

# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。  
<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。



# PIC16C7X

## A/D コンバータ内蔵 8 ビット CMOS マイクロコントローラ

このデータシートには下記のデバイスが含まれています。

- ・PIC16C710
- ・PIC16C71
- ・PIC16C711
- ・PIC16C72
- ・PIC16C73
- ・PIC16C73A
- ・PIC16C74
- ・PIC16C74A

### PIC16C7X マイクロコントローラの特徴

- ・高性能 RISC CPU
- ・覚える必要があるのは35個のシングルワード命令のみ
- ・2サイクルのプログラム分岐命令を除いて、全てシングルサイクル命令(200ns)
- ・動作速度：DC-20MHz クロック入力  
DC-200ns 命令サイクル
- ・割込み機能
- ・深さ 8 レベルのハードウェアスタック
- ・直接(ダイレクト)、(間接)インダイレクト、相対(リラティブ)の各アドレッシングモード
- ・パワーオンリセット(POR)
- ・パワーアップタイム(PWRT)とオシレータスタートアップタイム(OST)
- ・確実な動作のために専用のオンチップRCオシレータを内蔵したウォッチドッグタイマ(WDT)
- ・プログラム可能なコードプロテクション
- ・消費電力を節約する SLEEP モード
- ・選択可能なオシレータオプション
- ・8 ビット・マルチチャンネルの A/D コンバータ

- ・低消費電力、高速 CMOS EPROM 技術
- ・完全スタティック設計
- ・幅広い動作電圧範囲(2.5V-6.0V)
- ・最大シンク電流 25mA/ 最大ソース電流 25mA
- ・商用、工業用、車載用の温度範囲
- ・低消費電力：
  - < 2mA @ 5V, 4MHz
  - 15  $\mu$  A typ. @ 3V, 32kHz
  - < 1  $\mu$  A typ. スタンバイ電流

### PIC16C7X 周辺機能の特徴

- ・タイマ0: 8ビットプリスケラを持った8ビットのタイマ/カウンタ
- ・タイマ1: 16ビットのタイマ/カウンタ  
タイマ1は、外部クリスタル/クロック使用の場合スリープ期間中も動作が可能
- ・タイマ2: 8ビットのピリオドレジスタで、プリスケラ、ポストスケラを持った8ビットのタイマ/カウンタ
- ・キャプチャ、コンペア、PWM モジュール
- ・キャプチャは 16 ビットで、最大分解能が 12.5ns  
コンペアは 16 ビットで、最大分解能が 200ns、PWM の分解能は、10 ビット
- ・SPI™ と I<sup>2</sup>C™ を持った同期シリアルポート (SSP)
- ・ユニバーサル同期 / 非同期レシーバとトランスミッタ (USART/SCI)
- ・RD、WR、CS を外部からコントロールする 8 ビット幅のパラレルスレーブポート (PSP)
- ・電源電圧の低下時にリセットさせるためのブラウンアウト (電圧低下) 検出回路 (BOR)

PIC16C7Xの特徴	710	71	711	72	73	73A	74	74A
プログラムメモリ (EPROM) x14bit	512	1K	1K	2K	4K	4K	4K	4K
データメモリ (Bytes) x8bit	36	36	68	128	192	192	192	192
I/O ピン数	13	13	13	22	22	22	33	33
パラレルスレーブポート	-	-	-	-	-	-	Yes	Yes
Capture/Compare/PWMモジュール	-	-	-	1	2	2	2	2
Timer モジュール	1	1	1	3	3	3	3	3
A/D チャンネル	4	4	4	5	5	5	8	8
シリアル コミュニケーション	-	-	-	SPI/I <sup>2</sup> C	SPI/I <sup>2</sup> C USART	SPI/I <sup>2</sup> C USART	SPI/I <sup>2</sup> C USART	SPI/I <sup>2</sup> C USART
インサート シリアル フログ ラミング	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
電圧低下リセット	Yes	-	Yes	Yes	-	Yes	-	Yes
インターラプト数	4	4	4	8	11	11	12	12

I<sup>2</sup>Cは Phillips Corporation の登録商標です。 SPI は Motorola Corporation の登録商標です。

# 暫定版

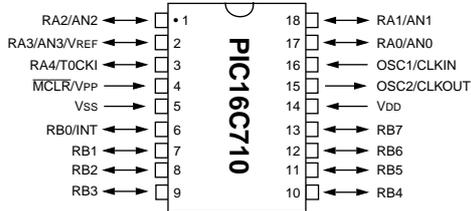
英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。

## PIC16C7X

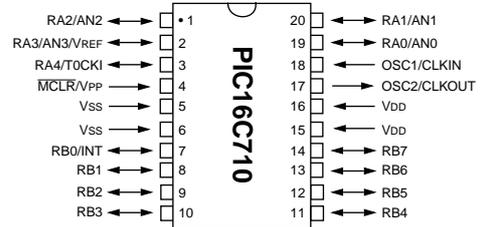
<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

### ピン・ダイアグラム

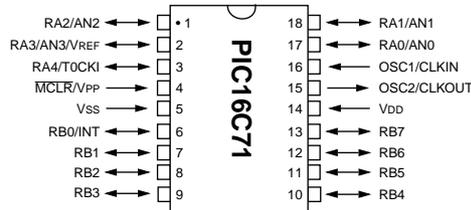
PDIP, SOIC, Windowed CERDIP



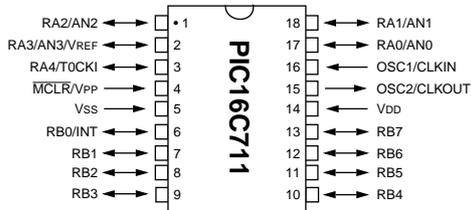
SSOP



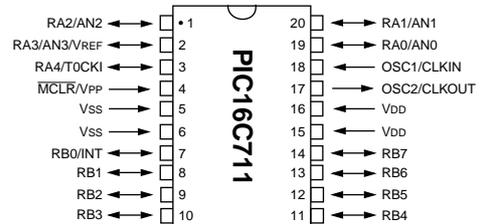
PDIP, SOIC, Windowed CERDIP



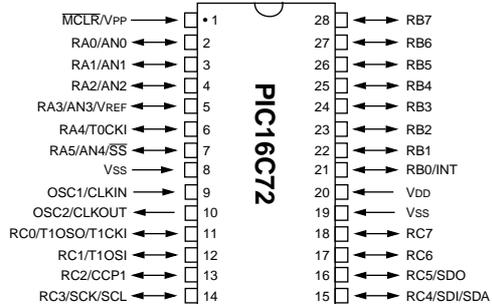
PDIP, SOIC, Windowed CERDIP



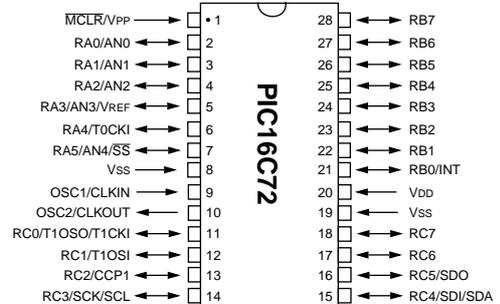
SSOP



SDIP, SOIC, Windowed Side Brazed Ceramic



SSOP

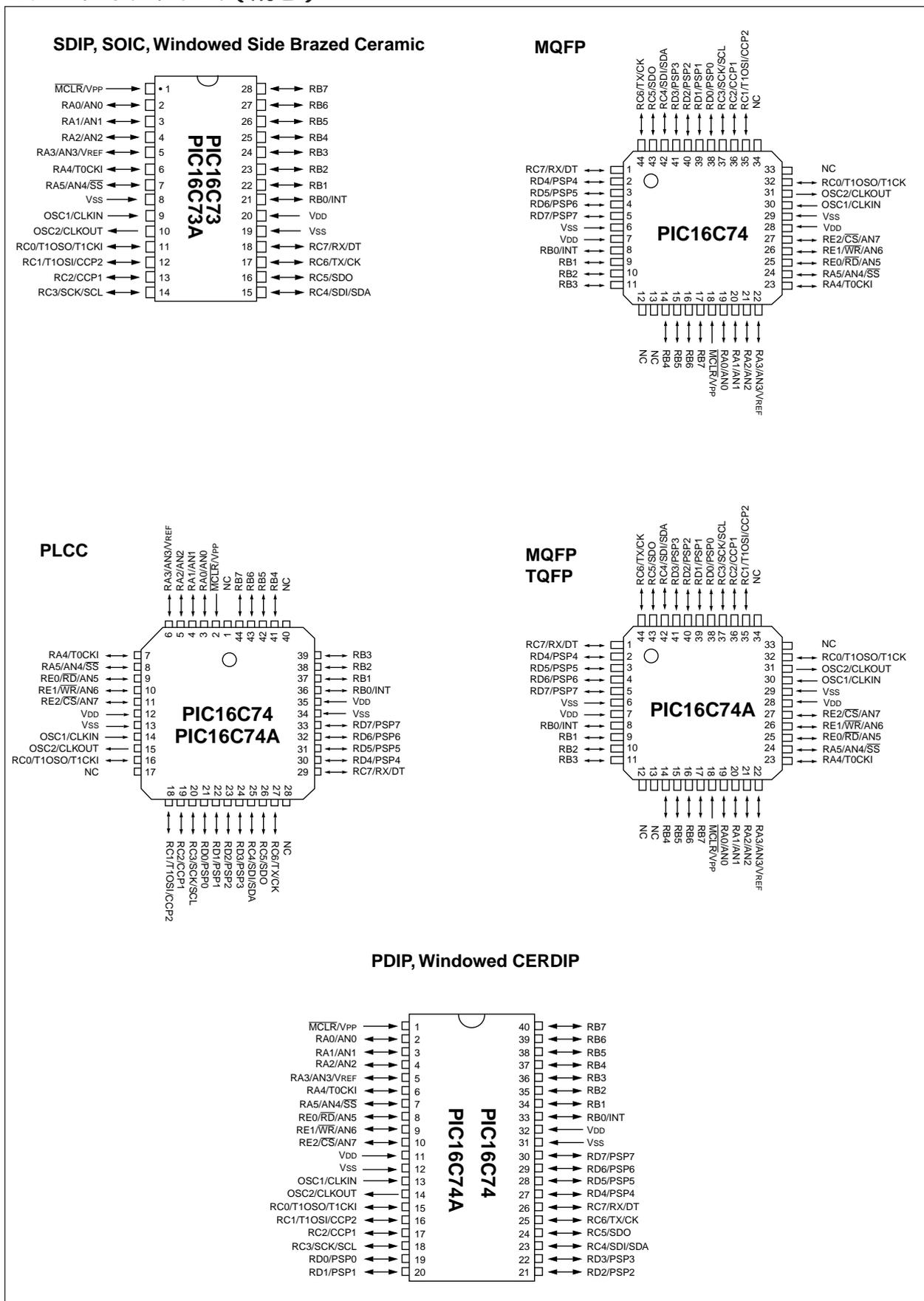


# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。  
<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

# PIC16C7X

## ピン・ダイアグラム ( 続き )



# 暫定版

英語最新版データシートと併用にご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。

## PIC16C7X

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

### 目次

1.0 概要	5
2.0 PIC16C7Xデバイスの種類	7
3.0 アーキテクチャの概要	9
4.0 メモリ構成	21
5.0 I/Oポート	43
6.0 タイマ・モジュールの概要	57
7.0 タイマ0モジュール	59
8.0 タイマ1モジュール	65
9.0 タイマ2モジュール	69
10.0 キャプチャ/コンペア/PWMモジュール	71
11.0 同期シリアルポート(SSP)モジュール	77
12.0 ユニバーサル同期・非同期レシーバ・トランスミッター(USART)	93
13.0 アナログ・デジタル変換器(A/D)モジュール	109
14.0 CPUの特殊機能	121
15.0 命令セット概要	141
16.0 開発サポート	153
17.0 PIC16C710とPIC16C711における電気的特性	
18.0 PIC16C710とPIC16C711におけるDC・AC特性のグラフと表	
19.0 PIC16C71における電気的特性	
20.0 PIC16C71におけるDC・AC特性のグラフと表	
21.0 PIC16C72における電気的特性	
22.0 PIC16C72におけるDC・AC特性のグラフと表	
23.0 PIC16C73/74における電気的特性	
24.0 PIC16C73/74におけるDC・AC特性のグラフと表	
25.0 PIC16C73A/74Aにおける電気的特性	
26.0 PIC16C73A/74AにおけるDC・AC特性のグラフと表	
27.0 パッケージング情報	267
付録 A:	
付録 B: 互換性	
付録 C: 新規追加事項	
付録 D: 変更事項	
付録 E: PIC16/17マイクロコントローラ	
索引	
Microchip BBSへの接続方法	
読者のご意見	
製品情報システム	312

**AC/DC スペックは、最新版の  
英文データシートをご覧ください。**

**本資料では、この部分は省略。**

このデータ・シートのレジスタとモジュールの説明において、デバイスの凡例は、それぞれのセクションに適用するデバイスであることを示しています。例えば、下記の凡例は、次のセクションがPIC16C711、PIC16C72、PIC16C73A、PIC16C74Aのデバイスだけに適用することを意味しています。

適用デバイス							
710	71	711	72	73	73A	74	74A

#### お客様へ

当社では、当社製品や解説書の品質を高めるために常に努力をしております。この解説書も正確を期すために非常に多くの時間を費やしておりますが、少しの見落としがあるかもしれません。もし見落としや間違っている情報にお気づきになりましたら、当社までお知らせください。より良い解説書を作るために皆様のご協力に感謝いたします。

## 1.0 概要

PIC16C7Xは、PIC16CXXの中規模ファミリの中で、集積化されたアナログ-ディジタル(A/D)変換器を持つ低価格、高性能、CMOS、完全スタティックな8ビット・マイクロコントローラです。

すべてのPIC16/17マイクロコントローラは、先進のRISCアーキテクチャを採用しています。PIC16CXXマイクロコントローラ・ファミリには、拡張コア機能、8レベル・ディープ・スタック、複数の内部、外部割込みソースが採用されています。ハーバードアーキテクチャによる命令とデータが分離したバスにより、分離された8ビット幅データと14ビット幅の命令語が可能になります。2段の命令パイプラインにより、プログラム分岐(2サイクル必要)を除いて、すべての命令がシングル・サイクルで実行できます。合計35個の命令(縮小命令セット)が使用可能です。さらに、多くのレジスタセットは、高性能を実現するために利用されるアーキテクチャの革新をもたらします。

**PIC16CXXマイクロコントローラは、同クラスの他社製8ビットマイクロコントローラと比べて、一般的に2:1のコード圧縮、4:1のスピード改善を実現します。**

PIC16C710/711デバイスは36バイトのRAM、PIC16C711は68バイトのRAMを持ち、PIC16C710/711デバイスには、13本のI/Oピンがあります。さらに、タイマ/カウンタも使用可能です。また、4チャンネルの高速8ビットA/D変換器も内蔵されています。8ビットの分解能は、低価格のアナログインターフェイスを要求する応用、例えば、サーモスタットの制御、圧力センサーなどに対して最適です。

PIC16C72デバイスは、128バイトのRAMと22本のI/Oピンを持っています。さらに、3個のタイマ/カウンタ、1個のキャプチャ/コンペア/PWMモジュール、1個のシリアル・ポートを含むいくつかのペリフェラル(周辺機能)があります。同期シリアルポートは、3ワイヤのシリアルペリフェラルインターフェイス(SPI)や、2ワイヤのI<sup>2</sup>Cバスも構成できます。また、5チャンネルの高速8ビットA/D変換器が内蔵されています。8ビットの分解能は、低価格のアナログインターフェイスを要求する応用、例えば、サーモスタットの制御、圧力センサーなどに対して最適です。

PIC16C73/73Aデバイスは、192バイトのRAMと22本のI/Oピンを持っています。さらに、3個のタイマ/カウンタ、2個のキャプチャ/コンペア/PWMモジュール、2個のシリアル・ポートを含むいくつかのペリフェラルがあります。シンクロナスシリアルポートは、3ワイヤのシリアルペリフェラルインターフェイス(SPI)や、2ワイヤのI<sup>2</sup>Cバスも構成できます。この汎用同期・非同期レシーバトランスミッタ(USART)は、シリアルコミュニケーション・シオンインターフェイスもしくはSCIとしても知られています。また、5チャンネルの高速8ビットA/D変換器も内蔵されています。8ビットの分解能は、低価格のアナログインターフェイスを要求する応用、例えば、サーモスタットの制御、圧力センサーなどに対して最適です。

PIC16C74/74Aデバイスは、192バイトのRAMと33本のI/Oピンを持っています。さらに、3個のタイマ/カウンタ、2個のキャプチャ/コンペア/PWMモジュール、2個のシリアル・ポートを含むいくつかのペリフェラルがあります。同期シリアルポートは、3ワイヤのシリアルペリ

フェラルインターフェイス(SPI)や、2ワイヤのI<sup>2</sup>Cバスも構成できます。この汎用同期・非同期レシーバトランスミッタ(USART)は、シリアルコミュニケーション・シオンインターフェイスもしくはSCIとしても知られています。8ビットのパラレルスレーブポートも備わっています。また、8チャンネルの高速8ビットA/D変換器も内蔵されています。8ビットの分解能は、低価格のアナログインターフェイスを要求する応用、例えば、サーモスタットの制御、圧力センサーなどに対して最適です。

PIC16C7Xファミリには、外部コンポーネントを減らすための特別な機能があり、それにより、コストの削減、システムの信頼性向上、消費電力の低減が可能になります。4種類のオシレータがあり、低価格化が可能なシングル・ピン構成のRCオシレータ、消費電力を最小限におさえるLPオシレータ、標準クリスタルのXT、高速クリスタルのHSの中から選択できます。SLEEP(パワーダウン)機能により電力の節約モードにできます。複数の外部・内部割込みトリセットによりチップをSLEEPからウェーク・アップさせることができます。

**専用の内蔵RCオシレータを持った信頼性の高いウォッチドッグタイマによりソフトウェアを誤動作から保護することができます。**

UV消去可能なCERDIPパッケージ版はコード開発時に、コストの点からワンタイムプログラマブル(OTP)版は量産時に最適です。

PIC16C7Xファミリは、セキュリティとリモートセンサから機器制御や車載用までの応用範囲に完全に適応します。EPROM技術により、応用プログラム(トランスミッタコード、モータスピード、受信周波数など)を非常に速く便利にカスタマイズできます。小型表面実装パッケージにより、このマイクロコントローラシリーズはスペースに制限のある応用にも完全に適応します。低価格、低消費電力、高性能、使いやすさ、I/Oのフレキシビリティにより、PIC16C7Xは今までマイクロコントローラの使用が検討されなかった分野(タイマ機能、シリアルコミュニケーション、キャプチャとコンペア、PWM機能、コプロセッサアプリケーションなど)にも大きな能力を発揮します。

### 1.1 ファミリと上位互換性

PIC16C5Xのマイクロコントローラファミリに精通したユーザは、これがPIC16C5Xアーキテクチャの拡張版であるとお分かりでしょう。改良点の詳細なリストは付録Aをご覧ください。PIC16C5X用に書かれたコードは、PIC16CXXファミリの製品に簡単に移植できます(付録B参照)。

### 1.2 開発サポート

PIC16CXXファミリには完全機能のマクロアセンブラ、ソフトウェアシミュレータ、インサーキットエミュレータ、低価格開発用プログラマ、完全機能のプログラマがサポートされています。“C”コンパイラやファジーロジック・サポートツールも利用できます(16.0章参照)。

# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。

**PIC16C7X** <http://www.microchip.com> より最新ERRATAシートを入手しご参照下さい。

表 1 - 1 : PIC16C7X ファミリのデバイス

最大動作周波数  
プログラマブル EPROM (x14bit)  
データメモリー (Byte)  
キャッチ/比較/PWM モジュラー  
SPI/RS485-ポート (SPI/12C, USART)  
A/Dコンバータ (8-bit) 5チャンネル  
I/Oポート (10ピン)  
低電力モード (BOR)  
ソフトウェアリセット (BOR)

PIC16C710	20	512	36	TMR0	-	-	-	4	4	13	3.0-6.0	Yes	Yes	18-pin DIP, SQC 20-pin SSOP
PIC16C71	20	1K	36	TMR0	-	-	-	4	4	13	3.0-6.0	Yes	-	18-pin DIP, SQC
PIC16C711	20	1K	68	TMR0	-	-	-	4	4	13	3.0-6.0	Yes	Yes	18-pin DIP, SQC 20-pin SSOP
PIC16C72	20	2K	128	TMR0 TMR1, TMR2	1	SPI/12C	-	5	8	22	2.5-6.0	Yes	Yes	20-pin SDI P, SQC, SSOP
PIC16C73	20	4K	192	TMR0 TMR1, TMR2	2	SPI/12C USART	-	5	11	22	3.0-6.0	Yes	-	28-pin SDI P, SQC
PIC16C73A	20	4K	192	TMR0 TMR1, TMR2	2	SPI/12C USART	-	5	11	22	2.5-6.0	Yes	Yes	28-pin SDI P, SQC
PIC16C74	20	4K	192	TMR0 TMR1, TMR2	2	SPI/12C USART	Yes	8	12	33	3.0-6.0	Yes	-	40-pin DIP 44-pin PLCC, MQFP
PIC16C74A	20	4K	192	TMR0 TMR1, TMR2	2	SPI/12C USART	Yes	8	12	33	2.5-6.0	Yes	Yes	40-pin DIP 44-pin PLCC, MQFP, TQFP

すべてのPIC16/17ファミリのデバイスは、パワーオン・リセット、選択可能なウォッチドッグ・タイマとコード・プロテクション、高いI/O電流容量を持っています。  
すべてのPIC16CXXファミリのデバイスは、クロック・ピンRB6とデータ・ピンRB7を持ったシリアル・プログラミングを使っています。

## 2.0 PIC16C7Xデバイスの種類

色々な周波数範囲、パッケージの種類が利用できません。アプリケーションと量産の必要に応じて、このデータシートの最後にあるPIC16C7X製品識別システム情報により適当なデバイスを選択できます。ご注文の際は、このデータシートのページを使って正しい製品番号をご指定ください。

PIC16C7Xファミリは、デバイス型番で示されるように2つのデバイス・タイプがあります。

### 1. Cタイプ(例えばPIC16C74)

これらのデバイスはEPROMタイプのメモリで、標準の電圧レンジで動きます。

### 2. LCタイプ(例えばPIC16LC74)

これらのデバイスはEPROMタイプのメモリで、拡張した電圧レンジで動きます。

## 2.1 UV消去デバイス

紫外線消去タイプ(窓付き)は、CERDIPパッケージで提供されており、試作開発用とパイロットランに最適です。

紫外線消去タイプは消去することができ、どのコンフィギュレーションモードにも再プログラムすることができます。

**紫外線消去タイプで、コードプロテクトはサポートされておりません。また、プロテクト後のコード変更はできませんので注意して下さい。**

Microchip社のPICSTART™とPRO MATE™プログラマがPIC16C7Xをサポートします。サードパーティのプログラマも利用できます。詳細についてはMicrochip社

サードパーティガイドを参照してください。

## 2.2 ワンタイムプログラマブル(OTP)デバイス

OTPデバイスは、小規模のアプリケーションで頻繁なコード更新が必要なお客様に有益です。

プラスチックパッケージのOTPデバイスは、1度のみプログラムするユーザに限られます。プログラムメモリに加え、コンフィギュレーションワードもプログラムする必要があります。

## 2.3 クイック・ターンアラウンド・プロダクション(QTP)デバイス

Microchip社は、工場生産注文のためのQTPプログラミングサービスを行なっています。このサービスは、量産数量が多く、自社工場でプログラムを行いたくなく、かつ、コードパターンが安定しているユーザが利用できます。受注可能数量については弊社製品取扱店にお問い合わせ下さい。このデバイスはOTPデバイスと同等ですが、すべてのEPROMロケーションとコンフィギュレーションワードが工場でプログラムされます。実際のコードと試作の検査手順は、出荷前に行なわれます。

## 2.4 シリアル・クイック・ターンアラウンド・プロダクション(SQTP<sup>SM</sup>)デバイス

Microchip社は、各デバイスのシリアル番号を異なったロケーションにプログラムする特定ユーザに対する特殊なプログラミングサービスを行なっています。このシリアル番号には、ランダム/擬似ランダム/シーケンシャル番号があります。シリアルプログラミングにより、各デバイスがエントリコードやパスワードやID番号として使用できる番号を持つことができます。

# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。

**PIC16C7X** <http://www.microchip.com> より最新 ERRATA シートを入手しご参照下さい。

---

### 3.0 アーキテクチャの概要

高性能 PIC16CXX のファミリには、RISC 型マイクロプロセッサに見られる多くのアーキテクチャの特徴があります。まず、PIC16CXX はプログラムとデータが別のバスを使い異なったメモリからアクセスされるハーバードアーキテクチャを使っています。これによって、プログラムとデータが同じバスを使った同じメモリからフェッチされる伝統的なフォンノイマンアーキテクチャよりも帯域幅が改善されています。さらにプログラムとデータのバスを分離することにより、8 ビット幅のデータワードと異なるサイズの命令が可能になります。命令オペコードは、すべてのシングルワード命令を持つことができる14ビット幅です。14ビット幅のプログラムメモリアクセスバスは、14ビット命令をシングルサイクルでフェッチします。2層のパイプラインは、命令のフェッチと実行をオーバーラップします(図3-1参照)。したがって、すべての命令(35)がプログラム分岐を除きシングルサイクル(200ns @20MHz)で実行されます。

下記の表は、PIC16C7X 各デバイスのプログラムメモリ(EEPROM)とデータメモリ(RAM)のリストです。

デバイス	プログラムメモリ	データメモリ
PIC16C710	512 x 14	36 x 8
PIC16C71	1K x 14	36 x 8
PIC16C711	1K x 14	68 x 8
PIC16C72	2K x 14	128 x 8
PIC16C73	4K x 14	192 x 8
PIC16C73A	4K x 14	192 x 8
PIC16C74	4K x 14	192 x 8
PIC16C74A	4K x 14	192 x 8

PIC16CXX は、そのレジスタファイルやデータメモリを直接的または間接的にアドレスできます。プログラムカウンタを含むすべての特殊機能レジスタは、データメモリに配置されます。PIC16CXXは直交的(対称的)な命令を持っており、どのようなアドレッシングモードを使用しても全てのレジスタ上での実行が可能になります。このような対称的な性質と“特定の制約条件”がないために、簡単で効率的な PIC16CXX でのプログラミングが可能です。さらに、学習効率が極めて向上します。

PIC16CXX デバイスには、8ビットALUとワーキングレジスタが内蔵されています。ALUは、汎用演算ユニットです。これによってワーキングレジスタデータと全てのレジスタファイルの間での演算とブール機能が実行できます。

ALUは8ビット幅で、加算、減算、シフトと論理演算が可能です。指示がない限り、演算の実行は2の補数で行なわれます。2個のオペランドの命令では、一般的に一つのオペランドはワーキングレジスタ(Wレジスタ)で、もう一つはファイルレジスタ、または、即値定数です。1個のオペランド命令では、Wレジスタかファイルレジスタのどちらかです。

Wレジスタは、ALUの実行に使われる8ビットのワーキングレジスタで、アドレス指定できるレジスタではありません。

実行される命令によっては、ALUがSTATUSレジスタのキャリ(C)、ディジット・キャリ(DC)、ゼロ(Z)の各ビットの値に影響を与えます。減算命令では、CとDCのビットは、borrowビットとdigit borrowビットとして扱われます。一例としてSUBLWとSUBWF命令をご覧ください。

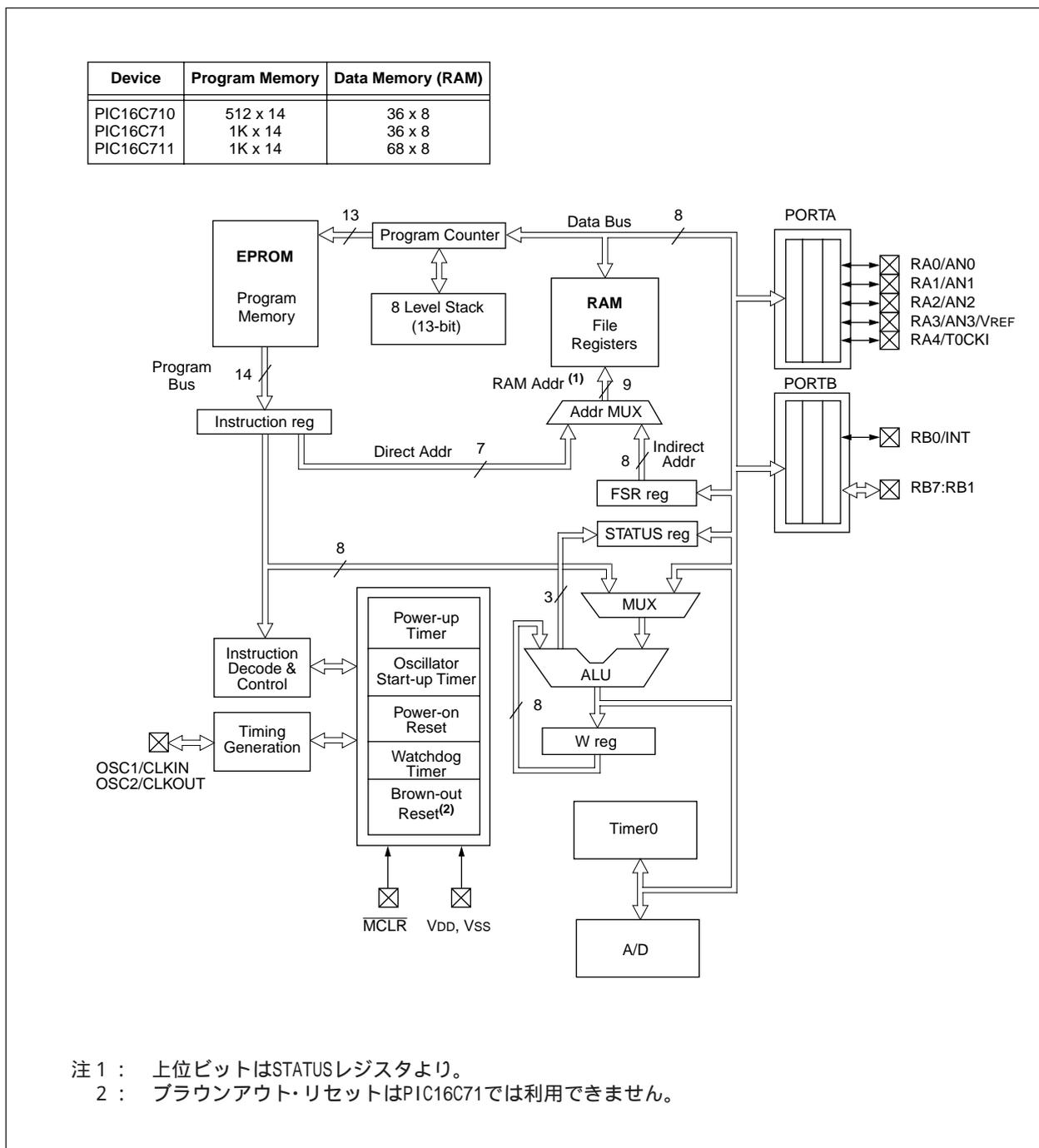
# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。

## PIC16C7X

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

図3-1: PIC16C710/71/711 ブロック図

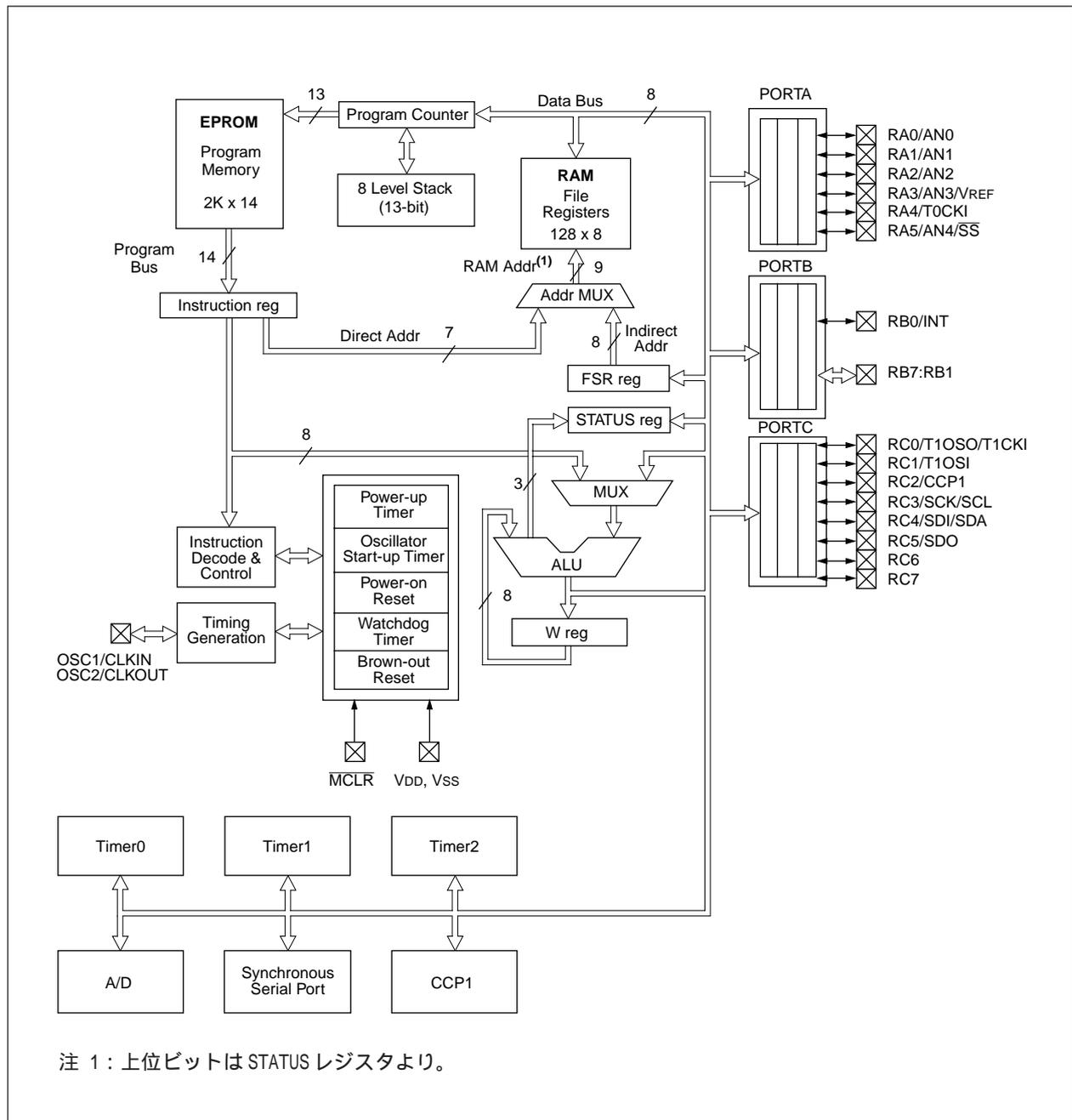


# 暫定版

英語最新版データシートと併用しご利用下さい。  
 AC/DC仕様は最新英文データシートをご参照下さい。  
<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

図3-2: PIC16C72 ブロック図



# 暫定版

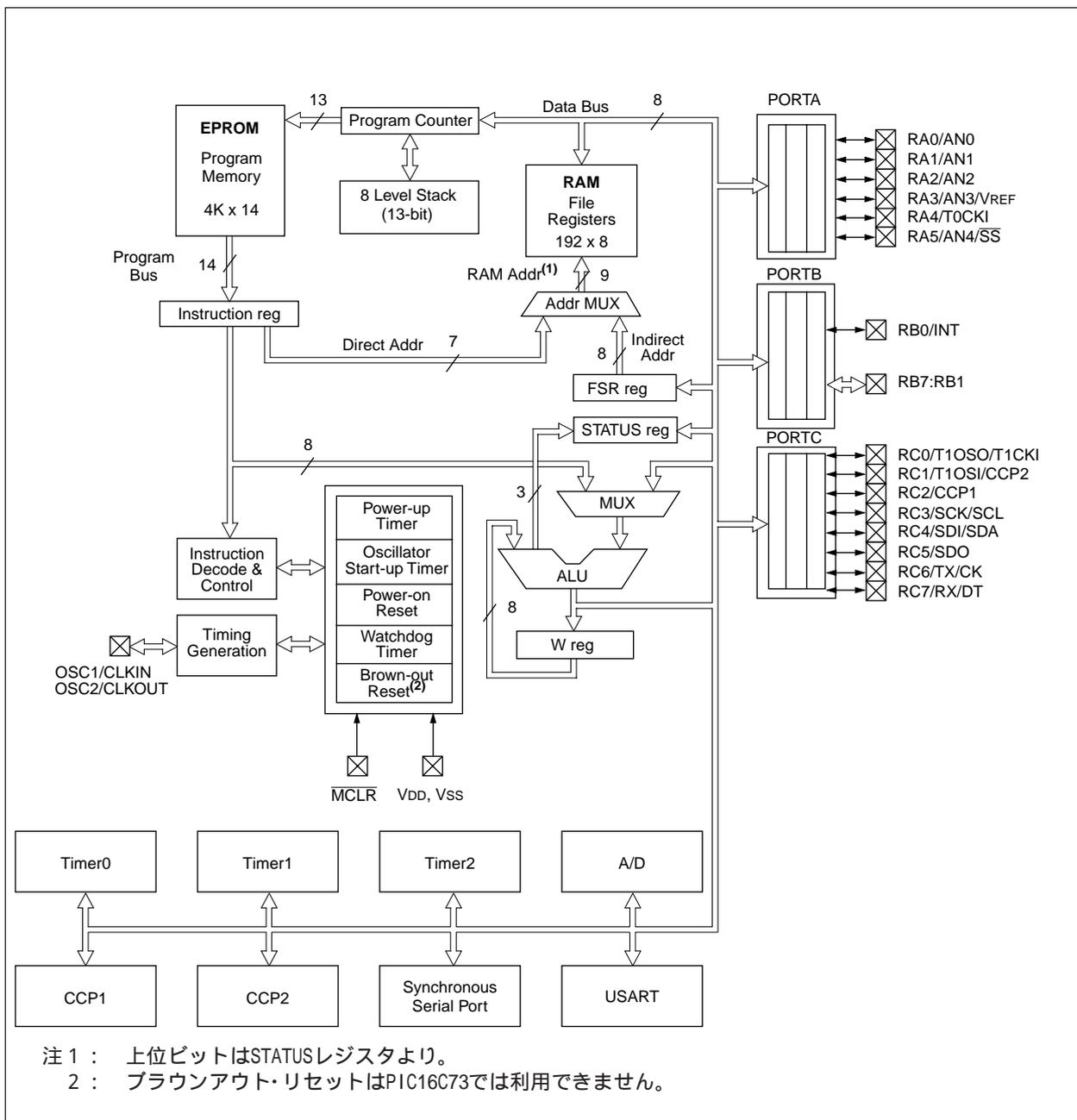
英語最新版データシートと併用しご利用下さい。

AC/DC仕様は最新英文データシートをご参照下さい。

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

図3-3: PIC16C73/73A ブロック図

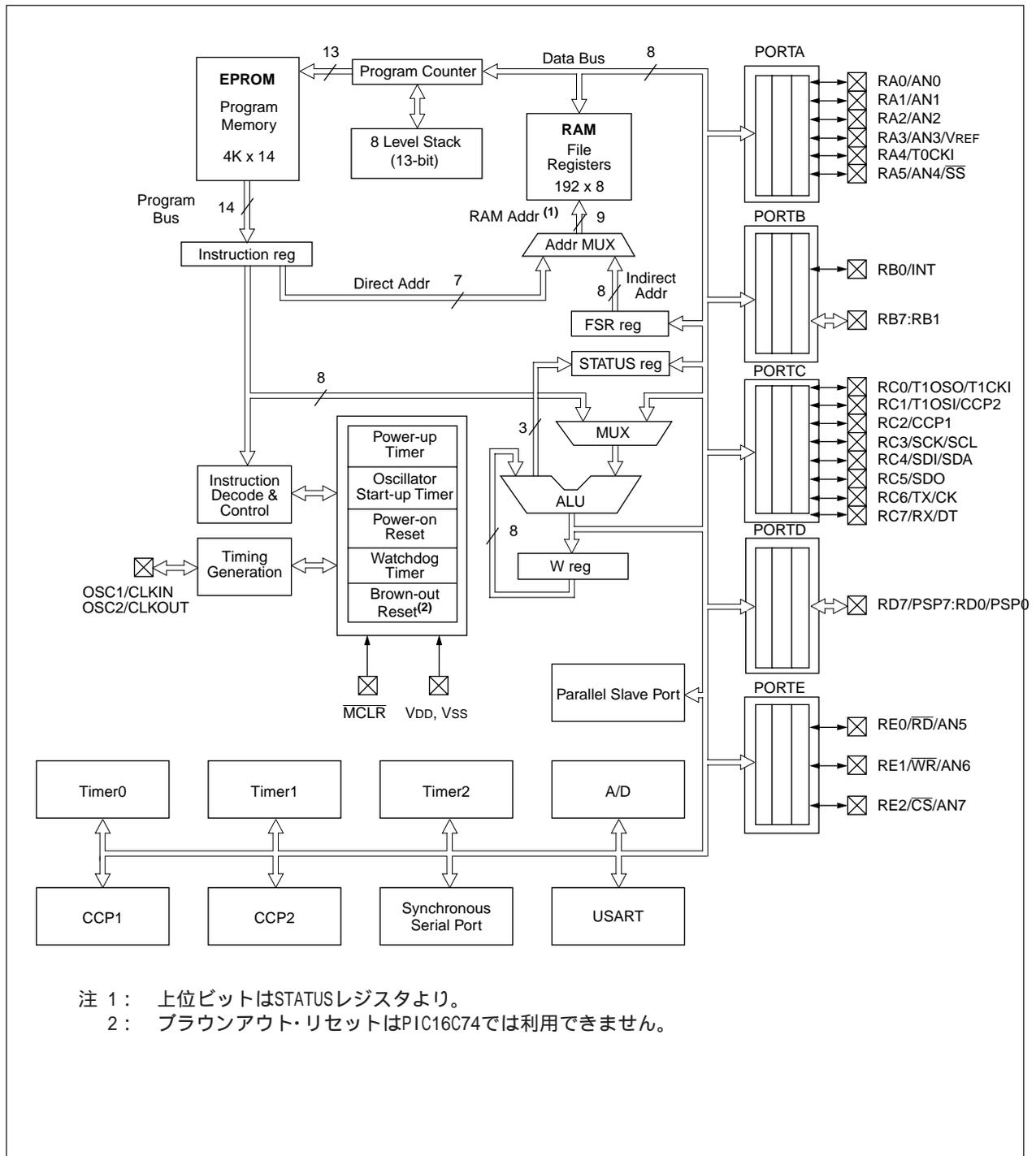


# 暫定版

英語最新版データシートと併用しご利用下さい。  
 AC/DC仕様は最新英文データシートをご参照下さい。  
<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

図3-4: PIC16C74/74A ブロック図



注 1: 上位ビットはSTATUSレジスタより。  
 注 2: ブラウンアウト・リセットはPIC16C74では利用できません。

# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。

## PIC16C7X

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

表3-1: PIC16C710/711ピンアウトの説明

ピンの名称	DIP Pin#	SSOP Pin#	SOIC Pin#	I/O/P Type	Buffer Type	説明
OSC1/CLKIN	16	18	16	I	ST/CMOS <sup>(3)</sup>	オシレータ・クリスタル入力/外部クロック・ソース入力
OSC2/CLKOUT	15	17	15	O	-	オシレータ・クリスタル出力。クリスタル・オシレータ・モードでクリスタルあるいはレゾネータと接続。RCモードでは、OSC2ピンは、OSC1の1/4の周波数を持つCLKOUTを出力し、命令サイクルの割合を表示。
MCLR/V <sub>PP</sub>	4	4	4	I/P	ST	マスタ・クリア(リセット)入力/プログラム電圧入力。このピンはデバイスに対してアクティブ・ロー・リセット。
RA0/AN0	17	19	17	I/O	TTL	PORTAは双方向 I/O ポート。 アナログ入力0
RA1/AN1	18	20	18	I/O	TTL	アナログ入力1
RA2/AN2	1	1	1	I/O	TTL	アナログ入力2
RA3/AN3/V <sub>REF</sub>	2	2	2	I/O	TTL	アナログ入力3/V <sub>REF</sub>
RA4/TOCKI	3	3	3	I/O	ST	タイマ0モジュールへのクロック入力も選択可能。 出力はオープン・ドレイン・タイプ。
RB0/INT	6	7	6	I/O	TTL/ST <sup>(1)</sup>	PORTBは双方向 I/O ポート。PORTBのすべての入力は内部ウィーク・プルアップとしてソフトウェアでプログラム可能。 RB0/INTは外部割込みピンとしても選択可能。
RB1	7	8	7	I/O	TTL	
RB2	8	9	8	I/O	TTL	
RB3	9	10	9	I/O	TTL	
RB4	10	11	10	I/O	TTL	レベル変化で割込み。
RB5	11	12	11	I/O	TTL	レベル変化で割込み。
RB6	12	13	12	I/O	TTL/ST <sup>(2)</sup>	レベル変化で割込み。シリアル・プログラム・クロック。
RB7	13	14	13	I/O	TTL/ST <sup>(2)</sup>	レベル変化で割込み。シリアル・プログラム・データ。
V <sub>SS</sub>	5	4,6	5	P	-	ロジック・ピンおよびI/Oピン用のグランド。
V <sub>DD</sub>	14	15,16	14	P	-	ロジック・ピンおよびI/Oピン用の電源。

凡例: I=入力 O=出力 I/O=入力/出力 P=電源  
- =未使用 TTL=TTL入力 ST=シュミット・トリガ入力

- 注1: このバッファは外部入力として設定された場合はシュミット・トリガ入力となります。  
2: このバッファはシリアル・プログラム・モードとして使用された場合はシュミット・トリガ入力となります。  
3: このバッファはRCオシレータ・モードおよびCMOS入力として設定された場合はシュミット・トリガ入力となります。

# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。  
<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

表3-2: PIC16C71ピンアウトの説明

ピンの名称	DIP Pin#	SOIC Pin#	I/O/P Type	Buffer Type	説明
OSC1/CLKIN	16	16	I	ST/CMOS <sup>(3)</sup>	オシレータ・クリスタル入力/外部クロック・ソース入力
OSC2/CLKOUT	15	15	O	-	オシレータ・クリスタル出力。クリスタル・オシレータ・モードでクリスタルあるいはレゾネータと接続。RCモードでは、OSC2ピンは、OSC1の1/4の周波数を持つCLKOUTを出力し、命令サイクルの割合を表示。
MCLR/V <sub>pp</sub>	4	4	I/P	ST	マスタ・クリア(リセット)入力/プログラム電圧入力。このピンはデバイスに対してアクティブ・ロー・リセット。
RA0/AN0	17	17	I/O	TTL	PORTAは双方向I/Oポート。 アナログ入力0
RA1/AN1	18	18	I/O	TTL	アナログ入力1
RA2/AN2	1	1	I/O	TTL	アナログ入力2
RA3/AN3/V <sub>REF</sub>	2	2	I/O	TTL	アナログ入力3/V <sub>REF</sub>
RA4/TOCKI	3	3	I/O	ST	タイマ0モジュールへのクロック入力も選択可能。出力はオープン・ドレイン・タイプ。
RB0/INT	6	6	I/O	TTL/ST <sup>(1)</sup>	PORTBは双方向I/Oポート。PORTBのすべての入力は内部ウィーク・プルアップとしてソフトウェアでプログラム可能。 RB0/INTは外部割込みピンとしても選択可能。
RB1	7	7	I/O	TTL	
RB2	8	8	I/O	TTL	
RB3	9	9	I/O	TTL	
RB4	10	10	I/O	TTL	レベル変化で割込み。
RB5	11	11	I/O	TTL	レベル変化で割込み。
RB6	12	12	I/O	TTL/ST <sup>(2)</sup>	レベル変化で割込み。シリアル・プログラム・クロック。
RB7	13	13	I/O	TTL/ST <sup>(2)</sup>	レベル変化で割込み。シリアル・プログラム・データ。
V <sub>SS</sub>	5	5	P	-	ロジック・ピンおよびI/Oピン用のグランド。
V <sub>DD</sub>	14	14	P	-	ロジック・ピンおよびI/Oピン用の電源。

凡例: I=入力 O=出力 I/O=入力/出力 P=電源  
- =未使用 TTL=TTL入力 ST=シュミット・トリガ入力

- 注1: このバッファは外部入力として設定された場合はシュミット・トリガ入力となります。  
2: このバッファはシリアル・プログラム・モードとして使用された場合はシュミット・トリガ入力となります。  
3: このバッファはRCオシレータ・モードおよびCMOS入力として設定された場合はシュミット・トリガ入力となります。

# 暫定版

英語最新版データシートと併用しご利用下さい。

AC/DC仕様は最新英文データシートをご参照下さい。

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

表3-3: PIC16C72ピンアウトの説明

ピンの名称	DIP Pin#	SSOP Pin#	SOIC Pin#	I/O/P Type	Buffer Type	説明
OSC1/CLKIN	9	9	9	I	ST/CMOS <sup>(2)</sup>	オシレータ・クリスタル入力/外部クロック・ソース入力
OSC2/CLKOUT	10	10	10	O	-	オシレータ・クリスタル出力。クリスタル・オシレータ・モードでクリスタルあるいはレゾネータと接続。RCモードでは、OSC2ピンは、OSC1の1/4の周波数を持つCLKOUTを出力し、命令サイクルの割合を表示。
MCLR/V <sub>PP</sub>	1	1	1	I/P	ST	マスタ・クリア(リセット)入力/プログラム電圧入力。このピンはデバイスに対してアクティブ・ロー・リセット。
RA0/AN0	2	2	2	I/O	TTL	PORTAは双方向I/Oポート。 アナログ入力0
RA1/AN1	3	3	3	I/O	TTL	アナログ入力1
RA2/AN2	4	4	4	I/O	TTL	アナログ入力2
RA3/AN3/V <sub>REF</sub>	5	5	5	I/O	TTL	アナログ入力3/V <sub>REF</sub>
RA4/T0CKI	6	6	6	I/O	ST	タイマ0モジュールへのクロック入力も選択可能。 出力はオープン・ドレイン・タイプ。
RA5/AN4/SS	7	7	7	I/O	TTL	アナログ入力4は、同期シリアル・ポートに対するスレーブ選択も可能。
RB0/INT	21	21	21	I/O	TTL/ST <sup>(1)</sup>	PORTBは双方向I/Oポート。PORTBのすべての入力は内部ウィーク・プルアップとしてソフトウェアでプログラム可能。 RB0/INTは外部割込みピンとしても選択可能
RB1	22	22	22	I/O	TTL	
RB2	23	23	23	I/O	TTL	
RB3	24	24	24	I/O	TTL	
RB4	25	25	25	I/O	TTL	レベル変化で割込み。
RB5	26	26	26	I/O	TTL	レベル変化で割込み。
RB6	27	27	27	I/O	TTL/ST <sup>(2)</sup>	レベル変化で割込み。シリアル・プログラム・クロック。
RB7	28	28	28	I/O	TTL/ST <sup>(2)</sup>	レベル変化で割込み。シリアル・プログラム・データ。
RC0/T10S0/-T1CKI	11	11	11	I/O	ST	PORTCは双方向I/Oポート。 RC0/T10S0/T1CKIはタイマ1オシレータ出力/タイマ1クロック入力としても選択可能。
RC1/T10S1	12	12	12	I/O	ST	RC1/T10S1/CCP2はタイマ1オシレータ入力またはキャプチャ2入力/コンペア2出力/PWM2出力としても選択可能。
RC2/CCP1	13	13	13	I/O	ST	RC2/CCP1はキャプチャ1入力/コンペア1出力/PWM1出力としても選択可能。
RC3/SCK/SCL	14	14	14	I/O	ST	RC3/SCK/SCLはSPIとI <sup>2</sup> Cモード両方に対して同期シリアル・クロック入力/出力としても選択可能。
RC4/SDI/SDA	15	15	15	I/O	ST	RC4/SDI/SDAは(SPIモードで)SPIデータ入力、あるいは(I <sup>2</sup> Cモードで)データI/Oとしても選択可能。
RC5/SDO	16	16	16	I/O	ST	RC5/SDOは(SPIモードで)SPIデータ出力としても選択可能。
RC6	17	17	17	I/O	ST	
RC7	18	18	18	I/O	ST	
V <sub>SS</sub>	8, 19	8, 19	8, 19	P	-	ロジック・ピンおよびI/Oピン用のグラウンド。
V <sub>DD</sub>	20	20	20	P	-	ロジック・ピンおよびI/Oピン用の電源。

凡例: I=入力 O=出力 I/O=入力/出力 P=電源  
- =未使用 TTL=TTL入力 ST=シュミット・トリガ入力

- 注 1: このバッファは外部入力として設定された場合はシュミット・トリガ入力となります。  
 注 2: このバッファはシリアル・プログラム・モードとして使用された場合はシュミット・トリガ入力となります。  
 注 3: このバッファはRCオシレータ・モードおよびCMOS入力として設定された場合はシュミット・トリガ入力となります。

# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。  
<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

表3-4: PIC16C73/73Aピンアウトの説明

ピンの名称	DIP Pin#	SOIC Pin#	I/O/P Type	Buffer Type	説明
OSC1/CLKIN	9	9	I	ST/CMOS <sup>(3)</sup>	オシレータ・クリスタル入力/外部クロック・ソース入力
OSC2/CLKOUT	10	10	O	-	オシレータ・クリスタル出力。クリスタル・オシレータ・モードでクリスタルあるいはレゾネータと接続。RCモードでは、OSC2ピンは、OSC1の1/4の周波数を持つCLKOUTを出力し、命令サイクルの割合を表示。
MCLR/V <sub>PP</sub>	1	1	I/P	ST	マスタ・クリア(リセット)入力/プログラム電圧入力。このピンはデバイスに対してアクティブ・ロー・リセット。
RA0/AN0	2	2	I/O	TTL	PORTAは双方向I/Oポート。 アナログ入力0
RA1/AN1	3	3	I/O	TTL	アナログ入力1
RA2/AN2	4	4	I/O	TTL	アナログ入力2
RA3/AN3/V <sub>REF</sub>	5	5	I/O	TTL	アナログ入力3/V <sub>REF</sub>
RA4/T0CKI	6	6	I/O	ST	タイマ0モジュールへのクロック入力も選択可能。 出力はオープン・ドレイン・タイプ。
RA5/AN4/SS	7	7	I/O	TTL	アナログ入力4は、同期シリアル・ポートに対するスレーブ選択も可能。
RB0/INT	21	21	I/O	TTL/ST <sup>(1)</sup>	PORTBは双方向I/Oポート。PORTBはすべての入力上で内部ウィーク・プルアップとしてソフトウェアでプログラム可能。 RB0/INTは外部割込みピンとしても選択可能
RB1	22	22	I/O	TTL	
RB2	23	23	I/O	TTL	
RB3	24	24	I/O	TTL	
RB4	25	25	I/O	TTL	レベル変化で割込み。
RB5	26	26	I/O	TTL	レベル変化で割込み。
RB6	27	27	I/O	TTL/ST <sup>(2)</sup>	レベル変化で割込み。シリアル・プログラム・クロック。
RB7	28	28	I/O	TTL/ST <sup>(2)</sup>	レベル変化で割込み。シリアル・プログラム・データ。
RC0/T10S0/T1CKI	11	11	I/O	ST	PORTCは双方向I/Oポート。 RC0/T10S0/T1CKIはタイマ1オシレータ出力/タイマ1クロック入力としても選択可能。
RC1/T10SI	12	12	I/O	ST	RC1/T10SI/CCP2はタイマ1オシレータ入力またはキャプチャ2入力/コンペア2出力/PWM2出力としても選択可能。
RC2/CCP1	13	13	I/O	ST	RC2/CCP1はキャプチャ1入力/コンペア1出力/PWM1出力としても選択可能。
RC3/SCK/SCL	14	14	I/O	ST	RC3/SCK/SCLはSPIとI <sup>2</sup> Cモード両方に対して同期シリアル・クロック入力/出力としても選択可能。
RC4/SDI/SDA	15	15	I/O	ST	RC4/SDI/SDAは(SPIモードで)SPIデータ入力、あるいは(I <sup>2</sup> Cモードで)データI/Oとしても選択可能。
RC5/SDO	16	16	I/O	ST	RC5/SDOは(SPIモードで)SPIデータ出力としても選択可能。
RC6/TX/CK	17	17	I/O	ST	RC6/TX/CKは非同期トランスミットあるいはUSART同期クロックとしても選択可能。
RC7/RX/DT	18	18	I/O	ST	RC7/RX/DTは非同期レシーブあるいはUSART同期データとしても選択可能。
V <sub>SS</sub>	8, 19	8, 19	P	-	ロジック・ピンおよびI/Oピン用のグラウンド。
V <sub>DD</sub>	20	20	P	-	ロジック・ピンおよびI/Oピン用の電源。

凡例: I=入力 O=出力 I/O=入力/出力 P=電源  
- =未使用 TTL=TTL入力 ST=シュミット・トリガ入力

- 注 1: このバッファは外部入力として設定された場合はシュミット・トリガ入力となります。  
注 2: このバッファはシリアル・プログラム・モードとして使用された場合はシュミット・トリガ入力となります。  
注 3: このバッファはRCオシレータ・モードおよびCMOS入力として設定された場合はシュミット・トリガ入力となります。

# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。

## PIC16C7X

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

表3-4： PIC16C74/74Aピンアウトの説明

ピンの名称	DIP Pin#	PLCC Pin#	QFP Pin#	I/O/P Type	Buffer Type	説明
OSC1/CLKIN	13	14	30	I	ST/CMOS <sup>(3)</sup>	オシレータ・クリスタル入力/外部クロック・ソース入力
OSC2/CLKOUT	14	15	31	O	-	オシレータ・クリスタル出力。クリスタル・オシレータ・モードでクリスタルあるいはレゾネータと接続。RCモードでは、OSC2ピンは、OSC1の1/4の周波数を持つCLKOUTを出力し、命令サイクルの割合を表示。
---- MCLR/V <sub>pp</sub>	1	2	18	I/P	ST	マスタ・クリア(リセット)入力/プログラム電圧入力。このピンはデバイスに対してアクティブ・ロー・リセット。
RA0/AN0	2	3	19	I/O	TTL	PORTAは双方向I/Oポート。 アナログ入力0
RA1/AN1	3	4	20	I/O	TTL	アナログ入力1
RA2/AN2	4	5	21	I/O	TTL	アナログ入力2
RA3/AN3/V <sub>REF</sub>	5	6	22	I/O	TTL	アナログ入力3/V <sub>REF</sub>
RA4/T0CKI	6	7	23	I/O	ST	タイマ0モジュールへのクロック入力も選択可能。出力はオープン・ドレイン・タイプ。
RA5/AN4/SS	7	8	24	I/O	TTL	アナログ入力4は、同期シリアル・ポートに対するスレーブ選択も可能。
RB0/INT	33	36	8	I/O	TTL/ST <sup>(1)</sup>	PORTBは双方向I/Oポート。PORTBはすべての入力上で内部ウィーク・プルアップとしてソフトウェアでプログラム可能。 RB0/INTは外部割込みピンとしても選択可能。
RB1	34	37	9	I/O	TTL	
RB2	35	38	10	I/O	TTL	
RB3	36	39	11	I/O	TTL	
RB4	37	41	14	I/O	TTL	レベル変化で割込み。
RB5	38	42	15	I/O	TTL	レベル変化で割込み。
RB6	39	43	16	I/O	TTL/ST <sup>(2)</sup>	レベル変化で割込み。シリアル・プログラム・クロック。
RB7	40	44	17	I/O	TTL/ST <sup>(2)</sup>	レベル変化で割込み。シリアル・プログラム・データ。

凡例： I=入力 O=出力 I/O=入力/出力 P=電源  
- =未使用 TTL=TTL入力 ST=シュミット・トリガ入力

- 注 1： このバッファは外部入力として設定された場合はシュミット・トリガ入力となります。  
注 2： このバッファはシリアル・プログラム・モードとして使用された場合はシュミット・トリガ入力となります。  
注 3： このバッファはRCオシレータ・モードおよびCMOS入力として設定された場合はシュミット・トリガ入力となります。

# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。  
<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

表3-5: PIC16C74/74Aピンアウトの説明 (続き)

ピンの名称	DIP Pin#	PLCC Pin#	QFP Pin#	I/O/P Type	Buffer Type	説明
RC0/T10S0/T1CKI	15	16	32	I/O	ST	PORTCは双方向I/Oポート。 RC0/T10S0/T1CKIはタイマ1オシレータ出力/タイマ1クロック入力としても選択可能。
RC1/T10S1	16	18	35	I/O	ST	RC1/T10S1/CCP2はタイマ1オシレータ入力またはキャプチャ2入力/コンペア2出力/PWM2出力としても選択可能。
RC2/CCP1	17	19	36	I/O	ST	RC2/CCP1はキャプチャ1入力/コンペア1出力/PWM1出力としても選択可能。
RC3/SCK/SCL	18	20	37	I/O	ST	RC3/SCK/SCLはSPIとI <sup>2</sup> Cモード両方に対して同期シリアル・クロック入力/出力としても選択可能。
RC4/SDI/SDA	23	25	42	I/O	ST	RC4/SDI/SDAは(SPIモードで)SPIデータ入力、あるいは(I <sup>2</sup> Cモードで)データI/Oとしても選択可能。
RC5/SDO	24	26	43	I/O	ST	RC5/SDOは(SPIモードで)SPIデータ出力としても選択可能。
RC6/TX/CK	25	27	44	I/O	ST	RC6/TX/CKは非同期トランスミットあるいはUSART同期クロックとしても選択可能。
RC7/RX/DT	26	29	1	I/O	ST	RC7/RX/DTは非同期レシーブあるいはUSART同期データとしても選択可能。
RD0/PSP0	19	21	38	I/O	ST/TTL <sup>(3)</sup>	PORTDは双方向I/Oポート、あるいはマイクロプロセッサ・バスにインターフェイスする場合はパラレル・スレーブ・ポート。
RD1/PSP1	20	22	39	I/O	ST/TTL <sup>(3)</sup>	
RD2/PSP2	21	23	40	I/O	ST/TTL <sup>(3)</sup>	
RD3/PSP3	22	24	41	I/O	ST/TTL <sup>(3)</sup>	
RD4/PSP4	27	30	2	I/O	ST/TTL <sup>(3)</sup>	
RD5/PSP5	28	31	3	I/O	ST/TTL <sup>(3)</sup>	
RD6/PSP6	29	32	4	I/O	ST/TTL <sup>(3)</sup>	
RD7/PSP7	30	33	5	I/O	ST/TTL <sup>(3)</sup>	
RE0/RD/AN5	8	9	25	I/O	ST/TTL <sup>(3)</sup>	PORTEは双方向I/Oポート。 RE0/RD/AN5パラレル・スレーブ・ポートあるいはアナログ入力5用のリード制御
RE1/WR/AN6	9	10	26	I/O	ST/TTL <sup>(3)</sup>	RE1/WR/AN6パラレル・スレーブ・ポートあるいはアナログ入力6用のライト制御
RE2/CS/AN7	10	11	27	I/O	ST/TTL <sup>(3)</sup>	RE2/CS/AN7パラレル・スレーブ・ポートあるいはアナログ入力7用の制御
V <sub>SS</sub>	12-, 31	13-, 34	6-, 29	P	-	ロジック・ピンおよびI/Oピン用の接地基準。
V <sub>DD</sub>	11-, 32	12-, 35	7-, 20	P	-	ロジック・ピンおよびI/Oピン用の電源正極。
NC	-	1-, 17-, 28-, 40	12-, 1-, 3-, 33-, 34	-	-	これらのピンは内部接続されていません。これらのピンは接続しないままにしておかなければなりません。

凡例: I=入力 O=出力 I/O=入力/出力 P=電源  
- =未使用 TTL=TTL入力 ST=シュミット・トリガ入力

- 注 1: このバッファは外部入力として設定された場合はシュミット・トリガ入力となります。  
 注 2: このバッファはシリアル・プログラム・モードとして使用された場合はシュミット・トリガ入力となります。  
 注 3: このバッファは汎用I/Oとして設定された場合にはシュミット・トリガ入力となり、(マイクロプロセッサ・バスにインターフェイスするために)パラレル・スレーブ・ポート・モードで使用された場合はTTL入力となります。

# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。

## PIC16C7X

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

### 3.1 クロック方式/命令サイクル

クロック入力(OSC1から)は、内部で4分周され、Q1、Q2、Q3、Q4と呼ばれる4相のオーバーラップしない矩形波クロックを発生させます。内部で、プログラムカウンタ(PC)はQ1毎にインクリメントされ、命令はプログラムメモリからフェッチされ、Q4で命令レジスタにラッチされます。命令は次のQ1からQ4の間でデコードされ実行されます。クロックと命令の実行フローを図3-5に示します。

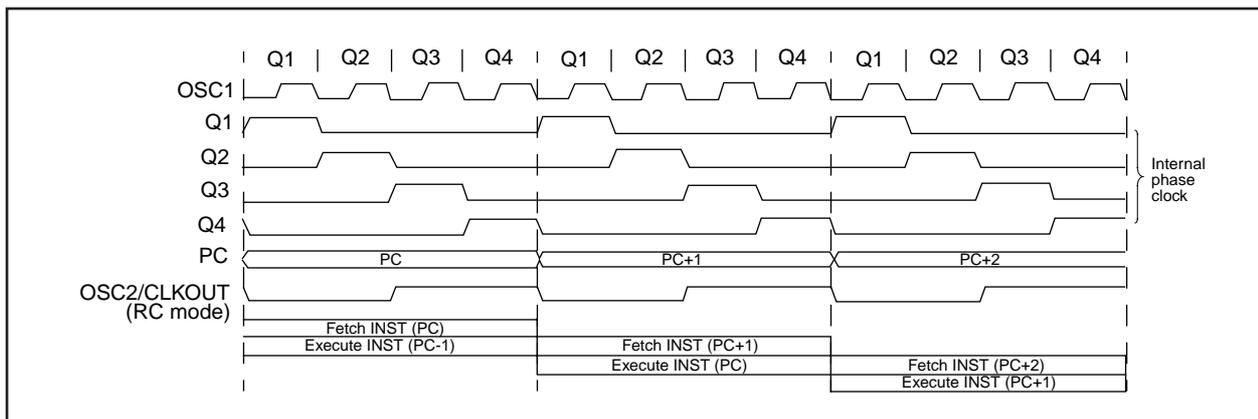
### 3.2 命令のフロー/パイプライン

命令サイクルは、4つのQサイクル(Q1、Q2、Q3、Q4)で構成されています。命令のフェッチと実行は、パイプライン方式で行なわれ、フェッチに1命令サイクル、デコードと実行に1命令サイクルがかかります。しかしパイプラインによって、各命令は効率よく1サイクルで実行されます。命令によってプログラムカウンタを変更する場合(GOTOなどは、命令を完了するために2サイクルが必要です。(例3-1参照)

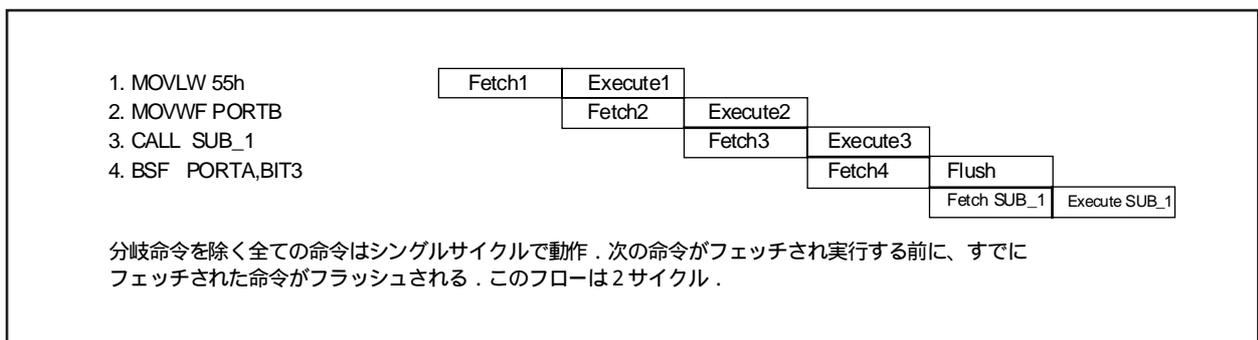
フェッチサイクルはプログラムカウンタ(PC)がQ1でインクリメントすることで始まります。

実行サイクルでは、フェッチされた命令はQ1サイクルで"命令レジスタ"(IR)にラッチされます。この命令はそれからQ2、Q3、Q4サイクルの間でデコードされ、実行されます。データメモリは、Q2の間でリード(オペランドリード)され、Q4の間でライト(デスティネーションライト)されます。

図 3-5: クロック/インストラクション・サイクル



例 3-1: インストラクション・パイプライン・フロー



### 4.0 メモリ構成

適用デバイス							
710	71	711	72	73	73A	74	74A

#### 4.1 プログラムメモリ構成

PIC16C7Xファミリには、8K×14のプログラムメモリ空間をアドレス指定できる13ビットのプログラムカウンタがあります。

PIC16C710は、最初の512×14(0000h-01FFh)のみ物理的にインプリメントされています。PIC16C71/711は最初の1K×14(0000h-03FFh)のみがインプリメントされています。PIC16C72は最初の2K×14(0000h-07FFh)のみがインプリメントされています。PIC16C73、PIC16C73A、PIC16C74、PIC16C74Aは、最初の4K×14(0000h-0FFFh)のみが物理的にインプリメントされています。物理的にインプリメントされたアドレス以上のロケーションにアクセスすると、ラップアラウンドを引き起こします。リセットベクタは0000hに、割込みベクタは0004hにあります。

図4-1: PIC16C710プログラムメモリマップとスタック

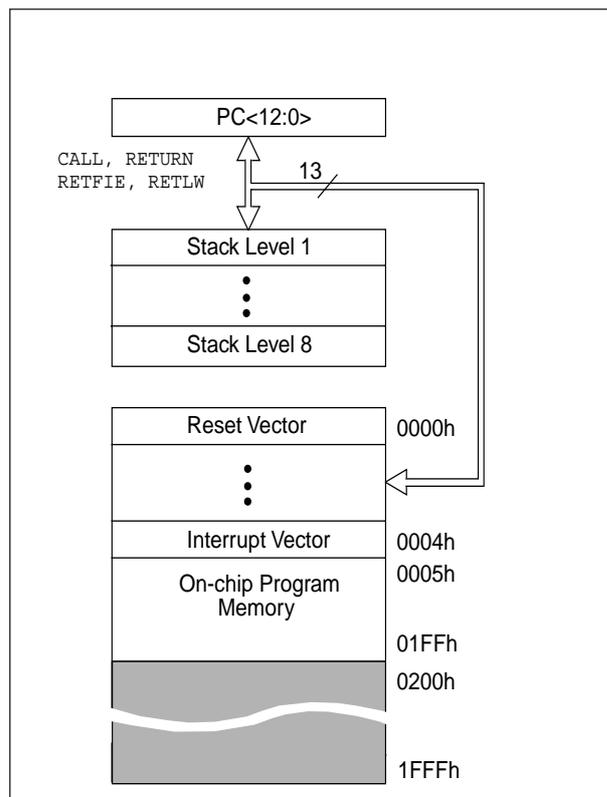
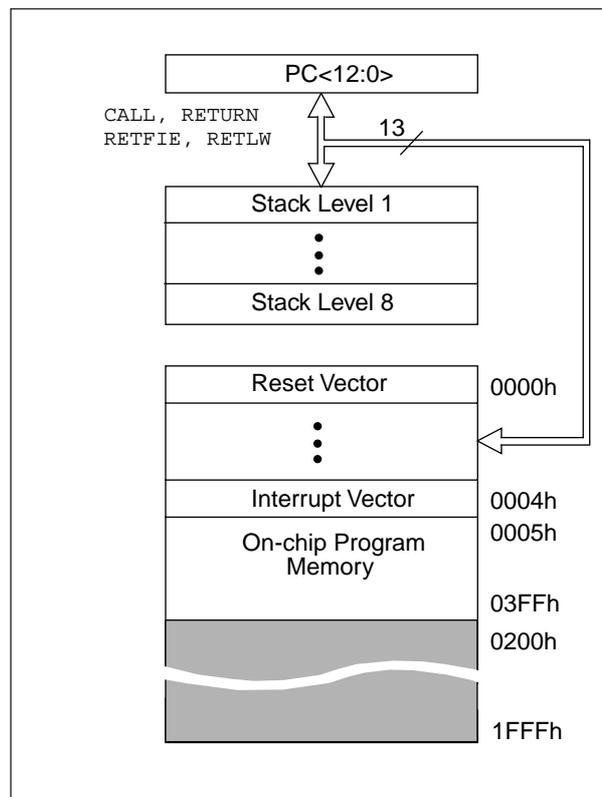
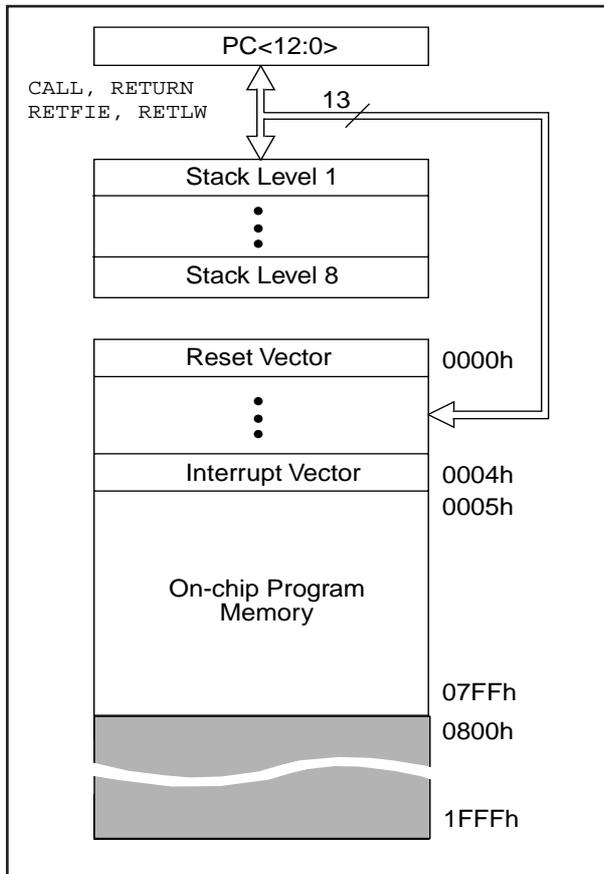


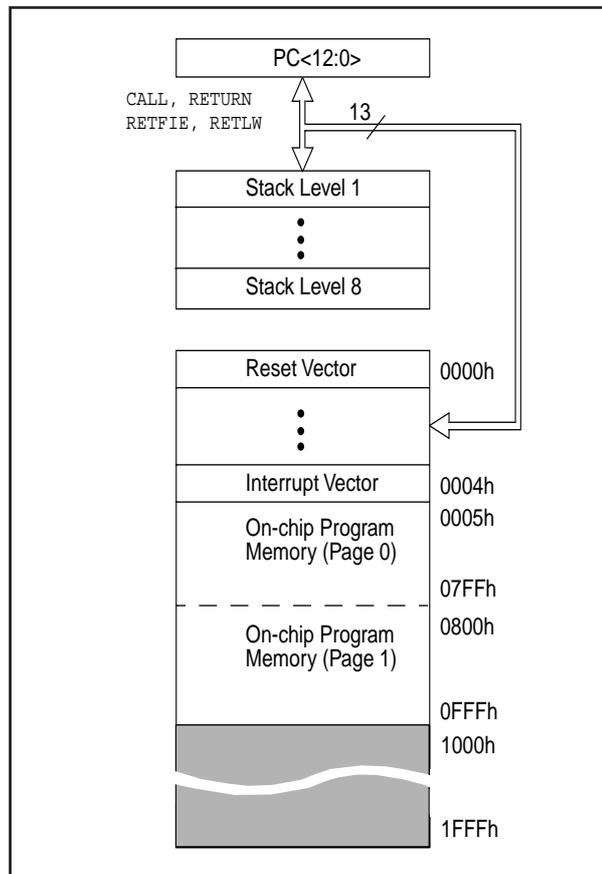
図4-2: PIC16C71/711プログラムメモリマップとスタック



**図4-3 : PIC16C72プログラムメモリマップとスタック**



**図4-4 : PIC16C73/73A/74/74Aプログラムメモリマップとスタック**



## 4.2 データメモリ構成

適用デバイス							
710	71	711	72	73	73A	74	74A

データメモリは、汎用レジスタと特殊機能レジスタを含む2個のバンクで構成されています。ビットRP0は、バンクセレクトビットです。

RP0 (STATUS<5>)=1 Bank1

RP0 (STATUS<5>)=0 Bank0

各バンクは7Fh(128バイト)まで拡張できます。各バンクの低いロケーションは、特殊機能レジスタのためにリザーブされます。特殊機能レジスタ以外の部分はスタティックRAMとしてインプリメントされている汎用レジスタです。バンク0とバンク1は、特殊機能レジスタを含んでいます。使用頻度の高いいくつかの特殊機能レジスタは、プログラムメモリ内の命令数の低減と、より早いアクセスのためにバンク0、バンク1どちらからでもアクセス可能です。

### 4.2.1 汎用レジスタファイル

レジスタファイルは、直接または、ファイル選択レジスタFSRにより間接のどちらからでもアクセスできます(4.5章参照)。

# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。  
<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

図4-5: PIC16C710/71レジスタファイルマップ

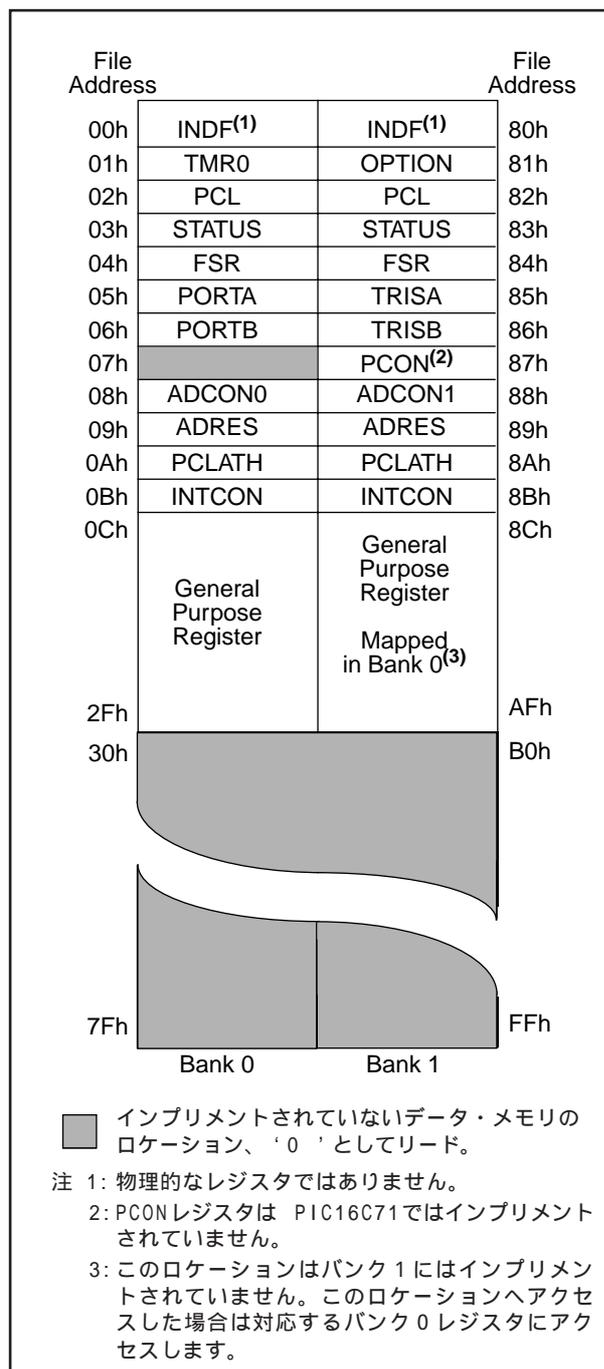
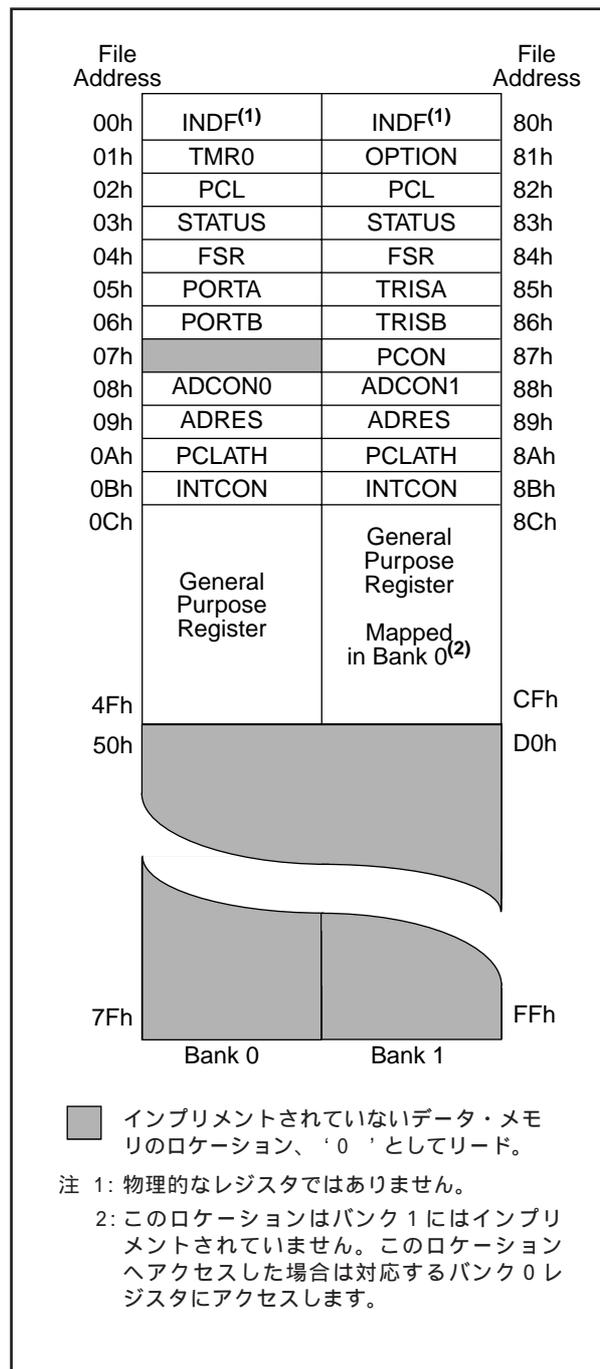


図4-6: PIC16C711レジスタファイルマップ



**図4-7: PIC16C72 レジスタファイルマップ**

File Address			File Address	
00h	INDF(1)	INDF(1)	80h	
01h	TMR0	OPTION	81h	
02h	PCL	PCL	82h	
03h	STATUS	STATUS	83h	
04h	FSR	FSR	84h	
05h	PORTA	TRISA	85h	
06h	PORTB	TRISB	86h	
07h	PORTC	TRISC	87h	
08h			88h	
09h			89h	
0Ah	PCLATH	PCLATH	8Ah	
0Bh	INTCON	INTCON	8Bh	
0Ch	PIR1	PIE1	8Ch	
0Dh			8Dh	
0Eh	TMR1L	PCON	8Eh	
0Fh	TMR1H		8Fh	
10h	T1CON		90h	
11h	TMR2		91h	
12h	T2CON	PR2	92h	
13h	SSPBUF	SSPADD	93h	
14h	SSPCON	SSPSTAT	94h	
15h	CCPR1L		95h	
16h	CCPR1H		96h	
17h	CCP1CON		97h	
18h			98h	
19h			99h	
1Ah			9Ah	
1Bh			9Bh	
1Ch			9Ch	
1Dh			9Dh	
1Eh	ADRES		9Eh	
1Fh	ADCON0	ADCON1	9Fh	
20h	General Purpose Register	General Purpose Register	A0h	
				B0h
				C0h
				D0h
		E0h		
		F0h		
7Fh			FFh	

Bank 0
Bank 1

インプリメントされていないデータ・メモリのロケーション、'0'としてリード。  
 注 1: 物理的なレジスタではありません。

**図4-8: PIC16C73/73A/74/74A レジスタファイルマップ**

File Address			File Address	
00h	INDF(1)	INDF(1)	80h	
01h	TMR0	OPTION	81h	
02h	PCL	PCL	82h	
03h	STATUS	STATUS	83h	
04h	FSR	FSR	84h	
05h	PORTA	TRISA	85h	
06h	PORTB	TRISB	86h	
07h	PORTC	TRISC	87h	
08h	PORTD(2)	TRISD(2)	88h	
09h	PORTE(2)	TRISE(2)	89h	
0Ah	PCLATH	PCLATH	8Ah	
0Bh	INTCON	INTCON	8Bh	
0Ch	PIR1	PIE1	8Ch	
0Dh	PIR2	PIE2	8Dh	
0Eh	TMR1L	PCON	8Eh	
0Fh	TMR1H		8Fh	
10h	T1CON		90h	
11h	TMR2		91h	
12h	T2CON	PR2	92h	
13h	SSPBUF	SSPADD	93h	
14h	SSPCON	SSPSTAT	94h	
15h	CCPR1L		95h	
16h	CCPR1H		96h	
17h	CCP1CON		97h	
18h	RCSTA	TXSTA	98h	
19h	TXREG	SPBRG	99h	
1Ah	RCREG		9Ah	
1Bh	CCPR2L		9Bh	
1Ch	CCPR2H		9Ch	
1Dh	CCP2CON		9Dh	
1Eh	ADRES		9Eh	
1Fh	ADCON0	ADCON1	9Fh	
20h	General Purpose Register	General Purpose Register	A0h	
				B0h
				C0h
				D0h
				E0h
				F0h
7Fh			FFh	

Bank 0
Bank 1

インプリメントされていないデータ・メモリのロケーション、'0'としてリード。  
 注 1: 物理的なレジスタではありません。  
 注 2: これらのレジスタはPIC16C73/73Aでは物理的にインプリメントされていません、'0'としてリード。

# 暫定版

英語最新版データシートと併用ご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。  
<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

### 4.2.2 特殊機能レジスタ

特殊機能レジスタはデバイスの適切な実行を制御するために、CPUと周辺モジュールによって使われるレジスタです。これらのレジスタはスタティックRAMとしてインプリメントされています。

特殊機能レジスタは2つ(コアと周辺)に分類されます。一つはコアのためのレジスタで、この章で説明されています。周辺機能のオペレーションに関連するレジスタは、その周辺機能の章で説明します。

表4-1: PIC16C710/711/711特殊機能レジスタ一覧

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	POR BOR での値	他の リセット での値 (1)
<b>Bank0</b>											
00h(3)	INDF	間接アドレッシング時のリード/ライトレジスタ(物理的には存在しません)								0000 0000	0000 0000
01h	TMRO	TMROモジュールレジスタ								xxxx xxxx	uuuu uuuu
02h(3)	PCL	プログラムカウンタ(PC)の下位バイト								0000 0000	0000 0000
03h(3)	STATUS	IRP(5)	RP1(5)	RPO	T0	PD	Z	DC	C	0001 1xxx	000q quuu
04h(3)	FSR	間接データメモリアドレスリングポインタ								xxxx xxxx	uuuu uuuu
05h	PORTA	-	-	-	ライト時、PORTAデータラッチ:リード時、PORTA端子					---x 0000	---u 0000
06h	PORTB	ライト時、PORTBデータラッチ:リード時、PORTB端子								xxxx xxxx	uuuu uuuu
07h	-	未使用								-	-
08h	ADCON0	ADCS1	ADCS0	(6)	CHS1	CHS0	GO/DONE	ADIF	ADON	00-0 0000	00-0 0000
09h(3)	ADRES	A/D変換結果レジスタ								xxxx xxxx	uuuu uuuu
0Ah(2,3)	PCLATH	-	-	-	プログラムカウンタ上位バイトのライトリッパ					---0 0000	---0 0000
0Bh(3)	INTCON	GIE	ADIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
<b>Bank1</b>											
80h(3)	INDF	間接アドレッシング時のリード/ライトレジスタ(物理的には存在しません)								0000 0000	0000 0000
81h	OPTION	RBPU	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
82h(3)	PCL	プログラムカウンタ(PC)の下位バイト								0000 0000	0000 0000
83h(3)	STATUS	IRP(5)	RP1(5)	RPO	T0	PD	Z	DC	C	0001 1xxx	000q quuu
84h(3)	FSR	間接データメモリアドレスリングポインタ								xxxx xxxx	uuuu uuuu
85h	TRISA	-	-	-	PORTA 入出力切り替えレジスタ					---1 1111	---1 1111
86h	TRISB	PORTB 入出力切り替えレジスタ								1111 1111	1111 1111
87h(3)	PCON	-	-	-	-	-	-	POR	BOR	---- --gq	---- --uu
88h	ADCON1	-	-	-	-	-	-	PCFG1	PCFG0	---- --00	---- --00
89h(3)	ADRES	A/D結果格納レジスタ								xxxx xxxx	uuuu uuuu
8Ah(2,3)	PCLATH	-	-	-	プログラムカウンタ上位バイトのライトリッパ					---0 0000	---0 0000
8Bh(3)	INTCON	GIE	ADIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u

凡例: x=未知、u=不変、q=値は条件による、-=未使用('0'としてリード)。  
網掛け部分は未使用('0'としてリード)。

- 注1: 他の(non power-up)リセットは、MCLRとウォッチドッグタイマリセットを含みます。
- プログラムカウンタの上位バイトは直接アクセスすることはできません。PCLATHはPC<12:8>のホールディングレジスタで、その内容はプログラムカウンタの上位バイトに転送されます。
  - これらのレジスタは、全てのバンクからアドレス可能です。
  - PCONレジスタはPIC16C71では物理的にインプリメントされていません。'0'としてリード。
  - IRPとRP1ビットはPIC16C7Xではリザーブされており、常にこれらのビットはクリアされた状態です。
  - ADCON0のビット5は、PIC16C71用の汎用R/Wビットです。PIC16C710/711に関しては、このビットはインプリメントされていません。'0'としてリード。

# 暫定版

英語最新版データシートと併用ご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。

## PIC16C7X

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

表4-2: PIC16C72特殊機能レジスタ一覧

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR BOR での値	他の リセット での値(3)
<b>Bank 0</b>											
00h(1)	INDF	間接アドレッシング時のリード/ライトレジスタ (物理的には存在しません)								0000 0000	0000 0000
01h	TMR0	タイマ0モジュールのレジスタ								xxxx xxxx	uuuu uuuu
02h(1)	PCL	プログラムカウンタ(PC)の最下位バイト								0000 0000	0000 0000
03h(1)	STATUS	IRP(4)	RP1(4)	RPO	T0	PD	Z	DC	C	0001 1xxx	000q quuu
04h(1)	FSR	間接データ・メモリ・アドレス・ポインタ								xxxx xxxx	uuuu uuuu
05h	PORTA	-	-	ライト時PORTAデータラッチ:リード時 PORTAピン						--0x 0000	--0u 0000
06h	PORTB	ライト時PORTB データ・ラッチ:リード時 PORTB ピン								xxxx xxxx	uuuu uuuu
07h	PORTC	ライト時PORTC データ・ラッチ:リード時 PORTC ピン								xxxx xxxx	uuuu uuuu
08h	-	未使用								-	-
09h	-	未使用								-	-
0Ah(1,2)	PCLATH	-	-	-	プログラムカウンタの上位5ビット用のライトラッチ					---0 0000	---0 0000
0Bh(1)	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	-	ADIF	-	-	SSPIF	CCP1IF	TMR2IF	TMR1IF	-0-- 0000	-0-- 0000
0Dh	-	未使用								-	-
0Eh	TMR1L	16ビットTMR1レジスタの下位バイト用保持レジスタ								xxxx xxxx	uuuu uuuu
0Fh	TMR1H	16ビットTMR1レジスタの上位バイト用保持レジスタ								xxxx xxxx	uuuu uuuu
10h	T1CON	-	-	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	--00 0000	--uu uuuu
11h	TMR2	タイマ2モジュールのレジスタ								0000 0000	0000 000
12h	T2CON	-	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
13h	SSPBUF	同期シリアル・ポート受信バッファ/送信レジスタ								xxxx xxxx	uuuu uuuu
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
15h	CCPR1L	キャプチャ/コンペア/PWMレジスタ(LSB)								xxxx xxxx	uuuu uuuu
16h	CCPR1H	キャプチャ/コンペア/PWMレジスタ (MSB)								xxxx xxxx	uuuu uuuu
17h	CCP1CON	-	-	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
18h	-	未使用								-	-
19h	-	未使用								-	-
1Ah	-	未使用								-	-
1Bh	-	未使用								-	-
1Ch	-	未使用								-	-
1Dh	-	未使用								-	-
1Eh	ADRES	A/D結果レジスタ								xxxx xxxx	uuuu uuuu
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	-	ADCON	0000 00-0	0000 00-0

凡例: x = 未知、u = 不変、q = 値は条件による、- = 未使用 ('0' としてリード)。  
網掛け部分は未使用 ('0' としてリード)。

注1: これらのレジスタは、全てのバンクからアドレス可能です。

2: プログラムカウンタの上位バイトは直接アクセスすることはできません。PCLATH は PC<12:8> の ホールディングレジスタで、その内容はプログラムカウンタの上位バイトに転送されます。

3: 他の (non power-up) リセットには、MCLR とウォッチドッグタイマリセットを含みます。

4: IRP と RP1 ビットは、PIC16C7Xではリザーブされており、常にこれらのビットはクリアされた状態です。

# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。  
<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

表4-2: PIC16C72特殊機能レジスタ一覧 (続き)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR BOR での値	他の リセット での値(3)
<b>Bank 1</b>											
80h(1)	INDF	間接アドレッシング時のリード・ライトレジスタ (物理的には存在しません)								0000 0000	0000 0000
81h	OPTION	RBPV	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
82h(1)	PCL	プログラム・カウンタ(PC)の最下位バイト								0000 0000	0000 0000
83h(1)	STATUS	IRP(4)	RP1(4)	RPO	T0	PD	Z	DC	C	0001 1xxx	000q quuu
84h(1)	FSR	間接データ・メモリ・アドレス・ポインタ								xxxx xxxx	uuuu uuuu
85h	TRISA	-	-	PORTAのデータ指示レジスタ						--11 1111	--11 1111
86h	TRISB	PORTBのデータ指示レジスタ								1111 1111	1111 1111
87h	TRISC	PORTCのデータ指示レジスタ								1111 1111	1111 1111
88h	-	未使用								-	-
89h	-	未使用								-	-
8Ah(1,2)	PCLATH	-	-	-	PCの上位5ビット用のライト・バッファ					---0 0000	---0 0000
8Bh(1)	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBF	0000 000x	0000 000u
8Ch	PIE1	-	ADIE	-	-	SSPIE	CCP1IE	TMR2IE	TMR1IE	-0-- 0000	-0-- 0000
8Dh	-	未使用								-	-
8Eh	PCON	-	-	-	-	-	-	POR	BOR	---- --qq	---- --uu
8Fh	-	未使用								-	-
90h	-	未使用								-	-
91h	-	未使用								-	-
92h	PR2	タイマ2の周期レジスタ								1111 1111	1111 1111
93h	SSPADD	同期シリアルポート(12Cモード)のアドレスレジスタ								0000 0000	0000 0000
94h	SSPSTAT	-	-	D/A	P	S	R/W	UA	BF	--00 0000	--00 0000
95h	-	未使用								-	-
96h	-	未使用								-	-
97h	-	未使用								-	-
98h	-	未使用								-	-
99h	-	未使用								-	-
9Ah	-	未使用								-	-
9Bh	-	未使用								-	-
9Ch	-	未使用								-	-
9Dh	-	未使用								-	-
9Eh	-	未使用								-	-
9Fh	ADCON1	-	-	-	-	-	PCFG2	PCFG1	PCFG0	---- -000	---- -000

凡例: x = 未知、u = 不変、q = 値は条件による、- = 未使用 ('0' としてリード)。  
網掛け部分は未使用 ('0' としてリード)。

注1: これらのレジスタは、全てのバンクからアドレス可能です。

- 2: プログラムカウンタの上位バイトは直接アクセスすることはできません。PCLATH は PC<12:8>のホールディングレジスタで、その内容はプログラムカウンタの上位バイトに転送されます。
- 3: 他の (non power-up) リセットには、 $\overline{\text{MCLR}}$  とウォッチドッグタイマリセットを含みます。
- 4: IRP と RP1 ビットは、PIC16C7Xではリザーブされており、常にこれらのビットはクリアされた状態です。

# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。

## PIC16C7X

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

表4-3: PIC16C73/73A/74/74A 特殊機能レジスタ一覧

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR BOR での値	他の リセット での値(2)
<b>Bank0</b>											
00h (4)	INDF	間接アドレッシング時のリード/ライトレジスタ (物理的には存在しません)								0000 0000	0000 0000
01h	TMR0	タイマ0モジュールのレジスタ								xxxx xxxx	uuuu uuuu
02h (4)	PCL	プログラム・カウンタ(PC)の最下位バイト								0000 0000	0000 0000
03h (4)	STATUS	IRP (7)	RP1 (7)	RPO	$\overline{TO}$	PD	Z	DC	C	0001 1xxx	000q quuu
04h (1)	FSR	間接データメモリアドレスポインタ								xxxx xxxx	uuuu uuuu
05h	PORTA	-	-	ライト時PORTA データラッチ: リード時 PORTA ピン						--0x 0000	--0u 0000
06h	PORTB	ライト時PORTB データラッチ: リード時 PORTB ピン								xxxx xxxx	uuuu uuuu
07h	PORTC	ライト時PORTC データラッチ: リード時 PORTC ピン								xxxx xxxx	uuuu uuuu
08h	PORTD	ライト時PORTD データラッチ: リード時 PORTD ピン								xxxx xxxx	uuuu uuuu
09h	PORTE	-	-	-	-	-	RE2	RE1	RE0	---- -000	---- -000
0Ah (1,2)	PCLATH	-	-	-	プログラムカウンタの上位5ビット用のライトバッファ					---0 0000	---0 0000
0Bh (1)	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSP1F (3)	ADIF	RCIF	TXIF	SSPIF	CCP1F	TMR2IF	TMR1IF	0000 0000	0000 0000
0Dh	PIR2	-	-	-	-	-	-	-	CCP2IF	---- -000	---- -000
0Eh	TMR1L	16ビットTMR1レジスタの下位バイト用保持レジスタ								xxxx xxxx	uuuu uuuu
0Fh	TMR1H	16ビットTMR1レジスタの上位バイト用保持レジスタ								xxxx xxxx	uuuu uuuu
10h	T1CON	-	-	T1CKPS1	T1CKPS0	T10SCEN	T1SYNC	TMR1CS	TMR1ON	--00 0000	--uu uuuu
11h	TMR2	タイマ2モジュールのレジスタ								0000 0000	0000 0000
12h	T2CON	-	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
13h	SSPBUF	同期シリアル・ポート受信バッファ/送信レジスタ								xxxx xxxx	uuuu uuuu
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
15h	CCPR1L	キャプチャ/コンペア/PWMレジスタ1 (LSB)								xxxx xxxx	uuuu uuuu
16h	CCPR1H	キャプチャ/コンペア/PWMレジスタ1 (MSB)								xxxx xxxx	uuuu uuuu
17h	CCP1CON	-	-	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	-	FERR	OERR	RX9D	0000 -00x	0000 -00x
19h	TXREG	USART送信データレジスタ								0000 0000	0000 0000
1Ah	RCREG	USART受信データレジスタ								0000 0000	0000 0000
1Bh	CCPR2L	キャプチャ/コンペア/PWMレジスタ2 (LSB)								xxxx xxxx	uuuu uuuu
1Ch	CCPR2H	キャプチャ/コンペア/PWMレジスタ2 (MSB)								xxxx xxxx	uuuu uuuu
1Dh	CCP2CON	-	-	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	--00 0000
1Eh	ADRES	A/D結果レジスタ								xxxx xxxx	uuuu uuuu
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	-	ADON	0000 00-0	0000 00-0

凡例: x = 未知、u = 不変、q = 値は条件による、- = 未使用 ('0'としてリード)。  
網掛け部分は未使用 ('0'としてリード)。

注1: プログラムカウンタの上位バイトは直接アクセスすることはできません。PCLATHはPC<12:8>のホールディングレジスタで、その内容はプログラム・カウンタの上位バイトに転送されます。

2: 他の(non power-up)リセットは、MCLRとウォッチドッグ・タイマ・リセットを含みます。

3: PSP1EとPSP1FビットはPIC16C73/73Aではリザーブされ、常にこれらのビットはクリアされた状態です。

4: これらのレジスタは、全てのバンクからアドレス可能です。

5: PORTDとPORTEはPIC16C73/73Aでは物理的にインプリメントされていません。'0'としてリード。

6: ブラウン-アウトリセットは、PIC16C73またはPIC16C74ではインプリメントされていません。'0'としてリード。

7: IRPとRP1ビットはPIC16C7Xではリザーブされており、常にこれらのビットはクリアされた状態です。

# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。  
<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

表4-3: PIC16C73/73A/74/74A特殊機能レジスタ一覧 (続き)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR BOR での値	他の リセット での値(1)	
<b>Bank 1</b>												
80h(4)	INDF	間接アドレッシングのアクセスレジスタ(物理的には存在しません)								0000 0000	0000 0000	
81h	OPTION	RBP $\bar{U}$	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0	1111 1111	1111 1111	
82h(4)	PCL	プログラム・カウンタ(PC)の最下位バイト								0000 0000	0000 0000	
83h(4)	STATUS	IRP(7)	RP1(7)	RP0	$\bar{T}O$	$\bar{P}D$	Z	DC	C	0001 1xxx	000q quuu	
84h(4)	FSR	間接データメモリアドレスポインタ								xxxx xxxx	uuuu uuuu	
85h	TRISA	-	-	PORTAのデータ指示レジスタ						--11 1111	-- 11 1111	
86h	TRISB	PORTBのデータ指示レジスタ								1111 1111	1111 1111	
87h	TRISC	PORTCのデータ指示レジスタ								1111 1111	1111 1111	
88h(5)	TRISD	PORTDのデータ指示レジスタ								1111 1111	1111 1111	
89h(5)	TRISE	IBF	OBF	IBOV	PSPMODE	-	TRISE2	TRISE1	TRISE0	0000 -111	0000 -111	
8Ah(1,2)	PCLATH	-	-	-	PCの上位5ビット用のライトバッファ					---0 0000	---0 0000	
8Bh(1)	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u	
8Ch	PIE1	PSP1E(3)		ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
8Dh	PIE2	-	-	-	-	-	-	-	CCP2IE	---- ---0	---- ---0	
8Eh	PCON	-	-	-	-	-	-	POR	BOR(6)	---- --qq	---- --uu	
8Fh	-	未使用								-	-	
90h	-	未使用								-	-	
91h	-	未使用								-	-	
92h	PR2	Timer2周期レジスタ								1111 1111	1111 1111	
93h	SSPADD	同期シリアルポート(1 <sup>2</sup> Cモード)アドレスレジスタ								0000 0000	0000 0000	
94h	SSPSTAT	-	-	D/ $\bar{A}$	P	S	R/ $\bar{W}$	UA	BF	--00 0000	--00 0000	
95h	-	未使用								-	-	
96h	-	未使用								-	-	
97h	-	未使用								-	-	
98h	TXSTA	CSRC	TX9	TXEN	SYNC	-	BRGH	TRMT	TX9D	0000 -010	0000 -010	
99h	SPBRG	ポーレートジェネレータレジスタ								0000 0000	0000 0000	
9Ah	-	未使用								-	-	
9Bh	-	未使用								-	-	
9Ch	-	未使用								-	-	
9Dh	-	未使用								-	-	
9Eh	-	未使用								-	-	
9Fh	ADCON1	-	-	-	-	-	PCFG2	PCFG1	PCFG0	---- -000	---- -000	

凡例: x = 未知、u = 不変、q = 値は条件による、- = 未使用 ('0' としてリード)。  
網掛け部分は未使用 ('0' としてリード)。

注1: プログラムカウンタの上位バイトは直接アクセスすることはできません。PCLATHはPC<12:8>のホールディングレジスタで、その内容はプログラム・カウンタの上位バイトに転送されます。

- 2: 他の(non power-up)リセットは、MCLRとウォッチドッグ・タイマ・リセットを含みます。
- 3: PSP1EとPSP1FビットはPIC16C73/73Aではリザーブされ、常にこれらのビットはクリアされた状態です。
- 4: これらのレジスタは、全てのバンクからアドレス可能です。
- 5: PORTDとPORTEはPIC16C73/73Aでは物理的にインプリメントされていません。'0' としてリード。
- 6: ブラウンアウト・リセットは、PIC16C73またはPIC16C74ではインプリメントされていません。'0' としてリード。
- 7: IRPとRP1ビットはPIC16C7Xではリザーブされており、常にこれらのビットはクリアされた状態です。

### 4.2.2.1 STATUSレジスタ

適用デバイス							
710	71	711	72	73	73A	74	74A

STATUSレジスタは図4-9で示すように、ALUの演算ステータス、RESETステータス、データメモリのバンク選択ビットを含んでいます。

STATUSレジスタは他のレジスタと同様に、命令のデスティネーションになることができます。STATUSレジスタがZ、DC、Cビットに影響する命令のデスティネーションである場合は、この3個のビットへのライトはできません。これらのビットはデバイスのロジックによってセットまたはクリアされます。さらに $\overline{TO}$ と $\overline{PD}$ ビットは書込みができません。したがってデスティネーションとしてSTATUSレジスタを伴う命令の結果が意図したものと異なることがあります。

例えば、CLRF STATUSは上位3ビットをクリアし、Zビットをセットします。これによってSTATUSレジスタは000u u1uu(u=一定)のままです。

したがってBCF、BSF、SWAPF、MOVWFの命令だけはSTATUSレジスタからZ、C、DCビットに影響しないので、これらの命令だけをSTATUSレジスタを変えるのに使うことをお勧めします。ステータス・ビットに影響を与えない他の命令については、“命令セット一覧”をご覧ください。

注意1: IRPとRP1のビット(STATUS<7:6>)はPIC16C7Xでは使われず、クリアのままです。汎用R/Wビットとしてこれらのビットを使うことは、将来の製品との上位互換性に影響を及ぼすため、お勧めできません。

注意2: CとDCのビットは、減算命令実行時にはborrowとdigit borrowビットとして動作します。一例としてSUBLWとSUBWFの命令をご覧ください。

図 4-9: STATUSレジスタ(アドレス03h、83h)

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x	
IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	
bit7								bit0
bit 7:	<b>IRP</b> : レジスタバンク選択ビット(インダイレクトアドレス指定用) 1 = バンク2、3(100h-1FFh) 0 = バンク0、1(000h-0FFh) IRPビットはPIC16C7Xではリザーブされており、常にこれらのビットはクリアされた状態です。							
bit 6-5:	<b>RP1</b> : <b>RP0</b> : レジスタバンク選択ビット(ダイレクトアドレス指定用) 11 = バンク3(180h-1FFh) 10 = バンク2(100h-17Fh) 01 = バンク1(080h-0FFh) 00 = バンク0(000h-07Fh) 各バンクは128バイトです。RP1ビットはPIC16C7Xではリザーブされており、常にこれらのビットはクリアされた状態です。							
bit 4:	<b><math>\overline{TO}</math></b> : タイムアウトビット 1 = パワーアップ後、CLRWDT命令またはSLEEP命令を実行後 0 = WDTタイムアウトの発生							
bit 3:	<b><math>\overline{PD}</math></b> : パワーダウンビット 1 = パワーアップ後、またはCLRWDT命令を実行後 0 = SLEEP命令の実行後							
bit 2:	<b>Z</b> : ゼロビット 1 = 数値演算または論理演算の結果が0 0 = 数値演算または論理演算の結果が0以外							
bit 1:	<b>DC</b> : デジットキャリイ/borrowビット(ADDWF、ADDLW、SUBLW、SUBWF命令)(borrowでは極性は逆) 1 = 結果の下位4ビットからのキャリイが出力された 0 = 結果の下位4ビットからのキャリイが出力されない							
bit 0:	<b>C</b> : キャリイ/borrowビット(ADDWF、ADDLW、SUBLW、SUBWF命令) 1 = 結果の最上位ビットからのキャリイが出力された 0 = 結果の最上位ビットからのキャリイが出力されない							
注意: borrowでは極性は逆です。減算は2番目のオペランドの2の補数を加えることにより実行されます。ローテート命令(RRF、RLF)では、このビットはソースレジスタの上位ビットまたは下位ビットにロードされません。								

R = 読み込み可能なビット  
W = 書き込み可能なビット  
U = 未使用のビット、  
'0'としてリード  
- n = PORリセットでの値

### 4.2.2.2 OPTIONレジスタ

適用デバイス							
710	71	711	72	73	73A	74	74A

OPTIONレジスタはリードとライトができるレジスタで、TMRO/WDTプリスケアラ、外部INT割込み、TMRO、PORTBのプルアップを構成するための色々な制御ビットを含んでいます。

注意: TMROレジスタに対して1:1のプリスケアラの割当てを達成するためには、セッティングビットPSA(OPTION<3>)によってウォッチドッグタイマにプリスケアラを割当てます。

図 4-10: OPTION レジスタ (アドレス 81h)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
RBPU	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0
bit7							bit0

bit7: **RBPU**: PORTBプルアップイネーブルビット  
 1 = PORTBプルアップをディセーブル  
 0 = それぞれのポートラッチ値でPORTBプルアップをイネーブル

bit6: **INTEDG**: 割込みエッジ選択ビット  
 1 = RB0/INTピンの立ち上がりエッジで割込み  
 0 = RB0/INTピンの立ち下がりエッジで割込み

bit5: **TOCS**: TMROクロックソース選択ビット  
 1 = RA4/TOCKIピン上の変化  
 0 = 内部命令サイクル・クロック (CLKOUT)

bit4: **TOSE**: TMROソースエッジ選択ビット  
 1 = RA4/TOCKIピンでのハイからローへの変化でインクリメント  
 0 = RA4/TOCKIピンでのローからハイへの変化でインクリメント

bit3: **PSA**: プリスケアラアサインメントビット  
 1 = プリスケアラをWDTにアサイン  
 0 = プリスケアラをタイマ0モジュールにアサイン

bit2-0: **PS2:PS0**: プリスケアラレート選択ビット

Bit Value	TMRO Rate	WDT Rate
000	1:2	1:1
001	1:4	1:2
010	1:8	1:4
011	1:16	1:8
100	1:32	1:16
101	1:64	1:32
110	1:128	1:64
111	1:256	1:128

R = 読み込み可能なビット  
 W = 書き込み可能なビット  
 U = 未使用のビット、  
 '0' としてリード  
 - n = PORリセットでの値

# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。

## PIC16C7X

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

### 4.2.2.3 INTCON レジスタ

適用デバイス							
710	71	711	72	73	73A	74	74A

INTCONレジスタはリードとライトができるレジスタで、TMROレジスタのオーバーフロー、RBポートのレベル変化、外部RB0/INTピンの割込みのための色々なイネーブルビットとフラグビットが含まれています。

注意： その対応するイネーブルビットやグローバルイネーブルビット、GIE(INTCON<7>)の状態にかかわらず割込み条件が生じる時には、割込みフラグビットがセットされます。

図4-11： PIC16C710/71/711用 INTCON レジスタ ( アドレス0Bh、8Bh )

R/W-0	R/W-x						
GIE	ADIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF
bit7						bit0	

R = 読み込み可能なビット  
W = 書き込み可能なビット  
U = 未使用のビット、  
'0'としてリード  
- n = PORリセットでの値

- bit7: **GIE**: グローバル割込みイネーブルビット(1)  
1=マスクされていない割込みが発生時、割込みベクタに分岐します。  
0=割込みベクタへの分岐を禁止します。
- bit6: **ADIE**: A/Dコンバータ割込みイネーブルビット  
1=A/D割込みをイネーブルにします。  
0=A/D割込みをディセーブルにします。
- bit5: **TOIE**: TMROオーバーフロー割込みイネーブルビット  
1=TMRO割込みをイネーブルにします。  
0=TMRO割込みをディセーブルにします。
- bit4: **INTE**: RB0/INT外部割込みイネーブルビット  
1=RB0/INT外部割込みをイネーブルにします。  
0=RB0/INT外部割込みをディセーブルにします。
- bit3: **RBIE**: RBポート変更割込みイネーブルビット  
1=RBポート変更割込みをイネーブルにします。  
0=RBポート変更割込みをディセーブルにします。
- bit2: **TOIF**: TMROオーバーフロー割込みフラグビット  
1=TMROレジスタがオーバーフローした場合(ソフトウェアでクリア)。  
0=TMROレジスタがオーバーフローしなかった場合。
- bit1: **INTF**: RB0/INT外部割込みフラグビット  
1=RB0/INT外部割込みが発生した場合(ソフトウェアでクリア)。  
0=RB0/INT外部割込みが発生しなかった場合。
- bit0: **RBIF**: RBポート変更割込みフラグビット  
1=少なくともRB7:RB4ピンの1個が変化した場合(ソフトウェアでクリア)。  
0=RB7:RB4ピンのどちらも変化していない場合。

注 1: PIC16C711は、GIEビットがクリアされている間に割込みが発生した場合、GIEビットは、ユーザの割込みサービスルーチンでのRETFIE命令により、意図せず再イネーブルされることがあります。詳細については、14.5章をご覧ください。

図4-12 : PIC16C72/73/73A/74/74A用 INTCONレジスタ (アドレス0Bh、8Bh)

R/W-0	R/W-x						
GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF
bit7				bit0			

bit7 : **GIE** : グローバル割込みイネーブルビット(1)  
1=マスクされていない割込みが発生時、割込みベクタに分岐します。  
0=割込みベクタへの分岐を禁止します。

bit6 : **PEIE** : 周辺割込みイネーブルビット  
1=マスクされていないすべての周辺割込みをイネーブルにします。  
0=すべての周辺割込みをディセーブルにします。

bit5 : **TOIE** : TMR0オーバーフロー割込みイネーブルビット  
1=TMR0割込みをイネーブルにします。  
0=TMR0割込みをディセーブルにします。

bit4 : **INTE** : RB0/INT外部割込みイネーブルビット  
1=RB0/INT外部割込みをイネーブルにします。  
0=RB0/INT外部割込みをディセーブルにします。

bit3 : **RBIE** : RBポート変更割込みイネーブルビット  
1=RBポート変更割込みをイネーブルにします。  
0=RBポート変更割込みをディセーブルにします。

bit2 : **TOIF** : TMR0オーバーフロー割込みフラグビット  
1=TMR0レジスタがオーバーフローした場合(ソフトウェアでクリア)  
0=TMR0レジスタがオーバーフローしなかった場合

bit1 : **INTF** : RB0/INT外部割込みフラグビット  
1=RB0/INT外部割込みが発生した場合(ソフトウェアでクリア)  
0=RB0/INT外部割込みが発生しなかった場合

bit0 : **RBIF** : RBポート変更割込みフラグビット  
1=少なくともRB7:RB4ピンの1個が変化した場合(ソフトウェアでクリア)  
0=RB7:RB4ピンのどちらも変化していない場合

注1 : PIC16C73とPIC16C74は、GIEビットがクリアされている間に割込みが発生した場合、GIEビットは、ユーザの割込みサービスルーチンでのRETFIE命令により、意図せず再イネーブルされることがあります。詳細については、14.5章をご覧ください。

R = 読み込み可能なビット  
W = 書き込み可能なビット  
U = 未使用のビット、  
‘0’としてリード  
- n = PORリセットでの値

# 暫定版

英語最新版データシートと併用にご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。  
<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

### 4.2.2.4 PIE1 レジスタ

適用デバイス							
710	71	711	72	73	73A	74	74A

注意: 周辺割込みを使用する場合、PEIEビット (INTCON<6>) をイネーブルにする必要があります。

このレジスタには、周辺割込みに対応した個々のイネーブルビットが含まれています。

図 4-13: PIC16C72用 PIE1 レジスタ ( アドレス 8Ch)

U-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
-	ADIE	-	-	SSPIE	CCP1IE	TMR2IE	TMR1IE
bit7				bit0			

R = 読み込み可能なビット  
W = 書き込み可能なビット  
U = 未使用のビット、  
‘0’ としてリード  
- n = PORリセットでの値

bit7: **未使用**: 0としてリード

bit6: **ADIE**: A/D変換器割込みイネーブルビット  
1=A/D変換器割込みをイネーブルにします。  
0=A/D変換器割込みをディセーブルにします。

bit5-4: **未使用**: 0としてリード

bit3: **SSPIE**: 同期シリアルポート割込みイネーブルビット  
1=SSP割込みをイネーブルにします。  
0=SSP割込みをディセーブルにします。

bit2: **CCP1IE**: CCP1割込みイネーブルビット  
1=CCP1割込みをイネーブルにします。  
0=CCP1割込みをディセーブルにします。

bit1: **TMR2IE**: PR2とTMR2がマッチした時の割込みイネーブルビット  
1=PR2とTMR2がマッチした時の割込みをイネーブルにします。  
0=PR2とTMR2がマッチした時の割込みをディセーブルにします。

bit0: **TMR1IE**: TMR1オーバーフロー割込みイネーブルビット  
1=TMR1オーバーフロー割込みをイネーブルにします。  
0=TMR1オーバーフロー割込みをディセーブルにします。

図 4-14: PIC16C73/73A/74/74A用 PIE1レジスタ(アドレス8Ch)

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	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	
bit7								bit0

R = 読み込み可能なビット  
W = 書き込み可能なビット  
U = 未使用のビット、  
‘0’としてリード  
- n = PORリセットでの値

bit7: **PSPIE<sup>(1)</sup>**: パラレルスレーブポートの リード/ライト割込みイネーブルビット  
1=PSPリード/ライト割込みをイネーブルにします。  
0=PSPリード/ライト割込みをディセーブルにします。

bit6: **ADIE**: A/D変換器割込みイネーブルビット  
1=A/D変換器割込みをイネーブルにします。  
0=A/D変換器割込みをディセーブルにします。

bit5: **RCIE**: USART受信割込みイネーブルビット  
1=USART受信割込みをイネーブルにします。  
0=USART受信割込みをディセーブルにします。

bit4: **TXIE**: USART送信割込みイネーブルビット  
1=USART送信割込みをイネーブルにします。  
0=USART送信割込みをディセーブルにします。

bit3: **SSPIE**: 同期シリアルポート割込みイネーブルビット  
1=SSP割込みをイネーブルにします。  
0=SSP割込みをディセーブルにします。

bit2: **CCP1IE**: CCP1割込みイネーブルビット  
1=CCP1割込みをイネーブルにします。  
0=CCP1割込みをディセーブルにします。

bit1: **TMR2IE**: PR2とTMR2がマッチした時の割込みイネーブルビット  
1=PR2とTMR2がマッチした時の割込みをイネーブルにします。  
0==PR2とTMR2がマッチした時の割込みをディセーブルにします。

bit0: **TMR1IE**: TMR1オーバーフロー割込みイネーブルビット  
1=TMR1オーバーフロー割込みをイネーブルにします。  
0=TMR1オーバーフロー割込みをディセーブルにします。

注 1: PIC16C73とPIC16C73Aデバイスには、パラレルスレーブポートがありません。このビットのロケーションはこれら2つのデバイスでリザーブされており、常にこのビットはクリアされた状態です。

# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。  
<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

### 4.2.2.5 PIR1 レジスタ

適用デバイス							
710	71	711	72	73	73A	74	74A

このレジスタには、周辺割込みに対応する個々のフラグビットが含まれます。

注意： 割込みフラグビットは、割込みの条件が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、GIE (INTCON<7>)の状態に関わらず発生する場合にセットされます。ユーザは、割込みをイネーブルする前に適切な割込みフラグビットをソフトウェアにて確実にクリアする必要があります。

図4-15： PIC16C72用 PIR1 レジスタ(アドレス 0Ch)

U-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
-	ADIF	-	-	SSPIF	CCP1IF	TMR2IF	TMR1IF
bit7				bit0			

R = 読み込み可能なビット  
W = 書き込み可能なビット  
U = 未使用のビット、  
‘0’としてリード  
- n = PORリセットでの値

- bit7： 未使用：‘0’としてリード
- bit6： **ADIF**：A/D変換器割込みフラグビット  
1=A/D変換が完了(ソフトウェアでクリア)。  
0=A/D変換が未完了。
- bit5-4： **未使用**：‘0’としてリード
- bit3： **SSPIF**：同期シリアルポート割込みフラグビット  
1=送信/受信が完了(ソフトウェアでクリア)。  
0=送信/受信を待機。
- bit2： **CCP1IF**：CCP1割込みフラグビット  
キャプチャモード  
1=TMR1レジスタのキャプチャが起こった場合(ソフトウェアでクリア)。  
0=どのTMR1レジスタのキャプチャも起こらなかった場合。  
コンペアモード  
1=TMR1レジスタのコンペアマッチが起こった場合(ソフトウェアでクリア)。  
0=どのTMR1レジスタのコンペアマッチも起こらなかった場合。  
PWMモード  
このモードでは未使用。
- bit1： **TMR2IF**：PR2とTMR2がマッチした時の割込みフラグビット  
1=PR2とTMR2のマッチが起こった場合(ソフトウェアでクリア)。  
0=PR2とTMR2のマッチが起こらなかった場合。
- bit0： **TMR1IF**：TMR1オーバーフロー割込みフラグビット  
1=TMR1レジスタがオーバーフローした場合(ソフトウェアでクリア)。  
0=TMR1レジスタがオーバーフローしなかった場合。

図4-16: PIC16C73/73A/74/74A用 PIR1レジスタ(アドレス0Ch)

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	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
bit7				bit0			

R = 読み込み可能なビット  
W = 書き込み可能なビット  
U = 未使用のビット、  
‘0’としてリード  
- n = PORリセットでの値

bit7: **PSPIF(1)**: パラレルスレーブポートのリード/ライト割込みフラグビット  
1=リード/ライト操作が行われた場合(ソフトウェアでクリア)。  
0=リード/ライト操作が行われなかった場合。

bit6: **ADIF**: A/D変換器割込みフラグビット  
1=A/D変換が完了(ソフトウェアでクリア)。  
0=A/D変換が未完了。

bit5: **RCIF**: USART受信割込みフラグビット  
1=USART受信バッファがフル(ソフトウェアでクリア)。  
0=USART受信バッファが空。

bit4: **TXIF**: USART送信割込みフラグビット  
1=USART送信バッファが空(ソフトウェアでクリア)。  
0=USART送信バッファがフル。

bit3: **SSPIF**: 同期シリアルポート割込みフラグビット  
1=送信/受信が完了(ソフトウェアでクリア)。  
0=送信/受信を待機。

bit2: **CCP1IF**: CCP1割込みフラグビット  
キャプチャモード  
1=TMR1レジスタのキャプチャが起こった場合(ソフトウェアでクリア)。  
0=どのTMR1レジスタのキャプチャも起こらなかった場合。  
コンペアモード  
1=TMR1レジスタのコンペアマッチが起こった場合(ソフトウェアでクリア)。  
0=どのTMR1レジスタのコンペアマッチも起こらなかった場合。  
PWMモード  
このモードでは未使用。

bit1: **TMR2IF**: PR2とTMR2がマッチした時の割込みフラグビット  
1=PR2とTMR2のマッチが起こった場合(ソフトウェアでクリア)。  
0==PR2とTMR2のマッチが起こらなかった場合。

bit0: **TMR1IF**: TMR1オーバーフロー割込みフラグビット  
1=TMR1レジスタがオーバーフローした場合(ソフトウェアでクリア)。  
0=TMR1レジスタがオーバーフローしなかった場合。

注 1: PIC16C73とPIC16C73Aデバイスにはパラレルスレーブポートはありません。このビットのロケーションは、これら2つのデバイスでリザーブされており、常にこのビットはクリアされた状態です。

# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。

## PIC16C7X

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

### 4.2.2.6 PIE2 レジスタ

適用デバイス							
710	71	711	72	73	73A	74	74A

このレジスタには、CCP2の周辺割込みに対応した個々のイネーブル・ビットが含まれています。

図4-17: PIE2 レジスタ (アドレス 8Dh)



### 4.2.2.7 PIR2 レジスタ

適用デバイス							
710	71	711	72	73	73A	74	74A

このレジスタには、CCP2割込みフラグビットが含まれています。

注意：割込みフラグビットは、割込みの条件が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、GIE (INTCON<7>)の状態に関わらず発生する場合にセットされます。ユーザは、割込みをイネーブルする前に適切な割込みフラグビットをソフトウェアにて確実にクリアする必要があります。

図4-18: PIR2 レジスタ (アドレス 0Dh)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
-	-	-	-	-	-	-	CCP2IF
bit7							bit0

R = 読み込み可能なビット  
 W = 書き込み可能なビット  
 U = 未使用のビット、  
 '0'としてリード  
 - n = PORリセットでの値

bit7-1: **未使用**：'0'としてリード

bit0: **CCP2IF**：CCP2割込みフラグビット

キャプチャモード  
 1=TMR1レジスタキャプチャが起こった場合(ソフトウェアでクリア)。  
 0=どのTMR1レジスタキャプチャも起こらなかった場合。

コンペアモード  
 1=TMR1レジスタのコンペアマッチが起こった場合(ソフトウェアでクリア)。  
 0=どのTMR1レジスタのコンペアマッチも起こらなかった場合。

PWMモード  
 未使用。

# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。

## PIC16C7X

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

### 4.2.2.8 PCON レジスタ

適用デバイス							
710	71	711	72	73	73A	74	74A

パワーコントロール(PCON)レジスタには、パワーオンリセット(POR)とそれ以外のリセットを判断するためのフラグビットが含まれています。またブラウンアウトリセット(BOR)が発生したかどうかを判断するステータスビットも含まれています。

注意:  $\overline{BOR}$ はパワーオン・リセットでは不定です。したがって、イニシャライズルーチンなどでBORビットをセットした後このビットをモニターリングする必要があります。ブラウンアウト回路がディスエーブルの場合(コンフィギュレーションワードのBODENビットをクリアすることにより)、BORステータスビットは無効になります。

図 4-19: PCON レジスタ (アドレス 8Eh)

U-0	U-0	U-0	U-0	U-0	U-0	R/W-q	R/W-q
-	-	-	-	-	-	$\overline{POR}$	$\overline{BOR}^{(1)}$
bit7						bit0	

R = 読み込み可能なビット  
W = 書き込み可能なビット  
U = 未使用のビット、  
‘0’としてリード  
- n = PORリセットでの値

bit7-2: **未使用:** ‘0’としてリード

bit1: **POR:** パワーオンリセットのステータスビット  
1=パワーオンリセットが起こらなかった場合。  
0 = パワーオンリセットが起こった場合( パワーオンリセットが起こった後は、ソフトウェアでセット)。

bit0: **BOR<sup>(1)</sup>:** ブラウンアウトリセットのステータスビット  
1=ブラウンアウトリセットが起こらなかった場合。  
0 = ブラウンアウトリセットが起こった場合( ブラウンアウトリセットが起こった後、ソフトウェアでセット)。

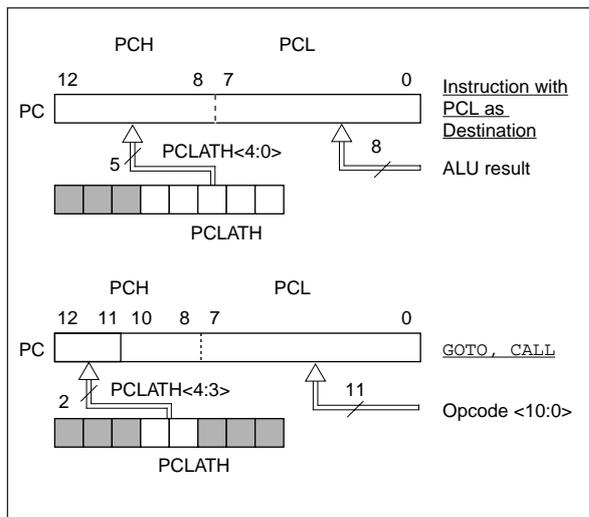
注1: ブラウンアウトリセットはPIC16C73/74ではインプリメントされていません。

### 4.3 PCL と PCLATH

適用デバイス							
710	71	711	72	73	73A	74	74A

プログラムカウンタ(PC)は13ビット幅です。下位バイトはPCLレジスタから生成され、このレジスタはリードとライトができるレジスタです。上位バイト(PC<12:8>)は直接リードできません。ライトは、PCLATHから行います。どんなリセットでもPCはクリアされます。図4-20に示すようにP C のローディングには2つの状態があります。上の図例はPCがPCL(PCLATH<4:0> PCH)へのライトの上にロードされる方法を示しています。下の図例はPCがCALLまたはGOTO命令(PCLATH<4:3> PCH)の間にロードされる方法を示しています。

図4-20: 異なる状態でのPCのローディング



#### 4.3.1 計算結果によるGOTO

計算結果によるGOTOは、プログラムカウンタ(ADDWF PCL)にオフセットを加えると完了します。計算結果によるGOTOの方法を使ったテーブルリードの時は、テーブルロケーションがPCLメモリの境界(各256バイト・ブロック)を越えるかどうかを検査する必要があります。アプリケーションノート“テーブルリードのインプリメンテーション”(AN556)を参照してください。

#### 4.3.2 スタック

PIC16CXXファミリには、8レベル×13ビット幅のハードウェアスタックがあります。スタック空間はプログラムメモリまたはデータメモリとは別に用意されており、スタックポインタは読み込み、書き込みもができません。CALL命令が実行されるか、割込みが発生した場合、PCがスタックにPUSHされます。スタックはRETURN、RETLW、RETFIEの命令実行時にPOPされます。PCLATHはPUSHまたはPOP動作の影響を受けません。

スタックはサーキュラバッファとして動作します。これはスタックが8回PUSHされた後、9回目のプッシュが1回目のプッシュによって保存された値をオーバーライトすることを意味します。10回目のプッシュは2回目のプッシュをオーバーライトします。

注意 1: スタックのオーバーフローかアンダーフローを示すステータスビットはありません。

注意 2: PUSHまたはPOPと呼ばれる命令やニーモニックはありません。

### 4.4 プログラムメモリのページング

適用デバイス							
710	71	711	72	73	73A	74	74A

PIC16C73/73AとPIC16C74/74Aには、4Kのプログラムメモリがありますが、CALLとGOTO命令には11ビットのアドレス幅しかありません。この11ビットのアドレス幅は2Kのプログラムメモリのページサイズ以内でのみ分岐することができます。CALLとGOTO命令によって、4Kのプログラムメモリをアドレッシングするためには、プログラムメモリのページを指定するために、もう1つのビットが必要となります。このページビットは、PCLATH<3>ビットから生じています(図4-20参照)。CALLとGOTO命令を使用するときは、確実にこのページビット(PCLATH<3>)をプログラムしなければなりません。CALL命令(または割込み)実行時は全13ビットのPCがスタックにPUSHされます。リターンする場合は、スタックから全13ビットのPCデータがPOPされますので、PCLATH<3>の操作は不要です。(リターン時、PCLATH<3>のデータは変化しません。)

注意 1: PIC16C710/71/711/72は両方のページビット(PCLATH<4:3>)が無効です。PIC16C7Xにおいて汎用リード/ライトビットとしてPCLATH<4:3>を使うことは、将来の製品との上位互換性に影響を及ぼすため、お勧めできません。

PIC16C73/73A/74/74Aはページビット(PCLATH<4>)が無効です。このビットはプログラムメモリの2ページと3ページ(1000h-1FFFh)をアクセスするために使われます。汎用リード/ライトビットとしてPCLATH<4>を使うことは、将来の製品との上位互換性に影響を及ぼすため、お勧めできません。

# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。  
<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

例4-1は、プログラムメモリのページ1へのサブルーチンコールの例を示しています。もし、割込みが使用される場合、割込みサービスルーチンでPCLATHがセーブとストアされていることを前提としております。

### 例 4-1: ページ0からページ1内の サルーチン・コール

```

ORG    0x500
BSF    PCLATH,3    ;Select page1(800h-FFFh)
CALL   SUB1_P1     ;Call subroutine in
           :         ;page1 (800h-FFFh)
           :
           :
ORG    0x900
SUB1_P1 :         ;called subroutine
           :         ;page1 (800h-FFFh)
           :
RETURN          ;return to Call subroutine
           ;in page 0 (000h-7FFh)
    
```

### 4.5 間接アドレッシング指定、INDF、FSRレジスタ

適用デバイス							
710	71	711	72	73	73A	74	74A

INDFレジスタは物理的なレジスタではありません。INDFレジスタのアドレス指定によって間接アドレス指定ができます。

INDFレジスタを使って間接アドレス指定ができます。INDFレジスタを使ったどの命令も、実際にはファイル選択レジスタ(FSR)により示されたレジスタをアクセスします。INDFレジスタ自身を間接的に読み込むと(FSR='0'), 00hが読み出されます。INDFレジスタを間接的に書き込むことは、何も動作しない状態になります(ステータスビットが影響を受けることがあります)。有効な9ビットアドレスは、図4-21に示すように8ビットFSRレジスタとIRPビット(STATUS<7>)を結合することで得られます。ただしIRPはPIC16C7Xでは使用されません。

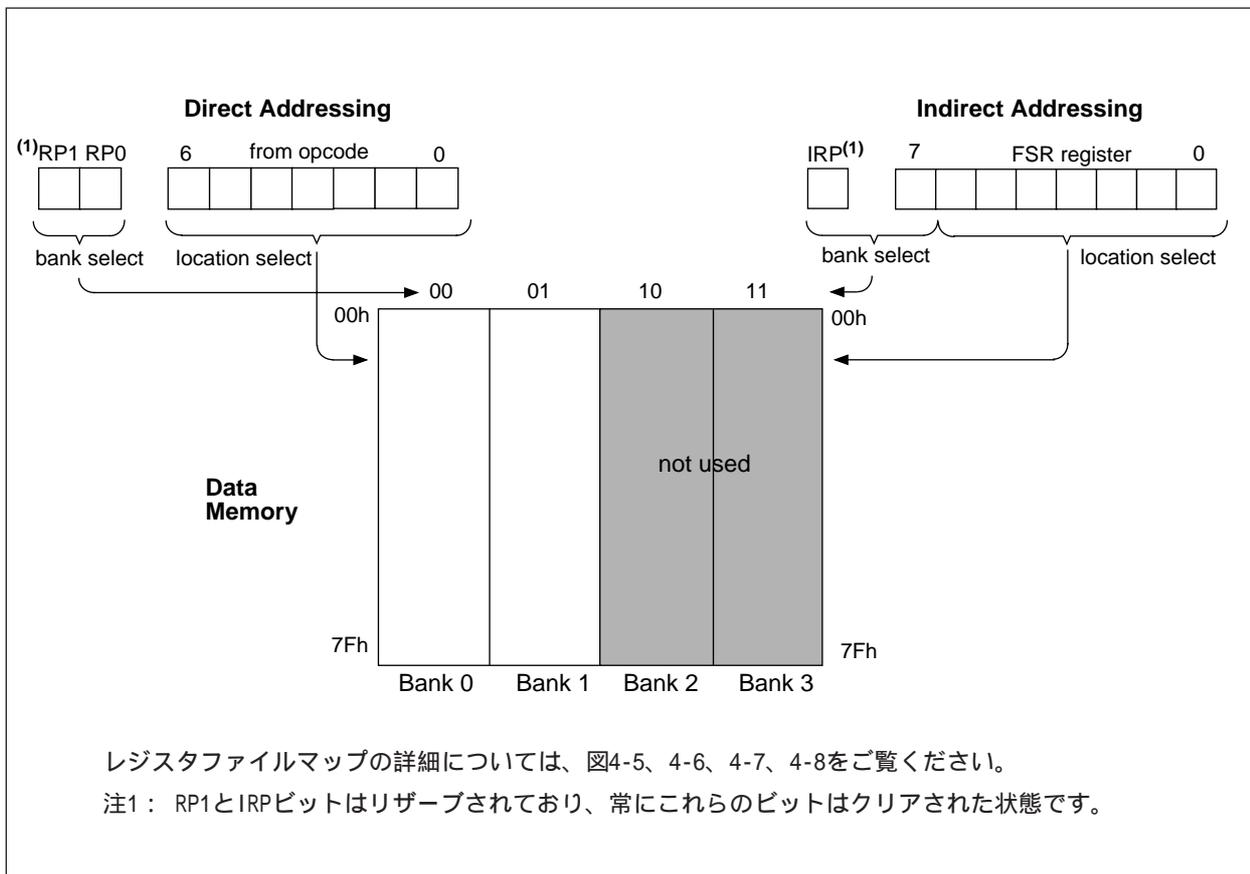
間接アドレス指定によってRAMロケーション20h-2Fhをクリアする簡単なプログラムを例4-2に示します。

### 例 4-2: 間接アドレッシング

```

movlw  0x20    ;initialize pointer
movwf  FSR     ;to RAM
NEXT   clr     INDF ;clear INDF register
       incf   FSR,F ;inc pointer
       btfs  FSR,4 ;all done?
       goto  NEXT ;no clear next
CONTINUE
       :         ;yes continue
    
```

図4-21: 直接 / 間接アドレッシング



### 5.0 I/Oポート

適用デバイス							
710	71	711	72	73	73A	74	74A

これらのI/Oポートに対するいくつかのピンは、デバイス上の周辺機能を多重化しています。一般に周辺回路がイネーブルされている場合、そのピンは汎用I/Oピンとして使用されません。

#### 5.1 PORTAとTRISAレジスタ

適用デバイス							
710	71	711	72	73	73A	74	74A

PIC16C710/71/711のPORTAは、5ビット幅のラッチです。PIC16C72/73/73A/74/74AのPORTAは、6ビット幅のラッチです。

RA4/TOCKIピンは、シュミットトリガ入力とオープンドレイン出力です。他のすべてのRAポートピンには、TTL入力レベルとCMOC出力ドライバがあります。すべてのピンにはデータ方向ビット(TRISレジスタ)があり、出力または入力としてこれらのピンを構成することができます。

TRISAレジスタビットのセットは、対応する出力ドライバを高インピーダンスモードにします。TRISAレジスタのビットをクリアにすると、選択したピン上に出カラッチの内容を出します。

PORTAレジスタを読み込むと、そのピンのステータスを読み出し、それに書き込むとポートラッチに書き入れます。すべてのライト動作は、リード-モディファイ-

ライト動作です。したがってポートへのライトは、ポートピンが読み込まれたことを意味し、この値が変更されたあと、ポートデータラッチに書き込まれません。

RA4ピンはTOCKIピンになるためにタイマ0モジュールクロック入力とマルチプレクスされています。

他のPORTAピンはアナログ入力とアナログV<sub>REF</sub>入力でもマルチプレクスされます。各ピンの動作はADCON1レジスタ(A/D制御レジスタ1)の制御ビットをクリアまたはセットすることで選択します。

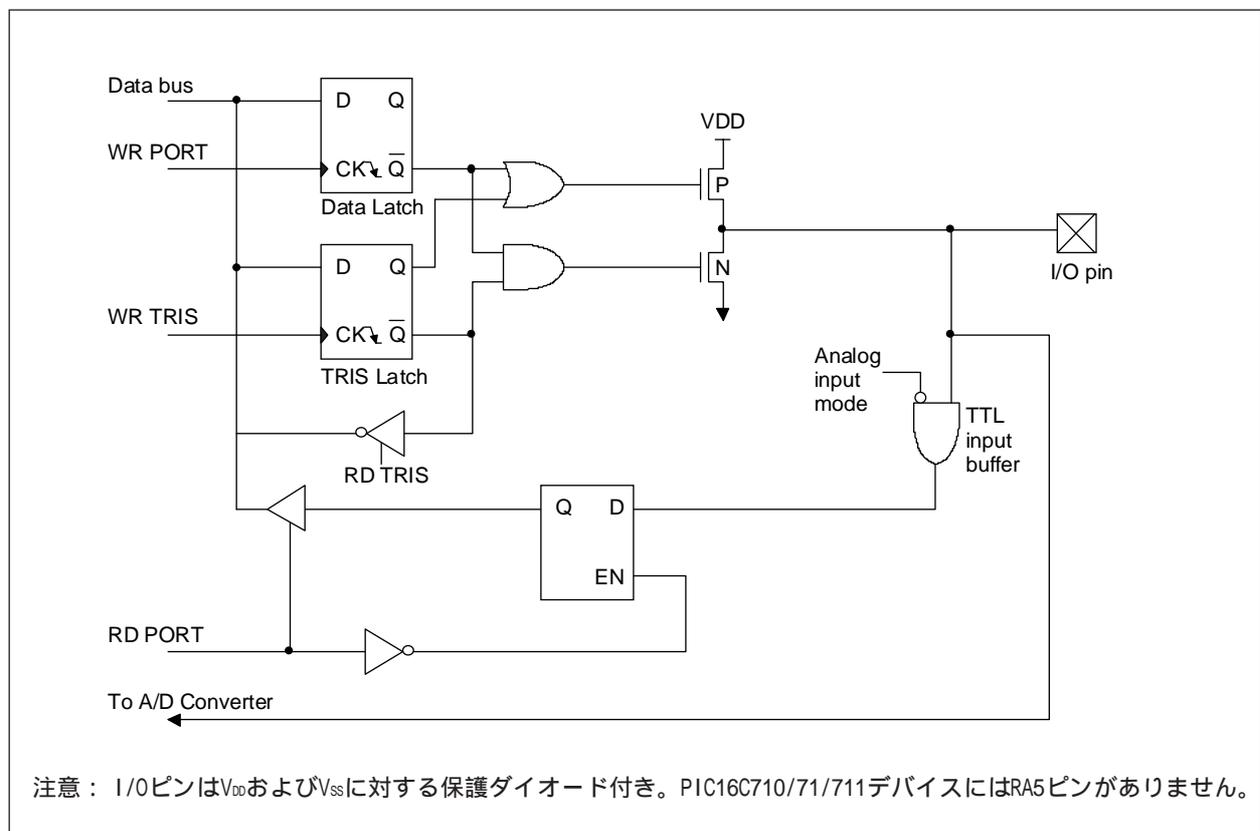
注意: パワーオンリセットでは、これらのピンはアナログ入力として設定され、'0'としてリードされます。

アナログ入力時においてもTRISAレジスタは、RAピンの方向を制御します。TRISAレジスタのビットをアナログ入力として使う場合は、セットし続けなければなりません。

#### 例5-1: PORTAの初期化

```
CLRF   PORTA      ; Initialize PORTA by
                ; setting output
                ; data latches
BSF    STATUS, RP0 ; Select Bank 1
MOVLW  0xCF       ; Value used to
                ; initialize data
                ; direction
MOVWF  TRISA      ; Set RA<3:0> as inputs
                ; RA<5:4> as outputs
                ; TRISA<7:6> are always
                ; read as '0'.
```

図5-1: RA3:RA0とRA5ピンのブロック図



注意: I/OピンはV<sub>DD</sub>およびV<sub>SS</sub>に対する保護ダイオード付き。PIC16C710/71/711デバイスにはRA5ピンがありません。

# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。  
<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

図5-2： RA4/T0CKIのブロック図

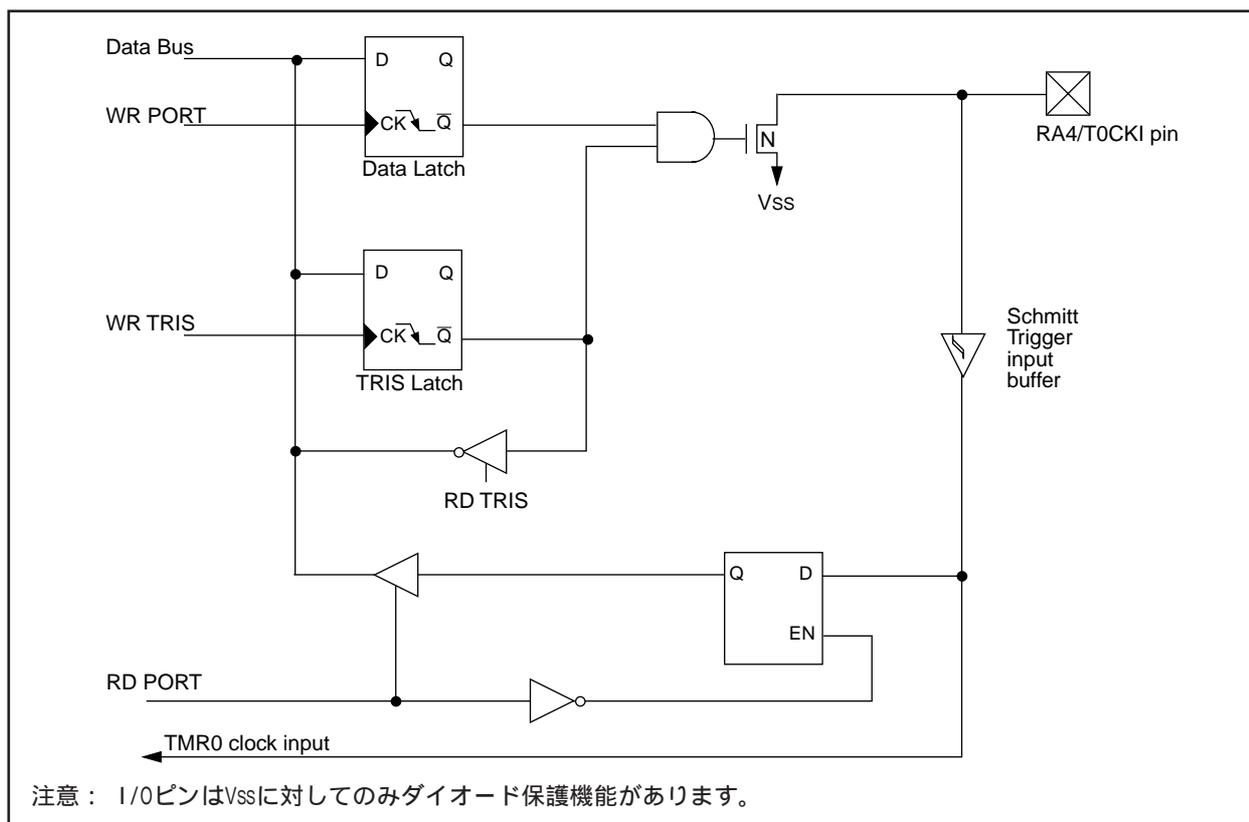


表5-1： PORTAの機能

名称	Bit #	Buffer	機能
RA0/AN0	bit 0	TTL	入出力またはアナログ入力
RA1/AN1	bit 1	TTL	入出力またはアナログ入力
RA2/AN2	bit 2	TTL	入出力またはアナログ入力
RA3/AN3/VREF	bit 3	TTL	入出力またはアナログ入力/VREF
RA4/T0CKI	bit 4	ST	入出力またはタイマ0用の外部クロック入力 出力はオープンドレインタイプ
RA5/AN4/SS(1)	bit 5	TTL	入出力または同期シリアルポート用のスレーブセレクト入力またはアナログ入力

凡例： TTL=TTL入力、ST=シュミットトリガ入力

注 1： PIC16C710/71/711にはPORTA<5>またはTRISA<5>がありません。'0'としてリード。

表5-2： PORTAに関連するレジスタの要約

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR・BOR での値	他のリセット での値
05h	PORTA	-	-	RA5(1)	RA4	RA3	RA2	RA1	RA0	--0x 0000	--0u 0000
85h	TRISA	-	-	TRISA5(1)	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111
9Fh	ADCON1	-	-	-	-	-	PCFG2(2)	PCFG1	PCFG0	---- 0000	---- 0000

凡例： x=未知、u=不変、-=未使用のロケーション、'0'としてリード。網掛けの部分はPORTAでは使われません。

注 1： PORTA<5>とTRISA<5>は PIC16C710/71/711ではインプリメントされていません。

2： PCFG2ビットは PIC16C710/71/711ではインプリメントされていません。

### 5.2 PORTB と TRISB レジスタ

適用デバイス							
710	71	711	72	73	73A	74	74A

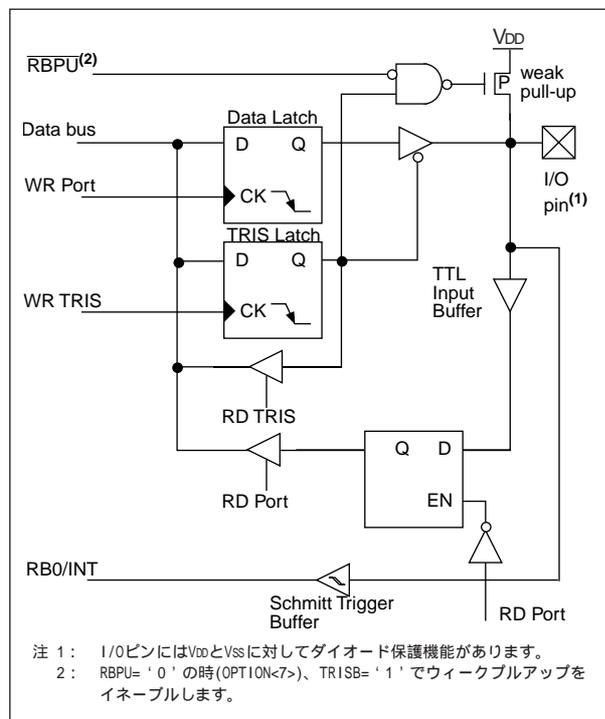
PORTBは8ビット幅の双方向性のポートです。対応するデータ方向レジスタはTRISBです。TRISBレジスタにビットをセットすることにより、対応する出力ドライバをハイインピーダンス入力モードにします。TRISBレジスタのビットをクリアすることにより、出力ラッチの内容を選択されたピン上に出力します。

#### 例5-2: PORTBの初期化

```
CLRF    PORTB        ; Initialize PORTB by
                    ; setting output
                    ; data latches
BSF     STATUS, RP0  ; Select Bank 1
MOVLW  0xCF          ; Value used to
                    ; initialize data
                    ; direction
MOVWF  TRISB        ; Set RB<3:0> as inputs
                    ; RB<5:4> as outputs
                    ; RB<7:6> as inputs
```

PORTBの各ピンには内部ウィークプルアップがあります。1本の制御ビットですべてのプルアップをオンにすることができます。これはビットRBPU(OPTION<7>)をクリアすることにより可能です。ウィークプルアップはポートピンが出力として設定された時は自動的にオフになります。このプルアップはパワーオンリセット時はディセーブルです。

図5-3: RB3:RB0 ピンのブロック図



PORTBのピンの4本、つまりRB7:RB4にはレベル変化割込み機能があります。入力として設定されたピンのみ割込みを発生させることができます(すなわち、出力として設定されたRB7:RB4ピンがレベル変化割込みから除外されます)。入力ピンは(RB7:RB4のうち)、PORTBの最後のリードでラッチされた古い値と比較されます。RB7:

RB4のミスマッチ出力をすべてORし、結果をフラグビットRBIF(INTCON<0>)に反映させ、RBポートレベル変化割込みを発生させます。

この割込みによりデバイスをSLEEPからウェークすることができます。次の方法により割込みサービスルーチンで割込みをクリアすることができます。

- PORTBのリードまたはライトによりミスマッチ状態が終了します。
- フラグビットRBIFをクリアします。

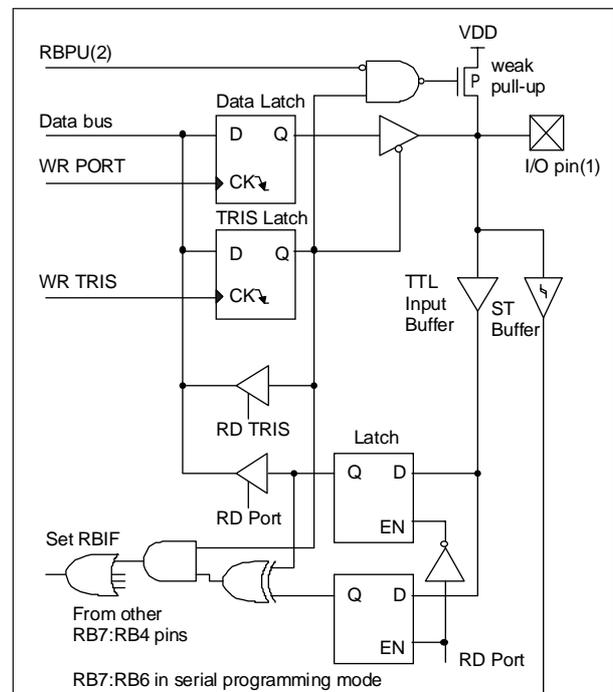
ミスマッチ状態はフラグビットRBIFをセットし続けます。PORTBを読み込むとミスマッチ状態が終了しますので、ユーザはフラグビットRBIFをクリアすることができます。

このレベル変化割込み機能は、4本のピンのソフトウェアで設定可能なプルアップにより、キーパッドとのインターフェイスを容易にし、キーを押すことによるウェークアップを可能にします。添付のコントロールハンドブック“キーストロークにおけるウェークアップのインプリメンテーション”(AN552)を参照してください。

注意: PIC16C71/73/74については、リード操作の実行中(Q2サイクルの開始)、I/Oピン上で変更を行なう場合には、割込みフラグビットRBIFが設定されないことがあります。

レベル変化割込みは、キーを押した時のウェークアップや、PORTBをレベル変化割込みのみ使用している場合の使用にお勧めします。PORTBのポーリングは、レベル変化割込みを使う間はお勧めできません。

図5-4: RB7:RB4 ピンのブロック図



# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。

## PIC16C7X

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

表5-3: PORTB の機能

名称	Bit #	Buffer Type	機能
RB0/INT	bit0	TTL/ST(1)	入出力ピンまたは外部割込み入力。内部ソフトウェアでプログラム可能なウィークプルアップ。
RB1	bit1	TTL	入出力ピン。内部ソフトウェアでプログラム可能なウィークプルアップ。
RB2	bit2	TTL	入出力ピン。内部ソフトウェアでプログラム可能なウィークプルアップ。
RB3	bit3	TTL	入出力ピン。内部ソフトウェアでプログラム可能なウィークプルアップ。
RB4	bit4	TTL	入出力ピン(変化により割込み)。内部ソフトウェアでプログラム可能なウィークプルアップ。
RB5	bit5	TTL	入出力ピン(変化により割込み)。内部ソフトウェアでプログラム可能なウィークプルアップ。
RB6	bit6	TTL/ST(2)	入出力ピン(変化により割込み)。内部ソフトウェアでプログラム可能なウィークプルアップ。シリアルプログラミングクロック。
RB7	bit7	TTL/ST(2)	入出力ピン(変化により割込み)。内部ソフトウェアでプログラム可能なウィークプルアップ。シリアルプログラミングデータ。

凡例： TTL=TTL入力、ST=シュミットトリガ

注 1： このバッファは外部割込みとして構成された場合にはシュミットトリガ入力となります。

2： このバッファはシリアルプログラミングモードで使用された場合にはシュミットトリガ入力となります。

表5-4: PORTBに関連するレジスタのまとめ

アドレス	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	BOR POR での値	他の リセット での値
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	uuuu uuuu
86h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	1111 1111
81h	OPTION	RBPU	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0	1111 1111	1111 1111

凡例： x = 未知、u = 不変。網掛けの部分は PORTB では使われません。

### 5.3 PORTC と TRISC レジスタ

適用デバイス							
710	71	711	72	73	73A	74	74A

PORTCは8ビットの双方向ポートです。各ピンはTRISCレジスタを通して個別に入力または出力として設定できます。PORTCはいくつかの周辺機能をマルチプレクスしています(表5-5参照)。PORTCピンにはシュミットトリガ入力バッファがあります。

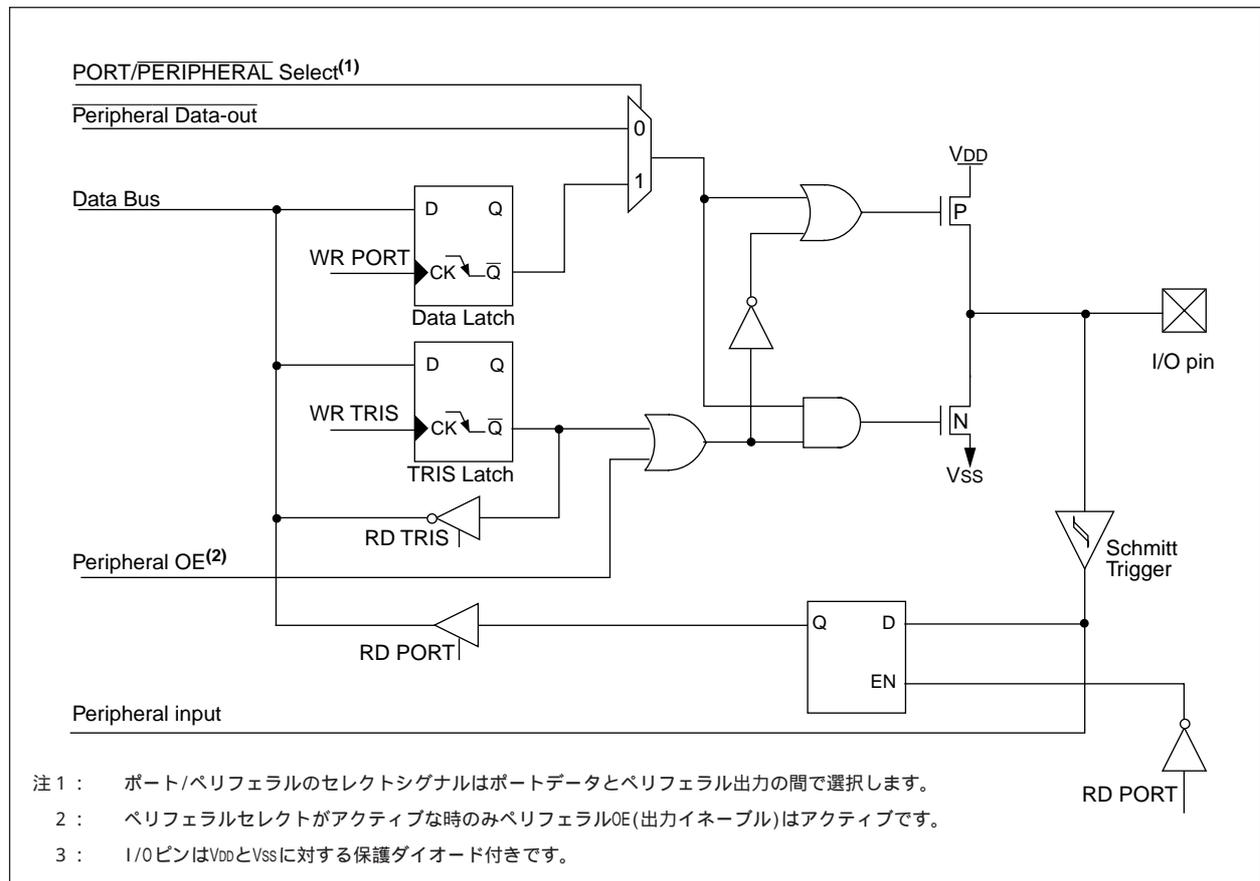
周辺機能をイネーブルする時、各PORTCピンに対してTRISビットを定義する必要があります。いくつかの周辺機能はピンを出力にさせるためにTRISビットをオーバーライドします。一方で他の周辺機能はピンを入力にさせるためにTRISビットをオーバーライドします。要するにTRISビットのオーバーライドは周辺機能がイネーブルされている間だけです。正しいTRISビットのセッティングについては、周辺機能の章を参照してください。

#### 例5-3: PORTCの初期化

```

CLRWF PORTC      ; Initialize PORTC by
                  ; setting output
                  ; data latches
BSF STATUS, RP0  ; Select Bank 1
MOVLW 0xCF       ; Value used to
                  ; initialize data
                  ; direction
MOVWF TRISC      ; Set RC<3:0> as inputs
                  ; RC<5:4> as outputs
                  ; RC<7:6> as inputs
    
```

図5-5: PORTCのブロック図(周辺出力オーバーライド)



# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。

## PIC16C7X

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

表5-5： PORTCの機能

名称	Bit#	Buffer Type	機能
RC0/T1OSO/T1CKI	bit0	ST	入出力ポートピンまたはタイマ1オシレータ出力/タイマ1クロック入力。
RC1/T1OSI/CCP2(1)	bit1	ST	入出力ポートピン、タイマ1オシレータ入力、キャプチャ2入力/コンペア2出力/PWM 2 出力。
RC2/CCP1	bit2	ST	入出力ポートピンまたはキャプチャ1入力/コンペア1出力/PWM1出力。
RC3/SCK/SCL	bit3	ST	RC3/SCK/SCLはSPI、I <sup>2</sup> Cの両モード用の同期シリアルクロックとしても選択可能。
RC4/SDI/SDA	bit4	ST	RC4/SDI/SDAは、SPIモードでSPIデータ入力、またはI <sup>2</sup> Cモードでデータ入出力としても選択可能。
RC5/SDO	bit5	ST	入出力ポートピンまたは同期シリアルポートのデータ出力。
RC6/TX/CK(2)	bit6	ST	入出力ポートピン、USART非同期トランスミットまたはUSART同期クロック。
RC7/RX/DT(2)	bit7	ST	入出力ポートピン、USART非同期レシーブまたはUSART同期データ。

凡例：ST=シュミット・トリガ

注 1： CCP2機能はPIC16C72では使用できません。

注 2： TX/CKとRX/DT機能はPIC16C72では使用できません。

表5-6： PORTCに関連するレジスタの要約

アドレス	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	BOR POR での値	他の リセット での値
07h	PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	uuuu uuuu
87h	TRISC	TRIS7	TRIS6	TRIS5	TRIS4	TRIS3	TRIS2	TRIS1	TRIS0	1111 1111	1111 1111

凡例：x = 未知、u = 不変

### 5.4 PORTD と TRISD レジスタ

適用デバイス							
710	71	711	72	73	73A	74	74A

PORTDはシュミットトリガ入力バッファを持った8ビットポートです。各ピンは個別に入力または出力として設定できます。

PORTDは、制御ビットPSPMODE(TRISE<4>)をセットすることにより、8ビット幅のマイクロプロセッサのポート(パラレルスレーブポート)として設定できます。このモードでは、入力バッファはTTLです。

図5-6: PORTDのブロック図 ( I/O ポートモード時)

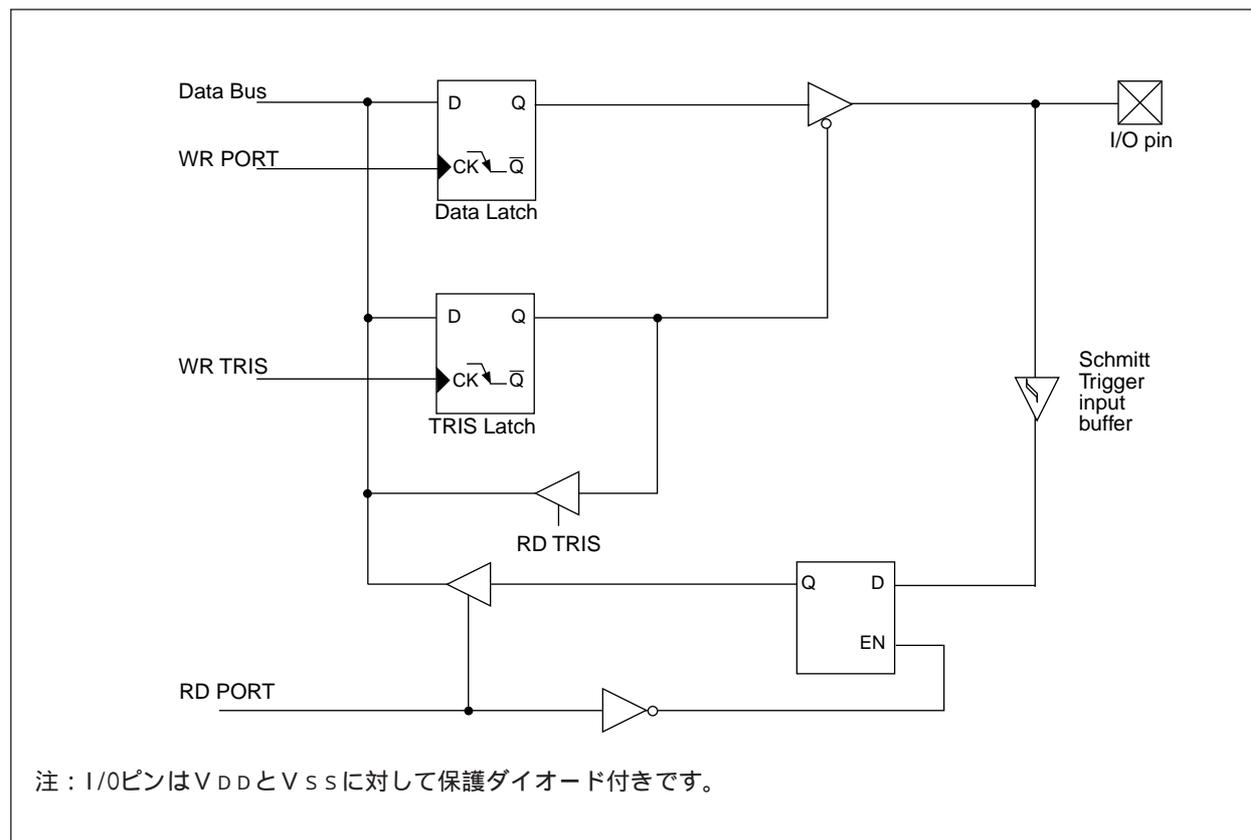


表5-7: PORTDの機能

名称	Bit #	Buffer Type	機能
RD0/PSP0	bit0	ST/TTL(1)	入出力ポートピン、またはパラレルスレーブポートのビット0
RD1/PSP1	bit1	ST/TTL(1)	入出力ポートピン、またはパラレルスレーブポートのビット1
RD2/PSP2	bit2	ST/TTL(1)	入出力ポートピン、またはパラレルスレーブポートのビット2
RD3/PSP3	bit3	ST/TTL(1)	入出力ポートピン、またはパラレルスレーブポートのビット3
RD4/PSP4	bit4	ST/TTL(1)	入出力ポートピン、またはパラレルスレーブポートのビット4
RD5/PSP5	bit5	ST/TTL(1)	入出力ポートピン、またはパラレルスレーブポートのビット5
RD6/PSP6	bit6	ST/TTL(1)	入出力ポートピン、またはパラレルスレーブポートのビット6
RD7/PSP7	bit7	ST/TTL(1)	入出力ポートピン、またはパラレルスレーブポートのビット7

凡例: ST=シュミットトリガ入力 TTL=TTL入力

注1: 入力バッファは、I/Oモード時にシュミットトリガで、パラレルスレーブポートモード時にはTTLバッファです。

# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。

## PIC16C7X

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

表 5-8: PORTDに関連するレジスタの要約

アドレス	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR BOR での値	他の リセット での値
08h	PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	xxxx xxxx	uuuu uuuu
88h	TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	1111 1111	1111 1111
89h	TRISE	IBF	OBF	I0V	PSPMODE	-	TRISE2	TRISE1	TRISE0	0000 -111	0000 -111

凡例: x = 未知、u = 不変、- = 未使用 リード時は'0'。網掛けの部分は PORTD では使用されません。

### 5.5 PORTE と TRISE レジスタ

適用デバイス							
710	71	711	72	73	73A	74	74A

PORTEには個別に入力または出力として設定できる3本のピン、すなわちRE0/RD/AN5、RE1/WR/AN6、RE2/CS/AN7があります。これらのピンにはシュミットトリガ入力バッファがあります。

I/O PORTEは、ビットPSPMODE(TRISE<4>)がセットされている時、マイクロプロセッサポートのための制御入力となります。このモードでは、TRISE<2:0>ビットがセットされている(ピンがデジタル入力として設定されている)ことと、レジスタADCON1がデジタルI/Oとして設定されていることを確認してください。このモードでは入力バッファはTTLです。

図5 - 7 に、パラレルスレーブポート動作も制御するTRISEレジスタを示します。

PORTEピンはアナログ入力とマルチプレクスされています。これらのピンの動作はADCON1レジスタの制御ビットによって選択されています。アナログ入力として選択する場合、これらのピンのリード時は‘0’として読み込みます。

TRISEはREピンがアナログ入力として使用されている場合でも、それらの方向を制御します。アナログ入力として使う場合、それらのピンを入力に設定して下さい。

注意：パワーオンリセットでは、これらのピンはアナログ入力として設定されています。

図5-7: TRISEレジスタ (アドレス 89h)

	R-0	R-0	R/W-0	R/W-0	U-0	R/W-1	R/W-1	R/W-1
	IBF	OBF	IBOV	PSPMODE	-	TRISE2	TRISE1	TRISE0
bit7								bit0
bit7 :	<b>IBF</b> : 入力バッファ・フル・ステータスビット 1=ワードが受信され、CPUによるリードを待機中の場合。 0=ワードが何も受信されていない場合。							
bit6 :	<b>OBF</b> : 出力バッファ・フル・ステータスビット 1=出力バッファに直前に入力されたワードがまだ保持されている場合。 0=出力バッファが読み込まれている場合。							
bit5 :	<b>IBOV</b> : (マイクロプロセッサモードでの)入力バッファオーバーフロー検出ビット 1=直前の入力ワードがまだ読み込まれていない時にライトが発生した場合 (ソフトウェアでクリアが必要)。 0=オーバーフローが発生しなかった場合。							
bit4 :	<b>PSPMODE</b> : 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							
bit3 :	未使用 : ‘0’ としてリード							
bit2 :	<b>TRISE2</b> : ピンRE2/CS/AN7用の方向制御ビット 1=入力 0=出力							
bit1 :	<b>TRISE1</b> : ピンRE1/WR/AN6用の方向制御ビット 1=入力 0=出力							
bit0 :	<b>TRISE0</b> : ピンRE0/RD/AN5用の方向制御ビット 1=入力 0=出力							

R = 読み込み可能なビット  
 W = 書き込み可能なビット  
 U = 未使用のビット、  
 ‘0’ としてリード  
 - n = PORリセットでの値

# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。  
<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

図5-8: PORTEのロック図 ( I/Oポートモード時 )

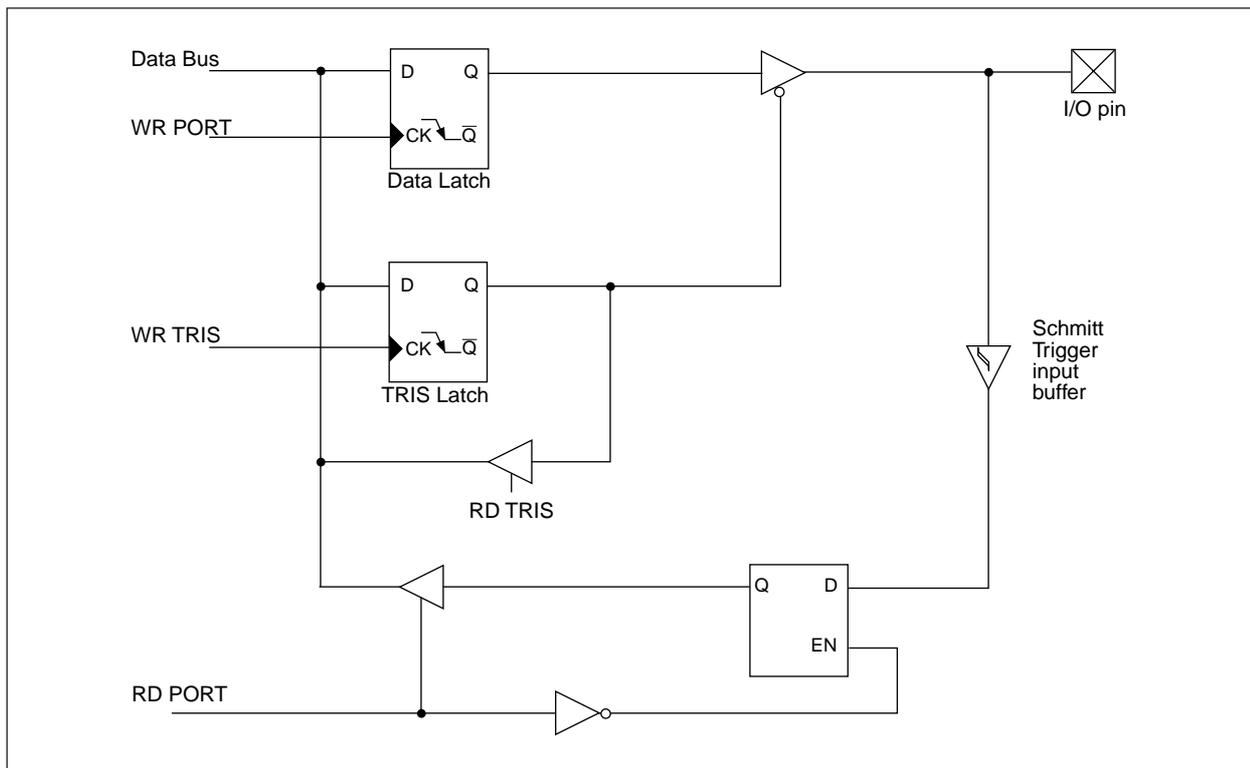


表5-9: PORTEの機能

名称	Bit #	Buffer Type	機能
RE0/RD/AN5	bit0	ST/TTL(1)	入出力ポートピン、パラレルスレーブポートにおけるリード制御入力またはアナログ入力: RD 1=リード操作を行わない 0=リード操作。PORTDレジスタを読み込む(チップセレクトがあるとき)。
RE1/WR/AN6	bit1	ST/TTL(1)	入出力ポートピン、パラレルスレーブポートにおけるライト制御入力またはアナログ入力: WR 1=ライト操作を行わない。 0=ライト操作。PORTDレジスタを書き込む(チップセレクトがあるとき)。
RE2/CS/AN7	bit2	ST/TTL(1)	入出力ポートピン、パラレルスレーブポートにおけるチップセレクト制御入力またはアナログ入力: CS 1=デバイスが選択されていない。 0=デバイスが選択されている。

凡例: ST=シュミットトリガ入力 TTL=TTL入力

注 1: 入力バッファは、I/Oモードではシュミットトリガ、パラレルスレーブポートモードではTTLバッファ。

表5-10: PORTEに関連するレジスタのまとめ

アドレス	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR, BOR での値	他のリセット での値
09h	PORTE	-	-	-	-	-	RE2	RE1	RE0	---- -000	---- -000
89h	TRISE	IBF	OBF	IBOV	PSPMODE	-	TRISE2	TRISE1	TRISE0	0000 -111	0000 -111
9fh	ADCON1	-	-	-	-	-	PCFG2	PCFG1	PCFG0	---- -000	---- -000

凡例: x=未知、u=不変、-=未使用、'0'としてリード。網掛けの部分はPORTEでは使われません。

### 5.6 I/Oプログラミングの注意点

適用デバイス							
710	71	711	72	73	73A	74	74A

#### 5.6.1 双方向I/Oポート

ライトのいくつかの命令は、内部でライト動作が後に続くリードのような動作をします。例えば、BCFとBSF命令はそのレジスタをCPUに読み込み、ビット動作を実行し、その結果をレジスタに書き戻します。これらの命令は、入力と出力が混在しているポートに使用する時は注意が必要です。例えば、PORTBのビット5へのBSF動作により、PORTBの8ビットすべてがCPUに読み込まれます。そして5ビット目をセットし、修正された8ビットのデータがPORTBの出力ラッチに書き込まれます。もしPORTBの0ビット目が双方向の入出力ピンとして使われた場合：0ビット目が入力として定義されているとき、このピンに与えられている入力信号がCPUに読み込まれ、データラッチに再び書き込まれ、前の内容はオーバーライトされてしまいます。そのピンが入力モードを保持している限り、問題は起こりませんが、ビット0が出力に切り変わった場合、データラッチの内容が不明になり、意図しないデータが出力されることがあります。

ポートレジスタのリードはポートピンの値を読み込むことです。ポートレジスタへのライトはポートラッチに値を書き込むことです。あるポートでリード・モディファイ・ライト命令(例えばBCF、BSFなど)を使用する場合、そのポートピンの値が読み込まれ、この値に対して求められる操作が行われ、その結果がポートラッチに書き込まれます。

例5-4に、I/Oポートでの2つのシーケンシャルなリード・モディファイ・ライト命令の影響を示します。

#### 例 5-4: I/Oポートでのリード・モディファイ・ライト命令

```

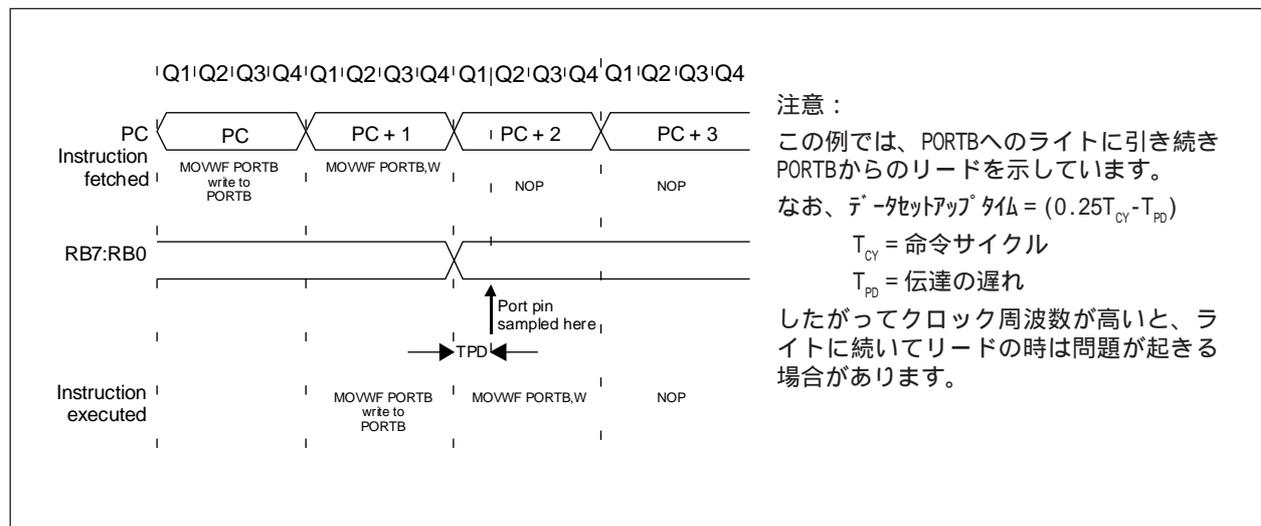
;Initial PORT settings:   PORTB<7:4> Inputs
;                           PORTB<3:0> Outputs
;PORTB<7:6> have external pull-ups and are
;not connected to other circuitry
;
;
;                           PORT latch  PORT pins
;                           -----
BCF PORTB, 7           ; 01pp ppp    11pp ppp
BCF PORTB, 6           ; 10pp ppp    11pp ppp
BSF STATUS, RP0       ;
BCF TRISB, 7           ; 10pp ppp    11pp ppp
BCF TRISB, 6           ; 10pp ppp    10pp ppp
;
;Note that the user may have expected the
;pin values to be 00pp ppp. The 2nd BCF
;caused RB7 to be latched as the pin value
;(high).
    
```

ローまたはハイを出力しているピンに対し、外部デバイスから駆動(ワイヤードOR、ワイヤードAND)すると、このピンのレベルが変化してしまうため、このような状態は避けて下さい。結果として、過大電流がチップにダメージを与えることがあります。

#### 5.6.2 I/Oポートの連続動作

I/Oポートへの実際のライトは命令サイクルの最後に起こりますが、リードに対しては、データは命令サイクルの始めに有効になっている必要があります(図5-9参照)。そのためにリード動作が後に続くライトが同じI/Oポートで実行された場合には注意が必要です。次の命令でCPUヘファイルのリードが実行される前に、ピンの電圧(負荷による)が安定できるような命令のシーケンスにする必要があります。そうでないと、そのピンの新しい状態ではなく前の状態がCPUに読み込まれてしまうことがあります。不確かな時はNOPまたはこのI/Oポートをアクセスしない他の命令を使用しポートをアクセスする命令が連続しないようにして下さい。

図5-9: 連続的なI/O動作



注意：  
 この例では、PORTBへのライトに引き続きPORTBからのリードを示しています。  
 なお、データセットアップタイム =  $(0.25T_{CY} - T_{PD})$   
 $T_{CY}$  = 命令サイクル  
 $T_{PD}$  = 伝達の遅れ  
 したがってクロック周波数が高いと、ライトに続いてリードの時は問題が起きる場合があります。

## PIC16C7X

### 5.7 パラレルスレーブポート

適用デバイス							
710	71	711	72	73	73A	74	74A

PORTDは制御ビットPSPMODE (TRISE<4>)がセットされた時、8ビット幅のパラレルスレーブポートあるいはマイクロプロセッサポートとして動作します。スレーブモードではRD制御入力ピンRE0/RD/AN5とWR制御入力ピンRE1/WR/AN6を通して外部から非同期にリードとライトができます。

これは8ビットのマイクロプロセッサデータバスに直接インターフェイスできます。外部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リードとライトができます。ビットPSPMODEをセットすることにより、ポートピンRE0/RD/AN5をRD入力、RE1/WR/AN6をWR入力、RE2/CS/AN7をCS(チップセレクト)入力にできます。この機能のためには、TRISEレジスタ (TRISE<2:0>)に対応するデータ方向ビットを入力(セット)として設定する必要があります。またそのA/Dポート設定ビットPCFG2:PCFG0 (ADCON1<2:0>)をセットする必要があり、デジタルI/OとしてRE2:RE0ピンを設定します。

実際には2個の8ビットラッチがあり、1個はデータ出力用(PIC16/17からの)、もう1個はデータ入力用です。PORTDのデータラッチに8ビットデータを書き込み、ポートピンのラッチからデータを読み込みます(それらは同じアドレスであることに注意)。

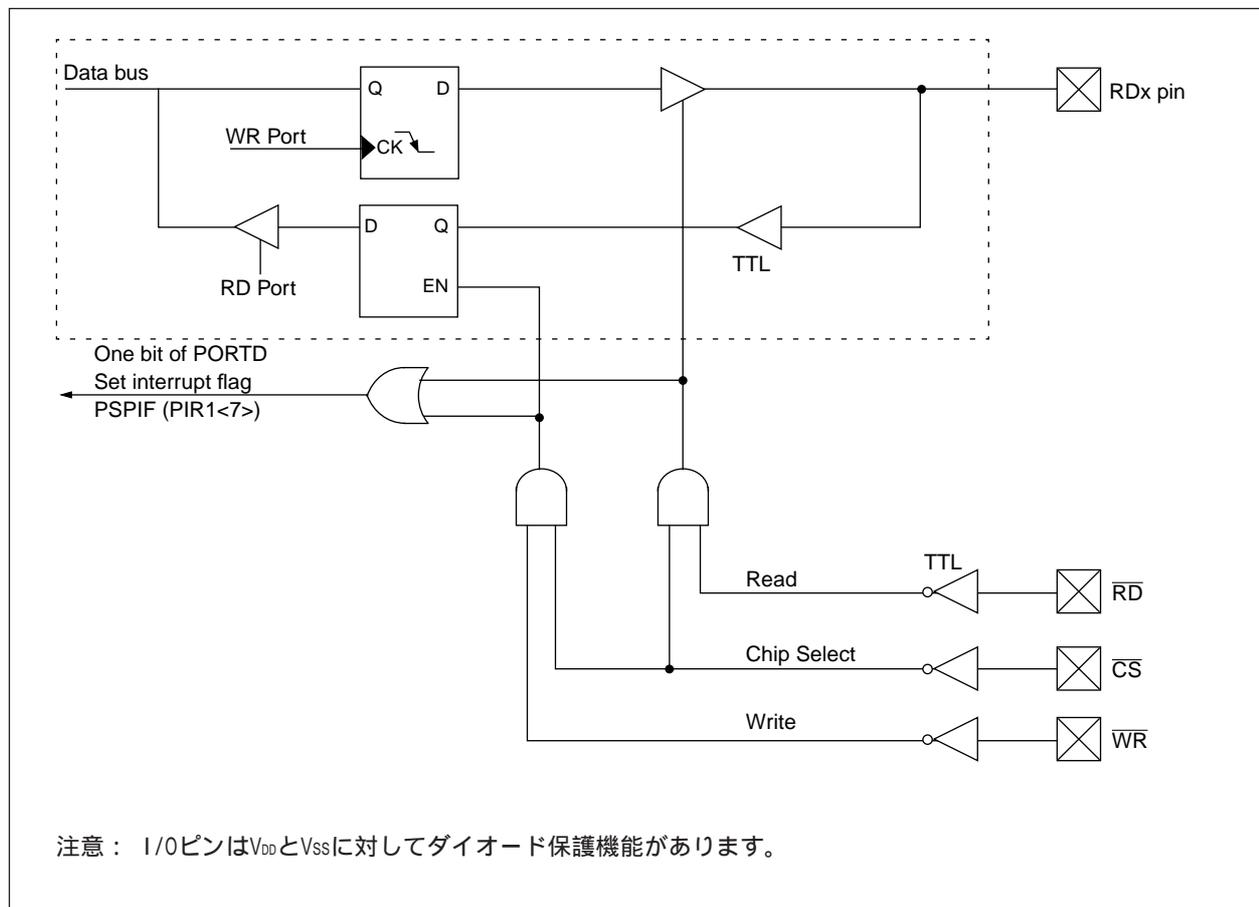
このモードでは、マイクロプロセッサがデータフローの方向を制御しているので、TRISDレジスタは無視されます。

受信したワードがCPUに読み込まれるのを待っている場合は、入力バッファフルステータスフラグビットIBF (TRISE<7>)がセットされます。PORTD入力ラッチが読み込まれると、IBFはクリアされます。IBFはリードのみのステータスビットです。PORTDラッチに書き込まれたワードが外部バスから読み込まれるのを待っている場合は、出力バッファフルステータスフラグビットOBF (TRISE<6>)がセットされます。PORTD出力ラッチがマイクロプロセッサに読み込まれると、OBFはクリアされます。前のワードがCPUに読み込まれなかった時に(1番目のワードはバッファの中にあるままです)、2番目がマイクロプロセッサのポートに書き込まれた場合は、入力バッファオーバーステータスフラグビットIBOV (TRISE<5>)がセットされます。

パラレルスレーブポートモードでない時は、IBFとOBFのビットはクリアのまま保持されます。しかしフラグビットIBOVが前もってセットされていた場合は、ソフトウェアでクリアする必要があります。

リードまたはライト動作が完了した時、割込みが発生し、フラグビットPSPIF (PIR1<7>)にラッチされます。割込みフラグビットPSPIFはユーザソフトウェアでクリアする必要があり、その割込みを割込みイネーブルビットPSPIE (PIE1<7>)をクリアすることによりディセーブルできます。

図5-10: PORTDとPORTEのブロック図 (パラレルスレーブポート)



# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。  
<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

表5-11： パラレルスレーブポートに関連するレジスタ

アドレス	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR BOR での値	その他の リセット での値
08h	PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	xxxx xxxx	uuuu uuuu
09h	PORTE	-	-	-	-	-	RE2	RE1	RE0	---- -000	---- -000
89h	TRISE	IBF	OBF	IBOV	PSPMODE	-	TRISE2	TRISE1	TRISE0	0000 -111	0000 -111
0Ch	PIR1	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
9fh	ADCON1	-	-	-	-	-	PCFG2	PCFG1	PCFG0	---- -000	---- -000

凡例： x = 未知、u =不変、 - = 未使用、 '0' としてリード。網掛けの部分はパラレルスレーブポートでは使用されません。

# 暫定版

英語最新版データシートと併用しご利用下さい。

AC/DC仕様は最新英文データシートをご参照下さい。

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

---

---

### NOTES:

### 6.0 タイマモジュールの概要

適用デバイス							
710	71	711	72	73	73A	74	74A

PIC16C710/71/711には1個のタイマモジュールがあり、PIC16C72、PIC16C73/73A、PIC16C74/74Aには3個のタイマモジュールがあります。

各モジュールはイベント(タイマオーバーフロー)が起こったことを示すために割込みを発生できます。それぞれのモジュールの詳細については、次の章で説明します。タイマ・モジュールには下記の種類があります。

- ・タイマ0 (TMR0)モジュール(7章参照)
- ・タイマ1 (TMR1)モジュール(8章参照)
- ・タイマ2 (TMR2)モジュール(9章参照)

#### 6.1 タイマ0の概要

適用デバイス							
710	71	711	72	73	73A	74	74A

タイマ0モジュール(以前の名称はRTCC)は、8ビットのオーバーフローカウンタです。クロックのソースは内部システムクロック(Fosc/4)または外部クロックのどちらでも可能です。クロックのソースが外部クロックの時、タイマ0モジュールはインクリメントするために立ち上がりエッジまたは立ち下がりエッジどちらでも選択できます。

タイマ0モジュールにはプログラム可能なプリスケラがあります。このプリスケラはタイマ0モジュールまたはウォッチドッグタイマのいずれかに設定できます。PSAビット(OPTION<3>)がプリスケラを指定し、PS2:PS0ビット(OPTION<2:0>)がプリスケラ値を決定します。タイマ0は次のような比でインクリメントできます。1:1(プリスケラがウォッチドッグタイマに指定された時)、1:2、1:4、1:8、1:16、1:32、1:64、1:128、1:256(タイマ0のみ)。

外部クロックの同期は、プリスケール後に起こります。プリスケールが使われるとき、外部クロック周波数はデバイスの周波数より高いことがあります。最大周波数はクロックに要求されたHiとLow幅において、50MHzです。

#### 6.2 タイマ1の概要

適用デバイス							
710	71	711	72	73	73A	74	74A

タイマ1は16ビットのタイマ/カウンタです。クロックのソースは内部システムクロック(Fosc/4)、外部クロック、外部クリスタルのいずれかで可能です。タイマ1はタイマまたはカウンタのどちらでも動作できます。カウンタとして動作している時(外部クロックソース)、そのカウンタはデバイスに対して同期、非同期どちらでも動作できます。非同期動作によりタイマ1はスリープ中も動作させることができ、SLEEPモードでの電力節約と同時にリアルタイムクロックを必要とする応用に便利です。

タイマ1にはプリスケラオプションがあり、タイマ1は次のような比でインクリメントすることができます。

1:1、1:2、1:4、1:8。タイマ1をキャプチャ/コンペア/PWMモジュールと共に使うことができます。CCPモジュールと共に使われた時は、タイマ1は16ビットのキャプチャ用または16ビットのコンペア用のタイムベースになり、デバイスに同期している必要があります。

#### 6.3 タイマ2の概要

適用デバイス							
710	71	711	72	73	73A	74	74A

タイマ2は、8ビットの周期レジスタ(PR2)と同時にプログラム可能なプリスケラとポストスケラを持つ8ビットのタイマです。タイマ2は同期シリアルポート(SSP)用のボーレートジェネレータと同時にCCP1モジュール(PWMモードにおいての)と併用することができます。プリスケラオプションによりタイマ2は次のような比でインクリメントできます。1:1、1:4、1:16。

ポストスケラにより、TMR2レジスタは割込みが発生する前にプログラム可能な回数と周期レジスタ(PR2)を一致させることができます。ポストスケラは1:1から1:16の比でプログラムできます。

#### 6.4 CCPの概要

適用デバイス							
710	71	711	72	73	73A	74	74A

CCPモジュールは次の3種類のモードのいずれか1つが動作します。16ビットのキャプチャ、16ビットのコンペア、10ビットのPWM。

キャプチャモードはTMR1の16ビットの値をCCPRxH:CCPRxLレジスタに取り込みます。そのキャプチャイベントはCCPxピンの立ち下がりエッジ、立ち上がりエッジ、4番目の立ち上がりエッジ、16番目の立ち上がりエッジのいずれかにプログラムできます。

コンペアモードはTMR1H:TMR1LレジスタとCCPRxH:CCPRxLレジスタを比較します。その一致が起こると、割込みを発生させることができ、出力ピンCCPxを与えられた状態(HiまたはLo)に強制的にすることができ、TMR1をリセットすることができます(CCP1)。あるいはタイマ1はA/D変換(CCP2)をリセットとスタートすることができます。これは制御ビットCCPxM3:CCPxM0によります。

PWMモードはTMR2レジスタを8ビットの周期レジスタ(PR2)と同時に10ビットのデューティサイクルレジスタ(CCPxH:CCPxL<5:4>)と比較します。TMR2レジスタ=デューティサイクルレジスタの時、CCPxピンは強制的にローになります。TMR2=PR2の時、TMR2は00hにクリアされ、CCPxピンは(出力の時)強制的にHiになり、割込みを発生させることができます。

# 暫定版

英語最新版データシートと併用しご利用下さい。

AC/DC仕様は最新英文データシートをご参照下さい。

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

---

### NOTES:

### 7.0 タイマ0モジュール

適用デバイス							
710	71	711	72	73	73A	74	74A

タイマ0モジュールのタイマ/カウンタには次のような特徴があります。

- ・ 8ビットのタイマ/カウンタ
- ・ リードとライトが可能
- ・ 8ビットのプログラム可能なプリスケアラ
- ・ 内部または外部のクロックの選択
- ・ FFh から 00h へのオーバーフロー割込み
- ・ 外部クロックのエッジの選択

図7-1はタイマ0モジュールの概略ブロック図です。

ビットTOCS(OPTION<5>)をクリアしてタイマモードを選択します。タイマモードでは、タイマ0モジュールは命令サイクル毎に(プリスケアラなしで)インクリメントします。TMR0レジスタに書き込んだ場合、続く2命令サイクルに対してインクリメントが禁止されます(図7-2、7-3参照)。TMR0レジスタにこれを考慮した値を書き込むことにより、回避できます。

TOCSビット(OPTION<5>)をセットすることによりカウンタモードを選択します。カウンタモードでは、タイマ0はRA4/T0CKIピンの立ち上がりエッジまたは立ち下がりエッジのどちらでもインクリメントします。インクリメントするエッジはタイマ0ソースエッジ選択ビットTOSE(OPTION<4>)によって決定されます。TOSE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。外部クロック入力の制限に関しては7.2章で詳しく説明します。

プリスケアラはタイマ0モジュールとウォッチドッグタイマの間で排他的に共用されます。プリスケアラの設定は制御ビットPSA(OPTION<3>)によりソフトウェアで制御されます。PSAビットをクリアするとプリスケアラをタイマ0モジュールに設定します。プリスケアラはリードとライトはできません。プリスケアラがタイマ0モジュールに設定された場合、プリスケアラ値は1:2, 1:4, ..., 1:256の中から選択できます。プリスケアラの動作についての詳細は7.3章で説明します。

#### 7.1 タイマ0割込み

適用デバイス							
710	71	711	72	73	73A	74	74A

TMR0レジスタがFFhから00hにオーバーフローした時、TMR0割込みが発生します。このオーバーフローがT0IF(INTCON<2>)をセットします。T0IEビット(INTCON<5>)をクリアすることにより割込みをマスクすることができます。この割込みを再びイネーブルする前に、タイマ0モジュール割込みサービスルーチンによりソフトウェアでT0IFビットをクリアしておかなければなりません。SLEEPの間タイマが停止しているため、TMR0割込みでプロセッサをSLEEPからウェイクすることはできません。タイマ0割込みのタイミングについては図7-4をご覧ください。

図7-1: タイマ0のブロック図

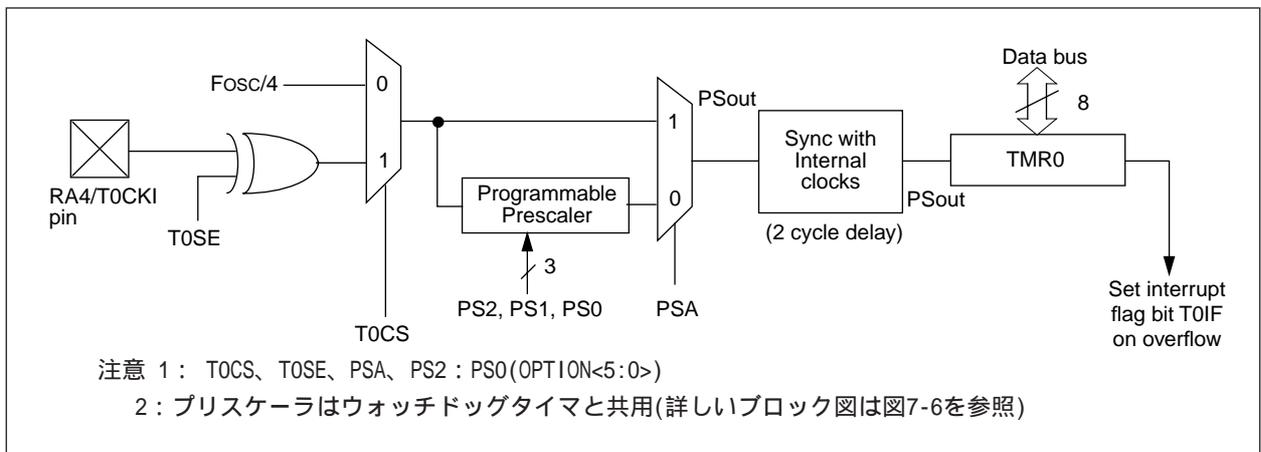
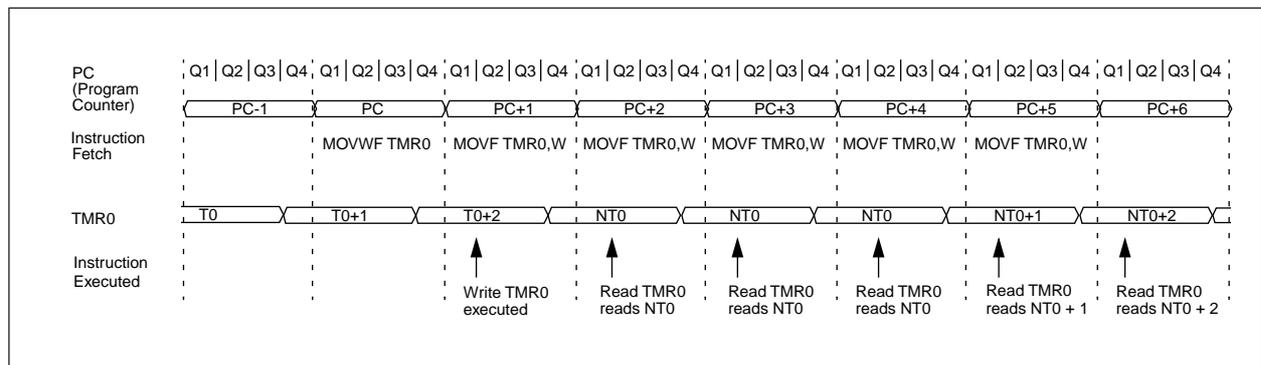


図 7-2: タイマ0のタイミング: 内部クロック / プリスケールなし



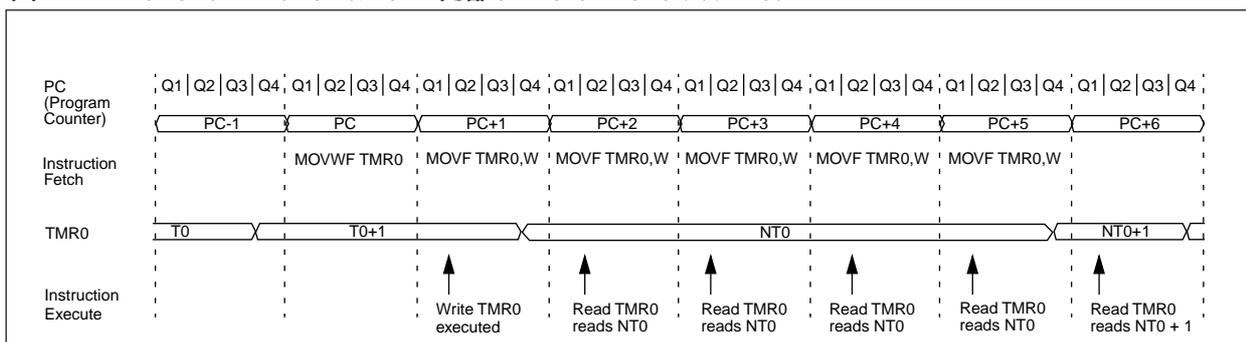
# 暫定版

英語最新版データシートと併用にご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。

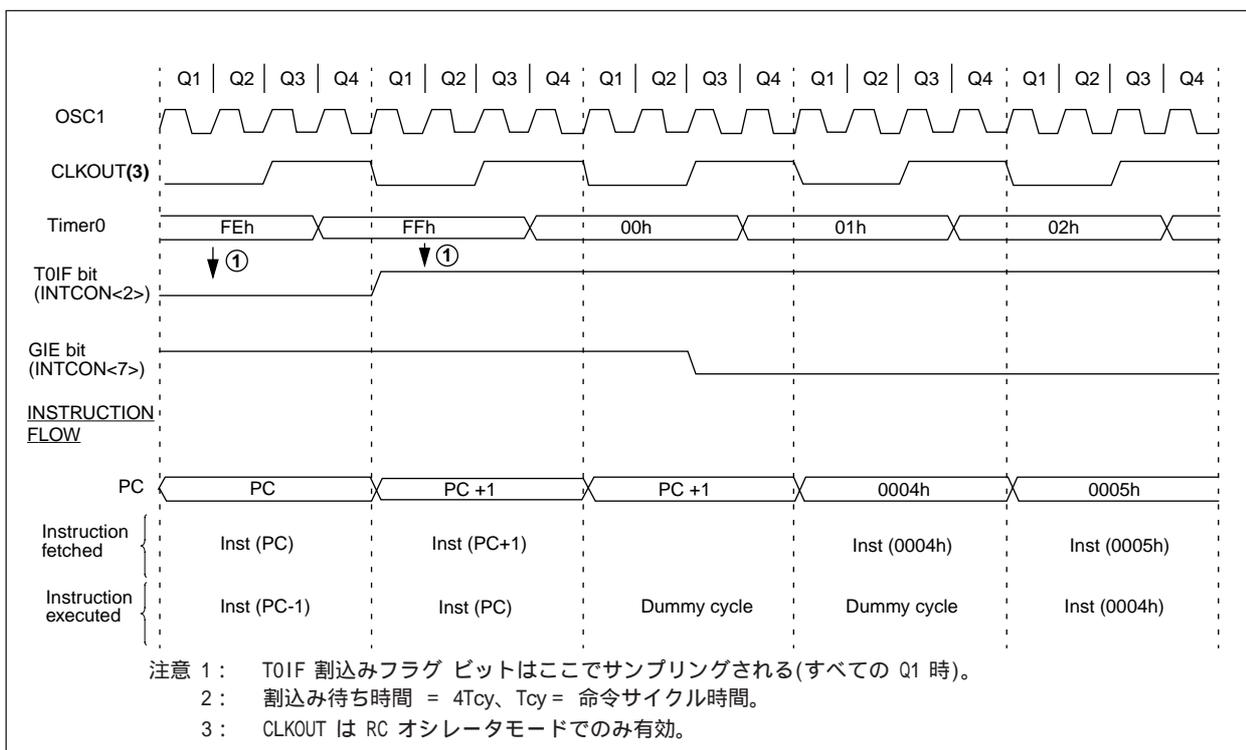
## PIC16C7X

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

**図7-3： タイマ0のタイミング：内部クロック/プリスケール 1：2**



**図7-4： タイマ0 割込みのタイミング**



### 7.2 外部クロックによるタイマ0の使い方

適用デバイス							
710	71	711	72	73	73A	74	74A

外部クロック入力（TIM0）に使われている時、正確に必要な条件を満たさなければなりません。その必要条件は外部クロックが内部位相クロック（ $T_{osc}$ ）に同期できることで確実にあります。また同期後、タイマ0の実際のインクリメントまで少しの遅れがあります。

プリスケータを使用する時、そのプリスケータ出力は対称性なので、外部クロック入力（非同期リップルカウンタ型プリスケータ）により分周されています。外部クロックがサンプリングの必要条件を満たすためには、リップルカウンタに注意しなければなりません。したがって $T_{OCLKI}$ は最低  $4T_{osc}$ （40nsの微少 RC 遅延時間）の周期を持つためにプリスケータ値で分ける必要があります。 $T_{OCLKI}$  のハイタイムとロータイムの必要条件だけが、10nsの最低パルス幅の条件にそむかないということです。必要なデバイスの電気的仕様については、パラメータ 40、41、42を参照してください。

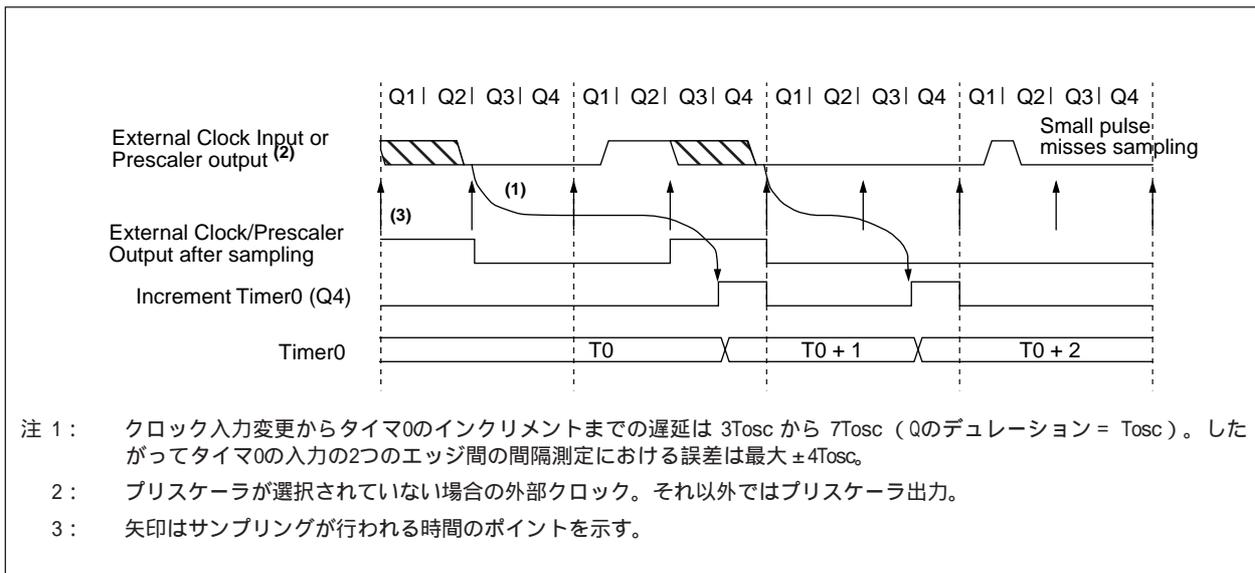
#### 7.2.1 外部クロック同期

プリスケータを使用しない時、外部クロック入力（TIM0）はプリスケータ出力と同じです。内部位相クロックを伴う $T_{OCLKI}$ の同期は内部位相クロックのQ2とQ4サイクル上でプリスケータ出力をサンプリングすることにより完了します（図7-5参照）。したがって、 $T_{OCLKI}$ は最低 $2T_{osc}$ の間ハイ（20nsの微少RC遅延時間）と、最低 $2T_{osc}$ の間ロー（20nsの微少RC遅延時間）になっている必要があります。必要なデバイスの電気的仕様を参照してください。

#### 7.2.2 TMR0のインクリメントの遅延時間

プリスケータ出力が内部クロックに同期しているため、外部クロックのエッジが起こった時からタイマ0モジュールが実際にインクリメントする時まで少しの遅れがあります。図7-5に外部クロックのエッジからタイマがインクリメントするまでの遅れを示します。

図7-5： 外部クロックによるタイマ0のタイミング



### 7.3 プリスケーラ

適用デバイス							
710	71	711	72	73	73A	74	74A

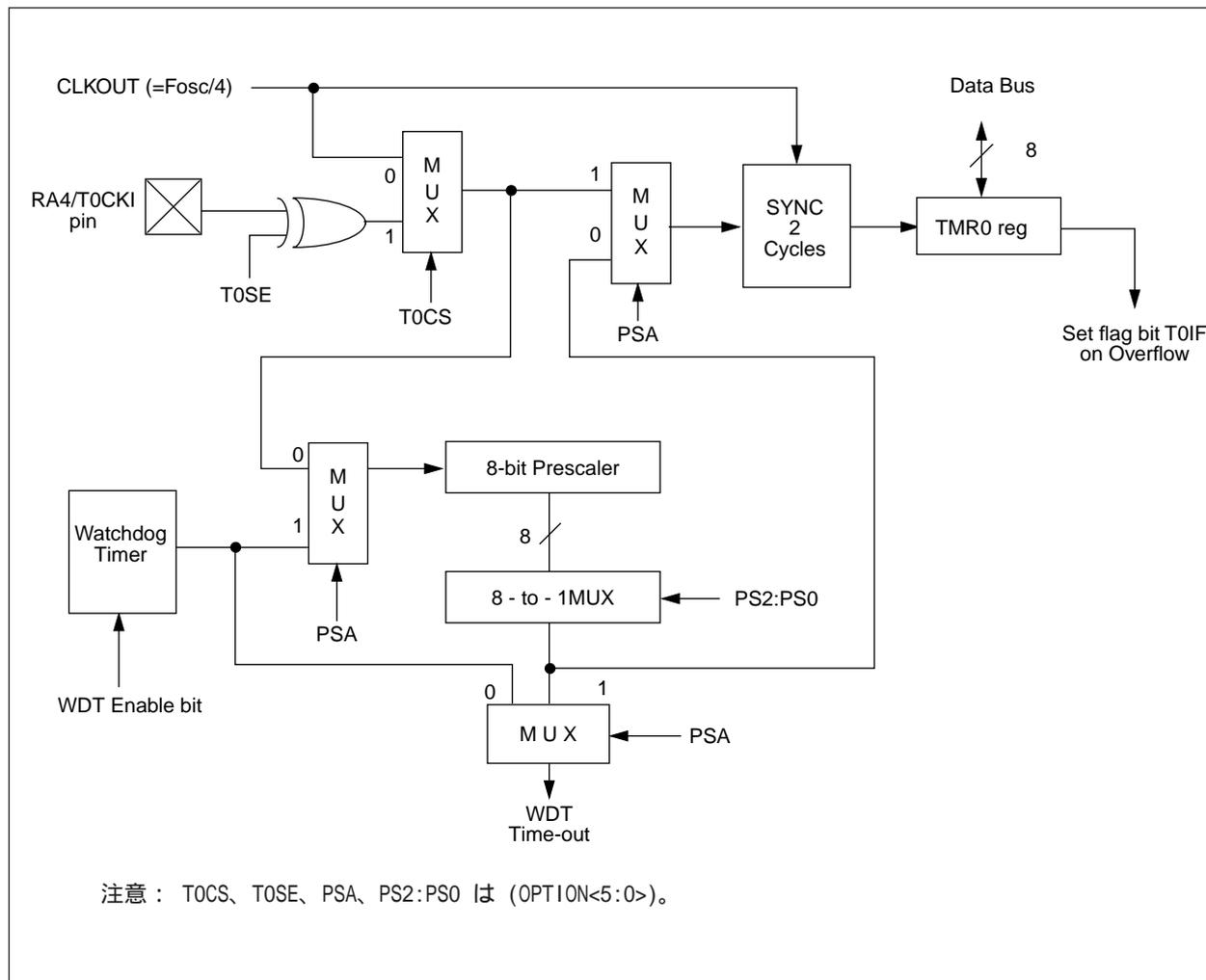
タイマ0モジュール用のプリスケーラ、またはウォッチドッグタイマ用のポストスケーラとして、それぞれに8ビットのカウンタがあります(図7-6参照)。分かりやすくするためにこのデータシートでは、このカウンタを“プリスケーラ”と表わします。プリスケーラは1個あるだけで、タイマ0モジュールとウォッチドッグタイマとで互いに排他的に共用されていることに注意してください。つまりタイマ0モジュールにプリスケーラが指

定されると、ウォッチドッグタイマ用のプリスケーラはなくなります。またこの逆のことも言えます。

PSA と PS2:PS0 ビット (OPTION<3:0>) がプリスケーラの指定とプリスケール比を決定します。

タイマ0モジュールに指定されると、TMR0レジスタに書き込むすべての命令(例えば、CLRWF1、MOVWF1、BSF1、x・・・など)はプリスケーラをクリアします。WDTに指定されると、CLRWDT命令はウォッチドッグタイマと一緒にプリスケーラをクリアします。プリスケーラはリードとライトはできません。

**図7-6: タイマ0/WDT プリスケーラのブロック図**



# 暫定版

英語最新版データシートと併用ご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。  
<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

### 7.3.1 プリスケラ指定の切り替え

プリスケラの指定はソフトウェアによって完全に制御されます。すなわちプログラム実行中に変更することができます。

注意： 不慮のデバイス RESET を避けるために、プリスケラ指定をタイマ0から WDT に変更する時、次のような命令のシーケンス(例7-1参照)を実行する必要があります。このシーケンスは WDT がディセーブルされた場合でも必要です。

プリスケラを WDT からタイマ0モジュールに変更する時は、例 7-2 に示すシーケンスを使います。

### 例7-2 : プリスケラの変更 (WDT タイマ0)

```
CLRWDT          ;Clear WDT and
                ;prescaler
BSF             STATUS, RP0 ;Bank 1
MOVLW          b'xxxx0xxx' ;Select TMR0, new
                ;prescale value and
MOVWF          OPTION      ;clock source
BCF             STATUS, RP0 ;Bank 0
```

### 例7-1 : プリスケラの変更(タイマ0 WDT)

```
BCF             STATUS, RP0 ;Bank 0
CLRWF          TMR0        ;Clear TMR0 & Prescaler
BSF             STATUS, RP0 ;Bank 1
CLRWDT          ;Clears WDT
MOVLW          b'xxxxlxxx' ;Select new prescale
MOVWF          OPTION      ;value & WDT
BCF             STATUS, RP0 ;Bank 0
```

表7-1 : タイマ0に関連するレジスタ (PIC16C710/71/711)

アドレス	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR BOR での値	他の リセット での値
01h	TMR0	タイマ0モジュールのレジスタ								xxxx xxxx	uuuu uuuu
0Bh/8Bh	INTCON	GIE	ADIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
81h	OPTION	RBPU	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
85h	TRISA	-	-	-	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	---1 1111	---1 1111

凡例： x = 未知、u = 不変、- = 未使用のロケーション、'0'としてリード。網掛けの部分はタイマ0では使用されません。

表7-2 : タイマ0に関連するレジスタ(PIC16C72/73/73A/74/74A)

アドレス	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR BOR での値	他の リセット での値
01h	TMR0	タイマ0モジュールのレジスタ								xxxx xxxx	uuuu uuuu
0Bh/8Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
81h	OPTION	RBPU	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
85h	TRISA	-	-	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111

凡例： x = 未知、u = 不変、- = 未使用のロケーション、'0'としてリード。網掛けの部分はタイマ0では使用されません。

# 暫定版

英語最新版データシートと併用しご利用下さい。

AC/DC仕様は最新英文データシートをご参照下さい。

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

---

### NOTES:

### 8.0 タイマ1 モジュール

適用デバイス							
710	71	711	72	73	73A	74	74A

タイマ1は、リードとライトができる2個の8ビットレジスタ (TMR1HとTMR1L) で構成されている16ビットのタイマ / カウンタです。TMR1レジスタのペア (TMR1H+TMR1L) は、0000h からFFFFh までインクリメントでき、さらに0000h へ戻ります。TMR1の割込みは、イネーブルの時オーバーフローで発生し、割込みフラグビットTMR1IF (PIR1<0>) にラッチされます。この割込みは TMR1 割込みイネーブルビット TMR1IE (PIE1<0>) を使ってイネーブルまたはディセーブルできます。

タイマ1は次の2つのモードのどちらか1つで動作することができます。

- ・タイマとして
- ・カウンタとして

この動作モードはクロック選択ビットTMR1CS (T1CON<1>) によって決められます。

タイマモードでは、タイマ1 が命令サイクル毎にインクリメントします。カウンタモードでは、ピンRC0 / T10S0/T1CKIの外部クロック入力の立ち上がりエッジ毎にインクリメントします。

タイマ1は制御ビットTMR1ON (T1CON<0>) を使って、オンとオフすることができます。

さらにタイマ1には内部の“リセット入力”があります。このリセットは2つのCCPモジュール(10.0章参照)のどちらかによって発生します。図 8-1 にタイマ1制御レジスタを示します。

PIC16C72/73A/74Aでは、タイマ1のオシレータがイネーブルの時 (T10SCENがセットされている)、RC1/T10S1/CCP2とRC0/T10S0/T1CKIピンは入力になります。つまりTRISC<1:0>の値は無視されます。

PIC1673/74 では、タイマ1のオシレータがイネーブルの時 (T10SCENがセットされている)、RC1/T10S1/CCP2ピンは入力になります。しかし RC0/T10S0/T1CKI ピンは、TRISC<0>ビットをセットすることにより入力として設定する必要があります。

図8-1: T1CON: タイマ1の制御レジスタ(アドレス 10h )

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
-	-	T1CKPS1	T1CKPS0	T10SCEN	T1SYNC	TMR1CS	TMR1ON	
bit7								bit0
bit7-6: <b>未使用</b> : 0 としてリード								
bit5-4: <b>TICKPS1: TICKPS0</b> : タイマ1入力クロックプリスケール選択ビット								
11 = 1:8 プリスケール値								
10 = 1:4 プリスケール値								
01 = 1:2 プリスケール値								
00 = 1:1 プリスケール値								
bit3: <b>T10SCEN</b> : タイマ1オシレータイネーブル制御ビット								
1 = オシレータはイネーブル								
0 = オシレータはオフ								
注意: オシレータインバータとフィードバック抵抗はパワードレインを消去するためオフになります。								
bit2: <b>T1SYNC</b> : タイマ1外部クロック入力同期制御ビット								
TMR1CS = 1								
1 = 外部クロック入力と非同期								
0 = 外部クロック入力と同期								
TMR1CS = 0								
注意: このビットは無視される。TMR1CS = 0 の時、タイマ1は外部クロックを使う。								
bit1: <b>TMR1CS</b> : タイマ1クロックソース選択ビット								
1 = RC0/T10S0/T1CKI ピンからの外部クロック(立ち上がりエッジ上)								
0 = 内部クロック (OSC/4)								
bit0: <b>TMR1ON</b> : タイマ1オンビット								
1 = タイマ1をイネーブル								
0 = タイマ1を停止								

R = 読み込み可能なビット  
W = 書き込み可能なビット  
U = 未使用のビット、  
'0' としてリード  
- n = PORリセットでの値

### 8.1 タイムモードでのタイマ1の動作

適用デバイス							
710	71	711	72	73	73A	74	74A

TMR1CS (T1CON<1>)ビットをクリアすることにより、タイマモードを選択します。このモードでは、タイマへの入力クロックはOSC/4です。内部クロックは常に同期しているので、同期制御ビットT1SYNC (T1CON<2>)は影響ありません。

### 8.2 同期カウンタモードでのタイマ1の動作

適用デバイス							
710	71	711	72	73	73A	74	74A

TMR1CS ビットをセットすることにより、カウンタモードを選択します。このモードでは、タイマはビットT10SCEN がセットされる時はピンRC1/T1OSI/CCP2、またビット T10SCEN がクリアされる時はピンRC0/T1OSO/T1CKI 上で、クロック入力の立ち上がりエッジ毎にインクリメントされます。

T1SYNC がクリアされると、外部クロック入力は内部位相クロックと同期されます。この同期はプリスケアラ回路の後で行われます。プリスケアラ回路は非同期リップルカウンタです。

この設定では、SLEEP モードの間、タイマ1は同期回路がオフになっているので外部クロックがあってもインクリメントしません。しかしプリスケアラはインクリメントを継続します。

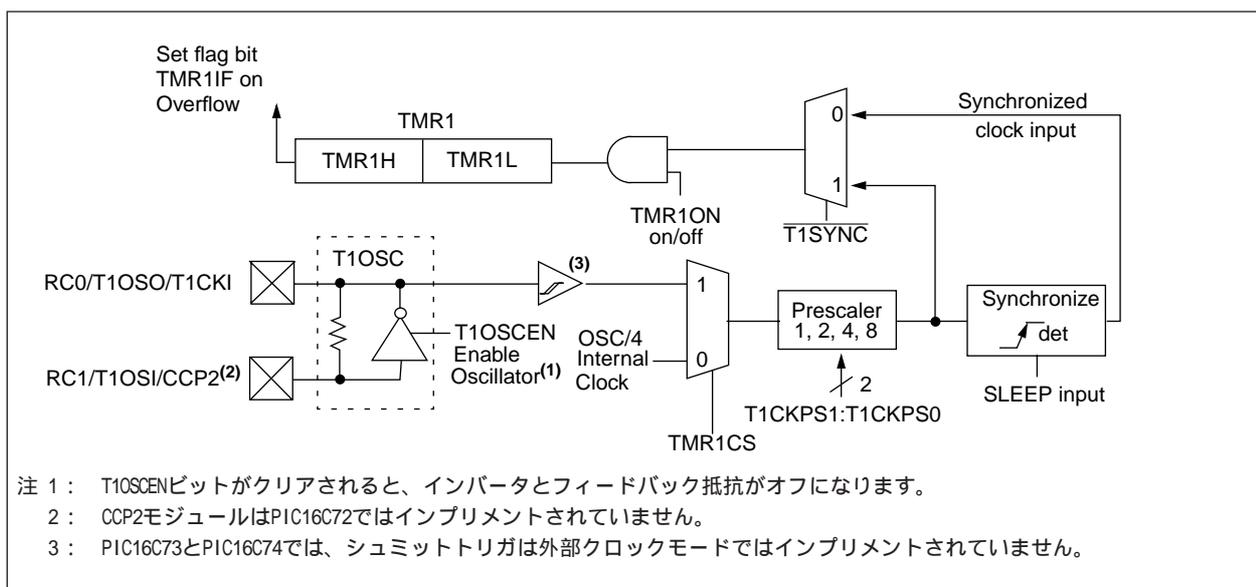
### 8.2.1 同期カウンタモードでの外部クロック入力のタイミング

外部クロック入力が同期カウンタモードでタイマ1として使用されている場合、正確に必要な条件を満たさなければなりません。外部クロックの必要条件は内部位相クロック(Tosc)同期によります。また同期後 TMR1 の実際のインクリメントまでに少しの遅れがあります。

プリスケアラが 1:1 の時、外部クロック入力はプリスケアラ出力と同じです。内部位相クロックを伴う T1CKI の同期は内部位相クロックの Q2 と Q4 サイクル上でプリスケアラ出力をサンプリングすることにより完成します。したがって、T1CKI は最低2Toscの間ハイ(20nsの微少RC遅延時間)と最低2Toscの間ロー(20nsの微少RC遅延時間)になっている必要があります。適切な電気的仕様について、パラメータ 45、46、47 を参照してください。

1:1 以外のプリスケアラが使われている場合、外部クロック入力は非同期リップルカウンタ型プリスケアラにより分周されているので、そのプリスケアラ出力は対称です。外部クロックがサンプリングの必要条件を満たすためには、リップルカウンタに注意しなければなりません。したがって T1CKI は最低 4Tosc (40nsの微少RC遅延時間)の周期をプリスケアラ値で分ける必要があります。最小10nsのパルス幅が、T1CKI のハイ・タイムとロー・タイムに必要です。適切な電気的仕様について、パラメータ 40、42、45、46、47 を参照してください。

図8-2: タイマ1のブロック図



### 8.3 非同期カウンタモードでのタイマ1動作

適用デバイス							
710	71	711	72	73	73A	74	74A

制御ビット  $\overline{T1SYNC}$  (T1CON<2>) がセットされている場合、外部クロック入力は同期されません。タイマは内部相クロックに対して非同期にインクリメントを継続します。タイマは SLEEP の間動作を継続し、オーバーフロー時に割込みを発生し、プロセッサをウェークさせます。しかしタイマをリード/ライトするためには、ソフトウェアでの特別な注意が必要です ( 8.3.2 章参照)。

非同期カウンタモードでは、タイマ1 をキャプチャまたはコンペア動作のタイムベースとして使うことはできません。

#### 8.3.1 非同期クロックによる外部クロック入力のタイミング

制御ビット  $\overline{T1SYNC}$  がセットされている場合、タイマは完全に非同期でインクリメントします。入力クロックは、正確に最低限のハイ・タイムとロー・タイム条件を満たす必要があります。適切な電氣的仕様について、タイミングパラメータ 45、46、47 を参照してください。

#### 8.3.2 非同期カウンタモードでのタイマ1のリードとライト

タイマが外部非同期クロックで動作している間の TMR1H または TMR1L のリードは有効なリードを保証します(ハードウェアでの保証が条件)。しかし16ビットのタイマを2個の8ビット値自身で読み込むと、タイマが両方のリードの間でオーバーフローすることがあり、問題が生じることに注意する必要があります。

ライトでは、単にタイマを停止して、必要な値を書き込むことをお勧めします。レジスタがインクリメントしている間にそのタイマレジスタに書き込むと書き込み衝突が起こることがあります。これによってタイマレジスタに予知できない値を作ることになります。

16ビット値を読み込む場合には注意が必要です。例8-1は16ビットのタイマ値を読み込むルーチン例です。タイマを停止できない場合にはこれが便利です。

### 例8-1: 16ビットフリーランタイムのリード

インターラプトはディセーブル

```

MOVWF TMR1H, W ;Read high byte
MOVWF TMPH ;
MOVWF TMR1L, W ;Read low byte
MOVWF TMPL ;
MOVWF TMR1H, W ;Read high byte
SUBWF TMPH, W ;Sub 1st read
; with 2nd read
BTFSC STATUS, Z ;Is result = 0
GOTO CONTINUE ;Good 16-bit read

```

; TMR1L may have rolled over between the read  
; of the high and low bytes. Reading the high  
; and low bytes now will read a good value.

```

MOVWF TMR1H, W ;Read high byte
MOVWF TMPH ;
MOVWF TMR1L, W ;Read low byte
MOVWF TMPL ;
; Re-enable the Interrupt (if required)
CONTINUE ;Continue with
; your code

```

### 8.4 タイマ1オシレータ

適用デバイス							
710	71	711	72	73	73A	74	74A

クリスタルオシレータ回路はT10S1ピン(入力)とT10S0(アンプ出力)間に作られます。これは制御ビットT10SCEN (T1CON<3>) をセットすることによりイネーブルになります。オシレータは200kHzまでの低出力オシレータです。SLEEPの間動作を継続します。本来は32kHzクリスタル用に意図されたものです。表8-1にタイマ1オシレータに対するキャパシタ選択を示します。

タイマ1オシレータはLPオシレータと同じです。ソフトウェアによるタイム遅延で正しいオシレータの始動を確実にする必要があります。

表8-1: タイマ1オシレータ用容量の選択

Osc Type	Freq	C1	C2
LP	32 kHz <sup>(1)</sup>	15 pF	15 pF
	100 kHz	15 pF	15 pF
	200 kHz	0 - 15 pF	0 - 15 pF
キャパシタが高ければオシレータの安定性は増しますが、始動時間も増加します。この表の数値は設計のガイダンスを示すだけのものです。			
注 1: $V_{DD} > 4.5V$ では、 $C1 = C2 = 30pF$ が推奨値。			
Crystals Tested			
32.768 kHz	Epson C-001R32.768K-A	± 20 PPM	
100 kHz	Epson C-2 100.00 KC-P	± 20 PPM	
200 kHz	STD XTL 200.000 kHz	± 20 PPM	

# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。

## PIC16C7X

http://www.microchip.comより最新ERRATAシートを入手しご参照下さい。

### 8.5 CCPトリガ出力を使ったタイマ1のリセット

適用デバイス								
710	71	711	72	73	73A	74	74A	

CCP2 モジュールは、PIC16C72 デバイスにはインプリメントされていません。

CCP1 または CCP2 モジュールが“スペシャル・イベントトリガ” (CCP1M3:CCP1M0=1011) を発生するためにコンペアモードに設定されている場合、この信号がタイマ1をリセットします。

注意： CCP1 と CCP2 モジュールからのスペシャルイベントトリガは、割込みフラグビット TMR1IF (PIR1<0>) をセットしません。

この特徴を生かすためには、タイマ1がタイマまたは同期カウンタモードのいずれかに設定される必要があります。タイマ1が非同期カウンタモードで動作している場合には、このリセット操作は実行されないことがあります。

タイマ1へのライトが CCP1 または CCP2 からのスペシャルイベントトリガと一致する場合には、ライトが優先されます。

この動作モードでは、CCPRxH:CCPRxL のレジスタペアがタイマ1の周期レジスタとなります。

### 8.6 タイマ1レジスタのリセット(TMR1H、TMR1L)

適用デバイス								
710	71	711	72	73	73A	74	74A	

TMR1H と TMR1L レジスタは CCP1 スペシャルイベントトリガによる以外はPORまたは他のリセットでリセットされません。

T1CONレジスタはパワーオンリセットまたはブラウンアウトリセットで 00h にリセットされます。ほかのリセットではこのレジスタは影響されません。

### 8.7 タイマ1のプリスケアラ

適用デバイス								
710	71	711	72	73	73A	74	74A	

プリスケアラカウンタは TMR1H または TMR1L のレジスタへのライトでクリアされます。

表8-2: タイマ/カウンタとしてタイマ1に関連するレジスタ

アドレス	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR BOR での値	その他のリセットでの値
0Bh/8Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSP1F <sup>(1,2)</sup>	ADIF	RC1F <sup>(2)</sup>	TX1F <sup>(2)</sup>	SSP1F	CCP1F	TMR21F	TMR11F	0000 0000	0000 0000
8Ch	PIE1	PSP1E <sup>(1,2)</sup>	ADIE	RC1E <sup>(2)</sup>	TX1E <sup>(2)</sup>	SSP1E	CCP1E	TMR21E	TMR11E	0000 0000	0000 0000
0Eh	TMR1L	16ビットのTMR1レジスタの下位バイト用ホールディングレジスタ								xxxx xxxx	uuuu uuuu
0Fh	TMR1H	16ビットのTMR1レジスタの上位バイト用ホールディングレジスタ								xxxx xxxx	uuuu uuuu
10h	T1CON	-	-	T1CKPS1	T1CKPS0	T10SCEN	T1SYNC	TMR1CS	TMR10N	--00 0000	--uu uuuu

凡例： x = 未知、u = 不変、- = 未使用、'0'としてリード。網掛け部分はタイマ1モジュールでは使われません。

注 1： PSP1EとPSP1FビットはPIC16C73/73Aではリザーブされており、常にこれらのビットはクリアされた状態です。

2： PIC16C72 にはパラレルスレーブポートまたは USART がありません。これらのビットは未使用で、'0'としてリードします。

### 9.0 タイマ2 モジュール

適用デバイス							
710	71	711	72	73	73A	74	74A

タイマ2はプリスケアラとポストスケアラを持つ8ビットのタイマです。CCPモジュールのPWMモード用PWMタイムベースとして使われます。TMR2レジスタはリードとライトができ、どのデバイスのリセットでもクリアされます。

入力クロック(Fosc/4)には、(制御ビットT2CKPS1:T2CKPS0(T2CON<1:0>)により選択された)1:1、1:4、1:16のプリスケールオプションがあります。

タイマ2モジュールには8ビットの周期レジスタPR2があります。タイマ2は、00hからPR2に一致するまでインクリメントし、次のインクリメントの周期上で00hにリセットします。PR2はリードとライトができるレジスタです。PR2レジスタはRESETの間にセットされます。

TMR2の一致出力は、TMR2 割込み(フラグビットTMR2IF、PIR1<1>にラッチされた)を発生するために、4ビットのポストスケアラ(両端を含む1:1から1:16のスケールを与える)を通して行われます。

消費電力を最小限にするために、制御ビットTMR2ON(T2CON<2>)をクリアすることによりタイマ2をオフできます。

図9-2にタイマ2制御レジスタを示します。

### 9.1 タイマ2 のプリスケアラとポストスケアラ

適用デバイス							
710	71	711	72	73	73A	74	74A

プリスケアラとポストスケアラのカウンタは下記のいずれかが起こるとクリアされます。

- TMR2レジスタへのライト
- T2CONレジスタへのライト
- いずれ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)

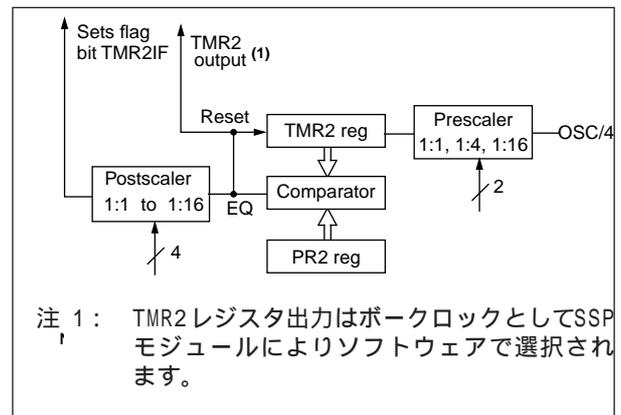
WDT、POR、MCLRリセットに対してだけ、T2CONが書き込まれている時にTMR2はクリアされません。

### 9.2 TMR2の出力

適用デバイス							
710	71	711	72	73	73A	74	74A

TMR2の出力(ポストスケアラの前)は同期シリアルポートモジュールへ送られ、シフトクロックを発生させるためにその出力をオプションとして使います。

図9-1 : タイマ2のブロック図



# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。  
<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

図9-2 : T2CON : タイマ2の制御レジスタ(アドレス 12h )

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
-	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit7							bit0

R = 読み込み可能なビット  
W = 書き込み可能なビット  
U = 未使用のビット、  
‘0’としてリード  
- n = PORリセットでの値

bit 7: **未使用** : ‘0’としてリード

bit 6-3: **TOUTPS3:TOUTPS0**: タイマ2出力ポストスケール選択ビット  
0000 = 1:1 ポストスケール  
0001 = 1:2 ポストスケール  
.  
.  
.  
1111 = 1:16 ポストスケール

bit 2: **TMR2ON**: タイマ2オンビット  
1 = タイマ2はオン  
0 = タイマ2はオフ

bit 1-0: **T2CKPS1:T2CKPS0**: タイマ2クロックプリスケール選択ビット  
00 = プリスケールは1  
01 = プリスケールは4  
1x = プリスケールは16

表9-1 : タイマ/カウンタとしてタイマ2に関連するレジスタ

アドレス	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR BOR での値	その他の リセット での値
0Bh/8Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF <sup>(1,2)</sup>	ADIF	RCIF(2)	TXIF(2)	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE <sup>(1,2)</sup>	ADIE	RCIE(2)	TXIE(2)	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
11h	TMR2	タイマ2モジュールのレジスタ								0000 0000	0000 0000
12h	T2CON	-	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
92h	PR2	タイマ2周期レジスタ								1111 1111	1111 1111

凡例 : x=未知、u=不変、-=未使用、‘0’としてリード。網掛け部分はタイマ2モジュールでは使われません。

注 1: PSPIEとPSPIF ビットは PIC16C73/73A ではリザーブされており、常にこれらのビットはクリアされた状態です。

注 2: PIC16C72にはパラレルスレーブポートまたはUSARTがありません。

これらのビットは未使用で、‘0’としてリードします。

### 10.0 キャプチャ/コンペア/PWMモジュール

適用デバイス								
710	71	711	72	73	73A	74	74A	CCP1
								CCP1
710	71	711	72	73	73A	74	74A	CCP2
								CCP2

各CCP(キャプチャ/コンペア/PWM)モジュールには、16ビットのレジスタが含まれており、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、PWM出力として動作することができます。CCP1とCCP2のモジュールは、スペシャルイベントトリガの操作を除いて、全く同じです。表10-1と10-2に、CCPモジュールのリソースと相互作用を示します。次の章では、CCP1に関するCCPモジュールの操作を説明します。CCP2の操作は、そこに記した以外はCCP1と同じです。

#### CCP1モジュール:

キャプチャ/コンペア/PWMレジスタ1(CCPR1)は2個の8ビットレジスタ: CCPR1L(下位バイト)とCCPR1H(上位バイト)で構成されています。両方ともリードとライトができます。

#### CCP2モジュール:

キャプチャ/コンペア/PWMレジスタ2(CCPR2)は2個の8ビットレジスタ: CCPR2L(下位バイト)とCCPR2H(上位バイト)で構成されています。両方ともリードとライトができます。

CCPモジュールを使うためには、コントロールハンドブック“CCPモジュールの使い方”(AN594)を参照してください。

表10-1: CCPモード-タイマのリソース

CCP Mode	Timer Resource
Capture	Timer1
Compare	Timer1
PWM	Timer2

表10-2: 2個のCCPモジュールの相互作用

CCPx Mode	CCPy Mode	相互作用
キャプチャ	キャプチャ	TMR1と同じタイムベース。
キャプチャ	コンペア	コンペアはスペシャルイベントトリガに対して設定され、TMR1をクリアする。
コンペア	コンペア	コンペアはスペシャルイベントトリガに対して設定され、TMR1をクリアする。
PWM	PWM	PWMは同じ周期で、デューティをアップデートする。(TMR2の割込み)
PWM	キャプチャ	なし
PWM	コンペア	なし

# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。  
<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

図10-1: CCP1CONレジスタ(アドレス17h)/CCP2CONレジスタ(アドレス1Dh)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
-	-	CCPxX	CCPxY	CCPxM3	CCPxM2	CCPxM1	CCPxM0	
bit7								bit0
bit 7-6: <b>未使用</b> : '0' としてリード								
bit 5-4: <b>CCPxX:CCPxY</b> : PWM 高解像度、下位選択ビット キャプチャモード: 使用せず コンペアモード: 使用せず PWMモード: 高解像度(10ビット)モードで2種類の下位ビットをライト。 8ビット解像度(標準解像度モード)が必要な場合、定数('0')で保たれます。								
bit 3-0: <b>CCPxM3:CCPxM0</b> : CCPx モード選択ビット 0000 = キャプチャ/コンペア/PWMはオフ(CCPxモジュールをリセット) 0100 = キャプチャモード、立ち下がりエッジ毎 0101 = キャプチャモード、立ち上がりエッジ毎 0110 = キャプチャモード、4回毎の立ち上がりエッジ 0111 = キャプチャモード、16回毎の立ち上がりエッジ 1000 = コンペアモード、一致で出力をセット(CCPxIFビットをセット) 1001 = コンペアモード、一致で出力をクリア(CCPxIFビットをセット) 1010 = コンペアモード、一致でソフトウェア割込みを発生(CCPxIFビットをセット、CCPxピンには影響なし) 1011 = コンペアモード、トリガスペシャルイベント(CCPxIFビットをセット、CCP1がTMR1をリセット; CCP2がTMR1をリセットしA/D変換を始める(A/Dモジュールがイネーブルの場合)) 11xx = PWMモード								

R = 読み込み可能なビット  
W = 書き込み可能なビット  
U = 未使用のビット、  
'0' としてリード  
- n = PORリセットでの値

### 10.1 キャプチャモード

適用デバイス							
710	71	711	72	73	73A	74	74A

キャプチャモードでは、ピンRC2/CCP1にイベントが起きた時、CCPR1H:CCPR1LがTMR1レジスタの16ビット値を取り込みます。そのイベントは次のように定義されます。

- ・ 立ち下がりエッジ
- ・ 立ち上がりエッジ
- ・ 4回毎の立ち上がりエッジ
- ・ 16回毎の立ち上がりエッジ

1つのイベントが制御ビットCCP1M3:CCP1M0(CCP1CON<3:0>)によって選択されます。キャプチャが行われた時、割込み要求フラグビットCCP1IF(PIR1<2>)がセットされます。これはソフトウェアでリセットしなければなりません。レジスタCCPR1の値が読み込まれる前に他のキャプチャが起こった場合は、取り込まれていた古い値は失われます。キャプチャモードでは、RC2/CCP1ピンをそれに対応するTRISビットをセットすることにより入力として設定する必要があります。

**注意:** RC2/CCP1が出力として設定されている場合は、そのポートへのライトによってキャプチャ条件とすることができます。

このキャプチャモードを変えた時、偽キャプチャ割込みが発生することがあります。偽割込みを避けるために、ビットCCP1IE(PIE1<2>)をクリアにしたままにし、動作中のモードでそのような変更の後にはフラグビットCCP1IFをクリアする必要があります。

### 10.1.1 プリスケーラ

ビットCCP1M3:CCP1M0により設定される4種類のプリスケーラのセッティングがあります。CCPモジュールがオフされるかキャプチャモードでない時は、プリスケーラカウンタが常にクリアされています。つまり、どのリセットもプリスケーラカウンタをクリアします。

キャプチャプリスケーラが他の状態に切り替わると、割込みが発生します。さらにそのプリスケーラカウンタがクリアされず、それによって最初のキャプチャがノンゼロプリスケーラからであることがあります。例10-1にキャプチャプリスケーラ間の切り替えの推奨方法を示します。この例では、プリスケーラカウンタをクリアしますが、“偽”割込みは発生しません。

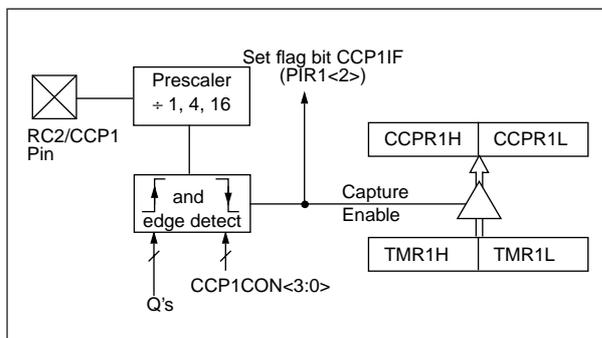
#### 例10-1: キャプチャプリスケーラ間の切り替え

```
CLRFB CCP1CON      ;Turn CCP module off
MOVLW NEW_CAPT_PS  ;Load the W reg with
                   ; the new prescaler
                   ; mode value and CCP ON
MOVWF CCP1CON      ;Load CCP1CON with this
                   ; value
```

### 10.1.2 キャプチャモードの選択

CCPモジュールでキャプチャ機能を使うためには、タイマ1が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キャプチャ動作は働きません。

#### 図10-2: キャプチャモード動作のブロック図



## 10.2 コンペアモード

適用デバイス							
710	71	711	72	73	73A	74	74A

コンペアモードでは、16ビットのCCPR1レジスタの値がTMR1レジスタペアの値と継続的に比較されています。一致が起こった時、RC2/CCP1ピンは下記のいずれかになります。

- ・ハイになる
- ・ローになる
- ・変化せず

そのピンの動作は制御ビットCCP1M3:CCP1M0 (CCP1CON<3:0>)の値を元にしています。同時にコンペア割込みも発生します。TRISC<2>ビットをクリアすることによりRC2/CCP1ピンを出力として設定する必要があります。

注意: CCP1CONレジスタをクリアすると、RC2/CCP1コンペア出力ラッチが初期状態のロー・レベルになりますが、これはデータラッチではありません。

### 10.2.1 コンペアモードの選択

CCPモジュールでコンペア機能を使う時には、タイマ1が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動作は働きません。

### 10.2.2 ソフトウェア割込みモード

もう1つのコンペアモードにはソフトウェア割込みモードがあり、その時はCCP1ピンは影響されません。CCP割込みだけが発生します(イネーブルの場合)。

### 10.2.3 スペシャルイベントトリガ

このモードでは、内部ハードウェアトリガが発生し、動作を始めるために使われます。

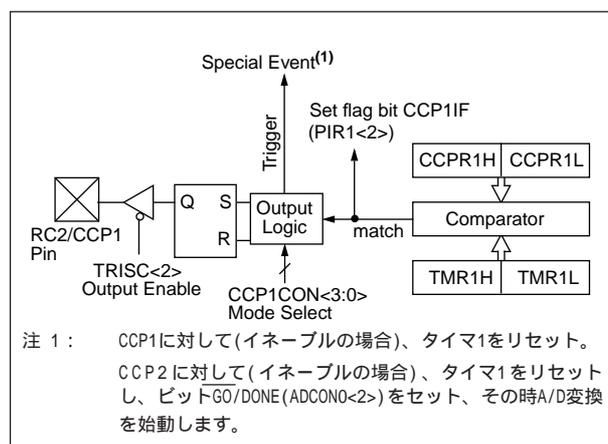
CCP1のスペシャルイベントトリガ出力は、TMR1レジスタペアをリセットします。これによって、CCPR1レジスタはタイマ1に対する16ビットのプログラム可能な周期レジスタとなることができます。

CCP2のスペシャルトリガ出力は、TMR1レジスタペアをリセットし、A/D変換を始めます(A/Dモジュールがイネーブルの場合)。

PIC16C72だけは、CCP1のスペシャルイベントトリガ出力は、TMR1レジスタをリセットし、A/D変換を始めます(A/Dモジュールがイネーブルの場合)。

注意: CCP1とCCP2モジュールからのスペシャルイベントトリガは、割込みフラグビットTMR1IF(PIR1<0>)をセットしません。

#### 図10-3: コンペアモードの動作ブロック図



注 1: CCP1に対して(イネーブルの場合)、タイマ1をリセット。CCP2に対して(イネーブルの場合)、タイマ1をリセットし、ビットGO/DONE(ADCON0<2>)をセット、その時A/D変換を開始します。

## PIC16C7X

### 10.3 PWMモード

適用デバイス							
710	71	711	72	73	73A	74	74A

パルス幅変調モード(PWM)では、ピンRC2/CCP1は10ビットまでの分解能のPWM出力を生成します。このピンはTRISC<2>ビットをクリアすることにより出力として設定されなければなりません。そのピンはデータラッチでマルチプレクスされています。PWMモードでは、8ビットのデューティサイクル値をCCPR1レジスタの下位バイトであるCCPR1Lに書き込みます。上位バイトのCCPR1Hは下位バイトに対するスレーブバッファとして使われます。PWM1出力がセットされている時(つまりデューティサイクルの始めで)、8ビットのデータがマスタからスレーブに転送されます。このダブルバッファはグリッチをなくすPWM出力のために重要です。PWMモードでは、CCPR1Hは読み込みますが、書き込みはできません。PWMの周期はタイマ2周期レジスタ(PR2)により決定されます。

PWM周期=

$$[(PR2)+1] \cdot 4T_{osc} \cdot (TMR2 \text{ プリスケール値})$$

PWMデューティ・サイクル=

$$(DC1) \cdot T_{osc} \cdot (TMR2 \text{ プリスケール値})$$

ただしDC1=CCPRxLとCCPxCON<5:4>が結合した10ビットの値。したがって、PWM出力の解像度は最高10ビットまでプログラムできます。

注意: CCP1CONレジスタをクリアすると、RC2/CCP1 PWM出力ラッチが初期状態のロー・レベルになります。これはI/Oデータラッチではありません。タイマ2のポストスケラはPWM周波数の決定では使われません。ポストスケラはPWM出力よりも異なる周波数のサーボアップデートレートを持つために使われます。

図10-4 PWMのブロック図

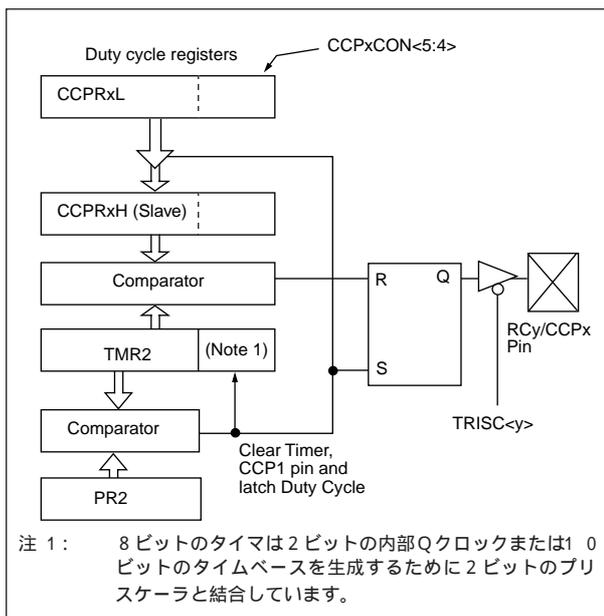


表10-3: 20MHzでのPWM周波数 VS 分解能

最大分解能 (High Resolution Mode)	周波数		
	TMR2 Prescale=1	TMR2 Prescale=4	TMR2 Prescale=16
10-bit	19.53 kHz	4.88 kHz	1.22 kHz
9-bit	39.06 kHz	9.77 kHz	2.44 kHz
8-bit	78.13 kHz	19.53 kHz	4.88 kHz

表10-4: 20MHzでのPWM周波数と分解能の実例

PWM 周波数	1.22 kHz	4.88 kHz	19.53 kHz	78.12 kHz	156.3 kHz	208.3 kHz
Timer Prescaler (16, 4, 1)	16	4	1	1	1	1
PR2 Value	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
分解能 (High-resolution mode)	10-bit	10-bit	10-bit	8-bit	7-bit	5.5-bit
分解能 (Standard-resolution mode)(1)	8-bit	8-bit	8-bit	6-bit	5-bit	3.5-bit

注 1: 標準的な分解能モードには、CCPxX:CCPyYビットの定数(または'0')があり、PR2レジスタ値に対してTMR2レジスタ値を比較するだけです。Qサイクルは使用されません。

# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。  
<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

表10-5: キャプチャとタイマ1に関連するレジスタ

アドレス	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR BOR での値	その他の リセット での値
0Bh/8Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF <sup>(1,2)</sup>	ADIF	RCIF(2)	TXIF(2)	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0Dh(2)	PIR2	-	-	-	-	-	-	-	CCP2IF	---- --0	---- --0
8Ch	PIE1	PSPIE <sup>(1,2)</sup>	ADIE	RCIE(2)	TXIE(2)	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
8Dh(2)	PIE2	-	-	-	-	-	-	-	CCP2IE	---- --0	---- --0
0Eh	TMR1L	16ビットTMR 1 レジスタの最下位バイト用にレジスタを保持。								xxxx xxxx	uuuu uuuu
0Fh	TMR1H	16ビットTMR 1 レジスタの最上位バイト用にレジスタを保持。								xxxx xxxx	uuuu uuuu
10h	T1CON	-	-	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	--00 0000	--uu uuuu
15h	CCPR1L	キャプチャ/コンペア/PWMレジスタ1 (LSB)								xxxx xxxx	uuuu uuuu
16h	CCPR1H	キャプチャ/コンペア/PWMレジスタ1 (MSB)								xxxx xxxx	uuuu uuuu
17h	CCP1CON	-	-	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
1Bh(2)	CCPR2L	キャプチャ/コンペア/PWMレジスタ2 (LSB)								xxxx xxxx	uuuu uuuu
1Ch(2)	CCPR2H	キャプチャ/コンペア/PWMレジスタ2 (MSB)								xxxx xxxx	uuuu uuuu
1Dh(2)	CCP2CON	-	-	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	--00 0000

凡例: x=未知、u=不変、-=未使用、'0'としてリード。網掛け部分はキャプチャとタイマ1では使用しない。

注 1: PSPIEとPSPIFビットはPIC16C73/73Aではリザーブされ、常にこれらのビットはクリアされた状態です。

2: PIC16C72にはパラレル・スレーブ・ポート、USART、CCP2モジュールがありません。これらのビットは未使用で、'0'としてリードします。

表10-6: コンペアとタイマ1に関連するレジスタ

アドレス	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR BOR での値	その他の リセット での値
0Bh/8Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF <sup>(1,2)</sup>	ADIF	RCIF(2)	TXIF(2)	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0Dh(2)	PIR2	-	-	-	-	-	-	-	CCP2IF	---- --0	---- --0
8Ch	PIE1	PSPIE <sup>(1,2)</sup>	ADIE	RCIE(2)	TXIE(2)	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
8Dh(2)	PIE2	-	-	-	-	-	-	-	CCP2IE	---- --0	---- --0
0Eh	TMR1L	16ビットTMR 1 レジスタの最下位バイト用にレジスタを保持。								xxxx xxxx	uuuu uuuu
0Fh	TMR1H	16ビットTMR 1 レジスタの最上位バイト用にレジスタを保持。								xxxx xxxx	uuuu uuuu
10h	T1CON	-	-	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	--00 0000	--uu uuuu
15h	CCPR1L	キャプチャ/コンペア/PWMレジスタ1 (LSB)								xxxx xxxx	uuuu uuuu
16h	CCPR1H	キャプチャ/コンペア/PWMレジスタ1 (MSB)								xxxx xxxx	uuuu uuuu
17h	CCP1CON	-	-	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
1Bh(2)	CCPR2L	キャプチャ/コンペア/PWMレジスタ2 (LSB)								xxxx xxxx	uuuu uuuu
1Ch(2)	CCPR2H	キャプチャ/コンペア/PWMレジスタ2 (MSB)								xxxx xxxx	uuuu uuuu
1Dh(2)	CCP2CON	-	-	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	--00 0000

凡例: x=未知、u=不変、-=未使用、'0'としてリード。網掛け部分はコンペアとタイマ1では使われません。

注1: PSPIEとPSPIFビットはPIC16C73/73Aではリザーブされており、常にこれらのビットはクリアされた状態です。

2: PIC16C72にはパラレル・スレーブ・ポート、USART、CCP2モジュールがありません。これらのビットは未使用で、'0'としてリードします。

# 暫定版

英語最新版データシートと併用にご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。

## PIC16C7X

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

表10-7: PWMとタイマ2に関連するレジスタ

アドレス	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR BOR での値	その他の リセット での値
0Bh/8Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF <sup>(1,2)</sup>	ADIF	RCIF(2)	TXIF(2)	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0Dh(2)	PIR2	-	-	-	-	-	-	-	CCP2IF	---- --0	---- --0
8Ch	PIE1	PSPIE <sup>(1,2)</sup>	ADIE	RCIE(2)	TXIE(2)	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
8Dh(2)	PIE2	-	-	-	-	-	-	-	CCP2IE	---- --0	---- --0
11h	TMR2	タイマ2モジュールのレジスタ								0000 0000	0000 0000
92h	PR2	タイマ2モジュールの周期レジスタ								1111 1111	1111 1111
12h	T2CON	-	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
15h	CCPR1L	キャプチャ/コンペア/PWMレジスタ1 (LSB)								xxxx xxxx	uuuu uuuu
16h	CCPR1H	キャプチャ/コンペア/PWMレジスタ1 (MSB)								xxxx xxxx	uuuu uuuu
17h	CCP1CON	-	-	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
1Bh(2)	CCPR2L	キャプチャ/コンペア/PWMレジスタ2 (LSB)								xxxx xxxx	uuuu uuuu
1Ch(2)	CCPR2H	キャプチャ/コンペア/PWMレジスタ2 (MSB)								xxxx xxxx	uuuu uuuu
1Dh(2)	CCP2CON	-	-	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	--00 0000

凡例: x=未知、u=不変、-=未使用、'0'としてリード。網掛け部分はPWMとタイマ2では使われません。

注 1: PSPIEとPSPIFビットはPIC16C73/73Aではリザーブされており、常にこれらのビットはクリアされた状態です。

注 2: PIC16C72にはパラレルスレーブポート、USART、CCP2モジュールがありません。これらのビットは未使用で、'0'としてリードします。

### 11.0 同期シリアルポート(SSP)モジュール

同期シリアルポート(SSP)モジュールは他の周辺回路またはマイクロコントローラデバイスと通信するのに便利なシリアルインターフェイスです。これらの周辺回路デバイスはシリアルEEPROM、シフトレジスタ、ディスプレイドライバ、A/Dコンバータなどがあります。SSPモジュールは次の2種類のモードのどちらか1つで動作することができます。

- ・シリアル周辺インターフェイス(SPI)
- ・集積回路間(I<sup>2</sup>C)

アプリケーション・ノートAN578 “ I<sup>2</sup>Cマルチ-マスター環境でのSSPモジュールの使い方 ” を参照してください。

図11-1 : SSPSTAT : 同期シリアルポートステータスレジスタ(アドレス94h)

	U-0	U-0	R-0	R-0	R-0	R-0	R-0	R-0	
			D/A	P	S	R/W	UA	BF	
	bit7								bit0
bit 7-6:	<b>未使用</b> : ‘0’ としてリード。								
bit 5:	<b>D/A</b> : データ/アドレスビット (I <sup>2</sup> Cモードのみ) 1=受信または送信した最後のバイトがデータであることを示す。 0=受信または送信した最後のバイトがアドレスであることを示す。								
bit 4:	<b>P</b> : ストップビット (I <sup>2</sup> Cモードのみ。このビットはSSPモジュールがディセーブルになり、SSPENがクリアされた時クリアされる)。 1=ストップビットが最後に検出されたことを示す(このビットはRESET上は‘0’)。 0=ストップビットが検出されなかったことを示す。								
bit 3:	<b>S</b> : スタートビット (I <sup>2</sup> Cモードのみ。このビットはSSPモジュールがディセーブルになり、SSPENがクリアされた時クリアされる)。 1=スタートビットが最後に検出されたことを示す(このビットはRESET上は‘0’)。 0=スタートビットが検出されなかったことを示す。								
bit 2:	<b>R/W</b> : リード/ライトビットの情報 (I <sup>2</sup> Cモードのみ)。 このビットは最終アドレスマッチに続くR/Wビット情報を保持。このビットは送信中のみ有効。 1=リード 0=ライト								
bit 1:	<b>UA</b> : アップデートアドレス (10ビットのI <sup>2</sup> Cモードのみ)。 1=SSPADDレジスタのアドレスを更新する必要があることを示す。 0=アドレス更新の必要なし。								
bit 0:	<b>BF</b> : バッファフルステータスビット <u>受信</u> (SPIとI <sup>2</sup> Cモード) 1=受信完了。SSPBUFはフル。 0=受信中。SSPBUFは空。 <u>送信</u> (I <sup>2</sup> Cモードのみ) 1=送信中。SSPBUFはフル。 0=送信完了。SSPBUFは空。								

R = 読み込み可能なビット  
 W = 書き込み可能なビット  
 U = 未使用のビット、  
 ‘0’ としてリード  
 -n = POR リセットでの値

# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。  
<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

図11-2 : SSPCON : 同期シリアルポート制御レジスタ(アドレス14h)

| R/W-0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| WCOL  | SSPOV | SSPEN | CKP   | SSPM3 | SSPM2 | SSPM1 | SSPM0 |
| bit7  |       |       |       |       |       |       | bit0  |

R = 読み込み可能なビット  
W = 書き込み可能なビット  
U = 未使用のビット、  
'0'としてリード  
-n = POR リセットでの値

bit7: **WCOL**: ライト衝突検出ビット  
1=SSPBUFレジスタは、前のワードをまだ送信中でも書き込まれます(ソフトウェアでクリアが必要)。  
0=衝突なし。

bit6: **SSPOV**: 受信オーバーフロー指示ビット  
SPIモードで  
1=SSPBUFレジスタがまだ前のデータを保持している時に新しい1バイトを受信。オーバーフローの場合にはSSPSRのデータは失われます。オーバーフローはスレープモードでのみ起こります。オーバーフローを避けるためには、データ送信を行っている時でもSSPBUFを読み込む必要があります。マスタモードでは、それぞれの新しい受信(および送信)はSSPBUFレジスタへのライトによって開始されるので、オーバーフロービットはセットされません。  
0=オーバーフローなし。  
I<sup>2</sup>Cモードで  
1=SSPBUFレジスタがまだ前のバイトを保持している時に新しい1バイトを受信。送信モードではSSPOVは無視できます。SSPOVはどちらかのモードのソフトウェアでクリアが必要です。  
0=オーバーフローなし。

bit5: **SSPEN**: 同期シリアルポートイネーブルビット  
SPIモードで  
1=シリアルポートをイネーブルにし、SDK、SDO、SDIをシリアルポートピンとして構成。  
0=シリアルポートをディセーブルにし、上記のピンをI/Oポートピンとして構成。  
I<sup>2</sup>Cモードで  
1=シリアルポートをイネーブルにし、SDA、SCLピンをシリアルポートピンとして構成。  
0=シリアルポートをディセーブルにし、上記のピンをI/Oポートピンとして構成。  
両モードにおいて、イネーブルの時はこれらのピンは入力または出力として適切に構成する必要があります。

bit4: **CKP**: クロック極性選択ビット  
SPIモードで  
1=立ち下がりエッジで送信開始、立ち上がりエッジで受信。クロックのアイドル状態はハイレベル。  
0=立ち上がりエッジで送信開始、立ち下がりエッジで受信。クロックのアイドル状態はローレベル。  
I<sup>2</sup>Cモードで  
SCKリリース制御  
1=イネーブルクロック  
0=クロックをローに保持(クロックストレッチ)(データのセットアップ時間確保のために使用されます)。

bit3-0: **SSPM3:SSPM0**: 同期シリアルポートモード選択ビット  
0000=SPIマスタモード、クロック=Fosc/4  
0001=SPIマスタモード、クロック=Fosc/16  
0010=SPIマスタモード、クロック=Fosc/64  
0011=SPIマスタモード、クロック=TMR2出力/2  
0100=SPIスレープモード、クロック=SCKピン。SSピン制御はイネーブル。  
0101=SPIスレープモード、クロック=SCKピン。SSピン制御はディセーブル。SSはI/Oピンとして使用可能。  
0110=I<sup>2</sup>Cスレープモード、7ビットアドレス。  
0111=I<sup>2</sup>Cスレープモード、10ビットアドレス。  
1011=I<sup>2</sup>Cスタートとストップビットの割込みがイネーブル(スレープアイドル)。  
1110=I<sup>2</sup>Cスレープモード、スタートとストップビットの割込みがイネーブルでの7ビットアドレス。  
1111=I<sup>2</sup>Cスレープモード、スタートとストップビットの割込みがイネーブルでの10ビットアドレス。

## 11.1 SPIモード

適用デバイス							
710	71	711	72	73	73A	74	74A

SPIモードにより、8ビットのデータを同期させて送信と受信を同時にすることができます。通信を行うためには、一般的に次の3本のピンが使われます。

- ・シリアルデータ出力(SDO)RC5/SD0
- ・シリアルデータ入力(SDI)RC4/SDI
- ・シリアルクロック(SCK)RC3/SCK

スレープモードの動作中は、さらに4番目のピンが使われることもあります。

- ・スレープ選択(SS)RA5/AN4/SS

SPIの初期化時には、いくつかのオプションを設定する必要があります。それはSSPCONレジスタ(SSPCON<5:0>)の制御ビットを適切にプログラムすることにより行われます。これらの制御ビットにより、次のような設定が可能です。

- ・マスタモード(SCKはクロック出力)
- ・スレープモード(SCKはクロック入力)
- ・クロックの極性(SCKの立ち上がり/立ち下がりエッジ時の出力/入力のデータ)
- ・クロックレート(マスタモードのみ)
- ・スレープ選択モード(スレープモードのみ)

SSPIは送信/受信シフトレジスタ(SSPSR)とバッファレジスタ(SSPBUF)で構成されています。SSPSRはデバイスの入力と出力のデータをシフトします。MSBを先にシフトします。SSPBUFは受信されたデータが用意できるまで、SSPSRに前もって書き込まれたデータを保持します。8ビットのデータが受信されてしまうと、その情報はSSPBUFレジスタに移されます。それからバッファフル検出ビットBF(SSPSTAT<0>)と割込みフラグビットSPIF(PIR1<3>)がセットされます。この受信されたデータ(SSPBUF)をダブルバッファすることにより、受信されたデータを読み出す前に次のバイトの受信が始まるのを可能にします。データの送信/受信中のSSPBUFレジスタへのライトは無視され、書き込み衝突検出ビットWCOL(SSPCON<7>)がセットされます。ソフトウェアでWCOLビットをクリアする必要があります。それによってSSPBUFレジスタへの次のライトをうまく完了するかどうかを決定できます。アプリケーションソフトウェアが有効なデータを受信しようとしている時、転送するデータの次のバイトをSSPBUFに書き込む前にそのSSPBUFを読み込む必要があります。SSPBUFが受信されたデータにロードされている時(転送完了)、バッファフルビットBF(SSPSTAT<0>)がそのことを示します。SSPBUFが読み込まれた時、ビットBFがクリアされます。SPIが送信のみの場合、このデータは不適切なことがあります。一般的には、送信/受信が完了した時を決めるのにSSPI割込みが使われます。さらにSSPBUFを読み込むか(データが有効な場合)、そして/または、そのSSPBUF(SSPSR)を書き込むことができます。その割込み方法を使わない場合は、ソフトウェアポーリングによって、ライト衝突が起こらないよう確実にすることができます。例11-1にデータ送信のためのSSPBUF(SSPSR)のロード方法を示します。網掛けの命令は受信されたデータが有効な場合にだけ必要です。

### 例11-1: SSPBUF(SSPSR)レジスタのロード

```

BSF    STATUS,RP0    ;Specify Bank 1
LOOP  BTFSS  SSPSTAT,BF ;Has data been
                                ;received
                                ;(transmit
                                ;complete)?

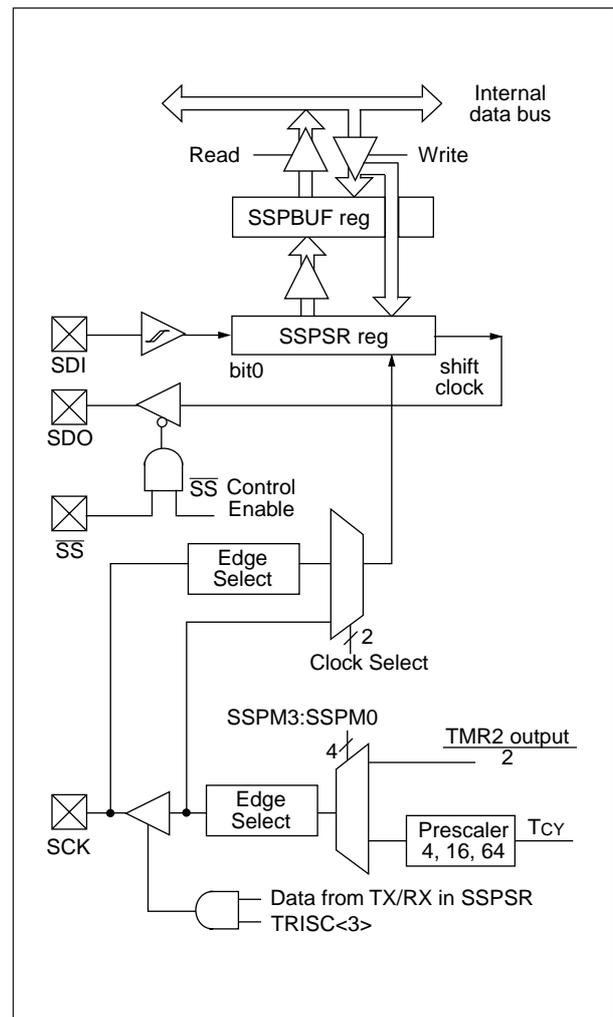
GOTO   LOOP          ;No
BCF    STATUS,RP0    ;Specify Bank 0
MOVF   SSPBUF,W      ;W reg = contents
                                ; of SSPBUF

MOVWF  RXDATA        ;Save in user RAM
MOVF   TXDATA,W      ;W reg = contents
                                ; of TXDATA

MOVWF  SSPBUF        ;New data to xmit
    
```

SPIモード時のSSPモジュールのブロック図(図11-3)は、SSPSRが直接リードまたはライトができず、SSPBUFレジスタをアドレス指定することによりアクセスできるのみであることを示します。さらに、SSPステータスレジスタ(SSPSTAT)が種々のステータス条件を示します。

### 図11-3: SSP ブロック図 (SPIモード)



シリアルポートをイネーブルにするためには、SSPイネーブルビットSSPEN(SSPCON<5>)をセットしなければなりません。SPIモードをリセットまたは再設定するためには、ビットSSPENをクリアし、SSPCONレジスタを再初期化し、さらにビットSSPENをセットします。これにより、SDI、SDO、SCK、SSピンがシリアルポートピンとして設定されます。シリアルポート機能として作用するためのピンとして、それぞれのデータ方向ビット(TRISCレジスタ内)を適切にプログラムする必要があります。

- SDIはTRISC<4>をセット
- SDOはTRISC<5>をクリア
- SCK(マスタモード)はTRISC<3>をクリア
- SCK(スレーブモード)はTRISC<3>をセット
- SSはTRISA<5>をセット

必要でないシリアルポート機能は、対応するデータ方向(TRIS)レジスタを反対の値にプログラムすることにより優先します。例えば、データを(ディスプレイドライバへ)送信しているだけのマスタモードでは、対応するTRISレジスタビットをクリアすることにより、SDIとSSの両方が汎用出力として使われます。

図11-4に2個のマイクロコントローラ間の一般的な接続を示します。マスタコントローラ(プロセッサ1)はSCK信号を送っ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で両方のシフトレジスタでシフトアウトされ、そのクロックの反対のエッジでラッチされます。両方のプロセッサは、同じクロック極性(CKP)にプログラムされる必要があり、そうすれば両方のコントローラが同時にデータの送信と受信を行います。そのデータが有効かどうか(ダミーデータかどうか)はアプリケーションソフトウェアによります。これによりデータ送信に関して、次の3つの場合が考えられます。

- マスタがデータを送る - スレーブがダミーデータを送る
- マスタがデータを送る - スレーブがデータを送る
- マスタがダミーデータを送る - スレーブがデータを送る

マスタがSCKを制御するので、いつでもデータ転送を開始できます。スレーブ(プロセッサ2)がデータを同報通信したい時は、マスタはソフトウェアのプロトコルによって決められます。

マスタモードでは、SSPBUFレジスタに書き込まれると同時にデータが送信/受信されます。SPIが受信だけを行う場合は、SCK出力をディセーブルすることができます(入力としてプログラムする)。SSPSRレジスタはプログラムされたクロックレートでSDIピンに与えられた信号のシフトインを継続します。各バイトが受信されると、通常の受信されたバイトのようにSSPBUFレジスタにロードされます(割込みとステータスビットがセット)。これは“ライン状態モニタ”モードとしての受信应用到に便利です。

スレーブモードでは、SCKに現れた外部クロックパルスとしてデータの送信と受信が行われます。最後のビットがラッチされると、割込みフラグビットSSPIF(PIR1<3>)がセットされます。

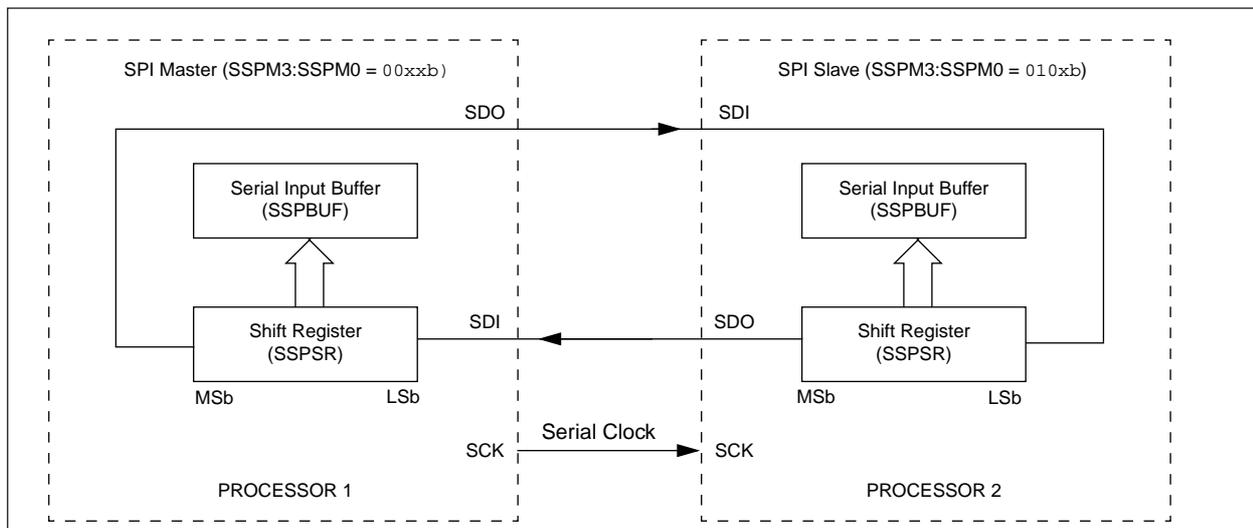
クロック極性はビットCKP(SSPCON<4>)を適切にプログラムすることにより選択されます。これにより、図11-5と11-6に示すように、SPI通信の波形が与えられ、ここではMSBが最初に送信されます。マスタ・モードでは、SPIクロックレート(ビットレート)は次のうちいずれか1つにプログラム可能です。

- $F_{osc}/4$ (または $T_{cy}$ )
- $F_{osc}/16$ (または $4 \cdot T_{cy}$ )
- $F_{osc}/64$ (または $16 \cdot T_{cy}$ )
- タイマ2出力/2

これにより、最大ビットクロック周波数(20MHzにて)5MHzが可能になります。スレーブモードでは、外部クロックは最小のハイ・タイムとロー・タイムに一致する必要があります。

スレーブモードでは、スレーブはデータを送信と受信でき、スレーブからデバイスをウェークできます。

図11-4: SPI マスタ/スレーブ接続



$\overline{SS}$ ピンによって同期スレーブ・モードが可能です。SPIはスレーブモード(SSPCON<3:0>=04h)で、同期スレーブモードがイネーブルになるようにTRISA<5>をセットする必要があります。 $\overline{SS}$ ピンがローの時、送信と受信がイネーブルとなり、SD0ピンが駆動します。 $\overline{SS}$ ピンがハイになると、SD0ピンが送信中のバイトの間であっても駆動せず、フローティング出力となります。アプリケーションによっては、外部のプルアップ/プルダウン抵抗が望ましい場合があります。

2ワイヤ通信をエミュレートするには、SD0ピンをSD1ピンに接続することができます。SPIが受信の動作を必要とする時はSD0ピンを入力として設定します。これにより、SD0からの送信がディセーブルされます。SD1はバス衝突を起こすことがないため、常に入力(SD1機能)のままにされます。

図11-5: SPI モードのタイミング (マスタモードまたはスレーブモード  $\overline{SS}$  制御なし)

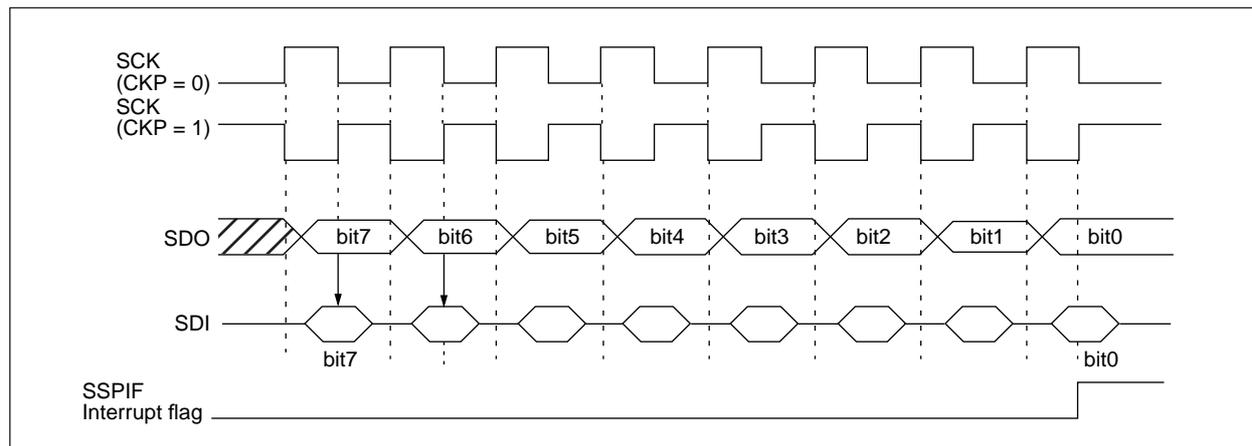
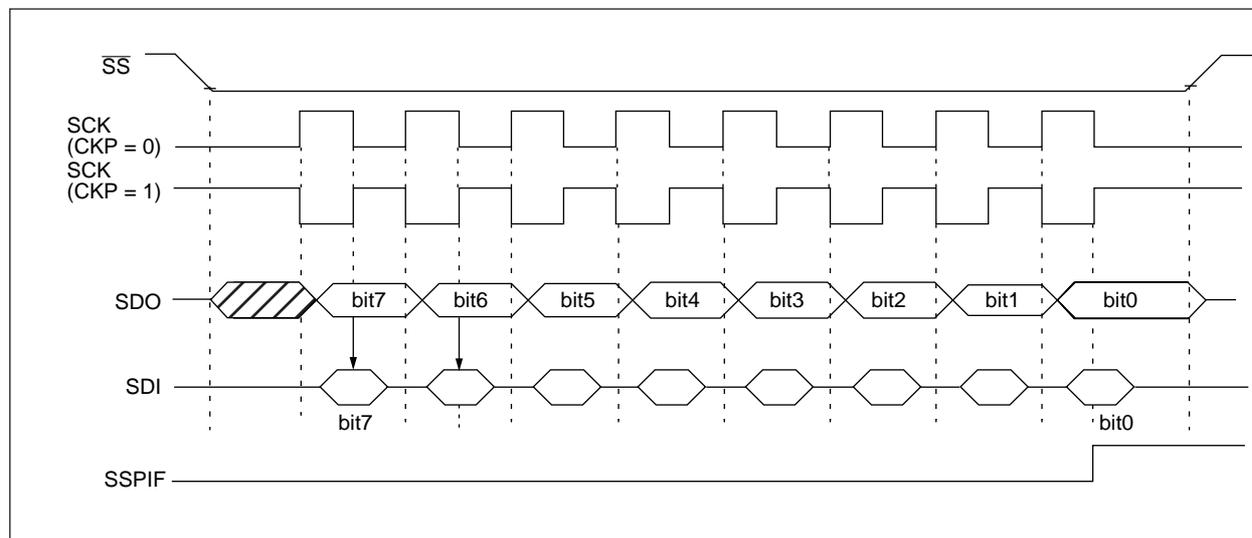


図11-6: SPIモードのタイミング ( $\overline{SS}$  制御によるスレーブモード)



# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。

## PIC16C7X

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

表11-1: SPI 動作に関連するレジスタ

アドレス	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR BOR での値	その他の リセット での値
0Bh/8Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF <sup>(1,2)</sup>	ADIF	RCIF(2)	TXIF(2)	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE <sup>(1,2)</sup>	ADIE	RCIE(2)	TXIE(2)	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
13h	SSPBUF	同期シリアルポート受信バッファ / 送信レジスタ								xxxx xxxx	uuuu uuuu
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
85h	TRISA	-	-	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111
94h	SSPSTAT	-	-	D/ $\bar{A}$	P	S	R/ $\bar{W}$	UA	BF	--00 0000	--00 0000

凡例: x=未知、u=不変、- =未使用、'0'としてリード。網掛け部分はSPIモードにおいてSSPでは使われません。

注 1: PSPIEとPSPIFビットはPIC16C73/73Aではリザーブされており、常にこれらのビットはクリアされた状態です。

2: PIC16C72にはパラレルスレーブポートまたはUSARTがありません。これらのビットは未使用で、'0'としてリードします。

### 11.2 I<sup>2</sup>C™の概要

適用デバイス							
710	71	711	72	73	73A	74	74A

この章では、集積回路間(I<sup>2</sup>C)バスの概要について述べ、11.3章でI<sup>2</sup>CモードでのSSPモジュールの動作について説明します。

I<sup>2</sup>Cバスは、Phillips Corporationにより開発された2ワイヤシリアルインターフェイスです。本来の規格、つまり標準モードは100Kbpsまでのデータ転送用でした。改訂規格、つまり高速モードでは400Kbpsまでのデータ転送をサポートしています。標準モードと高速モードの両方の製品は同じバスに取り付けられた場合、お互いに動作します。

I<sup>2</sup>Cインターフェイスにはデータの確実な送信と受信を行うための広範なプロトコルが含まれます。データを送信している時、一方のデバイスが“マスタ”で(クロックを発生する)、他方のデバイスが“スレーブ”として働きます。スレーブプロトコルのすべての機能はSSPモジュールのハードウェアにインプリメントされ、マスタ・プロトコルの機能はPIC16CXXのソフトウェアでアドレス指定される必要があります。表11-2にI<sup>2</sup>Cバス用語をいくつか定義しておきます。I<sup>2</sup>Cインターフェイス規格に関する他の情報についてはPhillips Corporationの資料“I<sup>2</sup>Cバスとその使い方”を参照してください。

I<sup>2</sup>Cインターフェイスプロトコルでは、各デバイスにアドレスがあります。マスタがデータ送信をしたい時には“talk”したいデバイスのアドレスを最初に送信します。すべてのデバイスがそれが自分のアドレスかどうかを知るために“listen”します。このアドレス内に、マスタがスレーブのデバイスに対してリード/ライトしたいかどうかをビットが示します。マスタとスレーブはデータ転送の間、常に反対のモード(送信/受信)で動作しています。つまりお互いに次の2つの関係のいずれかで動作していると考えることができます。

- ・ マスタ-トランスミッタとスレーブ-レシーバ
- ・ スレーブ-トランスミッタとマスタ-レシーバ

両方の場合、マスタがクロック信号を発生します。

クロック(SCL)とデータ(SDA)ラインの出力回路には、バスのワイヤードAND機能を持つためにオープン-ドレインまたはオープン-コレクタがなければいけません。ラインをプルダウンするデバイスがない時、ハイレベルを確実にするために外部のプルアップ抵抗を使います。I<sup>2</sup>Cバスに接続されるデバイスの数はバスの最大負荷規格400pFだけに制限されます。

#### 11.2.1 データ転送の開始と終了

データ転送が行われていない間(アイドルタイム)、クロックライン(SCL)とデータライン(SDA)の両方は外部のプルアップ抵抗でハイに引き上げられています。STARTとSTOPの条件はデータ送信の開始と終了を決定します。START条件はSCLがハイの時SDAのハイからローへの変化で定義されます。STOP条件はSCLがハイの時SDAのローからハイへの変化で定義されます。図11-7にSTARTとSTOP条件を示します。データ転送の開始と終了のために、マスタがこれらの条件を発生させます。STARTとSTOP条件の定義により、データが送信されている時、SCLラインがローの時SDAラインが状態を変化させることができます。

図11-7: STARTとSTOPの条件

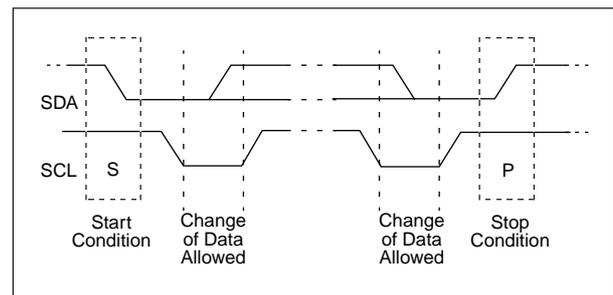


表11-2: I<sup>2</sup>Cバスの用語

用語	説明
トランスミッタ	データをバスに送信するデバイス。
レシーバ	データをバスから受信するデバイス。
マスタ	転送の開始、クロックの生成、転送の終了を行うデバイス。
スレーブ	マスタによってアドレスされるデバイス。
マルチ-マスタ	システム内の1基以上のマスタ・デバイス。これらのマスタは、メッセージに影響を与えないで同時にバスを制御することができる。
アービトレーション	マスタ・デバイスのうち1つだけがバスを制御するのを確実にする手順。これにより転送データへの影響が排除される。
同期	2つ以上のデバイスのクロック信号を同期させる手順。

## PIC16C7X

### 11.2.2 I<sup>2</sup>Cデバイスのアドレス指定

2種類のアドレス形式があります。最も簡単なのは、R/Wビットを持った7ビットのアドレス形式です(図11-8参照)。もっと複雑なのは、R/Wビットを持った10ビットのアドレス形式です(図11-9参照)。10ビットのアドレス形式では、10ビットのアドレス形式であることを示す最初の5ビットを持った2バイトを送信する必要があります。

図11-8: 7ビットのアドレスフォーマット

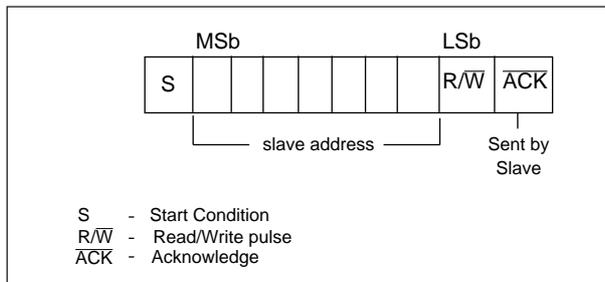
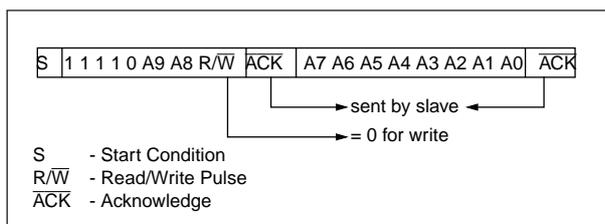


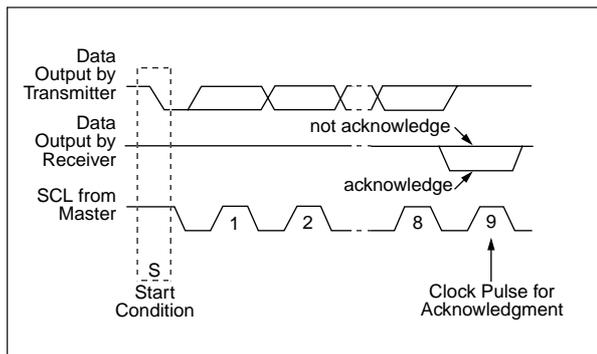
図11-9: I<sup>2</sup>Cの10ビットアドレスのフォーマット



### 11.2.3 応答の転送

すべてのデータはバイト毎に送信されなければならない。データ転送毎に送信されるバイト数は無制限です。各バイトの後に、スレーブレシーバが応答ビット(ACK)を発生します(図11-10参照)。スレーブレシーバがスレーブアドレスまたは受信データに応答しない時は、マスタは転送を打ち切ります。スレーブはマスタがSTOP条件を発生できるようにSDAをハイにしておく必要があります(図11-7参照)。

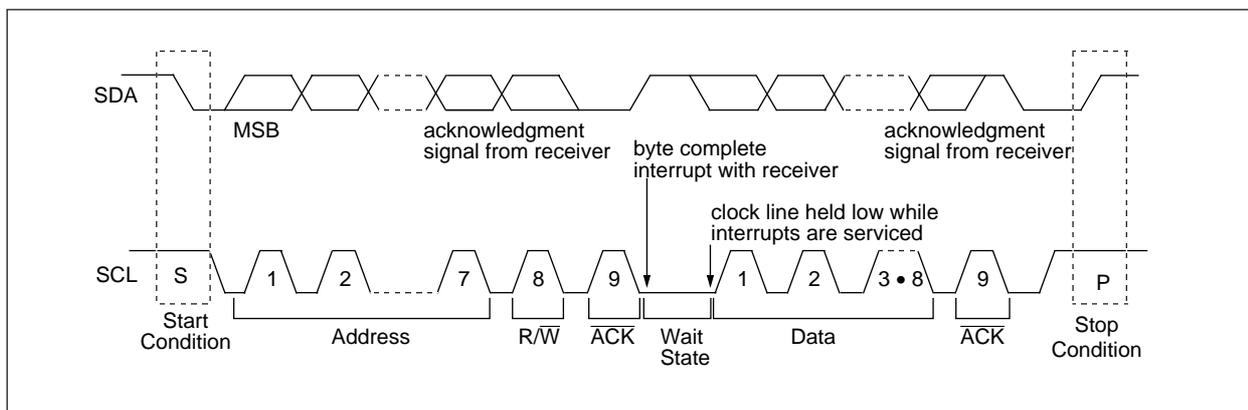
図11-10: スレーブ-レシーバの応答



マスタがデータを受信中(マスタレシーバ)の場合、最後のバイトを除いて、データの各受信されたバイトに対して応答信号を発生します。スレーブトランスミッタへのデータの最後を示す信号のために、マスタは応答を出しません(非応答)。そしてスレーブはマスタがSTOP条件を出せるようにSDAラインをリリースします。マスタはさらにデータ転送の有効な終了のための応答パルスの間にSTOP条件を出すこともできます。

スレーブが次のバイトの送信を遅らせる必要がある場合は、SCLラインをローに保持することにより、マスタがウェイトステートになります。スレーブがSCLラインをリリースすると、データ転送が継続されます。これによりクロックをスタートさせる前に、スレーブが受信したデータを移したり、転送する必要のあるデータをフェッチすることができます。このウェイトステート方法は、図11-11に示すようにビットレベルにも実施できます。

図11-11: データ転送のウェイトステート



# 暫定版

英語最新版データシートと併用しご利用下さい。  
 AC/DC仕様は最新英文データシートをご参照下さい。  
<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

# PIC16C7X

図11-12と11-13に、マスタ送信とマスタ受信のデータ転送の手順を示します。

マスタがバスを放棄したくない場合 (STOP条件を出すことにより)、START条件(Sr)を繰り返して発生する必要があります。この条件は開始条件(SCLがハイの間、SDAがハイからローになる)と同じですが、データ転送応答パルスの後に起こります(バスフリーステートではな

い)。これにより、マスタが“コマンド”をスレーブに送り、要求された情報を受け取るか、他のスレーブデバイスをアドレス指定することができます。この手順を図11-14に示します。

図11-12: マスタ送信のシーケンス

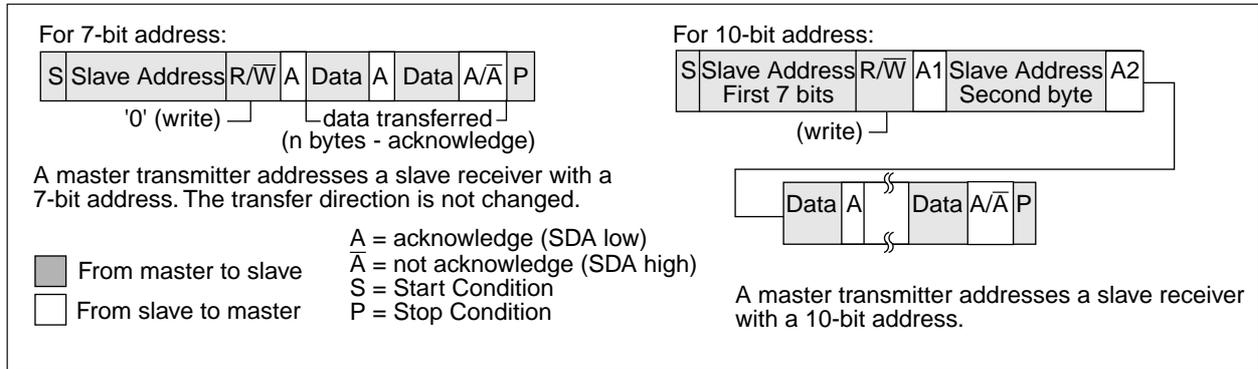


図11-13: マスタ受信のシーケンス

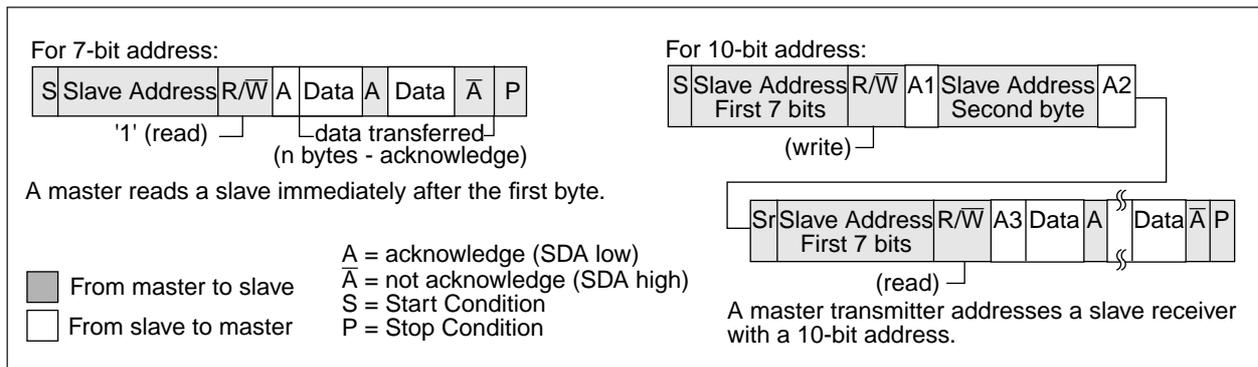
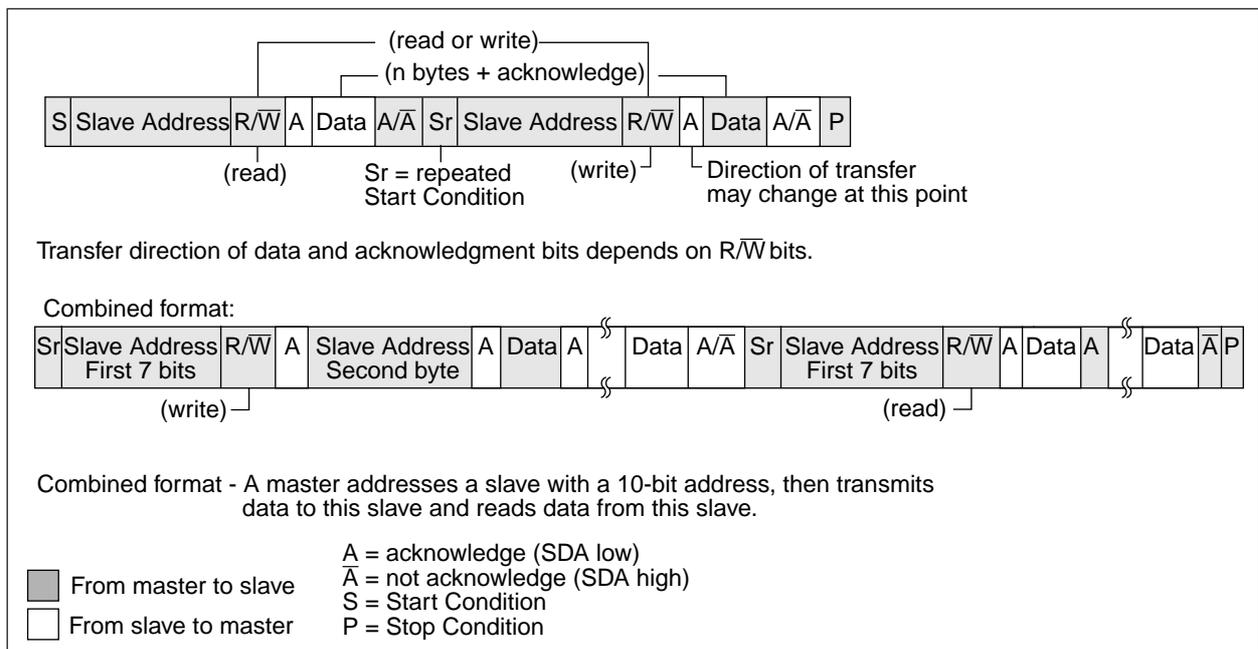


図11-14: 結合フォーマット



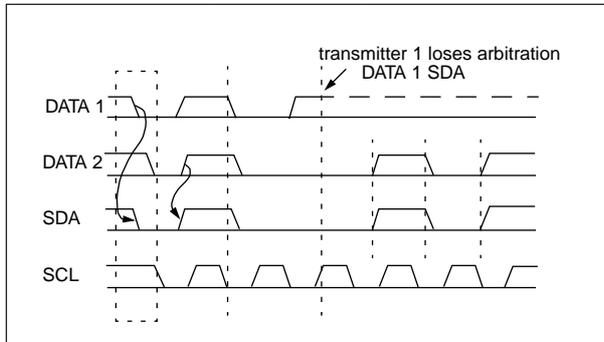
### 11.2.4 マルチ-マスタ

I<sup>2</sup>C プロトコルにより、システムが複数のマスタを持つことができます。これをマルチ-マスタと呼びます。2個以上のマスタが同時にデータを転送しようとする時、アービトレーションおよび同期が起こります。

#### 11.2.4.1 アービトレーション

SCLラインがハイの間、SDAラインでアービトレーションが起こります。他のマスタがローを送信する時にハイを送信するマスタは、アービトレーションを失い(図11-15参照)、そのデータ出力部分をオフにします。アービトレーションを失ったマスタは、アービトレーションを失ったデータバイトの最後までクロックパルスが発生することができます。マスタデバイスが同じデバイスをアドレス指定している時、アービトレーションはそのデータに続きます。

**図11-15 : マルチマスタアービトレーション (2 マスタ)**



スレーブ機能を取り入れ、アービトレーションを失ったマスタは、直ちにスレーブレシーバのモードに切り替わる必要があります。これは優位に立ったマスタ送信がそれをアドレス指定することがあるからです。

アービトレーションは次の間では許されません。

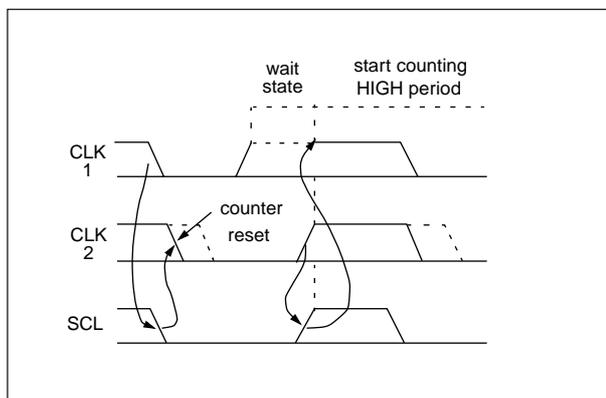
- START条件が繰り返されている間
- STOP条件とデータビットの間
- START条件の繰り返しとSTOP条件の間

これらの条件が起こらないよう注意が必要です。

### 11.2.4.2 クロックの同期

デバイスがアービトレーションを開始した後、クロックの同期が起こります。これはSCLラインとのワイヤードAND接続を使って実行されます。SCLラインでのハイからローへの変化によって、関係のあるデバイスがそのロー周期のカウントを始めることになります。デバイスのクロックがローになると、SCLのハイ状態に達するまで、SCLラインをローに保持します。他のデバイスのクロックがまだロー周期以内の場合、このクロックのローからハイへの変化がSCLラインの状態を変化させないことがあります。SCLラインは最も長いロー周期を持ったデバイスによりローに保持されます。より短いロー周期を持ったデバイスは、SCLラインがハイになるまで、ハイのウェイトステートに入ります。SCLラインがハイになると、すべてのデバイスがそのハイ周期のカウントを始めます。そのハイ周期を終えるための最初のデバイスはSCLラインをローに引き下げます。SCLラインのハイ・タイムは最も短いハイ周期を持ったデバイスにより決定されます(図11-16参照)。

**図11-16 : クロック同期**

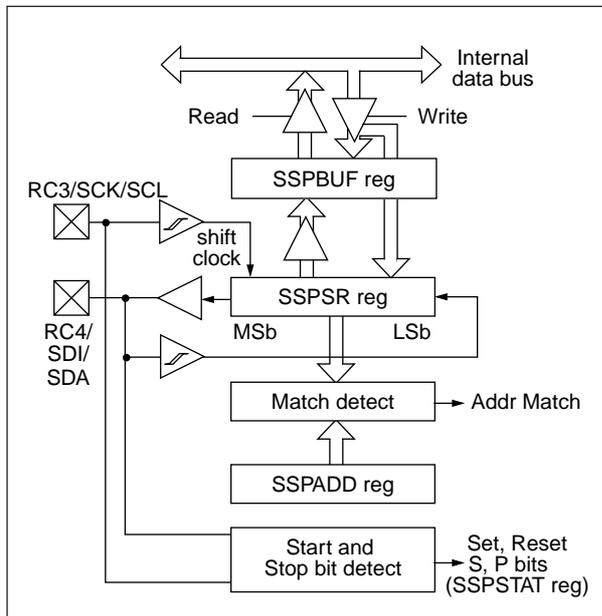


### 11.3 SSP I<sup>2</sup>Cの動作

適用デバイス							
710	71	711	72	73	73A	74	74A

I<sup>2</sup>CモードのSSPモジュールはすべてのスレーブ機能を完全に満たし、マスタ機能のソフトウェアの実行を容易にするためにハードウェアのスタートとストップのビットでの割り込みを備えています。SSPモジュールは7ビットと10ビットのアドレス指定と同様に、標準と高速モード規格を満たしています。2本のピンがデータ転送に使用されます。それらはクロック(SCL)であるRC3/SCK/SCLピンと、データ(SDA)であるRC4/SDI/SDAピンです。TRISC<4:3>ビットによって、これらのピンを入力または出力として設定する必要があります。SSPイネーブルビットSSPEN(SSPCON<5>)をセットすることにより、SSPモジュール機能をイネーブルにします。

図11-17: SPブロック図 (I<sup>2</sup>Cモード)



SSPモジュールにはI<sup>2</sup>C動作のために、次のような5種類のレジスタがあります。

- SSP制御レジスタ(SSPCON)
- SSPステータスレジスタ(SSPSTAT)
- シリアル受信/送信バッファ(SSPBUF)
- SSPシフトレジスタ(SSPSR)直接アクセス不可
- SSPアドレスレジスタ(SSPADD)

SSPCONレジスタによりI<sup>2</sup>C動作の制御が可能です。4種類のモード選択ビット(SSPCON<3:0>)により次のI<sup>2</sup>Cモードから1つを選択することができます。

- 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)
  - イネーブルされたスタートとストップビットの割り込みを伴う
- I<sup>2</sup>Cスレーブモード(10ビットアドレス)
  - イネーブルされたスタートとストップビットの割り込みを伴う
- I<sup>2</sup>Cのイネーブルされたスタートとストップビットの割り込み、スレーブはアイドル状態

SSPENビットのセットを持ったI<sup>2</sup>Cモードの選択により、SCLとSDAピンがオープンドレインとなり、これらのピンが適切なTRISCビットをセットすることにより、入力にプログラムされます。

SSPSTATレジスタはデータ転送の状態を示します。この情報にはSTARTとSTOPビットの検出が含まれ、受信されたバイトがデータかアドレスか、次のバイトが10ビットのアドレスの終わりかどうか、そしてこれがリードまたはライトのデータ転送かどうかを表わします。SSPSTATレジスタはリードのみです。

SSPBUFは転送データをライトまたはリードするレジスタです。SSPSRレジスタはそのデータをデバイスに対してシフトインまたはシフトアウトします。受信動作では、SSPBUFとSSPSRはダブルバッファされたレシーバを作ります。これにより、受信したデータの最後のバイトを読み込む前に、次のバイトの受信開始が可能になります。完全なバイトが受信された時、SSPBUFレジスタにそれを転送し、フラグビットSSPIFがセットされます。SSPBUFレジスタが読み込まれる前に他の完全なバイトが受信されると、レシーバオーバーフローが起こり、ビットSSPOV(SSPCON<6>)がセットされます。

SSPADDレジスタはスレーブアドレスを保持します。10ビットのモードでは、アドレスの上位バイトを書き込む必要があります(1111 0 A9 A8 0)。上位バイトのアドレス一致に続き、アドレスの下位バイトをロードする必要があります(A7:A0)。

# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。

## PIC16C7X

http://www.microchip.comより最新ERRATAシートを入手しご参照下さい。

### 11.3.1 スレーブモード

スレーブモードでは、SCLとSDAのピンを入力として設定する必要があります (TRISC<4:3>セット)。SSPモジュールは入力状態を優先し、必要な時にデータを出します (スレーブ-トランスミッタ)。

アドレスが一致するか、一致アドレスからのデータ転送が受信された時、ハードウェアが自動的に応答 (ACK) パルスを発生し、SSPSRレジスタが受信した値をSSPBUFレジスタにロードします。

SSPモジュールがこのACKパルスを与えなくなる条件があります。それは次のうちのどちらかまたは両方です。

- 転送が受信される前に、バッファフルビットBF (SSPSTAT<0>) がセットされていた。
- 転送が受信される前に、オーバーフロービットSSPOV (SSPCON<6>) がセットされていた。

この場合、SSPSRレジスタの値はSSPBUFにロードされませんが、ビットSSPIF (PIR1<3>) がセットされます。表11-3にデータ転送バイトが受信され、BFとSSPOVビットの状態が与えられた時に起こる状況を示します。網かけの部分は、ソフトウェアでオーバーフロー条件を正しくクリアしなかった場合の条件を示します。フラグビットBFはSSPBUFレジスタをリードすることでクリアされ、ビットSSPOVはソフトウェアでクリアされます。

SCLクロック入力は完全な動作をするために、最小のハイとローの時間が必要です。SSPモジュールの必要規格と同様に、1<sup>2</sup>Cの規格のハイ・タイムとロー・タイムをタイミングパラメータ#100と#101に示します。

#### 11.3.1.1 アドレス指定

SSPモジュールがイネーブルされると、SSPIはSTART条件が起きるのを待ちます。START条件の後、8ビットがSSPSRレジスタにシフトされます。入力されたすべてのビットはクロック (SCL) ラインの立ち上がりエッジでサンプリングされます。SSPSR<7:1>レジスタの値はSSPADDレジスタの値と比較されます。アドレスは8番目のクロック (SCL) パルスの立ち上がりエッジで比較されます。もしアドレスが一致し、BFとSSPOVビットがクリアの場合、次のことが起こります。

- SSPSRレジスタの値がSSPBUFレジスタにロードされる。
- バッファフルビットBFがセットされる。
- ACKパルスを発生。
- SSP割り込みフラグビットSSPIF (PIR1<3>) が9番目のSCLパルスの立ち上がりエッジでセットされる (イネーブルの時、割り込みが発生)。

10ビットのアドレスモードでは、2アドレスバイトがスレーブに受信される必要があります (図11-10参照)。最初のアドレスバイトの上位5ビット (MSBs) は、これが10ビットアドレスかどうかを示しています。ビットR/W (SSPSTAT<2>) はライトを示している必要があり、それによりスレーブデバイスは2番目のアドレスバイトを受信します。10ビットアドレスでは最初のバイトは '11110 A9 A8 0' で、ただしA9とA8はアドレスの上位2ビットです。10ビットアドレスで起こる順番は次のようになります。ステップ7-9はスレーブ-トランスミッタ用です。

- アドレスの最初の (上位) バイトを受信 (ビットSSPIF、BFとビットUA (SSPSTAT<1>) をセット)。
- アドレスの2番目の (下位) バイトでSSPADDレジスタをアップデート (ビットUAをクリアし、SCLラインをオフ)。
- SSPBUFレジスタをリード (ビットBFをクリア) し、フラグビットSSPIFをクリア。
- アドレスの2番目の (下位) バイトを受信 (ビットSSPIF、BF、UAをセット)。
- アドレスの最初の (上位) バイトで、SSPADDレジスタをアップデート (一致によりSCLラインがオフとなった場合、ビットUAをクリア)。
- SSPBUFレジスタをリード (ビットBFをクリア) し、フラグビットSSPIFをクリア。
- 繰り返しSTART条件を受信。
- アドレスの最初の (上位) バイトを受信 (ビットSSPIF、BFをセット)。
- SSPBUFレジスタをリード (ビットBFをクリア) し、フラグビットSSPIFをクリア。

表11-3: データ転送受信時のバイト動作

データ転送受信時のステータス・ビット				ACK Pulseの生成	ビット SSPIFのセット (イネーブル時のSSP割り込み)
BF	SSPOV	SSPSR	SSPBUF		
0	0	Yes		Yes	Yes
1	0	No		No	Yes
1	1	No		No	Yes
0	1	No		No	Yes

### 11.3.1.2 受信

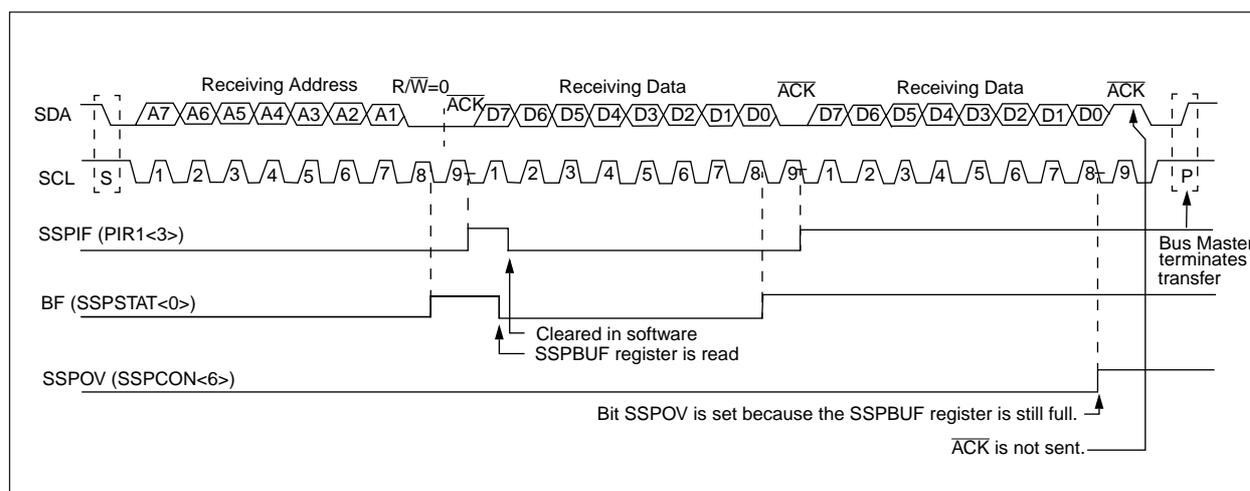
アドレスバイトのR/Wビットがクリアで、アドレスの一致が起こった時、SSPSTATレジスタのR/Wビットはクリアされます。受信されたアドレスはSSPBUFレジスタにロードされます。

アドレスバイトのオーバーフロー条件があると、応答(ACK)パルスは与えられません。オーバーフロー条件では、ビットBF(SSPSTAT<0>)がセットされているか、ビットSSPOV(SSPCON<6>)がセットされているかどちらかで定義されます。

SSP割込みは各データの転送毎に発生します。フラグビットSSPIF(PIR1<3>)はソフトウェアでクリアされる必要があり、SSPSTATレジスタはバイトの状態を確認するために使われます。

データを受信したか、していないかの確認には、SSPSTAT<BF> ビットは使用しないで下さい。PIR1<SSPIF> ビットを使用して下さい。

図11-18: 受信時のI<sup>2</sup>C波形(7ビットアドレス)



## PIC16C7X

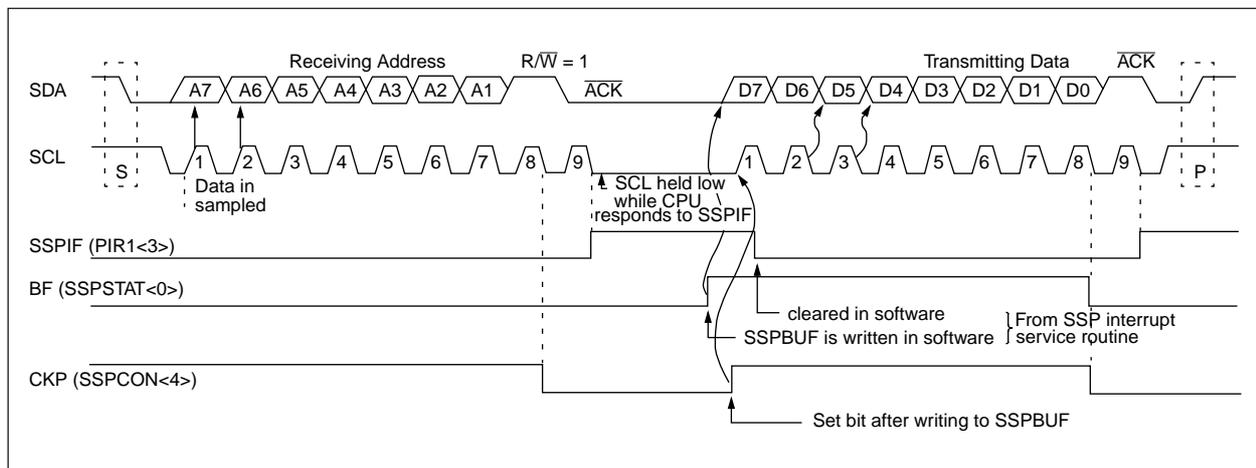
### 11.3.1.3 送信

入力されたアドレスバイトのR /  $\bar{W}$  ビットがセットされ、アドレスの一致が起こると、SSPSTATレジスタのR/ $\bar{W}$  ビットがセットされます。受信されたアドレスはSSPBUFレジスタにロードされます。ACKパルスは9番目のビットで送り出され、ピンRC3/SCK/SCLがローに保持されます。送信データをSSPBUFレジスタにロードする必要があり、そこからSSPSRレジスタにロードします。それからビットCKP (SSPCON<4>) をセットすることにより、ピンRC3/SCK/SCLをイネーブルにする必要があります。8個のデータ・ビットはSCL入力の立ち下がりエッジでシフトアウトされます。これによりSCLがハイ・タイムの間、SDA信号の有効性を確実にします(図11-19参照)。

SSP 割込みは各データの転送バイト毎に発生します。フラグビットSSPIFはソフトウェアでクリアされる必要があります。SSPSTATレジスタはバイトの状態を確認するために使われます。フラグビットSSPIFは9番目のクロックパルスの立ち下がりエッジでセットされます。

スレーブ-トランスミッタとして、マスタ-レシーバからのACKパルスは9番目のSCL入力パルスの立ち上がりエッジでラッチされます。SDAラインがハイ(ACKなし)の時、データ転送が終わります。それからスレーブは他のSTARTビットが起こるのを監視します。SDAラインがローの時(ACK)、送信データをSSPBUFレジスタにロードする必要があり、SSPBUFはSSPSRレジスタもロードします。そしてピンRC3/SCK/SCLは、ビットCKPをセットすることによりイネーブルにする必要があります。

図11-19 : 送信時のI2C 波形 (7ビットアドレス)



### 11.3.2 マスタモード

マスタモードの動作はSTARTとSTOP条件の検出時の割り込み発生によりサポートされています。STOP(P)とSTART(S)のビットがリセットか、SSPモジュールがディセーブルになるとクリアされます。P ビットがセットされるか、バスがアイドル状態でSとPビットの両方がクリアされる時、I<sup>2</sup>Cバスの制御が行われます。

マスタモードでは、対応するTRISC<4:3>ビットをクリアすることにより、SCLとSDAのラインを操作します。この出力レベルは常にローで、PORTB<4:3>の値には関係しません。したがってデータの送信時は、'1' データビットはTRISC<4>ビットがセット(入力)で、'0' データビットはTRISC<4>ビットがクリア(出力)になっている必要があります。TRISC<3>ビットによるSCLラインについても同様のことが言えます。

次のイベントでSSP割り込みフラグビットSSPIFがセットされます(イネーブルの時SSP割り込み)。

- ・ START条件
- ・ STOP条件
- ・ データ転送バイトの送信/受信

マスタモードの動作はスレーブモードのアイドル状態(SSPM3:SSPM0=1011)か、スレーブアクティブ時に実行できます。マスタとスレーブの両方のモードがイネーブルの場合、ソフトウェアで異なった割り込み要因にする必要があります。

### 11.3.3 マルチ-マスタモード

マルチ-マスタモードでは、STARTとSTOPの条件検出時の割り込み発生により、バスがフリーの時を判定できます。STOP(P)とSTART(S)のビットがリセットか、SSPモジュールがディセーブルになるとクリアされます。ビットP(SSPSTAT<4>)がセットされるか、バスがアイドル状態でSとPビットの両方がクリアされる時、I<sup>2</sup>Cバスの制御が行われます。バスがビジーの時、SSP割り込みをイネーブルにすることで、STOP条件が起こった時その割り込みを発生します。

マルチ-マスタ動作では、信号レベルが必要な出力レベルかどうか見るために、SDAラインを監視する必要があります。このチェックはハイレベルが出力された時だけ行う必要があります。ハイレベルを期待している時にローレベルが出力された場合、デバイスはSDAとSCLラインをリリースする(TRISC<4:3>をセット)必要があります。次の2つの状態では、このアービトレーションが失われます。

- ・ アドレス転送
- ・ データ転送

スレーブロジックがイネーブルの場合、スレーブは受信を継続します。アドレス転送中にアービトレーションを失うと、そのデバイスがアドレス指定を受けることもあります。アドレス指定されると、ACKパルスを発生します。データ転送中にアービトレーションを失うと、そのデバイスは後でデータを再転送する必要があります。

表11-4: I<sup>2</sup>C 操作に関連するレジスタ

アドレス	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR BOR での値	その他 リセット での値
0Bh/8Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBF	0000 000x	0000 000u
0Ch	PIR1	PSPIF <sup>(1,2)</sup>	ADIF	RCIF <sup>(2)</sup>	TXIF <sup>(2)</sup>	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE <sup>(1,2)</sup>	ADIE	RCIE <sup>(2)</sup>	TXIE <sup>(2)</sup>	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
13h	SSPBUF	同期シリアルポート受信バッファ/送信レジスタ								xxxx xxxx	uuuu uuuu
93h	SSPADD	同期シリアルポート (I <sup>2</sup> C モード) アドレスレジスタ								0000 0000	0000 0000
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
94h	SSPSTAT	-	-	D/Ā	P	S	R/W	UA	BF	--00 0000	--00 0000
89h	TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISCO	1111 1111	1111 1111

凡例: x=未知、u=不変、-=未使用、'0'としてリード。網かけ部分はI<sup>2</sup>CモードのSSPでは使われません。

注 1: PSPIEとPSPIFビットはPIC16C73/73Aではリザーブされ、常にこれらのビットはクリアされた状態です。

2: PIC16C72にはパラレル・スレーブ・ポート、USARTがありません。

これらのビットは未使用で、'0'としてリードします。

図11-20 : IDLE\_MODE、RCV\_MODE、XMIT\_MODEでのI<sup>2</sup>C モジュール操作

<p><b>IDLE_MODE (7-bit):</b> if (Addr_match )</p> <pre>                 {      Set interrupt;                 if (R/W = 1)  {      Send <math>\overline{\text{ACK}}</math> = 0;                                 set XMIT_MODE;                                 }                 else if (R/W = 0) set RCV_MODE;                 }         </pre>
<p><b>RCV_MODE:</b> if ((SSPBUF=Full) OR (SSPOV = 1))</p> <pre>         {      Set SSPOV;           Do not acknowledge;         } else {      transfer SSPSR → SSPBUF;         send ACK = 0;     }     Receive 8-bits in SSPSR;     Set interrupt;     </pre>
<p><b>XMIT_MODE:</b> While ((SSPBUF = Empty) AND (CKP=0)) Hold SCL Low; Send byte; Set interrupt; if ( <math>\overline{\text{ACK}}</math> Received = 1)      {      End of transmission;                                 Go back to IDLE_MODE;                                 } else if ( <math>\overline{\text{ACK}}</math> Received = 0)      Go back to XMIT_MODE;</p>
<p><b>IDLE_MODE (10-Bit):</b> If (High_byte_addr_match AND (R/W = 0))</p> <pre>         {      PRIOR_ADDR_MATCH = FALSE;           Set interrupt;           if ((SSPBUF = Full) OR ((SSPOV = 1))               {      Set SSPOV;                 Do not acknowledge;               }           else {      Set <math>\overline{\text{UA}}</math> = 1;                 Send <math>\overline{\text{ACK}}</math> = 0;                 While (SSPADD not updated) Hold SCL low;                 Clear <math>\overline{\text{UA}}</math> = 0;                 Receive Low_addr_byte;                 Set interrupt;                 Set <math>\overline{\text{UA}}</math> = 1;                 If (Low_byte_addr_match)                     {      PRIOR_ADDR_MATCH = TRUE;                       Send <math>\overline{\text{ACK}}</math> = 0;                       while (SSPADD not updated) Hold SCL low;                       Clear <math>\overline{\text{UA}}</math> = 0;                       Set RCV_MODE;                     }                 }         }     </pre> <p>else if (High_byte_addr_match AND (R/W = 1))</p> <pre>         {      if (PRIOR_ADDR_MATCH)             {      send <math>\overline{\text{ACK}}</math> = 0;                 set XMIT_MODE;             }         }     else PRIOR_ADDR_MATCH = FALSE;     }     </pre>

### 12.0 ユニバーサル同期非同期の受信と送信(USART)

適用デバイス							
710	71	711	72	73	73A	74	74A

ユニバーサル同期非同期のレシーバトランスミッタ(USART)モジュールは、2つのシリアルI/Oモジュールの1つです(USARTはシリアルコミュニケーションインターフェイスまたはSCIとしても知られています)。USARTは、CRT端子やパーソナルコンピュータのように周辺機能のデバイスと通信することができる全二重非同期システムとして設定することができます。またはA/Dある

いはD/Aの集積回路やシリアルEEPROMなどのように周辺機能のデバイスと通信することができる半二重同期システムとして設定することができます。

USARTは次のモードに設定することができます。

- ・ 非同期(全二重)
- ・ 同期 - マスタ(半二重)
- ・ 同期 - スレーブ(半二重)

ビットSPEN(RCSTA<7>)とビットTRISC<7:6>は、シリアルコミュニケーションインターフェイスとしてピンRC6/TX/CKとRC7/RX/DTを設定するためにセットする必要があります。

図12-1: TXSTA: 送信ステータスと制御レジスタ(アドレス 98h)

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-1	R/W-0	
CSRC	TX9	TXEN	SYNC		BRGH	TRMT	TX9D	
bit7					bit0			
bit 7:	<b>CSRC:</b> クロックソース選択ビット							
	非同期モード							
	無視							
	同期モード							
	1 = マスタモード(BRGから内部に発生したクロック)							
	0 = スレーブモード(外部ソースからのクロック)							
bit 6:	<b>TX9:</b> 9ビットの送信イネーブルビット							
	1 = 9ビットの送信を選択							
	0 = 8ビットの送信を選択							
bit 5:	<b>TXEN:</b> 送信イネーブルビット							
	1 = 送信をイネーブル							
	0 = 送信をディセーブル							
	注意: SREN/CRENはSYNCモードでのTXENを優先。							
bit 4:	<b>SYNC:</b> USARTモード選択ビット							
	1 = 同期モード							
	0 = 非同期モード							
bit 3:	<b>未使用:</b> '0'としてリード							
bit 2:	<b>BRGH:</b> ハイボーレート選択ビット							
	非同期モード							
	1 = 高スピード(PIC16C63, 63A, 65, 65A, 73, 73A, 74, 74Aでは受信時使用できません。)							
	0 = 低スピード							
	同期モード							
	このモードでは使われない。							
bit 1:	<b>TRMT:</b> 送信シフトレジスタステータスビット							
	1 = TSRは空							
	0 = TSRはフル							
bit 0:	<b>TX9D:</b> 送信データの9番目のビット。パリティビットとして可能(自動生成されません。)							

R = 読み込み可能なビット  
W = 書き込み可能なビット  
U = 未使用のビット、  
'0'としてリード  
- n = PORリセットでの値

# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。

## PIC16C7X

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

図12-2 : RCSTA : 受信ステータスと制御レジスタ ( アドレス 18h )

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	-	FERR	OERR	RX9D
bit7							bit0

R = 読み込み可能なビット  
W = 書き込み可能なビット  
U = 未使用のビット、  
‘0’としてリード  
- n = PORリセットでの値

bit 7: **SPEN**: シリアルポートイネーブル ビット  
1 = シリアルポートをイネーブル(RC7/RX/DTとRC6/TX/CKピンをシリアルポートピンとして設定)  
0 = シリアルポートをディセーブル

bit 6: **RX9**: 9ビットの受信イネーブルビット  
1 = 9ビットの受信を選択  
0 = 8ビットの受信を選択

bit 5: **SREN**: シングル受信イネーブルビット  
非同期モード  
無視  
同期モード - マスタ  
1 = シングル受信をイネーブル  
0 = シングル受信をディセーブル  
このビットは受信が完了した後クリアされる。  
同期モード - スレーブ  
このモードでは使われない。

bit 4: **CREN**: 継続受信イネーブルビット  
非同期モード  
1 = 継続受信をイネーブル  
0 = 継続受信をディセーブル  
同期モード  
1 = イネーブルビットCRENがクリアされるまで継続受信をイネーブル(CRENはSRENを優先)  
0 = 継続受信をディセーブル

bit 3: **未使用**: ‘0’としてリード

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(RCREGレジスタを読み込むことによりアップデート可能)  
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(ビットCRENをクリアすることによりクリア可能)  
0 = オーバーランエラーなし

bit 0: **RX9D**: 受信データの9番目のビット(パリティビット可能)

### 12.1 USARTボーレートジェネレータ(BRG)

適用デバイス							
710	71	711	72	73	73A	74	74A

BRGはUSARTの非同期、同期の両モードをサポートしています。それは与えられた8ビットのボーレートジェネレータです。SPBRGレジスタはフリーランニングの8ビットタイマの周期を制御します。非同期モードでは、ビットBRGH(TXSTA<2>)もボーレートを制御します。同期モードではビットBRGHは無視されます。表12-1に異なったUSARTモードに対するボーレートの計算の公式を示しますが、それはマスタモード(内部クロック)にのみ適用します。

必要なボーレートとFoscが与えられると、SPBRGレジスタに最も近い整数値を表12-1の公式を使って数えることができます。これによりボーレートのエラーが決まります。

例12-1に次の条件に対するボーレートエラーの計算を示します。

Fosc=16MHz  
必要なボーレート=9600  
BRGH=0  
SYNC=0

#### 例12-1: ボーレートエラーの計算

$$\begin{aligned} \text{Desired Baud rate} &= \text{Fosc} / (64 (X + 1)) \\ 9600 &= 16000000 / (64 (X + 1)) \\ X &= \lceil 25.042 \rceil = 25 \\ \text{Calculated Baud Rate} &= 16000000 / (64 (25 + 1)) \\ &= 9615 \\ \text{Error} &= \frac{(\text{Calculated Baud Rate} - \text{Desired Baud Rate})}{\text{Desired Baud Rate}} \\ &= (9615 - 9600) / 9600 \\ &= 0.16\% \end{aligned}$$

より遅いボーレートに対してでもハイボーレート(BRGH=1)を使うのが便利です。それはFosc/(16(x+1))の方程式が、いくつかの場合においてボーレートエラーを減らすことができるからです。

PIC16C63, 63A, 65, 65A, 73, 73A, 74, 74AのBRGH=1は、受信時使用できません。

SPBRGレジスタに新しい値を書き込むことによりBRGタイマがリセットされ(またはクリアされ)、これにより新しいボーレートを出力する前にBRGがタイマーオーバーフローを待たないことを確実にします。

表12-1: ボーレートの公式

SYNC	BRGH = 0 (ロースピード)	BRGH = 1 (ハイスピード)
0	(非同期)ボーレート = Fosc/(64(X+1))	ボーレート = Fosc/(16(X+1))
1	(同期)ボーレート = Fosc/(4(X+1))	NA

X = SPBRG の値(0から255)

(PIC16C63, 63A, 65, 65A, 73, 73A, 74, 74AのBRGH=1は、受信時使用できません。)

表12-2: ボーレートジェネレータに関連するレジスタ

アドレス	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR BOR での値	その他の リセット での値
98h	TXSTA	CSRC	TX9	TXEN	SYNC	-	BRGH	TRMT	TX9D	0000 -010	0000 -010
18h	RCSTA	SPEN	RX9	SREN	CREN	-	FERR	OERR	RX9D	0000 -00x	0000 -00x
99h	SPBRG	ボーレートジェネレータレジスタ								0000 0000	0000 0000

凡例: x = 未知, - = 未使用、' 0 ' 網掛け部分はBRGでは使われません。

# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。

## PIC16C7X

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

表12-3: 同期モードに関するボーレート

BAUD RATE (K)	Fosc = 20 MHz			16 MHz			10 MHz			7.15909 MHz		
	KBAUD	%ERROR	SPBRG value (decimal)	KBAUD	%ERROR	SPBRG value (decimal)	KBAUD	%ERROR	SPBRG value (decimal)	KBAUD	%ERROR	SPBRG value (decimal)
0.3	NA	-	-	NA	-	-	NA	-	-	NA	-	-
1.2	NA	-	-	NA	-	-	NA	-	-	NA	-	-
2.4	NA	-	-	NA	-	-	NA	-	-	NA	-	-
9.6	NA	-	-	NA	-	-	9.766	1.73	255	9.622	0.23	185
19.2	19.53	1.73	255	19.23	0.16	207	19.23	0.16	129	19.24	0.23	92
76.8	76.92	0.16	64	76.92	0.16	51	75.76	-1.36	32	77.82	1.32	22
96	96.15	0.16	51	95.24	-0.79	41	96.15	0.16	25	94.2	-1.88	18
300	294.1	-1.96	16	307.69	2.56	12	312.5	4.17	7	298.3	-0.57	5
500	500	0	9	500	0	7	500	0	4	NA	-	-
HIGH	5000	-	0	4000	-	0	2500	-	0	1789.8	-	0
LOW	19.53	-	255	15.625	-	255	9.766	-	255	6.991	-	255

BAUD RATE (K)	Fosc = 5.0688 MHz			3.579545 MHz			1 MHz			32.768 kHz		
	KBAUD	%ERROR	SPBRG value (decimal)	KBAUD	%ERROR	SPBRG value (decimal)	KBAUD	%ERROR	SPBRG value (decimal)	KBAUD	%ERROR	SPBRG value (decimal)
0.3	NA	-	-	NA	-	-	NA	-	-	0.303	1.14	26
1.2	NA	-	-	NA	-	-	1.202	0.16	207	1.17	-2.48	6
2.4	NA	-	-	NA	-	-	2.404	0.16	103	NA	-	-
9.6	9.6	0	131	9.622	0.23	92	9.615	0.16	25	NA	-	-
19.2	19.2	0	65	19.04	-0.83	46	19.24	0.16	12	NA	-	-
76.8	79.2	3.13	15	74.57	-2.9	11	83.34	8.51	2	NA	-	-
96	97.48	1.54	12	99.43	3.57	8	NA	-	-	NA	-	-
300	316.8	5.6	3	298.3	-0.57	2	NA	-	-	NA	-	-
500	NA	-	-	NA	-	-	NA	-	-	NA	-	-
HIGH	1267	-	0	894.9	-	0	250	-	0	8.192	-	0
LOW	4.95	-	255	3.496	-	255	0.9766	-	255	0.032	-	255

表12-4: 非同期モードに関するボーレート (BRGH=0)

BAUD RATE (K)	Fosc = 20 MHz			16 MHz			10 MHz			7.15909 MHz		
	KBAUD	%ERROR	SPBRG value (decimal)	KBAUD	%ERROR	SPBRG value (decimal)	KBAUD	%ERROR	SPBRG value (decimal)	KBAUD	%ERROR	SPBRG value (decimal)
0.3	NA	-	-	NA	-	-	NA	-	-	NA	-	-
1.2	1.221	1.73	255	1.202	0.16	207	1.202	0.16	129	1.203	0.23	92
2.4	2.404	0.16	129	2.404	0.16	103	2.404	0.16	64	2.38	-0.83	46
9.6	9.469	-1.36	32	9.615	0.16	25	9.766	1.73	15	9.322	-2.9	11
19.2	19.53	1.73	15	19.23	0.16	12	19.53	1.73	7	18.64	-2.9	5
76.8	78.13	1.73	3	83.33	8.51	2	78.13	1.73	1	NA	-	-
96	104.2	8.51	2	NA	-	-	NA	-	-	NA	-	-
300	312.5	4.17	0	NA	-	-	NA	-	-	NA	-	-
500	NA	-	-	NA	-	-	NA	-	-	NA	-	-
HIGH	312.5	-	0	250	-	0	156.3	-	0	111.9	-	0
LOW	1.221	-	255	0.977	-	255	0.6104	-	255	0.437	-	255

BAUD RATE (K)	Fosc = 5.0688 MHz			3.579545 MHz			1 MHz			32.768 kHz		
	KBAUD	%ERROR	SPBRG value (decimal)	KBAUD	%ERROR	SPBRG value (decimal)	KBAUD	%ERROR	SPBRG value (decimal)	KBAUD	%ERROR	SPBRG value (decimal)
0.3	0.31	3.13	255	0.301	0.23	185	0.3	0.16	51	0.256	-14.67	1
1.2	1.2	0	65	1.19	-0.83	46	1.202	0.16	12	NA	-	-
2.4	2.4	0	32	2.432	1.32	22	2.232	-6.99	6	NA	-	-
9.6	9.9	3.13	7	9.322	-2.9	5	NA	-	-	NA	-	-
19.2	19.8	3.13	3	18.64	-2.9	2	NA	-	-	NA	-	-
76.8	79.2	3.13	0	NA	-	-	NA	-	-	NA	-	-
96	NA	-	-	NA	-	-	NA	-	-	NA	-	-
300	NA	-	-	NA	-	-	NA	-	-	NA	-	-
500	NA	-	-	NA	-	-	NA	-	-	NA	-	-
HIGH	79.2	-	0	55.93	-	0	15.63	-	0	0.512	-	0
LOW	0.3094	-	255	0.2185	-	255	0.061	-	255	0.002	-	255

# 暫定版

英語最新版データシートと併用ご利用下さい。  
 AC/DC仕様は最新英文データシートをご参照下さい。  
<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

表12-5: 非同期モードに関するボーレート (BRGH=1)

BAUD RATE (K)	Fosc = 20 MHz			16 MHz			10 MHz			7.16 MHz		
	KBAUD	%ERROR	SPBRG value (decimal)	KBAUD	%ERROR	SPBRG value (decimal)	KBAUD	%ERROR	SPBRG value (decimal)	KBAUD	%ERROR	SPBRG value (decimal)
9.6	9.615	0.16	129	9.615	0.16	103	9.615	0.16	64	9.52	-0.83	46
19.2	19.23	0.16	64	19.23	0.16	51	18.939	-1.3632	19.454	1.32	22	
38.4	37.878	-1.36	32	38.461	0.16	25	39.062	1.7	15	37.286	-2.9	11
57.6	56.818	-1.3621	58.823	2.12	16	56.818	-1.36	10	55.93	-2.9	7	
115.2	113.636	-1.36	10	111.111	-3.55	8	125	8.51	4	111.86	-2.903	
250	250	0	4	250	0	3	NA	-	-	NA	-	-
625	625	0	1	NA	--	625	0	0	NA	-	-	-
1250	1250	0	0	NA	-	-	NA	-	-	NA	-	-

BAUD RATE (K)	Fosc = 5.0688 MHz			3.579 MHz			1 MHz			32.768 kHz		
	KBAUD	%ERROR	SPBRG value (decimal)	KBAUD	%ERROR	SPBRG value (decimal)	KBAUD	%ERROR	SPBRG value (decimal)	KBAUD	%ERROR	SPBRG value (decimal)
9.6	9.6	0	32	9.727	1.32	22	8.928	-6.99	6	NA	-	-
19.2	18.645	-2.94	16	18.643	-2.9	11	20.833	8.51	2	NA	-	-
38.4	39.6	3.12	7	37.286	-2.9	5	31.25	-18.61	1	NA	-	-
57.6	52.8	-8.33	5	55.93	-2.9	3	62.5	8.51	0	NA	-	-
115.2	105.6	-8.33	2	111.86	-2.9	1	NA	-	-	NA	-	-
250	NA	-	-	223.721	-10.51	0	NA	-	-	NA	-	-
625	NA	-	-	NA	-	-	NA	-	-	NA	-	-
1250	NA	-	-	NA	-	-	NA	-	-	NA	-	-

# 暫定版

英語最新版データシートと併用にご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。

## PIC16C7X

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

### 12.1.1 サンプルング

RC7/RX/DTピンのデータは、RXピンが高レベルまたは低レベルであるかどうかを決めるために多数検出回路により3回サンプルングされます。ビットBRGH(TXSTA<2>)がクリアされると(ローポーレートにおいて)、サンプルングはx16クロックの7番目、8番目、9番目の立ち下

がりエッジで行われます(図12-3参照)。ビットBRGHがセットされると(ハイポーレートにおいて)、サンプルングはx4クロックの最初の立ち下がりエッジの後、2番目の立ち上がりエッジより先に起こる3クロックエッジ上で起こります(図12-4と12-5参照)。

図12-3: RXピンのサンプルング図 (BRGH=0)

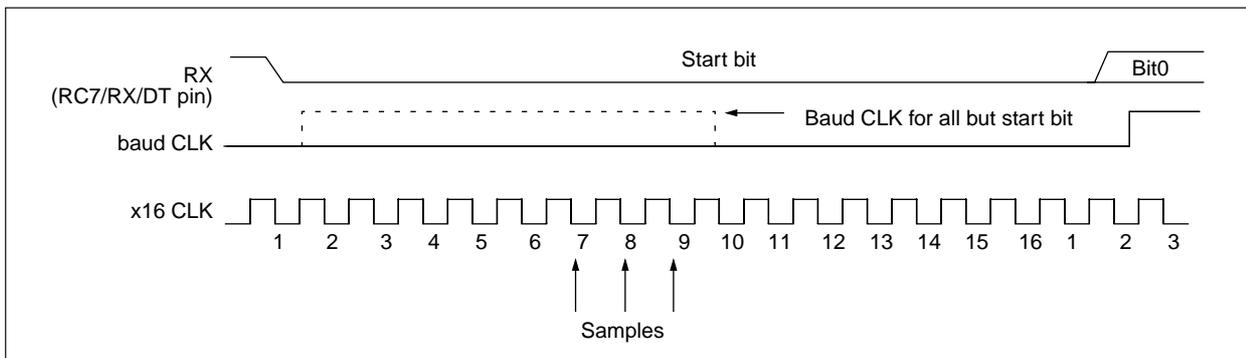


図12-4: RXピンのサンプルング図 (BRGH=1)

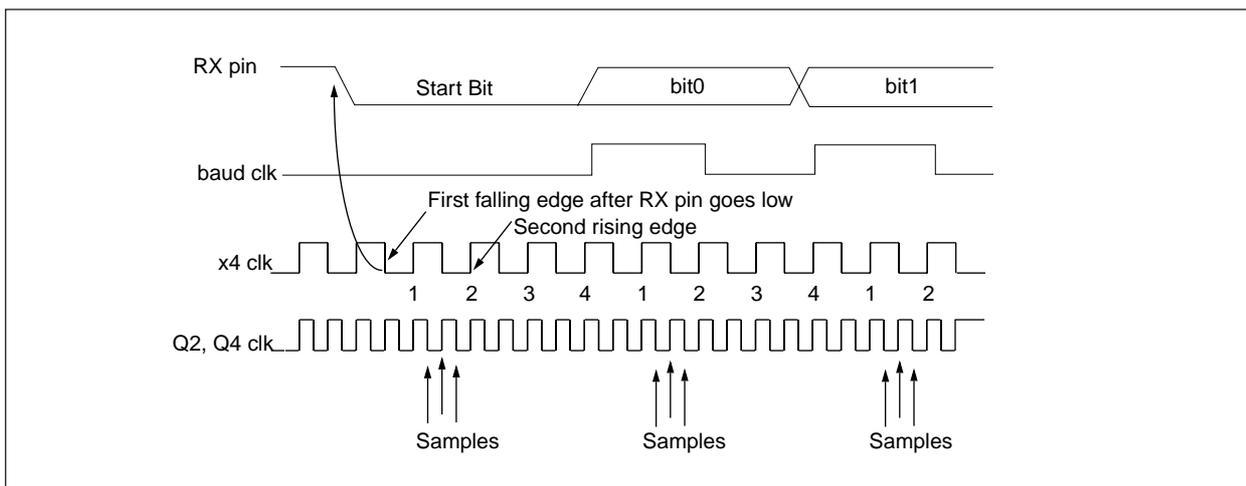
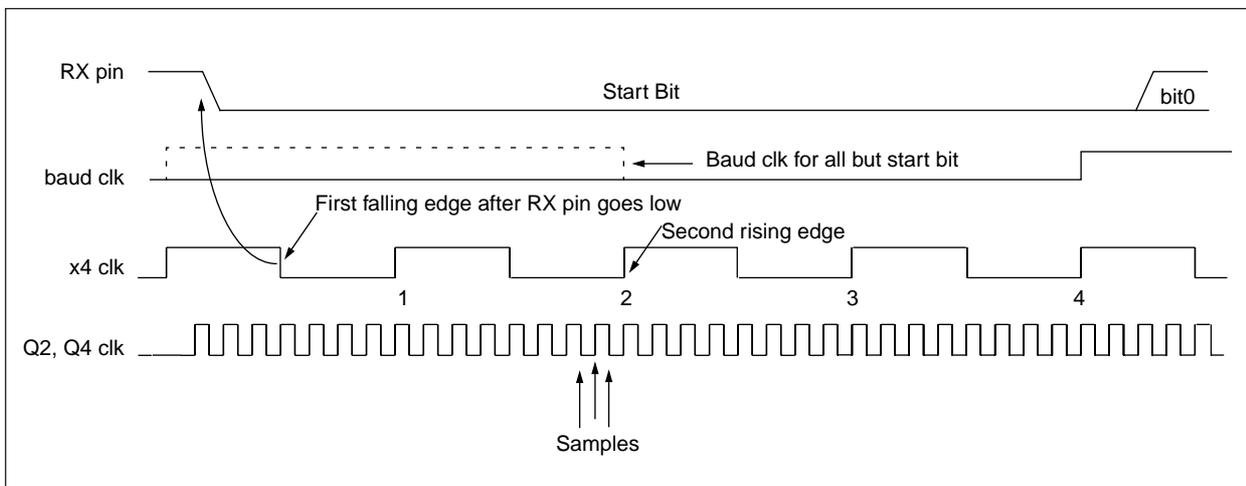


図12-5: RXピンのサンプルング図 (BRGH=1)



### 12.2 USART非同期モード

適用デバイス							
710	71	711	72	73	73A	74	74A

このモードでは、USARTは標準の非ゼロ復帰(NRZ)フォーマットを使用します(1スタートビット、8または9データビット、1ストップビット)。最も一般的なデータフォーマットは8ビットです。8ビットのオンチップポーレートジェネレータはオシレータから標準のポーレート周波数を得るために使われます。USARTは最初にLSbを送信し受信します。USARTの送信機と受信機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ポーレートを使います。ポーレートジェネレータは、BRGHビット(TXSTA<2>)に頼りながら、ビットシフトレートの×16または×64どちらかのクロックを作ります。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、ソフトウェアで実行することができます(9番目のデータビットとして保持することも可能)。非同期モードは、SLEEPの間は止まっています。

非同期モードはビットSYNC(TXSTA<4>)をクリアすることにより選ばれます。

USARTの非同期モジュールは次のような重要な要素から成っています。

- ・ポーレートジェネレータ
- ・サンプリング回路
- ・非同期送信機
- ・非同期受信機

#### 12.2.1 USART非同期送信機

USART送信機のブロック図を図12-6に示します。送信機の中心は、送信(シリアル)シフトレジスタです(TSR)。シフトレジスタはリード/ライト送信バッファTXREGからのデータを得ます。TXREGレジスタはソフトウェアでのデータにロードします。TSRレジスタはSTOPビットが前の負荷から送信されるまでロードされません。STOPビットが送信されるとすぐ、TSRはTXREGレジスタ(有効なら)から新しいデータにロードします。TXREGレジスタがそのデータをTSRレジスタに転送すると(1つの $T_{cy}$ の中で起こる)、TXREGレジスタは空になり、フラグビットTXIF(PIR1<4>)がセットされます。この割込みはイネーブルビットTXIE(PIE1<4>)をセット/クリアすることによりイネーブルまたはディセーブルできま

す。フラグビットTXIFはイネーブルビットTXIEの状態を無視してセットされ、ソフトウェアではクリアできず、新しいデータがTXREGレジスタにロードされた時にだけリセットします。フラグビットTXIFがTXREGレジスタのステータスを示している間に、もう1つのビットTRMT(TXSTA<1>)がTSRレジスタのステータスを示します。ステータスビットTRMTはTSRレジスタが空の時セットされるリードのみのビットです。割込みなしの論理はこのビットにつながるため、TSRレジスタが空かどうかを確かめるためにこのビットを登録しなければなりません。

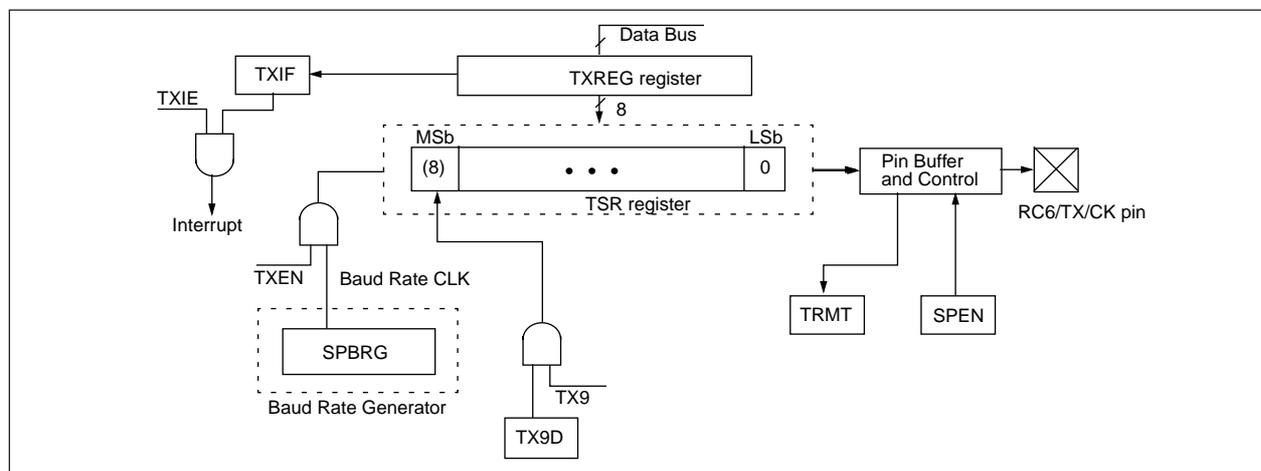
注意1: TSRレジスタはユーザ用ではないので、データメモリには表われません。

注意2: フラグビットTXIFはイネーブルビットTXENがセットされる時にセットされます。

送信はイネーブルビットTXEN(TXSTA<5>)をセットすることによりイネーブルになります。現実の送信はTXREGレジスタがデータにロードされ、ポーレートジェネレータ(BRG)がシフトクロックを作るまで起こりません(図12-6参照)。送信は最初にTXREGレジスタをロードし、イネーブルビットTXENをセットすることにより始動させることもできます。一般的には、最初に送信を始動させるとTSRレジスタが空になり、TXREGレジスタへの転送が結果的に、空になったTXREGであるTSRにただちに転送することになります。このように連続した転送は可能です(図12-8参照)。送信中にイネーブルビットTXENをクリアすることにより送信は打ち切れ、送信機をリセットします。結果としてRC6/TX/CKピンは高インピーダンスに戻ります。

9ビットの送信を選ぶためには、送信ビットTX9(TXSTA<6>)をセットし、9番目のビットをTX9D(TXSTA<0>)に書き込まなければなりません。9番目のビットは、8ビットのデータをTXREGレジスタに書き込む前に、書き込む必要があります。これはTXREGレジスタに書き込むデータが結果的にTSRレジスタへのデータにただちに転送することが可能だからです(TSRが空の場合)。このようなケースでは、正しくない9番目のデータビットはTSRレジスタにロードされることがあります。

図12-6: USART送信のブロック図



# 暫定版

英語最新版データシートと併用にご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。

## PIC16C7X

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

非同期の送信をセットする場合は次のように進めません。

1. 適当なボーレートにSPBRGレジスタを初期化する。高スピードのボーレートが必要な場合はビットBRGHをセットする(12.1章参照)。
2. ビットSYNCをクリアし、ビットSPENをセットすることにより非同期のシリアルポートをイネーブルにする。
3. 割込みが必要な場合、イネーブルビットTXIEをセットする。

4. 9ビットの送信が必要な場合は、送信ビットTX9をセットする。
5. ビットTXENをセットすることにより送信をイネーブルにすると、ビットTXIFもセットすることになる。
6. 9ビットの送信を選んだ場合は、9番目のビットはビットTX9Dにロードする必要がある。
7. TXREGレジスタにデータをロードする(送信開始)。

図12-7： 非同期のマスタ送信

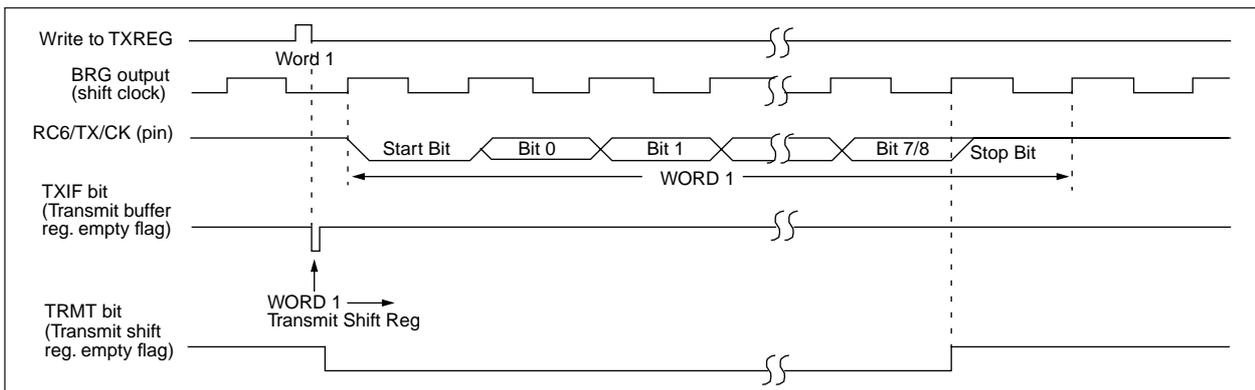


図12-8： 非同期のマスタ送信(連続)

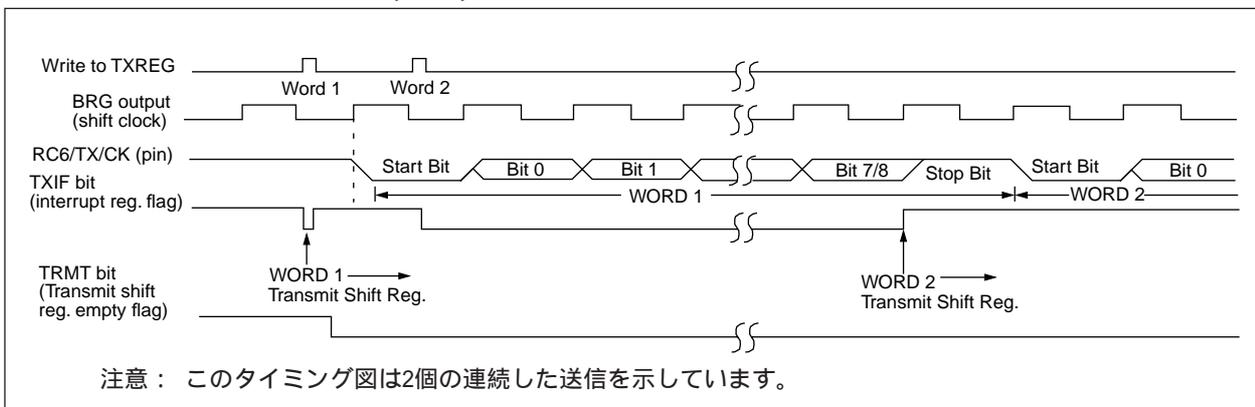


表12-6： 非同期の送信に関連するレジスタ

アドレス	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR BORでの値	他のリセットでの値
0Ch	PIR1	PSPIF(1)	ADIF	RC1F	TX1F	SSPIF	CCP1F	TMR21F	TMR11F	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	-	FERR	OERR	RX9D	0000 -00x	0000 -00x
19h	TXREG	TX7	TX6	TX5	TX4	TX3	TX2	TX1	TX0	0000 0000	0000 0000
8Ch	PIE1	PSPIE(1)	ADIE	RC1E	TX1E	SSPIE	CCP1E	TMR21E	TMR11E	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	-	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	ボーレートジェネレータレジスタ								0000 0000	0000 0000

凡例： x=未知、-=未使用のロケーション、'0'としてリード。網掛け部分は非同期送信には使われません。

注1： PSPIEとPSPIFビットはPIC16C73/73Aではリザーブされており、常にこれらのビットはクリアされた状態です。

### 12.2.2 USARTの非同期受信機

受信機のブロック図を図12-9に示します。データはRC7/RX/DTピン上で受信され、データ回復ブロックを動かします。主な受信のシリアルシフタがビットレートまたはFoscで動作しているのに反して、データ回復ブロックは現実にはx16回でボーレートを動作している高スピードのシフタです。

非同期モードが選ばれると、受信はビットCREN(RCSTA<4>)をセットすることによりイネーブルになります。

受信機の中心は受信(シリアル)シフトレジスタです(RSR)。STOPビットをサンプリングした後、RSRに受信されたデータはRCREGレジスタに転送されます(空の場合)。転送が終了すると、フラグビットRCIF(PIR1<5>)がセットされます。現実の割込みはイネーブルビットRCIE(PIE1<5>)をセット/クリアすることによりイネーブルまたはディセーブルが可能です。フラグビットRCIFはリードのみのビットで、ハードウェアによりリセットされます。RCREGレジスタが読み込まれた時と空の時はクリアされます。RCREGはダブルバッファレジスタで、2個の深いFIFOです。2個のバイトのデータがRCREG FIFOに受信・転送されることや、3番目のバイトがRSRレジ

スタにシフティングを始めることは可能です。3番目のバイトのSTOPビットの検出では、RCREGレジスタがまだいっぱいの場合、オーバーランエラービットOERR(RCSTA<1>)がセットされます。RSRのワードは失われます。RCREGレジスタはFIFOの2個のバイトを取り出すために2度読み込むことができます。オーバーランビットOERRはソフトウェアでクリアする必要があります。これは受信の論理をリセットすることにより行われます(CRENをクリアしてからセット)。ビットOERRがセットされる場合、RSRレジスタからRCREGレジスタへの転送は禁止されていますので、もしセットする場合はエラービットFERR(RCSTA<2>)のフレーミングは、ストップビットがクリアとして検出されるとセットされます。ビットFERRと9番目の受信ビットは受信データと同じ方法でバッファされます。RCREGのリードは、新しい値でビットRX9DとFERRをロードします。したがって古いIFERRとRX9Dの情報を失わないために、RCREGレジスタを読み込む前にRCSTAレジスタを読み込むことが重要です。

図12-9 : USART受信ブロック図

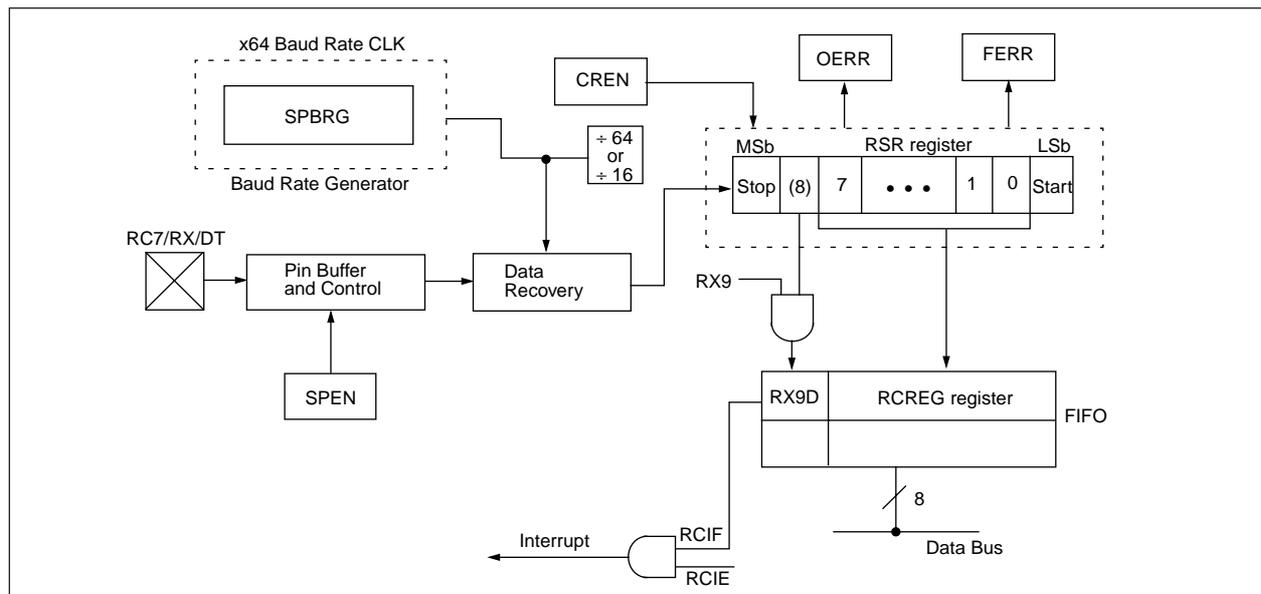
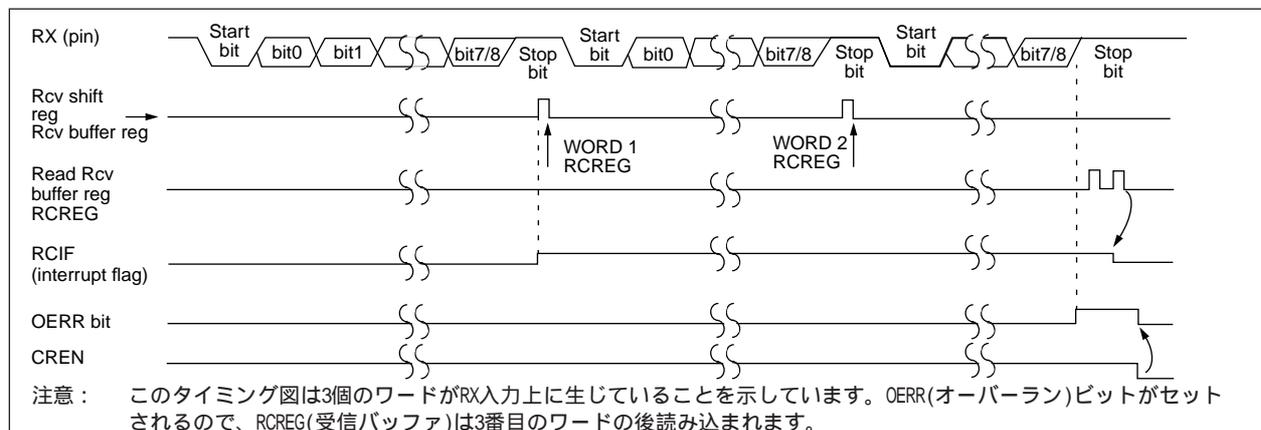


図12-10 : 非同期受信



# 暫定版

英語最新版データシートと併用しご利用下さい。

AC/DC仕様は最新英文データシートをご参照下さい。

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

非同期の受信をセットする場合は次のように進め  
ず。

1. 適当なボーレートにSPBRGレジスタを初期化する。  
高スピードのボーレートが必要な場合はビットBRGH  
をセットする(12.1章参照)。
2. ビットSYNCをクリアし、ビットSPENをセットするこ  
とにより非同期のシリアルポートをイネーブルに  
する。
3. 割込みが必要な場合、イネーブルビットRCIEをセッ  
トする。
4. 9ビットの受信が必要な場合は、ビットRX9をセッ  
トする。
5. ビットCRENをセットすることにより受信をイネーブ  
ルにする。

6. フラグビットRCIFは受信が完了した時にセットさ  
れ、割込みはイネーブルビットRCIEがセットされ  
ると発生する。
7. 9番目のビットを取るためにはRCSTAレジスタを読  
み込み(イネーブルの場合)、受信中にエラーが起  
こったかどうかを確認する。
8. RCREGレジスタを読み込むことにより8ビットの受  
信データを読み込む。
9. エラーが起こった場合は、イネーブルビットCREN  
をクリアすることによりエラーをクリアする。

表12-7: 非同期の受信に関連するレジスタ

アドレス	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR BOR での値	その他の リセット での値
0Ch	PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	-	FERR	OERR	RX9D	0000 -00x	0000 -00x
1Ah	RCREG	RX7	RX6	RX5	RX4	RX3	RX2	RX1	RX0	0000 0000	0000 0000
8Ch	PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	-	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	ボーレート・ジェネレータ・レジスタ								0000 0000	0000 0000

凡例: x=未知、-=未使用のロケーション、'0'としてリード。網掛け部分は非同期受信には使われません。

注 1: PSPIEとPSPIFビットはPIC16C73/73Aではリザーブされており、常にこれらのビットはクリアされた状態です

### 12.3 USART同期マスタモード

適用デバイス							
710	71	711	72	73	73A	74	74A

マスタ同期モードでは、データは半二重の方法で送信されます。すなわち送信と受信は同時には起こりません。データを送信している時は受信は禁止され、逆もまた同じです。ビットSYNC(TXSTA<4>)をセットすることにより同期モードになります。さらにイネーブルビットSPEN(RCSTA<7>)は、RC6/TX/CKとRC7/RX/DT I/OピンをそれぞれCK(クロック)とDT(データ)ラインに設定するためにセットされます。マスタモードはプロセッサがCKライン上でマスタクロックを送信することを示します。ビットCSRC(TXSTA<7>)をセットすることによりマスタモードになります。

#### 12.3.1 USART同期マスタ送信

USART送信機のブロック図を図12-6に示します。送信機を中心は、送信(シリアル)シフトレジスタです(TSR)。シフトレジスタはリード/ライト送信バッファレジスタTXREGからのデータを得ます。TXREGレジスタはソフトウェアでのデータにロードします。TSRレジスタは最後のビットがその前の負荷から送信されるまでロードされません。最後のビットが送信されるとすぐ、TSRはTXREG(有効なら)から新しいデータにロードします。TXREGレジスタがそのデータをTSRレジスタに転送すると(1つのTcycleの中で起こる)、TXREGは空になり、割り込みビットTXIF(PIR1<4>)がセットされます。この割り込みはイネーブルビットTXIE(PIE1<4>)をセット/クリアすることによりイネーブルまたはディセーブルできます。フラグビットTXIFはイネーブルビットTXIEの状態を無視してセットされ、ソフトウェアではクリアできません。新しいデータがTXREGレジスタにロードされた時にだけリセットします。フラグビットTXIFがTXREGレジスタのステータスを示している間に、もう1つのビットTRMT(TXSTA<1>)がTSRレジスタのステータスを示します。TRMTはTSRが空の時セットされるリードのみのビットです。割り込みなしの論理はこのビットにつながるため、TSRレジスタが空かどうかを確認するためにこのビットを登録しなければなりません。TSRはユーザに用ではないので、データメモリには表われません。

送信はイネーブルビットTXEN(TXSTA<5>)をセットすることによりイネーブルになります。現実の送信はTXREGレジスタがデータにロードされるまで起こりません。最初のデータビットはCKライン上のクロックの次に有効な立ち上がりエッジにシフトアウトします。データ出力は、同期クロックの立ち下がりエッジのあたりでは安定しています(図12-11参照)。送信は最初にTXREGレジスタをロードし、ビットTXENをセットすることにより始動させることもできます。これは、遅いボーレートが選ばれた時に便利で、ビットTXEN、CREN、SRENがクリアされた時、BRGがリセット状態を続けるためです。イネーブルビットTXENをセットすると、ただちにシフトクロックを作りながらBRGを始動します。一般的には、最初に送信を始動させるとTSRレジスタが空になり、TXREGレジスタへの転送が結果的に、空になったTXREGであるTSRにただちに転送することになります。連続した転送は可能です。

送信中にイネーブルビットTXENをクリアすることにより送信は打ち切れ、送信機をリセットします。DTとCKピンは高インピーダンスに戻ります。送信中にビットCRENかビットSRENのどちらかがセットされると、送信は打ち切れ、DTピンが高インピーダンス状態に戻ります(受信として)。ビットCSRCがセットされると(内部クロック)、CKピンは出力のままです。しかし送信機の論理は、ピンから分離するけれどもリセットされません。送信機をリセットするためには、ビットTXENをクリアする必要があります。ビットSRENがセットされると(継続中のシングルワードの送信と受信に割り込むために)、シングルワードの後に受信され、ビットSRENはクリアされます。またシリアルポートは、ビットTXENがまだセットされているので送信状態に戻ります。DTラインはただちに高インピーダンスの受信モードから送信に切り替わり、駆動を開始します。

9ビットの送信を選ぶためには、TX9(TXSTA<6>)ビットをセットし、9番目のビットをビットTX9D(TXSTA<0>)に書き込まなければなりません。9番目のビットは、8ビットのデータをTXREGレジスタに書き込む前に、書き込む必要があります。これはTXREGに書き込むデータが結果的に、TSRレジスタへのデータにただちに転送することが可能だからです(TSRが空の場合)。TSRが空で、TXREGが“新しい”TX9Dを書き込む前に書き込まれる場合は、ビットTX9Dの“現在の”値はロードされます。

同期のマスタ送信をセットする場合は次のように進めます。

1. 適当なボーレートにSPBRGレジスタを初期化する。(12.1章参照)。
2. ビットSYNC、SPEN、CSRCをセットすることにより同期のマスタシリアルポートをイネーブルにする。
3. 割り込みが必要な場合は、イネーブルビットTXIEをセットする。
4. 9ビットの送信が必要な場合は、ビットTX9をセットする。
5. ビットTXENをセットすることにより送信をイネーブルにする。
6. 9ビットの送信を選んだ場合は、9番目のビットはビットTX9Dにロードする必要がある。
7. TXREGレジスタにデータをロードすることにより送信を開始する。

# 暫定版

英語最新版データシートと併用しご利用下さい。

AC/DC仕様は最新英文データシートをご参照下さい。

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

表12-8： 同期のマススタ送信に関連するレジスタ

アドレス	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR BOR での値	他の リセット での値
0Ch	PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	-	FERR	OERR	RX9D	0000 -00x	0000 -00x
19h	TXREG	TX7	TX6	TX5	TX4	TX3	TX2	TX1	TX0	0000 0000	0000 0000
8Ch	PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	-	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	ポー・レート・ジェネレータ・レジスタ								0000 0000	0000 0000

凡例： x=未知、-=未使用、'0'としてリード。網掛け部分は同期マススタ送信には使われません。

注1： PSPIEとPSPIFビットはPIC16C73/73Aではリザーブされており、常にこれらのビットはクリアされた状態です。

図12-11： 同期送信

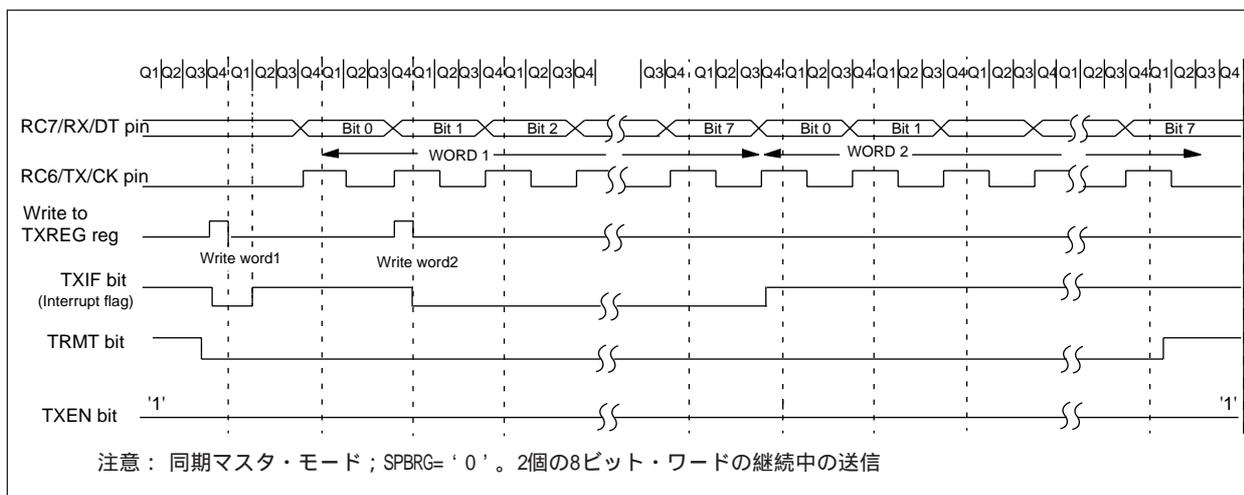
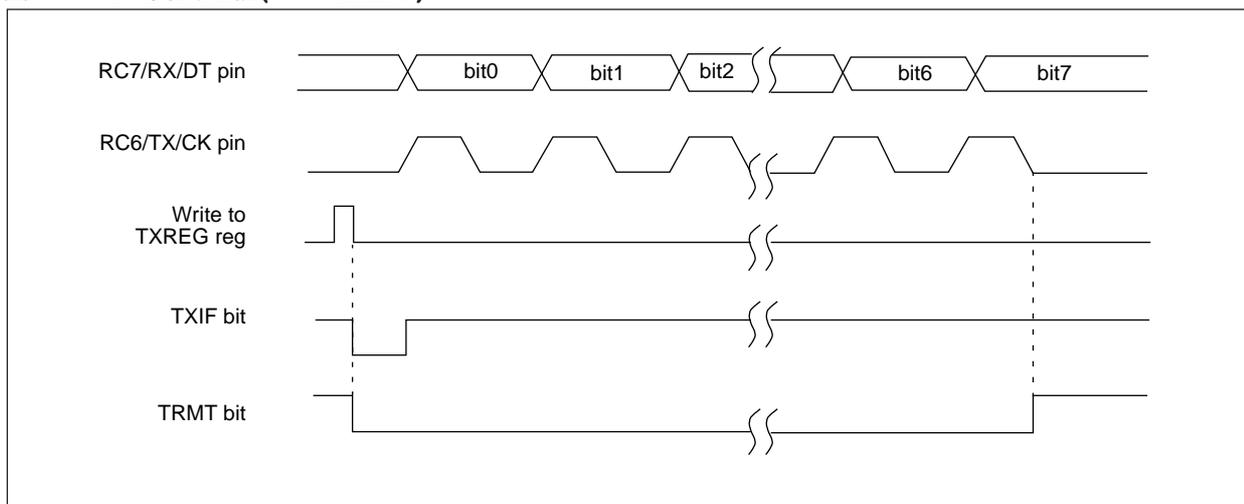


図12-12： 同期送信(TXENを通る)



### 12.3.2 USARTの同期マスタ受信

同期モードが選ばれると、受信はイネーブルビットのSREN(RCSTA<5>)またはCREN(RCSTA<4>)のどちらかをセットすることによりイネーブルになります。データはクロックの立ち下がりエッジにてRC7/RX/DTピンにてサンプリングされます。イネーブルビットSRENがセットされると、シングルワードのみが受信されます。イネーブルビットCRENがセットされると、受信はCRENがクリアされるまで続きます。両方のビットがセットされる場合は、CRENが優先します。最後のビットをクロックした後、受信シフトレジスタ(RSR)に受信されたデータはRCREGレジスタに転送されます(空の場合)。転送が終了すると、割込みフラグビットRCIF(PIR1<5>)がセットされます。現実の割込みはイネーブルビットRCIE(PIE1<5>)をセット/クリアすることによりイネーブルまたはディセーブルが可能です。フラグビットRCIFはリードオンリーのビットで、ハードウェアによりリセットされます。この場合ではRCREGレジスタが読み込まれた時と空の時にリセットされます。RCREGはダブルバッファレジスタで、2個の深いFIFOです。2個のバイトのデータがRSR FIFOに受信・転送されることや、3番目のバイトがRSRレジスタにシフティングを始めることは可能です。3番目のバイトの最後のビットのクロッキングでは、RCREGレジスタがまだいっぱいの場合は、オーバーランエラービットOERR(RCSTA<1>)がセットされます。RSRのワードは失われます。RCREGレジスタはFIFOの2個のバイトを取り出すために2度読み込むことができます。ビットOERRはソフトウェアでクリアする必要があります(ビットCRENをクリアすることにより)。ビットOERRがセットされている場合、RSRからRCREGへの転送は禁止されていますので、もしセットする場合はビットOERR

をクリアすることが重要です。9番目の受信ビットは受信データと同じ方法でバッファされます。RCREGレジスタのリードは、新しい値でビットRX9Dをロードします。したがって古いRX9Dの情報を失わないために、RCREGを読み込む前にRCSTAレジスタを読み込むことが重要です。

同期のマスタ受信をセットする場合は次のように進めます。

1. 適当なボーレートにSPBRGレジスタを初期化する。(12.1章参照)。
2. ビットSYNC、SPEN、CSRCをセットすることにより同期のマスタシリアルポートをイネーブルにする。
3. ビットCRENとSRENのクリアを確実にする。
4. 割込みが必要な場合、イネーブルビットRCIEをセットする。
5. 9ビットの受信が必要な場合は、ビットRX9をセットする。
6. 単独受信が必要な場合は、ビットSRENをセットする。継続受信の場合はビットCRENをセットする。
7. 割込みフラグビットRCIFは受信が完了した時にセットされ、割込みはイネーブルビットRCIEがセットされると発生する。
8. 9番目のビットを取るためにはRCSTAレジスタを読み込む(イネーブルの場合)。受信中にエラーが起こったかどうかを確かめる。
9. RCREGレジスタを読み込むことにより8ビットの受信データを読み込む。
10. エラーが起こった場合は、ビットCRENをクリアすることによりエラーをクリアする。

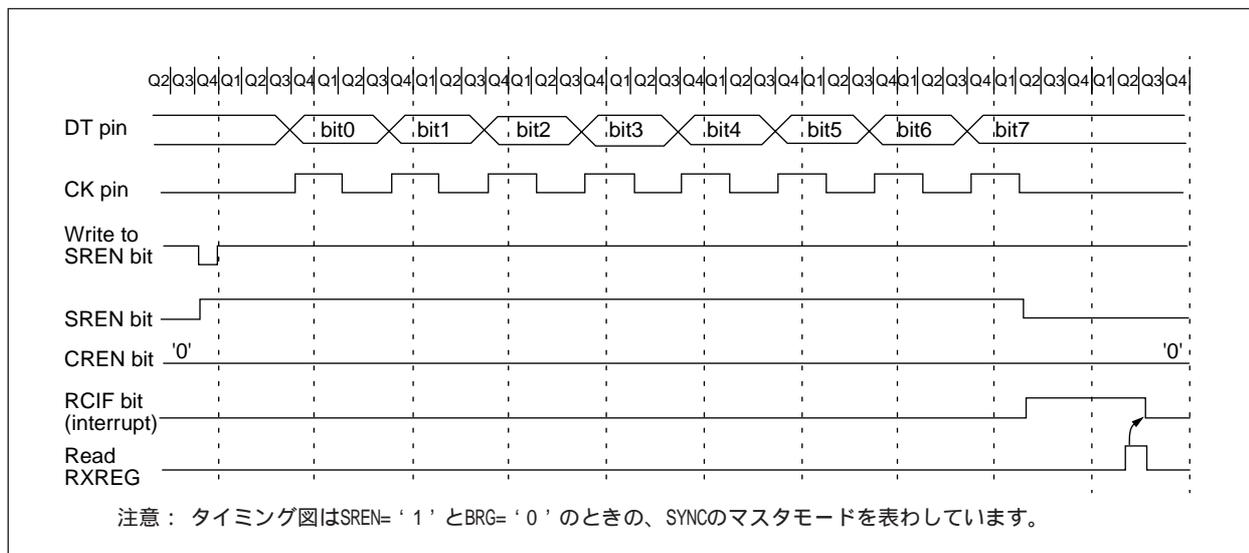
表12-9: 同期のマスタ受信に関連するレジスタ

アドレス	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR BORでの値	その他のリセットでの値
0Ch	PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	-	FERR	OERR	RX9D	0000 -00x	0000 -00x
1Ah	RCREG	RX7	RX6	RX5	RX4	RX3	RX2	RX1	RX0	0000 0000	0000 0000
8Ch	PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	-	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	ボーレートジェネレータレジスタ								0000 0000	0000 0000

凡例: x=x=未知、- =未使用、'0'としてリード。網掛け部分は同期マスタ受信には使われません。

注 1: PSPIEとPSPIFビットはPIC16C73/73Aではリザーブされ、常にこれらのビットはクリアされた状態です。

図12-13： 同期受信(マスターモード, SREN)



### 12.4 USART 同期スレーブモード

適用デバイス							
710	71	711	72	73	73A	74	74A

同期スレーブモードは、外部からシフトクロックがRC6/TX/CKピンに与えられているという点でマスタモードとは違います(マスタモードでは内部から供給されます)。これによりデバイスはSLEEPモード中にデータを転送または受信することができます。ビットC S R C (TXSTA<7>)をクリアすることによりスレーブモードになります。

#### 12.4.1 USART同期スレーブ送信

同期マスタとスレーブモードの操作は、SLEEPモードの場合を除いて全く同じです。

2個のワードをTXREGに書き込み、SLEEP命令を実行すると、次のことが起こります。

- 最初のワードはただちにTSRレジスタに転送され送信されます。
- 2番目のワードはTXREGレジスタに残ります。
- フラグ・ビットTXIFはセットされません。
- 最初のワードがTSRからシフトアウトされた時には、TXREGレジスタは2番目のワードをTSRに転送し、フラグビットTXIFがセットされます。
- イネーブルビットTXIEがセットされていると、割り込みがSLEEPからチップをウェークし、全体的な割り込みがイネーブルされるとプログラムは割り込みベクトル(0004h)に分岐します。

同期のスレーブ送信をセットする場合は次のように進めます。

- ビットSYNCとSPENをセットし、ビットCSRCをクリアすることにより同期のスレーブシリアルポートをイネーブルにする。
- ビットCRENとSRENをクリアする。
- 割り込みが必要な場合は、イネーブルビットTXIEをセットする。
- 9ビットの送信が必要な場合は、ビットTX9をセットする。
- イネーブルビットTXENをセットすることにより送信をイネーブルにする。
- 9ビットの送信を選んだ場合は、9番目のビットはビットTX9Dにロードする必要がある。
- TXREGレジスタにデータをロードすることにより送信を開始する。

#### 12.4.2 USART同期スレーブ受信

同期マスタとスレーブモードの操作は、SLEEPモードの場合を除いて全く同じです。さらにビットSRENもスレーブモードでは関係ありません。

SLEEP命令より前に、ビットCRENをセットして受信をイネーブルすると、1個のワードがSLEEP中に受信されることがあります。完全にそのワードを受信することで、RSRレジスタはそのデータをRCREGレジスタに転送します。そしてイネーブルビットRCIEがセットされると、発生した割り込みがSLEEPからチップをウェークします。全割り込みGIEがイネーブルされると、プログラムは割り込みベクトル(0004h)に分岐します。

同期のスレーブ受信をセットする場合は次のように進めます。

- ビットSYNCとSPENをセットし、ビットCSRCをクリアすることにより同期のマスタシリアルポートをイネーブルにする。
- 割り込みが必要な場合は、イネーブルビットRCIEをセットする。
- 9ビットの受信が必要な場合は、ビットRX9をセットする。
- 受信をイネーブルするためには、イネーブルビットCRENをセットする
- フラグビットRCIFは受信が完了した時にセットされます。イネーブルビットRCIEがセットされると割り込みを発生する。
- 9番目のビットを取るためにはRCSTAレジスタを読み込み(イネーブルの場合)、受信中にエラーが起こったかどうかを確かめる。
- RCREGレジスタを読み込むことにより8ビットの受信データを読み込む。
- エラーが起こった場合は、ビットCRENをクリアすることによりエラーをクリアする。

# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。

## PIC16C7X

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

表12-10： 同期スレーブ送信に関連するレジスタ

アドレス	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR BOR での値	他の リセット での値
0Ch	PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	-	FERR	OERR	RX9D	0000 -00x	0000 -00x
19h	TXREG	TX7	TX6	TX5	TX4	TX3	TX2	TX1	TX0	0000 0000	0000 0000
8Ch	PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	-	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	ボー・レート・ジェネレータ・レジスタ								0000 0000	0000 0000

凡例： x=未知、 - =未使用、 '0'としてリード。網掛け部分は同期スレーブ送信には使われません。

注 1： PSPIEとPSPIFビットはPIC16C73/73Aではリザーブされており、常にこれらのビットはクリアされた状態です。

表12-11： 同期スレーブ受信に関連するレジスタ

アドレス	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR BOR での値	その他の リセット での値
0Ch	PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	-	FERR	OERR	RX9D	0000 -00x	0000 -00x
1Ah	RCREG	RX7	RX6	RX5	RX4	RX3	RX2	RX1	RX0	0000 0000	0000 0000
8Ch	PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	-	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	ボーレートジェネレータレジスタ								0000 0000	0000 0000

凡例： x=未知、 - =未使用、 '0'としてリード。網掛け部分は同期スレーブ受信には使われません。

注 1： PSPIEとPSPIFビットはPIC16C73/73Aではリザーブされており、常にこれらのビットはクリアされた状態です。

### 13.0 アナログ-デジタル変換器(A/D)モジュール

適用デバイス							
710	71	711	72	73	73A	74	74A

アナログ-デジタル(A/D)変換器モジュールは、PIC16C710/71/711には4個のアナログ入力、PIC16C72/73/73Aには5個、PIC16C74/74Aには8個あります。

A/Dは、アナログ入力信号を対応する8ビットのデジタルに変換することができます(A/D変換器の使い方についてはアプリケーションノートAN546を参照)。サンプルホールドの出力は変換器への入力であり、それは漸近法によって数値を発生します。アナログの基準電圧は、デバイスの正電源電圧(V<sub>DD</sub>)かRA3/AN3/V<sub>REF</sub>ピンの電圧レベルのどちらかのソフトウェアが選択可能です。A/D変換器にはデバイスがSLEEPモードの間に動作できるという独特の特徴があります。

A/Dモジュールには次のような3個のレジスタがあります。

- A/D結果レジスタ(ADRES)
- A/D制御レジスタ0(ADCON0)
- A/D制御レジスタ1(ADCON1)

図13-1と13-2に示すように、ADCON0レジスタはA/Dモジュールの動作を制御します。ADCON1レジスタは、図13-3と13-4に示すように、ポートピンの機能を設定します。ポートピンはアナログ入力(RA3は基準電圧となることも可能)、またはデジタルI/Oとして設定することもできます。

図13-1: ADCON0レジスタ PIC16C710/71/711(アドレス08h)

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
ADCS1	ADCS0	( <sup>1</sup> )	CHS1	CHS0	GO/DONE	ADIF	ADON	
bit7								bit0
bit 7-6: <b>ADCS1:ADCS0</b> : A/D変換クロック選択ビット 00 = Fosc/2 01 = Fosc/8 10 = Fosc/32 11=Frc (RC発振から出たクロック)								
bit 5: <b>未使用</b> : '0'としてリード。								
bit 4-3: <b>CHS2:CHS0</b> : アナログ・チャンネル選択ビット 00=チャンネル0, (RA0/AN0) 01=チャンネル1, (RA1/AN1) 10=チャンネル2, (RA2/AN2) 11=チャンネル3, (RA3/AN3)								
bit 2: <b>GO/DONE</b> : A/D変換ステータスビット ADON=1の場合 1=A/D変換進行中 (このビットをセットすると A/D変換が始動)。 0=A/D変換は進行していない(A/D変換が完了すると、このビットは自動的にハードウェアによりクリアされる)。								
bit 1: <b>ADIF</b> : A/D変換完了割り込みフラグビット 1=変換は完了(ソフトウェアでのクリアが必要)。 0=変換は完了していない。								
bit 0: <b>ADON</b> : A/Dオンビット 1=A/Dコンバータモジュールが動作中。 0=A/Dコンバータモジュールがシャットオフされ動作中の電流を消費しない。								
注1: ADCON0のビット5はPIC16C71だけの汎用R/Wビット。PIC16C710/711についてはこのビットは未使用で、'0'としてリード。								

R = 読み込み可能なビット  
 W = 書き込み可能なビット  
 U = 未使用のビット、  
 '0'としてリード  
 - n = PORリセットでの値

# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。  
<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

図13-2 : ADCON0レジスタ PIC16C72/73/73A/74/74A(アドレス1Fh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	-	ADON
bit7						bit0	
bit7-6: <b>ADCS1:ADCS0</b> : A/D変換クロック選択ビット 00=Fosc/2 01=Fosc/8 10=Fosc/32 11=FRC (RC発振から出たクロック)							
bit5-3: <b>CHS2:CHS0</b> : アナログチャンネル選択ビット 000=チャンネル0, (RA0/AN0) 001=チャンネル1, (RA1/AN1) 010=チャンネル2, (RA2/AN2) 011=チャンネル3, (RA3/AN3) 100=チャンネル4, (RA5/AN4) 101=チャンネル5, (RE0/AN5) (1) 110=チャンネル6, (RE1/AN6) (1) 111=チャンネル7, (RE2/AN7) (1)							
bit2: <b>GO/DONE</b> : A/D変換ステータスビット ADON=1の場合 1=A/D変換進行中 (このビットをセットすると A/D変換が始動)。 0=A/D変換は進行していない(A/D変換が完了すると、このビットは自動的にハードウェアによりクリアされる)。							
bit1: <b>未使用</b> : '0' としてリード							
bit0: <b>ADON</b> : A/Dオンビット 1=A/Dコンバータ・モジュールが動作中。 0=A/Dコンバータ・モジュールがシャットオフされ動作中の電流を消費しない。							

R = 読み込み可能なビット  
W = 書き込み可能なビット  
U = 未使用のビット、  
'0' としてリード  
- n = PORリセットでの値

注1: A/Dチャンネル5、6、7はPIC16C74/74Aのみ。

図13-3 : ADCON1レジスタ PIC16C710/71/711(アドレス88h)

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
-	-	-	-	-	-	PCFG1	PCFG0
bit7						bit0	
bit 7-2: 未使用: '0' としてリード							
bit 1-0: <b>PCFG1:PCFG0</b> : A/Dポート設定制御ビット							
<b>PCFG1:PCFG0</b>		<b>RA1 &amp; RA0</b>	<b>RA2</b>	<b>RA3</b>	<b>VREF</b>		
00		A	A	A	VDD		
01		A	A	VREF	RA3		
10		A	D	D	VDD		
11		D	D	D	VDD		

R = 読み込み可能なビット  
W = 書き込み可能なビット  
U = 未使用のビット、  
'0' としてリード  
- n = PORリセットでの値

A = アナログ入力  
D = デジタルI/O

# 暫定版

英語最新版データシートと併用しご利用下さい。  
 AC/DC仕様は最新英文データシートをご参照下さい。  
<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

図13-4: ADCON1レジスタ PIC16C72/73/73A/74/74A(アドレス9Fh)

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0			
-	-	-	-	-	PCFG2	PCFG1	PCFG0			
bit7					bit0					

R = 読み込み可能なビット  
 W = 書き込み可能なビット  
 U = 未使用のビット、  
 '0'としてリード  
 - n = PORリセットでの値

bit 7-3: 未使用: '0'としてリード  
 bit 2-0: **PCFG2:PCFG0**: A/Dポート設定制御ビット

PCFG2:PCFG0	RA0	RA1	RA2	RA5	RA3	RE0	RE1	RE2	VREF
000	A	A	A	A	A	A	A	A	VDD
001	A	A	A	A	VREF	A	A	A	RA3
010	A	A	A	A	A	D	D	D	VDD
011	A	A	A	A	VREF	D	D	D	RA3
100	A	A	D	D	A	D	D	D	VDD
101	A	A	D	D	VREF	D	D	D	RA3
11x	D	D	D	D	D	D	D	D	-

A = アナログ入力  
 D = デジタルI/O

注1: RE0, RE1, RE2は、PIC16C74/74Aのみ。

ADRESレジスタはA/D変換の結果を含んでいます。A/D変換が完了すると、結果はADRESレジスタにロードされ、GO/DONEビット(ADCON0<2>)がクリアされ、A/D割込みフラグビットADIFがセットされます。A/Dモジュールのブロック図を図13-5と13-6に示します。

A/Dモジュールが必要により設定された後、選択されたチャンネルは変換が始まる前にサンプリングする必要があります。アナログ入力チャンネルは、対応するTRISビットを入力として選択する必要があります。サンプル時間を決めるためには、13-1章を参照してください。このサンプル時間が経過した後、A/D変換を始動することができます。A/D変換は次のように進めます。

1. A/Dモジュールを設定する。

- ・アナログピン/電圧基準/ディジタルI/O(ADCON1)を設定
- ・A/D入力チャンネルを選択(ADCON0)
- ・A/D変換クロックを選択(ADCON0)
- ・A/Dモジュールをオン(ADCON0)

2. A/D割込みを設定する(必要な場合)。

- ・ADIFビットをクリア
- ・ADIEビットをセット
- ・GIEビットをセット

3. 必要とされるサンプリング時間を待つ。

4. 変換を始動する。

- ・GO/DONEビットをセット(ADCON0)

5. 下記のいずれかによりA/D変換が完了するのを待つ。

- ・GO/DONEビットがクリアされるまでポーリング

または

- ・A/D割込みを待つ

6. A/D結果レジスタ(ADRES)を読み込み、必要ならビットADIFをクリアする。

7. 次の変換のために、ステップ1または2に戻る必要がある。ビットごとのA/D変換時間は $T_{AD}$ として定義される。 $2T_{AD}$ の最少のウェイト時間が、次のサンプリングが始まる前に要求されます。

図13-5: A/Dブロック図 PIC16C710/71/711

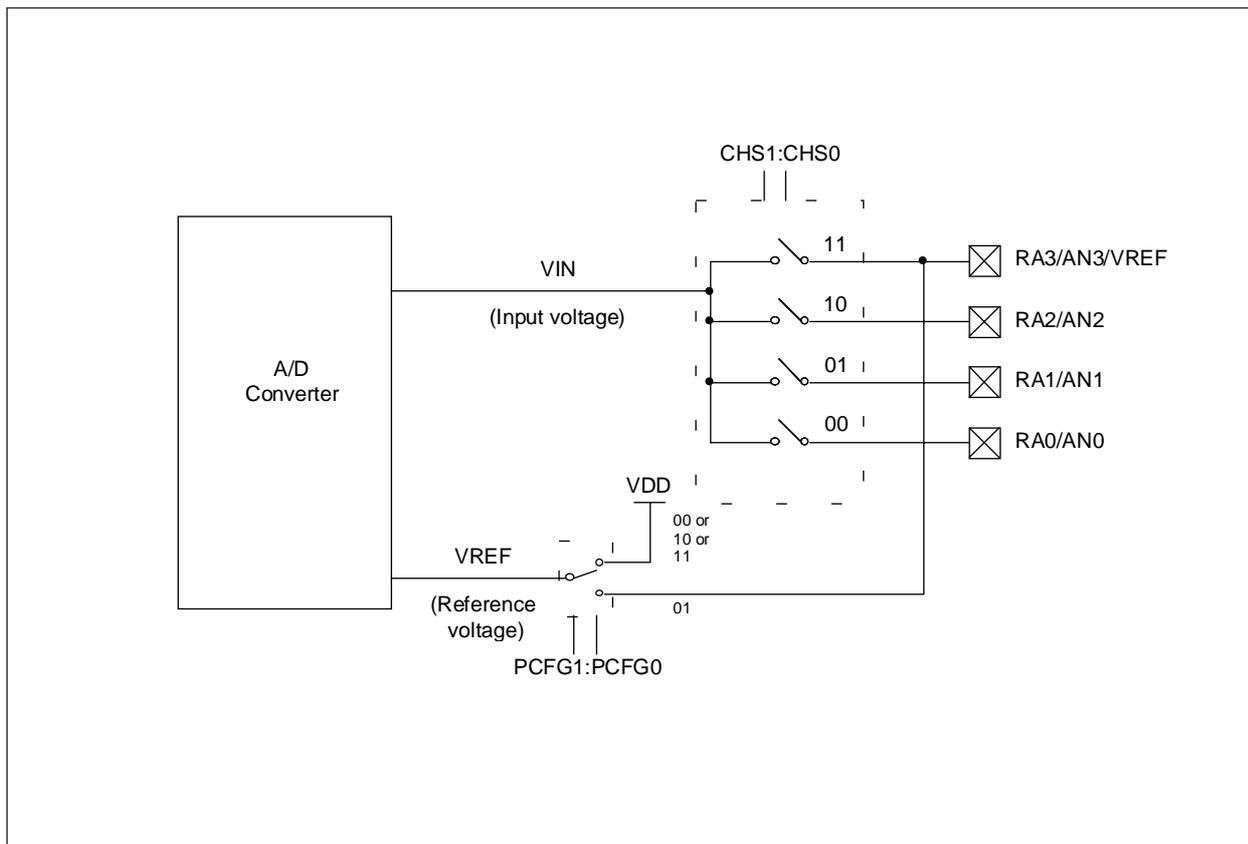
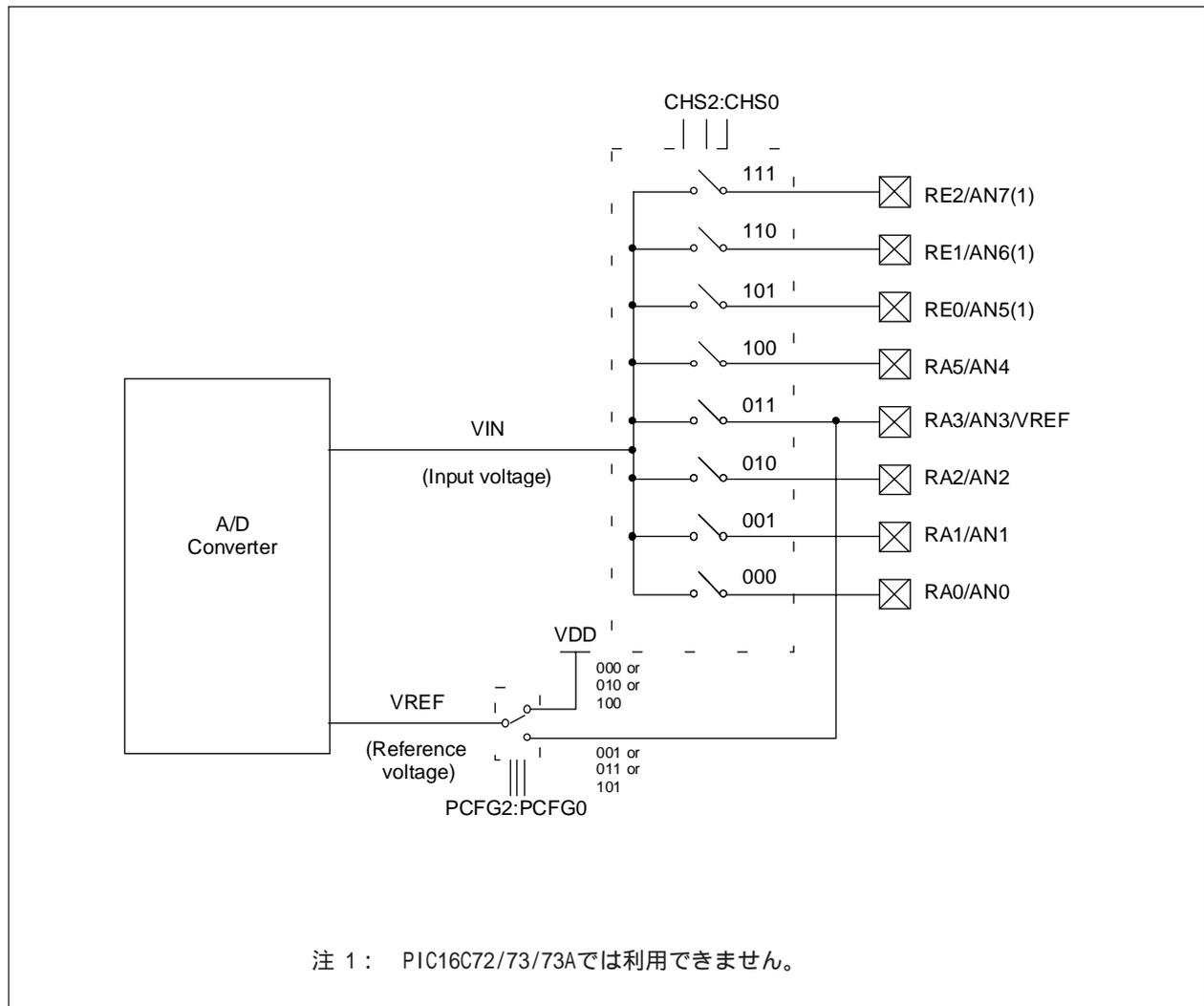


図13-6: A/Dブロック図 PIC16C72/73/73A/74/74A



13.1 A/Dサンプリングリクエスト

適用デバイス							
710	71	711	72	73	73A	74	74A

A/D変換器の設定された精度を満たすためには、チャージ保持キャパシタ(CHOLD)により入力チャンネル電圧レベルに完全に充電することができなければなりません。図13-7にアナログ入力モデルを示します。電源インピーダンス(Rs)と内部サンプリングスイッチインピーダンス(Rss)は直接キャパシタCHOLDを充電するために必要な時間に影響します。図13-7に示すように、サンプリングスイッチインピーダンス(Rss)はデバイス電圧(VDD)で変わります。**アナログ電源のインピーダンスは10kΩ以内をお勧めします。**アナログ入力チャンネルを選択した(変化した)後、このサンプリングは変換が始動する前に行われなければなりません。

最小のサンプリング時間を計算するためには、方程式13-1を使用します。この方程式は1/2LSbエラーが使われると仮定しています(A/Dに関して512ステップ)。1/2LSbエラーは、設定された分解能を満たすためにA/Dに許されている最大のエラーです。

方程式 13-1: A/Dの最小充電時間

$$V_{hold} = (V_{ref} - (V_{ref}/512)) \times (1 - e^{-(T_c/C_{hold})(R_{ic} + R_{ss} + R_s)})$$

または

$$T_c = -(51.2 \text{ pF})(1 \text{ k}\Omega + R_{ss} + R_s) \ln(1/511)$$

例13-1は必要とされる最小のサンプル時間T<sub>ACQ</sub>の計算を示しています。この計算は次のようなシステムの仮定に基づいています。

- Rs = 10 kΩ
- 1/2 LSb error
- Vdd = 5V → Rss = 7 kΩ
- Temp (system max.) = 50°C
- Vhold = 0 @ t = 0

注意1: 基準電圧(VREF)はそれ自体の出力をキャンセルするので、方程式に影響を与えません。

注意2: チャージ保持キャパシタ(CHOLD)はそれぞれの変換後放電しません。

注意3: アナログ電源のインピーダンスは最大10kΩにして下さい。これはピンからの漏れ特性を考慮する上で重要です。

注意4: 変換が完了した後、2.0TAD遅延時間はサンプリングが再び始まる前にとらなければなりません。この時間中、保持しているキャパシタは選択されたA/D入力チャンネルに接続されていません。

例 13-1: 必要とされる最小サンプル時間の計算

$$T_{ACQ} = \text{Amplifier Settling Time} + \text{Holding Capacitor Charging Time} + \text{Temperature Coefficient}$$

$$T_{ACQ} = 5 \mu\text{s} + T_c + [(Temp - 25)(0.05 \text{ ms}/^\circ\text{C})]$$

$$T_c = -C_{hold} (R_{ic} + R_{ss} + R_s) \ln(1/512)$$

$$= -51.2 \text{ pF} (1 \text{ k}\Omega + 7 \text{ k}\Omega + 10 \text{ k}\Omega) \ln(0.0020)$$

$$= -51.2 \text{ pF} (18 \text{ k}\Omega) \ln(0.0020)$$

$$= -0.921 \mu\text{s} (-6.2146)$$

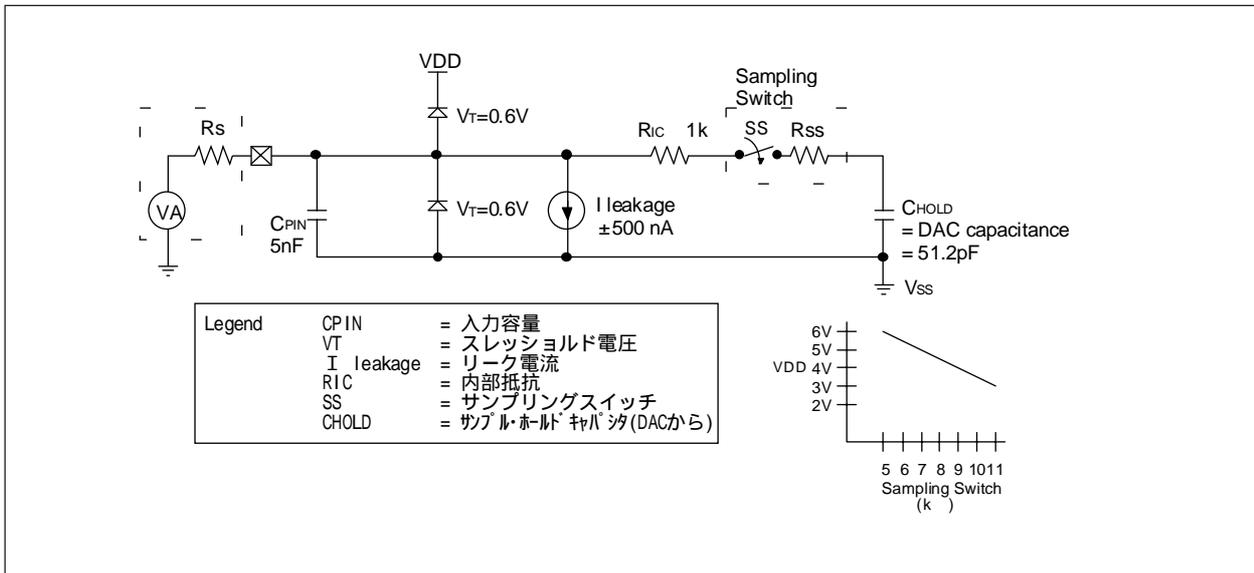
$$= 5.724 \mu\text{s}$$

$$T_{ACQ} = 5 \mu\text{s} + 5.724 \mu\text{s} + [(50 - 25)(0.05 \mu\text{s}/^\circ\text{C})]$$

$$= 10.724 \mu\text{s} + 1.25 \mu\text{s}$$

$$= 11.974 \mu\text{s}$$

図13-7: アナログ入力モデル



# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。  
<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

### 13.2 A/D 変換クロックの選択

適用デバイス							
710	71	711	72	73	73A	74	74A

ビットごとのA/D変換時間は $T_{AD}$ として定義されています。A/D変換は8ビット変換につき $9.5T_{AD}$ が必要です。A/D変換クロック・ソースはソフトウェアで選択されます。 $T_{AD}$ には次のような4個のオプションが使用できます。

- ・ 2Tosc
- ・ 8Tosc
- ・ 32Tosc
- ・ 内部RCオシレータ

正しいA/D変換のためには、次のような最小の $T_{AD}$ 時間を確実にするためにA/D変換クロック( $T_{AD}$ )を選択する必要があります。

PIC16C71には2.0 $\mu$ s :

その他のPIC16C7Xデバイスには1.6 $\mu$ s :

表13-1と13-2に、デバイスの動作中の周波数と選択されたA/Dクロックから求めた $T_{AD}$ 時間を示します。

### 13.3 アナログポートピンの設定

適用デバイス							
710	71	711	72	73	73A	74	74A

ADCON1、TRISA、TRISEレジスタはA/Dポートピンの動作を制御します。アナログ入力として要求されるポートピンは、対応するTRISビットをセット(入力)しなければなりません。TRISビットがクリア(出力)されると、デジタル出力レベル( $V_{OH}$ または $V_{OL}$ )は変換されます。

A/D動作はCHS2:CHS0ビットとTRISビットの状態から独立しています。

注意1: ポートレジスタを読み込む時、アナログ入力チャンネルとして設定されたすべてのピンは、クリアされたものとして読み込まれます(Lowレベル)。デジタル入力として設定されたピンは、アナログ入力を変換します。

注意2: デジタル入力として定義されたピン(AN7:AN0ピンを含む)のアナログレベルにより、入力バッファがデバイスの仕様をこえた電流を消費することがあります。

表13-1:  $T_{AD}$ 対デバイスの動作中の周波数 PIC16C71

AD Clock Source (TAD)		Device Frequency				
Operation	ADCS1:ADCS0	20 MHz	16 MHz	4 MHz	1 MHz	333.33 kHz
2Tosc	00	100 ns(2)	125 ns(2)	500 ns(2)	2.0 us	6 us
8Tosc	01	400 ns(2)	500 ns(2)	2.0 us	8.0 us	24 us(3)
32Tosc	10	1.6 us(2)	2.0 us	8.0 us	32.0 us(3)	96 us(3)
RC <sup>(5)</sup>	11	2 - 6 us(1,4)	2 - 6 us(1,4)	2 - 6 us(1,4)	2 - 6 us(1)	2 - 6 us(1)

- 注意
1. RC電源には4usの基準的なTAD時間があります。
  2. これらの値は必要とされる最小のTAD時間を違反します。
  3. 変換時間をより速くするためには、別のクロック・ソースを選ぶことをお勧めします。
  4. 1MHz以上のデバイス周波数で、RCモードは変換精度は規定外です。
  5. 拡張電圧仕様(LC)デバイスについては、電氣的仕様を参照。

表13-2:  $T_{AD}$ 対デバイスの動作中の周波数 PIC16C710/711/72/73/73A/74/74A

AD Clock Source (TAD)		Device Frequency			
Operation	ADCS1:ADCS0	20 MHz	5 MHz	1.25 MHz	333.33 kHz
2Tosc	00	100 ns(2)	400 ns(2)	1.6 us	6 us
8Tosc	01	400 ns(2)	1.6 us	6.4 us	24 us(3)
32Tosc	10	1.6 us	6.4 us	25.6 us(3)	96 us(3)
RC <sup>(5)</sup>	11	2 - 6 us(1,4)	2 - 6 us(1,4)	2 - 6 us(1,4)	2 - 6 us(1)

- 注意
1. RC電源には4usの基準的なTAD時間があります。
  2. これらの値は必要とされる最小のTAD時間を違反します。
  3. 変換時間をより速くするためには、別のクロック・ソースを選ぶことをお勧めします。
  4. 1MHz以上のデバイス周波数で、RCモードは変換精度は規定外です。
  5. 拡張電圧仕様(LC)デバイスについては、電氣的仕様を参照。

# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。  
<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

### 13.4 A/D変換

適用デバイス							
710	71	711	72	73	73A	74	74A

例13-2と13-3に、A/D変換の実行方法を示します。RAピンはアナログ入力として設定されます。アナログ基準 ( $V_{REF}$ ) は、デバイスポルテージ  $V_{DD}$  です。A/D割込みはイネーブルされ、A/D変換クロックは  $F_{RC}$  です。その変換は RA0チャンネル上で実行されます。

注意：  $\overline{GO/DONE}$  ビットはA/Dをオンする同じ命令でセットしてはいけません。

変換中に  $\overline{GO/DONE}$  ビットをクリアすると電流変換を打ち切ります。ADRESレジスタは部分的に完了したA/D変換のサンプルではアップデートされません。すなわちADRESレジスタは最後に完了した変換の値を持ち続けます（またはADRESレジスタに書き込まれた最後の値）。A/D変換が打ち切られた後、次のサンプリングが始動する前に  $2T_{AD}$  ウェイトが必要です。この  $2T_{AD}$  ウェイトの後、サンプリングは選ばれたチャンネルで自動的に始動します。

#### 例13-2： A/D変換の実行(PIC16C710/71/711)

```
BSF STATUS,RP0 ; Select Page 1
CLRF ADCON1 ; Configure A/D inputs
BCF STATUS,RP0 ; Select Page 0
MOVLW 0xC1 ; RC Clock, A/D is on, Channel 0 is selected
MOVWF ADCON0 ;
BSF INTCON,ADIE ; Enable A/D Interrupt
BSF INTCON,GIE ; Enable all interrupts
;
; Ensure that the required sampling time for the selected input channel has elapsed.
; Then the conversion may be started.
;
BSF ADCON0,GO ; Start A/D Conversion
: ; The ADIF bit will be set and the GO/DONE bit
: ; is cleared upon completion of the A/D Conversion.
```

#### 例13-3： A/D変換の実行(PIC16C72/73/73A/74/74A)

```
BSF STATUS,RP0 ; Select Page 1
CLRF ADCON1 ; Configure A/D inputs
BSF PIE1,ADIE ; Enable A/D interrupts
BCF STATUS,RP0 ; Select Page 0
MOVLW 0xC1 ; RC Clock, A/D is on, Channel 0 is selected
MOVWF ADCON0 ;
BCF PIR1,ADIF ; Clear A/D interrupt flag bit
BSF INTCON,PEIE ; Enable peripheral interrupts
BSF INTCON,GIE ; Enable all interrupts
;
; Ensure that the required sampling time for the selected input channel has elapsed.
; Then the conversion may be started.
;
BSF ADCON0,GO ; Start A/D Conversion
: ; The ADIF bit will be set and the GO/DONE bit
: ; is cleared upon completion of the A/D Conversion.
```

### 13.4.1 より速い変換 - 低分解能とのトレードオフ

すべての応用に8ビットの分解能の結果が必要とは限りませんが、代わりにより速い変換時間を要求することができます。A/Dモジュールは、変換スピードと分解能を交換することができます。要求された分解能に関係なくサンプリング時間は同じです。変換をスピードアップするために、A/Dモジュールのクロックソースは、 $T_{AD}$ 時間が規定された最小の時間を違反するように変更することができます(応用可能な電氣的仕様を参照)。 $T_{AD}$ 時間が規定された最小の時間を妨げると、次のようにすべてのA/D結果ビットが有効ではなくなります(電氣的仕様の章のA/D変換のタイミング参照)。クロックソースは、3個のオシレータのバージョンの間で交換されるだけです(RCからまたはRCへは交換不可能)。オシレータを交換される前に時間を決めるための方程式は次の通りです。

$$\text{変換時間} = 2T_{AD} + N \cdot T_{AD} + (8 - N)(2T_{OSC})$$

ただし:  $N$  = 要求分解能のビット数

$T_{AD}$ はデバイスオシレータに基づいているので、A/Dオシレータを変更する時いくつかの方法があります(タイマ、ソフトウェアループなど)。例13-4に4ビットの分解能の変換と、8ビットの分解能の変換に必要な時間の比較を示します。その例は20MHzと16MHzで動作しているデバイスに関するもので(A/Dクロックは $32T_{OSC}$ でプログラム)、 $6T_{AD}$ の後ただちにA/Dクロックは $2T_{OSC}$ にプログラムされると仮定しています。

$2T_{OSC}$ は最小の $T_{AD}$ 時間を違反するので、最後の4ビットは正しい値に変換しません。

#### 例13-4: 4ビット対8ビットの変換時間

	Freq. (MHz) <sup>(1)</sup>	分解能	
		4-bit	8-bit
$T_{AD}$	20	1.6 $\mu$ s	1.6 $\mu$ s
	16	2.0 $\mu$ s	2.0 $\mu$ s
$T_{OSC}$	20	50 ns	50 ns
	16	62.5 ns	62.5 ns
$2T_{AD} + N \cdot T_{AD} + (8 - N)(2T_{OSC})$	20	10 $\mu$ s	16 $\mu$ s
	16	12.5 $\mu$ s	20 $\mu$ s

注1: PIC16C71の最小 $T_{AD}$ 時間は2.0  $\mu$ sです。

その他のPIC16C7Xデバイスの最小 $T_{AD}$ 時間には1.6  $\mu$ sです。

# 暫定版

英語最新版データシートと併用しご利用下さい。

AC/DC仕様は最新英文データシートをご参照下さい。

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

### 13.5 SLEEP中のA/D動作

適用デバイス							
710	71	711	72	73	73A	74	74A

A/DモジュールはSLEEPモード中に動作することができます。これはA/DクロックソースをRC(ADCS1:ADCS0=11)にセットすることが要求されます。RCクロックソースが選ばれると、A/Dモジュールは変換を始める前に1つ命令サイクルを待ちます。これによりSLEEP命令が実行されるのを可能にし、変換からすべてのデジタルスイッチングノイズを消します。変換が完了すると、GO/DONEビットがクリアされ、その結果がADRESレジスタにロードされます。A/D割込みがイネーブルされると、デバイスはSLEEPからウェークします。A/D割込みがイネーブルされない場合、ADONビットはセットされたままですが、A/Dモジュールはオフになります。

A/Dクロックソースが別のクロックオプション(RCではなく)の時は、ADONビットはセットされたままですが、SLEEP命令により現在の変換が打ち切れ、A/Dモジュールがオフになります。

A/Dのオフは、A/Dモジュールを最も低い電流消費状態にすることで。

注意： A/DモジュールがSLEEPで動作するためには、A/DクロックソースがRC(ADCS1:ADCS0=11)にセットされる必要があります。SLEEPでA/D変換を実行するには、GO/DONEビットがセットされSLEEP命令を続けなければなりません。

### 13.6 A/Dの精度/エラー

適用デバイス							
710	71	711	72	73	73A	74	74A

A/Dの総合精度は $V_{DD}=5V \pm 10\%$ とアナログの $V_{REF}=V_{DD}$ に対して $\pm 1LSb$ 以下です。この総合精度は、オフセットエラー、フルスケールエラー、積分エラーを含みます。A/D変換器はモノトニックであることが保証されています。分解能と精度は、アナログ基準( $V_{DD}$ )が5.0V以下か、アナログ基準( $V_{REF}$ )が $V_{DD}$ 以下のどちらかの時に少なくともあります。

ピンの最大リーク電流は $\pm 5\mu A$ です。

デバイス周波数が低いシステムでは、デバイスオシレータからのA/D RCクロックの使用を選択した方が良いでしょう。高い周波数では、 $T_{AD}$ はデバイスオシレータから引き出さなければなりません。 $T_{AD}$ は最小を妨げることなく、選ばれた動作に対して $8\mu s$ である必要があります。これは、 $T_{AD}$ が $T_{OSC}$ から出ている時、チップ上のクロックの位相からずれているためです。これは広範囲にデジタルスイッチングノイズの影響を減らします。これはRCからのクロックでは不可能です。デジタルスイッチングノイズによる精度の損失は、多くのI/Oピンが動作中に大きくなります。

デバイスがA/D変換の始動後SLEEPモードに入るシステムでは、RCクロックソースの選択が必要です。このモードではSLEEP中のモジュールからのデジタルノイズは停止します。この方法により高精度となります。

### 13.7 RESETの影響

適用デバイス							
710	71	711	72	73	73A	74	74A

デバイスのリセットにより、すべてのレジスタがリセット状態になります。これによりA/Dモジュールはオフになり、どんな変換も打ち切られます。ADRESレジスタにある値はパワーオンリセットに関しては変更されません。ADRESレジスタはパワーオンリセット後の不明のデータを含みます。

### 13.8 CCPトリガの使用

適用デバイス							
710	71	711	72	73	73A	74	74A

注意： PIC16C72において、“スペシャルイベントトリガ”はCCP1モジュールに組み込まれています。

A/D変換はCCP2モジュールの“スペシャルイベントトリガ”により始動することができます(PIC16C72のみCCP1)。これはCCP2M3:CCP2M0ビット(CCP2CON<3:0>)が1011としてプログラムされ、A/Dモジュールがイネーブルされる(ADONビットをセット)ことが必要です。トリガが起るとA/D変換を始動しながらGO/DONEビットがセットされ、タイマ1カウンタはゼロにリセットされます。タイマ1はA/Dのサンプリング周期に、最小限度のソフトウェアのオーバーヘッド(ADRESから他のレジスタへのデータ移動)を含めた時間で自動的にリセットを繰り返します。“スペシャルイベントトリガ”がGO/DONEビットをセットする前に(変換を始動)、適切なアナログ入力チャンネルが選択され、最小のサンプリングが行われる必要があります。

A/Dモジュールがイネーブルされない(ADONがクリアされる)場合、“スペシャルイベントトリガ”はA/Dモジュールにより無視されますが、それでもタイマ1カウンタをリセットします。

### 13.9 接続の考察

適用デバイス							
710	71	711	72	73	73A	74	74A

入力電圧がレール値( $V_{SS}-0.2V$ または $V_{DD}+0.2V$ )を越える場合、変換の精度は保証範囲外となります。

注意： PIC16C710/71/711に関しては、OSC1ピンの隣がRA0ピンなので、A/D変換でのRA0ピンを使う時にはノイズに注意が必要です。

外部RCフィルタは時々入力信号のアンチ・エイリアシングのために付け加えられます。Rコンポーネントは、信号源インピーダンスが推奨値10k以下となるような値を選択されることを推奨します。アナログ入力ピン(ハイインピーダンス)に接続された外部のコンポーネント(キャパシタ、ツェナーダイオードなど)は、ピンのもとで漏れ電流が非常に僅かであればいけません。

### 13.10 転送機能

適用デバイス							
710	71	711	72	73	73A	74	74A

A/D変換器の理想的な伝達関数は、次の通りです。最初の変化は、アナログ入力電圧( $V_{AIN}$ )が1LSb(またはアナログ $V_{REF}/256$ )の時に起こります(図13-8参照)。

図 13-8: A/D 転送機能

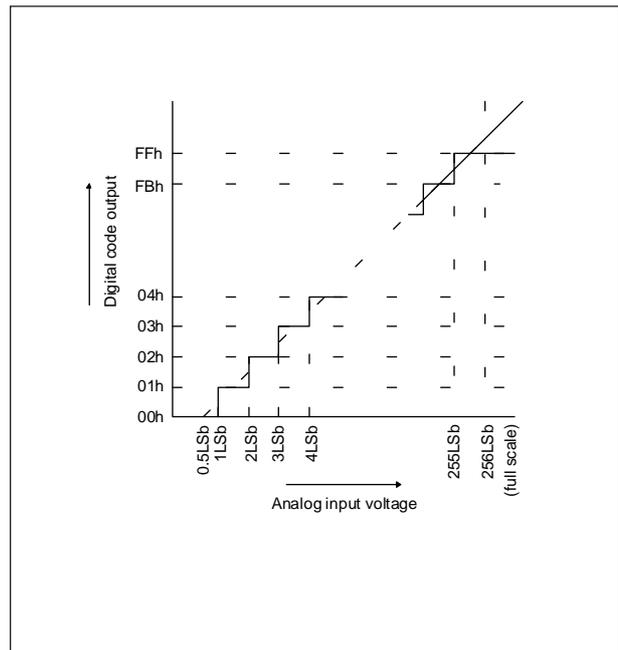
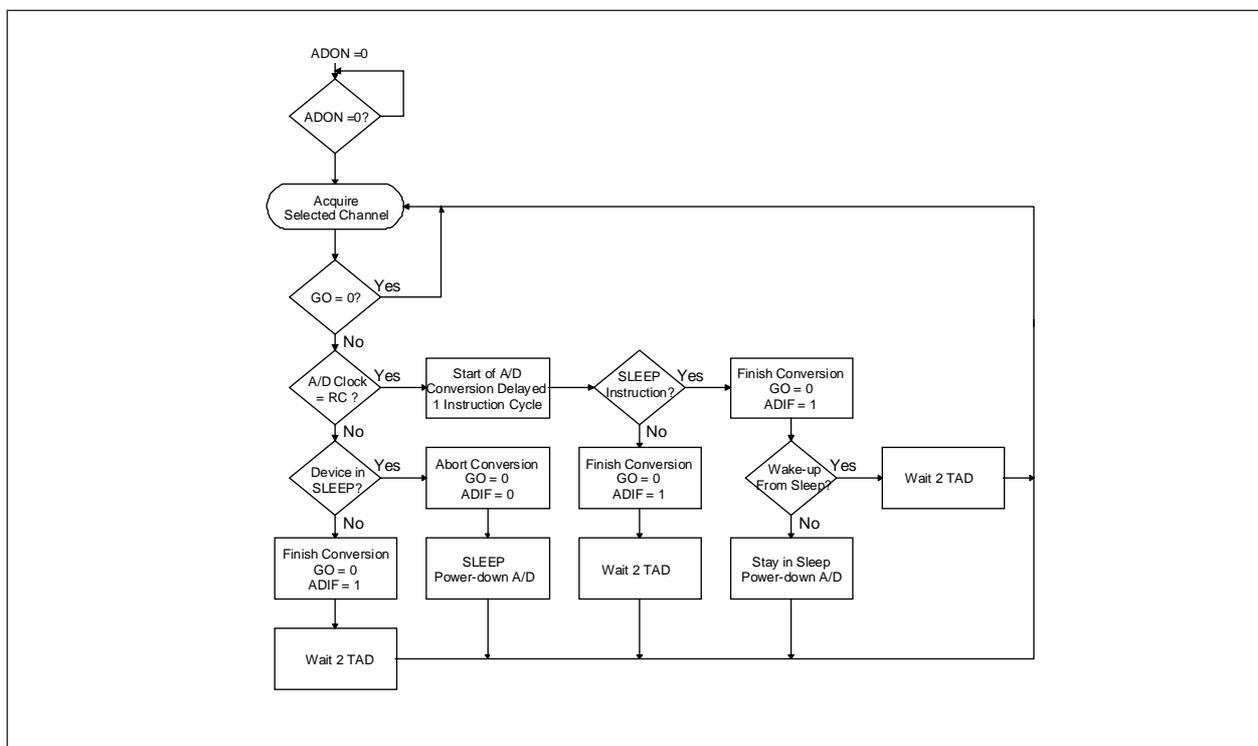


図 13-9: A/D 動作のフローチャート



# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。

## PIC16C7X

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

表13-3: A/Dレジスタのまとめ PIC16C710/71/711

アドレス	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR BOR での値	その他の リセット での値
0Bh/8Bh	INTCON	GIE	ADIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
89h	ADRES	A/D Result Register								xxxx xxxx	uuuu uuuu
08h	ADCON0	ADCS1	ADCS0	-	CHS1	CHS0	GO/DONE	ADIF	ADON	00-0 0000	00-0 0000
88h	ADCON1	-	-	-	-	-	-	PCFG1	PCFG0	---- --00	---- --00
05h	PORTA	-	-	-	RA4	RA3	RA2	RA1	RA0	---x 0000	---u 0000
85h	TRISA	-	-	-	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	---1 1111	---1 1111

凡例: x=未知、u=変化なし、- =未使用、'0'としてリード。網掛け部分はA/D変換には使われません。

表13-4: A/Dレジスタのまとめ PIC16C72

アドレス	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR BOR での値	その他の リセット での値
0Bh/8Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	-	ADIF	-	-	SSPIF	CCP1IF	TMR2IF	TMR1IF	-0-- 0000	-0-- 0000
8Ch	PIE1	-	ADIE	-	-	SSPIE	CCP1IE	TMR2IE	TMR1IE	-0-- 0000	-0-- 0000
1Eh	ADRES	A/D Result Register								xxxx xxxx	uuuu uuuu
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	-	ADON	0000 00-0	0000 00-0
9Fh	ADCON1	-	-	-	-	-	PCFG2	PCFG1	PCFG0	---- -000	---- -000
05h	PORTA	-	-	RA5	RA4	RA3	RA2	RA1	RA0	--0x 0000	--00 0000
85h	TRISA	-	-	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111

凡例: x=未知、u=不変、- =未使用、'0'としてリード。網掛け部分はA/D変換には使われません。

表13-5: A/Dレジスタのまとめ PIC16C73/73A/74/74A

アドレス	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR BOR での値	その他の リセット での値
0Bh/8Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
0Dh	PIR2	-	-	-	-	-	-	-	CCP2IF	---- ---0	---- ---0
8Dh	PIE2	-	-	-	-	-	-	-	CCP2IE	---- ---0	---- ---0
1Eh	ADRES	A/D Result Register								xxxx xxxx	uuuu uuuu
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	-	ADON	0000 00-0	0000 00-0
9Fh	ADCON1	-	-	-	-	-	PCFG2	PCFG1	PCFG0	---- -000	---- -000
05h	PORTA	-	-	RA5	RA4	RA3	RA2	RA1	RA0	--0x 0000	--00 0000
85h	TRISA	-	-	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111
09h	PORTE	-	-	-	-	-	RE2	RE1	RE0	---- -xxx	---- -uuu
89h	TRISE	IBF	OBF	IBOV	PSPMODE	-	TRISE2	TRISE1	TRISE0	0000 -111	0000 -111

凡例: x=未知、u=不変、- =未使用、'0'としてリード。網掛け部分はA/D変換には使われません。

注 1: PSPIEとPSPIFビットはPIC16C73/73Aではリザーブされており、常にこれらのビットはクリアされた状態です。



# 暫定版

英語最新版データシートと併用にご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。

## PIC16C7X

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

図14-2： PIC16C710/711用コンフィギュレーションワード

CP0	CP0	CP0	CP0	CP0	CP0	CP0	BODEN	CP0	CP0	PWRTE	WDTE	F0SC1	F0SC0	Register: CONFIG
													Address 2007h	
													bit0	
bit13														
bit 13-7: <b>CP0</b> :コード保護ビット(2)														
5-4: 1=コード保護オフ														
0=全メモリがコード保護、ただし、00h-3Fhは書き込み可														
bit 6: <b>BODEN</b> :ブラウンアウトリセットイネーブルビット(1)														
1=BORはイネーブル														
0=BORはディセーブル														
bit 3: <b>PWRTE</b> :パワーアップタイマイネーブルビット(1)														
1=PWRTはディセーブル														
0=PWRTはイネーブル														
bit 2: <b>WDTE</b> :ウォッチドッグタイマイネーブルビット														
1=WDTはイネーブル														
0=WDTはディセーブル														
bit 1-0: <b>FOSC1:FOSC0</b> :オシレータ選択ビット														
11=RCオシレータ														
10=HSオシレータ														
01=XTオシレータ														
00=LPオシレータ														
注1: ブラウンアウトリセットをイネーブルすると、ビットPWRTEの値を無視して自動的にパワーアップタイム(PWRT)をイネーブルします。ブラウンアウトリセットがイネーブルの時はいつでも、確実にパワーアップタイムがイネーブルであるようにします。														
2: すべてのCP0ビットは、記載されたコード保護表をイネーブルするために同じ値を与えなければなりません。														

図14-3： PIC16C73/74用コンフィギュレーションワード

-	-	-	-	-	-	-	-	CP1	CP0	PWRTE	WDTE	F0SC1	F0SC0	Register: CONFIG
													Address 2007h	
													bit0	
bit13														
bit 13-5: <b>未使用</b> : '1'としてリード														
bit 4: <b>CP1:CP0</b> :コード保護ビット														
11 = コード保護オフ														
10 = プログラムメモリコードの上位半分を保護														
01 = プログラムメモリコードの上位3/4を保護														
00 = 全メモリがコード保護														
bit 3: <b>PWRTE</b> : パワーアップタイマイネーブルビット														
1 = パワーアップタイムはイネーブル														
0 = パワーアップタイムはディセーブル														
bit 2: <b>WDTE</b> :ウォッチドッグタイマイネーブルビット														
1 = WDTはイネーブル														
0 = WDTはディセーブル														
bit 1-0: <b>FOSC1:FOSC0</b> : オシレータ選択ビット														
11 = RCオシレータ														
10 = HSオシレータ														
01 = XTオシレータ														
00 = LPオシレータ														

図14-4: PIC16C72/73A/74A用コンフィギュレーションワード

CP1	CP0	CP1	CP0	CP1	CP0	-	BODEN	CP1	CP0	PWRTE	WDTE	F0SC1	F0SC0	Register: CONFIG Address 2007h
bit3														bit0
bit 13-8	<b>CP1:CP0:</b> コード保護ビット <sup>(2)</sup>													
5-4:	11 = コード保護オフ 10 = プログラムメモリコードの上位半分を保護 01 = プログラムメモリコードの上位3/4を保護 00 = 全メモリがコード保護													
bit 7:	<b>未使用:</b> '1' としてリード													
bit 6:	<b>BODEN:</b> ブラウンアウトリセットイネーブルビット <sup>(1)</sup> 1 = BORはイネーブル 0 = BORはディセーブル													
bit 3:	<b>PWRTE:</b> パワーアップタイムイネーブルビット <sup>(1)</sup> 1 = PWRTはディセーブル 0 = PWRTはイネーブル													
bit 2:	<b>WDTE:</b> ウォッチドッグタイムイネーブルビット 1 = WDTはイネーブル 0 = WDTはディセーブル													
bit 1-0:	<b>FOSC1:FOSC0:</b> オシレータ選択ビット 11 = RCオシレータ 10 = HSオシレータ 01 = XTオシレータ 00 = LPオシレータ													
注1: ブラウンアウトリセットをイネーブルすると、PWRTEビットの値を無視して自動的にパワーアップタイム(PWRT)をイネーブルします。ブラウンアウトリセットがイネーブルの時はいつでも、確実にパワーアップタイムがイネーブルであるようにします。														
2: すべてのCP1:CP0ペアは、記載されたコード保護表をイネーブルするために同じ値を与えなければなりません。														

## 14.2 オシレータの設定

適用デバイス							
710	71	711	72	73	73A	74	74A

### 14.2.1 オシレータの種類

PIC16CXXは4種類の異なるオシレータのモードで動作できます。次の4種類のモードから1つを選択するために、2個のコンフィギュレーションビット(FOSC1とFOSC0)をプログラミングできます。

- ・ LP ロー・パワークリスタル
- ・ XT クリスタル/レゾネータ
- ・ HS 高速クリスタル/レゾネータ
- ・ RC 抵抗/容量

### 14.2.2 クリスタルオシレータ/セラミックレゾネータ

XT、LP、HSモードでは、発振を起こすためにクリスタルまたはセラミックレゾネータ(共振子)がOSC1/CLKINとOSC2/CLKOUTのピンに接続されます(図14-5参照)。PIC16CXXのオシレータの設計にはパラレルカットクリスタルを使用する必要があります。シリーズカットクリスタルの使用はクリスタル製造会社の規格外の周波数を与えることがあります。XT、LP、HSモードでは、そのデバイスはOSC1/CLKINピンを駆動するための外部クロックソースを使うことができます(図14-6参照)。

図 14-5: クリスタル/セラミックレゾネータ操作 (HS、XT、LPのOSC構成)

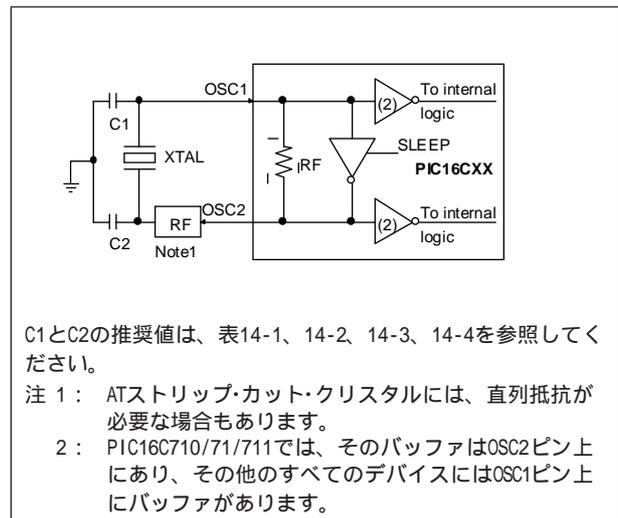
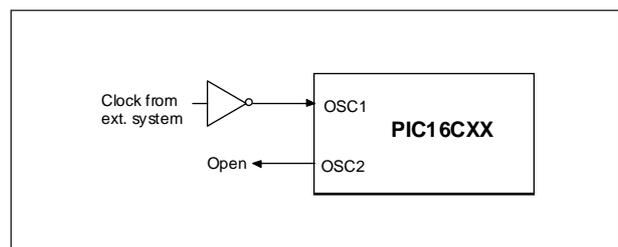


図14-6: 外部クロック入力の操作 (HS、XT、LPのOSC構成)



# 暫定版

英語最新版データシートと併用しご利用下さい。

AC/DC仕様は最新英文データシートをご参照下さい。

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

表14-1: PIC16C71のセラミックレゾネータ

テスト範囲:			
Mode	Freq	OSC1	OSC2
XT	455 kHz	47 - 100 pF	47 - 100 pF
	2.0 MHz	15 - 68 pF	15 - 68 pF
	4.0 MHz	15 - 68 pF	15 - 68 pF
HS	8.0 MHz	15 - 68 pF	15 - 68 pF
	16.0 MHz	10 - 47 pF	10 - 47 pF
注: C1とC2の推奨値はテストされた範囲の表と全く同じです。高キャパシタンスはオシレータの安定性を増しますが、スタートアップ時間も増えます。これらの値は設計の指針を示すためのものです。個々のレゾネータにはそれぞれの特性がありますので、外部コンポーネントの適切な値についてはレゾネータの製造元にお問い合わせください。			
使用レゾネータ:			
455 kHz	Panasonic EFO-A455K04B	±0.3%	
2.0 MHz	Murata Erie CSA2.00MG	±0.5%	
4.0 MHz	Murata Erie CSA4.00MG	±0.5%	
8.0 MHz	Murata Erie CSA8.00MT	±0.5%	
16.0 MHz	Murata Erie CSA16.00MX	±0.5%	
容量内蔵タイプは使用していません。			

表14-3: PIC16C710/711/72/73/73A/74/74Aのセラミックレゾネータ

テスト範囲:			
Mode	Freq	OSC1	OSC2
XT	455 kHz	68 - 100 pF	68 - 100 pF
	2.0 MHz	15 - 68 pF	15 - 68 pF
	4.0 MHz	15 - 68 pF	15 - 68 pF
HS	8.0 MHz	10 - 68 pF	10 - 68 pF
	16.0 MHz	10 - 22 pF	10 - 22 pF
注: C1とC2の推奨値はテストされた範囲の表と全く同じです。高キャパシタンスはオシレータの安定性を増しますが、スタートアップ時間も増えます。これらの値は設計の指針を示すためのものです。個々のレゾネータにはそれぞれの特性がありますので、外部コンポーネントの適切な値についてはレゾネータの製造元にお問い合わせください。			
使用レゾネータ:			
455 kHz	Panasonic EFO-A455K04B	±0.3%	
2.0 MHz	Murata Erie CSA2.00MG	±0.5%	
4.0 MHz	Murata Erie CSA4.00MG	±0.5%	
8.0 MHz	Murata Erie CSA8.00MT	±0.5%	
16.0 MHz	Murata Erie CSA16.00MX	±0.5%	
容量内蔵タイプは使用していません。			

表14-2: PIC16C71のクリスタルオシレータ用の容量選択

Mode	Freq	OSC1	OSC2
LP	32 kHz	33 - 68 pF	33 - 68 pF
	200 kHz	15 - 47 pF	15 - 47 pF
XT	100 kHz	47 - 100 pF	47 - 100 pF
	500 kHz	20 - 68 pF	20 - 68 pF
	1 MHz	15 - 68 pF	15 - 68 pF
	2 MHz	15 - 47 pF	15 - 47 pF
	4 MHz	15 - 33 pF	15 - 33 pF
HS	8 MHz	15 - 47 pF	15 - 47 pF
	20 MHz	15 - 47 pF	15 - 47 pF
注: 高キャパシタンスはオシレータの安定性を増しますが、スタートアップ時間も増えます。これらの値は設計の指針を示すためのものです。低駆動レベルの設定でのクリスタルをオーバードライブしないために、XTモードと同様にHSモードではRsが必要となります。個々のクリスタルにはそれぞれの特性がありますので、外部コンポーネントの適切な値についてはクリスタルの製造元にお問い合わせください。			

表14-4: PIC16C710/711/72/73/73A/74/74Aのクリスタル・オシレータ用の容量選択

Mode	Freq	OSC1	OSC2
LP	32 kHz(1)	15 - 47 pF	15 - 47 pF
	200 kHz	15 - 33 pF	15 - 33 pF
XT	100 kHz	47 - 100 pF	47 - 100 pF
	500 kHz	20 - 68 pF	20 - 68 pF
	1 MHz	15 - 68 pF	15 - 68 pF
	2 MHz	15 - 47 pF	15 - 47 pF
HS	8 MHz	15 - 47 pF	15 - 47 pF
	20 MHz	15 - 47 pF	15 - 47 pF
注: 高キャパシタンスはオシレータの安定性を増しますが、スタートアップ時間も増えます。これらの値は設計の指針を示すためのものです。低駆動レベルの設定でのクリスタルをオーバードライブしないために、XTモードと同様にHSモードではRsが必要となります。個々のクリスタルにはそれぞれの特性がありますので、外部コンポーネントの適切な値についてはクリスタルの製造元にお問い合わせください。 注1: VDD>4.5V、C1=C2 30pFが推奨値です。			

### 14.2.3 外部クリスタルオシレータ回路

オシレータモジュールの使用や、TTLゲートを使って簡単なオシレータ回路を作ることができます。オシレータモジュールは広い動作範囲と良い安定性を持っています。良く設計されたクリスタルオシレータはTTLゲートで良い性能を発揮します。2種類のクリスタルオシレータ回路は、直列共振または並列共振として使うことができます。

図14-7に並列共振オシレータ回路の構成を示します。この回路はクリスタルの基本周波数を使うために設計されています。74AS04インバータが並列オシレータに必要な180度の位相シフトを行います。4.7kの抵抗は安定用のネガティブフィードバックのためです。10kのポテンショメータは74AS04を線形領域で使うためのバイアス用です。これは外部オシレータの設計に使えます。

図14-7： 外部並列共振クリスタルオシレータ回路

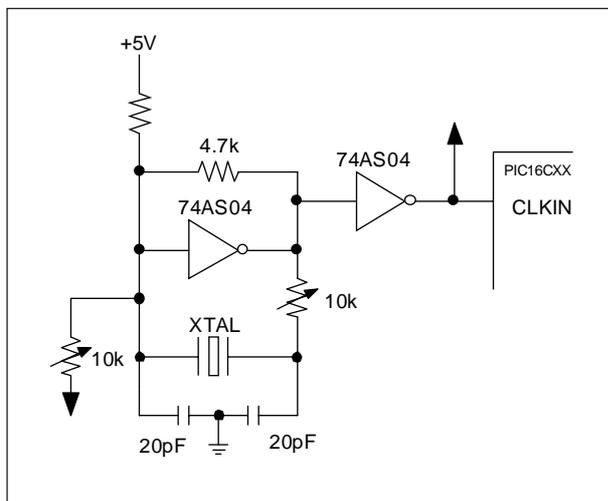
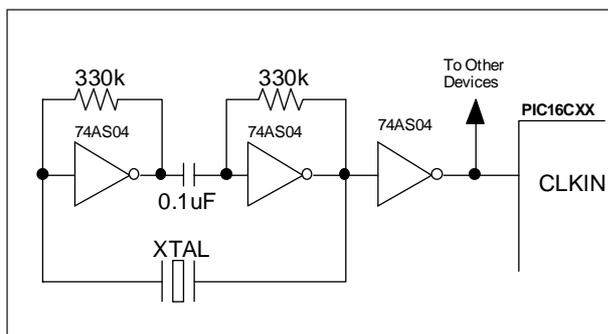


図14-8に直列共振オシレータ回路を示します。この回路もクリスタルの基本周波数を使うために設計されています。インバータが直列共振オシレータ回路での180度の位相シフトを行います。330kの抵抗はインバータを線形領域にバイアスするためのネガティブフィードバックを構成します。

図14-8： 外部直列共振クリスタルオシレータ回路



### 14.2.4 RCオシレータ

それほどタイミング精度を必要としないアプリケーションでは、“RC”部品を使用すればさらにコストを節約できます。RCオシレータ周波数は供給電圧、抵抗(Rext)とキャパシタ(Cext)の値、動作温度により変わります。これに加え、このオシレータ周波数は通常の生産パラメータのばらつきにより、同製品間でも異なります。さらに、パッケージの種類によるリードフレーム容量の差も、特に低いCextの値で、発振周波数に影響を与えます。外部に使うRとCの部品に誤差があるため、そのばらつきを考慮する必要があります。図14-9にR/Cの組み合わせがどのようにPIC16CXXに接続されるかを示します。2.2k以下のRext値では、オシレータ動作が不安定になるか、完全に停止することがあります。非常に高いRext値では(例えば、1M)、オシレータがノイズ、湿度、漏れ電流の影響を受けやすくなります。したがって、Rextは3kから100kの間をお勧めします。

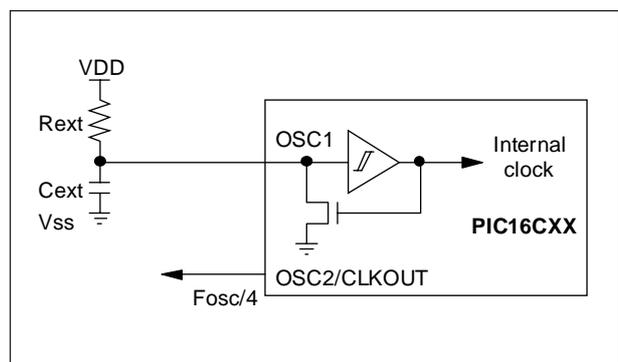
オシレータは外部キャパシタが無くても(Cext=0pF)動作しますが、ノイズと安定のために、20pF以上の値を使うことをお勧めします。外部容量が無い小さいと、PCBのトレース容量やパッケージのリードフレーム容量などの外部容量の変化で、発振周波数が非常に異なることがあります。

通常の製造工程のばらつきによる製品毎のRC周波数の変化に関しては、必要とするデバイスの特性データをご覧ください。その変化は、より大きなR(漏れ電流のばらつきが大きなRに対してRC周波数に多くの影響を与えるため)、そしてより小さなC(入力容量のばらつきがRC周波数に多くの影響を与えるため)に対して大きくなります。

与えられたR、C、VDDの値に対する動作温度による周波数変化と同様に、与えられたRext/Cextの値に対するVDDによるオシレータ周波数変化に関しては、必要とするデバイスの特性データをご覧ください。

4分周されたオシレータ周波数がOSC2/CLKOUTピンに出力されます。これは、テストの目的や、他のロジックと同期にするために使用できます(波形に関しては図3-5参照)。

図14-9： RCオシレータモード



## PIC16C7X

### 14.3 リセット

適用デバイス							
710	71	711	72	73	73A	74	74A

PIC16CXXは色々な種類のリセットを区別します。

- ・パワーオンリセット(POR)
- ・通常動作中のMCLRリセット
- ・SLEEP中のMCLRリセット
- ・WDTリセット(通常動作時)
- ・ブラウンアウトリセット(BOR)(PIC16C710/711/72/73A/74Aのみ)

いくつかのレジスタは、どのリセット条件からも影響を受けません。それらのステータスはPORでは不定で、他のリセットでは変化しません。他のほとんどのレジスタは、パワーオンリセット(POR)、MCLRとWDTリセット、SLEEP中のMCLRリセット、ブラウンアウトリセット(BOR)で“リセット状態”にセットされます。それらはWDTウェイクアップによって影響を受けず、通常動作の再開として見なされます。TOとPDビットは表14-7と14-8に示すように、異なったりリセット状態では、それぞれ異なってセットまたはクリアされます。これらのビットはリセットの性質を判断するためにソフトウェアで使われま

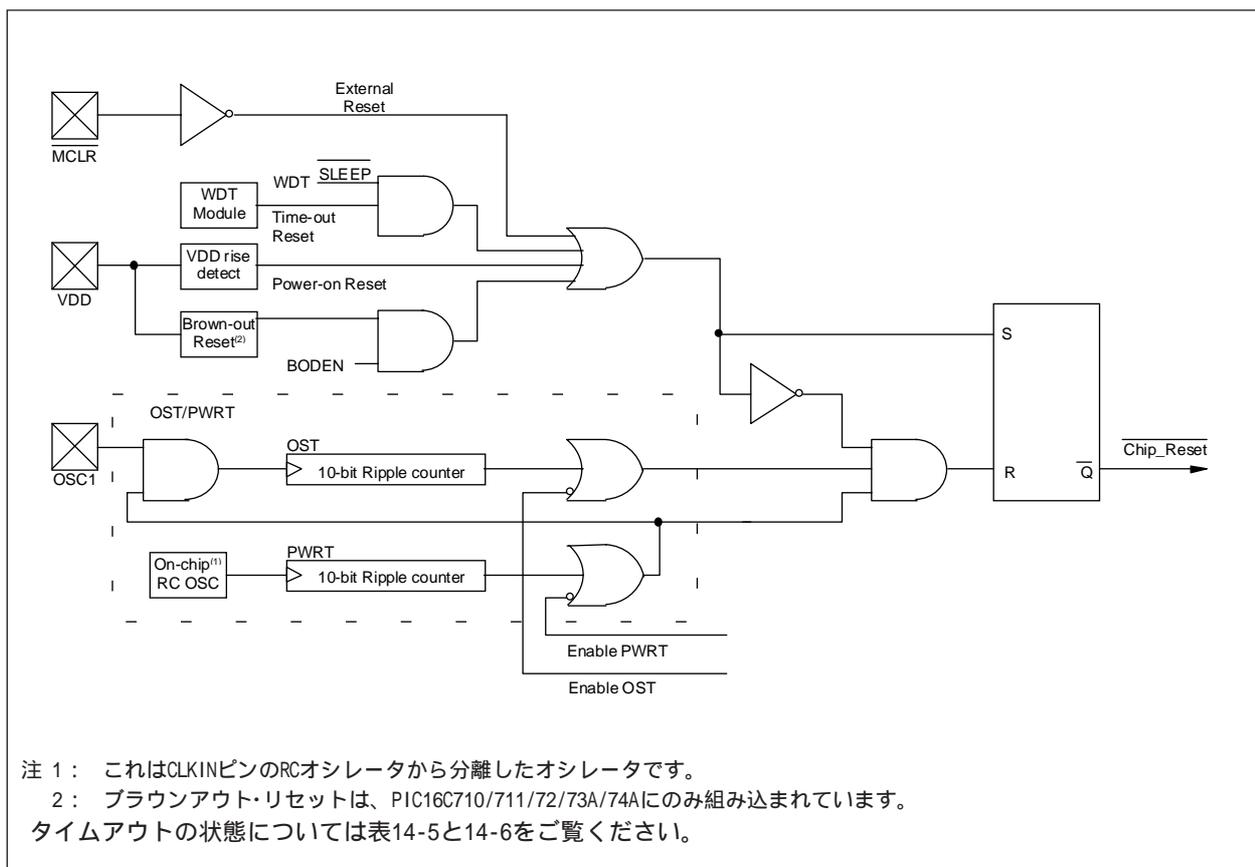
す。すべてのレジスタのリセット状態の詳細に関しては表14-10をご覧ください。

内蔵リセット回路の簡易ブロック図は図14-10に示します。

PIC16C710/711/72/73A/74Aには、MCLRリセットノイズフィルタがあります。そのフィルタは小さいパルスを検出し無視します。

WDTリセットはMCLRピンをLowに駆動しないので注意する必要があります。

図14-10： 内蔵リセット回路のブロック図



### 14.4 パワーオンリセット(POR) パワーアップ タイム(PWRT) オシレータスタートアップタイム(OST) ブラウンアウトリセット(BOR)

適用デバイス							
710	71	711	72	73	73A	74	74A

#### 14.4.1 パワーオンリセット(POR)

$V_{DD}$ の立ち上がりが検出された時(1.5V-2.1Vの範囲で)、チップ内にパワーオンリセットパルスが発生します。PORの利点を使うためには、MCLRピンを $V_{DD}$ に直接(または抵抗を通して)接続するだけです。これにより、パワーオンリセットに通常必要とされる外部のRC部品を省略できます。 $V_{DD}$ の最大立ち上がり時間が規定されています。詳細については電氣的仕様をご覧ください。

デバイスが通常の動作を始めると(リセット状態から出ると)、デバイスの動作しているパラメータ(電圧、周波数、温度など)は動作を確実にするために適切にする必要があります。これらの状態に達しない場合、そのデバイスは動作状態に達するまでリセット状態を保持する必要があります。

追加情報については、アプリケーションノートAN607 “パワーアップトラブルシューティング”をご覧ください。

#### 14.4.2 パワーアップタイム(PWRT)

パワーアップタイムは、PORからのパワーアップ時のみ、72msで公称のタイムアウトを生成します。パワーアップタイムは内部RCオシレータで動作します。PWRTが動作状態の間は、チップをリセット状態に保持します。PWRTの遅延時間の間に $V_{DD}$ が規定の電圧レベルに達している必要があります。コンフィギュレーションビットにより、PWRTをイネーブル/ディセーブルにできます。

パワーアップ時間の遅れはチップごとにより、また

$V_{DD}$ 、温度、製造工程のばらつきにより異なります。詳細はDCパラメータの章をご覧ください。

#### 14.4.3 オシレータスタートアップタイム(OST)

オシレータスタートアップタイム(OST)はPWRTの遅延時間が終わった後、1024オシレータサイクル(OSC1入力から)の遅延時間を作ります。これにより、クリスタルオシレータまたはレゾネータの開始と安定性を確実にします。

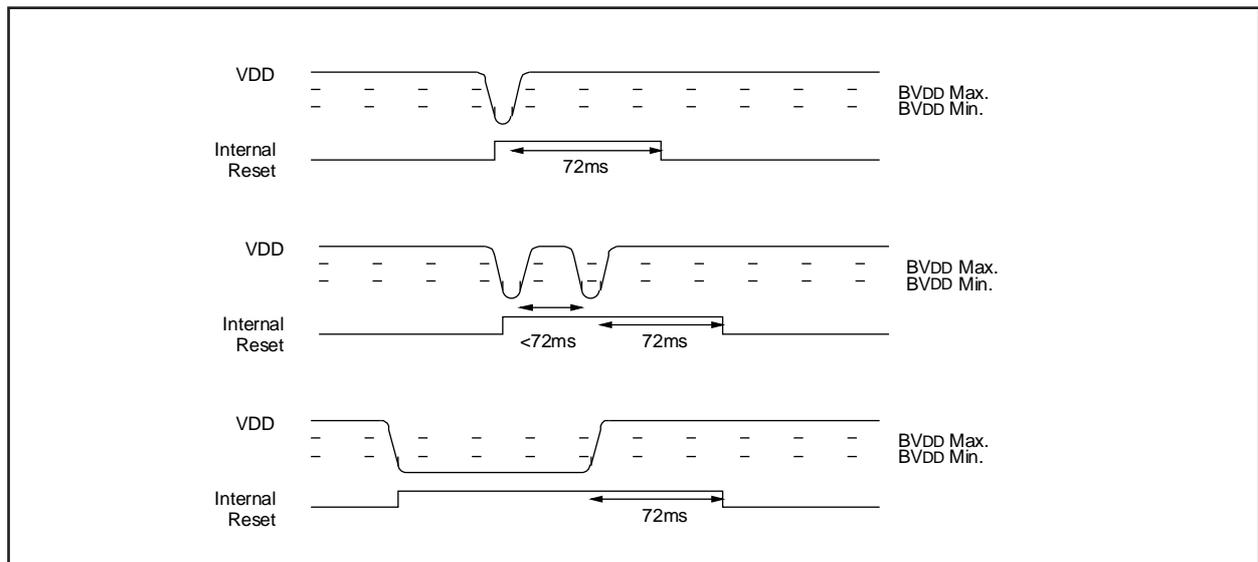
OSTタイムアウトはXT、LP、HSのモードとパワーオンリセットやSLEEPからのウェークアップ時のみ起動されます。

#### 14.4.4 ブラウンアウトリセット(BOR)

適用デバイス							
710	71	711	72	73	73A	74	74A

コンフィギュレーションビットBODENは、ブラウンアウトリセットの回路設計をディセーブルする(クリア/プログラムされた場合)、またはイネーブルする(セットされた場合)ことができます。 $V_{DD}$ がパラメータ#35よりも大きくなるために4.0V(3.8V-4.2V範囲)以下で立ち下げる場合は、ブラウンアウトの状態はチップをリセットします。 $V_{DD}$ がパラメータ#35よりも小さくなるために4.0V以下で立ち下げる場合は、リセットは起こりません。 $V_{DD}$ が $BV_{DD}$ 以上で立ち上がるまで、チップはブラウンアウトリセットのままです。パワーアップタイムを起動し、リセット状態のチップを付加の72ms間リセット状態を保ちます。パワーアップタイムが動作中に、 $V_{DD}$ が $BV_{DD}$ 以下で立ち下がる場合は、チップはブラウンアウトリセットに戻り、パワーアップタイムは初期化されます。 $V_{DD}$ が $BV_{DD}$ 以上で立ち上がると、パワーアップタイムは72msリセットを実行します。ブラウンアウトリセットがイネーブルの場合は、パワーアップタイムは常にイネーブルである必要があります。図14-11に種類別のブラウンアウトの状態を示します。

図14-11: ブラウンアウトの状態



# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。

## PIC16C7X

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

### 14.4.5 タイムアウトのシーケンス

パワーアップ時のタイムアウトシーケンスは次のようになります。

POR遅延時間が終わった後、最初にPWRTタイムアウトを起動し、それからOSTをアクティブにします。全体のタイムアウトはオシレータの設定とPWRTの状態を基に変化します。例えば、PWRTがディセーブルであるRCモードでは、タイムアウトが全くなくなります。図14-12、14-13、14-14にパワーアップ時のタイムアウト手順を示します。

タイムアウトがPORパルスから起こるため、MCLRが十分な時間、ローを保持された場合、そのタイムアウトは終了します。それから、MCLRをハイにすることにより、実行をすぐに開始します(図14-13参照)。これはテスト目的や並列に動作している複数のPIC16CXXデバイスを同期させるために役立ちます。

表14-9にいくつかの特殊機能のレジスタのリセット条件を示し、表14-10にはすべてのレジスタのリセット条件を示します。

### 14.4.6 パワー制御/ステータスレジスタ(PCON)

適用デバイス							
710	71	711	72	73	73A	74	74A

パワー制御/ステータスレジスタPCONは、そのデバイスにより異なりますが、2ビットまであります。ビット0はPIC16C73またはPIC16C74では未使用です。

ビット0はブラウンアウトリセットステータスビットBORです。ビットBORはパワーオンリセット時は不定なので、ユーザがこのビットをセットしなければなりません。この後、BORが発生した事を示す、BORのクリアをモニターすることで、引き続き発生したりセットを確認することができます。そのBORビットは“関係のない”ビットで、ブラウンアウトリセット回路設計がディセーブルされる場合(コンフィギュレーションワードでのビットBODENをクリアすることにより)、予知する必要はありません。

ビット1はパワーオンリセットステータスビットPORです。パワーオンリセットでクリアされ、他からは影響を受けません。パワーオンリセットに続いてこのビットをセットする必要があります。

表14-5: 様々な状態でのタイムアウト(PIC16C71/73/74)

Oscillator Configuration	Power-up		Wake-up from SLEEP
	PWRTE = 1	PWRTE = 0	
XT, HS, LP	72 ms + 1024Tosc	1024Tosc	1024 Tosc
RC	72 ms	-	-

表14-6: 様々な状態でのタイムアウト(PIC16C710/711/72/73A/74A)

Oscillator Configuration	Power-up		Brown-out	Wake-up from SLEEP
	PWRTE = 0	PWRTE = 1		
XT, HS, LP	72 ms + 1024Tosc	1024Tosc	72ms + 1024Tosc	1024 Tosc
RC	72 ms	-	72ms	-

表14-7: ステータスビットとその意味(PIC16C71/73/74)

POR <sup>(1)</sup>	T0	PD	
0	1	1	パワーオン・リセット
0	0	x	不定。POR時に、T0はセット。
0	x	0	不定。POR時に、PDはセット。
1	0	1	WDTリセット
1	0	0	WDTウェークアップ
1	u	u	ノーマル動作時MCLRリセット
1	1	0	SLEEP時MCLRリセットまたはSLEEPからの割込みウェークアップ。

凡例: u=不定、x=未知

注 1: ビットPORはPIC16C71ではインプリメントされていません。

表14-8: ステータスビットとその意味(PIC16C710/711/72/73A/74A)

POR	BOR	T0	PD	
0	x	1	1	パワーオン・リセット
0	x	0	x	不定。POR時に、T0はセット。
0	x	x	0	不定。POR時に、PDはセット。
1	0	x	x	ブラウンアウト・リセット
1	1	0	1	WDTリセット
1	1	0	0	WDTウェークアップ
1	1	1	1	ノーマル動作時MCLRリセット
1	1	1	0	SLEEP時MCLRリセットまたはSLEEPからの割込みウェークアップ。

凡例: u=不定、x=未知

# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。  
<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

表14-9： 特殊レジスタのリセット条件

Condition	Program Counter	STATUS Register	PCON Register PIC16C710/711	PCON Register PIC16C73/74	PCON Register PIC16C72/73A/74A
パワーオン・リセット	000h	0001 1xxx	---- --0x	---- --0-	---- --0x
ノーマル動作時のMCLRリセット	000h	000u uuuu	---- --uu	---- --u-	---- --uu
SLEEP時のMCLRリセット	000h	0001 0uuu	---- --uu	---- --u-	---- --uu
WDTリセット	000h	0000 1uuu	---- --uu	---- --u-	---- --uu
WDTウェークアップ	PC + 1	uuu0 0uuu	---- --uu	---- --u-	---- --uu
ブラウンアウト・リセット	000h	0001 1uuu	---- --u0	N/A	---- --u0
SLEEPからの割込みウェークアップ	PC + 1 <sup>(1)</sup>	uuu1 0uuu	---- --uu	---- --u-	---- --uu

凡例： u=不変 x=未知、-=未使用のビット、'0'としてリード

注 1： 割込みによりウェークアップし、GIEビットがセットされた場合、PCは割込みベクタ(0004h)によりロードされます。

表14-10： すべてのレジスタに関する初期化条件

Register	Applicable Devices								Power-on Reset, Brown-out Reset	MCLR Resets WDT Reset	Wake-up via WDT or Interrupt
	710	71	711	72	73	73A	74	74A			
W	710	71	711	72	73	73A	74	74A	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF	710	71	711	72	73	73A	74	74A	N/A	N/A	N/A
TMRO	710	71	711	72	73	73A	74	74A	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	710	71	711	72	73	73A	74	74A	0000h	0000h	PC + 1 <sup>(2)</sup>
STATUS	710	71	711	72	73	73A	74	74A	0001 1xxx	000q quuu <sup>(3)</sup>	uuuq quuu <sup>(3)</sup>
FSR	710	71	711	72	73	73A	74	74A	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA	710	71	711	72	73	73A	74	74A	--x 0000	--u uuuu	--u uuuu
	710	71	711	72	73	73A	74	74A	--0x 0000	--uu uuuu	--uu uuuu
PORTB	710	71	711	72	73	73A	74	74A	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTC	710	71	711	72	73	73A	74	74A	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTD	710	71	711	72	73	73A	74	74A	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTE	710	71	711	72	73	73A	74	74A	---- -xxx	--- --uuu	--- --uuu
PCLATH	710	71	711	72	73	73A	74	74A	---0 0000	---0 0000	---u uuuu
INTCON	710	71	711	72	73	73A	74	74A	0000 000x	0000 000u	uuuu uuuu <sup>(1)</sup>
PIR1	710	71	711	72	73	73A	74	74A	-0-- 0000	-0-- 0000	-u--- uuuu <sup>(1)</sup>
	710	71	711	72	73	73A	74	74A	-000 0000	-000 0000	-uuu uuuu <sup>(1)</sup>
	710	71	711	72	73	73A	74	74A	0000 0000	0000 0000	uuuu uuuu <sup>(1)</sup>
PIR2	710	71	711	72	73	73A	74	74A	---- ---0	---- ---0	---- ---u <sup>(1)</sup>
TMR1L	710	71	711	72	73	73A	74	74A	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR1H	710	71	711	72	73	73A	74	74A	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1CON	710	71	711	72	73	73A	74	74A	--00 0000	--uu uuuu	--uu uuuu

凡例： u = 不変、x=未知、-=未使用のビット、'0'としてリード、q = 条件によって決まる値。

注 1： INTCON、PIR1、PIR2の1個以上のビットは影響を受けません(ウェークアップを引き起こすため)。

2： 割込みによりウェークアップし、GIEビットがセットされた場合、PCは割込みベクタ(0004h)に分岐します。

3： 特定条件でのリセット値については表14-9を参照。

# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。

## PIC16C7X

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

表14-10: すべてのレジスタに関する初期化条件 ( 続き )

Register	Applicable Devices								Power-on Reset, Brown-out Reset	MCLR Resets WDT Reset	Wake-up via WDT or Interrupt
TMR2	710	71	711	72	73	73A	74	74A	0000 0000	0000 0000	uuuu uuuu
T2CON	710	71	711	72	73	73A	74	74A	-000 0000	-000 0000	-uuu uuuu
SSPBUF	710	71	711	72	73	73A	74	74A	xxxx xxxx	uuuu uuuu	uuuu uuuu
SSPCON	710	71	711	72	73	73A	74	74A	0000 0000	0000 0000	uuuu uuuu
CCPR1L	710	71	711	72	73	73A	74	74A	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR1H	710	71	711	72	73	73A	74	74A	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP1CON	710	71	711	72	73	73A	74	74A	--00 0000	--00 0000	--uu uuuu
RCSTA	710	71	711	72	73	73A	74	74A	0000 -00x	0000 -00x	uuuu -uuu
TXREG	710	71	711	72	73	73A	74	74A	0000 0000	0000 0000	uuuu uuuu
RCREG	710	71	711	72	73	73A	74	74A	0000 0000	0000 0000	uuuu uuuu
CCPR2L	710	71	711	72	73	73A	74	74A	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR2H	710	71	711	72	73	73A	74	74A	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP2CON	710	71	711	72	73	73A	74	74A	0000 0000	0000 0000	uuuu uuuu
ADRES	710	71	711	72	73	73A	74	74A	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON0	710	71	711	72	73	73A	74	74A	00-0 0000	00-0 0000	uu-u uuuu
	710	71	711	72	73	73A	74	74A	0000 00-0	0000 00-0	uuuu uu-u
OPTION	710	71	711	72	73	73A	74	74A	1111 1111	1111 1111	uuuu uuuu
TRISA	710	71	711	72	73	73A	74	74A	---1 1111	---1 1111	---u uuuu
	710	71	711	72	73	73A	74	74A	--11 1111	--11 1111	--uu uuuu
TRISB	710	71	711	72	73	73A	74	74A	1111 1111	1111 1111	uuuu uuuu
TRISC	710	71	711	72	73	73A	74	74A	1111 1111	1111 1111	uuuu uuuu
TRISD	710	71	711	72	73	73A	74	74A	1111 1111	1111 1111	uuuu uuuu
TRISE	710	71	711	72	73	73A	74	74A	0000 -111	0000 -111	uuuu -uuu
PIE1	710	71	711	72	73	73A	74	74A	-0-- 0000	-0-- 0000	-u-- uuuu
	710	71	711	72	73	73A	74	74A	-000 0000	-000 0000	-uuu uuuu
	710	71	711	72	73	73A	74	74A	0000 0000	0000 0000	uuuu uuuu
PIE2	710	71	711	72	73	73A	74	74A	---- ---0	---- ---0	---- ---u
PCON	710	71	711	72	73	73A	74	74A	---- --0-	---- --u-	---- --u-
	710	71	711	72	73	73A	74	74A	---- --0u	---- --uu	---- --uu
PR2	710	71	711	72	73	73A	74	74A	1111 1111	1111 1111	1111 1111
SSPADD	70	71	71A	72	73	73A	74	74A	0000 0000	0000 0000	uuuu uuuu
SSPSTAT	70	71	71A	72	73	73A	74	74A	--00 0000	--00 0000	--uu uuuu
TXSTA	70	71	71A	72	73	73A	74	74A	0000 -010	0000 -010	uuuu -uuu
SPBRG	70	71	71A	72	73	73A	74	74A	0000 0000	0000 0000	uuuu uuuu
ADCON1	70	71	71A	72	73	73A	74	74A	---- --00	---- --00	---- --uu
	70	71	71A	72	73	73A	74	74A	---- --00	---- --00	---- --uu

- 凡例: u = 不変、x = 未知、- = 未使用のビット、'0' としてリード、q = 条件によって決まる値。  
注 1: INTCON、PIR1、PIR2の1個以上のビットは影響を受けます(ウェイクアップを引き起こすため)。  
2: 割込みによりウェイクアップし、GIEビットがセットされた場合、PCは割込みベクタ(0004h)に  
分岐します。  
3: 特定条件でのリセット値については表14-9を参照。

図14-12 : パワーアップ時のタイムアウトシーケンス(MCLRがV<sub>DD</sub>に接続されていない場合) : ケース1

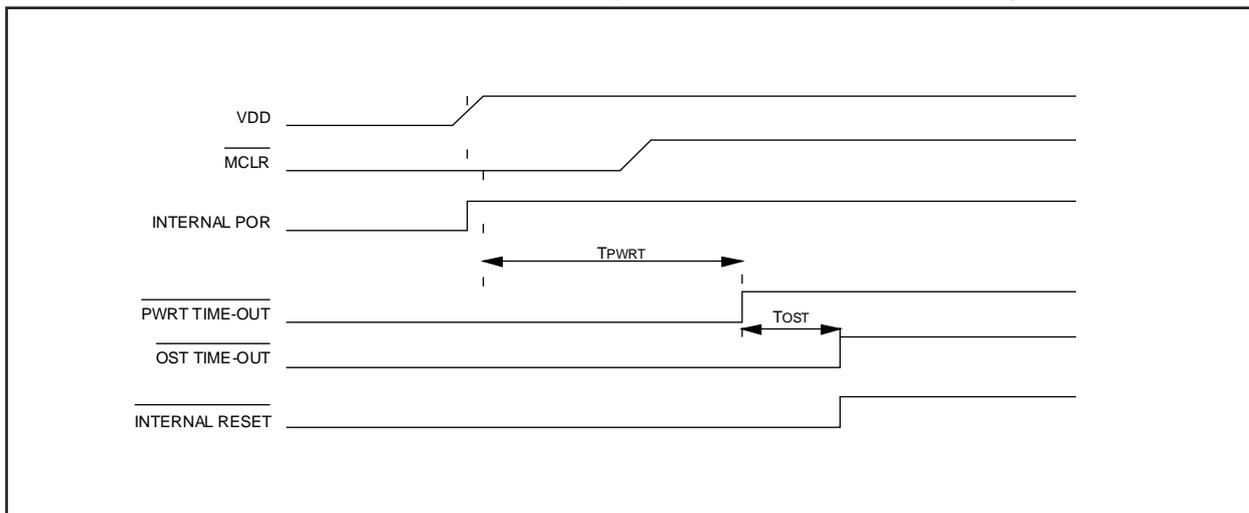


図14-13 : パワーアップ時のタイムアウトシーケンス(MCLRがV<sub>DD</sub>に接続されていない場合) : ケース2

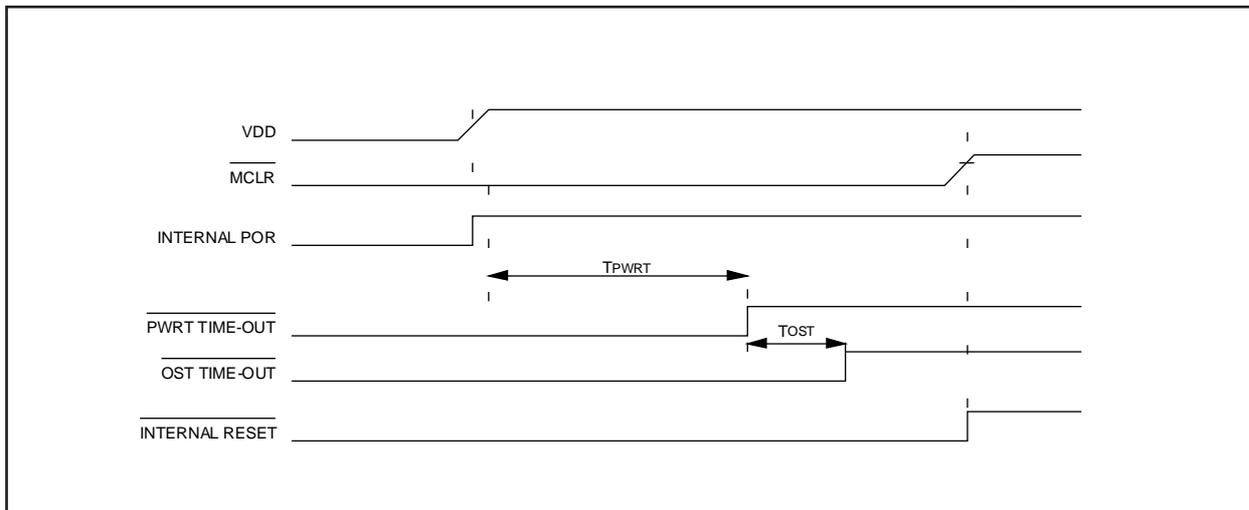


図14-14 : パワーアップ時のタイムアウトシーケンス(MCLRがV<sub>DD</sub>に接続されている場合)

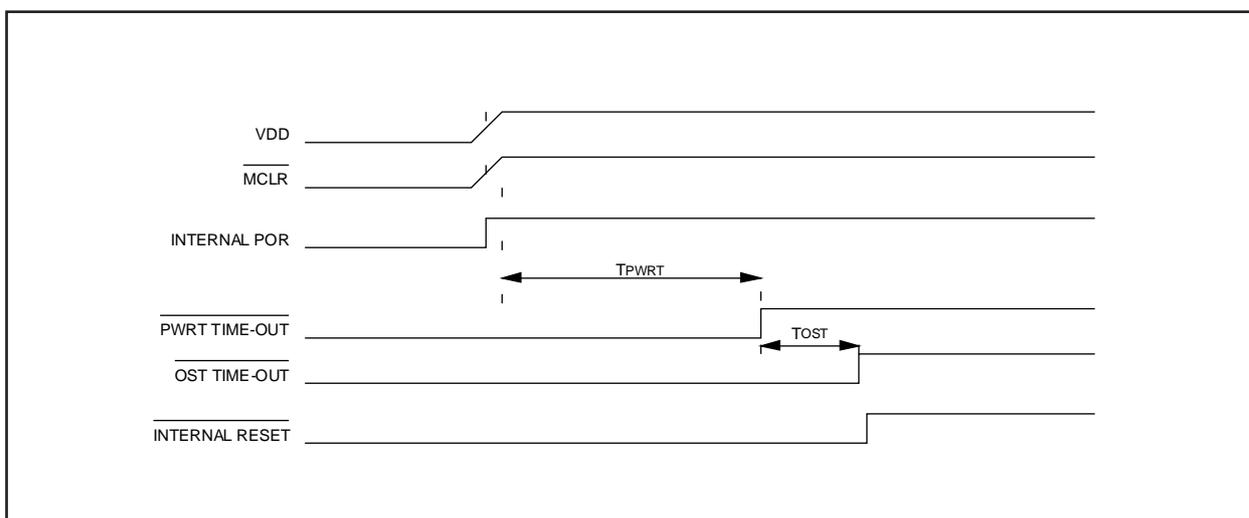


図14-15： 外部パワーオンリセット回路  
(V<sub>DD</sub>の立ち上がりが遅い時)

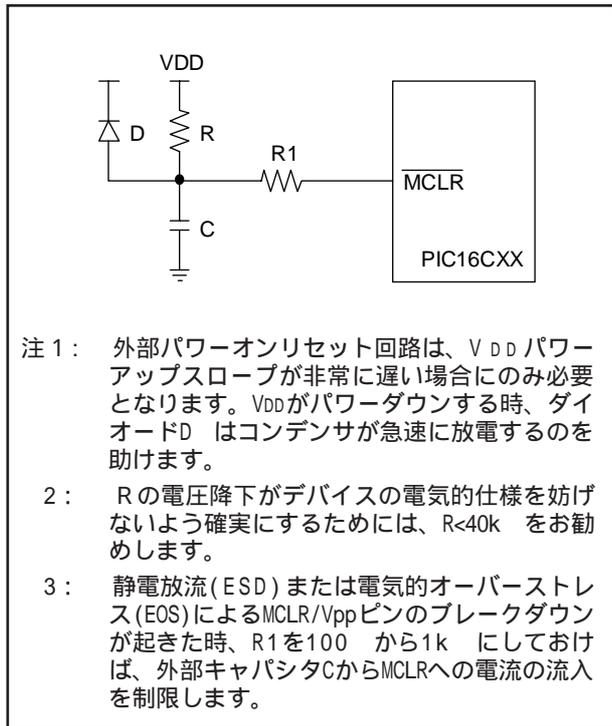


図14-16： 外部ブラウンアウト保護回路 1

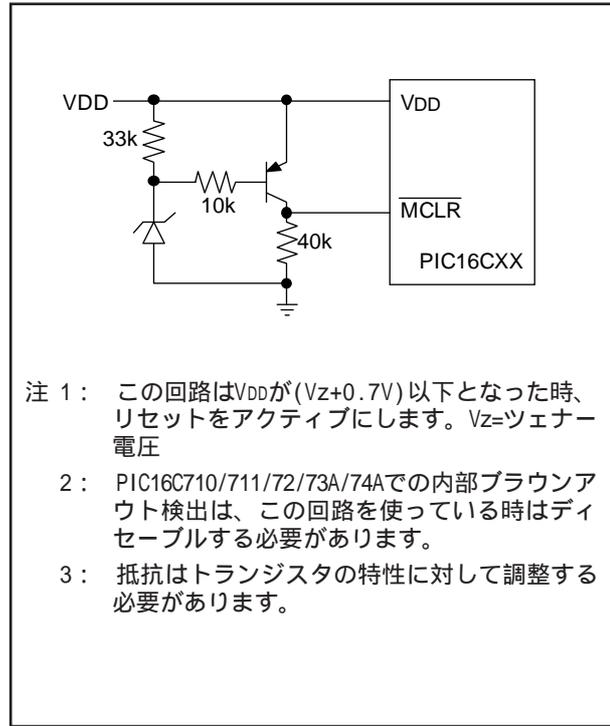


図14-17： 外部ブラウンアウト保護回路 2

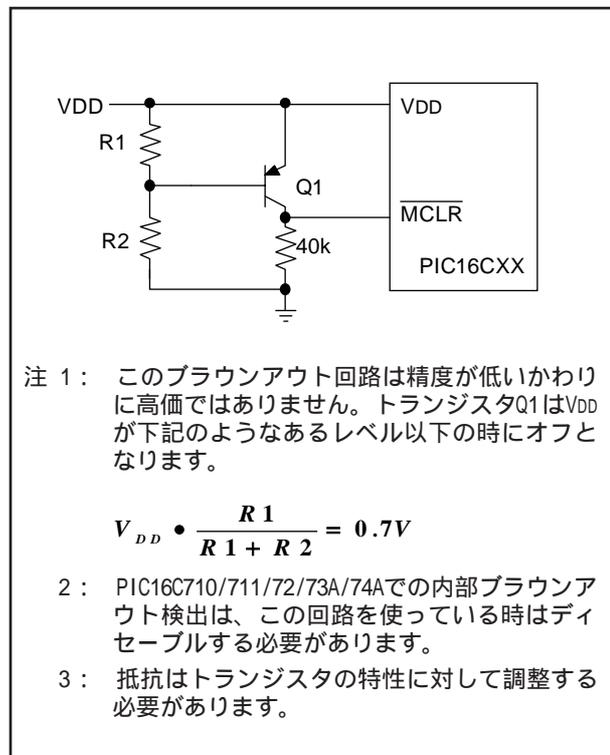




図14-18 : 割り込みロジック PIC16C710/71/711

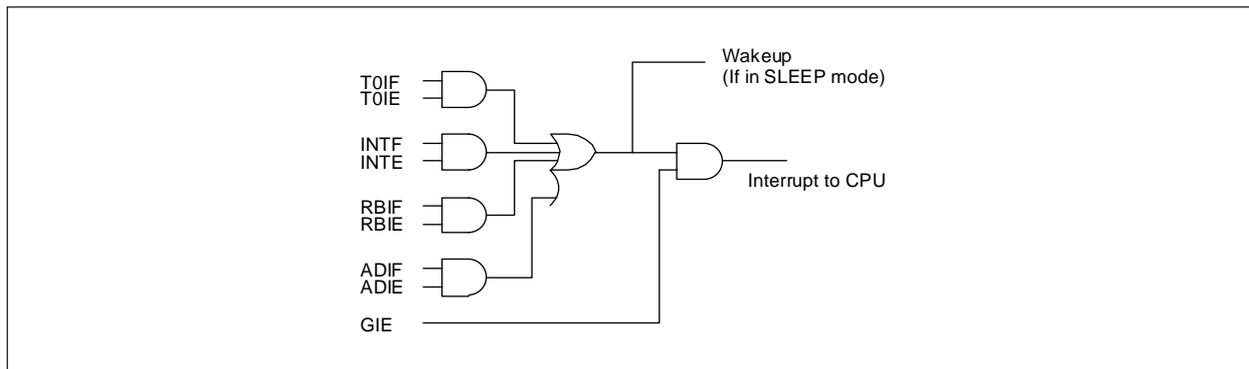


図14-19 : 割り込みロジック PIC16C72

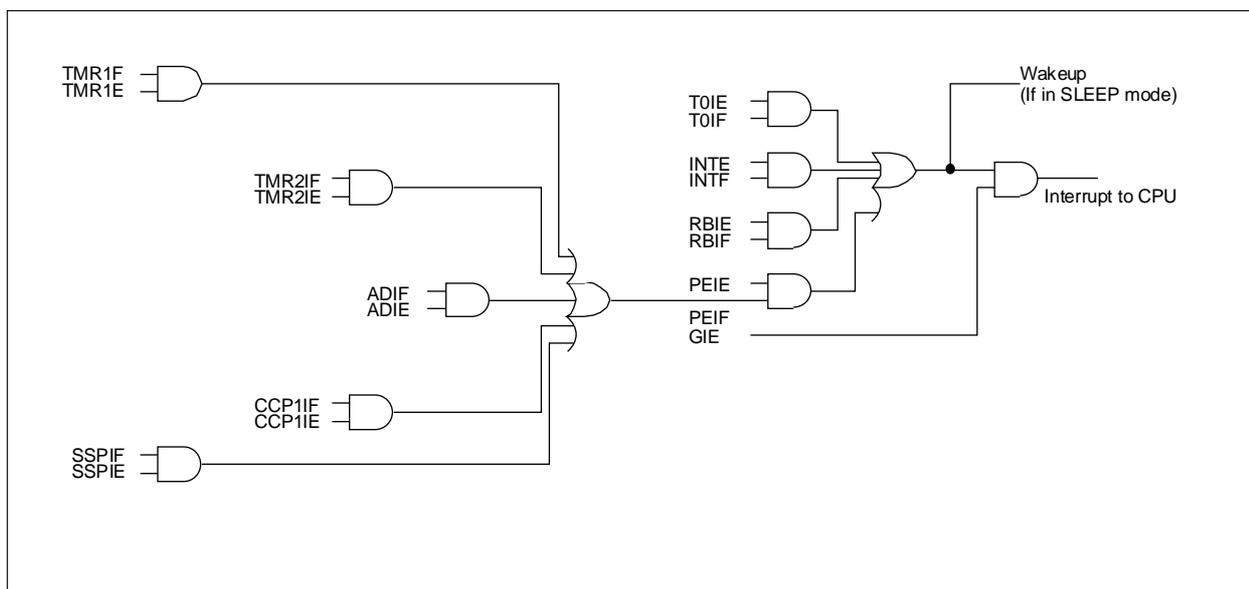


図14-20 : 割り込みロジック PIC16C73/73A

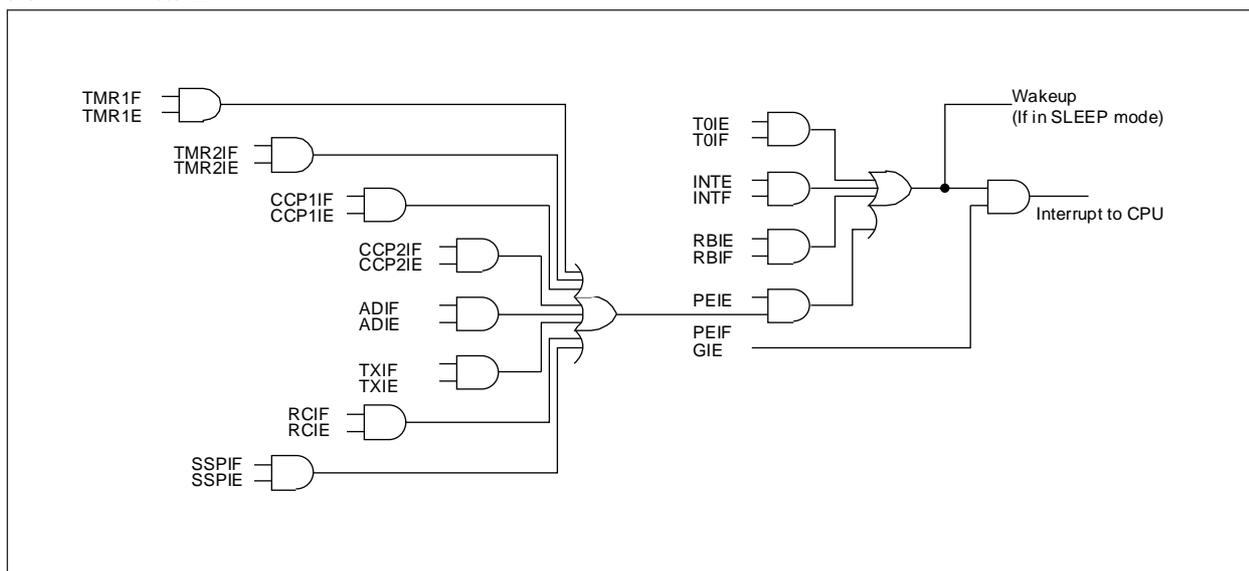


図14-21 : 割り込みロジック PIC16C74/74A

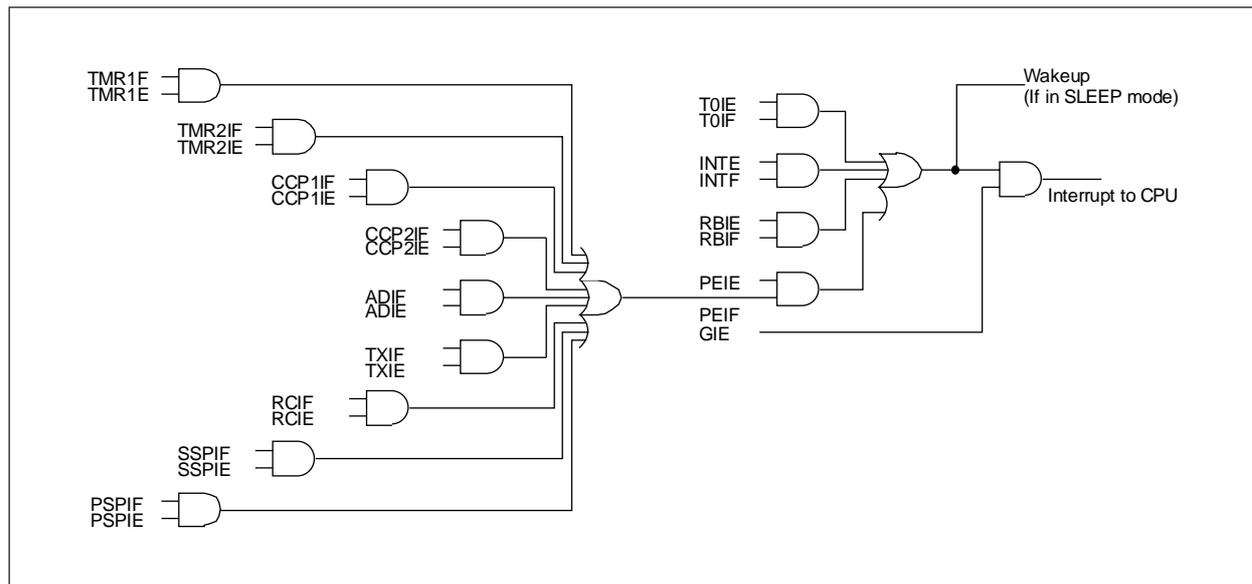
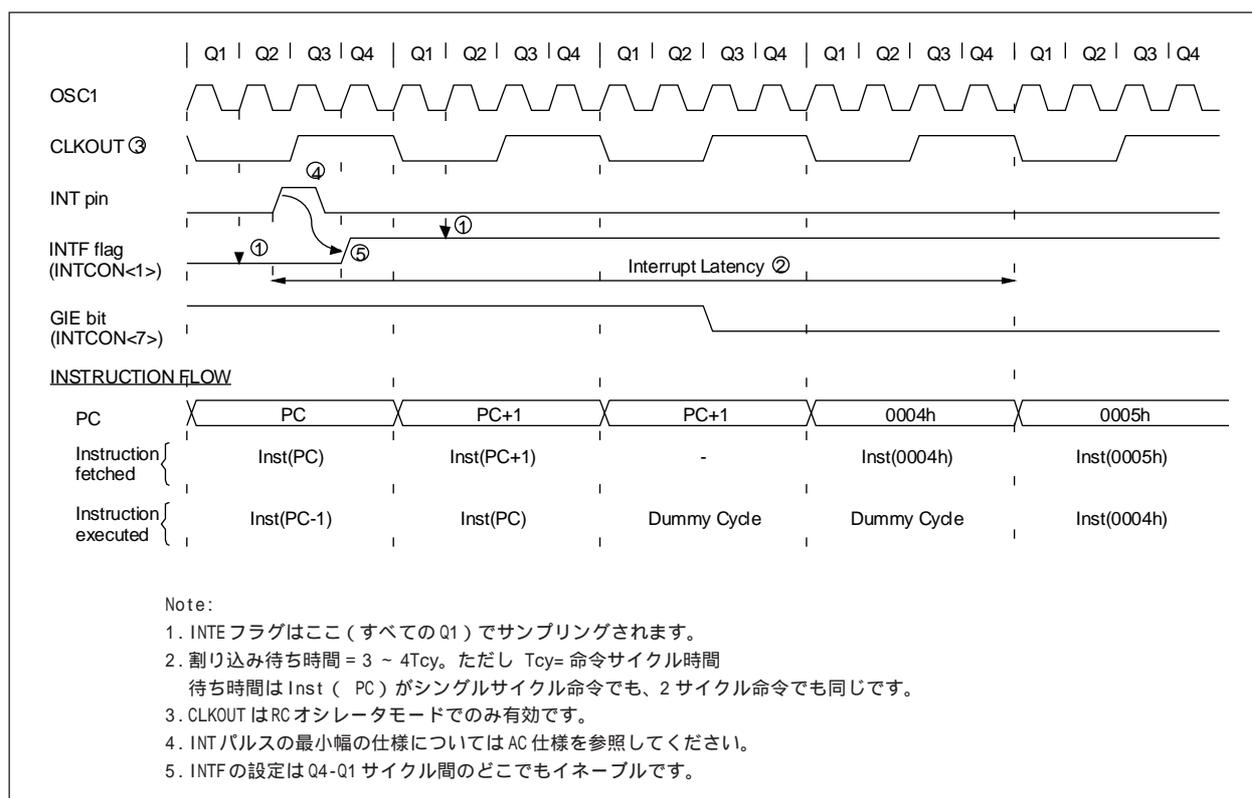


図14-22 : INTピン 割り込みのタイミング



### 14.5.1 INT インターラプト

RB0/INTピンの外部割込みは、エッジトリガで、INTEG (OPTION<6>)=1のときは立ち上がりエッジ、INTEG=0の時は立ち下がりエッジです。RB0/INTピンに有効なエッジが与えられたとき、INTF (OPTION<1>) ビットがセットされます。この割り込みはINTE (INTCON<4>) ビットをクリアすることでディセーブルできます。フラグビットINTFは、割り込みサービスルーチン内での処理が終了時点で必ずソフトウェアにてクリアする必要があります。INT割り込みは、INTE ビットがセットされている時、スリープモードからウェイクアップすることができます。ウェイクアップ後、割り込みベクタに分岐するかどうかは、グローバルインターラプトイネーブルビットGIEの状況によって決定します。(スリープモードの詳細は14.8章を参照)

### 14.5.2 TMR0 インターラプト

TMR0のオーバフロー (FFh 00h) によってTOIF (INTCON<2>) ビットがセットされます。TOIE (INTCON<5>) ビットのセット/クリアによってこの割り込みをイネーブル/ディセーブルすることができます。(7.0章参照)

### 14.5.3 PORTB 変化インターラプト

PORTB<7:4>のレベル変化によってRBIF (INTCON<0>) ビットがセットされます。RBIE (INTCON<4>) ビットのセット/クリアによってこの割り込みをイネーブル/ディセーブルすることができます。(5.2章を参照)

注意：PIC16C71、PIC16C73、PIC16C74のみ

リード動作の実行開始時(Q 2 サイクルから開始)に、I/Oピンの変化が発生した場合、RBIF割り込みフラグはセットされません。

## 14.6 インターラプト期間中のコンテキストセーブ

適用デバイス							
710	71	711	72	73	73A	74	74A

割り込みが発生すると、CPUコアはPC値のみスタックにセーブします。一般的に割り込み発生時、重要なレジスタ、たとえば、Wレジスタや、STATUSレジスタなどを、ユーザがソフトウェアにてセーブしなければなりません。

例14-1は、Wレジスタや、STATUSレジスタの待避/復帰の例です。Wレジスタの内容は、W\_TEMPに待避されます。W\_TEMPのアドレスは、バンク0、バンク1の双方に存在します。それ以外は、バンク0のみです。

例：

- Wレジスタを待避 (バンク0またはバンク1)
- STATUSレジスタを待避 (バンク0)
- PCLATHを待避 (バンク0)
- FSRを待避 (バンク0)
- 割り込みサービスルーチンを実行
- FSRを復帰
- PCLATHを復帰
- STATUSレジスタの内容を復帰
- Wレジスタの内容を復帰

### 例14-1： STATUS、Wレジスタのセーブ

```

MOVWF  W_TEMP           ; Copy W to TEMP register, could be bank one or zero
SWAPF  STATUS,W         ; Swap status to be saved into W
CLRF   STATUS            ; bank 0, regardless of current bank, Clears IRP,RP1,RP0
MOVWF  STATUS_TEMP      ; Save status to bank zero STATUS_TEMP register
MOVF   PCLATH, W        ; Only required if using pages 1, 2 and/or 3
MOVWF  PCLATH_TEMP      ; Save PCLATH into W
CLRF   PCLATH           ; Page zero, regardless of current page
MOVF   FSR, W           ; Copy FSR to W
MOVWF  FSR_TEMP         ; Copy FSR from W to FSR_TEMP
:
:   (ISR)
:
MOVF   FSR_TEMP, W      ; Restore FSR
MOVWF  FSR              ; Move W to FSR
MOVF   PCLATH_TEMP, W  ; Restore PCLATH
MOVWF  PCLATH          ; Move W into PCLATH
SWAPF  STATUS_TEMP,W   ; Swap STATUS_TEMP register into W
                        ; (sets bank to original state)
MOVWF  STATUS          ; Move W into STATUS register
SWAPF  W_TEMP,F        ; Swap W_TEMP
SWAPF  W_TEMP,W        ; Swap W_TEMP into W
    
```

### 14.7 ウォッチドックタイマ (WDT)

適用デバイス							
710	71	711	72	73	73A	74	74A

ウォッチドックタイマはフリーランニング オンチップ RCオシレータで動作します。したがって外部部品は不要です。また、このウォッチドックタイマは、OSC1/CLKINに接続するRCオシレータとは別のオシレータです。したがって、CPUクロック (OSC1/OSC2に接続するオシレータ) のためのオシレータが停止しても動作します。一例として、SLEEP命令を実行すると、CPUクロックは停止しますが、WDTは動作を継続し、タイムアウトによってウエークアップ (ウォッチドックタイマ ウエークアップ) し、通常動作を開始します。通常動作中にWDTのタイムアウトが発生すると、デバイスをリセット (ウォッチドックタイマ タイムアウト) します。WDTは、コンフィギュレーションワードのWDTEビットをクリアすることでディセーブルできます。(14.1章を参照)

#### 14.7.1 ウォッチドッグタイマの周期

通常 (プレスケラが割り当てられていない時) の、ウォッチドックタイマのタイムアウト周期は18msです。このタイムアウト周期は、デバイス間や、温度、VDDによって変化します。(DCスペックを参照) もっとも長いタイムアウト

ト周期を必要とする場合は、OPTIONレジスタの設定で1:128の比率を設定します。これによって、約2.3秒のタイムアウト周期を実現できます。

CLRWDT、SLEEP命令は、WDTとポストスケラ (割り当てられている場合) をクリアし、タイムアウトによるリセット状態が発生することを防ぎます。

ウォッチドックタイマ タイムアウトによって、STATUSレジスタ内のTOビットがクリアされます。

#### 14.7.2 ウォッチドッグタイマ使用時の注意

最悪条件 (VDD = 最小、温度 = 最大、最大 WDT ポストスケラ) の条件下では、WDTのタイムアウトが起こるまで数秒かかる場合があることを計算にいれる必要があります。

**注意: 意図しないリセットを避けるためにプレスケラがWDTに割り当てられている時で、プレスケラ値を変更する場合、必ず、事前にCLEWDT命令を実行して下さい。**

図 14-23: ウォッチドッグタイマのブロック図

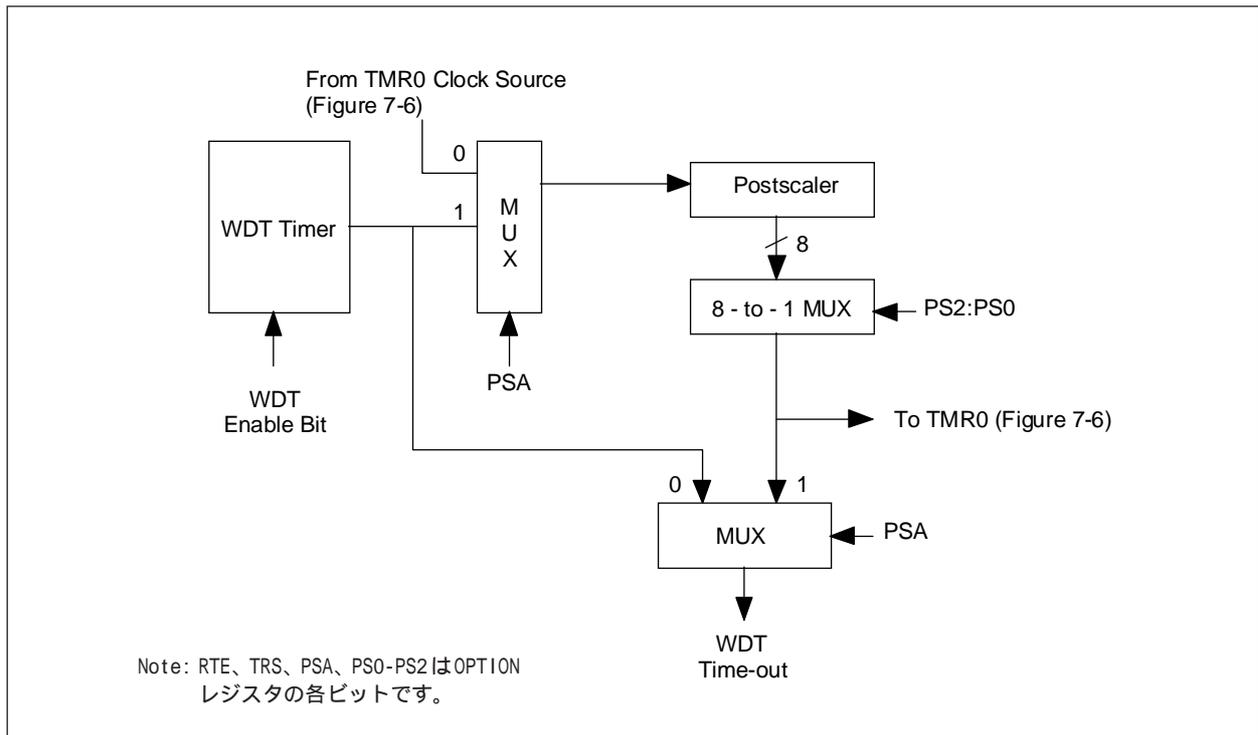


表 14-24: ウォッチドッグタイマレジスタのまとめ

アドレス	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
2007h	Config. bits	(1)	BODEN(1)	CP1	CP0	PWRTE(1)	WDTE	FOSC1	FOSCO
81h	OPTION	RBPU	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0

凡例: 影の部分は、ウォッチドックタイマでは使用しません。

注 1: これらのビットの操作は、図14-1,14-2,14-3,14-4を参照下さい。

### 14.8 パワーダウンモード (SLEEP)

適用デバイス							
710	71	711	72	73	73A	74	74A

SLEEP 命令を実行するとデバイスはパワーダウンモードに入ります。

WDT がイネーブルの時、ウォッチドックタイマーはクリアされますが実行は継続し、 $\overline{PD}$  ビット (STATUS<3>) はクリア、 $\overline{TO}$  ビット (STATUS<4>) はセット、外部オシレータは発振を停止します。I/Oポートは、SLEEP 命令を実行する以前の状態を維持します。(Hi または、Low を出力、またはハイインピーダンス) 低消費電力を実現するために、全ての出力ピンは外部回路からの電流の流入/流出がないようにHi または、Low に設定して下さい。また、内部A/D回路と外部クロックもオフして下さい。全ての入力ピンと $\overline{TOCKI}$ ピンは、フローティング状態による、スイッチング電流を防止するために、プルアップもしくは、プルダウンしてください。PORTBのオンチッププルアップも考慮に入れる必要があります。

MCLRピンは、ロジックハイレベル(VIHMC) にして下さい。

#### 14.8.1 SLEEPモードからのウエークアップ

以下の事象によってデバイスは、SLEEP からウエークアップします。

1. MCLRピンに、外部リセットを入力したとき。
2. WDTのタイムアウトの発生 (イネーブル時)
3. INTピンの入力、PORTBのレベル変化、いくつかの周辺割込み。

外部MCLRリセットは、デバイスリセットを発生させます。それ以外の全ての事象は、ウエークアップによって、引き続きプログラムの実行を継続します。STATUSピットの $\overline{TO}$ と $\overline{PD}$ によって、デバイスリセットの原因を知ることができます。

$\overline{PD}$  ビットはパワーアップ時にセットされ、SLEEP 命令実行時にクリアされます。 $\overline{TO}$  ビットは、WDT タイムアウトの発生でクリアされます。

以下の周辺割込みは、デバイスをSLEEPモードからウエークアップさせます。

1. TMR1割込み。(非同期カウンタモード時)
2. SSP (Start/Stop) ビット検出割込み。
3. スレープモード時の、SSP送信、受信。(SPI/I<sup>2</sup>C)
4. CCPキャプチャモード割込み。
5. パラレルスレープポートのリード/ライト。
6. A/D変換終了 (内部RCクロック使用時)。
7. スペシャルイベントトリガ (TMR1が非同期モードで、かつ、外部クロック使用時)
8. USARTのTX、RX。(同期スレープモード時)

他のペリフェラルは、SLEEPモード時、Qクロックが与えられないのでウエークアップすることができません。

SLEEP 命令実行中の時、 $\overline{PC+1}$  がプリフェッチされます。割り込みによってデバイスをウエークアップさせるために、対応するイネーブル・ビットをセット (イネーブルに) する必要があります。ウエークアップはGIEビットとは無関係です。GIEビットがクリア (ディセーブル) のとき、デバイスはSLEEP命令の後の命令を実行します。GIEビットがセット (イネーブル) のときデバイスはSLEEP命令の後の命令を実行し、割り込みアドレス (0004h) へ分岐します。SLEEPの後の命令の実行が影響する場合は、SLEEP命令の後にNOPを挿入する必要があります。

#### 14.8.2 割り込みを使用したウエークアップ

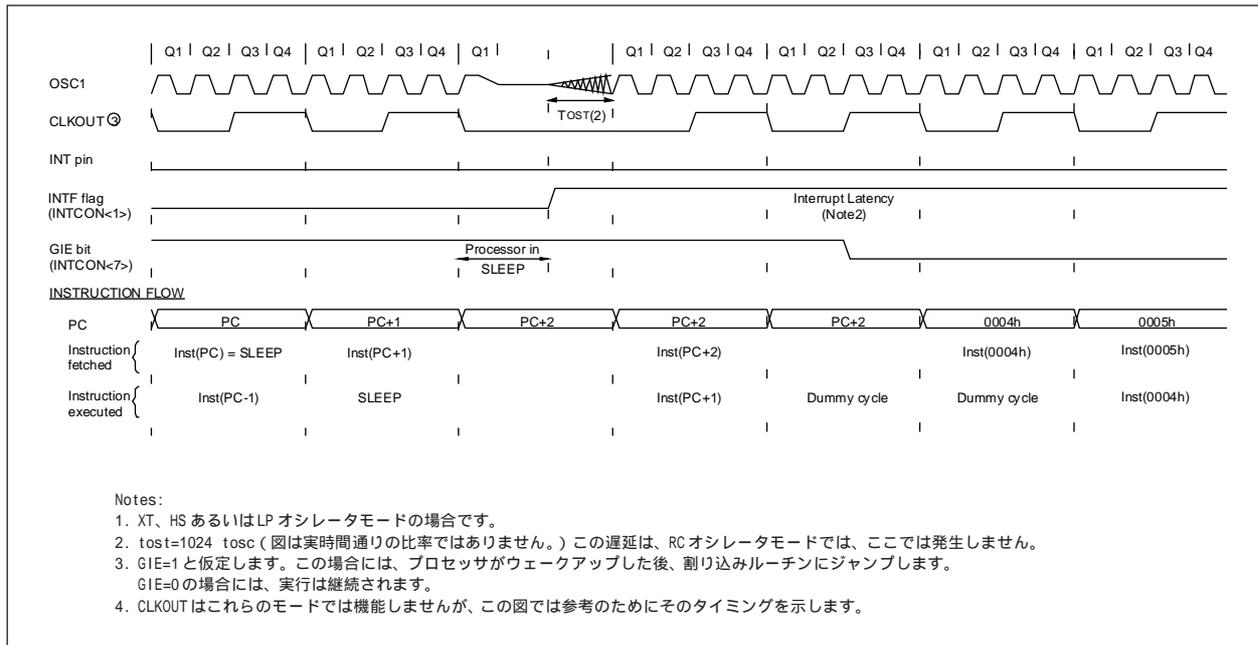
グローバルインターラプトがディセーブル (GIE=0) で、一つでも割り込みイネーブルと割り込みフラグビットの両方がセットされているとき、以下の状況が発生します。

- SLEEP 命令を実行する直前に、割り込みが発生した時、SLEEP 命令はNOP処理となり、そして、WDTのプレスケラはクリアされず、 $\overline{TO}$  ビットはセットされず、 $\overline{PD}$  ビットはクリアされません。
- SLEEP 命令を実行中または、実行直後、割り込みが発生した時、SLEEP 命令は正常に実行されますが、デバイスはすぐにウエークアップします。そして、WDTのプレスケラはクリアされ、 $\overline{TO}$  ビットはセットされ、 $\overline{PD}$  ビットはクリアされます。

SLEEP 命令が正常終了したか、NOP処理となったかは、 $\overline{PD}$  ビットを調べることで判断できます。もし、 $\overline{PD}$  ビットがセットされていたならば、SLEEP 命令はNOPとして処理されたことを意味します。

WDT がクリアされたことを保証するためには、SLEEP 命令を実行する前に、CLRWDTC命令を実行しなければなりません。

図14-25: インターラプトによるSLEEPモードからのウエークアップ



### 14.9 コードプロテクト/ プログラミングの検証

適用デバイス							
710	71	711	72	73	73A	74	74A

コードプロテクトビットをプログラムしない場合は、プログラミングの検証のために、プログラムメモリの内容を読み出すことができます。

Microchip社は、窓付きデバイスでのコードプロテクトは、保証いたしません。

### 14.10 IDロケーション

適用デバイス							
710	71	711	72	73	73A	74	74A

メモリーロケーションの 2000h-2003hは、IDロケーション用に用意されており、ユーザーは、ここにソフトウェアの認識番号や、チェックサムを書き込むことができます。このロケーションは、通常使用時はアクセスすることができませんが、プログラミング/ ベリファイモードの時のみ、読み出しや書き込みができます。IDロケーションの下位4ビットのみユーザは使用することができます。

### 14.11 インサーキット シリアル プログラミング

適用デバイス							
710	71	711	72	73	73A	74	74A

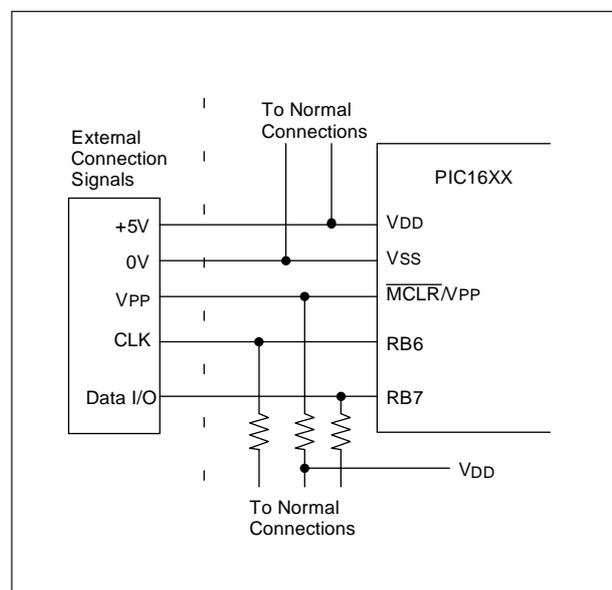
PIC16CXXマイクロコントローラは、PCボードの組み立て後もシリアルプログラミング機能を利用しユーザプログラムをプログラミングできます。

この方法は、簡単で、データとクロックラインの2本と、電源とグランドとプログラミング電圧の3本のラインで実現できます。ユーザーは、プログラミングされていないマイコンを使用しPCボードを組み立てておき、製品を出荷直前でコードをプログラミングします。これによって常に最新のソフトウェアの製品をエンドユーザに供給できます。また、特別仕様のソフトウェアにも柔軟に対応できます。

デバイスは、RB6/RB7ピンがLow状態で、MCLR (VPP)がVILからVIHHに立ち上がったとき、プログラミング/ベリファイモード(プログラミング仕様を参照)に移行します。RB6はプログラミングのためのクロック信号、RB7はプログラミングのためのデータ信号です。RB6/RB7は、プログラミング/ベリファイモードでは、シュミットトリガ入力になります。

プログラミング/ベリファイモード中に、リセットを行うとプログラムカウンタ(PC)は、00hになります。6ビットコマンドの後に14ビットのプログラムデータが続きます。この14ビットデータが、リードであるかライトであるかは、6ビットのコマンドによって決定されます。シリアルプログラミングの詳細を知りたい方は、PIC16C6X/7X Programming Specifications (#DS30228)をご参照下さい。

図14-26: インサーキットシリアルプログラミングの参考接続図



# 暫定版

英語最新版データシートと併用しご利用下さい。

AC/DC仕様は最新英文データシートをご参照下さい。

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

---

## PIC16C7X

### 15.0 命令セットの概要

適用デバイス							
710	71	711	72	73	73A	74	74A

PIC16CXXの命令はすべて14ビットワードで、命令の種類を指定するOPCODEの部分と、命令の実行方法を詳細に指定する1個または2個以上のオペランドの部分から成り立っています。表15-2には、PIC16CXXの命令セットの概要を、**バイト操作**、**ビット操作**、**リテラルおよびコントロール操作**に分けてリストしてあります。表15-1には、OPCODEフィールドの説明をまとめました。

**バイト操作命令**では、“f”をファイルレジスタの指名字、“d”を宛先の指名字として使います。ファイルレジスタ指名字では、命令で使用するファイルレジスタを指定します。

**宛先指定子**では、命令の実行結果を格納する場所を指定します。“d”がゼロの場合、結果はWレジスタに格納されます。“d”が1の場合、結果は命令で指定されたファイルレジスタに格納されます。

**ビット操作命令**では、ビットフィールド指名字“b”を使って、この命令実行によって影響を受けるビットの番号を選択します。また、ファイル番号指定子“f”を使って、そのビットが置かれているファイルの番号を指定します。

**リテラルおよびコントロール操作命令**では、“k”を使って8ビットまたは11ビットの定数やリテラル値を指定します。

表15-1: OPCODEフィールドの説明

フィールド	説明
f	レジスタファイルのアドレス (0x00 ~ 0x7F)
W	ワーキングレジスタ (アキュムレータ)
b	ファイルレジスタ内のビットアドレス
k	定数データ、または、ラベル(リテラルフィールド)
x	無効ロケーション (=0または1) アセンブラはx=0を伴うコードを生成。これはあらゆるソフトウェアツールとの互換性を確保するために推奨されている形式です。
d	宛先指定子; d=0ならば結果をWに格納、 d=1ならばファイルレジスタfに格納。 デフォルトはd=1
label	ラベル名
TOS	最上位スタック
PC	プログラムカウンタ
PCLATH	プログラムカウンタのハイラッチ
GIE	グローバル割込みイネーブルビット
WDT	ウォッチドッグタイマ/カウンタ
TO	タイムアウトビット
PD	パワーダウンビット
dest	宛先 (Wレジスタまたは指定レジスタファイルのロケーション)
[ ]	オプション
( )	内容
	割当て先
< >	レジスタビットフィールド
ε	セットを表わす。
イタリック	ユーザ定義用語

命令セットは非常に直交的で、次の3つの基本カテゴリーに分類されます。

- ・ バイト対応の命令
- ・ ビット対応の命令
- ・ リテラルおよびコントロール命令

すべての命令は基本的には1つのシングル命令サイクルの中で実行されますが、命令を実行した結果、条件付きテストの結果が真になったり、プログラムカウンタが変化したりすると、その命令の実行に2命令サイクルかかります。この場合、2番目のサイクルはNOPとして実行されます。1命令サイクルは4つのオシレータ周期から成ります。したがって、オシレータ周波数が4MHzの場合、通常の命令実行時間は1μsになります。命令を実行した結果、条件付きテストの結果が真になったり、プログラムカウンタが変化したりすると、命令実行時間は2μsになります。

表15-2に、MPASMアセンブラが認識する命令のリストを示します。

図15-1には、命令が取ることのできる3つの一般的なフォーマットを示します。

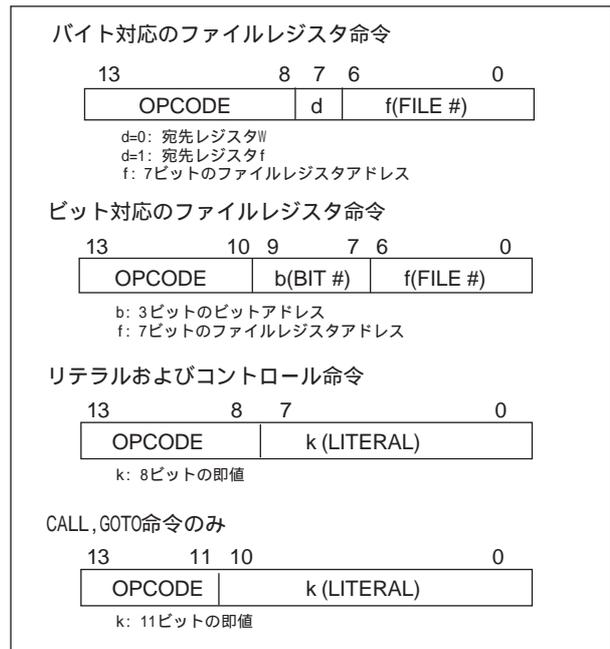
本書のすべての例では、次のフォーマットで16進数を表わします。

0xhh

注意: 将来のPIC16CXX製品との上位互換性を維持するために、OPTION命令とTRIS命令は使用しないでください。

上記フォーマットのhは16進数を表わします。

図15-1: 命令の一般的なフォーマット



# 暫定版

英語最新版データシートと併用しご利用下さい。

AC/DC仕様は最新英文データシートをご参照下さい。

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

表15-2: 命令セット

ニーモニック オペランド	説明	サイクル	14ビットのOPCODE		影響を受ける ステータス	注意事項
			msb	lsb		
<b>バイト対応のファイルレジスタ命令</b>						
ADDWF	f, d Add W and f	1	00	0111 dfff ffff	C, DC, Z	1, 2
ANDWF	f, d AND W and f	1	00	0101 dfff ffff	Z	1, 2
CLRF	f Clear f	1	00	0001 1fff ffff	Z	2
CLRWF	- Clear W	1	00	0001 0xxx xxxx	Z	
COMF	f, d Complement f	1	00	1001 dfff ffff	Z	1, 2
DECF	f, d Decrement f	1	00	0011 dfff ffff	Z	1, 2
DECFSZ	f, d Decrement f, Skip if 0	1 (2)	00	1011 dfff ffff		1, 2, 3
INCF	f, d Increment f	1	00	1010 dfff ffff	Z	1, 2
INCFSZ	f, d Increment f, Skip if 0	1 (2)	00	1111 dfff ffff		1, 2, 3
IORWF	f, d Inclusive OR W and f	1	00	0100 dfff ffff	Z	1, 2
MOVF	f, d Move f	1	00	1000 dfff ffff	Z	1, 2
MOVWF	f Move W to f	1	00	0000 1fff ffff		
NOP	- No Operation	1	00	0000 0xx0 0000		
RLF	f, d Rotate left f through carry	1	00	1101 dfff ffff	C	1, 2
RRF	f, d Rotate right f through carry	1	00	1100 dfff ffff	C	1, 2
SUBWF	f, d Subtract W from f	1	00	0010 dfff ffff	C, DC, Z	1, 2
SWAPF	f, d Swap halves f	1	00	1110 dfff ffff		1, 2
XORWF	f, d Exclusive OR W and f	1	00	0110 dfff ffff	Z	1, 2
<b>ビット対応のファイルレジスタ命令</b>						
BCF	f, b Bit Clear f	1	01	00bb bfff ffff		1, 2
BSF	f, b Bit Set f	1	01	01bb bfff ffff		1, 2
BTFSC	f, b Bit Test f, Skip if Clear	1 (2)	01	10bb bfff ffff		3
BTFSS	f, b Bit Test f, Skip if Set	1 (2)	01	11bb bfff ffff		3
<b>リテラルおよびコントロール命令</b>						
ADDLW	k Add literal to W	1	11	111x kkkk kkkk	C, DC, Z	
ANDLW	k AND literal to W	1	11	1001 kkkk kkkk	Z	
CALL	k Call subroutine	2	10	0kkk kkkk kkkk		
CLRWDT	- Clear watchdog timer	1	00	0000 0110 0100	$\overline{TO}$ , $\overline{PD}$	
GOTO	k Go to address	2	10	1kkk kkkk kkkk		
IORLW	k Inclusive OR literal to W	1	11	1000 kkkk kkkk	Z	
MOVLW	k Move literal to W	1	11	00xx kkkk kkkk		
RETFIE	- Return from interrupt	2	00	0000 0000 1001		
RETLW	k Return with literal in W	2	11	01xx kkkk kkkk		
RETURN	- Return from subroutine	2	00	0000 0000 1000		
SLEEP	- Go into standby mode	1	00	0000 0110 0011	$\overline{TO}$ , $\overline{PD}$	
SUBLW	k Subtract W from literal	1	11	110x kkkk kkkk	C, DC, Z	
XORLW	k Excl. OR literal to W	1	11	1010 kkkk kkkk	Z	

- 注意：1. I/Oレジスタがそれ自身の働きによって変更されると (MOVWF PORTB、1など)、ピン自体の上に存在する値が使用されることとなります。たとえば、入力として構成されているピンのデータラッチが“1”の場合、外部デバイスによってローに引かれると、データ“0”が書き戻されることとなります。
2. この命令をTMR0レジスタに対して実行すると (さらに、宛先dの指定が可能な場合にはd=1が指定されていると)、TMR0に割り当てられているプリスケアラがクリアされます (プリスケアラが割り当てられているときのみ)。
3. プログラムカウンタ (PC) が変化したり、条件付きテストの結果が真になったりすると、命令の実行に2サイクルがかかります。2番目のサイクルはNOPとして実行されます。

### 15.1 命令の説明

#### **ADDLW      Add Literal to W**

Syntax:            [ *label* ] ADDLW k  
Operands:        0   k   255  
Operation:        (W) + k    W  
Status Affected: C, DC, Z  
Encoding:        

11	111X	kkkk	kkkk
----	------	------	------

  
Description:      Wレジスタの内容を8ビットのリテラル“k”に加え、この結果をWレジスタに書き込みます。  
  
Words:            1  
Cycles:           1  
Example:          ADDLW      0x15  
                  命令実行前  
                  W =        0x10  
                  命令実行後  
                  W =        0x25

#### **ANDLW      AND Literal and W**

Syntax:            [ *label* ] ANDLW k  
Operands:        0   k   255  
Operation:        (W) .AND. (k)    W  
Status Affected: Z  
Encoding:        

11	1001	kkkk	kkkk
----	------	------	------

  
Description:      Wレジスタの内容と8ビットのリテラル“k”のANDを取り、結果をWレジスタに書き込みます。  
  
Words:            1  
Cycles:           1  
Example:          ANDLW      0x5F  
                  命令実行前  
                  W =        0xA3  
                  命令実行後  
                  W =        0x03

#### **ADDWF      ADD W to f**

Syntax:            [ *label* ] ADDWF f,d  
Operands:        0   f   127  
                  d ∈ [0,1]  
Operation:        (W) + (f)    (dest)  
Status Affected: C, DC, Z  
Encoding:        

00	0111	dfff	ffff
----	------	------	------

  
Description:      Wレジスタの内容をレジスタ“f”に加え、この結果を、d=0であればWレジスタに、d = 1 であればレジスタ“f”に書き戻します。  
  
Words:            1  
Cycles:           1  
Example:          ADDWF      FSR, 0  
                  命令実行前  
                  W =        0x17  
                  FSR =     0xC2  
                  命令実行後  
                  W =        0xD9  
                  FSR =     0xC2

#### **ANDWF      AND W with f**

Syntax:            [ *label* ] ANDWF f,d  
Operands:        0   f   127  
                  d ∈ [0,1]  
Operation:        (W) .AND. (f)    (dest)  
Status Affected: Z  
Encoding:        

00	0101	dfff	ffff
----	------	------	------

  
Description:      Wレジスタとレジスタ“f”のANDを取り、この結果を、d=0であればWレジスタに、d = 1 であればレジスタ“f”に書き戻します。  
  
Words:            1  
Cycles:           1  
Example:          ANDWF      FSR, 1  
                  命令実行前  
                  W =        0x17  
                  FSR =     0xC2  
                  命令実行後  
                  W =        0x17  
                  FSR =     0x02

# 暫定版

英語最新版データシートと併用にご利用下さい。

AC/DC仕様は最新英文データシートをご参照下さい。

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

<b>BCF</b>	<b>Bit Clear f</b>				
Syntax:	[ <i>label</i> ] BCF f,b				
Operands:	0 f 127 0 b 7				
Operation:	0 (f<b>)				
Status Affected:	None				
Encoding:	<table border="1"><tr><td>01</td><td>00bb</td><td>bfff</td><td>ffff</td></tr></table>	01	00bb	bfff	ffff
01	00bb	bfff	ffff		
Description:	レジスタ“f”のビット“b”を0にリセットします。				
Words:	1				
Cycles:	1				
Example:	BCF FLAG_REG, 7 命令実行前 FLAG_REG = 0x07 命令実行後 FLAG_REG = 0x47				

<b>BTFSC</b>	<b>Bit Test, skip if Clear</b>				
Syntax:	[ <i>label</i> ] BTFSC f,b				
Operands:	0 f 127 0 b 7				
Operation:	skip if (f<b>) = 0				
Status Affected:	None				
Encoding:	<table border="1"><tr><td>01</td><td>10bb</td><td>bfff</td><td>ffff</td></tr></table>	01	10bb	bfff	ffff
01	10bb	bfff	ffff		
Description:	レジスタ“f”のビット“b”が0であれば、次の命令を飛ばします。 ビット“b”が0であれば、現在の命令を実行中にフェッチされた次の命令が廃棄され、これを2サイクル命令にするためにNOPを実行します。				
Words:	1				
Cycles:	1(2)				
Example:	HERE BTFSC FLAG,1 FALSE GOTO PROCESS_CODE TRUE . . . 命令実行前 PC = HEREのアドレス 命令実行後 if FLAG<1> = 0, PC = TRUEのアドレス if FLAG<1> = 1, PC = FALSEのアドレス				

<b>BSF</b>	<b>Bit Set f</b>				
Syntax:	[ <i>label</i> ] BSF f,b				
Operands:	0 f 127 0 b 7				
Operation:	1 (f<b>)				
Status Affected:	None				
Encoding:	<table border="1"><tr><td>01</td><td>01bb</td><td>bfff</td><td>ffff</td></tr></table>	01	01bb	bfff	ffff
01	01bb	bfff	ffff		
Description:	レジスタ“f”のビット“b”を1にセットします。				
Words:	1				
Cycles:	1				
Example:	BSF FLAG_REG, 7 命令実行前 FLAG_REG = 0x0A 命令実行後 FLAG_REG = 0x8A				

# 暫定版

英語最新版データシートと併用ご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。  
<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

### **BTFSS** Bit Test, skip if Set

Syntax: [ *label* ] BTFSS *f*,*b*  
Operands: 0 *f* 127  
          0 *b* 7  
Operation: skip if (*f*<*b*>) = 1  
Status Affected: None  
Encoding: 

01	11bb	bfff	ffff
----	------	------	------

  
Description: レジスタ“*f*”のビット“*b*”が 1であれば、次の命令を飛ばします。  
          ビット“*b*”が 0であれば、現在の命令を実行中にフェッチされた次の命令が廃棄され、これを 2サイクル命令にするためにNOPを実行します。  
Words: 1  
Cycles: 1 (2)  
Example: HERE BTFSS FLAG,1  
          FALSE GOTO PROCESS\_CODE  
          TRUE .  
          .  
          .  
          命令実行前  
          PC = HEREのアドレス  
          命令実行後  
          if FLAG<1> = 0,  
          PC = FALSEのアドレス  
          if FLAG<1> = 1,  
          PC = TRUEのアドレス

### **CALL** Subroutine Call

Syntax: [ *label* ] CALL *k*  
Operands: 0 *k* 2047  
Operation: (PC) + 1 TOS,  
          *k* PC<10:0>,  
          (PCLATH<4:3>) PC<12:11>;  
Status Affected: None  
Encoding: 

10	0kkk	kkkk	kkkk
----	------	------	------

  
Description: サブルーチンコール。まずリターンアドレス (PC+1) をスタックに押し込み、11ビットの即値アドレスをPCビット<10:0>にロードします。その後、PCLATHをPCの上位ビットにロードします。CALLは2サイクルの命令です。  
Words: 1  
Cycles: 2  
Example: HERE CALL THERE  
          命令実行前  
          PC = HEREのアドレス  
          命令実行後  
          PC = THEREのアドレス  
          TOS = HEREのアドレス

### **CLRF** Clear *f*

Syntax: [ *label* ] CLRF *f*  
Operands: 0 *f* 127  
Operation: 00h (*f*)  
          1 Z  
Status Affected: Z  
Encoding: 

00	0001	1fff	ffff
----	------	------	------

  
Description: レジスタ“*f*”の内容をクリアし、Zビットをセットします。  
Words: 1  
Cycles: 1  
Example: CLRF FLAG\_REG  
          命令実行前  
          FLAG\_REG = 0x5A  
          命令実行後  
          FLAG\_REG = 0x00  
          Z = 1

### **CLRW** Clear W Register

Syntax: [ *label* ] CLRW  
Operands: None  
Operation: 00h (W)  
          1 Z  
Status Affected: Z  
Encoding: 

00	0001	0XXX	XXXX
----	------	------	------

  
Description: Wレジスタをクリアし、Zeroビット (Z) をセットします。  
Words: 1  
Cycles: 1  
Example: CLRW  
          命令実行前  
          W = 0x5A  
          命令実行後  
          W = 0x00  
          Z = 1

# 暫定版

英語最新版データシートと併用しご利用下さい。

AC/DC仕様は最新英文データシートをご参照下さい。

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

### CLRWDT Clear Watchdog Timer

Syntax: [ *label* ] CLRWDT  
Operands: None  
Operation: 00h WDT  
0 WDT prescaler,  
1  $\overline{TO}$   
1  $\overline{PD}$   
Status Affected:  $\overline{TO}$ ,  $\overline{PD}$   
Encoding: 

00	0000	0110	0100
----	------	------	------

  
Description: ウォッチドッグタイマをリセットし、そのW D T に付いているプリスケアラモリセットします。ステータスピット $\overline{TO}$ と $\overline{PD}$ をセットします。  
Words: 1  
Cycles: 1  
Example: CLRWDT  
命令実行前  
WDT counter = ?  
命令実行後  
WDT counter = 0x00  
WDT prescaler = 0  
 $\overline{TO}$  = 0  
 $\overline{PD}$  = 0

### COMF Complement f

Syntax: [ *label* ] COMF f,d  
Operands: 0 f 127  
 $d \in [0,1]$   
Operation: ( $\overline{f}$ ) (dest)  
Status Affected: Z  
Encoding: 

00	1001	dfff	ffff
----	------	------	------

  
Description: レジスタ“f”の内容の補数を取り、この結果を、d=0であればWレジスタに、d=1であればレジスタ“f”に書き戻します。  
Words: 1  
Cycles: 1  
Example: COMF REG1, 0  
命令実行前  
REG1 = 0x13  
命令実行後  
REG1 = 0x13  
W = 0xEC

### DECF Decrement f

Syntax: [ *label* ] DECF f,d  
Operands: 0 f 127  
 $d \in [0,1]$   
Operation: (f)-1 (dest)  
Status Affected: Z  
Encoding: 

00	0011	dfff	ffff
----	------	------	------

  
Description: レジスタ“f”の内容を減分(-1)し、この結果を、d=0であればWレジスタに、d=1であればレジスタ“f”に書き戻します。  
Words: 1  
Cycles: 1  
Example: DECF CNT, 1  
命令実行前  
CNT = 0x01  
Z = 0  
命令実行後  
CNT = 0x00  
Z = 1

### DECFSZ Decrement f, skip if 0

Syntax: [ *label* ] DECFSZ f,d  
Operands: 0 f 127  
 $d \in [0,1]$   
Operation: (f) - 1 d; skip if result = 0  
Status Affected: None  
Encoding: 

00	1011	dfff	ffff
----	------	------	------

  
Description: レジスタ“f”の内容を減分(-1)し、この結果を、d=0であればWレジスタに、d=1であればレジスタ“f”に書き戻します。  
結果が0の場合は、既にフェッチされている次の命令を廃棄し、2サイクル命令にするためにNOPを実行します。  
Words: 1  
Cycles: 1 (2)  
Example: HERE DECFSZ CNT, 1  
GOTO LOOP  
CONTINUE  
:  
:  
:  
命令実行前  
PC = HEREのアドレス  
命令実行後  
CNT = CNT - 1  
if CNT = 0, PC = CONTINUEのアドレス  
if CNT  $\neq$  0, PC = HERE+1のアドレス

# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。  
<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

### **GOTO**                      **Unconditional Branch**

Syntax:                    [ *label* ]    GOTO k  
Operands:                0    k    2047  
Operation:                k    PC<10:0>  
                              PCLATH<4:3>    PC<12:11>  
Status Affected: None  
Encoding:                

10	1kkk	kkkk	kkkk
----	------	------	------

  
Description:              無条件の分岐命令。まず11ビットの即値をPCビット<10:0>にロードし、次にPCLATH<4:3>をPCの上位ビットにロードします。GOTOは2サイクルの命令です。  
Words:                    1  
Cycles:                   2  
Example:                 GOTO THERE  
                              命令実行後  
                              PC =        THEREのアドレス

### **INCF**                      **Increment f**

Syntax:                    [ *label* ]    INCF f,d  
Operands:                0    f    127  
                              d ∈ [0,1]  
Operation:                (f) + 1    (dest)  
Status Affected: Z  
Encoding:                

00	1010	dfff	ffff
----	------	------	------

  
Description:              レジスタ“f”の内容を増分(+1)し、この結果を、d=0であればWレジスタに、d=1であればレジスタ“f”に書き戻します。  
Words:                    1  
Cycles:                   1  
Example:                 INCF                    CNT, 1  
                              命令実行前  
                              CNT =    0xFF  
                              Z =        0  
                              命令実行後  
                              CNT =    0x00  
                              Z =        1

### **INCFSZ**                    **Increment f, skip if 0**

Syntax:                    [ *label* ]    INCFSZ f,d  
Operands:                0    f    127  
                              d ∈ [0,1]  
Operation:                (f) + 1 → (dest), skip if result=0  
Status Affected: None  
Encoding:                

00	1111	dfff	ffff
----	------	------	------

  
Description:              レジスタ“f”の内容を増分(+1)し、この結果を、d=0であればWレジスタに、d=1であればレジスタ“f”に書き戻します。結果が0の場合は、既にフェッチされている次の命令を廃棄し、この代わりにNOPを実行します。INCFSZは2サイクルの命令です。  
Words:                    1  
Cycles:                   1 (2)  
Example:                 HERE                    INCFSZ    CNT, 1  
    GOTO        LOOP  
    CONTINUE  
    .  
                              命令実行前  
                              PC =        HEREのアドレス  
                              命令実行後  
                              CNT = CNT + 1  
                              if CNT = 0,  
                              PC =        CONTINUEのアドレス  
                              if CNT ≠ 0,  
                              PC =        HERE + 1のアドレス

### **IORLW**                    **Inclusive OR Literal with W**

Syntax:                    [ *label* ]    IORLW k  
Operands:                0    k    255  
Operation:                (W) .OR. (k)    (W)  
Status Affected: Z  
Encoding:                

11	1000	kkkk	kkkk
----	------	------	------

  
Description:              Wレジスタの内容と8ビットのリテラル“k”のORを取り、この結果をWレジスタに書き戻します。  
Words:                    1  
Cycles:                   1  
Example:                 IORLW                0x35  
                              命令実行前  
                              W =        0x9A  
                              命令実行後  
                              W =        0xBF  
                              Z =        0

# 暫定版

英語最新版データシートと併用しご利用下さい。

AC/DC仕様は最新英文データシートをご参照下さい。

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

<b>IORWF</b>	<b>Inclusive OR W with f</b>				
Syntax:	[ <i>label</i> ] IORWF f,d				
Operands:	0 f 127 d ∈ [0,1]				
Operation:	(W) .OR. (f) (dest)				
Status Affected:	Z				
Encoding:	<table border="1"><tr><td>00</td><td>0100</td><td>dfff</td><td>ffff</td></tr></table>	00	0100	dfff	ffff
00	0100	dfff	ffff		
Description:	Wレジスタとレジスタ“f”の包含的ORを取り、この結果を、d=0であればWレジスタに、d=1であればレジスタ“f”に書き戻します。				
Words:	1				
Cycles:	1				
Example:	IORWF RESULT, 0 Before Instruction RESULT = 0x13 W = 0x91 After Instruction RESULT = 0x13 W = 0x93 Z = 0				

<b>MOVF</b>	<b>Move f</b>				
Syntax:	[ <i>label</i> ] MOVF f,d				
Operands:	0 f 127 d ∈ [0,1]				
Operation:	(f) (dest)				
Status Affected:	Z				
Encoding:	<table border="1"><tr><td>00</td><td>1000</td><td>dfff</td><td>ffff</td></tr></table>	00	1000	dfff	ffff
00	1000	dfff	ffff		
Description:	レジスタ“f”の内容を宛先“d”に移動します。d=0であれば宛先はWレジスタです。d = 1 であれば宛先はファイルレジスタ“f”自体ですが、このときステータスフラグZ が変化するので、ファイルレジスタのテストに便利です。				
Words:	1				
Cycles:	1				
Example:	MOVF FSR, 0 命令実行後 W = value in FSR register Z = 1				

<b>MOVLW</b>	<b>Move Literal to W</b>				
Syntax:	[ <i>label</i> ] MOVLW k				
Operands:	0 k 255				
Operation:	k (W)				
Status Affected:	None				
Encoding:	<table border="1"><tr><td>11</td><td>00XX</td><td>kkkk</td><td>kkkk</td></tr></table>	11	00XX	kkkk	kkkk
11	00XX	kkkk	kkkk		
Description:	8ビットのリテラル“k”をWレジスタにロードします。 “xx”は“00”とアセンブルされます。				
Words:	1				
Cycles:	1				
Example:	MOVLW 0x5A 命令実行後 W = 0x5A				

<b>MOVWF</b>	<b>Move W to f</b>				
Syntax:	[ <i>label</i> ] MOVWF f				
Operands:	0 f 127				
Operation:	(W) (f)				
Status Affected:	None				
Encoding:	<table border="1"><tr><td>00</td><td>0000</td><td>1fff</td><td>ffff</td></tr></table>	00	0000	1fff	ffff
00	0000	1fff	ffff		
Description:	Wレジスタからレジスタ“f”にデータを移動します。				
Words:	1				
Cycles:	1				
Example:	MOVWF OPTION 命令実行前 OPTION = 0xFF W = 0x4F 命令実行後 OPTION = 0x4F W = 0x4F				

# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。  
<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

### NOP No Operation

Syntax: [ label ] NOP  
Operands: None  
Operation: No operation  
Status Affected: None  
Encoding: 

00	0000	0XX0	0000
----	------	------	------

  
Description: 何の操作も行われません。  
Words: 1  
Cycles: 1  
Example: NOP

### RETFIE Return from Interrupt

Syntax: [ label ] RETFIE  
Operands: None  
Operation: TOS PC,  
1 GIE  
Status Affected: None  
Encoding: 

00	0000	0000	1001
----	------	------	------

  
Description: 割込みからの復帰。まずスタックをポップアップし、次にスタックのトップ(TOS)をPCにロードします。その後、GIEビットをセットして割込みをイネーブルにします。GIEはグローバル割込みイネーブルビットです(INTCON<7>)。RETFIEは2サイクルの命令です。  
Words: 1  
Cycles: 2  
Example: RETFIE  
割込みから復帰後  
PC = TOS  
GIE = 1

### OPTION Load Option Register

Syntax: [ label ] OPTION  
Operands: None  
Operation: W OPTION  
Status Affected: None  
Encoding: 

00	0000	0110	0010
----	------	------	------

  
Description: Wレジスタの内容をOPTIONレジスタにロードします。これは、PIC16C5X製品とのコード互換性を保つために用意された命令です。OPTIONはリードとライトの両方が可能なレジスタで、ユーザはこのレジスタを直接アドレス指定できます。  
Words: 1  
Cycles: 1  
Example:

将来のPIC16CXX製品との上位互換性を維持するために、この命令は使わないでください。

### RETLW Return Literal to W

Syntax: [ label ] RETLW k  
Operands: 0 k 255  
Operation: k (W);  
TOS PC  
Status Affected: None  
Encoding: 

11	01xx	kkkk	kkkk
----	------	------	------

  
Description: まず8ビットのリテラル“k”をWレジスタにロードし、次にプログラムカウンタにスタックのトップ(リターンアドレス)をロードします。これは2サイクルの命令です。  
Words: 1  
Cycles: 2  
Example: CALL TABLE ;W contains table  
;offset value  
.  
;W now has table value  
.  
.  
TABLE ADDWF PC ;W = offset  
RETLW k1 ;Begin table  
RETLW k2 ;  
.  
.  
RETLW kn ;End of table  
命令実行前  
W = 0x07  
命令実行後  
W = value of k8

# 暫定版

英語最新版データシートと併用にご利用下さい。

AC/DC仕様は最新英文データシートをご参照下さい。

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

### RETURN Return from Subroutine

Syntax: [ *label* ] RETURN

Operands: None

Operation: TOS PC

Status Affected: None

Encoding: 

00	0000	0000	1000
----	------	------	------

Description: サブルーチンからの復帰。まずスタックをポップアップし、次にスタックのトップ (TOS) をプログラムカウンタにロードします。これは2サイクルの命令です。

Words: 1

Cycles: 2

Example: RETURN  
サブルーチンからの復帰後  
PC = TOS

### RRF Rotate Right f through Carry

Syntax: [ *label* ] RRF f,d

Operands: 0 f 127

$d \in [0,1]$

Operation: 下記参照

Status Affected: C

Encoding: 

00	1100	dfff	ffff
----	------	------	------

Description: レジスタ "f" の内容をキャリーフラグを通して1ビット右に回転します。この結果を、d=0であればWレジスタに、d=1であればレジスタ "f" に書き戻します。



Words: 1

Cycles: 1

Example: RRF REG1,0  
命令実行前  
REG1 = 1110 0110  
C = 0  
命令実行後  
REG1 = 1110 0110  
W = 0111 0011  
C = 0

### RLF Rotate Left f through Carry

Syntax: [ *label* ] RLF f,d

Operands: 0 f 127

$d \in [0,1]$

Operation: 下記参照

Status Affected: C

Encoding: 

00	1101	dfff	ffff
----	------	------	------

Description: レジスタ "f" の内容をキャリーフラグを通して1ビット左に回転します。この結果を、d=0であればWレジスタに、d=1であればレジスタ "f" に書き戻します。



Words: 1

Cycles: 1

Example: RLF REG1,0  
命令実行前  
REG1 = 1110 0110  
C = 0  
命令実行後  
REG1 = 1110 0110  
W = 1100 1100  
C = 1

### SLEEP

Syntax: [ *label* ] SLEEP

Operands: None

Operation: 00h WDT,  
0 WDT prescaler,  
1  $\overline{TO}$ ,  
0  $\overline{PD}$

Status Affected:  $\overline{TO}$ ,  $\overline{PD}$

Encoding: 

00	0000	0110	0011
----	------	------	------

Description: パワーダウンステータスビット ( $\overline{PD}$ ) をクリアし、タイムアウトステータスビット ( $\overline{TO}$ ) をセットし、ウォッチドッグタイマとそのプリスケラをクリアします。  
オシレータの停止と共に、プロセッサはSLEEPモードに入ります。詳しくはSLEEPモードの章をご覧ください。

Words: 1

Cycles: 1

Example: SLEEP

# 暫定版

英語最新版データシートと併用ご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。  
<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

### **SUBLW**      **Subtract W from Literal**

Syntax:            [ *label* ]    SUBLW k  
Operands:        0    k    255  
Operation:        k - (W)    (W)  
Status Affected: C, DC, Z

Encoding:        

11	110X	kkkk	kkkk
----	------	------	------

Description:      8ビットのリテラル“k”からWレジスタの内容を引きます。(2の補数法)この結果をWレジスタに書き込みます。

Words:            1

Cycles:           1

Example 1:        SUBLW    0x02  
                  命令実行前  
                  W    =    1  
                  C    =    ?  
                  命令実行後  
                  W    =    1  
                  C    =    1;結果は正

Example 2:        命令実行前  
                  W    =    2  
                  C    =    ?  
                  命令実行後  
                  W    =    0  
                  C    =    1;結果は0

Example3:        命令実行前  
                  W    =    3  
                  C    =    ?  
                  命令実行後  
                  W    =    0xFF  
                  C    =    0;結果は負

### **SUBWF**      **Subtract W from f**

Syntax:            [ *label* ]    SUBWF    f,d  
Operands:        0    f    127  
                  d ∈ [0,1]

Operation:        (f)-(W)    (dest)

Status Affected: C, DC, Z

Encoding:        

00	0010	dfff	ffff
----	------	------	------

Description:      レジスタ“f”からWレジスタの内容を引きます。(2の補数法)この結果を、d=0であればWレジスタに、d=1であればレジスタ“f”に書き込みます。

Words:            1

Cycles:           1

Example 1:        SUBWF    REG1,1  
                  命令実行前  
                  REG1    =    3  
                  W        =    2  
                  C        =    ?  
                  命令実行後  
                  REG1    =    1  
                  W        =    2  
                  C        =    1;結果は正

Example 2:        命令実行前  
                  REG1    =    2  
                  W        =    2  
                  C        =    ?  
                  命令実行後  
                  REG1    =    0  
                  W        =    2  
                  C        =    1;結果は0

Example 3:        命令実行前  
                  REG1    =    1  
                  W        =    2  
                  C        =    ?  
                  命令実行後  
                  REG1    =    0xFF  
                  W        =    2  
                  C        =    0;結果は負

# 暫定版

英語最新版データシートと併用にご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。

## PIC16C7X

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

<b>SWAPF</b>		<b>Swap Nibbles in f</b>		
Syntax:	[ <i>label</i> ]	SWAPF	f, d	
Operands:	0	f	127	
		d ∈ [0,1]		
Operation:	f<0:3>	d<4:7>	, f<4:7>	d<0:3>
Status Affected:	None			
Encoding:	00	1110	dfff	ffff
Description:	レジスタ“f”の上位ニブルビットと下位ニブルビットを入れ替えます。この結果を、d=0であればWレジスタに、d=1であればレジスタ“f”に書き込みます。			
Words:	1			
Cycles:	1			
Example:	SWAPF	REG, 0		
	命令実行前	REG =	0xA5	
	命令実行後	REG =	0xA5	
	W =		0x5A	

<b>XORLW</b>		<b>Exclusive OR literal with W</b>		
Syntax:	[ <i>label</i> ]	XORLW	k	
Operands:	0	f	255	
Operation:	(W)	.XOR.	k (W)	
Status Affected:	Z			
Encoding:	11	1010	kkkk	kkkk
Description:	Wレジスタの内容と8ビットのリテラル“k”とのXORを取り、この結果をWレジスタに書き込みます。			
Words:	1			
Cycles:	1			
Example:	XORLW	0xAF		
	命令実行前	W =	0xB5	
	命令実行後	W =	0x1A	

<b>TRIS</b>		<b>Load TRIS Register</b>		
Syntax:	[ <i>label</i> ]	TRIS	f	
Operands:	5	f	7	
Operation:	(W)	TRIS register	f;	
Status Affected:	None			
Encoding:	00	0000	0110	0fff
Description:	これは、PIC16C5X製品とのコード互換性を保つために用意された命令です。TRISレジスタはリードとライトの両方が可能で、ユーザはこのレジスタを直接アドレス指定できます。			
Words:	1			
Cycles:	1			
Example:				
	将来のPIC16CXX製品との上位互換性を維持するために、この命令は使わないでください。			

<b>XORWF</b>		<b>Exclusive OR W with f</b>		
Syntax:	[ <i>label</i> ]	XORWF	f, d	
Operands:	0	f	127	
		d ∈ [0,1]		
Operation:	(W)	.XOR.	(f) (dest)	
Status Affected:	Z			
Encoding:	00	0110	dfff	ffff
Description:	Wレジスタの内容とレジスタ“f”のXORを取り、この結果を、d=0であればWレジスタに、d=1であればレジスタ“f”に書き戻します。			
Words:	1			
Cycles:	1			
Example:	XORWF	REG, 1		
	命令実行前	REG =	0xAF	
	W =		0xB5	
	命令実行後	REG =	0x1A	
	W =		0xB5	

## 16.0 開発サポート

### 16.1 開発ツール

PIC16/17マイクロコントローラは、ハードウェアとソフトウェアの開発ツールによってサポートされています。

- PICMASTERリアルタイム・インサーキットエミュレータ
- PRO MATEユニバーサルプログラマ
- PICSTART低価格プロトタイププログラマ
- PICDEM-1低価格デモンストレーションボード
- PICDEM-2低価格デモンストレーションボード
- MPASMアセンブラ
- MPSIMソフトウェアシミュレータ
- Cコンパイラ(MP-C)
- ファジィロジック開発システム(fuzzyTECH-MP)

### 16.2 PICMASTER:

MPLAB IDEと高性能ユニバーサルインサーキットエミュレータ

PICMASTERユニバーサルインサーキットエミュレータは、製品開発エンジニアが、PIC16C5X、PIC16CXX、PIC17CXXファミリのすべてのマイクロコントローラのために、完全な設計を行えるように考えられています。PICMASTERはMPLAB開発統合環境(IDE)が与えられ、単一環境からの、編集、“製作”とダウンロード、ソースデバッグが可能となります。

交換可能なターゲットプローブにより異なるプロセッサのエミュレーションのために、システムを簡単に再構成することができます。PICMASTERのユニバーサルなアーキテクチャにより、すべての新しいPIC16C5X、PIC16CXX、PIC17CXXのマイクロコントローラをサポートするための拡張性を持っています。

PICMASTERエミュレータシステムは、より高級な開発製品によく見られるような最新の機能を持ったリアルタイムエミュレーションシステムとして設計されました。PCと互換性のある486(より高級) マシンプラットフォームとマイクロソフトWindows3.xの環境が、エンドユーザにこれらの特徴を最適に提供するために選ばれました。

PICMASTERユニバーサルエミュレータシステムは主に 4種類の主要部品で構成されています。

- ホストインターフェースカード
- エミュレータコントロールポッド
- ターゲット専用エミュレータプローブ
- PCホストエミュレーション制御ソフトウェア

Windowsオペレーティングシステムにより、設計者はPICMASTERシステムの多くの強力な特徴と機能をフルに利用することができます。

PICMASTERエミュレーションはひとつのウィンドウで操作でき、同時にテキストエディタを2番目のウィンドウで実行できます。

PCホストエミュレーション制御ソフトウェアは、Windowsの特徴であるダイナミックデータエクステンジ(DDE)をフルに利用します。DDEにより、2個以上のWindowsのプログラム間でデータを機能的に転送できます。この特徴により、PICMASTERで収集されたデータを後の解析のためにスプレッドシートやデータベースプログラムに自動的に転送できます。

Windowsのもとで、4個のPICMASTERエミュレータはすべて同じPCで同時に動作でき、マルチ-マイクロコントローラシステムの開発を可能にします(例えば、PIC16CXXプロセッサとPIC17CXXプロセッサを持っているシステム)。

PICMASTERプローブの規格を表16-1に示します。

# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。

## PIC16C7X

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

表16-1：開発ツール一覧

製品	MPLAB	MPLAB-C	MP-Driveway	fuzzyTECH Explorer/ Edition	PICMASTER	ICEPIC	PROMATEII	PICSTART Plus
PIC12C508, 509	SW007002	SW006005	-	-	EM167015	-	DV007003	DV003001
PIC14C000	SW007002	SW006005	-	-	EM147001	-	DV007003	DV003001
PIC16C52, 54, 54A, 55, 56, 57, 58A	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167015	EM167201	DV007003	DV003001
PIC16C55A, 556, 558	SW007002	SW006005	-	DV005001/ DV005002	EM167033	-	DV007003	DV003001
PIC16C62, 62A, 64, 64A	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167025	EM167203	DV007003	DV003001
PIC16C620, 621, 622	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167023	EM167202	DV007003	DV003001
PIC16C63, 65, 65A 73, 73A, 74, 74A	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167025	EM167204	DV007003	DV003001
PIC16C642, 662	SW007002	SW006005	-	-	EM1670365	-	DV007003	DV003001
PIC16C710, 711	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167027	-	DV007003	DV003001
PIC16C72	SW007002	SW006005	SW006006	-	EM167025	-	DV007003	DV003001
PIC16C84	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167029	EM167206	DV007003	DV003001
PIC16F83, 84	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167029	-	DV007003	DV003001
PIC16C923, 924	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167031	-	DV007003	DV003001
PIC17C42A, 43, 44	SW007002	SW006005	SW006006	DV005001/ DV005002	EM177007	-	DV007003	DV003001

MPLABには、アセンブラ、ソフトウェアシミュレータが含まれます。  
PROMATEII、量産用ライターには、別途、各デバイスパッケージに対応したソケットモジュールが必要です。  
PICMASTER (ICE)の動作電圧は4.5V-5.5V、最大動作周波数はEM167015, EM177007が20MHz、その他は10MHz。

### 16.3 PRO MATE:ユニバーサルプログラマ

PRO MATEユニバーサルプログラマはPCホストモードと、スタンドアロンモードで動作できる完全な機能を備えたプログラマです。

PRO MATEには、プログラマブルVDDとVPPの電源があり、信頼性を最大にするために、VDD minとVDD maxで、プログラムされたメモリを検査することができます。エラーメッセージを表示するLCDディスプレイ、コマンドを入力するキー、いろいろなパッケージタイプをサポートする取り外し可能なモジュラソケットがあります。

スタンドアロンモードでは、PRO MATEはPIC16C5X、PIC16CXX、PIC17CXXのデバイスをリード、ベリファイ、プログラムすることができます。さらに、このモードでコンフィギュレーションとコード保護のビットをセットすることができます。

PCホストモードでは、PRO MATEはCOMポート(RS-232)よりPCに接続します。PCベースのユーザーインターフェースソフトウェアによって、このプログラマが簡単に効果的に利用できます。このユーザーインターフェースはフルスクリーンでメニュー方式です。フルスクリーン表示やデータ編集、ビット構成とパーツ種類の容易な選択、VDD min、VDD max、VPPレベルの容易な選択、ディスクファイル(Intel<sup>®</sup>6進フォーマット)に対するロードとストアなどは、ソフトウェアの特徴の数々です。リード、ベリファイ、プログラム、ブランクチェックのような主要コマンドはスクリーンから実行できます。さらに、各パーツを異なったシリアル番号、シーケンシャル、ランダムにプログラムするために、シリアルプログラミングをサポートが可能です。

PRO MATEにはモジュラ、“プログラミングソケットモジュール”があります。異なったプロセッサタイプとパッケージタイプには、異なったソケットモジュールが必要です。

PRO MATEはすべてのPIC16C5X、PIC16CXXとPIC17CXXのプロセッサをサポートします。

### 16.4 PICSTART低価格開発システム

PICSTARTプログラマは使いやすく、大変低価格なプロトタイププログラマです。COMポート(RS-232)によりPCに接続します。PCベースのユーザーインターフェースソフトウェアによって、プログラマが簡単に効果的に利用できます。このユーザーインターフェースはフルスクリーンでメニュー方式です。**製造プログラミングではPICSTARTはお勧めしておりません。**

### 16.5 PICDEM-1 低価格PIC16/17 デモンストレーションボード

PICDEM-1は、Microchip社のマイクロコントローラの数々の能力を実演する簡単なボードです。そのマイクロコントローラは、PIC16C5X (PIC16C54からPIC16C58A)、PIC16C62X、PIC16C71X、PIC16F8X、PIC17C42、PIC17C43、PIC17C44をサポートしています。基本デモプログラムを動かすために必要なハードウェアとソフトウェアは、すべて含まれています。PRO MATEまたはPICSTART-16Bプログラマ上のPICDEM-1ボードで備えられたサンプルマイクロコントローラをプログラムすることができ、簡単にファームウェアをテストできます。さらにPICDEM-1ボードをPICMASTERエミュレータに接続でき、ファームウェアをテストのためにエミュレータにダウンロードすることもできます。その上プロトタイプエリアは、いくつかハードウェアを追加したりマイクロコントローラソケットに接続するのに有効です。いくつかの特徴はRS-232インターフェース、模擬アナログ入力のためのポテンショメータ、プッシュボタン・スイッチ、PORTBに接続された8個のLEDを含んでいます。

### 16.6 PICDEM-2低価格PIC16CXX

#### デモンストレーションボード

PICDEM-2は、PIC16C62、PIC16C64、PIC16C65、PIC16C73、PIC16C74のマイクロコントローラをサポートする簡単なデモンストレーションボードです。基本的なデモプログラムを動かすために必要なハードウェアとソフトウェアは、すべて含まれています。PRO MATEプログラマまたはPICSTART-16C上のPICDEM-2ボードで備えられたサンプルマイクロコントローラをプログラムすることができ、簡単にファームウェアをテストできます。さらにPICMASTERエミュレータはファームウェアをテストするためにPICDEM-2ボードとともに使用することもできます。その上プロトタイプ・エリアは、ハードウェアを追加したりマイクロコントローラソケットに接続するのに用意されています。いくつかの特徴はRS-232インターフェース、プッシュボタンスイッチ、模擬アナログ入力のためのポテンショメータ、I<sup>2</sup>Cバスの使用を実演するためのシリアルEEPROMを含み、LCDモジュールとキーパッドに接続するためにヘッダを分けます。

### 16.7 MPLAB 統合開発環境ソフトウェア

MPLABソフトウェアは、8ビットのマイクロコントローラ市場で以前には見られなかったソフトウェア開発に容易さをもたらしました。MPLABは、下記を含むWindowsの基本的なアプリケーションです。

- ・ 完全な編集機能
- ・ 3個の動作モード
  - 編集
  - エミュレータ
  - シミュレータ
- ・ プロジェクトマネージャ
- ・ カスタマイズできるツールバーとキーマッピング
- ・ プロジェクト情報のあるステータスバー
- ・ 広範囲なオンラインヘルプ

MPLABは次のようなことができます。

- ・ ソースファイルを編集(アセンブリまたは“C”)
- ・ ワンタッチでアセンブル(またはコンパイル)し、PIC16/17製品にダウンロードする(すべてのプロジェクト情報を自動的にアップデート)
- ・ デバッグの使用
  - ソースファイル
  - リスティングファイル
- ・ DDEによりダイナミックにデータを転送
- ・ 同一PC上で、4個までのエミュレータが動作
  - Microchip社のシミュレータをMPLABで使えることにより、一致したプラットフォームが可能となり、また開発ツールによる最小の再訓練で低価格のシミュレータから完全な機能を備えたエミュレータに容易に交換することができます。

### 16.8 アセンブラ(MPASM)

MPASMクロスアセンブラはPCホストのシンボリックアセンブラです。PIC16C5X、PIC16CXX、PIC17CXXファミリを含めたすべてのマイクロコントローラシリーズをサポートしています。

MPASMは完全な機能を備えたマクロ機能、条件アセンブリ、いくつかのソースとリストのフォーマットを提供します。サードパーティプログラマと同様に、Microchip社の開発ツールをサポートするために、いろいろなオブジェクトコードフォーマットを生成します。

MPASMによって、Microchipユニバーサルエミュレータシステム(PICMASTER)から完全なシンボリックデバッグが可能です。

特殊用途のアプリケーション用ソフトウェア開発を支援するために、MPASMは次のような特徴を有しています。

- ・ アセンブラソースコードをすべてのMicrochipマイクロコントローラのオブジェクトコードへ変換。
- ・ マクロアセンブリ可能。
- ・ Microchipエミュレータシステムにおいてシンボリックデバッグのために必要とするすべてのファイル(オブジェクト、リスティング、シンボル、特殊)を製作。
- ・ 16進(デフォルト)、10進、8進のソースとリストフォーマットをサポート。

MPASMには、PIC16/17のプログラミングをサポートする高級なディレクティブ言語があります。ディレクティブにより、アセンブルソースコードの開発がより短時間になり、より保持しやすくなります。

- ・ **データディレクティブ**はメモリの配置を制御します。それにより、記号的な項目つまり意味のある名前を使ってデータを参照できるようになります。
- ・ **制御ディレクティブ**はMPASMのリスティング表示を制御します。それによって、タイトルとサブタイトル、ページジェクトと他のリスト制御の規格を可能にします。これによりプリントされた出力ファイルが読み込みやすくなります。
- ・ **条件ディレクティブ**は条件アセンブルコードの部分を許可します。これは、製品によっては（機能が少ない低機種のほか、高級機種にも）付加機能を追加したい際に最も有効です。また、プログラムのデバッグにも大変役立ちます。
- ・ **マクロディレクティブ**はマクロの実行とマクロ本体のデータ配置を制御します。これにより、複数のプログラム間と同様に、1個のプログラムにおける機能の再使用が非常に簡単になります。

### 16.9 ソフトウェアシミュレータ(MPLAB-SIM)

MPSIMソフトウェアシミュレータにより、PCホスト環境でコード開発が可能になります。PIC16/17シリーズマイクロコントローラをインストラクションのレベルでシミュレートできます。どの与えられた命令でも、データ領域の検査や変更、またそのピンのどれにも外部励振ができます。入力/出力の基数をセットすることができ、シングルステップ、ブレークまで、またはトレースモードで、その実行を実現できます。MPLAB-SIMは

MPLAB-CとMPASMに使用するシンボリックデバッグを完全にサポートします。ソフトウェアシミュレータにより、充実したラボラトリでしかできなかったようなコードの開発とデバッグを低価格かつ柔軟に実行することが可能となる、優れたマルチプロジェクトのソフトウェア開発ツールが提供されます。

### 16.10 Cコンパイラ(MPLAB-C)

MP-Cコード開発システムは完全な'C'コンパイラであり、Microchip社のPIC16/17ファミリのマイクロコントローラのための統合開発環境を提供しています。そのコンパイラは強力な統合能力と他のコンパイラでは見られない簡単な使用を提供しています。

ソースレベルデバッグをより容易にするために、そのコンパイラはPICMASTERユニバーサルエミュレータのメモリ表示と互換性があるシンボル情報を提供します(PICMASTERエミュレータソフトウェアバージョン1.13とそれ以後)。

### 16.11 ファジィロジック開発システム (fuzzyTECH-MP)

fuzzyTECH-MPファジィロジック開発製品は、2つのバージョンを利用できます。1つは低価格入門版、MPエクスプローラで、ファジィロジックシステムデザインの広範囲な作業知識を得るためのデザイナー用です。もう1つは完全な機能を備えたバージョン、fuzzyTECH-MPで、もっと複雑なシステムを実行するために作られています。

両バージョンともファジィロジックシステムのインプリメンテーションを実地経験するためMicrochip社のfuzzyLABデモボードを含んでいます。

# 暫定版

英語最新版データシートと併用しご利用下さい。

AC/DC仕様は最新英文データシートをご参照下さい。

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

---

本資料では、159から266ページは省略。

# 暫定版

英語最新版データシートと併用しご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。  
<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

---

本資料では、283から310ページは省略。

# 暫定版

英語最新版データシートと併用しご利用下さい。

AC/DC仕様は最新英文データシートをご参照下さい。

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

---

本資料では、283から310ページは省略。

# 暫定版

英語最新版データシートと併用ご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。  
<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

## PIC16C7X

### 製品情報システム

ご注文に関する、価格、納期などのお問い合わせは、弊社製品取扱店までご連絡下さい。

PART NO.	-XX	X	/XX	XXX	
					パターン名: QTP, SQTP, ROMコードまたは、特別仕様
					パッケージ: JW = 窓付きCERDIP PQ = MQFP (Metric PQFP) TQ = TQFP (Thin Quad Flatpack) SO = SOIC SP = Skinny plastic carrier SJ = Skinny CERDIP P = PDIP L = PLCC
					温度範囲: - = 0 - 70 (T for Tape/Reel) I = -40 - +85 (S for Tape/Reel) E = -40 - +125
					動作周波数: 04 = 200KHz (PIC16C7X-04) 04 = 4MHz 10 = 10MHz 20 = 20MHz
					デバイス: PIC16C7X : VDD= 4.0 - 6.0V PIC16C7XT : VDD= 4.0 - 6.0V (Tape/Reel) PIC16LC7X : VDD= 2.5 - 6.0V PIC16LC7XT : VDD= 2.5 - 6.0V (Tape/Reel)

例

a) PIC16C711 -04/P 301  
コマーシャル温度  
PDIPパッケージ  
動作周波数 4 MHz  
標準電源電圧レンジ  
QTPパターン番号#301

b) PIC16LC73A -04I/SO  
インダストリアル温度  
SOICパッケージ  
動作周波数 4 MHz  
広域電源電圧レンジ

c) PIC16C74A -10E/P  
オートモーティブ温度  
PDIPパッケージ  
動作周波数 10MHz  
標準電源電圧レンジ

# 暫定版

英語最新版データシートと併用にご利用下さい。  
AC/DC仕様は最新英文データシートをご参照下さい。

## PIC16C7X

<http://www.microchip.com>より最新ERRATAシートを入手しご参照下さい。

### アメリカ

本社事務所  
Microchip Technology Inc.  
2355 West Chandler Blvd.  
Chandler, AZ 85224-6199  
Tel: 602 786-7200 Fax: 602 786-7277  
テクニカルサポート: 602 786-7627  
Web: <http://www.microchip.com>

### アトランタ

Microchip Technology Inc.  
500 Sugar Mill Road, Suite 200B  
Atlanta, GA 30350  
Tel: 770 640-0034 Fax: 770 640-0307

### ボストン

Microchip Technology Inc.  
5 Mount Royal Avenue  
Marlborough, MA 01752  
Tel: 508 480-9990 Fax: 508 480-8575

### シカゴ

Microchip Technology Inc.  
333 Pierce Road, Suite 180  
Itasca, IL 60143  
Tel: 708 285-0071 Fax: 708 285-0075

### ダラス

Microchip Technology Inc.  
14651 Dallas Parkway, Suite 816  
Dallas, TX 75240-8809  
Tel: 972 991-7177 Fax: 972 991-8588

### デイトン

Microchip Technology Inc.  
Suite 150  
Two Prestige Place  
Miamisburg, OH 45342  
Tel: 513 291-1654 Fax: 513 291-9175

### ロサンゼルス

Microchip Technology Inc.  
18201 Von Karman, Suite 1090  
Irvine, CA 92612  
Tel: 714 263-1888 Fax: 714 263-1338

### ニューヨーク

Microchip Technology Inc.  
150 Motor Parkway, Suite 416  
Hauppauge, NY 11788  
Tel: 516 273-5305 Fax: 516 273-5335

### サンノゼ

Microchip Technology Inc.  
2107 North First Street, Suite 590  
San Jose, CA 95131  
Tel: 408 436-7950 Fax: 408 436-7955

### トロント

Microchip Technology Inc.  
5925 Airport Road, Suite 200  
Mississauga, Ontario L4V 1W1, Canada  
Tel: 905 405-6279 Fax: 905 405-6253

### アジア / 太平洋

中国  
Microchip Technology  
Unit 406 of Shanghai Golden Bridge Bldg.  
2077 Yan'an Road West, Hongqiao District  
Shanghai, Peoples Republic of China  
Tel: 86 21 6275 5700  
Fax: 011 86 21 6275 5060

### 香港

Microchip Technology  
RM 3801B, Tower Two  
Metroplaza  
223 Hing Fong Road  
Kwai Fong, N.T. Hong Kong  
Tel: 852 2401 1200 Fax: 852 2401 3431

### インド

Microchip Technology  
No.6, Legacy, Convent Road  
Bangalore 560 025 India  
Tel: 91 80 526 3148 Fax: 91 80 559 9840

### 韓国

Microchip Technology  
168-1, Youngbo Bldg. 3 Floor  
Samsung-Dong, Kangnam-ku,  
Seoul, Korea  
Tel: 82 2 554 7200 Fax: 82 2 558 5934

### シンガポール

Microchip Technology  
200 Middle Road  
#10-03 Prime Centre  
Singapore 188980  
Tel: 65 334 8870 Fax: 65 334 8850

### 台湾・中華民国

Microchip Technology  
10F-1C 207  
Tung Hua North Road  
Taipei, Taiwan, ROC  
Tel: 886 2 7177175 Fax: 886 2 545 0139

### ヨーロッパ

イギリス  
Arizona Microchip Technology Ltd.  
Unit 6, The Courtyard  
Meadow Bank, Furlong Road  
Bourne End, Buckinghamshire SL8 5AJ  
Tel: 44 1628 850303  
Fax: 44 1628 850178

### フランス

Arizona Microchip Technology SARL  
Zone Industrielle de la Bonde  
2 Rue du Buisson aux Fraises  
91300 Massy - France  
Tel: 33 1 69 53 63 20  
Fax: 33 1 69 30 90 79

### ドイツ

Arizona Microchip Technology GmbH  
Gustav-Heinemann-Ring 125  
D-81739 Muenchen, Germany  
Tel: 49 89 627 144 0  
Fax: 49 89 627 144 44

### イタリア

Arizona Microchip Technology SRL  
Centro Direzionale Colleone Pas Taurus  
1Viale Colleoni 1  
20041 Agrate Brianza  
Milan Italy  
Tel: 39 39 6899939 Fax: 39 39 6899883

### 日本

マイクロチップテクノロジー  
インターナショナル Inc. 日本支社  
〒222神奈川県横浜市港北区  
新横浜3-18-20 Benex S-1 6F  
Tel: 045 471-6166 Fax: 045 471-6122

9/3/96



**MICROCHIP**

著作権等の権利はすべて保護されます。 1996, Microchip Technology Incorporated, USA

デバイスアプリケーション等に関する本書中の情報は提案として記載したもので、更新されたものが最新版となります。Microchip Technology Incorporatedは、本書中の情報の精度または用途について保証等しないほか、そのような用途での使用が特許権その他の知的所有権に抵触することになっても開知しません。生命サポートシステムの主要構成部品としてMicrochip製品を使用する場合は、Microchipの明示的な文書による承認が事前に必要です。いかなる知的所有権についても、暗黙にもそのいかなる方法でもライセンスを供与するものではありません。Microchipのロゴと名称はMicrochip Technology Inc.の登録商標です。著作権等の権利はすべて保護されます。本書に記載されているその他の商標はすべて、それぞれ各社の所有権です。