

18 ピン フラッシュ /EEPROM 8 ビット マイクロコントローラ

このデータシートに記載されているデバイス :

- PIC16F83
- PIC16F84
- PIC16CR83
- PIC16CR84
- 動作電圧範囲が広いデバイス (PIC16LF8X、PIC16LCR8X)

高性能 RISC CPU の特徴 :

- すべての命令は 1 ワード、命令数 : 35
- 2 サイクルのプログラム分岐命令を除いて、すべて 1 サイクル命令
- 動作速度 : DC - 10 MHz クロック入力
DC - 400 ns 命令サイクル

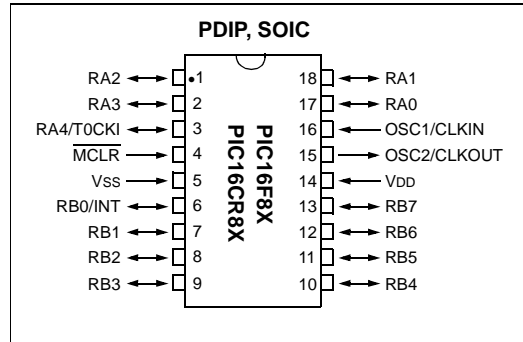
デバイス	プログラムメモリ (ワード)	データ RAM (バイト)	データ EEPROM (バイト)	最大周波数 (MHz)
PIC16F83	512 Flash	36	64	10
PIC16F84	1 K Flash	68	64	10
PIC16CR83	512 ROM	36	64	10
PIC16CR84	1 K ROM	68	64	10

- 14 ビット幅の命令
- 8 ビット幅のデータ
- 特殊機能レジスタ : 15
- ハードウェアスタック : 8 レベル
- 直接、間接、相対の各アドレッシングモード
- 割り込みソース :
 - 外部 RB0/INT ピン
 - TMR0 タイマーのオーバーフロー
 - PORTB<7:4> ピン変化による割り込み
 - データ EEPROM の書き込み完了
- フラッシュプログラムメモリの消去 / 書き込みサイクル : 1000 回
- EEPROM データメモリの消去 / 書き込みサイクル : 10,000,000 回
- EEPROM データ保持 > 40 年

周辺機能の特徴 :

- I/O ピン数 : 13 (ピンごとに入出力設定可能)
- シンク / ソース電流 (LED 直接駆動可能)
 - 1 ピンごとに最大シンク 25 mA
 - 1 ピンごとに最大ソース 20 mA
- TMR0 : 8 ビット タイマー / カウンタ プログラマブル 8 ビットプリスケアラ付き

ピン配置図



マイクロコントローラ特殊機能 :

- インサーキットシリアルプログラミング (ICSP™) (ROM デバイスではデータ EEPROM のみをサポート)
- パワーオンリセット (POR)
- パワーアップタイマー (PWRT)
- オシレータスタートアップタイマー (OST)
- ウォッチドッグタイマー (WDT) 専用のオンチップ RC オシレータ付き
- コードプロテクション
- スリープモード 消費電力を少なくできます
- オシレータオプション 選択可能です

CMOS フラッシュ /EEPROM テクノロジー :

- 低電力、高速テクノロジー
- 完全スタティック設計
- 幅広い動作電圧範囲 :
 - 商業用 : 2.0V to 6.0V
 - 工業用 : 2.0V to 6.0V
- 低消費電力 :
 - < 2 mA typ. @ 5V, 4 MHz
 - 15 μA typ. @ 2V, 32 kHz
 - スタンバイ電流 < 1 μA typ. @ 2V

PIC16F8X

目次

1.0 概要	3
2.0 PIC16F8X デバイスの種類	5
3.0 アーキテクチャの概要	7
4.0 メモリ構成	11
5.0 I/O ポート	21
6.0 タイマー 0 モジュールおよび TMR0 レジスタ	27
7.0 データ EEPROM メモリ	33
8.0 CPU の特殊機能	37
9.0 命令セット	53
10.0 開発サポート	69
11.0 電気的特性 - PIC16F83 および PIC16F84	73
12.0 電気的特性 - PIC16CR83 および PIC16CR84	85
13.0 DC および AC 特性のグラフ / 表	97
14.0 パッケージ	109
付録 A: PIC16C5X から PIC16F8X への機能改良点	113
付録 B: PIC16C5X と PIC16F8X とのコードの互換性	113
付録 C: このデータシートでの追加点	114
付録 D: このデータシートの変更点	114
付録 E: コンバージョンの注意点 - PIC16C84 から PIC16F83/F84 および PIC16CR83/F84 へ	115
索引	117
オンラインサポート	119
アンケートにお答えください	120
PIC16F8X の型番 / 品名	121
営業とサポート	121

お客様へ

当社では、当社製品やドキュメントの品質を高めるために常に努力をしております。このドキュメント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、多少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。より良いドキュメントを作るために皆様のご協力に感謝いたします。

PIC16F8X

1.0 概要

PIC16F8X は、低価格、高性能、CMOS、完全スタティック設計の 8 ビットマイクロコントローラ PIC16CXX ファミリーの 1 グループです。

このグループには、以下のデバイスが含まれます。

- PIC16F83
- PIC16F84
- PIC16CR83
- PIC16CR84

全ての PICmicro™ 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。PIC16F8X は、機能拡張したコア機能、8 レベルのスタック、複数の内部および外部の割り込みソースを持っています。ハーバードアーキテクチャは命令バスとデータバスが別なので、8 ビットデータが入る 14 ビット幅の命令ワードにすることができました。2 ステージ命令パイプラインによって、2 サイクルを必要とするプログラム分岐を除き、すべての命令が 1 サイクルで実行可能です。命令数は 35 です。(縮小命令セット) レジスタバンクが大きいので、高いパフォーマンスを達成できます。

PIC16F8X マイクロコントローラは、同クラスの他の 8 ビットマイクロコントローラと比較して、およそ 2:1 のコード圧縮、最大 4:1 のスピード向上 (20 MHz) を達成できます。

PIC16F8X は、最大 68 バイトの RAM、64 バイトのデータ EEPROM メモリ、13 本の I/O ピンを搭載しています。また、タイマー / カウンタも内蔵しています。

PIC16CXX ファミリーは、外付け部品点数を少なくすることにより、コスト削減、システム信頼性向上、低消費電力化できる特殊機能を備えています。4 種類のオシレータオプションは、低コストの RC オシレータ、低消費電力の LP オシレータ、標準クリスタルの XT、高速クリスタルの HS から選択できます。スリープ (パワーダウン) モードでは消費電力を少なくできます。内部、外部割り込みやリセットによりチップをスリープモードからウェークアップできます。

専用のオンチップ RC オシレータ付きの高信頼性ウォッチドッグタイマーを使用してソフトウェアのロックアップを防ぐことができます。

フラッシュプログラムメモリのデバイスは、同じパッケージを試作開発用と生産用の両方に使用することができます。インサーキットでの再プログラム書き込み可能なので、デバイスを製品から外さずにコードをアップデートできます。これは、デバイスを簡単に取り外しできないがコードの更新が必要なアプリケーションの試作品開発用に役立ちます。また、コードの更新が必要なリモートアプリケーションにも役立ちます。(課金情報など)

表 1-1 に、PIC16F8X の機能がまとめられています。図 3-1 は、PIC16F8X の簡単なブロック図です。

PIC16F8X は、高速自動車や家電製品のモーター制御から低電力リモートセンサー、電子錠、セキュリティデバイスおよびスマートカードにいたるまでのさまざま

なアプリケーションにフィットします。フラッシュ / EEPROM 技術により、アプリケーションプログラムのカスタマイズ (送信コード、モーターの速度、受信周波数、セキュリティコードなど) を非常に速く便利に行えます。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、省スペースの製品にも完璧にフィットします。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、PIC16F8X 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、シリアル通信、キャプチャ、コンペアおよび PWM 機能、コプロセッサ製品など) にも大きな能力を発揮します。

インサーキットシリアルプログラミング機能があるので、組み立て、試験終了後の製品をカスタマイズ可能です。製品に製造番号をつけたり、キャリブレーションデータを格納したり、出荷直前に最新ファームウェアをプログラム書き込みできます。

1.1 ファミリーや将来の製品との互換性

PIC16C5X ファミリーのマイクロプロセッサを使い慣れているユーザなら、この製品が PIC16C5X アーキテクチャの拡張バージョンであることをおわかりになるでしょう。機能拡張のリストについては付録 A を参照してください。PIC16C5X デバイス用に書かれたコードは、簡単に PIC16F8X デバイスに移植できます (付録 B)。

1.2 開発サポート

PIC16CXX ファミリーは、フル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ムライターおよびフル機能のプログラムライターによりサポートされています。また、C コンパイラやファジーロジックサポートツールもあります。

PIC16F8X

表 1-1 PIC16F8X ファミリーのデバイス

		PIC16F83	PIC16CR83	PIC16F84	PIC16CR84
クロック	最大動作周波数 (MHz)	10	10	10	10
	フラッシュプログラムメモリ (ワード)	512	-	1K	-
メモリ	EEPROM プログラムメモリ (ワード)	-	-	-	-
	ROM プログラムメモリ (ワード)	-	512	-	1K
	データメモリ (バイト)	36	36	68	68
周辺機能	データ EEPROM (バイト)	64	64	64	64
	タイマモジュール	TMRO	TMRO	TMRO	TMRO
特殊機能	割り込みソース	4	4	4	4
	I/O ピン	13	13	13	13
	動作電圧範囲 (ボルト)	2.0-6.0	2.0-6.0	2.0-6.0	2.0-6.0
	パッケージ	18 ピン DIP、 SOIC	18 ピン DIP、 SOIC	18 ピン DIP、 SOIC	18 ピン DIP、 SOIC

全ての PICmicro™ ファミリーデバイスは、パワー・オン・リセット、選択可能なウォッチドッグタイマ、選択可能なコードプロテクション、大きい I/O 電流駆動能力を持っています。

全ての PIC16F8X ファミリーデバイスは、クロックピン RB6 とデータピン RB7 を使用してシリアルプログラミングを行います。

2.0 PIC16F8X デバイスの種類

複数の周波数範囲およびパッケージからデバイスが選択できます。アプリケーションと量産数量に応じて、この項を参照して適切なデバイスを選択してください。ご注文の際は、このデータシートの最後にあるPIC16F8Xの型番 / 品名を参照して、正しい製品番号をご指定ください。

デバイスには4つのタイプがあり、デバイス番号の一部がタイプを示しています。

1. Fタイプ、PIC16F84、フラッシュプログラムメモリ搭載、標準的な電圧で動作
2. LFタイプ、PIC16LF84、フラッシュプログラムメモリ搭載、広範囲の電圧で動作
3. CRタイプ、PIC16CR83、ROMプログラムメモリ搭載、標準的な電圧で動作
4. LCRタイプ、PIC16LCR84、ROMプログラムメモリ搭載、広範囲の電圧で動作

メモリマップやその他のアーキテクチャについての説明に限り、F、CRと記述する場合、LF、LCRのバージョンも含まれます。

2.1 フラッシュデバイス

フラッシュデバイスは低コストのプラスチック製パッケージで提供されます。このデバイスは消去および再書き込みが可能なので、1つのデバイスを製品としてだけでなく、プロトタイプ開発およびパイロットプログラムにも使用することができます。

さらに、フラッシュバージョンは電氣的消去が可能ですので、インサーキットで、あるいはMicrochipのPICSTART(R) Plus プログラマやPRO MSTE (R) II プログラマなどのデバイス プログラマによって、消去および再書き込みができます。

2.2 クイック・ターンアラウンド・プロダクション (QTP) デバイス

マイクロチップテクノロジーは、量産ご注文に対してQTPプログラミングサービスを行っております。このサービスは、中 / 大量生産のためプログラム書き込みに工数をかけたくない、またコードパターンが一定している、という場合に便利です。デバイスのすべてのフラッシュロケーションとコンフィギュレーションオプションが、工場プログラムされます。コードの確認およびプロトタイプの動作確認を出荷前に行います。

QTPコード提出手順についての詳細は、マイクロチップ製品取扱店までお問い合わせください。

2.3 シリアル・クイック・ターンアラウンド・プロダクション (SQTP™) デバイス

マイクロチップは、各デバイスのシリアル番号をユーザが指定したプログラム番地にプログラムする独自のプログラミングサービスを行なっています。シリアル番号には、ランダム・疑似ランダム・連続番号があります。

シリアルプログラミングにより、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デバイスごとに異なる番号をプログラム書き込みすることができます。

SQTPコード提出手順の詳細は、マイクロチップ製品取扱店までお問い合わせください。

2.4 ROM デバイス

いくつかのマイクロチップのデバイスは、対応するプログラムメモリがROMのデバイスがあります。このタイプのデバイスは、他のマイクロチップのデバイス (EPROM、EEPROM) よりも低価格です。

ROMデバイス (PIC15CR8X) には、プログラムメモリにはデバイスごとに個別の情報はプログラム書き込みできません。データEEPROMに書き込むことは出来ます。

ROMコード提出手順の詳細は、マイクロチップ製品取扱店までお問い合わせください。

PIC16F8X

NOTES:

3.0 アーキテクチャ概要

PIC16CXX ファミリーの性能は、RISC 型マイクロプロセッサ特有のアーキテクチャによるものです。まず、PIC16CXX はハーバードアーキテクチャを採用しています。このアーキテクチャは、プログラムとデータが異なるメモリ領域を使ってアクセスされます。そのため、デバイスにはプログラムメモリバスとデータメモリバスが用意されています。これにより、プログラムとデータが（同一のバスでアクセスする）同一のメモリでフェッチされる従来型のフォン・ノイマン・アーキテクチャよりも処理能力が改善されています。更に、プログラムメモリとデータメモリを分離することで、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。PIC16CXX オペコードは命令を1ワードで構成可能な14ビット幅です。プログラムメモリバスが14ビット幅なので14ビットの命令を1サイクルでフェッチすることを可能にします。2ステージのパイプラインは、命令のフェッチと実行を同時実行します。（例 3-1）。したがって、すべての命令は、プログラム分岐を除き1サイクルで実行されます。

PIC16F83、PIC16CR83 のプログラムメモリは 512 x 14、PIC16F84、PIC16CR84 は 1K x 14 です。すべて内蔵のプログラムメモリです。

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可能です。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。非常に直交的（対照的）な命令セットにより、どのアドレッシングモードを使用してもすべてのレジスタにすべてのオペレーションが実行できます。このような対照的な性質と「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実現され、学習効率が極めて向上します。

PIC16CXX には、8ビット ALU およびワーキングレジスタがあります。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と全レジスタファイル間の計算とブール演算が実行できます。

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。特に指定がない限り、演算の実行は 2 の補数法で行なわれます。2 つのオペランドの命令では、一般的に 1 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 1 つはファイルレジスタまたはリテラルです。1 つ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になります。

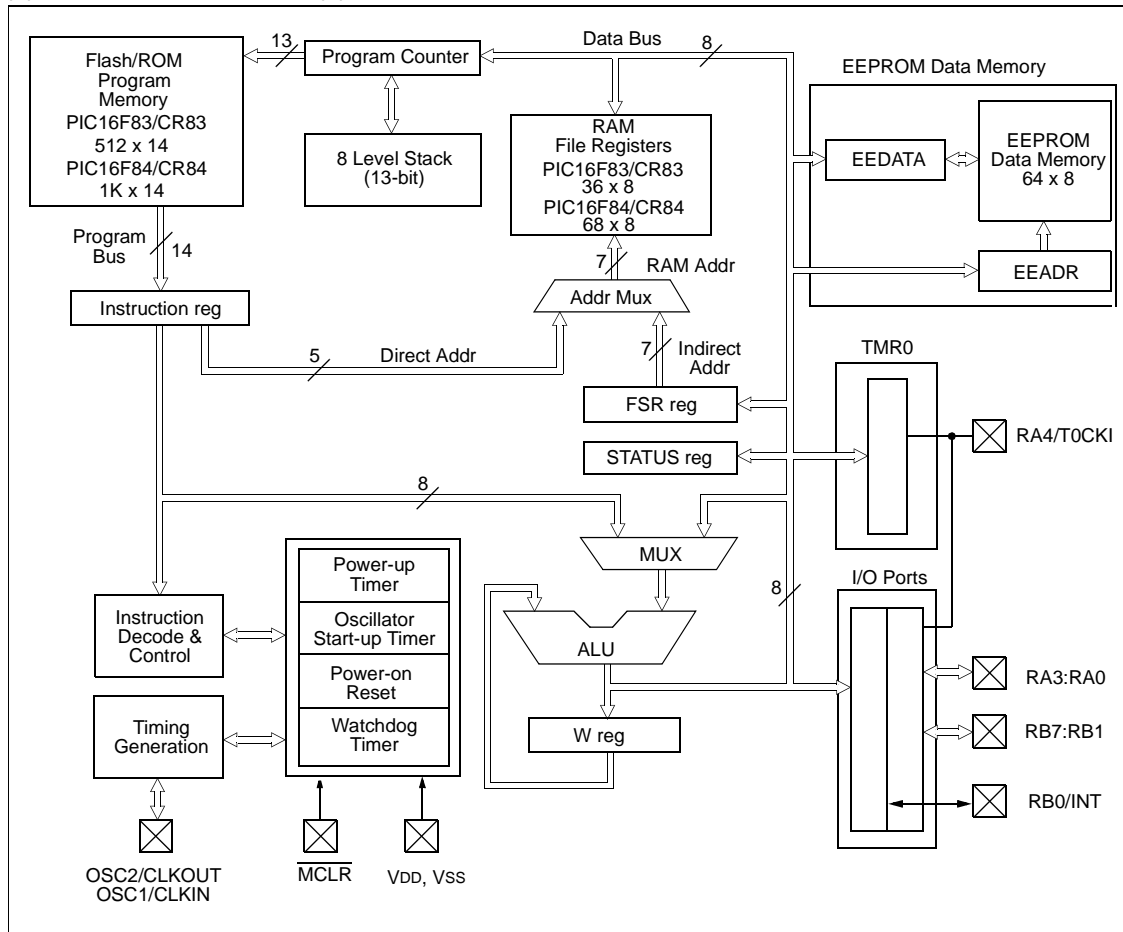
W レジスタは ALU の実行に使用される 8 ビットのワーキングレジスタで、アドレス指示できないレジスタです。

実行される命令によっては、ALU が STATUS レジスタのキャリー (C)、デジットキャリー (DC)、ゼロ (Z) ビットの値に影響を与えます。C と DC ビットはボービットとデジットボービットとして扱われます。一例として SUBLW と SUBWF 命令を参照してください。

PIC16F8X のブロック図を図 3-1 に、対応するピンを表 3-1 に示します。

PIC16F8X

図 3-1: PIC16F8X ブロック図



PIC16F8X

表 3-1 PIC16F8X ピンアウトの説明

ピンの名称	DIP No.	SOIC No.	I/O/P Type	Buffer Type	説明
OSC1/CLKIN	16	16	I	ST/CMOS ⁽³⁾	オシレータ水晶入力 / 外部クロックソース入力。
OSC2/CLKOUT	15	15	O	—	オシレータ水晶出力。水晶オシレータモード時に水晶またはセラミックレゾネータに接続。RC モードでは、OSC 1 の 1/4 の周波数の OSC2 ピン出力 CLKOUT、命令サイクルレートを示す。
MCLR	4	4	I/P	ST	マスター・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のアクティブ・ロー・リセットになります。
RA0 RA1 RA2 RA3 RA4/T0CKI	17 18 1 2 3	17 18 1 2 3	I/O I/O I/O I/O I/O	TTL TTL TTL TTL ST	PORTA は双方向 I/O ポートです。 TMRO タイマ / カウンタへのクロック入力として選択可能。出力はオープンドレインタイプ。
RB0/INT RB1 RB2 RB3 RB4 RB5 RB6 RB7	6 7 8 9 10 11 12 13	6 7 8 9 10 11 12 13	I/O I/O I/O I/O I/O I/O I/O	TTL/ST ⁽¹⁾ TTL TTL TTL TTL TTL TTL/ST ⁽²⁾ TTL/ST ⁽²⁾	PORTB は双方向 I/O ポートです。PORTB は全入力で内部弱プルアップがソフトウェアで選択可能です。 RB0/INT は外部割り込みピンとして選択可能です。 ピン変化による割り込み。 ピン変化による割り込み。 ピン変化による割り込み。シリアルプログラミングクロック。 ピン変化による割り込み。シリアルプログラミングデータ。
VSS	5	5	P	—	ロジックおよび I/O ピン用接地基準。
VDD	14	14	P	—	ロジックおよび I/O ピン用正極電源。

凡例： I = 入力, O = 出力, I/O = 入力 / 出力, P = 電力, — = なし, 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割り込み入力の場合はシュミットトリガ入力になります。

注 2: このバッファはシリアルプログラミングモードのときはシュミットトリガ入力になります。

注 3: このバッファは RC オシレータモードのときはシュミットトリガ入力になり、それ以外の時は CMOS 入力になります。

PIC16F8X

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 層の重ならない直角位相クロックを発生させます。内部で、プログラムカウンタ (PC) が Q1 毎にインクリメントされ、命令が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 の間に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 3-2 に示します。

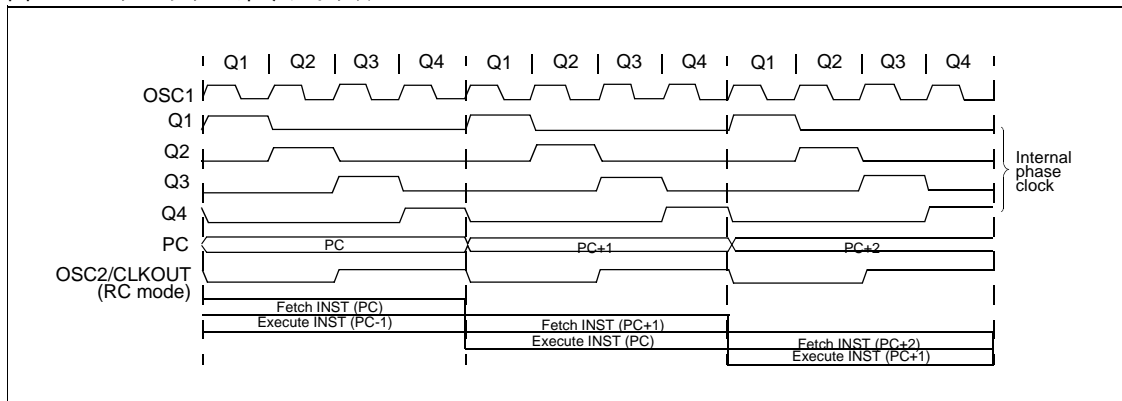
3.2 命令フロー/パイプライン

命令サイクルは、4 つの Q サイクル (Q1、Q2、Q3 および 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 命令サイクルかかります。しかし、パイプラインによって各命令は 1 サイクルで効率的に実行されます。命令によってプログラムカウンタに変更が生じる場合 (GOTO など) は、命令を実行するために 2 サイクルが必要です。(例 3-1 参照)

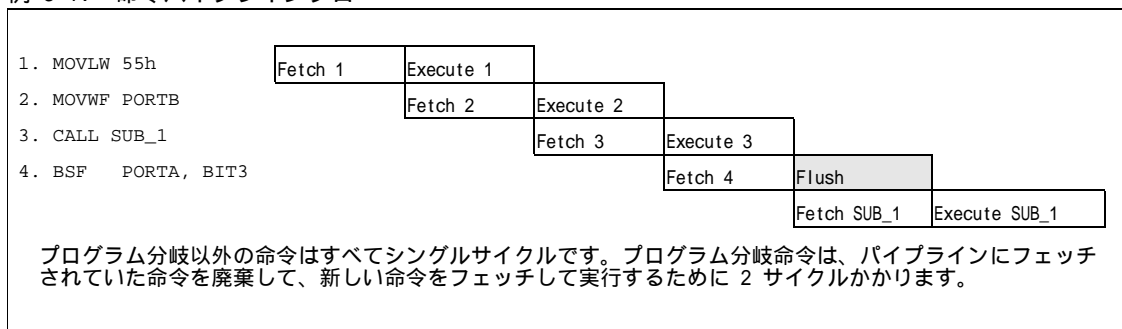
フェッチサイクルは、プログラムカウンタ (PC) が Q1 でインクリメントすることで始まります。

実行サイクルでは、フェッチされた命令はサイクル Q1 で命令レジスタ (IR) にラッチされています。この命令は、その後 Q2、Q3、Q4 サイクルの間に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 (オペランドのリード) され、Q4 の間でライト (結果格納先へのライト) されます。

図 3-2: クロック/命令サイクル



例 3-1: 命令パイプラインフロー



4.0 メモリ構成

PIC16F8X には 2 つのメモリ領域があります。プログラムメモリとデータメモリです。それぞれの領域には独自のバスがあるので、同じオシレータサイクルで各領域へアクセスすることが可能です。データメモリは更に汎用 RAM と特殊機能レジスタ (SFR) に分類されます。ここでは、"コア" を制御する SFR について説明します。周辺モジュールを制御する SFR については、各周辺モジュールに関する項で説明します。

データメモリ領域には、データ EEPROM メモリも含まれます。このメモリはデータメモリに直接アクセスできませんが間接的にアクセスできます。つまり、間接アドレスポインタでデータ EEPROM メモリのリード/ライトのためのアドレスを指定します。64 バイトのデータ EEPROM メモリのアドレス範囲は 0h から 3Fh までです。メモリの詳細については、7.0 項を参照してください。

4.1 プログラムメモリの構成

PIC16FXX には、8K x 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。PIC16F83 および PIC16CR83 の場合は、最初の 512 x 14 (0000h-01FFh) が物理的に存在します。(図 4-1)。PIC16F84 と PIC16CR84 の場合は、最初の 1K x 14 (0000h-03FFh) が物理的に存在します(図 4-2)。上記の領域を超えてアクセスした場合は、ラップアラウンドします。例えば、PIC16F84 のプログラム番地が 20h、420h、820h、C20h、1020h、1420h、1820h、1C20h の場合、すべて同じ命令となります。リセットベクタは 0000h、割り込みベクタは 0004h です。

図 4-1: プログラムメモリとスタック配置図 - PIC16F83/CR83

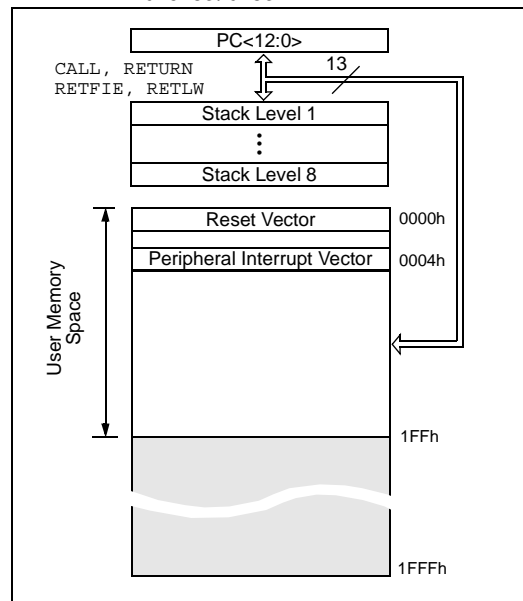
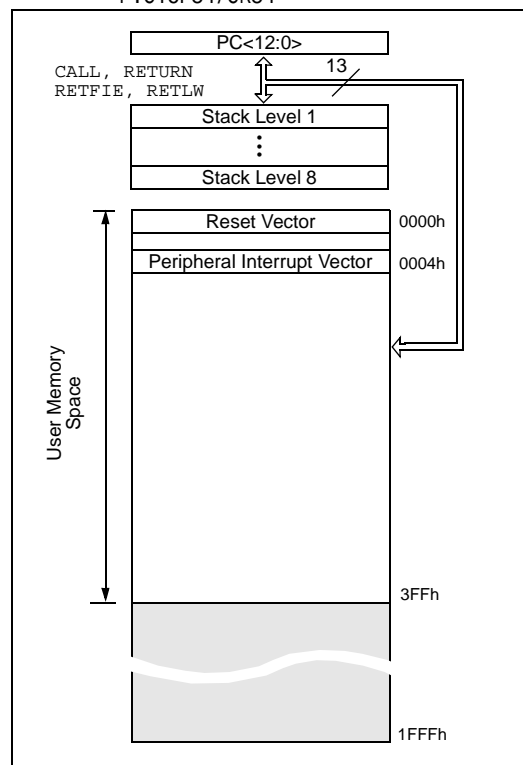


図 4-2: プログラムメモリとスタック配置図 - PIC16F84/CR84



PIC16F8X

4.2 データメモリ構成

データメモリは2つの領域に区分されます。1つは特殊機能レジスタ (SFR) 領域で、もう1つは汎用レジスタ (GPR) 領域です。SFR はデバイスの動作を制御します。

データメモリの SFR 領域も GPR 領域もバンク化されています。GPR 領域は、116 バイト以上の汎用 RAM を利用できるようにするためにバンク化されています。SFR のバンク化された領域は、周辺機能を制御するレジスタ用です。バンク切り替えにはバンク選択のための制御ビットが必要です。その制御ビットは、STATUS レジスタに配置されています。図 4-1 および図 4-2 に、データメモリマップの構成を示します。

命令 MOVWF および MOVF は、値を W レジスタからレジスタファイル ("F") の任意の位置に、またはその逆に移動させることができます。

データメモリのどこへでも、各レジスタファイルの絶対アドレスを使用して直接、またはファイルセレクトレジスタ (FSR) を通じて間接的にアクセスできます (4.5 項)。直接アドレッシングでは、データメモリのバンク化された領域へのアクセスに RP1:RP0 ビットの現在値が使用されます。

データメモリは2つの汎用レジスタおよび特殊機能レジスタがあるバンクに区分されます。バンク 0 は RP0 ビットをクリアすると選択されます (STATUS<5>)。RP0 ビットをセットすると、Bank 1 が選択されます。各バンクは最大 7Fh (128 バイト) まで拡張できます。各バンクの最初の 12 個のロケーションは、特殊機能レジスタのために予約されています。残りはスタティック RAM の汎用レジスタ用です。

4.2.1 汎用レジスタファイル

すべてのデバイスに汎用レジスタ (GPR) 領域があります。各 GPR は 8 ビット幅で、直接または FSR を通じて間接的にアクセスされます (4.5 項)。

バンク 1 の GPR アドレスは、バンク 0 のアドレスにマップされます。例えば、アドレス番地 0Ch と 8Ch は、同一の GPR にアクセスします。

4.2.2 特殊機能レジスタ

特殊機能レジスタ (図 4-1、図 4-2 および表 4-1) は CPU と周辺機能により使用され、デバイスの動作を制御します。このレジスタはスタティック RAM です。

特殊機能レジスタはコア機能と周辺機能の2種類に分類できます。コア機能に関連したレジスタは、この項で説明します。周辺機能の動作に関連したレジスタに関しては、各周辺機能の項に説明します。

PIC16F8X

図 4-1: レジスタファイル配置図 - PIC16F83/CR83

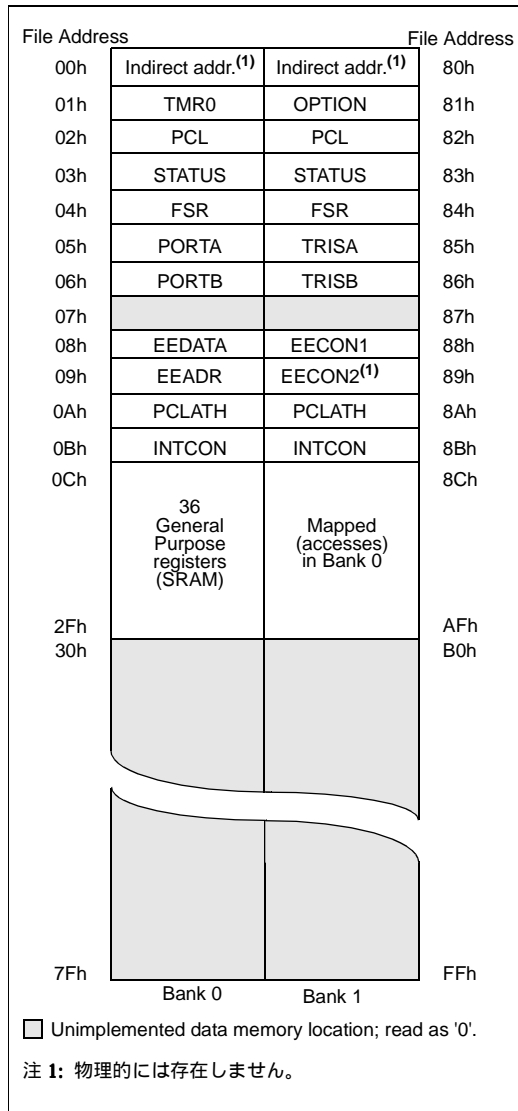
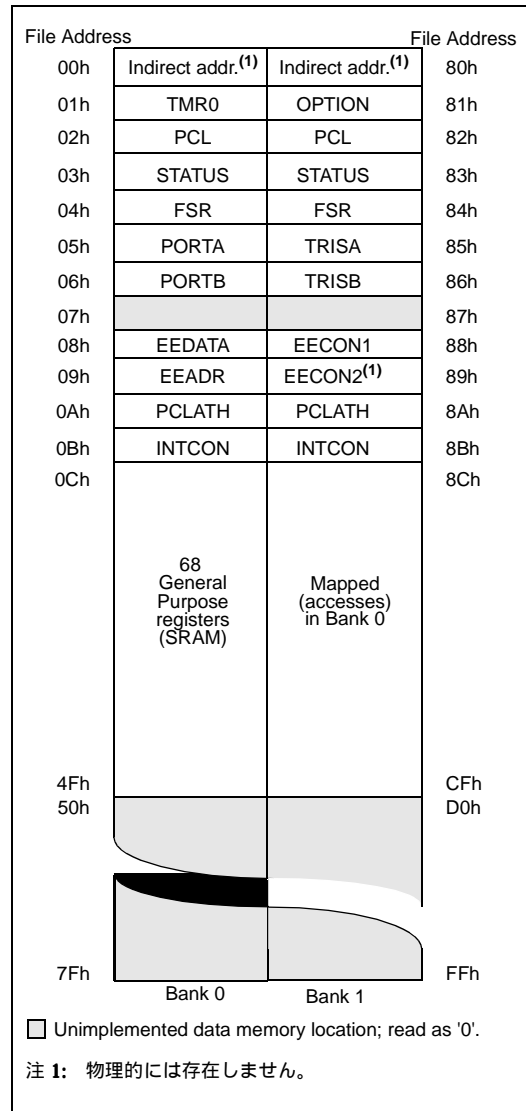


図 4-2: レジスタファイル配置図 - PIC16F84/CR84



PIC16F8X

表 4-1: レジスタファイル一覧

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Value on all other resets (Note3)		
Bank 0													
00h	INDF	FSR の内容のアドレスのデータメモリ (物理的には存在しない)								----	----		
01h	TMR0	8 ビットリアルタイム・クロック / カウンタ								xxxx	xxxx	uuuu	uuuu
02h	PCL	プログラムカウンタの下位 8 ビット								0000	0000	0000	0000
03h	STATUS ⁽²⁾	IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C	0001	1xxx	000q	quuu
04h	FSR	間接データメモリアドレスポインタ								xxxx	xxxx	uuuu	uuuu
05h	PORTA	—	—	—	RA4/TOCKI	RA3	RA2	RA1	RA0	---x	xxxx	---u	uuuu
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0/INT	xxxx	xxxx	uuuu	uuuu
07h		使用しない、「0」としてリードされる								----	----	----	----
08h	EEDATA	EEDATAEEPROM データレジスタ								xxxx	xxxx	uuuu	uuuu
09h	EEADR	EEDAREEPROM アドレスレジスタ								xxxx	xxxx	uuuu	uuuu
0Ah	PCLATH	—	—	—	PC の上位 5 ビットへの書き込みバッファ (1)					---	0000	---	0000
0Bh	INTCON	GIE	EEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000	000x	0000	000u
Bank 1													
80h	INDF	FSR の内容のアドレスのデータメモリ (物理的には存在しない)								----	----	----	----
81h	OPTION_REG	\overline{RBP}	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0	1111	1111	1111	1111
82h	PCL	プログラムカウンタ (PC) の下位 8 ビット								0000	0000	0000	0000
83h	STATUS ⁽²⁾	IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C	0001	1xxx	000q	quuu
84h	FSR	間接データメモリアドレスポインタ								xxxx	xxxx	uuuu	uuuu
85h	TRISA	—	—	—	PORTA データ入出力設定レジスタ					---	1111	---	1111
86h	TRISB	PORTB データ入出力設定レジスタ								1111	1111	1111	1111
87h		使用しない、「0」としてリードされる								----	----	----	----
88h	EECON1	—	—	—	EEIF	WRERR	WREN	WR	RD	---	x000	---	q000
89h	EECON2	EEPROM 制御レジスタ 2 (物理的には存在しない)								----	----	----	----
0Ah	PCLATH	—	—	—	PC の上位 5 ビットへの書き込みバッファ (1)					---	0000	---	0000
0Bh	INTCON	GIE	EEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000	000x	0000	000u

凡例: x = 不定、u = 不変、- = なし、「0」としてリードされる、q = 条件に応じて変化する値。

注 1: プログラムカウンタの上位バイトには直接アクセスできません。PCLATH は PC<12:8> のスレーブレジスタです。

PCLATH 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が、PC<12:8> の内容が PCLATH に転送されることはありません。

2: STATUS レジスタの TO と PD のステータスビットは、MCLR リセットによる影響を受けません。

3: 他の (電源 ON ではない) リセットには、MCLR ピン入力の外部リセット およびウォッチドッグ・タイマ・リセットがあります。

PIC16F8X

4.2.2.1 STATUS レジスタ

STATUS レジスタは、ALU の演算フラグ、リセット状態およびデータメモリのバンク選択ビットを含みます。

STATUS レジスタは他のレジスタと同様、命令の結果格納先になることができます。STATUS レジスタが Z、DC または C ビットに影響を与える命令の結果格納先になった場合は、これら 3 つのビットへの書き込みはできません。これらのビットはデバイスのロジックに従いセットまたはクリアされます。更に、TO と PD ビットには書き込みできません。したがって、STATUS レジスタを結果格納先とした命令の結果は、意図したものと異なる場合があります。

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 uluu (u = 不変) のままです。

したがって、BCF、BSF、SWAPF および MOVWF の命令だけを使用して STATUS レジスタを変更することを推奨します (表 9-2)。これらの命令はステータスビットに影響を与えないからです。

注意 1: IRP ビットおよび RP1 ビット (STATUS<7:6>) は、PIC16F8X では使用せず、クリアされているものとしてプログラミングされる必要があります。これらのビットを汎用 R/W ビットとして使用することは、将来の製品との互換性に影響するおそれがあるため、推奨できません。

注意 2: C ビットおよび DC ビットは、それぞれ減算で borrow および digit borrow out ビットとして動作します。使用例については、SUBWF 命令および SUBWF 命令を参照してください。

注意 3: STATUS レジスタが Z ビット、DC ビット、C ビットに影響を与える命令の結果格納先である場合、この 3 つのビットへの書き込みはできません。指定されたビットは、デバイスの論理に従って更新されます。

図 4-1: STATUS レジスタ (ADDRESS 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	TO	PD	Z	DC	C
bit 7						bit 0	
<p>R = リード可能ビット W = ライト可能ビット U = 使用しないビット、「0」としてリードされる - n = POR リセット時の値</p>							
<p>bit 7: IRP: レジスタバンク選択ビット (間接アドレス指定に使用) 0 = Bank 0, 1 (00h - FFh) 1 = Bank 2, 3 (100h - 1FFh) IRP ビットは、PIC16F8X では使用されません。IRP は常にクリアにしておいてください。</p>							
<p>bit 6-5: RPI:RP0: レジスタ・バンク選択ビット (直接アドレス指定に使用) 00 = バンク 0 (00h - 7Fh) 01 * バンク 1 (80h - FFh) 10 = バンク 2 (100h - 17Fh) 11 = バンク 3 (180h - 1FFh) 各バンクは 128 バイトです。ビット RP0 だけが PIC16F8X で使用されます。RPI は常にクリアにしておいてください。</p>							
<p>bit 4: TO: タイムアウトビット 1 = 電源 ON 後、CLRWT 命令または SLEEP 命令の実行後 0 = WDT タイムアウト発生</p>							
<p>bit 3: PD: パワーダウンビット 1 = 電源 ON 後または CLRWDT 命令による 0 = SLEEP 命令の実行による</p>							
<p>bit 2: Z: ゼロビット 1 = 計算またはロジック演算の結果がゼロ 0 = 計算またはロジック演算の結果がゼロでない</p>							
<p>bit 1: DC: デジットキャリー/ボロビット (ADDWF および ADDLW 命令用) (ボロの場合は極性が逆になります) 1 = 結果により下位 4 ビット目からキャリーが発生した 0 = 結果により下位 4 ビット目からキャリーが発生しなかった</p>							
<p>bit 0: C: キャリー/ボロビット (ADDWF および ADDLW 命令用) 1 = 結果により最上位ビットからキャリーが発生した 0 = 結果により最上位ビットからキャリーが発生しなかった 注: ボロの場合は極性が逆になります。2 番目のオペランドの 2 の補数を加算することにより減算が実行されます。 注: ローテート (RRF、RLF) 命令の場合は、このキャリービットはソースレジスタのビットを共にシフトされます。</p>							

PIC16F8X

4.2.2.2 OPTION_REG レジスタ

OPTION_REG レジスタは、PORTB に TMR0/WDT プリスケーラ、外部 INT 割り込み、TMR0、弱プルアップを設定するための各種コントロールビットを含む、リード / ライト可能なレジスタです。

注：プリスケーラが WDT (PSA = '1') に割り当てられている場合、TMR0 は 1:1 のプリスケーラ割り当てになります。

図 4-1 : OPTION レジスタ (ADDRESS 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
<div style="float: right; border: 1px solid black; padding: 2px; font-size: small;"> R = リード可能ビット W = ライト可能ビット U = 使用しないビット、「0」としてリードされる - n = POR リセットによる値 </div>								
bit 7:	RBPU: PORTB プルアップイネーブルビット 1 = PORTB プルアップを使用しない 0 = PORTB プルアップを使用する							
bit 6:	INTEDG: 割り込みエッジ選択ビット 1 = RB0/INT ピンの立ち上がりエッジにより割り込み 0 = RB0/INT ピンの立ち下がりエッジにより割り込み							
bit 5:	TOCS: TMR0 クロックソース選択ビット 1 = RA4/TOCKI ピンの入力 0 = 内部命令サイクルクロック (CLKOUT)							
bit 4:	TOSE: TMR0 ソースエッジ選択ビット 1 = RA4/TOCKI ピンの入力が High から Low でインクリメント 0 = RA4/TOCKI ピンの入力が Low から High でインクリメント							
bit 3:	PSA: プリスケーラ割り当てビット 1 = プリスケーラは WDT へ割り当て 0 = プリスケーラは TMR0 へ割り当て							
bit 2-0:	PS2:PS0: プリスケーラレート選択ビット							
	ビット値	TMR0 レート	WDT レート					
	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					

4.2.2.3 INTCON レジスタ

INTCON レジスタは、すべての割り込みソースのイネーブルビットを含む、リード / ライト可能なレジスタです。

注：割り込みフラグビットは割り込み条件が発生した場合に対応するイネーブルビットまたはグローバル割り込みイネーブルビット GIE (INTCON<7>) の状態に関係なくセットされます。

図 4-1: INTCON レジスタ (ADDRESS 0Bh, 8Bh)

	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
	GIE	EEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF
bit 7								bit0
bit 7:	GIE: グローバル割り込みイネーブルビット 1 = すべてのマスクされていない割り込み発生を許可する 0 = 全ての割り込み発生を禁止する 注: 割り込みの動作については、8.5 項を参照。							
bit 6:	EEIE: EE ライト完了割り込みイネーブルビット 1 = EE ライト完了割り込み発生を許可する 0 = EE ライト完了割り込み発生を禁止する							
bit 5:	TOIE: TMR0 オーバーフロー割り込みイネーブルビット 1 = TMR0 割り込み発生を許可する 0 = TMR0 割り込み発生を禁止する							
bit 4:	INTE: RB0/INT 割り込みイネーブルビット 1 = RB0/INT 割り込み発生を許可する 0 = RB0/INT 割り込み発生を禁止する							
bit 3:	RBIE: RB ポート変化割り込みイネーブルビット 1 = RB ポート変化割り込み発生を許可する 0 = RB ポート変化割り込み発生を禁止する							
bit 2:	TOIF: TMR0 オーバーフロー割り込みフラグビット 1 = TMR0 がオーバーフローした (ソフトウェアでクリア要) 0 = TMR0 がオーバーフローしていない							
bit 1:	INTF: RB0/INT 割り込みフラグビット 1 = RB0/INT 割り込みが発生した (ソフトウェアでクリア要) 0 = RB0/INT 割り込みが発生していない							
bit 0:	RBIF: RB ポート変更割り込みフラグビット 1 = 少なくとも 1 つ以上の RB7:RB4 ピンの状態が変化した (ソフトウェアでクリア要) 0 = 状態が変化した RB7:RB4 ピンはない							

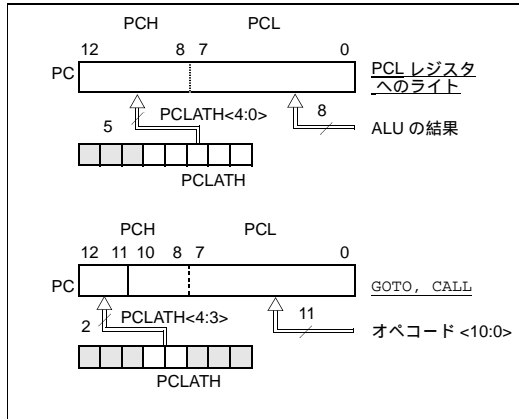
R = リード可能ビット
 W = ライト可能ビット
 U = 使用しないビット、「0」としてリードされる
 - n = POR リセットによる値

PIC16F8X

4.3 プログラムカウンタ：PCL と PCLATH

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 レジスタで、リード / ライト可能なレジスタです。PC (PC<12:8>) の上位バイトは、直接リード / ライトを行うことができず、PCLATH レジスタを使用します。PCLATH (PC latch high) レジスタは、PC<12:8> 用のレジスタです。PC に新しい値がロードされる時 (CALL、GOTO または PCL へのライトの時)、PCLATH の内容がプログラムカウンタの上位バイトに転送されます。PC の上位ビットは、図 4-1 に示す PCLATH からロードされます。

図 4-1: 各状態での PC のロード方法



4.3.1 計算 GOTO

計算 GOTO は、プログラムカウンタにオフセットを加算することによって実行されます (ADDWF PCL)。計算 GOTO を使用してテーブルリードを実行する時、テーブルのプログラムメモリ番地が PCL メモリの境界 (256 ワードごとの境界) を越える場合、注意が必要です。アプリケーションノート「Implementing a Table Read」(AN556) を参照してください。

4.3.2 プログラムメモリページング

PIC16F83 および PIC16CR83 には 512 ワードのプログラムメモリがあります。PIC16F84 および PIC16CR84 には 1K ワードのプログラムメモリがあります。CALL 命令および GOTO 命令には 11 ビットのアドレス指定ができます。この 11 ビットのアドレス範囲によって、2K のプログラムメモリのページサイズ内でのプログラム分岐ができます。将来の PIC16F8X プログラムメモリ拡張用にプログラムメモリページ指定用に 2 ビットがあります。そのページングビットは、PCLATH<4:3> ビットです。(図 4-1)。CALL 命令または GOTO 命令を実行する場合、ユーザはこれらのページビット (PCLATH<4:3>) が分岐先のプログラムメモリページになっていることを確認する必要があります。CALL 命令 (または割り込み) が実行される場合、13 ビットの PC 全体がスタックに 'プッシュ' されます (次の項

を参照)。これによって、リターン命令 (スタックから PC を "ポップ" する) には PCLATH<4:3> の操作は不要です。

注: PIC16F8X は、プログラムメモリページ 1、2 および 3 (0800h - 1FFFh) に使用される PCLATH<4:3> ビットを無視します。PCLATH<4:3> を汎用 R/W ビットとして使用することは、将来の製品との互換性に影響する可能性があるため、推奨できません。

4.4 スタック

PIC16FXX 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 (図 4-1)。スタック領域はプログラム領域やデータ領域の一部ではありません。スタック・ポインタはリード / ライトできません。

CALL 命令が実行された時や割り込みが発生した時、13 ビット PC 全体がスタックに "プッシュ" されます。スタックは、RETURN、RETLW または RETFIE 命令の実行により "ポップ" されます。PCLATH はプッシュまたはポップの影響を受けません。

注: プッシュまたはポップと呼ばれる命令のモニタリングはありません。これらは CALL、RETURN、RETLW、RETFIE などの命令の実行、または割り込み発生の際の動作です。

スタックは循環バッファとして動作します。すなわち、スタックが 8 回プッシュされると、9 回目のプッシュでは 1 回目のプッシュで格納された値が上書きされます。10 回目のプッシュでは 2 回目のプッシュが上書きされます (以下同様に続きます)。スタックが有効に 9 回ポップされると、PC 値は 1 回目のポップと同じになります。

注: スタック・オーバーフローやスタック・アンダースローを報告するステータスビットはありません。

4.5 間接データアドレッシング： INDF および FSR レジスタ

INDF は物理的には存在しないレジスタです。INDF へのアクセスは、実際には FSR レジスタ (FSR はポインタ) の内容のアドレスのレジスタをアクセスします。これが間接アドレッシングです。

例 4-1: 間接アドレッシング

- レジスタ番地 05 には値 10h が入っています
- レジスタ番地 06 には値 0Ah が入っています
- 値 05 を FSR レジスタにロードします。
- INDF をリードすると 10h がリードされます。
- FSR の値を インクリメントします (FSR = 06)
- INDF をリードすると 0Ah がリードされます。

INDF 自体を間接的にリードすると (FSR = 0)、00h がリードされます。INDF レジスタへ間接的に書き込むと、ノーオペレーションになります。(ただし、STATUS ビットは影響を受けることがあります。)

間接アドレッシングを使用して RAM 番地 20h-2Fh を初期化する簡単なプログラムを例 4-2 に示します。

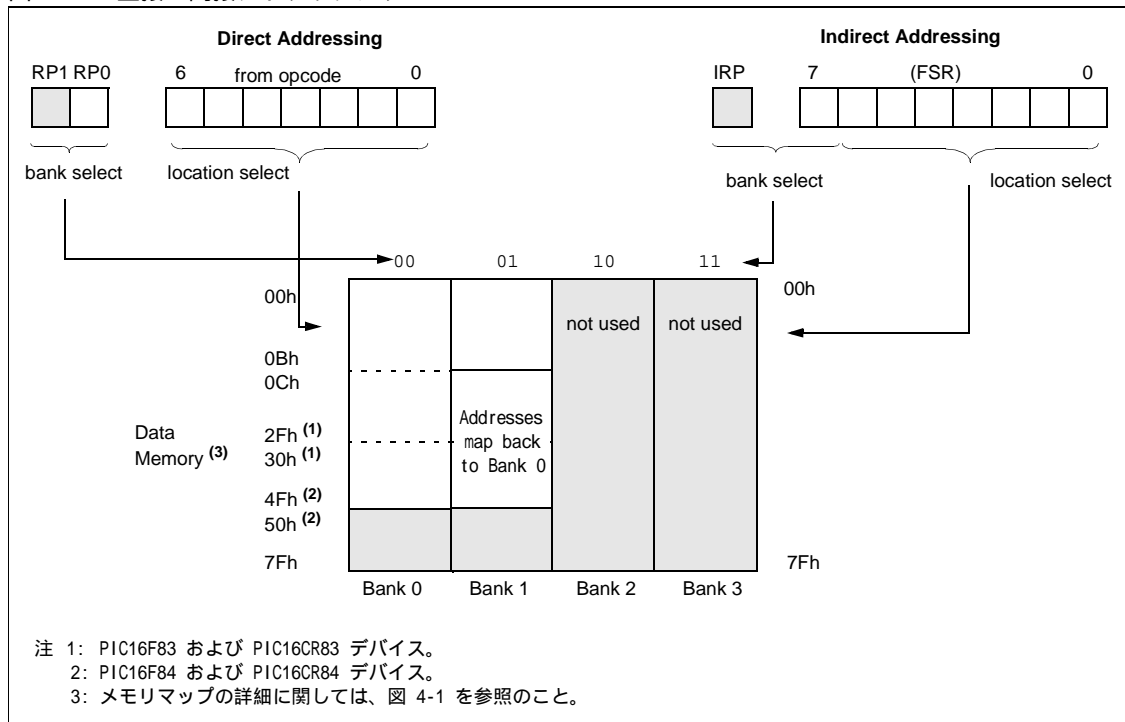
例 4-2: 間接アドレッシングを使用した RAM の初期化

```

movlw 0x20 ;initialize pointer
movwf FSR ; to RAM
NEXT  clrf INDF ;clear INDF register
      incf FSR ;inc pointer
      btfss FSR,4 ;all done?
      goto NEXT ;NO, clear next
CONTINUE
      : ;YES, continue
    
```

有効な 9 ビットアドレスは、図 4-1 に示すように、8 ビットの FSR レジスタと IRP ビット (STATUS<7>) を連結することによって取得できます。しかし、IRP は PIC16F8X では使用されません。

図 4-1: 直接/間接アドレッシング



PIC16F8X

NOTES:

5.0 I/O ポート

PIC16F8X には PORTA と PORTB の 2 つのポートがあります。ポートピンの一部は複数の機能とマルチプレクスされています。

5.1 PORTA と TRISA レジスタ

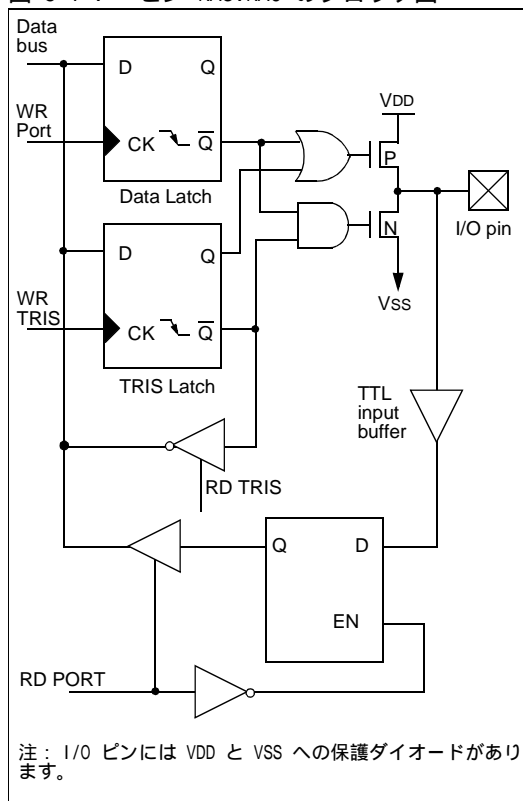
PORTA は 5 ビット幅のラッチです。RA4 はシュミットトリガ入力およびオープンドレイン出力です。他のすべての RA ポートピンは TTL レベル入力および CMOS 出力ドライバを備えています。すべてのピンは入出力設定ビット (TRIS レジスタ) を備えています。

TRISA ビットがセット (=1) されると、そのビットに対応する PORTA ピンが入力となり、出力ドライバがハイインピーダンスモードになります。TRISA ビットをクリア (=0) されると、そのビットに対応する PORTA ピンが出力となり、出力ラッチの内容がピンに出力されます。

PORTA レジスタをリードするとピンの状態がリードされ、PORTA レジスタへのライトはポートラッチへのライトになります。すべてのライトはリード・モディファイ・ライトです。つまり、ポートへのライトは、最初にポートピンがリードされ、次にこの値が変更され、ポートデータラッチにライトされることを意味します。

RA4 ピンは TMR0 クロック入力にも使用できます。

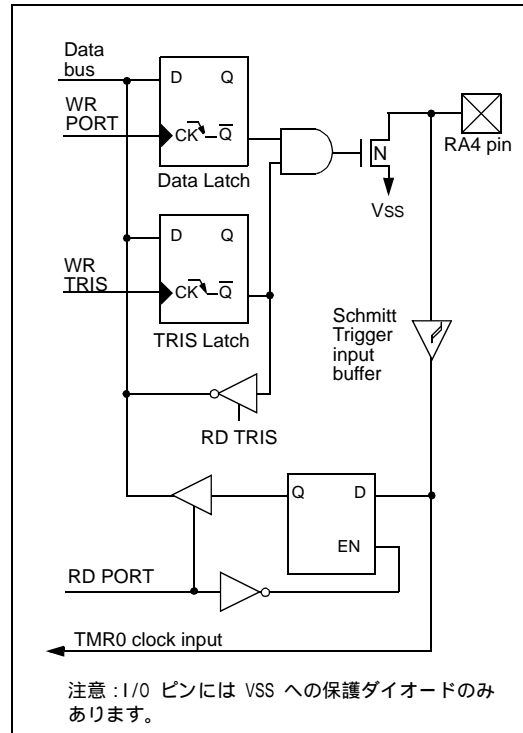
図 5-1 : ピン RA3:RA0 のブロック図



例 5-1: PORTA の初期化

```
CLRWF PORTA ; 出力データラッチへライトして PORTA を初期化する
BSF STATUS, RPO ; Bank 1 を選択する
MOVLW 0X0F ; データ方向の初期化に使用する値
MOVWF TRISA ; RA<3:0> を入力、RA4 を出力として設定。TRISA<7:5> は常に '0' とリードされる
```

図 5-2 ピン RA4 のブロック図



PIC16F8X

表 5-1 PORTA の機能

Name	Bit0	バッファタイプ	機能
RA0	bit0	TTL	入出力
RA1	bit1	TTL	入出力
RA2	bit2	TTL	入出力
RA3	bit3	TTL	入出力
RA4/T0CKI	bit4	ST	入出力または TMRO 用外部クロック入力。 出力はオープンドレインタイプ

汎用 : TTL = TTL 入力、ST = シュミットトリガ入力

表 5-2 PORTA に関連するレジスタの概要

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	パワー・オン・リセットでの値	他のすべてのリセットでの値
05h	PORTA	-	-	-	RA4/T0CKI	RA3	RA2	RA1	RA0	---x xxxx	---u uuuu
85h	TRISA	-	-	-	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	---1 1111	---1 1111

汎用 : x = 不定、u = 不変、- = なし、'0' とリードされる。網掛部分はビットがありません。'0' とリードされます。

5.2 PORTB と TRISB レジスタ

PORTB は 8 ビット幅の双方向ポートです。対応するデータ方向レジスタは TRISB です。TRISB レジスタのビットが '1' の場合、そのビットに対応する出力ドライバがハイインピーダンスモードになります。TRISB レジスタのビットが '0' の場合、選択されたピンに対し、出力ラッチの内容を出力します。

PORTB の各ピンには内部で弱プルアップの設定を行うことができます。1ビットのコントロールビットによって、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、RBPU (OPTION_REG<7>) ビットをクリアすると実行できます。ポートピンが出力に設定されている時は、弱プルアップが自動的にオフになります。プルアップはパワーオン・リセットによりオフになります。

PORTB の 4 ポート、RB7 から RB4 は変化による割り込み機能を持っています。入力として設定されているピンだけが割り込みの対象となります。(すなわち、出力設定の RB7:RB4 はピン変化割り込みの比較対象ではありません)。入力ピンの値は、最後にリードされたときの値と比較されます。各ピンの "ミスマッチ" 出力のORによりRB ポート変化割り込みが生成されます。

この割り込みによって、デバイスを SLEEP から起動することができます。割り込みルーチンでは、以下の方法で割り込みをクリアします。

- PORTB をリード (またはライト) する。これによってミスマッチでない状態になります。
- フラグビット RBIF をクリアします。

ミスマッチ状態は RBIF ビットをセットし続けます。PORTB のリードによってミスマッチでない状態となり、RBIF ビットをクリアできるようになります。

このミスマッチ割り込みとプルアップを設定可能な 4 ピンを使用してキーパットのインタフェースを簡単につくれます。また、キーを押したときに SLEEP から起動させることができます (Embedded Control Handbook の AN552 参照)。

注意 1: I/O ピン変化の検出には、TCY (4/fOSC) 以上のパルス幅が必要です。

このピン変化による割り込みはキーを押したときの SLEEP からの起動に使用することを推奨します。その時、PORTB はピン変化による割り込みのみに使用することを推奨します。ピン変化による割り込みを使用している間は PORTB のポーリングは推奨できません。

図 5-3: ピン RB7 : RB4 のブロック図

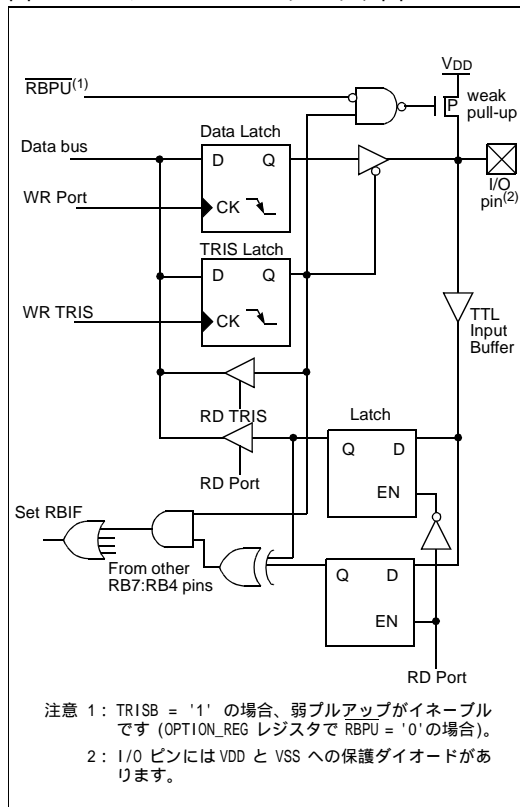
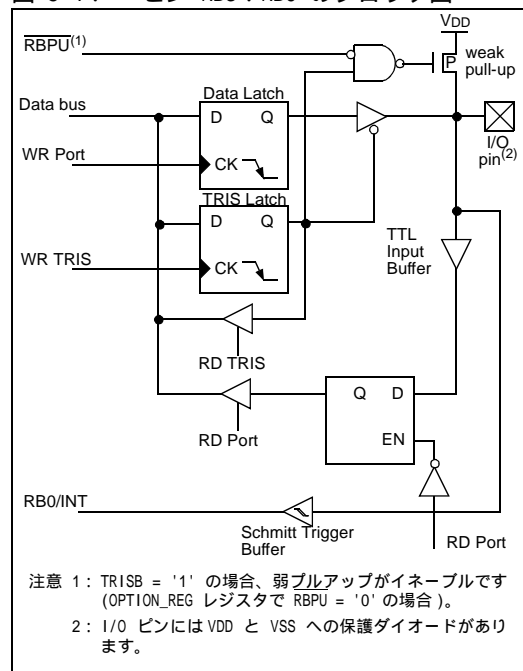


図 5-4: ピン RB3 : RB0 のブロック図



PIC16F8X

例 5-1: PORTB の初期化

```

CLRFB   PORTB       ; 出力データラッチへライトして
                    ; PORTB を初期化する
BSF     STATUS, RPO ; Bank 1 を選択する
MOVLW  0xCF         ; データ方向の初期化に使用する
MOVWF  TRISB       ; RB<3:0> を入力、RB<5:4> を出
                    ; 力、RB<7:6> を入力に設定する
    
```

表 5-3: PORTB の機能

Name	Bit	Buffer Type	バッファのタイプ I/O の整合性機能
RB0/INT	bit0	TTL/ST ⁽¹⁾	入出力ピンまたは外部割り込み入力。内部ソフトウェア・プログラマブル弱プルアップ。
RB1	bit1	TTL	入出力ピン。内部ソフトウェア・プログラマブル弱プルアップ。
RB2	bit2	TTL	入出力ピン。内部ソフトウェア・プログラマブル弱プルアップ。
RB3	bit3	TTL	入出力ピン。内部ソフトウェア・プログラマブル弱プルアップ。
RB4	bit4	TTL	入出力ピン (変化による割り込みあり)。内部ソフトウェア・プログラマブル弱プルアップ。
RB5	bit5	TTL	入出力ピン (変化による割り込みあり)。内部ソフトウェア・プログラマブル弱プルアップ。
RB6	bit6	TTL/ST ⁽²⁾	入出力ピン (変化による割り込みあり)。内部ソフトウェア・プログラマブル弱プルアップ。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 (変化による割り込みあり)。内部ソフトウェア・プログラマブル弱プルアップ。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。

注意 1: このバッファは外部割り込みに使用される場合、シュミットトリガ入力になります。

2: このバッファはシリアルプログラミングで使用される場合、シュミットトリガ入力になります。

表 5-4: PORTB に関連するレジスタの概要

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	パワーオン・リセットでの値	他の全てのリセットでの値
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0/INT	xxxx xxxx	uuuu uuuu
86h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	1111 1111
81h	OPTION_REG	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111

凡例: x = 不定、u = 不変。網掛部分は PORTB では使用しません。

5.3 I/O プログラミングの注意点

5.3.1 双方向 I/O ポート

ライトを実行する命令はすべて、内部でリード後、ライトを実行します。例えば、BCF および BSF 命令は、レジスタを CPU にリードして、ビット操作を実行後、その結果を再びライトします。このような命令が、入力と出力の両方を設定されているポートに対して実行される時には注意が必要です。例えば、PORTB のビット 5 の BSF は PORTB の全 8 ビットが CPU にリードされます。そして、BSF がビット 5 に実行され、PORTB の出力ラッチにライトされます。PORTB の別のビットが双方向の I/O ピン（仮にビット 0）として使用されていて、その時入力として使用されている場合、そのピンの入力信号が CPU にリードされ、このピンのデータラッチに再度ライトして、元の内容を上書きすることになります。そのピンが入力のままの場合は問題は起こりません。しかし、ビット 0 がその後出力に切り替えられた場合、データラッチの内容（出力）は不定です。

ポートレジスタのリードによって、ポートピンの値がリードされます。ポートレジスタのライトによってポートラッチの値がライトされます。ポートでリード・モディファイ・ライト命令（BCF、BSF など）を使用すると、ポートピンの値がリードされ、この値に対して操作が実行された後、この値がポートラッチにライトされます。

ハイまたはローを出力しているピンにはピンのレベル変更のためのドライブを外部デバイスから行わないでください。（"wired-or"、"wired-and"）。高出力電流によってチップが損傷を受ける場合があります。

5.3.2 I/O ポートの連続動作

I/O ポートへライトは命令サイクルの最後に起こりませんが、リードはデータは命令サイクルの最初で有効でなければいけません（図 5-5）。そのためにライトに続くリードが同じポートに対して実行された場合には注意が必要です。その次の命令で CPU へのリードが実行される前に、ピン電圧が安定する（外部負荷による）シーケンスで命令を実行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なく前の状態が CPU にリードされてしまうことがあります。不確かな場合は、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て、その命令を離してください。

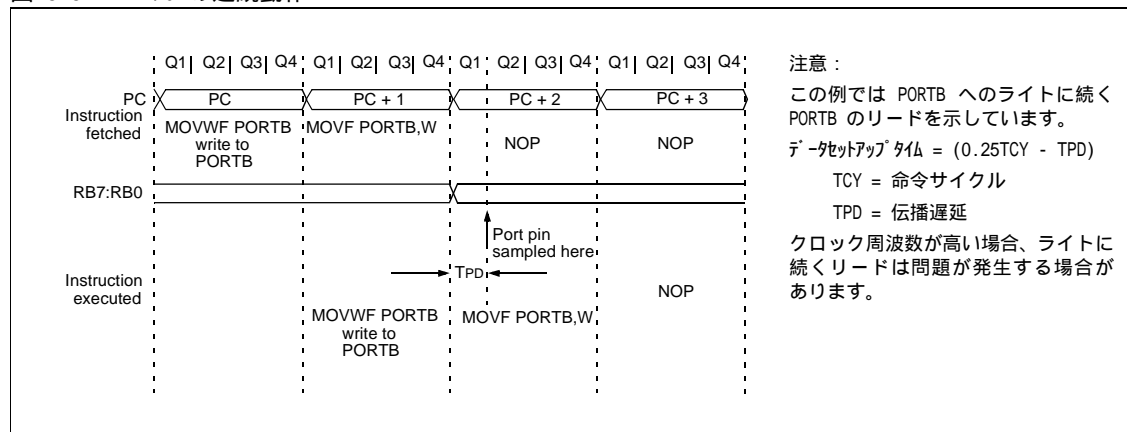
例 5-1 は I/O ポートで連続したリード・モディファイ・ライト命令（BCF、BSF など）による影響を示します。

例 5-1 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, RPO  ;
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).
    
```

図 5-5 I/O の連続動作



PIC16F8X

NOTES:

6.0 TIMER0 モジュールおよび TMR0 レジスタ

TIMER0 モジュールのタイマ/カウンタには次の機能があります。

- ・ 8 ビットタイマ/カウンタ
- ・ リードおよびライト可能
- ・ ソフトウェア・プログラマブル 8 ビットプリスケアラ
- ・ 内部または外部クロック選択
- ・ FFh から 00h オーバフロー割込み
- ・ 外部クロック用エッジ選択

TOCS ビット (OPTION_REG<5>) が 0 のとき、タイマモードになります。タイマモードではタイマ 0 モジュール (図 6-1) は (プリスケアラなしの場合) 命令サイクルごとにインクリメントされます。TMR0 レジスタがライトされた場合、インクリメントはそれに続く 2 サイクルはインクリメントしません (図 6-2、6-3)。その分の値を TMR0 レジスタにライトすることにより、調整することができます。

TOCS ビット (OPTION_REG<5>) が 1 のとき、カウンタモードになります。このモードにおいてタイマ 0 は RA4/TOCKI ピンの立ち上がり、または、立ち下がりエッジごとにインクリメントします。T0 ソースエッジ選択ビット、TOSE (OPTION_REG<4>) によりインクリメント

するエッジを選択します。TOSE ビットが 0 のとき、立ち上がりエッジが選択されます。外部クロック入力の規定については、第 6.2 項で詳しく説明します。

プリスケアラはタイマ 0 モジュールまたはウォッチドッグタイマのどちらかでのみ使用できます。プリスケアラの割り当てはソフトウェアでコントロールビット PSA (OPTION_REG<3>) により選択します。PSA ビットが 0 のとき、プリスケアラはタイマ 0 モジュールに割り当てられます。プリスケアラはリードもライトもできません。プリスケアラ (第 6.3 項) がタイマ 0 モジュールに割り当てられたとき、プリスケアラ値 (1:2、1:4、から 1:256 まで) はソフトウェアにより選択できます。

6.1 タイマ 0 割り込み

TMR0 レジスタが FFh から 00h にオーバーフローしたとき、TMR0 割り込みが発生します。このオーバーフローにより T0IF ビット (INTCON<2>) がセットされます。T0IE (INTCON<5>) イネーブルビットを 0 にすることにより割り込みをマスクできます。この割り込みを再度イネーブルにする前に、タイマ 0 モジュール割り込みサービスルーチンでソフトウェアにより T0IF ビットを 0 にしなければいけません。SLEEP の間はタイマが停止しているため、TMR0 割り込み (図 6-4) ではプロセッサを SLEEP から起動することはできません。

図 6-1: タイマ 0 ブロック図

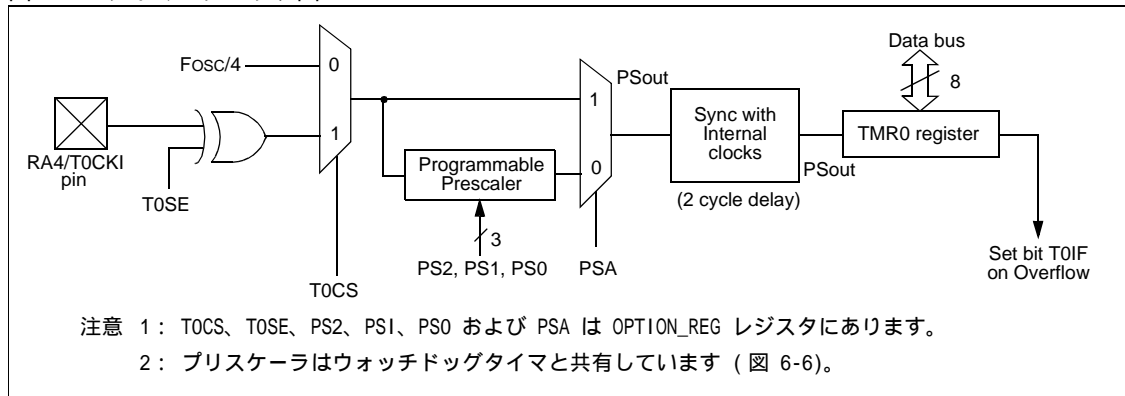
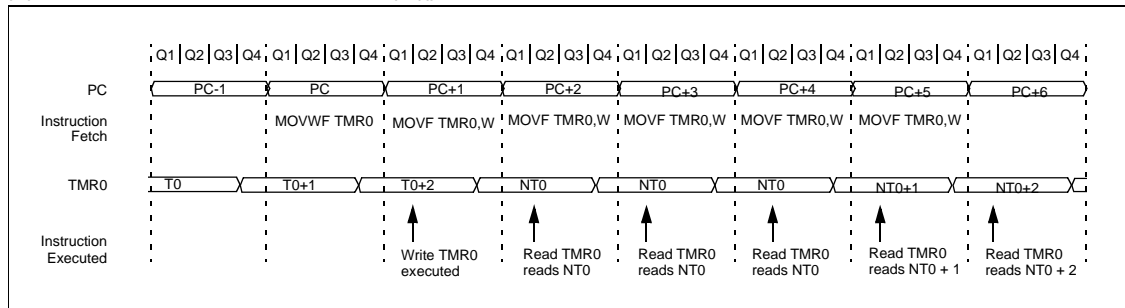


図 6-2: タイマ 0 のタイミング: 内部クロック/プリスケールなし



PIC16F8X

図 6-3: タイマ 0 のタイミング: 内部クロック / プリスケール 1 : 2

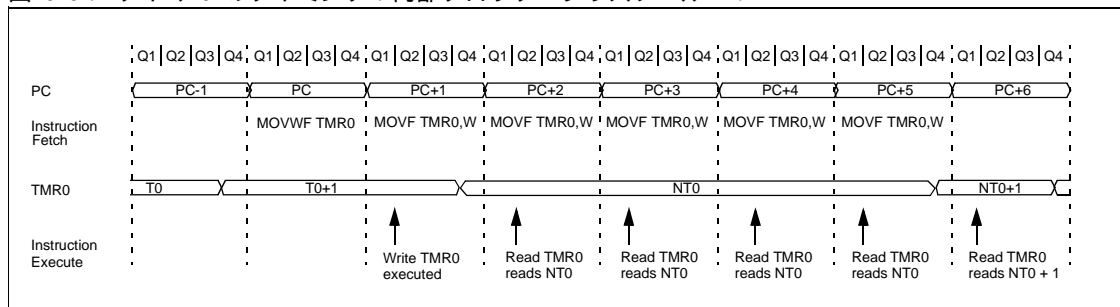
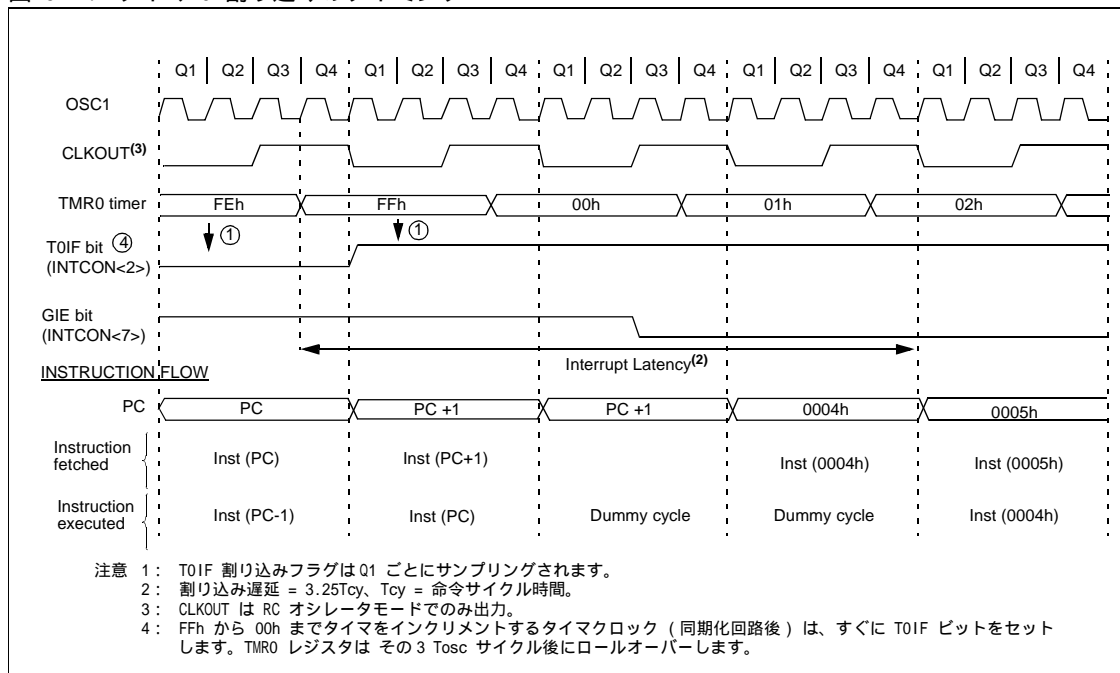


図 6-4: タイマ 0 割り込みのタイミング



6.2 外部クロックによるタイマ 0 使用方法

外部クロックがタイマ 0 に入力されるときは、必要条件を満たさなければなりません。その条件とは内部位相クロック (Tosc) との同期によるものです。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。

6.2.1 外部クロック同期

プリスケアラを使用しない時、外部クロック入力はプリスケアラ出力と同じです。RA4/T0CKI と内部位相クロックとの同期はプリスケアラ出力を内部位相クロックの Q2 と Q4 サイクルでサンプリングすることにより行います (図 6-5)。したがって、T0CKI は最低 2TOSC の間ハイ (と少しのRC 遅延時間) と最低 2TOSC の間ロー (と少しのRC 遅延時間) になっている必要があります。電氣的仕様の章を参照してください。

プリスケアラを使用する時、そのプリスケアラ出力が対称となるように、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割されます。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。T0CKI は最低 4TOSC (と少しのRC 遅延時間) をプリスケアラ値で割った周期でなければいけません。かつ、T0CKI のハイタイムとロータイムの最低パルス幅は 10ns 以上です。電氣的仕様についてはパラメータ 40、41 および 42 を参照してください。

6.2.2 タイマ 0 のインクリメントの遅延時間

プリスケアラ出力は内部クロックと同期がとられているので、外部クロックのエッジが発生した時からタイマ 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 6-5 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。

6.3 プリスケアラ

8 ビットカウンタは、タイマ 0 モジュール用のプリスケアラとして、または、ウォッチドッグタイマ用のポストアスケアラとして使用できます (図 6-6)。まぎらわしいのでこのデータシートでは、このカウンタを「プリスケアラ」と言います。プリスケアラはタイマ 0 モジュールと WDT のどちらかに使用できますが、両方に使用することはできないので注意してください。したがって、プリスケアラがタイマ 0 モジュールに使用されているときは、WDT はプリスケアラは使用できません。また、この逆も言えます。

PSA および PS2:PS0 ビット (OPTION_REG<3:0>) がプリスケアラ割り当てとプリスケール比を決定します。

タイマ 0 モジュールに割り当てられているときは、タイマ 0 モジュールへのライトはすべて (例えば、CLRWF 1、MOVWF 1、BSF 1、x など) によりプリスケアラがクリアされます。WDT に割り当てられているときは、CLRWDW 命令によりプリスケアラと WDT がクリアされます。プリスケアラはリードもライトもできません。

PIC16F8X

図 6-5: 外部クロックによるタイマ 0 のタイミング

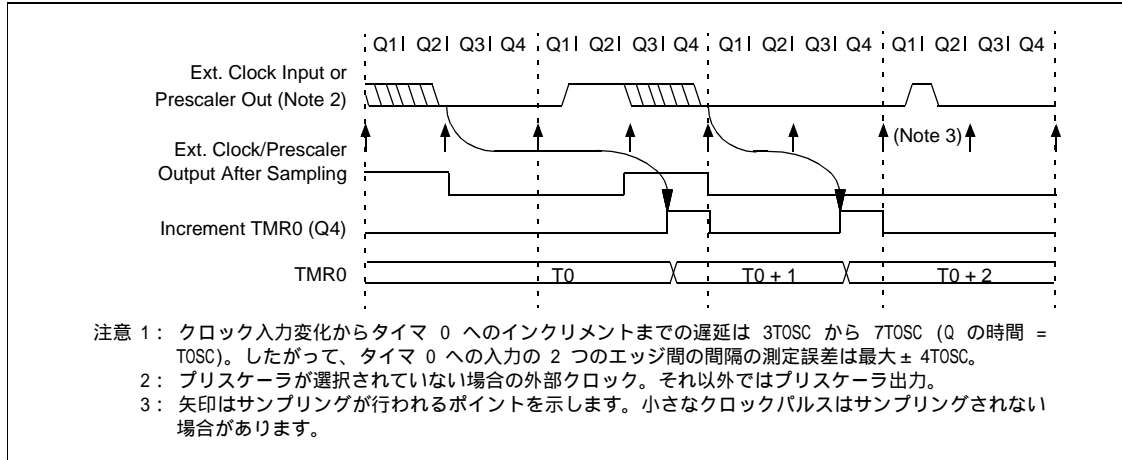
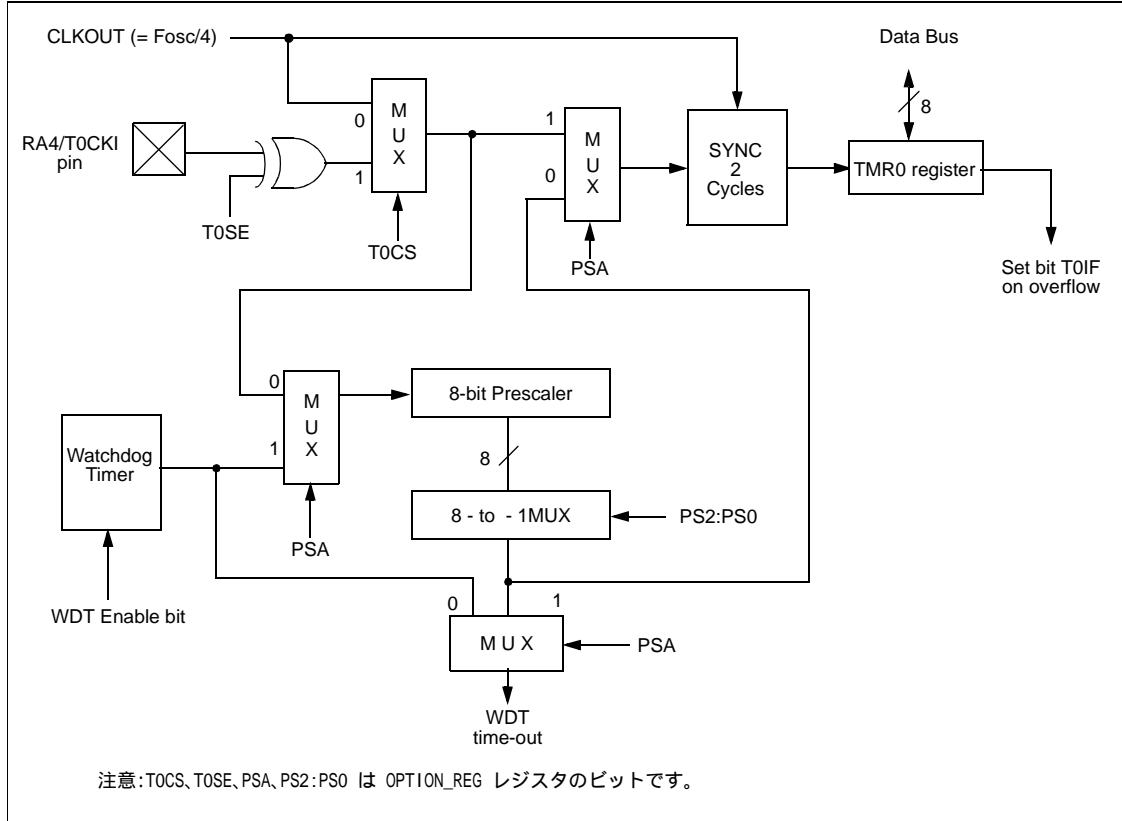


図 6-6: タイマ / WDT プリスケーラのブロック図



6.3.1 プリスケラ割り当ての切り替え

プリスケラの割り当てはソフトウェアにより制御します (プログラム実行中に変更できます)。

注意: 不必要なデバイスリセットを避けるために、プリスケラの割り当てをタイマ 0 から WDT に変更するときは、次のような順序で命令 (例 6-1) を実行する必要があります。このシーケンスは WDT がディスエーブルになっている場合でも実行しなくてはなりません。プリスケラを WDT からタイマ 0 モジュールに変更するには、例 6-2 に示すシーケンスを使用してください。

例 6-1: プリスケラの変更 (タイマ 0 WDT)

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

例 6-2: 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 clock source
MOVWF OPTION_REG ;
BCF STATUS, RP0 ;Bank 0
```

表 6-1 タイマ 0 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	パワーオンリセットでの値	他のすべてのリセットでの値
01h	TMR0	タイマ 0 モジュールのレジスタ								xxxx xxxx	uuuu uuuu
0Bh	INTCON	GIE	EEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 0000
81h	OPTION_REG	$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
85h	TRISA	—	—	—	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	---1 1111	---1 1111

凡例: x = 不定, u = 不変, - = 網掛部分はタイマ 0 では使用しません。

PIC16F8X

NOTES:

7.0 データ EEPROM メモリ

EEPROM データメモリは、通常動作 (VDD 動作電圧範囲) でリードおよびライトが可能です。このメモリはレジスタファイル空間に直接マッピングされていません。特殊機能レジスタを通して間接的にアドレス指定されます。このメモリのリードおよびライトには 4 つの SFR が使用されます。そのレジスタは以下の通りです。

- EECON1
- EECON2
- EEDATA
- EEADR

EEDATA はリード/ライトするための 8 ビットデータを保持して、EEADR はアクセスされる EEPROM 番地のアドレスを保持します。PIC16F8X デバイスには 0h から 3Fh までのアドレス範囲の 64 バイトのデータ EEPROM があります。

EEPROM データメモリはバイトリードおよびライトができます。バイトライトは自動的にその番地を消去して、新しいデータをライトします (ライトする前に消去)。EEPROM データメモリは消去 / ライトサイクルが高速です。ライト時間はオンチップタイマによって制御されます。ライト時間は チップのばらつき、および、電圧や温度によって変化します。詳細は、AC 仕様を参照してください。

デバイスがコードプロテクトされているときでも、CPU はデータ EEPROM メモリをリードおよびライトできます。プログラムライターは、このメモリにアクセスできなくなります。

7.1 EEADR

EEADR レジスタは最大 256 バイトのデータ EEPROM をアドレス指定できます。最初の 64 バイトのデータ EEPROM だけが物理的に存在します。

上位 2 ビットもアドレスとしてデコードされます。アドレスが必ず 64 バイトのメモリ空間にあるように、上位 2 ビットを常に '0' にしておく必要があります。

図 7-1: EECON1 レジスタ (アドレス 88h)

U	U	U	R/W-0	R/W-x	R/W-0	R/S-0	R/S-x	
—	—	—	EEIF	WRERR	WREN	WR	RD	
bit7								bit0
<p>bit7:5 使用しない: '0' としてリードされます</p> <p>bit4 EEIF: EEPROM ライト割込みフラグビット 1 = ライト完了 (ソフトウェアでクリアする必要があります) 0 = ライトが完了していないか、開始していない</p> <p>bit3 WRERR: EEPROM エラーフラグビット 1 = ライトが異常終了した (通常動作中の MCLR リセットまたは WDT リセット) 0 = ライトが完了した</p> <p>bit2 WREN: EEPROM ライトイネーブルビット 1 = ライトサイクルを許可します 0 = データ EEPROM のライトを許可しません。</p> <p>bit1 WR: ライトコントロールビット 1 = ライトサイクルを開始します (ライトが完了したら、ハードウェアによってビットがクリアされます)。WR ビットはソフトウェアではセットだけができます (クリアはできません)。 0 = データ EEPROM へのライトサイクルが完了しました</p> <p>bit0 RD: リードコントロールビット 1 = EEPROM リードを開始します (リードには 1 サイクルかかります。RD はハードウェアでクリアされます。RD ビットはソフトウェアではセットだけが可能です (クリアはできません))。 0 = EEPROM リードを開始しません</p>								

R = リード可能なビット
W = ライト可能なビット
S = セット可能なビット
U = 使用しないビット、'0' としてリード
- n = POR リセットでの値

PIC16F8X

7.2 EECON1 および EECON2 のレジスタ

EECON1 は、下位 5 ビットのみ物理的に存在するコントロールレジスタです。上位 3 ビットは存在しません。'0' としてリードされます。

コントロールビット RD および WR はリードとライトをそれぞれ開始します。これらのビットはソフトウェアではクリアできず、セットできるだけです。これらは、リードまたはライトが完了したときにハードウェアでクリアされます。ソフトウェアで WR ビットをクリアできないことにより、ライトの予期しない、異常終了を防ぐことができます。

WREN ビットがセットされると、ライトが可能になります。電源 ON のとき、WREN ビットはクリアされます。ライトが正常に動作しているときに MCLR リセットまたは WDT のタイムアウトリセットによって中断された場合、WRERR ビットがセットされます。そのようなとき、リセットの後に WRERR ビットをチェックし、その番地を再度ライトできます。EEDATA および EEADR レジスタの中のデータおよびアドレスはリセットされません。

ライトが完了すると、割込みフラグビット EEIF がセットされます。このビットはソフトウェアでクリアする必要があります。

EECON2 は物理的に存在するレジスタではありません。EECON2 のリードは、すべて '0' としてリードされません。EECON2 レジスタはデータ EEPROM ライトシーケンスで使用されます。

7.3 EEPROM データメモリのリード

EEPROM データメモリのある番地をリードするには、EEADR レジスタにアドレスをライトして、コントロールビット RD (EECON1 <0>) をセットします。次のサイクルには EEDATA レジスタにデータが入力されるので、次の命令でリードできます。この値は、その後リードまたはライトが行われるまで、EEDATA で保持されます。

例 7-1: データ EEPROM のリード

```
BCF STATUS, RP0 ; Bank 0
MOVLW CONFIG_ADDR ;
MOVWF EEADR ; Address to read
BSF STATUS, RP0 ; Bank 1
BSF EECON1, RD ; EE Read
BCF STATUS, RP0 ; Bank 0
MOVF EEDATA, W ; W = EEDATA
```

7.4 EEPROM データメモリへのライト

EEPROM データメモリのある番地をライトするには、まず、EEADR レジスタにアドレスをライトし、EEDATA レジスタにデータをライトします。次に、下記の特別なシーケンスを行います。このシーケンスはライトを 1 バイト実行することに必要です。

例 7-1: データ EEPROM のライト

```
BSF STATUS, RP0 ; Bank 1
BCF INTCON, GIE ; Disable INTs.
BSF EECON1, WREN ; Enable Write
```

Required Sequence	MOVLW	55h	;
	MOVWF	EECON2	;
	MOVLW	AAh	;
	MOVWF	EECON2	;
	BSF	EECON1, WR	;
			;
	BSF	INTCON, GIE	;

1 バイトごとに、上記のシーケンスが正確に実行されない場合は (EECON2 へ 55h をライトして EECON2 へ AAh をライトし、WR ビットをセット)、ライトは開始されません。このシーケンスを実行するルーチンは割込みをディセーブルにすることを推奨します。

さらに、EECON1 の WREN ビットをセットして、ライトをイネーブルにしておきます。このメカニズムによって、予期しないコード実行によるデータ EEPROM へのライトを防ぐのに役立ちます。EEPROM を更新するとき以外は、WREN ビットを常にクリアしておく必要があります。WREN ビットはハードウェアではクリアされません。

ライトシーケンスが開始された後は、WREN ビットをクリアしてもこのライトサイクルには影響ありません。WREN ビットが 0 のとき、WR ビットはセットできません。

ライトサイクルが完了したとき、ハードウェアは WR ビットをクリアして、EE ライト完了割込みフラグビット (EEIF) をセットします。この割り込みをイネーブルにすることも、または、このビットのポーリングすることもできます。EEIF はソフトウェアでクリアします。

7.5 ライトのベリファイ

ライトしたデータ EEPROM がライトした値になっていることをベリファイすることを推奨します。(例 7-1) EEPROM が規格値付近まで使用される場合にこのベリファイを実行することを推奨します。そのレベルを決定するには Total Endurance ディスクが役立ちます。通常、EEPROM のライト失敗は、'1' とライトされたビットが (ビットのリークにより) '0' とリードされます。

例 7-1: ライトのベリファイ

```
BCF STATUS, RP0 ; Bank 0
:                ; Any code can go here
:                ;
MOVWF EEDATA, W ; Must be in Bank 0
BSF STATUS, RP0 ; Bank 1
READ
BSF EECON1, RD ; YES, Read the
                ; value written
BCF STATUS, RP0 ; Bank 0
;
; Is the value written (in W reg) and
; read (in EEDATA) the same?
;
SUBWF EEDATA, W ;
BTFSZ STATUS, Z ; Is difference 0?
```

```
GOTO WRITE_ERR ; NO, Write error
:              ; YES, Good write
:              ; Continue program
```

7.6 不必要なライトからの保護

データ EEPROM メモリをライトさせたくない状態があります。不必要な EEPROM ライトから保護するために、いくつかのメカニズムが組み込まれています。電源 ON のとき、WREN はクリアされます。また、パワーアップ タイマ (72 ms 期間) が EEPROM ライトを防ぎます。ライト開始シーケンスと WREN ビットは、電源電圧低下、電源瞬断、ソフトウェアの誤動作などが発生したときの予期しないライトを防ぐのに役立ちます。

7.7 コードプロテクトしているときのデータ EEPROM

デバイスがコードプロテクトされていても、CPU はスクランブルされていないデータをデータ EEPROM へリードおよびライトできます。

ROM デバイスの場合は、2 つのコードプロテクトビットがあります (第 8.1 項)。1 つは ROM プログラムメモリ用で、もう 1 つはデータ EEPROM メモリ用です。

表 7-1 EEPROM に関連するレジスタ/ビット

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	パワーオンリセットによる値	他のすべてのリセットによる値
08h	EEDATA	EEPROM データレジスタ								xxxx xxxx	uuuu uuuu
09h	EEADR	EEPROM アドレス レジスタ								xxxx xxxx	uuuu uuuu
88h	EECON1	—	—	—	EEIF	WRERR	WREN	WR	RD	---0 x000	---0 q000
89h	EECON2	EEPROM コントロール レジスタ 2								---- ----	---- ----

凡例: x = 不定、u = 不変、- = 使用しない、'0' としてリードされる、q = 条件によって異なる値。網掛部分はデータ EEPROM では使用しません。

PIC16F8X

NOTES:

8.0 CPU の特殊機能

マイクロコントローラはリアルタイムアプリケーションに適した特殊回路を搭載しています。PIC16F8X は下記のようなシステムの信頼性を最大限に高めたり、外付け部品の削減によりコストを最小限にしたり、低消費電力モードを備えたり、コードを保護するための機能を搭載しています。

- ・ オシレータ選択
- ・ リセット
 - 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)
- ・ 割り込み
- ・ ウォッチドッグタイム (WDT)
- ・ SLEEP
- ・ コードプロテクト
- ・ ID ロケーション
- ・ インサーキットシリアルプログラミング

PIC16F8X にはコンフィグレーションビットでのみ停止できるウォッチドッグタイムがあります。ウォッチドッグタイムは信頼性を上げるために、専用の RC オシレータで動作します。電源 ON 時に必要な遅延時間を作るために 2 つのタイムがあります。1 つはオシレータスタートアップタイム (OST) で、水晶発振子が安定するまでデバイスをリセット状態にしておくために役立ちます。もう 1 つはパワーアップタイム (PWRT) で、電源 ON 時にのみ 72 ms (通常) の固定遅延時間を発生します。電源が安定するまでデバイスをリセット状態にしておくために役立ちます。チップに内蔵された 2 つのタイムにより、多くアプリケーションで外部リセット回路が不要となります。

SLEEP モードは非常に低消費電力が小さいモードです。外部リセット、ウォッチドッグタイムのタイムアウト、割り込みにより SLEEP から起動できます。複数のオシレータオプションが選択できます。RC オシレータオプションは低コストです。LP 水晶オプションは低消費電力です。コンフィグレーションビットの組み合わせでいろいろなオプションを選択できます。

8.1 コンフィグレーションビット

複数のデバイスの設定を選択するために、これらのコンフィグレーションビットをプログラム書き込みした状態 ('0' としてリード) またはプログラム書き込みしないままの状態 ('1' としてリード) にできます。これらのビットはプログラムメモリ番地 2007h に配置されています。

2007h 番地はユーザープログラムメモリの範囲外です。実際には、特殊テスト/コンフィグレーションメモリ範囲 (2000h - 3FFFh) に入っており、プログラム書き込み中のみアクセスできます。

PIC16F84 のプログラム書き込み方法については、『PIC16F8X EEPROM Memory Programming Specification (DS30262)』を参照してください。

PIC16F8X

図 8-1: コンフィグレーションワード - PIC16CR83 および PIC16CR84

R-u	R-u	R-u	R-u	R-u	R-u	R/P-u	R-u	R-u	R-u	R-u	R-u	R-u	R-u
CP	CP	CP	CP	CP	CP	DP	CP	CP	CP	PWRTE	WDTE	FOSC1	FOSC0
bit13										bit0			
<div style="border: 1px solid black; padding: 5px; width: fit-content; margin-left: auto; margin-right: auto;"> R = リード可能ビット P = プログラム可能ビット - n = POR リセットによる値 u = 不変 </div>													
bit13:8 CP: プログラム メモリ コードプロテクションビット 1 = コードプロテクションされていない 0 = プログラム メモリがコードプロテクションされている													
bit7 DP: データ メモリ コードプロテクションビット 1 = コードプロテクションされていない 0 = データ メモリがコードプロテクションされている													
bit6:4 CP: プログラム メモリ コードプロテクションビット 1 = コードプロテクションされていない 0 = プログラム メモリがコードプロテクションされている													
bit 3 PWRTE: パワーアップタイムアウトビット 1 = パワーアップタイムアウトを動作させない 0 = パワーアップタイムアウトを動作させる													
bit 2 WDTE: ウォッチドッグタイムアウトビット 1 = WDT を動作させる 0 = WDT を動作させない													
bit1:0 FOSC1:FOSC0: 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 8-2: コンフィグレーションワード - PIC16F83 および PIC16F84

R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u	R/P-u
CP	CP	CP	CP	CP	CP	CP	CP	CP	CP	PWRTE	WDTE	FOSC1	FOSC0
bit13										bit0			
<div style="border: 1px solid black; padding: 5px; width: fit-content; margin-left: auto; margin-right: auto;"> R = リード可能ビット P = プログラム可能ビット - n = POR リセットによる値 u = 不変 </div>													
bit13:4 CP: コードプロテクションビット 1 = コードプロテクションしない 0 = 全メモリをコードプロテクションする													
bit3 PWRTE: パワーアップタイムアウトビット 1 = パワーアップタイムアウトを動作させない 0 = パワーアップタイムアウトを動作させる													
bit2 WDTE: ウォッチドッグタイムアウトビット 1 = WDT を動作させる 0 = WDT を動作させない													
bit 1:0 FOSC1:FOSC0: 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													

PIC16F8X

8.2 オシレータの構成

8.2.1 オシレータの種類

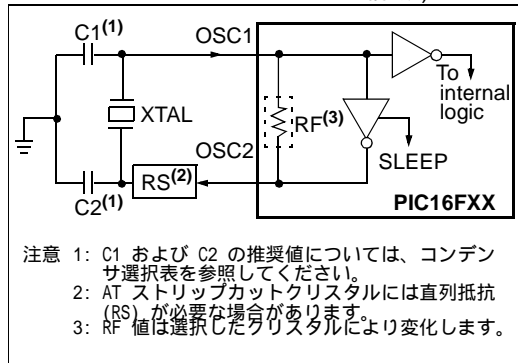
PIC16F8X は 4 つのオシレータモードで動作できます。2 ビットのコンフィグレーションビット (FOSC1 および FOSC0) をプログラムして、以下の 4 種類のモードから 1 つを選択することができます。

- ・ LP 低消費電力水晶
- ・ XT 水晶 / レゾネータ
- ・ HS 高速水晶 / レゾネータ
- ・ RC 抵抗 / コンデンサ

8.2.2 水晶オシレータ / セラミックレゾネータ

XT, LP または HS モードでは、クリスタルまたはセラミックレゾネータを OSC1/CLKIN および OSC2/CLKOUT ピンに接続して発振させます (図 8-3)。

図 8-3: 水晶 / セラミックレゾネータ (HS, XT または LP の OSC 構成)



PIC16F8X のオシレータの設計にはパラレルカット水晶を使用する必要があります。シリーズカット水晶を使用すると、周波数が水晶メーカーの規格外になる場合があります。XT, LP または HS のモードでは OSC1/CLKIN ピンへ外部クロックを入力することができます。(図 8-4)。

図 8-4: 外部クロック入力 (HS, XT または LP の OSC 構成)

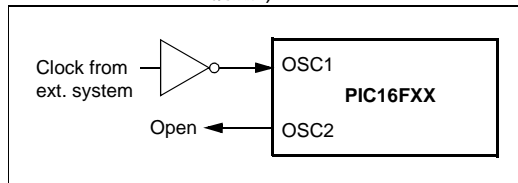


表 8-1 セラミックレゾネータ用コンデンサ選択表

テストした範囲:			
Mode	Freq	OSC1/C1	OSC2/C2
XT	455 kHz	47 - 100 pF	47 - 100 pF
	2.0 MHz	15 - 33 pF	15 - 33 pF
	4.0 MHz	15 - 33 pF	15 - 33 pF
HS	8.0 MHz	15 - 33 pF	15 - 33 pF
	10.0 MHz	15 - 33 pF	15 - 33 pF

注意: C1 と C2 の推奨値は表の範囲と同じです。容量が大きいと発振の安定性は高くなりますが、スタートアップ時間も長くなります。これらの値は参考値です。それぞれのレゾネータにはそれぞれの特性がありますので、外付け部品の適当な定数についてはレゾネータのメーカー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%
10.0 MHz	Murata Erie CSA10.00MTZ	± 0.5%

コンデンサ内蔵タイプはテストしておりません。

表 8-2 水晶用コンデンサ選択表

Mode	Freq	OSC1/C1	OSC2/C2
LP	32 kHz	68 - 100 pF	68 - 100 pF
	200 kHz	15 - 33 pF	15 - 33 pF
XT	100 kHz	100 - 150 pF	100 - 150 pF
	2 MHz	15 - 33 pF	15 - 33 pF
	4 MHz	15 - 33 pF	15 - 33 pF
HS	4 MHz	15 - 33 pF	15 - 33 pF
	10 MHz	15 - 33 pF	15 - 33 pF

注意: C1 と C2 の推奨値は表の範囲と同じです。容量が大きいと発振の安定性は高くなりますが、スタートアップ時間も長くなります。これらの値は参考値です。HS, XT モードでは、低駆動レベル規格の水晶のオーバードライブを防ぐため、Rs が必要な場合があります。それぞれの水晶にはそれぞれの特性がありますので、外付け部品の適当な定数については水晶のメーカーにお問い合わせください。VDD > 4.5V のとき、C1 = C2 = 30 pF が推奨です。

使用水晶:		
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
1.0 MHz	ECS ECS-10-13-2	± 50 PPM
2.0 MHz	ECS ECS-20-S-2	± 50 PPM
4.0 MHz	ECS ECS-40-S-4	± 50 PPM
10.0 MHz	ECS ECS-100-S-4	± 50 PPM

PIC16F8X

8.2.3 外部水晶オシレータ回路

オシレータモジュールを使用する、または、TTL ゲートを使用した簡単なオシレータ回路を組み立てて使用することもできます。オシレータモジュールは広い動作範囲と優れた安定性を持っています。良い設計の水晶は TTL ゲートと組み合わせるとよい性能を出します。2 種類的水晶オシレータ回路は、直列共振または並列共振として使用できます。

図 8-5 に並列共振オシレータ回路の構成を示します。この回路は水晶の基本周波数を使用するために設計されています。74AS04 インバータは並列オシレータに必要な 180 度の位相シフトを行います。4.7 k 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。10 k のポテンショメータは 74AS04 を線形領域で使用するためのバイアス用です。この回路は外部オシレータの設計に使用することができます。

図 8-5 外部並列共振水晶オシレータ回路

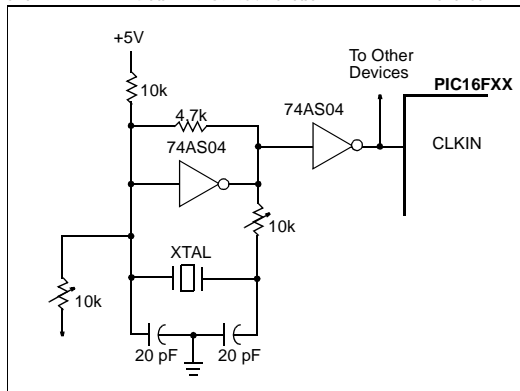
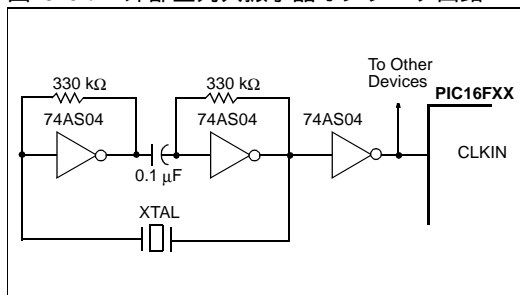


図 8-6 に直列共振オシレータ回路を示します。この回路も水晶の基本周波数を使用するために設計されています。インバータは直列共振オシレータ回路での 180 度の位相シフトを行います。330 k の抵抗はインバータを線形領域にバイアスするためにネガティブフィードバックを行います。

図 8-6 : 外部直列共振水晶オシレータ回路



8.2.4 RC オシレータ

それほどタイミングについて精度を必要としないアプリケーションでは、RC デバイスオプションを使用して低コスト化できます。RC 発振周波数は供給電圧、抵抗 (Rext) やコンデンサ (Cext) の値、および動作温度に

より変化します。これに加え、オシレータ周波数は製造上のばらつきによりデバイスごとに異なります。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 Cext の値が低いとき、発振周波数に影響します。外付けの R と C の誤差によるばらつきも考慮する必要があります。図 8-7 に R/C の組み合わせがどのように PIC16F8X に接続されるかを示します。Rext が 4k より小さいと、発振が不安定、停止する場合があります。Rext が非常に大きいと (例えば 1M)、発振はノイズ、湿度、リーク電流の影響を受けやすくなります。したがって、Rext 値は 5 k から 100 k を推奨します。

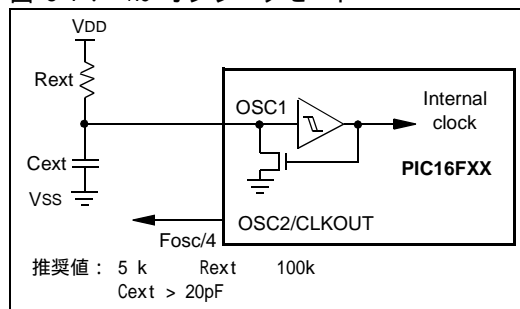
外付けコンデンサがなくても (Cext = 0pF) 発振しますが、ノイズ対策や安定のために 20pF 以上のコンデンサを推奨します。外部容量が小さいと、PCB のトレース容量やパッケージのリードフレーム容量などの外部容量の変化で、発振周波数が大きくばらつくことがあります。

製造上のばらつきによるデバイスごとの RC 周波数のばらつきに関しては、電気的特性の項をご覧ください。R が大きければ大きいほど (リーク電流のばらつきは R が大きいほど RC 周波数に影響するため)、C が小さければ小さいほど (入力容量が RC 周波数に影響するため)、ばらつきが大きくなります。

Rext/Cext の値と VDD による発振周波数の変化と、動作温度による周波数の変化に関しては、電気的特性の項を参照してください。

4 分周されたオシレータの周波数が OSC2/CLKOUT ピンに出力され、テストで使用したり、他のロジックと同期させるために使用することができます (波形に関しては、図 3-2 を参照してください)。

図 8-7 : RC オシレータモード



注意 : デバイスのオシレータが RC モードの場合は、OSC1 ピンに外部クロックを入力しないでください。デバイスが損傷する可能性があります。

8.3 リセット

PIC16F8X は以下のリセットを発生できます。

- ・ パワーオンリセット (POR)
- ・ 通常動作中の MCLR リセット
- ・ スリープ中の MCLR リセット
- ・ WDT タイムアウトリセット (通常動作中)
- ・ WDT スリープからの起動 (スリープ中)

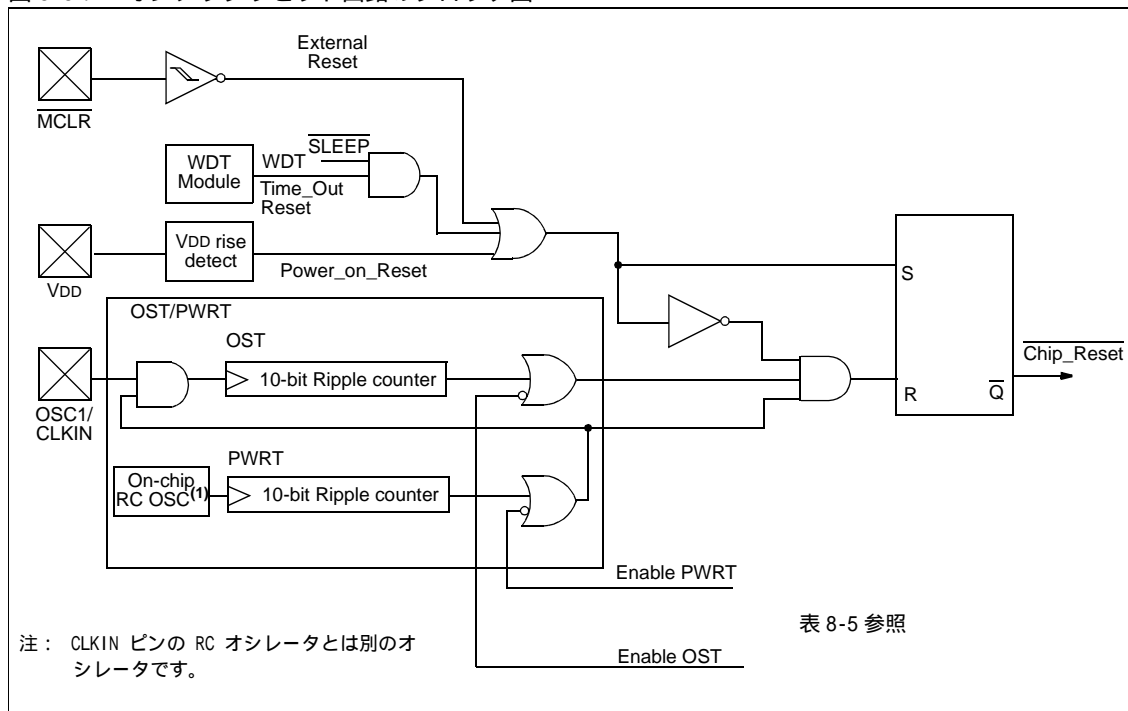
図 8-8 にオンチップのリセット回路の簡単なブロック図を示します。MCLR リセットパスには、小さなパルスを見逃すためのノイズフィルタがあります。電気的特性に MCLR ピンのパルス幅の必要条件を記載します。

リセットされないレジスタがあります。リセットされないレジスタの値は POR リセット後は不定、他のリセット後は不変です。他のレジスタは、通常動作中は POR、MCLR または WDT リセット、SLEEP 中は MCLR によりリセットされます。スリープ中の WDT リセットは通常動作の再開のように取り扱われるのでリセットされません。

表 8-3 にプログラムカウンタ (PC) および STATUS レジスタのリセット状態を示します。表 8-4 には、全レジスタのリセット状態を示します。

T0 ビットと PU ビットは、発生したリセットによりセットまたはクリアされます (第 8.7 項)。ソフトウェアはこのビットによって発生したリセットを識別で

図 8-8: オンチップリセット回路のブロック図



PIC16F8X

表 8-3 プログラムカウンタおよび STATUS レジスタのリセット状態

状態	プログラムカウンタ	STATUS レジスタ
パワーオンリセット	000h	0001 1xxx
通常動作中の MCLR リセット	000h	000u uuuu
スリープ中の MCLR リセット	000h	0001 0uuu
WDT リセット (通常動作中)	000h	0000 1uuu
WDT によるスリープからの起動	PC + 1	uuu0 0uuu
割り込みによるスリープからの起動	PC + 1 ⁽¹⁾	uuu1 0uuu

汎用： u = 不定、x = 不明。

注意 1：割り込みによるスリープからの起動、かつ、GIE ビットが1のとき、PC には割り込みベクタ (0004h) がロードされます。

表 8-4 レジスタのリセット状態

レジスタ	アドレス	パワーオンリセット	MCLR リセット： - 通常動作中 - SLEEP 中、通常動作中の WDT リセット	SLEEP からのウェークアップ： - - 割り込み - WDT タイムアウト
W	—	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF	00h	---- ----	---- ----	---- ----
TMR0	01h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	02h	0000h	0000h	PC + 1 ⁽²⁾
STATUS	03h	0001 1xxx	000q quuu ⁽³⁾	uuuq quuu ⁽³⁾
FSR	04h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA	05h	---x xxxx	---u uuuu	---u uuuu
PORTB	06h	xxxx xxxx	uuuu uuuu	uuuu uuuu
EEDATA	08h	xxxx xxxx	uuuu uuuu	uuuu uuuu
EEADR	09h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCLATH	0Ah	---0 0000	---0 0000	---u uuuu
INTCON	0Bh	0000 000x	0000 000u	uuuu uuuu ⁽¹⁾
INDF	80h	---- ----	---- ----	---- ----
OPTION_REG	81h	1111 1111	1111 1111	uuuu uuuu
PCL	82h	0000h	0000h	PC + 1
STATUS	83h	0001 1xxx	000q quuu ⁽³⁾	uuuq quuu ⁽³⁾
FSR	84h	xxxx xxxx	uuuu uuuu	uuuu uuuu
TRISA	85h	---1 1111	---1 1111	---u uuuu
TRISB	86h	1111 1111	1111 1111	uuuu uuuu
EECON1	88h	---0 x000	---0 q000	---0 uuuu
EECON2	89h	---- ----	---- ----	---- ----
PCLATH	8Ah	---0 0000	---0 0000	---u uuuu
INTCON	8Bh	0000 000x	0000 000u	uuuu uuuu ⁽¹⁾

凡例： u = 不変、x = 不定、- = 使用しない、'0' としてリードされる、q = 状態により変化する値

注意 1： INTCON のビットに影響を受けます。(スリープから起動する要因となります)

2：割り込みによるスリープからの起動、かつ、GIE ビットが1のとき、PC には割り込みベクタ (0004h) がロードされます。

3：表 8-3 に各リセット条件によるリセット状態を示しています。

8.4 パワーオンリセット (POR)

(1.2V - 1.7V の範囲で) VDD の立ち上がりが検出されたとき、パワーオンリセットパルスがチップ内で発生します。POR を利用するには、MCLR ピンを VDD に直接 (または抵抗を通して) 接続します。これにより、多くの場合、パワーオンリセットに必要なとされる外付け RC が不要となります。正しくリセットするためには、VDD の立ち上がり時間の規定を満たす必要があります。詳しくは、電気的特性の章を参照してください。

デバイスが (リセット状態から抜けて) 通常動作を開始するとき、デバイスが正常に動作するためには動作パラメータ (電圧、周波数、温度など) は動作範囲内であればいけません。それらが動作範囲外の場合、動作範囲内になるまでデバイスをリセット状態に保持しておく必要があります。

さらに詳しい説明は、アプリケーションノート AN607 「Power-up Trouble Shooting」を参照してください。

VDD の立ち下がりのときは、POR 回路は内部リセットを発生しません。

8.5 パワーアップタイム (PWRT)

パワーアップタイム (PWRT) は、POR から固定の 72 ms (通常) のタイムアウト (T_{PWRT}) を発生します (図 8-10、図 8-11、図 8-12 および図 8-13)。パワーアップタイムは内部の RC オシレータで動作します。PWRT が動作中は、チップをリセット状態に保持します。PWRT は VDD が動作範囲内に達するまでの遅延時間を発生させるために役立ちます。(図 8-13 に PWRT では不足な場合を示します)

コンフィグレーションビット \overline{PWRT} により、PWRT を動作させることができます。デバイスごとの \overline{PWRT} ビットの動作については、図 8-1 または図 8-2 を参照してください。

パワーアップタイムの遅延 T_{PWRT} は VDD、温度、製造上のばらつきにより、チップごとにばらつきます。詳しくは、DC パラメータの章を参照してください。

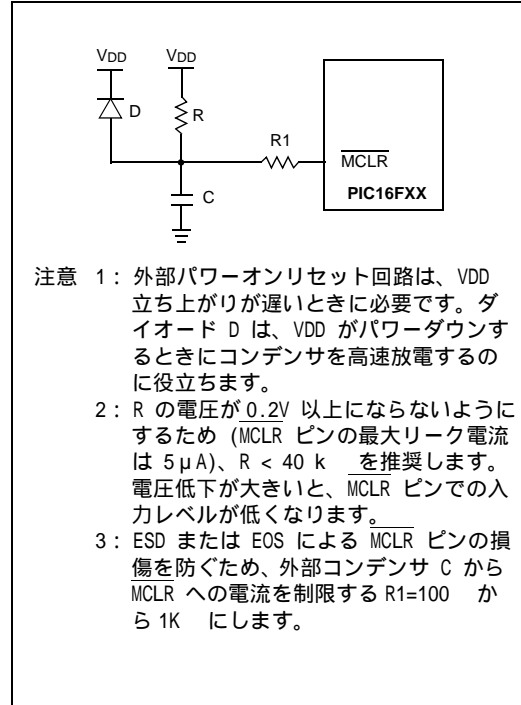
8.6 オシレータスタートアップタイム (OST)

オシレータスタートアップタイム (OST) は PWRT の遅延時間が終了後、1024 オシレータサイクル (OSC1 入力) の遅延時間を発生します (図 8-10、図 8-11、図 8-12 および図 8-13)。これにより、水晶またはレゾネータの発振が安定するまでリセット状態にしておくために役立ちます。

OST タイムアウト (T_{OST}) は XT、LP および HS のモードのときのパワーオンリセットやスリープからの起動のときに起動します。

VDD の立ち上がりが非常に遅いときは、VDD が最後まで立ち上げる前に T_{PWRT} タイムアウトおよび T_{OST} タイムアウトが発生する場合があります。この場合 (図 8-13)、外部パワーオンリセット回路が必要です (図 8-9)。

図 8-9: 外部パワーオンリセット回路 (VDD の立ち上がりが遅いとき用)



PIC16F8X

図 8-10 : 電源 ON 時のタイムアウトシーケンス (MCLR は VDD に接続されない) : ケース 1

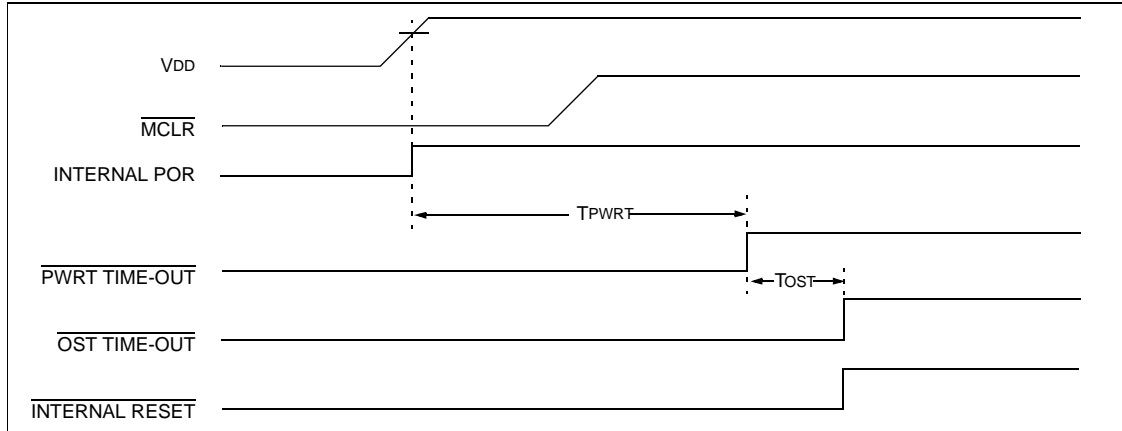
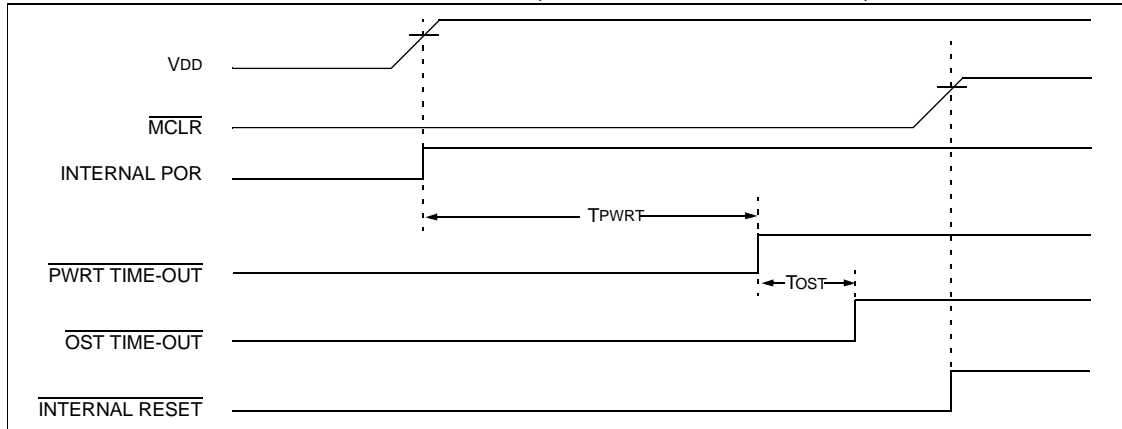


図 8-11 : 電源 ON 時のタイムアウトシーケンス (MCLR は VDD に接続されない) : ケース 2



PIC16F8X

図 8-12 : 電源 ON 時のタイムアウトシーケンス (MCLR は VDD に接続) : VDD の立ち上がりが早い

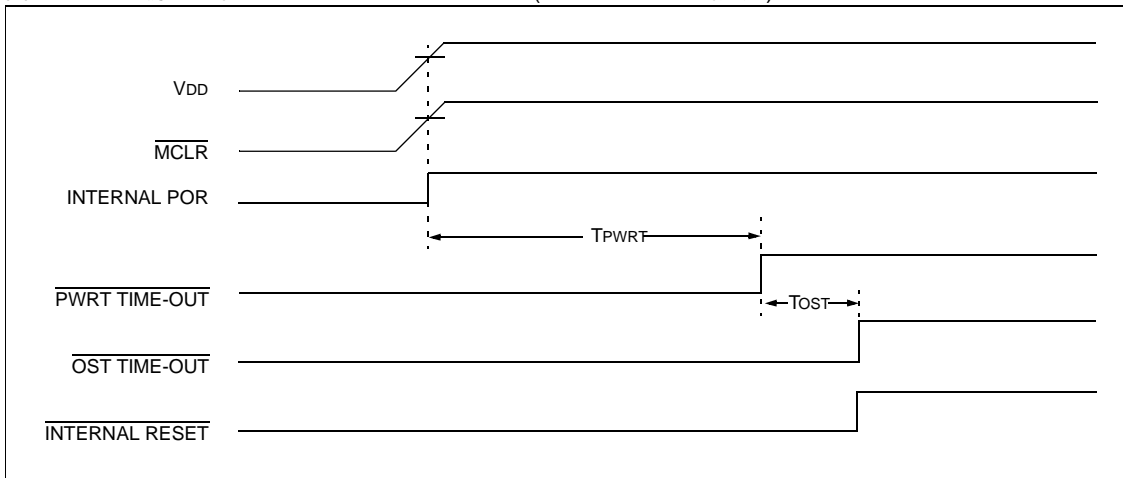
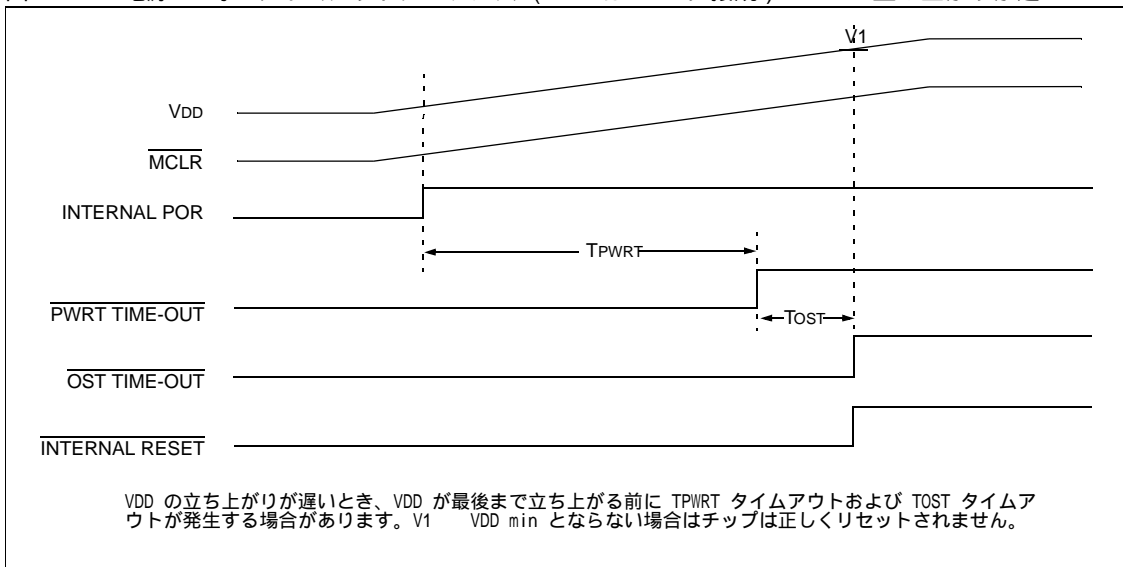


図 8-13 : 電源 ON 時のタイムアウトシーケンス (MCLR は VDD に接続) : VDD の立ち上がりが遅い



PIC16F8X

8.7 タイムアウトシーケンスおよびパワーダウンのステータスビット (TO/PD)

電源 ON のときのタイムアウトシーケンスでは (図 8-10、図 8-11、図 8-12 および図 8-13)、POR の後に PWRT タイムアウトが起動して、その後に OST が動作します。合計のタイムアウトはコンフィグレーションビットのオシレータと PWRT の設定により変化します。例えば、RC モード、かつ、PWRT を動作させない場合は、タイムアウトはありません。

表 8-5 オシレータのコンフィグレーションによるタイムアウト

Oscillator Configuration	Power-up		Wake-up from SLEEP
	PWRT Enabled	PWRT Disabled	
XT, HS, LP	72 ms + 1024TOSC	1024TOSC	1024TOSC
RC	72 ms	—	—

タイムアウトは POR リセットパルスから起動するので、MCLR を Low にある時間しておけば、タイムアウトは終了します。その後、MCLR を High にすると同時にプログラムの実行が開始されます。(図 8-10)。これはテストや並列動作の複数の PIC16F8X デバイスの同期させるため役立ちます。

表 8-6 は TO ビットおよび PD ビットの意味を示します。表 8-3 はいくつかの特別なレジスタのリセット状態、表 8-4 は全レジスタのリセット状態を示します。

表 8-6 ステータスビットおよびその意味

TO	PD	条件
1	1	パワーオンリセット
0	x	違法、POR で TO がセットされる
x	0	違法、POR で PD がセットされる
0	1	WDT リセット (通常動作中)
0	0	WDT によるスリープからの起動
1	1	通常動作中の MCLR リセット
1	0	スリープ中の MCLR リセットまたは 割り込みによるスリープからの起動

8.8 ブラウンアウトによるリセット

ブラウンアウトとは、デバイスの電源電圧が VDD min より低くなり、かつ、ゼロとはならないうちに元に戻る状態です。デバイスは、ブラウンアウトが発生したときはリセットされる必要があります。

ブラウンアウトが発生したときに PIC16F8X デバイスをリセットするために、外部リセット IC や図 8-14 および図 8-15 に示すような外部ブラウンアウト保護回路が必要な場合があります。

図 8-14: ブラウンアウト保護回路 1

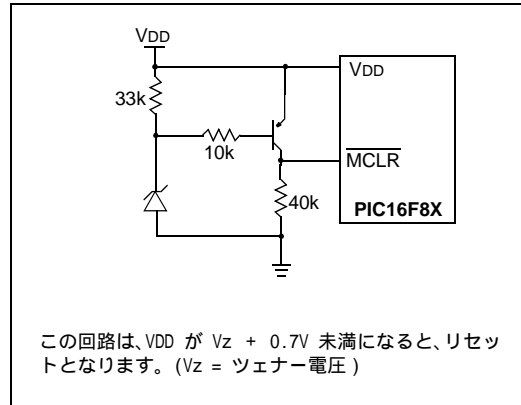
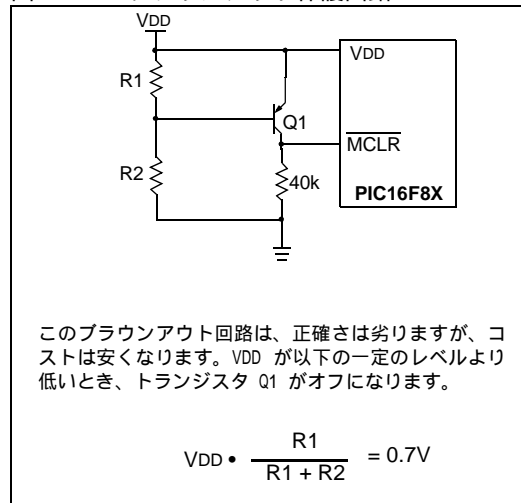


図 8-15: ブラウンアウト保護回路 2



8.9 割り込み

PIC16F8X には 4 本の割り込み要因があります。

- ・ 外部割り込み RB0/INT ピン
- ・ TMR0 オーバフロー割り込み
- ・ PORTB 変化割り込み (ピン RB7:RB4)
- ・ データ EEPROM ライト完了割り込み

割り込みコントロールレジスタ (INTCON) は、個別の割り込み要求フラグビットを記憶します。また、このレジスタは、個別/グローバルの割り込みイネーブルビットもあります。

グローバル割り込みイネーブルビット、GIE (INTCON<7>) は、マスクされていないすべての割り込みを発生許可するか (1 のとき)、またはすべての割り込みを発生禁止します (0 のとき)。個別の割り込みは、INTCON レジスタの対応するビットで発生禁止することができます。GIE ビットはリセットによりクリアされます。

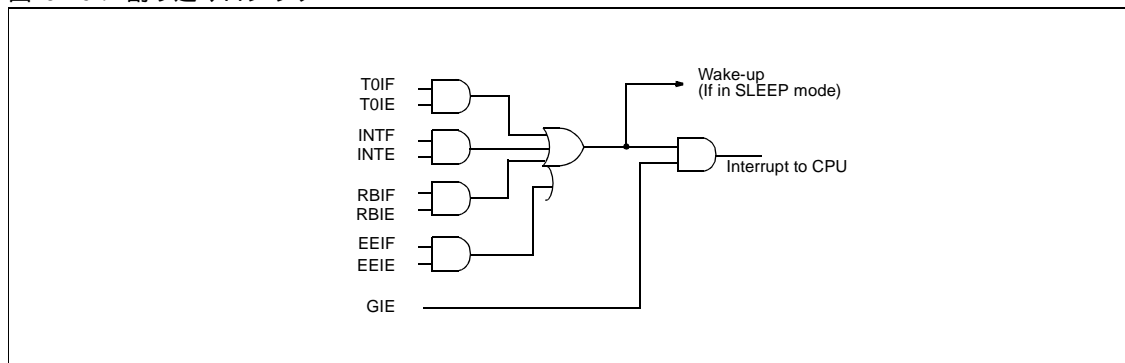
"return from interrupt" 命令、RETFIE は、割り込みルーチンからのリターンと割り込みを発生許可するため GIE ビットを 1 にします。

RB0/INT ピン割り込み、RB ポート変化割り込みおよび TMR0 オーバフロー割り込みフラグは、INTCON レジスタにあります。

割り込みが実行されると、GIE ビットが 0 になり、それ以上の割り込みが発生禁止となり、リターンアドレスがスタックにプッシュされて、PC に 0004h がロードされます。RB0/INT ピンまたは PORTB 変化割り込みなどの外部割り込みのときは、割り込み遅延時間は 3 から 4 命令サイクルです。正確な遅延時間は、割り込みが発生したタイミングによります (図 8-17)。遅延は、1 サイクル命令でも 2 サイクル命令でも同じです。割り込み処理ルーチンで割り込みフラグビットをポーリングすることにより発生した割り込み要因がわかります。割り込みを再度発生許可する前に、割り込みフラグをソフトウェアでクリアする必要があります。そうしないと割り込み要求の無限ループが発生します。

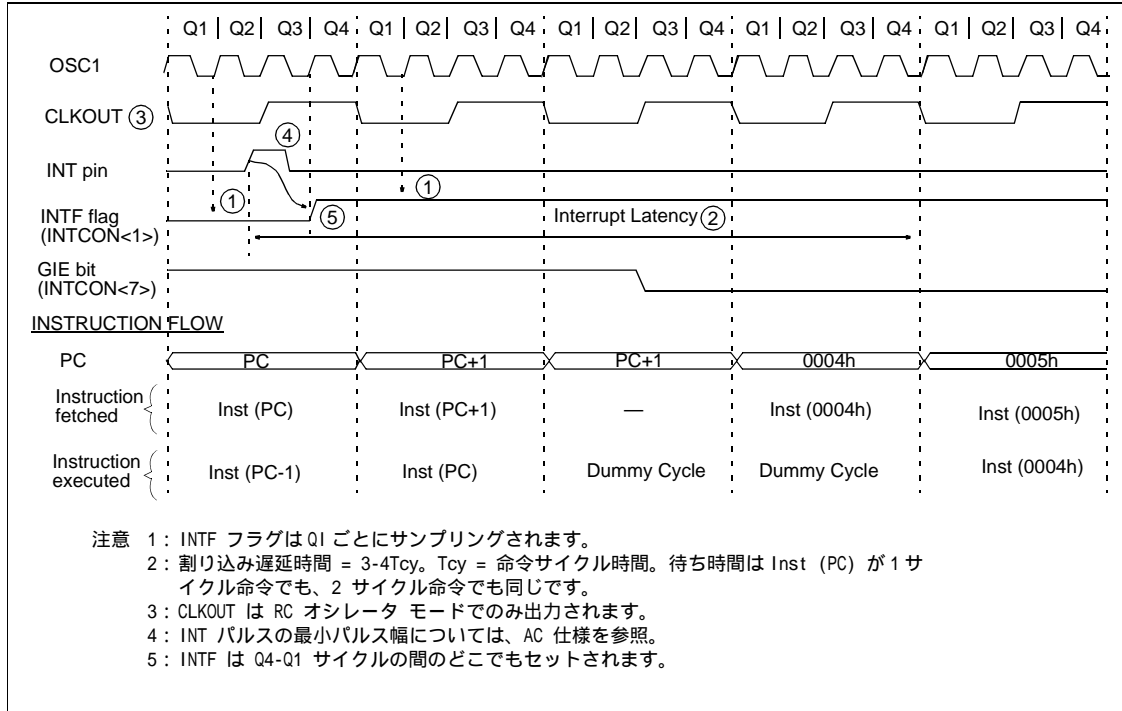
注意 1: 個別の割り込みフラグビットは、それに対応するイネーブルビットや GIE ビットが 0 でもセットされます。

図 8-16: 割り込みロジック



PIC16F8X

図 8-17: INT ピンの割り込みタイミング



8.9.1 INT 割り込み

RBO/INT ピンの外部割り込みはエッジで発生します。INTEDG ビット (OPTION_REG<6>) が 1 のときは立ち上がりエッジ、INTEDG ビットが 0 のときは立ち下がりエッジです。RBO/INT ピンに有効なエッジが入力されたとき、INTF ビット (INTCON<1>) がセットされます。この割り込みを発生許可するには INTE (INTCON<4>) ビットを 1 にします。割り込みを再度発生許可にする前に、割り込み処理ルーチン内でフラグビット INTF をソフトウェアでクリアする必要があります。スリープモードに入る前に INTE ビットが 1 のとき、INT 割り込みはプロセッサをスリープから起動できます (第 8.1 2 項)。GIE ビットが 1 のとき、スリープからの起動後、割り込みベクタにプログラム分岐します。

8.9.2 TMRO 割り込み

TMRO のオーバフロー (FFh ~ 00h) によってフラグビット TOIF (INTCON<2>) がセットされます。TOIF (INTCON<5>) ビットの 1/0 によって、この割り込み発生を許可/禁止することができます (第 6.0 項参照)。

8.9.3 PORT RB 割り込み

PORTB<7:4> の入力変化によって、フラグビット RBIF (INTCON<0>) がセットされます。RBIE (INTCON<3>) ビットの 1/0 によって、この割り込み発生を許可/禁止することができます (第 5.2 項参照)。

注意 1: I/O ピン の変化が認識されるには、パルス幅は TCY 以上でなくてはなりません。

8.10 割り込み中のコンテキストの保存

割り込み中は、PC のリターンアドレスがスタックに保存されます。ほとんどのソフトウェアでは、割り込み中はいくつかのレジスタの値を保存する必要があります（例えば、W レジスタや STATUS レジスタ）。これはソフトウェアで行う必要があります。

例 8-1 は STATUS レジスタおよび W レジスタの値を保存、復帰します。ユーザーが定義するレジスタ、W_TEMP および STATUS_TEMP は、W レジスタおよび STATUS レジスタの割り込み中の保存場所です。

例 8-1 は以下を実行します。

- a) W レジスタを保存する。
- b) STATUS レジスタを STATUS_TEMP に保存する。
- c) 割り込み処理ルーチンを実行する。
- d) STATUS (およびバンク選択ビット) レジスタを元に戻す。
- e) W レジスタを元に戻す。

例 8-1 STATUS レジスタおよび W レジスタの RAM への保存

```
PUSH  MOVWF  W_TEMP      ; Copy W to TEMP register,
      SWAPF  STATUS, W   ; Swap status to be saved into W
      MOVWF  STATUS_TEMP ; Save status to STATUS_TEMP register
ISR   :
      :
      :               ; Interrupt Service Routine
      :               ; should configure Bank as required
      :
POP   SWAPF  STATUS_TEMP, W ; Swap nibbles in STATUS_TEMP register
      :               ; and place result into W
      MOVWF  STATUS      ; Move W into STATUS register
      :               ; (sets bank to original state)
      SWAPF  W_TEMP, F   ; Swap nibbles in W_TEMP and place result in W_TEMP
      SWAPF  W_TEMP, W   ; Swap nibbles in W_TEMP and place result into W
```

注意：Zフラグが影響しないように SWAPF 命令を使用します。

FSR レジスタ、PCLATCH レジスタなども保存が必要な場合があります。

PIC16F8X

8.11 ウォッチドッグタイマ (WDT)

ウォッチドッグタイマは外付け部品不要のフリーランのオンチップ RC オシレータです。この RC オシレータは OSC1/CLKIN ピンの RC オシレータとは別のものです。例えば、SLEEP 命令の実行によりデバイスの OSC1/CLKIN ピンおよび OSC2/CLKOUT ピンのクロックが停止しているときでも WDT は動作します。通常動作中は、WDT タイムアウトはデバイスリセットを発生します。スリープ中は、WDT タイムアウトによるスリープからの起動は通常動作の継続です。ウォッチドッグタイマは、コンフィグレーションワードの WDTE を 0 にすることにより停止できます (第 8.1 項)。

8.11.1 ウォッチドッグタイマの周期

ウォッチドッグタイマのタイムアウト周期は通常 18 ms です (プリスケラなしのとき)。タイムアウト周期は温度、VDD および製造上のばらつきによりデバイスごとにばらつきます (DC 仕様参照)。長いタイムア

ウト周期が必要な場合は、ソフトウェアによる OPTION_REG レジスタの設定で最大 1:128 のプリスケラをウォッチドッグタイマに割り当てることができます。これによって、タイムアウト周期を最大 2.3 秒にできます。

CLRWDT 命令および SLEEP 命令はウォッチドッグタイマとポストスケラ (ウォッチドッグタイマに割り当てられている場合) をクリアしてタイムアウトさせないことでデバイスリセットの発生を遠ざけます。

WDT タイムアウトによって、STATUS レジスタ内の \overline{TO} ビットが 0 になります。

8.11.2 WDT プログラミングの注意点

ワースト条件 (VDD = 最小、温度 = 最大、最大 WDT プリスケラ) では、WDT のタイムアウトが発生するまでに数秒かかる場合があることを考慮しておく必要があります。

図 8-18 : ウォッチドッグタイマのブロック図

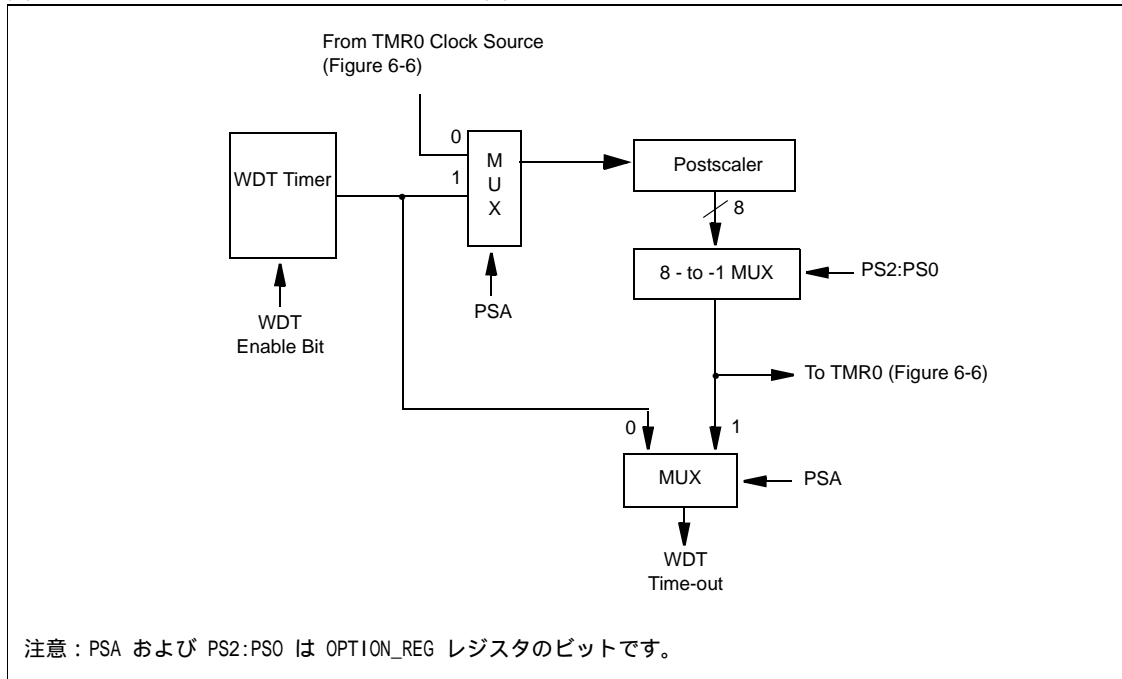


表 8-7 ウォッチドッグタイマに関連したレジスタの概要

アドレス	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	パワーオンリセットによる値	他のすべてのリセットによる値
2007h	Config. bits	(2)	(2)	(2)	(2)	PWRTE ⁽¹⁾	WDTE	FOSC1	FOSC0	(2)	
81h	OPTION_REG	\overline{RBPU}	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0	1111 1111	1111 1111

凡例 : x = 不定。網掛部分はウォッチドッグタイマでは使用しません。

注意 1 : PWRTE ビットでの動作については、図 8-1 および図 8-2 を参照。

2 : コードおよびコードとデータプロテクトビットの動作については、図 8-1、図 8-2 および第 8.13 項を参照。

8.12 パワーダウンモード (スリープ)

デバイスはパワーダウン (スリープ) させて、後にパワーアップ (スリープからの起動) させることができます。

8.12.1 スリープ

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。

ウォッチドッグタイマを動作させている場合はウォッチドッグタイマはクリアされ (動作は継続)、PD ビット (STATUS<3>) は 0、TO ビット (STATUS<4>) は 1、オシレータドライバは発振を停止します。I/O ポートは、SLEEP 命令を実行する前の状態を維持します (High または Low を出力、またはハイインピーダンス)。

スリープモードで消費電力を最も小さくするためには、外部クロックを停止、I/O ピンから外部回路へ電流は流れないようにして、すべての I/O ピンを VDD または VSS に固定します。ハイインピーダンス入力の I/O ピンは、フローティング入力によるスイッチング電流を防ぐため、外部でプルアップまたはプルダウンします。また、TOCKI 入力も VDD または VSS に固定してください。PORTB のオンチッププルアップも考慮する必要があります。

MCLR ピンは論理 High レベル (VIHMC) にしておく必要があります。

WDT タイムアウトによるリセットは、MCLR ピンを Low レベルに駆動しません。

8.12.2 SLEEP からの起動

以下のイベントによって、デバイスは SLEEP から起動します。

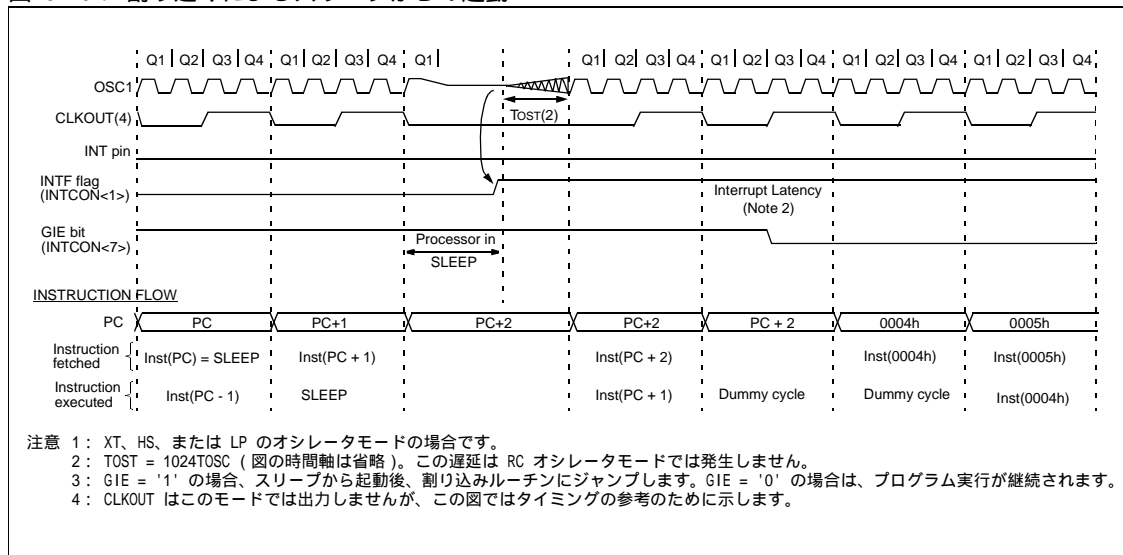
1. MCLR ピンへの外部リセット入力
2. WDT タイムアウト (WDT が動作しているとき)
3. 割り込み (RB0/1NT ピン、RB ポート変化、データ EEPROM ライト完了)

オンチップ 0 クロックがないため、スリープ中は周辺機能は割り込みを発生できません。

最初のイベント (MCLR リセット) は、デバイスリセットを発生させます。次の 2 つのイベントはプログラム実行を継続します。TO ビットおよび PD ビットによって、デバイスリセットの原因を知ることができます。電源 ON のときに 1 になる PD ビットは、SLEEP 命令の実行により 0 になります。WDT タイムアウトが発生する (スリープから起動する) と、TO ビットが 0 になります。

SLEEP 命令を実行しているとき、次の命令 (PC + 1) がフェッチされています。割り込みイベントによってスリープからの起動させるには、対応する割り込みイネーブルビットを 1 にする (割り込みを発生許可する) 必要があります。スリープからの起動は GIE ビットに関係なく発生します。GIE ビットが 0 (発生禁止) のとき、デバイスは SLEEP 命令の次の命令を実行します。GIE ビットが 1 (発生許可) のとき、デバイスは SLEEP 命令の次の命令を実行して、割り込みベクタ (0004h) へプログラム分岐します。SLEEP 命令の次の命令を実行したくないときは、SLEEP 命令の次に NOP を置く必要があります。

図 8-19: 割り込みによるスリープからの起動



PIC16F8X

8.12.3 割り込みによるスリープからの起動

グローバル割り込みが発生禁止 (GIE が 0)、かつ、割り込みイネーブルビットと割り込みフラグビットの両方が 1 となっている割り込み要因があるとき、以下のいずれかが発生します。

- SLEEP 命令の実行前に割り込みが発生した場合は、SLEEP 命令が NOP として完了します。これにより、WDT および WDT ポストスケールはクリアされず、TO ビットはセットされず、PD ビットはクリアされません。
- SLEEP 命令の実行中または実行後に割り込みが発生した場合は、デバイスがすぐにスリープから起動します。スリープからの起動前に SLEEP 命令が実行されます。WDT および WDT ポストスケールはクリアされ、TO ビットがセットされ、PD ビットがクリアされます。

SLEEP 命令実行前に割り込みフラグビットをチェックしても、SLEEP 命令が実行前に割り込みフラグビットがセットされる可能性はあります。SLEEP 命令が実行されたかどうかを判断するには、PD ビットをチェックします。SLEEP 命令は NOP として実行された場合は PD ビットが 1 です。

WDT を確実にクリアするには、SLEEP 命令の前に CLRWDT 命令を実行させる必要があります。

8.13 プログラムベリファイ/コードプロテクト

コードプロテクトが OFF のときは、オンチップのプログラムメモリをリードしてプログラムのベリファイを行えます。

注意： 窓付きデバイスはコードプロテクトを OFF で使用することを推奨します。

8.14 ID ロケーション

4 つの番地 (2000h - 2003h) が ID ロケーションとして、チェックサムまたは他のプログラム識別番号を格納できるようになっています。この番地は通常のプログラム実行中はアクセスできませんが、プログラム/ベリファイ中にリードとライトができます。ID ロケーションは下位 4 ビットのみが使用できます。

ROM デバイスの場合は、この値は ROM コードを指定します。

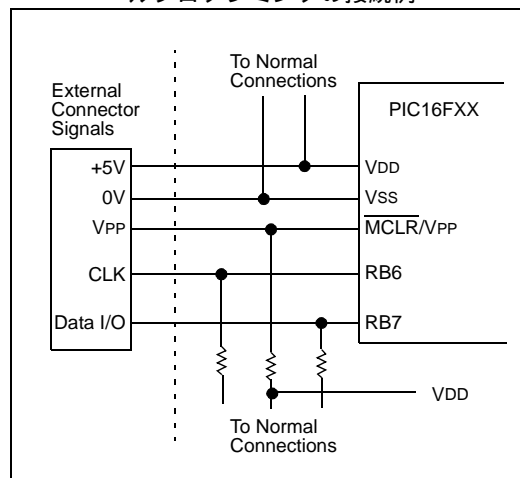
8.15 インサーキットシリアルプログラミング

PIC16F8X マイクロコントローラは、アプリケーション回路上でシリアルプログラミングすることができます。これは、クロックとデータの 2 本と電源、グランド、プログラミング電圧の 3 本を使用することで簡単に行えます。プログラム書き込みされていないデバイスを使用してボードを生産して、製品の出荷直前にマイクロコントローラへプログラム書き込みすることができます。最新のファームウェアや特別仕様のファームウェアをプログラム書き込みすることができます。

MCLR ピンが VIL から VIH に立ち上がるとき、RB6 ピンと RB7 ピンをローにしておくことで、デバイスのプログラム/ベリファイモードに入ります (プログラミング仕様を参照)。RB6 はプログラミングクロックになり、RB7 はプログラミングデータになります。RB6 と RB7 は、このモードではシュミットトリガ入力です。

リセット後、デバイスはプログラミング/ベリファイモードになり、プログラムカウンタ (PC) は 00h 番地をさします。6 ビットのコマンドをデバイスに入力して、ロードまたはリードの命令を使用して 14 ビットのプログラムデータをデバイスへ入力、またはデバイスから出力されます。シリアルプログラミングの詳細については、PIC16F8X EEPROM Memory Programming Specification を参照してください (DS30262)。

図 8-20 標準的なインターサーキットシリアルプログラミングの接続例



ROM デバイスの場合、プログラムメモリとデータ EEPROM メモリの両方をリードできますが、プログラムできるのは データ EEPROM メモリだけです。

9.0 命令セット

PIC16CXX の命令はすべて 14 ビットワードで、命令のタイプを表すオペコード部とその他の命令の動作を表す 1 つ以上のオペランド部からできています。表 9-2 では PIC16CXX の命令セットを、バイト対応、ビット対応、リテラルおよびコントロールに分類しています。表 9-1 には OPCODE フィールドを説明します。

バイト対応命令では、'f' をファイルレジスタ指名字、'd' を結果格納先指名字として使用します。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のアドレスを指定します。

結果格納先指名字では、命令の実行結果を格納する場所を指定します。'd' が 0 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。

ビット対応命令では、ビット番号指名字 '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、ファイルレジスタ指名字 'f' を使って、そのビットが置かれているファイルレジスタのアドレスを指定します。

リテラルおよびコントロール命令では、'k' を使って 8 ビットまたは 11 ビットの定数やリテラルを指定します。

表 9-1: OPCODE フィールドの説明

フィールド	説明
f	ファイルレジスタのアドレス (0x00 ~ 0x7F)
W	ワーキングレジスタ (アキュムレータ)
b	8 ビットファイルレジスタ内のビットアドレス
k	リテラル、定数データまたはラベル
x	無効 (= 0 または 1) アセンブラは x = 0 としてコードを生成。すべてのソフトウェアツールとの互換性を確保するために x = 0 を推奨します。
d	結果格納先指名字 ; d = 0 (結果は W に格納) d = 1 (結果はファイルレジスタ 'f' に格納) デフォルトは d = 1
label	ラベル名
TOS	スタックの最上位
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	グローバル割り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
T \overline{O}	タイムアウトビット
P \overline{D}	パワーダウンビット
dest	結果格納先 (W レジスタまたは指定されているレジスタファイルのロケーション)
[]	オプション
()	内容
→	割り当て先
<>	レジスタビットフィールド
ε	セットを表す
イタリック	ユーザー定義

命令セットは高い直交性を持っていて、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 1 命令サイクルで実行されますが、命令を実行した結果、条件付きテストの結果が真となったり、プログラムカウンタを変更すると、その命令の実行に 2 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 オシレータ周期です。したがって、オシレータ周波数が 4 MHz の場合、命令実行時間は 1 μs になります。命令を実行した結果、条件付きテストが真になったり、プログラムカウンタを変更した場合は、命令実行時間は 2 μs になります。

表 9-2 は、MPASM アセンブラの命令のリストです。

図 9-1 に、命令の一般的なフォーマットを示します。

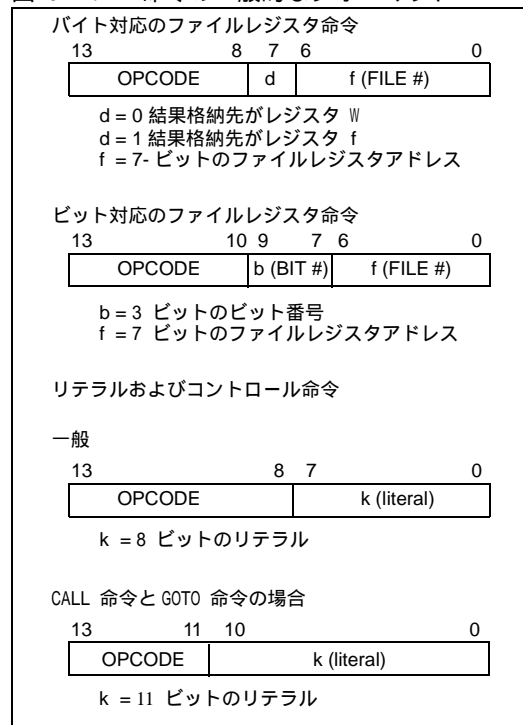
注意: 将来の PIC16CXX 製品との上位互換性を維持するために、OPTION 命令と TRIS 命令は使用しないでください。

例では、次のフォーマットで 16 進数を表します。

0xhh

上記の "h" は 16 進数を表します。

図 9-1: 命令の一般的なフォーマット



PIC16F8X

表 9-2 PIC16FXX の命令セット

ニーモニック オペランド	説明	サイク ル数	14 ビットオペコード				影響され るステ ータス	注意	
			MSb		LSb				
バイト対応のファイルレジスタ命令									
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C,DC,Z	1,2
ANDWF	f, d	AND W with 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 with 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 nibbles in f	1	00	1110	dfff	ffff		1,2
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	1,2
ビット対応のファイルレジスタ命令									
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
リテラルおよびコントロール命令									
ADDLW	k	Add literal and W	1	11	111x	kkkk	kkkk	C,DC,Z	
ANDLW	k	AND literal with 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 with 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	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

注意 1: I/O レジスタがその同じ I/O レジスタにより変更する場合、(MOVf, PORTB, 1 など)、ピンの入力レベルが使用されます。たとえば、入力ピンのデータラッチが "1"、そのピンが外部デバイスにより Low レベルとなっているとき、データラッチには "0" がライトされます。

2: この命令を TMR0 レジスタに対して実行すると (かつ、結果格納先 d の指定が可能な場合には d=1 が指定されている)、TMR0 モジュール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 TMR0 に割り当てられているときのみ)。

3: プログラムカウンタ (PC) を変更したり、条件付きテストの結果が真になると、命令実行は 2 サイクルかかります。2 番目のサイクルは NOP として実行されます。

9.1 命令の説明

ADDLW	Add Literal and W								
Syntax:	[label] ADDLW k								
Operands:	$0 \leq k \leq 255$								
Operation:	$(W) + k \rightarrow (W)$								
Status Affected:	C, DC, Z								
Encoding:	<table border="1"> <tr> <td>11</td> <td>111x</td> <td>kkkk</td> <td>kkkk</td> </tr> </table>	11	111x	kkkk	kkkk				
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 レジスタにライトします。								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1"> <tr> <td>Q1</td> <td>Q2</td> <td>Q3</td> <td>Q4</td> </tr> <tr> <td>Decode</td> <td>Read literal 'k'</td> <td>Process data</td> <td>Write to W</td> </tr> </table>	Q1	Q2	Q3	Q4	Decode	Read literal 'k'	Process data	Write to W
Q1	Q2	Q3	Q4						
Decode	Read literal 'k'	Process data	Write to W						

Example: ADDLW 0x15

命令実行前
W = 0x10

命令実行後
W = 0x25

ANDLW	AND Literal with W								
Syntax:	[label] ANDLW k								
Operands:	$0 \leq k \leq 255$								
Operation:	$(W) .AND. (k) \rightarrow (W)$								
Status Affected:	Z								
Encoding:	<table border="1"> <tr> <td>11</td> <td>1001</td> <td>kkkk</td> <td>kkkk</td> </tr> </table>	11	1001	kkkk	kkkk				
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 を行います。この結果を W レジスタにライトします。								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1"> <tr> <td>Q1</td> <td>Q2</td> <td>Q3</td> <td>Q4</td> </tr> <tr> <td>Decode</td> <td>Read literal "k"</td> <td>Process data</td> <td>Write to W</td> </tr> </table>	Q1	Q2	Q3	Q4	Decode	Read literal "k"	Process data	Write to W
Q1	Q2	Q3	Q4						
Decode	Read literal "k"	Process data	Write to W						

Example ANDLW 0x5F

命令実行前
W = 0xA3

命令実行後
W = 0x03

ADDWF	Add W and f								
Syntax:	[label] ADDWF f,d								
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$								
Operation:	$(W) + (f) \rightarrow (\text{destination})$								
Status Affected:	C, DC, Z								
Encoding:	<table border="1"> <tr> <td>00</td> <td>0111</td> <td>dfff</td> <td>ffff</td> </tr> </table>	00	0111	dfff	ffff				
00	0111	dfff	ffff						
Description:	W レジスタの内容をレジスタ "f" に加えます。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" にライトします。								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1"> <tr> <td>Q1</td> <td>Q2</td> <td>Q3</td> <td>Q4</td> </tr> <tr> <td>Decode</td> <td>Read register 'f'</td> <td>Process data</td> <td>Write to destination</td> </tr> </table>	Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process data	Write to destination
Q1	Q2	Q3	Q4						
Decode	Read register 'f'	Process data	Write to destination						

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:	$0 \leq f \leq 127$ $d \in [0,1]$								
Operation:	$(W) .AND. (f) \rightarrow (\text{destination})$								
Status Affected:	Z								
Encoding:	<table border="1"> <tr> <td>00</td> <td>0101</td> <td>dfff</td> <td>ffff</td> </tr> </table>	00	0101	dfff	ffff				
00	0101	dfff	ffff						
Description:	W レジスタとレジスタ "f" の AND を行います。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" にライトします。								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1"> <tr> <td>Q1</td> <td>Q2</td> <td>Q3</td> <td>Q4</td> </tr> <tr> <td>Decode</td> <td>Read register 'f'</td> <td>Process data</td> <td>Write to destination</td> </tr> </table>	Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process data	Write to destination
Q1	Q2	Q3	Q4						
Decode	Read register 'f'	Process data	Write to destination						

Example ANDWF FSR, 1

命令実行前
W = 0x17
FSR = 0xC2

命令実行後
W = 0x17
FSR = 0x02

PIC16F8X

BCF	Bit Clear f								
Syntax:	[label] BCF f,b								
Operands:	0 ≤ f ≤ 127 0 ≤ b ≤ 7								
Operation:	0 → (f)								
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" をクリアします。								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1"> <tr> <th>Q1</th> <th>Q2</th> <th>Q3</th> <th>Q4</th> </tr> <tr> <td>Decode</td> <td>Read register 'f'</td> <td>Process data</td> <td>Write register 'f'</td> </tr> </table>	Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process data	Write register 'f'
Q1	Q2	Q3	Q4						
Decode	Read register 'f'	Process data	Write register 'f'						

Example BCF FLAG_REG, 7

命令実行前
 FLAG_REG = 0xC7

命令実行後
 FLAG_REG = 0x47

BSF	Bit Set f								
Syntax:	[label] BSF f,b								
Operands:	0 ≤ f ≤ 127 0 ≤ b ≤ 7								
Operation:	1 → (f)								
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" がセットします。								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1"> <tr> <th>Q1</th> <th>Q2</th> <th>Q3</th> <th>Q4</th> </tr> <tr> <td>Decode</td> <td>Read register 'f'</td> <td>Process data</td> <td>Write register 'f'</td> </tr> </table>	Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process data	Write register 'f'
Q1	Q2	Q3	Q4						
Decode	Read register 'f'	Process data	Write register 'f'						

Example BSF FLAG_REG, 7

命令実行前
 FLAG_REG = 0x0A

命令実行後
 FLAG_REG = 0x8A

BTFSC	Bit Test, Skip if Clear								
Syntax:	[label] BTFSC f,b								
Operands:	0 ≤ f ≤ 127 0 ≤ b ≤ 7								
Operation:	skip if (f) = 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 1 の場合、次の命令を実行します。ビット "b" が 0 の場合は、次の命令を破棄して、かわりに NOP を実行します。2 サイクル命令になります。								
Words:	1								
Cycles:	1(2)								
Q Cycle Activity:	<table border="1"> <tr> <th>Q1</th> <th>Q2</th> <th>Q3</th> <th>Q4</th> </tr> <tr> <td>Decode</td> <td>Read register 'f'</td> <td>Process data</td> <td>No-Operation</td> </tr> </table>	Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process data	No-Operation
Q1	Q2	Q3	Q4						
Decode	Read register 'f'	Process data	No-Operation						

If Skip: (2nd Cycle)

Q1	Q2	Q3	Q4
No-Operation	No-Operation	No-Operation	No-Operation

Example

```

HERE     BTFSC   FLAG, 1
FALSE    GOTO    PROCESS_CODE
TRUE     :
          :
          :
命令実行前
          PC =    address HERE
命令実行後
          if FLAG<1> = 0,
          PC =    address TRUE
          if FLAG<1> = 1,
          PC =    address FALSE

```


PIC16F8X

BTFSS **Bit Test f, Skip if Set**

Syntax: `[label] BTFSS f,b`

Operands: $0 \leq f \leq 127$
 $0 \leq b < 7$

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 0 の場合、次の命令を実行します。
 "b" が 1 の場合は、次の命令を破棄して、かわりに NOP を実行します。2 サイクル命令になります。

Words: 1

Cycles: 1(2)

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process data	No-Operation

If Skip: (2nd Cycle)

Q1	Q2	Q3	Q4
No-Operation	No-Operation	No-Operation	No-Operation

Example

```

HERE    BTFSS    FLAG, 1
FALSE    GOTO    PROCESS_CODE
TRUE   
        
        
        
命令実行前
PC =    address HERE
命令実行後
if FLAG<1> = 0,
PC =    address FALSE
if FLAG<1> = 1,
PC =    address TRUE
  
```

CALL **Call Subroutine**

Syntax: `[label] CALL k`

Operands: $0 \leq k \leq 2047$

Operation: $(PC)+1 \rightarrow TOS,$
 $k \rightarrow PC<10:0>,$
 $(PCLATH<4:3>) \rightarrow PC<12:11>$

Status Affected: None

Encoding:

10	0kkk	kkkk	kkkk
----	------	------	------

Description: サブルーチンをコールします。まず、リターンアドレス (PC+1) をスタックにプッシュして、11 ビットのリテラルアドレスを PC のビット <10:0> にロードします。PC の上位ビットは PCLATH からロードします。CALL は 2 サイクルの命令です。

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k', Push PC to Stack	Process data	Write to PC

1st Cycle

2nd Cycle

No-Operation	No-Operation	No-Operation	No-Operation
--------------	--------------	--------------	--------------

Example

```

HERE    CALL    THERE
命令実行前
PC =    Address HERE
命令実行後
PC =    Address THERE
TOS =    Address HERE+1
  
```

PIC16F8X

CLRF	Clear f								
Syntax:	[label] CLRF f								
Operands:	$0 \leq f \leq 127$								
Operation:	00h → (f) 1 → Z								
Status Affected:	Z								
Encoding:	<table border="1"> <tr> <td>00</td> <td>0001</td> <td>1fff</td> <td>ffff</td> </tr> </table>	00	0001	1fff	ffff				
00	0001	1fff	ffff						
Description:	レジスタ "f" の内容をクリアして、Z ビットをセットします。								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1"> <thead> <tr> <th>Q1</th> <th>Q2</th> <th>Q3</th> <th>Q4</th> </tr> </thead> <tbody> <tr> <td>Decode</td> <td>Read register 'f'</td> <td>Process data</td> <td>Write register 'f'</td> </tr> </tbody> </table>	Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process data	Write register 'f'
Q1	Q2	Q3	Q4						
Decode	Read register 'f'	Process data	Write register 'f'						

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								
Syntax:	[label] CLRW								
Operands:	None								
Operation:	00h → (W) 1 → Z								
Status Affected:	Z								
Encoding:	<table border="1"> <tr> <td>00</td> <td>0001</td> <td>0xxxx</td> <td>xxxxx</td> </tr> </table>	00	0001	0xxxx	xxxxx				
00	0001	0xxxx	xxxxx						
Description:	W レジスタをクリアして、Z ビットをセットします。								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1"> <thead> <tr> <th>Q1</th> <th>Q2</th> <th>Q3</th> <th>Q4</th> </tr> </thead> <tbody> <tr> <td>Decode</td> <td>No-Operation</td> <td>Process data</td> <td>Write to W</td> </tr> </tbody> </table>	Q1	Q2	Q3	Q4	Decode	No-Operation	Process data	Write to W
Q1	Q2	Q3	Q4						
Decode	No-Operation	Process data	Write to W						

Example

```

CLRW
命令実行前
W = 0x5A
命令実行後
W = 0x00
Z = 1
    
```

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:	<table border="1"> <tr> <td>00</td> <td>0000</td> <td>0110</td> <td>0100</td> </tr> </table>	00	0000	0110	0100				
00	0000	0110	0100						
Description:	CLRWDT 命令は WDT をリセットします。また、WDT のプリスケールモリをセットします。ステータス ビット \overline{TO} および \overline{PD} をセットします。								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1"> <thead> <tr> <th>Q1</th> <th>Q2</th> <th>Q3</th> <th>Q4</th> </tr> </thead> <tbody> <tr> <td>Decode</td> <td>No-Operation</td> <td>Process data</td> <td>Clear WDT Counter</td> </tr> </tbody> </table>	Q1	Q2	Q3	Q4	Decode	No-Operation	Process data	Clear WDT Counter
Q1	Q2	Q3	Q4						
Decode	No-Operation	Process data	Clear WDT Counter						

Example

```

CLRWDT
命令実行前
WDT counter = ?
命令実行後
WDT counter = 0x00
WDT prescaler = 0
 $\overline{TO}$  = 1
 $\overline{PD}$  = 1
    
```

PIC16F8X

COMF Complement f

Syntax: [label] COMF f,d

Operands: $0 \leq f \leq 127$
 $d \in [0,1]$

Operation: $(f) \rightarrow (\text{destination})$

Status Affected: Z

Encoding:

00	1001	dfff	ffff
----	------	------	------

Description: レジスタ "f" の内容の補数をとります。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" にライトします。

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process data	Write to destination

Example

```
COMF    REG1, 0

命令実行前
REG1    = 0x13
命令実行後
REG1    = 0x13
W       = 0xEC
```

DECFSZ Decrement f

Syntax: [label] DECFSZ f,d

Operands: $0 \leq f \leq 127$
 $d \in [0,1]$

Operation: $(f) - 1 \rightarrow (\text{destination})$

Status Affected: Z

Encoding:

00	0011	dfff	ffff
----	------	------	------

Description: レジスタ "f" をデクリメントします。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" にライトします。

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process data	Write to destination

Example

```
DECFSZ  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: $0 \leq f \leq 127$
 $d \in [0,1]$

Operation: $(f) - 1 \rightarrow (\text{destination});$
skip if result = 0

Status Affected: None

Encoding:

00	1011	dfff	ffff
----	------	------	------

Description: レジスタ "f" をデクリメントします。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" にライトします。結果が 1 の場合は、次の命令を実行します。結果が 0 の場合は、次の命令を破棄、かわりに NOP を実行して、2 サイクル命令になります。

Words: 1

Cycles: 1(2)

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process data	Write to destination

If Skip: (2nd Cycle)

Q1	Q2	Q3	Q4
No-Operation	No-Operation	No-Operation	No-Operation

Example

```
HERE    DECFSZ  CNT, 1
        GOTO    LOOP
CONTINUE
        .
        .
        .

命令実行前
PC     = address HERE
命令実行後
CNT    = CNT - 1
if CNT = 0,
PC     = address CONTINUE
if CNT ≠ 0,
PC     = address HERE+1
```

PIC16F8X

GOTO **Unconditional Branch**

Syntax: [*label*] GOTO *k*

Operands: $0 \leq k \leq 2047$

Operation: $k \rightarrow PC<10:0>$
 $PCLATH<4:3> \rightarrow PC<12:11>$

Status Affected: None

Encoding:

10	1kkk	kkkk	kkkk
----	------	------	------

Description: GOTO は無条件の分岐命令です。11 ビットのリテラルアドレスを PC のビット <10:0> にロードします。PC の上位ビットへは PCLATH <4:3> をロードします。GOTO は 2 サイクルの命令です。

Words: 1

Cycles: 2

Q Cycle Activity:

	Q1	Q2	Q3	Q4
1st Cycle	Decode	Read literal 'k'	Process data	Write to PC
2nd Cycle	No-Operation	No-Operation	No-Operation	No-Operation

Example GOTO THERE

 命令実行後
 PC = Address THERE

INCF **Increment f**

Syntax: [*label*] INCF *f,d*

Operands: $0 \leq f \leq 127$
 $d \in [0,1]$

Operation: $(f) + 1 \rightarrow (\text{destination})$

Status Affected: Z

Encoding:

00	1010	dfff	ffff
----	------	------	------

Description: レジスタ "f" の内容をインクリメントします。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" にライトします。

Words: 1

Cycles: 1

Q Cycle Activity:

	Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process data	Write to destination	

Example INCF CNT, 1

命令実行前
 CNT = 0xFF
 Z = 0

命令実行後
 CNT = 0x00
 Z = 1

PIC16F8X

INCFSZ Increment f, Skip if 0

Syntax: [label] INCFSZ f,d

Operands: $0 \leq f \leq 127$
 $d \in [0,1]$

Operation: $(f) + 1 \rightarrow (\text{destination})$,
 skip if result = 0

Status Affected: None

Encoding:

00	1111	dfff	ffff
----	------	------	------

Description: レジスタ "f" の内容をインクリメントします。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" にライトします。結果が 1 の場合は、次の命令を実行します。結果が 0 の場合は、次の命令を破棄、かわりに NOP を実行して、2 サイクル命令になります。

Words: 1

Cycles: 1(2)

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process data	Write to destination

If Skip: (2nd Cycle)

Q1	Q2	Q3	Q4
No-Operation	No-Operation	No-Operation	No-Operation

Example

```
HERE      INCFSZ    CNT, 1
          GOTO     LOOP
CONTINUE  •
          •
          •
```

命令実行前

PC = address HERE

命令実行後

```
CNT = CNT + 1
if CNT= 0,
PC = address CONTINUE
if CNT≠ 0,
PC = address HERE +1
```

IORLW Inclusive OR Literal with W

Syntax: [label] IORLW k

Operands: $0 \leq k \leq 255$

Operation: $(W) .OR. k \rightarrow (W)$

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 を行います。この結果を W レジスタにライトします。

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process data	Write to W

Example

```
IORLW 0x35
```

命令実行前

W = 0x9A

命令実行後

W = 0xBF
 Z = 1

PIC16F8X

IORWF	Inclusive OR W with f								
Syntax:	[<i>label</i>] IORWF <i>f</i> , <i>d</i>								
Operands:	0 ≤ <i>f</i> ≤ 127 <i>d</i> ∈ [0,1]								
Operation:	(W) .OR. (<i>f</i>) → (destination)								
Status Affected:	\bar{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 OR を行います。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" にライトします。								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1"> <tr> <td>Q1</td> <td>Q2</td> <td>Q3</td> <td>Q4</td> </tr> <tr> <td>Decode</td> <td>Read register 'f'</td> <td>Process data</td> <td>Write to destination</td> </tr> </table>	Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process data	Write to destination
Q1	Q2	Q3	Q4						
Decode	Read register 'f'	Process data	Write to destination						

Example IORWF RESULT, 0

命令実行前

RESULT = 0x13
W = 0x91

命令実行後

RESULT = 0x13
W = 0x93
Z = 1

MOVF	Move f								
Syntax:	[<i>label</i>] MOVF <i>f</i> , <i>d</i>								
Operands:	0 ≤ <i>f</i> ≤ 127 <i>d</i> ∈ [0,1]								
Operation:	(<i>f</i>) → (destination)								
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" の内容を結果格納先 "d" に移動します。d=0 であれば結果格納先は W レジスタです。d=1 であれば結果格納先は同じファイルレジスタ "f" です。d=1 は、ステータスフラグ Z が影響するので、ファイル レジスタのテストに便利です。								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1"> <tr> <td>Q1</td> <td>Q2</td> <td>Q3</td> <td>Q4</td> </tr> <tr> <td>Decode</td> <td>Read register 'f'</td> <td>Process data</td> <td>Write to destination</td> </tr> </table>	Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process data	Write to destination
Q1	Q2	Q3	Q4						
Decode	Read register 'f'	Process data	Write to destination						

Example MOVF FSR, 0

命令実行後

W = value in FSR register
Z = 1

MOVLW	Move Literal to W								
Syntax:	[<i>label</i>] MOVLW <i>k</i>								
Operands:	0 ≤ <i>k</i> ≤ 255								
Operation:	<i>k</i> → (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								
Q Cycle Activity:	<table border="1"> <tr> <td>Q1</td> <td>Q2</td> <td>Q3</td> <td>Q4</td> </tr> <tr> <td>Decode</td> <td>Read literal 'k'</td> <td>Process data</td> <td>Write to W</td> </tr> </table>	Q1	Q2	Q3	Q4	Decode	Read literal 'k'	Process data	Write to W
Q1	Q2	Q3	Q4						
Decode	Read literal 'k'	Process data	Write to W						

Example

MOVLW 0x5A

命令実行後

W = 0x5A

MOVWF	Move W to f								
Syntax:	[<i>label</i>] MOVWF <i>f</i>								
Operands:	0 ≤ <i>f</i> ≤ 127								
Operation:	(W) → (<i>f</i>)								
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								
Q Cycle Activity:	<table border="1"> <tr> <td>Q1</td> <td>Q2</td> <td>Q3</td> <td>Q4</td> </tr> <tr> <td>Decode</td> <td>Read register 'f'</td> <td>Process data</td> <td>Write register 'f'</td> </tr> </table>	Q1	Q2	Q3	Q4	Decode	Read register 'f'	Process data	Write register 'f'
Q1	Q2	Q3	Q4						
Decode	Read register 'f'	Process data	Write register 'f'						

Example

MOVWF OPTION_REG

命令実行前

OPTION = 0xFF
W = 0x4F

命令実行後

OPTION = 0x4F
W = 0x4F

PIC16F8X

NOP	No Operation										
Syntax:	[<i>label</i>] NOP										
Operands:	None										
Operation:	No operation										
Status Affected:	None										
Encoding:	<table border="1"> <tr> <td>00</td> <td>0000</td> <td>0xx0</td> <td>0000</td> </tr> </table>	00	0000	0xx0	0000						
00	0000	0xx0	0000								
Description:	何も行いません。										
Words:	1										
Cycles:	1										
Q Cycle Activity:	<table border="1"> <thead> <tr> <th></th> <th>Q1</th> <th>Q2</th> <th>Q3</th> <th>Q4</th> </tr> </thead> <tbody> <tr> <td>Decode</td> <td>No-Operation</td> <td>No-Operation</td> <td>No-Operation</td> <td>No-Operation</td> </tr> </tbody> </table>		Q1	Q2	Q3	Q4	Decode	No-Operation	No-Operation	No-Operation	No-Operation
	Q1	Q2	Q3	Q4							
Decode	No-Operation	No-Operation	No-Operation	No-Operation							
Example	NOP										

RETFIE	Return from Interrupt															
Syntax:	[<i>label</i>] RETFIE															
Operands:	None															
Operation:	TOS → PC, 1 → GIE															
Status Affected:	None															
Encoding:	<table border="1"> <tr> <td>00</td> <td>0000</td> <td>0000</td> <td>1001</td> </tr> </table>	00	0000	0000	1001											
00	0000	0000	1001													
Description:	割り込みからの復帰。スタックをポップして、スタックの最上位 (TOS) を PC にロードします。GIE (グローバル割り込みイネーブル) ビットをセットして割り込みをイネーブルにします (INTCON<7>)。2 サイクル命令です。															
Words:	1															
Cycles:	2															
Q Cycle Activity:	<table border="1"> <thead> <tr> <th></th> <th>Q1</th> <th>Q2</th> <th>Q3</th> <th>Q4</th> </tr> </thead> <tbody> <tr> <td>1st Cycle</td> <td>Decode</td> <td>No-Operation</td> <td>Set the GIE bit</td> <td>Pop from the Stack</td> </tr> <tr> <td>2nd Cycle</td> <td>No-Operation</td> <td>No-Operation</td> <td>No-Operation</td> <td>No-Operation</td> </tr> </tbody> </table>		Q1	Q2	Q3	Q4	1st Cycle	Decode	No-Operation	Set the GIE bit	Pop from the Stack	2nd Cycle	No-Operation	No-Operation	No-Operation	No-Operation
	Q1	Q2	Q3	Q4												
1st Cycle	Decode	No-Operation	Set the GIE bit	Pop from the Stack												
2nd Cycle	No-Operation	No-Operation	No-Operation	No-Operation												
Example	RETFIE 命令実行後 PC = TOS GIE = 1															

OPTION	Load Option Register				
Syntax:	[<i>label</i>] OPTION				
Operands:	None				
Operation:	(W) → OPTION				
Status Affected:	None				
Encoding:	<table border="1"> <tr> <td>00</td> <td>0000</td> <td>0110</td> <td>0010</td> </tr> </table>	00	0000	0110	0010
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 製品との互換性を保つための命令です。OPTION レジスタ はリード、ライト可能なので直接アドレス指定できます。				
Words:	1				
Cycles:	1				
Example	<table border="1"> <tr> <td>将来の PIC16CXX 製品との上位互換性を維持するため、この命令は使用しないでください。</td> </tr> </table>	将来の PIC16CXX 製品との上位互換性を維持するため、この命令は使用しないでください。			
将来の PIC16CXX 製品との上位互換性を維持するため、この命令は使用しないでください。					

PIC16F8X

RETLW	Return with Literal in W															
Syntax:	[<i>label</i>] RETLW k															
Operands:	0 ≤ k ≤ 255															
Operation:	k → (W); TOS → PC															
Status Affected:	None															
Encoding:	<table border="1"> <tr> <td>11</td> <td>01xxx</td> <td>kkkk</td> <td>kkkk</td> </tr> </table>	11	01xxx	kkkk	kkkk											
11	01xxx	kkkk	kkkk													
Description:	8 ビットのリテラル 'k' を W レジスタにロードして、スタックの最上位（リターンアドレス）をプログラムカウンタへロードします。2 サイクル命令です。															
Words:	1															
Cycles:	2															
Q Cycle Activity:	<table border="1"> <thead> <tr> <th></th> <th>Q1</th> <th>Q2</th> <th>Q3</th> <th>Q4</th> </tr> </thead> <tbody> <tr> <td>1st Cycle</td> <td>Decode</td> <td>Read literal 'k'</td> <td>No-Operation</td> <td>Write to W, Pop from the Stack</td> </tr> <tr> <td>2nd Cycle</td> <td>No-Operation</td> <td>No-Operation</td> <td>No-Operation</td> <td>No-Operation</td> </tr> </tbody> </table>		Q1	Q2	Q3	Q4	1st Cycle	Decode	Read literal 'k'	No-Operation	Write to W, Pop from the Stack	2nd Cycle	No-Operation	No-Operation	No-Operation	No-Operation
	Q1	Q2	Q3	Q4												
1st Cycle	Decode	Read literal 'k'	No-Operation	Write to W, Pop from the Stack												
2nd Cycle	No-Operation	No-Operation	No-Operation	No-Operation												

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

RETURN	Return from Subroutine															
Syntax:	[<i>label</i>] RETURN															
Operands:	None															
Operation:	TOS → PC															
Status Affected:	None															
Encoding:	<table border="1"> <tr> <td>00</td> <td>0000</td> <td>0000</td> <td>1000</td> </tr> </table>	00	0000	0000	1000											
00	0000	0000	1000													
Description:	サブルーチンからの復帰。スタックをポップして、次にスタックの最上位 (TOS) をプログラムカウンタにロードします。2 サイクルの命令です。															
Words:	1															
Cycles:	2															
Q Cycle Activity:	<table border="1"> <thead> <tr> <th></th> <th>Q1</th> <th>Q2</th> <th>Q3</th> <th>Q4</th> </tr> </thead> <tbody> <tr> <td>1st Cycle</td> <td>Decode</td> <td>No-Operation</td> <td>No-Operation</td> <td>Pop from the Stack</td> </tr> <tr> <td>2nd Cycle</td> <td>No-Operation</td> <td>No-Operation</td> <td>No-Operation</td> <td>No-Operation</td> </tr> </tbody> </table>		Q1	Q2	Q3	Q4	1st Cycle	Decode	No-Operation	No-Operation	Pop from the Stack	2nd Cycle	No-Operation	No-Operation	No-Operation	No-Operation
	Q1	Q2	Q3	Q4												
1st Cycle	Decode	No-Operation	No-Operation	Pop from the Stack												
2nd Cycle	No-Operation	No-Operation	No-Operation	No-Operation												

Example

```
RETURN
命令実行後
PC = TOS
```


PIC16F8X

RLF Rotate Left f through Carry

Syntax: [*label*] RLF *f*,*d*

Operands: $0 \leq f \leq 127$
 $d \in [0,1]$

Operation: See description below

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" にライトします。



Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process data	Write to destination

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

RRF Rotate Right f through Carry

Syntax: [*label*] RRF *f*,*d*

Operands: $0 \leq f \leq 127$
 $d \in [0,1]$

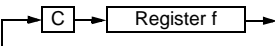
Operation: See description below

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" にライトします。



Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process data	Write to destination

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

PIC16F8X

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、タイムアウトステータスビット (\overline{TO}) をセット、ウォッチドッグタイマとそのプリスケアラをクリアします。プロセッサは SLEEP モードに入ります。オシレータは停止します。詳細は、14.8 項を参照。

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No-Operation	No-Operation	Go to Sleep

Example: SLEEP

SUBLW

Subtract W from Literal

Syntax: [*label*] SUBLW k

Operands: $0 \leq k \leq 255$

Operation: $k - (W) \rightarrow (W)$

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 レジスタにライトします。

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process data	Write to W

Example 1: SUBLW 0x02

命令実行前

W = 1
C = ?
Z = ?

命令実行後

W = 1
C = 1; result is positive
Z = 0

Example 2: SUBLW 0x02

命令実行前

W = 2
C = ?
Z = ?

命令実行後

W = 0
C = 1; result is zero
Z = 1

Example 3: SUBLW 0xFF

命令実行前

W = 3
C = ?
Z = ?

命令実行後

W = 0xFF
C = 0; result is negative
Z = 0

PIC16F8X

SUBWF Subtract W from f

Syntax: [label] SUBWF f,d

Operands: $0 \leq f \leq 127$
 $d \in [0,1]$

Operation: (f) - (W) → (destination)

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" にライトします。

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process data	Write to destination

Example 1: SUBWF REG1, 1

命令実行前

REG1 = 3
W = 2
C = ?
Z = ?

命令実行後

REG1 = 1
W = 2
C = 1; result is positive
Z = 0

Example 2: 命令実行前

REG1 = 2
W = 2
C = ?
Z = ?

命令実行後

REG1 = 0
W = 2
C = 1; result is zero
Z = 1

Example 3: 命令実行前

REG1 = 1
W = 2
C = ?
Z = ?

命令実行後

REG1 = 0xFF
W = 2
C = 0; result is negative
Z = 0

SWAPF Swap Nibbles in f

Syntax: [label] SWAPF f,d

Operands: $0 \leq f \leq 127$
 $d \in [0,1]$

Operation: (f<3:0>) → (destination<7:4>),
(f<7:4>) → (destination<3:0>)

Status Affected: None

Encoding:

00	1110	dfff	ffff
----	------	------	------

Description: レジスタ "f" の上位ニブルと下位ニブルを入れ替えます。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" にライトします。

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process data	Write to destination

Example SWAPF REG, 0

命令実行前

REG1 = 0xA5

命令実行後

REG1 = 0xA5
W = 0x5A

TRIS Load TRIS Register

Syntax: [label] TRIS f

Operands: $5 \leq f \leq 7$

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の命令です。TRIS レジスタはリード、ライト可能なので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、この命令は使用しないでください。

PIC16F8X

XORLW Exclusive OR Literal with W

Syntax: `[label] XORLW k`
 Operands: $0 \leq k \leq 255$
 Operation: $(W) \text{ .XOR. } k \rightarrow (W)$
 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 をとり、その結果を W レジスタにライトします。
 Words: 1
 Cycles: 1
 Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process data	Write to W

Example: `XORLW 0xAF`
 命令実行前
 W = 0xB5
 命令実行後
 W = 0x1A

XORWF Exclusive OR W with f

Syntax: `[label] XORWF f,d`
 Operands: $0 \leq f \leq 127$
 $d \in [0,1]$
 Operation: $(W) \text{ .XOR. } (f) \rightarrow (\text{destination})$
 Status Affected: Z
 Encoding:

00	0110	dfff	ffff
----	------	------	------

 Description: W レジスタの内容とレジスタ "f" との XOR をとります。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" にライトします。
 Words: 1
 Cycles: 1
 Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process data	Write to destination

Example `XORWF REG 1`
 命令実行前
 REG = 0xAF
 W = 0xB5
 命令実行後
 REG = 0x1A
 W = 0xB5

10.0 開発サポート

10.1 開発ツール

PICmicro™ マイクロコントローラは、以下のハードウェアおよびソフトウェアの開発ツールによりサポートされています。

- PICMASTERR/PICMASTER CE リアルタイムインサーキットエミュレータ
- ICEPIC™ PIC16C5X・PIC16CXXX 用低価格インサーキットエミュレータ
- PRO MATE_ IIユニバーサルプログラマ
- PICSTART_ Plus入門者向け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
- PICDEM-3低価格デモンストレーションボード
- MPASM アセンブラ
- MPLAB™ SIMソフトウェアシミュレータ
- MPLAB-C17(C コンパイラ)
- ファジーロジック開発システム (fuzzyTECH_-MP)

10.2 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、PIC14C000、PIC12CXXX、PIC16C5X、PIC16CXXX および PIC17CXX ファミリーのマイクロコントローラをサポートします。PICMASTER は MPLAB™ 統合開発環境 (IDE) 上でサポートされるので、編集、アセンブル/コンパイルとダウンロード、ソースデバッグが同一環境上で可能となります。

ターゲットプロンプが交換可能なので、エミュレーションするプロセッサが変更となったときでも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 Microchip 社のマイクロコントローラの新製品をサポートすることができるようにつくられています。

PICMASTER エミュレータシステムは、一般的にはもっと高価な開発ツールに装備される最新の機能を備えたリアルタイムのエミュレーションシステムです。PC と互換性のある 386 (以上の) DOS/V マシンプラットフォームおよび Microsoft 社の Windows 3.1/95 で動作します。

PICMASTER には CE 準拠バージョンがあり、EU 諸国でも使用できます。

* PICMASTER にかわる新しい PICmicro 用エミュレータ MPLAB-ICE がリリースされます。

10.3 ICEPIC: 低価格 PICmicro™ - インサーキットエミュレータ*

ICEPIC は PIC12CXXX、PIC16C5X および PIC16CXXX ファミリーの 8 ビット OTP マイクロコントローラ用の低価格インサーキットエミュレータです。

ICEPIC は、286-AT_ マシンから Pentium™ 搭載のマシンまで Windows 3.x 環境の PC に互換性のあるマシンで操作できるよう設計されています。

* ICEPIC にかわる新しい PICmicro 用エミュレータ MPLAB-ICE がリリースされます。

10.4 PRO MATE II: ユニバーサルプログラマ

PRO MATE II 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するプログラムライターです。

PRO MATE II は CE に準拠しています。PRO MATE II にはプログラム可能な V_{DD} と V_{PP} の電源があり、信頼性を最大にするために、 $V_{DD\ min}$ と $V_{DD\ max}$ で、プログラムされたメモリをベリファイすることができます。また、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が装備され、いろいろなパッケージタイプをサポートするためにソケットモジュールが交換可能になっています。スタンドアロンモードでは、PRO MATE II は、PIC12CXXX、PIC14C000、PIC16C5X、PIC16CXXX および PIC17CXX 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、このモードでコードプロテクトのビットを設定できます。

10.5 PICSTART Plus 入門者向け開発システム

PICSTART プログラムは使いやすく低価格のプロトタイププログラムライターで、COM (RS-232) ポート経由で PC に接続します。MPLAB 統合開発環境ソフトウェアにより、簡単かつ効率的に使用できます。PICSTART Plus は量産時のプログラム書き込みには推奨していません。

PICSTART Plus は 40 ピン以下の PIC12CXXX、PIC14C000、PIC16C5X、PIC16CXXX、および PIC17CXX のすべてのデバイスをサポートします (DIPパッケージのみ)。PIC16C923、PIC16C924 および PIC17C756 などのピン数の多いデバイスには、アダプタソケットで対応できます (LCC パッケージのみ)。PICSTART Plus は CE に準拠しています。

PIC16F8X

10.6 PICDEM-1 低価格 PICmicro 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いくつかの機能をデモする簡単なボードです。サポートしているマイクロコントローラは、PIC16C5X (PIC16C54、PIC16C58A)、PIC16C61、PIC16C62X、PIC16C71、PIC16C8X、PIC17C42、PIC17C43 および PIC17C44 です。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が付属しています。PRO MATE II または PICSTART Plus プログラムを使用して、PICDEM-1 付属の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。また、PICDEM-1 ボードを PICMASTER エミュレータに接続して、ファームウェアをダウンロードしてテストすることもできます。さらに、プロトタイプエリアがあるので、いくつかの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ことができます。RS-232 インターフェース、アナログ入力をテストするためのポテンシオメータ、プッシュボタンスイッチ、PORTB に接続された 8 個の LED など実装されています。

10.7 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 および PIC16C74 のマイクロコントローラに対応する簡単なデモンストレーションボードです。基本的なデモンストレーション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が付属しています。PRO MATE II プログラムまたは PICSTART-Plus を使用して PICDEM-2 付属の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。また、PICMASTER エミュレータを PICDEM-2 ボードに接続してファームウェアをテストすることができます。さらに、プロトタイプエリアがあるので、ハードウェアを追加して、マイクロコントローラソケットに接続することができます。RS-232 インターフェース、プッシュボタンスイッチ、アナログ入力をテストするためのポテンシオメータ、12C バスをテストするためのシリアル EEPROM、LCD モジュールとキーパッドを別々に接続するためのヘッダが実装されています。

10.8 PICDEM-3 低価格 PIC16CXXX デモンストレーションボード

PICDEM-3 は、PLCC パッケージの PIC16C923 と PIC16C924 に対応する簡単なデモボードです。また、将来の LCD モジュール付き 44 ピン PLCC パッケージのマイクロコントローラにも対応する予定です。基本的なデモンストレーション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付属しています。PRO MATE II プログラムまたはアダプタソケットと PICSTART Plus を使用して、PICDEM-3 付属の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することができます。また、PICMASTER エミュレータを PICDEM-3 ボードと接続してファームウェアをテストすることができます。さらに、プロトタイプエリアがあるので、ハ-

ドウェアを追加して、マイクロコントローラソケットに接続することができます。RS-232 インターフェース、プッシュボタンスイッチ、アナログ入力をテストするためのポテンシオメータ、サーミスタ、外部 LCD モジュールとキーパッドを分けて接続するためのヘッダが実装されています。また、PICDEM-3 ボード上には 4 コモン、12 セグメントの LCD パネルが備えられ、時間や温度や曜日表示することができます。PICDEM-3 には追加の RS-232 インターフェースと、Windows 3.1 のソフトウェアがあり、PC でデマルチプレクスされた LCD 信号を見ることができます。簡単なシリアルインターフェースにより、LCD 信号のハードウェアでのデマルチプレクスを実行させることができます。

10.9 MPLAB™ 統合開発環境ソフトウェア

MPLAB IDE ソフトウェアは、8 ビットマイクロコントローラ市場では今まで見られなかった使いやすい開発ソフトウェアです。MPLAB は、下記を含む Windows のアプリケーションです。

- フル機能エディタ
- 3 つの動作モード
 - エディタ
 - エミュレータ
 - シミュレータ
- プロジェクトマネージャ
- カスタマイズ可能なツールバーおよびキーマッピング
- プロジェクト情報を表示するステータスバー
- オンラインヘルプ

MPLAB は次のようなことができます。

- ソースファイルの編集 (アセンブラまたは 'C')
- ワンタッチのアセンブル (またはコンパイル) および PICmicro ツールへのダウンロード (全プロジェクト情報を自動的に更新する)
- 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 (今後、OLE に置換予定)
- 同一 PC 上で、4 ユニットまでのエミュレータが動作

Microchip のシミュレータが MPLAB で使用できるので、同じプラットフォーム上で、かつ、同様の操作方法で、低コストのシミュレータからフル機能を備えたエミュレータに簡単に移行することができます。

10.10 アセンブラ (MPASM)

MPASM ユニバーサルマクロアセンブラは、PC ホストのシンボリックアセンブラです。これは、PIC12C5XX、PIC14000、PIC16C5X、PIC16CXXX および PIC17CXX ファミリーを含む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、フル機能を備えたマクロ機能、条件付きアセンブリ、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ム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) による完全なシンボリックデバッグが可能です。

MPASM は、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、次のような機能を有しています。

- ・ アセンブラのソースコードをすべての PICmicro のマイクロコントローラのオブジェクトコードへ変換する。
- ・ マクロアセンブリ可能。
- ・ Microchip エミュレータシステムのシンボリックデバッグに必要なすべてのファイル (オブジェクト、リスティング、シンボル、特殊) を生成する。
- ・ 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する。

MPASM には PICmicro のプログラミングをサポートする高級な疑似命令言語があります。疑似命令により、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。

10.11 ソフトウェアシミュレータ (MPLAB-SIM)

MPLAB-SIM ソフトウェアシミュレータにより、PC ホスト環境でのコード開発が可能になります。PICmicro シリーズのマイクロコントローラを命令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、データ領域のリードや変更、またそのピンのどれにも疑似外部入力ができます。実行はシングルステップ、ブレーク、トレースモードなどで行えます。

MPLAB-SIM は、MPASM を使用して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、実験室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できる、優れたマルチプロジェクトの開発ツールです。

10.12 C コンパイラ (MPLAB-17)

MPLAB-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 PIC17CXXX ファミリー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 MPLAB IDE メモリ表示と互換性があるシンボル情報を提供します。

10.13 ファジロジック開発システム (fuzzyTECH-MP)

fuzzyTECH-MP ファジロジック開発ツールは、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 Explorer で、ファジ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ロジックシステムを実験するため、fuzzyLAB™ デモボードが付属しています。

10.14 SEEVAL® 評価およびプログラミングシステム

SEEVAL SEEPROM デザイナーズキットは マイクロチップの 2 線および 3 線のすべてのシリアル EEPROM をサポートします。このキットには、Smart Serials™ や secure serials を含む マイクロチップの SEEPROM 製品のリード、ライト、消去、プログラムに必要なすべてが付属しています。Total Endurance™ Disk はトレードオフ分析、および演算の信頼性を補助するために付属しています。トータルキットを使用すると、製品化のための時間を大幅に削減でき、最適化システムを生み出すことができます。

10.15 KEELQQ® 評価およびプログラミングツール

KEELQQ 評価およびプログラミングツールは HCS セキュリティデータ製品をサポートしています。HCS 評価キットには、コードの変化を表示するための LCD ディスプレイ、送信をデコードするためのデコーダ、テストトランスミッタをプログラムするためのプログラミングインターフェースが付属しています。

PIC16F8X

表 10-1: マイクロチップの開発ツール

	PIC12C5XX	PIC14000	PIC16C5X	PIC16CXX	PIC16C6X	PIC16C7XX	PIC16C8X	PIC16C9XX	PIC17C4X	PIC17C75X	24CXX 25CXX 93CXX	HCS200HC S300HCS3 01
Emulator Products												
PICMASTER® / PICMASTER-CE In-Circuit Emulator	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
ICEPIC™ Low-Cost In-Circuit Emulator	✓		✓	✓	✓	✓	✓	✓				
Software Tools												
MPLAB™ Integrated Development Environment	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
MPLAB™ C17 Compiler									✓	✓		
fuzzyTECH®-MP Explorer/Edition Fuzzy Logic Dev. Tool	✓	✓	✓	✓	✓	✓	✓	✓	✓			
MP-DriveWay™ Applications Code Generator			✓	✓	✓	✓	✓	✓	✓			
Total Endurance™ Software Model											✓	
Programmers												
PICSTART®Plus Low-Cost Universal Dev. Kit	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
PRO MATE® II Universal Programmer	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
KEELOQ® Programmer												✓
SEEVAL® Designers Kit											✓	
Demo Boards												
PICDEM-1			✓	✓			✓		✓			
PICDEM-2					✓	✓						
PICDEM-3								✓				
KEELOQ® Evaluation Kit												✓

11.0 電気的特性 --- PIC16F83/PIC16F84

絶対最大定格 †

バイアス下での周囲温度

保存温度

 V_{SS} に対する V_{DD} の電圧 V_{SS} に対する MCLR ピンの電圧 V_{SS} に対する他のすべてのピンの電圧消費電力の合計⁽¹⁾ V_{SS} ピンからの最大電流 V_{DD} ピンへの最大電流入力クランプ電流、 I_{IK} ($V_I < 0$ または $V_I > V_{DD}$)入力クランプ電流、 I_{OK} ($V_O < 0$ または $V_O > V_{DD}$)

I/O ピンごとの最大出力シンク電流

I/O ピンごとの最大出力ソース電流

PORTA の最大出力シンク電流

PORTA の最大出力ソース電流

PORTB の最大出力シンク電流

PORTB の最大出力ソース電流

注意 1: 消費電力は次の式で計算できます。 $P_{DIS} = V_{DD} \times \{(I_{DB} - V_{OH}) \times I_{OH}\} + (V_{OL} \times I_{OL})$ 注意 2: MCLR ピンでのスパイク電圧が V_{SS} 以下、電流が 80 mA 以上になると、ラッチアップが発生する原因となります。したがって、MCLR ピンを Low レベルにするときは、 V_{SS} に直接接続しないで、50 - 100 の直列抵抗を使用してください。

†注意: 上記の「最大定格」を超えるストレスは、デバイスに損傷を与える恐れがあります。これは、ストレスの定格を示すためだけのもの、上記の値、または、この仕様書に記載された値を超える条件で動作することを示すものではありません。継続した期間最大定格で使用した場合、デバイスの信頼性を損なう恐れがあります。

規格値については最新版データシート (英語) を参照してください。

PIC16F8X

PIC16F83/84

表 11-1 オシレータのコンフィグレーションおよび動作周波数のデバイス仕様クロスリファレンス (商業用デバイス)

オシレータ	PIC16F84-04 PIC16F83-04	PIC16F84-10 PIC16F83-10	PIC16LF84-04 PIC16LF83-04
RC	VDD: IDD: IPD: Freq:	VDD: IDD: IPD: Freq:	VDD: IDD: IPD: Freq:
XT	VDD: IDD: IPD: Freq:	VDD: IDD: IPD: Freq:	VDD: IDD: IPD: Freq:
HS	VDD: IDD: IPD: Freq:	VDD: IDD: IPD: Freq:	
LP	VDD: IDD: IPD: Freq:		VDD: IDD: IPD: Freq:
<p>網掛部分は機能はテストしていますが、最小 / 最大仕様のテストされません。必要な仕様が保証されているデバイスタイプを選択することを推奨します。</p>			

規格値については最新版データシート (英語) を参照してください。

11.1 DC 特性: PIC16F84, PIC16F83 (商業用、工業用)

DC 特性 電源ピン		標準動作条件 (特に指定のない場合) 動作温度 0 T _A +70 (商業用) -40 T _A +85 (工業用)					単位	条件
パラメータ 番号	記号	特性	Min	Typ †	Max			
D001 D001A	V _{DD}	電源電圧					XT、RC、LP のオシレータコンフィグレーション HS オシレータコンフィグレーション	
D002	V _{DR}	RAM データ保持電圧 ⁽¹⁾	*				SLEEP モードの場合	
D003	V _{POR}	パワーオンリセットを 確実にするための V _{DD} のスタート電圧					詳細はパワーオンリセットの項をご覧ください。	
D004	S _{VDD}	パワーオンリセットを 確実にするための V _{DD} の立ち上がり率	*				詳細はパワーオンリセットの項をご覧ください。	
D010 D010A D013	I _{DD}	消費電流 ⁽²⁾					RC および XT オシレータコンフィグレーション ⁽⁴⁾ F _{OSC} = 4.0 MHz, V _{DD} = 5.5V F _{OSC} = 4.0 MHz, V _{DD} = 5.5V (フラッシュプログラミング中) HS オシレータコンフィグレーション (PIC16F84-10) F _{OSC} = 10 MHz, V _{DD} = 5.5V	
D020 D021 D021A	I _{PD}	待機電流 ⁽³⁾					V _{DD} = 4.0 V, WDT オン、工業用 V _{DD} = 4.0 V, WDT オフ、商業用 V _{DD} = 4.0 V, WDT オフ、工業用	

* このパラメータは特性データです。テストはしていません。

† "Typ" の列のデータは、特に指定のない限り、5.0V、25°C でテストしています。このパラメータは設計ガイドランスのためのもので、テストはしていません。

- 注意
- 1: これは SLEEP モードで V_{DD} を下げたときに RAM データを失わない最小の値です。
 - 2: この消費電流は、主に動作電圧と周波数によって変化します。その他に I/O ピンの負荷、切り替え率、オシレータの種類、内部コード実行パターン、温度などの要素も消費電流に影響を与えます。
動作中の I_{DD} のテスト条件は次のようになります。
OSC1 = 外部方形波 (レール・トゥ・レール)、I/O ピンは入力の設定で V_{DD} にプルダウン、TOCKI = V_{DD}、MCLR = V_{DD}、WDT オン/オフは明記してある通り。
 - 3: SLEEP モードでの待機電流はオシレータの種類には影響を受けません。待機電流は、SLEEP モードのデバイス、I/O ピンを入力にして、V_{DD} または V_{SS} に接続して測定します。
 - 4: RC オシレータのコンフィグレーションについては、R_{ext} の電流は含みません。R_{ext} の電流は I_R = V_{DD} / 2R_{ext} (mA) という式から概算することができます。(R_{ext} の単位は k)

規格値については最新版データシート (英語) を参照してください。

PIC16F8X

PIC16F83/84

11.2 DC 特性: PIC16LF84、PIC16LF83 (商業用、工業用)

DC 特性 電源ピン		標準動作条件 (特に指定のない場合) 動作温度 0 T _A +70 (商業用) -40 T _A +85 (工業用)					単位	条件
パラメータ 番号	記号	特性	Min	Typ †	Max			
D001	VDD	電源電圧					XT、RC、LP のオシレータコンフィグレーション	
D002	VDR	RAM データ保持電圧 ⁽¹⁾	*				SLEEP モードの場合	
D003	VPOR	パワーオンリセットを 確実にするための V _{DD} のスタート電圧					詳細はパワーオンリセットの項をご覧ください。	
D004	SVDD	パワーオンリセットを 確実にするための V _{DD} の立ち上がり率	*				詳細はパワーオンリセットの項をご覧ください。	
D010 D010A D013	IDD	消費電流 ⁽²⁾					RC および XT オシレータコンフィグレーション ⁽⁴⁾ FOSC = 2.0 MHz, VDD = 5.5V FOSC = 2.0 MHz, VDD = 5.5V (ブラッシュプログラミング中) LP オシレータコンフィグレーション FOSC = 10 MHz, VDD = 5.5V, WDT オフ	
D020 D021 D021A	IPD	待機電流 ⁽³⁾					VDD = 2.0 V, WDT オン、工業用 VDD = 2.0 V, WDT オフ、商業用 VDD = 2.0 V, WDT オフ、工業用	

* このパラメータは特性データです。テストはしていません。

† "Typ" の列のデータは、特に指定のない限り、5.0V、25°C でテストしています。このパラメータは設計ガイドンスのためのもので、テストはしていません。

- 注意
- これは SLEEP モードで V_{DD} を下げたときに RAM データを失わない最小の値です。
 - この消費電流は、主に動作電圧と周波数によって変化します。その他に I/O ピンの負荷、切り替え率、オシレータの種類、内部コード実行パターン、温度などの要素も消費電流に影響を与えます。動作中の I_{DD} のテスト条件は次のようになります。OSC1= 外部方形波 (レール・トゥ・レール)、I/O ピンは入力の設定で V_{DD} にプルダウン、TOCKI = V_{DD}、MCLR = V_{DD}、WDT オン/オフは明記してある通り。
 - SLEEP モードでの待機電流はオシレータの種類には影響を受けません。待機電流は、SLEEP モードのデバイス、I/O ピンを入力にして、V_{DD} または V_{SS} に接続して測定します。
 - RC オシレータのコンフィグレーションについては、Rext の電流は含みません。Rext の電流は I_R = V_{DD} / 2Rext (mA) という式から概算することができます。(Rext の単位は k)

規格値については最新版データシート (英語) を参照してください。

11.3 DC 特性: PIC16F84, PIC16F83 (商業用、工業用)、
PIC16LF84、PIC16LF83 (商業用、工業用)

DC 特性		標準動作条件 (特に指定のない場合)					
電源ピン以外のすべてのピン		動作温度 0 T _A +70 (商業用) -40 T _A +85 (工業用)					
		動作電圧 VDD の範囲は、第 10.1 項と第 10.2 項の DC 仕様で説明されている通りです。					
パラメータ番号	記号	特性	Min	Typ †	Max	単位	条件
D030 D030A D031 D032 D033 D034	VIL	Low レベル入力電圧 I/O ポート TTL バッファ付き シュミットトリガバッファ付き MCLR, RA4/T0CKI OSC1 (XT, HS, LP の各モード) ⁽¹⁾ OSC1 (RC モード)				V	4.5 V ≤ VDD ≤ 5.5V ⁽⁴⁾ 全 VDD 範囲 ⁽⁴⁾ 全 VDD 範囲
D040 D040A D041 D042 D043	VIH	High レベル入力電圧 I/O ポート TTL バッファ シュミットトリガバッファ MCLR, RA4/T0CKI、OSC1 (RC モード) OSC1 (XT, HS, LP の各モード) ⁽¹⁾					4.5 V ≤ VDD ≤ 5.5V ⁽⁴⁾ 全 VDD 範囲 ⁽⁴⁾ 全 VDD 範囲
D050	VHYS	シュミットトリガ入力のヒステリシス TBD					
D070	IPURB	PORTB の弱プルアップ電流	*	*	*		VDD = 5.0V, VPIN = VSS
D060 D061 D063	IIL	入力リーク電流 ^(2,3) I/O ポート MCLR, RA4/T0CKI OSC1					VSS ≤ VPIN ≤ VDD, ハイインピーダンスでのピン VSS ≤ VPIN ≤ VDD VSS ≤ VPIN ≤ VDD, XT, HS, LP の各オシレータコン フィギュレーション
D080 D083	VOL	Low レベル出力電圧 I/O ポート OSC2/CLKOUT					IOL = 8.5 mA, VDD = 4.5V IOL = 1.6 mA, VDD = 4.5V
D090 D092	VOH	High レベル出力電圧 I/O ポート ⁽³⁾ High レベル出力電圧					IOH = -3.0 mA, VDD = 4.5V IOH = -1.3 mA, VDD = 4.5V

- * このパラメータは特性データです。テストはしていません。
- † "Typ" の列のデータは、特に指定のない限り、5.0V、25° でテストしています。このパラメータは設計ガイドランスのためのもので、テストはしていません。
- 注意 1: RC オシレータコンフィギュレーションでは、OSC1 ピンはシュミットトリガ入力です。デバイスが RC モードのときは、PIC16F8X を外部クロックとともに使用しないようにしてください。チップに損傷を与える可能性があります。
- 2: MCLR ピンのリーク電流は、供給電圧レベルにより大きく変化します。指定したレベルは、通常の動作状態に相当します。異なる供給電圧レベルではリーク電流が高くなる場合があります。
- 3: 負の電流は、ピンからの出力と定義されています。
- 4: 2 つの仕様のうち良いほうを使用することができます。

規格値については最新版データシート (英語) を参照してください。

PIC16F8X

PIC16F83/84

11.4 DC 特性 PIC16F84、PIC16F83 (商業用、工業用)、PIC16LF84、PIC16F83 (商業用、工業用)

DC 特性 電源ピン以外のすべてのピン		標準動作条件 (特に指定のない場合) 動作温度 0 T_A +70 (商業用) -40 T_A +85 (工業用) 動作電圧 V_{DD} の範囲は、第 10.1 項と第 10.2 項の DC 仕様で説明されている通りです。					
パラメータ番号	記号	特性	Min	Typ †	Max	単位	条件
D100	Cosc2	出力ピンでの負荷容量仕様 OSC2 ピン					XT、HS、LP の各モードでは、外部クロックを使用して OSC1 を駆動します。
D101	Cio	すべての I/O ピンと OSC2 (RC モード)					
D120	ED	データ EEPROM メモリ 耐久性					25°C at 5V V_{MIN} = 中レベルの動作電圧
D121	VDRW	リード/ライト用 V_{DD}					
D122	TDEW	消去/ライトサイクル時間			*		
D130	EP	プログラムフラッシュメモリ 耐久性					V_{MIN} = 中レベルの動作電圧
D131	VPR	リード用 V_{DD}					
D132	VPEW	消去/ライト用 V_{DD}					
D133	TPEW	消去/ライトサイクル時間					

* このパラメータは特性データです。テストはしていません。

† "Typ" の列のデータは、特に指定のない限り、5.0V、25°C でテストしています。このパラメータは設計ガイドランスのためのもので、テストはしていません。

規格値については最新版データシート (英語) を参照してください。

表 11-2 タイミングパラメータの記号

タイミングパラメータの記号は、次のフォーマットで次のように作られます。

1. TppS2ppS
2. TppS

T			
F	周波数	T	時間
小文字の記号 (pp) と意味			
pp		os,osc	OSC1
2	to	ost	オシレータスタートアップタイム
ck	CLKOUT	pwrt	パワーアップタイム
cy	サイクル時間	rbt	RBx ピン
io	I/O ポート	t0	T0CKI
inp	INT ピン	wdt	ウォッチドッグタイム
mc	MCLR		
大文字の記号と意味			
S		P	周期
F	立ち下がり	R	立ち上がり
H	High	V	有効
I	無効 (ハイインピーダンス)	Z	ハイインピーダンス
L	Low		

図 11-1: パラメータの測定情報

すべてのタイミングは、次の図に示すように、High と Low の間で測定されます。

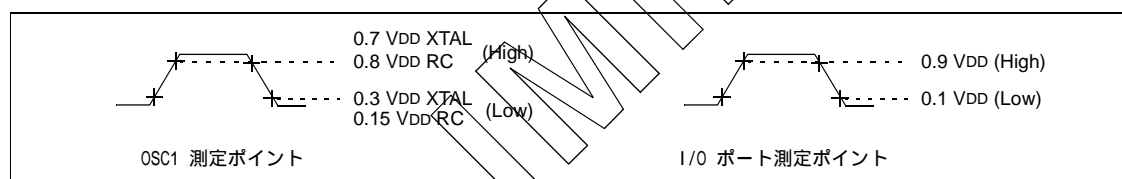


図 11-2: 負荷条件



規格値については最新版データシート (英語) を参照してください。

11.5 タイミング図および仕様

図 11-3: 外部クロックタイミング

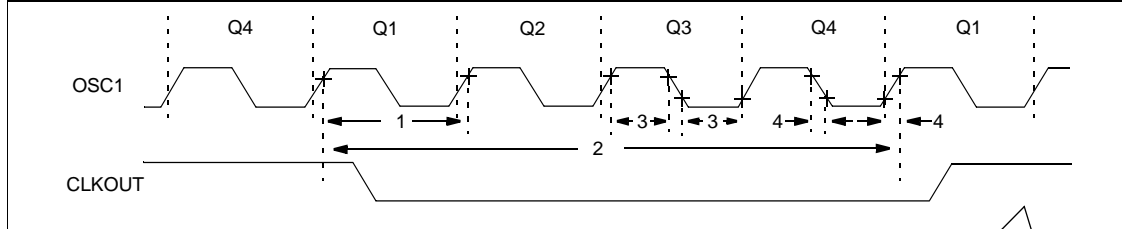


表 11-3 外部クロックタイミング条件

パラメータ番号	記号	特性	Min	Typ †	Max	単位	条件
	Fosc	外部 CLKIN 周波数 ⁽¹⁾				MHz	XT, RC osc PIC16LF8X-04 XT, RC osc PIC16F8X-04
		オシレータ周波数 ⁽¹⁾				MHz	HS osc PIC16F8X-10 LP osc PIC16LF8X-04
1	Tosc	外部 CLKIN 周期 ⁽¹⁾				ns	XT, RC osc PIC16LF8X-04 XT, RC osc PIC16F8X-04
		オシレータ周期 ⁽¹⁾				ns	HS osc PIC16F8X-10 LP osc PIC16LF8X-04
2	TCY	命令サイクル時間 ⁽¹⁾				μs	
3	TosL, TosH	クロック・イン (OSC1) High または Low 時間	*			ns	XT osc PIC16LF8X-04
			*			ns	XT osc PIC16F8X-04
			*			μs	LP osc PIC16LF8X-04
			*			ns	HS osc PIC16F8X-10
4	TosR, TosF	クロック・イン (OSC1) 立ち上がりまたは立ち上がり	*			ns	XT osc PIC16F8X-04
			*			ns	LP osc PIC16LF8X-04
			*			ns	HS osc PIC16F8X-10

* このパラメータは特性データです。テストはしていません。

† "Typ" の列のデータは、特に指定のない限り、5.0V、25°C でテストしていません。このパラメータは設計ガイダンスのためのものです。テストはしていません。

注意 1: 命令サイクル周期 (TCY) は入力オシレータ周期の 4 倍になります。規定されたすべての値は、コードを実行しているデバイスを使用して、標準条件下で、オシレータの種類ごとの特性データに基づき決定しました。規定範囲を超えた場合は、オシレータの動作が不安定になったり、予想消費電流を上回ることがあります。すべてのデバイスは OSC1 ピンに入力される外部クロックの「最小」値で動作をテストします。外部クロック入力を使用するときは、「最大」サイクル時間はすべてのデバイスで "DC" (クロックなし) になります。

規格値については最新版データシート (英語) を参照してください。

図 11-4: CLKOUT および I/O のタイミング

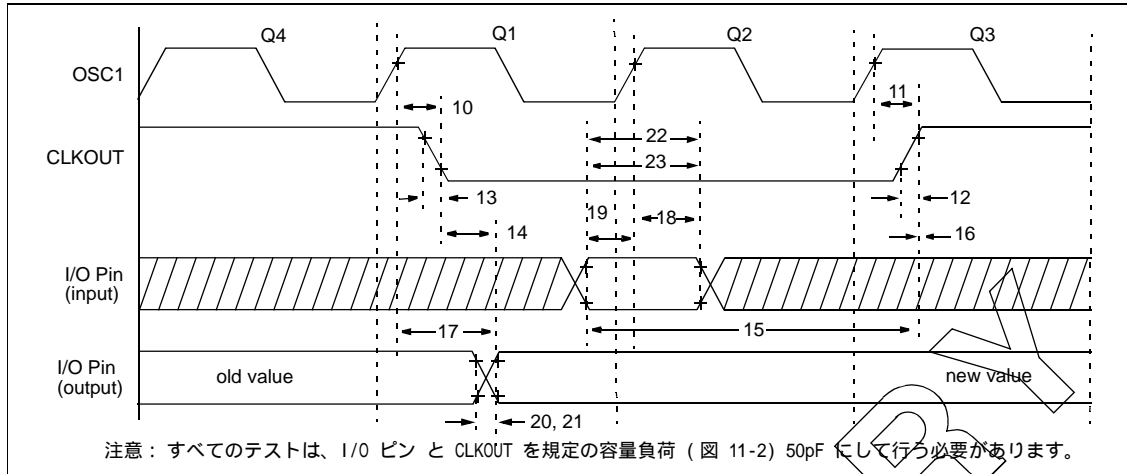


表 11-4: CLKOUT および I/O のタイミング条件

パラメータ番号	特性	Characteristic	Min	Typ†	Max	単位	条件
10	TosH2ckL	OSC1↑ to CLKOUT↓			*	ns	注意 1
10A					*	ns	注意 1
11	TosH2ckH	OSC1↑ to CLKOUT↑			*	ns	注意 1
11A					*	ns	注意 1
12	TckR	CLKOUT 立ち上がり時間			*	ns	注意 1
12A					*	ns	注意 1
13	TckF	CLKOUT 立ち下がり時間			*	ns	注意 1
13A					*	ns	注意 1
14	TckL2ioV	CLKOUT ポート出力有効			*	ns	注意 1
15	TioV2ckH	CLKOUT の前にポート入力有効	*			ns	注意 1
16	TckH2ioI	CLKOUT 後、ポート入力保持	*			ns	注意 1
17	TosH2ioV	OSC1 (Q1 サイクル) ポート出力有効			*	ns	
18	TosH2ioI	OSC1 (Q2 サイクル) ポート入力無効 (I/O のホールド時間)	*			ns	
19	TioV2osH	OSC1 (Q2 サイクル) ポート入力有効 (I/O のセットアップ時間)	*			ns	
20	TioR	ポート出力立ち上がり時間			*	ns	
20A					*	ns	
21	TioF	ポート出力立ち下がり時間			*	ns	
21A					*	ns	
22	Tinp	INT ピン、High または Low 時間	*			ns	
22A			*			ns	
23	Trbp	RB7:RB4 変化割り込み High または Low 時間				ns	
23A						ns	

* このパラメータは特性データです。テストはしていません。

† "Typ" の列のデータは、特に指定のない限り、5.0V、25°C でテストしています。このパラメータは設計ガイダンスのためのもので、テストはしていません。

設計による
注意 1: CLKOUT 出力が 4 x Tosc、RC モードでの計測値です。

規格値については最新版データシート (英語) を参照してください。

図 11-5: リセット、ウォッチドッグタイマ、デバイスリセットタイマのタイミング

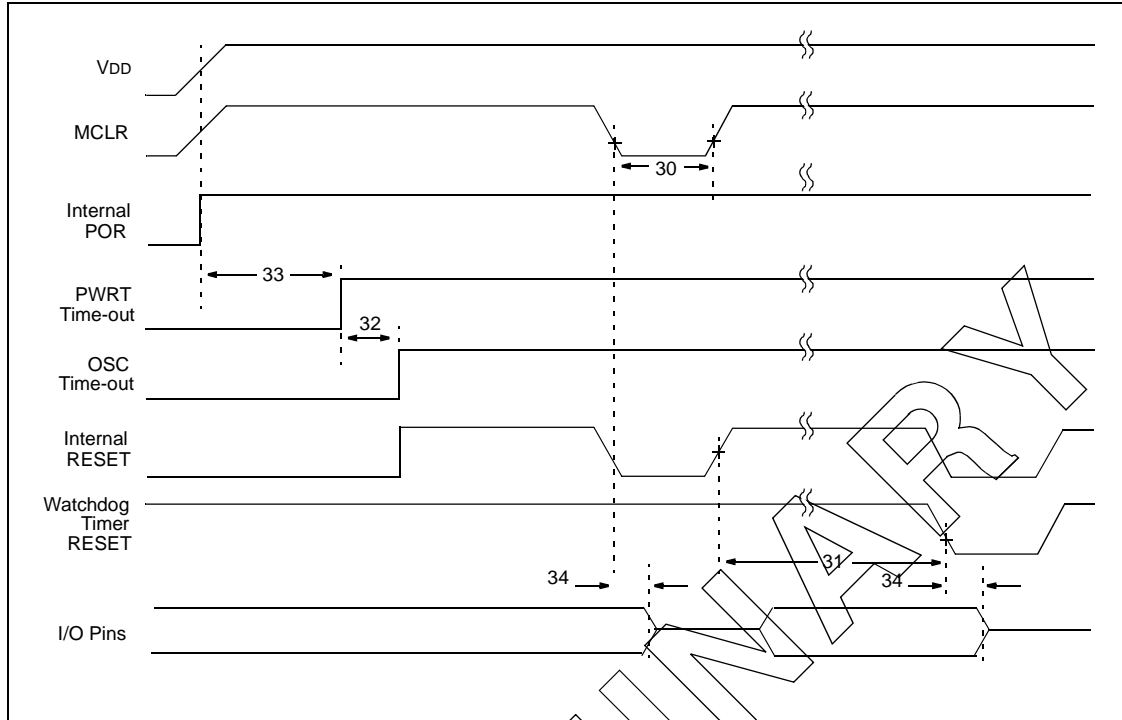


表 11-5 リセット、ウォッチドッグタイマ、オシレータスタートアップタイムおよびパワーアップタイムの条件

パラメータ番号	特性	Characteristic	Min	Typ †	Max	単位	条件
30	TmcL	MCLR パルス幅 (Low)	*			ns	$2.0V \leq V_{DD} \leq 6.0V$
31	Twdt	ウォッチドッグタイマ・タイムアウト周期 (プリスケアラなし)	*		*	ms	$V_{DD} = 5.0V$
32	Tost	オシレーション・スタートアップタイム周期				ms	$T_{OSC} = OSC1 \text{ period}$
33	Tpwrt	パワーアップタイム周期	*		*	ms	$V_{DD} = 5.0V$
34	Tioz	MCLR Low またはリセットからの I/O ハイインピーダンス			*	ns	

* このパラメータは特性データです。テストはしていません。

† "Typ" の列のデータは、特に指定のない限り、5.0V、25°C でテストしています。このパラメータは設計ガイダンスのためのもので、テストはしていません。

規格値については最新版データシート (英語) を参照してください。

図 11-6: タイマ 0 クロックのタイミング

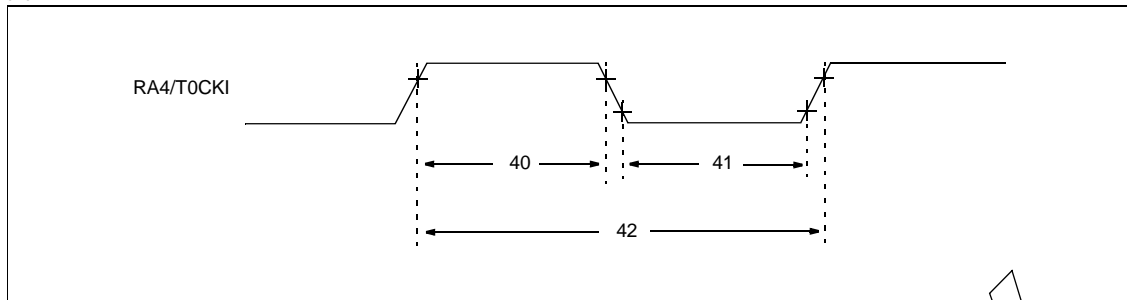


表 11-6 タイマ 0 クロックの条件

パラメータ番号	特性	Characteristic	Min	Typ †	Max	単位	条件
40	Tt0H	T0CKI High パルス幅	プリスケアラなし	*		ns	
			プリスケアラあり	*		ns	$2.0V \leq V_{DD} \leq 3.0V$ $3.0V \leq V_{DD} \leq 6.0V$
41	Tt0L	T0CKI Low パルス幅	プリスケアラなし	*		ns	
			プリスケアラあり	*		ns	$2.0V \leq V_{DD} \leq 3.0V$ $3.0V \leq V_{DD} \leq 6.0V$
42	Tt0P	T0CKI 周期	*			ns	N = プリスケアラ値 (2, 4, ..., 256)

* このパラメータは特性データです。テストはしていません。

† "Typ" の列のデータは、特に指定のない限り、5.0V、25°C でテストしています。このパラメータは設計ガイダンスのためのもので、テストはしていません。

PRELIMINARY

規格値については最新版データシート (英語) を参照してください。

NOTES:

PRELIMINARY

12.0 電気的特性 --- PIC16CR83/PIC16CR84

絶対最大定格 †

バイアス下での周囲温度
 保存温度
 V_{SS} に対する V_{DD} の電圧
 V_{SS} に対する MCLR ピンの電圧 (2)
 V_{SS} に対する他のすべてのピンの電圧
 消費電力の合計 (1)
 V_{SS} ピンからの最大電流
 V_{DD} ピンへの最大電流
 入力クランプ電流、I_{IK} (V_I < 0 または V_I > V_{DD})
 入力クランプ電流、I_{OK} (V_O < 0 または V_O > V_{DD})
 I/O ピンごとの最大出力シンク電流
 I/O ピンごとの最大出力ソース電流
 PORTA の最大出力シンク電流
 PORTA の最大出力ソース電流
 PORTB の最大出力シンク電流
 PORTB の最大出力ソース電流

注意 1: 消費電力は次の式で計算できます。P_{DIS} = V_{DD} × {(I_{DB} - V_{OH}) × I_{OH}} + (V_{OL} × I_{OL})

注意 2: MCLR ピンでのスパイク電圧が V_{SS} 以下、電流が 80 mA 以上になると、ラッチアップが発生する原因となります。したがって、MCLR ピンを Low レベルにするときは、V_{SS} に直接接続しないで、50 - 100 の直列抵抗を使用してください。

†注意: 上記の「最大定格」を超えるストレスは、デバイスに損傷を与える恐れがあります。これは、ストレスの定格を示すためだけのもので、上記の値、または、この仕様書に記載された値を超える条件で動作することを示すものではありません。継続した期間最大定格で使用了場合、デバイスの信頼性を損なう恐れがあります。

規格値については最新版データシート (英語) を参照してください。

PIC16F8X

PIC16CR83/84

表 12-1 オシレータのコンフィグレーションおよび動作周波数のデバイス仕様クロスリファレンス (商業用デバイス)

オシレータ	PIC16CR84-04 PIC16CR83-04	PIC16CR84-10 PIC16CR83-10	PIC16LCR84-04 PIC16LCR83-04
RC	VDD: IDD: IPD: Freq:	VDD: IDD: IPD: Freq:	VDD: IDD: IPD: Freq:
XT	VDD: IDD: IPD: Freq:	VDD: IDD: IPD: Freq:	VDD: IDD: IPD: Freq:
HS	VDD: IDD: IPD: Freq:	VDD: IDD: IPD: Freq:	VDD: IDD: IPD: Freq:
LP	VDD: IDD: IPD: Freq:	VDD: IDD: IPD: Freq:	VDD: IDD: IPD: Freq:

網掛部分は機能のテストに使用されるオシレータを示しています (最小 / 最大の仕様はテストされません)。必要な仕様が保証されているデバイスタイプを選択することをお勧めします。

規格値については最新版データシート (英語) を参照してください。

12.1 DC 特性: PIC16F84、PIC16F83 (商業用、工業用)

DC 特性 電源ピン		標準動作条件 (特に指定のない場合)					
		動作温度 0 T _A +70 (商業用)					
		-40 T _A +85 (工業用)					
パラメータ 番号	記号	特性	Min	Typ †	Max	単位	条件
D001 D001A	VDD	電源電圧					XT、RC、LP のオシレータコンフィギュレーション HS オシレータコンフィギュレーション
D002	VDR	RAM データ保持電圧 ⁽¹⁾	*				SLEEP モードの場合
D003	VPOR	パワーオンリセットを 確実にするための V _{DD} のスタート電圧					詳細はパワーオンリセットの項をご覧ください。
D004	SVDD	パワーオンリセットを 確実にするための V _{DD} の立ち上がり率	*				詳細はパワーオンリセットの項をご覧ください。
D010 D010A D013	IDD	消費電流 ⁽²⁾					RC および XT オシレータコンフィギュレーション ⁽⁴⁾ FOSC = 4.0 MHz, VDD = 5.5V FOSC = 4.0 MHz, VDD = 5.5V (フラッシュプログラミング中) HS オシレータコンフィギュレーション (PIC16R84-10) FOSC = 10 MHz, VDD = 5.5V
D020 D021 D021A	IPD	待機電流 ⁽³⁾					VDD = 4.0 V、WDT オン、工業用 VDD = 4.0 V、WDT オフ、商業用 VDD = 4.0 V、WDT オフ、工業用

* このパラメータは特性データです。テストはしていません。
 † "Typ" の列のデータは、特に指定のない限り、5.0V、25°C でテストしています。このパラメータは設計ガイドランスのためのもので、テストはしていません。
 注意 1: これは SLEEP モードで V_{DD} を下げたときに RAM データを失わない最小の値です。
 2: この消費電流は、主に動作電圧と周波数によって変化します。その他に I/O ピンの負荷、切り替え率、オシレータの種類、内部コード実行パターン、温度などの要素も消費電流に影響を与えます。動作中の I_{DD} のテスト条件は次のようになります。
 OSC1 = 外部方形波 (レベル・トゥ・レベル)、I/O ピンは入力の設定で V_{DD} にプルダウン、TOCKI = V_{DD}
 MCLR = V_{DD}、WDT オン/オフは明記してある通り。
 3: SLEEP モードでの待機電流はオシレータの種類には影響を受けません。待機電流は、SLEEP モードのデバイスを、I/O ピンを入力にして、V_{DD} または V_{SS} に接続して計測します。
 4: RC オシレータのコンフィギュレーションについては、R_{ext} の電流は含みません。R_{ext} の電流は I_R = V_{DD} / 2R_{ext} (mA) という式から概算することができます。(R_{ext} の単位は k)

規格値については最新版データシート (英語) を参照してください。

PIC16F8X

PIC16CR83/84

12.2 DC 特性: PIC16LCR84、PIC16LCR83 (商業用、工業用)

DC 特性 電源ピン		標準動作条件 (特に指定のない場合) 動作温度 0 T _A +70 (商業用) -40 T _A +85 (工業用)					単位	条件
パラメータ 番号	記号	特性	Min	Typ †	Max			
D001	VDD	電源電圧					XT、RC、LP のオシレータコンフィグレーション	
D002	VDR	RAM データ保持電圧 ⁽¹⁾	*				SLEEP モードの場合	
D003	VPOR	パワーオンリセットを 確実にするための V _{DD} のスタート電圧					詳細はパワーオンリセットの項をご覧ください。	
D004	SVDD	パワーオンリセットを 確実にするための V _{DD} の立ち上がり率	*				詳細はパワーオンリセットの項をご覧ください。	
D010 D010A D013	IDD	消費電流 ⁽²⁾					RC および XT オシレータコンフィグレーション ⁽⁴⁾ Fosc = 2.0 MHz, VDD = 5.5V Fosc = 2.0 MHz, VDD = 5.5V (フラッシュプログラミング中) LP オシレータコンフィグレーション Fosc = 10 MHz, VDD = 5.5V, WDT オフ	
D020 D021 D021A	IPD	待機電流 ⁽³⁾					VDD = 2.0 V, WDT オン、工業用 VDD = 2.0 V, WDT オフ、商業用 VDD = 2.0 V, WDT オフ、工業用	

* このパラメータは特性データです。テストはしていません。

† "Typ" の列のデータは、特に指定のない限り、5.0V、25°C でテストしています。このパラメータは設計ガイドランスのためのもので、テストはしていません。

- 注意
- これは SLEEP モードで V_{DD} を下げたときに RAM データを失わない最小の値です。
 - この消費電流は、主に動作電圧と周波数によって変化します。その他に I/O ピンの負荷、切り替え率、オシレータの種類、内部コード実行パターン、温度などの要素も消費電流に影響を与えます。
動作中の I_{DD} のテスト条件は次のようになります。
OSC1= 外部方形波 (レール・トゥ・レール)、I/O ピンは入力の設定で V_{DD} にプルダウン、TOCK1 = V_{DD}、MCLR = V_{DD}、WDT オン/オフは明記してある通り。
 - SLEEP モードでの待機電流はオシレータの種類には影響を受けません。待機電流は、SLEEP モードのデバイス、I/O ピンを入力にして、V_{DD} または V_{SS} に接続して計測します。
 - RC オシレータのコンフィグレーションについては、Rext の電流は含みません。Rext の電流は $I_R = V_{DD} / 2R_{ext}$ (mA) という式から概算することができます。(Rext の単位は k)

規格値については最新版データシート (英語) を参照してください。

12.3 DC 特性: PIC16CR84、PIC16CR83 (商業用、工業用)、
PIC16LCR84、PIC16LCR83 (商業用、工業用)

パラメータ番号	記号	特性	Min	Typ †	Max	単位	条件
		標準動作条件 (特に指定のない場合) 動作温度 0 T _A +70 (商業用) -40 T _A +85 (工業用) 動作電圧 V _{DD} の範囲は、第 10.1 項と第 10.2 項の DC 仕様で説明されている通りです。					
D030 D030A D031 D032 D033 D034	VIL	Low レベル入力電圧 I/O ポート TTL バッファ付き シュミットトリガバッファ付き MCLR, RA4/T0CKI OSC1 (XT、HS、LP の各モード) ⁽¹⁾ OSC1 (RC モード)				V	4.5 V ≤ V _{DD} ≤ 5.5V ⁽⁴⁾ 全 V _{DD} 範囲 ⁽⁴⁾ 全 V _{DD} 範囲
D040 D040A D041 D042 D043	VIH	High レベル入力電圧 I/O ポート TTL バッファ シュミットトリガバッファ MCLR, RA4/T0CKI、OSC1 (RC モード) OSC1 (XT、HS、LP の各モード) ⁽¹⁾					4.5 V ≤ V _{DD} ≤ 5.5V ⁽⁴⁾ 全 V _{DD} 範囲 ⁽⁴⁾ 全 V _{DD} 範囲
D050	VHYS	シュミットトリガ入力ヒステリシス TBD					
D070	IPURB	PORTB の弱プルアップ電流	*	*	*		V _{DD} = 5.0V, V _{PIN} = V _{SS}
D060 D061 D063	IIL	入力リーク電流 ^(2,3) I/O ポート MCLR, RA4/T0CKI OSC1					V _{SS} ≤ V _{PIN} ≤ V _{DD} , ハイインピーダンスでのピン V _{SS} ≤ V _{PIN} ≤ V _{DD} V _{SS} ≤ V _{PIN} ≤ V _{DD} , XT, HS, LP の各オシレータコン フィギュレーション
D080 D083	VOL	Low レベル出力電圧 I/O ポート OSC2/CLKOUT					IOL = 8.5 mA, V _{DD} = 4.5V IOL = 1.6 mA, V _{DD} = 4.5V
D090 D092	VOH	High レベル出力電圧 I/O ポート ⁽³⁾ High レベル出力電圧					IOH = -3.0 mA, V _{DD} = 4.5V IOH = -1.3 mA, V _{DD} = 4.5V

- * このパラメータは特性データです。テストはしていません。
† "Typ" の列のデータは、特に指定のない限り、5.0V、25° でテストしています。このパラメータは設計ガイダンスのためのもので、テストはしていません。
注意 1: RC オシレータコンフィギュレーションでは、OSC1 ピンはシュミットトリガ入力です。デバイスが RC モードのときは、PIC16F8X を外部クロックとともに使用しないようにしてください。チップに損傷を与える可能性があります。
2: MCLR ピンのリーク電流は、供給電圧レベルにより大きく変化します。指定したレベルは、通常の動作状態に相当します。リーク電流が高い場合は、異なる入力電圧で計測された可能性があります。
3: 負の電流は、ピンからの出力と定義されています。
4: 2つの仕様のうち良いほうを使用することができます。

規格値については最新版データシート (英語) を参照してください。

PIC16F8X

PIC16CR83/84

12.4 DC 特性 : PIC16CR84、PIC16CR83 (商業用、工業用)、PIC16LCR84、PIC16LCR83 (商業用、工業用)

DC 特性 電源ピン以外のすべてのピン		標準動作条件 (特に指定のない場合) 動作温度 0 T_A +70 (商業用) -40 T_A +85 (工業用) 動作電圧 V_{DD} の範囲は、第 10.1 項と第 10.2 項の DC 仕様で説明されている通りです。					
パラメータ 番号	特性	Characteristic	Min	Typ †	Max	単位	条件
D100	Cosc2	出力ピンでの負荷容量仕様 OSC2 ピン					XT、HS、LP の各モードでは、外部クロックを使用して OSC1 を駆動します。
D101	Cio	すべての I/O ピンと OSC2 (RC モード)					
D120	Ed	データ WWPROM メモリ 耐久性					25°C at 5V V_{MIN} = 中レベルの動作電圧
D121	VDRW	リード/ライト用 V_{DD}					
D122	TDEW	消去/ライトサイクル時間			*		

* このパラメータは特性データです。テストはしていません。

† "Typ" の列のデータは、特に指定のない限り、5.0V、25 °C でテストしています。このパラメータは設計ガイドランスのためのもので、テストはしていません。

PRELIMINARY

規格値については最新版データシート (英語) を参照してください。

表 12-2 タイミングパラメータの記号

タイミングパラメータの記号は、次のフォーマットで次のように作られます。

1. TppS2ppS
2. TppS

T	F 周波数	T 時間
小文字の記号 (pp) と意味		
pp	2 to ck CLKOUT cy サイクル時間 io I/O ポート inp INT ピン mc MCLR	os,osc OSC1 ost オシレータスタートアップタイム pwrt パワーアップタイム rbt RBx ピン t0 T0CKI wdt ウォッチドッグタイム
大文字の記号と意味		
S	F 立ち下がり H High I 無効 (ハイインピーダンス) L Low	P 周期 R 立ち上がり V 有効 Z ハイインピーダンス

図 12-1: パラメータの測定情報

すべてのタイミングは、次の図に示すように、ハイとローの間で測定されます。



図 12-2: 負荷条件



規格値については最新版データシート (英語) を参照してください。

12.5 タイミング図および仕様

図 12-3: 外部クロックタイミング

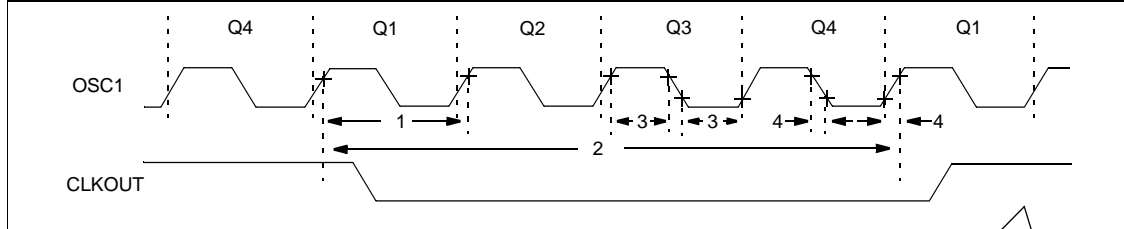


表 12-3 外部クロックタイミング条件

パラメータ番号	記号	特性	Min	Typ †	Max	単位	条件
	Fosc	外部 CLKIN 周波数 ⁽¹⁾				MHz	XT, RC osc PIC16LCR8X-04 MHz XT, RC osc PIC16CR8X-04 MHz HS osc PIC16CR8X-10 kHz LP osc PIC16LCR8X-04
		オシレータ周波数 ⁽¹⁾				MHz	RC osc PIC16LCR8X-04 MHz RC osc PIC16CR8X-04 MHz XT osc PIC16LCR8X-04 MHz XT osc PIC16CR8X-04 MHz HS osc PIC16CR8X-10 kHz LP osc PIC16LCR8X-04
1	Tosc	外部 CLKIN 周期 ⁽¹⁾				ns	XT, RC osc PIC16LCR8X-04 ns XT, RC osc PIC16CR8X-04 ns HS osc PIC16CR8X-10 μs LP osc PIC16LCR8X-04
		オシレータ周期 ⁽¹⁾				ns	RC osc PIC16LCR8X-04 ns RC osc PIC16CR8X-04 ns XT osc PIC16LCR8X-04 ns XT osc PIC16CR8X-04 ns HS osc PIC16CR8X-10 μs LP osc PIC16LCR8X-04
2	Tcy	命令サイクル時間 ⁽¹⁾				μs	
3	TosL, TosH	クロック・イン (OSC1) High または Low 時間	*			ns	XT osc PIC16LCR8X-04
			*			ns	XT osc PIC16CR8X-04
			*			μs	LP osc PIC16LCR8X-04
			*			ns	HS osc PIC16CR8X-10
4	TosR, TosF	クロック・イン (OSC1) 立ち上がりまたは立ち下がり	*			ns	XT osc PIC16CR8X-04
			*			ns	LP osc PIC16LCR8X-04
			*			ns	HS osc PIC16CR8X-10

* このパラメータは特性データです。テストはしていません。

† "Typ" の列のデータは、特に指定のない限り、5.0V、25°C でテストしています。このパラメータは設計ガイダンスのためのもので、テストはしていません。

注意 1: 命令サイクル周期 (Tcy) は入力オシレータ周期の 4 倍になります。規定されたすべての値は、コードを実行しているデバイスを使用して、標準条件下で、オシレータの種類ごとの特性データに基づき決定しました。規定範囲を超えた場合は、オシレータの動作が不安定になったり、予想消費電流を上回ることがあります。すべてのデバイスは OSC1 ピンに入力される外部クロックの「最小」値で動作をテストします。外部クロック入力を使用するときは、「最大」サイクル時間はすべてのデバイスで "DC" (クロックなし) になります。

規格値については最新版データシート (英語) を参照してください。

図 12-4: CLKOUT および I/O のタイミング

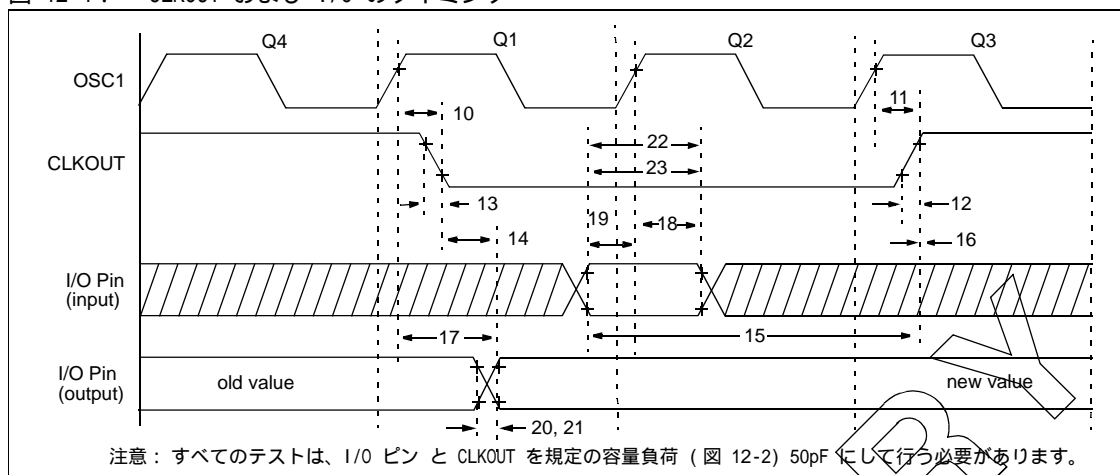


表 12-4: CLKOUT および I/O のタイミング条件

パラメータ番号	記号	特性	Min	Typ†	Max	単位	条件
10	TosH2ckL	OSC1↑ to CLKOUT↓			*	ns	注意 1
10A					*	ns	注意 1
11	TosH2ckH	OSC1↑ to CLKOUT↑			*	ns	注意 1
11A					*	ns	注意 1
12	TckR	CLKOUT 立ち上がり時間			*	ns	注意 1
12A					*	ns	注意 1
13	TckF	CLKOUT 立ち下がり時間			*	ns	注意 1
13A					*	ns	注意 1
14	TckL2ioV	CLKOUT ポート出力有効			*	ns	注意 1
15	TioV2ckH	CLKOUT の前にポート入力有効	*			ns	注意 1
16	TckH2iol	CLKOUT 後、ポート入力保持	*			ns	注意 1
17	TosH2ioV	OSC1 (Q1 サイクル) ポート出力有効			*	ns	
18	TosH2iol	OSC1 (Q2 サイクル) ポート入力無効 (I/O のホールド時間)	*			ns	
19	TioV2osH	OSC1 (Q2 サイクル) ポート入力有効 (I/O のセットアップ時間)	*			ns	
20	TioR	ポート出力立ち上がり時間			*	ns	
20A					*	ns	
21	TioF	ポート出力立ち下がり時間			*	ns	
21A					*	ns	
22	Tinp	INT ピン、High または	*			ns	
22A		Low 時間	*			ns	
23	Trbp	RB7:RB4 変化割り込み				ns	
23A		High または Low 時間				ns	

* このパラメータは特性データです。テストはしていません。

† "Typ" の列のデータは、特に指定のない限り、5.0V、25°C でテストしています。このパラメータは設計ガイダンスのためのもので、テストはしていません。

設計 注意 1: CLKOUT 出力が 4 x Tosc、RC モードでの計測値です。

規格値については最新版データシート (英語) を参照してください。

図 12-5: リセット、ウォッチドッグタイマ、デバイスリセットタイマのタイミング

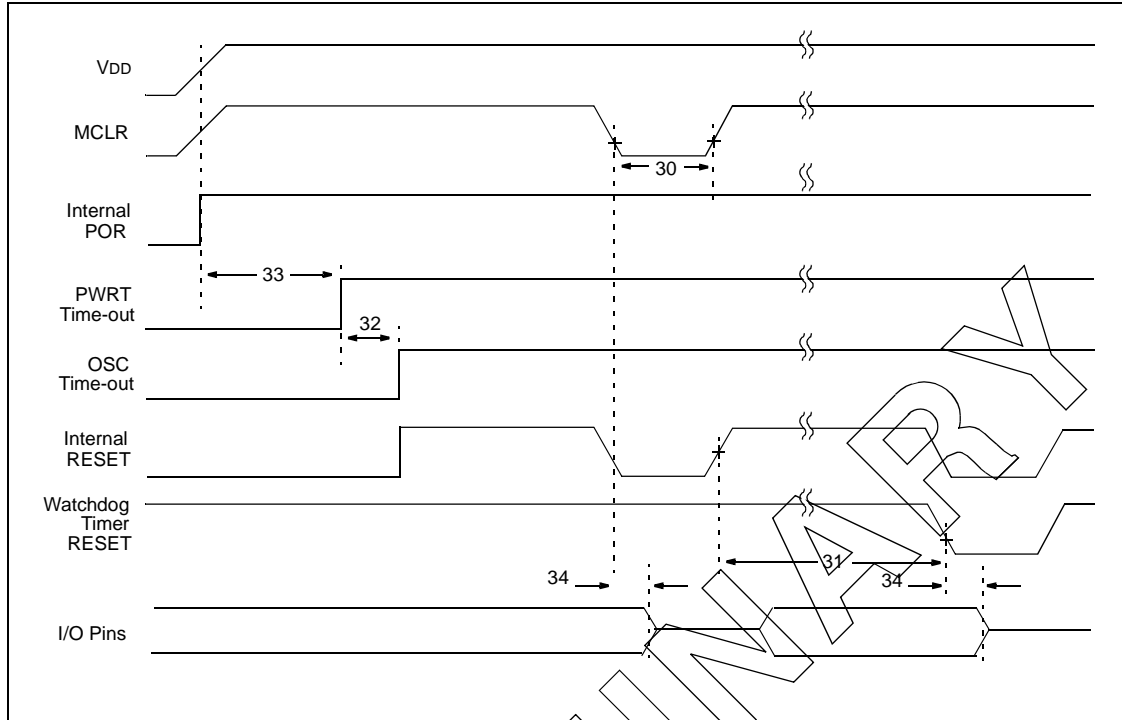


表 12-5 リセット、ウォッチドッグタイマ、オシレータスタートアップタイムおよびパワーアップタイムの条件

パラメータ番号	記号	特性	Min	Typ †	Max	単位	条件
30	TmcL	MCLR パルス幅 (Low)	*			ns	$2.0V \leq V_{DD} \leq 6.0V$
31	Twdt	ウォッチドッグタイマ・タイムアウト周期 (プリスケアラなし)	*		*	ms	$V_{DD} = 5.0V$
32	Tost	オシレーション・スタートアップタイム周期				ms	$T_{OSC} = OSC1 \text{ period}$
33	Tpwrt	パワーアップタイム周期	*		*	ms	$V_{DD} = 5.0V$
34	Tioz	MCLR Low またはリセットからの I/O ハイインピーダンス			*	ns	

* このパラメータは特性データです。テストはしていません。

† "Typ" の列のデータは、特に指定のない限り、5.0V、25 °C でテストしています。このパラメータは設計ガイダンスのためのもので、テストはしていません。

規格値については最新版データシート (英語) を参照してください。

図 12-6: タイマ 0 クロックのタイミング

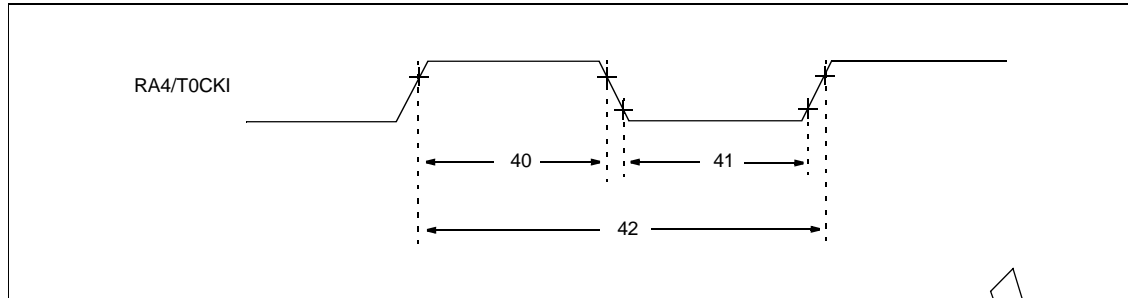


表 12-6 タイマ 0 クロックの条件

パラメータ番号	記号	特性	Min	Typ †	Max	単位	条件
40	Tt0H	T0CKI High パルス幅	プリスケアラなし	*		ns	
			プリスケアラあり	*		ns	$2.0V \leq V_{DD} \leq 3.0V$ $3.0V \leq V_{DD} \leq 6.0V$
41	Tt0L	T0CKI Low パルス幅	プリスケアラなし	*		ns	
			プリスケアラあり	*		ns	$2.0V \leq V_{DD} \leq 3.0V$ $3.0V \leq V_{DD} \leq 6.0V$
42	Tt0P	T0CKI 周期	*			ns	N = プリスケアラ値 (2, 4, ..., 256)

* このパラメータは特性データです。テストはしていません。

† "Typ" の列のデータは、特に指定のない限り、5.0V、25°C でテストしています。このパラメータは設計ガイダンスのためのもので、テストはしていません。

PRELIMINARY

規格値については最新版データシート (英語) を参照してください。

NOTES:

PRELIMINARY

13.0 DC および AC 特性 --- グラフと表

この章に規定したグラフと表は設計ガイダンスを示すためのもので、テストはしていません。また、保証値でもありません。

いくつかのグラフや表では、データが規定の動作範囲から外れているものもあります。(例えば、規定 V_{DD} 範囲外)。これは参考のためで、デバイスは規定の範囲内に限り正常に動作しています。

この章のデータは、一定期間にわたってロットの異なる製品から採集したデータの統計をまとめたものです。"Typical" は 25 °C での平均を表し、"max" や "min" はそれぞれ (平均 + 3 σ)、(平均 - 3 σ) を表します (σ は標準偏差)。

図 13-1 標準的な RC オシレータ周波数 VS 温度

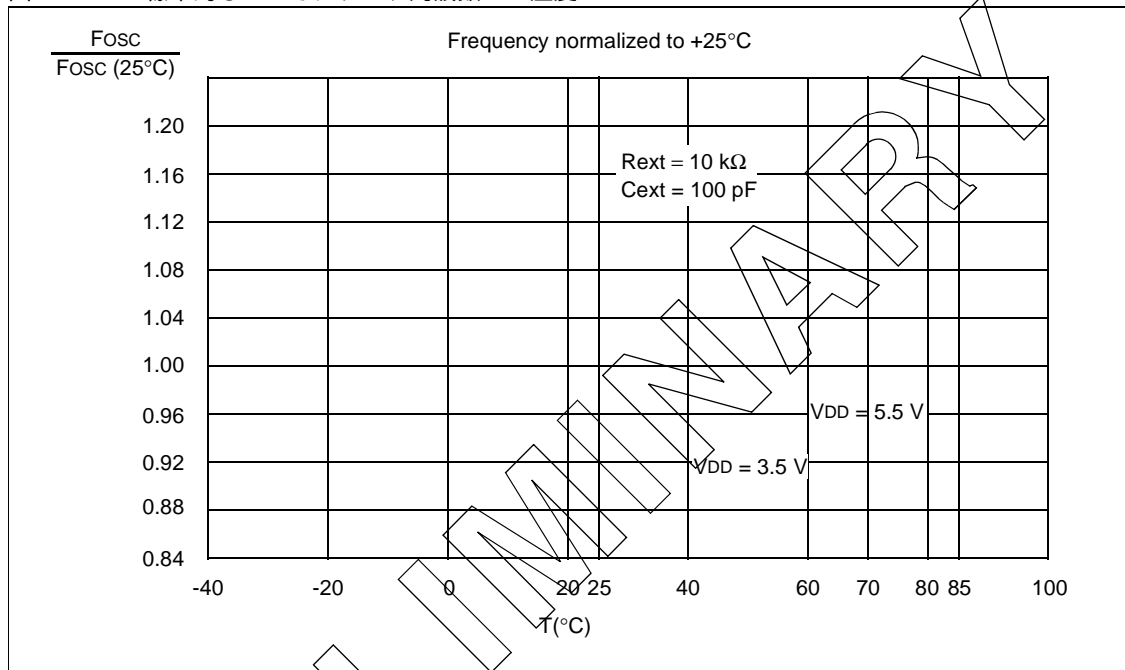


表 13-1 RC オシレータの周波数 *

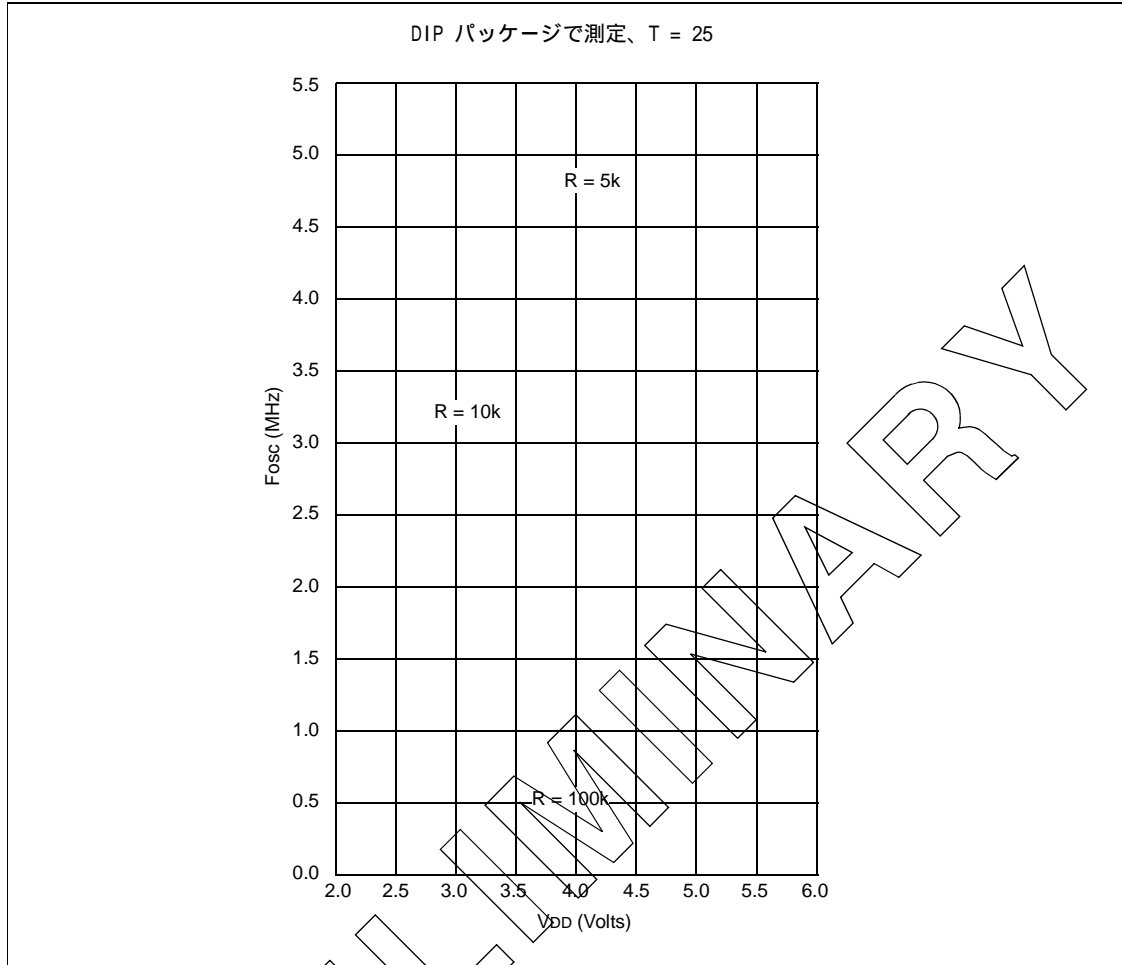
Cext	Rext	平均 Fosc @ 5V、25	
			誤差
20 pF	5 k		
	10 k		
	100 k		
100 pF	5 k		
	10 k		
	100 k		
300 pF	5 k		
	10 k		
	100 k		

* DIP パッケージで測定。この誤差は、通常の製造過程での製品ごとの誤差を示しています。全 V_{DD} 範囲の平均値の標準誤差は ± 3 です。

規格値については最新版データシート (英語) を参照してください。

PIC16F8X

表 13-2: 標準的な RC オシレータ周波数 VS V_{DD} 、 $C_{EXT} = 20 \text{ pF}$



規格値については最新版データシート (英語) を参照してください。

図 13-3: 標準的な RC オシレータ周波数 VS V_{DD} 、 $C_{EXT} = 100 \text{ pF}$

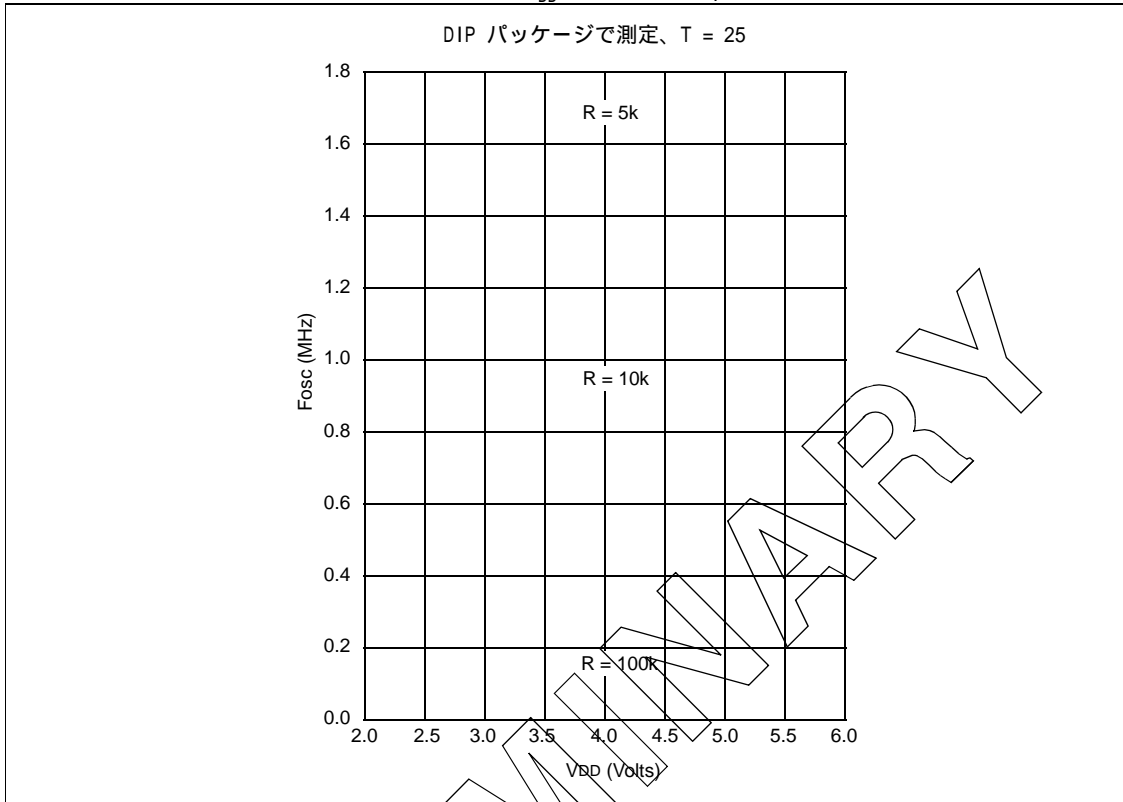
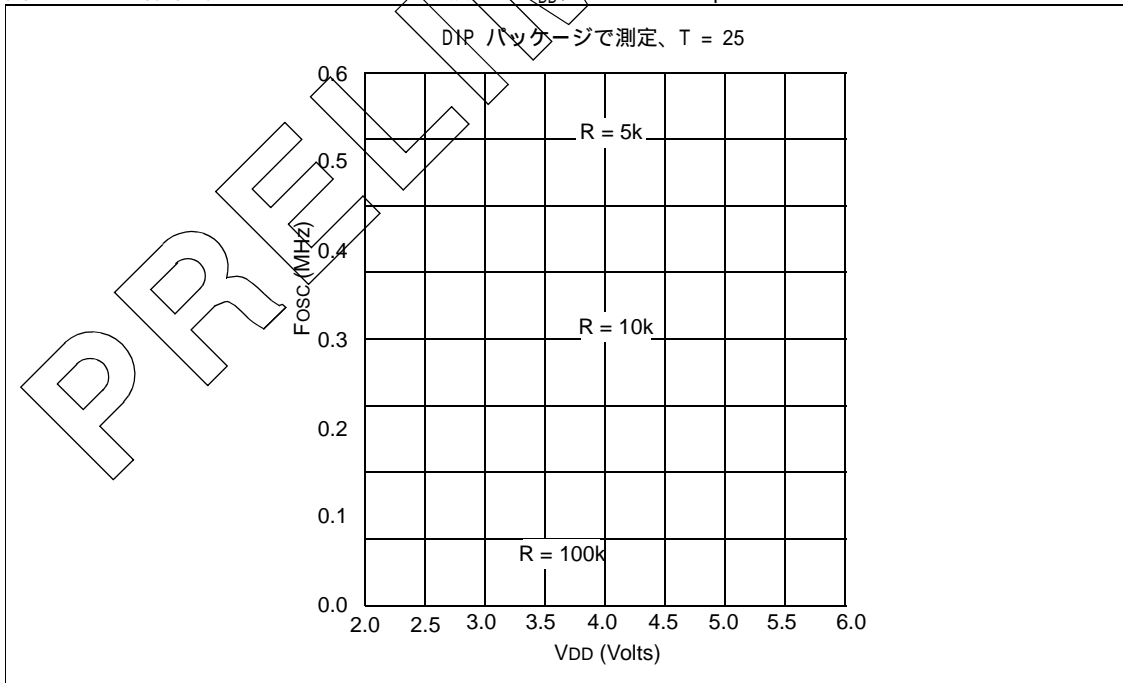


図 13-4: 標準的な RC オシレータ周波数 VS V_{DD} 、 $C_{EXT} = 300 \text{ pF}$



規格値については最新版データシート (英語) を参照してください。

PIC16F8X

図 13-5: 標準的な I_{PD} VS V_{DD} 、WDT オフ

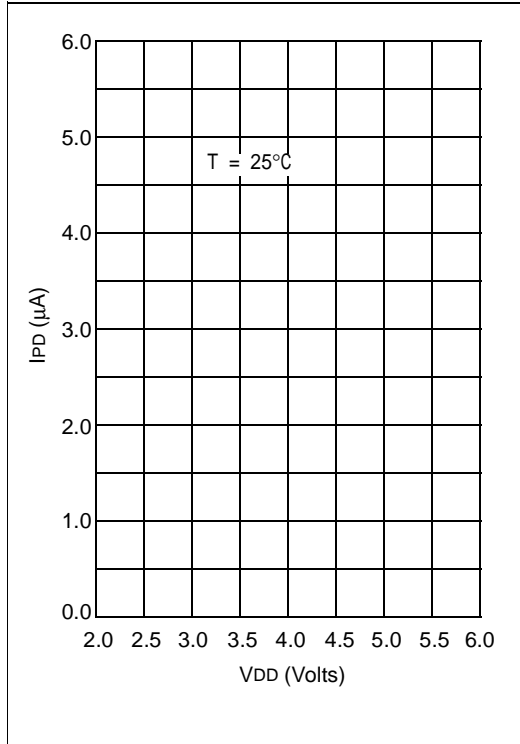


図 13-6: 標準的な I_{PD} VS V_{DD} 、WDT オフ

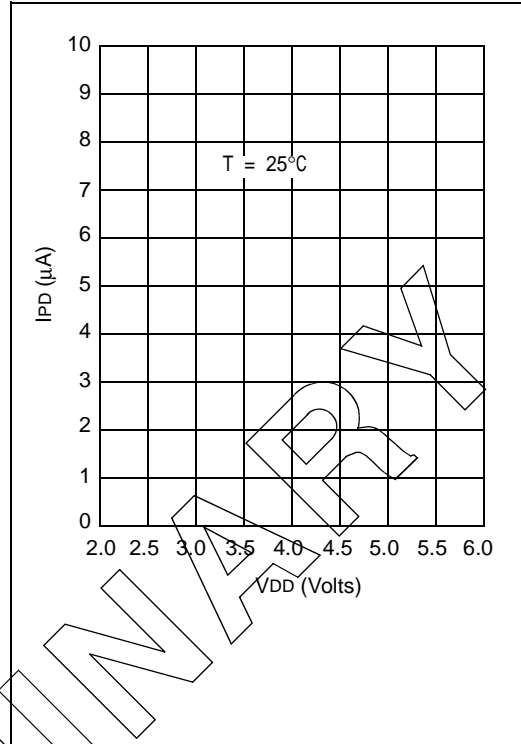
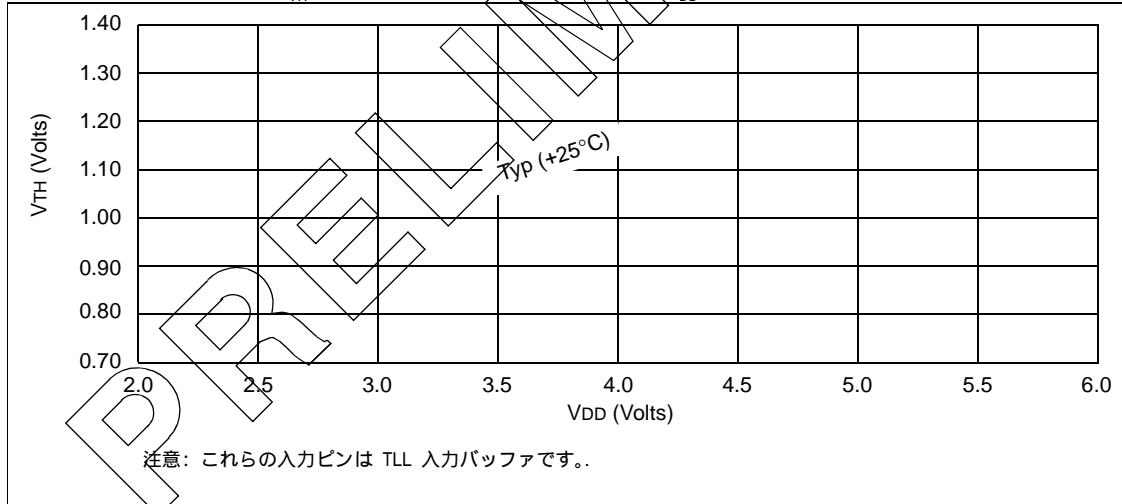


図 13-7: I/O ピンの V_{TH} (入力スレッショルド電圧) VS V_{DD}



規格値については最新版データシート (英語) を参照してください。

図 13-8: OSC1 INPUT (XT、HS、LP の各モード) の V_{TH} (入力スレッショルド電圧) VS V_{DD}

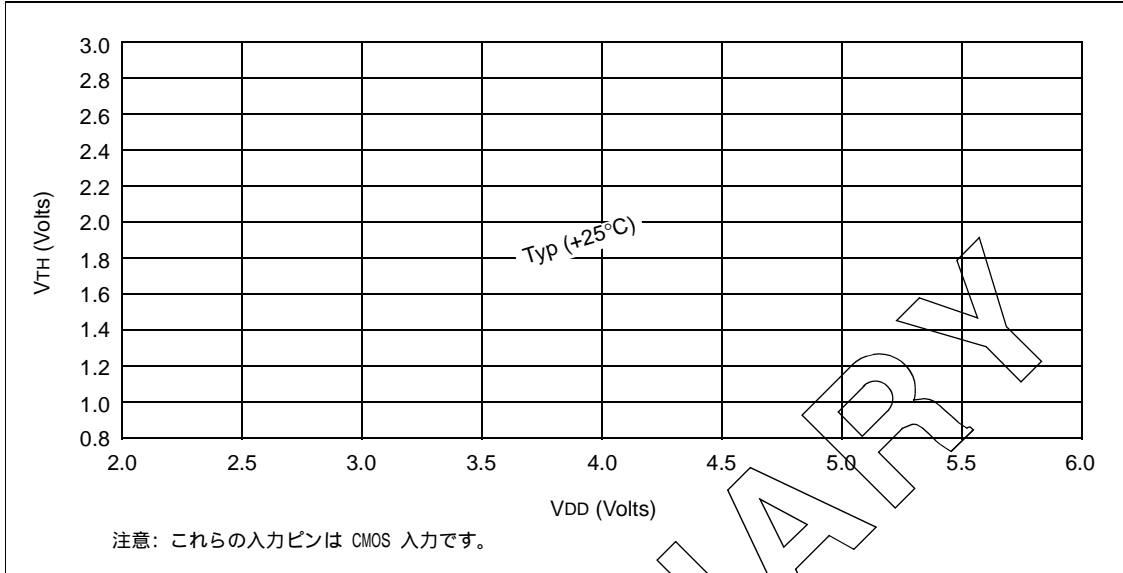
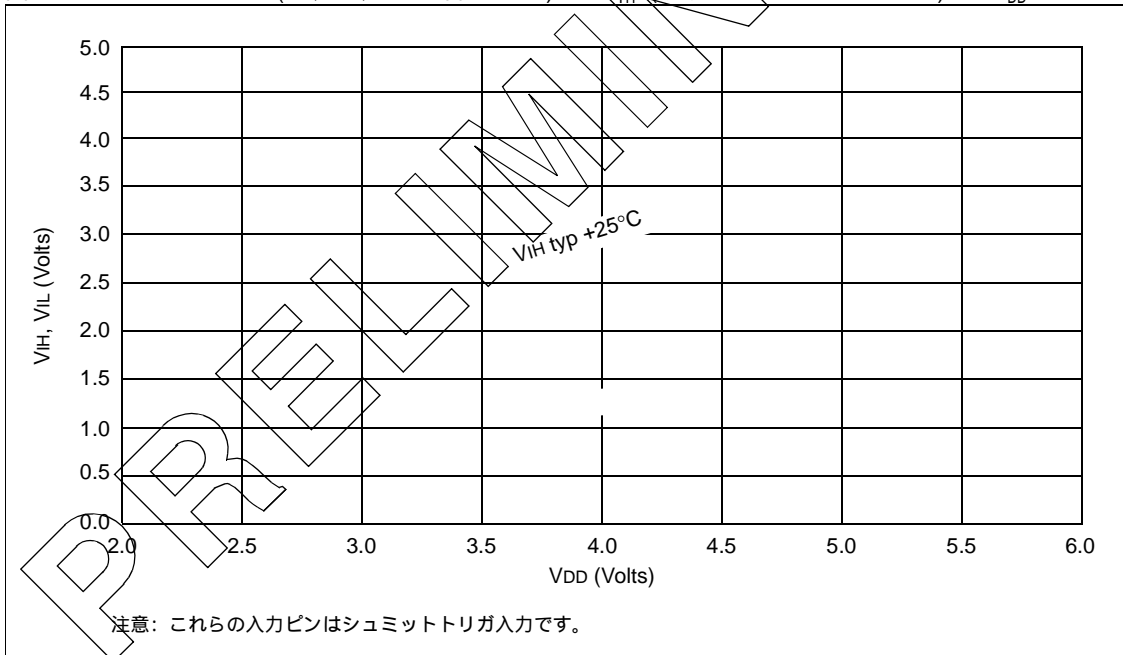


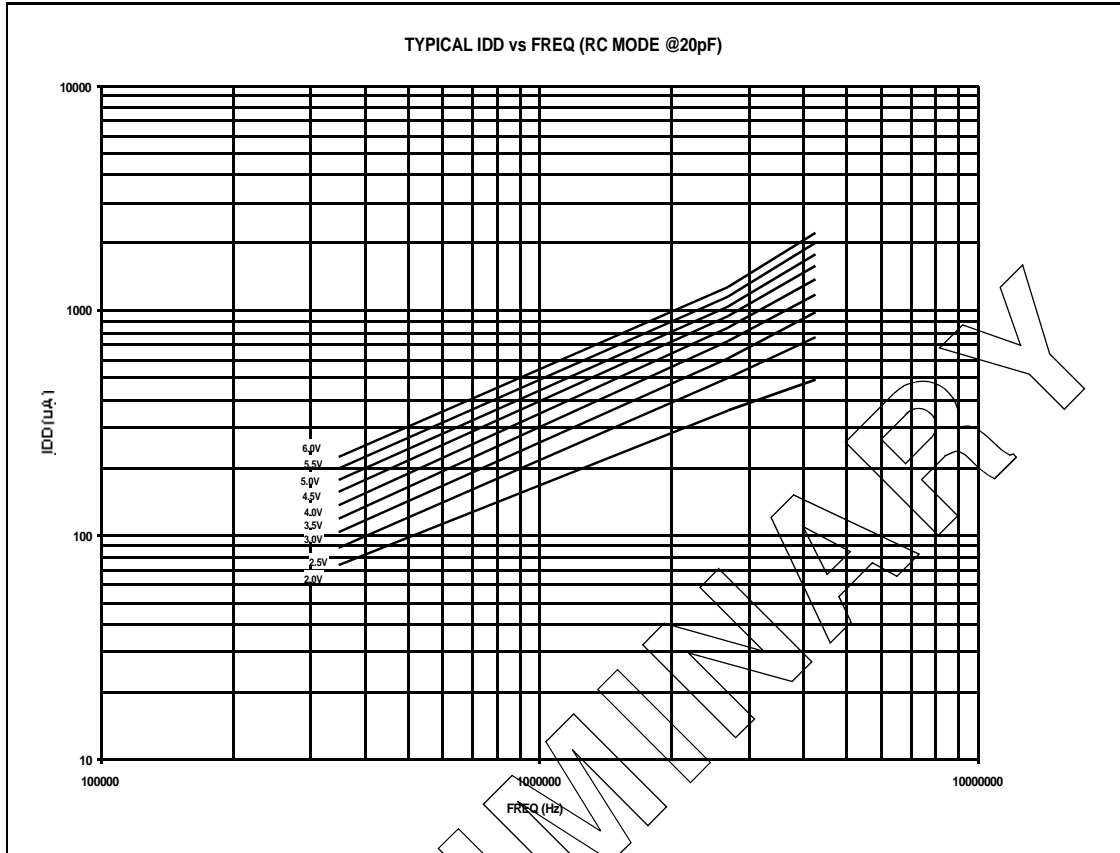
図 13-9: OSC1 INPUT (XT、HS、LP の各モード) の V_{TH} (入力スレッショルド電圧) VS V_{DD}



規格値については最新版データシート (英語) を参照してください。

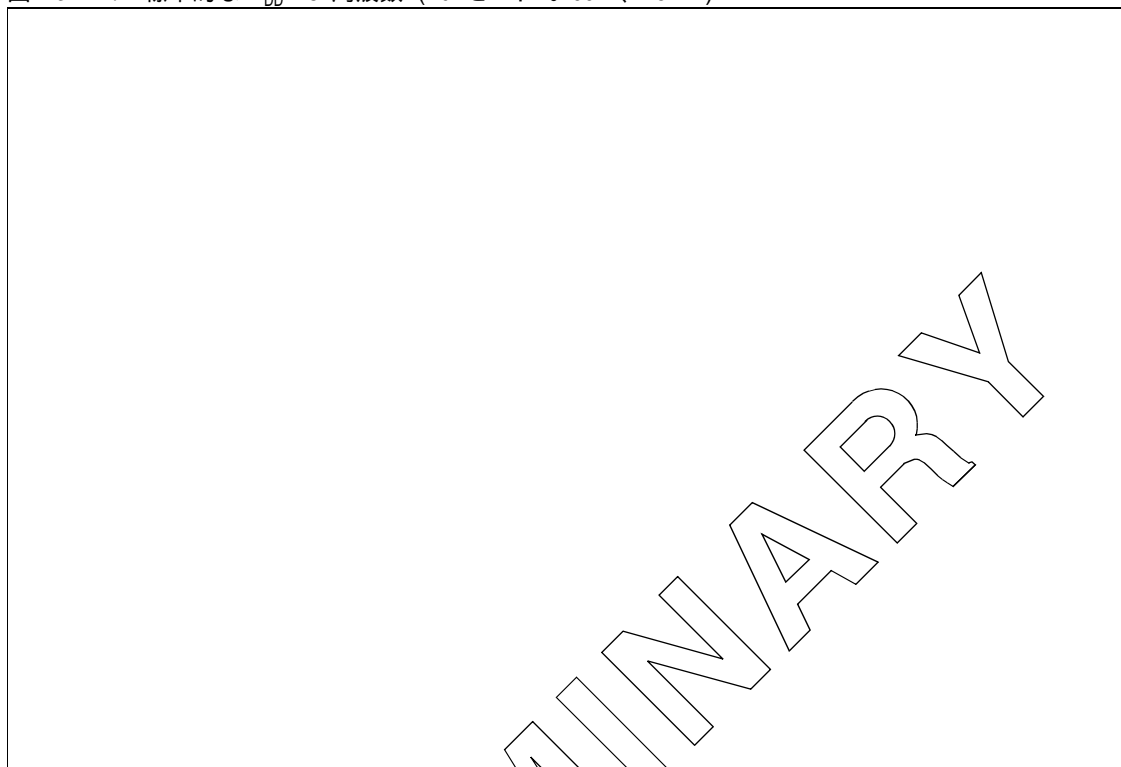
PIC16F8X

図 13-10： 標準的な I_{DD} VS 周波数 (RC モード @20PF、25)



規格値については最新版データシート (英語) を参照してください。

図 13-11: 標準的な I_{DD} VS 周波数 (RC モード @100PF、25)



PRELIMINARY

規格値については最新版データシート (英語) を参照してください。

PIC16F8X

図 13-12: 標準的な I_{DD} VS 周波数 (RC モード @300PF、25)



PRELIMINARY

規格値については最新版データシート (英語) を参照してください。

PIC16F8X

図 13-13: WDT タイムアウト周期 VS V_{DD} (RC モード @300PF、25)

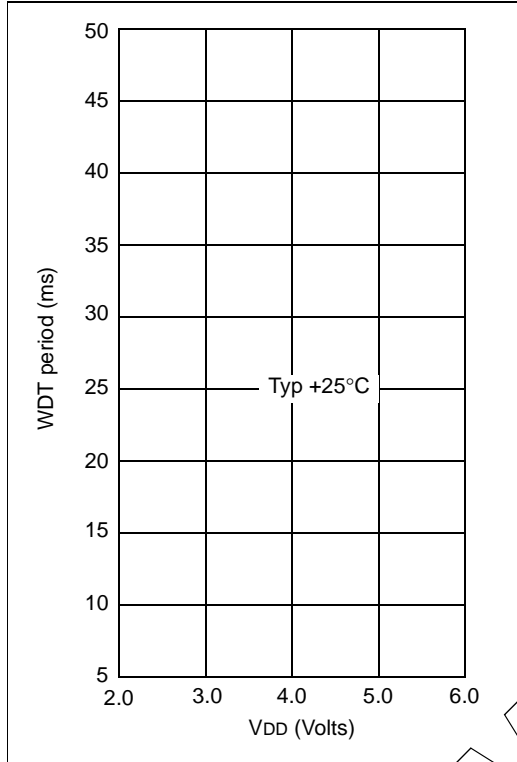
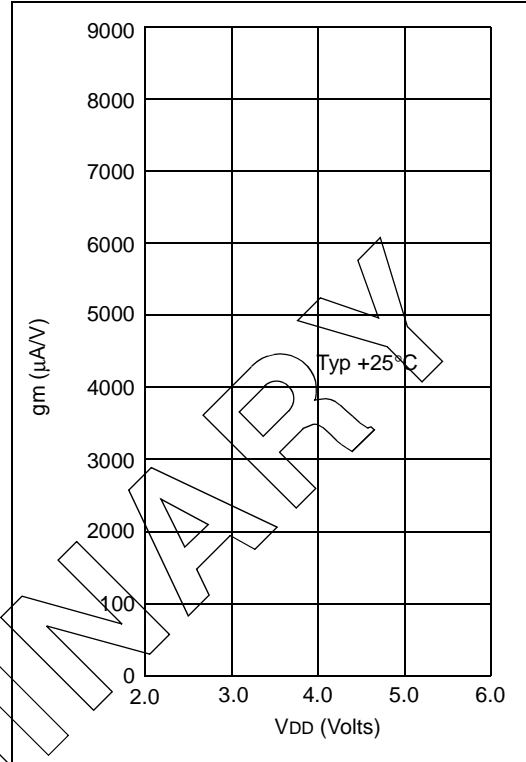


図 13-14: HS オシレータのトランスコンダクタンス (gm) VS V_{DD}



規格値については最新版データシート (英語) を参照してください。

PIC16F8X

図 13-15 : LP オシレータのトランスコンダクタンス (gm) VS V_{DD}

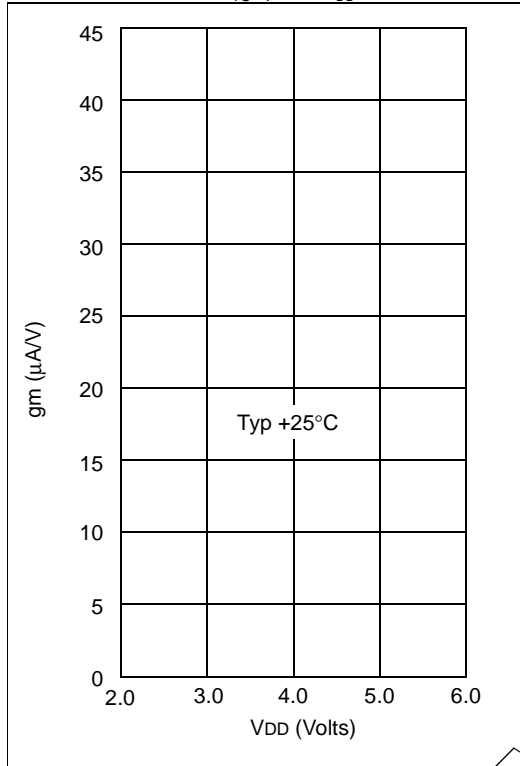
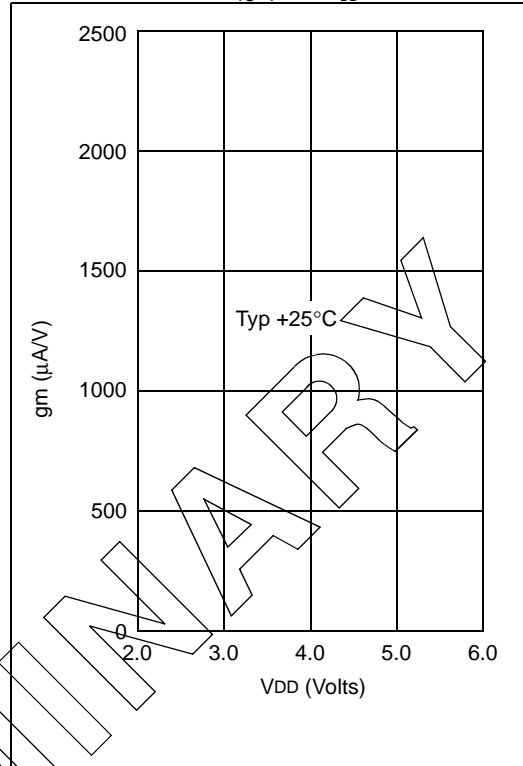


図 13-16 : XT オシレータのトランスコンダクタンス (gm) VS V_{DD}



規格値については最新版データシート (英語) を参照してください。

PIC16F8X

図 13-17: I_{OH} VS I_{OH} , $V_{DD} = 3V$

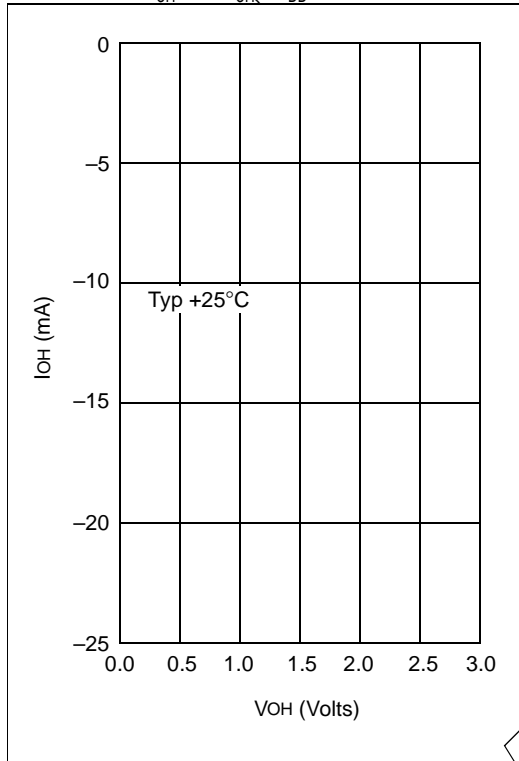


図 13-19: I_{OL} VS I_{OL} , $V_{DD} = 3V$

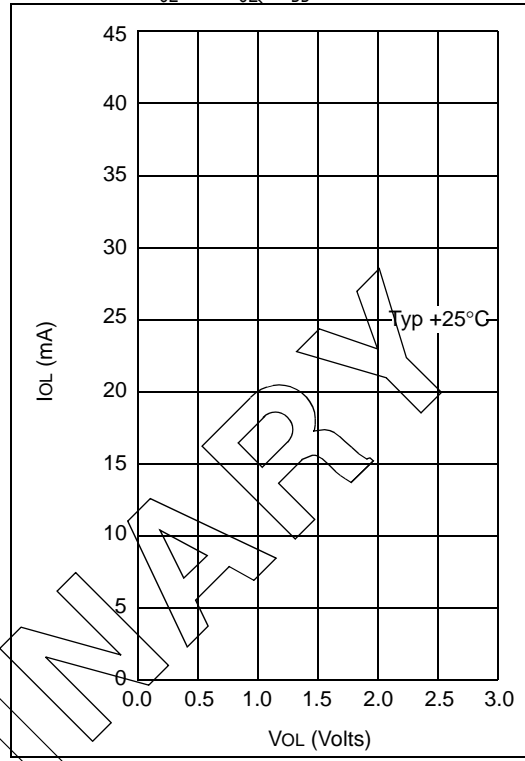


図 13-18: I_{OH} VS I_{OH} , $V_{DD} = 5V$

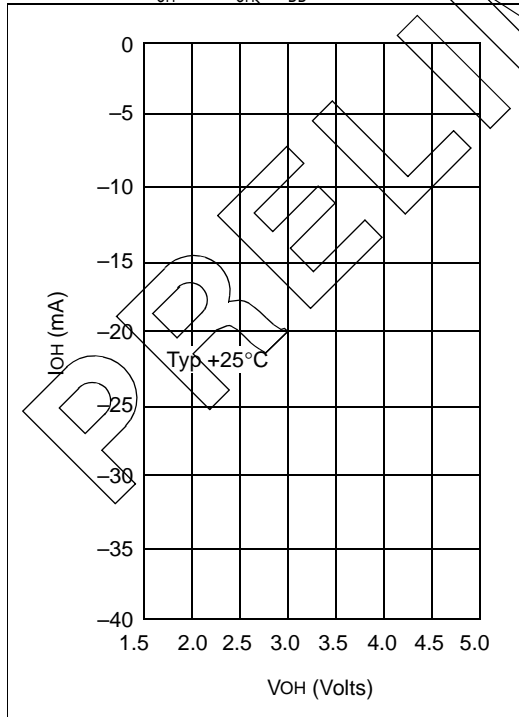
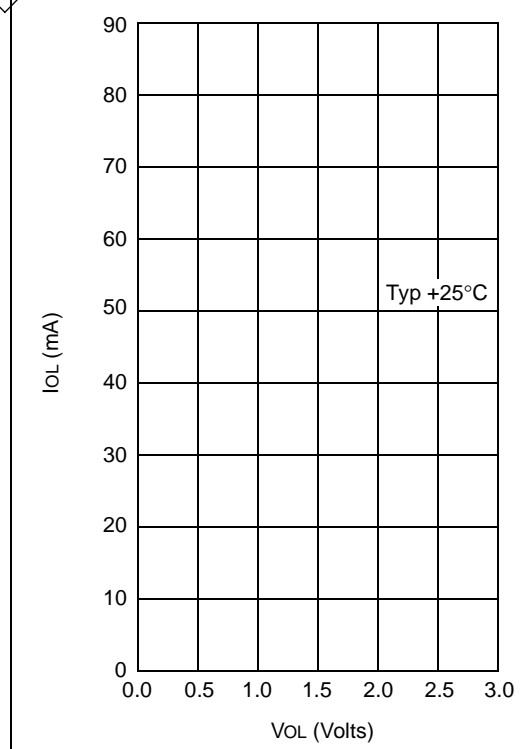


図 13-20: I_{OL} VS I_{OL} , $V_{DD} = 5V$



規格値については最新版データシート (英語) を参照してください。

PIC16F8X

図 13-21 : 標準的なデータメモリの消去 / ライトサイクルタイム VS V_{DD}

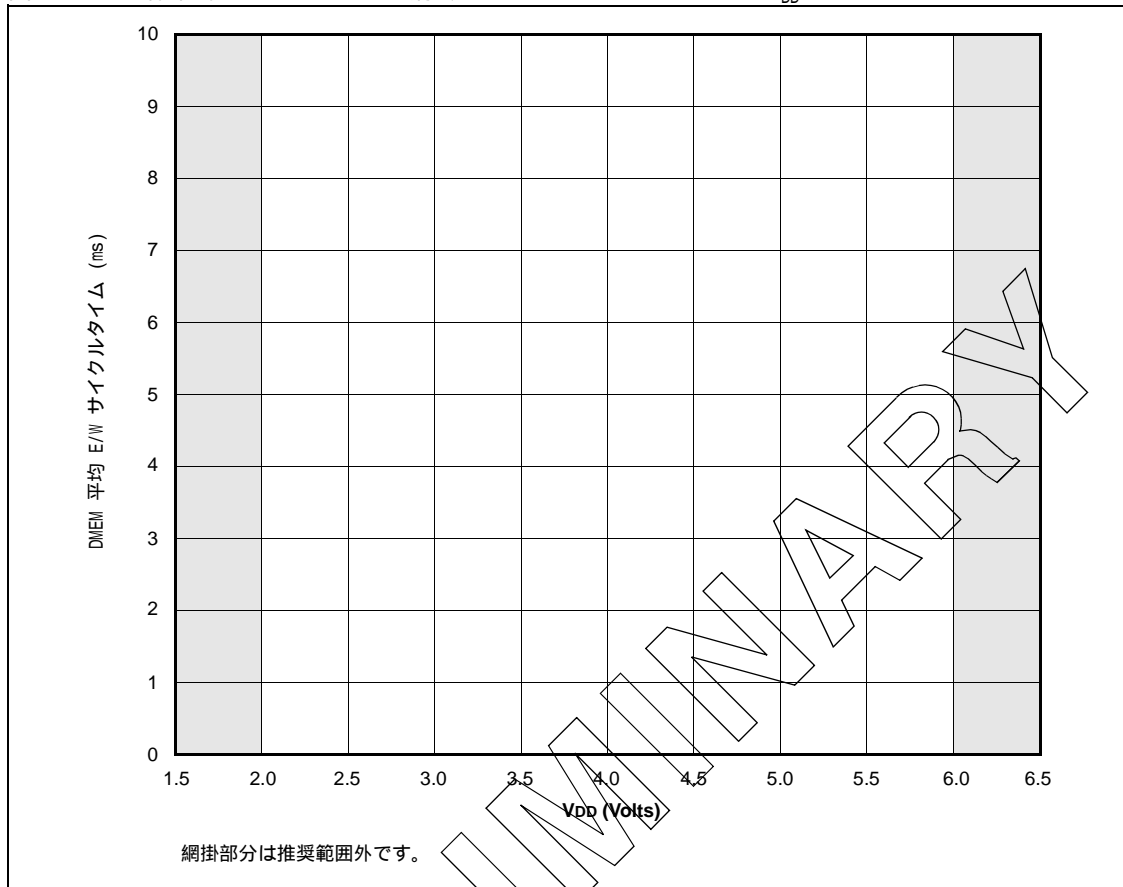


表 13-2 入力容量 *

ピン名	標準的な入力容量 (pF)	
	18L PDIP	18L SOIC
PORTA		
PORTB		
MCLR		
OSC1/CLKIN		
OSC2/CLKOUT		
T0CKI		

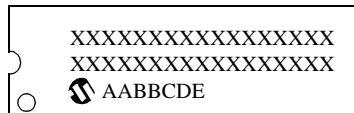
* すべての入力容量は平均 25 pF です。部品ごとの誤差 ± 25% (3 σ) の考慮が必要です。

規格値については最新版データシート (英語) を参照してください。

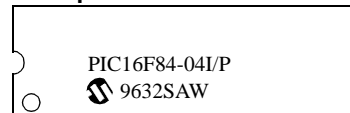
14.0 パッケージ

14.1 パッケージマーキング

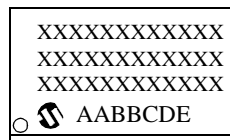
18L PDIP



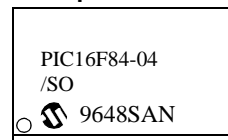
Example



18L SOIC



Example

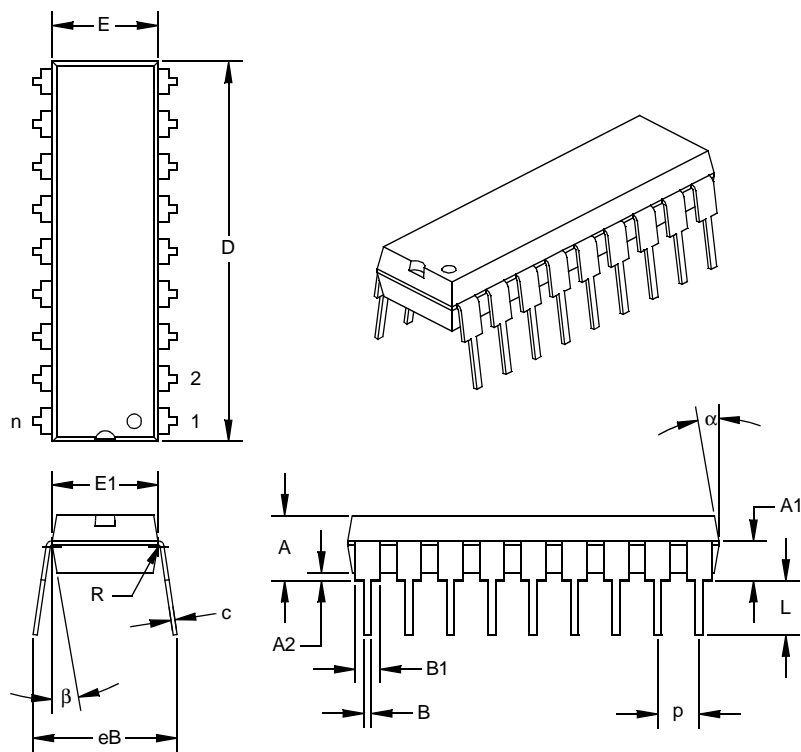


凡例：	XX...X	マイクロチップ製品番号 & ユーザー情報
	AA	年 (西暦の下 2 桁)
	BB	週 (1 月の第 1 週は "01" 週)
	C	ウェーハが製造された工場コード
		C = チャンドラー (アメリカ合衆国アリゾナ州)
		S = テンビ (アメリカ合衆国アリゾナ州)
	D	マスクレビジョン番号
	E	部品が組み立てられた工場または地域のアセンブリコード
注意： マイクロチップ製品番号を 1 行で表示できない場合は、次の行にまたがる場合があります。この場合、ユーザー情報に使用できる文字数が制限されます。		

* 標準 OTP マーキングは、マイクロチップ製品番号、年コード、週コード、工場コード、マスクレビジョン番号、およびアセンブリコードで構成されます。これを超える OTP マーキングについては、追加料金が加算されます。販売代理店にお問い合わせください。QTP デバイスについては、特別マーキングの追加料金が QTP 価格に含まれています。

PIC16F8X

パッケージタイプ : K04-007 18 リードプラスチックデュアルインライン (P) --- 300 ミル



Units		INCHES*			MILLIMETERS		
		MIN	NOM	MAX	MIN	NOM	MAX
Dimension Limits							
PCB Row Spacing							
Number of Pins	n						
Pitch	p						
Lower Lead Width	B						
Upper Lead Width	B1†						
Shoulder Radius	R						
Lead Thickness	c						
Top to Seating Plane	A						
Top of Lead to Seating Plane	A1						
Base to Seating Plane	A2						
Tip to Seating Plane	L						
Package Length	D‡						
Molded Package Width	E‡						
Radius to Radius Width	E1						
Overall Row Spacing	eB						
Mold Draft Angle Top	α						
Mold Draft Angle Bottom	β						

* コントロールパラメータ。

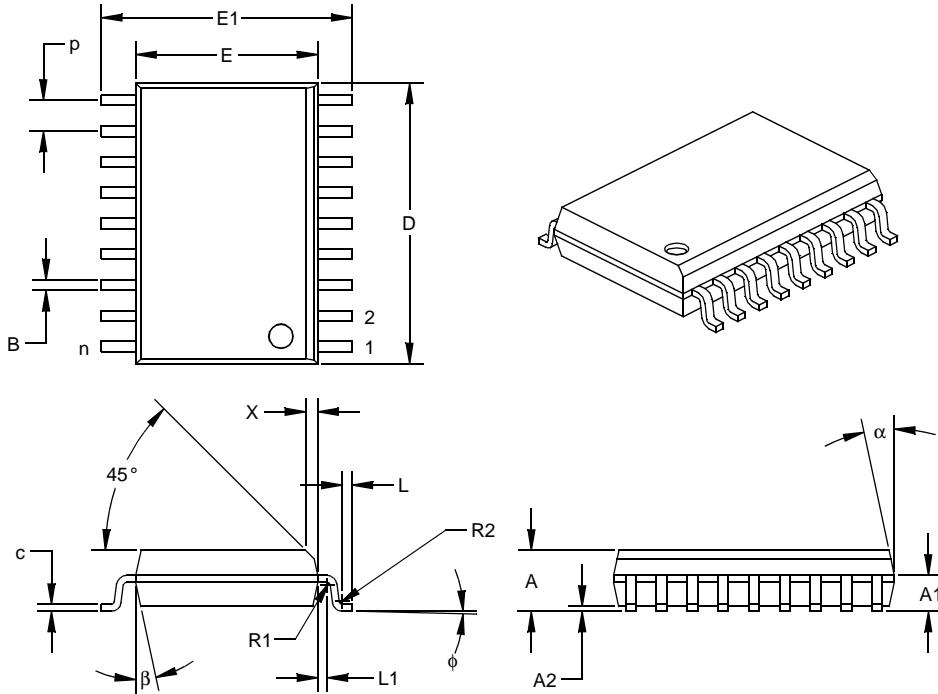
† "B1" には DAM バー の突出部は含まれません。DAM バー の突出部は、片側 0.003" (0.076 mm) または "B1" の寸法より 0.006" (0.152mm) を超えません。

‡ "D" および "E" にはモールドフラッシュまたは突出部は含まれません。モールドフラッシュまたは突出部は片側 0.010" (0.254 mm) または "D" や "E" の寸法より 0.020" (0.508mm) を超えません。

規格値については最新版データシート (英語) を参照してください。

PIC16F8X

製品タイプ: K04-051 18 リードプラスチックデュアルインライン (P) --- ワイド、300 ミル



Units		INCHES*			MILLIMETERS		
		MIN	NOM	MAX	MIN	NOM	MAX
Dimension Limits							
Pitch	p						
Number of Pins	n						
Overall Pack. Height	A						
Shoulder Height	A1						
Standoff	A2						
Molded Package Length	D [‡]						
Molded Package Width	E [‡]						
Outside Dimension	E1						
Chamfer Distance	X						
Shoulder Radius	R1						
Gull Wing Radius	R2						
Foot Length	L						
Foot Angle	φ						
Radius Centerline	L1						
Lead Thickness	c						
Lower Lead Width	B [†]						
Mold Draft Angle Top	α						
Mold Draft Angle Bottom	β						

* コントロールパラメータ。

† "B1" には DAM バーの突出部は含まれません。DAM バーの突出部は、片側 0.003" (0.076 mm) または "B1" の寸法より 0.006" (0.152mm) を超えません。

‡ "D" および "E" にはモールドフラッシュまたは突出部は含まれません。モールドフラッシュまたは突出部は片側 0.010" (0.254 mm) または "D" や "E" の寸法より 0.020" (0.508mm) を超えません。

規格値については最新版データシート (英語) を参照してください。

PIC16F8X

NOTES:

付録 A： PIC16C5X から PIC16F8X への機能改良点

以下に、PIC16C5X マイクロコントローラファミリーからの機能改良点をまとめます。

1. 命令ワード長が 14 ビットに増えました。これによって、プログラム メモリ (512 から 2K) とレジストリファイル (32 バイトから 128 バイト) のページ/バンクサイズが大きくなりました。
2. プログラムメモリページングの処理のために、PC ラッチレジスタ (PCLATH) が追加されました。PA2、PA1、および PA0 の各ビットがステータスレジスタから削除されました。
3. データメモリバンク選択が少し再定義されました。STATUS レジスタが変更されました。
4. RETURN、RETFIE、ADDLW、および SUBLW の 4 つの新しい命令が追加されました。TRIS と OPTION の 2 つの命令は PIC16C5X との互換性を維持していますが、将来は削除されます。
5. OPTION レジスタと TRIS レジスタはアドレス指定できます。
6. 割り込みが追加されました。割り込みベクタは 0004h です。
7. スタックサイズが 8 レベルに増えました。
8. リセットベクタが 0000h に変更されました。
9. レジスタのリセット状態を再検討しました。5 種類のリセット (およびウェークアップ) が識別できます。レジスタはリセットの種類によりリセットされます。
10. 割り込みによるスリープからの起動が追加されました。
11. より信頼性の高い電源 ON のために、オシレータスタートアップタイム (OST) とパワーアップタイム (PWRT) の 2 つのタイムが用意されました。この 2 つのタイムは、電源 ON およびスリープからの起動のときの不要な遅延を回避するために個別に起動されます。
12. PORTB には弱プルアップおよび変化への割り込みがあります。
13. TOCK1 は I/O ポートにもなります (RA4/TOCK1)。
14. FSR は 8 ビットになりました。
15. " インサーキットシリアルプログラミング " が可能になりました。ユーザは、 V_{DD} 、 V_{SS} 、 V_{PP} 、RB6 (クロック) および RB7 (データ入出力) の 5 つのピンだけを使用して PIC16CXX をプログラム書き込みできます。

付録 B： PIC16C5X と PIC16F8X とのコードの互換性

PIC16C5X 用に書かれたコードを PIC16F8X 用に交換するには、以下の手順を実行します。

1. CALL、GOTO 用のプログラムメモリページ選択 (PA2、PA1、PA0 ビット) を削除します。
2. 計算ジャンプ命令 (PC へのライト、PC への足し算など) を再確認して、ページビットが新しい方法で正しくセットされていることを確認します。
3. データメモリバンク選択を削除します。
4. STATUS、OPTION および FSR レジスタは内容変更されたので、すべてのそれらのレジスタへのライトする値を確認します。
5. リセットベクタを 0000h に変更します。

PIC16F8X

付録 C： このデータシートでの追加点

このデータシートでの追加点は以下の通りです。

1. PIC16F8X デバイスが追加された DC および AC 特性のグラフや表。
2. コンバージョンの注意点の付録が追加されました。ここでは、PIC16C84 から PIC16F84 または同様のデバイスへコンバージョンのために、相違点を説明しています。

付録 D： このデータシートでの変更点

このデータシートでの変更点は以下の通りです。

1. Errata の情報を追加しました。
2. オプション レジスタ名が OPTION から OPTION_REG に変更されました。これにより他のデータシートおよびヘッダファイルと共通で、OPTION コマンドと OPTION レジスタの名前の衝突がなくなります。
3. エラーを修正しました。
4. PIC16/17 マイクロコントローラについての付録を削除しました。

PIC16F8X

付録 E： コンバージョンの注意点 - PIC16C84 から /F84 および PIC16CR83/F84 へ

PIC16C84 から PIC16F84 へのコンバージョンに関する注意点を以下の表にまとめます。この注意点は PIC16C84 から PIC16F84 (プログラムおよびデータ RAM メモリのサイズを除いて PIC16F83 も同様) およ

び PIC16C84 と PIC16CR83 (フラッシュデバイスの ROM バージョン) へのコンバージョンにも適用できます。開発システムサポートは、すべての PIC16X8X デバイスで利用できます。

相違点	PIC16C84	PIC16F84
PWRTE ビットの極性が逆になりました。プログラム書き込み前にプログラマライターがこのビットを正しくセットしていることを確認してください。	PWRTE	PWRTE
PIC16F84 (および PIC16CR84) の RAM サイズが大きくなりました。これによってプログラムで問題が発生していないことを確認してください。	RAM = 36 バイト	RAM = 68 バイト
MCLR ピンにオンチップフィルタが付きました。MCLR ピンの入力信号にはリセットを生成するために長い Low パルスが必要となります。	MCLR パルス幅 (Low)	MCLR パルス幅 (Low)
一部の電氣的仕様が改善されました (例えば I _{PD} を参照)。 両方のデバイスの電氣的仕様を比較して、互換性に関する問題が発生していないことを確認してください。	I _{PD} (typ @ 2V) I _{PD} (最大 @ 4V、WDT オフ)	I _{PD} (typ @ 2V) I _{PD} (最大 @ 4V、WDT オフ)
PORTA および水晶オシレータの値が 500kHz 以下。	500 kHz 以下で動作する水晶オシレータの場合、PORTA<0>の状態が切り替わるとき、デバイスは不要な内部 Q クロックを発生します。	N/A
RB0/INT ピン	TTL	TTL/ST* (* このバッファは外部割り込みとして使用される場合、シュミットトリガ入力となります。)
EEADR<7:6> および IDD	EEADR<7:6> ビットをクリアすることを推奨します。どちらのビットがセットされているとき、両方のビットがクリアされるときより、最大 I _{DD} が大きくなります。	N/A
コードプロテクト	1 CP ビット	9 CP ビット
RC オシレータ回路の R _{EXT} の推奨値	REXT	REXT
GIE ビットの不要なセット	グローバル割り込みイネーブル (GIE) ビットをクリア中に割り込みが発生すると、割り込みサービスルーチン (RETFIE 命令) によって GIE ビットが再びセットされ、クリアされないことがあります。	N/A

規格値については最新版データシート (英語) を参照してください。

PIC16F8X

NOTES:

PIC16F8X

INDEX

Numerics

8.1 Configuration Bits 37

A

Absolute Maximum Ratings 73, 85

ALU 7

Architectural Overview 7

Assembler

MPASM Assembler 70

B

Block Diagram

Interrupt Logic 47

On-Chip Reset Circuit 41

RA3:RA0 and RA5 Port Pins 21

RA4 Pin 21

RB7:RB4 Port Pins 23

TMR0/WDT Prescaler 30

Watchdog Timer 50

Brown-out Protection Circuit 46

C

Carry 7

CLKIN 9

CLKOUT 9

Code Protection 37, 52

Compatibility, upward 3

Computed GOTO 18

Configuration Bits 37

D

DC Characteristics 75, 76, 77, 78, 87, 88, 89, 90

Development Support 69

Development Tools 69

Digit Carry 7

E

Electrical Characteristics 73, 85

External Power-on Reset Circuit 43

F

Family of Devices

PIC16C8X 3

FSR 19, 42

Fuzzy Logic Dev. System (*fuzzyTECH*®-MP) 71

G

GIE 47

I

I/O Ports 21

I/O Programming Considerations 25

ICEPIC Low-Cost PIC16CXXX In-Circuit Emulator 69

In-Circuit Serial Programming 37, 52

INDF 42

Instruction Format 53

Instruction Set

ADDLW 55

ADDWF 55

ANDLW 55

ANDWF 55

BCF 56

BSF 56

BTFS 56

BTFSS 57

CALL 57

CLRF 58

CLRWF 58

CLRWD 58

COMF 59

DECF 59

DECFSZ 59

GOTO 60

INCF 60

INCFSZ 61

IORLW 61

IORWF 62

MOVF 62

MOVLW 62

MOVWF 62

NOP 63

OPTION 63

RETFIE 63

RETLW 64

RETURN 64

RLF 65

RRF 65

SLEEP 66

SUBLW 66

SUBWF 67

SWAPF 67

TRIS 67

XORLW 68

XORWF 68

Section 53

Summary Table 54

INT Interrupt 48

INTCON 17, 42, 47, 48

INTEDG 48

Interrupts

Flag 47

Interrupt on Change Feature 23

Interrupts 37, 47

K

KeeLoq® Evaluation and Programming Tools 71

L

Loading of PC 18

M

MCLR 9, 41, 42

Memory Organization

Data Memory 12

Memory Organization 11

Program Memory 11

MP-DriveWay™ - Application Code Generator 71

MPLAB C 71

MPLAB Integrated Development Environment Software ... 70

O

OPCODE 53

OPTION 16, 42, 48

OSC selection 37

OSC1 9

OSC2 9

Oscillator

HS 39, 46

LP 39, 46

RC 39, 40

XT 39

Oscillator Configurations 39

PIC16F8X

P			
Paging, Program Memory	18		
PCL	18, 42		
PCLATH	18, 42		
PD	15, 41, 46		
PICDEM-1 Low-Cost PICmicro Demo Board	70		
PICDEM-2 Low-Cost PIC16CXX Demo Board	70		
PICDEM-3 Low-Cost PIC16CXXX Demo Board	70		
PICMASTER® In-Circuit Emulator	69		
PICSTART® Plus Entry Level Development System	69		
Pinout Descriptions	9		
POR	43		
Oscillator Start-up Timer (OST)	37, 43		
Power-on Reset (POR)	37, 42, 43		
Power-up Timer (PWRT)	37, 43		
Time-out Sequence	46		
Time-out Sequence on Power-up	44		
TO	15, 41, 46		
Port RB Interrupt	48		
PORTA	9, 21, 42		
PORTB	9, 23, 42		
Power-down Mode (SLEEP)	51		
Prescaler	29		
PRO MATE® II Universal Programmer	69		
Product Identification System	121		
R			
RBIF bit	23, 48		
RC Oscillator	46		
Read-Modify-Write	25		
Register File	12		
Reset	37, 41		
Reset on Brown-Out	46		
S			
Saving W Register and STATUS in RAM	49		
SEEVAL® Evaluation and Programming System	71		
SLEEP	37, 41, 51		
Software Simulator (MPLAB-SIM)	71		
Special Features of the CPU	37		
Special Function Registers	12		
Stack	18		
Overflows	18		
Underflows	18		
STATUS	7, 15, 42		
T			
time-out	42		
Timer0			
Switching Prescaler Assignment	31		
TOIF	48		
Timer0 Module	27		
TMR0 Interrupt	48		
TMR0 with External Clock	29		
Timing Diagrams			
Time-out Sequence	44		
Timing Diagrams and Specifications	80, 92		
TRISA	21		
TRISB	23, 42		
W			
W	42		
Wake-up from SLEEP	42, 51		
Watchdog Timer (WDT)	37, 41, 42, 50		
WDT	42		
Period	50		
		Programming Considerations	50
		Time-out	42
X			
XT	46		
Z			
Zero bit	7		

オンラインサポート

マイクロチップは Microchip World Wide Web (WWW) サイトにオンラインサポートを用意しています。

Web サイトは、ユーザーがファイルや情報を簡単に入手できるを目的としています。サイトを表示するには、Netscape や Microsoft Explorer などの Web ブラウザを使用してインターネットにアクセスする必要があります。ファイルは FTP ダウンロードでも利用できます。

マイクロチップのインターネット Web サイトへのアクセス

マイクロチップの Web サイトは、インターネットブラウザを使用して次のアドレスにアクセスすると利用できます。

www.microchip.com

FTP サービスは

<ftp://ftp.futureone.com/pub/microchip/>

Web サイトおよび FTP サイトでは、さまざまなサービスを提供しています。最新の開発ツール、データシート、アプリケーションノート、ユーザーズガイド、文献およびサンプルプログラムのファイルをダウンロードできます。マイクロチップの販売代理店、製品取扱店、工場窓口などのリストを含むさまざまな営業情報も入手できます。その他、ご利用いただけるデータは以下の通りです。

- ・ 最新のマイクロチップ プレスリリース
- ・ テクニカルサポートと FAQ
- ・ デザインのコツ
- ・ デバイスの Errata
- ・ 求人情報
- ・ マイクロチップ コンサルタントプログラムメンバーのリスト
- ・ マイクロチップ製品に関連する Web サイトへのリンク
- ・ 製品に関するお問い合わせ、開発システム、技術情報など
- ・ セミナーやイベントのリスト

システム情報とアップグレードホットライン

システム情報とアップグレードラインでは、マイクロチップの開発システムソフトウェア製品の最新版のリストを入手できます。さらに、このラインでは、現在入手可能なアップグレードキットの入手方法についての情報をご提供します。ホットラインの電話番号は以下の通りです。(英語)

1-800-755-2345 (アメリカおよびカナダの大部分の地域にお住まいの方)

1-602-786-7302 (上記以外の国にお住まいの方)

980106

Trademarks: The Microchip name, logo, PIC, PICSTART, PICMASTER and PRO MATE are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries. PICmicro, FlexROM, MPLAB and fuzzyLAB are trademarks and SQTP is a service mark of Microchip in the U.S.A.

fuzzyTECH is a registered trademark of Inform Software Corporation. IBM, IBM PC-AT are registered trademarks of International Business Machines Corp. Pentium is a trademark of Intel Corporation. Windows is a trademark and MS-DOS, Microsoft Windows are registered trademarks of Microsoft Corporation. CompuServe is a registered trademark of CompuServe Incorporated.

All other trademarks mentioned herein are the property of their respective companies.

PIC16F8X

アンケートにお答えください

マイクロチップ製品を正しくご理解いただけるように、より良いマニュアルにしたいと思っています。ご意見は、FAX: 045-471-6122 までご送付ください。

宛先: マイクロチップテクノロジーPICmicro 技術担当 総ページ数

RE: アンケートにお答えください

差出人: 名前 _____

会社名 _____

住所 _____

市町村 / 都道府県 / 郵便番号 / 国 _____

電話番号 (_____) _____ - _____ FAX: (_____) _____ - _____

(差し支えなければ) アプリケーション:

回答をご希望ですか はい ___ いいえ ___

デバイス: **PIC16F8X** 文献番号: **DS30430C**

質問:

1. このマニュアルの最も良い点はどこですか？

2. 開発するハードウェアおよびソフトウェアに必要な情報をこのマニュアルから得られましたか？

3. このデータシートは簡単に入手できましたか？

4. このデータシートに加えたほうが良いと思われることがあれば、教えてください。

5. このデータシートから削除してもかまわないと思われることがあれば、教えてください。

6. 誤りやわかりにくい記述がありましたら、お知らせください。

7. このマニュアルについて何かご要望がありましたら、お知らせください。

8. 弊社のソフトウェア、システム、シリコン製品について何かご要望がありましたら、お知らせください。

PIC16F8X

PIC16F8X の型番 / 品名

ご注文や価格や納期などの情報については販売代理店までお問い合わせください。

<u>PART NO.</u>	<u>-XX</u>	<u>X</u>	<u>/XX</u>	<u>XXX</u>	Examples:
Device	Frequency Range	Temperature Range	Package	Pattern	
Device	PIC16F8X ⁽²⁾ , PIC16F8XT ⁽³⁾ PIC16LF8X ⁽²⁾ , PIC16LF8XT ⁽³⁾ PIC16F8XA ⁽²⁾ , PIC16F8XAT ⁽³⁾ PIC16LF8XA ⁽²⁾ , PIC16LF8XAT ⁽³⁾ PIC16CR8X ⁽²⁾ , PIC16CR8XT ⁽³⁾ PIC16LCR8X ⁽²⁾ , PIC16LCR8XT ⁽³⁾				a) PIC16F84 -04/P 301 = Commercial temp., PDIP package, 4 MHz, normal VDD limits, QTP pattern #301. b) PIC16LF84 - 04/SO = Industrial temp., SOIC package, 200 kHz, Extended VDD limits. c) PIC16CR84 - 10I/P = ROM program memory, Industrial temp., PDIP package, 10MHz, normal VDD limits. Note 1: b = blank 2: F = Standard VDD range LF = Extended VDD range CR = ROM Version, Standard VDD range LCR = ROM Version, Extended VDD range 3: T = in tape and reel - SOIC, SSOP packages only.
Frequency Range	04 = 4 MHz 10 = 10 MHz 20 = 20 MHz				
Temperature Range	b ⁽¹⁾ = 0°C to +70°C (Commercial) I = -40°C to +85°C (Industrial)				
Package	P = PDIP SO = SOIC (Gull Wing, 300 mil body) SS = SSOP				
Pattern	3-digit Pattern Code for QTP, ROM (blank otherwise)				

営業とサポート

製品には、デバイスのデータシートとの動作上の相違点や推奨する回避方法などを記載した Errata シートが発行されている場合があります。Errata シートが存在するかどうかを確認するには、下記をご参照ください。

1. マイクロチップのインターネットホームページ (<http://www.microchip.com>)

また、この日本語データシートは参考資料です。最新版の英文データシートを参照してください。

PIC16F8X

NOTES:

PIC16F8X

NOTES:



WORLDWIDE SALES AND SERVICE

AMERICAS

Corporate Office

Microchip Technology Inc.
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 602-786-7200 Fax: 602-786-7277
Technical Support: 602 786-7627
Web: <http://www.microchip.com>

Atlanta

Microchip Technology Inc.
500 Sugar Mill Road, Suite 200B
Atlanta, GA 30350
Tel: 770-640-0034 Fax: 770-640-0307

Boston

Microchip Technology Inc.
5 Mount Royal Avenue
Marlborough, MA 01752
Tel: 508-480-9990 Fax: 508-480-8575

Chicago

Microchip Technology Inc.
333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 630-285-0071 Fax: 630-285-0075

Dallas

Microchip Technology Inc.
14651 Dallas Parkway, Suite 816
Dallas, TX 75240-8809
Tel: 972-991-7177 Fax: 972-991-8588

Dayton

Microchip Technology Inc.
Two Prestige Place, Suite 150
Miamisburg, OH 45342
Tel: 937-291-1654 Fax: 937-291-9175

Los Angeles

Microchip Technology Inc.
18201 Von Karman, Suite 1090
Irvine, CA 92612
Tel: 714-263-1888 Fax: 714-263-1338

New York

Microchip Technology Inc.
150 Motor Parkway, Suite 202
Hauppauge, NY 11788
Tel: 516-273-5305 Fax: 516-273-5335

San Jose

Microchip Technology Inc.
2107 North First Street, Suite 590
San Jose, CA 95131
Tel: 408-436-7950 Fax: 408-436-7955

Toronto

Microchip Technology Inc.
5925 Airport Road, Suite 200
Mississauga, Ontario L4V 1W1, Canada
Tel: 905-405-6279 Fax: 905-405-6253

ASIA/PACIFIC

Hong Kong

Microchip Asia Pacific
RM 3801B, Tower Two
Metroplaza
223 Hing Fong Road
Kwai Fong, N.T., Hong Kong
Tel: 852-2-401-1200 Fax: 852-2-401-3431

India

Microchip Technology Inc.
India Liaison Office
No. 6, Legacy, Convent Road
Bangalore 560 025, India
Tel: 91-80-229-0061 Fax: 91-80-229-0062

Japan

Microchip Technology Intl. Inc.
Benex S-1 6F
3-18-20, Shinyokohama
Kohoku-Ku, Yokohama-shi
Kanagawa 222 Japan
Tel: 81-45-471- 6166 Fax: 81-45-471-6122

Korea

Microchip Technology Korea
168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku
Seoul, Korea
Tel: 82-2-554-7200 Fax: 82-2-558-5934

Shanghai

Microchip Technology
RM 406 Shanghai Golden Bridge Bldg.
2077 Yan'an Road West, Hong Qiao District
Shanghai, PRC 200335
Tel: 86-21-6275-5700
Fax: 86 21-6275-5060

Singapore

Microchip Technology Taiwan
Singapore Branch
200 Middle Road
#07-02 Prime Centre
Singapore 188980
Tel: 65-334-8870 Fax: 65-334-8850

ASIA/PACIFIC (CONTINUED)

Taiwan, R.O.C

Microchip Technology Taiwan
10F-1C 207
Tung Hua North Road
Taipei, Taiwan, ROC
Tel: 886-2-2717-7175 Fax: 886-2-2545-0139

EUROPE

United Kingdom

Arizona Microchip Technology Ltd.
505 Eskdale Road
Winnersh Triangle
Wokingham
Berkshire, England RG41 5TU
Tel: 44-1189-21-5858 Fax: 44-1189-21-5835

France

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

Germany

Arizona Microchip Technology GmbH
Gustav-Heinemann-Ring 125
D-81739 München, Germany
Tel: 49-89-627-144 0 Fax: 49-89-627-144-44

Italy

Arizona Microchip Technology SRL
Centro Direzionale Colleoni
Palazzo Taurus 1 V. Le Colleoni 1
20041 Agrate Brianza
Milan, Italy
Tel: 39-39-6899939 Fax: 39-39-6899883

1/13/98

Microchip received ISO 9001 Quality System certification for its worldwide headquarters, design, and wafer fabrication facilities in January, 1997. Our field-programmable PICmicro™ 8-bit MCUs, Serial EEPROMs, related specialty memory products and development systems conform to the stringent quality standards of the International Standard Organization (ISO).

All rights reserved. © 2000, Microchip Technology Incorporated, USA. 1/00 Printed on recycled paper.

Information contained in this publication regarding device applications and the like is intended for suggestion only and may be superseded by updates. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights. The Microchip logo and name are registered trademarks of Microchip Technology Inc. in the U.S.A. and other countries. All rights reserved. All other trademarks mentioned herein are the property of their respective companies.