Bit
SUBWF PRODH, F ; PRODH = PRODH
; - ARG2
```

表 9-1: 各種乗算における性能比較

ルーチン	乗算の方法	プログラムメモリ (ワード)	サイクル数 (最大)	時間		
				@ 40 MHz	@ 10 MHz	@ 4 MHz
8 x 8 符号なし	ハードウェア乗算なし	13	69	6.9 μs	27.6 μs	69 μs
	ハードウェア乗算	1	1	100 ns	400 ns	1 μs
8 x 8 符号付き	ハードウェア乗算なし	33	91	9.1 μs	36.4 μs	91 μs
	ハードウェア乗算	6	6	600 ns	2.4 μs	6 μs
16 x 16 符号なし	ハードウェア乗算なし	21	242	24.2 μs	96.8 μs	242 μs
	ハードウェア乗算	28	28	2.8 μs	11.2 μs	28 μs
16 x 16 符号付き	ハードウェア乗算なし	52	254	25.4 μs	102.6 μs	254 μs
	ハードウェア乗算	35	40	4.0 μs	16.0 μs	40 μs

# PIC18F97J60 ファミリ

例 9-3 に、16 x 16 符号なし乗算のシーケンスを示します。式 9-1 に、適用されるアルゴリズムを示します。32 ビットの演算結果は 4 つのレジスタ (RES3:RES0) に格納されます。

## 式 9-1: 16 x 16 符号なし乗算のアルゴリズム

```
RES3:RES0 = ARG1H:ARG1L • ARG2H:ARG2L
           = (ARG1H • ARG2H • 216) +
             (ARG1H • ARG2L • 28) +
             (ARG1L • ARG2H • 28) +
             (ARG1L • ARG2L)
```

## 例 9-3: 16 x 16 符号なし乗算のルーチン

```
MOVF ARG1L, W
MULWF ARG2L          ; ARG1L * ARG2L->
                     ; PRODH:PRODL
MOVFF PRODH, RES1   ;
MOVFF PRODL, RES0   ;
;
MOVF ARG1H, W
MULWF ARG2H          ; ARG1H * ARG2H->
                     ; PRODH:PRODL
MOVFF PRODH, RES3   ;
MOVFF PRODL, RES2   ;
;
MOVF ARG1L, W
MULWF ARG2H          ; ARG1L * ARG2H->
                     ; PRODH:PRODL
MOVF PRODL, W       ;
ADDWF RES1, F       ; Add cross
MOVF PRODH, W       ; products
ADDWFC RES2, F      ;
CLRF WREG           ;
ADDWFC RES3, F      ;
;
MOVF ARG1H, W       ;
MULWF ARG2L          ; ARG1H * ARG2L->
                     ; PRODH:PRODL
MOVF PRODL, W       ;
ADDWF RES1, F       ; Add cross
MOVF PRODH, W       ; products
ADDWFC RES2, F      ;
CLRF WREG           ;
ADDWFC RES3, F      ;
;
MOVF ARG1H, W       ;
MULWF ARG2L          ; ARG1H * ARG2L->
                     ; PRODH:PRODL
MOVF PRODL, W       ;
ADDWF RES1, F       ; Add cross
MOVF PRODH, W       ; products
ADDWFC RES2, F      ;
CLRF WREG           ;
ADDWFC RES3, F      ;
```

例 9-4 に、16 x 16 符号付き乗算のシーケンスを示します。式 9-2 に、適用されるアルゴリズムを示します。32 ビットの演算結果は 4 つのレジスタ (RES3:RES0) に格納されます。引数の符号ビットを適切に処理するために、各引数ペアの最上位ビット (MSb) の値を調べ、必要な減算を実行しています。

## 式 9-2: 16 x 16 符号付き乗算のアルゴリズム

```
RES3:RES0 = ARG1H:ARG1L • ARG2H:ARG2L
           = (ARG1H • ARG2H • 216) +
             (ARG1H • ARG2L • 28) +
             (ARG1L • ARG2H • 28) +
             (ARG1L • ARG2L) +
             (-1 • ARG2H<7> • ARG1H:ARG1L • 216) +
             (-1 • ARG1H<7> • ARG2H:ARG2L • 216)
```

## 例 9-4: 16 x 16 符号付き乗算のルーチン

```
MOVF ARG1L, W
MULWF ARG2L          ; ARG1L * ARG2L ->
                     ; PRODH:PRODL
MOVFF PRODH, RES1   ;
MOVFF PRODL, RES0   ;
;
MOVF ARG1H, W
MULWF ARG2H          ; ARG1H * ARG2H ->
                     ; PRODH:PRODL
MOVFF PRODH, RES3   ;
MOVFF PRODL, RES2   ;
;
MOVF ARG1L, W
MULWF ARG2H          ; ARG1L * ARG2H ->
                     ; PRODH:PRODL
MOVF PRODL, W       ;
ADDWF RES1, F       ; Add cross
MOVF PRODH, W       ; products
ADDWFC RES2, F      ;
CLRF WREG           ;
ADDWFC RES3, F      ;
;
MOVF ARG1H, W       ;
MULWF ARG2L          ; ARG1H * ARG2L ->
                     ; PRODH:PRODL
MOVF PRODL, W       ;
ADDWF RES1, F       ; Add cross
MOVF PRODH, W       ; products
ADDWFC RES2, F      ;
CLRF WREG           ;
ADDWFC RES3, F      ;
;
BTFSS ARG2H, 7      ; ARG2H:ARG2L neg?
BRA SIGN_ARG1       ; no, check ARG1
MOVF ARG1L, W       ;
SUBWF RES2          ;
MOVF ARG1H, W       ;
SUBWFB RES3         ;
;
SIGN_ARG1
BTFSS ARG1H, 7      ; ARG1H:ARG1L neg?
BRA CONT_CODE       ; no, done
MOVF ARG2L, W       ;
SUBWF RES2          ;
MOVF ARG2H, W       ;
SUBWFB RES3         ;
;
CONT_CODE
:
```



## 10.0 割り込み

PIC18F97J60 ファミリに含まれるデバイスには複数の割り込み要因があり、そのほとんどに高 / 低優先度を設定できます。高優先度割り込みのベクタは 0008h、低優先度割り込みのベクタは 0018h です。高優先度の割り込みイベントは、全ての低優先度割り込みに、それらが処理中であっても割り込む事ができます。

割り込み動作の制御には、以下に示す 13 個のレジスタを使います。

- RCON
- INTCON
- INTCON2
- INTCON3
- PIR1、PIR2、PIR3
- PIE1、PIE2、PIE3
- IPR1、IPR2、IPR3

これらのレジスタ内のビットに対応するシンボル名を使うために、MPLAB® IDE に付属するマイクロチップ社のヘッダファイルの使用を推奨します。このヘッダファイルによって、アセンブラまたはコンパイラは指定されたレジスタ内のこれらのビット位置を自動的に処理できます。

通常、各割り込み要因には、その動作を制御する以下の 3 つのビットが実装されています。

- **フラグビット**：割り込みイベントが発生した事を示します。
- **イネーブルビット**：フラグビットがセットされた場合、実行中のプログラムが割り込みベクタアドレスに分岐する事を許可します。
- **優先度ビット**：割り込みの高優先度と低優先度を選択します。

割り込みの優先度機能は、IPEN ビット (RCON<7>) をセットすると有効になります。割り込み優先度機能が有効な場合、グローバル割り込みを有効化する 2 つのビットがあります。GIEH ビット (INTCON<7>) をセットすると、優先度ビットがセットされている (高優先度) 割り込みが全て有効になります。GIEL ビット (INTCON<6>) をセットすると、優先度ビットがクリアされている (低優先度) 割り込みが全て有効になります。割り込みフラグ、割り込みイネーブルビット、対応するグローバル割り込みイネーブルビットがセットされている場合、割り込みは優先度ビットの設定に応じて、ただちにアドレス 0008h または 0018h に分岐します。各割り込みは、対応するイネーブルビットによって個別に無効にできます。

IPEN ビットがクリアの場合 (既定値の状態)、割り込み優先度機能は無効であり、割り込み動作に関して PIC® ミッドレンジ デバイスと互換です。この互換モードでは、各割り込み要因の割り込み優先度ビットは無視されます。INTCON<6> は周辺機能の全ての割り込み要因を有効 / 無効にする PEIE ビットです。INTCON<7> は全ての割り込み要因を有効 / 無効にする GIE ビットです。互換モードの場合、全ての割り込みはアドレス 0008h に分岐します。

割り込みの処理が開始すると、グローバル割り込みイネーブルビットがクリアされ、以降の割り込みが無効になります。このグローバル割り込みイネーブルビットとは、IPEN ビットがクリアされている場合は GIE ビット、割り込み優先度レベルを使う場合は GIEH ビットと GIEL ビットのいずれかです。高優先度の割り込み要因は、低優先度の割り込み処理に割り込む事ができます。高優先度の割り込み処理の実行中は、低優先度の割り込みは処理されません。

割り込み処理からのリターンアドレスはスタックにプッシュされ、PIC には割り込みベクタアドレス (0008h または 0018h) が読み込まれます。割り込みサービスルーチン (ISR) への分岐後に割り込み要因を判断するには、割り込みフラグビットをポーリングします。割り込みが再帰的に繰り返されないように、割り込みを再度有効にする前に、割り込みフラグビットをクリアする必要があります。

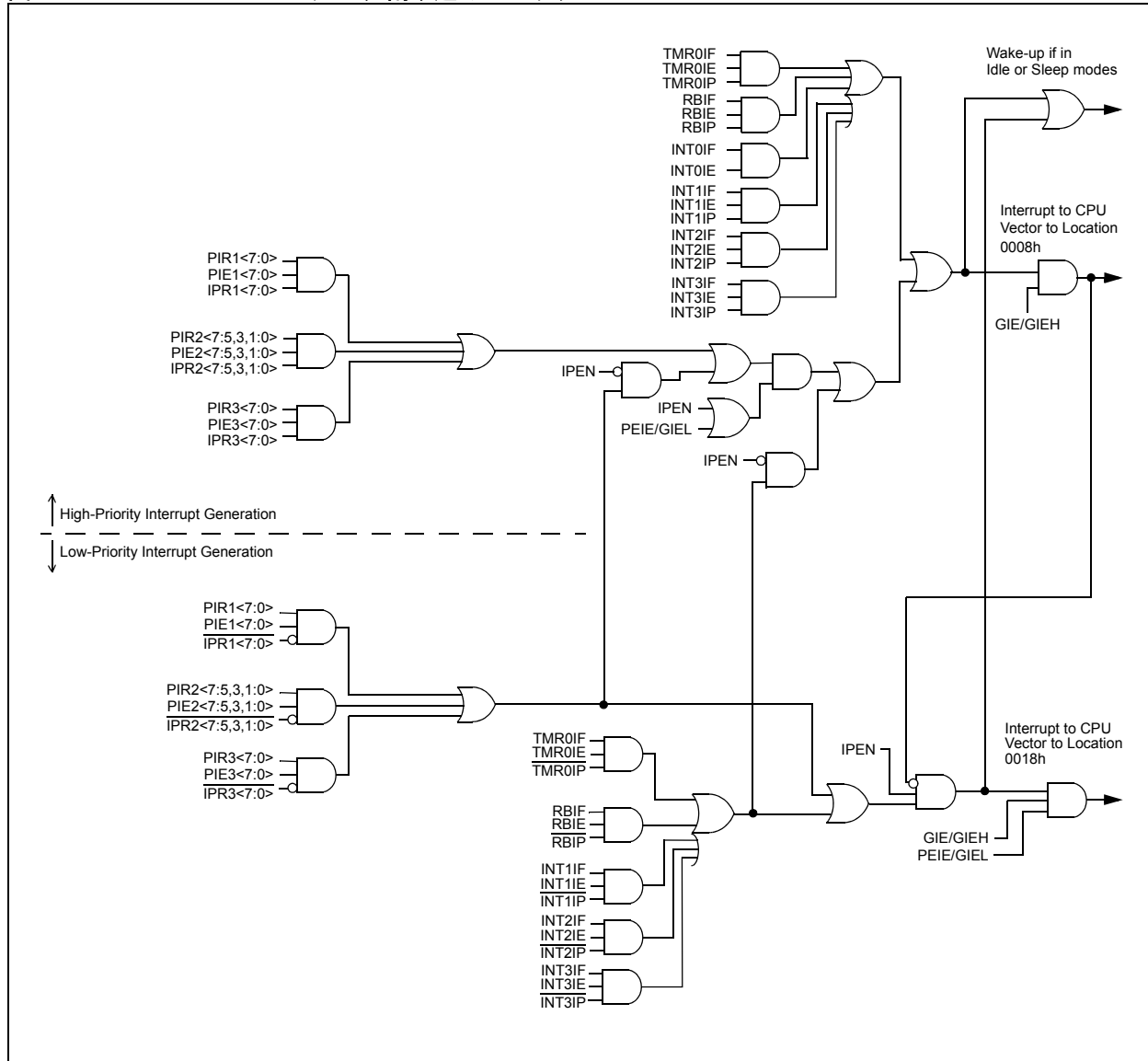
「割り込みからの戻り」命令 RETFIE は割り込みルーチンを終了し、割り込みを再度有効にする GIE ビット (優先度レベルを使う場合は GIEH または GIEL) をセットします。

INTx ピンや PORTB の入力状態変化割り込み等の外部割り込みイベントによる割り込みのレイテンシは 3 ~ 4 命令サイクルです。このレイテンシは、1 サイクル命令と 2 サイクル命令のどちらでも同じです。各割り込みフラグビットは、対応するイネーブルビットまたは GIE ビットの状態に関わらずセットされます。

**Note:** 割り込み有効時、割り込み制御レジスタを MOVFF 命令で変更してはいけません。マイクロコントローラの挙動が不安定になる場合があります。

# PIC18F97J60 ファミリ

図 10-1: PIC18F97J60 ファミリ 割り込みロジック



# PIC18F97J60 ファミリ

## 10.1 INTCON レジスタ

INTCON は、各種割り込みの有効化、優先度、フラグビットを格納する読み書き可能なレジスタです。

**Note:** 割り込み条件が発生すると、対応するイネーブルビットまたはグローバル割り込みイネーブルビットの状態に関わらず、各割り込みのフラグビットがセットされます。ユーザ ソフトウェアは、割り込みを有効にする前に、対応する割り込みフラグビットを確実にクリアしておく必要があります。上記の機能によって、ソフトウェアポーリングが可能になります。

レジスタ 10-1: INTCON: 割り込み制御レジスタ

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF <sup>(1)</sup>
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

- bit 7      **GIE/GIEH:** グローバル割り込みイネーブルビット  
IPEN = 0 の場合:  
1 = マスクされていない全ての割り込みを有効にする  
0 = 全ての割り込みを無効にする  
IPEN = 1 の場合:  
1 = 全ての高優先度割り込みを有効にする  
0 = 全ての割り込みを無効にする
- bit 6      **PEIE/GIEL:** 周辺割り込みイネーブルビット  
IPEN = 0 の場合:  
1 = マスクされていない全ての周辺割り込みを有効にする  
0 = 全ての周辺割り込みを無効にする  
IPEN = 1 の場合:  
1 = 全ての低優先度周辺割り込みを有効にする  
0 = 全ての低優先度周辺割り込みを無効にする
- bit 5      **TMR0IE:** TMR0 オーバーフロー割り込みイネーブルビット  
1 = TMR0 オーバーフロー割り込みを有効にする  
0 = TMR0 オーバーフロー割り込みを無効にする
- bit 4      **INT0IE:** INT0 外部割り込みイネーブルビット  
1 = INT0 外部割り込みを有効にする  
0 = INT0 外部割り込みを無効にする
- bit 3      **RBIE:** RB ポート状態変化割り込みイネーブルビット  
1 = RB ポートの状態変化割り込みを有効にする  
0 = RB ポートの状態変化割り込みを無効にする
- bit 2      **TMR0IF:** TMR0 オーバーフロー割り込みフラグビット  
1 = TMR0 レジスタがオーバーフローした (ソフトウェアによるクリアが必要)  
0 = TMR0 レジスタはオーバーフローしていない
- bit 1      **INT0IF:** INT0 外部割り込みフラグビット  
1 = INT0 外部割り込みが発生した (ソフトウェアによるクリアが必要)  
0 = INT0 外部割り込みは発生していない
- bit 0      **RBIF:** RB ポート状態変化割り込みフラグビット<sup>(1)</sup>  
1 = RB<7:4> ピンのいずれか 1 つ以上の状態が変化した (ソフトウェアによるクリアが必要)  
0 = RB<7:4> ピンの状態はいずれも変化していない

**Note 1:** 不一致条件が続く限り、このビットはセットされたままです。PORTB を読み出す事で不一致条件が終了し、ビットをクリアできるようになります。

# PIC18F97J60 ファミリ

## レジスタ 10-2: INTCON2: 割り込み制御レジスタ 2

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{RBPU}}$	INTEDG0	INTEDG1	INTEDG2	INTEDG3	TMR0IP	INT3IP	RBIP
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

- bit 7       **$\overline{\text{RBPU}}$** : PORTB プルアップ イネーブルビット  
1 = 全ての PORTB プルアップを無効にする  
0 = 各ポートラッチの値に応じて PORTB プルアップを有効にする
- bit 6      **INTEDG0**: 外部割り込み 0 エッジ選択ビット  
1 = 立ち上がりエッジで割り込み  
0 = 立ち下がりエッジで割り込み
- bit 5      **INTEDG1**: 外部割り込み 1 エッジ選択ビット  
1 = 立ち上がりエッジで割り込み  
0 = 立ち下がりエッジで割り込み
- bit 4      **INTEDG2**: 外部割り込み 2 エッジ選択ビット  
1 = 立ち上がりエッジで割り込み  
0 = 立ち下がりエッジで割り込み
- bit 3      **INTEDG3**: 外部割り込み 3 エッジ選択ビット  
1 = 立ち上がりエッジで割り込み  
0 = 立ち下がりエッジで割り込み
- bit 2      **TMR0IP**: TMR0 オーバーフロー割り込み優先度ビット  
1 = 高優先度  
0 = 低優先度
- bit 1      **INT3IP**: INT3 外部割り込み優先度ビット  
1 = 高優先度  
0 = 低優先度
- bit 0      **RBIP**: RB ポート状態変化割り込み優先度ビット  
1 = 高優先度  
0 = 低優先度

**Note:** 割り込み条件が発生すると、対応するイネーブルビットまたはグローバル割り込みイネーブルビットの状態に関わらず、各割り込みのフラグビットがセットされます。ユーザ ソフトウェアは、割り込みを有効にする前に、対応する割り込みフラグビットを確実にクリアしておく必要があります。上記の機能によって、ソフトウェア ポーリングが可能になります。

# PIC18F97J60 ファミリ

## レジスタ 10-3: INTCON3: 割り込み制御レジスタ 3

R/W-1	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INT2IP	INT1IP	INT3IE	INT2IE	INT1IE	INT3IF	INT2IF	INT1IF
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

bit 7	<b>INT2IP:</b> INT2 外部割り込み優先度ビット 1 = 高優先度 0 = 低優先度
bit 6	<b>INT1IP:</b> INT1 外部割り込み優先度ビット 1 = 高優先度 0 = 低優先度
bit 5	<b>INT3IE:</b> INT3 外部割り込みイネーブルビット 1 = INT3 外部割り込みを有効にする 0 = INT3 外部割り込みを無効にする
bit 4	<b>INT2IE:</b> INT2 外部割り込みイネーブルビット 1 = INT2 外部割り込みを有効にする 0 = INT2 外部割り込みを無効にする
bit 3	<b>INT1IE:</b> INT1 外部割り込みイネーブルビット 1 = INT1 外部割り込みを有効にする 0 = INT1 外部割り込みを無効にする
bit 2	<b>INT3IF:</b> INT3 外部割り込みフラグビット 1 = INT3 外部割り込みが発生した (ソフトウェアによるクリアが必要) 0 = INT3 外部割り込みは発生していない
bit 1	<b>INT2IF:</b> INT2 外部割り込みフラグビット 1 = INT2 外部割り込みが発生した (ソフトウェアによるクリアが必要) 0 = INT2 外部割り込みは発生していない
bit 0	<b>INT1IF:</b> INT1 外部割り込みフラグビット 1 = INT1 外部割り込みが発生した (ソフトウェアによるクリアが必要) 0 = INT1 外部割り込みは発生していない

**Note:** 割り込み条件が発生すると、対応するイネーブルビットまたはグローバル割り込みイネーブルビットの状態に関わらず、各割り込みのフラグビットがセットされます。ユーザ ソフトウェアは、割り込みを有効にする前に、対応する割り込みフラグビットを確実にクリアしておく必要があります。上記の機能によって、ソフトウェアポーリングが可能になります。

# PIC18F97J60 ファミリ

## 10.2 PIR レジスタ

PIR レジスタは、各周辺割り込みに対応するフラグビットを含みます。周辺割り込み要因の数が多いため、周辺割り込み要求 (フラグ) レジスタは 3 つ用意されています (PIR1、PIR2、PIR3)。

**Note 1:** 割り込み条件が発生すると、対応するイネーブルビットまたはグローバル割り込みイネーブルビット GIE (INTCON<7>) の状態に関わらず、各割り込みフラグビットがセットされます。

**2:** ユーザ ソフトウェアでは、割り込みを有効にする前と、割り込み処理の完了後に、対応する割り込みフラグビットを確実にクリアしてください。

レジスタ 10-4: PIR1: 周辺割り込み要求 (フラグ) レジスタ 1

R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIF <sup>(1)</sup>	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

- bit 7      **PSPIF:** パラレルスレーブ ポート読み書き割り込みフラグビット<sup>(1)</sup>  
1 = 読み出しまたは書き込み動作が実行された (ソフトウェアによるクリアが必要)  
0 = 読み書き動作は実行されていない
- bit 6      **ADIF:** A/D コンバータ割り込みフラグビット  
1 = A/D 変換が完了した (ソフトウェアによるクリアが必要)  
0 = A/D 変換は完了していない
- bit 5      **RC1IF:** EUSART1 受信割り込みフラグビット  
1 = EUSART1 受信バッファの RCREG1 がフル (RCREG1 が読み出されるとクリアされる)  
0 = EUSART1 受信バッファがエンプティ
- bit 4      **TX1IF:** EUSART1 送信割り込みフラグビット  
1 = EUSART1 送信バッファの TXREG1 がエンプティ (TXREG1 に書き込むとクリアされる)  
0 = EUSART1 送信バッファがフル
- bit 3      **SSP1IF:** MSSP1 割り込みフラグビット  
1 = 送信 / 受信が完了した (ソフトウェアによるクリアが必要)  
0 = 送信 / 受信待ち
- bit 2      **CCP1IF:** ECCP1 割り込みフラグビット  
キャプチャモード:  
1 = TMR1 レジスタのキャプチャが発生した (ソフトウェアによるクリアが必要)  
0 = TMR1 レジスタのキャプチャは発生していない  
コンペアモード:  
1 = TMR1 レジスタのコンペア一致が発生した (ソフトウェアによるクリアが必要)  
0 = TMR1 レジスタのコンペア一致は発生していない  
PWM モード:  
このモードでは未使用
- bit 1      **TMR2IF:** TMR2/PR2 一致割り込みフラグビット  
1 = TMR2 と PR2 が一致した (ソフトウェアによるクリアが必要)  
0 = TMR2 と PR2 は一致していない
- bit 0      **TMR1IF:** TMR1 オーバーフロー割り込みフラグビット  
1 = TMR1 レジスタがオーバーフローした (ソフトウェアによるクリアが必要)  
0 = TMR1 レジスタはオーバーフローしていない

**Note 1:** 100 ピンデバイスにのみ実装されており、利用できるのはマイクロコントローラ モードのみです。

# PIC18F97J60 ファミリ

## レジスタ 10-5: PIR2: 周辺割り込み要求 (フラグ) レジスタ 2

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
OSCFIF	CMIF	ETHIF	r	BCL1IF	—	TMR3IF	CCP2IF
bit 7							bit 0

<b>凡例:</b>	r = 予約済みビット
R = 読み出し可能ビット	W = 書き込み可能ビット
-n = POR 時の値	U = 未実装ビット、「0」として読み出し
	「1」= ビットはセット 「0」= ビットはクリア x = ビットは未知

- bit 7      **OSCFIF:** オシレータ障害割り込みフラグビット  
 1 = システム オシレータに障害が発生し、クロック入力を INTRC に変更した (ソフトウェアによるクリアが必要)  
 0 = システムクロックが動作中
- bit 6      **CMIF:** コンパレータ割り込みフラグビット  
 1 = コンパレータの入力が変化した (ソフトウェアによるクリアが必要)  
 0 = コンパレータの入力は変化していない
- bit 5      **ETHIF:** Ethernet モジュール割り込みフラグビット  
 1 = Ethernet モジュール割り込みイベントが発生した (割り込み要因は EIR レジスタを参照)  
 0 = Ethernet 割り込みイベントは発生していない
- bit 4      **予約済み:** 常に「0」
- bit 3      **BCL1IF:** バスコリジョン割り込みフラグビット (MSSP1 モジュール)  
 1 = バスコリジョンが発生した (ソフトウェアによるクリアが必要)  
 0 = バスコリジョンは発生していない
- bit 2      **未実装:** 「0」として読み出し
- bit 1      **TMR3IF:** TMR3 オーバーフロー割り込みフラグビット  
 1 = TMR3 レジスタがオーバーフローした (ソフトウェアによるクリアが必要)  
 0 = TMR3 レジスタはオーバーフローしていない
- bit 0      **CCP2IF:** ECCP2 割り込みフラグビット  
キャプチャモード:  
 1 = TMR1/TMR3 レジスタのキャプチャが発生した (ソフトウェアによるクリアが必要)  
 0 = TMR1/TMR3 レジスタのキャプチャは発生していない  
コンペアモード:  
 1 = TMR1/TMR3 レジスタのコンペア一致が発生した (ソフトウェアによるクリアが必要)  
 0 = TMR1/TMR3 レジスタのコンペア一致は発生していない  
PWM モード:  
 このモードでは未使用

# PIC18F97J60 ファミリ

## レジスタ 10-6: PIR3: 周辺割り込み要求 (フラグ) レジスタ 3

R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
SSP2IF <sup>(1)</sup>	BCL2IF <sup>(1)</sup>	RC2IF <sup>(2)</sup>	TX2IF <sup>(2)</sup>	TMR4IF	CCP5IF	CCP4IF	CCP3IF
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

- bit 7      **SSP2IF:** MSSP2 割り込みフラグビット<sup>(1)</sup>  
 1 = 送信 / 受信が完了した (ソフトウェアによるクリアが必要)  
 0 = 送信 / 受信待ち
- bit 6      **BCL2IF:** バスコリジョン割り込みフラグビット (MSSP2 モジュール)<sup>(1)</sup>  
 1 = バスコリジョンが発生した (ソフトウェアによるクリアが必要)  
 0 = バスコリジョンは発生していない
- bit 5      **RC2IF:** EUSART2 受信割り込みフラグビット<sup>(2)</sup>  
 1 = EUSART2 受信バッファの RCREG2 がフル (RCREG2 が読み出されるとクリアされる)  
 0 = EUSART2 受信バッファがエンプティ
- bit 4      **TX2IF:** EUSART2 送信割り込みフラグビット<sup>(2)</sup>  
 1 = EUSART2 送信バッファの TXREG2 がエンプティ (TXREG2 に書き込むとクリアされる)  
 0 = EUSART2 送信バッファがフル
- bit 3      **TMR4IF:** TMR4/PR4 一致割り込みフラグビット  
 1 = TMR4 と PR4 が一致した (ソフトウェアによるクリアが必要)  
 0 = TMR4 と PR4 は一致していない
- bit 2      **CCP5IF:** CCP5 割り込みフラグビット  
キャプチャモード:  
 1 = TMR1/TMR3 レジスタのキャプチャが発生した (ソフトウェアによるクリアが必要)  
 0 = TMR1/TMR3 レジスタのキャプチャは発生していない  
コンペアモード:  
 1 = TMR1/TMR3 レジスタのコンペア一致が発生した (ソフトウェアによるクリアが必要)  
 0 = TMR1/TMR3 レジスタのコンペア一致は発生していない  
PWM モード:  
 このモードでは未使用
- bit 1      **CCP4IF:** CCP4 割り込みフラグビット  
キャプチャモード:  
 1 = TMR1/TMR3 レジスタのキャプチャが発生した (ソフトウェアによるクリアが必要)  
 0 = TMR1/TMR3 レジスタのキャプチャは発生していない  
コンペアモード:  
 1 = TMR1/TMR3 レジスタのコンペア一致が発生した (ソフトウェアによるクリアが必要)  
 0 = TMR1/TMR3 レジスタのコンペア一致は発生していない  
PWM モード:  
 このモードでは未使用
- bit 0      **CCP3IF:** ECCP3 割り込みフラグビット  
キャプチャモード:  
 1 = TMR1/TMR3 レジスタのキャプチャが発生した (ソフトウェアによるクリアが必要)  
 0 = TMR1/TMR3 レジスタのキャプチャは発生していない  
コンペアモード:  
 1 = TMR1/TMR3 レジスタのコンペア一致が発生した (ソフトウェアによるクリアが必要)  
 0 = TMR1/TMR3 レジスタのコンペア一致は発生していない  
PWM モード:  
 このモードでは未使用

- Note 1:** 100 ピンデバイスにのみ実装されています。  
**Note 2:** 80 ピン / 100 ピンデバイスにのみ実装されています。



## 10.3 PIE レジスタ

PIE レジスタは、各周辺割り込みに対応するイネーブルビットを含みます。周辺割り込み要因の数が多いため、周辺割り込みイネーブルレジスタは3つ用意されています (PIE1、PIE2、PIE3)。IPEN = 0 の場合、これらの周辺割り込みのいずれを有効にする場合も PEIE ビットをセットする必要があります。

レジスタ 10-7: PIE1: 周辺割り込みイネーブルレジスタ 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIE <sup>(1)</sup>	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

- bit 7      **PSPIE:** パラレルスレーブポート読み書き割り込みイネーブルビット<sup>(1)</sup>  
 1 = 有効にする  
 0 = 無効にする
- bit 6      **ADIE:** A/D コンバータ割り込みイネーブルビット  
 1 = 有効にする  
 0 = 無効にする
- bit 5      **RC1IE:** EUSART1 受信割り込みイネーブルビット  
 1 = 有効にする  
 0 = 無効にする
- bit 4      **TX1IE:** EUSART1 送信割り込みイネーブルビット  
 1 = 有効にする  
 0 = 無効にする
- bit 3      **SSP1IE:** MSSP1 割り込みイネーブルビット  
 1 = 有効にする  
 0 = 無効にする
- bit 2      **CCP1IE:** ECCP1 割り込みイネーブルビット  
 1 = 有効にする  
 0 = 無効にする
- bit 1      **TMR2IE:** TMR2/PR2 一致割り込みイネーブルビット  
 1 = 有効にする  
 0 = 無効にする
- bit 0      **TMR1IE:** TMR1 オーバーフロー割り込みイネーブルビット  
 1 = 有効にする  
 0 = 無効にする

**Note 1:** 100 ピンデバイスにのみ実装されており、利用できるのはマイクロコントローラモード時のみです。

# PIC18F97J60 ファミリ

## レジスタ 10-8: PIE2: 周辺割り込みイネーブル レジスタ 2

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
OSCFIE	CMIE	ETHIE	r	BCL1IE	—	TMR3IE	CCP2IE
bit 7							bit 0

<b>凡例:</b>	r = 予約済みビット		
R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し	
-n = POR 時の値	「1」= ビットはセット	「0」= ビットはクリア	x = ビットは未知

- bit 7      **OSCFIE:** オシレータ障害割り込みイネーブルビット  
1 = 有効にする  
0 = 無効にする
- bit 6      **CMIE:** コンパレータ割り込みイネーブルビット  
1 = 有効にする  
0 = 無効にする
- bit 5      **ETHIE:** Ethernet モジュール割り込みイネーブルビット  
1 = 有効にする  
0 = 無効にする
- bit 4      **予約済み:** 常に「0」
- bit 3      **BCL1IE:** バスコリジョン割り込みイネーブルビット (MSSP1 モジュール)  
1 = 有効にする  
0 = 無効にする
- bit 2      **未実装:** 「0」として読み出し
- bit 1      **TMR3IE:** TMR3 オーバーフロー割り込みイネーブルビット  
1 = 有効にする  
0 = 無効にする
- bit 0      **CCP2IE:** ECCP2 割り込みイネーブルビット  
1 = 有効にする  
0 = 無効にする

# PIC18F97J60 ファミリ

## レジスタ 10-9: PIE3: 周辺割り込みイネーブル レジスタ 3

R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
SSP2IE <sup>(1)</sup>	BCL2IE <sup>(1)</sup>	RC2IE <sup>(2)</sup>	TX2IE <sup>(2)</sup>	TMR4IE	CCP5IE	CCP4IE	CCP3IE
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

- bit 7      **SSP2IE:** MSSP2 割り込みイネーブルビット<sup>(1)</sup>  
1 = 有効にする  
0 = 無効にする
- bit 6      **BCL2IE:** バスコリジョン割り込みイネーブルビット (MSSP2 モジュール)<sup>(1)</sup>  
1 = 有効にする  
0 = 無効にする
- bit 5      **RC2IE:** EUSART2 受信割り込みイネーブルビット<sup>(2)</sup>  
1 = 有効にする  
0 = 無効にする
- bit 4      **TX2IE:** EUSART2 送信割り込みイネーブルビット<sup>(2)</sup>  
1 = 有効にする  
0 = 無効にする
- bit 3      **TMR4IE:** TMR4/PR4 一致割り込みイネーブルビット  
1 = 有効にする  
0 = 無効にする
- bit 2      **CCP5IE:** CCP5 割り込みイネーブルビット  
1 = 有効にする  
0 = 無効にする
- bit 1      **CCP4IE:** CCP4 割り込みイネーブルビット  
1 = 有効にする  
0 = 無効にする
- bit 0      **CCP3IE:** ECCP3 割り込みイネーブルビット  
1 = 有効にする  
0 = 無効にする

- Note 1:** 100 ピンデバイスにのみ実装されています。  
**Note 2:** 80 ピン / 100 ピンデバイスにのみ実装されています。

# PIC18F97J60 ファミリ

## 10.4 IPR レジスタ

IPR レジスタは、各周辺割り込みに対応する優先度ビットを含みます。周辺割り込み要因の数が多いため、周辺割り込み優先度レジスタは3つ用意されています (IPR1、IPR2、IPR3)。優先度ビットを使うには割り込み優先度イネーブルビット(IPEN)をセットする必要があります。

### レジスタ 10-10: IPR1: 周辺割り込み優先度レジスタ 1

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
PSP1P <sup>(1)</sup>	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

- bit 7      **PSP1P:** パラレルスレーブポート読み書き割り込み優先度ビット<sup>(1)</sup>  
1 = 高優先度  
0 = 低優先度
- bit 6      **ADIP:** A/Dコンバータ割り込み優先度ビット  
1 = 高優先度  
0 = 低優先度
- bit 5      **RC1IP:** EUSART1受信割り込み優先度ビット  
1 = 高優先度  
0 = 低優先度
- bit 4      **TX1IP:** EUSART1送信割り込み優先度ビット  
1 = 高優先度  
0 = 低優先度
- bit 3      **SSP1IP:** MSSP1割り込み優先度ビット  
1 = 高優先度  
0 = 低優先度
- bit 2      **CCP1IP:** ECCP1割り込み優先度ビット  
1 = 高優先度  
0 = 低優先度
- bit 1      **TMR2IP:** TMR2/PR2一致割り込み優先度ビット  
1 = 高優先度  
0 = 低優先度
- bit 0      **TMR1IP:** TMR1オーバーフロー割り込み優先度ビット  
1 = 高優先度  
0 = 低優先度

**Note 1:** 100ピンデバイスにのみ実装されており、利用できるのはマイクロコントローラモード時のみです。

# PIC18F97J60 ファミリ

## レジスタ 10-11: IPR2: 周辺割り込み優先度レジスタ 2

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	U-0	R/W-1	R/W-1
OSCFIP	CMIP	ETHIP	r	BCL1IP	—	TMR3IP	CCP2IP
bit 7							bit 0

<b>凡例:</b>	r = 予約済みビット		
R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し	
-n = POR 時の値	「1」= ビットはセット	「0」= ビットはクリア	x = ビットは未知

- bit 7      **OSCFIP:** オシレータ障害割り込み優先度ビット  
1 = 高優先度  
0 = 低優先度
- bit 6      **CMIP:** コンパレータ割り込み優先度ビット  
1 = 高優先度  
0 = 低優先度
- bit 5      **ETHIP:** Ethernet モジュール割り込み優先度ビット  
1 = 高優先度  
0 = 低優先度
- bit 4      **予約済み:** 常に「1」
- bit 3      **BCL1IP:** バスコリジョン割り込み優先度ビット (MSSP1 モジュール)  
1 = 高優先度  
0 = 低優先度
- bit 2      **未実装:** 「0」として読み出し
- bit 1      **TMR3IP:** TMR3 オーバーフロー割り込み優先度ビット  
1 = 高優先度  
0 = 低優先度
- bit 0      **CCP2IP:** ECCP2 割り込み優先度ビット  
1 = 高優先度  
0 = 低優先度

# PIC18F97J60 ファミリ

## レジスタ 10-12: IPR3: 周辺割り込み優先度レジスタ 3

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
SSP2IP <sup>(1)</sup>	BCL2IP <sup>(1)</sup>	RC2IP <sup>(2)</sup>	TX2IP <sup>(2)</sup>	TMR4IP	CCP5IP	CCP4IP	CCP3IP
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

- bit 7      **SSP2IP:** MSSP2 割り込み優先度ビット<sup>(1)</sup>  
1 = 高優先度  
0 = 低優先度
- bit 6      **BCL2IP:** バスコリジョン割り込み優先度ビット (MSSP2 モジュール)<sup>(1)</sup>  
1 = 高優先度  
0 = 低優先度
- bit 5      **RC2IP:** EUSART2 受信割り込み優先度ビット<sup>(2)</sup>  
1 = 高優先度  
0 = 低優先度
- bit 4      **TX2IP:** EUSART2 送信割り込み優先度ビット<sup>(2)</sup>  
1 = 高優先度  
0 = 低優先度
- bit 3      **TMR4IE:** TMR4/PR4 割り込み優先度ビット  
1 = 高優先度  
0 = 低優先度
- bit 2      **CCP5IP:** CCP5 割り込み優先度ビット  
1 = 高優先度  
0 = 低優先度
- bit 1      **CCP4IP:** CCP4 割り込み優先度ビット  
1 = 高優先度  
0 = 低優先度
- bit 0      **CCP3IP:** ECCP3 割り込み優先度ビット  
1 = 高優先度  
0 = 低優先度

- Note 1:** 100 ピンデバイスにのみ実装されています。  
**Note 2:** 80 ピン / 100 ピンデバイスにのみ実装されています。

## 10.5 RCON レジスタ

RCON レジスタは、直近のリセット、またはアイドル/スリープからの復帰の原因を判断するために使うビットを含みます。さらに、割り込み優先度機能を有効にする IPEN ビットも含みます。

レジスタ 10-13: RCON: リセット制御レジスタ

R/W-0	U-0	R/W-1	R/W-1	R-1	R-1	R/W-0	R/W-0
IPEN	—	$\overline{CM}$	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

- bit 7      **IPEN:** 割り込み優先度イネーブルビット  
 1 = 割り込みに対する優先度レベルの設定を有効にする  
 0 = 割り込みに対する優先度レベルの設定を無効にする (PIC16CXXX 互換モード)
- bit 6      **未実装:** 「0」として読み出し
- bit 5      **CM:** コンフィグレーション不一致フラグビット  
 ビット動作の詳細は、[レジスタ 5-1](#) を参照してください。
- bit 4      **RI:** RESET 命令フラグビット  
 ビット動作の詳細は、[レジスタ 5-1](#) を参照してください。
- bit 3      **TO:** ウォッチドッグ タイマ タイムアウト フラグビット  
 ビット動作の詳細は、[レジスタ 5-1](#) を参照してください。
- bit 2      **PD:** パワーダウン検出フラグビット  
 ビット動作の詳細は、[レジスタ 5-1](#) を参照してください。
- bit 1      **POR:** パワーオン リセット ステータスビット <sup>(2)</sup>  
 ビット動作の詳細は、[レジスタ 5-1](#) を参照してください。
- bit 0      **BOR:** ブラウンアウト リセット ステータスビット  
 ビット動作の詳細は、[レジスタ 5-1](#) を参照してください。

## 10.6 INTx ピン割り込み

RB0/INT0/FLT0、RB1/INT1、RB2/INT2、RB3/INT3 ピンの外部割り込みはエッジトリガ方式です。INTCON2 レジスタの対応する INTEDGx ビットがセットされている場合 (= 1)、割り込みは立ち上がりエッジでトリガされます。クリアされている場合、立ち下がりエッジでトリガされます。RBx/INTx ピンで有効なエッジが発生すると、対応するフラグビット INTxIF がセットされます。この割り込みは、対応するイネーブルビット INTxIE をクリアする事で無効にできます。フラグビットの INTxIF は、再度この割り込みを有効にする前に、ソフトウェアの割り込みサービスルーチン (ISR) 内でクリアしておく必要があります。

全ての外部割り込み (INT0、INT1、INT2、INT3) は、電力管理モードからプロセッサを復帰させる事ができます。ただし、電力管理モードに移行する前に INTxIE をセットしておいた場合に限りです。グローバル割り込みイネーブルビット GIE をセットすると、プロセッサは復帰後に割り込みベクタに分岐します。

INT1、INT2、INT3 の割り込み優先度は、割り込み優先度ビット INT1IP (INTCON3<6>)、INT2IP (INTCON3<7>)、INT3IP (INTCON2<1>) に設定された値によって決まります。INT0 に対応する優先度ビットはありません。INT0 は常に高優先度の割り込み要因として処理されます。

## 10.7 TMR0 割り込み

8ビットモードの場合(既定値)、TMR0 レジスタでオーバーフローが発生すると (FFh → 00h)、フラグビット TMR0IF がセットされます。16ビットモードの場合、TMR0H:TMR0L レジスタペアでオーバーフローが発生すると (FFFFh → 0000h)、フラグビット TMR0IF がセットされます。これらの割り込みは、イネーブルビット TMR0IE (INTCON<5>) をセット/クリアする事によって有効/無効にできます。Timer0 の割り込み優先度は、割り込み優先度ビット TMR0IP (INTCON2<2>) の値によって決まります。Timer0 モジュールの詳細は、[セクション 12.0「Timer0 モジュール」](#)を参照してください。

## 10.8 PORTB 状態変化割り込み

PORTB<7:4> への入力の状態が変化すると、フラグビット RBIF (INTCON<0>) がセットされます。この割り込みは、イネーブルビット RBIE (INTCON<3>) をセット/クリアする事によって有効/無効にできます。PORTB 状態変化割り込みの優先度は、割り込み優先度ビット RBIF (INTCON2<0>) の値によって決まります。

## 10.9 割り込み処理中のコンテキスト保存

割り込みの処理中、PC のリターンアドレスはスタックに保存されます。さらに、WREG、STATUS、BSR レジスタの値が高速リターンスタック (FSR) に保存されます。割り込みからの高速リターンを使わない場合 ([セクション 6.3「データメモリの構成」](#)参照)、ユーザが割り込みサービスルーチンの入り口で WREG、STATUS、BSR レジスタの値を保存する必要があります。ユーザアプリケーションによっては、その他のレジスタ値の保存も必要になる場合があります。例 10-1 では、割り込みサービスルーチン内で WREG、STATUS、BSR レジスタの値を保存および復元しています。

### 例 10-1: STATUS、WREG、BSR レジスタ値の RAM への保存

```
MOVWF  W_TEMP           ; W_TEMP is in virtual bank
MOVFF  STATUS, STATUS_TEMP ; STATUS_TEMP located anywhere
MOVFF  BSR, BSR_TEMP     ; BSR_TEMP located anywhere
;
; USER ISR CODE
;
MOVFF  BSR_TEMP, BSR     ; Restore BSR
MOVF   W_TEMP, W         ; Restore WREG
MOVFF  STATUS_TEMP, STATUS ; Restore STATUS
```



## 11.0 I/O ポート

選択したデバイスと有効化した機能に応じて、最大 9 つのポートを使えます。I/O ポートピンの一部は代替機能と多重化されており、デバイスの周辺機能を割り当てる事ができます。通常、周辺機能を有効にしている時には、そのピンを汎用 I/O ピンとして使う事はできません。

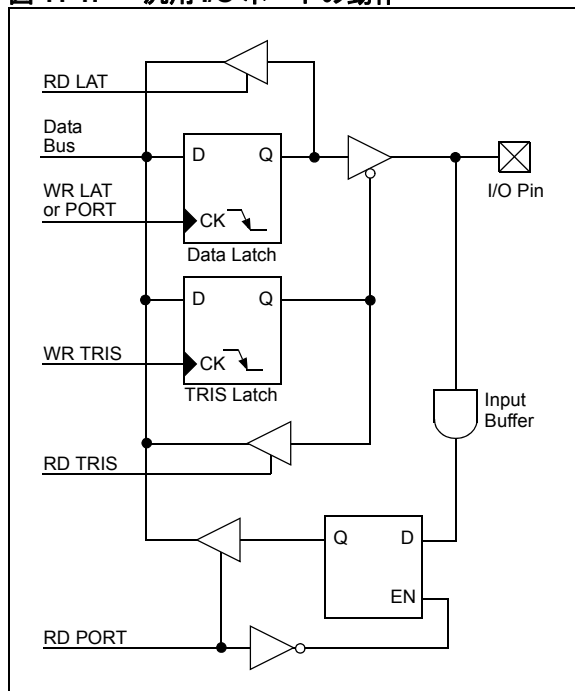
各ポートの動作には以下の3個のレジスタを使います。

- TRIS レジスタ (データ方向レジスタ)
- PORT レジスタ (デバイスピンのレベル読み取り)
- LAT レジスタ (出力ラッチレジスタ)

出力ラッチ (LAT レジスタ) は、I/O ピンが駆動している値の Read-Modify-Write 動作に便利です。

図 11-1 に、他の周辺機能へのインターフェイスを省略した、汎用 I/O ポートの簡易モデルを示します。

図 11-1: 汎用 I/O ポートの動作



## 11.1 I/O ポートピンの機能

アプリケーションを開発する際は、ポートピンの能力を考慮する必要があります。ピンの中には、他のピンよりも出力駆動力が大きいものがあります。同様に、VDD よりも高い入力レベルが許容されるピンもあります。

### 11.1.1 ピンの出力駆動

各種アプリケーションのニーズを満たすため、出力ピンの駆動力はピンのグループごとに異なります。PORTB と PORTC は、LED 等のより大きな負荷を駆動できるように設計されています。外部メモリ インターフェイスポート (PORTD、PORTE、PORTJ) は、中程度の負荷を駆動できるように設計されています。その他のポートは小さな負荷、通常は表示のみを目的として設計されています。表 11-1 に、出力駆動力の一覧を示します。詳細は、セクション 28.0「電気的特性」を参照してください。

表 11-1: 出力駆動レベル

ポート	駆動力	説明
PORTA <sup>(1)</sup>	最小	表示向け
PORTF <sup>(2)</sup>		
PORTG <sup>(2)</sup>		
PORTH <sup>(3)</sup>		
PORTD <sup>(2)</sup>	中程度	表示だけでなく外部メモリとのインターフェイスに十分なレベル
PORTE		
PORTJ <sup>(3)</sup>		
PORTB	大	LED の直接駆動に適したレベル
PORTC		

**Note 1:** RA<1:0> は例外です。これらのピンは LED を直接駆動できます。

**2:** 64 ピン / 80 ピンデバイスには一部だけ実装されています。100 ピンデバイスには全て実装されています。

**3:** 64 ピンデバイスには実装されていません。

# PIC18F97J60 ファミリ

## 11.1.2 入力ピンと電圧に関する注意事項

デバイスの入力として使うピンの許容電圧は、そのピンの入力機能によって異なります。デジタル入力専用ピンは、デジタルロジック回路で一般的な DC5.5 V まで対応できます。これに対して、アナログ入力機能も備えたピンは VDD までしか対応できません。これらのピンには VDD を超える電圧を印加しないでください。表 11-2 に入力電圧耐性の一覧を示します。詳細は、セクション 28.0 「電气的特性」を参照してください。

表 11-2: 入力電圧レベル

ポートまたはピン	入力電圧耐性	説明
PORTA<5,3:0>	VDD	≤ VDD
PORTF<6:1> <sup>(1)</sup>		
PORTH<7:4> <sup>(2)</sup>		
PORTA<4>	5.5 V	VDD を超える電圧に対応。ほとんどの標準デジタルロジックに対応
PORTB<7:0>		
PORTC<7:0>		
PORTD<7:0> <sup>(1)</sup>		
PORTE<7:0>		
PORTF<7>		
PORTG<7:0> <sup>(1)</sup>		
PORTH<3:0> <sup>(2)</sup>		
PORTJ<7:0> <sup>(2)</sup>		

**Note 1:** 64 ピン / 80 ピンデバイスには一部だけ実装されています。100 ピンデバイスには全てが実装されています。

**2:** 64 ピンデバイスでは使えません。

## 11.2 PORTA、TRISA、LATA レジスタ

PORTA は 6 ビット幅の双方向ポートで、全デバイスが全てのピンを実装しています。対応するデータ方向レジスタは TRISA です。TRISA ビットをセットする (= 1) と、対応する PORTA ピンが入力になります (すなわち、対応する出力ドライバがハイインピーダンスモードに移行します)。TRISA ビットをクリアする (= 0) と、対応する PORTA ピンが出力になります (すなわち、出力ラッチの内容が選択したピンに出力されます)。

PORTA レジスタを読み出すとピンの状態が出力され、PORTA レジスタに書き込むとポートラッチにデータが格納されます。

出力ラッチレジスタ (LATA) は、メモリにもマッピングされています。LATA レジスタに対する Read-Modify-Write 動作は、PORTA にラッチされた出力値を読み出して書き戻します。

RA4 ピンは Timer0 モジュールのクロック入力と多重化されていて、RA4/T0CKI ピンとして機能します。その他の PORTA ピンは、アナログ入力 VREF+、VREF- と多重化されています。RA<5:0> ピンの A/D コンバータ入力としての動作を選択するには、ADCON1 レジスタの PCFG<3:0> 制御ビットをクリアまたはセットします。

**Note:** RA5 と RA<3:0> は、全てのリセットでアナログ入力に設定され、「0」として読み出されます。RA4 はデジタル入力に設定されます。

RA4/T0CKI ピンはシュミットトリガ入力です。その他の PORTA ピンは全て入力レベルは TTL、出力はフル CMOS ドライバです。

PORTA ピンをアナログ入力として使う場合も、TRISA レジスタがポートピンの方向を制御します。これらのピンをアナログ入力として使う際は、必ず TRISA レジスタのビットをセットしたままにしてください。

RA0 ピンと RA1 ピンは、2 つの Ethernet LED インジケータ用の出力としても設定できます。そのように設定した場合に PORTA ピンの中で大きな出力駆動力を持つのは、これら 2 つのピンだけです。

ポートの幅は 6 ビットしかありませんが、RJPU として PORTA<7> が実装されています。これは PORTJ の弱プルアップ制御ビットです。同様に、LATA<7:6> ビットも実装されています。これらもラッチビットではなく、PORTD と PORTE のプルアップ制御ビットの RDPU と REPU です。これらのビットをセットすると、対応するポートの弱プルアップが有効になります。TRISA<7:6> ビットに対応するポートピンは使われないため、未実装です。

### 例 11-1: PORTA の初期化

```
CLRF   PORTA    ; Initialize PORTA by
                ; clearing output
                ; data latches
CLRF   LATA     ; Alternate method
                ; to clear output
                ; data latches
MOVLW  07h     ; Configure A/D
MOVWF  ADCON1  ; for digital inputs
MOVWF  07h     ; Configure comparators
MOVWF  CMCON   ; for digital input
MOVLW  0CFh   ; Value used to
                ; initialize data
                ; direction
MOVWF  TRISA   ; Set RA<3:0> as inputs
                ; RA<5:4> as outputs
```

# PIC18F97J60 ファミリ

表 11-3: PORTA の機能

ピン名	機能	TRIS の設定	I/O	I/O タイプ	説明
RA0/LEDA/AN0	RA0	0	O	DIG	LATA<0> のデータ出力です。アナログ入力の影響は受けません。
		1	I	TTL	PORTA<0> のデータ入力です。アナログ入力を有効にした場合、無効になります。
	LEDA	0	O	DIG	Ethernet LEDA の出力です。デジタルデータよりも優先されます。
	AN0	1	I	ANA	A/D 入力チャンネル 0 です。POR 時の既定値入力設定です。デジタル出力には影響を与えません。
RA1/LEDB/AN1	RA1	0	O	DIG	LATA<1> のデータ出力です。アナログ入力の影響は受けません。
		1	I	TTL	PORTA<1> のデータ入力です。アナログ入力を有効にした場合、無効になります。
	LEDB	0	O	DIG	Ethernet LEDB の出力です。デジタルデータよりも優先されます。
	AN1	1	I	ANA	A/D 入力チャンネル 1 です。POR 時の既定値入力設定です。デジタル出力には影響を与えません。
RA2/AN2/VREF-	RA2	0	O	DIG	LATA<2> のデータ出力です。アナログ入力の影響は受けません。CVREF 出力を有効にした場合、無効になります。
		1	I	TTL	PORTA<2> のデータ入力です。アナログ機能を有効にした場合、無効になります。CVREF 出力を有効にした場合、無効になります。
	AN2	1	I	ANA	A/D 入力チャンネル 2 とコンパレータ C2+ 入力です。POR 時の既定値入力設定です。アナログ出力の影響は受けません。
	VREF-	1	I	ANA	A/D とコンパレータの Low 側参照電圧入力です。
RA3/AN3/VREF+	RA3	0	O	DIG	LATA<3> のデータ出力です。アナログ入力の影響は受けません。
		1	I	TTL	PORTA<3> のデータ入力です。アナログ入力を有効にした場合、無効になります。
	AN3	1	I	ANA	A/D 入力チャンネル 3 です。POR 時の既定値入力設定です。
	VREF+	1	I	ANA	A/D の High 側参照電圧入力です。
RA4/T0CKI	RA4	0	O	DIG	LATA<4> のデータ出力です。
		1	I	ST	PORTA<4> のデータ入力です。POR 時の既定値設定です。
	T0CKI	x	I	ST	Timer0 クロック入力です。
RA5/AN4	RA5	0	O	DIG	LATA<5> のデータ出力です。アナログ入力の影響は受けません。
		1	I	TTL	PORTA<5> のデータ入力です。アナログ入力を有効にした場合、無効になります。
	AN4	1	I	ANA	A/D 入力チャンネル 4 です。POR 時の既定値設定です。

凡例: O = 出力、I = 入力、ANA = アナログ信号、DIG = デジタル出力、ST = シュミットトリガバッファ入力、TTL = TTL バッファ入力、x = ドントケア (この場合、TRIS ビットはポート方向に影響を与えないか優先されます)

表 11-4: PORTA に関連するレジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	リセット値の記載ページ
PORTA	RJPU <sup>(1)</sup>	-	RA5	RA4	RA3	RA2	RA1	RA0	72
LATA	RDPU	REPU	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0	72
TRISA	-	-	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	71
ADCON1	-	-	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	70

凡例: - = 未実装ビット、「0」として読み出されます。網掛けの部分は PORTA では使いません。

Note 1: 80 ピン / 100 ピンデバイスにのみ実装されています。

## 11.3 PORTB、TRISB、LATB レジスタ

PORTB は、8 ビット幅の双方向ポートで、全デバイスに全てのピンが実装されています。対応するデータ方向レジスタは TRISB です。TRISB ビットをセットする (= 1) と、対応する PORTB ピンが入力になります (すなわち、対応する出力ドライバがハイインピーダンスモードに移行します)。TRISB ビットをクリアする (= 0) と、対応する PORTB ピンが出力になります (すなわち、出力ラッチの内容が選択したピンに出力されます)。PORTB の全てのピンはデジタル専用で、最大許容電圧は 5.5 V です。

出力ラッチレジスタ (LATB) は、メモリにもマッピングされています。LATB レジスタに対する Read-Modify-Write 動作は、PORTB にラッチされた出力値を読み出して書き戻します。

### 例 11-2: PORTB の初期化

```
CLRF   PORTB   ; Initialize PORTB by
           ; clearing output
           ; data latches
CLRF   LATB    ; Alternate method
           ; to clear output
           ; data latches
MOVLW  0CFh   ; Value used to
           ; initialize data
           ; direction
MOVWF  TRISB  ; Set RB<3:0> as inputs
           ; RB<5:4> as outputs
           ; RB<7:6> as inputs
```

各 PORTB ピンは内部弱プルアップ機能を備えています。1 つの制御ビットで全てのプルアップを有効にできます。それには RBPU (INTCON2<7>) ビットをクリアします。ポートピンを出力に設定すると、弱プルアップは自動的に OFF になります。全てのリセットによってプルアップは無効になります。

PORTB ピンのうち 4 つ (RB<7:4>) には、状態変化割り込み機能があります。この割り込みを生成できるのは、入力として設定されているピンだけです (つまり、出力に設定されている RB<7:4> ピンは、いずれも状態変化割り込みの比較動作から除外されます)。RB<7:4> の入力ピンは、直近の PORTB 読み出しでラッチされた古い値と比較されます。RB<7:4> の「不一致」出力の全てを OR 演算して、RB ポート状態変化割り込みフラグビット RBIF (INTCON<0>) を生成します。

この割り込みによってデバイスを電力管理モードから復帰させる事ができます。割り込みサービスルーチン内で、以下の方法で割り込みをクリアします。

- PORTB に対する任意の読み書き動作 (MOVFF (ANY), PORTB 命令による場合を除く)。これにより、不一致条件は解除される
- フラグビット RBIF のクリア

不一致条件が続く場合、RBIF ビットもセットされたままです。PORTB を読み出すと不一致条件が終了し、フラグビットの RBIF をクリアできるようになります。

状態変化割り込みは、キー押下操作による復帰と、PORTB を状態変化割り込み機能のみに使う場合に推奨します。状態変化割り込み機能使用中の PORTB に対するポーリングは推奨しません。

拡張マイクロコントローラ モードで動作中の 100 ピンデバイスの場合、CCP2MX コンフィグレーション ビットをクリアする事で、RB3 を ECCP2 モジュールと拡張 PWM 出力 2A の代替周辺機能ピンに設定できます。デバイスがマイクロコントローラ モードで動作している場合、ECCP2 の代替割り当ては RE7 です。他の ECCP2 設定と同様、目的とする動作に合わせて TRISB<3> ビットを適切に設定する必要があります。

# PIC18F97J60 ファミリ

表 11-5: PORTB の機能

ピン名	機能	TRIS の設定	I/O	I/O タイプ	説明
RB0/INT0/FLT0	RB0	0	O	DIG	LATB<0> のデータ出力です。
		1	I	TTL	PORTB<0> のデータ入力です。RBPU ビットをクリアした場合、弱プルアップされます。
	INT0	1	I	ST	外部割り込み 0 入力です。
	FLT0	1	I	ST	拡張 PWM フォルト入力 (ECCP1 モジュール) です。ソフトウェアで有効にします。
RB1/INT1	RB1	0	O	DIG	LATB<1> のデータ出力です。
		1	I	TTL	PORTB<1> のデータ入力です。RBPU ビットをクリアした場合、弱プルアップされます。
	INT1	1	I	ST	外部割り込み 1 入力です。
RB2/INT2	RB2	0	O	DIG	LATB<2> のデータ出力です。
		1	I	TTL	PORTB<2> のデータ入力です。RBPU ビットをクリアした場合、弱プルアップされます。
	INT2	1	I	ST	外部割り込み 2 入力です。
RB3/INT3/ ECCP2/P2A	RB3	0	O	DIG	LATB<3> のデータ出力です。
		1	I	TTL	PORTB<3> のデータ入力です。RBPU ビットをクリアした場合、弱プルアップされます。
	INT3	1	I	ST	外部割り込み 3 入力です。
	ECCP2 <sup>(1)</sup>	0	O	DIG	ECCP2 のコンペア出力と PWM 出力です。ポートデータよりも優先されます。
		1	I	ST	ECCP2 キャプチャ入力です。
P2A <sup>(1)</sup>	0	O	DIG	ECCP2 拡張 PWM 出力、チャンネル A です。拡張 PWM シャットダウン時に 3 ステートに設定する事も可能です。ポートデータよりも優先されます。	
RB4/KBI0	RB4	0	O	DIG	LATB<4> のデータ出力です。
		1	I	TTL	PORTB<4> のデータ入力です。RBPU ビットをクリアした場合、弱プルアップされます。
	KBI0	1	I	TTL	ピン状態変化割り込みです。
RB5/KBI1	RB5	0	O	DIG	LATB<5> のデータ出力です。
		1	I	TTL	PORTB<5> のデータ入力です。RBPU ビットをクリアした場合、弱プルアップされます。
	KBI1	1	I	TTL	ピン状態変化割り込みです。
RB6/KBI2/PGC	RB6	0	O	DIG	LATB<6> のデータ出力です。
		1	I	TTL	PORTB<6> のデータ入力です。RBPU ビットをクリアした場合、弱プルアップされます。
	KBI2	1	I	TTL	ピン状態変化割り込みです。
	PGC	x	I	ST	ICSP™ および ICD 動作時のシリアル実行 (ICSP) クロック入力です <sup>(2)</sup> 。
RB7/KBI3/PGD	RB7	0	O	DIG	LATB<7> のデータ出力です。
		1	I	TTL	PORTB<7> のデータ入力です。RBPU ビットをクリアした場合、弱プルアップされます。
	KBI3	1	I	TTL	ピン状態変化割り込みです。
	PGD	x	O	DIG	ICSP および ICD 動作時のシリアル実行データ出力です <sup>(2)</sup> 。
		x	I	ST	ICSP および ICD 動作時のシリアル実行データ入力です <sup>(2)</sup> 。

凡例: O = 出力、I = 入力、DIG = デジタル出力、ST = シュミットトリガ パッファ入力、TTL = TTL パッファ入力、x = ドントケア (この場合、TRIS ビットはポート方向に影響を与えないか優先されます)

Note 1: CCP2MX コンフィグレーション ビットがクリアされている場合の ECCP2/P2A の代替ピン割り当てです (拡張マイクロコントローラ モードの 100 ピンデバイスの場合)。既定値の割り当ては RC1 です。

2: ICSP または ICD を有効にした場合、その他のピン機能は全て無効になります。

# PIC18F97J60 ファミリ

---

表 11-6: PORTB に関連するレジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	リセット 値の記載 ページ
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	72
LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	72
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	71
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	69
INTCON2	RBP $\bar{U}$	INTEDG0	INTEDG1	INTEDG2	INTEDG3	TMR0IP	INT3IP	RBIP	69
INTCON3	INT2IP	INT1IP	INT3IE	INT2IE	INT1IE	INT3IF	INT2IF	INT1IF	69

凡例： 網掛けの部分は PORTB では使いません。

## 11.4 PORTC、TRISC、LATC レジスタ

PORTC は、8 ビット幅の双方向ポートで、全デバイスに全てのピンが実装されています。対応するデータ方向レジスタは TRISC です。TRISC ビットをセットする (= 1) と、対応する PORTC ピンが入力になります (すなわち、対応する出力ドライバがハイインピーダンスモードに移行します)。TRISC ビットをクリアする (= 0) と、対応する PORTC ピンが出力になります (すなわち、出力ラッチの内容が選択したピンに出力されます)。PORTC の RC2 ~ RC7 ピンだけはデジタル専用ピンであり、最大 5.5 V の入力電圧に対応します。

出力ラッチレジスタ (LATC) は、メモリにもマッピングされています。LATC レジスタに対する Read-Modify-Write 動作は、PORTC にラッチされた出力値を読み出して書き戻します。

PORTC は複数の周辺機能と多重化されています (表 11-7)。ピンはシュミットトリガ入力バッファを備えています。通常、RC1 はコンフィグレーションビット CCP2MX によって ECCP2 モジュールと拡張 PWM 出力 P2A の既定値の周辺機能ピンに設定されます (既定値では CCP2MX = 1)。

周辺機能を有効にする場合、各 PORTC ピンの TRIS ビットの定義に注意してください。周辺機能の中には、TRIS ビットの設定をオーバーライドしてピンを入力または出力に設定するものがあります。対応する周辺機能のセクションを参照して、TRIS ビットを適切に設定してください。

**Note:** これらのピンは、全てのデバイスリセットでデジタル入力に設定されます。

TRISC レジスタの内容は、周辺機能によるオーバーライドの影響を受けます。周辺機能がピンをオーバーライドしている場合でも、TRISC の読み出し結果はレジスタの格納値です。

### 例 11-3: PORTC の初期化

```
CLRF    PORTC    ; Initialize PORTC by
                ; clearing output
                ; data latches
CLRF    LATC     ; Alternate method
                ; to clear output
                ; data latches
MOVLW  0CFh     ; Value used to
                ; initialize data
                ; direction
MOVWF  TRISC    ; Set RC<3:0> as inputs
                ; RC<5:4> as outputs
                ; RC<7:6> as inputs
```

# PIC18F97J60 ファミリ

表 11-7: PORTC の機能

ピン名	機能	TRIS の設定	I/O	I/O タイプ	説明
RC0/T1OSO/ T13CKI	RC0	0	O	DIG	LATC<0> のデータ出力です。
		1	I	ST	PORTC<0> のデータ入力です。
	T1OSO	x	O	ANA	Timer1 オシレータ出力です。Timer1 オシレータが有効な場合、有効になります。デジタル I/O は無効になります。
	T13CKI	1	I	ST	Timer1/Timer3 カウンタ入力です。
RC1/T1OSI/ ECCP2/P2A	RC1	0	O	DIG	LATC<1> のデータ出力です。
		1	I	ST	PORTC<1> のデータ入力です。
	T1OSI	x	I	ANA	Timer1 オシレータ入力です。Timer1 オシレータが有効な場合、有効になります。デジタル I/O は無効になります。
	ECCP2 <sup>(1)</sup>	0	O	DIG	ECCP2 のコンペア出力と PWM 出力です。ポートデータよりも優先されます。
		1	I	ST	ECCP2 キャプチャ入力です。
P2A <sup>(1)</sup>	0	O	DIG	ECCP2 拡張 PWM 出力、チャンネル A です。拡張 PWM シャットダウン時に 3 ステートに設定する事も可能です。ポートデータよりも優先されます。	
RC2/ECCP1/ P1A	RC2	0	O	DIG	LATC<2> のデータ出力です。
		1	I	ST	PORTC<2> のデータ入力です。
	ECCP1	0	O	DIG	ECCP1 のコンペア出力と PWM 出力です。ポートデータよりも優先されます。
		1	I	ST	ECCP1 キャプチャ入力です。
	P1A	0	O	DIG	ECCP1 拡張 PWM 出力、チャンネル A です。拡張 PWM シャットダウン時に 3 ステートに設定する事も可能です。ポートデータよりも優先されます。
RC3/SCK1/ SCL1	RC3	0	O	DIG	LATC<3> のデータ出力です。
		1	I	ST	PORTC<3> のデータ入力です。
	SCK1	0	O	DIG	SPI クロック出力 (MSSP1 モジュール) です。ポートデータよりも優先されます。
		1	I	ST	SPI クロック入力 (MSSP1 モジュール) です。
	SCL1	0	O	DIG	I <sup>2</sup> C™ クロック出力 (MSSP1 モジュール) です。ポートデータよりも優先されます。
		1	I	ST	I <sup>2</sup> C クロック入力 (MSSP1 モジュール) です。入力タイプはモジュールの設定によって決まります。
RC4/SDI1/ SDA1	RC4	0	O	DIG	LATC<4> のデータ出力です。
		1	I	ST	PORTC<4> のデータ入力です。
	SDI1	1	I	ST	SPI データ入力 (MSSP1 モジュール) です。
	SDA1	1	O	DIG	I <sup>2</sup> C データ出力 (MSSP1 モジュール) です。ポートデータよりも優先されます。
1		I	ST	I <sup>2</sup> C データ入力 (MSSP1 モジュール) です。入力タイプはモジュールの設定によって決まります。	
RC5/SDO1	RC5	0	O	DIG	LATC<5> のデータ出力です。
		1	I	ST	PORTC<5> のデータ入力です。
	SDO1	0	O	DIG	SPI データ出力 (MSSP1 モジュール) です。ポートデータよりも優先されます。
RC6/TX1/CK1	RC6	0	O	DIG	LATC<6> のデータ出力です。
		1	I	ST	PORTC<6> のデータ入力です。
	TX1	1	O	DIG	同期シリアルデータ出力 (EUSART1 モジュール) です。ポートデータよりも優先されます。
	CK1	1	O	DIG	同期シリアルデータ入力 (EUSART1 モジュール) です。ユーザは入力として設定する必要があります。
1		I	ST	同期シリアルクロック入力 (EUSART1 モジュール) です。	
RC7/RX1/DT1	RC7	0	O	DIG	LATC<7> のデータ出力です。
		1	I	ST	PORTC<7> のデータ入力です。
	RX1	1	I	ST	非同期シリアル受信データ入力 (EUSART1 モジュール) です。
	DT1	1	O	DIG	同期シリアルデータ出力 (EUSART1 モジュール) です。ポートデータよりも優先されます。
		1	I	ST	同期シリアルデータ入力 (EUSART1 モジュール) です。ユーザは入力として設定する必要があります。

凡例: O = 出力、I = 入力、ANA = アナログ信号、DIG = デジタル出力、ST = シュミットトリガ バッファ入力、  
x = ドントケア (この場合、TRIS ビットはポート方向に影響を与えないか優先されます)

Note 1: CCP2MX コンフィグレーション ビットをセットした場合の ECCP2/P2A の既定値のピン割り当てです。



# PIC18F97J60 ファミリ

表 11-8: PORTC に関連するレジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	リセット 値の記載 ページ
PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	<a href="#">72</a>
LATC	LATC7	LATC6	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	<a href="#">72</a>
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	<a href="#">71</a>

# PIC18F97J60 ファミリ

## 11.5 PORTD、TRISD、LATD レジスタ

PORTD は双方向ポートとして、以下の 2 つのビット幅で実装されています。

- 64 ピン / 80 ピンデバイス：3 ビット (RD<2:0>)
- 100 ピンデバイス：8 ビット (RD<7:0>)

対応するデータ方向レジスタは TRISD です。TRISD ビットをセットする (= 1) と、対応する PORTD ピンが入力になります (すなわち、対応する出力ドライバがハイインピーダンスモードに移行します)。TRISD ビットをクリアする (= 0) と、対応する PORTD ピンが出力になります (すなわち、出力ラッチの内容が選択したピンに出力されます)。PORTD の全てのピンはデジタル専用で、最大許容電圧は 5.5 V です。

出力ラッチレジスタ (LATD) は、メモリにもマッピングされています。LATD レジスタに対する Read-Modify-Write 動作は、PORTD にラッチされた出力値を読み出して書き戻します。

PORTD では、全てのピンにシュミットトリガ入力バッファが実装されています。各ピンは、個別に入力または出力として設定できます。

**Note:** これらのピンは、全てのデバイスリセットでデジタル入力に設定されます。

100 ピンデバイスの場合、PORTD は外部メモリ インターフェイスの一部としてシステムバスと多重化されています。I/O ポートやその他の機能は、EBDIS ビット (MEMCON<7>) をセットして、インターフェイスを無効にした場合にのみ使えます。インターフェイスを有効にした場合の PORTD は、多重化されたアドレス / データバスの下位バイト (AD<7:0>) です。TRISD ビットもオーバーライドされます。

各 PORTD ピンは内部弱プルアップ機能を備えています。1 つの制御ビットで全てのプルアップを有効にできます。それには、RDPU ビット (LATA<7>) をセットします。ポートピンを出力に設定すると、弱プルアップは自動的に OFF になります。全てのデバイスリセットによってプルアップは無効になります。

100 ピンデバイスの PORTD は、PSPMODE 制御ビット (PSPCON<4>) をセットする事で、8 ビット幅のパラレル マイクロプロセッサ ポートとして機能するようにも設定できます。このモードでは、パラレルポート データが他のデジタル I/O よりも優先されます (外部メモリ インターフェイスは除く)。パラレルポートが有効な場合の入力バッファは TTL です。詳細は [セクション 11.11 「パラレルスレーブポート \(PSP\)」](#) を参照してください。

### 例 11-4: PORTD の初期化

```
CLRF   PORTD   ; Initialize PORTD by
           ; clearing output
           ; data latches
CLRF   LATD    ; Alternate method
           ; to clear output
           ; data latches
MOVLW  0CFh   ; Value used to
           ; initialize data
           ; direction
MOVWF  TRISD   ; Set RD<3:0> as inputs
           ; RD<5:4> as outputs
           ; RD<7:6> as inputs
```

# PIC18F97J60 ファミリ

表 11-9: PORTD の機能

ピン名	機能	TRIS の設定	I/O	I/O タイプ	説明
RD0/AD0/PSP0 (RD0/P1B)	RD0	0	O	DIG	LATD<0> のデータ出力です。
		1	I	ST	PORTD<0> のデータ入力です。RDPU ビットをセットした場合、弱プルアップされます。
	AD0 <sup>(1)</sup>	x	O	DIG	外部メモリ インターフェイスのアドレス / データ Bit 0 の出力です <sup>(2)</sup> 。
		x	I	TTL	外部メモリ インターフェイスのデータ Bit 0 の入力です <sup>(2)</sup> 。
	PSP0 <sup>(1)</sup>	x	O	DIG	PSP 読み出しの出力データ (LATD<0>) です。ポートデータよりも優先されます。
		x	I	TTL	PSP 書き込みのデータ入力です。
	P1B <sup>(3)</sup>	0	O	DIG	ECCP1 拡張 PWM 出力、チャンネル B です。ポートデータと PSP データよりも優先されます。拡張 PWM シャットダウン時に 3 ステートに設定する事も可能です。
	RD1/AD1/PSP1 (RD1/ECCP3/ P3A)	RD1	0	O	DIG
1			I	ST	PORTD<1> のデータ入力です。RDPU ビットをセットした場合、弱プルアップされます。
AD1 <sup>(1)</sup>		x	O	DIG	外部メモリ インターフェイスのアドレス / データ Bit 1 の出力です <sup>(2)</sup> 。
		x	I	TTL	外部メモリ インターフェイスのデータ Bit 1 の入力です <sup>(2)</sup> 。
PSP1 <sup>(1)</sup>		x	O	DIG	PSP 読み出しの出力データ (LATD<1>) です。ポートデータよりも優先されます。
		x	I	TTL	PSP 書き込みのデータ入力です。
ECCP3 <sup>(3)</sup>		0	O	DIG	ECCP3 のコンペアおよび PWM 出力です。ポートデータよりも優先されます。
		1	I	ST	ECCP3 キャプチャ入力です。
P3A <sup>(3)</sup>	0	O	DIG	ECCP3 拡張 PWM 出力、チャンネル A です。ポートデータと PSP データよりも優先されます。拡張 PWM シャットダウン時に 3 ステートに設定する事も可能です。	
RD2/AD2/PSP2 (RD2/CCP4/ P3D)	RD2	0	O	DIG	LATD<2> のデータ出力です。
		1	I	ST	PORTD<2> のデータ入力です。RDPU ビットをセットした場合、弱プルアップされます。
	AD2 <sup>(1)</sup>	x	O	DIG	外部メモリ インターフェイスのアドレス / データ Bit 2 の出力です <sup>(2)</sup> 。
		x	I	TTL	外部メモリ インターフェイスのデータ Bit 2 の入力です <sup>(2)</sup> 。
	PSP2 <sup>(1)</sup>	x	O	DIG	PSP 読み出しの出力データ (LATD<2>) です。ポートデータよりも優先されます。
		x	I	TTL	PSP 書き込みのデータ入力です。
	CCP4 <sup>(3)</sup>	0	O	DIG	CCP4 のコンペア出力と PWM 出力です。ポートデータよりも優先されます。
		1	I	ST	CCP4 キャプチャ入力です。
P3D <sup>(3)</sup>	0	O	DIG	ECCP3 拡張 PWM 出力、チャンネル D です。ポートデータと PSP データよりも優先されます。拡張 PWM シャットダウン時に 3 ステートに設定する事も可能です。	
RD3/AD3/ PSP3 <sup>(1)</sup>	RD3 <sup>(1)</sup>	0	O	DIG	LATD<3> のデータ出力です。
		1	I	ST	PORTD<3> のデータ入力です。RDPU ビットをセットした場合、弱プルアップされます。
	AD3 <sup>(1)</sup>	x	O	DIG	外部メモリ インターフェイスのアドレス / データ Bit 3 の出力です <sup>(2)</sup> 。
		x	I	TTL	外部メモリ インターフェイスのデータ Bit 3 の入力です <sup>(2)</sup> 。
	PSP3 <sup>(1)</sup>	x	O	DIG	PSP 読み出しの出力データ (LATD<3>) です。ポートデータよりも優先されます。
		x	I	TTL	PSP 書き込みのデータ入力です。
RD4/AD4/ PSP4/SDO2 <sup>(1)</sup>	RD4 <sup>(1)</sup>	0	O	DIG	LATD<4> のデータ出力です。
		1	I	ST	PORTD<4> のデータ入力です。RDPU ビットをセットした場合、弱プルアップされます。
	AD4 <sup>(1)</sup>	x	O	DIG	外部メモリ インターフェイスのアドレス / データ Bit 4 の出力です <sup>(2)</sup> 。
		x	I	TTL	外部メモリ インターフェイスのデータ Bit 4 の入力です <sup>(2)</sup> 。
	PSP4 <sup>(1)</sup>	x	O	DIG	PSP 読み出しの出力データ (LATD<4>) です。ポートデータよりも優先されます。
		x	I	TTL	PSP 書き込みのデータ入力です。
SDO2 <sup>(1)</sup>	0	O	DIG	SPI データ出力 (MSSP2 モジュール) です。ポートデータよりも優先されます。	

凡例: O = 出力、I = 入力、DIG = デジタル出力、ST = シュミットトリガ バッファ入力、TTL = TTL バッファ入力、  
x = ドントケア (この場合、TRIS ビットはポート方向に影響を与えないか優先されます)

- Note 1: これらの機能またはポートピンは 100 ピンデバイスにのみ実装されています。  
2: 外部メモリ インターフェイスの I/O は、他の全てのデジタル I/O と PSP I/O よりも優先されます。  
3: このピンのこれらの機能は、64 ピンデバイスにのみ実装されています。その他の全てのデバイスでは RE6/RH7 (P1B)、RG0 (ECCP3/P3A)、RG3 (CCP4/P3D) と多重化されています。

# PIC18F97J60 ファミリ

表 11-9: PORTD の機能 ( 続き )

ピン名	機能	TRIS の設定	I/O	I/O タイプ	説明
RD5/AD5/ PSP5/SDI2/ SDA2 <sup>(1)</sup>	RD5 <sup>(1)</sup>	0	O	DIG	LATD<5> のデータ出力です。
		1	I	ST	PORTD<5> のデータ入力です。RDPU ビットをセットした場合、弱プルアップされます。
	AD5 <sup>(1)</sup>	x	O	DIG	外部メモリ インターフェイスのアドレス / データ Bit 5 の出力です <sup>(2)</sup> 。
		x	I	TTL	外部メモリ インターフェイスのデータ Bit 5 の入力です <sup>(2)</sup> 。
	PSP5 <sup>(1)</sup>	x	O	DIG	PSP 読み出しの出力データ (LATD<5>) です。ポートデータよりも優先されます。
		x	I	TTL	PSP 書き込みのデータ入力です。
	SDI2 <sup>(1)</sup>	1	I	ST	SPI データ入力 (MSSP2 モジュール) です。
	SDA2 <sup>(1)</sup>	1	O	DIG	I <sup>2</sup> C™ データ出力 (MSSP2 モジュール) です。ポートデータよりも優先されます。
1		I	ST	I <sup>2</sup> C データ入力 (MSSP2 モジュール) です。入力タイプはモジュールの設定によって決まります。	
RD6/AD6/ PSP6/SCK2/ SCL2 <sup>(1)</sup>	RD6 <sup>(1)</sup>	0	O	DIG	LATD<6> のデータ出力です。
		1	I	ST	PORTD<6> のデータ入力です。RDPU ビットをセットした場合、弱プルアップされます。
	AD6 <sup>(1)</sup>	x	O	DIG-3	外部メモリ インターフェイスのアドレス / データ Bit 6 の出力です <sup>(2)</sup> 。
		x	I	TTL	外部メモリ インターフェイスのデータ Bit 6 の入力です <sup>(2)</sup> 。
	PSP6 <sup>(1)</sup>	x	O	DIG	PSP 読み出しの出力データ (LATD<6>) です。ポートデータよりも優先されます。
		x	I	TTL	PSP 書き込みのデータ入力です。
	SCK2 <sup>(1)</sup>	0	O	DIG	SPI クロック出力 (MSSP2 モジュール) です。ポートデータよりも優先されます。
		1	I	ST	SPI クロック入力 (MSSP2 モジュール) です。
SCL2 <sup>(1)</sup>	0	O	DIG	I <sup>2</sup> C クロック出力 (MSSP2 モジュール) です。ポートデータよりも優先されます。	
	1	I	ST	I <sup>2</sup> C クロック入力 (MSSP2 モジュール) です。入力タイプはモジュールの設定によって決まります。	
RD7/AD7/ PSP7/SS2 <sup>(1)</sup>	RD7 <sup>(1)</sup>	0	O	DIG	LATD<7> のデータ出力です。
		1	I	ST	PORTD<7> のデータ入力です。RDPU ビットをセットした場合、弱プルアップされます。
	AD7 <sup>(1)</sup>	x	O	DIG	外部メモリ インターフェイスのアドレス / データ Bit 7 の出力です <sup>(2)</sup> 。
		x	I	TTL	外部メモリ インターフェイスのデータ Bit 7 の入力です <sup>(2)</sup> 。
	PSP7 <sup>(1)</sup>	x	O	DIG	PSP 読み出しの出力データ (LATD<7>) です。ポートデータよりも優先されます。
		x	I	TTL	PSP 書き込みのデータ入力です。
SS2 <sup>(1)</sup>	x	I	TTL	MSSP2 モジュールのスレーブ選択入力です。	

凡例: O = 出力, I = 入力, DIG = デジタル出力, ST = シュミットトリガ バッファ入力, TTL = TTL バッファ入力,  
x = ドントケア (この場合、TRIS ビットはポート方向に影響を与えないか優先されます)

- Note** 1: これらの機能またはポートピンは 100 ピンデバイスにのみ実装されています。  
2: 外部メモリ インターフェイスの I/O は、他の全てのデジタル I/O と PSP I/O よりも優先されます。  
3: このピンのこれらの機能は、64 ピンデバイスにのみ実装されています。その他の全てのデバイスでは RE6/RH7 (P1B)、RG0 (ECCP3/P3A)、RG3 (CCP4/P3D) と多重化されています。

表 11-10: PORTD に関連するレジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	リセット 値の記載 ページ
PORTD	RD7 <sup>(1)</sup>	RD6 <sup>(1)</sup>	RD5 <sup>(1)</sup>	RD4 <sup>(1)</sup>	RD3 <sup>(1)</sup>	RD2	RD1	RD0	72
LATD	LATD7 <sup>(1)</sup>	LATD6 <sup>(1)</sup>	LATD5 <sup>(1)</sup>	LATD4 <sup>(1)</sup>	LATD3 <sup>(1)</sup>	LATD2	LATD1	LATD0	72
TRISD	TRISD7 <sup>(1)</sup>	TRISD6 <sup>(1)</sup>	TRISD5 <sup>(1)</sup>	TRISD4 <sup>(1)</sup>	TRISD3 <sup>(1)</sup>	TRISD2	TRISD1	TRISD0	71
LATA	RDPU	REPU	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0	72

凡例: 網掛けの部分は PORTD では使いません。

- Note** 1: 64 ピン / 80 ピンデバイスでは未実装であり、「0」として読み出されます。

## 11.6 PORTE、TRISE、LATE レジスタ

PORTE は双方向ポートとして、以下の 2 つのビット幅で実装されています。

- 64 ピンデバイス：6 ビット幅 (RE<5:0>)
- 80 ピン/100 ピンデバイス：8 ビット幅 (RE<7:0>)

対応するデータ方向レジスタは TRISE です。TRISE ビットをセットする (= 1) と、対応する PORTE ピンが入力になります (すなわち、対応する出力ドライバがハイインピーダンス モードに移行します)。TRISE ビットをクリアする (= 0) と、対応する PORTE ピンが出力になります (すなわち、出力ラッチの内容が選択したピンに出力されます)。PORTE の全てのピンはデジタル専用で、最大許容電圧は 5.5 V です。

出力ラッチレジスタ (LATE) は、メモリにもマッピングされています。LATE レジスタに対する Read-Modify-Write 動作は、PORTE にラッチされた出力値を読み出して書き戻します。

PORTE では、全てのピンにシュミットトリガ入力バッファが実装されています。各ピンは、個別に入力または出力として設定できます。

**Note:** これらのピンは、全てのデバイスリセットでデジタル入力に設定されます。

100 ピンデバイスの場合、PORTE は外部メモリ インターフェイスの一部であるシステムバスと多重化されています。I/O ポートやその他の機能は、EBDIS ビット (MEMCON<7>) をセットして、インターフェイスを無効にした場合のみ使えます。インターフェイスを有効にした場合の PORTE は、多重化されたアドレス/データバスの上位バイト (AD<15:8>) です。TRISE ビットもオーバーライドされます。

各 PORTE ピンは内部弱プルアップ機能を備えています。1 つの制御ビットで全てのプルアップを有効にできます。それには、REPU ビット (LATA<6>) をセットします。ポートピンを出力に設定すると、弱プルアップは自動的に OFF になります。全てのデバイスリセットによってプルアップは無効になります。

PORTE は ECCP1 と ECCP3 の拡張 PWM 出力の B と C、ECCP2 の出力 B、C、D とも多重化されています。80 ピン/100 ピンデバイスの場合、これらの既定値の割り当ては PORTE<6:0> です。64 ピンデバイスの場合、これらの既定値の割り当ては PORTE<5:0> と PORTD<0> です。80 ピン/100 ピンデバイスの場合、ECCP1 と ECCP3 の出力の多重化は、ECCPMX コンフィグレーション ビットによって制御されます。このビットをクリアすると、PORTH への P1B/P1C と P3B/P3C 出力が再割り当てされます。

マイクロコントローラ モードで動作中の 80 ピン/100 ピンデバイスの場合、ピン RE7 を ECCP2 モジュールと拡張 PWM 出力 2A の代替周辺機能ピンとして設定できます。それには CCP2MX コンフィグレーション ビットをクリアします。

PORTD でパラレルスレーブ ポートを有効にしている場合、PORTE の 3 つのピン (RE0、RE1、RE2) はポートのデジタル制御入力に設定されます。表 11-11 に制御機能の一覧を示します。リコンフィグレーションは、PSPMODE 制御ビット (PSPCON<4>) がセットされると自動的に実行されます。それでもユーザはこれらのピンがデジタル入力に設定されるように、対応する TRISE ビットがセットされている事を確認する必要があります。

### 例 11-5: PORTE の初期化

```
CLRF    PORTE    ; Initialize PORTE by
                ; clearing output
                ; data latches
CLRF    LATE     ; Alternate method
                ; to clear output
                ; data latches
MOVLW  03h      ; Value used to
                ; initialize data
                ; direction
MOVWF  TRISE    ; Set RE<1:0> as inputs
                ; RE<7:2> as outputs
```

# PIC18F97J60 ファミリ

表 11-11: PORTE の機能

ピン名	機能	TRIS の設定	I/O	I/O タイプ	説明
RE0/AD8/RD/ P2D	RE0	0	O	DIG	LATE<0> のデータ出力です。
		1	I	ST	PORTE<0> のデータ入力です。REPU ビットをセットした場合、弱プルアップされます。
	AD8 <sup>(1)</sup>	x	O	DIG	外部メモリ インターフェイスのアドレス / データ Bit 8 の出力です <sup>(2)</sup> 。
		x	I	TTL	外部メモリ インターフェイスのデータ Bit 8 の入力です <sup>(2)</sup> 。
	R $\overline{D}$ <sup>(6)</sup>	1	I	TTL	パラレルスレーブ ポート読み出しイネーブル制御入力です。
P2D	0	O	DIG	ECCP2 拡張 PWM 出力、チャンネル D です。ポートデータと PSP データよりも優先されます。拡張 PWM シャットダウン時に 3 ステートに設定する事も可能です。	
RE1/AD9/WR/ P2C	RE1	0	O	DIG	LATE<1> のデータ出力です。
		1	I	ST	PORTE<1> のデータ入力です。REPU ビットをセットした場合、弱プルアップされます。
	AD9 <sup>(1)</sup>	x	O	DIG	外部メモリ インターフェイスのアドレス / データ Bit 9 の出力です <sup>(2)</sup> 。
		x	I	TTL	外部メモリ インターフェイスのデータ Bit 9 の入力です <sup>(2)</sup> 。
	W $\overline{R}$ <sup>(6)</sup>	1	I	TTL	パラレルスレーブ ポート書き込みイネーブル制御入力です。
P2C	0	O	DIG	ECCP2 拡張 PWM 出力、チャンネル C です。ポートデータと PSP データよりも優先されます。拡張 PWM シャットダウン時に 3 ステートに設定する事も可能です。	
RE2/AD10/ $\overline{CS}$ / P2B	RE2	0	O	DIG	LATE<2> のデータ出力です。
		1	I	ST	PORTE<2> のデータ入力です。REPU ビットをセットした場合、弱プルアップされます。
	AD10 <sup>(1)</sup>	x	O	DIG	外部メモリ インターフェイスのアドレス / データ Bit 10 の出力です <sup>(2)</sup> 。
		x	I	TTL	外部メモリ インターフェイスのデータ Bit 10 の入力です <sup>(2)</sup> 。
	$\overline{CS}$ <sup>(6)</sup>	1	I	TTL	パラレルスレーブ ポートチップセレクト制御入力です。
P2B	0	O	DIG	ECCP2 拡張 PWM 出力、チャンネル B です。ポートデータと PSP データよりも優先されます。拡張 PWM シャットダウン時に 3 ステートに設定する事も可能です。	
RE3/AD11/ P3C	RE3	0	O	DIG	LATE<3> のデータ出力です。
		1	I	ST	PORTE<3> のデータ入力です。REPU ビットをセットした場合、弱プルアップされます。
	AD11 <sup>(1)</sup>	x	O	DIG	外部メモリ インターフェイスのアドレス / データ Bit 11 の出力です <sup>(2)</sup> 。
		x	I	TTL	外部メモリ インターフェイスのデータ Bit 11 の入力です <sup>(2)</sup> 。
	P3C <sup>(3)</sup>	0	O	DIG	ECCP3 拡張 PWM 出力、チャンネル C です。ポートデータと PSP データよりも優先されます。拡張 PWM シャットダウン時に 3 ステートに設定する事も可能です。
RE4/AD12/ P3B	RE4	0	O	DIG	LATE<4> のデータ出力です。
		1	I	ST	PORTE<4> のデータ入力です。REPU ビットをセットした場合、弱プルアップされます。
	AD12 <sup>(1)</sup>	x	O	DIG	外部メモリ インターフェイスのアドレス / データ Bit 12 の出力です <sup>(2)</sup> 。
		x	I	TTL	外部メモリ インターフェイスのデータ Bit 12 の入力です <sup>(2)</sup> 。
	P3B <sup>(3)</sup>	0	O	DIG	ECCP3 拡張 PWM 出力、チャンネル B です。ポートデータと PSP データよりも優先されます。拡張 PWM シャットダウン時に 3 ステートに設定する事も可能です。

凡例: O = 出力, I = 入力, DIG = デジタル出力, ST = シュミットトリガ パッファ入力, TTL = TTL パッファ入力,  
x = ドントケア (この場合, TRIS ビットはポート方向に影響を与えないか優先されます)

- Note**
- 1: EMB 機能は 100 ピンデバイスにのみ実装されています。
  - 2: 外部メモリ インターフェイスの I/O は、他の全てのデジタル I/O と PSP I/O よりも優先されます。
  - 3: ECCPMX コンフィグレーション ビットをセットした場合の P1B/P1C と P3B/P3C の既定値の割り当てです (80 ピン / 100 ピンデバイス)。
  - 4: 64 ピンデバイスには実装されていません。
  - 5: CCP2MX コンフィグレーション ビットがクリアされている場合の ECCP2/P2A の代替ピン割り当てです (マイクロコントローラ モードの 80 ピン / 100 ピンデバイスの場合)。
  - 6: 64 ピン / 80 ピンデバイスには実装されていません。

# PIC18F97J60 ファミリ

表 11-11: PORTE の機能 (続き)

ピン名	機能	TRIS の設定	I/O	I/O タイプ	説明
RE5/AD13/ P1C	RE5	0	O	DIG	LATE<5> のデータ出力です。
		1	I	ST	PORTE<5> のデータ入力です。REPU ビットをセットした場合、弱プルアップされます。
	AD13 <sup>(1)</sup>	x	O	DIG	外部メモリ インターフェイスのアドレス / データ Bit 13 の出力です <sup>(2)</sup> 。
		x	I	TTL	外部メモリ インターフェイスのデータ Bit 13 の入力です <sup>(2)</sup> 。
P1C <sup>(3)</sup>	0	O	DIG	ECCP1 拡張 PWM 出力、チャンネル C です。ポートデータと PSP データよりも優先されます。拡張 PWM シャットダウン時に 3 ステートに設定する事も可能です。	
RE6/AD14/ P1B <sup>(4)</sup>	RE6	0	O	DIG	LATE<6> のデータ出力です。
		1	I	ST	PORTE<6> のデータ入力です。REPU ビットをセットした場合、弱プルアップされます。
	AD14 <sup>(1)</sup>	x	O	DIG	外部メモリ インターフェイスのアドレス / データ Bit 14 の出力です <sup>(2)</sup> 。
		x	I	TTL	外部メモリ インターフェイスのデータ Bit 14 の入力です <sup>(2)</sup> 。
P1B <sup>(3)</sup>	0	O	DIG	ECCP1 拡張 PWM 出力、チャンネル B です。ポートデータと PSP データよりも優先されます。拡張 PWM シャットダウン時に 3 ステートに設定する事も可能です。	
RE7/AD15/ ECCP2/P2A <sup>(4)</sup>	RE7	0	O	DIG	LATE<7> のデータ出力です。
		1	I	ST	PORTE<7> のデータ入力です。REPU ビットをセットした場合、弱プルアップされます。
	AD15 <sup>(1)</sup>	x	O	DIG	外部メモリ インターフェイスのアドレス / データ Bit 15 の出力です <sup>(2)</sup> 。
		x	I	TTL	外部メモリ インターフェイスのデータ Bit 15 の入力です <sup>(2)</sup> 。
	ECCP2 <sup>(5)</sup>	0	O	DIG	ECCP2 のコンペア出力と PWM 出力です。ポートデータよりも優先されます。
		1	I	ST	ECCP2 キャプチャ入力です。
P2A <sup>(5)</sup>	0	O	DIG	ECCP2 拡張 PWM 出力、チャンネル A です。ポートデータと PSP データよりも優先されます。拡張 PWM シャットダウン時に 3 ステートに設定する事も可能です。	

凡例: O = 出力、I = 入力、DIG = デジタル出力、ST = シュミットトリガ バッファ入力、TTL = TTL バッファ入力、x = ドントケア (この場合、TRIS ビットはポート方向に影響を与えないか優先されます)

- Note
- 1: EMB 機能は 100 ピンデバイスにのみ実装されています。
  - 2: 外部メモリ インターフェイスの I/O は、他の全てのデジタル I/O と PSP I/O よりも優先されます。
  - 3: ECCPMX コンフィグレーション ビットをセットした場合の P1B/P1C と P3B/P3C の既定値の割り当てです (80 ピン / 100 ピンデバイス)。
  - 4: 64 ピンデバイスには実装されていません。
  - 5: CCP2MX コンフィグレーション ビットがクリアされている場合の ECCP2/P2A の代替ピン割り当てです (マイクロコントローラ モードの 80 ピン / 100 ピンデバイスの場合)。
  - 6: 64 ピン / 80 ピンデバイスには実装されていません。

表 11-12: PORTE に関連するレジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	リセット 値の記載 ページ
PORTE	RE7 <sup>(1)</sup>	RE6 <sup>(1)</sup>	RE5	RE4	RE3	RE2	RE1	RE0	72
LATE	LATE7 <sup>(1)</sup>	LATE6 <sup>(1)</sup>	LATE5	LATE4	LATE3	LATE2	LATE1	LATE0	72
TRISE	TRISE7 <sup>(1)</sup>	TRISE6 <sup>(1)</sup>	TRISE5	TRISE4	TRISE3	TRISE2	TRISE1	TRISE0	71
LATA	RDPU	REPU	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0	72

凡例: - = 未実装ビット、「0」として読み出されます。網掛けの部分は PORTE では使いません。

Note 1: 64 ピンデバイスでは未実装であり、「0」として読み出されます。

# PIC18F97J60 ファミリ

## 11.7 PORTF、LATF、TRISF レジスタ

PORTF は双方向ポートとして、以下の 2 つのビット幅で実装されています。

- 64 ピン / 80 ピンデバイス : 7 ビット幅 (RF<7:1>)
- 100 ピンデバイス : 8 ビット幅 (RF<7:0>)

対応するデータ方向レジスタは TRISF です。TRISF ビットをセットする (= 1) と、対応する PORTF ピンが入力になります (すなわち、対応する出力ドライバがハイインピーダンスモードに移行します)。TRISF ビットをクリアする (= 0) と、対応する PORTF ピンが出力になります (すなわち、出力ラッチの内容が選択したピンに出力されます)。PORTF のピン 7 はアナログ入力機能を持たず、最大 5.5 V の入力電圧を許容する唯一のピンです。

出力ラッチレジスタ (LATF) は、メモリにもマッピングされています。LATF レジスタに対する Read-Modify-Write 動作は、PORTF にラッチされた出力値を読み出して書き戻します。

PORTF では、全てのピンにシュミットトリガ入力バッファが実装されています。各ピンは、個別に入力または出力として設定できます。

PORTF は、A/D コンバータ、コンパレータ入力、コンパレータ出力をはじめとする、複数のアナログ周辺機能と多重化されています。RF1 ~ RF6 ピンは、CMCON レジスタの適切なビットを設定する事で、コンパレータの入力または出力として使えます。RF<6:1> をデジタル入力として使うには、コンパレータも OFF にする必要があります。

**Note 1:** デバイスリセット時、RF<6:1> ピンはアナログ入力に設定され、「0」として読み出されます。

**2:** PORTF をデジタル I/O に設定するには、コンパレータを OFF にして、ADCON1 の値を設定します。

### 例 11-6: PORTF の初期化

```
CLRF    PORTF    ; Initialize PORTF by
                ; clearing output
                ; data latches
CLRF    LATF     ; Alternate method
                ; to clear output
                ; data latches

MOVLW   07h     ;
MOVWF   CMCON   ; Turn off comparators
MOVLW   0Fh     ;
MOVWF   ADCON1  ; Set PORTF as digital I/O
MOVLW   0CEh   ; Value used to
                ; initialize data
                ; direction
MOVWF   TRISF   ; Set RF3:RF1 as inputs
                ; RF5:RF4 as outputs
                ; RF7:RF6 as inputs
```



# PIC18F97J60 ファミリ

表 11-13: PORTF の機能

ピン名	機能	TRIS の設定	I/O	I/O タイプ	説明
RF0/AN5 <sup>(1)</sup>	RF0 <sup>(1)</sup>	0	O	DIG	LATF<0> のデータ出力です。アナログ入力の影響は受けません。
		1	I	ST	PORTF<0> のデータ入力です。アナログ入力を有効にした場合、無効になります。
	AN5 <sup>(1)</sup>	1	I	ANA	A/D 入力チャンネル 5 です。POR 時の既定値設定です。
RF1/AN6/ C2OUT	RF1	0	O	DIG	LATF<1> のデータ出力です。アナログ入力の影響は受けません。
		1	I	ST	PORTF<1> のデータ入力です。アナログ入力を有効にした場合、無効になります。
	AN6	1	I	ANA	A/D 入力チャンネル 6 です。POR 時の既定値設定です。
	C2OUT	0	O	DIG	コンパレータ 2 の出力です。ポートデータよりも優先されます。
RF2/AN7/ C1OUT	RF2	0	O	DIG	LATF<2> のデータ出力です。アナログ入力の影響は受けません。
		1	I	ST	PORTF<2> のデータ入力です。アナログ入力を有効にした場合、無効になります。
	AN7	1	I	ANA	A/D 入力チャンネル 7 です。POR 時の既定値設定です。
	C1OUT	0	O	TTL	コンパレータ 1 の出力です。ポートデータよりも優先されます。
RF3/AN8	RF3	0	O	DIG	LATF<3> のデータ出力です。アナログ入力の影響は受けません。
		1	I	ST	PORTF<3> のデータ入力です。アナログ入力を有効にした場合、無効になります。
	AN8	1	I	ANA	A/D 入力チャンネル 8 とコンパレータ C2+ 入力です。POR 時の既定値入力設定です。アナログ出力の影響は受けません。
RF4/AN9	RF4	0	O	DIG	LATF<4> のデータ出力です。アナログ入力の影響は受けません。
		1	I	ST	PORTF<4> のデータ入力です。アナログ入力を有効にした場合、無効になります。
	AN9	1	I	ANA	A/D 入力チャンネル 9 とコンパレータ C2- 入力です。POR 時の既定値入力設定です。デジタル出力には影響を与えません。
RF5/AN10/ CVREF	RF5	0	O	DIG	LATF<5> のデータ出力です。アナログ入力の影響は受けません。CVREF 出力を有効にした場合、無効になります。
		1	I	ST	PORTF<5> のデータ入力です。アナログ入力を有効にした場合、無効になります。CVREF 出力を有効にした場合、無効になります。
	AN10	1	I	ANA	A/D 入力チャンネル 10 とコンパレータ C1+ 入力です。POR 時の既定値入力設定です。
	CVREF	x	O	ANA	コンパレータの参照電圧出力です。この機能を有効にすると、デジタル I/O は無効になります。
RF6/AN11	RF6	0	O	DIG	LATF<6> のデータ出力です。アナログ入力の影響は受けません。
		1	I	ST	PORTF<6> のデータ入力です。アナログ入力を有効にした場合、無効になります。
	AN11	1	I	ANA	A/D 入力チャンネル 11 とコンパレータ C1- 入力です。POR 時の既定値入力設定です。デジタル出力には影響を与えません。
RF7/SS1	RF7	0	O	DIG	LATF<7> のデータ出力です。
		1	I	ST	PORTF<7> のデータ入力です。
	SS1	1	I	TTL	MSSP1 モジュールのスレーブ選択入力です。

凡例: O = 出力、I = 入力、ANA = アナログ信号、DIG = デジタル出力、ST = シュミットトリガバッファ入力、TTL = TTL バッファ入力、  
x = ドントケア (この場合、TRIS ビットはポート方向に影響を与えないか優先されます)

Note 1: 100 ピンデバイスにのみ実装されています。

表 11-14: PORTF に関連するレジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	リセット 値の記載 ページ
PORTF	RF7	RF6	RF5	RF4	RF3	RF2	RF1	RF0 <sup>(1)</sup>	72
LATF	LATF7	LATF6	LATF5	LATF4	LATF3	LATF2	LATF1	LATF0 <sup>(1)</sup>	72
TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	TRISF0 <sup>(1)</sup>	71
ADCON1	-	-	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	70
CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	70
CVRCON	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	70

凡例: - = 未実装ビット、「0」として読み出されます。網掛けの部分は PORTF では使いません。

Note 1: 100 ピンデバイスにのみ実装されています。

# PIC18F97J60 ファミリ

## 11.8 PORTG、TRISG、LATG レジスタ

デバイスの種類に応じて、PORTG は双方向ポートとして以下の3つのビット幅のいずれかで実装されています。

- 64 ピンデバイス：1 ビット幅 (RG<4>)
- 80 ピンデバイス：5 ビット幅 (RG<4:0>)
- 100 ピンデバイス：8 ビット幅 (RG<7:0>)

対応するデータ方向レジスタは TRISG です。TRISG ビットをセットする (= 1) と、対応する PORTG ピンが入力になります (すなわち、対応する出力ドライバがハイインピーダンス モードに移行します)。TRISG ビットをクリアする (= 0) と、対応する PORTG ピンが出力になります (すなわち、出力ラッチの内容が選択したピンに出力されます)。PORTG の全てのピンはデジタル専用で、最大許容電圧は 5.5 V です。

出力ラッチレジスタ (LATG) は、メモリにもマッピングされています。LATG レジスタに対する Read-Modify-Write 動作は、PORTG にラッチされた出力値を読み出して書き戻します。

80 ピン / 100 ピンデバイスの PORTG は EUSART2 機能と多重化されています (表 11-15)。PORTG ピンはシュミットトリガ入力バッファを備えています。

周辺機能を有効にする場合、各 PORTG ピンの TRIS ビットの定義に注意してください。周辺機能の中には、TRIS ビットの設定をオーバーライドしてピンを出力に設定するもの、入力に設定するものがあります。ユーザは、該当する周辺機能のセクションを参照して、TRIS ビットを適切に設定してください。ピンのオーバーライド値は TRIS レジスタに読み込まれません。このため周辺機能によるオーバーライドを心配せずに TRIS レジスタの Read-Modify-Write を実行できます。

### 例 11-7: PORTG の初期化

```
CLRF   PORTG   ; Initialize PORTG by
           ; clearing output
           ; data latches
CLRF   LATG    ; Alternate method
           ; to clear output
           ; data latches
MOVLW  04h    ; Value used to
           ; initialize data
           ; direction
MOVWF  TRISG  ; Set RG1:RG0 as outputs
           ; RG2 as input
           ; RG4:RG3 as inputs
```

# PIC18F97J60 ファミリ

表 11-15: PORTG の機能

ピン名	機能	TRIS の設定	I/O	I/O タイプ	説明
RG0/ECCP3/ P3A <sup>(1)</sup>	RG0 <sup>(1)</sup>	0	O	DIG	LATG<0> のデータ出力です。
		1	I	ST	PORTG<0> のデータ入力です。
	ECCP3 <sup>(1)</sup>	0	O	DIG	ECCP3 のコンペアおよび PWM 出力です。ポートデータよりも優先されます。
		1	I	ST	ECCP3 キャプチャ入力です。
	P3A <sup>(1)</sup>	0	O	DIG	ECCP3 拡張 PWM 出力、チャンネル A です。ポートデータと PSP データよりも優先されます。拡張 PWM シャットダウン時に 3 ステートに設定する事も可能です。
RG1/TX2/ CK2 <sup>(1)</sup>	RG1 <sup>(1)</sup>	0	O	DIG	LATG<1> のデータ出力です。
		1	I	ST	PORTG<1> のデータ入力です。
	TX2 <sup>(1)</sup>	1	O	DIG	同期シリアルデータ出力 (EUSART2 モジュール) です。ポートデータよりも優先されます。
		1	O	DIG	同期シリアルデータ入力 (EUSART2 モジュール) です。ユーザは入力として設定する必要があります。
	CK2 <sup>(1)</sup>	1	O	DIG	同期シリアルデータ入力 (EUSART2 モジュール) です。ユーザは入力として設定する必要があります。
		1	I	ST	同期シリアルクロック入力 (EUSART2 モジュール) です。
RG2/RX2/ DT2 <sup>(1)</sup>	RG2 <sup>(1)</sup>	0	O	DIG	LATG<2> のデータ出力です。
		1	I	ST	PORTG<2> のデータ入力です。
	RX2 <sup>(1)</sup>	1	I	ST	非同期シリアル受信データ入力 (EUSART2 モジュール) です。
		1	O	DIG	同期シリアルデータ出力 (EUSART2 モジュール) です。ポートデータよりも優先されます。
	DT2 <sup>(1)</sup>	1	O	DIG	同期シリアルデータ出力 (EUSART2 モジュール) です。ポートデータよりも優先されます。
		1	I	ST	同期シリアルデータ入力 (EUSART2 モジュール) です。ユーザは入力として設定する必要があります。
RG3/CCP4/ P3D <sup>(1)</sup>	RG3 <sup>(1)</sup>	0	O	DIG	LATG<3> のデータ出力です。
		1	I	ST	PORTG<3> のデータ入力です。
	CCP4 <sup>(1)</sup>	0	O	DIG	CCP4 のコンペア出力と PWM 出力です。ポートデータよりも優先されます。
		1	I	ST	CCP4 キャプチャ入力です。
	P3D <sup>(1)</sup>	0	O	DIG	ECCP3 拡張 PWM 出力、チャンネル D です。ポートデータと PSP データよりも優先されます。拡張 PWM シャットダウン時に 3 ステートに設定する事も可能です。
RG4/CCP5/ P1D	RG4	0	O	DIG	LATG<4> のデータ出力です。
		1	I	ST	PORTG<4> のデータ入力です。
	CCP5	0	O	DIG	CCP5 のコンペア出力と PWM 出力です。ポートデータよりも優先されます。
		1	I	ST	CCP5 キャプチャ入力です。
	P1D	0	O	DIG	ECCP1 拡張 PWM 出力、チャンネル D です。ポートデータと PSP データよりも優先されます。拡張 PWM シャットダウン時に 3 ステートに設定する事も可能です。
RG5 <sup>(2)</sup>	RG5 <sup>(2)</sup>	0	O	DIG	LATG<0> のデータ出力です。
		1	I	ST	PORTG<0> のデータ入力です。
RG6 <sup>(2)</sup>	RG6 <sup>(2)</sup>	0	O	DIG	LATG<0> のデータ出力です。
		1	I	ST	PORTG<0> のデータ入力です。
RG7 <sup>(2)</sup>	RG7 <sup>(2)</sup>	0	O	DIG	LATG<0> のデータ出力です。
		1	I	ST	PORTG<0> のデータ入力です。

凡例: O = 出力、I = 入力、DIG = デジタル出力、ST = シュミットトリガバッファ入力、  
 × = ドントケア (この場合、TRIS ビットはポート方向に影響を与えないか優先されます)

Note 1: 80 ピン / 100 ピンデバイスにのみ実装されています。  
 2: 100 ピンデバイスにのみ実装されています。

表 11-16: PORTG に関連するレジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	リセット 値の記載 ページ
PORTG	RG7 <sup>(1)</sup>	RG6 <sup>(1)</sup>	RG5 <sup>(1)</sup>	RG4	RG3 <sup>(2)</sup>	RG2 <sup>(2)</sup>	RG1 <sup>(2)</sup>	RG0 <sup>(2)</sup>	72
LATG	LATG7 <sup>(1)</sup>	LATG6 <sup>(1)</sup>	LATG5 <sup>(1)</sup>	LATG4	LATG3 <sup>(2)</sup>	LATG2 <sup>(2)</sup>	LATG1 <sup>(2)</sup>	LATG0 <sup>(2)</sup>	72
TRISG	TRISG7 <sup>(1)</sup>	TRISG6 <sup>(1)</sup>	TRISG5 <sup>(1)</sup>	TRISG4	TRISG3 <sup>(2)</sup>	TRISG2 <sup>(2)</sup>	TRISG1 <sup>(2)</sup>	TRISG0 <sup>(2)</sup>	71

Note 1: 100 ピンデバイスにのみ実装されています。  
 2: 80 ピン / 100 ピンデバイスにのみ実装されています。

# PIC18F97J60 ファミリ

## 11.9 PORTH、LATH、TRISH レジスタ

**Note:** PORTH は、80 ピン /100 ピンデバイスにのみ実装されています。

PORTH は、8 ビット幅の双方向 I/O ポートです。80 ピン/100ピンデバイスには全てのピンが実装されています。対応するデータ方向レジスタは TRISH です。TRISH ビットをセットする (=1) と、対応する PORTH ピンが入力になります (すなわち、対応する出力ドライバがハイインピーダンス モードに移行します)。TRISH ビットをクリアする (=0) と、対応する PORTH ピンが出力になります (すなわち、出力ラッチの内容が選択したピンに出力されます)。PORTH<3:0> ピンはデジタル専用で、最大許容電圧は 5.5 V です。

出力ラッチレジスタ (LATH) は、メモリにもマッピングされています。LATH レジスタに対する Read-Modify-Write 動作は、PORTH にラッチされた出力値を読み出して書き戻します。

PORTHでは、全てのピンにシュミットトリガ入力バッファが実装されています。各ピンは、個別に入力または出力として設定できます。

外部メモリ インターフェイス有効時、PORTH の 4 本のピンは、インターフェイスの上位アドレスラインとして機能します。インターフェイスからのアドレス出力は、他のデジタル I/O よりも優先されます。対応する TRISH ビットもオーバーライドされます。

PORTH の RH4 ~ RH7 ピンは、アナログ コンバータ入力と多重化されています。これらのピンのアナログ入力としての動作は、ADCON1 レジスタの PCFG<3:0> 制御ビットのクリアまたはセットによって選択します。

PORTH は、ECCP1 モジュールと ECCP3 モジュールの拡張 PWM 出力チャンネル B と C の代替ピンとしても設定できます。それには ECCPMX コンフィグレーション ビットをクリアします。

### 例 11-8: PORTH の初期化

```
CLRF   PORTH      ; Initialize PORTH by
                  ; clearing output
                  ; data latches
CLRF   LATH       ; Alternate method
                  ; to clear output
                  ; data latches
MOVLW  0Fh        ; Configure PORTH as
MOVWF  ADCON1     ; digital I/O
MOVLW  0CFh      ; Value used to
                  ; initialize data
                  ; direction
MOVWF  TRISH      ; Set RH3:RH0 as inputs
                  ; RH5:RH4 as outputs
                  ; RH7:RH6 as inputs
```

# PIC18F97J60 ファミリ

表 11-17: PORTH の機能

ピン名	機能	TRIS の設定	I/O	I/O タイプ	説明
RH0/A16	RH0	0	O	DIG	LATH<0> のデータ出力です。
		1	I	ST	PORTH<0> のデータ入力です。
	A16 <sup>(1)</sup>	x	O	DIG	外部メモリ インターフェイスのアドレスライン 16 です。ポートデータよりも優先されます。
RH1/A17	RH1	0	O	DIG	LATH<1> のデータ出力です。
		1	I	ST	PORTH<1> のデータ入力です。
	A17 <sup>(1)</sup>	x	O	DIG	外部メモリ インターフェイスのアドレスライン 17 です。ポートデータよりも優先されます。
RH2/A18	RH2	0	O	DIG	LATH<2> のデータ出力です。
		1	I	ST	PORTH<2> のデータ入力です。
	A18 <sup>(1)</sup>	x	O	DIG	外部メモリ インターフェイスのアドレスライン 18 です。ポートデータよりも優先されます。
RH3/A19	RH3	0	O	DIG	LATH<3> のデータ出力です。
		1	I	ST	PORTH<3> のデータ入力です。
	A19 <sup>(1)</sup>	x	O	DIG	外部メモリ インターフェイスのアドレスライン 19 です。ポートデータよりも優先されます。
RH4/AN12/P3C	RH4	0	O	DIG	LATH<4> のデータ出力です。
		1	I	ST	PORTH<4> のデータ入力です。
	AN12		I	ANA	A/D 入力チャンネル 12 です。POR 時の既定値入力設定です。デジタル出力には影響を与えません。
	P3C <sup>(2)</sup>	0	O	DIG	ECCP3 拡張 PWM 出力、チャンネル C です。ポートデータと PSP データよりも優先されます。拡張 PWM シャットダウン時に 3 ステートに設定する事も可能です。
RH5/AN13/P3B	RH5	0	O	DIG	LATH<5> のデータ出力です。
		1	I	ST	PORTH<5> のデータ入力です。
	AN13		I	ANA	A/D 入力チャンネル 13 です。POR 時の既定値入力設定です。デジタル出力には影響を与えません。
	P3B <sup>(2)</sup>	0	O	DIG	ECCP3 拡張 PWM 出力、チャンネル B です。ポートデータと PSP データよりも優先されます。拡張 PWM シャットダウン時に 3 ステートに設定する事も可能です。
RH6/AN14/P1C	RH6	0	O	DIG	LATH<6> のデータ出力です。
		1	I	ST	PORTH<6> のデータ入力です。
	AN14		I	ANA	A/D 入力チャンネル 14 です。POR 時の既定値入力設定です。デジタル出力には影響を与えません。
	P1C <sup>(2)</sup>	0	O	DIG	ECCP1 拡張 PWM 出力、チャンネル C です。ポートデータと PSP データよりも優先されます。拡張 PWM シャットダウン時に 3 ステートに設定する事も可能です。
RH7/AN15/P1B	RH7	0	O	DIG	LATH<7> のデータ出力です。
		1	I	ST	PORTH<7> のデータ入力です。
	AN15		I	ANA	A/D 入力チャンネル 15 です。POR 時の既定値入力設定です。デジタル出力には影響を与えません。
	P1B <sup>(2)</sup>	0	O	DIG	ECCP1 拡張 PWM 出力、チャンネル B です。ポートデータと PSP データよりも優先されます。拡張 PWM シャットダウン時に 3 ステートに設定する事も可能です。

凡例: O = 出力、I = 入力、ANA = アナログ信号、DIG = デジタル出力、ST = シュミットトリガ バッファ入力、  
x = ドントケア (この場合、TRIS ビットはポート方向に影響を与えないか優先されます)

Note 1: 80 ピンデバイスには実装されていません。

2: ECCPMX コンフィグレーション ビットがクリアされている場合の P1B/P1C と P3B/P3C の代替ピン割り当てです (80 ピン/100 ピンデバイスのみ)。既定値の割り当ては PORTE<6:3> です。

表 11-18: PORTH に関連するレジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	リセット 値の記載 ページ
PORTH	RH7	RH6	RH5	RH4	RH3	RH2	RH1	RH0	72
LATH	LATH7	LATH6	LATH5	LATH4	LATH3	LATH2	LATH1	LATH0	71
TRISH	TRISH7	TRISH6	TRISH5	TRISH4	TRISH3	TRISH2	TRISH1	TRISH0	71

# PIC18F97J60 ファミリ

## 11.10 PORTJ、TRISJ、LATJ レジスタ

**Note:** PORTJ は、80 ピン /100 ピンデバイスにのみ実装されています。

PORTJ は双方向ポートとして、以下の 2 つのビット幅で実装されています。

- 80 ピンデバイス：2 ビット幅 (RJ<5:4>)
- 100 ピンデバイス：8 ビット幅 (RJ<7:0>)

対応するデータ方向レジスタは TRISJ です。TRISJ ビットをセットする (= 1) と、対応する PORTJ ピンが入力になります (すなわち、対応する出力ドライバがハイインピーダンスモードに移行します)。TRISJ ビットをクリアする (= 0) と、対応する PORTJ ピンが出力になります (すなわち、出力ラッチの内容が選択したピンに出力されます)。PORTJ の全てのピンはデジタル専用で、最大許容電圧は 5.5 V です。

出力ラッチレジスタ (LATJ) は、メモリにもマッピングされています。LATJ レジスタに対する Read-Modify-Write 動作は、PORTJ にラッチされた出力値を読み出して書き戻します。

PORTJ では、全てのピンにシュミットトリガ入力バッファが実装されています。各ピンは、個別に入力または出力として設定できます。

**Note:** これらのピンは、全てのデバイスリセットでデジタル入力に設定されます。

外部メモリ インターフェイス有効時、PORTJ の全てのピンはインターフェイスの制御出力として機能します。EBDIS 制御ビット (MEMCON<7>) をクリアしてインターフェイスを有効にすれば、自動的にこの機能に切り換わります。TRISJ ビットもオーバーライドされます。

各PORTJピンは内部弱プルアップ機能を備えています。1 つの制御ビットで全てのプルアップを有効にできます。それには、RJPU ビット (PORTA<7>) をセットします。ポートピンを出力に設定すると、弱プルアップは自動的に OFF になります。全てのデバイスリセットによってプルアップは無効になります。

### 例 11-9: PORTJ の初期化

```
CLRF   PORTJ   ; Initialize PORTG by
              ; clearing output
              ; data latches
CLRF   LATJ    ; Alternate method
              ; to clear output
              ; data latches
MOVLW  0CFh   ; Value used to
              ; initialize data
              ; direction
MOVWF  TRISJ   ; Set RJ3:RJ0 as inputs
              ; RJ5:RJ4 as output
              ; RJ7:RJ6 as inputs
```

# PIC18F97J60 ファミリ

表 11-19: PORTJ の機能

ピン名	機能	TRIS の設定	I/O	I/O タイプ	説明
RJ0/ALE <sup>(1)</sup>	RJ0 <sup>(1)</sup>	0	O	DIG	LATJ<0> のデータ出力です。
		1	I	ST	PORTJ<0> のデータ入力です。RJPU ビットをセットした場合、弱プルアップされます。
	ALE <sup>(1)</sup>	x	O	DIG	外部メモリ インターフェイスのアドレスラッチ イネーブル制御出力です。デジタル I/O よりも優先されます。
RJ1/ $\overline{OE}$ <sup>(1)</sup>	RJ1 <sup>(1)</sup>	0	O	DIG	LATJ<1> のデータ出力です。
		1	I	ST	PORTJ<1> のデータ入力です。RJPU ビットをセットした場合、弱プルアップされます。
	$\overline{OE}$ <sup>(1)</sup>	x	O	DIG	外部メモリ インターフェイスの出カ イネーブル制御出力です。デジタル I/O よりも優先されます。
RJ2/ $\overline{WR}$ <sup>(1)</sup>	RJ2 <sup>(1)</sup>	0	O	DIG	LATJ<2> のデータ出力です。
		1	I	ST	PORTJ<2> のデータ入力です。RJPU ビットをセットした場合、弱プルアップされます。
	$\overline{WR}$ <sup>(1)</sup>	x	O	DIG	外部メモリバスの下位バイト書き込み制御です。デジタル I/O よりも優先されます。
RJ3/ $\overline{WRH}$ <sup>(1)</sup>	RJ3 <sup>(1)</sup>	0	O	DIG	LATJ<3> のデータ出力です。
		1	I	ST	PORTJ<3> のデータ入力です。RJPU ビットをセットした場合、弱プルアップされます。
	$\overline{WRH}$ <sup>(1)</sup>	x	O	DIG	外部メモリ インターフェイスの上位バイト書き込み制御出力です。デジタル I/O よりも優先されます。
RJ4/BA0	RJ4	0	O	DIG	LATJ<4> のデータ出力です。
		1	I	ST	PORTJ<4> のデータ入力です。RJPU ビットをセットした場合、弱プルアップされます。
	BA0 <sup>(2)</sup>	x	O	DIG	外部メモリ インターフェイスのバイトアドレス 0 制御出力です。デジタル I/O よりも優先されます。
RJ5/ $\overline{CE}$	RJ5	0	O	DIG	LATJ<5> のデータ出力です。
		1	I	ST	PORTJ<5> のデータ入力です。RJPU ビットをセットした場合、弱プルアップされます。
	$\overline{CE}$ <sup>(2)</sup>	x	O	DIG	外部メモリ インターフェイスのチップイネーブル制御出力です。デジタル I/O よりも優先されます。
RJ6/ $\overline{LB}$ <sup>(1)</sup>	RJ6 <sup>(1)</sup>	0	O	DIG	LATJ<6> のデータ出力です。
		1	I	ST	PORTJ<6> のデータ入力です。RJPU ビットをセットした場合、弱プルアップされます。
	$\overline{LB}$ <sup>(1)</sup>	x	O	DIG	外部メモリ インターフェイスの下位バイトイネーブル制御出力です。デジタル I/O よりも優先されます。
RJ7/ $\overline{UB}$ <sup>(1)</sup>	RJ7 <sup>(1)</sup>	0	O	DIG	LATJ<7> のデータ出力です。
		1	I	ST	PORTJ<7> のデータ入力です。RJPU ビットをセットした場合、弱プルアップされます。
	$\overline{UB}$ <sup>(1)</sup>	x	O	DIG	外部メモリ インターフェイスの上位バイトイネーブル制御出力です。デジタル I/O よりも優先されます。

凡例: O = 出力、I = 入力、DIG = デジタル出力、ST = シュミットトリガ バッファ入力、  
x = ドントケア (この場合、TRIS ビットはポート方向に影響を与えないか優先されます)

Note 1: 100 ピンデバイスにのみ実装されています。  
2: EMB 機能は 100 ピンデバイスにのみ実装されています。

表 11-20: PORTJ に関連するレジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	リセット値の 記載ページ
PORTJ	RJ7 <sup>(1)</sup>	RJ6 <sup>(1)</sup>	RJ5	RJ4	RJ3 <sup>(1)</sup>	RJ2 <sup>(1)</sup>	RJ1 <sup>(1)</sup>	RJ0 <sup>(1)</sup>	72
LATJ	LATJ7 <sup>(1)</sup>	LATJ6 <sup>(1)</sup>	LATJ5	LATJ4	LATJ3 <sup>(1)</sup>	LATJ2 <sup>(1)</sup>	LATJ1 <sup>(1)</sup>	LATJ0 <sup>(1)</sup>	71
TRISJ	TRISJ7 <sup>(1)</sup>	TRISJ6 <sup>(1)</sup>	TRISJ5	TRISJ4	TRISJ3 <sup>(1)</sup>	TRISJ2 <sup>(1)</sup>	TRISJ1 <sup>(1)</sup>	TRISJ0 <sup>(1)</sup>	71
PORTA	RJPU	-	RA5	RA4	RA3	RA2	RA1	RA0	72

凡例: - = 未実装ビット、「0」として読み出されます。網掛けの部分は PORTJ では使いません。

Note 1: 100 ピンデバイスにのみ実装されています。

# PIC18F97J60 ファミリ

## 11.11 パラレルスレーブ ポート (PSP)

**Note:** パラレルスレーブ ポートは、100 ピンデバイスにのみ実装されています。

制御ビット PSPMODE (PSPCON<4>) をセットすると、PORTD は 8 ビット幅の平行スレーブ ポートまたはマイクロプロセッサ ポートとしても機能させることができます。RD 制御入力ピン、RE0/AD8/RD/P2D および WR 制御入力ピン、RE1/AD9/WR/P2C によって外部から非同期で読み書きできます。

**Note:** パラレルスレーブ ポートは、マイクロコントローラ モードでのみ使えます。

PSP は 8 ビットのマイクロプロセッサ データバスに直接接続できます。外部のマイクロプロセッサは、PORTD ラッチを 8 ビットのラッチとして読み書きできます。PSPMODE ビットをセットすると、ポートピンの RE0/AD8/RD/P2D を RD 入力、RE1/AD9/WR/P2C を WR 入力、RE2/AD10/CS/P2B を CS (チップセレクト) 入力として有効化できます。この機能を使う場合、TRISE レジスタでデータ方向ビット (TRISE<2:0>) を入力に設定 (ビットをセット) する必要があります。

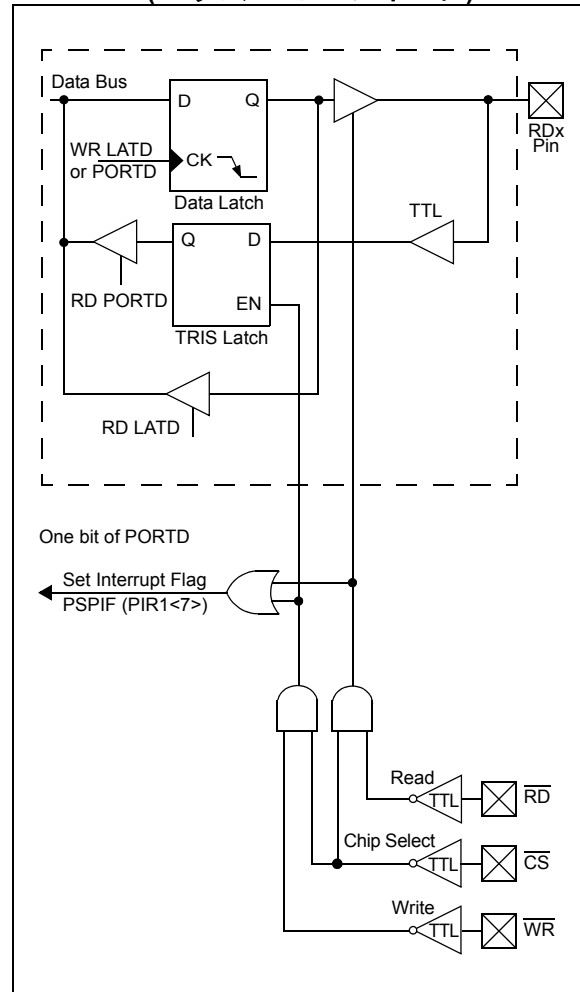
PSP への書き込みは、CS ラインと WR ラインの両方で最初に Low が検出された時点で開始し、いずれか一方で High が検出された時点で終了します。書き込みが終了すると、PSPIF および IBF フラグビットが両方共セットされます。

PSP からの読み出しは、CS ラインと RD ラインの両方で最初に Low が検出された時点で開始します。PORTD のデータが読み出され、OBF ビットがセットされます。ユーザが新しいデータを PORTD に書き込んで OBF をセットした場合、データはただちに読み出されますが、OBF ビットはセットされません。

CS または RD ラインのいずれかで High が検出されると、PORTD ピンは入力ステートに戻り、PSPIF ビットがセットされます。ユーザ アプリケーションは、PSP をサービスする前に、PSPIF がセットされるまで待機する必要があります。PSPIF ビットがセットされれば、IBF および OBF ビットをポーリングして、適切な動作を実行できます。

図 11-3 と 図 11-4 に、書き込みモードと読み出しモードの制御信号のタイミングをそれぞれ示します。

図 11-2: PORTD と PORTE のブロック図 (平行スレーブ ポート)





# PIC18F97J60 ファミリ

## レジスタ 11-1: PSPCON: パラレルスレーブポート制御レジスタ

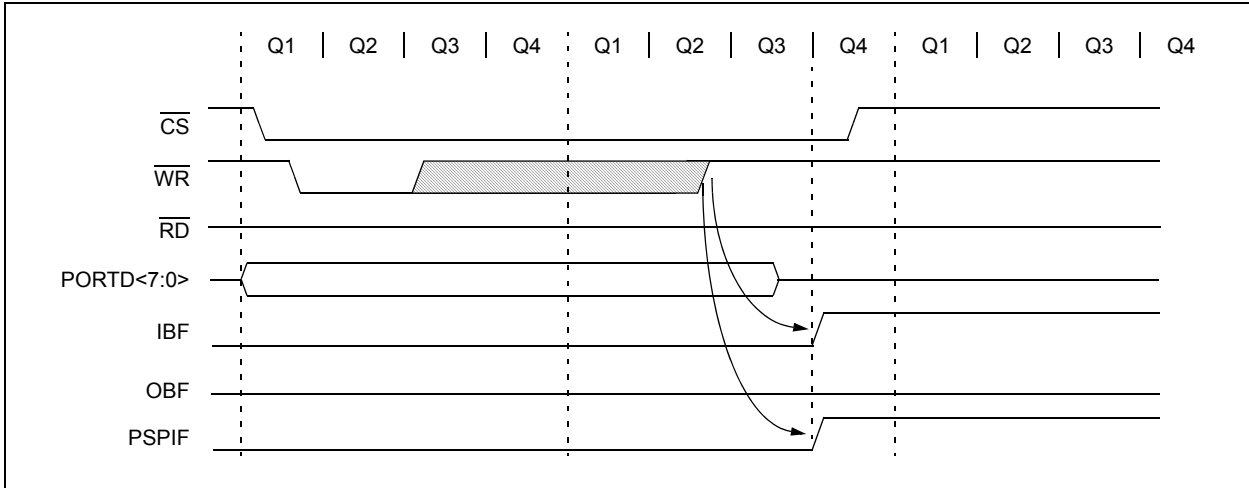
R-0	R-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
IBF	OBF	IBOV	PSPMODE	—	—	—	—
bit 7				bit 0			

### 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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

- bit 7      **IBF:** 入力バッファフルステータスビット  
 1 = 1ワードが受信され、CPUによる読み出しを待機中である  
 0 = ワードを受信していない
- bit 6      **OBF:** 出力バッファフルステータスビット  
 1 = 出力バッファには依然として前回書き込まれたワードが保持されている  
 0 = 出力バッファは読み出し済みである
- bit 5      **IBOV:** 入力バッファオーバーフロー検出ビット  
 1 = 前回の入力ワードの読み出し完了前に書き込みが発生した(ソフトウェアによるクリアが必要である)  
 0 = オーバーフローは発生していない
- bit 4      **PSPMODE:** パラレルスレーブポートモード選択ビット  
 1 = パラレルスレーブポートモード  
 0 = 汎用I/Oモード
- bit 3-0      **未実装:** 「0」として読み出し

図 11-3: パラレルスレーブポートの書き込み波形



# PIC18F97J60 ファミリ

図 11-4: パラレルスレーブポートの読み出し波形

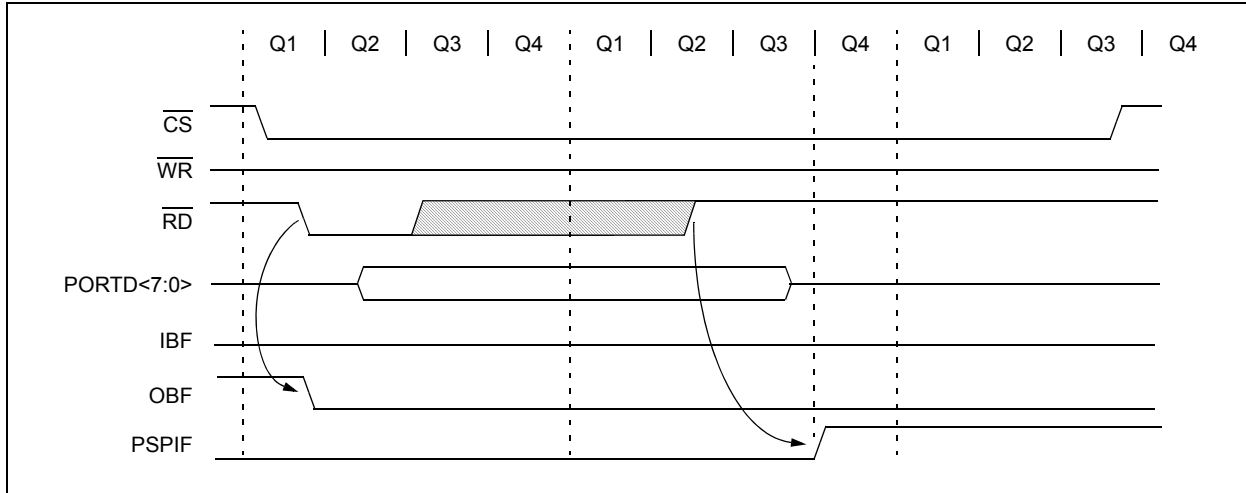


表 11-21: パラレルスレーブポートに関連するレジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	リセット値の記載ページ
PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	72
LATD	LATD7	LATD6	LATD5	LATD4	LATD3	LATD2	LATD1	LATD0	72
TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	71
PORTE	RE7	RE6	RE5	RE4	RE3	RE2	RE1	RE0	72
LATE	LATE7	LATE6	LATE5	LATE4	LATE3	LATE2	LATE1	LATE0	72
TRISE	TRISE7	TRISE6	TRISE5	TRISE4	TRISE3	TRISE2	TRISE1	TRISE0	71
PSPCON	IBF	OBF	IBOV	PSPMODE	-	-	-	-	71
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	69
PIR1	PSPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	71
PIE1	PSPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	71
IPR1	PSPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	71

凡例: - = 未実装ビット、「0」として読み出されます。網掛けの部分はパラレルスレーブポートでは使いません。

## 12.0 Timer0 モジュール

Timer0 モジュールには以下の機能があります。

- 8/16 ビットモードのタイマまたはカウンタ動作をソフトウェアによって選択可能
- 読み書き可能なレジスタ
- 専用の 8 ビット ソフトウェア プログラマブル プリスケーラ
- 選択可能なクロック源 (内部または外部)
- 外部クロックのエッジ選択
- オーバーフロー割り込み

T0CON レジスタ ( [レジスタ 12-1](#) ) は、プリスケール値の選択を含むモジュールのあらゆる機能を制御します。このレジスタは読み書きの両方が可能です。

[図 12-1](#) に、Timer0 モジュールの 8 ビットモードの概略ブロック図を示します。[図 12-2](#) に、Timer0 モジュールの 16 ビットモードの概略ブロック図を示します。

**レジスタ 12-1: T0CON: Timer0 制御レジスタ**

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

- bit 7      **TMR0ON:** Timer0 ON/OFF 制御ビット  
 1 = Timer0 を ON にする  
 0 = Timer0 を OFF にする
- bit 6      **T08BIT:** Timer0 8 ビット /16 ビット制御ビット  
 1 = Timer0 を 8 ビットのタイマ / カウンタとして設定する  
 0 = Timer0 を 16 ビットのタイマ / カウンタとして設定する
- bit 5      **T0CS:** Timer0 クロック源選択ビット  
 1 = T0CKI ピンの遷移  
 0 = 内部命令サイクルクロック (CLKO)
- bit 4      **T0SE:** Timer0 信号源エッジ選択ビット  
 1 = T0CKI ピンの High から Low への遷移時にインクリメントする  
 0 = T0CKI ピンの Low から High への遷移時にインクリメントする
- bit 3      **PSA:** Timer0 プリスケーラ割り当てビット  
 1 = Timer0 プリスケーラを割り当てず、Timer0 クロック入力はプリスケーラをバイパスする  
 0 = Timer0 プリスケーラを割り当て、Timer0 クロック入力をプリスケーラ出力から供給する
- bit 2-0      **T0PS<2:0>:** Timer0 プリスケーラ選択ビット  
 111 = 1:256 のプリスケール値を選択する  
 110 = 1:128 のプリスケール値を選択する  
 101 = 1:64 のプリスケール値を選択する  
 100 = 1:32 のプリスケール値を選択する  
 011 = 1:16 のプリスケール値を選択する  
 010 = 1:8 のプリスケール値を選択する  
 001 = 1:4 のプリスケール値を選択する  
 000 = 1:2 のプリスケール値を選択する

# PIC18F97J60 ファミリ

## 12.1 Timer0 の動作

Timer0 はタイマまたはカウンタのいずれかのモードによる動作が可能です。モードは T0CS ビット (T0CON<5>) によって選択します。タイマモード (T0CS = 0) の既定値設定では、異なるプリスケール値を選択していない限り、モジュールがクロックごとにインクリメントします ( [セクション 12.3「プリスケール」](#) 参照 )。TMR0 レジスタに書き込みが実行されると、その後の 2 命令サイクル間はインクリメントが抑止されます。TMR0 レジスタにあらかじめ調整した値を書き込む事によって、これを回避できます。

カウンタモードは T0CS ビットをセット (= 1) する事で選択します。このモードでは、RA4/T0CKI ピンの立ち上がりまたは立ち下がりエッジで Timer0 がインクリメントします。インクリメントするエッジは Timer0 信号源エッジ選択ビット T0SE (T0CON<4>) によって決まります。このビットをクリアすると立ち上がりエッジが選択されます。ここでは外部クロック入力を使う場合の制約について説明します。

Timer0 の駆動には外部クロック源が使えます。ただし、外部クロックが内部の位相クロック (Tosc) と確実に同期できるように、一定の要件を満たす必要があります。

まず、同期確立からタイマ / カウンタのインクリメント開始までには遅延があります。

## 12.2 16 ビットモードにおける Timer0 の読み書き

16 ビットモード動作時の TMR0H は Timer0 の実際の上位バイトではなく、バッファを介した値です。実際の上位バイトは、直接読み書きできません ( [図 12-2](#) 参照 )。TMR0H は TMR0L の読み出し時に Timer0 の上位バイトの内容によって更新されます。この方法により、上位と下位バイトの読み出しが有効であったかどうか検証する事なく、Timer0 の全 16 ビットを一括して読み出す事ができます。上位バイトと下位バイトを連続して読み出した際のロールオーバーを懸念する必要がないためです。

読み出しと同様に Timer0 の上位バイトへの書き込みも、TMR0H のバッファレジスタを介して実行する必要があります。上位バイトは、TMR0L への書き込みを実行した時点で、TMR0H の内容によって更新されます。これによって、Timer0 の全 16 ビットを一度に更新できます。

図 12-1: Timer0 のブロック図 (8 ビットモード)

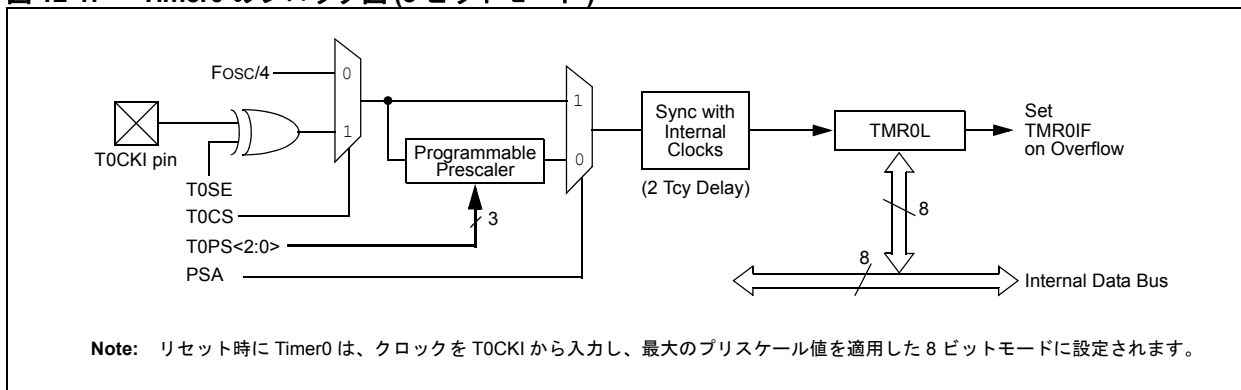
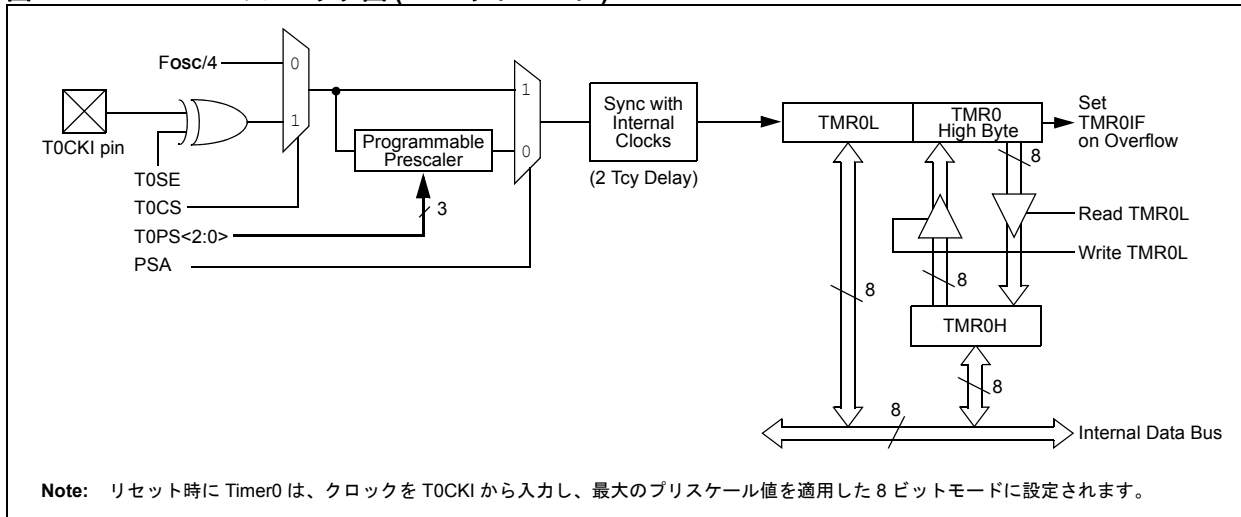


図 12-2: Timer0 のブロック図 (16 ビットモード)



## 12.3 プリスケーラ

Timer0モジュールのプリスケーラとして8ビットのカウンタが使えます。プリスケーラは直接読み書きできません。値は PSA ビットと T0PS<2:0> ビット (T0CON<3:0>) によって設定します。これらの値は、プリスケーラの割り当てと比率を決定します。

PSA ビットをクリアすると、Timer0 モジュールにプリスケーラが割り当てられます。その場合、1:2 ~ 1:256 の範囲の2のべき乗でプリスケール値を選択できます。

Timer0 モジュールにプリスケーラを割り当てた場合、TMR0 レジスタに書き込む全ての命令 (例: CLRF TMR0、MOVWF TMR0、BSF TMR0) は、プリスケーラのカウントをクリアします。

**Note:** プリスケーラをTimer0に割り当てている場合、TMR0へ書き込むとプリスケーラのカウントはクリアされますが、プリスケーラの割り当て自体は変わりません。

### 12.3.1 プリスケーラ割り当ての切り換え

プリスケーラの割り当てはソフトウェアによって完全に制御でき、プログラム実行中に変更できます。

## 12.4 Timer0 割り込み

TMR0 レジスタが、FFh から 00h (8 ビットモード) または FFFFh から 0000h (16 ビットモード) にオーバーフローすると TMR0 割り込みが発生します。このオーバーフローは TMR0IF フラグビットをセットします。TMR0IE ビット (INTCON<5>) をクリアすると、この割り込みをマスクできます。この割り込みを再度有効にする前に、ソフトウェアの割り込みサービスルーチン内で TMR0IF ビットをクリアしておく必要があります。

Timer0 はスリープでシャットダウンされるため、TMR0 の割り込みによってプロセッサをスリープから復帰させる事はできません。

表 12-1: Timer0 に関連するレジスタ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	リセット値の記載ページ
TMR0L	Timer0 レジスタ下位バイト								70
TMR0H	Timer0 レジスタ上位バイト								70
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	69
INTCON2	RBPU	INTEDG0	INTEDG1	INTEDG2	INTEDG3	TMR0IP	INT3IP	RBIP	69
T0CON	TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0	70
TRISA	-	-	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	71

凡例: - = 未実装ビット、「0」として読み出されます。網掛けの部分はTimer0では使いません。

# PIC18F97J60 ファミリ

---

NOTES:

## 13.0 Timer1 モジュール

Timer1 タイマ/カウンタ モジュールには以下の機能が  
あります。

- 16 ビットのタイマまたはカウンタ動作をソフト  
ウェアによって選択可能
- 読み書き可能な 8 ビットのレジスタ (TMR1H と  
TMR1L)
- 内部または外部のクロック源と、デバースクロック  
または Timer1 オシレータの内部オプションを選択  
可能
- オーバーフロー割り込み
- ECCP 特殊イベントトリガによるリセット
- デバースクロックのステータスフラグ (T1RUN)

図 13-1 に、Timer1 モジュールの概略ブロック図を示  
します。図 13-2 に、モジュールの読み書きモードの

動作を説明するブロック図を示します。

このモジュールは低消費電力オシレータを備えてお  
り、クロック駆動の選択肢を追加します。Timer1 オシ  
レータは、電力管理動作時にマイクロコントローラに  
クロックを供給する、低電力のクロック源としても使  
えます。

また、Timer1 を使うと外付け部品とコード オーバ  
ヘッドをほとんど増やす事なく、アプリケーションに  
リアルタイムクロック (RTC) 機能を持たせる事ができ  
ます。

Timer1 は T1CON 制御レジスタ (レジスタ 13-1) によ  
って制御します。このレジスタには、Timer1 オシレー  
タイネーブルビット (T1OSCN) も含まれます。Timer1  
は、制御ビット TMR1ON (T1CON<0>) をセット/クリ  
アする事で有効/無効に設定します。

レジスタ 13-1: T1CON: Timer1 制御レジスタ

R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCN	T1SYNC	TMR1CS	TMR1ON
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

- bit 7      **RD16:** 16 ビット読み書きモードイネーブルビット  
1 = 1 回の 16 ビット動作による Timer1 のレジスタ読み書きを有効に設定する  
0 = 2 回の 8 ビット動作による Timer1 のレジスタ読み書きを有効に設定する
- bit 6      **T1RUN:** Timer1 システムクロック ステータスビット  
1 = デバースクロックが Timer1 オシレータから供給されている  
0 = デバースクロックが他の信号源から供給されている
- bit 5-4      **T1CKPS<1:0>:** Timer1 入力クロック プリスケール値選択ビット  
11 = 1:8 のプリスケール値を選択する  
10 = 1:4 のプリスケール値を選択する  
01 = 1:2 のプリスケール値を選択する  
00 = 1:1 のプリスケール値を選択する
- bit 3      **T1OSCN:** Timer1 オシレータ イネーブルビット  
1 = Timer1 オシレータを有効にする  
0 = Timer1 オシレータを無効にする  
オシレータのインバータとフィードバック抵抗を遮断して消費電力を低減します。
- bit 2      **T1SYNC:** Timer1 外部クロック入力同期選択ビット  
**TMR1CS = 1 の場合:**  
1 = 外部クロック入力を同期させない  
0 = 外部クロック入力を同期させる  
**TMR1CS = 0 の場合:**  
このビットを無視する。TMR1CS = 0 の場合、Timer1 は内部クロックを使う
- bit 1      **TMR1CS:** Timer1 クロック源選択ビット  
1 = RC0/T1OSO/T13CKI ピンに入力される外部クロック (立ち上がりエッジ)  
0 = 内部クロック (Fosc/4)
- bit 0      **TMR1ON:** Timer1 ON/OFF 制御ビット  
1 = Timer1 を ON にする  
0 = Timer1 を OFF にする

# PIC18F97J60 ファミリ

## 13.1 Timer1 の動作

Timer1 は以下のいずれかのモードで動作します。

- タイマ
- 同期カウンタ
- 非同期カウンタ

動作モードはクロック選択ビット TMR1CS (T1CON<1>) によって決まります。TMR1CS をクリア (= 0) すると、Timer1 は内部命令サイクル (Fosc/4) ごとにインクリ

メントします。TMR1CS をセットすると、Timer1 は Timer1 外部クロック入力または Timer1 オシレータ (有効化している場合) の立ち上がりエッジごとにインクリメントします。

Timer1 有効時、RC1/T1OSI と RC0/T1OSO/T13CKI ピンは入力になります。これは TRISC<1:0> の値が無視され、ピンが「0」として読み出される事を意味します。

図 13-1: Timer1 のブロック図

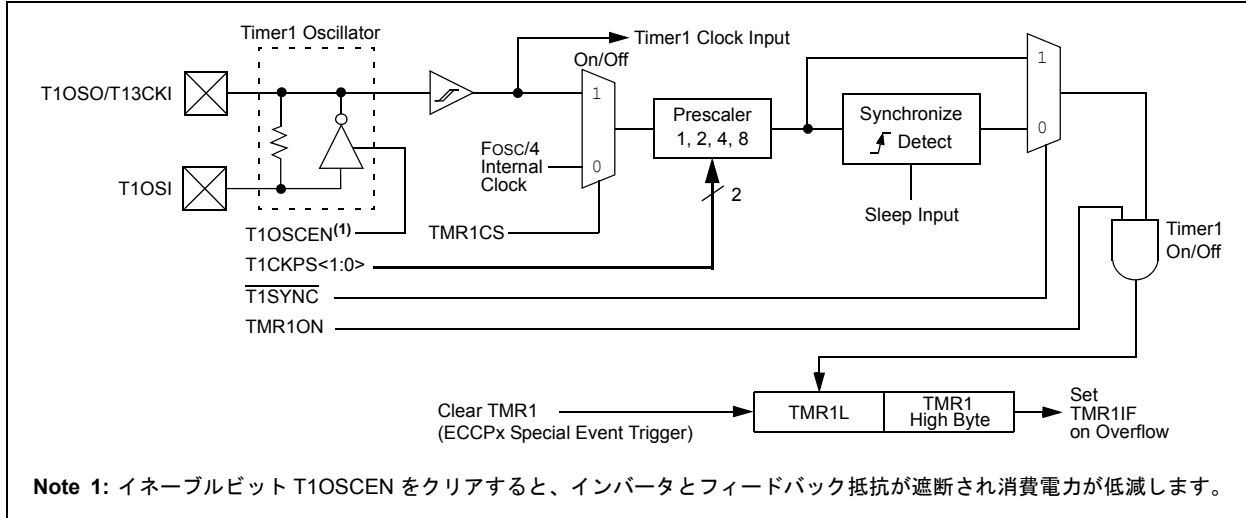
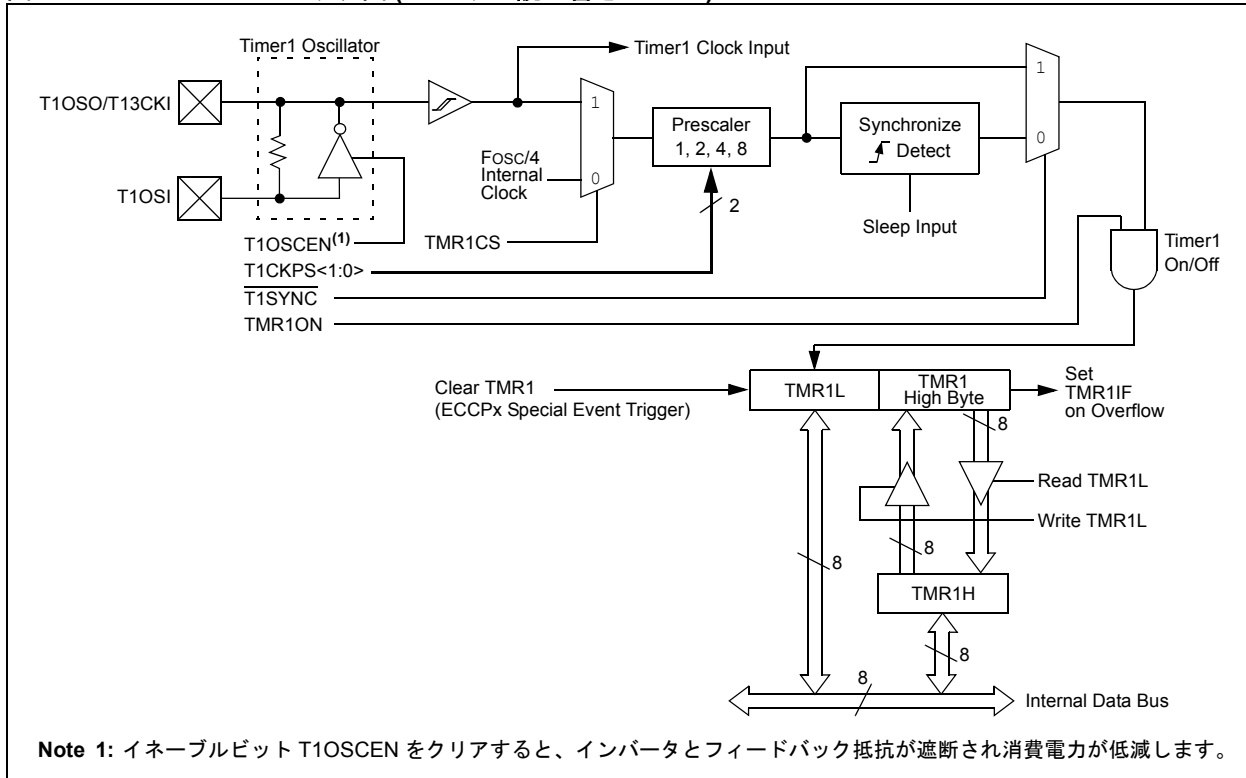


図 13-2: Timer1 のブロック図 (16 ビット読み書きモード)





## 13.2 Timer1 の 16 ビット読み書きモード

Timer1 は、16 ビットの読み書きが可能な設定 (図 13-2 参照) にする事ができます。RD16 制御ビット (T1CON<7>) をセットすると、TMR1H のアドレスが Timer1 上位バイトのバッファレジスタにマッピングされます。TMR1L を読み出す事で、Timer1 の上位バイトの内容が Timer1 上位バイト バッファレジスタに読み込まれます。この動作により、Timer1 の全 16 ビットを正確に読み出す事ができます。下位バイトの前に読み出された上位バイトが、2 回の読み出しの間のロールオーバーによって無効になっていないかを検証する必要があります。

読み出しと同様に、Timer1 の上位バイトへの書き込みも TMR1H のバッファレジスタを介して実行する必要があります。Timer1 の上位バイトは、TMR1L への書き込みを実行した時点で TMR1H の内容によって更新されます。これによって、ユーザは Timer1 の上位バイトと下位バイト両方の全 16 ビットを一括して書き込む事ができます。

このモードでは Timer1 の上位バイトを直接読み書きできません。読み書きは全て Timer1 上位バイト バッファレジスタを介して実行する必要があります。TMR1H に書き込んでも、Timer1 のプリスケアラはクリアされません。プリスケアラがクリアされるのは、TMR1L に書き込んだ場合だけです。

## 13.3 Timer1 オシレータ

T1OSI (入力) と T1OSO (アンプ出力) ピンの間には、内蔵の水晶振動子オシレータ回路が接続されています。このオシレータは、Timer1 オシレータ イネーブルビット T1OSCEN (T1CON<3>) をセットすると有効になります。定格 32 kHz の水晶振動子によって低消費電力で動作します。電力管理モードのいずれに移行しても、このオシレータは動作を続けます。図 13-3 に代表的な LP オシレータの回路を示します。表 13-1 に、Timer1 オシレータ用のコンデンサを選定する際の指針を示します。

Timer1 オシレータが確実に起動するように、ソフトウェアによる遅延を確保する必要があります。

図 13-3: Timer1 オシレータに必要な外付け部品

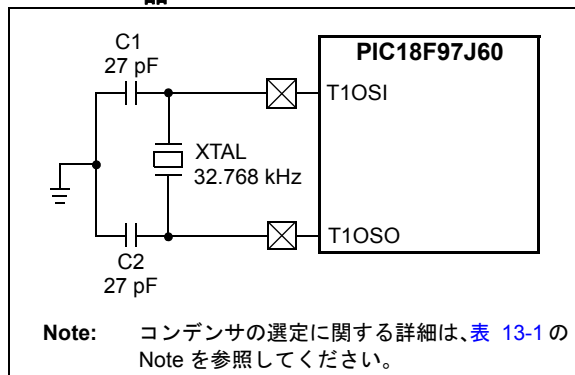


表 13-1: Timer1 オシレータ用コンデンサの選択 (2、3、4)

オシレータタイプ	周波数	C1	C2
LP	32 kHz	27 pF <sup>(1)</sup>	27 pF <sup>(1)</sup>

**Note 1:** マイクロチップ社は、オシレータ回路の検証に際して上記の値を初期値として推奨します。

**2:** 静電容量を大きくするとオシレータの安定性は増すものの、起動に要する時間が長くなります。

**3:** 振動子 / 水晶振動子は製品ごとに固有の特性を持つため、外付け部品の適切な数値は振動子 / 水晶振動子のメーカーに問い合わせてください。

**4:** ここに示すコンデンサの値は設計上の目安に過ぎません。

### 13.3.1 クロック源としての Timer1 の使用法

Timer1 オシレータは、電力管理モードのクロック源としても使えます。クロック選択ビット SCS<1:0> (OSCCON<1:0>) を「01」に設定すると、デバイスは SEC\_RUN モードに切り換わります。Timer1 オシレータから CPU と周辺モジュールの両方のクロックを供給します。IDLEN ビット (OSCCON<7>) をクリアして SLEEP 命令を実行すると、デバイスは SEC\_IDLE モードに移行します。詳細は、[セクション 4.0「電力管理モード」](#)を参照してください。

Timer1 オシレータをクロック源として使う場合は常に、Timer1 システムクロック ステータスフラグ T1RUN (T1CON<6>) がセットされます。従って、このフラグによってコントローラの現在のクロック供給モードを判断できます。このビットからは、フェイルセーフ クロックモニタが現在使っているクロック源も分かります。クロックモニタが有効で、クロック供給中の Timer1 オシレータに障害が発生した場合、T1RUN ビットをポーリングすれば、クロックの供給源が Timer1 オシレータか他のクロック源かを判断できます。

### 13.3.2 Timer1 オシレータのレイアウト上の考慮事項

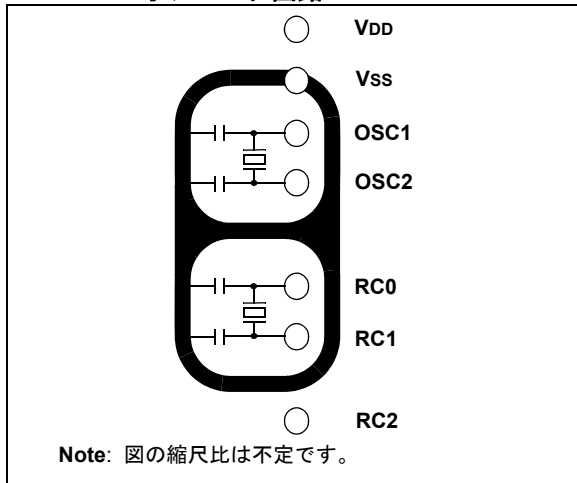
Timer1 オシレータ回路は、動作中にごくわずかな電力しか消費しません。低消費電力であるという事は、高速で変化する近くの信号から影響を受けやすい事を意味します。

このため、図 13-3 に示したオシレータ回路は、マイクログリッドコントローラのできるだけ近くに配置してください。また、オシレータ回路のレイアウト領域を横切る配線は Vss または VDD のみとしてください。

# PIC18F97J60 ファミリ

オシレータのそばに高速の回路を配置せざるを得ない場合は(出力コンペアまたはPWMモードのECCP1ピン、OSC2ピンを使うプライマリオシレータ等)、[図 13-4](#)に示すようにオシレータ回路の周りに接地されたガードリングを設けると効果的な場合があります。この方法は、片面のみのPCBまたはグラウンドプレーンで使う場合に適用できます。

図 13-4: 接地されたガードリングで囲まれたオシレータ回路



## 13.4 Timer1 割り込み

TMR1 レジスタペア (TMR1H:TMR1L) は 0000h から FFFFh までインクリメントした後、0000h にロールオーバーします。Timer1 割り込みを有効にしている場合、オーバーフローが発生した時点で割り込みが発生し、割り込みフラグビット TMR1IF (PIR1<0>) にラッチされます。この割り込みは、Timer1 割り込みイネーブルビット TMR1IE (PIE1<0>) をセットまたはクリアする事で、有効または無効に設定できます。

## 13.5 ECCPx 特殊イベントトリガによる Timer1 のリセット

ECCP1 または ECCP2 が Timer1 を使い、コンペアモードで特殊イベントトリガを生成するように設定されている場合 (CCPxM<3:0> = 1011)、このトリガ信号によって Timer3 がリセットされます。A/D モジュールを有効にしている場合、ECCP2 によるトリガは A/D 変換も起動します (詳細は[セクション 18.2.1「特殊イベントトリガ」](#)を参照してください)。

この機能を使うには、モジュールをタイマまたは同期カウンタのいずれかに設定する必要があります。この方法で使う場合、CCPRxH:CCPRxL レジスタペアは実質的に Timer1 の周期レジスタとして機能します。

Timer1 が非同期カウンタモードで動作している場合、特殊イベントトリガによるリセットが動作しない可能性があります。

Timer1 への書き込みと特殊イベントトリガが同時に発生した場合、書き込み動作が優先されます。

**Note:** ECCPxモジュールによる特殊イベントトリガは、TMR1IF 割り込みフラグビット (PIR1<0>) をセットしません。

## 13.6 リアルタイムクロックとしての Timer1 の使用法

Timer1 に外部 LP オシレータを接続すると (例えば[セクション 13.3「Timer1 オシレータ」](#)で示した回路)、アプリケーションに RTC 機能を実装する事ができます。正確なタイムベースを供給する腕時計用の安価な水晶振動子と、時間を計算する数行のアプリケーションコードがあれば実現できます。スリープで動作し、電源として電池またはスーパーキャパシタを使う場合、外付けの RTC デバイスとバッテリーバックアップは不要です。

**例 13-1** に示すアプリケーションコードルーチン RTcIsr は、割り込みサービ斯拉ーチンによって 1 秒間隔でカウンタをインクリメントする簡単な方法を示したものです。TMR1 レジスタペアがインクリメントによってオーバーフローすると割り込みがトリガされ、このルーチンが呼び出されます。これによって、秒カウンタが 1 つインクリメントされます。その他の「分」と「時間」のカウンタは、前段のカウンタがオーバーフローするごとにインクリメントされます。

レジスタペアは 16 ビット幅である事から、レジスタを 32.768 kHz のクロックで直接駆動した場合、カウンタアップしてオーバーフローさせるには 2 秒かかります。1 秒間隔でオーバーフローさせるには、あらかじめレジスタに値を読み込んでおく必要があります。最も簡単な方法は、BSF 命令によって TMR1H の MSb をセットする事です。TMR1L レジスタには、あらかじめ値を格納したり変更したりできない事に注意してください。このような操作を実行すると、サイクル数が増えると共に誤差が累積する場合があります。

この方法で正確な RTC を実現するには、RTcInit ルーチンにも示した通り、Timer1 を非同期モードで動作させ、Timer1 オーバーフロー割り込みを有効に (PIE1<0> = 1) しておく必要があります。また、Timer1 オシレータを有効に設定し、常時動作させておく必要もあります。

## 13.7 非同期カウンタモードに関する考慮事項

Timer1 割り込みと TMR1 レジスタの更新に続き、Timer1 モジュールはクロック源の立ち下がりエッジによって、立ち上がりエッジにおける次のレジスタ更新をトリガします。従って、更新がクロック入力の立ち下がりよりも後に完了した場合、次の立ち上がりエッジはカウントされません。

タイマ入力立ち下がる前に、アプリケーションが確実に TMR1 を更新できる場合、特別な措置は必要ありません。確実にとは思われない場合、後の Timer1 インクリメントに続いて更新されるように調整できます。それには、割り込みルーチン内で TMR1L のインクリメントを監視し、インクリメントされたらクロックが Low の間またはクロック源の周期の半分の期間に

# PIC18F97J60 ファミリ

TMR1H:TMR1L レジスタペアを更新します。Timer1 をリアルタイムクロックとして使う場合、クロック源は 32.768 kHz の水晶振動子です。その場合のクロックの半周期は 15.25  $\mu$ s です。

例 13-1 のリアルタイムクロックアプリケーションコードには、Timer1 の代表的な ISR と、TMR1 を必要な間隔で確実に更新する事が困難な場合に必要となる追加のコードを示しました。

## 例 13-1: Implementing a Real-Time Clock Using a Timer1 Interrupt Service

```

RTCinit
    MOVLW    80h          ; Preload TMR1 register pair
    MOVWF   TMR1H        ; for 1 second overflow
    CLRF    TMR1L
    MOVLW   b'00001111' ; Configure for external clock,
    MOVWF   T1CON        ; Asynchronous operation, external oscillator
    CLRF    secs         ; Initialize timekeeping registers
    CLRF    mins         ;
    MOVLW   .12
    MOVWF   hours
    BSF     PIE1, TMR1IE; Enable Timer1 interrupt
    RETURN

RTCisr
    ; Insert the next 4 lines of code when TMR1
    ; can not be reliably updated before clock pulse goes low
    BTFSC   TMR1L,0      ; wait for TMR1L to become clear
    BRA     $-2          ; (may already be clear)
    BTFSS   TMR1L,0      ; wait for TMR1L to become set
    BRA     $-2          ; TMR1 has just incremented
    ; If TMR1 update can be completed before clock pulse goes low
    ; Start ISR here
    BSF     TMR1H, 7     ; Preload for 1 sec overflow
    BCF     PIR1, TMR1IF; Clear interrupt flag
    INCF    secs, F      ; Increment seconds
    MOVLW   .59         ; 60 seconds elapsed?
    CPFSGT  secs
    RETURN          ; No, done
    CLRF    secs        ; Clear seconds
    INCF    mins, F     ; Increment minutes
    MOVLW   .59         ; 60 minutes elapsed?
    CPFSGT  mins
    RETURN          ; No, done
    CLRF    mins        ; clear minutes
    INCF    hours, F    ; Increment hours
    MOVLW   .23         ; 24 hours elapsed?
    CPFSGT  hours
    RETURN          ; No, done
    CLRF    hours       ; Reset hours
    RETURN          ; Done
    
```

表 13-2: Timer1 のタイマ / カウンタ動作に関連するレジスタ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	リセット値の記載ページ
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	69
PIR1	PSPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	71
PIE1	PSPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	71
IPR1	PSPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	71
TMR1L	Timer1 レジスタ下位バイト								70
TMR1H	Timer1 レジスタ上位バイト								70
T1CON	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	70

凡例: 網掛けの部分は Timer1 モジュールでは使いません。

# PIC18F97J60 ファミリ

## 14.0 Timer2 モジュール

Timer2 タイマモジュールには以下の機能があります。

- 8 ビットのタイマと周期レジスタ (それぞれ TMR2 と PR2)
- 読み書き可能 (両レジスタ)
- ソフトウェア プログラマブル プリスケアラ (1:1、1:4、1:16)
- ソフトウェア プログラマブル ポストスケアラ (1:1 ~ 1:16)
- TMR2/PR2 一致割り込み
- MSSPx モジュールのシフトクロックとしての使用

このモジュールは、タイマの有効/無効、プリスケアラとポストスケアラを設定する T2CON レジスタ (レジスタ 14-1) によって制御します。制御ビット TMR2ON (T2CON<2>) をクリアして Timer2 を遮断する事で消費電力を低減できます。

図 14-1 に、このモジュールの概略ブロック図を示します。

## 14.1 Timer2 の動作

通常動作時、TMR2 は 00h からクロックごと ( $F_{osc}/4$ ) にインクリメントされます。クロック入力に適用する 4 ビットのカウンタ/プリスケアラでは、直接入力、4 分周、16 分周のプリスケール値が使えます。これらの設定は、プリスケアラ制御ビット T2CKPS<1:0> (T2CON<1:0>) で選択します。クロックサイクルごとに、TMR2 の値と周期レジスタ PR2 の値が比較されます。2 つの値が一致すると、コンパレータはタイマ出力として一致信号を生成します。この信号は、次のサイクルで TMR2 の値を 00h にリセットし、出力カウンタ/ポストスケアラを駆動します (セクション 14.2 「Timer2 割り込み」参照)。

TMR2 と PR2 レジスタは、いずれも直接読み書きできます。TMR2 レジスタは全てのデバイスリセットによってクリアされますが、PR2 レジスタは FFh に初期化されます。プリスケアラ カウンタとポストスケアラ カウンタは、以下のイベントでクリアされます。

- TMR2 レジスタへの書き込み
- T2CON レジスタへの書き込み
- 何らかのデバイスリセット (パワーオン リセット、MCLR リセット、ウォッチドッグ タイマ リセット、ブラウンアウト リセット)

T2CON レジスタに書き込んでも TMR2 はクリアされません。

レジスタ 14-1: T2CON: Timer2 制御レジスタ

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

- bit 7      **未実装:** 「0」として読み出し
- bit 6-3      **T2OUTPS<3:0>:** Timer2 出力ポストスケール値選択ビット  
0000 = 1:1 のポストスケール値を選択する  
0001 = 1:2 のポストスケール値を選択する  
.  
.  
.  
1111 = 1:16 のポストスケール値を選択する
- bit 2      **TMR2ON:** Timer2 ON/OFF 制御ビット  
1 = Timer2 を ON にする  
0 = Timer2 を OFF にする
- bit 1-0      **T2CKPS<1:0>:** Timer2 クロック プリスケール値選択ビット  
00 = 1:1 のプリスケール値を選択する  
01 = 1:4 のプリスケール値を選択する  
1x = 1:16 のプリスケール値を選択する

## 14.2 Timer2 割り込み

Timer2 は、デバイス割り込みを生成する事もできます。Timer2 の出力信号 (TMR2/PR2 一致) は、4 ビットの出力カウンタ / ポストスケラに入力されます。このカウンタが、TMR2IF (PIR1<1>) にラッチされる TMR2 一致割り込みフラグを生成します。割り込みは、TMR2 一致割り込みイネーブルビット TMR2IE (PIE1<1>) をセットする事で有効にします。

ポストスケラ制御ビット T2OUTPS<3:0> (T2CON<6:3>) によって 16 通りのポストスケール 値 (1:1 ~ 1:16) を選択できます。

## 14.3 Timer2 の出力

TMR2 の分周していない出力は、主に CCP モジュールで PWM モード動作のタイムベースとして使われます。

Timer2 は、SPI モードで動作する MSSPx モジュールのシフトクロック源として使う事もできます。詳細は [セクション 20.0「マスタ同期シリアルポート \(MSSP\) モジュール」](#) を参照してください。

図 14-1: Timer2 のブロック図

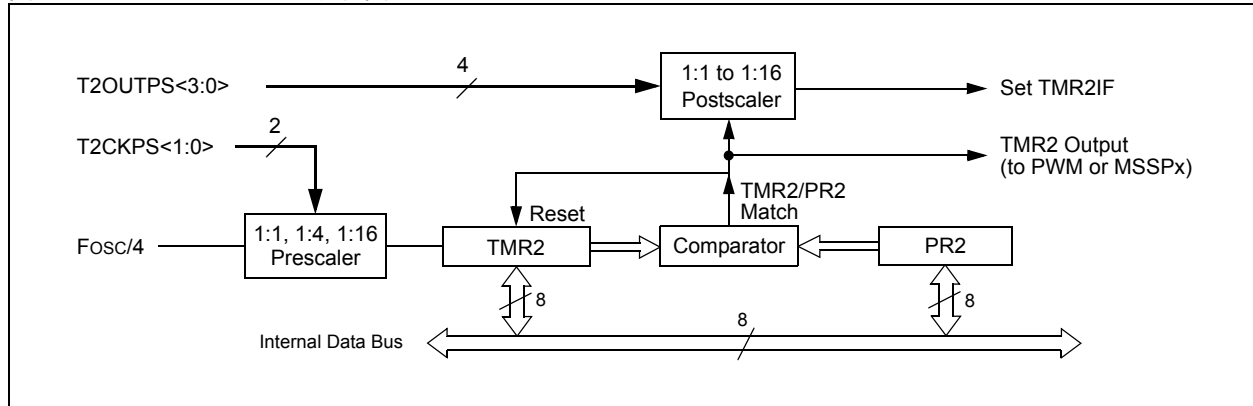


表 14-1: Timer2 のタイマ / カウンタ動作に関連するレジスタ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	リセット値の記載ページ
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	69
PIR1	PSPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	71
PIE1	PSPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	71
IPR1	PSPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	71
TMR2	Timer2 レジスタ								70
T2CON	-	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	70
PR2	Timer2 周期レジスタ								70

凡例: - = 未実装ビット、「0」として読み出されます。網掛けの部分は Timer2 モジュールでは使いません。

# PIC18F97J60 ファミリ

---

NOTES:

## 15.0 Timer3 モジュール

Timer3 タイマ/カウンタ モジュールには以下の機能が  
あります。

- 16 ビットのタイマまたはカウンタ動作をソフト  
ウェアによって選択可能
- 読み書き可能な 8 ビットのレジスタ (TMR3H と  
TMR3L)
- 内部または外部のクロック源と、デバイスクロック  
または Timer1 オシレータの内部オプションを選択  
可能
- オーバーフロー割り込み
- CCPx/ECCPx 特殊イベントトリガによるモジュール  
リセット

図 15-1 に、Timer3 モジュールの概略ブロック図を示  
します。図 15-2 に、モジュールの読み書きモードの  
動作を説明するブロック図を示します。

Timer3モジュールはT3CONレジスタ(レジスタ 15-1)  
によって制御します。このレジスタはCCPxとECCPx  
モジュールのクロック源を選択する時にも使います  
(詳細は、[セクション 17.1.1「CCPx/ECCPx モジュー  
ルとタイマ リソース」](#)を参照してください)。

レジスタ 15-1: T3CON: Timer3 制御レジスタ

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	$\overline{T3SYNC}$	TMR3CS	TMR3ON
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

- bit 7      **RD16:** 16 ビット読み書きモード イネーブルビット  
1 = 1 回の 16 ビット動作による Timer3 のレジスタ読み書きを有効に設定する  
0 = 2 回の 8 ビット動作による Timer3 のレジスタ読み書きを有効に設定する
- bit 6,3      **T3CCP<2:1>:** CCPx/ECCPx に対する Timer3/Timer1 イネーブルビット  
11 = Timer3 と Timer4 を全ての CCPx/ECCPx モジュールのクロック源とする  
10 = Timer3 と Timer4 を ECCP3、CCP4、CCP5 のクロック源とし、  
Timer1 と Timer2 を ECCP1、ECCP2 のクロック源とする  
01 = Timer3 と Timer4 を ECCP2、ECCP3、CCP4、CCP5 のクロック源とし、  
Timer1 と Timer2 を ECCP1 のクロック源とする  
00 = Timer1 と Timer2 を全ての CCPx/ECCPx モジュールのクロック源とする
- bit 5-4      **T3CKPS<1:0>:** Timer3 入力クロック プリスケール値選択ビット  
11 = 1:8 のプリスケール値を選択する  
10 = 1:4 のプリスケール値を選択する  
01 = 1:2 のプリスケール値を選択する  
00 = 1:1 のプリスケール値を選択する
- bit 2      **T3SYNC:** Timer3 外部クロック入力同期選択ビット  
(デバイスクロックを Timer1/Timer3 から供給している場合は使用不可)  
**TMR3CS = 1 の場合:**  
1 = 外部クロック入力を同期させない  
0 = 外部クロック入力を同期させる  
**TMR3CS = 0 の場合:**  
このビットを無視する。TMR3CS = 0 の場合、Timer3 は内部クロックを使う
- bit 1      **TMR3CS:** Timer3 クロック源選択ビット  
1 = Timer1 オシレータまたは T13CKI からの外部クロック入力 (最初の立ち上がりエッジ後の、  
立ち上がりエッジごと)  
0 = 内部クロック (Fosc/4)
- bit 0      **TMR3ON:** Timer3 ON/OFF 制御ビット  
1 = Timer3 を ON にする  
0 = Timer3 を OFF にする

# PIC18F97J60 ファミリ

## 15.1 Timer3 の動作

Timer3 は以下のいずれかのモードで動作します。

- タイマ
- 同期カウンタ
- 非同期カウンタ

動作モードはクロック選択ビット TMR3CS (T3CON<1>) によって決まります。TMR3CS をクリア (= 0) すると、Timer3 は内部命令サイクル (Fosc/4) ごとにインクリメントします。TMR3CS をセットすると、Timer3 は、Timer1 外部クロック入力または Timer1 オシレータ (有効化している場合) のいずれかの立ち上がりエッジごとにインクリメントします。

Timer1 と同様に、Timer1 オシレータを有効にした場合、RC1/T1OSI と RC0/T1OSO/T13CKI ピンは入力になります。これは TRISC<1:0> の値が無視され、ピンが「0」として読み出される事を意味します。

図 15-1: Timer3 のブロック図

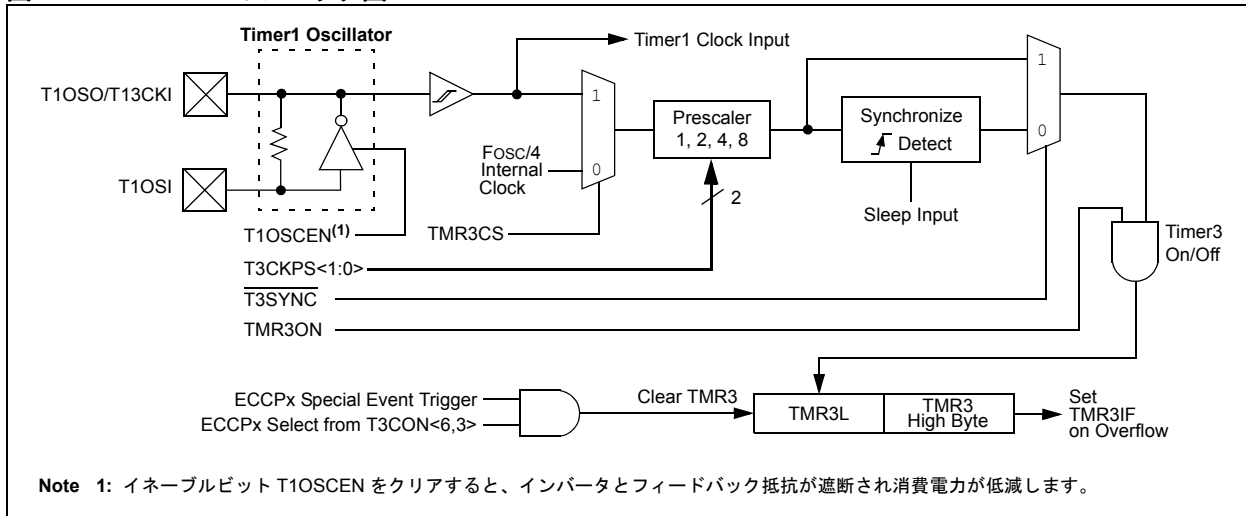
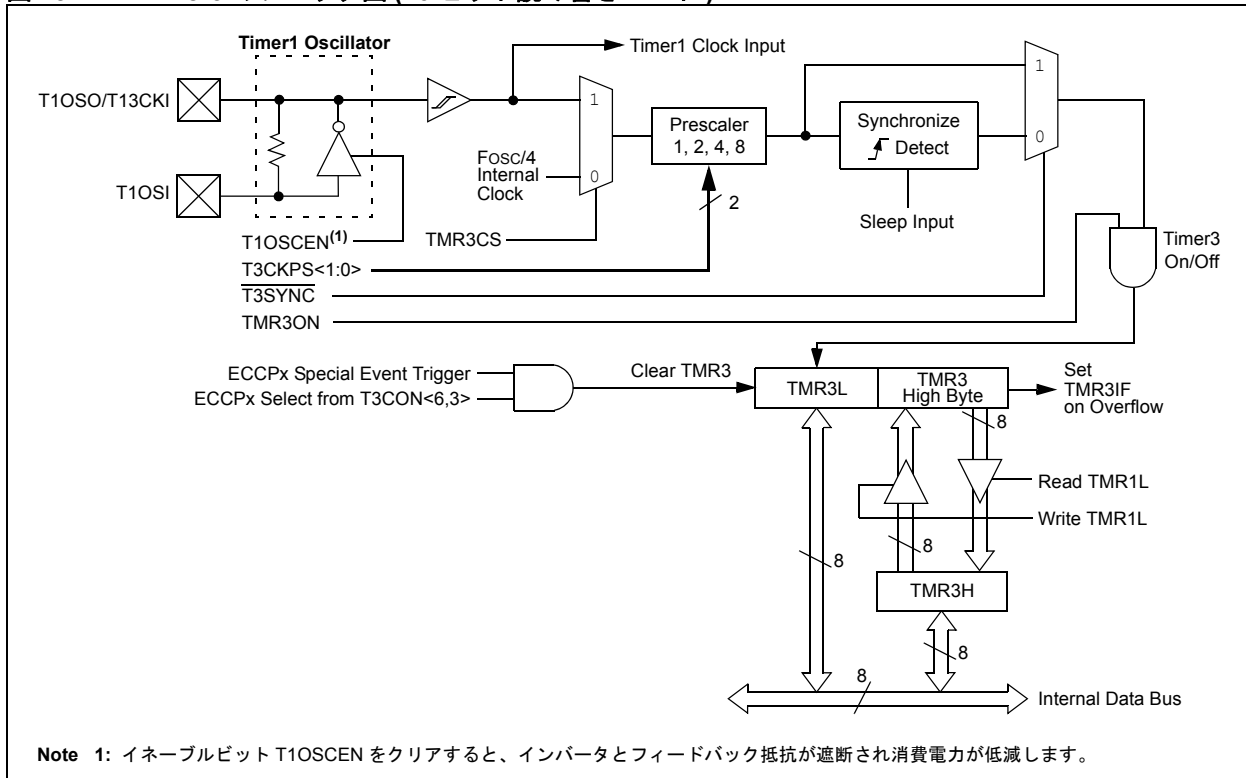


図 15-2: Timer3 のブロック図 (16 ビット読み書きモード)





## 15.2 Timer3 (16 ビット読み書きモード)

Timer3は、16ビットの読み書きが可能な設定(図 15-2 参照) にする事ができます。RD16 制御ビット (T3CON<7>) をセットすると、TMR3H のアドレスが Timer3 上位バイトのバッファレジスタにマッピングされます。TMR3L を読み出す事で、Timer3 の上位バイトの内容が Timer3 上位バイト バッファレジスタに格納されます。この動作により、Timer3 の全 16 ビットを正確に読み出す事ができます。下位バイトの前に読み出された上位バイトが、2 回の読み出しの間のロールオーバーによって無効になっていないかを検証する必要があります。

読み出しと同様に、Timer3 の上位バイトへの書き込みも TMR3H のバッファレジスタを介して実行する必要があります。Timer3 の上位バイトは、TMR3L への書き込みを実行した時点で TMR3H の内容によって更新されます。これによって、ユーザは Timer3 の上位バイトと下位バイト両方の全 16 ビットを一括して書き込む事ができます。

このモードでは Timer3 の上位バイトを直接読み書きできません。読み書きは全て Timer3 上位バイト バッファレジスタを介して実行する必要があります。

TMR3H に書き込んでも、Timer3 のプリスケラはクリアされません。プリスケラがクリアされるのは、TMR3L に書き込んだ場合だけです。

## 15.3 Timer3 のクロック源としての Timer1 オシレータの使用法

Timer3 のクロック源として Timer1 内部オシレータが使えます。Timer1 オシレータは、T1OSCEN (T1CON<3>) ビットをセットする事で有効にします。Timer3 のクロック源として使うには、TMR3CS ビットもセットする必要があります。前述の通り、この設定によって Timer3 はオシレータ信号源の立ち上がりエッジごとにインクリメントします。

Timer1 オシレータについては、[セクション 13.0 「Timer1 モジュール」](#) で説明しています。

表 15-1: Timer3 のタイマ / カウンタ動作に関連するレジスタ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	リセット値の記載ページ
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	69
PIR2	OSCFIF	CMIF	ETHIF	r	BCL1IF	-	TMR3IF	CCP2IF	71
PIE2	OSCFIE	CMIE	ETHIE	r	BCL1IE	-	TMR3IE	CCP2IE	71
IPR2	OSCFIP	CMIP	ETHIP	r	BCL1IP	-	TMR3IP	CCP2IP	71
TMR3L	Timer3 レジスタ下位バイト								70
TMR3H	Timer3 レジスタ上位バイト								70
T1CON	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	T1SYN $\bar{C}$	TMR1CS	TMR1ON	70
T3CON	RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	T3SYN $\bar{C}$	TMR3CS	TMR3ON	71

凡例: - = 未実装ビット、「0」として読み出されます。r = 予約済みです。網掛けの部分は Timer3 モジュールでは使いません。

## 15.4 Timer3 割り込み

TMR3 レジスタペア (TMR3H:TMR3L) は 0000h から FFFFh までインクリメントした後、0000h にオーバーフローします。Timer3 割り込みを有効にしている場合、オーバーフローが発生した時点で割り込みが発生し、割り込みフラグビット TMR3IF (PIR2<1>) にラッチされます。この割り込みは、Timer3 割り込みイネーブルビット TMR3IE (PIE2<1>) をセットまたはクリアする事で、有効または無効に設定できます。

## 15.5 ECCPx 特殊イベントトリガによる Timer3 のリセット

ECCP1 または ECCP2 が Timer3 を使い、コンペアモードで特殊イベントトリガを生成するように設定されている場合 (CCPxM<3:0> = 1011)、このトリガ信号によって Timer3 がリセットされます。A/D モジュールを有効にしている場合、ECCP2 によるトリガは A/D 変換も起動します (詳細は[セクション 18.2.1 「特殊イベントトリガ」](#)を参照してください)。

この機能を使うには、モジュールをタイマまたは同期カウンタのいずれかに設定する必要があります。この方法で使う場合、CCPRxH:CCPRxL レジスタペアは実質的に Timer3 の周期レジスタとして機能します。

Timer3 が非同期カウンタモードで動作している場合、リセット動作が機能しない可能性があります。

Timer3 への書き込みと ECCPx モジュールからの特殊イベントトリガが同時に発生した場合、書き込みが優先されます。

**Note:** ECCPxモジュールによる特殊イベントトリガは、TMR3IF 割り込みフラグビット (PIR2<1>) をセットしません。

# PIC18F97J60 ファミリ

---

NOTES:

## 16.0 Timer4 モジュール

Timer4 モジュールには以下の機能があります。

- 8ビットのタイマレジスタ (TMR4)
- 8ビットの周期レジスタ (PR4)
- 読み書き可能 (両レジスタ)
- ソフトウェア プログラマブル プリスケアラ (1:1、1:4、1:16)
- ソフトウェア プログラマブル ポストスケアラ (1:1 ~ 1:16)
- TMR4/PR4 一致割り込み

Timer4 には、[レジスタ 16-1](#) に示す制御レジスタがあります。制御ビット TMR4ON (T4CON<2>) をクリアして Timer4 を遮断する事で消費電力を低減できます。Timer4 のプリスケアラとポストスケアラの選択も、このレジスタによって制御します。[図 16-1](#) に、Timer4 モジュールの概略ブロック図を示します。

## 16.1 Timer4 の動作

Timer4 は、CCP モジュールの PWM モードのためのタイムベースとして使えます。TMR4 レジスタは読み書き可能で、全てのデバイスリセットによってクリアされます。入力クロック (Fosc/4) には、1:1、1:4、1:16 のプリスケアラ オプションがあり、制御ビット T4CKPS<1:0> (T4CON<1:0>) で選択します。TMR4 の一致出力は、4ビットのポストスケアラ (1:1 ~ 1:16 の分周が可能) を介して TMR4 割り込みを生成し、フラグビット TMR4IF (PIR3<3>) にラッチされます。

プリスケアラ カウンタとポストスケアラ カウンタは、以下のいずれかが発生するとクリアされます。

- TMR4 レジスタへの書き込み
- T4CON レジスタへの書き込み
- 何らかのデバイスリセット (パワーオンリセット、MCLR リセット、ウォッチドッグタイマ リセット、ブラウナウト リセット)

T4CON レジスタに書き込んでも TMR4 はクリアされません。

**レジスタ 16-1: T4CON: Timer4 制御レジスタ**

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	T4OUTPS3	T4OUTPS2	T4OUTPS1	T4OUTPS0	TMR4ON	T4CKPS1	T4CKPS0
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

- bit 7      **未実装:** 「0」として読み出し
- bit 6-3      **T4OUTPS<3:0>:** Timer4 出力ポストスケアラ値選択ビット  
 0000 = 1:1 のポストスケアラ値を選択する  
 0001 = 1:2 のポストスケアラ値を選択する  
 •  
 •  
 •  
 1111 = 1:16 のポストスケアラ値を選択する
- bit 2      **TMR4ON:** Timer4 ON/OFF 制御ビット  
 1 = Timer4 を ON にする  
 0 = Timer4 を OFF にする
- bit 1-0      **T4CKPS<1:0>:** Timer4 クロック プリスケアラ値選択ビット  
 00 = 1:1 のプリスケアラ値を選択する  
 01 = 1:4 のプリスケアラ値を選択する  
 1x = 1:16 のプリスケアラ値を選択する

# PIC18F97J60 ファミリ

## 16.2 Timer4 割り込み

Timer4 モジュールには 8 ビットの周期レジスタ PR4 があります。このレジスタは読み書き共に可能です。Timer4 は 00h から PR4 に一致するまでインクリメントし、次のインクリメントサイクルで 00h にリセットされます。リセット時、PR4 レジスタは FFh に初期化されます。

## 16.3 TMR4 の出力

TMR4 の出力(ポストスケラ入力前)は、CCPx/ECCPx モジュールの PWM タイムベースとしてのみ使われず。Timer2 の出力のように MSSPx モジュールの baud レートクロックとしては使われません。

図 16-1: Timer4 のブロック図

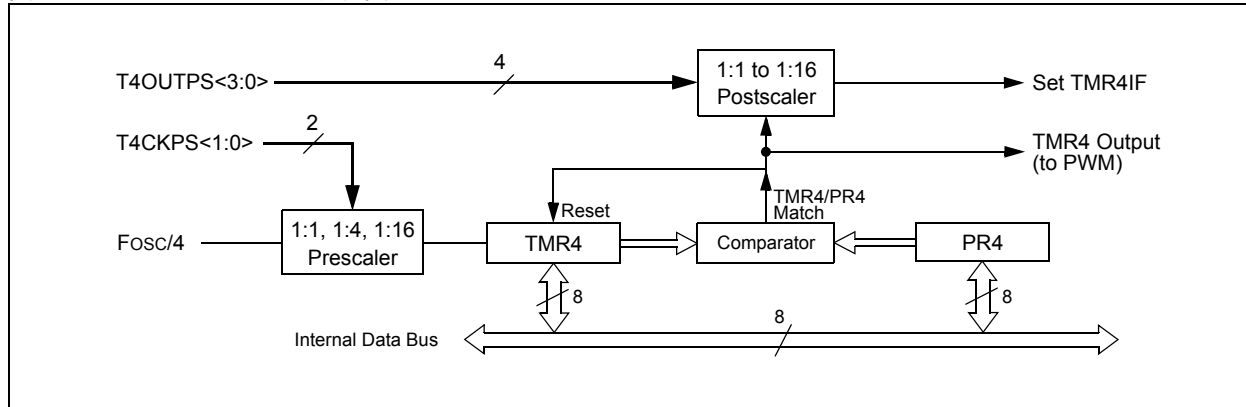


表 16-1: Timer4 のタイマ / カウンタ動作に関連するレジスタ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	リセット値の記載ページ
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	69
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	71
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	71
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	71
TMR4	Timer4 レジスタ								72
T4CON	-	T4OUTPS3	T4OUTPS2	T4OUTPS1	T4OUTPS0	TMR4ON	T4CKPS1	T4CKPS0	72
PR4	Timer4 周期レジスタ								72

凡例: - = 未実装ビット、「0」として読み出されます。網掛けの部分は Timer4 モジュールでは使いません。

## 17.0 キャプチャ/コンペア/PWM (CCP) モジュール

PIC18F97J60ファミリの全デバイスは合計5個のCCP (キャプチャ/コンペア/PWM) モジュールを搭載しています。これらのうち2つ (CCP4 と CCP5) は、標準型のキャプチャ、コンペア、パルス幅変調 (PWM) モードを実装します。ここでは、これらについて説明します。その他の3つのモジュール (ECCP1、ECCP2、ECCP3) は、標準的なキャプチャ/コンペアモードに加えて、拡張PWMモードを実装します。これらについては、[セクション 18.0「拡張キャプチャ/コンペア/PWM \(ECCP\) モジュール」](#)で説明します。

CCPx/ECCPx の各モジュールには、16 ビットのキャプチャレジスタ、16 ビットのコンペアレジスタまたはPWM マスタ/スレーブ デューティ サイクルレジスタとして動作する、16 ビットレジスタが1つ含まれます。説明を分かりやすくするために、このセクシ

ンでは全てのCCPxモジュールの動作を、CCP4の場合について説明しますが、その内容はCCP5にも同様に適用されます。

この章で説明するキャプチャとコンペアの動作は、全ての標準CCPxモジュールと拡張CCPxモジュールに適用されます。[セクション 17.4「PWM モード」](#)で説明するPWMモードの動作は、CCP4とCCP5だけに適用されます。

**Note:** このセクションと[セクション 18.0「拡張キャプチャ/コンペア/PWM \(ECCP\) モジュール」](#)の全体を通して、特定のCCPモジュールに関連するレジスタ名とビット名は、個々のモジュール番号の代わりに「x」または「y」を使った総称名で表します。従って、「CCPxCON」はECCP1、ECCP2、ECCP3、CCP4、CCP5のいずれかに対応する制御レジスタを表します。

レジスタ 17-1: CCPxCON: CCPx 制御レジスタ (CCP4 と CCP5)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	DCxB1	DCxB0	CCPxM3	CCPxM2	CCPxM1	CCPxM0
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

bit 7-6      **未実装:** 「0」として読み出し

bit 5-4      **DCxB<1:0>:** CCPx モジュール PWM デューティ サイクル Bit 1 と Bit 0

キャプチャモード:

未使用

コンペアモード:

未使用

PWM モード:

これらのビットは、10 ビットの PWM デューティ サイクル値の最下位 2 ビット (Bit 1 と Bit 0) です。デューティ サイクルの上位 8 ビット (DCxB<9:2>) は、CCPRxL 内で設定します。

bit 3-0      **CCPxM<3:0>:** CCPx モジュールモード選択ビット

0000 = キャプチャ/コンペア/PWM を無効にする (CCPx モジュールをリセット)

0001 = 予約済み

0010 = コンペアモード、一致時に出力をトグル (CCPxIF ビットがセットされる)

0011 = 予約済み

0100 = キャプチャモード、立ち下がりエッジごと

0101 = キャプチャモード、立ち上がりエッジごと

0110 = キャプチャモード、立ち上がりエッジ 4 回ごと

0111 = キャプチャモード、立ち上がりエッジ 16 回ごと

1000 = コンペアモード、CCPx ピンを Low に初期化しコンペアの一致時に CCPx ピンを High にする (CCPxIF ビットがセットされる)

1001 = コンペアモード、CCPx ピンを High に初期化しコンペアの一致時に CCPx ピンを Low にする (CCPxIF ビットがセットされる)

1010 = コンペアモード、コンペアの一致時にソフトウェア割り込みを生成する (CCPxIF ビットがセットされ、CCPx ピンは I/O の状態を反映する)

1011 = 予約済み

11xx = PWM モード

# PIC18F97J60 ファミリ

## 17.1 CCPx モジュールの設定

各キャプチャ/コンペア/PWM モジュールには、制御レジスタ (総称名は CCPxCON) とデータレジスタ (CCPRx) が関連付けられています。データレジスタは、さらに2つの8ビットレジスタ CCPRxL (下位バイト) と CCPRxH (上位バイト) に分けられます。全てのレジスタは読み書きの両方が可能です。

### 17.1.1 CCPx/ECCPx モジュールとタイマリソース

CCPx/ECCPx モジュールは、選択したモードに応じて、Timer1、2、3、4 のいずれかを使います。Timer1 と Timer3 はキャプチャまたはコンペアモードのモジュールで使えます。一方、Timer2 と Timer4 は PWM モードのモジュールで使えます。

表 17-1: CCPx/ECCPx モード - タイマリソース

CCPx/ECCPx モード	タイマリソース
キャプチャ コンペア PWM	Timer1 または Timer3 Timer1 または Timer3 Timer2 または Timer4

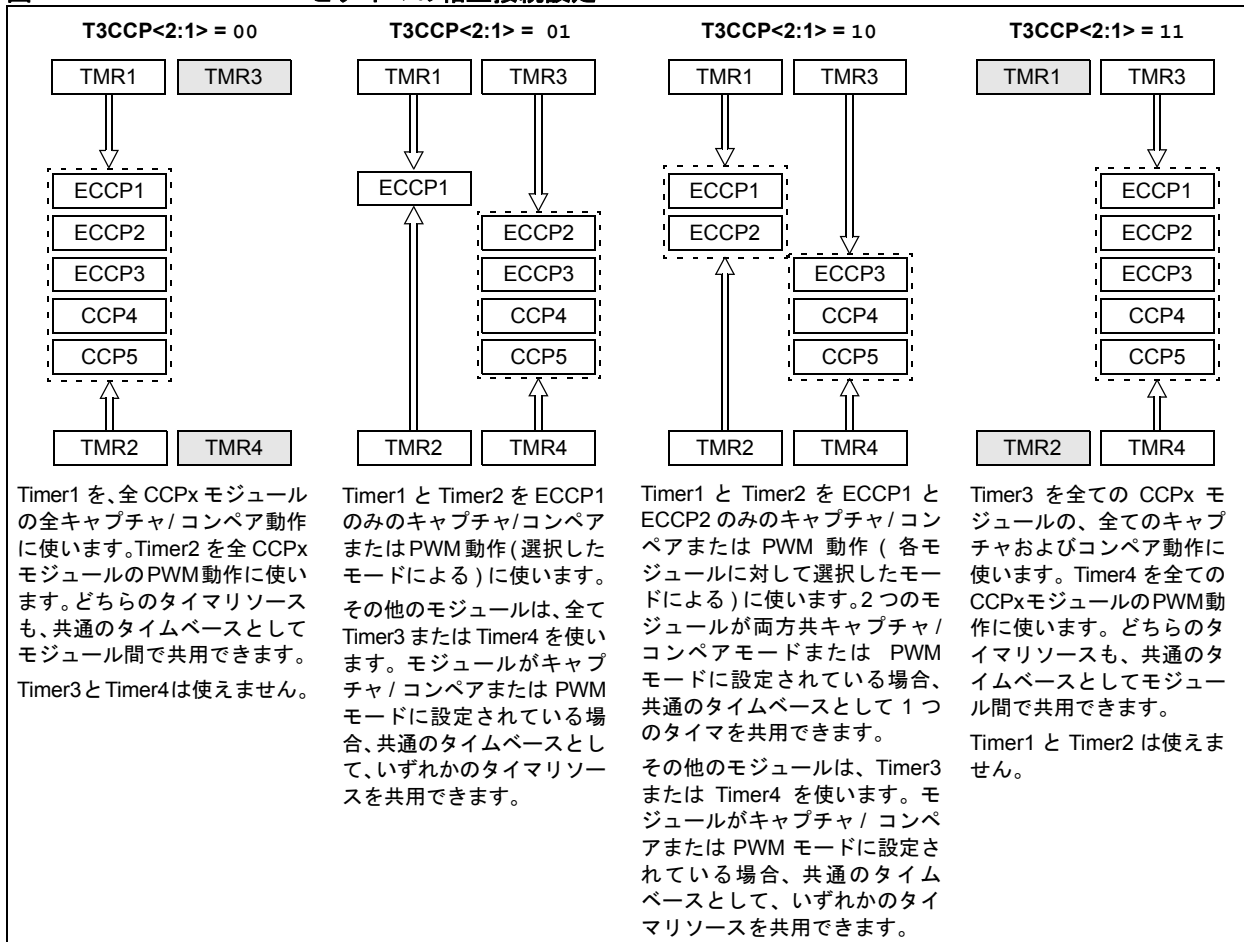
モジュールに対する特定のタイマの割り当ては、T3CONレジスタ(183 ページの [レジスタ 15-1](#))のCCPx に対するタイマ イネーブルビットで決まります。選択した設定に応じて、最大 4 つのタイマを同時にアクティブにできます。同じ設定 (キャプチャ/コンペア または PWM) のモジュールはタイマリソースを共用します。図 17-1 に可能な設定を示します。

### 17.1.2 ECCP2 のピン割り当て

ECCP2 のピン割り当て (キャプチャ入力、コンペアおよび PWM 出力) は、デバイス設定に基づいて変更できます。CCP2MX コンフィグレーション ビットは、ECCP2 をどのピンと多重化するかを決定します。既定値では RC1 と多重化されます (CCP2MX = 1)。コンフィグレーション ビットをクリアした場合、ECCP2 はマイクロコントローラモードの 80 ピンと 100 ピンデバイスでは RE7 と多重化され、拡張マイクロコントローラモードの 100 ピンデバイスでは RB3 と多重化されます。

ECCP2 のピン割り当てを変更した場合、ポートピンに必要な設定はいずれも自動的に変更されません。このため、ECCP2 を正常に動作させるには、TRIS レジスタがピン配置に関わらず適切に設定されている事をユーザが常に確認する必要があります。

図 17-1: CCPx/ECCPx とタイマの相互接続設定



## 17.2 キャプチャモード

キャプチャモードでは、対応する CCPx ピンでイベントが発生すると、TMR1 または TMR3 レジスタの 16 ビットの値が CCPRxH:CCPRxL レジスタペアにキャプチャされます。イベントとしては、以下のいずれか 1 つを指定します。

- 全ての立ち下がりエッジ
- 全ての立ち上がりエッジ
- 立ち上がりエッジ 4 回ごと
- 立ち上がりエッジ 16 回ごと

イベントはモード選択ビット CCPxM<3:0> (CCPxCON<3:0>) で選択します。キャプチャが実行されると、割り込み要求フラグビット CCPxIF がセットされます。このフラグは、ソフトウェアによってクリアする必要があります。CCPRx レジスタの値を読み出す前に再度キャプチャ イベントが発生した場合、前回キャプチャされた値は新しい値によって上書きされます。

### 17.2.1 CCPx ピンの設定

キャプチャモードで使う CCPx ピンは、対応する TRIS 方向ビットをセットする事で入力に設定する必要があります。

**Note:** RG4/CCP5/P1D を出力に設定した場合、そのポートへの書き込み動作を実行するとキャプチャ条件が発生します。

### 17.2.2 Timer1/Timer3 モード選択

キャプチャ機能で使うタイマ (Timer1 および Timer3 またはどちらか一方) は、タイマモードまたは同期カウンタモードで動作させておく必要があります。非同期カウンタモードではキャプチャ機能を使う事ができません。各 CCPx モジュールで使うタイマは、T3CON レジスタで選択します ([セクション 17.1.1「CCPx/ECCPx モジュールとタイマ リソース」](#) 参照)。

### 17.2.3 ソフトウェア割り込み

キャプチャモードを別のモードに変更した場合、不正なキャプチャ割り込みが発生される事があります。このような誤った割り込みの発生を避けるために、ユーザは割り込みイネーブルビット CCPxIE をクリアしておく必要があります。また、このような動作モードの変更後は、割り込みフラグビット CCPxIF もクリアする必要があります。

### 17.2.4 CCPx プリスケーラ

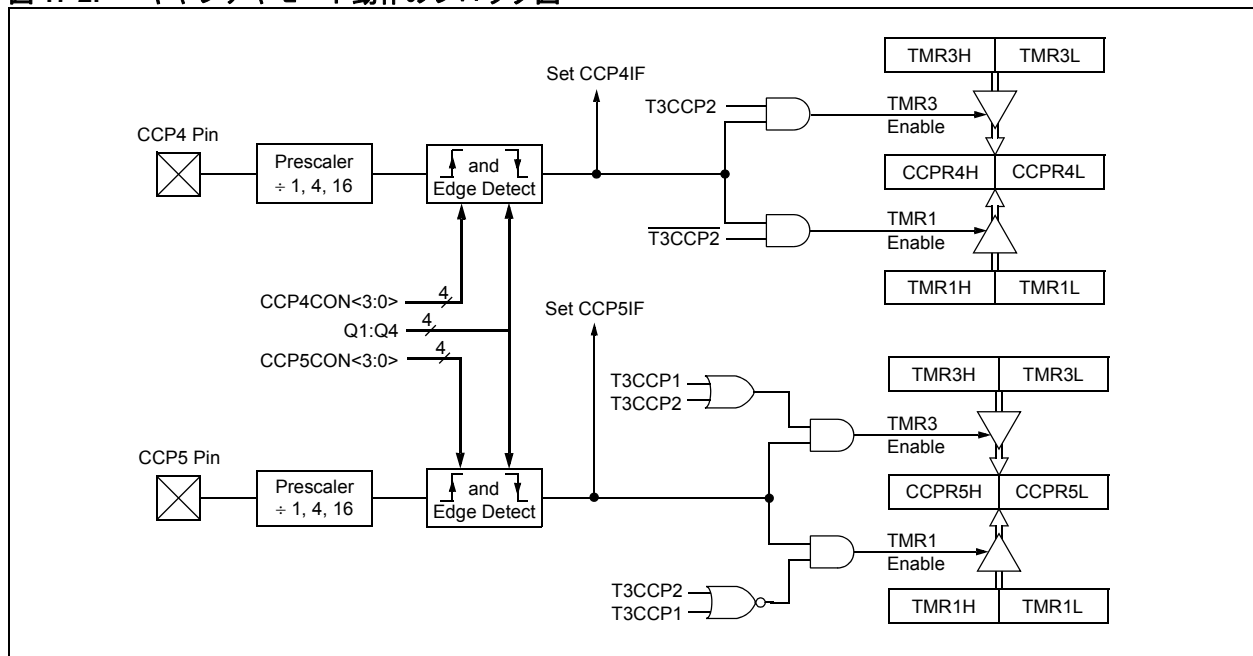
キャプチャモードには 4 つのプリスケーラ設定があります。いずれを選択するかは、動作モード選択の一部としてモード選択ビット (CCPxM<3:0>) によって指定します。CCPx モジュールを OFF、またはキャプチャモードを無効に設定すると、プリスケーラ カウンタがクリアされます。これは、全てのリセットでプリスケーラ カウンタがクリアされる事を意味します。

あるキャプチャ プリスケーラから他のプリスケーラに切り換えると割り込みが発生する場合があります。また、切り換え時にはプリスケーラ カウンタがクリアされないため、変更後の最初のキャプチャがゼロ以外からカウントを開始したプリスケーラによって発生する恐れもあります。[例 17-1](#) に、キャプチャ プリスケーラを切り換える際に推奨する方法を示します。この例ではプリスケーラ カウンタをクリアする事で、誤った割り込みの発生を回避します。

#### 例 17-1: キャプチャ プリスケーラの変更 (CCP5 の例を表示)

```
CLRf    CCP5CON    ; Turn CCP module off
MOVLW  NEW_CAPT_PS ; Load WREG with the
                ; new prescaler mode
                ; value and CCP ON
MOVWF   CCP5CON    ; Load CCP5CON with
                ; this value
```

図 17-2: キャプチャモード動作のブロック図



# PIC18F97J60 ファミリ

## 17.3 コンペアモード

コンペアモードでは、CCPRx レジスタの 16 ビットの値を、TMR1 または TMR3 レジスタペアのいずれかの値と常時比較します。2 つの値が一致すると、CCPx ピンは以下のいずれかの動作を示します。

- High に駆動
- Low に駆動
- トグル (High から Low へ、または Low から High へ)
- 変化なし (= I/O ラッチの状態を反映)

ピンの動作はモード選択ビット (CCPxM<3:0>) の値によって決まります。同時に、割り込みフラグビット CCPxIF がセットされます。

### 17.3.1 CCPx ピンの設定

ユーザは、対応する TRIS ビットをクリアして、CCPx ピンを出力に設定する必要があります。

**Note:** CCP5CON レジスタをクリアすると、RG4 コンペア出力ラッチ ( デバイスの設定による ) が強制的に既定値の Low レベルに設定されます。このラッチは、PORTB または PORTC の I/O データラッチとは別のラッチです。

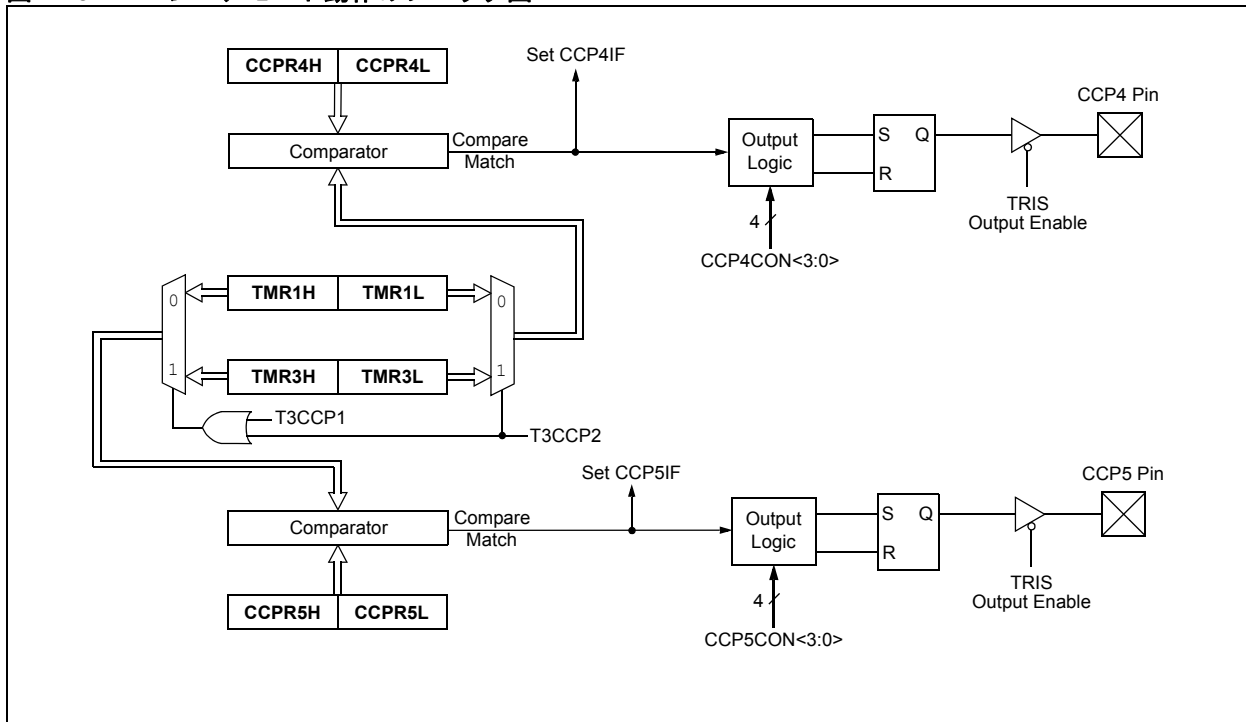
### 17.3.2 Timer1/Timer3 モード選択

CCPx モジュールでコンペア機能を使う場合、Timer1 および Timer3 ( またはどちらか一方 ) はタイマモードまたは同期カウンタモードで動作している必要があります。非同期カウンタモードでは、コンペア機能が動作しない可能性があります。

### 17.3.3 ソフトウェア割り込みモード

ソフトウェア割り込み生成モードを選択した場合 (CCPxM<3:0> = 1010)、対応する CCPx ピンの動作に変化はありません。CCPx 割り込みだけが生成され (有効にしている場合)、CCPxIF ビットがセットされます。

図 17-3: コンペアモード動作のブロック図





# PIC18F97J60 ファミリ

表 17-2: キャプチャ、コンペア、Timer1、Timer3に関連するレジスタ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	リセット 値の記載 ページ
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	69
RCON	IPEN	-	CM	RI	TO	PD	POR	BOR	70
PIR1	PSPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	71
PIE1	PSPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	71
IPR1	PSPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	71
PIR2	OSCFIF	CMIF	ETHIF	r	BCL1IF	-	TMR3IF	CCP2IF	71
PIE2	OSCFIE	CMIE	ETHIE	r	BCL1IE	-	TMR3IE	CCP2IE	71
IPR2	OSCFIP	CMIP	ETHIP	r	BCL1IP	-	TMR3IP	CCP2IP	71
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	71
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	71
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	71
TRISG	TRISG7	TRISG6	TRISG5	TRISG4	TRISG3 <sup>(1)</sup>	TRISG2	TRISG1	TRISG0	71
TMR1L	Timer1 レジスタ下位バイト								70
TMR1H	Timer1 レジスタ上位バイト								70
T1CON	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	70
TMR3H	Timer3 レジスタ上位バイト								70
TMR3L	Timer3 レジスタ下位バイト								70
T3CON	RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	T3SYNC	TMR3CS	TMR3ON	71
CCPR4L	キャプチャ/コンペア/PWM レジスタ 4 下位バイト								72
CCPR4H	キャプチャ/コンペア/PWM レジスタ 4 上位バイト								72
CCPR5L	キャプチャ/コンペア/PWM レジスタ 5 下位バイト								73
CCPR5H	キャプチャ/コンペア/PWM レジスタ 5 上位バイト								73
CCP4CON	-	-	DC4B1	DC4B0	CCP4M3	CCP4M2	CCP4M1	CCP4M0	73
CCP5CON	-	-	DC5B1	DC5B0	CCP5M3	CCP5M2	CCP5M1	CCP5M0	73

凡例: - = 未実装ビット、「0」として読み出されます。r = 予約済みです。網掛けの部分はキャプチャ/コンペア、Timer1、Timer3 では使いません。

Note 1: このビットは 80 ピンと 100 ピンデバイスにのみ実装されています。それ以外のデバイスでは未実装で「0」として読み出されます。

# PIC18F97J60 ファミリ

## 17.4 PWM モード

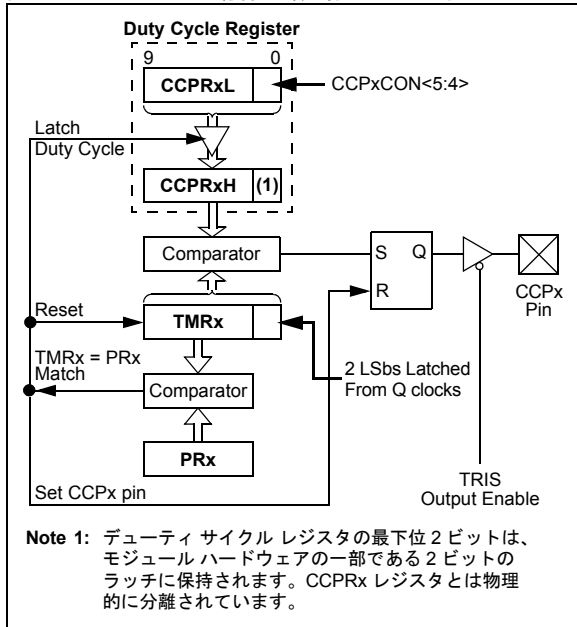
パルス幅変調 (PWM) モードでは、CCPx ピンに最大分解能 10 ビットの PWM 出力が生成されます。CCP4 ピンと CCP5 ピンは PORTG のデータラッチと多重化されているため、これらのピンを出力にするには対応する TRISG ビットをクリアする必要があります。

**Note:** CCP4CON または CCP5CON レジスタをクリアすると、RG3 または RG4 出力ラッチ (デバイスの設定による) が強制的に既定値の Low レベルに設定されます。これは PORTG の I/O データラッチとは別のラッチです。

図 17-4 に、PWM モードの CCPx モジュールの概略ブロック図を示します。

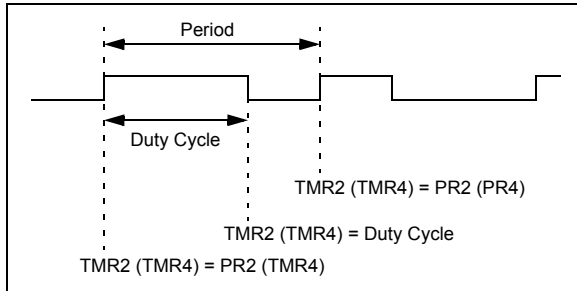
CCPx モジュールを PWM 動作用に設定する手順については、[セクション 17.4.3 「PWM 動作の設定」](#) を参照してください。

図 17-4: PWM 動作の概略ブロック図



PWM 出力 (図 17-5) にはタイムベース (周期) と、出力が High に保持される期間 (デューティ サイクル) があります。PWM の周波数は、周期の逆数です (1/周期)。

図 17-5: PWM 出力



### 17.4.1 PWM 周期

PWM 周期は、PR2 (PR4) レジスタに書き込んで指定します。PWM 周期は式 17-1 によって計算できます。

式 17-1:

$$\text{PWM 周期} = \frac{[(PR2) + 1] \cdot 4 \cdot T_{osc}}{(\text{TMR2 プリスケール値})}$$

PWM 周波数は、 $1/[\text{PWM 周期}]$  として定義されます。TMR2 (TMR4) と PR2 (PR4) が等しくなると、直後のインクリメントサイクルで以下の 3 つのイベントが実行されます。

- TMR2 (TMR4) がクリアされる
- CCPx ピンがセットされる (例外: PWM のデューティ サイクル = 0% の場合、CCPx ピンはセットされない)
- PWM のデューティ サイクルが CCPRxL から CCPRxH にラッチされる

**Note:** Timer2 と Timer4 のポストスケラ ([セクション 14.0 「Timer2 モジュール」](#) と [セクション 16.0 「Timer4 モジュール」](#) 参照) は、PWM 周波数の決定には使いません。ポストスケラによって、PWM 出力とは異なる周波数のサーボ更新レートが使えます。

### 17.4.2 PWM デューティ サイクル

PWM デューティ サイクルは、CCPRxL レジスタと CCPxCON<5:4> ビットに書き込む事で指定します。最大分解能は 10 ビットです。CCPRxL には上位 8 ビット、CCPxCON<5:4> には下位 2 ビットを格納します。この 10 ビットの値は CCPRxL:CCPxCON<5:4> のように表します。PWM のデューティ サイクル比を時間で計算するには、式 17-2 を使います。

式 17-2:

$$\text{PWM デューティ サイクル} = \frac{(\text{CCPRxL}:\text{CCPxCON}\langle 5:4 \rangle) \cdot T_{osc}}{(\text{TMRx プリスケール値})}$$

CCPRxL と CCPxCON<5:4> にはいつでも書き込む事ができますが、PR2 (PR4) と TMR2 (TMR4) が一致するまで、すなわち周期が完了するまでデューティ サイクルの値は CCPRxH にラッチされません。PWM モードでは、CCPRxH は読み出し専用レジスタです。

# PIC18F97J60 ファミリ

CCPRxH レジスタと 2 ビットの内部ラッチが PWM デューティサイクルの二重バッファとして使われます。この二重バッファは、グリッチのない PWM 動作に不可欠です。

CCPRxH と 2 ビットのラッチに格納された値が、TMR2 (TMR4) の値と内部の 2 ビット Q クロックまたは TMR2 (TMR4) プリスケアラの 2 ビットを連結した値に一致すると、CCPx ピンがクリアされます。

PWM の周波数に対する最大分解能 (ビット) は、式 17-3 で求める事ができます。

## 式 17-3:

$$\text{PWM Resolution (max)} = \frac{\log\left(\frac{F_{\text{OSC}}}{F_{\text{PWM}}}\right)}{\log(2)} \text{ bits}$$

**Note:** PWM のデューティサイクル値が PWM 周期よりも長い場合、CCPx ピンはクリアされません。

## 17.4.3 PWM 動作の設定

CCPx モジュールを PWM 動作に設定する手順は以下の通りです。

1. PR2 (PR4) レジスタへの書き込みにより PWM の周期を設定する。
2. CCPRxL レジスタと CCPxCON<5:4> ビットへの書き込みにより PWM のデューティサイクルを設定する。
3. 対応する TRIS ビットをクリアして CCPx ピンを出力に設定する。
4. TMR2 (TMR4) のプリスケール値を設定し、T2CON (T4CON) への書き込みにより Timer2 (Timer4) を有効にする。
5. CCPx モジュールを PWM 動作用に設定する。

表 17-3: 40 MHz における PWM の周波数と分解能の例

PWM の周波数	2.44 kHz	9.77 kHz	39.06 kHz	156.25 kHz	312.50 kHz	416.67 kHz
タイマのプリスケアラ (1、4、16)	16	4	1	1	1	1
PR2 の値	FFh	FFh	FFh	3Fh	1Fh	17h
最大分解能 (ビット)	10	10	10	8	7	6.58

# PIC18F97J60 ファミリ

表 17-4: PWM、Timer2、Timer4 に関連するレジスタ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	リセット 値の記載 ページ
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	69
RCON	IPEN	-	$\overline{CM}$	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$	70
PIR1	PSPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	71
PIE1	PSPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	71
IPR1	PSPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	71
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	71
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	71
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	71
TRISG	TRISG7	TRISG6	TRISG5	TRISG4	TRISG3 <sup>(1)</sup>	TRISG2	TRISG1	TRISG0	71
TMR2	Timer2 レジスタ								70
PR2	Timer2 周期レジスタ								70
T2CON	-	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	70
TMR4	Timer4 レジスタ								72
PR4	Timer4 周期レジスタ								72
T4CON	-	T4OUTPS3	T4OUTPS2	T4OUTPS1	T4OUTPS0	TMR4ON	T4CKPS1	T4CKPS0	72
CCPR4L	キャプチャ/コンペア/PWM レジスタ 4 下位バイト								72
CCPR4H	キャプチャ/コンペア/PWM レジスタ 4 上位バイト								72
CCPR5L	キャプチャ/コンペア/PWM レジスタ 5 下位バイト								73
CCPR5H	キャプチャ/コンペア/PWM レジスタ 5 上位バイト								73
CCP4CON	-	-	DC4B1	DC4B0	CCP4M3	CCP4M2	CCP4M1	CCP4M0	73
CCP5CON	-	-	DC5B1	DC5B0	CCP5M3	CCP5M2	CCP5M1	CCP5M0	73

凡例: — = 未実装、「0」として読み出されます。網掛けの部分は PWM、Timer2、Timer4 では使いません。

Note 1: このビットは 80 ピンと 100 ピンデバイスにのみ実装されています。それ以外のデバイスでは未実装で「0」として読み出されます。

## 18.0 拡張キャプチャ/コンペア/PWM (ECCP) モジュール

PIC18F97J60 ファミリでは、CCP モジュールのうち 3 つが拡張 PWM 機能を備えた標準 CCP モジュールとして実装されています。これらのモジュールは、2 つまたは 4 つの出力チャンネル、ユーザによる極性選択、デッドバンド制御、自動シャットダウンと再起動等の機能を備えています。拡張機能については、[セクション 18.4 「拡張 PWM モード」](#) で詳細に説明します。ECCPx モジュールのキャプチャ、コンペア、シングル出力 PWM 機能は、標準 CCPx モジュールで説明したものと同じです。

[レジスタ 18-1](#) に拡張 CCPx モジュールの制御レジスタを示します。PWM 機能を制御するために最上位 2 ビットが実装されている点が、CCP4CON/CCP5CON レジスタとの違いです。

拡張 CCPxCON レジスタにより使えるモードが多い事に加えて、ECCPx モジュールには拡張 PWM 動作と自動シャットダウン機能に関連する、以下の 2 つのレジスタが追加されています。

- ECCPxDEL (デッドバンド遅延)
- ECCPxAS (自動シャットダウンの設定)

# PIC18F97J60 ファミリ

## レジスタ 18-1: CCPxCON: 拡張 CCPx 制御レジスタ (ECCP1/ECCP2/ECCP3)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PxM1	PxM0	DCxB1	DCxB0	CCPxM3	CCPxM2	CCPxM1	CCPxM0
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

- bit 7-6      **PxM<1:0>**: 拡張 PWM 出力コンフィグレーション ビット  
CCPxM<3:2> = 00、01、10 の場合:  
 xx = PxA にキャプチャ/コンペア入力/出力を割り当て、PxB、PxC、PxD をポートピンに割り当てる  
CCPxM<3:2> = 11 の場合:  
 00 = シングル出力: PxA に変調信号を出力、PxB、PxC、PxD をポートピンに割り当てる  
 01 = フルブリッジ順方向出力: P1D に変調信号を出力、P1A をアクティブ、P1B、P1C を非アクティブに駆動する  
 10 = ハーフブリッジ出力: P1A、P1B にデッドバンド制御された変調信号を出力、P1C、P1D をポートピンに割り当てる  
 11 = フルブリッジ逆方向出力: P1B に変調信号を出力、P1C をアクティブ、P1A、P1D を非アクティブに駆動する
- bit 5-4      **DCxB<1:0>**: ECCPx モジュール PWM デューティ サイクル Bit 1 と Bit 0  
キャプチャモード:  
 未使用  
コンペアモード:  
 未使用  
PWM モード:  
 これらのビットは、10 ビットの PWM デューティ サイクル値の下位 2 ビットである。デューティ サイクルの上位 8 ビットは CCPxL 内で設定する。
- bit 3-0      **CCPxM<3:0>**: ECCPx モジュールモード選択ビット  
 0000 = キャプチャ/コンペア/PWM を無効にする (ECCPx モジュールをリセットする)  
 0001 = 予約済み  
 0010 = コンペアモード、一致時に出力をトグルする  
 0011 = キャプチャモード  
 0100 = キャプチャモード、立ち下がりエッジごと  
 0101 = キャプチャモード、立ち上がりエッジごと  
 0110 = キャプチャモード、立ち上がりエッジ 4 回ごと  
 0111 = キャプチャモード、立ち上がりエッジ 16 回ごと  
 1000 = コンペアモード、ECCPx ピンを Low に初期化し、比較一致時に出力をセットする (同時に CCPxIF をセットする)  
 1001 = コンペアモード、ECCPx ピンを High に初期化し、比較一致時に出力をクリアする (同時に CCPxIF をセットする)  
 1010 = コンペアモード、ソフトウェア割り込みのみを生成し、ECCPx ピンは I/O ポートの状態に戻す  
 1011 = コンペアモード、特殊イベントをトリガする (ECCPx が TMR1 または TMR3 をリセットし、CCPxIF ビットをセットする。A/D モジュールが有効な場合は A/D 変換を開始する)<sup>(1)</sup>  
 1100 = PWM モード、PxA と PxC はアクティブ High、PxB と PxD もアクティブ High  
 1101 = PWM モード、PxA と PxC はアクティブ High、PxB と PxD はアクティブ Low  
 1110 = PWM モード、PxA と PxC はアクティブ Low、PxB と PxD はアクティブ High  
 1111 = PWM モード、PxA と PxC はアクティブ Low、PxB と PxD もアクティブ Low

**Note 1:** ECCP1 と ECCP2 にのみ実装されています。ECCP3 の「1010」と同じです。

## 18.1 ECCPx 出力とコンフィグレーション

拡張 CCPx モジュールは、それぞれ選択した動作モードに応じて最大 4 つの PWM 信号を出力できます。PxA ~ PxD と呼ばれるこれらの出力は、各種 I/O ピンと多重化されています。ECCPx ピンの割り当てには固定しているものと、デバイス コンフィグレーションに応じて変化するものがあります。割り当てが変化するピンに対する制御ビットは以下の通りです。

- CCP2MX コンフィグレーション ビット (80 ピン /100 ピンデバイスのみ)
- ECCPMX コンフィグレーション ビット (80 ピン /100 ピンデバイスのみ)
- プログラムメモリの動作モードを設定する EMB コンフィグレーション ビット (100 ピンデバイスのみ)

表 18-1、表 18-2、表 18-3 に、拡張 CCPx モジュールのピン割り当て一覧を示します。I/O ピンを PWM 出力として設定するには、PxMx ビットと CCPxMx ビット (それぞれ CCPxCON<7:6> と <3:0>) を設定して適切な PWM モードを選択する必要があります。また、対応するポートピンの TRIS 方向ビットも出力として設定する必要があります。

### 18.1.1 ECCP1/ECCP3 の出力とプログラムメモリモード

100 ピンデバイスで拡張マイクロコントローラモードを使うと、拡張 PWM モードにおける ECCP1 と ECCP3 のピンに間接的な影響が及びます。既定値では、PWM 出力の P1B/P1C と P3B/P3C は外部メモリバスの上位バイトと共に PORTE ピンと多重化されます。バスが拡張マイクロコントローラモードでアクティブになると、拡張 CCPx 出力はこれにオーバーライドされて使えなくなります。このため、デバイスを既定値のピン設定により拡張マイクロコントローラモードで動作させる場合、ECCP1 と ECCP3 は互換 PWM モード (シングル出力) でしか使えません。

このコンフィグレーションに対する例外は、外部バスのアドレス幅として 12 ビットを選択した場合です (EMB<1:0> コンフィグレーション ビット = 10)。その場合、外部バスがアクティブでも PORTE の上位ピンはデジタル I/O として動作しつづけます。P1B/P1C と P3B/P3C も引き続き拡張 PWM 出力として使えます。

拡張マイクロコントローラモード動作中に追加の PWM 出力が必要なアプリケーションでは、P1B/P1C と P3B/P3C 出力の割り当てを PORTH の上位ビットに変更できます。それには ECCPMX コンフィグレーション ビットをクリアします。

### 18.1.2 ECCP2 の出力とプログラムメモリモード

100 ピンデバイスでは、デバイスのプログラムメモリモード (セクション 6.1.3 「PIC18F9XJ60/9XJ65 のプログラムメモリモード」) もモジュールのピン多重化に影響を与えます。

ECCP2 の入力 / 出力 (ECCP2/P2A) は、3 本のピンのいずれかと多重化できます。全デバイスの既定値割り当て (CCP2MX コンフィグレーション ビットをセット) は RC1 です。CCP2MX をクリアすると、80 ピン /100 ピンデバイスでは ECCP2/P2A の割り当てが RE7 に変更されます。

100 ピンデバイスには、もう 1 つのオプションがあります。これらのデバイスをマイクロコントローラモードで動作させた場合、上記の多重化オプションは引き続き有効です。拡張マイクロコントローラモードで動作させた場合、CCP2MX をクリアすると ECCP2/P2A が RB3 に割り当てられます。

### 18.1.3 ECCP1/ECCP3 による CCP4/CCP5 の使用

ECCP2 モジュールだけが、4 本の専用出力ピンを使えます。これらのピンの I/O ポートやその他の多重化された機能を使わない場合、他の CCPx モジュールに影響を与える事なくこれらのピンを使えます。

これに対して ECCP1 と ECCP3 は、専用出力ピンを 3 本 (ECCPx/PxA、PxB、PxC) しか備えていません。これらのモジュールをクワッド PWM モードに設定した場合、通常は CCP4 または CCP5 に使うピンが、それぞれ ECCP3 と ECCP1 の PxD 出力になります。CCP4 モジュールと CCP5 モジュールは引き続き動作しますが、出力はオーバーライドされます。

### 18.1.4 ECCPx モジュールとタイマリソース

標準 CCPx モジュールと同様に、ECCPx モジュールは、選択したモードに応じて、Timer1、2、3、4 のいずれかを使えます。Timer1 と Timer3 はキャプチャまたはコンペアモードに設定されたモジュールで使えます。一方、Timer2 と Timer4 は PWM モードのモジュールで使えます。タイマリソースの詳細はセクション 17.1.1 「CCPx/ECCPx モジュールとタイマリソース」で説明しています。

# PIC18F97J60 ファミリ

表 18-1: ECCP1 のピン設定

ECCP モード	CCP1CON コンフィグ レーション	RC2	RD0 または RE6 <sup>(1)</sup>	RE5	RG4	RH7 <sup>(2)</sup>	RH6 <sup>(2)</sup>
<b>64 ピンデバイス、80 ピンデバイス、ECCPMX = 1 100 ピンデバイス、ECCPMX = 1、マイクロコントローラ モードまたは アドレス幅 12 ビットの拡張マイクロコントローラ モード:</b>							
互換性のある CCP	00xx 11xx	ECCP1	RD0/RE6	RE5	RG4/CCP5	RH7/AN15	RH6/AN14
デュアル PWM	10xx 11xx	P1A	P1B	RE5	RG4/CCP5	RH7/AN15	RH6/AN14
クワッド PWM	x1xx 11xx	P1A	P1B	P1C	P1D	RH7/AN15	RH6/AN14
<b>80 ピンデバイス、ECCPMX = 0 100 ピンデバイス、ECCPMX = 0、全てのプログラムメモリ モード:</b>							
互換性のある CCP	00xx 11xx	ECCP1	RD0/RE6	RE5/AD13	RG4/CCP5	RH7/AN15	RH6/AN14
デュアル PWM	10xx 11xx	P1A	RD0/RE6	RE5/AD13	RG4/CCP5	P1B	RH6/AN14
クワッド PWM <sup>(3)</sup>	x1xx 11xx	P1A	RD0/RE6	RE5/AD13	P1D	P1B	P1C
<b>100 ピンデバイス、ECCPMX = 1、 アドレス幅 16 ビットまたは 20 ビットの拡張マイクロコントローラ モード:</b>							
互換性のある CCP	00xx 11xx	ECCP1	RD0/RE6	RE5/AD13	RG4/CCP5	RH7/AN15	RH6/AN14

凡例: x = ドントケア、網掛けの部分は、そのモードの ECCP1 では使わない事を意味します。

Note 1: P1B は 64 ピンデバイスでは RD0、80 ピン / 100 ピンデバイスでは RE6 と多重化されます。

2: これらのピンオプションは、64 ピンデバイスでは使えません。

3: ECCP1 をクワッド PWM モードで使う場合、CCP5 ピンの出力が P1D によってオーバーライドされます。CCP5 のこれ以外の機能は全て通常通り動作します。

表 18-2: ECCP2 のピン設定

ECCP モード	CCP2CON コンフィグ レーション	RB3	RC1	RE7	RE2	RE1	RE0
<b>全デバイス、CCP2MX = 1、全てのプログラムメモリ モード:</b>							
互換性のある CCP	00xx 11xx	RB3/INT3	ECCP2	RE7	RE2	RE1	RE0
デュアル PWM	10xx 11xx	RB3/INT3	P2A	RE7	P2B	RE1	RE0
クワッド PWM	x1xx 11xx	RB3/INT3	P2A	RE7	P2B	P2C	P2D
<b>80 ピン / 100 ピンデバイス、CCP2MX = 0、マイクロコントローラ モード:</b>							
互換性のある CCP	00xx 11xx	RB3/INT3	RC1/T1OS1	ECCP2	RE2	RE1	RE0
デュアル PWM	10xx 11xx	RB3/INT3	RC1/T1OS1	P2A	P2B	RE1	RE0
クワッド PWM	x1xx 11xx	RB3/INT3	RC1/T1OS1	P2A	P2B	P2C	P2D
<b>100 ピンデバイス、CCP2MX = 0、拡張マイクロコントローラ モード:</b>							
互換性のある CCP	00xx 11xx	ECCP2	RC1/T1OS1	RE7/AD15	RE2/ $\overline{CS}$	RE1/ $\overline{WR}$	RE0/ $\overline{RD}$
デュアル PWM	10xx 11xx	P2A	RC1/T1OS1	RE7/AD15	P2B	RE1/ $\overline{WR}$	RE0/ $\overline{RD}$
クワッド PWM	x1xx 11xx	P2A	RC1/T1OS1	RE7/AD15	P2B	P2C	P2D

凡例: x = ドントケア、網掛けの部分は、そのモードの ECCP2 では使わない事を意味します。



# PIC18F97J60 ファミリ

表 18-3: ECCP3 のピン設定

ECCP モード	CCP3CON コンフィグ レーション	RD1 または RG0 <sup>(1)</sup>	RE4	RE3	RD2 または RG3 <sup>(1)</sup>	RH5 <sup>(2)</sup>	RH4 <sup>(2)</sup>
<b>64 ピンデバイス、80 ピンデバイス、ECCPMX = 1 100 ピンデバイス、ECCPMX = 1、マイクロコントローラ モード:</b>							
互換性のある CCP	00xx 11xx	ECCP3	RE4	RE3	RD2/RG3	RH5/AN13	RH4/AN12
デュアル PWM	10xx 11xx	P3A	P3B	RE3	RD2/RG3	RH5/AN13	RH4/AN12
クワッド PWM	x1xx 11xx	P3A	P3B	P3C	P3D	RH5/AN13	RH4/AN12
<b>80 ピンデバイス、ECCPMX = 0 100 ピンデバイス、ECCPMX = 0、全てのプログラムメモリ モード:</b>							
互換性のある CCP	00xx 11xx	ECCP3	RE6/AD14	RE5/AD13	RD2/RG3	RH5/AN13	RH4/AN12
デュアル PWM	10xx 11xx	P3A	RE6/AD14	RE5/AD13	RD2/RG3	P3B	RH4/AN12
クワッド PWM <sup>(3)</sup>	x1xx 11xx	P3A	RE6/AD14	RE5/AD13	P3D	P3B	P3C
<b>100 ピンデバイス、ECCPMX = 1、 アドレス幅 12 ビットの拡張マイクロコントローラ モード:</b>							
互換性のある CCP	00xx 11xx	ECCP3	RE4/AD12	RE3/AD11	RD2/RG3	RH5/AN13	RH4/AN12
デュアル PWM	10xx 11xx	P3A	P3B	RE3/AD11	RD2/RG3	RH5/AN13	RH4/AN12
<b>100 ピンデバイス、ECCPMX = 1、 アドレス幅 16 ビットまたは 20 ビットの拡張マイクロコントローラ モード:</b>							
互換性のある CCP	00xx 11xx	ECCP3	RE6/AD14	RE5/AD13	RD2/RG3	RH5/AN13	RH4/AN12

凡例: x = ドントケア、網掛けの部分は、そのモードの ECCP3 では使わない事を意味します。

- Note 1:** ECCP3/P3A と CCP4/P3D は 64 ピンデバイスでは RD1 と RD2、80 ピン / 100 ピンデバイスでは RG0 と RG3 に多重化されます。
- 2:** これらのピンオプションは、64 ピンデバイスでは使えません。
- 3:** ECCP3 をクワッド PWM モードで使う場合、CCP4 ピンの出力が P3D によってオーバーライドされます。CCP4 のこれ以外の機能は全て通常通り動作します。

# PIC18F97J60 ファミリ

## 18.2 キャプチャ/コンペアモード

後述する特殊イベントトリガの動作を除き、ECCPx モジュールのキャプチャ/コンペアモードの動作は CCP4 の場合と同じです。詳細は [セクション 17.2「キャプチャモード」](#) と [セクション 17.3「コンペアモード」](#) で説明しています。

### 18.2.1 特殊イベントトリガ

ECCP1 と ECCP2 は、コンペアモードで CCPRx レジスタペアと選択したタイマの値が一致した時に生成される、内部ハードウェア トリガを備えています。このトリガはその後の動作の開始に使えます。このモードは、CCPxCON<3:0> を「1011」に設定する事で選択します。

ECCP1 または ECCP2 の特殊イベントトリガは、TMR1 または TMR3 レジスタペアをリセットします。どちらがリセットされるかは、現在選択されているタイマリソースによって決まります。この機能によって、CCPRx レジスタを実質的に Timer1 または Timer3 の 16 ビットのプログラマブル周期レジスタとして使えます。さらに、A/D モジュールが有効な場合、ECCP2 の特殊イベントトリガによって A/D 変換も起動されます。

特殊イベントトリガは ECCP3、CCP4、CCP5 には実装されていません。これらのモジュールに対して特殊イベントトリガモードを選択すると、ソフトウェア割り込みモードのコンペアを選択した場合 (CCPxM<3:0>=1010) と同じ効果が得られます。

**Note:** ECCP2 の特殊イベントトリガは、Timer1 または Timer3 の割り込みフラグビットをセットしません。

## 18.3 標準 PWM モード

シングル出力モードに設定された ECCPx モジュールは、[セクション 17.4「PWM モード」](#) で説明した PWM モードの標準 CCPx モジュールと同じ動作を示します。表 18-1 ~ 18-3 で示したように、この設定を「互換 CCP」モードと呼ぶ場合もあります。

**Note:** シングル出力の PWM 動作を設定する場合、[セクション 17.4.3「PWM 動作の設定」](#) と [セクション 18.4.9「PWM 動作の設定」](#) で示すプロセスのどちらでも選ぶ事ができます。後者はより一般的で、シングル出力 PWM とマルチ出力 PWM の両方に適用できます。

## 18.4 拡張 PWM モード

拡張 PWM モードでは、より広範な制御アプリケーションに対応できるように、PWM 出力オプションが追加されます。このモジュールは標準 CCPx モジュールと互換性があり、最大 4 つの出力 (PxA ~ PxD) を提供します。信号の極性も選択できます (アクティブ High またはアクティブ Low)。モジュールの出力モードと極性は、CCPxCON レジスタの PxM<1:0> および CCPxM<3:0> ビット (それぞれ CCPxCON<7:6> と <3:0>) によって設定します。

説明を分かりやすくするために、このセクションでは ECCP1 および TMR2 モジュールの拡張 PWM モード動作を説明します。制御レジスタの名前は ECCP1 で使うものを示します。3 つの拡張モジュールと 2 つのタイマリソースは、全て入れ換え可能であり、動作は同じです。TMR2 または TMR4 は、T3CON の該当するビットを選択する事で PWM 動作を選択できます。

図 18-1 に、PWM 動作の概略ブロック図を示します。出力のいずれにもグリッチが発生しないように、全ての制御レジスタはダブルバッファ構成を取り、新しい PWM サイクルの開始時点 (Timer2 がリセットされる周期の境界) で読み込みを実行します。例外は、ECCP1 のデッドバンド遅延レジスタ ECCP1DEL です。このレジスタには、デューティ サイクルの境界または周期の境界のいずれか (先に発生した方) の時点で値が読み込まれます。バッファ動作によって、モジュールはただちに起動せず、割り当てられたタイマがリセットされるまで待機します。これは、拡張 PWM の波形が、

標準 PWM の波形と完全には一致せず、1 命令サイクル全体 (4 T<sub>osc</sub>) に相当するオフセットが発生する事を意味します。

標準モジュールと同様、対応する TRIS ビットを手作業で出力用に設定する必要があります。

### 18.4.1 PWM 周期

PWM 周期は、PR2 レジスタに書き込んで指定します。PWM の周期は、以下の式で計算できます。

#### 式 18-1:

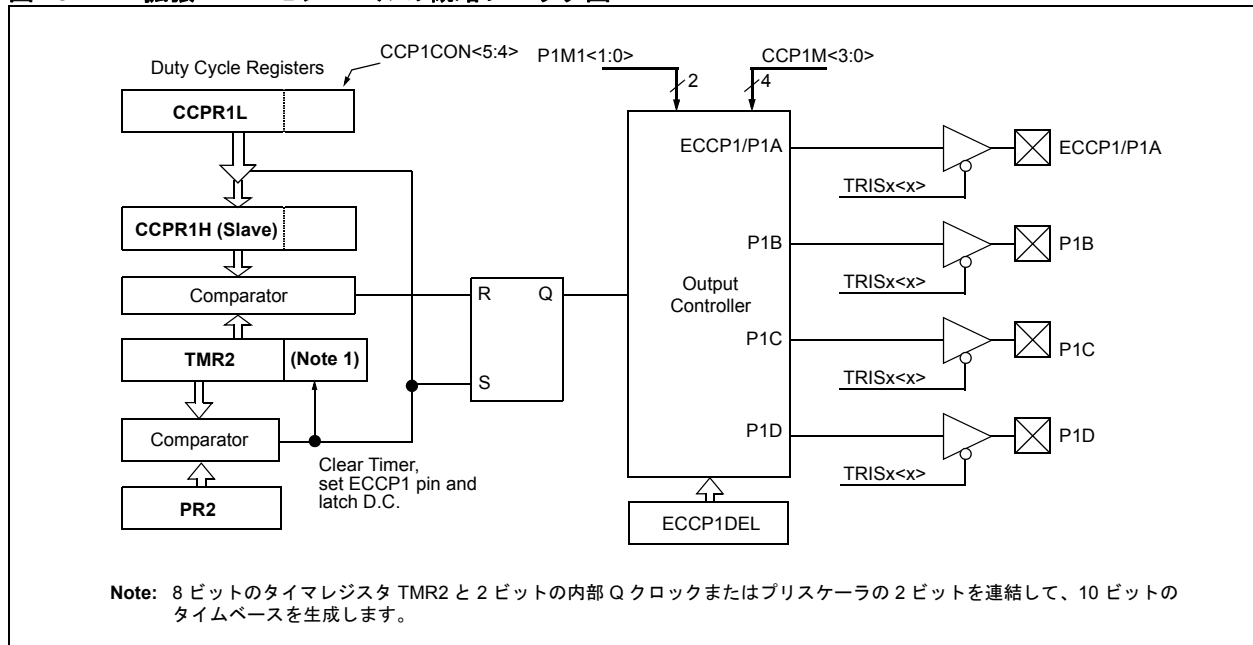
$$\text{PWM 周期} = \frac{[(\text{PR2}) + 1] \cdot 4 \cdot \text{Tosc} \cdot (\text{TMR2 プリスケール値})}{1}$$

PWM 周波数は、1/[PWM 周期] として定義されます。TMR2 と PR2 が等しくなると、直後のインクリメントサイクルで以下の 3 つのイベントが実行されます。

- TMR2 がクリアされる
- ECCP1 ピンがセットされる (PWM のデューティ サイクル = 0% の場合、ECCP1 ピンはセットされません)
- PWM デューティ サイクルが CCPR1L から CCPR1H にコピーされる

**Note:** PWM の周波数は、Timer2 のポストスケラ (セクション 14.0「Timer2 モジュール」参照) に関係なく決定します。ポストスケラによって、PWM 出力とは異なる周波数のサーボ更新レートが使えます。

図 18-1: 拡張 PWM モジュールの概略ブロック図



# PIC18F97J60 ファミリ

## 18.4.2 PWM のデューティ サイクル

PWM のデューティ サイクルは、CCPR1L レジスタと CCP1CON<5:4> ビットに書き込む事で指定します。最大 10 ビットの分解能を使えます。CCPR1L には上位 8 ビット、CCP1CON<5:4> には下位 2 ビットを格納します。この 10 ビットの値は CCPR1L:CCP1CON<5:4> のように表します。PWM のデューティ サイクルは、下式で求める事ができます。

### 式 18-2:

$$\text{PWM デューティ サイクル} = (\text{CCPR1L:CCP1CON<5:4>}) \cdot \text{Tosc} \cdot (\text{TMR2 プリスケール値})$$

CCPR1L と CCP1CON<5:4> にはいつでも書き込む事ができますが、PR2 と TMR2 が一致するまで (すなわち、周期が完了するまで)、デューティ サイクルの値は CCPR1H にコピーされません。PWM モードでは、CCPR1H は読み出し専用レジスタです。

CCPR1H レジスタと 2 ビットの内部ラッチが PWM デューティ サイクルの二重バッファとして使われます。この二重バッファは、グリッチのない PWM 動作に不可欠です。CCPR1H と 2 ビットのラッチに格納された値が、TMR2 の値と内部の 2 ビット Q クロックまたは TMR2 プリスケアラの 2 ビットを連結した値に一致すると、ECCP1 ピンがクリアされます。PWM の周波数に対する最大分解能は、以下の式から計算できます。

### 式 18-3:

$$\text{PWM Resolution (max)} = \frac{\log\left(\frac{F_{\text{OSC}}}{F_{\text{PWM}}}\right)}{\log(2)} \text{ bits}$$

表 18-4: 40 MHz 時の PWM 周波数と分解能の例

PWM の周波数	2.44 kHz	9.77 kHz	39.06 kHz	156.25 kHz	312.50 kHz	416.67 kHz
タイマのプリスケアラ (1, 4, 16)	16	4	1	1	1	1
PR2 の値	FFh	FFh	FFh	3Fh	1Fh	17h
最大分解能 (ビット)	10	10	10	8	7	6.58

**Note:** PWMのデューティ サイクル値がPWM周期よりも長い場合、ECCP1 ピンはクリアされません。

## 18.4.3 PWM の出力コンフィグレーション

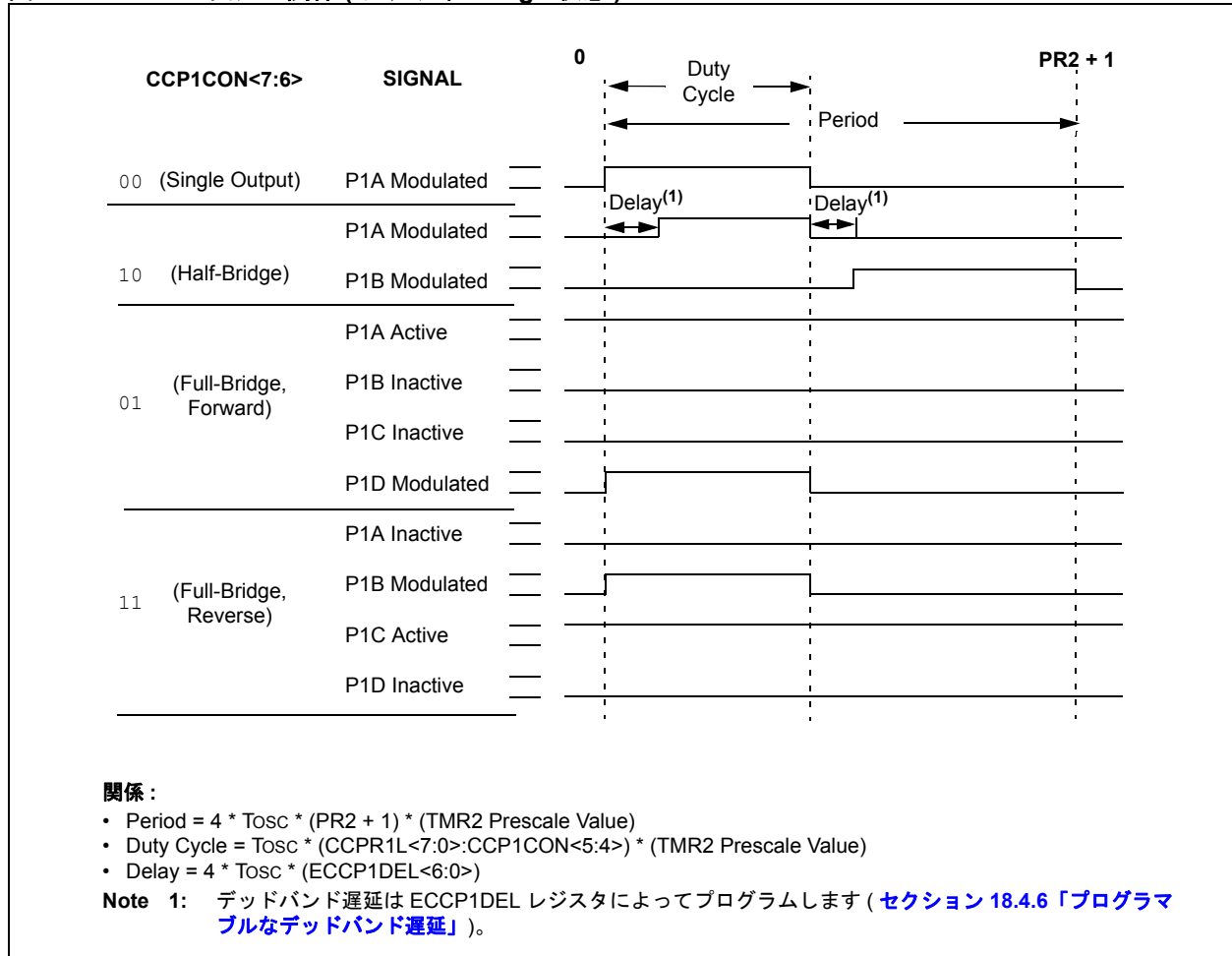
CCP1CON レジスタの P1M<1:0> ビットによって、以下の 4 つのコンフィグレーションのいずれかを選択できます。

- シングル出力
- ハーフブリッジ出力
- フルブリッジ出力、順方向モード
- フルブリッジ出力、逆方向モード

シングル出力モードは、[セクション 18.4「拡張 PWM モード」](#) で説明している標準 PWM モードです。ハーフブリッジおよびフルブリッジ出力モードの詳細は後述します。

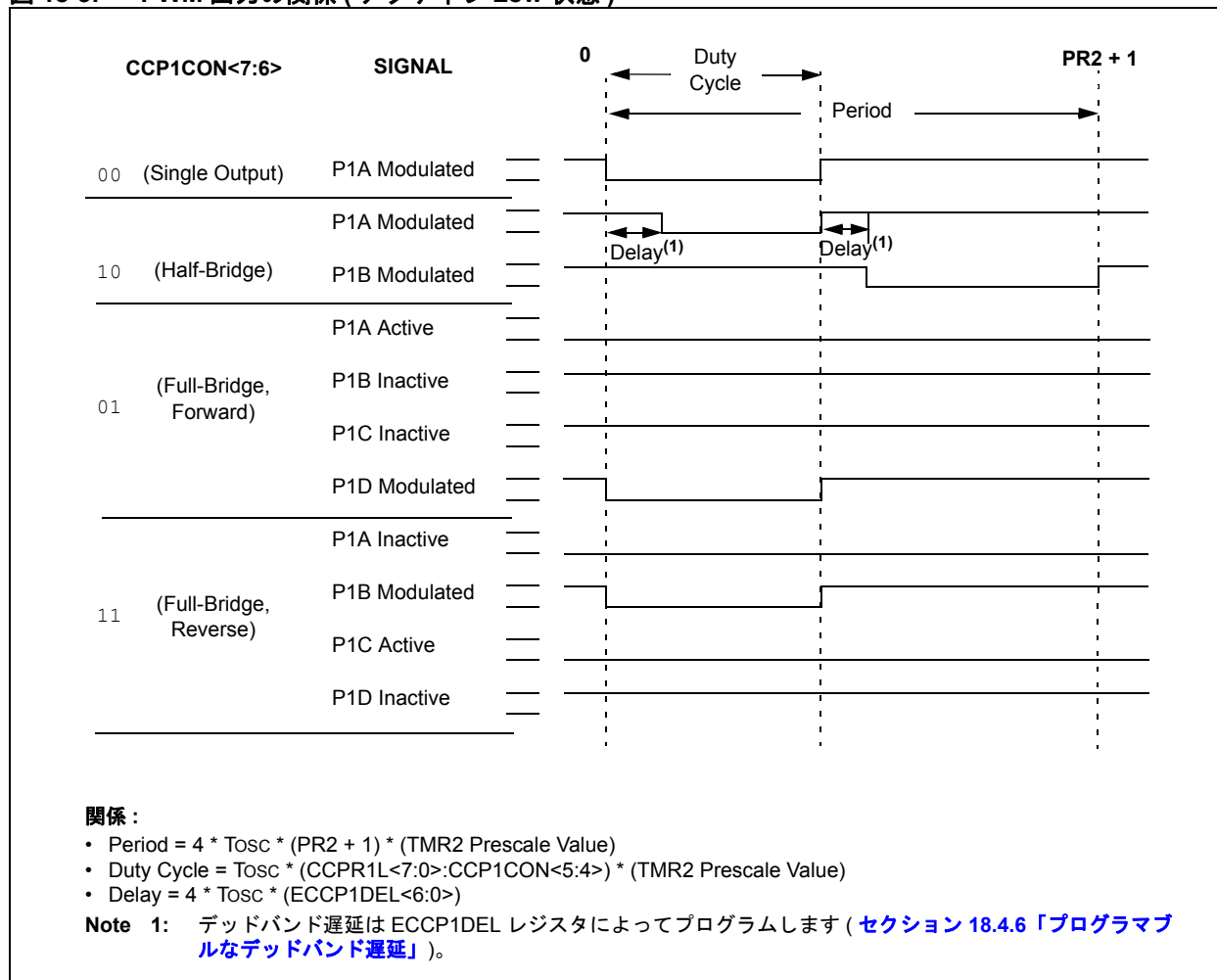
[図 18-2](#) に、全コンフィグレーションによる出力関係を示します。

図 18-2: PWM 出力の関係 (アクティブ High 状態)



# PIC18F97J60 ファミリ

図 18-3: PWM 出力の関係 (アクティブ Low 状態)



## 18.4.4 ハーフブリッジモード

ハーフブリッジ出力モードでは、プッシュプル型の負荷を駆動するために出力として2本のピンを使います。PWM 出力信号が P1A ピンに出力され、相補 PWM 出力信号が P1B ピンに出力されます (図 18-4)。図 18-5 に示すように、このモードはハーフブリッジアプリケーションに使える他、2つの PWM 信号で4つのパワースイッチを制御するフルブリッジアプリケーションにも使えます。

ハーフブリッジ出力モードでは、プログラマブルなデッドバンド遅延を使ってハーフブリッジパワーデバイスの貫通電流を防ぐことができます。P1DC<6:0> ビットの値によって、出力をアクティブに駆動するまでの遅延を命令サイクル数として設定します。この値がデューティサイクルより大きい場合、対応する出力はサイクル全体で非アクティブのままです。デッドバンド遅延動作の詳細は、[セクション 18.4.6 「プログラマブルなデッドバンド遅延」](#)を参照してください。

P1A と P1B 出力は PORTC<2> と PORTE<6> のデータラッチと多重化されているため、TRISC<2> と TRISE<6> ビットをクリアして、P1A と P1B を出力に設定する必要があります。

図 18-4: ハーフブリッジ PWM 出力

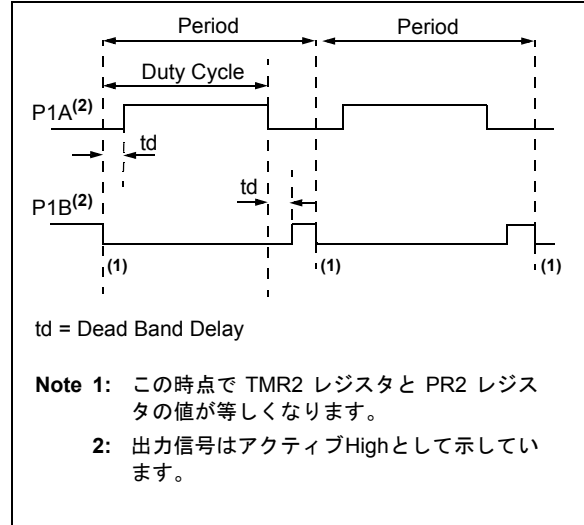
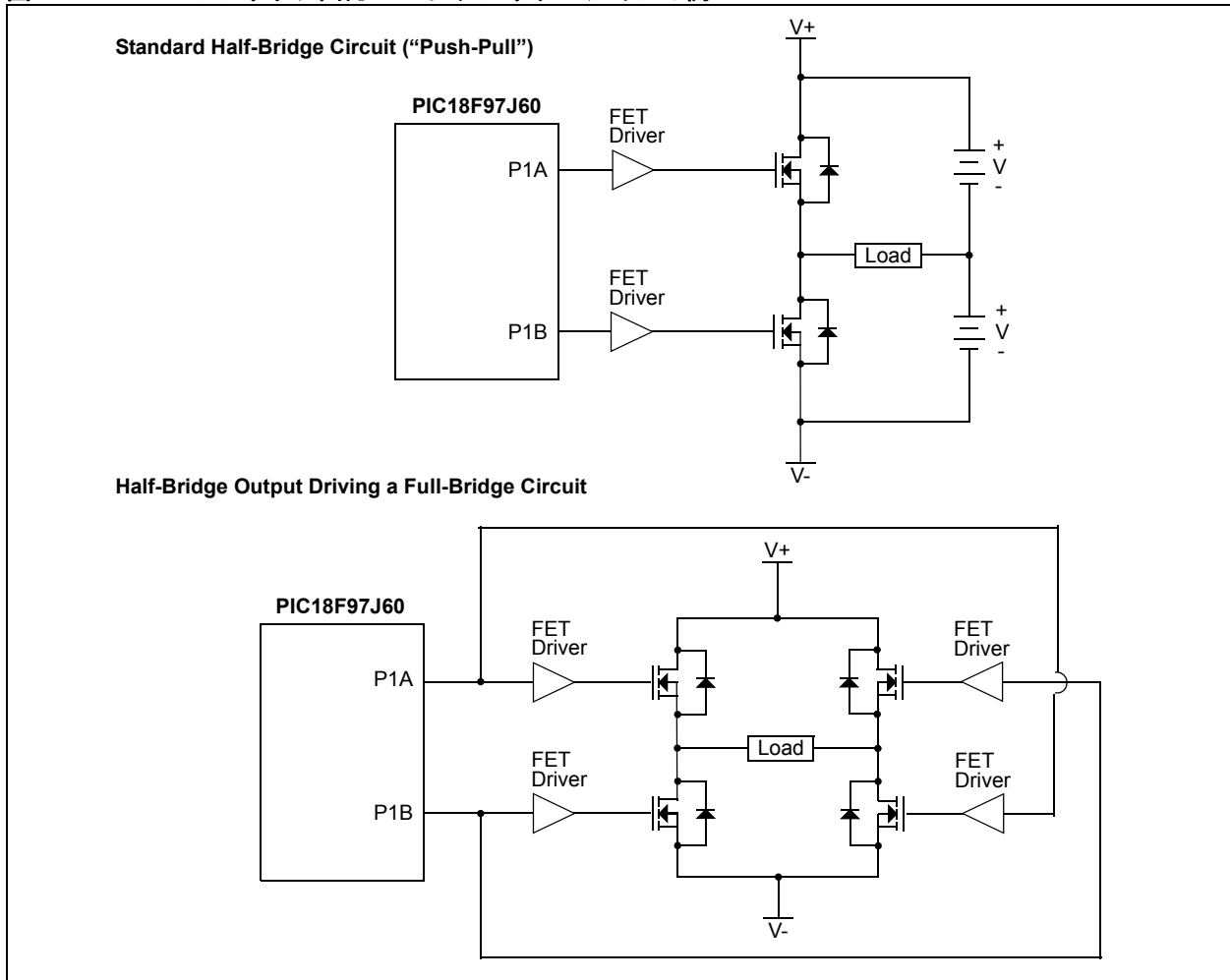


図 18-5: ハーフブリッジ出力モードアプリケーションの例



# PIC18F97J60 ファミリ

## 18.4.5 フルブリッジモード

フルブリッジ出力モードでは4本のピンを出力に使用します。しかし、同時にアクティブになるのは、このうち2つの出力だけです。順方向モードの場合、P1Aピンは常時アクティブであり、P1Dピンに変調信号が出力されます。逆方向モードでは、P1Cピンが常時アクティブであり、P1Bピンに変調信号が出力されます。

図 18-6 にこれらの状態を示します。

P1A、P1B、P1C、P1Dの出力は、表 18-1 と表 18-3 に示すポートピンのデータラッチと多重化されています。P1A、P1B、P1C、P1Dピンを出力に設定するために、対応する TRIS ビットをクリアする必要があります。

図 18-6: フルブリッジ PWM 出力

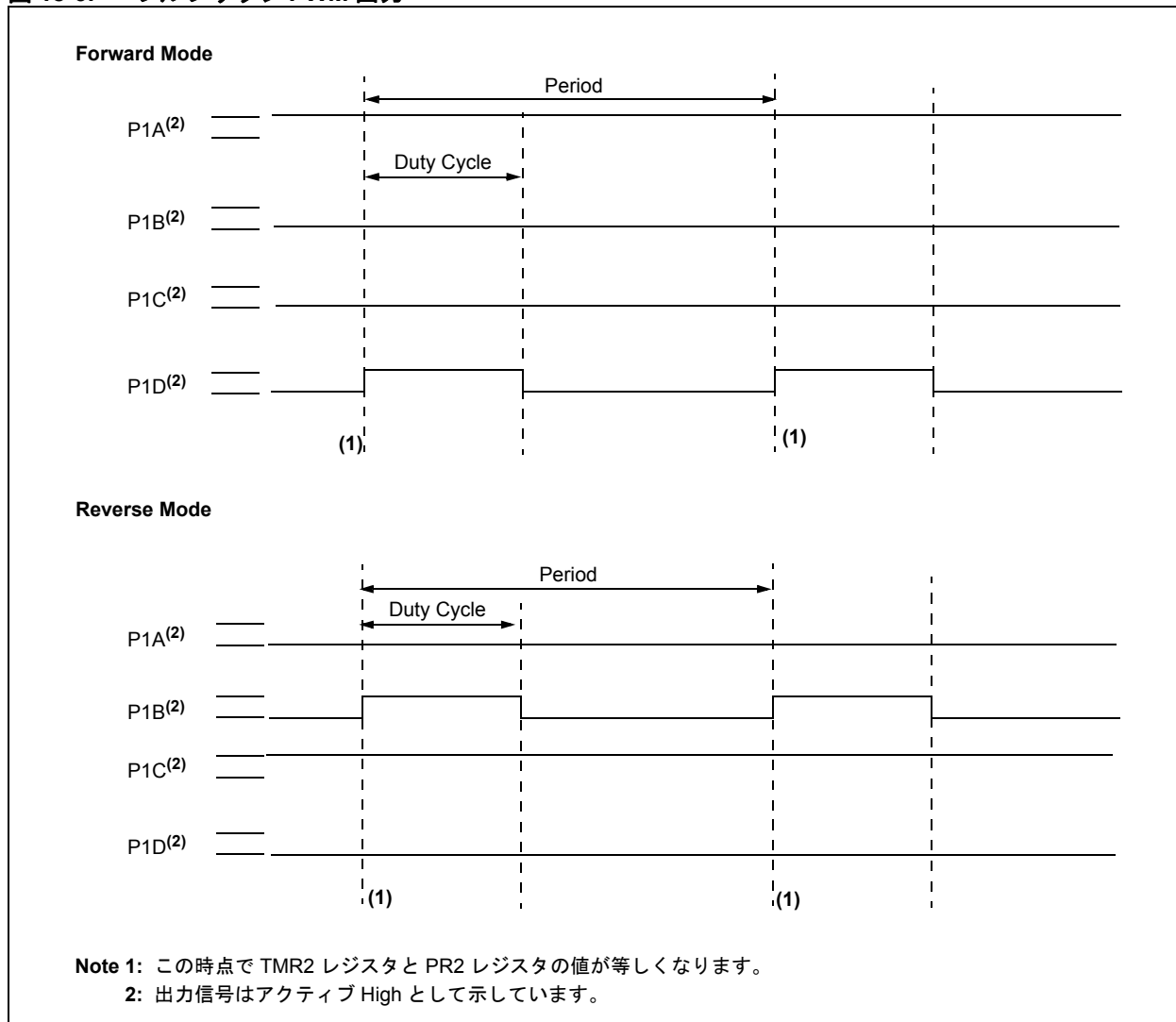
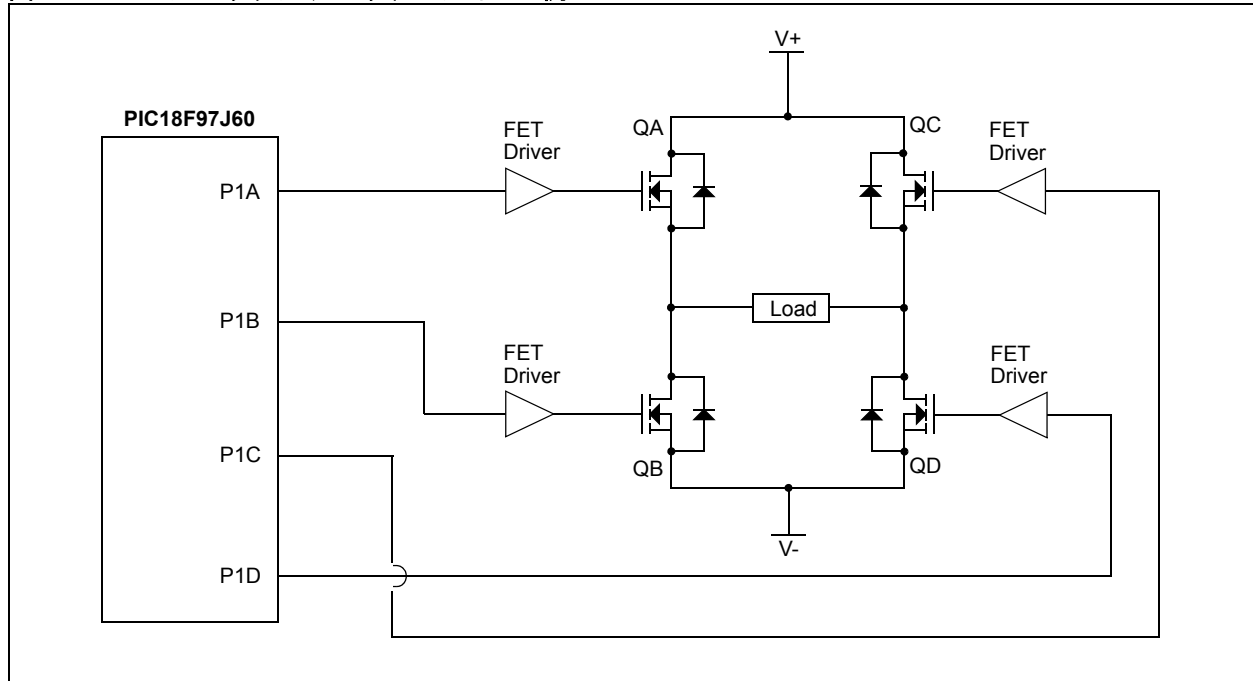




図 18-7: フルブリッジアプリケーションの例



## 18.4.5.1 フルブリッジモードにおける方向の変更

フルブリッジ出力モードでは、CCP1CON レジスタの P1M1 ビットによって、ユーザが順方向 / 逆方向を制御できます。アプリケーションファームウェアがこの方向制御ビットを変更すると、モジュールは次のPWM サイクルで方向を切り換えます。

現在の PWM 周期が完了する直前に、変調出力 (P1B と P1D) は非アクティブの状態に移行し、非変調出力 (P1A と P1C) は逆方向の駆動に切り換わります。この動作は、次の PWM 周期が始まる前、 $4 T_{osc} * (\text{Timer2 プリスケール値})$  の間に実行されます。Timer2 のプリスケール値は、T2CKPS ビット (T2CON<1:0>) の値に応じた、1、4、16 のいずれかです。非変調出力の切り換えから次の周期の始まりまでの期間、変調出力 (P1B と P1D) は非アクティブの状態を保ちます。図 18-8 にこの関係を示します。

フルブリッジ出力モードの ECCP1 モジュールではデッドバンド遅延機能を使えない事に注意してください。このモードの変調出力は常に 1 つだけであるため、通常デッドバンド遅延は不要です。しかしデッドバンド遅延が必要となり得るケースが 1 つ存在します。それは、以下の条件が 2 つとも真の場合です。

1. 出力のデューティサイクルが 100% またはそれに近い状態で PWM 出力の方向が変化した
2. パワースイッチ (パワーデバイスとドライバ回路を含む) のターンオフ時間がターンオン時間よりも長い

図 18-9 に、デューティサイクルが 100% に近い状態で、PWM の方向が順方向から逆方向に変化する例を示します。時間  $t_1$  では出力 P1A と P1D が非アクティブになり、出力 P1C がアクティブになります。この例では、パワーデバイスのターンオフ時間がターンオン時間よりも長い場合、パワーデバイスの QC と QD (図 18-7 参照) に「 $t_1$ 」の時間だけ貫通電流が流れる可能性があります。PWM の方向が逆方向から順方向に変わる場合も、パワーデバイス QA と QB に同様の現象が発生します。

アプリケーションで、高いデューティサイクルで PWM の方向を変更する必要がある場合、以下のいずれかの要件を満たす必要があります。

1. 方向を変更する前の PWM 周期の PWM デューティサイクルを小さくする
2. ターンオン時間よりもターンオフ時間の短いスイッチドライバを使う

貫通電流を防止する方法は上記以外にも考えられます。

# PIC18F97J60 ファミリ

図 18-8: PWM の方向変更

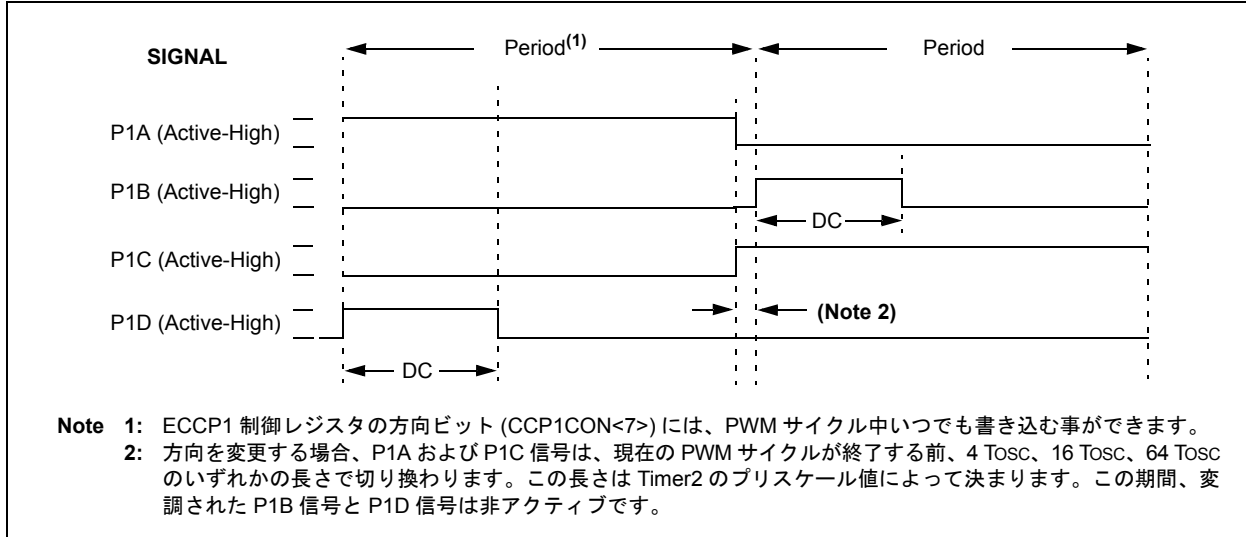
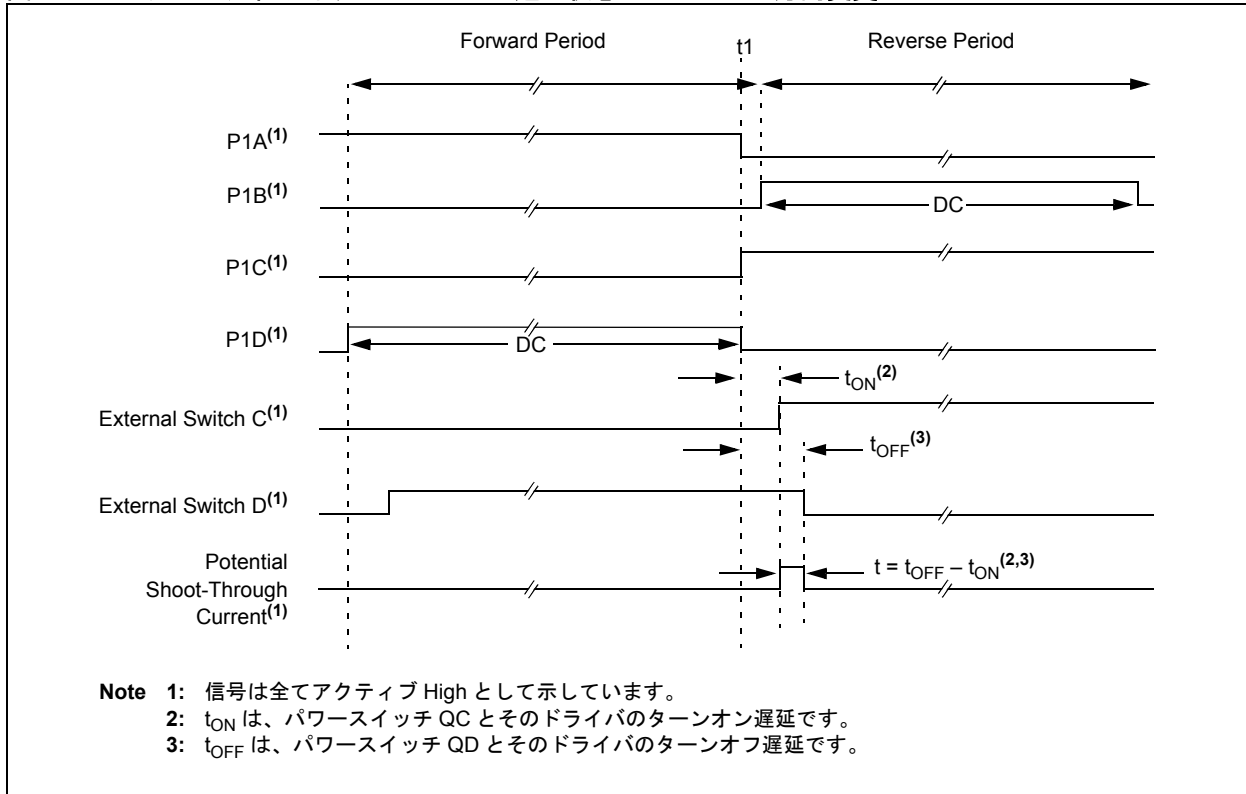


図 18-9: デューティ サイクルが 100% に近い状態での PWM の方向変更



## 18.4.6 プログラマブルなデッドバンド遅延

ハーフブリッジアプリケーションでは、常に全てのパワースイッチがPWM周波数で変調されるため、パワースイッチは通常、ターンオンよりターンオフに長い時間を必要とします。上下のパワースイッチが同時に切り換わる場合（一方がターンオン、他方がターンオフ）、一方のスイッチが完全にターンオフするまでに、短いながら両方のスイッチがONになる期間が発生する可能性があります。このわずかな期間、2つのパワースイッチを通して非常に大きな電流（貫通電流）が流れ、ブリッジの電源が短絡する場合があります。回路に損傷を与えかねないこのスイッチング時の貫通電流を防止するために、通常いずれかのパワースイッチのターンオンを遅延させ、その間にもう一方のスイッチが完全にターンオフできるようにします。

ハーフブリッジ出力モードでは、貫通電流がブリッジのパワースイッチを破壊するのを防ぐため、デジタルプログラムが可能なデッドバンド遅延を使えます。信号の非アクティブ状態からアクティブ状態への遷移を遅延させます。図 18-4 を参照してください。遅延時間は、ECCP1DEL レジスタ（レジスタ 18-2）の下位 7 ビットを使って、マイクロコントローラの命令サイクル数 (Tcy または 4 Tosc) 単位で設定します。

## 18.4.7 拡張 PWM の自動シャットダウン

ECCP1 を拡張 PWM のいずれかのモードにプログラムした場合、アクティブな出力ピンに自動シャットダウン機能を設定できます。自動シャットダウンは、シャットダウンイベント発生時に拡張 PWM の出力ピンをただちに指定されたシャットダウン状態に移行させます。

シャットダウン イベントは、2つのコンパレータ モジュールまたは FLT0 ピンのいずれか（またはこれら3つの要因の任意の組み合わせ）によって発生します。コ

ンパレータは、ブリッジ回路内で監視している電流に比例する電圧入力を監視できます。電圧がしきい値を超えるとコンパレータの状態が切り換わり、シャットダウンをトリガします。これに代わる方法として、FLT0 ピンの Low レベルのデジタル信号によってシャットダウンをトリガする事も可能です。自動シャットダウン機能は、トリガとなる要因を 1 つも選択しない事で無効にできます。自動シャットダウンのトリガに使う要因は、ECCP1AS<2:0> ビット (ECCP1AS<6:4>) によって選択します。

シャットダウンが発生すると、出力ピンは PSS1AC<1:0> および PSS1BD<1:0> ビット (ECCP1AS<3:0>) によって指定されたシャットダウン状態に非同期で移行します。各ピンペア (P1A/P1C と P1B/P1D) は、High 駆動、Low 駆動、3 ステート（駆動しない）のいずれかに設定できます。ECCP1ASE ビット (ECCP1AS<7>) も、拡張 PWM 出力をシャットダウン状態に保持するためにセットされます。

ECCP1ASE ビットは、シャットダウン イベントが発生するとハードウェアによってセットされます。自動再起動が無効の場合、ECCP1ASE ビットはシャットダウンの原因が解消された時点でファームウェアによってクリアされます。自動再起動が有効な場合、ECCP1ASE ビットは自動シャットダウンの原因が解消された時点で自動的にクリアされます。

PWM 周期の開始時に ECCP1ASE ビットがセットされた場合、その PWM 周期の全体にわたり、PWM 出力はシャットダウン状態に保たれます。ECCP1ASE ビットがクリアされると、PWM 出力は次の PWM 周期の開始時点で通常動作に戻ります。

**Note:** シャットダウン条件有効時、ECCP1ASE ビットへの書き込みは無効です。

## レジスタ 18-2: ECCP1DEL: ECCP1 デッドバンド遅延レジスタ

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
P1RSEN	P1DC6	P1DC5	P1DC4	P1DC3	P1DC2	P1DC1	P1DC0
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

bit 7      **P1RSEN:** PWM 再起動イネーブルビット  
 1 = 自動シャットダウン後、シャットダウン条件が解消されると自動的に ECCP1ASE ビットをクリアし、PWM 動作を自動的に再開する  
 0 = 自動シャットダウン後、PWM 動作を再開するにはソフトウェアで ECCP1ASE をクリアする必要がある

bit 6-0      **P1DC<6:0>:** PWM 遅延カウントビット  
 PWM 信号がアクティブに変化する設定された遷移時点から実際の遷移時点までの遅延時間である。Fosc/4 (4 \* Tosc) のサイクル数によって指定する

# PIC18F97J60 ファミリ

## レジスタ 18-3: ECCP1AS: ECCP1 自動シャットダウン コンフィグレーション レジスタ

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ECCP1ASE	ECCP1AS2	ECCP1AS1	ECCP1AS0	PSS1AC1	PSS1AC0	PSS1BD1	PSS1BD0
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

- bit 7      **ECCP1ASE:** ECCP1 自動シャットダウン イベント ステータスビット  
0 = ECCP1 出力は動作中である  
1 = シャットダウン イベントが発生し、ECCP1 出力がシャットダウン状態である
- bit 6-4      **ECCP1AS<2:0>:** ECCP1 自動シャットダウン要因選択ビット  
000 = 自動シャットダウンを無効にする  
001 = コンパレータ 1 出力  
010 = コンパレータ 2 出力  
011 = コンパレータ 1、コンパレータ 2 のいずれか  
100 = FLT0  
101 = FLT0 またはコンパレータ 1  
110 = FLT0 またはコンパレータ 2  
111 = FLT0、コンパレータ 1、コンパレータ 2 のいずれか
- bit 3-2      **PSS1AC<1:0>:** A および C ピン シャットダウン状態制御ビット  
00 = A および C ピンを「0」に駆動する  
01 = A および C ピンを「1」に駆動する  
1x = A および C ピンを 3 ステートに設定する
- bit 1-0      **PSS1BD<1:0>:** B および D ピン シャットダウン状態制御ビット  
00 = B および D ピンを「0」に駆動する  
01 = B および D ピンを「1」に駆動する  
1x = B および D ピンを 3 ステートに設定する

## 18.4.7.1 自動シャットダウンと自動再起動

自動シャットダウン機能には、シャットダウン イベントの発生後にモジュールを自動的に再起動する設定が用意されています。この設定は、ECCP1DEL レジスタの P1RSEN ビット (ECCP1DEL<7>) をセットする事で有効になります。

P1RSEN = 1 (図 18-10) のシャットダウンモードでは、シャットダウンの原因が解消されない限り、ECCP1ASE ビットはセットされたままです。シャットダウン条件が解消されると、ECCP1ASE ビットがクリアされます。P1RSEN = 0 の場合 (図 18-11)、シャットダウン条件が発生すると、ECCP1ASE ビットはファームウェアによってクリアされるまでセットされたままです。ECCP1ASE ビットがクリアされると、次の PWM 周期の開始時点から拡張 PWM 動作が再開します。

**Note:** シャットダウン条件が有効な間は、ECCP1ASE ビットへの書き込みは無効です。

自動シャットダウン要因がコンパレータの場合、P1RSEN ビットの設定に関わらず、シャットダウン条件はレベル検出となります。シャットダウンの原因が解消されない限り、ECCP1ASE ビットはクリアできません。

ECCP1ASE ビットに「1」を書き込む事で、自動シャットダウン状態に強制的に移行させます。

## 18.4.8 起動に関する注意点

ECCP1 モジュールを PWM モードで使う場合、アプリケーション ハードウェアでは PWM 出力ピンに適切な外部プルアップまたはプルダウン抵抗 (あるいは両方) を接続する必要があります。マイクロコントローラのリセット状態が解除されると、全ての I/O ピンがハイインピーダンス状態に移行します。マイクロコントローラが適切な信号レベルで I/O ピンを駆動しはじめるまで、あるいは PWM 出力をアクティブにするまで、外部回路によってパワースイッチ デバイスを OFF 状態に保つ必要があります。

CCP1M<1:0> ビット (CCP1CON<1:0>) によって、PWM 出力ピンの各ペア (P1A/P1C と P1B/P1D) の信号を、アクティブ High とアクティブ Low のどちらにするかを選択できます。PWM 出力の極性は、PWM ピンを出力に設定する前に選択しておく必要があります。PWM ピンの出力設定と同時に極性の設定を変える事は推奨できません。アプリケーション回路を損傷する恐れがあります。

PWM モジュールの初期化中は、P1A、P1B、P1C、P1D の出力ラッチが適正な状態にない可能性があります。PWM ピンを出力に設定する動作と、ECCP1 モジュールの有効化を同時に実行すると、アプリケーション回路を損傷する恐れがあります。ECCP1 モジュールを適切な出力モードで有効化し、PWM サイクルを 1 回完全に終了させてから PWM ピンを出力ピンに設定してください。PWM サイクルが完了した事は、2 サイクル目の PWM 周期が開始する時点で TMR2IF ビットがセットされる事で確認できます。

図 18-10: PWM の自動シャットダウン (P1RSEN = 1、自動再起動を有効にしている場合)

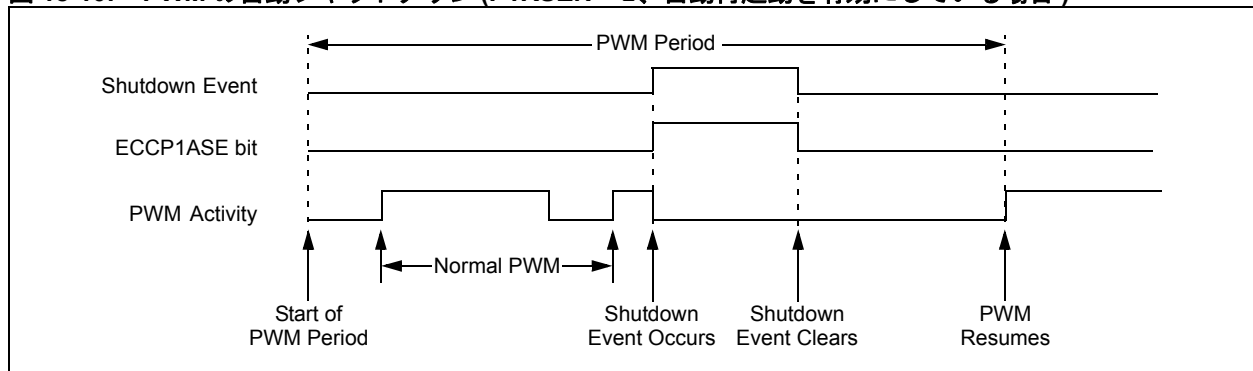
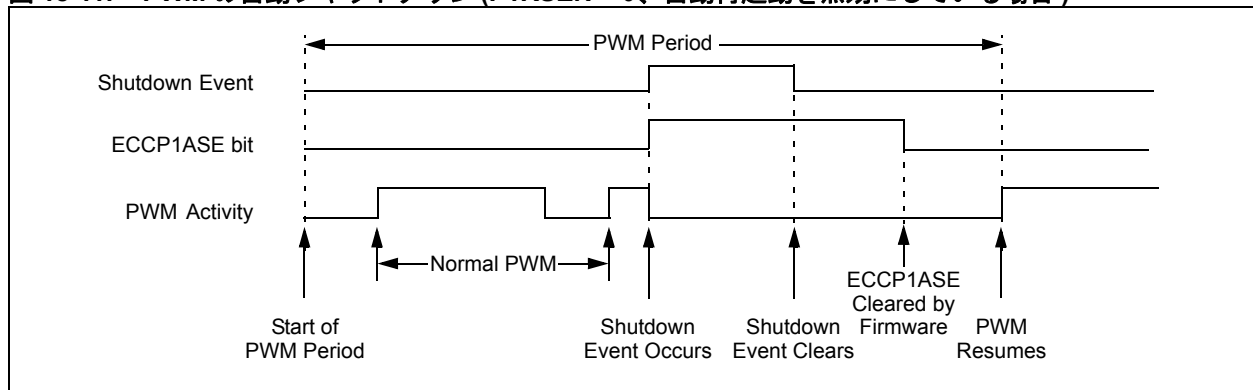


図 18-11: PWM の自動シャットダウン (P1RSEN = 0、自動再起動を無効にしている場合)



# PIC18F97J60 ファミリ

## 18.4.9 PWM 動作の設定

ECCP1 モジュールを PWM 動作に設定する手順は以下の通りです。

1. 対応する TRIS ビットをセットして、PWM ピンの P1A と P1B (使う場合は P1C と P1D も) を入力に設定する。
2. PR2 (PR4) レジスタへの読み込みにより PWM の周期を設定する。
3. CCP1CON レジスタに適切な値を読み込んで、ECCP1 モジュールに必要な PWM モードとコンフィグレーションに設定する。
  - P1M<1:0> ビットにより、使用可能な出力コンフィグレーションと方向の中から 1 つを選択する。
  - CCP1M<3:0> ビットにより、PWM 出力信号の極性を選択する。
4. CCPR1L レジスタと CCP1CON<5:4> ビットに値を読み込んで、PWM のデューティ サイクルを設定する。
5. 自動シャットダウンに関する設定：
  - 自動シャットダウンを無効にする (ECCP1ASE = 0)。
  - 自動シャットダウンの要因を設定する。
  - Run 条件を待機する。
6. ハーフブリッジ出力モードの場合、ECCP1DEL<6:0> に適切な値を読み込んで、デッドバンド遅延を設定する。
7. 自動シャットダウン動作が必要な場合、ECCP1AS レジスタに以下の通り値を読み込む。
  - ECCP1AS<2:0> ビットにより自動シャットダウン信号源を選択する。
  - PSS1AC<1:0> および PSS1BD<1:0> ビットにより PWM 出力ピンのシャットダウン状態を選択する。
  - ECCP1ASE ビット (ECCP1AS<7>) をセットする。

8. 自動再起動の動作が必要な場合、P1RSEN ビット (ECCP1DEL<7>) をセットする。
9. TMR2 (TMR4) を設定、起動する。
  - TMRxIF ビット (Timer2 の場合 PIR1<1>、Timer4 の場合 PIR3<3>) をクリアして TMRx 割り込みフラグビットをクリアする。
  - TxCKPS ビット (Timer2 の場合 T2CON<1:0>、Timer4 の場合 T4CON<1:0>) に値を読み込んで TMRx のプリスケール値を設定する。
  - TMRxON ビット (Timer2 の場合 T2CON<2>、Timer4 の場合 T4CON<2>) をセットして Timer2 (または Timer4) を有効にする。
10. 新しい PWM サイクルの開始後に PWM 出力を有効にする。
  - TMR2 (TMR4) がオーバーフローする (TMRxIF ビットがセットされる) まで待機する。
  - 対応する TRIS ビットをクリアして ECCP1/P1A、P1B、P1C、P1D ピン出力のいずれか、または全てを有効にする。
  - ECCP1ASE ビット (ECCP1AS<7>) をクリアする。

## 18.4.10 リセットの影響

パワーオン リセットおよびそれ以降のリセットは、いずれも全ポートを強制的に入力モードに設定し、CCPx/ECCPx レジスタをリセット状態に移行させます。

これによって、拡張 CCPx モジュールは標準 CCPx モジュールと互換性のある状態にリセットされます。

# PIC18F97J60 ファミリ

表 18-5: ECCPx モジュールと Timer1 ~ 4 に関連するレジスタ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	リセット 値の記載 ページ
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	69
RCON	IPEN	-	$\overline{CM}$	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$	70
PIR1	PSPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	71
PIE1	PSPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	71
IPR1	PSPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	71
PIR2	OSCFIF	CMIF	ETHIF	r	BCL1IF	-	TMR3IF	CCP2IF	71
PIE2	OSCFIE	CMIE	ETHIE	r	BCL1IE	-	TMR3IE	CCP2IE	71
IPR2	OSCFIP	CMIP	ETHIP	r	BCL1IP	-	TMR3IP	CCP2IP	71
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	71
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	71
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	71
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	71
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	71
TRISD <sup>(1)</sup>	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	71
TRISE	TRISE7 <sup>(2)</sup>	TRISE6 <sup>(2)</sup>	TRISE5	TRISE4	TRISE3	TRISE2	TRISE1	TRISE0	71
TRISG	TRISG7	TRISG6	TRISG5	TRISG4	TRISG3 <sup>(2)</sup>	TRISG2	TRISG1	TRISG0 <sup>(2)</sup>	71
TRISH <sup>(2)</sup>	TRISH7	TRISH6	TRISH5	TRISH4	TRISH3	TRISH2	TRISH1	TRISH0	71
TMR1L	Timer1 レジスタ下位バイト								70
TMR1H	Timer1 レジスタ上位バイト								70
T1CON	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	TMR1CS	TMR1ON	70
TMR2	Timer2 レジスタ								70
T2CON	-	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	70
PR2	Timer2 周期レジスタ								70
TMR3L	Timer3 レジスタ下位バイト								70
TMR3H	Timer3 レジスタ上位バイト								70
T3CON	RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	$\overline{T3SYNC}$	TMR3CS	TMR3ON	71
TMR4	Timer4 レジスタ								72
T4CON	-	T4OUTPS3	T4OUTPS2	T4OUTPS1	T4OUTPS0	TMR4ON	T4CKPS1	T4CKPS0	72
PR4	Timer4 周期レジスタ								72
CCPRxL <sup>(3)</sup>	キャプチャ/コンペア/PWM レジスタ x 下位バイト								70
CCPRxH <sup>(3)</sup>	キャプチャ/コンペア/PWM レジスタ x 上位バイト								70
CCPxCON <sup>(3)</sup>	PxM1	PxM0	DCxB1	DCxB0	CCPxM3	CCPxM2	CCPxM1	CCPxM0	70
ECCPxAS <sup>(3)</sup>	ECCPxASE	ECCPxAS2	ECCPxAS1	ECCPxAS0	PSSxAC1	PSSxAC0	PSSxBD1	PSSxBD0	70, 73
ECCPxDEL <sup>(3)</sup>	PxRSEN	PxDC6	PxDC5	PxDC4	PxDC3	PxDC2	PxDC1	PxDC0	72

凡例: - = 未実装ビット、「0」として読み出されます。r = 予約済みです。網掛けの部分は ECCPx 動作には使いません。

Note 1: 64 ピンデバイスにのみ適用されます。

2: レジスタおよび特定のビットは 64 ピンデバイスには実装されていません。

3: 全ての拡張 CCPx モジュールに対して使う、同一レジスタ全体を指す総称名です。「x」によって個別のモジュール (ECCP1、ECCP2、ECCP3) を区別します。総称名が同じレジスタ間では、ビット割り当てとリセット値も同じです。

# PIC18F97J60 ファミリ

---

NOTES:



## 19.0 Ethernet モジュール

PIC18F97J60 ファミリは、全て Ethernet コントローラ モジュールを内蔵しています。このモジュールは MAC (Media Access Control) モジュールと物理層 (PHY) トランシーバモジュールの両方を完全に実装した包括的なコネクティビティ ソリューションです。2つのパルストランスとわずかな受動部品を追加するだけで、マイクロコントローラを Ethernet ネットワークに直接接続できます。

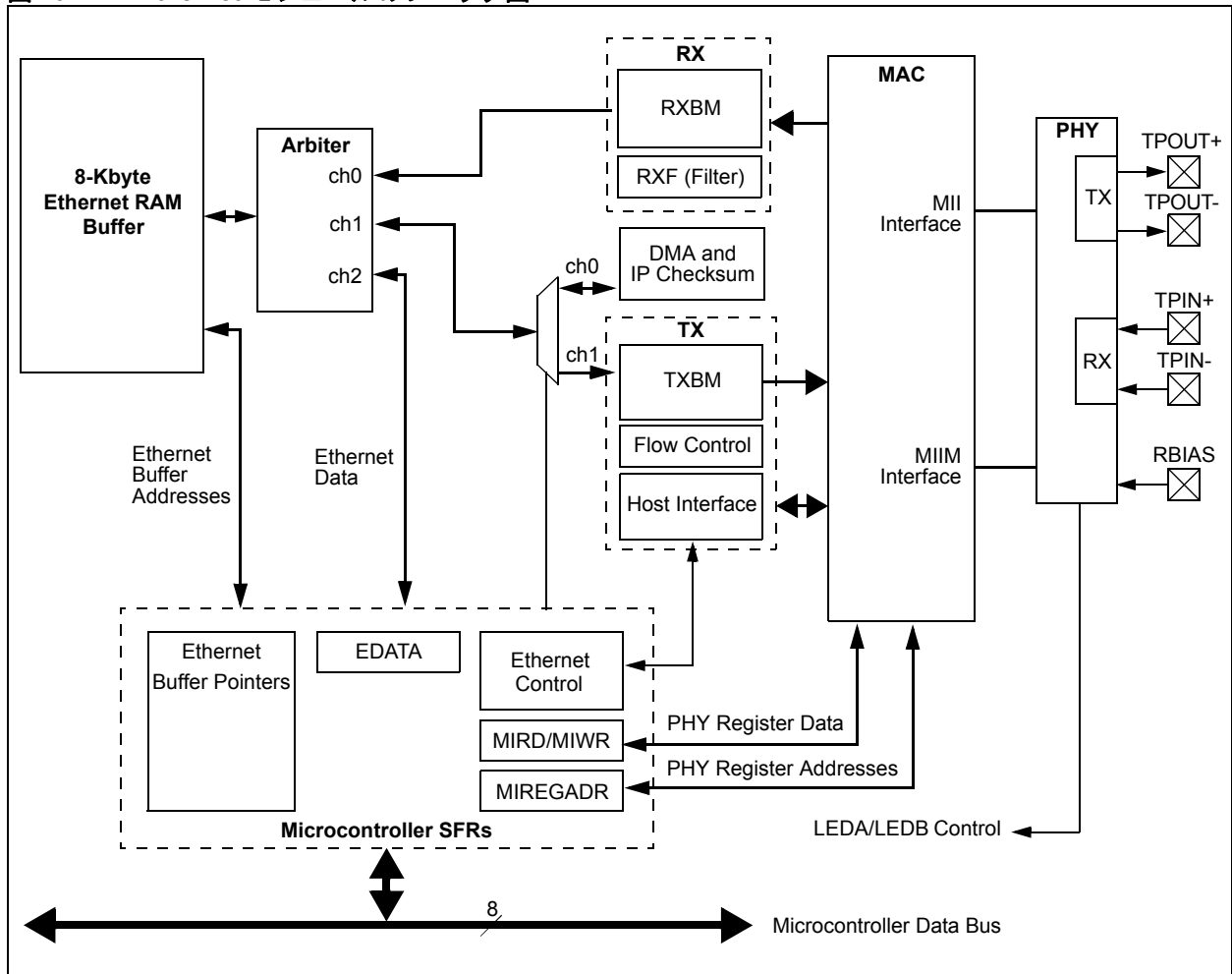
この Ethernet モジュールは、10-BaseT 接続の IEEE 802.3 規格に完全準拠しています。受信パケットを制限するフィルタも各種サポートしています。また、高スループットでのデータコピーやハードウェア アクセラレーションによる IP チェックサム計算を行う DMA モジュールも内蔵しています。リンクとネットワーク アクティビティを表示する 2つの LED 出力も備えています。

図 19-1 に Ethernet モジュールの概略ブロック図を示します。

Ethernet モジュールは主に以下の 5つの機能ブロックで構成されています。

1. PHY トランシーバモジュール：ツイストペア インターフェイスに現れるアナログデータをエンコード / デコードし、ネットワークとの間で送受信を行います。
2. MAC モジュール：IEEE 802.3 準拠の MAC ロジックを実装し、PHY を制御する MIIM (Media Independent Interface Management) を備えています。
3. 独立した 8 KB の RAM バッファ：受信済みパケットと送信待ちパケットを格納します。
4. アービタ：マイクロコントローラ コア、DMA、送信ブロック、受信ブロックから RAM バッファへのアクセス要求を制御します。
5. レジスタ インターフェイス：Ethernet モジュールとマイクロコントローラのSFRの間でコマンドと内部ステータス信号のインタプリタとして機能します。

図 19-1: Ethernet モジュールのブロック図



# PIC18F97J60 ファミリ

## 19.1 物理インターフェイスと外部接続

### 19.1.1 信号および電源インターフェイス

PIC18F97J60 ファミリは、全て Ethernet モジュール専用の 4 ピン信号インターフェイスを実装しています。これらのピンには他の機能が多重化されないため、ピンの競合が発生する心配はありません。これらのピンの機能は以下の通りです。

- TPIN+: 差動 (+) ツイストペア入力
- TPIN-: 差動 (-) ツイストペア入力
- TPOUT+: 差動 (+) ツイストペア出力
- TPOUT-: 差動 (-) ツイストペア出力

外部 Ethernet PHY との間でデジタル Ethernet データをやりとりする機能はありません。

信号接続以外に、Ethernet モジュールには PHY モジュール専用の独立した電源接続とグランド接続を備えています。これらの接続は、レシーバ用 (VDDR<sub>X</sub>、VSSR<sub>X</sub>)、トランスミッタ用 (VDDT<sub>X</sub>、VSS<sub>T</sub><sub>X</sub>)、トランスミッタの内部 PLL 用 (VDDPLL、VSSPLL) で別々です。電圧の要件はマイクロコントローラの VDD と VSS と共通ですが、これらのピンは内部接続されていません。Ethernet モジュールを正しく動作させるには、電源電圧とグランドをこれらのピンに接続する必要があります。マイクロコントローラの電源およびグランドピンは全て外部で共通の電源またはグランドノードに接続する必要があります。この時、マイクロコントローラと Ethernet モジュールの VDD ピンの間にはインダクタやその他のフィルタ部品を取り付けないようにしてください。

独立した電源接続以外に、PHY モジュールには専用のバイアス電流入力ピン (RBIAS) もあります。トランスミッタを正しく動作させるには、外付け抵抗からのバイアス電流を RBIAS に印加する必要があります。

### 19.1.2 LED の設定

PHY モジュールは標準の Ethernet インジケータ (LEDA と LEDB) を駆動するための独立した出力を備えています。LEDA 出力と LEDB 出力は、それぞれ PORTA ピンの RA0/RA1 と多重化されています。これらのピンを LED 出力として使うには、コンフィグレーションビットの ETHLED (レジスタ 25-6、CONFIG3H<2>) をセットします。LED 出力として設定すると、RA0/LEDA および RA1/LEDB ピンからは LED を直接駆動できる十分な電流 (最大 25 mA) が得られます。これらのピンは常に LED に電流を供給するように設定しておく必要があります。TRISA<1:0> をクリアして、これらのピンを出力に設定します。

これらの LED はリンクステータス、RX/TX アクティビティ等を自動的に表示するように設定できます。パケット 1 個の送信等、短時間のイベントに対する LED の点灯時間を延長してユーザが認識できるようにするストレッチ機能を備えています。これらの機能は PHLCON レジスタ (レジスタ 19-13) で制御します。表 19-1 に、点灯ストレッチ時間の代表値を示します。

表 19-1: LED 点灯ストレッチ時間の長さ

ストレッチ時間	代表値 (ms)
TNSTRCH (ノーマル)	40
TMSTRCH (ミディアム)	70
TLSTRCH (ロング)	140

### 19.1.3 オシレータの要件

Ethernet モジュールは 25 MHz で動作するように設計されています。この周波数は、マイクロコントローラのプライマリクロック (OSC1 および OSC2 ピンに接続した 25 MHz 水晶振動子または OSC1 ピンに接続した外部クロック源) から供給します。これ以外のクロック源から Ethernet モジュールにクロックを供給することはできません。

このため、Ethernet モジュール有効時はプライマリオシレータをクロック源とした動作 (PRI\_RUN または PRI\_IDLE モード) のみ可能です。他の電力管理モードを使う場合、Ethernet モジュールを無効にする必要があります。

#### 19.1.3.1 起動タイマ

Ethernet モジュールには起動タイマがあります。これはマイクロコントローラの OST (オシレータ起動タイマ) とは別のもので、PHY モジュールの PLL が安定してから Ethernet モジュールの動作を開始するためのものです。Ethernet モジュールイネーブルビットの ETHEN (ECON2<5>) をクリアすると、PHYRDY ステータスビット (ESTAT<0>) がクリアされます。ETHEN ビットをセットすると、PHY 起動タイマがカウントを開始します。1 ms 後にタイマのカウントが終了すると、PHYRDY ビットが自動的にセットされます。

ETHEN ビットをセットして Ethernet モジュールを有効にしたら、ソフトウェアで PHYRDY ビットを常時ポーリングして、通常の Ethernet 動作を開始できるタイミングを待機します。

## 19.1.4 トランス、終端、その他の外付け部品

Ethernet モジュールは外付け部品を必要とします。図 19-2 に、これらの部品の接続方法を示します。

PHY モジュール内部のアナログ回路では、RBIAS とグラウンドの間に外部抵抗 (2.26 kΩ) を接続する必要があります。この抵抗は、TPOUT+/- の信号振幅に影響します。ピンにノイズが静電結合してトランスミッタの動作に影響するのを防ぐため、信号トレースを隣接させずに抵抗をなるべくチップの近くに配置する必要があります。抵抗は表面実装タイプを推奨します。

TPIN+/TPIN- ピンと TPOUT+/TPOUT- ピンには、Ethernet (10/100 または 10/100/1000) の動作要件を満たした 1:1 センタータップ (CT) パルス変圧器が必要です。Ethernet モジュールを有効にすると、2 本の TPOUT ピンにシンク電流が常時流れます。PHY が送信動作中は、TPOUT+ と TPOUT- の相対的なシンク電流量が変化する事によって Ethernet ケーブルに差動電圧が発生します。

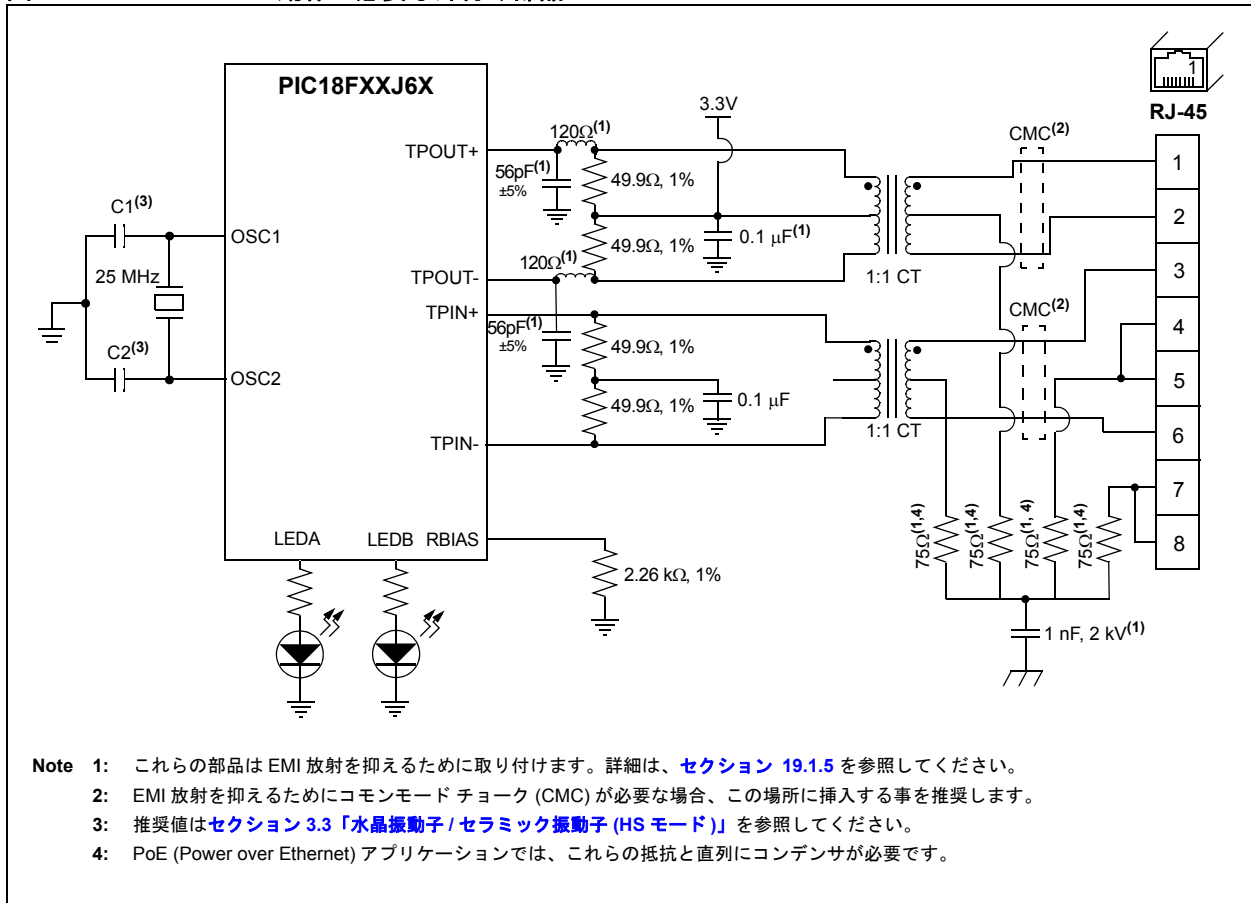
インターフェイスの PHY 側 (すなわちマイクロコントローラの TPOUT ピンと Ethernet トランスの間) に共通モードチョークを使う事は推奨しません。EMI 放射を抑えるために共通モードチョークを使う場合、

Ethernet トランスと RJ-45 コネクタのピン 1、2 の間に配置する必要があります。Ethernet トランス モジュールの多くは、パッケージ内に共通モードチョークを内蔵しています。静電電圧からの保護と IEEE 802.3 の絶縁要件への適合のため、トランスは表 28-28 に記載した絶縁定格を満たす必要があります (トランスの要求仕様は **セクション 28.5 「Ethernet の仕様と要件」** 参照)。送受信インターフェイスには信号反射を最小限に抑えるために 2 つの抵抗と 1 つのコンデンサを使い、伝送ラインを適切に終端処理する必要があります。

電源ピンは全て外部で共通の電源に接続する必要があります。同様に、グラウンド基準も全て共通のグラウンドノードに外部で接続する必要があります。VDD と VSS の各ピンペアには、なるべく近くに 0.1 μF のセラミックバイパスコンデンサを接続します。

ツイストペア インターフェイスの動作には比較的大きな電流が必要なため、いずれの配線もなるべく短くします。抵抗損失を抑えるため、電源ラインは適切な配線幅とします。差動データラインを短くできない場合、特性インピーダンスが 100 Ω となるように配線します。

図 19-2: Ethernet の動作に必要な外付け部品



# PIC18F97J60 ファミリ

## 19.1.5 EMI 放射に関する注意事項

ほとんどの国では、特定の周波数で環境に放出される電磁エネルギー量を規制する、意図しないEMIまたはEMC放射の限度値が規定されています。通常、Ethernet アプリケーションには無線周波数の意図的な放射源は含まれません。しかし、長い Ethernet ケーブルを接続すると高周波ノイズが比較的容易に放射されるため、これらの規制に違反する事も珍しくありません。長いケーブルは、意図せずアンテナとして作用する事があります。

PIC18F97J60 ファミリの Ethernet モジュールの送信エンジンは、25 MHz の Ethernet ベースクロックを PHY モジュールに内蔵の PLL を使って内部で高周波数に通信して動作します。この高周波数を使って、TPOUT+ および TPOUT- ピンの微小なシンク電流を ON/OFF にしています。この電流モード駆動方式を利用して PHY はアナログ信号に似た Ethernet TX 波形を生成しており、ほとんどのエネルギー スペクトルは 20 MHz 以下に集中しています。

しかし、この波形の合成に使っている高周波数は振幅が小さいとはいえ、アプリケーション回路によっては回路の外部に規制値を超える放射が発生する事があります。Ethernet モジュールが原因のこのような放射は、通常 200 MHz (場合によっては 400 MHz) です。

こうした問題を最小限に抑えるため、[図 19-2](#) に示すように TPOUT+ ピンと TPOUT- ピンに LC ローパスフィルタを使う事を推奨します。

この回路では、120  $\Omega$  のフェライトビーズと 56 pF $\pm$ 5% コンデンサを使って、カットオフ周波数が 20~200 MHz の間にあるローパスフィルタを形成しています。10Base-T Ethernet シグナリングに必要なスペクトル帯域幅は約 20 MHz であるため、このようなフィルタによる Ethernet 信号の歪みは最小限に抑えられます。一方、PHY によって発生する 200 MHz または 400 MHz のノイズが数 dB 低減されるため、規制値を超える放射を防ぐ事ができます。この回路では、飽和電流値が 100 mA 以上のフェライトビーズを使う必要があります。

EMI 規制が特に厳しい地域では、Ethernet ケーブルからの意図しないコモンモード信号の放射を最小限に抑えるため、Ethernet トランスの選択にはさらに注意が必要です。コモンモード除去比(CMRR)の高い Ethernet トランスを推奨します。

通常、CMRR は各種テスト周波数における値 (単位は -dB) がトランス製造者のデータシートに記載されています。特に反証となるテストデータがない限り、PIC18F97J60 ファミリの Ethernet モジュールには高周波数で CMRR (-dB) の絶対値がより大きいトランスが適しています。例えば、100 MHz で -40 dB のトランスと -33 dB のトランスでは、30 MHz での性能が同等または後者の方が優れていても、前者のトランスを推奨します。

多くの場合、TX パスにセンタータップ インダクタまたは自動トランスを使った「5 コア」のトランスを使う事も EMI 放射の抑制に有効です。

## 19.1.6 RX 極性の自動検出と反転

10Base-T Ethernet シグナリングは、Ethernet ケーブル上では差動マンチェスタ符号化方式を利用したデータストリームとして実行されます。このシグナリングには極性があるため、Ethernet ケーブルの RX+ Ethernet 信号を TPIN+ ピンに、RX- Ethernet 信号を TPIN- ピンに接続する必要があります。RX+ と TPIN-、RX- と TPIN+ を (Ethernet 絶縁トランスを介して) 接続した場合でも、PIC18F97J60 ファミリの Ethernet モジュールとリモートパートナーのリンクは正しく確立されませんが、受信データは極性不一致で全て破損するため、PHY はこれらをケーブル上のノイズと見なし内部で破棄します。

より高速の 100Base-TX および 1000Base-T 規格では、異なるシグナリング方式を使います。ケーブル上で、それぞれ MLT3 (Multi-Level Transition 3) および PAM5 (Five-Level Pulse Amplitude Modulation) 符号化を使います。これらの符号化方式には極性がありません。従って、差動ケーブルを入れ替えても Ethernet コントローラはリモートノードと問題なく通信できます。

ごくわずかですが、最近のサードパーティ製の 10/100 および 10/100/1000 Ethernet デバイス (スイッチ、ルータ、エンドデバイス) の中には、TX+/TX- 信号を RJ-45 Ethernet ジャックの誤ったピンに接続したものが 있습니다。これらのデバイスは IEEE 規格 802.3 に準拠していません。しかし、100Base-TX と 1000Base-T は極性が間違っても通信可能なため、こうした極性エラーに気付かず製品をリリースしているサードパーティベンダもあります。

このような状況から、PIC18F97J60 ファミリの現在のリビジョンの Ethernet コントローラはごく一部のサードパーティ製 Ethernet デバイスとは互換性がありません。こうしたパートナーとも PIC18FXXJXX はリンクを確立し、PIC18FXXJXX にパケットが送信されると PHY RX アクティビティ LED は点灯します (LED 有効時)。ただし極性が正しくない場合、パケットを正しく受信して Ethernet SRAM バッファに書き込む事はできません。こうした問題を防ぎ、サードパーティ製デバイスとの相互接続性を最大限に確保するために、PIC18F97J60 ファミリのアプリケーションに RX 極性反転回路を外付けする事ができます。[図 19-3](#) に、バススイッチを使って RX 信号の極性を反転する回路を示します。

[図 19-3](#) では、汎用出力ピンを使って RX 信号の極性を選択しています。セレクトラインが Low の場合、スイッチの A ポートが B0 ポートと接続し、B1 ポートは切断された状態です。この時、TPIN+ ピンは RJ-45 ジャックのピン 3 に接続され、TPIN- ピンはピン 6 に接続されます。これらの接続は、IEEE 規格 802.3 で規定された極性に適合しています。

セレクトラインが High の場合、スイッチの A ポートが B1 ポートと接続し、B0 ポートは切断された状態です。これにより RX の極性が反転し、TPIN+ ピンは RJ-45 ジャックのピン 6 に接続され、TPIN- ピンはピン 3 に接続されます。このように極性を反転する事で、リモートのリンクパートナーまたは途中のケーブルで生じた極性の誤りが補正されます。

# PIC18F97J60 ファミリ

MCU 側では、接続したリンクパートナーに合わせてセレクトラインの状態を適切なレベルにトグルするようソフトウェアを作成しておく必要があります。これは、パケットの正常な受信イベントを監視すると同時に、低周波数 (<5 Hz) でセレクトラインを周期的にトグルする事で達成できます。正しい極性を検出したらセレクトラインのトグルを停止し、Ethernet リンクが切断されるまで同じ状態を維持します。

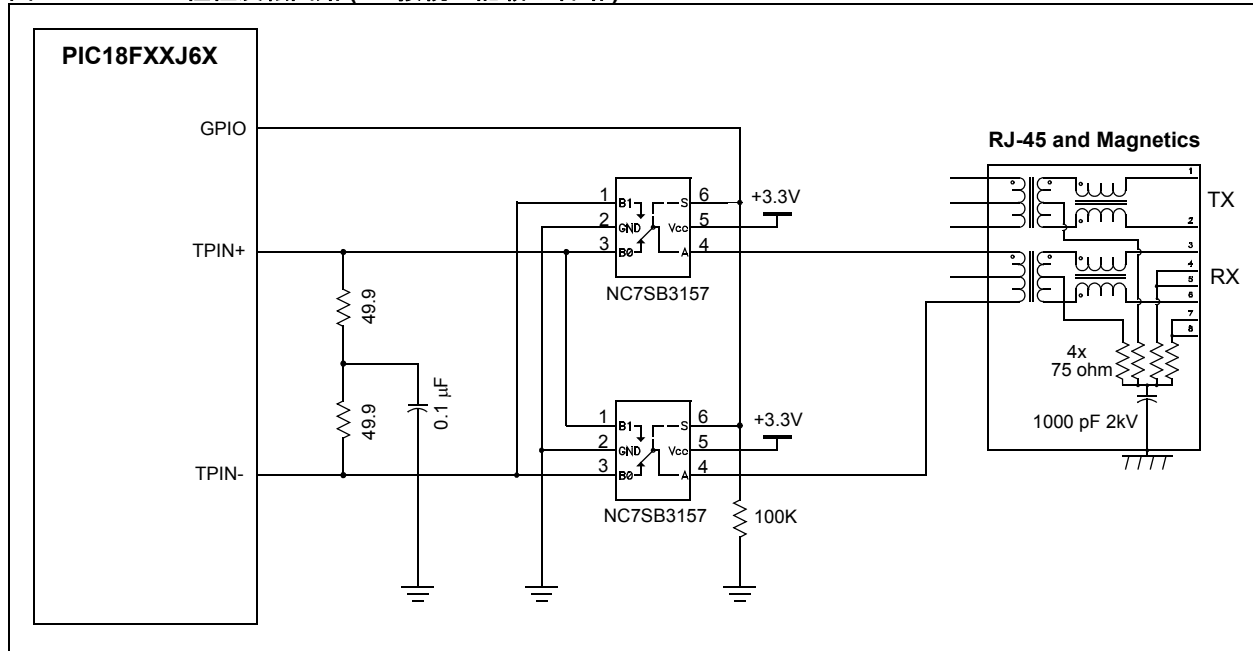
このように低周波数でセレクトラインを周期的にトグルしてパケットを受信するのではなく、パケットを送信してそのレスポンスによって短時間で正しい極性を判定できる場合もあります。この方法を使うには、ネットワークから応答パケットが返されるプロトコルが必

要です。多くのネットワークでは、DHCP (Dynamic Host Configuration Protocol) のディスカバリ パケットを使って TPIN の極性を素早く検出できます。

バススイッチを選択する際は、Ethernet 信号を歪ませずに通過できるものを選ぶように注意してください。TPIN±ピンの弱バイアスにより、RX コモンモード電圧は約  $V_{DD}/2$  まで上昇し、Ethernet RX 波形はこのコモンモード電圧に最大  $\pm 1.4$  V を加えた大きさです。従って、スイッチには最低でも 3.05 V の信号を通過できる能力が要求されます。

また、信号損失とインピーダンス不連続による RX 信号への影響を最小限に抑えるため、バススイッチは低キャパシタンスのものを選んでください。スイッチの -3 dB 帯域幅は 20 MHz を大きく超えている必要があります。

図 19-3: RX 極性反転回路 (TX 接続の記載は省略)



# PIC18F97J60 ファミリ

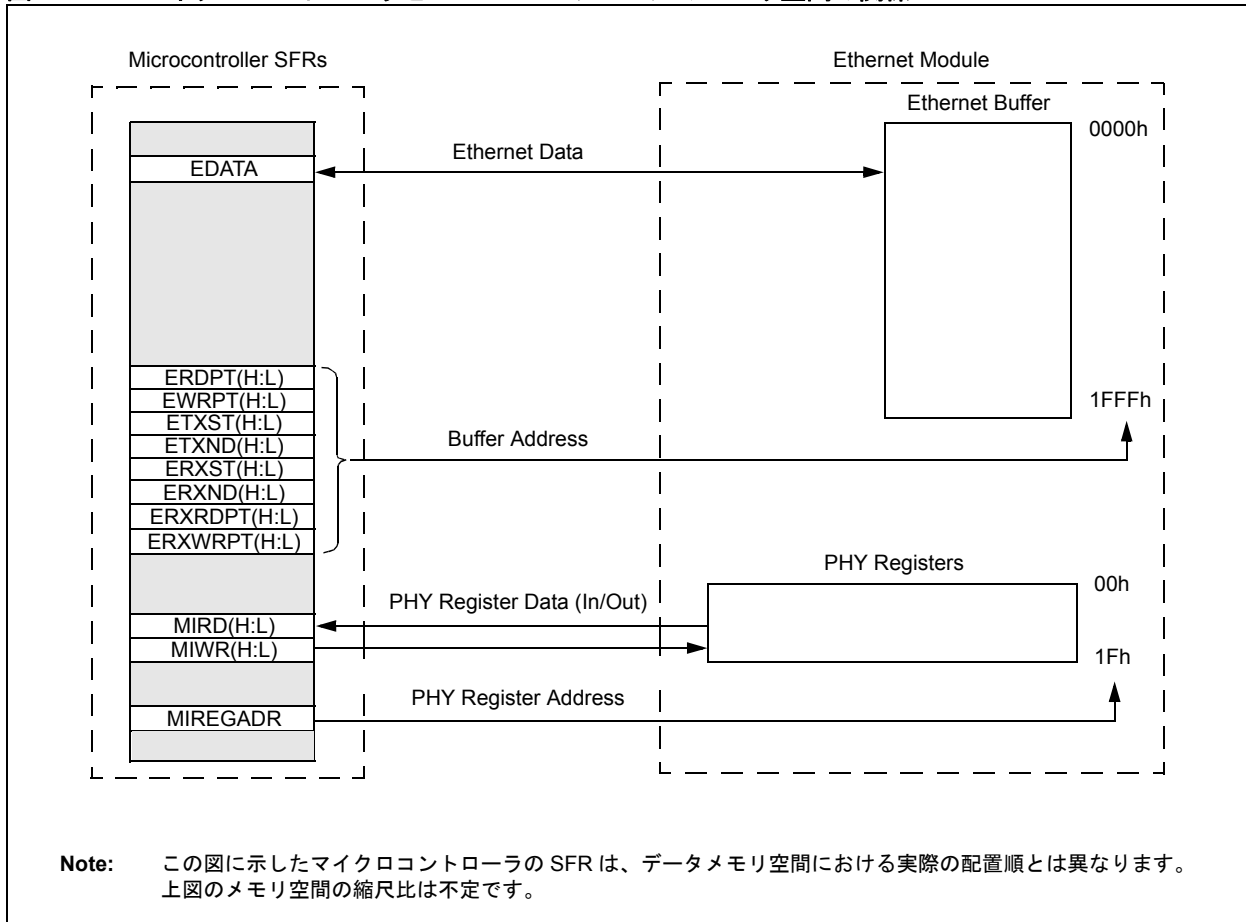
## 19.2 Ethernet バッファとレジスタ空間

Ethernet モジュールの動作には、3 つの独立したメモリ空間を使います。

- Ethernet RAM バッファ: 受信したパケットデータと送信前のパケットデータを格納します。
- 8 ビット幅の特殊機能レジスタ (SFR): モジュールの制御、Ethernet モジュールとマイクロコントローラコア間での双方向のデータ転送に使います。
- 16 ビットの PHY レジスタ: PHY の制御とステータス報告専用です。

Ethernet バッファと PHY 制御レジスタは Ethernet モジュールに完全に内包されており、マイクロコントローラからは直接アクセスできません。Ethernet モジュールとマイクロコントローラ間でデータを転送するには、マイクロコントローラの SFR 空間にマッピングされたポインタレジスタとバッファを使います。図 19-4 に、SFR と Ethernet モジュールのメモリ空間の関係を示します。

図 19-4: マイクロコントローラと Ethernet モジュールのメモリ空間の関係



## 19.2.1 Ethernet バッファと バッファポインタ レジスタ

Ethernet バッファには、Ethernet コントローラが使う送信メモリと受信メモリが含まれます。バッファ全体は 8 KB で、送信バッファ空間と受信バッファ空間に分割されます。送信メモリと受信メモリのサイズと位置は、Ethernet SFR 空間のポインタを使って定義します。図 19-5 に、メモリ空間の構成とポインタの関係を示します。

このバッファには、Ethernet モジュールが有効か無効かに関わらず、EDATA と Ethernet ポインタ SFR を使っていつでもアクセスできます。このため、Ethernet 通信が不要なアプリケーションで大容量 RAM が必要な場合、このバッファを使えます。この場合、Ethernet モジュールを無効にすると消費電力を低減でき、Ethernet バッファへのアクセスは可能のままです。

### 19.2.1.1 バッファの読み出しと書き込み

Ethernet バッファの内容は、EDATA レジスタを使ってアクセスします。EDATA レジスタは、マイクロコントローラのデータバスからバッファへのウィンドウとして機能します。このウィンドウの位置は、実行する動作の種類によって ERDPT または EWRPT ポインタによって決定します。例えば、EDATA に書き込みを実行すると、Ethernet バッファ内の EWRPT レジスタペアが現在示しているアドレスに書き込みが実行されます。同様に、EDATA の内容を別のレジスタに移動すると、実際には ERDPT ポインタが現在示しているアドレスのバッファ内容が移動されます。

AUTOINC ビット (ECON2<7>) をセットしている場合、読み出しまたは書き込みを 1 回実行するたびに、読み出しまたは書き込みポインタが 1 アドレス分インクリメントします。このため、読み書きを実行するたびにポインタを更新する必要がなく、複数の操作を連続実行する場合の手間が省けます。既定値では、AUTOINC ビットはセットされています。

受信バッファから連続的に読み出しを行うと、受信バッファの最後でラップアラウンド条件が発生します。ERXND ポインタに書き込まれたアドレスから EDATA を読み出すと、ERDPT レジスタがインクリメ

ントして ERXST ポインタ内の値になります。一方、書き込みバッファの場合は自動でラップアラウンドしません。

設計上、Ethernet メモリバッファは EDATA をオペランドと結果データ格納先の両方として使う命令をサポートしません。この制約に違反した場合、読み出しまたは書き込み結果が不正になります。また、プロセッサコアが Read-Modify-Write アーキテクチャを採用しているため、AUTOINC がセットされた状態でシングルサイクル命令を実行すると、EDATA レジスタに書き込みが行われ、ERDPT レジスタが自動的にインクリメントされてしまいます。2 サイクル命令の MOVFF、MOVSF、MOVSS 命令を使って EDATA に書き込みを実行すると、読み出しポインタは影響を受けません。以下の Note に例を示します。

**Note:** 1つの命令で EDATA SFR レジスタに対して読み出しと書き込みの両方を実行すると、不正な演算結果となります。

サポートされない例:

```
INCF    EDATA, F
XORWF  EDATA, F
MOVFF  EDATA, EDATA
MOVFF  INDF0, EDATA; (FSR0 = F61h)
```

1 回の読み出しまたは書き込みのみを実行する命令は使えます。

サポートされる例:

```
INCF    EDATA, W
MOVFF  EDATA, W
MOVFF  INDF0, EDATA; (FSR0 != F61h)
```

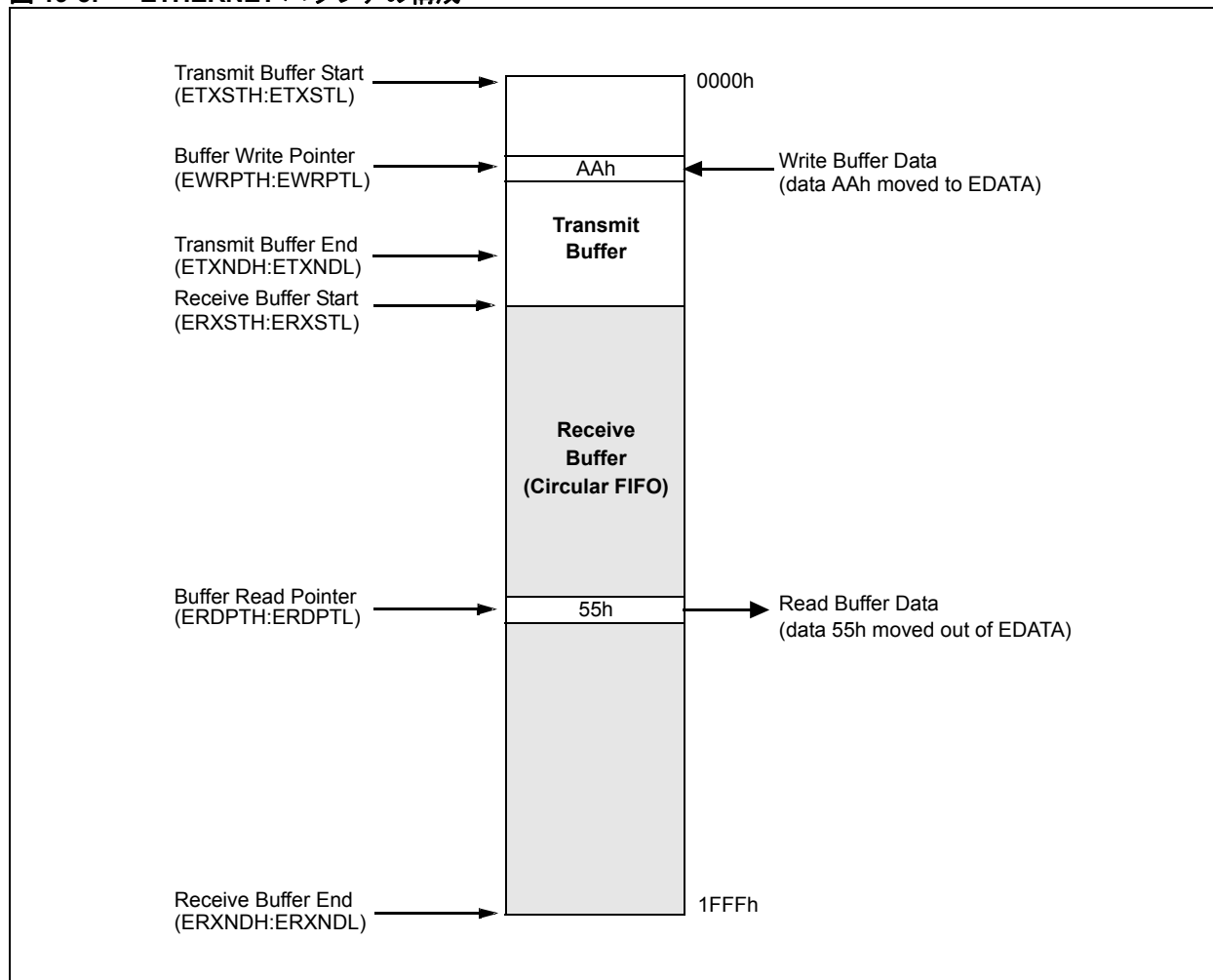
シングルサイクルで書き込みのみを実行する命令は使えますが、AUTOINC が有効になっている場合は ERDPT レジスタがインクリメントしてしまいます。

ERDPT と EWRPT の両方がインクリメントする例:

```
CLRF   EDATA
SETF   EDATA
MOVWF  EDATA
```

# PIC18F97J60 ファミリ

図 19-5: ETHERNET バッファの構成





## 19.2.1.2 受信バッファ

受信バッファは、ハードウェアが管理する環状 FIFO バッファとして構成されます。メモリ内における受信バッファのサイズと位置は、ERXSTH:ERXSTL と ERXNDH:ERXNDL の各レジスタペアをポインタとして使って定義します。受信 FIFO バッファは ERXST ペアが指し示すバイトから開始し、ERXND ペアが指し示すバイトで終了します。

Ethernet インターフェイスから受信したデータバイトは、順番に受信バッファに書き込まれます。そして ERXND ポインタが指し示すアドレスへの書き込みが完了すると、ハードウェアは次に受信したデータバイトを、ERXST ペアが指し示すアドレスに自動的に書き込みます。このため、受信ハードウェアが FIFO の境界外のアドレスへ書き込みを行う事はありません。

ERXST および ERXND ポインタの値を書き込めるのは、受信ロジックが無効な場合です。ERXEN ビット (ECON1<2>) をセットして受信ロジックを有効にした状態でポインタの値を書き換えてはいけません。

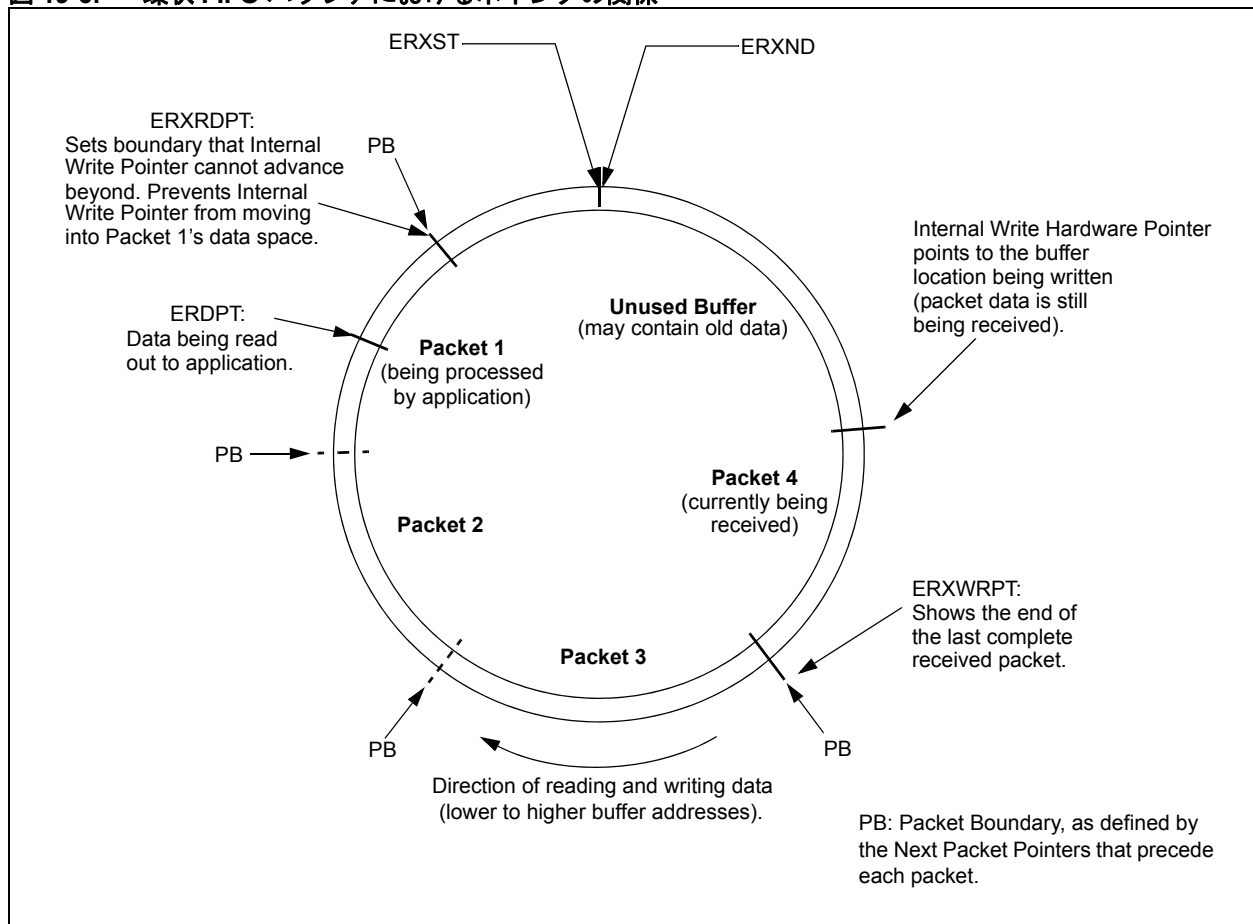
バッファ ハードウェアは、内部ポインタを使って受信中の (検証されていない) データの書き込み先アドレスを決定します (この内部ポインタはユーザアクセス可能

なレジスタには割り当てられていません)。パケットの受信と検証が完了したら、読み出し専用の ERXWRPTH:ERXWRPTL レジスタペアの値をこの内部ポインタの値で置き換えます。このように、ERXWRPT レジスタは受信バッファ内で現在データが書き込まれている場所を定義します。このレジスタは、FIFO の空き領域のサイズを調べるのに便利です。

ERXRDPT レジスタは、FIFO 内で受信ハードウェアによる書き込みを禁止するアドレスを定義します。通常動作時、受信ハードウェアは ERXRDPT レジスタが指し示すアドレスの手前のアドレスまでしかデータを書き込みできません。FIFO がデータで一杯になり、さらに新しいデータを受信しても、ハードウェアは受信済みのデータを上書きしません。新しいデータは破棄され、以前のデータが保持されます。新しいデータを受信するには、受信済みデータの一部または全部を処理したら、アプリケーションで ERXRDPT ポインタを定期的に前へ進める必要があります。

図 19-6 に、環状バッファ方式における受信バッファポインタとパケットデータの関係を示します。この例には 4 つのパケットが示してありますが、実際のパケット数は 4 つとは限りません。

図 19-6: 環状 FIFO バッファにおけるポインタの関係



# PIC18F97J60 ファミリ

## 19.2.1.3 送信バッファ

8 KB のメモリのうち、受信 FIFO バッファとしてプログラミングされていない空間は送信バッファと見なされます。パケットを送信バッファのどの位置に格納するかは、アプリケーションで正しく管理する必要があります。アプリケーションでパケットを送信する際は、送信バッファ内で送信パケットが格納されている場所の開始アドレスと終了アドレスを ETXST および ETXND ポインタにそれぞれ書き込みます。この開始アドレスと終了アドレスが受信バッファと重複しているかどうかは、ハードウェアではチェックしません。バッファの内容が破壊されるのを防ぐには、ETXST および ETXND ポインタが受信バッファ内にある場合、あるいは ETXND ポインタと受信バッファの先頭の間には十分な間隔がない場合、ファームウェアでパケットを送信しないようにする必要があります。詳細は、[セクション 19.5.2「パケットの送信」](#)を参照してください。

## 19.2.1.4 バッファアービタとアクセス調停

Ethernet バッファは、マイクロコントローラの半分のクロックレートで駆動されます。利用できるメモリアクセスの帯域幅は、このクロックレートによって変化します。総帯域幅 (B/s) は、命令レートの 2 倍 ( $2 * F_{cy}$  または  $F_{osc}/2$ ) です。例えばシステムクロック速度が 41.667 MHz の場合、利用可能な総メモリ帯域幅は 20.834 MB/s です。10 Mbit/s の Ethernet シグナリングレートで Ethernet RX エンジンがオーバーランを起こさないように動作するには、1.25 MB/s のバッファメモリ帯域幅が必要です。全二重モードを使う場合、送受信を同時に実行するため、さらに 1.25 MB/s が必要です。

利用可能なメモリ帯域幅には限りがあるため、3 チャンネルアービタを使って RX エンジン、TX/DMA エンジン、マイクロコントローラの CPU (すなわちアプリケーションから EDATA へのアクセス) の間で帯域幅

を割り当てます。このアービタは、EDATA レジスタへのアクセスを最優先し、残りの帯域幅を RX ブロックと TX/DMA ブロックの間で共有します。

この調停では帯域幅の制限があるため、Ethernet モジュールのハードウェアとアプリケーションの間で帯域幅への要求のバランスをとる必要があります。EDATA レジスタへのアクセス頻度が多すぎると、RX/TX ブロックでバッファオーバーラン/アンダーランが発生する可能性があります。そのようなメモリアクセス エラーが発生した場合、BUFFER ビット (ESTAT<6>) と TXERIF または RXERIF 割り込みフラグビットがセットされ、TX または RX 割り込みが発生します (割り込みを有効にしている場合)。この場合、現在の受信パケットが失われるかパケット送信が中止されます。

マイクロコントローラ コアの動作速度を高くすると、パケット損失のリスクは減少します。[表 19-2](#) に示すバッファ調停の制約を守ると、メモリアクセス エラーの発生を防ぐ事ができます。また、EDATA レジスタに連続してアクセスするようなアプリケーションコードセグメントの使用は避けてください。代わりに、EDATA に対する読み書きの間に 1 つ以上の命令 (NOP でも可) を挿入してください。

## 19.2.1.5 バッファへの DMA アクセス

内蔵 DMA コントローラは、チェックサム計算時にはバッファからの読み出しを実行し、メモリコピー時にはバッファに対する読み出しと書き込みを実行する必要があります。DMA も、既述の受信バッファと同じラップアラウンドの規則に従います。順次読み出し中に受信バッファの最後のアドレスに到達すると、ラップアラウンド条件が発生します。書き込み時にはラップアラウンド条件は発生しません。詳細は、[セクション 19.9「ダイレクトメモリアクセス \(DMA\) コントローラ」](#)を参照してください。

**表 19-2: バッファ調停の制約とクロック速度**

Fosc (MHz)	Fcy (MHz)	利用可能な帯域幅 (MB/s)			アンダーラン/オーバーランを防ぐためのアプリケーションの制約
		合計	RX 後	TX 後	
41.667	10.42	20.83	19.58	18.33	EDATA へのアクセスは 2 Tcy ごとに 1 回まで
31.250	7.81	15.63	14.38	13.13	EDATA へのアクセスは 2 Tcy ごとに 1 回まで
25.000	6.25	12.50	11.25	10.00	EDATA へのアクセスは 2 Tcy ごとに 1 回まで
20.833	5.21	10.42	9.17	7.92	EDATA へのアクセスは 2 Tcy ごとに 1 回まで
13.889	3.47	6.94	5.69	4.44	EDATA へのアクセスは 2 Tcy ごとに 1 回まで
12.500	3.13	6.25	5.00	3.75	EDATA へのアクセスは 2 Tcy ごとに 1 回まで
8.333	2.08	4.17	2.92	1.67	EDATA へのアクセスは 3 Tcy ごとに 1 回まで
6.250	1.56	3.13	1.88	0.63	EDATA へのアクセスは 5 Tcy ごとに 1 回まで
4.167	1.04	2.08	0.83	< 0	DMA を使わない。全二重モードを使わない。EDATA へのアクセスは 3 Tcy ごとに 1 回まで
2.778	0.69	1.39	0.14	< 0	DMA を使わない。全二重モードを使わない。EDATA へのアクセスは 10 Tcy ごとに 1 回まで

## 19.2.2 SFR と Ethernet モジュール

他の周辺モジュール同様、Ethernet モジュールもいくつかの SFR で制御します。ただし Ethernet モジュールの SFR は数が多いため、その大部分はマイクロコントローラのデータメモリ空間のバンク 14 の上半分に配置されています。

Ethernet モジュールに関する SFR のうち、以下に示す 5 つの主要 SFR はマイクロコントローラの通常の SFR 領域であるバンク 15 に配置されており、高速なアクセスが可能です。

- ECON1
- EDATA
- EIR
- Ethernet バッファ読み出しポインタペア (ERDPTH と ERDPTL)

ECON1については、以下のセクションで他の Ethernet 制御レジスタと一緒に説明します。EDATA と ERDPTH:ERDPTL は、それぞれ Ethernet データバッファ レジスタとそのレジスタを読み出す際に使うポインタです ( [セクション 19.2.1「Ethernet バッファ](#)

[とバッファポインタ レジスタ](#)参照)。EIRはEthernet 割り込み構造の一部で、[セクション 19.3「Ethernet 割り込み](#)」で説明します。

バンク 14 にある Ethernet SFR の多くは、専用の Ethernet バッファ内のパケットデータの読み書きアドレスを示すポインタレジスタです。この他、パケットパターンマスクまたはチェックサム計算のための情報を格納するレジスタがあります。いくつかのレジスタは、Ethernet モジュール全体の動作、特定の MAC および PHY 機能を制御します。

## 19.2.3 Ethernet 制御レジスタ

ECON1レジスタ ([レジスタ 19-1](#))は、Ethernetモジュールの主要な機能を制御します。受信イネーブルビット、送信要求ビット、DMA 制御ビットは全てこのレジスタに含まれます。それ以外の Ethernet モジュールの主要機能は、ECON2 レジスタ ([レジスタ 19-2](#)) で制御します。ESTAT レジスタ ([レジスタ 19-3](#)) は、Ethernet モジュールと Ethernet 通信の全体的なステータスを通知する目的で使います。

Ethernet SFR のうち先頭が「E」で始まっているものは、Ethernet モジュールが無効の時でもアクセスできます。

**レジスタ 19-1: ECON1: Ethernet 制御レジスタ 1**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
TXRST	RXRST	DMAST	CSUMEN	TXRTS	RXEN	—	—
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」= ビットはセット                  「0」= ビットはクリア                  x = ビットは未知

- bit 7      **TXRST:** 送信ロジック リセットビット  
 1 = 送信ロジックをリセット状態に保持する  
 0 = 通常動作
- bit 6      **RXRST:** 受信ロジック リセットビット  
 1 = 受信ロジックをリセット状態に保持する  
 0 = 通常動作
- bit 5      **DMAST:** DMA 開始 /BUSY ステータスビット  
 1 = DMA コピーまたはチェックサム計算が進行中である ( ソフトウェアでセットされ、ハードウェアまたはソフトウェアでクリアされる )  
 0 = DMA ハードウェアはアイドルである
- bit 4      **CSUMEN:** DMA チェックサム イネーブルビット  
 1 = DMA ハードウェアでチェックサムを計算する  
 0 = DMA ハードウェアでバッファメモリをコピーする
- bit 3      **TXRTS:** 送信要求ビット  
 1 = 送信ロジックがパケットを送信しようとしている ( ソフトウェアでセットされ、ハードウェアまたはソフトウェアでクリアされる )  
 0 = 送信ロジックはアイドルである
- bit 2      **RXEN:** 受信イネーブルビット  
 1 = 現在設定されているフィルタを通過したパケットを受信バッファに書き込む  
 0 = 受信した全てのパケットをハードウェアによって破棄する
- bit 1-0      **未実装:** 「0」として読み出し

# PIC18F97J60 ファミリ

## レジスタ 19-2: ECON2: Ethernet 制御レジスタ 2

R/W-1	R/W-0 <sup>(1)</sup>	R/W-0	U-0	U-0	U-0	U-0	U-0
AUTOINC	PKTDEC	ETHEN	—	—	—	—	—
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

- bit 7      **AUTOINC:** バッファポインタ自動インクリメント イネーブルビット  
 1 = EDATA の読み書きの際に、ERDPT/EWRPT レジスタを自動でインクリメントする  
 0 = EDATA へのアクセス後、ERDPT/EWRPT レジスタの値を自動でインクリメントしない
- bit 6      **PKTDEC:** パケット デクリメント ビット<sup>(1)</sup>  
 1 = EPKTCNT レジスタをデクリメント (-1) する  
 0 = EPKTCNT レジスタをデクリメントしない
- bit 5      **ETHEN:** Ethernet モジュール イネーブルビット  
 1 = Ethernet モジュールを有効にする  
 0 = Ethernet モジュールを無効にする
- bit 4-0      **未実装:** 「0」として読み出し

**Note 1:** このビットはセットされた後、自動的にクリアされます。

## レジスタ 19-3: ESTAT: Ethernet ステータス レジスタ

U-0	R/C-0	U-0	R/C-0	U-0	R-0	R/C-0	R-0
—	BUFFER	—	r	—	RXBUSY	TXABRT	PHYRDY
bit 7							bit 0

### 凡例:

r = 予約済みビット  
 R = 読み出し可能ビット      C = クリア可能ビット      U = 未実装ビット、「0」として読み出し  
 -n = POR 時の値      「1」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

- bit 7      **未実装:** 「0」として読み出し
- bit 6      **BUFFER:** Ethernet バッファエラー ステータスビット  
 1 = Ethernet 読み書き時にバッファエラー (オーバーランまたはアンダーラン) が発生した  
 0 = バッファエラーは発生していない
- bit 5      **未実装:** 「0」として読み出し
- bit 4      **予約済み:** 「0」として書き込み
- bit 3      **未実装:** 「0」として読み出し
- bit 2      **RXBUSY:** 受信 BUSY ビット  
 1 = 受信ロジックがデータパケットを受信中である  
 0 = 受信ロジックはアイドルである
- bit 1      **TXABRT:** 送信中止エラービット  
 1 = 送信要求が中止された  
 0 = 送信中止エラーは発生していない
- bit 0      **PHYRDY:** Ethernet PHY クロックレディ ビット  
 1 = Ethernet PHY 起動タイマのカウントが終了し、PHY の準備が完了している  
 0 = Ethernet PHY 起動タイマがカウント中で、PHY の準備ができていない

## 19.2.4 MAC および MII レジスタ

これらの SFR は、MAC の動作と、MIIM を介して PHY の動作を制御します。MAC レジスタと MII レジスタは、データアドレス E80h ~ E85h、E8Ah、EA0h ~ EB9h に配置されています。

MAC レジスタと MII レジスタはマイクロコントローラの汎用メモリマップに含まれますが、これらのレジスタは MAC モジュールに内包されています。ホストインターフェイス ロジックがマイクロコントローラのデータ / アドレスバス データを変換する事により、これらのレジスタにアクセスできます。このため、ファームウェアから MAC および MII SFR へのアクセス方法にはホストインターフェイス ロジックによる制約があります。詳細は以下の Note を参照してください。

**Note 1:** Ethernet モジュールが有効 (ETHEN = 1) でない場合は MAC および MII SFR にはアクセスしないでください。

**2:** MAC/MII レジスタへの連続アクセスはサポートされません。MAC または MII レジスタへ複数回アクセスする場合、間で 1 つ以上の命令 (NOP でも可) を実行する必要があります。

MAC 動作とパケット構成は 3 つの MACON レジスタで制御します。[レジスタ 19-4](#) ~ [レジスタ 19-6](#) にこれらのレジスタを示します。

II レジスタは MIIM インターフェイスの制御に使い、PHY レジスタとの通信チャンネルとして機能します。[レジスタ 19-7](#)と[レジスタ 19-8](#)にこれらのレジスタを示します。

### レジスタ 19-4: MACON1: MAC 制御レジスタ 1

U-0	U-0	U-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	r	TXPAUS	RXPAUS	PASSALL	MARXEN
bit 7							bit 0

<b>凡例:</b>	r = 予約済みビット		
R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し	
-n = POR 時の値	「1」= ビットはセット	「0」= ビットはクリア	x = ビットは未知

bit 7-5	<b>未実装:</b> 「0」として読み出し
bit 4	<b>予約済み:</b> 使用不可
bit 3	<b>TXPAUS:</b> PAUSE 制御フレーム送信イネーブルビット 1 = MAC からの PAUSE 制御フレームの送信を許可する (全二重モードでのフロー制御に必要) 0 = PAUSE フレームの送信を禁止する
bit 2	<b>RXPAUS:</b> PAUSE 制御フレーム受信イネーブルビット 1 = PAUSE 制御フレームを受信したら送信を禁止する (通常の動作) 0 = PAUSE 制御フレームを受信しても無視する
bit 1	<b>PASSALL:</b> 全受信フレーム通過イネーブルビット 1 = フィルタを通過して MAC が受信した制御フレームを受信バッファに書き込む 0 = MAC で処理した後に制御フレームを破棄する (通常の動作)
bit 0	<b>MARXEN:</b> MAC 受信イネーブルビット 1 = MAC によるパケット受信を有効にする 0 = パケット受信を無効にする

# PIC18F97J60 ファミリ

## レジスタ 19-5: MACON3: MAC 制御レジスタ 3

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PADCFG2	PADCFG1	PADCFG0	TXCRCEN	PHDREN	HFRMEN	FRMLNEN	FULDPX
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

- bit 7-5      **PADCFG<2:0>**: 自動パディング /CRC コンフィグレーション ビット
- 111 = 全てのショートフレームを 64 バイトになるまでゼロでパディングした後、有効な CRC を付加する
  - 110 = ショートフレームに自動でパディングしない
  - 101 = VLAN プロトコル フレーム (タイプフィールドが 8100h) を MAC で自動検出し、64 バイトになるまで自動でパディングする。VLAN フレームでない場合は 60 バイトになるまでパディングする。パディング後に有効な CRC を付加する
  - 100 = ショートフレームに自動でパディングしない
  - 011 = 全てのショートフレームを 64 バイトになるまでゼロでパディングした後、有効な CRC を付加する
  - 010 = ショートフレームに自動でパディングしない
  - 001 = 全てのショートフレームを 60 バイトになるまでゼロでパディングした後、有効な CRC を付加する
  - 000 = ショートフレームに自動でパディングしない
- bit 4      **TXCRCEN**: 送信 CRC イネーブルビット
- 1 = PADCFG<2:0> ビットの設定に関わらず、MAC は送信する全てのフレームに有効な CRC を付加する。PADCFG ビットで有効な CRC を付加するように設定している場合、TXCRCEN ビットもセットする必要がある
  - 0 = MAC は CRC を付加しない。最後の 4 バイトをチェックし、無効な CRC なら送信ステータスペクタで報告する
- bit 3      **PHDREN**: プロプライエタリ ヘッダ イネーブルビット
- 1 = MAC が受け取ったフレームに、CRC 計算で使わない 4 バイトのプロプライエタリ ヘッダが含まれている
  - 0 = プロプライエタリ ヘッダが含まれておらず、全てのデータを CRC で使う (通常動作)
- bit 2      **HFRMEN**: ジャンボフレーム イネーブルビット
- 1 = ジャンボフレームと規格違反のサイズのフレームの送受信を許可する
  - 0 = MAMXFL を超える長さのフレームは送受信時に切り捨てられる
- bit 1      **FRMLNEN**: フレーム長チェック イネーブルビット
- 1 = 送受信フレームのタイプ / 長さフィールドをチェックする。長さを表している場合はフレームサイズと比較する。一致しない場合は送信 / 受信ステータスペクタで報告する
  - 0 = タイプ / 長さフィールドと実際のフレーム長を比較しない
- bit 0      **FULDPX**: MAC 全二重イネーブルビット
- 1 = MAC の動作を全二重モードにする。PDPXMD ビット (PHCON1<8>) もアプリケーションでセットする必要がある
  - 0 = MAC の動作を半二重モードにする。PDPXMD ビットもアプリケーションでクリアする必要がある

# PIC18F97J60 ファミリ

レジスタ 19-6: MACON4: MAC 制御レジスタ 4

U-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R-0	R-0
—	DEFER	r	r	—	—	r	r
bit 7						bit 0	

**凡例:** r = 予約済みビット  
R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し  
-n = POR 時の値 「1」= ビットはセット 「0」= ビットはクリア x = ビットは未知

- bit 7 **未実装:** 「0」として読み出し  
bit 6 **DEFER:** 送信延期イネーブルビット (半二重モードの場合のみ適用)  
1 = 送信時に媒体が占有されている場合、MAC は媒体が解放されるまで無期限に待機する (IEEE 802.3 に準拠するにはこの設定を使う)  
0 = 媒体が占有されている場合、MAC は最大延期時間に達した後で送信を中止する  
bit 5-4 **予約済み:** 「0」のまま  
bit 3-2 **未実装:** 「0」として読み出し  
bit 1-0 **予約済み:** 「0」のまま

レジスタ 19-7: MICMD: MII コマンドレジスタ

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	MIISCAN	MIIRD
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」= ビットはセット 「0」= ビットはクリア x = ビットは未知

- bit 7-2 **未実装:** 「0」として読み出し  
bit 1 **MIISCAN:** MII スキャン イネーブルビット  
1 = MIREGADR が指し示すアドレスの PHY レジスタを連続して読み出して、データを MIIRD レジスタに格納する  
0 = MII マネジメント スキャンは進行中でない  
bit 0 **MIIRD:** MII 読み出しイネーブルビット  
1 = MIREGADR が指し示すアドレスの PHY レジスタを 1 回だけ読み出して、データを MIIRD レジスタに格納する  
0 = MII マネジメント読み出しは進行中でない

# PIC18F97J60 ファミリ

## レジスタ 19-8: MISTAT: MII ステータス レジスタ

U-0	U-0	U-0	U-0	R-0	R-0	R-0	R-0
—	—	—	—	r	NVALID	SCAN	BUSY
bit 7							bit 0

<b>凡例:</b>	r = 予約済みビット		
R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し	
-n = POR 時の値	「1」= ビットはセット	「0」= ビットはクリア	x = ビットは未知

bit 7-4 **未実装:** 「0」として読み出し

bit 3 **予約済み:** 使用不可

bit 2 **NVALID:** MII マネジメント読み出しデータ無効ビット

1 = MIRD レジスタの内容はまだ有効ではない

0 = MII マネジメントの読み出しサイクルが完了し、MIRD レジスタが更新済みである

bit 1 **SCAN:** MII マネジメント スキャン動作ビット

1 = MII マネジメント スキャンが進行中である

0 = MII マネジメント スキャンは進行中ではない

bit 0 **BUSY:** MII マネジメント BUSY ビット

1 = PHY レジスタが現在読み出し中または書き込み中である。内部同期のため、ファームウェアのコマンド (MIISCAN または MIIRD ビットをセットする) または MIWRH レジスタへの書き込みから 2 Tcy 遅れて、このビットはハードウェアによってセットされる

0 = MII マネジメント インターフェイスはアイドルである

### 19.2.5 PHY レジスタ

PHY レジスタは、PHY モジュールの設定、制御、ステータス情報に使用します。PHY レジスタは全て 16 ビット幅です。

PHY レジスタは 5 ビットのアドレスで指定するため合計で 32 個のレジスタにアクセスできますが、実装されているのは 7 アドレスのみです。表 19-3 に、実装されているレジスタの一覧を示します。レジスタ 19-9 ~ レジスタ 19-13 で、主な PHY 制御レジスタを説明します。その他の PHY 制御 / ステータス レジスタについては、本章で後述します。

未実装のレジスタには決して書き込まないでください。これらのレジスタを読み出すと、不確定なデータが返されます。実装されているレジスタの予約済みビットのうち、「書き込み可能ビット」と記載されているものは全て、レジスタの説明欄に記載されている値で書き込む必要があります。これらの予約済みビットを読み出しても無視されます。

PHY レジスタは MII マネジメント インターフェイス経由でしかアクセスできません。PHY 起動タイマのカウントが終了して PHYRDY ビット (ESTAT<0>) がセットされるまで、PHY レジスタを読み書きしてはいけません。

#### 19.2.5.1 PHSTAT レジスタ

PHSTAT1 および PHSTAT2 レジスタには、PHY モジュールの現在の動作状態 (特にネットワーク上の他のノードとの通信リンクの状態) を示す読み出し専用ビットがあります。

PHSTAT1 レジスタ (レジスタ 19-10) には LLSTAT ビットがあります。PHSTAT1 レジスタを前回読み出した後に PHY リンクがダウンすると、このビットはクリアされ Low にラッチします。アプリケーションで定期的に LLSTAT をポーリングすると、リンクがダウンした事をすぐに知る事ができます。この方法は、リンクステータス変化割り込みを使わない場合は特に便利です。

PHSTAT2 レジスタ (レジスタ 19-12) は、PHY モジュールがネットワークに接続されているかどうか、現在送受信中であることを示すステータスビットを含みます。

#### 19.2.5.2 PHY レジスタへのアクセス

前述の通り、PHY レジスタは他のレジスタとは異なるメモリ空間に存在し、マイクロコントローラから直接アクセスする事はできません。PHY レジスタには、MIIM (Media Independent Interface Management) を実装する Ethernet SFR バンク内の MII レジスタ経由でアクセスします。

アクセス方法は Ethernet バッファの場合とほぼ同じですが、読み書きにそれぞれ専用のバッファ (MIRDH:MIRDL と MIWRH:MIWRL) と 5 ビットのアドレスレジスタ (MIREGADR) を使います。また、読み書きの制御に MICMD および MISTAT レジスタを使います。



PHY レジスタからの読み出し手順は以下の通りです。

1. 読み出す PHY レジスタのアドレスを MIREGADR レジスタに書き込む。
2. MIIRD ビット (MICMD<0>) をセットする。読み出しが開始し、2 TcY 後に BUSY ビット (MISTAT<0>) がセットされる。
3. 10.24  $\mu$ s 待ってから BUSY ビットをポーリングし、読み出しが完了した事を確認する。MAC が PHY レジスタの内容を取得すると、BUSY ビットは自己クリアする。BUSY ビットがセットされている間、ユーザ アプリケーションは MIISCAN を開始する事も、MIWRH レジスタに書き込む事もできない。
4. MIIRD ビットをクリアする。
5. MIRD L および MIRD H レジスタから 16 ビットの PHY レジスタ全体を読み出す。

PHY レジスタへの書き込み手順は以下の通りです。

1. 書き込む PHY レジスタのアドレスを MIREGADR レジスタに書き込む。
2. 書き込むデータの下位 8 ビットを MIWRL レジスタに書き込む。
3. 書き込みデータの上位 8 ビットを MIWRH レジスタに書き込む。MIWRH レジスタに書き込むと MII トランザクションが自動的に開始するため、必ず MIWRL レジスタから先に書き込む必要がある。2 TcY 後に BUSY ビットが自動的にセットされる。

PHY レジスタは、MII の動作 (10.24  $\mu$ s かかる) が完了してから書き込まれます。書き込みが完了すると、BUSY ビットは自己クリアします。BUSY ビットがセットされている間は、アプリケーションで MII スキャンまたは読み出しを開始しないでください。

PHY レジスタに書き込む際は、16 ビット全体が一括で書き込まれます。ビット単位またはバイト単位での選択的な書き込み機能は実装されていません。レジスタの一部のビットのみを書き換える場合、コントローラでまず PHY レジスタを読み出し、読み出したデータを変更してから PHY レジスタに書き戻す必要があります。

MAC は、PHY レジスタに対して自動的に連続読み出しを実行するように設定する事もできます。このようなスキャン動作を実行する手順は以下の通りです。

1. スキャンする PHY レジスタのアドレスを MIREGADR レジスタに書き込む。
2. MIISCAN ビット (MICMD<1>) をセットする。スキャン動作が開始し、2 TcY 後に BUSY ビットがセットされる。

MIISCAN ビットをセットすると、NVALID (MISTAT<2>)、SCAN、BUSY ビットもセットされます。最初の読み出しは 10.24  $\mu$ s 後に終了します。スキャンがキャンセルされるまで、同じ間隔で 2 回目以降の読み出しが実行され、MIRD L および MIRD H レジスタが自動的に繰り返し更新されます。最初の読み出しが完了したかどうかは、NVALID ビットをポーリングして調べます。

MIRD レジスタが更新された事を知らせるステータス情報はありません。MII レジスタは一度に 1 つしか読み出せないため、MIRD L と MIRD H の値はスキャン中に PHY から全く同じタイミングで読み出されたものではない事に注意してください。

スキャン動作が必要な間は、MIISCAN ビットをセットしたままにしておいてください。MIISCAN ビットを「0」にクリアすると、最後の読み出しシーケンスが完了した時点で BUSY ビットと SCAN ビットが自動的にクリアされます。MIISCAN をセットしたまま MIREGADR を更新しないでください。

スキャンの進行中は、MIWRH レジスタの読み書き等、PHY に対して新しい処理を開始する事はできません。この場合、MIISCAN ビットをクリアし、BUSY ビットをポーリングすることによって、進行中のスキャンをキャンセルします。BUSY ビットがクリアされたら、新しい処理を開始できます。

表 19-3: PIC18F97J60 ファミリの PHY レジスタのまとめ

アドレス	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	リセット時の値
00h	PHCON1	r	r	-	-	r	r	-	PDPXMD	r	-	-	-	-	-	-	-	00-- 00-0 0--- ----
01h	PHSTAT1	-	-	-	r	r	-	-	-	-	-	-	-	-	LLSTAT	r	-	---1 1--- ---- -00-
10h	PHCON2	-	FRCLNK	r	r	r	r	r	HDLDIS	r	r	r	RXAPDIS	r	r	r	r	-000 0000 0000 0000
11h	PHSTAT2	-	-	TXSTAT	RXSTAT	COLSTAT	LSTAT	r	-	-	-	r	-	-	-	-	-	--00 00x- --0- ----
12h	PHIE	r	r	r	r	r	r	r	r	r	r	r	PLNKIE	r	r	r	r	xxxx xxxx xx00 xx00
13h	PHIR	r	r	r	r	r	r	r	r	r	r	r	PLNKIF	r	PGIF	r	r	xxxx xxxx xx00 00x0
14h	PHLCON	r	r	r	r	LACFG3	LACFG2	LACFG1	LACFG0	LBCFG3	LBCFG2	LBCFG1	LBCFG0	LFRQ1	LFRQ0	STRCH	r	0011 0100 0010 001x

凡例: x = 未知、u = 不変、- = 未実装 (「0」として読み出し)、r = 予約済みビットのため変更不可です。網掛けの部分は未実装で、「0」として読み出されます。

# PIC18F97J60 ファミリ

## レジスタ 19-9: PHCON1: PHY 制御レジスタ 1

R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	U-0	R/W-0
r	r	—	—	r	r	—	PDPXMD
bit 15							bit 8
R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
r	—	—	—	—	—	—	—
bit 7							bit 0

<b>凡例:</b>	r = 予約済みビット
R = 読み出し可能ビット	W = 書き込み可能ビット
U = 未実装ビット、「0」として読み出し	
-n = POR 時の値	「1」= ビットはセット 「0」= ビットはクリア x = ビットは未知

- bit 15-14    **予約済み:** 「0」として書き込み
- bit 13-12    **未実装:** 「0」として読み出し
- bit 11-10    **予約済み:** 「0」として書き込み
- bit 9        **未実装:** 「0」として読み出し
- bit 8        **PDPXMD:** PHY 二重モードビット  
               1 = PHY の動作を全二重モードにする。FULDPX ビット (MACON3<0>) もアプリケーションでセットする必要がある  
               0 = PHY の動作を半二重モードにする。FULDPX ビットもアプリケーションでクリアする必要がある
- bit 7        **予約済み:** 常に「0」
- bit 6-0     **未実装:** 「0」として読み出し

## レジスタ 19-10: PHSTAT1: PHY ステータス レジスタ 1

U-0	U-0	U-0	R-1	R-1	U-0	U-0	U-0
—	—	—	r	r	—	—	—
bit 15							bit 8
U-0	U-0	U-0	U-0	U-0	R/LL-0	R/LH-0	U-0
—	—	—	—	—	LLSTAT	r	—
bit 7							bit 0

<b>凡例:</b>	「1」= ビットはセット	r = 予約済みビット
R = 読み出し専用ビット	「0」= ビットはクリア	U = 未実装ビット、「0」として読み出し
-n = POR 時の値	R/L = 読み出し専用ラッチビット	LL = ラッチ Low ビット    LH = ラッチ High ビット

- bit 15-13    **未実装:** 「0」として読み出し
- bit 12-11    **予約済み:** 「1」として読み出し
- bit 10-3     **未実装:** 「0」として読み出し
- bit 2        **LLSTAT:** PHY リンクラッチ ステータスビット  
               1 = 前回 PHSTAT1 を読み出してから一度もリンクがダウンしていない  
               0 = リンクがダウンしている、または PHSTAT1 を前回読み出してからダウンしていた期間がある
- bit 1        **予約済み:** 読み出しは無視される
- bit 0        **未実装:** 「0」として読み出し

# PIC18F97J60 ファミリ

## レジスタ 19-11: PHCON2: PHY 制御レジスタ 2

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	FRCLNK	r	r	r	r	r	HDLDIS
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
r	r	r	RXAPDIS	r	r	r	r
bit 7							bit 0

**凡例:**  
 r = 予約済みビット  
 R = 読み出し可能ビット    W = 書き込み可能ビット    U = 未実装ビット、「0」として読み出し  
 -n = POR 時の値    「1」= ビットはセット    「0」= ビットはクリア    x = ビットは未知

- bit 15      **未実装:** 「0」として読み出し
- bit 14      **FRCLNK:** PHY 強制リンクアップ ビット  
             1 = リンクパートナー未検出時でも強制的にリンクアップする (送信を常に許可)  
             0 = 通常動作 (リンクパートナー未検出時は PHY が送信をブロック)
- bit 13-9    **予約済み:** 「0」として書き込み
- bit 8        **HDLDIS:** PHY 半二重ループバック ディセーブルビット  
             1 = 通常の PHY 動作  
             0 = 予約済み
- bit 7-5     **予約済み:** 「0」として書き込み
- bit 4        **RXAPDIS:** RX+/RX- 動作モードビット  
             1 = 通常動作  
             0 = 予約済み
- bit 3-0     **予約済み:** 「0」として書き込み

**Note:** HDLDIS または RXAPDIS ビットがクリアされている場合 (リセット時の既定値)、そのままでは Ethernet の動作が不正になる事があります。Ethernet モジュールを使う前の初期化でこれらのビットを必ずセットしてください。

# PIC18F97J60 ファミリ

## レジスタ 19-12: PHSTAT2: PHY ステータス レジスタ 2

U-0	U-0	R-0	R-0	R-0	R-0	R-x	U-0
—	—	TXSTAT	RXSTAT	COLSTAT	LSTAT	r	—
bit 15							bit 8

U-0	U-0	R-0	U-0	U-0	U-0	U-0	U-0
—	—	r	—	—	—	—	—
bit 7							bit 0

<b>凡例:</b>	r = 予約済みビット
R = 読み出し可能ビット	W = 書き込み可能ビット
-n = POR 時の値	「1」= ビットはセット
	U = 未実装ビット、「0」として読み出し
	「0」= ビットはクリア
	x = ビットは未知

- bit 15-14   **未実装:** 「0」として読み出し
- bit 13      **TXSTAT:** PHY 送信ステータスビット  
1 = PHY がデータを送信中である  
0 = PHY はデータを送信中でない
- bit 12      **RXSTAT:** PHY 受信ステータスビット  
1 = PHY がデータを受信中である  
0 = PHY はデータを受信中でない
- bit 11      **COLSTAT:** PHY コリジョン ステータスビット  
1 = コリジョンが発生中である (半二重モードで PHY が送信と受信を同時に実行しようとしている)  
0 = コリジョンは発生していない
- bit 10      **LSTAT:** PHY リンク ステータスビット  
1 = リンクアップしている  
0 = リンクダウンしている
- bit 9        **予約済み:** 読み出しは無視される
- bit 8-6     **未実装:** 「0」として読み出し
- bit 5        **予約済み:** 読み出しは無視される
- bit 4-0     **未実装:** 「0」として読み出し

# PIC18F97J60 ファミリ

## レジスタ 19-13: PHLCON: PHY モジュール LED 制御レジスタ

R/W-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-1	R/W-0	R/W-0
r	r	r	r	LACFG3	LACFG2	LACFG1	LACFG0
bit 15				bit 8			

R/W-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0	R/W-1	R/W-x
LBCFG3	LBCFG2	LBCFG1	LBCFG0	LFRQ1	LFRQ0	STRCH	r
bit 7							bit 0

<b>凡例:</b>	r = 予約済みビット
R = 読み出し可能ビット	W = 書き込み可能ビット
-n = POR 時の値	「1」= ビットはセット
	「0」= ビットはクリア
	x = ビットは未知

bit 15-14 **予約済み:** 「0」として書き込み

bit 13-12 **予約済み:** 「1」として書き込み

bit 11-8 **LACFG<3:0>:** LEDA コンフィグレーション ビット

- 0000 = 予約済み
- 0001 = 送信アクティビティを表示 (ストレッチ可能)
- 0010 = 受信アクティビティを表示 (ストレッチ可能)
- 0011 = コリジョン アクティビティを表示 (ストレッチ可能)
- 0100 = リンクステータスを表示
- 0101 = 全二重 / 半二重ステータスを表示
- 0110 = 予約済み
- 0111 = 送受信アクティビティを表示 (ストレッチ可能)
- 1000 = 点灯
- 1001 = 消灯
- 1010 = 高速点滅
- 1011 = 低速点滅
- 1100 = リンクステータスと受信アクティビティを表示 (常にストレッチ)
- 1101 = リンクステータスと送受信アクティビティを表示 (常にストレッチ)
- 111x = 予約済み

bit 7-4 **LBCFG<3:0>:** LEDB コンフィグレーション ビット

- 0000 = 予約済み
- 0001 = 送信アクティビティを表示 (ストレッチ可能)
- 0010 = 受信アクティビティを表示 (ストレッチ可能)
- 0011 = コリジョン アクティビティを表示 (ストレッチ可能)
- 0100 = リンクステータスを表示
- 0101 = 全二重 / 半二重ステータスを表示
- 0110 = 予約済み
- 0111 = 送受信アクティビティを表示 (ストレッチ可能)
- 1000 = 点灯
- 1001 = 消灯
- 1010 = 高速点滅
- 1011 = 低速点滅
- 1100 = リンクステータスと受信アクティビティを表示 (常にストレッチ)
- 1101 = リンクステータスと送受信アクティビティを表示 (常にストレッチ)
- 111x = 予約済み

bit 3-2 **LFRQ<1:0>:** LED パルスストレッチ時間コンフィグレーション ビット (表 19-1 参照)

- 11 = 予約済み
- 10 = LED イベントを TLSTRCH だけストレッチする
- 01 = LED イベントを TMSTRCH だけストレッチする
- 00 = LED イベントを TNSTRCH だけストレッチする

bit 1 **STRCH:** LED パルスストレッチ イネーブルビット

- 1 = ストレッチ可能な LED イベントで、LFRQ<1:0> の設定に基づいて LED の点灯時間を長くする
- 0 = ストレッチ可能な LED イベントで、実際にイベントが発生している間だけ LED を点灯する

bit 0 **予約済み:** 「0」として書き込み

## 19.3 Ethernet 割り込み

Ethernet モジュールは複数の割り込み条件を生成できます。これら全ての割り込み要因に対処するため、Ethernet モジュールはマイクロコントローラの割り込みロジック構造と同じような専用の割り込みロジック構造を備えています。各種割り込み条件のイネーブルビットとフラグビットには別々のレジスタが用意されています。

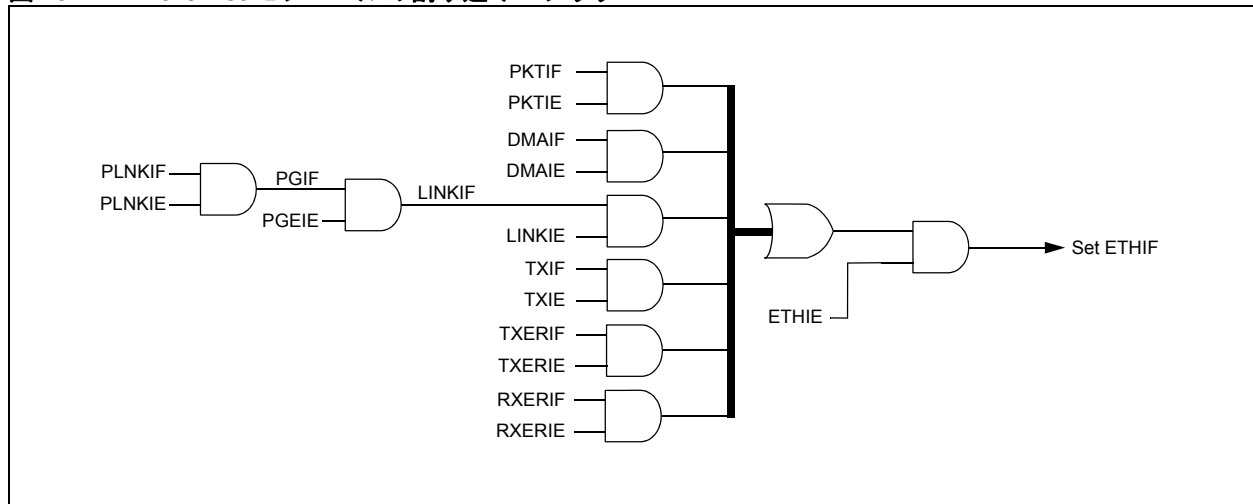
EIE レジスタには各割り込み要因に対応した割り込みイネーブルビットがあり、EIR レジスタには対応する割り込みフラグビットがあります。割り込みが発生すると、対応する割り込みフラグがセットされます。その割り込みが EIE レジスタで有効であり、かつ Ethernet グローバル割り込みイネーブルビット ETHIE がセットされていると、マイクロコントローラのマスタ Ethernet 割り込みフラグ (ETHIF) が適切にセットされます ( 図 19-7 参照 )。

**Note:** 割り込み条件が発生すると、それに対応するイネーブルビットまたはグローバルイネーブルビットの状態に関わらず、対応する割り込みフラグビットがセットされます (LINKIF 割り込みフラグビットを除く)。ユーザ ソフトウェアは、割り込みを有効にする前に、対応する割り込みフラグビットを確実にクリアしておく必要があります。これにより、ソフトウェアポーリングが可能になります。

### 19.3.1 制御割り込み (ETHIE)

レジスタ 19-14 ~ レジスタ 19-17 に、制御割り込みに関連する 4 つのレジスタを示します。

図 19-7: Ethernet モジュールの割り込みロジック



# PIC18F97J60 ファミリ

## レジスタ 19-14: EIE: Ethernet 割り込みイネーブル レジスタ

U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
—	PKTIE	DMAIE	LINKIE	TXIE	—	TXERIE	RXERIE
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

- bit 7      **未実装:** 「0」として読み出し
- bit 6      **PKTIE:** 受信パケット保留割り込みイネーブルビット  
1 = 受信パケット保留割り込みを有効にする  
0 = 受信パケット保留割り込みを無効にする
- bit 5      **DMAIE:** DMA 割り込みイネーブルビット  
1 = DMA 割り込みを有効にする  
0 = DMA 割り込みを無効にする
- bit 4      **LINKIE:** リンクステータス変化割り込みイネーブルビット  
1 = PHY からのリンクステータス変化割り込みを有効にする  
0 = リンクステータス変化割り込みを無効にする
- bit 3      **TXIE:** 送信割り込みイネーブルビット  
1 = 送信割り込みを有効にする  
0 = 送信割り込みを無効にする
- bit 2      **未実装:** 「0」として読み出し
- bit 1      **TXERIE:** 送信エラー割り込みイネーブルビット  
1 = 送信エラー割り込みを有効にする  
0 = 送信エラー割り込みを無効にする
- bit 0      **RXERIE:** 受信エラー割り込みイネーブルビット  
1 = 受信エラー割り込みを有効にする  
0 = 受信エラー割り込みを無効にする



# PIC18F97J60 ファミリ

## レジスタ 19-15: EIR: Ethernet 割り込み要求 (フラグ) レジスタ

U-0	R-0	R/C-0	R-0	R/C-0	U-0	R/C-0	R/C-0
—	PKTIF	DMAIF	LINKIF	TXIF	—	TXERIF	RXERIF
bit 7							bit 0

### 凡例:

R = 読み出し可能ビット      C = クリア可能ビット      U = 未実装ビット、「0」として読み出し  
-n = POR 時の値      「1」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

- bit 7      **未実装:** 「0」として読み出し
- bit 6      **PKTIF:** 受信パケット保留割り込みフラグビット  
1 = 受信バッファに未処理パケットがある。PKTDEC (ECON2<6>) をセットして EPKTCNT が 0 に  
    デクリメントした場合のみクリアされる  
0 = 受信バッファはエンプティである
- bit 5      **DMAIF:** DMA 割り込みフラグビット  
1 = DMA コピーまたはチェックサム計算が完了している  
0 = 保留中の DMA 割り込みは存在しない
- bit 4      **LINKIF:** リンクステータス変化割り込みフラグビット  
1 = リンクステータスが変化した事がPHYから通知された(PHIRレジスタを読み出すとクリアされる)  
0 = リンクステータスは変化していない
- bit 3      **TXIF:** 送信割り込みフラグビット  
1 = 送信要求が終了した  
0 = 保留中の送信割り込みは存在しない
- bit 2      **未実装:** 「0」として読み出し
- bit 1      **TXERIF:** 送信エラー割り込みフラグビット  
1 = 送信エラーが発生した  
0 = 送信エラーは発生していない
- bit 0      **RXERIF:** 受信エラー割り込みフラグビット  
1 = バッファ領域の不足またはバッファ オーバーランによってパケットの受信が中止された  
0 = 保留中の受信エラー割り込みは存在しない

# PIC18F97J60 ファミリ

## レジスタ 19-16: PHIE: PHY 割り込みイネーブル レジスタ

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
r	r	r	r	r	r	r	r
bit 15							bit 8

R-0	R-0	R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0
r	r	r	PLNKIE	r	r	PGEIE	r
bit 7							bit 0

**凡例:** r = 予約済みビット  
 R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し  
 -n = POR 時の値 「1」= ビットはセット 「0」= ビットはクリア x = ビットは未知

- bit 15-6 **予約済み:** 「0」として書き込む。読み出しは無視される
- bit 5 **予約済み:** 常に「0」
- bit 4 **PLNKIE:** PHY リンクステータス変化割り込みイネーブルビット  
 1 = PHY リンクステータス変化割り込みを有効にする  
 0 = PHY リンクステータス変化割り込みを無効にする
- bit 3-2 **予約済み:** 「0」として書き込む。読み出しは無視される
- bit 1 **PGEIE:** PHY グローバル割り込みイネーブルビット  
 1 = PHY 割り込みを有効にする  
 0 = PHY 割り込みを無効にする
- bit 0 **予約済み:** 常に「0」

## レジスタ 19-17: PHIR: PHY 割り込み要求 (フラグ) レジスタ

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
r	r	r	r	r	r	r	r
bit 15							bit 8

R-x	R-x	R-0	R/SC-0	R-0	R/SC-0	R-x	R-0
r	r	r	PLNKIF	r	PGIF	r	r
bit 7							bit 0

**凡例:** r = 予約済みビット  
 R = 読み出し可能ビット SC = 自己クリア可能ビット U = 未実装ビット、「0」として読み出し  
 -n = POR 時の値 「1」= ビットはセット 「0」= ビットはクリア x = ビットは未知

- bit 15-6 **予約済み:** 読み出しは無視される
- bit 5 **予約済み:** 「0」として読み出し
- bit 4 **PLNKIF:** PHY リンクステータス変化割り込みフラグビット  
 1 = 前回 PHIR を読み出してから PHY のリンクステータスが変化した。読み出すと「0」にリセットされる  
 0 = 前回 PHIR を読み出してから PHY のリンクステータスは変化していない
- bit 3 **予約済み:** 「0」として読み出し
- bit 2 **PGIF:** PHY グローバル割り込みフラグビット  
 1 = 前回 PHIR を読み出してから有効な PHY 割り込みが発生した。読み出すと「0」にリセットされる  
 0 = PHY 割り込みは発生していない
- bit 1 **予約済み:** 読み出しは無視される
- bit 0 **予約済み:** 「0」として読み出し

## 19.3.1.1 受信エラー割り込み (RXERIF)

受信エラー割り込みは、受信中のパケットがエラー条件によって受信が中止された事を示すために使います。エラーには以下の3種類があります。

1. 空きバッファ領域が不足して受信パケットを格納できなかった (バッファ オーバーフロー)
2. EPKTCNT カウンタの値が 255 に達しており、これ以上パケットを受信するとオーバーフローになる
3. Ethernet RX ハードウェアに割り当てられているメモリ帯域幅が不足しており、受信データをバッファに書き込む事ができない

パケットを受信中に受信エラーが発生すると、パケットの受信は中止され (パケットは完全に失われます)、RXERIF ビットが「1」にセットされます。セットされた RXERIF ビットは、ファームウェアまたはリセット条件によってのみクリアされます。受信エラー割り込みと Ethernet 割り込みの両方を有効に (RXERIF ビットと ETHIE ビットをセット) している場合、Ethernet 割り込みが生成されます。受信エラー割り込みを無効に (RXERIF ビットと ETHIE ビットのどちらかをクリア) している場合、アプリケーションで RXERIF ビットをポーリングして適切に対応する事ができます。

通常、上記 1、2 の受信エラー条件 (バッファ オーバーフローまたはあと 1 回の受信で EPKTCNT オーバーフロー) の場合、アプリケーションは受信バッファ内の未処理のパケットを処理した後、ERXRDPST レジスタペアを (下位バイトから先に) 進め、EPKTCNT レジスタをデクリメントして、次のパケット受信用の空き領域を確保します。パケット処理の詳細は、[セクション 19.5.3.3 「受信バッファ空間の解放」](#) を参照してください。処理が完了したら、RXERIF ビットをアプリケーションでクリアしてください。

上記の 3 番目のエラー条件 (RX メモリ帯域幅の不足) かどうかは、BUFFER ビット (ESTAT<6>) がセットされているかどうかで調べる事ができます。メモリアクセスエラーで BUFFER がセットされても通常は一過性であるため、運用時の対策は不要です。BUFFER エラーが頻繁または常時発生する場合、アプリケーションとそのバッファのメモリ帯域幅割り当ての調整が必要でしょう。

## 19.3.1.2 送信エラー割り込み (TXERIF)

送信エラー割り込みは、送信が中止された事を示すために使います。送信中止は以下のいずれかの条件で発生します。

1. 特定のパケットの送信中に 16 回以上のコリジョンが発生した場合
2. レイトコリジョン (パケットの 64 バイトが送信された後のコリジョン) が発生した場合
3. 媒体が占有されたままであったためパケットを送信できず、遅延リミットに達しており、かつ DEFER ビット (MACON4<6>) がクリアされている場合

4. HFRMEN ビット (MACON3<2>) をセットせず、個別パケット制御バイトの POVERRIDE ビットと PHUGEEN ビットもセットせずに、MAMXFL レジスタで定義された最大フレーム長を超えるパケットを送信しようとした場合
5. Ethernet バッファのメモリ帯域幅が不足しており、10 Mbit/s の転送レートを維持できなかった場合 (バッファ アンダーラン)

これらいずれかのエラー条件が発生すると、TXERIF フラグビットが「1」にセットされます。一旦セットされた TXERIF ビットは、ファームウェアまたはリセット条件によってのみクリアされます。送信エラー割り込みを有効に (TXERIF ビットと ETHIE ビットの両方をセット) している場合、Ethernet 割り込みが生成されます。送信エラー割り込みを無効に (TXERIF ビットと ETHIE ビットのどちらかをクリア) している場合、アプリケーションで TXERIF ビットをポーリングして適切に対応する事ができます。割り込みの処理が完了したら、フラグビットをクリアしてください。

送信が中止されると TXRTS ビット (ECON1<3>) がクリアされ、TXABRT ビット (ESTAT<1>) がセットされ、送信ステータスペクタが (ETXND レジスタ+1) のアドレスに書き込まれます。MAC は自動的にパケットの再送信を試みません。アプリケーションで送信ステータスペクタと BUFFER ビットを読み出し、送信中止の原因を調べます。問題と解決方法を特定したら、次の送信中止を検出できるように、アプリケーションで BUFFER ビット (セットされている場合) と TXABRT ビットをクリアします。

全二重モードで送信エラー割り込みの要因となるのは、上記の条件 4 と 5 の場合のみです。BUFFER ビットもセットされている場合、条件 5 であると判定できます。コリジョン等、ネットワークの共有に関連する問題は全二重ネットワークでは発生しません。送信エラー割り込みを引き起こすエラー条件は、送信割り込み条件と一致します。このため、送信エラー割り込みが発生すると TXIF ビットも同時にセットされます。

## 19.3.1.3 送信割り込み (TXIF)

送信割り込みは、要求されたパケット送信が完了した (TXRTS ビットが「1」から「0」に遷移した) 事を示すために使います。送信の完了、中止、アプリケーションによる送信キャンセルがあると、TXIF フラグビットは「1」にセットされます。TXRTS ビットをアプリケーションでクリアしておらず、なおかつ TXABRT ビットがセットされていないければ、パケットは正常に送信されています。一旦セットされた TXIF ビットは、ソフトウェアまたはリセット条件によってのみクリアされます。送信割り込みを有効に (TXIF ビットと ETHIE ビットの両方をセット) している場合、割り込みが生成されます。送信割り込みを無効に (TXIF ビットと ETHIE ビットのどちらかをクリア) した場合は、アプリケーションで TXIF ビットをポーリングして適切に対応する事ができます。

# PIC18F97J60 ファミリ

## 19.3.1.4 リンクステータス変化割り込み (LINKIF)

LINKIF はリンクステータスが変化した事を示します。実際のリンクステータスは、LLSTAT ビット (PHSTAT1<2>) または LSTAT ビット (PHSTAT2<10>) から取得できます (レジスタ 19-10 とレジスタ 19-12 参照)。他の割り込み要因とは異なり、リンクステータス変化割り込みは PHY モジュールで生成されるため、有効にするには特別な手順が必要です。

リセット時の既定値では、LINKIF がセットされる事は決してありません。この割り込みを受けるには、PLNKIE ビットと PGEIE ビットの両方をセットする必要があります。この割り込みを有効にすると、LINKIF ビットには PGIF ビットの内容がそのまま格納されます (シャドー)。PHY は 1 つしか割り込みをサポートしないため、PGIF ビットは常に PLNKIF ビットと同じ値です (2 つの PHY イネーブルビットを両方セットした場合)。

セットされた LINKIF ビットは、ソフトウェアまたはリセットによってのみクリアされます。リンクステータス変化割り込みを有効に (LINKIE、PLNKIE、PGEIE、ETHIE を全てセット) している場合、割り込みが生成されます。リンクステータス変化割り込みを無効に (LINKIE、PLNKIE、PGEIE、ETHIE のいずれかをクリア) している場合、ユーザアプリケーションで PLNKIF フラグビットをポーリングして適切に対応する事ができます。

LINKIF ビットは読み出し専用です。PHY レジスタの読み出しには比較的長い時間がかかるため、アプリケーションで PLNKIE ビットと PGEIE ビットをセットして LINKIF フラグビットをポーリングするという方法もあります。PHIR レジスタを MII レジスタ経由で読み出すと LINKIF、PGIF、PLNKIF ビットが自動的にクリアされ、次のリンクステータス変化割り込みを検出できます。PHY レジスタへのアクセスの詳細は、[セクション 19.2.5 「PHY レジスタ」](#) を参照してください。

## 19.3.1.5 DMA 割り込み (DMAIF)

DMA 割り込みは、DMA モジュールがメモリコピーまたはチェックサム計算を完了 (DMAST ビットが「1」から「0」へ遷移) した事を示します。さらに、DMAST ビットを手動でクリアしてアプリケーションが DMA 動作をキャンセルした場合にこの割り込みが発生します。セットされた DMAIF ビットは、ファームウェアまたはリセット条件によってのみクリアされます。DMA 割り込みを有効にしている場合、Ethernet 割り込みが生成されます。DMA 割り込みを無効にしている場合、ユーザアプリケーションで DMAIF フラグステータスをポーリングして適切に対応する事ができます。処理が完了したら、このフラグビットをクリアしてください。

## 19.3.1.6 受信パケット保留割り込み (PKTIF)

受信パケット保留割り込みは、受信バッファに 1 つ以上のデータパケットがある事を示すもので、新しいパケットが到着した事を通知する手段として使います。受信バッファに少なくとも 1 つのパケットがあると、

PKTIF フラグビットがセットされます。つまり、この割り込みフラグは Ethernet パケットカウンタレジスタ (EPKTCNT) がゼロ以外の場合、常にセットされます。

受信パケット保留割り込みを有効に (PKTIE ビットと ETHIE ビットの両方をセット) している場合、新しいパケットを正常に受信して受信バッファに書き込んだ時に Ethernet 割り込みが生成されます。受信パケット保留割り込みを無効に (PKTIE ビットと ETHIE ビットのいずれかをクリア) している場合、ユーザアプリケーションで PKTIF ビットをポーリングして適切に対応する事ができます。

PKTIF ビットは、ソフトウェアで EPKTCNT レジスタを「0」にデクリメントして間接的にクリアするか、リセット条件でのみクリアできます。EPKTCNT レジスタのクリアについての詳細は、[セクション 19.5.3 「パケットの受信」](#) を参照してください。受信バッファ内にある最後のデータパケットの処理が完了すると、EPKTCNT がゼロになり、PKTIF ビットは自動的にクリアされます。

## 19.3.2 Ethernet 割り込みと Wake-On-LAN

Ethernet モジュールの割り込み構造は、Magic Packet を使った Wake-On-LAN を実装しています。これにより、アイドル時にアプリケーションが消費する電力を低減し、特定のパケット (Magic Packet) を受信したらフルパワー動作に復帰させる事ができます。

Wake-On-LAN を使うには、Ethernet モジュールをが常時有効である必要があります。また、Magic Packet を選択できるように受信フィルタを設定しておく必要もあります。フィルタの設定方法の詳細は、[セクション 19.8 「受信フィルタ」](#) を参照してください。

Wake-On-LAN を使うには、以下の手順でマイクロコントローラを設定します。

1. Ethernet モジュールが有効で通常動作中に、CRC ポストフィルタと Magic Packet フィルタを有効にする (ERXFCN<5,3> = 1)。
2. Ethernet バッファ内に残っている全てのパケットの処理を完了する。
3. マイクロコントローラ レベルで Ethernet 割り込みを有効にし (PIE2<5> = 1)、モジュールレベルで受信パケット保留割り込みを有効にする (EIE<6> = 1)。
4. マイクロコントローラを PRI\_IDLE モードにする (プライマリクロック源を選択した状態で OSCCON<7> = 1 とし、SLEEP 命令を実行する)。

この設定で Magic Packet を受信すると、受信パケット保留割り込みが発生します。これにより、マイクロコントローラは割り込みから復帰します。

## 19.4 モジュールの初期化

Ethernet モジュールでパケットを送受信する前に、一部のデバイス設定を初期化する必要があります。アプリケーションによっては、一部の設定を変更する必要があります。通常、これらの作業はリセット後に1回行っておけば、それ以降は変更の必要はありません。

以下に説明する設定作業を行う前に、ETHEN ビット (ECON2<5>) をセットして Ethernet モジュールを有効にしておく事を推奨します。こうすると、PHYRDY フラグがセットされるまでの待ち時間をなくす事ができます。

### 19.4.1 受信バッファ

パケットを受信する前に、ERXST および ERXND ポインタを設定して受信バッファを初期化する必要があります。ERXST のアドレスから ERXND のアドレスまでのメモリ空間は全て受信ハードウェア専用です。ERXST ポインタには必ず偶数アドレスを書き込み、ERXND ポインタには必ず奇数アドレスを書き込む必要があります。

大容量のデータと高頻度のパケット受信が予測されるアプリケーションでは、メモリのほとんどを受信バッファに割り当てると良いでしょう。一方、古いパケットを保存しておきたい場合、または複数のパケットを送信バッファに格納しておきたい場合は受信バッファの割り当て量を少なくします。

ERXST または ERXND ポインタに書き込むと、ERXST レジスタの値が自動的に ERXWRPT ポインタレジスタに書き込まれます。受信ハードウェアが受信データの書き込みを開始する時、ERXWRPT レジスタに格納されたアドレスを先頭アドレスとして使います。ERXST および ERXND ポインタを初期化すると、さらに ERXRDPT レジスタに ERXND の値が書き込まれます。ERXRDPT レジスタペアに書き込みを行う場合、ERXRDPTL、ERXRDPTH の順に書き込みを行うようにします。詳細は、[セクション 19.5.3.3「受信バッファ空間の解放」](#)を参照してください。

### 19.4.2 送信バッファ

受信バッファで使っていないメモリは全て送信バッファと見なされます。送信待ちデータは、未使用領域に書き込みます。しかし、パケットの送信が完了すると、ハードウェアはパケットの最終バイトの後のメモリアドレスに7バイトのステータスペクタを書き込みます。従って、送信パケットの末尾と受信バッファの先頭の間には少なくとも7バイトの領域を空けておく必要があります。

### 19.4.3 受信フィルタ

受信フィルタは、ERXFCON レジスタに書き込む事によって有効/無効にします。設定方法は[セクション 19.8「受信フィルタ」](#)を参照してください。

### 19.4.4 PHY 起動タイマの遅延

ETHEN ビットを「1」にセットして Ethernet モジュールを有効にした直後に初期化手順を実行する場合、PHY レジスタを変更する前に 1 ms の遅延時間が必要なため、PHYRDY ビットをポーリングして確認する必要があります。PHY 起動タイマの詳細は、[セクション 19.1.3.1「起動タイマ」](#)を参照してください。

### 19.4.5 MAC の初期化設定

一部の MAC レジスタは、初期化中に設定が必要です。この操作は初期化時に1回だけ行います。書き込みの順序は問いません。

1. MARXEN ビット (MACON1<0>) をセットして MAC がフレームを受信できるようにする。全二重モードを使う場合、ほとんどのアプリケーションで TXPAUS ビットと RXPAUS ビットもセットして、IEEE で定義されたフロー制御を有効にしておく。
2. MACON3 レジスタの PADCFG<2:0>、TXCRCEN、FULDPX ビットを設定する。ほとんどのアプリケーションでは、自動パディングを有効にしてフレーム長を 60 バイト以上にし、常に有効な CRC を付加する。多くのアプリケーションでは、FRMLNEN ビットもセットしてフレーム長を取得すると便利である。アプリケーションを全二重モードに設定されたりリモートノードに接続する場合、FULDPX ビットをセットする。それ以外の場合、クリアしたままにする。
3. MACON4 のビットを設定する。IEEE 802.3 に準拠するため、DEFER ビット (MACON4<6>) を確実にセットする。
4. 送受信を許可する最大フレーム長を MAMXFL レジスタに書き込む。通常のネットワーク ノードは 1518 バイト以下のパケットを処理するよう設計されている。これより大きいパケットは IEEE 802.3 ではサポートされていない。
5. MAC連続パケット間ギャップレジスタ (MABBIPG) に 15h(全二重モードの場合) または 12h(半二重モードの場合) を書き込む。パケット間ギャップの設定方法の詳細は、[レジスタ 19-18](#) を参照してください。
6. MAC 不連続パケット間ギャップレジスタ下位バイト (MAIPGL) に 12h を書き込む。
7. 半二重モードの場合、MAC 不連続パケット間ギャップレジスタ上位バイト (MAIPGH) に 0Ch を書き込む。
8. MAADR1:MAADR6 レジスタにローカル MAC アドレスを書き込む。

# PIC18F97J60 ファミリ

## レジスタ 19-18: MABBIPG: MAC 連続パケット間ギャップレジスタ

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	BBIPG6	BBIPG5	BBIPG4	BBIPG3	BBIPG2	BBIPG1	BBIPG0
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

bit 7      **未実装:** 「0」として読み出し

bit 6-0      **BBIPG<6:0>:** 連続パケット間ギャップ遅延時間ビット

**FULDPX (MACON3<0>) = 1 の場合:**

パケットを連続して送信する際の、1つのパケット送信終了から次のパケット送信開始までのニブル時間オフセット遅延を示す。「所定の遅延時間 (単位はニブル時間) - 3」の値をレジスタに書き込む。推奨設定は 15h で、この値は IEEE で規定されたパケット間ギャップ (IPG) の最小値である 9.6  $\mu$ s に相当する

**FULDPX (MACON3<0>) = 0 の場合:**

パケットを連続して送信する際の、1つのパケット送信終了から次のパケット送信開始までのニブル時間オフセット遅延を示す。「所定の遅延時間 (単位はニブル時間) - 6」の値をレジスタに書き込む。推奨設定は 12h で、この値は IEEE で規定されたパケット間ギャップ (IPG) の最小値である 9.6  $\mu$ s に相当する

### 19.4.6 PHY の初期化設定

アプリケーションによっては、3つのPHYモジュールレジスタのビットも設定する必要があります。

PDPXMD ビット (PHCON1<8>) はPHYの全二重/半二重モードの設定を制御します。FULDPX ビット (MACON3<0>) の設定に合わせて、このビットをアプリケーションで正しく設定する必要があります。

HDLDIS ビット (PHCON2<8>) はデータの自動ループバックを無効にします。正しく動作させるには、HDLDIS ビットとRXAPDIS ビット (PHCON2<4>) を常にセットしておきます。

PHYレジスタのPHLCON ([レジスタ 19-13](#)) はLEDAとLEDBの出力を制御します。アプリケーションで既定値以外のLEDの設定が必要な場合、このレジスタを変更して対応します。LEDの動作の設定は、[セクション 19.1.2 「LEDの設定」](#)を参照してください。

### 19.4.7 Ethernet モジュールの無効化

状況によっては、Ethernetモジュールを長時間使う必要のない場合があります。例えば、ある特定のイベントが発生した時だけEthernetパケットを送受信したい場合です。このような場合、Ethernetモジュールだけを選択的にOFFにすることができます。

Ethernetモジュールだけを無効にする手順は以下の通りです。

1. RXEN ビットをクリアしてパケット受信を無効にする。
2. RXBUSY ビット (ESTAT<2>) をポーリングして現在受信中のパケットが終了するまで待つ。このビットがクリアになったら次の手順に進む。
3. 現在送信中のパケットが終了し、TXRTS ビット (ECON1<3>) がクリアになるのを待つ。
4. ETHEN ビットをクリアする。これでEthernetモジュールへの電源とクロックの供給が停止し、PHYレジスタにアクセスできなくなる。PHYRDY ビットも自動的にクリアされる。

## 19.5 データの送受信

Ethernet プロトコル (IEEE 802.3) では、フレームベースの 10 Mbps シリアル通信システムが詳細に記述されています。Ethernet モジュールの使い方を説明する前に、代表的な Ethernet データフレームの構造を簡単に説明しておきます。なお、IEEE 802.3 の基礎知識がある事を前提とします。さらに詳しい情報が必要な場合、正式な規格文書または Ethernet に関する解説書で、より詳細な説明を参照してください。

### 19.5.1 パケットのフォーマット

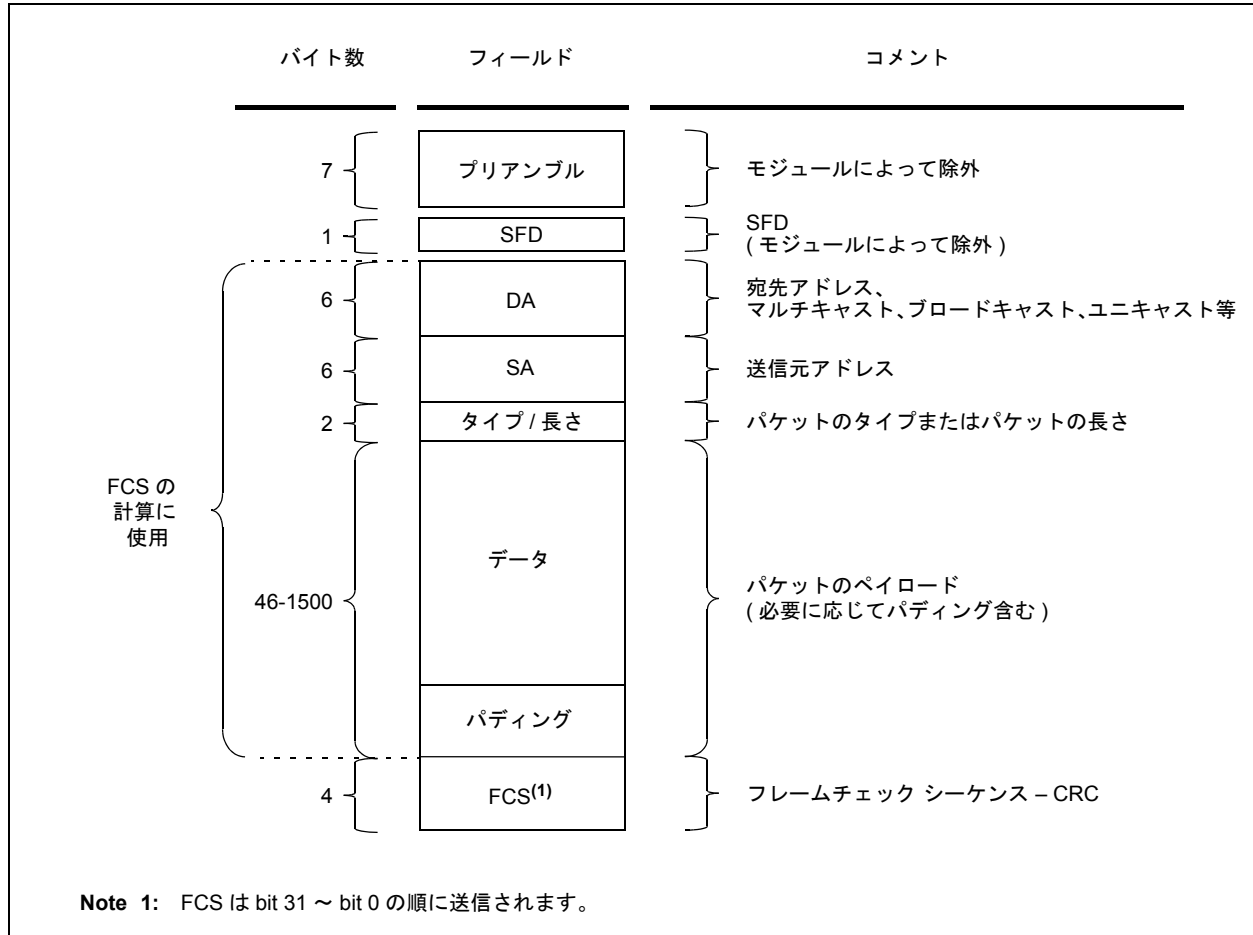
IEEE 802.3 に準拠した Ethernet フレームは、長さ 64 ~ 1518 バイトです。各フレームは、宛先 MAC アドレス、送信元 MAC アドレス、タイプ/長さフィールド、データペイロード、パディング フィールド (オプション)、CRC チェックの 5 つ (オプションを含めると 6 つ) のフィールドで構成されます。また、Ethernet 媒体上で送信を行う際は 7 バイトのプリアンブルフィー

ルドと 1 バイトの SFD (Start Frame Delimiter) が Ethernet パケット先頭に付加されます。このため、ツイストペア ケーブル上を流れるトラフィックの構造は図 19-8 に示す通りです。

#### 19.5.1.1 プリアンブル/SFD

Ethernet モジュールを使ってデータを送信する際にはプリアンブルと SFD の各バイトが自動的に生成され、受信時にパケットから自動的に取り除かれます。また、送信時には CRC フィールドとパディング (必要な場合) が自動的に生成され、受信時には CRC データが検証されます。ユーザアプリケーションはこれらのフィールドを作成または処理する必要はなく、明示的に CRC データを検証する必要もありません。しかし、パディングおよび CRC フィールドはパケット受信時に受信バッファに書き込まれるため、必要に応じてユーザアプリケーションで評価することができます。

図 19-8: Ethernet パケットのフォーマット



# PIC18F97J60 ファミリ

## 19.5.1.2 宛先アドレス (DA)

宛先アドレス (DA) は 6 バイトのフィールドで、パケットの送信先デバイスの MAC アドレスを格納します。MAC アドレスの先頭バイトの最下位ビットをセットした場合、宛先アドレスはマルチキャストです。例えば、01-00-00-00-F0-00 と 33-45-67-89-AB-CD はマルチキャスト アドレスで、00-00-00-00-F0-00 と 32-45-67-89-AB-CD はそうではありません。

宛先アドレスがマルチキャストのパケットは、特定の Ethernet ノードグループが受信する事を目的とし、そのノードグループで意味を持ちます。宛先アドレスに予約済みのマルチキャストアドレスである FF-FF-FF-FF-FF-FF を指定した場合、そのパケットをブロードキャストパケットと呼び、同じネットワークに接続している全てのノードが宛先です。MAC アドレスの先頭バイトの最下位ビットをクリアした場合、宛先アドレスはユニキャストで、指定した単一アドレスのノードのみが宛先です。

本モジュールは、宛先アドレスがマルチキャスト、ブロードキャスト、ユニキャストかによってパケットを破棄する受信フィルタを備えています。パケットを送信する際は、アプリケーションで所定の宛先アドレスを送信バッファに書き込む必要があります。

## 19.5.1.3 送信元アドレス (SA)

送信元アドレス (SA) は 6 バイトのフィールドで、Ethernet パケットを作成したノードの MAC アドレスを格納します。Ethernet モジュールを使う場合、各マイクロコントローラに一意の MAC アドレスを生成する必要があります。

MAC アドレスは 2 つの部分から成ります。最初の 3 バイトはベンダ識別子 (OUI) と呼ばれます。OUI は IEEE が割り当てています。最後の 3 バイトは、OUI を購入したベンダが自由に割り当てる事のできるアドレスバイトです。

パケットを送信する際は、アプリケーションで送信元に割り当てられた MAC アドレスを送信バッファに書き込む必要があります。MAADR レジスタ (ユニキャスト受信フィルタで使用) の内容を Ethernet モジュールが自動的に送信する事はありません。

## 19.5.1.4 タイプ / 長さ

タイプ / 長さフィールドは 2 バイトのフィールドで、後続のパケットデータがどのプロトコルに属するかを定義します。ただしこのフィールドに 05DCh (1500) 以下の数値を書き込んだ場合は長さフィールドと見なされ、後続するデータフィールド内のパディングを除いたデータの長さを表します。独自規格のネットワークを実装する場合はこのフィールドを長さフィールドとして扱い、IP (Internet Protocol) や ARP (Address Resolution Protocol) 等のプロトコルを実装するアプリケーションではこのフィールドに各プロトコルの規格で定義された適切なタイプを書き込んでからパケットを送信します。

## 19.5.1.5 データ

データフィールドは 0 ~ 1500 バイトの任意の長さをとることができます。これよりも長いデータパケットは Ethernet 規格に違反しており、ほとんどの Ethernet ノードはこのようなパケットを破棄します。ただし本モジュールは、ジャンボフレーム イネーブルビット (HFRMEN) をセット (MACON3<2> = 1) するとこれより大きなパケットの送受信が可能です。

## 19.5.1.6 パディング

パディングフィールドは、データペイロードが短い場合に IEEE 802.3 規格の要件を満たすために付加する可変長のフィールドです。DA、SA、タイプ、データ、パディングを合計した Ethernet パケットは 60 バイト以上である必要があります。4 バイトの必須フィールドである CRC を付加すると少なくとも 64 バイト必要です。データフィールドの長さが 46 バイト未満の場合、パディングフィールドが必要です。

PADCFG<2:0> ビット (MACON3<7:5>) の設定によっては、パケット送信時に Ethernet モジュールが自動的にゼロパディングを生成します。それ以外の場合、パケット送信前にユーザアプリケーションでパディングを付加する必要があります。アプリケーションでサイズの小さいパケットの送信を指示した場合、Ethernet モジュールはそのまま送信します。

本モジュールは、18 バイト未満のパケット受信を拒否します。18 バイト以上のパケットには通常の実受信フィルタを適用し、通常のトラフィックとして受け入れられます。本モジュールが拒否するのは 18 バイト未満のパケットのみのため、IEEE 802.3 規格の要件を満たすためには、ファームウェアで全ての受信パケットサイズをチェックして、64 バイト未満のパケットを拒否する必要があります。

## 19.5.1.7 CRC

CRC フィールドは、DA、SA、タイプ、データ、パディングの各フィールドのデータから計算した業界標準の 32 ビット CRC 値を格納する 4 バイトのフィールドです。このフィールドを使って Ethernet フレームの破損と、パケットコリジョンまたは他のホストの送信中止によるジャンクデータフラグメントを検出します。

Ethernet モジュールは、パケット受信時に各受信パケットの CRC をチェックします。CRCEN ビットがセットされている場合、CRC の無効なパケットは自動的に破棄されます。CRCEN ビットがクリアされている場合、パケットは他の全ての受信フィルタ基準を満たした場合に受信バッファに書き込まれます。CRC が有効だったかどうかは、アプリケーションで受信ステータスペクタを読み出して判定できます ([セクション 19.5.3「パケットの受信」](#) 参照)。

PADCFG<2:0> ビットの設定によっては、パケット送信時に Ethernet モジュールが自動的に有効な CRC を生成して送信します。それ以外の場合、ユーザアプリケーションで CRC を生成して送信バッファに格納する必要があります。CRC 計算は複雑なため、Ethernet モジュールで自動的に計算して CRC を付加する事を強く推奨します。



## 19.5.2 パケットの送信

Ethernet モジュールの MAC は、パケット送信時に自動的にプリアンブルと SFD フィールドを生成します。また、設定によってはパディング(必要な場合)と CRC も生成できます。それ以外の全てのフレーム フィールドはアプリケーションで生成して送信バッファメモリに書き込む必要があります。

また、本モジュールでは各送信パケットの先頭に個別パケット制御バイト(1バイト)を付加する必要があります。図 19-9 に、この個別パケット制御バイトの構成を示します。パケットを送信する前に、MAC レジスタをセクション 19.4「モジュールの初期化」の説明に従って初期化しておく必要があります。MAC レジスタは送信の特性を変更します。

図 19-9: 個別パケット制御バイトのフォーマット

	—	—	—	—	PHUGEEN	PPADN	PCRCEN	POVERRIDE
bit 7								bit 0
bit 7-4	<b>未実装:</b> 「0」として読み出し							
bit 3	<b>PHUGEEN:</b> 個別パケット ジャンボフレーム イネーブルビット							
	<b>POVERRIDE = 1 の場合:</b>							
	1 = パケット全体を送信する							
	0 = MAMXFL レジスタで指定したバイト数までを MAC で送信する。指定したバイト数よりもパケットが長い場合、MAMXFL レジスタで指定したバイト数に達した時点で送信を中止する							
	<b>POVERRIDE = 0 の場合:</b>							
	このビットを無視する							
bit 2	<b>PPADN:</b> 個別パケット パディング イネーブルビット							
	<b>POVERRIDE = 1 の場合:</b>							
	1 = パケットサイズが 60 バイト未満の場合、ゼロでパディングして 60 バイトにする							
	0 = パディングバイトを付加せずにパケットを送信する							
	<b>POVERRIDE = 0 の場合:</b>							
	このビットを無視する							
bit 1	<b>PCRCEN:</b> 個別パケット CRC イネーブルビット							
	<b>POVERRIDE = 1 の場合:</b>							
	1 = 有効な CRC を計算してフレームに付加する							
	0 = CRC を付加しない。フレーム末尾の 4 バイトを CRC として誤りチェックを行う							
	<b>POVERRIDE = 0 の場合:</b>							
	このビットを無視する							
bit 0	<b>POVERRIDE:</b> 個別パケット優先ビット							
	1 = MACON3 レジスタで定義した設定値ではなく、PCRCEN、PPADN、PHUGEEN ビットの値を優先する							
	0 = MACON3 レジスタの設定値でパケットの送信方法を決定する							

# PIC18F97J60 ファミリ

図 19-10 に、組み立てられた送信パケット全体のメモリ内での構造の例を示します。パケットは、以下の方法で構成して送信します。

1. ETXST ポインタにバッファ内の適切な未使用アドレスを書き込む。このアドレスは個別パケット制御バイトの位置を示す (この例では 0120h)。ETXST ポインタには偶数アドレスを指定する事が推奨される。
2. EDATA および EWRPT レジスタを使ってパケットデータを Ethernet バッファに順番に書き込む。データは、個別パケット制御バイト、DA、SA、タイプ/長さ、データペイロードの順に書き込む。
3. ETXND ポインタにデータペイロードの最後のバイトのアドレスを書き込む (この例では 0156h)。
4. TXIF フラグビット (EIR<3>) をクリアし、TXIE ビット (EIE<3>) と ETHIE ビットをセットして、完了時の割り込みを有効にする (必要な場合)。
5. TXRTS ビット (ECON1<3>) をセットして送信プロセスを開始する。

TXRTS ビットをセットした時に DMA 動作が進行中の場合、Ethernet モジュールはその DMA 動作が完了するまで待ってからパケットの送信を試みます。このような遅延が必要なのは、DMA と送信エンジンが同じメモリアービタ チャンネルを共有しているためです。同

様に、TXRTS がセットされている状態で DMAST ビットをセットした場合、DMA は TXRTS ビットがクリアされるのを待ってから処理を開始します。

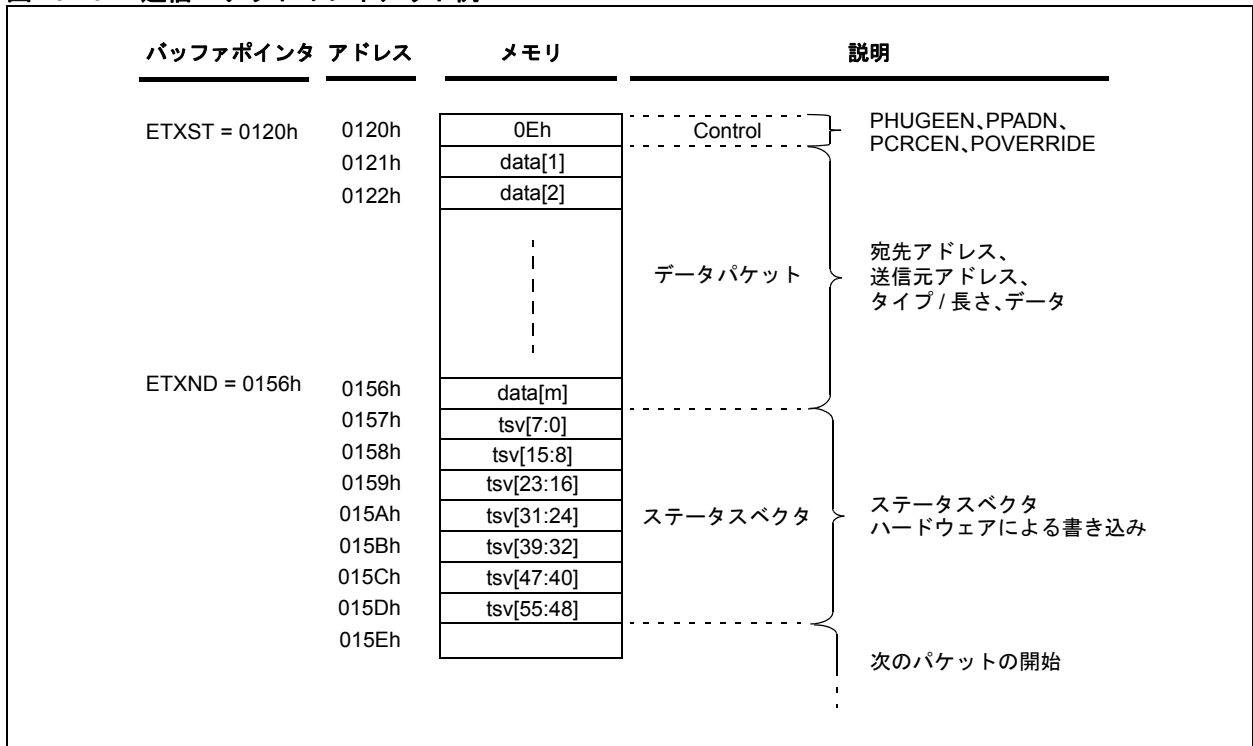
送信が進行中は、ETXST および ETXND ポインタの値を変更しないでください。送信をキャンセルするには、TXRTS ビットをクリアします。

パケットの送信が完了した場合、またはエラー / キャンセルによって送信が中止された場合、以下の処理が実行されます。

- TXRTS ビットがクリアされる
- 7 バイトの送信ステータスペクタが、「ETXND ポインタ + 1」で示されるアドレスのバッファに書き込まれる
- TXIF フラグビットがセットされる
- 割り込みが生成される (割り込みを有効にしている場合)
- ETXST および ETXND ポインタの値は変更されない

パケットが正常に送信されたかどうかをチェックするには、TXABRT ビットを読み出します。このビットがセットされている場合、BUFFER ビットと、送信ステータスペクタの各フィールドをポーリングして原因を特定します。表 19-4 に、送信ステータスペクタの構成を示します。複数バイトのフィールドはリトルエンディアン方式で書き込まれます。

図 19-10: 送信パケットのレイアウト例



# PIC18F97J60 ファミリ

表 19-4: 送信ステータスペクタ

Bit	フィールド	説明
55-52	ゼロ	0
51	VLAN タグ付きフレーム送信	フレームのタイプ/長さフィールドに VLAN プロトコル識別子の 8100h が含まれていた
50	バックプレッシャ対応	予約済み、使用不可
49	PAUSE 制御フレーム送信	送信されたフレームは有効な PAUSE オペコードを持つ制御フレームであった
48	制御フレーム送信	送信されたフレームは制御フレームであった
47-32	媒体に送信される総バイト数	現在のパケットでケーブルに送信された総バイト数 (コリジョンによる再送バイトも全て含む)
31	送信アンダーラン	バッファメモリの帯域幅が不足して 10 Mbps の送信レートを維持できなかったため送信が中止された
30	ジャンボフレーム送信	フレームのバイト数が MAMXFL レジスタの値を超えた
29	送信時のレートコリジョン	64 バイトを送信した後にコリジョンが発生した
28	送信コリジョン回数オーバー	コリジョンの回数が再送の上限である 15 回を超えたためパケット送信が中止された
27	送信延期時間オーバー	媒体が占有されたままであったため、パケットの送信延期時間が 24,287 ビット時間 (2.4287 ms) を超えた
26	パケット送信延期	パケットの送信が少なくとも 1 回延期された (ただし延期時間の上限は超えていない)
25	ブロードキャスト送信	パケットの宛先アドレスがブロードキャストアドレスだった
24	マルチキャスト送信	パケットの宛先アドレスがマルチキャストアドレスだった
23	送信完了	パケットの送信が正常に完了した
22	送信長さ上限オーバー	フレームのタイプ/長さフィールドが 1500 バイトを超えた (タイプフィールド)
21	送信長さチェックエラー	パケットフレームの長さフィールドの値が実際のデータバイトの長さとは一致しておらず、しかもタイプフィールドでない。このエラーを捕捉するには、FRMLNEN ビット (MACON3<1>) をセットする必要がある
20	送信 CRC エラー	パケットに付加された CRC が内部生成した CRC と一致しなかった
19-16	送信コリジョン回数	現在のパケット送信中に発生したコリジョンの回数を示す。正常に送信されたパケットのみが対象のため、コリジョン回数の最大値は 16 になる事はない。
15-0	送信バイト数	フレームの総バイト数 (コリジョンのバイト数を除く)

# PIC18F97J60 ファミリ

## 19.5.3 パケットの受信

受信バッファが初期化され、MAC と受信フィルタが正しく設定されていれば、アプリケーションは以下の手順で Ethernet パケットを受信します。

1. PKTIE および ETHIE ビットをセットして、パケット受信時に Ethernet 割り込みを生成する (必要な場合)。
2. RXERIF フラグビットをクリア、RXERIE ビットと ETHIE ビットをセットし、バッファ空き領域またはメモリアクセス帯域幅の不足によるパケット取りこぼし時に割り込みを生成する(必要な場合)。
3. RXEN ビット (ECON1<2>) をセットして受信を有効にする。

RXEN をセットした後は、全二重 / 半二重モードの設定と、受信バッファ開始 / 終了ポイントの値を変更してはいけません。また、予期しないパケットの到着を防ぐため、受信フィルタ設定 (ERXFCON) と MAC アドレスを変更する前に、RXEN ビットをクリアする事を推奨します。

受信を有効にすると、フィルタを通過したパケットは環状受信バッファに書き込まれます。フィルタ条件を満たさないパケットは破棄されます。パケットが破棄

された事をアプリケーション側で知る手段はありません。パケットが受け入れられ、バッファへの書き込みが完了すると以下の処理が実行されます。

- EPKTCNT レジスタがインクリメントする。
- PKTIF ビットがセットされる。
- 割り込みが生成される (割り込みを有効にしている場合)。
- ハードウェア書き込みポイントの ERXWRPT が自動的に進められる。

### 19.5.3.1 受信パケットのレイアウト

図 19-11 に、受信パケットのレイアウトを示します。パケットの前には 6 バイトのヘッダが付加されます。このヘッダは、Next Packet ポインタと受信ステータスペクタ (パケットサイズ等、受信に関する情報を記録したもの) で構成されます。表 19-5 に受信ステータスペクタを示します。

パケット末尾のバイトが奇数アドレスで終了している場合、ハードウェア書き込みポイントを進める際にハードウェアによって自動的にパディングバイトが付加されます。このため、全てのパケットは偶数境界から開始します。

図 19-11: 受信パケットのレイアウト例

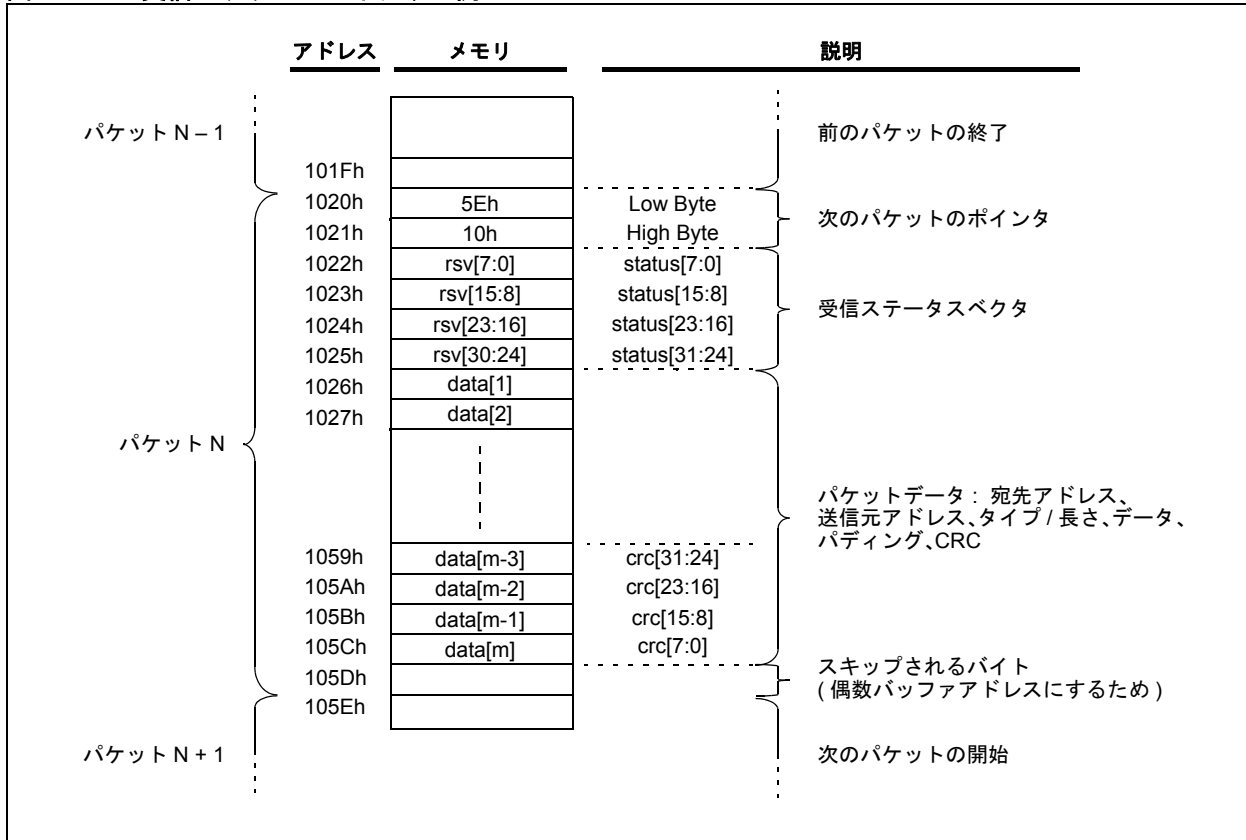


表 19-5: 受信ステータスペクタ

Bit	フィールド	説明
31	ゼロ	「0」
30	受信 VLAN タイプ検出	現在のフレームが VLAN タグ付きフレームとして認識された
29	不明なオペコード受信	現在のフレームが制御フレームとして認識されたが、不明なオペコードを含んでいた
28	PAUSE 制御フレーム受信	現在のフレームが有効な PAUSE フレーム オペコードと有効な宛先アドレスを含む制御フレームとして認識された
27	制御フレーム受信	現在のフレームが有効なタイプ/長さフィールドを持つ制御フレームとして認識された
26	ドリブルニブル	このパケットの終了後、さらに 1~7 ビットを受信した (これらの余分なビットは破棄された)
25	ブロードキャスト パケット受信	受信したパケットに有効なブロードキャスト アドレスが含まれていた
24	マルチキャスト パケット受信	受信したパケットに有効なマルチキャスト アドレスが含まれていた
23	受信 OK	パケットに有効な CRC があり、シンボルエラーが検出されなかった
22	長さ上限オーバー	フレームのタイプ/長さフィールドが 1500 バイトを超えた (タイプフィールド)
21	長さチェックエラー	パケット内のフレームの長さフィールドの値が実際のデータバイトの長さとは一致しない
20	CRC エラー	フレームの CRC フィールドの値が MAC で計算した CRC と一致しない
19	予約済み	
18	以前に観察されたキャリアイベント検出	最後の受信の後でキャリアイベントを検出した。そのキャリアイベントはこのパケットと関連していない。キャリアイベントとは、パケット受信の試みに至らない受信チャンネル上でのアクティビティを示す
17	予約済み	
16	ロングイベント/ドロップイベント	50,000 ビット時間を超えるパケットを受信した、または最後の受信の後パケットを取りこぼした
15-0	受信バイト数	受信したフレーム (DA、SA、タイプ/長さ、データ、パディング、CRC フィールド) の長さを示す。このフィールドはリトルエンディアン形式で格納される

### 19.5.3.2 受信パケットの読み出し

パケットを処理する際、アプリケーションは通常 Next Packet ポインタの先頭から読み出しを開始します。そして Next Packet ポインタと、受信ステータスペクタの必要なバイトを保存した後、実際のパケットの内容を読み出します。AUTOINC ビットがセットされている場合、ERDPT レジスタを書き換えなくてもパケット全体を順次読み出す事ができます。読み出しポインタは環状受信バッファの最後まで来ると自動的にバッファの先頭へラップアラウンドします。

パケットにランダムにアクセスするには、パケットが ERXND から ERXST へのバッファ境界をまたいでいる場合に受信バッファの末尾を超えないように注意しながら、適切な ERDPT レジスタをアプリケーションで計算する必要があります。つまり、パケットの開始アドレスと所定のオフセット値に基づき、式 19-1 に示すロジックで計算します。

### 式 19-1: ランダムアクセスの場合のアドレス計算

<p>If Packet Start Address + Offset &gt; ERXND, then  ERDPT = Packet Start Address + Offset - (ERXND - ERXST + 1)</p> <p>else:  ERDPT = Packet Start Address + Offset</p>
---

# PIC18F97J60 ファミリ

## 19.5.3.3 受信バッファ空間の解放

ユーザアプリケーションでパケット (またはパケットの一部) を処理した後、処理済みデータが占有しているバッファ空間を解放するには、受信バッファ読み出しポインタペアのERXRDPTを進める必要があります。Ethernet モジュールは ERXRDPT レジスタが指し示すメモリアドレスの直前までしか書き込みを行いません。ERXRDPT が示すアドレスへの上書きが試行されるとパケットの受信が中止され、RXERIF フラグがセットされて割り込みが生成されます (割り込み有効時)。このようにして、ハードウェアは未処理のパケットを決して上書きしないようにしています。通常、ERXRDPT ペアは Next Packet ポインタが指し示す値の近くまで進みます。Next Packet ポインタは、現在のパケットの受信ステータスペクタの前にあります。

受信バッファ読み出しポインタ下位バイト (ERXRDPTLレジスタ)への書き込みは内部でバッファされ、1バイトのみの更新ではポインタは進みません。ERXRDPT ペアを移動するには、アプリケーションで ERXRDPTL から先に書き込む必要があります。この書き込みによって内部バッファが更新されますが、レジスタは変更されません。次にアプリケーションが ERXRDPTHへの書き込みを実行した時点で、内部バッファに格納された下位バイトが ERXRDPTL レジスタに書き込まれます。ERXRDPT バイトはどちらの順番でも読み出す事ができます。これらのバイトを読み出すと、レジスタの実際の値が返されます。このため、内部バッファに格納された下位バイトは読み出す事ができません。

各パケットの処理が完全に終了したら ERXRDPT を進めると同時に、アプリケーションで PKTDEC ビット (ECON2<6>) をセットする必要があります。これにより EPKTCNT レジスタが1つデクリメントします。デクリメント後、EPKTCNT ビットが「0」の場合、PKTIF フラグビットが自動的にクリアされます。それ以外の場合、PKTIF フラグビットはセットされたままで、受信バッファにまだパケットがあり、処理を待っている事を示します。EPKTCNT = 0 の場合にさらにデクリメントしようとしても、255 にアンダーフローする事はありますが、予期しない割り込みが発生する場合があります。EPKTCNT = 0 の時に、アプリケーションで EPKTCNT をデクリメントしないでください。

また、EPKTCNT レジスタが最大値の 255 になると、バッファに空き領域があっても新しく受信したパケットは全て中止されます。このエラーを示すために、RXERIF ビットがセットされ、割り込みが発生します (割り込み有効時)。このような状況を防ぐには、パケット処理の際にユーザアプリケーションでカウンタを適切にデクリメントする必要があります。

### 式 19-2: 受信バッファの空き領域の計算

```
If ERXWRPT > ERXRDPT, then
    Free Space = (ERXND - ERXST) - (ERXWRPT - ERXRDPT)
else:
    if ERXWRPT = ERXRDPT, then
        Free Space = (ERXND - ERXST)
    else:
        Free Space = ERXRDPT - ERXWRPT - 1
```

バッファ領域の所有権を制御するポインタは1つしかないため、アプリケーションはパケットを受信した順に処理する必要があります。パケットを保存して後で処理する場合、アプリケーションでパケットを未使用のメモリ領域へコピーしてください。この処理は、内蔵 DMA コントローラを使うと効率的に行う事ができます ( [セクション 19.9「ダイレクトメモリアクセス \(DMA\) コントローラ」](#) 参照)。

## 19.5.3.4 受信バッファの空き領域

アプリケーションで受信バッファの空き領域を調べる必要がある場合、ハードウェア書き込みポインタ (ERXWRPT レジスタ) を読み出して ERXRDPT レジスタと比較します。既知である受信バッファサイズと組み合わせると、空き領域を求める事ができます。

**Note:** ERXWRPT レジスタが更新されるのはパケットの受信が正しく完了した時のみです。アプリケーションで ERXWRPT レジスタを読み出した直後に次のパケットの読み出しが正しく完了した場合は、返される値は最新のものではありません。最大フレーム長 (MAMXFLH:MAMXFLL) + 8) だけ先に進んでいる可能性があります。さらに、アプリケーションで ERXWRPT レジスタペアの片方のバイトを読み出している間に新しいパケットが到着すると、ERXWRPT レジスタペアのもう片方のバイトを読み出す前に13ビットのポインタ全体が更新される可能性があります。

受信ハードウェアを有効にした状態で ERXWRPT レジスタを読み出す際は、下位バイトと上位バイトを正しい組み合わせで読み出した事を確認する必要があります。

正しい組み合わせで取得したかどうかを確認する手順は以下の通りです。

1. EPKTCNT レジスタを読み出し、その内容を保存する。
2. ERXWRPTL と ERXWRPTH を読み出す。
3. EPKTCNT レジスタをもう一度読み出す。
4. 2つのパケット数を比較する。パケット数が一致しなければ手順2に戻る。

ハードウェア書き込みポインタの値を取得したら、[式 19-2](#) で空き領域を求める事ができます。書き込みポインタを ERXRDPT レジスタが占める値と同じ値にする事はハードウェアで禁止されているため、バッファには常に少なくとも1バイトの未使用領域が残ります。[式 19-2](#) には、この1バイトの未使用領域が反映されています。

# PIC18F97J60 ファミリ

表 19-6: パケット送信に関連するレジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	リセット値の 記載ページ
EIE	-	PKTIE	DMAIE	LINKIE	TXIE	-	TXERIE	RXERIE	73
EIR	-	PKTIF	DMAIF	LINKIF	TXIF	-	TXERIF	RXERIF	73
ESTAT	-	BUFFER	-	r	-	RXBUSY	TXABRT	PHYRDY	73
ECON1	TXRST	RXRST	DMAST	CSUMEN	TXRTS	RXEN	-	-	70
ETXSTL	送信開始レジスタ下位バイト (ETXST<7:0>)								74
ETXSTH	-	-	-	送信開始レジスタ上位バイト (ETXST<12:8>)					74
ETXNDL	送信終了レジスタ下位バイト (ETXND<7:0>)								74
ETXNDH	-	-	-	送信終了レジスタ上位バイト (ETXND<12:8>)					74
MACON1	-	-	-	r	TXPAUS	RXPAUS	PASSALL	MARXEN	75
MACON3	PADCFG2	PADCFG1	PADCFG0	TXCRCEN	PHDREN	HFRMEN	FRMLNEN	FULDPX	75
MACON4	-	DEFER	r	r	-	-	r	r	75
MABBIPG	-	BBIPG6	BBIPG5	BBIPG4	BBIPG3	BBIPG2	BBIPG1	BBIPG0	75
MAIPGL	-	MAC 不連続パケット間ギャップレジスタ下位バイト (MAIPGL<6:0>)							75
MAIPGH	-	MAC 不連続パケット間ギャップレジスタ上位バイト (MAIPGH<6:0>)							74
MAMXFL	最大フレーム長レジスタ下位バイト (MAMXFL<7:0>)								74
MAMXFLH	最大フレーム長レジスタ上位バイト (MAMXFL<15:8>)								74

凡例: — = 未実装、r = 予約済みビット、網掛けの部分は使いません。

表 19-7: パケット受信に関連するレジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	リセット値の 記載ページ
EIE	-	PKTIE	DMAIE	LINKIE	TXIE	-	TXERIE	RXERIE	73
EIR	-	PKTIF	DMAIF	LINKIF	TXIF	-	TXERIF	RXERIF	73
ESTAT	-	BUFFER	-	r	-	RXBUSY	TXABRT	PHYRDY	73
ECON2	AUTOINC	PKTDEC	ETHEN	-	-	-	-	-	73
ECON1	TXRST	RXRST	DMAST	CSUMEN	TXRTS	RXEN	-	-	70
ERXSTL	受信開始レジスタ下位バイト (ERXST<7:0>)								74
ERXSTH	-	-	-	受信開始レジスタ上位バイト (ERXST<12:8>)					74
ERXNDL	受信終了レジスタ下位バイト (ERXND<7:0>)								74
ERXNDH	-	-	-	受信終了レジスタ上位バイト (ERXND<12:8>)					73
ERXRDPTL	受信バッファ読み出しポインタ下位バイト (ERXRDPT<7:0>)								73
ERXRDPTH	-	-	-	受信バッファ読み出しポインタ上位バイト (ERXRDPT<12:8>)					73
ERXFCON	UCEN	ANDOR	CRCEN	PMEN	MPEN	HTEN	MCEN	BCEN	74
EPKTCNT	Ethernet パケットカウント レジスタ								74
MACON1	-	-	-	r	TXPAUS	RXPAUS	PASSALL	MARXEN	75
MACON3	PADCFG2	PADCFG1	PADCFG0	TXCRCEN	PHDREN	HFRMEN	FRMLNEN	FULDPX	75
MAMXFL	最大フレーム長レジスタ下位バイト (MAMXFL<7:0>)								74
MAMXFLH	最大フレーム長レジスタ上位バイト (MAMXFL<15:8>)								74

凡例: — = 未実装、r = 予約済みビット、網掛けの部分は使いません。

## 19.6 全二重 / 半二重モードの設定と ネゴシエーション

本モジュールは全二重 / 半二重モードの自動ネゴシエーションをサポートしていません。全二重 / 半二重モードの自動ネゴシエーションに対応したネットワークスイッチまたはEthernetコントローラに接続すると、本モジュールは半二重デバイスとして検出されます。全二重モードで通信するには、本モジュールとリモートノード（スイッチ、ルータ、Ethernet コントローラ）を手動で全二重動作に設定する必要があります。

### 19.6.1 半二重動作

FULDPIX ビット (MACON3<0>) と PDPXMD ビット (PHCON1<8>) をクリア (= 0) すると、本モジュールは半二重モードで動作します。これら 2 つのビットのいずれか 1 つのみがセットされている場合、本モジュールは中間状態となり、正しく動作しません。モード切り換え時にはこの中間状態となる可能性があるため、切り換え中はアプリケーションでパケットの送受信を無効 (TXRTS ビットと RXEN ビットをクリア) しておく事を推奨します。

半二重モードでは、1 つの物理媒体上で一度に送信を行える Ethernet コントローラは 1 つだけです。別の Ethernet コントローラがパケットを送信中に、アプリケーションが TXRTS ビットをセットしてパケットの送信を要求しても、Ethernet モジュールはリモートのトランスミッタが停止するまで待機します。停止したら、Ethernet モジュールはパケットの送信を試みます。送信とほぼ同時に別の Ethernet コントローラが送信を開始すると、ケーブル上のデータが破損し、コリジョンが発生します。

コリジョン発生時のハードウェアの処理には 2 種類あります。64 バイトが送信される前にコリジョンが発生した場合、以下のイベントが発生します。

1. TXRTS ビットはセットされたままとなる
2. 送信エラー割り込みは発生しない
3. IEEE 802.3 規格で定義されたランダム エクスポンシヤル バックオフ遅延が経過する
4. パケットの先頭から再送を試みる (アプリケーションの介入は不要)

再送を 15 回試みてもさらにコリジョンが発生する場合、パケットの送信が中止され、TXRTS ビットがクリアされます。この場合、アプリケーションで適切な対策をとる必要があります。パケットが正しく送信されずに中止された事は、アプリケーションで TXABRT フラグビットを読み出す事によって判定できます。詳細は [セクション 19.5.2 「パケットの送信」](#) を参照してください。

64 バイトを送信した後にコリジョン (レイトコリジョン) が発生した場合、パケットの送信はただちに中止され、再送は行われません。通常、正しく設定された IEEE 802.3 準拠ネットワークではレイトコリジョンは発生しません。この問題を解決するには、ユーザの介入が必要です。この問題は、半二重媒体上で全二重ノードが送信を試みた場合に発生する事があります。あるいは、全二重ネットワークに接続した Ethernet モジュールが半二重モードで送信を試みた場合にこの問題が発生する事があります。ケーブルが長すぎる場合とネットワークの規模が大きすぎる場合にもレイトコリジョンが発生する事があります。

### 19.6.2 全二重動作

FULDPIX ビット (MACON3<0>) と PDPXMD ビット (PHCON1<8>) を両方セット (= 1) すると、本モジュールは全二重モードで動作します。これら 2 つのビットのいずれか 1 つのみがクリアされている場合、本モジュールは中間状態となり、正しく動作しません。前述の通り、全二重モードと半二重モードの切り換え時にはこの中間状態となる可能性があるため、切り換え中はアプリケーションでパケットの送信と受信を無効にしておく事を推奨します。

全二重モードでは、パケットの送信と受信を同時に実行できます。このため、パケット送信時にコリジョンが発生する事は一切ありません。



## 19.7 フロー制御

本モジュールは、全二重モードと半二重モードの両方でハードウェアフロー制御を実装しています。この機能の動作は、使うモードによって異なります。

### 19.7.1 半二重モード

半二重モードでは、FCEN0 ビット (EFLOCON<0>) をセットするとフロー制御が有効になります。FCEN0 ビットをセットすると、「1」と「0」を交互に繰り返すプリアンブルパターン (55h) が Ethernet 媒体に自動的に送信されます。ネットワークに接続されたノードは、このパターンを検出すると、何も送信せずにパターンの送信が終わるのを待つか、送信を試みてただちにコリジョンが発生させるか、いずれかの動作を行います。この状態では必ずコリジョンが発生するため、ネットワーク上のノードは互いに通信できず、新しいパケットは到着しません。

アプリケーションで TXRTS ビットをセットして Ethernet モジュールからのパケット送信を要求すると、プリアンブル パターンの送信が停止します。MABBIPG レジスタで設定したパケット間ギャップが経過すると、Ethernet モジュールはパケットの送信を試みます。他のノードも、このパケット間ギャップが経過したら送信を開始できます。ただし直前までトラフィックが輻輳していたため、複数のノードが送信を開始して多くのコリジョンが発生する事があります。Ethernet モジュールからのパケット送信が正常に完了または中止すると、プリアンブルパターンの送信が自動的に再開します。ネットワークの輻輳を解消したい場合、アプリケーションで FCEN0 ビットをクリアします。プリアンブルの送信が停止し、通常のネットワーク動作が再開します。

テスト済みで閉じたネットワーク環境以外で半二重フロー制御を使う事は推奨しません。ネットワークに悪影響を与えかねない上に効果がないためです。

### 19.7.2 全二重モード

全二重モード (MACON3<0> = 1) では、IEEE 802.3 規格で定義されている PAUSE 制御フレームの送信によるハードウェアフロー制御を実装しています。PAUSE 制御フレームは、予約済みのマルチキャストアドレス 01-80-C2-00-00-01、送信元アドレス (SA)、特別な PAUSE オペコード、2 バイトの PAUSE タイマ値、パディング/CRC で構成された 64 バイトのフレームです。

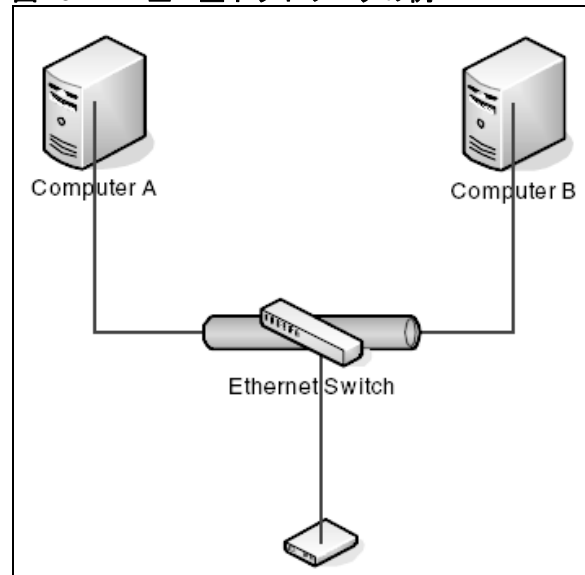
通常、MAC が PAUSE 制御フレームを受信すると、現在送信中のパケットが完了した時点で新しいフレームの送信を停止します。そして PAUSE 制御フレームから PAUSE タイマの値を取得し、この値を使って内部タイマを初期化します。このタイマは 512 ビット時間 (51.2  $\mu$ s) ごとに自動的にデクリメントします。タイマのカウントダウン中もパケットの受信は有効なままです。新しい PAUSE フレームが到着すると、タイマ

はその新しい PAUSE タイマの値で再び初期化されます。タイマがゼロになると (またはタイマ値がゼロの PAUSE フレームを受信すると)、その PAUSE フレームを受信した MAC は保留中のパケットの送信を再開します。PAUSE フレームによってネットワーク上の全てのトラフィックが停止するのを防ぐため、Ethernet スイッチおよびルータは全二重モードで PAUSE 制御フレームを伝搬しません。PAUSE フレームによる制御が適用されるのは、PAUSE フレームを直接受け取ったノードのみです。

図 19-12 にネットワークの例を示します。Computer A が全二重モードでマイクロコントローラベースのアプリケーションに大量のデータを送信している場合、Ethernet モジュールが PAUSE 制御フレームを送信してデータの送信を停止させる事ができます。Ethernet スイッチは、PAUSE フレームを受け取るとアプリケーションへのデータ送信を停止します。Computer A がさらにデータ送信を続けた場合、Ethernet スイッチは PAUSE タイマのカウント終了後に送信できるようにこれらのデータをバッファに格納します。Ethernet スイッチのバッファが足りなくなってくると、スイッチ自身が Computer A に PAUSE 制御フレームを送信します。

何らかの理由で Ethernet スイッチが PAUSE 制御フレームを生成しない場合、あるいはノードが受信した PAUSE フレームを正しく処理しない場合、パケットが失われます。いずれの場合にも、Computer A と Computer B の間の通信は全く影響を受けません。

図 19-12: 全二重ネットワークの例



# PIC18F97J60 ファミリ

全二重モードでフロー制御を有効にするには、MACON1レジスタのTXPAUSビットとRXPAUSビットをセットします。次に、受信バッファの空き容量が少なくなってきたら、フロー制御イネーブルビットFCEN<1:0> (EFLOCON<1:0>)を設定します。モジュールは進行中だった送信を完了させ、PAUSE タイマ値を格納した PAUSE フレームを送信します。選択したフロー制御モードによっては、前回もう一度アプリケーションでFCENビットに書き込みを実行してフロー制御モードをOFFにする必要があります。

RXPAUS ビットがセットされた状態でゼロ以外のPAUSEタイマ値を持つ有効なPAUSEフレームが到着すると、本モジュールは自動的に送信を禁止します。

パケットを送信するためにTXRTSビットがセットされると、ハードウェアはPAUSEタイマがゼロになるのを待ってからパケットの送信を試み、その後でTXRTSビットをクリアします。通常、この過程はマイクロコントローラに対して透過的に行われ、PAUSEフレームを受信した事をマイクロコントローラが認識する事はありません。MACがPAUSEフレームを受信したかどうかを調べる必要がある場合、ユーザアプリケーションでPASSALLビット(MACON1<1>)をセットして、PAUSE制御フレームを受信したらその内容を手動で解釈する必要があります。

## レジスタ 19-19: EFLOCON: Ethernet フロー制御レジスタ

U-0	U-0	U-0	U-0	U-0	R-0	R/W-0	R/W-0
—	—	—	—	—	r	FCEN1	FCEN0
bit 7						bit 0	

<b>凡例:</b>	r = 予約済みビット
R = 読み出し可能ビット	W = 書き込み可能ビット
-n = POR 時の値	U = 未実装ビット、「0」として読み出し
	「1」= ビットはセット 「0」= ビットはクリア x = ビットは未知

bit 7-3 **未実装:** 「0」として読み出し

bit 2 **予約済み:** 使用不可

bit 1-0 **FCEN<1:0>:** フロー制御イネーブルビット

**FULDPX (MACON3<0>) = 1 の場合:**

11 = タイマ値が「0」のPAUSEフレームを1つ送信してフロー制御をOFFにする

10 = PAUSEフレームを周期的に送信する

01 = PAUSEフレームを1つ送信してフロー制御をOFFにする

00 = フロー制御をOFFにする

**FULDPX (MACON3<0>) = 0 の場合:**

x1 = フロー制御をONにする

x0 = フロー制御をOFFにする

表 19-8: フロー制御で使うレジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	リセット値の記載ページ
ECON1	TXRST	RXRST	DMAST	CSUMEN	TXRTS	RXEN	—	—	70
MACON1	—	—	—	r	TXPAUS	RXPAUS	PASSALL	MARXEN	75
MABBIPG	—	BBIPG6	BBIPG5	BBIPG4	BBIPG3	BBIPG2	BBIPG1	BBIPG0	75
EFLOCON	—	—	—	—	—	r	FCEN1	FCEN0	75
EPAUSL	PAUSE タイマ値レジスタ下位バイト (EPAUS<7:0>)								75
EPAUSH	PAUSE タイマ値レジスタ上位バイト (EPAUS<15:8>)								75

凡例: — = 未実装、r = 予約済みビット、網掛けの部分は使いません。

## 19.8 受信フィルタ

マイクロコントローラの処理オーバーヘッドを最小限に抑えるため、本モジュールは不要なパケットを破棄する各種受信フィルタを内蔵しています。パケットフィルタは以下の6種類が実装されています。

- ユニキャスト
- マルチキャスト
- ブロードキャスト
- パターンマッチ
- Magic Packet™
- ハッシュテーブル

各フィルタの設定は、全て ERXFCON レジスタ (レジスタ 19-20)で行います。同時に複数のフィルタを有効にする事もできます。また、ANDOR ビットを使って複数のフィルタの論理積または論理和をとる事もできます。言い換えると、有効にした全てのフィルタで受け入れられたパケットのみを受信するか、いずれか1つのフィルタで受け入れられたパケットを受信するかを設定できます。図 19-13 と図 19-14 のフローチャートに、ANDOR ビットの設定できる各フィルタの働きを示します。

ERXFCON レジスタを 20h (CRC フィルタによるパケットの有効性チェックのみ有効) に設定すると、デバイスはプロミスキャストモードに移行し、有効な全てのパケットを受信します。ERXFCON レジスタの正しい設定は、アプリケーション要件によって決まります。

### 19.8.1 ユニキャスト フィルタ

ユニキャスト受信フィルタは、受信した全てのパケットの宛先アドレスをチェックします。宛先アドレスが MAADR レジスタの内容と一致するパケットは、ユニキャストフィルタの条件を満たします。

### 19.8.2 マルチキャスト フィルタ

マルチキャスト受信フィルタは、受信した全てのパケットの宛先アドレスをチェックします。宛先アドレスの先頭バイトの最下位ビットがセットされているパケットは、マルチキャストフィルタの条件を満たします。

### 19.8.3 ブロードキャスト フィルタ

ブロードキャスト受信フィルタは、受信した全てのパケットの宛先アドレスをチェックします。宛先アドレスが FF-FF-FF-FF-FF-FF のパケットは、ブロードキャストフィルタの条件を満たします。

### 19.8.4 ハッシュテーブル フィルタ

ハッシュテーブル受信フィルタは、主に特定のマルチキャストグループアドレスに送信されたトラフィックを受信する目的で使います。このフィルタはパケットの特定の宛先アドレスをチェックするため、不要パケットの除去能力はマルチキャストフィルタよりも優れています。

このフィルタは、パケットの宛先アドレス (6 バイト) に対して生成多項式 4C11DB7h を用いた 32 ビット CRC を実行します。この計算で 32 ビットの 2 進数を得たら、bit <28:23> の 6 ビット値を取得します。この値が、Ethernet ハッシュテーブルレジスタ ETH0 ~ ETH7 で形成されるテーブル内の位置を指し示します。その位置のビットがセットされていれば、そのパケットはハッシュテーブルフィルタの条件を満たしており、受け入れられます。表 19-9 に、テーブル内の各ビット位置とポインタ値の関係を示します。

例 19-1 に、ハッシュテーブルの使用例を示します。この例では、宛先アドレス 01-00-00-00-01-2C から得られるテーブルポインタ値 34h は、ETH6 の bit 4 を指し示しています。このビットが「1」の場合、パケットを受け入れられます。

つまりハッシュテーブルレジスタの全ビットをクリアした場合、フィルタ条件が満たされる事はなくなります。反対に、ハッシュテーブルの全ビットをセットすると、フィルタ条件は常に満たされます。

表 19-9: ハッシュテーブルレジスタのビット割り当て

レジスタ	ハッシュテーブル内のビット番号							
	7	6	5	4	3	2	1	0
EHT0	07	06	05	04	03	02	01	00
EHT1	0F	0E	0D	0C	0B	0A	09	08
EHT2	17	16	15	14	13	12	11	10
EHT3	1F	1E	1D	1C	1B	1A	19	18
EHT4	27	26	25	24	23	22	21	20
EHT5	2F	2E	2D	2C	2B	2A	29	28
EHT6	37	36	35	34	33	32	31	30
EHT7	3F	3E	3D	3C	3B	3A	39	38

### 例 19-1: ハッシュテーブル内のビット位置の求め方

パケットの宛先アドレス:

01-00-00-00-01-2C (16 進数)

4C11DB7h を用いた CRC-32 計算の結果:

1101 1010 0000 1011 0100 0101 0111 0101  
(2 進数)

CRC 計算結果の bit <28:23> から求めたポインタ:

110100 (2 進数) または 34 (16 進数)

対応するハッシュテーブル内の位置:

ETH6<4>

# PIC18F97J60 ファミリ

## レジスタ 19-20: ERXFCON: Ethernet 受信フィルタ制御レジスタ

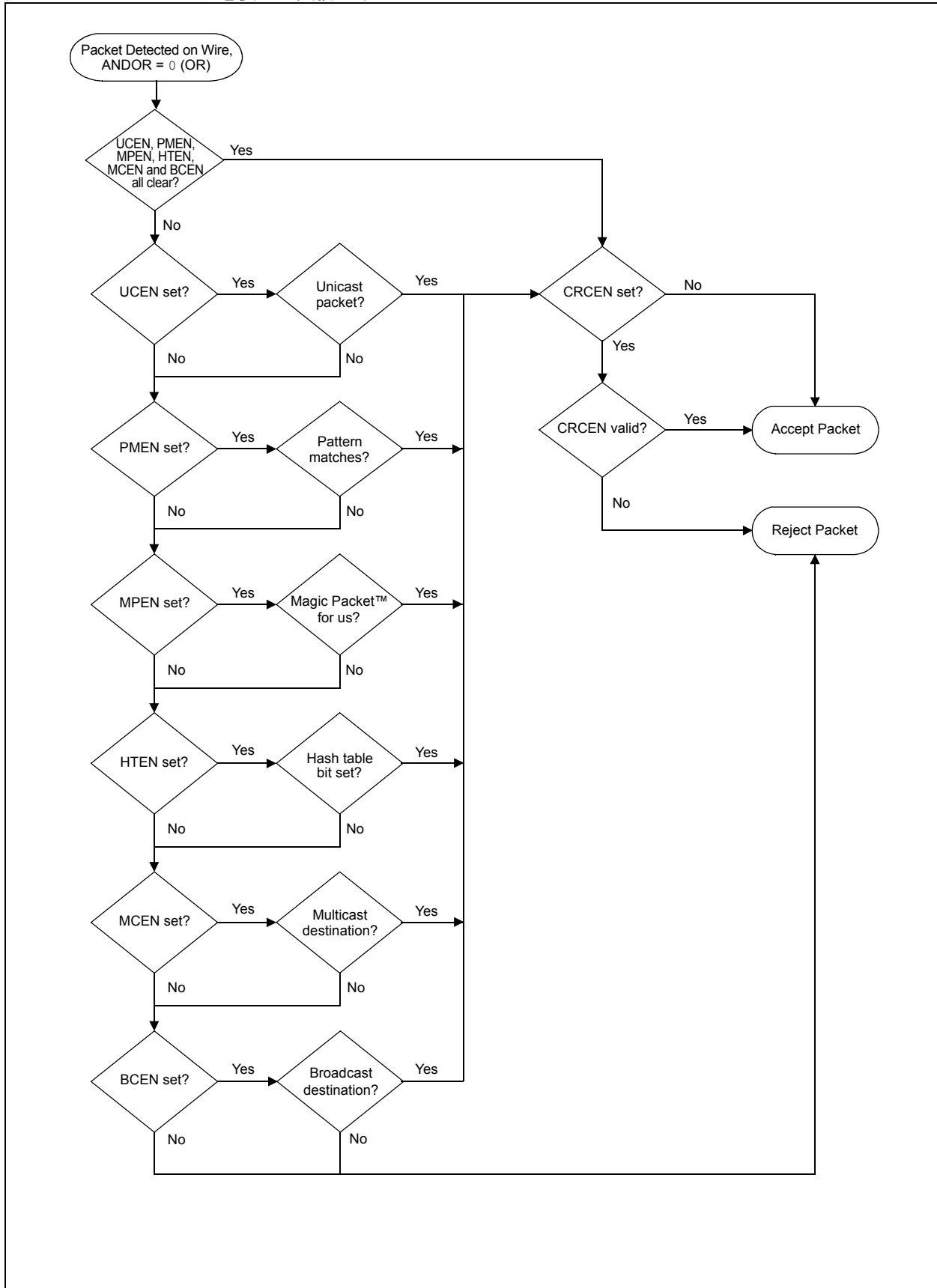
R/W-1	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1
UCEN	ANDOR	CRCEN	PMEN	MPEN	HTEN	MCEN	BCEN
bit 7							bit 0

### 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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

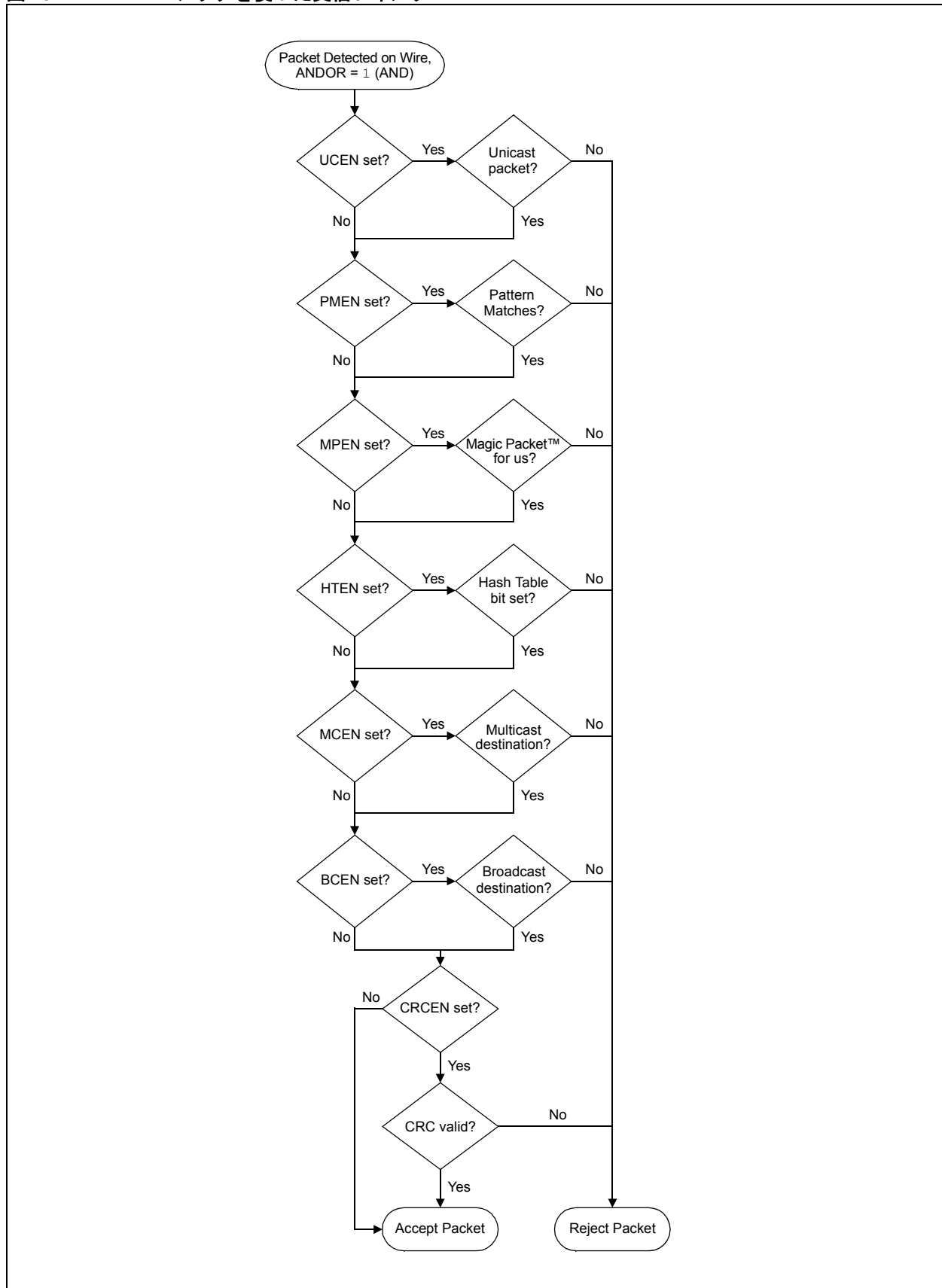
- bit 7      **UCEN:** ユニキャスト フィルタ イネーブルビット  
ANDOR = 1 の場合:  
 1 = 宛先アドレスがローカル MAC アドレスと一致しないパケットを破棄する  
 0 = フィルタを無効にする  
ANDOR = 0 の場合:  
 1 = 宛先アドレスがローカル MAC アドレスと一致するパケットを受け入れる  
 0 = フィルタを無効にする
- bit 6      **ANDOR:** AND/OR フィルタ選択ビット  
 1 = AND: 有効にした全てのフィルタが受け入れない限りパケットを拒否する  
 0 = OR: 有効にした全てのフィルタが拒否しない限りパケットを受け入れる
- bit 5      **CRCEN:** ポストフィルタ CRC チェック イネーブルビット  
 1 = CRC が無効なパケットを全て破棄する  
 0 = CRC の有効性を無視する
- bit 4      **PMEN:** パターンマッチ フィルタ イネーブルビット  
ANDOR = 1 の場合:  
 1 = パターンマッチの条件を満たさないパケットを破棄する  
 0 = フィルタを無効にする  
ANDOR = 0 の場合:  
 1 = パターンマッチの条件を満たすパケットを受け入れる  
 0 = フィルタを無効にする
- bit 3      **MPEN:** Magic Packet フィルタ イネーブルビット  
ANDOR = 1 の場合:  
 1 = ローカル MAC アドレスの Magic Packet でなければパケットを破棄する  
 0 = フィルタを無効にする  
ANDOR = 0 の場合:  
 1 = ローカル MAC アドレスの Magic Packet を受け入れる  
 0 = フィルタを無効にする
- bit 2      **HTEN:** ハッシュテーブル フィルタ イネーブルビット  
ANDOR = 1 の場合:  
 1 = ハッシュテーブルの条件を満たさないパケットを破棄する  
 0 = フィルタを無効にする  
ANDOR = 0 の場合:  
 1 = ハッシュテーブルの条件を満たすパケットを受け入れる  
 0 = フィルタを無効にする
- bit 1      **MCEN:** マルチキャスト フィルタ イネーブルビット  
ANDOR = 1 の場合:  
 1 = 宛先アドレスの先頭バイトの最下位ビットがセットされていないパケットを破棄する  
 0 = フィルタを無効にする  
ANDOR = 0 の場合:  
 1 = 宛先アドレスの先頭バイトの最下位ビットがセットされているパケットを受け入れる  
 0 = フィルタを無効にする
- bit 0      **BCEN:** ブロードキャスト フィルタ イネーブルビット  
ANDOR = 1 の場合:  
 1 = 宛先アドレスが FF-FF-FF-FF-FF-FF でないパケットを破棄する  
 0 = フィルタを無効にする  
ANDOR = 0 の場合:  
 1 = 宛先アドレスが FF-FF-FF-FF-FF-FF のパケットを受け入れる  
 0 = フィルタを無効にする

図 19-13: OR ロジックを使った受信フィルタ



# PIC18F97J60 ファミリ

図 19-14: AND ロジックを使った受信フィルタ



## 19.8.5 パターンマッチ フィルタ

パターンマッチ フィルタは受信パケットから最大 64 バイトを選択し、これらのバイトの IP チェックサムを計算します。次に、チェックサムと EPMCS レジスタを比較します。計算で求めたチェックサムと EPMCS レジスタが一致する場合、パケットはパターンマッチ フィルタの条件を満たします。パターンマッチ フィルタは、パケット内部に予測するデータが含まれるパケットのフィルタ処理に便利です。

パターンマッチ フィルタを使うには、パターンマッチ オフセット (EPMOH:EPMOL)、全てのパターンマッチ マスクバイト (EPMM0:EPMM7)、パターンマッチ チェックサム レジスタペア (EPMCSH:EPMCSL) をアプリケーションでプログラムする必要があります。パターンマッチ オフセットには、宛先アドレス フィールドの先頭からチェックサム計算に使う 64 バイト ウィンドウまでのオフセットを書き込みます。64 バイト ウィンドウ内の各バイトをチェックサム計算に含めるか除外するかは、パターンマッチ マスク内の対応するビットをセットまたはクリアする事によって選択できます。受信したパケットで CRC フィールドの末尾が 64 バイト ウィンドウを越えている場合、対応するマスクビットが全て「0」であっても、フィルタ条件を満たさないものと判定されます。

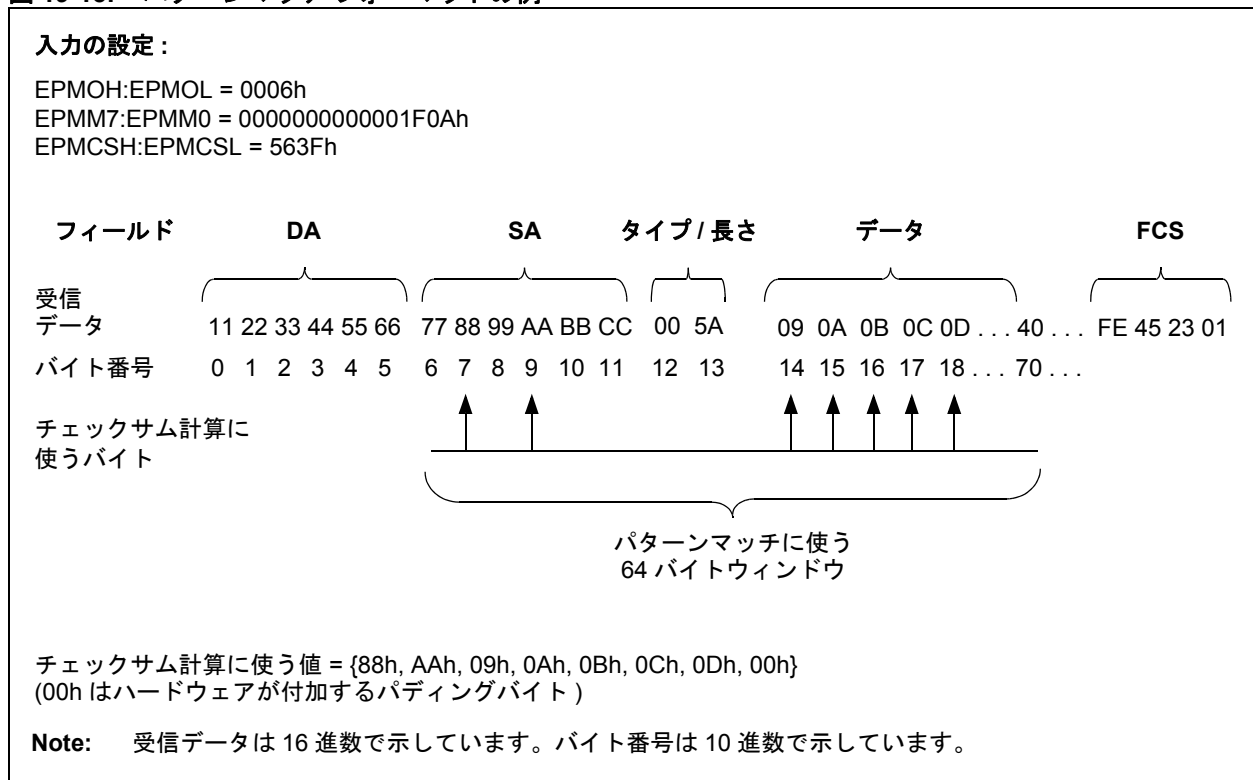
**Note:** いずれの場合も、正常に動作するにはパターンマッチ オフセットの値は偶数である必要があります。EPMO レジスタペアに奇数値を書き込むと、予測不能な結果が生じます。

パターンマッチ チェックサム レジスタには、選択したバイトに対して予測されるチェックサムを書き込みます。チェックサムは、DMA モジュールによるチェックサム計算 ( [セクション 19.9.2「チェックサム計算」](#) 参照) と同じ方法で計算します。対応するマスクビットを「0」にプログラムしたデータバイトは 0 として扱うのではなく、チェックサム計算から完全に除外されます。

例えば、アプリケーションで送信元 MAC アドレスが 00-04-A3-FF-FF-FF のパケットを全て拒否する場合、パターンマッチ オフセットに 0000h を書き込み、EPMM0 レジスタの bit 6 と 7、EPMM1 レジスタの bit 0、1、2、3 をセットします ( それ以外のマスクビットは全て「0」)。EPMCS レジスタに書き込む適切なチェックサムは 5BFCh です。別の設定方法として、オフセットに 0006h を書き込み、EPMM0 レジスタの bit 0、1、2、3、4、5 をセットする事もできます。この場合もチェックサムは 5BFCh です。しかし 2 つ目の方法では、長さが 70 バイト未満のパケットはマスク設定に基づいて正しいチェックサムが生成された場合でもパターンマッチ条件を満たさないため、あまり推奨できません。

[図 19-15](#) に、もう 1 つのパターンマッチ フィルタの例を示します。

**図 19-15: パターンマッチ フォーマットの例**

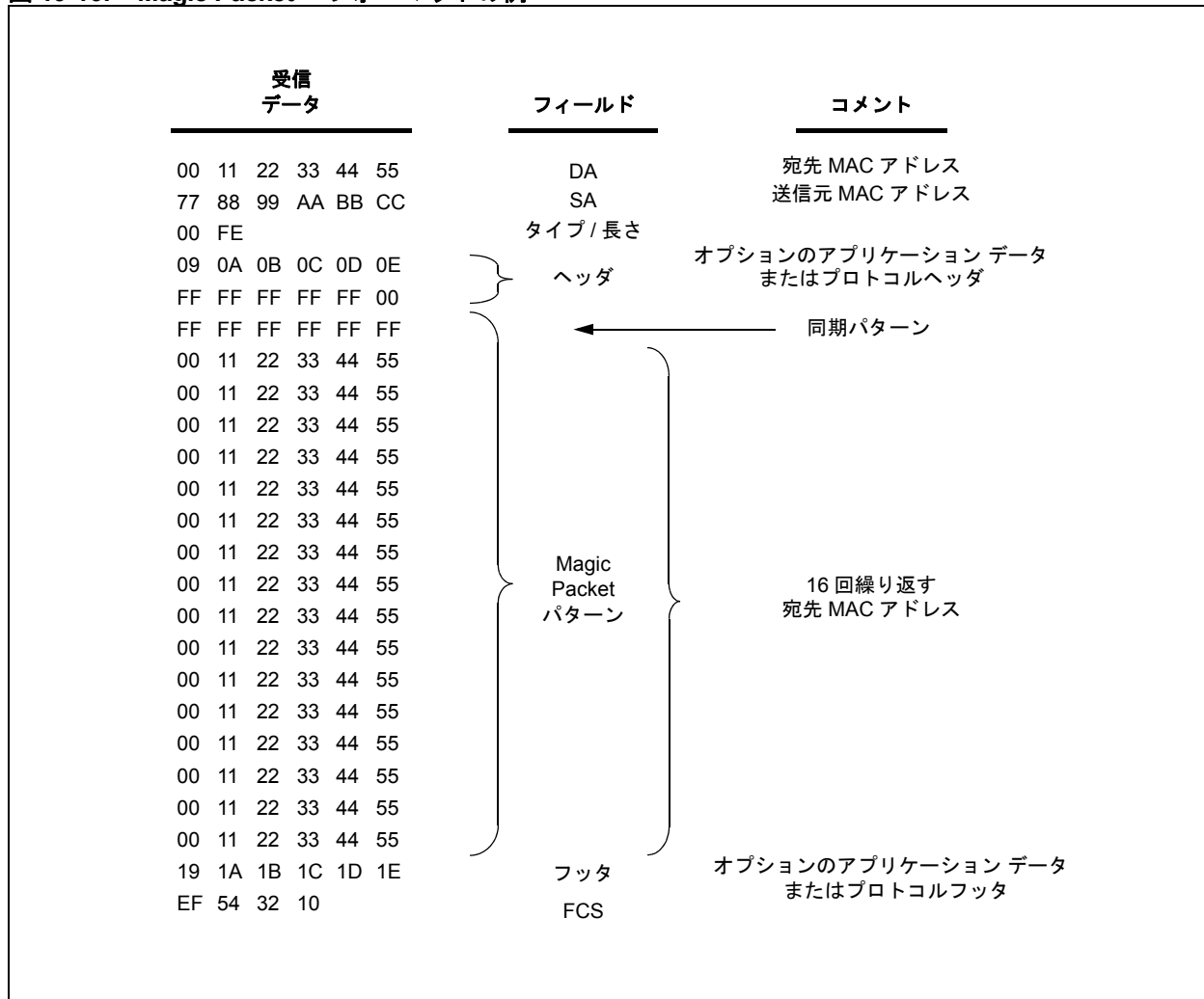


# PIC18F97J60 ファミリ

## 19.8.6 Magic Packet フィルタ

Magic Packet パターンは、同期用の 6 つの FFh バイトに続き宛先アドレスを 16 回繰り返した構造です ( 図 19-16)。Magic Packet フィルタは、全ての受信パケットの宛先アドレスとデータフィールドをチェックします。宛先アドレスが MAADR レジスタと一致し、データフィールド内のどこかに有効な Magic Packet パターンが含まれていれば、そのパケットは Magic Packet フィルタの条件を満たします。

図 19-16: Magic Packet™ フォーマットの例





## 19.9 ダイレクトメモリアクセス (DMA) コントローラ

Ethernet モジュールは 2 つの用途を持つ DMA コントローラが内蔵されています。1 つ目の用途は、8 KB のメモリバッファ内でデータをコピーする事です。2 つ目の用途は、TCP、UDP、IP、ICMP 等、業界標準の各種通信プロトコルと互換の 16 ビット チェックサム の計算です。

DMA は以下に示す 3 つのポインタと各種ステータス / 制御ビットを使って制御します。

- EDMASTH:EDMASTL: コピー元開始アドレス
- EDMANDH:EDMANDL: コピー元終了アドレス
- EDMADSTH:EDMADSTL: 格納先開始アドレス
- DMAST および CSUMEN ビット (ECON1<5,4>): DMA 開始 /BUSY およびチェックサム イネーブル ビット
- DMAIE および DMAIF ビット (EIE<5>、EIR<5>): DMA 割り込みイネーブルおよびフラグビット

コピー元開始アドレスとコピー元終了アドレスは、コピーまたはチェックサム計算するデータを定義します。格納先開始アドレスはデータコピーの場合のみ使い、コピーしたデータの格納先を定義します。これら 3 つのポインタは全て 8 KB の Ethernet メモリに対するもので、PIC<sup>®</sup> マイクロコントローラのデータメモリアクセスには使えません。

DMA の動作が開始すると、EDMAST レジスタペアが内部コピー元ポインタへコピーされます。DMA はバイト単位で実行し、実行後に内部コピー元ポインタをインクリメントします。しかし、1 バイトを処理した後、内部コピー元ポインタの値が受信バッファ終了ポインタペアの ERXND と同じ場合、コピー元ポインタはインクリメントしません。代わりに、受信バッファ開始ポインタペアの ERXST の値が内部コピー元ポインタに格納されます。このようにして、DMA は受信バッファの巡回 FIFO 構造に従って、受信バケットを 1 回の DMA 動作で処理します。内部コピー元ポインタの値が EDMAND ポインタと同じになると DMA の動作は終了します。

何らかの DMA 動作が進行中は、DMA ポインタと CSUMEN ビット (ECON1<4>) の値を変更しないでください。DMA 動作は、DMAST ビット (ECON1<5>) をクリアするといつでもキャンセルできます。レジスタは一切変更されませんが、DMA コピーが進行中の場合は一部のメモリバイトがコピー済みの事があります。

DMA を使う際は、動作に関するいくつかの要件に常に注意する必要があります。以下に示す要件に違反した場合、Ethernet バッファのデータが失われる事や、Ethernet の動作が完全にエラーになる事があります。

- 受信バッファのラップアラウンド動作のために EDMAND ポインタペアの値に到達しない場合、DMA 動作は永久に終了しません。
- 設計上、DMA モジュールは 1 バイトのみに対して (すなわち EDMAST = EDMAND として) コピーまたはチェックサム計算を実行する事はできません。このような処理を実行しようとする、バッファ内の

のメモリが全て上書きされ、処理が永久に終了しません。

- DMA 動作が終了した後 (DMAST ビットはハードウェアまたはファームウェアによってクリアされます)、4 命令サイクル以内にアプリケーションで DMAST ビットを再びセットする事はできません。
- 動作の信頼性を確保するため、DMA コピー動作中はアプリケーションから EDATA へのアクセスを避けてください。DMA チェックサム計算中は EDATA にアクセスしても安全です。

### 19.9.1 メモリのコピー

バッファ内でメモリをコピーする手順は以下の通りです。

1. EDMAST、EDMAND、EDMADST レジスタペアに、適切なコピー元開始アドレス、コピー元終了アドレス、格納先開始アドレスを書き込む。EDMAST レジスタはコピー元の先頭バイト、EDMAND レジスタはコピー元の最終バイト、EDMADST レジスタは格納先レンジの先頭バイトをそれぞれ指し示す。格納先レンジは、8 KB メモリ境界でのラップアラウンド (8191 から 0) を除き、一切ラップアラウンドせず常にリニアである。終了ポインタの値を正しく計算しないと DMA 動作が終了せず 8 KB バッファ全体が上書きされてしまうため、特に注意が必要である。
2. 必要に応じて、DMAIE ビット (EIE<5>) と ETHIE ビット (PIE2<5>) をセットし、DMAIF フラグビット (EIR<5>) をクリアして、コピー終了時に割り込みを発生させる。
3. CSUMEN ビット (ECON1<4>) をクリアする。
4. DMAST ビット (ECON1<5>) をセットすると、DMA コピーが開始する。

DMAST ビットをセットした時に送信動作が進行中 (TXRTS ビットがセットされている) の場合、DMA コントローラは送信動作が完了するのを待ってから DMA コピーを試みます。このような遅延が必要なのは、DMA と送信エンジンが同時にはバッファにアクセスできないためです。

コピーが完了すると、DMA ハードウェアが DMAST ビットをクリアし、DMAIF ビットをセットして、割り込みを生成します (割り込み有効時)。ポインタと EDMACS レジスタの値は変更されません。

DMA モジュールが初期化後にコピーを開始すると、1 命令サイクル (Tcy) で 1 バイトがコピーされます。しかし、Ethernet 受信ハードウェアが 1 バイトのデータをバッファに格納すると、DMA はストールして優先度の高い処理が実行されます。他にメモリ帯域幅を一切使わないで最大サイズ (1518 バイト) のパケットをコピーした場合、DMA モジュールは完了までに 145.7 μs (コア周波数 41.667 MHz 時) より若干長い時間を必要とします。最小サイズ (64 バイト) のパケットのコピーにかかる時間は、約 6.2 μs (コア周波数 41.667 MHz 時) とレジスタ コンフィグレーション時間の合計です。

# PIC18F97J60 ファミリ

## 19.9.2 チェックサム計算

チェックサム計算ロジックは、コピー元データを一連の16ビットビッグエンディアン整数として扱います。コピー元レンジに含まれるバイトが奇数個の場合、チェックサム計算を実行するために、末尾にパディングバイトの00hが付加されます。

計算したチェックサムは、全ての16ビット整数の1の補数(16ビット)の1の補数(16ビット)です。例えば、チェックサムに含まれるバイトが{89h, ABh, CDh}の場合、チェックサム計算はまず「89ABh + CD00h」から実行します。この例のように16番目のビットでキャリアウトが発生する場合、16ビットの1の補数演算では最初のビットに1が加算されます。こうして得られた56AChの補数をとって、最終的にA953hのチェックサム値を得ます。

チェックサムの計算手順は以下の通りです。

1. チェックサムに含めるバッファデータの最初と最後のバイトをEDMAST および EDMAND レジスタペアで指定する。ポインタ値を設定する際は、受信バッファのラップアラウンドによってチェックサム計算が終わらない事のないように十分注意する。
2. チェックサム計算の完了時にオプションとして割り込みを生成する場合、DMAIE ビット (EIE<5>) と ETHIE ビット (PIE2<5>) をセットして、DMAIF ビット (EIR<5>) をクリアする。
3. CSUMEN ビット (ECON1<4>) と DMAST ビット (ECON1<5>) をセットして、チェックサム計算を開始する。

チェックサム計算が完了したら、DMAハードウェアがDMASTビットをクリアし、DMAIFビットをセットして割り込みを生成します(割り込み有効時)。DMAポインタは変更されず、メモリへの書き込みは行われません。計算したチェックサムはEDMACSHおよびEDMACSLレジスタに格納されます。アプリケーションは、このチェックサム値をパケットに書き込み、ゼロと比較し(チェックサムフィールドを含む受信データブロックを検証する場合)、あるいは他のチェックサム(TCPやUDP等の各種プロトコルで使われる疑似ヘッダチェックサム等)と比較する事ができます。

DMAをチェックサムモードで動作させる場合、チェックサムに含まれる1バイトの処理に1命令サイクル(Tcy)かかります。このため、1446バイトに対するチェックサム計算を実行した場合、DMAモジュールは動作完了までに138.8μs(周波数41.667MHz時)より若干長い時間を必要とします。

同じ周波数なら、20バイトのヘッダフィールドの処理にはチェックサム計算のための約1.9μsと、DMAセットアップ時間の合計時間がかかります。ここに示した推定時間は、DMAが計算実行中に、Ethernet受信ハードウェアがメモリアクセス帯域幅を使っておらず、CPUがEDATAレジスタに対して読み書きコマンドを発行していない事を前提としています。

DMAコピーモードと同様に、チェックサム計算はTXRTSビット(ECON1<3>)がクリアされるまで開始しません。このため、アプリケーションで大きなパケットを送信した直後に受信パケットのチェックサムを検証しようとした場合、チェックサム計算時間にかかなり長い時間がかかる事があります。

表 19-10: DMA コントローラに関連するレジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	リセット値の記載ページ
EIE	-	PKTIE	DMAIE	LINKIE	TXIE	-	TXERIE	RXERIE	73
EIR	-	PKTIF	DMAIF	LINKIF	TXIF	-	TXERIF	RXERIF	73
ECON1	TXRST	RXRST	DMAST	CSUMEN	TXRTS	RXEN	-	-	70
ERXNDL	受信終了レジスタ下位バイト (ERXND<7:0>)								73
ERXNDH	-	-	-	受信終了レジスタ上位バイト (ERXND<12:8>)					73
EDMASTL	DMA 開始レジスタ下位バイト (EDMAST<7:0>)								73
EDMASTH	-	-	-	DMA 開始レジスタ上位バイト (EDMAST<12:8>)					73
EDMANDL	DMA 終了レジスタ下位バイト (EDMAND<7:0>)								73
EDMANDH	-	-	-	DMA 終了レジスタ上位バイト (EDMAND<12:8>)					73
EDMADSTL	DMA 格納先レジスタ下位バイト (EDMADST<7:0>)								73
EDMADSTH	-	-	-	DMA 格納先レジスタ上位バイト (EDMADST<12:8>)					73
EDMACSL	DMA チェックサム レジスタ下位バイト (EDMACS<7:0>)								73
EDMACSH	DMA チェックサム レジスタ上位バイト (EDMACS<15:8>)								73

凡例: — = 未実装、網掛けの部分は使いません。

## 19.10 モジュールのリセット

本モジュールでは以下の選択的リセットを使えます。

- 送信のみリセット
- 受信のみリセット

### 19.10.1 マイクロコントローラのリセット

標準のリセットイベントが発生すると、Ethernet モジュールは既知の状態に戻ります。Ethernet バッファメモリの内容は未知です。SFR と PHY レジスタには、リセットイベントの種類に応じて指定されたリセット値が読み込まれます。しかし、PHY 起動タイマのカウントが終了し、PHYRDY ビット (ESTAT<0>) がセットされるまで (すなわち、ETHEN ビットをセットしてから 1 ms 以上経過するまで) は、PHY レジスタにアクセスしないでください。詳細は、[セクション 19.1.3.1 「起動タイマ」](#) を参照してください。

### 19.10.2 送信のみリセット

TXRST ビット (ECON1<7>) に「1」を書き込むと、送信のみリセットが実行されます。この場合、送信ロジックのみがリセットされます。バッファ マネジメントやホスト インターフェイス等、他のレジスタと制御ブロックは送信のみリセットイベントの影響を受けません。通常動作に戻るには、TXRST ビットをソフトウェアでクリアする必要があります。TXRST をクリアしてから少なくとも 1.6  $\mu$ s が経過するまでは、ファームウェアから Ethernet モジュールの SFR への書き込みを行わないでください。この遅延経過後は、通常の動作を再開できます。

### 19.10.3 受信のみリセット

RXRST ビット (ECON1<6>) に「1」を書き込むと、受信のみリセットが実行されます。この場合、受信ロジックのみがリセットされます。バッファ マネジメントやホスト インターフェイス ブロック等、他のレジスタと制御ブロックは受信のみリセットイベントの影響を受けません。通常動作に戻るには、RXRST ビットをソフトウェアでクリアする必要があります。RXRST をクリアしてから少なくとも 1.6  $\mu$ s が経過するまでは、ファームウェアから Ethernet モジュールの SFR への書き込みを行わないでください。この遅延経過後は、通常の動作を再開できます。

# PIC18F97J60 ファミリ

---

NOTES:

## 20.0 マスタ同期シリアルポート (MSSP) モジュール

### 20.1 MSSP モジュールの概要

マスタ同期シリアルポート (MSSP) モジュールとは、他の周辺デバイスまたはマイクロコントローラとの通信に便利なシリアル インターフェイスです。ここでの周辺デバイスとは、シリアル EEPROM、シフトレジスタ、ディスプレイドライバ、A/D コンバータ (ADC) 等を指します。MSSP モジュールは、以下の 2 つのモードのいずれかで動作します。

- シリアル ペリフェラル インターフェイス (SPI)
- I<sup>2</sup>C™ (Inter-Integrated Circuit)
  - フルマスタモード
  - スレーブモード (ジェネラルコール アドレス対応)

I<sup>2</sup>C インターフェイスは、ハードウェアで以下のモードをサポートしています。

- マスタモード
- マルチマスタ モード
- スレーブモード

64 ピン /80 ピンの PIC18F97J60 ファミリは MSSP モジュールを 1 つ内蔵しています (MSSP1)。100 ピンデバイス は MSSP モジュールを 2 つ内蔵しています (MSSP1、MSSP2)。各モジュールは互いに独立して動作します。

**Note:** このセクション全体を通して、いずれの動作モードでも MSSP1 と MSSP2 で共通する内容を「MSSP モジュール」と総称して説明します。レジスタ名とモジュールの I/O 信号の総称には「x」を使います。必要に応じてモジュール識別番号に読み換えてください。制御ビット名はどちらのモジュールも共通です。

### 20.2 制御レジスタ

各 MSSP モジュールは関連する 3 つのレジスタを使います。ステータス レジスタが 1 つ (SSPxSTAT) と制御レジスタが 2 つ (SSPxCON1、SSPxCON2) です。これらのレジスタと各コンフィグレーションビットの使い方は、MSSP モジュールの動作モードが SPI か I<sup>2</sup>C かによって大きく異なります。

詳細は、各セクションで説明します。

**Note:** MSSP モジュールを複数内蔵したデバイスでは、SSPxCON レジスタの名前に注意を払う必要があります。SSP1CON1 と SSP1CON2 は、同じモジュールの異なる機能を制御します。一方、SSP1CON1 と SSP2CON1 は、2 つの異なるモジュールの同じ機能を制御します。

### 20.3 SPI モード

SPI モードでは、8 ビットのデータの送受信を同期方式により同時に実行できます。SPI の 4 つのモード全てに対応しています。通信を確立するために、通常、以下の 3 本のピンを使います。

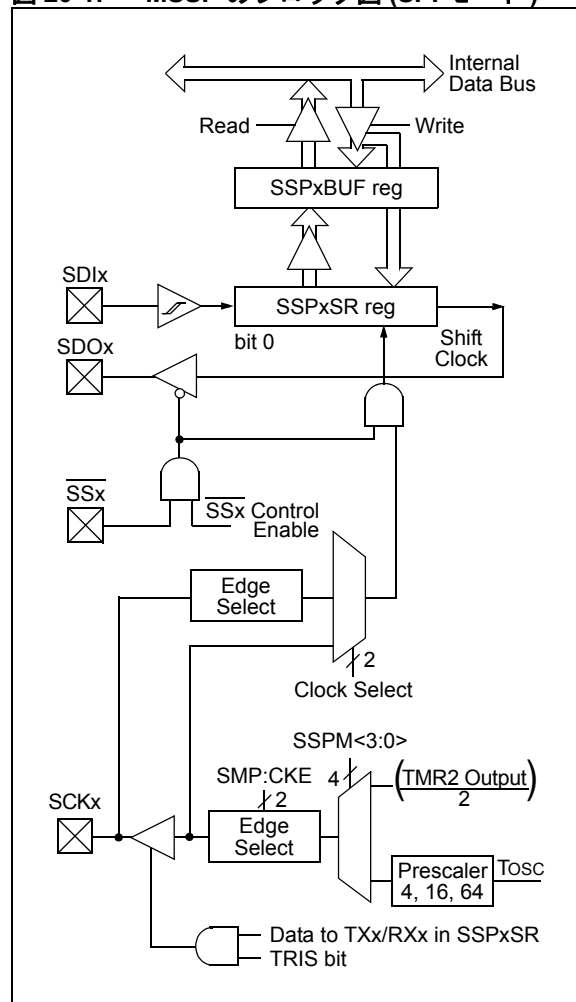
- シリアルデータ出力 (SDOx) – RC5/SDO1 (100 ピンデバイスの場合は RD4/SDO2)
- シリアルデータ入力 (SDIx) – RC4/SDI1/SDA1 (100 ピンデバイスの場合は RD5/SDI2/SDA2)
- シリアルクロック (SCKx) – RC3/SCK1/SCL1 (100 ピンデバイスの場合は RD6/SCK2/SCL2)

さらに、スレーブモードの動作では 4 本目のピンを使う場合があります。

- スレーブセレクト ( $\overline{SSx}$ ) – RF7/ $\overline{SS1}$  (100 ピンデバイスの場合は RD7/ $\overline{SS2}$ )

図 20-1 に、SPI モードで動作する MSSP モジュールのブロック図を示します。

図 20-1: MSSP のブロック図 (SPI モード)



# PIC18F97J60 ファミリ

## 20.3.1 レジスタ

各 MSSP モジュールは、SPI モードの動作で以下の 4 つのレジスタを使います。

- MSSPx 制御レジスタ 1 (SSPxCON1)
- MSSPx ステータス レジスタ (SSPxSTAT)
- シリアル受信 / 送信バッファレジスタ (SSPxBUF)
- MSSPx シフトレジスタ (SSPxSR) – 直接アクセスは不可

SSPxCON1 と SSPxSTAT は、SPI モード動作時の制御レジスタとステータス レジスタです。SSPxCON1 レジスタは、読み書き可能です。SSPxSTAT の下位 6 ビットは読み出し専用です。SSPxSTAT の上位 2 ビットは読み書き可能です。

SSPxSR はデータをシフト入出力するためのシフトレジスタです。SSPxBUF はデータバイトの読み書き用バッファレジスタです。

受信動作の場合、SSPxSR と SSPxBUF が連動して、ダブルバッファ レシーバを構成します。SSPxSR が 1 バイトを全て受信すると SSPxBUF に転送され、SSPxIF 割り込みがセットされます。

送信時の SSPxBUF はダブルバッファを構成しません。SSPxBUF に書き込むと、SSPxBUF と SSPxSR の両方に同じ内容が書き込まれます。

### レジスタ 20-1: SSPxSTAT: MSSPx ステータス レジスタ (SPI モード)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE <sup>(1)</sup>	D/A	P	S	R/W	UA	BF
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

bit 7      **SMP:** サンプルビット

#### SPI マスタモード:

- 1 = データ出力期間の最後に入力データをサンプルする
- 0 = データ出力期間の途中で入力データをサンプルする

#### SPI スレーブモード:

SPI をスレーブモードで使用する場合、SMP をクリアする必要がある。

bit 6      **CKE:** SPI クロック選択ビット<sup>(1)</sup>

- 1 = クロック状態がアクティブからアイドルに遷移する時に送信する
- 0 = クロック状態がアイドルからアクティブに遷移する時に送信する

bit 5      **D/A:** データ / アドレスビット

I<sup>2</sup>C モードでのみ使う

bit 4      **P:** STOP ビット

I<sup>2</sup>C モードでのみ使う。SSPEN をクリアして MSSP モジュールを無効にすると、このビットはクリアされる

bit 3      **S:** START ビット

I<sup>2</sup>C モードでのみ使う

bit 2      **R/W:** 読み出し / 書き込み情報ビット

I<sup>2</sup>C モードでのみ使う

bit 1      **UA:** 更新アドレスビット

I<sup>2</sup>C モードでのみ使う

bit 0      **BF:** バッファフル ステータスビット (受信モードのみ)

- 1 = 受信が完了し、SSPxBUF がフルである
- 0 = 受信が完了しておらず、SSPxBUF がエンptyである

**Note 1:** クロック状態の極性は、CKP ビット (SSPxCON1<4>) で設定します。

# PIC18F97J60 ファミリ

## レジスタ 20-2: SSPxCON1: MSSPx 制御レジスタ 1 (SPI モード)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV <sup>(1)</sup>	SSPEN <sup>(2)</sup>	CKP	SSPM3 <sup>(3)</sup>	SSPM2 <sup>(3)</sup>	SSPM1 <sup>(3)</sup>	SSPM0 <sup>(3)</sup>
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

- bit 7      **WCOL:** 書き込みコリジョン検出ビット (送信モードのみ)  
 1 = 前のワードを送信中の SSPxBUF レジスタに対して書き込みを実行した (ソフトウェアによるクリアが必要)  
 0 = コリジョンは発生していない
- bit 6      **SSPOV:** 受信オーバーフロー インジケータ ビット<sup>(1)</sup>  
SPIスレーブモード:  
 1 = SSPxBUF レジスタがまだ前のデータを保持している間に、新しいバイトを受信した。オーバーフローが発生すると、SSPxSR 内のデータは失われる。オーバーフローは、スレーブモードの場合のみ発生する。データ送信しか実行しない場合も、オーバーフロー ビットがセットされるのを防ぐために SSPxBUF の読み出しが必要 (このビットはソフトウェアによるクリアが必要)  
 0 = オーバーフローは発生していない
- bit 5      **SSPEN:** MSSP イネーブルビット<sup>(2)</sup>  
 1 = シリアルポートを有効にし、SCKx、SDOx、SDIx、 $\overline{SSx}$  をシリアルポート ピンとして設定する  
 0 = シリアルポートを無効にし、上記のピンを I/O ポートピンとして設定する
- bit 4      **CKP:** クロック極性選択ビット  
 1 = クロックのアイドル状態を High レベルに設定する  
 0 = クロックのアイドル状態を Low レベルに設定する
- bit 3-0      **SSPM<3:0>:** MSSP モード選択ビット<sup>(3)</sup>  
 0101 = SPIスレーブモード、クロック = SCKx ピン、 $\overline{SSx}$  ピン制御無効、 $\overline{SSx}$  は I/O ピンとして使用可能  
 0100 = SPIスレーブモード、クロック = SCKx ピン、 $\overline{SSx}$  ピン制御有効  
 0011 = SPI マスタモード、クロック = TMR2 出力 /2  
 0010 = SPI マスタモード、クロック = Fosc/64  
 0001 = SPI マスタモード、クロック = Fosc/16  
 0000 = SPI マスタモード、クロック = Fosc/4

**Note 1:** マスタモードの場合、新たな送受信は SSPxBUF レジスタへの書き込みによって開始するため、オーバーフロー ビットはセットされません。

**2:** このビットを有効にした場合、これらのピンは入出力として適切に設定する必要があります。

**3:** ここに値を示していないビットの組み合わせは、予約済みか、I<sup>2</sup>C™ モードでのみ使うものです。

# PIC18F97J60 ファミリ

## 20.3.2 動作

SPIの初期化時には、いくつかの設定を行う必要があります。それには、該当する制御ビット (SSPxCON1<5:0> と SSPxSTAT<7:6>) をプログラムします。これらの制御ビットにより、以下を指定できます。

- マスタモード (SCKx はクロック出力)
- スレーブモード (SCKx はクロック入力)
- クロック極性 (SCKx のアイドル状態)
- データ入力サンプル位相 (データ出力期間の間または最後)
- クロックエッジ (SCKx の立ち上がり / 立ち下がりエッジでデータを出力)
- クロックレート (マスタモードのみ)
- スレーブセレクトモード (スレーブモードのみ)

各 MSSP モジュールは、送信 / 受信シフトレジスタ (SSPxSR) とバッファレジスタ (SSPxBUF) を使います。SSPxSR は MSb から順にデータをシフト入力またはシフト出力します。SSPxBUF は受信データが揃うまで、SSPxSR に書き込まれたデータを保持します。8 ビットのデータの受信が完了すると、受信したバイトは SSPxBUF レジスタに転送されます。この時点でバッファフル検出ビット BF (SSPxSTAT<0>) と、割り込みフラグビット SSPxIF がセットされます。このようなダブルバッファリング方式 (SSPxBUF) でデータを受

信する事で、今受信したデータを読み出す前に、次のバイトの受信を開始できます。データ送受信中の SSPxBUF レジスタへの書き込みは全て無視され、書き込みコリジョン検出ビット WCOL (SSPxCON1<7>) がセットされます。SSPxBUF レジスタへの書き込みが正しく完了したかどうかを正しく判定できるように、WCOL ビットはユーザ ソフトウェアでクリアする必要があります。

アプリケーション ソフトウェアが有効なデータを受信する事が予測される場合、次の送信バイトデータが SSPxBUF へ書き込まれる前に SSPxBUF のデータを読み出す必要があります。SSPxBUF に受信データが読み込まれると、バッファフル ビット BF (SSPxSTAT<0>) がセットされて送信の完了を示します。BF ビットは SSPxBUF を読み出すとクリアされます。SPI が送信のみの場合、このデータは無意味です。通常、送受信の完了を判断するには、MSSP 割り込みを使います。この割り込みが発生したら SSPxBUF の読み書きを実行する必要があります。割り込みによる方法を使わずに書き込みコリジョンを避けるには、ソフトウェアポーリングを実行します。例 20-1 に、データ送信時の SSP1BUF (SSP1SR) への読み込み方法を示します。

SSPxSR は直接読み書きできないため、SSPxBUF レジスタのアドレスを指定してアクセスします。また、SSPxSTAT レジスタは各種ステータス条件を示します。

### 例 20-1: SSP1BUF (SSP1SR) レジスタへの読み込み

LOOP	BTFSS	SSP1STAT, BF	;Has data been received (transmit complete)?
	BRA	LOOP	;No
	MOVF	SSP1BUF, W	;WREG reg = contents of SSP1BUF
	MOVWF	RXDATA	;Save in user RAM, if data is meaningful
	MOVF	TXDATA, W	;W reg = contents of TXDATA
	MOVWF	SSP1BUF	;New data to xmit



## 20.3.3 SPI I/O の有効化

シリアルポートを有効にするには、MSSP イネーブルビット SSPEN (SSPxCON1<5>) をセットする必要があります。SPI モードをリセットまたは再設定するには、SSPEN ビットをクリアし、SSPxCON レジスタを再初期化してから SSPEN ビットをセットします。これにより、SDIx、SDOx、SCKx、SSx ピンがシリアルポートピンとして設定されます。これらのピンをシリアルポートとして動作させる場合、一部のピンでは TRIS レジスタでデータ方向ビットを以下のように設定する必要があります。

- SDIx: SPI モジュールによって自動制御される
- SDOx: TRISC<5> (または TRISD<4>) ビットをクリアする
- SCKx (マスタモード): TRISC<3> (または TRISD<6>) ビットをクリアする
- SCKx (スレーブモード): TRISC<3> (または TRISD<6>) ビットをセットする
- SSx: TRISF<7> (または TRISD<7>) ビットをセットする

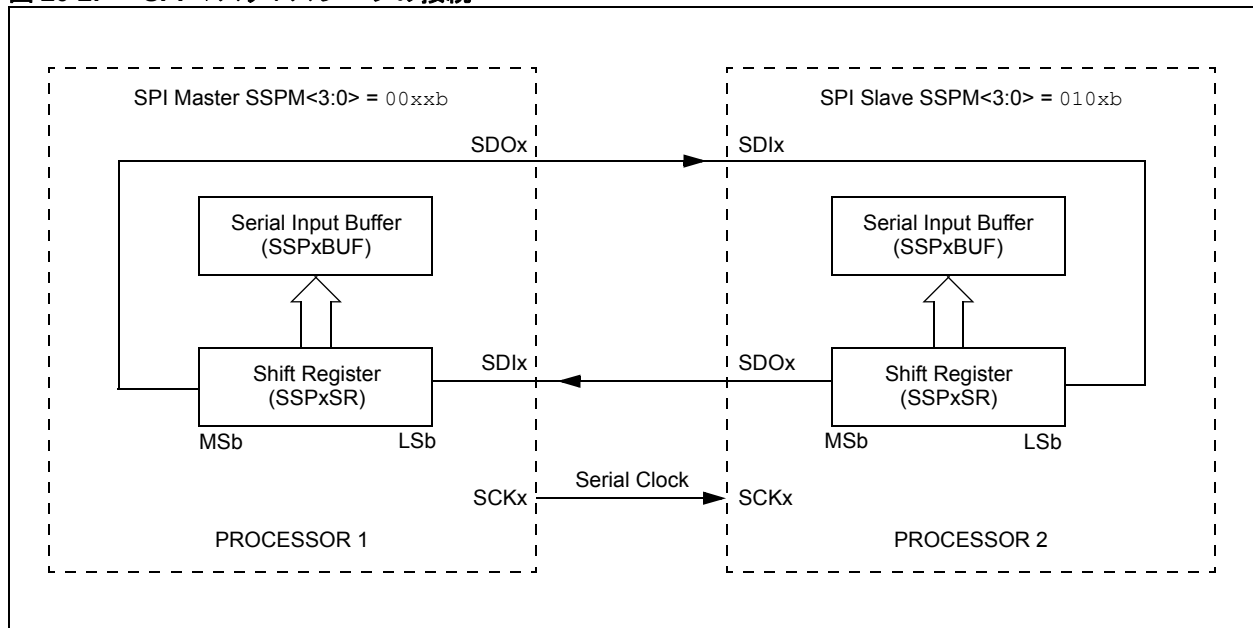
不要なシリアルポート機能は、対応するデータ方向 (TRIS) レジスタを逆の値にプログラミングする事で無効にできます。

## 20.3.4 代表的な接続

図 20-2 に、2つのマイクロコントローラ間の代表的な接続方法を示します。マスタコントローラ (プロセッサ 1) が、SCKx 信号を送信してデータ転送を開始します。データは、プログラムされたクロックエッジごとに両方のシフトレジスタからシフト出力され、逆のクロックエッジごとにラッチされます。両方のプロセッサには、同じクロック極性 (CKP) を設定します。これによって、両方のコントローラが同時にデータを送受信できます。データが有効なものか (またはダミーデータか) は、アプリケーションソフトウェアによって決まります。データ伝送における3つのシナリオは以下の通りです。

- マスタがデータを送信 - スレーブがダミーデータを送信
- マスタがデータを送信 - スレーブがデータを送信
- マスタがダミーデータを送信 - スレーブがデータを送信

図 20-2: SPI マスタ / スレーブの接続



# PIC18F97J60 ファミリ

## 20.3.5 マスタモード

マスタには SCKx の制御権があり、いつでもデータ転送を開始できます。スレーブ (図 20-2 のプロセッサ 2) がいつデータをブロードキャストするかは、ソフトウェア プロトコルに基づいてマスタが決定します。

マスタモードでは、SSPxBUF レジスタへの書き込み後、ただちにデータを送受信します。SPI を受信のみに使う場合、SDOx 出力を無効に (入力としてプログラム) してもかまいません。SSPxSR レジスタは、プログラムされたクロックレートで SDIx ピンの信号のシフト入力を継続します。1 バイト受信するごとに、通常受信バイトと同じように SSPxBUF レジスタに読み込みます (割り込みとステータスビットも必要に応じてセットされます)。これは、「ライン アクティビティ モニタ」モードのレシーバ アプリケーションに適しています。

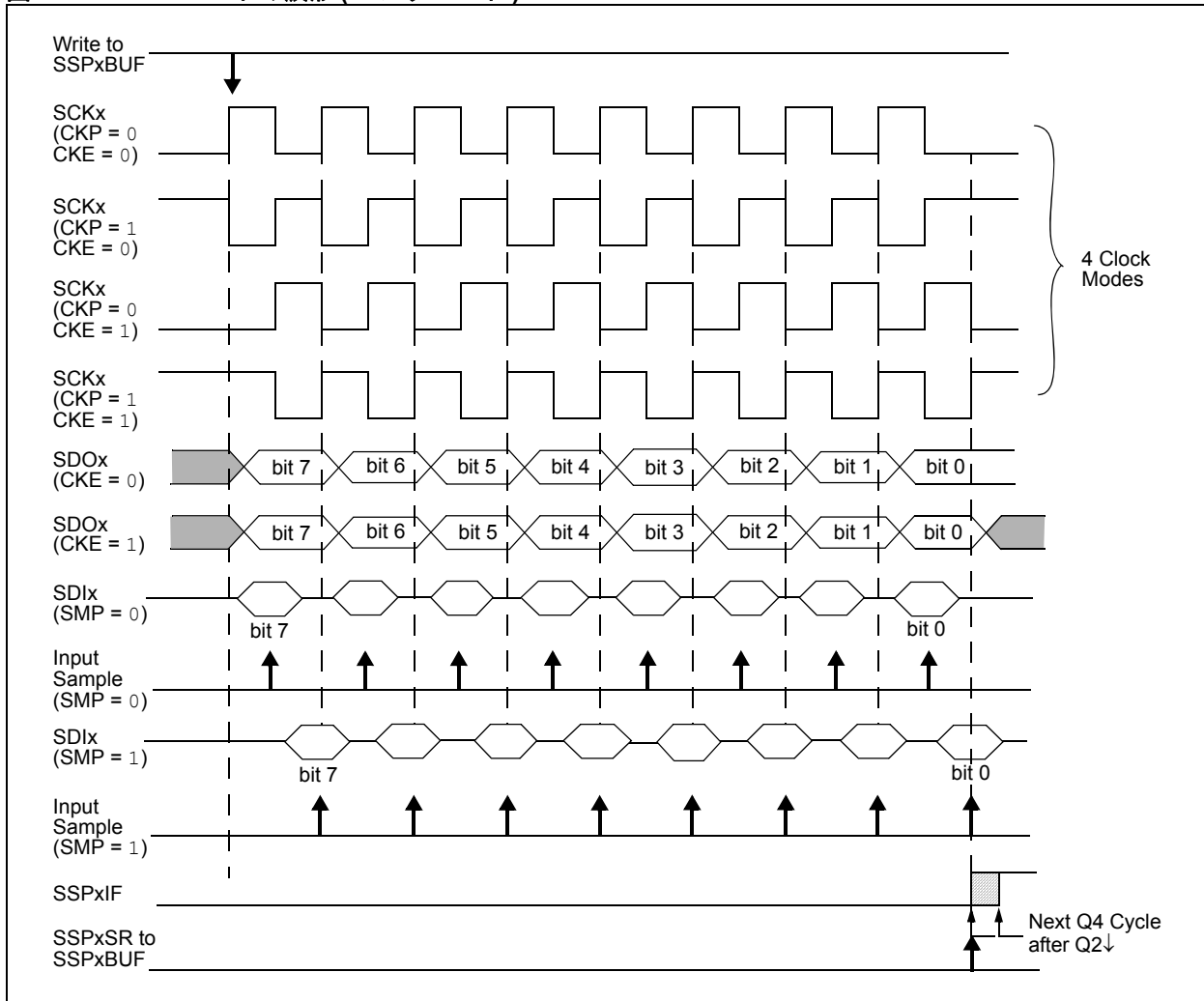
クロック極性は、CKP ビット (SSPxCON1<4>) を適切にプログラムする事によって選択します。これによって図 20-3、図 20-5、図 20-6 に示す SPI 通信の波形が得られます。この通信では MSb が最初に送信されます。マスタモードでは、SPI のクロックレート (ビットレート) を以下のいずれかにプログラムできます。

- Fosc/4 (または Tcy)
- Fosc/16 (または 4・Tcy)
- Fosc/64 (または 16・Tcy)
- Timer2 出力 /2

従って、最大データレートは 10.00 Mbps (@ 40 MHz) です。

図 20-3 にマスタモードの波形を示します。CKE ビットをセットした場合、SDOx のデータは SCKx にクロックエッジが現れる前に有効になります。この図には、SMP ビットの状態に応じた入力サンプリング時期の変化も示してあります。また、受信データが SSPxBUF に読み込まれるタイミングも示しています。

図 20-3: SPI モードの波形 (マスタモード)



## 20.3.6 スレーブモード

スレーブモードでは、SCKx に外部クロックパルスが印加されるとデータが送受信されます。最後のビットがラッチされると、SSPxIF 割り込みフラグビットがセットされます。

モジュールを SPI スレーブモードで動作させる前に、クロックラインのアイドル状態を一致させておく必要があります。クロックラインの状態は SCKx ピンを読み出す事で調べられます。アイドル状態は、CKP ビット (SSPxCON1<4>) によって決まります。

スレーブモードでは、SCKx ピンに外部クロック源からのクロックが供給されます。この外部クロックは、電氣的仕様に規定された High 期間と Low 期間の最小値を満たす必要があります。

スリープ中でもスレーブはデータを送受信できます。1 バイトを受信すると、デバイスはスリープから復帰します。

## 20.3.7 スレーブセレクトの同期

SSx ピンを使うと、同期スレーブモードを利用できます。SPI は、SSx ピン制御を有効 (SSPxCON1<3:0> = 04h) に設定したスレーブモードである必要があります。SSx ピンを Low にすると送受信が有効になり、SDOx

ピンが駆動されます。SSx ピンを High にすると、たとえ送信バイトの途中でであっても SDOx ピンの駆動は中断され、出力はフローティング状態に移行します。従って、アプリケーションによっては外付けのプルアップ/プルダウン抵抗の接続が望ましい場合があります。

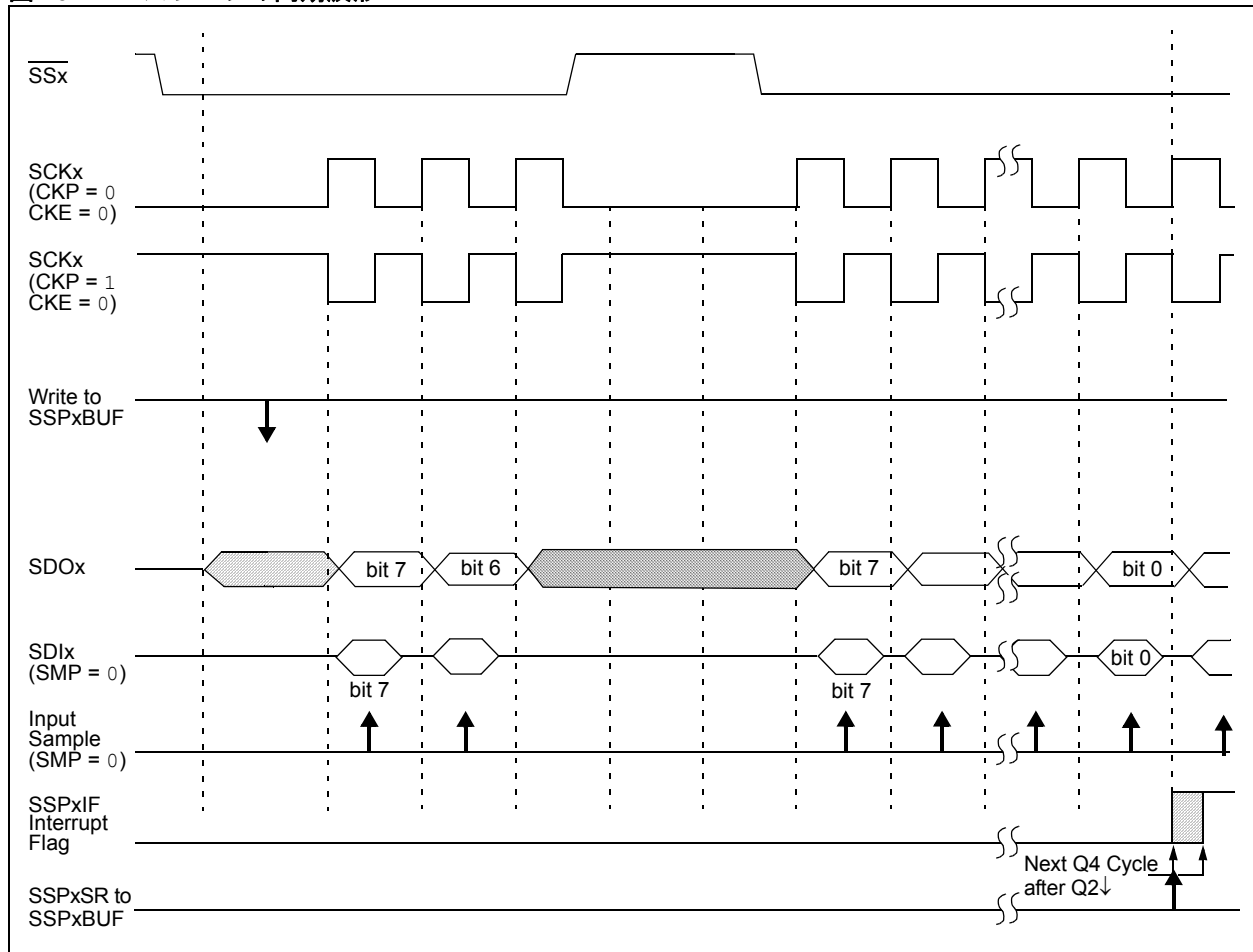
**Note 1:** SPIがSSxピン制御の有効なスレーブモード (SSPxCON1<3:0> = 0100) の場合、SSx ピンを VDD レベルにすると SPI モジュールがリセットされます。

**2:** CKE ビットをセットしたスレーブモードで SPI を使う場合、SSx ピンによる制御も有効に設定する必要があります。

SPI モジュールをリセットすると、ビットカウンタは強制的に「0」にクリアされます。リセットするには、SSx ピンを High レベルに駆動するか、SSPEN ビットをクリアします。

2線式通信をエミュレートするため、SDOx ピンを SDIx ピンに接続する事ができます。SPI をレシーバとして動作させる必要がある場合、SDOx ピンを入力として設定します。こうすると、SDOx からの送信が無効になります。SDIx はバス競合を発生する事がないため、常に入力のまま (SDIx 機能) としておく事ができます。

図 20-4: スレーブの同期波形



# PIC18F97J60 ファミリー

図 20-5: SPI モードの波形 (スレーブモード、CKE = 0)

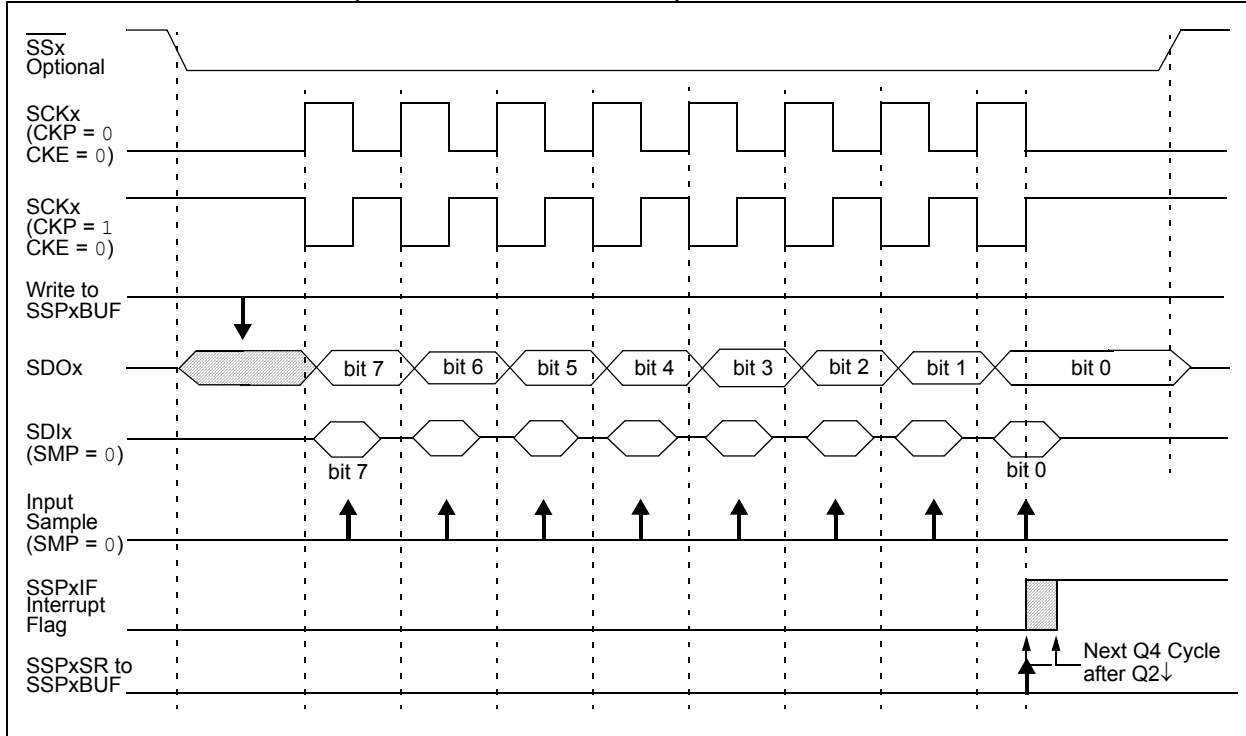
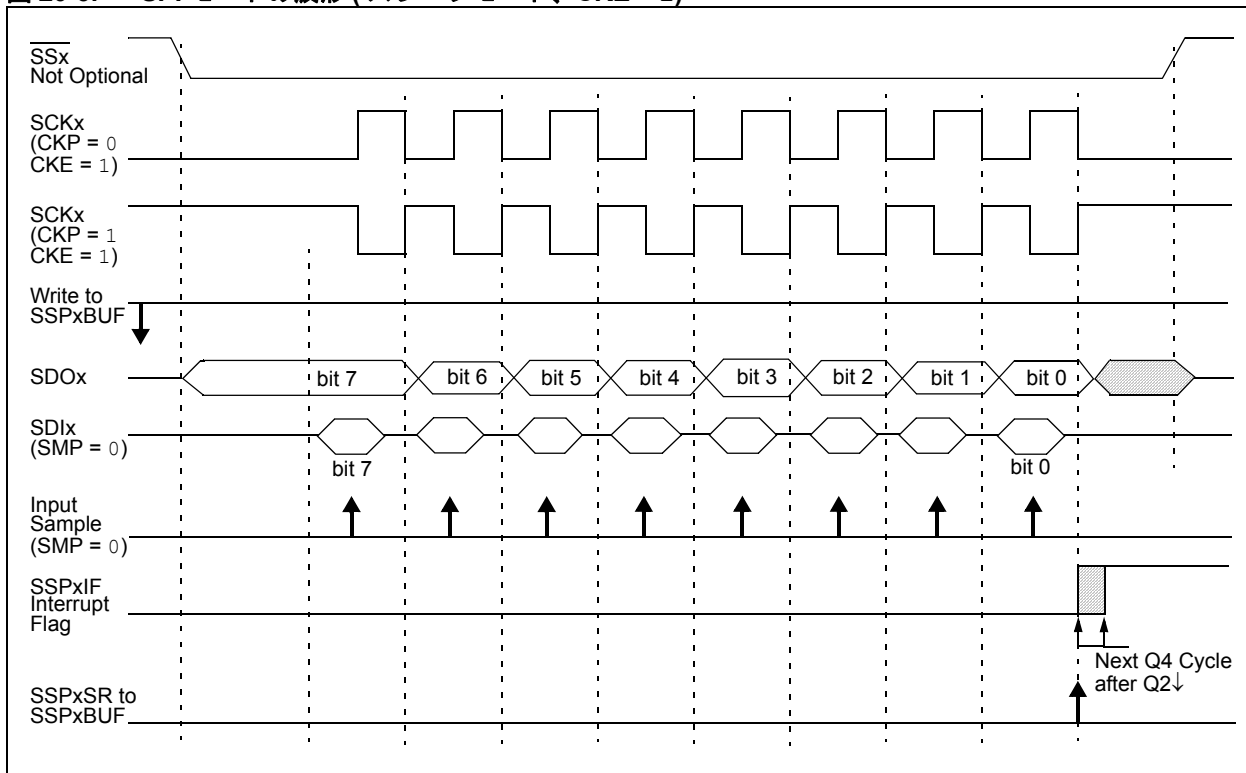


図 20-6: SPI モードの波形 (スレーブモード、CKE = 1)



## 20.3.8 電力管理モードにおける動作

SPI マスタモードでは、MSSP モジュールをフルパワーモード時と異なるクロックで動作させる事ができます。スリープの場合、全てのクロックが停止します。

アイドルでは、周辺モジュールにはクロックが供給されます。このクロックは、プライマリクロック源、セカンダリクロック (32.768 kHz の Timer1 オシレータ)、INTRC クロック源のいずれかから供給されます。詳細は、[セクション 3.7「クロック源とオシレータの切り換え」](#)を参照してください。

ほとんどの場合、マスタが SPI データをクロッキングする速度はそれほど重要ではありませんが、システムごとに評価しておく必要があります。

MSSP 割り込みを有効にした場合、マスタがデータの送信を完了すると、割り込みによってコントローラはスリープまたはアイドルから復帰します。スリープまたはアイドルを終了させたくない場合、MSSP 割り込みを無効にします。

スリープを選択すると全てのモジュールクロックが停止し、デバイスが復帰するまで送受信はその時点の状態を保ちます。デバイスが通常のランモードに戻ると、モジュールはデータの送受信を再開します。

SPI スレーブモードの場合、SPI 送受信シフトレジスタはデバイスと非同期で動作します。従って、デバイスがいずれの電力管理モードに移行しても、SPI 送受信シフトレジスタへのデータのシフト入力は継続できます。8 ビットを全て受信すると MSSP 割り込みフラグビットがセットされ、割り込みを有効にしている場合はデバイスが復帰します。

## 20.3.9 リセットの影響

リセットは MSSP モジュールを無効にし、現在の転送は中止されます。

## 20.3.10 バスモード互換性

表 20-1 に、標準 SPI モードと、制御ビット CKP/CKE の状態の関係を示します。

表 20-1: SPI バスモード

標準 SPI モードの表現	制御ビットの状態	
	CKP	CKE
0, 0	0	1
0, 1	0	0
1, 0	1	1
1, 1	1	0

データ サンプリングのタイミングを制御する SMP ビットもあります。

## 20.3.11 SPI のクロック速度とモジュールの関係

MSSP1 と MSSP2 はそれぞれ独立したモジュールであるため、別々のデータレートで同時に動作させる事ができます。各モジュールのデータレートは、SSPxCON1 レジスタの SSPM<3:0> ビットで設定します。

例外は、両方のモジュールがマスタモードで Timer2 をタイムベースとして使っている場合です。この場合、Timer2 の動作が変化すると両方の MSSP モジュールに同じ影響が現れます。2 つの異なるビットレートが必要な場合、どちらか一方のモジュールのタイムベースを Timer2 以外の 3 つから選択します。

# PIC18F97J60 ファミリ

表 20-2: SPI 動作に関連するレジスタ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	リセット 値の記載 ページ
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	69
PIR1	PSPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	71
PIE1	PSPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	71
IPR1	PSPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	71
PIR3	SSP2IF <sup>(1)</sup>	BCL2IF	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	71
PIE3	SSP2IE <sup>(1)</sup>	BCL2IE	RC2IE	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	71
IPR3	SSP2IP <sup>(1)</sup>	BCL2IP	RC2IP	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	71
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	71
TRISD	TRISD7 <sup>(1)</sup>	TRISD6 <sup>(1)</sup>	TRISD5 <sup>(1)</sup>	TRISD4 <sup>(1)</sup>	TRISD3	TRISD2	TRISD1	TRISD0	71
TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	TRISF0	71
SSP1BUF	MSSP1 受信バッファ / 送信レジスタ								70
SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	70
SSP1STAT	SMP	CKE	D/ $\bar{A}$	P	S	R/ $\bar{W}$	UA	BF	70
SSP2BUF	MSSP2 受信バッファ / 送信レジスタ								73
SSP2CON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	73
SSP2STAT	SMP	CKE	D/ $\bar{A}$	P	S	R/ $\bar{W}$	UA	BF	73

凡例: 網掛けの部分は SPI モードの MSSP では使いません。

**Note 1:** これらのビットが実装されているのは 100 ピンデバイスのみです。それ以外のデバイスでは「0」として読み出されます。

## 20.4 I<sup>2</sup>C モード

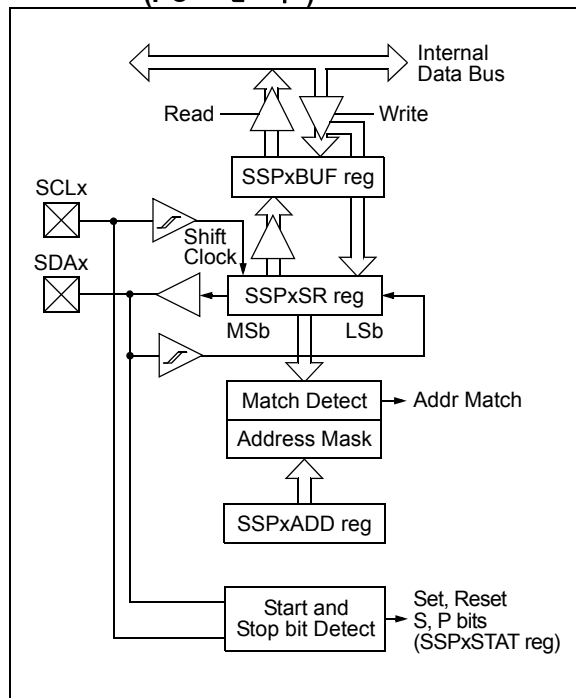
I<sup>2</sup>C モードの MSSP モジュールは、全てのマスタ / スレーブ機能 (ジェネラルコールのサポートを含む) を実装しており、空いているバスを判断するためのハードウェアによるスタート / ストップビット割り込みを生成できます (マルチマスタ機能)。MSSP モジュールは標準モードの仕様を実装すると共に、7 ビットと 10 ビットのアドレッシングにも対応しています。

データ転送には以下の 2 つのピンを使います。

- シリアルクロック (SCLx) – RC3/SCK1/SCL1 (100 ピンデバイスの場合は RD6/SCK2/SCL2)
- シリアルデータ (SDAx) – RC4/SDI1/SDA1 (100 ピンデバイスの場合は RD5/SDI2/SDA2)

これらのピンは、TRISC<4:3> または TRISD<5:4> ビットをセットして入力に設定しておく必要があります。

図 20-7: MSSP のブロック図 (I<sup>2</sup>C™ モード)



### 20.4.1 レジスタ

MSSP モジュールは、I<sup>2</sup>C 動作時に以下の 6 つのレジスタを使います。

- MSSPx 制御レジスタ 1 (SSPxCON1)
- MSSPx 制御レジスタ 2 (SSPxCON2)
- MSSPx ステータス レジスタ (SSPxSTAT)
- MSSPx 受信バッファ / 送信レジスタ (SSPxBUF)
- MSSPx シフトレジスタ (SSPxSR) – 直接アクセスは不可
- MSSPx アドレスレジスタ (SSPxADD)

SSPxCON1、SSPxCON2、SSPxSTAT は、I<sup>2</sup>C モード動作時の制御レジスタとステータス レジスタです。SSPxCON1 と SSPxCON2 は読み書き可能なレジスタです。SSPxSTAT の下位 6 ビットは読み出し専用です。SSPxSTAT の上位 2 ビットは読み書き可能です。

SSPxCON2 レジスタのビットの多くは、モジュールの動作がマスタモードかスレーブモードかによって異なる働きをします。SSPxCON2<5:1> に関しては、スレーブモードではビット名も異なります。SSPxCON2 レジスタは、マスタモード時のもの (レジスタ 20-5) とスレーブモード時のもの (レジスタ 20-6) を分けて記載しています。

SSPxSR はデータをシフト入出力するためのシフトレジスタです。SSPxBUF はデータバイトの読み書き用バッファレジスタです。

MSSP を I<sup>2</sup>C スレーブモードに設定した場合、SSPxADD レジスタにはスレーブデバイスのアドレスを格納します。MSSP をマスタモードに設定した場合、SSPxADD レジスタの下位 7 ビットは baud レート ジェネレータの再読み込み値です。

受信動作の場合、SSPxSR と SSPxBUF が連動して、ダブルバッファ レシーバを構成します。SSPxSR が 1 バイトを全て受信すると SSPxBUF に転送され、SSPxIF 割り込みがセットされます。

送信時の SSPxBUF はダブルバッファを構成しません。SSPxBUF に書き込むと、SSPxBUF と SSPxSR の両方に同じ内容が書き込まれます。

# PIC18F97J60 ファミリ

## レジスタ 20-3: SSPxSTAT: MSSPx ステータス レジスタ (I<sup>2</sup>C™ モード)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/A	P <sup>(1)</sup>	S <sup>(1)</sup>	R/W <sup>(2,3)</sup>	UA	BF
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

- bit 7      **SMP:** スルーレート制御ビット  
マスタまたはスレーブモード:  
 1 = 標準速度モード (100 kHz と 1 MHz) でスルーレート制御は無効である  
 0 = 高速モード (400 kHz) でスルーレート制御は有効である
- bit 6      **CKE:** SMBus 選択ビット  
マスタまたはスレーブモード:  
 1 = SMBus 仕様に準拠した入力は無効である  
 0 = SMBus 仕様に準拠した入力は無効である
- bit 5      **D/A:** データ / アドレスビット  
マスタモード:  
 予約済み  
スレーブモード:  
 1 = 最後に送受信したバイトがデータであることを示す  
 0 = 最後に送受信したバイトがアドレスであることを示す
- bit 4      **P:** ストップビット<sup>(1)</sup>  
 1 = 最後にストップビットが検出されたことを示す  
 0 = 最後にストップビットは検出されなかったことを示す
- bit 3      **S:** スタートビット<sup>(1)</sup>  
 1 = 最後にスタートビットが検出されたことを示す  
 0 = 最後にスタートビットは検出されなかったことを示す
- bit 2      **R/W:** 読み出し / 書き込み情報ビット (I<sup>2</sup>C モードのみ)<sup>(2, 3)</sup>  
スレーブモード:  
 1 = 読み出し  
 0 = 書き込み  
マスタモード:  
 1 = 送信中である  
 0 = 送信中ではない
- bit 1      **UA:** 更新アドレスビット (10 ビットのスレーブモードのみ)  
 1 = ユーザが SSPxADD レジスタ内のアドレスを更新する必要があることを示す  
 0 = アドレスの更新は不要である
- bit 0      **BF:** バッファフル ステータスビット  
送信モード:  
 1 = SSPxBUF がフル  
 0 = SSPxBUF がエンプティ  
受信モード:  
 1 = SSPxBUF がフル (ACK とストップビットを除く)  
 0 = SSPxBUF がエンプティ (ACK とストップビットを除く)

- Note 1:** このビットは、リセット時と SSPEN のクリア時にクリアされます。
- Note 2:** このビットは、最後のアドレス一致後の R/W ビット情報を保持します。このビットが有効な期間は、アドレス一致から次のスタートビット、ストップビットまたは非 ACK ビットまでです。
- Note 3:** このビットと SEN、RSEN、PEN、RCEN、ACKEN のいずれかとの論理和により、MSSP がアクティブモードにあるかどうかを示します。



# PIC18F97J60 ファミリ

## レジスタ 20-4: SSPxCON1: MSSPx 制御レジスタ 1 (I<sup>2</sup>C™ モード)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

- bit 7      **WCOL:** 書き込みコリジョン検出ビット  
マスタ送信モード:  
 1 = 送信を開始するための I<sup>2</sup>C 条件が有効でない状態で SSPxBUF レジスタへの書き込みを試みた (ソフトウェアによるクリアが必要)  
 0 = コリジョンは発生していない  
スレーブ送信モード:  
 1 = 前のワードの送信中に SSPxBUF レジスタへの書き込みが実行された (ソフトウェアによるクリアが必要)  
 0 = コリジョンは発生していない  
受信モード (マスタまたはスレーブモード):  
 ドントケアビット
- bit 6      **SSPOV:** 受信オーバーフロー インジケータ ビット  
受信モード:  
 1 = SSPxBUF レジスタに前のバイトが残っている間に次のバイトを受信した (ソフトウェアによるクリアが必要)  
 0 = オーバーフローは発生していない  
送信モード:  
 送信モードではドントケアビット
- bit 5      **SSPEN:** MSSP イネーブルビット  
 1 = シリアルポートを有効にし、SDAx ピンと SCLx ピンをシリアルポート ピンとして設定する <sup>(1)</sup>  
 0 = シリアルポートを無効にし、上記のピンを I/O ポートピンとして設定する <sup>(1)</sup>
- bit 4      **CKP:** SCKx 解放制御ビット  
スレーブモード:  
 1 = クロックを解放する  
 0 = クロックを Low に保持 (クロック ストレッチ) してデータ セットアップ時間を確保する  
マスタモード:  
 このモードでは未使用
- bit 3-0      **SSPM<3:0>:** MSSP モード選択ビット  
 1111 = I<sup>2</sup>C スレーブモード、10 ビットアドレス、スタートビットとストップビットの割り込みを有効にする <sup>(2)</sup>  
 1110 = I<sup>2</sup>C スレーブモード、7 ビットアドレス、スタートビットとストップビットの割り込みを有効にする <sup>(2)</sup>  
 1011 = I<sup>2</sup>C ファームウェア制御マスタモード (スレーブアイドル) <sup>(2)</sup>  
 1000 = I<sup>2</sup>C マスタモード、クロック = Fosc/(4 \* (SSPADD + 1)) <sup>(2)</sup>  
 0111 = I<sup>2</sup>C スレーブモード、10 ビットアドレス <sup>(2)</sup>  
 0110 = I<sup>2</sup>C スレーブモード、7 ビットアドレス <sup>(2)</sup>

- Note 1:** 有効にした場合、SDAx ピンと SCLx ピンを入力として設定する必要があります。  
**2:** ここに具体的に値を示していないビットの組み合わせは、予約済みか、SPI モードでのみ使うものです。

# PIC18F97J60 ファミリ

## レジスタ 20-5: SSPxCON2: MSSPx 制御レジスタ 2 (I<sup>2</sup>C™ マスタモード)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GCEN	ACKSTAT	ACKDT <sup>(1)</sup>	ACKEN <sup>(2)</sup>	RCEN <sup>(2)</sup>	PEN <sup>(2)</sup>	RSEN <sup>(2)</sup>	SEN <sup>(2)</sup>
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

- bit 7      **GCEN:** ジェネラルコール イネーブルビット (スレーブモードのみ)  
 マスタモードでは使わない
- bit 6      **ACKSTAT:** 肯定応答 (ACK) ステータスビット (マスタ送信モードのみ)  
 1 = スレーブから ACK を受信していない  
 0 = スレーブから ACK を受信した
- bit 5      **ACKDT:** 肯定応答 (ACK) データビット (マスタ受信モードのみ)<sup>(1)</sup>  
 1 = ACK なし  
 0 = ACK あり
- bit 4      **ACKEN:** 肯定応答 (ACK) シーケンス イネーブルビット<sup>(2)</sup>  
 1 = SDAx ピンと SCLx ピンで ACK シーケンスを開始し、ACKDT データビットを送信する。ハードウェアによって自動的にクリアされる  
 0 = ACK シーケンスをアイドルにする
- bit 3      **RCEN:** 受信イネーブルビット (マスタ受信モードのみ)<sup>(2)</sup>  
 1 = I<sup>2</sup>C の受信モードを有効にする  
 0 = 受信をアイドルにする
- bit 2      **PEN:** ストップ条件イネーブルビット<sup>(2)</sup>  
 1 = SDAx ピンと SCLx ピンでストップ条件を開始する。ハードウェアによって自動的にクリアされる  
 0 = ストップ条件をアイドルにする
- bit 1      **RSEN:** 反復スタート条件イネーブルビット<sup>(2)</sup>  
 1 = SDAx ピンと SCLx ピンで反復スタート条件を開始する。ハードウェアによって自動的にクリアされる  
 0 = 反復スタート条件をアイドルにする
- bit 0      **SEN:** スタート条件イネーブル/ストレッチ イネーブルビット<sup>(2)</sup>  
 1 = SDAx ピンと SCLx ピンでスタート条件を開始する。ハードウェアによって自動的にクリアされる  
 0 = スタート条件をアイドルにする

- Note 1:** 受信終了時に、ユーザが ACK シーケンスを始める時に送信される値です。  
**2:** I<sup>2</sup>C モジュールがアクティブな場合、これらのビットがセットされず (スプーリングなし)、SSPxBUF に書き込む事ができない (または SSPxBUF への書き込みが無効になる) 場合があります。

# PIC18F97J60 ファミリ

レジスタ 20-6: SSPxCON2: MSSPx 制御レジスタ 2 (I<sup>2</sup>C™ スレーブモード)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GCEN	ACKSTAT	ADMSK5	ADMSK4	ADMSK3	ADMSK2	ADMSK1	SEN <sup>(1)</sup>
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

- bit 7      **GCEN:** ジェネラルコール イネーブルビット (スレーブモードのみ)  
 1 = SSPxSR でジェネラルコール アドレス (0000h) を受信した時の割り込みを有効にする  
 0 = ジェネラルコール アドレスを無効にする
- bit 6      **ACKSTAT:** 肯定応答 (ACK) ステータスビット  
 スレーブモードでは使わない
- bit 5-2      **ADMSK5:ADMSK2:** スレーブアドレス マスク選択ビット  
 1 = SSPxADD で対応するビットのマスクを有効にする  
 0 = SSPxADD で対応するビットのマスクを無効にする
- bit 1      **ADMSK1:** スレーブアドレス下位マスク選択ビット  
7ビットアドレッシングモードの場合:  
 1 = SSPxADD<1> のマスクのみ有効にする  
 0 = SSPxADD<1> のマスクのみ無効にする  
10ビットアドレッシングモードの場合:  
 1 = SSPxADD<1:0> のマスクを有効にする  
 0 = SSPxADD<1:0> のマスクを無効にする
- bit 0      **SEN:** ストレッチ イネーブルビット <sup>(1)</sup>  
 1 = スレーブ送信とスレーブ受信の両方でクロック ストレッチを有効にする(ストレッチ イネーブル)  
 0 = クロック ストレッチを無効にする

**Note 1:** I<sup>2</sup>C モジュールがアクティブな場合、このビットがセットされず (スプーリングなし)、SSPxBUF に書き込む事ができない (または SSPxBUF への書き込みが無効になる) 場合があります。

## 20.4.2 動作

MSSP モジュールの動作は、MSSP イネーブルビット SSPEN (SSPxCON1<5>) をセットすると有効になります。

I<sup>2</sup>C の動作は、SSPxCON1 レジスタで制御できます。以下の I<sup>2</sup>C のモードは、SSPxCON1<3:0> の 4 つのモード選択ビットで選択できます。

- I<sup>2</sup>C マスタモード、  
クロック = (Fosc/4) x (SSPxADD + 1)
- I<sup>2</sup>C スレーブモード、7 ビットアドレス
- I<sup>2</sup>C スレーブモード、10 ビットアドレス
- I<sup>2</sup>C スレーブモード、7 ビットアドレス、  
スタート/ストップビットの割り込み有効
- I<sup>2</sup>C スレーブモード、10 ビットアドレス、  
スタート/ストップビットの割り込み有効
- I<sup>2</sup>C ファームウェア制御マスタモード、  
スレーブアイドル

SSPENビットをセットした状態でいずれかのI<sup>2</sup>Cモードを選択すると、TRISC または TRISD ビットをセットして SCLx ピンと SDAx ピンを入力に設定している場合、これらのピンは強制的にオープンドレインに設定されます。モジュールを正しく動作させるには、SCLx ピンと SDAx ピンにプルアップ抵抗を外付けする必要があります。

## 20.4.3 スレーブモード

スレーブモードでは、SCLx ピンと SDAx ピンを入力に設定 (TRISC<4:3> または TRISD<5:4> をセット) する必要があります。MSSP モジュールは必要に応じて、出力データによって入力状態を無効にします (スレーブトランスミッタ)。

I<sup>2</sup>C スレーブモードのハードウェアは、アドレス完全一致で常に割り込みを発生させます。また、アドレスマスクを使うと複数のアドレスに対してハードウェア割り込みを発生させる事もできます (7 ビットアドレッシングでは最大 31 アドレス、10 ビットアドレッシングでは最大 63 アドレス)。モード選択ビットを設定する事で、スタート/ストップビットによる割り込みも選択できます。

アドレスが一致した場合、またはアドレス一致後に転送データの受信が完了した時点で、ハードウェアは自動的に肯定応答 (ACK) パルスを生成し、現在 SSPxSR レジスタに格納されている受信データを SSPxBUF レジスタに読み込みます。

以下の条件が 1 つでも満たされると、MSSP モジュールは上記の ACK パルスを発生しません。

- 転送を受信する前にバッファフル ビット BF (SSPxSTAT<0>) がセットされた
- 転送を受信する前に MSSP オーバーフロー ビット SSPOV (SSPxCON1<6>) がセットされた

この場合、SSPxSR レジスタの値は SSPxBUF に読み込まれませんが、SSPxIF ビットはセットされます。BF ビットは SSPxBUF レジスタを読み出す事でクリアされますが、SSPOV ビットはソフトウェアによるクリアが必要です。

正しく動作させるには、SCLx クロック入力の High 期間と Low 期間が最小値の仕様を満たしている必要があります。I<sup>2</sup>C 仕様の High/Low 期間の仕様条件、MSSP モジュールに関する要件は、パラメータ 100 とパラメータ 101 に記載しています。

### 20.4.3.1 アドレッシング

MSSP モジュールを有効にすると、モジュールはスタート条件の発生を待ちます。スタート条件が発生すると、8 ビットが SSPxSR レジスタにシフト入力されます。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ルされます。次に、レジスタ SSPxSR<7:1> の値と SSPxADD レジスタの値を比較します。アドレスは、8 番目のクロックパルス (SCLx) の立ち下がりエッジで比較されます。アドレスが一致し、BF ビットと SSPOV ビットがクリアされている場合、以下のように動作します。

1. SSPxSR レジスタの値が SSPxBUF レジスタに読み込まれる。
2. バッファフル ビット BF がセットされる。
3. ACK パルスが生成される。
4. 9 番目の SCLx パルスの立ち下がりエッジで MSSP 割り込みフラグビット SSPxIF がセットされる (割り込み有効時は割り込みが発生する)。

10 ビット アドレッシング モードの場合、スレーブは 2 バイトのアドレスを受信する必要があります。1 バイト目のアドレスバイトの上位 5 ビット (MSb) が、10 ビットアドレスであるかどうかを示します。スレーブデバイスがアドレスの 2 バイト目を受信できるように、R/W ビット (SSPxSTAT<2>) は書き込みを設定しておく必要があります。10 ビットアドレスの場合、1 バイト目の値は「111110 A9 A8 0」です。この「A9」と「A8」がアドレスの上位 2 ビットです。10 ビットアドレス モードの動作シーケンスは以下の通りです。ただし、ステップ 7 ~ 9 はスレーブトランスミッタの場合に実行されます。

1. 最初のアドレスバイト(上位)を受信する(SSPxIF、BF、UA ビットがセットされる)。
2. SSPxADD レジスタを 2 番目のアドレスバイト (下位) の値で更新する (UA ビットがクリアされ SCLx ラインが解放される)。
3. SSPxBUF レジスタを読み出し (BF ビットがクリアされる)、フラグビット SSPxIF をクリアする。
4. 2 番目のアドレスバイト (下位) を受信する (SSPxIF、BF、UA ビットがセットされる)。
5. SSPxADD レジスタを最初のアドレスバイト (上位) の値で更新する。一致によって SCLx ラインが解放されると、UA ビットがクリアされる。
6. SSPxBUF レジスタを読み出し (BF ビットがクリアされる)、フラグビット SSPxIF をクリアする。
7. 反復スタート条件を受信する。
8. 最初のアドレスバイト(上位)を受信する(SSPxIF、BF ビットがセットされる)。
9. SSPxBUF レジスタを読み出し (BF ビットがクリアされる)、フラグビット SSPxIF をクリアする。

## 20.4.3.2 アドレスマスク

特定のアドレスビットをマスクすると、そのビットは「ドントケア」となります。アドレスビットを1つマスクすると、2つのアドレスにACKを返して割り込みが発生します。一度に複数のアドレスビットをマスクする事もできます。その場合、7ビットモードでは最大31アドレス、10ビットモードでは最大63アドレスに対してACKを返す事ができます(例 20-2参照)。

アドレスマスクを使っても使わなくてもI<sup>2</sup>Cスレーブの動作は変わりません。しかし、アドレスマスクを使った場合、I<sup>2</sup>Cスレーブは複数のアドレスにACKを返して割り込みを発生させます。この状況が発生したら、SSPxBUFを確認してどのアドレスが割り込みを発生したかを調べる必要があります。

7ビットアドレッシングモードでは、アドレスマスクビットADMSK<5:1>(SSPxCON2<5:1>)の各ビットを使って、SSPxADDレジスタの対応するアドレスビットをマスクします。ADMSKビットをセット(ADMSK<n>=1)すると、対応するアドレスビットは無視されます(SSPxADD<n>=x)。ADMSKビットがセットされていない位置のアドレスが一致すれば、モジュールはACKを返します。

10ビットアドレッシングモードでは、ADMSK<5:2>の各ビットを使って、SSPxADDレジスタの対応するアドレスビットをマスクします。また、ADMSK1はアドレスの下位2ビット(SSPxADD<1:0>)を同時にマスクします。ADMSKビットをセット(ADMSK<n>=1)すると、対応するアドレスビットは無視されます(SSPxADD<n>=x)。また、10ビットアドレッシングモードでは上位アドレスビットはSSPxADDレジスタの一部のビットを再利用します。アドレスマスクビットはこれらの上位ビットには適用されず、下位アドレスビットのみに適用されます。

**Note 1:** ADMSK1はアドレスの下位2ビットをマスクします。

**2:** アドレスの上位2ビットはアドレスマスクの影響を受けません。

### 例 20-2: アドレスマスクの例

#### 7ビットアドレッシング:

SSPxADD<7:1> = A0h (1010000) (SSPxADD<0>は「0」と仮定)

ADMSK<5:1> = 00111

ACKが返されるアドレス: A0h, A2h, A4h, A6h, A8h, AAh, ACh, AEh

#### 10ビットアドレッシング:

SSPxADD<7:0> = A0h (10100000) (この例では、アドレスの上位2ビットはマスクの影響を受けないため無視しています)

ADMSK<5:1> = 00111

ACKが返されるアドレス: A0h, A1h, A2h, A3h, A4h, A5h, A6h, A7h, A8h, A9h, AAh, ABh, ACh, ADh, AEh, AFh

# PIC18F97J60 ファミリ

## 20.4.3.3 受信

R/W ビットがクリアされたアドレスバイトとアドレスが一致した場合、SSPxSTAT レジスタの R/W ビットがクリアされます。受信したアドレスは SSPxBUF レジスタに読み込まれ、SDAx ラインが Low に保持されます (ACK)。

アドレスバイトのオーバーフロー条件が真の場合、ACK パルス (ACK) は生成されません。オーバーフロー条件とは、BF ビット (SSPxSTAT<0>) または SSPOV ビット (SSPxCON1<6>) のどちらかがセットされている状態として定義されます。

1 データバイトが転送されるたびに MSSP 割り込みが生成されます。割り込みフラグビット SSPxIF はソフトウェアでクリアする必要があります。バイトの状態は、SSPxSTAT レジスタを使って判定します。

SEN が有効な場合 (SSPxCON2<0> = 1)、SCKx/SCLx (RC3 または RD6) はデータ転送後に Low のまま保持されます (クロック ストレッチ)。クロックを解放するには、CKP ビット (SSPxCON1<4>) をセットする必要があります。詳細は、[セクション 20.4.4「クロック ストレッチ」](#) を参照してください。

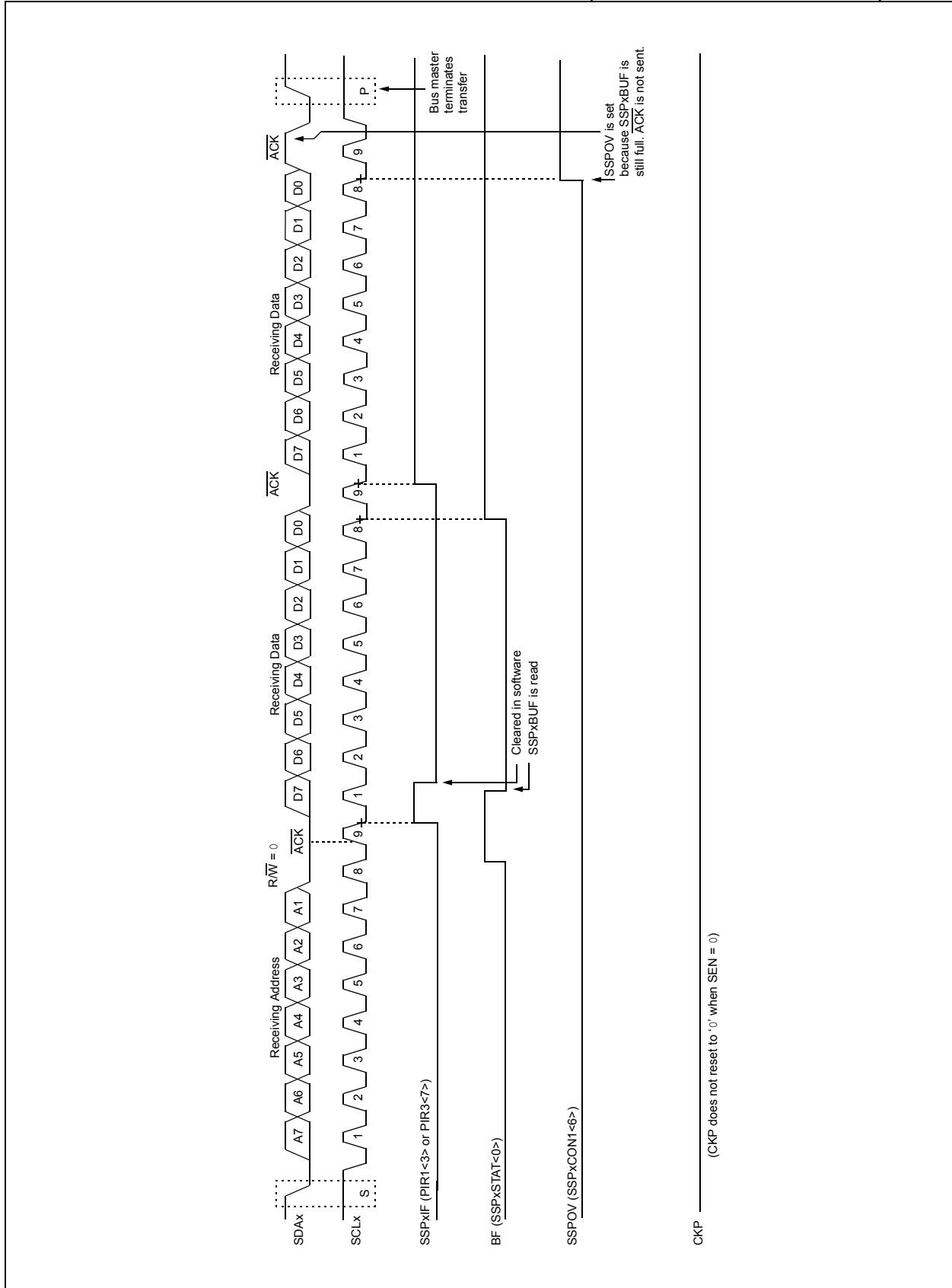
## 20.4.3.4 送信

受信したアドレスバイトの R/W ビットがセットされた状態でアドレスが一致した場合、SSPxSTAT レジスタの R/W ビットがセットされます。受信したアドレスは SSPxBUF レジスタに読み込まれます。9 番目のビットで ACK パルスが送信され、SEN の状態に関わらず RC3 または RD6 が Low に保持されます ([セクション 20.4.4「クロック ストレッチ」](#) 参照)。このクロック ストレッチ動作によって、マスタはスレーブ側で送信データの準備が整うまで、次のクロックパルスを送信できません。送信データは SSPxBUF レジスタに読み込まれ、さらに SSPxSR レジスタにも読み込まれます。次に、CKP ビット (SSPxCON1<4>) をセットして RC3 または RD6 ピンを有効にします。SCLx 入力の各立ち下がりがエッジで、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 期間中の SDAx 信号が有効データとしてサンプルされます ([図 20-10](#))。

マスタレーバからの  $\overline{\text{ACK}}$  パルスは、SCLx 入力パルスの 9 番目の立ち上がりエッジでラッチされます。SDAx ラインが High (非 ACK) の場合、データ送信は完了です。この場合、スレーブが ACK をラッチするとスレーブのロジックがリセットされ (SSPxSTAT レジスタをリセット)、スレーブは次のスタートビットの監視を開始します。SDAx ラインが Low (ACK) の場合、次の送信データが SSPxBUF レジスタに読み込まれます。ここでも、CKP ビットをセットして RC3 または RD6 ピンを有効にする必要があります。

1 データバイトが転送されるたびに MSSP 割り込みが生成されます。SSPxIF ビットはソフトウェアでクリアする必要があり、バイトの状態を判断するには SSPxSTAT レジスタを使います。SSPxIF ビットは 9 番目のクロックパルスの立ち下がりがエッジでセットされます。

図 20-8: SEN = 0 の場合の I<sup>2</sup>C™ スレーブモードのタイミング (7 ビットアドレスモードの受信)



# PIC18F97J60 ファミリ

図 20-9: SEN = 0、ADMSK<5:1> = 01011 の場合の I<sup>2</sup>C™ スレーブモードのタイミング (7 ビットアドレスモードの受信)

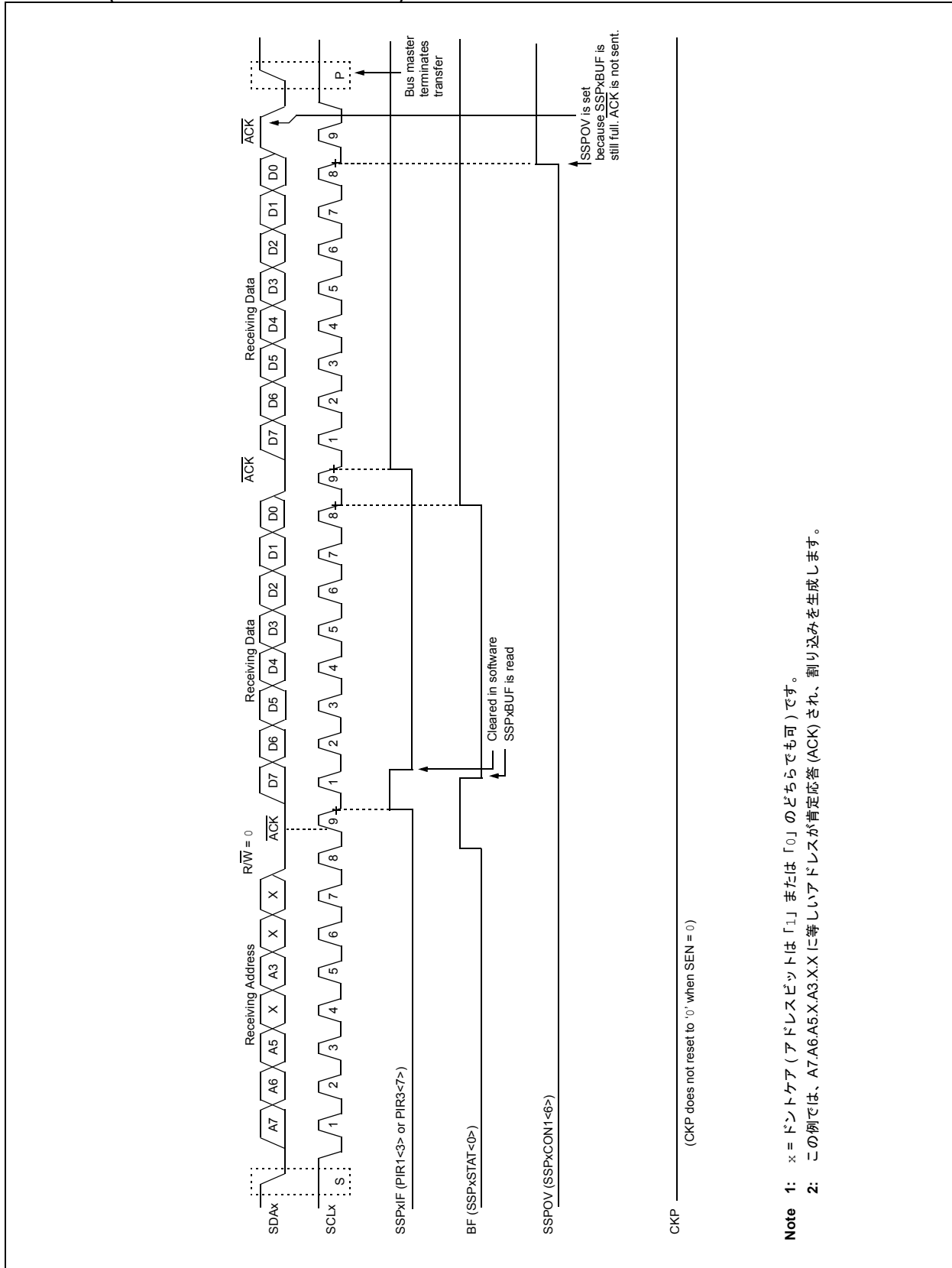
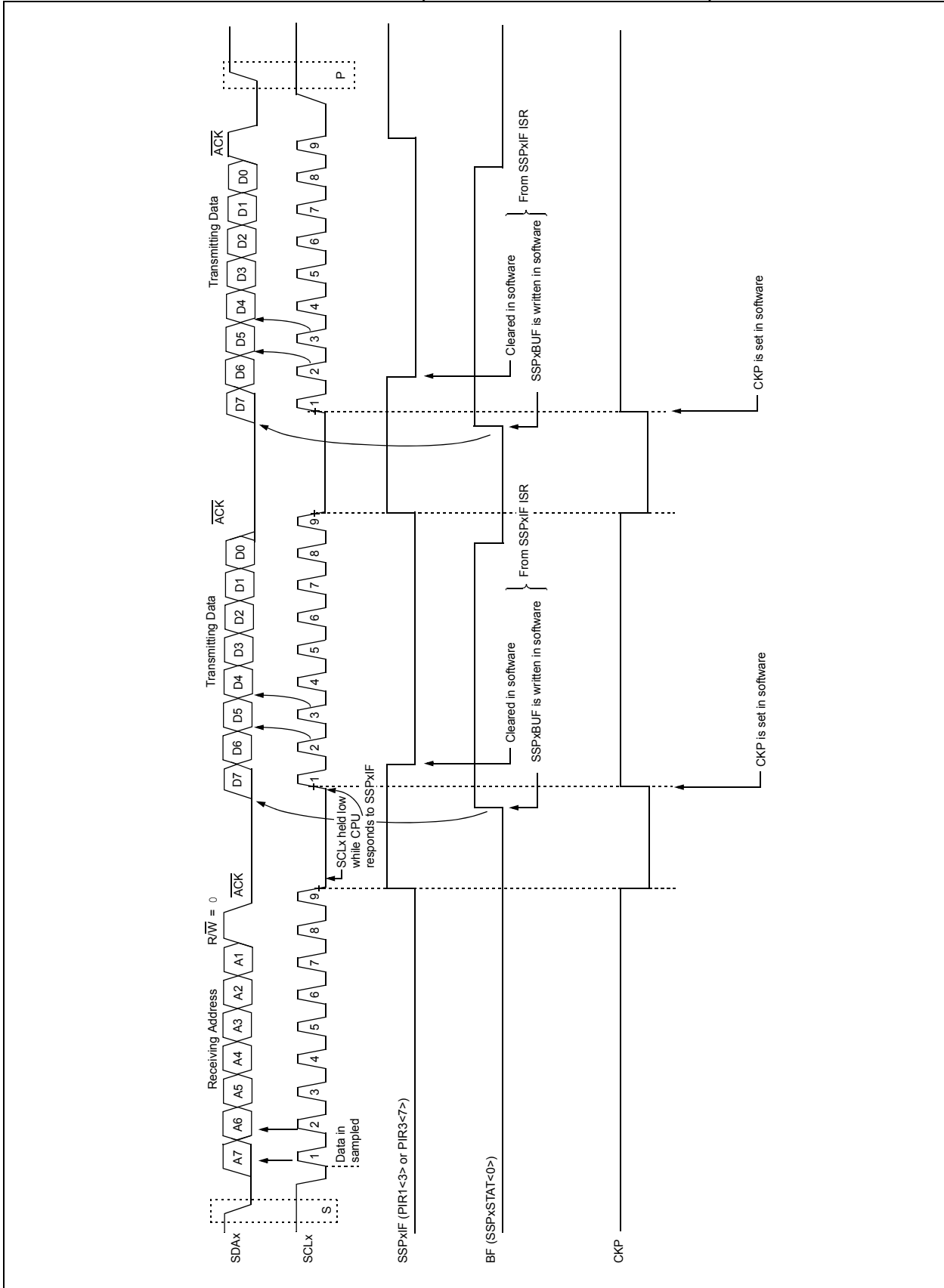




図 20-10: I<sup>2</sup>C™ スレーブモードのタイミング (7 ビットアドレスモードの送信)



# PIC18F97J60 ファミリ

図 20-11: SEN = 0 の場合の I<sup>2</sup>C™ スレーブモードのタイミング (10 ビットアドレスモードの受信)

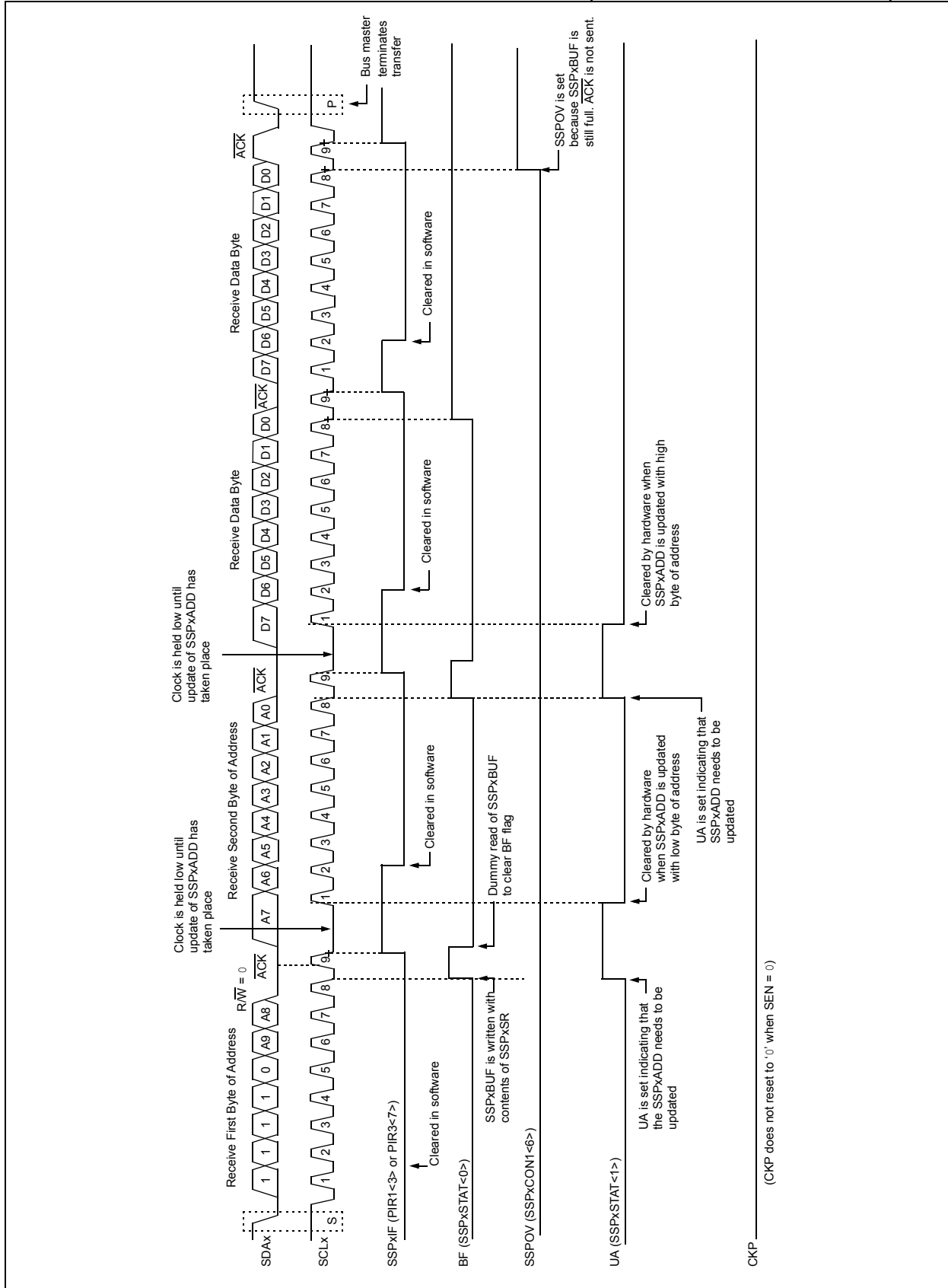
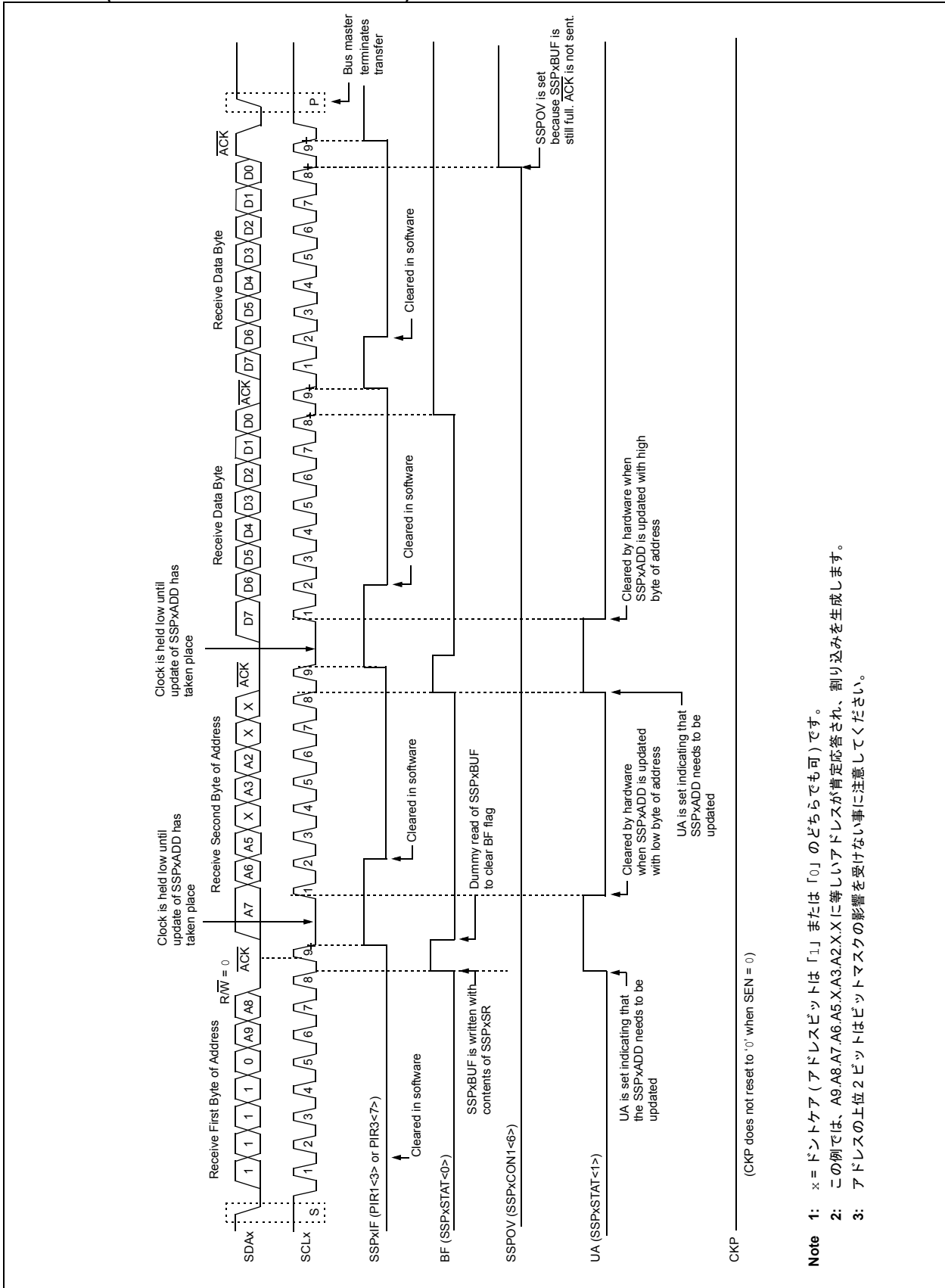
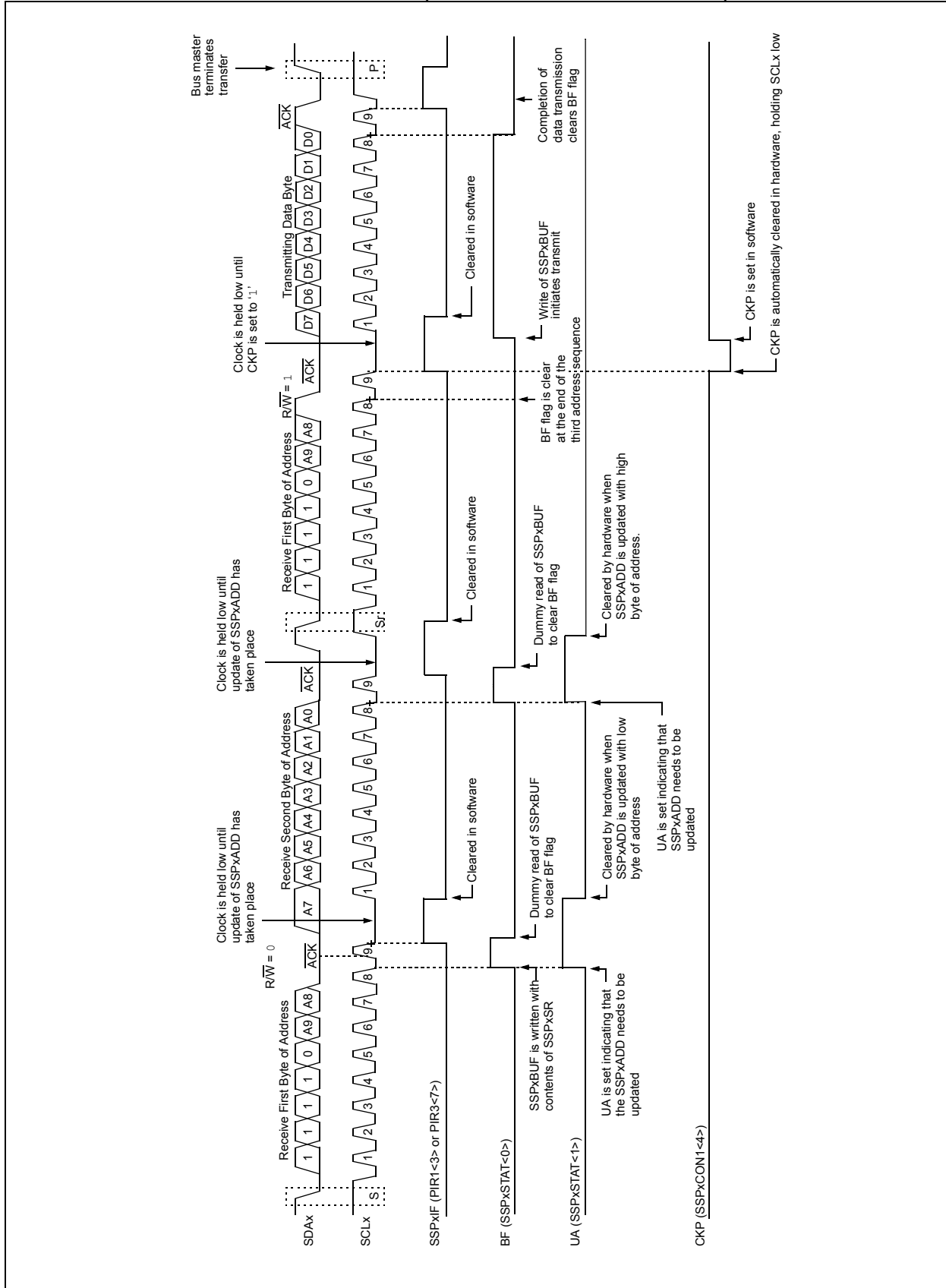


図 20-12: SEN = 0、ADMSK<5:1> = 01001 の場合の I<sup>2</sup>C™ スレーブモードのタイミング (10 ビットアドレスモードの受信)



# PIC18F97J60 ファミリ

図 20-13: I<sup>2</sup>C™ スレーブモードのタイミング (10 ビットアドレス モードの送信)



## 20.4.4 クロック ストレッチ

7ビットと10ビットのスレーブモードは、どちらも送信シーケンス実行中の自動クロック ストレッチ機能を実装しています。

受信時のクロック ストレッチは、SEN ビット (SSPxCON2<0>) で制御できます。SEN ビットをセットすると、1回のデータ受信シーケンスが終了するごとに SCLx ピンが Low に保持されます。

### 20.4.4.1 7ビットスレーブ受信モードのクロック ストレッチ (SEN = 1)

7ビットスレーブ受信モードでは、 $\overline{ACK}$  シーケンスの最後、9番目のクロックの立ち下がりエッジで BF ビットがセットされている場合、SSPxCON1 レジスタの CKP ビットが自動的にクリアされます。これによって、SCLx 出力が強制的に Low に保持されます。CKP を「0」にクリアすると、SCLx ラインが Low にアサートされます。受信を継続するには、ユーザ割り込みサービスマチン (ISR) 内で CKP ビットをセットする必要があります。SCLx ラインを Low に保持する事で、マスタデバイスが次の受信シーケンスを開始する前に、ユーザ ISR の処理と SSPxBUF の内容読み出しの時間を確保できます。これによって、バッファ オーバーランを防止します ( 図 20-15 参照 )。

**Note 1:** ユーザが9番目のクロックの立ち下がりエッジよりも前に SSPxBUF の内容を読み出して BF ビットがクリアされた場合、CKP ビットはクリアされず、クロック ストレッチは実行されません。

**2:** CKP ビットは、BF ビットの状態に関わらずソフトウェアでセットできます。ユーザ ISR 内で次の受信シーケンスの前に BF ビットをクリアする場合、オーバーフロー条件を生じないように慎重にコードを実装する必要があります。

### 20.4.4.2 10ビットスレーブ受信モードのクロック ストレッチ (SEN = 1)

10ビットスレーブ受信モードのアドレス シーケンスでは、クロック ストレッチが自動的に実行されますが、CKP はクリアされません。この場合、もし9番目のクロック後に UA ビットがセットされていると、クロック ストレッチが開始します。UA ビットがセットされるのは、10ビットアドレスの上位バイトを受信した後に、R/W ビットが「0」にクリアされている下位バイトを受信した後です。クロック ラインは

SSPxADD を更新した時点で解放されます。クロック ストレッチは、7ビットモードと同様に、1回のデータ受信シーケンスごとに発生します。

**Note:** ユーザが UA ビットをポーリングし、9番目のクロック立ち下がりエッジよりも前に SSPxADD レジスタを更新してこのビットをクリアした場合、かつユーザがその前に SSPxBUF レジスタを読み出して BF ビットをクリアしていない場合、CKP ビットはやはり Low にアサートされません。BF ビットの状態に基づくクロック ストレッチは、アドレス シーケンスではなく、データシーケンスでのみ実行されます。

### 20.4.4.3 7ビットスレーブ送信モードのクロック ストレッチ

7ビットスレーブ送信モードでは、BF ビットがクリアされている場合、9番目のクロックの立ち下がりエッジ後に CKP ビットをクリアするとクロック ストレッチが実行されます。この動作は、SEN ビットの状態に関わらず実行されます。

送信を継続するには、ユーザ ISR 内で CKP ビットをセットする必要があります。SCLx ラインを Low に保持する事で、マスタデバイスが次の送信シーケンスを開始する前に、ユーザ ISR の処理と SSPxBUF の内容読み込みの時間を確保できます ( 図 20-10 参照 )。

**Note 1:** ユーザが9番目のクロックの立ち下がりエッジよりも前に SSPxBUF の内容を読み込んで BF ビットがセットされた場合、CKP ビットはクリアされず、クロック ストレッチは実行されません。

**2:** CKP ビットは、BF ビットの状態に関わらずソフトウェアによってセットできます。

### 20.4.4.4 10ビットスレーブ送信モードのクロック ストレッチ

10ビットスレーブ送信モードでは、最初の2つのアドレス シーケンスのクロック ストレッチは、10ビットスレーブ受信モードの場合と同様に、UA ビットの状態に制御します。その後、10ビットアドレスの上位ビットと R/W ビット = 「1」を含む3番目のアドレス シーケンスが送信されます。3番目のアドレス シーケンスが終わっても UA ビットはセットされず、モジュールは送信モードに設定されます。この場合、クロック ストレッチは 7ビットスレーブ送信モードと同様に、BF フラグで制御します ( 図 20-13 参照 )。

# PIC18F97J60 ファミリ

## 20.4.4.5 クロック同期と CKP ビット

CKP ビットがクリアされると、SCLx 出力は強制的に「0」になります。しかし、CKP ビットをクリアしても、SCLx 出力が Low としてサンプルされるまで、SCLx 出力の Low はアサートされません。従って、外部 I<sup>2</sup>C マスタデバイスが既に SCLx ラインをアサートしていな

いと、CKP ビットは SCLx ラインをアサートしません。CKP ビットがセットされ、かつ I<sup>2</sup>C バス上にある他の全てのデバイスが SCLx をネゲートするまで SCLx 出力は Low のままです。このようにして、CKP ビットへの書き込みが SCLx の High 期間の最小要件に違反しないようにしています ( 図 20-14 参照 )。

図 20-14: クロックの同期タイミング

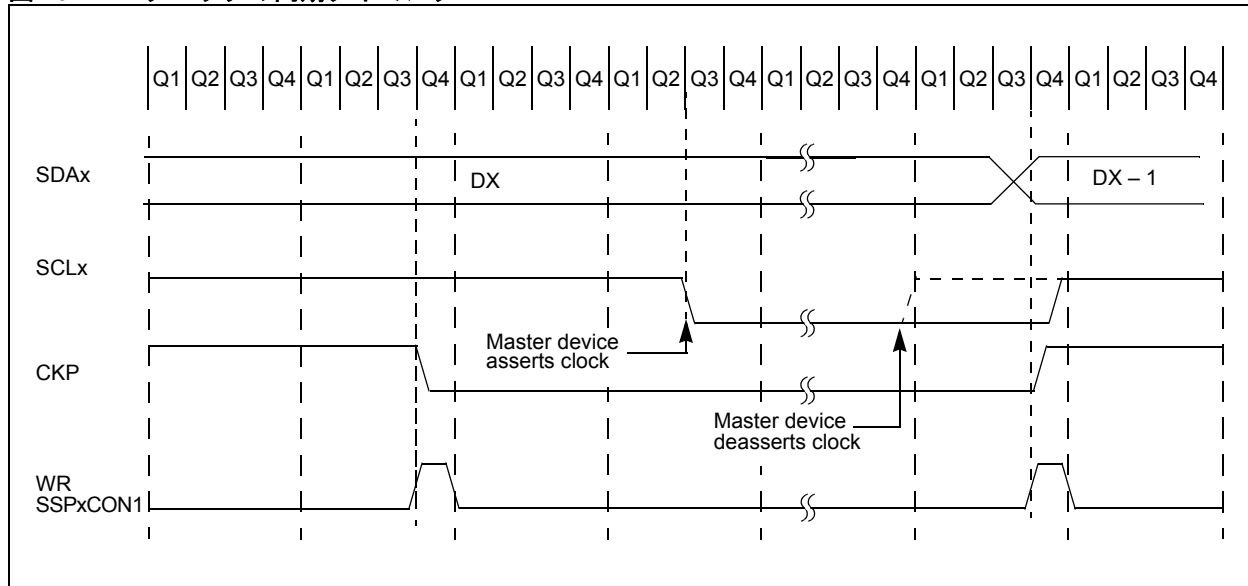
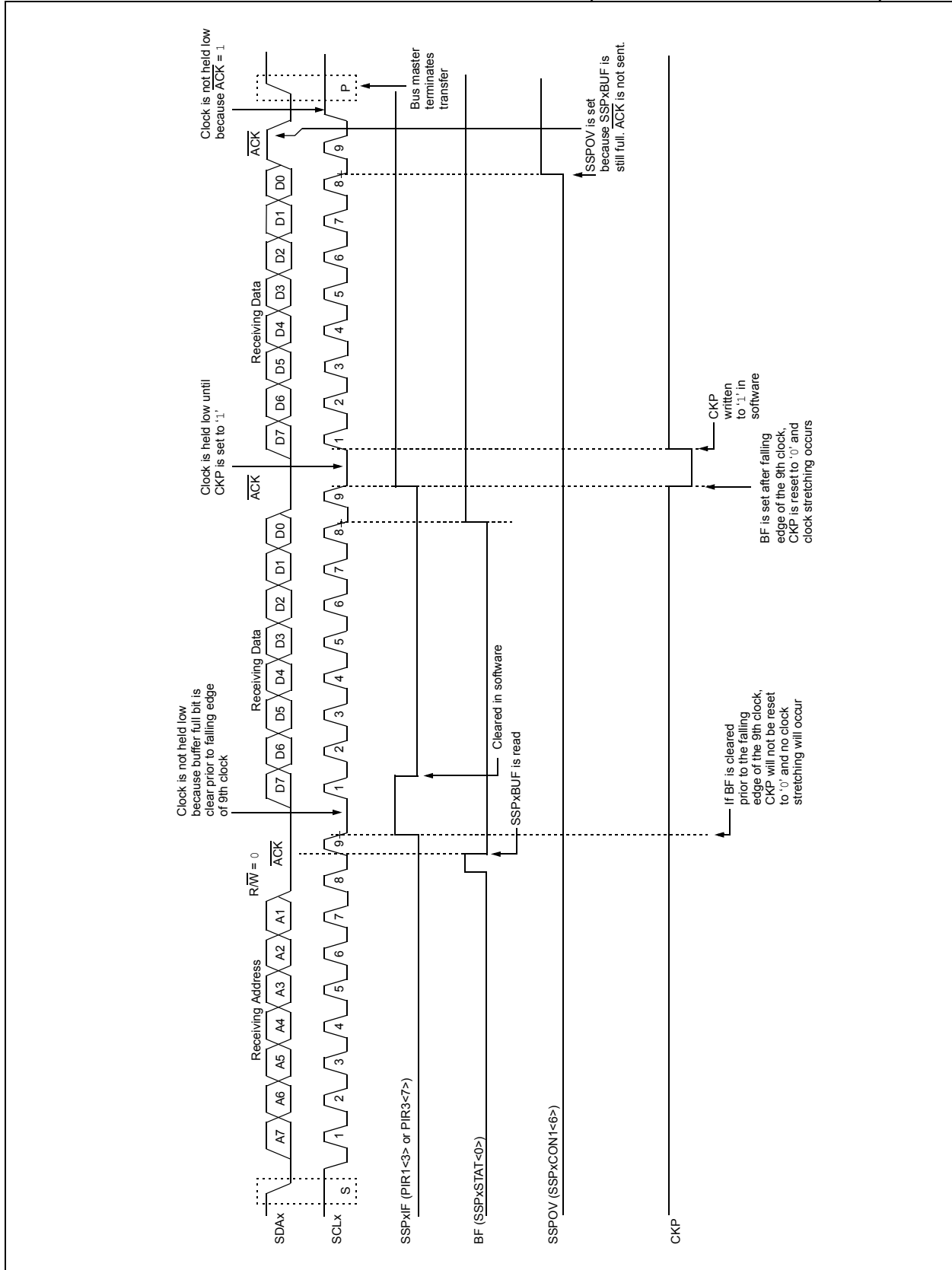
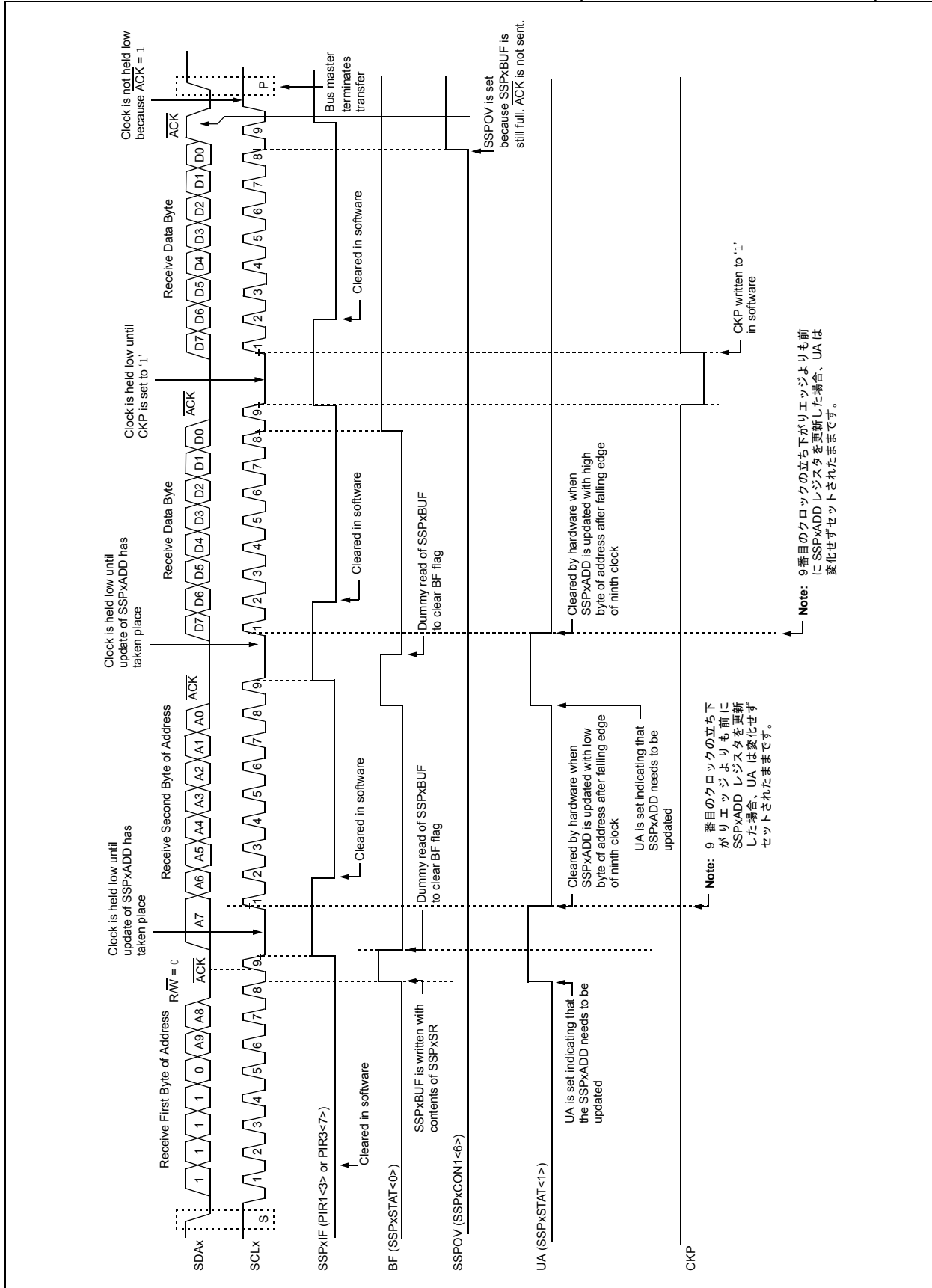


図 20-15: SEN = 1 の場合の I<sup>2</sup>C™ スレーブモードのタイミング (7 ビットアドレスモードの受信)



# PIC18F97J60 ファミリ

図 20-16: SEN = 1 の場合の I<sup>2</sup>C™ スレーブモードのタイミング (10 ビットアドレスモードの受信)





## 20.4.5 ジェネラルコールアドレスのサポート

通常の I<sup>2</sup>C バスのアドレッシング手順では、スタート条件に続く最初のバイトでマスタがスレーブデバイスのアドレスを指定します。例外として、ジェネラルコールアドレスは全てのデバイスをアドレス指定可能です。このアドレスを使うと、基本的には全てのデバイスが ACK を返します。

I<sup>2</sup>C プロトコルでは、特別な用途のために 8 つのアドレスが予約済みです。ジェネラルコールアドレスはその 1 つです。ジェネラルコールアドレスは、R/W = 0 も含め、全てのビットが「0」です。

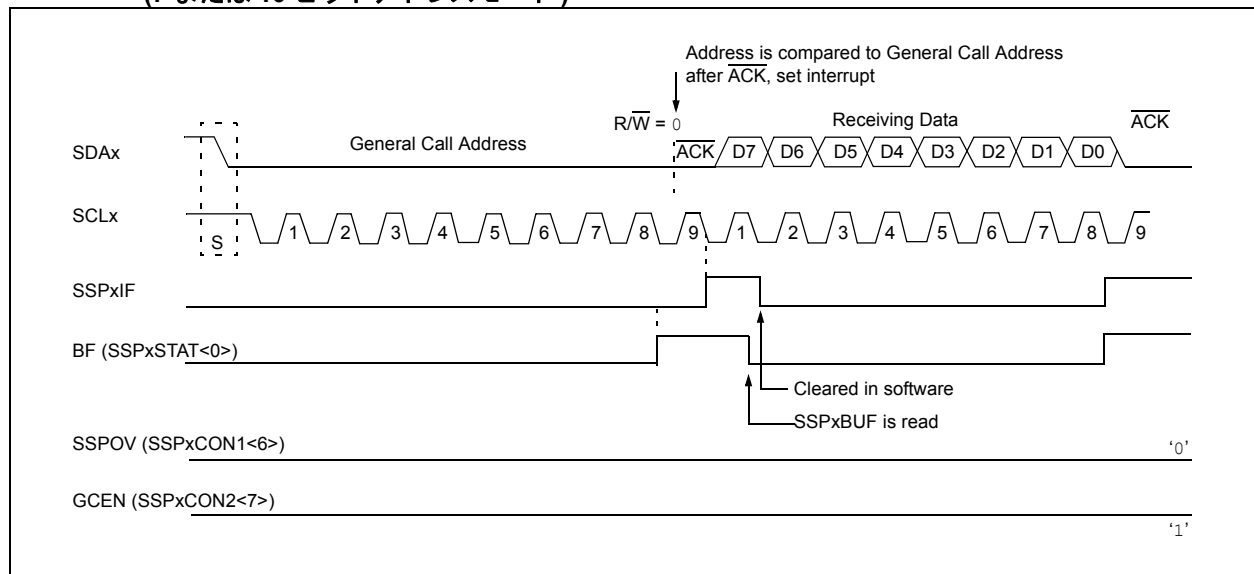
ジェネラルコールアドレスは、ジェネラルコールイネーブルビット GCEN (SSPxCON2<7>) がセットされている時に認識されます。スタートビットの検出に続き、8 ビットが SSPxSR にシフト入力され、アドレスが SSPxADD と比較されます。値はジェネラルコールアドレスとも比較され、ハードウェア内に固定されます。

ジェネラルコールアドレスに一致した場合、SSPxSR が SSPxBUF に転送され、8 番目のビットで BF フラグビットがセットされた後、9 番目のビット (ACK ビット) の立ち下がりがエッジで割り込みフラグビット SSPxIF がセットされます。

割り込みを処理する場合、割り込み要因は SSPxBUF の内容を読み出す事で確認できます。この値を使って、アドレスがデバイス固有か、ジェネラルコールアドレスのいずれであるかを判断できます。

10 ビットアドレスモードでは、アドレス後半部分を比較するために SSPxADD を更新する必要があるため、UA ビット (SSPxSTAT<1>) がセットされます。10 ビットアドレスモードのスレープで GCEN ビットをセットした場合、ジェネラルコールアドレスを検出するとアドレスの後半部分は不要になるため、UA ビットはセットされません。スレープは ACK 送信後にデータ受信を開始します (図 20-17)。

図 20-17: スレープモードのジェネラルコールアドレス シーケンス (7 または 10 ビットアドレスモード)



# PIC18F97J60 ファミリ

## 20.4.6 マスタモード

マスタモードは、SSPxCON1 の該当する SSPM ビットをセットまたはクリアし、SSPEN ビットをセットする事で有効になります。マスタモードでは、SCLxラインとSDAxラインがMSSPハードウェアによって制御されます。

マスタモードの動作は、スタート/ストップ条件の検出による割り込み生成でサポートされます。ストップ (P) ビットとスタート (S) ビットは、リセット時または MSSP モジュールが無効にされた時にクリアされます。I<sup>2</sup>C バスの制御権は、P ビットがセットされた場合、または S ビットと P ビットの両方がクリアされてバスがアイドル状態になった場合に取得できます。

ファームウェア制御マスタモードの場合、ユーザコードが全ての I<sup>2</sup>C バス動作をスタートビット条件とストップビット条件に基づいて制御します。

マスタモードを有効にした上で、ユーザは以下の 6 つの動作を選択できます。

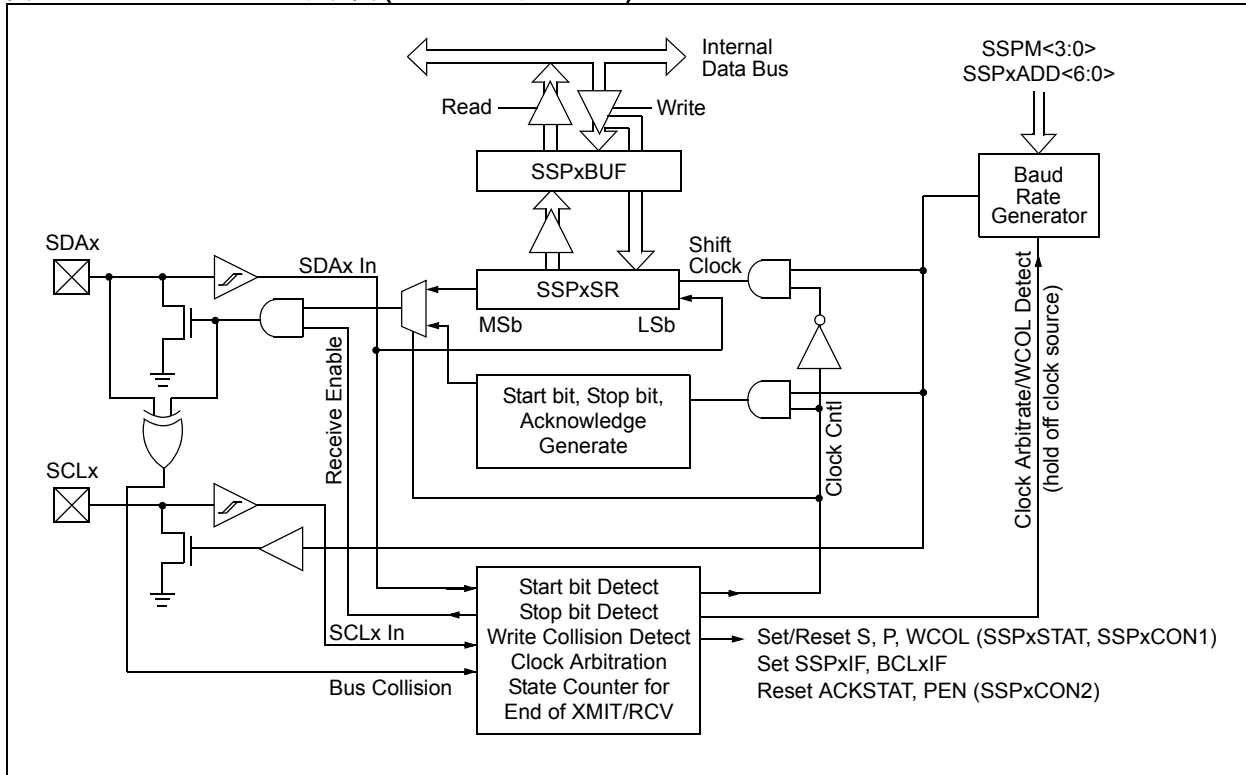
1. SDAx と SCLx でスタート条件を生成する。
2. SDAx と SCLx で反復スタート条件を生成する。
3. SSPxBUF レジスタに書き込み、データ / アドレスの送信を開始する。
4. I<sup>2</sup>C ポートをデータ受信モードに設定する。
5. 1バイトのデータ受信後に ACK 条件を生成する。
6. SDAx と SCLx でストップ条件を生成する。

**Note:** I<sup>2</sup>C マスタモードの場合、MSSP モジュールではイベントのキューイングはできません。例えばスタート条件を開始した後、スタート条件が完了する前に SSPxBUF レジスタへ書き込みを実行して、送信動作を開始する事はできません。この場合、SSPxBUF への書き込みは実行されず、SSPxBUF への書き込みが行われなかった事を示す WCOL ビットがセットされます。

以下のイベントが生じると、MSSP 割り込みフラグビット SSPxIF がセットされます (有効にしている場合は MSSP 割り込みが発生します)。

- スタート条件
- ストップ条件
- データ転送バイトの送受信
- ACK の送信
- 反復スタート

図 20-18: MSSP のブロック図 (I<sup>2</sup>C™ マスタモード)



## 20.4.6.1 I<sup>2</sup>C マスタモードの動作

マスタデバイスは、全てのシリアルクロックパルス、スタート/ストップ条件を生成します。転送は、ストップ条件または反復スタート条件で終了します。反復スタート条件は次のシリアル転送の開始でもあるため、I<sup>2</sup>C バスは解放されません。

マスタ トランスミッタ モードでは、SCLx からシリアルクロックを出力し、SDAx からシリアルデータを出力します。最初に送信するバイトは、受信デバイスのスレーブアドレス (7 ビット) と Read/Write (R/W) ビットを含みます。この場合、R/W ビットは論理「0」です。シリアルデータは 1 回に 8 ビット送信されます。1 バイト送信ごとに肯定応答 (ACK) ビットを受信します。シリアル転送の開始 / 終了は、スタート / ストップ条件の出力で示します。

マスタ受信モードの場合、最初に送信するバイトは送信デバイスのスレーブアドレス (7 ビット) と R/W ビットを含みます。この場合、R/W ビットは論理「1」です。従って、7 ビットのスレーブアドレスの末尾に受信ビットを示す「1」を付加したものが、最初に送信するバイトです。SDAx でシリアルデータを受信し、SCLx からシリアルクロックを出力します。シリアルデータは 1 回に 8 ビットずつ受信します。1 バイト受信ごとに肯定応答 (ACK) ビットを送信します。スタート / ストップ条件で、送信の開始 / 終了を示します。

SPI モードで使う baud レート ジェネレータで、100 kHz、400 kHz、1 MHz の I<sup>2</sup>C 動作に必要な SCLx クロック周波数を設定します。詳細は、[セクション 20.4.7 「baud レート」](#) を参照してください。

代表的な送信シーケンスは、以下のように進行します。

1. スタートイネーブルビット SEN (SSPxCON2<0>) をセットしてスタート条件を生成する。
2. SSPxIF がセットされる。MSSP モジュールは、新しい動作を開始する前に必要なスタート時間が経過するまで待機する。
3. ユーザが、SSPxBUF に送信先のスレーブアドレスを書き込む。
4. 8 ビット全てが転送されるまで、アドレスが SDAx ピンで順番にシフト出力される。
5. MSSP モジュールはスレーブデバイスから ACK ビットを受信し、その値を ACKSTAT ビット (SSPxCON2<6>) に書き込む。
6. MSSP モジュールは、9 番目のクロックサイクルの最後に SSPxIF ビットをセットして割り込みを生成する。
7. ユーザが、SSPxBUF に 8 ビットデータを読み込む。
8. 8 ビット全てが転送されるまで、データが SDAx ピンで順番にシフト出力される。
9. MSSP モジュールはスレーブデバイスから ACK ビットを受信し、その値を ACKSTAT ビット (SSPxCON2<6>) に書き込む。
10. MSSP モジュールは、9 番目のクロックサイクルの最後に SSPxIF ビットをセットして割り込みを生成する。
11. ストップイネーブルビット PEN (SSPxCON2<2>) をセットしてストップ条件を生成する。
12. ストップ条件が完了すると割り込みが生成される。

# PIC18F97J60 ファミリ

## 20.4.7 baud レート

I<sup>2</sup>C マスタモードでは、SSPxADD レジスタの下位 7 ビットに baud レート ジェネレータ (BRG) の再読み込み値が格納されます ( 図 20-19)。SSPxBUF への書き込みが実行されると、baud レート ジェネレータが自動的にカウントを開始します。BRG が 0 までカウントダウンすると、新しい再読み込み値が格納されるまでカウントを停止します。BRG は、1 命令サイクル (T<sub>CY</sub>) で 2 回 (Q2 クロックと Q4 クロックで) デクリメントします。I<sup>2</sup>C マスタモードでは、BRG の再読み込みは自動で行われます。

実行中の動作が完了すると ( すなわち、最後のデータビットの後に ACK が送信されると )、内部クロックが自動的にカウントを停止し、SCLx ピンは最後の状態を保持します。

表 20-3 に、命令サイクルに基づくクロックレートと SSPxADD に読み込まれる BRG の値を示します。

### 20.4.7.1 baud レートとモジュールの相互依存関係

MSSP1 と MSSP2 は独立しているため、これらを別々の baud レートの I<sup>2</sup>C マスタモードで同時に動作させる事ができます。そのためには、各モジュールに別々の BRG 再読み込み値を使う必要があります。

このモードではシステムクロックから基準クロック源を得ているため、このクロックに変更を加えてしまうと両方のモジュールに影響を与えてしまいます。BRG 再読み込み値を変更する事で、一方または両方の baud レートを前の値に戻す事ができます。

図 20-19: baud レート ジェネレータのブロック図

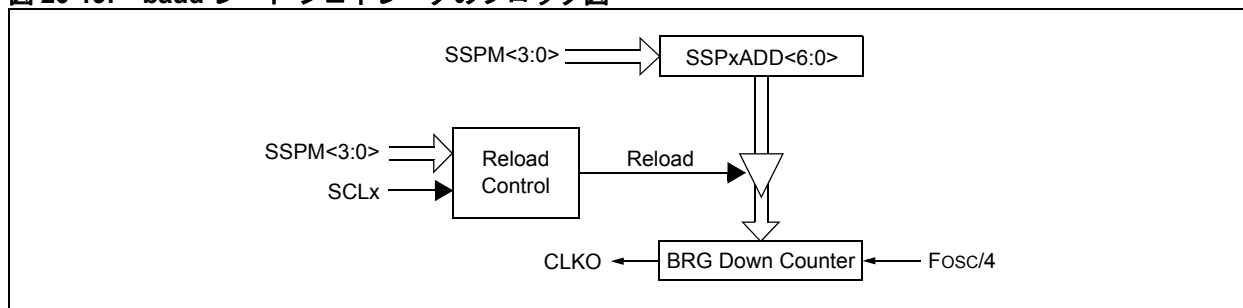


表 20-3: I<sup>2</sup>C™ のクロックレートと BRG

Fosc	BRG の値	F <sub>SCL</sub> (BRG ロールオーバー 2 回)
41.667 MHz	19h	400 kHz <sup>(1)</sup>
41.667 MHz	67h	100 kHz
31.25 MHz	13h	400 kHz <sup>(1)</sup>
31.25 MHz	4Dh	100 kHz
20.833 MHz	09h	400 kHz <sup>(1)</sup>
20.833 MHz	33h	100 kHz

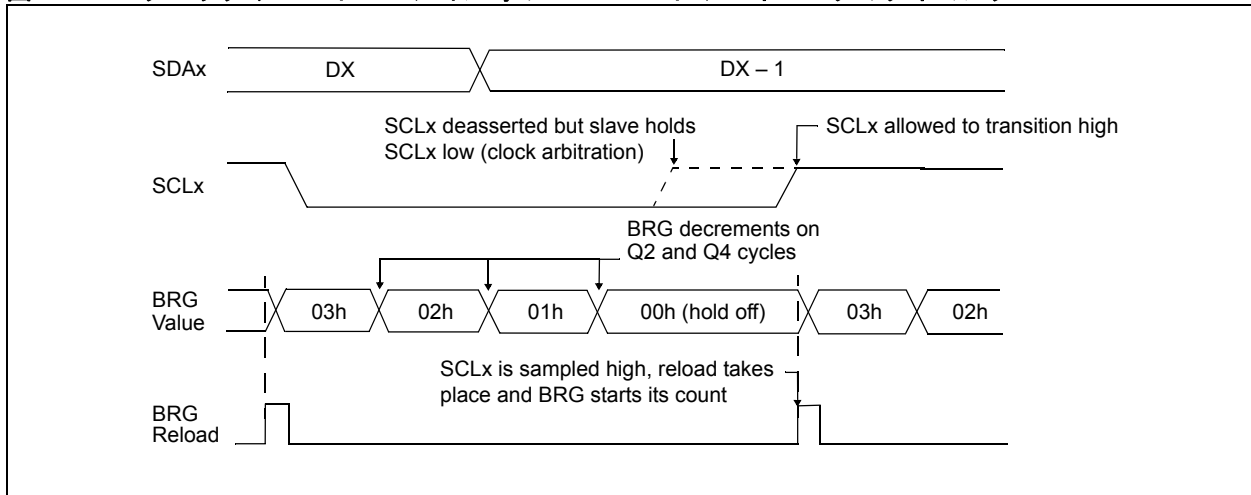
**Note 1:** この I<sup>2</sup>C™ インターフェイスは、400 kHz I<sup>2</sup>C 仕様 (100 kHz よりも高いレートに適用される仕様) に厳密には準拠していません。しかし、アプリケーションでより高速なレートが必要な場合、十分な注意を払えば使う事ができます。

## 20.4.7.2 クロック アービトレーション

送信中、受信中、反復スタート / ストップ条件の発生中にマスタが SCLx ピンをネゲートする (SCLx のフローティング High を許可する) と、クロック アービトレーションが発生します。SCLx ピンのフローティング High が許可されても、実際に High がサンプルされるまで baud レート ジェネレータ (BRG) のカウントは中断されます。SCLx ピンで High がサンプルされる

と、baud レート ジェネレータに SSPxADD<6:0> の内容が再度読み込まれ、カウントが始まります。外部デバイスによってクロックが Low に保持されていたとしても、上記の動作によって、SCLx の High 期間が最短でも BRG のロールオーバー 1 回分確保されます (図 20-20)。

図 20-20: クロック アービトレーション時の baud レート ジェネレータのタイミング



# PIC18F97J60 ファミリ

## 20.4.8 I<sup>2</sup>C マスタモードのスタート条件のタイミング

スタート条件を開始するには、スタート イネーブルビット SEN (SSPxCON2<0>) をセットします。SDAx ピンと SCLx ピンで High がサンプルされると、baud レート ジェネレータに SSPxADD<6:0> の内容が再度読み込まれ、カウントが始まります。baud レート ジェネレータがタイムアウト (TBRG) した時に SCLx ピンと SDAx ピンが両方共 High としてサンプルされると、SDAx ピンは Low に駆動されます。SCLx ピンが High の時に SDAx が Low に駆動されるとスタート条件となり、S ビット (SSPxSTAT<3>) がセットされます。続いて、baud レート ジェネレータに SSPxADD<6:0> の内容が再度読み込まれ、カウントが再開します。baud レート ジェネレータがタイムアウト (TBRG) すると、SEN ビット (SSPxCON2<0>) はハードウェアによって自動的にクリアされます。baud レート ジェネレータは動作を中断し、SDAx ラインが Low に保持されたままスタート条件が完了します。

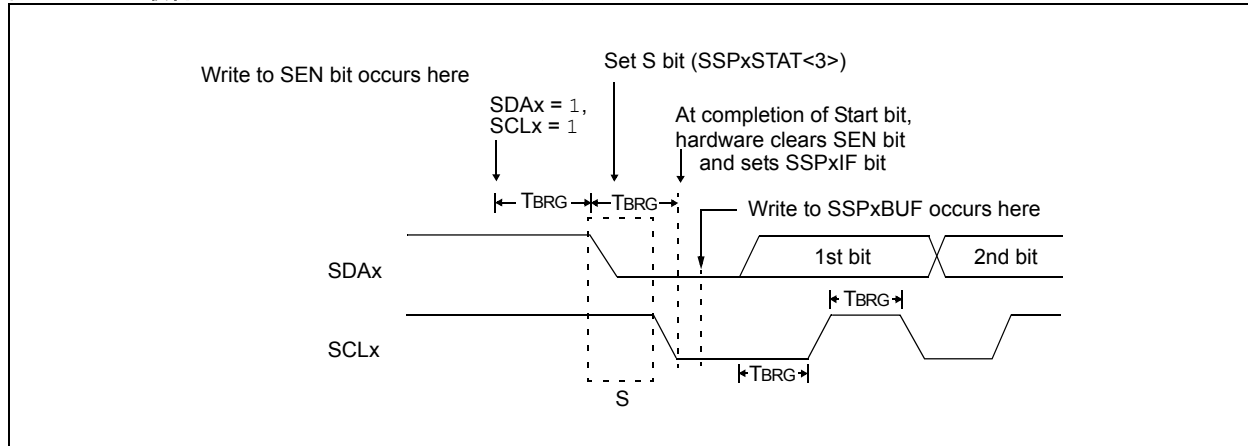
**Note:** スタート条件の開始時に SDAx ピンと SCLx ピンが既に Low としてサンプルされている場合、またはスタート条件中に SDAx ラインが Low に駆動される前に SCLx ラインが Low としてサンプルされた場合、バスコリジョンが発生します。バスコリジョン割り込みフラグ BCLxIF がセットされ、スタート条件は中止されて I<sup>2</sup>C モジュールはアイドル状態にリセットされます。

### 20.4.8.1 WCOL ステータスフラグ

スタートシーケンス中にユーザが SSPxBUF に書き込みを実行すると WCOL がセットされ、バッファの内容は変化しません (書き込みは実行されません)。

**Note:** イベントのキューイングはできないため、スタート条件が終了するまで SSPxCON2 の下位 5 ビットには書き込む事ができません。

図 20-21: 最初のスタートビットのタイミング



## 20.4.9 I<sup>2</sup>C マスタモードの反復スタート条件のタイミング

反復スタート条件は、RSEN ビット (SSPxCON2<1>) が High にプログラムされており、I<sup>2</sup>C ロジック モジュールがアイドル状態の場合に発生します。RSEN ビットがセットされると、SCLx ピンが Low にアサートされます。SCLx ピンで Low がサンプルされると、baud レート ジェネレータに SSPxADD<6:0> の内容が読み込まれ、カウントが始まります。SDAx ピンは、baud レート ジェネレータが 1 回カウントする間 (TBRG)、High に解放されます。baud レート ジェネレータがタイムアウトした時に SDAx で High がサンプルされると、SCLx ピンがネゲートされ High になります。SCLx ピンで High がサンプルされると、baud レート ジェネレータに SSPxADD<6:0> の内容が再度読み込まれ、カウントが始まります。SDAx と SCLx は 1 TBRG の期間 High を維持する必要があります。この後、SCLx が High の間に SDAx ピンを 1 TBRG の期間アサート (SDAx = 0) します。その後、RSEN ビット (SSPxCON2<1>) が自動的にクリアされ、baud レート ジェネレータには値が再読み込みされず、SDAx ピンは Low に保持されます。SDAx ピンと SCLx ピンでスタート条件が検出されると、S ビット (SSPxSTAT<3>) がただちにセットされます。SSPxIF ビットは、baud レート ジェネレータがタイムアウトするまでセットされません。

**Note 1:** その他のいかなるイベントが進行している間に RSEN がセットされても、実行されません。

**2:** 以下の場合、反復スタート条件中にバスコリジョンが発生します。

- SCLx が Low から High に遷移した時に SDAx が Low としてサンプルされる。
- SDAx が Low にアサートされる前に SCLx が Low に遷移した (別のマスタがデータ「1」を送信しようとしている可能性があります)。

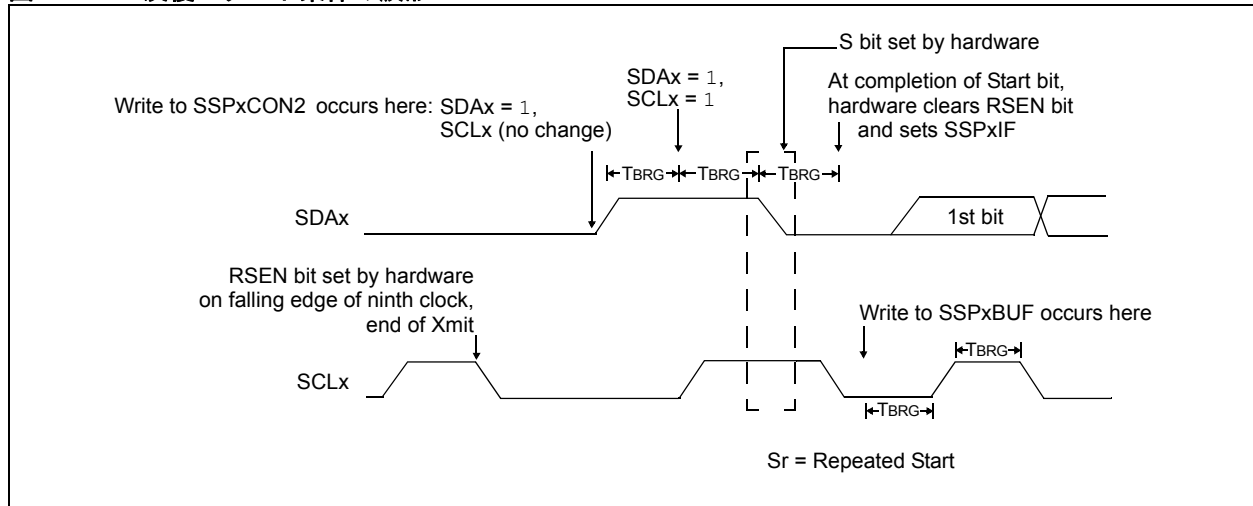
SSPxIF ビットがセットされた直後に、ユーザは SSPxBUF に 7 ビットのアドレス (7 ビットモード時) または既定値の最初のアドレスバイト (10 ビットモード時) を書き込むことができます。最初の 8 ビットを送信して ACK を受信したら、ユーザはアドレスの残り 8 ビット (10 ビットモード) または 8 ビットのデータ (7 ビットモード) を送信できます。

### 20.4.9.1 WCOL ステータスフラグ

反復スタート シーケンス中にユーザが SSPxBUF に書き込むと WCOL がセットされ、バッファの内容は変化しません (書き込みは実行されません)。

**Note:** イベントのキューイングはできないため、反復スタート条件が終了するまで SSPxCON2 の下位 5 ビットには書き込む事ができません。

図 20-22: 反復スタート条件の波形



# PIC18F97J60 ファミリ

## 20.4.10 I<sup>2</sup>C マスタモード送信

データバイト、7 ビットアドレス、10 ビットアドレスの残り半分の送信は、SSPxBUF レジスタへ値を書き込むだけで完了します。書き込み動作によってバッファフルフラグビット BF がセットされ、baud レートジェネレータのカウントと共に次の送信が開始します。SCLx の立ち下がリエッジのアサート後、アドレス / データの各ビットは SDAx ピンにシフト出力されます (データホールド時間仕様のパラメータ 106 参照)。SCLx は baud レートジェネレータのロールオーバーカウント 1 回分 (TBRG) の間、Low に保持されます。データは SCLx が High に解放されるまでは有効です (データセットアップ時間仕様のパラメータ 107 参照)。SCLx ピンは High に解放された後、TBRG の期間、その状態を保持します。SDAx ピン上のデータは、この期間と、SCLx の次の立ち下がリエッジ後のホールド時間まで安定する必要があります。8 番目のビットがシフト出力された (8 番目のクロックの立ち下がリエッジ) 後、BF フラグがクリアされてマスタが SDAx を解放します。アドレスが一致した場合、またはデータを正常に受信した場合、アドレス指定されたスレーブデバイスは 9 番目のビットと同時に ACK ビットを送信します。ACK の状態は、9 番目のクロックの立ち下がリエッジで ACKDT ビットに書き込まれます。マスタが ACK を受信すると、ACK ステータスビット ACKSTAT がクリアされます (ACK を受信していない場合、ビットはセット)。9 番目のクロックの後、SSPxIF ビットがセットされ、マスタクロック (baud レートジェネレータ) は次のデータバイトが SSPxBUF に読み込まれるまで一時停止し、SCLx は Low、SDAx はそのままの状態を保持します (図 20-23)。

SSPxBUF への書き込み後、7 ビットのアドレスと R/W ビットが全て出力されるまで SCLx の立ち下がリエッジでアドレスのビットが1つずつシフト出力されます。8 番目のクロックの立ち下がリエッジでマスタは SDAx ピンをネゲートし、スレーブが ACK を返せるようにします。9 番目のクロックの立ち下がリエッジでマスタは SDAx ピンをサンプルし、スレーブがアドレスを認識したかどうかを確認します。ACK ビットの状態が ACKSTAT ステータスビット (SSPxCON2<6>) に読み込まれます。アドレス送信の 9 番目のクロックが立ち下がると SSPxIF がセットされ、BF フラグがクリアされます。これによって次に SSPxBUF への書き込みが実行されるまで baud レートジェネレータが OFF になり、SCLx が Low に保持され、SDAx はフローティング状態に移行します。

### 20.4.10.1 BF ステータスフラグ

送信モードの場合、BF ビット (SSPxSTAT<0>) は CPU が SSPxBUF に書き込むとセットされ、8 ビットが全てシフト出力された時点でクリアされます。

### 20.4.10.2 WCOL ステータスフラグ

送信中に (SSPxSR がデータバイトをシフト出力中に) ユーザが SSPxBUF に書き込むと、SSPxBUF への書き込みから 2 Tcy 後に WCOL がセットされ、バッファの内容は変化しません (書き込みは実行されません)。2 Tcy 以内に SSPxBUF に再度書き込みを実行すると、

WCOL ビットがセットされたまま SSPxBUF が更新されます。結果として不正なデータが転送される可能性があります。

不正なデータの転送を回避するため、SSPxBUF への書き込みの後は毎回 WCOL ビットがクリアされている事を確認する必要があります。いずれの場合も、WCOL はソフトウェアでクリアする必要があります。

### 20.4.10.3 ACKSTAT ステータスフラグ

送信モードでは、スレーブが ACK を送信 ( $\overline{ACK} = 0$ ) すると ACKSTAT ビット (SSPxCON2<6>) がクリアされ、スレーブが ACK を送信しない ( $\overline{ACK} = 1$ ) とセットされます。スレーブは、自分自身のアドレスを認識するか (ジェネラルコールを含む)、正常にデータ受信を完了すると ACK を送信します。

## 20.4.11 I<sup>2</sup>C マスタモード受信

マスタモード受信は、受信イネーブルビット RCEN (SSPxCON2<3>) をセットすると有効になります。

**Note:** MSSP モジュールがアイドル状態の時に RCEN ビットをセットしないと、RCEN ビットは無視されます。

baud レートジェネレータがカウントを開始し、ロールオーバーごとに SCLx ピンが遷移 (High → Low/Low → High) してデータが SSPxSR にシフト入力されます。8 番目のクロックの立ち下がリエッジの後、受信イネーブルフラグが自動的にクリアされます。これによって SSPxSR の内容が SSPxBUF に読み込まれ、BF フラグビットと SSPxIF フラグビットがセットされる事で、baud レートジェネレータがカウントを停止して SCLx が Low に保持されます。MSSP はアイドル状態であり、次のコマンドを待ちます。CPU がバッファを読み出すと、BF フラグビットは自動的にクリアされます。ユーザは、受信の最後に ACK シーケンス イネーブルビット ACKEN (SSPxCON2<4>) をセットして ACK ビットを送信できます。

### 20.4.11.1 BF ステータスフラグ

受信動作の場合、SSPxSR から SSPxBUF へアドレスバイトまたはデータバイトが読み込まれると、BF ビットがセットされます。このビットは SSPxBUF レジスタを読み出すとクリアされます。

### 20.4.11.2 SSPOV ステータスフラグ

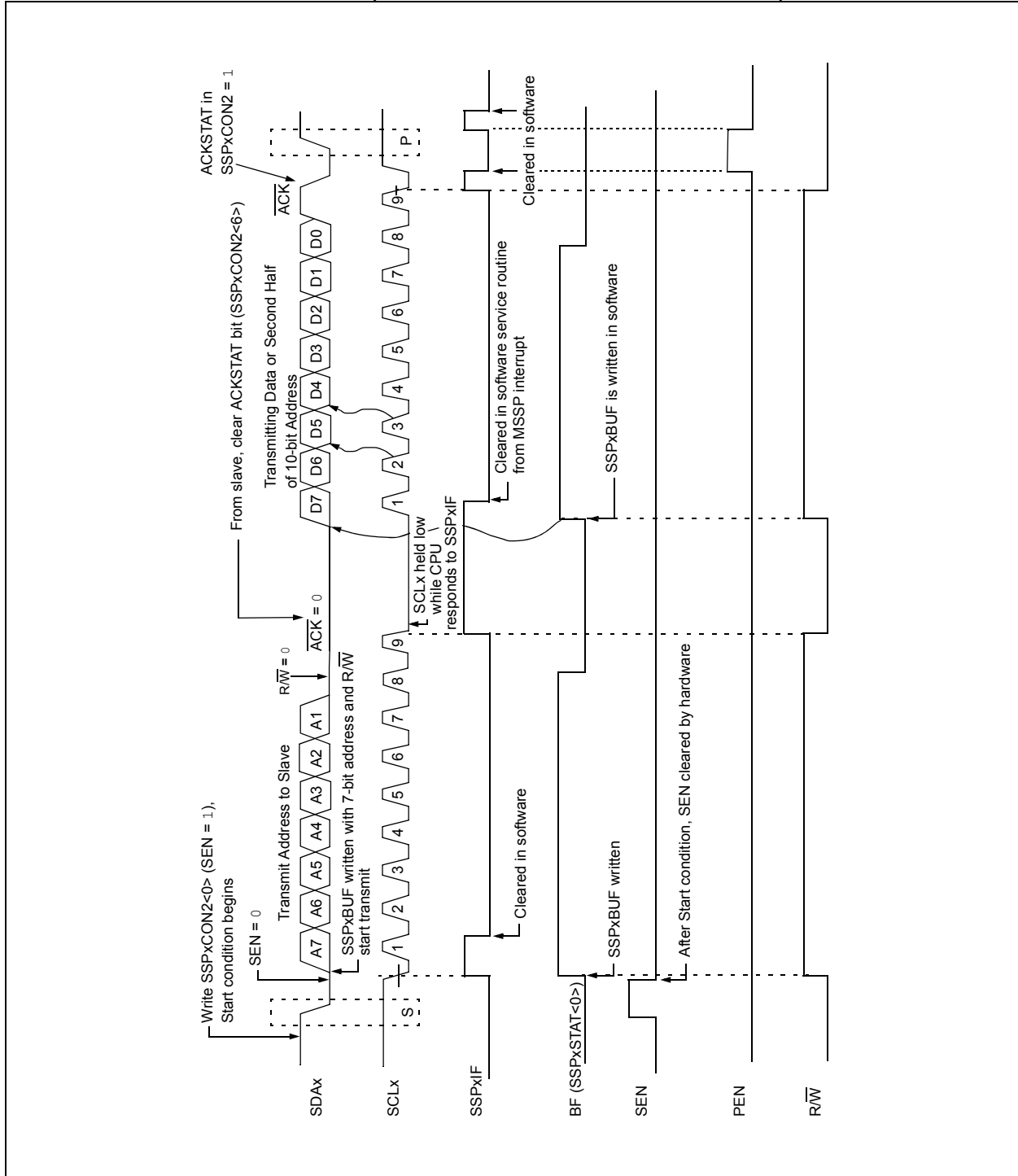
受信動作の場合、以前の受信で BF フラグが既にセットされている時に SSPxSR が 8 ビットを受信すると、SSPOV ビットがセットされます。

### 20.4.11.3 WCOL ステータスフラグ

受信が既に開始した状態で (すなわち、SSPxSR がデータバイトをシフト入力している間に) ユーザが SSPxBUF に書き込むと WCOL がセットされ、バッファの内容は変化しません (書き込みは実行されません)。



図 20-23: I<sup>2</sup>C™ マスタモードの波形 (7 または 10 ビットアドレスモードの送信)





## 20.4.12 ACK シーケンスのタイミング

ACK シーケンスは、ACK シーケンス イネーブルビット ACKEN (SSPxCON2<4>) をセットすると有効になります。このビットをセットすると SCLx ピンが Low に駆動され、ACK データビットの内容が SDAx ピンに出力されます。ACK を生成したい場合、ACKDT ビットをクリアします。ACK を生成しない場合、ACK シーケンスが開始する前に ACKDT ビットをセットする必要があります。baud レート ジェネレータが 1 ロールオーバー期間 (TBRG) カウントし、SCLx ピンがネゲートされます (High に遷移)。SCLx ピンが High としてサンプルされた場合 (クロックアービトレーション)、baud レート ジェネレータは TBRG の期間カウントします。その後 SCLx ピンが Low に駆動されます。続いて ACKEN ビットが自動的にクリアされて baud レート ジェネレータが OFF になり、MSSP モジュールはアイドルに移行します (図 20-25)。

### 20.4.12.1 WCOL ステータスフラグ

ACK シーケンス中にユーザが SSPxBUF に書き込むと WCOL がセットされ、バッファの内容は変化しません (書き込みは実行されません)。

## 20.4.13 ストップ条件のタイミング

ストップ シーケンス イネーブルビット PEN (SSPxCON2<2>) をセットすると、送受信終了時に SDAx ピンでストップビットがアサートされます。送受信が完了する 9 番目のクロックの立ち上がりエッジ後、SCLx ラインは Low に保持されます。PEN ビットをセットすると、マスタは SDAx ラインを Low にアサートします。SDAx ラインが Low としてサンプルされると、baud レート ジェネレータが再読み込みされ「0」までカウントダウンします。baud レート ジェネレータがタイムアウトすると、SCLx ピンが High に駆動され、1 TBRG (BRG のロールオーバー カウント) 後に SDAx ピンがネゲートされます。SCLx が High の間に SDAx ピンが High としてサンプルされると、P ビット (SSPxSTAT<4>) がセットされます。1 TBRG 経過後、PEN ビットがクリアされ、SSPxIF ビットがセットされます (図 20-26)。

### 20.4.13.1 WCOL ステータスフラグ

ストップ シーケンス中にユーザが SSPxBUF に書き込みを実行すると WCOL がセットされ、バッファの内容は変化しません (書き込みは実行されません)。

図 20-25: ACK シーケンスの波形

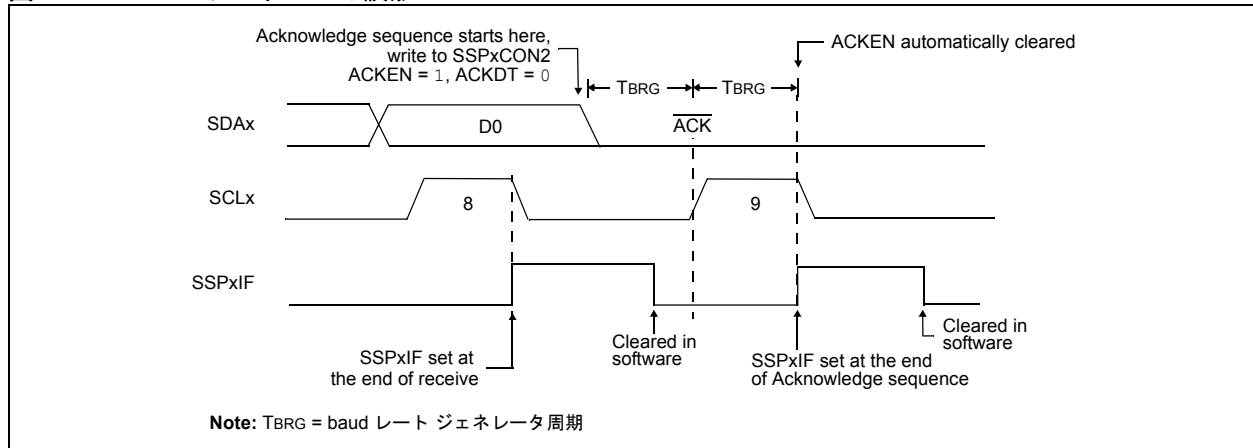
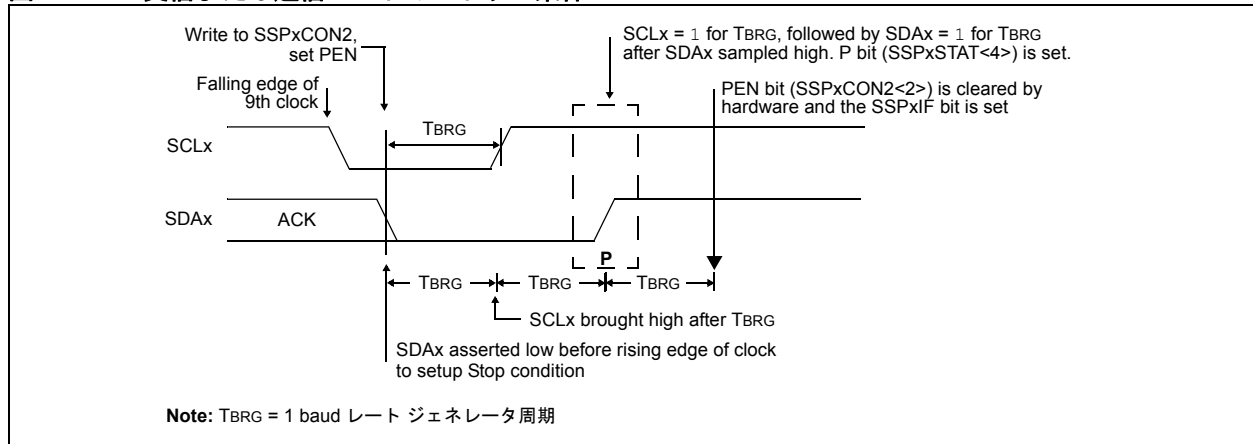


図 20-26: 受信または送信モードのストップ条件



# PIC18F97J60 ファミリ

## 20.4.14 スリープ動作

I<sup>2</sup>C モジュールはスリープ時でもアドレス/データを受信でき、アドレスが一致した場合または 1 バイトの転送が完了した場合にプロセッサをスリープから復帰させる事ができます (MSSP 割り込みを有効にした場合)。

## 20.4.15 リセットの影響

リセットは MSSP モジュールを無効にし、現在の転送は中止されます。

## 20.4.16 マルチマスタ モード

マルチマスタ モードでは、スタート/ストップ条件の検出時に生成される割り込みを使ってバスが空いているかどうかを判断できます。ストップ (P) ビットとスタート (S) ビットは、リセット時または MSSP モジュールが無効にされた時にクリアされます。I<sup>2</sup>C バスの制御権は、P ビット (SSPxSTAT<4>) がセットされた場合、または S ビットと P ビットの両方がクリアされてバスがアイドル状態になった場合に取得できます。バスがビジーの場合、MSSP 割り込みを有効にすると、ストップ条件が発生した時に割り込みが生成されます。

マルチマスタ動作ではバス アービトレーションの必要があるため、SDAx ラインを監視して実際の信号レベルが予測する出力レベルと一致しているかどうかを確認する必要があります。このチェックはハードウェアで行い、結果は BCLxIF ビットに格納されます。

下記のステートではバス アービトレーションに敗れる可能性があります。

- アドレス転送
- データ転送
- スタート条件
- 反復スタート条件
- ACK 条件

## 20.4.17 マルチマスタ通信、バスコリジョン、バス アービトレーション

マルチマスタ モードは、バス アービトレーションによってサポートされています。マスタがアドレス/データビットを SDAx ピンに出力する際、マスタが出力する値が「1」ならアービトレーションによって SDAx がフローティング High となり、別のマスタは「0」をアサートします。SCLx ピンがフローティング High の間、データは安定している必要があります。SDAx で予測されるデータ値が「1」であるのに、サンプルされた SDAx ピンの値が「0」の場合、バスコリジョンが発生しています。マスタはバスコリジョン割り込みフラグ BCLxIF をセットし、I<sup>2</sup>C ポートをアイドル状態にリセットします (図 20-27)。

送信中にバスコリジョンが発生した場合、送信が停止され、BF フラグがクリアされ、SDAx ラインと SCLx ラインがネゲートされ、SSPxBUF への書き込みが可能になります。ユーザがバスコリジョン割り込みサービスルーチンを使う場合、I<sup>2</sup>C バスが空いていれば、スタート条件をアサートする事で通信を再開できます。

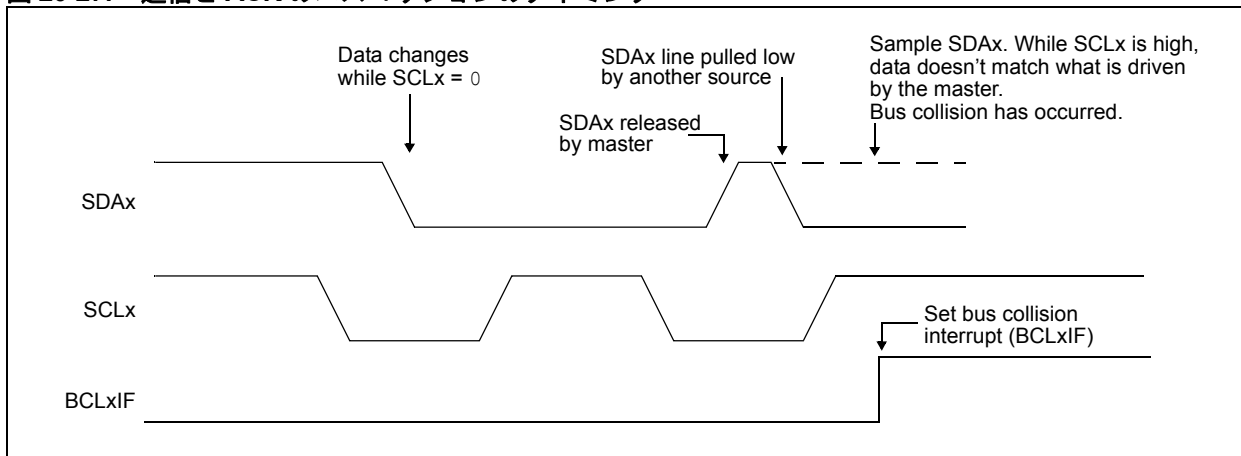
スタート、反復スタート、ストップ、ACK 条件中にバスコリジョンが発生した場合、これらの条件は中止され、SDAx ラインと SCLx ラインがネゲートされ、SSPxCON2 レジスタの対応する制御ビットがクリアされます。ユーザがバスコリジョン割り込みサービスルーチンを使う場合、I<sup>2</sup>C バスが空いていれば、スタート条件をアサートする事で通信を再開できます。

マスタは SDAx ピンと SCLx ピンの監視を続けます。ストップ条件が発生すると、SSPxIF ビットがセットされます。

SSPxBUF に書き込むと、バスコリジョン発生時にトランスミッタがどこまで送信を完了していたかに関わらず、データの先頭ビットから送信が開始します。

マルチマスタ モードでは、スタート/ストップ条件の検出時に生成される割り込みを使ってバスが空いているかどうかを判断できます。I<sup>2</sup>C バスの制御権は、SSPxSTAT レジスタの P ビットがセットされた場合、または S ビットと P ビットの両方がクリアされてバスがアイドル状態になった場合に取得できます。

図 20-27: 送信と ACK のバスコリジョンのタイミング



## 20.4.17.1 スタート条件中のバスコリジョン

スタート条件中は、以下の場合にバスコリジョンが発生します。

- スタート条件の開始時に SDAx または SCLx で Low がサンプルされる (図 20-28)
- SDAx が Low にアサートされる前に SCLx で Low がサンプルされる (図 20-29)

スタート条件中は、SDAx ピンと SCLx ピンの両方を監視します。

SDAx ピンが既に Low の場合、または SCLx ピンが既に Low の場合、以下の全てが実行されます。

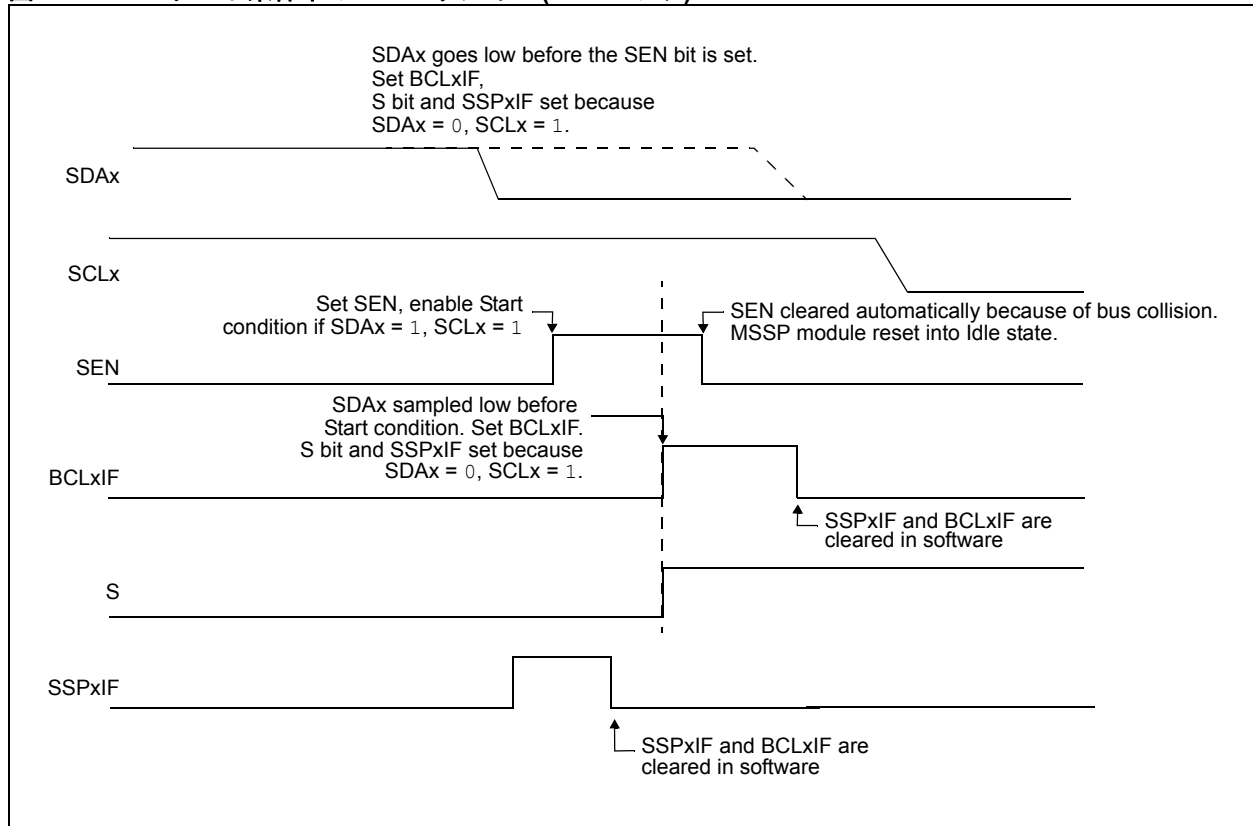
- スタート条件を中止する
- BCLxIF フラグをセットする
- MSSP モジュールをアイドル状態にリセットする (図 20-28)

スタート条件は、SDAx ピンと SCLx ピンのネゲートから開始します。SDAx ピンが High としてサンプルされると、baud レート ジェネレータに SSPxADD<6:0> の内容が読み込まれ、0 になるまでカウントダウンします。SDAx が High の時に SCLx ピンが Low としてサンプルされると、スタート条件中に別のマスタがデータ「1」を駆動しているものと判断し、バスコリジョンが発生します。

上記の BRG カウント中に SDAx ピンで Low がサンプルされると BRG はリセットされ、SDAx ラインは BRG のタイムアウトを待たずにアサートされます (図 20-30)。しかし、SDAx ピンで「1」がサンプルされた場合、SDAx ピンは BRG カウントの終了時に Low にアサートされます。次に baud レート ジェネレータに値が再度読み込まれて 0 までカウントダウンします。この間に SCLx ピンで「0」がサンプルされてもバスコリジョンは発生しません。BRG のカウント終了時、SCLx ピンが Low にアサートされます。

**Note:** スタート条件中にバスコリジョンが問題にならないのは、2つのバスマスタが全く同じタイミングでスタート条件をアサートする事があり得ないためです。従って、必ずどちらか一方のマスタが他方よりも先に SDAx をアサートします。そのような状態ではバスコリジョンは発生しません。2つのマスタはスタート条件に続く最初のアドレスでバス アービトレーションを行うためです。アドレスが同じだった場合、続くデータ部分、反復スタート、ストップ条件でアービトレーションが必要です。

図 20-28: スタート条件中のバスコリジョン (SDAx のみ)



# PIC18F97J60 ファミリ

図 20-29: スタート条件中のバスコリジョン (SCLx = 0)

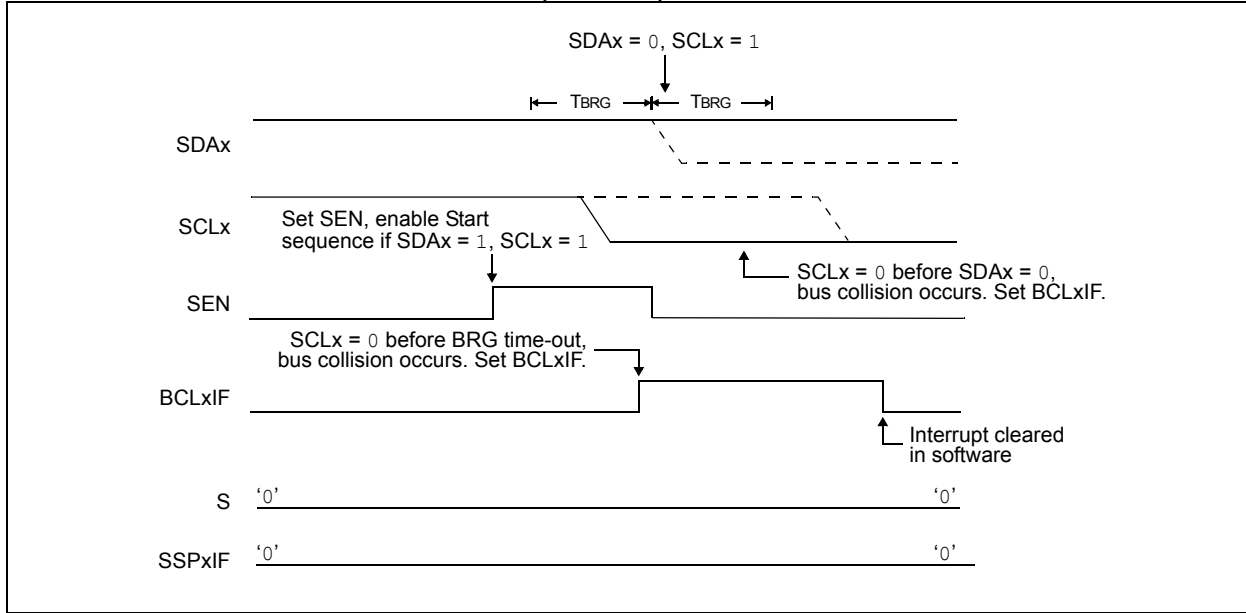
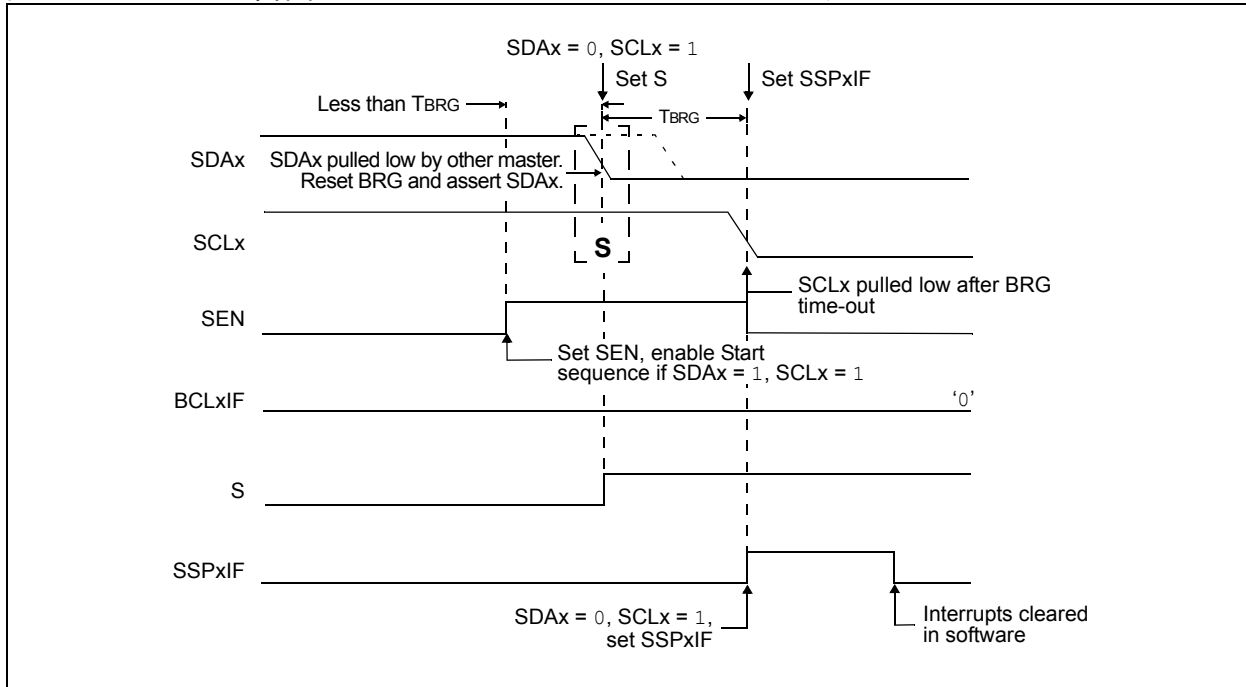


図 20-30: スタート条件中の SDAx アービトレーションによる BRG リセット



20.4.17.2 反復スタート条件中のバスコリジョン  
反復スタート条件中は、以下の場合にバスコリジョンが発生します。

- SCLx が Low から High に遷移する時に SDAx で Low がサンプルされる
- SDAx で Low がアサートされる前に SCLx が Low に遷移する (別のマスタがデータ「1」を送信しようとしている事を示す)

SDAx がユーザによってネゲートされ、ピンが High にフローティングすると、BRG に SSPxADD<6:0> の内容が読み込まれ、0 になるまでカウントダウンします。続いて SCLx ピンがネゲートされ、High がサンプルされると、SDAx ピンがサンプルされます。

この時 SDAx が Low なら、バスコリジョンが発生しています (すなわち、別のマスタがデータ「0」を送信しようとしています。図 20-31 参照)。SDAx で High が

サンプルされた場合、BRG に値が再度読み込まれてカウントが始まります。BRG がタイムアウトする前に SDAx が High から Low に遷移した場合、バスコリジョンは発生しません。2 つのマスタが全く同じタイミングで SDAx をアサートする事があり得ないためです。

BRG がタイムアウトする前に SCLx が High から Low に遷移し、SDAx がまだアサートされていない場合はバスコリジョンが発生します。この場合、反復スタート条件中に別のマスタがデータ「1」を送信しようとしています (図 20-32 参照)。

BRG タイムアウト時に SCLx と SDAx の両方が High のままの場合、SDAx ピンが Low に駆動され、BRG の再読み込み後にカウントを開始します。カウントが終了すると、SCLx ピンはその状態に関係なく Low に駆動され、反復スタート条件が終了します。

図 20-31: 反復スタート条件中のバスコリジョン (ケース 1)

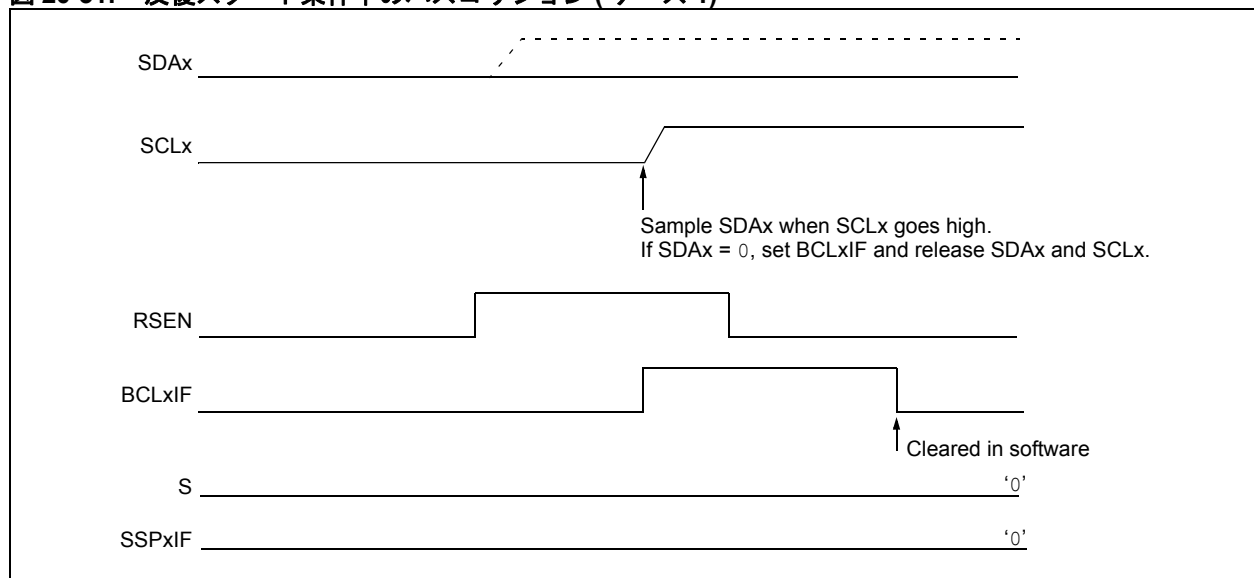
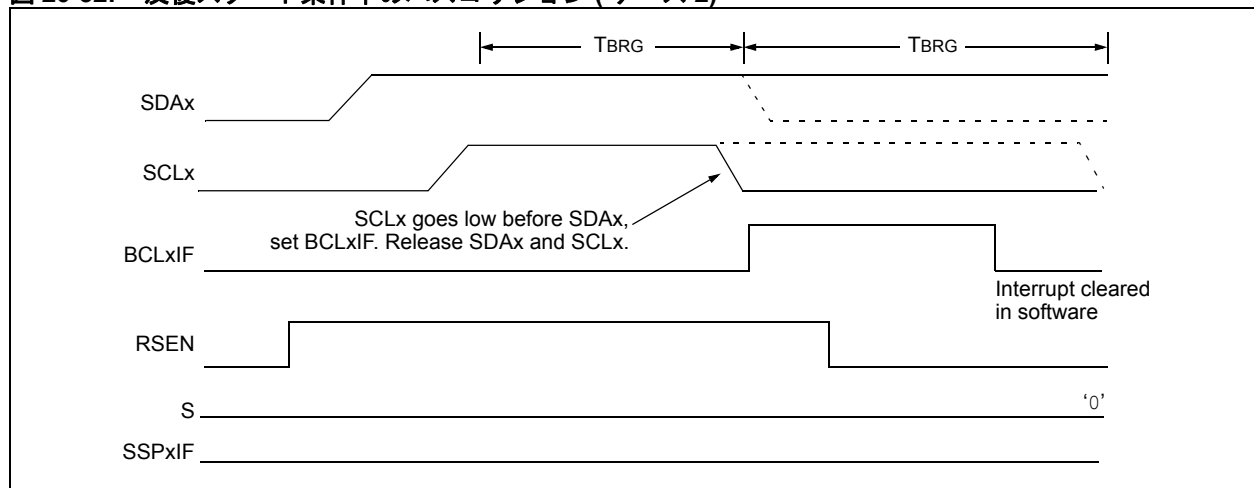


図 20-32: 反復スタート条件中のバスコリジョン (ケース 2)



# PIC18F97J60 ファミリ

## 20.4.17.3 ストップ条件中のバスコリジョン

ストップ条件中は、以下の場合にバスコリジョンが発生します。

- a) SDAx ピンがネゲートされて High フローティングが可能になった後、BRG のタイムアウト後に SDAx で Low がサンプルされる
- b) SCLx ピンがネゲートされた後、SDAx が High に遷移する前に SCLx で Low がサンプルされる

ストップ条件は、SDAx の Low アサートから開始します。SDAx で Low がサンプルされると、SCLx ピンをフローティングにする事ができます。SCLx ピンで

High がサンプルされると(クロックアービトレーション)、baud レートジェネレータに SSPxADD<6:0> の値が読み込まれ、0 になるまでカウントダウンします。BRG のタイムアウト後、SDAx がサンプルされます。SDAx で Low がサンプルされた場合、バスコリジョンが発生しています。これは、別のマスタがデータ「0」を駆動しようとしているためです(図 20-33)。SDAx のフローティング High が許可される前に SCLx ピンで Low がサンプルされると、バスコリジョンが発生します。この場合も、別のマスタがデータ「0」を駆動しようとしています(図 20-34)。

図 20-33: ストップ条件中のバスコリジョン(ケース 1)

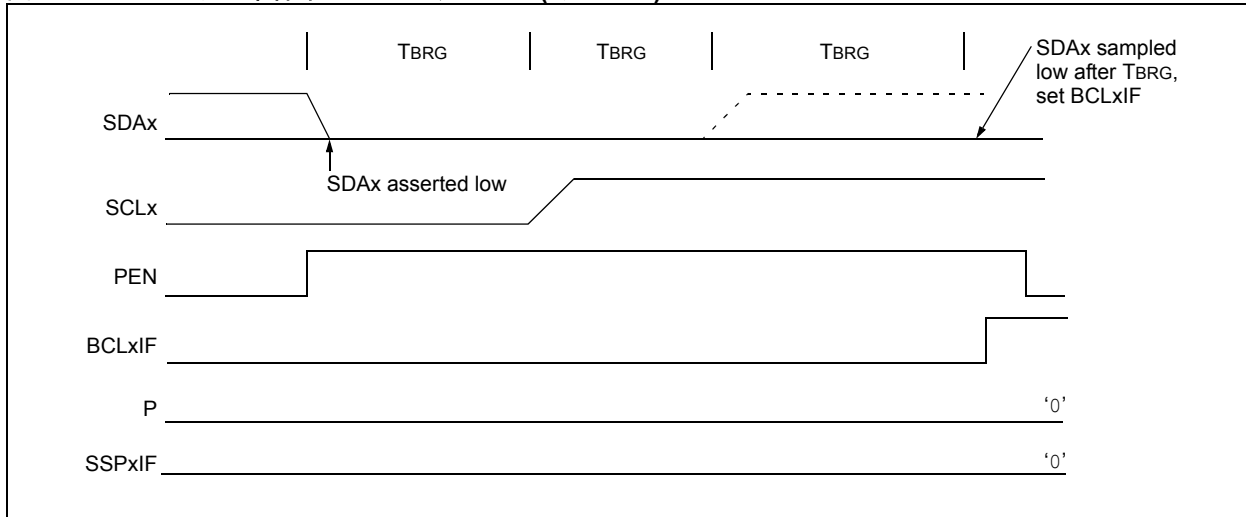
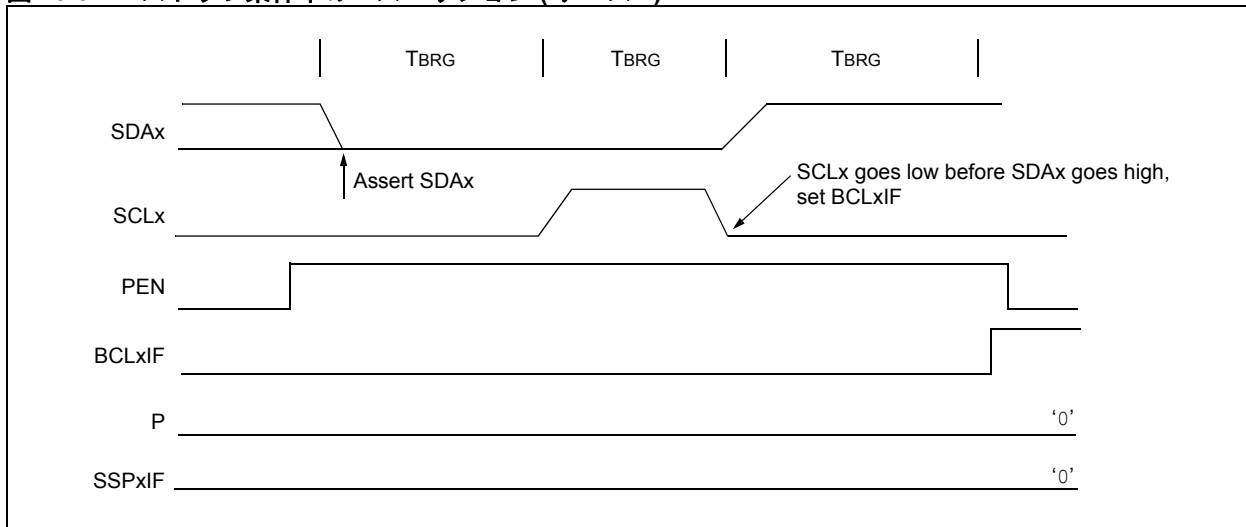


図 20-34: ストップ条件中のバスコリジョン(ケース 2)





# PIC18F97J60 ファミリ

表 20-4: I<sup>2</sup>C™ の動作に関連するレジスタ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	リセット値の記載ページ
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	69
PIR1	PSP1IF	AD1IF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	71
PIE1	PSP1IE	AD1IE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	71
IPR1	PSP1IP	AD1IP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	71
PIR2	OSCFIF	CMIF	ETHIF	r	BCL1IF	-	TMR3IF	CCP2IF	71
PIE2	OSCFIE	CMIE	ETHIE	r	BCL1IE	-	TMR3IE	CCP2IE	71
IPR2	OSCFIP	CMIP	ETHIP	r	BCL1IP	-	TMR3IP	CCP2IP	71
PIR3	SSP2IF <sup>(1)</sup>	BCL2IF <sup>(1)</sup>	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	71
PIE3	SSP2IE <sup>(1)</sup>	BCL2IE <sup>(1)</sup>	RC2IE	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	71
IPR3	SSP2IP <sup>(1)</sup>	BCL2IP <sup>(1)</sup>	RC2IP	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	71
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	71
TRISD	TRISD7	TRISD6 <sup>(1)</sup>	TRISD5 <sup>(1)</sup>	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	71
SSP1BUF	MSSP1 受信バッファ / 送信レジスタ								70
SSP1ADD	MSSP1 アドレスレジスタ (I <sup>2</sup> C™ スレーブモード)、MSSP1 baud レート再読み込みレジスタ (I <sup>2</sup> C マスタモード)								70
SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	70
SSP1CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	70
	GCEN	ACKSTAT	ADMSK5 <sup>(2)</sup>	ADMSK4 <sup>(2)</sup>	ADMSK3 <sup>(2)</sup>	ADMSK2 <sup>(2)</sup>	ADMSK1 <sup>(2)</sup>	SEN	70
SSP1STAT	SMP	CKE	D/A	P	S	R/W	UA	BF	70
SSP2BUF	MSSP2 受信バッファ / 送信レジスタ								73
SSP2ADD	MSSP2 アドレスレジスタ (I <sup>2</sup> C スレーブモード)、MSSP2 baud レート再読み込みレジスタ (I <sup>2</sup> C マスタモード)								73
SSP2CON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	73
SSP2CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	73
	GCEN	ACKSTAT	ADMSK5 <sup>(2)</sup>	ADMSK4 <sup>(2)</sup>	ADMSK3 <sup>(2)</sup>	ADMSK2 <sup>(2)</sup>	ADMSK1 <sup>(2)</sup>	SEN	73
SSP2STAT	SMP	CKE	D/A	P	S	R/W	UA	BF	73

凡例: - = 未実装ビット、「0」として読み出されます。r = 予約済みです。網掛けの部分は I<sup>2</sup>C™ モードの MSSP モジュールでは使いません。

Note 1: これらのビットを利用できるのは 100 ピンデバイスのみです。それ以外のデバイスには実装されておらず、「0」として読み出されます。

2: I<sup>2</sup>C™ スレーブモードで使う代替ビット定義です。

# PIC18F97J60 ファミリ

---

NOTES:

## 21.0 EUSART (Enhanced Universal Synchronous Asynchronous Receiver Transmitter)

EUSART (Enhanced Universal Synchronous Asynchronous Receiver Transmitter) モジュールは、2つのシリアル I/O モジュールのうちの1つです。(EUSART はシリアル通信インターフェイス (SCI) と呼びます。) EUSART は、CRT 端末やパーソナルコンピュータ等の周辺機器との通信できる全二重非同期通信系として設定できます。また、A/D または D/A IC、シリアル EEPROM 等の周辺デバイスとの通信が可能な、半二重同期システムとしても設定できます。

本モジュールは、baud レートの自動検出と校正、同期ブレーク受信による自動復帰、12ビットのブレーク文字送信等の追加機能も実装しています。これらの機能は、ローカル インターコネクト ネットワークバス (LIN/J2602 バス) システムでの使用に最適です。

PIC18F97J60ファミリの64ピンデバイスは、EUSART1 という名前で EUSART モジュールを1つ搭載しています。80ピン/100ピンデバイスは、2つの独立した EUSART モジュール、EUSART1 と EUSART2 を搭載しています。これらのモジュールは、以下のモードに設定できます。

- 以下の機能を備えた非同期 (全二重) モード：
  - 文字受信による自動復帰
  - 自動 baud レート校正
  - 12ビットブレーク文字送信
- 同期モードークロック極性を選択可能なマスタ (半二重)
- 同期モードークロック極性を選択可能なスレーブ (半二重)

EUSART1 および EUSART2 のピンは、それぞれ PORTC (RC6/TX1/CK1 と RC7/RX1/DT1) と PORTG (RG1/TX2/CK2 と RG2/RX2/DT2) の機能と多重化されています。これらのピンを EUSART で使うには、以下の設定が必要です。

- EUSART1 に対する設定：
  - SPEN ビット (RCSTA1<7>) をセット (= 1)
  - TRISC<7> ビットをセット (= 1)
  - 非同期モードと同期マスタモードの場合 TRISC<6> ビットをクリア (= 0)
  - 同期スレーブモードの場合 TRISC<6> ビットをセット (= 1)
- EUSART2 に対する設定：
  - SPEN ビット (RCSTA2<7>) をセット (= 1)
  - TRISG<2> ビットをセット (= 1)
  - 非同期モードと同期マスタモードの場合 TRISG<1> ビットをクリア (= 0)
  - 同期スレーブモードの場合 TRISG<1> ビットをセット (= 1)

**Note:** EUSARTx の制御により、ピン設定は必要に応じて入力から出力に自動的に変更されます。

各 EUSART モジュールの動作は、以下の3つのレジスタで制御します。

- TXSTAx (送信ステータス / 制御)
- RCSTAx (受信ステータス / 制御)
- BAUDCONx (baud レート制御)

これらのレジスタの詳細は、[レジスタ 21-1](#)、[レジスタ 21-2](#)、[レジスタ 21-3](#)に記載します。

**Note:** このセクション全体を通して、特定の EUSART モジュールに関連するレジスタ名とビット名は、個々のモジュール番号の代わりに「x」を使った総称名で表します。従って、「RCSTAx」は EUSART1 または EUSART2 のいずれかに対応する受信ステータス レジスタを表します。

# PIC18F97J60 ファミリ

## レジスタ 21-1: TXSTAx: 送信ステータス / 制御レジスタ x

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN <sup>(1)</sup>	SYNC	SENDB	BRGH	TRMT	TX9D
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

- bit 7      **CSRC:** クロック源選択ビット  
非同期モード:  
ドントケア  
同期モード:  
1 = マスタモード (クロックを BRG によって内部で生成)  
0 = スレーブモード (外部クロック源を使う)
- bit 6      **TX9:** 9 ビット送信イネーブルビット  
1 = 9 ビット送信を選択する  
0 = 8 ビット送信を選択する
- bit 5      **TXEN:** 送信イネーブルビット<sup>(1)</sup>  
1 = 送信を有効にする  
0 = 送信を無効にする
- bit 4      **SYNC:** EUSARTx モード選択ビット  
1 = 同期モード  
0 = 非同期モード
- bit 3      **SENDB:** ブレーク文字送信ビット  
非同期モード:  
1 = 次の送信で同期ブレークを送信する (完了時にハードウェアによってクリア)  
0 = 同期ブレークの送信が完了している  
同期モード:  
ドントケア
- bit 2      **BRGH:** 高 baud レート選択ビット  
非同期モード:  
1 = 高速  
0 = 低速  
同期モード:  
このモードでは未使用
- bit 1      **TRMT:** 送信シフトレジスタ ステータスビット  
1 = TSR がエンプティである  
0 = TSR がフルである
- bit 0      **TX9D:** 送信データの 9 ビット目  
アドレス / データビットまたはパリティビットとして使える

**Note 1:** 同期モードでは、SREN/CREN の設定が TXEN よりも優先されます。

# PIC18F97J60 ファミリ

## レジスタ 21-2: RCSTAx: 受信ステータス / 制御レジスタ x

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

- bit 7      **SPEN:** シリアルポート イネーブルビット  
 1 = シリアルポートを有効にする (RXx/DTx と TXx/CKx ピンをシリアルポート ピンに設定する)  
 0 = シリアルポートを無効にする (リセットに保持する)
- bit 6      **RX9:** 9 ビット受信イネーブルビット  
 1 = 9 ビット受信を選択する  
 0 = 8 ビット受信を選択する
- bit 5      **SREN:** シングル受信イネーブルビット  
非同期モード:  
 ドントケア  
同期モード - マスタ:  
 1 = シングル受信を有効にする  
 0 = シングル受信を無効にする  
 このビットは受信完了後にクリアされます。  
同期モード - スレーブ:  
 ドントケア
- bit 4      **CREN:** 連続受信イネーブルビット  
非同期モード:  
 1 = レシーバを有効にする  
 0 = レシーバを無効にする  
同期モード:  
 1 = イネーブルビット CREN がクリアされるまで連続受信を有効にする (CREN の設定が SREN より優先される)  
 0 = 連続受信を無効にする
- bit 3      **ADDEN:** アドレス検出イネーブルビット  
9 ビット非同期モード (RX9 = 1) の場合:  
 1 = アドレス検出を有効にし、割り込みを有効にする。RSR<8> がセットされたら受信バッファに転送する  
 0 = アドレス検出を無効にする。全てのバイトが受信され、9 ビット目はパリティビットとして使える  
9 ビット非同期モード (RX9 = 0) の場合:  
 ドントケア
- bit 2      **FERR:** フレーミング エラービット  
 1 = フレーミング エラーあり (RCREGx レジスタを読み出し、次の有効なバイトを受信する事で更新できる)  
 0 = フレーミング エラーなし
- bit 1      **OERR:** オーバーラン エラービット  
 1 = オーバーラン エラーあり (CREN ビットをクリアする事でクリアできる)  
 0 = オーバーラン エラーなし
- bit 0      **RX9D:** 受信データの 9 ビット目  
 アドレス / データビットまたはパリティビットとして使える。ユーザ ファームウェアによる計算が必要である

# PIC18F97J60 ファミリ

## レジスタ 21-3: BAUDCONx: baud レート制御レジスタ x

R/W-0	R-1	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

- bit 7      **ABDOVF:** 自動 baud レート アクイジション ロールオーバー ステータスビット  
 1 = 自動 baud レート検出モードで BRG のロールオーバーが発生した (ソフトウェアによるクリアが必要である)  
 0 = BRG のロールオーバーは発生していない
- bit 6      **RCIDL:** 受信動作アイドル ステータスビット  
 1 = 受信動作がアイドルである  
 0 = 受信動作がアクティブである
- bit 5      **RXDTP:** 受信データ極性選択ビット  
非同期モード:  
 1 = 受信データ (RXx) を反転する。アイドル状態を Low レベルに設定する  
 0 = 受信データ (RXx) を反転しない。アイドル状態を High レベルに設定する  
同期モード:  
 1 = データ (DTx) を反転する。アイドル状態を Low レベルに設定する  
 0 = データ (DTx) を反転しない。アイドル状態を High レベルに設定する
- bit 4      **TXCKP:** クロック/データ極性選択ビット  
非同期モード:  
 1 = 送信データ (TXx) を反転する。アイドル状態を Low レベルに設定する  
 0 = 送信データ (TXx) を反転しない。アイドル状態を High レベルに設定する  
同期モード:  
 1 = クロック (CKx) のアイドル状態を High レベルに設定する  
 0 = クロック (CKx) のアイドル状態を Low レベルに設定する
- bit 3      **BRG16:** 16 ビット baud レート レジスタ イネーブルビット  
 1 = 16 ビット baud レート ジェネレータ - SPBRGHx と SPBRGx を使う  
 0 = 8 ビット baud レート ジェネレータ - SPBRGx のみを使う。SPBRGHx の値は無視する (互換モード)
- bit 2      **未実装:** 「0」として読み出し
- bit 1      **WUE:** 復帰イネーブルビット  
非同期モード:  
 1 = EUSARTx は RXx ピンのサンプリングを継続する。割り込みは立ち上がりエッジで発生し、次の立ち上がりエッジでこのビットはハードウェアによってクリアされる  
 0 = RXx ピンを監視しない、または立ち上がりエッジが検出された  
同期モード:  
 このモードでは未使用
- bit 0      **ABDEN:** 自動 baud レート検出イネーブルビット  
非同期モード:  
 1 = 次の文字で baud レート計測を有効にする。同期フィールド (55h) を受信する必要がある。完了時にハードウェアによってクリアされる  
 0 = baud レート計測を無効にする、または計測が完了した  
同期モード:  
 このモードでは未使用

## 21.1 baud レート ジェネレータ (BRG)

BRG は、EUSARTx の非同期および同期モードの両方をサポートする、8 ビットまたは 16 ビットの専用ジェネレータです。既定値では BRG は 8 ビットモードで動作します。BRG16 ビット (BAUDCONx<3>) をセットすると、16 ビットモードで動作します。

フリーランニングタイマの周期は SPBRGHx:SPBRGx レジスタペアによって制御します。非同期モードでは、BRGH (TXSTAx<2>) および BRG16 (BAUDCONx<3>) ビットも baud レートを制御します。同期モードでは、BRGH は無視されます。表 21-1 に、マスタモード (クロックを内部で生成するモード) のみに適用される、EUSARTx の各種モードに対する baud レートの計算式を示します。

目標 baud レートと Fosc から、表 21-1 の式を使って、SPBRGHx:SPBRGx レジスタに設定する最も近い整数値を計算できます。その値から baud レートの誤差を求められます。式 21-1 に計算例を示します。表 21-2 に、各種非同期モードにおける代表的な baud レートと誤差を示します。baud レート誤差の低減が必要な場合、ま

たは高速のオシレータ周期から低速の baud レートを実現する場合、高 baud レート (BRGH = 1) を選択するか、16 ビットの BRG を使った方が有利な場合があります。

SPBRGHx:SPBRGx レジスタペアに新しい値を書き込むと、BRG タイマがリセット (クリア) されます。これにより、BRG はタイマ オーバーフローを待たずに新しい baud レートを出力できます。

### 21.1.1 電力管理モードにおける動作

目標 baud レートの生成にはデバイスクロックを使います。いずれかの電力管理モードに移行した場合、新しいクロック源はそれまでと異なる周波数で動作している可能性があります。その場合、SPBRGx レジスタペアの値に調整が必要になる事があります。

### 21.1.2 サンプリング

RXx ピン (RC7/RX1/DT1 または RG2/RX2/DT2 のいずれか) のデータを多数決検出回路によって 3 回サンプリングして、High/Low いずれのレベルにあるかを判断します。

表 21-1: baud レートの計算式

コンフィグレーション ビット			BRG/EUSARTx のモード	baud レートの計算式
SYNC	BRG16	BRGH		
0	0	0	8 ビット / 非同期	$F_{osc}/[64 (n + 1)]$
0	0	1	8 ビット / 非同期	$F_{osc}/[16 (n + 1)]$
0	1	0	16 ビット / 非同期	
0	1	1	16 ビット / 非同期	$F_{osc}/[4 (n + 1)]$
1	0	x	8 ビット / 同期	
1	1	x	16 ビット / 同期	

凡例: x = ドントケア、n = SPBRGHx:SPBRGx レジスタペアの値

# PIC18F97J60 ファミリ

## 式 21-1: baud レート誤差の計算

FOSC = 16 MHz、目標 baud レート = 9600、非同期モード、8 ビット BRG のデバイスの場合：

目標 baud レート =  $F_{OSC}/(64 ([SPBRGHx:SPBRGx] + 1))$

SPBRGHx:SPBRGx について解くと：

$$\begin{aligned} X &= ((F_{OSC}/\text{目標 baud レート})/64) - 1 \\ &= ((16000000/9600)/64) - 1 \\ &= [25.042] = 25 \end{aligned}$$

baud レートの計算値 =  $16000000/(64(25 + 1))$

$$= 9615$$

誤差 =  $(\text{baud レートの計算値} - \text{目標 baud レート})/\text{目標 baud レート}$

$$= (9615 - 9600)/9600 = 0.16\%$$

表 21-2: baud レート ジェネレータに関連するレジスタ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	リセット値の 記載ページ
TXSTAx	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	71
RCSTAx	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	71
BAUDCONx	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	-	WUE	ABDEN	72
SPBRGHx	EUSARTx baud レート ジェネレータ レジスタ上位バイト								72
SPBRGx	EUSARTx baud レート ジェネレータ レジスタ下位バイト								71

凡例： - = 未実装ビット、「0」として読み出されます。網掛けの部分は BRG では使いません。



# PIC18F97J60 ファミリ

表 21-3: 各種非同期モードにおける baud レート

baud レート (K)	SYNC = 0、BRG16 = 0、BRGH = 0											
	Fosc = 41.667 MHz			Fosc = 31.25 MHz			Fosc = 25.000 MHz			Fosc = 20.833 MHz		
	実際の レート (K)	% 誤差	SPBRG の値 (10 進数)	実際の レート (K)	% 誤差	SPBRG の値 (10 進数)	実際の レート (K)	% 誤差	SPBRG の値 (10 進数)	実際の レート (K)	% 誤差	SPBRG の値 (10 進数)
0.3	—	—	—	—	—	—	—	—	—	—	—	—
1.2	—	—	—	—	—	—	—	—	—	1.271	5.96	255
2.4	2.543	5.96	255	2.405	0.22	202	2.396	-0.15	162	2.393	-0.27	135
9.6	9.574	-0.27	67	9.574	-0.27	50	9.527	-0.76	40	9.574	-0.27	33
19.2	19.148	-0.27	33	19.531	1.73	24	19.531	1.73	19	19.147	-0.27	16
57.6	59.186	2.75	10	61.035	5.96	7	55.804	-3.12	6	54.253	-5.81	5
115.2	108.508	-5.81	5	122.070	5.96	3	130.208	13.03	2	108.505	-5.81	2

baud レート (K)	SYNC = 0、BRG16 = 0、BRGH = 0								
	Fosc = 13.889 MHz			Fosc = 6.250 MHz			Fosc = 4.167 MHz		
	実際の レート (K)	% 誤差	SPBRG の値 (10 進数)	実際の レート (K)	% 誤差	SPBRG の値 (10 進数)	実際の レート (K)	% 誤差	SPBRG の値 (10 進数)
0.3	—	—	—	—	—	—	0.300	0.01	216
1.2	1.198	-0.08	180	1.206	0.47	80	1.206	0.48	53
2.4	2.411	0.47	89	2.382	-0.76	40	2.411	0.48	26
9.6	9.435	-1.71	22	9.766	1.73	9	9.301	-3.11	6
19.2	19.279	2.75	10	19.531	1.73	4	21.703	13.04	2
57.6	54.254	-5.81	3	48.828	-15.23	1	65.109	13.04	0
115.2	108.508	-5.81	1	97.656	-15.23	0	65.109	-43.48	0

baud レート (K)	SYNC = 0、BRG16 = 0、BRGH = 1											
	Fosc = 41.667 MHz			Fosc = 31.25 MHz			Fosc = 25.000 MHz			Fosc = 20.833 MHz		
	実際の レート (K)	% 誤差	SPBRG の値 (10 進数)	実際の レート (K)	% 誤差	SPBRG の値 (10 進数)	実際の レート (K)	% 誤差	SPBRG の値 (10 進数)	実際の レート (K)	% 誤差	SPBRG の値 (10 進数)
0.3	—	—	—	—	—	—	—	—	—	—	—	—
1.2	—	—	—	—	—	—	—	—	—	—	—	—
2.4	—	—	—	—	—	—	—	—	—	—	—	—
9.6	10.172	5.96	255	9.621	0.22	202	9.586	-0.15	162	9.573	-0.27	135
19.2	19.148	-0.27	135	19.148	-0.27	101	19.290	0.47	80	19.147	-0.27	67
57.6	57.871	0.47	44	57.445	-0.27	33	57.870	0.47	26	56.611	-1.72	22
115.2	113.226	-1.71	22	114.890	-0.27	16	111.607	-3.12	13	118.369	2.75	10

baud レート (K)	SYNC = 0、BRG16 = 0、BRGH = 1								
	Fosc = 13.889 MHz			Fosc = 6.250 MHz			Fosc = 4.167 MHz		
	実際の レート (K)	% 誤差	SPBRG の値 (10 進数)	実際の レート (K)	% 誤差	SPBRG の値 (10 進数)	実際の レート (K)	% 誤差	SPBRG の値 (10 進数)
0.3	—	—	—	—	—	—	—	—	—
1.2	—	—	—	—	—	—	1.200	0.01	216
2.4	—	—	—	2.396	-0.15	162	2.389	-0.44	108
9.6	9.645	0.47	89	9.527	-0.76	40	9.645	0.48	26
19.2	19.290	0.47	44	19.531	1.73	19	18.603	-3.11	13
57.6	57.871	0.47	14	55.804	-3.12	6	52.088	-9.57	4
115.2	108.508	-5.81	7	130.208	13.03	2	130.219	13.04	1

# PIC18F97J60 ファミリ

表 21-3: 各種非同期モードにおける baud レート ( 続き )

baud レート (K)	SYNC = 0、BRG16 = 1、BRGH = 0											
	Fosc = 41.667 MHz			Fosc = 31.25 MHz			Fosc = 25.000 MHz			Fosc = 20.833 MHz		
	実際の レート (K)	% 誤差	SPBRG の値 (10 進数)	実際の レート (K)	% 誤差	SPBRG の値 (10 進数)	実際の レート (K)	% 誤差	SPBRG の値 (10 進数)	実際の レート (K)	% 誤差	SPBRG の値 (10 進数)
0.3	0.300	0.00	8680	0.300	0.00	6509	0.300	0.01	5207	0.300	0.00	4339
1.2	1.200	0.01	2169	1.200	-0.02	1627	1.200	0.01	1301	1.200	0.00	1084
2.4	2.400	0.01	1084	2.399	-0.02	813	2.400	0.01	650	2.398	-0.09	542
9.6	9.609	0.10	270	9.621	0.22	202	9.586	-0.15	162	9.574	-0.27	135
19.2	19.148	-0.27	135	19.148	-0.27	101	19.290	0.47	80	19.148	-0.27	67
57.6	57.871	0.47	44	57.444	-0.27	33	57.870	0.47	26	56.611	-1.72	22
115.2	113.226	-1.71	22	114.890	-0.27	16	111.607	-3.12	13	118.369	2.75	10

baud レート (K)	SYNC = 0、BRG16 = 1、BRGH = 0								
	Fosc = 13.889 MHz			Fosc = 6.250 MHz			Fosc = 4.167 MHz		
	実際の レート (K)	% 誤差	SPBRG の値 (10 進数)	実際の レート (K)	% 誤差	SPBRG の値 (10 進数)	実際の レート (K)	% 誤差	SPBRG の値 (10 進数)
0.3	0.300	-0.02	2893	0.300	0.01	1301	0.300	0.01	867
1.2	1.201	0.05	722	1.198	-0.15	325	1.200	0.01	216
2.4	2.398	-0.08	361	2.396	-0.15	162	2.389	-0.44	108
9.6	9.645	0.47	89	9.527	-0.76	40	9.646	0.48	26
19.2	19.290	0.47	44	19.531	1.73	19	18.603	-3.11	13
57.6	57.871	0.47	14	55.804	-3.12	6	52.088	-9.57	4
115.2	108.508	-5.81	7	130.208	13.03	2	130.218	13.04	1

baud レート (K)	SYNC = 0、BRG16 = 1、BRGH = 1 または SYNC = 1、BRG16 = 1											
	Fosc = 41.667 MHz			Fosc = 31.25 MHz			Fosc = 25.000 MHz			Fosc = 20.833 MHz		
	実際の レート (K)	% 誤差	SPBRG の値 (10 進数)	実際の レート (K)	% 誤差	SPBRG の値 (10 進数)	実際の レート (K)	% 誤差	SPBRG の値 (10 進数)	実際の レート (K)	% 誤差	SPBRG の値 (10 進数)
0.3	0.300	0.00	34722	0.300	0.00	26041	0.300	0.00	20832	0.300	0.00	17360
1.2	1.200	0.00	8680	1.200	0.01	6509	1.200	0.01	5207	1.200	0.00	4339
2.4	2.400	0.01	4339	2.400	0.01	3254	2.400	0.01	2603	2.400	0.00	2169
9.6	9.601	0.01	1084	9.598	-0.02	813	9.601	0.01	650	9.592	-0.09	542
19.2	19.184	-0.08	542	19.195	-0.02	406	19.172	-0.15	325	19.219	0.10	270
57.6	57.551	-0.08	180	57.445	-0.27	135	57.339	-0.45	108	57.869	0.47	89
115.2	115.742	0.47	89	114.890	-0.27	67	115.741	0.47	53	115.739	0.47	44

baud レート (K)	SYNC = 0、BRG16 = 1、BRGH = 1 または SYNC = 1、BRG16 = 1								
	Fosc = 13.889 MHz			Fosc = 6.250 MHz			Fosc = 4.167 MHz		
	実際の レート (K)	% 誤差	SPBRG の値 (10 進数)	実際の レート (K)	% 誤差	SPBRG の値 (10 進数)	実際の レート (K)	% 誤差	SPBRG の値 (10 進数)
0.3	0.300	0.00	11573	0.300	0.01	5207	0.300	-0.01	3472
1.2	1.200	-0.02	2893	1.200	0.01	1301	1.200	0.01	867
2.4	2.400	-0.02	1446	2.400	0.01	650	2.400	0.01	433
9.6	9.592	-0.08	361	9.586	-0.15	162	9.557	-0.44	108
19.2	19.184	-0.08	180	19.290	0.47	80	19.292	0.48	53
57.6	57.870	0.47	59	57.870	0.47	26	57.875	0.48	17
115.2	115.742	0.47	29	111.607	-3.12	13	115.750	0.48	8

## 21.1.3 自動 baud レート検出

EUSARTx モジュールは baud レートの自動検出/校正をサポートしています。この機能は、非同期モードで WUE ビットがクリアされている場合のみ有効です。

スタートビットを受信し ABDEN ビットがセットされるたびに自動 baud レートの計測シーケンス(図 21-1)が開始します。計算は平均値です。

自動 baud レート検出 (ABD) モードでは、BRG へのクロックが逆方向です。BRG が入力 RXx 信号にクロックを供給するのではなく、RXx 信号が BRG のタイミングを生成します。ABD モードでは、内部の baud レートジェネレータを、入力シリアルバイトストリームのビット周期タイミングを生成するカウンタとして使います。

ABDEN ビットがセットされると、ステートマシンは BRG をクリアしてスタートビットを待機します。自動 baud レート検出では、正確なビットレートを計算するために、値が 55h (ASCII の「U」、LIN/J2602 バスの同期文字でもある)のバイトを受信する必要があります。Low ビット時間と High ビット時間の両方を計測するため、入力信号の非対称性による影響を最小限に抑えられます。スタートビットの受信後、あらかじめ選択したクロック源により、RXx の最初の立ち上がりエッジから SPBRGx のカウントアップを開始します。RXx ピンで 8 ビットを受信した時点、つまり 5 番目の立ち上がりエッジで、SPBRGHx:SPBRGx レジスタペアには正確な BRG 周期を合計した累積値が格納されています。5 番目のエッジ (ストップビットに相当) が検出された時点で、ABDEN ビットは自動的にクリアされます。

BRG でロールオーバー(FFFFh から 0000h へのオーバーフロー)が発生した場合、このイベントは ABDOVF ステータスビット (BAUDCONx<7>) にトラップされます。ステータスビットは BRG のロールオーバーによってハードウェアでセットされ、ユーザがソフトウェアでセットまたはクリアできます。ABD モードはロールオーバー イベントの発生後アクティブな状態を保ち、ABDEN ビットはセットされたままです (図 21-2)。

baud レート周期の校正時、BRG レジスタは、あらかじめ設定されたクロックレートの 1/8 でクロック供給されます。BRG クロックは、BRG16 ビットと BRGH ビットで設定される事に注意してください。BRG16 ビットの設定に関わらず、SPBRGx と SPBRGHx の両方を 16 ビットのカウントとして使います。これによって、ユーザは SPBRGHx レジスタの 00h を確認する事で、8 ビットモードでキャリーが発生していない事を検証できます。BRG のカウンタクロックのレートは、表 21-4 を参照してください。

ABD シーケンスの実行中、EUSARTx のステートマシンはアイドルに保持されます。RXx で 5 番目の立ち上がりエッジが検出されると、RCxIF 割り込みがセットされます。RCxIF 割り込みをクリアするには、RCREGx の値を読み出す必要があります。RCREGx の内容は破棄します。

**Note 1:** ABDEN ビットと WUE ビットを両方セットした場合、自動 baud レート検出はブレーク文字直後のバイトで実行されます。

**2:** 入力文字の baud レートが、選択した BRG クロック源のレンジ内であるかどうかの判断は、ユーザに委ねられています。オシレータ周波数と EUSARTx の baud レートの組み合わせによっては、ビットエラー率のために使えないものがあります。自動 baud レート検出機能を使う場合、システム全体のタイミングと通信の baud レートを考慮する必要があります。

表 21-4: BRG カウンタのクロックレート

BRG16	BRGH	BRG のカウンタクロック
0	0	Fosc/512
0	1	Fosc/128
1	0	Fosc/128
1	1	Fosc/32

**Note:** ABD シーケンスでは、BRG16 の設定に関係なく SPBRGx と SPBRGHx の両方を 16 ビットカウンタとして使います。

### 21.1.3.1 ABD と EUSARTx の送信

ABD アクイジションの実行中は BRG クロックの方向が反転するため、ABD モードでは EUSARTx のトランスミッタを使えません。すなわち、ABDEN ビットがセットされている間は TXREGx に書き込みできません。送信シーケンスの実行中に ABDEN ビットがセットされないようにする事も必要です。これを怠ると、EUSARTx の予期せぬ動作が生じる恐れがあります。

# PIC18F97J60 ファミリ

図 21-1: baud レートの自動計算

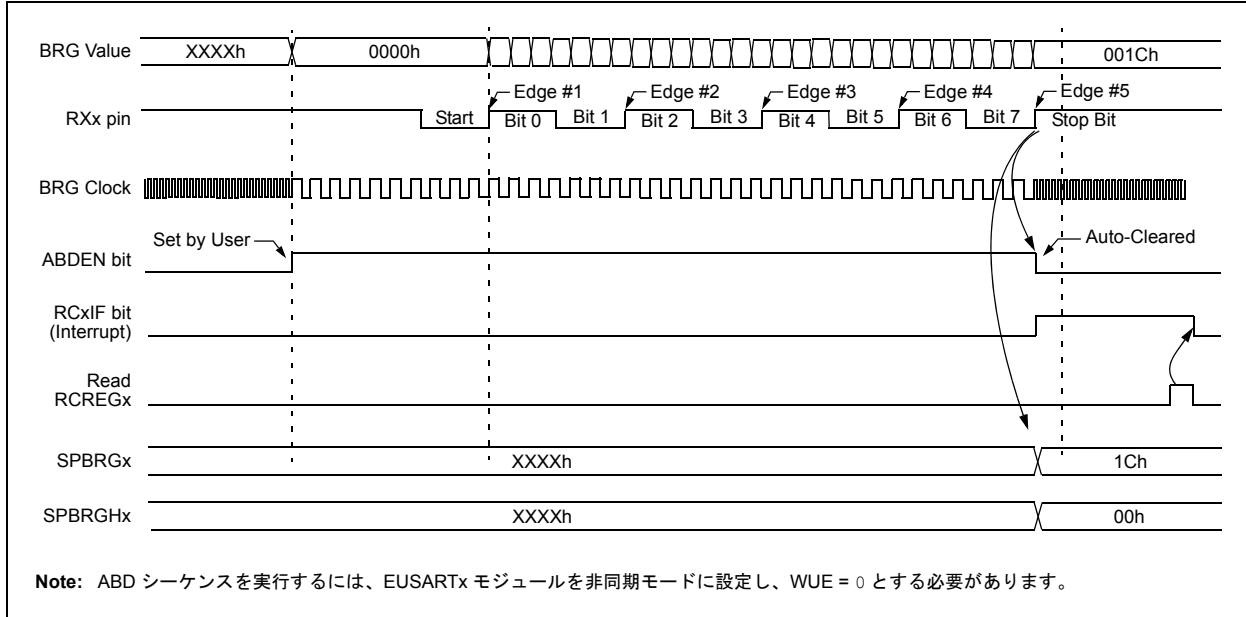
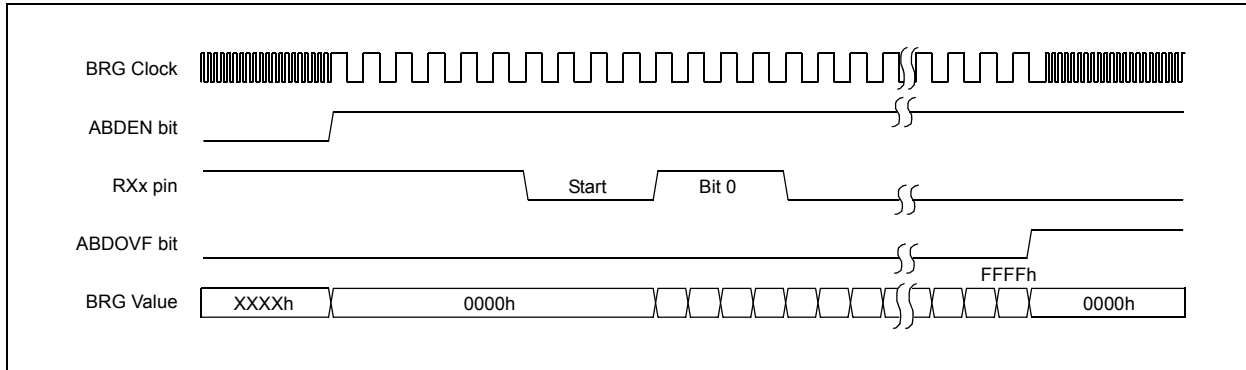


図 21-2: BRG のオーバーフロー シーケンス



## 21.2 EUSARTx 非同期モード

非同期モードの動作は、SYNC ビット (TXSTAx<4>) をクリアする事で選択します。このモードで EUSARTx は標準の非ゼロ復帰 (NRZ) フォーマット (1x スタートビット、8 または 9x データビット、1x ストップビット) を使います。最も一般的なデータ フォーマットは 8 ビットです。内蔵の専用 8 ビット / 16 ビット baud レート ジェネレータを使って、オシレータから標準 baud レート周波数を生成できます。

EUSARTx は LSb から順に送受信します。EUSARTx モジュールのトランスミッタとレシーバは機能的には独立していますが、同じデータ フォーマットと baud レートを使います。baud レート ジェネレータは BRGH ビットと BRG16 ビット (TXSTAx<2> と BAUDCONx<3>) に応じて、x16 または x64 いずれかのビットシフトレートのクロックを生成します。パリティはハードウェアではサポートされていませんが、ソフトウェアによって実装し、9 番目のデータビットに格納できます。

TXCKP (BAUDCONx<4>) ビットと RXDTP (BAUDCONx<5>) ビットにより、Tx 信号と Rx 信号を反転 (極性反転) できます。TTL と RS-232 レベル間で信号をバッファリングするデバイスも信号を反転します。TXCKP および RXDTP ビットをセットすると、信号を反転せずにバッファリングする回路が使えます。

非同期モードで動作する場合、EUSARTx モジュールは以下の重要な要素で構成されます。

- baud レート ジェネレータ
- サンプリング回路
- 非同期トランスミッタ
- 非同期レシーバ
- 同期ブレイク文字による自動復帰
- 12 ビットのブレイク文字送信
- 自動 baud レート検出

### 21.2.1 EUSARTx 非同期トランスミッタ

図 21-3 に、EUSARTx トランスミッタのブロック図を示します。トランスミッタの中核となるのが送信 (シリアル) シフトレジスタ (TSR) です。シフトレジスタには、読み書き送信バッファレジスタ TXREGx からデータを入力します。TXREGx レジスタにはソフトウェアでデータを書き込みます。前回書き込まれたデータのストップビットが送信されるまで、TSR レジスタに新しいデータは転送されません。ストップビットが送信されると、TXREGx レジスタから TSR に新しいデータが転送されます (送信すべきデータが存在する場合)。

TXREGx レジスタは、TSR レジスタへのデータ転送が完了すると (1 TcY で完了) エンプティになり、TXxIF フラグビットがセットされます。この割り込みは、割り込みイネーブルビット TXxIE をセットまたはクリアする事で有効または無効にできます。TXxIF は、TXxIE の状態に関係なくセットされ、ソフトウェアではクリアできません。TXxIF は、TXREGx の書き込みによっても即座にはクリアされず、読み込み命令後 2 番目の命令で有効になります。TXREGx への書き込み直後に TXxIF をポーリングしても、戻り値は無効です。

TXxIF が TXREGx レジスタのステータスを示す一方、TRMT ビット (TXSTAx<1>) は TSR レジスタのステータスを示します。TRMT は TSR レジスタがエンプティになるとセットされる、読み出し専用のビットです。このビットには割り込みロジックが関連付けられていないため、TSR レジスタがエンプティであるかどうかを判断するには、このビットをポーリングする必要があります。

**Note 1:** TSRレジスタはデータメモリに割り当てられていないため、ユーザからはアクセスできません。

**2:** イネーブルビットTXENをセットすると、フラグビットTXxIFがセットされます。

非同期送信を設定するには、以下の手順を実行します。

1. 適切な baud レートに合わせて、SPBRGHx:SPBRGx レジスタを初期化する。必要に応じて BRGH および BRG16 ビットをセットまたはクリアして、目標 baud レートに設定する。
2. SYNC ビットをクリアし、SPEN ビットをセットして非同期シリアルポートを有効にする。
3. Tx ピンからの信号を反転する場合、TXCKP ビットをセットする。
4. 割り込みを使う場合、イネーブルビット TXxIE をセットする。
5. 9 ビット送信を使う場合、送信ビット TX9 をセットする。このビットはアドレス / データとして使える。
6. TXEN ビットをセットして (TXxIF ビットもセットされる)、送信を有効にする。
7. 9 ビット送信を選択した場合、TX9D ビットに 9 ビット目を読み込む必要がある。
8. TXREGx レジスタにデータを読み込む (送信を開始)。
9. 割り込みを使う場合、INTCON レジスタの GIE および PEIE ビット (INTCON<7:6>) がセットされている事を確認する。

# PIC18F97J60 ファミリ

図 21-3: EUSARTx 送信のブロック図

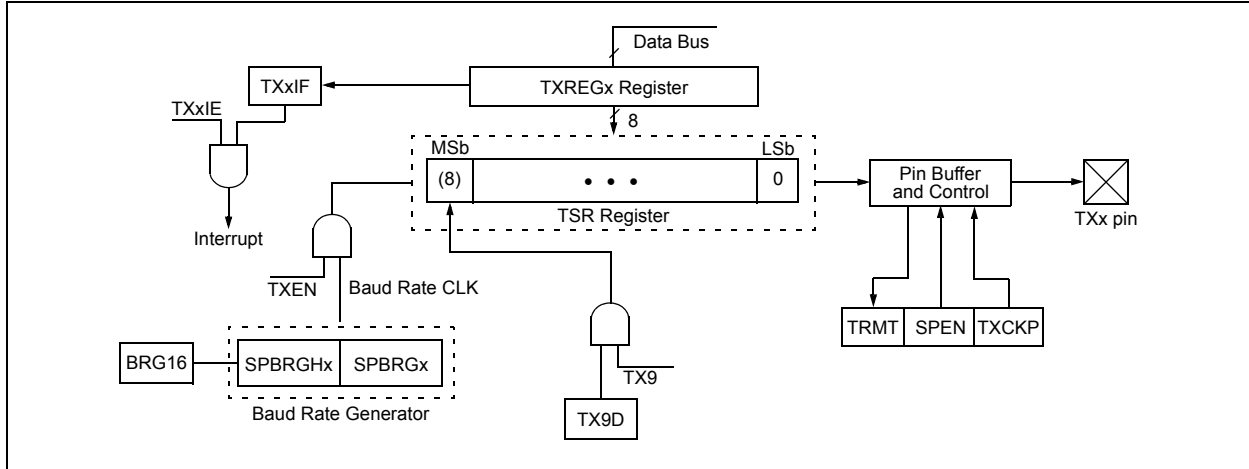


図 21-4: 非同期送信、TXCKP = 0 (txx を反転しない)

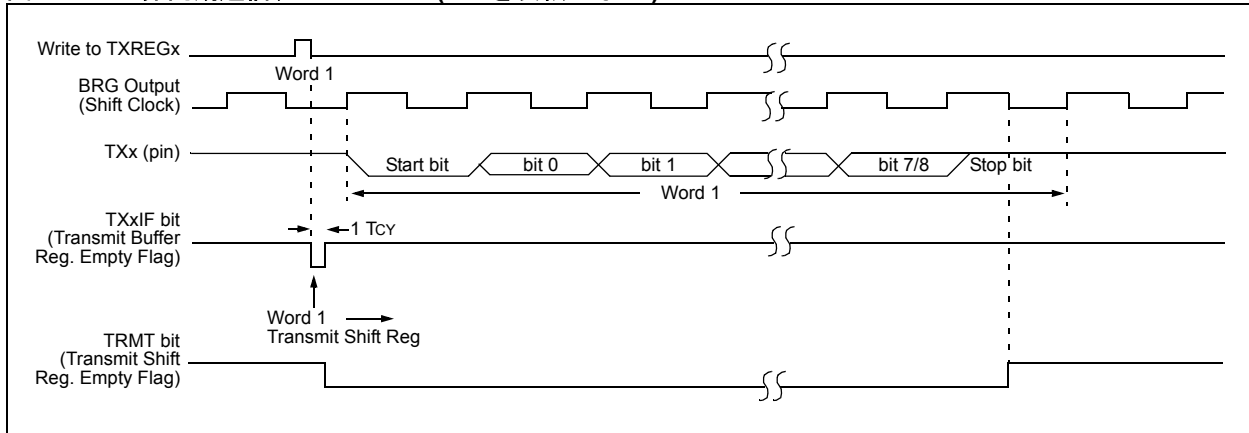
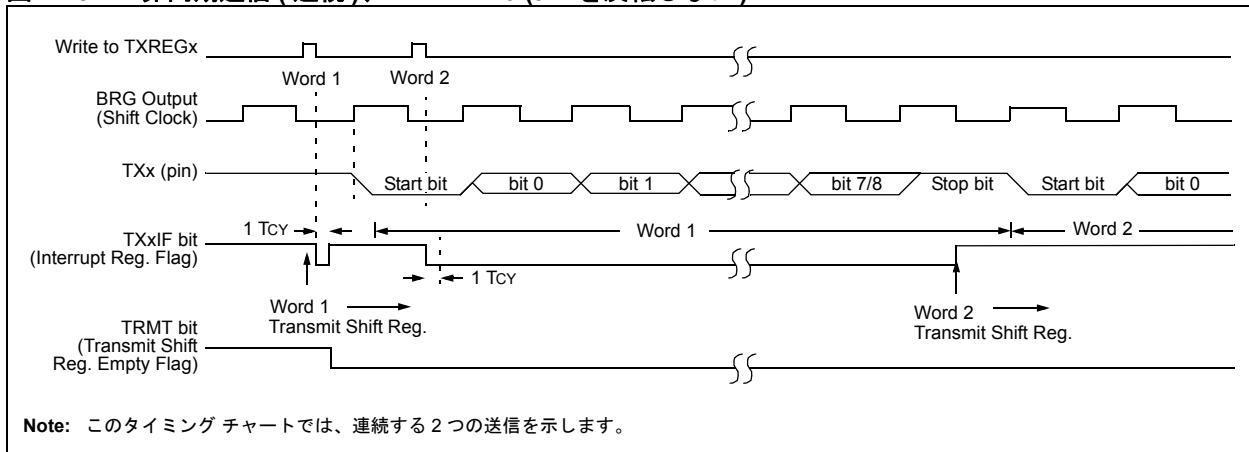


図 21-5: 非同期送信 (連続)、TXCKP = 0 (txx を反転しない)



Note: このタイミングチャートでは、連続する2つの送信を示します。

# PIC18F97J60 ファミリ

表 21-5: 非同期送信に関連するレジスタ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	リセット 値の記載 ページ
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	69
PIR1	PSPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	71
PIE1	PSPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	71
IPR1	PSPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	71
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF <sup>(1)</sup>	TMR4IF	CCP5IF	CCP4IF	CCP3IF	71
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE <sup>(1)</sup>	TMR4IE	CCP5IE	CCP4IE	CCP3IE	71
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP <sup>(1)</sup>	TMR4IP	CCP5IP	CCP4IP	CCP3IP	71
RCSTAx	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	71
TXREGx	EUSARTx 送信レジスタ								71
TXSTAx	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	71
BAUDCONx	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	-	WUE	ABDEN	72
SPBRGHx	EUSARTx baud レート ジェネレータ レジスタ上位バイト								72
SPBRGx	EUSARTx baud レート ジェネレータ レジスタ下位バイト								73

凡例: - = 未実装ビット、「0」として読み出されます。網掛けの部分は非同期送信では使いません。

Note 1: これらのビットは 80 ピン/100 ピンデバイスにのみ実装されています。それ以外のデバイスでは未実装で「0」として読み出されます。

## 21.2.2 EUSARTx 非同期レシーバ

図 21-6 に、レシーバのブロック図を示します。RXx ピンで受信したデータでデータ復元ブロックを駆動します。データ復元ブロックの実体は、baud レートの 16 倍速で動作する高速シフタです。一方、メイン受信シリアルシフタはビットレートまたは FOSC で動作します。通常、このモードは RS-232 システムで使います。

RXDTP (BAUDCON<5>) ビットにより、RXx 信号を反転 (極性反転) できます。RS-232 から TTL レベルへの信号をバッファリングするデバイスも信号を反転させます (RS-232 = 正の場合、TTL = 0)。RXDTP ビットをセットして RXx ピンのデータの極性を反転する事で、信号を反転せずにバッファリングする回路が使えます。

非同期受信を設定するには、以下の手順を実行します。

1. 適切な baud レートに合わせて、SPBRGHx:SPBRGx レジスタを初期化する。必要に応じて BRGH および BRG16 ビットをセットまたはクリアして、目標 baud レートに設定する。
2. SYNC ビットをクリアし、SPEN ビットをセットして非同期シリアルポートを有効にする。
3. RXx ピンの信号を反転する場合、RXDTP ビットをセットする。
4. 割り込みを使う場合、イネーブルビット RCxIE をセットする。
5. 9 ビット受信を使う場合、RX9 ビットをセットする。
6. CREN ビットをセットして、受信を有効にする。
7. イネーブルビット RCxIE をセットした場合、受信が完了するとフラグビット RCxIF がセットされ、割り込みが生成される。
8. 9 ビット受信が有効な場合は RCSTAx レジスタを読み出して 9 ビット目を取得し、受信中に何らかのエラーが発生しなかったかを判断する。
9. RCREGx レジスタを読み出して、8 ビットの受信データを読み出す。
10. 何らかのエラーが発生していた場合、イネーブルビット CREN をクリアしてエラーをクリアする。
11. 割り込みを使う場合、INTCON レジスタの GIE および PEIE ビット (INTCON<7:6>) がセットされている事を確認する。

## 21.2.3 9 ビットアドレス検出モードの設定

通常、このモードは RS-485 システムで使います。非同期受信でアドレス検出を有効にするには、以下の手順を実行します。

1. 適切な baud レートに合わせて、SPBRGHx:SPBRGx レジスタを初期化する。必要に応じて BRGH および BRG16 ビットをセットまたはクリアして、目標 baud レートに設定する。
2. SYNC ビットをクリアし、SPEN ビットをセットして非同期シリアルポートを有効にする。
3. RXx ピンの信号を反転する場合、RXDTP ビットをセットする。TXx ピンからの信号を反転する場合、TXCKP ビットをセットする。
4. 割り込みが必要な場合、RCEN ビットをセットし、RCxIP ビットで優先度レベルを選択する。
5. RX9 ビットをセットして、9 ビット受信を有効にする。
6. ADDEN ビットをセットして、アドレス検出を有効にする。
7. CREN ビットをセットして、受信を有効にする。
8. 受信が完了すると、RCxIF ビットがセットされる。RCxIE および GIE ビットをセットしている場合、割り込みが通知される。
9. RCSTAx レジスタを読み出して受信中に何らかのエラーが発生しなかったかを判断する。9 ビットデータ受信の場合は 9 ビット目のデータビットも読み出す。
10. RCREGx を読み出す事で、このデバイスのアドレスが指定されているかを判断する。
11. 何らかのエラーが発生していた場合、CREN ビットをクリアする。
12. このデバイスのアドレスが指定された場合、ADDEN ビットをクリアして全ての受信データを受信バッファに取り込み、CPU に対する割り込みが生成できるようにする。



図 21-6: EUSARTx 受信のブロック図

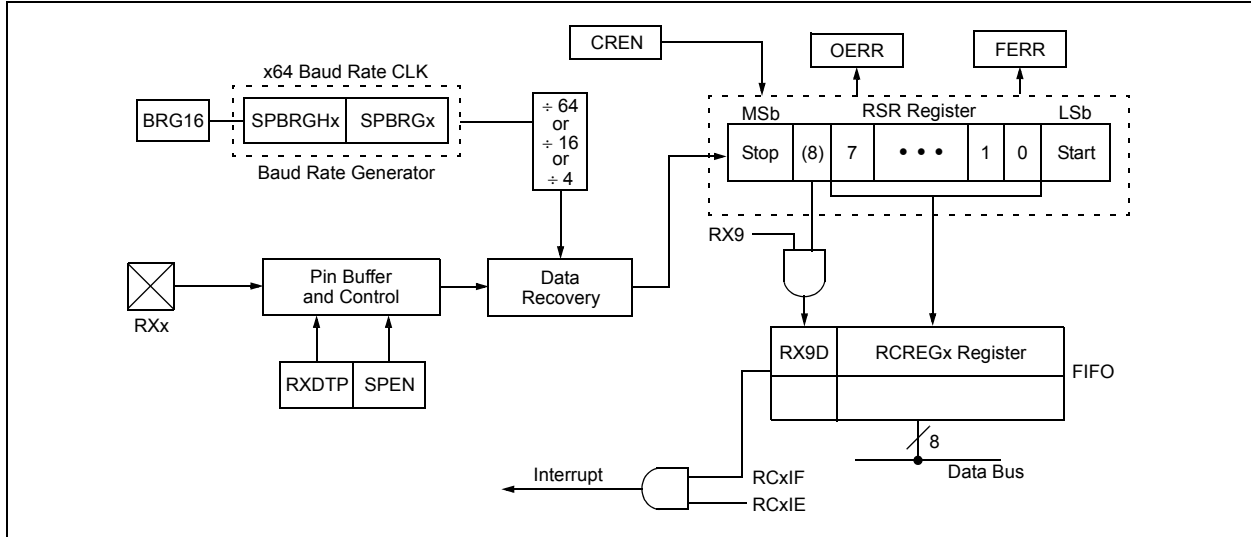


図 21-7: 非同期受信、RXDTP = 0 (RXx を反転しない)

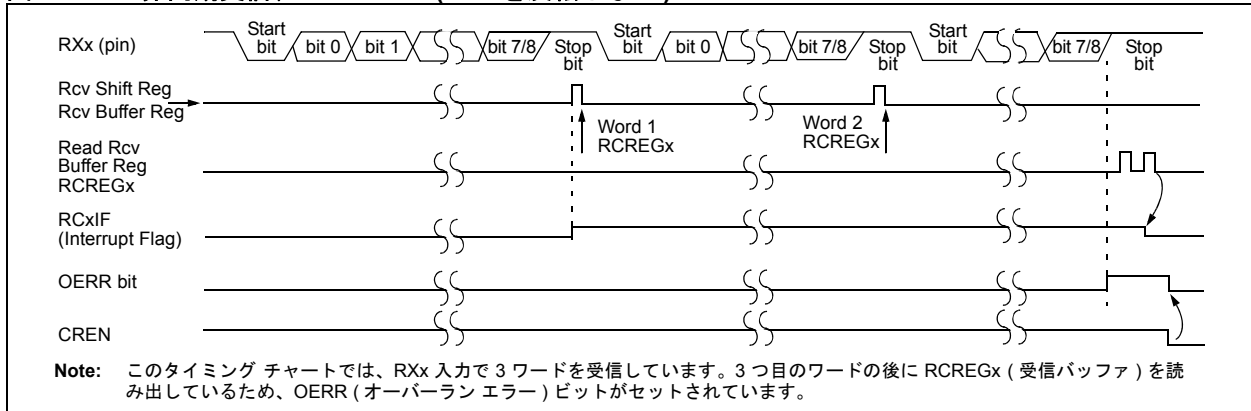


表 21-6: 非同期受信に関連するレジスタ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	リセット値の記載ページ
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	69
PIR1	PSPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	71
PIE1	PSPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	71
IPR1	PSPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	71
PIR3	SSP2IF	BCL2IF	RC2IF <sup>(1)</sup>	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	71
PIE3	SSP2IE	BCL2IE	RC2IE <sup>(1)</sup>	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	71
IPR3	SSP2IP	BCL2IP	RC2IP <sup>(1)</sup>	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	71
RCSTAx	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	71
RCREGx	EUSARTx 受信レジスタ								71
TXSTAx	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	71
BAUDCONx	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	-	WUE	ABDEN	72
SPBRGHx	EUSARTx baud レート ジェネレータ レジスタ上位バイト								72
SPBRGx	EUSARTx baud レート ジェネレータ レジスタ下位バイト								73

凡例: - = 未実装ビット、「0」として読み出されます。網掛けの部分は非同期受信では使いません。

**Note 1:** これらのビットは 80 ピン/100 ピンデバイスにのみ実装されています。それ以外のデバイスでは未実装で「0」として読み出されます。

# PIC18F97J60 ファミリ

## 21.2.4 同期ブレーク文字による自動復帰

スリープ中は、EUSARTx へのクロックは全て停止します。このため、baud レート ジェネレータは非アクティブで、バイトを適切に受信できません。EUSARTx の非同期モード動作中に自動復帰機能を利用すると、RXx/DTxラインのアクティビティによってコントローラを復帰できます。

自動復帰機能を有効にするには、WUE ビット (BAUDCONx<1>) をセットします。このビットをセットするとRXx/DTxでの通常の受信シーケンスは無効になり、EUSARTx はアイドル状態のまま、CPU のモードとは関係なく復帰イベントの発生を監視します。RXx/DTx ラインが High から Low に遷移すると復帰イベントです (これは、LIN/J2602 プロトコルの同期ブレークまたは復帰信号文字の開始に相当します)。

復帰イベントの発生後、モジュールは RCxIF 割り込みを生成します。この割り込みは、通常の動作モードの場合は Q クロックに同期して発生し (図 21-8)、デバイスがスリープの場合は非同期に発生します (図 21-9)。RCREGx レジスタを読み出すと、割り込み条件がクリアされます。

復帰イベント後に RXx ラインで Low から High への遷移が検出されると、WUE ビットは自動的にクリアされます。この時点で EUSARTx モジュールはアイドルであり、通常動作に戻ります。これにより、ユーザは同期ブレークイベントの終了を知る事ができます。

### 21.2.4.1 自動復帰使用時の注意事項

自動復帰は RXx/DTx の立ち上がりエッジ遷移を検出する事で動作するため、ストップビットよりも前の何らかの状態変化の情報が EOC (End-of-Character) と誤認識され、データまたはフレーミングのエラーにつながる可能性があります。従って、適切に動作させるには、最初を送信する文字が全て「0」である必要があります。これは、標準 RS-232 デバイスでは 00h (8 バイト)、LIN/J2602 バスでは 000h (12 ビット) です。

特に起動時間の長いオシレータを使っているアプリケーション (HS または HSPLL モード) では、オシレータの起動時間を考慮する必要があります。十分な長さの同期ブレーク文字 (または復帰信号) を送信したら、選択したオシレータが起動するまで十分な間隔を置いてから EUSARTx を適切に初期化するようにしてください。

### 21.2.4.2 WUE ビット使用時の注意事項

受信データの有効性を判断する際に、WUE ビットと RCxIF ビットのイベント発生タイミングが、若干の混乱を生む場合があります。前述の通り、WUE ビットをセットすると EUSARTx はアイドルに移行します。復帰イベントが発生すると、RCxIF ビットのセットによって受信割り込みが発生します。その後、WUE ビットは、RXx/DTx の立ち上がりエッジが検出された時点でクリアされます。続いて RCREGx レジスタを読み出す事で割り込み条件がクリアされます。通常、RCREGx のデータはダミーデータであり破棄します。

WUE ビットがクリアされた事 (またはセットされたままである事) と RCxIF フラグがセットされている事を、RCREGx のデータの完全性を保証する指標として使う事はできません。受信データの完全性を検証するには、ファームウェア内に並列手法を実装する事を検討してください。

データが失われないようにするには、RCIDL ビットをチェックして、受信動作中でない事を確認します。受信動作中でなければ、スリープに移行する直前に WUE ビットをセットします。

図 21-8: 通常動作時の自動復帰ビット (WUE) のタイミング

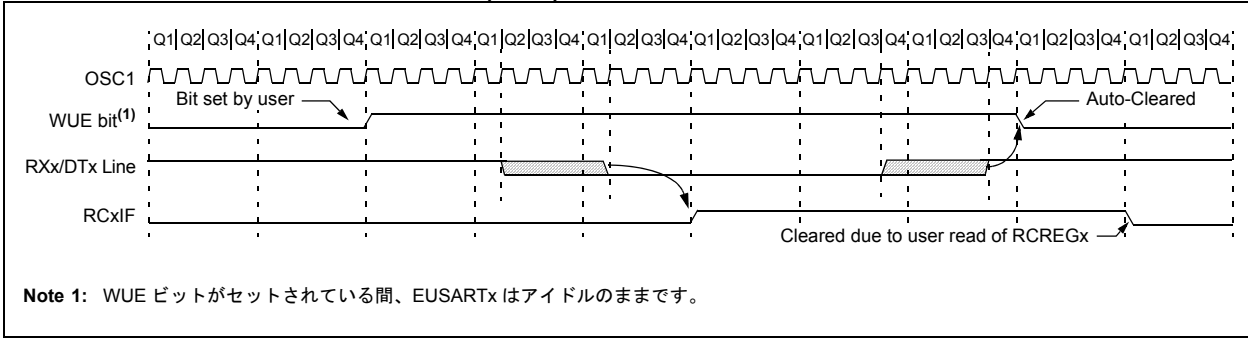
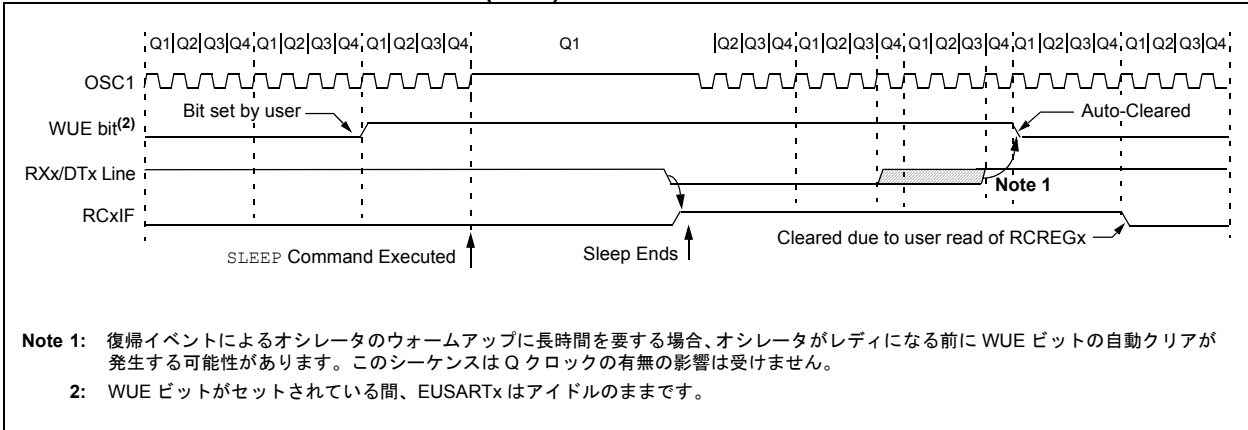


図 21-9: スリープ時の自動復帰ビット (WUE) のタイミング



# PIC18F97J60 ファミリ

## 21.2.5 ブレーク文字のシーケンス

EUSARTx モジュールには、LIN/J2602 バス規格で必須とされる特殊なブレーク文字シーケンスを送信する機能があります。ブレーク文字送信は、スタートビット、12 ビットの「0」、ストップビットの順で構成されます。フレームブレーク文字は、SENDB および TXEN ビット (TXSTAx<3>と TXSTAx<5>) をセットした状態で送信シフトレジスタ (TSR) にデータを読み込むと送信されます。TXREGx に書き込んだデータの値は無視され、全て「0」で送信される事に注意してください。

対応するストップビットを送信したら、SENDB ビットはハードウェアによって自動的にリセットされます。このため、ブレーク文字の後に送信するバイト (LIN/J2602 対応仕様の場合は主に同期文字) を送信 FIFO にあらかじめ読み込んでおく事が可能です。

ブレーク文字用に TXREGx に書き込まれたデータ値は無視される事に注意してください。この書き込みは、正しいシーケンスを開始する事だけを目的とします。

TRMT ビットは、通常の送信時と同様に、送信動作がアクティブかアイドルかを示します。ブレーク文字シーケンスのタイミングは、[図 21-10](#) を参照してください。

### 21.2.5.1 ブレークおよび同期送信シーケンス

下記のシーケンスは、ブレークとこれに続く自動 baud レート検出同期バイトで構成されたメッセージ フレーム ヘッダを送信します。このシーケンスは、LIN/J2602 バスマスタで代表的なものです。

1. EUSARTx を必要なモードに設定する。
2. TXEN および SENDB ビットをセットしてブレーク文字を設定する。
3. TXREGx にダミー文字を読み込んで送信を開始する (読み込んだ値は無視される)。
4. TXREGx に「55h」を書き込んで送信 FIFO バッファに同期文字を読み込む。
5. ブレーク送信後に、ハードウェアが SENDB ビットをリセットする。この時点で、あらかじめ設定されたモードにより同期文字が送信される。

TXREGx がエンプティになった事が TXxIF によって示された時点で、次のデータバイトを TXREGx に書き込む事ができます。

## 21.2.6 ブレーク文字の受信

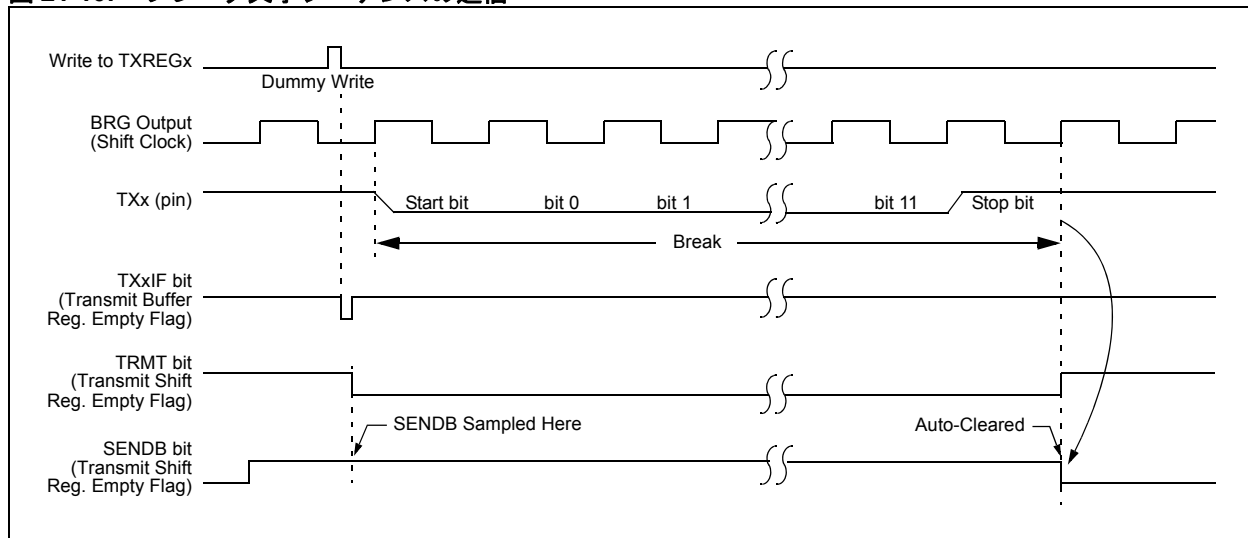
EUSARTx モジュールには、ブレーク文字の受信方法が 2 通りあります。

1 つ目は、baud レートを強制的に通常スピードの 9/13 の周波数に設定する方法です。これによって、ストップビット遷移を適切なサンプリング位置で発生させる事ができます (通常データの場合、ブレークの 13 ビットに対してスタートビットと 8 ビットのデータ)。

2 つ目は、[セクション 21.2.4 「同期ブレーク文字による自動復帰」](#) で説明した自動復帰機能を使う方法です。自動復帰機能を有効にすると、EUSARTx は RXx/DTx の次の 2 つの遷移をサンプリングし、RCxIF 割り込みを生成して、次のデータバイトを受信後、再度割り込みを生成します。

通常はブレーク文字の後に自動 baud レート検出機能を有効にします。どちらの方法も、TXxIF 割り込みが検出されれば ABDEN ビットをセットできます。

図 21-10: ブレーク文字シーケンスの送信



## 21.3 EUSARTx 同期マスターモード

同期マスターモードには、CSRC ビット (TXSTAx<7>) をセットする事で移行します。このモードでは、データが半二重方式で送信されます (すなわち送信と受信が同時には発生しません)。データの送信中は、受信が禁止され、受信中は送信が禁止されます。同期モードには、SYNC ビット (TXSTAx<4>) をセットする事で移行します。さらに、Tx と Rx ピンをそれぞれ CKx (クロック) と DTx (データ) ラインに設定するために、イネーブルビット SPEN (RCSTAx<7>) をセットします。

クロック極性 (CKx) は、TXCKP ビット (BAUDCON<4>) で選択します。TXCKP ビットをセットすると CKx のアイドル状態が High に設定され、クリアするとアイドル状態が Low に設定されます。データ極性 (DTx) は、RXDTP ビット (BAUDCON<5>) で選択します。RXDTP ビットをセットすると DTx のアイドル状態が High に設定され、クリアするとアイドル状態が Low に設定されます。CKx がアイドル状態に戻ると DTx がサンプリングされます。この設定は、このモジュールと Microwire デバイスを使うためのものです。

### 21.3.1 EUSARTx 同期マスター送信

図 21-3 に、EUSARTx トランスミッタのブロック図を示します。トランスミッタの中核となるのが送信 (シリアル) シフトレジスタ (TSR) です。送信シフトレジスタには、読み書き送信バッファレジスタ TXREGx からデータを入力します。TXREGx レジスタにはソフトウェアでデータを書き込みます。前回読み込まれたデータの最終ビットが送信されるまで、TSR レジスタに新しいデータは転送されません。最終ビットが送信されると、TXREGx レジスタから TSR に新しいデータが転送されます (送信すべきデータが存在する場合)。

TXREGx レジスタは、TSR レジスタへのデータ転送が完了すると (1 Tcy で完了) エンプティになり、TXxIF フラグビットがセットされます。この割り込みは、割り込みイネーブルビット TXxIE をセットまたはクリア

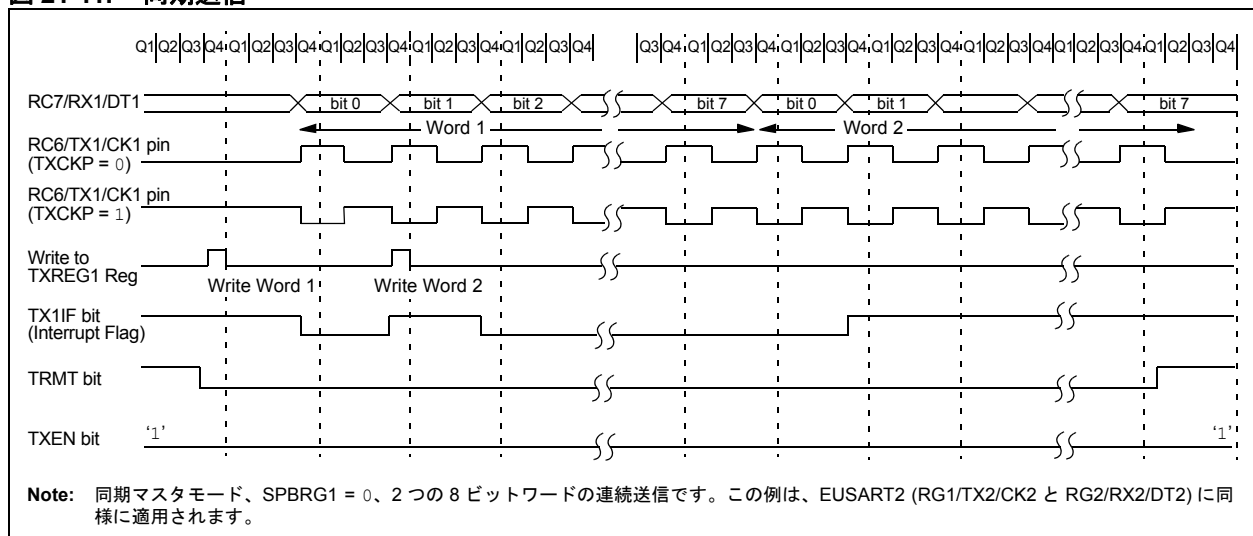
する事で有効または無効にできます。TXxIF はイネーブルビット TXxIE の状態に関係なくセットされ、ソフトウェアではクリアできません。ビットがリセットされるのは TXREGx レジスタに新しいデータが読み込まれた場合のみです。

フラグビットの TXxIF が TXREGx レジスタのステータスを示す一方、もう 1 つのビット TRMT (TXSTAx<1>) は TSR レジスタのステータスを示します。TRMT は、TSR がエンプティになるとセットされる読み出し専用のビットです。このビットには割り込みロジックが関連付けられていません。従って、TSR レジスタがエンプティであるかどうかを判断するには、このビットをポーリングする必要があります。TSR はデータメモリに割り当てられていないため、ユーザからはアクセスできません。

同期マスター送信を設定するには、以下の手順を実行します。

1. 適切な baud レートに合わせて、SPBRGHx:SPBRGx レジスタを初期化する。必要に応じて BRG16 ビットをセットまたはクリアして、目標 baud レートに設定する。
2. SYNC、SPEN、CSRC ビットをセットして同期マスターシリアルポートを有効にする。
3. 割り込みを使う場合、イネーブルビット TXxIE をセットする。
4. 9ビット送信を使う場合、TX9ビットをセットする。
5. TXEN ビットをセットして送信を有効にする。
6. 9ビット送信を選択した場合、TX9D ビットに 9ビット目を読み込む必要がある。
7. TXREGx レジスタにデータを読み込み、送信を開始する。
8. 割り込みを使う場合、INTCON レジスタの GIE および PEIE ビット (INTCON<7:6>) がセットされている事を確認する。

図 21-11: 同期送信



# PIC18F97J60 ファミリ

図 21-12: 同期送信 (TXEN を利用)

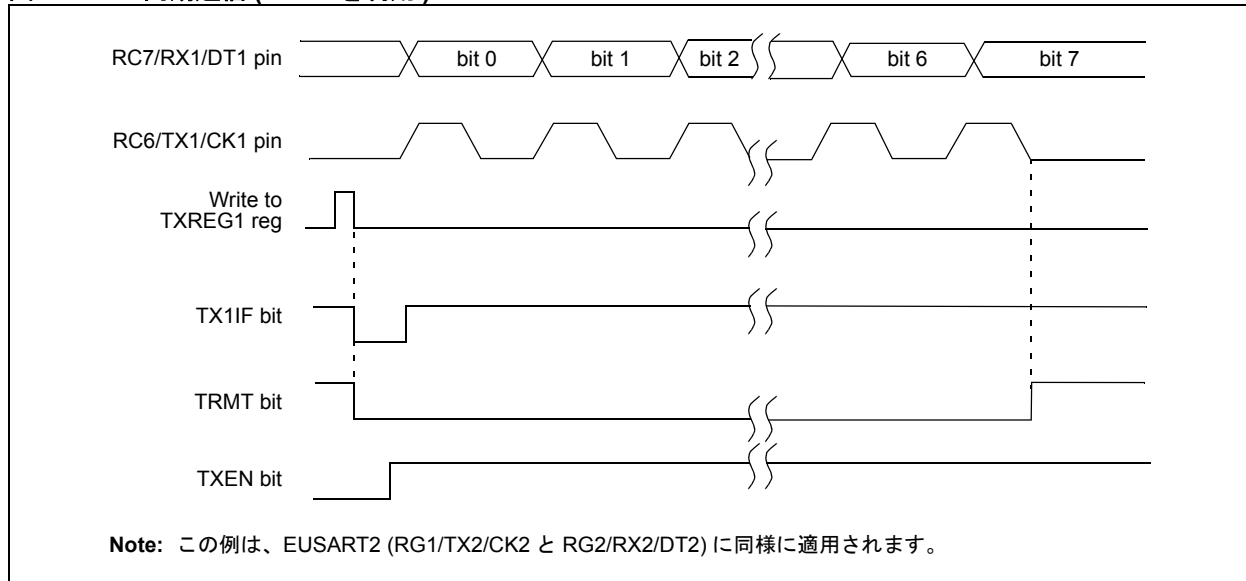


表 21-7: 同期マスタ送信に関連するレジスタ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	リセット値の記載ページ
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	69
PIR1	PSP1F	AD1F	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	71
PIE1	PSP1E	AD1E	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	71
IPR1	PSP1P	AD1P	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	71
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF <sup>(1)</sup>	TMR4IF	CCP5IF	CCP4IF	CCP3IF	71
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE <sup>(1)</sup>	TMR4IE	CCP5IE	CCP4IE	CCP3IE	71
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP <sup>(1)</sup>	TMR4IP	CCP5IP	CCP4IP	CCP3IP	71
RCSTAx	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	71
TXREGx	EUSARTx 送信レジスタ								71
TXSTAx	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	71
BAUDCONx	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	-	WUE	ABDEN	72
SPBRGHx	EUSARTx baud レート ジェネレータ レジスタ上位バイト								72
SPBRGx	EUSARTx baud レート ジェネレータ レジスタ下位バイト								73

凡例: -= 未実装ビット、「0」として読み出されます。網掛けの部分は同期マスタ送信では使いません。

Note 1: これらのビットは 80 ピン / 100 ピンデバイスにのみ実装されています。それ以外のデバイスでは未実装で「0」として読み出されます。

## 21.3.2 EUSARTx 同期マスタ受信

同期モードを選択した場合、シングル受信イネーブルビット SREN (RCSTAx<5>) または連続受信イネーブルビット CREN (RCSTAx<4>) のいずれかをセットする事で受信動作が有効になります。クロックの立ち下がりエッジで、RXx ピンからデータをサンプリングします。

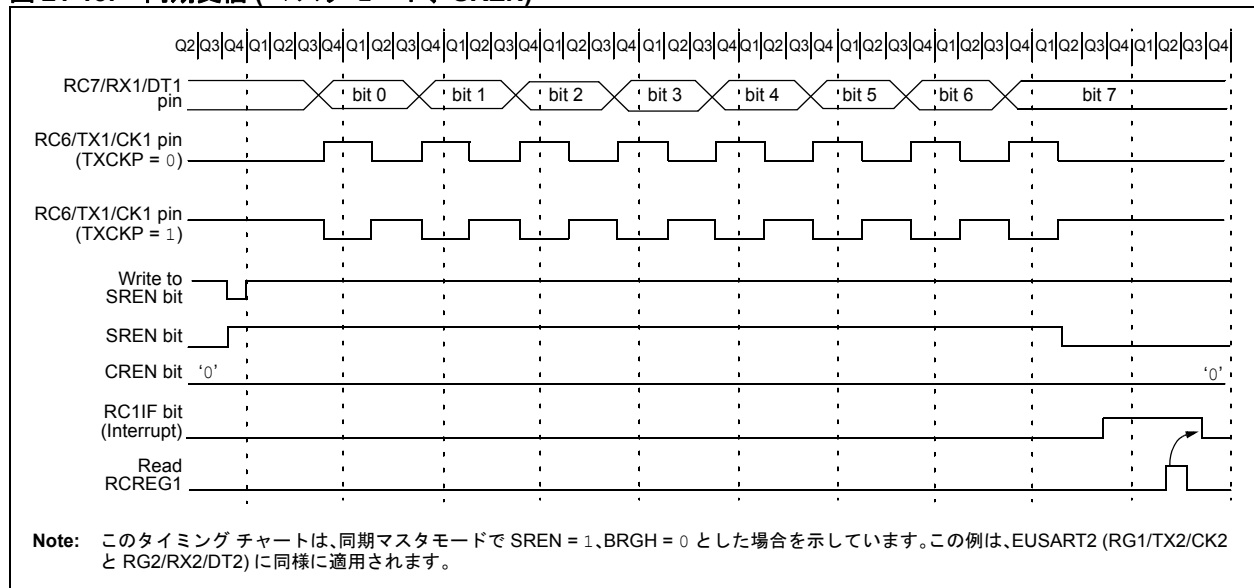
イネーブルビット SREN がセットされている場合、シングルワードだけを受信します。イネーブルビット CREN がセットされている場合、CREN がクリアされるまで連続して受信します。これらのビットが両方共セットされている場合、CREN が優先されます。

同期マスタ受信を設定するには、以下の手順を実行します。

1. 適切な baud レートに合わせて、SPBRGHx:SPBRGx レジスタを初期化する。必要に応じて BRG16 ビットをセットまたはクリアして、目標 baud レートに設定する。
2. SYNC、SPEN、CSRC ビットをセットして同期マスタ シリアルポートを有効にする。
3. CREN および SREN ビットがクリアされている事を確認する。

4. CKx ピンからの信号を反転する場合、TXCKP ビットをセットする。DTx ピンからの信号を反転する場合、RXDTP ビットをセットする。
5. 割り込みを使う場合、イネーブルビット RCxIE をセットする。
6. 9 ビット受信を使う場合、RX9 ビットをセットする。
7. シングル受信が必要な場合、SREN ビットをセットする。連続受信の場合、CREN ビットをセットする。
8. イネーブルビット RCxIE をセットした場合、受信が完了すると割り込みフラグビット RCxIF がセットされ、割り込みが生成される。
9. 9 ビット受信が有効な場合は RCSTAx レジスタを読み出して 9 ビット目を取得し、受信中に何らかのエラーが発生しなかったかどうかを判断する。
10. RCREGx レジスタを読み出して、8 ビットの受信データを読み出す。
11. 何らかのエラーが発生していた場合、CREN ビットをクリアしてエラーをクリアする。
12. 割り込みを使う場合、INTCON レジスタの GIE および PEIE ビット (INTCON<7:6>) がセットされている事を確認する。

図 21-13: 同期受信 (マスタモード、SREN)



# PIC18F97J60 ファミリ

表 21-8: 同期マスタ受信に関連するレジスタ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	リセット 値の記載 ページ
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	69
PIR1	PSPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	71
PIE1	PSPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	71
IPR1	PSPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	71
PIR3	SSP2IF	BCL2IF	RC2IF <sup>(1)</sup>	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	71
PIE3	SSP2IE	BCL2IE	RC2IE <sup>(1)</sup>	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	71
IPR3	SSP2IP	BCL2IP	RC2IP <sup>(1)</sup>	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	71
RCSTAx	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	71
RCREGx	EUSARTx 受信レジスタ								71
TXSTAx	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	71
BAUDCONx	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	-	WUE	ABDEN	72
SPBRGHx	EUSARTx baud レート ジェネレータ レジスタ上位バイト								72
SPBRGx	EUSARTx baud レート ジェネレータ レジスタ下位バイト								73

凡例: - = 未実装ビット、「0」として読み出されます。網掛けの部分は同期マスタ受信では使いません。

Note 1: これらのビットは 80 ピン/100 ピンデバイスにのみ実装されています。それ以外のデバイスでは未実装で「0」として読み出されます。



## 21.4 EUSARTx 同期スレーブモード

同期スレーブモードには、CSRC (TXSTAx<7>) をクリアする事で移行します。このモードは、シフトクロックが CKx ピンを介して外部から供給される点が、同期マスタモードとは異なります ( マスタモードではチップ内部で供給されます )。これにより、デバイスが低消費電力モードで動作中でもデータを転送または受信できます。

### 21.4.1 EUSARTx 同期スレーブ送信

同期マスタモードと同期スレーブモードの動作は、スリープの場合を除いて同じです。

TXREGx に 2 ワードを書き込んだ後に SLEEP 命令を実行すると、以下の動作が発生します。

- 最初のワードがただちに TSR レジスタに転送され、送信される。
- 2 番目のワードは TXREGx レジスタに残る。
- フラグビット TXxIF はセットされない。
- 最初のワードが TSR からシフト出力されると、TXREGx レジスタから TSR に 2 番目のワードが転送され、この時点でフラグビット TXxIF がセットされる。
- イネーブルビット TXxIE がセットされている場合、割り込みがチップをスリープから復帰させる。グローバル割り込みが有効の場合、プログラムは割り込みベクタに分岐する。

同期スレーブ送信を設定するには、以下の手順を実行します。

- SYNC および SPEN ビットをセット、CSRC ビットをクリアして、同期スレーブシリアルポートを有効にする。
- CREN および SREN ビットをクリアする。
- CKx ピンからの信号を反転する場合、TXCKP ビットをセットする。DTx ピンからの信号を反転する場合、RXDTP ビットをセットする。
- 割り込みを使う場合、イネーブルビット TXxIE をセットする。
- 9 ビット送信を使う場合、TX9 ビットをセットする。
- イネーブルビット TXEN をセットして、送信を有効にする。
- 9 ビット送信を選択した場合、TX9D ビットに 9 ビット目を読み込む必要がある。
- TXREGx レジスタにデータを読み込み、送信を開始する。
- 割り込みを使う場合、INTCON レジスタの GIE および PEIE ビット (INTCON<7:6>) がセットされている事を確認する。

表 21-9: 同期スレーブ送信に関連するレジスタ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	リセット値の記載ページ
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	69
PIR1	PSPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	71
PIE1	PSPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	71
IPR1	PSPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	71
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF <sup>(1)</sup>	TMR4IF	CCP5IF	CCP4IF	CCP3IF	71
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE <sup>(1)</sup>	TMR4IE	CCP5IE	CCP4IE	CCP3IE	71
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP <sup>(1)</sup>	TMR4IP	CCP5IP	CCP4IP	CCP3IP	71
RCSTAx	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	71
TXREGx	EUSARTx 送信レジスタ								71
TXSTAx	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	71
BAUDCONx	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	-	WUE	ABDEN	72
SPBRGHx	EUSARTx baud レート ジェネレータ レジスタ上位バイト								72
SPBRGx	EUSARTx baud レート ジェネレータ レジスタ下位バイト								73

凡例: - = 未実装ビット、「0」として読み出されます。網掛けの部分は同期スレーブ送信では使いません。

Note 1: これらのビットは 80 ピン/100 ピンデバイスにのみ実装されています。それ以外のデバイスでは未実装で「0」として読み出されます。

# PIC18F97J60 ファミリ

## 21.4.2 EUSARTx 同期スレーブ受信

同期マスタモードと同期スレーブモードの動作は同じです。例外は、スリープまたはいずれかのアイドルにおける動作と、スレーブモードでは SREN ビットがドントケアである点です。

スリープまたはいずれかのアイドルに移行する前に CREN ビットをセットして受信を有効にした場合、移行した低消費電力モードによる動作中に 1 ワードを受信できます。ワードを受信すると、RSR レジスタはデータを RCREGx レジスタに転送します。RCxIE イネーブルビットがセットされている場合、生成された割り込みがチップを低消費電力モードから復帰させます。グローバル割り込みが有効の場合、プログラムは割り込みベクタに分岐します。

同期スレーブ受信を設定するには、以下の手順を実行します。

1. SYNC および SPEN ビットをセット、CSRC ビットをクリアして同期マスタ シリアルポートを有効にする。
2. 割り込みを使う場合、イネーブルビット RCxIE をセットする。
3. CKx ピンからの信号を反転する場合、TXCKP ビットをセットする。DTx ピンからの信号を反転する場合、RXDTP ビットをセットする。
4. 9 ビット受信を使う場合、RX9 ビットをセットする。
5. 受信を有効にするためにイネーブルビット CREN をセットする。
6. 受信が完了すると、フラグビット RCxIF ビットがセットされる。イネーブルビット RCxIE をセットしている場合、割り込みが生成される。
7. 9 ビット受信が有効な場合は RCSTAx レジスタを読み出して 9 ビット目を取得し、受信中に何らかのエラーが発生しなかったかどうかを判断する。
8. RCREGx レジスタを読み出して、8 ビットの受信データを読み出す。
9. 何らかのエラーが発生していた場合、CREN ビットをクリアしてエラーをクリアする。
10. 割り込みを使う場合、INTCON レジスタの GIE および PEIE ビット (INTCON<7:6>) がセットされている事を確認する。

表 21-10: 同期スレーブ受信に関連するレジスタ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	リセット値の記載ページ
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	69
PIR1	PSPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	71
PIE1	PSPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	71
IPR1	PSPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	71
PIR3	SSP2IF	BCL2IF	RC2IF <sup>(1)</sup>	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	71
PIE3	SSP2IE	BCL2IE	RC2IE <sup>(1)</sup>	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	71
IPR3	SSP2IP	BCL2IP	RC2IP <sup>(1)</sup>	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	71
RCSTAx	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	71
RCREGx	EUSARTx 受信レジスタ								71
TXSTAx	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	71
BAUDCONx	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	-	WUE	ABDEN	72
SPBRGHx	EUSARTx baud レート ジェネレータ レジスタ上位バイト								72
SPBRGx	EUSARTx baud レート ジェネレータ レジスタ下位バイト								73

凡例: - = 未実装ビット、「0」として読み出されます。網掛けの部分は同期スレーブ受信では使いません。

Note 1: これらのビットは 80 ピン/100 ピンデバイスにのみ実装されています。それ以外のデバイスでは未実装で「0」として読み出されます。

## 22.0 10 ビット A/D コンバータ (ADC) モジュール

ADC (Analog-to-Digital Converter) モジュールの入力は、64 ピンデバイスに 11 個、80 ピンデバイスに 15 個、100 ピンデバイスに 16 個あります。このモジュールを使うと、アナログ入力信号を 10 ビットのデジタル値に変換できます。

このモジュールは以下の 5 つのレジスタを使います。

- A/D 変換結果レジスタ上位バイト (ADRESH)
- A/D 変換結果レジスタ下位バイト (ADRESL)
- A/D 制御レジスタ 0 (ADCON0)
- A/D 制御レジスタ 1 (ADCON1)
- A/D 制御レジスタ 2 (ADCON2)

**レジスタ 22-1** に示す ADCON0 レジスタは、A/D モジュールの動作を制御します。**レジスタ 22-2** に示す ADCON1 レジスタは、ポートピンの機能を設定します。**レジスタ 22-3** に示す ADCON2 レジスタは、A/D コンバータのクロック源、プログラムされたアキュイジション時間と位置調整を設定します。

### レジスタ 22-1: ADCON0: A/D 制御レジスタ 0

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADCAL	—	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

- bit 7      **ADCAL:** A/D 校正ビット  
 1 = 次の A/D 変換で校正を実行する  
 0 = 通常の A/D 変換を実行する (校正は実行しない)
- bit 6      **未実装:** 「0」として読み出し
- bit 5-2      **CHS<3:0>:** アナログ チャンネル選択ビット  
 0000 = チャンネル 0 (AN0)  
 0001 = チャンネル 1 (AN1)  
 0010 = チャンネル 2 (AN2)  
 0011 = チャンネル 3 (AN3)  
 0100 = チャンネル 4 (AN4)  
 0101 = チャンネル 5 (AN5)<sup>(1,3)</sup>  
 0110 = チャンネル 6 (AN6)  
 0111 = チャンネル 7 (AN7)  
 1000 = チャンネル 8 (AN8)  
 1001 = チャンネル 9 (AN9)  
 1010 = チャンネル 10 (AN10)  
 1011 = チャンネル 11 (AN11)  
 1100 = チャンネル 12 (AN12)<sup>(2,3)</sup>  
 1101 = チャンネル 13 (AN13)<sup>(2,3)</sup>  
 1110 = チャンネル 14 (AN14)<sup>(2,3)</sup>  
 1111 = チャンネル 15 (AN15)<sup>(2,3)</sup>
- bit 1      **GO/DONE:** A/D 変換ステータスビット  
ADON = 1 の場合:  
 1 = A/D 変換が進行中である  
 0 = A/D 変換はアイドルである
- bit 0      **ADON:** A/D ON ビット  
 1 = A/D コンバータ モジュールを有効にする  
 0 = A/D コンバータ モジュールを無効にする

- Note 1:** このチャンネルは 100 ピンデバイスにのみ実装されています。  
**2:** これらのチャンネルは 80 ピン / 100 ピンデバイスにのみ実装されています。  
**3:** 未実装のチャンネルに対して変換を実行すると、ランダムな値が返されます。

# PIC18F97J60 ファミリ

## レジスタ 22-2: ADCON1: A/D 制御レジスタ 1

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

bit 7-6      **未実装:** 「0」として読み出し

bit 5      **VCFG1:** 参照電圧コンフィグレーション ビット (VREF- 電源)

1 = VREF- (AN2)

0 = AVSS

bit 4      **VCFG0:** 参照電圧コンフィグレーション ビット (VREF+ 電源)

1 = VREF+ (AN3)

0 = AVDD

bit 3-0      **PCFG<3:0>:** A/D ポート コンフィグレーション制御ビット:

PCFG<3:0>	AN15 <sup>(1)</sup>	AN14 <sup>(1)</sup>	AN13 <sup>(1)</sup>	AN12 <sup>(1)</sup>	AN11	AN10	AN9	AN8	AN7	AN6	AN5 <sup>(2)</sup>	AN4	AN3	AN2	AN1 <sup>(3)</sup>	AN0 <sup>(3)</sup>
0000	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
0001	D	D	A	A	A	A	A	A	A	A	A	A	A	A	A	A
0010	D	D	D	A	A	A	A	A	A	A	A	A	A	A	A	A
0011	D	D	D	D	A	A	A	A	A	A	A	A	A	A	A	A
0100	D	D	D	D	D	A	A	A	A	A	A	A	A	A	A	A
0101	D	D	D	D	D	D	A	A	A	A	A	A	A	A	A	A
0110	D	D	D	D	D	D	D	A	A	A	A	A	A	A	A	A
0111	D	D	D	D	D	D	D	D	A	A	A	A	A	A	A	A
1000	D	D	D	D	D	D	D	D	D	A	A	A	A	A	A	A
1001	D	D	D	D	D	D	D	D	D	D	A	A	A	A	A	A
1010	D	D	D	D	D	D	D	D	D	D	D	A	A	A	A	A
1011	D	D	D	D	D	D	D	D	D	D	D	D	A	A	A	A
1100	D	D	D	D	D	D	D	D	D	D	D	D	D	A	A	A
1101	D	D	D	D	D	D	D	D	D	D	D	D	D	D	A	A
1110	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	A
1111	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D

A = アナログ入力

D = デジタル I/O

**Note 1:** AN12 ~ AN15 は 80 ピン /100 ピンデバイスにのみ実装されています。

**2:** AN5 は 100 ピンデバイスにのみ実装されています。

**3:** AN0 と AN1 は、アナログまたはデジタル I/O モードで Ethernet LED 出力としても動作させる事ができます。

# PIC18F97J60 ファミリ

## レジスタ 22-3: ADCON2: A/D 制御レジスタ 2

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

bit 7      **ADFM:** A/D 変換結果フォーマット選択ビット

1 = 右詰め  
0 = 左詰め

bit 6      **未実装:** 「0」として読み出し

bit 5-3      **ACQT<2:0>:** A/D アクイジション時間選択ビット:

111 = 20 TAD  
110 = 16 TAD  
101 = 12 TAD  
100 = 8 TAD  
011 = 6 TAD  
010 = 4 TAD  
001 = 2 TAD  
000 = 0 TAD<sup>(1)</sup>

bit 2-0      **ADCS<2:0>:** A/D 変換クロック選択ビット

111 = FRC (A/D コンバータの RC オシレータから供給されたクロック)<sup>(1)</sup>  
110 = FOSC/64  
101 = FOSC/16  
100 = FOSC/4  
011 = FRC (A/D コンバータの RC オシレータから供給されたクロック)<sup>(1)</sup>  
010 = FOSC/32  
001 = FOSC/8  
000 = FOSC/2

**Note 1:** A/D コンバータの FRC クロック源を選択した場合、A/D クロックの起動前に 1 Tcy (命令サイクル) の遅延が発生します。これにより、変換開始前に SLEEP 命令を実行できます。

# PIC18F97J60 ファミリ

アナログ参照電圧は、デバイスの正負電源 (AVDD と AVSS) または RA3/AN3/VREF+ と RA2/AN2/VREF- ピン間の電圧レベルのいずれかをソフトウェアによって選択できます。

この A/D コンバータは、デバイスがスリープ中でも動作が可能です。スリープ中に A/D モジュールを動作させるには、A/D 変換クロックを A/D コンバータの内部 RC オシレータから供給する必要があります。

サンプルホールドの出力をコンバータに入力し、コンバータは逐次比較型の処理によって結果を生成します。

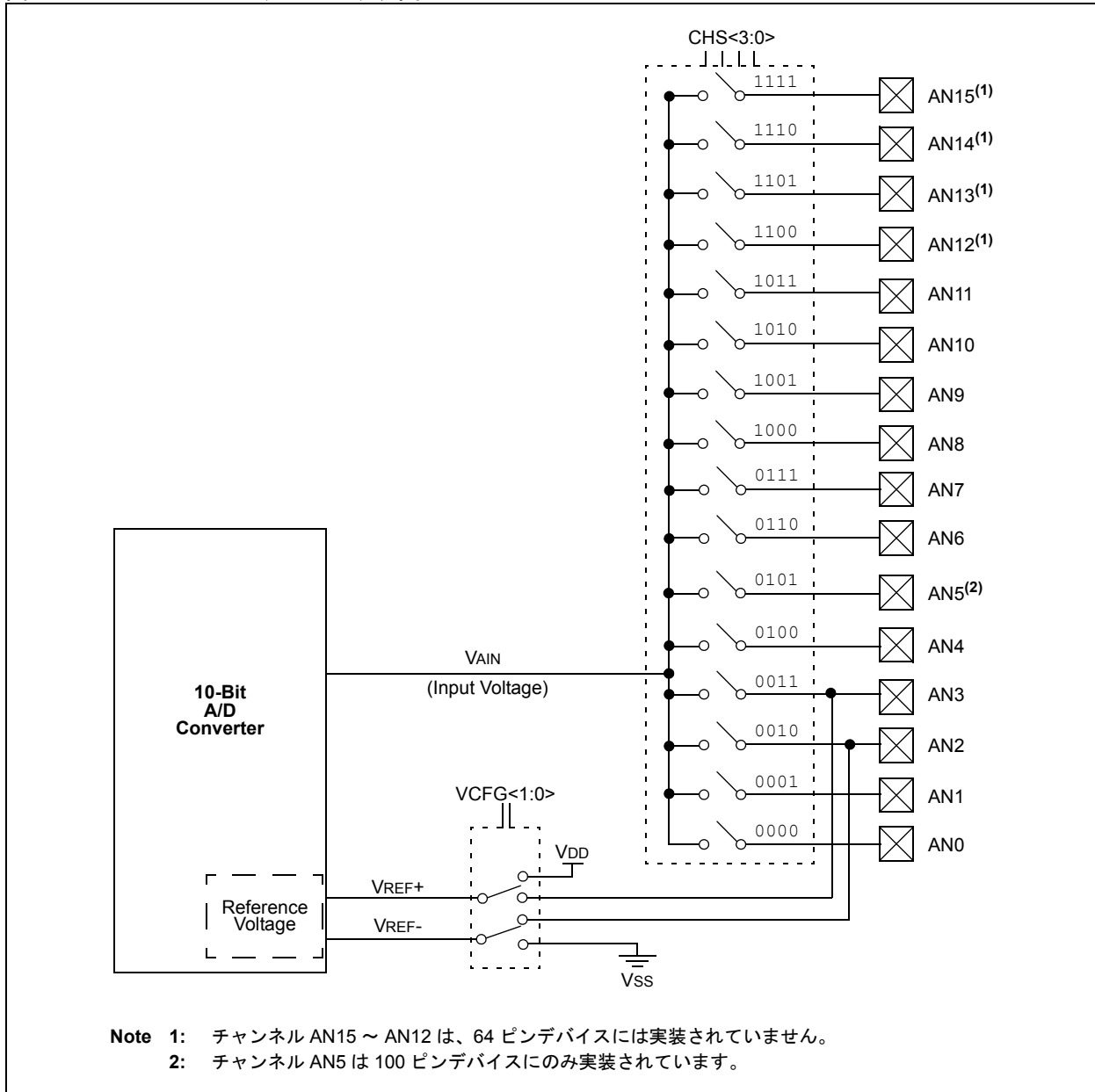
A/D コンバータに関連する各ポートピンはアナログ入力またはデジタル I/O として設定できます。A/D 変換の結果は、ADRESH および ADRESL レジスタに格納し

ます。A/D 変換が完了すると、結果が ADRESH:ADRESL レジスタペアに書き込まれ、GO/DONE ビット (ADCON0<1>) がクリアされ、A/D 割り込みフラグビット ADIF がセットされます。

デバイスをリセットすると、全てのレジスタがリセット状態に戻ります。これにより A/D モジュールは停止し、実行中の変換は全て中止されます。ADRESH:ADRESL レジスタペアの値はパワーオン リセットでは変更されません。パワーオン リセット後にこれらのレジスタに格納されているデータは未知です。

図 22-1 に A/D モジュールのブロック図を示します。

図 22-1: A/D コンバータのブロック図



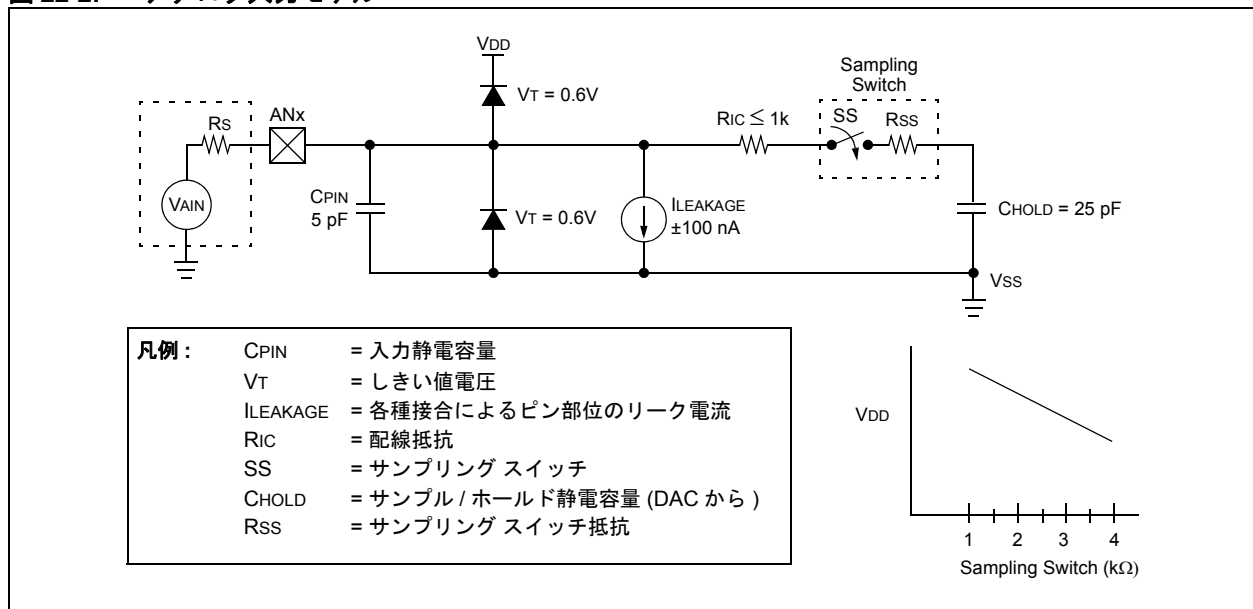
A/D モジュールに対する必要な設定が完了したら、変換開始前に選択したチャンネルのアクイジションを実行する必要があります。アナログ入力チャンネルに対応する TRIS ビットを入力に設定する必要があります。アクイジション時間の決め方は、[セクション 22.1「A/D アクイジションの要件」](#)を参照してください。このアクイジション時間経過後、A/D 変換を開始できます。アクイジション時間は、GO/DONE ビットのセットから変換開始までの間に実行するようにプログラムできます。

A/D 変換を実行する手順は以下の通りです。

1. A/D モジュールを設定する。
  - アナログピン、参照電圧、デジタル I/O を設定する (ADCON1)。
  - A/D 入力チャンネルを選択する (ADCON0)。
  - A/D アクイジション時間を選択する (ADCON2)。
  - A/D 変換クロックを選択する (ADCON2)。
  - A/D モジュールを有効にする (ADCON0)。
2. A/D 割り込みを設定する (必要な場合)。

- ADIF ビットをクリアする。
  - ADIE ビットをセットする。
  - GIE ビットをセットする。
3. 必要なアクイジション時間が経過するまで待つ (必要な場合)。
  4. 変換を開始する。
    - GO/DONE ビット (ADCON0<1>) をセットする。
  5. 以下のいずれかの方法で A/D 変換の完了を待つ。
    - GO/DONE ビットをポーリングしてクリアされている事を確認する。
 または
    - A/D 割り込みを待つ。
  6. A/D 変換結果レジスタ (ADRESH:ADRESL) を読み出し、必要に応じて ADIF ビットをクリアする。
  7. 次の変換を実行するために、必要に応じて手順 1 または手順 2 に戻る。ビットあたりの A/D 変換時間を TAD と定義する。次のアクイジションを開始する前に、2 TAD 以上待機する必要がある。

図 22-2: アナログ入力モデル



# PIC18F97J60 ファミリ

## 22.1 A/D アクイジションの要件

A/D コンバータが仕様上の精度を実現するには、電荷保持コンデンサ (CHOLD) を入力チャンネルの電圧レベルまで完全に充電する必要があります。図 22-2 に、アナログ入力モデルを示します。ソース インピーダンス ( $R_s$ ) と内部サンプリング スイッチのインピーダンス ( $R_{ss}$ ) は、コンデンサ (CHOLD) の充電に必要な時間に直接影響します。サンプリング スイッチのインピーダンス ( $R_{ss}$ ) はデバイス電圧 (VDD) によって変化します。ソース インピーダンスは、アナログ入力のオフセット電圧に影響します (ピンリーク電流による)。アナログソースの最大推奨インピーダンスは、2.5 k $\Omega$  です。アナログ入力チャンネルを選択または変更した後は、変換を開始する前に、最小アクイジション時間以上の期間、そのチャンネルをサンプルする必要があります。

**Note:** 変換が始まると、保持コンデンサは入力ピンから切り離されます。

最小アクイジション時間の計算には式 22-1が使えます。この式は、1/2 LSB 誤差の適用を前提としています (A/D 変換は 1024 ステップ)。1/2 LSB の誤差は、仕様上の分解能を A/D 変換で実現できる最大許容誤差です。

式 22-3 に、必要な最小アクイジション時間 TACQ の計算を示します。この計算は、以下の条件のアプリケーション システムを前提としています。

CHOLD	=	25 pF
$R_s$	=	2.5 k $\Omega$
変換誤差	$\leq$	1/2 LSB
VDD	=	3 V $\rightarrow$ $R_{ss} = 2$ k $\Omega$
温度	=	85 $^{\circ}$ C (システム最大値)

### 式 22-1: アクイジション時間

$$\begin{aligned} \text{TACQ} &= \text{アンプのセトリングタイム} + \text{保持コンデンサの充電時間} + \text{温度係数} \\ &= \text{TAMP} + \text{Tc} + \text{Tcoff} \end{aligned}$$

### 式 22-2: 最小 A/D 充電時間

$$\begin{aligned} \text{V}_{\text{HOLD}} &= (\text{V}_{\text{REF}} - (\text{V}_{\text{REF}}/2048)) \cdot (1 - e^{-(\text{Tc}/\text{CHOLD})(\text{RIC} + \text{RSS} + \text{RS})}) \\ \text{または} \\ \text{Tc} &= -(\text{CHOLD})(\text{RIC} + \text{RSS} + \text{RS}) \ln(1/2048) \end{aligned}$$

### 式 22-3: 必要な最小アクイジション時間の計算

$$\begin{aligned} \text{TACQ} &= \text{TAMP} + \text{Tc} + \text{Tcoff} \\ \text{TAMP} &= 0.2 \mu\text{s} \\ \text{Tcoff} &= (\text{温度} - 25 \text{ }^{\circ}\text{C})(0.02 \mu\text{s}/^{\circ}\text{C}) \\ &\quad (85 \text{ }^{\circ}\text{C} - 25 \text{ }^{\circ}\text{C})(0.02 \mu\text{s}/^{\circ}\text{C}) \\ &\quad 1.2 \mu\text{s} \end{aligned}$$

温度係数が必要となるのは、温度が >25  $^{\circ}$ C の場合のみです。25  $^{\circ}$ C 未満の場合、Tcoff = 0 ms です。

$$\begin{aligned} \text{Tc} &= -(\text{CHOLD})(\text{RIC} + \text{RSS} + \text{RS}) \ln(1/2048) \mu\text{s} \\ &\quad -(25 \text{ pF})(1 \text{ k}\Omega + 2 \text{ k}\Omega + 2.5 \text{ k}\Omega) \ln(0.0004883) \mu\text{s} \\ &\quad 1.05 \mu\text{s} \\ \text{TACQ} &= 0.2 \mu\text{s} + 1 \mu\text{s} + 1.2 \mu\text{s} \\ &\quad 2.4 \mu\text{s} \end{aligned}$$



## 22.2 自動アキュイジション時間の選択と設定

GO/DONE ビットがセットされるたびに発生するアキュイジション時間は、ADCON2 レジスタを使ってユーザが設定できます。

GO/DONE ビットをセットすると、サンプリングが停止し、変換が開始します。入力チャンネルを選択してから GO/DONE ビットをセットするまでに、必要なアキュイジション時間が経過した事は、ユーザが確認する必要があります。この確認動作は ACQT<2:0> ビット (ADCON2<5:3>) がリセット状態のままの場合 (「000」) に必要となり、アキュイジション時間をプログラミングできないデバイスとの互換性が保たれます。

必要に応じて ACQT ビットをセットし、A/D モジュールにプログラマブルなアキュイジション時間を選択できます。GO/DONE ビットをセットすると、A/D モジュールは選択したアキュイジション時間入力のサンプリングを継続した後、自動的に変換を開始します。アキュイジション時間はプログラミング済みであるため、チャンネルを選択後アキュイジション時間の経過を待たずに GO/DONE ビットをセットできます。

いずれの場合も、変換が完了すると GO/DONE ビットがクリアされ、ADIF フラグがセットされ、A/D コンバータは現在選択されているチャンネルのサンプリングを再開します。アキュイジション時間をプログラミングした場合、アキュイジション時間が終了したかどうか、変換が開始したかどうかを知る方法はありません。

## 22.3 A/D 変換クロックの選択

ビットあたりの A/D 変換時間を TAD と定義します。10 ビット変換では、A/D 変換に 11 TAD を要します。A/D 変換のクロック源はソフトウェアで選択できます。

TAD は以下の 7 種類から選択できます。

- 2 TOSC
- 4 TOSC
- 8 TOSC
- 16 TOSC
- 32 TOSC
- 64 TOSC
- 内部 RC オシレータ

正確な A/D 変換を実行するには、A/D 変換クロック (TAD) をできるだけ短く、ただし最小 TAD より長く設定する必要があります。詳細は、[セクション 28.0「電気的特性」](#)の表 28-27 の A/D パラメータ 130 を参照してください。

表 22-1 に、デバイス動作周波数と、選択した A/D クロック源の組み合わせから求めた TAD 時間の計算結果を示します。

表 22-1: TAD とデバイス動作周波数

AD クロック源 (TAD)		最大デバイス周波数
動作	ADCS<2:0>	
2 TOSC	000	2.86 MHz
4 TOSC	100	5.71 MHz
8 TOSC	001	11.43 MHz
16 TOSC	101	22.86 MHz
32 TOSC	010	41.67 MHz
64 TOSC	110	41.67 MHz
RC <sup>(2)</sup>	x11	1.00 MHz <sup>(1)</sup>

**Note 1:** RC クロック源の TAD 時間は 4 ms (typ.) です。

**2:** A/D モジュールの RC クロックの仕様は、[表 28-27](#) のパラメータ 130 を参照してください。

## 22.4 アナログポートピンの設定

ADCON1、TRISA、TRISF、TRISH レジスタは、A/D ポートピンの動作を制御します。ポートピンをアナログ入力として動作させる場合、対応する TRIS ビットをセット (入力として設定) する必要があります。TRIS ビットをクリア (出力として設定) すると、デジタル出力レベル (VOH または VOL) が変換されます。

A/D 変換動作は、CHS<3:0> ビットと TRIS ビットの状態とは無関係に実行されます。

**Note 1:** PORT レジスタを読み出すと、アナログ入力チャンネルとして設定されている全てのピンが、クリア状態 (Low レベル) として読み出されます。デジタル入力として設定しているピンは、アナログ入力を変換します。デジタル設定された入力のアナログレベルは正確に変換されます。

**2:** デジタル入力として定義したピンにアナログレベルを印加すると、デジタル入力バッファにデバイスの仕様限界値を超える電流が流れる場合があります。

# PIC18F97J60 ファミリ

## 22.5 A/D 変換

図 22-3に、GO/DONE ビットをセットして ACQT<2:0> ビットをクリアした後の A/D コンバータの動作を示します。スリープへの移行を可能にするため、変換は次の命令の後に開始します。

図 22-4に、GO/DONE ビットをセットして ACQT<2:0> ビットを「010」に設定し、変換開始前に 4 TAD のアキュイジション時間を選択した場合の A/D コンバータの動作を示します。

変換中に GO/DONE ビットをクリアすると、その変換は中止されます。一部しか完了していない A/D 変換サンプルによって A/D 変換結果レジスタペアが更新される事はありません。これは、ADRESH:ADRESL レジスタに前回の変換値（または ADRESH:ADRESL レジスタに最後に書き込まれた値）が引き続き保持される事を意味します。

A/D 変換が完了または中止されてから次のアキュイジションを開始するまでには、2 TAD の待ち時間が必要です。この待ち時間が経過すると、選択したチャンネルのアキュイジションが自動的に開始します。

**Note:** A/D コンバータを ON にする命令と同じ命令で GO/DONE ビットをセットしないでください。

## 22.6 ECCP2 トリガの使用法

A/D 変換は ECCP2 モジュールの「特殊イベントトリガ」によって開始する事もできます。それには、CCP2M<3:0> ビット (CCP2CON<3:0>) を「1011」にプログラムし、A/D モジュールを有効に (ADON ビットをセット) する必要があります。このトリガが発生すると、GO/DONE ビットがセットされて A/D アキュイジションと変換が開始し、Timer1 または Timer3 カウンタが 0 にリセットされます。Timer1 または Timer3 がリセットされ、最小限のソフトウェア オーバーヘッド (ADRESH:ADRESL の内容を所定のアドレスに移動) で A/D アキュイジション周期を自動的に繰り返します。特殊イベントトリガが GO/DONE ビットをセットする (変換を開始する) 前に、適切なアナログ入力チャンネルを選択し、最小アキュイジション時間をユーザがタイミング設定するか、適切な TACQ 時間を選択する必要があります。

A/D モジュール無効時 (ADON がクリア)、A/D モジュールは特殊イベントトリガを無視しますが、Timer1 (または Timer3) のカウンタはリセットします。

図 22-3: A/D 変換の TAD サイクル (ACQT<2:0> = 000、TACQ = 0)

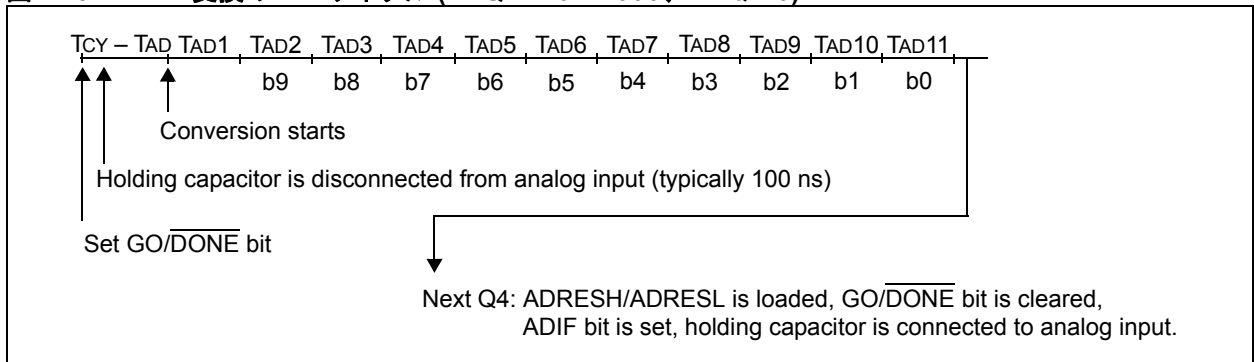
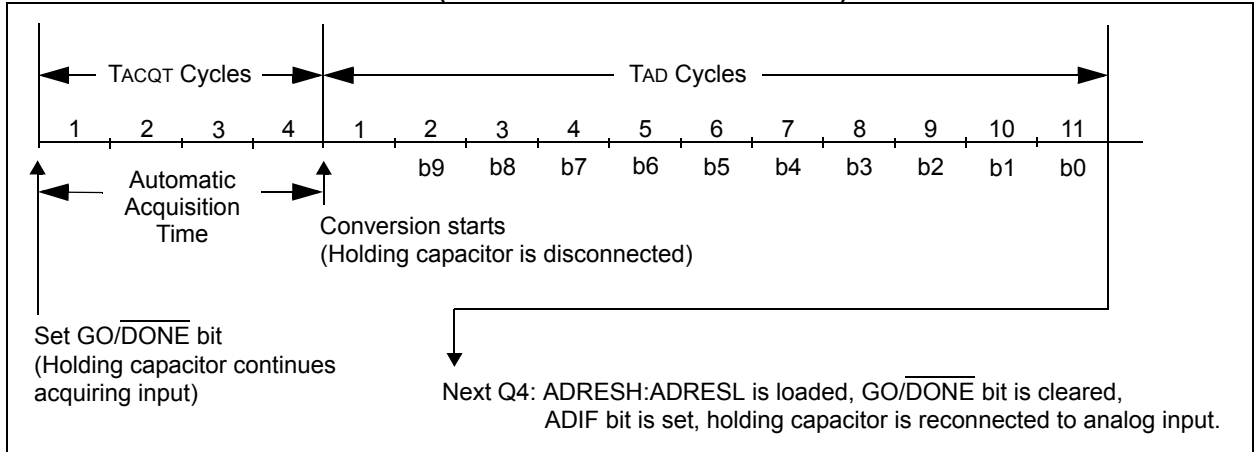


図 22-4: A/D 変換の TAD サイクル (ACQT<2:0> = 010、TACQ = 4 TAD)



## 22.7 A/D コンバータの校正

PIC18F97J60 ファミリの A/D コンバータは、モジュール内で発生するあらゆるオフセットを補償する自己校正機能を備えています。校正のプロセスは自動化されており、ADCAL ビット (ADCON0<7>) をセットする事で開始します。次に GO/DONE ビットがセットされると、モジュールは「ダミー」の変換 (入力チャンネルのいずれも読み出さない) を実行し、オフセットを補償するために結果値を内部で格納します。これにより、それ以後のオフセットが補償されます。

校正プロセスは、デバイスが比較的安定した動作条件にある事を前提としています。A/D 校正を使う場合、各デバイスリセット後、あるいは動作条件に大きな変化が生じた後で校正を実行します。

## 22.8 電力管理モードにおける動作

自動アキュイジション時間と A/D 変換クロックを選択する際は、電力管理モードで使われるクロック源と周波数も考慮に入れる必要があります。

デバイスが電力管理モードの時に A/D コンバータを動作させる場合、そのモードで使うクロック源に合わせて ADCON2 レジスタの ACQT<2:0> および ADCS<2:0> ビットを更新する必要があります。電力管理モードに移行後 (電力管理ランモードのいずれか)、A/D アキュイジションまたは A/D 変換を開始できます。アキュイジションまたは変換を一旦開始したら、これらの動作が完了するまで、デバイスには継続して同じ電力管理モードのクロック源からクロックを供給する必要があります。必要なら、変換中にデバイスを対応する電力管理アイドルに移行する事もできます。

電力管理モードのクロック周波数が 1 MHz 未満の場合、A/D コンバータのクロック源には RC を選択します。

スリープ中の動作には、A/D コンバータの RC クロックを選択する必要があります。ACQT<2:0> ビットが「000」に設定された状態で変換を開始する場合、変換動作までに 1 命令サイクルの遅延があるため、SLEEP 命令を実行してスリープに移行できます。変換開始前に、OSCCON レジスタの IDLEN および SCS ビットをクリアしておく必要があります。

表 22-2: A/D コンバータ レジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	リセット値の記載ページ
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	69
PIR1	PSPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	71
PIE1	PSPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	71
IPR1	PSPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	71
PIR2	OSCFIF	CMIF	ETHIF	r	BCL1IF	-	TMR3IF	CCP2IF	71
PIE2	OSCFIE	CMIE	ETHIE	r	BCL1IE	-	TMR3IE	CCP2IE	71
IPR2	OSCFIP	CMIP	ETHIP	r	BCL1IP	-	TMR3IP	CCP2IP	71
ADRESH	A/D 変換結果レジスタ上位バイト								70
ADRESL	A/D 変換結果レジスタ下位バイト								70
ADCON0	ADCAL	-	CHS3	CHS3	CHS1	CHS0	GO/DONE	ADON	70
ADCON1	-	-	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	70
ADCON2	ADFM	-	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0	70
CCP2CON	P2M1	P2M0	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	70
PORTA	RJPU	-	RA5	RA4	RA3	RA2	RA1	RA0	72
TRISA	-	-	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	71
PORTF	RF7	RF6	RF5	RF4	RF3	RF2	RF1	RF0 <sup>(1)</sup>	72
TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	TRISF0 <sup>(1)</sup>	71
PORTH <sup>(2)</sup>	RH7	RH6	RH5	RH4	RH3	RH2	RH1	RH0	72
TRISH <sup>(2)</sup>	TRISH7	TRISH6	TRISH5	TRISH4	TRISH3	TRISH2	TRISH1	TRISH0	71

凡例: - = 未実装ビット、「0」として読み出されます。r = 予約済みです。網掛けの部分は A/D 変換では使いません。

Note 1: 100 ピンデバイスにのみ実装されています。

2: このレジスタは 64 ピンデバイスには実装されていません。

# PIC18F97J60 ファミリ

---

NOTES:

## 23.0 コンパレータ モジュール

このアナログ コンパレータ モジュールは、各種設定が可能な2つのコンパレータを収めています。入力は、RF1 ~ RF6 ピンと多重化されたアナログ入力と内蔵参照電圧 ( [セクション 24.0「コンパレータ参照電圧モジュール」](#) 参照 ) から選択できます。デジタル出力 ( 通常または反転 ) はピンレベルで使用可能であり、制御レジスタを介して読み出す事もできます。

CMCON レジスタ ( [レジスタ 23-1](#) ) でコンパレータの入出力を設定します。 [図 23-1](#) に、コンパレータの各種設定のブロック図を示します。

**レジスタ 23-1: CMCON: コンパレータ制御レジスタ**

R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1	R/W-1
C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

- bit 7      **C2OUT:** コンパレータ 2 出力ビット  
C2INV = 0 の場合:  
 1 = C2 VIN+ > C2 VIN-  
 0 = C2 VIN+ < C2 VIN-  
C2INV = 1 の場合:  
 1 = C2 VIN+ < C2 VIN-  
 0 = C2 VIN+ > C2 VIN-
- bit 6      **C1OUT:** コンパレータ 1 出力ビット  
C1INV = 0 の場合:  
 1 = C1 VIN+ > C1 VIN-  
 0 = C1 VIN+ < C1 VIN-  
C1INV = 1 の場合:  
 1 = C1 VIN+ < C1 VIN-  
 0 = C1 VIN+ > C1 VIN-
- bit 5      **C2INV:** コンパレータ 2 出力反転ビット  
 1 = C2 出力を反転する  
 0 = C2 出力を反転しない
- bit 4      **C1INV:** コンパレータ 1 出力反転ビット  
 1 = C1 出力を反転する  
 0 = C1 出力を反転しない
- bit 3      **CIS:** コンパレータ入力スイッチビット  
CM<2:0> = 110 の場合:  
 1 = C1 VIN- を RF5/AN10/CVREF に接続し  
       C2 VIN- を RF3/AN8 に接続する  
 0 = C1 VIN- を RF6/AN11 に接続し  
       C2 VIN- を RA4/AN9 に接続する
- bit 2-0      **CM<2:0>:** コンパレータ モードビット  
[図 23-1](#) に、コンパレータのモードと CM<2:0> ビットの設定を示します。

# PIC18F97J60 ファミリ

## 23.1 コンパレータの設定

コンパレータには、図 23-1 に示す 8 つの動作モードがあります。これらのモードの選択には、CMCON レジスタの CM<2:0> ビットを使います。各モードのコンパレータピンのデータ方向はTRISFレジスタによって制御します。コンパレータのモードを変更すると、

セクション 28.0「電気的特性」に記載された規定のモード変更遅延時間が経過するまで、コンパレータの出力レベルが無効になる場合があります。

**Note:** コンパレータ モード変更中は、コンパレータ割り込みを無効にする必要があります。有効のままの場合、誤った割り込みが発生する場合があります。

図 23-1: コンパレータの I/O 動作モード

<p><b>Comparator Outputs Disabled</b> CM&lt;2:0&gt; = 000</p>	<p><b>Comparators Off (POR Default Value)</b> CM&lt;2:0&gt; = 111</p>
<p><b>Two Independent Comparators</b> CM&lt;2:0&gt; = 010</p>	<p><b>Two Independent Comparators with Outputs</b> CM&lt;2:0&gt; = 011</p>
<p><b>Two Common Reference Comparators</b> CM&lt;2:0&gt; = 100</p>	<p><b>Two Common Reference Comparators with Outputs</b> CM&lt;2:0&gt; = 101</p>
<p><b>One Independent Comparator with Output</b> CM&lt;2:0&gt; = 001</p>	<p><b>Four Inputs Multiplexed to Two Comparators</b> CM&lt;2:0&gt; = 110</p>
<p>A = アナログ入力、ポートは常にゼロとして読み出し      D = デジタル入力      CIS (CMCON&lt;3&gt;) はコンパレータ入力スイッチ * TRISF&lt;2:1&gt; ビットをセットすると、対応するピンが入力に設定され、コンパレータ出力が無効になります。</p>	

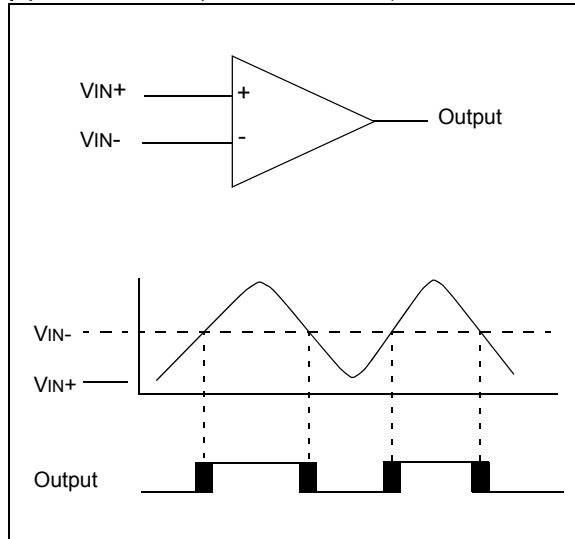
## 23.2 コンパレータの動作

図 23-2 に、シングルコンパレータと、アナログ入力レベルとデジタル出力の関係を示します。VIN+ のアナログ入力が VIN- より小さい場合、コンパレータの出力はデジタル Low レベルです。VIN+ のアナログ入力が VIN- より大きい場合、コンパレータの出力はデジタル High レベルです。図 23-2 でコンパレータ出力が塗りつぶされている領域は、入力オフセットと応答時間によって不確定である事を示します。

## 23.3 コンパレータの参照電圧

コンパレータの動作モードに応じて、外部または内部の参照電圧を使えます。VIN- に印加されているアナログ信号を、VIN+ の信号と比較し、その結果に基づいてコンパレータのデジタル出力を調整します (図 23-2)。

図 23-2: シングルコンパレータ



### 23.3.1 外部参照信号

外部の参照電圧を使う場合、コンパレータ モジュール内の 2 つのコンパレータを、同じ参照電圧源によって動作させるか、異なる参照電圧源によって動作させるかを設定できます。しかし、しきい値検出アプリケーションでは、同じ参照電圧が必要となる場合があります。参照信号は、VSS から VDD のレンジで、コンパレータのどちらのピンに印加してもかまいません。

### 23.3.2 内部参照信号

コンパレータ モジュールでは、コンパレータ参照電圧モジュールによって内部生成した参照電圧を選択する事もできます。このモジュールの詳細は、セクション 24.0「コンパレータ参照電圧モジュール」に記載しています。

内部参照電圧を使うのは、4 つの入力が 2 つのコンパレータ入力に多重化されるモード (CM<2:0> = 110) のみです。このモードでは、内部参照電圧が両方のコンパレータの VIN+ ピンに印加されます。

## 23.4 コンパレータの応答時間

応答時間とは、新しい参照電圧または入力源を選択してからコンパレータ出力が有効なレベルに達するまでの最小時間です。内部参照電圧を変更した場合、コンパレータ出力を使う際に内部参照電圧の最大遅延を考慮する必要があります。または、コンパレータの最大遅延を使います (セクション 28.0「電気的特性」参照)。

## 23.5 コンパレータ出力

コンパレータの出力は CMCON レジスタから読み出します。これらのビットは読み出し専用です。コンパレータの出力を I/O ピンの RF1 と RF2 に直接出力する事もできます。RF1 と RF2 ピンの出力パスにあるマルチプレクサを有効にすると出力の切り換えが行われ、各ピンはコンパレータの非同期出力になります。各コンパレータの不確かさは、仕様に記載されている入力オフセット電圧と応答時間に関連しています。図 23-2 に、コンパレータ出力のブロック図を示します。

このモードでも、TRISF ビットが RF1 と RF2 ピンの出力を有効 / 無効にする機能を果たします。

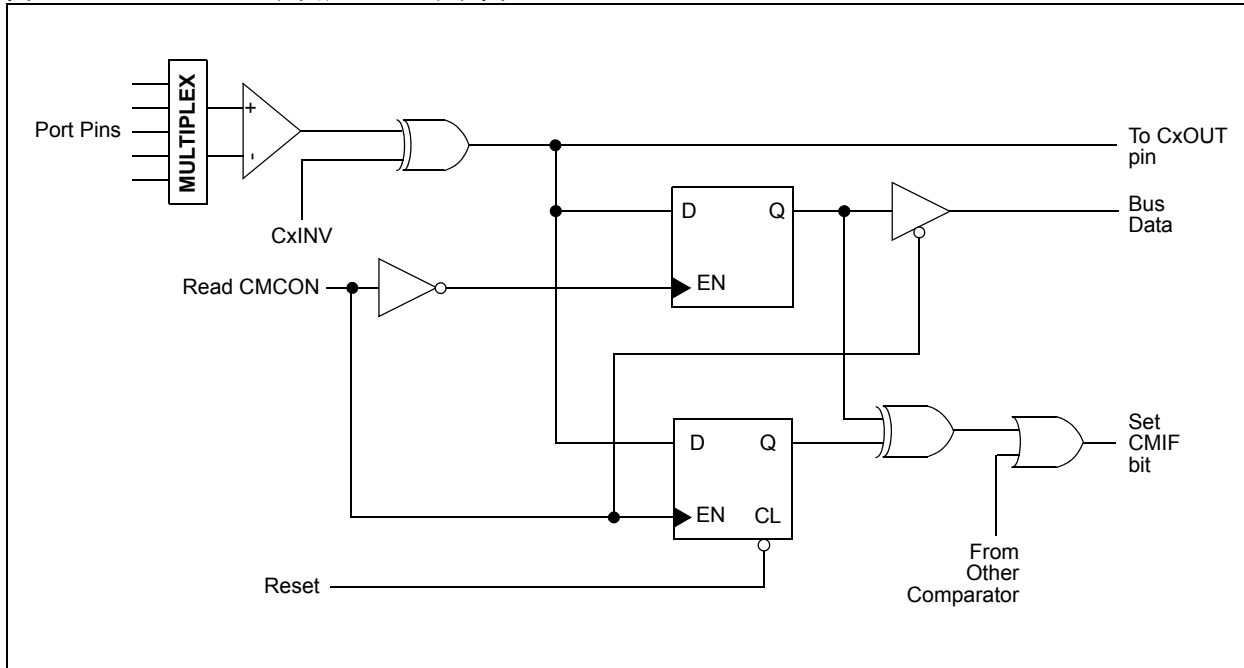
コンパレータの出力極性は、C2INV および C1INV ビット (CMCON<5:4>) によって変更できます。

**Note 1:** ポートレジスタを読み出す場合、アナログ入力として設定したピンは全て「0」として読み出されます。デジタル入力として設定されているピンは、シュミットトリガ入力仕様に従ってアナログ入力を変換します。

**2:** デジタル入力として定義したピンにアナログ電圧を印加すると、入力バッファに仕様を超える電流が流れる場合があります。

# PIC18F97J60 ファミリ

図 23-3: コンパレータ出力のブロック図



## 23.6 コンパレータ割り込み

いずれかのコンパレータの出力値が変化するとコンパレータ割り込みフラグがセットされます。実際に起こった変化を判断するには、CMCON<7:6> から読み出される出力ビットのステータス情報をソフトウェアに保持しておく必要があります。CMIF ビット (PIR2<6>) は、コンパレータ割り込みフラグです。CMIF ビットは、クリアしてリセットする必要があります。このレジスタには「1」を書き込む事ができるため、割り込みをシミュレートする事もできます。

割り込みを有効にするには、CMIE ビット (PIE2<6>) と PEIE ビット (INTCON<6>)の両方をセットする必要があります。さらに、GIE ビット (INTCON<7>) もセットする必要があります。これらのビットのどちらかがクリアの場合、割り込み条件の発生時に CMIF ビットはセットされますが、割り込みは有効になりません。

**Note:** 読み出し動作の実行中 (Q2 サイクルの開始時点) に、CMCON レジスタ (C1OUT または C2OUT) が変化した場合、CMIF (PIR2 レジスタ) 割り込みフラグがセットされない事があります。

割り込みサービスルーチン内で、以下の方法で割り込みをクリアできます。

- CMCON に対する読み書きで不一致条件を解消する
- フラグビット CMIF をクリアする

条件が不一致のままである場合、フラグビット CMIF はセットされたままです。CMCON を読み出す事で不一致条件が解消し、フラグビット CMIF をクリアできるようになります。

## 23.7 スリープ中のコンパレータの動作

コンパレータがアクティブでデバイスがスリープの場合、コンパレータはアクティブのままです。割り込みが有効の場合、割り込みは機能します。この割り込みによって、デバイスはスリープから復帰します (有効な場合)。コンパレータ仕様に記載されているように、動作中の各コンパレータは電流を消費します。スリープ中の電力消費を最小限とするには、スリープに移行する前にコンパレータを OFF にします (CM<2:0> = 111)。デバイスがスリープから復帰しても、CMCON レジスタの内容は変化しません。

## 23.8 リセットの影響

デバイスをリセットすると、CMCON レジスタが強制的にリセット状態になり、コンパレータ モジュールが OFF になります (CM<2:0> = 111)。しかし、入力ピン (RF3 ~ RF6) はデバイスリセット時に既定値でアナログ入力に設定されます。これらのピンの I/O 設定は、PCFG<3:0> ビット (ADCON1<3:0>) の設定によって決まります。従って、リセット時にアナログ入力が与えられている場合、デバイス電流は最小限に抑えられます。



## 23.9 アナログ入力接続に関する注意事項

図 23-4 に、アナログ入力回路の概略図を示します。アナログピンは、デジタル出力に接続されているため、VDDとVssへの逆バイアスのダイオードが接続されています。従って、アナログ入力はVssからVDDのレンジ内の電圧である必要があります。入力電圧がこのレンジの上下に0.6 Vを超えて逸脱すると、いずれか

一方のダイオードが順バイアスされラッチアップ条件が発生する恐れがあります。アナログソースの最大推奨ソースインピーダンス値は10 kΩです。アナログ入力ピンに接続するコンデンサやツェナーダイオード等の外付け部品は、リーク電流がほとんどないものにします。

図 23-4: コンパレータのアナログ入力モデル

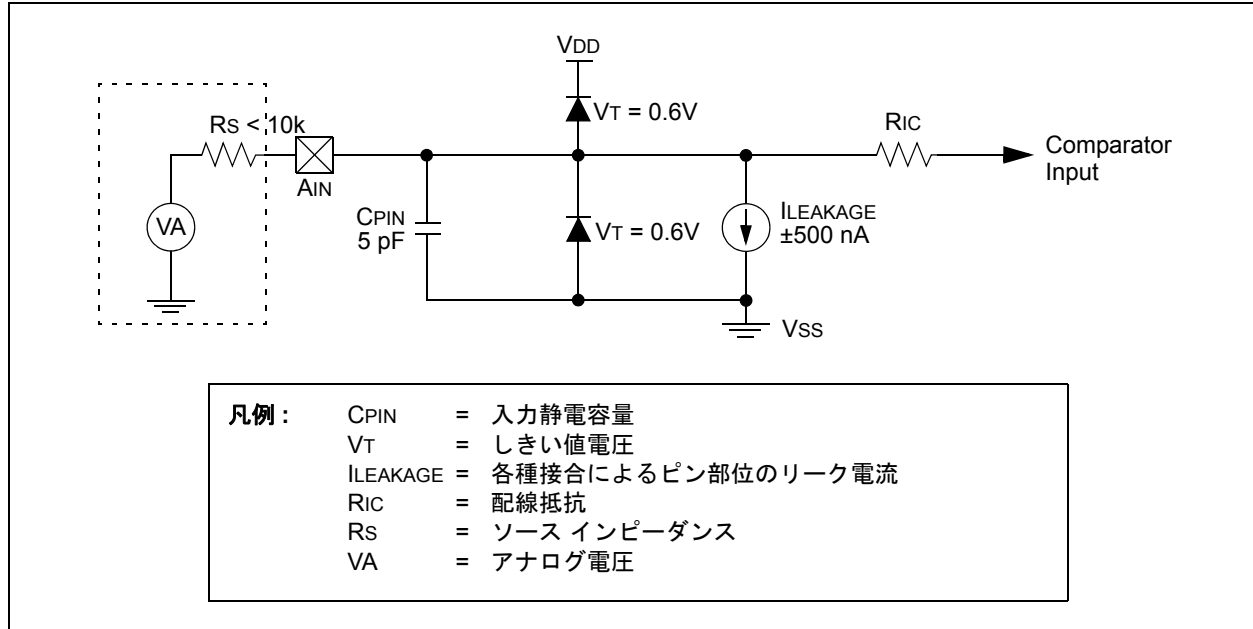


表 23-1: コンパレータ モジュールに関連するレジスタ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	リセット値の記載ページ
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	69
PIR2	OSCFIF	CMIF	ETHIF	r	BCL1IF	-	TMR3IF	CCP2IF	71
PIE2	OSCFIE	CMIE	ETHIE	r	BCL1IE	-	TMR3IE	CCP2IE	71
IPR2	OSCFIP	CMIP	ETHIP	r	BCL1IP	-	TMR3IP	CCP2IP	71
CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	70
CVRCON	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	70
PORTF	RF7	RF6	RF5	RF4	RF3	RF2	RF1	RF0	72
TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	TRISF0	71

凡例: - = 未実装ビット、「0」として読み出されます。r = 予約済みです。網掛けの部分はコンパレータ モジュールでは使いません。

# PIC18F97J60 ファミリ

---

NOTES:

## 24.0 コンパレータ参照電圧モジュール

コンパレータ参照電圧モジュールは、参照電圧を選択できる 16 タップの抵抗ラダー ネットワークです。その主目的はアナログ コンパレータへの参照電圧供給ですが、コンパレータとは別に使う事もできます。

図 24-1 に、このモジュールのブロック図を示します。抵抗ラダーは、CVREF の値として 2 つのレンジを提供できるように分割されています。また、非使用時の消費電力を低減するパワーダウン機能も備えています。本モジュールの参照電圧源は、デバイスの VDD/VSS または外部参照電圧のいずれかより供給できます。

### 24.1 コンパレータ参照電圧の設定

参照電圧モジュールは、CVRCON レジスタ ( レジスタ 24-1) を使います。コンパレータ参照電圧は、それぞれ 16 段階からなる 2 つの出力電圧レンジを提供します。使うレンジは、CVRR ビット (CVRCON<5>) に

よって選択します。2 つのレンジ間の主な差異は、CVREF 選択ビット (CVR<3:0>) で選択するステップサイズであり、一方のレンジの方がより細かい分解能を提供します。コンパレータ参照電圧は以下の式で計算します。

$$\text{CVRR} = 1 \text{ の場合:} \\ \text{CVREF} = ((\text{CVR}<3:0>)/24) \times (\text{CVRSRC})$$

$$\text{CVRR} = 0 \text{ の場合:} \\ \text{CVREF} = (\text{CVRSRC}/4) + ((\text{CVR}<3:0>)/32) \times (\text{CVRSRC})$$

コンパレータ参照電圧源は、VDD/VSS と、RA2/RA3 と多重化した外部 VREF+/VREF- のいずれかより供給できます。この電圧源は、CVRSS ビット (CVRCON<4>) で選択します。

CVREF 出力を変更する際は、コンパレータ参照電圧のセトリングタイムを考慮する必要があります ( セクション 28.0 「電気的特性」 の表 28-3 参照 )。

レジスタ 24-1: CVRCON: コンパレータ参照電圧制御レジスタ

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CVREN	CVROE <sup>(1)</sup>	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

- bit 7      **CVREN:** コンパレータ参照電圧イネーブルビット  
 1 = CVREF 回路の電源を ON にする  
 0 = CVREF 回路の電源を OFF にする
- bit 6      **CVROE:** コンパレータ VREF 出力イネーブルビット<sup>(1)</sup>  
 1 = CVREF 電圧レベルを RF5/AN10/CVREF ピンにも出力する  
 0 = CVREF 電圧レベルを RF5/AN10/CVREF ピンから切り離す
- bit 5      **CVRR:** コンパレータ VREF レンジ選択ビット  
 1 = 0 ~ 0.667 CVRSRC、ステップサイズ CVRSRC/24 (Low レンジ)  
 0 = 0.25 CVRSRC ~ 0.75 CVRSRC、ステップサイズ CVRSRC/32 (High レンジ)
- bit 4      **CVRSS:** コンパレータ VREF 参照電圧源選択ビット  
 1 = コンパレータ参照電圧源を CVRSRC = (VREF+) - (VREF-) とする  
 0 = コンパレータ参照電圧源を CVRSRC = VDD - VSS とする
- bit 3-0      **CVR<3:0>:** コンパレータ VREF 値選択ビット (0 ≤ (CVR<3:0>) ≤ 15)  
 CVRR = 1 の場合:  

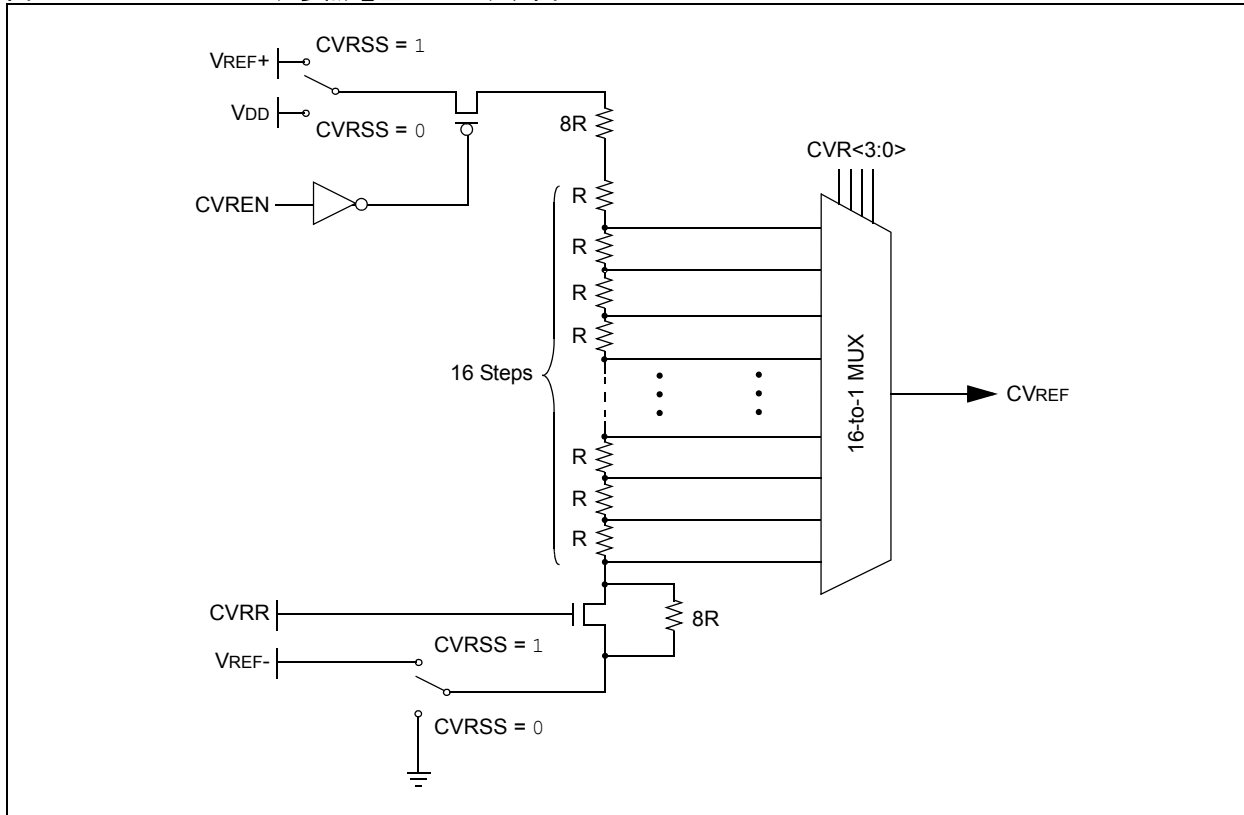
$$\text{CVREF} = ((\text{CVR}<3:0>)/24) \cdot (\text{CVRSRC})$$
  
 CVRR = 0 の場合:  

$$\text{CVREF} = (\text{CVRSRC}/4) + ((\text{CVR}<3:0>)/32) \cdot (\text{CVRSRC})$$

**Note 1:** CVROE は TRISF<5> ビットの設定より優先されます。

# PIC18F97J60 ファミリ

図 24-1: コンパレータ参照電圧のブロック図



## 24.2 参照電圧の精度と誤差

モジュールの回路構成上、参照電圧の全レンジを実現する事はできません。抵抗ラダー ネットワークの上下にトランジスタ (図 24-1) がある事から、CVREF を参照電圧源のレールに近づけられないためです。参照電圧はこの参照電圧源から供給されるため、その変動に従って CVREF 出力も変化します。参照電圧の試験済み絶対精度は、[セクション 28.0「電気的特性」](#)を参照してください。

## 24.3 スリープ中の動作

割り込み、またはウォッチドッグ タイマのタイムアウトによって、デバイスがスリープから復帰した場合、CVRCON レジスタの内容は変化しません。スリープ中の消費電流をできるだけ小さくするには、参照電圧を無効にします。

## 24.4 リセットの影響

デバイスリセットは、CVREN (CVRCON<7>) ビットをクリアする事で参照電圧を無効化します。このリセットは、さらに CVROE (CVRCON<6>) ビットをクリアして参照電圧を RA2 ピンから切り離し、CVRR (CVRCON<5>) ビットをクリアして高压レンジを選択します。CVR 値選択ビットもクリアされます。

## 24.5 接続に関する注意事項

参照電圧モジュールは、コンパレータ モジュールとは独立して動作します。CVROE ビットがセットされている場合、参照電圧ジェネレータの出力を RF5 ピンに接続できます。RA2 ピンをデジタル入力に設定した状態で、このピンへの参照電圧出力を有効にすると、消費電流が増加します。CVRSS を有効にして RF5 をデジタル出力として接続した場合も、消費電流が増加します。

RF5 ピンは、(駆動能力は限定されているけれども) 簡単な D/A 出力として使えます。VREF を外部に接続して参照電圧を出力する場合、電流駆動能力に制約があるため、バッファを使う必要があります。図 24-2 に、バッファの使用例を示します。

図 24-2: コンパレータ参照電圧の出力バッファの例

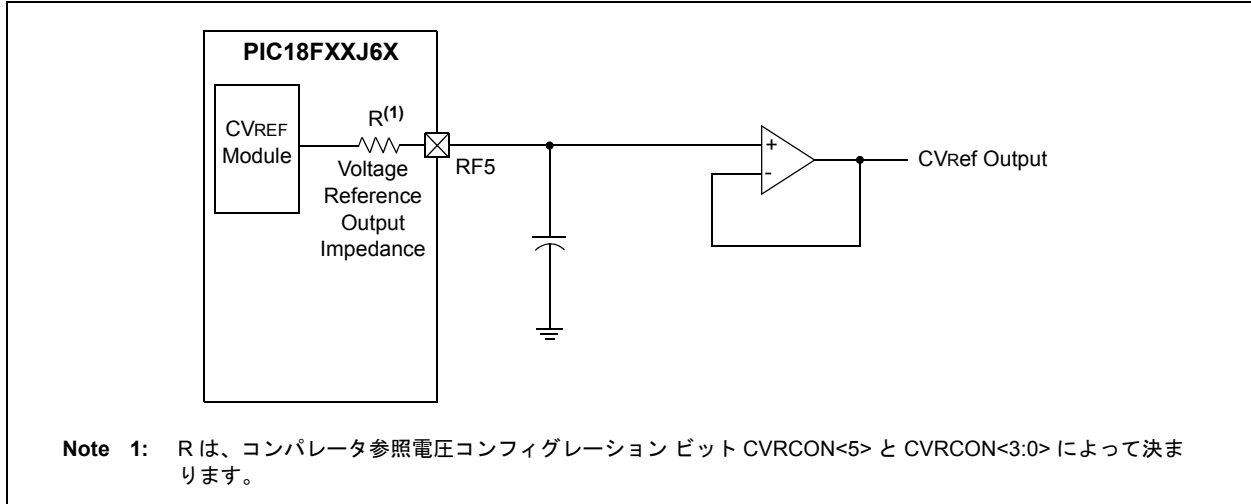


表 24-1: コンパレータ参照電圧に関連するレジスタ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	リセット値の記載ページ
CVRCON	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	70
CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	70
TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	TRISF0	71

凡例: - = 未実装ビット、「0」として読み出されます。網掛けの部分はコンパレータ参照電圧モジュールでは使いません。

# PIC18F97J60 ファミリ

---

NOTES:

## 25.0 CPU の特殊機能

PIC18F97J60 ファミリは、外付け部品を省いてコストを低減、信頼性を向上させるために以下の機能を備えています。

- オシレータの選択
- リセット：
  - パワーオン リセット (POR)
  - パワーアップ タイマ (PWRT)
  - オシレータ起動タイマ (OST)
  - ブラウンアウト リセット (BOR)
- 割り込み
- ウォッチドッグ タイマ (WDT)
- フェイルセーフ クロック モニタ
- 2 段階起動
- コード保護
- インサーキット シリアル プログラミング

オシレータは、アプリケーションの周波数、電力、精度、コストに合わせて構成できます。全オプションの詳細は、[セクション 3.0「オシレータの設定」](#)で説明しています。

デバイスのリセットと割り込みの詳細は、本データシートの別のセクションで詳しく説明しています。

PIC18F97J60 ファミリは、リセット用のパワーアップ タイマとオシレータ起動タイマに加えて、ソフトウェアで設定可能なウォッチドッグ タイマを備えています。

また、内部 RC オシレータによりフェイルセーフ クロック モニタ (FSCM) と 2 段階起動も実現しています。FSCM は周辺クロックをバックグラウンドで監視し、障害発生時に自動的に内部 RC オシレータのクロックに切り換える機能です。2 段階起動とは、プライマリ クロック源が起動して安定するのを待たずに、ほぼ瞬時にコード実行を開始できる機能です。

これらの機能は全て適切なコンフィグレーション レジスタのビットによって有効 / 無効を切り換え、設定を行います。

### 25.1 コンフィグレーション ビット

コンフィグレーション ビットをプログラムする (「0」として読み出し) か、プログラムしない (「1」として読み出し) かによって、デバイスの各種設定が可能です。これらのビットは、プログラムメモリの 300000h から始まるアドレスに割り当てられています。[表 25-1](#)に、コンフィグレーション ビットの全リストを示します。各種ビット機能の詳細な説明を [レジスタ 25-1 ~ レジスタ 25-8](#) に記載します。

#### 25.1.1 PIC18F97J60 ファミリの設定における注意事項

PIC18F97J60 ファミリは、コンフィグレーション情報の格納に永続的なメモリレジスタを使いません。コンフィグレーションバイトは揮発性メモリとして実装されます。これは、デバイスの電源を投入するたびに、コンフィグレーション データをプログラムする必要のある事を意味します。

コンフィグレーション データは内蔵プログラムメモリ空間最上位の 4 ワードに格納されます。これらのワードはフラッシュ コンフィグレーション ワードと呼ばれ、[表 6-1](#) に示すプログラムメモリ空間内に配置されています。コンフィグレーション ワードは、[表 25-1](#) の記載と同じ順序で格納されます。つまり、CONFIG1L が最下位、CONFIG3H が最上位です。このデータはデバイス起動時に、自動的に適切なコンフィグレーション レジスタに読み込まれます。

これらのデバイス用にアプリケーションを作成する場合、コンフィグレーション データのためのフラッシュ コンフィグレーション ワードのアドレスを具体的に割り当てる必要があります。これは、コードのコンパイル時にこのアドレスにプログラムコードが格納されないようにするためです。

コンフィグレーション ビットに使う揮発性メモリセルは、パワーオン リセット時に必ず「1」にリセットされます。その他の種類のリセットイベントでは、前回プログラムされた値が保持され、プログラムメモリから再読み込みせずに、そのまま適用されます。

プログラムメモリ内の CONFIG1H、CONFIG2H、CONFIG3H の上位 4 ビットも「1111」とする事が必要です。これは、万一これらのアドレスが誤って実行されたとしても、これらのコンフィグレーション ワードが NOP 命令と見なされるようにするためです。対応する位置にはコンフィグレーション ビットが実装されていないため、その位置に「1」を書き込んでもデバイスの動作には影響しません。

コード実行中に誤ってコンフィグレーションを変更する事のないよう、プログラマブルなコンフィグレーション ビットは全てリセット後の書き込みは 1 回限りです。電源投入によって最初にビットが書き込まれたら、再度書き込む事はできません。デバイスのコンフィグレーションを変更するには、電源の遮断 / 再投入が必要です。

# PIC18F97J60 ファミリ

表 25-1: コンフィグレーションビットとデバイス ID

ファイル名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	既定値 / 未プログラム時の値 <sup>(1)</sup>	
300000h	CONFIG1L	DEBUG	XINST	STVREN	—	—	—	—	WDTEN	110- ---1
300001h	CONFIG1H	_(2)	_(2)	_(2)	_(2)	_(3)	CP0	—	—	---- 01--
300002h	CONFIG2L	IESO	FCMEN	—	—	—	FOSC2	FOSC1	FOSC0	11-- -111
300003h	CONFIG2H	_(2)	_(2)	_(2)	_(2)	WDTPS3	WDTPS2	WDTPS1	WDTPS0	---- 1111
300004h	CONFIG3L	WAIT <sup>(4)</sup>	BW <sup>(4)</sup>	EMB1 <sup>(4)</sup>	EMB0 <sup>(4)</sup>	EASHFT <sup>(4)</sup>	—	—	—	1111 1---
300005h	CONFIG3H	_(2)	_(2)	_(2)	_(2)	—	ETHLED	ECCPMX <sup>(5)</sup>	CCP2MX <sup>(5)</sup>	---- -111
3FFFFEh	DEVID1	DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0	xxxx xxxx <sup>(6)</sup>
3FFFFFh	DEVID2	DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3	xxxx xxxx <sup>(6)</sup>

凡例: x = 未知、u = 不変、— = 未実装です。網掛けの部分は未実装で、「0」として読み出されます。

- Note 1:** 工場出荷時の未プログラム状態のデバイスに対してパワーオンリセットを実行した後の値です。その他の全てのリセット状態では、コンフィグレーションバイトは前回プログラムした状態を維持します。
- 2:** プログラムメモリ内のこれらのビットの値は常に「1」にしておきます。これにより、このアドレスが誤って実行されたとしてもNOPと見なされるようにします。
- 3:** このビットは常に「0」のままにしておく必要があります。
- 4:** 100ピンデバイスにのみ実装されています。
- 5:** 80ピン/100ピンデバイスにのみ実装されています。
- 6:** DEVIDの値については、[レジスタ 25-7](#)と[レジスタ 25-8](#)を参照してください。これらのレジスタは読み出し専用で、ユーザは書き込みできません。



# PIC18F97J60 ファミリ

## レジスタ 25-1: CONFIG1L: コンフィグレーション レジスタ 1 Low (バイトアドレス 300000h)

R/WO-1	R/WO-1	R/WO-0	U-0	U-0	U-0	U-0	R/WO-1
$\overline{\text{DEBUG}}$	XINST	STVREN	—	—	—	—	WDTEN
bit 7							bit 0

### 凡例:

R = 読み出し可能ビット WO = ライトワンス ビット U = 未実装ビット、「0」として読み出し  
 -n = デバイス未プログラム時の値 「1」= ビットはセット 「0」= ビットはクリア

- bit 7 **DEBUG:** バックグラウンド デバッガ イネーブルビット  
 1 = バックグラウンド デバッガを無効にする。RB6 と RB7 を汎用 I/O ピンに設定する  
 0 = バックグラウンド デバッガを有効にする。RB6 と RB7 をインサーキット デバッグ専用 に設定する
- bit 6 **XINST:** 拡張命令セット イネーブルビット  
 1 = 拡張命令セットとインデックス付きアドレッシング モードを有効にする  
 0 = 拡張命令セットとインデックス付きアドレッシング モードを無効にする (レガシーモード)
- bit 5 **STVREN:** スタック オーバーフロー/アンダーフロー リセット イネーブルビット  
 1 = スタック オーバーフロー/アンダーフローによるリセットを有効にする  
 0 = スタック オーバーフロー/アンダーフローによるリセットを無効にする
- bit 4-1 **未実装:** 「0」として読み出し
- bit 0 **WDTEN:** ウォッチドッグ タイマ イネーブルビット  
 1 = WDT を有効にする  
 0 = WDT を無効にする (SWDTEN ビットで制御)

## レジスタ 25-2: CONFIG1H: コンフィグレーション レジスタ 1 High (バイトアドレス 300001h)

U-0	U-0	U-0	U-0	U-0 <sup>(1)</sup>	R/WO-1	U-0	U-0
— <sup>(2)</sup>	— <sup>(2)</sup>	— <sup>(2)</sup>	— <sup>(2)</sup>	—	CP0	—	—
bit 7							bit 0

### 凡例:

R = 読み出し可能ビット WO = ライトワンス ビット U = 未実装ビット、「0」として読み出し  
 -n = デバイス未プログラム時の値 「1」= ビットはセット 「0」= ビットはクリア

- bit 7-3 **未実装:** 「0」として読み出し
- bit 2 **CP0:** コード保護ビット  
 1 = プログラムメモリのコード保護を無効にする  
 0 = プログラムメモリのコード保護を有効にする
- bit 1-0 **未実装:** 「0」として読み出し

- Note 1:** このビットは常に「0」のままにしておく必要があります。
- 2:** プログラムメモリ内のこれらのビットの値は常に「1」にしておきます。これにより、このアドレスが誤って実行されたとしても NOP と見なされるようにします。

# PIC18F97J60 ファミリ

## レジスタ 25-3: CONFIG2L: コンフィグレーション レジスタ 2 Low (バイトアドレス 300002h)

R/WO-1	R/WO-1	U-0	U-0	U-0	R/WO-1	R/WO-1	R/WO-1
IESO	FCMEN	—	—	—	FOSC2	FOSC1	FOSC0
bit 7							bit 0

### 凡例:

R = 読み出し可能ビット WO = ライトワンス ビット U = 未実装ビット、「0」として読み出し

-n = デバイス未プログラム時の値 「1」= ビットはセット 「0」= ビットはクリア

bit 7 **IESO:** 2 段階起動 ( 内部 / 外部オシレータ切り換え ) 制御ビット

1 = 2 段階起動を有効にする

0 = 2 段階起動を無効にする

bit 6 **FCMEN:** フェイルセーフ クロックモニタ イネーブルビット

1 = フェイルセーフ クロックモニタを有効にする

0 = フェイルセーフ クロックモニタを無効にする

bit 5-3 **未実装:** 「0」として読み出し

bit 2 **FOSC2:** 既定値 / リセット時システムクロック選択ビット

1 = OSCCON<1:0> = 00 の場合、FOSC<1:0> でシステムクロックとして選択したクロックを有効にする

0 = OSCCON<1:0> = 00 の場合、INTRC をシステムクロックとして有効にする

bit 1-0 **FOSC<1:0>:** オシレータ選択ビット

11 = EC オシレータを選択し、ソフトウェア制御で PLL を有効にする。OSC2 ピンは CLKO を出力する

10 = EC オシレータを選択する。OSC2 ピンは CLKO を出力する

01 = HS オシレータを選択し、ソフトウェア制御で PLL を有効にする

00 = HS オシレータを選択する

# PIC18F97J60 ファミリ

レジスタ 25-4: CONFIG2H: コンフィグレーション レジスタ 2 High (バイトアドレス 300003h)

U-0	U-0	U-0	U-0	R/WO-1	R/WO-1	R/WO-1	R/WO-1
_(1)	_(1)	_(1)	_(1)	WDTPS3	WDTPS2	WDTPS1	WDTPS0
bit 7							bit 0

**凡例:**

R = 読み出し可能ビット    WO = ライトワンス ビット    U = 未実装ビット、「0」として読み出し  
-n = デバイス未プログラム時の値    「1」= ビットはセット    「0」= ビットはクリア

bit 7-4    **未実装:** 「0」として読み出し

bit 3-0    **WDTPS<3:0>:** ウォッチドッグ タイマ ポストスケール選択ビット

1111 = 1:32,768  
1110 = 1:16,384  
1101 = 1:8,192  
1100 = 1:4,096  
1011 = 1:2,048  
1010 = 1:1,024  
1001 = 1:512  
1000 = 1:256  
0111 = 1:128  
0110 = 1:64  
0101 = 1:32  
0100 = 1:16  
0011 = 1:8  
0010 = 1:4  
0001 = 1:2  
0000 = 1:1

**Note 1:** プログラムメモリ内のこれらのビットの値は常に「1」にしておきます。これにより、このアドレスが誤って実行されたとしても NOP と見なされるようにします。

# PIC18F97J60 ファミリ

## レジスタ 25-5: CONFIG3L: コンフィグレーション レジスタ 3 Low (バイトアドレス 300004h)

R/WO-1	R/WO-1	R/WO-1	R/WO-1	R/WO-1	U-0	U-0	U-0
WAIT <sup>(1)</sup>	BW <sup>(1)</sup>	EMB1 <sup>(1)</sup>	EMB0 <sup>(1)</sup>	EASHFT <sup>(1)</sup>	—	—	—
bit 7							bit 0

### 凡例:

R = 読み出し可能ビット WO = ライトワンス ビット U = 未実装ビット、「0」として読み出し

-n = デバイス未プログラム時の値 「1」= ビットはセット 「0」= ビットはクリア

- bit 7      **WAIT:** 外部バスウェイト イネーブルビット<sup>(1)</sup>  
1 = 外部メモリバスに対する操作のウェイトステートを無効にする  
0 = 外部メモリバスに対する操作のウェイトステートを有効にし、MEMCON<5:4> で選択する
- bit 6      **BW:** データバス幅選択ビット<sup>(1)</sup>  
1 = 16 ビットデータ幅モード  
0 = 8 ビットデータ幅モード
- bit 5-4    **EMB<1:0>:** 外部メモリバス コンフィグレーション ビット<sup>(1)</sup>  
11 = マイクロコントローラ モード、外部バスを無効にする  
10 = 拡張マイクロコントローラ モード、12 ビット アドレッシング モード  
01 = 拡張マイクロコントローラ モード、16 ビット アドレッシング モード  
00 = 拡張マイクロコントローラ モード、20 ビット アドレッシング モード
- bit 3      **EASHFT:** 外部アドレスバス シフト イネーブルビット<sup>(1)</sup>  
1 = アドレスシフトを有効にする。外部バスのアドレスが 000000h から開始するようにオフセットする  
0 = アドレスシフトを無効にする。外部バスのアドレスをそのまま PC の値にする
- bit 2-0    **未実装:** 「0」として読み出し

**Note 1:** 100 ピンデバイスにのみ実装されています。

# PIC18F97J60 ファミリ

## レジスタ 25-6: CONFIG3H: コンフィグレーションレジスタ 3 High (バイトアドレス 300005h)

U-0	U-0	U-0	U-0	U-0	R/WO-1	R/WO-1	R/WO-1
— <sup>(1)</sup>	— <sup>(1)</sup>	— <sup>(1)</sup>	— <sup>(1)</sup>	—	ETHLED	ECCPMX <sup>(2)</sup>	CCP2MX <sup>(2)</sup>
bit 7							bit 0

### 凡例:

R = 読み出し可能ビット    WO = ライトワンス ビット    U = 未実装ビット、「0」として読み出し

-n = デバイス未プログラム時の値

「1」= ビットはセット

「0」= ビットはクリア

bit 7-3    **未実装:** 「0」として読み出し

bit 2    **ETHLED:** Ethernet LED イネーブルビット

1 = Ethernet モジュール有効時には RA0/RA1 を LEDA/LEDB と多重化し、無効時は I/O として機能させる

0 = RA0/RA1 を、Ethernet モジュールの状態に関係なく I/O として機能させる

bit 1    **ECCPMX:** ECCP MUX ビット<sup>(2)</sup>

1 = ECCP1 出力 (P1B/P1C) を RE6/RE5 と多重化する

ECCP3 出力 (P3B/P3C) を RE4/RE3 と多重化する

0 = ECCP1 出力 (P1B/P1C) を RH7/RH6 と多重化する

ECCP3 出力 (P3B/P3C) を RH5/RH4 と多重化する

bit 0    **CCP2MX:** ECCP2 MUX ビット<sup>(2)</sup>

1 = ECCP2/P2A を RC1 と多重化する

0 = マイクロコントローラ モード時、ECCP2/P2A を RE7 と多重化する (80 ピン /100 ピンデバイス)  
または、拡張マイクロコントローラ モード時、RB3 と多重化する (100 ピンデバイスのみ)

**Note 1:** プログラムメモリ内のこれらのビットの値は常に「1」にしておきます。これにより、このアドレスが誤って実行されたとしても NOP と見なされるようにします。

**2:** 80 ピン /100 ピンデバイスにのみ実装されています。

# PIC18F97J60 ファミリ

## レジスタ 25-7: DEVID1: PIC18F97J60 ファミリのデバイス ID レジスタ 1

R	R	R	R	R	R	R	R
DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0
bit 7							bit 0

### 凡例:

R = 読み出し専用ビット    P = プログラム可能ビット    U = 未実装ビット、「0」として読み出し  
-n = デバイス未プログラム時の値    u = プログラムした状態から不変

bit 7-5    **DEV<2:0>**: デバイス ID ビット

[レジスタ 25-8](#) の一覧参照

bit 4-0    **REV<4:0>**: リビジョン ID ビット

デバイスのリビジョンを示す

## レジスタ 25-8: DEVID2: PIC18F97J60 ファミリのデバイス ID レジスタ 2

R	R	R	R	R	R	R	R
DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3
bit 7							bit 0

### 凡例:

R = 読み出し専用ビット    P = プログラム可能ビット    U = 未実装ビット、「0」として読み出し  
-n = デバイス未プログラム時の値    u = プログラムした状態から不変

bit 7-0    **DEV<10:3>**: デバイス ID ビット:

DEV<10:3> (DEVID2<7:0>)	DEV<2:0> (DEVID1<7:5>)	デバイス名
0001 1000	000	PIC18F66J60
0001 1111	000	PIC18F66J65
0001 1111	001	PIC18F67J60
0001 1000	001	PIC18F86J60
0001 1111	010	PIC18F86J65
0001 1111	011	PIC18F87J60
0001 1000	010	PIC18F96J60
0001 1111	100	PIC18F96J65
0001 1111	101	PIC18F97J60

## 25.2 ウォッチドッグ タイマ (WDT)

PIC18F97J60 ファミリでは、INTRC オシレータによって WDT を駆動します。WDT が有効な場合、クロック源も有効です。WDT の周期は公称 4 ms で、安定度は INTRC オシレータと同じです。

WDT の周期 4 ms を 16 ビット ポストスケーラで逡倍します。WDT ポストスケーラの出力は、コンフィグレーションレジスタ 2H の WDTPS ビットで制御するマルチプレクサによって選択します。出力可能な周期は、4 ms ~ 131.072 s (2.18 min) の範囲です。WDT とポストスケーラは、SLEEP または CLRWDT 命令が実行された場合、またはクロック障害 (プライマリまたは Timer1 オシレータ) が発生した場合クリアされます。

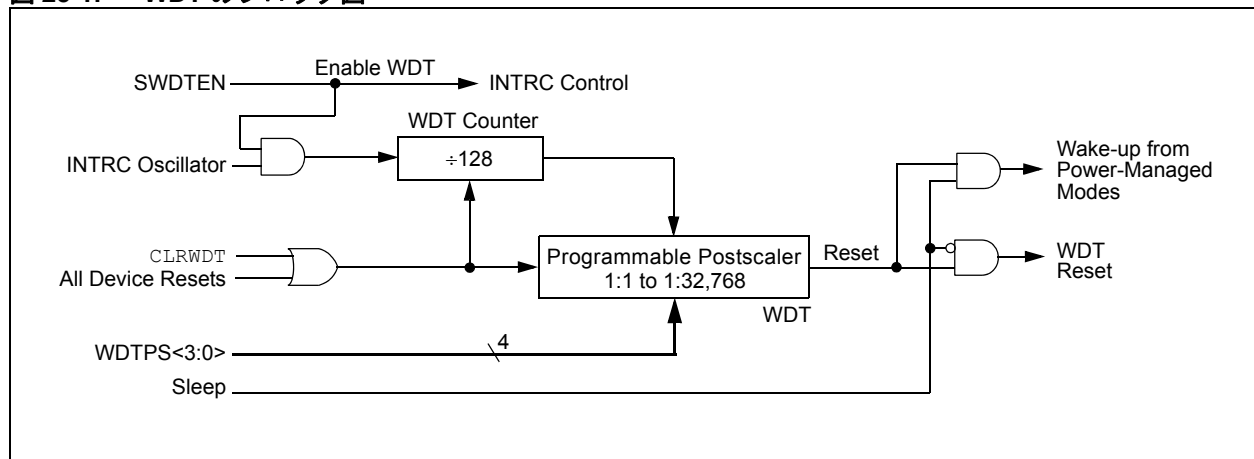
**Note 1:** CLRWDT および SLEEP 命令を実行すると WDT とポストスケーラのカウン트가クリアされます。

**2:** CLRWDT 命令を実行すると、ポストスケーラのカウン트가クリアされます。

### 25.2.1 制御レジスタ

WDTCON レジスタ (レジスタ 25-9) は読み書き可能なレジスタです。SWDTEN ビットは WDT の動作を有効または無効にします。これにより、ソフトウェアが WDTPS コンフィグレーション ビットの設定をオーバーライドして WDT を有効化できるようになります。ただし、コンフィグレーション ビットによって WDT が無効化されている場合に限りです。

図 25-1: WDT のブロック図



# PIC18F97J60 ファミリ

## レジスタ 25-9: WDTCON: ウォッチドッグ タイマ制御レジスタ

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	SWDTEN <sup>(1)</sup>
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」= ビットはセット      「0」= ビットはクリア      x = ビットは未知

bit 7-1      **未実装:** 「0」として読み出し

bit 0      **SWDTEN:** ソフトウェア制御ウォッチドッグ タイマ イネーブルビット<sup>(1)</sup>

1 = ウォッチドッグ タイマを有効にする

0 = ウォッチドッグ タイマを無効にする

**Note 1:** コンフィグレーション ビット WDTEN をセットしている場合、このビットは無効です。

表 25-2: ウォッチドッグ タイマに関連するレジスタのまとめ

レジスタ名	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	リセット値の 記載ページ
RCON	IPEN	—	CM	RI	TO	PD	POR	BOR	70
WDTCON	—	—	—	—	—	—	—	SWDTEN	70

**凡例:** — = 未実装、「0」として読み出されず。網掛けの部分はウォッチドッグ タイマでは使いません。



## 25.3 内蔵電圧レギュレータ

PIC18F97J60 ファミリの全デバイスはコア デジタルロジックに公称 2.5 V を給電しています。このため、これより高い標準電圧 (3.3 V 等) による動作が必要な回路設計では問題が生じる可能性があります。システム設計を容易にするため、PIC18F97J60 ファミリの全デバイスは内蔵レギュレータを備え、VDD でコアロジックを実行できるようにしています。

レギュレータは ENVREG ピンによって制御します。このピンに VDD を接続する事でレギュレータが有効になります。レギュレータはこのピンとは別の VDD ピンからコアに電力を供給します。レギュレータを有効にする場合、VDDCORE/VCAP ピンに低 ESR のフィルタ コンデンサを接続する必要があります ( 図 25-2)。これは、レギュレータの安定性維持に役立ちます。フィルタ コンデンサの推奨値は [セクション 28.3 「DC 特性 : PIC18F97J60 ファミリ \(産業用\)」](#) に記載しています。

ENVREG ピンを VSS に接続すると、レギュレータは無効になります。その場合、コアロジック用の別電源 (公称 2.5 V) をデバイスの VDDCORE/VCAP ピンに接続する必要があります。これにより I/O ピンをより高い電圧 (3.3 V 等) で使えます。または、VDDCORE/VCAP ピンと VDD ピンを接続して、より低い公称電圧で動作させる事もできます。可能な構成は、[図 25-2](#) を参照してください。

### 25.3.1 内蔵レギュレータと BOR

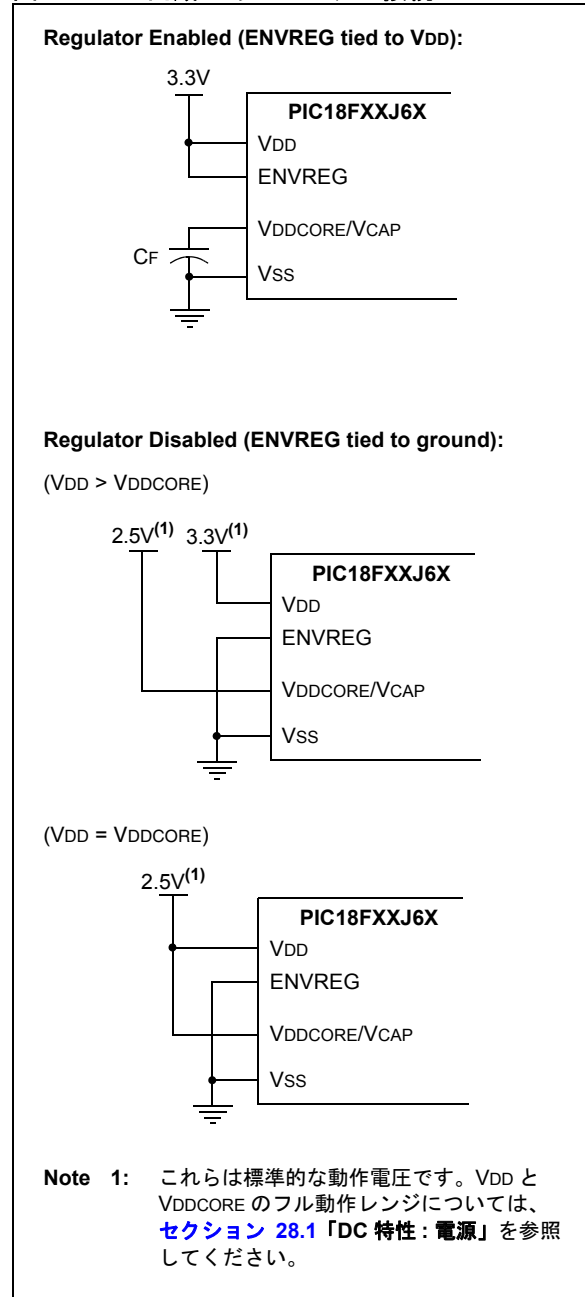
内蔵レギュレータを有効にした場合、PIC18F97J60 ファミリではシンプルなブラウンアウト リセット機能を利用できます。レギュレータに供給される電圧が動作に必要なレベルを下回ると、レギュレータ リセット回路によってブラウンアウト リセット (BOR) が発生します。BOR イベントは、BOR フラグビット (RCON<0>) によってキャプチャされます。

BOR 動作の詳細は、[セクション 5.4 「ブラウンアウト リセット \(BOR\)」](#) と [セクション 5.4.1 「BOR の検出」](#) で説明しています。ブラウンアウト リセットの電圧レベルは、[セクション 28.1 「DC 特性 : 電源電圧 PIC18F97J60 ファミリ \(産業用\)」](#) に記載しています。

### 25.3.2 電源投入時の要件

内蔵レギュレータは、デバイスの電源投入時の要件を満たすように設計されています。レギュレータを使わない場合、電源投入時の要件を逸脱しないように注意する必要があります。電源投入中は、VDDCORE が VDD + 0.3 V を上回らないようにする必要があります。

図 25-2: 内蔵レギュレータの接続



## 25.4 2段階起動

2段階起動を使うと、マイクロコントローラはプライマリクロック源が利用可能になるまで INTRC オシレータをクロック源として使います。これにより、コード実行開始までの遅れを短縮できます。この機能は IESO コンフィグレーション ビットをセットすると有効になります。

2段階起動を有効にすべきなのは、プライマリオシレータモードが HS または HSPLL (水晶振動子) モードの場合のみです。EC および ECPLL モードではオシレータ起動タイマによる遅延が不要のため、2段階起動は無効にします。

2段階起動を有効にすると、リセット時とスリープからの復帰時に、パワーオンリセット後のパワーアップタイマのタイムアウト後、デバイスはクロック源として内部オシレータブロックを使って動作します。これにより、プライマリオシレータの安定動作を待たずに (OST 実行中に)、コード実行を開始できます。OST がタイムアウトすると、デバイスは自動的に PRI\_RUN モードに切り換わります。

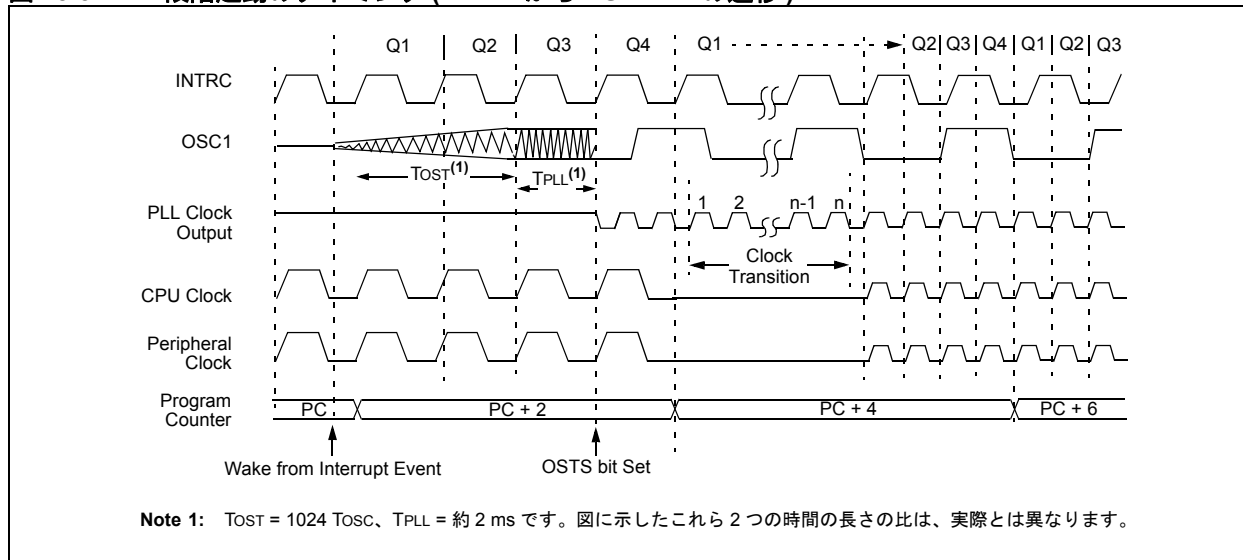
他の電力管理モードでは2段階起動は使いません。プライマリクロック源が使用可能になるまで、デバイスは現在選択されているクロック源によって駆動されます。IESO ビットの設定は無視されます。

### 25.4.1 2段階起動を使う場合の注意事項

2段階起動中に INTRC オシレータで動作している間も、デバイスは SLEEP 命令 ( [セクション 4.1.4「複数の SLEEP 命令」](#) 参照 ) を含む電力管理モード移行のための通常のコマンドシーケンスに従います。これは、OST がタイムアウトする前でも、ユーザコードによる SCS<1:0> ビットの設定変更または SLEEP 命令の発行が可能であることを意味します。従って、デバイスを短時間だけ復帰させて定期的なタスクを実行させ、プライマリオシレータによる動作に切り換わる前にスリープに戻るといった使い方も可能です。

ユーザコードで OST ビット (OSCCON<3>) のステータスを調べ、プライマリクロック源が現在デバイスにクロックを供給しているかどうかを確認する事もできます。このビットがセットされていれば、プライマリオシレータがクロックを供給中です。クリアされている場合、リセットまたはスリープからの復帰中で、内部オシレータブロックがクロックを供給しています。

図 25-3: 2段階起動のタイミング (INTRC から HSPLL への移行)

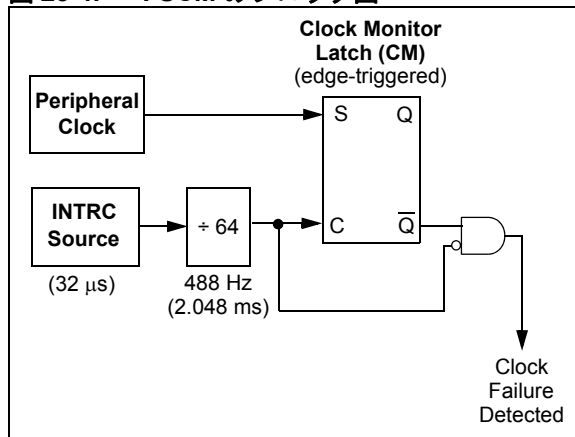


## 25.5 フェイルセーフ クロックモニタ

フェイルセーフ クロックモニタ (FSCM) を使うと、外部オシレータに障害が発生した場合でも、自動的にデバイスクロックを内部オシレータ ブロックに切り換えて動作を継続できます。FSCM 機能は、FCMEN コンフィグレーション ビットをセットすると有効になります。

FSCM を有効にすると INTRC オシレータが常時動作し、周辺モジュールへのクロックを監視して、クロック障害が発生した場合にバックアップ クロックを供給します。クロックの監視 (図 25-4 参照) は、INTRC 出力を 64 分周したサンプルクロック信号を生成する事で行います。これにより、FSCM サンプルクロックのエッジ間に、周辺モジュールのクロックエッジが発生する十分な時間を確保できます。周辺デバイスクロックとサンプルクロックは、クロックモニタ (CM) ラッチに入力されます。CM はデバイスクロック源の立ち下がりエッジでセットされますが、サンプルクロックの立ち上がりエッジでクリアされます。

図 25-4: FSCM のブロック図



クロック障害は、サンプルクロックの立ち下がりエッジでテストします。CM がセットされた状態でサンプルクロックが立ち下がった場合、クロック障害が検出されています (図 25-5)。これにより、以下が実行されます。

- FSCM が OSCFIF (PIR2<7>) ビットをセットしてオシレータ障害割り込みを発生させる
- デバイスクロック源を内部オシレータ ブロックに切り換える (切り換え後のクロック源を示す値に OSCCON が更新される事はありません — これはフェイルセーフ条件であるためです)
- WDT をリセットする

切り換え中に供給される内部オシレータ ブロックからのポストスケアラ周波数では、高いタイミング精度が要求されるアプリケーションの場合、精度が十分でない可能性があります。その場合、別のクロック設定を選択し、別の電力管理モードに移行する方が望ましいでしょう。それには部分的な回復を試みるか、アプリケーションでシャットダウン処理を実行します。詳細はセクション 4.1.4「複数の SLEEP 命令」とセクション 25.4.1「2 段階起動を使う場合の注意事項」を参照してください。

FSCM は、プライマリまたはセカンダリ クロック源の障害のみ検出します。内部オシレータ ブロックに障害が発生しても一切検出されず、対応もできません。

### 25.5.1 FSCM とウォッチドッグ タイマ

FSCM と WDT は、いずれも INTRC オシレータからクロックを供給されます。WDT は独立した分周器とカウンタによって動作するため、WDT を無効にしても、FSCM が有効な場合、INTRC オシレータの動作には何の影響も及ぼしません。

前述の通り、クロック障害が検出されると、クロック源が INTRC クロックに切り換わります。これによりコード実行速度が著しく低下する場合があります。WDT を小さなプリスケール値で有効にしている場合、クロック速度の低下によって WDT タイムアウトが発生し、それがデバイスリセットを引き起こす可能性があります。このため、フェイルセーフ クロック イベントは WDT とポストスケアラもリセットします。これにより実行速度が変化した時点からタイミングを計測しはじめる事ができ、誤ったタイムアウトが起きにくくします。

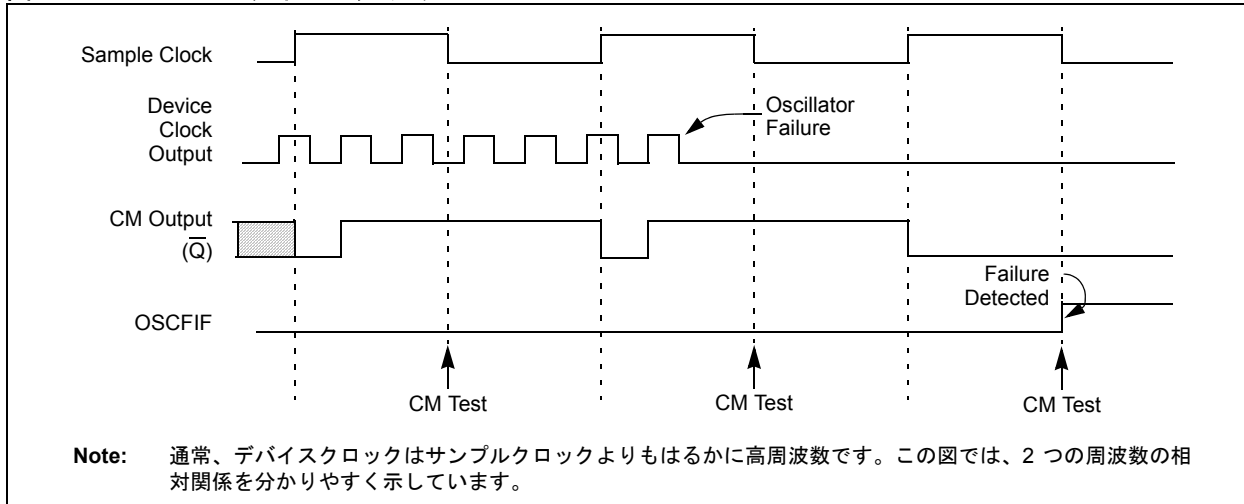
### 25.5.2 フェイルセーフ動作の終了

フェイルセーフ条件は、デバイスリセットまたは電力管理モードへの移行によって終了します。リセット時、コンフィグレーション レジスタ 2H で指定されているプライマリ クロック源が起動します (OST または PLL タイマ等、オシレータモード別の起動遅延が加わります)。プライマリ クロック源が使用可能になるまで、INTRC オシレータがデバイスクロックを供給します (2 段階起動と同様)。その後、クロック源はプライマリ クロックに切り換わります (切り換わると OSCCON レジスタの OSTS ビットがセットされる)。フェイルセーフ クロックモニタは周辺クロックの監視を再開します。

起動中にプライマリ クロック源が全く使用可能にならない場合があります。その場合、INTRC オシレータがクロックを供給します。電力管理モードに移行するまで OSCCON レジスタはリセット状態のままです。

# PIC18F97J60 ファミリ

図 25-5: FSCM のタイミング チャート



## 25.5.3 電力管理モードにおける FSCM 割り込み

電力管理モードに移行すると、クロック マルチプレクサは OSCCON レジスタで指定されているクロック源を選択します。この電力管理モード用のクロック源に対するフェイルセーフ モニタが、電力管理モードで再開します。

電力管理モードで動作中にオシレータ障害が発生した場合、対応はオシレータ障害割り込みが有効かどうかで異なります。割り込みが有効の場合 (OSCFIF = 1)、コード実行のクロックは INTRC マルチプレクサから供給されます。障害が発生したクロック源に自動的に復帰する事はありません。

割り込みが無効の場合、アイドル中の割り込みによって、CPU は INTRC のクロック供給で命令実行を開始します。

## 25.5.4 POR またはスリープからの復帰

FSCM はデバイスのパワーオン リセット (POR) または低消費電力スリープの終了後のあらゆる時点におけるオシレータ障害を検出できるように設計されています。プライマリ デバイスクロックが EC または INTRC の場合、これらのイベントの直後から監視を開始できます。

HS または HSPLL モードの場合、状況は若干異なります。オシレータは FSCM サンプルクロック時間よりも大幅に長い起動時間を必要とする可能性があるため、クロック障害の誤検出が発生する恐れがあります。これを防ぐために、デバイスクロックとして自動的に内部オシレータ ブロックが設定され、プライマリ クロックが安定するまで (OST と PLL タイマがタイムアウトするまで) 動作します。これは 2 段階起動モードと同じ動作です。プライマリ クロックが安定すると、INTRC は FSCM 用クロック源としての役割に戻ります。

**Note:** POR またはスリープからの復帰時における誤ったオシレータ障害割り込みを防止するロジックは、これらのイベント以降の全てのオシレータ起動障害の検出も妨げてしまいます。これを回避するには、OSTS ビットを監視してタイミング ルーチンを使い、オシレータの起動に時間がかかりすぎているかを判断します。その場合も、オシレータ障害割り込みフラグはセットされません。

**セクション 25.4.1「2 段階起動を使う場合の注意事項」**で説明した通り、他のクロック設定を選択し、プライマリ クロックが安定するまで待つ間に代替の電力管理モードに移行する事も可能です。新しい電力管理モードを選択すると、プライマリ クロックは無効になります。

## 25.6 プログラムの検証とコード保護

PIC18F97J60 ファミリの全デバイスは、内蔵プログラムメモリ空間を単一のブロックとして扱います。このブロックに対するコード保護は、コンフィグレーションビット CP0 で制御します。このビットは、プログラムメモリ空間に対する外部からの読み書きを禁止しますが、通常の実行モードには直接影響を与えません。

### 25.6.1 コンフィグレーションレジスタの保護

コンフィグレーションレジスタは、不適切な変更または読み出しから 2 つの方法で保護されています。第 1 の保護は、電源投入後にビットがプログラムされた後の変更を防ぐ、コンフィグレーションビットのライトワンス機能です。予測不能なイベントに対する保護対策として、ESD イベント等、個々のセルレベルのデータ破壊によるコンフィグレーションビットの変化はパリティエラーを発生させ、これがデバイスリセットをトリガします。

コンフィグレーションレジスタのデータは、プログラムメモリ内のフラッシュコンフィグレーションワードから生成します。CP0 ビットがプログラム状態(クリア)ならば、デバイスコンフィグレーションのソースデータも結果的に保護されます。

## 25.7 インサーキットシリアルプログラミング

PIC18F97J60 ファミリのマイクロコントローラは、最終アプリケーション回路に実装した状態でのシリアルプログラミングが可能です。これは、クロック / データの 2 本のライン、電源 / グランド / 書き込み電圧の 3 本のラインのみで簡単に利用できます。この機能を利用すると、未プログラムデバイスを実装したボードを製造し、製品の出荷直前にマイクロコントローラをプログラムできます。また、最新ファームウェアまたはカスタマイズしたファームウェアをプログラムすることもできます。

## 25.8 インサーキットデバッグ

`DEBUG` コンフィグレーションビットに「0」を書き込むと、インサーキットデバッグ機能が有効になります。この機能を利用すると、MPLAB<sup>®</sup> IDE を使って簡単なデバッグを実行できます。マイクロコントローラでこの機能を有効にすると、一部の汎用リソースが使えなくなります。表 25-3 に、バックグラウンドデバッグで必要になるリソースを示します。

表 25-3: デバッグで使うリソース

I/O ピン:	RB6、RB7
スタック:	2 段
プログラムメモリ:	512 バイト
データメモリ:	10 バイト

# PIC18F97J60 ファミリ

---

NOTES:

## 26.0 命令セットの概要

PIC18F97J60 ファミリのデバイスは、75 個の PIC18 コア標準命令セットに加えて、再帰コードやソフトウェア スタックを利用するコード向けに最適化された 8 つの新しい拡張命令をサポートします。拡張命令セットについては、本セクション内で後述します。

### 26.1 標準命令セット

PIC18 標準命令セットは以前の PIC<sup>®</sup> MCU 命令セットから大幅に強化されていますが、これらの PIC MCU から容易に移行できるように互換性も維持しています。ほとんどの命令は 1 プログラムメモリワード (16 ビット) ですが、2 プログラムメモリワードを必要とする命令が 4 つあります。

1 ワード命令は、命令の種類を表す 1 つのオペコードと、命令の動作を詳しく指定する 1 つまたは複数のオペランドから成り、全体で 16 ビットワードです。

この命令セットは直交性が高く、以下の 4 つの基本カテゴリに分類されます。

- バイト指向命令
- ビット指向命令
- リテラル命令
- 制御命令

表 26-2 に、PIC18 命令セットのバイト指向命令、ビット指向命令、リテラル命令、制御命令を示します。

表 26-1 に、オペコードフィールドの説明を示します。

バイト指向命令の大部分は、以下の 3 つのオペランドを使います。

1. ファイルレジスタ (「f」と表記)
2. 結果格納先 (「d」と表記)
3. アクセス先メモリ (「a」と表記)

ファイルレジスタの指定文字「f」には、命令で使うファイルレジスタを指定します。結果格納先の指定文字「d」は、演算の結果を格納する場所を指定します。「d」= 0 の場合は演算結果を WREG レジスタに格納します。「d」= 1 の場合は演算結果を「f」で指定したファイルレジスタに格納します。

ビット指向命令は、いずれも以下の 3 つのオペランドを使います。

1. ファイルレジスタ (「f」と表記)
2. ファイルレジスタ内のビット (「b」と表記)
3. アクセス先メモリ (「a」と表記)

ビットフィールドの指定文字「b」は演算の対象となるビットの番号を指定し、ファイルレジスタの指定文字「f」はそのビットが存在するファイルの番号を表します。

リテラル命令で使うオペランドには、以下の種類があります。

- ファイルレジスタに読み込むリテラル値 (「k」と表記)
- リテラル値を読み込む FSR レジスタ (「f」と表記)
- オペランド不要 (「-」と表記)

制御命令で使うオペランドには、以下の種類があります。

- プログラムメモリのアドレス (「n」と表記)
- CALL または RETURN 命令のモード (「s」と表記)
- テーブル読み書き命令のモード (「m」と表記)
- オペランド不要 (「-」と表記)

命令は基本的には 1 ワードですが、2 ワード命令が 4 つだけあります。これらの命令は、32 ビットの情報を収めるため 2 ワードです。2 ワード目の上位 4 ビットは「1」です。このため、2 ワード目のみを単独の命令として実行すると、NOP として実行されます。

1 ワード命令は全て 1 命令サイクルで実行されますが、条件テストが真の場合、または命令の実行結果によりプログラムカウンタが変わる場合はこの限りではありません。これらの場合には、NOP 命令サイクルが追加されるため、実行には 2 命令サイクルを要します。

2 ワード命令の実行には 2 命令サイクルが必要です。

1 命令サイクルは、オシレータの 4 周期に相当します。従って、オシレータの周波数が 4 MHz ならば、命令実行時間は標準で 1  $\mu$ s です。条件付きテストの結果が真の場合、または命令実行の結果プログラムカウンタが変化した場合の命令実行時間は 2  $\mu$ s です。2 ワードの分岐命令に必要な時間は 3  $\mu$ s です (条件付きテストの結果が真の場合)。

図 26-1 に、命令の全般的な形式を示します。この例では、16 進数を全て「nnh」と表記しています。

表 26-2 に、マイクロチップ社の MPASM<sup>™</sup> アセンブラで認識できる標準命令セットの一覧を示します。

各命令については、[セクション 26.1.1「標準命令セット」](#)で詳しく説明します。

# PIC18F97J60 ファミリ

表 26-1: オペコード フィールドの説明

フィールド	説明
a	RAM アクセスビット: a = 0: アクセス RAM 内の RAM アドレス (BSR レジスタは無視) a = 1: BSR レジスタで RAM バンクを指定
bbb	8 ビット ファイルレジスタ内のビットアドレス (0 ~ 7)
BSR	バンクセレクト レジスタ、現在の RAM バンクを選択
C, DC, Z, OV, N	ALU ステータスビット: <b>C</b> (CARRY)、 <b>DC</b> (DIGIT CARRY)、 <b>Z</b> (ZERO)、 <b>OV</b> (OVERFLOW)、 <b>N</b> (NEGATIVE)
d	格納先選択ビット: d = 0: 結果を WREG に格納 d = 1: 結果をファイルレジスタ f に格納
dest	格納先: WREG レジスタまたは指定したレジスタファイルのアドレス
f	8 ビット レジスタファイルのアドレス (00h ~ FFh) または 2 ビットの FSR 指定文字 (0h ~ 3h)
f <sub>s</sub>	送信元の 12 ビット レジスタファイルのアドレス (000h ~ FFFh)
f <sub>d</sub>	格納先の 12 ビット レジスタファイルのアドレス (000h ~ FFFh)
GIE	グローバル割り込みイネーブルビット
k	リテラル フィールド、定数データ、ラベル (8/12/20 ビットのいずれか)
label	ラベル名
mm	テーブル読み書き命令の TBLPTR レジスタのモード テーブル読み書き命令にのみ使用 レジスタ変更なし (テーブル読み書きの場合の TBLPTR 等) * 命令実行後にレジスタをインクリメントする (テーブル読み書きの場合の TBLPTR 等) *+ 命令実行後にレジスタをデクリメントする (テーブル読み書きの場合の TBLPTR 等) *- 命令実行前にレジスタをインクリメントする (テーブル読み書きの場合の TBLPTR 等)
n	相対分岐命令の相対アドレス (2 の補数) または CALL/BRANCH および RETURN 命令の直接アドレス
PC	プログラム カウンタ
PCL	プログラム カウンタ下位バイト
PCH	プログラム カウンタ中位バイト
PCLATH	プログラム カウンタ中位バイト用ラッチ
PCLATU	プログラム カウンタ最上位バイト用ラッチ
$\overline{PD}$	パワーダウン ビット
PRODH	乗算命令の結果の上位バイト
PRODL	乗算命令の結果の下位バイト
s	高速コール/リターン モード選択ビット: s = 0: シャドールレジスタへの保存、復元を行わない s = 1: 特定レジスタの値をシャドールレジスタに保存、復元する (高速モード)
TBLPTR	21 ビット テーブルポインタ (プログラムメモリのアドレスを指し示す)
TABLAT	8 ビット テーブルラッチ
TO	タイムアウト ビット
TOS	Top-of-Stack
u	未使用または変化なし
WDT	ウォッチドッグ タイマ
WREG	ワーキング レジスタ (アキュムレータ)
x	ドントケア ('0' または '1')、アセンブラは x = 0 でコードを生成 (マイクロチップ社のソフトウェア ツール全般との互換性のために、この値を推奨します。)
z <sub>s</sub>	レジスタファイル (送信元) の間接アドレッシングに使う 7 ビットのオフセット値
z <sub>d</sub>	レジスタファイル (格納先) の間接アドレッシングに使う 7 ビットのオフセット値
{ }	オプションの引数
[text]	インデックス付きアドレス
(text)	text の内容
[expr] <n>	ポインタ expr が指し示すレジスタの bit n
→	割り当て先
< >	レジスタのビットフィールド
ε	左項は右項に属する
italics	ユーザ定義語 (Courier New フォント)



図 26-1: 命令の一般的な形式

Byte-oriented file register operations	Example Instruction																																																																																				
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 5%; text-align: right;">15</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">10</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">9</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">8</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">7</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">0</td> </tr> <tr> <td colspan="2">OPCODE</td> <td>d</td> <td>a</td> <td colspan="7">f (FILE #)</td> </tr> </table> <p>d = 0 for result destination to be WREG register  d = 1 for result destination to be file register (f)  a = 0 to force Access Bank  a = 1 for BSR to select bank  f = 8-bit file register address</p>	15		10		9		8		7		0	OPCODE		d	a	f (FILE #)							ADDWF MYREG, W, B																																																														
15		10		9		8		7		0																																																																											
OPCODE		d	a	f (FILE #)																																																																																	
<p><b>Byte to Byte move operations (2-word)</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 5%; text-align: right;">15</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">12</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">11</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">0</td> </tr> <tr> <td colspan="2">OPCODE</td> <td colspan="5">f (Source FILE #)</td> </tr> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 5%; text-align: right;">15</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">12</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">11</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">0</td> </tr> <tr> <td colspan="2">1111</td> <td colspan="5">f (Destination FILE #)</td> </tr> </table> <p>f = 12-bit file register address</p>	15		12		11		0	OPCODE		f (Source FILE #)					15		12		11		0	1111		f (Destination FILE #)					MOVFF MYREG1, MYREG2																																																								
15		12		11		0																																																																															
OPCODE		f (Source FILE #)																																																																																			
15		12		11		0																																																																															
1111		f (Destination FILE #)																																																																																			
<p><b>Bit-oriented file register operations</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 5%; text-align: right;">15</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">12</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">11</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">9</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">8</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">7</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">0</td> </tr> <tr> <td colspan="2">OPCODE</td> <td colspan="2">b (BIT #)</td> <td>a</td> <td colspan="7">f (FILE #)</td> </tr> </table> <p>b = 3-bit position of bit in file register (f)  a = 0 to force Access Bank  a = 1 for BSR to select bank  f = 8-bit file register address</p>	15		12		11		9		8		7		0	OPCODE		b (BIT #)		a	f (FILE #)							BSF MYREG, bit, B																																																											
15		12		11		9		8		7		0																																																																									
OPCODE		b (BIT #)		a	f (FILE #)																																																																																
<p><b>Literal operations</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 5%; text-align: right;">15</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">8</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">7</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">0</td> </tr> <tr> <td colspan="2">OPCODE</td> <td colspan="5">k (literal)</td> </tr> </table> <p>k = 8-bit immediate value</p>	15		8		7		0	OPCODE		k (literal)					MOVLW 7Fh																																																																						
15		8		7		0																																																																															
OPCODE		k (literal)																																																																																			
<p><b>Control operations</b></p> <p><b>CALL, GOTO and Branch operations</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 5%; text-align: right;">15</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">8</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">7</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">0</td> </tr> <tr> <td colspan="2">OPCODE</td> <td colspan="5">n&lt;7:0&gt; (literal)</td> </tr> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 5%; text-align: right;">15</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">12</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">11</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">0</td> </tr> <tr> <td colspan="2">1111</td> <td colspan="5">n&lt;19:8&gt; (literal)</td> </tr> </table> <p>n = 20-bit immediate value</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 5%; text-align: right;">15</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">8</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">7</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">0</td> </tr> <tr> <td colspan="2">OPCODE</td> <td>S</td> <td colspan="4">n&lt;7:0&gt; (literal)</td> </tr> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 5%; text-align: right;">15</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">12</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">11</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">0</td> </tr> <tr> <td colspan="2">1111</td> <td colspan="5">n&lt;19:8&gt; (literal)</td> </tr> </table> <p>S = Fast bit</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 5%; text-align: right;">15</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">11</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">10</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">0</td> </tr> <tr> <td colspan="2">OPCODE</td> <td colspan="5">n&lt;10:0&gt; (literal)</td> </tr> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 5%; text-align: right;">15</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">8</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">7</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: right;">0</td> </tr> <tr> <td colspan="2">OPCODE</td> <td colspan="5">n&lt;7:0&gt; (literal)</td> </tr> </table>	15		8		7		0	OPCODE		n<7:0> (literal)					15		12		11		0	1111		n<19:8> (literal)					15		8		7		0	OPCODE		S	n<7:0> (literal)				15		12		11		0	1111		n<19:8> (literal)					15		11		10		0	OPCODE		n<10:0> (literal)					15		8		7		0	OPCODE		n<7:0> (literal)					<p>GOTO Label</p> <p>CALL MYFUNC</p> <p>BRA MYFUNC</p> <p>BC MYFUNC</p>
15		8		7		0																																																																															
OPCODE		n<7:0> (literal)																																																																																			
15		12		11		0																																																																															
1111		n<19:8> (literal)																																																																																			
15		8		7		0																																																																															
OPCODE		S	n<7:0> (literal)																																																																																		
15		12		11		0																																																																															
1111		n<19:8> (literal)																																																																																			
15		11		10		0																																																																															
OPCODE		n<10:0> (literal)																																																																																			
15		8		7		0																																																																															
OPCODE		n<7:0> (literal)																																																																																			

# PIC18F97J60 ファミリ

表 26-2: PIC18F97J60 ファミリ 命令セット

ニーモニック、オペランド	説明	サイクル	16 ビット命令ワード				影響を受けるステータス	Notes
			MSb		LSb			
<b>バイト指向命令</b>								
ADDWF f, d, a	Add WREG and f	1	0010	01da	ffff	ffff	C, DC, Z, OV, N	1, 2
ADDWFC f, d, a	Add WREG and Carry bit to f	1	0010	00da	ffff	ffff	C, DC, Z, OV, N	1, 2
ANDWF f, d, a	AND WREG with f	1	0001	01da	ffff	ffff	Z, N	1, 2
CLRF f, a	Clear f	1	0110	101a	ffff	ffff	Z	2
COMF f, d, a	Complement f	1	0001	11da	ffff	ffff	Z, N	1, 2
CPFSEQ f, a	Compare f with WREG, Skip =	1 (2 または 3)	0110	001a	ffff	ffff	なし	4
CPFSGT f, a	Compare f with WREG, Skip >	1 (2 または 3)	0110	010a	ffff	ffff	なし	4
CPFSLT f, a	Compare f with WREG, Skip <	1 (2 または 3)	0110	000a	ffff	ffff	なし	1, 2
DECF f, d, a	Decrement f	1	0000	01da	ffff	ffff	C, DC, Z, OV, N	1, 2, 3, 4
DECFSZ f, d, a	Decrement f, Skip if 0	1 (2 または 3)	0010	11da	ffff	ffff	なし	1, 2, 3, 4
DCFSNZ f, d, a	Decrement f, Skip if Not 0	1 (2 または 3)	0100	11da	ffff	ffff	なし	1, 2
INCF f, d, a	Increment f	1	0010	10da	ffff	ffff	C, DC, Z, OV, N	1, 2, 3, 4
INCFSZ f, d, a	Increment f, Skip if 0	1 (2 または 3)	0011	11da	ffff	ffff	なし	4
INFSNZ f, d, a	Increment f, Skip if Not 0	1 (2 または 3)	0100	10da	ffff	ffff	なし	1, 2
IORWF f, d, a	Inclusive OR WREG with f	1	0001	00da	ffff	ffff	Z, N	1, 2
MOVF f, d, a	Move f	1	0101	00da	ffff	ffff	Z, N	1
MOVFF f <sub>s</sub> , f <sub>d</sub>	Move f <sub>s</sub> (source) to f <sub>d</sub> (destination)	2	1100	ffff	ffff	ffff	なし	
MOVWF f, a	Move WREG to f	1	0110	111a	ffff	ffff	なし	
MULWF f, a	Multiply WREG with f	1	0000	001a	ffff	ffff	なし	1, 2
NEGF f, a	Negate f	1	0110	110a	ffff	ffff	C, DC, Z, OV, N	
RLCF f, d, a	Rotate Left f through Carry	1	0011	01da	ffff	ffff	C, Z, N	1, 2
RLNCF f, d, a	Rotate Left f (No Carry)	1	0100	01da	ffff	ffff	Z, N	
RRCF f, d, a	Rotate Right f through Carry	1	0011	00da	ffff	ffff	C, Z, N	
RRNCF f, d, a	Rotate Right f (No Carry)	1	0100	00da	ffff	ffff	Z, N	
SETF f, a	Set f	1	0110	100a	ffff	ffff	なし	1, 2
SUBFWB f, d, a	Subtract f from WREG with Borrow	1	0101	01da	ffff	ffff	C, DC, Z, OV, N	
SUBWF f, d, a	Subtract WREG from f	1	0101	11da	ffff	ffff	C, DC, Z, OV, N	1, 2
SUBWFB f, d, a	Subtract WREG from f with Borrow	1	0101	10da	ffff	ffff	C, DC, Z, OV, N	
SWAPF f, d, a	Swap Nibbles in f	1	0011	10da	ffff	ffff	なし	4
TSTFSZ f, a	Test f, Skip if 0	1 (2 または 3)	0110	011a	ffff	ffff	なし	1, 2
XORWF f, d, a	Exclusive OR WREG with f	1	0001	10da	ffff	ffff	Z, N	

- Note 1:** PORT レジスタの値を同じ PORT レジスタによって変更する場合 (例: MOVF PORTB, 1, 0)、これらのピンの入力レベルが使われず。例えば、入力ピンのデータラッチが「1」で、そのピンが外部デバイスによって Low に駆動されている場合、データ「0」が書き戻されます。
- 2:** この命令を TMR0 レジスタに対して実行した場合 (かつ、「d」の指定が可能な場合に d = 1 の時)、TMR0 にプリスケアラが割り当てられているとプリスケアラもクリアされます。
- 3:** プログラムカウンタ (PC) が変更された場合、または条件付きテストの結果が真の場合、命令実行には 2 サイクルが必要です。2 サイクル目は、NOP として実行されます。
- 4:** 一部の命令は、2 ワード命令です。これらの命令の 2 ワード目は、命令の 1 ワード目によって 2 ワード目の 16 ビットに埋め込まれた情報を読み出す場合を除き、NOP として実行されます。これにより、全てのプログラムメモリアドレスで命令の有効性が維持されます。

# PIC18F97J60 ファミリ

表 26-2: PIC18F97J60 ファミリ 命令セット ( 続き )

ニーモニック、オペランド	説明	サイクル	16 ビット命令ワード				影響を受けるステータス	Notes	
			MSb	LSb					
<b>ビット指向命令</b>									
BCF	f, b, a	Bit Clear f	1	1001	bbba	ffff	ffff	なし	1, 2
BSF	f, b, a	Bit Set f	1	1000	bbba	ffff	ffff	なし	1, 2
BTFS	f, b, a	Bit Test f, Skip if Clear	1 (2 または 3)	1011	bbba	ffff	ffff	なし	3, 4
BTFS	f, b, a	Bit Test f, Skip if Set	1 (2 または 3)	1010	bbba	ffff	ffff	なし	3, 4
BTG	f, b, a	Bit Toggle f	1	0111	bbba	ffff	ffff	なし	1, 2
<b>制御命令</b>									
BC	n	Branch if Carry	1 (2)	1110	0010	nnnn	nnnn	なし	4
BN	n	Branch if Negative	1 (2)	1110	0110	nnnn	nnnn	なし	
BNC	n	Branch if Not Carry	1 (2)	1110	0011	nnnn	nnnn	なし	
BNN	n	Branch if Not Negative	1 (2)	1110	0111	nnnn	nnnn	なし	
BNOV	n	Branch if Not Overflow	1 (2)	1110	0101	nnnn	nnnn	なし	
BNZ	n	Branch if Not Zero	1 (2)	1110	0001	nnnn	nnnn	なし	
BOV	n	Branch if Overflow	1 (2)	1110	0100	nnnn	nnnn	なし	
BRA	n	Branch Unconditionally	2	1101	0nnn	nnnn	nnnn	なし	
BZ	n	Branch if Zero	1 (2)	1110	0000	nnnn	nnnn	なし	
CALL	n, s	Call Subroutine 1st word 2nd word	2	1110	110s	kkkk	kkkk	なし	
CLRWDT	—	Clear Watchdog Timer	1	0000	0000	0000	0100	$\overline{TO}$ , $\overline{PD}$	
DAW	—	Decimal Adjust WREG	1	0000	0000	0000	0111	C	
GOTO	n	Go to Address 1st word 2nd word	2	1110	1111	kkkk	kkkk	なし	
NOP	—	NOP	1	0000	0000	0000	0000	なし	
NOP	—	NOP	1	1111	xxxx	xxxx	xxxx	なし	
POP	—	Pop Top of Return Stack (TOS)	1	0000	0000	0000	0110	なし	
PUSH	—	Push Top of Return Stack (TOS)	1	0000	0000	0000	0101	なし	
RCALL	n	Relative Call	2	1101	1nnn	nnnn	nnnn	なし	
RESET	—	Software Device Reset	1	0000	0000	1111	1111	All	
RETFIE	s	Return from Interrupt Enable	2	0000	0000	0001	000s	GIE/GIEH, PEIE/GIEL	
RETLW	k	Return with Literal in WREG	2	0000	1100	kkkk	kkkk	なし	
RETURN	s	Return from Subroutine	2	0000	0000	0001	001s	なし	
SLEEP	—	Go into Standby mode	1	0000	0000	0000	0011	$\overline{TO}$ , $\overline{PD}$	

- Note 1:** PORT レジスタの値を同じ PORT レジスタによって変更する場合 (例: MOVF PORTB, 1, 0)、これらのピンの入力レベルが使われます。例えば、入力ピンのデータラッチが「1」で、そのピンが外部デバイスによって Low に駆動されている場合、データ「0」が書き戻されます。
- 2:** この命令を TMR0 レジスタに対して実行した場合 (かつ、「d」の指定が可能な場合に d = 1 の時)、TMR0 にプリスケアラが割り当てられているとプリスケアラもクリアされます。
- 3:** プログラム カウンタ (PC) が変更された場合、または条件付きテストの結果が真の場合、命令実行には 2 サイクルが必要です。2 サイクル目は、NOP として実行されます。
- 4:** 一部の命令は、2 ワード命令です。これらの命令の 2 ワード目は、命令の 1 ワード目によって 2 ワード目の 16 ビットに埋め込まれた情報を読み出す場合を除き、NOP として実行されます。これにより、全てのプログラムメモリアドレスで命令の有効性が維持されます。

# PIC18F97J60 ファミリ

表 26-2: PIC18F97J60 ファミリ 命令セット ( 続き )

ニーモニック、オペランド	説明	サイクル	16 ビット命令ワード				影響を受けるステータス	Notes
			MSb	LSb				
<b>リテラル命令</b>								
ADDLW k	Add Literal and WREG	1	0000	1111	kkkk	kkkk	C, DC, Z, OV, N	
ANDLW k	AND Literal with WREG	1	0000	1011	kkkk	kkkk	Z, N	
IORLW k	Inclusive OR Literal with WREG	1	0000	1001	kkkk	kkkk	Z, N	
LFSR f, k	Move Literal (12-bit) 2nd word to FSR(f) 1st word	2	1110	1110	00ff	kkkk	なし	
MOVLB k	Move Literal to BSR<3:0>	1	0000	0001	0000	kkkk	なし	
MOVLW k	Move Literal to WREG	1	0000	1110	kkkk	kkkk	なし	
MULLW k	Multiply Literal with WREG	1	0000	1101	kkkk	kkkk	なし	
RETLW k	Return with Literal in WREG	2	0000	1100	kkkk	kkkk	なし	
SUBLW k	Subtract WREG from Literal	1	0000	1000	kkkk	kkkk	C, DC, Z, OV, N	
XORLW k	Exclusive OR Literal with WREG	1	0000	1010	kkkk	kkkk	Z, N	
<b>データメモリ ↔ プログラムメモリ命令</b>								
TBLRD*	Table Read	2	0000	0000	0000	1000	なし	
TBLRD*+	Table Read with Post-Increment		0000	0000	0000	1001	なし	
TBLRD*-	Table Read with Post-Decrement		0000	0000	0000	1010	なし	
TBLRD*+	Table Read with Pre-Increment		0000	0000	0000	1011	なし	
TBLWT*	Table Write	2	0000	0000	0000	1100	なし	
TBLWT*+	Table Write with Post-Increment		0000	0000	0000	1101	なし	
TBLWT*-	Table Write with Post-Decrement		0000	0000	0000	1110	なし	
TBLWT*+	Table Write with Pre-Increment		0000	0000	0000	1111	なし	

- Note 1:** PORT レジスタの値を同じ PORT レジスタによって変更する場合 (例: MOVF PORTB, 1, 0)、これらのピンの入力レベルが使われます。例えば、入力ピンのデータラッチが「1」で、そのピンが外部デバイスによって Low に駆動されている場合、データ「0」が書き戻されます。
- 2:** この命令を TMR0 レジスタに対して実行した場合 (かつ、「d」の指定が可能な場合に d = 1 の時)、TMR0 にプリスケアラが割り当てられているとプリスケアラもクリアされます。
- 3:** プログラムカウンタ (PC) が変更された場合、または条件付きテストの結果が真の場合、命令実行には 2 サイクルが必要です。2 サイクル目は、NOP として実行されます。
- 4:** 一部の命令は、2 ワード命令です。これらの命令の 2 ワード目は、命令の 1 ワード目によって 2 ワード目の 16 ビットに埋め込まれた情報を読み出す場合を除き、NOP として実行されます。これにより、全てのプログラムメモリアドレスで命令の有効性が維持されます。

## 26.1.1 標準命令セット

ADDLW	ADD Literal to W								
構文:	ADDLW k								
オペランド:	$0 \leq k \leq 255$								
動作:	$(W) + k \rightarrow W$								
影響を受けるステータス:	N, OV, C, DC, Z								
エンコード:	<table border="1"> <tr> <td>0000</td> <td>1111</td> <td>kkkk</td> <td>kkkk</td> </tr> </table>	0000	1111	kkkk	kkkk				
0000	1111	kkkk	kkkk						
説明:	Wレジスタの内容を8ビットのリテラル「k」に加算し、その結果をWレジスタに格納する。								
ワード数:	1								
サイクル数:	1								
Qサイクルの動作:	<table border="1"> <thead> <tr> <th>Q1</th> <th>Q2</th> <th>Q3</th> <th>Q4</th> </tr> </thead> <tbody> <tr> <td>デコード</td> <td>リテラル「k」を読み出し</td> <td>データを処理</td> <td>Wに書き込み</td> </tr> </tbody> </table>	Q1	Q2	Q3	Q4	デコード	リテラル「k」を読み出し	データを処理	Wに書き込み
Q1	Q2	Q3	Q4						
デコード	リテラル「k」を読み出し	データを処理	Wに書き込み						
例:	ADDLW 15h								
命令実行前									
W =	10h								
命令実行後									
W =	25h								

ADDWF	ADD W to f								
構文:	ADDWF f{,d{,a}}								
オペランド:	$0 \leq f \leq 255$ $d \in [0,1]$ $a \in [0,1]$								
動作:	$(W) + (f) \rightarrow \text{dest}$								
影響を受けるステータス:	N, OV, C, DC, Z								
エンコード:	<table border="1"> <tr> <td>0010</td> <td>01da</td> <td>ffff</td> <td>ffff</td> </tr> </table>	0010	01da	ffff	ffff				
0010	01da	ffff	ffff						
説明:	<p>Wレジスタをレジスタ「f」に加算する。「d」=「0」の場合、結果をWレジスタに格納する。「d」=「1」の場合、結果をレジスタ「f」に書き戻す(既定値)。</p> <p>「a」=「0」の場合、アクセスバンクを選択する。「a」=「1」の場合、BSRを使ってGPRバンクを選択する(既定値)。</p> <p>「a」=「0」で拡張命令セットを有効にしている場合、<math>f \leq 95</math> (5Fh) ならば、この命令は常にインデックス付きリテラルオフセットアドレッシングモードで実行される。詳細は、<a href="#">セクション 26.2.3「インデックス付きリテラルオフセットモードでのバイト/ビット指向命令」</a> 参照。</p>								
ワード数:	1								
サイクル数:	1								
Qサイクルの動作:	<table border="1"> <thead> <tr> <th>Q1</th> <th>Q2</th> <th>Q3</th> <th>Q4</th> </tr> </thead> <tbody> <tr> <td>デコード</td> <td>レジスタ「f」を読み出し</td> <td>データを処理</td> <td>格納先に書き込み</td> </tr> </tbody> </table>	Q1	Q2	Q3	Q4	デコード	レジスタ「f」を読み出し	データを処理	格納先に書き込み
Q1	Q2	Q3	Q4						
デコード	レジスタ「f」を読み出し	データを処理	格納先に書き込み						
例:	ADDWF REG, 0, 0								
命令実行前									
W =	17h								
REG =	0C2h								
命令実行後									
W =	0D9h								
REG =	0C2h								

**Note:** PIC18の命令は全て、命令ニーモニックの前にオプションの引数としてラベルを指定する事によってシンボリックアドレッシングが可能です。この場合の命令の形式は「{ラベル} 命令 引数」です。

# PIC18F97J60 ファミリ

## ADDWFC

### ADD W and Carry bit to f

構文: ADDWFC f{,d{,a}}

オペランド:  $0 \leq f \leq 255$   
 $d \in [0,1]$   
 $a \in [0,1]$

動作:  $(W) + (f) + (C) \rightarrow \text{dest}$

影響を受けるステータス: N,OV, C, DC, Z

エンコード:

0010	00da	ffff	ffff
------	------	------	------

説明: Wレジスタ、キャリーフラグ、データメモリのアドレス「f」の内容を加算する。「d」=「0」の場合、結果をWレジスタに格納する。「d」=「1」の場合、結果をデータメモリのアドレス「f」に格納する。

「a」=「0」の場合、アクセスバンクを選択する。「a」=「1」の場合、BSRを使ってGPRバンクを選択する(既定値)。

「a」=「0」で拡張命令セットを有効にしている場合、 $f \leq 95$  (5Fh) ならば、この命令は常にインデックス付きリテラル オフセット アドレッシング モードで実行される。詳細は、[セクション 26.2.3「インデックス付きリテラル オフセットモードでのバイト/ビット指向命令」](#) 参照。

ワード数: 1

サイクル数: 1

Qサイクルの動作:

Q1	Q2	Q3	Q4
デコード	レジスタ「f」を読み出し	データを処理	格納先に書き込み

例: ADDWFC REG, 0, 1

命令実行前  
 Carry bit = 1  
 REG = 02h  
 W = 4Dh

命令実行後  
 Carry bit = 0  
 REG = 02h  
 W = 50h

## ANDLW

### AND Literal with W

構文: ANDLW k

オペランド:  $0 \leq k \leq 255$

動作:  $(W) \text{ .AND. } k \rightarrow W$

影響を受けるステータス: N, Z

エンコード:

0000	1011	kkkk	kkkk
------	------	------	------

説明: Wレジスタの内容と8ビットのリテラル「k」をAND演算する。結果をWレジスタに格納する。

ワード数: 1

サイクル数: 1

Qサイクルの動作:

Q1	Q2	Q3	Q4
デコード	リテラル「k」を読み出し	データを処理	Wに書き込み

例: ANDLW 05Fh

命令実行前  
 W = A3h

命令実行後  
 W = 03h

# PIC18F97J60 ファミリ

## ANDWF AND W with f

構文: ANDWF f{,d{,a}}

オペランド:  $0 \leq f \leq 255$   
 $d \in [0,1]$   
 $a \in [0,1]$

動作: (W).AND. (f) → dest

影響を受けるステータス: N, Z

エンコード:

0001	01da	ffff	ffff
------	------	------	------

説明: W レジスタの内容とレジスタ「f」をAND演算する。「d」=「0」の場合、結果をWレジスタに格納する。「d」=「1」の場合、結果をレジスタ「f」に書き戻す(既定値)。「a」=「0」の場合、アクセスバンクを選択する。「a」=「1」の場合、BSRを使ってGPRバンクを選択する(既定値)。「a」=「0」で拡張命令セットを有効にしている場合、 $f \leq 95$  (5Fh) ならば、この命令は常にインデックス付きリテラルオフセットアドレッシングモードで実行される。詳細は、[セクション 26.2.3 「インデックス付きリテラルオフセットモードでのバイト/ビット指向命令」](#) 参照。

ワード数: 1

サイクル数: 1

Q サイクルの動作:

Q1	Q2	Q3	Q4
デコード	レジスタ「f」を読み出し	データを処理	格納先に書き込み

例: ANDWF REG, 0, 0

命令実行前  
W = 17h  
REG = C2h  
命令実行後  
W = 02h  
REG = C2h

## BC Branch if Carry

構文: BC n

オペランド:  $-128 \leq n \leq 127$

動作: キャリービットが「1」の場合  
 $(PC) + 2 + 2n \rightarrow PC$

影響を受けるステータス: なし

エンコード:

1110	0010	nnnn	nnnn
------	------	------	------

説明: キャリービットが「1」の場合、プログラムが分岐する。

PCには2の補数「2n」を加算する。PCは次の命令をフェッチするためにインクリメントしているため、新しいアドレスは  $PC + 2 + 2n$  となる。この場合、この命令は2サイクル命令となる。

ワード数: 1

サイクル数: 1(2)

Q サイクルの動作:

ジャンプする場合:

Q1	Q2	Q3	Q4
デコード	リテラル「n」を読み出し	データを処理	PCに書き込み
NOP	NOP	NOP	NOP

ジャンプしない場合:

Q1	Q2	Q3	Q4
デコード	リテラル「n」を読み出し	データを処理	NOP

例: HERE BC 5

命令実行前  
PC = address (HERE)  
命令実行後  
If Carry = 1;  
PC = address (HERE + 12)  
If Carry = 0;  
PC = address (HERE + 2)

# PIC18F97J60 ファミリ

## BCF Bit Clear f

構文: BCF f, b {,a}

オペランド:  $0 \leq f \leq 255$   
 $0 \leq b \leq 7$   
 $a \in [0,1]$

動作:  $0 \rightarrow f < b$

影響を受けるステータス: なし

エンコード:

1001	bbba	ffff	ffff
------	------	------	------

説明: レジスタ「f」のビット「b」をクリアする。

「a」=「0」の場合、アクセスバンクを選択する。「a」=「1」の場合、BSRを使ってGPRバンクを選択する(既定値)。

「a」=「0」で拡張命令セットを有効にしている場合、 $f \leq 95$  (5Fh) ならば、この命令は常にインデックス付きリテラルオフセットアドレッシングモードで実行される。詳細は、[セクション 26.2.3 「インデックス付きリテラルオフセットモードでのバイト/ビット指向命令」](#) 参照。

ワード数: 1

サイクル数: 1

Q サイクルの動作:

Q1	Q2	Q3	Q4
デコード	レジスタ「f」を読み出し	データを処理	レジスタ「f」に書き込み

例: BCF FLAG\_REG, 7, 0

命令実行前  
FLAG\_REG = C7h

命令実行後  
FLAG\_REG = 47h

## BN Branch if Negative

構文: BN n

オペランド:  $-128 \leq n \leq 127$

動作: NEGATIVE ビットが「1」の場合  
 $(PC) + 2 + 2n \rightarrow PC$

影響を受けるステータス: なし

エンコード:

1110	0110	nnnn	nnnn
------	------	------	------

説明: NEGATIVE ビットが「1」の場合、プログラムが分岐する。

PCには2の補数「2n」を加算する。PCは次の命令をフェッチするためにインクリメントしているため新しいアドレスは  $PC + 2 + 2n$  となる。この場合、この命令は2サイクル命令となる。

ワード数: 1

サイクル数: 1(2)

Q サイクルの動作:

ジャンプする場合:

Q1	Q2	Q3	Q4
デコード	リテラル「n」を読み出し	データを処理	PCに書き込み
NOP	NOP	NOP	NOP

ジャンプしない

場合:

Q1	Q2	Q3	Q4
デコード	リテラル「n」を読み出し	データを処理	NOP

例: HERE BN Jump

命令実行前  
PC = address (HERE)

命令実行後  
If Negative = 1;  
PC = address (Jump)  
If Negative = 0;  
PC = address (HERE + 2)









# PIC18F97J60 ファミリ

## BTFSC Bit Test File, Skip if Clear

構文: BTFSC f, b {,a}  
 オペランド:  $0 \leq f \leq 255$   
 $0 \leq b < 7$   
 $a \in [0,1]$

動作:  $(f < b) = 0$  の場合にスキップ  
 影響を受けるステータス: なし

エンコード:	1011	bbba	ffff	ffff
--------	------	------	------	------

説明: レジスタ「f」のビット「b」が「0」の場合、次の命令をスキップする。ビット「b」が「0」の場合、現在の命令実行中にフェッチした次の命令を破棄し、代わりにNOP命令を実行して2サイクル命令とする。

「a」=「0」の場合、アクセスバンクを選択する。「a」=「1」の場合、BSRを使ってGPRバンクを選択する(既定値)。

「a」=「0」で拡張命令セットを有効にしている場合、 $f \leq 95$  (5Fh) ならば、この命令は常にインデックス付きリテラルオフセットアドレッシングモードで実行される。詳細は、[セクション 26.2.3 「インデックス付きリテラルオフセットモードでのバイト/ビット指向命令」](#) 参照。

ワード数: 1  
 サイクル数: 1(2)  
**Note:** スキップ先の命令が2ワード命令の場合は3サイクルである。

Q サイクルの動作:

Q1	Q2	Q3	Q4
デコード	レジスタ「f」を読み出し	データを処理	NOP

スキップする場合:

Q1	Q2	Q3	Q4
NOP	NOP	NOP	NOP

スキップ先が2ワード命令の場合:

Q1	Q2	Q3	Q4
NOP	NOP	NOP	NOP
NOP	NOP	NOP	NOP

例: `HERE BTFSC FLAG, 1, 0`  
`FALSE :`  
`TRUE :`

命令実行前  
 PC = address (HERE)  
 命令実行後  
 If FLAG<1> = 0;  
 PC = address (TRUE)  
 If FLAG<1> = 1;  
 PC = address (FALSE)

## BTFSS Bit Test File, Skip if Set

構文: BTFSS f, b {,a}  
 オペランド:  $0 \leq f \leq 255$   
 $0 \leq b < 7$   
 $a \in [0,1]$

動作:  $(f < b) = 1$  の場合にスキップ  
 影響を受けるステータス: なし

エンコード:	1010	bbba	ffff	ffff
--------	------	------	------	------

説明: レジスタ「f」のビット「b」が「1」の場合、次の命令をスキップする。ビット「b」が「1」の場合、現在の命令実行中にフェッチした次の命令を破棄し、代わりにNOP命令を実行して2サイクル命令とする。

「a」=「0」の場合、アクセスバンクを選択する。「a」=「1」の場合、BSRを使ってGPRバンクを選択する(既定値)。

「a」=「0」で拡張命令セットを有効にしている場合、 $f \leq 95$  (5Fh) ならば、この命令は常にインデックス付きリテラルオフセットアドレッシングモードで実行される。詳細は、[セクション 26.2.3 「インデックス付きリテラルオフセットモードでのバイト/ビット指向命令」](#) 参照。

ワード数: 1  
 サイクル数: 1(2)  
**Note:** スキップ先の命令が2ワード命令の場合は3サイクルである。

Q サイクルの動作:

Q1	Q2	Q3	Q4
デコード	レジスタ「f」を読み出し	データを処理	NOP

スキップする場合:

Q1	Q2	Q3	Q4
NOP	NOP	NOP	NOP

スキップ先が2ワード命令の場合:

Q1	Q2	Q3	Q4
NOP	NOP	NOP	NOP
NOP	NOP	NOP	NOP

例: `HERE BTFSS FLAG, 1, 0`  
`FALSE :`  
`TRUE :`

命令実行前  
 PC = address (HERE)  
 命令実行後  
 If FLAG<1> = 0;  
 PC = address (FALSE)  
 If FLAG<1> = 1;  
 PC = address (TRUE)

# PIC18F97J60 ファミリ

## BTG Bit Toggle f

構文: BTG f, b {,a}

オペランド:  $0 \leq f \leq 255$   
 $0 \leq b < 7$   
 $a \in [0,1]$

動作:  $(\bar{f} < b) \rightarrow f < b$

影響を受ける なし

ステータス:

エンコード:

0111	bbba	ffff	ffff
------	------	------	------

説明: データメモリのアドレス「f」のビット「b」を反転する。

「a」=「0」の場合、アクセスバンクを選択する。「a」=「1」の場合、BSRを使ってGPRバンクを選択する(既定値)。

「a」=「0」で拡張命令セットを有効にしている場合、 $f \leq 95$  (5Fh) ならば、この命令は常にインデックス付きリテラルオフセットアドレッシングモードで実行される。詳細は、[セクション 26.2.3「インデックス付きリテラルオフセットモードでのバイト/ビット指向命令」](#) 参照。

ワード数: 1

サイクル数: 1

Q サイクルの動作:

Q1	Q2	Q3	Q4
デコード	レジスタ「f」を読み出し	データを処理	レジスタ「f」に書き込み

例: BTG PORTC, 4, 0

命令実行前  
 PORTC = 0111 0101 [75h]

命令実行後  
 PORTC = 0110 0101 [65h]

## BOV Branch if Overflow

構文: BOV n

オペランド:  $-128 \leq n \leq 127$

動作: オーバーフロービットが「1」の場合  
 $(PC) + 2 + 2n \rightarrow PC$

影響を受ける なし

ステータス:

エンコード:

1110	0100	nnnn	nnnn
------	------	------	------

説明: オーバーフロービットが「1」の場合、プログラムが分岐する。

PCには2の補数「2n」を加算する。PCは次の命令をフェッチするためにインクリメントしているため、新しいアドレスは $PC + 2 + 2n$ となる。この場合、この命令は2サイクル命令となる。

ワード数: 1

サイクル数: 1(2)

Q サイクルの動作:

ジャンプする場合:

Q1	Q2	Q3	Q4
デコード	リテラル「n」を読み出し	データを処理	PCに書き込み
NOP	NOP	NOP	NOP

ジャンプしない

場合:

Q1	Q2	Q3	Q4
デコード	リテラル「n」を読み出し	データを処理	NOP

例: HERE BOV Jump

命令実行前  
 PC = address (HERE)

命令実行後  
 If Overflow = 1;  
 PC = address (Jump)  
 If Overflow = 0;  
 PC = address (HERE + 2)

# PIC18F97J60 ファミリ

## BZ Branch if Zero

構文: BZ n

オペランド:  $-128 \leq n \leq 127$

動作: ゼロビットが「1」の場合  
(PC) + 2 + 2n → PC

影響を受ける ステータス: なし

エンコード:

1110	0000	nnnn	nnnn
------	------	------	------

説明: ゼロビットが「1」の場合、プログラムが分岐する。

PCには2の補数「2n」を加算する。PCは次の命令をフェッチするためにインクリメントしているため、新しいアドレスはPC + 2 + 2nとなる。この場合、この命令は2サイクル命令となる。

ワード数: 1

サイクル数: 1(2)

Q サイクルの動作:

ジャンプする場合:

Q1	Q2	Q3	Q4
デコード	リテラル「n」を読み出し	データを処理	PCに書き込み
NOP	NOP	NOP	NOP

ジャンプしない場合:

Q1	Q2	Q3	Q4
デコード	リテラル「n」を読み出し	データを処理	NOP

例:                    HERE            BZ    Jump

命令実行前  
PC            =    address (HERE)

命令実行後  
If Zero       =    1;  
PC            =    address (Jump)  
If Zero       =    0;  
PC            =    address (HERE + 2)

## CALL Subroutine Call

構文: CALL k {,s}

オペランド:  $0 \leq k \leq 1048575$   
s ∈ [0,1]

動作: (PC) + 4 → TOS,  
k → PC<20:1>;  
s = 1 の場合  
(W) → WS,  
(STATUS) → STATUSS,  
(BSR) → BSRS

影響を受ける ステータス: なし

エンコード:

1 ワード目 (k<7:0>)  
2 ワード目 (k<19:8>)

1110	110s	k <sub>7</sub> kkk	kkkk <sub>0</sub>
1111	k <sub>19</sub> kkk	kkkk	kkkk <sub>8</sub>

説明: 2MB のメモリ空間全体を対象に、サブルーチン呼び出す。まず、リターンアドレス (PC + 4) をリターンスタックにプッシュする。  
「s」= 1 の場合、W、STATUS、BSR レジスタもそれぞれのシャドーレジスタの WS、STATUSS、BSRS にプッシュする。  
「s」= 0 の場合、これらのレジスタは更新されない (既定値)。次に、20 ビットの値「k」を PC<20:1> に読み込む。  
CALL は 2 サイクル命令である。

ワード数: 2

サイクル数: 2

Q サイクルの動作:

Q1	Q2	Q3	Q4
デコード	リテラル「k」<7:0>を読み出し	PC をスタックにプッシュ	リテラル「k」<19:8>を読み出し、PCに書き込み
NOP	NOP	NOP	NOP

例:                    HERE            CALL    THERE,1

命令実行前  
PC            =    address (HERE)

命令実行後  
PC            =    address (THERE)  
TOS           =    address (HERE + 4)  
WS            =    W  
BSRS          =    BSR  
STATUSS      =    STATUS

# PIC18F97J60 ファミリ

## CLRF Clear f

構文: CLRF f{,a}

オペランド:  $0 \leq f \leq 255$   
 $a \in [0,1]$

動作:  $000h \rightarrow f$ ,  
 $1 \rightarrow Z$

影響を受けるステータス: Z

エンコード:

0110	101a	ffff	ffff
------	------	------	------

説明: 指定したレジスタの値をクリアする。  
「a」=「0」の場合、アクセスバンクを選択する。「a」=「1」の場合、BSRを使ってGPRバンクを選択する(既定値)。  
「a」=「0」で拡張命令セットを有効にしている場合、 $f \leq 95$  (5Fh) ならば、この命令は常にインデックス付きリテラルオフセットアドレッシングモードで実行される。詳細は、[セクション 26.2.3 「インデックス付きリテラルオフセットモードでのバイト/ビット指向命令」](#) 参照。

ワード数: 1

サイクル数: 1

Q サイクルの動作:

Q1	Q2	Q3	Q4
デコード	レジスタ「f」を読み出し	データを処理	レジスタ「f」に書き込み

例: CLRF FLAG\_REG, 1

命令実行前  
FLAG\_REG = 5Ah  
命令実行後  
FLAG\_REG = 00h

## CLRWDTClear Watchdog Timer

構文: CLRWDTClear Watchdog Timer

オペランド: なし

動作:  $000h \rightarrow WDT$ ,  
 $000h \rightarrow WDT\ postscaler$ ,  
 $1 \rightarrow \overline{TO}$ ,  
 $1 \rightarrow \overline{PD}$

影響を受けるステータス:  $\overline{TO}$ ,  $\overline{PD}$

エンコード:

0000	0000	0000	0100
------	------	------	------

説明: CLRWDTClear Watchdog Timer 命令はウォッチドッグタイマをリセットする。WDTのポストスケールも同時にリセットする。ステータスビット  $\overline{TO}$  と  $\overline{PD}$  がセットされる。

ワード数: 1

サイクル数: 1

Q サイクルの動作:

Q1	Q2	Q3	Q4
デコード	NOP	データを処理	NOP

例: CLRWDTClear Watchdog Timer

命令実行前  
WDT Counter = ?  
命令実行後  
WDT Counter = 00h  
WDT Postscaler = 0  
 $\overline{TO}$  = 1  
 $\overline{PD}$  = 1

# PIC18F97J60 ファミリ

## COMF Complement f

構文: COMF f{,d{,a}}

オペランド:  $0 \leq f \leq 255$   
 $d \in [0,1]$   
 $a \in [0,1]$

動作:  $(\bar{f}) \rightarrow \text{dest}$

影響を受けるステータス: N, Z

エンコード: 

0001	11da	ffff	ffff
------	------	------	------

説明: レジスタ「f」の内容の補数をとる。「d」=「0」の場合、結果をWレジスタに格納する。「d」=「1」の場合、結果をレジスタ「f」に書き戻す(既定値)。

「a」=「0」の場合、アクセスバンクを選択する。「a」=「1」の場合、BSRを使ってGPRバンクを選択する(既定値)。

「a」=「0」で拡張命令セットを有効にしている場合、 $f \leq 95$  (5Fh) ならば、この命令は常にインデックス付きリテラルオフセットアドレッシングモードで実行される。詳細は、[セクション 26.2.3「インデックス付きリテラルオフセットモードでのバイト/ビット指向命令」](#) 参照。

ワード数: 1

サイクル数: 1

Qサイクルの動作:

Q1	Q2	Q3	Q4
デコード	レジスタ「f」を読み出し	データを処理	格納先に書き込み

例: COMF REG, 0, 0

命令実行前  
 REG = 13h  
 命令実行後  
 REG = 13h  
 W = ECh

## CPFSEQ Compare f with W, Skip if f = W

構文: CPFSEQ f{,a}

オペランド:  $0 \leq f \leq 255$   
 $a \in [0,1]$

動作: (f) - (W),  
 (f) = (W) の場合スキップ  
 (符号なし比較)

影響を受けるステータス: なし

エンコード: 

0110	001a	ffff	ffff
------	------	------	------

説明: データメモリのアドレス「f」の内容とWレジスタの内容を、符号なし減算を実行して比較する。

「f」= W ならばフェッチした命令を破棄し、代わりにNOPを実行して2サイクル命令とする。

「a」=「0」の場合、アクセスバンクを選択する。「a」=「1」の場合、BSRを使ってGPRバンクを選択する(既定値)。

「a」=「0」で拡張命令セットを有効にしている場合、 $f \leq 95$  (5Fh) ならば、この命令は常にインデックス付きリテラルオフセットアドレッシングモードで実行される。詳細は、[セクション 26.2.3「インデックス付きリテラルオフセットモードでのバイト/ビット指向命令」](#) 参照。

ワード数: 1

サイクル数: 1(2)

**Note:** スキップ先の命令が2ワード命令の場合は3サイクルである。

Qサイクルの動作:

Q1	Q2	Q3	Q4
デコード	レジスタ「f」を読み出し	データを処理	NOP

スキップする場合:

Q1	Q2	Q3	Q4
NOP	NOP	NOP	NOP

スキップ先が2ワード命令の場合:

Q1	Q2	Q3	Q4
NOP	NOP	NOP	NOP
NOP	NOP	NOP	NOP

例: 

```
HERE CPFSEQ REG, 0
NEQUAL :
EQUAL :
```

命令実行前  
 PC Address = HERE  
 W = ?  
 REG = ?  
 命令実行後  
 If REG = W;  
 PC = Address (EQUAL)  
 If REG ≠ W;  
 PC = Address (NEQUAL)



# PIC18F97J60 ファミリ

## CPFSGT Compare f with W, Skip if f > W

構文: CPFSGT f{,a}  
 オペランド:  $0 \leq f \leq 255$   
 $a \in [0,1]$   
 動作: (f) - (W)  
 (f) > (W) の場合スキップ  
 (符号なし比較)  
 影響を受けるステータス: なし  
 エンコード:

0110	010a	ffff	ffff
------	------	------	------

説明: データメモリのアドレス「f」の内容と W レジスタの内容を、符号なし減算を実行して比較する。

「f」の内容の方が WREG の内容より大きい場合はフェッチした命令を破棄し、代わりに NOP を実行して 2 サイクル命令とする。

「a」= 「0」の場合、アクセスバンクを選択する。「a」= 「1」の場合、BSR を使って GPR バンクを選択する (既定値)。

「a」= 「0」で拡張命令セットを有効にしている場合、 $f \leq 95$  (5Fh) ならば、この命令は常にインデックス付きリテラル オフセット アドレッシング モードで実行される。詳細は、[セクション 26.2.3 「インデックス付きリテラル オフセットモードでのバイト/ビット指向命令」](#) 参照。

ワード数: 1

サイクル数: 1(2)

**Note:** スキップ先の命令が 2 ワード命令の場合は 3 サイクルである。

Q サイクルの動作:

Q1	Q2	Q3	Q4
デコード	レジスタ「f」を読み出し	データを処理	NOP

スキップする場合:

Q1	Q2	Q3	Q4
NOP	NOP	NOP	NOP

スキップ先が 2 ワード命令の場合:

Q1	Q2	Q3	Q4
NOP	NOP	NOP	NOP
NOP	NOP	NOP	NOP

例: 

```
HERE    CPFSGT REG, 0
NGREATER :
GREATER  :
```

命令実行前  
 PC = Address (HERE)  
 W = ?  
 命令実行後  
 If REG > W;  
 PC = Address (GREATER)  
 If REG ≤ W;  
 PC = Address (NGREATER)

## CPFSLT Compare f with W, Skip if f < W

構文: CPFSLT f{,a}  
 オペランド:  $0 \leq f \leq 255$   
 $a \in [0,1]$   
 動作: (f) - (W),  
 (f) < (W) の場合スキップ  
 (符号なし比較)  
 影響を受けるステータス: なし  
 エンコード:

0110	000a	ffff	ffff
------	------	------	------

説明: データメモリのアドレス「f」の内容と W レジスタの内容を、符号なし減算を実行して比較する。

「f」の内容の方が W レジスタの内容より小さい場合はフェッチした命令を破棄し、代わりに NOP を実行して 2 サイクル命令とする。

「a」= 「0」の場合、アクセスバンクを選択する。「a」= 「1」の場合、BSR を使って GPR バンクを選択する (既定値)。

ワード数: 1

サイクル数: 1(2)

**Note:** スキップ先の命令が 2 ワード命令の場合は 3 サイクルである。

Q サイクルの動作:

Q1	Q2	Q3	Q4
デコード	レジスタ「f」を読み出し	データを処理	NOP

スキップする場合:

Q1	Q2	Q3	Q4
NOP	NOP	NOP	NOP

スキップ先が 2 ワード命令の場合:

Q1	Q2	Q3	Q4
NOP	NOP	NOP	NOP
NOP	NOP	NOP	NOP

例: 

```
HERE    CPFSLT REG, 1
NLESS   :
LESS    :
```

命令実行前  
 PC = Address (HERE)  
 W = ?  
 命令実行後  
 If REG < W;  
 PC = Address (LESS)  
 If REG ≥ W;  
 PC = Address (NLESS)

# PIC18F97J60 ファミリ

## DAW Decimal Adjust W Register

構文: DAW  
 オペランド: なし  
 動作: [W<3:0> > 9] または [DC = 1] の場合  
 (W<3:0>) + 6 → W<3:0>;  
 それ以外の場合  
 (W<3:0>) → W<3:0>

[W<7:4> > 9] または [C = 1] の場合  
 (W<7:4>) + 6 → W<7:4>;  
 C = 1;  
 それ以外の場合  
 (W<7:4>) → W<7:4>

影響を受けるステータス: C

エンコード: 

0000	0000	0000	0111
------	------	------	------

説明: DAW 命令は、2 つの変数 (いずれもパック形式の BCD) を加算して W レジスタに格納した 8 ビットの値を調整し、正しいパック形式の BCD の結果を生成する。

ワード数: 1

サイクル数: 1

Q サイクルの動作:

	Q1	Q2	Q3	Q4
デコード				
W レジスタを 読み出し				
データを 処理				
W に 書き込み				

例 1:

DAW  
 命令実行前  
 W = A5h  
 C = 0  
 DC = 0

命令実行後  
 W = 05h  
 C = 1  
 DC = 0

例 2:

命令実行前  
 W = CEh  
 C = 0  
 DC = 0

命令実行後  
 W = 34h  
 C = 1  
 DC = 0

## DECF Decrement f

構文: DECF f{,d{,a}}  
 オペランド:  $0 \leq f \leq 255$   
 $d \in [0,1]$   
 $a \in [0,1]$

動作: (f) - 1 → dest

影響を受けるステータス: C, DC, N, OV, Z

エンコード: 

0000	01da	ffff	ffff
------	------	------	------

説明: レジスタ「f」をデクリメントする。「d」=「0」の場合、結果を W レジスタに格納する。「d」=「1」の場合、結果をレジスタ「f」に書き戻す (既定値)。

「a」=「0」の場合、アクセスバンクを選択する。「a」=「1」の場合、BSR を使って GPR バンクを選択する (既定値)。

「a」=「0」で拡張命令セットを有効にしている場合、 $f \leq 95$  (5Fh) ならば、この命令は常にインデックス付きリテラル オフセット アドレッシング モードで実行される。詳細は、[セクション 26.2.3 「インデックス付きリテラル オフセットモードでのバイト / ビット指向命令」](#) 参照。

ワード数: 1

サイクル数: 1

Q サイクルの動作:

	Q1	Q2	Q3	Q4
デコード				
レジスタ 「f」を 読み出し				
データを 処理				
格納先に 書き込み				

例:

DECF CNT, 1, 0  
 命令実行前  
 CNT = 01h  
 Z = 0  
 命令実行後  
 CNT = 00h  
 Z = 1

# PIC18F97J60 ファミリ

## DECFSZ Decrement f, Skip if 0

構文: DECFSZ f{,d{,a}}

オペランド:  $0 \leq f \leq 255$   
 $d \in [0,1]$   
 $a \in [0,1]$

動作:  $(f) - 1 \rightarrow \text{dest}$ ,  
 結果 = 0 の場合スキップ

影響を受けるステータス: なし

エンコード:	0010	11da	ffff	ffff
--------	------	------	------	------

説明: レジスタ「f」の内容をデクリメントする。「d」=「0」の場合、結果を W レジスタに格納する。「d」=「1」の場合、結果をレジスタ「f」に書き戻す (既定値)。

結果が「0」の場合、フェッチ済みの次の命令を破棄し、代わりに NOP を実行して 2 サイクル命令とする。

「a」=「0」の場合、アクセスバンクを選択する。「a」=「1」の場合、BSR を使って GPR バンクを選択する (既定値)。

「a」=「0」で拡張命令セットを有効にしている場合、 $f \leq 95$  (5Fh) ならば、この命令は常にインデックス付きリテラル オフセット アドレッシング モードで実行される。詳細は、[セクション 26.2.3「インデックス付きリテラル オフセットモードでのバイト/ビット指向命令」](#) 参照。

ワード数: 1

サイクル数: 1(2)

**Note:** スキップ先の命令が 2 ワード命令の場合は 3 サイクルである。

Q サイクルの動作:

Q1	Q2	Q3	Q4
デコード	レジスタ「f」を読み出し	データを処理	格納先に書き込み

スキップする場合:

Q1	Q2	Q3	Q4
NOP	NOP	NOP	NOP

スキップ先が 2 ワード命令の場合:

Q1	Q2	Q3	Q4
NOP	NOP	NOP	NOP
NOP	NOP	NOP	NOP

例:                    HERE        DECFSZ   CNT, 1, 1  
                       GOTO        LOOP  
                       CONTINUE

命令実行前  
 PC = Address (HERE)  
 命令実行後  
 CNT = CNT - 1  
 If CNT = 0;  
   PC = Address (CONTINUE)  
 If CNT  $\neq$  0;  
   PC = Address (HERE + 2)

## DCFSNZ Decrement f, Skip if Not 0

構文: DCFSNZ f{,d{,a}}

オペランド:  $0 \leq f \leq 255$   
 $d \in [0,1]$   
 $a \in [0,1]$

動作:  $(f) - 1 \rightarrow \text{dest}$ ,  
 結果  $\neq 0$  の場合スキップ

影響を受けるステータス: なし

エンコード:	0100	11da	ffff	ffff
--------	------	------	------	------

説明: レジスタ「f」の内容をデクリメントする。「d」=「0」の場合、結果を W レジスタに格納する。「d」=「1」の場合、結果をレジスタ「f」に書き戻す (既定値)。

結果が「0」でなければフェッチ済みの次の命令を破棄し、代わりに NOP を実行して 2 サイクル命令とする。

「a」=「0」の場合、アクセスバンクを選択する。「a」=「1」の場合、BSR を使って GPR バンクを選択する (既定値)。

「a」=「0」で拡張命令セットを有効にしている場合、 $f \leq 95$  (5Fh) ならば、この命令は常にインデックス付きリテラル オフセット アドレッシング モードで実行される。詳細は、[セクション 26.2.3「インデックス付きリテラル オフセットモードでのバイト/ビット指向命令」](#) 参照。

ワード数: 1

サイクル数: 1(2)

**Note:** スキップ先の命令が 2 ワード命令の場合は 3 サイクルである。

Q サイクルの動作:

Q1	Q2	Q3	Q4
デコード	レジスタ「f」を読み出し	データを処理	格納先に書き込み

スキップする場合:

Q1	Q2	Q3	Q4
NOP	NOP	NOP	NOP

スキップ先が 2 ワード命令の場合:

Q1	Q2	Q3	Q4
NOP	NOP	NOP	NOP
NOP	NOP	NOP	NOP

例:                    HERE        DCFSNZ   TEMP, 1, 0  
                       ZERO        :  
                       NZERO       :

命令実行前  
 TEMP = ?  
 命令実行後  
 TEMP = TEMP - 1,  
 If TEMP = 0;  
   PC = Address (ZERO)  
 If TEMP  $\neq$  0;  
   PC = Address (NZERO)

# PIC18F97J60 ファミリ

## GOTO Unconditional Branch

構文: GOTO k  
 オペランド:  $0 \leq k \leq 1048575$   
 動作:  $k \rightarrow PC \langle 20:1 \rangle$   
 影響を受けるステータス: なし

エンコード:				
1 ワード目 (k<7:0>)	1110	1111	k <sub>7</sub> kkk	kkkk <sub>0</sub>
2 ワード目 (k<19:8>)	1111	k <sub>19</sub> kkk	kkkk	kkkk <sub>8</sub>

説明: GOTO は、2 MB の全メモリ空間の任意のアドレスへの無条件分岐を実行する。20 ビットの値「k」を PC<20:1> に読み込む。GOTO は常に 2 サイクル命令である。

ワード数: 2  
 サイクル数: 2

Q サイクルの動作:

	Q1	Q2	Q3	Q4
デコード		リテラル「k」<7:0>を読み出し	NOP	リテラル「k」<19:8>を読み出し、PCに書き込み
NOP	NOP	NOP	NOP	NOP

例: GOTO THERE  
 命令実行後 PC = Address (THERE)

## INCF Increment f

構文: INCF f{,d{,a}}

オペランド:  $0 \leq f \leq 255$   
 $d \in [0,1]$   
 $a \in [0,1]$

動作:  $(f) + 1 \rightarrow dest$

影響を受けるステータス: C, DC, N, OV, Z

エンコード:	0010	10da	ffff	ffff
--------	------	------	------	------

説明: レジスタ「f」の内容をインクリメントする。「d」=「0」の場合、結果を W レジスタに格納する。「d」=「1」の場合、結果をレジスタ「f」に書き戻す(既定値)。

「a」=「0」の場合、アクセスバンクを選択する。「a」=「1」の場合、BSR を使って GPR バンクを選択する(既定値)。

「a」=「0」で拡張命令セットを有効にしている場合、 $f \leq 95$  (5Fh) ならば、この命令は常にインデックス付きリテラル オフセット アドレッシング モードで実行される。詳細は、[セクション 26.2.3 「インデックス付きリテラル オフセットモードでのバイト/ビット指向命令」](#) 参照。

ワード数: 1  
 サイクル数: 1

Q サイクルの動作:

	Q1	Q2	Q3	Q4
デコード		レジスタ「f」を読み出し	データを処理	格納先に書き込み

例: INCF CNT, 1, 0

命令実行前  
 CNT = FFh  
 Z = 0  
 C = ?  
 DC = ?

命令実行後  
 CNT = 00h  
 Z = 1  
 C = 1  
 DC = 1

# PIC18F97J60 ファミリ

## INCFSZ Increment f, Skip if 0

構文: INCFSZ f{,d{,a}}

オペランド:  $0 \leq f \leq 255$   
 $d \in [0,1]$   
 $a \in [0,1]$

動作:  $(f) + 1 \rightarrow \text{dest}$ ,  
 結果 = 0 の場合スキップ

影響を受けるステータス: なし

エンコード:

0011	11da	ffff	ffff
------	------	------	------

説明: レジスタ「f」の内容をインクリメントする。「d」=「0」の場合、結果をWレジスタに格納する。「d」=「1」の場合、結果はレジスタ「f」に書き戻される。(既定値)

結果が「0」の場合、フェッチ済みの次の命令を破棄し、代わりにNOPを実行して2サイクル命令とする。

「a」=「0」の場合、アクセスバンクを選択する。「a」=「1」の場合、BSRを使ってGPRバンクを選択する(既定値)。

「a」=「0」で拡張命令セットを有効にしている場合、 $f \leq 95$  (5Fh) ならば、この命令は常にインデックス付きリテラルオフセットアドレッシングモードで実行される。詳細は、[セクション 26.2.3「インデックス付きリテラルオフセットモードでのバイト/ビット指向命令」](#) 参照。

ワード数: 1

サイクル数: 1(2)

**Note:** スキップ先の命令が2ワード命令の場合は3サイクルである。

Qサイクルの動作:

Q1	Q2	Q3	Q4
デコード	レジスタ「f」を読み出し	データを処理	格納先に書き込み

スキップする場合:

Q1	Q2	Q3	Q4
NOP	NOP	NOP	NOP

スキップ先が2ワード命令の場合:

Q1	Q2	Q3	Q4
NOP	NOP	NOP	NOP
NOP	NOP	NOP	NOP

例: `HERE INCFSZ CNT, 1, 0`  
`NZERO :`  
`ZERO :`

命令実行前  
 PC = Address (HERE)  
 命令実行後  
 CNT = CNT + 1  
 If CNT = 0;  
 PC = Address (ZERO)  
 If CNT  $\neq$  0;  
 PC = Address (NZERO)

## INFSNZ Increment f, Skip if Not 0

構文: INFSNZ f{,d{,a}}

オペランド:  $0 \leq f \leq 255$   
 $d \in [0,1]$   
 $a \in [0,1]$

動作:  $(f) + 1 \rightarrow \text{dest}$ ,  
 結果  $\neq$  0 の場合スキップ

影響を受けるステータス: なし

エンコード:

0100	10da	ffff	ffff
------	------	------	------

説明: レジスタ「f」の内容をインクリメントする。「d」=「0」の場合、結果をWレジスタに格納する。「d」=「1」の場合、結果はレジスタ「f」に書き戻す(既定値)。

結果が「0」でなければフェッチ済みの次の命令を破棄し、代わりにNOPを実行して2サイクル命令とする。

「a」=「0」の場合、アクセスバンクを選択する。「a」=「1」の場合、BSRを使ってGPRバンクを選択する(既定値)。

「a」=「0」で拡張命令セットを有効にしている場合、 $f \leq 95$  (5Fh) ならば、この命令は常にインデックス付きリテラルオフセットアドレッシングモードで実行される。詳細は、[セクション 26.2.3「インデックス付きリテラルオフセットモードでのバイト/ビット指向命令」](#) 参照。

ワード数: 1

サイクル数: 1(2)

**Note:** スキップ先の命令が2ワード命令の場合は3サイクルである。

Qサイクルの動作:

Q1	Q2	Q3	Q4
デコード	レジスタ「f」を読み出し	データを処理	格納先に書き込み

スキップする場合:

Q1	Q2	Q3	Q4
NOP	NOP	NOP	NOP

スキップ先が2ワード命令の場合:

Q1	Q2	Q3	Q4
NOP	NOP	NOP	NOP
NOP	NOP	NOP	NOP

例: `HERE INFSNZ REG, 1, 0`  
`ZERO :`  
`NZERO :`

命令実行前  
 PC = Address (HERE)  
 命令実行後  
 REG = REG + 1  
 If REG  $\neq$  0;  
 PC = Address (NZERO)  
 If REG = 0;  
 PC = Address (ZERO)

# PIC18F97J60 ファミリ

## IORLW Inclusive OR Literal with W

構文: IORLW k

オペランド:  $0 \leq k \leq 255$

動作: (W).OR. k → W

影響を受けるステータス: N, Z

エンコード:

0000	1001	kkkk	kkkk
------	------	------	------

説明: Wレジスタの内容と8ビットのリテラル「k」をOR演算する。結果をWレジスタに格納する。

ワード数: 1

サイクル数: 1

Qサイクルの動作:

	Q1	Q2	Q3	Q4
デコード	リテラル「k」を読み出し	データを処理	Wに書き込み	

例: IORLW 35h

命令実行前  
W = 9Ah

命令実行後  
W = BFh

## IORWF Inclusive OR W with f

構文: IORWF f{,d{,a}}

オペランド:  $0 \leq f \leq 255$

$d \in [0,1]$

$a \in [0,1]$

動作: (W).OR. (f) → dest

影響を受けるステータス: N, Z

エンコード:

0001	00da	ffff	ffff
------	------	------	------

説明: Wレジスタとレジスタ「f」の内容をOR演算する。「d」=「0」の場合、結果をWレジスタに格納する。「d」=「1」の場合、結果をレジスタ「f」に書き戻す(既定値)。

「a」=「0」の場合、アクセスバンクを選択する。「a」=「1」の場合、BSRを使ってGPRバンクを選択する(既定値)。

「a」=「0」で拡張命令セットを有効にしている場合、 $f \leq 95$  (5Fh) ならば、この命令は常にインデックス付きリテラルオフセットアドレッシングモードで実行される。詳細は、[セクション 26.2.3 「インデックス付きリテラルオフセットモードでのバイト/ビット指向命令」](#) 参照。

ワード数: 1

サイクル数: 1

Qサイクルの動作:

	Q1	Q2	Q3	Q4
デコード	レジスタ「f」を読み出し	データを処理	格納先に書き込み	

例: IORWF RESULT, 0, 1

命令実行前  
RESULT = 13h  
W = 91h

命令実行後  
RESULT = 13h  
W = 93h

# PIC18F97J60 ファミリ

LFSR	Load FSR								
構文:	LFSR f, k								
オペランド:	$0 \leq f \leq 2$ $0 \leq k \leq 4095$								
動作:	$k \rightarrow \text{FSRf}$								
影響を受けるステータス:	なし								
エンコード:	<table border="1"> <tr> <td>1110</td> <td>1110</td> <td>00ff</td> <td>k<sub>11</sub>kkk</td> </tr> <tr> <td>1111</td> <td>0000</td> <td>k<sub>7</sub>kkk</td> <td>kkkk</td> </tr> </table>	1110	1110	00ff	k <sub>11</sub> kkk	1111	0000	k <sub>7</sub> kkk	kkkk
1110	1110	00ff	k <sub>11</sub> kkk						
1111	0000	k <sub>7</sub> kkk	kkkk						
説明:	12ビットのリテラル「k」を、「f」が指し示すFSRに読み込む。								
ワード数:	2								
サイクル数:	2								
Qサイクルの動作:									

	Q1	Q2	Q3	Q4
デコード	リテラル「k」のMSBを読み出し	データを処理	FSRfHにリテラル「k」のMSBを書き込み	
デコード	リテラル「k」のLSBを読み出し	データを処理	FSRfLにリテラル「k」を書き込み	

例: LFSR 2, 3ABh

命令実行後  
FSR2H = 03h  
FSR2L = ABh

MOVf	Move f				
構文:	MOVf f{,d{,a}}				
オペランド:	$0 \leq f \leq 255$ $d \in [0,1]$ $a \in [0,1]$				
動作:	$f \rightarrow \text{dest}$				
影響を受けるステータス:	N, Z				
エンコード:	<table border="1"> <tr> <td>0101</td> <td>00da</td> <td>ffff</td> <td>ffff</td> </tr> </table>	0101	00da	ffff	ffff
0101	00da	ffff	ffff		
説明:	レジスタ「f」の内容を、「d」のステータスに応じて格納先に移動する。 「d」=「0」の場合、結果をWレジスタに格納する。「d」=「1」の場合、結果をレジスタ「f」に書き戻す(既定値)。「f」のアドレスは、256バイトバンクの任意の場所を指定できる。 「a」=「0」の場合、アクセスバンクを選択する。「a」=「1」の場合、BSRを使ってGPRバンクを選択する(既定値)。 「a」=「0」で拡張命令セットを有効にしている場合、 $f \leq 95$ (5Fh) ならば、この命令は常にインデックス付きリテラルオフセットアドレッシングモードで実行される。詳細は、 <a href="#">セクション 26.2.3「インデックス付きリテラルオフセットモードでのバイト/ビット指向命令」</a> 参照。				
ワード数:	1				
サイクル数:	1				
Qサイクルの動作:					

	Q1	Q2	Q3	Q4
デコード	レジスタ「f」を読み出し	データを処理	Wに書き込み	

例: MOVf REG, 0, 0

命令実行前  
REG = 22h  
W = FFh  
命令実行後  
REG = 22h  
W = 22h

# PIC18F97J60 ファミリ

## MOVFF

Move f to f

構文: MOVFF  $f_s, f_d$

オペランド:  $0 \leq f_s \leq 4095$   
 $0 \leq f_d \leq 4095$

動作:  $(f_s) \rightarrow f_d$

影響を受ける ステータス: なし

エンコード:

1ワード目 (送信元)	1100	ffff	ffff	ffff $f_s$
2ワード目 (格納先)	1111	ffff	ffff	ffff $f_d$

説明: 送信元レジスタ「 $f_s$ 」の内容を格納先レジスタ「 $f_d$ 」に移動する。  
 送信元「 $f_s$ 」と格納先「 $f_d$ 」のアドレスはいずれも 4096 バイトのデータ空間 (000h ~ FFFh) の任意の場所を指定できる。

送信元と格納先のいずれかを W レジスタとする事も可能 (特別な場合) である。

MOVFF は、特にデータメモリのアドレスを周辺レジスタ (送信バッファや I/O ポート等) に移す際に使う。

MOVFF 命令では、PCL、TOSU、TOSH、TOSL を格納先レジスタには使えない。

ワード数: 2

サイクル数: 2

Q サイクルの動作:

	Q1	Q2	Q3	Q4
デコード	レジスタ「f」(送信元)を読み出し		データを処理	NOP
デコード		NOP ダミー読み出しなし	NOP	レジスタ「f」(格納先)に書き込み

例: MOVFF REG1, REG2

命令実行前  
 REG1 = 33h  
 REG2 = 11h  
 命令実行後  
 REG1 = 33h  
 REG2 = 33h

## MOVLB

Move Literal to Low Nibble in BSR

構文: MOVLW k

オペランド:  $0 \leq k \leq 255$

動作:  $k \rightarrow \text{BSR}$

影響を受ける ステータス: なし

エンコード:	0000	0001	kkkk	kkkk
--------	------	------	------	------

説明: 8 ビットのリテラル「k」をバンクセレクトレジスタ (BSR) に読み込む。k<sub>7</sub>:k<sub>4</sub> の値に関わらず、BSR<7:4> の値は常に「0」である。

ワード数: 1

サイクル数: 1

Q サイクルの動作:

	Q1	Q2	Q3	Q4
デコード		リテラル「k」を読み出し	データを処理	BSR にリテラル「k」を書き込み

例: MOVLB 5

命令実行前  
 BSR Register = 02h  
 命令実行後  
 BSR Register = 05h



# PIC18F97J60 ファミリ

## MOVLW Move Literal to W

構文: MOVLW k

オペランド:  $0 \leq k \leq 255$

動作:  $k \rightarrow W$

影響を受けるステータス: なし

エンコード:

0000	1110	kkkk	kkkk
------	------	------	------

説明: 8ビットのリテラル「k」をWレジスタに読み込む。

ワード数: 1

サイクル数: 1

Qサイクルの動作:

	Q1	Q2	Q3	Q4
デコード		リテラル「k」を読み出し	データを処理	Wに書き込み

例: MOVLW 5Ah

命令実行後  
W = 5Ah

## MOVWF Move W to f

構文: MOVWF f{,a}

オペランド:  $0 \leq f \leq 255$

$a \in [0,1]$

動作:  $(W) \rightarrow f$

影響を受けるステータス: なし

エンコード:

0110	111a	ffff	ffff
------	------	------	------

説明: Wレジスタからレジスタ「f」へデータを移動する。「f」のアドレスは、256バイトバンクの任意の場所を指定できる。

「a」=「0」の場合、アクセスバンクを選択する。「a」=「1」の場合、BSRを使ってGPRバンクを選択する(既定値)。

「a」=「0」で拡張命令セットを有効にしている場合、 $f \leq 95$  (5Fh) ならば、この命令は常にインデックス付きリテラルオフセットアドレッシングモードで実行される。詳細は、[セクション 26.2.3 「インデックス付きリテラルオフセットモードでのバイト/ビット指向命令」](#) 参照。

ワード数: 1

サイクル数: 1

Qサイクルの動作:

	Q1	Q2	Q3	Q4
デコード		レジスタ「f」を読み出し	データを処理	レジスタ「f」に書き込み

例: MOVWF REG, 0

命令実行前  
W = 4Fh  
REG = FFh

命令実行後  
W = 4Fh  
REG = 4Fh

# PIC18F97J60 ファミリ

## MULLW Multiply Literal with W

構文: MULLW k

オペランド:  $0 \leq k \leq 255$

動作:  $(W) \times k \rightarrow \text{PRODH:PRODL}$

影響を受ける ステータス: なし

エンコード:

0000	1101	kkkk	kkkk
------	------	------	------

説明: Wレジスタの内容と8ビットのリテラル「k」とで符号なし乗算を実行する。16ビットの結果をPRODH:PRODLレジスタペアに格納する。PRODHには上位バイトを格納する。

Wレジスタの内容は変化しない。

ステータスフラグは一切変化しない。

この演算では、オーバーフローもキャリーも不可能である。結果がゼロになる事はあっても、検出されない。

ワード数: 1

サイクル数: 1

Qサイクルの動作:

Q1	Q2	Q3	Q4
デコード	リテラル「k」を読み出し	データを処理	PRODH:PRODLレジスタに書き込み

例: MULLW 0C4h

命令実行前  
W = E2h  
PRODH = ?  
PRODL = ?

命令実行後  
W = E2h  
PRODH = ADh  
PRODL = 08h

## MULWF Multiply W with f

構文: MULWF f{,a}

オペランド:  $0 \leq f \leq 255$   
 $a \in [0,1]$

動作:  $(W) \times (f) \rightarrow \text{PRODH:PRODL}$

影響を受ける ステータス: なし

エンコード:

0000	001a	ffff	ffff
------	------	------	------

説明: Wレジスタの内容とアドレス「f」のレジスタファイルとで符号なし乗算を実行する。16ビットの結果をPRODH:PRODLレジスタペアに格納する。PRODHには上位バイトを格納する。Wと「f」の内容はどちらも変化しない。

ステータスフラグは一切変化しない。

この演算では、オーバーフローもキャリーも不可能である。結果がゼロになる事はあっても、検出されない。

「a」=「0」の場合、アクセスバンクを選択する。「a」=「1」の場合、BSRを使ってGPRバンクを選択する(既定値)。

「a」=「0」で拡張命令セットを有効にしている場合、 $f \leq 95$  (5Fh)ならば、この命令は常にインデックス付きリテラルオフセットアドレッシングモードで実行される。詳細は、[セクション 26.2.3「インデックス付きリテラルオフセットモードでのバイト/ビット指向命令」](#)参照。

ワード数: 1

サイクル数: 1

Qサイクルの動作:

Q1	Q2	Q3	Q4
デコード	レジスタ「f」を読み出し	データを処理	PRODH:PRODLレジスタに書き込み

例: MULWF REG, 1

命令実行前  
W = C4h  
REG = B5h  
PRODH = ?  
PRODL = ?

命令実行後  
W = C4h  
REG = B5h  
PRODH = 8Ah  
PRODL = 94h

# PIC18F97J60 ファミリ

**NEGF**      **Negate f**

---

構文:        NEGF f{,a}

オペランド:   $0 \leq f \leq 255$   
 $a \in [0,1]$

動作:         $(\bar{f}) + 1 \rightarrow f$

影響を受けるステータス: N, OV, C, DC, Z

エンコード:

0110	110a	ffff	ffff
------	------	------	------

説明:        2の補数を使ってアドレス「f」の符号を反転する。結果をデータメモリのアドレス「f」に格納する。

「a」=「0」の場合、アクセスバンクを選択する。「a」=「1」の場合、BSRを使ってGPRバンクを選択する(既定値)。

「a」=「0」で拡張命令セットを有効にしている場合、 $f \leq 95$  (5Fh) ならば、この命令は常にインデックス付きリテラル オフセット アドレッシング モードで実行される。詳細は、[セクション 26.2.3 「インデックス付きリテラル オフセットモードでのバイト/ビット指向命令」](#) 参照。

ワード数:    1

サイクル数:  1

Q サイクルの動作:

Q1	Q2	Q3	Q4
デコード	レジスタ「f」を読み出し	データを処理	レジスタ「f」に書き込み

例:                NEGF    REG, 1

命令実行前  
REG    =    0011 1010 [3Ah]

命令実行後  
REG    =    1100 0110 [C6h]

**NOP**        **NOP**

---

構文:        NOP

オペランド:  なし

動作:        NOP

影響を受けるステータス: なし

エンコード:

0000	0000	0000	0000
1111	xxxx	xxxx	xxxx

説明:        何も実行しない。

ワード数:    1

サイクル数:  1

Q サイクルの動作:

Q1	Q2	Q3	Q4
デコード	NOP	NOP	NOP

例:

なし

# PIC18F97J60 ファミリ

## POP Pop Top of Return Stack

構文: POP

オペランド: なし

動作: (TOS) → bit bucket

影響を受ける

ステータス:

エンコード: 

0000	0000	0000	0110
------	------	------	------

説明: TOS の値をリターンスタックから取り出して破棄する。これにより、その前にリターンスタックにプッシュした値が TOS の値になる。  
この命令は、ユーザがリターンスタックを適切に管理してソフトウェアスタックを実装するためのものである。

ワード数: 1

サイクル数: 1

Q サイクルの動作:

Q1	Q2	Q3	Q4
デコード	NOP	TOS の値をポップ	NOP

例:

POP  
GOTO NEW

命令実行前  
TOS = 0031A2h  
Stack (1 level down) = 014332h

命令実行後  
TOS = 014332h  
PC = NEW

## PUSH Push Top of Return Stack

構文: PUSH

オペランド: なし

動作: (PC + 2) → TOS

影響を受ける

ステータス:

エンコード: 

0000	0000	0000	0101
------	------	------	------

説明: PC + 2 をリターンスタックの先頭にプッシュする。それまでの TOS 値はスタックの次の位置にプッシュされる。  
この命令を使って TOS を変更し、リターンスタックにプッシュする事で、ソフトウェアスタックを実装できる。

ワード数: 1

サイクル数: 1

Q サイクルの動作:

Q1	Q2	Q3	Q4
デコード	PC + 2 をリターンスタックにプッシュ	NOP	NOP

例:

PUSH

命令実行前  
TOS = 345Ah  
PC = 0124h

命令実行後  
PC = 0126h  
TOS = 0126h  
Stack (1 level down) = 345Ah



# PIC18F97J60 ファミリ

## RETFIE Return from Interrupt

構文: RETFIE {s}

オペランド:  $s \in [0,1]$

動作: (TOS) → PC,  
 1 → GIE/GIEH または PEIE/GIEL;  
 s = 1 の場合  
 (WS) → W,  
 (STATUS) → STATUS,  
 (BSRS) → BSR,  
 PCLATU, PCLATH は変更されない

影響を受ける GIE/GIEH, PEIE/GIEL.

ステータス:

エンコード: 

0000	0000	0001	000s
------	------	------	------

説明: 割り込みから戻る。スタックをポップして、TOS を PC に読み込む。割り込みを有効にするには、高優先度または低優先度のグローバル割り込みイネーブルビットをセットする。「s」= 1 の場合、シャドウレジスタ WS、STATUS、BSRS の内容を対応する W、STATUS、BSR の各レジスタに読み込む。「s」= 0 の場合、これらのレジスタは更新されない (既定値)。

ワード数: 1

サイクル数: 2

Q サイクルの動作:

	Q1	Q2	Q3	Q4
デコード	NOP	NOP	NOP	スタックから PC をポップ GIEH または GIEL をセット
	NOP	NOP	NOP	NOP

例:

```
RETFIE 1
```

割り込み後

```
PC           = TOS
W            = WS
BSR          = BSR
STATUS       = STATUS
GIE/GIEH, PEIE/GIEL = 1
```

## RETLW Return Literal to W

構文: RETLW k

オペランド:  $0 \leq k \leq 255$

動作: k → W,  
 (TOS) → PC,  
 PCLATU, PCLATH は変更されない

影響を受ける なし

ステータス:

エンコード: 

0000	1100	kkkk	kkkk
------	------	------	------

説明: W レジスタに 8 ビットのリテラル「k」を読み込む。TOS (リターンアドレス) を PC に読み込む。上位アドレスラッチ (PCLATH) は変化しない。

ワード数: 1

サイクル数: 2

Q サイクルの動作:

	Q1	Q2	Q3	Q4
デコード	リテラル「k」を読み出し	データを処理	スタックから PC をポップし、W に書き込み	NOP
	NOP	NOP	NOP	NOP

例:

```
CALL TABLE ; W contains table
              ; offset value
              ; W now has
              ; table value
```

```
TABLE
```

```
ADDWF PCL ; W = offset
RETLW k0 ; Begin table
RETLW k1 ;
:
:
RETLW kn ; End of table
```

```
命令実行前
W = 07h
```

```
命令実行後
W = value of kn
```

# PIC18F97J60 ファミリ

## RETURN Return from Subroutine

構文: RETURN {s}

オペランド:  $s \in [0,1]$

動作: (TOS) → PC;  
 $s = 1$  の場合  
 (WS) → W,  
 (STATUS) → STATUS,  
 (BSRS) → BSR,  
 PCLATU, PCLATH は変更されない

影響を受けるステータス: なし

エンコード: 

0000	0000	0001	001s
------	------	------	------

説明: サブルーチンから戻る。スタックをポップし、TOS を PC に読み込む。「s」= 1 の場合、シャドーレジスタ WS、STATUS、BSRS の内容をそれぞれ W、STATUS、BSR の各レジスタに読み込む。「s」= 0 の場合、これらのレジスタは更新されない (既定値)。

ワード数: 1

サイクル数: 2

Q サイクルの動作:

	Q1	Q2	Q3	Q4
デコード	NOP	NOP	データを処理	スタックから PC をポップ
	NOP	NOP	NOP	NOP

例: RETURN

命令実行後  
 PC = TOS

## RLCF Rotate Left f through Carry

構文: RLCF f{,d{,a}}

オペランド:  $0 \leq f \leq 255$   
 $d \in [0,1]$   
 $a \in [0,1]$

動作: (f<n>) → dest<n + 1>,  
 (f<7>) → C,  
 (C) → dest<0>

影響を受けるステータス: C, N, Z

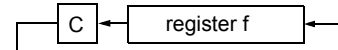
エンコード: 

0011	01da	ffff	ffff
------	------	------	------

説明: レジスタ「f」の内容を、キャリーフラグを通して左回りに 1 ビット移動させる。「d」= 「0」の場合、結果を W レジスタに格納する。「d」= 「1」の場合、結果をレジスタ「f」に書き戻す (既定値)。

「a」= 「0」の場合、アクセスバンクを選択する。「a」= 「1」の場合、BSR を使って GPR バンクを選択する (既定値)。

「a」= 「0」で拡張命令セットを有効にしている場合、 $f \leq 95$  (5Fh) ならば、この命令は常にインデックス付きリテラル オフセット アドレッシング モードで実行される。詳細は、[セクション 26.2.3 「インデックス付きリテラル オフセットモードでのバイト/ビット指向命令」](#) 参照。



ワード数: 1

サイクル数: 1

Q サイクルの動作:

	Q1	Q2	Q3	Q4
デコード		レジスタ「f」を読み出し	データを処理	格納先に書き込み

例: RLCF REG, 0, 0

命令実行前  
 REG = 1110 0110  
 C = 0

命令実行後  
 REG = 1110 0110  
 W = 1100 1100  
 C = 1

# PIC18F97J60 ファミリ

## RLNCF Rotate Left f (no carry)

構文: RLNCF f{,d{,a}}

オペランド:  $0 \leq f \leq 255$   
 $d \in [0,1]$   
 $a \in [0,1]$

動作:  $(f<n>) \rightarrow \text{dest}<n+1>$ ,  
 $(f<7>) \rightarrow \text{dest}<0>$

影響を受けるステータス: N, Z

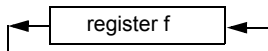
エンコード:

0100	01da	ffff	ffff
------	------	------	------

説明: レジスタ「f」の内容を左回りに1ビット移動させる。「d」=「0」の場合、結果をWレジスタに格納する。「d」=「1」の場合、結果をレジスタ「f」に書き戻す(既定値)。

「a」=「0」の場合、アクセスバンクを選択する。「a」=「1」の場合、BSRを使ってGPRバンクを選択する(既定値)。

「a」=「0」で拡張命令セットを有効にしている場合、 $f \leq 95$  (5Fh) ならば、この命令は常にインデックス付きリテラルオフセットアドレッシングモードで実行される。詳細は、[セクション 26.2.3 「インデックス付きリテラルオフセットモードでのバイト/ビット指向命令」](#) 参照。



ワード数: 1

サイクル数: 1

Q サイクルの動作:

Q1	Q2	Q3	Q4
デコード	レジスタ「f」を読み出し	データを処理	格納先に書き込み

例: RLNCF REG, 1, 0

命令実行前  
 REG = 1010 1011  
 命令実行後  
 REG = 0101 0111

## RRCF Rotate Right f through Carry

構文: RRCF f{,d{,a}}

オペランド:  $0 \leq f \leq 255$   
 $d \in [0,1]$   
 $a \in [0,1]$

動作:  $(f<n>) \rightarrow \text{dest}<n-1>$ ,  
 $(f<0>) \rightarrow C$ ,  
 $(C) \rightarrow \text{dest}<7>$

影響を受けるステータス: C, N, Z

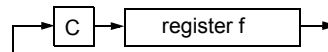
エンコード:

0011	00da	ffff	ffff
------	------	------	------

説明: レジスタ「f」の内容を、キャリーフラグを通して右回りに1ビット移動させる。「d」=「0」の場合、結果をWレジスタに格納する。「d」=「1」の場合、結果をレジスタ「f」に書き戻す(既定値)。

「a」=「0」の場合、アクセスバンクを選択する。「a」=「1」の場合、BSRを使ってGPRバンクを選択する(既定値)。

「a」=「0」で拡張命令セットを有効にしている場合、 $f \leq 95$  (5Fh) ならば、この命令は常にインデックス付きリテラルオフセットアドレッシングモードで実行される。詳細は、[セクション 26.2.3 「インデックス付きリテラルオフセットモードでのバイト/ビット指向命令」](#) 参照。



ワード数: 1

サイクル数: 1

Q サイクルの動作:

Q1	Q2	Q3	Q4
デコード	レジスタ「f」を読み出し	データを処理	格納先に書き込み

例: RRCF REG, 0, 0

命令実行前  
 REG = 1110 0110  
 C = 0  
 命令実行後  
 REG = 1110 0110  
 W = 0111 0011  
 C = 0



# PIC18F97J60 ファミリ

## RRNCF Rotate Right f (no carry)

構文: RRNCF f{,d{,a}}

オペランド:  $0 \leq f \leq 255$   
 $d \in [0,1]$   
 $a \in [0,1]$

動作:  $(f \langle n \rangle) \rightarrow \text{dest} \langle n - 1 \rangle$ ,  
 $(f \langle 0 \rangle) \rightarrow \text{dest} \langle 7 \rangle$

影響を受けるステータス: N, Z

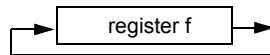
エンコード: 

0100	00da	ffff	ffff
------	------	------	------

説明: レジスタ「f」の内容を右回りに1ビット移動させる。「d」=「0」の場合、結果をWレジスタに格納する。「d」=「1」の場合、結果をレジスタ「f」に書き戻す(既定値)。

「a」=「0」の場合、BSRの値に関わらずアクセスバンクを選択する。「a」=「1」の場合、BSRの値でバンクを選択する(既定値)。

「a」=「0」で拡張命令セットを有効にしている場合、 $f \leq 95$  (5Fh) ならば、この命令は常にインデックス付きリテラルオフセットアドレッシングモードで実行される。詳細は、[セクション 26.2.3 「インデックス付きリテラルオフセットモードでのバイト/ビット指向命令」](#) 参照。



ワード数: 1

サイクル数: 1

Qサイクルの動作:

Q1	Q2	Q3	Q4
デコード	レジスタ「f」を読み出し	データを処理	格納先に書き込み

例 1: RRNCF REG, 1, 0

命令実行前  
REG = 1101 0111  
命令実行後  
REG = 1110 1011

例 2: RRNCF REG, 0, 0

命令実行前  
W = ?  
REG = 1101 0111  
命令実行後  
W = 1110 1011  
REG = 1101 0111

## SETF Set f

構文: SETF f{,a}

オペランド:  $0 \leq f \leq 255$   
 $a \in [0,1]$

動作: FFh  $\rightarrow$  f

影響を受けるステータス: なし

エンコード: 

0110	100a	ffff	ffff
------	------	------	------

説明: 指定したレジスタの内容をFFhにセットする。

「a」=「0」の場合、アクセスバンクを選択する。「a」=「1」の場合、BSRを使ってGPRバンクを選択する(既定値)。

「a」=「0」で拡張命令セットを有効にしている場合、 $f \leq 95$  (5Fh) ならば、この命令は常にインデックス付きリテラルオフセットアドレッシングモードで実行される。詳細は、[セクション 26.2.3 「インデックス付きリテラルオフセットモードでのバイト/ビット指向命令」](#) 参照。

ワード数: 1

サイクル数: 1

Qサイクルの動作:

Q1	Q2	Q3	Q4
デコード	レジスタ「f」を読み出し	データを処理	レジスタ「f」に書き込み

例: SETF REG, 1

命令実行前  
REG = 5Ah  
命令実行後  
REG = FFh

# PIC18F97J60 ファミリ

## SLEEP Enter Sleep Mode

構文: SLEEP

オペランド: なし

動作: 00h → WDT,  
0 → WDT postscaler,  
1 →  $\overline{TO}$ ,  
0 → PD

影響を受けるステータス:  $\overline{TO}$ , PD

エンコード:

0000	0000	0000	0011
------	------	------	------

説明: パワーダウンステータスビット ( $\overline{PD}$ ) をクリアする。タイムアウトステータスビット ( $\overline{TO}$ ) をセットする。ウォッチドッグタイマをポストスケラを含めてクリアする。  
オシレータを停止してプロセッサをスリープにする。

ワード数: 1

サイクル数: 1

Q サイクルの動作:

Q1	Q2	Q3	Q4
デコード	NOP	データを処理	スリープに移行する

例: SLEEP

命令実行前  
 $\overline{TO}$  = ?  
PD = ?

命令実行後  
 $\overline{TO}$  = 1 +  
PD = 0

† WDTによって復帰した場合、このビットはクリアされます。

## SUBFWB Subtract f from W with Borrow

構文: SUBFWB f{,d{,a}}

オペランド:  $0 \leq f \leq 255$   
 $d \in [0,1]$   
 $a \in [0,1]$

動作:  $(W) - (f) - (\overline{C}) \rightarrow \text{dest}$

影響を受けるステータス: N, OV, C, DC, Z

エンコード:

0101	01da	ffff	ffff
------	------	------	------

説明: Wレジスタからレジスタ「f」とキャリーフラグ(ボロー)を減算する(2の補数法)。「d」=「0」の場合、結果をWレジスタに格納する。「d」=「1」の場合、結果をレジスタ「f」に格納する(既定値)。

「a」=「0」の場合、アクセスバンクを選択する。「a」=「1」の場合、BSRを使ってGPRバンクを選択する(既定値)。

「a」=「0」で拡張命令セットを有効にしている場合、 $f \leq 95$  (5Fh) ならば、この命令は常にインデックス付きリテラルオフセットアドレッシングモードで実行される。詳細は、[セクション 26.2.3「インデックス付きリテラルオフセットモードでのバイト/ビット指向命令」](#) 参照。

ワード数: 1

サイクル数: 1

Q サイクルの動作:

Q1	Q2	Q3	Q4
デコード	レジスタ「f」を読み出し	データを処理	格納先に書き込み

例1: SUBFWB REG, 1, 0

命令実行前  
REG = 3  
W = 2  
C = 1

命令実行後  
REG = FF  
W = 2  
C = 0  
Z = 0  
N = 1 ; result is negative

例2: SUBFWB REG, 0, 0

命令実行前  
REG = 2  
W = 5  
C = 1

命令実行後  
REG = 2  
W = 3  
C = 1  
Z = 0  
N = 0 ; result is positive

例3: SUBFWB REG, 1, 0

命令実行前  
REG = 1  
W = 2  
C = 0

命令実行後  
REG = 0  
W = 2  
C = 1  
Z = 1 ; result is zero  
N = 0

# PIC18F97J60 ファミリ

## SUBLW Subtract W from Literal

構文: SUBLW k  
 オペランド:  $0 \leq k \leq 255$   
 動作:  $k - (W) \rightarrow W$   
 影響を受けるステータス: N, OV, C, DC, Z  
 エンコード:

0000	1000	kkkk	kkkk
------	------	------	------

説明: 8ビットのリテラル「k」から W レジスタの内容を減算する。結果を W レジスタに格納する。

ワード数: 1

サイクル数: 1

Q サイクルの動作:

Q1	Q2	Q3	Q4
デコード	リテラル「k」を読み出し	データを処理	W に書き込み

例 1: SUBLW 02h

命令実行前  
 W = 01h  
 C = ?  
 命令実行後  
 W = 01h  
 C = 1 ; result is positive  
 Z = 0  
 N = 0

例 2: SUBLW 02h

命令実行前  
 W = 02h  
 C = ?  
 命令実行後  
 W = 00h  
 C = 1 ; result is zero  
 Z = 1  
 N = 0

例 3: SUBLW 02h

命令実行前  
 W = 03h  
 C = ?  
 命令実行後  
 W = FFh ; (2's complement)  
 C = 0 ; result is negative  
 Z = 0  
 N = 1

## SUBWF Subtract W from f

構文: SUBWF f{,d{,a}}

オペランド:  $0 \leq f \leq 255$   
 $d \in [0,1]$   
 $a \in [0,1]$

動作:  $(f) - (W) \rightarrow \text{dest}$

影響を受けるステータス: N, OV, C, DC, Z  
 エンコード:

0101	11da	ffff	ffff
------	------	------	------

説明: レジスタ「f」から W レジスタの内容を減算する (2 の補数法)。  
 「d」=「0」の場合、結果を W レジスタに格納する。「d」=「1」の場合、結果をレジスタ「f」に書き戻す (既定値)。  
 「a」=「0」の場合、アクセスバンクを選択する。「a」=「1」の場合、BSR を使って GPR バンクを選択する (既定値)。  
 「a」=「0」で拡張命令セットを有効にしている場合、 $f \leq 95$  (5Fh) ならば、この命令は常にインデックス付きリテラル オフセット アドレッシング モードで実行される。詳細は、[セクション 26.2.3 「インデックス付きリテラル オフセットモードでのバイト/ビット指向命令」](#) 参照。

ワード数: 1

サイクル数: 1

Q サイクルの動作:

Q1	Q2	Q3	Q4
デコード	レジスタ「f」を読み出し	データを処理	格納先に書き込み

例 1: SUBWF REG, 1, 0

命令実行前  
 REG = 3  
 W = 2  
 C = ?  
 命令実行後  
 REG = 1  
 W = 2  
 C = 1 ; result is positive  
 Z = 0  
 N = 0

例 2: SUBWF REG, 0, 0

命令実行前  
 REG = 2  
 W = 2  
 C = ?  
 命令実行後  
 REG = 2  
 W = 0  
 C = 1 ; result is zero  
 Z = 1  
 N = 0

例 3: SUBWF REG, 1, 0

命令実行前  
 REG = 1  
 W = 2  
 C = ?  
 命令実行後  
 REG = FFh ; (2's complement)  
 W = 2  
 C = 0 ; result is negative  
 Z = 0  
 N = 1

# PIC18F97J60 ファミリ

## SUBWFB Subtract W from f with Borrow

構文: SUBWFB f{,d{,a}}

オペランド:  $0 \leq f \leq 255$   
 $d \in [0,1]$   
 $a \in [0,1]$

動作:  $(f) - (W) - (\bar{C}) \rightarrow \text{dest}$

影響を受けるステータス: N, OV, C, DC, Z

エンコード:

0101	10da	ffff	ffff
------	------	------	------

説明: レジスタ「f」からWレジスタとキャリーフラグ(ポロー)を減算する(2の補数法)。「d」=「0」の場合、結果をWレジスタに格納する。「d」=「1」の場合、結果をレジスタ「f」に書き戻す(既定値)。「a」=「0」の場合、アクセスバンクを選択する。「a」=「1」の場合、BSRを使ってGPRバンクを選択する(既定値)。「a」=「0」で拡張命令セットを有効にしている場合、 $f \leq 95$  (5Fh)ならば、この命令は常にインデックス付きリテラル オフセット アドレッシング モードで実行される。詳細は、[セクション 26.2.3「インデックス付きリテラル オフセットモードでのバイト/ビット指向命令」](#) 参照。

ワード数: 1

サイクル数: 1

Q サイクルの動作:

Q1	Q2	Q3	Q4
デコード	レジスタ「f」を読み出し	データを処理	格納先に書き込み

例 1: SUBWFB REG, 1, 0

命令実行前  
 REG = 19h (0001 1001)  
 W = 0Dh (0000 1101)  
 C = 1  
 命令実行後  
 REG = 0Ch (0000 1011)  
 W = 0Dh (0000 1101)  
 C = 1  
 Z = 0  
 N = 0 ; result is positive

例 2: SUBWFB REG, 0, 0

命令実行前  
 REG = 1Bh (0001 1011)  
 W = 1Ah (0001 1010)  
 C = 0  
 命令実行後  
 REG = 1Bh (0001 1011)  
 W = 00h  
 C = 1  
 Z = 1 ; result is zero  
 N = 0

例 3: SUBWFB REG, 1, 0

命令実行前  
 REG = 03h (0000 0011)  
 W = 0Eh (0000 1101)  
 C = 1  
 命令実行後  
 REG = F5h (1111 0100)  
 ; [2's comp]  
 W = 0Eh (0000 1101)  
 C = 0  
 Z = 0  
 N = 1 ; result is negative

## SWAPF Swap f

構文: SWAPF f{,d{,a}}

オペランド:  $0 \leq f \leq 255$   
 $d \in [0,1]$   
 $a \in [0,1]$

動作:  $(f<3:0>) \rightarrow \text{dest}<7:4>$ ,  
 $(f<7:4>) \rightarrow \text{dest}<3:0>$

影響を受けるステータス: なし

エンコード:

0011	10da	ffff	ffff
------	------	------	------

説明: レジスタ「f」の上位ニブルと下位ニブルを入れ換える。「d」=「0」の場合、結果をWレジスタに格納する。「d」=「1」の場合、結果をレジスタ「f」に格納する(既定値)。「a」=「0」の場合、アクセスバンクを選択する。「a」=「1」の場合、BSRを使ってGPRバンクを選択する(既定値)。「a」=「0」で拡張命令セットを有効にしている場合、 $f \leq 95$  (5Fh)ならば、この命令は常にインデックス付きリテラル オフセット アドレッシング モードで実行される。詳細は、[セクション 26.2.3「インデックス付きリテラル オフセットモードでのバイト/ビット指向命令」](#) 参照。

ワード数: 1

サイクル数: 1

Q サイクルの動作:

Q1	Q2	Q3	Q4
デコード	レジスタ「f」を読み出し	データを処理	格納先に書き込み

例: SWAPF REG, 1, 0

命令実行前  
 REG = 53h  
 命令実行後  
 REG = 35h

# PIC18F97J60 ファミリ

## TBLRD Table Read

構文: TBLRD (\*; \*+; \*-; +\*)

オペランド: なし

動作: if TBLRD\*,  
(Prog Mem (TBLPTR)) → TABLAT;  
TBLPTR – No Change  
if TBLRD \*+,  
(Prog Mem (TBLPTR)) → TABLAT;  
(TBLPTR) + 1 → TBLPTR  
if TBLRD \*-,  
(Prog Mem (TBLPTR)) → TABLAT;  
(TBLPTR) – 1 → TBLPTR  
if TBLRD +\*,  
(TBLPTR) + 1 → TBLPTR;  
(Prog Mem (TBLPTR)) → TABLAT

影響を受けるステータス: なし

エンコード:

0000	0000	0000	10nn nn=0 * =1 *+ =2 *- =3 +*
------	------	------	---

説明: この命令は、プログラムメモリ (P.M.) の内容の読み出しに使う。プログラムメモリのアドレスは、テーブルポインタ (TBLPTR) と呼ばれるポインタを使って指定する。

TBLPTR (21 ビットのポインタ) は、プログラムメモリをバイト単位で指定する。TBLPTR では、2MB のアドレス範囲を指定できる。

- TBLPTR[0] = 0: プログラムメモリ ワードの LSB
- TBLPTR[0] = 1: プログラムメモリ ワードの MSB

TBLRD 命令では、TBLPTR の値を以下のように変更できる。

- 変更なし
- ポストインクリメント
- ポストデクリメント
- プリインクリメント

ワード数: 1

サイクル数: 2

Q サイクルの動作:

	Q1	Q2	Q3	Q4
デコード	NOP	NOP	NOP	NOP
NOP	NOP	NOP (プログラムメモリを読み出し)	NOP	NOP (TABLAT に書き込み)

## TBLRD Table Read ( 続き )

例 1: TBLRD \*+ ;

命令実行前  
TABLAT = 55h  
TBLPTR = 00A356h  
MEMORY(00A356h) = 34h

命令実行後  
TABLAT = 34h  
TBLPTR = 00A357h

例 2: TBLRD \*+ ;

命令実行前  
TABLAT = AAh  
TBLPTR = 01A357h  
MEMORY(01A357h) = 12h  
MEMORY(01A358h) = 34h

命令実行後  
TABLAT = 34h  
TBLPTR = 01A358h

# PIC18F97J60 ファミリ

## TBLWT Table Write

構文: TBLWT (\*; \*+; \*-; +\*)

オペランド: なし

動作:

```

if TBLWT*,
(TABLAT) → Holding Register;
TBLPTR – No Change
if TBLWT*+,
(TABLAT) → Holding Register;
(TBLPTR) + 1 → TBLPTR
if TBLWT*- ,
(TABLAT) → Holding Register;
(TBLPTR) – 1 → TBLPTR
if TBLWT+*,
(TBLPTR) + 1 → TBLPTR;
(TABLAT) → Holding Register
    
```

影響を受けるステータス: なし

エンコード:

0000	0000	0000	11nn nn=0 * =1 *+ =2 *- =3 +*
------	------	------	---

説明: この命令は、TBLPTR の下位 3 ビットを使って 8 つの保持レジスタのどれに TABLAT を書き込むかを判断する。保持レジスタは、プログラムメモリ (P.M.) の内容を書き込むために使う (フラッシュメモリへの書き込みの詳細は、[セクション 6.0「メモリ構成」](#)参照)。

TBLPTR (21 ビットのポインタ) は、プログラムメモリをバイト単位で指定する。TBLPTR では、2MB のアドレス範囲を指定できる。TBLPTR の最下位ビットで、プログラムメモリのどのアドレスのバイトにアクセスするかを選択する。

TBLPTR[0] = 0: プログラムメモリワードの LSB

TBLPTR[0] = 1: プログラムメモリワードの MSB

TBLWT 命令では、TBLPTR の値を以下のように変更できる。

- 変更なし
- ポストインクリメント
- ポストデクリメント
- プリインクリメント

ワード数: 1

サイクル数: 2

Q サイクルの動作:

	Q1	Q2	Q3	Q4
デコード	NOP	NOP	NOP	NOP
NOP	NOP	NOP (TABLAT を読み出し)	NOP	NOP (保持レジスタに書き込み)

## TBLWT Table Write (続き)

例 1: TBLWT \*+;

命令実行前

```

TABLAT           = 55h
TBLPTR          = 00A356h
HOLDING REGISTER (00A356h) = FFh
    
```

命令実行 (テーブル書き込み完了) 後

```

TABLAT           = 55h
TBLPTR          = 00A357h
HOLDING REGISTER (00A356h) = 55h
    
```

例 2: TBLWT +-;

命令実行前

```

TABLAT           = 34h
TBLPTR          = 01389Ah
HOLDING REGISTER (01389Ah) = FFh
HOLDING REGISTER (01389Bh) = FFh
    
```

命令実行 (テーブル書き込み完了) 後

```

TABLAT           = 34h
TBLPTR          = 01389Bh
HOLDING REGISTER (01389Ah) = FFh
HOLDING REGISTER (01389Bh) = 34h
    
```

# PIC18F97J60 ファミリ

## TSTFSZ Test f, Skip if 0

構文: TSTFSZ f{,a}

オペランド:  $0 \leq f \leq 255$   
 $a \in [0,1]$

動作:  $f = 0$  の場合にスキップ

影響を受ける ステータス: なし

エンコード: 

0110	011a	ffff	ffff
------	------	------	------

説明: 「f」= 0 の場合、現在の命令実行中にフェッチした次の命令を破棄し、NOP 命令を実行して 2 サイクル命令とする。

「a」= 「0」の場合、アクセスバンクを選択する。「a」= 「1」の場合、BSR を使って GPR バンクを選択する (既定値)。

「a」= 「0」で拡張命令セットを有効にしている場合、 $f \leq 95$  (5Fh) ならば、この命令は常にインデックス付きリテラル オフセット アドレッシング モードで実行される。詳細は、[セクション 26.2.3 「インデックス付きリテラル オフセットモードでのバイト/ビット指向命令」](#) 参照。

ワード数: 1

サイクル数: 1(2)

**Note:** スキップ先の命令が 2 ワード命令の場合は 3 サイクルである。

Q サイクルの動作:

Q1	Q2	Q3	Q4
デコード	レジスタ「f」を読み出し	データを処理	NOP

スキップする場合:

Q1	Q2	Q3	Q4
NOP	NOP	NOP	NOP

スキップ先が 2 ワード命令の場合:

Q1	Q2	Q3	Q4
NOP	NOP	NOP	NOP
NOP	NOP	NOP	NOP

例: 

```
HERE    TSTFSZ  CNT, 1
NZERO   :
ZERO    :
```

命令実行前  
PC = Address (HERE)  
命令実行後  
If CNT = 00h,  
PC = Address (ZERO)  
If CNT ≠ 00h,  
PC = Address (NZERO)

## XORLW Exclusive OR Literal with W

構文: XORLW k

オペランド:  $0 \leq k \leq 255$

動作:  $(W) .XOR. k \rightarrow W$

影響を受ける ステータス: N, Z

エンコード:

0000	1010	kkkk	kkkk
------	------	------	------

説明: W レジスタの内容と 8 ビットのリテラル「k」を XOR 演算する。結果を W レジスタに格納する。

ワード数: 1

サイクル数: 1

Q サイクルの動作:

Q1	Q2	Q3	Q4
デコード	リテラル「k」を読み出し	データを処理	W に書き込み

例: 

```
XORLW  0AFh
```

命令実行前  
W = B5h  
命令実行後  
W = 1Ah

# PIC18F97J60 ファミリ

## XORWF Exclusive OR W with f

構文: XORWF f{,d{,a}}

オペランド:  $0 \leq f \leq 255$   
 $d \in [0,1]$   
 $a \in [0,1]$

動作: (W).XOR.(f) → dest

影響を受けるステータス: N, Z

エンコード:

0001	10da	ffff	ffff
------	------	------	------

説明: Wレジスタとレジスタ「f」の内容をXOR演算する。「d」=「0」の場合、結果をWレジスタに格納する。「d」=「1」の場合、結果をレジスタ「f」に書き戻す(既定値)。

「a」=「0」の場合、アクセスバンクを選択する。「a」=「1」の場合、BSRを使ってGPRバンクを選択する(既定値)。

「a」=「0」で拡張命令セットを有効にしている場合、 $f \leq 95$  (5Fh) ならば、この命令は常にインデックス付きリテラルオフセットアドレッシングモードで実行される。詳細は、[セクション 26.2.3「インデックス付きリテラルオフセットモードでのバイト/ビット指向命令」](#)参照。

ワード数: 1

サイクル数: 1

Qサイクルの動作:

Q1	Q2	Q3	Q4
デコード	レジスタ「f」を読み出し	データを処理	格納先に書き込み

例: XORWF REG, 1, 0

命令実行前  
REG = AFh  
W = B5h  
命令実行後  
REG = 1Ah  
W = B5h



## 26.2 拡張命令セット

PIC18F97J60 ファミリは、75 命令からなる PIC18 標準命令セットに加え、拡張命令セットもサポートしています。間接アドレッシングおよびインデックス付きアドレッシング動作を強化する 8 つの拡張命令を追加すると共に、多くの PIC18 標準命令向けにインデックス付きリテラル オフセット アドレッシングを実装しました。

拡張命令セットによる追加機能は、未プログラム デバイスの既定値では有効に設定されています。ユーザは、プログラミング時に XINST コンフィグレーションビットを正しくセットまたはクリアして、これらの機能の有効 / 無効を設定する必要があります。

拡張命令セットの命令は全てリテラル命令に分類され、FSR を操作するか、FSR を使ってインデックス付きアドレッシングを行います。拡張命令のうち、ADDFSR と SUBFSR の 2 つには、それぞれ FSR2 を使う命令も用意しています。これらの命令 (ADDULNK と SUBULNK) では、実行後に自動的にリターンできます。

拡張命令は、主に C 言語等の高級言語で作成した再入可能プログラムコード (すなわち、再帰コードまたはソフトウェアスタックを使うコード) を最適化するために実装しています。特に、これらの命令を使うと、データ構造に対する一部の演算を高級言語で効率的に実行できます。例えば以下のような場合です。

- サブルーチン開始時と終了時における、ソフトウェアスタック空間の動的な割り当てと割り当て解除
- 関数ポインタの呼び出し
- ソフトウェアスタックポインタの操作
- ソフトウェアスタック内の変数の操作

表 26-3 に、拡張命令セットの命令一覧を示します。各命令の詳細は、[セクション 26.2.2 「拡張命令セット」](#) で説明します。オペコード フィールドの説明は、[表 26-1 \(376 ページ\)](#) に示した PIC18 標準命令セットと共通です。

**Note:** 拡張命令セットとインデックス付きリテラル オフセット アドレッシング モードは C 言語で作成したアプリケーションの最適化を目的としています。これらの命令をアセンブラで直接使う事は想定していません。ただし、コンパイラが生成したコードをチェックする際の参考として、各命令の構文を説明しています。

### 26.2.1 拡張命令の構文

拡張命令のほとんどはインデックス付き引数を使い、ファイルセレクトレジスタの 1 つとオフセットを使って送信元または格納先のレジスタを指定します。命令の引数をインデックス付きアドレッシングに使う場合、引数を角カッコ (「[]」) で囲みます。これによって、その引数をインデックスまたはオフセットとして使う事を明示します。インデックスまたはオフセットの値を角カッコで囲まないと、MPASM™ アセンブラでエラーが発生します。

拡張命令セットを有効にした場合、バイト指向命令とビット指向命令のインデックス引数を示すためにも角カッコを使います。これ以外にも構文の変更があります。詳細は、[セクション 26.2.3.1 「拡張命令有効時の PIC18 標準コマンドの構文」](#) を参照してください。

**Note:** 以前の文書では、PIC18 とそれ以前の命令セットにおけるオプションの引数を角カッコで表していました。今後は、オプションの引数は中カッコ (「{ }」) で表します。

表 26-3: PIC18 拡張命令セット

ニーモニック、オペランド	説明	サイクル数	16 ビット命令ワード				影響を受けるステータス
			MSb		LSb		
ADDFSR f, k	Add Literal to FSR	1	1110	1000	ffkk	kkkk	なし
ADDULNK k	Add Literal to FSR2 and Return	2	1110	1000	11kk	kkkk	なし
CALLW	Call Subroutine using WREG	2	0000	0000	0001	0100	なし
MOVSF z <sub>s</sub> , f <sub>d</sub>	Move z <sub>s</sub> (source) to 1st word f <sub>d</sub> (destination) 2nd word	2	1110	1011	0zzz	zzzz	なし
MOVSS z <sub>s</sub> , z <sub>d</sub>	Move z <sub>s</sub> (source) to 1st word z <sub>d</sub> (destination) 2nd word	2	1110	1011	1zzz	zzzz	なし
PUSHL k	Store Literal at FSR2, Decrement FSR2	1	1110	1010	kkkk	kkkk	なし
SUBFSR f, k	Subtract Literal from FSR	1	1110	1001	ffkk	kkkk	なし
SUBULNK k	Subtract Literal from FSR2 and Return	2	1110	1001	11kk	kkkk	なし

# PIC18F97J60 ファミリ

## 26.2.2 拡張命令セット

### ADDFSR Add Literal to FSR

構文: ADDFSR f, k

オペランド:  $0 \leq k \leq 63$   
 $f \in [0, 1, 2]$

動作:  $FSR(f) + k \rightarrow FSR(f)$

影響を受ける

ステータス: なし

エンコード: 

1110	1000	ffkk	kkkk
------	------	------	------

説明: 6ビットのリテラル「k」を、「f」で指定したFSRの内容に加算する。

ワード数: 1

サイクル数: 1

Qサイクルの動作:

Q1	Q2	Q3	Q4
デコード	リテラル「k」を読み出し	データを処理	FSRに書き込み

例: ADDFSR 2, 23h

命令実行前  
FSR2 = 03FFh

命令実行後  
FSR2 = 0422h

### ADDULNK Add Literal to FSR2 and Return

構文: ADDULNK k

オペランド:  $0 \leq k \leq 63$

動作:  $FSR2 + k \rightarrow FSR2,$   
 $(TOS) \rightarrow PC$

影響を受ける

ステータス: なし

エンコード: 

1110	1000	11kk	kkkk
------	------	------	------

説明: 6ビットのリテラル「k」をFSR2の内容に加算する。次に、PCにTOSを読み込んでRETURNを実行する。

この命令の実行は2サイクルで、2サイクル目にはNOPを実行する。

これは、ADDFSR命令で $f=3$ (2進数「11」)である特殊な場合と考える事ができる。この命令はFSR2に対してのみ実行できる。

ワード数: 1

サイクル数: 2

Qサイクルの動作:

Q1	Q2	Q3	Q4
デコード	リテラル「k」を読み出し	データを処理	FSRに書き込み
NOP	NOP	NOP	NOP

例: ADDULNK 23h

命令実行前  
FSR2 = 03FFh  
PC = 0100h

命令実行後  
FSR2 = 0422h  
PC = (TOS)

**Note:** PIC18の命令は全て、命令ニーモニックの前にオプションの引数としてラベルを指定する事によってシンボリックアドレッシングが可能です。この場合、命令の形式は「{ラベル}命令 引数」です。

# PIC18F97J60 ファミリ

## CALLW Subroutine Call using WREG

構文: CALLW  
 オペランド: なし  
 動作: (PC + 2) → TOS,  
 (W) → PCL,  
 (PCLATH) → PCH,  
 (PCLATU) → PCU

影響を受けるステータス: なし

エンコード:

0000	0000	0001	0100
------	------	------	------

説明: まず、リターンアドレス (PC + 2) をリターンスタックにプッシュする。次に、Wレジスタの内容を PCL に書き込む (既存の値は破棄される)。さらに、PCLATH と PCLATU の内容を PCH と PCU にそれぞれラッチする。  
 2 サイクル目は NOP 命令として実行され、その間に次の命令をフェッチする。  
 CALL とは異なり、W、STATUS、BSR を更新する事はできない。

ワード数: 1

サイクル数: 2

Q サイクルの動作:

Q1	Q2	Q3	Q4
デコード	WREG を読み出し	PC をスタックにプッシュ	NOP
NOP	NOP	NOP	NOP

例:

```

HERE CALLW
命令実行前
PC = address (HERE)
PCLATH = 10h
PCLATU = 00h
W = 06h
命令実行後
PC = 001006h
TOS = address (HERE + 2)
PCLATH = 10h
PCLATU = 00h
W = 06h
  
```

## MOVSF Move Indexed to f

構文: MOVSF [z<sub>s</sub>], f<sub>d</sub>  
 オペランド: 0 ≤ z<sub>s</sub> ≤ 127  
 0 ≤ f<sub>d</sub> ≤ 4095

動作: ((FSR2) + z<sub>s</sub>) → f<sub>d</sub>

影響を受けるステータス: なし

エンコード:  
 1 ワード目 (送信元)  
 2 ワード目 (格納先)

1110	1011	0zzz	zzzz <sub>s</sub>
1111	ffff	ffff	ffff <sub>d</sub>

説明: 送信元レジスタの内容を格納先レジスタ「f<sub>d</sub>」に移動する。送信元レジスタの実際のアドレスは、1 ワード目にある 7 ビットのリテラル オフセット「z<sub>s</sub>」を FSR2 の値に加算して求める。格納先レジスタのアドレスは、2 ワード目にある 12 ビットのリテラル「f<sub>d</sub>」で指定する。どちらのアドレスも、4096 バイトのデータ空間 (000h ~ FFFh) の任意の位置を指定できる。

MOVSF 命令では、格納先レジスタに PCL、TOSU、TOSH、TOSL を使えない。

上記で求めた送信元アドレスが間接アドレッシングレジスタを指し示している場合、戻り値は 00h となる。

ワード数: 2

サイクル数: 2

Q サイクルの動作:

Q1	Q2	Q3	Q4
デコード	送信元アドレスを 求める	送信元 アドレスを 求める	送信元 レジスタを 読み出す
デコード	NOP	NOP	レジスタ「f」 (格納先)に 書き込み
	ダミー 読み出し なし		

例:

```

MOVSF [05h], REG2
命令実行前
FSR2 = 80h
Contents of 85h REG2 = 11h
命令実行後
FSR2 = 80h
Contents of 85h REG2 = 33h
  
```

# PIC18F97J60 ファミリ

## MOVSS Move Indexed to Indexed

構文: MOVSS [z<sub>s</sub>], [z<sub>d</sub>]  
 オペランド: 0 ≤ z<sub>s</sub> ≤ 127  
 0 ≤ z<sub>d</sub> ≤ 127  
 動作: ((FSR2) + z<sub>s</sub>) → ((FSR2) + z<sub>d</sub>)  
 影響を受けるステータス: なし  
 エンコード:

1ワード目 (送信元)	1110	1011	1zzz	zzzz <sub>s</sub>
2ワード目 (格納先)	1111	xxxx	xzzz	zzzz <sub>d</sub>

説明: 送信元レジスタの内容を格納先レジスタに移動する。送信元レジスタと格納先レジスタのアドレスは、それぞれ7ビットのリテラルオフセット「z<sub>s</sub>」または「z<sub>d</sub>」をFSR2の値に加算して求める。どちらのレジスタも、4096バイトのデータメモリ空間(000h ~ FFFh)の任意の位置を指定できる。

MOVSS 命令では、格納先レジスタに PCL、TOSU、TOSH、TOSL を使えない。

上記で求めた送信元アドレスが間接アドレッシングレジスタを指し示している場合、戻り値は 00h となる。上記で求めた格納先アドレスが間接アドレッシングレジスタを指し示している場合、この命令は NOP として実行される。

ワード数: 2  
 サイクル数: 2

Q サイクルの動作:

	Q1	Q2	Q3	Q4
デコード	送信元アドレスを求める	送信元アドレスを求める	送信元レジスタを読み出す	
デコード	宛先アドレスを求める	宛先アドレスを求める	宛先レジスタへ書き込む	

例: MOVSS [05h], [06h]

命令実行前  
 FSR2 = 80h  
 Contents of 85h = 33h  
 Contents of 86h = 11h  
 命令実行後  
 FSR2 = 80h  
 Contents of 85h = 33h  
 Contents of 86h = 33h

## PUSHL Store Literal at FSR2, Decrement FSR2

構文: PUSHL k  
 オペランド: 0 ≤ k ≤ 255  
 動作: k → (FSR2),  
 FSR2 - 1 → FSR2  
 影響を受けるステータス: なし  
 エンコード:

1110	1010	kkkk	kkkk
------	------	------	------

説明: 8ビットのリテラル「k」をFSR2で指定したデータメモリアドレスに書き込む。この命令を実行後、FSR2の値をデクリメントする。

この命令を使って、ユーザはソフトウェアスタックに値をプッシュできる。

ワード数: 1  
 サイクル数: 1  
 Q サイクルの動作:

	Q1	Q2	Q3	Q4
デコード		「k」を読み出す	データを処理	格納先へ書き込み

例: PUSHL 08h

命令実行前  
 FSR2H:FSR2L = 01ECh  
 Memory (01ECh) = 00h  
 命令実行後  
 FSR2H:FSR2L = 01EBh  
 Memory (01ECh) = 08h

# PIC18F97J60 ファミリ

## SUBFSR Subtract Literal from FSR

構文: SUBFSR f, k

オペランド:  $0 \leq k \leq 63$   
 $f \in [0, 1, 2]$

動作:  $FSRf - k \rightarrow FSRf$

影響を受ける ステータス: なし

エンコード:

1110	1001	ffkk	kkkk
------	------	------	------

説明: 「f」で指定した FSR の内容から 6 ビットのリテラル「k」を減算する。

ワード数: 1

サイクル数: 1

Q サイクルの動作:

Q1	Q2	Q3	Q4
デコード	レジスタ「f」を読み出し	データを処理	格納先に書き込み

例: SUBFSR 2, 23h

命令実行前  
 FSR2 = 03FFh

命令実行後  
 FSR2 = 03DCh

## SUBULNK Subtract Literal from FSR2 and Return

構文: SUBULNK k

オペランド:  $0 \leq k \leq 63$

動作:  $FSR2 - k \rightarrow FSR2$ ,  
 $(TOS) \rightarrow PC$

影響を受ける ステータス: なし

エンコード:

1110	1001	11kk	kkkk
------	------	------	------

説明: FSR2 の内容から 6 ビットのリテラル「k」を減算する。次に、PC に TOS を読み込んで RETURN を実行する。

この命令の実行は 2 サイクルで、2 サイクル目には NOP を実行する。

これは、SUBFSR 命令で  $f = 3$  (2 進数「11」) である特殊な場合と考える事ができる。この命令は FSR2 に対してのみ実行できる。

ワード数: 1

サイクル数: 2

Q サイクルの動作:

Q1	Q2	Q3	Q4
デコード	レジスタ「f」を読み出し	データを処理	格納先に書き込み
NOP	NOP	NOP	NOP

例: SUBULNK 23h

命令実行前  
 FSR2 = 03FFh  
 PC = 0100h

命令実行後  
 FSR2 = 03DCh  
 PC = (TOS)

# PIC18F97J60 ファミリ

## 26.2.3 インデックス付きリテラル オフセットモードでのバイト/ビット指向命令

**Note:** PIC18 拡張命令セットを有効にすると、レガシー アプリケーションの挙動が不安定になったり、全く動作しなくなる場合があります。

拡張命令セットを有効にすると、8 つの新命令だけでなく、インデックス付きリテラル オフセット アドレッシングも有効になります ( [セクション 6.6.1「リテラル オフセットを使う インデックス付きアドレッシング」](#) 参照 )。このモードでは、PIC18 標準命令セットの多くのコマンドの解釈が大きく変化します。

拡張命令セットが無効の場合、オペコードに埋め込んだアドレスはリテラル メモリアドレスとして、すなわちアクセスバンクのアドレス (a=0 の場合) または BSR で指定した GPR バンクのアドレス (a=1 の場合) として扱われます。しかし拡張命令セットを有効にして、a=0 にすると、5Fh 以下のファイルレジスタ引数はリテラルアドレスではなく FSR2 のポインタ値からのオフセットと解釈されます。つまりこれは、アクセス RAM ビットを引数として使う全ての命令 (コア PIC18 命令の約半分に相当する全てのバイト/ビット指向命令) の挙動が、拡張命令セットを有効にすると変わる事を意味します。

FSR2 の内容が 00h であれば、アクセス RAM の境界は実質的に元の値に再マッピングされます。この性質を利用すれば、下位互換コードを容易に作成できます。ただしこの方法を使った場合、C 言語とアセンブリのルーチン間を移動する際に FSR2 の値を保存、復元してスタックポインタを正しく維持する必要があります。また、拡張命令セットの構文要件についても考慮しておく必要があります ( [セクション 26.2.3.1「拡張命令有効時の PIC18 標準コマンドの構文」](#) 参照 )。

インデックス付きリテラル オフセットモードはスタックとポインタの動的操作に非常に便利ですが、簡単な算術命令を誤ったレジスタに対して実行する可能性もあるため、注意が必要です。特に PIC18 プログラミングの経験が長い方は、拡張命令セットを有効にすると 5Fh 以下のレジスタアドレスがインデックス付きリテラル オフセット アドレッシングに使われるという点に十分注意してください。

この後のページでは、インデックス付きリテラル オフセットモードでバイト/ビット指向命令の実行結果がどのように変化するか、代表的な例を紹介します。これらの例で示すオペランドの条件は、この種の命令全てにあてはまります。

## 26.2.3.1 拡張命令有効時の PIC18 標準コマンドの構文

拡張命令セットを有効にした場合、標準命令セットのバイト/ビット指向命令におけるファイルレジスタ引数「f」は、リテラル オフセット値「k」で置き換えられます。ただし前述の通り、これは「f」が 5Fh 以下の場合に限ります。オフセット値を使う場合、角カッコ (「[]」) で囲んで示す必要があります。拡張命令の場合と同様、コンパイラは角カッコで囲んだ値をインデックスまたはオフセットとして解釈します。角カッコを省略した場合、または角カッコで囲んだ値が 5Fh より大きい場合、MPASM アセンブラでエラーが発生します。

インデックス引数をインデックス付きリテラル オフセット アドレッシング用に正しく角カッコで囲んでいれば、アクセス RAM 引数が指定される事はありません。自動的に「0」と見なされます。これは、拡張命令セットを無効にした標準の動作 (ターゲットアドレスに基づいて「a」がセットされる) とは対照的です。このモードでアクセス RAM ビットを宣言した場合も、MPASM アセンブラでエラーが発生します。

格納先の引数「d」は、以前と同様の働きをします。

最新バージョンの MPASM アセンブラでは、拡張命令セットの言語サポートを明示的に起動する必要があります。これを行うには、コマンドライン オプションの /y を使うか、ソースリストで PE ディレクティブを使います。

## 26.2.4 拡張命令セットを有効にする際の注意事項

命令セットの拡張は必ずしも全てのユーザにメリットがあるわけではないので注意が必要です。特に、ソフトウェア スタックを使うコードを作成するのでなければ、拡張命令セットを使うメリットはあまりありません。

また、インデックス付きリテラル オフセット アドレッシング モードは、PIC18 アセンブラ用に作成したレガシー アプリケーションで問題を起こすことがあります。つまり、レガシーコードの中にアクセスバンクの 5Fh 以下のレジスタをアドレッシングする命令が含まれていると問題が生じます。拡張命令セットを有効にしていると、これらのアドレスは FSR2 からのリテラル オフセットと解釈されるため、アプリケーションが誤ったデータアドレスに対して読み書きを実行する恐れがあります。

アプリケーションを PIC18F97J60 ファミリに移植する際は、コードの性格を見極める事が重要です。C 言語で作成した大規模な再入可能アプリケーションで、コンパイラ最適化が効果を発揮するようなアプリケーションは、拡張命令セットの使用に適しています。一方、アクセスバンクを多用するレガシー アプリケーションでは、ほとんどの場合、拡張命令セットを使うメリットはありません。

# PIC18F97J60 ファミリ

## ADDWF ADD W to Indexed (Indexed Literal Offset mode)

構文: ADDWF [k] {,d}

オペランド:  $0 \leq k \leq 95$   
 $d \in [0,1]$

動作:  $(W) + ((FSR2) + k) \rightarrow dest$

影響を受けるステータス: N, OV, C, DC, Z

エンコード:

0010	01d0	kkkk	kkkk
------	------	------	------

説明: Wレジスタの内容と、FSR2 + オフセット値「k」が示すレジスタの内容を加算する。  
「d」=「0」の場合、結果をWレジスタに格納する。「d」=「1」の場合、結果をレジスタ「f」に書き戻す(既定値)。

ワード数: 1

サイクル数: 1

Qサイクルの動作:

Q1	Q2	Q3	Q4
デコード	「k」を読み出す	データを処理	格納先に書き込み

例: ADDWF [OFST], 0

命令実行前

W	=	17h
OFST	=	2Ch
FSR2	=	0A00h
Contents of 0A2Ch	=	20h

命令実行後

W	=	37h
Contents of 0A2Ch	=	20h

## BSF Bit Set Indexed (Indexed Literal Offset mode)

構文: BSF [k], b

オペランド:  $0 \leq f \leq 95$   
 $0 \leq b \leq 7$

動作:  $1 \rightarrow ((FSR2) + k) <b>$

影響を受けるステータス: なし

エンコード:

1000	bbb0	kkkk	kkkk
------	------	------	------

説明: FSR2 + オフセット値「k」が示すレジスタのビット「b」をセットする。

ワード数: 1

サイクル数: 1

Qサイクルの動作:

Q1	Q2	Q3	Q4
デコード	レジスタ「f」を読み出し	データを処理	格納先に書き込み

例: BSF [FLAG\_OFST], 7

命令実行前

FLAG_OFST	=	0Ah
FSR2	=	0A00h
Contents of 0A0Ah	=	55h

命令実行後

Contents of 0A0Ah	=	D5h
-------------------	---	-----

## SETF Set Indexed (Indexed Literal Offset mode)

構文: SETF [k]

オペランド:  $0 \leq k \leq 95$

動作:  $FFh \rightarrow ((FSR2) + k)$

影響を受けるステータス: なし

エンコード:

0110	1000	kkkk	kkkk
------	------	------	------

説明: FSR2 + オフセット値「k」が示すレジスタの内容をFFhにセットする。

ワード数: 1

サイクル数: 1

Qサイクルの動作:

Q1	Q2	Q3	Q4
デコード	「k」を読み出す	データを処理	レジスタに書き込み

例: SETF [OFST]

命令実行前

OFST	=	2Ch
FSR2	=	0A00h
Contents of 0A2Ch	=	00h

命令実行後

Contents of 0A2Ch	=	FFh
-------------------	---	-----

# PIC18F97J60 ファミリ

---

## 26.2.5 マイクロチップ社の MPLAB® IDE ツールに関する注意事項

マイクロチップ社のソフトウェア ツールの最新バージョンは、PIC18F97J60 ファミリの拡張命令セットを完全にサポートできるように設計されています。これには、MPLAB C18 C コンパイラ、MPASM アセンブリ言語、MPLAB 統合開発環境 (IDE) が含まれます。

ソフトウェア開発のターゲット デバイスを選択すると、MPLAB IDE がそのデバイスのコンフィグレーション ビットを自動的に既定値に設定します。XINST コンフィグレーション ビットの既定値は「0」であり、拡張命令セットとインデックス付きリテラル オフセット アドレッシングは無効になります。拡張命令セットを利用して開発したアプリケーションを正しく実行するには、デバイスのプログラミング時に XINST ビットをセットする必要があります。

拡張命令セット対応のソフトウェアを開発する際は、使用言語ツールで拡張命令とインデックス付きアドレッシング モードのサポートを有効にする必要があります。有効にする方法は、使う環境に応じて何通りかあります。

- ツール環境内のメニュー オプションまたはダイアログ ボックスで、言語ツールとプロジェクトの設定を指定
- コマンドライン オプション
- ソースコードでディレクティブを使用

どの方法を選択するかは、使うコンパイラ、アセンブラ、開発環境によって異なります。詳細は、開発システムに付属する文書を参照してください。



## 27.0 開発サポート

PIC® マイクロコントローラと dsPIC® デジタルシグナルコントローラは、以下に示す各種ソフトウェア/ハードウェア開発ツールによって幅広くサポートされています。

- 統合開発環境
  - MPLAB® IDE ソフトウェア
- コンパイラ/アセンブラ/リンカ
  - 各種デバイスファミリ用 MPLAB C コンパイラ
  - 各種デバイスファミリ用 HI-TECH C
  - MPASM™ アセンブラ
  - MPLINK™ オブジェクトリンカ / MPLIB™ オブジェクトライブラリアン
  - 各種デバイスファミリ用 MPLAB アセンブラ/リンカ/ライブラリアン
- シミュレータ
  - MPLAB SIM ソフトウェア シミュレータ
- エミュレータ
  - MPLAB REAL ICE™ インサーキット エミュレータ
- インサーキット デバッグ
  - MPLAB ICD 3
  - PICkit™ 3 Debug Express
- デバイス プログラマ
  - PICkit™ 2 プログラマ
  - MPLAB PM3 デバイス プログラマ
- 低コストのデモボード、開発ボード、評価キット、スタータキット

## 27.1 MPLAB 統合開発環境ソフトウェア

MPLAB IDE ソフトウェアを使うと、従来の 8/16/32 ビット マイクロコントローラ市場では考えられないほど、ソフトウェアを容易に開発できます。MPLAB IDE は Windows® オペレーティングシステム上で動作するアプリケーションで、以下の機能を備えています。

- 全てのデバッグツールで共通のグラフィカルインターフェイス
  - シミュレータ
  - プログラマ (別売り)
  - インサーキット エミュレータ (別売り)
  - インサーキット デバッグ (別売り)
- コンテキスト色分け表示対応のフル機能エディタ
- マルチプロジェクト マネージャ
- 値を直接編集できるカスタマイズ可能なデータウィンドウ
- 高度なソースコード デバッグ
- マウスオーバーで変数の現在値を表示
- ソースウィンドウからウォッチ ウィンドウへの変数のドラッグ & ドロップ
- 充実したオンラインヘルプ
- IAR C コンパイラ等、一部他社製ツールの統合もサポート

MPLAB IDE を使うと、以下の作業が可能です。

- ソースファイル (C またはアセンブリ) の編集
- ワンタッチでのコンパイル/アセンブルと、エミュレータ/シミュレータ ツールへのダウンロード (全てのプロジェクト情報を自動更新)
- 以下を使ったデバッグ:
  - ソースファイル (C またはアセンブリ)
  - C とアセンブリの混在使用
  - マシンコード

MPLAB IDE は、対費用効果の高いシミュレータから低価格のインサーキット デバッグ、フル機能のエミュレータに至る複数のデバッグツールを 1 つの開発パラダイムでサポートします。このため、より柔軟で強力なツールにアップグレードした場合でも短期間で使用方法を習得できます。

# PIC18F97J60 ファミリ

## 27.2 各種デバイスファミリ用 MPLAB C コンパイラ

MPLAB C コンパイラは、マイクロチップ社の PIC18、PIC24、PIC32 マイクロコントローラ ファミリと、dsPIC30、dsPIC33 デジタル シグナル コントローラ ファミリ用コード開発に対応した ANSI C コンパイラです。これらのコンパイラは強力な統合機能と優れたコード最適化機能を備え、容易に使えます。

また、MPLAB IDE デバッガ用に最適化されたシンボル情報を出力できるため、ソースレベルのデバッグも容易です。

## 27.3 各種デバイスファミリ用 HI-TECH C

HI-TECH C コンパイラコード開発システムは、マイクロチップ社の PIC マイクロコントローラ ファミリと dsPIC デジタルシグナル コントローラ ファミリ向けの、総合的な機能を備えた ANSI C コンパイラです。これらのコンパイラは、強力な統合機能とインテリジェントなコード生成機能を備え、容易に使えます。

また、MPLAB IDE デバッガ用に最適化されたシンボル情報を出力できるため、ソースレベルのデバッグも容易です。

このコンパイラは、マクロアセンブラ、リンカ、プリプロセッサ、ワンステップ ドライバを備え、複数のプラットフォーム上で動作します。

## 27.4 MPASM アセンブラ

MPASM アセンブラは、PIC10/12/16/18 MCU に対応したフル機能の汎用マクロアセンブラです。

MPASM アセンブラは、MPLINK オブジェクト リンカ用の再配置可能なオブジェクト ファイル、Intel® 標準 HEX ファイル、メモリ使用状況とシンボル参照を詳述する MAP ファイル、ソース行と生成後のマシンコードを含む絶対 LST ファイル、デバッグ用の COFF ファイルを生成します。

MPASM アセンブラには以下のような機能があります。

- MPLAB IDE プロジェクトへの統合
- ユーザ定義マクロによるアセンブリコードの最適化
- 多用途ソースファイルに対応する条件付きアセンブリ
- アセンブリ プロセスを完全に制御できるディレクティブ

## 27.5 MPLINK オブジェクト リンカ / MPLIB オブジェクト ライブラリアン

MPLINK オブジェクト リンカは、MPASM アセンブラと MPLAB C18 C コンパイラが作成した再配置可能なオブジェクトを結合します。このオブジェクト リンカは、リンカスクリプトのディレクティブを使って、プリコンパイル済みのライブラリ内の再配置可能オブジェクトをリンクできます。

MPLIB オブジェクト ライブラリアンは、プリコンパイル済みコードのライブラリ ファイルの作成と変更を管理します。ライブラリのルーチンをソースファイルから呼び出すと、そのルーチンが含まれているモジュールのみがアプリケーションとリンクされます。これにより、大きなライブラリを各種アプリケーションで効率的に使えます。

オブジェクト リンカ / ライブラリには以下のような特長があります。

- 多数の小さなファイルをリンクするのではなく、1つのライブラリを効率的にリンクする
- 関連するモジュールをグループ化する事により、コードの保守性が向上する
- モジュールのリスト作成、置換、削除、抽出が容易なライブラリを柔軟に作成する

## 27.6 各種デバイスファミリ用 MPLAB アセンブラ、リンカ、ライブラリアン

MPLAB アセンブラは、PIC24、PIC32、dsPIC 用のシンボリック アセンブリ言語から、再配置可能なマシンコードを生成します。MPLAB C コンパイラはこのアセンブラを使ってオブジェクト ファイルを生成します。このアセンブラが生成した再配置可能なオブジェクト ファイルをアーカイブまたは他の再配置可能なオブジェクト ファイルとリンクして、実行ファイルを生成します。このアセンブラには以下のような特長があります。

- デバイスの全命令セットのサポート
- 固定 / 浮動小数点データのサポート
- コマンドライン インターフェイス
- 豊富なディレクティブセット
- 柔軟なマクロ言語
- MPLAB IDE との互換性

## 27.7 MPLAB SIM ソフトウェア シミュレータ

MPLAB SIM ソフトウェア シミュレータを使うと、PIC MCU と dsPIC<sup>®</sup> DSC を命令レベルでシミュレートする事によって、PC 環境でコード開発ができます。任意の命令に対してデータ領域を検証または変更でき、総合的なステミュラス コントローラから外部信号を印加できます。レジスタをファイルに記録して、より詳細な実行時解析が可能です。トレースバッファとロジック アナライザ ディスプレイを使うと、プログラム実行、I/O アクティビティ、ほとんどの周辺機能と内部レジスタを記録 / 追跡でき、シミュレータの機能をさらに強化できます。

MPLAB SIM ソフトウェア シミュレータは、MPLAB C コンパイラ、MPASM/MPLAB アセンブラを使ったシンボリック デバッグを完全にサポートしています。このソフトウェア シミュレータは、ハードウェアラボから離れた環境下でのコード開発 / デバッグに柔軟性を提供する、経済的で優れたソフトウェア開発ツールです。

## 27.8 MPLAB REAL ICE インサーキット エミュレータ システム

MPLAB REAL ICE インサーキット エミュレータ システムは、マイクロチップ社のフラッシュ DSC と MCU 用にマイクロチップ社が提供する次世代高速エミュレータです。このエミュレータでは、各キットに付属する MPLAB 統合開発環境 (IDE) の強力で使いやすい GUI を利用して、PIC<sup>®</sup> フラッシュ MCU と dsPIC<sup>®</sup> フラッシュ DSC をデバッグ / プログラムできます。

このエミュレータをハイスピード USB 2.0 インターフェイスで設計エンジニアの PC に接続し、ターゲット デバイスとはインサーキット デバッグシステムと互換の RJ-11 コネクタか、高速で耐ノイズ性に優れた最新の LVDS インターフェイス (CAT5) によって接続します。

エミュレータの更新用ファームウェアは、MPLAB IDE からダウンロードできます。MPLAB IDE の最新リリースに伴って、サポートするデバイスと新機能が追加されます。MPLAB REAL ICE は、低コスト、高速エミュレーション、実行時変数ウォッチ、トレース解析、複雑なブレークポイント、高耐久性のプロープ インターフェイス、接続ケーブルの長尺対応 (最長 3 m) 等、競合他社のエミュレータに比べて多くの利点があります。

## 27.9 MPLAB ICD 3 インサーキット デバッグ システム

MPLAB ICD 3 インサーキット デバッグシステムは、マイクロチップ社のフラッシュ デジタルシグナル コントローラ (DSC) とマイクロコントローラ (MCU) に対応した、最も対費用効果の高い高速ハードウェア デバッグ / プログラムです。このデバッグでは、MPLAB 統合開発環境 (IDE) の強力で使いやすい GUI を利用して PIC<sup>®</sup> フラッシュ マイクロコントローラと dsPIC<sup>®</sup> DSC をデバッグ / プログラムできます。

MPLAB ICD 3 インサーキット デバッグのプロープは、ハイスピード USB 2.0 インターフェイスで設計エンジニアの PC に接続し、ターゲット デバイスとは MPLAB ICD 2 または MPLAB REAL ICE システムと互換のコネクタ (RJ-11) によって接続します。MPLAB ICD 3 は全ての MPLAB ICD 2 ヘッダをサポートしています。

## 27.10 PICkit 3 インサーキット デバッグ / プログラムと PICkit 3 Debug Express

MPLAB PICkit 3 は、MPLAB 統合開発環境 (IDE) の強力な GUI を使って PIC<sup>®</sup> および dsPIC<sup>®</sup> フラッシュ マイクロコントローラをデバッグ / プログラムできる低価格なツールです。MPLAB PICkit 3 と設計エンジニアの PC の接続にはフルスピード USB インターフェイスを使います。ターゲット デバイスとの接続には、MPLAB ICD 3/MPLAB REAL ICE と互換のマイクロチップ デバッグ コネクタ (RJ-11) を使います。このコネクタは 2 本のデバイス I/O ピンとリセットラインを使って、インサーキット デバッグとインサーキット シリアル プログラミング (In-Circuit Serial Programming™) を実現します。

PICkit 3 Debug Express は、PICkit 3、デモボードとマイクロコントローラ、フックアップ ケーブル、CD-ROM (ユーザガイド、レッスン、チュートリアル、コンパイル、MPLAB IDE ソフトウェアを収録) を含みます。

# PIC18F97J60 ファミリ

## 27.11 PICKit 2 開発用プログラマ / デバッグと PICKit 2 Debug Express

PICKit™ 2 開発用プログラマ / デバッグは、マイクロチップ社のフラッシュ マイクロコントローラ ファミリの書き込みとデバッグを使いやすいインターフェイスで実現する、低コストの開発ツールです。Windows® 環境でプログラミング機能を実行できる本製品は、ベースライン (PIC10F、PIC12F5xx、PIC16F5xx)、ミッドレンジ (PIC12F6xx、PIC16F)、PIC18F、PIC24、dsPIC30、dsPIC33、PIC32 ファミリを含む 8/16/32 ビット マイクロコントローラ、マイクロチップ社製各種シリアル EEPROM をサポートしています。PICKit™ 2 は、マイクロチップ社の強力な MPLAB 統合開発環境 (IDE) を利用して、ほとんどの PIC® マイクロコントローラに対してインサーキット デバッグを実行できます。インサーキット デバッグでは、PIC マイクロコントローラをアプリケーションに組み込んだままの状態でのプログラムの実行 / 停止とシングルステップ実行が可能です。ブレークポイントで停止させてファイルレジスタを確認 / 変更する事もできます。

PICKit 2 Debug Express は、PICKit 2、デモボードとマイクロコントローラ、フックアップケーブル、CD-ROM (ユーザガイド、レッスン、チュートリアル、コンパイラ、MPLAB IDE ソフトウェアを収録) を含みます。

## 27.12 MPLAB PM3 デバイス プログラマ

MPLAB PM3 デバイス プログラマは CE 準拠のユニバーサル デバイス プログラマで、VDDMIN と VDDMAX でのプログラマブル電圧検証によって最大限の信頼性を確保します。このデバイス プログラマはメニューとエラーメッセージ表示用の大型 LCD ディスプレイ (128 x 64) を装備している他、着脱式のモジュール式ソケット アセンブリによって各種パッケージタイプに対応します。ICSP™ ケーブル アセンブリは標準で付属しています。スタンドアロン モードでは、MPLAB PM3 デバイス プログラマを PC に接続せずに、PIC の読み出し、検証、書き込みが可能です。このモードでコード保護も設定できます。MPLAB PM3 とホスト PC との接続には、RS-232 または USB ケーブルを使います。さらに、大容量メモリデバイスの高速書き込みを可能にする高速通信と最適化されたアルゴリズムを備え、ファイル保存とデータ アプリケーションのための MMC カードを内蔵しています。

## 27.13 デモボード、開発ボード、評価キット、スタータキット

各種 PIC MCU と dsPIC DSC に対応するデモボード、開発ボード、評価用ボードを豊富に取り揃え、完全に機能するシステム上でアプリケーションを迅速に開発できます。ほとんどのボードは、カスタム回路を追加するためのプロトタイプ領域を備えています。また、検討と変更が可能なようにアプリケーション ファームウェアとソースコードが付属しています。

これらのボードは LED、温度センサ、スイッチ、スピーカ、RS-232 インターフェイス、LCD、ポテンショメータ、増設 EEPROM メモリをはじめとする各種機能をサポートします。

デモボードと開発ボードは、カスタム回路の試作と各種マイクロコントローラ アプリケーションの学習教材として使う事ができます。

PICDEM™ と dsPICDEM™ デモ / 開発ボードシリーズの回路の他に、マイクロチップ社ではアナログフィルタ設計、KEELOQ® セキュリティ IC、CAN、IrDA®、PowerSmart バッテリ管理、SEEVAL® 評価システム、 $\Delta \Sigma$  ADC、流量センシング、その他多数のアプリケーションに対応する評価キットとデモソフトウェアを取り揃えています。

また、特定のデバイスを体験するために必要なものを全てを収めたスタータキットも提供しています。スタータキットは通常、1 つのアプリケーションとデバッグ機能の全てを 1 つのボードに搭載した形で提供します。

デモボード、開発ボード、評価キットの全一覧は、マイクロチップ社のウェブページ ([www.microchip.com](http://www.microchip.com)) をご覧ください。

## 28.0 電気的特性

### 絶対最大定格 (†)

通電中の周囲温度 .....	-40 ~ +100 °C
保管温度 .....	-65 ~ +150 °C
VSS に対する全てのデジタル専用入力ピンまたは MCLR の電圧 (VDD を除く) .....	-0.3 ~ 6.0 V
VSS に対するデジタル / アナログ兼用ピンの電圧 .....	-0.3 V ~ (VDD + 0.3 V)
VSS に対する VDDCORE の電圧 .....	-0.3 ~ 2.75 V
VSS に対する VDD の電圧 .....	-0.3 ~ 4.0 V
総消費電力 (Note 1) .....	1.0 W
VSS ピンからの最大電流 .....	300 mA
VDD ピンへの最大電流 .....	250 mA
入力クランプ電流、I <sub>IK</sub> (V <sub>I</sub> < 0 または V <sub>I</sub> > VDD) (Note 2) .....	±0 mA
出力クランプ電流、I <sub>OK</sub> (V <sub>O</sub> < 0 または V <sub>O</sub> > VDD) (Note 2) .....	±0 mA
PORTB、PORTC I/O ピンの最大出力シンク電流 .....	25 mA
PORTD、PORTE、PORTJ I/O ピンの最大出力シンク電流 .....	8 mA
PORTA、PORTF、PORTG、PORTH I/O ピンの最大出力シンク電流 (Note 3) .....	2 mA
PORTB、PORTC I/O ピンの最大出力ソース電流 .....	25 mA
PORTD、PORTE、PORTJ I/O ピンの最大出力ソース電流 .....	8 mA
PORTA、PORTF、PORTG、PORTH I/O ピンの最大出力ソース電流 (Note 3) .....	2 mA
全ポートの合計最大シンク電流 .....	200 mA
全ポートの合計最大ソース電流 .....	200 mA

**Note 1:** 消費電力は以下の式によって計算できます。

$$P_{dis} = VDD \times \{I_{DD} - \sum I_{OH}\} + \sum \{(VDD - V_{OH}) \times I_{OH}\} + \sum (V_{OL} \times I_{OL}) + \sum (V_{TPOUT} \times I_{TPOUT})$$

**2:** クランプ ダイオードは接続していません。

**3:** RA<1> と RA<0> を除きます。これらのピンは最大 25 mA で LED を直接駆動できます。

† **NOTICE:** 上記の「絶対最大定格」を超えるストレスを与えると、デバイスに恒久的な損傷を与える可能性があります。絶対最大定格は定格ストレスのみを示すものであり、上記の状態または本仕様書の動作条件に示されている規定値を超える状態でデバイスが正常に機能する事を示すものではありません。長期間にわたる最大定格条件での動作は、デバイスの信頼性に影響する可能性があります。

# PIC18F97J60 ファミリ

図 28-1: PIC18F97J60 ファミリの周波数に対する電圧のグラフ、レギュレータ有効時 (ENVREG を VDD に接続)

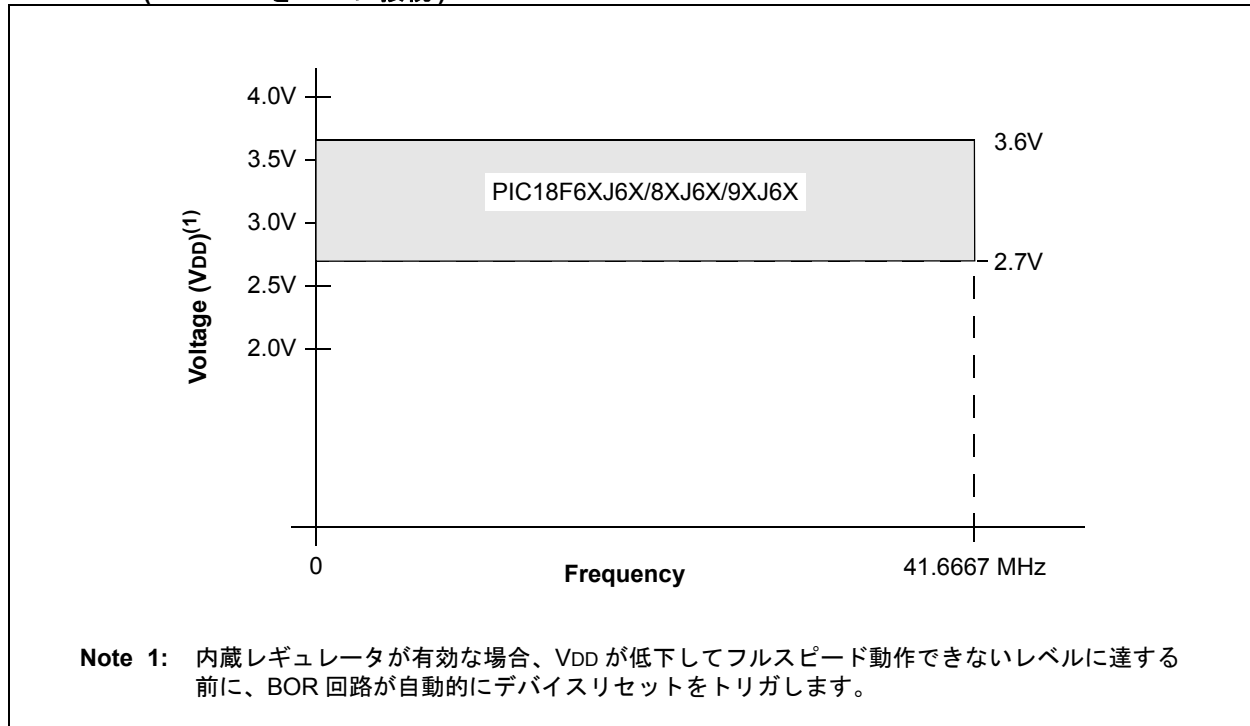
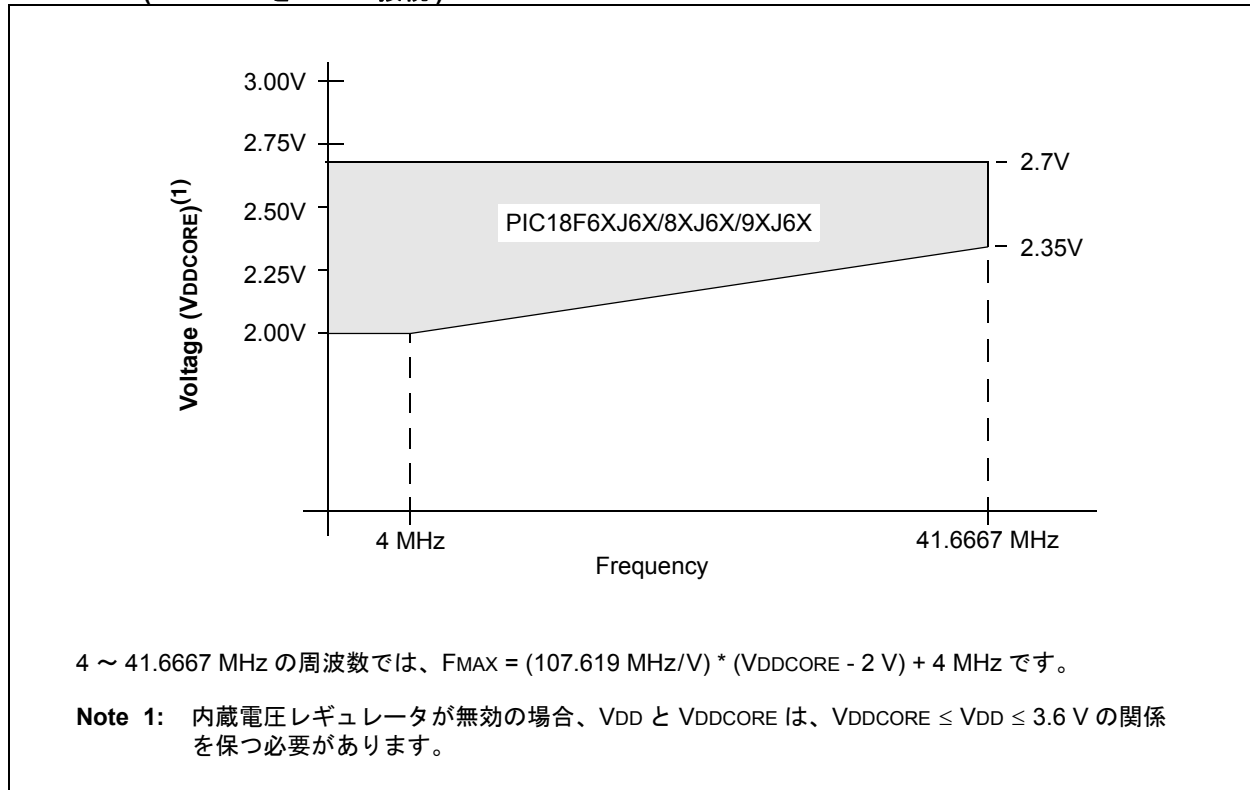


図 28-2: PIC18F97J60 ファミリの周波数に対する電圧のグラフ、レギュレータ無効時 (ENVREG を VSS に接続)



# PIC18F97J60 ファミリ

## 28.1 DC 特性 :

### 電源電圧

#### PIC18F97J60 ファミリ (産業用)

PIC18F97J60 ファミリ (産業用)			標準動作条件 (特に明記しない場合) 動作温度 $-40\text{ }^{\circ}\text{C} \leq T_A \leq +85\text{ }^{\circ}\text{C}$ (産業用温度レンジ)				
パラメータ 番号	記号	特性	最小値	代表値	最大値	単位	条件
D001	VDD	電源電圧	VDDCORE 2.7 3.1	— — —	3.6 3.6 3.6	V V V	ENVREG を VSS に接続 ENVREG を VDD に接続 Ethernet モジュールが有効 (ECON2<5> = 1)
D001B	VDDCORE	マイクロコントローラ コアの外部電源	2.0	—	2.7	V	
D001C	AVDD	アナログ電源電圧	VDD - 0.3	—	VDD + 0.3	V	
D002	VDR	RAM データ保持 電圧 <sup>(1)</sup>	1.5	—	—	V	
D003	VPOR	VDD パワーオンリセット 電圧	—	—	0.7	V	詳細は、 <a href="#">セクション 5.3「パワーオンリセット (POR)」</a> 参照。
D004	SVDD	内部パワーオンリセット を確実に動作させるための VDD 立ち上がり速度	0.05	—	—	V/ms	詳細は、 <a href="#">セクション 5.3「パワーオンリセット (POR)」</a> 参照。
D005	BOR	ブラウンアウトリセット	2.35	2.4	2.7	V	

**Note 1:** これは、スリープまたはデバイスのリセット中に VDD を下げて RAM データを失わない下限値です。

# PIC18F97J60 ファミリ

## 28.2 DC 特性 :                    パワーダウン電流と消費電流 PIC18F97J60 ファミリ (産業用)

PIC18F97J60 ファミリ (産業用)		標準動作条件 (特に明記しない場合) 動作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (産業用温度レンジ)			
パラメータ 番号	デバイス	代表値	最大値	単位	条件
<b>パワーダウン電流 (IPD)<sup>(1)</sup></b>					
	全デバイス	19.0	69.0	$\mu\text{A}$	$-40^{\circ}\text{C}$
		21.0	69.0	$\mu\text{A}$	$+25^{\circ}\text{C}$
		45.0	149.0	$\mu\text{A}$	$+85^{\circ}\text{C}$
	全デバイス	26.0	104.0	$\mu\text{A}$	$-40^{\circ}\text{C}$
		29.0	104.0	$\mu\text{A}$	$+25^{\circ}\text{C}$
		60.0	184.0	$\mu\text{A}$	$+85^{\circ}\text{C}$
	全デバイス	40.0	203.0	$\mu\text{A}$	$-40^{\circ}\text{C}$
		44.0	203.0	$\mu\text{A}$	$+25^{\circ}\text{C}$
		105.0	209.0	$\mu\text{A}$	$+85^{\circ}\text{C}$

- Note 1:** スリープ時のパワーダウン電流は、オシレータの種類に関わらず一定です。パワーダウン電流は、デバイスをスリープに移行させ、全ての I/O ピンをハイインピーダンス状態にして VDD または VSS に接続し、電流増 ( $\Delta$  電流) をもたらず機能 (WDT、Timer1、オシレータ等) を全て無効にして計測しています。
- 2:** 消費電流は主に動作電圧、周波数、モードによって決まります。他の要因として I/O ピンの負荷とスイッチングレート、オシレータのタイプと回路、内部コード実行パターン、温度等があり、これらも消費電流に影響を与えます。アクティブ動作モードにおける IDD の計測値は、全て以下の条件でテストしています。  
OSC1 = 外部矩形波 (レールツーレール)、I/O ピンは全て 3 ステートで VDD にプル、MCLR = VDD、WDT は指定の通り有効化または無効化しています。
- 3:** 標準的な低コストの 32 kHz 水晶振動子の動作温度レンジは  $-10 \sim +70^{\circ}\text{C}$  です。コストは増加しますが、拡張温度仕様に対応した水晶振動子も利用可能です。
- 4:** 電圧レギュレータを無効に (ENVREG = 0、VSS に接続) した場合は。
- 5:** 電圧レギュレータを有効に (ENVREG = 1、VDD に接続) した場合は。
- 6:**  $\Delta\text{IETH}$  を規定する電流には、TPOUT+ と TPOUT- のシンク電流が含まれます。LEDA と LEDB は全てのテストで無効です。



# PIC18F97J60 ファミリ

## 28.2 DC 特性 :

### パワーダウン電流と消費電流 PIC18F97J60 ファミリ (産業用) (続き)

PIC18F97J60 ファミリ (産業用)		標準動作条件 (特に明記しない場合) 動作温度 -40 °C ≤ TA ≤ +85 °C (産業用温度レンジ)					
パラメータ 番号	デバイス	代表値	最大値	単位	条件		
<b>消費電流 (IDD)(2,3)</b>							
全デバイス		12.0	34.0	μA	-40°C	VDD = 2.0V, VDDCORE = 2.0V <sup>(4)</sup>	FOSC = 31 kHz (RC_RUN モード、 内部オシレータ信号源)
		12.0	34.0	μA	+25°C		
		74.0	108.0	μA	+85°C		
全デバイス		20.0	45.0	μA	-40°C	VDD = 2.5V, VDDCORE = 2.5V <sup>(4)</sup>	
		20.0	45.0	μA	+25°C		
		82.0	126.0	μA	+85°C		
全デバイス		105.0	168.0	μA	-40°C	VDD = 3.3V <sup>(5)</sup>	
		105.0	168.0	μA	+25°C		
		182.0	246.0	μA	+85°C		
全デバイス		8.0	32.0	μA	-40°C	VDD = 2.0V, VDDCORE = 2.0V <sup>(4)</sup>	FOSC = 31 kHz (RC_IDLE モード、 内部オシレータ信号源)
		8.0	32.0	μA	+25°C		
		62.0	98.0	μA	+85°C		
全デバイス		12.0	35.0	μA	-40°C	VDD = 2.5V, VDDCORE = 2.5V <sup>(4)</sup>	
		12.0	35.0	μA	+25°C		
		70.0	95.0	μA	+85°C		
全デバイス		90.0	152.0	μA	-40°C	VDD = 3.3V <sup>(5)</sup>	
		90.0	152.0	μA	+25°C		
		170.0	225.0	μA	+85°C		

- Note 1:** スリープ時のパワーダウン電流は、オシレータの種類に関わらず一定です。パワーダウン電流は、デバイスをスリープに移行させ、全ての I/O ピンをハイインピーダンス状態にして VDD または VSS に接続し、電流増 (Δ 電流) をもたらず機能 (WDT、Timer1、オシレータ等) を全て無効にして計測しています。
- 2:** 消費電流は主に動作電圧、周波数、モードによって決まります。他の要因として I/O ピンの負荷とスイッチング レート、オシレータのタイプと回路、内部コード実行パターン、温度等があり、これらも消費電流に影響を与えます。アクティブ動作モードにおける IDD の計測値は、全て以下の条件でテストしています。  
OSC1 = 外部矩形波 (レールツーレール)、I/O ピンは全て 3 ステートで VDD にプル、  
MCLR = VDD、WDT は指定の通り有効化または無効化しています。
- 3:** 標準的な低コストの 32 kHz 水晶振動子の動作温度レンジは -10 ~ +70 °C です。コストは増加しますが、拡張温度仕様に対応した水晶振動子も利用可能です。
- 4:** 電圧レギュレータを無効に (ENVREG = 0、VSS に接続) した場合です。
- 5:** 電圧レギュレータを有効に (ENVREG = 1、VDD に接続) した場合です。
- 6:** ΔIETH を規定する電流には、TPOUT+ と TPOUT- のシンク電流が含まれます。LEDA と LEDB は全てのテストで無効です。

# PIC18F97J60 ファミリ

## 28.2 DC 特性 :                      パワーダウン電流と消費電流 PIC18F97J60 ファミリ (産業用) (続き)

PIC18F97J60 ファミリ (産業用)		標準動作条件 (特に明記しない場合) 動作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (産業用温度レンジ)						
パラメータ 番号	デバイス	代表値	最大値	単位	条件			
<b>消費電流 (IDD)<sup>(2)</sup></b>								
全デバイス		0.8	1.5	mA	-40°C	VDD = 2.0V, VDDCORE = 2.0V <sup>(4)</sup>	FOSC = 1 MHz (PRI_RUN モード、 EC オシレータ)	
		0.8	1.5	mA	+25°C			
		0.9	1.7	mA	+85°C			
全デバイス		1.1	1.8	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V <sup>(4)</sup>		
		1.1	1.8	mA	+25°C			
		1.2	2.0	mA	+85°C			
全デバイス		2.1	3.4	mA	-40°C	VDD = 3.3V <sup>(5)</sup>		
		2.0	3.4	mA	+25°C			
		2.1	3.4	mA	+85°C			
全デバイス		9.2	14.5	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V <sup>(4)</sup>	FOSC = 25 MHz (PRI_RUN モード、 EC オシレータ)	
		9.0	14.5	mA	+25°C			
		9.2	14.5	mA	+85°C			
全デバイス		13.0	18.4	mA	-40°C	VDD = 3.3V <sup>(5)</sup>		
		12.4	18.4	mA	+25°C			
		13.0	18.4	mA	+85°C			
全デバイス		13.4	19.8	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V <sup>(4)</sup>		FOSC = 41.6667 MHz (PRI_RUN モード、 EC オシレータ)
		13.0	19.8	mA	+25°C			
		13.4	19.8	mA	+85°C			
全デバイス		14.5	21.6	mA	-40°C	VDD = 3.3V <sup>(5)</sup>		
		14.4	21.6	mA	+25°C			
		14.5	21.6	mA	+85°C			

- Note 1:** スリープ時のパワーダウン電流は、オシレータの種類に関わらず一定です。パワーダウン電流は、デバイスをスリープに移行させ、全ての I/O ピンをハイインピーダンス状態にして VDD または VSS に接続し、電流増 (Δ 電流) をもたらず機能 (WDT、Timer1、オシレータ等) を全て無効にして計測しています。
- 2:** 消費電流は主に動作電圧、周波数、モードによって決まります。他の要因として I/O ピンの負荷とスイッチングレート、オシレータのタイプと回路、内部コード実行パターン、温度等があり、これらも消費電流に影響を与えます。アクティブ動作モードにおける IDD の計測値は、全て以下の条件でテストしています。  
 OSC1 = 外部矩形波 (レールツーレール)、I/O ピンは全て 3 ステートで VDD にプル、  
 MCLR = VDD、WDT は指定の通り有効化または無効化しています。
- 3:** 標準的な低コストの 32 kHz 水晶振動子の動作温度レンジは -10 ~ +70 °C です。コストは増加しますが、拡張温度仕様に対応した水晶振動子も利用可能です。
- 4:** 電圧レギュレータを無効に (ENVREG = 0、VSS に接続) した場合はです。
- 5:** 電圧レギュレータを有効に (ENVREG = 1、VDD に接続) した場合はです。
- 6:** ΔIETH を規定する電流には、TPOUT+ と TPOUT- のシンク電流が含まれます。LEDA と LEDB は全てのテストで無効です。

# PIC18F97J60 ファミリ

## 28.2 DC 特性 :

### パワーダウン電流と消費電流 PIC18F97J60 ファミリ (産業用) (続き)

PIC18F97J60 ファミリ (産業用)		標準動作条件 (特に明記しない場合) 動作温度 -40 °C ≤ TA ≤ +85 °C (産業用温度レンジ)						
パラメータ 番号	デバイス	代表値	最大値	単位	条件			
<b>消費電流 (IDD)(2)</b>								
全デバイス		2.8	5.2	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	Fosc = 25 MHz, 2.7778 MHz (内部) (PRI_RUN HS モード)	
		2.5	5.2	mA	+25°C			
		2.8	5.2	mA	+85°C			
全デバイス		3.6	6.4	mA	-40°C	VDD = 3.3V(5)		
		3.3	6.4	mA	+25°C			
		3.6	6.4	mA	+85°C			
全デバイス		6.4	11.0	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)		Fosc = 25 MHz, 13.8889 MHz (内部) (PRI_RUN HSPLL モード)
		6.0	11.0	mA	+25°C			
		6.4	11.0	mA	+85°C			
全デバイス		7.8	12.5	mA	-40°C	VDD = 3.3V(5)		
		7.4	12.5	mA	+25°C			
		7.8	12.5	mA	+85°C			
全デバイス		9.2	14.5	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	Fosc = 25 MHz, 25 MHz (内部) (PRI_RUN HS モード)	
		9.0	14.5	mA	+25°C			
		9.2	14.5	mA	+85°C			
全デバイス		13.0	18.4	mA	-40°C	VDD = 3.3V(5)		
		12.4	18.4	mA	+25°C			
		13.0	18.4	mA	+85°C			
全デバイス		13.4	19.8	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)		Fosc = 25 MHz, 41.6667 MHz (内部) (PRI_RUN HSPLL モード)
		13.0	19.8	mA	+25°C			
		13.4	19.8	mA	+85°C			
全デバイス		14.5	21.6	mA	-40°C	VDD = 3.3V(5)		
		14.4	21.6	mA	+25°C			
		14.5	21.6	mA	+85°C			

- Note 1:** スリープ時のパワーダウン電流は、オシレータの種類に関わらず一定です。パワーダウン電流は、デバイスをスリープに移行させ、全ての I/O ピンをハイインピーダンス状態にして VDD または VSS に接続し、電流増 (Δ 電流) をもたらず機能 (WDT、Timer1、オシレータ等) を全て無効にして計測しています。
- 2:** 消費電流は主に動作電圧、周波数、モードによって決まります。他の要因として I/O ピンの負荷とスイッチング レート、オシレータのタイプと回路、内部コード実行パターン、温度等があり、これらも消費電流に影響を与えます。アクティブ動作モードにおける IDD の計測値は、全て以下の条件でテストしています。  
OSC1 = 外部矩形波 (レールツーレール)、I/O ピンは全て 3 ステートで VDD にプル、MCLR = VDD、WDT は指定の通り有効化または無効化しています。
- 3:** 標準的な低コストの 32 kHz 水晶振動子の動作温度レンジは -10 ~ +70 °C です。コストは増加しますが、拡張温度仕様に対応した水晶振動子も利用可能です。
- 4:** 電圧レギュレータを無効に (ENVREG = 0、VSS に接続) した場合は。
- 5:** 電圧レギュレータを有効に (ENVREG = 1、VDD に接続) した場合は。
- 6:** ΔIETH を規定する電流には、TPOUT+ と TPOUT- のシンク電流が含まれます。LEDA と LEDB は全てのテストで無効です。

# PIC18F97J60 ファミリ

## 28.2 DC 特性 :                   パワーダウン電流と消費電流 PIC18F97J60 ファミリ (産業用) (続き)

PIC18F97J60 ファミリ (産業用)		標準動作条件 (特に明記しない場合) 動作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (産業用温度レンジ)					
パラメータ 番号	デバイス	代表値	最大値	単位	条件		
<b>消費電流 (IDD)<sup>(2)</sup></b>							
全デバイス		0.5	1.1	mA	-40°C	VDD = 2.0V, VDDCORE = 2.0V <sup>(4)</sup>	FOSC = 1 MHz (PRI_IDLE モード、 EC オシレータ)
		0.5	1.1	mA	+25°C		
		0.6	1.2	mA	+85°C		
全デバイス		0.9	1.4	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V <sup>(4)</sup>	
		0.9	1.4	mA	+25°C		
		1.0	1.5	mA	+85°C		
全デバイス		1.9	2.6	mA	-40°C	VDD = 3.3V <sup>(5)</sup>	
		1.8	2.6	mA	+25°C		
		1.9	2.6	mA	+85°C		
全デバイス		5.9	9.5	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V <sup>(4)</sup>	FOSC = 25 MHz (PRI_IDLE モード、 EC オシレータ)
		5.6	9.5	mA	+25°C		
		5.9	9.5	mA	+85°C		
全デバイス		7.5	13.2	mA	-40°C	VDD = 3.3V <sup>(5)</sup>	
		7.2	13.2	mA	+25°C		
		7.5	13.2	mA	+85°C		
全デバイス		8.6	14.0	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V <sup>(4)</sup>	FOSC = 41.6667 MHz (PRI_IDLE モード、 EC オシレータ)
		8.0	14.0	mA	+25°C		
		8.6	14.0	mA	+85°C		
全デバイス		9.8	16.0	mA	-40°C	VDD = 3.3V <sup>(5)</sup>	
		9.4	16.0	mA	+25°C		
		9.8	16.0	mA	+85°C		

- Note 1:** スリープ時のパワーダウン電流は、オシレータの種類に関わらず一定です。パワーダウン電流は、デバイスをスリープに移行させ、全ての I/O ピンをハイインピーダンス状態にして VDD または VSS に接続し、電流増 (Δ 電流) をもたらず機能 (WDT、Timer1、オシレータ等) を全て無効にして計測しています。
- 2:** 消費電流は主に動作電圧、周波数、モードによって決まります。他の要因として I/O ピンの負荷とスイッチング レート、オシレータのタイプと回路、内部コード実行パターン、温度等があり、これらも消費電流に影響を与えます。アクティブ動作モードにおける IDD の計測値は、全て以下の条件でテストしています。  
OSC1 = 外部矩形波 (レールツーレール)、I/O ピンは全て 3 ステートで VDD にプル、MCLR = VDD、WDT は指定の通り有効化または無効化しています。
- 3:** 標準的な低コストの 32 kHz 水晶振動子の動作温度レンジは -10 ~ +70 °C です。コストは増加しますが、拡張温度仕様に対応した水晶振動子も利用可能です。
- 4:** 電圧レギュレータを無効に (ENVREG = 0、VSS に接続) した場合です。
- 5:** 電圧レギュレータを有効に (ENVREG = 1、VDD に接続) した場合です。
- 6:** ΔIETH を規定する電流には、TPOUT+ と TPOUT- のシンク電流が含まれます。LEDA と LEDB は全てのテストで無効です。

# PIC18F97J60 ファミリ

## 28.2 DC 特性 :

### パワーダウン電流と消費電流 PIC18F97J60 ファミリ (産業用) (続き)

PIC18F97J60 ファミリ (産業用)		標準動作条件 (特に明記しない場合) 動作温度 -40 °C ≤ TA ≤ +85 °C (産業用温度レンジ)					
パラメータ 番号	デバイス	代表値	最大値	単位	条件		
<b>消費電流 (IDD)(2)</b>							
全デバイス		22.0	45.0	μA	-10°C	VDD = 2.0V, VDDCORE = 2.0V(4)	FOSC = 32 kHz(3) (SEC_RUN モード、 クロックは Timer1)
		22.0	45.0	μA	+25°C		
		78.0	114.0	μA	+70°C		
全デバイス		27.0	52.0	μA	-10°C	VDD = 2.5V, VDDCORE = 2.5V(4)	
		27.0	52.0	μA	+25°C		
		92.0	135.0	μA	+70°C		
全デバイス		106.0	168.0	μA	-10°C	VDD = 3.3V(5)	
		106.0	168.0	μA	+25°C		
		188.0	246.0	μA	+70°C		
全デバイス		18.0	37.0	μA	-10°C	VDD = 2.0V, VDDCORE = 2.0V(4)	FOSC = 32 kHz(3) (SEC_IDLE モード、 クロックは Timer1)
		18.0	37.0	μA	+25°C		
		75.0	105.0	μA	+70°C		
全デバイス		21.0	40.0	μA	-10°C	VDD = 2.5V, VDDCORE = 2.5V(4)	
		21.0	40.0	μA	+25°C		
		84.0	98.0	μA	+70°C		
全デバイス		94.0	152.0	μA	-10°C	VDD = 3.3V(5)	
		94.0	152.0	μA	+25°C		
		182.0	225.0	μA	+70°C		

- Note 1:** スリープ時のパワーダウン電流は、オシレータの種類に関わらず一定です。パワーダウン電流は、デバイスをスリープに移行させ、全ての I/O ピンをハイインピーダンス状態にして VDD または VSS に接続し、電流増 (Δ 電流) をもたらず機能 (WDT、Timer1、オシレータ等) を全て無効にして計測しています。
- 2:** 消費電流は主に動作電圧、周波数、モードによって決まります。他の要因として I/O ピンの負荷とスイッチング レート、オシレータのタイプと回路、内部コード実行パターン、温度等があり、これらも消費電流に影響を与えます。アクティブ動作モードにおける IDD の計測値は、全て以下の条件でテストしています。  
OSC1 = 外部矩形波 (レールツーレール)、I/O ピンは全て 3 ステートで VDD にプル、  
MCLR = VDD、WDT は指定の通り有効化または無効化しています。
- 3:** 標準的な低コストの 32 kHz 水晶振動子の動作温度レンジは -10 ~ +70 °C です。コストは増加しますが、拡張温度仕様に対応した水晶振動子も利用可能です。
- 4:** 電圧レギュレータを無効に (ENVREG = 0、VSS に接続) した場合です。
- 5:** 電圧レギュレータを有効に (ENVREG = 1、VDD に接続) した場合です。
- 6:** ΔIETH を規定する電流には、TPOUT+ と TPOUT- のシンク電流が含まれます。LEDA と LEDB は全てのテストで無効です。

# PIC18F97J60 ファミリ

## 28.2 DC 特性 : パワーダウン電流と消費電流 PIC18F97J60 ファミリ (産業用) (続き)

PIC18F97J60 ファミリ (産業用)		標準動作条件 (特に明記しない場合) 動作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (産業用温度レンジ)					
パラメータ 番号	デバイス	代表値	最大値	単位	条件		
D022 ( $\Delta I_{WDT}$ )	モジュールの電流増 ( $\Delta I_{WDT}$ , $\Delta I_{OSCB}$ , $\Delta I_{AD}$ , $\Delta I_{ETH}$ ) ウォッチドッグ タイマ	2.4	7.0	$\mu\text{A}$	$-40^{\circ}\text{C}$	$V_{DD} = 2.0\text{V}$ , $V_{DDCORE} = 2.0\text{V}^{(4)}$	
		2.4	7.0	$\mu\text{A}$	$+25^{\circ}\text{C}$		
		12.0	19.0	$\mu\text{A}$	$+85^{\circ}\text{C}$		
		3.0	8.0	$\mu\text{A}$	$-40^{\circ}\text{C}$	$V_{DD} = 2.5\text{V}$ , $V_{DDCORE} = 2.5\text{V}^{(4)}$	
		3.0	8.0	$\mu\text{A}$	$+25^{\circ}\text{C}$		
		14.0	22.0	$\mu\text{A}$	$+85^{\circ}\text{C}$		
		5.0	12.0	$\mu\text{A}$	$-40^{\circ}\text{C}$	$V_{DD} = 3.3\text{V}^{(5)}$	
		5.0	12.0	$\mu\text{A}$	$+25^{\circ}\text{C}$		
		19.0	30.0	$\mu\text{A}$	$+85^{\circ}\text{C}$		
		D025 ( $\Delta I_{OSCB}$ )	Timer1 オシレータ	12.0	20.0	$\mu\text{A}$	
12.0	20.0			$\mu\text{A}$	$+25^{\circ}\text{C}$		
24.0	36.0			$\mu\text{A}$	$+85^{\circ}\text{C}$		
13.0	21.0			$\mu\text{A}$	$-40^{\circ}\text{C}$	$V_{DD} = 2.5\text{V}$ , $V_{DDCORE} = 2.5\text{V}^{(4)}$	
13.0	21.0			$\mu\text{A}$	$+25^{\circ}\text{C}$		
26.0	38.0			$\mu\text{A}$	$+85^{\circ}\text{C}$		
14.0	25.0			$\mu\text{A}$	$-40^{\circ}\text{C}$	$V_{DD} = 3.3\text{V}^{(5)}$	
14.0	25.0			$\mu\text{A}$	$+25^{\circ}\text{C}$		
29.0	40.0	$\mu\text{A}$	$+85^{\circ}\text{C}$				
D026 ( $\Delta I_{AD}$ )	A/D コンバータ	1.2	10.0	$\mu\text{A}$	$-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$	$V_{DD} = 2.0\text{V}$ , $V_{DDCORE} = 2.0\text{V}^{(4)}$	A/D を ON、変換は実行していない状態
		1.2	10.0	$\mu\text{A}$	$-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$	$V_{DD} = 2.5\text{V}$ , $V_{DDCORE} = 2.5\text{V}^{(4)}$	
		1.2	11.0	$\mu\text{A}$	$-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$	$V_{DD} = 3.3\text{V}^{(5)}$	
D027 $\Delta I_{ETH}^{(6)}$	Ethernet モジュール	130.0	156.0	$\text{mA}$	$-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$	$V_{DD} = 3.3\text{V}^{(5)}$	送信を実行していない状態
		180.0	214.0	$\text{mA}$	$-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$		送信を実行している状態

- Note 1:** スリープ時のパワーダウン電流は、オシレータの種類に関わらず一定です。パワーダウン電流は、デバイスをスリープに移行させ、全ての I/O ピンをハイインピーダンス状態にして  $V_{DD}$  または  $V_{SS}$  に接続し、電流増 ( $\Delta$  電流) をもたらず機能 (WDT、Timer1、オシレータ等) を全て無効にして計測しています。
- 2:** 消費電流は主に動作電圧、周波数、モードによって決まります。他の要因として I/O ピンの負荷とスイッチング レート、オシレータのタイプと回路、内部コード実行パターン、温度等があり、これらも消費電流に影響を与えます。アクティブ動作モードにおける  $I_{DD}$  の計測値は、全て以下の条件でテストしています。  
OSC1 = 外部矩形波 (レールツーレール)、I/O ピンは全て 3 ステートで  $V_{DD}$  にプル、  
MCLR =  $V_{DD}$ 、WDT は指定の通り有効化または無効化しています。
- 3:** 標準的な低コストの 32 kHz 水晶振動子の動作温度レンジは  $-10 \sim +70^{\circ}\text{C}$  です。コストは増加しますが、拡張温度仕様に対応した水晶振動子も利用可能です。
- 4:** 電圧レギュレータを無効に ( $ENVREG = 0$ 、 $V_{SS}$  に接続) した場合はです。
- 5:** 電圧レギュレータを有効に ( $ENVREG = 1$ 、 $V_{DD}$  に接続) した場合はです。
- 6:**  $\Delta I_{ETH}$  を規定する電流には、 $TP_{OUT+}$  と  $TP_{OUT-}$  のシンク電流が含まれます。LEDA と LEDB は全てのテストで無効です。

# PIC18F97J60 ファミリ

## 28.3 DC 特性 : PIC18F97J60 ファミリ (産業用)

DC 特性		標準動作条件 (特に明記しない場合) 動作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (産業用温度レンジ)					
パラメータ 番号	記号	特性	最小値	最大値	単位	条件	
D030	VIL	入力 Low 電圧 全 I/O ポート : TTL バッファ使用	VSS	0.15VDD	V	VDD < 2.7 V	
D031			VSS	0.8	V		2.7 V ≤ VDD ≤ 3.6 V
D032		シュミットトリガ バッファ使用	VSS	0.2 VDD	V	HS、HSPLL モード EC モード	
D033		MCLR	VSS	0.2 VDD	V		
D033A		OSC1	VSS	0.3 VDD	V		
D034		T13CKI	VSS	0.3	V		
D040		VIH	入力 High 電圧 アナログ機能付き I/O ポート : TTL バッファ使用	0.25 VDD + 0.8 V	VDD	V	HS、HSPLL モード EC モード
D041	シュミットトリガ バッファ使用			0.8 VDD	VDD	V	
D042	デジタル専用 I/O ポート : TTL バッファ使用		0.25 VDD + 0.8 V	5.5	V		
D043	シュミットトリガ バッファ使用		0.8 VDD	5.5	V		
D043A	MCLR		0.8 VDD	VDD	V		
D044	OSC1		0.7 VDD	VDD	V		
	OSC1		0.8 VDD	VDD	V		
	T13CKI		1.6	VDD	V		
D060	IIL	入力リーク電流 <sup>(1)</sup> I/O ポート	—	±1	μA	VSS ≤ VPIN ≤ VDD、 ハイインピーダンス状態の ピン	
D061			MCLR	—	±1		μA
D063			OSC1	—	±1		μA
D070	IPU IPURB	弱プルアップ電流 PORTB、PORTD、PORTE、 PORTJ	80	400	μA	VDD = 3.3 V、VPIN = VSS	

Note 1: 負の電流値は、ピンによるソース電流として定義しています。

# PIC18F97J60 ファミリ

## 28.3 DC 特性 : PIC18F97J60 ファミリ (産業用) (続き)

DC 特性			標準動作条件 (特に明記しない場合) 動作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (産業用温度レンジ)			
パラメータ 番号	記号	特性	最小値	最大値	単位	条件
D080	VOL	<b>出力 Low 電圧</b>				
		I/O ポート:				
		PORTD、PORTE、 PORTJ	—	0.4	V	$I_{OL} = 4\text{ mA}$ 、 $V_{DD} = 3.3\text{ V}$ 、 $-40 \sim +85^{\circ}\text{C}$
		PORTA<5:2>、PORTF、 PORTG、PORTH	—	0.4	V	$I_{OL} = 2\text{ mA}$ 、 $V_{DD} = 3.3\text{ V}$ 、 $-40 \sim +85^{\circ}\text{C}$
D083	VOL	PORTA<1:0>、PORTB、 PORTC	—	0.4	V	$I_{OL} = 8\text{ mA}$ 、 $V_{DD} = 3.3\text{ V}$ 、 $-40 \sim +85^{\circ}\text{C}$
		OSC2/CLKO (EC、ECPLL モード)	—	0.4	V	$I_{OL} = 2\text{ mA}$ 、 $V_{DD} = 3.3\text{ V}$ 、 $-40 \sim +85^{\circ}\text{C}$
D090	VOH	<b>出力 High 電圧<sup>(1)</sup></b>				
		I/O ポート:				
		PORTD、PORTE、 PORTJ	2.4	—	V	$I_{OH} = -4\text{ mA}$ 、 $V_{DD} = 3.3\text{ V}$ 、 $-40 \sim +85^{\circ}\text{C}$
		PORTA<5:2>、PORTF、 PORTG、PORTH	2.4	—	V	$I_{OH} = -2\text{ mA}$ 、 $V_{DD} = 3.3\text{ V}$ 、 $-40 \sim +85^{\circ}\text{C}$
D092	VOH	PORTA<1:0>、PORTB、 PORTC	2.4	—	V	$I_{OH} = -8\text{ mA}$ 、 $V_{DD} = 3.3\text{ V}$ 、 $-40 \sim +85^{\circ}\text{C}$
		OSC2/CLKO (EC、ECPLL モード)	2.4	—	V	$I_{OH} = -1.0\text{ mA}$ 、 $V_{DD} = 3.3\text{ V}$ 、 $-40 \sim +85^{\circ}\text{C}$
D100	COSC2	<b>出力ピンに対する容量性負荷の仕様</b> OSC2 ピン	—	15	pF	OSC1 の駆動に外部クロックを用いた HS モードの場合
D101	Cio	全 I/O ピンと OSC2 (内部 RC モード、EC、ECPLL の 場合)	—	50	pF	AC タイミング仕様を満たすため
D102	CB	SCLx、SDAx	—	400	pF	I <sup>2</sup> C™ 仕様

**Note 1:** 負の電流値は、ピンによるソース電流として定義しています。



# PIC18F97J60 ファミリ

表 28-1: メモリプログラミングの要件

DC 特性			標準動作条件 (特に明記しない場合) 動作温度 $-40\text{ }^{\circ}\text{C} \leq T_A \leq +85\text{ }^{\circ}\text{C}$ (産業用温度レンジ)				
パラメータ 番号	記号	特性	最小値	代表値†	最大値	単位	条件
D130	EP	プログラムフラッシュメモリ セル書き込み耐性	100	1K	-	E/W	$-40 \sim +85\text{ }^{\circ}\text{C}$
D131	VPR	読み出し用 VDD	V <sub>MIN</sub>	—	3.6	V	V <sub>MIN</sub> = 最小動作電圧
D132B	VPEW	自己タイマによる消去または 書き込みの電圧					
		VDD	2.70	—	3.6	V	ENVREG を VDD に接続
		VDDCORE	2.35	-	2.7	V	ENVREG を Vss に接続
D133A	TiW	自己タイマによる書き込みの サイクル時間	—	2.8	—	ms	
D134	TRETD	特性保持期間	20	-	—	年	他の仕様値に違反していない 場合
D135	IDDP	プログラミング中の消費電流	—	10	—	mA	Ethernet モジュールは無効

† 「代表値」欄のデータは、特に明記しない限り 3.3 V、25 °C の場合の値です。これらのパラメータは設計上の目安であり、テストしていません。

# PIC18F97J60 ファミリ

表 28-2: コンパレータの仕様

動作条件: $3.0\text{ V} \leq V_{DD} \leq 3.6\text{ V}$ 、 $-40\text{ }^{\circ}\text{C} \leq T_A \leq +85\text{ }^{\circ}\text{C}$ (特に明記しない場合)							
パラメータ番号	記号	特性	最小値	代表値	最大値	単位	備考
D300	VIOFF	入力オフセット電圧 *	—	$\pm 5.0$	$\pm 25$	mV	
D301	VICM	入力コモンモード電圧 *	0	—	$AV_{DD} - 1.5$	V	
D302	CMRR	コモンモード除去率 *	55	—	—	dB	
300	TRESP	応答時間 <sup>(1)</sup> *	—	150	400	ns	
301	TMC2OV	コンパレータのモード変更から出力有効までの時間 *	—	—	10	$\mu\text{s}$	

\* これらのパラメータは特性データであり、テストしていません。

**Note 1:** 応答時間は、コンパレータの一方の入力を  $(AV_{DD} - 1.5)/2$  とし、もう一方の入力を  $V_{SS}$  から  $AV_{DD}$  に遷移させて計測しています。

表 28-3: 参照電圧仕様

動作条件: $3.0\text{ V} \leq V_{DD} \leq 3.6\text{ V}$ 、 $-40\text{ }^{\circ}\text{C} \leq T_A \leq +85\text{ }^{\circ}\text{C}$ (特に明記しない場合)							
パラメータ番号	記号	特性	最小値	代表値	最大値	単位	備考
D310	VRES	分解能	$V_{DD}/24$	—	$V_{DD}/32$	LSb	
D311	VRAA	絶対精度	—	—	1/2	LSb	
D312	VRUR	単位抵抗値 (R)	—	2k	—	$\Omega$	
310	TSET	セトリングタイム <sup>(1)</sup>	—	—	10	$\mu\text{s}$	

**Note 1:** セトリングタイムは、 $CVRR = 1$  とし、 $CVR<3:0>$  を「0000」から「1111」に遷移させて計測しています。

表 28-4: 内部電圧レギュレータ仕様

動作条件: $-40\text{ }^{\circ}\text{C} \leq T_A \leq +85\text{ }^{\circ}\text{C}$ (特に明記しない場合)							
パラメータ番号	記号	特性	最小値	代表値	最大値	単位	備考
	VRGOUT	レギュレータ出力電圧	—	2.5	—	V	
	CF	外部フィルタ コンデンサ値	1	10	—	$\mu\text{F}$	コンデンサは低直列抵抗にする

## 28.4 AC (タイミング) 特性

### 28.4.1 タイミングパラメータの記号

タイミングパラメータの記号は、以下のいずれかの書式で表します。

- |             |           |                         |
|-------------|-----------|-------------------------|
| 1. TppS2ppS | 3. TCC:ST | (I <sup>2</sup> C 仕様のみ) |
| 2. TppS     | 4. Ts     | (I <sup>2</sup> C 仕様のみ) |

T		T	時間
F	周波数		

小文字 (pp) の種類と意味:

pp			
cc	ECCP1	osc	OSC1
ck	CLKO	rd	$\overline{RD}$
cs	$\overline{CS}$	rw	$\overline{RD}$ または $\overline{WR}$
di	SDIx	sc	SCKx
do	SDOx	ss	$\overline{SSx}$
dt	データ入力	t0	T0CKI
io	I/O ポート	t1	T13CKI
mc	MCLR	wr	$\overline{WR}$

大文字の種類と意味:

S			
F	立ち下がり	P	周期
H	High	R	立ち上がり
I	無効 (ハイインピーダンス)	V	有効
L	Low	Z	ハイインピーダンス
I <sup>2</sup> C のみ			
AA	出力アクセス	High	High
BUF	バスフリー	Low	Low

TCC:ST (I<sup>2</sup>C 仕様のみ)

CC			
HD	ホールド	SU	セットアップ
ST			
DAT	データ入力ホールド	STO	ストップ条件
STA	スタート条件		

# PIC18F97J60 ファミリ

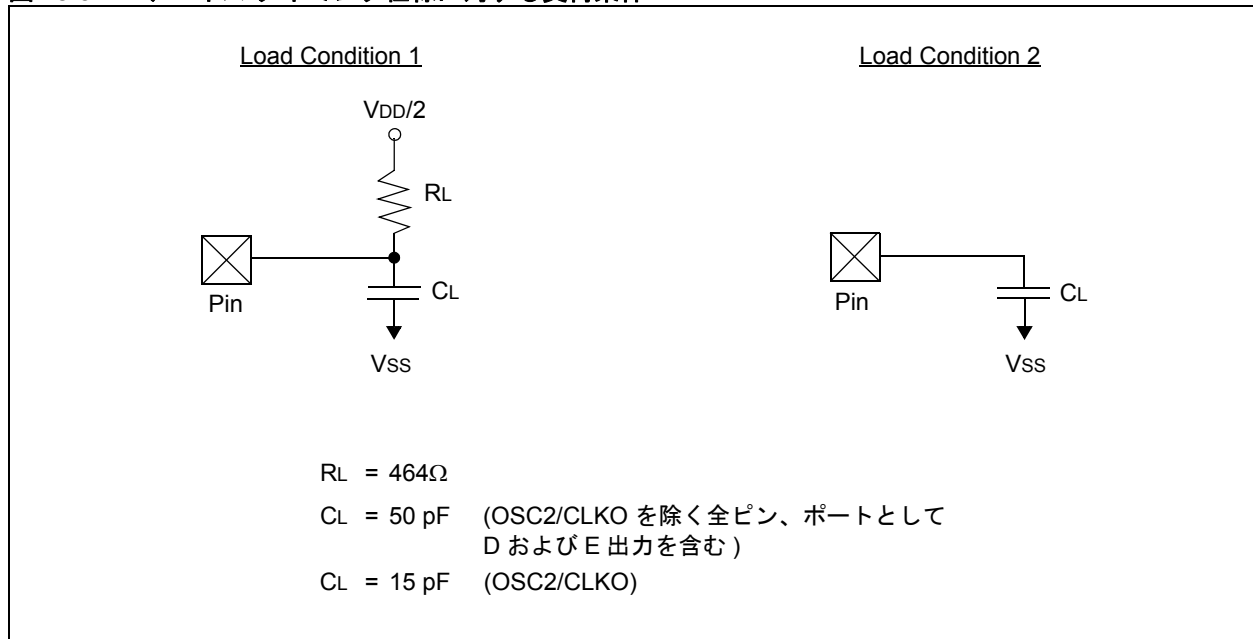
## 28.4.2 タイミングの条件

表 28-5 に指定された温度と電圧は、特に明記しない限り、全てのタイミング仕様に適用されます。図 28-3 に、タイミング仕様の負荷条件を示します。

表 28-5: 温度および電圧仕様 - AC

AC 特性	標準動作条件 (特に明記しない場合)
	動作温度 $-40\text{ }^{\circ}\text{C} \leq T_A \leq +85\text{ }^{\circ}\text{C}$ (産業用温度レンジ) セクション 28.1 「DC 特性: 電源電圧 PIC18F97J60 ファミリ (産業用)」とセクション 28.3 「DC 特性: PIC18F97J60 ファミリ (産業用)」の DC 仕様に記載された動作電圧 $V_{DD}$ レンジ

図 28-3: デバイス タイミング仕様に対する負荷条件



## 28.4.3 タイミング チャートと仕様

図 28-4: 外部クロック タイミング (PLL を除く全モード)

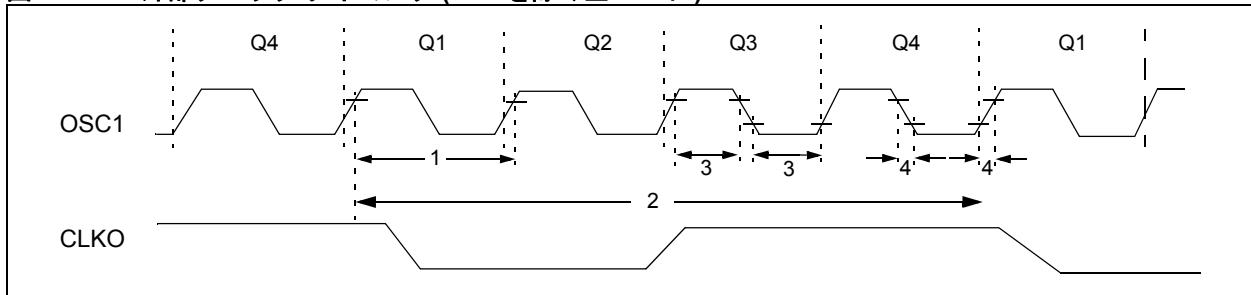


表 28-6: 外部クロックのタイミング要件

パラメータ番号	記号	特性	最小値	最大値	単位	条件
1A	FOSC	外部 CLKI 周波数 <sup>(1)</sup> オシレータ周波数 <sup>(1)</sup>	DC 6	41.6667 25	MHz MHz	EC オシレータモード HS オシレータモード
1	TOSC	外部 CLKI 周期 <sup>(1)</sup> オシレータ周期 <sup>(1)</sup>	24 40	— 167	ns ns	EC オシレータモード HS オシレータモード
2	TCY	命令サイクル時間 <sup>(1)</sup>	96	-	ns	TCY = 4/FOSC、産業用
3	TosL、 TosH	外部クロック入力 (OSC1) High または Low 時間	10	—	ns	EC オシレータモード
4	TosR、 TosF	外部クロック入力 (OSC1) の立ち上がり / 立ち下がり 時間	—	7.5	ns	EC オシレータモード
5		クロック周波数許容誤差	—	±50	ppm	Ethernet モジュールが有効

**Note 1:** PLL を除く全ての設定において、命令サイクル周期 (TCY) は、入力オシレータのタイムベース周期の 4 倍です。仕様の値は全て、標準の動作条件下でデバイスにコードを実行させた場合の特定のオシレータタイプの特性データに基づいています。これらの仕様の制限値を超えるとオシレータの動作が不安定になったり、予測以上の電流を消費する事があります。全てのデバイスは、OSC1/CLKI ピンに外部クロックを入力した状態で「最小値」で動作をテストしています。外部クロック入力を使った場合、サイクル時間の「最大」側の制限値はいずれのデバイスの場合も「DC」(クロックなし)です。

# PIC18F97J60 ファミリ

表 28-7: PLL クロック タイミング仕様 (VDD = 2.6 ~ 3.6 V)

パラメータ番号	記号	特性	最小値	代表値†	最大値	単位	条件
F10	FOSC	オシレータ周波数レンジ	8	—	25	MHz	HSPLL モード
			8	—	37.5	MHz	ECPLL モード
F11	FSYS	内蔵 VCO システム周波数	20	—	62.5	MHz	
F12	t <sub>rc</sub>	PLL 起動時間 (ロック時間)	—	—	2	ms	
F13	ΔCLK	CLKO 安定性 (ジッタ)	-2	—	+2	%	

† 「代表値」欄のデータは、特に明記しない限り 3.3 V、25 °C の場合の値です。これらのパラメータは設計上の目安であり、テストしていません。

表 28-8: AC 特性: 内部 RC 精度  
PIC18F97J60 ファミリ (産業用)

パラメータ番号	特性	最小値	代表値	最大値	単位	条件
	INTRC 精度 @ 周波数 = 31 kHz <sup>(1)</sup>	21.7	—	40.3	kHz	

Note 1: INTRC の周波数は、VDDCORE の変動と共に変化します。

図 28-5: CLKO と I/O タイミング

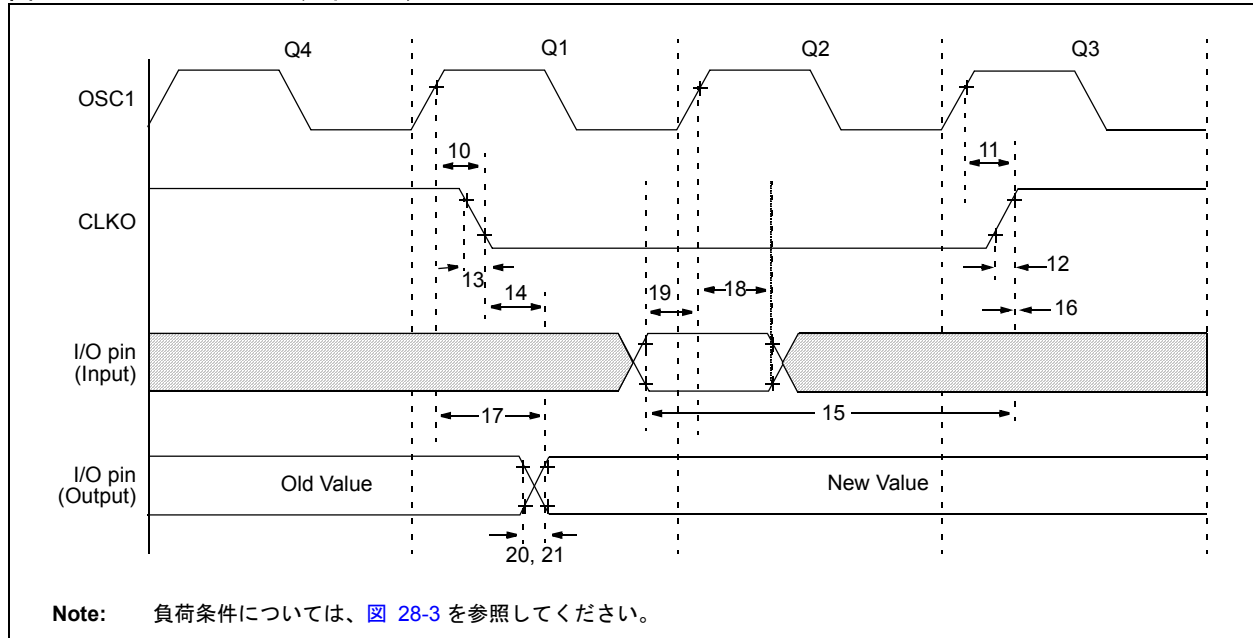


表 28-9: CLKO と I/O のタイミング要件

パラメータ番号	記号	特性	最小値	代表値	最大値	単位	条件
10	TosH2ckL	OSC1 ↑ から CLKO ↓	—	75	200	ns	
11	TosH2ckH	OSC1 ↑ から CLKO ↑	—	75	200	ns	
12	TckR	CLKO 立ち上がり時間	—	15	30	ns	
13	TckF	CLKO 立ち下がり時間	—	15	30	ns	
14	TckL2ioV	CLKO ↓ からポート出力有効	—	—	0.5 Tcy + 20	ns	
15	TioV2ckH	CLKO ↑ 前のポート入力有効	0.25 Tcy + 25	—	—	ns	
16	TckH2ioI	CLKO ↑ 後のポート入力ホールド	0	—	—	ns	
17	TosH2ioV	OSC1 ↑ (Q1 サイクル) からポート出力有効	—	50	150	ns	
18	TosH2ioI	OSC1 ↑ (Q2 サイクル) からポート入力無効 (I/O 入力のホールド時間)	100	—	—	ns	
19	TioV2osH	ポート入力有効から OSC1 ↑ (I/O 入力のセットアップ時間)	0	—	—	ns	
20	TioR	ポート出力立ち上がり時間	—	—	6	ns	
21	TioF	ポート出力立ち下がり時間	—	—	5	ns	
22 †	TINP	INTx ピンの High または Low 時間	Tcy	—	—	ns	
23 †	TRBP	RB<7:4> 状態変化 INTx の High または Low 時間	Tcy	—	—	ns	

† これらは非同期イベントに対するパラメータであり、内部クロックエッジとの関連性はありません。

# PIC18F97J60 ファミリ

図 28-6: プログラムメモリ読み出しタイミングチャート

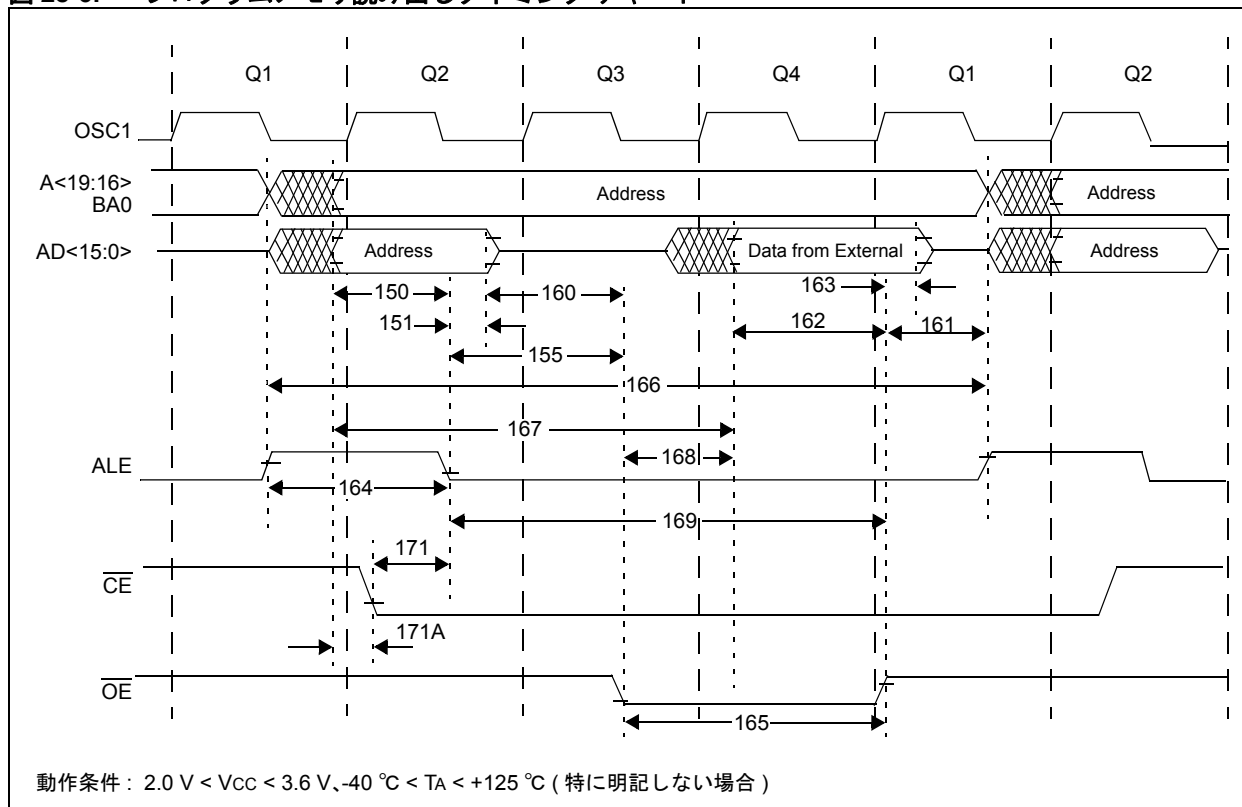


表 28-10: CLK0 と I/O のタイミング要件

パラメータ番号	記号	特性	最小値	代表値	最大値	単位
150	TadV2alL	アドレス出力有効から ALE↓ (アドレス セットアップ時間)	0.25 T <sub>CY</sub> - 10	—	—	ns
151	TalL2adl	ALE ↓ からアドレス出力無効 (アドレスホールド時間)	5	—	—	ns
155	TalL2oeL	ALE ↓ から $\overline{OE}$ ↓	10	0.125 T <sub>CY</sub>	—	ns
160	TadZ2oeL	AD ハイインピーダンスから $\overline{OE}$ ↓ (バス解放から $\overline{OE}$ )	0	—	—	ns
161	ToeH2adD	$\overline{OE}$ ↑ から AD 駆動	0.125 T <sub>CY</sub> - 5	—	—	ns
162	TadV2oeH	$\overline{OE}$ ↑ 前の最下位データ有効 (データ セットアップ時間)	20	—	—	ns
163	ToeH2adl	$\overline{OE}$ ↑ からデータ入力無効 (データホールド時間)	0	—	—	ns
164	TalH2alL	ALE パルス幅	—	T <sub>CY</sub>	—	ns
165	ToeL2oeH	$\overline{OE}$ パルス幅	0.5 T <sub>CY</sub> - 5	0.5 T <sub>CY</sub>	—	ns
166	TalH2alH	ALE ↑ から ALE ↑ (サイクル時間)	—	0.25 T <sub>CY</sub>	—	ns
167	Tacc	アドレス有効からデータ有効	0.75 T <sub>CY</sub> - 25	—	—	ns
168	Toe	$\overline{OE}$ ↓ からデータ有効	—	—	0.5 T <sub>CY</sub> - 25	ns
169	TalL2oeH	ALE ↓ から $\overline{OE}$ ↑	0.625 T <sub>CY</sub> - 10	—	0.625 T <sub>CY</sub> + 10	ns
171	TalH2csL	チップイネーブル アクティブから ALE ↓	0.25 T <sub>CY</sub> - 20	—	—	ns
171A	TubL2oeH	AD 有効からチップイネーブル アクティブ	—	—	10	ns



# PIC18F97J60 ファミリ

図 28-7: プログラムメモリ書き込みタイミングチャート

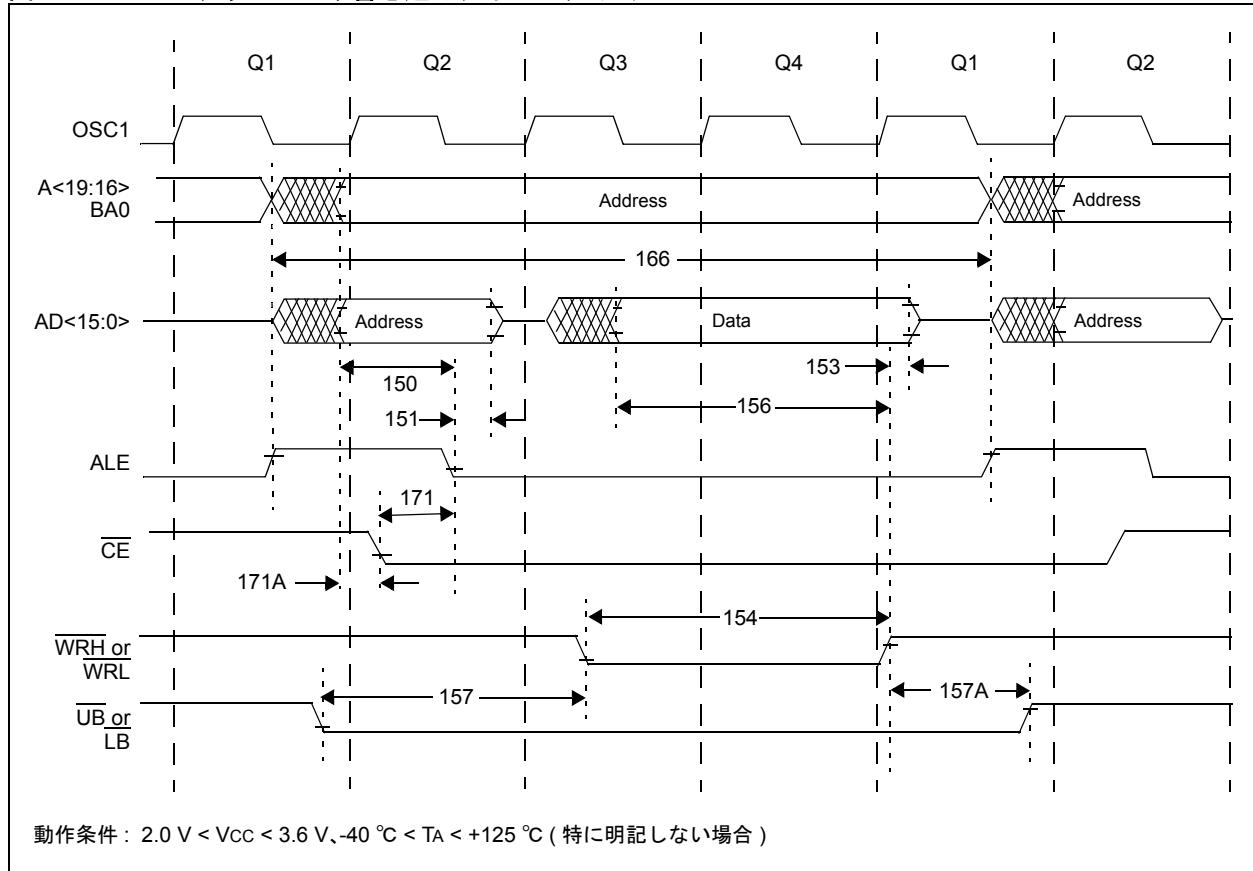


表 28-11: プログラムメモリ書き込みタイミング要件

パラメータ番号	記号	特性	最小値	代表値	最大値	単位
150	TadV2alL	アドレス出力有効から ALE ↓ (アドレス セットアップ時間)	0.25 T <sub>CY</sub> - 10	—	—	ns
151	TalL2adI	ALE ↓ からアドレス出力無効 (アドレスホールド時間)	5	—	—	ns
153	TwrH2adI	WR <sub>n</sub> ↑ からデータ出力無効 (データホールド時間)	5	—	—	ns
154	TwrL	WR <sub>n</sub> パルス幅	0.5 T <sub>CY</sub> - 5	0.5 T <sub>CY</sub>	—	ns
156	TadV2wrH	WR <sub>n</sub> ↑ 前のデータ有効 (データ セットアップ時間)	0.5 T <sub>CY</sub> - 10	—	—	ns
157	TbsV2wrL	WR <sub>n</sub> ↓ 前のバイト選択有効 (バイト選択セットアップ時間)	0.25 T <sub>CY</sub>	—	—	ns
157A	TwrH2bsI	WR <sub>n</sub> ↑ からバイト選択無効 (バイト選択ホールド時間)	0.125 T <sub>CY</sub> - 5	—	—	ns
166	TalH2alH	ALE ↑ から ALE ↑ (サイクル時間)	—	0.25 T <sub>CY</sub>	—	ns
171	TalH2csL	チップイネーブル アクティブから ALE ↓	0.25 T <sub>CY</sub> - 20	—	—	ns
171A	TubL2oeH	AD 有効からチップイネーブル アクティブ	—	—	10	ns

# PIC18F97J60 ファミリ

図 28-8: リセット、ウォッチドッグタイマ、オシレータ起動タイマ、パワーアップタイマのタイミング

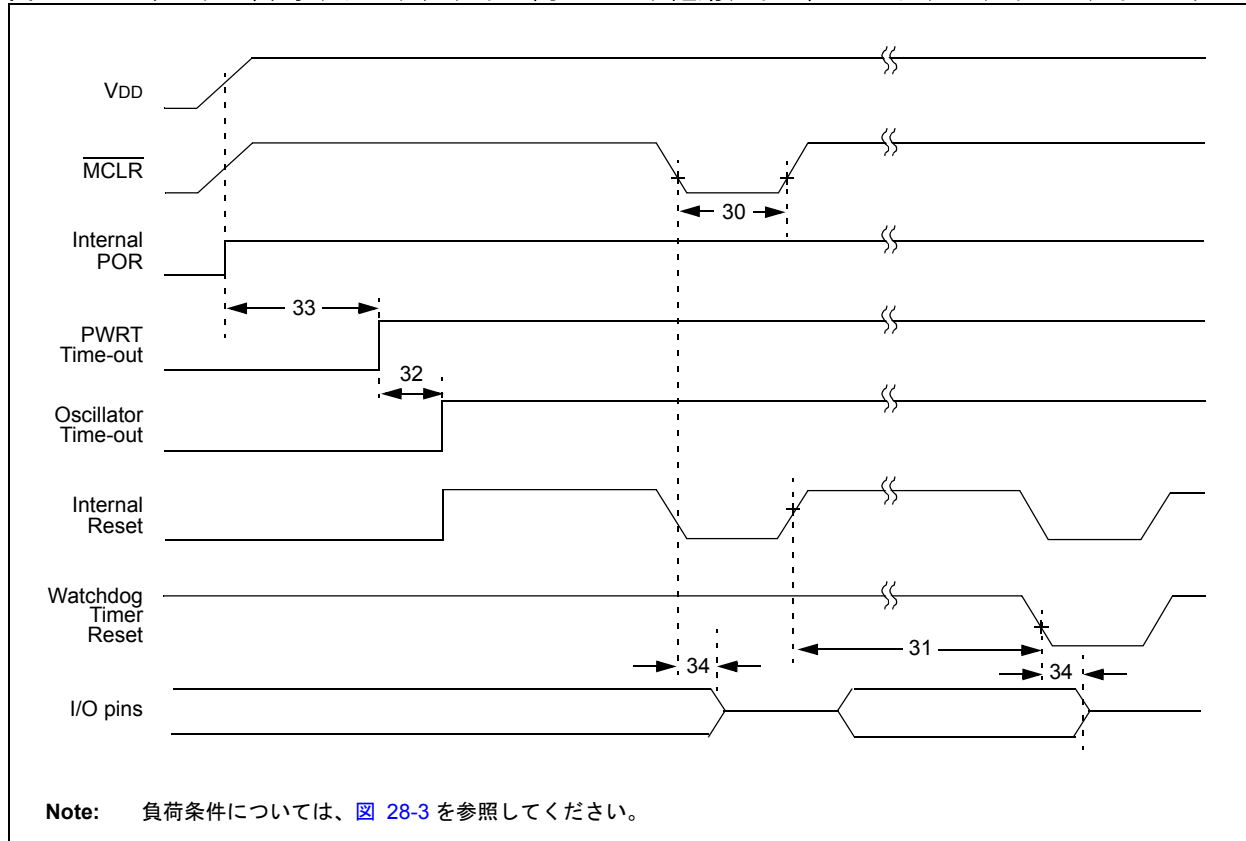


表 28-12: リセット、ウォッチドッグタイマ、オシレータ起動タイマ、パワーアップタイマ、ブラウンアウト リセットの要件

パラメータ番号	記号	特性	最小値	代表値	最大値	単位	条件
30	TMCL	MCLR パルス幅 (Low)	2	—	—	μs	
31	TWDT	ウォッチドッグタイマ タイムアウト時間 (ポストスケラなし)	2.8	4.1	5.4	ms	
32	TOST	オシレータ起動タイマ周期	1024 T <sub>osc</sub>	-	1024 T <sub>osc</sub>	—	T <sub>osc</sub> = OSC1 周期
33	TPWRT	パワーアップ タイマ周期	46.2	66	85.8	ms	
34	TIOZ	MCLR = Low またはウォッチドッグタイマリセットから I/O ピン High-Z	—	—	3T <sub>cy</sub> + 2	μs	システムクロック使用可能
			—	—	415	μs	システムクロック使用不可 (スリープまたはプライマリ オシレータ OFF)
38	T <sub>CSD</sub>	CPU 起動時間	—	200	—	μs	

# PIC18F97J60 ファミリ

図 28-9: Timer0 と Timer1 の外部クロック タイミング

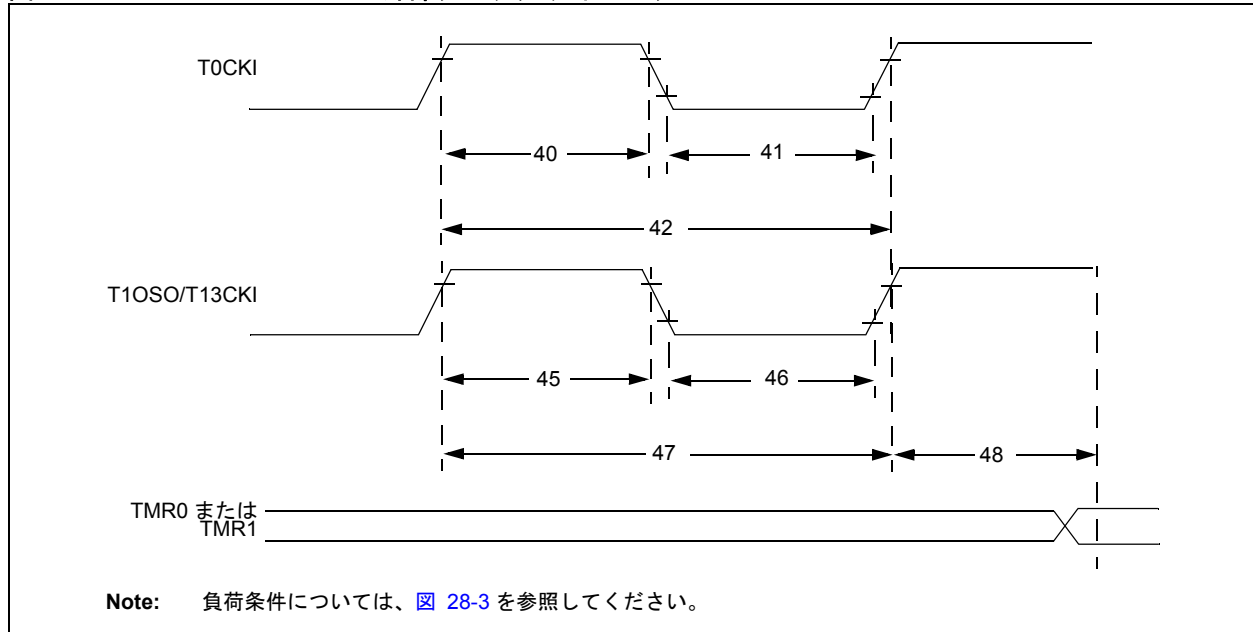


表 28-13: Timer0 と Timer1 の外部クロックの要件

パラメータ番号	記号	特性		最小値	最大値	単位	条件
40	Tt0H	T0CKI パルス幅 (High)	プリスケールなし	$0.5 T_{CY} + 20$	—	ns	
			プリスケールあり	10	—	ns	
41	Tt0L	T0CKI パルス幅 (Low)	プリスケールなし	$0.5 T_{CY} + 20$	—	ns	
			プリスケールあり	10	—	ns	
42	Tt0P	T0CKI 周期	プリスケールなし	$T_{CY} + 10$	—	ns	
			プリスケールあり	以下のいずれか大きい方: 20 ns または $(T_{CY} + 40)/N$	—	ns	
45	Tt1H	T13CKI High 時間	同期、プリスケールなし	$0.5 T_{CY} + 20$	—	ns	
			同期、プリスケールあり	10	—	ns	
			非同期	30	—	ns	
46	Tt1L	T13CKI Low 時間	同期、プリスケールなし	$0.5 T_{CY} + 5$	—	ns	
			同期、プリスケールあり	10	—	ns	
			非同期	30	—	ns	
47	Tt1P	T13CKI 入力周期	同期	以下のいずれか大きい方: 20 ns または $(T_{CY} + 40)/N$	—	ns	N = プリスケール値 (1, 2, 4, 8)
			非同期	60	—	ns	
	Ft1	T13CKI オシレータ入力周波数レンジ		DC	50	kHz	
48	Tcke2TMR1	外部 T13CKI クロックエッジからタイマのインクリメントまでの遅延		$2 T_{OSC}$	$7 T_{OSC}$	—	

# PIC18F97J60 ファミリ

図 28-10: キャプチャ/コンペア/PWM タイミング (ECCPx モジュールを含む)

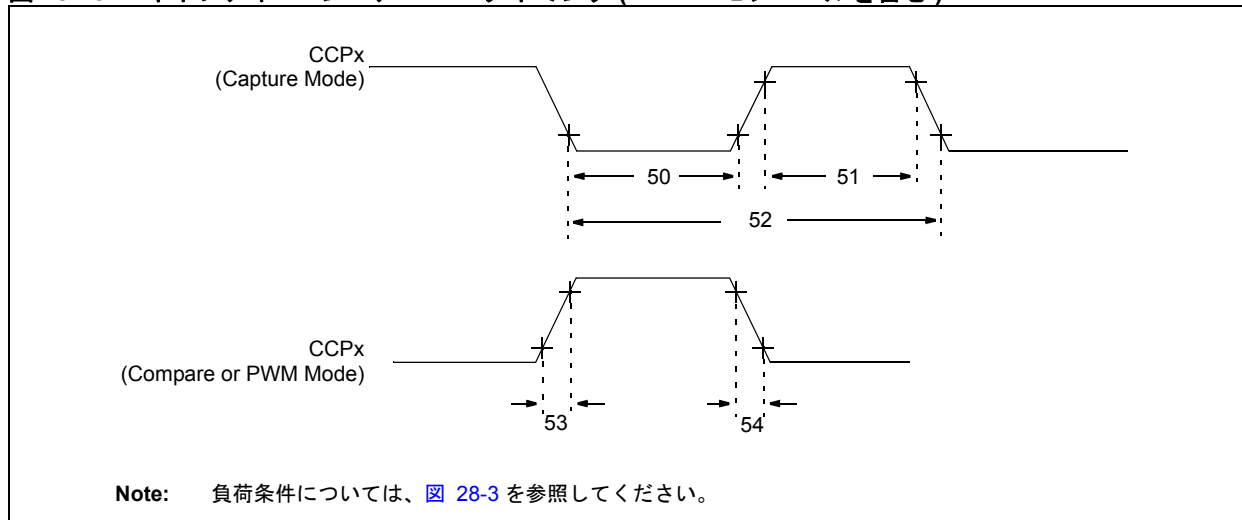


表 28-14: キャプチャ/コンペア/PWM の要件 (ECCPx モジュールを含む)

パラメータ番号	記号	特性	プリスケールなし	プリスケールあり	最小値	最大値	単位	条件
50	TccL	CCPx 入力 Low 時間	$0.5 T_{CY} + 20$	10	—	—	ns	
51	TccH	CCPx 入力 High 時間	$0.5 T_{CY} + 20$	10	—	—	ns	
52	TccP	CCPx 入力周期	$\frac{3 T_{CY} + 40}{N}$	—	—	—	ns	N = プリスケール値 (1, 4, 16)
53	TccR	CCPx 出力立ち上がり時間	—	—	—	25	ns	
54	TccF	CCPx 出力立ち下がり時間	—	—	—	25	ns	

表 28-15: パラレルスレーブポートの要件

パラメータ番号	記号	特性	最小値	最大値	単位	条件
62	TdtV2wrH	$\overline{WR}$ ↑ または $\overline{CS}$ ↑ 前のデータ入力有効 (セットアップ時間)	20	—	ns	
63	TwrH2dtl	$\overline{WR}$ ↑ または $\overline{CS}$ ↑ からデータ入力無効 (ホールド時間)	20	—	ns	
64	TrdL2dtV	$\overline{RD}$ ↓ かつ $\overline{CS}$ ↓ からデータ出力有効	—	80	ns	
65	TrdH2dtl	$\overline{RD}$ ↑ または $\overline{CS}$ ↓ からデータ出力無効	10	30	ns	
66	TibflNH	$\overline{WR}$ ↑ または $\overline{CS}$ ↑ から IBF フラグビットのクリア禁止	—	$3 T_{CY}$		

図 28-11: SPI マスタモードのタイミング例 (CKE = 0)

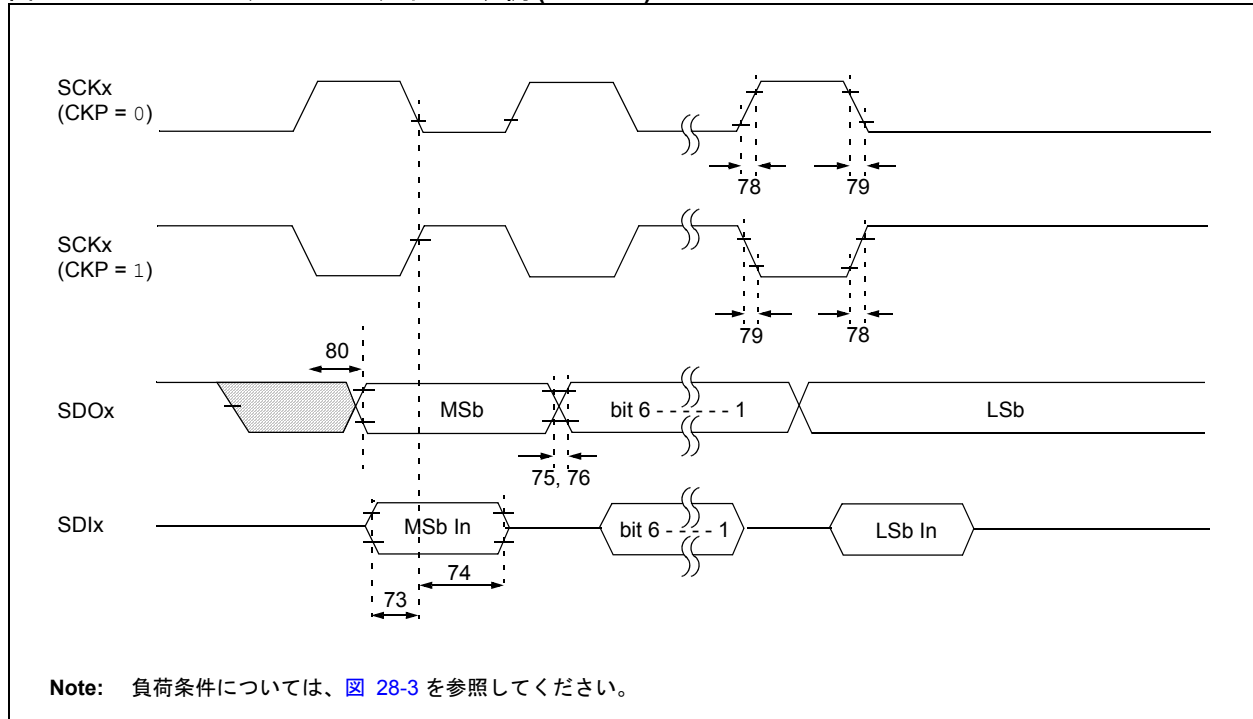


表 28-16: SPI モードの要件の例 ( マスタモード、Cke = 0)

パラメータ番号	記号	特性	最小値	最大値	単位	条件
73	TdIV2scH、 TdIV2scL	SCKx エッジに対する SDIx データ入力の セットアップ時間	100	—	ns	
74	Tsch2diL、 TscL2diL	SCKx エッジに対する SDIx データ入力の ホールド時間	100	—	ns	
75	TdoR	SDOx データ出力立ち上がり時間	—	25	ns	
76	TdoF	SDOx データ出力立ち下がり時間	—	25	ns	
78	TscR	SCKx 出力立ち上がり時間	—	25	ns	
79	TscF	SCKx 出力立ち下がり時間	—	25	ns	
80	Tsch2doV、 TscL2doV	SCKx エッジ後から SDOx データ出力有効	—	50	ns	

# PIC18F97J60 ファミリ

図 28-12: SPI マスタモードのタイミング例 (CKE = 1)

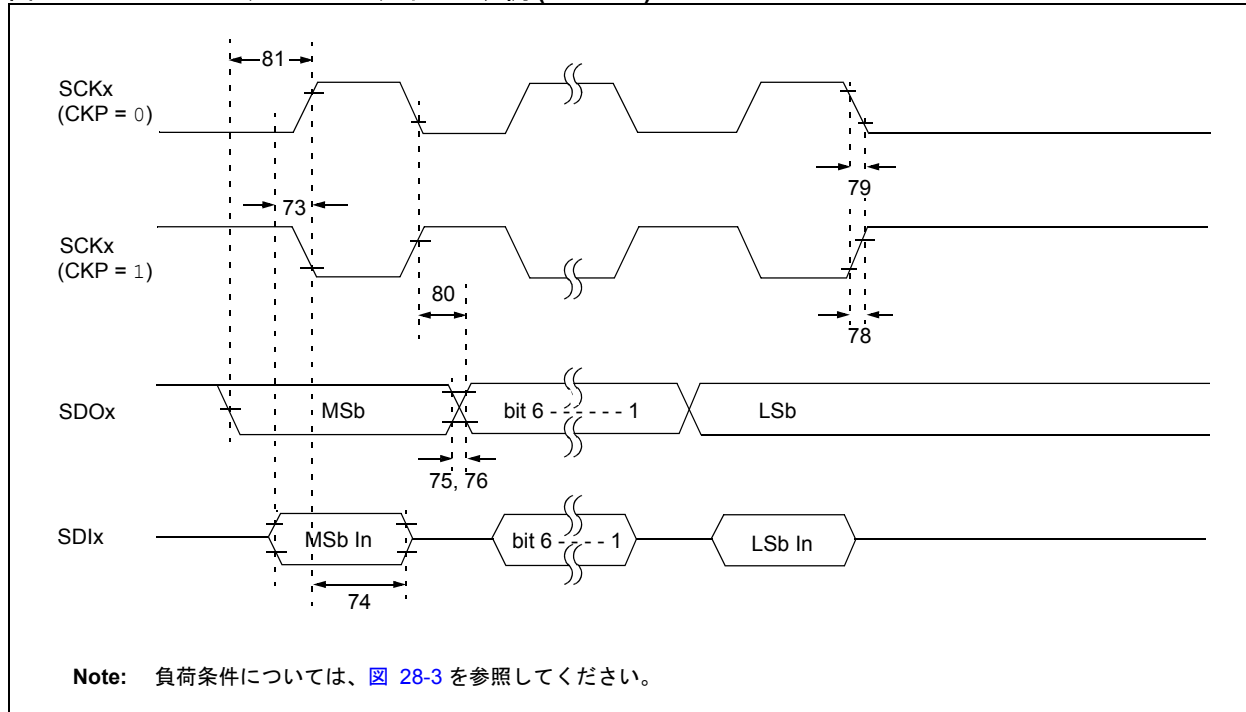


表 28-17: SPI モードの要件の例 ( マスタモード、CKE = 1)

パラメータ番号	記号	特性	最小値	最大値	単位	条件
73	TdIV2scH、 TdIV2scL	SCKx エッジに対する SDIx データ入力の セットアップ時間	100	—	ns	
74	Tsch2diL、 TscL2diL	SCKx エッジに対する SDIx データ入力の ホールド時間	100	—	ns	
75	TdoR	SDOx データ出力立ち上がり時間	—	25	ns	
76	TdoF	SDOx データ出力立ち下がり時間	—	25	ns	
78	TscR	SCKx 出力立ち上がり時間	—	25	ns	
79	TscF	SCKx 出力立ち下がり時間	—	25	ns	
80	Tsch2doV、 TscL2doV	SCKx エッジ後から SDOx データ出力有効	—	50	ns	
81	TdoV2scH、 TdoV2scL	SCKx エッジに対する SDOx データ出力 セットアップ	Tcy	—	ns	

# PIC18F97J60 ファミリ

図 28-13: SPI スレーブモードのタイミング例 (CKE = 0)

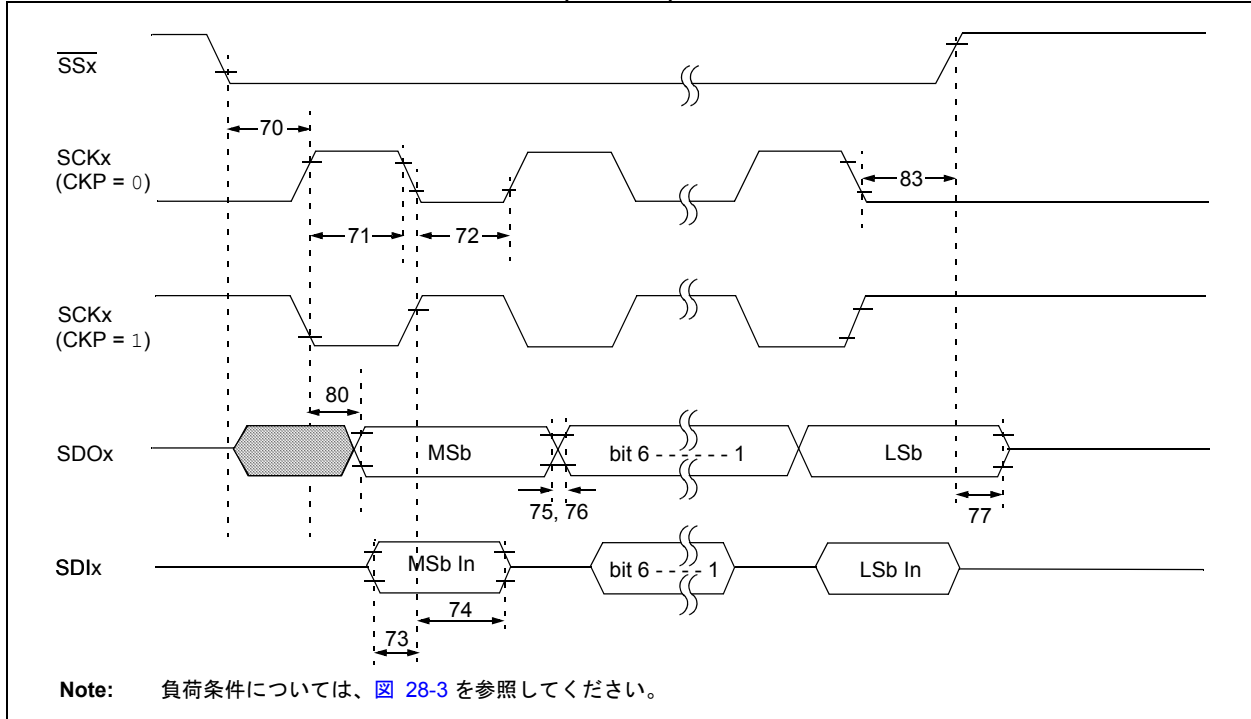


表 28-18: SPI モードの要件の例 (スレーブモード、CKE = 0)

パラメータ番号	記号	特性	最小値	最大値	単位	条件
70	TssL2scH、TssL2scL	SSx ↓ から SCKx ↓ または SCKx ↑ 入力	Tcy	—	ns	
71	Tsch	SCKx 入力 High 時間	連続	1.25 Tcy + 30	—	ns
71A			シングルバイト	40	—	ns
72	TscL	SCKx 入力 Low 時間	連続	1.25 Tcy + 30	—	ns
72A			シングルバイト	40	—	ns
73	TdiV2scH、TdiV2scL	SCKx エッジに対する SDIx データ入力のセットアップ時間	100	—	ns	
73A	Tb2B	バイト 1 の最後のクロックエッジからバイト 2 の最初のクロックエッジ	1.5 Tcy + 40	—	ns	(Note 2)
74	Tsch2diL、TscL2diL	SCKx エッジに対する SDIx データ入力のホールド時間	100	—	ns	
75	TdoR	SDOx データ出力立ち上がり時間	—	25	ns	
76	TdoF	SDOx データ出力立ち下がり時間	—	25	ns	
77	TssH2doZ	SSx ↑ から SDOx 出力ハイインピーダンス	10	50	ns	
80	Tsch2doV、TscL2doV	SCKx エッジ後から SDOx データ出力有効	—	50	ns	
83	Tsch2ssH、TscL2ssH	SCKx エッジ後から SSx ↑	1.5 Tcy + 40	—	ns	

**Note 1:** パラメータ #73A を使う必要があります。

**2:** パラメータ #71A と #72A を使う場合のみ適用されます。

# PIC18F97J60 ファミリ

図 28-14: SPI スレーブモードのタイミング例 (CKE = 1)

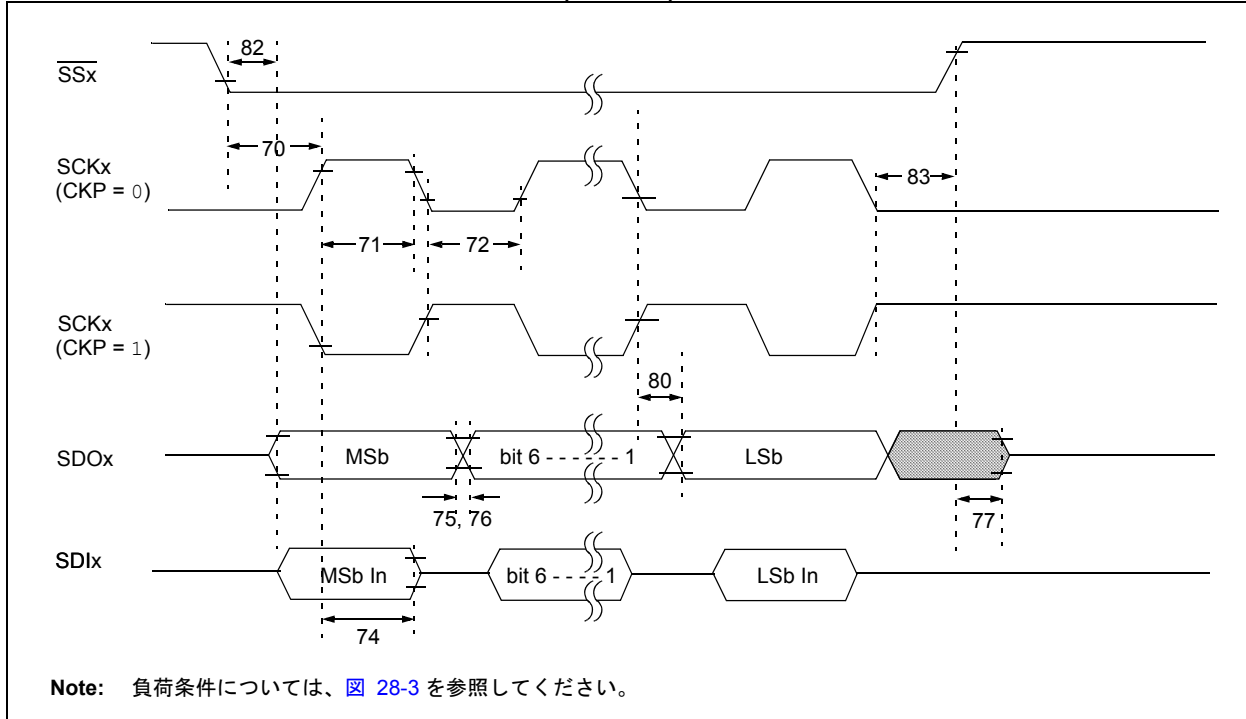


表 28-19: SPI スレーブモードの要件の例 (CKE = 1)

パラメータ番号	記号	特性	最小値	最大値	単位	条件
70	TssL2sch、TssL2scl	$\overline{SSx} \downarrow$ から $SCKx \downarrow$ または $SCKx \uparrow$ 入力	$T_{CY}$	—	ns	
71	Tsch	SCKx 入力 High 時間	連続	$1.25 T_{CY} + 30$	—	ns
71A			シングルバイト	40	—	ns (Note 1)
72	Tscl	SCKx 入力 Low 時間	連続	$1.25 T_{CY} + 30$	—	ns
72A			シングルバイト	40	—	ns (Note 1)
73A	Tb2b	バイト 1 の最後のクロックエッジからバイト 2 の最初のクロックエッジ	$1.5 T_{CY} + 40$	—	ns	(Note 2)
74	Tsch2diL、TscL2diL	SCKx エッジに対する $SDIx$ データ入力のホールド時間	100	—	ns	
75	TdoR	$SDOx$ データ出力立ち上がり時間	—	25	ns	
76	TdoF	$SDOx$ データ出力立ち下がり時間	—	25	ns	
77	TssH2doZ	$\overline{SSx} \uparrow$ から $SDOx$ 出力ハイインピーダンス	10	50	ns	
80	Tsch2doV、TscL2doV	SCKx エッジ後から $SDOx$ データ出力有効	—	50	ns	
82	TssL2doV	$\overline{SSx} \downarrow$ エッジ後から $SDOx$ データ出力有効	—	50	ns	
83	Tsch2ssh、TscL2ssh	SCKx エッジ後から $\overline{SSx} \uparrow$	$1.5 T_{CY} + 40$	—	ns	

**Note 1:** パラメータ #73A を使う必要があります。

**2:** パラメータ #71A と #72A を使う場合のみ適用されます。



図 28-15: I<sup>2</sup>C™ バス スタート/ストップビットのタイミング

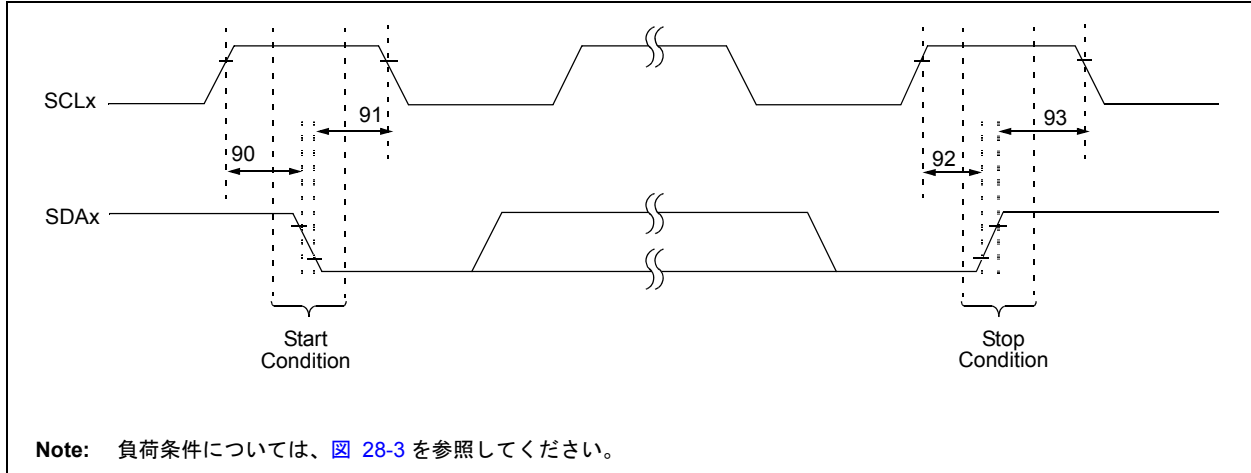
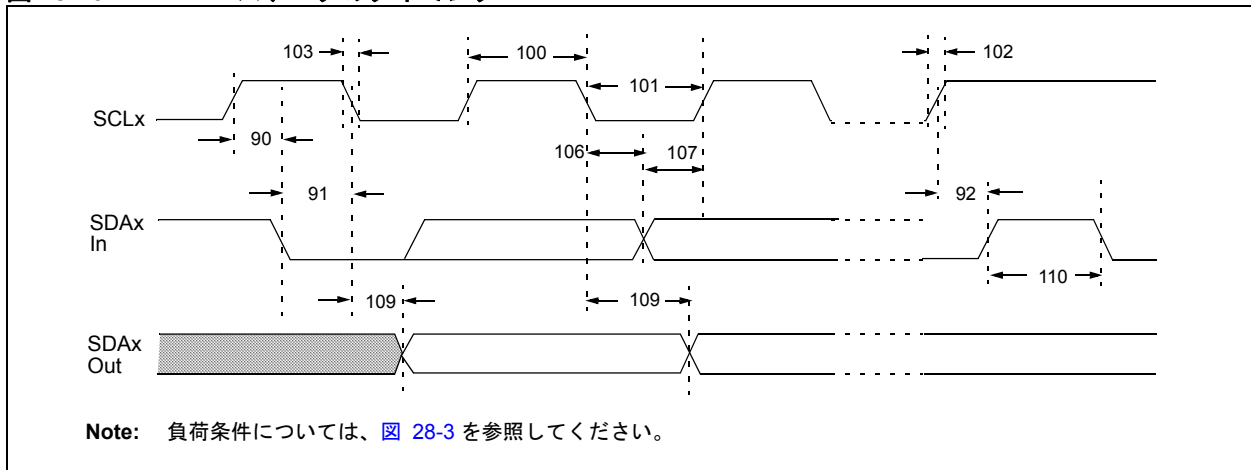


表 28-20: I<sup>2</sup>C™ バス スタート/ストップビットの要件 (スレーブモード)

パラメータ番号	記号	特性	最小値	最大値	単位	条件	
90	TSU:STA	スタート条件	100 kHz モード	4700	—	ns	反復スタート条件時のみ
		セットアップ時間	400 kHz モード	600	—		
91	THD:STA	スタート条件	100 kHz モード	4000	—	ns	この期間後、最初のクロックパルスが生成される
		ホールド時間	400 kHz モード	600	—		
92	TSU:STO	ストップ条件	100 kHz モード	4700	—	ns	
		セットアップ時間	400 kHz モード	600	—		
93	THD:STO	ストップ条件	100 kHz モード	4000	—	ns	
		ホールド時間	400 kHz モード	600	—		

図 28-16: I<sup>2</sup>C™ バスデータのタイミング



# PIC18F97J60 ファミリ

表 28-21: I<sup>2</sup>C™ バスデータ要件 (スレーブモード)

パラメータ番号	記号	特性		最小値	最大値	単位	条件
100	THIGH	クロック High 時間	100 kHz モード	4.0	—	μs	PIC18F97J60 ファミリが 1.5 MHz 以上で動作する必要がある
			400 kHz モード	0.6	—	μs	PIC18F97J60 ファミリが 10 MHz 以上で動作する必要がある
			MSSP モジュール	1.5 T <sub>cy</sub>	—		
101	TLOW	クロック Low 時間	100 kHz モード	4.7	—	μs	PIC18F97J60 ファミリが 1.5 MHz 以上で動作する必要がある
			400 kHz モード	1.3	—	μs	PIC18F97J60 ファミリが 10 MHz 以上で動作する必要がある
			MSSP モジュール	1.5 T <sub>cy</sub>	—		
102	TR	SDAx と SCLx 立ち上がり時間	100 kHz モード	—	1000	ns	
			400 kHz モード	20 + 0.1 C <sub>B</sub>	300	ns	C <sub>B</sub> の規定値は 10 ~ 400 pF
103	TF	SDAx と SCLx 立ち下がり時間	100 kHz モード	—	300	ns	
			400 kHz モード	20 + 0.1 C <sub>B</sub>	300	ns	C <sub>B</sub> の規定値は 10 ~ 400 pF
90	TSU:STA	スタート条件 セットアップ時間	100 kHz モード	4.7	—	μs	反復スタート条件時のみ
			400 kHz モード	0.6	—	μs	
91	THD:STA	スタート条件 ホールド時間	100 kHz モード	4.0	—	μs	この期間後、最初のクロックパルスが生成される
			400 kHz モード	0.6	—	μs	
106	THD:DAT	データ入力 ホールド時間	100 kHz モード	0	—	ns	
			400 kHz モード	0	0.9	μs	
107	TSU:DAT	データ入力 セットアップ時間	100 kHz モード	250	—	ns	(Note 2)
			400 kHz モード	100	—	ns	
92	TSU:STO	ストップ条件 セットアップ時間	100 kHz モード	4.7	—	μs	
			400 kHz モード	0.6	—	μs	
109	TAA	クロックから出力有効	100 kHz モード	—	3500	ns	(Note 1)
			400 kHz モード	—	—	ns	
110	TBUF	バスフリー時間	100 kHz モード	4.7	—	μs	次の転送が開始可能になるまでに必要なバスの空き時間
			400 kHz モード	1.3	—	μs	
D102	CB	バス容量性負荷		—	400	pF	

- Note 1:** デバイスがトランスマッタの場合、意図しないスタート/ストップ条件の発生を防ぐために SCLx の立ち下がりエッジの未定義領域をブリッジする内部最小遅延時間 (min. 300 ns) を確保する必要があります。
- 2:** 高速モードの I<sup>2</sup>C™ バスデバイスを標準モードの I<sup>2</sup>C バスシステムで使う事は可能ですが、その場合も TSU:DAT ≥ 250 ns の要件を満たす必要があります。デバイスが SCLx 信号の Low 期間を伸長しない場合、自動的にこの状態となります。SCLx 信号の Low 期間を伸長する場合、標準モードの I<sup>2</sup>C バス仕様に従い、SCLx ラインを解放する Tr max. + TSU:DAT = 1000 + 250 = 1250 ns 前に、次のデータビットを SDAx ラインに出力する必要があります。

# PIC18F97J60 ファミリ

図 28-17: マスタ SSP I<sup>2</sup>C™ バス スタート/ストップビットのタイミング波形

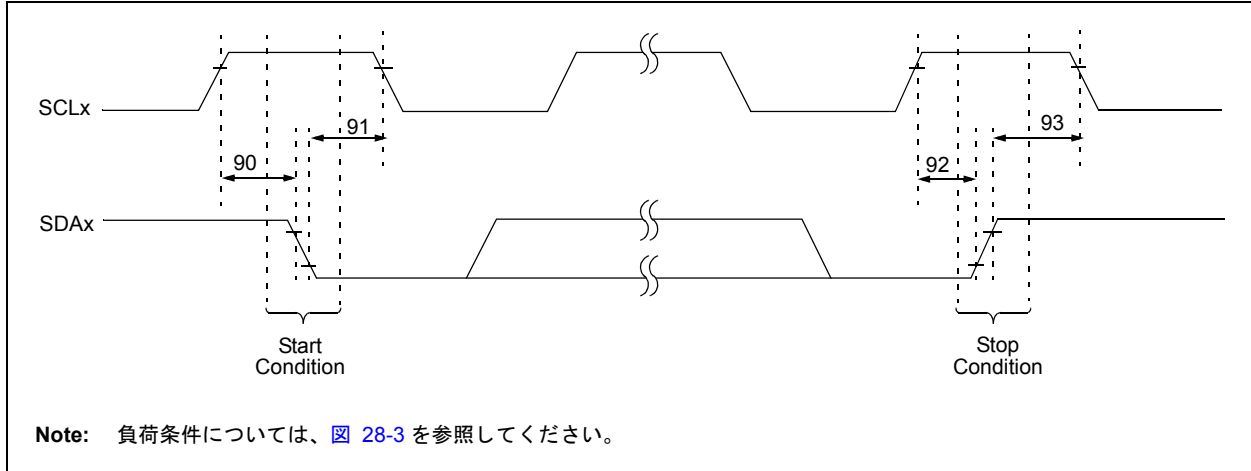
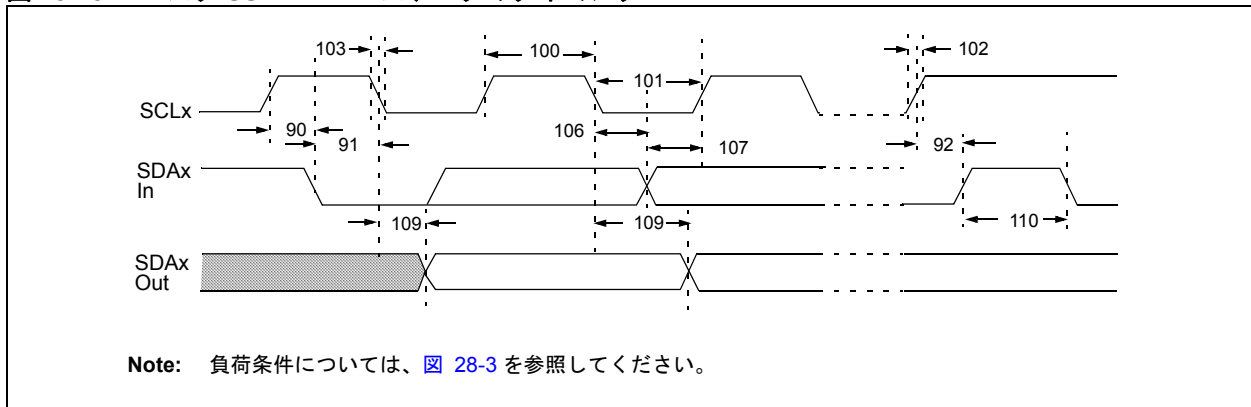


表 28-22: マスタ SSP I<sup>2</sup>C™ バス スタート/ストップビットの要件

パラメータ番号	記号	特性	最小値	最大値	単位	条件	
90	TSU:STA	スタート条件 セットアップ時間	100 kHz モード	2 (Tosc)(BRG + 1)	—	ns	反復スタート条件の場合のみ
			400 kHz モード	2 (Tosc)(BRG + 1)	—		
			1 MHz モード <sup>(1)</sup>	2 (Tosc)(BRG + 1)	—		
91	THD:STA	スタート条件 ホールド時間	100 kHz モード	2 (Tosc)(BRG + 1)	—	ns	この期間後、最初のクロックパルスが生成される
			400 kHz モード	2 (Tosc)(BRG + 1)	—		
			1 MHz モード <sup>(1)</sup>	2 (Tosc)(BRG + 1)	—		
92	TSU:STO	ストップ条件 セットアップ時間	100 kHz モード	2 (Tosc)(BRG + 1)	—	ns	
			400 kHz モード	2 (Tosc)(BRG + 1)	—		
			1 MHz モード <sup>(1)</sup>	2 (Tosc)(BRG + 1)	—		
93	THD:STO	ストップ条件 ホールド時間	100 kHz モード	2 (Tosc)(BRG + 1)	—	ns	
			400 kHz モード	2 (Tosc)(BRG + 1)	—		
			1 MHz モード <sup>(1)</sup>	2 (Tosc)(BRG + 1)	—		

Note 1: 全ての I<sup>2</sup>C™ ピンの最大静電容量は 10 pF です。

図 28-18: マスタ SSP I<sup>2</sup>C™ バスデータのタイミング



# PIC18F97J60 ファミリ

表 28-23: マスタ SSP I<sup>2</sup>C™ バスデータの要件

パラメータ番号	記号	特性		最小値	最大値	単位	条件
100	THIGH	クロック High 時間	100 kHz モード	2 (Tosc)(BRG + 1)	—	ms	
			400 kHz モード	2 (Tosc)(BRG + 1)	—	ms	
			1 MHz モード (1)	2 (Tosc)(BRG + 1)	—	ms	
101	TLOW	クロック Low 時間	100 kHz モード	2 (Tosc)(BRG + 1)	—	ms	
			400 kHz モード	2 (Tosc)(BRG + 1)	—	ms	
			1 MHz モード (1)	2 (Tosc)(BRG + 1)	—	ms	
102	TR	SDAx と SCLx 立ち上がり時間	100 kHz モード	—	1000	ns	Cb の規定値は 10 ~ 400 pF
			400 kHz モード	20 + 0.1 Cb	300	ns	
			1 MHz モード (1)	—	300	ns	
103	TF	SDAx と SCLx 立ち下がり時間	100 kHz モード	—	300	ns	Cb の規定値は 10 ~ 400 pF
			400 kHz モード	20 + 0.1 Cb	300	ns	
			1 MHz モード (1)	—	100	ns	
90	TSU:STA	スタート条件 セットアップ時間	100 kHz モード	2 (Tosc)(BRG + 1)	—	ms	反復スタート条件の場合のみ
			400 kHz モード	2 (Tosc)(BRG + 1)	—	ms	
			1 MHz モード (1)	2 (Tosc)(BRG + 1)	—	ms	
91	THD:STA	スタート条件 ホールド時間	100 kHz モード	2 (Tosc)(BRG + 1)	—	ms	この期間後、最初のクロックパルスが生成される
			400 kHz モード	2 (Tosc)(BRG + 1)	—	ms	
			1 MHz モード (1)	2 (Tosc)(BRG + 1)	—	ms	
106	THD:DAT	データ入力 ホールド時間	100 kHz モード	0	—	ns	
			400 kHz モード	0	0.9	ms	
			1 MHz モード (1)	TBD	—	ns	
107	TSU:DAT	データ入力 セットアップ時間	100 kHz モード	250	—	ns	(Note 2)
			400 kHz モード	100	—	ns	
			1 MHz モード (1)	TBD	—	ns	
92	TSU:STO	ストップ条件 セットアップ時間	100 kHz モード	2 (Tosc)(BRG + 1)	—	ms	
			400 kHz モード	2 (Tosc)(BRG + 1)	—	ms	
			1 MHz モード (1)	2 (Tosc)(BRG + 1)	—	ms	
109	TAA	クロックから出力有効	100 kHz モード	—	3500	ns	
			400 kHz モード	—	1000	ns	
			1 MHz モード (1)	—	—	ns	
110	TBUF	バスフリー時間	100 kHz モード	4.7	—	ms	次の転送が開始可能になるまでに必要なバスの空き時間
			400 kHz モード	1.3	—	ms	
			1 MHz モード (1)	TBD	—	ms	
D102	Cb	バス容量性負荷		—	400	pF	

凡例: TBD = 未定

Note 1: 全ての I<sup>2</sup>C™ ピンの最大静電容量は 10 pF です。

- 2: 高速モードの I<sup>2</sup>C バスデバイスを標準モードの I<sup>2</sup>C バスシステムで使う事は可能ですが、その場合もパラメータ #107 ≥ 250 ns を満たす必要があります。デバイスが SCLx 信号の Low 期間を伸長しない場合、自動的にこの状態となります。SCLx 信号の Low 期間を伸長する場合、SCLx ラインを解放するパラメータ #102 + パラメータ #107 = 1000 + 250 = 1250 ns 前に (100 kHz モードの場合)、次のデータビットを SDAx ラインに出力する必要があります。

図 28-19: EUSARTx 同期送信 (マスタ/スレーブ) のタイミング

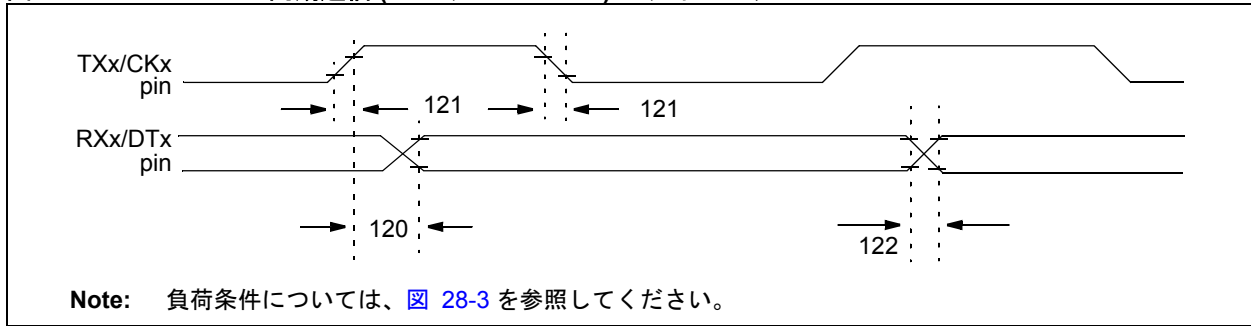


表 28-24: EUSARTx 同期送信の要件

パラメータ番号	記号	特性	最小値	最大値	単位	条件
120	TCKH2DTV	同期送信 (マスタとスレーブ) クロック High からデータ出力有効	—	40	ns	
121	TCKRF	クロック出力の立ち上がり / 立ち下がり時間 (マスタモード)	—	20	ns	
122	TDTRF	データ出力の立ち上がり / 立ち下がり時間	—	20	ns	

図 28-20: EUSARTx 同期受信 (マスタ/スレーブ) のタイミング

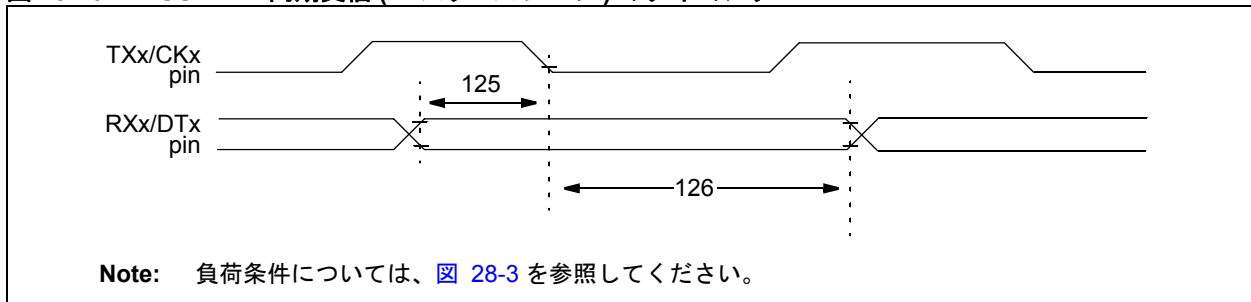


表 28-25: EUSARTx 同期受信の要件

パラメータ番号	記号	特性	最小値	最大値	単位	条件
125	TDTV2CKL	同期受信 (マスタとスレーブ) CKx ↓ 前のデータホールド (DTx ホールド時間)	10	—	ns	
126	TCKL2DTL	CKx ↓ 後のデータホールド (DTx ホールド時間)	15	—	ns	

# PIC18F97J60 ファミリ

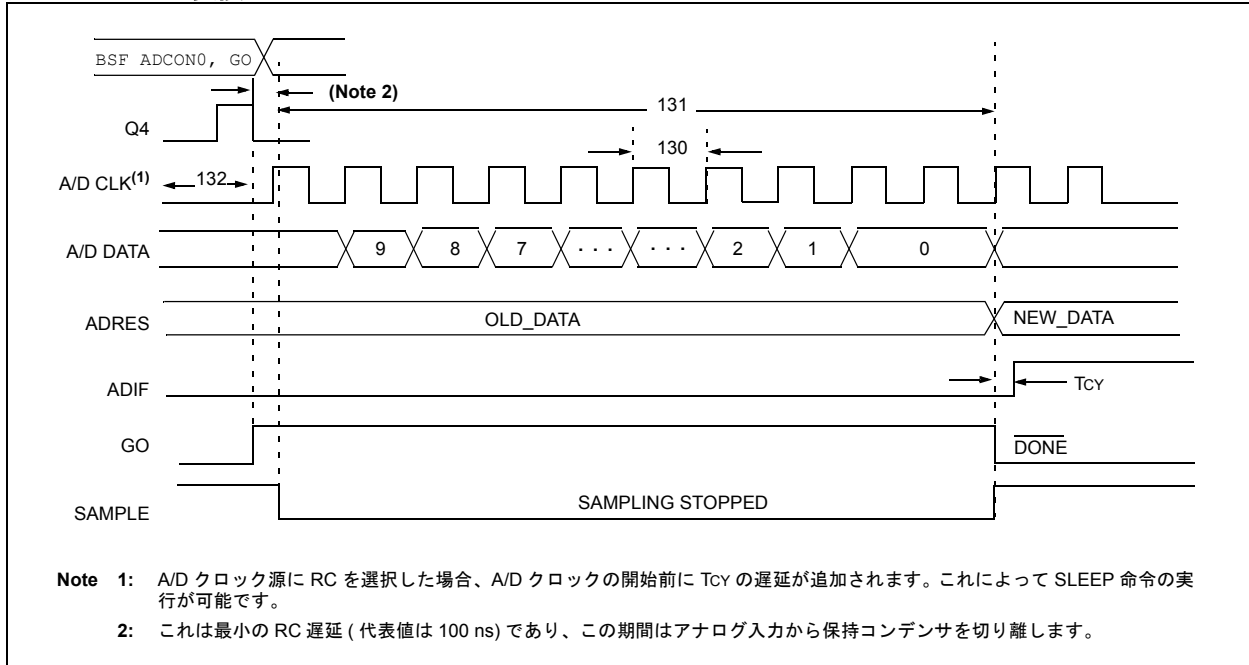
表 28-26: A/D コンバータの特性:

PIC18F97J60 ファミリ (産業用)

パラメータ番号	記号	特性	最小値	代表値	最大値	単位	条件
A01	NR	分解能	—	—	10	bit	$\Delta V_{REF} \geq 2.0 V$
A03	EIL	積分直線性誤差	—	—	$< \pm 1$	LSb	$\Delta V_{REF} \geq 2.0 V$
A04	EDL	微分直線性誤差	—	—	$< \pm 1$	LSb	$\Delta V_{REF} \geq 2.0 V$
A06	E0FF	オフセットエラー	—	—	$< \pm 3$	LSb	$\Delta V_{REF} \geq 2.0 V$
A07	E0N	ゲインエラー	—	—	$< \pm 3$	LSb	$\Delta V_{REF} \geq 2.0 V$
A10	-	単調性	保証 <sup>(1)</sup>			—	$V_{SS} \leq V_{AIN} \leq V_{REF}$
A20	$\Delta V_{REF}$	参照電圧レンジ ( $V_{REFH} - V_{REFL}$ )	1.8	—	—	V	$V_{DD} < 3.0 V$
			3	—	—	V	$V_{DD} \geq 3.0 V$
	$V_{REFSUM}$	参照電圧合計 ( $V_{REFH} + V_{REFL}$ )	—	—	$AV_{DD} + 0.5$	V	
A21	$V_{REFH}$	参照電圧 High	$V_{REFL}$	—	$AV_{DD}$	V	
A22	$V_{REFL}$	参照電圧 Low	$AV_{SS}$	—	$V_{REFH}$	V	
A25	$V_{AIN}$	アナログ入力電圧	$V_{REFL}$	—	$V_{REFH}$	V	
A30	$Z_{AIN}$	アナログ電圧源の 推奨インピーダンス	—	—	2.5	$k\Omega$	
A50	$I_{REF}$	$V_{REF}$ 入力電流 <sup>(2)</sup>	—	—	5	$\mu A$	$V_{AIN}$ のアキュイジション中 A/D 変換サイクル中
			—	—	1000	$\mu A$	

- Note 1:** A/D 変換結果は、入力電圧が増大しても減少する事はありません。また、ミッシングコードもありません。
- 2:**  $V_{REFH}$  電流は、いずれの  $V_{REFH}$  信号源を選択した場合も、 $RA3/AN3/V_{REF+}$  ピンまたは  $AV_{DD}$  から流れます。  
 $V_{REFL}$  電流は、いずれの  $V_{REFL}$  信号源を選択した場合も、 $RA2/AN2/V_{REF-}$  ピンまたは  $AV_{SS}$  から流れます。

図 28-21: A/D 変換のタイミング



# PIC18F97J60 ファミリ

表 28-27: A/D 変換の要件

パラメータ番号	記号	特性	最小値	最大値	単位	条件
130	TAD	A/D クロック周期	0.7	25.0 <sup>(1)</sup>	μs	TOSC ベース、VREF ≥ 2.0 V
			TBD	1	μs	A/D RC モード
131	T <sub>CNV</sub>	変換時間 (アキュジション時間を除く)(Note 2)	11	12	TAD	
132	T <sub>ACQ</sub>	アキュジション時間 (Note 3)	1.4	—	μs	-40 ~ +85 °C
135	T <sub>SWC</sub>	変換 → サンプリグの切り換え時間	—	(Note 4)		
TBD	T <sub>DIS</sub>	放電時間	0.2	—	μs	

凡例: TBD = 未定

- Note 1:** A/D クロック周期の時間は、デバイスの周波数と TAD クロック分周回路によって決まります。
- 2:** ADRES レジスタは次の T<sub>CY</sub> サイクルで読み出しが可能です。
- 3:** 変換後に電圧がフルスケールで変化 (V<sub>DD</sub> から V<sub>SS</sub> または V<sub>SS</sub> から V<sub>DD</sub>) した場合に保持コンデンサが「新しい」入力電圧を取得するのにかかる時間です。入力チャンネルのソース インピーダンス (R<sub>S</sub>) は 50 Ω です。
- 4:** デバイスクロックの次のサイクルで切り換わります。

## 28.5 Ethernet の仕様と要件

表 28-28: Ethernet トランシーバ外部磁気回路の要件

パラメータ	最小値	標準値	最大値	単位	条件
RX 巻線比	—	1:1	—	—	
TX 巻線比	—	1:1	—	—	トランス中央タップ = 3.3 V
挿入損失	—	—	-1.1	dB	
一次側インダクタンス	350	—	—	μH	8 mA バイアス
トランス絶縁	1.5	—	—	kVrms	IEEE 802.3™ 要件を満たすために必要
差動 - コモンモード除去比	40	—	—	dB	0.1 ~ 10 MHz
リターンロス	-16	—	—	dB	

# PIC18F97J60 ファミリ

---

NOTES:

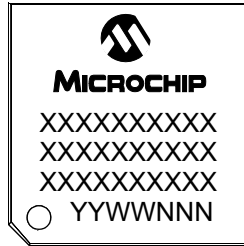


# PIC18F97J60 ファミリ

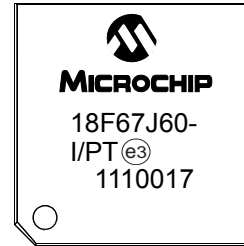
## 29.0 パッケージ情報

### 29.1 パッケージのマーキング情報

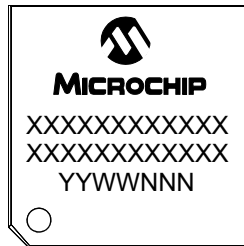
64-Lead TQFP



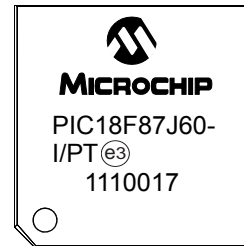
Example



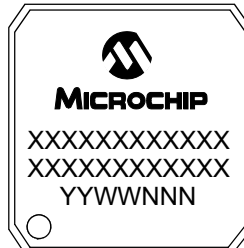
80-Lead TQFP



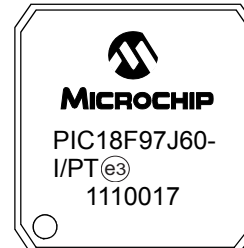
Example



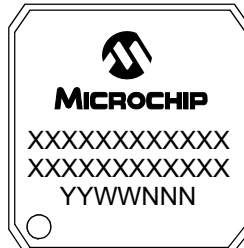
100-Lead TQFP (12x12x1 mm)



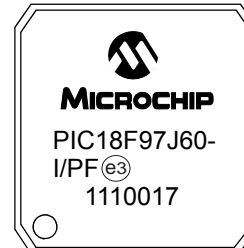
Example



100-Lead TQFP (14x14x1 mm)



Example



凡例: XX...Xお客様固有情報

Y 年コード (西暦の下1桁)

YY年コード (西暦の下2桁)

WW週コード (1月の第1週が「01」)

NNN英数字のトレーサビリティコード

<sup>e3</sup> 消し錫 (Sn) の使用を示す鉛フリーの JEDEC マーク

\* 本パッケージは鉛フリーです。鉛フリー JEDEC マーク ( <sup>e3</sup> ) は外箱に表記しています。

**Note:** マイクロチップ社の製品番号が1行に収まりきらない場合は複数行を使います。この場合お客様固有情報に使える文字数が制限されます。

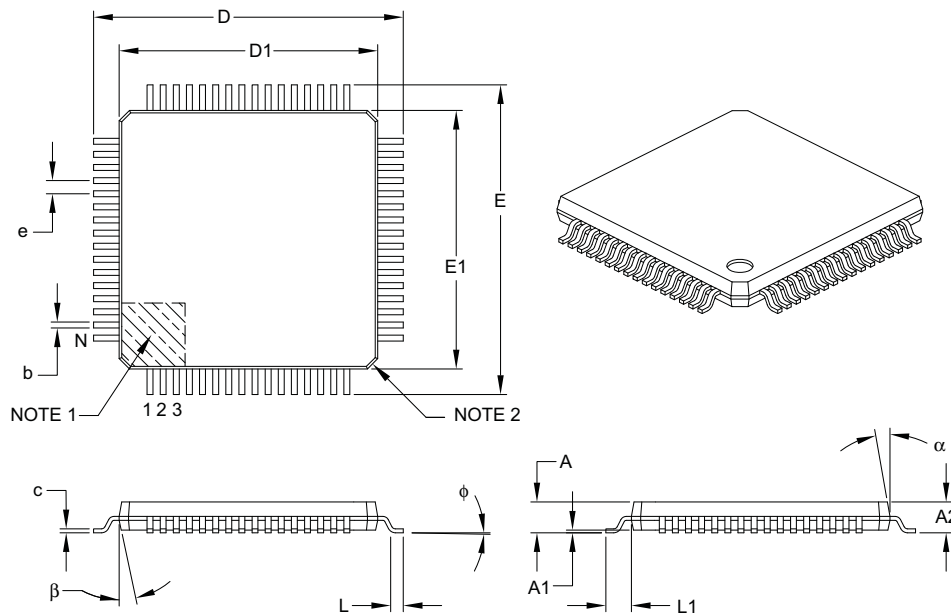
# PIC18F97J60 ファミリ

## 29.2 パッケージ詳細

以下に、各パッケージの技術的詳細を記載します。

### 64 ピン プラスチック薄型クワッドフラットパック (PT) - 10x10x1 mm ボディ、2.00 mm [TQFP]

**Note:** 最新のパッケージ図面については、以下のウェブサイトにある「Microchip Packaging Specification ( マイクロチップ社パッケージ仕様 )」を参照してください。  
<http://www.microchip.com/packaging>



寸法限界	単位	ミリメートル		
		最小	公称	最大
リード数	N	64		
リードピッチ	e	0.50 BSC		
全高	A	-	-	1.20
モールドパッケージ厚	A2	0.95	1.00	1.05
スタンドオフ	A1	0.05	-	0.15
足長	L	0.45	0.60	0.75
フットプリント	L1	1.00 REF		
足角	$\phi$	0°	3.5°	7°
全幅	E	12.00 BSC		
全長	D	12.00 BSC		
モールドパッケージ幅	E1	10.00 BSC		
モールドパッケージ長	D1	10.00 BSC		
リード厚	c	0.09	-	0.20
リード幅	b	0.17	0.22	0.27
モールドドラフト角トップ	$\alpha$	11°	12°	13°
モールドドラフト角ボトム	$\beta$	11°	12°	13°

**Notes:**

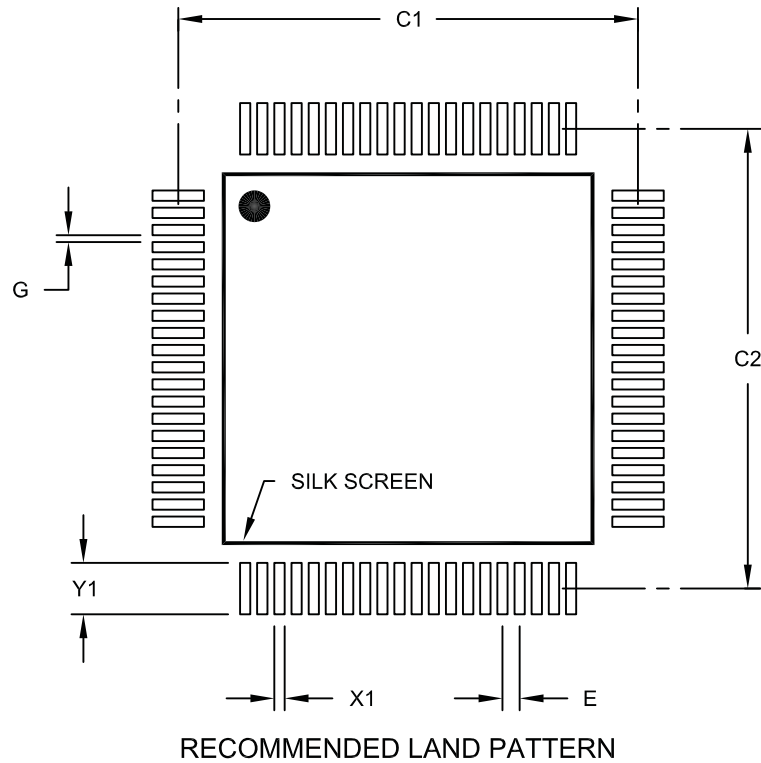
- 1 ピンマークの場所にはばらつきがありますが、必ず斜線部分内にあります。
- 2 コーナー部の面取りはオプションで、サイズは異なります。
- 3 D1 と E1 の寸法はモールドのはみ出しまたは突出部を含みません。モールドのはみ出しまたは突出部は側面から 0.25 mm を超えません。
- 4 寸法と公差は ASME Y14.5M に準拠しています。  
 BSC: 基本寸法、理論的に正確な値、公差なしで表示  
 REF: 参考寸法、通常は公差を含まない、情報としてのみ使われる値

Microchip Technology Drawing C04-085B

# PIC18F97J60 ファミリ

## 64ピンプラスチック薄型クワッドフラットパック (PT) - 10x10x1 mm ボディ、2.00 mm [TQFP]

**Note:** 最新のパッケージ図面については、以下のウェブサイトにある「Microchip Packaging Specification (マイクロチップ社パッケージ仕様)」を参照してください。  
<http://www.microchip.com/packaging>



寸法限界	単位	ミリメートル		
		最小	公称	最大
コンタクトピッチ	E	0.50 BSC		
コンタクトパッド間隔	C1		11.40	
コンタクトパッド間隔	C2		11.40	
コンタクトパッド幅 (X64)	X1			0.30
コンタクトパッド長 (X64)	Y1			1.50
パッド間距離	G	0.20		

**Notes:**

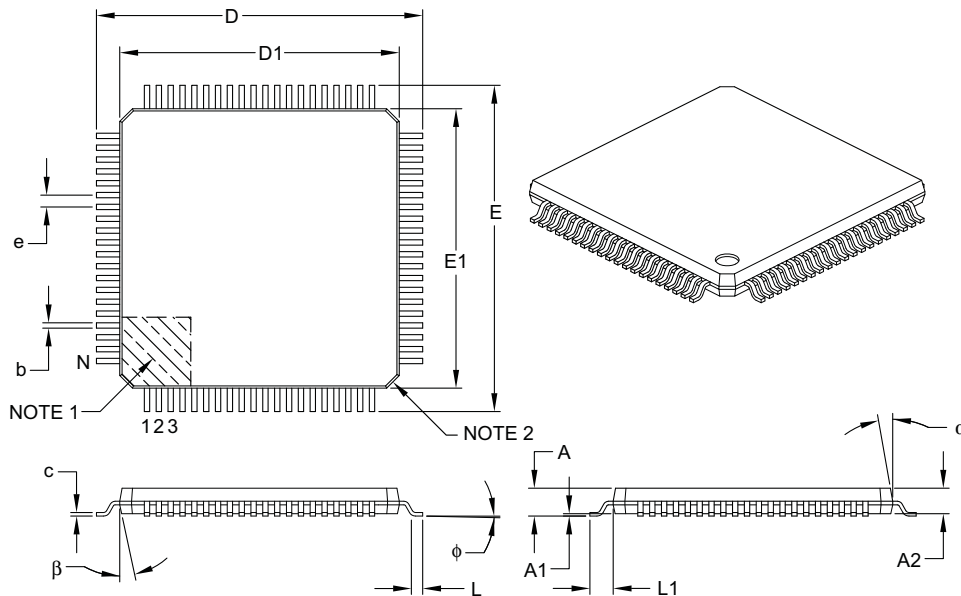
- 寸法と公差は ASME Y14.5M に準拠しています。  
 BSC: 基本寸法、理論的に正確な値、公差なしで表示

Microchip Technology Drawing No. C04-2085A

# PIC18F97J60 ファミリ

## 80 ピン プラスチック薄型クワッドフラットパック (PT) - 12x12x1 mm ボディ、2.00 mm [TQFP]

**Note:** 最新のパッケージ図面については、以下のウェブサイトにある「Microchip Packaging Specification ( マイクロチップ社パッケージ仕様 )」を参照してください。  
<http://www.microchip.com/packaging>



寸法限界	単位	ミリメートル		
		最小	公称	最大
リード数	N	80		
リードピッチ	e	0.50 BSC		
全高	A	-	-	1.20
モールドパッケージ厚	A2	0.95	1.00	1.05
スタンドオフ	A1	0.05	-	0.15
足長	L	0.45	0.60	0.75
フットプリント	L1	1.00 REF		
足角	$\phi$	0°	3.5°	7°
全幅	E	14.00 BSC		
全長	D	14.00 BSC		
モールドパッケージ幅	E1	12.00 BSC		
モールドパッケージ長	D1	12.00 BSC		
リード厚	c	0.09	-	0.20
リード幅	b	0.17	0.22	0.27
モールドドラフト角トップ	$\alpha$	11°	12°	13°
モールドドラフト角ボトム	$\beta$	11°	12°	13°

### Notes:

- 1 ピンマークの場所にはばらつきがありますが、必ず斜線部分内にあります。
- 2 コーナー部の面取りはオプションで、サイズは異なります。
- 3 D1 と E1 の寸法はモールドのはみ出しまたは突出部を含みません。モールドのはみ出しまたは突出部は側面から 0.25 mm を超えません。
4. 寸法と公差は ASME Y14.5M に準拠しています。

BSC: 基本寸法、理論的に正確な値、公差なしで表示

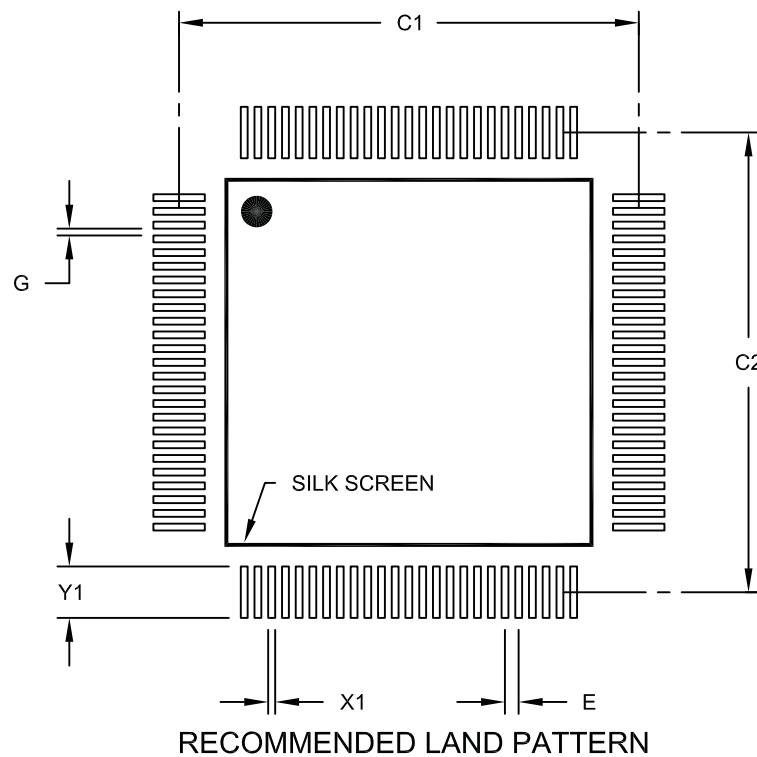
REF: 参考寸法、通常は公差を含まない、情報としてのみ使われる値

Microchip Technology Drawing C04-092B

# PIC18F97J60 ファミリ

## 80 ピン プラスチック薄型クワッドフラットパック (PT) - 12x12x1 mm ボディ、2.00 mm [TQFP]

**Note:** 最新のパッケージ図面については、以下のウェブサイトにある「Microchip Packaging Specification ( マイクロチップ社パッケージ仕様 )」を参照してください。  
<http://www.microchip.com/packaging>



寸法限界	単位	ミリメートル		
		最小	公称	最大
コンタクトピッチ	E	0.50 BSC		
コンタクトパッド間隔	C1		13.40	
コンタクトパッド間隔	C2		13.40	
コンタクトパッド幅 (X80)	X1			0.30
コンタクトパッド長 (X80)	Y1			1.50
パッド間距離	G	0.20		

**Notes:**

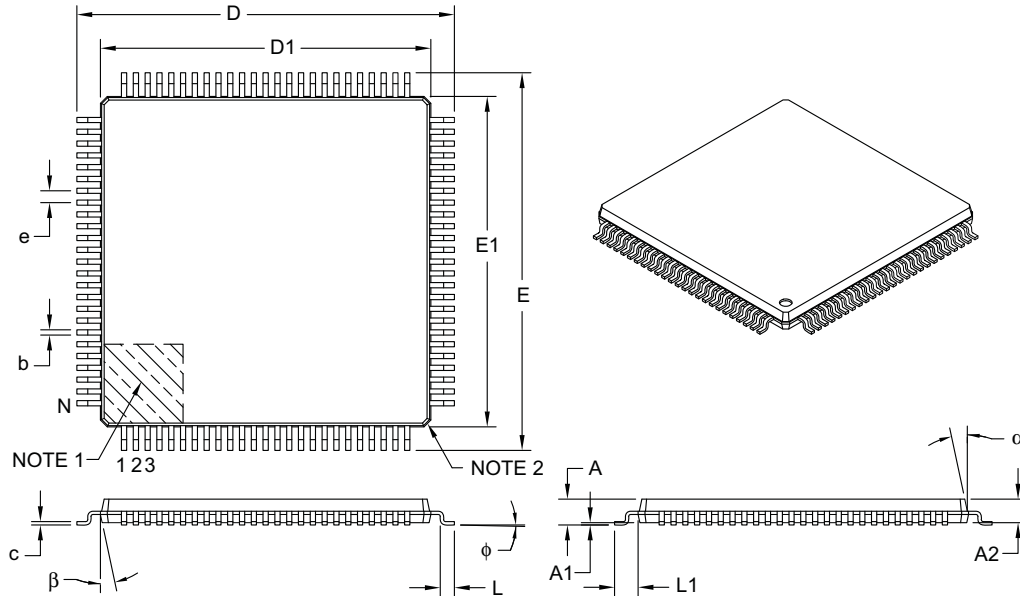
- 寸法と公差は ASME Y14.5M に準拠しています。  
 BSC: 基本寸法、理論的に正確な値、公差なしで表示

Microchip Technology Drawing No. C04-2092A

# PIC18F97J60 ファミリ

## 100ピンプラスチック薄型クワッドフラットパック (PT) - 12x12x1 mm ボディ、2.00 mm [TQFP]

**Note:** 最新のパッケージ図面については、以下のウェブサイトにある「Microchip Packaging Specification (マイクロチップ社パッケージ仕様)」を参照してください。  
<http://www.microchip.com/packaging>



寸法限界	単位	ミリメートル		
		最小	公称	最大
リード数	N	100		
リードピッチ	e	0.40 BSC		
全高	A	-	-	1.20
モールドパッケージ厚	A2	0.95	1.00	1.05
スタンドオフ	A1	0.05	-	0.15
足長	L	0.45	0.60	0.75
フットプリント	L1	1.00 REF		
足角	$\phi$	0°	3.5°	7°
全幅	E	14.00 BSC		
全長	D	14.00 BSC		
モールドパッケージ幅	E1	12.00 BSC		
モールドパッケージ長	D1	12.00 BSC		
リード厚	c	0.09	-	0.20
リード幅	b	0.13	0.18	0.23
モールドドラフト角トップ	$\alpha$	11°	12°	13°
モールドドラフト角ボトム	$\beta$	11°	12°	13°

### Notes:

- 1ピンマークの場所にはばらつきがありますが、必ず斜線部分内にあります。
- コーナー部の面取りはオプションで、サイズは異なります。
- D1とE1の寸法はモールドのはみ出しまたは突出部を含みません。モールドのはみ出しまたは突出部は側面から0.25 mmを超えません。
- 寸法と公差はASME Y14.5Mに準拠しています。

BSC: 基本寸法、理論的に正確な値、公差なしで表示

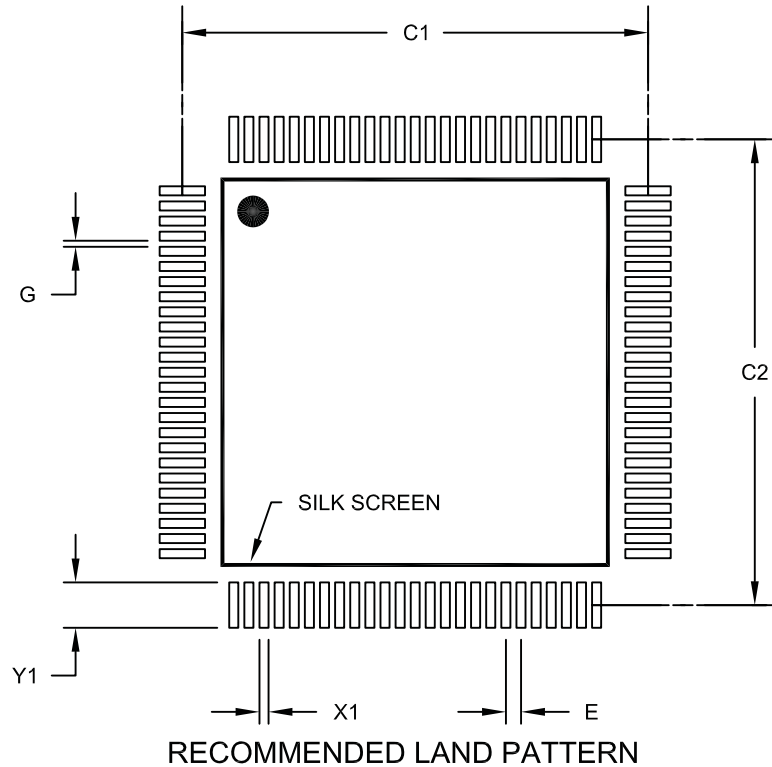
REF: 参考寸法、通常は公差を含まない、情報としてのみ使われる値

Microchip Technology Drawing C04-100B

# PIC18F97J60 ファミリ

## 100 ピン プラスチック薄型クワッドフラットパック (PT) - 12x12x1 mm ボディ、2.00 mm [TQFP]

**Note:** 最新のパッケージ図面については、以下のウェブサイトにある「Microchip Packaging Specification ( マイクロチップ社パッケージ仕様 )」を参照してください。  
<http://www.microchip.com/packaging>



寸法限界	単位	ミリメートル		
		最小	公称	最大
コンタクトピッチ	E	0.40 BSC		
コンタクトパッド間隔	C1		13.40	
コンタクトパッド間隔	C2		13.40	
コンタクトパッド幅 (X100)	X1			0.20
コンタクトパッド長 (X100)	Y1			1.50
パッド間距離	G	0.20		

**Notes:**

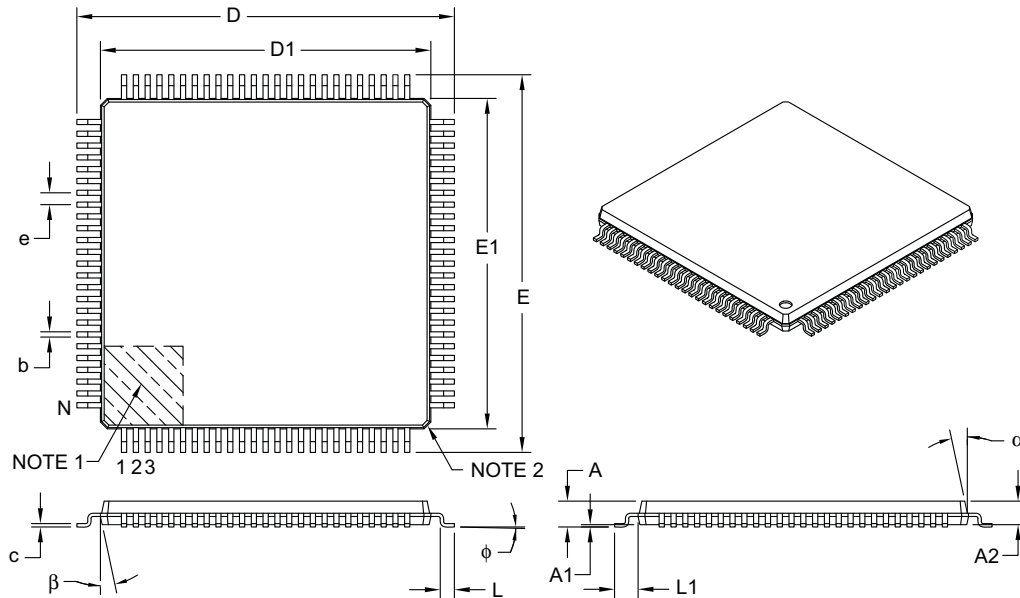
- 寸法と公差は ASME Y14.5M に準拠しています。  
 BSC: 基本寸法、理論的に正確な値、公差なしで表示

Microchip Technology Drawing No. C04-2100A

# PIC18F97J60 ファミリ

## 100ピンプラスチック薄型クワッドフラットパック (PF) - 14x14x1 mm ボディ、2.00 mm [TQFP]

**Note:** 最新のパッケージ図面については、以下のウェブサイトにある「Microchip Packaging Specification (マイクロチップ社パッケージ仕様)」を参照してください。  
<http://www.microchip.com/packaging>



寸法限界	単位	ミリメートル		
		最小	公称	最大
リード数	N	100		
リードピッチ	e	0.50 BSC		
全高	A	-	-	1.20
モールドパッケージ厚	A2	0.95	1.00	1.05
スタンドオフ	A1	0.05	-	0.15
足長	L	0.45	0.60	0.75
フットプリント	L1	1.00 REF		
足角	$\phi$	0°	3.5°	7°
全幅	E	16.00 BSC		
全長	D	16.00 BSC		
モールドパッケージ幅	E1	14.00 BSC		
モールドパッケージ長	D1	14.00 BSC		
リード厚	c	0.09	-	0.20
リード幅	b	0.17	0.22	0.27
モールドドラフト角トップ	$\alpha$	11°	12°	13°
モールドドラフト角ボトム	$\beta$	11°	12°	13°

**Notes:**

- 1 ピンマークの場所にはばらつきがありますが、必ず斜線部分内にあります。
- 2 コーナー部の面取りはオプションで、サイズは異なります。
- 3 D1 と E1 の寸法はモールドのはみ出しまたは突出部を含みません。モールドのはみ出しまたは突出部は側面から 0.25 mm を超えません。
- 4 寸法と公差は ASME Y14.5M に準拠しています。  
 BSC: 基本寸法、理論的に正確な値、公差なしで表示  
 REF: 参考寸法、通常は公差を含まない、情報としてのみ使われる値

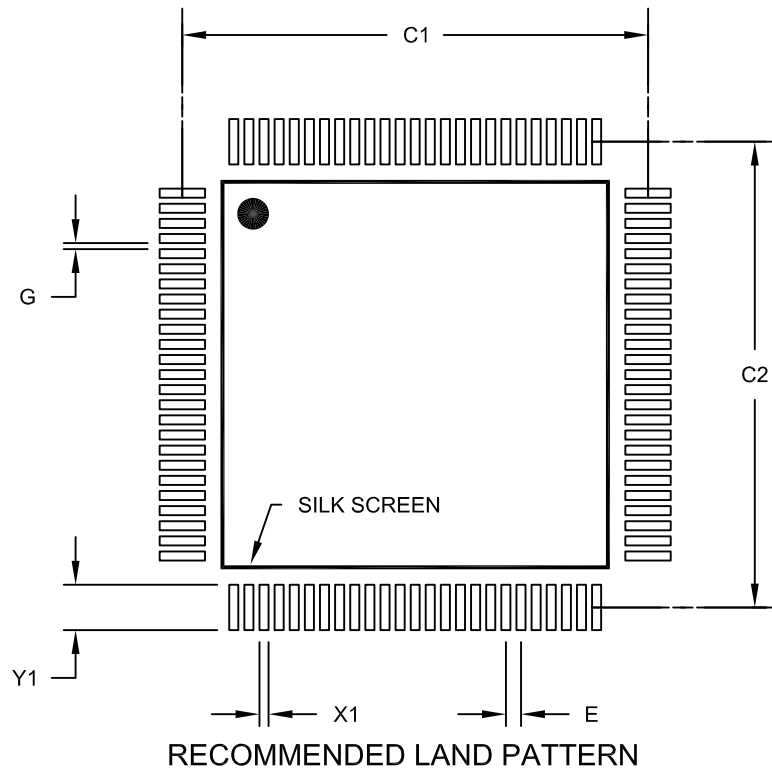
Microchip Technology Drawing C04-110B



# PIC18F97J60 ファミリ

## 100ピンプラスチック薄型クワッドフラットパック (PF) - 14x14x1 mm ボディ、2.00 mm [TQFP]

**Note:** 最新のパッケージ図面については、以下のウェブサイトにある「Microchip Packaging Specification (マイクロチップ社パッケージ仕様)」を参照してください。  
<http://www.microchip.com/packaging>



単位	ミリメートル			
	寸法限界	最小	公称	最大
コンタクトピッチ	E	0.50 BSC		
コンタクトパッド間隔	C1		15.40	
コンタクトパッド間隔	C2		15.40	
コンタクトパッド幅 (X100)	X1			0.30
コンタクトパッド長 (X100)	Y1			1.50
パッド間距離	G	0.20		

**Notes:**

- 寸法と公差は ASME Y14.5M に準拠しています。  
 BSC: 基本寸法、理論的に正確な値、公差なしで表示

Microchip Technology Drawing No. C04-2110A

# PIC18F97J60 ファミリ

---

NOTES:

## 補遺 A: 改訂履歴

### リビジョン A (2006 年 3 月)

PIC18F97J60 ファミリ データシートの初版リリース

### リビジョン B (2006 年 10 月)

最初の改訂、電氣的仕様 ( 暫定版 ) を追加、Ethernet モジュールに関する内容を改訂および更新、リセットの統合に関する内容を更新、デバイス メモリマップを更新

### リビジョン C (2007 年 6 月)

表 10.2 の入力電圧レベルを訂正、Ethernet モジュールのバッファ読み書きに関する内容を追加、「パッケージのマーキング情報」と「パッケージ詳細」のセクションに新しい 100 ピン PT 12x12x1 mm TQFP パッケージを追加、その他のパッケージ詳細図面を更新、製品識別システムの例を変更

### リビジョン D (2008 年 1 月)

「Ethernet 機能」の説明に 1 行追加、各パッケージにランドパターンの図面を追加

### リビジョン E (2009 年 10 月)

本更新により暫定版のステータスを解除

### リビジョン F (2011 年 4 月)

ブラウンアウト リセット (BOR) の仕様を追加、Ethernet RX 自動極性回路のセクションを追加、EMI フィルタのセクションを追加、**セクション 2.0 「PIC18FJ マイクロコントローラの使用にあたってのガイドライン」**を追加、PUSHL 命令のオペコード エンコードを 1110 1010 kkk kkkk に変更、**表 22-1** の 2 TOSC 最大デバイス周波数を 2.68 MHz から 2.86 MHz に訂正、コンパレータ入力オフセット電圧の最大値を 25 mV に訂正

# PIC18F97J60 ファミリ

## 補遺 B: デバイス間の違い

表 B-1 に、このデータシートに記載されているデバイス間の違いを示します。

表 B-1: PIC18F97J60 ファミリのデバイス間の違い

特長	PIC18F66J60	PIC18F66J65	PIC18F67J60	PIC18F86J60	PIC18F86J65	PIC18F87J60	PIC18F96J60	PIC18F96J65	PIC18F97J60
プログラムメモリ (バイト)	64K	96K	128K	64K	96K	128K	64K	96K	128K
プログラムメモリ (命令数)	32764	49148	65532	32764	49148	65532	32764	49148	65532
割り込み要因	26			27			29		
I/O ポート (ピン)	ポート A、B、C、D、E、 F、G (39)			ポート A、B、C、D、E、 F、G、H、J (55)			ポート A、B、C、D、E、 F、G、H、J (70)		
拡張 USART (EUSART) モジュール	1			2					
MSSP モジュール	1						2		
パラレルスレーブポート通信 (PSP)	なし						あり		
外部メモリバス	なし						あり		
10 ビット A/D コンバータモジュール	11 入力チャンネル			15 入力チャンネル			16 入力チャンネル		
パッケージ	64 ピン TQFP			80 ピン TQFP			100 ピン TQFP		

## 索引

### 数字

2 段階起動 .....	359, 370
2 ワード命令	
例 .....	85

### A

A/D .....	339
ADCAL ビット .....	347
ADCON0 レジスタ .....	339
ADCON1 レジスタ .....	339
ADCON2 レジスタ .....	339
ADRESH レジスタ .....	339, 342
ADRESL レジスタ .....	339
ECCP2 トリガの使用法 .....	346
アキュイジションの要件 .....	344
アナログポートピン、 コンフィグレーション .....	345
関連レジスタ .....	347
コンバータの校正 .....	347
コンバータの特性 .....	462
コンバータ割り込み、設定 .....	343
自動アキュイジション時間、選択と設定 .....	345
電力管理モードにおける動作 .....	347
特殊イベントトリガ (ECCP) .....	202, 346
変換 .....	346
変換クロック (TAD) .....	345
変換ステータス (GO/DONE ビット) .....	342
変換の要件 .....	463
モジュールの設定 .....	343
A/D コンバータ (ADC) → 「A/D」参照	
AC ( タイミング ) 特性 .....	443
温度 / 電圧仕様 .....	444
タイミングの条件 .....	444
デバイス タイミング仕様に対する 負荷条件 .....	444
パラメータの記号 .....	443
ACKSTAT .....	304
ACKSTAT ステータスフラグ .....	304
ADCAL ビット .....	347
ADCON0 レジスタ .....	339
GO/DONE ビット .....	342
ADCON1 レジスタ .....	339
ADCON2 レジスタ .....	339
ADDFSR .....	418
ADDLW .....	381
ADDULNK .....	418
ADDWF .....	381
ADDWFC .....	382
ADRESH レジスタ .....	339
ADRESL レジスタ .....	339, 342
ANDLW .....	382
ANDWF .....	383

### B

baud レート ジェネレータ .....	300
BC .....	383
BCF .....	384
BF .....	304
BF ステータスフラグ .....	304
BN .....	384
BNC .....	385
BNN .....	385
BNOV .....	386
BNZ .....	386
BOR → 「ブラウンアウト リセット」参照	
BOV .....	389
BRA .....	387
BRG → 「baud レート ジェネレータ」参照	
BSF .....	387
BTFSC .....	388
BTFSS .....	388
BTG .....	389
BZ .....	390

### C

CALL .....	390
CALLW .....	419
CLRF .....	391
CLRWDT .....	391
COMF .....	392
CPFSEQ .....	392
CPFSGT .....	393
CPFSLT .....	393
CPU の特殊機能 .....	359
C コンパイラ	
MPLAB C18 .....	426

### D

DAW .....	394
DCFSNZ .....	395
DC 特性 .....	439
電源電圧 .....	431
パワーダウン電流と消費電流 .....	432
DECF .....	394
DECFSZ .....	395

### E

ECCP2	
ピン割り当て .....	190
ENVREG ピン .....	369
Ethernet 動作、マイクロコントローラ	
クロック .....	51
Ethernet モジュール .....	217
CRC .....	248
EMI 放射に関する注意事項 .....	220
Ethernet 制御レジスタ .....	227

# PIC18F97J60 ファミリ

Ethernet モジュールとマイクロ コントローラのメモリの関係	222	バッファの構成	224
LED の設定	218	フロー制御	257
MAC および MII レジスタ	229	無効化	246
MAC の初期化設定	245	メモリマップ	234
PHSTAT レジスタ	232	リセット	267
PHY 起動タイマ	218	受信のみ	267
PHY の初期化設定	246	送信のみ	267
PHY レジスタ	232	マイクロコントローラの リセット	267
PHY レジスタからの読み出し	233	割り込み	239
RX 極性の自動検出、反転	220	割り込みと Wake-On-LAN	244
オシレータの要件	218	EUSART (Enhanced Universal Synchronous Asynchronous Receiver Transmitter) → 「EUSART」参照	
関連レジスタ、受信	255	EUSARTx	
関連レジスタ、送信	255	baud レート ジェネレータ	
関連レジスタ、ダイレクト メモリ アクセス (DMA) コントローラ	266	電力管理モードにおける動作	319
関連レジスタ、フロー制御	258	baud レート ジェネレータ (BRG)	319
個別パケット制御バイト	249	baud レート、非同期モード	321
受信フィルタ	259	関連レジスタ	320
Magic Packet	259	高 baud レート選択ビット (BRGH ビット)	319
パターンマッチ	259	サンプリング	319
ハッシュテーブル	259	自動 baud レート検出	323
ブロードキャスト	259	同期スレーブモード	337
マルチキャスト	259	関連レジスタ、受信	338
ユニキャスト	259	関連レジスタ、送信	337
初期化	245	受信	338
信号および電源インターフェイス	218	送信	337
全二重 / 半二重モードの設定と ネゴシエーション	256	同期マスタモード	333
ダイレクト メモリアクセス (DMA) コントローラ	265	関連レジスタ、受信	336
チェックサム計算	266	関連レジスタ、送信	334
メモリのコピー	265	受信	335
データの送受信	247	送信	333
受信ステータスベクタ	253	非同期モード	325
受信パケットの読み出し	253	9 ビットアドレス検出モードの設定	328
受信パケットのレイアウト	252	関連レジスタ、受信	329
受信バッファ空間	254	関連レジスタ、送信	327
送信ステータスベクタ	251	同期ブレイク文字による自動復帰	330
送信パケットのレイアウト	250	トランスミッタ	325
パケットの各フィールドの 定義	247 ~ 248	ブレイク文字のシーケンス	332
パケットの受信	252	受信	332
パケットの送信	249	レシーバ	328
特殊機能レジスタ (SFR)	227	<b>F</b>	
トランス、終端、その他の 外付け部品	219	FSCM → 「フェイルセーフ クロックモニタ」参照	
パケットのフォーマット	247	<b>G</b>	
バッファとバッファポインタ	223	GOTO	396
DMA アクセス	226	<b>I</b>	
受信バッファ	225	I/O ポート	145
送信バッファ	226	ピン機能	145
バッファアービタ	226		
バッファとレジスタ空間	222		

# PIC18F97J60 ファミリ

I <sup>2</sup> C モード (MSSP).....	279	IPR レジスタ .....	140
ACK シーケンスのタイミング .....	307	<b>L</b>	
baud レート ジェネレータ.....	300	LFSR .....	399
関連レジスタ .....	313	<b>M</b>	
クロック アービトレーション .....	301	MOVF .....	399
クロック ストレッチ .....	293	MOVFF .....	400
10 ビットスレーブ受信モード		MOVLB .....	400
(SEN = 1).....	293	MOVLW .....	401
10 ビットスレーブ送信モード .....	293	MOVSF .....	419
7 ビットスレーブ受信モード		MOVSS .....	420
(SEN = 1).....	293	MOVWF .....	401
7 ビットスレーブ送信モード .....	293	MPLAB ASM30 アセンブラ、リンカ、	
クロック同期と CKP ビット.....	294	ライブラリアン.....	426
クロックレートと BRG .....	300	MPLAB PM3 デバイス プログラマ.....	428
ジェネラルコール アドレスの		MPLAB REAL ICE インサーキット	
サポート.....	297	エミュレータ システム.....	427
シリアルクロック (SCKx/SCLx).....	286	MPLAB 統合開発環境ソフトウェア .....	425
ストップ条件のタイミング .....	307	MPLINK オブジェクト リンカ /MPLIB	
スリープ動作 .....	308	オブジェクトライブラリアン .....	426
スレーブモード.....	284	MSSP	
アドレスマスク .....	285	ACK パルス .....	284, 286
アドレッシング .....	284	SPI マスタ /スレーブの接続.....	273
受信 .....	286	SSPxBUF レジスタ .....	274
送信 .....	286	SSPxSR レジスタ .....	274
動作.....	284	制御レジスタ (汎用).....	269
バスコリジョン		モジュールの概要 .....	269
スタート条件中 .....	309	MULLW .....	402
ストップ条件中 .....	312	MULWF .....	402
反復スタート条件中 .....	311	<b>N</b>	
マスタモード .....	298	NEGF .....	403
baud レート ジェネレータ .....	300	NOP .....	403
受信 .....	304	<b>O</b>	
スタート条件のタイミング .....	302	OUI → 「ベンダ識別子」 参照	
送信 .....	304	<b>P</b>	
動作 .....	299	PIE レジスタ .....	137
反復スタート条件のタイミング .....	303	PIR レジスタ .....	134
マルチマスタ モード .....	308	PLL ブロック .....	51
マルチマスタ通信、バスコリジョン、		各種設定とクロック速度.....	52
バス アービトレーション.....	308	POP .....	404
リセットの影響.....	308	POR → 「パワーオン リセット」 参照	
レジスタ .....	279	PORTA	
読み出し / 書き込みビット		LATA レジスタ .....	146
情報 (R/W ビット).....	284, 286	PORTA レジスタ .....	146
INCF .....	396	TRISA レジスタ .....	146
INCFSZ .....	397	関連レジスタ .....	147
INFSNZ .....	397	PORTB	
Initialization Conditions		LATB レジスタ .....	148
for All Registers .....	69 ~ 75	PORTB レジスタ .....	148
INTCON レジスタ .....	131		
RBIF ビット .....	148		
Inter-Integrated Circuit → 「I <sup>2</sup> C モード」 参照			
INTRC → 「内部オシレータ ブロック」 参照			
IORLW .....	398		
IORWF .....	398		

# PIC18F97J60 ファミリ

RB7:RB4 状態変化割り込みフラグ (RBIF ビット) .....	148	PWM (CCP モジュール)	
TRISB レジスタ .....	148	TMR2/PR2 一致 .....	203
関連レジスタ .....	150	TMR4/PR4 一致 .....	187
PORTC		関連レジスタ .....	196
LATC レジスタ .....	151	周期 .....	194
PORTC レジスタ .....	151	周波数 / 分解能の例 .....	195
RC3/SCK1/SCL1 ピン .....	286	デューティ サイクル .....	194
TRISC レジスタ .....	151	動作の設定 .....	195
関連レジスタ .....	153	PWM (ECCP モジュール) .....	203
PORTD		CCPR1H:CCPR1L レジスタ .....	203
LATD レジスタ .....	154	PWM 動作の設定 .....	214
PORTD レジスタ .....	154	拡張 PWM の自動シャットダウン .....	211
TRISD レジスタ .....	154	起動に関する注意点 .....	213
関連レジスタ .....	156	周期 .....	203
PORTE		周波数 / 分解能の例 .....	204
LATE レジスタ .....	157	出力関係 (アクティブ High) .....	205
PORTE レジスタ .....	157	出力関係 (アクティブ Low) .....	206
PSP モード選択 (PSPMODE ビット) ..	168	出力コンフィグレーション .....	204
RE0/AD8/RD/P2D ピン .....	168	デューティ サイクル .....	204
RE1/AD9/WR/P2C ピン .....	168	ハーフブリッジモード .....	207
RE2/AD10/CS/P2B ピン .....	168	フルブリッジモード .....	208
TRISE レジスタ .....	157	フルブリッジ出力モードにおける 方向の変更 .....	209
関連レジスタ .....	159	プログラマブルなデッドバンド遅延 .....	211
PORTF		リセットの影響 .....	214
LATF レジスタ .....	160		
PORTF レジスタ .....	160	<b>Q</b>	
TRISF レジスタ .....	160	Q クロック .....	195, 204
関連レジスタ .....	161		
PORTG		<b>R</b>	
LATG レジスタ .....	162	RAM → 「データメモリ」参照	
PORTG レジスタ .....	162	RCALL .....	405
TRISG レジスタ .....	162	RC_IDLE モード .....	61
関連レジスタ .....	163	RCON レジスタ	
PORTH		初期化時のビット状態 .....	68
LATH レジスタ .....	164	RC_RUN モード .....	58
PORTH レジスタ .....	164	RESET .....	405
TRISH レジスタ .....	164	RETFIE .....	406
関連レジスタ .....	165	RETLW .....	406
PORTJ		RETURN .....	407
LATJ レジスタ .....	166	RLCF .....	407
PORTJ レジスタ .....	166	RLNCF .....	408
TRISJ レジスタ .....	166	RRCF .....	408
関連レジスタ .....	167	RRNCF .....	409
PRI_IDLE モード .....	60		
PRI_RUN モード .....	56	<b>S</b>	
PSP → 「パラレルスレーブ ポート」参照		SCKx .....	269
PUSH .....	404	SDIx .....	269
PUSHL .....	420	SDOx .....	269
PUSH および POP 命令 .....	82	SEC_IDLE モード .....	60
		SEC_RUN モード .....	56
		SETF .....	409
		SLEEP .....	410



# PIC18F97J60 ファミリ

SPI モード (MSSP)		TMR1H レジスタ .....	175
SPI I/O の有効化 .....	273	TMR1L レジスタ .....	175
SPI クロック .....	274	オーバーフロー割り込み .....	175
関連レジスタ .....	278	オシレータ .....	175, 177
クロック速度とモジュールの関係 .....	277	レイアウトに関する考慮事項 .....	177
シリアルクロック .....	269	関連レジスタ .....	179
シリアルデータ出力 .....	269	クロック源としての使用法 .....	177
シリアルデータ入力 .....	269	動作 .....	176
スレーブセレクト .....	269	特殊イベントトリガ (ECCP) .....	202
スレーブセレクトの同期 .....	275	非同期カウンタモードに関する	
スレーブモード .....	275	考慮事項 .....	178
代表的な接続 .....	273	リアルタイム クロックとしての	
電力管理モードにおける動作 .....	277	使用法 .....	178
動作 .....	272	割り込み .....	178
バスモード互換性 .....	277	Timer2 .....	180
マスタ / スレーブの接続 .....	273	PR2 レジスタ .....	194, 203
マスタモード .....	274	TMR2/PR2 一致割り込み .....	203
リセットの影響 .....	277	関連レジスタ .....	181
SSPOV .....	304	出力 .....	181
SSPOV ステータスフラグ .....	304	動作 .....	180
SSPSTAT レジスタ		割り込み .....	181
R/W ビット .....	286	Timer3 .....	183
SSPxSTAT レジスタ		16 ビット読み書きモード .....	185
R/W ビット .....	284	ECCPx 特殊イベントトリガによる	
SSx .....	269	リセット .....	185
SUBFSR .....	421	TMR3H レジスタ .....	183
SUBFWB .....	410	TMR3L レジスタ .....	183
SUBLW .....	411	オーバーフロー割り込み .....	183, 185
SUBLNK .....	421	オシレータ .....	183, 185
SUBWF .....	411	関連レジスタ .....	185
SUBWFB .....	412	動作 .....	184
SWAPF .....	412	Timer4 .....	187
<b>T</b>		PR4 レジスタ .....	187, 194
TBLRD .....	413	TMR4/PR4 一致割り込み .....	187, 188
TBLWT .....	414	TMR4 レジスタ .....	187
Timer0 .....	171	関連レジスタ .....	188
16 ビットモードにおける読み書き .....	172	出力、PWM タイムベース .....	188
オーバーフロー割り込み .....	173	動作 .....	187
関連レジスタ .....	173	プリスケアラ →「プリスケアラ、Timer4」参照	
クロック源選択 (T0CS ビット) .....	172	ポストスケアラ →	
信号源エッジ選択 (T0SE ビット) .....	172	「ポストスケアラ、Timer4」参照	
動作 .....	172	Top-of-Stack へのアクセス .....	81
プリスケアラ .....	173	TRISE レジスタ	
プリスケアラ →「プリスケアラ、Timer0」参照		PSPMODE ビット .....	168
プリスケアラ選択		TSTFSZ .....	415
(T0PS2:T0PS0 ビット) .....	173	TXSTAx レジスタ	
プリスケアラの割り当て		BRGH ビット .....	319
(PSA ビット) .....	173	<b>V</b>	
プリスケアラ、割り当ての切り換え .....	173	VDDCORE/VCAP ピン .....	369
Timer1 .....	175	<b>W</b>	
16 ビット読み書きモード .....	177	WCOL .....	302, 303, 304, 307
ECCPx 特殊イベントトリガによる		WCOL ステータスフラグ .....	302, 303, 304, 307
リセット .....	178		

# PIC18F97J60 ファミリ

WWW アドレス .....	491	外部クロック入力 (EC モード) .....	50
WWW、オンラインサポート .....	9	外部メモリバス .....	115
<b>X</b>		16 ビットデータ幅モード .....	118
XORLW .....	415	16 ビットバイト書き込みモード .....	119
XORWF .....	416	16 ビットバイト選択モード .....	121
<b>あ</b>		16 ビットモードのタイミング .....	122
アセンブラ		16 ビットワード書き込みモード .....	120
MPASM アセンブラ .....	426	21 ビットアドレッシング .....	117
<b>い</b>		8 ビットデータ幅モード .....	123
インサーキット シリアル プログラミング		8 ビットモードのタイミング .....	124
(ICSP) .....	359, 373	I/O ポートの機能 .....	115
インサーキット デバッグ .....	373	アドレスシフト .....	117
インターネット アドレス .....	491	アドレス幅とデータ幅 .....	117
インデックス付きリテラル オフセット		アドレスラインとデータラインの	
アドレッシング		使用 (表) .....	117
と標準 PIC18 命令 .....	422	ウェイトステート .....	118
インデックス付きリテラル		制御 .....	116
オフセットモード .....	422	電力管理モードにおける動作 .....	125
<b>う</b>		プログラムメモリ モード .....	118
ウォッチドッグ タイマ (WDT) .....	359, 367	拡張マイクロコントローラ .....	118
関連レジスタ .....	368	マイクロコントローラ .....	118
制御レジスタ .....	367	ポートピンの弱プルアップ .....	118
プログラミングの注意事項 .....	367	拡張キャプチャ / コンペア / PWM (ECCP) .....	197
<b>え</b>		ECCP1/ECCP3 による	
エラッタ .....	9	CCP4/CCP5 の使用 .....	199
<b>お</b>		ECCP1/ECCP3 の出力と	
お客様アンケート .....	492	プログラムメモリ モード .....	199
お客様向け通知サービス .....	491	ECCP1 のピン設定 .....	200
お客様向け変更通知サービス .....	491	ECCP2 の出力とプログラム	
オシレータ起動タイマ (OST) .....	54	メモリ モード .....	199
オシレータの移行 .....	54	ECCP2 のピン設定 .....	200
オシレータの設定 .....	49	ECCP3 のピン設定 .....	201
EC .....	49	PWM モード →	
ECPLL .....	49	「PWM (ECCP モジュール)」参照	
HS .....	49	拡張 PWM モード .....	203
HSPLL .....	49	キャプチャ / コンペアモード .....	202
INTRC .....	49	キャプチャモード →	
内部オシレータ ブロック .....	51	「キャプチャ (ECCP モジュール)」参照	
オシレータの選択 .....	359	出力とコンフィグレーション .....	199
オシレータ、Timer1 .....	175, 185	タイマリソース .....	199
オシレータ、Timer3 .....	183	標準 PWM モード .....	202
オペコード フィールドの説明 .....	376	拡張キャプチャ / コンペア / PWM (ECCPx)	
<b>か</b>		関連レジスタ .....	215
改訂履歴 .....	475	拡張命令セット	
開発サポート .....	425	ADDFSR .....	418
		ADDULNK .....	418
		CALLW .....	419
		MOVSF .....	419
		MOVSS .....	420
		PUSHL .....	420
		SUBFSR .....	421
		SUBULNK .....	421
		カスタマサポート .....	491
		間接アドレッシング .....	99

# PIC18F97J60 ファミリ

## き

既定値のシステムクロック .....	54
キャプチャ (CCP モジュール) .....	191
CCPRxH:CCPRxL レジスタ .....	191
CCPx ピンの設定 .....	191
Timer1/Timer3 モード選択 .....	191
関連レジスタ .....	193
ソフトウェア割り込み .....	191
プリスケアラ .....	191
キャプチャ (ECCP モジュール) .....	202
キャプチャ / コンペア / PWM (CCP) .....	189
CCPRxH レジスタ .....	190
CCPRxL レジスタ .....	190
CCPx/ECCPx 相互接続設定 .....	190
CCPx/ECCPx モードと タイマリソース .....	190
キャプチャモード → 「キャプチャ」 参照	
コンペアモード → 「コンペア」 参照	
モジュール設定 .....	190

## く

クロック源	
オシレータの切り換え .....	52
電力管理モードの影響 .....	54
リセット時の既定値の システムクロック .....	54

## け

計算型 GOTO .....	83
計算式	
受信バッファの空き領域の計算 .....	254
ランダムアクセスの場合の アドレス計算 .....	253

## こ

コアの特長	
オシレータの選択肢 .....	11
外部メモリバス .....	11
拡張命令セット .....	11
メモリ空間の拡張 .....	11
容易なデバイス移行 .....	11
高速レジスタスタック .....	83
コード保護 .....	359
コンパレータ .....	349
アナログ入力接続に関する注意事項 .....	353
応答時間 .....	351
関連レジスタ .....	353
参照電圧 .....	351
外部信号 .....	351
内部信号 .....	351
出力 .....	351
スリープ中の動作 .....	352
設定 .....	350

動作 .....	351
リセットの影響 .....	352
割り込み .....	352
コンパレータ参照電圧 .....	355
関連レジスタ .....	357
スリープ中の動作 .....	356
精度と誤差 .....	356
接続に関する注意事項 .....	356
設定 .....	355
リセットの影響 .....	356
コンパレータの仕様 .....	442
コンフィグレーション ビット .....	359
コンフィグレーション レジスタの保護 .....	373
コンフィグレーション不一致 (CM) リセット .....	65
コンペア (CCP モジュール) .....	192
CCPRx レジスタ .....	192
CCPx ピンの設定 .....	192
Timer1/Timer3 モード選択 .....	192
関連レジスタ .....	193
ソフトウェア割り込みモード .....	192
コンペア (ECCP モジュール) .....	202
特殊イベントトリガ .....	202, 346

## さ

参照電圧仕様 .....	442
サンプルコード	
16 x 16 符号付き乗算のルーチン .....	128
16 x 16 符号なし乗算のルーチン .....	128
8 x 8 符号付き乗算のルーチン .....	127
8 x 8 符号なし乗算のルーチン .....	127
PORTA の初期化 .....	146
PORTB の初期化 .....	148
PORTC の初期化 .....	151
PORTD の初期化 .....	154
PORTE の初期化 .....	157
PORTF の初期化 .....	160
PORTG の初期化 .....	162
PORTH の初期化 .....	164
PORTJ の初期化 .....	166
SSP1BUF (SSP1SR) レジスタへの 読み込み .....	272
STATUS、WREG、BSR レジスタの RAM への保存 .....	144
Timer1 割り込みサービスによる リアルタイムクロックの実装 .....	179
オフセット値を使う計算型 GOTO .....	83
間接アドレッシングによる RAM (バンク 1) のクリア .....	98
キャプチャ プリスケアラの切り換え .....	191
高速レジスタスタック .....	83
フラッシュ プログラムメモリ ワードの読み出し .....	109
フラッシュ プログラムメモリの行消去 .....	110

# PIC18F97J60 ファミリ

フラッシュ プログラムメモリへの書き込み .....	112	I <sup>2</sup> C スレーブモード (10 ビット送信) .....	292
<b>し</b>		I <sup>2</sup> C スレーブモード (7 ビット受信、SEN = 0) .....	287
<b>式</b>		I <sup>2</sup> C スレーブモード (7 ビット受信、SEN = 1) .....	295
16 x 16 符号付き乗算のアルゴリズム .....	128	I <sup>2</sup> C スレーブモード (7 ビット送信) .....	289
16 x 16 符号なし乗算のアルゴリズム .....	128	I <sup>2</sup> C バス スタート / ストップビット .....	457
A/D アクイジション時間 .....	344	I <sup>2</sup> C バスデータ .....	457
baud レート誤差の計算 .....	320	I <sup>2</sup> C マスタモード (7 ビット受信) .....	306
最小 A/D 充電時間 .....	344	I <sup>2</sup> C マスタモード (7 または 10 ビット送信) .....	305
必要な最小 A/D アクイジション時間の計算 .....	344	PWM 出力 .....	194
受信フィルタ		PWM の自動シャットダウン (P1RSEN = 0、自動再起動を無効にしている場合) .....	213
AND ロジックフロー .....	262	PWM の自動シャットダウン (P1RSEN = 1、自動再起動を有効にしている場合) .....	213
Magic Packet フォーマット .....	264	PWM の方向変更 .....	210
OR ロジックフロー .....	261	RC_RUN モードから PRI_RUN モードへの遷移 .....	58
パターンマッチ フィルタのフォーマット .....	263	RC_RUN モードへの遷移 .....	58
シリアル ペリフェラル インターフェイス → 「SPI モード」参照		SEC_RUN モードから PRI_RUN モード (HSPLL) への遷移 .....	57
シリアルクロック、SCKx .....	269	SEC_RUN モードへの移行時の遷移 .....	57
シリアルデータ出力 (SDOx) .....	269	SPI スレーブモードの例 (CKE = 0) .....	455
シリアルデータ入力 (SDIx) .....	269	SPI スレーブモードの例 (CKE = 1) .....	456
<b>す</b>		SPI マスタモードの例 (CKE = 0) .....	453
水晶振動子 / セラミック振動子 (HS モード) .....	50	SPI マスタモードの例 (CKE = 1) .....	454
スリープ時		SPI モード (スレーブモード、CKE = 0) .....	276
OSC1 および OSC2 ピンの状態 .....	54	SPI モード (スレーブモード、CKE = 1) .....	276
スレーブセレクト (SSx) .....	269	SPI モード (マスタモード) .....	274
<b>せ</b>		TBLRD の外部メモリバス (拡張マイクロコントローラ モード) .....	122, 124
絶対最大定格 .....	429	Timer0 と Timer1 の外部クロック .....	451
<b>そ</b>		アイドルからランモードへの復帰時の遷移 .....	60
ソフトウェア シミュレータ (MPLAB SIM) .....	427	アイドルへの移行時の遷移 .....	60
<b>た</b>		外部クロック (PLL を除く全モード) .....	445
タイミング チャート		キャプチャ / コンペア / PWM (ECCPx モジュールを含む) .....	452
2 段階起動 (INTRC から HSPLL への遷移) .....	370	クロック / 命令サイクル .....	84
A/D 変換 .....	462	クロックの同期 .....	294
baud レート ジェネレータとクロック アービトレーション .....	301	最初のスタートビット .....	302
baud レートの自動計算 .....	324	受信または送信モードの I <sup>2</sup> C ストップ条件 .....	307
BRG のオーバーフロー シーケンス .....	324	スタート条件中の I <sup>2</sup> C バスコリジョン (SCLx = 0) .....	310
CLKO と I/O .....	447	スタート条件中の I <sup>2</sup> C バスコリジョン (SDAx のみ) .....	309
EUSARTx 同期受信 (マスタ / スレーブ) .....	461	スタート条件中の SDAx アービトレーションによる BRG リセット .....	310
EUSARTx 同期送信 (マスタ / スレーブ) .....	461	ストップ条件中の I <sup>2</sup> C バスコリジョン (ケース 1) .....	312
I <sup>2</sup> C ACK シーケンス .....	307		
I <sup>2</sup> C スレーブモード (10 ビット受信、SEN = 0) .....	290		
I <sup>2</sup> C スレーブモード (10 ビット受信、SEN = 1) .....	296		

# PIC18F97J60 ファミリ

ストップ条件中の I <sup>2</sup> C バスコリジョン (ケース 2).....	312
スリープからの復帰時の遷移 (HSPLL).....	59
スリープ時の自動復帰ビット (WUE).....	331
スリープ時の外部メモリバス (拡張マイクロ コントローラモード).....	122, 124
スリープへの移行時の遷移.....	59
スレーブの同期.....	275
送信と ACK の I <sup>2</sup> C バスコリジョン.....	308
通常動作時の自動復帰ビット (WUE).....	331
デューティ サイクルが 100% に近い 状態での PWM の方向変更.....	210
同期受信 (マスタモード、SREN).....	335
同期送信.....	333
同期送信 (TXEN を利用).....	334
ハーフブリッジ PWM 出力.....	207
パラレルスレーブ ポート (PSP) 書き込み.....	169
パラレルスレーブ ポート (PSP) 読み出し.....	170
反復スタート条件.....	303
反復スタート条件中の I <sup>2</sup> C バスコリジョン (ケース 1).....	311
反復スタート条件中の I <sup>2</sup> C バスコリジョン (ケース 2).....	311
非同期受信、RXDTP = 0 (RXx を反転しない).....	329
非同期送信 (連続)、TXCKP = 0 (TXx を反転しない).....	326
非同期送信、TXCKP = 0 (TXx を反転しない).....	326
フェイルセーフ クロック モニタ.....	372
フルブリッジ PWM 出力.....	208
ブレーク文字シーケンスの送信.....	332
プログラムメモリ書き込み.....	449
プログラムメモリ読み出し.....	448
マスタ SSP I <sup>2</sup> C バス スタート / ストップビット.....	459
マスタ SSP I <sup>2</sup> C バス データ.....	459
リセット、ウォッチドッグ タイマ (WDT)、 オシレータ 起動 タイマ (OST)、 パワーアップ タイマ (PWRT).....	450
I <sup>2</sup> C スレーブモード (10 ビット受信、SEN = 0、 ADMSK = 01001).....	291
I <sup>2</sup> C スレーブモード (7 ビット受信、SEN = 0、 ADMSK = 01011).....	288
I <sup>2</sup> C スレーブモードのジェネラルコール アドレス シーケンス (7 または 10 ビット アドレス モード).....	297
パワーアップ後のタイムアウト シーケンス (MCLR を VDD に接続しない)、例 1.....	66
パワーアップ後のタイムアウト シーケンス (MCLR を VDD に接続しない)、例 2.....	67

パワーアップ後のタイムアウト シーケンス (MCLR を VDD に接続、 VDD の立ち上がり時間 < TPWRT).....	66
立ち上がり時間が遅い場合 (MCLR を VDD に接続、 VDD 立ち上がり時間 > TPWRT).....	67
タイミング チャートと仕様	
AC 特性	
内部 RC 精度.....	446
CLKO と I/O の要件.....	447, 448
EUSARTx 同期受信の要件.....	461
EUSARTx 同期送信の要件.....	461
I <sup>2</sup> C バス スタート / ストップビットの要件 (スレーブモード).....	457
I <sup>2</sup> C バス データ要件 (スレーブモード).....	458
PLL クロック.....	446
SPI スレーブモードの要件の例 (CKE = 1).....	456
SPI モードの要件の例 (スレーブモード、CKE=0).....	455
SPI モードの要件の例 (マスタモード、CKE=0).....	453
SPI モードの要件の例 (マスタモード、CKE=1).....	454
Timer0 と Timer1 の外部クロックの要件.....	451
外部クロックの要件.....	445
キャプチャ / コンペア / PWM の要件 (ECCPx モジュールを含む).....	452
パラレルスレーブ ポートの要件.....	452
プログラムメモリ書き込み要件.....	449
マスタ SSP I <sup>2</sup> C バス スタート / ストップビットの要件.....	459
マスタ SSP I <sup>2</sup> C バス データの要件.....	460
リセット、ウォッチドッグ タイマ、オシレータ 起動タイマ、パワーアップタイマ、ブラウン アウト リセットの要件.....	450

## ち

直接アドレッシング.....	99
----------------	----

## て

データ アドレッシング モード.....	98
インデックス付きリテラル オフセット.....	101
BSR.....	103
アクセスバンクのマッピング.....	103
影響を受ける命令.....	101
拡張命令セットが有効な場合の アドレッシング モードの比較.....	102
含意およびリテラル.....	98
間接.....	98
直接.....	98

# PIC18F97J60 ファミリ

データメモリ .....	86	な	
Ethernet SFR.....	90	内部 RC オシレータ	
アクセスバンク .....	88	WDT と併用.....	367
拡張命令セット .....	100	内部オシレータ ブロック .....	51
特殊機能レジスタ .....	89	内部電圧レギュレータ仕様 .....	442
バンクセレクト レジスタ (BSR).....	86		
汎用レジスタファイル .....	88	は	
メモリマップ		ハードウェア乗算器.....	127
Ethernet 関連の特殊機能レジスタ .....	90	性能比較 .....	127
PIC18F97J60 ファミリ.....	87	動作 .....	127
特殊機能レジスタ .....	89	はじめに .....	127
テーブルへの読み書き.....	83	パッケージ .....	465
テーブルポインタに対する操作 (表).....	108	詳細 .....	466
デバイス間の違い.....	476	マーキング.....	465
デバイス概要 .....	11	パラレルスレーブ ポート (PSP).....	168
機能 (100 ピンデバイス) .....	14	PORTD.....	168
機能 (64 ピンデバイス) .....	13	関連レジスタ .....	170
機能 (80 ピンデバイス) .....	13	選択 (PSPMODE ビット).....	168
ファミリ各製品の詳細 .....	12	パルス幅変調 → 「PWM (CCP モジュール) と	
電圧レギュレータ (内蔵).....	369	PWM (ECCP モジュール)」参照	
電気的特性.....	429	パワーアップ タイマ (PWRT).....	54, 66
Ethernet トランシーバ		パワーアップ遅延 .....	54
外部磁気回路の要件 .....	463	パワーオン リセット (POR).....	65
電力管理モード.....	55	タイムアウト シーケンス .....	66
アイドル.....	59	パワーアップ タイマ (PWRT) .....	66
PRI_IDLE.....	60	ひ	
RC_IDLE.....	61	標準 PIC 命令への影響 .....	422
SEC_IDLE .....	60	ピン配置と入出力の説明	
アイドル/スリープの終了.....	61	PIC18F66J60/66J65/67J60 .....	18
WDT タイムアウトによる .....	61	PIC18F86J60/86J65/87J60 .....	25
オシレータ起動タイマ遅延を伴わない.....	61	PIC18F96J60/96J65/97J60 .....	33
リセットによる .....	61	ピン割り当て	
割り込みによる .....	61	AVDD.....	24, 32, 42
移行 .....	55	AVss .....	24, 32, 42
一覧表.....	55	ENVREG .....	24, 32, 42
クロック切り換えとステータス		MCLR .....	18, 25, 33
インジケータ .....	56	OSC1/CLKI.....	18, 25, 33
クロック源 .....	55	OSC2/CLKO .....	18, 25, 33
スリープ .....	59	RA0/LEDA/AN0 .....	18, 25, 33
選択 .....	55	RA1/LEDB/AN1 .....	18, 25, 33
と SPI の動作 .....	277	RA2/AN2/VREF- .....	18, 25, 33
複数の SLEEP 命令 .....	56	RA3/AN3/VREF+ .....	18, 25, 33
ランモード .....	56	RA4/T0CKI .....	18, 25, 33
PRI_RUN .....	56	RA5/AN4.....	18, 25, 33
RC_RUN.....	58	RB0/INT0/FLT0 .....	19, 26, 34
SEC_RUN.....	56	RB1/INT1.....	19, 26, 34
と		RB2/INT2.....	19, 26, 34
特殊イベントトリガー		RB3/INT3.....	19, 26
「コンペア (ECCP モジュール)」参照		RB3/INT3/ECCP2/P2A .....	34
特殊機能レジスタ		RB4/KB10.....	19, 26, 34
Ethernet SFR.....	90	RB5/KB11.....	19, 26, 34
		RB6/KB12/PGC .....	19, 26, 34



# PIC18F97J60 ファミリ

書き込みシーケンス .....	111	EUSARTx 受信 .....	329
関連レジスタ .....	113	EUSARTx 送信 .....	326
コード保護中の動作 .....	113	MSSP (I <sup>2</sup> C マスタモード) .....	298
消去 .....	110	MSSP (I <sup>2</sup> C モード) .....	279
消去シーケンス .....	110	MSSP (SPI モード) .....	269
制御レジスタ .....	106	PIC18F66J60/66J65/67J60 .....	15
EECON1 および EECN2 .....	106	PIC18F86J60/86J65/87J60 .....	16
TABLAT ( テーブルラッチ ) .....	108	PIC18F96J60/96J65/97J60 .....	17
TBLPTR ( テーブルポインタ ) .....	108	PORTD と PORTE ( パラレルスレーブ ポート ) .....	168
テーブルポインタ 動作ごとの境界 .....	108	PWM 動作 ( 概略図 ) .....	194
テーブルポインタの境界 .....	108	RX 極性反転回路 ( TX 接続の記載は省略 ) .....	221
テーブル読み書き .....	105	Timer1 .....	176
読み出し .....	109	Timer1 ( 16 ビット読み書きモード ) .....	176
プリスケアラ Timer2 .....	204	Timer2 .....	181
プリスケアラ、Timer0 .....	173	Timer3 .....	184
プリスケアラ、Timer2 .....	195	Timer3 ( 16 ビット読み書きモード ) .....	184
プログラミング、デバイス命令 .....	375	Timer4 .....	188
プログラムカウンタ .....	81	アナログ入力モデル .....	343
PCLATH および PCLATU レジスタ .....	81	ウォッチドッグ タイマ .....	367
PCL、PCH、PCU レジスタ .....	81	外部 POR 回路 ( VDD 電源の 立ち上がりが遅い場合 ) .....	65
プログラムの検証とコード保護 .....	373	外部クロック入力動作 ( EC ) .....	50
プログラムメモリ 拡張命令セット .....	100	外部クロック入力動作 ( HS ) .....	50
命令 .....	85	拡張 PWM .....	203
2 ワード .....	85	キャプチャモード動作 .....	191
メモリマップ .....	77	コンパレータ参照電圧 .....	356
ハードベクタとコンフィグレーション ワード .....	78	コンパレータ参照電圧の 出力バッファの例 .....	357
メモリマップ、モード .....	80	コンパレータ出力 .....	352
モード メモリアクセス ( テーブル ) .....	80	コンパレータの I/O 動作モード .....	350
リセットベクタ .....	78	コンパレータのアナログ入力モデル .....	353
ルックアップ テーブル .....	83	コンペアモードの動作 .....	192
割り込みベクタ .....	78	シングル コンパレータ .....	351
プログラムメモリ モード .....	79	水晶発振回路の動作 ( HS、HSPLL ) .....	50
アドレスシフト ( 拡張マイクロコントローラ ) .....	80	テーブル書き込み動作 .....	106
拡張マイクロコントローラ .....	79	テーブル読み出し動作 .....	105
マイクロコントローラ .....	79	デバイスクロック .....	49
ブロック図		内蔵電圧レギュレータの接続 .....	369
16 ビットバイト書き込みモード .....	119	内蔵リセット回路 .....	63
16 ビットバイト選択モード .....	121	ハーフブリッジ出力モード アプリケーション .....	207
16 ビットモードの Timer0 .....	172	汎用 I/O ポートの動作 .....	145
16 ビットワード書き込みモード .....	120	フェイルセーフ クロック モニタ .....	371
8 ビット多重化モード .....	123	フラッシュ プログラムメモリからの 読み出し .....	109
8 ビットモードの Timer0 .....	172	フラッシュ プログラムメモリに対する テーブル書き込み .....	111
A/D .....	342	フルブリッジ アプリケーションの例 .....	209
baud レート ジェネレータ .....	300	割り込みロジック .....	130
Ethernet の動作に必要な外付け部品 .....	219		
Ethernet モジュール .....	217		
Ethernet モジュールの 割り込みロジック .....	239		



# PIC18F97J60 ファミリ

へ		IORWF .....	398
ベンダ識別子 (OUI) .....	248	LFSR .....	399
ま		MOVF .....	399
マイクロチップ社ウェブサイト .....	491	MOVFF .....	400
マスタクリア (MCLR) .....	65	MOVLB .....	400
マスタ同期シリアルポート (MSSP) → 「MSSP」参照		MOVLW .....	401
め		MOVWF .....	401
命令サイクル .....	84	MULLW .....	402
クロック方式 .....	84	MULWF .....	402
フロー/パイプライン処理 .....	84	NEGF .....	403
命令セット .....	375	NOP .....	403
ADDLW .....	381	POP .....	404
ADDWF .....	381	PUSH .....	404
ADDWF (インデックス付き リテラル オフセットモード) .....	423	RCALL .....	405
ADDWFC .....	382	RESET .....	405
ANDLW .....	382	RETFIE .....	406
ANDWF .....	383	RETLW .....	406
BC .....	383	RETURN .....	407
BCF .....	384	RLCF .....	407
BN .....	384	RLNCF .....	408
BNC .....	385	RRCF .....	408
BNN .....	385	RRNCF .....	409
BNOV .....	386	SETF .....	409
BNZ .....	386	SETF (インデックス付き リテラル オフセットモード) .....	423
BOV .....	389	SLEEP .....	410
BRA .....	387	SUBFWB .....	410
BSF .....	387	SUBLW .....	411
BSF (インデックス付き リテラル オフセットモード) .....	423	SUBWF .....	411
BTFSC .....	388	SUBWFB .....	412
BTFSS .....	388	SWAPF .....	412
BTG .....	389	TBLRD .....	413
BZ .....	390	TBLWT .....	414
CALL .....	390	TSTFSZ .....	415
CLRF .....	391	XORLW .....	415
CLRWDT .....	391	XORWF .....	416
COMF .....	392	一般的な形式 .....	377
CPFSEQ .....	392	拡張命令 .....	417
CPFSGT .....	393	MPLAB IDE ツールでの使用 .....	424
CPFSLT .....	393	構文 .....	417
DAW .....	394	有効にする際の注意事項 .....	422
DCFSNZ .....	395	標準命令 .....	375
DECF .....	394	メモリ プログラミングの要件 .....	441
DECFSZ .....	395	メモリ構成 .....	77
GOTO .....	396	データメモリ .....	86
INCF .....	396	プログラムメモリ .....	77
INCFSZ .....	397	り	
INFSNZ .....	397	リセット .....	63, 359
IORLW .....	398	RESET 命令 .....	63
		オシレータ起動タイマ (OST) .....	359
		コード実行中のウォッチドッグ タイマ (WDT) リセット .....	63
		コンフィグレーション不一致 (CM) .....	63

# PIC18F97J60 ファミリ

スタック アンダーフロー リセット .....	63
スタックフル リセット .....	63
スタックフル / アンダーフロー .....	83
通常動作中の MCLR リセット .....	63
電力管理モード時の MCLR リセット .....	63
パワーアップ タイマ (PWRT) .....	359
パワーオン リセット (POR) .....	63, 359
ブラウンアウト リセット (BOR) .....	63, 359
レジスタの状態 .....	68
リターン スタックポインタ (STKPTR) .....	82
リターンアドレス スタック .....	81

## れ

### レジスタ

ADCON0 (A/D 制御 0) .....	339
ADCON1 (A/D 制御 1) .....	340
ADCON2 (A/D 制御 2) .....	341
BAUDCONx (baud レート制御 x) .....	318
CCPxCON (拡張 CCPx 制御、 ECCP1/ECCP2/ECCP3) .....	198
CCPxCON (キャプチャ / コンペア / PWM 制御、CCP4 と CCP5) .....	189
CMCON (コンパレータ制御) .....	349
CONFIG1H (コンフィグレーション 1 High) .....	361
CONFIG1L (コンフィグレーション 1 Low) .....	361
CONFIG2H (コンフィグレーション 2 High) .....	363
CONFIG2L (コンフィグレーション 2 Low) .....	362
CONFIG3H (コンフィグレーション 3 High) .....	365
CONFIG3L (コンフィグレーション 3 Low) ...	79, 364
CVRCON (コンパレータ 参照電圧制御) .....	355
DEVID1 (デバイス ID 1) .....	366
DEVID2 (デバイス ID 2) .....	366
ECCP1AS (ECCP1 自動シャットダウン コンフィグレーション) .....	212
ECCP1DEL (ECCP1 デッドバンド遅延) .....	211
ECON1 (Ethernet 制御 1) .....	227
ECON2 (Ethernet 制御 2) .....	228
EECON1 (EEPROM 制御 1) .....	107
EFLOCON (Ethernet フロー制御) .....	258
EIE (Ethernet 割り込みイネーブル) .....	240
EIR (Ethernet 割り込み要求、フラグ) ...	241
ERXFCON (Ethernet 受信フィルタ制御) ...	260
ESTAT (Ethernet ステータス) .....	228
INTCON (割り込み制御) .....	131
INTCON2 (割り込み制御 2) .....	132
INTCON3 (割り込み制御 3) .....	133
IPR1 (周辺割り込み優先度 1) .....	140

IPR2 (周辺割り込み優先度 2) .....	141
IPR3 (周辺割り込み優先度 3) .....	142
MABBIPG (MAC 連続パケット間ギャップ) .....	246
MACON1 (MAC 制御 1) .....	229
MACON3 (MAC 制御 3) .....	230
MACON4 (MAC 制御 4) .....	231
MEMCON (外部メモリバス制御) .....	116
MICMD (MII コマンド) .....	231
MISTAT (MII ステータス) .....	232
OSCCON (オシレータ制御) .....	53
OSCTUNE (PLL ブロック制御) .....	51
PHCON1 (PHY 制御 1) .....	235
PHCON2 (PHY 制御 2) .....	236
PHIE (PHY 割り込みイネーブル) .....	242
PHIR (PHY 割り込み要求、フラグ) .....	242
PHLCON (PHY モジュール LED 制御) ..	238
PHSTAT1 (PHY ステータス 1) .....	235
PHSTAT2 (PHY ステータス 2) .....	237
PIE1 (周辺割り込みイネーブル 1) .....	137
PIE2 (周辺割り込みイネーブル 2) .....	138
PIE3 (周辺割り込みイネーブル 3) .....	139
PIR1 (周辺割り込み要求 (フラグ) 1) ...	134
PIR2 (周辺割り込み要求 (フラグ) 2) ...	135
PIR3 (周辺割り込み要求 (フラグ) 3) ...	136
PSPCON (パラレルスレーブ ポート制御) .....	169
RCON (リセット制御) .....	64, 143
RCSTAx (受信ステータス / 制御 x) .....	317
SSPxCON1 (MSSPx 制御 1、 I <sup>2</sup> C モード) .....	281
SSPxCON1 (MSSPx 制御 1、 SPI モード) .....	271
SSPxCON2 (MSSPx 制御 2、 I <sup>2</sup> C スレーブモード) .....	283
SSPxCON2 (MSSPx 制御 2、 I <sup>2</sup> C マスタモード) .....	282
SSPxSTAT (MSSPx ステータス、 I <sup>2</sup> C モード) .....	280
SSPxSTAT (MSSPx ステータス、 SPI モード) .....	270
STATUS .....	97
STKPTR (スタックポインタ) .....	82
T0CON (Timer0 制御) .....	171
T1CON (Timer1 制御) .....	175
T2CON (Timer2 制御) .....	180
T3CON (Timer3 制御) .....	183
T4CON (Timer4 制御) .....	187
TXSTAx (送信ステータス / 制御 x) .....	316
WDTCON (ウォッチドッグタイマ制御) .....	368
レジスタファイルのまとめ .....	91 ~ 96

## わ

割り込み .....	129
コンテキスト保存 .....	144
割り込み要因 .....	359
INTx ピン .....	144
PORTB、状態変化割り込み .....	144
TMR0 .....	144
TMR0 オーバーフロー .....	173
TMR1 オーバーフロー .....	175
TMR2/PR2 一致 (PWM) .....	203
TMR3 オーバーフロー .....	183, 185
TMR4/PR4 一致 .....	188
TMR4/PR4 一致 (PWM) .....	187
状態変化割り込み (RB7:RB4) .....	148
割り込み、フラグビット	
状態変化割り込み (RB7:RB4) フラグ (RBIF ビット) .....	148

# PIC18F97J60 ファミリ

---

NOTE:

## マイクロチップ社ウェブサイト

マイクロチップ社は、弊社が運営する WWW サイト ([www.microchip.com](http://www.microchip.com)) を通してオンラインサポートを提供しています。このウェブサイトを活用する事で、ファイルや情報を簡単に入手できます。お好みのインターネット ブラウザを使って、下記の内容をご覧になれます。

- 製品サポート — データシートとエラッタ、アプリケーションノートとサンプル プログラム、設計関連リソース、ユーザガイドとハードウェア サポート文書、最新のソフトウェア リリースと過去のソフトウェア
- 技術サポート — よく寄せられる質問 (FAQ)、技術サポート リクエスト、オンライン ディスカッショングループ、マイクロチップ社コンサルタント プログラム メンバーの一覧
- マイクロチップ社の事業 — 製品セレクトと注文のガイド、マイクロチップ社の最新プレスリリース、セミナーとイベントの一覧、マイクロチップ社の各営業所、販売代理店、工場の一覧

## お客様向け変更通知サービス

マイクロチップ社のお客様向け変更通知サービスは、お客様がマイクロチップ社製品の最新情報を入手できるようにします。ご興味のある製品ファミリまたは開発ツールに関する変更、更新、リビジョン、エラッタ情報をいち早くメールにてお知らせします。

当サービスをご希望のお客様は、マイクロチップ社ウェブサイト ([www.microchip.com](http://www.microchip.com)) でご登録ください。[ サポート ] → [ お客様向け変更通知 ] をクリックし、画面の指示に従ってください。

## カスタマサポート

マイクロチップ社製品のお客様は、以下の複数のルートでサポートをご利用頂けます。

- 代理店または販売担当者
- 最寄りの営業所
- フィールド アプリケーション エンジニア (FAE)
- 技術サポート
- 開発システム情報ライン

サポートは販売代理店、販売担当者、フィールド アプリケーション エンジニア (FAE) までお問い合わせください。各地の営業所もご利用頂けます。本書の末尾には各国の営業所の一覧を記載しています。

技術サポートは下記のウェブサイトからもご利用頂けます : <http://microchip.com/support>

# PIC18F97J60 ファミリ

---

## お客様アンケート

マイクロチップ社は、弊社製品を存分にご活用頂くために、文書の作成に最善の努力を尽くしています。本書の構成、明確さ、内容等に関するご意見を FAX にてお寄せください (宛先: マイクロチップ・テクノロジー・ジャパン株式会社: FAX 番号: 045-471-6122)。

下記のアンケート フォームにお客様情報と本書に関するご意見をご記入ください。

宛先: マイクロチップ・テクノロジー・ジャパン株式会社

送信ページ数 \_\_\_\_\_

件名: お客様アンケート

送信者: お名前 \_\_\_\_\_

貴社名 \_\_\_\_\_

ご住所 \_\_\_\_\_

郵便番号 \_\_\_\_\_

電話番号: (\_\_\_\_\_) \_\_\_\_\_ - \_\_\_\_\_ FAX 番号: (\_\_\_\_\_) \_\_\_\_\_ - \_\_\_\_\_

アプリケーション (任意):

返信をご希望ですか?  はい  いいえ

デバイス: PIC18F97J60 ファミリ

文書番号: DS39762F\_JP

質問:

1. 本書で最も良かった点は何ですか。

---

---

2. 本書はお客様のハードウェア/ソフトウェア開発に役立ちますか。

---

---

3. 本書の構成は分かりやすいですか。分かりにくい場合はその理由もご記入ください。

---

---

4. 本書の構成と内容を改善するには何を追加すべきですか。

---

---

5. 本書から省略してもかまわない内容は何ですか。

---

---

6. 不正確であったり誤解を招きやすい箇所がございましたらご指摘ください。

---

---

7. その他、本書の改善に向けてご意見がございましたらご記入ください。

---

---

# PIC18F97J60 ファミリ

## 製品識別システム

ご注文または製品の価格や納期等に関するお問い合わせは、弊社または販売代理店までお問い合わせください。

<b>PART NO.</b>	<b>X</b>	<b>/XX</b>	<b>XXX</b>
<b>Device</b>	<b>Temperature Range</b>	<b>Package</b>	<b>Pattern</b>
Device	PIC18F66J60/66J65/67J60, PIC18F86J60/86J65/87J60, PIC18F96J60/96J65/97J60, PIC18F66J60/66J65/67J60T <sup>(1)</sup> , PIC18F86J60/86J65/87J60T <sup>(1)</sup> , PIC18F96J60/96J65/97J60T <sup>(1)</sup>		
Temperature Range	I = -40°C to +85°C (Industrial)		
Package	PT = 64, 80 and 100-Lead, 12x12x1 mm TQFP (Thin Quad Flatpack) PF = 100-Lead, 14x14x1 mm TQFP		
Pattern	QTP, SQTP, Code or Special Requirements (blank otherwise)		

**例 :**

- a) PIC18F67J60-I/PT 301 = 産業用温度レンジ、TQFP パッケージ、QTP パターン #301
- b) PIC18F67J60T-I/PT = テープ&リール、産業用温度レンジ、TQFP パッケージ

**Note 1:** T = テープ & リール

## 各国の営業所とサービス

### 北米

#### 本社

2355 West Chandler Blvd.  
Chandler, AZ 85224-6199

Tel: 480-792-7200

Fax: 480-792-7277

技術サポート:

[http://www.microchip.com/  
support](http://www.microchip.com/support)

URL:

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

#### アトランタ

Duluth, GA

Tel: 678-957-9614

Fax: 678-957-1455

#### ボストン

Westborough, MA

Tel: 774-760-0087

Fax: 774-760-0088

#### シカゴ

Itasca, IL

Tel: 630-285-0071

Fax: 630-285-0075

#### クリーブランド

Independence, OH

Tel: 216-447-0464

Fax: 216-447-0643

#### ダラス

Addison, TX

Tel: 972-818-7423

Fax: 972-818-2924

#### デトロイト

Farmington Hills, MI

Tel: 248-538-2250

Fax: 248-538-2260

#### インディアナポリス

Noblesville, IN

Tel: 317-773-8323

Fax: 317-773-5453

#### ロサンゼルス

Mission Viejo, CA

Tel: 949-462-9523

Fax: 949-462-9608

#### サンタクララ

Santa Clara, CA

Tel: 408-961-6444

Fax: 408-961-6445

#### トロント

Mississauga, Ontario,  
Canada

Tel: 905-673-0699

Fax: 905-673-6509

### アジア / 太平洋

#### アジア太平洋支社

Suites 3707-14, 37th Floor

Tower 6, The Gateway

Harbour City, Kowloon

Hong Kong

Tel: 852-2401-1200

Fax: 852-2401-3431

#### オーストラリア - シドニー

Tel: 61-2-9868-6733

Fax: 61-2-9868-6755

#### 中国 - 北京

Tel: 86-10-8569-7000

Fax: 86-10-8528-2104

#### 中国 - 成都

Tel: 86-28-8665-5511

Fax: 86-28-8665-7889

#### 中国 - 重慶

Tel: 86-23-8980-9588

Fax: 86-23-8980-9500

#### 中国 - 杭州

Tel: 86-571-2819-3187

Fax: 86-571-2819-3189

#### 中国 - 香港 SAR

Tel: 852-2943-5100

Fax: 852-2401-3431

#### 中国 - 南京

Tel: 86-25-8473-2460

Fax: 86-25-8473-2470

#### 中国 - 青島

Tel: 86-532-8502-7355

Fax: 86-532-8502-7205

#### 中国 - 上海

Tel: 86-21-5407-5533

Fax: 86-21-5407-5066

#### 中国 - 瀋陽

Tel: 86-24-2334-2829

Fax: 86-24-2334-2393

#### 中国 - 深圳

Tel: 86-755-8864-2200

Fax: 86-755-8203-1760

#### 中国 - 武漢

Tel: 86-27-5980-5300

Fax: 86-27-5980-5118

#### 中国 - 西安

Tel: 86-29-8833-7252

Fax: 86-29-8833-7256

#### 中国 - 厦門

Tel: 86-592-2388138

Fax: 86-592-2388130

#### 中国 - 珠海

Tel: 86-756-3210040

Fax: 86-756-3210049

### アジア / 太平洋

#### インド - バンガロール

Tel: 91-80-3090-4444

Fax: 91-80-3090-4123

#### インド - ニューデリー

Tel: 91-11-4160-8631

Fax: 91-11-4160-8632

#### インド - プネ

Tel: 91-20-2566-1512

Fax: 91-20-2566-1513

#### 日本 - 大阪

Tel: 81-66-152-7160

Fax: 81-66-152-9310

#### 日本 - 横浜

Tel: 81-45-471-6166

Fax: 81-45-471-6122

#### 韓国 - 大邱

Tel: 82-53-744-4301

Fax: 82-53-744-4302

#### 韓国 - ソウル

Tel: 82-2-554-7200

Fax: 82-2-558-5932 または

82-2-558-5934

#### マレーシア - クアラルンプール

Tel: 60-3-6201-9857

Fax: 60-3-6201-9859

#### マレーシア - ペナン

Tel: 60-4-227-8870

Fax: 60-4-227-4068

#### フィリピン - マニラ

Tel: 63-2-634-9065

Fax: 63-2-634-9069

#### シンガポール

Tel: 65-6334-8870

Fax: 65-6334-8850

#### 台湾 - 新竹

Tel: 886-3-5778-366

Fax: 886-3-5770-955

#### 台湾 - 高雄

Tel: 886-7-213-7828

Fax: 886-7-330-9305

#### 台湾 - 台北

Tel: 886-2-2508-8600

Fax: 886-2-2508-0102

#### タイ - バンコク

Tel: 66-2-694-1351

Fax: 66-2-694-1350

### ヨーロッパ

#### オーストリア - ヴェルス

Tel: 43-7242-2244-39

Fax: 43-7242-2244-393

#### デンマーク - コペンハーゲン

Tel: 45-4450-2828

Fax: 45-4485-2829

#### フランス - パリ

Tel: 33-1-69-53-63-20

Fax: 33-1-69-30-90-79

#### ドイツ - ミュンヘン

Tel: 49-89-627-144-0

Fax: 49-89-627-144-44

#### イタリア - ミラノ

Tel: 39-0331-742611

Fax: 39-0331-466781

#### オランダ - ドリュウネン

Tel: 31-416-690399

Fax: 31-416-690340

#### スペイン - マドリッド

Tel: 34-91-708-08-90

Fax: 34-91-708-08-91

#### イギリス - ウォーキングム

Tel: 44-118-921-5869

Fax: 44-118-921-5820