

## 20Pin Flash、 8Bit マイクロコントローラーと XLP テクノロジー

### ハイ・パフォーマンスRISC CPU:

- Cコンパイラに最適化されたアーキテクチャ
- 49の命令のみ  
動作速度:
  - DC ~ 20MHzクロック入力
  - DC ~ 200ns命令サイクル
- 自動コンテキスト保存による割り込み機能
- オプションのオーバーフロー/アンダーフロー・リセットを備えた16レベルのディープ・ハードウェア・スタック
- 直接、間接、相対アドレッシング・モード:
  - 2つの完全な16ビットファイル選択レジスタ (FSR)
  - FSRはプログラムとデータ・メモリーを読み取ることができます

### 柔軟な発振器構造:

- 16MHz内部発振器ブロック:
  - 工場ですら±1%にキャリブレーション (標準)
  - 16MHz ~ 31kHzのソフトウェアで選択可能な周波数範囲
- 31kHzの低電力内部発振器
- 最大20MHzの3つの外部クロックモード

### 特別なマイクロコントローラー機能:

- 動作電圧範囲:
  - 1.8Vから3.6V (PIC16LF1508/9)
  - 2.3Vから5.5V (PIC16F1508/9)
- ソフトウェア制御下で自己プログラム可能
- パワーオン・リセット (POR)
- パワーアップ・タイマー (PWRT)
- プログラム可能な低電力ブラウンアウト・リセット (LPBOR)
- 拡張ウォッチドッグ・タイマー (WDT):
  - 1msから256sまでのプログラム可能な期間
- プログラム可能なコード保護
- 2つのピンを介したインサーキット・シリアル・プログラミング™ (ICSP™)
- 強化された低電圧プログラミング (LVP)
- 2つのピンを介したインサーキット・デバッグ (ICD)
- 省電スリープ・モード:
  - 低電力スリープ・モード
  - 低電力BOR (LPBOR)
- 統合された温度インジケータ
- 128バイトの高耐久性フラッシュ
  - 100,000回の書き込みフラッシュ耐久性 (最小)

### メモリー:

- 最大8 Kwordsの線形プログラム・メモリー・アドレス指定
- 最大512バイトのリニア・データ・メモリー・アドレス指定
- 高耐久性フラッシュ・データ・メモリー (HEF)
  - 不揮発性データ・ストレージの場合は128バイト
  - 10万回の消去/書き込みサイクル

### eXtreme Low-Power (XLP)

#### 機能 (PIC16LF1508/9):

- スリープ電流:
  - 20nA @1.8V (標準)
- ウォッチドッグ・タイマー電流:
  - 260nA @1.8V (標準)
- 動作電流:
  - 30µA/MHz @1.8V (標準)
- 2次発振器電流:
  - 700nA @32kHz、1.8V、標準

### Peripheral Features:

- アナログ・デジタル・コンバーター (ADC):
  - 10ビットの解像度
  - 12の外部チャンネル
  - 3つの内部チャンネル:
    - 固定電圧リファレンス
    - D/Aコンバーター (DAC)
    - 温度インジケータチャンネル
  - 自動取得機能
  - スリープ中に変換が利用可能
- 5ビットD/Aコンバーター (DAC):
  - 外部で利用可能な出力
  - 正の参照選択
  - コンパレータとADCへの内部接続
- 2つのコンパレータ:
  - レール・ツー・レール入力
  - パワー・モード制御
  - ソフトウェアで制御可能なヒステリシス
- 電圧リファレンス:
  - 1.024V固定電圧リファレンス (FVR)、1×、2×、4×ゲイン出力レベル
- 18 I/Oピン (1ピンのみ入力):
  - 大電流シンク/ソース25mA/25mA
  - 個別にプログラム可能な弱プルアップ
  - 個別にプログラム可能な変更割り込み (IOC) ピン
- Timer0: 8ビットプログラマブル・プリスケイラ付き8ビット・タイマー/カウンター
- 強化されたTimer1:
  - プリスケイラ付き16ビット・タイマー/カウンター
  - 外部ゲート入力モード
- Timer2: 8ビット周期レジスター、プリスケイラおよびポストスケイラを備えた8ビット・タイマー/カウンター
- 4つの10ビットPWMモジュール
- SPIおよびI<sup>2</sup>Cを備えたマスター同期シリアル・ポート (MSSP):
  - 7ビット・アドレス・マスキング
  - SMBus/PMBus™の互換性

# PIC16(L)F1508/9

## 周辺機能(続き):

- 拡張ユニバーサル同期非同期レシーバー・トランスミッター (EUSART)
  - RS-232、RS-485およびLIN互換
  - 自動ボー検出
  - 起動時の自動ウェイクアップ
- 4つの構成可能ロジック・セル (CLC) モジュール:
  - 16の選択可能な入力ソース信号
  - モジュールごとに4つの入力
  - 組み合わせ/順次ロジック/状態/クロック機能のソフトウェア制御
  - AND/OR/XOR/Dフリップ/Dラッチ/SR/JK
  - 外部および内部ソースからの入力
  - ピンと周辺機器で利用可能な出力
  - スリープ中の動作
- 数値制御発振器 (NCO) :
  - 20ビット・アキュムレータ
  - 16ビット・インクリメント
  - 真の線形周波数制御
  - 高速クロック入力
  - 選択可能な出力モード
  - 固定デューティ・サイクル (FDC) モード
  - パルス周波数 (PF) モード
- 相補波形ジェネレーター (CWG) :
  - 8つの選択可能な信号源
  - 選択可能な立ち下がりエッジと立ち上がりエッジのデッドバンド制御
  - 極性制御
  - 4つの自動シャットダウン・ソース
  - 複数の入力ソース: PWM、CLC、NCO

## PIC12(L)F1501/PIC16(L)F150X FAMILY TYPES

デバイス	データシート目次	プログラムのメモリ (word)	フラッシュメモリ (byte)	I/O (2)	10 bit ADC (ch)	コンパレーター	DAコンバータ	タイマー (8bit/16bit)	PWM	EUSART	MSP (I <sup>2</sup> C/SPI)	CWG	CLC	NCO	デバッグ (1)	XL P
PIC12(L)F1501	(1)	1024	64	6	4	1	1	2/1	4	—	—	1	2	1	H	—
PIC16(L)F1503	(2)	2048	128	12	8	2	1	2/1	4	—	1	1	2	1	H	—
PIC16(L)F1507	(3)	2048	128	18	12	—	—	2/1	4	—	—	1	2	1	H	—
PIC16(L)F1508	(4)	4096	256	18	12	2	1	2/1	4	1	1	1	4	1	I/H	Y
PIC16(L)F1509	(4)	8192	512	18	12	2	1	2/1	4	1	1	1	4	1	I/H	Y

注1: デバッグ方法: (I) - オンチップに統合; (H) - デバッグ・ヘッダーの使用; (E) - エミュレーション・ヘッダーの使用。

2: 1つのピンは入力専用です。

データシート目次: (陰影のないデバイスについては、このドキュメントで説明しています。)

- 1: DS40001615 PIC12(L)F1501 Data Sheet, 8-Pin Flash, 8-bit Microcontrollers.
- 2: DS40001607 PIC16(L)F1503 Data Sheet, 14-Pin Flash, 8-bit Microcontrollers.
- 3: DS40001586 PIC16(L)F1507 Data Sheet, 20-Pin Flash, 8-bit Microcontrollers.
- 4: DS40001609 PIC16(L)F1508/9 Data Sheet, 20-Pin Flash, 8-bit Microcontrollers.

注釈: 他のスモール・フォーム・ファクター・パッケージの可用性とマーキング情報については、次のWebサイトをご覧ください <http://www.microchip.com/packaging> または最寄りの営業所にお問い合わせください。



# PIC16(L)F1508/9

## ピン割り当て表

表1: 20ピン割り当てテーブル (PIC16(L)F1508/9)

I/O	20pin PDIP/ SOIC / SSOP	20pin QFN/ UQFN	ADC	リファレンス	コン パ レ ー タ ー	タ イ マ ー	E U S A R T	MSS P	CWG	NCO	CLC	PWM	割 り 込 み	プ ル ア ッ プ	基本
RA0	19	16	AN0	DAC1OUT1	C1IN+	—	—	—	—	—	—	—	IOC	Y	ICSPDAT ICDDAT
RA1	18	15	AN1	VREF+	C1IN0- C2IN0-	—	—	—	—	—	CLC4IN1	—	IOC	Y	ICSPCLK ICDCLK
RA2	17	14	AN2	DAC1OUT2	C1OUT	T0CKI	—	—	CWG1FLT	—	CLC1	PWM3	INT/ IOC	Y	—
RA3	4	1	—	—	—	T1G(1)	—	SS(1)	—	—	CLC1IN0	—	IOC	Y	MCLR VPP
RA4	3	20	AN3	—	—	SOSCO T1G	—	—	—	—	—	—	IOC	Y	CLKOUT OSC2
RA5	2	19	—	—	—	SOSCI T1CKI	—	—	—	NCO1CLK	—	—	IOC	Y	CLKIN OSC1
RB4	13	10	AN10	—	—	—	—	SDA/ SDI	—	—	CLC3IN0	—	IOC	Y	—
RB5	12	9	AN11	—	—	—	—	RX/ DT	—	—	CLC4IN0	—	IOC	Y	—
RB6	11	8	—	—	—	—	—	SCL/ SCK	—	—	—	—	IOC	Y	—
RB7	10	7	—	—	—	—	—	TX/ CK	—	—	CLC3	—	IOC	Y	—
RC0	16	13	AN4	—	C2IN+	—	—	—	—	—	CLC2	—	—	—	—
RC1	15	12	AN5	—	C1IN1- C2IN1-	—	—	—	—	NCO1	—	PWM4	—	—	—
RC2	14	11	AN6	—	C1IN2- C2IN2-	—	—	—	—	—	—	—	—	—	—
RC3	7	4	AN7	—	C1IN3- C2IN3-	—	—	—	—	—	CLC2IN0	PWM2	—	—	—
RC4	6	3	—	—	C2OUT	—	—	—	CWG1B	—	CLC4 CLC2IN1	—	—	—	—
RC5	5	2	—	—	—	—	—	—	CWG1A	—	CLC1(1)	PWM1	—	—	—
RC6	8	5	AN8	—	—	—	—	SS	—	NCO1(1)	CLC3IN1	—	—	—	—
RC7	9	6	AN9	—	—	—	—	SDO	—	—	CLC1IN1	—	—	—	—
VDD	1	18	—	—	—	—	—	—	—	—	—	—	—	—	VDD
VSS	20	17	—	—	—	—	—	—	—	—	—	—	—	—	VSS

注1: APFCON (レジスタ11-1) レジスターで選択される代替ピン機能。

## 目次

1.0	デバイスの概要	8
2.0	拡張ミッドレンジCPU	13
3.0	メモリ構成	15
4.0	デバイス構成	40
5.0	発振器モジュール (フェイル・セーフ・クロック・モニター付き)	46
6.0	リセット	62
7.0	割り込み	70
8.0	パワーダウン・モード (スリープ)	83
9.0	ウォッチ・ドッグ・タイマー (WDT)	86
10.0	フラッシュ・プログラム・メモリ制御	90
11.0	I/Oポート	106
12.0	変更時の割り込み	119
13.0	固定電圧リファレンス (FVR)	124
14.0	温度インジケータ・モジュール	126
15.0	アナログ to デジタル・コンバーター (ADC) モジュール	128
16.0	5-Bit デジタル to アナログ・コンバーター (DAC) モジュール	142
17.0	コンパレータ・モジュール	145
18.0	タイマー0・モジュール	152
19.0	タイマー1・モジュールとゲート制御	155
20.0	タイマー2・モジュール	166
21.0	マスター同期シリアルポート (MSSP) モジュール	169
22.0	拡張ユニバーサル同期非同期レシーバー・トランスミッター (EUSART)	223
23.0	パルス幅変調 (PWM) モジュール	251
24.0	構成可能なロジック・セル (CLC)	257
25.0	数値制御発振器 (NCO) モジュール	273
26.0	相補波形発生器 (CWG) モジュール	280
27.0	インサーキット・シリアル・プログラミング (ICSP.)	292
28.0	命令セットの要約	294
29.0	電気仕様	309
30.0	DCおよびAC特性のグラフとチャート	339
31.0	開発サポート	380
32.0	パッケージ情報	384
付録A:	データシートの改訂履歴	397
	マイクロチップのウェブサイト	398
	顧客変更通知サービス	398
	顧客サポート	398
	製品識別システム	399

## 大切なお客様へ

大切なお客様にマイクロチップを正しくご利用いただくために、可能な限り最高のドキュメントを提供することが私たちの目的です。製品。この目的のために、私たちはあなたのニーズによりよく適合するように私たちの出版物を改善し続けます。私たちの出版物は、新しいボリュームとアップデートが導入されると、洗練され、強化されます。

この出版物について質問やコメントがある場合は、マーケティングコミュニケーション部門に連絡してください。docerrors@microchip.comに電子メールを送信するか、このデータシートの裏にあるリーダーレスポンスフォームを(480) 792-4150にFAXで送信してください。フィードバックをお待ちしています。

## 最新のデータシート

このデータシートの最新版を入手するには、次のWebサイトで登録してください：

<http://www.microchip.com>

データシートのバージョンを確認するには、ページの下部の外側の隅にある資料番号を調べます。文献番号の最後の文字はバージョン番号です（たとえば、DS30000AはドキュメントDS30000のバージョンAです）。

## 正誤表

正誤表シート、データシートとの運用上の小さな違いと推奨される回避策を説明し、現在のデバイスに存在する可能性があります。デバイス/ドキュメントの問題が判明すると、正誤表シートを発行します。正誤表は、シリコンのリビジョンとそれが適用されるドキュメントのリビジョンを指定します。

特定のデバイスの正誤表シートが存在するかどうかを確認するには、次のいずれかを確認してください：

- マイクロチップのワールドワイドウェブサイト；<http://www.microchip.com>
- 最寄りのマイクロチップ営業所（最後のページを参照）

営業所にお問い合わせの際は、ご使用のデバイス、シリコンのリビジョン、データシート（資料番号を含む）を明記してください。

## 顧客通知システム

当社のWebサイト（[www.microchip.com](http://www.microchip.com)）に登録して、すべての製品の最新情報を入手してください。



# PIC16(L)F1508/9

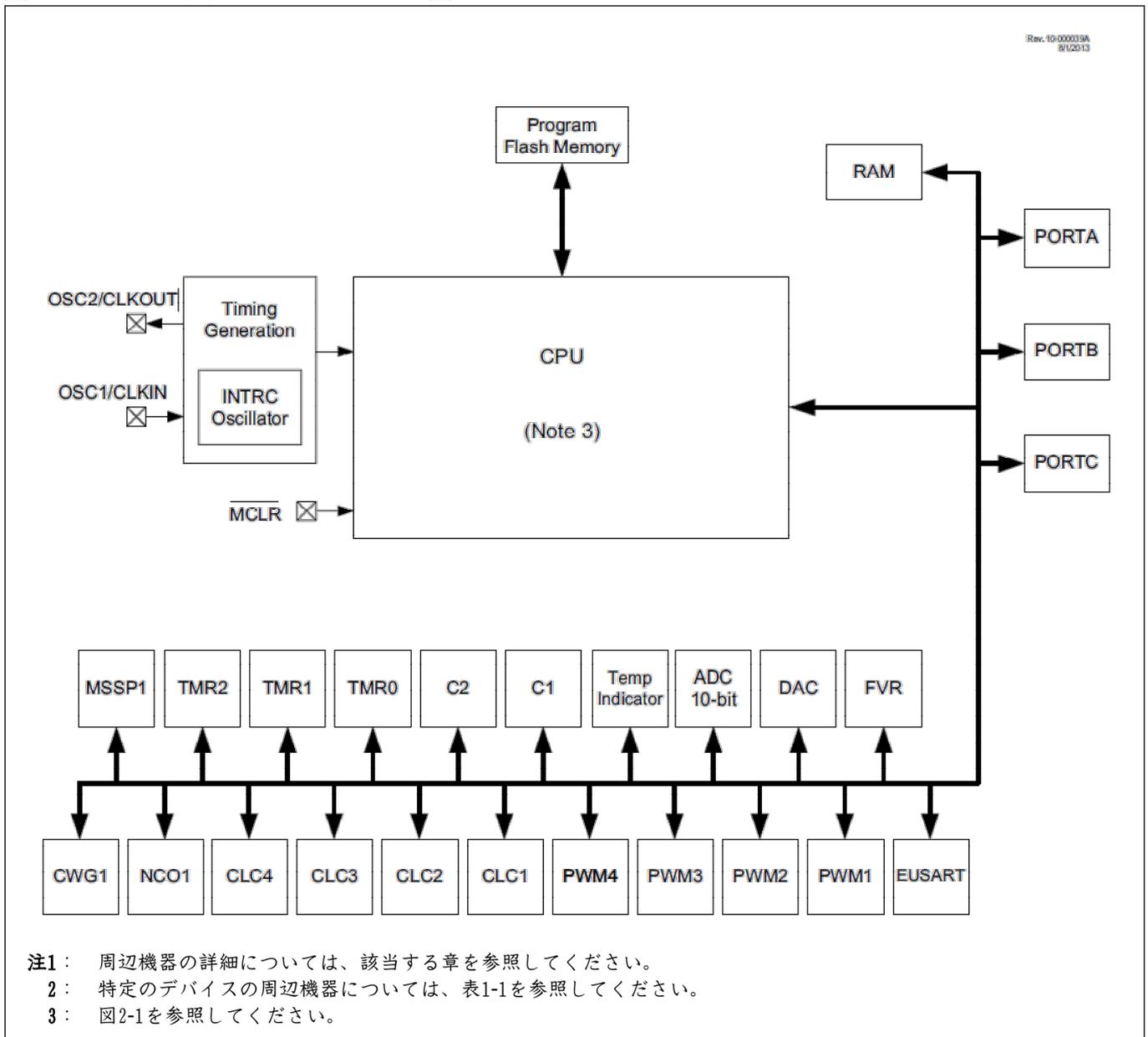
## 1.0 デバイスの概要

これらのデバイスのブロック図を図1-1に示し、利用可能な周辺機器を表1-1に示し、ピン配置の説明を表1-2に示します。

表1-1: デバイス周辺の概要

周辺		PIC12(L)F 1501	PIC16(L)F 1503	PIC16(L)F 1507	PIC16(L)F 1508	PIC16(L)F 1509
アナログ to デジタル・コンバーター(ADC)		●	●	●	●	●
相補波発生器(CWG)		●	●	●	●	●
デジタル to アナログ・コンバーター(DAC)		●	●		●	●
拡張ユニバーサル同期/ 非同期レシーバー/ 送信機 (EUSART)					●	●
固定参照電圧 (FVR)		●	●	●	●	●
数値制御発振器 (NCO)		●	●	●	●	●
温度インジケータ		●	●	●	●	●
コンパレータ						
	C1	●	●		●	●
	C2		●		●	●
構成可能なロジック・セル(CLC)						
	CLC1	●	●	●	●	●
	CLC2	●	●	●	●	●
	CLC3				●	●
	CLC4				●	●
マスター同期シリアル・ポート						
	MSSP1		●		●	●
PWMモジュール						
	PWM1	●	●	●	●	●
	PWM2	●	●	●	●	●
	PWM3	●	●	●	●	●
	PWM4	●	●	●	●	●
タイマー						
	タイマー0	●	●	●	●	●
	タイマー1	●	●	●	●	●
	タイマー2	●	●	●	●	●

図1-1: PIC16(L)F1508/9のブロック図



- 注1: 周辺機器の詳細については、該当する章を参照してください。
- 2: 特定のデバイスの周辺機器については、表1-1を参照してください。
- 3: 図2-1を参照してください。

# PIC16(L)F1508/9

表1-2: PIC16(L)F1508/9のピン配置の説明:

名前	機能	入力 タイプ	出力 タイプ	説明
RA0/AN0/C1IN+/DAC1OUT1/ ICSPDAT/ICDDAT	RA0	TTL	CMOS	汎用I/O。
	AN0	AN	—	ADCチャンネル入力。
	C1IN+	AN	—	コンパレータ正入力。
	DAC1OUT1	—	AN	デジタル to アナログ・コンバータ出力。
	ICSPDAT	ST	CMOS	ICSP™.データI/O。
	ICDDAT	ST	CMOS	インサーキット・デバッグ・データ。
RA1/AN1/CLC4IN1/VREF+/ C1IN0-/C2IN0-/ICSPCLK/ ICDCLK	RA1	TTL	CMOS	汎用I/O。
	AN1	AN	—	ADCチャンネル入力。
	CLC4IN1	ST	—	構成可能なロジック・セルのソース入力。
	VREF+	AN	—	ADC正電圧リファレンス入力。
	C1IN0-	AN	—	コンパレータの負入力。
	C2IN0-	AN	—	コンパレータの負入力。
	ICSPCLK	ST	—	ICSPプログラミング・クロック。
	ICDCLK	ST	—	インサーキット・デバッグ・クロック。
RA2/AN2/C1OUT/DAC1OUT2/ T0CKI/INT/PWM3/CLC1/ CWG1FLT	RA2	ST	CMOS	汎用I/O。
	AN2	AN	—	ADCチャンネル入力。
	C1OUT	—	CMOS	コンパレータ出力。
	DAC1OUT2	—	AN	デジタル to アナログ・コンバータ出力。
	T0CKI	ST	—	タイマー0・クロック入力。
	INT	ST	—	外部割り込み。
	PWM3	—	CMOS	PWM出力。
	CLC1	—	CMOS	構成可能なロジック・セル・ソース出力。
RA3/CLC1IN0/VPP/T1G(1)/SS(1)/ MCLR	RA3	TTL	—	IOCおよびWPUによる汎用入力。
	CLC1IN0	ST	—	構成可能なロジック・セルのソース入力。
	VPP	HV	—	プログラミング電圧
	T1G	ST	—	タイマー1・ゲート入力
	SS	ST	—	スレーブ選択入力。
	MCLR	ST	—	内部プルアップ付きマスタークリア。
RA4/AN3/SOSCO/ CLKOUT/T1G	RA4	TTL	CMOS	汎用I/O。
	AN3	AN	—	ADCチャンネル入力。
	SOSCO	XTAL	XTAL	第2オシレーター接続。
	CLKOUT	—	CMOS	FOSC/4 出力。
	T1G	ST	—	タイマー1・ゲート入力。
RA5/CLKIN/T1CKI/NCO1CLK/ SOSCI	RA5	TTL	CMOS	汎用I/O。
	CLKIN	CMOS	—	外部クロック入力 (ECモード)。
	T1CKI	ST	—	タイマー1・クロック入力。
	NCO1CLK	ST	—	数値制御オシレーター・クロック・ソース入力。
	SOSCI	XTAL	XTAL	第2オシレーター接続。

記録: AN=アナログ入力または出力      CMOS=CMOS互換入力または出力      OD=オープン・ドレイン  
TTL=TTL互換入力      ST=シュミット・トリガー入力でCMOSレベル      I<sup>2</sup>C=シュミット・トリガー入力でI<sup>2</sup>Cレベル  
HV=高電圧      XTAL=クリスタル

注1: APFCON (レジスター11-1) レジスターで選択される代替ピン機能。

表1-2: PIC16(L)F1508/9のピン配置の説明(続き):

名前	機能	入力 タイプ	出力 タイプ	説明
RB4/AN10/CLC3IN0/SDA/SDI	RB4	TTL	CMOS	汎用I/O。
	AN10	AN	—	ADCチャンネル入力。
	CLC3IN0	ST	—	構成可能なロジック・セルのソース入力。
	SDA	I <sup>2</sup> C	OD	I <sup>2</sup> Cデータ入力/出力。
	SDI	CMOS	—	SPIデータ入力。
RB5/AN11/CLC4IN0/RX/DT	RB5	TTL	CMOS	汎用I/O。
	AN11	AN	—	ADCチャンネル入力。
	CLC4IN0	ST	—	構成可能なロジック・セルのソース入力。
	RX	ST	—	USART非同期入力。
	DT	ST	CMOS	USART同期データ。
RB6/SCL/SCK	RB6	TTL	CMOS	汎用I/O。
	SCL	I <sup>2</sup> C	OD	I <sup>2</sup> Cクロック。
	SCK	ST	CMOS	SPIクロック。
RB7/CLC3/TX/CK	RB7	TTL	CMOS	汎用I/O。
	CLC3	—	CMOS	構成可能なロジック・セルのソース出力。
	TX	—	CMOS	USART非同期送信。
	CK	ST	CMOS	USART同期クロック。
RC0/AN4/CLC2/C2IN+	RC0	TTL	CMOS	汎用I/O。
	AN4	AN	—	ADCチャンネル入力。
	CLC2	—	CMOS	構成可能なロジック・セルのソース出力。
	C2IN+	AN	—	コンパレータ正入力。
RC1/AN5/C1IN1-/C2IN1-/PWM4/ NCO1	RC1	TTL	CMOS	汎用I/O。
	AN5	AN	—	ADCチャンネル入力。
	C1IN1-	AN	—	コンパレータ負入力。
	C2IN1-	AN	—	コンパレータ負入力。
	PWM4	—	CMOS	PWM出力。
	NCO1	—	CMOS	数値制御発振器のソース出力。
RC2/AN6/C1IN2-/C2IN2-	RC2	TTL	CMOS	汎用I/O。
	AN6	AN	—	ADCチャンネル入力。
	C1IN2-	AN	—	コンパレータ負入力。
	C2IN2-	AN	—	コンパレータ負入力。
RC3/AN7/C1IN3-/C2IN3-/PWM2/ CLC2IN0	RC3	TTL	CMOS	汎用I/O。
	AN7	AN	—	ADCチャンネル入力。
	C1IN3-	AN	—	コンパレータ負入力。
	C2IN3-	AN	—	コンパレータ負入力。
	PWM2	—	CMOS	PWM出力
	CLC2IN0	ST	—	構成可能なロジック・セルのソース入力。
RC4/C2OUT/CLC2IN1/CLC4/ CWG1B	RC4	TTL	CMOS	汎用I/O。
	C2OUT	—	CMOS	コンパレータ出力。
	CLC2IN1	ST	—	構成可能なロジック・セルのソース入力。
	CLC4	—	CMOS	構成可能なロジック・セルのソース出力。
	CWG1B	—	CMOS	CWG相補出力。

記録: AN=アナログ入力または出力      CMOS=CMOS互換入力または出力      OD=オープン・ドレイン  
TTL=TTL互換入力      ST=シュミット・トリガー入力でCMOSレベル      I<sup>2</sup>C=シュミット・トリガー入力でI<sup>2</sup>Cレベル  
HV=高電圧      XTAL=クリスタル

注1: APFCON (レジスター11-1) レジスターで選択される代替ピン機能。

# PIC16(L)F1508/9

表1-2: PIC16(L)F1508/9のピン配置の説明(続き):

名前	機能	入力 タイプ	出力 タイプ	説明
RC5/PWM1/CLC1(1)/ CWG1A	RC5	TTL	CMOS	汎用I/O。
	PWM1	—	CMOS	PWM出力。
	CLC1	—	CMOS	構成可能なロジック・セルのソース出力。
	CWG1A	—	CMOS	CWG一次出力。
RC6/AN8/NCO1(1)/CLC3IN1/ SS	RC6	TTL	CMOS	汎用I/O。
	AN8	AN	—	ADCチャンネル入力。
	NCO1	—	CMOS	数値制御発振器ソース出力。
	CLC3IN1	ST	—	構成可能なロジック・セルのソース入力。
	SS	ST	—	第2次選択入力。
RC7/AN9/CLC1IN1/SDO	RC7	TTL	CMOS	汎用I/O。
	AN9	AN	—	ADCチャンネル入力。
	CLC1IN1	ST	—	構成可能なロジック・セルのソース入力。
	SDO	—	CMOS	SPIデータ出力。
VDD	VDD	電源	—	正の供給。
VSS	VSS	電源	—	グラウンド基準。

記録: AN=アナログ入力または出力      CMOS=CMOS互換入力または出力      OD=オープン・ドレイン  
TTL=TTL互換入力      ST=シュミットトリガー入力でCMOSレベル      I<sup>2</sup>C=シュミットトリガー入力でI<sup>2</sup>Cレベル  
HV=高電圧      XTAL=クリスタル

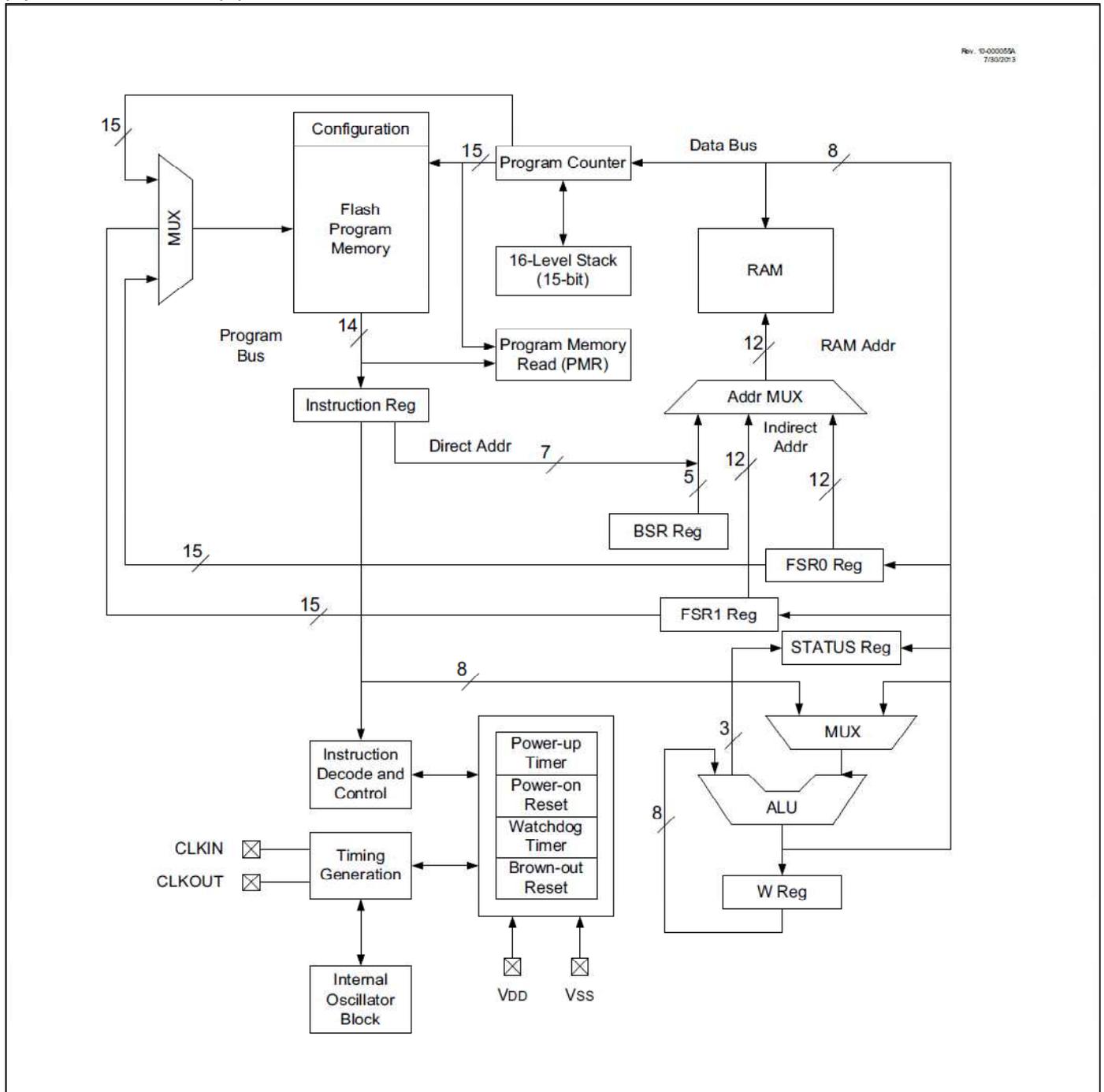
注1: APFCON (レジスター11-1) レジスターで選択される代替ピン機能。

## 2.0 拡張ミッドレンジCPU

このデバイス・ファミリには、拡張ミッドレンジ8ビットCPUコアが含まれています。CPUには49の命令があります。割り込み機能には、自動コンテキスト保存が含まれます。ハードウェア・スタックは16レベルの深さで、オーバーフローおよびアンダーフロー・リセット機能を備えています。直接、間接、および相対アドレッシング・モードを使用できます。2つのファイル選択レジスタ（FSR）は、プログラムおよびデータ・メモリーを読み取る機能を提供します。

- 自動割り込みコンテキスト保存。
- オーバーフローとアンダーフローを備えた16レベルのスタック。
- ファイル選択レジスタ。
- 命令セット

図2-1: コアブロック図



## 2.1 自動割り込みコンテキスト保存

割り込み中に、特定のレジスターは自動的にシャドウ・レジスターに保存され、割り込みから戻るときに復元されます。これにより、スタック・スペースとユーザー・コードが節約されます。詳細はセクション7.5「コンテキストの自動保存」を参照してください。

## 2.2 オーバーフローとアンダーフローのある16レベルのスタック

これらのデバイスには、15ビット幅で16ワード深さのハードウェア・スタック・メモリーがあります。スタック・オーバーフローまたはアンダーフローにより、PCONレジスターの適切なビット（STKOVFまたはSTKUNF）が設定されます。ソフトウェア・リセットが発生します。詳細はセクション3.5「スタック」を参照してください。

## 2.3 ファイル選択レジスタ

2つの16ビット・ファイル選択レジスター（FSR）があります。FSRはすべてのファイル・レジスターとプログラム・メモリーにアクセスできます。これにより、すべてのメモリーに対して1つのデータ・ポインタが許可されます。FSRがプログラム・メモリーを指す場合、INDFを使用してデータをフェッチできるようにする命令には、1つの追加の命令サイクルがあります。汎用メモリーもリニアにアドレス指定できるようになりました。80バイトを超える連続したデータにアクセスする機能を提供します。FSRをサポートするための新しい手順もあります。詳細はセクション3.6「間接アドレス指定」を参照してください。

## 2.4 命令セット

拡張ミッド・レンジCPUには、CPUの機能をサポートするための49の命令があります。詳細はセクション28.0「命令セットの概要」を参照してください。

### 3.0 メモリー組織

これらのデバイスには、次のタイプのメモリーが含まれています：

- プログラム・メモリー
  - 構成ワード
  - デバイスID
  - ユーザーID
  - フラッシュ・プログラム・メモリー
- データ・メモリー
  - コア・レジスター
  - 特殊機能レジスター
  - 汎用RAM
  - 共通RAM

次の機能は、プログラム・メモリーとデータ・メモリーのアクセスと制御に関連しています：

- PCLおよびPCLATH
- スタック
- 間接アドレス指定

### 3.1 プログラムのメモリー構成

拡張ミッド・レンジ・コアには、32K×14のプログラム・メモリー空間をアドレス指定できる15ビット・プログラム・カウンタがあります。表3-1に、実装されているメモリー・サイズを示します。これらの境界を超える場所にアクセスすると、実装されたメモリー空間内でラップアラウンドが発生します。リセット・ベクターは0000hにあり、割り込みベクターは0004hにあります（図3-1を参照）。

### 3.2 高耐久性フラッシュ

このデバイスは、データEEPROMの代わりに、128バイト・セクションの高耐久性プログラム・フラッシュ・メモリー（PFM）を備えています。この領域は、最終製品の寿命にわたって頻繁に更新されることが予想される不揮発性データ・ストレージに特に適しています。PFMへのデータの書き込みの詳細は、セクション10.2「フラッシュ・プログラム・メモリーの概要」を参照してください。FSRレジスターを使用してPFMに格納されているバイト・データを読み取る方法の詳細は、セクション3.2.1.2「FSRによる間接読み取り」を参照してください。

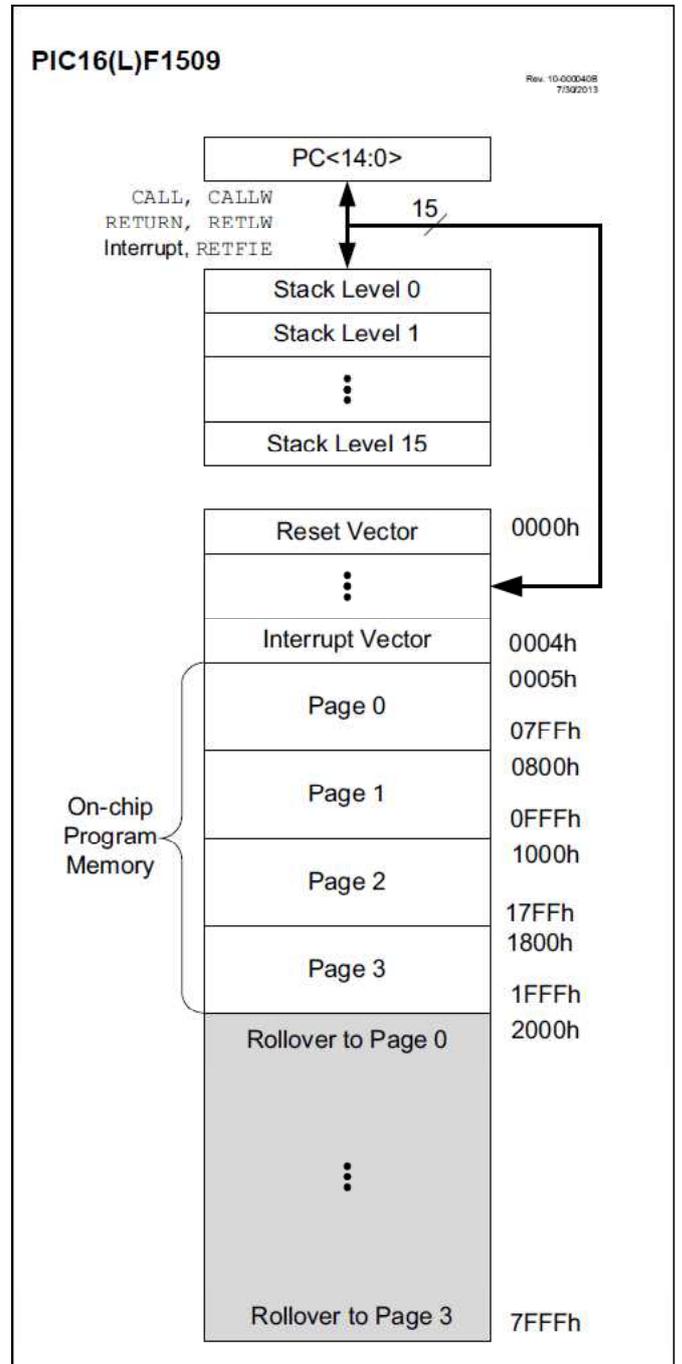
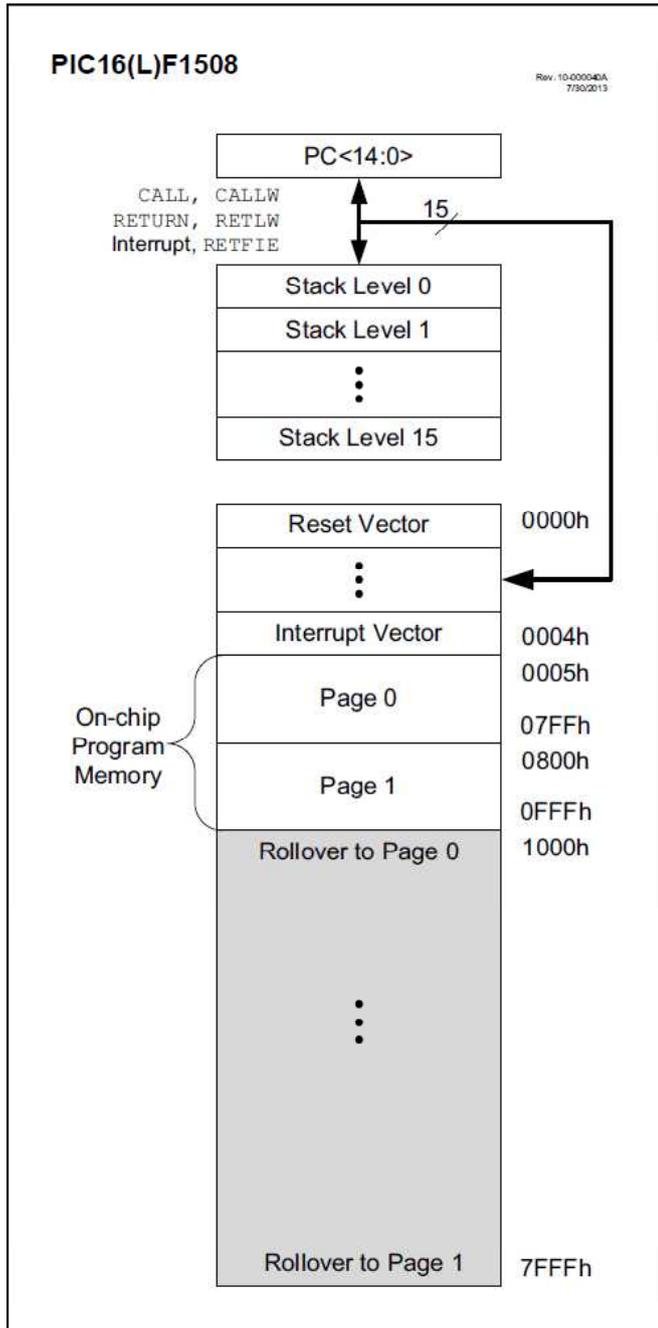
表3-1: デバイスのサイズとアドレス

デバイス	プログラム・メモリー空間 (Words)	最後のプログラム・ メモリー・アドレス	高耐久性フラッシュ・メモリー ・アドレス範囲 <sup>(1)</sup>
PIC16LF1508 PIC16F1508	4,096	0FFFh	0F80h ~ 0FFFh
PIC16LF1509 PIC16F1509	8,192	1FFFh	1F80h ~ 1FFFh

注1： 高耐久性フラッシュは、範囲内の各アドレスの下位バイトに適用されます。

# PIC16(L)F1508/9

図3-1: PIC16(L)F1508のプログラム・メモリーマップとスタック



### 3.2.1 プログラムメモリをデータとして読み取る

プログラム・メモリーの定数にアクセスする方法は2つあります。最初の方法は、**RETLW**命令のテーブルを使用することです。2つ目の方法は、FSRをプログラム・メモリーを指すように設定することです。

#### 3.2.1.1 RETLW命令

**RETLW**命令を使用して、定数のテーブルにアクセスできます。このようなテーブルを作成するための推奨方法を例3-1に示します。

#### 例3-1: RETLW命令

```
constants
    BRW                ;Add Index in W to
                    ;program counter to
                    ;select data
    RETLW DATA0      ;Index0 data
    RETLW DATA1      ;Index1 data
    RETLW DATA2
    RETLW DATA3
my_function
    ;... LOTS OF CODE...
    MOVLW DATA_INDEX
    call constants
    ;... THE CONSTANT IS IN W
```

**BRW**命令を使用すると、このタイプのテーブルの実装が非常に簡単になります。コードが前世代のマイクロコントローラーとの移植性を維持する必要がある場合、その場合、**BRW**命令は使用できないため、古いテーブル読み取りメソッドを使用する必要があります。

#### 3.2.1.2 FSRによる間接読み取り

FSRxHレジスターのビット7を設定し、一致するINDFxレジスターを読み取ることにより、プログラム・メモリーにデータとしてアクセスできます。**MOVIW**命令は、アドレス指定されたワードの下位8ビットをWレジスターに配置します。プログラム・メモリーへの書き込みは、INDFレジスターを介して実行できません。FSRを介してプログラム・メモリーにアクセスする命令は、完了するまでに1つの追加の命令サイクルを必要とします。例3-2は、FSRを介したプログラム・メモリーへのアクセスを示しています。

**HIGH**演算子は、ラベルがプログラム・メモリー内の場所を指す場合、ビット<7>を設定します。

#### 例3-2: FSRを介したプログラムメモリーへのアクセス

```
constants
    DW DATA0          ;First constant
    DW DATA1          ;Second constant
    DW DATA2
    DW DATA3
my_function
    ;... LOTS OF CODE...
    MOVLW DATA_INDEX
    ADDLW LOW constants
    MOVWF FSR1L
    MOVLW HIGH constants ;MSb sets
                        automatically
    MOVWF FSR1H
    BTFSC STATUS, C    ;carry from ADDLW?
    INCF FSR1h, f      ;yes
    MOVIW 0[FSR1]
    ;THE PROGRAM MEMORY IS IN W
```

## 3.3 データ・メモリーの構成

データ・メモリーは、32バイトのメモリー・バンクに分割され、1バンクは128バイトです。各バンクは(図3-2)で構成されています。

- 12コア・レジスタ
- 20の特殊機能レジスタ (SFR)
- 最大80バイトの汎用RAM (GPR)
- 16バイトの一般的RAM

アクティブなバンクは、バンク選択レジスタ (BSR) にバンク番号を書き込むことによって選択されます。未実装のメモリーは「0」として読み込まれます。すべてのデータ・メモリーは、直接 (ファイル・レジスタを使用する命令を介して)、または2つのファイル選択レジスタ (FSR) を介して間接的にアクセスできます。詳細はセクション3.6「間接アドレス指定」を参照してください。

データ・メモリーは12ビットのアドレスを使用します。アドレスの上位5ビットはバンク・アドレスを定義し、下位7ビットはそのバンクのレジスタ/RAMを選択します。

### 3.3.1 コア・レジスタ

コア・レジスタには、基本動作に直接影響するレジスタが含まれています。コア・レジスタは、すべてのデータ・メモリー・バンクの最初の12アドレスを占有します (アドレス x00h/x08hからx0Bh/x8Bh)。これらのレジスタを以下の表3-2に示します。詳細については、表3-8を参照してください。

表3-2: コア・レジスタ

アドレス	BANKx
x00h or x80h	INDF0
x01h or x81h	INDF1
x02h or x82h	PCL
x03h or x83h	STATUS
x04h or x84h	FSR0L
x05h or x85h	FSR0H
x06h or x86h	FSR1L
x07h or x87h	FSR1H
x08h or x88h	BSR
x09h or x89h	WREG
x0Ah or x8Ah	PCLATH
x0Bh or x8Bh	INTCON

## 3.3.1.1 STATUSレジスター

レジスター3-1に示すSTATUSレジスターには、次のものが含まれています：

- ALUの算術ステータス
- リセット状態

STATUSレジスターは、任意の命令の宛先にすることができます。他のレジスターと同様に、STATUSレジスターがZに影響する命令の宛先である場合、DCまたはCビット、その後、これらの3ビットへの書き込みは無効になります。これらのビットは、デバイスのロジックに従ってセットまたはクリアされます。さらに、TOビットとPDビットは書き込みできません。したがって、STATUSレジスターを宛先として指定した命令の結果は、意図したものとは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ビットを設定します。これにより、STATUSレジスターは「000u u1uu」のままになります（u=変更なし）。

それをお勧めします、したがって、その**BCF**だけ、**BSF**、**SWAPF**および**MOVWF**命令は、STATUSレジスターを変更するために使用されます。これらの命令はステータス・ビットに影響を与えないためです。ステータス・ビットに影響を与えないその他の命令については、セクション28.0「命令セットの概要」を参照してください。

**注1:** 減算では、CビットとDCビットはそれぞれBorrowビットとDigit Borrow outビットとして動作します。

### レジスター3-1: STATUS : ステータス・レジスター

U-0	U-0	U-0	R-1/q	R-1/q	R/W-0/u	R/W-0/u	R/W-0/u
—	—	—	TO	PD	Z	DC <sup>(1)</sup>	C <sup>(1)</sup>
bit 7						bit 0	

<b>記録:</b>		
R = 読み取り可能なビット	W = 書き込み可能なビット	U = 実装されていないビット、「0」として読み取られる
u = ビットは変更されません	x = ビットは不明です	-n/n = POR時の値およびBOR /他のすべてのリセット時の値
'1' = ビットはセットされています	'0' = ビットはクリアされています	q = 値は条件に依存します

- bit 7~5 **未実装:** 「0」として読み取り
- bit 4 **TO:** タイムアウト・ビット  
1 = 電源投入後、**CLRWDT**命令または**SLEEP**命令  
0 = WDTタイムアウトが発生しました
- bit 3 **PD:** パワーダウン・ビット  
1 = 電源投入後または**CLRWDT**命令による  
0 = **SLEEP**命令の実行による
- bit 2 **Z:** ゼロ・ビット  
1 = 算術演算または論理演算の結果がゼロです  
0 = 算術演算または論理演算の結果がゼロではありません
- bit 1 **DC:** Digit Carry/Digit Borrowビット (**ADDWF**、**ADDLW**、**SUBLW**、**SUBWF**命令) (1)  
1 = 結果の下位4ビットからの桁上がりが発生した  
0 = 結果の下位4番目のビットからのキャリーアウトなし
- bit 0 **C:** Carry/Borrowビット(1) (**ADDWF**、**ADDLW**、**SUBLW**、**SUBWF**命令) (1)  
1 = 結果の最上位ビットからのキャリーアウトが発生しました  
0 = 結果の最上位ビットからの繰り越しは発生しませんでした

**注1:** Borrowの場合、極性が逆になります。減算は、第2オペランドの2の補数を加算することによって実行されます。回転(**RRF**、**RLF**)命令の場合、このビットには、ソースレジスターの上位ビットまたは下位ビットが読み込まれます。

## 3.3.2 特殊機能レジスター

特殊機能レジスターは、デバイスの周辺機能に必要な動作を制御するためにアプリケーションによって使用されるレジスターです。特殊機能レジスターは、すべてのデータ・メモリー・バンクのコア・レジスター（アドレスx0Ch/x8Ch～x1Fh/x9Fh）の後に20バイトを占有します。周辺機器の操作に関連するレジスターは、このデータ・シートの適切な周辺機器の章に記載されています。

## 3.3.3 ゼネラル・パーパス(汎用)RAM

各データ・メモリー・バンクには最大80バイトのGPRがあります。特殊機能レジスターは、すべてのデータ・メモリー・バンクのコア・レジスター（アドレスx0Ch/x8Ch～x1Fh/x9Fh）の後に20バイトを占有します。

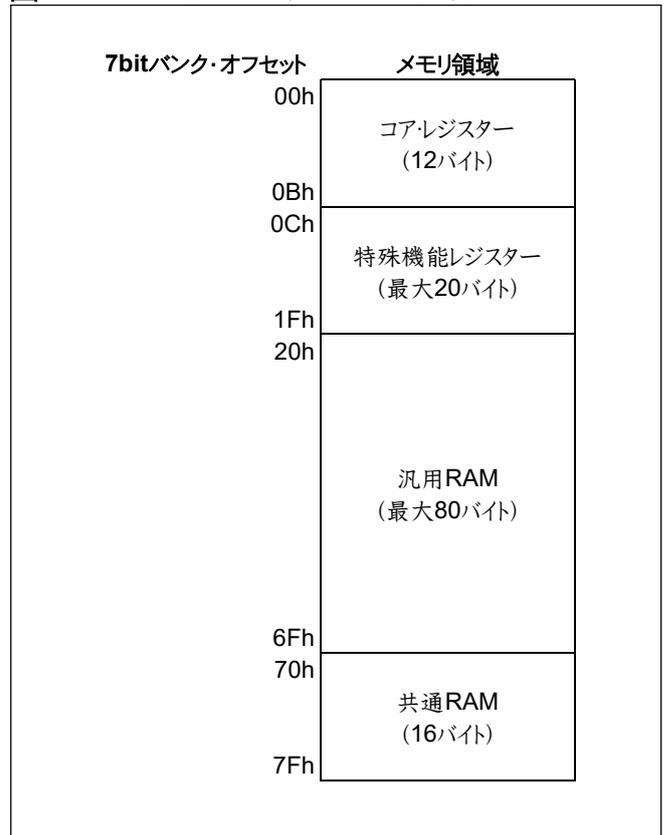
### 3.3.3.1 GPRへの線形アクセス

汎用RAMは、FSRを介して非バンク方式でアクセスできます。これにより、大規模なメモリ構造へのアクセスが簡素化されます。詳細はセクション3.6.2「リニア・データ・メモリー」を参照してください。

## 3.3.4 共通RAM

すべてのバンクからアクセス可能な16バイトの共通RAMがあります。

図3-2: バンク・メモリー・パーティション



### 3.3.5 デバイス・メモリー・マップ

バンク0からバンク31のメモリーマップは、このセクションの表に示されています。

表3-3: PIC16(L)F1508メモリーマップ、バンク0～7

バンク0		バンク1		バンク2		バンク3		バンク4		バンク5		バンク6		バンク7	
000h	コア・レジスター (Table 3-2)	080h	コア・レジスター (Table 3-2)	100h	コア・レジスター (Table 3-2)	180h	コア・レジスター (Table 3-2)	200h	コア・レジスター (Table 3-2+)	280h	コア・レジスター (Table 3-2)	300h	コア・レジスター (Table 3-2)	380h	コア・レジスター (Table 3-2)
00Bh		08Eh		10Bh		18Eh		20Bh		28Eh		30Bh		38Eh	
00Ch	PORTA	08Ch	TRISA	10Ch	LATA	18Ch	ANSELA	20Ch	WPUA	28Ch	—	30Ch	—	38Ch	—
00Dh	PORTB	08Dh	TRISB	10Dh	LATB	18Dh	ANSELB	20Dh	WPUB	28Dh	—	30Dh	—	38Dh	—
00Eh	PORTC	08Eh	TRISC	10Eh	LATC	18Eh	ANSELC	20Eh	—	28Eh	—	30Eh	—	38Eh	—
00Fh	—	08Fh	—	10Fh	—	18Fh	—	20Fh	—	28Fh	—	30Fh	—	38Fh	—
010h	—	090h	—	110h	—	190h	—	210h	—	290h	—	310h	—	390h	—
011h	PIR1	091h	PIE1	111h	CM1CON0	191h	PMADRL	211h	SSP1BUF	291h	—	311h	—	391h	IOCAP
012h	PIR2	092h	PIE2	112h	CM1CON1	192h	PMADRH	212h	SSP1ADD	292h	—	312h	—	392h	IOCAN
013h	PIR3	093h	PIE3	113h	CM2CON0	193h	PMDATL	213h	SSP1MSK	293h	—	313h	—	393h	IOCAF
014h	—	094h	—	114h	CM2CON1	194h	PMDATH	214h	SSP1STAT	294h	—	314h	—	394h	IOCBP
015h	TMR0	095h	OPTION_REG	115h	CMOUT	195h	PMCON1	215h	SSP1CON1	295h	—	315h	—	395h	IOCBN
016h	TMR1L	096h	PCON	116h	BORCON	196h	PMCON2	216h	SSP1CON2	296h	—	316h	—	396h	IOCBF
017h	TMR1H	097h	WDTCON	117h	FVRCON	197h	VREGCON	217h	SSP1CON3	297h	—	317h	—	397h	—
018h	T1CON	098h	—	118h	DAC1CON0	198h	—	218h	—	298h	—	318h	—	398h	—
019h	T1GCON	099h	OSCCON	119h	DAC1CON1	199h	RCREG	219h	—	299h	—	319h	—	399h	—
01Ah	TMR2	09Ah	OSCSTAT	11Ah	—	19Ah	TXREG	21Ah	—	29Ah	—	31Ah	—	39Ah	—
01Bh	PR2	09Bh	ADRESL	11Bh	—	19Bh	SPBRG	21Bh	—	29Bh	—	31Bh	—	39Bh	—
01Ch	T2CON	09Ch	ADRESH	11Ch	—	19Ch	SPBRGH	21Ch	—	29Ch	—	31Ch	—	39Ch	—
01Dh	—	09Dh	ADCON0	11Dh	APFCON	19Dh	RCSTA	21Dh	—	29Dh	—	31Dh	—	39Dh	—
01Eh	—	09Eh	ADCON1	11Eh	—	19Eh	TXSTA	21Eh	—	29Eh	—	31Eh	—	39Eh	—
01Fh	—	09Fh	ADCON2	11Fh	—	19Fh	BAUDCON	21Fh	—	29Fh	—	31Fh	—	39Fh	—
020h	汎用レジスター 80バイト	0A0h	汎用レジスター 80バイト	120h	汎用レジスター 80バイト	1A0h	未実装 「0」として読み 取る	220h	未実装 「0」として読み 取る	2A0h	未実装 「0」として読み 取る	320h	未実装 「0」として読み 取る	3A0h	未実装 「0」として読み 取る
06Fh		0EFh		16Fh		1EFh		26Fh		2EFh		36Fh		3EFh	
070h	共通RAM	0F0h	共通RAM (アクセス 70h ~ 7Fh)	170h	共通RAM (アクセス 70h ~ 7Fh)	1F0h	共通RAM (アクセス 70h ~ 7Fh)	270h	共通RAM (アクセス 70h ~ 7Fh)	2F0h	共通RAM (アクセス 70h ~ 7Fh)	370h	共通RAM (アクセス 70h ~ 7Fh)	3F0h	共通RAM (アクセス 70h ~ 7Fh)
07Fh		0FFh		17Fh		1FFh		27Fh		2FFh		37Fh		3FFh	

記録: ■ = 未実装のデータメモリー位置。「0」として読み取られます。

表3-4: PIC16(L)F1509メモリー・マップ、バンク0~7

バンク0		バンク1		バンク2		バンク3		バンク4		バンク5		バンク6		バンク7								
000h	コア・レジスター (Table 3-2)	080h	コア・レジスター (Table 3-2)	100h	コア・レジスター (Table 3-2)	180h	コア・レジスター (Table 3-2)	200h	コア・レジスター (Table 3-2+)	280h	コア・レジスター (Table 3-2)	300h	コア・レジスター (Table 3-2)	380h	コア・レジスター (Table 3-2)							
00Bh		08Eh		10Bh		18Eh		20Bh		28Eh		30Bh		38Eh								
00Ch	PORTA	08Ch	TRISA	10Ch	LATA	18Ch	ANSELA	20Ch	WPUA	28Ch	—	30Ch	—	38Ch	—							
00Dh	PORTB	08Dh	TRISB	10Dh	LATB	18Dh	ANSELB	20Dh	WPUB	28Dh	—	30Dh	—	38Dh	—							
00Eh	PORTC	08Eh	TRISC	10Eh	LATC	18Eh	ANSELC	20Eh	—	28Eh	—	30Eh	—	38Eh	—							
00Fh	—	08Fh	—	10Fh	—	18Fh	—	20Fh	—	28Fh	—	30Fh	—	38Fh	—							
010h	—	090h	—	110h	—	190h	—	210h	—	290h	—	310h	—	390h	—							
011h	PIR1	091h	PIE1	111h	CM1CON0	191h	PMADRL	211h	SSP1BUF	291h	—	311h	—	391h	IOCAP							
012h	PIR2	092h	PIE2	112h	CM1CON1	192h	PMADRH	212h	SSP1ADD	292h	—	312h	—	392h	IOCAN							
013h	PIR3	093h	PIE3	113h	CM2CON0	193h	PMDATL	213h	SSP1MSK	293h	—	313h	—	393h	IOCAF							
014h	—	094h	—	114h	CM2CON1	194h	PMDATH	214h	SSP1STAT	294h	—	314h	—	394h	IOCBP							
015h	TMR0	095h	OPTION_REG	115h	CMOUT	195h	PMCON1	215h	SSP1CON1	295h	—	315h	—	395h	IOCBN							
016h	TMR1L	096h	PCON	116h	BORCON	196h	PMCON2	216h	SSP1CON2	296h	—	316h	—	396h	IOCBF							
017h	TMR1H	097h	WDTCON	117h	FVRCON	197h	VREGCON	217h	SSP1CON3	297h	—	317h	—	397h	—							
018h	T1CON	098h	—	118h	DAC1CON0	198h	—	218h	—	298h	—	318h	—	398h	—							
019h	T1GCON	099h	OSCCON	119h	DAC1CON1	199h	RCREG	219h	—	299h	—	319h	—	399h	—							
01Ah	TMR2	09Ah	OSCSTAT	11Ah	—	19Ah	TXREG	21Ah	—	29Ah	—	31Ah	—	39Ah	—							
01Bh	PR2	09Bh	ADRESL	11Bh	—	19Bh	SPBRG	21Bh	—	29Bh	—	31Bh	—	39Bh	—							
01Ch	T2CON	09Ch	ADRESH	11Ch	—	19Ch	SPBRGH	21Ch	—	29Ch	—	31Ch	—	39Ch	—							
01Dh	—	09Dh	ADCON0	11Dh	APFCON	19Dh	RCSTA	21Dh	—	29Dh	—	31Dh	—	39Dh	—							
01Eh	—	09Eh	ADCON1	11Eh	—	19Eh	TXSTA	21Eh	—	29Eh	—	31Eh	—	39Eh	—							
01Fh	—	09Fh	ADCON2	11Fh	—	19Fh	BAUDCON	21Fh	—	29Fh	—	31Fh	—	39Fh	—							
020h	汎用レジスター 80バイト	0A0h	汎用レジスター 80バイト	120h	汎用レジスター 80バイト	1A0h	汎用レジスター 80バイト	220h	汎用レジスター 80バイト	2A0h	汎用レジスター 80バイト	320h	汎用レジスター 16バイト	3A0h	未実装 「0」として読み 取る							
06Fh		共通RAM		0EFh		共通RAM (アクセス 70h~7Fh)		16Fh		共通RAM (アクセス 70h~7Fh)		1EFh	共通RAM (アクセス 70h~7Fh)	26Fh		共通RAM (アクセス 70h~7Fh)	2EFh	共通RAM (アクセス 70h~7Fh)	36Fh	未実装「0」 として読み取る		
070h				共通RAM				170h				共通RAM		1F0h			共通RAM		270h	共通RAM	370h	共通RAM
07Fh				共通RAM				17Fh				共通RAM		1FFh			共通RAM		27Fh	共通RAM	37Fh	共通RAM

記録: ■ = 未実装のデータメモリー位置。「0」として読み取られます。

表3-5: PIC16(L)F1508/9メモリー・マップ、バンク8~23

バンク8		バンク9		バンク10		バンク11		バンク12		バンク13		バンク14		バンク15	
400h	コア・レジスター (Table 3-2)	480h	コア・レジスター (Table 3-2)	500h	コア・レジスター (Table 3-2)	580h	コア・レジスター (Table 3-2)	600h	コア・レジスター (Table 3-2+)	680h	コア・レジスター (Table 3-2)	700h	コア・レジスター (Table 3-2)	780h	コア・レジスター (Table 3-2)
40Bh	—	48Eh	—	50Bh	—	58Eh	—	60Bh	—	68Eh	—	70Bh	—	78Eh	—
40Ch	—	48Ch	—	50Ch	—	58Ch	—	60Ch	—	68Ch	—	70Ch	—	78Ch	—
40Dh	—	48Dh	—	50Dh	—	58Dh	—	60Dh	—	68Dh	—	70Dh	—	78Dh	—
40Eh	—	48Eh	—	50Eh	—	58Eh	—	60Eh	—	68Eh	—	70Eh	—	78Eh	—
40Fh	—	48Fh	—	50Fh	—	58Fh	—	60Fh	—	68Fh	—	70Fh	—	78Fh	—
410h	—	490h	—	510h	—	590h	—	610h	—	690h	—	710h	—	790h	—
411h	—	491h	—	511h	—	591h	—	611h	PWM1DCL	691h	CWG1DBR	711h	—	791h	—
412h	—	492h	—	512h	—	592h	—	612h	PWM1DCH	692h	CWG1DBF	712h	—	792h	—
413h	—	493h	—	513h	—	593h	—	613h	PWM1CON	693h	CWG1CON0	713h	—	793h	—
414h	—	494h	—	514h	—	594h	—	614h	PWM2DCL	694h	CWG1CON1	714h	—	794h	—
415h	—	495h	—	515h	—	595h	—	615h	PWM2DCH	695h	CWG1CON2	715h	—	795h	—
416h	—	496h	—	516h	—	596h	—	616h	PWM2CON	696h	—	716h	—	796h	—
417h	—	497h	—	517h	—	597h	—	617h	PWM3DCL	697h	—	717h	—	797h	—
418h	—	498h	NCO1ACCL	518h	—	598h	—	618h	PWM3DCH	698h	—	718h	—	798h	—
419h	—	499h	NCO1ACCH	519h	—	599h	—	619h	PWM3CON	699h	—	719h	—	799h	—
41Ah	—	49Ah	NCO1ACCU	51Ah	—	59Ah	—	61Ah	PWM4DCL	69Ah	—	71Ah	—	79Ah	—
41Bh	—	49Bh	NCO1INCL	51Bh	—	59Bh	—	61Bh	PWM4DCH	69Bh	—	71Bh	—	79Bh	—
41Ch	—	49Ch	NCO1INCH	51Ch	—	59Ch	—	61Ch	PWM4CON	69Ch	—	71Ch	—	79Ch	—
41Dh	—	49Dh	—	51Dh	—	59Dh	—	61Dh	—	69Dh	—	71Dh	—	79Dh	—
41Eh	—	49Eh	NCO1CON	51Eh	—	59Eh	—	61Eh	—	69Eh	—	71Eh	—	79Eh	—
41Fh	—	49Fh	NCO1CLK	51Fh	—	59Fh	—	61Fh	—	69Fh	—	71Fh	—	79Fh	—
420h	未実装 「0」として読み 取る	4A0h	未実装 「0」として読み 取る	520h	未実装 「0」として読み 取る	5A0h	未実装 「0」として読み 取る	620h	未実装 「0」として読み 取る	6A0h	未実装 「0」として読み 取る	720h	未実装 「0」として読み 取る	7A0h	未実装 「0」として読み 取る
470h	アクセス 70h~7Fh	4F0h	共通RAM (アクセス 70h~7Fh)	570h	共通RAM (アクセス 70h~7Fh)	5F0h	共通RAM (アクセス 70h~7Fh)	670h	共通RAM (アクセス 70h~7Fh)	6F0h	共通RAM (アクセス 70h~7Fh)	770h	共通RAM (アクセス 70h~7Fh)	7F0h	共通RAM (アクセス 70h~7Fh)
47Fh	—	4FFh	—	57Fh	—	5FFh	—	67Fh	—	6FFh	—	77Fh	—	7FFh	—
バンク16		バンク17		バンク18		バンク19		バンク20		バンク21		バンク22		バンク23	
800h	コア・レジスター (表3-2)	880h	コア・レジスター (表3-2)	900h	コア・レジスター (表3-2)	980h	コア・レジスター (表3-2)	A00h	コア・レジスター (表3-2+)	A80h	コア・レジスター (表3-2)	B00h	コア・レジスター (表3-2)	B80h	コア・レジスター (表3-2)
80Bh	—	88Eh	—	90Bh	—	98Eh	—	A0Bh	—	A8Eh	—	B0Bh	—	B8Eh	—
820h	未実装 「0」として読み 取る	8A0h	未実装 「0」として読み 取る	920h	未実装 「0」として読み 取る	9A0h	未実装 「0」として読み 取る	A20h	未実装 「0」として読み 取る	AA0h	未実装 「0」として読み 取る	B20h	未実装 「0」として読み 取る	BA0h	未実装 「0」として読み 取る
86Fh	アクセス 70h~7Fh	8EFh	アクセス 70h~7Fh	96Fh	アクセス 70h~7Fh	9EFh	アクセス 70h~7Fh	A6Fh	アクセス 70h~7Fh	AEFh	アクセス 70h~7Fh	B6Fh	アクセス 70h~7Fh	BEFh	アクセス 70h~7Fh
870h	—	8F0h	—	970h	—	9F0h	—	A70h	—	AF0h	—	B70h	—	BF0h	—
87Fh	—	8FFh	—	97Fh	—	9FFh	—	A7Fh	—	AFFh	—	B7Fh	—	BFFh	—

記録: ■ = 未実装のデータ・メモリー位置。「0」として読み取られます。

表3-6: PIC16(L)F1508/9メモリー・マップ、バンク24~31

バンク24		バンク25		バンク26		バンク27		バンク28		バンク29		バンク30		バンク31	
C00h	コア・レジスター (Table 3-2)	C80h	コア・レジスター (Table 3-2)	D00h	コア・レジスター (Table 3-2)	D80h	コア・レジスター (Table 3-2)	E00h	コア・レジスター (Table 3-2+)	E80h	コア・レジスター (Table 3-2)	F00h	コア・レジスター (Table 3-2)	F80h	コア・レジスター (Table 3-2)
C0Bh	—	C8Eh	—	D0Bh	—	D8Eh	—	E0Bh	—	E8Eh	—	F0Bh	レジスター・ マッピングの 詳細に ついては、 表3-7を 参照して ください	F8Eh	レジスター・ マッピングの 詳細に ついては、 表3-7を 参照して ください
C0Ch	—	C8Ch	—	D0Ch	—	D8Ch	—	E0Ch	—	E8Ch	—	F0Ch	—	F8Ch	—
C0Dh	—	C8Dh	—	D0Dh	—	D8Dh	—	E0Dh	—	E8Dh	—	F0Dh	—	F8Dh	—
C0Eh	—	C8Eh	—	D0Eh	—	D8Eh	—	E0Eh	—	E8Eh	—	F0Eh	—	F8Eh	—
C0Fh	—	C8Fh	—	D0Fh	—	D8Fh	—	E0Fh	—	E8Fh	—	F0Fh	—	F8Fh	—
C10h	—	C90h	—	D10h	—	D90h	—	E10h	—	E90h	—	F10h	—	F90h	—
C11h	—	C91h	—	D11h	—	D91h	—	E11h	—	E91h	—	F11h	—	F91h	—
C12h	—	C92h	—	D12h	—	D92h	—	E12h	—	E92h	—	F12h	—	F92h	—
C13h	—	C93h	—	D13h	—	D93h	—	E13h	—	E93h	—	F13h	—	F93h	—
C14h	—	C94h	—	D14h	—	D94h	—	E14h	—	E94h	—	F14h	—	F94h	—
C15h	—	C95h	—	D15h	—	D95h	—	E15h	—	E95h	—	F15h	—	F95h	—
C16h	—	C96h	—	D16h	—	D96h	—	E16h	—	E96h	—	F16h	—	F96h	—
C17h	—	C97h	—	D17h	—	D97h	—	E17h	—	E97h	—	F17h	—	F97h	—
C18h	—	C98h	—	D18h	—	D98h	—	E18h	—	E98h	—	F18h	—	F98h	—
C19h	—	C99h	—	D19h	—	D99h	—	E19h	—	E99h	—	F19h	—	F99h	—
C1Ah	—	C9Ah	—	D1Ah	—	D9Ah	—	E1Ah	—	E9Ah	—	F1Ah	—	F9Ah	—
C1Bh	—	C9Bh	—	D1Bh	—	D9Bh	—	E1Bh	—	E9Bh	—	F1Bh	—	F9Bh	—
C1Ch	—	C9Ch	—	D1Ch	—	D9Ch	—	E1Ch	—	E9Ch	—	F1Ch	—	F9Ch	—
C1Dh	—	C9Dh	—	D1Dh	—	D9Dh	—	E1Dh	—	E9Dh	—	F1Dh	—	F9Dh	—
C1Eh	—	C9Eh	—	D1Eh	—	D9Eh	—	E1Eh	—	E9Eh	—	F1Eh	—	F9Eh	—
C1Fh	—	C9Fh	—	D1Fh	—	D9Fh	—	E1Fh	—	E9Fh	—	F1Fh	—	F9Fh	—
C20h	未実装 「0」として 読み取る	CA0h	未実装 「0」として 読み取る	D20h	未実装 「0」として 読み取る	DA0h	未実装 「0」として 読み取る	E20h	未実装 「0」として 読み取る	EA0h	未実装 「0」として 読み取る	F20h	—	FA0h	—
C6Fh	—	CEFh	—	D6Fh	—	DEFh	—	E6Fh	—	EEFh	—	F6Fh	—	FEFh	—
C70h	アクセス 70h ~ 7Fh	CF0h	アクセス 70h ~ 7Fh	D70h	アクセス 70h ~ 7Fh	DF0h	アクセス 70h ~ 7Fh	E70h	アクセス 70h ~ 7Fh	EF0h	アクセス 70h ~ 7Fh	F70h	アクセス 70h ~ 7Fh	FF0h	アクセス 70h ~ 7Fh
C7Fh	—	CFFh	—	D7Fh	—	DFh	—	E7Fh	—	EFFh	—	F7Fh	—	FFFh	—

記録: ■ = 未実装のデータメモリー位置。「0」として読み取られます。

表3-7: PIC16(L)F1508/9メモリマップ、バンク30～31

バンク30		バンク31	
F0Ch	—	F8Ch	未実装 「0」として読み取る
F0Dh	—		
F0Eh	—		
F0Fh	CLCDATA	FE3h	
F10h	CLC1CON	FE4h	STATUS_SHAD
F11h	CLC1POL	FE5h	WREG_SHAD
F12h	CLC1SEL0	FE6h	BSR_SHAD
F13h	CLC1SEL1	FE7h	PCLATH_SHAD
F14h	CLC1GLS0	FE8h	FSR0L_SHAD
F15h	CLC1GLS1	FE9h	FSR0H_SHAD
F16h	CLC1GLS2	FEAh	FSR1L_SHAD
F17h	CLC1GLS3	FEBh	FSR1H_SHAD
F18h	CLC2CON	FECh	—
F19h	CLC2POL	FEDh	STKPTR
F1Ah	CLC2SEL0	FEEh	TOSL
F1Bh	CLC2SEL1	FEFh	TOSH
F1Ch	CLC2GLS0		
F1Dh	CLC2GLS1		
F1Eh	CLC2GLS2		
F1Fh	CLC2GLS3		
F20h	CLC3CON		
F21h	CLC3POL		
F22h	CLC3SEL0		
F23h	CLC3SEL1		
F24h	CLC3GLS0		
F25h	CLC3GLS1		
F26h	CLC3GLS2		
F27h	CLC3GLS3		
F28h	CLC4CON		
F29h	CLC4POL		
F2Ah	CLC4SEL0		
F2Bh	CLC4SEL1		
F2Ch	CLC4GLS0		
F2Dh	CLC4GLS1		
F2Eh	CLC4GLS2		
F2Fh	CLC4GLS3		
F30h	未実装 「0」として読み取る		
F6Fh			

記録: ■ =未実装のデータ・メモリーの場合。「0」として読み取られます。

# PIC16(L)F1508/9

## 3.2.6 コア機能レジスタの概要

表3-8にリストされているコア機能レジスタは、任意のバンクからアドレス指定できます。

表3-8: コア機能レジスタの概要

アドレス	名前	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR, BOR の値	他のすべての リセットの 値	
バンク0~31												
x00h or x80h	INDF0	この場所をアドレス指定すると、FSR0H/FSR0Lの内容を使用してデータ・メモリーをアドレス指定します (物理レジスタではありません)								xxxx xxxx	uuuu uuuu	
x01h or x81h	INDF1	この場所のアドレス指定では、FSR1H / FSR1Lの内容を使用してデータ・メモリーをアドレス指定します (物理レジスタではありません)								xxxx xxxx	uuuu uuuu	
x02h or x82h	PCL	プログラム・カウンタ (PC) 最下位バイト								0000 0000	0000 0000	
x03h or x83h	STATUS	—	—	—	TO	PD	Z	DC	C	---1 1000	---q quuu	
x04h or x84h	FSR0L	間接データ・メモリー・アドレス0の低ポインタ								0000 0000	uuuu uuuu	
x05h or x85h	FSR0H	間接データ・メモリー・アドレス0の高ポインタ								0000 0000	0000 0000	
x06h or x86h	FSR1L	間接データ・メモリー・アドレス1の低ポインタ								0000 0000	uuuu uuuu	
x07h or x87h	FSR1H	間接データ・メモリー・アドレス1の高ポインタ								0000 0000	0000 0000	
x08h or x88h	BSR	—	—	—	BSR<4:0>					---0 0000	---0 0000	
x09h or x89h	WREG	ワーキング・レジスタ								0000 0000	uuuu uuuu	
x0Ah or x8Ah	PCLATH	—	プログラム・カウンタの上位7ビット用の書き込みバッファ								-000 0000	-000 0000
x0Bh or x8Bh	INTCON	GIE	PEIE	TMR0IE	INTE	IOCFIE	TMR0IF	INTF	IOCFIF	0000 0000	0000 0000	

記録: x = 不明。u = 変更なし。q = 値は条件によって異なります。- = 未実装、「0」として読み取り。r = 予約済み。  
網掛けの場所は実装されていないため、「0」として読み取られます。

表3-9: 特殊機能レジスタの概要

アドレス	名前	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR, BOR の値	他のすべての のリセットの 値
バンク0											
00Ch	PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--xx xxxx	--xx xxxx
00Dh	PORTB	RB7	RB6	RB5	RB4	—	—	—	—	xxxx ----	xxxx ----
00Eh	PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	xxxx xxxx
010h	—	未実装								—	—
011h	PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	—	TMR2IF	TMR1IF	0000 0-00	0000 0-00
012h	PIR2	OSFIF	C2IF	C1IF	—	BCL1IF	NCO1IF	—	—	000- -00-	000- -00-
013h	PIR3	—	—	—	—	CLC4IF	CLC3IF	CLC2IF	CLC1IF	---- 0000	---- 0000
014h	—	未実装								—	—
015h	TMR0	8ビットTimer0カウントの保持レジスタ								xxxx xxxx	uuuu uuuu
016h	TMR1L	16ビットTMR1カウントの最下位バイトの保持レジスタ								xxxx xxxx	uuuu uuuu
017h	TMR1H	16ビットTMR1カウントの最上位バイトの保持レジスタ								xxxx xxxx	uuuu uuuu
018h	T1CON	TMR1CS<1:0>		T1CKPS<1:0>		T1OSCEN	T1SYNC	—	TMR1ON	0000 00-0	uuuu uu-u
019h	T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/ DONE	T1GVAL	T1GSS<1:0>		0000 0x00	uuuu uxuu
01Ah	TMR2	Timer2モジュール・レジスタ								0000 0000	0000 0000
01Bh	PR2	Timer2周期レジスタ								1111 1111	1111 1111
01Ch	T2CON	—	T2OUTPS<3:0>				TMR2ON	T2CKPS<1:0>		-000 0000	-000 0000
01Dh ~ 01Fh	—	未実装								—	—
バンク1											
08Ch	TRISA	—	—	TRISA5	TRISA4	— <sup>(2)</sup>	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111
08Dh	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	1111 ----	1111 ----
08Eh	TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	1111 1111
08Fh	—	未実装								—	—
090h	—	未実装								—	—
091h	PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	—	TMR2IE	TMR1IE	0000 0-00	0000 0-00
092h	PIE2	OSFIE	C2IE	C1IE	—	BCL1IE	NCO1IE	—	—	000- 00--	000- 00--
093h	PIE3	—	—	—	—	CLC4IE	CLC3IE	CLC2IE	CLC1IE	---- 0000	---- 0000
094h	—	未実装								—	—
095h	OPTION_ REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>		1111 111	1111 1111	
096h	PCON	STKOVF	STKUNF	—	RWDT	RMCLR	R $\bar{I}$	POR	BOR	00-1 11qq	qq-q qquu
097h	WDTCON	—	—	WDTPS<4:0>				SWDTEN	—	--01 0110	--01 0110
098h	—	未実装								—	—
099h	OSCCON	—	IRCF<3:0>				—	SCS<1:0>		-011 1-00	-011 1-00
09Ah	OSCSTAT	SOSCR	—	OSTS	HFIOFR	—	—	LFIOFR	HFIOFS	0-q0 --00	q-qq --qq
09Bh	ADRESL	ADC結果レジスタ下位								xxxx xxxx	uuuu uuuu
09Ch	ADRESH	ADC結果レジスタ上位								xxxx xxxx	uuuu uuuu
09Dh	ADCON0	—	CHS<4:0>				GO/DONE	ADON	—	-000 0000	-000 0000
09Eh	ADCON1	ADFM	ADCS<2:0>			—	—	ADPREF<1:0>		0000 --00	0000 --00
09Fh	ADCON2	TRIGSEL<3:0>				—	—	—	—	0000 ----	0000 ----

記録: x = 不明。u = 変更なし。q = 値は条件に依存。- = 未実装。r = 予約済み。網掛けの場所は実装されていないため、「0」として読み取られます。

注1: PIC16F1508/9のみ。

2: 未実装。「1」として読み取られます。

# PIC16(L)F1508/9

表3-9: 特殊機能レジスタの概要(続き)

アドレス	名前	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR, BOR の値	他のすべての リセットの 値	
バンク2												
10Ch	LATA	—	—	LATA5	LATA4	—	LATA2	LATA1	LATA0	--xx -xxx	--uu -uuu	
10Dh	LATB	LATB7	LATB6	LATB5	LATB4	—	—	—	—	xxxx ----	uuuu ----	
10Eh	LATC	LATC7	LATC6	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	xxxx xxxx	uuuu uuuu	
10Fh	—	未実装									—	—
110h	—	未実装									—	—
111h	CM1CON0	C1ON	C1OUT	C1OE	C1POL	—	C1SP	C1HYS	C1SYNC	0000 -100	0000 -100	
112h ~ 114h	—	未実装									—	—
115h	CMOUT	—	—	—	—	—	—	MC2OUT	MC1OUT	---- --00	---- --00	
116h	BORCON	SBOREN	BORFS	—	—	—	—	—	BORRDY	10-- ---q	uu-- ---u	
117h	FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		0q00 0000	0q00 0000	
118h	DAC1CON0	DACEN	—	DACOE1	DACOE2	—	DACPSS	—	—	0-00 -0--	0-00 -0--	
119h	DAC1CON1	—	—	—	DACR<4:0>				—	—	--0 0000	--0 0000
11Ah ~ 11Ch	—	未実装									—	—
11Dh	APFCON	—	—	—	SSSEL	T1GSEL	—	CLC1SEL	NC01SEL	---0 0-00	---0 0-00	
11Eh	—	未実装									—	—
11Fh	—	未実装									—	—

バンク3

18Ch	ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	---1 -111	---1 -111	
18Dh	ANSELB	—	—	ANSB5	ANSB4	—	—	—	—	--11 ----	--11 ----	
18Eh	ANSELC	ANSC7	ANSC6	—	—	ANSC3	ANSC2	ANSC1	ANSC0	11-- 1111	11-- 1111	
18Fh	—	未実装									—	—
190h	—	未実装									—	—
191h	PMADRL	フラッシュ・プログラム・メモリー・アドレス・レジスター下位バイト								0000 0000	0000 0000	
192h	PMADRH	— <sup>(2)</sup>	フラッシュ・プログラム・メモリー・アドレス・レジスター上位バイト								1000 0000	1000 0000
193h	PMDATL	フラッシュ・プログラム・メモリー読み取りデータ・レジスター下位バイト								xxxx xxxx	uuuu uuuu	
194h	PMDATH	—	—	フラッシュ・プログラム・メモリー読み取りデータ・レジスター上位バイト							--xx xxxx	--uu uuuu
195h	PMCON1	— <sup>(2)</sup>	CFG5	LWLO	FREE	WRERR	WREN	WR	RD	1000 x000	1000 q000	
196h	PMCON2	フラッシュ・プログラム・メモリー制御レジスター2								0000 0000	0000 0000	
197h	VREGCON <sup>(1)</sup>	—	—	—	—	—	—	VREGPM	予約済み	---- --01	---- --01	
198h	—	未実装									—	—
199h	RCREG	USART受信データ・レジスター								0000 0000	0000 0000	
19Ah	TXREG	USART送信データ・レジスター								0000 0000	0000 0000	
19Bh	SPBRGL	ボーレート・ジェネレーター・データ・レジスター下位								0000 0000	0000 0000	
19Ch	SPBRGH	ボーレート・ジェネレーター・データ・レジスター高位								0000 0000	0000 0000	
19Dh	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x	
19Eh	TXSTA	CSRC	TX9	TXEN	SYNC	SEnDB	BRGH	TRMT	TX9D	0000 0010	0000 0010	
19Fh	BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	01-0 0-00	

記録: x = 不明。u = 変更なし。q = 値は条件によって異なります。- = 未実装。r = 予約済み。網掛けの場所は実装されておらず、「0」と表示されます。

注1: PIC16F1508/9のみ。

2: 未実装、「1」と読みます。

表3-9: 特殊機能レジスタの概要(続き)

アドレス	名前	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR、BOR の値	他のすべての リセットの 値
バンク4											
20Ch	WPUA	—	—	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0	--11 1111	--11 1111
20Dh	WPUB	WPUB7	WPUB6	WPUB5	WPUB4	—	—	—	—	1111 ----	1111 ----
<del>E20Eh</del> 20Eh ~ 212h	—	未実装								—	—
213h	SSP1MSK	MSK<7:0>								1111 1111	1111 1111
214h	SSP1STAT	SMP	CKE	D/Ā	P	S	R/W	UA	BF	0000 0000	0000 0000
215h	SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>				0000 0000	0000 0000
216h	SSP1CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	0000 0000
217h	SSP1CON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	0000 0000	0000 0000
218h ~ 21Fh	—	未実装								—	—
バンク5											
28Ch ~ 29Fh	—	未実装								—	—
バンク6											
30Ch ~ 31Fh	—	未実装								—	—
バンク7											
38Ch ~ 390h	—	未実装								—	—
391h	IOCAP	—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	--00 0000	--00 0000
392h	IOCAN	—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	--00 0000	--00 0000
393h	IOCAF	—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	--00 0000	--00 0000
394h	IOCBP	IOCBP7	IOCBP6	IOCBP5	IOCBP4	—	—	—	—	0000 ----	0000 ----
395h	IOCBN	IOCBN7	IOCBN6	IOCBN5	IOCBN4	—	—	—	—	0000 ----	0000 ----
396h	IOCBF	IOCBF7	IOCBF6	IOCBF5	IOCBF4	—	—	—	—	0000 ----	0000 ----
397h ~ 39Fh	—	未実装								—	—
バンク8											
40Ch ~ 41Fh	—	未実装								—	—
バンク9											
48Ch ~ 497h	—	未実装								—	—
498h	NCO1ACCL	NCO1ACC<7:0>								0000 0000	0000 0000
499h	NCO1ACCH	NCO1ACC<15:8>								0000 0000	0000 0000
49Ah	NCO1ACCU	NCO1ACC<19:16>								0000 0000	0000 0000
49Bh	NCO1INCL	NCO1INC<7:0>								0000 0001	0000 0001
49Ch	NCO1INCH	NCO1INC<15:8>								0000 0000	0000 0000
49Dh	—	未実装								—	—
49Eh	NCO1CON	N1EN	N1OE	N1OUT	N1POL	—	—	—	N1PFM	0000 ---0	0000 ---0
49Fh	NCO1CLK	N1PWS<2:0>				—	—	—	N1CKS<1:0>	0000 --00	0000 --00

記録: x = 不明。u = 変更なし。q = 値は条件によって異なります。- = 未実装。r = 予約済み。網掛けの場所は実装されておらず、「0」と表示されます。

注1: PIC16F1508/9のみ。

2: 未実装、「1」と読みます。

# PIC16(L)F1508/9

表3-9: 特殊機能レジスタの概要(続き)

アドレス	名前	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR, BOR の値	他のすべての のリセットの 値
バンク10											
50Ch ~ 51Fh	—	未実装								—	—
バンク11											
58Ch ~ 59Fh	—	未実装								—	—
バンク12											
60Ch ~ 610h	—	未実装								—	—
611h	PWM1DCL	PWM1DCL<7:6>		—	—	—	—	—	—	00-- ----	00-- ----
612h	PWM1DCH	PWM1DCH<7:0>								xxxx xxxx	uuuu uuuu
613h	PWM1CON0	PWM1EN	PWM1OE	PWM1OUT	PWM1POL	—	—	—	—	0000 ----	0000 ----
614h	PWM2DCL	PWM2DCL<7:6>		—	—	—	—	—	—	00-- ----	00-- ----
615h	PWM2DCH	PWM2DCH<7:0>								xxxx xxxx	uuuu uuuu
616h	PWM2CON0	PWM2EN	PWM2OE	PWM2OUT	PWM2POL	—	—	—	—	0000 ----	0000 ----
617h	PWM3DCL	PWM3DCL<7:6>		—	—	—	—	—	—	00-- ----	00-- ----
618h	PWM3DCH	PWM3DCH<7:0>								xxxx xxxx	uuuu uuuu
619h	PWM3CON0	PWM3EN	PWM3OE	PWM3OUT	PWM3POL	—	—	—	—	0000 ----	0000 ----
61Ah	PWM4DCL	PWM4DCL<7:6>		—	—	—	—	—	—	00-- ----	00-- ----
61Bh	PWM4DCH	PWM4DCH<7:0>								xxxx xxxx	uuuu uuuu
61Ch	PWM4CON0	PWM4EN	PWM4OE	PWM4OUT	PWM4POL	—	—	—	—	0000 ----	0000 ----
61Dh ~ 61Fh	—	未実装								—	—
バンク13											
68Ch ~ 690h	—	未実装								—	—
691h	CWG1DBR	—	—	CWG1DBR<5:0>						--00 0000	--00 0000
692h	CWG1DBF	—	—	CWG1DBF<5:0>						--xx xxxx	--xx xxxx
693h	CWG1CON0	G1EN	G1OEB	G1OEA	G1POLB	G1POLA	—	—	G1CS0	0000 0--0	0000 0--0
694h	CWG1CON1	G1ASDLB<1:0>		G1ASDLA<1:0>		—	G1IS<2:0>			0000 -000	0000 -000
695h	CWG1CON2	G1ASE	G1ARSEN	—	—	G1ASDSC 2	G1ASDSC 1	G1ASDSF LT	G1ASDS CLC2	00-- 0000	00-- 0000
696h ~ 69Fh	—	未実装								—	—

記録: x =不明。u =変更なし。q =値は条件によって異なります。— =未実装。r =予約済み。網掛けの場所は実装されておらず、「0」と表示されます。

注1: PIC16F1508/9のみ。

2: 未実装、「1」と読みます。

表3-9: 特殊機能レジスタの概要(続き)

アドレス	名前	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR, BOR の値	他のすべての リセットの 値
バンク14~29											
x0Ch/ x8Ch ~ x1Fh/ x9Fh	—	未実装								—	—
バンク30											
F0Ch ~ F0Eh	—	未実装								—	—
F0Fh	CLCDATA	—	—	—	—	MLC4OUT	MLC3OUT	MLC2OUT	MLC1OUT	---- 0000	---- 0000
F10h	CLC1CON	LC1EN	LC1OE	LC1OUT	LC1INTP	LC1INTN	LC1MODE<2:0>			0000 0000	0000 0000
F11h	CLC1POL	LC1POL	—	—	—	LC1G4POL	LC1G3POL	LC1G2POL	LC1G1POL	0-- xxxx	0-- uuuu
F12h	CLC1SEL0	—	LC1D2S<2:0>			—	LC1D1S<2:0>			-xxx -xxx	-uuu -uuu
F13h	CLC1SEL1	—	LC1D4S<2:0>			—	LC1D3S<2:0>			-xxx -xxx	-uuu -uuu
F14h	CLC1GLS0	LC1G1D4T	LC1G1D4N	LC1G1D3T	LC1G1D3N	LC1G1D2T	LC1G1D2N	LC1G1D1T	LC1G1D1N	xxxx xxxx	uuuu uuuu
F15h	CLC1GLS1	LC1G2D4T	LC1G2D4N	LC1G2D3T	LC1G2D3N	LC1G2D2T	LC1G2D2N	LC1G2D1T	LC1G2D1N	xxxx xxxx	uuuu uuuu
F16h	CLC1GLS2	LC1G3D4T	LC1G3D4N	LC1G3D3T	LC1G3D3N	LC1G3D2T	LC1G3D2N	LC1G3D1T	LC1G3D1N	xxxx xxxx	uuuu uuuu
F17h	CLC1GLS3	LC1G4D4T	LC1G4D4N	LC1G4D3T	LC1G4D3N	LC1G4D2T	LC1G4D2N	LC1G4D1T	LC1G4D1N	xxxx xxxx	uuuu uuuu
F18h	CLC2CON	LC2EN	LC2OE	LC2OUT	LC2INTP	LC2INTN	LC2MODE<2:0>			0000 0000	0000 0000
F19h	CLC2POL	LC2POL	—	—	—	LC2G4POL	LC2G3POL	LC2G2POL	LC2G1POL	0-- xxxx	0-- uuuu
F1Ah	CLC2SEL0	—	LC2D2S<2:0>			—	LC2D1S<2:0>			-xxx -xxx	-uuu -uuu
F1Bh	CLC2SEL1	—	LC2D4S<2:0>			—	LC2D3S<2:0>			-xxx -xxx	-uuu -uuu
F1Ch	CLC2GLS0	LC2G1D4T	LC2G1D4N	LC2G1D3T	LC2G1D3N	LC2G1D2T	LC2G1D2N	LC2G1D1T	LC2G1D1N	xxxx xxxx	uuuu uuuu
F1Dh	CLC2GLS1	LC2G2D4T	LC2G2D4N	LC2G2D3T	LC2G2D3N	LC2G2D2T	LC2G2D2N	LC2G2D1T	LC2G2D1N	xxxx xxxx	uuuu uuuu
F1Eh	CLC2GLS2	LC2G3D4T	LC2G3D4N	LC2G3D3T	LC2G3D3N	LC2G3D2T	LC2G3D2N	LC2G3D1T	LC2G3D1N	xxxx xxxx	uuuu uuuu
F1Fh	CLC2GLS3	LC2G4D4T	LC2G4D4N	LC2G4D3T	LC2G4D3N	LC2G4D2T	LC2G4D2N	LC2G4D1T	LC2G4D1N	xxxx xxxx	uuuu uuuu
F20h	CLC3CON	LC3EN	LC3OE	LC3OUT	LC3INTP	LC3INTN	LC3MODE<2:0>			0000 0000	0000 0000
F21h	CLC3POL	LC3POL	—	—	—	LC3G4POL	LC3G3POL	LC3G2POL	LC3G1POL	0-- xxxx	0-- uuuu
F22h	CLC3SEL0	—	LC3D2S<2:0>			—	LC3D1S<2:0>			-xxx -xxx	-uuu -uuu
F23h	CLC3SEL1	—	LC3D4S<2:0>			—	LC3D3S<2:0>			-xxx -xxx	-uuu -uuu
F24h	CLC3GLS0	LC3G1D4T	LC3G1D4N	LC3G1D3T	LC3G1D3N	LC3G1D2T	LC3G1D2N	LC3G1D1T	LC3G1D1N	xxxx xxxx	uuuu uuuu
F25h	CLC3GLS1	LC3G2D4T	LC3G2D4N	LC3G2D3T	LC3G2D3N	LC3G2D2T	LC3G2D2N	LC3G2D1T	LC3G2D1N	xxxx xxxx	uuuu uuuu
F26h	CLC3GLS2	LC3G3D4T	LC3G3D4N	LC3G3D3T	LC3G3D3N	LC3G3D2T	LC3G3D2N	LC3G3D1T	LC3G3D1N	xxxx xxxx	uuuu uuuu
F27h	CLC3GLS3	LC3G4D4T	LC3G4D4N	LC3G4D3T	LC3G4D3N	LC3G4D2T	LC3G4D2N	LC3G4D1T	LC3G4D1N	xxxx xxxx	uuuu uuuu
F28h	CLC4CON	LC4EN	LC4OE	LC4OUT	LC4INTP	LC4INTN	LC4MODE<2:0>			0000 0000	0000 0000
F29h	CLC4POL	LC4POL	—	—	—	LC4G4POL	LC4G3POL	LC4G2POL	LC4G1POL	0-- xxxx	0-- uuuu
F2Ah	CLC4SEL0	—	LC4D2S<2:0>			—	LC4D1S<2:0>			-xxx -xxx	-uuu -uuu
F2Bh	CLC4SEL1	—	LC4D4S<2:0>			—	LC4D3S<2:0>			-xxx -xxx	-uuu -uuu
F2Ch	CLC4GLS0	LC4G1D4T	LC4G1D4N	LC4G1D3T	LC4G1D3N	LC4G1D2T	LC4G1D2N	LC4G1D1T	LC4G1D1N	xxxx xxxx	uuuu uuuu
F2Dh	CLC4GLS1	LC4G2D4T	LC4G2D4N	LC4G2D3T	LC4G2D3N	LC4G2D2T	LC4G2D2N	LC4G2D1T	LC4G2D1N	xxxx xxxx	uuuu uuuu
F2Eh	CLC4GLS2	LC4G3D4T	LC4G3D4N	LC4G3D3T	LC4G3D3N	LC4G3D2T	LC4G3D2N	LC4G3D1T	LC4G3D1N	xxxx xxxx	uuuu uuuu
F2Fh	CLC4GLS3	LC4G4D4T	LC4G4D4N	LC4G4D3T	LC4G4D3N	LC4G4D2T	LC4G4D2N	LC4G4D1T	LC4G4D1N	xxxx xxxx	uuuu uuuu
F30h ~ F6Fh	—	未実装								—	—

記録: x = 不明。u = 変更なし。q = 値は条件によって異なります。— = 未実装。r = 予約済み。網掛けの場所は実装されておらず、「0」と表示されます。

注1: PIC16F1508/9のみ。

2: 未実装、「1」と読みます。

# PIC16(L)F1508/9

表3-9: 特殊機能レジスタの概要(続き)

アドレス	名前	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR、BOR の値	他のすべての リセットの 値	
バンク31												
F8Ch ~ FE3h	—	未実装								—	—	
FE4h	STATUS_SHAD	—	—	—	—	—	Z_SHAD	DC_SHAD	C_SHAD	---- -xxx	---- -uuu	
FE5h	WREG_SHAD	ワーキング・レジスター・シャドウ								xxxx xxxx	uuuu uuuu	
FE6h	BSR_SHAD	—	—	—	バンク・セレクト・レジスター・シャドウ					---x xxxx	---u uuuu	
FE7h	PCLATH_SHAD	—	プログラム・カウンタ・ラッチ上位レジスター・シャドウ								-xxx xxxx	uuuu uuuu
FE8h	FSR0L_SHAD	間接データ・メモリー・アドレス0下位ポインタ・シャドウ								xxxx xxxx	uuuu uuuu	
FE9h	FSR0H_SHAD	間接データ・メモリー・アドレス0上位ポインタ・シャドウ								xxxx xxxx	uuuu uuuu	
FEAh	FSR1L_SHAD	間接データ・メモリー・アドレス1下位ポインタ・シャドウ								xxxx xxxx	uuuu uuuu	
FEBh	FSR1H_SHAD	間接データ・メモリー・アドレス1上位ポインタ・シャドウ								xxxx xxxx	uuuu uuuu	
FECh	—	未実装								—	—	
FEDh	STKPTR	—	—	—	現在のスタック・ポインタ					---1 1111	---1 1111	
FEEh	TOSL	トップのスタック下位バイト								xxxx xxxx	uuuu uuuu	
FEFh	TOSH	—	トップのスタック上位バイト								-xxx xxxx	-uuu uuuu

記録: x =不明。u =変更なし。q =値は条件によって異なります。— =未実装。r =予約済み。網掛けの場所は実装されておらず、「0」と表示されます。

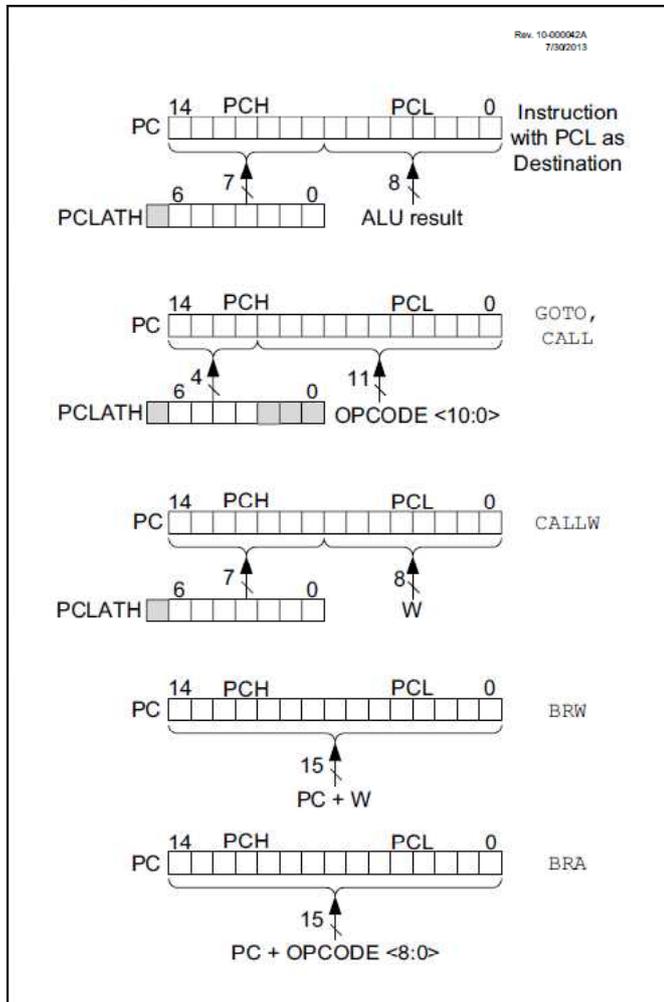
注1: PIC16F1508/9のみ。

2: 未実装、「1」と読みます。

### 3.3 PCLおよびPCLATH

プログラム・カウンタ（PC）は15ビット幅です。下位バイトはPCLレジスタから来ます。これは読み取りと書き込みが可能なレジスタです。上位バイト（PC<14:8>）は直接読み取りまたは書き込み可能ではなく、PCLATHから取得されます。リセットすると、PCはクリアされます。図3-3は、PCをロードするための5つの状況を示しています。

図3-4: 異なる状況でのPCのロード



#### 3.3.1 PCLの変更

PCLレジスタを受け側として同時に命令を実行すると、プログラム・カウンタPC<14:8>ビット（PCH）がPCLATHレジスタの内容に置き換えられます。これにより、必要な上位7ビットをPCLATHレジスタに書き込むことにより、プログラム・カウンタの内容全体を変更できます。下位8ビットがPCLレジスタに書き込まれると、プログラム・カウンタの15ビットすべてがPCLATHレジスタに含まれている値とPCLレジスタに書き込まれている値に変更されます。

#### 3.3.2 計算されたGOTO

計算されたGOTOは、プログラム・カウンタ（ADDWF PCL）にオフセットを追加することによって実現されます。計算されたGOTOメソッドを使用してテーブルの読み取りを実行する場合、テーブルの場所がPCLメモリ境界（各256バイト・ブロック）を超える場合は注意が必要です。アプリケーションノートAN556を参照してください。「テーブル読み取りの実装」（DS00556）。

#### 3.3.3 計算された関数呼び出し

計算関数CALLを使用すると、プログラムは関数のテーブルを維持し、スタート・マシンまたはルック・アップ・テーブルを実行する別の方法を提供できます。計算関数CALLを使用してテーブルの読み取りを実行する場合、テーブルの場所がPCLメモリ境界（各256バイト・ブロック）を超える場合は注意が必要です。

CALL命令を使用する場合、PCH<2:0>およびPCLレジスタにはCALL命令のオペランドがロードされます。PCH<6:3>にはPCLATH<6:3>がロードされています。

CALLW命令は、PCLATHとWを組み合わせて宛先アドレスを形成することにより、計算された呼び出しを可能にします。計算されたCALLWは、Wレジスタに目的のアドレスをロードし、CALLWを実行することによって実行されます。PCLレジスタにはWの値がロードされ、PCHにはPCLATHがロードされます。

#### 3.3.4 分岐

分岐命令はPCにオフセットを追加します。これにより、再配置可能なコードとページの境界を超えるコードが可能になります。分岐には2つの形式があります。BRWとBRA。どちらの場合も、PCは次の命令をフェッチするためにインクリメントされます。いずれかの分岐命令を使用する場合、PCLメモリの境界を超える可能性があります。

BRWを使用する場合、Wレジスタに目的の符号なしアドレスをロードし、BRWを実行します。PC全体にアドレスPC + 1 + Wがロードされます。

BRAを使用している場合、PC全体にPC + 1 + がロードされ、BRA命令のオペランドの符号付きの値。



図3-6: スタックへのアクセス例2

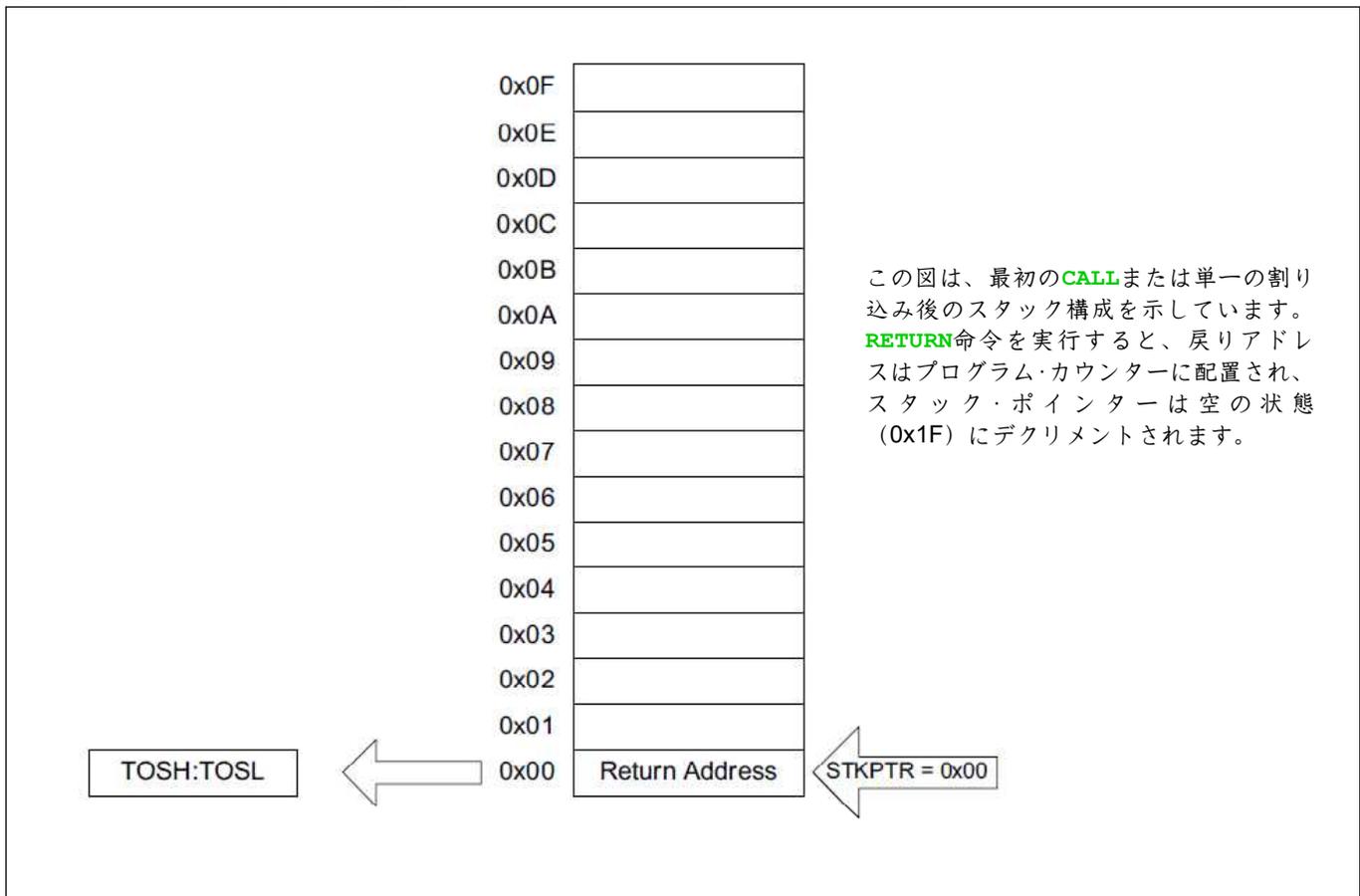


図3-7: スタックへのアクセス例3

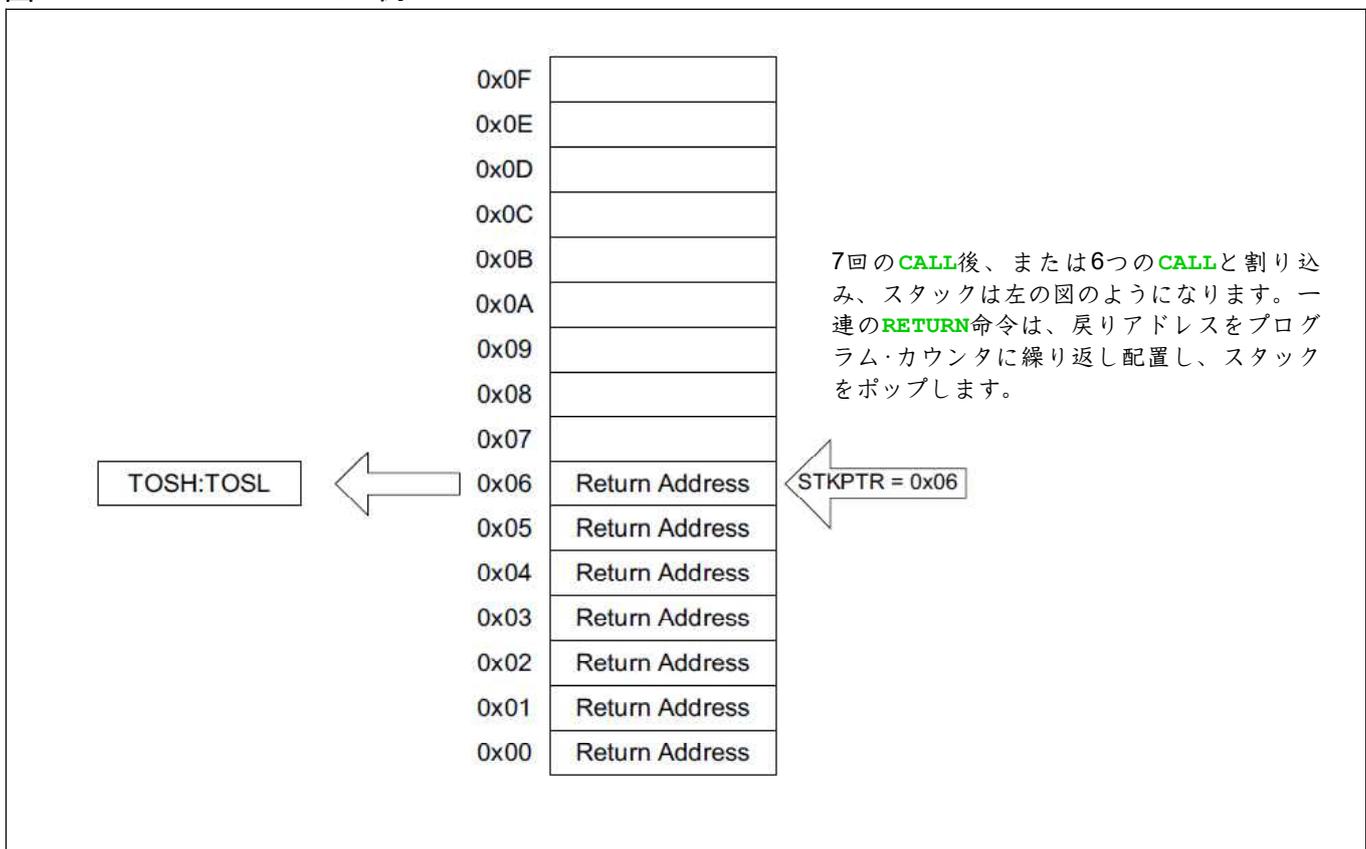
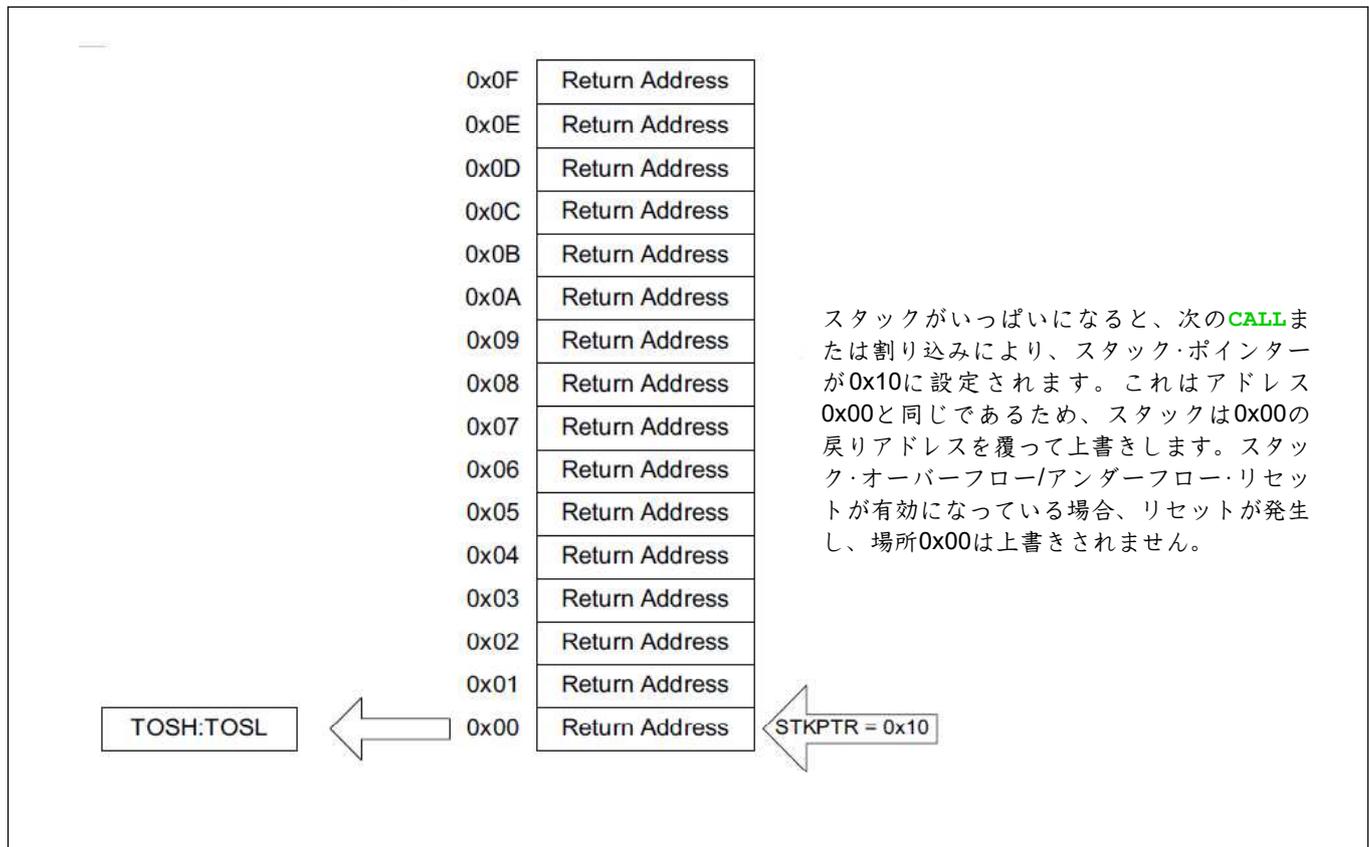


図3-8: スタックへのアクセス例4



### 3.4.2 オーバーフロー/アンダーフロー・リセット

構成ワードのSTVRENビットが「1」にプログラムされている場合、スタックが16レベルを超えてプッシュされるか、1レベルを超えてPOPされると、デバイスはリセットされます。PCONレジスターに適切なビット（それぞれSTKOVFまたはSTKUNF）を設定します。

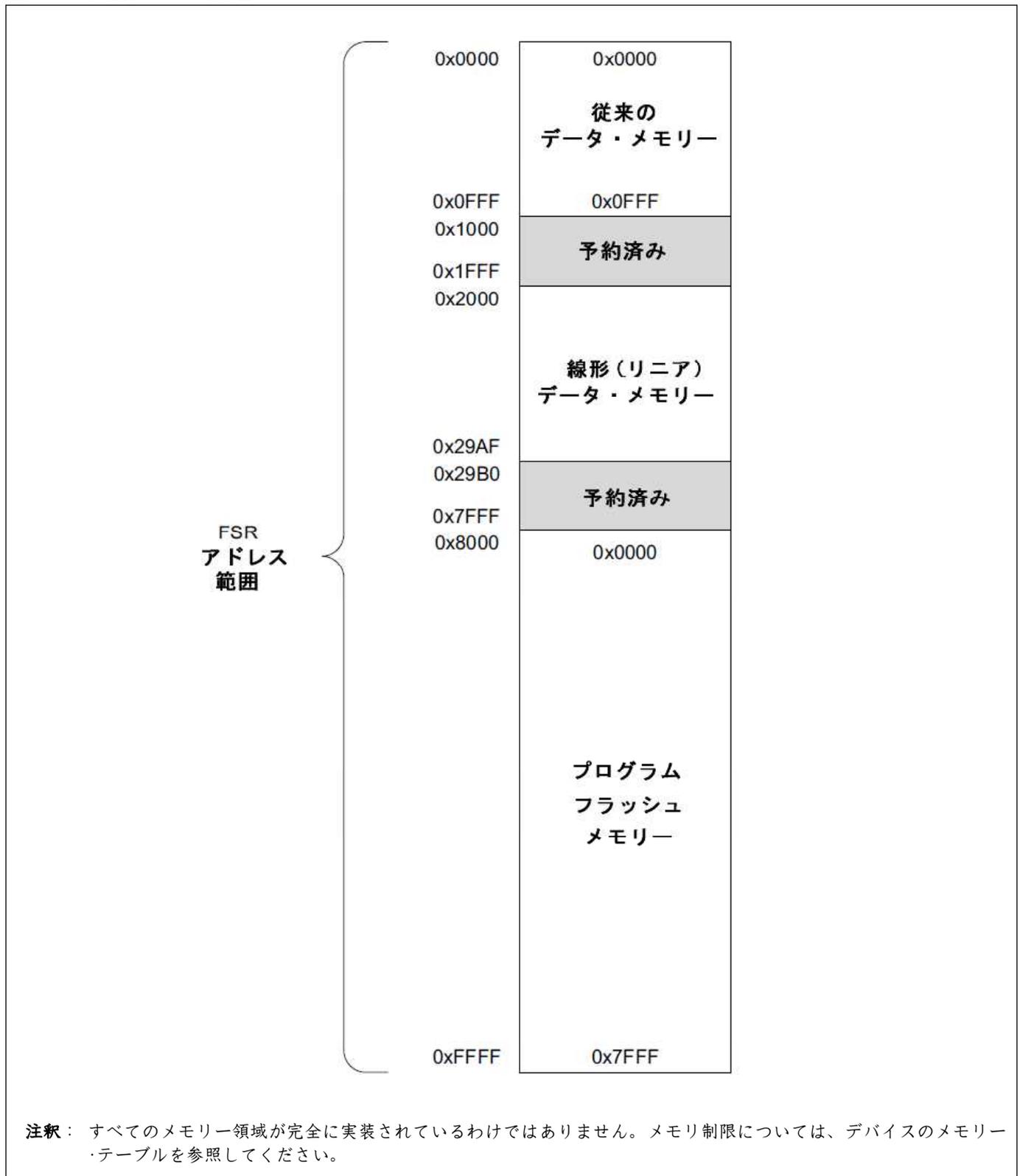
### 3.5 間接アドレス指定

INDFnレジスターは物理レジスターではありません。INDFnレジスターにアクセスする命令は、実際にはファイル選択レジスター（FSR）で指定されたアドレスのレジスターにアクセスします。FSRnアドレスが2つのINDFnレジスターの1つを指定している場合、読み取りは「0」を返し、書き込みは発生しません（ただしステータスビット影響を受ける可能性があります）。FSRnレジスター値は、FSRnHとFSRnLのペアによって作成されます。

FSRレジスターは16ビット・アドレスを形成し、65536個のロケーションを持つアドレス空間を可能にします。これらの場所は、次の3つのメモリー領域に分割されます：

- 従来のデータ・メモリー
- 線形データ・メモリー
- プログラム・フラッシュ・メモリー

図3-9: 間接アドレス指定



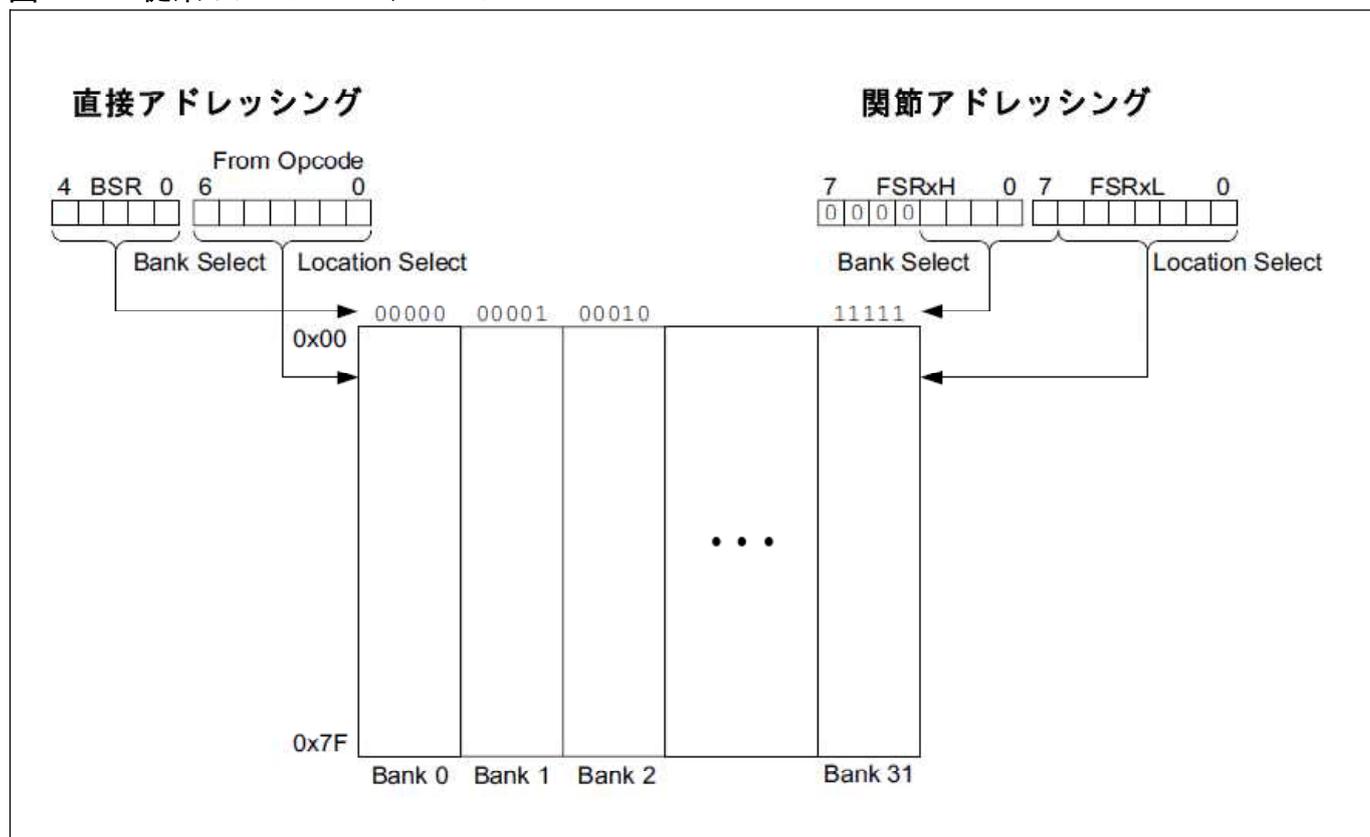
注釈: すべてのメモリー領域が完全に実装されているわけではありません。メモリー制限については、デバイスのメモリー・テーブルを参照してください。

# PIC16(L)F1508/9

## 3.5.1 従来のデータ・メモリー

従来のデータ・メモリーは、FSRアドレス0x000からFSRアドレス0xFFFまでの領域です。アドレスは、すべてのSFR、GPR、および共通レジスタの絶対アドレスに対応します。

図3-10: 従来のデータ・メモリー・マップ



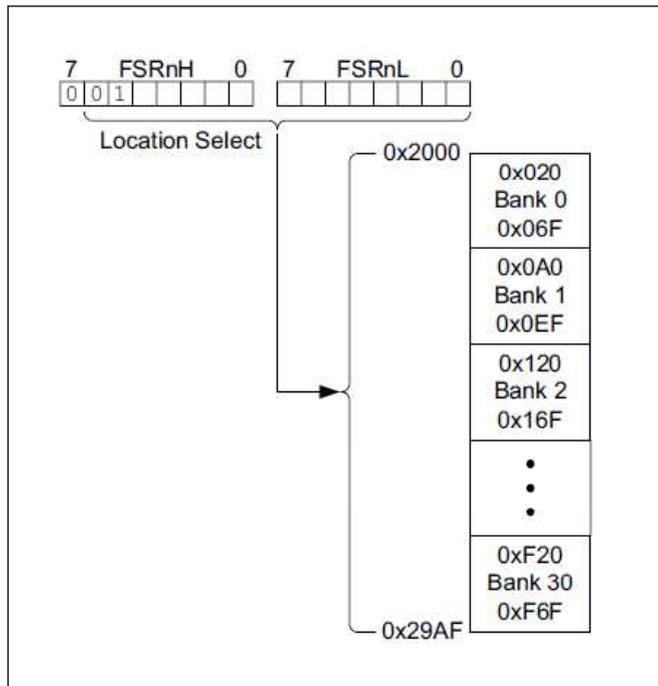
3.5.2 線形(リニア)データ・メモリー

リニア・データ・メモリーは、FSRアドレス0x2000からFSRアドレス0x29AFまでの領域です。この領域は、すべてのバンクのGPRメモリの80バイト・ブロックを指す仮想領域です。

実装されていないメモリーは0x00として読み取られます。線形データ・メモリー領域を使用すると、バッファを大きくすることができます1つのバンクを超えてFSRをインクリメントすると、次のバンクのGPRメモリーに直接移動するため、80バイトを超えます。

16バイトの共通メモリーは線形データ・メモリー領域に含まれていません。

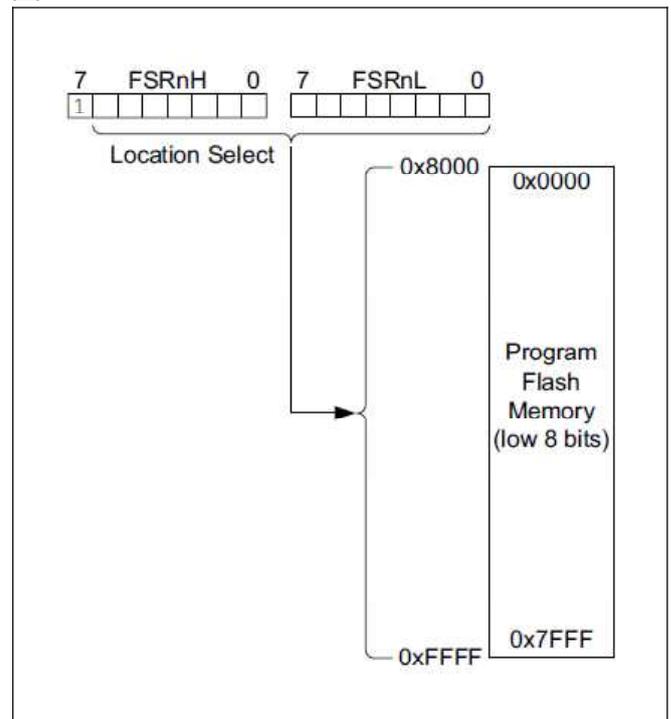
図3-11: リニア・データ・メモリー・マップ



3.5.3 プログラム・フラッシュ・メモリー

常時データ・アクセスを容易にするために、プログラム全体のフラッシュ・メモリーは、FSRアドレス空間の上半分にマップされます。FSRnHのMSbが設定されている場合、下位15ビットは、INDFを介してアクセスされるプログラム・メモリー内のアドレスです。各メモリー位置の下位8ビットのみがINDFを介してアクセスできます。プログラム・フラッシュ・メモリーへの書き込みは、FSR/INDFインターフェイスを介して実行することはできません。FSR/INDFインターフェイスを介してプログラム・フラッシュ・メモリーにアクセスするすべての命令は、完了するために1つの追加の命令サイクルを必要とします。

図3-12: プログラム・フラッシュ・メモリー・マップ



## 4.0 デバイス構成

デバイス構成は、構成ワード、コード保護、およびデバイスIDで構成されます。

### 4.1 構成ワード

さまざまな発信器およびメモリー保護オプションを可能にするいくつかの構成ワード・ビットがあります。これらは、8007hの構成ワード1および8008hの構成ワード2として実装されます。

**注釈：** Configuration Wordの $\overline{\text{DEBUG}}$ ビットは、デバッガーやプログラマーなどのデバイス開発ツールによって自動的に管理されます。通常のデバイス操作では、このビットを「1」として維持する必要があります。

## 4.2 レジスタ定義: 構成ワード

### レジスタ4-1: CONFIG1: 構成ワード1

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	U-1
FCMEN <sup>(1)</sup>	IESO <sup>(1)</sup>	CLKOUTEN	BOREN<1:0> <sup>(2)</sup>	—	
bit 13					bit 8

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
CP <sup>(3)</sup>	MCLRE	PWRTE	WDTE<1:0>	FOSC<2:0>			
bit 7							bit 0

#### 記録:

R = 読み取り可能なビット      P = プログラム可能なビット      U = 未実装ビット。「1」として読み取られます  
 「0」 = ビットがクリアされます      「1」 = ビットが設定されています      -n = 空白または一括消去後の値

- bit 13      **FCMEN**: フェイルセーフ・クロック・モニター・イネーブル・ビット  
 1 = フェイルセーフ・クロック・モニターが有効になっている (1)  
 0 = フェイルセーフ・クロック・モニターが無効になっている
- bit 12      **IESO**: 内部外部スイッチ・オーバー・ビット (1)  
 1 = 内部/外部スイッチ・オーバー (2速起動) モードが有効になっている  
 0 = 内部/外部スイッチ・オーバー・モードは無効です
- bit 11      **CLKOUTEN**: クロック出カイネーブル・ビット  
 1 = CLKOUT機能は無効です。CLKOUTピンのI/O機能  
 0 = CLKOUT機能はCLKOUTピンで有効になります
- bit 10~9    **BOREN<1:0>**: ブラウンアウト・リセット・イネーブル・ビット (2)  
 11 = BORが有効  
 10 = BORは動作中に有効になり、スリープ時に無効になります  
 01 = BORCONレジスタのSBORENビットによって制御されるBOR  
 00 = BORが無効
- bit 8      **未実装**: 「1」として読み取ります
- bit 7      **CP**: コード保護ビット (3)  
 1 = プログラム・メモリー・コード保護が無効になっている  
 0 = プログラム・メモリー・コード保護が有効になっている
- bit 6      **MCLRE**: MCLR/VPPピン機能選択ビット  
**LVPビット = 1の場合**:  
 このビットは無視されます。  
**LVPビット = 0の場合**:  
 1 = MCLR/VPPピン機能はMCLRです。弱いプルアップが有効になっています。  
 0 = MCLR/VPPピン機能はデジタル入力です。MCLRは内部的に無効になっています。WPUA3ビットの制御下での弱いプルアップ。
- bit 5      **PWRTE**: パワーアップ・タイマー・イネーブル・ビット  
 1 = PWRTが無効  
 0 = PWRTが有効
- bit 4~3    **WDTE<1:0>**: ウォッチドッグ・タイマー・イネーブル・ビット  
 11 = WDTが有効  
 10 = WDTが有効で、実行中はスリープは使用禁止  
 01 = WDTはWDTCONレジスタのSWDTENビットによって制御される  
 00 = WDTが無効

# PIC16(L)F1508/9

---

## レジスタ 4-1: CONFIG1: 構成ワード1(続き)

bit 2~0 **FOSC<2:0>**: 発信器選択ビット

111 = ECH: 外部クロック、高電力モード: CLKINピン上

110 = ECM: 外部クロック、中電力モード: CLKINピン上

101 = ECL: 外部クロック、低電力モード: CLKINピン上

100 = INTOSCオシレータ: CLKINピンのI/O機能

011 = EXTRCオシレータ: CLKINピンに接続された外部RC回路

010 = HS発振器: OSC1ピンとOSC2ピンの間に接続された高速水晶/共振器

001 = XT発振器: OSC1ピンとOSC2ピンの間に接続された水晶/共振器

000 = LP発振器: OSC1ピンとOSC2ピンの間に接続された低電力水晶

- 注1: FSCMを有効にすると、IESOのビット値に関係なく、2速起動が自動的に有効になります。
- 2: ブラウンアウト・リセットを有効にしても、パワーアップ・タイマーは自動的に有効になりません。
- 3: 一度有効にすると、コード保護はデバイスを一括消去することによってのみ無効にできます。

## レジスタ-4-2: CONFIG2: 構成ワード2

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	U-1
LVP(1)	DEBUG(3)	LPBOR	BORV(2)	STVREN	—
bit 13					bit 8

U-1	U-1	U-1	U-1	U-1	U-1	R/P-1	R/P-1
—	—	—	—	—	—	WRT<1:0>	
bit 7						bit 0	

### 記録:

R = 読み取り可能なビット      P = プログラム可能なビット      U = 未実装ビット。「1」として読み取られます  
 「0」 = ビットがクリアされます      「1」 = ビットが設定されています      -n = 空白または一括消去後の値

- bit 13      **LVP**: 低電圧プログラミング許可ビット (1)  
 1 = 低電圧プログラミングが有効  
 0 = MCLRの高電圧をプログラミングに使用する必要があります
- bit 12      **DEBUG**: 回路内デバッグ・モード・ビット (3)  
 1 = 回路内デバッガが無効、ICSPCLKおよびICSPDATは汎用I/Oピンです  
 0 = インサーキット・デバッガが有効で、ICSPCLKとICSPDATはデバッガ専用です
- bit 11      **LPBOR**: 低電力BORイネーブル・ビット  
 1 = 低電力ブラウンアウト・リセットが無効になっている  
 0 = 低電力ブラウンアウト・リセットが有効になっている
- bit 10      **BORV**: ブラウンアウト・リセット電圧選択ビット (2)  
 1 = ブラウンアウト・リセット電圧 (VBOR)、低トリップ・ポイントが選択されています  
 0 = ブラウンアウト・リセット電圧 (VBOR)、選択された高トリップ・ポイント
- bit 9      **STVREN**: スタック・オーバーフロー/アンダーフロー・リセット・イネーブル・ビット  
 1 = スタック・オーバー・フローまたはアンダーフローはリセットを引き起こします  
 0 = スタック・オーバー・フローまたはアンダーフローはリセットを引き起こしません
- bit 8~2      **未実装**: 「1」として読み取ります
- bit 1~0      **WRT<1:0>**: フラッシュ・メモリー自己書き込み保護ビット  
4kWフラッシュ・メモリー (PIC16(L)F1508/9のみ)  
 11 = 書き込み保護をオフ  
 10 = 000h~1FFhの書き込み保護、200h~FFFhは変更可能  
 01 = 000h~7FFhの書き込み保護、800h~FFFhは変更可能  
 00 = 000h~FFFhの書き込み保護、アドレスは変更できません  
8kWフラッシュメモリ (PIC16(L)F1509のみ)  
 11 = 書き込み保護をオフ  
 10 = 0000h~01FFhの書き込み保護、0200h~1FFFhは変更可能  
 01 = 0000h~0FFFhの書き込み保護、1000h~1FFFhは変更可能  
 00 = 0000h~1FFFhの書き込み保護、アドレスは変更できません

- 注1: LVP経由でプログラミング・モードに入った場合、LVPビットを「0」にプログラムすることはできません。  
 2: 特定のトリップ・ポイント電圧については、VBORパラメーターを参照してください。  
 3: 構成ワードのDEBUGビットは、デバッガやプログラマーなどのデバイス開発ツールによって自動的に管理されます。通常のデバイス操作では、このビットを「1」として維持する必要があります。

## 4.3 コード保護

コード保護により、デバイスを不正アクセスから保護できます。プログラム・メモリーへの内部アクセスは、コード保護設定の影響を受けません。

### 4.3.1 プログラム・メモリー保護

プログラム・メモリー空間全体は、構成ワードの $\overline{CP}$ ビットによって外部の読み取りおよび書き込みから保護されます。 $\overline{CP} = 0$ の場合、プログラム・メモリーの外部読み取りおよび書き込みは禁止されており、読み取りはすべて「0」を返します。CPUはプログラム・メモリーの読み取りを続行できます。保護ビットの設定に関係なく、プログラム・メモリーへの書き込みは、書き込み保護の設定に依存します。詳細については、セクション4.4「書き込み保護」を参照してください。

## 4.4 書き込み保護

書き込み保護により、デバイスを意図しない自己書き込みから保護できます。アプリケーション、ブートローダー・ソフトウェアなど、プログラム・メモリーの他の領域を変更できるようにしながら保護することができます。構成ワードの $WRT<1:0>$ ビットは、保護されるプログラム・メモリー・ブロックのサイズを定義します。

## 4.5 ユーザーID

4つのメモリー位置（8000h～8003h）は、ユーザーがチェックサムまたはその他のコード識別番号を格納できるID位置として指定されます。これらの場所は、通常の実行中に読み取りおよび書き込みが可能です。セクション10.4「ユーザーID、これらのメモリー位置へのアクセスの詳細については、「デバイスIDと構成ワード・アクセス」を参照してください。チェックサム計算の詳細については、「PIC12(L)F1501/PIC16(L)F150Xメモリー・プログラミング仕様」（DS41573）を参照してください。

#### 4.6 デバイスIDとリビジョンID

メモリー位置8006hは、デバイスIDとリビジョン（改訂）IDが格納される場所です。上位9ビットはデバイスIDを保持します。下位5ビットはリビジョンIDを保持します。これらのメモリー位置へのアクセスの詳細については、セクション10.4「ユーザーID、デバイスID、および構成ワード・アクセス」を参照してください。

開発ツール、デバイス・プログラマーやデバッガーなど、デバイスIDとリビジョンIDの読み取りに使用できます。

#### 4.7 レジスター定義: デバイスID

レジスター 4-3: DEVID: デバイスIDレジスター

R	R	R	R	R	R
DEV<8:3>					
bit 13			bit 8		

R	R	R	R	R	R	R	R
DEV<2:0>			REV<4:0>				
bit 7							bit 0

**記録:**

R = 読み取り可能なビット

「1」 = ビットが設定されています 「0」 = ビットがクリアされます

bit 13~5 **DEV<8:0>**: デバイスIDビット

デバイス	DEVID<13:0> 値	
	DEV<8:0>	REV<4:0>
PIC16LF1508	10 1101 111	x xxxx
PIC16F1508	10 1101 001	x xxxx
PIC16LF1509	10 1110 000	x xxxx
PIC16F1509	10 1101 010	x xxxx

bit 4~0 **REV<4:0>**: リビジョンIDビット

これらのビットは、リビジョンを識別するために使用されます（上記のDEV<8:0>の表を参照）。

## 5.0 発信器モジュール(フェイルセーフ・クロック・モニター付き)

### 5.1 概要

オシレーター(発信器)モジュールには、パフォーマンスを最大化し、消費電力を最小化しながら、幅広いアプリケーションで使用できるようにするさまざまなクロック・ソースと選択機能があります。図5-1に、発振器モジュールのブロック図を示します。

クロック・ソースは外部発振器から供給できます。水晶振動子、セラミック共振子と抵抗-コンデンサー(RC)回路。加えて、システムクロック・ソースは、2つの内部発信器のいずれかから供給できます。ソフトウェアを介して選択可能な速度の選択で。追加のクロック機能は次のとおりです:

- ソフトウェアを介して外部ソースまたは内部ソースの間で選択可能なシステム・クロック・ソース。
- 2速スタートアップ・モード。外部発信器の起動とコード実行の間の遅延を最小限に抑えます。
- 外部クロック・ソース(LP、XT、HS、ECH、ECM、ECL、またはEXTRCモード)の障害を検出し、自動的に内部発振器に切り替えるように設計されたフェイルセーフ・クロック・モニター(FSCM)。
- オシレーター・スタートアップ・タイマー(OST)は、水晶発振器ソースの安定性を保証します
- 高速起動発振器により、16 MHz HFINTOSCに切り替える前に、内部回路の電源を入れて安定させることができます。

オシレーター・モジュールは、次のクロック・モードのいずれかで設定できます。

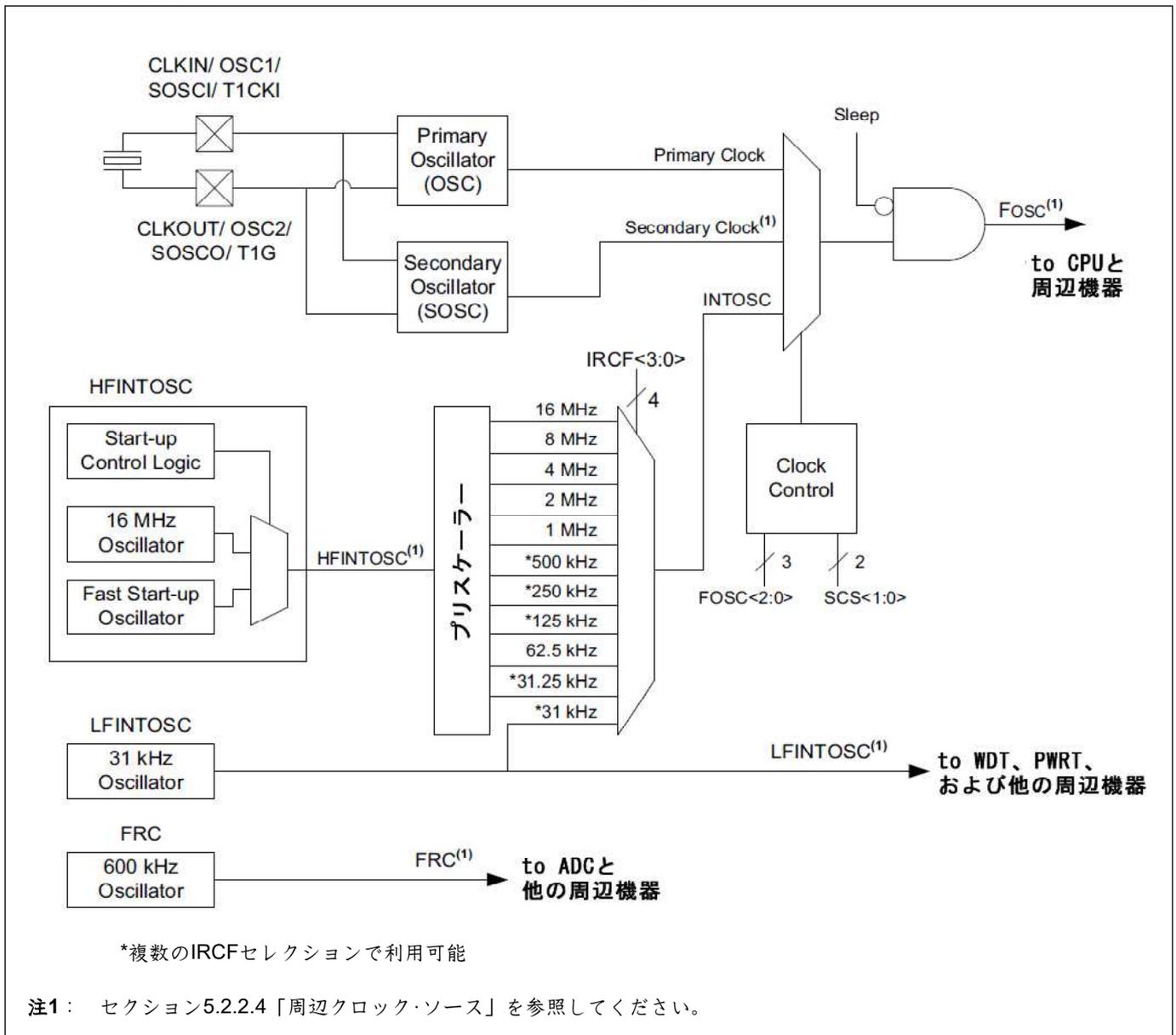
1. ECL—外部クロック低電力モード(0 MHz~0.5 MHz)
2. ECM—外部クロック中電力モード(0.5 MHz~4 MHz)
3. ECH—外部クロック高電力モード(4 MHz~20 MHz)
4. LP—32kHz低電力水晶モード。
5. XT—中ゲイン(利得)水晶またはセラミック共振子発振器モード(最大4 MHz)
6. HS—高ゲイン水晶またはセラミック共振子モード(4 MHz~20 MHz)
7. EXTRC—外部抵抗-コンデンサー
8. INTOSC—内部発振器(31 kHz~16 MHz)

クロック・ソース・モードは、構成ワードのFOSC<2:0>ビットによって選択されます。FOSCビットは、デバイスに最初に電力が供給されたときに使用される発振器のタイプを決定します。

ECH、ECM、およびECLクロック・モードは、デバイス・クロック・ソースとして外部ロジック・レベル信号に依存しています。LP、XT、およびHSクロック・モードでは、外部クリスタルまたは共振器をデバイスに接続する必要があります。各モードは、異なる周波数範囲に最適化されています。EXTRCクロック・モードでは、発振器周波数を設定するために外部抵抗とコンデンサーが必要です。

INTOSC内部発振器ブロックは、低周波数および高周波数のクロック・ソースを生成します。LFINTOSCおよびHFINTOSCと指定されています(内部発振器ブロック、図5-1を参照)。これらの2つのクロック・ソースから、デバイスのクロック周波数を幅広く選択できます。

図5-1: 簡略化されたPIC®MCUクロック・ソース・ブロック図



## 5.2 クロック・ソース・タイプ

クロック・ソースは、外部、内部、または周辺機器に分類できます。

外部クロック・ソースは、クロック・ソースが機能するために外部回路に依存しています。例：発信器モジュール（ECH、ECM、ECLモード）、水晶振動子またはセラミック共振子（LP、XT、HSモード）および抵抗 - コンデンサー（EXTRC）モード回路。

内部クロック・ソースは、発信器モジュール内に含まれています。内部発信器ブロックには、内部システム・クロック・ソースを生成するために使用される2つの内部発信器があります。16 MHz高周波内部発信器（HFINTOSC）と31 kHz低周波内部発信器（LFINTOSC）です。

周辺クロック・ソースは、公称600 kHzの内部RC発振器、FRCです。FRCは伝統的にADCモジュールで使用されます。ただし、他の周辺機器で利用できる場合もあります。セクション5.2.2.4「周辺クロック・ソース」を参照してください。

システム・クロックは、OSCCONレジスタのシステム・クロック選択（SCS）ビットを介して外部または内部クロック・ソースから選択できます。詳細については、セクション5.3「クロック切り替え」を参照してください。

### 5.2.1 外部クロック・ソース

次のいずれかのアクションを実行することにより、外部クロック・ソースをデバイス・システム・クロックとして使用できます：

- 構成ワードのFOSC<2:0>ビットをプログラムして、デバイスのリセット時にデフォルトのシステム・クロックとして使用される外部クロック・ソースを選択します。
- OSCCONレジスタにSCS<1:0>ビットを書き込んで、システム・クロック・ソースを次のように切り替えます。
  - 実行時の2次発信器、または
  - FOSCビットの値によって決定される外部クロック・ソース。

詳細については、セクション5.3「クロック切り替え」を参照してください。

#### 5.2.1.1 ECモード

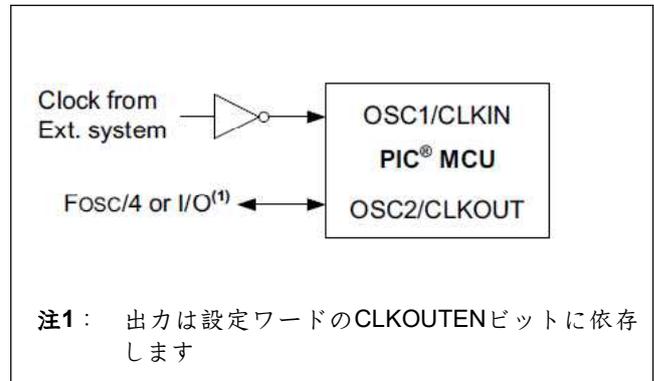
外部クロック（EC）モードでは、外部で生成されたロジック・レベル信号をシステム・クロック・ソースにすることができます。このモードで動作する場合、外部クロック・ソースがOSC1入力に接続されます。OSC2/CLKOUTは、汎用I/OまたはCLKOUTに使用できます。図5-2に、ECモードのピン接続を示します。

ECモードには、構成ワードのFOSCビットを介して選択できる3つの電源モードがあります。

- ECH—高電力。4 ~ 20 MHz
- ECM—中電力。0.5 ~ 4 MHz
- ECL—低電力。0 ~ 0.5 MHz

ECモードを選択すると、オシレーター・スタートアップ・タイマー（OST）が無効になります。したがって、パワー・オン・リセット（POR）またはスリープからの目覚め後の動作に遅延はありません。なぜならPIC®MCUの設計は完全に静的であり、外部クロック入力を停止すると、すべてのデータをそのままにしてデバイスを停止する効果があります。外部クロックを再起動すると、デバイスは、時間が経過していないかのように動作を再開します。

図5-2: 外部クロック(EC)モードの操作



#### 5.2.1.2 LP、XT、HSモード

LP、XT、およびHSモードは、OSC1およびOSC2に接続された水晶振動子またはセラミック共振子の使用をサポートします（図5-3）。3つのモードは、さまざまな共振器タイプと速度をサポートするために、内部インバーター・アンプの低、中、高ゲイン設定を選択します。

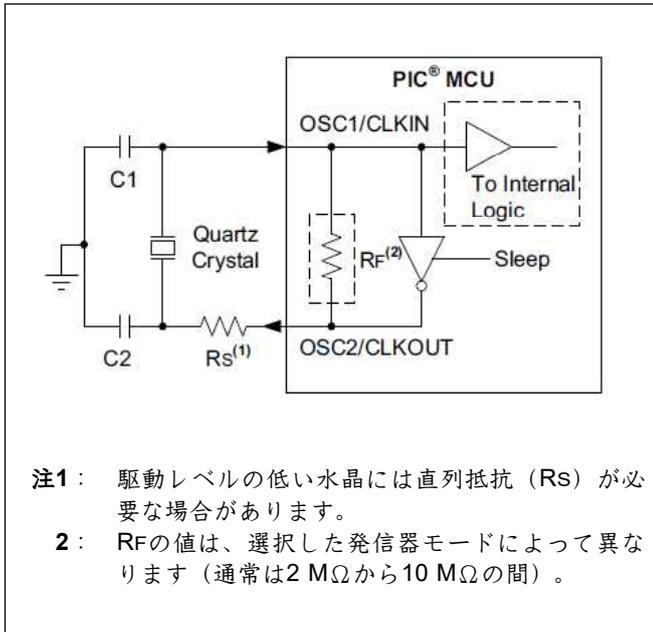
**LP**発信器モードは、内部インバーター・アンプの最低ゲイン設定を選択します。LPモードの消費電流は、3つのモードの中で最小です。このモードは、32.768 kHzの音叉タイプの水晶（時計の水晶）のみを駆動するように設計されています。

**XT**発信器モードは、内部インバーター・アンプの中間ゲイン設定を選択します。XTモードの消費電流は、3つのモードの中間です。このモードは、中程度のドライブ・レベル仕様のドライブ共振器に最適です。

**HS**発信器モードは、内部インバーター・アンプの最高ゲイン設定を選択します。HSモードの消費電流は、3つのモードの中で最も高くなっています。このモードは、高いドライブ設定を必要とする共振器に最適です。

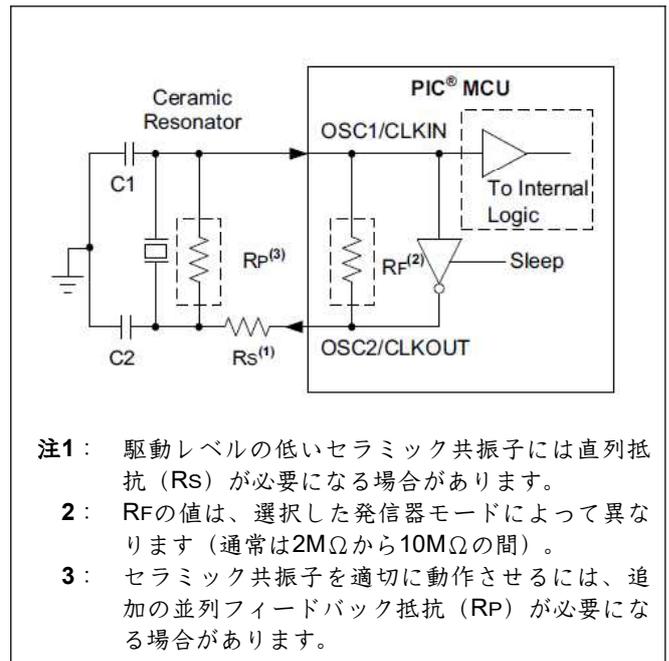
図5-3と図5-4に、それぞれ水晶振動子とセラミック共振子の代表的な回路を示します。

図5-3: クォーツ・クリスタルの操作 (LP、XT、またはHSモード)



- 注1: 駆動レベルの低い水晶には直列抵抗 (Rs) が必要な場合があります。
- 2: Rfの値は、選択した発信器モードによって異なります (通常は2 MΩから10 MΩの間)。

図5-4: セラミック共振子の動作 (XTまたはHSモード)



- 注1: 駆動レベルの低いセラミック共振子には直列抵抗 (Rs) が必要になる場合があります。
- 2: Rfの値は、選択した発信器モードによって異なります (通常は2MΩから10MΩの間)。
- 3: セラミック共振子を適切に動作させるには、追加の並列フィードバック抵抗 (Rp) が必要になる場合があります。

注1: 水晶の特性は、種類、パッケージ、メーカーによって異なります。仕様および推奨されるアプリケーションについては、メーカーのデータシートを参照してください。

- 2: アプリケーションに期待されるVDDおよび温度範囲での発振器性能を常に確認してください。
- 3: 発振器の設計支援については、次のマイクロチップ・アプリケーション・ノートを参照してください。
- AN826、「rfPIC®およびPIC®デバイス用の水晶発振器の基本と水晶の選択」(DS00826)
  - AN849、「基本的なPIC®発振器の設計」(DS00849)
  - AN943、「実用的なPIC®発振器の分析と設計」(DS00943)
  - AN949、「オシレータを機能させる」(DS00949)

### 5.2.1.3 オシレーター・スタートアップ・タイマー (OST)

発信器モジュールがLP、XT、またはHSモード用に構成されている場合、オシレーター・スタートアップ・タイマー (OST) はOSC1からの1024回の発振をカウントします。これは、パワーオン・リセット (POR) の後で、パワーアップ・タイマー (PWRT) の期限が切れたとき (構成されている場合)、またはスリープからのウェイクアップ時に発生します。この間、FSCMまたは2速起動のいずれかが有効になっていない限り、プログラム・カウンタは増加せず、プログラムの実行は一時停止されます。この場合、OSTがカウントしている間、コードは選択したINTOSC周波数で実行され続けます。OSTは、水晶振動子またはセラミック共振子を使用した発振回路が起動し、発振器モジュールに安定したシステム・クロックを提供していることを確認します。

外部発信器の起動とコード実行の間の遅延を最小限に抑えるために、2速クロック起動モードを選択できます (セクション5.4「2速クロック起動モード」を参照)。

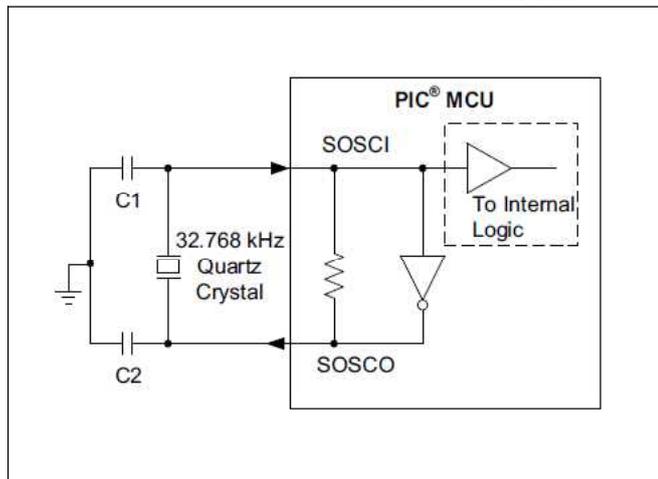
# PIC16(L)F1508/9

## 5.2.1.4 セカンダリ(2次)発振器

2次発振器は、タイマー1周辺機器に関連付けられている別の水晶発振器です。これは、SOSCOデバイス・ピンとSOSCIデバイス・ピンの間に接続された32.768 kHz水晶を使用した計時操作用に最適化されています。

2次発振器は、代替システム・クロック・ソースとして使用でき、クロック・スイッチングを使用して実行時に選択できます。詳細については、セクション5.3「クロック切り替え」を参照してください。

図5-5: クォーツ・クリスタルの動作(2次発振器)



- 注1:** 水晶の特性は、種類、パッケージ、メーカーによって異なります。仕様および推奨されるアプリケーションについては、メーカーのデータシートを参照してください。
- 2:** アプリケーションに期待されるVDDおよび温度範囲での発振器性能を常に確認してください。
- 3:** 発振器の設計支援については、次のマイクロチップ・アプリケーション・ノートを参照してください:
- AN826、「rfPIC®およびPIC®デバイス用の水晶発振器の基本と水晶の選択」(DS00826)
  - AN849、「基本的なPIC®発振器の設計」(DS00849)
  - AN943、「実用的なPIC®発振器の分析と設計」(DS00943)
  - AN949、「オシレーターを機能させる」(DS00949)
  - TB097、「マイクロ・クリスタルのインターフェースMS1V-T1K 32.768kHz音叉水晶をPIC16F690/SSに調整」(DS91097)
  - AN1288、「低電力外部発振器の設計手法」(DS01288)

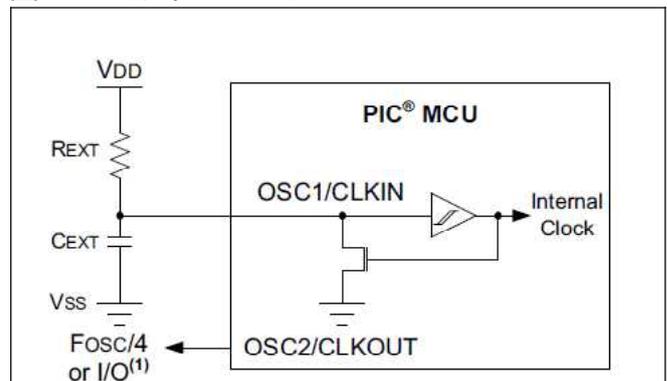
## 5.2.1.5 外部RCモード

外部抵抗-コンデンサー (EXTRC) モードは、外部RC回路の使用をサポートします。これにより、設計者は、クロックの精度が必要ない場合にコストを最小限に抑えながら、周波数の選択に最大限の柔軟性を持たせることができます。

RC回路はOSC1に接続します。OSC2/CLKOUTは、汎用I/OまたはCLKOUTに使用できます。OSC2/CLKOUTピンの機能は、構成ワードのCLKOUTENビットによって決定されます。

図5-6に、外部RCモードの接続を示します。

図5-6: 外部RCモード



- 推奨値:  $10\text{k}\Omega \leq R_{EXT} \leq 100\text{k}\Omega$ 、 $< 3\text{V}$   
 $3\text{k}\Omega \leq R_{EXT} \leq 100\text{k}\Omega$ 、 $3 \sim 5\text{V}$   
 $C_{EXT} > 20\text{pF}$ 、 $2 \sim 5\text{V}$

**注1:** 出力は設定ワードのCLKOUTENビットに依存します。

RC発振器の周波数は、電源電圧の関数です。抵抗 (REXT) とコンデンサー (CEXT) の値と動作温度。発振器周波数に影響を与える他の要因は次のとおりです:

- しきい値電圧変動
- コンポーネントの公差
- 静電容量のパッケージ変動

ユーザーは、使用される外部RCコンポーネントの許容誤差による変動も考慮する必要があります。

## 5.2.2 内部クロック・ソース

デバイスは、次のいずれかのアクションを実行することにより、内部発振器ブロックをシステム・クロックとして使用するように構成できます：

- 構成ワードのFOSC<2:0>ビットをプログラムして、デバイスのリセット時にデフォルトのシステム・クロックとして使用されるINTOSCクロック・ソースを選択します。
- OSCCONレジスターにSCS<1:0>ビットを書き込んで、実行時にシステム・クロック・ソースを内部発振器に切り替えます。詳細については、セクション5.3「クロック切り替え」を参照してください。

INTOSCモードでは、OSC1/CLKINを汎用I/Oに使用できます。OSC2/CLKOUTは、汎用I/OまたはCLKOUTに使用できます。

OSC2/CLKOUTピンの機能は、構成ワードのCLKOUTENビットによって決定されます。

内部発信器ブロックには、内部システム・クロック・ソースを提供する2つの独立したオシレーターがあります。

1. **HFINTOSC** (高周波内部発振器) は工場で校正されており、16 MHzで動作します。
2. **LFINTOSC** (低周波内部発振器) は31 kHzで動作します。

### 5.2.2.1 HFINTOSC

ハイレケンシー・インターナル・オシレーター (HFINTOSC) は、工場で校正された16 MHzの内部クロック・ソースです。

HFINTOSCの出力は、ポストスケラーとマルチプレクサーに接続します (図5-1を参照)。HFINTOSCから得られる周波数は、OSCCONレジスターのIRCF<3:0>ビットを使用してソフトウェアで選択できます。詳細については、セクション5.2.2.6「内部発信器クロック・スイッチのタイミング」を参照してください。

HFINTOSCは、次の方法で有効になります：

- OSCCONレジスターのIRCF<3:0>ビットを目的の高周波数に設定し、そして
- FOSC<2:0> = 100、または
- OSCCONレジスターのシステム・クロック・ソース (SCS) ビットを「1x」に設定します。

高速起動発振器により、HFINTOSCに切り替える前に内部回路の電源を入れて安定させることができます。

OSCSTATレジスターの高周波内部発信器レディ・ビット (HFIOFR) は、HFINTOSCが実行されていることを示します。

OSCSTATレジスターの高周波内部発信器安定ビット (HFIOFS) は、HFINTOSCが最終値の0.5%以内で実行されていることを示します。

### 5.2.2.2 LFINTOSC

低周波内部発振器 (LFINTOSC) は、31kHzの内部クロック・ソースです。

LFINTOSCの出力は、マルチプレクサーに接続します (図5-1を参照)。31kHzを選択し、ソフトウェアを介して、OSCCONレジスターのIRCF<3:0>ビットを使用します。詳細については、セクション5.2.2.6「内部発信器クロック・スイッチのタイミング」を参照してください。LFINTOSCは、パワーアップ・タイマー (PWRT)、ウォッチドッグ・タイマー (WDT)、およびフェイルセーフ・クロック・モニター (FSCM) の周波数でもあります。

LFINTOSCは、システム・クロック・ソース (OSCCONレジスターのSCSビット = 1x) として31 kHz (OSCCONレジスターのIRCF<3:0>ビット = 000) を選択するか、次のいずれかが有効になっている場合に有効になります：

- OSCCONレジスターのIRCF<3:0>ビットを目的の低周波数に設定します。そして
- FOSC<2:0> = 100、または
- OSCCONレジスターのシステム・クロック・ソース (SCS) ビットを「1x」に設定します。

LFINTOSCを使用する周辺機器は次のとおりです：

- パワーアップ・タイマー (PWRT)
- ウォッチドッグ・タイマー (WDT)
- フェイルセーフ・クロック・モニター (FSCM)

OSCSTATレジスターの低周波数内部発信器レディ・ビット (LFIOFR) は、LFINTOSCが実行されていることを示します。

### 5.2.2.3 FRC

FRCクロックは、調整されていない公称600 kHzの周辺クロック・ソースです。

FRCは、FRCクロックを要求する周辺機器によって自動的にオンになります。

FRCクロックは、スリープ中も動作し続けます。

## 5.2.2.4 周辺クロック・ソース

この章で説明するクロック・ソースとタイマーは、さまざまな周辺機器で使用できます。表5-1に、各周辺で使用可能なクロックとタイマーを示します。

表5-1: 周辺クロック・ソース

	FOSC	FRC	HFINTOSC	LFINTOSC	TMR0	TMR1	TMR2	SOSC
ADC	●	●						
CLC	●	●	●	●	●	●	●	●
COMP						●		●
CWG	●		●					
EUSART	●						●	
MSSP	●						●	
NCO	●		●					
PWM	●						●	
PWRT				●				
TMR0	●							
TMR1	●			●				●
TMR2	●							
WDT				●				

## 5.2.2.5 内部発振器周波数の選択

システム・クロック速度は、OSCCONレジスタの内部発振器周波数選択ビットIRCF<3:0>を使用してソフトウェアで選択できます。

16 MHz HFINTOSCのポストスケーリングされた出力と\*31 kHz LFINTOSCはマルチプレクサーに接続します（図5-1を参照）。OSCCONレジスタ（レジスタ5-1）の内部オシレータ周波数選択ビットIRCF<3:0>は、内部発信器の周波数出力を選択します。

**注釈：** リセット後、OSCCONレジスタのIRCF<3:0>ビットは「0111」に設定され、周波数選択は500 kHzに設定されます。ユーザーは、IRCFビットを変更して別の周波数を選択できます。

OSCCONレジスタのIRCF<3:0>ビットにより、一部の周波数で重複した選択が可能になります。これらの重複した選択は、システム設計のトレードオフ（一得一失）をもたらす可能性があります。特定の周波数の発振器ソースを変更すると、より低い消費電力を得ることができます。同じ発信器ソースを使用する周波数変更間で、より速い遷移時間を得ることができます。

## 5.2.2.6 内部発信器クロック・スイッチ・タイミング

HFINTOSCとLFINTOSCを切り替える場合、新しい発振器は、電力を節約するためにすでにシャットダウンされている場合があります（図5-7を参照）。このような場合は、OSCCONレジスタのIRCF<3:0>ビットが変更された後、周波数選択が行われるまでに遅延があります。OSCSTATレジスタは、HFINTOSCおよびLFINTOSC発信器の現在のアクティブ・ステータスを反映します。周波数選択の順序は次のとおりです：

1. OSCCONレジスタのIRCF<3:0>ビットが変更されます。
2. 新しいクロックがシャットダウンされると、クロック起動遅延が開始されます。
3. クロック・スイッチ回路は、現在のクロックの立ち下がりエッジを待ちます。
4. 現在のクロックはローに保持され、クロック・スイッチ回路は新しいクロックの立ち上がりエッジを待ちます。
5. これで新しいクロックがアクティブになります。
6. OSCSTATレジスタは必要に応じて更新されます。
7. 時計の切り替えが完了しました。

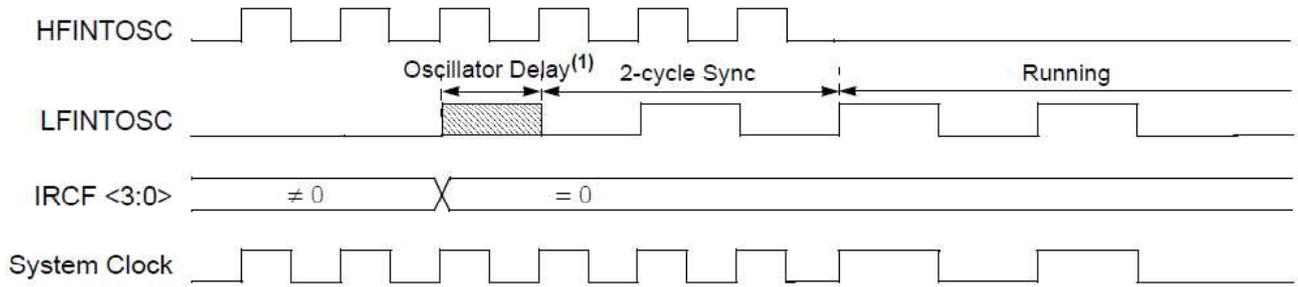
詳細については、図5-7を参照してください。

内部発振器の速度が同じソースの2つのクロック間で切り替えられる場合、新しい周波数が選択されるまでの起動遅延はありません。クロック・スイッチングの時間遅延を表5-3に示します。

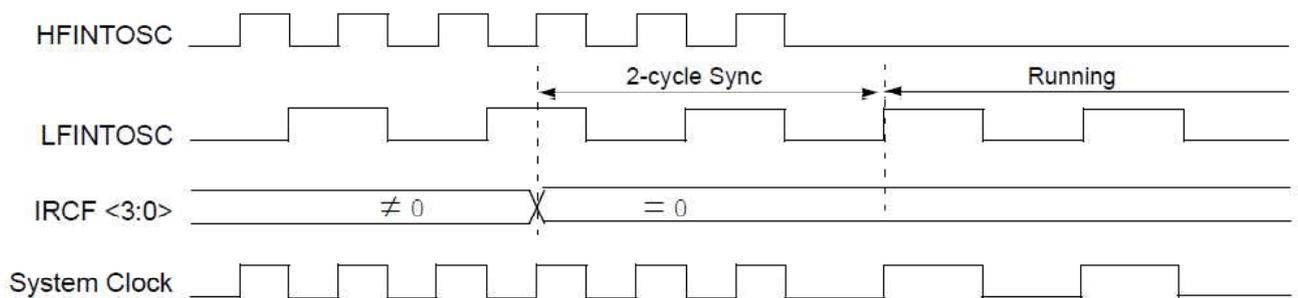
起動遅延の仕様は、表29-8「発振器のパラメーター」に記載されています。

図5-7: 内部発振器スイッチのタイミング

HFINTOSC → LFINTOSC (FSCM and WDT disabled)

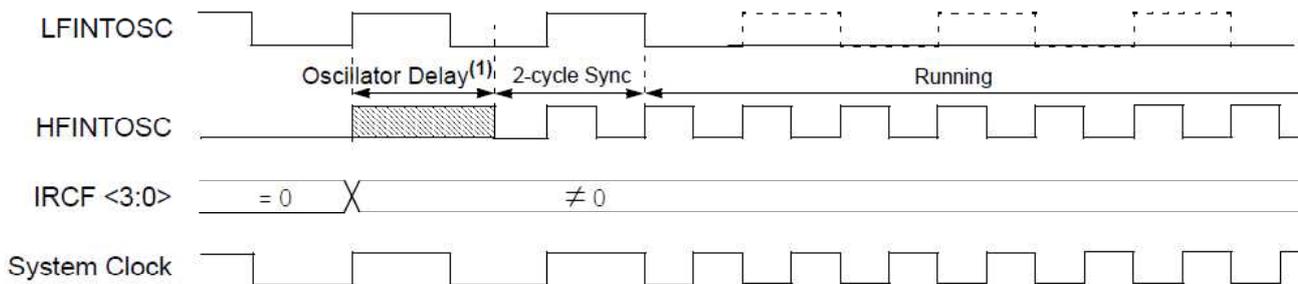


HFINTOSC → LFINTOSC (FSCMまたはWDTのいずれかが有効)



LFINTOSC → HFINTOSC

WDTまたはFSCMが有効になっていない限り、LFINTOSCはオフになります<sup>(2)</sup>



注1: 詳細については、表5-3「発振器のスイッチング遅延」を参照してください。

注2: 周辺機器がクロック・ソースとして選択した場合、LFINTOSCは実行を継続します。セクション5.2.2.4「周辺機器クロック・ソース」を参照してください。

## 5.3 クロック切り替え

システム・クロック・ソースは、OSCCONレジスタのシステム・クロック・選択 (SCS) ビットを使用して、ソフトウェアを介して外部クロック・ソースと内部クロック・ソースの間で切り替えることができます。SCSビットを使用して、次のクロック・ソースを選択できます。

- 構成ワードのFOSCビットによって決定されるデフォルトのシステム発信器
- 2次 (セカンダリ) 発信器32kHzクリスタル
- 内部発信器ブロック (INTOSC)

### 5.3.1 システム・クロック選択 (SCS) ビット

OSCCONレジスタのシステム・クロック・セレクト (SCS) ビットは、CPUと周辺機器に使用されるシステム・クロック・ソースを選択します。

- OSCCONレジスタのSCSビット=00の場合、システム・クロック・ソースは、構成ワードのFOSC<2:0>ビットの値によって決定されます。
- OSCCONレジスタのSCSビット=01の場合、システム・クロック・ソースは2次発信器です。
- OSCCONレジスタのSCSビット=1xの場合、システム・クロック・ソースは、OSCCONレジスタのIRCF<3:0>ビットによって選択された内部発信器周波数によって選択されます。リセット後、OSCCONレジスタのSCSビットは常にクリアされます。

**注釈:** 自動時計スイッチ、これは、2速起動またはフェイルセーフ・クロック・モニターから発生する可能性があります。OSCCONレジスタのSCSビットを更新しません。ユーザーは、OSCSTATレジスタのOSTSビットを監視して、現在のシステム・クロック・ソースを判別できます。表5-2を参照してください。

クロック・ソースを切り替えるときは、新しいクロックを安定させるために遅延が必要です。これらの発信器の遅延を表5-3に示します。

### 5.3.2 発信器起動タイマー・ステータス (OSTS) ビット

OSCSTATレジスタの発信器起動タイマー・ステータス (OSTS) ビットには、構成ワードでのFOSCビットの選択に依存するさまざまな定義があります。表5-2は、FOSC選択のOSTSビット値を定義しています。

OSTSビットの通常の機能は、FOSC<2:0>が外部発信器モードの1つ、HS、XT、またはLPを選択し、OSTが外部発信器からOSC1ピンのパルスをカウントしている場合です、OSTS = 0。OSTが1024パルスをカウントすると、OSTSビットを設定する必要があります、OSTS = 1。これは、発信器が安定して使用できる状態であることを示します。

フェイルセーフ・クロック・モニターおよび/または2速起動が有効になっている場合 (FCMEN= 1 および/またはIESO= 1)、デバイスはIRCF<3:0>によって選択された内部発信器 (INTOSC) を使用して動作します。ビット、OSTS= 0 の場合。OST期間が終了すると (OSTS= 1)、システム・クロックは選択した外部発信器に切り替わります。

フェイルセーフ・クロック・モニターと2速起動が無効になっている場合 (FCMEN= 0 およびIESO= 0)、OSTS= 0 の間デバイスはリセット状態に保持されます。OST期間が終了すると (OSTS= 1)、リセットが解放され、選択した外部発信器を使用して10 Foscサイクル後に実行が開始されます。

外部発信器モード (HS、XT、またはLP) 以外のクロック・ソースを使用したOSTSビットの定義については、表5-2を参照してください。

OSTSビットは、2次発信器のステータスを反映しています。

**表5-2: OSTSビットの定義**

FOSC<2:0>の選択	SCS<1:0> ビット		
	00	01	1x
OSTSの値			
INTOSC	0	0	0
ECH、ECM、ECL、EXTRC	1	0	0
HS、XT、LP	普通*	0	0

\* 発信器モードの通常の機能 (OSTS = 0)、同時にOSTカウント中 (OSTS = 1)、OSTカウントの期限が切れた後。

### 5.3.3 2次 (セカンダリ) 発信器

2次発信器は、タイマー1周辺機器に関連付けられた別個の水晶発信器です。これは、SOSCOデバイス・ピンとSOSCIデバイス・ピンの間に接続された32.768 kHz水晶振動子を使用した計時操作用に最適化されています。

2次発信器は、T1CONレジスタのT1OSCEN制御ビットを使用して有効になります。タイマー1周辺の詳細については、セクション19.0「ゲート制御を備えたタイマー1モジュール」を参照してください。

### 5.3.4 セカンダリ・オシレータ・レディ (SOSCR) ビット

ユーザーは、システム・クロック・ソースとして選択する前に、2次発信器を使用する準備ができていないことを確認する必要があります。OSCSTATレジスタのセカンダリ・オシレータ・レディ (SOSCR) ビットは、2次発信器を使用する準備ができていないかどうかを示します。SOSCRビットが設定された後、SCSビットを構成して2次発信器を選択できます。

### 5.3.5 スリープ前のクロックの切り替え

スリープ・モードに入る直前に古いクロックから新しいクロックへのクロック切り替えが必要な場合は、**SLEEP**命令を実行する前に切り替えが完了したことを確認する必要があります。そうしないと、切り替えが不完全になり、結果としてシステムクロックが完全に失われる可能性があります。クロックの切り替えは、**OSCSTAT**レジスタのクロック・ステータス・ビットを監視することによって確認されます。スイッチの確認は、新しいクロックのレディ・ビットがセットされているか、古いクロックのレディ・ビットがクリアされていることを検知することで実行できます。たとえば、PLL付きの内部発振器とPLLなしの内部発振器を切り替えるときは、**PLL**ビットを監視します。**PLL**が設定されると、32 MHz動作への切り替えが完了します。逆に、**PPLR**がクリアされると、32 MHz動作から選択された内部クロックへの切り替えが完了します。

### 5.4 2速クロック起動モード

2速起動モードは、外部発振器の起動とコード実行の間の待ち時間を最小限に抑えることで、さらに電力を節約します。スリープ・モードを多用するアプリケーションでは、2速起動により、起動に費やされた時間から外部発振器の起動時間が削除され、デバイスの全体的な消費電力を削減できます。このモードでは、アプリケーションはスリープからウェイクアップし、クロック・ソースとして**INTOSC**内部発振器ブロックを使用していくつかの命令を実行し、外部発振器が安定するのを待たずにスリープに戻ることができます。

2速起動は、発振器モジュールがLP、XT、またはHSモード用に構成されている場合に利点を提供します。オシレーター・スタートアップ・タイマー (OST) はこれらのモードで有効になっており、発振器をシステム・クロック・ソースとして使用するには、1024回の振動をカウントする必要があります。

発振器モジュールがLP、XT、またはHSモード以外のモードに設定されている場合、2速起動は無効になります。これは、外部クロック発振器が**POR**またはスリープの終了後に安定化時間を必要としないためです。

デバイスがスリープ・モードに入る前にOSTカウントが1024に達すると、**OSCSTAT**レジスタのOSTSビットが設定され、プログラム実行が外部発振器に切り替わります。ただし、覚醒 (ウェイクアップ) に費やす時間が非常に短い場合、システムが外部発振器から動作することはありません。

**注釈：** **SLEEP**命令を実行すると、発振器の起動時間が中止され、**OSCSTAT**レジスタのOSTSビットがクリアされたままになります。

## 5.4.1 2速起動モードの構成

2速起動モードは、次の設定で構成されます：

- IESO (構成ワードの) = 1;  
内部/外部スイッチ・オーバー・ビット (2速起動モードが有効)。
- (OSCCONレジスタの) SCS = 00。
- LP、XT、またはHSモード用に構成された構成ワードの FOSC<2:0>ビット。

2速スタートアップ・モードは、次の後に入ります：

- パワーオン・リセット (POR)、および有効になっている場合は、パワーアップ・タイマー (PWRT) の期限が切れた後、または
- スリープからのウェイクアップ (目覚め)。

**注釈：** FSCMを有効にすると、2速起動が自動的に有効になります。

## 5.4.2 2速起動シーケンス

1. パワーオン・リセットまたはスリープからのウェイクアップ。
2. 命令は、OSCCONレジスタのIRCF<3:0>ビットで設定された周波数で内部発振器によって実行を開始します。
3. OSTは1024クロック・サイクルをカウントできるようになっています。
4. OSTがタイムアウトし、内部発振器の立ち下がりエッジを待ちます。
5. OSTSが設定されます。
6. システム・クロックは、新しいクロックの次の立ち下がりエッジまでローに保持されます (LP、XT、またはHSモード)。
7. システム・クロックが外部クロック・ソースに切り替えられます。

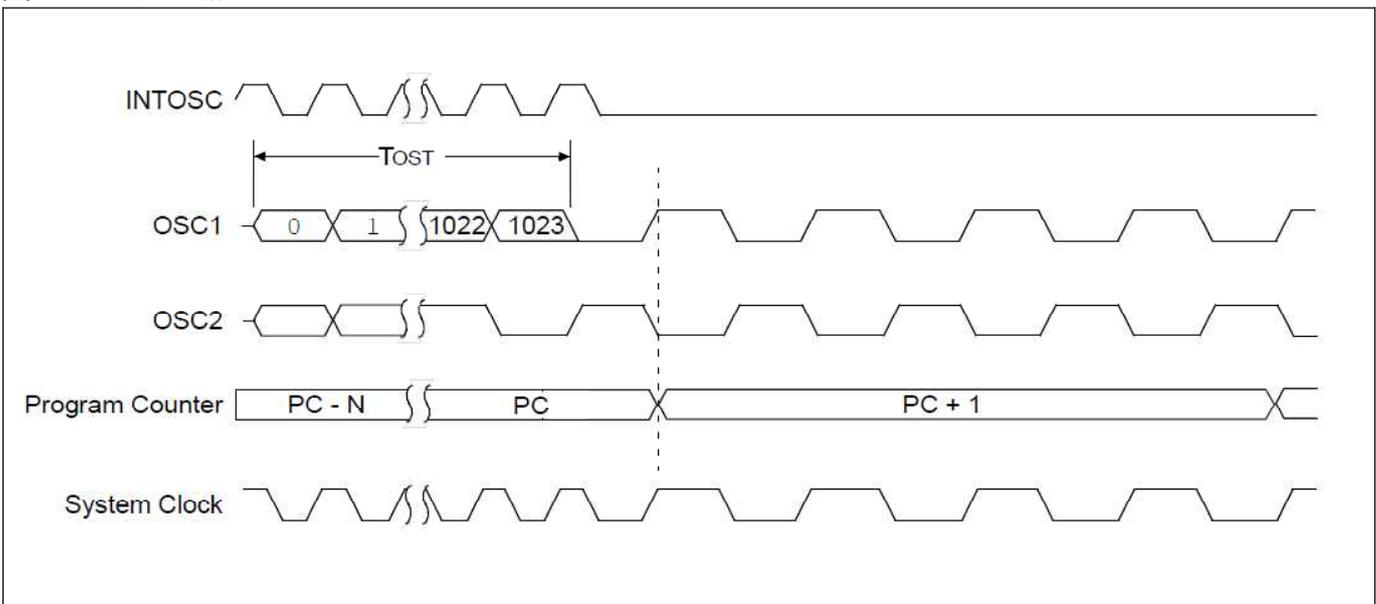
## 5.4.3 2速クロックの状態を確認する

OSCCONレジスタのOSTSビットの状態をチェックすると、CPUが構成ワードのFOSC<2:0>ビットで定義されている外部クロック・ソースから実行されているか、内部発振器から実行されているかが確認されます。表5-2を参照してください。

**表5-3: 発振器のスイッチング遅延**

切り替え前	切り替え後	発信器遅延
任意のクロック・ソース	LFINTOSC	各クロック・ソースの1サイクル
	HFINTOSC	2 $\mu$ s (約)
	ECH、ECM、ECL、EXTRC	2サイクル
	LP、XT、HS	1024クロック・サイクル (OST)
	2次発信器	1024の2次発振器サイクル

**図5-8: 2速起動**

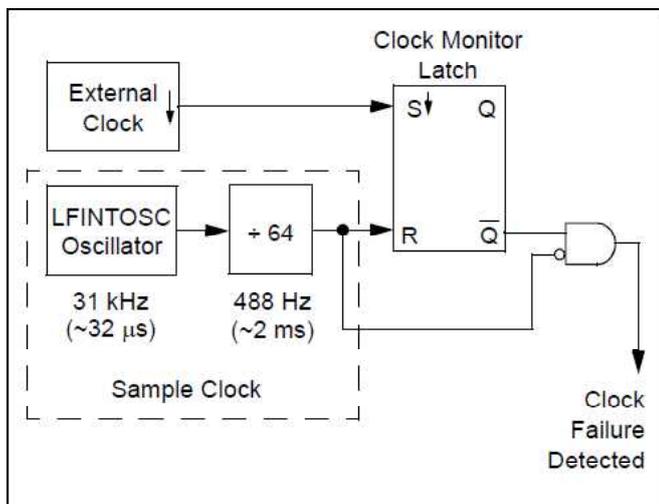


## 5.5 フェイルセーフ・クロック・モニター

フェイルセーフ・クロック・モニター (FSCM) を使用すると、外部発信器または外部クロックに障害が発生した場合でも、デバイスは動作を継続できます。発信器モードが選択されている場合、FSCMは、オシレーター・スタートアップ・タイマー (OST) の期限が切れた後、いつでも発信器の障害を検出できます。外部クロック・モードを選択した場合、FSCMは、デバイスがリセットから解放されるとすぐに障害を検出できます。

FSCMは、構成ワードのFCMENビットを設定することで有効になります。FSCMは、外部発信器モード (LP、XT、HS)、外部クロック・モード (ECH、ECM、ECL、EXTRC)、およびセカンダリ・オシレーター (SOSC) に適用できます。

図5-9: FSCMのブロック図



### 5.5.1 フェイルセーフ検出

FSCMモジュールは、クロックの立ち上がりエッジを監視し、タイム・ベースとしてLFINTOSCを使用することにより、障害が発生した発信器を検出します。図5-9を参照してください。故障した発信器の検出には、LFINTOSCの32~96サイクルかかります。図5-10に、FSCMモジュールのタイミング図を示します。

### 5.5.2 フェイルセーフ操作

外部クロックに障害が発生すると、FSCMはCPUクロックを内部クロック・ソースに切り替え、PIR2レジスタのOSFIFビットを設定します。内部クロック・ソースは、OSCCONレジスタのIRCF<3:0>ビットによって決定されます。

OSFIFビットがセットされている場合、PIE2レジスタのOSFIEビットが有効になっていると、割り込みが生成されます。その後、割り込みサービス・ルーチン (ISR) 内のユーザーのファームウェアは、クロックの障害から発生する可能性のある問題を軽減するための措置を講じることができます。

システム・クロックは、フェイルセーフ状態がクリアされるまで、内部クロック・ソースから供給され続けます。セクション5.5.3「フェイルセーフ状態のクリア」を参照してください。

### 5.5.3 フェイルセーフ状態のクリア

フェイルセーフ状態が存在する場合、ユーザーは、外部ソースで通常の操作に戻る前に、次のアクションを実行して状態をクリアする必要があります。

次のセクションでは、特定のクロック選択 (FOSCビット) およびクロック・スイッチング・モード (SCSビット設定) のフェイルセーフ条件をクリアする方法について説明します。

#### 5.5.3.1 SCS<1:0> = 00の外部発振器

FOSCビットが外部発信器を選択してフェイルセーフ状態が発生し (FOSC<2:0> = HS、XT、LP)、そして、クロック・スイッチがFOSC選択から実行するように選択されている場合 (SCS<1:0> = 00)、以下の手順で条件をクリアします。

##### SCS<1:0> = 00の場合 (FOSC選択から実行)

SCS<1:0> = 1x:

OSCCONレジスタのSCSビットを変更して、内部発信器ブロックを選択します。これにより、OSTタイマーがリセットされ、再び動作できるようになります。

OSFIF = 0:

PIR2レジスタのOSFIFビットをクリアします。

SCS<1:0> = 00:

OSCCONレジスタのSCSビットを変更して、FOSC構成ワードのクロック選択を選択します。これにより、OSTが開始されます。CPUは、OSTカウントに達するまで、内部オシレーターから動作し続けます。OSTが期限切れになると、クロック・モジュールが外部発信器に切り替わり、フェイルセーフ状態がクリアされます。

フェイルセーフ状態がまだ存在する場合、OSFIFビットはハードウェアによって再び設定されます。

#### 5.5.3.2 SCS<1:0> = 00の外部クロック

FOSCビットが外部クロックを選択してフェイルセーフ状態が発生し (FOSC<2:0> = ECH、ECM、ECL、EXTRC)、そして、クロック・スイッチがFOSC選択から実行するように選択されている場合 (SCS<1:0> = 00) の場合、以下の手順で条件をクリアします。

##### SCS<1:0> = 00の場合 (FOSC選択から実行)

SCS<1:0> = 1x:

OSCCONレジスタのSCSビットを変更して、内部発信器ブロックを選択します。これにより、OSTタイマーがリセットされ、再び動作できるようになります。

OSFIF = 0:

PIR2レジスタのOSFIFビットをクリアします。

SCS<1:0> = 00 :

OSCCONレジスタのSCSビットを変更して、FOSC構成ワードのクロック選択を選択します。OSTは外部クロックには適用できないため、クロックモジュールはすぐに外部クロックに切り替わり、そして、フェイルセーフ状態はクリアされず。

フェイルセーフ状態がまだ存在する場合、OSFIFビットはハードウェアによって再び設定されます。

### 5.5.3.3 SCS<1:0> = 01の2次発振器

クロック・スイッチが2次発信器の選択 (SCS<1:0> = 01) から実行するように選択された状態でフェイルセーフ状態が発生した場合、FOSCの選択に関係なく、次の手順を実行して状態をクリアします。

SCS<1:0> = 01 (2次発信器)

SCS<1:0> = 1x :

OSCCONレジスタのSCSビットを変更して、内部発信器ブロックを選択します。

OSFIF = 0 :

PIR2レジスタのOSFIFビットをクリアします。

SOSCRを読む :

OSTは2次発信器では使用されないため、ユーザーはOSCSTATレジスタのSOSCRビットを監視して、2次発信器の準備ができているかどうかを判断する必要があります。SOSCRビットがセットされると、2次発振器の準備が整います。

SCS<1:0> = 01 :

OSCCONレジスタのSCSビットを変更して、2次発振器を選択します。クロック・モジュールはすぐに2次発振器に切り替わり、フェイルセーフ状態がクリアされます。

フェイルセーフ状態がまだ存在する場合、OSFIFビットはハードウェアによって再び設定されます。

### 5.5.4 リセットまたはスリープからのウェイクアップ

FSCMは、外部発振器または外部クロックの障害を検出するように設計されています。

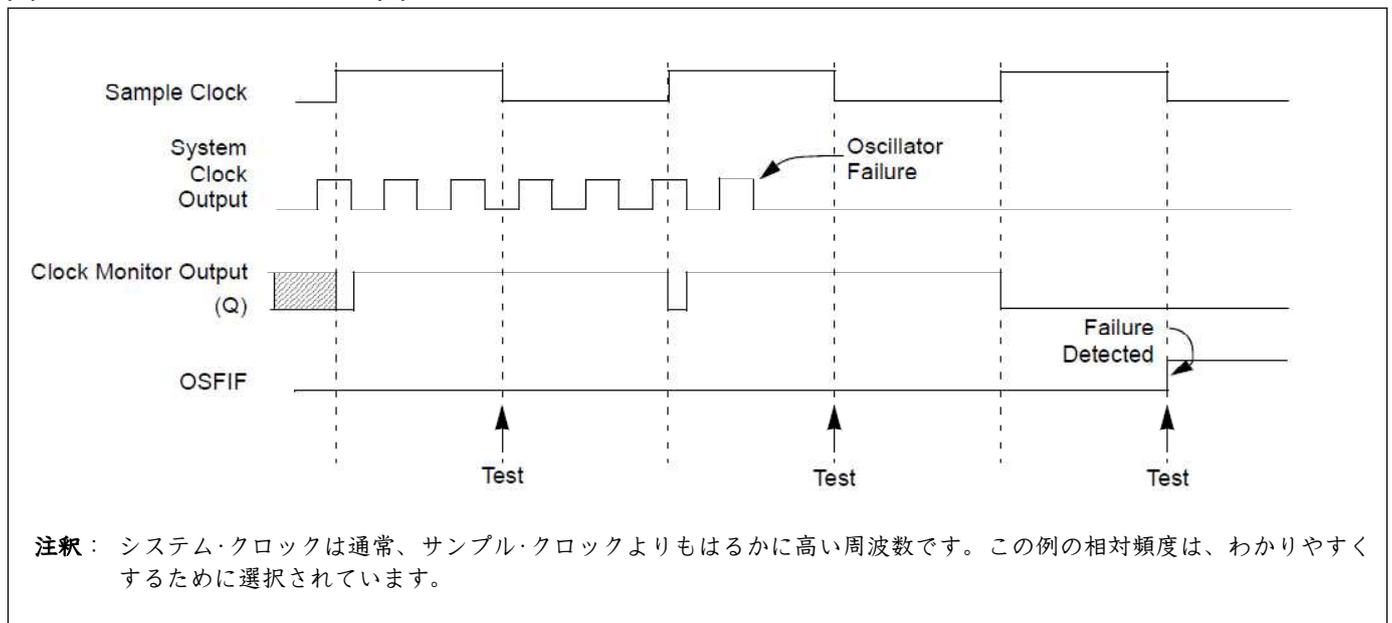
FSCMを外部発振器とともに使用する場合、FSCMがアクティブになる前に、オシレーター・スタートアップ・タイマー (OST) カウントが期限切れになる必要があります。OSTは、スリープから復帰した後、および任意のタイプのリセット後に使用されます。

FSCMを外部クロックとともに使用する場合、OSTは使用されず、リセットまたはウェイクアップが完了するとすぐにFSCMがアクティブになります。

FSCMを有効にすると、2速起動も有効になります。したがって、OSTが動作している間、デバイスは常にコードを実行します。

**注釈：** 発信器の起動時間の範囲が広い場合、発信器の起動中（つまり、リセットまたはスリープを終了した後）はフェイルセーフ回路がアクティブになりません。

図5-10: FSCMのタイミング図



## 5.6 レジスタの定義:発振器制御

### レジスタ5-1: OSCCON: 発信器制御レジスタ

U-0	R/W-0/0	R/W-1/1	R/W-1/1	R/W-1/1	U-0	R/W-0/0	R/W-0/0
—	IRCF<3:0>				—	SCS<1:0>	
bit 7						bit 0	

<b>記録:</b>		
R =読み取り可能ビット	W =書き込み可能ビット	U =実装されていないビット、「0」として読み取られる
u =ビットは変更されていません	x =ビットは不明です	-n/n =PORでの値およびBOR/他のすべてのリセットでの値
「1」=ビットが設定されている 「0」=ビットがクリアされている		

- bit 7     **未実装:** 「0」として読み取ります
- bit 6~3   **IRCF<3:0>**: 内部発振器周波数選択ビット
  - 1111 = 16 MHz
  - 1110 = 8 MHz
  - 1101 = 4 MHz
  - 1100 = 2 MHz
  - 1011 = 1 MHz
  - 1010 = 500 kHz<sup>(1)</sup>
  - 1001 = 250 kHz<sup>(1)</sup>
  - 1000 = 125 kHz<sup>(1)</sup>
  - 0111 = 500 kHz (リセット時のデフォルト)
  - 0110 = 250 kHz
  - 0101 = 125 kHz
  - 0100 = 62.5 kHz
  - 001x = 31.25 kHz
  - 000x = 31 kHz LF
- bit 2     **未実装:** 「0」として読み取ります
- bit 1~0   **SCS<1:0>**: システム・クロック選択ビット
  - 1x =内部発信器ブロック
  - 01 =2次発振器
  - 00 =構成ワードのFOSC<2:0>によって決定されるクロック。

注1: HFINTOSCから導出された重複周波数。

# PIC16(L)F1508/9

## レジスタ-5-2: OSCSTAT: オシレーター・ステータス・レジスタ

R-1/q	U-0	R-q/q	R-0/q	U-0	U-0	R-0/q	R-0/q
SOSCR	—	OSTS	HFIOFR	—	—	LFIOFR	HFIOFS
bit 7							bit 0

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません    x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている    「0」=ビットがクリアされている    q =条件付き

- bit 7      **SOSCR**: セカンダリ・オシレータ・レディ・ビット  
T1OSCEN = 1の場合:  
1 = 2次発振器の準備ができています  
0 = 2次発振器の準備ができていません  
T1OSCEN = 0の場合:  
1 = タイマー1クロックソースは常に準備ができています
- bit 6      **未実装**: 「0」として読み取ります
- bit 5      **OSTS**: オシレーター起動タイマー・ステータス・ビット  
FOSC<2:0>ビットがHS、XT、またはLPオシレーターを選択する場合:  
1 = OSTは1024クロックをカウントし、デバイスはFOSC<2:0>ビット選択によってクロックされます  
0 = OSTがカウントし、デバイスはIRCF<3:0>ビットによって選択された内部発振器 (INTOSC) からクロックされます。  
他のすべてのFOSC<2:0>ビット選択の場合:  
表5-2「OSTSビット定義」を参照してください。
- bit 4      **HFIOFR**: ハイ・フレッケンシー・インターナル・オシレーター・レディ・ビット (高周波内部発振器レディ)  
1 = HFINTOSCの準備ができました  
0 = HFINTOSCの準備ができていません
- bit 3~2    **未実装**: 「0」として読み取ります
- bit 1      **LFIOFR**: ロー・フレッケンシー・インターナル・オシレーター・レディ・ビット (低周波内部発振器レディ)  
1 = LFINTOSCの準備ができました  
0 = LFINTOSCの準備ができていません
- bit 0      **HFIOFS**: ハイ・フレッケンシー・インターナル・オシレーター・ステータス・ビット (高周波内部発振器ステータス)  
1 = HFINTOSC 16MHz発振器は安定しており、INTOSCを駆動しています  
0 = HFINTOSC 16MHzが安定しておらず、起動発振器がINTOSCを駆動しています

**表5-4: クロック・ソースに関連付けられたレジスタの概要**

名前	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ページ上の登録
OSCCON	—	IRCF<3:0>				—	SCS<1:0>		59
OSCSTAT	SOSCR	—	OSTS	HFIOFR	—	—	LFIOFR	HFIOFS	60
PIE2	OSFIE	C2IE	C1IE	—	BCL1IE	NCO1IE	—	—	77
PIR2	OSFIF	C2IF	C1IF	—	BCL1IF	NCO1IF	—	—	80
T1CON	TMR1CS<1:0>		T1CKPS<1:0>		T1OSCEN	T1SYNC	—	TMR1ON	163

記録： — =実装されていない場所、「0」と読みます。網掛けのセルはクロック・ソースでは使用されません。

**表5-5: クロック・ソースを使用した構成ワードの概要**

名前	ビット	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	ページ上の登録
CONFIG1	13:8	—	—	FCMEN	IESO	CLKOUTEN	BOREN<1:0>		—	41
	7:0	CP	MCLRE	PWRTE	WDTE<1:0>		FOSC<2:0>			

記録： — =実装されていない場所、「0」と読みます。網掛けのセルはクロック・ソースでは使用されません。

## 6.0 リセット

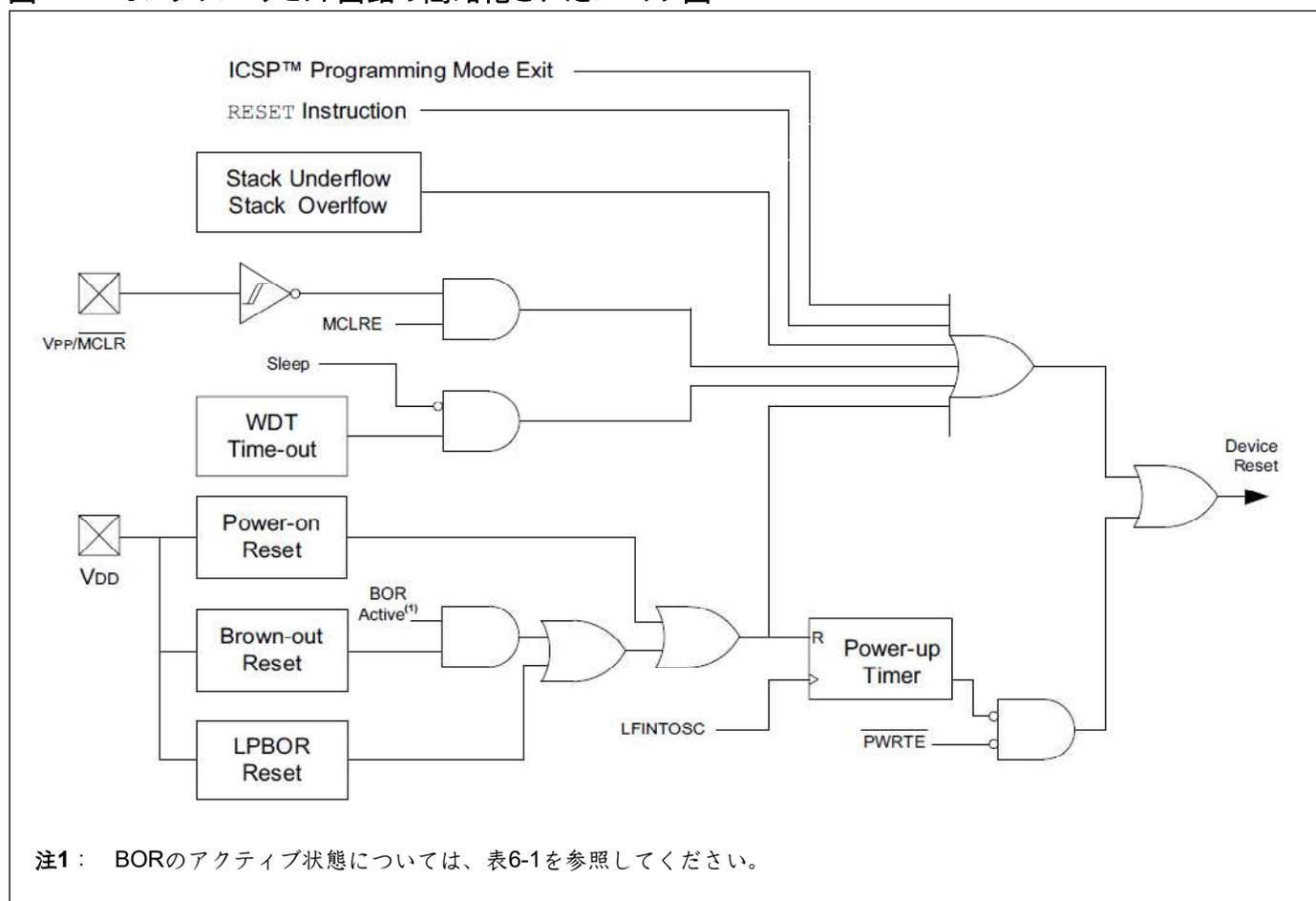
このデバイスをリセットするには、複数の方法があります：

- パワーオン・リセット (POR)
- ブラウンアウト・リセット (BOR)
- 低電力ブラウンアウト・リセット (LPBOR)
- MCLRリセット
- WDTリセット
- **RESET**命令
- スタック・オーバーフロー
- スタック・アンダーフロー
- プログラミング・モードの終了

VDDを安定させるために、オプションのパワーアップ・タイマーを有効にして、BORまたはPORイベント後のリセット時間を延長できます。

オンチップ・リセット回路の簡略ブロック図を図6-1に示します。

図6-1: オンチップ・リセット回路の簡略化されたブロック図



## 6.1 パワーオン・リセット(POR)

POR回路は、VDDが最小動作の許容レベルに達するまで、デバイスをリセット状態に保ちます。立ち上がりが遅いVDD、速い動作速度、またはアナログ性能には、最小VDDよりも大きい値が必要になる場合があります。PWRT、BOR、またはMCLR機能を使用して、すべてのデバイス動作条件が満たされるまで起動期間を延長できます。

### 6.1.1 パワーアップ・タイマー(PWRT)

パワーアップ・タイマーは、PORまたはブラウンアウト・リセットで公称64 msのタイムアウトを提供します。

PWRTがアクティブである限り、デバイスはリセット状態に保たれます。PWRT遅延により、VDDが許容レベルまで上昇するための追加時間が可能になります。パワーアップ・タイマーは、構成ワードのPWRTビットをクリアすることで有効になります。

パワーアップ・タイマーは、PORとBORの解除後に開始されます。

詳細については、アプリケーション・ノートAN607「電源投入時のトラブル・シューティング」(DS00607)を参照してください。

表6-1: BORの動作モード

BOREN<1:0>	SBOREN	デバイス・モード	BORモード	命令の実行: PORの解放またはスリープからのウェイクアップ
11	X	X	アクティブ	BORの準備ができるまで待機します <sup>(1)</sup> (BORRDY = 1)
10	X	起きている	アクティブ	BORの準備ができるのを待ちます (BORRDY = 1)
		スリープ	無効	
01	1	X	アクティブ	BORの準備ができるまで待機します <sup>(1)</sup> (BORRDY = 1)
	0	X	アクティブ	
00	X	X	アクティブ	すぐに始まります (BORRDY = x)

注1: これらの特定のケース、「PORの解放」および「スリープからのウェイクアップ」では、起動の遅延はありません。BOR回路はBOREN<1:0>ビットによって強制的にオンになるため、CPUが命令を実行する準備ができる前にBORレディフラグ (BORRDY = 1) が設定されます。

### 6.2.1 BORが常にオン

構成ワードのBORENビットが「11」にプログラムされている場合、BORは常にオンです。デバイスの起動は、BORの準備が整い、VDDがBORしきい値より高くなるまで遅延されません。

BOR保護は、スリープ中にアクティブになります。BORは、スリープからのウェイクアップを遅らせません。

### 6.2.2 BORはスリープ状態でオフになっている

構成ワードのBORENビットが「10」にプログラムされている場合、スリープ時を除いて、BORはオンになります。デバイスの起動は、BORの準備が整い、VDDがBORしきい値より高くなるまで遅延されます。

## 6.2 ブラウンアウト・リセット(BOR)

VDDが選択可能な最小レベルに達すると、BOR回路はデバイスをリセット状態に保ちます。PORとBORの間で、実行保護のための完全な電圧範囲掩護を実装できます。

ブラウンアウト・リセット・モジュールには、構成ワードのBOREN<1:0>ビットによって制御される4つの動作モードがあります。4つの動作モードは次のとおりです:

- BORは常にオンです
- スリープ中はBORがオフになっています
- BORはソフトウェアによって制御されます
- BORは常にオフです

詳細については、表6-1を参照してください。

ブラウンアウト・リセット電圧レベルは、設定ワードでBORVビットを設定することで選択できます。

VDDノイズ除去フィルターは、BORが小さなイベントでトリガーされるのを防ぎます。パラメータTBORDCよりも長い時間VDDがVporを下回ると、デバイスはリセットされます。詳細については、図6-2を参照してください。

スリープ中はBOR保護はアクティブではありません。デバイスのウェイクアップは、BORの準備ができるまで遅延されます。

### 6.2.3 ソフトウェアによって制御されるBOR

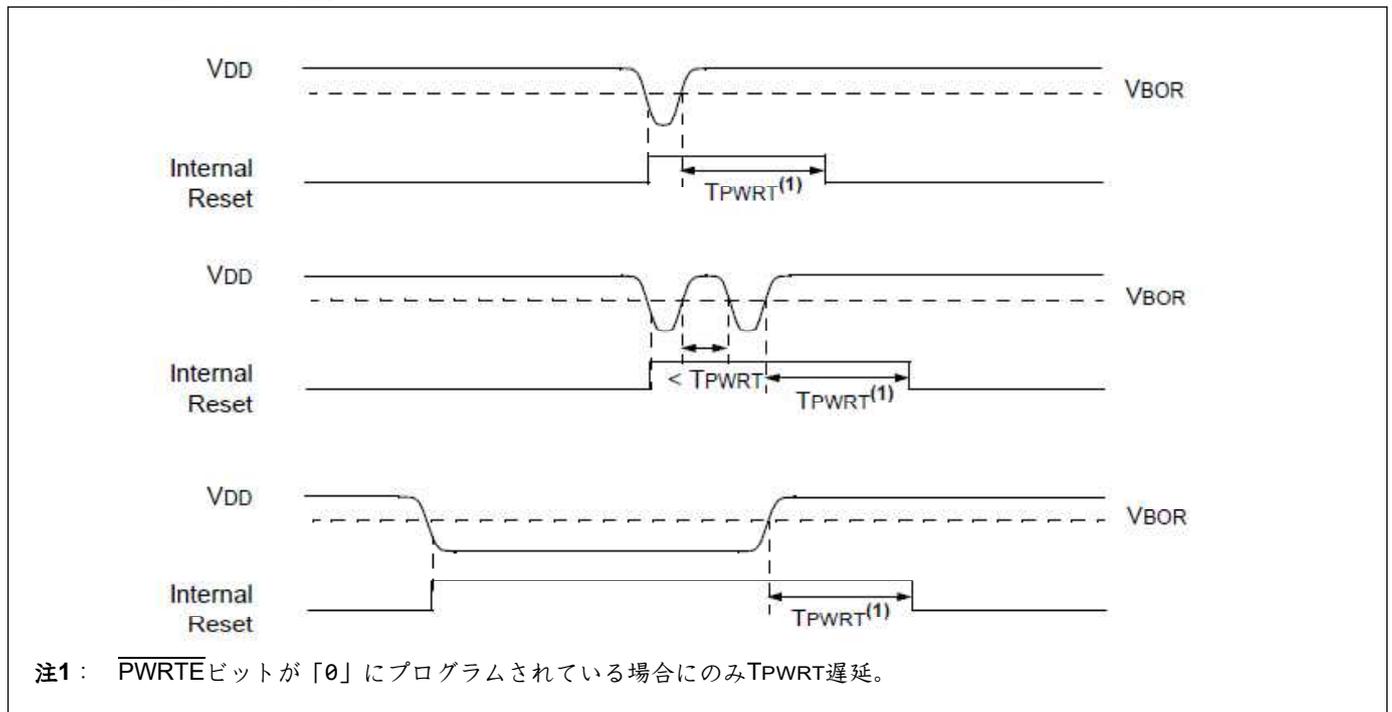
構成ワードのBORENビットが「01」にプログラムされている場合、BORはBORCONレジスターのSBORENビットによって制御されます。デバイスの起動は、BOR準備完了状態またはVDDレベルによって遅延されません。

BOR保護は、BOR回路の準備ができるとすぐに開始されます。BOR回路のステータスは、BORCONレジスターのBORRDYビットに反映されます。

BOR保護はSleepによって変更されません。

# PIC16(L)F1508/9

図6-2: 電圧低下の状況



## 6.3 レジスタ定義: BOR制御

レジスタ6-1: BORCON: ブラウンアウト・リセット制御レジスタ

R/W-1/u	R/W-0/u	U-0	U-0	U-0	U-0	U-0	R-q/u
SBOREN	BORFS	—	—	—	—	—	BORRDY
bit 7							bit 0

記録:  
 R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
 u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
 「1」=ビットが設定されている      「0」=ビットがクリアされている      q =値は条件によって異なります

bit 7 **SBOREN**: ソフトウェア・ブラウンアウト・リセット・イネーブル・ビット

**BOREN<1:0>**が構成ワード = 01の場合:

1 = BOR有効

0 = BOR無効

**BOREN<1:0>**が構成ワード = 01の場合:

SBORENは読み取り/書き込み可能ですが、BORには影響しません。

bit 6 **BORFS**: ブラウンアウト・リセット・ファスト・スタート・ビット(1)

**BOREN<1:0>** = 10 (スリープ状態で無効) または **BOREN<1:0>** = 01 (ソフトウェア制御下):

1 = バンド・ギャップは常に強制されます (スリープ/ウェイクアップ/動作の場合をカバーします)

0 = バンド・ギャップは正常に動作し、オフになる場合があります

**BOREN<1:0>** = 11 (常にオン) または **BOREN<1:0>** = 00 (常にオフ) の場合

BORFSは読み取り/書き込みですが、効果はありません。

bit 5~1 **未実装**: 「0」として読み取ります

bit 0 **BORRDY**: ブラウンアウト・リセット・サーキット・レディ・ステータス・ビット

1 = ブラウンアウトリセット回路がアクティブです

0 = ブラウンアウトリセット回路は非アクティブです

注1: **BOREN<1:0>**ビットは構成ワードにあります。

## 6.4 ローパワー・ブラウンアウト・リセット(LPBOR)

低電力電圧低下リセット (LPBOR) は、BORのように動作して、VDDピンの低電圧状態を検出します。低すぎる電圧が検出されると、デバイスはリセット状態に保たれます。これが発生すると、レジスター・ビット (BOR) が変更され、BORリセットが発生したことを示します。PCONのBORビットは、BORとLPBORの両方に使用されます。レジスター6-2を参照してください。

LPBOR電圧しきい値 (ラップ・ポード) はBOR (Vpor) よりも許容範囲が広いですが、動作に必要な電流 (LPBOR電流) ははるかに少なくなります。LPBORは、BORが無効 (BOREN = 00) またはスリープ・モードで無効 (BOREN = 10) として構成されている場合に使用することを目的としています。

LPBORが他のモジュールとどのように相互作用するかについては、図6-1を参照してください。

### 6.4.1 LPBORを有効にする

LPBORは、構成ワードのLPBORビットによって制御されます。デバイスが消去されると、LPBORモジュールはデフォルトで無効になります。

## 6.5 MCLR

MCLRは、デバイスをリセットできるオプションの外部入力です。MCLR機能は、構成ワードのMCLREビットと構成ワードのLVPビットによって制御されます (表6-2)。

表6-2: MCLR構成

MCLRE	LVP	MCLR
0	0	無効
1	0	有効
x	1	有効

### 6.5.1 MCLRが有効

MCLRが有効で、ピンがローに保持されている場合、デバイスはリセットに保持されます。MCLRピンは内部の弱いプルアップを介してVDDに接続されています。

デバイスのMCLRリセット・パスにノイズ・フィルタがあります。フィルタは小さなパルスを検出しても無視します。

**注釈:** リセットはMCLRピンをローに駆動しません。

### 6.5.2 MCLRが無効

MCLRが無効になっている場合、ピンは汎用入力として機能し、内部の弱いプルアップはソフトウェアの制御下にありません。詳細については、セクション11.3「PORTAレジスター」を参照してください。

## 6.6 ウォッチドッグ・タイマー (WDT)リセット

ファームウェアがタイムアウト期間内にCLRWD命令を発行しない場合、ウォッチドッグ・タイマーはリセットを生成します。STATUSレジスターのTOビットとPDビットは、WDTリセットを示すために変更されます。詳細については、セクション9.0「ウォッチドッグ・タイマー (WDT)」を参照してください。

## 6.7 RESET命令

RESET命令により、デバイスがリセットされます。PCONレジスターのRIビットは「0」に設定されます。RESET命令が発生した後のデフォルト条件については、表6-4を参照してください。

## 6.8 スタック・オーバーフロー/アンダーフロー・リセット

スタックがオーバーフローまたはアンダーフローすると、デバイスがリセットされます。PCONレジスターのSTKOVFまたはSTKUNFビットは、リセット条件を示します。これらのリセットは、構成ワードのSTVRENビットを設定することで有効になります。詳細については、セクション3.5.2「オーバーフロー/アンダーフローリセット」を参照してください。

## 6.9 プログラミング・モード終了

プログラミング・モードを終了すると、デバイスはPORが発生したかのように動作します。

## 6.10 パワーアップ・タイマー

パワーアップ・タイマーは、オプションで、BORまたはPORイベント後のデバイスの実行を遅らせます。このタイマーは通常、デバイスの実行を開始する前に、VDDを安定させるために使用されます。

パワーアップ・タイマーは、構成ワードのPWRT $\overline{E}$ ビットによって制御されます。

## 6.11 スタートアップ・シーケンス

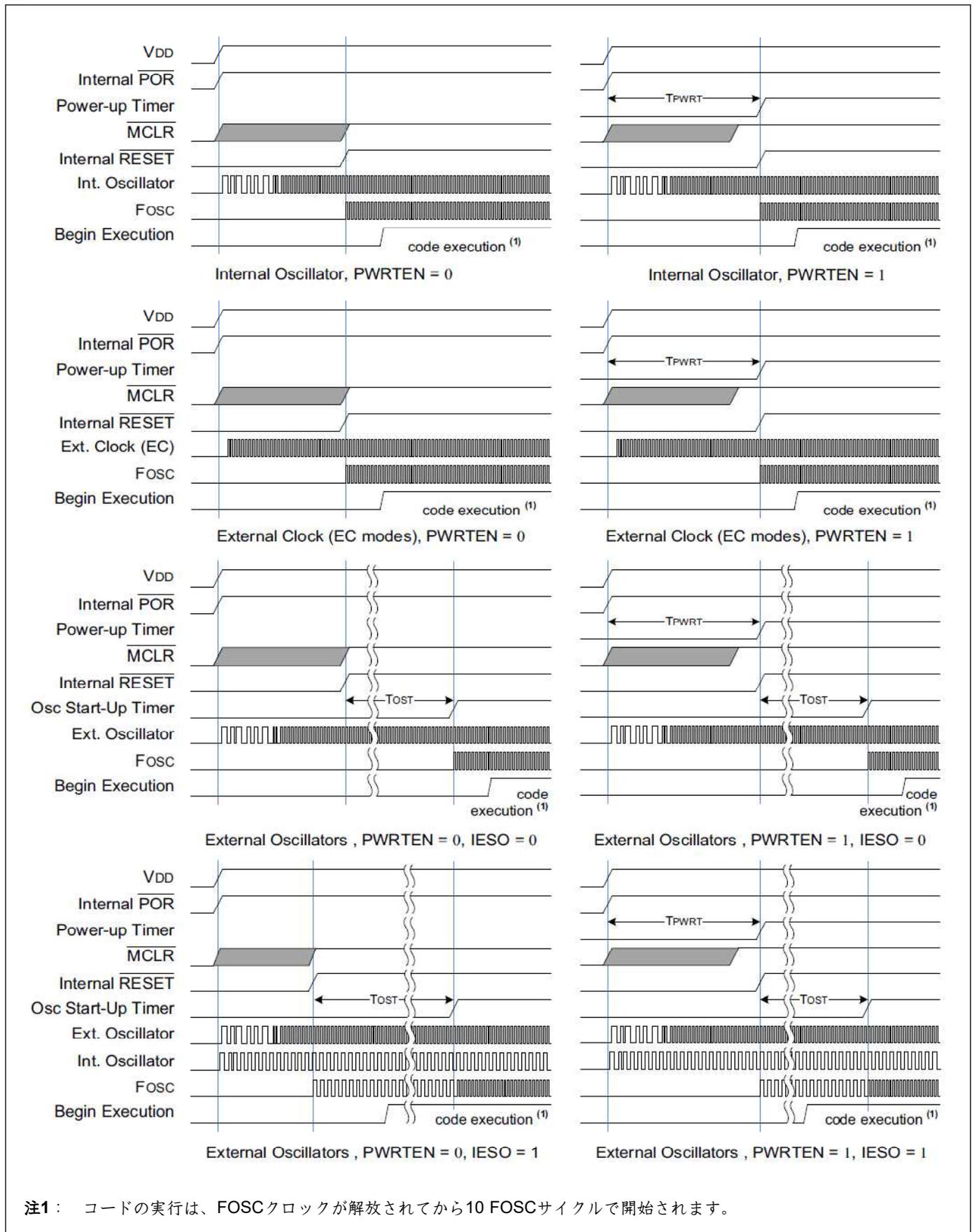
PORまたはBORのリリース時に、デバイスが実行を開始する前に、次のことが発生する必要があります：

1. パワーアップ・タイマーは完了するまで実行されます (有効になっている場合)。
2. MCLRを解放する必要があります (有効になっている場合)。

合計タイムアウトは、発振器の構成とパワーアップ・タイマーの構成によって異なります。詳細については、セクション5.0「発信器モジュール (フェイルセーフ・クロック・モニタ付き)」を参照してください。

パワーアップ・タイマーは、MCLRリセットとは独立して実行されます。MCLRが十分に長くローに保たれると、パワーアップタイマーは期限切れになります。MCLRをHighにすると、デバイスは10 FOSSサイクル後に実行を開始します (図6-3を参照)。これは、テスト目的や、並行して動作する複数のデバイスを同期する場合に役立ちます。

図6-3: スタートアップ・シーケンスのリセット



注1: コードの実行は、FOSCクロックが解放されてから10 FOSCサイクルで開始されます。

## 6.12 リセットの原因の特定

リセットすると、STATUSレジスタとPCONレジスタの複数のビットが更新され、リセットの原因が示されます。表6-3および表6-4に、これらのレジスタのリセット条件を示します。

表6-3: ステータス・ビットとその重要性をリセットする

STKOVF	STKUNF	RWDT	RMCLR	RI	POR	BOR	TO	PD	状態
0	0	1	1	1	0	x	1	1	パワーオン・リセット
0	0	1	1	1	0	x	0	x	非合法、TOがPORに設定されている
0	0	1	1	1	0	x	x	0	非合法、PDがPORに設定されている
0	0	u	1	1	u	0	1	1	ブラウンアウト・リセット
u	u	0	u	u	u	u	0	u	WDTリセット
u	u	u	u	u	u	u	0	0	WDTのスリープからのウェイクアップ
u	u	u	u	u	u	u	1	0	スリープからのウェイクアップ割り込み
u	u	u	0	u	u	u	u	u	通常動作中のMCLRリセット
u	u	u	0	u	u	u	1	0	スリープ中のMCLRリセット
u	u	u	u	0	u	u	u	u	RESET命令が実行されました
1	u	u	u	u	u	u	u	u	スタック・オーバーフローのリセット (STVREN = 1)
u	1	u	u	u	u	u	u	u	スタック・アンダーフローリセット (STVREN = 1)

表6-4: 特殊レジスタのリセット条件

状態	プログラム・カウンタ	STATUSレジスタ	PCONレジスタ
パワーオン・リセット	0000h	---1 1000	00-- 110x
通常動作中のMCLRリセット	0000h	---u mu u mu u	uu-- 0uuu
スリープ中のMCLRリセット	0000h	---1 0uuu	uu-- 0uuu
WDTリセット	0000h	---0 mu u mu u	uu-- uuuu
WDTのスリープからのウェイクアップ	PC + 1	---0 0uuu	uu-- uuuu
ブラウンアウト・リセット	0000h	---1 1uuu	00-- 11u0
スリープからのウェイクアップで割り込み	PC + 1 <sup>(1)</sup>	---1 0uuu	uu-- uuuu
RESET命令が実行されました	0000h	---u uuuu	uu-- u0uu
スタック・オーバーフロー・リセット (STVREN = 1)	0000h	---u uuuu	1u-- uuuu
スタック・アンダーフロー・リセット (STVREN = 1)	0000h	---u uuuu	u1-- uuuu

記録: u = 変更なし。x = 不明。- = 未実装ビット、「0」として読み取ります。

注1: ウェイクアップが割り込みによるもので、グローバル割り込みイネーブルビット (GIE) が設定されている場合、PC + 1の実行後、リターンアドレスがスタックにプッシュされ、PCに割り込みベクター (0004h) がロードされます。

# PIC16(L)F1508/9

## 6.13 電力制御(PCON)レジスター

電力制御 (PCON) レジスターには、以下を区別するためのフラグ・ビットが含まれています：

- パワーオン・リセット ( $\overline{\text{POR}}$ )
- ブラウンアウト・リセット ( $\overline{\text{BOR}}$ )
- リセット命令リセット ( $\overline{\text{RI}}$ )
- $\overline{\text{MCLR}}$ リセット ( $\overline{\text{RMCLR}}$ )
- ウォッチドッグ・タイマーのリセット ( $\overline{\text{RWDT}}$ )
- スタック・アンダーフロー・リセット (STKUNF)
- スタック・オーバーフロー・リセット (STKOVF)

PCONレジスター・ビットはレジスター6-2に示されています。

## 6.14 レジスターの定義： 電力制御

### レジスター6-2: PCON: パワー・コントロール・レジスター(電力制御レジスター)

R/W/HS-0/q	R/W/HS-0/q	U-0	R/W/HC-1/q	R/W/HC-1/q	R/W/HC-1/q	R/W/HC-q/u	R/W/HC-q/u
STKOVF	STKUNF	—	RWDT	RMCLR	RI	POR	BOR
bit 7							bit 0

<b>記録：</b>	
HC = ビットはハードウェアによってクリアされます	HS = ビットはハードウェアによって設定されます
R = 読み取り可能ビット	W = 書き込み可能ビット
u = ビットは変更されていません	x = ビットは不明です
「1」 = ビットが設定されている	「0」 = ビットがクリアされている
	U = 未実装ビット、「0」として読み取られる
	-n/n = PORでの値およびBOR /他のすべてのリセットでの値
	q = 値は条件によって異なります

- bit 7 **STKOVF**： スタック・オーバーフロー・フラグ・ビット  
 1 = スタックオーバーフローが発生しました  
 0 = スタックオーバーフローが発生していないか、ファームウェアによってクリアされていません
- bit 6 **STKUNF**： スタック・アンダーフロー・フラグ・ビット  
 1 = スタック・アンダーフローが発生しました  
 0 = スタック・アンダーフローが発生していないか、ファームウェアによってクリアされていません
- bit 5 **未実装**： 「0」として読み取ります
- bit 4 **RWDT**： ウォッチドッグ・タイマー・リセット・フラグ・ビット  
 1 = ウォッチドッグ・タイマーのリセットが発生していないか、ファームウェアによって設定されていません  
 0 = ウォッチドッグ・タイマーのリセットが発生しました (ハードウェアによってクリアされます)
- bit 3 **RMCLR**：  $\overline{\text{MCLR}}$ リセット・フラグ・ビット  
 1 =  $\overline{\text{MCLR}}$ リセットが発生していないか、ファームウェアによって設定されていません  
 0 =  $\overline{\text{MCLR}}$ リセットが発生しました (ハードウェアによってクリアされます)
- bit 2 **RI**： **RESET**命令フラグ・ビット  
 1 = **RESET**命令が実行されていないか、ファームウェアによって設定されていません  
 0 = **RESET**命令が実行されました (ハードウェアによってクリアされます)
- bit 1 **POR**： パワーオン・リセット・ステータス・ビット  
 1 = パワーオン・リセットは発生しませんでした  
 0 = パワーオン・リセットが発生しました (パワーオン・リセットが発生した後にソフトウェアで設定する必要があります)
- bit 0 **BOR**： ブラウンアウト・リセット・ステータス・ビット  
 1 = ブラウンアウト・リセットは発生しませんでした  
 0 = ブラウンアウト・リセットが発生しました (パワーオン・リセットまたはブラウンアウト・リセットが発生した後、ソフトウェアで設定する必要があります)

表6-5: リセットに関連するレジスタの要約

名前	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ページ上の登録
BORCON	SBOREN	BORFS	—	—	—	—	—	BORRDY	64
PCON	STKOVF	STKUNF	—	RWDT	RMCLR	RI	POR	BOR	68
STATUS	—	—	—	TO	PD	Z	DC	C	19
WDTCON	—	—	WDTPS<4:0>					SWDTEN	88

記録: — =未実装ビット、「0」として読み取ります。網掛けのセルはリセットでは使用されません。

表6-6: 構成ワードの要約とリセット

名前	ビット	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	ページ上の登録
CONFIG1	13:8	—	—	FCMEN	IESO	CLKOUTEN	BOREN<1:0>		—	43
	7:0	CP	MCLRE	PWRTE	WDTE<1:0>		FOSC<2:0>			
CONFIG2	13:8	—	—	LVP	DEBUG	LPBOR	BORV	STVREN	—	43
	7:0	—	—	—	—	—	WRT<1:0>			

記録: — =実装されていない場所、「0」と読みます。網掛けのセルはリセットでは使用されません。

## 7.0 割り込み

割り込み機能により、特定のイベントが通常のプログラム・フローを先制できます。ファームウェアは、割り込みの原因を特定し、それに応じて動作するために使用されます。一部の割り込みは、MCUをスリープ・モードからウェイクアップするように構成できます。

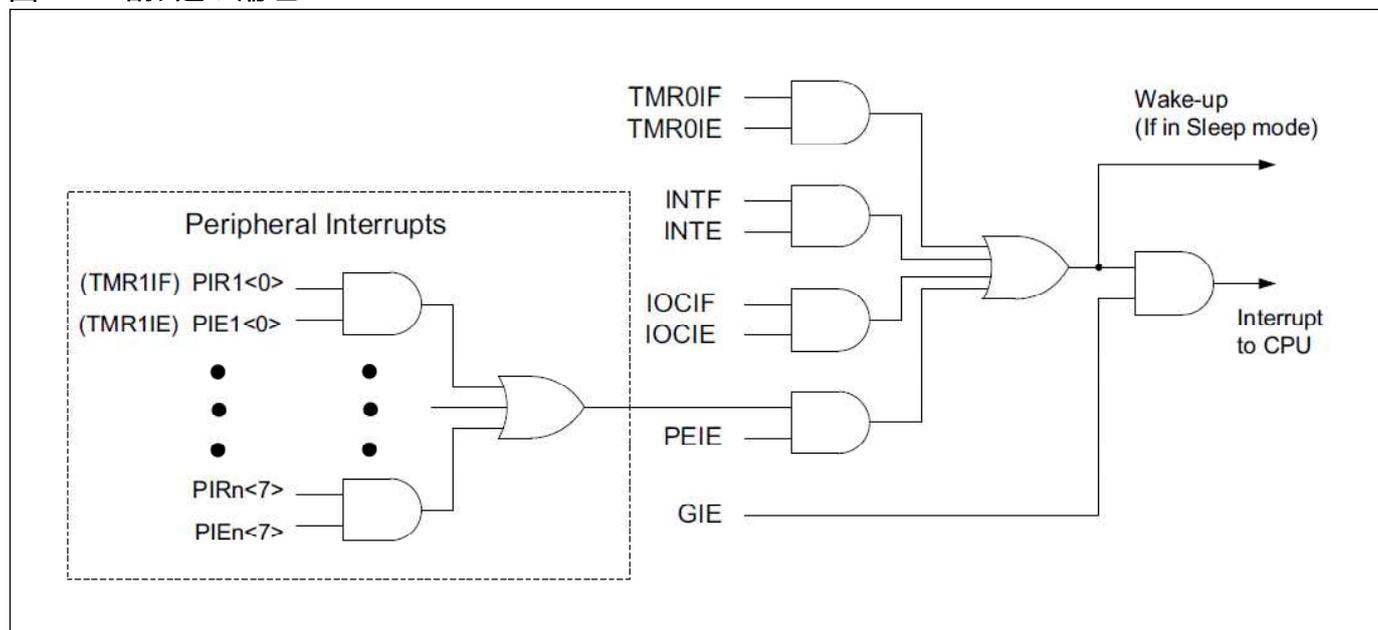
この章には、割り込みに関する次の情報が含まれています：

- 操作
- 割り込み潜在（レイテンシー）
- スリープ中に中断する
- INTピン
- 自動文脈（コンテキスト）保存

多くの周辺機器は割り込みを生成します。詳細については、対応する章を参照してください。

割り込み論理のブロック図を図7-1に示します。

図7-1: 割り込み論理



## 7.1 操作(動作)

デバイスをリセットすると、割り込みは無効になります。これらは、次のビットを設定することで有効になります：

- INTCONレジスタのGIEビット
- 特定の割り込みイベントの割り込み許可ビット
- INTCONレジスタのPEIEビット（割り込みイベントの割り込み許可ビットがPIE1、PIE2、およびPIE3レジスタに含まれている場合）

INTCON、PIR1、PIR2、およびPIR3レジスタは、割り込みフラグ・ビットを介して個々の割り込みを記録します。GIE、PEIE、および個々の割り込み許可ビットのステータスに関係なく、割り込みフラグ・ビットが設定されます。

GIEビットが設定されているときに割り込みイベントが発生すると、次のイベントが発生します：

- 現在のプリフェッチされた命令が流されます
- GIEビットがクリアされます
- 現在のプログラム・カウンタ（PC）がスタックにプッシュされます
- 重要なレジスタは自動的にシャドウ・レジスタに保存されます（「セクション7.5「自動コンテキスト保存」を参照）。
- PCには割り込みベクタ0004hがロードされています

割り込みサービス・ルーチン（ISR）内のファームウェアは、割り込みフラグ・ビットをポーリングすることによって割り込みのソースを決定する必要があります。割り込みの繰り返しを避けるために、ISRを終了する前に割り込みフラグ・ビットをクリアする必要があります。GIEビットがクリアされているため、ISRの実行中に発生した割り込みは、その割り込みフラグを介して記録されますが、プロセッサが割り込みベクタにリダイレクトすることはありません。

**RETFIE**命令は、スタックから前のアドレスをポップし、シャドウ・レジスタから保存された文脈（コンテキスト）を復元し、GIEビットを設定することにより、ISRを終了します。

特定の割り込みの動作の詳細については、その周辺の章を参照してください。

- 注1：** 他の許可（イネーブル）ビットの状態に関係なく、個々の割り込みフラグ・ビットが設定されます。
- 2：** GIEビットがクリアされている間、すべての割り込みは無視されます。GIEビットがクリアされている間に発生した割り込みは、GIEビットが再度セットされたときに処理されます。

## 7.2 割り込み潜在

割り込み潜在（レイテンシ）は、割り込みイベントが発生してから、割り込みベクタでのコード実行が開始されるまでの時間として定義されます。同期割り込みの潜在は、3または4命令サイクルです。非同期割り込みの場合、割り込みが発生するタイミングに応じて、潜在は3~5命令サイクルです。詳細については、図7-2および図7-3を参照してください。

# PIC16(L)F1508/9

図7-2: 割り込みの潜在

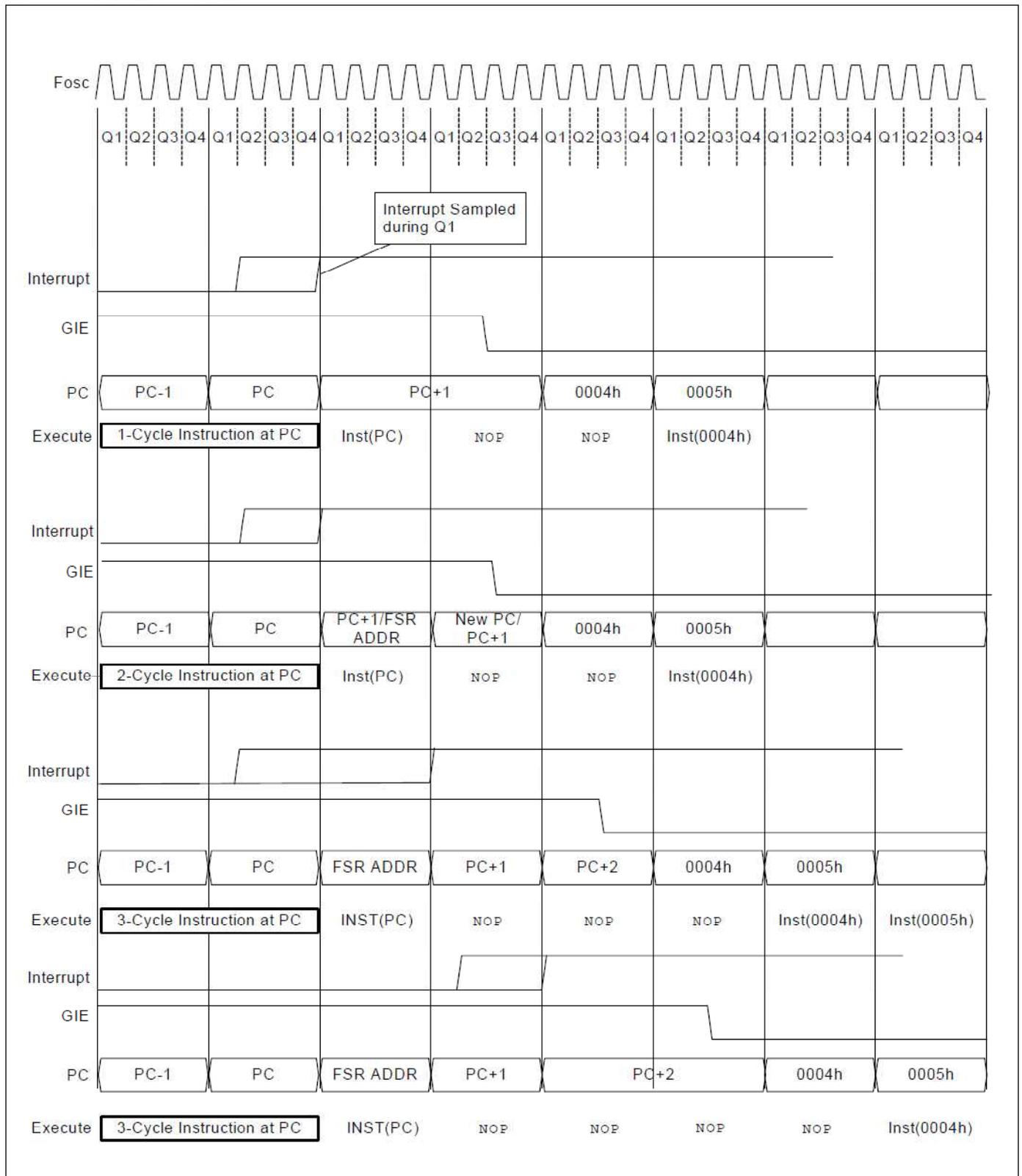
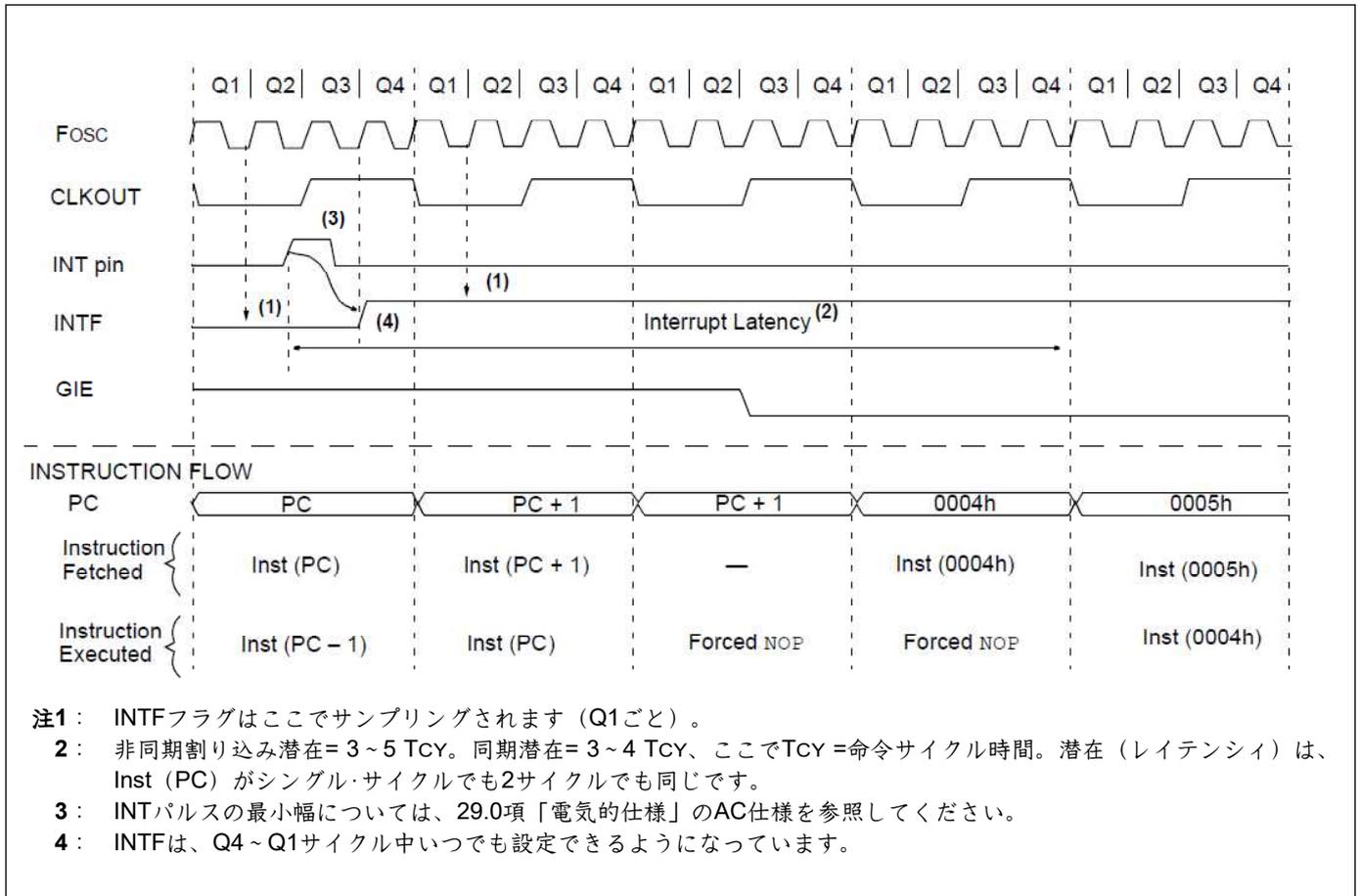


図7-3: INTピンの割り込みタイミング



## 7.3 スリープ中の割り込み

一部の割り込みは、スリープから復帰するために使用できます。スリープから復帰するには、周辺機器がシステム・クロックなしで動作できる必要があります。割り込みソースは、スリープに入る前に適切な割り込み許可ビットを設定する必要があります。

スリープから復帰するとき、GIEビットも設定されている場合、プロセッサは割り込みベクターに分岐します。それ以外の場合、プロセッサは**SLEEP**命令の後に命令を実行し続けます。**SLEEP**命令の直後の命令は、常にISRに分岐する前に実行されます。詳細については、セクション8.0「パワーダウン・モード（スリープ）」を参照してください。

## 7.4 INTピン

INTピンを使用して、非同期エッジ・トリガー割り込みを生成できます。この割り込みは、INTCONレジスタのINTEビットをセットすることで有効になります。OPTION\_REGレジスタのINTEDGビットは、割り込みが発生するエッジを決定します。INTEDGビットがセットされると、立ち上がりエッジで割り込みが発生します。INTEDGビットがクリアされると、立ち下がりエッジによって割り込みが発生します。INTCONレジスタのINTFビットは、INTピンに有効なエッジが現れるとセットされます。GIEビットとINTEビットも設定されている場合、プロセッサはプログラムの実行を割り込みベクターにリダイレクトします。

## 7.5 自動文脈(コンテキスト)保存

割り込みを入力すると、戻りPCアドレスがスタックに保存されます。さらに、次のレジスタがシャドウ・レジスタに自動的に保存されます：

- Wレジスタ
- STATUSレジスタ ( $\overline{TO}$ と $\overline{PD}$ を除く)
- BSRレジスタ
- FSRレジスタ
- PCLATHレジスタ

割り込みサービス・ルーチンを終了すると、これらのレジスタは自動的に復元されます。ISR中のこれらのレジスタへの変更はすべて失われます。これらのレジスタのいずれかに変更が必要な場合は、対応するシャドウ・レジスタを変更する必要があります。ISRを終了するとき値が復元されます。シャドウ・レジスタはバンク31で利用可能であり、読み取りと書き込みが可能です。ユーザーのアプリケーションによっては、他のレジスタも保存する必要がある場合があります。

## 7.6 レジスター定義: 割り込み制御

## レジスター7-1: INTCON: 割り込み制御レジスター (インターラプト・コントロール・レジスター)

R/W-0/0	R-0/0						
GIE	PEIE	TMR0IE	INTE	IOCFIE	TMR0IF	INTF	IOCFIF <sup>(1)</sup>
bit 7							bit 0

## 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません    x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている    「0」=ビットがクリアされている

- bit 7      **GIE**: グローバル割り込み許可ビット<sup>(1)</sup>  
1 = すべてのアクティブな割り込みを有効にします  
0 = すべての割り込みを無効にします
- bit 6      **PEIE**: 周辺割り込み許可ビット (ペリフェラル・インタラプト・イネーブル・ビット)<sup>(2)</sup>  
1 = すべてのアクティブな周辺割り込みを有効にします  
0 = すべての周辺割り込みを無効にします
- bit 5      **TMR0IE**: タイマー0オーバーフロー割り込み許可ビット  
1 = タイマー0割り込みを有効にします  
0 = タイマー0割り込みを無効にします
- bit 4      **INTE**: INT外部割り込み許可ビット  
1 = INT外部割り込みを有効にします  
0 = INT外部割り込みを無効にします
- bit 3      **IOCFIE**: 変更時割り込み許可ビット (インタラプト・オン・チェンジ・イネーブル)  
1 = 変更時の割り込みを有効にします  
0 = 変更時の割り込みを無効にします
- bit 2      **TMR0IF**: タイマー0オーバーフロー割り込みフラグ・ビット  
1 = TMR0レジスターがオーバーフローしました  
0 = TMR0レジスターはオーバーフローしませんでした
- bit 1      **INTF**: INT外部割り込みフラグ・ビット  
1 = INT外部割り込みが発生しました  
0 = INT外部割り込みは発生しませんでした
- bit 0      **IOCFIF**: 変更時割り込みの割り込みフラグ・ビット<sup>(3)</sup>  
1 = 変更時割り込みピンの少なくとも1つが状態を変更したとき  
0 = 変更時割り込みピンのいずれも状態が変更されていません

- 注1**: 割り込みフラグ・ビットは、対応する許可ビットまたはグローバル割り込み許可ビット (INTCONレジスタのGIE) の状態に関係なく、割り込み条件が発生するとセットされます。ユーザー・ソフトウェアは、割り込みを有効にする前に、適切な割り込みフラグ・ビットがクリアされていることを確認する必要があります。
- 2**: 周辺割り込みを有効にするには、INTCONレジスターのビットPEIEを設定する必要があります。
- 3**: IOCFIFフラグ・ビットは読み取り専用であり、IOCFレジスター内のすべての変更時割り込みフラグがソフトウェアによってクリアされるとクリアされます。

# PIC16(L)F1508/9

## レジスタ7-2: PIE1: 周辺割り込み許可レジスタ1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0
TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	—	TMR2IE	TMR1IE
bit 7						bit 0	

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている

- bit 7      **TMR1GIE**: タイマー1ゲート割り込み許可ビット  
1 = タイマー1ゲート取得割り込みを有効にします  
0 = タイマー1ゲート取得割り込みを無効にします
- bit 6      **ADIE**: アナログ-デジタル・コンバータ (ADC) 割り込み許可ビット  
1 = ADC割り込みを有効にします  
0 = ADC割り込みを無効にします
- bit 5      **RCIE**: USART受信割り込み許可ビット  
1 = USART受信割り込みを有効にします  
0 = USART受信割り込みを無効にします
- bit 4      **TXIE**: USART送信割り込み許可ビット  
1 = USART送信割り込みを有効にします  
0 = USART送信割り込みを無効にします
- bit 3      **SSP1IE**: 同期シリアル・ポート (MSSP) 割り込み許可ビット  
1 = MSSP割り込みを有効にします  
0 = MSSP割り込みを無効にします
- bit 2      **未実装**: 「0」として読み取ります
- bit 1      **TMR2IE**: TMR2からPR2への一致割り込み許可ビット  
1 = タイマー2からPR2への一致割り込みを有効にします  
0 = タイマー2からPR2への一致割り込みを無効にします
- bit 0      **TMR1IE**: タイマー1オーバーフロー割り込み許可ビット  
1 = タイマー1オーバーフロー割り込みを有効にします  
0 = タイマー1オーバーフロー割り込みを無効にします

**注釈:** 周辺割り込みを有効にするには、INTCONレジスタのビットPEIEを設定する必要があります。

## レジスタ-7-3: PIE2: 周辺割り込み許可レジスタ-2

R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	U-0	U-0
OSFIE	C2IE	C1IE	—	BCL1IE	NCO1IE	—	—
bit 7						bit 0	

## 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている

- bit 7      **OSFIE**: 発振器失敗 (フェイル) 割り込み許可ビット  
1 = 発振器フェイル割り込みを有効にします  
0 = 発振器フェイル割り込みを無効にします
- bit 6      **C2IE**: 比較器 (コンパレータ) C2割り込み許可ビット  
1 = 比較器C2割り込みを有効にします  
0 = 比較器C2割り込みを無効にします
- bit 5      **C1IE**: 比較器C1割り込み許可ビット  
1 = 比較器C1割り込みを有効にします  
0 = 比較器C1割り込みを無効にします
- bit 4      **未実装**: 「0」として読み取ります
- bit 3      **BCL1IE**: MSSPバス衝突割り込み許可ビット  
1 = MSSPバス衝突割り込みを有効にします  
0 = MSSPバス衝突割り込みを無効にします
- bit 2      **NCO1IE**: 数値制御発振器割り込み許可ビット  
1 = NCO割り込みを有効にします  
0 = NCO割り込みを無効にします
- bit 1~0      **未実装**: 「0」として読み取ります

**注釈**: 周辺割り込みを有効にするには、INTCONレジスタのビットPEIEを設定する必要があります。

# PIC16(L)F1508/9

## レジスタ7-4: PIE3: 周辺割り込み許可レジスタ3

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	CLC4IE	CLC3IE	CLC2IE	CLC1IE
bit 7							bit 0

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
[1]=ビットが設定されている      [0]=ビットがクリアされている

- bit 7~4    **未実装**: 「0」として読み取ります
- bit 3      **CLC4IE**: 構成可能なロジック・ブロック4割り込み許可ビット  
1 = CLC4割り込みを有効にします  
0 = CLC4割り込みを無効にします
- bit 2      **CLC3IE**: 構成可能なロジック・ブロック3割り込み許可ビット  
1 = CLC3割り込みを有効にします  
0 = CLC3割り込みを無効にします
- bit 1      **CLC2IE**: 構成可能なロジック・ブロック2割り込み許可ビット  
1 = CLC2割り込みを有効にします  
0 = CLC2割り込みを無効にします
- bit 0      **CLC1IE**: 構成可能なロジック・ブロック1割り込み許可ビット  
1 = CLC1割り込みを有効にします  
0 = CLC1割り込みを無効にします

**注釈**: 周辺割り込みを有効にするには、INTCONレジスタのビットPEIEを設定する必要があります。

## レジスタ-7-5: PIR1: 周辺割り込みリクエスト・レジスタ-1

R/W-0/0	R/W-0/0	R-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0
TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	—	TMR2IF	TMR1IF
bit 7							bit 0

## 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている

- bit 7      **TMR1GIF**: タイマー1ゲート割り込みフラグ・ビット  
1 = 割り込みは保留中です  
0 = 割り込みは保留されていません
- bit 6      **ADIF**: ADC割り込みフラグ・ビット  
1 = 割り込みは保留中です  
0 = 割り込みは保留されていません
- bit 5      **RCIF**: USART受信割り込みフラグ・ビット  
1 = 割り込みは保留中です  
0 = 割り込みは保留されていません
- bit 4      **TXIF**: USART送信割り込みフラグ・ビット  
1 = 割り込みは保留中です  
0 = 割り込みは保留されていません
- bit 3      **SSP1IF**: 同期シリアル・ポート (MSSP) 割り込みフラグ・ビット  
1 = 割り込みは保留中です  
0 = 割り込みは保留されていません
- bit 2      **未実装**: 「0」として読み取ります
- bit 1      **TMR2IF**: タイマー2~PR2割り込みフラグ・ビット  
1 = 割り込みは保留中です  
0 = 割り込みは保留されていません
- bit 0      **TMR1IF**: タイマー1オーバーフロー割り込みフラグ・ビット  
1 = 割り込みは保留中です  
0 = 割り込みは保留されていません

**注釈:** 割り込みフラグ・ビットは、対応する許可ビットまたはグローバル割り込み許可ビット (INTCONレジスタのGIE) の状態に関係なく、割り込み条件が発生するとセットされます。ユーザー・ソフトウェアは、割り込みを有効にする前に、適切な割り込みフラグ・ビットがクリアされていることを確認する必要があります。

# PIC16(L)F1508/9

## レジスタ7-6: PIR2: 周辺割り込みリクエスト・レジスタ2

R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	U-0	U-0
OSFIF	C2IF	C1IF	—	BCL1IF	NCO1IF	—	—
bit 7							bit 0

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
[1]=ビットが設定されている      [0]=ビットがクリアされている

- bit 7      **OSFIF**: 発振器フェイル割り込みフラグ・ビット  
1 = 割り込みは保留中です  
0 = 割り込みは保留されていません
- bit 6      **C2IF**: 比較器 (コンパレータ) C2割り込みフラグ・ビット  
1 = 割り込みは保留中です  
0 = 割り込みは保留されていません
- bit 5      **C1IF**: 比較器C1割り込みフラグ・ビット  
1 = 割り込みは保留中です  
0 = 割り込みは保留されていません
- bit 4      **未実装**: 「0」として読み取ります
- bit 3      **BCL1IF**: MSSPバス衝突割り込みフラグ・ビット  
1 = 割り込みは保留中です  
0 = 割り込みは保留されていません
- bit 2      **NCO1IF**: 数値制御発振器フラグ・ビット  
1 = 割り込みは保留中です  
0 = 割り込みは保留されていません
- bit 1~0      **未実装**: 「0」として読み取ります

**注釈:** 割り込みフラグ・ビットは、対応する許可ビットまたはグローバル割り込み許可ビット (INTCONレジスタのGIE) の状態に関係なく、割り込み条件が発生するとセットされます。ユーザー・ソフトウェアは、割り込みを有効にする前に、適切な割り込みフラグ・ビットがクリアされていることを確認する必要があります。

## レジスタ7-7: PIR3: 周辺割り込みリクエスト・レジスタ3

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	CLC4IF	CLC3IF	CLC2IF	CLC1IF
bit 7							bit 0

### 記録:

**R** =読み取り可能ビット      **W** =書き込み可能ビット      **U** =実装されていないビット、「0」として読み取られる  
**u** =ビットは変更されていません      **x** =ビットは不明です      **-n/n** =PORでの値およびBOR/他のすべてのリセットでの値  
**[1]** =ビットが設定されている      **[0]** =ビットがクリアされている

- bit 7~4    **未実装**: 「0」として読み取ります
- bit 3      **CLC4IF**: 構成可能なロジック・ブロック4割り込みフラグ・ビット  
 1 = 割り込みは保留中です  
 0 = 割り込みは保留されていません
- bit 2      **CLC3IF**: 構成可能なロジック・ブロック3割り込みフラグ・ビット  
 1 = 割り込みは保留中です  
 0 = 割り込みは保留されていません
- bit 1      **CLC2IF**: 構成可能なロジック・ブロック2割り込みフラグ・ビット  
 1 = 割り込みは保留中です  
 0 = 割り込みは保留されていません
- bit 0      **CLC1IF**: 構成可能なロジック・ブロック1割り込みフラグ・ビット  
 1 = 割り込みは保留中です  
 0 = 割り込みは保留されていません

**注釈:** 割り込みフラグ・ビットは、対応する許可ビットまたはグローバル許可ビット (INTCONレジスタのGIE) の状態に関係なく、割り込み条件が発生するとセットされます。ユーザー・ソフトウェアは、割り込みを有効にする前に、適切な割り込みフラグ・ビットがクリアされていることを確認する必要があります。

# PIC16(L)F1508/9

表7-1: 割り込みに関連するレジスタの概要

名前	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ページ上の登録
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	75
OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			154
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	—	TMR2IE	TMR1IE	76
PIE2	OSFIE	C2IE	C1IE	—	BCL1IE	NCO1IE	—	—	77
PIE3	—	—	—	—	CLC4IE	CLC3IE	CLC2IE	CLC1IE	78
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	—	TMR2IF	TMR1IF	79
PIR2	OSFIF	C2IF	C1IF	—	BCL1IF	NCO1IF	—	—	80
PIR3	—	—	—	—	CLC4IF	CLC3IF	CLC2IF	CLC1IF	81

記録: — =実装されていない場所、「0」と読みます。網掛けのセルは割り込みでは使用されません。

## 8.0 パワーダウン・モード(スリープ)

**SLEEP**命令を実行すると、パワーダウンモードに入ります。

スリープ・モードに入ると、次の条件が存在します：

1. スリープ中に操作が有効になっている場合、WDTはクリアされますが、実行を継続します。
2. STATUSレジスタの $\overline{\text{PD}}$ ビットがクリアされます。
3. STATUSレジスタの $\overline{\text{TO}}$ ビットがセットされます。
4. CPUクロックが無効になります。
5. 31kHz LFINTOSCは影響を受けず、それから動作する周辺機器はスリープ状態で動作を継続できます。
6. タイマー1およびタイマー1から動作する周辺機器は、選択されたタイマー1クロック・ソースが次の場合、スリープ状態で動作を継続します：
  - LFINTOSC
  - T1CKI
  - タイマー1発信器
7. 専用FRC発振器が選択されている場合、ADCは影響を受けません。
8. I/Oポートは、**SLEEP**が実行される前のステータスを維持します（ハイ、ロー、またはハイ・インピーダンスの駆動）。
9. WDT以外のリセットは、スリープ・モードの影響を受けません。

スリープ時の周辺動作の詳細については、各章を参照してください。

消費電流を最小限に抑えるには、次の条件を考慮する必要があります：

- I/Oピンはフローティングであってはなりません
- I/Oピンから電流を流入（シンク）する外部回路
- I/Oピンから電流を供給する内部回路
- 内部の弱いプルアップを備えたピンからの電流引き込み
- 31 kHz LFINTOSCを使用するモジュール
- HFINTOSCを使用したCWG、NCO、CLCモジュール

ハイ・インピーダンス入力であるI/Oピンは、フローティング入力によって引き起こされるスイッチング電流を回避するために、外部でVDDまたはVSSにプルする必要があります。

電流を供給する可能性のある内部回路の例には、FVRモジュールが含まれます。このモジュールの詳細については、セクション13.0「固定電圧リファレンス (FVR)」を参照してください。

### 8.1 スリープからの目覚め(ウェイクアップ)

デバイスは、次のいずれかのイベントを通じてスリープからウェイクアップできます：

1. 有効な場合、 $\overline{\text{MCLR}}$ ピンの外部リセット入力
2. 有効になっている場合、BORリセット
3. PORリセット
4. ウォッチドッグ・タイマー（有効な場合）
5. 外部割り込み
6. スリープ中に実行できる周辺機器による割り込み（詳細については、個々の周辺機器を参照してください）

最初の3つのイベントにより、デバイスがリセットされず。最後の3つのイベントは、プログラム実行の継続と見なされます。デバイスのリセットまたはウェイクアップ・イベントが発生したかどうかを判断するには、6.12項「リセットの原因の特定」を参照してください。

**SLEEP**命令の実行中は、次の命令 (PC + 1) がプリフェッチされます。デバイスが割り込みイベントを介してウェイクアップするには、対応する割り込み許可（イネーブル）ビットを有効にする必要があります。ウェイクアップは、GIEビットの状態に関係なく発生します。GIEビットが無効の場合、デバイスは**SLEEP**命令の後の命令で実行を継続します。GIEビットが有効な場合、デバイスは**SLEEP**命令の後に命令を実行し、デバイスは割り込みサービス・ルーチン呼び出しします。**SLEEP**に続く命令の実行が望ましくない場合、ユーザーは**SLEEP**命令の後に**NOP**を持っている必要があります。

WDTは、ウェイクアップのソースに関係なく、デバイスがスリープからウェイクアップするとクリアされます。

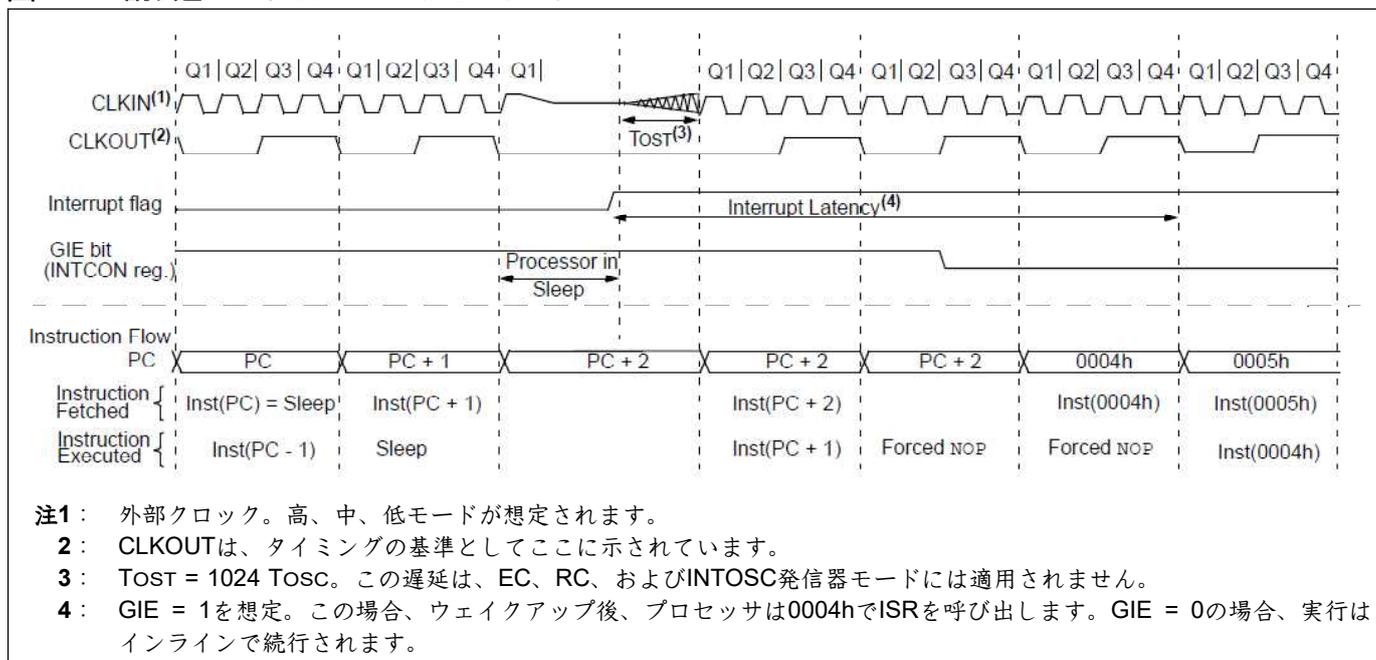
#### 8.1.1 割り込みを使用したウェイクアップ

グローバル割り込みが無効（GIEクリア）で、割り込みソースの割り込み許可ビットと割り込みフラグ・ビットの両方が設定されている場合、次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
  - **SLEEP**命令は**NOP**として実行されます。
  - WDTおよびWDTプリスケアラはクリアされません。
  - STATUSレジスタの $\overline{\text{TO}}$ ビットは設定されません。
  - STATUSレジスタの $\overline{\text{PD}}$ 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
  - **SLEEP**命令は完全に実行されます
  - デバイスはすぐにスリープからウェイクアップします
  - WDTおよびWDTプリスケアラがクリアされます
  - STATUSレジスタの $\overline{\text{TO}}$ ビットが設定されます
  - STATUSレジスタの $\overline{\text{PD}}$ ビットがクリアされます

**SLEEP**命令を実行する前にフラグ・ビットをチェックした場合でも、**SLEEP**命令が完了する前にフラグ・ビットがセットされる可能性があります。**SLEEP**命令が実行されたかどうかを判断するには、 $\overline{\text{PD}}$ ビットをテストします。 $\overline{\text{PD}}$ ビットがセットされている場合、**SLEEP**命令は**NOP**として実行されます。

図8-1: 割り込みによるスリープからのウェイクアップ



## 8.2 低電力スリープ・モード

このデバイスには、内部低ドロップアウト (LDO) 電圧安定器が含まれています。これにより、デバイスI/Oピンは最大5.5Vの電圧で動作し、内部デバイス・ロジックはより低い電圧で動作します。LDOとそれに関連する基準回路は、デバイスがスリープ・モードのときにアクティブのままである必要があります。

低電力スリープ・モードでは、ユーザーはスリープ時の動作電流を最適化できます。低電力スリープ・モードは、VREGCONレジスタのVREGPMビットを設定し、デバイスがスリープ状態のときは常にLDOとリファレンス回路を低電力状態にすることで選択できます。

### 8.2.1 スリープ電流 対 ウェイクアップ時間

デフォルト動作モードでは、LDOとリファレンス回路はスリープ状態の間、通常の構成のままです。すべての回路がアクティブなままであるため、デバイスはスリープ・モードをすばやく終了できます。低電力スリープ・モードでは、スリープから復帰するときに、これらの回路が通常の構成に戻って安定するまでに追加の遅延時間が必要です。

低電力スリープ・モードは、長期間スリープ・モードを維持するアプリケーションに役立ちます。通常モードは、スリープからすばやく頻繁に復帰する必要があるアプリケーションに役立ちます。

### 8.2.2 スリープの周辺機器の使用

スリープ・モードで動作できる一部の周辺機器は、低電力スリープ・モードを選択すると正しく動作しません。これらの周辺機器が有効になっている場合、LDOは通常電源モードのままになります。低電力スリープ・モードは、これらの周辺機器での使用を目的としています：

- ブラウンアウト・リセット (BOR)
- ウォッチドッグ・タイマー (WDT)
- 外部割り込みピン/変更時割り込みピン
- タイマー1 (外部クロック・ソース付き)

相補型波形発生器 (CWG)、数値制御発振器 (NCO) および構成可能論理セル (CLC) モジュールは、HFINTOSC発振器をクロック・ソースまたは入力ソースとして利用できます。特定の条件下で、HFINTOSCがCWG、NCO、またはCLCモジュールで使用するために選択されている場合、HFINTOSCはスリープ中にアクティブのままになります。これは、スリープモード電流に直接影響します。

詳細については、セクション24.5「スリープ中の操作」、25.7「スリープ中の操作」、および26.10「スリープ中の操作」のセクションを参照してください。

**注釈:** PIC16LF1508/9には構成可能なものはありません。低電力スリープモード。PIC16LF1508/9は調整されていないデバイスであり、スリープ状態のときは常に最低電力状態にあり、ウェイクアップ時間のペナルティはありません。このデバイスの最大VDDおよびI/O電圧は、PIC16F1508/9よりも低くなっています。詳細については、セクション29.0「電氣的仕様」を参照してください。

## 8.3 レジスタ定義: 電圧安定器(レギュレーター)制御

### レジスタ-8-1: VREGCON: 電圧レギュレーター制御レジスタ(1)

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-1/1	
—	—	—	—	—	—	VREGPM	予約済み	
bit 7							bit 0	

<b>記録:</b>		
R =読み取り可能ビット	W =書き込み可能ビット	U =実装されていないビット、「0」として読み取られる
u =ビットは変更されていません	x =ビットは不明です	-n/n =PORでの値およびBOR/他のすべてのリセットでの値
[1]=ビットが設定されている	[0]=ビットがクリアされている	

- bit 7~2 **未実装:** 「0」として読み取ります
- bit 1 **VREGPM:** 電圧レギュレータ電源モード選択ビット
  - 1 = スリープ<sup>(2)</sup>で有効になっている低電力スリープ・モード  
スリープ時に最小の電流を引き出し、ウェイクアップを遅くします
  - 0 = スリープ<sup>(2)</sup>で有効になっている通常の電源モード  
スリープでより高い電流を引き出し、より速いウェイクアップ
- bit 0 **予約済み:** 「1」と読みます。このビットセットを維持します。

- 注1: PIC16F1508/9のみ。
- 2: セクション29.0「電氣的仕様」を参照してください。

表8-1: パワーダウン・モードに関連するレジスタの概要

名前	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ページ上の登録
INTCON	GIE	PEIE	TMR0IE	INTE	IOCFIE	TMR0IF	INTF	IOCFIF	75
IOCAF	—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	121
IOCAN	—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	121
IOCAP	—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	121
IOCBF	IOCBF7	IOCBF6	IOCBF5	IOCBF4	—	—	—	—	122
IOCBN	IOCBN7	IOCBN6	IOCBN5	IOCBN4	—	—	—	—	122
IOCBP	IOCBP7	IOCBP6	IOCBP5	IOCBP4	—	—	—	—	122
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	—	TMR2IE	TMR1IE	76
PIE2	OSFIE	C2IE	C1IE	—	BCL1IE	NCO1IE	—	—	77
PIE3	—	—	—	—	CLC4IE	CLC3IE	CLC2IE	CLC1IE	78
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	—	TMR2IF	TMR1IF	78
PIR2	OSFIF	C2IF	C1IF	—	BCL1IF	NCO1IF	—	—	78
PIR3	—	—	—	—	CLC4IF	CLC3IF	CLC2IF	CLC1IF	81
STATUS	—	—	—	$\overline{TO}$	PD	Z	DC	C	19
WDTCON	—	—	WDTPS<4:0>					SWDTEN	88

記録: — =未実装、「0」と読みます。網掛けのセルは、パワーダウンモードでは使用されません。

# PIC16(L)F1508/9

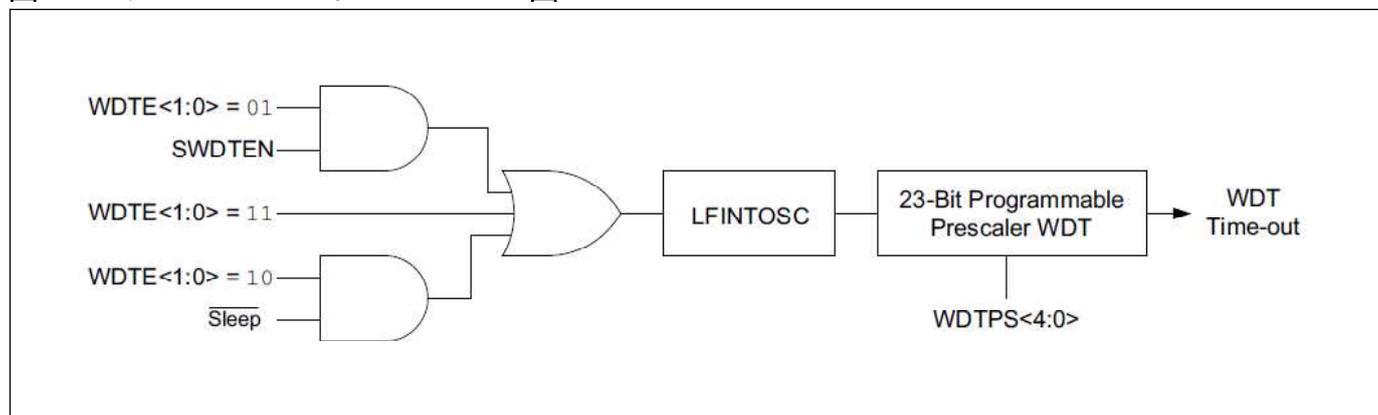
## 9.0 ウォッチドッグ・タイマー (WDT)

ウォッチドッグ・タイマーは、ファームウェアがタイムアウト期間内に**CLRWDTC**命令を発行しない場合にリセットを生成するシステム・タイマーです。ウォッチドッグ・タイマーは通常、予期しないイベントからシステムを回復するために使用されます。

WDTには次の機能があります：

- 独立したクロック・ソース
- 複数の動作モード
  - WDTは常にオンです
  - スリープ中はWDTがオフになっています
  - WDTはソフトウェアによって制御されます
  - WDTは常にオフです
- 設定可能なタイムアウト期間は1msから256秒（公称）
- 複数のリセット条件
- スリープ中の操作

図9-1: ウォッチドッグ・タイマーのブロック図



## 9.1 独立したクロック・ソース

WDTは、31 kHz LFINTOSC内部発振器からタイムベースを導き出します。この章の時間間隔は、1 msの公称間隔に基づいています。LFINTOSCの許容誤差については、セクション29.0「電氣的仕様」を参照してください。

## 9.2 WDTの動作モード

ウォッチドッグ・タイマー・モジュールには、構成ワードのWDTE<1:0>ビットによって制御される4つの動作モードがあります。表9-1を参照してください。

### 9.2.1 WDTが常にオン

構成ワードのWDTEビットが「11」に設定されている場合、WDTは常にオンになっています。

WDT保護は、スリープ中にアクティブになります。

### 9.2.2 WDTがスリープ状態でオフになっている

構成ワードのWDTEビットが「10」に設定されている場合、スリープ時を除いて、WDTはオンになります。

スリープ中はWDT保護はアクティブではありません。

### 9.2.3 ソフトウェアによって制御される

構成ワードのWDTEビットが「01」に設定されている場合、WDTはWDTCONレジスタのSWDTENビットによって制御されます。

WDT保護はSleepによって変更されません。詳細については、表9-1を参照してください。

表9-1: WDT動作モード

WDTE<1:0>	SWDTEN	デバイスモード	WDTモード
11	X	X	アクティブ
10	X	起きている	アクティブ
		スリープ	無効
01	1	X	アクティブ
	0	X	無効
00	X	X	無効

表9-2: WDTのクリアー条件

条件	WDT
WDTE<1:0> = 00	クリアされる
WDTE<1:0> = 01 と、SWDTEN = 0	
WDTE<1:0> = 10 と、スリープに入る	
CLRWDT命令	
発振器の故障（フェイル）が検出された	
スリープを終了 + システム・クロック = T1OSC、EXTRC、INTOSC、EXTCLK	
スリープを終了 + システム・クロック = XT、HS、LP	OSTが終了するまでクリア
INTOSC分周器（IRCFビット）を変更する	影響を受けません

## 9.3 タイムアウト期間

WDTCONレジスタのWDTPSビットは、タイムアウト期間を1 msから256 s（公称）に設定します。リセット後、デフォルトのタイムアウト期間は2秒です。

## 9.4 WDTのクリア

次のいずれかの条件が発生すると、WDTはクリアされます：

- リセット
- CLRWDT命令が実行される
- デバイスがスリープ状態になる
- デバイスがスリープからウェイクアップする
- 発振器が故障する
- WDTが無効になっている
- 発信器（オシレーター）起動タイマー（OST）が実行されている

詳細については、表9-2を参照してください。

## 9.5 スリープ中の操作

デバイスがスリープに入ると、WDTはクリアされます。スリープ中にWDTが有効になっている場合、WDTはカウントを再開します。デバイスがスリープを終了すると、WDTは再びクリアされます。

WDTは、OSTが有効になっている場合、完了するまでクリアのままです。OSTの詳細については、セクション5.0「発信器モジュール（フェイルセーフ・クロック・モニタ付き）」を参照してください。

デバイスがスリープ状態のときにWDTタイムアウトが発生すると、リセットは生成されません。代わりに、デバイスがウェイクアップして動作を再開します。STATUSレジスタのTOビットとPDビットは、イベントを示すために変更されます。PCONレジスタのRWDTPSビットも使用できます。詳細については、セクション3.0「メモリー構成」を参照してください。

# PIC16(L)F1508/9

## 9.6 レジスタ定義: ウォッチドッグ・タイマー制御

### レジスタ9-1: WDTCON: ウォッチドッグ・タイマー制御レジスタ

U-0	U-0	R/W-0/0	R/W-1/1	R/W-0/0	R/W-1/1	R/W-1/1	R/W-0/0
—	—	WDTPS<4:0>				SWDTEN	
bit 7							bit 0

<b>記録:</b>		
R =読み取り可能ビット	W =書き込み可能ビット	U =実装されていないビット、「0」として読み取られる
u =ビットは変更されていません	x =ビットは不明です	-n/n =PORでの値およびBOR/他のすべてのリセットでの値
[1]=ビットが設定されている	[0]=ビットがクリアされている	

bit 7~6 **未実装:** 「0」として読み取ります

bit 5~1 **WDTPS<4:0>:** ウォッチドッグ・タイマー期間選択ビット<sup>(1)</sup>

ビット値 = プリスケール・レート

11111 = 予約済み。最小間隔 (1:32) になります

・  
・  
・

10011 = 予約済み。最小間隔 (1:32) になります

10010 = 1:8388608 (223) (公称間隔256s)

10001 = 1:4194304 (222) (公称間隔128s)

10000 = 1:2097152 (221) (公称間隔64s)

01111 = 1:1048576 (220) (公称間隔32s)

01110 = 1:524288 (219) (公称間隔16s)

01101 = 1:262144 (218) (公称間隔8s)

01100 = 1:131072 (217) (公称間隔4s)

01011 = 1:65536 (公称間隔2s) (リセット値)

01010 = 1:32768 (公称間隔1s)

01001 = 1:16384 (公称間隔512ms)

01000 = 1:8192 (公称間隔256ms)

00111 = 1:4096 (公称間隔128ms)

00110 = 1:2048 (公称間隔64ms)

00101 = 1:1024 (公称間隔32ms)

00100 = 1:512 (公称間隔16ms)

00011 = 1:256 (公称間隔8ms)

00010 = 1:128 (公称間隔4ms)

00001 = 1:64 (公称間隔2ms)

00000 = 1:32 (公称間隔1ms)

bit 0 **SWDTEN:** ウォッチドッグ・タイマー・ビットのソフトウェアの有効化/無効化

WDTE<1:0> = 1xの場合:

このビットは無視されます。

WDTE<1:0> = 01の場合:

1 = WDTがオンになっている

0 = WDTがオフになっている

WDTE<1:0> = 00の場合:

このビットは無視されます。

**注1:** 時間は概算です。WDT時間は、31 kHz LFINTOSCに基づいています。

**表9-3: ウォッチドッグ・タイマーに関連付けられたレジスタの概要**

名前	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ページ上の登録
OSCCON	—	IRCF<3:0>				—	SCS<1:0>		59
PCON	STKOVF	STKUNF	—	RWDT	RMCLR	RI	POR	BOR	68
STATUS	—	—	—	TO	PD	Z	DC	C	19
WDTCON	—	—	WDTPS<4:0>					SWDTEN	88

記録：x = 不明、u = 変更なし、— = 未実装の場所は「0」として読み取られます。網掛けのセルはウォッチドッグ・タイマーでは使用されません

**表9-4: ウォッチドッグ・タイマーを使用した構成ワードの概要**

名前	ビット	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	ページ上の登録
CONFIG1	13:8	—	—	FCMEN	IESO	CLKOUTEN	BOREN<1:0>		—	41
	7:0	CP	MCLRE	PWRTE	WDTE<1:0>		FOSC<2:0>			

記録：— = 実装されていない場所、「0」と読みます。網掛けのセルはウォッチドッグ・タイマーでは使用されません。

## 10.0 フラッシュ・プログラム・メモリー制御

フラッシュ・プログラム・メモリーは、VDDの全範囲で通常の動作中に読み取りおよび書き込みが可能です。プログラム・メモリーは、特殊機能レジスター（SFR）を使用して間接的にアドレス指定されます。プログラム・メモリーへのアクセスに使用されるSFRは次のとおりです：

- PMCON1
- PMCON2
- PMDATL
- PMDATH
- PMADRL
- PMADRH

プログラム・メモリーにアクセスすると、PMDATH:PMDATLレジスター・ペアは読み取り/書き込み用の14ビット・データを保持する2バイト・ワードを形成し、そしてPMADRH:PMADRLレジスター・ペアは、読み取られるプログラム・メモリー位置の15ビット・アドレスを保持する2バイト・ワードを形成します。

書き込み時間は、オンチップ・タイマーによって制御されます。書き込み/消去電圧は、デバイスの動作電圧範囲で動作する定格のオンチップ・チャージポンプによって生成されます。

フラッシュ・プログラムのメモリーは、2つの方法で保護できます。コード保護（構成ワードのCPビット）および書き込み保護（構成ワードのWRT<1:0>ビット）による。

コード保護（ $\overline{CP} = 0$ ）<sup>(1)</sup>は、外部デバイス・プログラマーを介したフラッシュ・プログラム・メモリーへのアクセス、読み取り、および書き込みを無効にします。コード保護は、自己書き込みおよび消去機能には影響しません。コード保護は、デバイス・プログラマーがデバイスに対して一括消去を実行し、すべてのフラッシュ・プログラム・メモリー、構成ビット、およびユーザーIDをクリアすることによってのみリセットできます。

書き込み保護は、ビットWRT<1:0>で定義されているように、フラッシュ・プログラム・メモリーの一部またはすべてへの自己書き込みおよび消去を禁止します。書き込み保護は、デバイス・プログラマーがデバイスを読み取り、書き込み、または消去する機能には影響しません。

**注1：** フラッシュ・プログラム・メモリー・アレイ全体のコード保護は、構成ワードのCPビットをクリアすることで有効になります。

### 10.1 PMADRLとPMADRHレジスター

PMADRH:PMADRLレジスター・ペアは、最大32Kワードのプログラム・メモリーをアドレス指定できます。プログラム・アドレス値を選択すると、アドレスのMSBがPMADRHレジスターに書き込まれ、LSBがPMADRLレジスターに書き込まれます。

### 10.1.1 PMCON1とPMCON2レジスター

PMCON1は、フラッシュ・プログラムのメモリー・アクセス用の制御レジスターです。

制御ビットRDとWRは、それぞれ読み取りと書き込みを開始します。これらのビットは、ソフトウェアではクリアできず、設定するだけです。これらは、読み取りまたは書き込み操作の完了時にハードウェアによってクリアされます。ソフトウェアでWRビットをクリアできないため、書き込み操作が誤って早期に終了するのを防ぐことができます。

WRENビットを設定すると、書き込み操作を実行できます。電源投入時、WRENビットはクリアされています。WRERRビットは、通常の動作中にリセットによって書き込み動作が中断されるとセットされます。このような状況では、リセット後、ユーザーはWRERRビットをチェックして、適切なエラー処理ルーチンを実行できます。

PMCON2レジスターは書き込み専用レジスターです。PMCON2レジスターを読み取ろうとすると、すべて「0」が返されます。

プログラム・メモリーへの書き込みを有効にするには、特定のパターン（ロック解除シーケンス）をPMCON2レジスターに書き込む必要があります。必要なロック解除シーケンスは、プログラム・メモリーの書き込みラッチとフラッシュ・プログラム・メモリーへの不注意な書き込みを防ぎます。

## 10.2 フラッシュ・プログラム・メモリーの概要

消去およびプログラミング操作のためのフラッシュ・プログラムのメモリー構造を理解することが重要です。フラッシュ・プログラム・メモリーは行(rows)に配置されています。行は、固定数の14ビット・プログラム・メモリー・ワードで構成されます。行は、ユーザー・ソフトウェアで消去できる最小サイズです。

行(rows)が消去された後、ユーザーはこの行の全部または一部を再プログラムできます。プログラム・メモリー行に書き込まれるデータは、14ビット幅のデータ書き込みラッチに書き込まれます。これらの書き込みラッチは、ユーザーが直接アクセスすることはできませんが、PMDATH:PMDATLレジスター・ペアへの順次書き込みを介してロードできます。

**注釈：** ユーザーが以前にプログラムされた行の一部のみを変更する場合は、消去する前に、行全体の内容を読み取ってRAMに保存する必要があります。次に、新しいデータと保持されたデータを書き込みラッチに書き込んで、フラッシュプログラム・メモリーの行を再プログラムできます。ただし、プログラムされていない場所は、最初に行を消去せずに書き込むことができます。この場合、以前にプログラムされた他の場所を保存して書き換える必要はありません。

フラッシュ・プログラム・メモリーの消去行サイズと書き込みラッチの数については、表10-1を参照してください。

**表10-1: デバイスによるフラッシュ・メモリーの編成**

デバイス	行の消去 (ワード)	書<ラッチ (ワード)
PIC16(L)F1508	32	32
PIC16(L)F1509		

## 10.2.1 フラッシュ・プログラム・メモリーを読む

プログラムのメモリー位置を読み取るには、ユーザーは：

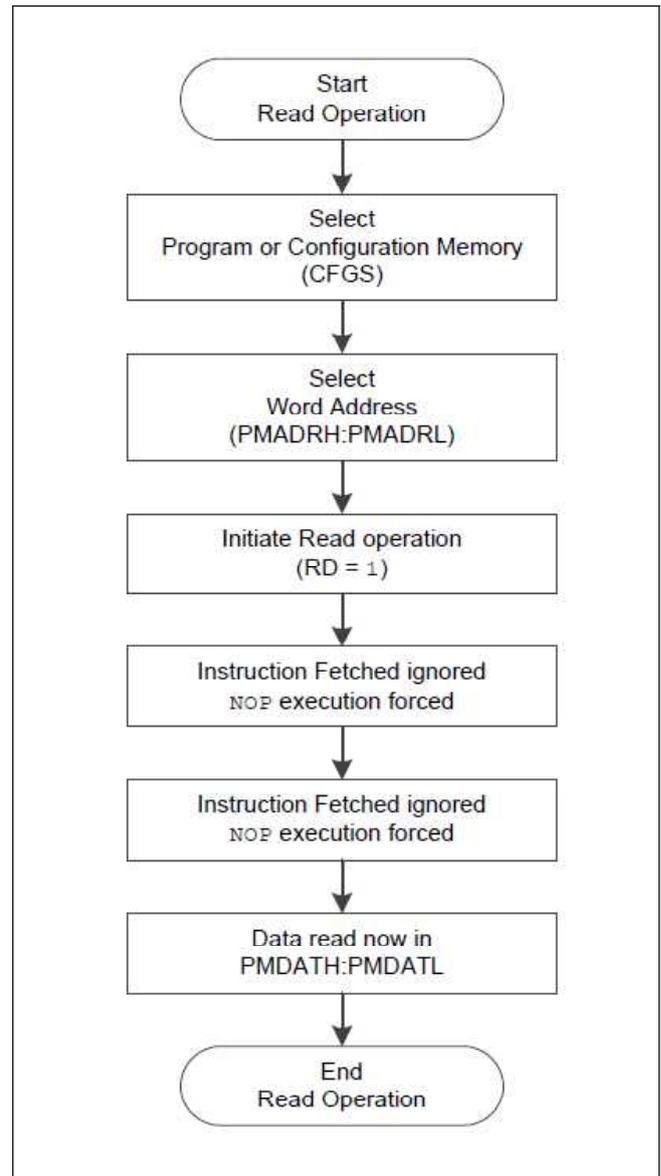
1. 目的のアドレスをPMADRH:PMADRLレジスター・ペアに書き込みます。
2. PMCON1レジスターのCFGFSビットをクリアします。
3. 次に、PMCON1レジスターの制御ビットRDをセットします。

読み取り制御ビットが設定されると、プログラム・メモリー・フラッシュ・コントローラーは2番目の命令サイクルを使用してデータを読み取ります。これにより、「BSFPMCON1、RD」命令の直後の2番目の命令が無視されます。データは、次のサイクルであるPMDATH:PMDATLレジスター・ペアで利用できます。したがって、次の手順では2バイトとして読み取ることができます。

PMDATH:PMDATLレジスター・ペアは、別の読み取りが行われるまで、またはユーザーが書き込まれるまで、この値を保持します。

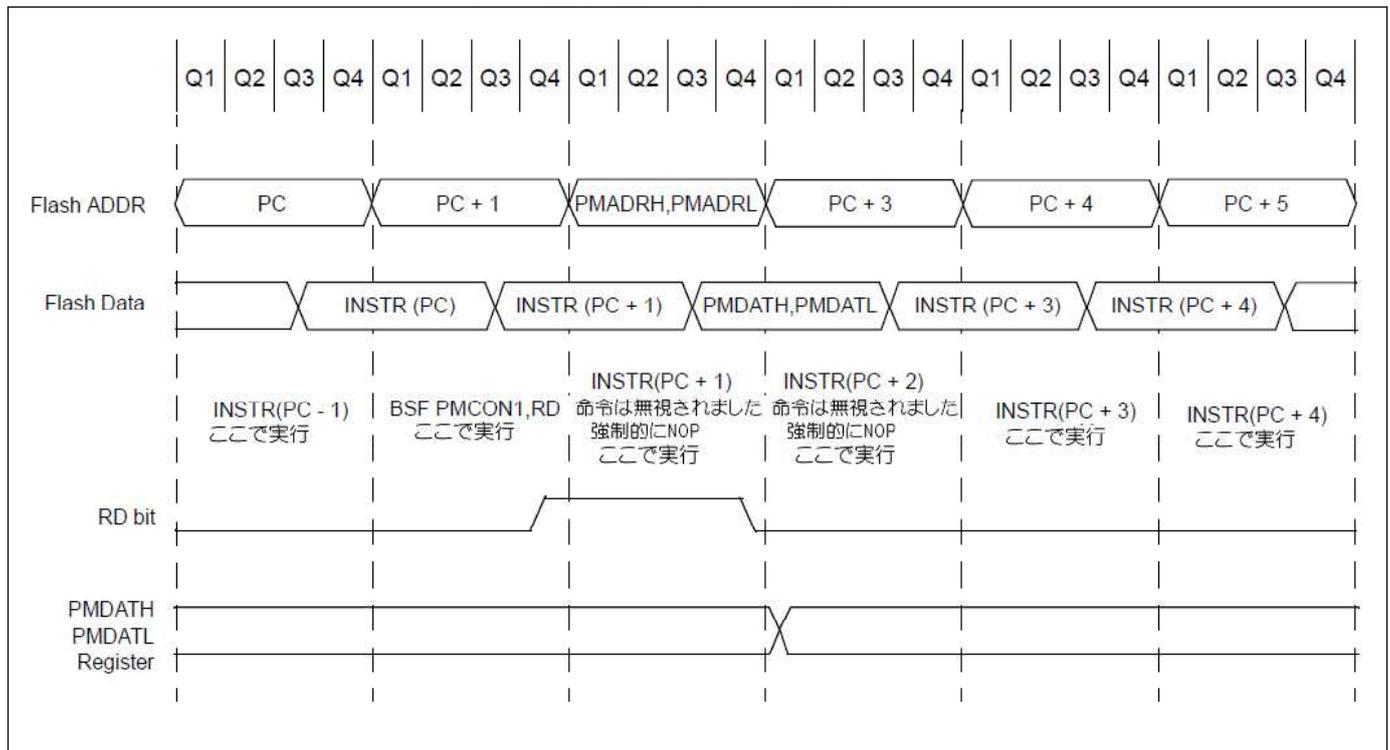
**注釈：** プログラム・メモリーの読み取りに続く2つの命令は、NOPである必要があります。これにより、RDビットがセットされた後、ユーザーが次の命令で2サイクルの命令を実行するのを防ぐことができます。

図10-1: フラッシュ・プログラム・メモリーの読み取りフローチャート



# PIC16(L)F1508/9

図10-2: フラッシュ・プログラム・メモリー の読み取りサイクルの実行



例10-1: フラッシュ・プログラム・メモリー の読み取り

```

* This code block will read 1 word of program
* memory at the memory address:
  PROG_ADDR_HI : PROG_ADDR_LO
* data will be returned in the variables;
* PROG_DATA_HI, PROG_DATA_LO

BANKSEL  PMADRL      ; Select Bank for PMCON registers
MOVLW   PROG_ADDR_LO ;
MOVWF   PMADRL      ; Store LSB of address
MOVLW   PROG_ADDR_HI ;
MOVWF   PMADRH      ; Store MSB of address
BCF     PMCON1,CFG5 ; Do not select Configuration Space
BSF     PMCON1,RD   ; Initiate read
NOP     ; Ignored (Figure 10-2)
NOP     ; Ignored (Figure 10-2)
MOVF    PMDATL,W    ; Get LSB of word
MOVWF   PROG_DATA_LO ; Store in user location
MOVF    PMDATH,W    ; Get MSB of word
MOVWF   PROG_DATA_HI ; Store in user location
    
```

## 10.2.2 フラッシュ・メモリーのロック解除シーケンス

ロック解除シーケンスは、フラッシュ・プログラム・メモリーを意図しない自己書き込みプログラミングまたは消去から保護するメカニズムです。次の操作を正常に完了するには、シーケンスを中断せずに実行して完了する必要があります：

- 行(Row)の消去
- プログラム・メモリー書き込みラッチのロード
- プログラム・メモリーへの書き込みラッチをプログラム・メモリーに書き込む
- ユーザーIDへのプログラム・メモリー書き込みラッチの書き込み

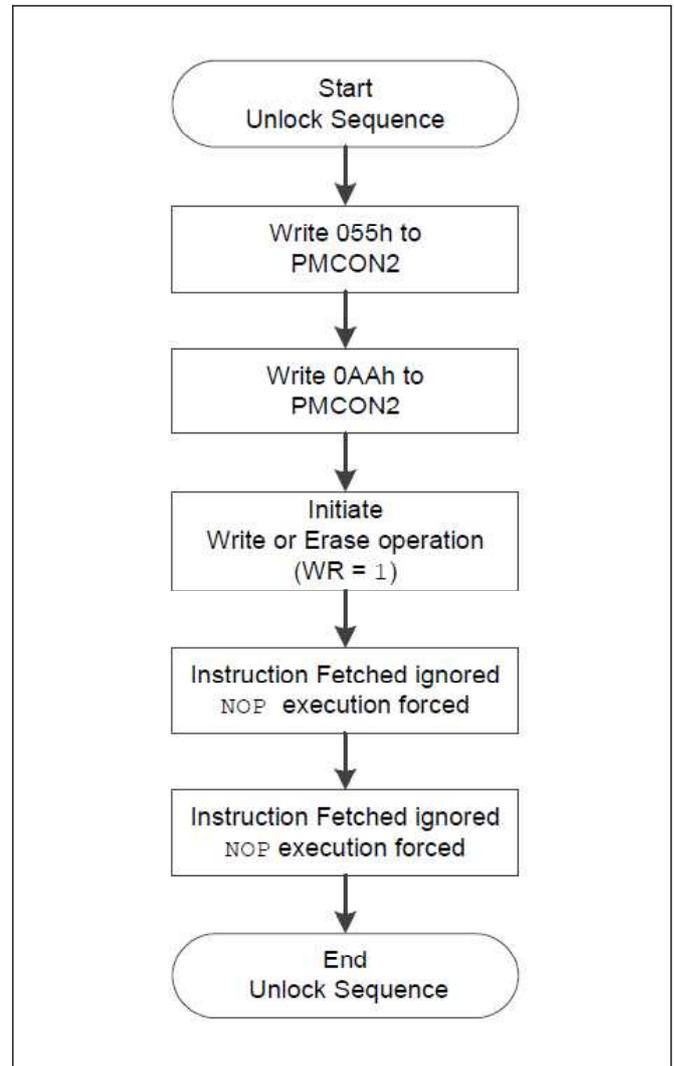
ロック解除シーケンスは、次の手順で構成されます：

1. PMCON2に55hを書き込みます
2. AAhをPMCON2に書き込みます
3. PMCON1のWRビットを設定します
4. **NOP**命令
5. **NOP**命令

WRビットが設定されると、プロセッサは常に2つの**NOP**命令を強制します。行の消去またはプログラム行の操作が実行されている場合、プロセッサは、操作が完了するまで内部操作（通常は2ms）を停止し、次の命令から再開します。操作がプログラム・メモリー書き込みラッチをロードしているとき、プロセッサは常に2つの**NOP**命令を強制し、次の命令で中断されることなく続行します。

ロック解除シーケンスを中断してはならないため、グローバル割り込みは、ロック解除シーケンスの前に無効にし、ロック解除シーケンスの完了後に再度有効にする必要があります。

図10-3: フラッシュ・プログラム・メモリーのロック解除シーケンス・フローチャート



# PIC16(L)F1508/9

## 10.2.3 フラッシュ・プログラム・メモリーの消去

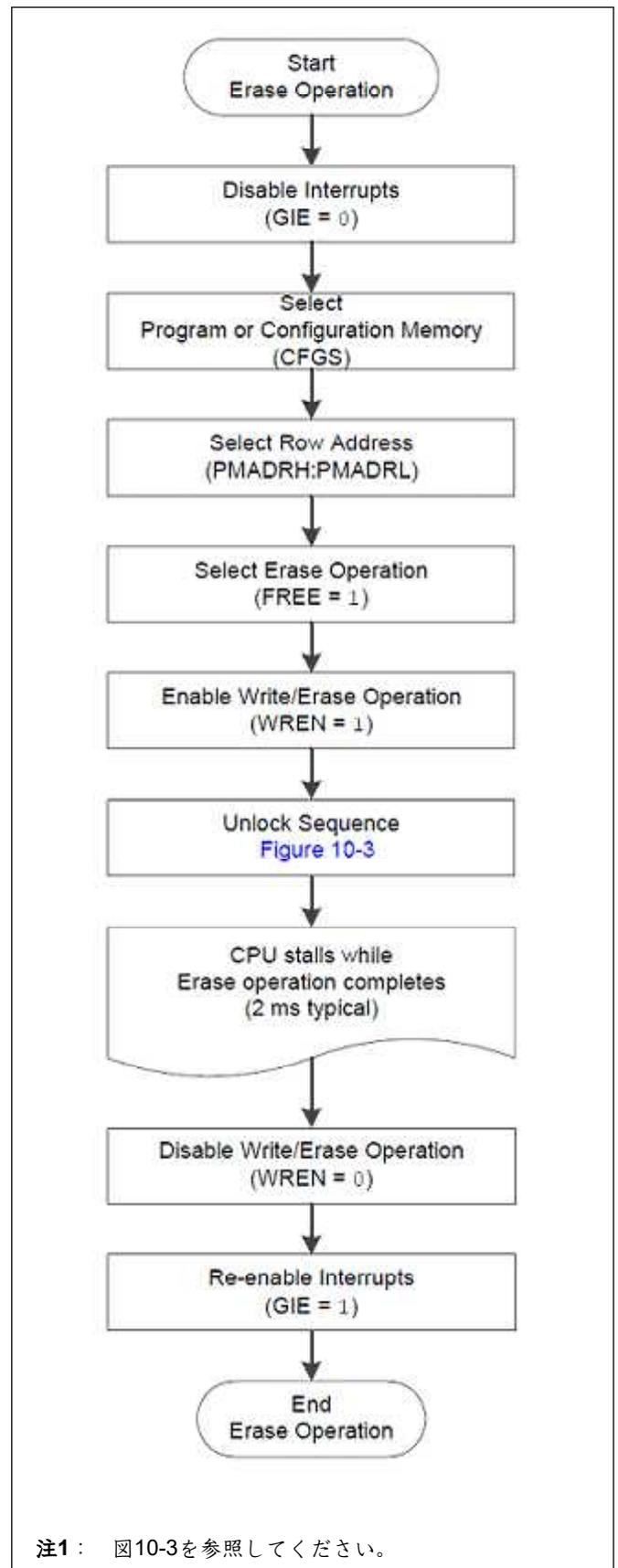
コードの実行中、プログラム・メモリーは行(row)によるみ消去できます。行を消去するには：

1. PMADRH:PMADRLレジスター・ペアに、消去する行内の任意のアドレスをロードします。
2. PMCON1レジスターのCFGFSビットをクリアします。
3. PMCON1レジスターのFREEビットとWRENビットを設定します。
4. 55h、次にAAhをPMCON2（フラッシュ・プログラミング・ロック解除シーケンス）に書き込みます。
5. PMCON1レジスターの制御ビットWRをセットして、消去動作を開始します。

例10-2を参照してください。

「BSF PMCON1,WR」命令の後、プロセッサは消去操作を設定するために2サイクルを必要とします。ユーザーは、WRビット・セット命令の直後に2つのNOP命令を配置する必要があります。プロセッサは、通常2msの消去時間の間内部操作を停止します。クロックと周辺機器は引き続き動作するため、これはスリープ・モードではありません。消去サイクル後、プロセッサはPMCON1書き込み命令の後の3番目の命令で動作を再開します。

図10-4: フラッシュ・プログラム・メモリー消去フローチャート



注1： 図10-3を参照してください。

## 例10-2: プログラム・メモリーの1行を消去する

```

; This row erase routine assumes the following:
; 1. A valid address within the erase row is loaded in ADDRH:ADDRL
; 2. ADDRH and ADDRL are located in shared data memory 0x70 - 0x7F (common RAM)

BCF  INTCON,GIE    ; Disable ints so required sequences will execute properly
BANKSEL  PMADRL
MOVF  ADDRL,W      ; Load lower 8 bits of erase address boundary
MOVWF  PMADRL
MOVF  ADDRH,W      ; Load upper 6 bits of erase address boundary
MOVWF  PMADRH
BCF  PMCON1,CFG5   ; Not configuration space
BSF  PMCON1,FREE   ; Specify an erase operation
BSF  PMCON1,WREN   ; Enable writes

MOV LW  55h        ; Start of required sequence to initiate erase
MOV WF  PMCON2     ; Write 55h
MOV LW  0AAh       ;
MOV WF  PMCON2     ; Write AAh
BSF  PMCON1,WR     ; Set WR bit to begin erase
NOP                                           ; NOP instructions are forced as processor starts
NOP                                           ; row erase of program memory.
;
; The processor stalls until the erase process is complete
; after erase processor continues with 3rd instruction
BCF  PMCON1,WREN   ; Disable writes
BSF  INTCON,GIE    ; Enable interrupts

```

必要なシーケンス

## 10.2.4 フラッシュ・プログラム・メモリーへの書き込み

プログラム・メモリーは、次の手順を使用してプログラムされます：

1. プログラムする行のPMADRH:PMADRLにアドレスをロードします。
2. 各書き込みラッチにデータをロードします。
3. プログラミング操作を開始します。
4. すべてのデータが書き込まれるまで、手順1~3を繰り返します。

プログラム・メモリーに書き込む前に、書き込むワードを消去するか、事前に書き込む必要があります。プログラム・メモリーは、一度に1行しか消去できません。書き込みの開始時に自動消去は発生しません。

プログラム・メモリーは、一度に1つ以上のワードを書き込むことができます。一度に書き込まれるワードの最大数は、書き込みラッチの数と同じです。詳細については、図10-5 (32個の書き込みラッチを使用したプログラム・メモリーへの行書き込み)を参照してください。

書き込みラッチは、PMADRH:PMADRLの上位10ビット (PMADRH<6:0>:PMADRL<7:5>)とPMADRLの下部5ビット (PMADRL<4:0>)で定義されるフラッシュ行アドレス境界に揃えられ、(PMADRL<4:0>)ロードされている書き込みラッチを決定します。書き込み操作はこれらの境界を越えません。プログラム・メモリーの書き込み操作が完了すると、書き込みラッチのデータがリセットされ、0x3FFFが含まれるようになります。

書き込みラッチをロードし、プログラム・メモリーの行をプログラムするには、次の手順を実行する必要があります。これらのステップは2つの部分に分かれています。まず、各書き込みラッチに、 $LWLO = 1$  のロック解除シーケンスを使用してPMDATH:PMDATLからのデータがロードされます。書き込みラッチにロードされる最後のワードの準備ができると、 $LWLO$ ビットがクリアされ、ロック解除シーケンスが実行されます。これによりプログラミング操作が開始され、すべてのラッチがフラッシュ・プログラム・メモリーに書き込まれます。

**注釈：** 書き込みラッチにデータをロードしたり、フラッシュ・プログラミング操作を開始したりするには、特別なロック解除シーケンスが必要です。ロック解除シーケンスが中断された場合、ラッチまたはプログラム・メモリーへの書き込みは開始されません。

1. PMCON1レジスターのWRENビットをセットします。
2. PMCON1レジスターのCFGSビットをクリアします。
3. PMCON1レジスターのLWLOビットをセットします。PMCON1レジスターのLWLOビットが「1」の場合、書き込みシーケンスは書き込みラッチのみをロードし、フラッシュ・プログラム・メモリーへの書き込みを開始しません。
4. PMADRH:PMADRLレジスター・ペアに、書き込む場所のアドレスをロードします。
5. PMDATH:PMDATLレジスター・ペアに、書き込むプログラム・メモリー・データをロードします。
6. ロック解除シーケンスを実行します (セクション10.2.2「フラッシュ・メモリーのロック解除シーケンス」)。これで書き込みラッチがロードされます。
7. PMADRH:PMADRLレジスター・ペアをインクリメントして、次の場所を指します。
8. 最後の書き込みラッチを除くすべてがロードされるまで、手順5~7を繰り返します。
9. PMCON1レジスターのLWLOビットをクリアします。PMCON1レジスターのLWLOビットが「0」の場合、書き込みシーケンスはフラッシュ・プログラム・メモリーへの書き込みを開始します。
10. PMDATH:PMDATLレジスター・ペアに、書き込むプログラム・メモリー・データをロードします。
11. ロック解除シーケンスを実行します (セクション10.2.2「フラッシュ・メモリーのロック解除シーケンス」)。プログラム・メモリーのラッチ内容全体がフラッシュ・プログラム・メモリーに書き込まれます。

**注釈：** プログラム・メモリーの書き込みラッチは、すべての書き込みまたは消去操作が完了すると、ブランク状態 (0x3FFF) にリセットされます。その結果、すべてのプログラム・メモリー書き込みラッチをロードする必要はありません。アンロードされたラッチはブランク状態のままになります。

完全な書き込みシーケンスの例を例10-3に示します。初期アドレスはPMADRH:PMADRLレジスター・ペアにロードされ、データは間接アドレス指定を使用してロードされます。

図10-5: 32個の書き込みラッチを使用したフラッシュ・プログラム・メモリーへのブロック書き込み

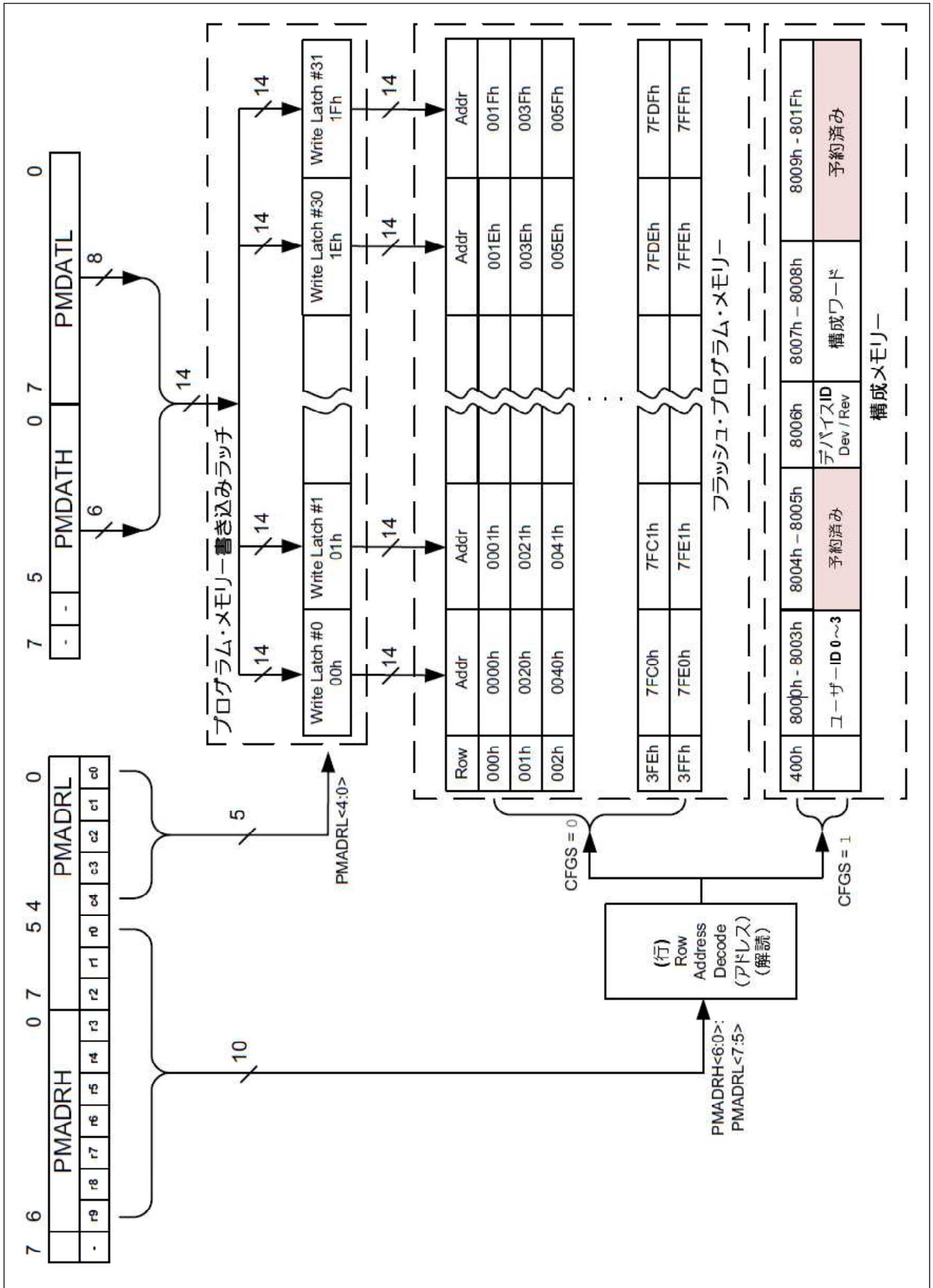
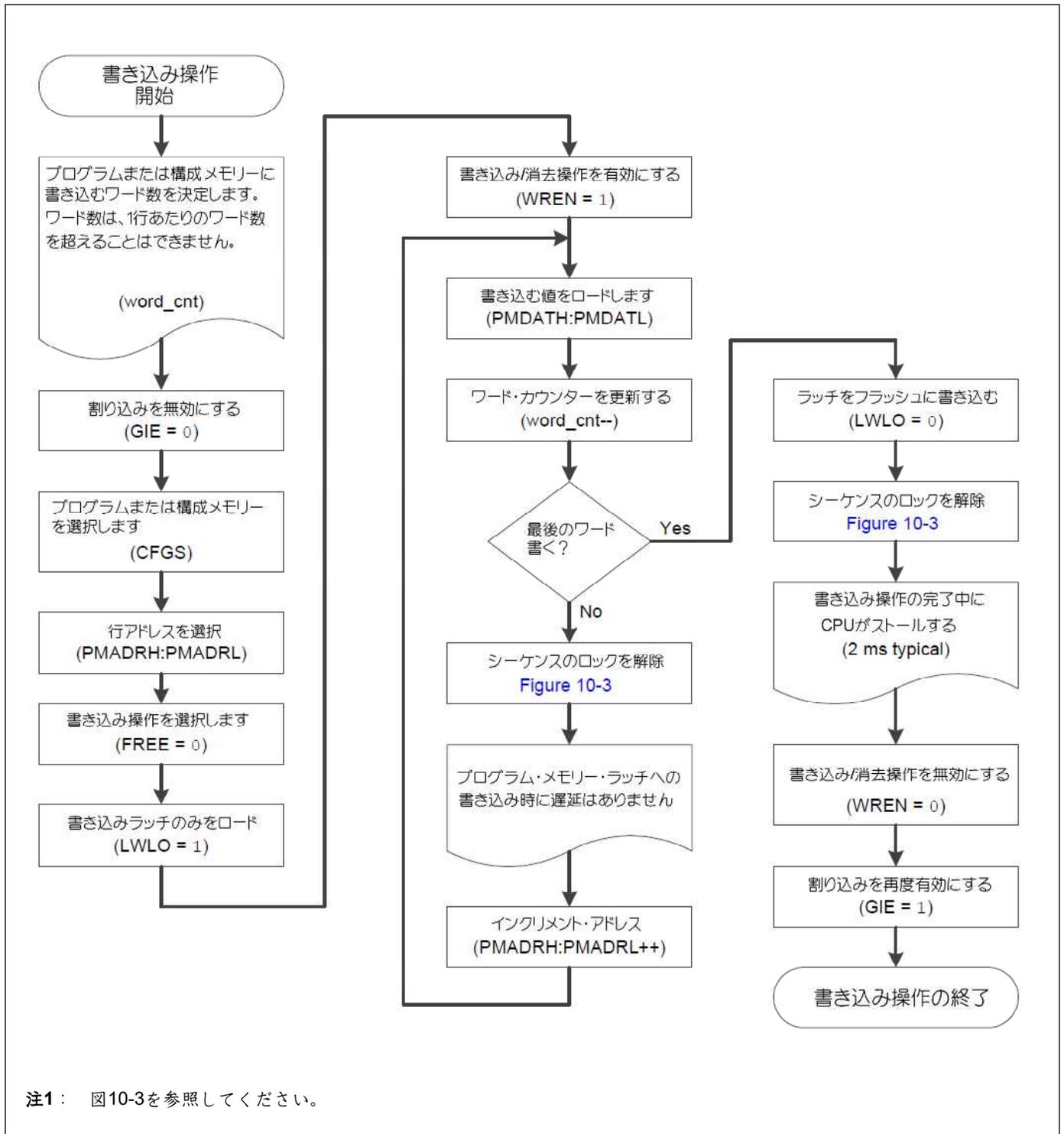


図10-6: フラッシュ・プログラム・メモリーの書き込みフローチャート



注1: 図10-3を参照してください。

## 例10-3: フラッシュ・プログラム・メモリーへの書き込み(32個の書き込みラッチ)

```

; This write routine assumes the following:
; 1. 64 bytes of data are loaded, starting at the address in DATA_ADDR
; 2. Each word of data to be written is made up of two adjacent bytes in DATA_ADDR,
; stored in little endian format
; 3. A valid starting address (the Least Significant bits = 00000) is loaded in ADDRH:ADDRL
; 4. ADDRH and ADDRL are located in shared data memory 0x70 - 0x7F (common RAM)
;
BCF      INTCON,GIE      ; Disable ints so required sequences will execute properly
BANKSEL  PMADRH          ; Bank 3
MOVF     ADDRH,W         ; Load initial address
MOVWF    PMADRH          ;
MOVF     ADDRL,W        ;
MOVWF    PMADRL         ;
MOVLW   LOW DATA_ADDR  ; Load initial data address
MOVWF    FSR0L          ;
MOVLW   HIGH DATA_ADDR ; Load initial data address
MOVWF    FSR0H          ;
BCF      PMCON1,CFG5     ; Not configuration space
BSF      PMCON1,WREN     ; Enable writes
BSF      PMCON1,LWLO     ; Only Load Write Latches

LOOP
MOVW     FSR0++          ; Load first data byte into lower
MOVWF    PMDATL         ;
MOVW     FSR0++          ; Load second data byte into upper
MOVWF    PMDATH         ;

MOVW     PMADRL,W        ; Check if lower bits of address are '00000'
XORLW   0x1F            ; Check if we're on the last of 32 addresses
ANDLW   0x1F            ;
BTFSC   STATUS,Z        ; Exit if last of 32 words,
GOTO    START_WRITE     ;

MOVW     55h             ; Start of required write sequence:
MOVWF    PMCON2          ; Write 55h
MOVW     0AAh           ;
MOVWF    PMCON2          ; Write AAh
BSF      PMCON1,WR      ; Set WR bit to begin write
NOP      ; NOP instructions are forced as processor
          ; loads program memory write latches
NOP      ;
INCF     PMADRL,F        ; Still loading latches Increment address
GOTO    LOOP            ; Write next latches

START_WRITE
BCF      PMCON1,LWLO     ; No more loading latches - Actually start Flash program
          ; memory write

MOVW     55h             ; Start of required write sequence:
MOVWF    PMCON2          ; Write 55h
MOVW     0AAh           ;
MOVWF    PMCON2          ; Write AAh
BSF      PMCON1,WR      ; Set WR bit to begin write
NOP      ; NOP instructions are forced as processor writes
          ; all the program memory write latches simultaneously
          ; to program memory.
NOP      ; After NOPs, the processor
          ; stalls until the self-write process is complete
          ; after write processor continues with 3rd instruction

BCF      PMCON1,WREN     ; Disable writes
BSF      INTCON,GIE     ; Enable interrupts

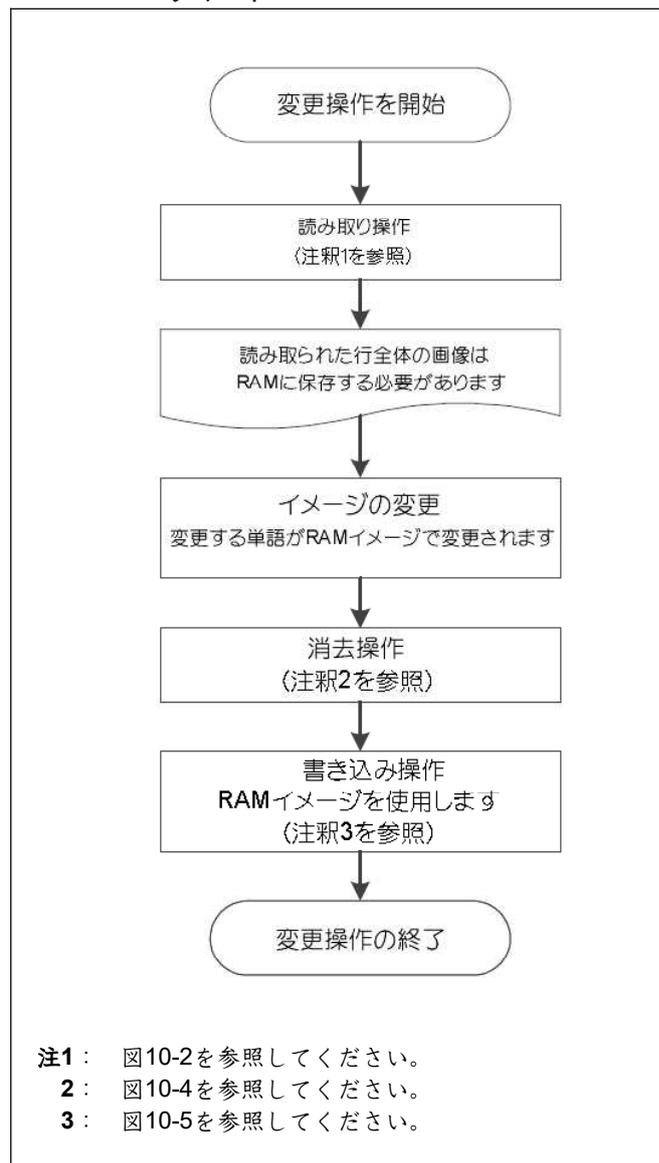
```

## 10.3 フラッシュ・プログラム・メモリーの変更

プログラム・メモリー行(row)の既存のデータを変更し、その行内のデータを保持する必要がある場合は、最初にデータを読み取ってRAMイメージに保存する必要があります。プログラム・メモリーは、次の手順を使用して変更されます：

1. 変更する行の開始アドレスをロードします。
2. 行から既存のデータをRAMイメージに読み込みます。
3. プログラム・メモリーに書き込まれる新しいデータを含むようにRAMイメージを変更します。
4. 書き換える行の開始アドレスをロードします。
5. プログラムメモリー行を消去します。
6. RAMイメージからのデータを書き込みラッチにロードします。
7. プログラミング操作を開始します。

図10-7: フラッシュ・プログラム・メモリーの変更フローチャート



## 10.4 ユーザーID、デバイスID、および構成ワード・アクセス

PMCON1レジスタのCFGS = 1 の場合、プログラム・メモリーにアクセスする代わりに、ユーザーID、デバイスID/リビジョン（改訂）ID、および構成ワードにアクセスできます。これは、PC<15> = 1 が指す領域ですが、すべてのアドレスにアクセスできるわけではありません。読み取りと書き込みで異なるアクセスが存在する場合があります。表10-2を参照してください。

表10-2にリストされているパラメーター以外のアドレスで読み取りアクセスが開始されると、PMDATH:PMDATLレジスタ一ペアがクリアされ、「0」が読み戻されます。

**表10-2: ユーザーID、デバイスID、および構成ワードへのアクセス (CFGS = 1)**

アドレス	機能	読み取りアクセス	書き込みアクセス
8000h ~ 8003h	ユーザーID	Yes	Yes
8006h	デバイスID/リビジョンID	Yes	No
8007h ~ 8008h	構成ワード1と2	Yes	No

### 例10-4: 構成ワードとデバイスIDアクセス

```
* This code block will read 1 word of program memory at the memory address:
* PROG_ADDR_LO (must be 00h-08h) data will be returned in the variables;
* PROG_DATA_HI, PROG_DATA_LO
```

```
BANKSEL    PMADRL    ; Select correct Bank
MOVLW     PROG_ADDR_LO    ;
MOVWF     PMADRL    ; Store LSB of address
CLRF      PMADRH    ; Clear MSB of address

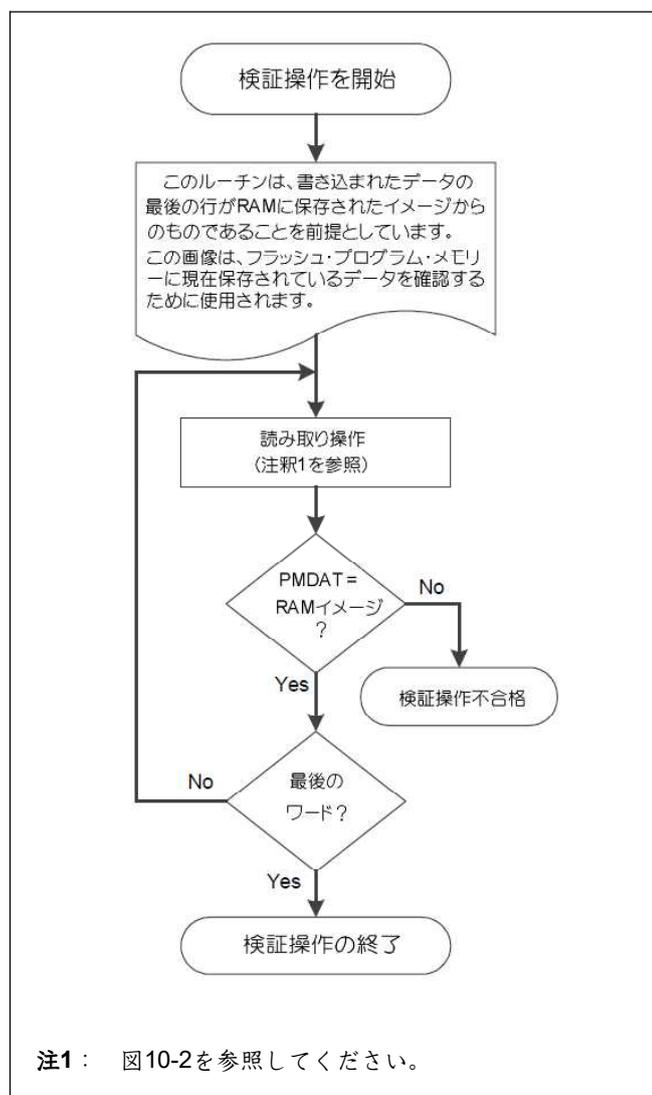
BSF       PMCON1,CFGS    ; Select Configuration Space
BCF       INTCON,GIE     ; Disable interrupts
BSF       PMCON1,RD      ; Initiate read
NOP       ; Executed (See Figure 10-2)
NOP       ; Ignored (See Figure 10-2)
BSF       INTCON,GIE     ; Restore interrupts

MOVF     PMDATL,W        ; Get LSB of word
MOVWF    PROG_DATA_LO    ; Store in user location
MOVF     PMDATH,W        ; Get MSB of word
MOVWF    PROG_DATA_HI    ; Store in user location
```

## 10.5 書き込みベリファイ(検証)

プログラム・メモリーへの書き込みが意図した値と一致することを確認することは、優れたプログラミング手法と見なされます。プログラム・メモリーはフルページとして保存されるため、保存されたプログラム・メモリーの内容は、最後の書き込みが完了した後にRAMに保存された目的のデータと比較されます。

図10-8: フラッシュ・プログラム・メモリーのベリファイ・フローチャート



## 10.6 レジスタ定義: フラッシュ・プログラム・メモリー制御

### レジスタ10-1: PMDATL: プログラム・メモリー・データ・ロー・バイト・レジスタ

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
PMDAT<7:0>							
bit 7							bit 0

**記録:**

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている

bit 7~0    **PMDAT<7:0>**: プログラム・メモリーの最下位ビットの読み取り/書き込み値

### レジスタ10-2: PMDATH: プログラム・メモリー・データ・ハイ・バイト・レジスタ

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
PMDAT<13:8>							
bit 7							bit 0

**記録:**

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている

bit 7~6    **未実装**: 「0」として読み取ります

bit 5~0    **PMDAT<13:8>**: プログラム・メモリーの最上位ビットの読み取り/書き込み値

### レジスタ10-3: PMADRL: プログラム・メモリー・アドレス・ロー・バイト・レジスタ

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
PMADR<7:0>							
bit 7							bit 0

**記録:**

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている

bit 7~0    **PMADR<7:0>**: プログラム・メモリー・アドレスの最下位ビットを指定します

### レジスタ10-4: PMADRH: プログラム・メモリー・アドレス・ハイ・バイト・レジスタ

U-1	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—(1)	PMADR<14:8>						
bit 7							bit 0

**記録:**

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている

bit 7        **未実装**: 「1」と読みます

bit 6~0    **PMADR<14:8>**: プログラム・メモリー・アドレスの最上位ビットを指定します

注1:    未実装、「1」と読みます。

# PIC16(L)F1508/9

## レジスタ10-5: PMCON1: プログラム・メモリー制御1レジスタ

U-1	R/W-0/0	R/W-0/0	R/W/HC-0/0	R/W/HC-x/q <sup>(2)</sup>	R/W-0/0	R/S/HC-0/0	R/S/HC-0/0
— <sup>(1)</sup>	CFGS	LWLO	FREE	WRERR	WREN	WR	RD
bit 7							bit 0

### 記録:

R =読み取り可能ビット	W =書き込み可能ビット	U =実装されていないビット、「0」として読み取られる
u =ビットは変更されていません	x =ビットは不明です	-n/n =PORでの値およびBOR/他のすべてのリセットでの値
[1]=ビットが設定されている	[0]=ビットがクリアされている	HC =ビットはハードウェアによってクリアされます

- bit 7 **未実装**: 「1」と読みます
- bit 6 **CFGS**: 構成選択ビット  
 1 = アクセス構成、ユーザーIDおよびデバイスIDレジスタ  
 0 = フラッシュ・プログラム・メモリーにアクセスする
- bit 5 **LWLO**: (ロード・ライト・ラッチズ・オンリー) 書き込みラッチのみのロード・ビット<sup>(3)</sup>  
 1 = 次のWRコマンドでは、アドレス指定されたプログラム・メモリー書き込みラッチのみがロード/更新されます。  
 0 = アドレス指定されたプログラム・メモリー書き込みラッチがロード/更新され、すべてのプログラム・メモリー書き込みラッチの書き込みが次のWRコマンドで開始されます。
- bit 4 **FREE**: プログラム・フラッシュ消去許可ビット  
 1 = 次のWRコマンドで消去操作を実行します (完了時にハードウェアがクリアされます)  
 0 = 次のWRコマンドで書き込み操作を実行します
- bit 3 **WRERR**: プログラム/消去エラー・フラグ・ビット  
 1 = 状態は、不適切なプログラムまたは消去シーケンスの試行または終了を示します (WRビットの設定試行 (書き込み「1」) 時にビットが自動的に設定されます)。  
 0 = プログラムまたは消去操作は正常に完了しました。
- bit 2 **WREN**: プログラム/消去許可ビット  
 1 = プログラム/消去サイクルを許可します  
 0 = プログラム・フラッシュのプログラミング/消去を禁止します
- bit 1 **WR**: 書き込み制御ビット  
 1 = プログラム・フラッシュのプログラム/消去操作を開始します。  
 操作はセルフタイムで行われ、操作が完了するとビットはハードウェアによってクリアされます。  
 WRビットはソフトウェアでのみセット (クリアされない) できます。  
 0 = フラッシュへのプログラム/消去操作は完了し、非アクティブです。
- bit 0 **RD**: 読み取り制御ビット  
 1 = プログラムのフラッシュ読み取りを開始します。読み取りには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のみ設定 (クリアされない) できます。  
 0 = プログラムのフラッシュ読み取りを開始しません。

注1: 実装されていないビット。「1」と読みます。

2: WRERRビットは、プログラム・メモリーの書き込みまたは消去操作が開始されると (WR = 1)、ハードウェアによって自動的に設定されます。

3: LWLOビットは、プログラム・メモリー消去操作中は無視されます (FREE = 1)。

## レジスタ10-6: PMCON2: プログラム・メモリー制御2レジスタ

W-0/0	W-0/0	W-0/0	W-0/0	W-0/0	W-0/0	W-0/0	W-0/0
プログラム・メモリー制御レジスタ2							
bit 7							bit 0

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている

bit 7-0      フラッシュ・メモリー・ロック解除パターン・ビット  
書き込みのロックを解除するには、PMCON1レジスタのWRビットを設定する前に、最初に55hを書き込み、次にAAhを書き込む必要があります。このレジスタに書き込まれた値は、書き込みのロックを解除するために使用されます。これらの書き込みには特定のタイミング要件があります。

表10-3: フラッシュ・プログラム・メモリーに関連付けられたレジスタの概要

名前	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ページ上の登録
INTCON	GIE	PEIE	TMR0IE	INTE	IOCFE	TMR0IF	INTF	IOCF	75
PMCON1	— <sup>(1)</sup>	CFGS	LWLO	FREE	WRERR	WREN	WR	RD	104
PMCON2	プログラム・メモリー制御レジスタ2								105
PMADRL	PMADRL<7:0>								103
PMADRH	— <sup>(1)</sup>	PMADRH<6:0>							103
PMDATL	PMDATL<7:0>								103
PMDATH	—	—	PMDATH<5:0>					103	

記録: — =未実装の場所、「0」として読み取ります。網掛けのセルはフラッシュ・プログラム・メモリーでは使用されません。

注1: 未実装、「1」と読みます。

表10-4: フラッシュ・プログラム・メモリーを使用した構成ワードの概要

名前	ビット	Bit -7	Bit -6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	ページ上の登録
CONFIG1	13:8	—	—	FCMEN	IESO	CLKOUTEN	BOREN<1:0>		—	41
	7:0	CP	MCLRE	PWRTÉ	WDTE<1:0>		FOSC<2:0>			
CONFIG2	13:8	—	—	LVP	DEBUG	LPBOR	BORV	STVREN	—	43
	7:0	—	—	—	—	—	WRT<1:0>			

記録: — =実装されていない場所、「0」と読みます。網掛けのセルはフラッシュ・プログラム・メモリーでは使用されません。

# PIC16(L)F1508/9

## 11.0 I/Oポート

各ポートには、その動作用に3つの標準レジスタがあります。これらのレジスタは：

- TRISxレジスタ（データ方向）
- PORTxレジスタ（デバイスのピンのレベルを読み取ります）
- LATxレジスタ（出力ラッチ）

一部のポートには、次の追加レジスタが1つ以上ある場合があります。これらのレジスタは：

- ANSELx（アナログ選択）
- WPUx（弱いプルアップ）

一般に、ポート・ピンで周辺が有効になっている場合、そのピンを汎用出力として使用することはできません。ただし、ピンは引き続き読み取ることができます。

表11-1: デバイスごとのポートの可用性

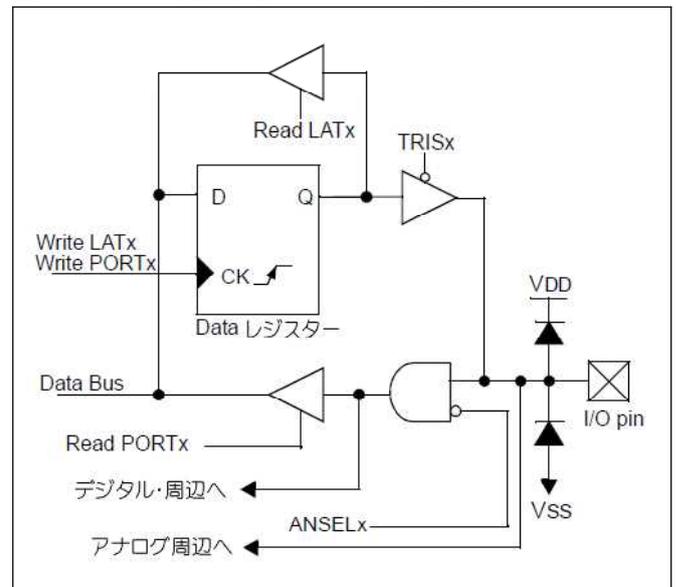
デバイス	ポートA	ポートB	ポートC
PIC16(L)F1508/9	●	●	●

データ・ラッチ（LATxレジスタ）は、I/Oピンが駆動している値のリード・モデファイ・ライト（読み取り - 変更 - 書き込み）操作に役立ちます。

LATxレジスタへの書き込み操作は、対応するPORTxレジスタへの書き込みと同じ効果があります。LATxレジスタの読み取りでは、I/Oポート・ラッチに保持されている値が読み取られ、PORTxレジスタの読み取りでは、実際のI/Oピンの値が読み取られます。

アナログ入力をサポートするポートには、ANSELxレジスタが関連付けられています。ANSELビットがセットされると、そのビットに関連付けられたデジタル入力バッファは無効になります。入力バッファを無効にすると、ロジック・ハイとロジック・ローの間のピンのアナログ信号レベルがロジック入力回路に過大な電流を発生させるのを防ぎます。図11-1に、他の周辺機器へのインターフェイスがない汎用I/Oポートの簡略化モデルを示します。

図11-1: 一般的なI/Oポートの操作



## 11.1 代替ピン機能

代替（オルタネート）ピン機能制御（APFCON）レジスタは、特定の周辺機器の入力および出力機能を異なるピン間で操作するために使用されます。APFCONレジスタはレジスタ11-1に示されています。このデバイス・ファミリーでは、次の機能を異なるピン間で移動できます。

- $\overline{SS}$
- T1G
- CLC1
- NCO1

これらのビットは、TRISレジスタの値には影響しません。PORTおよびTRISの上書きは、正しいピンに経路制御されます。選択されていないピンは影響を受けません。

## 11.2 レジスタ定義：代替ピン機能制御

レジスタ11-1: APFCON: 代替ピン機能制御レジスタ

U-0	U-0	U-0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0
—	—	—	SSSEL	T1GSEL	—	CLC1SEL	NCO1SEL
bit 7							bit 0

**記録：**

R =読み取り可能ビット	W =書き込み可能ビット	U =実装されていないビット、「0」として読み取られる
u =ビットは変更されていません	x =ビットは不明です	-n/n =PORでの値およびBOR/他のすべてのリセットでの値
[1]=ビットが設定されている	[0]=ビットがクリアされている	

- bit 7~5 **未実装**：「0」として読み取ります
- bit 4 **SSSEL**：ピン選択ビット
  - 1 =  $\overline{SS}$ 機能はRA3にあります
  - 0 =  $\overline{SS}$ 機能はRC6にあります
- bit 3 **T1GSEL**：ピン選択ビット
  - 1 = T1G機能はRA3にあります
  - 0 = T1G機能はRA4にあります
- bit 2 **未実装**：「0」として読み取ります
- bit 1 **CLC1SEL**：ピン選択ビット
  - 1 = CLC1機能はRC5にあります
  - 0 = CLC1機能はRA2にあります
- bit 0 **NCO1SEL**：ピン選択ビット
  - 1 = NCO1機能はRC6にあります
  - 0 = NCO1機能はRC1にあります

## 11.3 ポートAレジスター

### 11.3.1 データレジスター

ポートAは、6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ーはTRISA（レジスター11-3）です。TRISAビット（= 1）を設定すると、対応するポートAピンが入力になります（つまり、出力ドライバーが無効になります）。TRISAビット（= 0）をクリアすると、対応するポートAピンが出力になります（つまり、出力ドライバーが有効になり、出力ラッチの内容が選択されたピンに配置されます）。例外はRA3で、これは入力専用であり、そのTRISビットは常に「1」として読み取られます。例11-1に、I/Oポートを初期化する方法を示します。

PORTAレジスター（レジスター11-2）を読み取るとピンのステータスが読み取られますが、PORTAレジスターに書き込むとポート・ラッチに書き込まれます。すべての書き込み操作は、リード・モデファイ・ライト（読み取り - 変更 - 書き込み）操作です。したがって、ポートへの書き込みは、ポートピンが読み取られ、この値が変更されてから、ポート・データ・ラッチ（LATA）に書き込まれることを意味します。

### 11.3.2 方向制御

TRISAレジスター（レジスター11-3）は、アナログ入力として使用されている場合でも、PORTAピン出力ドライバを制御します。ユーザーは、TRISAレジスターのビットをアナログ入力として使用する場合、それらがセットされたままであることを確認する必要があります。アナログ入力として構成されたI/Oピンは常に「0」を読み取ります。

### 11.3.3 アナログ制御

ANSELAレジスター（レジスター11-5）は、I/Oピンの入力モードをアナログに設定するために使用されます。適切なANSELAビットをハイに設定すると、ピンのすべてのデジタル読み取りが「0」で読み取られ、ピンのアナログ機能が正しく動作できるようになります。

ANSELAビットの状態は、デジタル出力機能には影響しません。TRISクリアとANSELが設定されたピンは引き続きデジタル出力として動作しますが、入力モードはアナログになります。これにより、影響を受けるポートでリード・モデファイ・ライト命令を実行するときに予期しない動作が発生する可能性があります。

**注釈：** ANSELAビットは、リセット後、デフォルトでアナログ・モードになります。ピンをデジタル汎用または周辺入力として使用するには、対応するANSELビットをユーザー・ソフトウェアで「0」に初期化する必要があります。

### 例11-1: ポートAの初期化

```
BANKSEL          PORTA    ;
CLRF   PORTA    ;Init PORTA
BANKSEL          LATA     ;Data Latch
CLRF   LATA     ;
BANKSEL          ANSELA   ;
CLRF   ANSELA   ;digital I/O
BANKSEL          TRISA    ;
MOVLW   B'00111000' ;Set RA<5:3> as inputs
MOVWF   TRISA    ;and set RA<2:0> as
                    ;outputs
```

### 11.3.4 ポートAの機能と出力の優先順位

各ポートAピンは他の機能と多重化されています。ピン、それらを組み合わせた機能、およびそれらの出力優先順位を表11-2に示します。

複数の出力が有効になっている場合、実際のピン制御は最も優先度の高い周辺機器に送られます。

ADCやコンパレータ入力などのアナログ入力機能は、優先順位リストには表示されません。これらの入力は、ANSELxレジスターを使用してI/Oピンがアナログ・モードに設定されている場合にアクティブになります。デジタル出力機能は、アナログ・モードのときにピンを制御する場合があります。優先順位は以下の表11-2に示されています。

表11-2: ポートA出力の優先順位

ピン名	機能の優先順位
RA0	ICSPDAT DAC1OUT1 RA0
RA1	RA1
RA2	DAC1OUT2 CLC1 <sup>(2)</sup> C1OUT PWM3 RA2
RA3	なし
RA4	CLKOUT SOSCO RA4
RA5	SOSCI RA5

- 注1：** 優先度は高いものから低いものへとリストされています。
- 2：** デフォルトのピン（APFCONレジスタを参照）。
- 3：** 代替ピン（APFCONレジスタを参照）。

## 11.4 レジスター定義:ポートA

## レジスター11-2: PORTA: ポートAレジスター

U-0	U-0	R/W-x/x	R/W-x/x	R-x/x	R/W-x/x	R/W-x/x	R/W-x/x
—	—	RA5	RA4	RA3	RA2	RA1	RA0
bit 7							bit 0

## 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている

bit 7~6    **未実装**: 「0」として読み取ります  
bit 5~0    **RA<5:0>**: ポートA I/Oの値ビット<sup>(1)</sup>  
          1 = ポート・ピンは  $\geq V_{IH}$   
          0 = ポート・ピンは  $\leq V_{IL}$

注1: ポートAへの書き込みは、実際には対応するLATAレジスタに書き込まれます。 PORTAレジスタからの読み取りは、実際のI/Oピン値の戻りです。

## レジスター11-3: TRISA: ポートAトライステート・レジスター

U-0	U-0	R/W-1/1	R/W-1/1	U-1	R/W-1/1	R/W-1/1	R/W-1/1
—	—	TRISA5	TRISA4	— <sup>(1)</sup>	TRISA2	TRISA1	TRISA0
bit 7							bit 0

## 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている

bit 7~6    **未実装**: 「0」として読み取ります  
bit 5~4    **TRISA<5:4>**: ポートAトライステート制御ビット  
          1 = 入力として構成されたポートAピン (トライステート)  
          0 = 出力として構成されたポートAピン  
bit 3      **未実装**: 「1」と読みます  
bit 2~0    **TRISA<2:0>**: ポートAトライステート制御ビット  
          1 = 入力として構成されたポートAピン (トライステート)  
          0 = 出力として構成されたポートAピン

注1: 未実装、「1」と読みます。

# PIC16(L)F1508/9

## レジスタ11-4: LATA: ポートAデータ・ラッチ・レジスタ

U-0	U-0	R/W-x/u	R/W-x/u	U-0	R/W-x/u	R/W-x/u	R/W-x/u
—	—	LATA5	LATA4	—	LATA2	LATA1	LATA0
bit 7							bit 0

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている

- bit 7~6    **未実装**: 「0」として読み取ります
- bit 5~4    **LATA<5:4>**: RA<5:4>出力ラッチ (保持) 値ビット<sup>(1)</sup>
- bit 3       **未実装**: 「0」として読み取ります
- bit 2~0    **LATA<2:0>**: RA<2:0>出力ラッチ値ビット<sup>(1)</sup>

**注1**: ポートAへの書き込みは、実際には対応するLATAレジスタに書き込まれます。PORTAレジスタからの読み取りは、実際のI/Oピン値の戻りです。

## レジスタ11-5: ANSELA: ポートAアナログ選択レジスタ

U-0	U-0	U-0	R/W-1/1	U-0	R/W-1/1	R/W-1/1	R/W-1/1
—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0
bit 7							bit 0

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている

- bit 7~5    **未実装**: 「0」として読み取ります
- bit 4       **ANSA4**: アナログ・ピンRA4のアナログ機能またはデジタル機能をそれぞれ選択します  
1 = アナログ入力。ピンはアナログ入力<sup>(1)</sup>として割り当てられます。デジタル入力バッファが無効になっています。  
0 = デジタルI/O。ピンはポートまたはデジタル特殊機能に割り当てられます。
- bit 3       **未実装**: 「0」として読み取ります
- bit 2~0    **ANSA<2:0>**: アナログ・ピンRA<2:0>のアナログまたはデジタル機能からそれぞれ選択  
1 = アナログ入力。ピンはアナログ入力<sup>(1)</sup>として割り当てられます。デジタル入力バッファが無効になっています。  
0 = デジタルI/O。ピンはポートまたはデジタル特殊機能に割り当てられます。

**注1**: ピンをアナログ入力に設定する場合、ピンの電圧を外部制御できるようにするには、対応するTRISビットを入力モードに設定する必要があります。

## レジスタ11-6: WPUA: 弱いプルアップのポートAレジスタ

U-0	U-0	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
—	—	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0
bit 7							bit 0

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている

- bit 7~6    未実装: 「0」として読み取ります
- bit 5~0    **WPUA<5:0>**: 弱い(Weak)プルアップ・レジスタ・ビット<sup>(3)</sup>
  - 1 = プルアップが有効
  - 0 = プルアップが無効

- 注1: 個々のプルアップを有効にするには、OPTION\_REGレジスタのグローバル $\overline{\text{WPUEN}}$ ビットをクリアする必要があります。
- 2: ピンが出力として構成されている場合、弱いプルアップ・デバイスは自動的に無効になります。
- 3: WPUA3ビットの場合、MCLRE = 1の場合、弱いプルアップが内部で有効になりますが、ここでは報告されません。

## 表11-3: ポートAに関連するレジスタの概要

名前	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ページ上の登録
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	110
APFCON	—	—	—	SSSEL	T1GSEL	—	CLC1SEL	NCO1SEL	107
LATA	—	—	LATA5	LATA4	—	LATA2	LATA1	LATA0	110
OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			154
PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	109
TRISA	—	—	TRISA5	TRISA4	— <sup>(1)</sup>	TRISA2	TRISA1	TRISA0	109
WPUA	—	—	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0	111

記録: x =不明。u =変更なし。— =実装されていない場所は「0」として読み取られます。網掛けのセルはポートAでは使用されません。

注1: 未実装、「1」と読みます。

## 表11-4: ポートAを使用した構成ワードの概要

名前	ビット	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	ページ上の登録
CONFIG1	13:8	—	—	FCMEN	IESO	CLKOUTEN	BOREN<1:0>		—	41
	7:0	$\overline{\text{CP}}$	MCLRE	PWRTE	WDTE<1:0>		FOSC<2:0>			

記録: — =未実装の場所、「0」として読み取ります。網掛けのセルはポートAでは使用されません。

## 11.5 PORTBレジスター

### 11.5.1 データレジスター

ポートBは、4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ーはTRISB（レジスター11-8）です。TRISBビット（= 1）を設定すると、対応するポートBピンが入力になります（つまり、出力ドライバーが無効になります）。TRISBビット（= 0）をクリアすると、対応するポートBピンが出力になります（つまり、出力ドライバーが有効になり、出力ラッチの内容が選択されたピンに配置されます）。例11-1に、I/Oポートを初期化する方法を示します。

PORTBレジスター（レジスター11-7）を読み取ると、ピンのステータスが読み取られますが、PORTBレジスターに書き込むと、ポート・ラッチに書き込まれます。すべての書き込み操作は、リード・モデファイ・ライト（読み取り - 変更 - 書き込み）操作です。したがって、ポートへの書き込みは、ポートピンが読み取られ、この値が変更されてから、ポート・データラッチ（LATB）に書き込まれることを意味します。

### 11.5.2 方向制御

TRISBレジスター（レジスター11-8）は、アナログ入力として使用されている場合でも、PORTBピン出力ドライバを制御します。ユーザーは、TRISBレジスターのビットをアナログ入力として使用する場合、それらがセットされたままであることを確認する必要があります。アナログ入力として構成されたI/Oピンは常に「0」を読み取ります。

### 11.5.3 アナログ制御

ANSELBレジスター（レジスター11-10）は、I/Oピンの入力モードをアナログに設定するために使用されます。適切なANSELBビットをハイに設定すると、ピンのすべてのデジタル読み取りが「0」として読み取られ、ピンのアナログ機能が正しく動作できるようになります。

ANSELBビットの状態は、デジタル出力機能には影響しません。TRISクリアとANSELが設定されたピンは引き続きデジタル出力として動作しますが、入力モードはアナログになります。これにより、影響を受けるポートでリード・モデファイ・ライト（読み取り - 変更 - 書き込み）命令を実行するときに予期しない動作が発生する可能性があります。

**注釈：** リセット後、ANSELBビットはデフォルトでアナログ・モードになります。ピンをデジタル汎用または周辺入力として使用するには、対応するANSELビットをユーザー・ソフトウェアで「0」に初期化する必要があります。

### 11.5.4 PORTBの機能と出力の優先順位

各ポートBピンは他の機能と多重化されています。ピン、それらの組み合わせ機能、およびそれらの出力優先順位を表11-5に示します。

複数の出力が有効になっている場合、実際のピン制御は最も優先度の高い周辺機器に送られます。

ADCやコンパレータ入力などのアナログ入力機能は、優先順位リストには表示されません。これらの入力は、ANSELxレジスターを使用してI/Oピンがアナログ・モードに設定されている場合にアクティブになります。デジタル出力機能は、アナログ・モードのときにピンを制御する場合があります。優先順位は以下の表11-5に示されています。

表11-5: ポートB出力の優先順位

ピン名	機能の優先順位 <sup>(1)</sup>
RB4	SDA RB4
RB5	RB5
RB6	SCL SCK RB6
RB7	CLC3 TX RB7

注1: 優先度は高いものから低いものへとリストされています。

- 2: デフォルト・ピン（APFCONレジスターを参照）。
- 3: 代替ピン（APFCONレジスターを参照）。

## 11.6 レジスター定義: ポートB

## レジスター11-7: PORTB: ポートBレジスター

R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	U-0	U-0	U-0	U-0
RB7	RB6	RB5	RB4	—	—	—	—
bit 7							bit 0

## 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている

bit 7~4    **RB<7:4>** : PORTB I/O値ビット<sup>(1)</sup>

1 = ポート・ピン  $\geq V_{IH}$

0 = ポート・ピン  $\leq V_{IL}$

bit 3~0    **未実装** : 「0」として読み取ります

注1: PORTBへの書き込みは、実際には対応するLATBレジスターに書き込まれます。PORTBレジスターからの読み取りは、実際のI/Oピン値の戻りです。

## レジスター11-8: TRISB: ポートBトライステート・レジスター

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	U-0	U-0	U-0	U-0
TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—
bit 7							bit 0

## 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている

bit 7~4    **RB<7:4>** : ポートBトライステート制御ビット

1 = 入力として構成されたポートBピン (トライステート)

0 = 出力として構成されたポートBピン

bit 3~0    **未実装** : 「0」として読み取ります

# PIC16(L)F1508/9

## レジスタ11-9: LATB: ポートBデータ・ラッチ・レジスタ

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	U-0	U-0	U-0	U-0
LATB7	LATB6	LATB5	LATB4	—	—	—	—
bit 7							bit 0

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている

bit 7~4    **LATB<7:4>**: RB<7:4>出力ラッチ値ビット<sup>(1)</sup>

bit 3~0    **未実装**: 「0」として読み取ります

注1: ポートBへの書き込みは、実際には対応するLATBレジスタに書き込まれます。PORTBレジスタからの読み取りは、実際のI/Oピン値の戻りです。

## レジスタ11-10: ANSELB: ポートBアナログ選択レジスタ

U-0	U-0	R/W-1/1	R/W-1/1	U-0	U-0	U-0	U-0
—	—	ANSB5	ANSB4	—	—	—	—
bit 7							bit 0

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている

bit 7~6    **未実装**: 「0」として読み取ります

bit 5~4    **ANSB<5:4>**: アナログ・ピンRB<5:4>のアナログまたはデジタル機能からそれぞれ選択

1 = アナログ入力。ピンはアナログ入力<sup>(1)</sup>として割り当てられます。デジタル入力バッファが無効になっています。

0 = デジタルI/O。ピンはポートまたはデジタル特殊機能に割り当てられます。

bit 3~0    **未実装**: 「0」として読み取ります

注1: ピンをアナログ入力に設定する場合、ピンの電圧を外部制御できるようにするには、対応するTRISビットを入力モードに設定する必要があります。

## レジスタ11-11: WPUB: 弱いプルアップのポートBレジスタ

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	U-0	U-0	U-0	U-0
WPUB7	WPUB6	WPUB5	WPUB4	—	—	—	—
bit 7							bit 0

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている

bit 7~4    **WPUB<7:4>**: 弱いプルアップ・レジスタ・ビット

- 1 = プルアップが有効
- 0 = プルアップ無効

bit 3~0    **未実装**: 「0」として読み取ります

- 注1: 個々のプルアップを有効にするには、OPTION\_REGレジスタのグローバル $\overline{\text{WPUEN}}$ ビットをクリアする必要があります。
- 2: ピンが出力として構成されている場合、弱いプルアップ・デバイスは自動的に無効になります。

表11-6: ポートBに関連付けられているレジスタの要約

名前	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ページ上の登録
ANSELB	—	—	ANSB5	ANSB4	—	—	—	—	114
APFCON	—	—	—	SSSEL	T1GSEL	—	CLC1SEL	NCO1SEL	107
LATB	LATB7	LATB6	LATB5	LATB4	—	—	—	—	114
OPTION_REG	$\overline{\text{WPUEN}}$	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			154
PORTB	RB7	RB6	RB5	RB4	—	—	—	—	113
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	113
WPUB	WPUB7	WPUB6	WPUB5	WPUB4	—	—	—	—	113

記録: x =不明。u =変更なし。— =未実装の場所は「0」として読み取られます。網掛けのセルはポートBでは使用されません。

注1: 未実装、「1」と読みます。

表11-7: ポートBを使用した構成ワードの要約

名前	ビット	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	ページ上の登録
CONFIG1	13:8	—	—	FCMEN	IESO	$\overline{\text{CLKOUTEN}}$	BOREN<1:0>		—	41
	7:0	$\overline{\text{CP}}$	MCLRE	$\overline{\text{PWRTE}}$	WDTE<1:0>		FOSC<2:0>		—	

記録: — =実装されていない場所、「0」と読みます。網掛けのセルはポートBでは使用されません。

## 11.7 PORTCレジスター

### 11.7.1 データレジスター

ポートC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ーはTRISC（レジスター11-13）です。TRISCビット（=1）を設定すると、対応するPORTCピンが入力になります（つまり、出カドライバーが無効になります）。TRISCビット（=0）をクリアすると、対応するPORTCピンが出力になります（つまり、出カドライバーを有効にして、出カラッチの内容を選択したピンに配置します）。例11-1に、I/Oポートを初期化する方法を示します。

PORTCレジスター（レジスター11-12）を読み取ると、ピンのステータスが読み取られますが、書き込みを行うと、ポート・ラッチに書き込まれます。すべての書き込み操作は、リード・モデファイ・ライト（読み取り - 変更 - 書き込み）操作です。したがって、ポートへの書き込みは、ポート・ピンが読み取られ、この値が変更されてから、ポート・データ・ラッチ（LATC）に書き込まれることを意味します。

### 11.7.2 方向制御

TRISCレジスター（レジスター11-13）は、アナログ入力として使用されている場合でも、PORTCピン出カドライバーを制御します。ユーザーは、TRISCレジスターのビットをアナログ入力として使用する場合、それらのビットがセットされたままであることを確認する必要があります。アナログ入力として構成されたI/Oピンは常に「0」を読み取ります。

### 11.7.3 アナログ制御

ANSELCレジスター（レジスター11-15）は、I/Oピンの入力モードをアナログに設定するために使用されます。適切なANSELCビットをハイに設定すると、ピンのすべてのデジタル読み取りが「0」として読み取られ、ピンのアナログ機能が正しく動作できるようになります。

ANSELCビットの状態は、デジタル出力機能には影響しません。TRISクリアとANSELCが設定されたピンは引き続きデジタル出力として動作しますが、入力モードはアナログになります。これにより、影響を受けるポートでリード・モデファイ・ライト（読み取り - 変更 - 書き込み）命令を実行するときに予期しない動作が発生する可能性があります。

**注：** ANSELCビットは、リセット後、デフォルトでアナログ・モードになります。ピンをデジタル汎用または周辺入力として使用するには、対応するANSELビットをユーザー・ソフトウェアで「0」に初期化する必要があります。

### 11.7.4 PORTCの機能と出力の優先順位

各ポートCピンは他の機能と多重化されています。ピン、それらを組み合わせた機能、およびそれらの出力優先順位を表11-8に示します。

複数の出力が有効になっている場合、実際のピン制御は最も優先度の高い周辺機器に送られます。

アナログ入力および一部のデジタル入力機能は、出力優先順位リストに含まれていません。これらの入力機能は、ピンが出力として構成されている場合でもアクティブのままにすることができます。特定のデジタル入力機能は他のポート機能を上書きし、出力優先順位リストに含まれます。

表11-8: ポートC出力の優先順位

ピン名	機能の優先順位 <sup>(1)</sup>
RC0	CLC2 RC0
RC1	NCO1 <sup>(2)</sup> PWM4 RC1
RC2	RC2
RC3	PWM2 RC3
RC4	CWG1B CLC4 C2OUT RC4
RC5	CWG1A CLC1 <sup>(3)</sup> PWM1 RC5
RC6	NCO1 <sup>(3)</sup> RC6
RC7	SDO RC7

注1: 優先度は高いものから低いものへとリストされています。

2: デフォルト・ピン（APFCONレジスターを参照）。

3: 代替ピン（APFCONレジスターを参照）。

## 11.8 レジスター定義: ポートC

## レジスター11-12: PORTC: ポートCレジスター

| R/W-x/u |
|---------|---------|---------|---------|---------|---------|---------|---------|
| RC7     | RC6     | RC5     | RC4     | RC3     | RC2     | RC1     | RC0     |
| bit 7   |         |         |         |         |         |         | bit 0   |

## 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている

bit 7~0 **RC<7:0>**: ポートC汎用目的I/Oピン・ビット  
1 = ポート・ピン  $\geq$  VIH  
0 = ポート・ピン  $\leq$  VIL

## レジスター11-13: TRISC: ポートCトライステート・レジスター

| R/W-1/1 |
|---------|---------|---------|---------|---------|---------|---------|---------|
| TRISC7  | TRISC6  | TRISC5  | TRISC4  | TRISC3  | TRISC2  | TRISC1  | TRISC0  |
| bit 7   |         |         |         |         |         |         | bit 0   |

## 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている

bit 7~0 **TRISC<7:0>**: ポートCトライステート制御ビット  
1 = 入力として構成されたポートCピン (トライステート)  
0 = 出力として構成されたポートCピン

## レジスター11-14: LATC: ポートCデータ・ラッチ・レジスター

| R/W-x/u |
|---------|---------|---------|---------|---------|---------|---------|---------|
| LATC7   | LATC6   | LATC5   | LATC4   | LATC3   | LATC2   | LATC1   | LATC0   |
| bit 7   |         |         |         |         |         |         | bit 0   |

## 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている

bit 7~0 **LATC<7:0>**: ポートC出力ラッチ値ビット<sup>(1)</sup>

注1: ポートCへの書き込みは、実際には対応するLATCレジスターに書き込まれます。PORTCレジスターからの読み取りは、実際のI/Oピン値の戻りです。

# PIC16(L)F1508/9

## レジスタ11-15: ANSELC: ポートCアナログ選択レジスタ

R/W-1/1	R/W-1/1	U-0	U-0	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
ANSC7	ANSC6	—	—	ANSC3	ANSC2	ANSC1	ANSC0
bit 7							bit 0

<b>記録:</b>		
R =読み取り可能ビット	W =書き込み可能ビット	U =実装されていないビット、「0」として読み取られる
u =ビットは変更されていません	x =ビットは不明です	-n/n =PORでの値およびBOR/他のすべてのリセットでの値
「1」=ビットが設定されている	「0」=ビットがクリアされている	

- bit 7~6 **ANSC<7:6>**: アナログ・ピンRC<7:6>のアナログまたはデジタル機能からそれぞれ選択  
 1 = アナログ入力。ピンはアナログ入力<sup>(1)</sup>として割り当てられます。デジタル入力バッファが無効になっています。  
 0 = デジタルI/O。ピンはポートまたはデジタル特殊機能に割り当てられます。
- bit 5~4 **未実装**: 「0」として読み取ります
- bit 3~0 **ANSC<3:0>**: アナログ・ピンRC<3:0>のアナログまたはデジタル機能からそれぞれ選択  
 1 = アナログ入力。ピンはアナログ入力<sup>(1)</sup>として割り当てられます。デジタル入力バッファが無効になっています。  
 0 = デジタルI/O。ピンはポートまたはデジタル特殊機能に割り当てられます。

**注1:** ピンをアナログ入力に設定する場合、ピンの電圧を外部制御できるようにするには、対応するTRISビットを入力モードに設定する必要があります。

**表11-9: ポートCに関連付けられているレジスタの概要**

名前	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ページ上の登録
ANSELC	ANSC7	ANSC6	—	—	ANSC3	ANSC2	ANSC1	ANSC0	118
LATC	LATC7	LATC6	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	117
PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	117
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	117

**記録:** x =不明。u =変更なし。— =未実装の場所は「0」として読み取られます。網掛けのセルはポートCでは使用されません。

## 12.0 変更時の割り込み

ポートAピンとポートBピンは、インタラプト・オン・チェンジ (IOC) ピンとして動作するように構成できます。割り込みは、立ち上がりエッジまたは立ち下がりエッジのいずれかを持つ信号を検出することによって生成できます。個々のポートピン、またはポートピンの組み合わせは、割り込みを生成するように構成できます。変更時割り込みモジュールには、次の機能があります：

- 変更時割り込みイネーブル (マスター・スイッチ)
- 個別のピン構成
- 立ち上がりエッジと立ち下がりエッジの検出
- 個々のピン割り込みフラグ

図12-1は、IOCモジュールのブロック図です。

### 12.1 モジュールの有効化

個々のポート・ピンが割り込みを生成できるようにするには、INTCONレジスタのIOCIEビットを設定する必要があります。IOCIEビットが無効になっている場合でも、ピンのエッジ検出は発生しますが、割り込みは生成されません。

### 12.2 個別のピン構成

各ポート・ピンには、立ち上がりエッジ検出器と立ち下がりエッジ検出器があります。ピンが立ち上がりエッジを検出できるようにするには、IOCxPレジスタの関連ビットが設定されます。ピンが立ち下がりエッジを検出できるようにするには、IOCxNレジスタの関連ビットが設定されます。

## 12.3 割り込みフラグ

IOCAFレジスタとIOCBFレジスタにそれぞれあるIOCAFxビットとIOCBFxビットは、関連するポートの変更時割り込みピンに対応するステータス・フラグです。適切に有効化されたピンで予期されるエッジが検出されると、そのピンのステータス・フラグが設定され、IOCIEビットが設定されると割り込みが生成されます。INTCONレジスタのIOCIFビットは、すべてのIOCAFxビットとIOCBFxビットのステータスを反映します。

### 12.4 割り込みフラグのクリア

個々のステータス・フラグ (IOCAFxおよびIOCBFxビット) は、ゼロにリセットすることでクリアできます。このクリア操作中に別のエッジが検出された場合、実際に書き込まれている値に関係なく、関連するステータス・フラグがシーケンスの最後に設定されます。

フラグのクリア中に検出されたエッジが失われないようにするには、既知の変更されたビットをマスクするAND演算のみを実行する必要があります。次のシーケンスは、実行する必要があるものの例です。

#### 例12-1: 割り込みフラグのクリア (PORTAの例)

MOVLW	0xff
XORWF	IOCAF, W
ANDWF	IOCAF, F

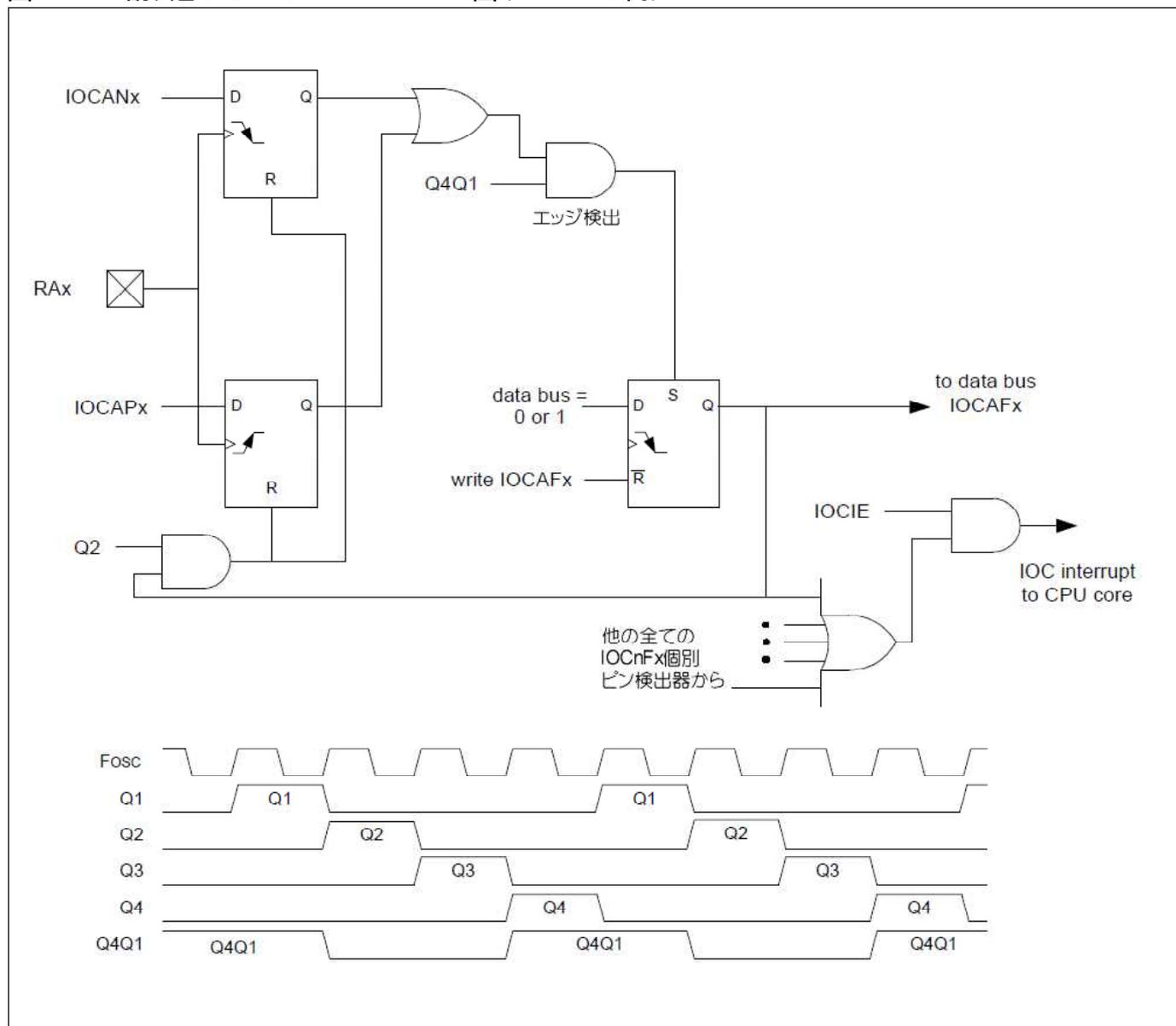
## 12.5 スリープ状態での操作

IOCIEビットが設定されている場合、割り込みオン・チェンジ割り込みシーケンスはデバイスをスリープ・モードからウェイクアップ (蘇り) します。

スリープ・モード中にエッジが検出された場合、IOCxFレジスタは、スリープから実行される最初の命令の前に更新されます。

# PIC16(L)F1508/9

図12-1: 割り込みオン・チェンジ・ブロック図(ポートAの例)



## 12.6 レジスタ定義: 変更時割り込み制御

## レジスタ12-1: IOCAP: 割り込みオン・チェンジ・ポートAポジティブ・エッジ・レジスタ

U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0
bit 7							bit 0

## 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている

bit 7~6 未実装: 「0」として読み取ります

bit 5~0 IOCAP<5:0>: 割り込みオン・チェンジ・ポートA正方向エッジ許可ビット

1 = 正方向エッジのピンで割り込みオン・チェンジが有効になっています。エッジを検出すると、IOCAF<sub>x</sub>ビットとIOCIFフラグが設定されます。

0 = 関連するピンの変更時割り込み(割り込みオン・チェンジ)は無効です。

## レジスタ12-2: IOCAN: 割り込みオン・チェンジ・ポートAネガティブ・エッジ・レジスタ

U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0
bit 7							bit 0

## 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている

bit 7~6 未実装: 「0」として読み取ります

bit 5~0 IOCAN<5:0>: 割り込みオン・チェンジ・ポートA負方向エッジ許可ビット

1 = 立ち下がりがエッジのピンで割り込みオン・チェンジが有効になっています。エッジを検出すると、IOCAF<sub>x</sub>ビットとIOCIFフラグが設定されます。

0 = 関連するピンの変更時割り込みは無効です。

## レジスタ12-3: IOCAF: 割り込みオン・チェンジ・ポートAフラグ・レジスタ

U-0	U-0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0
—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0
bit 7							bit 0

## 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている      HSビットはハードウェアに設定されます

bit 7~6 未実装: 「0」として読み取ります

bit 5~0 IOCAF<5:0>: 割り込みオン・チェンジ・ポートAフラグ・ビット

1 = 関連するピンで有効な変更が検出されました。

IOCAP<sub>x</sub> = 1でRAXで立ち上がりエッジが検出された場合、またはIOCAN<sub>x</sub> = 1で立ち下がりがエッジがRAXで検出された場合に設定します。

0 = 変更が検出されなかったか、ユーザーが検出された変更をクリアしました。

# PIC16(L)F1508/9

## レジスタ12-4: IOCBP: 割り込みオン・チェンジ・ポートBポジティブ(正方向)エッジ・レジスタ

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
IOCBP7	IOCBP6	IOCBP5	IOCBP4	—	—	—	—
bit 7							bit 0

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている

- bit 7~4 **IOCBP<7:4>**: インタラプト・オン・チェンジ(変更時割り込み)・ポートB正方向エッジ許可ビット  
1 = 正方向エッジのピンで変更時割り込みが有効になっています。エッジを検出すると、IOCBFxビットとIOCIFフラグが設定されます。  
0 = 関連するピンの変更時割り込みは無効です。
- bit 3~0 **未実装**: 「0」として読み取ります

## レジスタ12-5: IOCBN: 割り込みオン・チェンジ・ポートBネガティブ(負方向)エッジ・レジスタ

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
IOCBN7	IOCBN6	IOCBN5	IOCBN4	—	—	—	—
bit 7							bit 0

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている

- bit 7~4 **IOCBN<7:4>**: 変更時の割り込みポートB負方向エッジ許可ビット  
1 = 立ち下がりエッジのピンで変更時の割り込みが有効になっています。エッジを検出すると、IOCBFxビットとIOCIFフラグが設定されます。  
0 = 関連するピンの変更時割り込みは無効です。
- bit 3~0 **未実装**: 「0」として読み取ります

## レジスタ12-6: IOCBF: 割り込みオン・チェンジ・ポートBフラグ・レジスタ

R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	U-0	U-0	U-0	U-0
IOCBF7	IOCBF6	IOCBF5	IOCBF4	—	—	—	—
bit 7							bit 0

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている      HSビットはハードウェアに設定されます

- bit 7~4 **IOCBF<7:4>**: 変更時割り込みポートBフラグ・ビット  
1 = 関連するピンで有効な変更が検出されました。  
IOCBPx = 1でRBxで立ち上がりエッジが検出された場合、またはIOCBNx = 1でRBxで立ち下がりエッジが検出された場合に設定されます。  
0 = 変更が検出されなかったか、ユーザーが検出された変更をクリアしました。
- bit 3~0 **未実装**: 「0」として読み取ります

表12-1: 変更時の割り込みに関連するレジスタの概要

名前	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ページ上の登録
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	110
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	75
IOCAF	—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	121
IOCAN	—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	121
IOCAP	—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	121
IOCBF	IOCBF7	IOCBF6	IOCBF5	IOCBF4	—	—	—	—	122
IOCBN	IOCBN7	IOCBN6	IOCBN5	IOCBN4	—	—	—	—	122
IOCBP	IOCBP7	IOCBP6	IOCBP5	IOCBP4	—	—	—	—	122
TRISA	—	—	TRISA5	TRISA4	— <sup>(1)</sup>	TRISA2	TRISA1	TRISA0	109
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	113

記録： — =実装されていない場所、「0」と読みます。網掛けのセルは、変更時の割り込みでは使用されません。

注1： 未実装、「1」と読みます。

# PIC16(L)F1508/9

## 13.0 固定電圧リファレンス (FVR)

固定電圧リファレンス (FVR) は、VDDに依存しない安定した電圧リファレンスであり、公称出力レベル (VFVR) は1.024Vです。FVRの出力は、以下に基準電圧を供給するように構成できます：

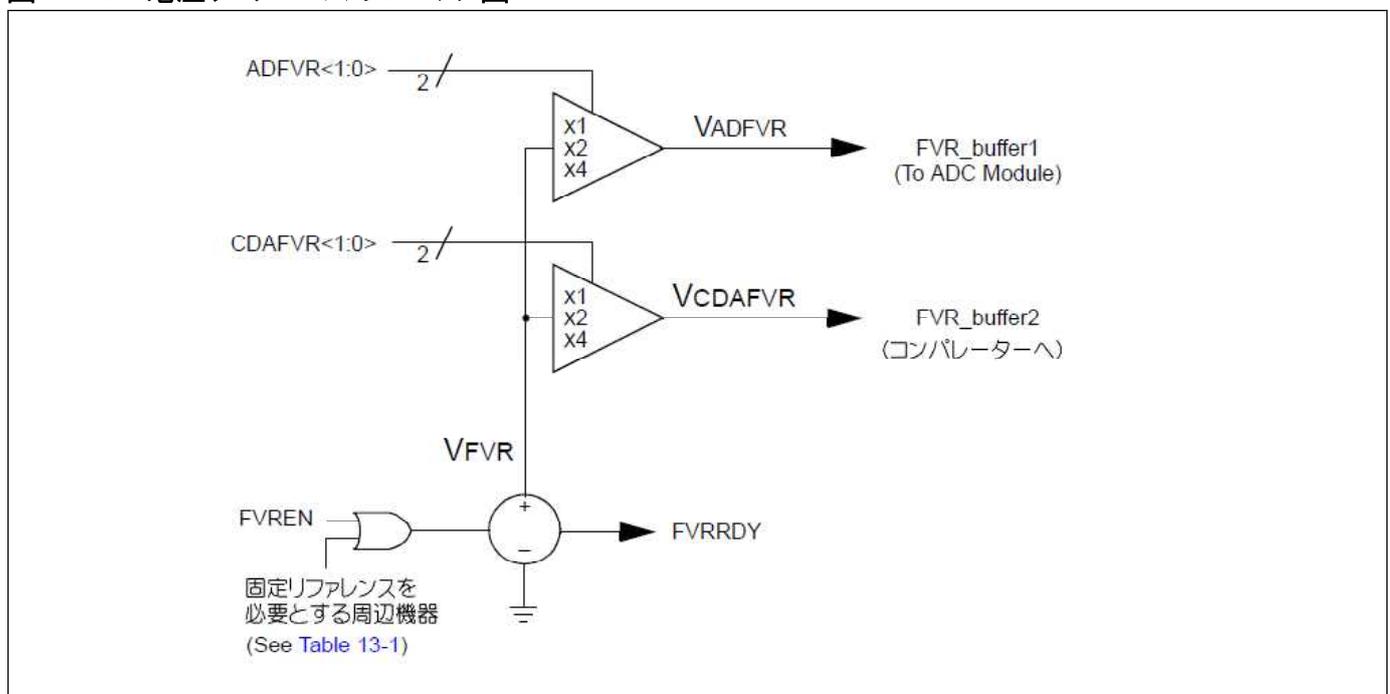
- ADC入力チャンネル
- コンパレーターの正の入力
- コンパレーターの負の入力

FVRは、FVRCONレジスタのFVRENビットをセットすることで有効にできます。

### 13.1 独立したゲイン・アンプ

周辺機器に供給されるFVRの出力 (上記にリストされている) は、プログラム可能なゲイン・アンプを介して経路割り出しされます。各アンプは、1x、2x、または4xのゲインにプログラムして、3つの可能な電圧レベルを生成できます。

図13-1: 電圧リファレンスのブロック図



FVRCONレジスタのADFVR<1:0>ビットは、ADCモジュールに供給されるリファレンスのゲイン・アンプ設定を有効にして構成するために使用されます。詳細については、セクション15.0「アナログ・デジタル・コンバーター (ADC) モジュール」を参照してください。

FVRCONレジスタのCDAFVR<1:0>ビットは、コンパレーター・モジュールに供給されるリファレンスのゲイン・アンプ設定を有効にして構成するために使用されます。詳細については、セクション17.0「コンパレーター・モジュール」を参照してください。

### 13.2 FVR安定化期間

固定電圧リファレンス・モジュールを有効にすると、リファレンス回路とアンプ回路が安定するまでに時間がかかります。回路が安定して使用できるようになると、FVRCONレジスタのFVRRDYビットがセットされます。FVR安定化期間の特性グラフ (図30-65) を参照してください。

表13-1: 固定電圧リファレンス (FVR) を必要とする周辺機器

周辺	条件	説明
HFINTOSC	FOSC<2:0> = 010 と IRCF<3:0> = 000x	INTOSCはアクティブであり、デバイスはスリープ状態ではありません。
BOR	BOREN<1:0> = 11	BORは常に有効です。
	BOREN<1:0> = 10 と BORFS = 1	スリープモードではBORが無効になり、BORファストスタートが有効になります。
LDO	BOREN<1:0> = 01 と BORFS = 1	ソフトウェア制御下のBOR、BORファストスタートが有効。
	VREGPM = 1でスリープ状態にない場合、すべてのPIC16F1508/9デバイス	スリープモードの場合、デバイスは低電力レギュレータで動作します。

13.3 レジスター定義: FVR制御

レジスター13-1: FVRCON: 固定電圧リファレンス制御レジスター

R/W-0/0	R-q/q	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
FVREN	FVRRDY <sup>(1)</sup>	TSEN <sup>(3)</sup>	TSRNG <sup>(3)</sup>	CDAFVR<1:0>		ADFVR<1:0>	
bit 7							bit 0

記録:		
R =読み取り可能ビット	W =書き込み可能ビット	U =実装されていないビット、「0」として読み取られる
u =ビットは変更されていません	x =ビットは不明です	-n/n =PORでの値およびBOR/他のすべてのリセットでの値
[1]=ビットが設定されている	[0]=ビットがクリアされている	q =値は条件によって異なります

- bit 7 **FVREN**: 固定電圧リファレンス (参照) 許可ビット  
 1 = 固定電圧リファレンスが有効になっている  
 0 = 固定電圧リファレンスが無効になっています
- bit 6 **FVRRDY**: 固定電圧リファレンス・レディ (準備) フラグ・ビット<sup>(1)</sup>  
 1 = 固定電圧リファレンス出力はすぐに使用できます  
 0 = 固定電圧リファレンス出力の準備ができていないか、有効になっていません
- bit 5 **TSEN**: 温度インジケータ (表示板) 許可ビット<sup>(3)</sup>  
 1 = 温度インジケータが有効になっている  
 0 = 温度インジケータが無効になっています
- bit 4 **TSRNG**: 温度インジケータ・レンジ (範囲) 選択ビット<sup>(3)</sup>  
 1 =  $V_{OUT} = V_{DD} - 4V_T$  (高レンジ)  
 0 =  $V_{OUT} = V_{DD} - 2V_T$  (低レンジ)
- bit 3~2 **CDAFVR<1:0>**: コンパレータFVRバッファ・ゲイン (利得) 選択ビット  
 11 = コンパレータFVRバッファ利得は4xで、出力 $V_{CDAFVR} = 4x V_{FVR}$ <sup>(2)</sup>  
 10 = コンパレータFVRバッファ利得は2xで、出力 $V_{CDAFVR} = 2x V_{FVR}$ <sup>(2)</sup>  
 01 = コンパレータFVRバッファ利得は1xで、出力 $V_{CDAFVR} = 1x V_{FVR}$   
 00 = コンパレータFVRバッファはオフ
- bit 1~0 **ADFVR<1:0>**: ADC FVRバッファ利得選択ビット  
 11 = ADC FVRバッファ利得は4xで、出力 $V_{ADFVR} = 4x V_{FVR}$ <sup>(2)</sup>  
 10 = ADC FVRバッファ利得は2xで、出力 $V_{ADFVR} = 2x V_{FVR}$ <sup>(2)</sup>  
 01 = ADC FVRバッファ利得は1xで、出力 $V_{ADFVR} = 1x V_{FVR}$   
 00 = ADC FVRバッファはオフ

注1: PIC16F1508/9デバイスの場合、FVRRDYは常に「1」です。  
 2: 固定電圧リファレンス出力はVDDを超えることはできません。  
 3: 追加情報については、セクション14.0「温度インジケータ・モジュール」を参照してください。

表13-2: 固定電圧リファレンスに関連付けられているレジスターの概要

名前	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ページ上の登録
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR>1:0>		ADFVR<1:0>		125

記録: 影付きのセルは、固定電圧リファレンス・モジュールでは使用されていません。

# PIC16(L)F1508/9

## 14.0 温度インジケータ・モジュール

このデバイス・ファミリーには、シリコン・ダイ（チップ）の動作温度を測定するように設計された温度回路が装備されています。回路の動作温度範囲は-40°Cから+85°Cの間です。出力は、デバイスの温度に比例する電圧です。温度インジケータ（表示板）の出力は、内部でデバイスADCに接続されています。

この回路は、実行される校正のレベルに応じて、温度しきい値検出器またはより正確な温度インジケータ（表示板）として使用できます。ワンポイント校正により、回路はそのポイントのすぐ周囲の温度を示すことができます。2点校正により、回路は温度範囲全体をより正確に検出できます。校正プロセスの詳細については、アプリケーション・ノートAN1333「内部温度インジケータの使用と校正」（DS01333）を参照してください。

### 14.1 回路動作

図14-1に、温度回路の簡略ブロック図を示します。比例電圧出力は、複数のシリコン接合間の順方向電圧降下を測定することによって実現されます。

式14-1に、温度インジケータの出力特性を示します。

#### 式14-1: $V_{OUT}$ レンジ(範囲)

高レンジ:  $V_{OUT} = V_{DD} - 4V_T$

低レンジ:  $V_{OUT} = V_{DD} - 2V_T$

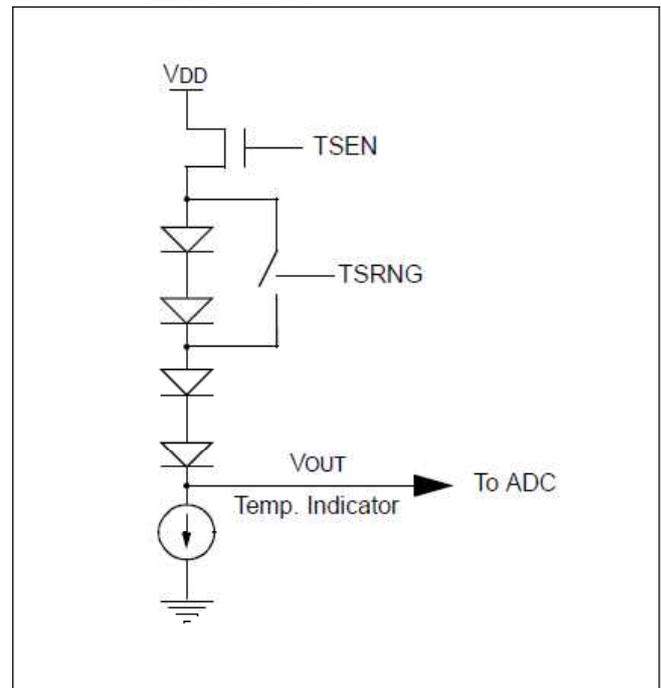
温度検出回路は、固定電圧リファレンス（FVR）モジュールと統合されています。詳細については、セクション13.0「固定電圧リファレンス（FVR）」を参照してください。

この回路は、FVRCONレジスタのTSENビットをセットすることによって有効になります。無効にすると、回路には電流が流れません。

回路は高域または低域で動作します。FVRCONレジスタのTSRNGビットをセットすることによって選択された高レンジは、より広い出力電圧を提供します。これにより、温度範囲全体でより高い分解能が得られますが、部品ごとに一貫性が失われる可能性があります。この範囲では、動作するためにより高いバイアス電圧が必要であるため、より高いVDDが必要です。

低域は、FVRCONレジスタのTSRNGビットをクリアすることによって選択されます。低レンジでは電圧降下が低くなるため、回路を動作させるために必要なバイアス電圧は低くなります。低電圧範囲は、低電圧動作用に提供されています。

図14-1: 温度回路図



### 14.2 最小動作VDD

温度回路が低範囲で動作する場合、デバイスは仕様の範囲内の任意の動作電圧で動作する可能性があります。

温度回路が高範囲で動作する場合、デバイスの動作電圧VDDは、温度回路が正しくバイアスされるように十分に高くなければなりません。

表14-1に、推奨される最小VDDと範囲の設定を示します。

表14-1: 推奨されるVDDと範囲

最小VDD, TSRNG = 1	最小VDD, TSRNG = 0
3.6V	1.8V

### 14.3 温度出力

回路の出力は、内部のアナログ・デジタル・コンバーターを使用して測定されます。チャネルは温度回路出力用に予約されています。詳細については、セクション15.0「アナログ・デジタル・コンバーター（ADC）モジュール」を参照してください。

### 14.4 ADC取得時間

正確な温度測定を行うために、ADC入力マルチプレクサー（選択装置？）が温度インジケータ出力に接続されてから少なくとも200s待つてから変換を実行する必要があります。また、温度表示器の出力が順次変換されるまで、200s待つて必要があります。

表14-2: 温度インジケータに関連付けられているレジスタの概要

名前	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ページ上の登録
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		125

記録: 網掛けのセルは、温度インジケータ・モジュールによって使用されていません。

# PIC16(L)F1508/9

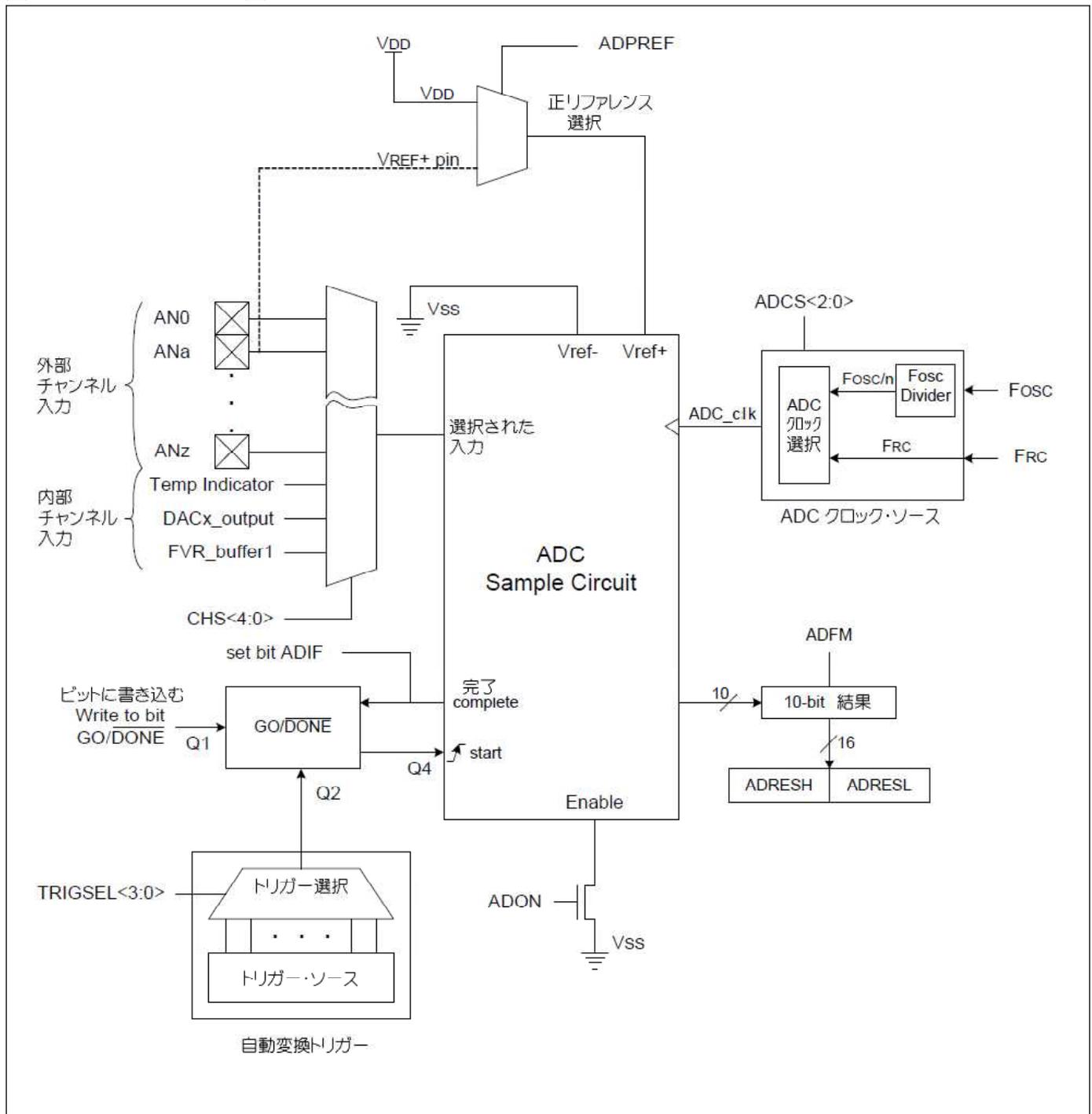
## 15.0 アナログ・デジタル・コンバーター (ADC) モジュール

アナログ・デジタル・コンバーター (ADC) を使用すると、アナログ入力信号をその信号の10ビット・バイナリ表現に変換できます。このデバイスは、単一のサンプル・アンド・ホールド回路に多重化されたアナログ入力を使用します。サンプル・アンド・ホールドの出力は、コンバータの入口に接続されています。コンバータは、逐次比較によって10ビットのバイナリ結果を生成し、変換結果をADC結果レジスタ (ADRESH:ADRESLレジスター・ペア) に格納します。図15-1に、ADCのブロック図を示します。

ADC電圧リファレンス (参照) は、ソフトウェアで選択可能で、内部で生成するか、外部から供給することができます。

ADCは、変換の完了時に割り込みを生成できます。この割り込みは、デバイスをスリープからウェイクアップするために使用できます。

図15-1: ADCブロック図



## 15.1 ADC構成

ADCを構成して使用するときは、次の機能を考慮する必要があります：

- ポート構成
- チャンネル選択
- ADC電圧リファレンス（参照）の選択
- ADC変換クロック・ソース
- 割り込み制御
- 結果のフォーマット

### 15.1.1 ポート構成

ADCは、アナログ信号とデジタル信号の両方を変換するために使用できます。アナログ信号を変換するときは、関連するTRISビットとANSELビットを設定して、I/Oピンをアナログ用に構成する必要があります。詳細については、セクション11.0「I/Oポート」を参照してください。

**注釈：** デジタル入力として定義されているピンのアナログ電圧により、入力バッファに過電流が流れる可能性があります。

### 15.1.2 チャンネル選択

利用可能な15のチャンネル選択があります：

- AN<11:0>ピン
- 温度インジケータ
- DAC1\_output
- FVR\_buffer1

ADCON0レジスタのCHSビットは、どのチャンネルがサンプル・アンド・ホールド回路に接続されているかを決定します。

チャンネルを変更する場合、次の変換を開始する前に遅延（TACQ）が必要です。詳細については、セクション15.2.6「ADC変換手順」を参照してください。

### 15.1.3 ADC電圧リファレンス

ADCモジュールは正と負の電圧リファレンスを使用します。正の参照にはref+のラベルが付けられ、負の参照にはref-のラベルが付けられます。

正の電圧リファレンス（ref+）は、ADCON1レジスタのADPREFビットによって選択されます。正の電圧リファレンス・ソースは：

- VREF+ピン
- VDD

負の電圧リファレンス（ref-）ソースは：

- VSS

### 15.1.4 変換クロック

変換クロックのソースは、ADCON1レジスタのADCSビットを介してソフトウェアで選択できます。7つの可能なクロック・オプションがあります：

- FOSC/2
- FOSC/4
- FOSC/8
- FOSC/16
- FOSC/32
- FOSC/64
- FRC（内部RC発信器）

1ビット変換を完了する時間はTADとして定義されます。図15-2に示すように、1回の完全な10ビット変換には11.5 TAD期間が必要です。

正しく変換するには、適切なTAD仕様を満たす必要があります。詳細については、セクション29.0「電氣的仕様」のADC変換要件を参照してください。表15-1に、適切なADCクロックの選択例を示します。

**注釈：** FRCを使用しない限り、システムクロック周波数を変更すると、ADCクロック周波数に変更され、ADCの結果に悪影響を与える可能性があります。

# PIC16(L)F1508/9

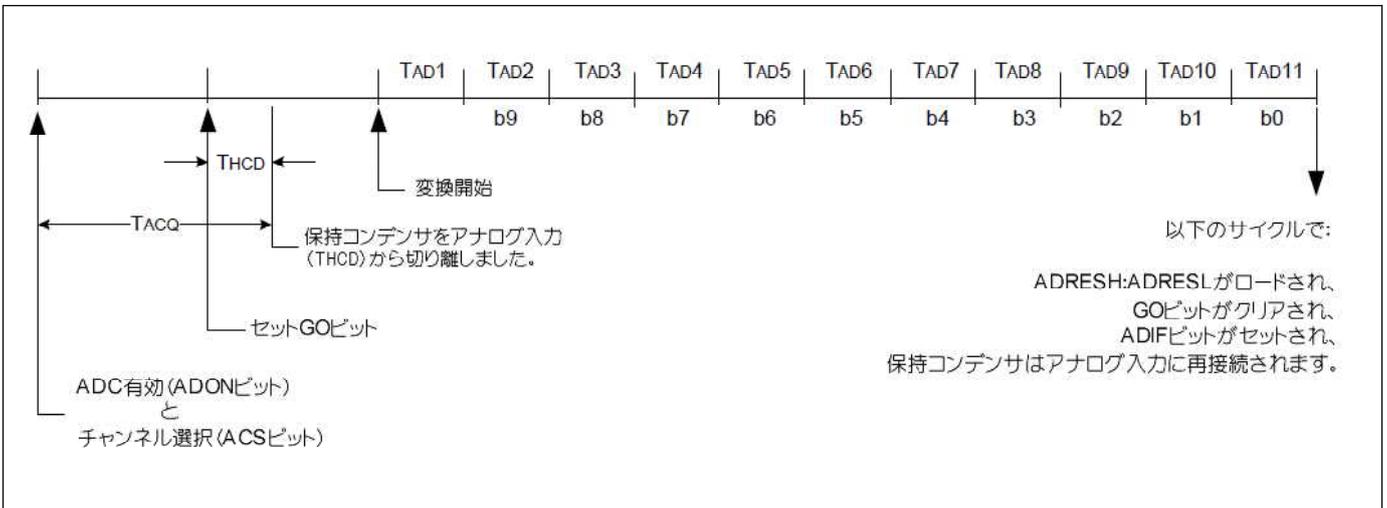
表15-1: ADCクロック周期(TAD)とデバイスの動作周波数

ADCクロック周期(TAD)		デバイス周波数(Fosc)				
ADCクロック ソース	ADCS<2:0>	20 MHz	16 MHz	8 MHz	4 MHz	1 MHz
Fosc/2	000	100 ns	125 ns	250 ns	500 ns	2.0 μs
Fosc/4	100	200 ns	250 ns	500 ns	1.0 μs	4.0 μs
Fosc/8	001	400 ns	500 ns	1.0 μs	2.0 μs	8.0 μs
Fosc/16	101	800 ns	1.0 μs	2.0 μs	4.0 μs	16.0 μs
Fosc/32	010	1.6 μs	2.0 μs	4.0 μs	8.0 μs	32.0 μs
Fosc/64	110	3.2 μs	4.0 μs	8.0 μs	16.0 μs	64.0 μs
FRC	x11	1.0 ~ 6.0 μs	1.0 ~ 6.0 μs	1.0 ~ 6.0 μs	1.0 ~ 6.0 μs	1.0 ~ 6.0 μs

記録： 影付きのセルは推奨範囲外です。

注釈： FRCクロック・ソースを使用する場合のTAD周期は、指定された範囲内に入る可能性があります（TADパラメーターを参照）。Foscベースのクロックソースを使用する場合のTAD周期は、より正確なTAD周期に設定できます。ただし、デバイスをスリープ・モードにして変換を実行する場合は、FRCクロック・ソースを使用する必要があります。

図15-2: アナログからデジタルへの変換TADサイクル



## 15.1.5 割り込み

ADCモジュールは、アナログからデジタルへの変換の完了時に割り込みを生成する機能を可能にします。ADC割り込みフラグは、PIR1レジスタのADIFビットです。ADC割り込みイネーブルは、PIE1レジスタのADIEビットです。ADIFビットはソフトウェアでクリアする必要があります。

- 注1:** ADIFビットは、ADC割り込みが有効かどうかに関係なく、すべての変換の完了時にセットされます。
- 2:** ADCは、FRC発振器が選択されている場合にのみスリープ中に動作します。

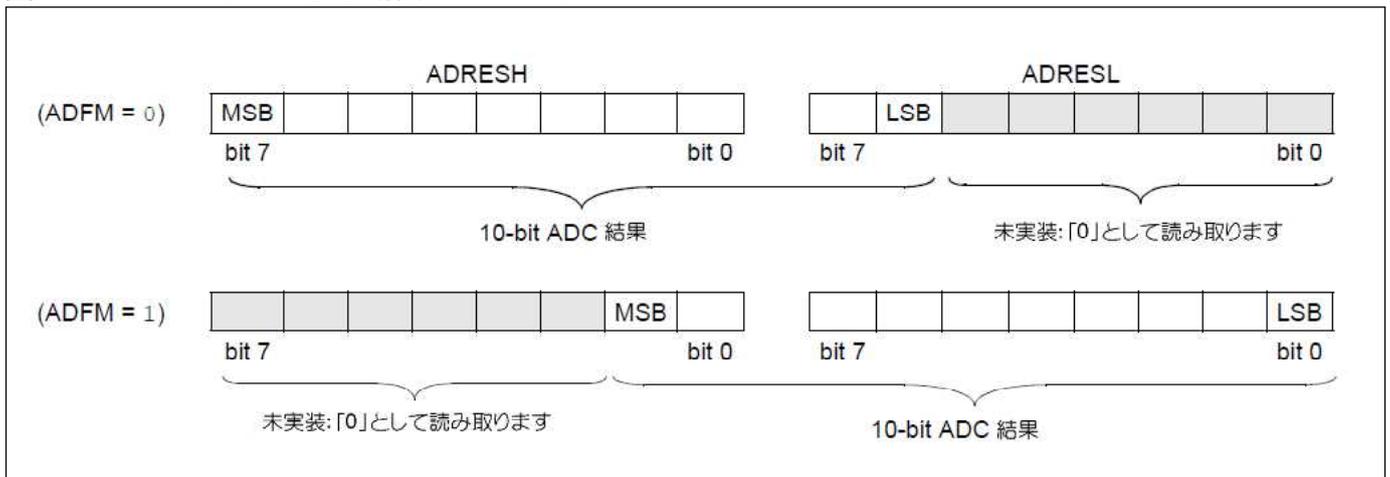
この割り込みは、デバイスの動作中またはスリープ中に生成される可能性があります。デバイスがスリープ状態の場合、割り込みによってデバイスがウェイクアップします。スリープから復帰すると、**SLEEP**命令に続く次の命令が常に実行されます。ユーザーがスリープからウェイクアップしてインライン・コード実行を再開しようとしている場合は、INTCONレジスタのGIEビットとPEIEビットを無効にする必要があります。INTCONレジスタのGIEビットとPEIEビットが有効になっている場合、実行は割り込みサービス・ルーチンに切り替わります。

## 15.1.6 結果のフォーマット

10ビットADC変換結果は、左揃えまたは右揃えの2つの形式で提供できます。ADCON1レジスタのADFMビットは出力フォーマットを制御します。

図15-3に、2つの出力フォーマットを示します。

**図15-3: 10ビットADC変換結果フォーマット**



## 15.2 ADCの動作

### 15.2.1 変換の開始

ADCモジュールを有効にするには、ADCON0レジスタのADONビットを「1」に設定する必要があります。ADCON0レジスタのGO/DONEビットを「1」に設定すると、アナログ・デジタル変換が開始されます。

**注意：** GO/DONEビットは、ADCをオンにするのと同じ命令で設定しないでください。セクション15.2.6「ADC変換手順」を参照してください。

### 15.2.2 変換の完了

変換が完了すると、ADCモジュールは：

- GO/DONEビットをクリアします
- ADIF割り込みフラグ・ビットを設定します
- ADRESHおよびADRESLレジスタを新しい変換結果で更新します

### 15.2.3 変換の終了

変換が完了する前に終了する必要がある場合は、ソフトウェアでGO/DONEビットをクリアできます。ADRESHおよびADRESLレジスタは、部分的に完全なアナログ・デジタル変換サンプルで更新されます。不完全なビットは、変換された最後のビットと一致します。

**注意：** デバイスのリセットにより、すべてのレジスタが強制的にリセット状態になります。したがって、ADCモジュールはオフになり、保留中の変換はすべて終了します。

### 15.2.4 スリープ中のADC動作

ADCモジュールはスリープ中に動作できます。これには、ADCクロック・ソースをFRCオプションに設定する必要があります。スリープ中にADC変換を実行すると、システム・ノイズを減らすことができます。ADC割り込みが有効になっている場合、変換が完了すると、デバイスはスリープからウェイクアップします。ADC割り込みが無効になっている場合、ADONビットはセットされたままですが、変換が完了するとADCモジュールはオフになります。

ADCクロック・ソースがFRC以外の場合、ADONビットはセットされたままですが、SLEEP命令により現在の変換が中止され、ADCモジュールがオフになります。

### 15.2.5 自動変換トリガー

自動変換トリガーにより、ソフトウェアの介入なしに定期的なADC測定が可能になります。選択したソースの立ち上がりエッジが発生すると、ハードウェアによってGO/DONEビットが設定されます。

自動変換トリガー・ソースは、ADCON2レジスタのTRIGSEL<3:0>ビットで選択されます。

自動変換トリガーを使用しても、適切なADCタイミングが保証されません。ADCのタイミング要件が満たされていることを確認するのはユーザーの責任です。

自動変換ソースについては、表15-2を参照してください。

表15-2: 自動変換ソース

周辺ソース	信号名
タイマー0	T0_overflow
タイマー1	T1_overflow
タイマー2	T2_match
コンパレータC1	C1OUT_sync
コンパレータC2	C2OUT_sync
CLC1	LC1_out
CLC2	LC2_out
CLC3	LC3_out
CLC4	LC4_out

### 15.2.6 ADC変換手順

これは、ADCを使用してアナログ・デジタル変換を実行する手順の例です：

1. ポートを構成する：
  - ピン出カドライバを無効にします（TRISレジスターを参照）
  - ピンをアナログとして構成します（ANSELレジスターを参照）
2. ADCモジュールを構成します：
  - ADC変換クロックを選択
  - 電圧リファレンスを構成する
  - ADC入力チャンネルを選択
  - ADCモジュールをオンにします
3. ADC割り込みの設定（オプション）：
  - ADC割り込みフラグをクリアする
  - ADC割り込みを有効にする
  - 周辺割り込みを有効にする
  - グローバル割り込みを有効にする<sup>(1)</sup>
4. 必要な取得時間を待つ<sup>(2)</sup>。
5.  $\overline{GO/DONE}$ ビットをセットして変換を開始します。
6. 次のいずれかによってADC変換が完了するのを待ちます：
  - $\overline{GO/DONE}$ ビットのポーリング
  - ADC割り込みを待機しています（割り込みが有効）
7. ADC結果を読み取ります。
8. ADC割り込みフラグをクリアします（割り込みが有効な場合に必要）。

**注1：** ユーザーがスリープからウェイクアップしてインライン・コード実行を再開しようとしている場合は、グローバル割り込みを無効にすることができます。

**2：** セクション15.4「ADC取得要件」を参照してください。

### 例15-1: ADC変換

```

;This code block configures the ADC
;for polling, Vdd and Vss references, FRC
;oscillator and AN0 input.
;
;Conversion start & polling for completion
;are included.
;
BANKSEL  ADCON1          ;
MOVLW    B'11110000'    ;Right justify, FRC
                                ;oscillator
MOVWF    ADCON1          ;Vdd and Vss Vref+
BANKSEL  TRISA           ;
BSF      TRISA,0         ;Set RA0 to input
BANKSEL  ANSEL           ;
BSF      ANSEL,0         ;Set RA0 to analog
                                BANKSEL WPUA
BCF      WPUA,0          ;Disable weak
                                pull-up on RA0
BANKSEL  ADCON0          ;
MOVLW    B'00000001'    ;Select channel AN0
MOVWF    ADCON0          ;Turn ADC On
CALL     SampleTime      ;Acquisition delay
BSF      ADCON0,ADGO     ;Start conversion
BTFSC    ADCON0,ADGO     ;Is conversion done?
GOTO     $-1              ;No, test again
BANKSEL  ADRESH          ;
MOVF     ADRESH,W        ;Read upper 2 bits
MOVWF    RESULTHI        ;store in GPR space
BANKSEL  ADRESL          ;
MOVF     ADRESL,W        ;Read lower 8 bits
MOVWF    RESULTLO        ;Store in GPR space

```

# PIC16(L)F1508/9

## 15.3 レジスター定義: ADC制御

### レジスター15-1: ADCON0: ADCコントロール・レジスター0

U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	CHS<4:0>					GO/DONE	ADON
bit 7							bit 0

<b>記録:</b>		
R =読み取り可能ビット	W =書き込み可能ビット	U =実装されていないビット、「0」として読み取られる
u =ビットは変更されていません	x =ビットは不明です	-n/n =PORでの値およびBOR/他のすべてのリセットでの値
[1]=ビットが設定されている	[0]=ビットがクリアされている	

- bit 7 **未実装:** 「0」として読み取ります
- bit 6~2 **CHS<4:0>:** アナログ・チャンネル選択ビット
  - 00000 = AN0
  - 00001 = AN1
  - 00010 = AN2
  - 00011 = AN3
  - 00100 = AN4
  - 00101 = AN5
  - 00110 = AN6
  - 00111 = AN7
  - 01000 = AN8
  - 01001 = AN9
  - 01010 = AN10
  - 01011 = AN11
  - 01100 = 予約済み。 チャンネルが接続されていません。
  - 
  - 
  - 
  - 11100 = 予約済み。 チャンネルが接続されていません。
  - 11101 = 温度インジケータ<sup>(1)</sup>
  - 11110 = DAC (デジタルTOアナログ・コンバーター) <sup>(2)</sup>
  - 11111 = FVR (固定電圧リファレンス) バッファ1出力<sup>(3)</sup>
- bit 1 **GO/DONE:** ADC変換ステータス・ビット
  - 1 = ADC変換サイクルが進行中。このビットを設定すると、ADC変換サイクルが開始されます。このビットは、ADC変換が完了すると、ハードウェアによって自動的にクリアされます。
  - 0 = ADC変換が完了/進行中ではありません
- bit 0 **ADON:** ADC許可ビット
  - 1 = ADCが有効になっている
  - 0 = ADCは無効であり、動作電流を消費しません

**注1:** 詳細については、セクション14.0「温度インジケータ・モジュール」を参照してください。  
**2:** 詳細については、セクション16.0「5ビット・デジタル・アナログ・コンバーター (DAC) モジュール」を参照してください。  
**3:** 詳細については、セクション13.0「固定電圧リファレンス (FVR)」を参照してください。

## レジスタ-15-2: ADCON1: ADC制御レジスタ-1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
ADFM	ADCS<2:0>			—	—	ADPREF<1:0>	
bit 7							bit 0

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている

- bit 7      **ADFM**: ADC結果のフォーマット選択ビット  
1 = 右寄せ。変換結果がロードされると、ADRESHの最上位6ビットが「0」に設定されます。  
0 = 左寄せ。変換結果がロードされると、ADRESLの6つの最下位ビットが「0」に設定されます。
- bit 6~4      **ADCS<2:0>**: ADC変換クロック選択ビット  
000 = Fosc/2  
001 = Fosc/8  
010 = Fosc/32  
011 = FRC (内部RC発振器から供給されるクロック)  
100 = Fosc/4  
101 = Fosc/16  
110 = Fosc/64  
111 = FRC (内部RC発振器から供給されるクロック)
- bit 3~2      **未実装**: 「0」として読み取ります
- bit 1~0      **ADPREF<1:0>**: ADC正電圧リファレンス構成ビット  
00 = VRPOSはVDDに接続  
01 = 予約済み  
10 = VRPOSは外部VREF+ピン<sup>(1)</sup>に接続  
11 = 予約済み

**注1**: 正のリファレンスのソースとしてVREF+ピンを選択する場合、最小電圧仕様が存在することに注意してください。詳細については、セクション29.0「電氣的仕様」を参照してください。

# PIC16(L)F1508/9

## レジスタ15-3: ADCON2: ADC制御レジスタ2

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
TRIGSEL<3:0>(1)				—	—	—	—
bit 7							bit 0

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている

bit 7~4    **TRIGSEL<3:0>**: 自動変換トリガー選択ビット(1)

0000 = 自動変換トリガーが選択されていません  
0001 = 予約済み  
0010 = 予約済み  
0011 = タイマー0。T0\_overflow(2)  
0100 = タイマー1。T1\_overflow(2)  
0101 = タイマー2。T2\_match  
0110 = コンパレータC1。C1OUT\_sync  
0111 = コンパレータC2。C2OUT\_sync  
1000 = CLC1。LC1\_out  
1001 = CLC2。LC2\_out  
1010 = CLC3。LC3\_out  
1011 = CLC4。LC4\_out  
1100 = 予約済み  
1101 = 予約済み  
1110 = 予約済み  
1111 = 予約済み

bit 3~0    **未実装**: 「0」として読み取ります

注1: これはすべてのソースの立ち上がりエッジ・センシティブ（扱いに細心の注意を要する）入力です。

2: シグナルは対応する割り込みフラグも設定します。

## レジスタ-15-4: ADRESH: ADC結果レジスタ・上位(ADRESH) ADFM = 0

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
ADRES<9:2>							
bit 7							bit 0

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
 u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
 「1」=ビットが設定されている      「0」=ビットがクリアされている

bit 7~0      **ADRES<9:2>**: ADC結果レジスタ・ビット  
 10ビット変換結果の上位8ビット

## レジスタ-15-5: ADRESL: ADC結果レジスタ下位(ADRESL) ADFM = 0

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
ADRES<1:0>		—	—	—	—	—	—
bit 7							bit 0

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
 u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
 「1」=ビットが設定されている      「0」=ビットがクリアされている

bit 7~6      **ADRES<1:0>**: ADC結果レジスタ・ビット  
 10ビット変換結果の下位2ビット

bit 5~0      **予約済み**: 使用禁止。

# PIC16(L)F1508/9

## レジスタ15-6: ADRESH: ADC結果レジスタ上位(ADRESH) ADFM = 1

R/W-x/u	R/W-x/u						
—	—	—	—	—	—	ADRES<9:8>	
bit 7							bit 0

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
[1]=ビットが設定されている      [0]=ビットがクリアされている

bit 7~2      予約済み: 使用禁止。  
bit 1~0      **ADRES<9:8>**: ADC結果レジスタ・ビット  
10ビット変換結果の上位2ビット

## レジスタ15-7: ADRESL: ADC結果レジスタ下位(ADRESL) ADFM = 1

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
ADRES<7:0>							
bit 7							bit 0

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
[1]=ビットが設定されている      [0]=ビットがクリアされている

bit 7~0      **ADRES<7:0>**: ADC結果レジスタ・ビット  
10ビット変換結果の下位8ビット

## 15.4 ADC取得要件

ADCが指定された精度を満たすには、電荷保持コンデンサー (CHOLD) が入力チャネル電圧レベルまで完全に充電できるようにする必要があります。アナログ入力モデルを図15-4に示します。ソースインピーダンス (RS) と内部サンプリング・スイッチ (Rss) のインピーダンスは、コンデンサーCHOLDの充電に必要な時間に直接影響します。サンプリング・スイッチ (Rss) のインピーダンスは、デバイス電圧 (VDD) によって変化します。図15-4を参照してください。アナログ・ソースの最大推奨インピーダンスは10 kΩです。

ソースインピーダンスが減少すると、収集時間が減少する可能性があります。アナログ入力チャネルを選択 (または変更) した後、変換を開始する前にADC取得を実行する必要があります。最小取得時間を計算するには、式15-1を使用できます。この式は、1/2 LSBエラーが使用されることを前提としています (ADCの場合は1,024ステップ)。1/2 LSBエラーは、ADCが指定された分解能を満たすために許容される最大エラーです。

### 式15-1: 取得時間の例

仮定: 温度 = 50°C そして 10kΩ 5.0V VDD の外部インピーダンス

$$\begin{aligned} T_{ACQ} &= \text{アンプのセトリング時間} + \text{ホールド・キャパシター充電時間} + \text{温度係数} \\ &= T_{AMP} + T_C + T_{COFF} \\ &= 2\mu\text{s} + T_C + [( \text{温度} - 25^\circ\text{C} ) (0.05\mu\text{s}/^\circ\text{C})] \end{aligned}$$

TCの値は、次の式で概算できます:

$$V_{APPLIED} \left( 1 - \frac{1}{(2^{n+1}) - 1} \right) = V_{CHOLD} \quad ; [1] V_{CHOLD} \text{ は } 1/2 \text{ lsb 以内に充電されます}$$

$$V_{APPLIED} \left( 1 - e^{-\frac{T_C}{RC}} \right) = V_{CHOLD} \quad ; [2] V_{APPLIED} \text{ に対する } V_{CHOLD} \text{ 充電応答}$$

$$V_{APPLIED} \left( 1 - e^{-\frac{T_C}{RC}} \right) = V_{APPLIED} \left( 1 - \frac{1}{(2^{n+1}) - 1} \right) \quad ; [1] \text{ と } [2] \text{ を組み合わせる}$$

注意: ここで  $n = \text{ADCのビット数}$ 。

TCの解決:

$$\begin{aligned} T_C &= -CHOLD(RIC + R_{SS} + R_S) \ln(1/2047) \\ &= -12.5\text{pF}(1\text{k}\Omega + 7\text{k}\Omega + 10\text{k}\Omega) \ln(0.0004885) \\ &= 1.12\mu\text{s} \end{aligned}$$

したがって:

$$\begin{aligned} T_{ACQ} &= 2\mu\text{s} + 1.12\mu\text{s} + [(50^\circ\text{C} - 25^\circ\text{C})(0.05\mu\text{s}/^\circ\text{C})] \\ &= 4.37\mu\text{s} \end{aligned}$$

注1: 基準電圧 (VREF+) は、それ自体が相殺されるため、式に影響を与えません。

注2: 電荷保持コンデンサー (CHOLD) は、各変換後に放電されません。

注3: アナログ・ソースの最大推奨インピーダンスは10kΩです。これは、ピン・リーク仕様を満たすために必要です。

# PIC16(L)F1508/9

図15-4: アナログ入力モデル

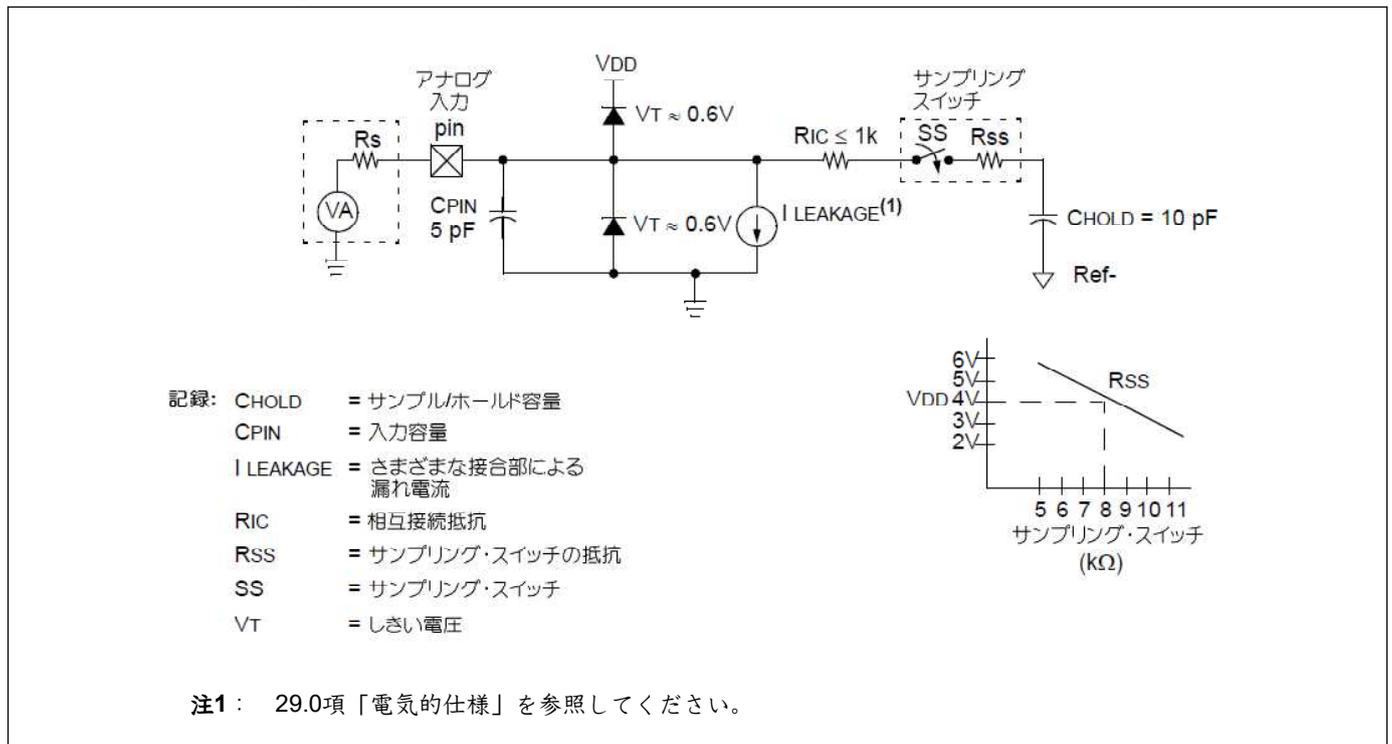


図15-5: ADC伝達関数

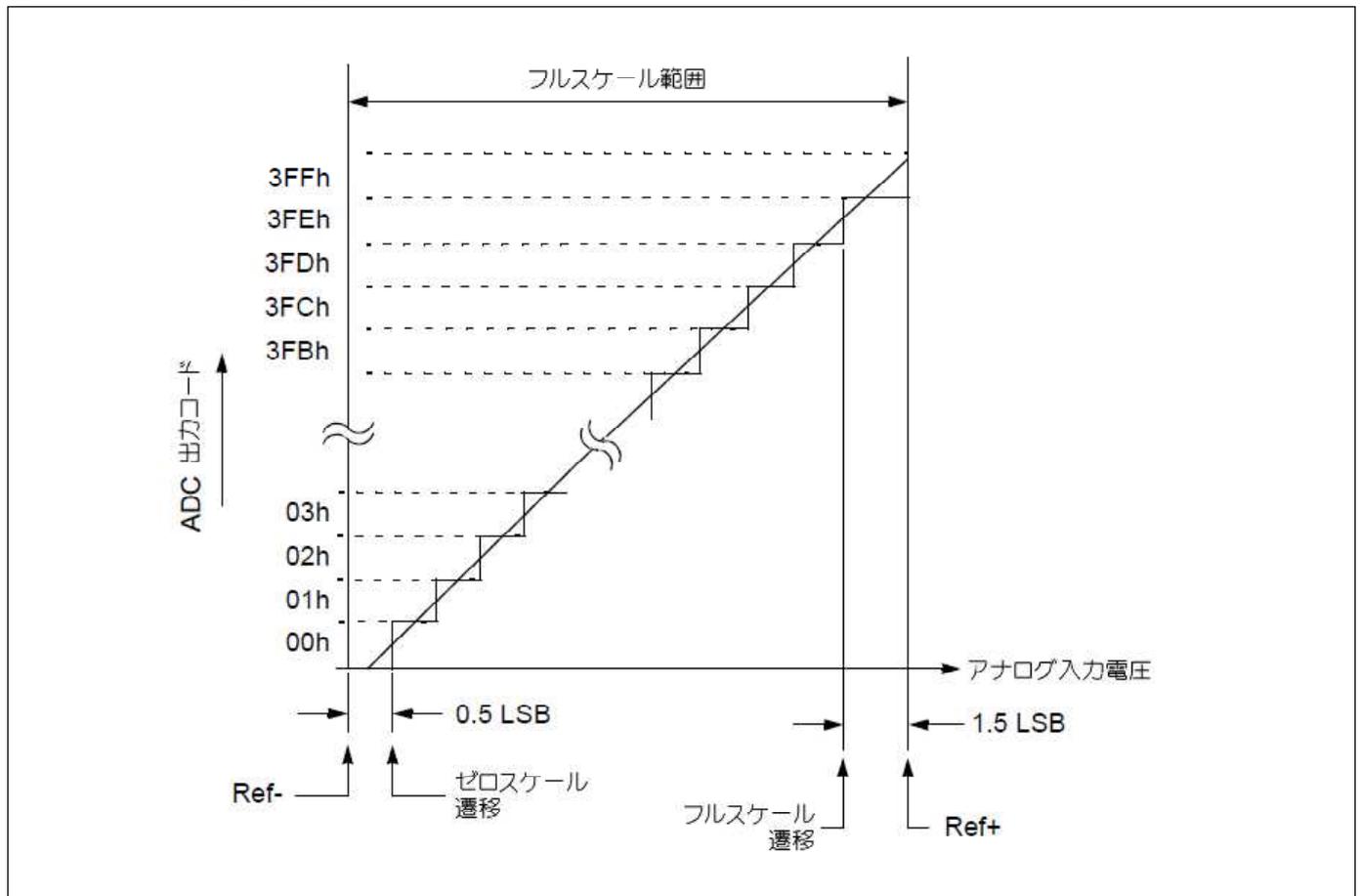


表15-3: ADCに関連付けられているレジスタの概要

名前	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ページ上の登録
ADCON0	—	CHS<4:0>					GO/DONE	ADON	134
ADCON1	ADFM	ADCS<2:0>			—	—	ADPREF<1:0>		135
ADCON2	TRIGSEL<3:0>				—	—	—	—	136
ADRESH	ADC結果レジスタ上位								137, 138
ADRESL	ADC結果レジスタ下位								137, 138
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	110
ANSELB	—	—	ANSB5	ANSB4	—	—	—	—	114
ANSELC	ANSC7	ANSC6	—	—	ANSC3	ANSC2	ANSC1	ANSC0	118
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	75
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	—	TMR2IE	TMR1IE	76
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	—	TMR2IF	TMR1IF	79
TRISA	—	—	TRISA5	TRISA4	— <sup>(1)</sup>	TRISA2	TRISA1	TRISA0	109
TRISB	TRISB7	TRISA5	TRISB5	TRISB4	—	—	—	—	113
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	117
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		125

記録: x = 不明。u = 変更なし。— = 未実装の読み取りは「0」、q = 値は条件によって異なります。網掛けのセルはADCモジュールには使用されません。

注1: 未実装、「1」と読みます。

# PIC16(L)F1508/9

## 16.0 5ビット・デジタル・アナログ・コンバーター (DAC) モジュール

デジタル・アナログ・コンバーターは、32の選択可能な出力レベルを備えた、入力ソースとレシオメトリック（電源電圧が変動すると出力も変動する）な可変電圧リファレンスを提供します。

DACの正の入力ソース (VSOURCE+) はに接続できます：

- 外部 VREF+ピン
- VDD供給電圧

DACの負の入力ソース (VSOURCE-) はに接続できます：

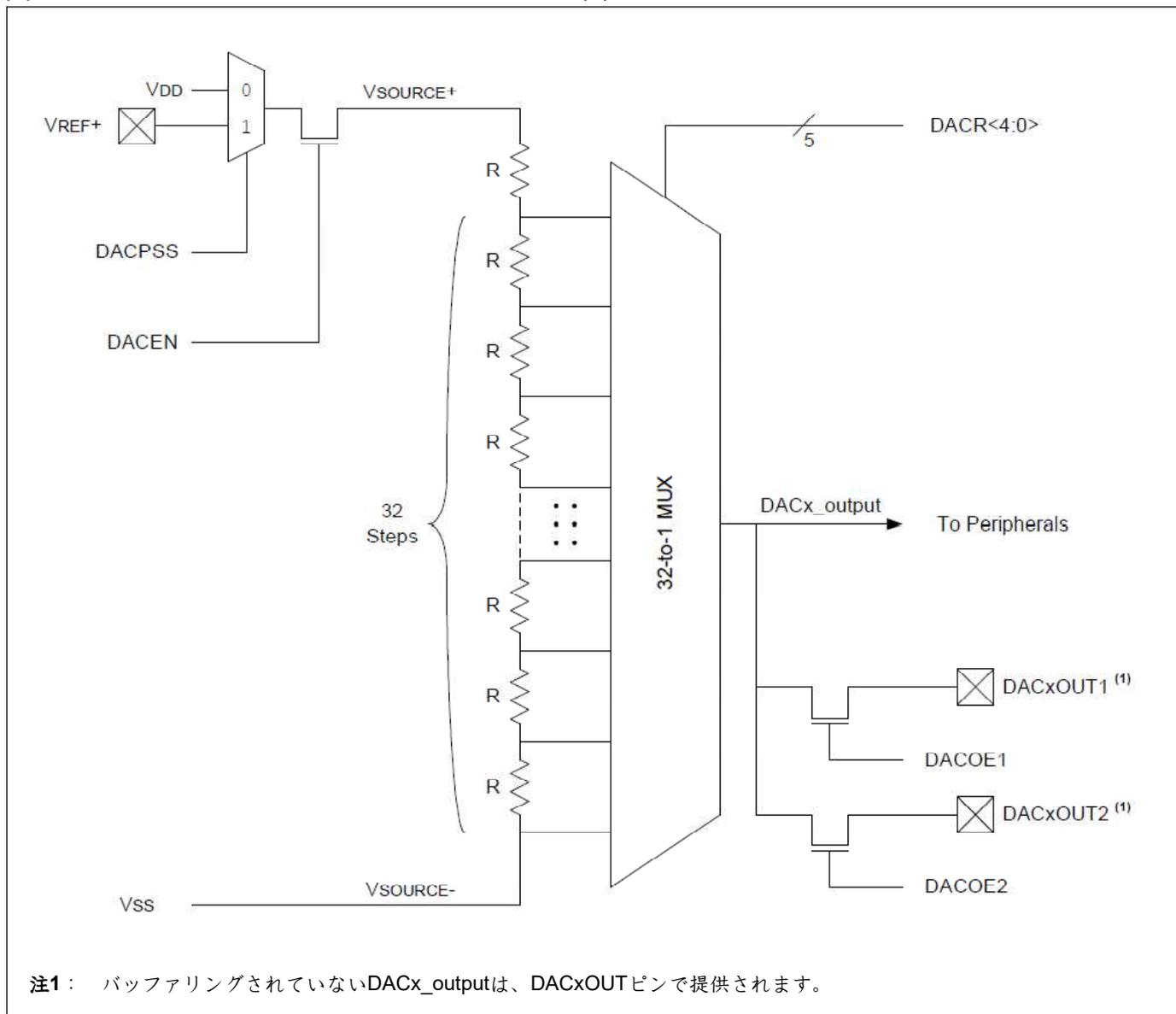
- Vss

DACの出力 (DACx\_output) は、以下の基準電圧として選択できます：

- コンパレーター正入力
- ADC入力チャンネル
- DACxOUT1 ピン
- DACxOUT2 ピン

デジタル・アナログ・コンバーター (DAC) は、DACxCON0レジスタのDACENビットをセットすることで有効にできます。

図16-1: デジタル・アナログ・コンバーター・ブロック図



注1: バッファリングされていないDACx\_outputは、DACxOUTピンで提供されます。

## 16.1 出力電圧の選択

DACには32の電圧レベル範囲があります。32レベルは、ADCON1レジスタのFACE<4:0>ビットで設定されます。

DAC出力電圧は式16-1を使用して決定できます。

## 16.2 レジオメトリック出力レベル

DAC出力値は、抵抗ラダーを使用して導出され、ラダーの両端は正および負の電圧リファレンス入力ソースに接続されています。どちらかの入力ソースの電圧が変動する場合、同様の変動がDAC出力値になります。

ラダー内の個々の抵抗の値は、表29-16に記載されています。

## 16.3 DAC電圧リファレンス出力

バッファリングされていないDAC電圧は、DACxCON0レジスタのそれぞれのDACOEnビットを設定することによってDACxOUTnピンに出力できます。いずれかのDACxOUTnピンで出力するDAC基準電圧を選択すると、そのピンのデジタル出力バッファ、弱いプルアップ、およびデジタル入力しきい値検出器の機能が自動的に上書きされます。

### 式16-1: DAC出力電圧

#### DACEN = 1の場合

$$DACx\_output = \left( (V_{SOURCE+} - V_{SOURCE-}) \times \frac{DACR[4:0]}{2^5} \right) + V_{SOURCE-}$$

注釈： 使用可能な V<sub>SOURCE+</sub> および V<sub>SOURCE-</sub> の選択については、DACxCON0レジスタを参照してください。

DAC基準電圧出力用に構成されているときにDACxOUTnピンを読み取ると、常に「0」が返されます。

注意： バッファなしDAC出力 (DACxOUTn) は、外部負荷を駆動するためのものではありません。

## 16.4 スリープ中の動作

デバイスが割り込みまたはウォッチドッグ・タイマーのタイムアウトによってスリープからウェイクアップする場合、DACxCON0レジスタの内容は影響を受けません。スリープモードでの消費電流を最小限に抑えるには、電圧リファレンスを無効にする必要があります。

## 16.5 リセットの影響

デバイスのリセットは以下に影響します：

- DACxは無効になっています。
- DACx出力電圧はDACxOUTnピンから削除されます。
- DACR<4:0>範囲選択ビットがクリアされます。

# PIC16(L)F1508/9

## 16.6 レジスター定義: DAC制御

### レジスター16-1: DACxCON0: 電圧リファレンス制御レジスター0

R/W-0/0	U-0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	U-0	U-0
DACEN	—	DACOE1	DACOE2	—	DACPSS	—	—
bit 7							bit 0

<b>記録:</b>		
R =読み取り可能ビット	W =書き込み可能ビット	U =実装されていないビット、「0」として読み取られる
u =ビットは変更されていません	x =ビットは不明です	-n/n =PORでの値およびBOR/他のすべてのリセットでの値
[1]=ビットが設定されている	[0]=ビットがクリアされている	

- bit 7 **DACEN**: DAC許可ビット  
1 = DACxが有効になっている  
0 = DACxが無効になっている
- bit 6 **未実装**: 「0」として読み取ります
- bit 5 **DACOE1**: DAC電圧出力許可ビット  
1 = DACx電圧レベルはDACxOUT1ピンに出力されます  
0 = DACx電圧レベルがDACxOUT1ピンから切断されている
- bit 4 **DACOE2**: DAC電圧出力許可ビット  
1 = DACx電圧レベルはDACxOUT2ピンに出力されます  
0 = DACx電圧レベルがDACxOUT2ピンから切断されている
- bit 3 **未実装**: 「0」として読み取ります
- bit 2 **DACPSS**: DAC正ソース選択ビット  
1 = VREF+ ピン  
0 = VDD
- bit 1~0 **未実装**: 「0」として読み取ります

### レジスター16-2: DACxCON1: 電圧リファレンス制御レジスター1

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	DACR<4:0>				—
bit 7							bit 0

<b>記録:</b>		
R =読み取り可能ビット	W =書き込み可能ビット	U =実装されていないビット、「0」として読み取られる
u =ビットは変更されていません	x =ビットは不明です	-n/n =PORでの値およびBOR/他のすべてのリセットでの値
[1]=ビットが設定されている	[0]=ビットがクリアされている	

- bit 7~5 **未実装**: 「0」として読み取ります
- bit 4~0 **DACR<4:0>**: DAC電圧出力選択ビット

### 表16-1: DACモジュールに関連付けられたレジスターの概要

名前	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ページ上の登録
DAC1CON0	DACEN	—	DACOE1	DACOE2	—	DACPSS	—	—	144
DAC1CON1	—	—	—	DACR<4:0>				—	144

記録: — =実装されていない場所、「0」と読みます。網掛けのセルはDACモジュールでは使用されません。

## 17.0 コンパレータ・モジュール

コンパレータ（比較器）は、2つのアナログ電圧を比較し、それらの相対的な大きさをデジタル表示することにより、アナログ回路をデジタル回路に接続するために使用されます。コンパレータは、プログラムの実行とは独立したアナログ機能を提供するため、非常に便利なミックスド・シグナル・ビルディング・ブロック（混合された信号構造）です。アナログ・コンパレータ・モジュールには、次の機能が含まれています：

- 独立したコンパレータ制御
- プログラム可能な入力選択
- コンパレータ出力は内部/外部で利用可能です
- プログラム可能な出力極性
- 変更時の割り込み
- スリープからの目覚め
- プログラム可能な速度/電力の最適化
- PWMシャットダウン
- プログラム可能で固定の電圧リファレンス

## 17.1 コンパレータの概要

図17-2に、アナログ入力レベルとデジタル出力の関係とともに、単一のコンパレータを示します。VIN+ のアナログ電圧が VIN- のアナログ電圧よりも低い場合、コンパレータの出力はデジタル・ロー・レベルになります。VIN+ のアナログ電圧が VIN- のアナログ電圧よりも大きい場合、コンパレータの出力はデジタル・ハイ・レベルになります。

このデバイスで使用可能なコンパレータを表17-1に示します。

表17-1: 利用可能なコンパレータ

デバイス	C1	C2
PIC16(L)F1508	●	●
PIC16(L)F1509	●	●

図17-1: コンパレータ・モジュールの簡略化されたブロック図

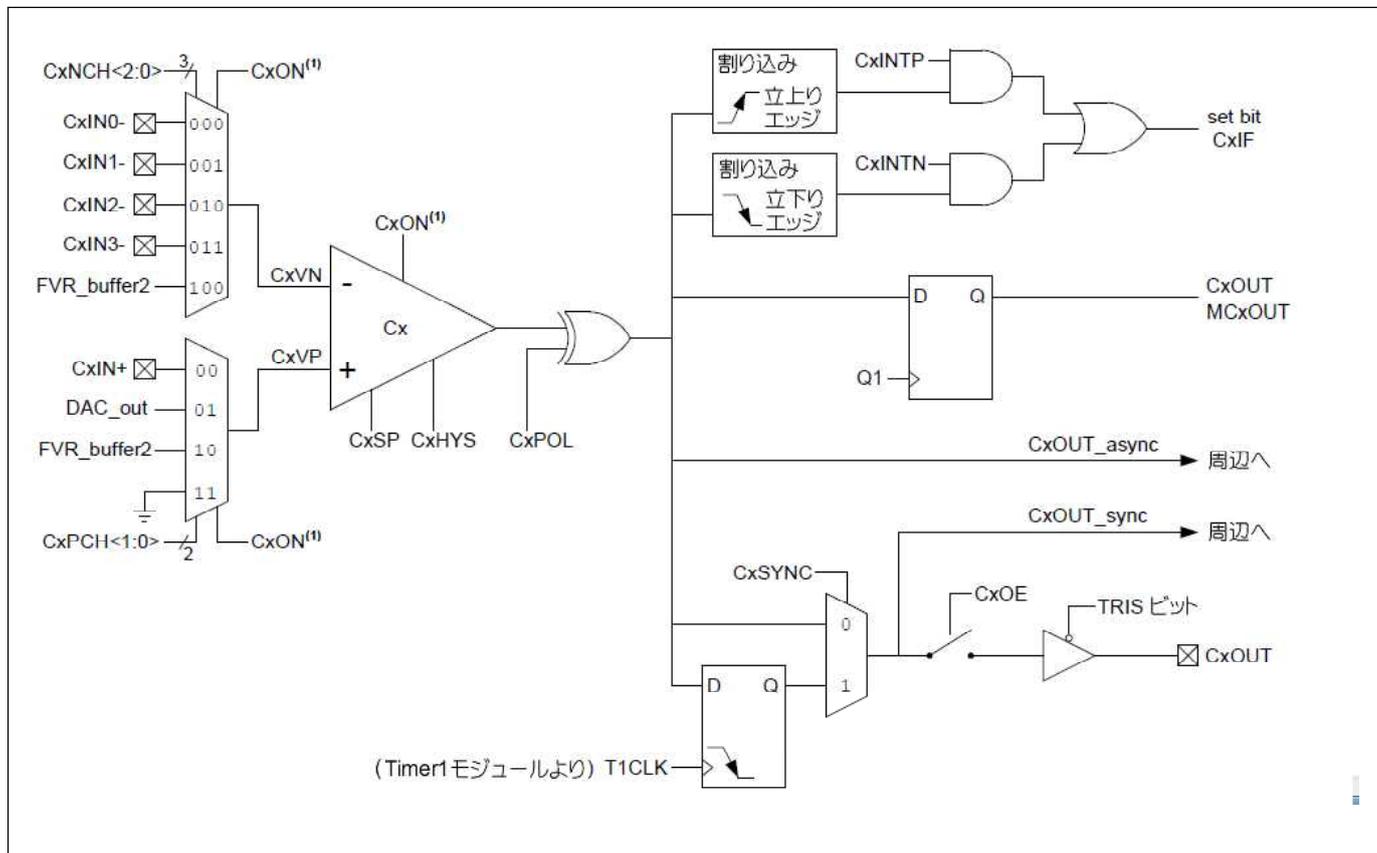
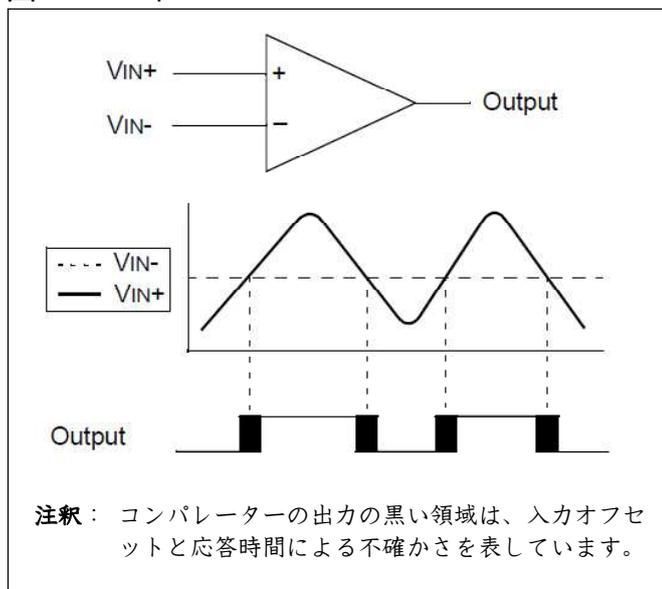


図17-2: 単一のコンパレーター



## 17.2 コンパレーター制御

各コンパレーターには、CMxCON0とCMxCON1の2つの制御レジスターがあります。CMxCON0レジスター（レジスター17-1を参照）には、次の制御ビットとステータス・ビットが含まれています：

- 有効にする
- 出力の選択
- 出力極性
- 速度/電力の選択
- ヒステリシスを有効にする
- 出力同期

CMxCON1レジスター（レジスター17-2を参照）には、次の制御ビットが含まれています。

- 割り込み有効
- 割り込みエッジ極性
- 正入力チャンネル選択
- 負の入力チャンネル選択

### 17.2.1 コンパレータを有効にする

CMxCON0レジスターのCxONビットをセットすると、コンパレーターが動作できるようになります。CxONビットをクリアすると、コンパレーターが無効になり、消費電流が最小になります。

### 17.2.2 コンパレータの正の入力選択

CMxCON1レジスターのCxPCH<1:0>ビットを構成すると、内部電圧リファレンスまたはアナログ・ピンがコンパレータの非反転入力に送られます：

- CxIN+ アナログ・ピン
- DAC1\_output
- FVR\_buffer2
- Vss

固定電圧リファレンス・モジュールの詳細については、セクション13.0「固定電圧リファレンス（FVR）」を参照してください。

DAC入力信号の詳細については、セクション16.0「5ビットデジタル・アナログ・コンバータ（DAC）モジュール」を参照してください。

コンパレータが無効（CxON = 0）になると、すべてのコンパレーター入力が無効になります。

### 17.2.3 コンパレータの負入力選択

CMxCON0レジスターのCxNCH<2:0>ビットは、入力ソースの1つをコンパレーター反転入力に送ります。

注釈: CxIN+ およびCxINx- ピンをアナログ入力として使用するには、適切なビットをANSELレジスターに設定し、対応するTRISビットも設定して出力ドライバーを無効にする必要があります。

### 17.2.4 コンパレータの出力選択

コンパレータの出力は、CMxCON0レジスターのCxOUTビットまたはCMOUTレジスターのMCxOUTビットのいずれかを読み取ることで監視できます。出力を外部接続で使用できるようにするには、次の条件が満たされている必要があります：

- CMxCON0レジスタのCxOEビットをセットする必要があります
- 対応するTRISビットをクリアする必要があります
- CMxCON0レジスタのCxONビットをセットする必要があります

同期コンパレーター出力信号（CxOUT\_sync）は、次の周辺で使用できます：

- 構成可能なロジック・セル（CLC）
- アナログ・デジタル・コンバータ（ADC）
- タイマー1

非同期コンパレーター出力信号（CxOUT\_async）は、次の周辺で使用できます。：

- 相補型波形発生器（CWG）

注1: CMxCON0レジスターのCxOEビットは、ポート・データ・ラッチを上書きします。CMxCON0レジスターのCxONビットを設定しても、ポート上書きには影響しません。

2: コンパレータの内部出力は各命令サイクルでラッチされます。特に指定のない限り、外部出力はラッチされません。

## 17.2.5 コンパレーターの出力極性

コンパレーターの出力を反転することは、コンパレーターの入力を交換することと機能的に同等です。CMxCON0レジスタのCxPOLビットをセットすることにより、コンパレーター出力の極性を反転させることができます。CxPOLビットをクリアすると、出力が反転しません。

表17-2に、極性制御を含む出力状態と入力条件を示します。

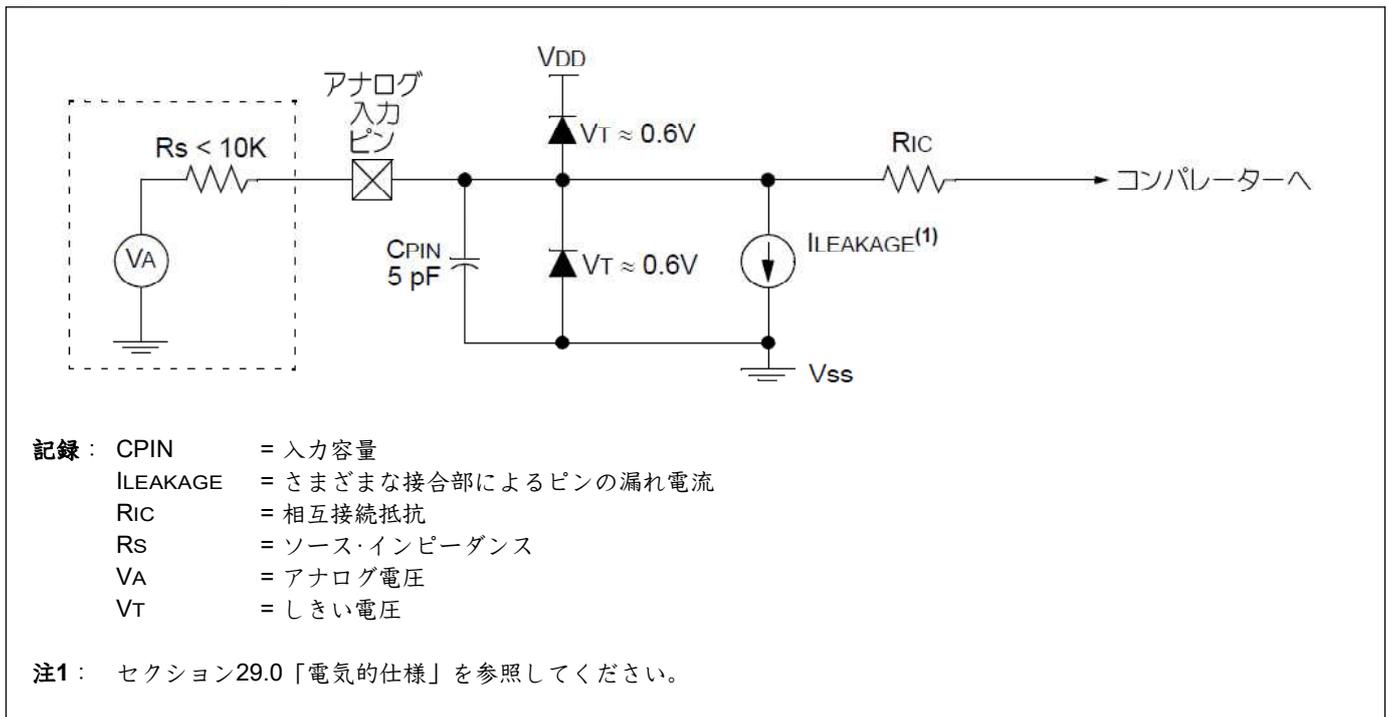
表17-2: コンパレーターの出力状態と入力条件

入力条件	CxPOL	CxOUT
$CxVN > CxVP$	0	0
$CxVN < CxVP$	0	1
$CxVN > CxVP$	1	1
$CxVN < CxVP$	1	0

## 17.2.6 コンパレーターの速度/電力 選択

速度と電力のトレードオフ（二つの仕方・在り方の間の関係）は、CxSP制御ビットを使用してプログラムの実行中に最適化できます。このビットのデフォルト状態は「1」で、通常速度モードを選択します。CxSPビットを「0」にクリアすることにより、コンパレーターの伝搬遅延を遅くする代わりに、デバイスの消費電力を最適化できます。

図17-3: アナログ入力モデル



## 17.3 アナログ入力接続に関する考慮事項

アナログ入力の簡略化した回路を図17-3に示します。アナログ入力ピンはデジタル入力と接続を共有しているため、VDDおよびVSSに逆バイアスされたESD保護ダイオードがあります。したがって、アナログ入力はVSSとVDDの間にある必要があります。入力電圧がこの範囲からいずれかの方向に0.6Vを超えて逸脱すると、ダイオードの1つが順方向にバイアスされ、ラッチアップが発生する可能性があります。

アナログ・ソースには、10 kΩの最大ソース・インピーダンスが推奨されます。また、コンデンサーやツェナー・ダイオードなど、アナログ入力ピンに接続されている外部コンポーネントには、導入される不正確さを最小限に抑えるために、漏れ電流が非常に少ない必要があります。

**注1:** PORTレジスタを読み取る場合、アナログ入力として構成されたすべてのピンは「0」として読み取られます。デジタル入力として構成されたピンは、入力仕様に従ってアナログ入力として変換されます。

**2:** デジタル入力として定義されたピンのアナログレベルにより、入力バッファが指定されたよりも多くの電流を消費する可能性があります。

## 17.4 コンパレータ・ヒステリシス

選択可能な量の分離電圧を各コンパレータの入力ピンに追加して、動作全体にヒステリシス機能を提供できます。ヒステリシスは、CMxCON0レジスタのCxHYSビットを設定することで有効になります。

詳細については、セクション29.0「電氣的仕様」を参照してください。

## 17.5 タイマー1ゲート操作

コンパレータ動作からの出力は、タイマー1のゲート制御のソースとして使用できます。詳細については、セクション19.6「タイマー1ゲート」を参照してください。この機能は、アナログ・イベントの期間または間隔のタイミングを計るのに役立ちます。

コンパレータ出力をタイマー1に同期させることをお勧めします。これにより、コンパレータの変更が発生している間、タイマー1がインクリメントされなくなります。

### 17.5.1 コンパレータ出力同期

CMxCON0レジスタのCxSYNCビットをセットすることにより、Cxコンパレータからの出力をタイマー1と同期させることができます。

有効にすると、コンパレータ出力はタイマー1ソース・クロックの立ち下がりエッジでラッチされます。タイマー1でプリスケラを使用する場合、コンパレータ出力はプリスケリング機能の後にラッチされます。競合状態を防ぐために、コンパレータ出力はタイマー1クロック・ソースの立ち下がりエッジでラッチされ、タイマー1はクロック・ソースの立ち上がりエッジでインクリメントします。詳細については、コンパレータのブロック図（図17-2）およびタイマー1のブロック図（図19-1）を参照してください。

## 17.6 コンパレータ割り込み

各コンパレータのコンパレータの出力値が変化すると、割り込みが発生する可能性があり、立ち上がりエッジ検出器と立ち下がりエッジ検出器が存在します。

いずれかのエッジ検出器がトリガーされ、関連する許可ビット（CMxCON1レジスタのCxINTPおよび/またはCxINTNビット）が設定されると、対応する割り込みフラグ・ビット（PIR2レジスタのCxIFビット）が設定されます。

割り込みを有効にするには、次のビットを設定する必要があります：

- CMxCON0レジスタのCxON、CxPOL、およびCxSPビット
- PIE2レジスタのCxIEビット
- CMxCON1レジスタのCxINTPビット（立ち上がりエッジ検出用）
- CMxCON1レジスタのCxINTNビット（立ち下がりエッジ検出用）
- INTCONレジスタのPEIEビットとGIEビット

関連する割り込みフラグ・ビット、PIR2レジスタのCxIFビットは、ソフトウェアでクリアする必要があります。このフラグがクリアされている間に別のエッジが検出された場合でも、シーケンスの最後にフラグが設定されます。

**注釈：** コンパレータは無効ですが、CMxCON0レジスタのCxPOLビットで出力極性を変更するか、CMxCON0レジスタのCxONビットでコンパレータのオン/オフを切り替えることにより、割り込みを生成できます。

## 17.7 コンパレータの応答時間

コンパレータの出力は、入力ソースの変更後または新しい基準電圧の選択後、一定期間不確定です。この期間は、応答時間と呼ばれます。コンパレータの応答時間は、電圧リファレンスのセトリング時間（出力が最終値に落ち着くまでの時間）とは異なります。したがって、コンパレータ入力の変化に対する合計応答時間を決定するときは、これらの両方の時間を考慮する必要があります。詳細については、セクション29.0「電氣的仕様」のコンパレータおよび電圧リファレンス仕様を参照してください。

## 17.8 レジスタ定義: コンパレータ制御

### レジスタ17-1: CMxCON0: コンパレータCx制御レジスタ-0

R/W-0/0	R-0/0	R/W-0/0	R/W-0/0	U-0	R/W-1/1	R/W-0/0	R/W-0/0
CxON	CxOUT	CxOE	CxPOL	—	CxSP	CxHYS	CxSYNC
bit 7							bit 0

<b>記録:</b>		
R =読み取り可能ビット	W =書き込み可能ビット	U =実装されていないビット、「0」として読み取られる
u =ビットは変更されていません	x =ビットは不明です	-n/n =PORでの値およびBOR/他のすべてのリセットでの値
[1]=ビットが設定されている	[0]=ビットがクリアされている	

- bit 7 **CxON**: コンパレータ許可ビット  
 1 = コンパレータが有効になっている  
 0 = コンパレータが無効になっており、有効電力を消費していません
- bit 6 **CxOUT**: コンパレータ出力ビット  
**CxPOL = 1**の場合 (極性が反転):  
 1 =  $CxVP < CxVN$   
 0 =  $CxVP > CxVN$   
**CxPOL = 0** (非反転極性)の場合:  
 1 =  $CxVP > CxVN$   
 0 =  $CxVP < CxVN$
- bit 5 **CxOE**: コンパレータ出力許可ビット  
 1 = CxOUTはCxOUTピンに存在します。実際にピンを駆動するには、関連するTRISビットをクリアする必要があります。CxONの影響を受けません。  
 0 = CxOUTは内部のみです
- bit 4 **CxPOL**: コンパレータ出力極性選択ビット  
 1 = コンパレータ出力が反転します  
 0 = コンパレータ出力は反転されません
- bit 3 **未実装**: 「0」として読み取ります
- bit 2 **CxSP**: コンパレータ速度/電力選択ビット  
 1 = 通常電力、高速のコンパレータモード  
 0 = 低電力、低速のコンパレータモード
- bit 1 **CxHYS**: コンパレータヒステリシス許可ビット  
 1 = コンパレータヒステリシス有効  
 0 = コンパレータヒステリシス無効
- bit 0 **CxSYNC**: コンパレータ出力同期モードビット  
 1 = タイマー1およびI/Oピンへのコンパレータ出力は、タイマー1クロックソースの変更に同期しています。タイマー1クロックソースの立ち下がりがエッジで出力が更新されました。  
 0 = タイマー1とI/Oピンへのコンパレータ出力は非同期です

# PIC16(L)F1508/9

## レジスタ-17-2: CMxCON1: コンパレータ-Cx制御レジスタ-1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
CxINTP	CxINTN	CxPCH<1:0>		—	CxNCH<2:0>		
bit 7							bit 0

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている

- bit 7      **CxINTP**: 正方向エッジ許可ビットでのコンパレータ割り込み  
1 = CxIF割り込みフラグは、CxOUTビットの立ち上がりエッジで設定されます。  
0 = CxOUTビットの立ち上がりエッジに割り込みフラグは設定されません
- bit 6      **CxINTN**: 負方向エッジ許可ビットでのコンパレータ割り込み  
1 = CxIF割り込みフラグは、CxOUTビットの立ち下がりエッジで設定されます。  
0 = CxOUTビットの立ち下がりエッジに割り込みフラグは設定されません
- bit 5~4      **CxPCH<1:0>**: コンパレータの正の入力チャネル選択ビット  
11 = CxVPはVssに接続します  
10 = CxVPはFVR電圧リファレンスに接続します  
01 = CxVPはDAC電圧リファレンスに接続します  
00 = CxVPはCxIN +ピンに接続します
- bit 3      **未実装**: 「0」として読み取ります
- bit 2~0      **CxNCH<2:0>**: コンパレータの負の入力チャネル選択ビット  
111 = 予約済み  
110 = 予約済み  
101 = 予約済み  
100 = CxVNはFVR電圧リファレンスに接続します  
011 = CxVNはCxIN3ピンに接続します  
010 = CxVNはCxIN2ピンに接続します  
001 = CxVNはCxIN1-ピンに接続します  
000 = CxVNはCxIN0-ピンに接続します

## レジスタ-17-3: CMOUT: コンパレータ出力レジスタ

U-0	U-0	U-0	U-0	U-0	U-0	R-0/0	R-0/0
—	—	—	—	—	—	MC2OUT	MC1OUT
bit 7							bit 0

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている

- bit 7~2      **未実装**: 「0」として読み取ります
- bit 1      **MC2OUT**: C2OUTビットのミラーコピー
- bit 0      **MC1OUT**: C1OUTビットのミラーコピー

表17-3: コンパレータ・モジュールに関連付けられたレジスターの概要

名前	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ページ上の登録
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	110
ANSELC	ANSC7	ANSC6	—	—	ANSC3	ANSC2	ANSC1	ANSC0	118
CM1CON0	C1ON	C1OUT	C1OE	C1POL	—	C1SP	C1HYS	C1SYNC	149
CM2CON0	C2ON	C2OUT	C2OE	C2POL	—	C2SP	C2HYS	C2SYNC	149
CM1CON1	C1NTP	C1INTN	C1PCH<1:0>		—	C1NCH<2:0>			150
CM2CON1	C2NTP	C2INTN	C2PCH<1:0>		—	C2NCH<2:0>			150
CMOUT	—	—	—	—	—	—	MC2OUT	MC1OUT	150
DAC1CON0	DACEN	—	DACOE1	DACOE2	—	DACPSS	—	—	144
DAC1CON1	—	—	—	DACR<4:0>				—	144
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		125
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	75
PIE2	OSFIE	C2IE	C1IE	—	BCL1IE	NCO1IE	—	—	77
PIR2	OSFIF	C2IF	C1IF	—	BCL1IF	NCO1IF	—	—	80
PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	109
PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	117
LATA	—	—	LATA5	LATA4	—	LATA2	LATA1	LATA0	110
LATC	LATC7	LATC6	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	117
TRISA	—	—	TRISA5	TRISA4	— <sup>(1)</sup>	TRISA2	TRISA1	TRISA0	109
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	117

記録： — =実装されていない場所、「0」と読みます。網掛けのセルはコンパレータ・モジュールでは使用されません。

注1： 未実装、「1」と読みます。

# PIC16(L)F1508/9

## 18.0 タイマー0モジュール

タイマー0モジュールは、次の機能を備えた8ビットのタイマー/カウンターです：

- 8ビット・タイマ/カウンター・レジスタ (TMR0)
- 3ビット・プリスケaler (ウォッチドッグ・タイマーに依存しない)
- プログラム可能な内部または外部クロック・ソース
- プログラム可能な外部クロック・エッジの選択
- オーバーフロー時に割り込み
- TMR0はタイマー1のゲートに使用できます

図18-1に、タイマー0モジュールのブロック図を示します。

### 18.1 タイマー0の操作

タイマー0モジュールは、8ビット・タイマーまたは8ビット・カウンターとして使用できます。

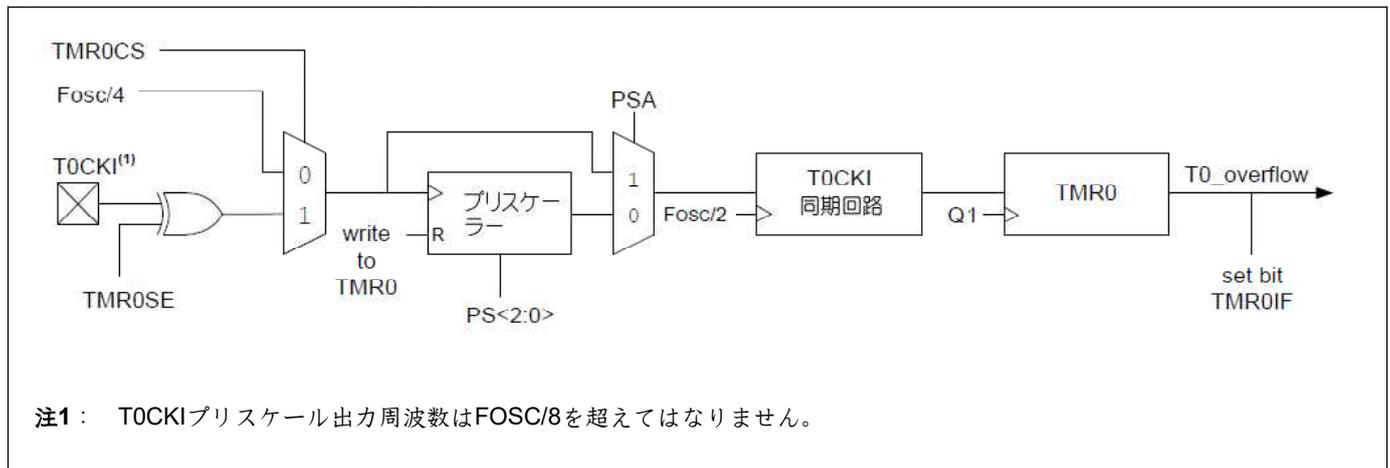
#### 18.1.1 8ビット・タイマー・モード

タイマー0モジュールは、プリスケalerなしで使用された場合、命令サイクルごとにインクリメントします。OPTION\_REGレジスタのTMR0CSビットをクリアすると、8ビット・タイマー・モードが選択されます。

TMR0が書き込まれると、書き込み直後の2命令サイクルの間インクリメントが禁止されます。

**注意：** TMR0が書き込まれるときの2命令サイクルの遅延を考慮して、TMR0レジスタに書き込まれる値を調整できます。

図18-1: タイマー0のブロック図



#### 18.1.2 8ビット・カウンター・モード

8ビット・カウンター・モードでは、タイマー0モジュールはT0CKIピンの立ち上がりエッジまたは立ち下がりエッジごとにインクリメントします。

OPTION\_REGレジスタのTMR0CSビットを「1」に設定することにより、T0CKIピンを使用する8ビット・カウンター・モードを選択します。

いずれかの入カソースのインクリメント・エッジの立ち上がりまたは立ち下がり遷移は、OPTION\_REGレジスタのTMR0SEビットによって決定されます。

### 18.1.3 ソフトウェア・プログラマブル・プリスケイラー

ソフトウェアでプログラム可能なプリスケイラーは、タイマー0専用です。プリスケイラーは、OPTION\_REGレジスタのPSAビットをクリアすることで有効になります。

**注：** ウォッチドッグ・タイマー (WDT) は、独自の独立したプリスケイラーを使用します。

タイマー0モジュールには1:2から1:256の範囲の8つのプリスケイラー・オプションがあります。プリスケール値は、OPTION\_REGレジスタのPS<2:0>ビットを介して選択できます。タイマー0モジュールに1:1のプリスケール値を設定するには、OPTION\_REGレジスタのPSAビットをセットしてプリスケイラを無効にする必要があります。

プリスケイラーは読み取りまたは書き込みができません。TMR0レジスタに書き込むすべての命令は、プリスケイラーをクリアします。

### 18.1.4 タイマー0割り込み

タイマー0は、TMR0レジスタがFFhから00hにオーバーフローすると、割り込みを生成します。INTCONレジスタのTMR0IF割り込みフラグ・ビットは、タイマー0割り込みが有効かどうかに関係なく、TMR0レジスタがオーバーフローするたびにセットされます。TMR0IFビットは、ソフトウェアでのみクリアできます。タイマー0割り込み許可は、INTCONレジスタのTMR0IEビットです。

**注釈：** スリープ中にタイマーがフリーズするため、タイマー0割り込みはプロセッサをスリープから復帰させることはできません。

### 18.1.5 8ビット・カウンタ・モードの同期

8ビット・カウンタ・モードの場合、T0CKIピンのインクリメント・エッジを命令クロックに同期させる必要があります。同期は、命令クロックのQ2およびQ4サイクルでプリスケイラー出力をサンプリングすることによって実現できます。外部クロッキング・ソースの高周期と低周期は、セクション29.0「電氣的仕様」に示されているタイミング要件を満たす必要があります。

### 18.1.6 スリープ中の動作

プロセッサがスリープ・モードのとき、タイマー0は動作できません。プロセッサがスリープ・モードになっている間、TMR0レジスタの内容は変更されません。

# PIC16(L)F1508/9

## 18.2 レジスターの定義: オプション・レジスター

### レジスター18-1: OPTION\_REG: オプション・レジスター

R/W-1/1							
WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>		
bit 7							bit 0

<b>記録:</b>		
R =読み取り可能ビット	W =書き込み可能ビット	U =実装されていないビット、「0」として読み取られる
u =ビットは変更されていません	x =ビットは不明です	-n/n =PORでの値およびBOR/他のすべてのリセットでの値
「1」=ビットが設定されている	「0」=ビットがクリアされている	

- bit 7 **WPUEN**: 弱いプルアップ許可ビット  
1 = すべての弱いプルアップが無効になります (MCLRが有効になっている場合を除く)  
0 = 弱いプルアップは、個々のWPUxラッチ値によって有効になります
- bit 6 **INTEDG**: 割り込みエッジ選択ビット  
1 = INTピンの立ち上がりエッジで割り込み  
0 = INTピンの立ち下がりエッジで割り込み
- bit 5 **TMR0CS**: タイマー0クロック・ソース選択ビット  
1 = TOCKIピンの遷移  
0 = 内部命令サイクル・クロック (FOSC/4)
- bit 4 **TMR0SE**: タイマー0ソース・エッジ選択ビット  
1 = TOCKIピンのハイからローへの遷移の増分  
0 = TOCKIピンのローからハイへの遷移の増分
- bit 3 **PSA**: プリスケaler割り当てビット  
1 = プリスケalerがタイマー0モジュールに割り当てられていません  
0 = プリスケalerはタイマー0モジュールに割り当てられています
- bit 2~0 **PS<2:0>**: プリスケaler・レート選択ビット

ビット値	タイマー0レート
000	1 : 2
001	1 : 4
010	1 : 8
011	1 : 16
100	1 : 32
101	1 : 64
110	1 : 128
111	1 : 256

表18-1: タイマー0に関連付けられたレジスターの概要

名前	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ページ上の登録
ADCON2	TRIGSEL<3:0>				—	—	—	—	136
INTCON	GIE	PEIE	TMR0IE	INTE	IOCF	TMR0IF	INTF	IOCF	75
OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			154
TMR0	8ビット・タイマー0カウントの保持レジスター								152*
TRISA	—	—	TRISA5	TRISA4	— <sup>(1)</sup>	TRISA2	TRISA1	TRISA0	109

記録: — = 実装されていない場所。「0」と読みます。網掛けのセルはタイマー0モジュールでは使用されません。

\* ページは登録情報を提供します。

注1: 未実装、「1」と読みます。

## 19.0 ゲート制御付きタイマー1モジュール

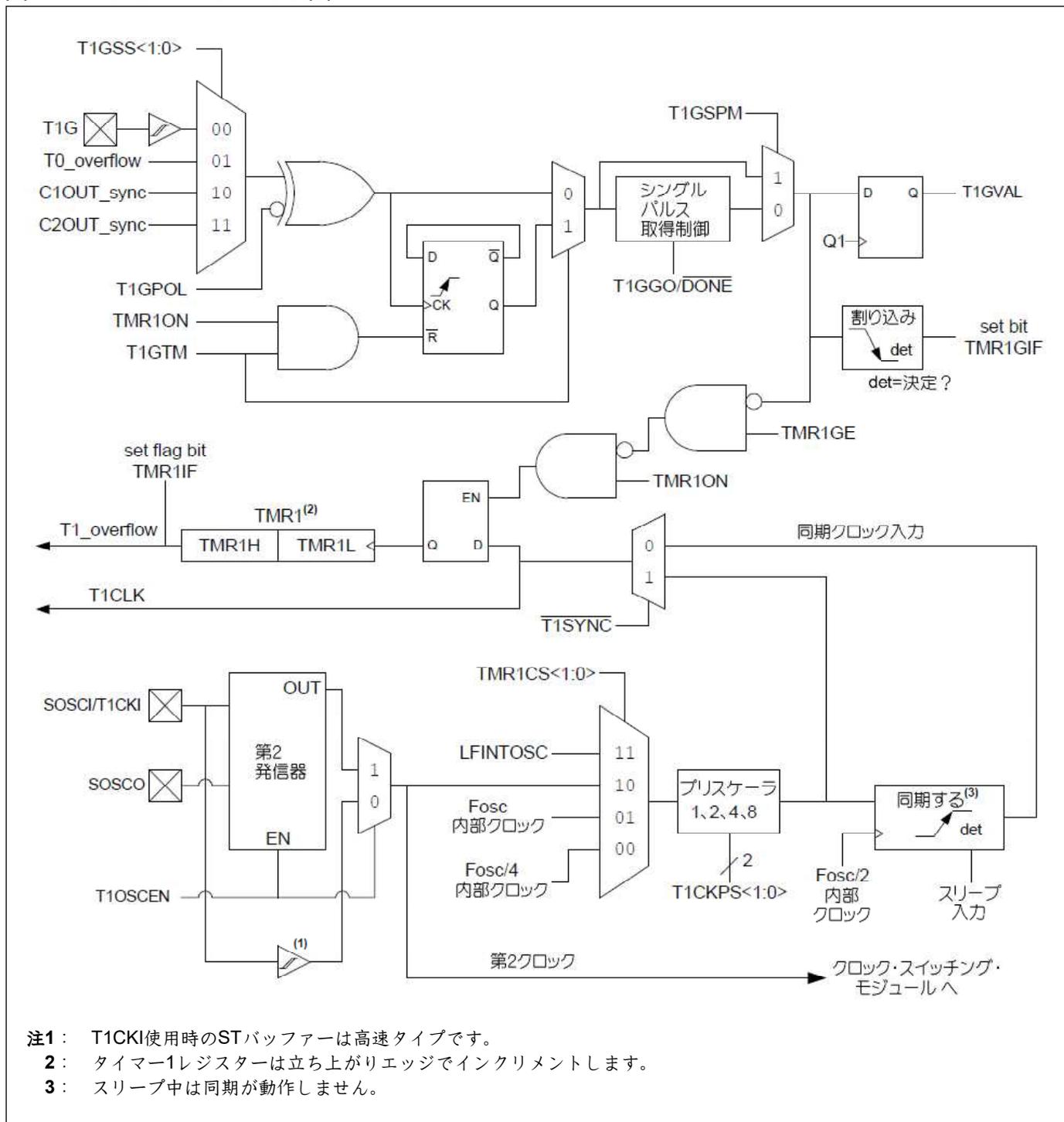
タイマー1モジュールは、次の機能を備えた16ビットのタイマー/カウンターです：

- 16ビット・タイマ/カウンター・レジスタ・ペア (TMR1H:TMR1L)
- プログラム可能な内部または外部クロック・ソース
- 2ビット・プリスケaler
- オプションで同期コンパレータ出力
- 複数のタイマー1ゲート (カウント許可) ソース
- オーバーフロー時に割り込み

- オーバーフロー時のウェイクアップ (外部クロック、非同期モードのみ)
- ADC自動変換トリガ
- 選択可能なゲート・ソース極性
- ゲート・トグル (入れ替え?) ・モード
- ゲート・シングル・パルス・モード
- ゲート値ステータス
- ゲート・イベント割り込み

図19-1に、タイマー1モジュールのブロック図を示します。

図19-1: タイマー1のブロック図



注1: T1CKI使用時のSTバッファは高速タイプです。  
 注2: タイマー1レジスタは立ち上がりエッジでインクリメントします。  
 注3: スリープ中は同期が動作しません。

## 19.1 タイマー1の動作

タイマー1モジュールは、TMR1H:TMR1Lレジスター・ペアを介してアクセスされる16ビットのインクリメント・カウンタです。TMR1HまたはTMR1Lへの書き込みは、カウンタを直接更新します。

内部クロック・ソースとともに使用する場合、モジュールはタイマーであり、命令サイクルごとにインクリメントします。

モジュールを外部クロック・ソースとともに使用すると、タイマーまたはカウンタとして使用でき、外部ソースの選択されたすべてのエッジでインクリメントします。

タイマー1は、T1CONレジスターとT1GCONレジスターのTMR1ONビットとTMR1GEビットをそれぞれ設定することで有効になります。表19-1に、タイマー1の有効化の選択を示します。

表19-1: タイマー1の有効な選択

TMR1ON	TMR1GE	タイマー1の操作
0	0	オフ
0	1	オフ
1	0	常にオン
1	1	カウント可能

## 19.2 クロック・ソース選択

T1CONレジスターのTMR1CS<1:0>ビットとT1OSCENビットは、タイマー1のクロック・ソースを選択するために使用されます。表19-2に、クロック・ソースの選択を示します。

### 19.2.1 内部クロック・ソース

内部クロック・ソースが選択されると、TMR1H:TMR1Lレジスター・ペアは、タイマー1プリスケイラーによって決定されたFOSCの倍数でインクリメントします。

FOSC内部クロック・ソースが選択されている場合、タイマー1レジスター値は命令クロック・サイクルごとに4カウントずつインクリメントします。この状態のため、タイマー1値を読み取るときに分解能に2 LSBエラーが発生します。タイマー1のフル分解能を利用するには、非同期入力信号を使用してタイマー1クロック入力をゲートする必要があります。

次の非同期ソースを使用できます:

- T1Gピンからタイマー1ゲートへの非同期イベント
- タイマー1ゲートへのC1またはC2コンパレータ入力

### 19.2.2 外部クロック・ソース

外部クロック・ソースが選択されている場合、タイマー1モジュールはタイマーまたはカウンタとして機能する場合があります。

カウントを有効にすると、タイマー1は外部クロック入力T1CKIの立ち上がりエッジでインクリメントされます。外部クロック・ソースは、マイクロコントローラのシステム・クロックに同期させることも、非同期で実行することもできます。

**注釈:** カウンタ・モードでは、次の条件のいずれか1つ以上の後、最初の増分立ち上がりエッジの前に立ち下がりエッジをカウンタで登録する必要があります:

- POR後にタイマー1が有効になります
- TMR1HまたはTMR1Lに書き込みます
- タイマー1が無効になっている
- タイマー1が無効になっています (TMR1ON = 0)  
T1CKIがハイの場合、T1CKIがローの場合、タイマー1が有効になります (TMR1ON = 1)。

表19-2: クロック・ソースの選択

TMR1CS<1:0>	T1OSCEN	クロック・ソース
11	x	LFINTOSC
10	1	SOSCI/SOSCOピンの2次発振器回路
	0	T1CKIピンの外部クロッキング
01	x	システム・クロック (FOSC)
00	x	命令クロック (FOSC/4)

## 19.3 タイマー1プリスケaler

タイマー1には、クロック入力の1、2、4、または8分割を可能にする4つのプリスケaler・オプションがあります。T1CONレジスターのT1CKPSビットはプリスケaler・カウンタを制御します。プリスケaler・カウンタは、直接読み取りまたは書き込みができません；ただし、プリスケaler・カウンタは、TMR1HまたはTMR1Lへの書き込み時にクリアされます。

## 19.4 タイマー1(2次)発信器

専用の低電力32.768 kHz発振回路が、ピンS1OSI（入力）とS1OSO（アンプ出力）の間に組み込まれています。この内部回路は、外部の32.768 kHz水晶と組み合わせて使用します。発振回路は、T1CONレジスターのT1OSCENビットをセットすることで有効になります。発信器はスリープ中も動作し続けます。

**注釈：** 発信器は、使用前に起動して安定するまでに時間がかかります。OSCSTATレジスターのSOSCRビットは発信器を監視し、発信器を使用する準備ができたことを示します。T1OSCENがセットされると、SOSCRビットがクリアされます。発振器の1024サイクルが打ち消された後、SOSCRビットが設定され、発振器が安定して使用できる状態になっている必要があることを示します。

## 19.5 非同期カウンタ・モードでのタイマー1の動作

T1CONレジスターの制御ビットT1SYNCがセットされている場合、外部クロック入力は同期されません。タイマーは、内部相クロックとは非同期にインクリメントします。外部クロック・ソースが選択されている場合、タイマーはスリープ中も実行を継続し、オーバーフロー時に割り込みを生成して、プロセッサをウェイクアップさせることができます。ただし、タイマーの読み取り/書き込みには、ソフトウェアで特別な注意が必要です（セクション19.5.1「非同期カウンタ・モードでのタイマー1の読み取りと書き込み」を参照）。

**注釈：** 同期操作から非同期操作に切り替える場合、インクリメントをスキップすることができます。非同期操作から同期操作に切り替えるときに、追加の増分を生成することができます。

### 19.5.1 非同期カウンタ・モードでのタイマー1の読み取りと書き込み

タイマーが外部非同期クロックから実行されているときにTMR1HまたはTMR1Lを読み取ると、有効な読み取りが保証されます（ハードウェアで処理されます）。ただし、16ビットタイマーを2つの8ビット値で読み取ること自体が特定の問題を引き起こすことに注意する必要があります。これは、タイマーが読み取りの間にオーバーフローする可能性があるためです。

書き込みの場合、ユーザーは単にタイマーを停止して目的の値を書き込むことをお勧めします。レジスターがインクリメントしているときにタイマー・レジスターに書き込むと、書き込み競合が発生する可能性があります。これにより、TMR1H:TMR1Lレジスター・ペアに予測できない値が生成される可能性があります。

## 19.6 タイマー1ゲート

タイマー1は、自由にカウントするように構成することも、タイマー1ゲート回路を使用してカウントを有効または無効にすることもできます。これは、タイマー1ゲート有効化とも呼ばれます。タイマー1ゲートは、複数の選択可能なソースによって駆動することもできます。

### 19.6.1 タイマー1ゲート有効化

タイマー1ゲート有効化モードは、T1GCONレジスターのTMR1GEビットをセットすることで有効になります。タイマー1ゲート有効化モードの極性は、T1GCONレジスターのT1GPOLビットを使用して設定されます。

タイマー1ゲート有効化モードが有効になっている場合、タイマー1はタイマー1クロック・ソースの立ち上がりエッジでインクリメントします。タイマー1ゲート有効化モードが無効になっている場合、インクリメントは発生せず、タイマー1は現在のカウントを保持します。タイミングの詳細については、図19-3を参照してください。

表19-3: タイマー1ゲートを有効にする選択

T1CLK	T1GPOL	T1G	タイマー1の操作
↑	0	0	カウント
↑	0	1	カウント保持
↑	1	0	カウント保持
↑	1	1	カウント

### 19.6.2 タイマー1ゲート・ソース選択

表19-4に、タイマー1のゲート・ソースの選択を示します。ソースの選択は、T1GCONレジスターのT1GSS<1:0>ビットによって制御されます。利用可能な各ソースの極性も選択できます。極性の選択は、T1GCONレジスターのT1GPOLビットによって制御されます。

表19-4: タイマー1ゲート・ソース

T1GSS	タイマー1ゲート・ソース
00	タイマー1ゲート・ピン (T1G)
01	タイマー0のオーバーフロー (T0_overflow) (TMR0はFFhから00hに増分します)
10	コンパレーター1出力 (C1OUT_sync) <sup>(1)</sup>
11	コンパレーター2出力 (C2OUT_sync) <sup>(1)</sup>

注1: オプションで同期コンパレータ出力。

## 19.6.2.1 T1Gピン・ゲート操作

T1Gピンは、タイマー1ゲート制御の1つのソースです。これは、タイマー1ゲート回路に外部ソースを供給するために使用できます。

## 19.6.2.2 タイマー0オーバーフロー・ゲート操作

タイマー0がFFhから00hにインクリメントすると、ローからハイへのパルスが自動的に生成され、タイマー1ゲート回路に内部的に供給されます。

## 19.6.3 タイマー1ゲート・トグル・モード

タイマー1ゲート・トグル・モードを有効にすると、シングル・レベル・パルスの持続時間ではなく、タイマー1ゲート信号の全サイクル長を測定できます。

タイマー1ゲート・ソースは、信号のインクリメント・エッジごとに状態を変更するフリップフロップを介してルーティングされます。タイミングの詳細については、図19-4を参照してください。

タイマー1ゲート・トグル・モードは、T1GCONレジスタのT1GTMビットをセットすることで有効になります。T1GTMビットがクリアされると、フリップフロップはクリアされ、クリアされたままになります。これは、測定するエッジを制御するために必要です。

**注意：** ゲート極性の変更と同時にトグル・モードを有効にすると、動作が不確定になる可能性があります。

## 19.6.4 タイマー1ゲート・シングル・パルス・モード

タイマー1ゲート・シングル・パルス・モードが有効になっている場合、シングル・パルス・ゲート・イベントをキャプチャすることができます。タイマー1ゲート・シングル・パルス・モードは、T1GCONレジスタのT1GSPMビットをセットすることによって最初に有効になります。次に、T1GCONレジスタのT1GGO/DONEビットをセットする必要があります。タイマー1は、次の増分エッジで完全に有効になります。パルスの次の立ち下がりエッジで、T1GGO/DONEビットが自動的にクリアされます。T1GGO/DONEビットがソフトウェアで再度設定されるまで、他のゲート・イベントがタイマー1をインクリメントすることは許可されません。タイミングの詳細については、図19-5を参照してください。

T1GCONレジスタのT1GSPMビットをクリアしてシングル・パルス・ゲート・モードを無効にする場合は、T1GGO/DONEビットもクリアする必要があります。

トグル・モードとシングル・パルス・モードを同時に有効にすると、両方のセクションが連携して機能します。これにより、タイマー1ゲート・ソースのサイクル・タイムを測定できます。タイミングの詳細については、図19-6を参照してください。

## 19.6.5 タイマー1ゲート値ステータス

タイマー1ゲート値ステータスを利用すると、ゲート制御値の最新レベルを読み取ることができます。値はT1GCONレジスタのT1GVALビットに格納されます。T1GVALビットは、タイマー1ゲートが有効になっていない（TMR1GEビットがクリアされている）場合でも有効です。

## 19.6.6 タイマー1ゲート・イベント割り込み

タイマー1ゲート・イベント割り込みを有効にすると、ゲート・イベントの完了時に割り込みを生成することができます。T1GVALの立ち下がりエッジが発生すると、PIR1レジスタのTMR1GIFフラグ・ビットがセットされます。PIE1レジスタのTMR1GIEビットがセットされている場合、割り込みが認識されます。

TMR1GIFフラグ・ビットは、タイマー1ゲートが有効になっていない（TMR1GEビットがクリアされている）場合でも動作します。

## 19.7 タイマー1割り込み

タイマー1レジスター・ペア (TMR1H:TMR1L) はFFFFhにインクリメントし、0000hにロールオーバーします。タイマー1がロールオーバーすると、PIR1レジスターのタイマー1割り込みフラグビットがセットされます。ロールオーバー時の割り込みを有効にするには、これらのビットを設定する必要があります:

- T1CONレジスターのTMR1ONビット
- PIE1レジスターのTMR1IEビット
- INTCONレジスターのPEIEビット
- INTCONレジスターのGIEビット

割り込みは、割り込みサービス・ルーチンのTMR1IFビットをクリアすることによってクリアされます。

**注意:** 割り込みを有効にする前に、TMR1H:TMR1Lレジスター・ペアとTMR1IFビットをクリアする必要があります。

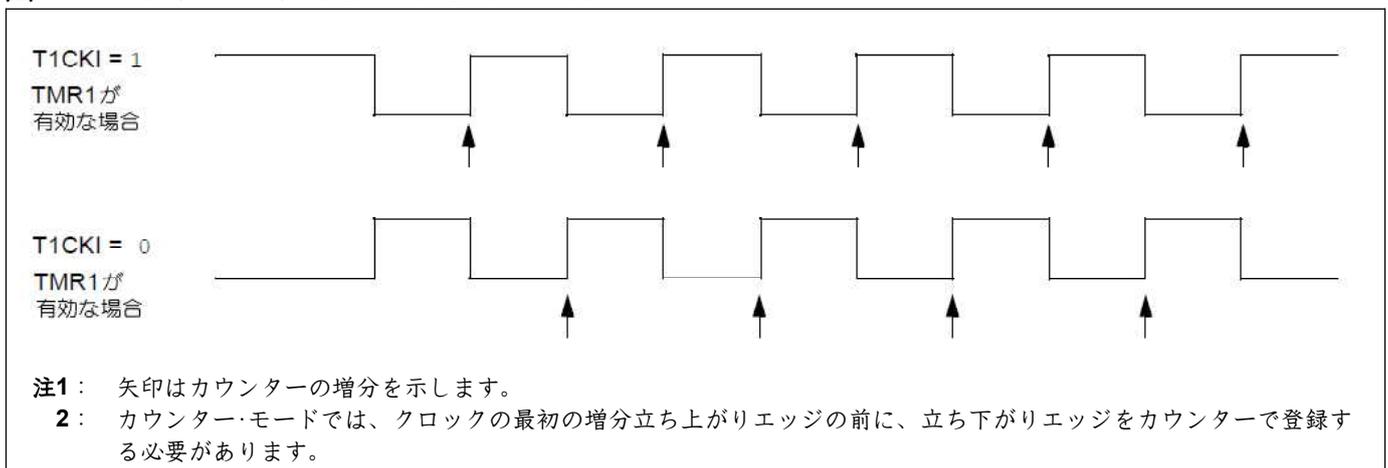
## 19.8 スリープ中のタイマー1の動作

タイマー1は、非同期カウンター・モードでセットアップされている場合、スリープ中にのみ動作できます。このモードでは、外部水晶またはクロック・ソースを使用してカウンターをインクリメントできます。デバイスをウェイクアップするようにタイマーを設定するには:

- T1CONレジスターのTMR1ONビットをセットする必要があります
- PIE1レジスターのTMR1IEビットをセットする必要があります
- INTCONレジスターのPEIEビットを設定する必要があります
- T1CONレジスターの $\overline{T1SYNC}$ ビットをセットする必要があります
- T1CONレジスターのTMR1CSビットを設定する必要があります
- T1CONレジスターのT1OSCENビットを設定する必要があります

デバイスはオーバフロー時にウェイクアップし、次の命令を実行します。INTCONレジスターのGIEビットが設定されている場合、デバイスは割り込みサービス・ルーチンを呼び出します。

図19-2: タイマー1インクリメンティング・エッジ



タイマー1発信器は、 $\overline{T1SYNC}$ ビットの設定に関係なく、スリープ状態で動作し続けます。

### 19.8.1 代替ピンの場所

このモジュールには、代替ピン機能レジスタAPFCONを使用して他の場所に移動できるI/Oピンが組み込まれています。移動できるピンと、リセット時のデフォルトの位置を決定するには、詳細についてセクション11.1「代替ピン機能」を参照してください。

# PIC16(L)F1508/9

図19-3:タイマー1ゲート有効化モード

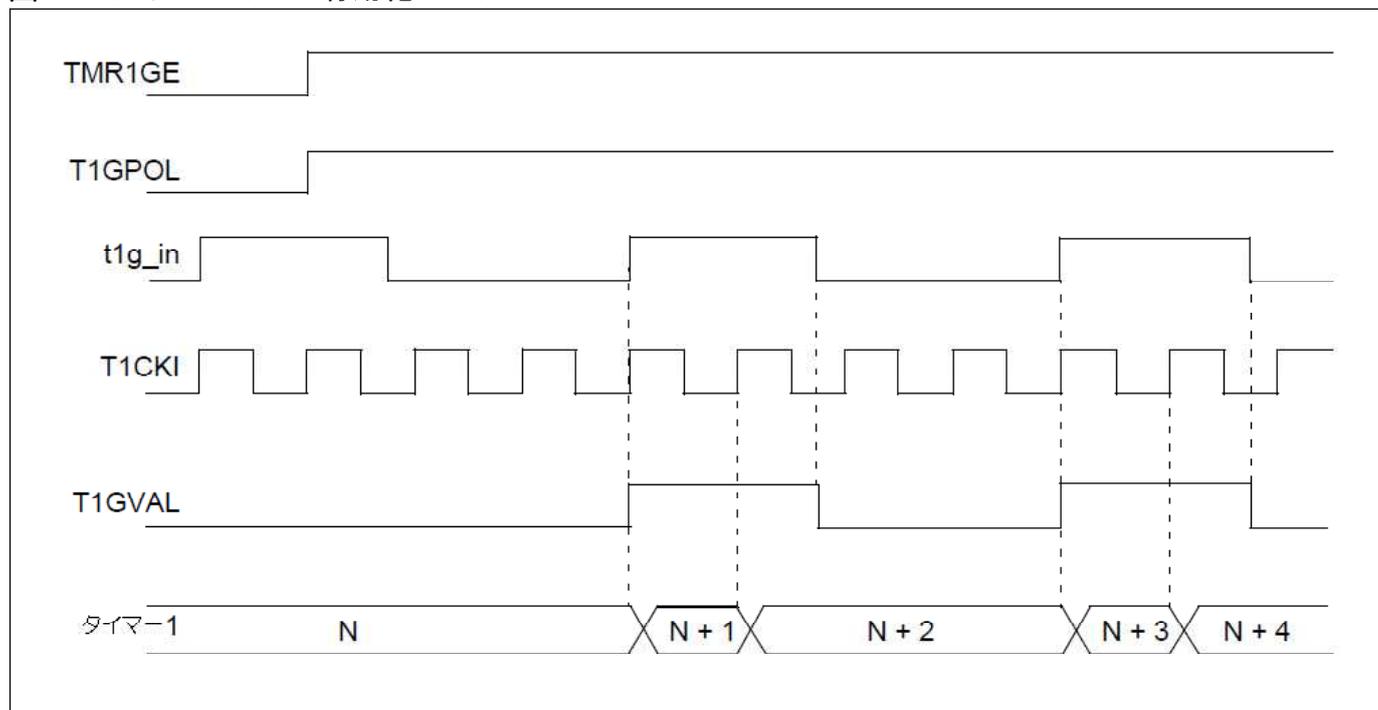


図19-4: タイマー1ゲート・トグル・モード

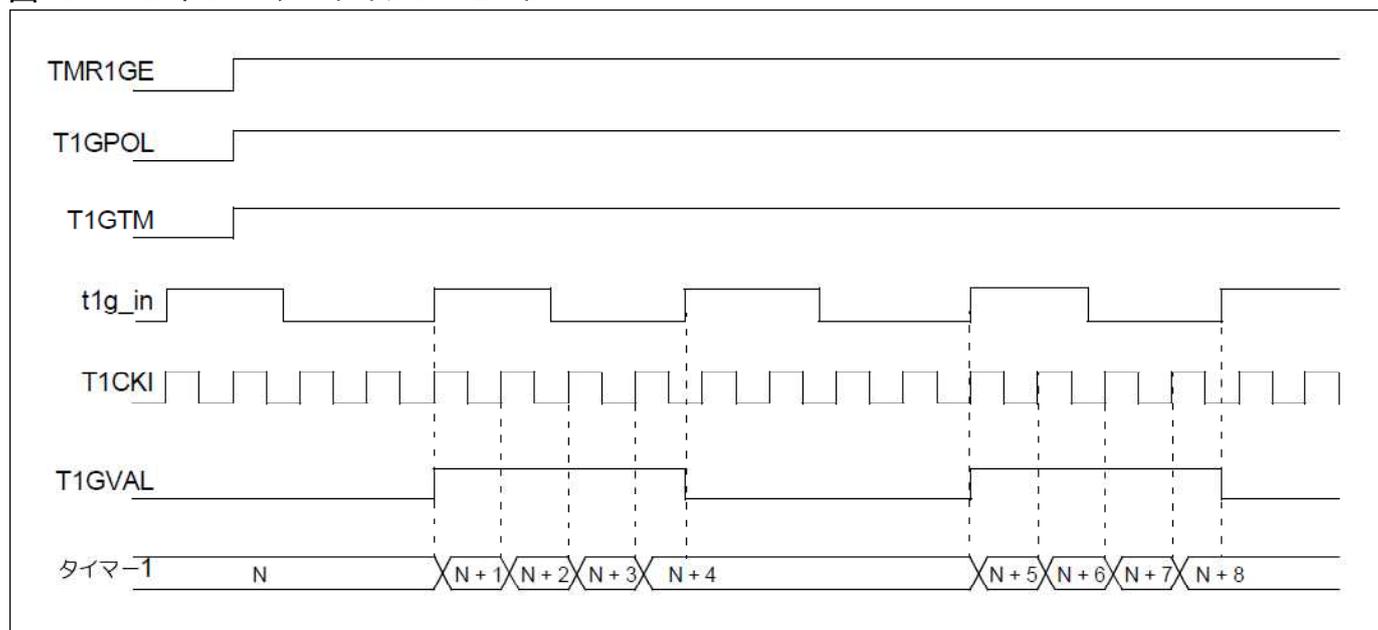
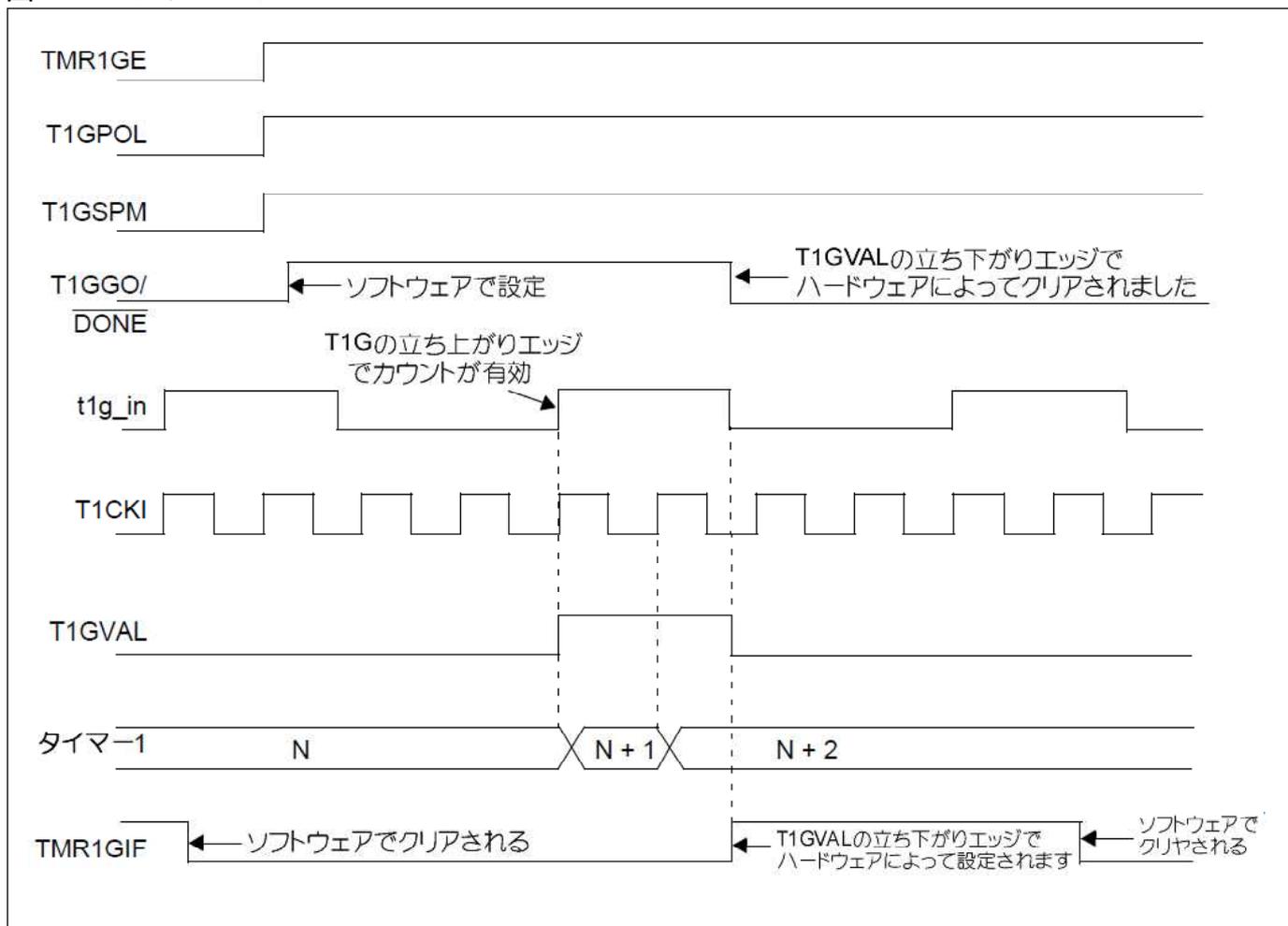
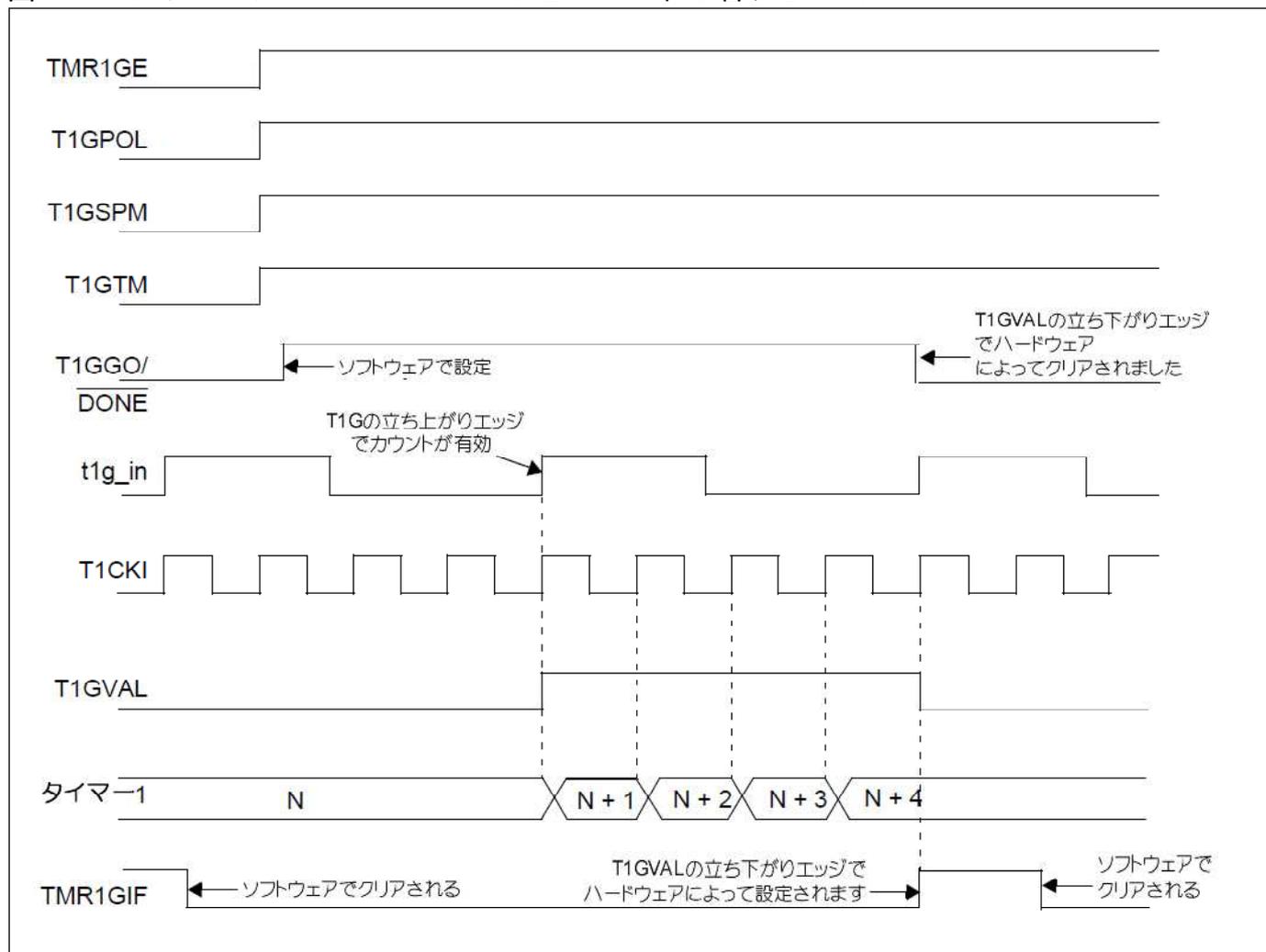


図19-5: タイマー1ゲート・シングル・パルス・モード



# PIC16(L)F1508/9

図19-6: タイマー1ゲート・シングル・パルスとトグルの組み合わせモード



## 19.9 レジスター定義: タイマー1制御

## レジスター19-1: T1CON: タイマー1制御レジスター

R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	U-0	R/W-0/u
TMR1CS<1:0>		T1CKPS<1:0>		T1OSCEN	T1SYNC	—	TMR1ON
bit 7							bit 0

## 記録:

R = 読み取り可能ビット	W = 書き込み可能ビット	U = 実装されていないビット、「0」として読み取られる
u = ビットは変更されていません	x = ビットは不明です	-n/n = PORでの値およびBOR/他のすべてのリセットでの値
[1]=ビットが設定されている	[0]=ビットがクリアされている	

- bit 7~6 **TMR1CS<1:0>**: タイマー1クロック・ソース選択ビット  
 11 = タイマー1のクロック・ソースはLFINTOSCです  
 10 = タイマー1のクロック・ソースはピンまたは発信器です:  
   T1OSCEN = 0の場合:  
     T1CKIピンからの外部クロック (立ち上がりエッジ)  
   T1OSCEN = 1の場合:  
     SOSCI/SOSCOピンの水晶発振器  
 01 = タイマー1クロック・ソースはシステム・クロック (Fosc) です  
 00 = タイマー1クロック・ソースは命令クロック (Fosc/4) です
- bit 5~4 **T1CKPS<1:0>**: タイマー1入力クロック・プリスケール選択ビット  
 11 = 1:8 プリスケール値  
 10 = 1:4 プリスケール値  
 01 = 1:2 プリスケール値  
 00 = 1:1 プリスケール値
- bit 3 **T1OSCEN**: LPオシレータ許可制御ビット  
 1 = 2次発振回路はタイマー1で有効になっている  
 0 = 2次発振回路はタイマー1で無効になっている
- bit 2 **T1SYNC**: タイマー1同期制御ビット  
 1 = 非同期クロック入力を同期しないでください  
 0 = 非同期クロック入力をシステムクロック (Fosc) と同期します
- bit 1 **未実装**: 「0」として読み取ります
- bit 0 **TMR1ON**: タイマー1オン・ビット  
 1 = タイマー1を有効にします  
 0 = タイマー1を停止し、タイマー1ゲート・フリップフロップをクリアします

# PIC16(L)F1508/9

## レジスタ19-2: T1GCON: タイマー1ゲート制御レジスタ

R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W/HC-0/u	R-x/x	R/W-0/u	R/W-0/u
TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/ DONE	T1GVAL	T1GSS<1:0>	
bit 7						bit 0	

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている      HC =ビットはハードウェアによってクリアされます

- bit 7      **TMR1GE**: タイマー1ゲート許可ビット  
**TMR1ON = 0** の場合:  
このビットは無視されます  
**TMR1ON = 1** の場合:  
1 = タイマー1のカウンタはタイマー1ゲート機能によって制御されます  
0 = タイマー1は、タイマー1ゲート機能に関係なくカウンタされます
- bit 6      **T1GPOL**: タイマー1ゲート極性ビット  
1 = タイマー1ゲートはアクティブ・ハイです (タイマー1はゲートがハイのときにカウンタされます)  
0 = タイマー1ゲートはアクティブ・ローです (タイマー1はゲートがローのときにカウンタされます)
- bit 5      **T1GTM**: タイマー1ゲート・トグル・モード・ビット  
1 = タイマー1ゲート・トグル・モードが有効になっている  
0 = タイマー1ゲート・トグル・モードが無効になり、トグル・フリップフロップがクリアされます  
タイマー1ゲート・フリップフロップは、すべての立ち上がりエッジで切り替わります。
- bit 4      **T1GSPM**: タイマー1ゲート・シングル・パルス・モード・ビット  
1 = タイマー1ゲート・シングル・パルス・モードが有効になっており、タイマー1ゲートを制御しています  
0 = タイマー1ゲート・シングル・パルス・モードが無効になっている
- bit 3      **T1GGO/DONE**: タイマー1ゲート・シングル・パルス取得ステータス・ビット  
1 = タイマー1ゲートのシングル・パルス取得の準備が整い、エッジを待機しています  
0 = タイマー1ゲートのシングル・パルス取得が完了したか、開始されていません
- bit 2      **T1GVAL**: タイマー1ゲート値ステータス・ビット  
TMR1H:TMR1Lに提供できるタイマー1ゲートの現在の状態を示します。  
タイマー1ゲート許可 (TMR1GE) の影響を受けません。
- bit 1~0      **T1GSS<1:0>**: タイマー1ゲート・ソース選択ビット  
11 = コンパレータ2オプションで同期された出力 (C2OUT\_sync)  
10 = コンパレータ1オプションで同期された出力 (C1OUT\_sync)  
01 = タイマー0オーバーフロー出力 (T0\_overflow)  
00 = タイマー1ゲートピン (T1G)

表19-5: タイマー1に関連付けられたレジスタの概要

名前	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ページ上の登録
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	110
APFCON	—	—	—	SSSEL	T1GSEL	—	CLC1SEL	NCO1SEL	107
INTCON	GIE	PEIE	TMR0IE	INTE	IOIE	TMR0IF	INTF	IOIF	75
OSCSTAT	SOSCR	—	OSTS	HFIOFR	—	—	LFIOFR	HFIOFS	60
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	—	TMR2IE	TMR1IE	76
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	—	TMR2IF	TMR1IF	79
TMR1H	16ビットTMR1カウントの最上位バイトの保持レジスタ								159*
TMR1L	16ビットTMR1カウントの最下位バイトの保持レジスタ								159*
TRISA	—	—	TRISA5	TRISA4	— <sup>(1)</sup>	TRISA2	TRISA1	TRISA0	109
T1CON	TMR1CS<1:0>		T1CKPS<1:0>		T1OSCEN	T1SYNC	—	TMR1ON	163
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/ DONE	T1GVAL	T1GSS<1:0>		164

記録： — =実装されていない場所、「0」と読みます。網掛けのセルはタイマー1モジュールでは使用されません。

\* ページは登録情報を提供します。

注1： 未実装、「1」と読みます。

## 20.0 タイマー2モジュール

タイマー2モジュールには次の機能が組み込まれています：

- 8ビットのタイマーおよび周期レジスター（それぞれTMR2およびPR2）
- 読み取り可能および書き込み可能（両方のレジスター）
- ソフトウェアでプログラム可能なプリスケイラー（1:1、1:4、1:16、および1:64）
- ソフトウェアでプログラム可能なポストスケイラー（1:1から1:16）
- PR2とのTMR2一致の割り込み

タイマー2のブロック図については、図20-1を参照してください。

図20-1: タイマー2のブロック図

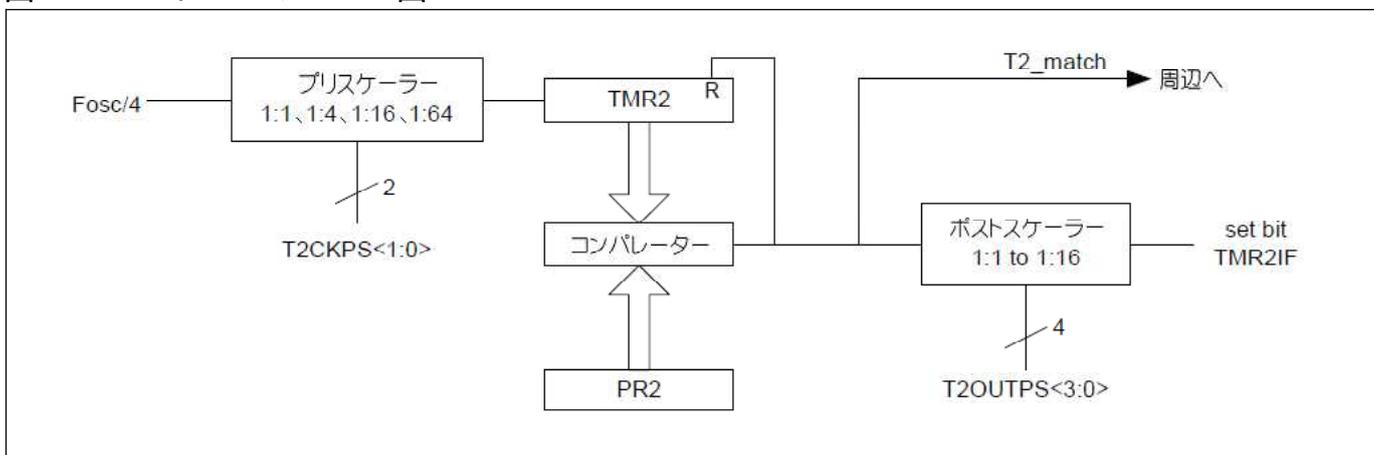
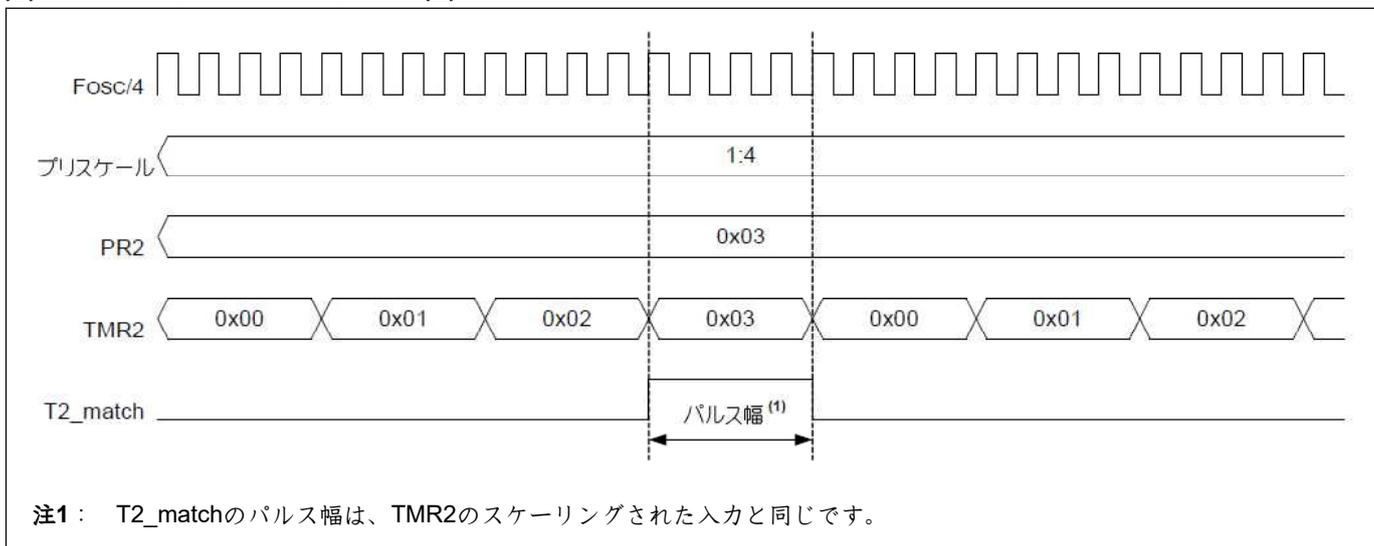


図20-2: タイマー2のタイミング図



注1：  $T2\_match$ のパルス幅は、TMR2のスケールされた入力と同じです。

## 20.1 タイマー2の動作

タイマー2モジュールへのクロック入力にはシステム命令クロック (Fosc/4) です。

TMR2は、各クロック・エッジで00hから増分します。

クロック入力の4ビット・カウンタ/プリスケalerにより、直接入力、4分周および16分周のプリスケaler・オプションが可能です。これらのオプションは、T2CONレジスタのプリスケaler制御ビットT2CKPS<1:0>によって選択されます。TMR2の値は、各クロック・サイクルで周期レジスタPR2の値と比較されます。2つの値が一致すると、コンパレータはタイマー出力として一致信号を生成します。この信号はまた、次のサイクルでTMR2の値を00hにリセットし、出力カウンタ/ポストスケalerを駆動します (セクション20.2「タイマー2割り込み」を参照)。

TMR2およびPR2レジスタは、どちらも直接読み取りおよび書き込みが可能です。TMR2レジスタはデバイスのリセット時にクリアされますが、PR2レジスタはFFhに初期化されます。プリスケalerとポストスケalerの両方のカウンタは、次のイベントでクリアされます：

- TMR2レジスタへの書き込み
- T2CONレジスタへの書き込み
- パワーオン・リセット (POR)
- ブラウンアウト・リセット (BOR)
- MCLRリセット
- ウォッチドッグ・タイマー (WDT) リセット
- スタック・オーバーフローのリセット
- スタック・アンダーフロー・リセット
- **RESET** 命令

**注意：** T2CONが書き込まれるとき、TMR2はクリアされません。

## 20.2 タイマー2割り込み

タイマー2は、オプションのデバイス割り込みを生成することもできます。タイマー2出力信号 (T2\_match) は、4ビットカウンタ/ポストスケalerの入力を提供します。このカウンタは、PIR1レジスタのTMR2IFにラッチされるTMR2一致割り込みフラグを生成します。割り込みは、PIE1レジスタのTMR2マッチ割り込みイネーブルビットTMR2IEをセットすることで有効になります。

T2CONレジスタのポストスケaler制御ビットT2OUTPS<3:0>を使用して、16のポストスケaler・オプション (1:1から1:16まで) の範囲を選択できます。

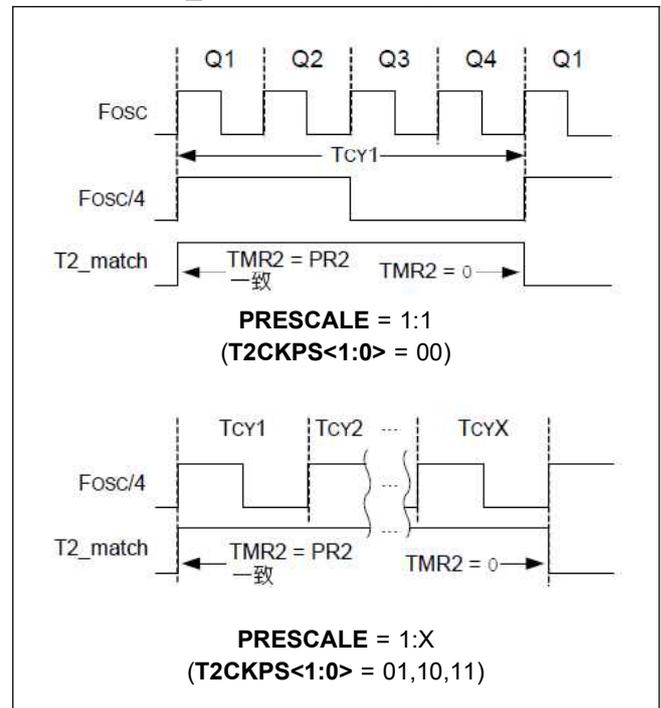
## 20.3 タイマー2出力

TMR2の出力はT2\_matchです。T2\_matchは、次の周辺機器で使用できます：

- 構成可能なロジック・セル (CLC)
- マスター同期シリアル・ポート (MSSP)
- 数値制御発振器 (NCO)
- パルス幅変調器 (PWM)

T2\_match信号はシステム・クロックと同期しています。図20-3は、Foscおよびプリスケaler値T2CKPS<1:0>に対するT2\_match信号のタイミングの2つの例を示しています。上の図は1:1のプリスケalerタイミングを示し、下の図は1:Xのプリスケaler・タイミングを示しています。

図20-3: T2\_matchタイミング図



## 20.4 スリープ中のタイマー2の動作

プロセッサがスリープ・モードのときは、タイマー2を操作できません。プロセッサがスリープ・モードになっている間、TMR2およびPR2レジスタの内容は変更されません。

# PIC16(L)F1508/9

## 20.5 レジスター定義: タイマー2制御

### レジスター20-1: T2CON: タイマー2制御レジスター

U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	T2OUTPS<3:0>				TMR2ON	T2CKPS<1:0>	
bit 7							bit 0

記録:		
R =読み取り可能ビット	W =書き込み可能ビット	U =実装されていないビット、「0」として読み取られる
u =ビットは変更されていません	x =ビットは不明です	-n/n =PORでの値およびBOR/他のすべてのリセットでの値
[1]=ビットが設定されている	[0]=ビットがクリアされている	

- bit 7 未実装: 「0」として読み取ります
- bit 6~3 **T2OUTPS<3:0>**: タイマー2出力ポストスケーラー選択ビット
  - 0000 = 1:1 ポストスケーラー
  - 0001 = 1:2 ポストスケーラー
  - 0010 = 1:3 ポストスケーラー
  - 0011 = 1:4 ポストスケーラー
  - 0100 = 1:5 ポストスケーラー
  - 0101 = 1:6 ポストスケーラー
  - 0110 = 1:7 ポストスケーラー
  - 0111 = 1:8 ポストスケーラー
  - 1000 = 1:9 ポストスケーラー
  - 1001 = 1:10 ポストスケーラー
  - 1010 = 1:11 ポストスケーラー
  - 1011 = 1:12 ポストスケーラー
  - 1100 = 1:13 ポストスケーラー
  - 1101 = 1:14 ポストスケーラー
  - 1110 = 1:15 ポストスケーラー
  - 1111 = 1:16 ポストスケーラー
- bit 2 **TMR2ON**: タイマー2オン・ビット
  - 1 = タイマー2がオンになっています
  - 0 = タイマー2はオフです
- bit 1~0 **T2CKPS<1:0>**: タイマー2クロック・プリスケール選択ビット
  - 00 = プリスケラーは 1
  - 01 = プリスケラーは 4
  - 10 = プリスケラーは 16
  - 11 = プリスケラーは 64

表20-1: タイマー2に関連付けられたレジスターの概要

名前	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ページ上の登録
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	75
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	—	TMR2IE	TMR1IE	76
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	—	TMR2IF	TMR1IF	76
PR2	タイマー2モジュール期間レジスター								166*
T2CON	—	T2OUTPS<3:0>				TMR2ON	T2CKPS<1:0>		168
TMR2	8ビットTMR2カウンタの保持レジスター								166*

記録: — =実装されていない場所、「0」と読みます。網掛けのセルはタイマー2モジュールには使用されません。

\* ページは登録情報を提供します。



# PIC16(L)F1508/9

I<sup>2</sup>Cインターフェースは、以下のモードと機能をサポートします：

- マスター・モード
- スレーブ・モード
- バイトNACKing (スレーブ・モード)
- 限定的なマルチ・マスター・サポート
- 7ビットおよび10ビットのアドレス指定
- 割り込みの開始と停止
- 割り込みマスクング
- クロックの延長
- バス衝突検出
- 一般的な呼び出しアドレスの照合
- アドレス一致
- アドレス保持モードとデータ保持モード
- 選択可能なSDAx保持ドタイム

図21-2は、マスター・モードのI<sup>2</sup>Cインターフェイス・モジュールのブロック図です。図21-3は、スレーブ・モードのI<sup>2</sup>Cインターフェイス・モジュールの図です。

- 注1：** 複数のMSSPモジュールを備えたデバイスでは、SSPxCONxレジスター名に細心の注意を払うことが非常に重要です。SSPxCON1およびSSPxCON2レジスターは、同じモジュールの異なる動作面を制御し、SSPxCON1およびSSPxCON2は、2つの異なるモジュールの同じ機能を制御します。
- 2：** このセクション全体を通して、任意の動作モードでのMSSPxモジュールへの一般的な参照は、MSSPxまたはMSSP2に等しく適用可能であると解釈される場合があります。レジスター名、モジュールI/O信号、およびビット名は、必要に応じて特定のモジュールを区別するために数字を使用することを示すために、一般的な指定子「x」を使用できます。

図21-2: MSSPxブロック図 (I<sup>2</sup>C™ マスター・モード)

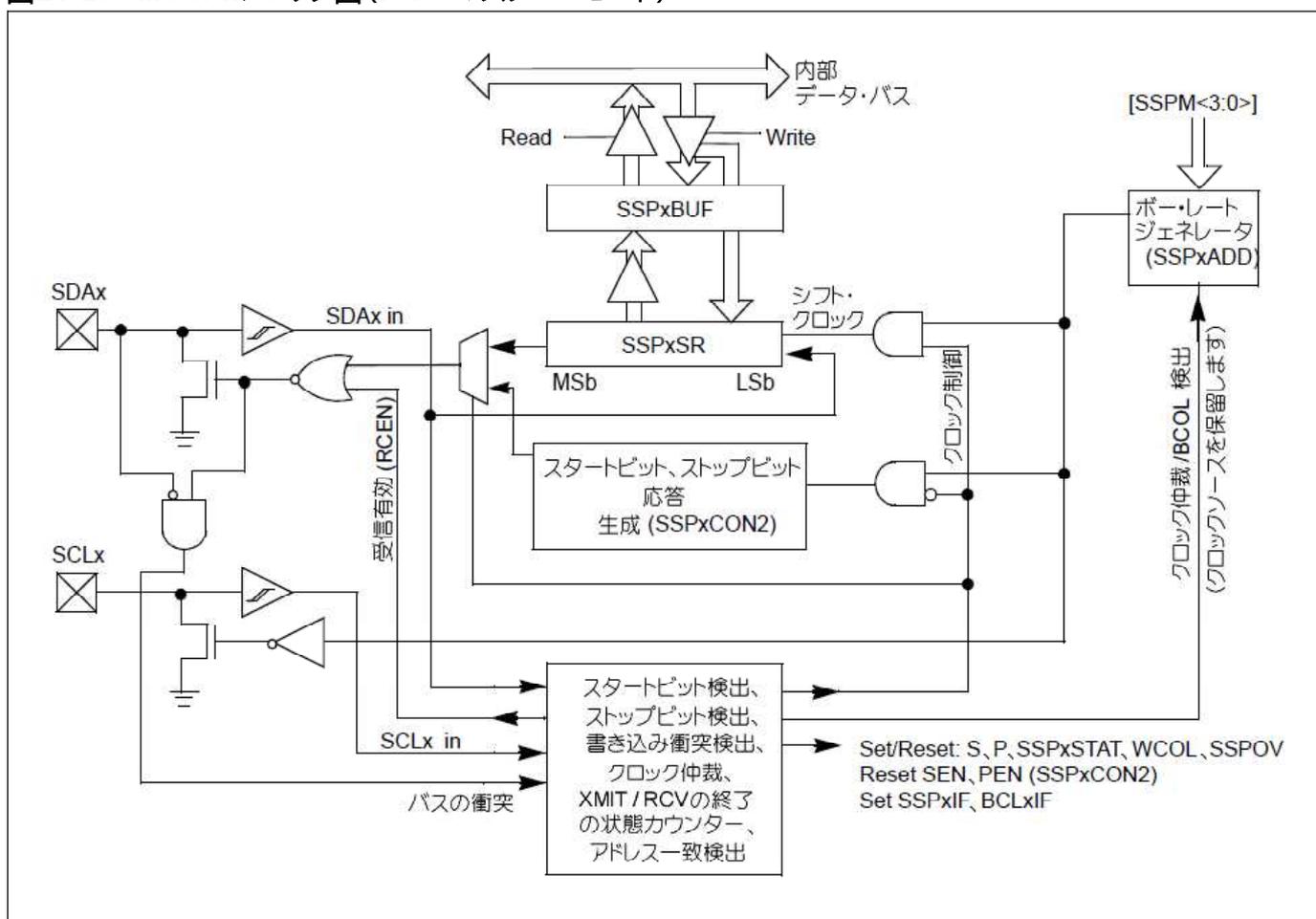
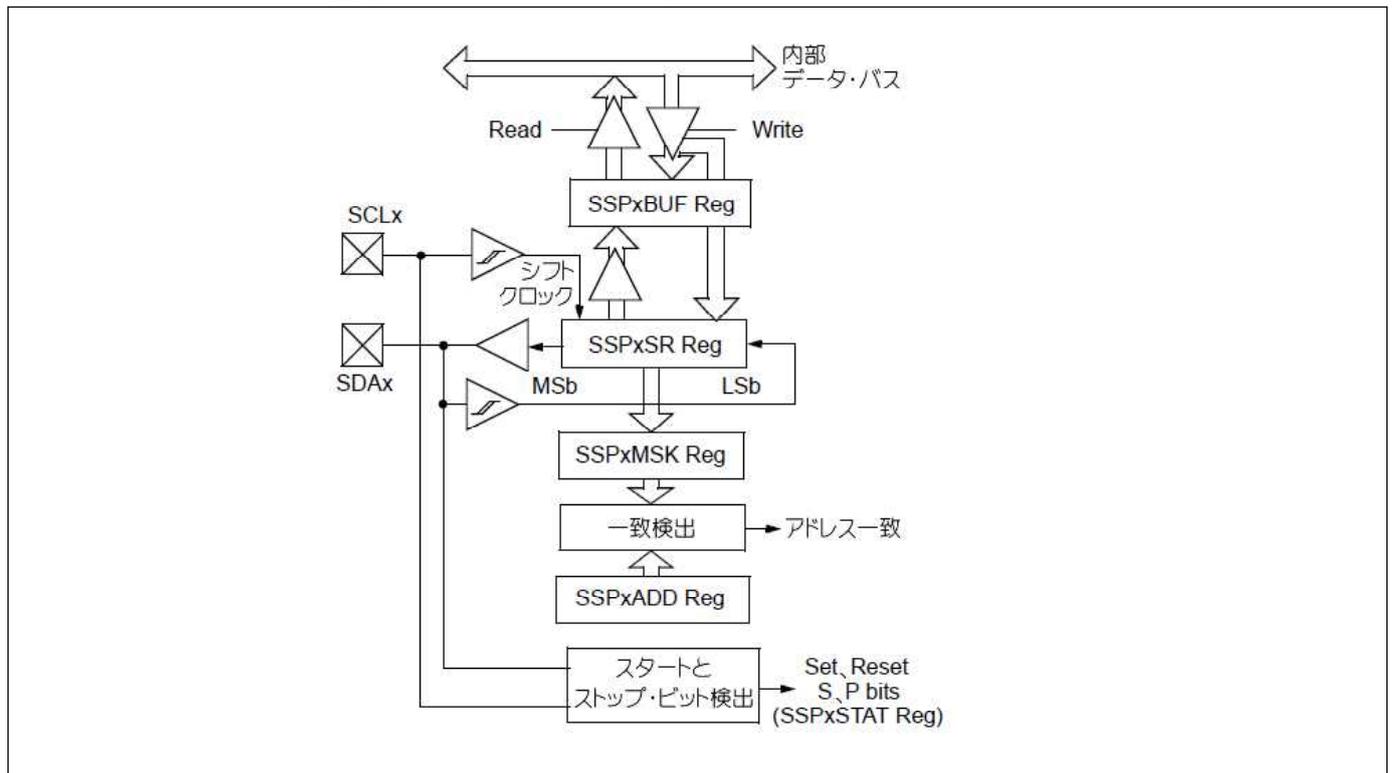


図21-3: MSSPブロック図 (I<sup>2</sup>C™スレーブ・モード)



## 21.2 SPIモードの概要

シリアル・ペリフェラル（周辺）インターフェイス（SPI）バスは、全二重モードで動作する同期シリアル・データ通信バスです。デバイスは、マスター・デバイスが通信を開始するマスター/スレーブ環境で通信します。スレーブ・デバイスは、スレーブ選択と呼ばれるチップ選択を介して制御されます。

SPIバスは4つの信号接続を指定します：

- シリアル・クロック（SCKx）
- シリアル・データ出力（SDOx）
- シリアル・データ入力（SDIx）
- スレーブ・セレクト（選択）（SSx）

図21-1に、SPIモードで動作しているときのMSSPモジュールのブロック図を示します。

SPIバスは、単一のマスター・デバイスと1つ以上のスレーブ・デバイスで動作します。複数のスレーブ・デバイスを使用する場合は、マスター・デバイスから各スレーブ・デバイスへの独立したスレーブ選択接続が必要です。

図21-4に、マスター・デバイスと複数のスレーブ・デバイス間の一般的な接続を示します。

マスターは一度に1つのスレーブのみを選択します。ほとんどのスレーブ・デバイスにはトライステート出力があるため、選択されていない場合、出力信号はバスから切断されているように見えます。

送信には、マスターに1つ、スレーブに1つ、サイズが8ビットの2つのシフト・レジスタが含まれます。マスター・デバイスまたはスレーブ・デバイスのいずれかを使用すると、データは常に一度に1ビットずつシフト出力され、最上位ビット（MSb）が最初にシフト出力されます。同時に、新しい最下位ビット（LSb）が同じレジスタにシフトされます。

図21-5は、マスター・デバイスとスレーブ・デバイスとして構成された2つのプロセッサ間の一般的な接続を示しています。

データは、プログラムされたクロック・エッジで両方のシフト・レジスタからシフト出力され、クロックの反対側のエッジでラッチされます。

マスター・デバイスは、スレーブのSDIx入力ピンに接続されて受信されるSDOx出力ピンで情報を送信します。スレーブ・デバイスは、マスターのSDIx入力ピンに接続されて受信されるSDOx出力ピンで情報を送信します。

通信を開始するために、マスター・デバイスは最初にクロック信号を送信します。マスター・デバイスとスレーブ・デバイスの両方が同じクロック極性に設定されている必要があります。

マスター・デバイスは、シフト・レジスタからMSbを送信することによって送信を開始します。スレーブ・デバイスは同じラインからこのビットを読み取り、シフト・レジスタのLSb位置に保存します。

各SPIクロック・サイクル中に、全二重データ送信が発生します。これは、マスター・デバイスが（SDOxピン上の）シフト・レジスタからMSbを送信し、スレーブ・デバイスがこのビットを読み取ってシフト・レジスタのLSbとして保存している間、スレーブ・デバイスも送信していることを意味します。シフト・レジスタ（SDOxピン上）からのMSbとマスター・デバイスはこのビットを読み取り、シフト・レジスタのLSbとして保存しています。

8ビットがシフト出力された後、マスターとスレーブはレジスタ値を交換されます。

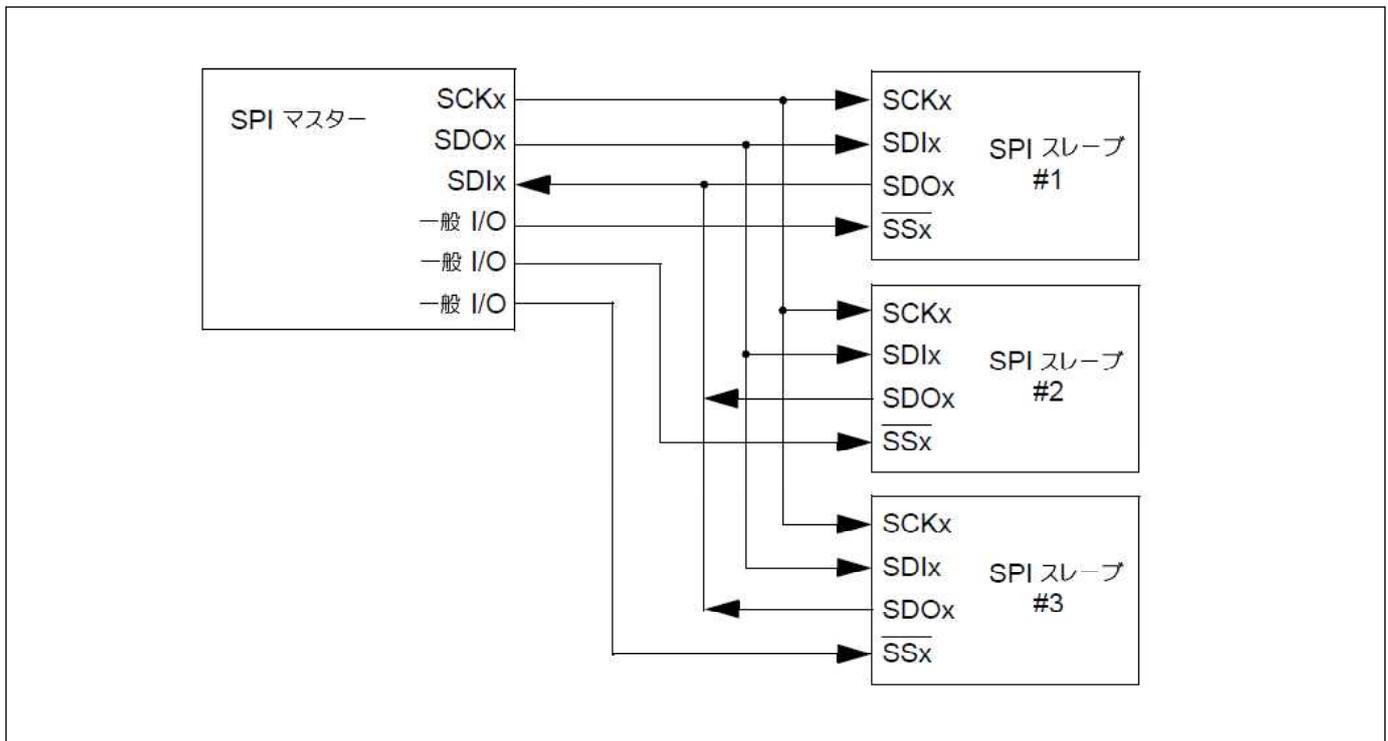
交換するデータがさらにある場合、シフト・レジスタに新しいデータがロードされ、処理が繰り返されます。

データに意味があるかどうか（ダミー・データ）は、アプリケーション・ソフトウェアによって異なります。これは、データ送信の3つのシナリオにつながります：

- マスターは有用なデータを送信し、スレーブはダミー・データを送信します。
- マスターは有用なデータを送信し、スレーブは有用なデータを送信します。
- マスターはダミー・データを送信し、スレーブは有用なデータを送信します。

送信には、任意の数のクロック・サイクルが含まれる場合があります。送信するデータがなくなると、マスターはクロック信号の送信を停止し、スレーブの選択を解除します。スレーブ選択ラインを介して選択されていないバスに接続されたすべてのスレーブ・デバイスは、クロックおよび送信信号を無視する必要があり、独自のデータを送信してはなりません。

図21-4: SPIマスターと複数のスレーブ接続



### 21.2.1 SPIモード・レジスター

MSSPモジュールには、SPIモード動作の5つのレジスターがあります。これらは：

- MSSPステータス・レジスター (SSPxSTAT)
- MSSP制御レジスター1 (SSPxCON1)
- MSSP制御レジスター3 (SSPxCON3)
- MSSPデータ・バッファ・レジスター (SSPxBUF)
- MSSPアドレス・レジスター (SSPxADD)
- MSSPシフト・レジスター (SSPxSR)  
(直接アクセスできない)

SSPxCON1およびSSPxSTATは、SPIモード動作の制御レジスターおよびステータス・レジスターです。SSPxCON1レジスターは読み取りと書き込みが可能です。SSPxSTATの下位6ビットは読み取り専用です。SSPxSTATの上位2ビットは読み取り/書き込みです。

SPIマスター・モードでは、SSPxADDにポーレート・ジェネレーター（生成器）で使用される値をロードできます。ポーレート生成器の詳細については、セクション21.7「ポーレート生成器」を参照してください。

SSPxSRは、データをシフト・インおよびシフト・アウトするために使用されるシフト・レジスタです。SSPxBUFは、SSPxSRレジスターへの間接アクセスを提供します。SSPxBUFは、データ・バイトが書き込まれ、そこからデータ・バイトが読み取られるバッファ・レジスターです。

受信操作では、SSPxSRとSSPxBUFと一緒にバッファリングされた受信器を作成します。SSPxSRが完全なバイトを受信すると、それはSSPxBUFに転送され、SSPxIF割り込みが設定されます。

送信中、SSPxBUFはバッファリングされません。SSPxBUFへの書き込みは、SSPxBUFとSSPxSRの両方に書き込みます。

## 21.2.2 SPIモード動作

SPIを初期化するとき、いくつかのオプションを指定する必要があります。これは、適切な制御ビット (SSPxCON1<5:0> およびSSPxSTAT<7:6>) をプログラミングすることによって行われます。これらの制御ビットにより、以下を指定できます：

- マスター・モード (SCKxはクロック出力です)
- スレーブ・モード (SCKxはクロック入力です)
- クロック極性 (SCKxのアイドル状態)
- データ入力サンプル・フェーズ (データ出力時間の間または終了)
- クロック・エッジ (SCKxの立ち上がり/立ち下がりエッジの出力データ)
- クロック・レート (マスター・モードのみ)
- スレーブ選択モード (スレーブ・モードのみ)

シリアル・ポートを有効にするには、SSPxCON1レジスターのSSP許可ビットSSPENを設定する必要があります。SPIモードをリセットまたは再構成するには、SSPENビットをクリアし、SSPxCONxレジスターを再初期化してから、SSPENビットをセットします。これにより、SDI、SDO、SCK、およびSSピンがシリアル・ポート・ピンとして構成されます。ピンがシリアル・ポート機能として動作するためには、一部のピンのデータ方向ビット (TRISレジスタ内) を次のように適切にプログラムする必要があります：

- SDIxには対応するTRISビットが設定されている必要があります
- SDOxでは、対応するTRISビットをクリアする必要があります
- SCKx (マスター・モード) では、対応するTRISビットをクリアする必要があります
- SCKx (スレーブ・モード) には、対応するTRISビットが設定されている必要があります
- SSxには対応するTRISビットが設定されている必要があります

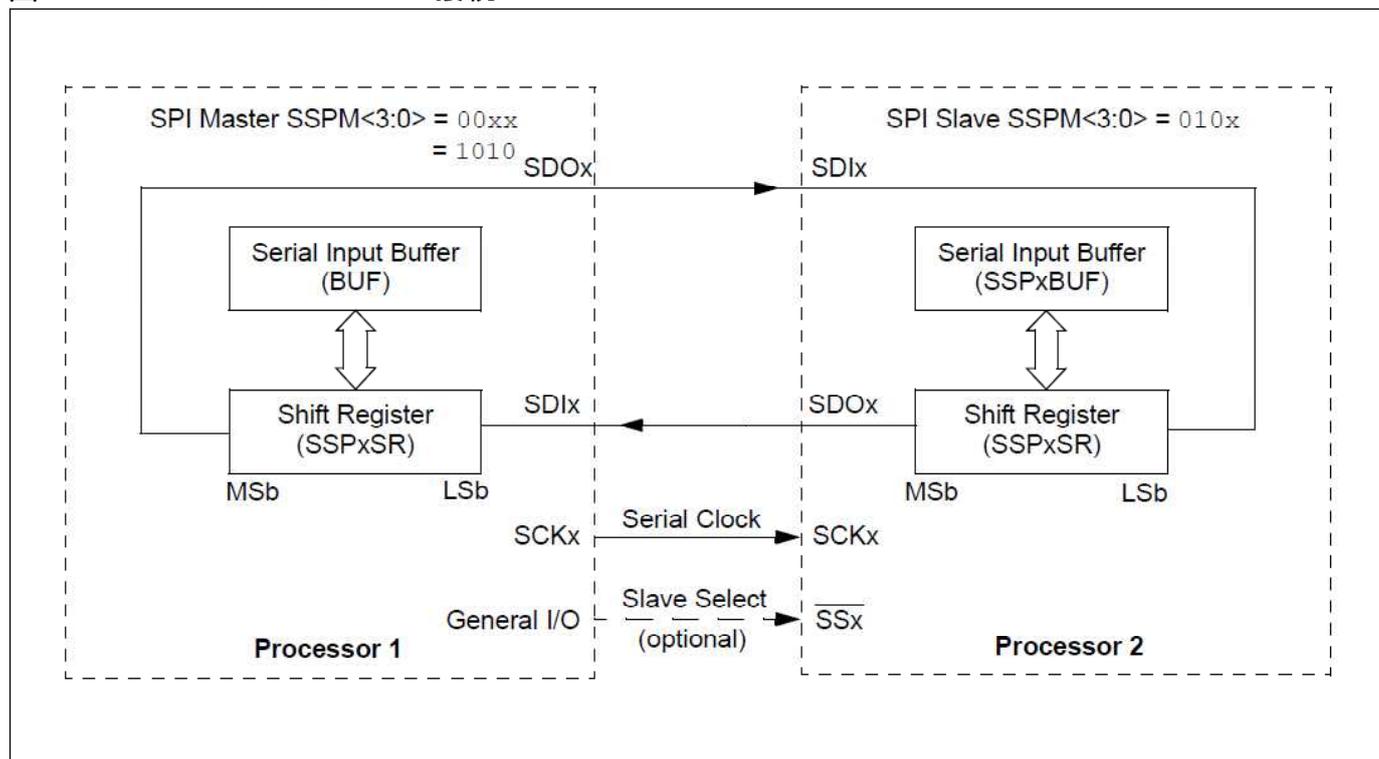
対応するデータ方向 (TRIS) レジスターを反対の値にプログラムすることにより、不要なシリアル・ポート機能を上書きできます。

MSSPは、送信/受信シフト・レジスター (SSPxSR) とバッファ・レジスター (SSPxBUF) で構成されています。SSPxSRは、MSbを最初に、データをデバイスにシフト・インおよびデバイスからシフト・アウトします。SSPxBUFは、受信したデータの準備ができるまで、SSPxSRに書き込まれたデータを保持します。8ビットのデータが受信されると、そのバイトはSSPxBUFレジスターに移動されます。次に、SSPxSTATレジスターのバッファ・フル検出ビットBFと割り込みフラグビットSSPxIFがセットされます。受信データのこのダブル・バッファリング (SSPxBUF) により、受信したばかりのデータを読み取る前に、次のバイトが受信を開始できます。データの送受信中のSSPxBUFレジスターへの書き込みはすべて無視され、書き込み衝突検出ビットであるSSPxCON1レジスターのWCOLが設定されます。ユーザー・ソフトウェアは、SSPxBUFレジスターへの次の書き込みが正常に完了するように、WCOLビットをクリアする必要があります。

アプリケーション・ソフトウェアが有効なデータを受信することを期待している場合、転送するデータの次のバイトがSSPxBUFに書き込まれる前に、SSPxBUFを読み取る必要があります。SSPxSTATレジスターのバッファ・フル・ビットBFは、SSPxBUFに受信データがロードされたとき (送信が完了したとき) を示します。SSPxBUFが読み取られると、BFビットがクリアされます。SPIが送信機の場合、このデータは無関係である可能性があります。通常、MSSP割り込みは、送信/受信がいつ完了したかを判別するために使用されます。割り込み方式を使用しない場合は、ソフトウェア・ポーリングを実行して、書き込みの衝突が発生しないようにすることができます。

SSPxSRは直接読み取りまたは書き込み可能ではなく、SSPxBUFレジスターをアドレス指定することによってのみアクセスできます。さらに、SSPxSTATレジスターはさまざまなステータス条件を示します。

図21-5: SPIマスター/スレーブ接続



# PIC16(L)F1508/9

## 21.2.3 SPIマスター・モード

マスターはSCKxラインを制御するため、いつでもデータ転送を開始できます。マスターは、スレーブ（プロセッサ2、図21-5）がソフトウェア・プロトコルによってデータをブロードキャスト（複数の受信者に同時に転送）するタイミングを決定します。

マスター・モードでは、SSPxBUFレジスターに書き込まれるとすぐにデータが送受信されます。SPIが受信のみを行う場合は、SDOx出力を無効にすることができます（入力としてプログラムされます）。SSPxSRレジスターは、プログラムされたクロック・レートでSDIxピンに存在する信号をシフトし続けます。各バイトが受信されると、通常の受信バイトのようにSSPxBUFレジスターにロードされます（割り込みとステータスピットが適切に設定されます）。

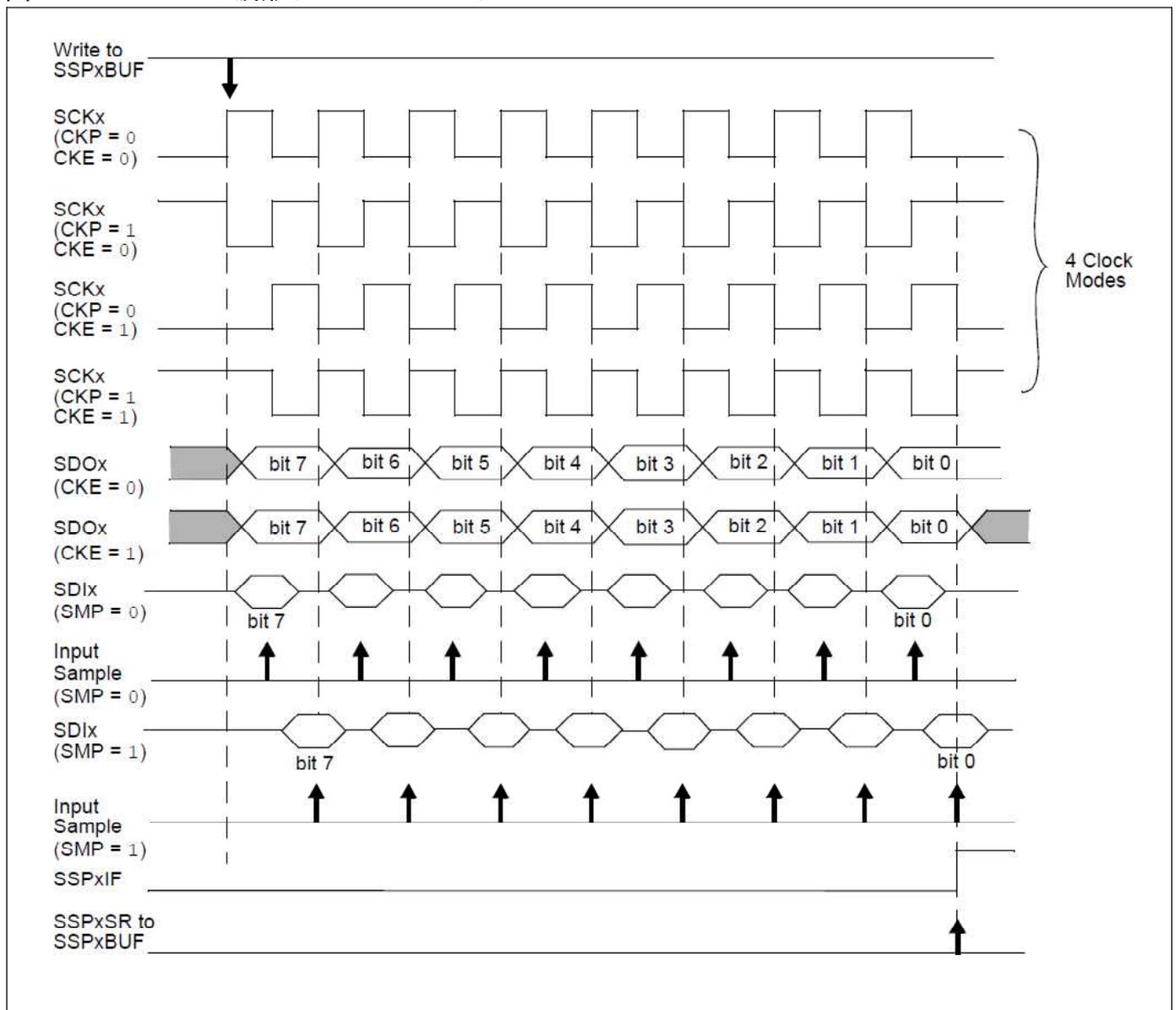
クロックの極性は、SSPxCON1レジスタのCKPビットとSSPxSTATレジスタのCKEビットを適切にプログラミングすることによって選択されます。これにより、図21-6、図21-8、図21-9、および図21-10に示すように、SP通信の波形が得られます。MSbが最初に送信されます。マスター・モードでは、SPIクロック・レート（ビット・レート）はユーザーがプログラム可能で、次のいずれかになります：

- $F_{osc}/4$ （または $T_{CY}$ ）
- $F_{osc}/16$ （または $4 \times T_{CY}$ ）
- $F_{osc}/64$ （または $16 \times T_{CY}$ ）
- タイマー2出力/2
- $F_{osc}/(4 \times (SSPxADD + 1))$

図21-6に、マスター・モードの波形を示します。

CKEビットがセットされている場合、SCKxにクロック・エッジが発生する前にSDOxデータが有効になります。入力サンプルの変化は、SMPビットの状態に基づいて表示されます。受信したデータがSSPxBUFにロードされた時刻が表示されます。

図21-6: SPIモード波形(マスター・モード)



### 21.2.4 SPIスレーブ・モード

スレーブ・モードでは、外部クロック・パルスがSCKxに現れると、データが送受信されます。最後のビットがラッチされると、SSPxIF割り込みフラグ・ビットがセットされます。

モジュールをSPIスレーブ・モードで有効にする前に、クロック・ラインが適切なアイドル状態と一致している必要があります。クロック・ラインは、SCKxピンを読み取ることで確認できます。アイドル（不動）状態は、SSPxCON1レジスタのCKPビットによって決定されます。

スレーブ・モードでは、外部クロックはSCKxピンの外部クロック・ソースから供給されます。この外部クロックは、電氣的仕様で指定されている最小のハイ・タイムとロー・タイムを満たす必要があります。

スレーブ・モード中、スレーブはデータを送受信できます。シフト・レジスタはSCKxピン入力からクロックされ、バイトが受信されると、デバイスは割り込みを生成します。有効になると、デバイスはスリープからウェイクアップします。

#### 21.2.4.1 デイジーチェーン構成

SPIバスは、デイジーチェーン構成で接続できる場合があります。最初のスレーブ出力は2番目のスレーブ入力に接続され、2番目のスレーブ出力は3番目のスレーブ入力に接続されます。最終的なスレーブ出力はマスター入力に接続されます。各スレーブは、クロック・パルスの2番目のグループの間に、クロック・パルスの最初のグループの間に受信されたものの正確なコピーを送信します。チェーン全体が1つの大きな通信シフト・レジスタとして機能します。デイジーチェーン機能は、マスター・デバイスからの単一のスレーブ選択ラインのみを必要とします。

図21-7に、SPIモードで動作する場合の一般的なデイジーチェーン接続のブロック図を示します。

デイジーチェーン構成では、バス上の最新のバイトのみがスレーブに必要です。SSPxCON3レジスタのBOENビットを設定すると、前のバイトが読み取られていなくても、SSPxBUFレジスタへの書き込みが有効になります。これにより、ソフトウェアは、適用されない可能性のあるデータを無視できます。

### 21.2.5 スレーブ選択の同期

スレーブ選択を使用して、通信を同期することもできます。マスター・デバイスが通信する準備ができるまで、スレーブ選択ラインはハイに保持されます。スレーブ選択ラインがローに引き下げられると、スレーブは新しい送信が開始されていることを認識します。

スレーブが通信を適切に受信できない場合、スレーブ選択ラインがハイ状態に戻ったときに、送信の最後にリセットされます。スレーブは、スレーブ選択ラインが再びローに引き下げられると、新しい送信を受信する準備が整います。スレーブ選択ラインを使用しない場合、スレーブが最終的にマスターと同期しなくなるリスクがあります。スレーブがビットを逃した場合、将来の送信では常に1ビットオフになります。スレーブ選択ラインを使用すると、スレーブとマスターは各送信の開始時に位置合わせできます。

$\overline{\text{SSx}}$ ピンは、同期スレーブ・モードを許可します。SPIは、 $\overline{\text{SSx}}$ ピン制御が有効になっているスレーブ・モードである必要があります（SSPxCON1<3:0> = 0100）。

$\overline{\text{SSx}}$ ピンがローの場合、送信と受信が有効になり、SDOxピンが駆動されます。

SSxピンがハイになると、送信バイトの途中でフローティング（浮動）出力になったとしても、SDOxピンは駆動されなくなります。アプリケーションによっては、外部プルアップ/プルダウン抵抗が望ましい場合があります。

- |   |
|---|
| <p><b>注1:</b> SPIが<math>\overline{\text{SSx}}</math>ピン制御が有効なスレーブ・モード（SSPxCON1&lt;3:0&gt; = 0100）の場合、<math>\overline{\text{SSx}}</math>ピンがVDDに設定されているとSPIモジュールはリセットされます。</p> <p><b>2:</b> SPIがCKEが設定されたスレーブ・モードで使用される場合。ユーザーは<math>\overline{\text{SSx}}</math>ピン制御を有効にする必要があります。</p> <p><b>3:</b> SPIスレーブ・モードで動作している間、SSPxSTATレジスタのSMPビットはクリアのままである必要があります。</p> |
|---|

SPIモジュールがリセットされると、ビットカウンターは強制的に「0」になります。これは、 $\overline{\text{SSx}}$ ピンを強制的にハイ・レベルにするか、SSPENビットをクリアすることで実行できます。

# PIC16(L)F1508/9

図21-7: SPIダイジーチェーン接続

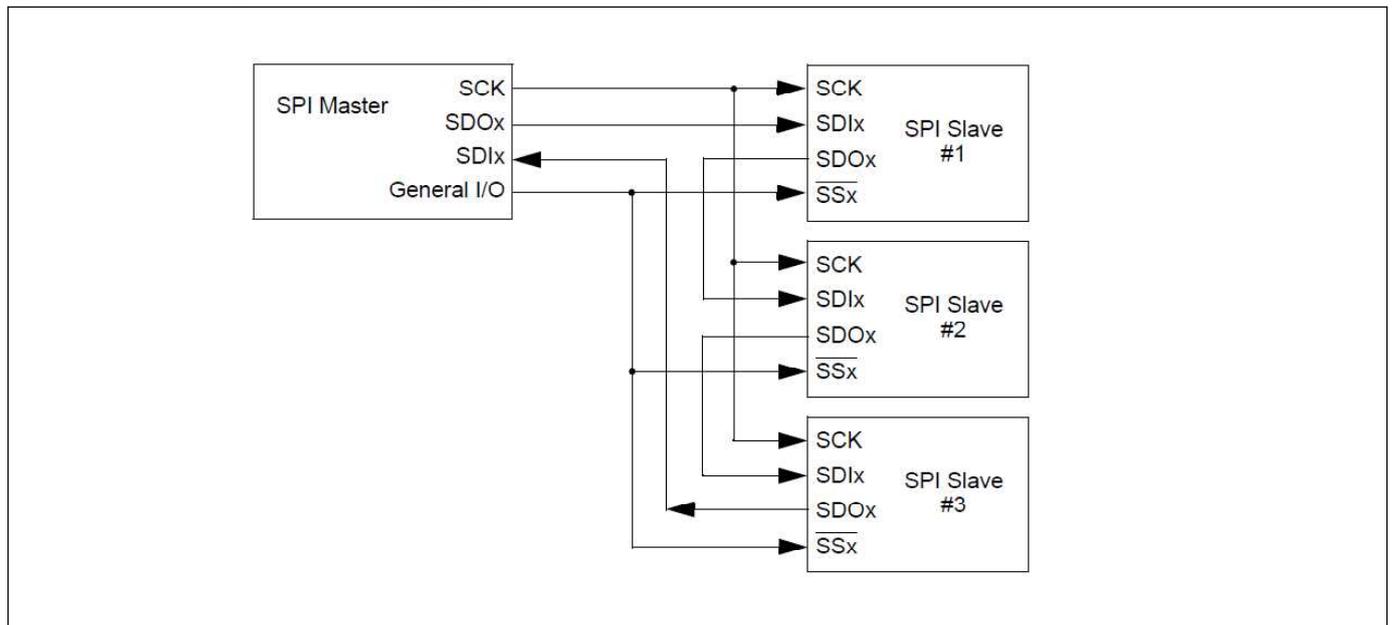


図21-8: スレーブ選択同期波形

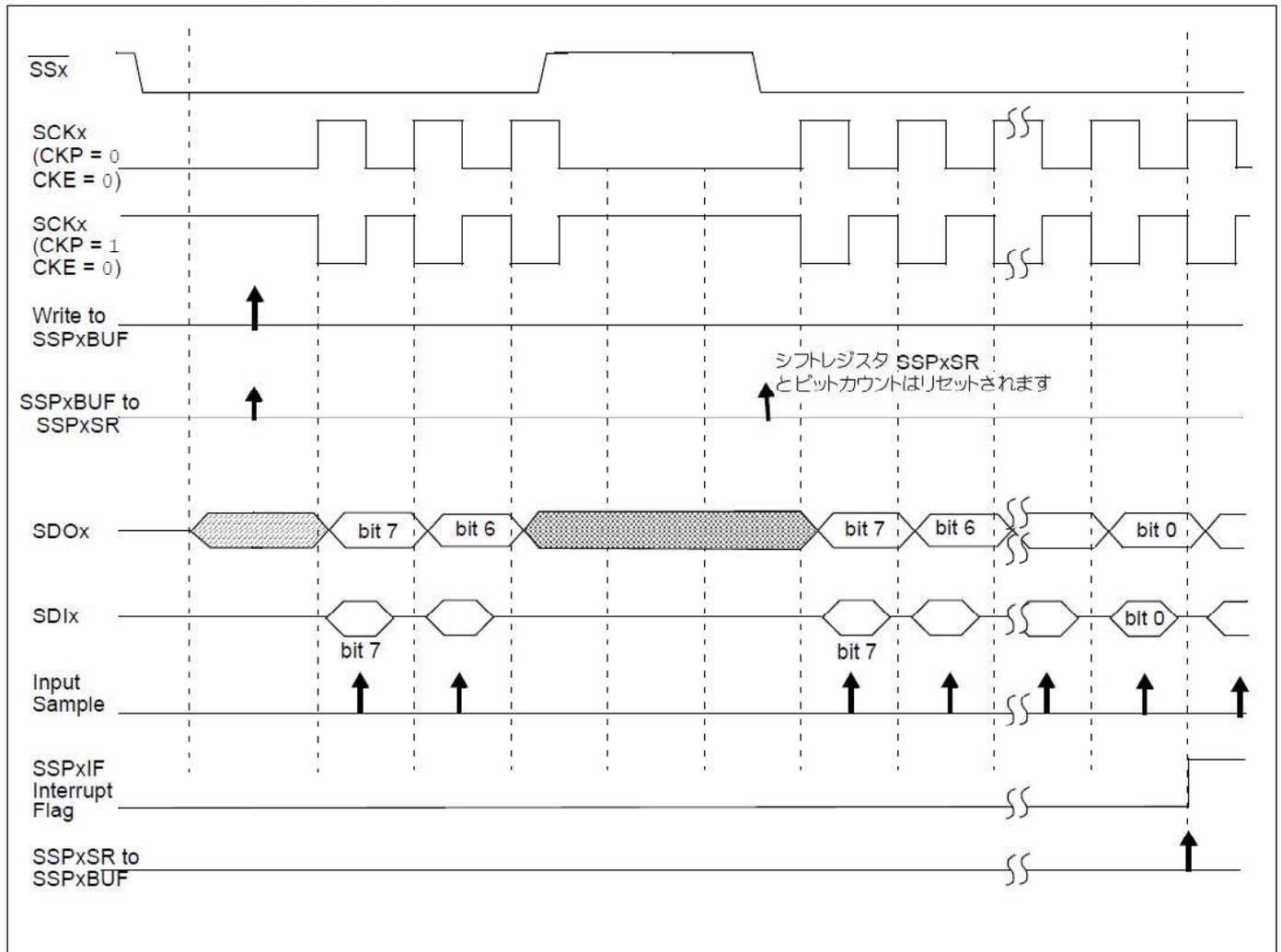


図21-9: SPIモード波形(CKE = 0のスレーブ・モード)

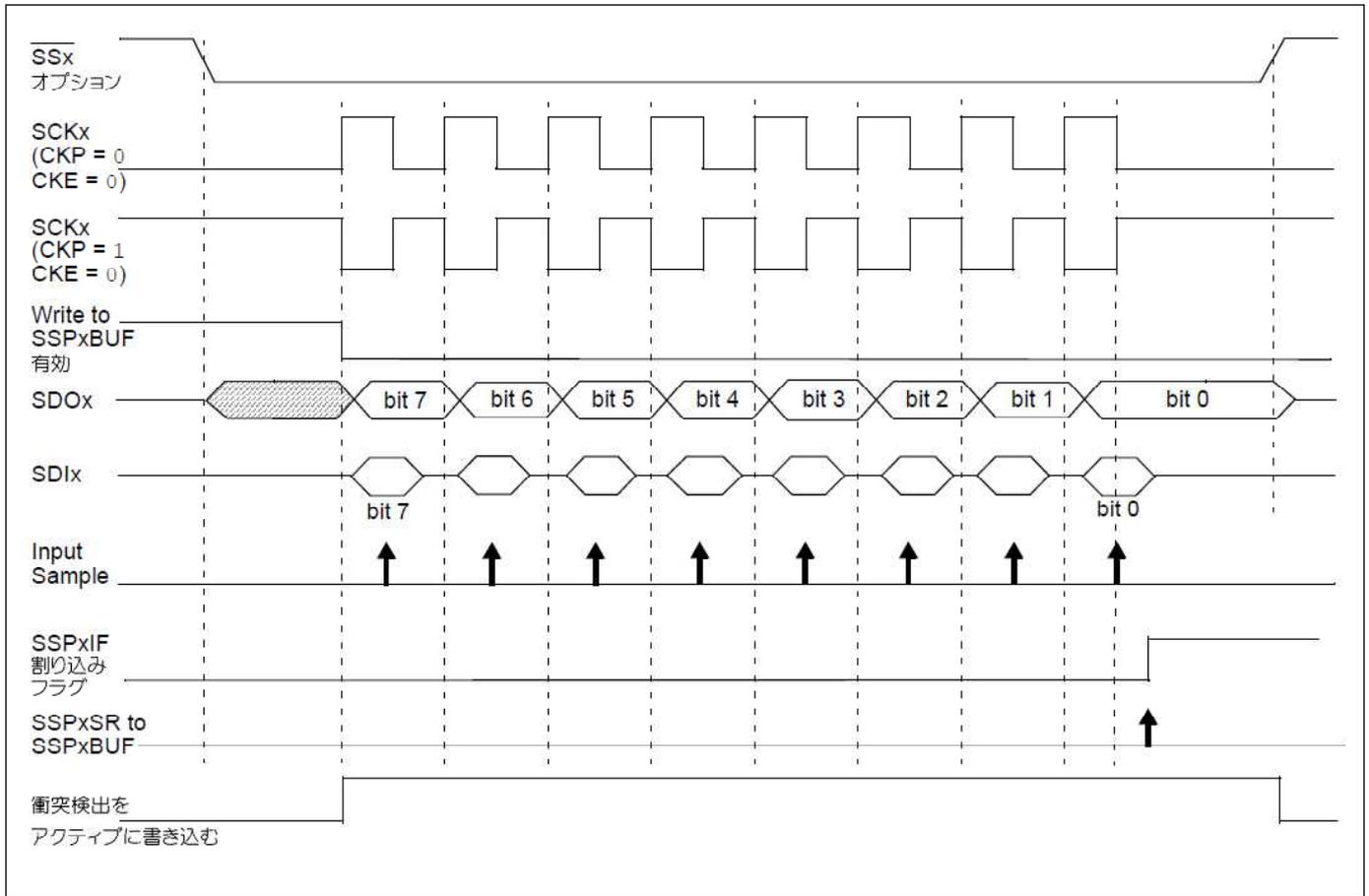
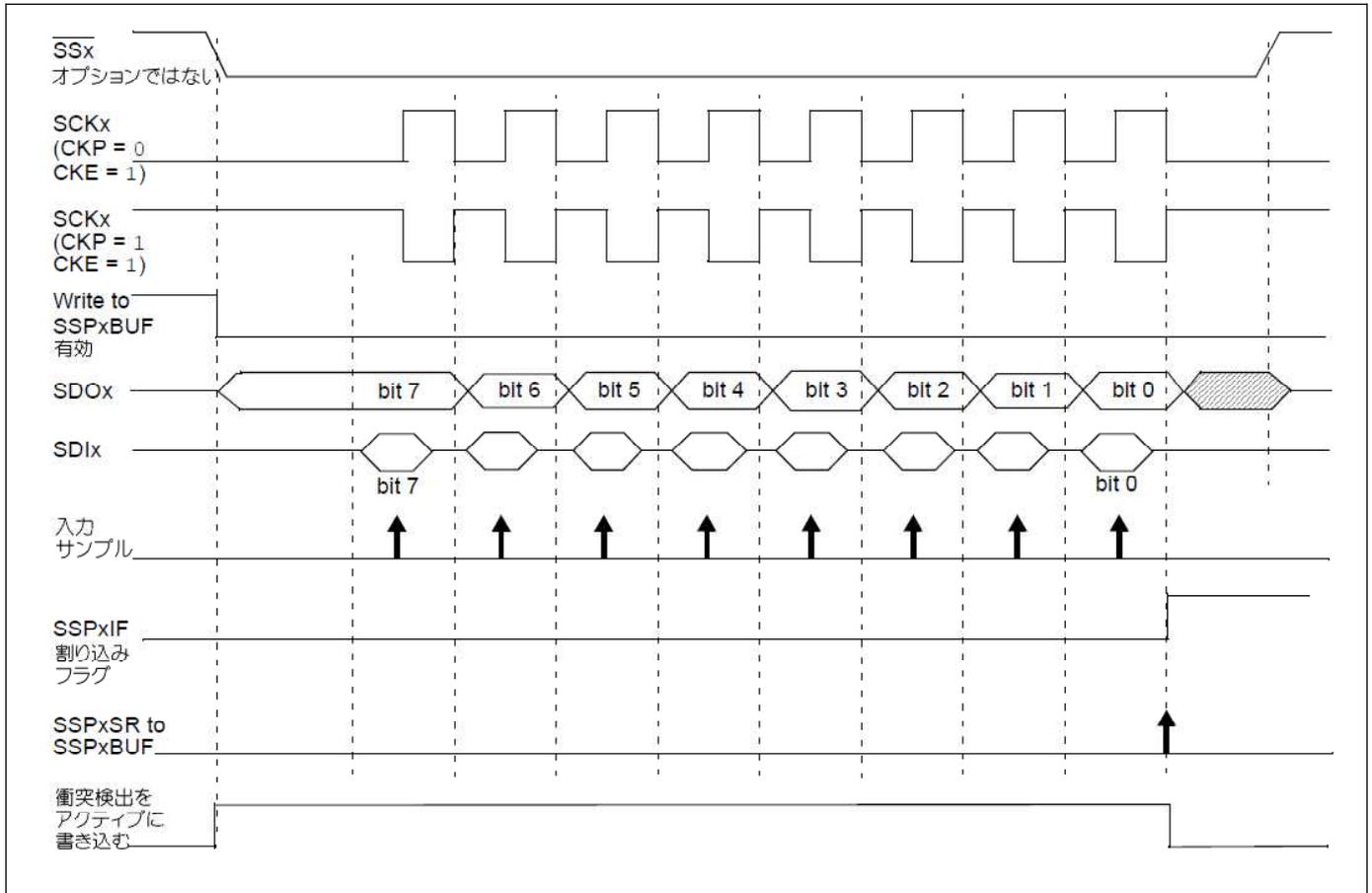


図21-10: SPIモード波形(CKE = 1のスレーブモード)



# PIC16(L)F1508/9

## 21.2.6 スリープモードでのSPI動作

SPIマスターモードでは、モジュールクロックはフルパワーモードの場合とは異なる速度で動作している可能性があります。スリープモードの場合、すべてのクロックが停止します。

MSSPクロックがシステムクロックよりもはるかに速い場合は、ユーザーが特別な注意を払う必要があります。

スリープモードでは、MSSP割り込みが有効になっていると、マスターがデータの送信を完了した後、MSSP割り込みによってコントローラーがスリープから目覚めます。

スリープモードを終了したくない場合は、MSSP割り込みを無効にする必要があります。

SPIマスターモードでは、スリープモードが選択されると、すべてのモジュールクロックが停止し、デバイスがウェイクアップするまで送受信はその状態のままになります。デバイスが実行モードに戻った後、モジュールはデータの送受信を再開します。

SPIスリープモードでは、SPI送信/受信シフトレジスタはデバイスと非同期に動作します。これにより、デバイスをスリープモードにし、データをSPI送信/受信シフトレジスタにシフトすることができます。8ビットすべてが受信されると、MSSP割り込みフラグビットが設定され、有効になっている場合はデバイスがウェイクアップします。

表21-1: SPI操作に関連するレジスターの要約

名前	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ページ上の登録
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	110
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	75
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	—	TMR2IE	TMR1IE	76
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	—	TMR2IF	TMR1IF	79
SSP1BUF	同期シリアルポート受信バッファ/送信レジスター								173*
SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>				219
SSP1CON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	221
SSP1STAT	SMP	CKE	D/A	P	S	R/W	UA	BF	218
TRISA	—	—	TRISA5	TRISA4	— <sup>(1)</sup>	TRISA2	TRISA1	TRISA0	109
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	117

記録： — =実装されていない場所、「0」と読みます。網掛けのセルは、SPIモードのMSSPでは使用されません。

\* ページは登録情報を提供します。

注1: 未実装、「1」と読みます。

## 21.3 I<sup>2</sup>Cモード概要

Inter-Integrated Circuit Bus (I<sup>2</sup>C) (複数の通信主体が同じ信号線を共有するバス型の接続方式) は、マルチ・マスター・シリアル・データ通信バスです。デバイスは、マスター・デバイスが通信を開始するマスター/スレーブ環境で通信します。スレーブ・デバイスは、アドレス指定によって制御されます。

I<sup>2</sup>Cバスは2つの信号接続を指定します：

- シリアル・クロック (SCLx)
- シリアル・データ (SDAx)

図21-2および図21-3に、I<sup>2</sup>Cモードで動作しているときのMSSPモジュールのブロック図を示します。

SCLx接続とSDAx接続はどちらも双方向のオープン・ドレイン・ラインであり、それぞれに電源電圧用のプルアップ抵抗が必要です。ラインをグラウンドに引き寄せることは論理ゼロと見なされ、ラインをフロートさせることは論理ゼロと見なされません。

図21-11は、マスター・デバイスとスレーブ・デバイスとして構成された2つのプロセッサ間の一般的な接続を示しています。

I<sup>2</sup>Cバスは、1つ以上のマスター・デバイスと1つ以上のスレーブ・デバイスで動作できます。

特定のデバイスには、4つの潜在的な動作モードがあります：

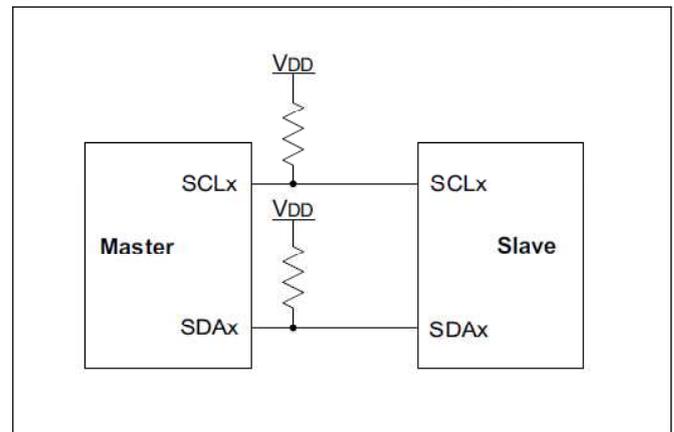
- マスター送信モード  
(マスターはスレーブにデータを送信しています)
- マスター受信モード  
(マスターはスレーブからデータを受信しています)
- スレーブ送信モード  
(スレーブはマスターにデータを送信しています)
- スレーブ受信モード  
(スレーブはマスターからデータを受信しています)

通信を開始するために、マスター・デバイスはマスター送信モードで起動します。マスター・デバイスは、スタート・ビットに続いて、通信しようとしているスレーブのアドレス・バイトを送信します。この後に単一の読み取り/書き込みビットが続き、マスターがスレーブ・デバイスとの間でデータを送信するか受信するかを決定します。

要求されたスレーブがバス上に存在する場合、確認応答ビット (ACKとも呼ばれます) で応答します。次に、マスターは送信モードまたは受信モードのいずれかで続行し、スレーブはそれぞれ受信モードまたは送信モードのいずれかで補完を続行します。

スタート・ビットは、SCLxラインがハイに保持されている間のSDAxラインのハイからローへの遷移によって示されます。アドレスとデータ・バイトが送信され、最上位ビット (MSb) が最初に送信されます。読み取り/書き込みビットは、マスターがスレーブからデータを読み取りようとする論理1として送信され、スレーブにデータを書き込もうとする論理0として送信されます。

図21-11: I<sup>2</sup>Cマスター/スレーブ接続



確認応答ビット (ACK) はアクティブ・ロー信号であり、SDAxラインをローに保持して、スレーブ・デバイスが送信データを受信し、さらに受信する準備ができていることを送信機に示します。

データ・ビットの遷移は、SCLxラインがローに保持されている間は常に実行されます。SCLxラインがハイに保持されている間に発生する遷移は、スタート・ビットとストップ・ビットを示すために使用されます。

マスターがスレーブに書き込む場合は、データのバイトを繰り返し送信し、各バイトの後にスレーブがACKビットで応答します。この例では、マスター・デバイスはマスター送信モードであり、スレーブはスレーブ受信モードです。

マスターがスレーブから読み取る場合は、スレーブから1バイトのデータを繰り返し受信し、各バイトの後にACKビットで応答します。この例では、マスター・デバイスはマスター受信モードであり、スレーブはスレーブ送信モードです。

通信されたデータの最後のバイトで、マスター・デバイスはストップ・ビットを送信することによって送信を終了することができます。マスター・デバイスが受信モードの場合、最後のACKビットの代わりにストップ・ビットを送信します。ストップ・ビットは、SCLxラインがハイに保持されている間のSDAxラインのローからハイへの遷移によって示されます。

場合によっては、マスターはバスの制御を維持し、別の送信を再開したい場合があります。その場合、マスター・デバイスは、受信モードのときに、ストップ・ビットまたは最後のACKビットの代わりに別のスタート・ビットを送信する場合があります。

I<sup>2</sup>Cバスは3つのメッセージ・プロトコルを指定します：

- マスターがスレーブにデータを書き込む単一のメッセージ。
- マスターがスレーブからデータを読み取る単一のメッセージ。
- マスターが1つ以上のスレーブに対して、最低2回の書き込み、2回の読み取り、または書き込みと読み取りの組み合わせを開始する複合メッセージ。

1つのデバイスが論理デバイスを送信しているとき、またはラインをフロートさせているとき、2番目のデバイスが論理ゼロを送信しているとき、またはラインをローに保持しているとき、最初のデバイスはラインが論理的なものではないことを検出できます。この検出は、SCLxラインで使用される場合、クロック・ストレッチと呼ばれます。クロック・ストレッチは、スレーブ・デバイスにデータの流れを制御するメカニズムを提供します。この検出がSDAx回線で使われる場合、アービトレーション（仲裁）と呼ばれます。アービトレーションにより、一度に通信するマスター・デバイスは1つだけになります。

## 21.3.1 クロック・ストレッチング(延び?)

スレーブ・デバイスがデータの処理を完了していない場合、クロック・ストレッチのプロセスを通じてより多くのデータの転送を遅らせることができます。アドレス指定されたスレーブ・デバイスは、ビットを受信または送信した後、SCLxクロック・ラインをローに保持する場合があります。これは、続行する準備がまだ整っていないことを示します。スレーブと通信しているマスターは、次のビットを転送するためにSCLxラインを上げようとしていますが、クロック・ラインがまだ解放されていないことを検出します。SCLx接続はオープン・ドレインであるため、スレーブは通信を継続する準備ができるまでそのラインをローに保持することができます。

クロック・ストレッチにより、送信機に追いつけない受信機が着信データの流れを制御できます。

## 21.3.2 アービトレーション(仲裁)

各マスター・デバイスは、スタート・ビットとストップ・ビットについてバスを監視する必要があります。バスがビジー（忙しい）であることをデバイスが検出した場合、バスがアイドル（不動）状態に戻るまで、デバイスは新しいメッセージを開始できません。

ただし、2つのマスター・デバイスが、またはほぼ同時に送信を開始しようとする場合があります。これが発生すると、調停のプロセスが始まります。各送信機はSDAxデータ・ラインのレベルをチェックし、それを検出する予定のレベルと比較します。2つのレベルが一致せず、アービトレーションを失い、そしてSDAxラインで転送しているのを停止しなければなりません。

たとえば、1つの送信機がSDAxラインを論理1に保持し（フロートさせ）、2番目の送信機がSDAxラインを論理ゼロに保持（ローにプル(引く)）すると、SDAxラインはローになります。次に、最初の送信機は、回線のレベルが予想とは異なることを確認し、別の送信機が通信していると結論付けます。

この違いに最初に気付いた送信機は、アービトレーションを失い、SDAx回線の駆動を停止する必要がある送信機です。この送信機がマスター・デバイスでもある場合は、SCLxラインの駆動も停止する必要があります。次に、送信を再発行する前に、停止状態について回線を監視できます。その間、SDAxラインの予想レベルと実際のレベルの違いに気づかなかった他のデバイスは、元の送信を続行します。これまでのところ、他の送信機がメッセージを妨害することなく、送信は期待どおりに表示されるため、問題なく実行できます。

マスターが複数のスレーブをアドレス指定する場合、スレーブ送信モードも調停できますが、これはあまり一般的ではありません。

2つのマスター・デバイスがアドレス・ステージで2つの異なるスレーブ・デバイスにメッセージを送信している場合、下位のスレーブ・アドレスを送信するマスターは常にアービトレーションに勝ちます。2つのマスター・デバイスが同じスレーブ・アドレスにメッセージを送信し、アドレスが複数のスレーブを参照する場合がある場合、アービトレーション手順はデータ・ステージまで続行する必要があります。

通常、調停はめったに発生しませんが、適切なマルチ・マスター・サポートに必要な手順です。

## 21.4 I<sup>2</sup>Cモード動作

すべてのMSSP I<sup>2</sup>C通信はバイト指向であり、最初にMSbからシフトアウトされます。6つのSFRレジスターと2つの割り込みフラグが、モジュールをPICRマイクロコントローラーおよびユーザー・ソフトウェアとインターフェイスさせます。モジュールは、SDAxとSCLxの2つのピンを使用して、他の外部I<sup>2</sup>Cデバイスと通信します。

### 21.4.1 バイト・フォーマット

I<sup>2</sup>Cでのすべての通信は、9ビット・セグメントで行われます。バイトがマスターからスレーブに、またはその逆に送信され、続いて確認応答ビットが返送されます。SCLxラインの8番目の立ち下がりがエッジの後、SDAxにデータを入力するデバイスは、そのピンを入力に変更し、次のクロックパルスで確認応答値を読み込みます。

クロック信号SCLxは、マスターによって提供されます。データは、SCLx信号がローのときに変更でき、クロックの立ち上がりエッジでサンプリングされます。SCLxラインがハイのときのSDAxラインの変更は、バス上の特別な条件を定義します。これについては以下で説明します。

### 21.4.2 I<sup>2</sup>C用語の定義

I<sup>2</sup>C通信の説明には、I<sup>2</sup>Cに固有の定義を持つ言語と用語があります。その単語の使用法は以下に定義されており、説明なしでこのドキュメントの残りの部分で使用される場合があります。この表は、Philips (フィリップス) I<sup>2</sup>C™仕様を基にしています。

### 21.4.3 SDAxおよびSCLxピン

SSPENビットが設定されたI<sup>2</sup>Cモードを選択すると、SCLxピンとSDAxピンが強制的にオープン・ドレインになります。これらのピンは、適切なTRISビットを設定することにより、ユーザーが入力に設定する必要があります。

**注釈:** I<sup>2</sup>Cモードが有効な場合、データは出力ゼロに関連付けられます。

### 21.4.4 SDAx保持時間

SDAxピンのホールド・タイム (保持時間) は、SSPxCON3レジスターのSDAHTビットによって選択されます。保持時間は、SCLxの立ち下がりがエッジ後にSDAxが有効に保持される時間です。SDAHTビットを設定すると、300 nsの最小保持時間が長くなり、容量の大きいバスに役立つ場合があります。

表21-2: I<sup>2</sup>Cバスの用語

用語	説明
送信器 (トランスミッター)	データをバスにシフトアウトするデバイス。
受信器 (レシーバー)	バスからデータをシフトインするデバイス。
マスター	転送を開始し、クロック信号を生成し、転送を終了するデバイス。
スレーブ	マスターによってアドレス指定されたデバイス。
マルチ・マスター	データ転送を開始できる複数のデバイスを備えたバス。
仲裁 (アービトラーション)	一度に1人のマスターだけがバスを制御するようにする手順。仲裁に勝つことで、メッセージが破損しないことが保証されます。
同期	バス上の2つ以上のデバイスのクロックを同期する手順。
アイドル (不働)	バスを制御しているマスターはなく、SDAxラインとSCLxラインの両方がハイです。
アクティブ (活動)	1つ以上のマスター・デバイスがバスを制御しているときはいつでも。
アドレス指定されたスレーブ	一致するアドレスを受信し、マスターによってアクティブにクロックされているスレーブ・デバイス。
一致アドレス	SSPxADDに格納されている値と一致するスレーブにクロックされるアドレス・バイト。
書き込みリクエスト	スレーブは、R/Wビットがクリアされた一致するアドレスを受信し、データをクロックインする準備ができています。
読み込みリクエスト	マスターは、R/Wビットが設定されたアドレス・バイトを送信し、スレーブからデータをクロック・アウトすることを示します。このデータは、再起動または停止するまでの次のすべてのバイトです。
クロックの延び (ストレッチ)	バス上のデバイスがSCLxをローに保持して通信を停止した場合。
バスの衝突	SDAxラインが出力中にモジュールによってLowにサンプリングされ、High状態が予想されるときはいつでも。

## 21.4.5 スタート(開始)条件

I<sup>2</sup>C仕様では、開始条件を、SCLxラインがハイのときにSDAxがハイ状態からロー状態に遷移することとして定義されています。開始条件は常にマスターによって生成され、バスがアイドル状態からアクティブ状態に移行することを示します。図21-12に、開始条件と停止条件の波形を示します。

モジュールがSDAxラインをローにアサート（論理有効）する前にローにサンプリングすると、開始条件でバスの衝突が発生する可能性があります。これは、スタート時にバスの衝突が発生しないことを規定しているI<sup>2</sup>C仕様に準拠していません。

## 21.4.6 ストップ(停止)条件

停止条件は、SCLxラインがハイのときにSDAxラインがローからハイの状態に遷移することです。

注釈： 停止が有効になるには、少なくとも1つのSCLxのロー時間が表示される必要があります。したがって、SCLxラインがハイのままSDAxラインがローになり、再びハイになると、開始条件のみが検出されます。

## 21.4.7 リスタート(再起動、再開)条件

再起動は、停止が有効になるときはいつでも有効です。マスターは、現在の転送を終了した後にバスを保持したい場合、再起動を発行できます。リスタートは、スタートと同じ効果をスレーブに与え、すべてのスレーブロジックをリセットし、アドレスをクロックインする準備をします。マスターは、同じまたは別のスレーブをアドレス指定したい場合があります。図21-13に、再起動条件の波形を示します。

10ビットのアドレス指定スレーブ・モードでは、マスターがアドレス指定されたスレーブからデータをクロックアウトするために再起動が必要です。スレーブが完全にアドレス指定され、上位アドレス・バイトと下位アドレス・バイトの両方が一致すると、マスターはR/Wビットが設定された状態で再起動と上位アドレス・バイトを発行できます。次に、スレーブ論理はクロックを保持し、データをクロックアウトする準備をします。

10ビット・モードでR/Wクリアと完全に一致した後、前の一致フラグが設定され、維持されます。停止状態になるまで、R/Wがクリアされた上位アドレス、または上位アドレスの一致は失敗します。

## 21.4.8 条件割り込みマスクの開始/停止

SSPxCON3レジスタのSCIEビットとPCIEビットは、通常この機能をサポートしないスレーブ・モードでの割り込みの生成を有効にすることができます。開始および停止検出の割り込みがすでに有効になっているスレーブ・モードでは、これらのビットは効果がありません。

図21-12: I<sup>2</sup>Cの開始条件と停止条件

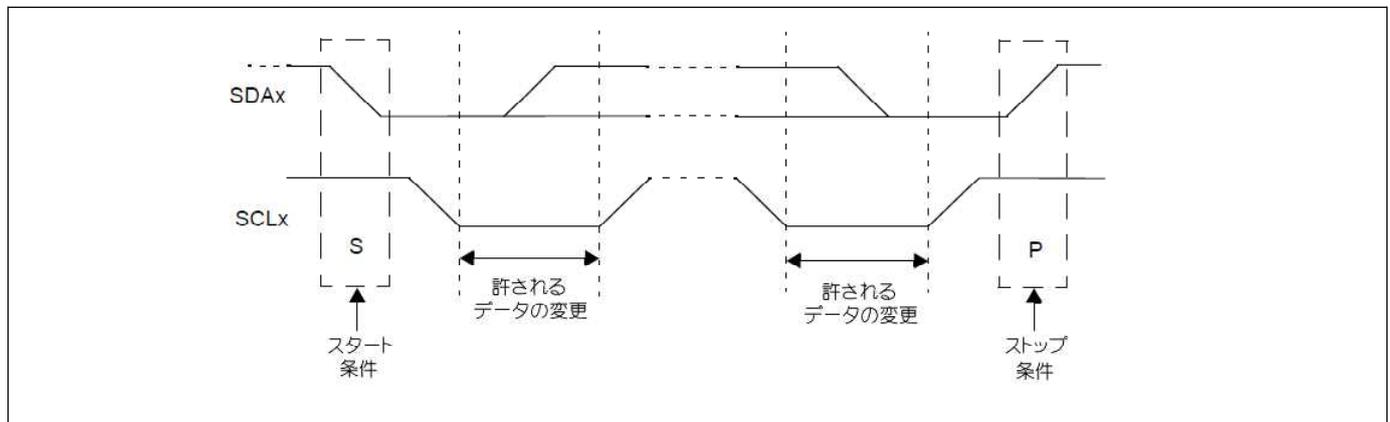
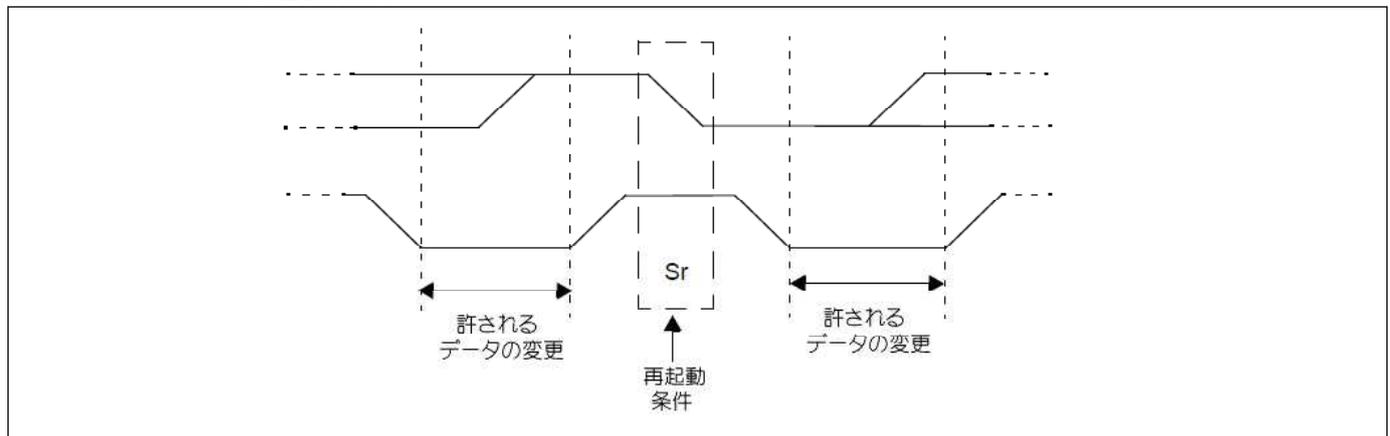


図21-13: I<sup>2</sup>Cの再起動条件



### 21.4.9 応答シーケンス

I<sup>2</sup>Cで転送されたバイトの9番目のSCLxパルスは、確認応答専用です。これにより、SDAxラインをローに引き下げることにより、受信デバイスが送信器に回答できるようになります。送信器は、回答をシフトするために、この時間中に回線の制御を解放する必要があります。確認応答 (ACK) はアクティブロー信号であり、SDAxラインをローに引き下げて、デバイスが送信データを受信し、さらに受信する準備ができていないことを送信機に示します。

ACKの結果は、SSPxCON2レジスタのACKSTATビットに配置されます。

スレーブソフトウェアは、AHENビットとDHENビットが設定されている場合、ユーザーが送信器に返送されるACK値を設定できるようにします。SSPxCON2レジスタのACKDTビットは、回答を決定するためにセット/クリアされます。

SSPxCON3レジスタのAHENビットとDHENビットがクリアされている場合、スレーブハードウェアはACK回答を生成します。

ACKがスレーブによって送信されない特定の条件があります。バイトを受信したときに、SSPxSTATレジスタのBFビットまたはSSPxCON1レジスタのSSPOVビットがセットされている場合。

モジュールがアドレス指定されると、バス上のSCLxの8番目の立ち上がりエッジの後、SSPxCON3レジスタのACKTIMビットがセットされます。ACKTIMビットは、アクティブバスの確認応答時間を示します。ACKTIMステータスビットは、AHENビットまたはDHENビットが有効になっている場合にのみアクティブになります。

## 21.5 I<sup>2</sup>Cスレーブモード動作

MSSPスレーブモードは、SSPxCON1レジスタのSSPMビットで選択された4つのモードのいずれかで動作します。モードは、7ビットと10ビットのアドレッシングモードに分けることができます。10ビットアドレッシングモードは7ビットと同じように動作しますが、より大きなアドレスを処理するためのオーバーヘッドが追加されます。

スタートビットとストップビットの割り込みがあるモードは、スタート、リスタート、またはストップ状態の検出時にSSPxIFが追加で設定される他のモードと同じように動作します。

### 21.5.1 スレーブモードアドレス

SSPxADDレジスタ (レジスタ21-6) には、スレーブモードアドレスが含まれています。開始条件または再起動条件の後に受信した最初のバイトは、このレジスタに格納されている値と比較されます。バイトが一致すると、値がSSPxBUFレジスタにロードされ、割り込みが生成されます。値が一致しない場合、モジュールはアイドル状態になり、ソフトウェアに何かが起こったことを示すものではありません。

SSPマスクレジスタ (レジスタ21-5) は、アドレス照合プロセスに影響を与えます。詳細については、セクション21.5.9「SSPxマスクレジスタ」を参照してください。

### 21.5.1.1 I<sup>2</sup>Cスレーブ7ビットアドレッシングモード

7ビットアドレッシングモードでは、アドレスが一致するかどうかを判断するときに、受信したデータバイトのLSbは無視されます。

### 21.5.1.2 I<sup>2</sup>Cスレーブ10ビットアドレッシングモード

10ビットアドレッシングモードでは、最初に受信したバイトが「1 1 1 0 A9 A8 0」のバイナリ値と比較されます。A9とA8は、10ビットアドレスの2つのMSbであり、SSPxADDレジスタのビット2と1に格納されます。

上位バイトの確認応答後、UAビットが設定され、ユーザーが下位アドレスでSSPxADDを更新するまでSCLxは下位に保持されます。下位アドレスバイトがクロックインされ、8ビットすべてがSSPxADDの下位アドレス値と比較されます。アドレスが一致していても、SSPxIFとUAが設定され、SSPxADDが更新されて上位バイトを再度受信するまでSCLxはLowに保持されます。SSPxADDが更新されると、UAビットがクリアされます。これにより、モジュールは次の通信で上位アドレスバイトを受信する準備ができています。

すべての10ビットアドレッシング通信の開始時に、書き込み要求としてのハイアドレスとローアドレスの一致が必要です。スレーブがアドレス指定されたらリスタートを発行し、R/Wビットを設定して上位アドレスをクロッキングすることにより、送信を開始できます。次に、スレーブハードウェアは読み取り要求を確認し、データをクロックアウトする準備をします。これは、スレーブが完全な上位アドレスと下位アドレスのバイト一致を受信した後にのみ有効です。

### 21.5.2 スレーブのレセプション(取得)

一致する受信アドレスバイトのR/Wビットがクリアされると、SSPxSTATレジスタのR/Wビットがクリアされます。受信したアドレスはSSPxBUFレジスタにロードされ、確認応答されます。

受信したアドレスにオーバーフロー条件が存在する場合、確認応答は与えられません。オーバーフロー条件は、SSPxSTATレジスタのビットBFが設定されるか、SSPxCON1レジスタのビットSSPOVが設定されると定義されます。SSPxCON3レジスタのBOENビットは、この動作を変更します。詳細については、レジスタ21-4を参照してください。

転送されたデータバイトごとにMSSP割り込みが生成されます。フラグビットSSPxIFは、ソフトウェアでクリアする必要があります。

SSPxCON2レジスタのSENビットがセットされると、SCLxは各受信バイトの後にローに保持されます (クロックの延び)。10ビットモードの場合を除いて、SSPxCON1レジスタのCKPビットをセットしてクロックを解放する必要があります。詳細については、セクション21.2.3「SPIマスターモード」を参照してください。

### 21.5.2.1 7ビットアドレッシングのレセプション(取得)

このセクションでは、7ビットアドレッシングモードでI<sup>2</sup>Cスレーブとして構成されたMSSPモジュールのイベントの標準シーケンスについて説明します。図21-14および図21-15は、この説明の視覚的な参照として使用されます。

これは、I<sup>2</sup>C通信を実現するために通常実行する必要があることの段階的な手順です。

1. スタートビットが検出されました。
2. SSPxSTATのSビットが設定されます。SSPxIFは、開始検出時の割り込みが有効になっている場合に設定されます。
3. R/Wビットがクリアされた一致しているアドレスを受信します。
4. スレーブはSDAxをローにプル（引く）してマスターにACKを送信し、SSPxIFビットをセットします。
5. ソフトウェアはSSPxIFビットをクリアします。
6. ソフトウェアはSSPxBUFから受信アドレスを読み取り、BFフラグをクリアします。
7. SEN = 1の場合。スレーブ・ソフトウェアは、CKPビットを設定してSCLxラインを解放します。
8. マスターはデータ・バイトをクロックアウトします。
9. スレーブはSDAxをローに駆動してマスターにACKを送信し、SSPxIFビットをセットします。
10. ソフトウェアがSSPxIFをクリアします。
11. ソフトウェアは、受信したバイトをSSPxBUFから読み取り、BFをクリアします。
12. マスターから受信したすべてのバイトに対してステップ8~12が繰り返されます。
13. マスターはSSPxSTATのPビットを設定して停止条件を送信し、バスはアイドル状態になります。

## 21.5.2.2 AHENとDHENによる7ビットの取得

AHENとDHENが設定されたスレーブ・デバイスの受信は、SCLxの8番目の立ち下がりエッジの後に追加の割り込みとクロックの延び（ストレッチ）を追加して、これらのオプションがない場合と同じように動作します。これらの追加の割り込みにより、スレーブ・ソフトウェアは、ハードウェアではなく、受信アドレスまたはデータ・バイトのどちらをACKするかを決定できます。この機能により、PMBusのサポートが追加されます。これは、このモジュールの以前のバージョンには存在しませんでした。

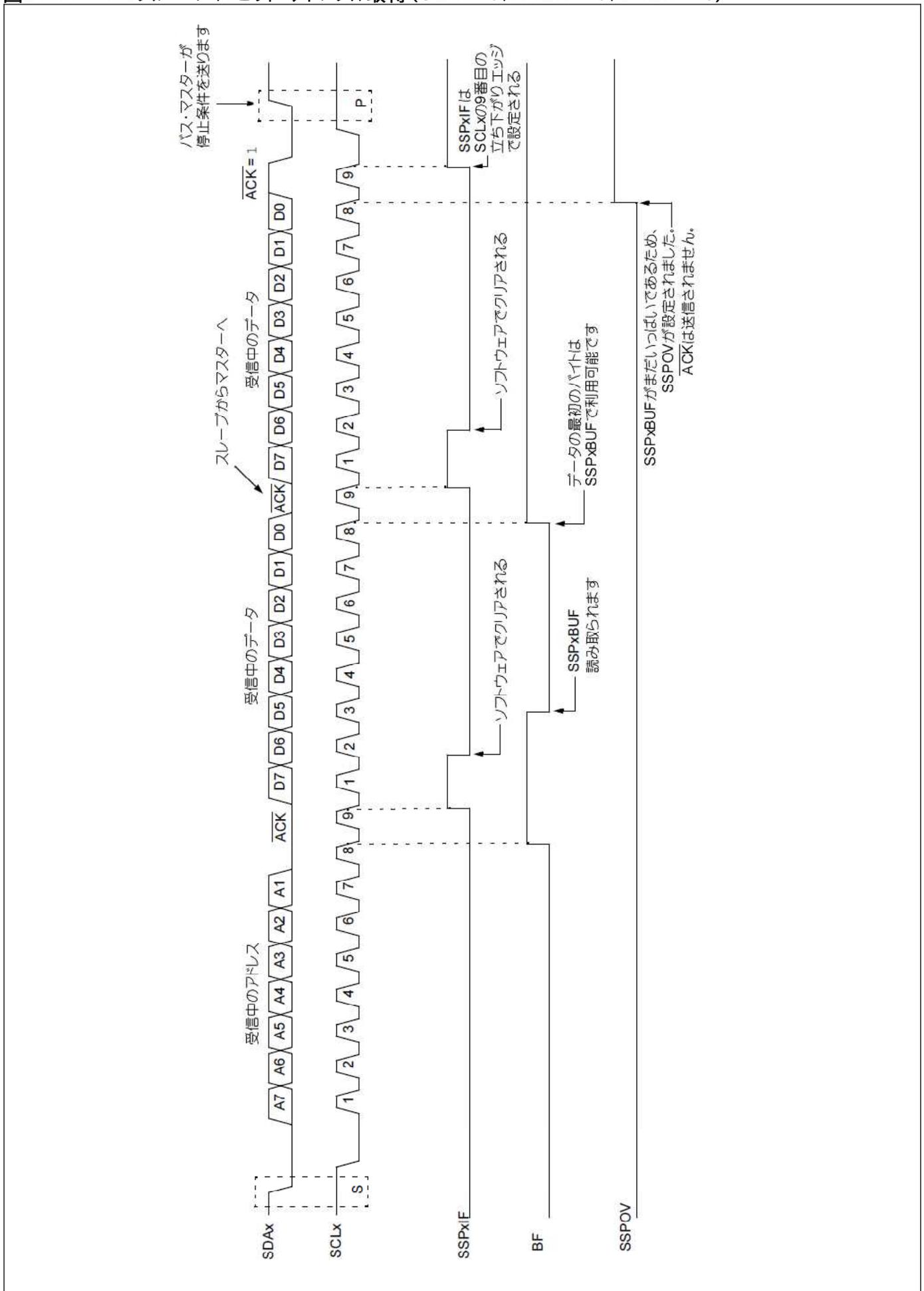
このリストでは、I<sup>2</sup>C通信にこれらのオプションを使用するためにスレーブ・ソフトウェアが実行する必要がある手順について説明します。図21-16は、アドレスとデータの両方を保持するモジュールを示しています。図21-17に、SSPxCON2レジスタのSENビットをセットした操作を示します。

1. SSPxSTATのSビットが設定されます。SSPxIFは、開始検出時の割り込みが有効になっている場合に設定されます。
2. R/Wビットクリアと一致するアドレスがクロックインされます。SSPxIFがセットされ、SCLxの8番目の立ち下がりエッジの後にCKPがクリアされます。
3. スレーブはSSPxIFをクリアします。
4. スレーブは、SSPxCON3レジスタのACKTIMビットを調べて、SSPxIFがACKの後か前かを判断できます。
5. スレーブはSSPxBUFからアドレス値を読み取り、BFフラグをクリアします。
6. スレーブは、ACKDTを設定することにより、マスターにクロック・アウトされたACK値を設定します。
7. スレーブは、CKPを設定してクロックを解放します。
8. SSPxIFは、NACKの後ではなく、ACKの後に設定されます。
9. SEN = 1の場合、スレーブ・ハードウェアはACKの後にクロックを延長します。
10. スレーブはSSPxIFをクリアします。

注釈： SSPxIFは、クロック伸縮がなく、BFがクリアされている場合でも、SCLxの9番目の立ち下がりエッジの後に設定されます。NACKがマスターに送信された場合のみSSPxIFは設定されません

11. 受信データ・バイトのSCLxの8番目の立ち下がりエッジの後にSSPxIFセットとCKPがクリアされました。
12. スレーブは、SSPxCON3のACKTIMビットを調べて、割り込みのソースを判別します。
13. スレーブはSSPxBUFから受信データを読み取り、BFをクリアします。
14. ステップ7~14は、受信した各データ・バイトで同じです。
15. 通信は、スレーブがACK = 1を送信するか、マスターが停止条件を送信することによって終了します。停止が送信され、停止検出時の割り込みが無効になっている場合、スレーブはSSPxSTATレジスタのPビットをポーリング（ホスト側が各機器に対して定期的に問い合わせを）することによってのみ認識します。

図21-14: I<sup>2</sup>Cスレーブ、7ビット・アドレス、取得 (SEN = 0, AHEN = 0, DHEN = 0)



# PIC16(L)F1508/9

図21-15: I<sup>2</sup>Cスレーブ、7ビット・アドレス、レセプション(取得)(SEN = 1、AHEN = 0、DHEN = 0)

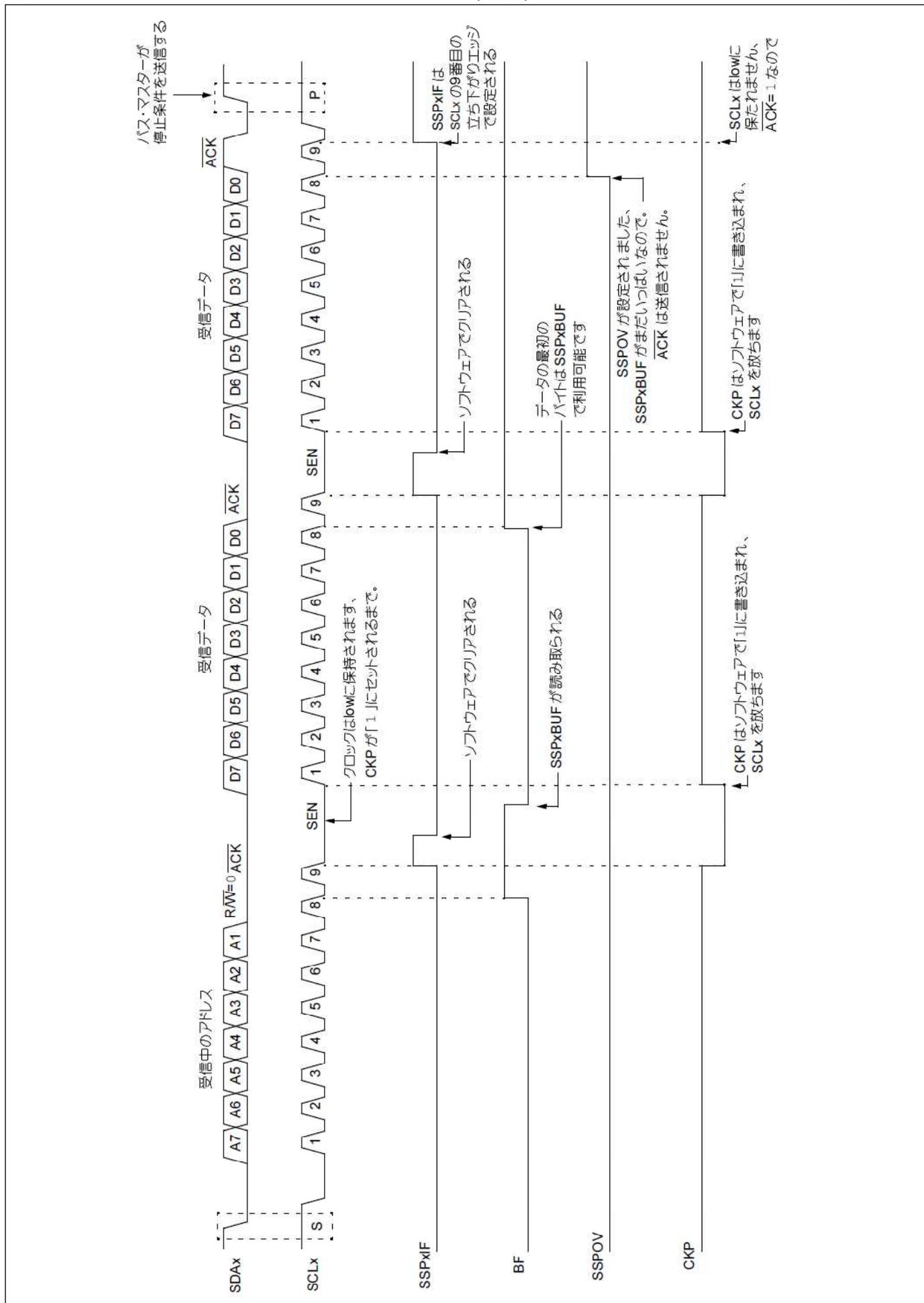
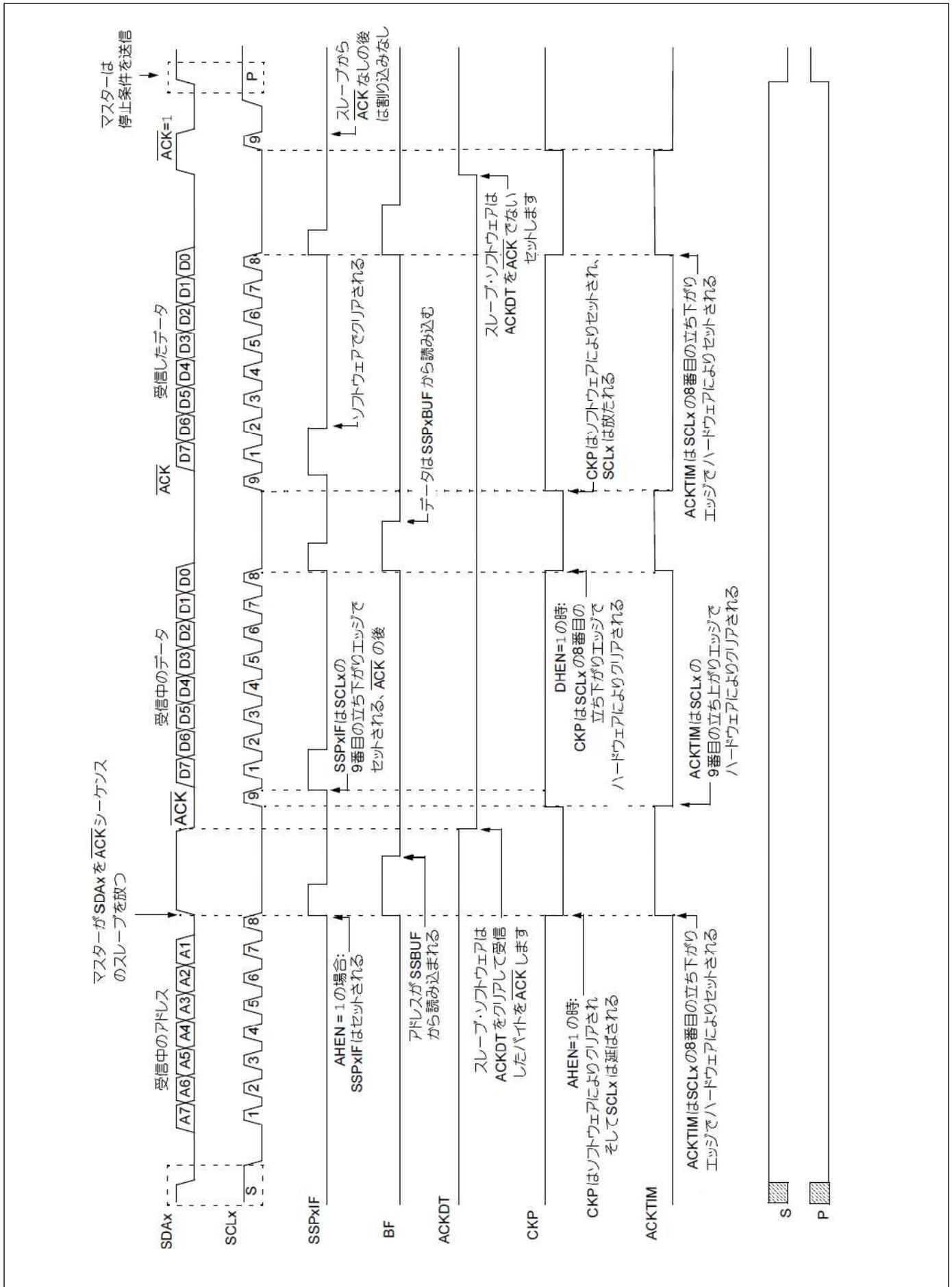
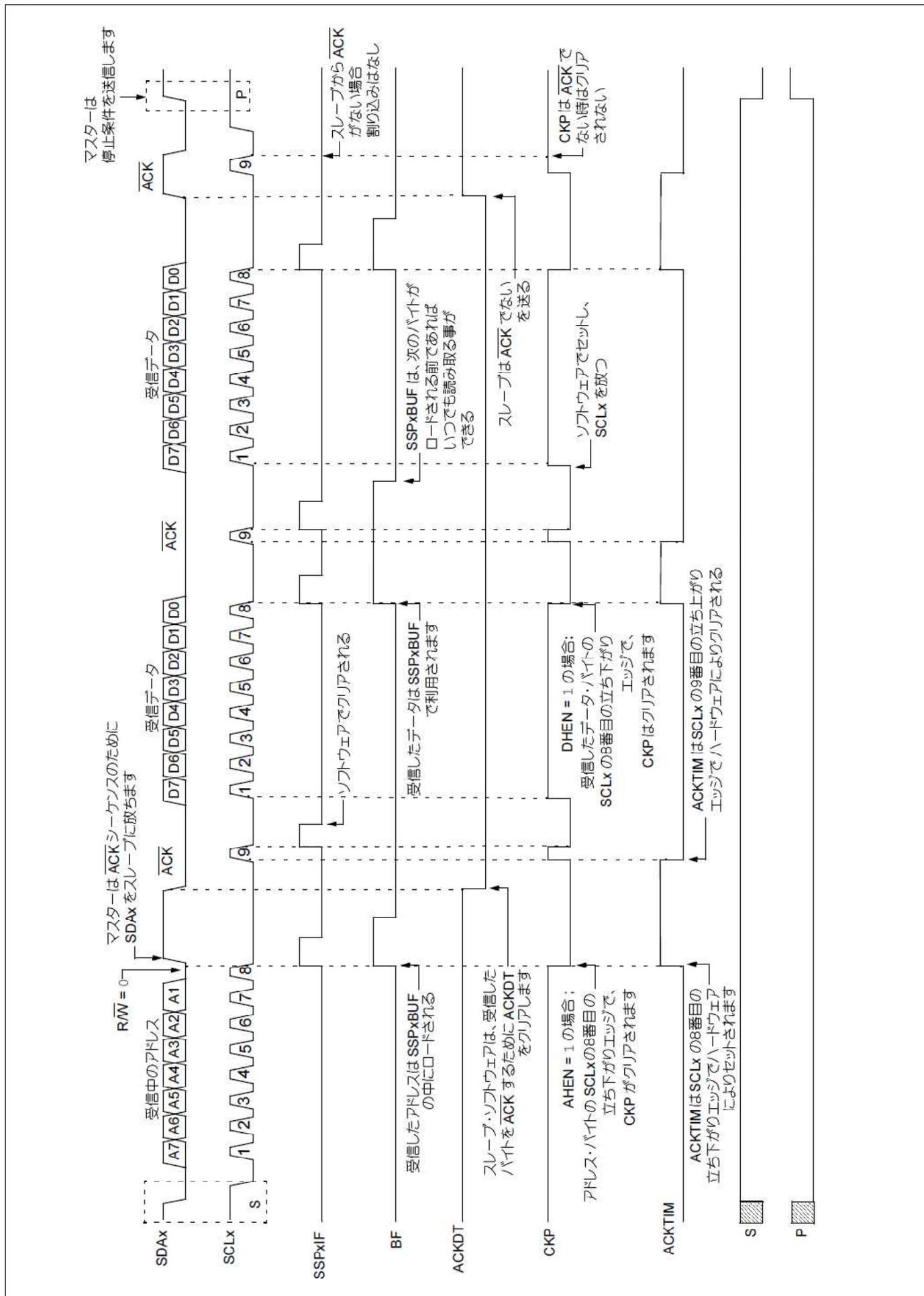


図21-16: I<sup>2</sup>Cスレーブ、7ビット・アドレス、レセプション (SEN = 0, AHEN = 1, DHEN = 1)



# PIC16(L)F1508/9

図21-17: I<sup>2</sup>Cスレーブ、7ビット・アドレス、レセプション (SEN = 1, AHEN = 1, DHEN = 1)



### 21.5.3 スレーブ送信

着信アドレスバイトの $\overline{R\overline{W}}$ ビットがセットされ、アドレスが一致すると、SSPxSTATレジスタの $\overline{R\overline{W}}$ ビットがセットされます。受信したアドレスはSSPxBUFレジスタにロードされ、9番目のビットでスレーブから $\overline{ACK}$ パルスが送信されます。

$\overline{ACK}$ に続いて、スレーブ・ハードウェアはCKPビットをクリアし、SCLxピンはローに保持されます（詳細については、セクション21.5.6「クロックの延び」を参照してください）。クロックを引き伸ばすことにより、スレーブが送信データの準備を完了するまで、マスターは別のクロック・パルスのアサート（論理有効）できなくなります。

送信データは、SSPxSRレジスタもロードするSSPxBUFレジスタにロードする必要があります。次に、SSPxCON1レジスタのCKPビットをセットしてSCLxピンを解放する必要があります。8つのデータ・ビットは、SCLx入力の立ち下がリエッジでシフトアウトされます。これにより、SDAx信号がSCLxハイ・タイム中に有効になります。

マスター受信器からの $\overline{ACK}$ パルスは、9番目のSCLx入力パルスの立ち上がりエッジでラッチされます。この $\overline{ACK}$ 値は、SSPxCON2レジスタのACKSTATビットにコピーされます。ACKSTATが設定されている場合（ $\overline{ACK}$ ではない）、データ転送は完了しています。この場合、非 $\overline{ACK}$ がスレーブによってラッチされると、スレーブはアイドル状態になり、スタート・ビットが再び発生するのを待ちます。SDAxラインがロー（ $\overline{ACK}$ ）の場合、次の送信データをSSPxBUFレジスタにロードする必要があります。この場合も、ビットCKPを設定してSCLxピンを解放する必要があります。

MSSP割り込みは、データ転送バイトごとに生成されます。SSPxIFビットはソフトウェアによってクリアされる必要があり、SSPxSTATレジスタはバイトのステータスを決定するために使用されます。SSPxIFビットは、9番目のクロック・パルスの立ち下がリエッジでセットされます。

#### 21.5.3.1 スレーブ・モード・バス衝突

スレーブは読み取り要求を受信し、SDAxラインでデータのシフトアウトを開始します。バスの衝突が検出され、SSPxCON3レジスタのSBCDEビットがセットされると、PIRxレジスタのBCLxIFビットがセットされます。バスの衝突が検出されると、スレーブはアイドル状態になり、再度アドレス指定されるのを待ちます。ユーザー・ソフトウェアは、BCLxIFビットを使用してスレーブ・バスの衝突を処理できます。

#### 21.5.3.2 7ビット送信

マスター・デバイスは、読み取り要求をスレーブに送信してから、スレーブからデータをクロックアウトできます。以下のリストは、標準の送信を実現するためにスレーブ用のソフトウェアが実行する必要があることの概要を示しています。図21-18は、このリストへの参照として使用できます。

1. マスターはSDAxとSCLxで開始条件を送信します。
2. SSPxSTATのSビットが設定されます。SSPxIFは、開始検出時の割り込みが有効になっている場合に設定されます。
3.  $\overline{R\overline{W}}$ ビットが設定された一致アドレスは、スレーブ設定SSPxIFビットで受信されます。
4. スレーブ・ハードウェアは $\overline{ACK}$ を生成し、SSPxIFを設定します。
5. SSPxIFビットはユーザーによってクリアされます。
6. ソフトウェアはSSPxBUFから受信したアドレスを読み取り、BFをクリアします。
7.  $\overline{R\overline{W}}$ が設定されているため、 $\overline{ACK}$ 後にCKPが自動的にクリアされます。
8. スレーブ・ソフトウェアは送信データをSSPxBUFにロードします。
9. CKPビットがセットされてSCLxが解放され、マスターがスレーブからデータをクロックアウトできるようになります。
10. SSPxIFは、マスターからの $\overline{ACK}$ 応答がACKSTATレジスタにロードされた後に設定されます。
11. SSPxIFビットがクリアされます。
12. スレーブソフトウェアはACKSTATビットをチェックして、マスターがさらにデータをクロックアウトするかどうかを確認します。

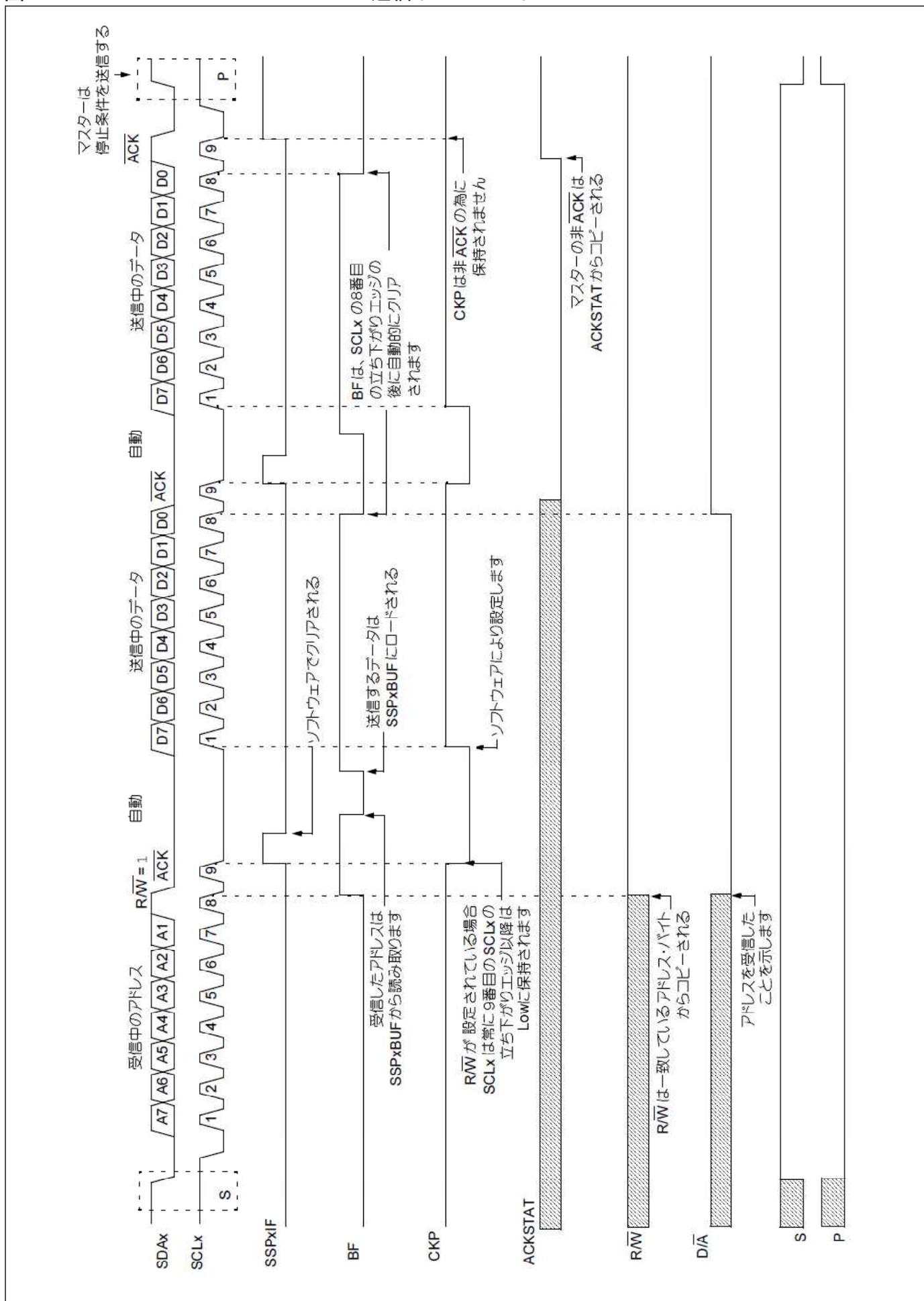
**注1:** マスターが $\overline{ACK}$ を確認すると、クロックが引き延ばされます。

**2:** ACKSTATは、SCLxの立ち上がりではなく立ち上がりエッジ（9番目）で更新される唯一のビットです。

13. 送信されたバイトごとにステップ9~13が繰り返されます。
14. マスターが非 $\overline{ACK}$ を送信した場合、クロックは保持されませんが、SSPxIFは設定されたままです。
15. マスターは再起動条件または停止を送信します。
16. スレーブはアドレス指定されなくなりました。

# PIC16(L)F1508/9

図21-18: I<sup>2</sup>Cスレーブ、7ビット・アドレス、送信 (AHEN = 0)



### 21.5.3.3 許可されたアドレス保持で7ビット送信

SSPxCON3レジスタのAHENビットを設定すると、受信した一致アドレスの8番目の立ち下がリエッジの後に追加のクロック延長と割り込み生成が可能になります。一致するアドレスがクロックインされると、CKPがクリアされ、SSPxIF割り込みが設定されます。

図21-19に、AHENを有効にした7ビット・アドレス・スレーブ送信の標準波形を示します。

1. バスがアイドル（不動）状態になります。
2. マスターは開始条件を送信します。SSPxSTATのSビットが設定されます。SSPxIFは、開始検出時の割り込みが有効になっている場合に設定されます。
3. マスターは、R/Wビットが設定された一致するアドレスを送信します。SCLxラインの8番目の立ち下がリエッジの後、CKPビットがクリアされ、SSPxIF割り込みが生成されます。
4. スレーブ・ソフトウェアはSSPxIFをクリアします。
5. スレーブ・ソフトウェアは、SSPxCON3レジスタのACKTIMビット、およびSSPxSTATレジスタのR/WとD/Aを読み取って、割り込みのソースを判別します。
6. スレーブはSSPxBUFレジスタからアドレス値を読み取り、BFビットをクリアします。
7. スレーブ・ソフトウェアは、この情報から $\overline{\text{ACK}}$ を実行するかどうかを決定し、それに応じてSSPxCON2レジスタのACKDTビットを設定します。
8. スレーブは、SCLxを解放するCKPビットを設定します。
9. マスターはスレーブからの $\overline{\text{ACK}}$ 値をクロックします。
10. R/Wビットが設定されている場合、スレーブ・ハードウェアは自動的にCKPビットをクリアし、 $\overline{\text{ACK}}$ の後にSSPxIFを設定します。
11. スレーブ・ソフトウェアはSSPxIFをクリアします。
12. スレーブは、BFビットを設定してマスターに送信する値をSSPxBUFにロードします。

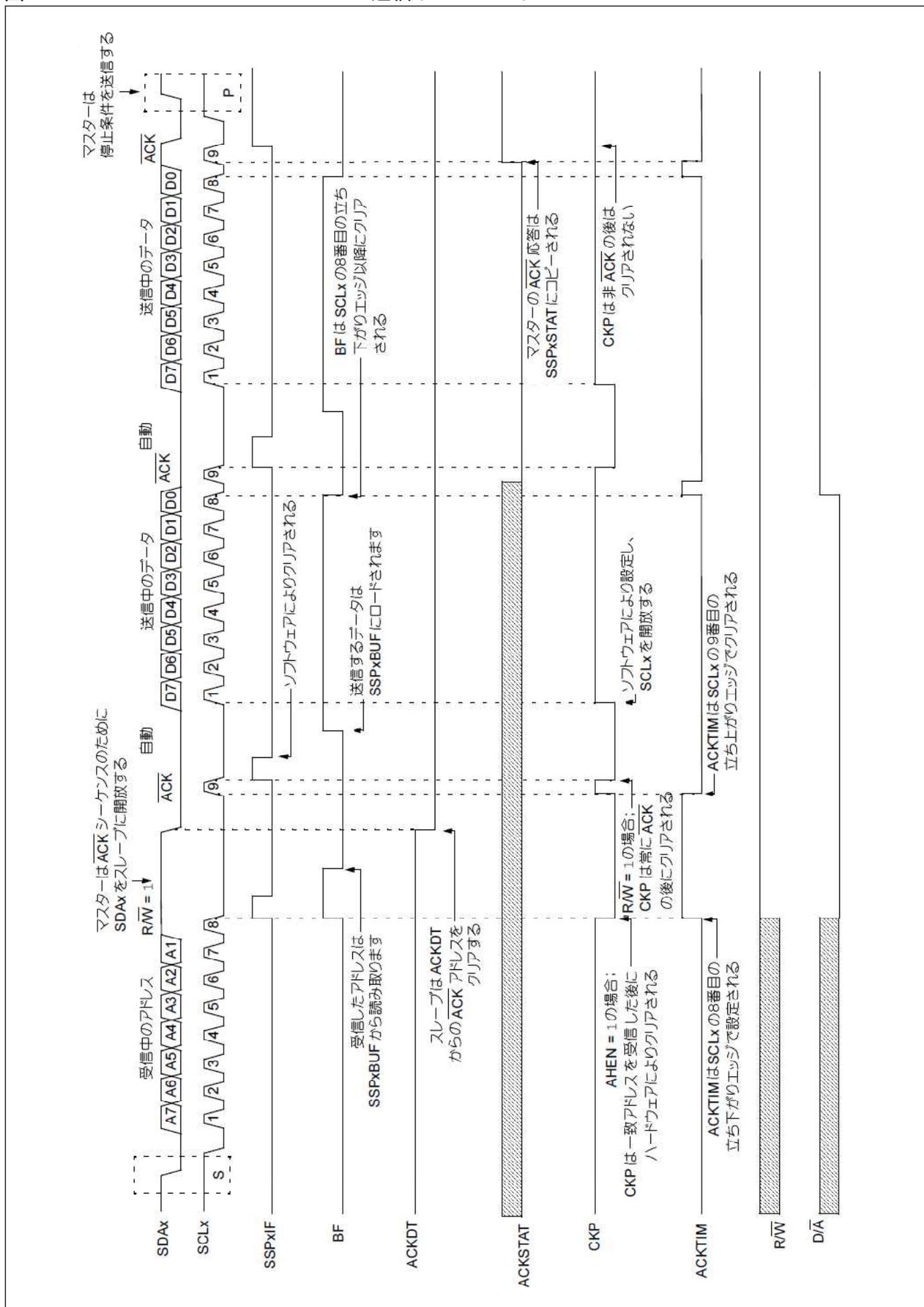
**注釈：** SSPxBUFは、 $\overline{\text{ACK}}$ が完了するまでロードできません。

13. スレーブはCKPビットを設定し、クロックを解放します。
14. マスターはスレーブからデータをクロックアウトし、9番目のSCLxパルスで $\overline{\text{ACK}}$ 値を送信します。
15. スレーブ・ハードウェアは、 $\overline{\text{ACK}}$ 値をSSPxCON2レジスタのACKSTATビットにコピーします。
16. スレーブからマスターに送信されるバイトごとにステップ10~15が繰り返されます。
17. マスターが非 $\overline{\text{ACK}}$ を送信すると、スレーブはバスを解放し、マスターが停止を送信して通信を終了できるようにします。

**注釈：** マスターは、スレーブが停止を受信するためにSCLxラインを解放することを保証するために、最後のバイトで非 $\overline{\text{ACK}}$ を送信する必要があります。

# PIC16(L)F1508/9

図21-19: I<sup>2</sup>Cスレーブ、7ビット・アドレス、送信 (AHEN = 1)



### 21.5.4 スレーブ・モード10ビット・アドレス取得

このセクションでは、10ビット・アドレッシング・モードでI<sup>2</sup>Cスレーブとして構成されたMSSPモジュールのイベントの標準シーケンスについて説明します。

図21-20は、この説明の視覚的な参照として使用されます。

これは、I<sup>2</sup>C通信を実現するためにスレーブ・ソフトウェアが実行する必要があることの段階的な処理です。

1. バスがアイドル（不動）状態になります。
2. マスターは開始条件を送信します。SSPxSTATのSビットが設定されます。SSPxIFは、開始検出時の割り込みが有効になっている場合に設定されます。
3. マスターは、一致する上位アドレスをR/Wビットをクリアして送信します。SSPxSTATレジスターのUAビットがセットされます。
4. スレーブが $\overline{\text{ACK}}$ を送信し、SSPxIFが設定されます。
5. ソフトウェアはSSPxIFビットをクリアします。
6. ソフトウェアはSSPxBUFから受信アドレスを読み取り、BFフラグをクリアします。
7. スレーブはロー・アドレスをSSPxADDにロードし、SCLxを解放します。
8. マスターは一致する下位アドレス・バイトをスレーブに送信します。UAビットが設定されます。

**注釈：** SSPxADDレジスターの更新は、 $\overline{\text{ACK}}$ シーケンスが完了するまで許可されません。

9. スレーブが $\overline{\text{ACK}}$ を送信し、SSPxIFが設定されます。

**注釈：** 下位アドレスが一致しない場合でも、SSPxIFとUAは設定されているため、スレーブ・ソフトウェアはSSPxADDを上位アドレスに戻すことができます。一致するものがないため、BFは設定されていません。CKPは影響を受けません。

10. スレーブはSSPxIFをクリアします。
11. スレーブは、受信した一致アドレスをSSPxBUFから読み取り、BFをクリアします。
12. スレーブは上位アドレスをSSPxADDにロードします。
13. マスターはデータ・バイトをスレーブにクロックアウトし、9番目のSCLxパルスでスレーブACKをクロックアウトします。SSPxIFが設定されます。
14. SSPxCON2のSENビットがセットされている場合、CKPはハードウェアによってクリアされ、クロックが引き伸ばされます。
15. スレーブはSSPxIFをクリアします。
16. スレーブはSSPxBUFから受信したバイトを読み取り、BFをクリアします。
17. SENが設定されている場合、スレーブはCKPを設定してSCLxを解放します。
18. ステップ13~17は、受信したバイトごとに繰り返されます。
19. マスターはStopを送信して送信を終了します。

### 21.5.5 アドレスまたはデータ保持での10ビット・アドレッシング

AHENまたはDHENを設定した10ビット・アドレッシングを使用した受信は、7ビット・モードの場合と同じです。唯一の違いは、UAビットを使用してSSPxADDレジスターを更新する必要があることです。特にCKPビットがクリアされ、SCLxラインがローに保持されている場合、すべての機能は同じです。図21-21は、AHENが設定された10ビット・アドレッシングのスレーブのリファレンスとして使用できます。

図21-22に、10ビット・アドレッシング・モードのスレーブ送信の標準波形を示します。

# PIC16(L)F1508/9

図21-20: I<sup>2</sup>Cスレーブ、10ビット・アドレス、取得 (SEN = 1、AHEN = 0、DHEN = 0)

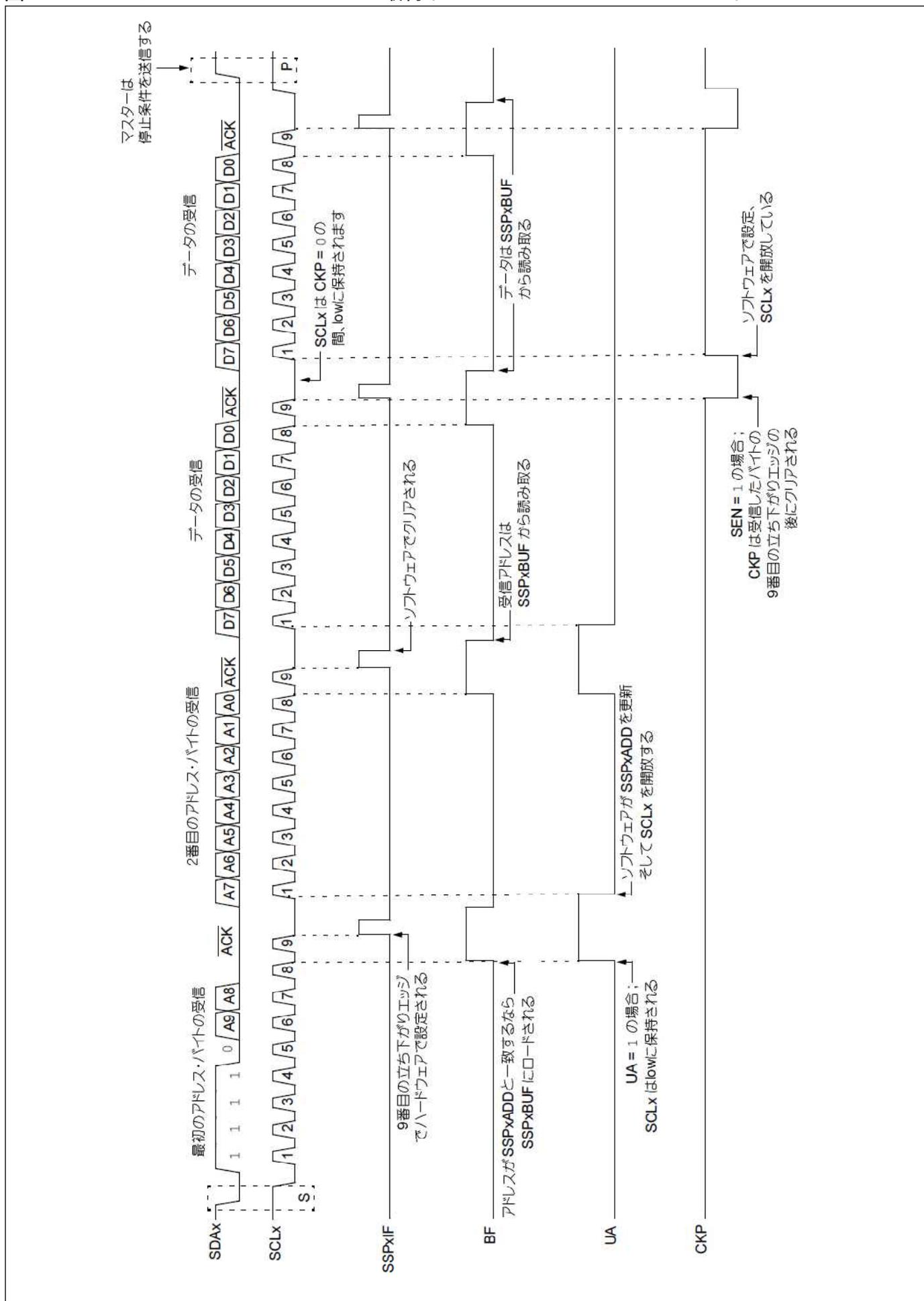
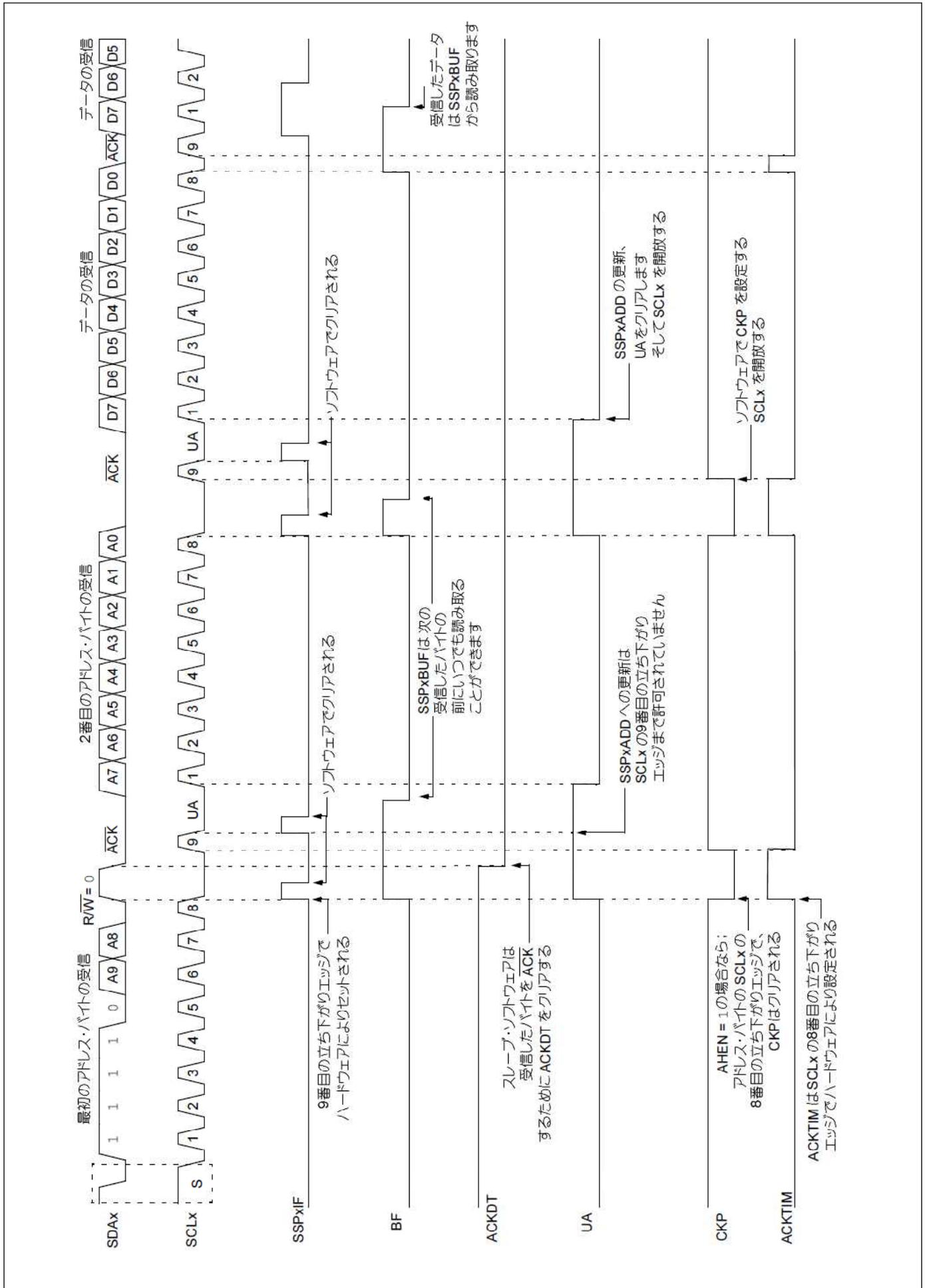
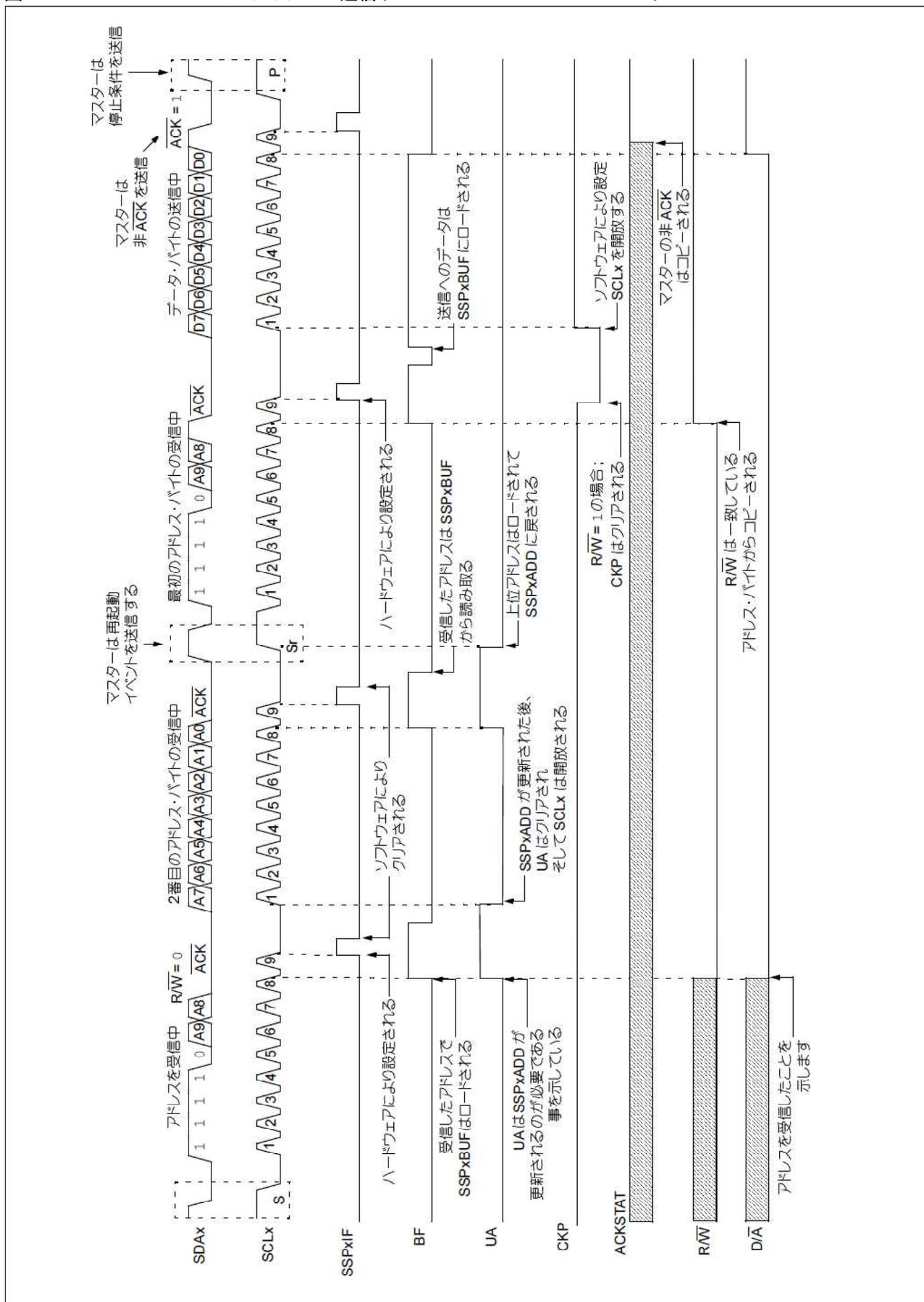


図21-21: I<sup>2</sup>Cスレーブ、10ビット・アドレス、取得 (SEN = 0, AHEN = 1, DHEN = 0)



# PIC16(L)F1508/9

図21-22: I<sup>2</sup>Cスレーブ、10ビット・アドレス、送信 (SEN = 0, AHEN = 0, DHEN = 0)



## 21.5.6 クロック延長

バス上のデバイスがSCLxラインをローに保持し、通信を効果的に一時停止すると、クロック・ストレッチ（延長）が発生します。スレーブはクロックを延ばして、データを処理したり、マスターデバイスの応答を準備したりするための時間を増やすことができます。マスター・デバイスは、バス上でアクティブであり、延長しているデータを転送しないため、延長には関与しません。スレーブによって行われた延長は、マスター・ソフトウェアからは見え、SCLxを生成するハードウェアによって処理されます。

SSPxCON1レジスタのCKPビットは、ソフトウェアの延長を制御するために使用されます。CKPビットがクリアされるたびに、モジュールはSCLxラインがローになるのを待ってからそれを保持します。CKPを設定すると、SCLxが解放され、より多くの通信が可能になります。

### 21.5.6.1 通常のクロック延長

SSPxSTATのR $\overline{W}$ ビットが設定されている場合の $\overline{ACK}$ 、読み取り要求に続いて、スレーブ・ハードウェアはCKPをクリアします。これにより、スレーブはマスターに転送するデータでSSPxBUFを更新できます。SSPxCON2のSENビットが設定されている場合、スレーブ・ハードウェアは常に $\overline{ACK}$ シーケンスの後にクロックを延長します。スレーブの準備が整うと、ソフトウェアによってCKPが設定され、通信が再開されます。

- 注1:** BFビットは、クロックが引き伸ばされるかどうかには影響しません。これは、SCLxの9番目の立ち下がりエッジの前にSSPxBUFが読み取られた場合に、クロックを引き伸ばさない以前のバージョンのモジュールとは異なり、CKPをクリアします。
- 2:** SSPxBUFがSCLxの9番目の立ち下がりエッジの前にロードされた場合、モジュールの以前のバージョンは送信のクロックを拡張しませんでした。読み取り要求に対して常にクリアされるようになりました。

### 21.5.6.2 10ビット・アドレッシング・モード

10ビット・アドレッシング・モードでは、UAビットがセットされると、クロックは常に延長されます。これは、CKPがクリアされずにSCLxが引き伸ばされる唯一の時間です。SCLxは、SSPxADDへの書き込み直後に解放されます。

**注釈:** モジュールの以前のバージョンは、2番目のアドレス・バイトが一致しなかった場合、クロックを延長しませんでした。

### 21.5.6.3 バイトNACKする

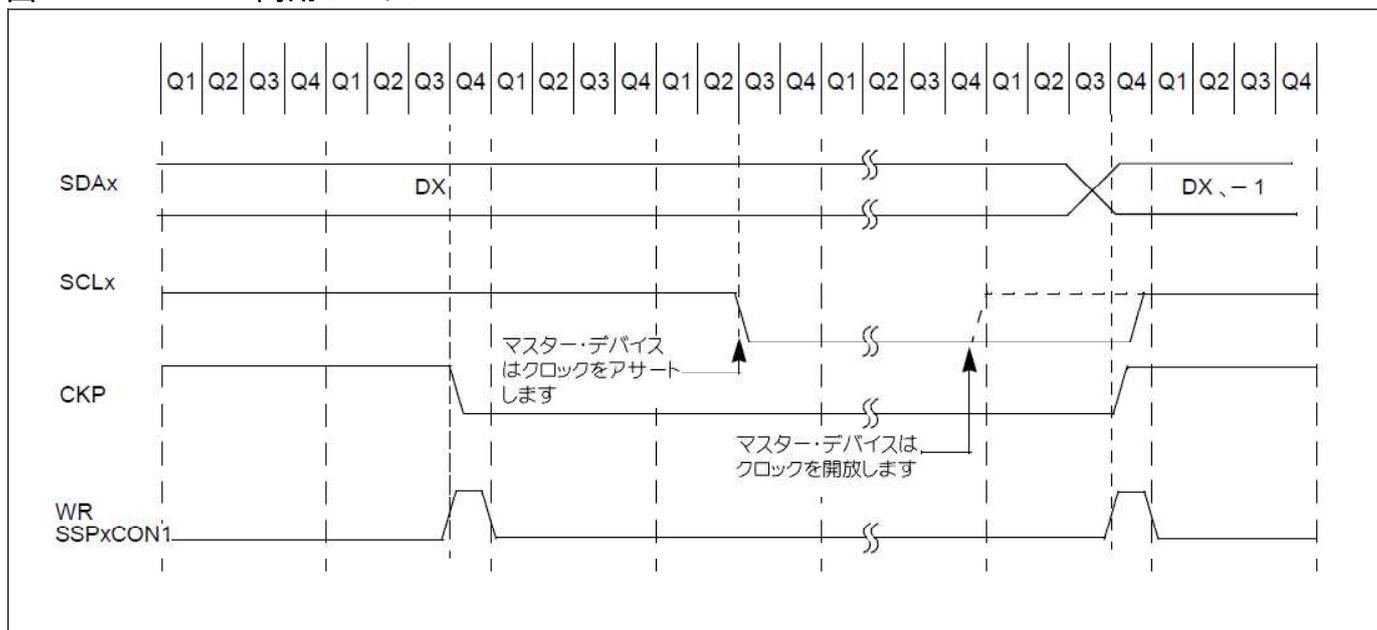
SSPxCON3のAHENビットが設定されている場合、CKPは、受信した一致するアドレス・バイトのSCLxの8番目の立ち下がりエッジの後にハードウェアによってクリアされます。SSPxCON3のDHENビットがセットされると、受信データのSCLxの8番目の立ち下がりエッジの後にCKPがクリアされます。

SCLxの8番目の立ち下がりエッジの後で延長すると、スレーブは受信したアドレスまたはデータを確認し、受信したデータをACKするかどうかを決定できます。

### 21.5.7 クロック同期とCKPビット

CKPビットがクリアされるたびに、モジュールはSCLxラインがLowになるのを待ってからそれを保持します。ただし、CKPビットをクリアしても、SCLx出力がすでにLowにサンプリングされるまで、SCLx出力はLowにアサート（論理有効）されません。したがって、外部I<sup>2</sup>CマスターデバイスがすでにSCLxラインをアサートするまで、CKPビットはSCLxラインをアサートしません。SCLx出力は、CKPビットがセットされ、I<sup>2</sup>Cバス上の他のすべてのデバイスがSCLxを解放するまでLowのままになります。これにより、CKPビットへの書き込みがSCLxの最小ハイ・タイム要件に違反しないことが保証されます（図21-23を参照）。

図21-23: クロック同期のタイミング



## 21.5.8 一般的な呼び出しアドレス・サポート

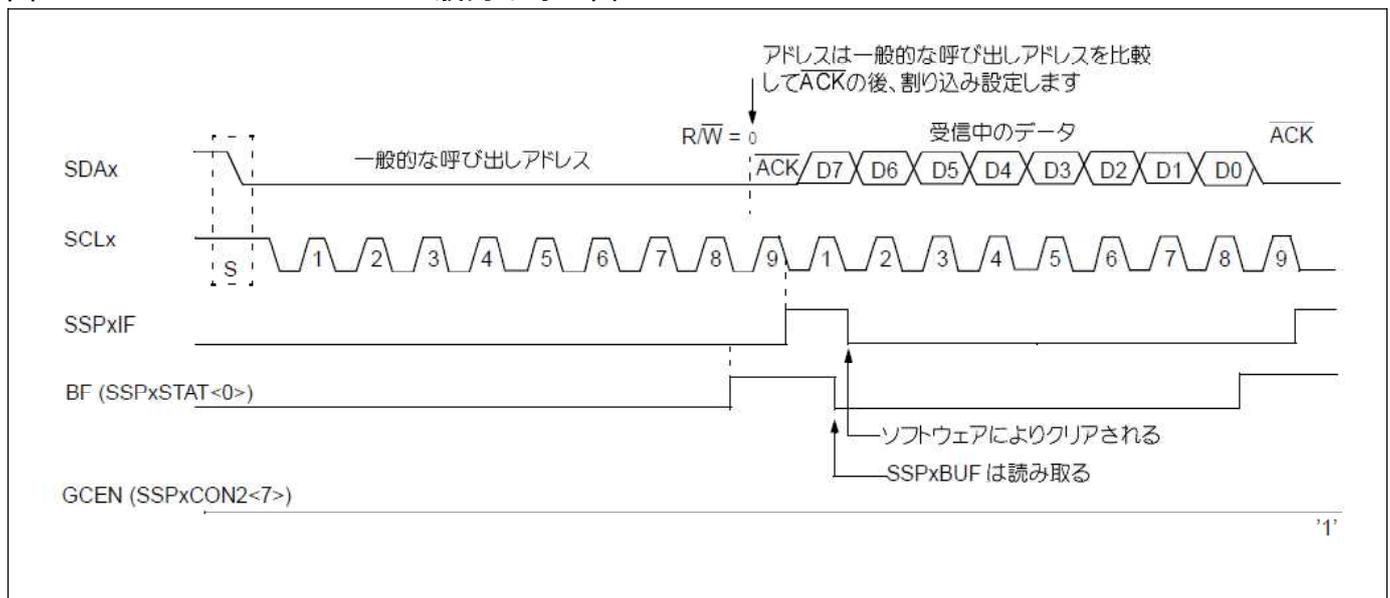
I<sup>2</sup>Cバスのアドレス指定手順では、通常、開始条件の後の最初のバイトによって、マスター・デバイスによってアドレス指定されるスレーブになるデバイスが決定されます。例外は、すべてのデバイスをアドレス指定できる一般的な呼び出しアドレスです。このアドレスを使用する場合、理論的には、すべてのデバイスが確認応答で応答する必要があります。

一般的な呼び出しアドレスは、アドレス0x00として定義されているI<sup>2</sup>Cプロトコルの予約済みアドレスです。SSPxCON2レジスタのGCENビットがセットされると、スレーブ・モジュールはSSPxADDに格納されている値に関係なく、このアドレスの受信を自動的にACKします。スレーブがR/Wビットをクリアしてすべてゼロのアドレスをクロックした後、割り込みが生成され、スレーブ・ソフトウェアはSSPxBUFを読み取って応答できます。図21-24に、一般的な呼び出しの受信シーケンスを示します。

10ビット・アドレス・モードでは、一般呼び出しアドレスの受信時にUAビットは設定されません。スレーブは、7ビット・モードの場合と同様に、2番目のバイトをデータとして受信する準備をします。

SSPxCON3レジスタのAHENビットが設定されている場合、他のアドレス受信と同様に、スレーブ・ハードウェアはSCLxの8番目の立ち下がりエッジの後にクロックを延長します。次に、スレーブはACKDT値を設定し、通常どおり通信が進行するようにクロックを解放する必要があります。

図21-24: スレーブ・モードの一般的な呼び出しアドレス・シーケンス



## 21.5.9 SSPxマスク・レジスタ

SSPxマスク (SSPxMSK) レジスタ (レジスタ21-5) は、アドレス比較操作中にSSPxSRレジスタに保持される値のマスクとしてI<sup>2</sup>Cスレーブ・モードで使用できます。SSPxMSKレジスタのゼロ (「0」) ビットは、受信したアドレスの対応するビットを「ドント・ケア (気にしない)」にする効果があります。

このレジスタは、リセット条件が発生するとすべて「1」にリセットされるため、マスク値で書き込まれるまで、標準のSSPx動作に影響を与えません。

SSPxマスク・レジスタは：

- 7ビット・アドレス・モード：A<7:1>のアドレス比較。
  - 10ビット・アドレス・モード：A<7:0>のみのアドレス比較。
- SSPxマスクは、アドレスの最初の (上位) バイトの受信中は効果がありません。

## 21.6 I<sup>2</sup>Cマスター・モード

マスター・モードは、SSPxCON1レジスターの適切なSSPMビットをセットおよびクリアし、SSPENビットをセットすることによって有効になります。マスター・モードでは、SDAxピンとSCKxピンを入力として構成する必要があります。MSSP周辺ハードウェアは、ピンをLowに駆動する必要がある場合、出カドライバートリス制御を上書きします。

マスター動作モードは、開始条件と停止条件の検出時の割り込み生成によってサポートされます。停止 (P) ビットと開始 (S) ビットは、リセットから、またはMSSPxモジュールが無効になるとクリアされます。I<sup>2</sup>Cバスの制御は、Pビットがセットされているとき、またはバスがアイドル (不動) 状態のときに行われる可能性があります。

ファームウェア制御マスター・モードでは、ユーザー・コードはスタート・ビットとストップ・ビットの状態検出に基づいてすべてのI<sup>2</sup>Cバス操作を実行します。開始および停止状態の検出は、このモードで唯一アクティブ (動的、活動的) な回路です。他のすべての通信は、SDAxおよびSCLx回線を直接操作するユーザー・ソフトウェアによって行われます。

次のイベントにより、SSPx割り込みフラグ・ビットSSPxIFが設定されます (有効な場合はSSPx割り込み) :

- 開始条件が検出されました
- 停止状態が検出されました
- 送受信されたデータ転送バイト
- 送信/受信を確認する
- 繰り返し開始が生成されました

**注1:** MSSPxモジュールは、I<sup>2</sup>Cマスター・モードで構成されている場合、イベントのキューイングを許可しません。たとえば、ユーザーは、開始条件を開始し、開始条件が完了する前に送信を開始するためにSSPxBUFレジスターにすぐに書き込むことは許可されていません。この場合、SSPxBUFへの書き込みは行われず、WCOLビットが設定され、SSPxBUFへの書き込みが行われなかったことを示します。

- 2:** マスター・モードの場合、SEN/PENビットがクリアされて生成が完了すると、開始/停止検出がマスクされ、割り込みが生成されます。

### 21.6.1 I<sup>2</sup>Cマスター・モード動作

マスター・デバイスは、すべてのシリアル・クロック・パルスと開始条件と停止条件を生成します。転送は、停止条件または繰り返し開始条件で終了します。繰り返し開始条件は次のシリアル転送の開始でもあるため、I<sup>2</sup>Cバスは解放されません。

マスター送信モードでは、シリアル・データはSDAxを介して出力され、SCLxはシリアル・クロックを出力します。送信される最初のバイトには、受信デバイスのスレーブ・アドレス (7ビット) と読み取り/書き込み (R/W) ビットが含まれます。この場合、R/Wビットは論理「0」になります。シリアル・データは一度に8ビット送信されます。各バイトが送信された後、確認応答ビットが受信されます。シリアル転送の開始と終了を示す開始条件と停止条件が出力されます。

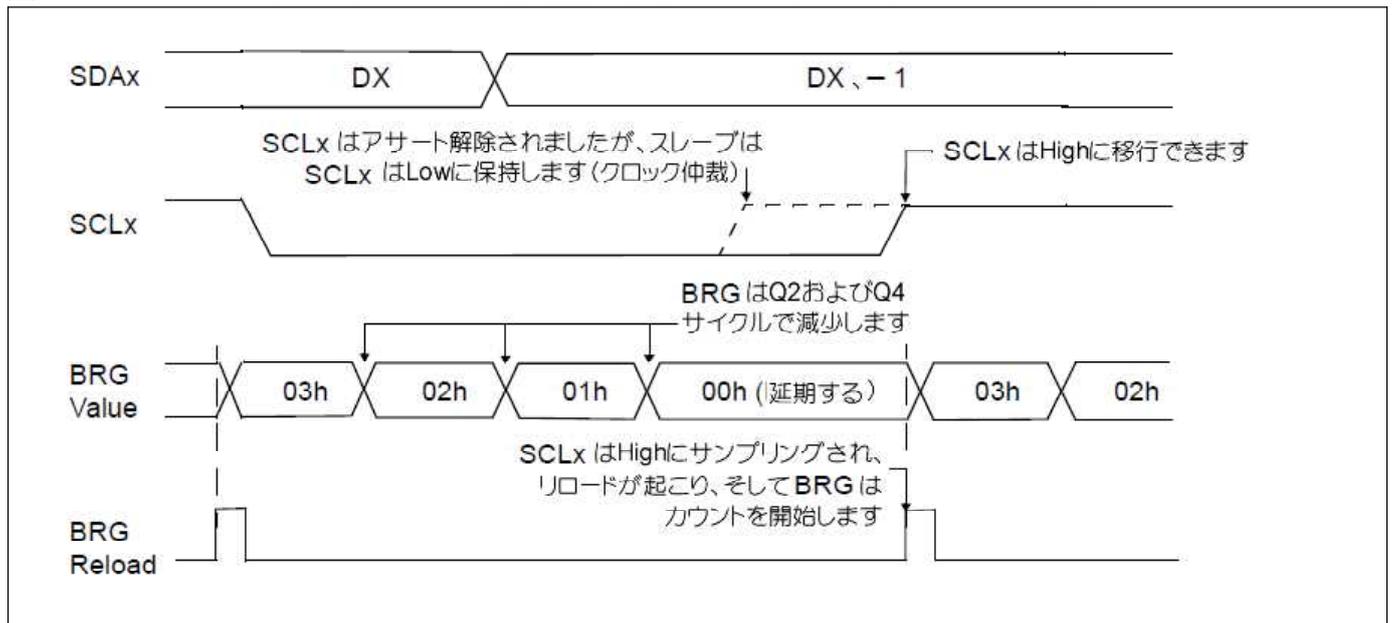
マスター受信モードでは、送信される最初のバイトには、送信デバイスのスレーブ・アドレス (7ビット) とR/Wビットが含まれます。この場合、R/Wビットは論理「1」になります。したがって、送信される最初のバイトは7ビットのスレーブ・アドレスであり、その後受信ビットを示す「1」が続きます。シリアルデータはSDAxを介して受信され、SCLxはシリアル・クロックを出力します。シリアル・データは一度に8ビット受信されます。各バイトが受信された後、確認応答ビットが送信されます。開始条件と停止条件は、送信の開始と終了を示します。

ボーレート・ジェネレータは、SCLxのクロック周波数出力を設定するために使用されます。詳細については、セクション21.7「ボーレート・ジェネレータ」を参照してください。

## 21.6.2 クロック仲裁

クロック・アービトレーション（仲裁）は、マスターが受信、送信、または繰り返し開始/停止状態の間にSCLxピンを解放したときに発生します（SCLxはHighにフロートできます）。SCLxピンがHighにフロートすることが許可されると、ポーレート・ジェネレーター（BRG）は、SCLxピンが実際にHighにサンプリングされるまでカウントを一時停止します。SCLxピンがHighにサンプリングされると、ポーレート・ジェネレーターにSSPxADD<7:0>の内容がリロードされ、カウントが開始されます。これにより、外部デバイスによってクロックがLowに保持された場合に、SCLxハイ・タイムが常に少なくとも1つのBRGロールオーバー・カウントになることが保証されます（図21-25）。

図21-25: クロック・アービトレーション（仲裁）によるポーレート・ジェネレーターのタイミング



## 21.6.3 WCOLステータス・フラグ

開始、再起動、停止、受信、または送信シーケンスの進行中にユーザーがSSPxBUFに書き込むと、WCOLビットが設定され、バッファの内容は変更されません（書き込みは発生しません）。WCOLビットが設定されている場合は常に、モジュールがアイドル（不動）状態でないときにSSPxBUFでアクションが試行されたことを示します。

**注釈:** イベントのキューイングは許可されていないため、SSPxCON2の下位5ビットへの書き込みは、開始条件が完了するまで無効になります。

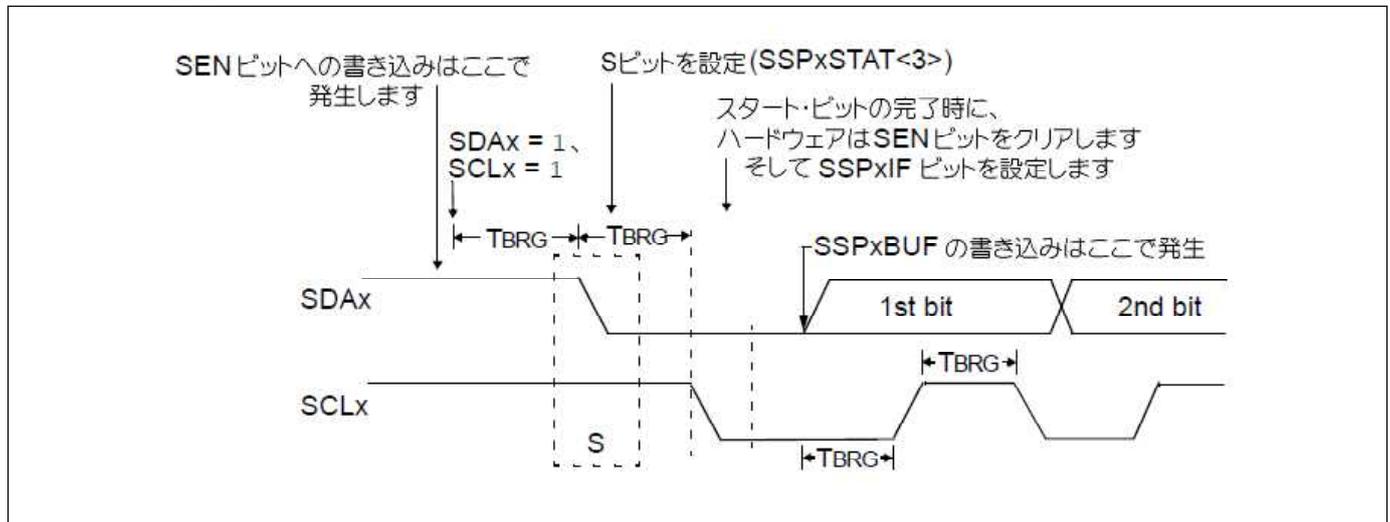
21.6.4 I<sup>2</sup>Cマスター・モード開始条件タイミング

開始条件（図21-26）を開始するには、ユーザーはSSPxCON2レジスタの開始許可ビットであるSENビットを設定します。SDAxピンとSCLxピンがHighにサンプリングされると、ポーレート・ジェネレーターにSSPxADD<7:0>の内容が再ロードされ、カウントが開始されます。ポーレート・ジェネレーターがタイムアウトしたときにSCLxとSDAxの両方がHighにサンプリングされると（TBRG）、SDAxピンはLowに

駆動されます。SCLxがHighのときにSDAxがLowに駆動される動作が開始条件であり、SSPxSTAT1レジスタのSビットがセットされます。これに続いて、ポーレート・ジェネレーターにSSPxADD<7:0>の内容が再ロードされ、カウントが再開されます。ポーレート・ジェネレーターがタイムアウト（TBRG）すると、SSPxCON2レジスタのSENビットがハードウェアによって自動的にクリアされます。ポーレート・ジェネレーターは一時停止され、SDAxラインはLowのままになり、開始条件が完了します。

- 注1：** 開始条件の開始時にSDAxピンとSCLxピンがすでにLowにサンプリングされている場合、または開始条件中にSDAxラインがLowに駆動される前にSCLxラインがLowにサンプリングされている場合、バスの衝突が発生し、バス衝突割り込みフラグBCLxIFが設定され、開始条件が中止され、I<sup>2</sup>Cモジュールがアイドル（不動）状態にリセットされます。
- 2：** Philips I<sup>2</sup>C仕様では、スタート時にバスの衝突は発生しないと規定されています。

図21-26: 最初のスタート・ビット・タイミング



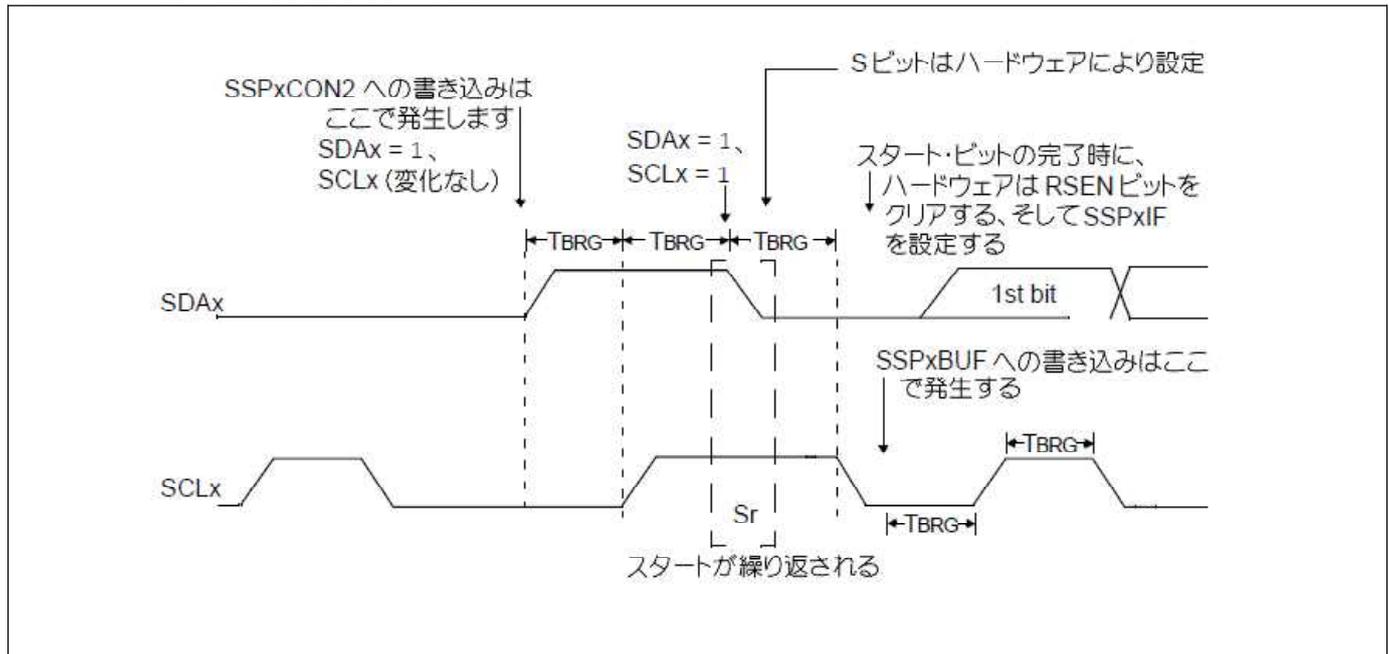
## 21.6.5 I<sup>2</sup>Cマスター・モードの繰り返し開始条件のタイミング

繰り返し開始条件（図21-27）は、SSPxCON2レジスタのRSENビットがHighにプログラムされ、マスター・ステート・マシンがアクティブでなくなったときに発生します。RSENビットがセットされると、SCLxピンはLowにアサート（論理有効）されます。SCLxピンがLowにサンプリングされると、ポーレート・ジェネレーターがロードされ、カウントが開始されます。1つのポーレート・ジェネレーター・カウント（TBRG）に対して、SDAxピンが解放されます（Highになります）。ポーレート・ジェネレーターがタイムアウトしたときに、SDAxがHighにサンプリングされると、SCLxピンがディアサート

（アサート解除）されます（Highになります）。SCLxがHighにサンプリングされると、ポーレート・ジェネレーターがリロードされ、カウントが開始されます。SDAxとSCLxは、1つのTBRGに対してHighにサンプリングする必要があります。このアクションの後に、SCLxがHighのときに1つのTBRGに対してSDAxピン（SDAx = 0）がアサート（論理有効）されます。SCLxはLowにアサートされます。これに続いて、SSPxCON2レジスタのRSENビットが自動的にクリアされ、ポーレート・ジェネレーターはリロードされず、SDAxピンはLowのままになります。SDAxピンとSCLxピンで開始条件が検出されるとすぐに、SSPxSTATレジスタのSビットがセットされます。SSPxIFビットは、ポーレート・ジェネレーターがタイムアウトするまで設定されません。

- 注1:** 他のイベントの進行中にRSENがプログラムされている場合、RSENは有効になりません。
- 2:** 繰り返し開始状態でのバスの衝突は、次の場合に発生します：
- SCLxがLowからHighになると、SDAxはLowにサンプリングされます。
  - SDAxがLowにアサート（論理有効）される前にSCLxがLowになります。これは、別のマスターがデータ「1」を送信しようとしていることを示している可能性があります。

図21-27: 繰り返し開始条件の波形



### 21.6.6 I<sup>2</sup>Cマスター・モード送信

データ・バイト、7ビット・アドレス、または10ビット・アドレスの残りの半分の送信は、SSPxBUFレジスターに値を書き込むだけで実行されます。このアクションにより、バッファフル・フラグ・ビットBFが設定され、ポーレート・ジェネレーターがカウントを開始して次の送信を開始できるようになります。SCLxの立ち下がりエッジがアサート（論理有効）された後、アドレス/データの各ビットがSDAxピンにシフトアウトされます。SCLxは、1つのポーレート・ジェネレーター・ロールオーバー・カウント（TBRG）の間Lowに保持されます。SCLxがHighに開放される前に、データが有効である必要があります。SCLxピンがHighに解放されると、TBRGに対してそのように保持されます。SDAxピンのデータは、SCLxの次の立ち下がりエッジの後、その期間とある程度のホールド・タイムの間安定している必要があります。8番目のビットがシフトアウトされた後（8番目のクロックの立ち下がりエッジ）、BFフラグがクリアされ、マスターはSDAxを解放します。これにより、アドレスの一致が発生した場合、またはデータが適切に受信された場合、アドレス指定されているスレーブ・デバイスが9番目のビット時間中にACKビットで応答できるようになります。ACKのステータスは、9番目のクロックの立ち上がりエッジでACKSTATビットに書き込まれます。マスターが確認応答を受信すると、確認応答ステータス・ビットACKSTATがクリアされます。そうでない場合、ビットがセットされます。9番目のクロックの後、SSPxIFビットがセットされ、マスター・クロック（ポーレート・ジェネレーター）は次のデータ・バイトがSSPxBUFにロードされるまで中断され、SCLxはLowのままSDAxは変更されません（図21-28）。

SSPxBUFへの書き込み後、アドレスの各ビットは、7つのアドレス・ビットすべてでR/Wビットが完了するまで、SCLxの立ち下がりエッジでシフトアウトされます。8番目のクロックの立ち下がりエッジで、マスターはSDAxピンを解放し、スレーブが確認応答で応答できるようにします。9番目のクロックの立ち下がりエッジで、マスターはSDAxピンをサンプリングして、アドレスがスレーブによって認識されたかどうかを確認します。ACKビットのステータスは、SSPxCON2レジスターのACKSTATステータス・ビットにロードされます。アドレスの9番目のクロック送信の立ち下がりエッジに続いて、SSPxIFが設定され、BFフラグがクリアされ、SSPxBUFへの別の書き込みが行われるまでポーレート・ジェネレータがオフになり、SCLxをLowに保持してSDAxをフロートさせます。

#### 21.6.6.1 BFステータス・フラグ

送信モードでは、CPUがSSPxBUFに書き込むときにSSPxSTATレジスターのBFビットがセットされ、8ビットすべてがシフトアウトされるとクリアされます。

#### 21.6.6.2 WCOLステータス・フラグ

送信がすでに進行中のときにユーザーがSSPxBUFに書き込む場合（つまり、SSPxSRがまだデータ・バイトをシフトアウトしている場合）、WCOLビットが設定され、バッファの内容は変更されません（書き込みは発生しません）。

WCOLは、次の送信の前にソフトウェアによってクリアされる必要があります。

### 21.6.6.3 ACKSTATステータス・フラグ

送信モードでは、SSPxCON2レジスターのACKSTATビットは、スレーブが確認応答を送信したときにクリアされ（ACK = 0）、スレーブが確認応答を送信しなかったときにセットされます（ACK = 1）。スレーブは、アドレス（一般呼び出しを含む）を認識したとき、またはスレーブがデータを適切に受信したときに、確認応答を送信します。

#### 21.6.6.4 典型的な送信シーケンス:

1. ユーザーは、SSPxCON2レジスターのSENビットをセットして開始条件を生成します。
2. SSPxIFは、開始の完了時にハードウェアによって設定されます。
3. SSPxIFはソフトウェアによってクリアされます。
4. MSSPxモジュールは、他の操作が行われる前に、必要な開始時間を待機します。
5. ユーザーは、送信するスレーブ・アドレスをSSPxBUFにロードします。
6. アドレスは、8ビットすべてが送信されるまでSDAxピンからシフトアウトされます。SSPxBUFが書き込まれるとすぐに送信が開始されます。
7. MSSPxモジュールは、スレーブ・デバイスからACKビットをシフトインし、その値をSSPxCON2レジスタのACKSTATビットに書き込みます。
8. MSSPxモジュールは、SSPxIFビットをセットすることにより、9番目のクロック・サイクルの終わりに割り込みを生成します。
9. ユーザーはSSPxBUFに8ビットのデータをロードします。
10. データは、8ビットすべてが送信されるまでSDAxピンからシフトアウトされます。
11. MSSPxモジュールは、スレーブ・デバイスからACKビットをシフトインし、その値をSSPxCON2レジスターのACKSTATビットに書き込みます。
12. 送信されたすべてのデータ・バイトに対して、手順8~11が繰り返されます。
13. ユーザーは、SSPxCON2レジスターのPENまたはRSENビットを設定することにより、停止または再起動条件を生成します。停止/再起動条件が完了すると、割り込みが生成されます。



### 21.6.7 I<sup>2</sup>Cマスター・モード受信

マスター・モードの受信（図21-29）は、SSPxCON2レジスタの受信許可ビットであるRCENビットをプログラミングすることで有効になります。

**注釈：** RCENビットが設定される前にMSSPxモジュールがアイドル（不動）状態である必要があります。そうでない場合、RCENビットは無視されます。

ポーレート・ジェネレーターはカウントを開始し、ロールオーバーごとにSCLxピンの状態が変化し（HighからLow/LowからHigh）、データがSSPxSRにシフトされます。8番目のクロックの立ち上がりエッジの後、受信許可フラグが自動的にクリアされ、SSPxSRの内容がSSPxBUFにロードされ、BFフラグ・ビットが設定され、SSPxIFフラグ・ビットが設定され、ポーレート・ジェネレーターのカウントが一時停止され、SCLxをLowに保持します。MSSPはアイドル（不動）状態になり、次のコマンドを待機しています。CPUがバッファーを読み取ると、BFフラグ・ビットが自動的にクリアされます。次に、ユーザーは、SSPxCON2レジスタのアクノーリッジ・シーケンス・イネーブルACKENビットを設定することにより、受信の最後にアクノーリッジ（確認応答）ビットを送信できます。

#### 21.6.7.1 BFステータス・フラグ

受信動作では、アドレスまたはデータ・バイトがSSPxSRからSSPxBUFにロードされるときにBFビットがセットされ、SSPxBUFレジスタが読み取られるとクリアされます。

#### 21.6.7.2 SSPOVステータス・フラグ

受信動作では、8ビットがSSPxSRに受信され、BFフラグ・ビットが前の受信からすでに設定されている場合、SSPOVビットが設定されます。

#### 21.6.7.3 WCOLステータス・フラグ

受信がすでに進行中のときにユーザーがSSPxBUFに書き込む場合（つまり、SSPxSRがまだデータ・バイトでシフトしている場合）、WCOLビットが設定され、バッファーの内容は変更されません（書き込みは発生しません）。

#### 21.6.7.4 典型的な受信シーケンス:

1. ユーザーは、SSPxCON2レジスタのSENビットをセットして開始条件を生成します。
2. SSPxIFは、開始の完了時にハードウェアによって設定されます。
3. SSPxIFはソフトウェアによってクリアされます。
4. ユーザーは、送信するスレーブ・アドレスとR/Wビットを設定してSSPxBUFに書き込みます。
5. 8ビットすべてが送信されるまで、アドレスはSDAxピンからシフトアウトされます。SSPxBUFが書き込まれるとすぐに送信が開始されます。
6. MSSPモジュールは、スレーブ・デバイスからACKビットをシフトインし、その値をSSPxCON2レジスタのACKSTATビットに書き込みます。
7. MSSPモジュールは、SSPxIFビットをセットすることにより、9番目のクロック・サイクルの終わりに割り込みを生成します。
8. ユーザーは、SSPxCON2レジスタのRCENビットとマスター・クロックをスレーブからのバイトに設定します。
9. SCLxの8番目の立ち上がりエッジの後、SSPxIFとBFが設定されます。
10. マスターはSSPxIFをクリアし、受信したバイトをSSPxBUFから読み取り、BFをクリアします。
11. マスターは、スレーブに送信されるACK値をSSPxCON2レジスタのACKDTビットに設定し、ACKENビットをセットしてACKを開始します。
12. マスターACKがスレーブにクロックアウトされ、SSPxIFが設定されます。
13. ユーザーがSSPxIFをクリアします。
14. ステップ8~13は、スレーブから受信したバイトごとに繰り返されます。
15. マスターは非ACKまたは停止を送信して通信を終了します。



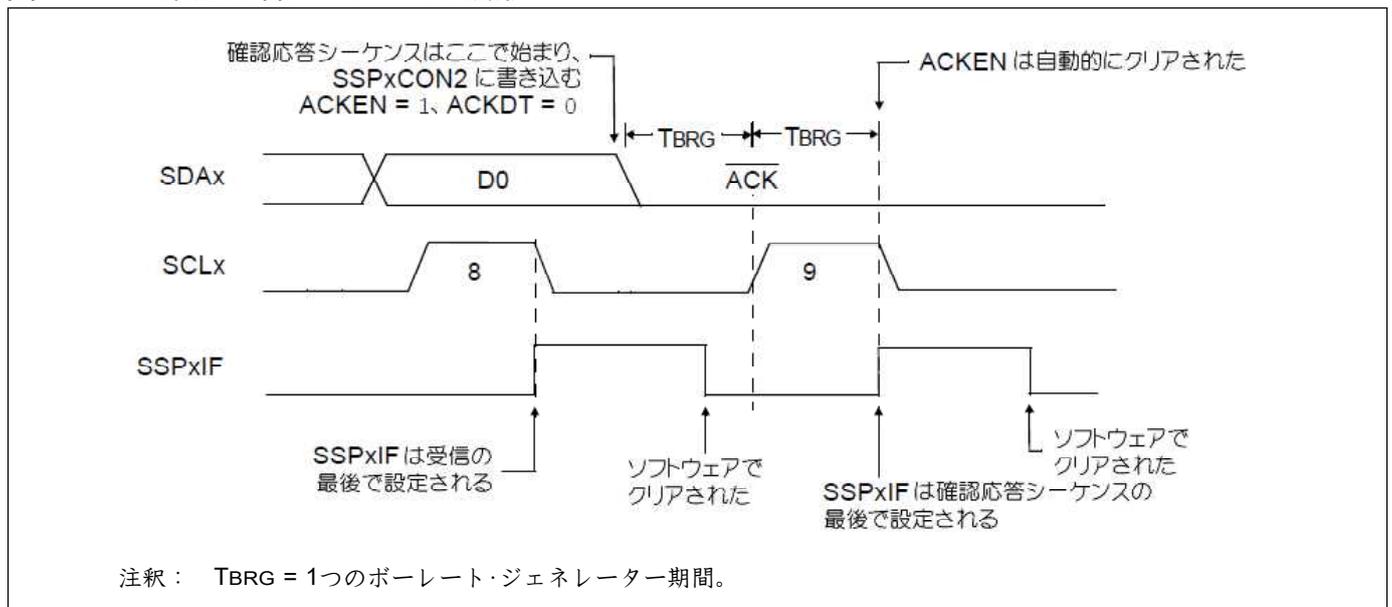
21.6.8 確認応答シーケンス・タイミング

確認応答シーケンスは、SSPxCON2レジスタの確認応答シーケンス有効化ビット、ACKENビットを設定することによって有効になります。このビットがセットされると、SCLxピンがLowにプル（引く）され、確認データ・ビットの内容がSDAxピンに表示されます。ユーザーが確認応答を生成したい場合は、ACKDTビットをクリアする必要があります。そうでない場合、ユーザーは確認応答シーケンスを開始する前にACKDTビットを設定する必要があります。次に、ポーレート・ジェネレーターは1ロールオーバー周期（TBRG）をカウントし、SCLxピンはディアサート（アサート解除）されます（Highにプルされます）。SCLxピンがHighにサンプリングされると（クロック仲裁）、ポーレート・ジェネレーターはTBRGをカウントします。次に、SCLxピンがLowに引き下げられます。これに続いて、ACKENビットが自動的にクリアされ、ポーレート・ジェネレーターがオフになり、MSSPモジュールがアイドル（不動）モードになります（図21-30）。

21.6.8.1 WCOLステータス・フラグ

確認応答シーケンスの進行中にユーザーがSSPxBUFに書き込むと、WCOLビットが設定され、バッファの内容は変更されません（書き込みは発生しません）。

図21-30: 確認応答シーケンスの波形



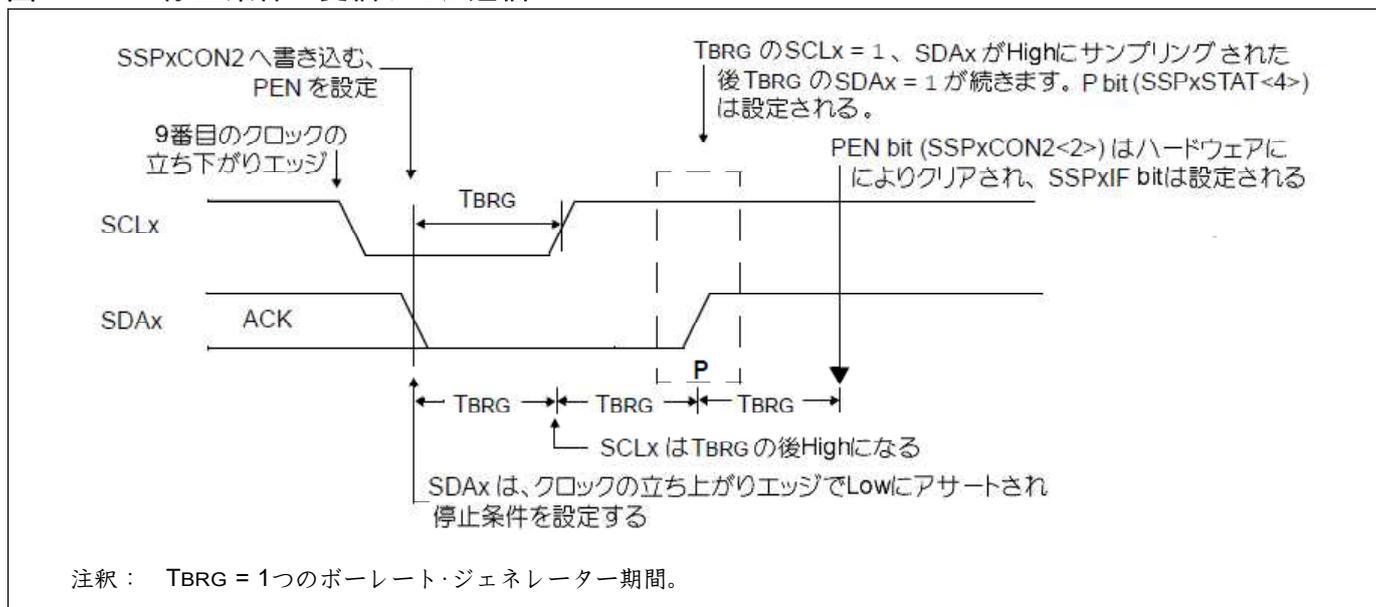
21.6.9 停止条件タイミング

ストップ・ビットは、SSPxCON2レジスタのストップ（停止）シーケンス許可ビット、PENビットをセットすることにより、受信/送信の最後にSDAxピンにアサート（論理有効）されます。受信/送信の終了時に、SCLxラインは9番目のクロックの立ち下がりエッジの後でLowに保持されます。PENビットがセットされると、マスターはSDAxラインをLowにアサートします。SDAxラインがLowにサンプリングされると、ポーレート・ジェネレーターがリロードされ、「0」までカウント・ダウンされます。ポーレート・ジェネレーターがタイムアウトすると、SCLxピンがHighになり、1 TBRG（ポーレート・ジェネレーターのロールオーバー・カウント）後にSDAxピンがディアサート（アサート解除）されます。SCLxがHighのときにSDAxピンがHighにサンプリングされると、SSPxSTATレジスタのPビットがセットされます。TBRGの後、PENビットがクリアされ、SSPxIFビットがセットされます（図21-31）。

21.6.9.1 WCOLステータス・フラグ

停止シーケンスの進行中にユーザーがSSPxBUFに書き込むと、WCOLビットが設定され、バッファの内容は変更されません（書き込みは発生しません）。

図21-31: 停止条件の受信または送信モード



## 21.6.10 スリープ動作

スリープ・モードでは、I<sup>2</sup>Cスリープ・モジュールはアドレスまたはデータを受信でき、アドレスの一致または完全なバイト転送が発生すると、プロセッサをスリープから復帰させます (MSSP割り込みが有効になっている場合)。

## 21.6.11 リセットの影響

リセットはMSSPモジュールを無効にし、現在の転送を終了します。

## 21.6.12 マルチ・マスター・モード

マルチ・マスター・モードでは、開始条件と停止条件の検出時に割り込みを生成することで、バスがいつ空いているかを判断できます。停止 (P) ビットと開始 (S) ビットは、リセットから、またはMSSPモジュールが無効になるとクリアされます。I<sup>2</sup>Cバスの制御は、SSPxSTATレジスタのPビットがセットされている場合、またはバスがアイドル状態で、SビットとPビットの両方がクリアされている場合に実行できます。バスがビジー (忙しい) の場合、SSP割り込みを有効にすると、停止条件が発生したときに割り込みが生成されます。

マルチ・マスター・モードでは、SDAxラインの仲裁を監視して、信号レベルが期待される出力レベルであるかどうかを確認する必要があります。このチェックはハードウェアによって実行され、結果はBCLxIFビットに配置されます。

仲裁が失われる可能性のあるステートは:

- アドレス転送
- データ転送
- 開始条件
- 繰り返し開始条件
- 確認応答条件

## 21.6.13 マルチ・マスター通信、バス衝突およびバス仲裁

マルチ・マスター・モードのサポートは、バス仲裁によって実現されます。マスターがアドレス/データ・ビットをSDAxピンに出力すると、マスターがSDAxに「1」を出力するときにSDAxをHighにフロートさせ、別のマスターが「0」をアサート (論理有効) することにより、仲裁が行われます。SCLxピンがHighにフロートすると、データは安定しているはずで、SDAxで予想されるデータが「1」で、SDAxピンでサンプリングされたデータが「0」の場合、バスの衝突が発生しています。マスターはバス衝突割り込みフラグBCLxIFを設定し、I<sup>2</sup>Cポートをアイドル (不動) 状態にリセットします (図21-32)。

バスの衝突が発生したときに送信が進行中であった場合、送信は停止され、BFフラグがクリアされ、SDAxおよびSCLxラインがディアサート (アサート解除) され、SSPxBUFに書き込むことができます。ユーザーがバス衝突割り込みサービス・ルーチンを処理し、I<sup>2</sup>Cバスが空いている場合、ユーザーは開始条件をアサートすることで通信を再開できます。

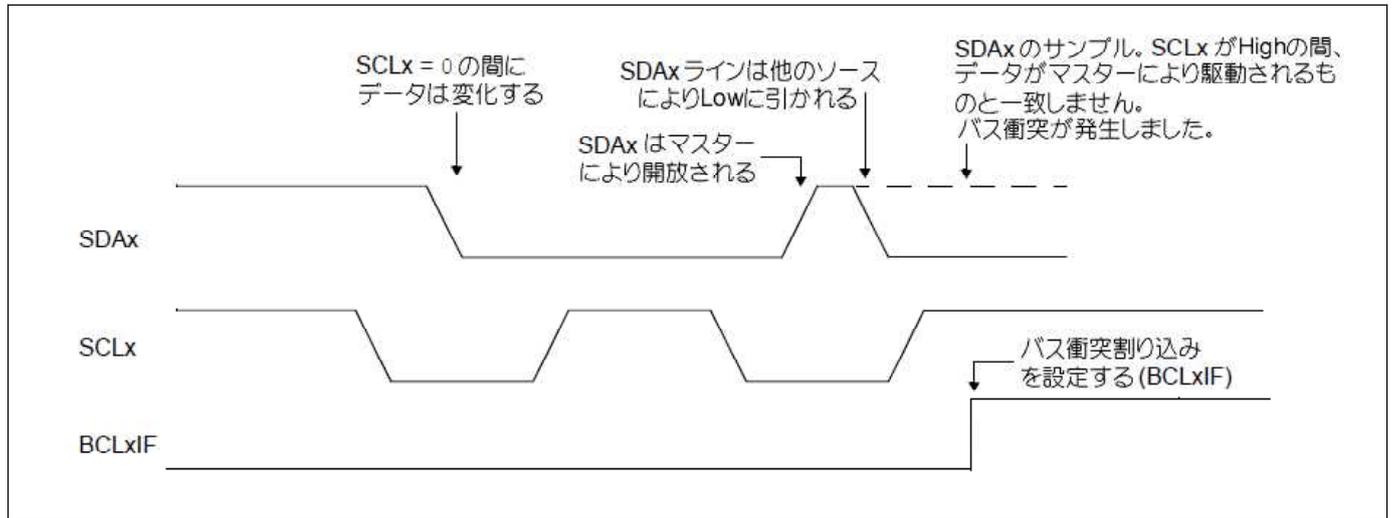
バス衝突が発生したときに開始、繰り返し開始、停止、または確認応答条件が進行中だった場合、条件は中止され、SDAxおよびSCLxラインがディアサート (アサート解除) され、SSPxCON2レジスタのそれぞれの制御ビットがクリアされます。ユーザーがバス衝突割り込みサービス・ルーチンを処理し、I<sup>2</sup>Cバスが空いている場合、ユーザーは開始条件をアサートすることで通信を再開できます。

マスターは引き続きSDAxピンとSCLxピンを監視します。停止条件が発生すると、SSPxIFビットがセットされます。

SSPxBUFへの書き込みは、バスの衝突が発生したときに送信器が中断した場所に関係なく、最初のデータ・ビットでデータの送信を開始します。

マルチ・マスター・モードでは、開始条件と停止条件の検出時に割り込みを生成することで、バスがいつ空いているかを判断できます。I<sup>2</sup>Cバスの制御は、SSPxSTATレジスタにPビットが設定されている場合、またはバスがアイドル状態でSビットとPビットがクリアされている場合に実行できます。

図21-32: 送信と確認応答のためのバス衝突のタイミング



## 21.6.13.1 始動状態の中でのバスの衝突

開始状態の間で、次の場合にバスの衝突が発生します：

- a) SDAまたはSCLは、開始条件の開始時にLowでサンプリングされます（図21-33）。
- b) SDAがLowにアサート（論理有効）される前に、SCLがLowにサンプリングされます（図21-34）。

開始状態の間、SDAxピンとSCLピンの両方が監視されます。

SDAピンがすでにLowになっている場合、またはSCLピンがすでにLowになっている場合は、次のすべてが発生します：

- 開始条件が中止され、
- BCL1IFフラグが設定され、
- MSSPモジュールはアイドル（不動）状態にリセットされます（Figure 21-33）。

開始条件は、SDAxピンとSCLxピンがディアサート（アサート解除）された状態で始まります。SDAxピンがHighにサンプリングされると、ボーレート・ジェネレーターがロードされ、カウント・ダウンされます。SDAxがHighのときにSCLxピンがLowにサンプリングされると、別のマスターが開始状態でデータ「1」を駆動しようとしていると想定されるため、バスの衝突が発生します。

このカウント中にSDAxピンがLowにサンプリングされると、BRGがリセットされ、SDAxラインが早期にアサート（論理有効）されます（図21-35）。ただし、SDAピンで「1」がサンプリングされた場合、BRGカウントの最後にSDAピンがLowにアサートされます。次に、ボーレート・ジェネレーターがリロードされ、ゼロまでカウントダウンされます。この間にSCLピンが「0」としてサンプリングされた場合、バスの衝突は発生しません。BRGカウントの最後に、SCLピンがLowにアサートされます。

注釈： 開始条件でバスの衝突が要因にならない理由は、2つのバス・マスターがまったく同時に開始条件をアサートできないためです。したがって、一方のマスターは常にもう一方のマスターよりも先にSDAxをアサートします。2つのマスターが開始条件に続く最初のアドレスを調停することを許可されなければならないので、この条件はバス衝突を引き起こしません。アドレスが同じである場合、調停はデータ部分、繰り返し開始または停止条件に続くことを許可されなければなりません。

図21-33: 開始条件中のバス衝突(SDAXのみ)

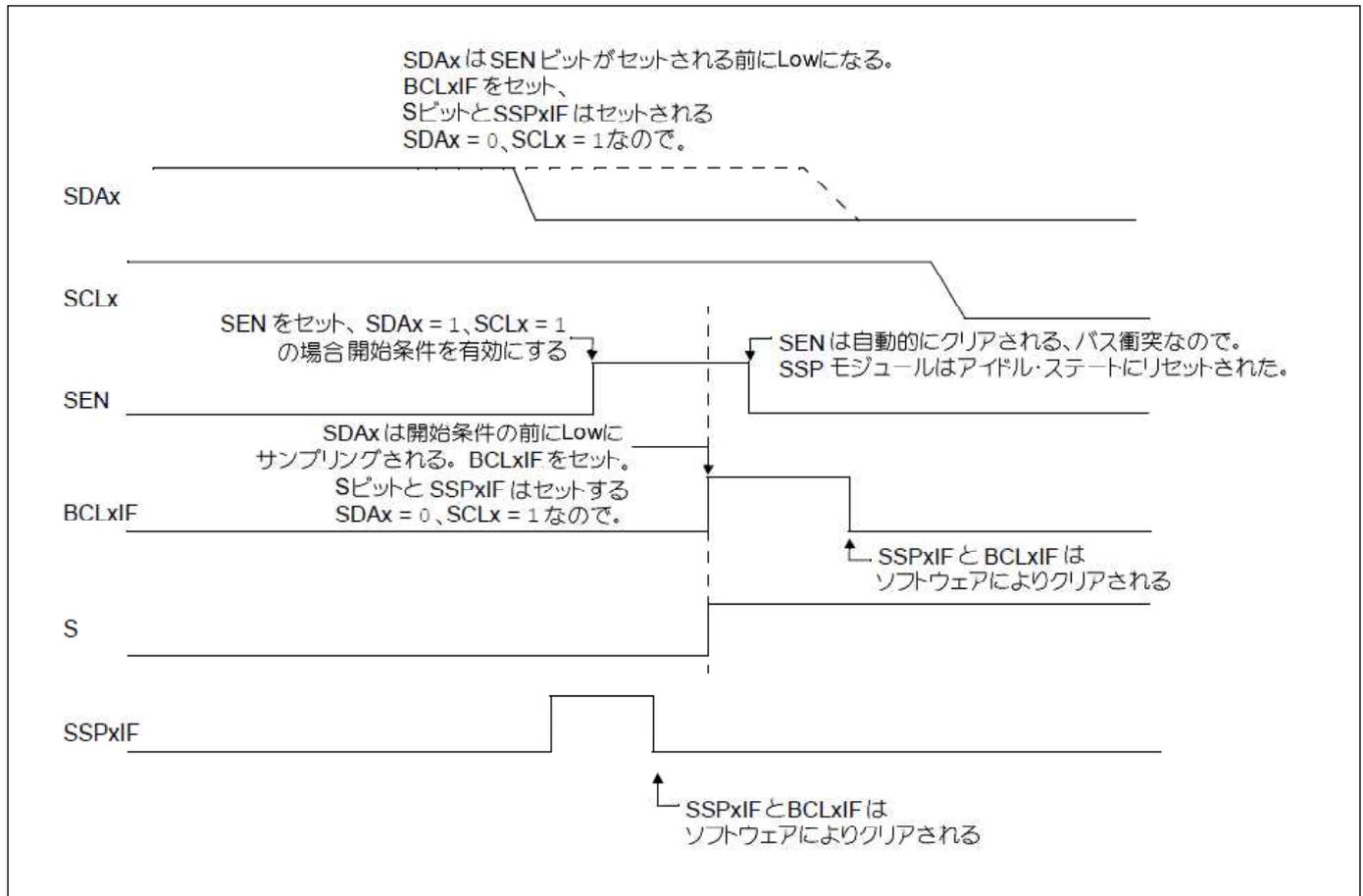


図21-34: 開始条件中のバス衝突 (SCLx = 0)

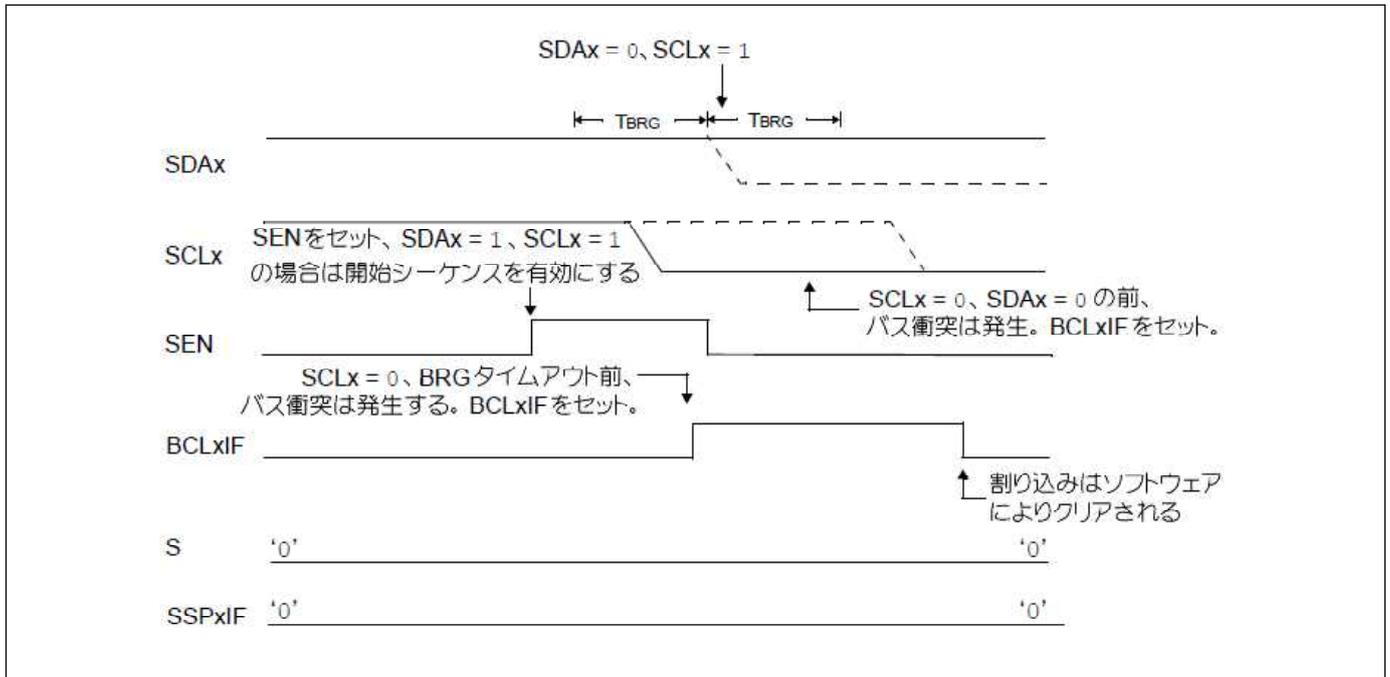
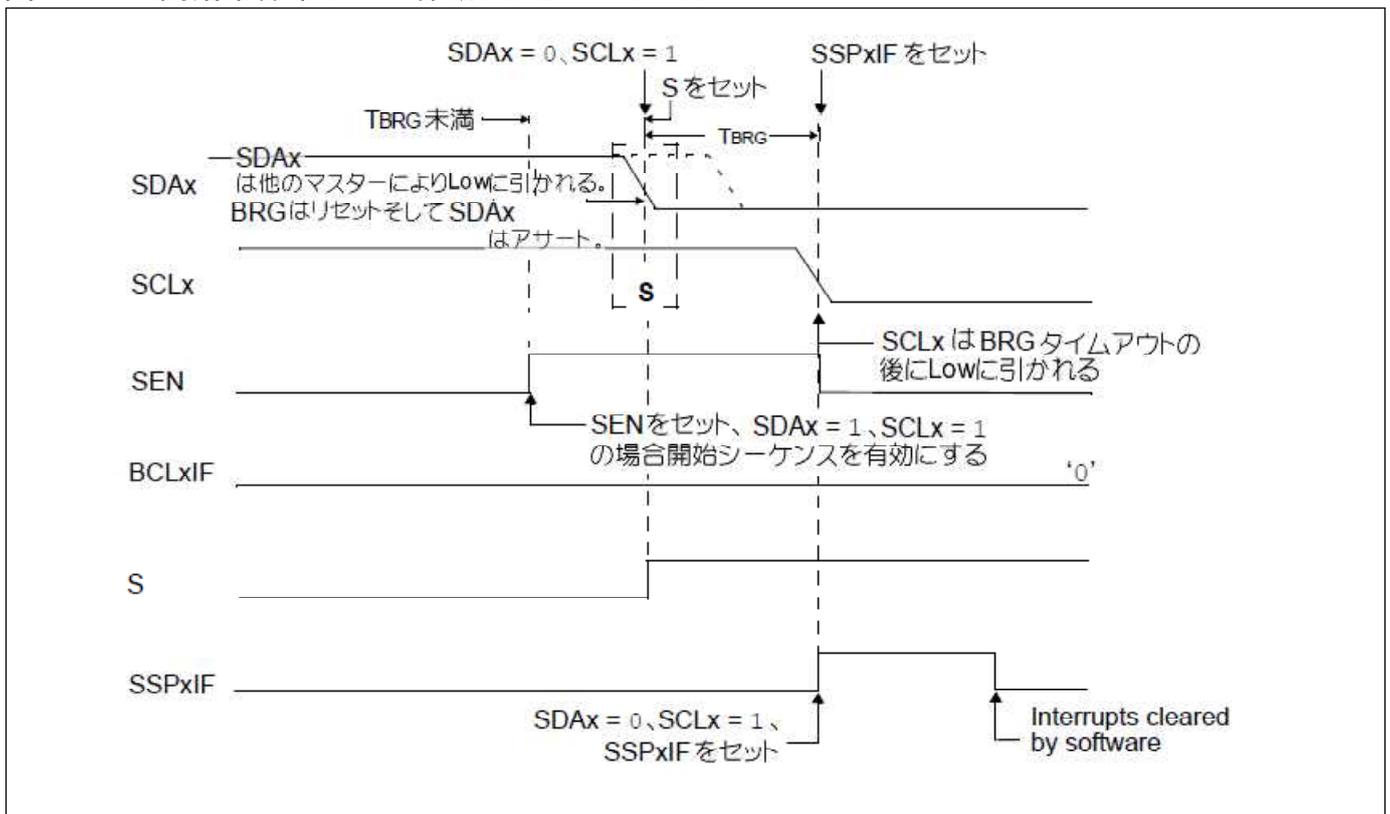


図21-35: 開始条件中のSDA仲裁によるBRGリセット



## 21.6.13.2 繰り返し開始状態でのバス衝突

繰り返し開始状態で、次の場合にバスの衝突が発生します：

- a) SCLxが低レベルから高レベルに移行すると、SDAxで低レベルがサンプリングされます（ケース1）。
- b) SDAxがLowにアサート（論理有効）される前にSCLxがLowになり、別のマスターがデータ「1」を送信しようとしていることを示します（ケース2）。

ユーザーがSDAxを解放し、ピンがHighにフロートできるようになると、BRGにSSPxADDがロードされ、ゼロまでカウントダウンします。次に、SCLxピンがディアサート（アサート解除）され、Highにサンプリングされると、SDAxピンがサンプリングされます。

SDAxが低い場合、バスの衝突が発生しています（つまり、別のマスターがデータ「0」を送信しようとしています（図21-36））。SDAxがHighにサンプリングされると、BRGがリロードされ、カウントが開始されます。BRGがタイムアウトする前にSDAxがHighからLowに移行した場合、2つのマスターがSDAxをまったく同時にアサートできないため、バスの衝突は発生しません。

BRGがタイムアウトする前にSCLxがHighからLowになり、SDAxがまだアサートされていない場合、バスの衝突が発生します。この場合、別のマスターが繰り返し開始条件の間にデータ「1」を送信しようとしています。図21-37を参照してください。

BRGタイムアウトの終了時に、SCLxとSDAxの両方がまだHighの場合、SDAxピンはLowに駆動され、BRGがリロードされてカウントを開始します。カウントの終了時に、SCLxピンのステータスに関係なく、SCLxピンがLowに駆動され、リピータースタート条件が完了します。

図21-36: 繰り返し開始条件でのバス衝突(ケース1)

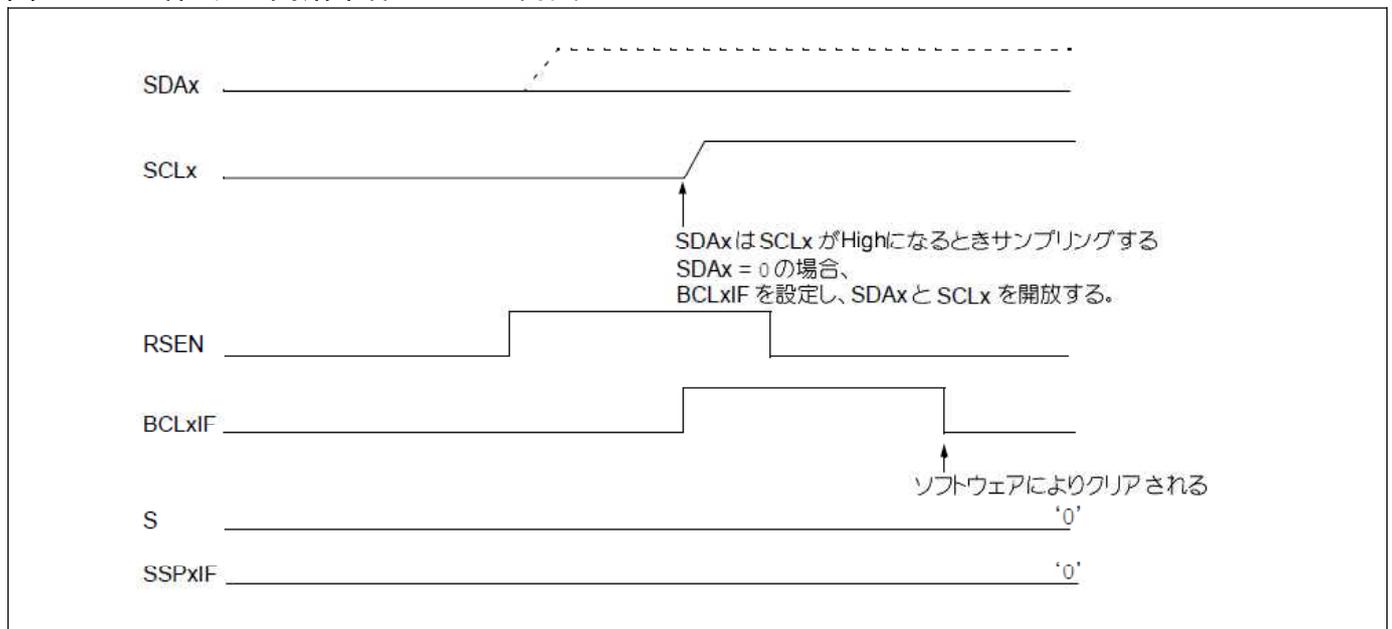
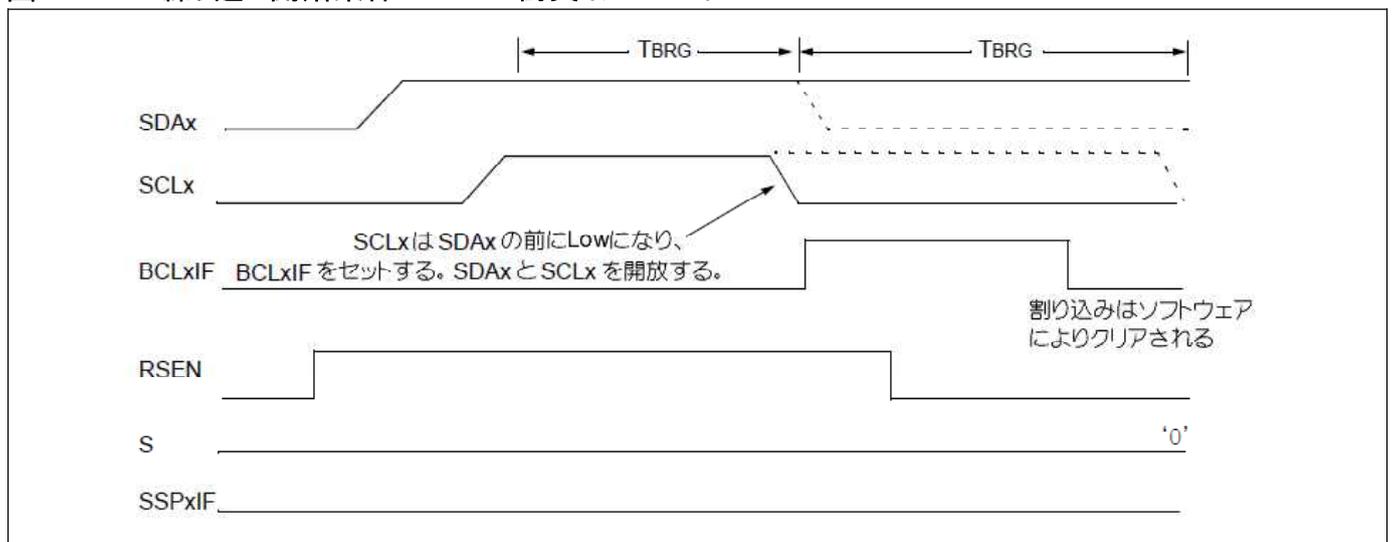


図21-37: 繰り返し開始条件でのバス衝突(ケース2)



## 21.6.13.3 停止状態でのバスの衝突

バスの衝突は、停止状態の間に発生します：

- a) SDAxピンがディアサート（アサート解除）され、Highフロートが許可された後、BRGがタイムアウトした後、SDAxがLowにサンプリングされます（ケース1）。
- b) SCLxピンがディアサートされた後、SDAxがHighになる前にSCLxがLowにサンプリングされます（ケース2）。

停止条件は、SDAxがLowにアサート（論理有効）された状態で始まります。SDAxがLowにサンプリングされると、SCLxピンはフロートします。ピンがHighにサンプリングされると（クロック仲裁）、ポーレート・ジェネレーターにSSPxADDがロードされ、0までカウントダウンします。BRGがタイムアウトした後、SDAxがサンプリングされます。SDAxが低くサンプリングされている場合、バスの衝突が発生しています。これは、別のマスターがデータ「0」を駆動しようとしたためです（図21-38）。SDAxがHighにフロートする前にSCLxピンがLowにサンプリングされると、バスの衝突が発生します。これは、別のマスターがデータ「0」を駆動しようとする別のケースです（図21-39）。

図21-38: 停止状態でのバスの衝突(ケース1)

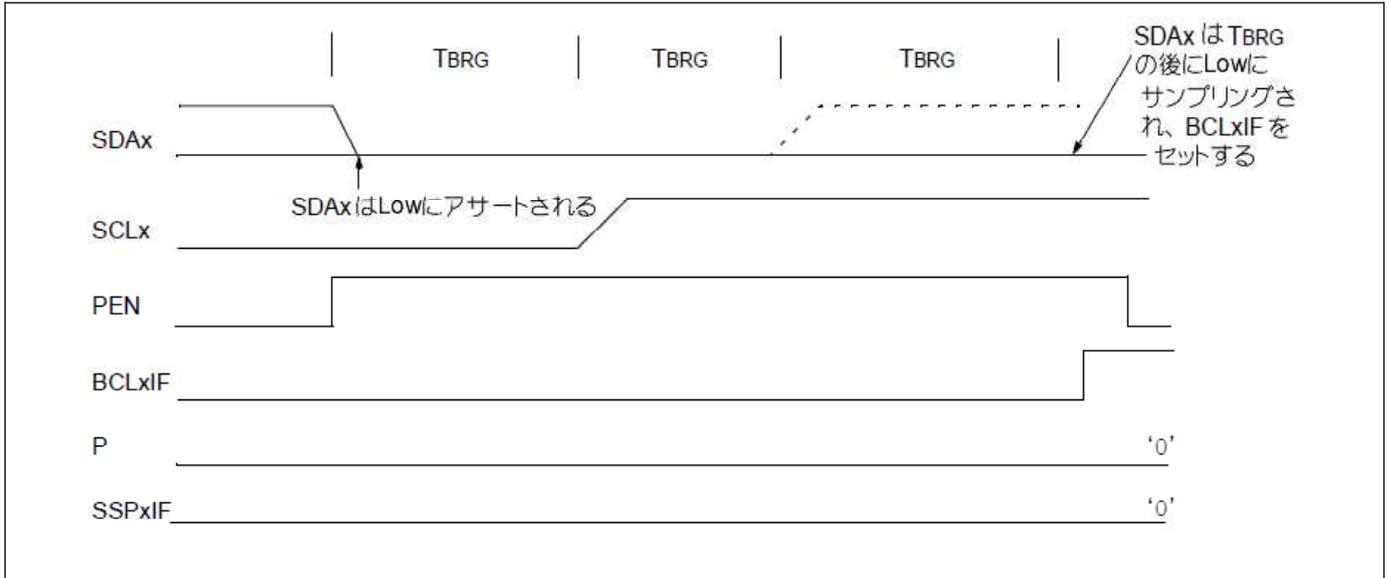
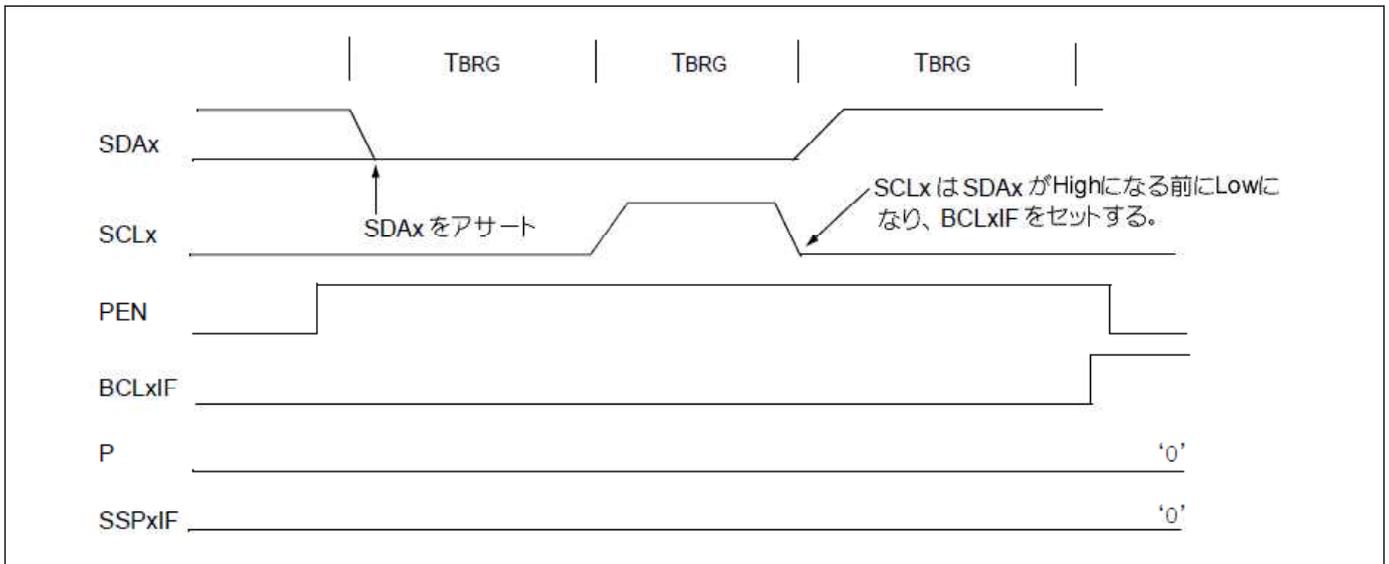


図21-39: 停止状態でのバスの衝突(ケース2)



# PIC16(L)F1508/9

表21-3: I<sup>2</sup>C™操作に関連するレジスタの概要

名前	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ページ上の登録
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	75
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	—	TMR2IE	TMR1IE	76
PIE2	OSFIE	C2IE	C1IE	—	BCL1IE	NCO1IE	—	—	77
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	—	TMR2IF	TMR1IF	79
PIR2	OSFIF	C2IF	C1IF	—	BCL1IF	NCO1IF	—	—	80
TRISA	—	—	TRISA5	TRISA4	— <sup>(1)</sup>	TRISA2	TRISA1	TRISA0	109
SSP1ADD	ADD<7:0>								222
SSP1BUF	MSSP受信バッファ/送信レジスタ								173*
SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>				219
SSP1CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	220
SSP1CON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	221
SSP1MSK	MSK<7:0>								222
SSP1STAT	SMP	CKE	D $\bar{A}$	P	S	R $\bar{W}$	UA	BF	218

記録: — =実装されていない場所。「0」と読みます。網掛けのセルは、I<sup>2</sup>CモードのMSSPモジュールでは使用されません。

\* ページは登録情報を提供します。

注1: 未実装、「1」と読みます。

## 21.7 ポーレート・ジェネレーター

MSSPモジュールには、I<sup>2</sup>CモードとSPIマスター・モードの両方でクロック生成に使用できるポーレート・ジェネレーターがあります。ポーレート・ジェネレーター (BRG) のリロード値は、SSPxADDレジスター (レジスター21-6) に配置されます。SSPxBUFへの書き込みが発生すると、ポーレート・ジェネレーターは自動的にカウントダウンを開始します。

指定された操作が完了すると、内部クロックは自動的にカウントを停止し、クロック・ピンは最後の状態のままになります。

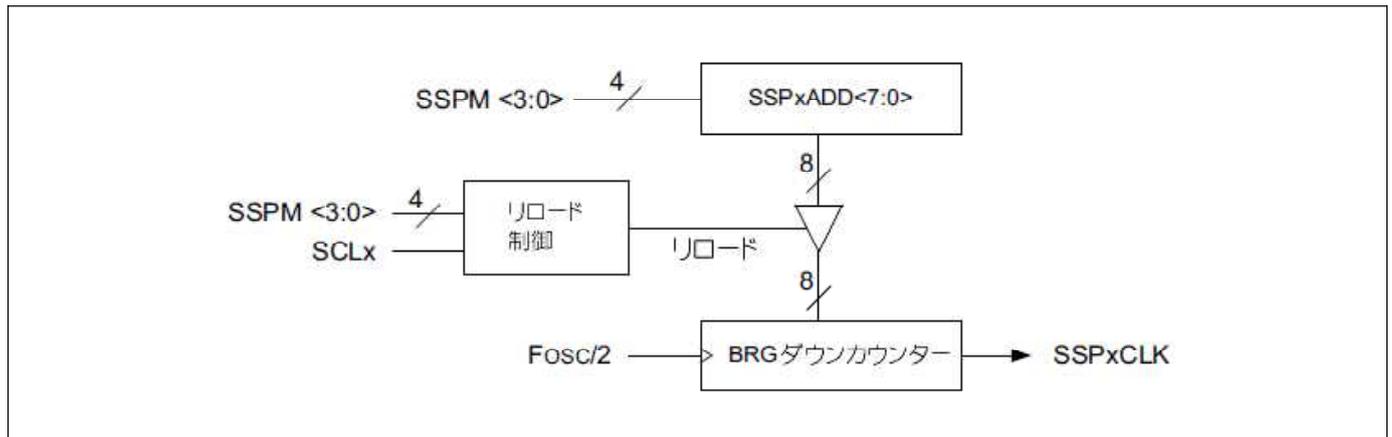
図21-40の内部信号「リロード」は、SSPxADDからの値をトリガーしてBRGカウンターにロードします。これは、モジュール・クロック・ラインの発振ごとに2回発生します。リロード信号がアサート (論理有効) されるタイミングを決定するロジックは、MSSPが動作しているモードによって異なります。

表21-4に、命令サイクルとSSPxADDにロードされたBRG値に基づくクロック・レートを示します。

式21-1:

$$F_{CLOCK} = \frac{F_{OSC}}{(SSPxADD + 1)(4)}$$

図21-40:ポーレート・ジェネレーターのブロック図



注釈: 0x00、0x01、および0x02の値は、I<sup>2</sup>Cのポーレート・ジェネレーターとして使用される場合、SSPxADDには無効です。これは実装上の制限です。

表21-4: BRGを使用したMSSPクロック・レート

Fosc	Fcy	BRG値	Fclock (BRGの2ロールオーバー)
16 MHz	4 MHz	09h	400 kHz
16 MHz	4 MHz	0Ch	308 kHz
16 MHz	4 MHz	27h	100 kHz
4 MHz	1 MHz	09h	100 kHz

注釈: システムがI/Oタイミング要件をサポートするように設計されていることを確認するには、表29-9および図29-7のI/Oポートの電気的およびタイミング仕様を参照してください。

# PIC16(L)F1508/9

## 21.8 レジスター定義: MSSP制御

### レジスター21-1: SSPxSTAT: SSPステータス・レジスター

R/W-0/0	R/W-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0
SMP	CKE	D $\bar{A}$	P	S	R $\bar{W}$	UA	BF
bit 7							bit 0

<b>記録:</b>		
R = 読み取り可能ビット	W = 書き込み可能ビット	U = 実装されていないビット、「0」として読み取られる
u = ビットは変更されていません	x = ビットは不明です	-n/n = PORでの値およびBOR/他のすべてのリセットでの値
[1] = ビットが設定されている	[0] = ビットがクリアされている	

- bit 7 **SMP**: SPIデータ入力サンプル・ビット
- SPIマスター・モードで:  
 1 = データ出力時間の終了時にサンプリングされた入力データ  
 0 = データ出力時間の途中でサンプリングされた入力データ
- SPIスレーブ・モードで:  
 SPIをスレーブ・モードで使用する場合は、SMPをクリアする必要があります
- I<sup>2</sup>Cマスターまたはスレーブ・モードで:  
 1 = スルーレート (動作速度) 制御は無効  
 0 = スルーレート制御は有効
- bit 6 **CKE**: SPIクロック・エッジ選択ビット (SPIモードのみ)
- SPIマスターまたはスレーブ・モードで:  
 1 = アクティブからアイドル・クロック状態への移行時に送信が発生します  
 0 = 送信は、アイドル状態からアクティブ・クロック状態への移行時に発生します
- I<sup>2</sup>Cモードのみで:  
 1 = しきい値がSMBus仕様に準拠するように、入力ロジックを有効にします  
 0 = SMBus固有の入力を無効にする
- bit 5 **D $\bar{A}$** : データ/アドレス・ビット (I<sup>2</sup>Cモードのみ)
- 1 = 受信または送信された最後のバイトがデータであったことを示します  
 0 = 受信または送信された最後のバイトがアドレスであったことを示します
- bit 4 **P**: ストップ・ビット
- (I<sup>2</sup>Cモードのみ。このビットは、MSSPモジュールが無効になるとクリアされ、SSPENがクリアされます。)  
 1 = ストップ・ビットが最後に検出されたことを示します (このビットはリセット時に「0」です)  
 0 = ストップ・ビットが最後に検出されなかった
- bit 3 **S**: スタート・ビット
- (I<sup>2</sup>Cモードのみ。このビットは、MSSPモジュールが無効になるとクリアされ、SSPENがクリアされます。)  
 1 = スタート・ビットが最後に検出されたことを示します (このビットはリセット時に「0」です)  
 0 = スタート・ビットが最後に検出されませんでした
- bit 2 **R $\bar{W}$** : リード/ライト・ビット情報 (I<sup>2</sup>Cモードのみ)
- このビットは、最後のアドレス一致に続くR $\bar{W}$ ビット情報を保持します。このビットは、アドレスの一致から次のスタート・ビット、ストップ・ビット、またはACKビットまでのみ有効です。
- I<sup>2</sup>Cスレーブ・モードで:  
 1 = リード (読み込み)  
 0 = ライト (書き込み)
- I<sup>2</sup>Cマスター・モードで:  
 1 = 送信は進行中です  
 0 = 送信は進行中ではありません  
 このビットをSEN、RSEN、PEN、RCEN、またはACKENとOR接続すると、MSSPがアイドル・モードであるかどうかを示されます。
- bit 1 **UA**: アドレス更新ビット (10ビットI<sup>2</sup>Cモードのみ)
- 1 = ユーザーがSSPxADDレジスターのアドレスを更新する必要があることを示します  
 0 = アドレスを更新する必要はありません
- bit 0 **BF**: バッファ・フル・ステータス・ビット
- 受信 (SPIとI<sup>2</sup>Cモード):  
 1 = 受信が完了し、SSPxBUFがいっぱいです  
 0 = 受信が完了していません。SSPxBUFが空です
- 送信 (I<sup>2</sup>Cモードのみ):  
 1 = データ送信が進行中 (ACKビットとストップ・ビットを含まない)、SSPxBUFがいっぱいです  
 0 = データ送信が完了し (ACKビットとストップ・ビットは含まれません)、SSPxBUFは空です

## レジスタ21-2: SSPxCON1: SSP制御レジスタ1

R/C/HS-0/0	R/C/HS-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
WCOL	SSPOV <sup>(1)</sup>	SSPEN	CKP	SSPM<3:0>			
bit 7							bit 0

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている      HS =ビットはハードウェアによりセット      C =ユーザーがクリア

- bit 7      **WCOL**: 書き込み衝突検出ビット  
マスター・モード:  
1 = 送信を開始するためのI<sup>2</sup>C条件が無効なときに、SSPxBUFレジスタへの書き込みが試行されました  
0 = 衝突なし  
スレーブ・モード:  
1 = SSPxBUFレジスタは、前のワードを送信している間に書き込まれます (ソフトウェアでクリアする必要があります)  
0 = 衝突なし
- bit 6      **SSPOV**: 受信オーバーフロー・インジケータ・ビット<sup>(1)</sup>  
SPIモードで:  
1 = SSPxBUFレジスタが前のデータを保持している間に、新しいバイトが受信されます。オーバーフローが発生すると、SSPxSRのデータは失われます。オーバーフローはスレーブ・モードでのみ発生します。スレーブ・モードでは、設定のオーバーフローを回避するために、データを送信するだけの場合でも、ユーザーはSSPxBUFを読み取る必要があります。マスター・モードでは、SSPxBUFレジスタへの書き込みによって新しい受信 (および送信) が開始されるため、オーバーフロー・ビットは設定されません (ソフトウェアでクリアする必要があります)。  
0 = オーバーフローなし  
I<sup>2</sup>Cモードで:  
1 = SSPxBUFレジスタが前のバイトを保持している間にバイトが受信されます。SSPOVは、送信モードでは「ドントケア (心配するな)」です (ソフトウェアでクリアする必要があります)。  
0 = オーバーフローなし
- bit 5      **SSPEN**: 同期シリアル・ポート許可ビット  
どちらのモードでも、有効にすると、これらのピンを入力または出力として適切に構成する必要があります  
SPIモードで:  
1 = シリアル・ポートを有効にし、SCKx、SDOx、SDIx、およびSSxをシリアル・ポート・ピンのソースとして構成します<sup>(2)</sup>  
0 = シリアル・ポートを無効にし、これらのピンをI/Oポート・ピンとして構成します  
I<sup>2</sup>Cモードで:  
1 = シリアル・ポートを有効にし、SDAxピンとSCLxピンをシリアル・ポート・ピンのソースとして構成します<sup>(3)</sup>  
0 = シリアル・ポートを無効にし、これらのピンをI/Oポート・ピンとして構成します
- bit 4      **CKP**: クロック極性選択ビット  
SPIモードで:  
1 = クロックのアイドル状態はHighレベルです  
0 = クロックのアイドル状態がLowレベルです  
I<sup>2</sup>Cスレーブ・モードで:  
SCLx開放制御  
1 = クロックを有効にする  
0 = クロックをLowに保持します (クロック延長)。(データのセットアップ時間を確保するために使用されます)  
I<sup>2</sup>Cマスター・モードで:  
このモードでは未使用
- bit 3~0      **SSPM<3:0>**: 同期シリアル・ポート・モード選択ビット  
0000 = SPIマスター・モード、クロック = Fosc/4  
0001 = SPIマスター・モード、クロック = Fosc/16  
0010 = SPIマスター・モード、クロック = Fosc/64  
0011 = SPIマスター・モード、クロック = T2\_match/2  
0100 = SPIスレーブ・モード、クロック = SCKxピン、 $\overline{SS}$ ピン制御有効  
0101 = SPIスレーブ・モード、クロック = SCKxピン、 $\overline{SS}$ ピン制御無効、SSxはI/Oピンとして使用できます  
0110 = I<sup>2</sup>Cスレーブ・モード、7ビット・アドレス  
0111 = I<sup>2</sup>Cスレーブ・モード、10ビット・アドレス  
1000 = I<sup>2</sup>Cマスター・モード、クロック = Fosc/(4×(SSPxADD+1))<sup>(4)</sup>  
1001 = 予約済み  
1010 = SPIマスター・モード、クロック = Fosc/(4×(SSPxADD+1))<sup>(5)</sup>  
1011 = I<sup>2</sup>Cファームウェア制御マスター・モード (スレーブ・アイドル)  
1100 = 予約済み  
1101 = 予約済み  
1110 = I<sup>2</sup>Cスレーブ・モード、7ビット・アドレス、スタートとストップ・ビット割り込み有効で  
1111 = I<sup>2</sup>Cスレーブ・モード、10ビット・アドレス、スタートとストップ・ビット割り込み有効で

- 注1: マスター・モードでは、SSPxBUFレジスタへの書き込みによって新しい受信 (および送信) が開始されるため、オーバーフロー・ビットは設定されません。  
2: 有効にした場合、これらのピンは入力または出力として適切に構成する必要があります。  
3: 有効にした場合、SDAxピンとSCLxピンを入力として構成する必要があります。  
4: SSPxADD値の 0、1または2は、I<sup>2</sup>Cモードのサポートされていません。  
5: SSPxADD値の「0」はサポートされていません。代わりにSSPM = 0000 をお使い下さい。

# PIC16(L)F1508/9

## レジスタ-21-3: SSPxCON2: SSP制御レジスタ-2<sup>(1)</sup>

R/W-0/0	R-0/0	R/W-0/0	R/S/HS-0/0	R/S/HS-0/0	R/S/HS-0/0	R/S/HS-0/0	R/W/HS-0/0
GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
bit 7							bit 0

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている      HC =ハードウェアによりクリアされる      S =ユーザーがセット

- bit 7      **GCEN**: ゼネラル・コール・イネーブル・ビット (I<sup>2</sup>Cスレーブ・モードのみで)  
1 = SSPxSRで一般コール (呼び出し) アドレス (0x00または00h) を受信したときに割り込みを有効にする  
0 = 一般コール・アドレスが無効
- bit 6      **ACKSTAT**: 確認応答ステータス・ビット (I<sup>2</sup>Cモードのみで)  
1 = 確認応答が受信されませんでした  
0 = 確認応答が受信されました
- bit 5      **ACKDT**: 確認応答データ・ビット (I<sup>2</sup>Cモードのみで)  
受信モードで:  
ユーザーが受信の最後に確認応答シーケンスを開始したときに送信される値  
1 = 確認応答でない  
0 = 確認応答
- bit 4      **ACKEN**: 確認応答シーケンス許可ビット (I<sup>2</sup>Cマスター・モードのみで)  
マスター受信モードで:  
1 = SDAxおよびSCLxピンで確認応答シーケンスを開始し、ACKDTデータ・ビットを送信します。  
ハードウェアによって自動的にクリアされます。  
0 = 確認応答シーケンス・アイドル
- bit 3      **RCEN**: 受信許可ビット (I<sup>2</sup>Cマスター・モードのみで)  
1 = I<sup>2</sup>Cの受信モードを有効にします  
0 = 受信アイドル
- bit 2      **PEN**: 停止条件許可ビット (I<sup>2</sup>Cマスター・モードのみで)  
SCKx開放制御:  
1 = SDAxおよびSCLxピンで停止条件を開始します。ハードウェアによって自動的にクリアされます。  
0 = 停止条件アイドル
- bit 1      **RSEN**: 再開条件許可ビット (I<sup>2</sup>Cマスターモードのみで)  
1 = SDAxおよびSCLxピンで繰り返し開始条件を開始します。ハードウェアによって自動的にクリアされます。  
0 = 再開条件アイドル
- bit 0      **SEN**: 開始条件許可/延長許可ビット  
マスターモードで:  
1 = SDAxおよびSCLxピンで開始条件を開始します。ハードウェアによって自動的にクリアされます。  
0 = 開始条件アイドル  
スレーブ・モードで:  
1 = スレーブ送信とスレーブ受信の両方でクロック延長が有効になっています (延長が有効)  
0 = クロック延長は無効にされています

注1: ビットACKEN、RCEN、PEN、RSEN、SENの場合: I<sup>2</sup>Cモジュールがアイドル・モードでない場合、このビットは設定されず (スプーリング (バッファへの一時的保存) なし)、SSPxBUFは書き込まれない可能性があります (またはSSPxBUFへの書き込みは無効)。

## レジスタ21-4: SSPxCON3: SSP制御レジスタ-3

R-0/0	R/W-0/0						
ACKTIM <sup>(3)</sup>	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN
bit 7							bit 0

## 記録:

R =読み取り可能ビット	W =書き込み可能ビット	U =実装されていないビット、「0」として読み取られる
u =ビットは変更されていません	x =ビットは不明です	-n/n =PORでの値およびBOR/他のすべてのリセットでの値
[1]=ビットが設定されている	[0]=ビットがクリアされている	

- bit 7 **ACKTIM**: 確認応答時間ステータス・ビット (I<sup>2</sup>Cモードのみ)<sup>(3)</sup>  
 1 = I<sup>2</sup>Cバスが確認応答シーケンスにあり、SCLxクロックの8番目の立ち下がりエッジに設定されていることを示します  
 0 = 確認応答シーケンスではなく、SCLxクロックの9番目の立ち上がりエッジでクリアされます
- bit 6 **PCIE**: 停止条件割り込み許可ビット (I<sup>2</sup>Cモードのみ)  
 1 = 停止条件の検出時に割り込みを有効にする  
 0 = 停止条件割り込みは無効にされています<sup>(2)</sup>
- bit 5 **SCIE**: 開始条件割り込み許可ビット (I<sup>2</sup>Cモードのみ)  
 1 = 開始条件または再起動条件の検出時に割り込みを有効にする  
 0 = 開始条件割り込みは無効にされています<sup>(2)</sup>
- bit 4 **BOEN**: バッファ上書き許可ビット  
**SPIスレーブ・モードで:**<sup>(1)</sup>  
 1 = SSPxBUFは、BFビットを無視して新しいデータ・バイトがシフトされるたびに更新されます  
 0 = 新しいバイトが受信された場合、SSPxSTATレジスタのBFビットがすでに設定されている状態では、SSPxCON1レジスタのSSPOVビットが設定され、バッファは更新されません。  
**I<sup>2</sup>Cマスター・モードで:**  
 このビットは無視されます。  
**I<sup>2</sup>Cスレーブ・モードで:**  
 1 = SSPxBUFが更新され、受信したアドレス/データ・バイトに対してACKが生成され、BFビット= 0の場合のみSSPOVビットの状態が無視されます。  
 0 = SSPxBUFは、SSPOVがクリアされている場合のみ更新されます
- bit 3 **SDAHT**: SDAX保持時間選択ビット (I<sup>2</sup>Cモードのみ)  
 1 = SCLxの立ち下がりエッジ後のSDAxでの最小300 nsの保持時間  
 0 = SCLxの立ち下がりエッジ後のSDAxでの最小100 nsの保持時間
- bit 2 **SBCDE**: スレーブ・モード・バス衝突検出許可ビット (I<sup>2</sup>Cスレーブ・モードのみ)  
 SCLxの立ち上がりエッジにある場合、モジュールがHigh状態を出力しているとき、SDAxはLowにサンプリングされ、PIR2レジスタのBCLxIFビットがセットされ、バスがアイドル状態になります。  
 1 = スレーブ・バス衝突割り込みを有効にする  
 0 = スレーブ・バス衝突割り込みは無効にされています
- bit 1 **AHEN**: アドレス・ホールド (保持) 許可ビット (I<sup>2</sup>Cスレーブ・モードのみ)  
 1 = 一致する受信アドレスバイトのSCLxの8番目の立ち下がりエッジに続いて、SSPxCON1レジスタのCKPビットがクリアされ、SCLxはローに保持されます。  
 0 = アドレス保持は無効にされています
- bit 0 **DHEN**: データ・ホールド (保持) 許可ビット (I<sup>2</sup>Cスレーブ・モードのみ)  
 1 = 受信データ・バイトのSCLxの8番目の立ち下がりエッジに続いて、スレーブ・ハードウェアはSSPxCON1レジスタのCKPビットをクリアし、SCLxはLowに保持されます。  
 0 = データ保持は無効にされています

- 注1: デイジーチェーン化されたSPI操作の場合、ユーザーは最後に受信したバイトを除くすべてを無視できます。SSPOVは、新しいバイトが受信され、BF = 1の場合でも設定されますが、ハードウェアは最新のバイトをSSPxBUFに書き込み続けます。
- 2: このビットは、開始および停止条件の検出が有効として明示的にリストされているスレーブ・モードでは効果がありません。
- 3: ACKTIMステータス・ビットは、AHENビットまたはDHENビットがセットされている場合のみアクティブになります。

# PIC16(L)F1508/9

## レジスタ21-5: SSPxMSK: SSPマスク・レジスタ

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
MSK<7:0>							
bit 7							bit 0

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
[1]=ビットが設定されている      [0]=ビットがクリアされている

bit 7~1 **MSK<7:1>**: マスク・ビット

1 = 受信したアドレス・ビットnはSSPxADD<n>と比較され、I<sup>2</sup>Cアドレスの一致が検出されます。  
0 = 受信したアドレス・ビットnは、I<sup>2</sup>Cアドレスの一致の検出には使用されません。

bit 0 **MSK<0>**: I<sup>2</sup>Cスレーブ・モード10ビット・アドレスの為のマスク・ビット

I<sup>2</sup>Cスレーブ・モード、10ビット・アドレス (SSPM<3:0> = 0111 or 1111) :

1 = 受信したアドレス・ビット0はSSPxADD<0>と比較され、I<sup>2</sup>Cアドレスの一致が検出されます。  
0 = 受信したアドレス・ビット0は、I<sup>2</sup>Cアドレスの一致の検出には使用されません。  
I<sup>2</sup>Cスレーブ・モード、7ビット・アドレス、ビットは無視されます

## レジスタ21-6: SSPxADD: MSSPアドレスとポーレート・レジスタ (I<sup>2</sup>Cモード)

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
ADD<7:0>							
bit 7							bit 0

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
[1]=ビットが設定されている      [0]=ビットがクリアされている

### マスター・モード:

bit 7~0 **ADD<7:0>**: ポーレート・クロック分周器ビット  
 $SCLx \text{ピン・クロック周期} = ((ADD<7:0> + 1) \times 4) / F_{osc}$

### 10ビット・スレーブ・モード。最上位アドレス・バイト:

bit 7~3 **使用されていない**: 最も重要なアドレス・バイトには使用されません。このレジスタのビット状態は「ドントケア (気にするな)」です。マスターによって送信されるビット・パターンはI<sup>2</sup>C仕様によって固定されており、「11110」と等しくなければなりません。ただし、これらのビットはハードウェアによって比較され、このレジスタの値の影響を受けません。

bit 2~1 **ADD<2:1>**: 10ビット・アドレスの最上位2ビット

bit 0 **使用されていない**: このモードでは使用されません。ビット状態は「ドントケア」です。

### 10ビット・スレーブ・モード。最下位アドレス・バイト:

bit 7~0 **ADD<7:0>**: 10ビット・アドレスの最下位8ビット

### 7ビット・スレーブ・モード:

bit 7~1 **ADD<7:1>**: 7ビット・アドレス

bit 0 **使用されていない**: このモードでは使用されません。ビット状態は「ドントケア」です。

## 22.0 エンハンスド・ユニバーサル・シンクロナス・アシンクロナス・レシーバー・トランスミッター (EUSART)

エンハンスド・ユニバーサル・シンクロナス・アシンクロナス・レシーバー・トランスミッター (EUSART: 強化された全般同期非同期受信送信器) モジュールは、シリアルI/O通信周辺機器です。これには、デバイス・プログラムの実行とは無関係に入力または出力シリアル・データ転送を実行するために必要なすべてのクロック・ジェネレーター、シフトレジスター、およびデータ・バッファーが含まれています。シリアル通信インターフェース (SCI) としても知られるEUSARTは、全二重同期システムまたは半二重同期システムとして構成できます。全二重モードは、CRT端末やパーソナルコンピュータなどの周辺システムとの通信に役立ちます。半二重同期モードは、A/DまたはD/A集積回路、シリアルEEPROM、その他のマイクロコントローラなどの周辺機器との通信を目的としています。これらのデバイスには通常、ボーレート生成用の内部クロックがなく、マスター同期デバイスによって提供される外部クロック信号が必要です。

EUSARTモジュールには次の機能が含まれています:

- 全二重非同期送信および受信
- 2文字の入力バッファー
- 1文字の出力バッファー
- プログラム可能な8ビットまたは9ビットの文字長
- 9ビットモードでのアドレス検出
- 入力バッファー・オーバーラン・エラー検出
- 受信した文字フレーミング (作成) ・エラー検出
- 半二重同期マスター
- 半二重同期スレーブ
- 同期モードでのプログラム可能なクロック極性
- スリープ操作

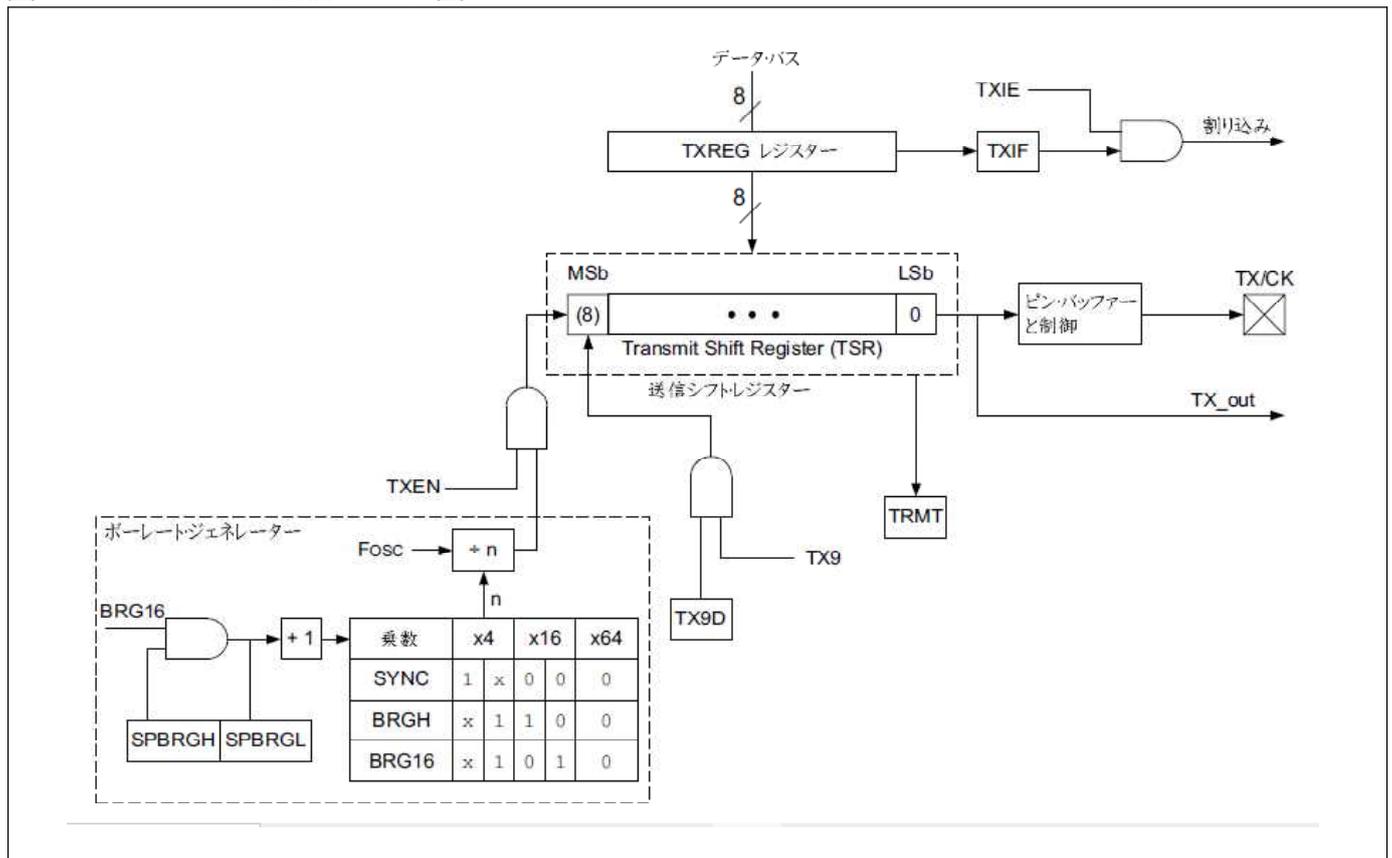
EUSARTモジュールは、次の追加機能を実装しているため、ローカル相互接続ネットワーク (LIN) バス・システムでの使用に最適です:

- ボーレートの自動検出とキャリブレーション (校正)
- 通信中断でのウェイクアップ
- 13ビット中断文字送信

EUSART送信器と受信器のブロック図を図22-1と図22-2に示します。EUSART送信出力 (TX\_out) は、TX/CKピンと、内部で次の周辺機器で使用できます:

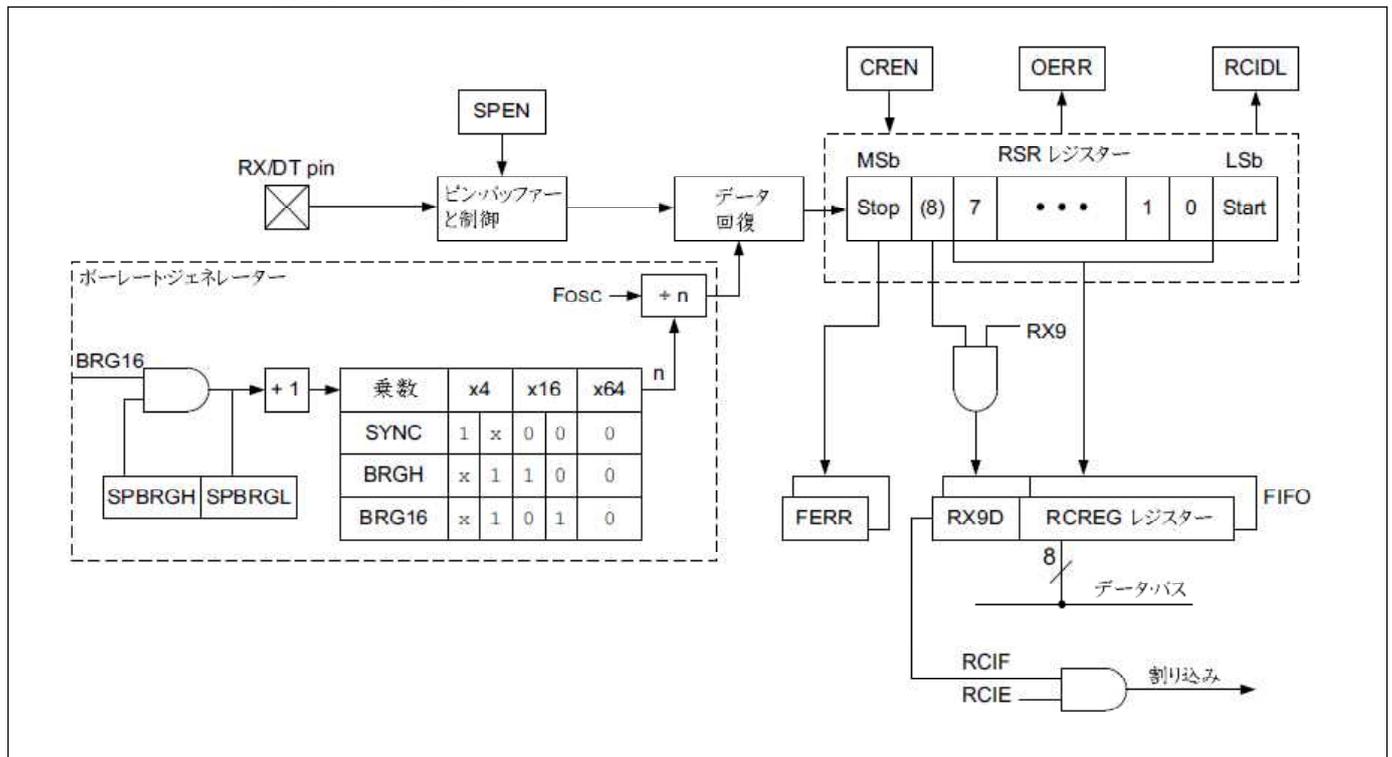
- 構成可能なロジック・セル (CLC)

図22-1: EUSART送信ブロック図



# PIC16(L)F1508/9

図22-2: EUSART受信ブロック図



EUSARTモジュールの動作は3つのレジスタを介して制御されます：

- 送信ステータスと制御 (TXSTA)
- 受信ステータスと制御 (RCSTA)
- ポーレート制御 (BAUDCON)

これらのレジスタについては、レジスタ22-1で詳しく説明していて、レジスタ22-2とレジスタ22-3、それぞれ。

受信器または送信器セクションが有効になっていない場合、対応するRXまたはTXピンを汎用入力および出力に使用できます。

## 22.1 EUSART非同期モード

EUSARTは、標準の非ゼロ復帰（NRZ）形式を使用してデータを送受信します。NRZは、「1」データ・ビットを表すVOHマーク状態と「0」データ・ビットを表すVOL空間状態の2つのレベルで実装されます。NRZは、同じ値の連続して送信されたデータ・ビットが、各ビット送信の間にニュートラル・レベルに戻ることなく、そのビットの出力レベルに留まるという事実を指します。NRZ送信ポートはマーク状態でアイドル（不動）状態になります。各文字送信は、1つのスタート・ビットとそれに続く8つまたは9つのデータ・ビットで構成され、常に1つ以上のストップ・ビットで終了します。スタート・ビットは常にスペースであり、ストップ・ビットは常にマークです。最も一般的なデータ形式は8ビットです。送信された各ビットは、1/(ポーレート)の期間持続します。オンチップの専用8ビット/16ビット・ポーレート・ジェネレーターを使用して、システム発信器から標準のポーレート周波数を導出します。ポーレート設定の例については、表22-5を参照してください。

EUSARTは最初にLSbを送受信します。EUSARTの送信器と受信器は機能的に独立していますが、同じデータ形式とポーレートを共有しています。パリティはハードウェアではサポートされていませんが、ソフトウェアで実装して9番目のデータ・ビットとして格納できます。

### 22.1.1 EUSART非同期送信器

EUSART送信器のブロック図を図22-1に示します。送信器の心臓部はシリアル送信シフトレジスタ（TSR）であり、ソフトウェアから直接アクセスすることはできません。TSRは、TXREGレジスタである送信バッファからデータを取得します。

#### 22.1.1.1 送信器の有効化

EUSART送信器は、次の3つの制御ビットを設定することで非同期操作が可能になります：

- TXEN = 1
- SYNC = 0
- SPEN = 1

他のすべてのEUSART制御ビットはデフォルト状態であると見なされます。

TXSTAレジスタのTXENビットを設定すると、EUSARTの送信回路が有効になります。TXSTAレジスタのSYNCビットをクリアすると、EUSARTが非同期動作に設定されます。RCSTAレジスタのSPENビットを設定すると、EUSARTが有効になり、TX/CK I/Oピンが出力として自動的に構成されます。TX/CKピンがアナログ周辺機器と共有されている場合は、対応するANSELビットをクリアしてアナログI/O機能を無効にする必要があります。

**注釈：** TXEN許可ビットがセットされると、TXIF送信器割り込みフラグがセットされます。

#### 22.1.1.2 送信中のデータ

TXREGレジスタに文字を書き込むことにより、送信が開始されます。これが最初の文字である場合、または前の文字がTSRから完全にフラッシュ（流す？）されている場合、TXREG内のデータはすぐにTSRレジスタに転送されます。TSRに前の文字の全部または一部がまだ含まれている場合、前の文字のストップ・ビットが送信されるまで、新しい文字データはTXREGに保持されます。TXREGの保留中の文字は、ストップ・ビット送信の直後に1つのTCYでTSRに転送されます。スタート・ビット、データ・ビット、ストップ・ビット・シーケンスの送信は、TXREGからTSRへのデータ転送の直後に開始されます。

#### 22.1.1.3 送信データの極性

送信データの極性は、BAUDCONレジスタのSCKPビットで制御できます。このビットのデフォルト状態は「0」で、真の送信アイドル・ビットとデータ・ビットを選択します。SCKPビットを「1」に設定すると、送信データが反転し、真のアイドル・ビットとデータ・ビットが低くなります。SCKPビットは、非同期モードでのみ送信データの極性を制御します。同期モードでは、SCKPビットの機能は異なります。セクション22.5.1.2「クロック極性」を参照してください。

#### 22.1.1.4 送信割り込みフラグ

PIR1レジスタのTXIF割り込みフラグ・ビットは、EUSART送信器が有効で、TXREGでの送信用に文字が保持されていない場合は常にセットされます。言い換えると、TXIFビットは、TSRが文字でビジー（忙しい）であり、新しい文字がTXREGでの送信のためにキューに入れられている場合にのみクリアされます。TXREGを書き込んだ直後にTXIFフラグ・ビットがクリアされることはありません。TXIFは、書き込み実行後の2番目の命令サイクルで有効になります。TXREG書き込みの直後にTXIFをポーリングすると、無効な結果が返されます。TXIFビットは読み取り専用であり、ソフトウェアで設定またはクリアすることはできません。

TXIF割り込みは、PIE1レジスタのTXIE割り込み許可ビットをセットすることで有効にできます。ただし、TXIE許可ビットの状態に関係なく、TXREGが空の場合は常にTXIFフラグビットがセットされます。

データを送信するときに割り込みを使用するには、送信するデータがまだある場合にのみTXIEビットを設定します。送信の最後の文字をTXREGに書き込むときに、TXIE割り込み許可ビットをクリアします。

## 22.1.1.5 TSRステータス

TXSTAレジスタのTRMTビットは、TSRレジスタのステータスを示します。これは読み取り専用ビットです。TRMTビットは、TSRレジスタが空のときにセットされ、文字がTXREGからTSRレジスタに転送されるときにクリアされます。TRMTビットは、すべてのビットがTSRレジスタからシフトアウトされるまでクリアされたままです。このビットには割り込みロジックが関連付けられていないため、ユーザーはこのビットをポーリング（なんども定期的を確認）してTSRステータスを判別する必要があります。

注釈：TSRレジスタはデータメモリにマップされていないため、ユーザーは使用できません。

## 22.1.1.6 送信している9ビット・キャラクター

EUSARTは9ビットの文字伝送をサポートしています。TXSTAレジスタのTX9ビットがセットされると、EUSARTは送信される文字ごとに9ビットをシフトアウトします。TXSTAレジスタのTX9Dビットは、9番目の最上位データビットです。9ビット・データを送信する場合、8つの最下位ビットをTXREGに書き込む前に、TX9Dデータビットを書き込む必要があります。TXREGが書き込まれた直後に、9ビットのデータすべてがTSRシフトレジスタに転送されます。

特別な9ビット・アドレス・モードは、複数の受信器で使用できます。アドレス・モードの詳細については、セクション22.1.2.7「アドレス検出」を参照してください。

## 22.1.1.7 非同期送信セットアップ:

1. SPBRGH、SPBRGLレジスタペア、およびBRGHビットとBRG16ビットを初期化して、目的のボーレートを実現します（セクション22.4「EUSARTボーレート・ジェネレーター（BRG）」を参照）。
2. SYNCビットをクリアし、SPENビットを設定して、非同期シリアル・ポートを有効にします。
3. 9ビットの送信が必要な場合は、TX9制御ビットを設定します。設定された9番目のデータビットは、受信器がアドレス検出用に設定されている場合、8つの最下位データビットがアドレスであることを示します。
4. 反転送信が必要な場合は、SCKPビットを設定します。
5. TXEN制御ビットを設定して、送信を有効にします。これにより、TXIF割り込みビットが設定されます。
6. 割り込みが必要な場合は、PIE1レジスタのTXIE割り込み許可ビットをセットします。INTCONレジスタのGIEビットとPEIEビットも設定されている場合、割り込みはすぐに発生します。
7. 9ビット送信を選択した場合、9番目のビットをTX9Dデータビットにロードする必要があります。
8. 8ビット・データをTXREGレジスタにロードします。これにより、送信が開始されます。

図22-3: 非同期送信

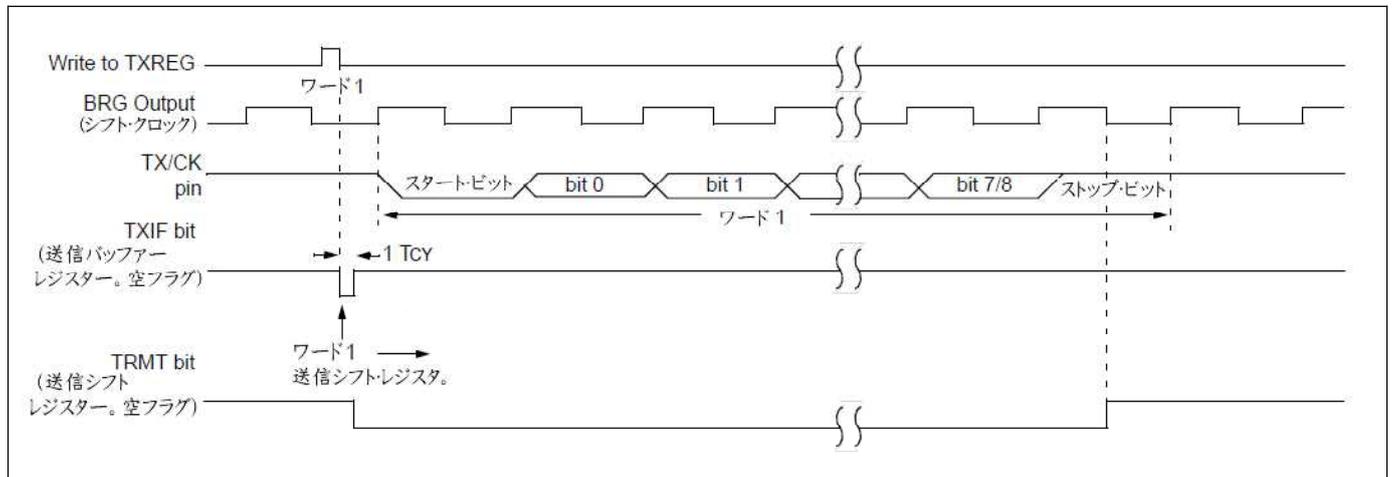
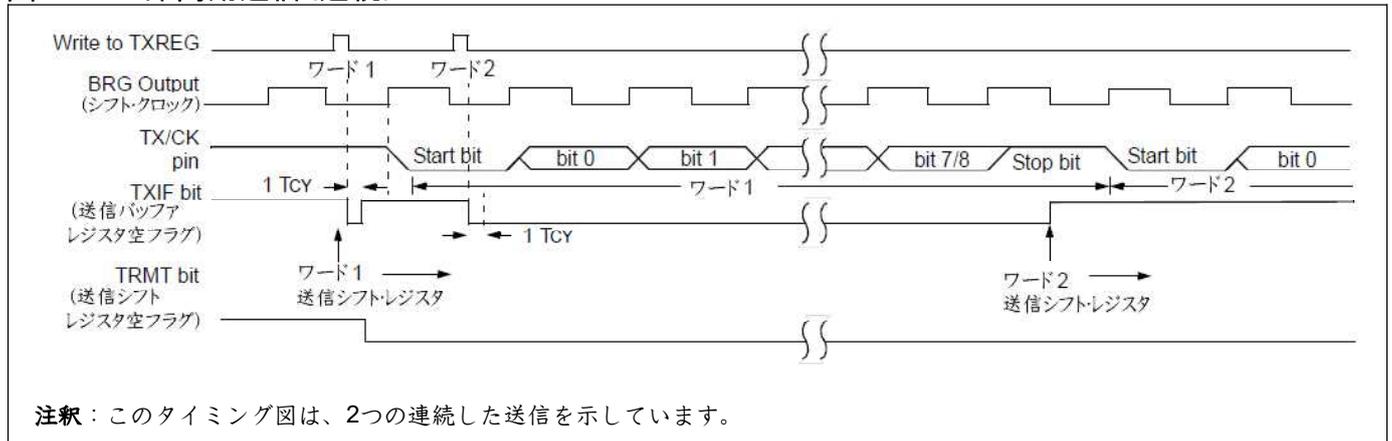


図22-4: 非同期送信(連続)



注釈：このタイミング図は、2つの連続した送信を示しています。

表22-1: 非同期送信に関連付けられたレジスタの概要

名前	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ページ上の登録
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	235
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	75
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	—	TMR2IE	TMR1IE	76
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	—	TMR2IF	TMR1IF	79
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	234*
SPBRGL	BRG<7:0>								236*
SPBRGH	BRG<15:8>								236*
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	113
TXREG	EUSART送信データレジスタ								225
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	233

記録： — =実装されていない場所、「0」と読みます。網掛けのセルは非同期送信には使用されません。

\* ページは登録情報を提供します。

## 22.1.2 EUSART非同期受信器

非同期モードは通常、RS-232システムで使用されます。受信器のブロック図を図22-2に示します。データはRX/DTピンで受信され、データ回復ブロックを駆動します。データ・リカバリブ（回復）ロックは、実際にはボーレートの16倍で動作する高速シフターですが、シリアル受信シフト・レジスタ（RSR）はビット・レートで動作します。文字の8ビットまたは9ビットがすべてシフトインされると、それらはすぐに2文字の先入れ先出し（FIFO）メモリに転送されます。FIFOバッファリングにより、ソフトウェアがEUSART受信器のサービスを開始する前に、2つの完全な文字の受信と3番目の文字の開始が可能になります。FIFOおよびRSRレジスタには、ソフトウェアから直接アクセスすることはできません。受信したデータへのアクセスは、RCREGレジスタを介して行われます。

### 22.1.2.1 受信器の有効化

EUSART受信器は、次の3つの制御ビットを構成することにより、非同期操作が可能になります：

- CREN = 1
- SYNC = 0
- SPEN = 1

他のすべてのEUSART制御ビットはデフォルト状態であると見なされます。

RCSTAレジスタのCRENビットをセットすると、EUSARTの受信器回路が有効になります。TXSTAレジスタのSYNCビットをクリアすると、EUSARTが非同期動作に設定されます。RCSTAレジスタのSPENビットをセットすると、EUSARTが有効になります。プログラマーは、RX/DT I/Oピンを入力として設定するために、対応するTRISビットを設定する必要があります。

**注釈：** RX/DT機能がアナログ・ピンにある場合、受信器が機能するには、対応するANSELビットをクリアする必要があります。

### 22.1.2.2 データ受信中

受信器のデータ回復回路は、最初のビットの立ち下がりエッジで文字受信を開始します。スタート・ビットとも呼ばれる最初のビットは常にゼロです。データ回復回路は、スタート・ビットの中心までの半分のビット時間をカウントし、ビットがまだゼロであることを確認します。ゼロでない場合、データ回復回路はエラーを生成せずに文字の受信を中止し、スタート・ビットの立ち下がりエッジの検索を再開します。スタート・ビットゼロの検証が成功した場合、データ回復回路は次のビットの中心までフル・ビット時間をカウントします。次に、ビットは多数決検出回路によってサンプリングされ、結果の「0」または「1」がRSRにシフトされます。これは、すべてのデータビットがサンプリングされてRSRにシフトされるまで繰り返されます。最後の1ビット時間が測定され、レベルがサンプリングされます。これはストップ・ビットであり、常に「1」です。データ回復回路がストップ・ビット位置で「0」をサンプリングする場合、この文字にフレーミング（作成）エラーが設定されます。それ以外の場合、この文字のフレーミング・エラーはクリアされます。フレーミング・エラーの詳細については、セクション22.1.2.4「フレーミング・エラーの受信」を参照してください。

すべてのデータ・ビットとストップ・ビットが受信されるとすぐに、RSRの文字がEUSART受信FIFOに転送され、PIR1レジスタのRCIF割り込みフラグ・ビットがセットされます。FIFOの先頭の文字は、RCREGレジスタを読み取ることによってFIFOから転送されます。

**注釈：** 受信FIFOがオーバーランした場合、オーバーラン状態が解消されるまで、追加の文字は受信されません。オーバーラン・エラーの詳細については、セクション22.1.2.5「オーバーラン・エラーの受信」を参照してください。

### 22.1.2.3 受信割り込み

PIR1レジスタのRCIF割り込みフラグ・ビットは、EUSART受信器が有効で、受信FIFOに未読文字がある場合は常にセットされます。RCIF割り込みフラグ・ビットは読み取り専用であり、ソフトウェアで設定またはクリアすることはできません。

RCIF割り込みは、次のすべてのビットを設定することで有効になります：

- RCIE、PIE1レジスタの割り込み許可ビット
- PEIE、INTCONレジスタの周辺割り込み許可ビット
- GIE、INTCONレジスタのグローバル割り込み許可ビット

割り込み許可ビットの状態に関係なく、FIFOに未読文字がある場合、RCIF割り込みフラグ・ビットがセットされます。

#### 22.1.2.4 受信フレーミング・エラー

受信FIFOバッファの各文字には、対応するフレーミング（作成）エラー・ステータス・ビットがあります。フレーミング・エラーは、予想された時間にストップ・ビットが見られなかったことを示します。フレーミング・エラー・ステータスは、RCSTAレジスタのFERRビットを介してアクセスされます。FERRビットは、受信FIFOの先頭の未読文字のステータスを表します。したがって、RCREGを読み取る前にFERRビットを読み取る必要があります。

FERRビットは読み取り専用であり、受信FIFOの先頭の未読文字にのみ適用されます。フレーミング・エラー（FERR = 1）は、追加の文字の受信を妨げるものではありません。FERRビットをクリアする必要はありません。FIFOバッファから次の文字を読み取ると、FIFOが次の文字に進み、対応する次のフレーミング・エラーが発生します。

FERRビットは、EUSARTをリセットするRCSTAレジスタのSPENビットをクリアすることで強制的にクリアできます。RCSTAレジスタのCRENビットをクリアしても、FERRビットには影響しません。フレーミング・エラー自体は割り込みを生成しません。

**注釈：**受信FIFO内のすべての受信文字にフレーミング（作成）エラーがある場合、RCREGを繰り返し読み取っても、FERRビットはクリアされません。

#### 22.1.2.5 受信オーバーラン・エラー

受信FIFOバッファは2文字を保持できます。FIFOにアクセスする前に、3番目の文字全体を受信すると、オーバーラン・エラーが生成されます。これが発生すると、RCSTAレジスタのOERRビットがセットされます。すでにFIFOバッファにある文字は読み取ることができますが、エラーがクリアされるまで追加の文字は受信されません。エラーは、RCSTAレジスタのCRENビットをクリアするか、RCSTAレジスタのSPENビットをクリアしてEUSARTをリセットすることによってクリアする必要があります。

#### 22.1.2.6 9ビット・キャラクター受信

EUSARTは9ビットの文字受信をサポートしています。RCSTAレジスタのRX9ビットが設定されると、EUSARTは受信した文字ごとに9ビットをRSRにシフトします。RCSTAレジスタのRX9Dビットは、受信FIFOの最上位の未読文字の9番目で最上位のデータ・ビットです。受信FIFOバッファから9ビット・データを読み取る場合、RCREGから8つの最下位ビットを読み取る前に、RX9Dデータ・ビットを読み取る必要があります。

#### 22.1.2.7 アドレス検出

RS-485システムのように、複数の受信器が同じ伝送ラインを共有する場合は、特別なアドレス検出モードを使用できません。アドレス検出は、RCSTAレジスタのADDENビットをセットすることで有効になります。

アドレス検出には9ビットの文字受信が必要です。アドレス検出を有効にすると、9番目のデータ・ビットが設定された文字のみが受信FIFOバッファに転送され、RCIF割り込みビットが設定されます。他のすべての文字は無視されます。

アドレス文字を受信すると、ユーザー・ソフトウェアはアドレスが自分のアドレスと一致するかどうかを判断します。アドレスが一致すると、ユーザー・ソフトウェアは、次のストップ・ビットが発生する前にADDENビットをクリアしてアドレス検出を無効にする必要があります。ユーザー・ソフトウェアが、使用されているメッセージ・プロトコルによって決定されるメッセージの終わりを検出すると、ソフトウェアはADDENビットを設定することによって受信器をアドレス検出モードに戻します。

## 22.1.2.8 非同期受信のセットアップ:

1. SPBRGH、SPBRGLレジスター・ペア、およびBRGHビットとBRG16ビットを初期化して、目的のポーレートを實現します（セクション22.4「EUSARTポーレート・ジェネレーター（BRG）」を参照）。
2. RXピンのANSELビットをクリアします（該当する場合）。
3. SPENビットを設定して、シリアル・ポートを有効にします。非同期操作では、SYNCビットをクリアする必要があります。
4. 割り込みが必要な場合は、PIE1レジスターのRCIEビットとINTCONレジスターのGIEおよびPEIEビットを設定します。
5. 9ビットの受信が必要な場合は、RX9ビットを設定します。
6. CRENビットを設定して、受信を有効にします。
7. 文字がRSRから受信バッファーに転送されると、RCIF割り込みフラグ・ビットがセットされます。RCIE割り込み許可ビットも設定されている場合、割り込みが生成されます。
8. RCSTAレジスターを読み取ってエラー・フラグを取得し、9ビットのデータ受信が有効になっている場合は9番目のデータ・ビットを取得します。
9. RCREGレジスターを読み取って、受信した8つの最下位データ・ビットを受信バッファーから取得します。
10. オーバーランが発生した場合は、CREN受信器許可ビットをクリアしてOERRフラグをクリアします。

## 22.1.2.9 9ビット・アドレス検出モード・セットアップ

このモードは通常、RS-485システムで使用されます。アドレス検出を有効にして非同期受信を設定するには：

1. SPBRGH、SPBRGLレジスター・ペア、およびBRGHビットとBRG16ビットを初期化して、目的のポーレートを實現します（セクション22.4「EUSARTポーレート・ジェネレーター（BRG）」を参照）。
2. RXピンのANSELビットをクリアします（該当する場合）。
3. SPENビットを設定して、シリアル・ポートを有効にします。非同期操作では、SYNCビットをクリアする必要があります。
4. 割り込みが必要な場合は、PIE1レジスターのRCIEビットとINTCONレジスターのGIEおよびPEIEビットを設定します。
5. RX9ビットを設定して、9ビット受信を有効にします。
6. ADDENビットを設定して、アドレス検出を有効にします。
7. CRENビットを設定して、受信を有効にします。
8. 9番目のビットが設定された文字がRSRから受信バッファーに転送されると、RCIF割り込みフラグビットが設定されます。RCIE割り込み許可ビットも設定されている場合、割り込みが生成されます。
9. RCSTAレジスターを読み取って、エラー・フラグを取得します。9番目のデータ・ビットは常にセットされます。
10. RCREGレジスターを読み取って、受信バッファーから受信した8つの最下位データ・ビットを取得します。ソフトウェアは、これがデバイスのアドレスであるかどうかを判断します。
11. オーバーランが発生した場合は、CREN受信器許可ビットをクリアしてOERRフラグをクリアします。
12. デバイスがアドレス指定されている場合は、ADDENビットをクリアして、受信したすべてのデータを受信バッファーに入れ、割り込みを生成できるようにします。

図22-5: 非同期受信

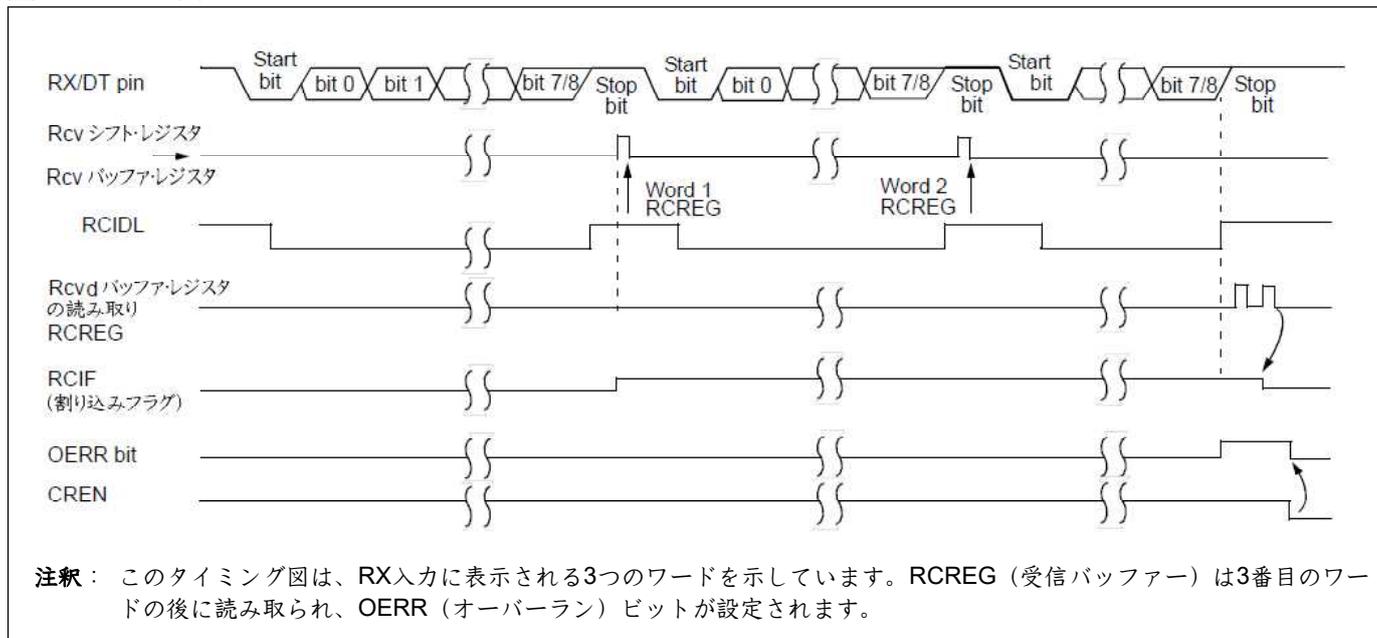


表22-2: 非同期受信に関連付けられたレジスタの概要

名前	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ページ上の登録
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	235
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	75
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	—	TMR2IE	TMR1IE	76
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	—	TMR2IF	TMR1IF	79
RCREG	EUSART受信データレジスター								228*
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	234*
SPBRGL	BRG<7:0>								236*
SPBRGH	BRG<15:8>								236*
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	113
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	233

記録: — =実装されていない場所、「0」と読みます。網掛けのセルは非同期受信には使用されません。

\* ページは登録情報を提供します。

## 22.2 非同期動作でのクロック精度

工場では、内部発振器ブロック出力 (INTOSC) を校正しています。ただし、INTOSC周波数は、VDDまたは温度の変化に応じてドリフト (偏差) する可能性があり、これは非同期ポーレートに直接影響します。

自動ポー検出機能 (セクション22.4.1「自動ポー検出」を参照) を使用して、INTOSC周波数の変化を補正できます。

ポーレート・ジェネレーターを調整して周辺クロック周波数の段階的な変化を補正する場合、十分な解像度が得られない場合があります。

## 22.3 レジスター定義: EUSART制御

### レジスター22-1: TXSTA: ステータスおよび制御レジスターの送信

R/W-/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R-1/1	R/W-0/0
CSRC	TX9	TXEN <sup>(1)</sup>	SYNC	SENDB	BRGH	TRMT	TX9D
bit 7							bit 0

<b>記録:</b>		
R =読み取り可能ビット	W =書き込み可能ビット	U =実装されていないビット、「0」として読み取られる
u =ビットは変更されていません	x =ビットは不明です	-n/n =PORでの値およびBOR/他のすべてのリセットでの値
[1]=ビットが設定されている	[0]=ビットがクリアされている	

- bit 7 **CSRC**: クロック・ソース選択ビット  
非同期モード:  
 気にするな  
同期モード:  
 1 = マスター・モード (BRGから内部的に生成されたクロック)  
 0 = スレーブ・モード (外部ソースからのクロック)
- bit 6 **TX9**: 9ビット送信許可ビット  
 1 = 9ビット送信を選択  
 0 = 8ビット送信を選択
- bit 5 **TXEN**: 送信許可ビット<sup>(1)</sup>  
 1 = 送信可能  
 0 = 送信無効
- bit 4 **SYNC**: EUSARTモード選択ビット  
 1 = 同期モード  
 0 = 非同期モード
- bit 3 **SENDB**: ブレーク・キャラクターを送るビット  
非同期モード:  
 1 = 次の送信で同期ブレークを送る (完了時にハードウェアによってクリアされます)  
 0 = 同期ブレーク送信が完了しました  
同期モード:  
 気にするな
- bit 2 **BRGH**: 高ボーレート選択ビット  
非同期モード:  
 1 = 高速  
 0 = 低速  
同期モード:  
 このモードでは未使用
- bit 1 **TRMT**: 送信シフト・レジスター・ステータス・ビット  
 1 = TSRは空  
 0 = TSRは満タン
- bit 0 **TX9D**: 送信データの9番目のビット  
 アドレス/データ・ビットまたはパリティ・ビットにすることができます。

注1: SREN/CRENは、同期モードでTXENを上書きします。

# PIC16(L)F1508/9

## レジスタ22-2: RCSTA: 受信ステータスと制御レジスタ

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R-0/0	R-0/0	R-0/0
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit 7							bit 0

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
[1]=ビットが設定されている      [0]=ビットがクリアされている

- bit 7      **SPEN**: シリアル・ポート許可ビット  
1 = シリアル・ポートは有効 (RX/DTおよびTX/CKピンをシリアル・ポートピンとして構成します)  
0 = シリアル・ポートは無効 (リセットで保持)
- bit 6      **RX9**: 9ビット受信許可ビット  
1 = 9ビット受信を選択  
0 = 8ビット受信を選択
- bit 5      **SREN**: シングル受信許可ビット  
非同期モード:  
気にするな  
同期モード—マスター:  
1 = シングル受信を有効  
0 = シングル受信を無効  
このビットは、受信が完了するとクリアされます。  
同期モード—スレーブ  
気にするな
- bit 4      **CREN**: 連続受信許可ビット  
非同期モード:  
1 = 受信器を有効にする  
0 = 受信器を無効にする  
同期モード:  
1 = 許可ビットCRENがクリアされるまで連続受信を有効にします (CRENはSRENを上書きします)  
0 = 連続受信を無効にします
- bit 3      **ADDEN**: アドレス検出許可ビット  
非同期モード9ビット (RX9 = 1):  
1 = RSR<8>が設定されている場合、アドレス検出を有効にし、割り込みを有効にして、受信バッファをロードします  
0 = アドレス検出を無効にし、すべてのバイトを受信し、9番目のビットをパリティ・ビットとして使用できます  
非同期モード8ビット (RX9 = 0):  
気にするな
- bit 2      **FERR**: フレーミング・エラー・ビット  
1 = フレーミング (作成) エラーです (RCREGレジスタを読み取って更新し、次の有効なバイトを受信できません)  
0 = フレーミング・エラーはありません
- bit 1      **OERR**: オーバーラン・エラー・ビット  
1 = オーバーラン・エラーです (ビットCRENをクリアすることでクリアできます)  
0 = オーバーラン・エラーはありません
- bit 0      **RX9D**: 受信データの9番目のビット  
これはアドレス/データビットまたはパリティ・ビットであり、ユーザー・ファームウェアによって計算される必要があります。

## レジスタ22-3: BAUDCON: ボーレート制御レジスタ

R-0/0	R-1/1	U-0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0
ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN
bit 7							bit 0

## 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている

bit 7 **ABDOVF**: 自動ポー検出オーバーフロービット

非同期モード:

- 1 = 自動ポー・タイマーはオーバーフローしました  
0 = 自動ポー・タイマーはオーバーフローしませんでした Auto-baud timer did not overflow

同期モード:

気にするな

bit 6 **RCIDL**: 受信アイドルフラグビット

非同期モード:

- 1 = 受信器はアイドル（不動）です  
0 = スタートビットが受信され、受信器が受信しています

同期モード:

気にするな

bit 5 **未実装**: 「0」として読み取ります

bit 4 **SCKP**: 同期クロック極性選択ビット

非同期モード:

- 1 = 反転データをTX/CKピンに送信します  
0 = 非反転データをTX/CKピンに送信します

同期モード:

- 1 = データはクロックの立ち上がりエッジでクロックされます  
0 = データはクロックの立ち下がりエッジでクロックされます

bit 3 **BRG16**: 16ビット・ボーレート・ジェネレータービット

- 1 = 16ビット・ボーレート・ジェネレーターは使われます  
0 = 8ビット・ボーレート・ジェネレーターは使われます

bit 2 **未実装**: 「0」として読み取ります

bit 1 **WUE**: ウェイクアップ許可ビット

非同期モード:

- 1 = 受信器は立ち下がりエッジを待っています。文字は受信されず、RCIFビットが設定されます。RCIFが設定されると、WUEは自動的にクリアされます。  
0 = 受信器は正常に動作しています

同期モード:

気にするな

bit 0 **ABDEN**: 自動ポー検出許可ビット

非同期モード:

- 1 = 自動ポー検出モードは有効になっています（自動ポーが完了するとクリアされます）  
0 = 自動ポー検出モードは無効になっています

同期モード:

気にするな

## 22.4 EUSARTボーレート・ジェネレーター (BRG)

ボーレート・ジェネレーター（生成器。BRG）は、非同期と同期の両方のEUSART動作のサポート専用の8ビットまたは16ビットのタイマーです。デフォルトでは、BRGは8ビットモードで動作します。BAUDCONレジスタのBRG16ビットを設定すると、16ビット・モードが選択されます。

SPBRGH、SPBRGLレジスタ・ペアは、フリー・ランニング・ボーレート・タイマーの期間を決定します。非同期モードでは、ボーレート周期の乗数は、TXSTAレジスタのBRGHビットとBAUDCONレジスタのBRG16ビットの両方によって決定されます。同期モードでは、BRGHビットは無視されます。

表22-3に、ボーレートを決定するための式を示します。例22-1に、ボーレートとボーレート・エラーを決定するための計算例を示します。

さまざまな非同期モードの一般的なボーレートとエラー値は、便宜上計算されており、表22-3に示されています。ボーレート・エラーを減らすには、高いボーレート（BRGH = 1）または16ビットBRG（BRG16 = 1）を使用すると有利な場合があります。16ビットBRGモードは、高速発振器周波数の低速ボーレートを実現するために使用されます。

SPBRGH、SPBRGLレジスタペアに新しい値を書き込むと、BRGタイマーがリセット（またはクリア）されます。これにより、BRGは新しいボーレートを出力する前にタイマー・オーバーフローを待機しなくなります。

アクティブな受信操作中にシステム・クロックが変更されると、受信エラーまたはデータ損失が発生する可能性があります。この問題を回避するには、システム・クロックを変更する前に、RCIDLビットのステータスをチェックして、受信操作がアイドル（不動）状態であることを確認してください。

### 例22-1: ボーレート・エラーの計算

FOSCが16MHz、必要なボーレートが9600、非同期モード、8ビットBRGのデバイスの場合：

$$\text{希望するボーレート} = \frac{F_{OSC}}{64([SPBRGH:SPBRGL] + 1)}$$

SPBRGH:SPBRGLの解決：

$$X = \frac{\frac{F_{OSC}}{\text{希望するボーレート}}}{64} - 1$$

$$= \frac{\frac{16000000}{9600}}{64} - 1$$

$$= [25.042] = 25$$

$$\text{計算されたボーレート} = \frac{16000000}{64(25 + 1)}$$

$$= 9615$$

$$\text{Error} = \frac{\text{計算されたボーレート} - \text{希望するボーレート}}{\text{希望するボーレート}}$$

$$= \frac{(9615 - 9600)}{9600} = 0.16\%$$

表22-3: ポーレートの式

構成ビット			BRG/EUSARTモード	ポーレート式
SYNC	BRG16	BRGH		
0	0	0	8ビット/非同期	$F_{osc}/[64(n+1)]$
0	0	1	8ビット/非同期	$F_{osc}/[16(n+1)]$
0	1	0	16ビット/非同期	
0	1	1	16ビット/非同期	$F_{osc}/[4(n+1)]$
1	0	x	8ビット/同期	
1	1	x	16ビット/同期	

記録: x =気にしない、n = SPBRGHの値、SPBRGLレジスター・ペア。

表22-4: ポーレート・ジェネレーターに関連付けられたレジスターの要約

名前	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ページ上の登録
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	235
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	234
SPBRGL	BRG<7:0>								236*
SPBRGH	BRG<15:8>								236*
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	233

記録: — =実装されていない場所、「0」と読みます。網掛けのセルは、ポーレートジェネレーターには使用されません。

\* ページは登録情報を提供します。

# PIC16(L)F1508/9

表22-5: 非同期モードのポーレート

ポーレート	SYNC = 0、BRGH = 0、BRG16 = 0											
	Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 16.000 MHz			Fosc = 11.0592 MHz		
	実際のレート	% エラー	SPBRG 値 (10進)	実際のレート	% エラー	SPBRG 値 (10進)	実際のレート	% エラー	SPBRG 値 (10進)	実際のレート	% エラー	SPBRG 値 (10進)
300	—	—	—	—	—	—	—	—	—	—	—	—
1200	1221	1.73	255	1200	0.00	239	1202	0.16	207	1200	0.00	143
2400	2404	0.16	129	2400	0.00	119	2404	0.16	103	2400	0.00	71
9600	9470	-1.36	32	9600	0.00	29	9615	0.16	25	9600	0.00	17
10417	10417	0.00	29	10286	-1.26	27	10417	0.00	23	10165	-2.42	16
19.2k	19.53k	1.73	15	19.20k	0.00	14	19.23k	0.16	12	19.20k	0.00	8
57.6k	—	—	—	57.60k	0.00	7	—	—	—	57.60k	0.00	2
115.2k	—	—	—	—	—	—	—	—	—	—	—	—

ポーレート	SYNC = 0、BRGH = 0、BRG16 = 0											
	Fosc = 8.000 MHz			Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 1.000 MHz		
	実際のレート	% エラー	SPBRG 値 (10進)	実際のレート	% エラー	SPBRG 値 (10進)	実際のレート	% エラー	SPBRG 値 (10進)	実際のレート	% エラー	SPBRG 値 (10進)
300	—	—	—	300	0.16	207	300	0.00	191	300	0.16	51
1200	1202	0.16	103	1202	0.16	51	1200	0.00	47	1202	0.16	12
2400	2404	0.16	51	2404	0.16	25	2400	0.00	23	—	—	—
9600	9615	0.16	12	—	—	—	9600	0.00	5	—	—	—
10417	10417	0.00	11	10417	0.00	5	—	—	—	—	—	—
19.2k	—	—	—	—	—	—	19.20k	0.00	2	—	—	—
57.6k	—	—	—	—	—	—	57.60k	0.00	0	—	—	—
115.2k	—	—	—	—	—	—	—	—	—	—	—	—

ポーレート	SYNC = 0、BRGH = 1、BRG16 = 0											
	Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 16.000 MHz			Fosc = 11.0592 MHz		
	実際のレート	% エラー	SPBRG 値 (10進)	実際のレート	% エラー	SPBRG 値 (10進)	実際のレート	% エラー	SPBRG 値 (10進)	実際のレート	% エラー	SPBRG 値 (10進)
300	—	—	—	—	—	—	—	—	—	—	—	—
1200	—	—	—	—	—	—	—	—	—	—	—	—
2400	—	—	—	—	—	—	—	—	—	—	—	—
9600	9615	0.16	129	9600	0.00	119	9615	0.16	103	9600	0.00	71
10417	10417	0.00	119	10378	-0.37	110	10417	0.00	95	10473	0.53	65
19.2k	19.23k	0.16	64	19.20k	0.00	59	19.23k	0.16	51	19.20k	0.00	35
57.6k	56.82k	-1.36	21	57.60k	0.00	19	58.82k	2.12	16	57.60k	0.00	11
115.2k	113.64k	-1.36	10	115.2k	0.00	9	111.1k	-3.55	8	115.2k	0.00	5

表22-5: 非同期モードのポーレート(続き)

ポーレート	SYNC = 0、BRGH = 1、BRG16 = 0											
	Fosc = 8.000 MHz			Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 1.000 MHz		
	実際のレート	% エラー	SPBRG 値 (10進)	実際のレート	% エラー	SPBRG 値 (10進)	実際のレート	% エラー	SPBRG 値 (10進)	実際のレート	% エラー	SPBRG 値 (10進)
300	—	—	—	—	—	—	—	—	—	300	0.16	207
1200	—	—	—	1202	0.16	207	1200	0.00	191	1202	0.16	51
2400	2404	0.16	207	2404	0.16	103	2400	0.00	95	2404	0.16	25
9600	9615	0.16	51	9615	0.16	25	9600	0.00	23	—	—	—
10417	10417	0.00	47	10417	0.00	23	10473	0.53	21	10417	0.00	5
19.2k	19231	0.16	25	19.23k	0.16	12	19.2k	0.00	11	—	—	—
57.6k	55556	-3.55	8	—	—	—	57.60k	0.00	3	—	—	—
115.2k	—	—	—	—	—	—	115.2k	0.00	1	—	—	—

ポーレート	SYNC = 0、BRGH = 0、BRG16 = 1											
	Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 16.000 MHz			Fosc = 11.0592 MHz		
	実際のレート	% エラー	SPBRG 値 (10進)	実際のレート	% エラー	SPBRG 値 (10進)	実際のレート	% エラー	SPBRG 値 (10進)	実際のレート	% エラー	SPBRG 値 (10進)
300	300.0	-0.01	4166	300.0	0.00	3839	300.03	0.01	3332	300.0	0.00	2303
1200	1200	-0.03	1041	1200	0.00	959	1200.5	0.04	832	1200	0.00	575
2400	2399	-0.03	520	2400	0.00	479	2398	-0.08	416	2400	0.00	287
9600	9615	0.16	129	9600	0.00	119	9615	0.16	103	9600	0.00	71
10417	10417	0.00	119	10378	-0.37	110	10417	0.00	95	10473	0.53	65
19.2k	19.23k	0.16	64	19.20k	0.00	59	19.23k	0.16	51	19.20k	0.00	35
57.6k	56.818	-1.36	21	57.60k	0.00	19	58.82k	2.12	16	57.60k	0.00	11
115.2k	113.636	-1.36	10	115.2k	0.00	9	111.11k	-3.55	8	115.2k	0.00	5

ポーレート	SYNC = 0、BRGH = 0、BRG16 = 1											
	Fosc = 8.000 MHz			Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 1.000 MHz		
	実際のレート	% エラー	SPBRG 値 (10進)	実際のレート	% エラー	SPBRG 値 (10進)	実際のレート	% エラー	SPBRG 値 (10進)	実際のレート	% エラー	SPBRG 値 (10進)
300	299.9	-0.02	1666	300.1	0.04	832	300.0	0.00	767	300.5	0.16	207
1200	1199	-0.08	416	1202	0.16	207	1200	0.00	191	1202	0.16	51
2400	2404	0.16	207	2404	0.16	103	2400	0.00	95	2404	0.16	25
9600	2404	0.16	51	9615	0.16	25	9600	0.00	23	—	—	—
10417	10417	0.00	47	10417	0.00	23	10473	0.53	21	10417	0.00	5
19.2k	19.23k	0.16	25	19.23k	0.16	12	19.20k	0.00	11	—	—	—
57.6k	55556	-3.55	8	—	—	—	57.60k	0.00	3	—	—	—
115.2k	—	—	—	—	—	—	115.2k	0.00	1	—	—	—

# PIC16(L)F1508/9

表22-5: 非同期モードのポーレート(続き)

ポーレート	SYNC = 0、BRGH = 1、BRG16 = 1 または SYNC = 1、BRG16 = 1											
	Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 16.000 MHz			Fosc = 11.0592 MHz		
	実際のレート	% エラー	SPBRG 値 (10進)	実際のレート	% エラー	SPBRG 値 (10進)	実際のレート	% エラー	SPBRG 値 (10進)	実際のレート	% エラー	SPBRG 値 (10進)
300	300.0	0.00	16665	300.0	0.00	15359	300.0	0.00	13332	300.0	0.00	9215
1200	1200	-0.01	4166	1200	0.00	3839	1200.1	0.01	3332	1200	0.00	2303
2400	2400	0.02	2082	2400	0.00	1919	2399.5	-0.02	1666	2400	0.00	1151
9600	9597	-0.03	520	9600	0.00	479	9592	-0.08	416	9600	0.00	287
10417	10417	0.00	479	10425	0.08	441	10417	0.00	383	10433	0.16	264
19.2k	19.23k	0.16	259	19.20k	0.00	239	19.23k	0.16	207	19.20k	0.00	143
57.6k	57.47k	-0.22	86	57.60k	0.00	79	57.97k	0.64	68	57.60k	0.00	47
115.2k	116.3k	0.94	42	115.2k	0.00	39	114.29k	-0.79	34	115.2k	0.00	23

ポーレート	SYNC = 0、BRGH = 1、BRG16 = 1 または SYNC = 1、BRG16 = 1											
	Fosc = 8.000 MHz			Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 1.000 MHz		
	実際のレート	% エラー	SPBRG 値 (10進)	実際のレート	% エラー	SPBRG 値 (10進)	実際のレート	% エラー	SPBRG 値 (10進)	実際のレート	% エラー	SPBRG 値 (10進)
300	300.0	0.00	6666	300.0	0.01	3332	300.0	0.00	3071	300.1	0.04	832
1200	1200	-0.02	1666	1200	0.04	832	1200	0.00	767	1202	0.16	207
2400	2401	0.04	832	2398	0.08	416	2400	0.00	383	2404	0.16	103
9600	9615	0.16	207	9615	0.16	103	9600	0.00	95	9615	0.16	25
10417	10417	0	191	10417	0.00	95	10473	0.53	87	10417	0.00	23
19.2k	19.23k	0.16	103	19.23k	0.16	51	19.20k	0.00	47	19.23k	0.16	12
57.6k	57.14k	-0.79	34	58.82k	2.12	16	57.60k	0.00	15	—	—	—
115.2k	117.6k	2.12	16	111.1k	3.55	8	115.2k	0.00	7	—	—	—

## 22.4.1 自動ポー検出

EUSARTモジュールは、ボーレートの自動検出とキャリブレーション（校正）をサポートしています。

オート・ポー・ディテクト（自動ポー検出。ABD）モードでは、BRGへのクロックが逆になります。BRGが着信RX信号をクロックするのではなく、RX信号がBRGのタイミングを計っています。ボーレート・ジェネレーターは、LINバスの同期文字である受信55h（ASCII「U」）の期間を計測するために使用されます。このキャラクターのユニークな特徴は、ストップ・ビット・エッジを含む5つの立ち上がりエッジがあることです。

BAUDCONレジスターのABDENビットをセットすると、自動ポー校正シーケンスが開始されます（図22-6）。ABDシーケンスが実行されている間、EUSARTステート・マシンはアイドル（不動）状態に保たれます。表22-6に示すように、受信ラインの最初の立ち上がりエッジで、スタート・ビットの後、SPBRGはBRGカウンター・クロックを使用してカウント・アップを開始します。5番目の立ち上がりエッジは、8番目のビット期間の終わりにRXピンで発生します。その際、SPBRGH、SPBRGLレジスター・ペアには適切なBRG周期を合計した累積値が残り、ABDENビットは自動的にクリアされ、RCIF割り込みフラグがセットされます。RCIF割り込みをクリアするには、RCREGの値を読み取る必要があります。RCREG内容は破棄する必要があります。SPBRGHレジスターを使用しないモードを校正する場合、ユーザーはSPBRGHレジスターの00hをチェックすることにより、SPBRGLレジスターがオーバーフローしていないことを確認できます。

表22-6に示すように、BRG自動ポー・クロックはBRG16ビットとBRGHビットによって決定されます。ABD中、SPBRGH

レジスターとSPBRGLレジスターの両方が、BRG16ビット設定とは関係なく16ビット・カウンターとして使用されます。ボーレート周期の校正中、SPBRGHおよびSPBRGLレジスターはBRGベース・クロック・レートの1/8でクロックされます。結果のバイト測定値は、フルスピードでクロックされたときの平均ビット時間です。

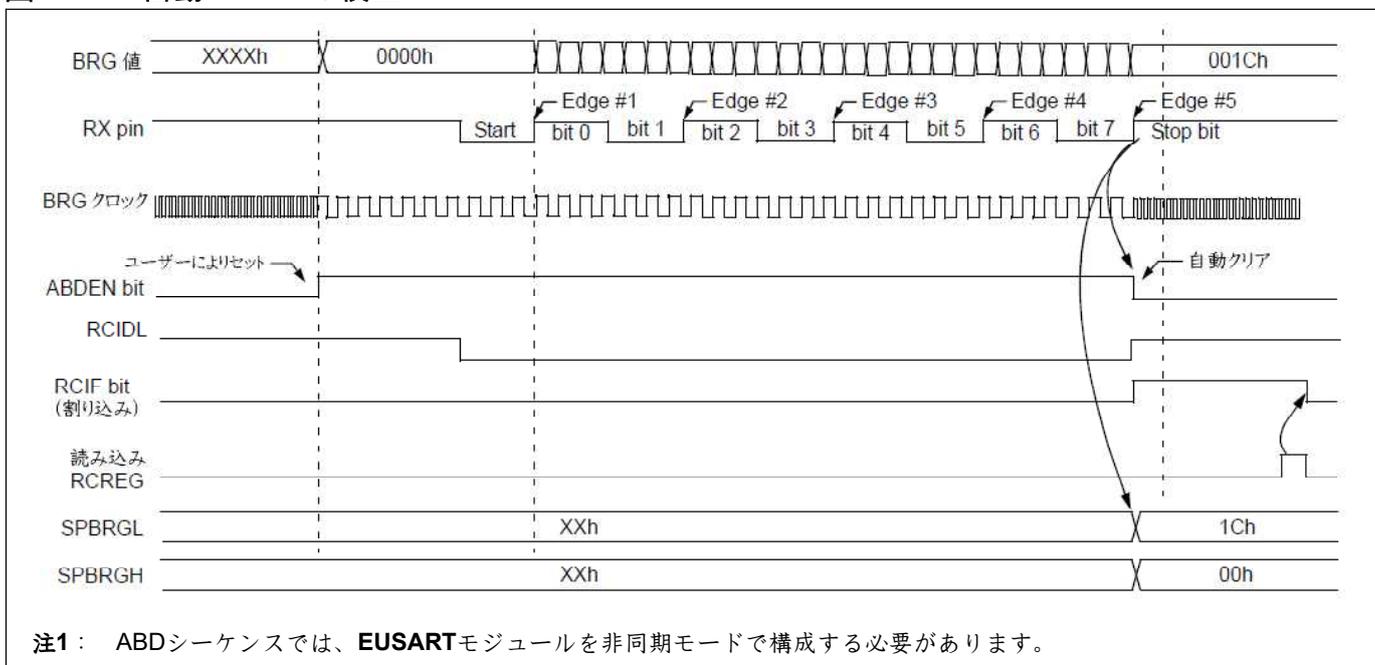
- 注1： WUEビットがABDENビットで設定されている場合、自動ポー検出はブレイク文字に続くバイトで発生します（セクション22.4.3「ブレイク時の自動ウェイクアップ」を参照）。
- 2： 着信文字のボーレートが、選択したBRGクロック・ソースの範囲内にあるかどうかを判断するのはユーザーの責任です。発振器周波数とEUSARTボーレートのいくつかの組み合わせは不可能です。
- 3： 自動ポー処理中、自動ポー・カウンターは1からカウントを開始します。自動ポーシーケンスが完了すると、最大の精度を実現するために、SPBRGH：SPBRGLレジスター・ペアから1を減算します。

表22-6: BRGカウンター・クロック・レート

BRG16	BRGH	BRG 基本クロック	BRG ABD クロック
0	0	Fosc/64	Fosc/512
0	1	Fosc/16	Fosc/128
1	0	Fosc/16	Fosc/128
1	1	Fosc/4	Fosc/32

注釈： ABDシーケンス中、SPBRGLおよびSPBRGHレジスターは、BRG16設定とは関係なく、両方とも16ビット・カウンターとして使用されます。

図22-6: 自動ポーレート校正



## 22.4.2 自動ポー・オーバーフロー

自動ポー検出の過程で、RXピンで5番目の立ち上がりエッジが検出される前にポーレート・カウンターがオーバーフローすると、BAUDxCONレジスターのABDOVFビットがセットされます。ABDOVFビットは、カウンターがSPxBRGH:SPxBRGLレジスター・ペアの16ビットに収まる最大カウントを超えたことを示します。オーバーフロー条件により、RCIFフラグが設定されます。RXピンで5番目の立ち上がりエッジが検出されるまで、カウンタはカウントを続けます。RCIDLビットは5番目の立ち上がりエッジまでfalse（「0」）のままになり、5番目の立ち上がりエッジでRCIDLビットがセットされます。オーバーフローが発生した後、5番目の立ち上がりエッジの前にRCREGが読み取られた場合、5番目の立ち上がりエッジによってRCIFが再度設定されます。

オーバーフロー状態をクリアするために自動ポー・プロセス（過程）を早期に終了すると、同期文字の5番目の立ち上がりエッジを適切に検出できなくなります。ABDENビットがクリアされたときに同期文字の立ち下がりエッジがまだ発生していない場合、それらはスタート・ビットとして誤って検出されます。オーバーフロー状態をクリアするには、次の手順をお勧めします：

1. RCREGを読み取って、RCIFをクリアします。
2. RCIDLがゼロの場合は、RCIFを待つて、ステップ1を繰り返します。
3. ABDOVFビットをクリアします。

## 22.4.3 ブ레이크（休憩）での自動ウェイクアップ

スリープ・モード中、EUSARTへのすべてのクロックが一時停止されます。このため、ポーレート・ジェネレーターは非アクティブであり、適切な文字受信を実行できません。自動ウェイクアップ機能を使用すると、RX/DT回線のアクティビティ（活動）によってコントローラーをウェイクアップできます。この機能は、非同期モードでのみ使用できます。

自動ウェイクアップ機能は、BAUDCONレジスターのWUEビットをセットすることで有効になります。設定されると、RX/DTの通常の受信シーケンスは無効になり、EUSARTはアイドル（不動）状態のままになり、CPUモードに関係なくウェイクアップ・イベントを監視します。ウェイクアップ・イベントは、RX/DTラインのHighからLowへの遷移で構成されます。（これは、同期ブレークの開始またはLINプロトコルのウェイクアップ信号文字と一致します。）

EUSARTモジュールは、ウェイクアップ・イベントと同時にRCIF割り込みを生成します。割り込みは、通常のCPU動作モードではQクロックに同期して生成され（図22-7）、デバイスがスリープ・モードの場合は非同期に生成されます（図22-8）。割り込み条件は、RCREGレジスターを読み取ることによってクリアされます。

WUEビットは、ブレークの終了時にRXラインのLowからHighへの遷移によって自動的にクリアされます。これは、ブレーク・イベントが終了したことをユーザーに通知します。この時点で、EUSARTモジュールはアイドル・モードになり、次の文字を受信するのを待っています。

## 22.4.3.1 特別な考慮事項

### ブレーク・キャラクター

ウェイクアップ・イベント中の文字エラーまたは文字フラグメントを回避するには、ウェイクアップ文字をすべてゼロにする必要があります。

ウェイクアップが有効になっている場合、この機能はデータ流れの低時間とは関係なく機能します。WUEビットが設定され、有効なゼロ以外の文字が受信された場合、スタート・ビットから最初の立ち上がりエッジまでの低時間はウェイクアップ・イベントとして解釈されます。文字の残りのビットは断片化された文字として受信され、後続の文字はフレーミング（作成）またはオーバーラン・エラーを引き起こす可能性があります。

したがって、送信の最初の文字はすべて「0」である必要があります。これは10またはビット時間、LINバスに推奨される13ビット時間、または、標準のRS-232デバイスの場合は任意のビット時間以上である必要があります。

### 発信器セットアップ時間

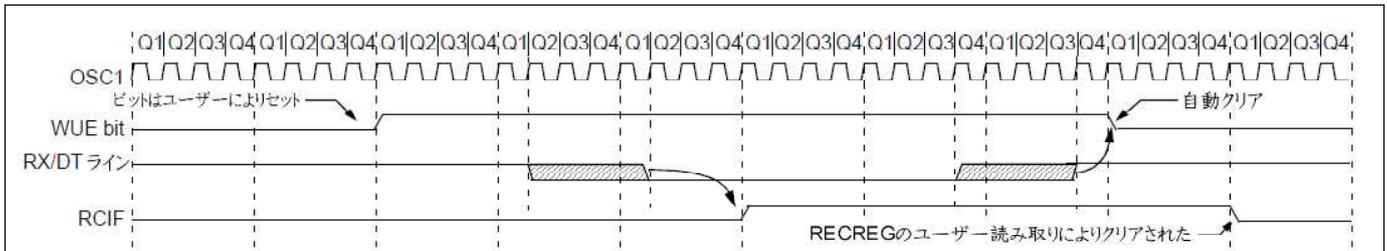
特に起動間隔が長い発振器を使用するアプリケーション（LP、XT、HS/PLLモードなど）では、発振器の起動時間を考慮する必要があります。同期ブレーク（またはウェイクアップ信号）文字は、選択した発信器が起動してEUSARTの適切な初期化を提供するのに十分な時間を与えるために、十分な長さであり、その後十分な間隔が続く必要があります。

### WUEビット

ウェイクアップ・イベントは、RCIFビットをセットすることによって受信割り込みを引き起こします。WUEビットは、RX/DTの立ち上がりエッジによってハードウェアでクリアされます。次に、ソフトウェアでRCREGレジスターを読み取り、その内容を破棄することにより、割り込み条件がクリアされます。

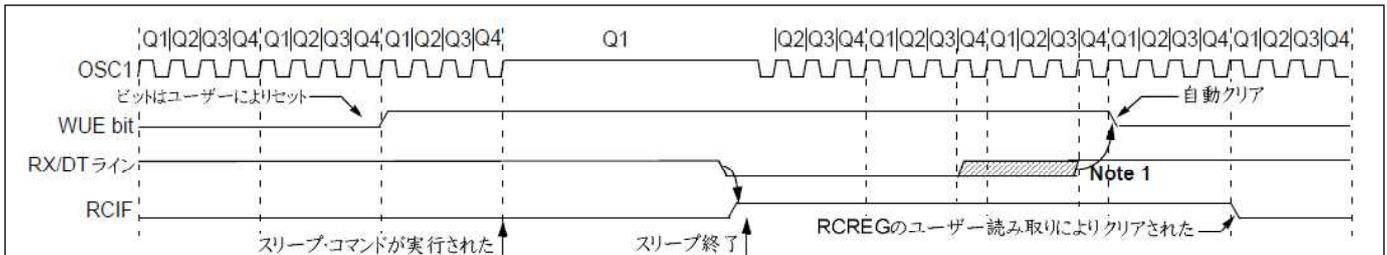
実際のデータが失われないようにするには、WUEビットを設定する前に、RCIDLビットをチェックして、受信操作が処理中でないことを確認します。受信動作が発生していない場合は、スリープ・モードに入る直前にWUEビットが設定されることがあります。

図22-7: 通常の操作中の自動ウェイクアップ・ビット(WUE)のタイミング



注1: WUEビットが設定されている間、EUSARTはアイドル状態のままです。

図22-8: スリープ中の自動ウェイクアップ・ビット(WUE)のタイミング



注1: ウェイクアップイベントに長い発信器のウォームアップ時間が必要な場合、**stposc**信号がまだアクティブなときにWUEビットの自動クリアが発生する可能性があります。このシーケンスは、Qクロックの存在に依存するべきではありません。

2: WUEビットが設定されている間、EUSARTはアイドル状態のままです。

## 22.4.4 ブレーク文字シーケンス

EUSARTモジュールには、LINバス規格で必要とされる特別なブレーク・キャラクター（文字）シーケンスを送信する機能があります。ブレーク・キャラクターは、スタート・ビットとそれに続く12個の「0」ビットおよびストップ・ビットで構成されます。

ブレーク文字を送信するには、TXSTAレジスタのSENDBビットとTXENビットを設定します。次に、ブレーク文字の送信は、TXREGへの書き込みによって開始されます。TXREGに書き込まれたデータの値は無視され、すべての「0」が送信されます。

SENDBビットは、対応するストップ・ビットが送信された後、ハードウェアによって自動的にリセットされます。これにより、ユーザーは、ブレーク文字（通常、LIN仕様の同期文字）に続く次の送信バイトで送信FIFOをプリロード（先読み）できます。

TXSTAレジスタのTRMTビットは、通常、送信時と同様に、送信操作がアクティブまたはアイドルのときを示します。ブレーク文字シーケンスのタイミングについては、図22-9を参照してください。

### 22.4.4.1 ブレークと同期送信シーケンス

次のシーケンスは、bブレークで構成されるメッセージ・フレーム・ヘッダーを開始し、その後、自動ポー同期バイトが続きます。このシーケンスは、LINバス・マスターの典型的なものです。

1. EUSARTを目的のモードに設定します。
2. TXENビットとSENDBビットを設定して、ブレーク・シーケンスを有効にします。
3. TXREGにダミー文字をロードして送信を開始します（値は無視されます）。
4. TXREGに「55h」を書き込んで、同期文字を送信FIFOバッファにロードします。
5. ブレークが送信された後、SENDBビットがハードウェアによってリセットされ、同期文字が送信されます。

TXIFで示されるように、TXREGが空になると、次のデータバイトをTXREGに書き込むことができます。

## 22.4.5 ブレーク文字を受信する

拡張EUSARTモジュールは、2つの方法でブレーク文字を受け取ることができます。

ブレーク文字を検出する最初の方法は、RCSTAレジスタのFERRビットとRCREGで示される受信データを使用します。ポーレート・ジェネレーターは、予想されるポーレートに初期化されていると想定されます。

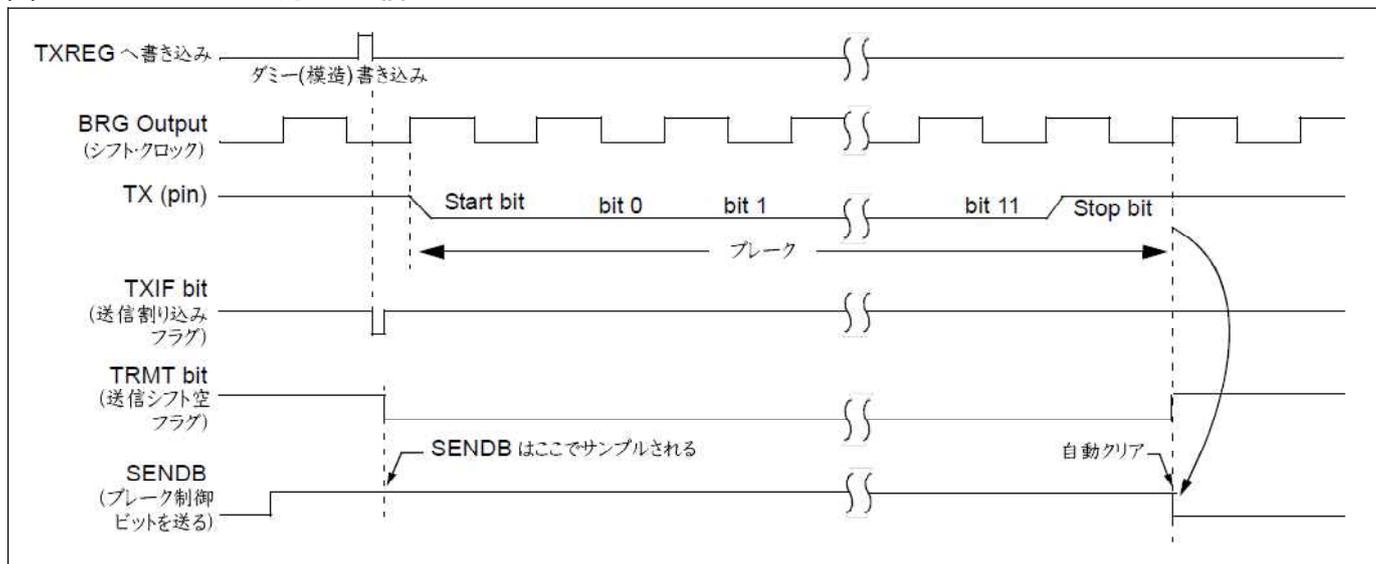
ブレーク文字は次の場合に受信されます；

- RCIFビットはセット
- FERRビットはセット
- RCREG = 00h

2番目の方法は、セクション22.4.3「ブレーク時の自動ウェイクアップ」で説明されている自動ウェイクアップ機能を使用します。この機能を有効にすることにより、EUSARTはRX/DTで次の2つの遷移をサンプリングし、RCIF割り込みを発生させ、次のデータ・バイトとそれに続く別の割り込みを受信します。

ブレーク文字に続いて、ユーザーは通常、自動ポー検出機能を有効にする必要があることに注意してください。どちらの方法でも、ユーザーはEUSARTをスリープ・モードにする前にBAUDCONレジスタのABDENビットを設定できます。

図22-9: ブレーク文字の送信シーケンス



## 22.5 EUSART同期モード

同期シリアル通信は通常、単一のマスターと1つ以上のスレーブを備えたシステムで使用されます。マスター・デバイスには、ボーレート生成に必要な回路が含まれており、システム内のすべてのデバイスにクロックを供給します。スレーブ・デバイスは、内部クロック生成回路を排除することにより、マスター・クロックを利用できます。

同期モードには、双方向データ・ラインとクロック・ラインの2つの信号ラインがあります。スレーブは、マスターから供給される外部クロックを使用して、シリアル・データをそれぞれの受信および送信シフト・レジスタにシフトインおよびシフトアウトします。データ・ラインは双方向であるため、同期動作は半二重のみです。半二重とは、マスター・デバイスとスレーブ・デバイスがデータを送受信できるが、両方を同時に受信することはできないという事実を指します。EUSARTは、マスター・デバイスまたはスレーブ・デバイスとして動作できます。

スタート・ビットとストップ・ビットは同期送信では使用されません。

### 22.5.1 同期マスター・モード

以下のビットは、同期マスター動作にEUSARTを構成するために使用されます：

- SYNC = 1
- CSRC = 1
- SREN = 0 (送信用) ; SREN = 1 (受信用)
- CREN = 0 (送信用) ; CREN = 1 (受信用)
- SPEN = 1

TXSTAレジスタのSYNCビットを設定すると、デバイスが同期動作するように構成されます。TXSTAレジスタのCSRCビットを設定すると、デバイスがマスターとして設定されます。RCSTAレジスタのSRENビットとCRENビットをクリアすると、デバイスが送信モードになります。クリアしないと、デバイスは受信するように構成されます。RCSTAレジスタのSPENビットをセットすると、EUSARTが有効になります。

#### 22.5.1.1 マスター・クロック

同期データ転送は、データと同期する別個のクロック・ラインを使用します。マスターとして構成されたデバイスは、TX/CKラインでクロックを送信します。EUSARTが同期送信または受信動作に設定されている場合、TX/CKピン出カドライバは自動的に有効になります。シリアルデータ・ビットは、各クロックの立ち下がりエッジで有効になるように、読み込みエッジで変更されます。データ・ビットごとに1クロック・サイクルが生成されます。データ・ビットと同じ数のクロック・サイクルのみが生成されます。

#### 22.5.1.2 クロック極性

Microwireとの互換性のために、クロック極性オプションが提供されています。クロックの極性は、BAUDCONレジスタのSCKPビットで選択されます。SCKPビットを設定すると、クロックのアイドル状態がHighに設定されます。SCKPビットをセットすると、各クロックの立ち下がりエッジでデータが変化します。

SCKPビットをクリアすると、アイドル状態がLowに設定されます。SCKPビットがクリアされると、各クロックの立ち上がりエッジでデータが変化します。

#### 22.5.1.3 同期マスター送信

RX/DTピンでデバイスからデータが転送されます。EUSARTが同期マスター送信動作に構成されている場合、RX/DTおよびTX/CKピン出カドライバは自動的に有効になります。

TXREGレジスタに文字を書き込むことにより、送信が開始されます。TSRに前の文字の全部または一部がまだ含まれている場合、前の文字の最後のビットが送信されるまで、新しい文字データがTXREGに保持されます。これが最初の文字である場合、または前の文字がTSRから完全にフラッシュ（追い出す）されている場合、TXREG内のデータはすぐにTSRに転送されます。キャラクター（文字）の送信は、TXREGからTSRへのデータ転送の直後に開始されます。

各データ・ビットはマスター・クロックの読み込みエッジで変化し、次の読み込みクロック・エッジまで有効です。

**注釈：** TSRレジスタはデータ・メモリにマップされていないため、ユーザーは使用できません。

#### 22.5.1.4 同期マスター送信セットアップ:

1. SPBRGH、SPBRGLレジスタ・ペア、およびBRGHビットとBRG16ビットを初期化して、目的のボーレートを実現します（セクション22.4「EUSARTボーレート・ジェネレーター（BRG）」を参照）。
2. ビットSYNC、SPEN、およびCSRCを設定して、同期マスター・シリアル・ポートを有効にします。
3. ビットSRENおよびCRENをクリアして、受信モードを無効にします。
4. TXENビットを設定して、送信モードを有効にします。
5. 9ビット送信が必要な場合は、TX9ビットを設定します。
6. 割り込みが必要な場合は、PIE1レジスタのTXIEビットとINTCONレジスタのGIEおよびPEIEビットを設定します。
7. 9ビット送信を選択した場合、9番目のビットをTX9Dビットにロードする必要があります。
8. TXREGレジスタにデータをロードして送信を開始します。

# PIC16(L)F1508/9

図22-10: 同期送信

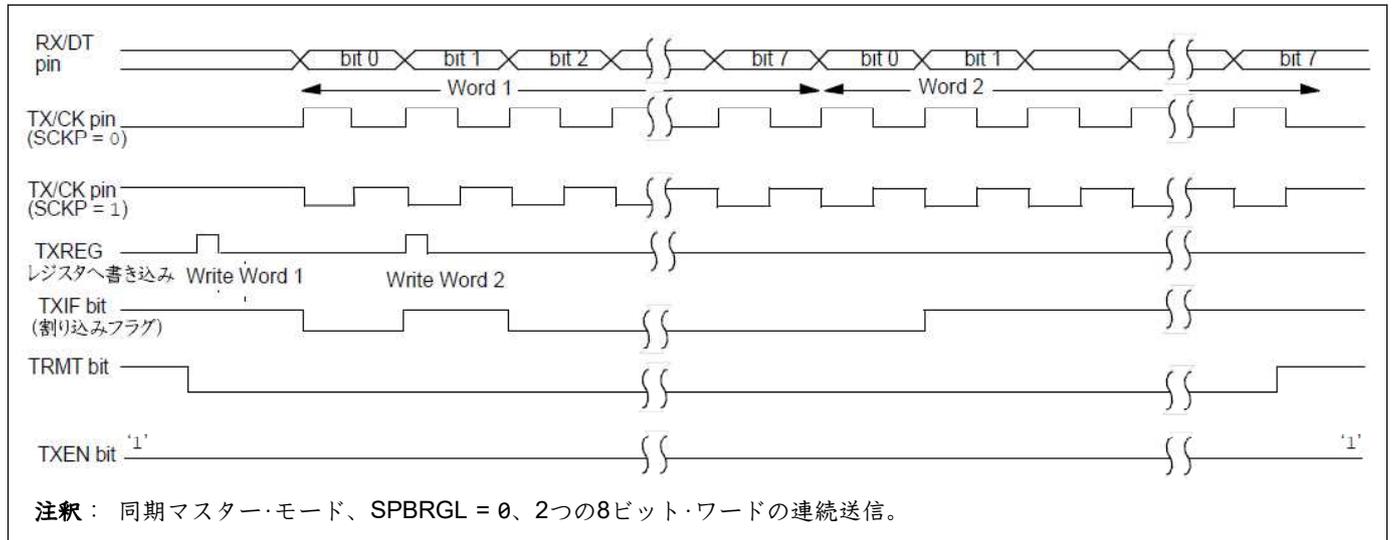


図22-11: 同期送信 (TXENを介して)

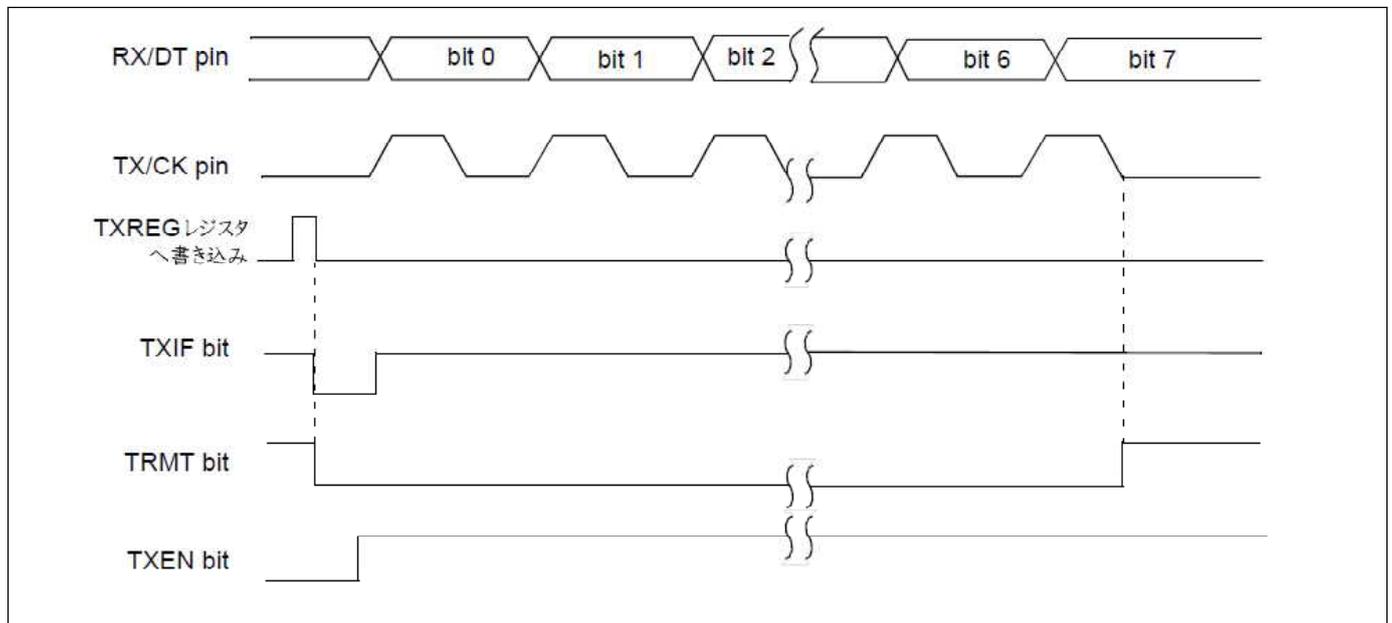


表22-7: 同期マスター送信に関連付けられたレジスタの要約

名前	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ページ上の登録
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	235
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	75
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	—	TMR2IE	TMR1IE	76
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	—	TMR2IF	TMR1IF	79
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	234
SPBRGL	BRG<7:0>								236*
SPBRGH	BRG<15:8>								236*
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	113
TXREG	EUSART送信データレジスター								225*
TXSTA	CSRC	TX9	TXEN	SYNC	SEnDB	BRGH	TRMT	TX9D	233

記録: — =実装されていない場所、「0」と読みます。網掛けのセルは、同期マスター送信には使用されません。

\* ページは登録情報を提供します。

### 22.5.1.5 同期マスター受信

RX/DTピンでデータを受信します。EUSARTが同期マスター受信動作に設定されている場合、RX/DTピン出カドライバは自動的に無効になります。

同期モードでは、シングル受信許可ビット（RCSTAレジスタのSREN）または連続受信許可ビット（RCSTAレジスタのCREN）のいずれかを設定することにより、受信が有効になります。

SRENが設定され、CRENがクリアされている場合、1文字のデータ・ビットと同じ数のクロック・サイクルのみが生成されます。SRENビットは、1文字が完了すると自動的にクリアされます。CRENが設定されると、CRENがクリアされるまでクロックが継続的に生成されます。文字の途中でCRENがクリアされると、CKクロックはすぐに停止し、部分的な文字は破棄されます。SRENとCRENの両方が設定されている場合、SRENは最初の文字の完了時にクリアされ、CRENが優先されます。

受信を開始するには、SRENまたはCRENのいずれかを設定します。データは、TX/CKクロック・ピンの立ち下がりエッジのRX/DTピンでサンプリングされ、受信シフト・レジスタ（RSR）にシフトされます。完全な文字がRSRに受信されると、RCIFビットがセットされ、文字は2文字の受信FIFOに自動的に転送されます。受信FIFOの先頭文字の最下位8ビットはRCREGで使用できます。受信FIFOに未読文字がある限り、RCIFビットはセットされたままです。

**注釈：** RX/DT機能がアナログ・ピンにある場合、受信器が機能するには、対応するANSELビットをクリアする必要があります。

### 22.5.1.6 スレーブ・クロック

同期データ転送は、データと同期する別個のクロック・ラインを使用します。スレーブとして設定されたデバイスは、TX/CKラインでクロックを受信します。デバイスが同期スレーブ送信または受信動作に構成されている場合、TX/CKピン出カドライバは自動的に無効になります。シリアルデータビットは、各クロックの立ち下がりエッジで有効になるように、読み込みエッジで変更されます。クロック・サイクルごとに1つのデータ・ビットが転送されます。データ・ビットと同じ数のクロック・サイクルのみを受信する必要があります。

**注釈：** デバイスがスレーブとして構成されており、TX/CK機能がアナログ・ピン上にある場合、対応するANSELビットをクリアする必要があります。

### 22.5.1.7 受信オーバーラン・エラー

受信FIFOバッファは2文字を保持できます。FIFOにアクセスするためにRCREGが読み取られる前に、3番目の文字全体が受信されると、オーバーラン・エラーが生成されます。こ

れが発生すると、RCSTAレジスタのOERRビットがセットされます。FIFO内の以前のデータは上書きされません。FIFOバッファ内の2文字は読み取ることができますが、エラーがクリアされるまで追加の文字は受信されません。OERRビットは、オーバーラン状態をクリアすることによってのみクリアできます。SRENビットがセットされてCRENがクリアされているときにオーバーラン・エラーが発生した場合は、RCREGを読み取ることでエラーがクリアされます。CRENビットがセットされているときにオーバーランが発生した場合、エラー状態は、RCSTAレジスタのCRENビットをクリアするか、EUSARTをリセットするSPENビットをクリアすることによってクリアされます。

### 22.5.1.8 9ビット・キャラクター(文字)を受信する

EUSARTは9ビットの文字受信をサポートしています。RCSTAレジスタのRX9ビットが設定されると、EUSARTは受信した各文字のRSRに9ビットをシフトします。RCSTAレジスタのRX9Dビットは、受信FIFOの最上位の未読文字の9番目で最も重要なデータ・ビットです。受信FIFOバッファから9ビット・データを読み取る場合、RCREGから8つの最下位ビットを読み取る前に、RX9Dデータ・ビットを読み取る必要があります。

### 22.5.1.9 同期マスター受信セットアップ:

1. 適切なボーレートのSPBRGH、SPBRGLレジスタ・ペアを初期化します。必要に応じて、BRGHビットとBRG16ビットを設定またはクリアして、目的のボーレートを実現します。
2. RXピンのANSELビットをクリアします（該当する場合）。
3. ビットSYNC、SPEN、およびCSRCを設定して、同期マスター・シリアル・ポートを有効にします。
4. ビットCRENとSRENがクリアされていることを確認します。
5. 割り込みが必要な場合は、PIE1レジスタのRCIEビットとINTCONレジスタのGIEおよびPEIEビットを設定します。
6. 9ビットの受信が必要な場合は、ビットRX9を設定します。
7. SRENビットを設定して受信を開始するか、連続受信の場合はCRENビットを設定します。
8. 文字の受信が完了すると、割り込みフラグ・ビットRCIFがセットされます。許可ビットRCIEが設定されている場合、割り込みが生成されます。
9. RCSTAレジスタを読み取って9番目のビットを取得し（有効な場合）、受信中にエラーが発生したかどうかを判断します。
10. RCREGレジスタを読み取って、8ビットの受信データを読み取ります。
11. オーバーラン・エラーが発生した場合は、RCSTAレジスタのCRENビットをクリアするか、EUSARTをリセットするSPENビットをクリアしてエラーをクリアします。

# PIC16(L)F1508/9

図22-12: 同期受信(マスター・モード、SREN)

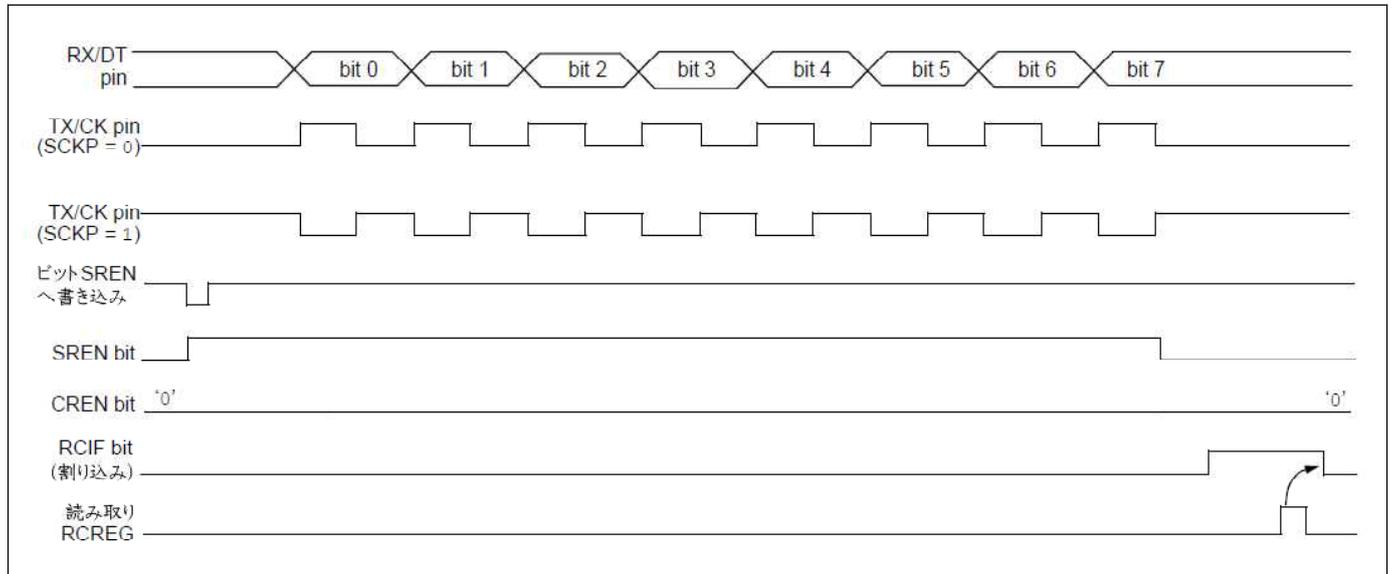


表22-8: 同期マスター受信に関連付けられたレジスタの要約

名前	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ページ上の登録
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	235
INTCON	GIE	PEIE	TMR0IE	INTE	IOCF	TMR0IF	INTF	IOCF	75
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	—	TMR2IE	TMR1IE	76
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	—	TMR2IF	TMR1IF	79
RCREG	EUSART受信データ・レジスタ								228*
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	234
SPBRGL	BRG<7:0>								236*
SPBRGH	BRG<15:8>								236*
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	113
TXSTA	CSRC	TX9	TXEN	SYNC	SENCB	BRGH	TRMT	TX9D	233

記録: — =実装されていない場所、「0」と読みます。網掛けのセルは、同期マスター受信には使用されません。

\* ページは登録情報を提供します。

## 22.5.2 同期スレーブ・モード

以下のビットは、同期スレーブ動作用にEUSARTを構成するために使用されます：

- SYNC = 1
- CSRC = 0
- SREN = 0 (送信用) ; SREN = 1 (受信用)
- CREN = 0 (送信用) ; CREN = 1 (受信用)
- SPEN = 1

TXSTAレジスタのSYNCビットを設定すると、デバイスが同期動作するように構成されます。TXSTAレジスタのCSRCビットをクリアすると、デバイスがスレーブとして設定されます。RCSTAレジスタのSRENビットとCRENビットをクリアすると、デバイスが送信モードになります。クリアしないと、デバイスは受信するように構成されます。RCSTAレジスタのSPENビットをセットすると、EUSARTが有効になります。

### 22.5.2.1 EUSART同期スレーブ送信

同期マスター・モードとスレーブ・モードの動作は、スレーブ・モードの場合を除いて同じです（セクション22.5.1.3「同期マスター送信」を参照）。

TXREGに2ワードを書き込んでからSLEEP命令を実行すると、次のようになります：

1. 最初の文字はすぐにTSRレジスターに転送され、送信されます。
2. 2番目のワードはTXREGレジスターに残ります。
3. TXIFビットは設定されません。
4. 最初の文字がTSRからシフトアウトされた後、TXREGレジスターは2番目の文字をTSRに転送し、TXIFビットが設定されます。
5. PEIEビットとTXIEビットが設定されている場合、割り込みはデバイスをスリープから復帰させ、次の命令を実行します。GIEビットも設定されている場合、プログラムは割り込みサービス・ルーチンを呼び出します。

### 22.5.2.2 同期スレーブ送信セットアップ:

1. SYNCビットとSPENビットを設定し、CSRCビットをクリアします。
2. CKピンのANSELビットをクリアします（該当する場合）。
3. CRENビットとSRENビットをクリアします。
4. 割り込みが必要な場合は、PIE1レジスタのTXIEビットとINTCONレジスタのGIEおよびPEIEビットを設定します。
5. 9ビット送信が必要な場合は、TX9ビットを設定します。
6. TXENビットを設定して、送信を有効にします。
7. 9ビット送信が選択されている場合は、最上位ビットをTX9Dビットに挿入します。
8. 最下位8ビットをTXREGレジスターに書き込んで送信を開始します。

表22-9: 同期スレーブ送信に関連付けられたレジスタの概要

名前	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ページ上の登録
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	235
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	75
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	—	TMR2IE	TMR1IE	76
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	—	TMR2IF	TMR1IF	79
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	234
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	113
TXREG	EUSART送信データ・レジスター								225*
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	233

記録： — =未実装の場所、「0」として読み取ります。網掛けのセルは同期スレーブ送信には使用されません。

\* ページは登録情報を提供します。

## 22.5.2.3 EUSART同期スレーブ受信

同期マスター・モードとスレーブ・モードの動作は同じです（セクション22.5.1.5「同期マスター受信」）。ただし、次の例外があります：

- スリープ
- CRENビットは常にセットされているため、受信器がアイドル状態になることはありません
- SRENビット、これはスレーブモードでは「ドントケア（気にするな）」です

スリープに入る前にCRENビットを設定することにより、スリープ・モード中に文字を受信することができます。ワードが受信されると、RSRレジスタはデータをRCREGレジスタに転送します。RCIE許可ビットがセットされている場合、生成された割り込みはデバイスをスリープから復帰させ、次の命令を実行します。GIEビットも設定されている場合、プログラムは割り込みベクターに分岐します。

## 22.5.2.4 同期スレーブ受信セットアップ:

1. SYNCビットとSPENビットを設定し、CSRCビットをクリアします。
2. CKピンとDTピンの両方のANSELビットをクリアします（該当する場合）。
3. 割り込みが必要な場合は、PIE1レジスタのRCIEビットとINTCONレジスタのGIEおよびPEIEビットを設定します。
4. 9ビットの受信が必要な場合は、RX9ビットを設定します。
5. CRENビットをセットして受信を有効にします。
6. 受信が完了すると、RCIFビットがセットされます。RCIEビットが設定されている場合、割り込みが生成されます。
7. 9ビットモードが有効になっている場合は、RCSTAレジスタのRX9Dビットから最上位ビットを取得します。
8. RCREGレジスタを読み取って、受信FIFOから8つの最下位ビットを取得します。
9. オーバーラン・エラーが発生した場合は、RCSTAレジスタのCRENビットをクリアするか、EUSARTをリセットするSPENビットをクリアしてエラーをクリアします。

表22-10: 同期スレーブ受信に関連付けられたレジスタの要約

名前	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ページ上の登録
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	235
INTCON	GIE	PEIE	TMR0IE	INTE	IOCF	TMR0IF	INTF	IOCF	75
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	—	TMR2IE	TMR1IE	76
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	—	TMR2IF	TMR1IF	79
RCREG	EUSART受信データレジスタ								228*
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	234
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	113
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	233

記録： — =実装されていない場所、「0」と読みます。網掛けのセルは同期スレーブ受信には使用されません。

\* ページは登録情報を提供します。

## 23.0 パルス・ワイズ・モジュレーション (PWM) モジュール

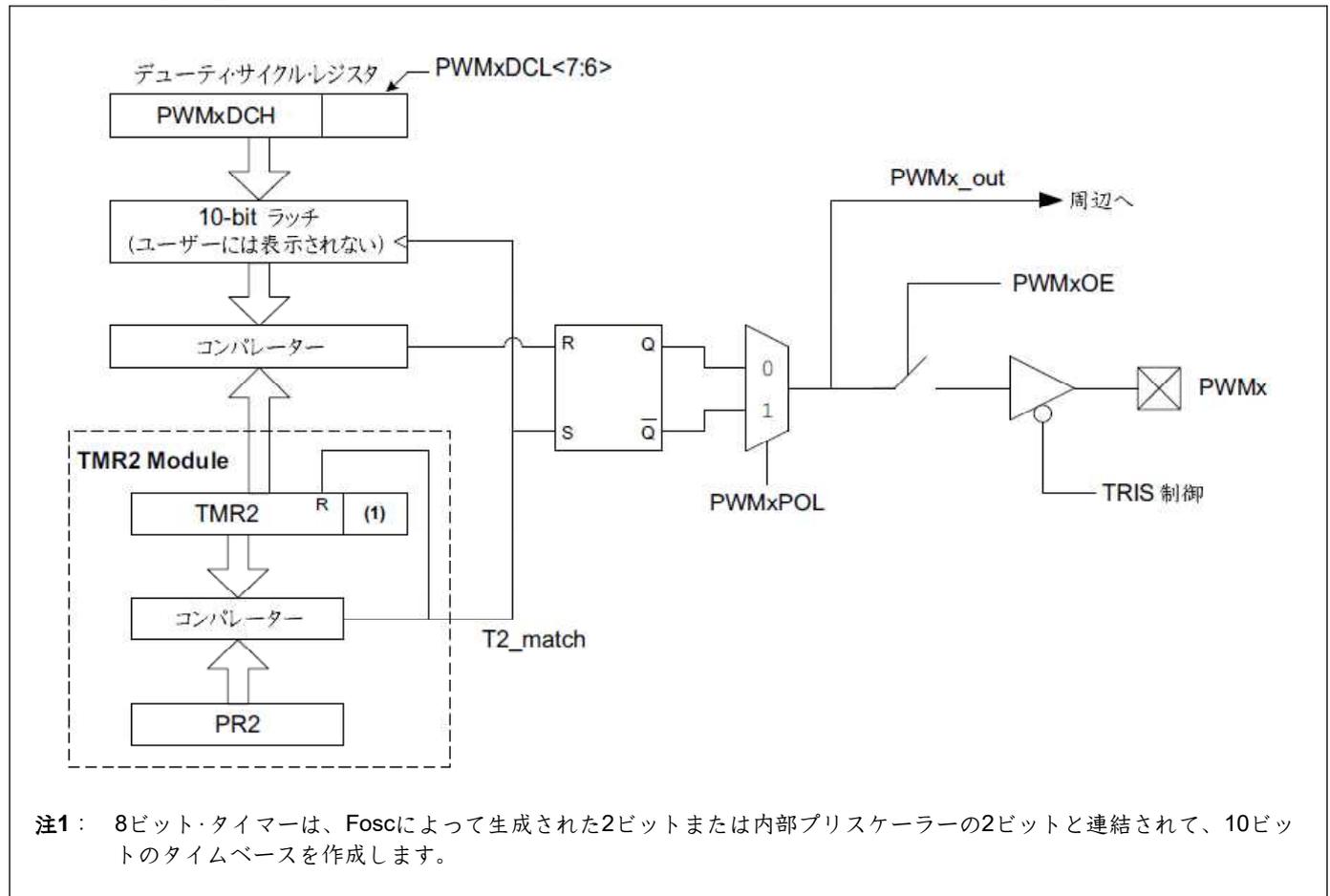
PWMモジュールは、次のレジスタによって構成されるデューティ・サイクル、周期、および分解能によって決定されるパルス幅変調信号を生成します：

- PR2
- T2CON
- PWMxDCH
- PWMxDCL
- PWMxCON

図23-1に、PWM動作の簡略ブロック図を示します。

このモジュールをPWM動作にセットアップする方法のステップ・バイ・ステップの手順については、セクション23.1.9「PWMxピンを使用したPWM動作のセットアップ」を参照してください。

図23-1: 簡略化されたPWMブロック図



## 23.1 PWMxピン構成

すべてのPWM出力はPORTデータ・ラッチで多重化されます。ユーザーは、関連するTRISビットをクリアして、ピンを出力として構成する必要があります。

**注釈：** PWMxOEビットをクリアすると、PWMxピンの制御が放棄されます。

### 23.1.1 基本的な動作

PWMモジュールは10ビットの分解能の出力を生成します。Timer2とPR2はPWMの周期を設定します。PWMxDCLおよびPWMxDCHレジスタは、デューティ・サイクルを構成します。周期はすべてのPWMモジュールに共通ですが、デューティ・サイクルは独立して制御されます。

**注釈：** Timer2ポスト・スケーラーはPWM周波数の決定には使用されません。ポストスケーラを使用して、PWM出力とは異なる周波数でサーボ更新レートを設定できます。

Timer2に関連するすべてのPWM出力は、TMR2がクリアされると設定されます。TMR2が対応するPWMxDCH (8 MSb) およびPWMxDCL<7:6> (2 LSb) レジスタで指定された値に等しい場合、各PWMxはクリアされます。値がPR2以上の場合、PWM出力はクリアされません (100%デューティ・サイクル)。

**注釈：** PWMxDCHおよびPWMxDCLレジスタはダブル・バッファリングされています。Timer2がPR2と一致すると、バッファが更新されます。タイマーの一致が発生する前に、両方のレジスタを更新するように注意する必要があります。

### 23.1.2 PWM出力極性

PWMxCONレジスタのPWMxPOLビットをセットすることにより、出力極性が反転します。

### 23.1.3 PWM周期

PWM周期は、Timer2のPR2レジスタによって指定されます。PWM周期は、式23-1の式を使用して計算できます。

#### 式23-1: PWM周期

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

**注釈：**  $T_{osc} = 1/F_{osc}$

TMR2がPR2と等しい場合、次の3つのイベントが次の増分サイクルで発生します：

- TMR2がクリアされます
- PWM出力がアクティブです。(例外：PWMデューティ・サイクル= 0%の場合、PWM出力は非アクティブのままになります。)
- PWMxDCHおよびPWMxDCLレジスタ値はバッファーにラッチされます。

**注釈：** Timer2ポストスケーラーはPWM動作に影響を与えません。

### 23.1.4 PWMデューティ・サイクル

PWMデューティ・サイクルは、PWMxDCHとPWMxDCLレジスタ・ペアに10ビット値を書き込むことによって指定されます。PWMxDCHレジスタには、8つのMSbとPWMxDCL<7:6>、2つのLSbが含まれています。PWMxDCHおよびPWMxDCLレジスタはいつでも書き込むことができます。

式23-2を使用して、PWMパルス幅を計算します。

式23-3を使用して、PWMデューティ・サイクル比を計算します。

#### 式23-2: パルス幅

$$\text{パルス幅} = (PWMxDCH:PWMxDCL<7:6>) \cdot T_{osc} \cdot (TMR2 \text{ プリスケール値})$$

**注釈：**  $T_{osc} = 1/F_{osc}$

#### 式23-3: デューティ・サイクル比

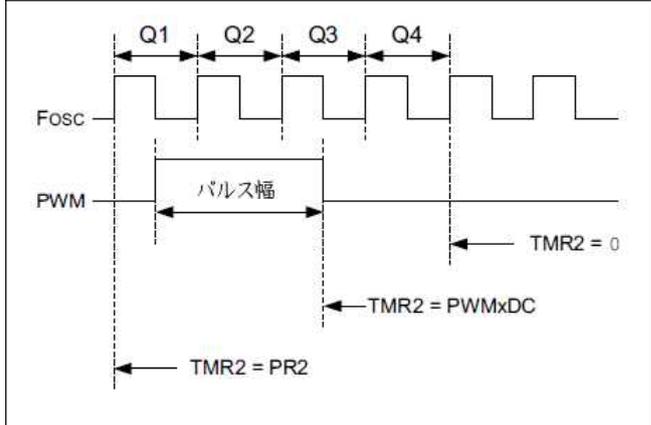
$$\text{デューティサイクル比} = \frac{(PWMxDCH:PWMxDCL<7:6>)}{4(PR2 + 1)}$$

**注釈：**  $T_{osc} = 1/F_{osc}$

8ビット・タイマーTMR2レジスタは、1/Foscの2つの最下位ビットと連結され、Timer2プリスケラーによって調整されて10ビットのタイムベースを作成します。Timer2プリスケラーが1:1に設定されている場合は、システム・クロックが使用されます。

図23-2に、デューティ・サイクルを可能な限り最小のパルスに設定した場合のPWM信号の波形を示します。

図23-2: PWM出力



### 23.1.5 PWM分解能

分解能は、特定の期間に使用可能なデューティ・サイクルの数を決定します。たとえば、10ビットの分解能では1024のディスクリット（単機能）デューティ・サイクルになりますが、8ビットの分解能では256の単機能デューティ・サイクルになります。

PR2が255の場合、最大PWM分解能は10ビットです。分解能は、式23-4に示すようにPR2レジスター値の関数です。

#### 式23-4: PWM分解能

$$\text{分解能} = \frac{\log[4(PR2 + 1)]}{\log(2)} \text{ bits}$$

**注釈：** パルス幅の値が周期よりも大きい場合、割り当てられたPWMピンは変更されません。

表23-1: PWM周波数と分解能の例 (Fosc = 20 MHz)

PWM周波数	0.31 kHz	4.88 kHz	19.53 kHz	78.12 kHz	156.3 kHz	208.3 kHz
タイマー・プリスケール	64	4	1	1	1	1
PR2値	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
最大分解能 (bits)	10	10	10	8	7	6.6

表23-2: PWM周波数と分解能の例 (Fosc = 8 MHz)

PWM周波数	0.31 kHz	4.90 kHz	19.61 kHz	76.92 kHz	153.85 kHz	200.0 kHz
タイマー・プリスケール	64	4	1	1	1	1
PR2値	0x65	0x65	0x65	0x19	0x0C	0x09
最大分解能 (bits)	8	8	8	6	5	5

### 23.1.6 スリープ・モードでの動作

スリープ・モードでは、TMR2レジスターはインクリメントせず、モジュールの状態は変化しません。PWMxピンが値を駆動している場合、PWMxピンはその値を駆動し続けます。デバイスがウェイクアップすると、TMR2は以前の状態から続行します。

### 23.1.7 システム・クロック周波数の変更

PWM周波数は、システム・クロック周波数 (Fosc) から導出されます。システム・クロック周波数を変更すると、PWM周波数も変更されます。詳細については、セクション5.0「発信器モジュール (フェイルセーフ・クロック・モニタ付き)」を参照してください。

### 23.1.8 リセットの影響

リセットを行うと、すべてのポートが強制的に入力モードになり、PWMレジスターがリセット状態になります。

## 23.1.9 PWMxピンを使用したPWM操作のセットアップ

PWMxピンを使用してPWM動作にモジュールを構成する場合は、次の手順を実行する必要があります：

1. 関連するTRISビットを設定して、PWMxピン出力ドライバーを無効にします。
2. PWMxCONレジスターをクリアします。
3. PR2レジスターにPWM周期値をロードします。
4. PWMxDCHレジスターとPWMxDCLレジスターのビット<7:6>をクリアします。
5. Timer2を構成して開始します。
  - PIR1レジスターのTMR2IF割り込みフラグ・ビットをクリアします。以下の注を参照してください。
  - T2CONレジスターのT2CKPSビットをTimer2プリスケール値で構成します。
  - T2CONレジスターのTMR2ONビットをセットしてTimer2を有効にします。
6. PWM出力ピンを有効にし、Timer2がオーバーフローするまで待ちます。PIR1レジスターのTMR2IFビットがセットされます。以下の注を参照してください。
7. 関連するTRISビットをクリアし、PWMxCONレジスターのPWMxOEビットを設定して、PWMxピン出力ドライバーを有効にします。
8. PWMxCONレジスターに適切な値をロードして、PWMモジュールを構成します。

- 注1：** 最初のPWM出力で完全なデューティ・サイクルと周期を送信するには、上記の手順を指定された順序で実行する必要があります。完全なPWM信号で開始することが重要でない場合は、ステップ8に移動してステップ4を置き換えます。
- 2：** 他の周辺機器でのみ動作する場合は、PWMxピン出力を無効にします。

## 23.2 レジスター定義: PWM制御

## レジスター23-1: PWMxCON: PWM制御レジスター

R/W-0/0	R/W-0/0	R-0/0	R/W-0/0	U-0	U-0	U-0	U-0
PWMxEN	PWMxOE	PWMxOUT	PWMxPOL	—	—	—	—
bit 7							bit 0

## 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている

- bit 7      **PWMxEN** : PWMモジュール許可ビット  
1 = PWMモジュールは有効です  
0 = PWMモジュールは無効です
- bit 6      **PWMxOE** : PWMモジュール出力許可ビット  
1 = PWMxピンへの出力は有効です  
0 = PWMxピンへの出力は無効です
- bit 5      **PWMxOUT** : PWMモジュール出力値ビット
- bit 4      **PWMxPOL** : PWMx出力極性選択ビット  
1 = PWM出力はアクティブLow  
0 = PWM出力はアクティブHigh
- bit 3~0      未実装 : 「0」として読み取ります

# PIC16(L)F1508/9

## レジスタ23-2: PWMxDCH: PWMデューティ・サイクル上位ビット

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
PWMxDCH<7:0>							
bit 7							bit 0

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている

bit 7~0 **PWMxDCH<7:0>**: PWMデューティ・サイクル最重要ビット  
これらのビットは、PWMデューティ・サイクルのMSbです。2つのLSbはPWMxDCLレジスタにあります。

## レジスタ23-3: PWMxDCL: PWMデューティ・サイクル下位ビット

R/W-x/u	R/W-x/u	U-0	U-0	U-0	U-0	U-0	U-0
PWMxDCL<7:6>							
bit 7							bit 0

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている

bit 7~6 **PWMxDCL<7:6>**: PWMデューティ・サイクル最下位ビット  
これらのビットは、PWMデューティ・サイクルのLSbsです。MSbはPWMxDCHレジスタにあります。

bit 5~0 **未実装**: 「0」として読み取ります

## 表23-3: PWMに関連付けられたレジスタの概要

名前	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ページ上の登録
PR2	Timer2モジュール期間レジスタ								166*
PWM1CON	PWM1EN	PWM1OE	PWM1OUT	PWM1POL	—	—	—	—	255
PWM1DCH	PWM1DCH<7:0>								256
PWM1DCL	PWM1DCL<7:6>		—	—	—	—	—	—	256
PWM2CON	PWM2EN	PWM2OE	PWM2OUT	PWM2POL	—	—	—	—	255
PWM2DCH	PWM2DCH<7:0>								256
PWM2DCL	PWM2DCL<7:6>		—	—	—	—	—	—	256
PWM3CON	PWM3EN	PWM3OE	PWM3OUT	PWM3POL	—	—	—	—	255
PWM3DCH	PWM3DCH<7:0>								256
PWM3DCL	PWM3DCL<7:6>		—	—	—	—	—	—	256
PWM4CON	PWM4EN	PWM4OE	PWM4OUT	PWM4POL	—	—	—	—	255
PWM4DCH	PWM4DCH<7:0>								256
PWM4DCL	PWM4DCL<7:6>		—	—	—	—	—	—	256
T2CON	—	T2OUTPS<3:0>				TMR2ON	T2CKPS<1:0>		168
TMR2	Timer2モジュールレジスタ								166*
TRISA	—	—	TRISA5	TRISA4	— <sup>(1)</sup>	TRISA2	TRISA1	TRISA0	109
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	117

記録: — =実装されていない場所、「0」として読み取られる。u =変更されていない。x =不明。網掛けのセルはPWMでは使用されません。

\* ページは登録情報を提供します。

注1: 未実装、「1」と読みます。

## 24.0 コンフィギュアエイブル・ロジック・セル (CLC)

構成可能ロジック・セル (CLCx) は、ソフトウェア実行の速度制限の範囲外で動作するプログラマブル・ロジックを提供します。ロジック・セルは最大16の入力信号を受け取り、構成可能なゲートを使用して、16の入力を8つの選択可能な単一出カロジック機能の1つを駆動する4つのロジック・ラインに減らします。

入力ソースは、次の組み合わせです：

- I/Oピン
- 内部クロック
- 周辺
- レジスター・ビット

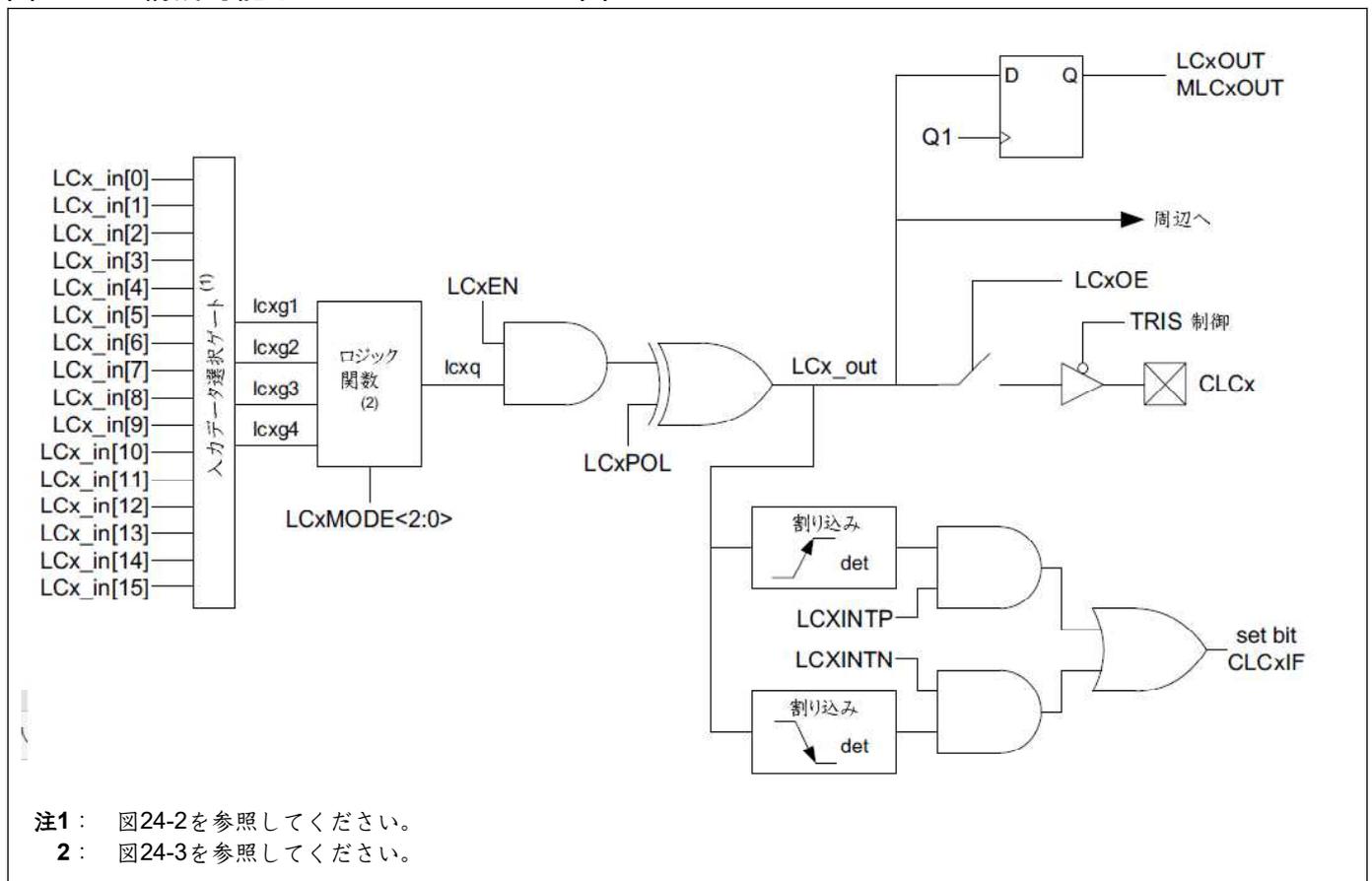
出力は、内部で周辺機器と出力ピンに送ることができます。

CLCxを通る信号の流れを示す簡略図については、図24-1を参照してください。

可能な構成は次のとおりです：

- 組み合わせロジック
  - AND
  - NAND
  - AND-OR
  - AND-OR-INVERT
  - OR-XOR
  - OR-XNOR
- ラッチ
  - S-R
  - セットとリセット付きのクロックされたD
  - セットとリセット付きの通過D
  - リセット付きのクロックされたJ-K

図24-1: 構成可能なロジック・セル・ブロック図



注1： 図24-2を参照してください。  
 注2： 図24-3を参照してください。

## 24.1 CLCxセットアップ

CLCxモジュールのプログラミングは、ロジック（論理）信号フローの4つのステージを構成することによって実行されます。4つの段階は：

- データの選択
- データ・ゲーティング
- ロジック関数の選択
- 出力極性

各ステージは、対応するCLCx特殊機能レジスターに書き込むことにより、実行時にセットアップされます。これには、プログラムの実行中にオン・ザ・フライ（実行中）でロジックを再構成できるという追加の利点があります。

### 24.1.1 データ選択

構成可能なロジックへの入力として使用できる16個の信号があります。4つの8入力マルチプレクサーを使用して、次の

ステージに渡す入力を選択します。マルチプレクサーへの16の入力は、4つのグループに配置されています。各グループは、4つのマルチプレクサーのうち2つで使用でき、いずれの場合も、異なるグループとペアになっています。この配置により、別のグループからの選択を妨げることなく、グループから最大2つを選択できます。

データの選択は、図24-2の左側に示されているように、4つのマルチプレクサーを介して行われます。図のデータ入力は、一般的な番号の入力名で識別されます。

表24-1は、一般的な入力名を各CLCxモジュールの実際の信号に関連付けています。lcxd1からlcxd4のラベルが付いた列は、選択したデータ入力のMUX出力を示します。D1SからD4Sは、MUX選択入力コードの略語です。それぞれLCxD1S<2:0>からLCxD4S<2:0>です。列のデータ入力を選択すると、その列の他のすべての入力が除外されます。

データ入力は、CLCxSEL0およびCLCxSEL1レジスター（それぞれレジスター24-3およびレジスター24-5）で選択されます。

**注釈：** データの選択は、電源投入時に定義されていません。

表24-1: CLCxデータ入力の選択

データ入力	lcxd1 D1S	lcxd2 D2S	lcxd3 D3S	lcxd4 D4S	CLC 1	CLC 2	CLC 3	CLC 4
LCx_in[0]	000	—	—	100	CLC1IN0	CLC2IN0	CLC3IN0	CLC4IN0
LCx_in[1]	001	—	—	101	CLC1IN1	CLC2IN1	CLC3IN1	CLC4IN1
LCx_in[2]	010	—	—	110	C1OUT_sync	C1OUT_sync	C1OUT_sync	C1OUT_sync
LCx_in[3]	011	—	—	111	C2OUT_sync	C2OUT_sync	C2OUT_sync	C2OUT_sync
LCx_in[4]	100	000	—	—	Fosc	Fosc	Fosc	Fosc
LCx_in[5]	101	001	—	—	T0_overflow	T0_overflow	T0_overflow	T0_overflow
LCx_in[6]	110	010	—	—	T1_overflow	T1_overflow	T1_overflow	T1_overflow
LCx_in[7]	111	011	—	—	T2_match	T2_match	T2_match	T2_match
LCx_in[8]	—	100	000	—	LC1_out	LC1_out	LC1_out	LC1_out
LCx_in[9]	—	101	001	—	LC2_out	LC2_out	LC2_out	LC2_out
LCx_in[10]	—	110	010	—	LC3_out	LC3_out	LC3_out	LC3_out
LCx_in[11]	—	111	011	—	LC4_out	LC4_out	LC4_out	LC4_out
LCx_in[12]	—	—	100	000	NCO1_out	LFINTOSC	TX_out(EUSART)	SCK_out(MSSP)
LCx_in[13]	—	—	101	001	HFINTOSC	FRC	LFINTOSC	SDO_out(MSSP)
LCx_in[14]	—	—	110	010	PWM3_out	PWM1_out	PWM2_out	PWM1_out
LCx_in[15]	—	—	111	011	PWM4_out	PWM2_out	PWM3_out	PWM4_out

### 24.1.2 データ・ゲーティング

入カマルチプレクサーからの出力は、データ・ゲーティング・ステージを介して目的の論理関数入力に送られます。各データ・ゲートは、選択した4つの入力の任意の組み合わせを指示できます。

**注釈：** データ・ゲーティングは、電源投入時に定義されていません。

ゲート・ステージは単なる信号方向ではありません。ゲートは、各入力信号を反転データまたは非反転データとして送信するように構成できます。有向信号は、各ゲートでAND演算されます。各ゲートの出力は、論理機能ステージに進む前に反転させることができます。

ゲーティングは、本質的に1対4の入力AND/NAND/OR/NORゲートです。すべての入力が反転され、出力が反転されると、ゲートはすべての有効なデータ入力のORになります。入力と出力が反転されていない場合、ゲートはANDまたはすべての有効な入力です。

表24-2に、ゲート・ロジック選択ビットを使用してゲート1で取得できる基本ロジックをまとめます。この表は4つの入力変数のロジックを示していますが、各ゲートは4つ未満を使用するように構成できます。入力が選択されていない場合、出力はゲート出力極性ビットに応じて0または1になります。

**表24-2: データ・ゲーティング・ロジック**

CLCxGLS0	LCxG1POL	ゲート・ロジック
0x55	1	AND
0x55	0	NAND
0xAA	1	NOR
0xAA	0	OR
0x00	0	ロジック0
0x00	1	ロジック1

入力の真の値と負の値の両方を選択することは可能です（ただし推奨されません）。これが行われると、他の入力に関係なく、ゲート出力はゼロになりますが、論理グリッチ（過渡誘導パルス）を放出する可能性があります。チャンネルの出力を0または1にする必要がある場合は、すべてのゲート・ビットをゼロに設定し、ゲート極性ビットを使用して目的のレベルを設定することをお勧めします。

データ・ゲーティングは、論理ゲート選択レジスターを使用して次のように構成されます：

- ゲート1： CLCxGLS0（レジスター24-5）
- ゲート2： CLCxGLS1（レジスター24-6）
- ゲート3： CLCxGLS2（レジスター24-7）
- ゲート4： CLCxGLS3（レジスター24-8）

このモジュールの他のバリエーションでは、同じレジスターに複数のゲートが選択されているため、レジスター番号のサフィックスはゲート番号とは異なります。

データ・ゲーティングは、図24-2の右側に示されています。1つのゲートのみが詳細に示されています。残りの3つのゲートは、データ・イネーブルがそのゲートのイネーブルに対応することを除いて、同じように構成されています。

### 24.1.3 ロジック・フンクション(関数)

利用可能な論理関数を含み8つあります：

- AND-OR
- OR-XOR
- AND
- S-Rラッチ
- Dフリップフロップ、セットとリセット付き
- Dフリップフロップ、リセット付き
- J-Kフリップフロップ、リセット付き
- 透過ラッチ、セットとリセット付き

論理関数を図24-3に示します。各論理関数には、4つの入力と1つの出力があります。4つの入力は、前のステージの4つのデータ・ゲート出力です。出力は反転ステージに送られ、そこから他の周辺、出力ピンに送られ、CLCx自体に戻されます。

### 24.1.4 出力極性

構成可能なロジック・セルの最後のステージは、出力極性です。CLCxCONレジスターのLCxPOLビットをセットすると、ロジック・ステージからの出力信号が反転します。割り込みが有効なときに極性を変更すると、結果の出力遷移に割り込みが発生します。

## 24.1.5 CLCxセットアップ・ステップ

CLCxを設定するときは、次の手順に従う必要があります：

- LCxENビットをクリアしてCLCxを無効にします。
- CLCxSEL0およびCLCxSEL1レジスターを使用して目的の入力を選択します（表24-1を参照）。
- 関連するANSELビットをすべてクリアします。
- 入力に関連付けられているすべてのTRISビットを設定します。
- 出力に関連付けられているすべてのTRISビットをクリアします。
- CLCxGLS0、CLCxGLS1、CLCxGLS2、およびCLCxGLS3レジスターを使用して、4つのゲートを介して選択した入力を有効にします。
- CLCxPOLレジスターのLCxPOLyビットを使用してゲート出力極性を選択します。
- CLCxCONレジスターのLCxMODE<2:0>ビットを使用して目的の論理関数を選択します。
- CLCxPOLレジスターのLCxPOLビットを使用して、ロジック出力の目的の極性を選択します。（このステップは、前のゲート出力極性ステップと組み合わせることができます）。
- デバイスを駆動する場合は、CLCxCONレジスターのLCxOEビットをセットし、その出力に対応するTRISビットもクリアします。
- 割り込みが必要な場合は、次のビットを設定します。
  - 立ち上がりイベントのCLCxCONレジスターのLCxINTPビットを設定します。
  - CLCxCONレジスターまたは立ち下がりイベントのLCxINTNビットを設定します。
  - 関連するPIEレジスターのCLCxIEビットを設定します。
  - INTCONレジスターのGIEビットとPEIEビットを設定します。
- CLCxCONレジスターのLCxENビットをセットしてCLCxを有効にします。

## 24.2 CLCx割り込み

適切な割り込み許可が設定されている場合、CLCxの出力値が変更されると、割り込みが生成されます。この目的のために、立ち上がりエッジ検出器と立ち下がりエッジ検出器が各CLCに存在します。

関連するPIRレジスターのCLCxIFビットは、いずれかのエッジ検出器がトリガーされ、関連する許可ビットが設定されると設定されます。LCxINTPは立ち上がりエッジの割り込みを有効にし、LCxINTNビットは立ち下がりエッジの割り込みを有効にします。どちらもCLCxCONレジスターにあります。

割り込みを完全に有効にするには、次のビットを設定します：

- CLCxCONレジスターのLCxONビット
- 関連するPIEレジスターのCLCxIEビット
- CLCxCONレジスターのLCxINTPビット（立ち上がりエッジ検出用）
- CLCxCONレジスターのLCxINTNビット（立ち下がりエッジ検出用）
- INTCONレジスターのPEIEビットとGIEビット

関連するPIRレジスターのCLCxIFビットは、割り込みサービスの一部としてソフトウェアでクリアする必要があります。このフラグがクリアされている間に別のエッジが検出された場合でも、シーケンスの最後にフラグが設定されます。

## 24.3 出力ミラー・コピー

すべてのLCxCON出力ビットのミラー・コピーはCLCxDATAレジスターに含まれています。このレジスターを読み取ると、すべてのCLCの出力が同時に読み取られます。これにより、個々のCLCxCONレジスターのCLCxOUTビットをテストまたは読み取ることによって生じる読み取りスキュー（歪み？）が防止されます。

## 24.4 リセットの影響

リセットの結果、CLCxCONレジスターはゼロにクリアされます。他のすべての選択値とゲーティング値は変更されません。

## 24.5 スリープ中の動作

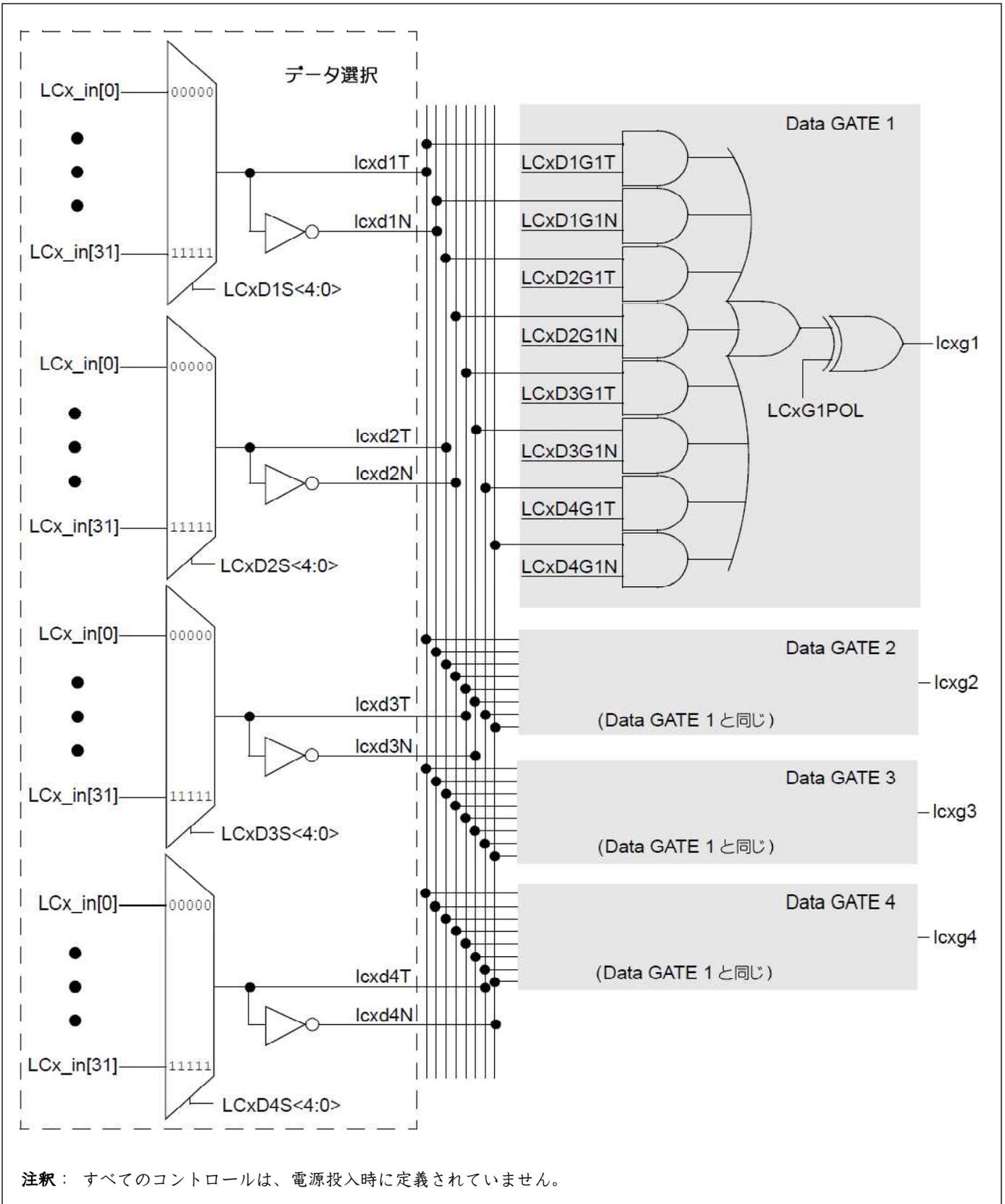
CLCモジュールは、システム・クロックとは独立して動作し、選択した入力ソースがアクティブのままである限り、スリープ中も実行を継続します。

選択したシステム・クロック・ソースに関係なく、CLCモジュールが有効で、HFINTOSCが入力ソースとして選択されている場合、スリープ中はHFINTOSCがアクティブのままになります。

つまり、HFINTOSCがシステム・クロックとCLC入力ソースとして同時に選択されている場合、CLCが有効になっていると、スリープ中にCPUはアイドル状態になりますが、CLCは動作を継続し、HFINTOSCはアクティブのままになります。

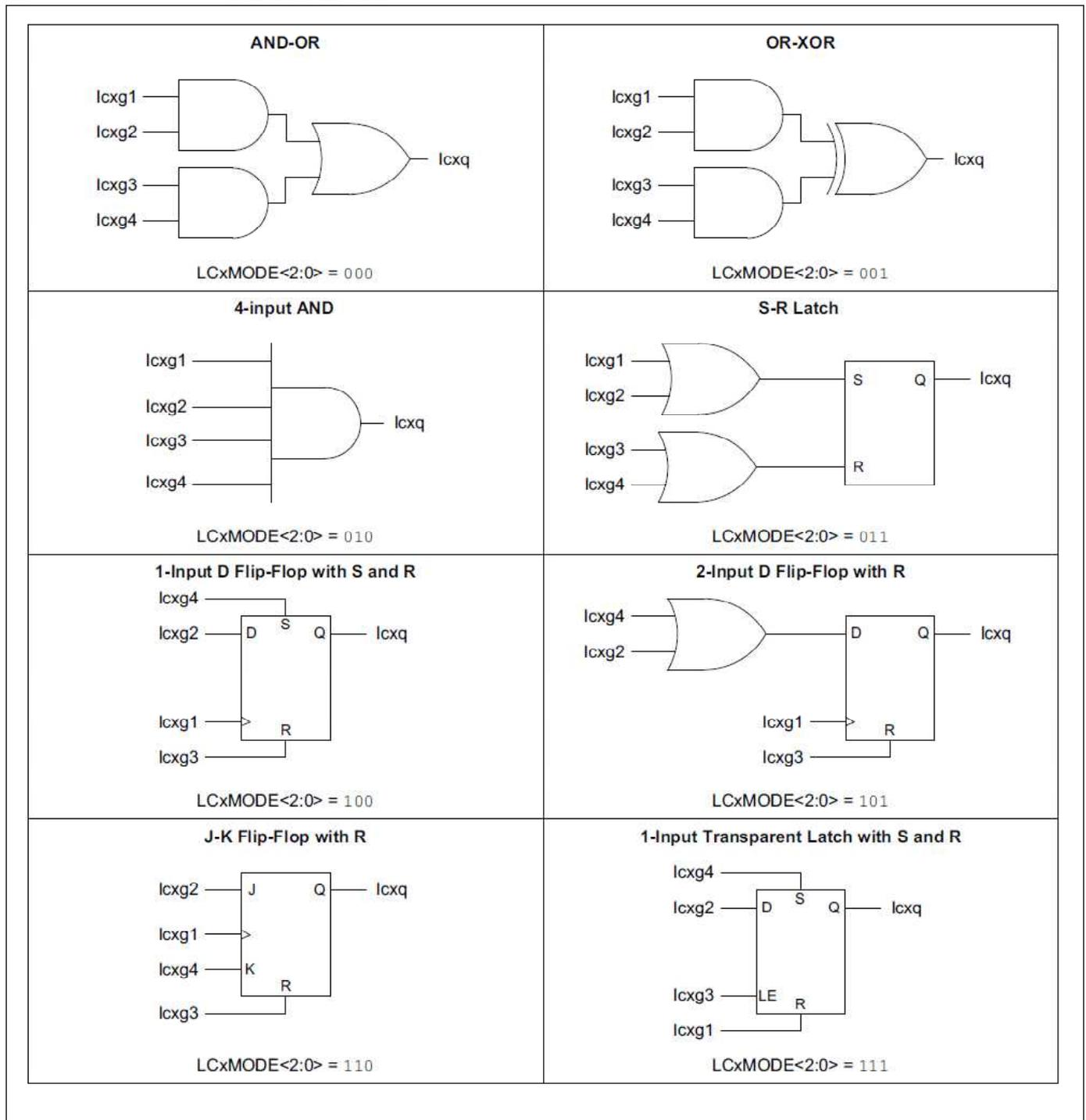
これは、スリープ・モード電流に直接影響します。

図24-2: 入力データの選択とゲーティング



# PIC16(L)F1508/9

図24-3: プログラム可能な論理関数



## 24.6 レジスタ定義: CLC制御

### レジスタ24-1: CLCxCON: 構成可能ロジック・セル制御レジスタ

R/W-0/0	R/W-0/0	R-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
LCxEN	LCxOE	LCxOUT	LCxINTP	LCxINTN	LCxMODE<2:0>		
bit 7						bit 0	

#### 記録:

R =読み取り可能ビット	W =書き込み可能ビット	U =実装されていないビット、「0」として読み取られる
u =ビットは変更されていません	x =ビットは不明です	-n/n =PORでの値およびBOR/他のすべてのリセットでの値
[1]=ビットが設定されている	[0]=ビットがクリアされている	

- bit 7    **LCxEN**: 構成可能ロジック・セル許可ビット  
 1 = 構成可能なロジック・セルが有効になっており、入力信号を混合しています  
 0 = 構成可能なロジック・セルが無効になっており、論理ゼロ出力があります
- bit 6    **LCxOE**: 構成可能ロジック・セル出力許可ビット  
 1 = 構成可能なロジック・セル・ポート・ピン出力が有効  
 0 = 設定可能なロジック・セル・ポート・ピン出力が無効
- bit 5    **LCxOUT**: 構成可能ロジック・セル・データ出力ビット  
 読み取り専用: LCxPOL後のロジック・セル出力データ。lcx\_out ワイヤーからサンプリング。
- bit 4    **LCxINTP**: 構成可能ロジック・セルの割り込みになる正方向エッジ許可ビット  
 1 = CLCxIFは、lcx\_out で立ち上がりエッジが発生したときに設定されます  
 0 = CLCxIFは設定されません
- bit 3    **LCxINTN**: 構成可能ロジック・セルの割り込みになる負方向エッジ許可ビット  
 1 = CLCxIFは、lcx\_out で立ち下がりエッジが発生したときに設定されます  
 0 = CLCxIFは設定されません
- bit 2~0    **LCxMODE<2:0>**: 構成可能ロジック・セル関数モード・ビット  
 111 = セルは1入力透過ラッチで、SとR付き  
 110 = セルはJ-Kフリップフロップで、R付き  
 101 = セルは2入力Dフリップフロップで、R付き  
 100 = セルは1入力Dフリップフロップで、SとR付き  
 011 = セルはS-Rラッチ  
 010 = セルは4入力AND  
 001 = セルはOR-XOR  
 000 = セルはAND-OR

# PIC16(L)F1508/9

## レジスタ24-2: CLCxPOL: 信号極性制御レジスタ

R/W-0/0	U-0	U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
LCxPOL	—	—	—	LCxG4POL	LCxG3POL	LCxG2POL	LCxG1POL
bit 7							bit 0

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている

- bit 7      **LCxPOL**: LCOOUT極性制御ビット  
1 = ロジック・セルの出力は反転されます  
0 = ロジック・セルの出力は反転されません
- bit 6~4      **未実装**: 「0」として読み取ります
- bit 3      **LCxG4POL**: ゲート4出力極性制御ビット  
1 = ゲート4の出力は、ロジック・セルに適用されると反転されます  
0 = ゲート4の出力は、反転されません
- bit 2      **LCxG3POL**: ゲート3出力極性制御ビット  
1 = ゲート3の出力は、ロジック・セルに適用されると反転されます  
0 = ゲート3の出力は、反転されません
- bit 1      **LCxG2POL**: ゲート2出力極性制御ビット  
1 = ゲート2の出力は、ロジック・セルに適用されると反転されます  
0 = ゲート2の出力は、反転されません
- bit 0      **LCxG1POL**: ゲート1出力極性制御ビット  
1 = ゲート1の出力は、ロジック・セルに適用されると反転されます  
0 = ゲート1の出力は、反転されません

## レジスタ24-3: CLCxSEL0: マルチプレクサー・データ1と2の選択レジスタ

U-0	R/W-x/u	R/W-x/u	R/W-x/u	U-0	R/W-x/u	R/W-x/u	R/W-x/u
—	LCxD2S<2:0> <sup>(1)</sup>			—	LCxD1S<2:0> <sup>(1)</sup>		
bit 7							bit 0

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
 u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
 「1」=ビットが設定されている      「0」=ビットがクリアされている

- bit 7      **未実装**: 「0」として読み取ります
- bit 6~4      **LCxD2S<2:0>**: 入力データ2選択制御ビット<sup>(1)</sup>
- 111 = LCx\_in[11] は lcx2 のために選択されます
  - 110 = LCx\_in[10] は lcx2 のために選択されます
  - 101 = LCx\_in[9] は lcx2 のために選択されます
  - 100 = LCx\_in[8] は lcx2 のために選択されます
  - 011 = LCx\_in[7] は lcx2 のために選択されます
  - 010 = LCx\_in[6] は lcx2 のために選択されます
  - 001 = LCx\_in[5] は lcx2 のために選択されます
  - 000 = LCx\_in[4] は lcx2 のために選択されます
- bit 3      **未実装**: 「0」として読み取ります
- bit 2~0      **LCxD1S<2:0>**: 入力データ1選択制御ビット<sup>(1)</sup>
- 111 = LCx\_in[7] は lcx1 のために選択されます
  - 110 = LCx\_in[6] は lcx1 のために選択されます
  - 101 = LCx\_in[5] は lcx1 のために選択されます
  - 100 = LCx\_in[4] は lcx1 のために選択されます
  - 011 = LCx\_in[3] は lcx1 のために選択されます
  - 010 = LCx\_in[2] は lcx1 のために選択されます
  - 001 = LCx\_in[1] は lcx1 のために選択されます
  - 000 = LCx\_in[0] は lcx1 のために選択されます

**注1**: 入力に関連する信号名については、表24-1を参照してください。

# PIC16(L)F1508/9

## レジスタ24-4: CLCxSEL1: マルチプレクサー・データ3と4選択レジスタ

U-0	R/W-x/u	R/W-x/u	R/W-x/u	U-0	R/W-x/u	R/W-x/u	R/W-x/u
—	LCxD4S<2:0> <sup>(1)</sup>			—	LCxD3S<2:0> <sup>(1)</sup>		
bit 7							bit 0

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている

- bit 7      **未実装**: 「0」として読み取ります
- bit 6~4      **LCxD4S<2:0>**: 入力データ4選択制御ビット<sup>(1)</sup>  
111 = LCx\_in[3] は lcx4 のために選択されます  
110 = LCx\_in[2] は lcx4 のために選択されます  
101 = LCx\_in[1] は lcx4 のために選択されます  
100 = LCx\_in[0] は lcx4 のために選択されます  
011 = LCx\_in[15] は lcx4 のために選択されます  
010 = LCx\_in[14] は lcx4 のために選択されます  
001 = LCx\_in[13] は lcx4 のために選択されます  
000 = LCx\_in[12] は lcx4 のために選択されます
- bit 3      **未実装**: 「0」として読み取ります
- bit 2~0      **LCxD3S<2:0>**: 入力データ3選択制御ビット<sup>(1)</sup>  
111 = LCx\_in[15] は lcx3 のために選択されます  
110 = LCx\_in[14] は lcx3 のために選択されます  
101 = LCx\_in[13] は lcx3 のために選択されます  
100 = LCx\_in[12] は lcx3 のために選択されます  
011 = LCx\_in[11] は lcx3 のために選択されます  
010 = LCx\_in[10] は lcx3 のために選択されます  
001 = LCx\_in[9] は lcx3 のために選択されます  
000 = LCx\_in[8] は lcx3 のために選択されます

注1: 入力に関連する信号名については、表24-1を参照してください。

## レジスタ24-5: CLCxGLS0: ゲート1ロジック選択レジスタ

| R/W-x/u  |
|----------|----------|----------|----------|----------|----------|----------|----------|
| LCxG1D4T | LCxG1D4N | LCxG1D3T | LCxG1D3N | LCxG1D2T | LCxG1D2N | LCxG1D1T | LCxG1D1N |
| bit 7    |          |          |          |          |          |          | bit 0    |

## 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている

- bit 7      **LCxG1D4T**: ゲート1データ4真 (非反転) ビット  
1 = lcx4T は lcxg1 にゲートされます  
0 = lcx4T は lcxg1 にゲートされません
- bit 6      **LCxG1D4N**: ゲート1データ4否定 (反転) ビット  
1 = cxd4N は lcxg1 にゲートされます  
0 = lcx4N は lcxg1 にゲートされません
- bit 5      **LCxG1D3T**: ゲート1データ3真 (非反転) ビット  
1 = lcx3T は lcxg1 にゲートされます  
0 = lcx3T は lcxg1 にゲートされません
- bit 4      **LCxG1D3N**: ゲート1データ3否定 (反転) ビット  
1 = lcx3N は lcxg1 にゲートされます  
0 = lcx3N は lcxg1 にゲートされません
- bit 3      **LCxG1D2T**: ゲート1データ2真 (非反転) ビット  
1 = lcx2T は lcxg1 にゲートされます  
0 = lcx2T は lcxg1 にゲートされません
- bit 2      **LCxG1D2N**: ゲート1データ2否定 (反転) ビット  
1 = lcx2N は lcxg1 にゲートされます  
0 = lcx2N は lcxg1 にゲートされません
- bit 1      **LCxG1D1T**: ゲート1データ1真 (非反転) ビット  
1 = lcx1T は lcxg1 にゲートされます  
0 = lcx1T は lcxg1 にゲートされません
- bit 0      **LCxG1D1N**: ゲート1データ1否定 (反転) ビット  
1 = lcx1N は lcxg1 にゲートされます  
0 = lcx1N は lcxg1 にゲートされません

# PIC16(L)F1508/9

## レジスタ24-6: CLCxGLS1: ゲート2ロジック選択レジスタ

| R/W-x/u  |
|----------|----------|----------|----------|----------|----------|----------|----------|
| LCxG2D4T | LCxG2D4N | LCxG2D3T | LCxG2D3N | LCxG2D2T | LCxG2D2N | LCxG2D1T | LCxG2D1N |
| bit 7    |          |          |          |          |          |          | bit 0    |

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている

- bit 7      **LCxG2D4T**: ゲート2データ4真 (非反転) ビット  
1 = lcx4T は lcxg2 にゲートされます  
0 = lcx4T は lcxg2 にゲートされません
- bit 6      **LCxG2D4N**: ゲート2データ4否定 (反転) ビット  
1 = lcx4N は lcxg2 にゲートされます  
0 = lcx4N は lcxg2 にゲートされません
- bit 5      **LCxG2D3T**: ゲート2データ3真 (非反転) ビット  
1 = lcx3T は lcxg2 にゲートされます  
0 = lcx3T は lcxg2 にゲートされません
- bit 4      **LCxG2D3N**: ゲート2データ3否定 (反転) ビット  
1 = lcx3N は lcxg2 にゲートされます  
0 = lcx3N は lcxg2 にゲートされません
- bit 3      **LCxG2D2T**: ゲート2データ2真 (非反転) ビット  
1 = lcx2T は lcxg2 にゲートされます  
0 = lcx2T は lcxg2 にゲートされません
- bit 2      **LCxG2D2N**: ゲート2データ2否定 (反転) ビット  
1 = lcx2N は lcxg2 にゲートされます  
0 = lcx2N は lcxg2 にゲートされません
- bit 1      **LCxG2D1T**: ゲート2データ1真 (非反転) ビット  
1 = lcx1T は lcxg2 にゲートされます  
0 = lcx1T は lcxg2 にゲートされません
- bit 0      **LCxG2D1N**: ゲート2データ1否定 (反転) ビット  
1 = lcx1N は lcxg2 にゲートされます  
0 = lcx1N は lcxg2 にゲートされません

## レジスタ24-7: CLCxGLS2: ゲート3ロジック選択レジスタ

| R/W-x/u  |
|----------|----------|----------|----------|----------|----------|----------|----------|
| LCxG3D4T | LCxG3D4N | LCxG3D3T | LCxG3D3N | LCxG3D2T | LCxG3D2N | LCxG3D1T | LCxG3D1N |
| bit 7    |          |          |          |          |          |          | bit 0    |

## 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている

- bit 7      **LCxG3D4T**: ゲート3データ4真 (非反転) ビット  
1 = lcx4T は lcxg3 にゲートされます  
0 = lcx4T は lcxg3 にゲートされません
- bit 6      **LCxG3D4N**: ゲート3データ4否定 (反転) ビット  
1 = lcx4N は lcxg3 にゲートされます  
0 = lcx4N は lcxg3 にゲートされません
- bit 5      **LCxG3D3T**: ゲート3データ3真 (非反転) ビット  
1 = lcx3T は lcxg3 にゲートされます  
0 = lcx3T は lcxg3 にゲートされません
- bit 4      **LCxG3D3N**: ゲート3データ3否定 (反転) ビット  
1 = lcx3N は lcxg3 にゲートされます  
0 = lcx3N は lcxg3 にゲートされません
- bit 3      **LCxG3D2T**: ゲート3データ2真 (非反転) ビット  
1 = lcx2T は lcxg3 にゲートされます  
0 = lcx2T は lcxg3 にゲートされません
- bit 2      **LCxG3D2N**: ゲート3データ2否定 (反転) ビット  
1 = lcx2N は lcxg3 にゲートされます  
0 = lcx2N は lcxg3 にゲートされません
- bit 1      **LCxG3D1T**: ゲート3データ1真 (非反転) ビット  
1 = lcx1T は lcxg3 にゲートされます  
0 = lcx1T は lcxg3 にゲートされません
- bit 0      **LCxG3D1N**: ゲート3データ1否定 (反転) ビット  
1 = lcx1N は lcxg3 にゲートされます  
0 = lcx1N は lcxg3 にゲートされません

# PIC16(L)F1508/9

## レジスタ24-8: CLCxGLS3: ゲート4ロジック選択レジスタ

| R/W-x/u  |
|----------|----------|----------|----------|----------|----------|----------|----------|
| LCxG4D4T | LCxG4D4N | LCxG4D3T | LCxG4D3N | LCxG4D2T | LCxG4D2N | LCxG4D1T | LCxG4D1N |
| bit 7    |          |          |          |          |          |          | bit 0    |

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている

- bit 7      **LCxG4D4T**: ゲート4データ4真 (非反転) ビット  
1 = lcx4T は lcxg4 にゲートされます  
0 = lcx4T は lcxg4 にゲートされません
- bit 6      **LCxG4D4N**: ゲート4データ4否定 (反転) ビット  
1 = lcx4N は lcxg4 にゲートされます  
0 = lcx4N は lcxg4 にゲートされません
- bit 5      **LCxG4D3T**: ゲート4データ3真 (非反転) ビット  
1 = lcx3T は lcxg4 にゲートされます  
0 = lcx3T は lcxg4 にゲートされません
- bit 4      **LCxG4D3N**: ゲート4データ3否定 (反転) ビット  
1 = lcx3N は lcxg4 にゲートされます  
0 = lcx3N は lcxg4 にゲートされません
- bit 3      **LCxG4D2T**: ゲート4データ2真 (非反転) ビット  
1 = lcx2T は lcxg4 にゲートされます  
0 = lcx2T は lcxg4 にゲートされません
- bit 2      **LCxG4D2N**: ゲート4データ2否定 (反転) ビット  
1 = lcx2N は lcxg4 にゲートされます  
0 = lcx2N は lcxg4 にゲートされません
- bit 1      **LCxG4D1T**: ゲート4データ1真 (非反転) ビット  
1 = lcx1T は lcxg4 にゲートされます  
0 = lcx1T は lcxg4 にゲートされません
- bit 0      **LCxG4D1N**: ゲート4データ1否定 (反転) ビット  
1 = lcx1N は lcxg4 にゲートされます  
0 = lcx1N は lcxg4 にゲートされません

## レジスタ-24-9: CLCDATA: CLCデータ出力

U-0	U-0	U-0	U-0	R-0	R-0	R-0	R-0
—	—	—	—	MLC4OUT	MLC3OUT	MLC2OUT	MLC1OUT
bit 7							bit 0

### 記録:

**R** =読み取り可能ビット      **W** =書き込み可能ビット      **U** =実装されていないビット、「0」として読み取られる  
**u** =ビットは変更されていません      **x** =ビットは不明です      **-n/n** =PORでの値およびBOR/他のすべてのリセットでの値  
**[1]** =ビットが設定されている      **[0]** =ビットがクリアされている

bit 7~4    **未実装**: 「0」として読み取ります  
 bit 3      **MLC4OUT**: LC4OUTのミラー・コピー・ビット  
 bit 2      **MLC3OUT**: LC3OUTのミラー・コピー・ビット  
 bit 1      **MLC2OUT**: LC2OUTのミラー・コピー・ビット  
 bit 0      **MLC1OUT**: LC1OUTのミラー・コピー・ビット

# PIC16(L)F1508/9

表24-3: CLCxに関連付けられたレジスタの概要

名前	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ページ上の登録
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	110
ANSELB	—	—	ANSB5	ANSB4	—	—	—	—	114
ANSELC	ANSC7	ANSC6	—	—	ANSC3	ANSC2	ANSC1	ANSC0	118
CLC1CON	LC1EN	LC1OE	LC1OUT	LC1INTP	LC1INTN	LC1MODE<2:0>			263
CLCDATA	—	—	—	—	—	MLC3OUT	MLC2OUT	MLC1OUT	271
CLC1GLS0	LC1G1D4T	LC1G1D4N	LC1G1D3T	LC1G1D3N	LC1G1D2T	LC1G1D2N	LC1G1D1T	LC1G1D1N	267
CLC1GLS1	LC1G2D4T	LC1G2D4N	LC1G2D3T	LC1G2D3N	LC1G2D2T	LC1G2D2N	LC1G2D1T	LC1G2D1N	268
CLC1GLS2	LC1G3D4T	LC1G3D4N	LC1G3D3T	LC1G3D3N	LC1G3D2T	LC1G3D2N	LC1G3D1T	LC1G3D1N	269
CLC1GLS3	LC1G4D4T	LC1G4D4N	LC1G4D3T	LC1G4D3N	LC1G4D2T	LC1G4D2N	LC1G4D1T	LC1G4D1N	270
CLC1POL	LC1POL	—	—	—	LC1G4POL	LC1G3POL	LC1G2POL	LC1G1POL	264
CLC1SEL0	—	LC1D2S<2:0>			—	LC1D1S<2:0>			265
CLC1SEL1	—	LC1D4S<2:0>			—	LC1D3S<2:0>			266
CLC2CON	LC2EN	LC2OE	LC2OUT	LC2INTP	LC2INTN	LC2MODE<2:0>			263
CLC2GLS0	LC2G1D4T	LC2G1D4N	LC2G1D3T	LC2G1D3N	LC2G1D2T	LC2G1D2N	LC2G1D1T	LC2G1D1N	267
CLC2GLS1	LC2G2D4T	LC2G2D4N	LC2G2D3T	LC2G2D3N	LC2G2D2T	LC2G2D2N	LC2G2D1T	LC2G2D1N	268
CLC2GLS2	LC2G3D4T	LC2G3D4N	LC2G3D3T	LC2G3D3N	LC2G3D2T	LC2G3D2N	LC2G3D1T	LC2G3D1N	269
CLC2GLS3	LC2G4D4T	LC2G4D4N	LC2G4D3T	LC2G4D3N	LC2G4D2T	LC2G4D2N	LC2G4D1T	LC2G4D1N	270
CLC2POL	LC2POL	—	—	—	LC2G4POL	LC2G3POL	LC2G2POL	LC2G1POL	264
CLC2SEL0	—	LC2D2S<2:0>			—	LC2D1S<2:0>			265
CLC2SEL1	—	LC2D4S<2:0>			—	LC2D3S<2:0>			266
CLC3CON	LC3EN	LC3OE	LC3OUT	LC3INTP	LC3INTN	LC3MODE<2:0>			263
CLC3GLS0	LC3G1D4T	LC3G1D4N	LC3G1D3T	LC3G1D3N	LC3G1D2T	LC3G1D2N	LC3G1D1T	LC3G1D1N	267
CLC3GLS1	LC3G2D4T	LC3G2D4N	LC3G2D3T	LC3G2D3N	LC3G2D2T	LC3G2D2N	LC3G2D1T	LC3G2D1N	268
CLC3GLS2	LC3G3D4T	LC3G3D4N	LC3G3D3T	LC3G3D3N	LC3G3D2T	LC3G3D2N	LC3G3D1T	LC3G3D1N	269
CLC3GLS3	LC3G4D4T	LC3G4D4N	LC3G4D3T	LC3G4D3N	LC3G4D2T	LC3G4D2N	LC3G4D1T	LC3G4D1N	270
CLC3POL	LC3POL	—	—	—	LC3G4POL	LC3G3POL	LC3G2POL	LC3G1POL	264
CLC3SEL0	—	LC3D2S<2:0>			—	LC3D1S<2:0>			265
CLC3SEL1	—	LC3D4S<2:0>			—	LC3D3S<2:0>			266
CLC4CON	LC4EN	LC4OE	LC4OUT	LC4INTP	LC4INTN	LC4MODE<2:0>			263
CLC4GLS0	LC4G1D4T	LC4G1D4N	LC4G1D3T	LC4G1D3N	LC4G1D2T	LC4G1D2N	LC4G1D1T	LC4G1D1N	267
CLC4GLS1	LC4G2D4T	LC4G2D4N	LC4G2D3T	LC4G2D3N	LC4G2D2T	LC4G2D2N	LC4G2D1T	LC4G2D1N	268
CLC4GLS2	LC4G3D4T	LC4G3D4N	LC4G3D3T	LC4G3D3N	LC4G3D2T	LC4G3D2N	LC4G3D1T	LC4G3D1N	269
CLC4GLS3	LC4G4D4T	LC4G4D4N	LC4G4D3T	LC4G4D3N	LC4G4D2T	LC4G4D2N	LC4G4D1T	LC4G4D1N	270
CLC4POL	LC4POL	—	—	—	LC4G4POL	LC4G3POL	LC4G2POL	LC4G1POL	264
CLC4SEL0	—	LC4D2S<2:0>			—	LC4D1S<2:0>			265
CLC4SEL1	—	LC4D4S<2:0>			—	LC4D3S<2:0>			266
INTCON	GIE	PEIE	TMR0IE	INTE	IOCFIE	TMR0IF	INTF	IOCFIF	75
PIE3	—	—	—	—	CLC4IE	CLC3IE	CLC2IE	CLC1IE	78
PIR3	—	—	—	—	CLC4IF	CLC3IF	CLC2IF	CLC1IF	81
TRISA	—	—	TRISA5	TRISA4	— <sup>(1)</sup>	TRISA2	TRISA1	TRISA0	109
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	113
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	117

記録: — =未実装の読み取りは「0」として。網掛けのセルはCLCモジュールには使用されません。

注1: 未実装、「1」と読みます。

## 25.0 数値制御発振器(NCO)モジュール

数値制御発振器 (NCOx) モジュールは、増分値の加算によるオーバーフローを使用して入力周波数を除算するタイマーです。単純なカウンタ駆動タイマーに対する加算方法の利点は、除算の分解能が除算値によって変化しないことです。NCOxは、固定デューティ・サイクルで周波数精度と高分解能を必要とするアプリケーションに最も役立ちます。

NCOxの機能は次のとおりです：

- 16ビット・インクリメント機能
- 固定デューティ・サイクル (FDC) モード
- パルス周波数 (PF) モード
- 出力パルス幅制御
- 複数のクロック入力ソース
- 出力極性制御
- 割り込み機能

図25-1は、NCOxモジュールの簡略化されたブロック図です。

### 25.1 NCOx動作

NCOxは、アキュムレーターに固定値を繰り返し加算することで動作します。追加は入力クロック・レートで発生します。アキュムレーターは、生のNCOx出力 (NCO\_overflow) であるキャリーで定期的にオーバーフローします。これにより、加算値と最大アキュムレーター値の比率によって入力クロックが効果的に削減されます。式25-1を参照してください。

NCOx出力は、パルスをストレッチ (伸縮) するか、フリップフロップを切り替えることでさらに変更できます。変更されたNCOx出力は、内部で他の周辺に分配され、オプションでピンに出力されます。アキュムレータのオーバーフローも割り込み (NCO\_interrupt) を生成します。

NCOx周期は離散ステップで変化し、平均頻度を作成します。この出力は、不確実性を低減するためにNCOx出力を平均化する受信回路 (つまり、CWGまたは外部共振コンバータ回路) の能力に依存します。

#### 25.1.1 NCOxクロック・ソース

NCOxで利用可能なクロックソースには次のものがあります：

- HFINTOSC
- Fosc
- LC1\_out
- CLKINピン

NCOxクロック・ソースは、NCOxCLKレジスタのNxCKS<2:0>ビットを構成することによって選択されます。

#### 式25-1:

$$F_{OVERFLOW} = \frac{NCO \text{ クロック周波数} \times \text{インクリメント値}}{2^n}$$

$n$  = ビット単位のアキュムレータ幅

#### 25.1.2 アキュムレーター

アキュムレーターは20ビット・レジスタです。アキュムレーターへの読み取りおよび書き込みアクセスは、3つのレジスタを介して利用できます。：

- NCOxACCL
- NCOxACCH
- NCOxACCU

#### 25.1.3 ADDER

NCOx加算器は全加算器であり、システム・クロックとは独立して動作します。前の結果とインクリメント値を加算すると、各入力クロックの立ち上がりエッジのアキュムレーター値が置き換えられます。

#### 25.1.4 インクリメント・レジスタ

インクリメント値は、16ビットのインクリメントを構成する2つの8ビットレジスタに格納されます。LSBからMSBの順に：

- NCOxINCL
- NCOxINCH

NCOモジュールが有効になっている場合は、最初にNCOxINCHを書き込み、次にNCOxINCLレジスタを書き込む必要があります。NCOxINCLレジスタに書き込むと、インクリメント・バッファ・レジスタが開始され、NCOx\_clk信号の2番目の立ち上がりエッジで同時にロードされます。

レジスタは読み取りと書き込みが可能です。インクリメント・レジスタはダブル・バッファリングされているため、最初にNCOxモジュールを無効にすることなく値を変更できます。

NCOモジュールが無効になっている場合、インクリメント・バッファは、インクリメント・レジスタへの書き込みの直後にロードされます。

注釈： インクリメント・バッファ・レジスタはユーザーがアクセスできません。



## 25.2 固定デューティ・サイクル (FDC) モード

固定デューティ・サイクル (FDC) モードでは、アキュムレーターがオーバーフローするたびに (NCO\_overflow)、出力が切り替えられます。これにより、増分値が一定のままである場合、50%のデューティ・サイクルが提供されます。詳細については、図25-2を参照してください。

FDCモードは、NCOxCONレジスタのNxPFMビットをクリアすることによって選択されます。

## 25.3 パルス周波数 (PF) モード

パルス・フレクシー (周波数) (PF) モードでは、アキュムレーターがオーバーフローするたびに (NCO\_overflow)、出力は1つ以上のクロック周期でアクティブになります。クロック周期が経過すると、出力は非アクティブ状態に戻ります。これにより、パルス出力が提供されます。

出力は、オーバーフロー・イベントの直後のクロックの立ち上がりエッジでアクティブになります。詳細については、図25-2を参照してください。

アクティブ状態と非アクティブ状態の値は、NCOxCONレジスタの極性ビットNxPOLによって異なります。

PFモードは、NCOxCONレジスタのNxPFMビットをセットすることによって選択されます。

### 25.3.1 出力パルス幅制御

PFモードで動作している場合、出力のアクティブ状態の幅は複数のクロック周期で変化する可能性があります。NCOxCLKレジスタのNxPWS<2:0>ビットを使用して、さまざまなパルス幅が選択されます。

選択したパルス幅がアキュムレーターのオーバーフロー時間枠よりも大きい場合、NCOx動作の出力は不確定です。

## 25.4 出力極性制御

NCOxモジュールの最後のステージは出力極性です。NCOxCONレジスタのNxPOLビットは、出力極性を選択します。割り込みが有効なときに極性を変更すると、結果の出力遷移に割り込みが発生します。

NCOx出力は、ソースコードまたはその他の周辺機器によって内部的に使用できます。これを実現するには、NCOxCONレジスタのNxOUT (読み取り専用) ビットを読み取ります。

NCOx出力信号は、次の周辺機器で使用できます：

- CLC
- CWG

## 25.5 割り込み

アキュムレーターがオーバーフローすると (NCO\_overflow)、PIRxレジスタのNCOx割り込みフラグ・ビットNCOxIFがセットされます。割り込みイベント (NCO\_interrupt) を有効にするには、次のビットを設定する必要があります：

- NCOxCONレジスタのNxENビット
- PIRxレジスタのNCOxIEビット
- INTCONレジスタのPEIEビット
- INTCONレジスタのGIEビット

割り込みは、割り込みサービス・ルーチンのNCOxIFビットをクリアすることにより、ソフトウェアによってクリアされる必要があります。

## 25.6 リセットの影響

リセットの結果、すべてのNCOxレジスタがゼロにクリアされます。

## 25.7 スリープでの動作

NCOモジュールは、システム・クロックとは独立して動作し、選択したクロック・ソースがアクティブのままである限り、スリープ中も実行を継続します。

NCOモジュールが有効で、HFINTOSCがクロック・ソースとして選択されている場合、選択されているシステム・クロック・ソースに関係なく、スリープ中はHFINTOSCがアクティブのままになります。

つまり、HFINTOSCがシステム・クロックとNCOクロック・ソースとして同時に選択されている場合、NCOが有効になっていると、スリープ中にCPUはアイドル状態になりますが、NCOは動作を継続し、HFINTOSCはアクティブのままになります。

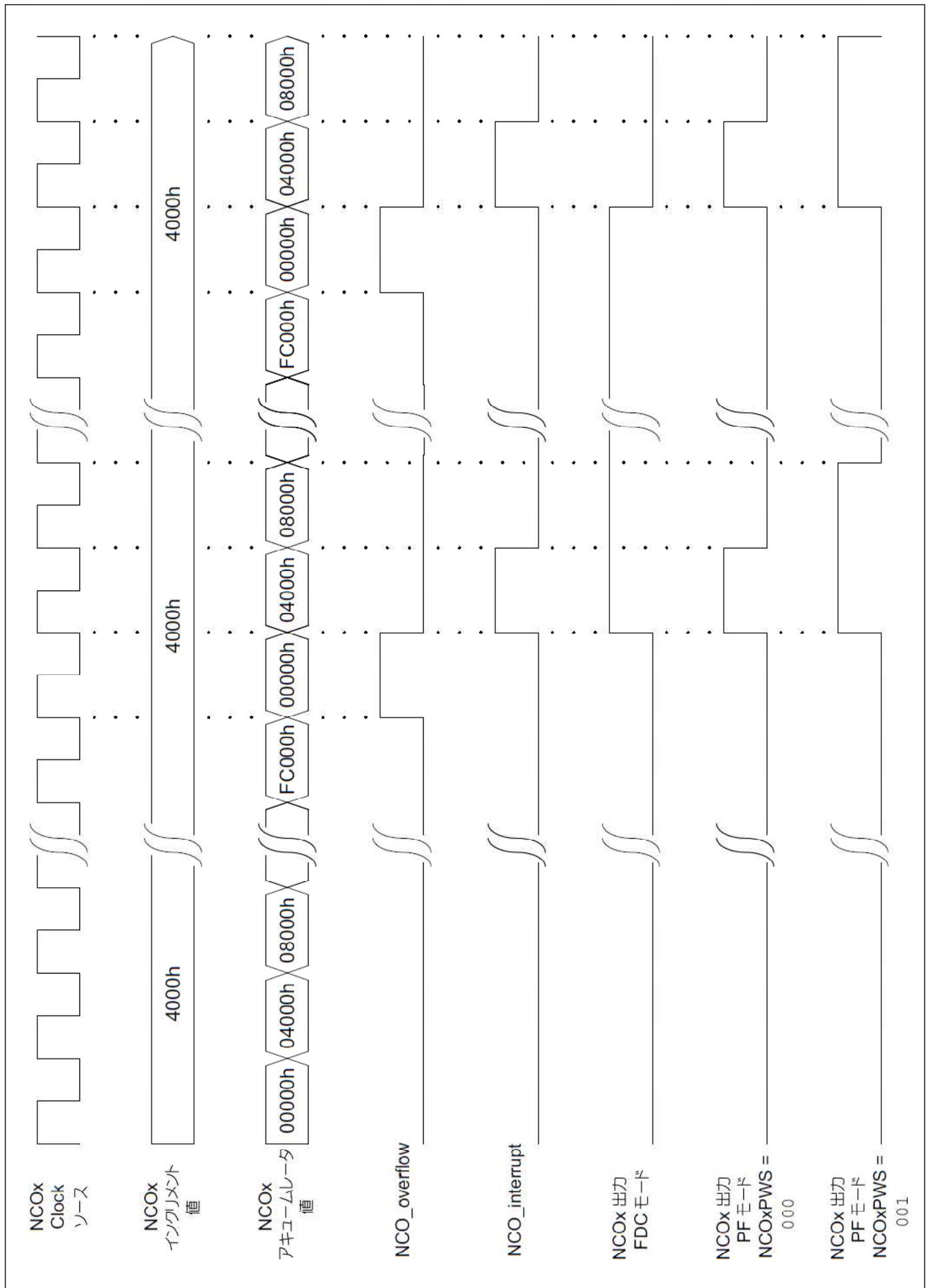
これは、スリープ・モード電流に直接影響します。

## 25.8 代替ピン位置

このモジュールには、オルタネート (代替) ピン機能レジスタAPFCONを使用して他の場所に移動できるI/Oピンが組み込まれています。移動できるピンと、リセット時のデフォルトの位置を決定するには、詳細についてセクション11.1「代替ピン機能」を参照してください。

# PIC16(L)F1508/9

図25-2: NCO—固定デューティ・サイクル(FDC)およびパルス周波数モード(PFM)の出力動作図



## 25.9 レジスタ定義: NCOx 制御レジスタ

### レジスタ25-1: NCOxCON: NCOx 制御レジスタ

R/W-0/0	R/W-0/0	R-0/0	R/W-0/0	U-0	U-0	U-0	R/W-0/0
NxEN	NxOE	NxOUT	NxPOL	—	—	—	NxPFM
bit 7							bit 0

<b>記録:</b>		
R =読み取り可能ビット	W =書き込み可能ビット	U =実装されていないビット、「0」として読み取られる
u =ビットは変更されていません	x =ビットは不明です	-n/n =PORでの値およびBOR/他のすべてのリセットでの値
[1]=ビットが設定されている	[0]=ビットがクリアされている	

- bit 7 **NxEN**: NCOx 許可ビット  
1 = NCOx モジュールは有効にされている  
0 = NCOx モジュールは無効にされている
- bit 6 **NxOE**: NCOx出力許可ビット  
1 = NCOx 出力ピンは有効にされている  
0 = NCOx 出力ピンは無効にされている
- bit 5 **NxOUT**: NCOx出力ビット  
1 = NCOx 出力はHigh  
0 = NCOx 出力はLow
- bit 4 **NxPOL**: NCOx 極性ビット  
1 = NCOx 出力信号はアクティブLow (反転)  
0 = NCOx 出力信号はアクティブHigh (非反転)
- bit 3~1 **未実装**: 「0」として読み取ります
- bit 0 **NxPFM**: NCOx パルス周波数モード・ビット  
1 = NCOx はパルス周波数モードで動作します  
0 = NCOx は固定デューティ・サイクル・モードで動作します

### レジスタ25-2: NCOxCLK: NCOx 入力クロック制御レジスタ

R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	R/W-0/0	R/W-0/0
NxPWS<2:0> <sup>(1,2)</sup>			—	—	—	NxCKS<1:0>	
bit 7							bit 0

<b>記録:</b>		
R =読み取り可能ビット	W =書き込み可能ビット	U =実装されていないビット、「0」として読み取られる
u =ビットは変更されていません	x =ビットは不明です	-n/n =PORでの値およびBOR/他のすべてのリセットでの値
[1]=ビットが設定されている	[0]=ビットがクリアされている	

- bit 7~5 **NxPWS<2:0>**: NCOx 出力パルス幅選択ビット<sup>(1,2)</sup>  
111 = 128 NCOx クロック周期  
110 = 64 NCOx クロック周期  
101 = 32 NCOx クロック周期  
100 = 16 NCOx クロック周期  
011 = 8 NCOx クロック周期  
010 = 4 NCOx クロック周期  
001 = 2 NCOx クロック周期  
000 = 1 NCOx クロック周期
- bit 4~2 **未実装**: 「0」として読み取ります
- bit 1~0 **NxCKS<1:0>**: NCOx クロック・ソース選択ビット  
11 = NCO1CLKピン  
10 = LC1\_out  
01 = Fosc  
00 = HFINTOSC (16 MHz)

注1: NxPWSは、パルス周波数モードで動作している場合にのみ適用されます。  
 2: NCOx パルス幅がNCO\_overflow周期よりも大きい場合、動作は不確定です。

# PIC16(L)F1508/9

## レジスタ-25-3: NCOxACC<sub>L</sub>: NCOx アキュムレータ・レジスタ—Lowバイト

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
NCOxACC<7:0>							
bit 7							bit 0

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
[1]=ビットが設定されている      [0]=ビットがクリアされている

bit 7~0    **NCOxACC<7:0>**: NCOx アキュムレータ、Lowバイト

## レジスタ-25-4: NCOxACC<sub>H</sub>: NCOx アキュムレータ・レジスタ—Highバイト

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
NCOxACC<15:8>							
bit 7							bit 0

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
[1]=ビットが設定されている      [0]=ビットがクリアされている

bit 7~0    **NCOxACC<15:8>**: NCOx アキュムレータ、Highバイト

## レジスタ-25-5: NCOxACC<sub>U</sub>: NCOx アキュムレータ・レジスタ—上位バイト

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	NCOxACC<19:16>			
bit 7							bit 0

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
[1]=ビットが設定されている      [0]=ビットがクリアされている

bit 7~4    **未実装**: 「0」として読み取ります

bit 3~0    **NCOxACC<19:16>**: NCOx アキュムレータ、上位バイト

## レジスタ-25-6: NCOxINCL: NCOx インクリメント・レジスタ—Lowバイト<sup>(1)</sup>

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-1/1
NCOxINC<7:0>							
bit 7							bit 0

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている

bit 7~0    **NCOxINC<7:0>**: NCOx インクリメント、Lowバイト

注1: 最初にNCOxINCHレジスタを書き込み、次にNCOxINCLレジスタを書き込みます。詳細については、25.1.4「インクリメント・レジスタ」を参照してください。

## レジスタ-25-7: NCOxINCH: NCOx インクリメント・レジスタ—Highバイト<sup>(1)</sup>

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
NCOxINC<15:8>							
bit 7							bit 0

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている

bit 7~0    **NCOxINC<15:8>**: NCOx インクリメント、Highバイト

注1: 最初にNCOxINCHレジスタを書き込み、次にNCOxINCLレジスタを書き込みます。詳細については、25.1.4「インクリメント・レジスタ」を参照してください。

## 表25-1: NCOx に関連付けられているレジスタの概要

名前	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ページ上の登録
APFCON	—	—	—	SSSEL	T1GSEL	—	CLC1SEL	NCO1SEL	107
INTCON	GIE	PEIE	TMR0IE	INTE	IOCF	TMR0IF	INTF	IOCF	75
NCO1ACCH	NCO1ACC<15:8>								278
NCO1ACCL	NCO1ACC<7:0>								278
NCO1ACCU	—				NCO1ACC<19:16>				278
NCO1CLK	N1PWS<2:0>			—	—	—	N1CKS<1:0>		277
NCO1CON	N1EN	N1OE	N1OUT	N1POL	—	—	—	N1PFM	277
NCO1INCH	NCO1INC<15:8>								279
NCO1INCL	NCO1INC<7:0>								279
PIE2	OSFIE	C2IE	C1IE	—	BCL1IE	NCO1IE	—	—	77
PIR2	OSFIF	C2IF	C1IF	—	BCL1IF	NCO1IF	—	—	80
TRISA	—	—	TRISA5	TRISA4	— <sup>(1)</sup>	TRISA2	TRISA1	TRISA0	109
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	117

記録: x =不明。u =変更なし。— =未実装の読み取りは「0」。q =値は条件によって異なります。網掛けのセルはNCOxモジュールには使用されません。

注1: 未実装、「1」と読みます。

## 26.0 補数波形発生器(CWG)モジュール

コンプリメンタリー・ウェーブフォーム・ジェネレーター (CWG) (補数波形発生器) は、選択した入力ソースからデッドバンド (不感帯) 遅延のある補数波形を生成します。

CWGモジュールには次の機能があります：

- 選択可能なデッドバンド・クロック・ソース制御
- 選択可能な入力ソース
- 出力許可制御
- 出力極性制御
- 独立した6ビットの立ち上がりエッジと立ち下がりエッジのデッドバンド・カウンタによる不感帯制御
- 自動シャットダウン制御：
  - 選択可能なシャットダウン・ソース
  - 自動再起動の有効化
  - 自動シャットダウン・ピン上書き制御

### 26.1 基本的な動作

CWGは、選択した入力ソースから2つの出力波形を生成します。

各出力のオフからオンへの遷移は、他の出力のオンからオフへの遷移から遅延する可能性があります、それによって、どちらの出力も駆動されない場合にすぐに時間遅延が発生します。これはデッドタイムと呼ばれ、セクション26.5「デッドバンド制御」で説明されています。図26-2に、単一の入力信号から生成されたデッドバンドの典型的な動作波形を示します。

回路障害やフィードバック・イベントの到着が遅すぎるか、まったく到着しない可能性を防ぐ必要がある場合があります。この場合、障害状態が損傷を引き起こす前に、アクティブ・ドライブを終了する必要があります。これは自動シャットダウンと呼ばれ、セクション26.9「自動シャットダウン制御」で説明されています。

### 26.2 クロック・ソース

CWGモジュールでは、次のクロック・ソースを選択できます：

- Fosc (システム・クロック)
- HFINTOSC (16 MHzのみ)

クロック・ソースは、CWGxCON0レジスター (レジスター26-1) のG1CS0ビットを使用して選択されます。

## 26.3 選択可能な入力ソース

CWGは、表26-1の入力ソースから出力波形を生成します。

表26-1: 選択可能な入力ソース

ソース周辺	信号名
コンパレーターC1	C1OUT_sync
コンパレーターC2	C2OUT_sync
PWM1	PWM1_out
PWM2	PWM2_out
PWM3	PWM3_out
PWM4	PWM4_out
NCO1	NCO1_out
CLC1	LC1_out

入力ソースは、CWGxCON1レジスター (レジスター26-2) のGxIS<2:0>ビットを使用して選択されます。

### 26.4 出力制御

CWGモジュールが有効になった直後に、CWGxAドライブとCWGxBドライブの両方がクリアされた状態で相補ドライブが構成されます。

#### 26.4.1 出力許可

各CWG出力ピンには個別の出力許可制御があります。出力許可は、CWGxCON0レジスターのGxOEAビットとGxOEBビットで選択されます。出力許可制御がクリアされると、モジュールはピンに対する制御をアサート (論理有効) しません。出力許可が設定されている場合、ポート優先順位の選択に従って、上書き値またはアクティブPWM波形がピンに適用されず、出力ピンの許可は、モジュールの許可ビットGxENに依存します。GxENがクリアされると、CWG出力が有効になり、CWGドライブ・レベルは効果がありません。

#### 26.4.2 極性制御

各CWG出力の極性は個別に選択できます。出力極性ビットがセットされると、対応する出力がアクティブHighになります。出力極性ビットをクリアすると、対応する出力がアクティブLowとして構成されます。ただし、極性は上書きレベルには影響しません。出力極性は、CWGxCON0レジスターのGxPOLAビットとGxPOLBビットで選択されます。

図26-1: 簡略化されたCWGブロック図

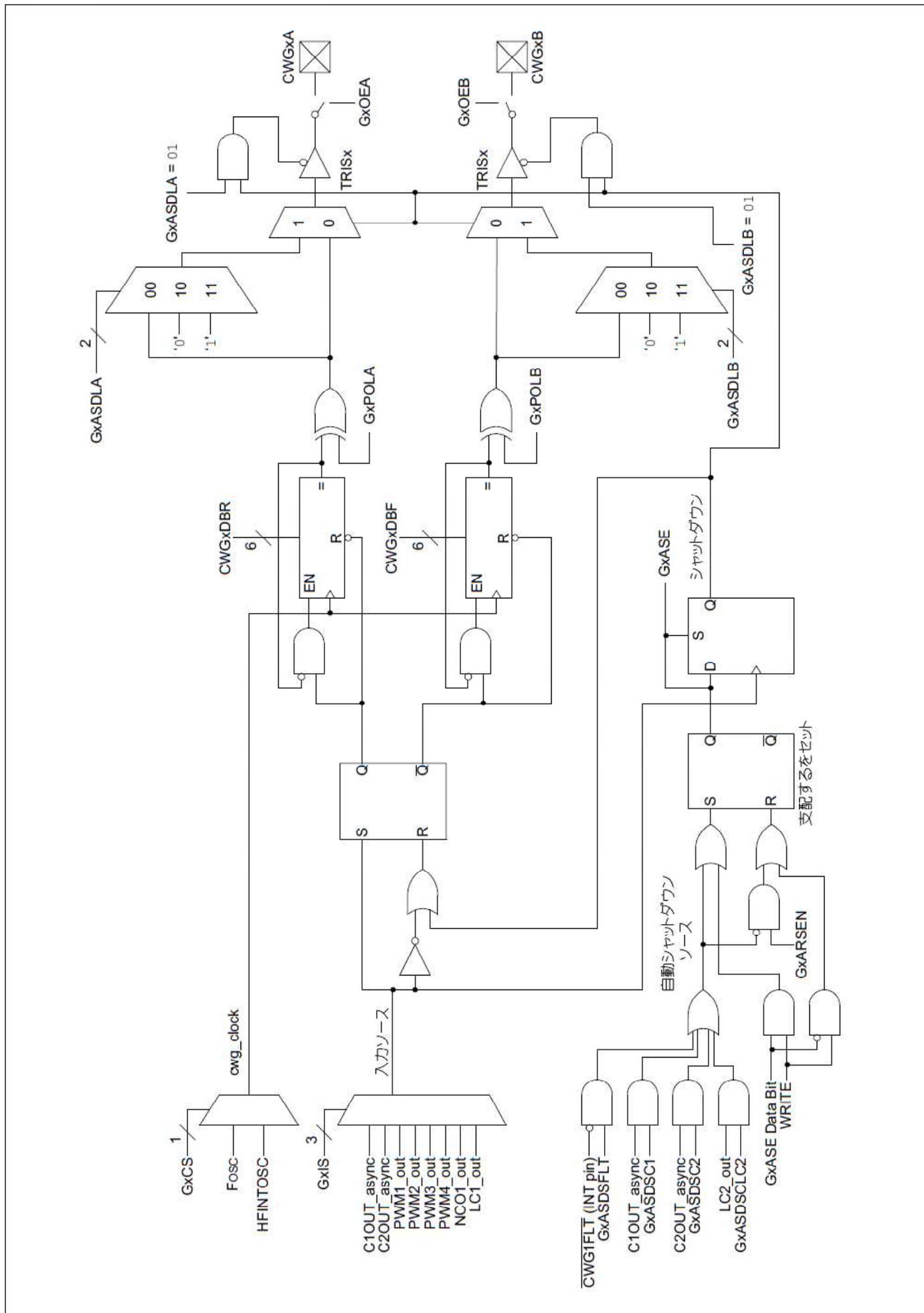
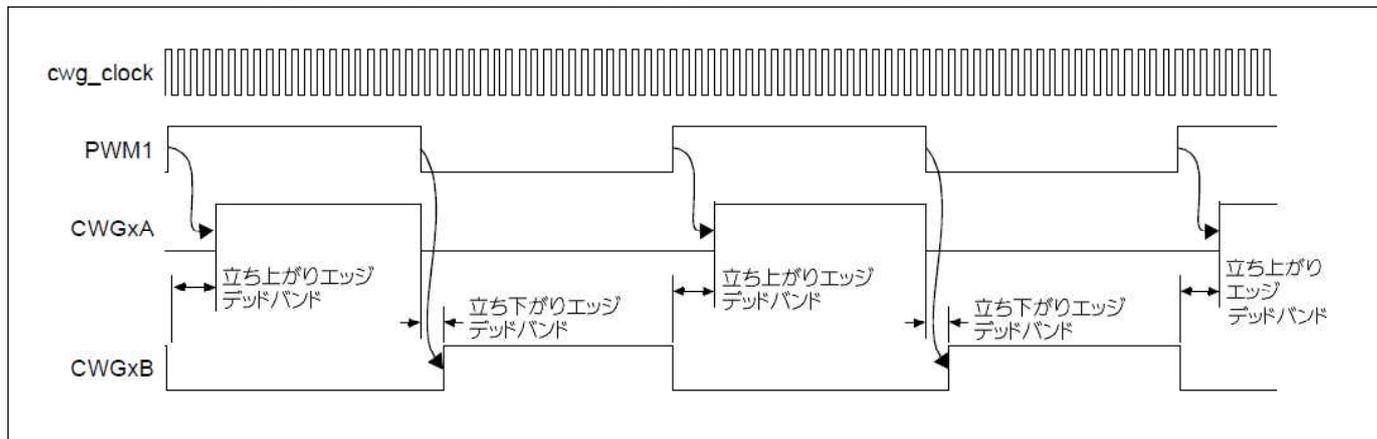


図26-2: PWM1を使用した典型的なCWG操作(自動シャットダウンなし)



## 26.5 デッドバンド制御

デッドバンド制御は、電源スイッチのシュートスルー電流を防ぐために、重複しない出力信号を提供します。CWGには、2つの6ビット・デッドバンド・カウンタが含まれています。1つのデッドバンド・カウンタが入カソース管理の立ち上がりエッジに使用されます。もう1つは、入カソース管理の立ち下がりエッジに使用されます。

デッドバンドは、ゼロから立ち上がりまたは立ち下がりデッドバンド・カウンタ・レジスタの値までのCWGクロック周期をカウントすることによって計時されます。CWGxDBRおよびCWGxDBFレジスタ（それぞれレジスタ26-4およびレジスタ26-5）を参照してください。

## 26.6 立ち上がりエッジ・デッドバンド

立ち上がりエッジのデッドバンドは、CWGxB出力がオフになったときからCWGxA出力のターンオンを遅らせます。立ち上がりエッジのデッドバンド時間は、入カソース信号の立ち上がりエッジが真になると始まります。これが発生すると、CWGxB出力はすぐにオフになり、立ち上がりエッジのデッドバンド遅延時間が始まります。立ち上がりエッジのデッドバンド遅延時間に達すると、CWGxA出力がオンになります。

CWGxDBRレジスタは、入カソース信号の立ち上がりエッジでのデッドバンド間隔の持続時間を設定します。この期間は、デッドバンドの0~64カウントです。

デッドバンドは常に入カソース信号のエッジからカウントされます。カウント0（ゼロ）は、デッドバンドが存在しないことを示します。

カウントが完了するのに十分な時間入カソース信号が存在しない場合、それぞれの出力に出力は表示されません。

## 26.7 立ち下がりエッジ・デッドバンド

立ち下がりエッジのデッドバンドは、CWGxA出力がオフになったときからCWGxB出力のターンオンを遅らせます。立ち下がりエッジの不感帯時間は、入カソースの立ち下がりエッジが真になったときに始まります。これが発生すると、CWGxA出力はすぐにオフになり、立ち下がりエッジの不感帯遅延時間が始まります。立ち下がりエッジの不感帯遅延時間に達すると、CWGxB出力がオンになります。

CWGxDBFレジスタは、入カソース信号の立ち下がりエッジでのデッドバンド間隔の持続時間を設定します。この期間は、デッドバンドの0~64カウントです。

デッドバンドは常に入カソース信号のエッジからカウントされます。カウント0（ゼロ）は、デッドバンドが存在しないことを示します。

カウントが完了するのに十分な時間入カソース信号が存在しない場合、それぞれの出力に出力は表示されません。

例については、図26-3および図26-4を参照してください。

図26-3: デッドバンド動作、CWGxDBR = 01H、  
CWGxDBF = 02H

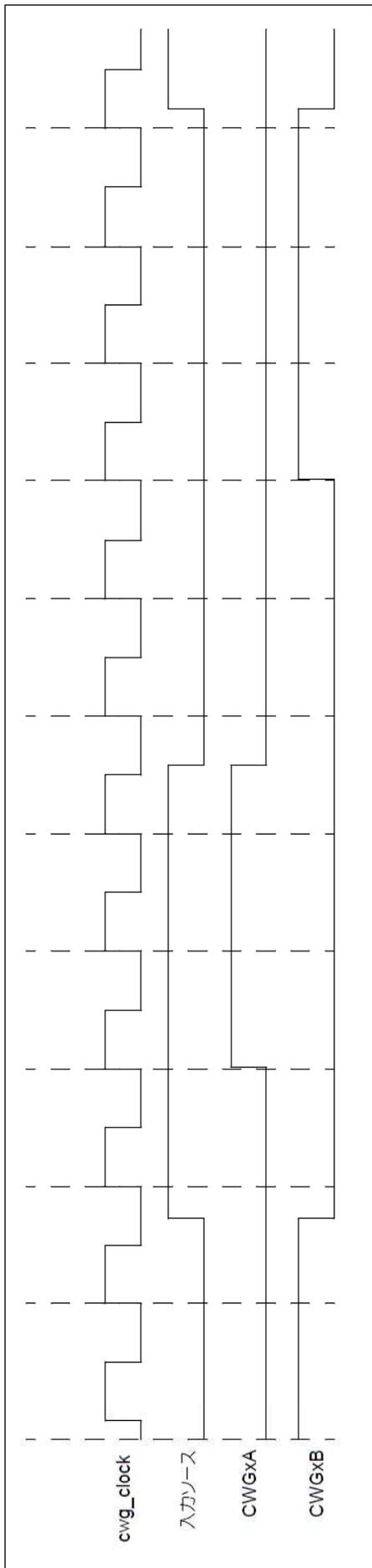
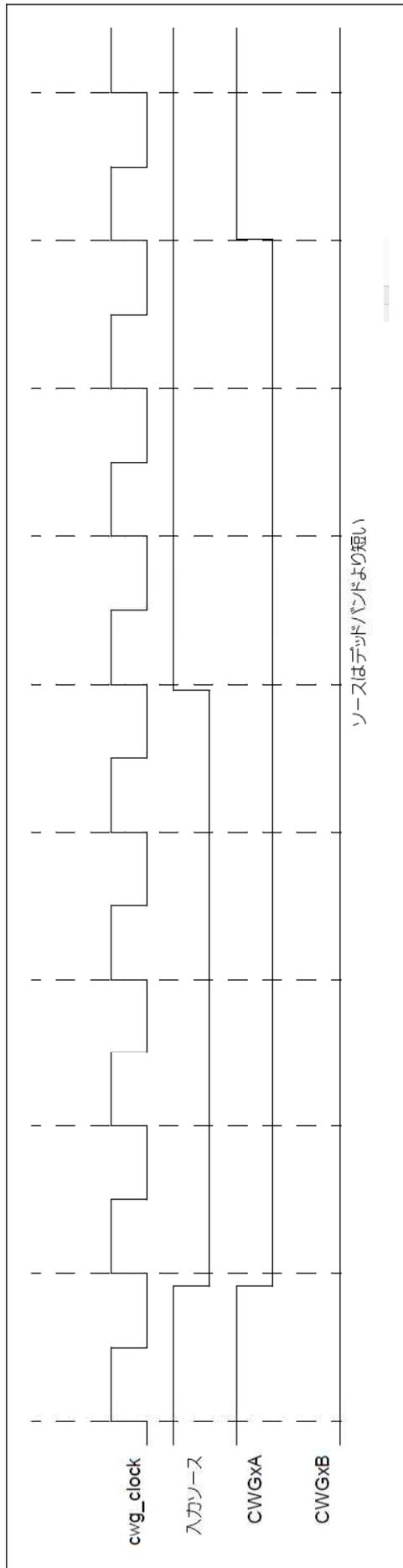


図26-4: デッドバンド動作、CWGxDBR = 03H、  
CWGxDBF = 04H、デッドバンドよりもソ  
ースが短い



## 26.8 デッドバンドの不確実性

入力ソースの立ち上がりエッジと立ち下がりエッジがデッドバンド・カウンタをトリガーすると、入力が非同期になる場合があります。これにより、デッドバンドの時間遅延に不確実性が生じます。最大の不確かさは、1 CWGクロック周期に等しくなります。詳細については、式26-1を参照してください。

### 式26-1: デッドバンドの不確実性

$$T_{\text{デッドバンド\_不確実性}} = \frac{1}{F_{\text{cwg\_clock}}}$$

例:

$$F_{\text{cwg\_clock}} = 16 \text{ MHz}$$

したがって:

$$\begin{aligned} T_{\text{デッドバンド\_不確実性}} &= \frac{1}{F_{\text{cwg\_clock}}} \\ &= \frac{1}{16 \text{ MHz}} \\ &= 62.5 \text{ ns} \end{aligned}$$

## 26.9 自動シャットダウン制御

自動シャットダウンは、回路の安全なシャットダウンを可能にする特定の上書きでCWG出力レベルを即座に上書きする方法です。シャットダウン状態は、自動的にクリアするか、ソフトウェアによってクリアされるまで保持することができます。

### 26.9.1 シャットダウン

シャットダウン状態は、次の2つの方法のいずれかで開始できます:

- 生成されたソフトウェア
- 外部入力

#### 26.9.1.1 ソフトウェア生成シャットダウン

CWGxCON2レジスタのGxASEビットをセットすると、CWGは強制的にシャットダウン状態になります。

自動再起動が無効になっている場合、GxASEビットが設定されている限りシャットダウン状態が持続します。

自動再起動が有効になっている場合、GxASEビットは自動的にクリアされ、次の立ち上がりエッジ・イベントで動作を再開します。図26-6を参照してください。

#### 26.9.1.2 外部入力ソース

外部シャットダウン入力は、障害状態が発生した場合にCWG操作を安全に一時停止するための最速の方法を提供します。選択したシャットダウン入力のいずれかがアクティブになると、CWG出力はソフトウェアの遅延なしに選択した上書きレベルにすぐに移行します。2つの入力ソースの任意の組み合わせを選択して、シャットダウン状態を発生させることができます。ソースは:

- Comparator C1 — C1OUT\_async
- Comparator C2 — C2OUT\_async
- CLC2 — LC2\_out
- CWG1FLT

シャットダウン入力は、CWGxCON2レジスタで選択されます。(レジスタ26-3)。

**注釈:** シャットダウン入力はレベル・センシティブ（レベル入力）であり、エッジ・センシティブではありません。シャットダウン入力レベルが持続する限り、自動シャットダウンを無効にする場合を除いて、シャットダウン状態をクリアすることはできません。

## 26.10 スリープ中の動作

CWGモジュールは、システム・クロックとは独立して動作し、選択したクロックと入力ソースがアクティブのままである限り、スリープ中も実行を継続します。

CWGモジュールが有効で、入力ソースがアクティブであり、選択されたシステム・クロック・ソースに関係なく、HFINTOSCがクロック・ソースとして選択されている場合、HFINTOSCはスリープ中もアクティブのままです。

つまり、HFINTOSCがシステム・クロックとCWGクロック・ソースとして同時に選択されている場合、CWGが有効で入力ソースがアクティブな場合、CPUはスリープ中にアイドル状態になりますが、CWGは動作を継続し、HFINTOSCは動作し続けます。アクティブなままになります。

これは、スリープ・モード電流に直接影響します。

## 26.11 CWGの構成

次の手順は、同期開始を確実にするためにCWGを適切に構成する方法を示しています：

1. CWGxAとCWGxBに対応するTRIS制御ビットが、両方が入力として構成されるように設定されていることを確認します。
2. まだクリアされていない場合は、GxENビットをクリアします。
3. CWGxDBRおよびCWGxDBFレジスターを使用して必要なデッドバンド時間を設定します。
4. CWGxCON2自動シャットダウン・レジスターに次の制御を設定します。
  - 目的のシャットダウン・ソースを選択します。
  - 両方の出力上書きを目的のレベルに選択します（これは、自動シャットダウンを使用しない場合でも、起動はシャットダウン状態から行われるため必要です）。
  - GxASEビットを設定し、GxARSENビットをクリアします。
5. CWGxCON1レジスターを使用して、目的の入力ソースを選択します。
6. CWGxCON0レジスターで次の制御を構成します。
  - 目的のクロック・ソースを選択します。
  - 目的の出力極性を選択します。
  - 出力を設定して、出力を使用できるようにします。
7. GxENビットを設定します。
8. CWGxAおよびCWGxBに対応するTRIS制御ビットをクリアして、これらのピンを出力として構成するために使用します。
9. 自動再起動を使用する場合は、GxARSENビットを設定すると、GxASEビットが自動的にクリアされます。それ以外の場合は、GxASEビットをクリアしてCWGを開始します。

## 26.11.1 ピン上書きレベル

シャットダウン入力が真のときに出力ピンに駆動されるレベルは、CWGxCON1レジスター（レジスター26-3）のGxASDLAビットとGxASDLBビットによって制御されます。GxASDLAはCWG1A上書きレベルを制御し、GxASDLBはCWG1B上書きレベルを制御します。制御ビットの論理レベルは、シャットダウン状態の出力論理ドライブ・レベルに対応します。極性制御は上書きレベルには適用されません。

## 26.11.2 自動シャットダウンの再起動

自動シャットダウン・イベントが発生した後、操作を再開するには2つの方法があります：

- ソフトウェア制御
- 自動再起動

再起動方法は、CWGxCON2レジスターのGxARSENビットで選択されます。ソフトウェア制御および自動再起動の波形を図26-5および図26-6に示します。

### 26.11.2.1 ソフトウェア制御再起動

CWGxCON2レジスターのGxARSENビットがクリアされると、ソフトウェアによる自動シャットダウン・イベントの後にCWGを再起動する必要があります。

シャットダウン状態をクリアするには、選択したすべてのシャットダウン入力をLowにする必要があります。Lowでない場合、GxASEビットはセットされたままになります。上書きは、GxASEビットがクリアされた後の最初の立ち上がりエッジ・イベントまで有効です。その後、CWGは動作を再開します。

### 26.11.2.2 自動再起動

CWGxCON2レジスターのGxARSENビットがセットされると、CWGは自動シャットダウン状態から自動的に再起動します。

すべてのシャットダウン・ソースがLowになると、GxASEビットは自動的にクリアされます。上書きは、GxASEビットがクリアされた後の最初の立ち上がりエッジ・イベントまで有効です。その後、CWGは動作を再開します。

# PIC16(L)F1508/9

図26-5: シャットダウン機能、自動再起動が無効  
(GxARSEN = 0、GxASDLA = 01、  
GxASDLB = 01)

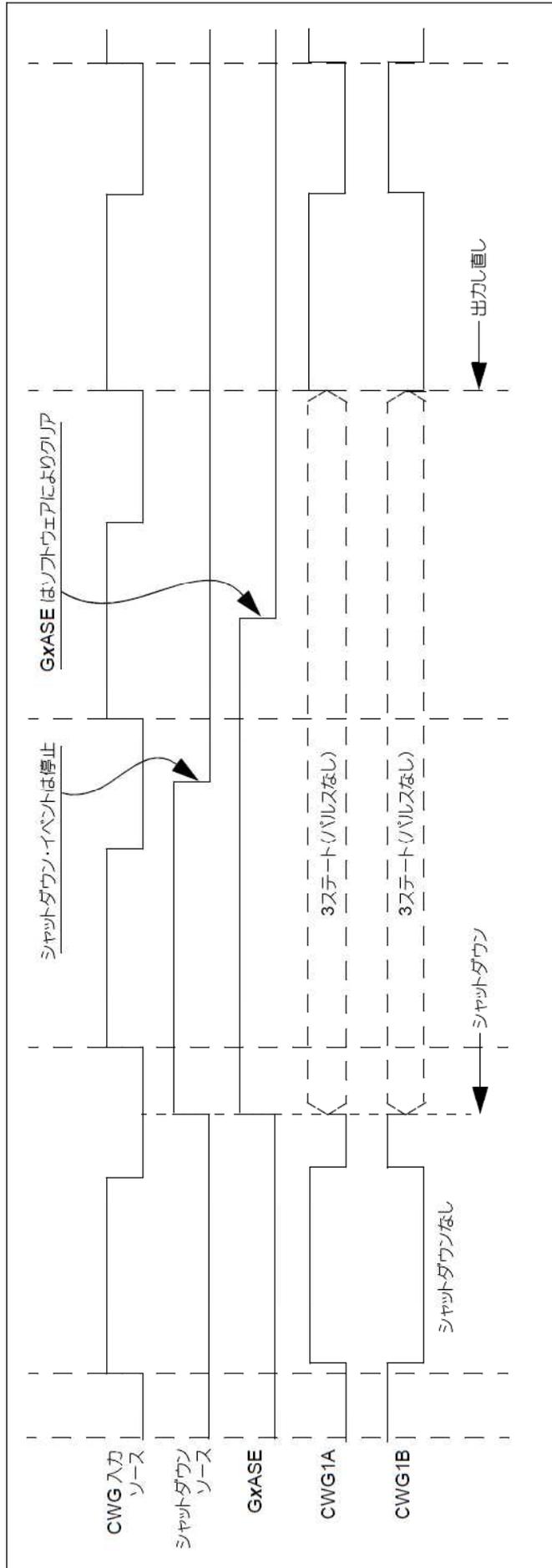
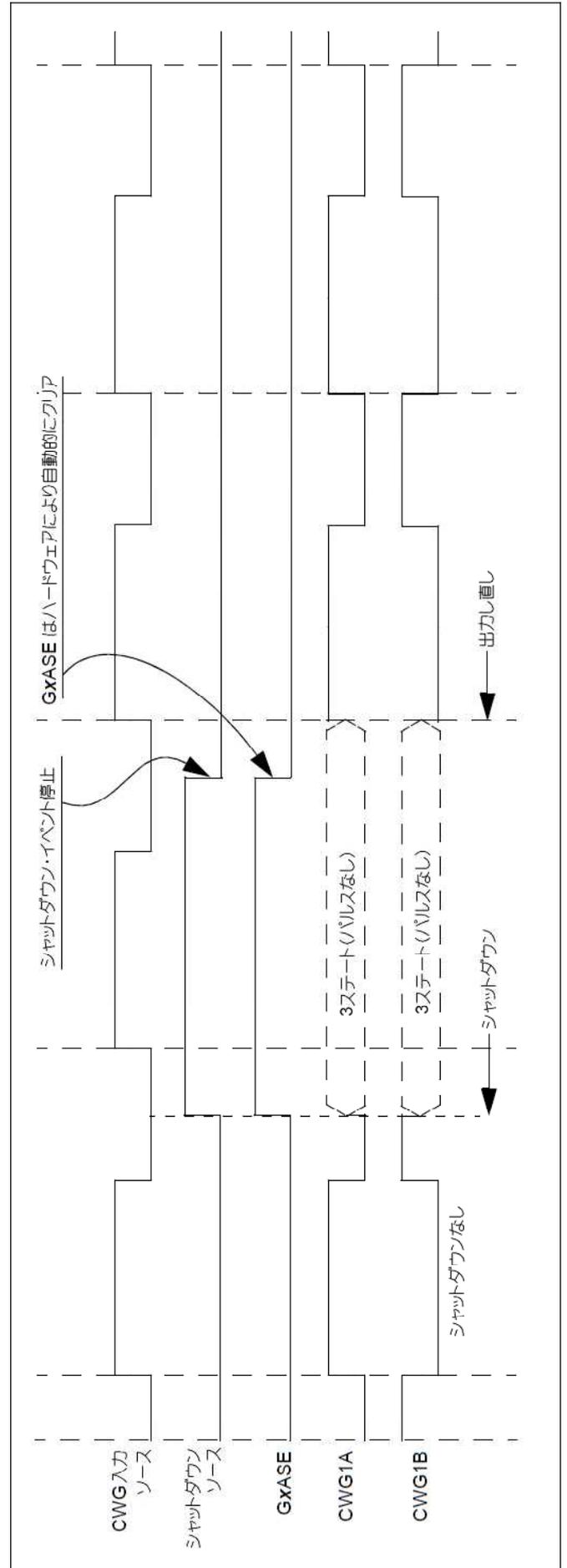


図26-6: シャットダウン機能、自動再起動が有効  
(GxARSEN = 1、GxASDLA = 01、  
GxASDLB = 01)



## 26.12 Register Definitions: CWG Control

### レジスタ-26-1: CWGxCON0: CWG制御レジスタ-0

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0
GxEN	GxOEB	GxOEA	GxPOLB	GxPOLA	—	—	GxCS0
bit 7							bit 0

<b>記録:</b>		
R =読み取り可能ビット	W =書き込み可能ビット	U =実装されていないビット、「0」として読み取られる
u =ビットは変更されていません	x =ビットは不明です	-n/n =PORでの値およびBOR/他のすべてのリセットでの値
[1]=ビットが設定されている	[0]=ビットがクリアされている	q =値は条件によって異なります

- bit 7     **GxEN** : CWGx許可ビット  
           1 = モジュールは有効にされている  
           0 = モジュールは無効にされている
- bit 6     **GxOEB** : CWGxB出力許可ビット  
           1 = CWGxBは適切なI/Oピンで利用可能です  
           0 = CWGxBは適切なI/Oピンで利用可能ではありません
- bit 5     **GxOEA** : CWGxA出力許可ビット  
           1 = CWGxAは適切なI/Oピンで利用可能です  
           0 = CWGxAは適切なI/Oピンで利用可能ではありません
- bit 4     **GxPOLB** : CWGxB出力極性ビット  
           1 = 出力は反転された極性です  
           0 = 出力は通常の極性です
- bit 3     **GxPOLA** : CWGxA出力極性ビット  
           1 = 出力は反転された極性です  
           0 = 出力は通常の極性です
- bit 2~1   **未実装** : 「0」として読み取ります
- bit 0     **GxCS0** : CWGxクロック・ソース選択ビット  
           1 = HFINTOSC  
           0 = FOSC

# PIC16(L)F1508/9

## レジスタ-26-2: CWGxCON1: CWG制御レジスタ-1

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	U-0	R/W-0/0	R/W-0/0	R/W-0/0
GxASDLB<1:0>		GxASDLA<1:0>		—	GxIS<2:0>		
bit 7							bit 0

### 記録:

R =読み取り可能ビット	W =書き込み可能ビット	U =実装されていないビット、「0」として読み取られる
u =ビットは変更されていません	x =ビットは不明です	-n/n =PORでの値およびBOR/他のすべてのリセットでの値
[1]=ビットが設定されている	[0]=ビットがクリアされている	q =値は条件によって異なります

- bit 7~6 **GxASDLB<1:0>**: CWGxBのBCWGxシャットダウン状態  
 自動シャットダウンイベントが存在する場合 (GxASE = 1) :  
 11 = CWGxBピンは「1」に駆動されます、GxPOLBビットの設定に関係なく。  
 10 = CWGxBピンは「0」に駆動されます、GxPOLBビットの設定に関係なく。  
 01 = CWGxBピンは3ステートです  
 00 = CWGxBピンは、選択されたデッドバンド間隔の後、非アクティブ状態に駆動されます。GxPOLBは引き続き出力の極性を制御します。
- bit 5~4 **GxASDLA<1:0>**: CWGxAのCWGxシャットダウン状態  
 自動シャットダウンイベントが存在する場合 (GxASE = 1) :  
 11 = CWGxAピンは「1」にドライブされます、GxPOLAビットの設定に関係なく。  
 10 = CWGxAピンは「0」にドライブされます、GxPOLAビットの設定に関係なく。  
 01 = CWGxAピンは3ステートです  
 00 = CWGxAピンは、選択されたデッドバンド間隔の後、非アクティブ状態に駆動されます。GxPOLAは引き続き出力の極性を制御します。
- bit 3 **未実装**: 「0」として読み取ります
- bit 2~0 **GxIS<2:0>**: CWGx入力ソース選択ビット  
 111 = CLC1 — LC1\_out  
 110 = NCO1 — NCO1\_out  
 101 = PWM4 — PWM4\_out  
 100 = PWM3 — PWM3\_out  
 011 = PWM2 — PWM2\_out  
 010 = PWM1 — PWM1\_out  
 001 = コンパレータ-C2 — C2OUT\_async  
 000 = コンパレータ-C1 — C1OUT\_async

## レジスタ-26-3: CWGxCON2: CWG制御レジスタ-2

R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
GxASE	GxARSEN	—	—	GxASDSC2	GxASDSC1	GxASDSFLT	GxASDSC2
bit 7							bit 0

## 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
「1」=ビットが設定されている      「0」=ビットがクリアされている      q =値は条件によって異なります

- bit 7      **GxASE**: 自動シャットダウン・イベントス・ステータス・ビット  
1 = 自動シャットダウン・イベントが発生されました  
0 = 自動シャットダウン・イベントが発生されてません
- bit 6      **GxARSEN**: 自動リセット許可ビット  
1 = 自動リセットは有効とされています  
0 = 自動リセットは無効とされています
- bit 5~4      **未実装**: 「0」として読み取ります
- bit 3      **GxASDSC2**: コンパレータ-C2でのCWG自動シャットダウン許可ビット  
1 = シャットダウンはコンパレータ-C2出力 (C2OUT\_async) がHighの場合  
0 = コンパレータ-C2出力はシャットダウンでは影響はなし
- bit 2      **GxASDSC1**: コンパレータ-C1でのCWG自動シャットダウン許可ビット  
1 = シャットダウンはコンパレータ-C1出力 (C1OUT\_async) がHighの場合  
0 = コンパレータ-C1出力はシャットダウンでは影響はなし
- bit 1      **GxASDSFLT**: FLTでのCWG自動シャットダウン許可ビット  
1 = シャットダウンはCWG1FLT入力がLowの場合  
0 = CWG1FLT入力はシャットダウンでの影響はなし
- bit 0      **GxASDSC2**: CLC2でのCWG自動シャットダウン許可ビット  
1 = シャットダウンはCLC2出力 (LC2\_out) がHighの場合  
0 = CLC2出力はシャットダウンでは影響なし

# PIC16(L)F1508/9

## レジスタ26-4: CWGxDBR: 補数型波形発生器(CWGx)立ち上がりデッドバンド・カウント・レジスタ

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
CWXDBR<5:0>							
bit 7							bit 0

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
[1]=ビットが設定されている      [0]=ビットがクリアされている      q =値は条件によって異なります

bit 7~6    **未実装**: 「0」と読みます

bit 5~0    **CWXDBR<5:0>**: 補数波形発生器(CWGx)立ち上がりカウント

11 1111 = 63~64カウント、デッドバンドの

11 1110 = 62~63カウント、デッドバンドの

⋮

⋮

⋮

00 0010 = 2~3カウント、デッドバンドの

00 0001 = 1~2カウント、デッドバンドの

00 0000 = 0カウント、デッドバンドの

## レジスタ26-5: CWGxDBF: 補数波形発生器(CWGx)立ち下がりデッドバンド・カウント・レジスタ

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
CWXDBF<5:0>							
bit 7							bit 0

### 記録:

R =読み取り可能ビット      W =書き込み可能ビット      U =実装されていないビット、「0」として読み取られる  
u =ビットは変更されていません      x =ビットは不明です      -n/n =PORでの値およびBOR/他のすべてのリセットでの値  
[1]=ビットが設定されている      [0]=ビットがクリアされている      q =値は条件によって異なります

bit 7~6    **未実装**: 「0」と読みます

bit 5~0    **CWXDBF<5:0>**: 補数波形発生器(CWGx)立ち下がりカウント

11 1111 = 63~64カウント、デッドバンドの

11 1110 = 62~63カウント、デッドバンドの

⋮

⋮

⋮

00 0010 = 2~3カウント、デッドバンドの

00 0001 = 1~2カウント、デッドバンドの

00 0000 = 0カウント、デッドバンドの。デッドバンド発生はバイパスされます

**表26-2: CWGに関連付けられているレジスターの概要**

名前	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ページ上の登録
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	110
CWG1CON0	G1EN	G1OEB	G1OEA	G1POLB	G1POLA	—	—	G1CS0	287
CWG1CON1	G1ASDLB<1:0>		G1ASDLA<1:0>		—	—	G1IS<1:0>		288
CWG1CON2	G1ASE	G1ARSEN	—	—	G1ASDSC2	G1ASDSC1	G1ASDSFLT	G1ASDSCLC2	289
CWG1DBF	—	—	CWG1DBF<5:0>						290
CWG1DBR	—	—	CWG1DBR<5:0>						290
TRISA	—	—	TRISA5	TRISA4	— <sup>(1)</sup>	TRISA2	TRISA1	TRISA0	109
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	117

**記録:** x = 不明。u = 変更なし。— = 実装されていない場所は「0」として読み取られます。網掛けのセルはCWGでは使用されません。

**注1:** 未実装、「1」と読みます。

# PIC16(L)F1508/9

## 27.0 インサーキット・シリアル・プログラミング™ (ICSP™)

ICSP™プログラミングにより、顧客はプログラムされていないデバイスで回路基板を製造できます。プログラミングは、組み立て過程の後に実行できるため、デバイスを最新のファームウェアまたはカスタム・ファームウェアでプログラミングできます。ICSP™プログラミングには5つのピンが必要です：

- ICSPCLK
- ICSPDAT
- MCLR/VPP
- VDD
- VSS

プログラム/検証（ヴェリファイ）モードでは、プログラムメモリ、ユーザーID、および設定ワードはシリアル通信を介してプログラムされます。ICSPDATピンはシリアルデータの転送に使用される双方向I/Oであり、ICSPCLKピンはクロック入力です。ICSPの詳細については、「PIC12(L)F1501/PIC16(L)F150Xメモリー・プログラミング仕様」（DS41573）を参照してください。

### 27.1 高電圧プログラミング入力モード

ICSPCLKピンとICSPDATピンをLowに保持し、MCLR/VPPの電圧をVIHHに上げるにより、デバイスは高電圧プログラミング・エンタリイ（入力）モードになります。

### 27.2 低電圧プログラミング入力モード

低電圧プログラミング・エンタリイ・モードでは、PIC®フラッシュMCUを、高電圧なしで、VDDのみを使用してプログラミングできます。構成ワードのLVPビットが「1」に設定されている場合、ICSP低電圧プログラミング・エンタリイ・モードが有効になります。低電圧ICSPモードを無効にするには、LVPビットを「0」にプログラムする必要があります。

低電圧プログラミング・エンタリイ・モードに入るには、次の手順が必要です。：

1. MCLRはVILに持ってきます。
2. 32ビット・キー・シーケンスはis presented on ICSPDATで表示されます、ICSPCLKのクロッキング中は。

キー・シーケンスが完了すると、プログラム/検証モードが維持される限り、MCLRはVILで保持される必要があります。

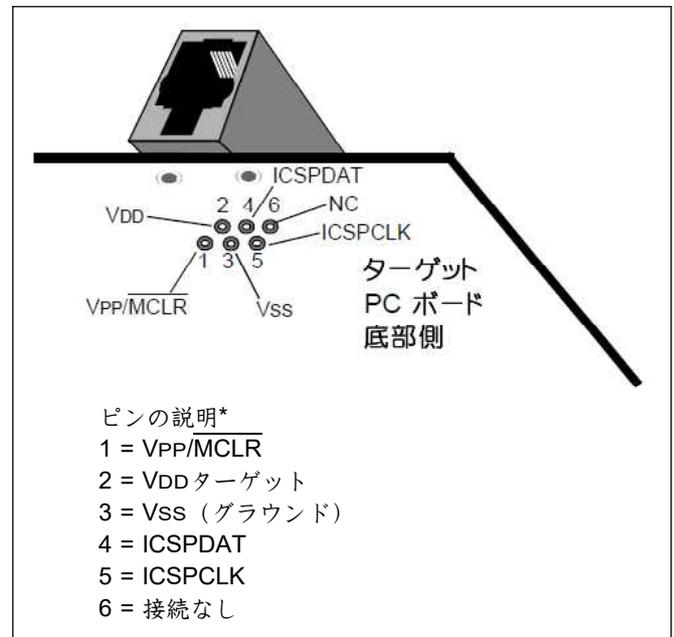
低電圧プログラミングが有効になっている場合（LVP = 1）、MCLRリセット機能は自動的に有効になり、無効にすることはできません。詳細については、セクション6.5「MCLR」を参照してください。

LVPビットは、高電圧プログラミング・モードを使用することによってのみ「0」に再プログラムできます。

## 27.3 一般的なプログラミング・インターフェイス

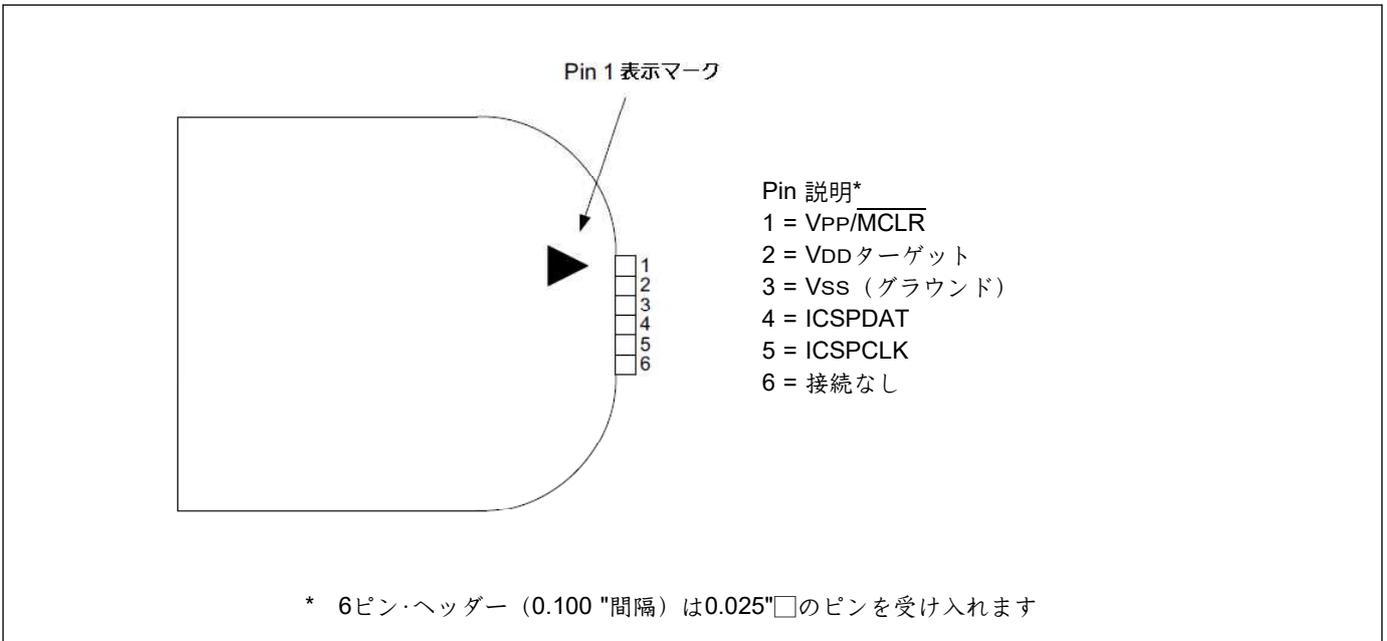
ターゲットデバイスへの接続は、通常、ICSP™ヘッダーを介して行われます。開発ツールで一般的に見られるコネクタは、6P6C（6ピン、6コネクタ）構成のRJ-11です。図27-1を参照してください。

図27-1: ICD RJ-11スタイル・コネクタ・インターフェイス



PICKit™でよく使用される別のコネクタ。プログラマーは、0.1インチ間隔の標準的な6ピン・ヘッダーです。図27-2を参照してください。

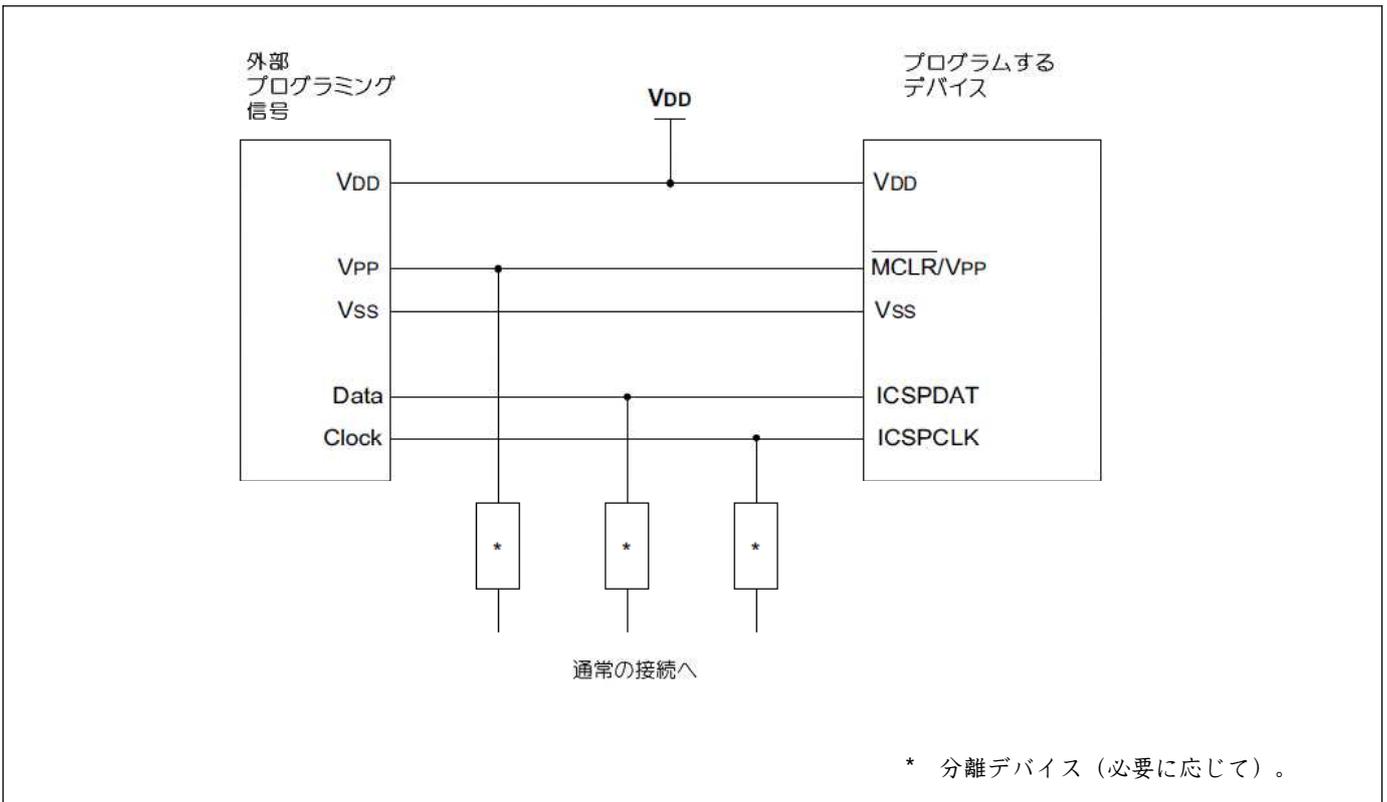
図27-2: PICkit™プログラマー・スタイル・コネクタ・インターフェース



その他の推奨インターフェースについては、PCB設計の前に特定のデバイス・プログラマーのマニュアルを参照してください。

プログラミング・ピンを他の回路から分離するために、絶縁デバイスを使用することをお勧めします。絶縁のタイプは特定のアプリケーションに大きく依存し、抵抗、ダイオード、さらにはジャンパーなどのデバイスが含まれる場合があります。詳細については、図27-3を参照してください。

図27-3: ICSP™プログラミングの一般的な接続



## 28.0 命令セットの概要

各命令は、演算コード（オペコード）と必要なすべてのオペランドを含む14ビット・ワードです。オペコードは大きく3つのカテゴリに分類されます。

- バイト指向
- ビット指向
- リテラル（定数）と制御

リテラルおよび制御カテゴリには、最も多様な命令ワード形式が含まれています。

表28-3に、MPASM™アセンブラによって認識される命令を示します。

以下の例外を除いて、すべての命令は単一の命令サイクル内で実行されます。これには2~3サイクルかかる場合があります：

- サブルーチンには2サイクルかかります（**CALL**、**CALLW**）
- 割り込みまたはサブルーチンからの戻りには2サイクルかかります（**RETURN**、**RETLW**、**RETFIE**）
- プログラムの分岐には2サイクルかかります（**GOTO**、**BRA**、**BRW**、**BTFSS**、**BTFSC**、**DECFSZ**、**INCSFZ**）
- いずれかの命令が間接ファイル・レジスターを参照し、ファイル選択レジスターがプログラム・メモリを指している場合、1つの追加の命令サイクルが使用されます。

1つの命令サイクルは4つの発信器サイクルで構成されます。発振器周波数が4 MHzの場合、これにより公称命令実行レートは1MHzになります。

すべての命令例では、「0xhh」の形式を使用して16進数を表します。ここで、「h」は16進数を示します。

## 28.1 リード・モディファイ・ライト動作（操作）

命令の一部としてファイル・レジスターを指定する命令は、リード・モディファイ・ライト（R-M-W）操作を実行します。レジスターが読み取られ、データが変更され、命令または宛先指定子「d」のいずれかに従って結果が格納されます。命令がそのレジスタに書き込んだ場合でも、そのレジスタに対して読み取り操作が実行されます。

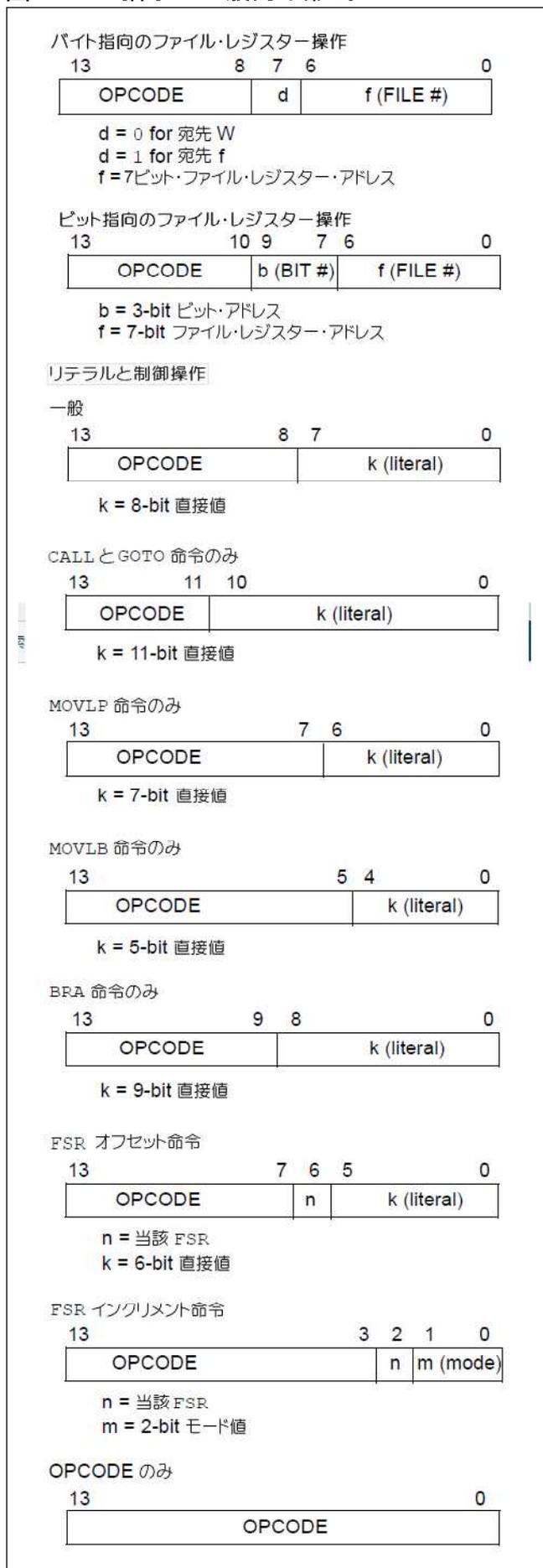
表28-1: オペコード・フィールド(場)の説明

場	説明
f	レジスター・ファイル・アドレス (0x00 ~ 0x7F)
W	ワーキング・レジスター (アキュムレーター)
b	ビット・アドレス : 8ビット・ファイル・レジスター内
k	リテラル・フィールド、定数データまたはラベル
x	場所は気にしない (= 0または1)。 アセンブラは、x = 0のコードを生成します。 これは、すべてのMicrochipソフトウェア・ツールとの互換性のために推奨される使用形態です。
d	ディスティネーション (宛先) 選択 ; d = 0 : 結果をWに格納します。 d = 1 : 結果をファイルレジスタfに格納します。 デフォルトはd = 1です。
n	FSRまたはINDF番号 (0 ~ 1)
mm	前後のインクリメント・デクリメントモードの選択

表28-2: 略語の説明

場	説明
PC	プログラム・カウンタ
$\overline{TO}$	タイムアウト・ビット
C	キャリー・ビット
DC	デジット・キャリー・ビット
Z	ゼロ・ビット
$\overline{PD}$	パワーダウン・ビット

図28-1: 指示の一般的な形式



# PIC16(L)F1508/9

表28-3: 拡張されたミッドレンジ命令セット

ニモニック、 オペランド	説明	サイク ル	14ビット・オペコード		影響される ステータス	注釈	
			MSb	LSb			
バイト指向のファイル・レジスター操作							
ADDWF	f, d	Add W and f	1	00 0111	dfff ffff	C, DC, Z	2
ADDWFC	f, d	Add with Carry W and f	1	11 1101	dfff ffff	C, DC, Z	2
ANDWF	f, d	AND W with f	1	00 0101	dfff ffff	Z	2
ASRF	f, d	Arithmetic Right Shift	1	11 0111	dfff ffff	C, Z	2
LSLF	f, d	Logical Left Shift	1	11 0101	dfff ffff	C, Z	2
LSRF	f, d	Logical Right Shift	1	11 0110	dfff ffff	C, Z	2
CLRF	f	Clear f	1	00 0001	1fff ffff	Z	2
CLRW	—	Clear W	1	00 0001	0000 00xx	Z	
COMF	f, d	Complement f	1	00 1001	dfff ffff	Z	2
DECF	f, d	Decrement f	1	00 0011	dfff ffff	Z	2
INCF	f, d	Increment f	1	00 1010	dfff ffff	Z	2
IORWF	f, d	Inclusive OR W with f	1	00 0100	dfff ffff	Z	2
MOVF	f, d	Move f	1	00 1000	dfff ffff	Z	2
MOVWF	f	Move W to f	1	00 0000	1fff ffff		2
RLF	f, d	Rotate Left f through Carry	1	00 1101	dfff ffff	C	2
RRF	f, d	Rotate Right f through Carry	1	00 1100	dfff ffff	C	2
SUBWF	f, d	Subtract W from f	1	00 0010	dfff ffff	C, DC, Z	2
SUBWFB	f, d	Subtract with Borrow W from f	1	11 1011	dfff ffff	C, DC, Z	2
SWAPF	f, d	Swap nibbles in f	1	00 1110	dfff ffff		2
XORWF	f, d	Exclusive OR W with f	1	00 0110	dfff ffff	Z	2
バイト指向のスキップ操作							
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00 1011	dfff ffff		1, 2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00 1111	dfff ffff		1, 2
ビット指向のファイル・レジスター操作							
BCF	f, b	Bit Clear f	1	01 00bb	bfff ffff		2
BSF	f, b	Bit Set f	1	01 01bb	bfff ffff		2
ビット指向のスキップ操作							
BTFSC	f, b	Bit Test f, Skip if Clear	1(2)	01 10bb	bfff ffff		1, 2
BTFSS	f, b	Bit Test f, Skip if Set	1(2)	01 11bb	bfff ffff		1, 2
リテラル操作							
ADDLW	k	Add literal and W	1	11 1110	kkkk kkkk	C, DC, Z	
ANDLW	k	AND literal with W	1	11 1001	kkkk kkkk	Z	
IORLW	k	Inclusive OR literal with W	1	11 1000	kkkk kkkk	Z	
MOVLB	k	Move literal to BSR	1	00 0000	001k kkkk		
MOVLP	k	Move literal to PCLATH	1	11 0001	1kkk kkkk		
MOVLW	k	Move literal to W	1	11 0000	kkkk kkkk		
SUBLW	k	Subtract W from literal	1	11 1100	kkkk kkkk	C, DC, Z	
XORLW	k	Exclusive OR literal with W	1	11 1010	kkkk kkkk	Z	

注1: プログラム・カウンタ (PC) が変更された場合、または条件付きテストが真の場合、命令には2サイクルが必要です。2番目のサイクルはNOPとして実行されます。

2: この命令がINDFレジスタをアドレス指定し、対応するFSRのMSbが設定されている場合、この命令には1つの追加の命令サイクルが必要になります。

表28-3: 拡張されたミッドレンジ命令セット(続き)

ニモニック、 オペランド	説明	サイク ル	14ビット・オペコード		影響される ステータス	注釈
			MSb	LSb		
<b>制御操作</b>						
BRA	k	Relative Branch	2	11 001k	kkkk kkkk	
BRW	—	Relative Branch with W	2	00 0000	0000 1011	
CALL	k	Call Subroutine	2	10 0kkk	kkkk kkkk	
CALLW	—	Call Subroutine with W	2	00 0000	0000 1010	
GOTO	k	Go to address	2	10 1kkk	kkkk kkkk	
RETFIE	k	Return from interrupt	2	00 0000	0000 1001	
RETLW	k	Return with literal in W	2	11 0100	kkkk kkkk	
RETURN	—	Return from Subroutine	2	00 0000	0000 1000	
<b>固有の操作</b>						
CLRWDT	—	Clear Watchdog Timer	1	00 0000	0110 0100	$\overline{TO}$ , $\overline{PD}$
NOP	—	No Operation	1	00 0000	0000 0000	
OPTION	—	Load OPTION_REG register with W	1	00 0000	0110 0010	
RESET	—	Software device Reset	1	00 0000	0000 0001	
SLEEP	—	Go into Standby mode	1	00 0000	0110 0011	$\overline{TO}$ , $\overline{PD}$
TRIS	f	Load TRIS register with W	1	00 0000	0110 0fff	
<b>Cコンパイラ最適化</b>						
ADDFSR	n, k	Add Literal k to FSRn	1	11 0001	0nkk kkkk	
MOVIW	n mm	Move Indirect FSRn to W with pre/post inc/dec modifier, mm	1	00 0000	0001 0nmm	Z 2, 3
	k[n]	Move INDFn to W, Indexed Indirect.	1	11 1111	0nkk kkkk	Z 2
MOVWI	n mm	Move W to Indirect FSRn with pre/post inc/dec modifier, mm	1	00 0000	0001 1nmm	2, 3
	k[n]	Move W to INDFn, Indexed Indirect.	1	11 1111	1nkk kkkk	2

注1: プログラム・カウンタ (PC) が変更された場合、または条件付きテストが真の場合、命令には2サイクルが必要です。2番目のサイクルはNOPとして実行されます。

- 2: この命令がINDFレジスタをアドレス指定し、対応するFSRのMSbが設定されている場合、この命令には1つの追加の命令サイクルが必要になります。
- 3: MOVIWおよびMOVWI命令の説明の表を参照してください。

# PIC16(L)F1508/9

## 28.2 命令の説明

ADDFSR	Add Literal to FSRn
構文:	[label] ADDFSR FSRn, k
オペランド:	$-32 \leq k \leq 31$ $n \in [0, 1]$
操作:	$FSR(n) + k \rightarrow FSR(n)$
影響されるステータス:	なし
説明:	符号付き6ビットリテラル「k」が FSRnH:FSRnLレジスター・ペアの内容に追加されます。  FSRnは、0000h~FFFFhの範囲に制限されています。これらの境界を超えて移動すると、FSRがラップアラウンドします。

ADDLW	Add literal and W
構文:	[label] ADDLW k
オペランド:	$0 \leq k \leq 255$
操作:	$(W) + k \rightarrow (W)$
影響されるステータス:	C、DC、Z
説明:	Wレジスターの内容が8ビットリテラル「k」に追加され、結果がWレジスターに配置されます。

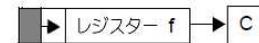
ADDWF	Add W and f
構文:	[label] ADDWF f,d
オペランド:	$0 \leq f \leq 127$ $d \in [0, 1]$
操作:	$(W) + (f) \rightarrow (\text{宛先})$
影響されるステータス:	C、DC、Z
説明:	Wレジスターの内容をレジスター「f」で追加します。「d」が「0」の場合、結果はWレジスターに格納されます。「d」が「1」の場合、結果はレジスター「f」に保存されます。

ADDWF	ADD W and CARRY bit to f
構文:	[label] ADDWFC f{,d}
オペランド:	$0 \leq f \leq 127$ $d \in [0, 1]$
操作:	$(W) + (f) + (C) \rightarrow \text{宛先}$
影響されるステータス:	C、DC、Z
説明:	W、キャリー・フラグ、およびデータ・メモリーの場所「f」を追加します。「d」が「0」の場合、結果はWに配置されます。「d」が「1」の場合、結果はデータ・メモリー位置「f」に配置されます。

ANDLW	AND literal with W
構文:	[label] ANDLW k
オペランド:	$0 \leq k \leq 255$
操作:	$(W) .AND. (k) \rightarrow (W)$
影響されるステータス:	Z
説明:	Wレジスターの内容は、8ビットリテラル「k」とAND演算されます。結果はWレジスターに配置されます。

ANDWF	AND W with f
構文:	[label] ANDWF f,d
オペランド:	$0 \leq f \leq 127$ $d \in [0, 1]$
操作:	$(W) .AND. (f) \rightarrow (\text{宛先})$
影響されるステータス:	Z
説明:	Wレジスターとレジスター「f」とのAND演算。「d」が「0」の場合、結果はWレジスターに格納されます。「d」が「1」の場合、結果はレジスター「f」に保存されます。

ASRF	Arithmetic Right Shift
構文:	[label] ASRF f{,d}
オペランド:	$0 \leq f \leq 127$ $d \in [0, 1]$
操作:	$(f < 7 >) \rightarrow \text{宛先} < 7 >$ 、 $(f < 7:1 >) \rightarrow \text{宛先} < 6:0 >$ 、 $(f < 0 >) \rightarrow C$
影響されるステータス:	C、Z
説明:	レジスター「f」の内容は、キャリー・フラグを介して1ビット右にシフトされます。MSbは変更されません。「d」が「0」の場合、結果はWに配置されます。「d」が「1」の場合、結果はレジスター「f」に格納されます。



BCF	Bit Clear f
構文:	[ <i>label</i> ] BCF f,b
オペランド:	$0 \leq f \leq 127$ $0 \leq b \leq 7$
操作:	$0 \rightarrow (f<b>)$
影響されるステータス:	なし
説明:	レジスター「f」のビット「b」がクリアされます。

BRA	Relative Branch
構文:	[ <i>label</i> ] BRA <i>label</i> [ <i>label</i> ] BRA \$+k
オペランド:	$-256 \leq \text{label} - \text{PC} + 1 \leq 255$ $-256 \leq k \leq 255$
操作:	$(\text{PC}) + 1 + k \rightarrow \text{PC}$
影響されるステータス:	なし
説明:	符号付き9ビット・リテラル「k」をPCに追加します。PCは次の命令をフェッチするためにインクリメントされるため、新しいアドレスは $\text{PC} + 1 + k$ になります。この命令は2サイクル命令です。このブランチの範囲は限られています。

BRW	Relative Branch with W
構文:	[ <i>label</i> ] BRW
オペランド:	なし
操作:	$(\text{PC}) + (W) \rightarrow \text{PC}$
影響されるステータス:	なし
説明:	W（符号なし）の内容をPCに追加します。PCは次の命令をフェッチするためにインクリメントされるため、新しいアドレスは $\text{PC} + 1 + (W)$ になります。この命令は2サイクル命令です。

BSF	Bit Set f
構文:	[ <i>label</i> ] BSF f,b
オペランド:	$0 \leq f \leq 127$ $0 \leq b \leq 7$
操作:	$1 \rightarrow (f<b>)$
影響されるステータス:	なし
説明:	レジスター「f」のビット「b」がセットされます。

BTFSC	Bit Test f, Skip if Clear
構文:	[ <i>label</i> ] BTFSC f,b
オペランド:	$0 \leq f \leq 127$ $0 \leq b \leq 7$
操作:	$(f<b>) = 0$ ならスキップ
影響されるステータス:	なし
説明:	レジスター「f」のビット「b」が「1」の場合、次の命令が実行されます。 レジスター「f」のビット「b」が「0」の場合、次の命令は破棄され、代わりに <b>NOP</b> が実行されるため、これは2サイクルの命令になります。

BTFSS	Bit Test f, Skip if Set
構文:	[ <i>label</i> ] BTFSS f,b
オペランド:	$0 \leq f \leq 127$ $0 \leq b \leq 7$
操作:	$(f<b>) = 1$ ならスキップ
影響されるステータス:	なし
説明:	レジスター「f」のビット「b」が「0」の場合、次の命令が実行されます。 ビット「b」が「1」の場合、次の命令は破棄され、代わりに <b>NOP</b> が実行されるため、これは2サイクルの命令になります。

# PIC16(L)F1508/9

CALL	Call Subroutine
構文:	[ <i>label</i> ] CALL k
オペランド:	$0 \leq k \leq 2047$
操作:	(PC)+1 → TOS、 k → PC<10:0>、 (PCLATH<6:3>) → PC<14:11>
影響されるステータス:	なし
説明:	サブルーチンを呼び出します。まず、リターンアドレス (PC + 1) がスタックにプッシュされます。11ビットの直接アドレスはPCビット<10:0>にロードされます。PCの上位ビットはPCLATHからロードされます。 <b>CALL</b> は2サイクルの命令です。

CALLW	Subroutine Call With W
構文:	[ <i>label</i> ] CALLW
オペランド:	なし
操作:	(PC)+1 → TOS、 (W) → PC<7:0>、 (PCLATH<6:0>) → PC<14:8>
影響されるステータス:	なし
説明:	Wを使用したサブルーチン呼び出し。最初に、戻りアドレス (PC + 1) がリターン・スタックにプッシュされます。次に、Wの内容がPC<7:0>にロードされ、PCLATHの内容がPC<14:8>にロードされます。 <b>CALLW</b> は2サイクルの命令です。

CLRF	Clear f
構文:	[ <i>label</i> ] CLRF f
オペランド:	$0 \leq f \leq 127$
操作:	00h → (f) 1 → Z
影響されるステータス:	Z
説明:	レジスター「f」の内容がクリアされ、Zビットがセットされます。

CLRW	Clear W
構文:	[ <i>label</i> ] CLRW
オペランド:	なし
操作:	00h → (W) 1 → Z
影響されるステータス:	Z
説明:	Wレジスターがクリアされます。ゼロ・ビット (Z) がセットされます。

CLRWDI	Clear Watchdog Timer
構文:	[ <i>label</i> ] CLRWDI
オペランド:	なし
操作:	00h → WDT、 0 → WDTプリスケイラー、 1 → $\overline{TO}$ 、 1 → $\overline{PD}$
影響されるステータス:	$\overline{TO}$ 、 $\overline{PD}$
説明:	<b>CLRWDI</b> 命令は、ウォッチドッグ・タイマーをリセットします。また、WDTのプリスケイラーをリセットします。ステータス・ビット $\overline{TO}$ と $\overline{PD}$ が設定されます。

COMF	Complement f
構文:	[ <i>label</i> ] COMF f,d
オペランド:	$0 \leq f \leq 127$ d ∈ [0, 1]
操作:	(f) → (宛先)
影響されるステータス:	Z
説明:	レジスター「f」の内容が補完されます。「d」が「0」の場合、結果はWに格納されます。「d」が「1」の場合、結果はレジスター「f」に格納されます。

DECF	Decrement f
構文:	[ <i>label</i> ] DECF f,d
オペランド:	$0 \leq f \leq 127$ d ∈ [0, 1]
操作:	(f) - 1 → (宛先)
影響されるステータス:	Z
説明:	レジスター「f」をデクリメントします。「d」が「0」の場合、結果はWレジスターに格納されます。「d」が「1」の場合、結果はレジスター「f」に保存されます。

DECFSZ	Decrement f, Skip if 0
構文:	[ <i>label</i> ] DECFSZ f,d
オペランド:	$0 \leq f \leq 127$ $d \in [0, 1]$
操作:	(f) - 1 → (宛先); 結果 = 0 ならスキップ
影響されるステータス:	なし
説明:	レジスタ「f」の内容がデクリメントされます。「d」が「0」の場合、結果はWレジスタに配置されます。「d」が「1」の場合、結果はレジスタ「f」に戻されます。 結果が「≠0」以外の場合、次の命令が実行されます。結果が「0」の場合、代わりに <b>NOOP</b> が実行され、2サイクルの命令になります。

GOTO	Unconditional Branch
構文:	[ <i>label</i> ] GOTO k
オペランド:	$0 \leq k \leq 2047$
操作:	$k \rightarrow PC<10:0>$ 、 $PCLATH<6:3> \rightarrow PC<14:11>$
影響されるステータス:	なし
説明:	<b>GOTO</b> は無条件の分岐です。11ビットの即値はPCビット<10:0>にロードされます。PCの上位ビットはPCLATH<4:3>からロードされます。 <b>GOTO</b> は2サイクルの命令です。

INCF	Increment f
構文:	[ <i>label</i> ] INCF f,d
オペランド:	$0 \leq f \leq 127$ $d \in [0, 1]$
操作:	(f) + 1 → (宛先)
影響されるステータス:	Z
説明:	レジスタ「f」の内容がインクリメントされます。「d」が「0」の場合、結果はWレジスタに配置されます。「d」が「1」の場合、結果はレジスタ「f」に戻されます。

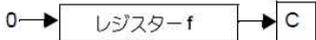
INCFSZ	Increment f, Skip if 0
構文:	[ <i>label</i> ] INCFSZ f,d
オペランド:	$0 \leq f \leq 127$ $d \in [0, 1]$
操作:	(f) + 1 → (宛先)、 結果 = 0 ならスキップ
影響されるステータス:	なし
説明:	レジスタ「f」の内容がインクリメントされます。「d」が「0」の場合、結果はWレジスタに配置されます。「d」が「1」の場合、結果はレジスタ「f」に戻されます。 結果が「≠0」以外の場合、次の命令が実行されます。結果が「0」の場合、代わりに <b>NOOP</b> が実行され、2サイクルの命令になります。

IORLW	Inclusive OR literal with W
構文:	[ <i>label</i> ] IORLW k
オペランド:	$0 \leq k \leq 255$
操作:	(W) .OR. k → (W)
影響されるステータス:	Z
説明:	Wレジスタの内容は、8ビット・リテラル「k」とOR演算されます。結果はWレジスタに配置されます。

IORWF	Inclusive OR W with f
構文:	[ <i>label</i> ] IORWF f,d
オペランド:	$0 \leq f \leq 127$ $d \in [0, 1]$
操作:	(W) .OR. (f) → (宛先)
影響されるステータス:	Z
説明:	Wレジスタとレジスタ「f」のインクルーシブ（包括的）OR。「d」が「0」の場合、結果はWレジスタに配置されます。「d」が「1」の場合、結果はレジスタ「f」に戻されます。

# PIC16(L)F1508/9

LSLF	Logical Left Shift
構文:	[ <i>label</i> ] LSLF f {,d}
オペランド:	$0 \leq f \leq 127$ $d \in [0, 1]$
操作:	(f<7>) → C (f<6:0>) → 宛先<7:1> 0 → 宛先<0>
影響されるステータス:	C、Z
説明:	レジスタ「f」の内容は、キャリー・フラグによって1ビット左にシフトされます。「0」はLSbにシフトされます。「d」が「0」の場合、結果はWに配置されます。「d」が「1」の場合、結果はレジスタ「f」に格納されます。 

LSRF	Logical Right Shift
構文:	[ <i>label</i> ] LSRF f {,d}
オペランド:	$0 \leq f \leq 127$ $d \in [0, 1]$
操作:	0 → 宛先<7>、 (f<7:1>) → 宛先<6:0>、 (f<0>) → C
影響されるステータス:	C、Z
説明:	レジスタ「f」の内容は、キャリー・フラグを介して1ビット右にシフトされます。「0」はMSbにシフトされます。「d」が「0」の場合、結果はWに配置されます。「d」が「1」の場合、結果はレジスタ「f」に格納されます。 

MOVf	Move f
構文:	[ <i>label</i> ] MOVf f,d
オペランド:	$0 \leq f \leq 127$ $d \in [0, 1]$
操作:	(f) → (宛先)
影響されるステータス:	Z
説明:	レジスタfの内容は、dのステータスに応じて宛先に移動されます。d = 0の場合、宛先はWレジスタです。d = 1の場合、宛先はファイルレジスタf自体です。d = 1は、ステータスフラグZが影響を受けるため、ファイルレジスタをテストするのに役立ちます。
ワード:	1
サイクル:	1
例:	<b>MOVf FSR,0</b>  命令の後 W = FSRレジスタの値 Z = 1

MOVIW	Move INDFn to W
構文:	[ <i>label</i> ] MOVIW ++FSRn [ <i>label</i> ] MOVIW --FSRn [ <i>label</i> ] MOVIW FSRn++ [ <i>label</i> ] MOVIW FSRn-- [ <i>label</i> ] MOVIW k[FSRn]
オペランド:	$n \in [0, 1]$ $mm \in [00, 01, 10, 11]$ $-32 \leq k \leq 31$
操作:	INDFn → W 有効なアドレスは次により決定されます: <ul style="list-style-type: none"> <li>FSR + 1 (前置インクリメント)</li> <li>FSR - 1 (前置デクリメント)</li> <li>FSR + k (相対オフセット)</li> </ul> 移動後、FSR値は次のいずれかになります: <ul style="list-style-type: none"> <li>FSR + 1 (全てのインクリメント)</li> <li>FSR - 1 (全てのデクリメント)</li> <li>変更されません</li> </ul>
影響されるステータス:	Z

モード	構文	mm
前置インクリメント	++FSRn	00
前置デクリメント	--FSRn	01
後置インクリメント	FSRn++	10
後置デクリメント	FSRn--	11

説明:	<p>この命令は、Wと間接レジスタの1つ (INDFn) の間でデータを移動するために使用されます。この移動の前後に、ポインタ (FSRn) は、インクリメント/デクリメントの前後に更新されます。</p> <p><b>注釈:</b> INDFnレジスタは物理レジスタではありません。INDFnレジスタにアクセスする命令は、実際にはFSRnで指定されたアドレスのレジスタにアクセスします。</p> <p>FSRnは、0000h ~ FFFFhの範囲に制限されています。これらの境界を超えてインクリメント/デクリメントすると、ラップアラウンドが発生します。</p>
-----	--

MOVLB	Move literal to BSR
構文:	[ <i>label</i> ] MOVLB k
オペランド:	$0 \leq k \leq 31$
操作:	$k \rightarrow \text{BSR}$
影響されるステータス:	なし
説明:	5ビット・リテラル「k」はバンク選択レジスタ (BSR) にロードされます。

MOVLP	Move literal to PCLATH
構文:	[ <i>label</i> ] MOVLP k
オペランド:	$0 \leq k \leq 127$
操作:	$k \rightarrow \text{PCLATH}$
影響されるステータス:	なし
説明:	7ビットのリテラル「k」がPCLATHレジスタにロードされます。

MOVLW	Move literal to W
構文:	[ <i>label</i> ] MOVLW k
オペランド:	$0 \leq k \leq 255$
操作:	$k \rightarrow (W)$
影響されるステータス:	なし
説明:	8ビットリ・テラル「k」がWレジスタにロードされます。「ドントケア」(気にするな) は「0」として組み立てられます。
ワード:	1
サイクル:	1
例:	<pre>MOVWLW 0x5A</pre> <p>命令の後 W = 0x5A</p>

MOVWF	Move W to f
構文:	[ <i>label</i> ] MOVWF f
オペランド:	$0 \leq f \leq 127$
操作:	$(W) \rightarrow (f)$
影響されるステータス:	なし
説明:	Wレジスタからレジスタ「f」にデータを移動します。
ワード:	1
サイクル:	1
例:	<pre>MOVWF OPTION_REG</pre> <p>命令の前 OPTION_REG = 0xFF W = 0x4F</p> <p>命令の後 OPTION_REG = 0x4F W = 0x4F</p>

# PIC16(L)F1508/9

MOVWI	Move W to INDFn
構文:	[ <i>label</i> ] MOVWI ++FSRn [ <i>label</i> ] MOVWI --FSRn [ <i>label</i> ] MOVWI FSRn++ [ <i>label</i> ] MOVWI FSRn-- [ <i>label</i> ] MOVWI k[FSRn]
オペランド:	$n \in [0, 1]$ $mm \in [00, 01, 10, 11]$ $-32 \leq k \leq 31$
操作:	W → INDFn 有効アドレスは次により決定されます: <ul style="list-style-type: none"> <li>FSR + 1 (前置インクリメント)</li> <li>FSR - 1 (前置デクリメント)</li> <li>FSR + k (相対オフセット)</li> </ul> 移動の後、FSR値はどちらかになります: <ul style="list-style-type: none"> <li>FSR + 1 (全てのインクリメント)</li> <li>FSR - 1 (全てのデクリメント)</li> </ul> 変更されません
影響されるステータス:	なし

モード	構文	mm
前置インクリメント	++FSRn	00
前置デクリメント	--FSRn	01
後置インクリメント	FSRn++	10
後置デクリメント	FSRn--	11

説明:

この命令は、Wと間接レジスタの1つ (INDFn) の間でデータを移動するために使用されます。この移動の前後に、ポインタ (FSRn) は、インクリメント/デクリメントの前後に更新されます。

**注釈:** INDFnレジスタは物理レジスタではありません。INDFnレジスタにアクセスする命令は、実際にはFSRnで指定されたアドレスのレジスタにアクセスします。

FSRnは、0000h ~ FFFFhの範囲に制限されています。これらの境界を超えてインクリメント/デクリメントすると、ラップアラウンドが発生します。

FSRnのインクリメント/デクリメント操作は、ステータス・ビットには影響しません。

NOP	No Operation
構文:	[ <i>label</i> ] NOP
オペランド:	なし
操作:	操作なし
影響されるステータス:	なし
説明:	操作なし
ワード:	1
サイクル:	1
例:	<b>NOP</b>

OPTION	Load OPTION_REG Register with W
構文:	[ <i>label</i> ] OPTION
オペランド:	なし
操作:	(W) → OPTION_REG
影響されるステータス:	なし
説明:	WレジスタからOPTION_REGレジスタにデータを移動します。

RESET	Software Reset
構文:	[ <i>label</i> ] RESET
オペランド:	なし
操作:	デバイスリセットを実行します。PCONレジスタのnRIフラグをリセットします。
影響されるステータス:	なし
説明:	この命令は、ソフトウェアによるハードウェア・リセットを実行する方法を提供します。

RETFIE	Return from Interrupt
構文:	[ label ] RETFIE
オペランド:	なし
操作:	TOS → PC、 1 → GIE
影響されるステータス:	なし
説明:	割り込みから戻る。スタックがポップされ、トップ・オブ・スタック (TOS) がPCにロードされます。割り込みは、グローバル割り込み許可ビットGIE (INTCON<7>) を設定することで有効になります。これは2サイクルの命令です。
ワード:	1
サイクル:	2
例:	<b>RETFIE</b> 割り込みの後 PC = TOS GIE = 1

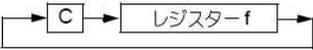
RETLW	Return with literal in W
構文:	[ label ] RETLW k
オペランド:	0 ≤ k ≤ 255
操作:	k → (W); TOS → PC
影響されるステータス:	なし
説明:	Wレジスターには、8ビットのリテラル「k」がロードされます。プログラム・カウンタは、スタックの最上位 (戻りアドレス) からロードされます。これは2サイクルの命令です。
ワード:	1
サイクル:	2
例:	<b>CALL TABLE;W contains table ;offset value ;W now has table value</b> . . . <b>ADDWF PC ;W = offset RETLW k1 ;Begin table RETLW k2 ;</b> . . . <b>RETLW kn ; End of table</b>
	命令の前 W = 0x07 命令の後 W = k8の値

RETURN	Return from Subroutine
構文:	[ label ] RETURN
オペランド:	なし
操作:	TOS → PC
影響されるステータス:	なし
説明:	サブルーチンから戻ります。スタックがポップされ、スタックの最上位 (TOS) がプログラム・カウンタにロードされます。これは2サイクルの命令です。

RLF	Rotate Left f through Carry
構文:	[ label ] RLF f,d
オペランド:	0 ≤ f ≤ 127 d ∈ [0, 1]
操作:	以下の説明を参照してください
影響されるステータス:	C
説明:	レジスター「f」の内容は、キャリー・フラグを介して1ビット左にローテーションされます。「d」が「0」の場合、結果はWレジスターに配置されます。「d」が「1」の場合、結果はレジスター「f」に保存されます。
	

ワード:	1
サイクル:	1
例:	<b>RLF REG1,0</b>  命令の前 REG1 = 1110 0110 C = 0 命令の後 REG1 = 1110 0110 W = 1100 1100 C = 1

# PIC16(L)F1508/9

RRF	Rotate Right f through Carry
構文:	[label] RRF f,d
オペランド:	$0 \leq f \leq 127$ $d \in [0, 1]$
操作:	以下の説明を参照してください
影響されるステータス:	C
説明:	レジスター「f」の内容は、キャリー・フラグを介して1ビット右にローテーションされます。「d」が「0」の場合、結果はWレジスターに配置されます。「d」が「1」の場合、結果はレジスター「f」に戻されます。 

SLEEP	Enter Sleep mode
構文:	[label] SLEEP
オペランド:	なし
操作:	00h → WDT、 0 → WDTプリスケラー、 1 → $\overline{TO}$ 、 0 → PD
影響されるステータス:	$\overline{TO}$ 、PD
説明:	パワーダウン・ステータス・ビット、PDがクリアされます。タイムアウト・ステータスビット、 $\overline{TO}$ がセットされます。ウォッチドッグ・タイマーとそのプリスケラーがクリアされます。 プロセッサは、発信器が停止した状態でスリープ・モードになります。

SUBLW	Subtract W from literal								
構文:	[label] SUBLW k								
オペランド:	$0 \leq k \leq 255$								
操作:	$k - (W) \rightarrow (W)$								
影響されるステータス:	C、DC、Z								
説明:	Wレジスターは、8ビット・リテラル「k」から減算されます（2の補数法）。結果はWレジスタに配置されます。 <table border="1" data-bbox="1054 488 1468 613"> <tbody> <tr> <td>C = 0</td> <td>W &gt; k</td> </tr> <tr> <td>C = 1</td> <td>W ≤ k</td> </tr> <tr> <td>DC = 0</td> <td>W&lt;3:0&gt; &gt; k&lt;3:0&gt;</td> </tr> <tr> <td>DC = 1</td> <td>W&lt;3:0&gt; ≤ k&lt;3:0&gt;</td> </tr> </tbody> </table>	C = 0	W > k	C = 1	W ≤ k	DC = 0	W<3:0> > k<3:0>	DC = 1	W<3:0> ≤ k<3:0>
C = 0	W > k								
C = 1	W ≤ k								
DC = 0	W<3:0> > k<3:0>								
DC = 1	W<3:0> ≤ k<3:0>								

SUBWF	Subtract W from f								
構文:	[label] SUBWF f,d								
オペランド:	$0 \leq f \leq 127$ $d \in [0, 1]$								
操作:	(f) - (W) → (宛先)								
影響されるステータス:	C、DC、Z								
説明:	レジスター「f」から（2の補数法）Wレジスターを減算します。「d」が「0」の場合、結果はWレジスターに格納されます。「d」が「1」の場合、結果はレジスター「f」に保存されます。 <table border="1" data-bbox="1054 1115 1468 1240"> <tbody> <tr> <td>C = 0</td> <td>W &gt; f</td> </tr> <tr> <td>C = 1</td> <td>W ≤ f</td> </tr> <tr> <td>DC = 0</td> <td>W&lt;3:0&gt; &gt; f&lt;3:0&gt;</td> </tr> <tr> <td>DC = 1</td> <td>W&lt;3:0&gt; ≤ f&lt;3:0&gt;</td> </tr> </tbody> </table>	C = 0	W > f	C = 1	W ≤ f	DC = 0	W<3:0> > f<3:0>	DC = 1	W<3:0> ≤ f<3:0>
C = 0	W > f								
C = 1	W ≤ f								
DC = 0	W<3:0> > f<3:0>								
DC = 1	W<3:0> ≤ f<3:0>								

SUBWFB	Subtract W from f with Borrow
構文:	SUBWFB f {,d}
オペランド:	$0 \leq f \leq 127$ $d \in [0, 1]$
操作:	(f) - (W) - ( $\overline{B}$ ) → 宛先
影響されるステータス:	C、DC、Z
説明:	レジスター「f」（2の補数法）からWとBORROWフラグ（CARRY）を減算します。「d」が「0」の場合、結果はWに格納されます。「d」が「1」の場合、結果はレジスター「f」に格納されます。

SWAPF	Swap Nibbles in f
構文:	[ <i>label</i> ] SWAPF f,d
オペランド:	$0 \leq f \leq 127$ $d \in [0, 1]$
操作:	(f<3:0>) → (宛先<7:4>), (f<7:4>) → (宛先<3:0>)
影響されるステータス:	なし
説明:	レジスタ「f」の上部ニブルと下部ニブルが交換されます。「d」が「0」の場合、結果はWレジスタに配置されます。「d」が「1」の場合、結果はレジスタ「f」に配置されます。

TRIS	Load TRIS Register with W
構文:	[ <i>label</i> ] TRIS f
オペランド:	$5 \leq f \leq 7$
操作:	(W) → TRIS register [f]
影響されるステータス:	なし
説明:	WレジスタからTRISレジスタにデータを移動します。 「f」 = 5 の場合、TRISAはロードされる。 「f」 = 6 の場合、TRISBはロードされる。 「f」 = 7 の場合、TRISCはロードされる。

XORLW	Exclusive OR literal with W
構文:	[ <i>label</i> ] XORLW k
オペランド:	$0 \leq k \leq 255$
操作:	(W) .XOR. k → (W)
影響されるステータス:	Z
説明:	Wレジスタの内容は、8ビット・リテラル「k」とXORされます。結果はWレジスタに配置されます。

XORWF	Exclusive OR W with f
構文:	[ <i>label</i> ] XORWF f,d
オペランド:	$0 \leq f \leq 127$ $d \in [0, 1]$
操作:	(W) .XOR. (f) → (宛先)
影響されるステータス:	Z
説明:	Wレジスタの内容とレジスタ「f」の排他的論理和。「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、結果はレジスタ「f」に保存されます。

# PIC16(L)F1508/9

---

注釈: