

PIC24FV32KA304 データシート

XLP テクノロジを採用した 20/28/44/48 ピン、汎用、 16 ビット フラッシュ マイクロコントローラ

マイクロチップ社製デバイスのコード保護機能に関して次の点にご注意ください。

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

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

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

商標

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

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

Analog-for-the-Digital Age,Application Maestro、chipKIT、chipKIT logo、CodeGuard、dsPICDEM、dsPICDEM.net、dsPICworks、dsSPEAK、ECAN、ECONOMONITOR、FanSense、HI-TIDE、In-Circuit Serial Programming、ICSP、Mindi、MiWi、MPASM、MPLAB Certifiedr ロゴ、MPLIB、MPLINK、mTouch、Omniscient Code Generation、PICC、PICC-18、PICDEM、PICDEM.net、PICkit、PICtail、REAL ICE、rfLAB、Select Mode、Total Endurance、TSHARC、UniWinDriver、WiperLock、ZENA は、米国およびその他の国におけるマイクロチップ・テクノロジー社の登録商標です。SQTP は、米国におけるマイクロチップ・テクノロジー社のサービスマークです。

その他、本書に記載されている商標は各社に帰属します。

© 2011, Microchip Technology Incorporated, All Rights Reserved.

ISBN: 978-1-61341-032-5

QUALITY MANAGEMENT SYSTEM CERTIFIED BY DNV ISO/TS 16949:2009

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



20/28/44/48 ピン汎用 16 ビットフラッシュ XLP テクノロジ採用マイクロコントローラ

雷源管理モード:

- 動作モード CPU/ フラッシュ /SRAM/ 周辺モジュールは ON
- Doze モード CPU クロックが周辺モジュール用クロックよりも低速で動作
- アイドルモード CPU は OFF、フラッシュ/SRAM/ 周辺モジュールは ON
- スリープモード CPU/ フラッシュ/ 周辺モジュール は OFF、SRAM は ON
- ディープスリープ モード CPU/ フラッシュ /SRAM と大部分の周辺モジュールは OFF; 複数の自律的復帰 要因
- 低消費電力モード:
 - 動作モード時の電流: 8 μA (Typical) まで低減
 - アイドルモード時の電流: 2.2 μA (Typical) まで低減
 - ディープスリープ モード時の電流:20 nA (Typical) まで低減
 - リアルタイム クロック / カレンダ電流: 700 nA まで低減 (32 kHz/1.8 V 時)
 - ウォッチドッグ タイマ電流: 500 nA (1.8 V時、typical)

高性能 CPU:

- 改良型ハーバード アーキテクチャ
- 最大 16 MIPS の動作 (32 MHz 時)
- 8 MHz 内部オシレータに 4x PLL オプションと各種分 周オプションを併用可能
- 17 ビットx17 ビット シングルサイクル ハードウェア 乗算器
- 32 ビット/16 ビットのハードウェア除算器; 16 ビット x 16 ビットのワーキング レジスタアレイ
- Cコンパイラ向けに最適化された命令セットアーキテクチャ

周辺モジュールの機能:

- ハードウェア リアルタイム クロック/カレンダ(RTCC)
 - クロック、カレンダ、アラーム機能を提供
 - ディープスリープモード中も動作可能
 - 50/60 Hz 電源ライン入力をクロック源として使用 可能
- 32 ビット プログラマブル巡回冗長検査 (CRC)
- ・ 複数のシリアル通信モジュール
- 2x3線/4線式SPIモジュール
- 2 x l²C™ モジュール (マルチマスタ / スレーブをサポート)
- 2 x UART モジュール (RS-485、RS-232、LIN/J2602、IrDA[®] をサポート)
- 5x16ビットタイマ/カウンタ (プログラマブルプリスケーラ付き)
 - ペアで32ビットタイマ/カウンタとして使用可能
- 3 x 16 ビット キャプチャ入力 (専用タイマ付き)
- 3 x 16 ビット コンペア /PWM 出力 (専用タイマ付き)
- 設定可能なオープンドレイン出力(デジタル I/O ピン)
- 最大3x外部割り込み要因

アナログ機能:

- 12 ビット アナログ / デジタル コンバータ (最大 16 チャンネル)
 - 100 ksps の変換レート
 - スリープ/アイドルモード時も変換可能
 - スリープおよびアイドルモード時の自動サンプリング タイマ対応オプション
 - 自動コンペアオプションによる復帰
- 2xレールツーレールアナログコンパレータ (入出力コンフィグレーションはプログラマブル)
- 内蔵参照電圧
- ・ 内部温度センサ
- 充電時間計測ユニット (CTMU)
 - 静電容量計測 (16 チャンネル)
 - 時間計測 (最高分解能: 200 ps)
 - 遅延/パルス生成(最高分解能:1ns)

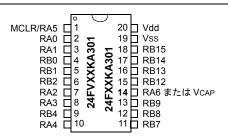
その他の特殊なマイクロコントローラ機能:

- ・ 幅広い動作電圧レンジ
 - 1.8 ~ 3.6 V (PIC24F デバイス)
 - 2.0 ~ 5.5 V (PIC24FV デバイス)
- 低消費電力復帰要因とスーパーバイザ
 - 超低消費電力復帰 (ULPWU);スリープ / ディープスリープ用
 - 低消費電力 ウォッチドッグ タイマ (DSWDT); ディープスリープ用
 - 外部低消費電カブラウンアウト リセット (DSBOR); ディープスリープ用;
 - 他の全てのモードには LPBOR を使用
- システム周波数レンジ宣言ビット
 - 周波数レンジの宣言による消費電流の最適化
- 内蔵低消費電力 RC オシレータを使用する標準ウォッチドッグ タイマ (WDT) による高信頼性動作
- プログラマブル HIGH/LOW 電圧検出 (HLVD)
- スリープ時に無効化可能な3点のプログラマブルトリップポイントを使用する標準ブラウンアウトリセット(BOR)
- 大電流シンク / ソース (18 mA/18 mA; 全 I/O ピンにて)
- ・ フラッシュ プログラムメモリ
 - 消去/書き込みサイクル:10,000回以上
 - データ保持期間:40年以上
- データ EEPROM
 - 消去/書き込みサイクル: 100,000 回以上
 - データ保持期間:40年以上
- ・ フェイルセーフ クロック監視機能
- プログラマブルな参照クロック出力
- ソフトウェア制御で自己書き込み可能
- In-Circuit Serial Programming™ (ICSP™) と 2 本のピン を介するインサーキット デバッグ (ICD)

		メモリ					/PWM				(ch)			
PIC24F デバイス	ピン数	フラッシュ プログラム (byte)	SRAM (byte)	EE データ (byte)	タイマ 16 ピット	キャプチャ 入力	カコンペア /PV 出力	UART IrDA [®] 付き	IdS	I ² Стм	12 ピット A/D (ダー つ》/ヘロ	(стми (сь)	RTCC
PIC24FV08KA101 /PIC24F16KA301	20	16K	2K	512	5	3	3	2	2	2	12	3	12	Υ
PIC24FV16KA101 /PIC24F32KA301	20	32K	2K	512	5	3	3	2	2	2	12	3	12	Υ
PIC24FV08KA102 /PIC24F16KA302	28	16K	2K	512	5	3	3	2	2	2	13	3	13	Υ
PIC24FV16KA102 /PIC24F32KA302	28	32K	2K	512	5	3	3	2	2	2	13	3	13	Υ
PIC24FV16GA004 /PIC24F16KA304	44	16K	2K	512	5	3	3	2	2	2	16	3	16	Y
PIC24FV32GA004 /PIC24F32KA304	44	32K	2K	512	5	3	3	2	2	2	16	3	16	Y

ピン配置図

20 ピン SPDIP/SSOP/SOIC(1)

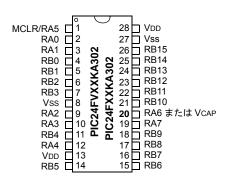


	ť٤	機能
ピン	PIC24FVXXKA301	PIC24FXXKA301
1	MCLR/Vpp/RA5	MCLR/Vpp/RA5
2	PGEC2/VREF+/CVREF+/AN0/C3INC/SCK2/CN2/RA0	PGEC2/VREF+/CVREF+/AN0/C3INC/SCK2/CN2/RA0
3	PGED2/CVREF-/VREF-/AN1/SDO2/CN3/RA1	PGED2/CVREF-/VREF-/AN1/SDO2/CN3/RA1
4	PGED1/AN2/ULPWU/CTCMP/C1IND/C2INB/C3IND/U2TX/SDI2/ OC2/CN4/RB0	PGED1/AN2/ULPWU/CTCMP/C1IND/C2INB/C3IND/U2TX/SDI2/ OC2/CN4/RB0
5	PGEC1/AN3/C1INC/C2INA/U2RX/OC3/CTED12/CN5/RB1	PGEC1/AN3/C1INC/C2INA/U2RX/OC3/CTED12/CN5/RB1
6	AN4/SDA2/T5CK/T4CK/U1RX/CTED13/CN6/RB2	AN4/SDA2/T5CK/T4CK/U1RX/CTED13/CN6/RB2
7	OSCI/AN13/C1INB/C2IND/CLKI/CN30/RA2	OSCI/AN13/C1INB/C2IND/CLKI/CN30/RA2
8	OSCO/AN14/C1INA/C2INC/CLKO/CN29/RA3	OSCO/AN14/C1INA/C2INC/CLKO/CN29/RA3
9	PGED3/SOSCI/AN15/U2RTS/CN1/RB4	PGED3/SOSCI/AN15/U2RTS/CN1/RB4
10	PGEC3/SOSCO/SCLKI/U2CTS/CN0/RA4	PGEC3/SOSCO/SCLKI/U2CTS/CN0/RA4
11	U1TX/C2OUT/OC1/IC1/CTED1/INT0/CN23/RB7	U1TX/INT0/CN23/RB7
12	SCL1/U1CTS/C3OUT/CTED10/CN22/RB8	SCL1/U1CTS/C3OUT/CTED10/CN22/RB8
13	SDA1/T1CK/U1RTS/IC2/CTED4/CN21/RB9	SDA1/T1CK/U1RTS/IC2/CTED4/CN21/RB9
14	VCAP	C2OUT/OC1/IC1/CTED1/INT2/CN8/RA6
15	AN12/LVDIN/SCK1/SS2/IC3/CTED2/INT2/CN14/RB12	AN12/LVDIN/SCK1/SS2/IC3/CTED2/CN14/RB12
16	AN11/SDO1/OCFB/CTPLS/CN13/RB13	AN11/SDO1/OCFB/CTPLS/CN13/RB13
17	CVREF/AN10/C3INB/RTCC/SDI1/C1OUT/OCFA/CTED5/INT1/ CN12/RB14	CVREF/AN10/C3INB/RTCC/SDI1/C1OUT/OCFA/CTED5/INT1/ CN12/RB14
18	AN9/C3INA/SCL2/T3CK/T2CK/REFO/SS1/CTED6/CN11/RB15	AN9/C3INA/SCL2/T3CK/T2CK/REFO/SS1/CTED6/CN11/RB15
19	Vss/AVss	Vss/AVss
20	VDD/AVDD	VDD/AVDD

Note 1: 太字のピン番号は、PIC24FV と PIC24F でピン機能が異なる事を意味します。 **Note 1:** PIC24FJ64GA ファミリのピンの最大電圧は 3.6 V であり、5 V を許容しません。

ピン配置図

28 ピン SPDIP/SSOP/SOIC(1,2)



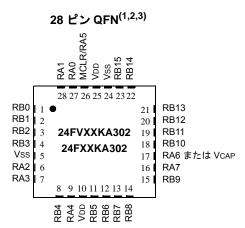
	ピン	幾能
ピン	PIC24FVXXKA302	PIC24FXXKA302
1	MCLR/Vpp/RA5	MCLR/Vpp/RA5
2	VREF+/CVREF+/AN0/C3INC/CTED1/CN2/RA0	VREF+/CVREF+/AN0/C3INC/CTED1/CN2/RA0
3	CVREF-/VREF-/AN1/CN3/RA1	CVREF-/VREF-/AN1/CN3/RA1
4	PGED1/AN2/ULPWU/CTCMP/C1IND/C2INB/C3IND/U2TX/CN4/RB0	PGED1/AN2/ULPWU/CTCMP/C1IND/C2INB/C3IND/U2TX/CN4/RB0
5	PGEC1/AN3/C1INC/C2INA/U2RX/CTED12/CN5/RB1	PGEC1/AN3/C1INC/C2INA/U2RX/CTED12/CN5/RB1
6	AN4/C1INB/C2IND/SDA2/T5CK/T4CK/U1RX/CTED13/CN6/RB2	AN4/C1INB/C2IND/SDA2/T5CK/T4CK/U1RX/CTED13/CN6/RB2
7	AN5/C1INA/C2INC/SCL2/CN7/RB3	AN5/C1INA/C2INC/SCL2/CN7/RB3
8	Vss	Vss
9	OSCI/AN13/CLKI/CN30/RA2	OSCI/AN13/CLKI/CN30/RA2
10	OSCO/AN14/CLKO/CN29/RA3	OSCO/AN14/CLKO/CN29/RA3
11	SOSCI/AN15/U2RTS/CN1/RB4	SOSCI/AN15/U2RTS/CN1/RB4
12	SOSCO/SCLKI/U2CTS/CN0/RA4	SOSCO/SCLKI/U2CTS/CN0/RA4
13	VDD	VDD
14	PGED3/ASDA ⁽¹⁾ /SCK2/CN27/RB5	PGED3/ASDA ⁽¹⁾ /SCK2/CN27/RB5
15	PGEC3/ASCL ⁽¹⁾ /SDO2/CN24/RB6	PGEC3/ASCL ⁽¹⁾ /SDO2/CN24/RB6
16	U1TX/C2OUT/OC1/INT0/CN23/RB7	U1TX/INT0/CN23/RB7
17	SCL1/U1CTS/C3OUT/CTED10/CN22/RB8	SCL1/U1CTS/C3OUT/CTED10/CN22/RB8
18	SDA1/T1CK/U1RTS/IC2/CTED4/CN21/RB9	SDA1/T1CK/U1RTS/IC2/CTED4/CN21/RB9
19	SDI2/IC1/CTED3/CN9/RA7	SDI2/IC1/CTED3/CN9/RA7
20	VCAP	C2OUT/OC1/CTED1/INT2/CN8/RA6
21	PGED2/SDI1/OC3/CTED11/CN16/RB10	PGED2/SDI1/OC3/CTED11/CN16/RB10
22	PGEC2/SCK1/OC2/CTED9/CN15/RB11	PGEC2/SCK1/OC2/CTED9/CN15/RB11
23	AN12/LVDIN/SS2/IC3/CTED2/INT2/CN14/RB12	AN12/LVDIN/SS2/IC3/CTED2/CN14/RB12
24	AN11/SDO1/OCFB/CTPLS/CN13/RB13	AN11/SDO1/OCFB/CTPLS/CN13/RB13
25	CVREF/AN10/C3INB/RTCC/C1OUT/OCFA/CTED5/INT1/CN12/RB14	CVREF/AN10/C3INB/RTCC/C1OUT/OCFA/CTED5/INT1/CN12/ RB14
26	AN9/C3INA/T3CK/T2CK/REFO/SS1/CTED6/CN11/RB15	AN9/C3INA/T3CK/T2CK/REFO/SS1/CTED6/CN11/RB15
27	Vss/AVss	Vss/AVss
28	VDD/AVDD	VDD/AVDD

凡例: 太字のピン番号は、PIC24FV と PIC24F でピン機能が異なる事を意味します。

Note 1: I2CSEL コンフィグレーション ビットは、SDA1/ASDA1 と SCL1 /ASCL1 のピンペアを選択します。

2: PIC24FJ64GA ファミリのピンの最大電圧は 3.6 V であり、5 V を許容しません。

ピン配置図



	ピン機能										
ピン	PIC24FVXXKA302	PIC24FXXKA302									
1	PGED1/AN2/ULPWU/CTCMP/C1IND/C2INB/C3IND/U2TX/CN4/RB0	PGED1/AN2/ULPWU/CTCMP/C1IND/C2INB/C3IND/U2TX/CN4/RB0									
2	PGEC1/AN3/C1INC/C2INA/U2RX/CTED12/CN5/RB1	PGEC1/AN3/C1INC/C2INA/U2RX/CTED12/CN5/RB1									
3	AN4/C1INB/C2IND/SDA2/T5CK/T4CK/U1RX/CTED13/CN6/RB2	AN4/C1INB/C2IND/SDA2/T5CK/T4CK/U1RX/CTED13/CN6/RB2									
4	AN5/C1INA/C2INC/SCL2/CN7/RB3	AN5/C1INA/C2INC/SCL2/CN7/RB3									
5	Vss	Vss									
6	OSCI/AN13/CLKI/CN30/RA2	OSCI/AN13/CLKI/CN30/RA2									
7	OSCO/AN14/CLKO/CN29/RA3	OSCO/AN14/CLKO/CN29/RA3									
8	SOSCI/AN15/U2RTS/CN1/RB4	SOSCI/AN15/U2RTS/CN1/RB4									
9	SOSCO/SCLKI/U2CTS/CN0/RA4	SOSCO/SCLKI/U2CTS/CN0/RA4									
10	VDD	VDD									
11	PGED3/ASDA1 ⁽²⁾ /SCK2/CN27/RB5	PGED3/ASDA1 ⁽²⁾ /SCK2/CN27/RB5									
12	PGEC3/ASCL1 ⁽²⁾ /SDO2/CN24/RB6	PGEC3/ASCL1 ⁽²⁾ /SDO2/CN24/RB6									
13	U1TX/C2OUT/OC1/INT0/CN23/RB7	U1TX/INT0/CN23/RB7									
14	SCL1/U1CTS/C3OUT/CTED10/CN22/RB8	SCL1/U1CTS/C3OUT/CTED10/CN22/RB8									
15	SDA1/T1CK/U1RTS/IC2/CTED4/CN21/RB9	SDA1/T1CK/U1RTS/IC2/CTED4/CN21/RB9									
16	SDI2/IC1/CTED3/CN9/RA7	SDI2/IC1/CTED3/CN9/RA7									
17	VCAP	C2OUT/OC1/CTED1/INT2/CN8/RA6									
18	PGED2/SDI1/OC3/CTED11/CN16/RB10	PGED2/SDI1/OC3/CTED11/CN16/RB10									
19	PGEC2/SCK1/OC2/CTED9/CN15/RB11	PGEC2/SCK1/OC2/CTED9/CN15/RB11									
20	AN12/LVDIN/SS2/IC3/CTED2/INT2/CN14/RB12	AN12/LVDIN/SS2/IC3/CTED2/CN14/RB12									
21	AN11/SDO1/OCFB/CTPLS/CN13/RB13	AN11/SDO1/OCFB/CTPLS/CN13/RB13									
22	CVREF/AN10/C3INB/RTCC/C1OUT/OCFA/CTED5/INT1/CN12/ RB14	CVREF/AN10/C3INB/RTCC/C1OUT/OCFA/CTED5/INT1/CN12/ RB14									
23	AN9/C3INA/T3CK/T2CK/REFO/SS1/CTED6/CN11/RB15	AN9/C3INA/T3CK/T2CK/REFO/SS1/CTED6/CN11/RB15									
24	Vss/AVss	Vss/AVss									
25	VDD/AVDD	VDD/AVDD									
26	MCLR/Vpp/RA5	MCLR/Vpp/RA5									
27	VREF+/CVREF+/AN0/C3INC/CTED1/CN2/RA0	VREF+/CVREF+/AN0/C3INC/CTED1/CN2/RA0									
28	CVRFF-/VRFF-/AN1/CN3/RA1	CVREE-/VREE-/AN1/CN3/RA1									

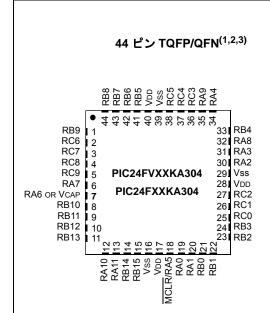
凡例: 太字のピン番号は、PIC24FVと PIC24Fでピン機能が異なる事を意味します。

Note 1: デバイス下面の露出パッドは Vss に接続されています。

2: I2CSEL コンフィグレーション ビットは、SDA1/ASDA1 と SCL1 /ASCL1 のピンペアを選択します。

3: PIC24FJ64GA ファミリのピンの最大電圧は 3.6 V であり、5 V を許容しません。

ピン配置図



	195	146 Div
ピン	PIC24FVXXKA304	ノ機能 PIC24FXXKA304
1	SDA1/T1CK/U1RTS/CTED4/CN21/ RB9	SDA1/T1CK/U1RTS/CTED4/CN21/ RB9
2	U1RX/CN18/RC6	U1RX/CN18/RC6
3	U1TX/CN17/RC7	U1TX/CN17/RC7
4	OC2/CN20/RC8	OC2/CN20/RC8
5	IC2/CTED7/CN19/RC9	IC2/CTED7/CN19/RC9
6	IC1/CTED3/CN9/RA7	IC1/CTED3/CN9/RA7
7	VCAP	C2OUT/OC1/CTED1/INT2/CN8/RA6
8	PGED2/SDI1/CTED11/CN16/RB10	PGED2/SDI1/CTED11/CN16/RB10
9	PGEC2/SCK1/CTED9/CN15/RB11	PGEC2/SCK1/CTED9/CN15/RB11
10	AN12/LVDIN/CTED2/INT2/CN14/ RB12	AN12/LVDIN/CTED2/CN14/RB12
11	AN11/SDO1/CTPLS/CN13/RB13	AN11/SDO1/CTPLS/CN13/RB13
12	OC3/CN35/RA10	OC3/CN35/RA10
13	IC3/CTED8/CN36/RA11	IC3/CTED8/CN36/RA11
14	CVREF/AN10/C3INB/RTCC/ C1OUT/OCFA/CTED5/INT1/CN12/ RB14	CVREF/AN10/C3INB/RTCC/ C1OUT/OCFA/CTED5/INT1/CN12/ RB14
15	AN9/C3INA/T3CK/T2CK/REFO/ SS1/CTED6/CN11/RB15	AN9/C3INA/T3CK/T2CK/REFO/ SS1/CTED6/CN11/RB15
16	Vss/AVss	Vss/AVss
17	VDD/AVDD	VDD/AVDD
18	MCLR/Vpp/RA5	MCLR/VPP/RA5
19	VREF+/CVREF+/AN0/C3INC/ CTED1/CN2/RA0	VREF+/CVREF+/AN0/C3INC/CN2/ RA0
20	CVREF-/VREF-/AN1/CN3/RA1	CVREF-/VREF-/AN1/CN3/RA1
21	PGED1/AN2/ULPWU/CTCMP/ C1IND/C2INB/C3IND/U2TX/CN4/RB0	PGED1/AN2/ULPWU/CTCMP/C1IND/ C2INB/C3IND/U2TX/CN4/RB0
22	PGEC1/AN3/C1INC/C2INA/U2RX/ CTED12/CN5/RB1	PGEC1/AN3/C1INC/C2INA/U2RX/ CTED12/CN5/RB1
23	AN4/C1INB/C2IND/SDA2/T5CK/ T4CK/CTED13/CN6/RB2	AN4/C1INB/C2IND/SDA2/T5CK/ T4CK/CTED13/CN6/RB2
24	AN5/C1INA/C2INC/SCL2/CN7/ RB3	AN5/C1INA/C2INC/SCL2/CN7/RB3
25	AN6/CN32/RC0	AN6/CN32/RC0
26	AN7/CN31/RC1	AN7/CN31/RC1
27	AN8/CN10/RC2	AN8/CN10/RC2
28	VDD	VDD
29	Vss	Vss
30	OSCI/AN13/CLKI/CN30/RA2	OSCI/AN13/CLKI/CN30/RA2
31	OSCO/AN14/CLKO/CN29/RA3	OSCO/AN14/CLKO/CN29/RA3
32	OCFB/CN33/RA8	OCFB/CN33/RA8
33	SOSCI/AN15/U2RTS/CN1/RB4	SOSCI/AN15/U2RTS/CN1/RB4
34	SOSCO/SCLKI/U2CTS/CN0/RA4	SOSCO/SCLKI/U2CTS/CN0/RA4
35	SS2/CN34/RA9	SS2/CN34/RA9
36	SDI2/CN28/RC3	SDI2/CN28/RC3
37	SDO2/CN25/RC4	SDO2/CN25/RC4
38	SCK2/CN26/RC5	SCK2/CN26/RC5
39	Vss	Vss
40	VDD PGED3/ASDA1 ⁽²⁾ /CN27/RB5	VDD PGED3/ASDA1 ⁽²⁾ /CN27/RB5
41 42	PGEC3/ASDA1(-*/CN27/RB5) PGEC3/ASCL1(2)/CN24/RB6	PGEC3/ASDA1*-/CN2//RB5
43	INT0/CN23/RB7	INTO/CN23/RB7
44	SCL1/U1CTS/C3OUT/CTED10/ CN22/RB8	SCL1/U1CTS/C3OUT/CTED10/ CN22/RB8

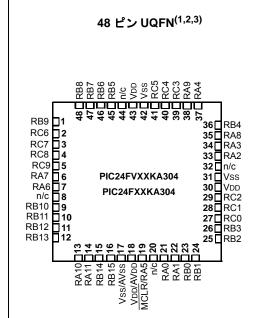
凡例: 太字のピン番号は、PIC24FV と PIC24F でピン機能が異なる事を意味します。

Note 1: デバイス下面の露出パッドはVssに接続されています。

2: I2CSEL コンフィグレーション ビット は、SDA1/ASDA1 と SCL1 /ASCL1 のピ ンペアを選択します。

3: PIC24FJ64GA ファミリのピンの最大電 圧は 3.6 V であり、5 V を許容しません。

ピン配置図



	ピン様							
ピン	PIC24FVXXKA304	PIC24FXXKA304						
1	SDA1/T1CK/U1RTS/CTED4/CN21/RB9	SDA1/T1CK/U1RTS/CTED4/CN21 RB9						
2	U1RX/CN18/RC6	U1RX/CN18/RC6						
3	U1TX/CN17/RC7	U1TX/CN17/RC7						
4	OC2/CN20/RC8	OC2/CN20/RC8						
5	IC2/CTED7/CN19/RC9	IC2/CTED7/CN19/RC9						
6	IC1/CTED3/CN9/RA7	IC1/CTED3/CN9/RA7						
7	VCAP	INT2/RA6						
8	n/c	n/c						
9	PGED2/SDI1/CTED11/CN16/RB10	PGED2/SDI1/CTED11/CN16/RB10						
10	PGEC2/SCK1/CTED9/CN15/RB11	PGEC2/SCK1/CTED9/CN15/RB11						
11	AN12/LVDIN/CTED2/INT2/CN14/RB12	AN12/LVDIN/CTED2/CN14/RB12						
12	AN11/SDO1/CTPLS/CN13/RB13	AN11/SDO1/CTPLS/CN13/RB13						
13	OC3/CN35/RA10	OC3/CN35/RA10						
14	IC3/CTED8/CN36/RA11	IC3/CTED8/CN36/RA11						
15	CVREF/AN10/C3INB/RTCC/	CVREF/AN10/C3INB/RTCC/C1OUT						
16	C1OUT/OCFA/CTED5/INT1/CN12/RB14	OCFA/CTED5/INT1/CN12/RB14						
16	AN9/C3INA/T3CK/T2CK/REFO/ SS1/CTED6/CN11/RB15	AN9/C3INA/T3CK/T2CK/REFO/ SS1/CTED6/CN11/RB15						
17	Vss/AVss	Vss/AVss						
18	VDD/AVDD	VDD/AVDD						
19	MCLR/RA5	MCLR/RA5						
20	n/c	n/c						
21	VREF+/CVREF+/AN0/C3INC/ CTED1/CN2/RA0	VREF+/CVREF+/AN0/C3INC/ CTED1/CN2/RA0						
22	CVREF-/VREF-/AN1/CN3/RA1	CVREF-/VREF-/AN1/CN3/RA1						
23	PGED1/AN2/ULPWU/CTCMP/C1IND/ C2INB/C3IND/U2TX/CN4/RB0	PGED1/AN2/ULPWU/CTCMP/C1IN C2INB/C3IND/U2TX/CN4/RB0						
24	PGEC1/AN3/C1INC/C2INA/U2RX/ CTED12/CN5/RB1	PGEC1/AN3/C1INC/C2INA/U2RX CTED12/CN5/RB1						
25	AN4/C1INB/C2IND/SDA2/T5CK/ T4CK/CTED13/CN6/RB2	AN4/C1INB/C2IND/SDA2/T5CK/ T4CK/CTED13/CN6/RB2						
26	AN5/C1INA/C2INC/SCL2/CN7/RB3	AN5/C1INA/C2INC/SCL2/CN7/RB						
27	AN6/CN32/RC0	AN6/CN32/RC0						
28	AN7/CN31/RC1	AN7/CN31/RC1						
29	AN8/CN10/RC2	AN8/CN10/RC2						
30	VDD	VDD						
31	Vss	Vss						
32	n/c	n/c						
33	OSCI/AN13/CLKI/CN30/RA2	OSCI/AN13/CLKI/CN30/RA2						
34	OSCO/AN14/CLKO/CN29/RA3	OSCO/AN14/CLKO/CN29/RA3						
35	OCFB/CN33/RA8	OCFB/CN33/RA8						
36	SOSCI/AN15/U2RTS/CN1/RB4	SOSCI/AN15/U2RTS/CN1/RB4						
37	SOSCO/SCLKI/U2CTS/CN0/RA4	SOSCO/SCLKI/U2CTS/CN0/RA4						
38	SS2/CN34/RA9	SS2/CN34/RA9						
39	SDI2/CN28/RC3	SDI2/CN28/RC3						
40	SDO2/CN25/RC4	SDO2/CN25/RC4						
41	SCK2/CN26/RC5	SCK2/CN26/RC5						
42	Vss	Vss						
43	VDD	VDD						
44	n/c	n/c						
45	PGED3/ASDA1 ⁽²⁾ /CN27/RB5	PGED3/ASDA1 ⁽²⁾ /CN27/RB5						
46	PGEC3/ASCL1 ⁽²⁾ /CN24/RB6	PGEC3/ASCL1 ⁽²⁾ /CN24/RB6						
47	C2OUT/OC1/INT0/CN23/RB7	C2OUT/OC1/INT0/CN23/RB7						
48	SCL1/U1CTS/C3OUT/CTED10/ CN22/RB8	SCL1/U1CTS/C3OUT/CTED10/ CN22/RB8						

凡例: 太字のピン番号は、PIC24FV と PIC24F でピン機能が異なる事を意味します。

Note 1: デバイス下面の露出パッドは Vss に接続されています。

2: I2CSEL コンフィグレーション ビットは、SDA1/ASDA1 と SCL1 /ASCL1 のピンペアを選択します。

3: PIC24F32KA3XX のピンの最大電圧は 3.6 V であり、5 V を許容しません。

目次

1.0	デバイス概要	12
	ナハイ へ似安	
2.0 3.0	16 ビット マイクロコントローラの使用にあたってのカイトライン	
4.0	メモリの構成	
5.0	フラッシュ プログラムメモリ	
6.0	データ EEPROM メモリ	
7.0	リセット	
8.0	割り込みコントローラ	
9.0	オシレータ コンフィグレーション	
10.0		
	1/0 ポート	
	Timer1	
13.0	Timer2/3 と Timer4/5	145
	専用タイマを備えた入力キャプチャ	
	専用タイマを備えた出力コンペア	
	シリアル ペリフェラル インターフェイス (SPI)	
17.0	I ² C™ (Inter-Integrated Circuit™)	173
18.0	UART (Universal Asynchronous Receiver Transmitter)	181
19.0	リアルタイム クロック / カレンダ (RTCC)	189
20.0	32 ビット プログラマブル巡回冗長検査 (CRC) ジェネレータ	203
21.0	HIGH/LOW 電圧検出 (HLVD)	209
22.0	しきい値検出機能を備えた 12 ビット A/D コンバータ	211
23.0	コンパレータ モジュール	225
	コンパレータ参照電圧	
	充電時間計測ユニット (CTMU)	
26.0	その他の特殊機能	239
27.0	開発サポート	251
28.0	命令セットの要約	255
	電気的特性	
	パッケージ情報	
補遺		
	クロチップ社ウェブサイト	
	変更通知サービス	
	サポート	
	アンケート	
制口	悪別シュフ テ / .	215

大切なお客様へ

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

本書に関してご質問、ご意見をお持ちのお客様は、メール (docerrors@microchip.com) にて、あるいは本データシート巻末の 「読者アンケート」にご記入の上ファックス ((480)792-4150)にてお知らせください。お客様からのご感想をお待ちしております。

最新のデータシート

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

http://www.microchip.com

データシートのバージョンは、各ページの欄外下隅に記載されている文書番号で確認できます。文書番号の末尾文字がバージョ ン番号を表します (例: DS30000A_JP であれば文書 DS30000_JP のバージョン A)。

エラッタ

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

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

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

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

顧客変更通知システム

弊社ウェブサイト (www.microchip.com) でご登録頂くと、弊社の全製品に関する最新情報をお受け取り頂けます。

NOTE:

1.0 デバイスの概要

本データシートには、下記の各デバイスに関する情報 を記載しています。

- PIC24FV16KA301、PIC24F16KA301
- PIC24FV16KA302、PIC24F16KA302
- PIC24FV16KA304、PIC24F16KA304
- PIC24FV32KA301、PIC24F32KA301
- PIC24FV32KA302、PIC24F32KA302
- PIC24FV32KA304、PIC24F32KA304

マイクロチップ社の超低消費電力デバイスの新製品ラインである PIC24FV32KA304 ファミリは、豊富な周辺モジュール機能セットと拡張された計算性能を備えた 16 ビット マイクロコントローラ ファミリです。このファミリは、8 ビット プラットフォームからのアップグレードは必要であるものの、デジタルシグナルコントローラの数値処理能力を要求しない、比較的高性能なアプリケーションの移行用に、新たな選択肢を提供します。

1.1 コアの特長

1.1.1 16 ビット アーキテクチャ

全てのPIC24Fデバイスの中核を成すのが16ビット改良型ハーバードアーキテクチャです。これはマイクロチップ社の dsPIC® デジタルシグナル コントローラに初めて採用されたアーキテクチャです。PIC24F CPUコアは、下記のように各種の性能 / 機能を備えます。

- 16 ビット データパスと 24 ビット アドレスパス: データメモリ空間とプログラムメモリ空間の間で データの移動が可能
- 単一の連続したアドレッシング: 最大 12M バイトのプログラム空間と 64K バイトの データ空間
- 16 個のワーキング レジスタで構成されたワーキング レジスタアレイ:ビルトイン ソフトウェア スタックをサポート
- 17 ビット x17 ビット ハードウェア乗算器: 整数計算をサポート
- ・ 32 ビット/16 ビット除算のハードウェア サポート
- 複数のアドレッシング モードをサポートし、高級言語 (C 等) 用に最適化された命令セット
- 最大 16 MIPS の動作性能

1.1.2 省電カテクノロジ

PIC24FV32KA304 ファミリの全デバイスは、動作時の 消費電力を大幅に低減する下記の機能を備えます。

・ 動作中クロック切り換え:

動作中に、ソフトウェア制御でデバイスクロックを Timer1 または内部低消費電力RCオシレータに切り 換える事ができます。これを利用して、ソフトウェ アにユーザ独自の省電力機能を組み込む事ができま す。

・ Doze モード動作:

シリアル通信等のタイミングが重要なアプリケーションにおいて、周辺モジュールを中断なく動作させる必要がある場合、CPU クロック速度だけを選択的に低減する事により、周辺モジュール用クロックを維持しながら消費電力を低減できます。

・ 命令ベースの省電力モード:

命令により下記の3種類の省電力モードを利用できます。

- アイドルモード:周辺モジュールを動作させたま まコアをシャットダウンします。
- スリープモード:コアと、システムクロックを必要とする周辺モジュールをシャットダウンします。独自のクロックまたは外部デバイスからのクロックを使用する周辺モジュールは動作を維持します。
- ディープスリープ モード: コア、周辺モジュール (RTCC と DSWDT を除く)、フラッシュ、SRAM をシャットダウンします。

1.1.3 オシレータ オプションと機能

PIC24FV32KA304 ファミリは 5 種類のオシレータ オプションを備え、アプリケーション ハードウェアの開発に幅広い選択肢を提供します。これには下記が含まれます。

- 水晶またはセラミック振動子用の2種類の水晶振動 子モード
- 2 分周クロック出力オプションを備えた 2 種類の外 部クロックモード
- 2種類の高速内部オシレータ (FRC): 1 つは公称 8 MHz を出力し、もう 1 つは公称 500 kHz を出力します。これらの出力をソフトウェア制御で分周する事により、それぞれ 31 kHz または 2 kHz までクロック周波数を低減できます。
- 位相ロックループ (PLL) 周波数マルチプライヤを外部オシレータモードおよび 8 MHz FRC オシレータに併用する事により、クロック速度を最大 32 MHzまで増速できます。
- 処理速度を重視しないアプリケーション向けの低消 費電力オプションとして、出力が公称 31 kHz の内 部 RC オシレータ (LPRC) を使用できます。

内部オシレータ ブロックは、フェイルセーフ クロック監視機能 (FSCM) にも安定した参照ソースを供給します。FSCM は、メインクロック源を内部オシレータからの参照信号と対比する事によってクロックを常時監視します。メインクロック源に障害が生じた場合、コントローラはクロック源を内部オシレータに切り換えて低速動作を継続するか、あるいは安全にアプリケーションをシャットダウンできます。

1.1.4 容易なデバイス移行

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

ピン配置もファミリ全体で一貫しているため、同一ピン数デバイス間だけでなく、例えば 20/28 ピンデバイスから 44/48 ピンデバイスへも容易に移行できます。

PIC24FファミリはdsPIC33ファミリとピン互換です。 さらに、PIC18 および dsPIC30 とも一部互換性を持ち ます。このため、比較的単純なものから高性能で複雑 なものまで、アプリケーションの能力を幅広く拡張で きます。

1.2 その他の特殊機能

・ 通信:

PIC24FV32KA304 ファミリは、幅広いアプリケーションの要求に応えるために、各種のシリアル通信周辺モジュールを実装しています。これには、マスタモードとスレーブモードの両方をサポートする I^2C^{TM} モジュール、 I^2DA^{BM} エンコーダ / デコーダを内蔵した UART モジュール、SPI モジュールが含まれます。

- リアル タイム クロック/カレンダ:
 このモジュールは、機能を完備したアラーム付きのクロック/カレンダをハードウェアで実装し、コアアプリケーション用のタイマ リソースとプログラムメモリ空間を消費しません。
- 12 ビット A/D コンバータ:
 プログラマブルなアクイジション時間を備えたこのモジュールは、サンプリング周期を待たずにチャンネルを選択して変換を開始でき、より高速な計測が可能です。16 段の結果バッファを使用する事により、スリープ時の消費電力を低減するか、あるいは動作モード時のスループットを改善できます。
- 充電時間計測ユニット (CTMU) インターフェイス: PIC24FV32KA304 ファミリは、静電容量式タッチセンシングおよび近接センシングと、精密な時間計測およびパルス生成に使用できる新しい CTMU インターフェイス モジュールを備えます。

1.3 ファミリ各製品の詳細

PIC24FV32KA304 ファミリは、20/28/44/48 ピンパッケージのデバイスを提供します。図 1-1 に、全デバイスに共通の概略ブロック図を示します。

デバイスによって下記の4項目が異なります。

- フラッシュ プログラムメモリ: PIC24FV16KA は 16K バイト、PIC24FV32KA は 32K バイトを実装
- I/O ピンとポートの数:
 20 ピンデバイスでは 18 ピン /2 ポート、28 ピンデバイスでは 22 ピン /2 ポート、44/48 ピンデバイスでは 38 ピン/3 ポート
- 代替 SCL および SDA ピン:
 20 ピンデバイスを除く 28/44/48 ピンデバイスの みで利用可能
- 4. PIC24FV32KA301 ファミリには、標準デバイスと高耐圧デバイスが存在します。高耐圧デバイスの製品番号には「FV」が付きます(例:PIC24FV32KA304)。これらのデバイスはコア電源用の電圧レギュレータを内蔵し、2.0~5.5 Vの VDD レンジに対応します。周辺モジュールは VDD で動作します。標準デバイスの製品番号には「F」が付きます(例:PIC24F32KA304)。これらのデバイスは1.8~3.6 Vの VDD レンジで動作します。標準デバイスは電圧レギュレータを内蔵しないため、コアと周辺モジュールは直接 VDD からの電圧で動作します。

その他の全ての機能は本ファミリ内のデバイスで共通です。それらを表 1-1 に要約します。

PIC24FV32KA304ファミリデバイスで利用可能なピン機能を名称のアルファベット順に並べた一覧を表 1-3 に記載します。

Note:

表 1-3 は、個々の周辺機能に割り当てられているピン番号を示しますが、同一ピンに多重化されている他の機能については、p.5、p.6、p.7、p.8、p.9 のピン配置図を参照してください。これらの図には、多重化された複数の機能を優先度順に左から列挙しています(左端が最優先機能)。

表 1-1: PIC24FV32KA304 ファミリのデバイス機能

表 1-1: PIC24FV32KA304 フラ	アミリのナハイ	へ (成形	1		ı	1					
機能	PIC24FV16KA301	PIC24FV32KA301	PIC24FV16KA302	PIC24FV32KA302	PIC24FV16KA304	PIC24FV32KA304					
動作周波数			DC ~ 32	MHz							
プログラムメモリ (byte)	16K	32K	16K	32K	16K	32K					
プログラムメモリ (命令数)	5632	11264	5632	11264	5632	11264					
データメモリ (byte)			2048								
データ EEPROM (byte)			512								
割り込み要因 (ソフトベクタ/NMIトラップ)			30 (26/	4)							
I/O ポート	PORTA PORTB<15:1		PORTA PORTB	,	PORTA<11:7,5:0> PORTB<15:0> PORTC<9:0>						
I/O ピンの総数	17	7	2:	3	3	8					
タイマ:総数 (16 ビット)	5										
32 ビット (16 ビットタイマのペアで構成)			2								
入力キャプチャ チャンネル			3								
出力コンペア /PWM チャンネル			3								
入力状態変化通知割り込み	16	6	2:	2	37						
シリアル通信 : UART SPI (3 線 /4 線)	2										
I ² C™			2								
12 ビット アナログ / デジタル モジュール (入力チャンネル数)	12	2	1:	3	16						
アナログ コンパレータ			3								
リセット (および遅延)	POR、BOR、RESET 命令、MCLR、WDT、不正オペコード、REPEAT 命令、ハードウェア トラップ、コンフィグレーション ワード不一致(PWRT、OST、PLL ロック)										
命令セット	76 個	の基本命令、神			-ド オプショ	ン					
パッケージ	20 E PDIP/SSC	<u>:</u> ン	28 t SPDIP/SSOF	ピン	44 ピン QFN/TQFP 48 ピン UQFN						

表 1-2: PIC24F32KA304 ファミリのデバイス機能

衣 1-2: PIC24F32NA3U4 ファ		* 134 11G	1		1	1					
機能	PIC24F16KA301	PIC24F32KA301	PIC24F16KA302	PIC24F32KA302	PIC16F16KA304	PIC24F32KA304					
動作周波数			DC ~ 32	MHz							
プログラムメモリ (byte)	16K	32K	16K	32K	16K	32K					
プログラムメモリ(命令数)	5632	11264	5632	11264	5632	11264					
データメモリ (byte)			2048								
データ EEPROM (byte)			512								
割り込み要因 (ソフトベクタ/NMIトラップ)			30 (26/	4)							
I/O ポート	PORTA PORTB<15:12		PORTA PORTB		PORTA<11:0> PORTB<15:0> PORTC<9:0>						
I/O ピンの総数	18	3	24	4	3	9					
タイマ:総数 (16 ビット)	5										
32 ビット (16 ビットタイマのペアで構成)	2										
入力キャプチャ チャンネル	3										
出力コンペア /PWM チャンネル			3								
入力状態変化通知割り込み	17	7	2:	3	38						
シリアル通信 : UART SPI (3 線 /4 線)			2								
I ² C™			2								
12 ビット アナログ / デジタル モ ジュール (入力チャンネル数)	12	2	1:	3	1	6					
アナログ コンパレータ			3								
リセット (および遅延)	POR、BOR、RESET 命令、MCLR、WDT、不正オペコード、REPEAT 命令、ハードウェア トラップ、コンフィグレーション ワード不-(PWRT、OST、PLL ロック)										
命令セット	76 個 (の基本命令、袖	复数のアドレ [・]	ッシング モー	-ドオプショ	ン					
パッケージ	20 E PDIP/SSC		28 t SPDIP/SSOF		44 ピン QFN/TQFP 48 ピン UQFN						

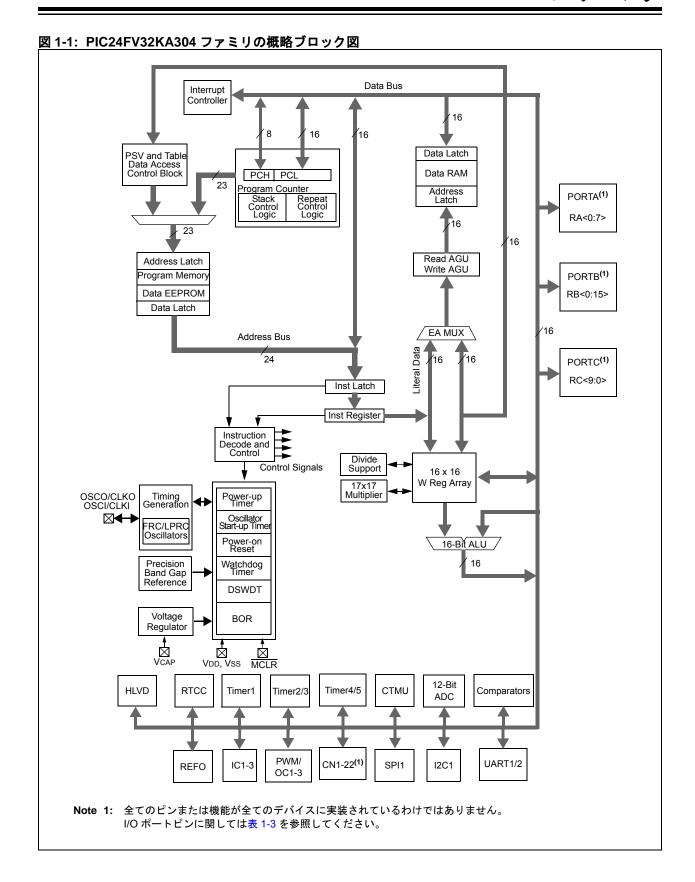


表 1-3: PIC24FV32KA304 ファミリのピン配置

<u> </u>	10241 \$321		F		- 				F	V					
			ピン番	号					ピン	番号					
機能	20 ピン PDIP/SSOP/ SOIC	20 ピン QFN	28 ピン SPDIP/SSOP/ SOIC	28 ピン QFN	44 ピン QFN/TQFP	48 ピン UQFN	20 ピン PDIP/SSOP/ SOIC	20 ピン QFN	28 ピン SPDIP/S SOP/ SOIC	28 ピン QFN	44 ピン QFN/TQFP	48 ピン UQFN	1/0	バッ ファ	内容
AN0	2	19	2	27	19	21	2	19	2	27	19	21	Ι	ANA	A/D アナログ入力
AN1	3	20	3	28	20	22	3	20	3	28	20	22	Ι	ANA	
AN2	4	1	4	1	21	23	4	1	4	1	21	23	Ι	ANA	
AN3	5	2	5	2	22	24	5	2	5	2	22	24	Ι	ANA	
AN4	6	3	6	3	23	25	6	3	6	3	23	25	Ι	ANA	
AN5	_	_	7	4	24	26	_	ı	7	4	24	26	Ι	ANA	
AN6	_	_	_	_	25	27	_	ı	_	_	25	27	Ι	ANA	
AN7	_	_	_	_	26	28	1		_	_	26	28	- 1	ANA	
AN8	_	_	_	_	27	29	1		_	_	27	29	- 1	ANA	
AN9	18	15	26	23	15	16	18	15	26	23	15	16	- 1	ANA	
AN10	17	14	25	22	14	15	17	14	25	22	14	15	- 1	ANA	
AN11	16	13	24	21	11	12	16	13	24	21	11	12	- 1	ANA	
AN12	15	12	23	20	10	11	15	12	23	20	10	11	ı	ANA	
AN13	7	4	9	6	30	33	7	4	9	6	30	33	ı	ANA	
AN14	8	5	10	7	31	34	8	5	10	7	31	34	ı	ANA	
AN15	9	6	11	8	33	36	9	6	11	8	33	36	ı	ANA	
ASCL1	_	_	15	12	42	46	_	_	15	12	42	46	I/O	I ² C™	代替 I ² C 1 クロック入力 / 出力
ASDA1	_	_	14	11	41	45	_	_	14	11	41	45	I/O	I ² C	代替 I ² C 1 デジタル入 / 出力
AVDD	20	17	28	25	17	18	20	17	28	25	17	18	ı	ANA	A/D 電源ピン
AVss	19	16	27	24	16	17	19	16	27	24	16	17	ı	ANA	
C1INA	8	5	7	4	24	26	8	5	7	4	24	26	ı	ANA	コンパレータ1入力 A (+)
C1INB	7	4	6	3	23	25	7	4	6	3	23	25	ı	ANA	コンパレータ1入力 B (-)
C1INC	5	2	5	2	22	24	5	2	5	2	22	24	ı	ANA	コンパレータ1入力C(+)
C1IND	4	1	4	1	21	23	4	1	4	1	21	23	ı	ANA	コンパレータ1入力 D (-)
C1OUT	17	14	25	22	14	15	17	14	25	22	14	15	0	_	コンパレータ 1 出力
C2INA	5	2	5	2	22	24	5	2	5	2	22	24	ı	ANA	コンパレータ 2 入力 A (+)
C2INB	4	1	4	1	21	23	4	1	4	1	21	23	ı	ANA	コンパレータ 2 入力 B (-)
C2INC	8	5	7	4	24	26	8	5	7	4	24	26	ı	ANA	コンパレータ 2 入力 C (+)
C2IND	7	4	6	3	23	25	7	4	6	3	23	25	ı	ANA	コンパレータ 2 入力 D (-)
C2OUT	14	11	20	17	7	7	11	8	16	13	43	47	0		コンパレータ2出力

PIC24FV32KA304 ファ

2 1 0. 1			<i>y y y y y y y y y y</i>	<u> </u>	<u> </u>			F\	/						
			ピン番	号					ピン	番号					
機能	20 ピン PDIP/SSOP/ SOIC	20 ピン QFN	28 ピン SPDIP/SSOP/ SOIC	28 ピン QFN	44 ピン QFN/TQFP	48 ピン UQFN	20 ピン PDIP/SSOP/ SOIC	20 ピン QFN	28 ピン SPDIP/S SOP/ SOIC	28 ピン QFN	44ピン QFN/TQFP	48 ピン UQFN	I/O	バッ ファ	内容
C3INA	18	15	26	23	15	16	18	15	26	23	15	16	I	ANA	コンパレータ 3 入力 A (+)
C3INB	17	14	25	22	14	15	17	14	25	22	14	15	- 1	ANA	コンパレータ 3 入力 B (-)
C3INC	2	19	2	27	19	21	2	19	2	27	19	21	- 1	ANA	コンパレータ 3 入力 C (+)
C3IND	4	1	4	1	21	23	4	1	4	1	21	23	- 1	ANA	コンパレータ 3 入力 D (-)
C3OUT	12	9	17	14	44	48	12	9	17	14	44	48	0	_	コンパレータ 3 出力
CLK I	7	4	9	6	30	33	7	4	9	6	30	33	- 1	ANA	メインクロック入力
CLKO	8	5	10	7	31	34	8	5	10	7	31	34	0	_	システムクロック出力
CN0	10	7	12	9	34	37	10	7	12	9	34	37	ı	ST	チャージ割り込み入力
CN1	9	6	11	8	33	36	9	6	11	8	33	36	ı	ST	
CN2	2	19	2	27	19	21	2	19	2	27	19	21	I	ST	
CN3	3	20	3	28	20	22	3	20	3	28	20	22	I	ST	
CN4	4	1	4	1	21	23	4	1	4	1	21	23	I	ST	
CN5	5	2	5	2	22	24	5	2	5	2	22	24	I	ST	
CN6	6	3	6	3	23	25	6	3	6	3	23	25	I	ST	
CN7		_	7	4	24	26		l	7	4	24	26	I	ST	
CN8	14	11	20	17	7	7		l		1	_		I	ST	
CN9			19	16	6	6		l	19	16	6	6	I	ST	
CN10			_	_	27	29		l		1	27	29	I	ST	
CN11	18	15	26	23	15	16	18	15	26	23	15	16	I	ST	
CN12	17	14	25	22	14	15	17	14	25	22	14	15	_	ST	
CN13	16	13	24	21	11	12	16	13	24	21	11	12	ı	ST	
CN14	15	12	23	20	10	11	15	12	23	20	10	11	_	ST	
CN15	-	_	22	19	9	10		ı	22	19	9	10	I	ST	
CN16	-	_	21	18	8	9	_	ı	21	18	8	9	I	ST	
CN17	-	_	_	_	3	3	_	ı	_		3	3	Ι	ST	

表 1-3: PIC24FV32KA304 ファミリのピン配置 (続き)

			F						F	٧					
			ピン種	号				ピン番号							
機能	20 ピン PDIP/SSOP/ SOIC	20 ピン QFN	28 ピン SPDIP/SSOP/ SOIC	28 ピン QFN	44 ピン QFN/TQFP	48 ピン UQFN	20 ピン PDIP/SSOP/ SOIC	20 ピン QFN	28 ピン SPDIP/S SOP/ SOIC	28 ピン QFN	44 ピン QFN/TQFP	48 ピン UQFN	I/O	バッ ファ	内容
CN18	_	_	_	_	2	2	_	_	_	_	2	2	I	ST	
CN19			_	_	5	5	_	_	_		5	5	I	ST	
CN20		_	_	_	4	4	_	_			4	4	I	ST	
CN21	13	10	18	15	1	1	13	10	18	15	1	1	I	ST	
CN22	12	9	17	14	44	48	12	9	17	14	44	48	I	ST	
CN23	11	8	16	13	43	47	11	8	16	13	43	47	I	ST	
CN24			15	12	42	46		_	15	12	42	46	I	ST	
CN25			_	_	37	40		_			37	40	I	ST	
CN26			_	_	38	41		_			38	41	I	ST	
CN27			14	11	41	45		_	14	11	41	45	I	ST	
CN28			_	_	36	39		_			36	39	I	ST	
CN29	8	5	10	7	31	34	8	5	10	7	31	34	I	ST	
CN30	7	4	9	6	30	33	7	4	9	6	30	33	_	ST	
CN31		_	_	_	26	28	_	_	_	_	26	28	_	ST	
CN32		_	_	_	25	27	_	_	_	_	25	27	-	ST	
CN33		_	_	_	32	35	_	_	_	_	32	35	-	ST	
CN34		_	_	_	35	38	_	_	_	_	35	38	-	ST	
CN35		_	_	_	12	13	_	_	_	_	12	13	-	ST	
CN36			_	_	13	14	_		_	_	13	14	- 1	ST	
CVREF	17	14	25	22	14	15	17	14	25	22	14	15	_	ANA	コンパレータ 参照電圧出力
CVREF+	2	19	2	27	19	21	2	19	2	27	19	21	ı	ANA	コンパレータ 参照電圧 正極性 入力
CVREF-	3	20	3	28	20	22	3	20	3	28	20	22	I	ANA	コンパレータ 参照電圧 負極性 入力
CTCMP	4	1	4	1	21	23	4	1	4	1	21	23	I	ANA	CTMU コンパレータ入力
CTED1	11	11	20	17	7	7	11	8	2	27	19	21	ı	ST	

表 1-3: PIC24FV32KA304 ファミリのピン配置 (続き)	続き	を 配置 を	ピン	の	<u>ا ج</u>	ア	1	/32KA304	PIC24FV	表 1-3:	3
-------------------------------------	---	----	--------	----	---	------------	---	---	----------	---------	--------	---

			F						F	1					
			ピン番	号					ピン	番号					
機能	20 ピン PDIP/SSOP/ SOIC	20 ピン QFN	28 ピン SPDIP/SSOP/ SOIC	28 ピン QFN	44 ピン QFN/TQFP	48 ピン UQFN	20 ピン PDIP/SSOP/ SOIC	20 ピン QFN	28 ピン SPDIP/S SOP/ SOIC	28 ピン QFN	44ピン QFN/TQFP	48ピン UQFN	I/O	バッ ファ	内容
CTED2	12	9	17	14	44	48	12	9	17	14	44	48	I	ST	CTMU トリガエッジ入力
CTED3	_	_	21	18	8	9	_	_	21	18	8	9	I	ST	
CTED4	5	2	5	2	22	24	5	2	5	2	22	24	I	ST	
CTED5	6	3	6	3	23	25	6	3	6	3	23	25	I	ST	
CTED6	15	12	23	20	10	11	15	12	23	20	10	11	I	ST	
CTED7	_	_	19	16	6	6	_	_	19	16	6	6	I	ST	
CTED8	13	10	18	15	1	1	13	10	18	15	1	1	I	ST	
CTED9	17	14	25	22	14	15	17	14	25	22	14	15	I	ST	
CTED10	18	15	26	23	15	16	18	15	26	23	15	16	I	ST	
CTED11	_	_	_	_	5	5	_		_	1	5	5	I	ST	1
CTED12	_	_	_	_	13	14	_		_		13	14	I	ST	
CTED13	_	_	22	19	9	10	_		22	19	9	10	I	ST	
CTPLS	16	13	24	21	11	12	16	13	24	21	11	12	0	_	CTMU パルス出力
HLVDIN	15	12	23	20	10	11	15	12	23	20	10	11	I	ST	
IC1	11	11	19	16	6	6	11	8	19	16	6	6	I	ST	HIGH/LOW 電圧検出入力
IC2	13	10	18	15	5	5	13	10	18	15	5	5	I	ST	入力キャプチャ1入力
IC3	15	12	23	20	13	14	15	12	23	20	13	14	I	ST	入力キャプチャ2入力
INT0	11	8	16	13	43	47	11	8	16	13	43	47	I	ST	入力キャプチャ3入力
INT1	17	14	25	22	14	15	17	14	25	22	14	15	I	ST	割り込み0入力
INT2	14	11	20	17	7	7	15	12	23	20	10	11	I	ST	割り込み1入力
MCLR	1	18	1	26	18	19	1	18	1	26	18	19	I	ST	割り込み2入力
OC1	11	11	20	17	7	7	11	8	16	13	43	47	0	_	出カコンペア /PWM1 出力
OC2	4	1	22	19	4	4	4	1	22	19	4	4	0	_	出力コンペア /PWM2 出力
OC3	5	2	21,5	18,2	8,12,22	9,13,24	5	2	21,5	18,2	8,12,22	9,13,24	0	_	出力コンペア /PWM3 出力
OCFA	17	14	25	22	14	15	17	14	25	22	14	15	0	_	出力コンペア フォルト A
OFCB	16	13	24	21	11,32	12,35	16	13	24	21	11,32	12,35	0	_	出力コンペア フォルト B
OSCI	7	4	9	6	30	33	7	4	9	6	30	33	I	ANA	メインオシレータ入力
OSCO	8	5	10	7	31	34	8	5	10	7	31	34	0	ANA	メインオシレータ出力
PGEC1	5	2	5	2	22	24	5	2	5	2	22	24	I/O	ST	ICSP™ クロック 1
PCED1	4	1	4	1	21	23	4	1	4	1	21	23	I/O	ST	ICSP データ 1
PGEC2	2	19	22,2	19,27	9,19	10,21	2	19	22,2	19,27	9,19	10,21	I/O	ST	ICSP クロック 2

表 1-3:	PIC24FV32KA304	ファミリ	のピン	∕配置 ((続き)	
--------	----------------	------	-----	-------	------	--

			F		•	•			F	/					
			ピン番	号					ピン	番号					
機能	20 ピン PDIP/SSOP/ SOIC	20 ピン QFN	28 ピン SPDIP/SSOP/ SOIC	28 ピン QFN	44ピン QFN/TQFP	48 ピン UQFN	20 ピン PDIP/SSOP/ SOIC	20 ピン QFN	28 ピン SPDIP/S SOP/ SOIC	28 ピン QFN	44 ピン QFN/TQFP	48 ピン UQFN	1/0	バッ ファ	内容
PGED2	3	20	21,3	18,28	8,20	9,22	3	20	21,3	18,28	8,20	9,22	I/O	ST	ICSP データ 2
PGEC3	10	7	12,15	9,12	34,42	37,46	10	7	12,15	9,12	34,42	37,46	I/O	ST	ICSP クロック 3
PGED3	9	6	11,14	8,11	33,41	36,45	9	6	11,14	8,11	33,41	36,45	I/O	ST	ICSP データ 3
RA0	2	19	2	27	19	21	2	19	2	27	19	21	I/O	ST	PORTA ピン
RA1	3	20	3	28	20	22	3	20	3	28	20	22	I/O	ST	
RA2	7	4	9	6	30	33	7	4	9	6	30	33	I/O	ST	
RA3	8	5	10	7	31	34	8	5	10	7	31	34	I/O	ST	
RA4	10	7	12	9	34	37	10	7	12	9	34	37	I/O	ST	
RA5	1	18	1	26	18	19	1	18	1	26	18	19	I/O	ST	
RA6	14	11	20	17	7	7	_	_	_	_	_		I/O	ST	
RA7	_	_	19	16	6	6	_	_	19	16	6	6	I/O	ST	
RA8	_	_	_	_	32	35	_	_	_	_	32	35	I/O	ST	
RA9	_	_	_	_	35	38	_	_	_	_	35	38	I/O	ST	
RA10	_	_	_	_	12	13	_	_	_	_	12	13	I/O	ST	
RA11	_	_	_	_	13	14	_	_	_	_	13	14	I/O	ST]
RB0	4	1	4	1	21	23	4	1	4	1	21	23	I/O	ST	PORTB ピン
RB1	5	2	5	2	22	24	5	2	5	2	22	24	I/O	ST	
RB2	6	3	6	3	23	25	6	3	6	3	23	25	I/O	ST]
RB3	_	_	7	4	24	26	_	_	7	4	24	26	I/O	ST	
RB4	9	6	11	8	33	36	9	6	11	8	33	36	I/O	ST	
RB5	_	_	14	11	41	45	_	_	14	11	41	45	I/O	ST	
RB6	_	_	15	12	42	46	_	_	15	12	42	46	I/O	ST]
RB7	11	8	16	13	43	47	11	8	16	13	43	47	I/O	ST]
RB8	12	9	17	14	44	48	12	9	17	14	44	48	I/O	ST]
RB9	13	10	18	15	1	1	13	10	18	15	1	1	I/O	ST]
RB10	_	_	21	18	8	9	_	_	21	18	8	9	I/O	ST]
RB11	_	_	22	19	9	10	_	_	22	19	9	10	I/O	ST]
RB12	15	12	23	20	10	11	15	12	23	20	10	11	I/O	ST	
RB13	16	13	24	21	11	12	16	13	24	21	11	12	I/O	ST]
RB14	17	14	25	22	14	15	17	14	25	22	14	15	I/O	ST]
RB15	18	15	26	23	15	16	18	15	26	23	15	16	I/O	ST]

表 1-3:	PIC24FV32KA304	ファミリのビ	『ン配置(続き)
--------	----------------	--------	-------	-----

			F		•				F	/					
			ピン猫	号					ピン	番号					
機能	20 ピン PDIP/SSOP/ SOIC	20 ピン QFN	28 ピン SPDIP/SSOP/ SOIC	28 ピン QFN	44 ピン QFN/TQFP	48 ピン UQFN	20 ピン PDIP/SSOP/ SOIC	20 ピン QFN	28 ピン SPDIP/S SOP/ SOIC	28 ピン QFN	44ピン QFN/TQFP	48 ピン UQFN	I/O	バッ ファ	内容
RC0	_	_	_	_	25	27	_	_	_	_	25	27	I/O	ST	PORTC ピン
RC1	_	_	_	_	26	28	_	_	_	_	26	28	I/O	ST]
RC2	_	_	_	_	27	29	_	_	_	_	27	29	I/O	ST	
RC3	_	_	_	_	36	39	_	_	_	_	36	39	I/O	ST]
RC4	_	_	_	_	37	40	_	_	_	_	37	40	I/O	ST]
RC5	_	_	_	_	38	41	_	_	_	_	38	41	I/O	ST]
RC6	_	_	_	_	2	2	_	_	_	_	2	2	I/O	ST	
RC7	_	_	_	_	3	3	_	_	_	_	3	3	I/O	ST]
RC8	_	_	_	_	4	4	_	_	_	_	4	4	I/O	ST]
RC9	_	_	_	_	5	5	_	_	_	_	5	5	I/O	ST]
REFO	18	15	26	23	15	16	18	15	26	23	15	16	0	_	参照クロック出力
RTCC	17	14	25	22	14	15	17	14	25	22	14	15	0	_	リアルタイム クロック / カレン ダ 出力
SCK1	15	12	22,23	19,20	9,10	10,11	15	12	22,23	19,20	9,10	10,11	I/O	ST	SPI1 シリアル入力 / 出力クロック
SCK2	2	19	2,14	27,11	19,38,41	21,41,45	2	19	2,14	27,11	19,38,41	21,41,45	I/O	ST	SPI2 シリアル入力 / 出力クロック
SCL1	12	9	17	14	44	48	12	9	17	14	44	48	I/O	I ² C	I2C1 クロック入力 / 出力
SCL2	18	15	26,7	23,4	15,24	16,26	18	15	26,7	23,4	15,24	16,26	I/O	I ² C	I2C2 クロック入力 / 出力
SCLKI	10	7	12	9	34	37	10	7	12	9	34	37	I	ST	デジタル セカンダリクロック入力
SDA1	13	10	18	15	1	1	13	10	18	15	1	1	I/O	I ² C	I2C1 デジタル入力 / 出力
SDA2	6	3	6	3	23	25	6	3	6	3	23	25	I/O	I ² C	I2C2 デジタル入力 / 出力
SDI1	17	14	21,25	18,22	8,14	9,15	17	14	21,25	18,22	8,14	9,15	I	ST	SPI1 シリアルデータ入力
SDI2	4	1	19,4	16,1	6,21,36	6,23,39	4	1	19,4	16,1	6,21,36	6,23,39	ı	ST	SPI2 シリアルデータ入力
SDO1	16	13	24	21	11	12	16	13	24	21	11	12	0	_	SPI1 シリアルデータ出力
SDO2	3	20	3,15	28,12	20,37,42	22,40,46	3	20	3,15	28,12	20,37,42	22,40,46	0	_	SPI2 シリアルデータ出力
SOSCI	9	6	11	8	33	36	9	6	11	8	33	36	ı	ANA	セカンダリ オシレータ入力
SOSCO	10	7	12	9	34	37	10	7	12	9	34	37	0	ANA	セカンダリ オシレータ出力
SS1	18	15	26	23	15	16	18	15	26	23	15	16	0	_	SPI1 スレーブ選択
SS2	15	12	23	20	10,35	11,38	15	12	23	20	10,35	11,38	0	_	SPI2 スレーブ選択
T1CK	13	10	18	15	1	1	13	10	18	15	1	1	ı	ST	Timer1 クロック
T2CK	18	15	26	23	15	16	18	15	26	23	15	16	I	ST	Timer2 クロック
T3CK	18	15	26	23	15	16	18	15	26	23	15	16	I	ST	Timer3 クロック
T4CK	6	3	6	3	23	25	6	3	6	3	23	25	I	ST	Timer4 クロック

表 1-3: PIC24FV32KA304 ファミリのピン配置 (続き)

X 1-3. F	10241 V321		F			,			F۱	/					
			ピン番	·号					ピン	番号				_	
機能	20 ピン PDIP/SSOP/ SOIC	20 ピン QFN	28 ピン SPDIP/SSOP/ SOIC	28 ピン QFN	44 ピン QFN/TQFP	48 ピン UQFN	20 ピン PDIP/SSOP/ SOIC	20 ピン QFN	28 ピン SPDIP/S SOP/ SOIC	28 ピン QFN	44 ピン QFN/TQFP	48ピン UQFN	I/O	バッ ファ	内容
T5CK	6	3	6	3	23	25	6	3	6	3	23	25	Ι	ST	Timer5 クロック
U1CTS	12	9	17	14	44	48	12	9	17	14	44	48	- 1	ST	UART1 送信クリア入力
U1RTS	13	10	18	15	1	1	13	10	18	15	1	1	0	_	UART1 送信要求出力
U1RX	6	3	6	3	2,23	2,25	6	3	6	3	2,23	2,25	- 1	ST	UART1 受信
U1TX	11	8	16	13	3,43	3,47	11	8	16	13	3,43	3,47	0	_	UART1 送信
U2CTS	10	7	12	9	34	37	10	7	12	9	34	37	I	ST	UART2 送信クリア入力
U2RTS	9	6	11	8	33	36	9	6	11	8	33	36	0	_	UART2 送信要求出力
U2RX	5	2	5	2	22	24	5	2	5	2	22	24	ı	ST	UART2 受信
U2TX	4	1	4	1	21	23	4	1	4	1	21	23	0	_	UART2 送信
ULPWU	4	1	4	1	21	23	4	1	4	1	21	23	I	ANA	超低消費電力復帰入力
VCAP	_	_	_	_	_	_	14	11	20	17	7	7	Р	_	コア電源
VDD	20	17	28,13	25,10	17,28,40	18,30,43	20	17	28,13	25,10	17,28,40	18,30,43	Р	_	
VREF+	2	19	2	27	19	21	2	19	2	27	19	21	I	ANA	A/D 参照電圧入力 (+)
VREF-	3	20	3	28	20	22	3	20	3	28	20	22	I	ANA	A/D 参照電圧入力 (-)
Vss	19	16	27,8	24,5	16,29,39	17,31,42	19	16	27,8	24,5	16,29,39	17,31,42	Р	_	

2.0 16 ビット マイクロコントローラ の使用にあたってのガイドライン

2.1 基本的な接続要件

PIC24FV32KA304 ファミリ 16 ビット マイクロコントローラを使用して開発を始める前に、デバイスピンの接続に関する要件に注意が必要です。

下記のピンは常に接続する必要があります。

- 全ての VDD/VSS ピン (2.2「電源ピン」参照)
- 全ての AVDD および AVSS ピン(アナログデバイス 機能を使用するかどうかに関係なく常に接続) (2.2「電源ピン」参照)
- MCLR ピン (2.3「マスタクリア (MCLR) ピン」参照)
- VCAP ピン (2.4「電圧レギュレータ ピン (VCAP)」参照)

以下のピンは、最終的なアプリケーションで使用する 場合に接続する必要があります。

- PGECx/PGEDx ピン (In-Circuit Serial Programming™ (ICSP™) とデバッグ用に使用)
 (2.5 「ICSP ピン」参照)
- OSCI/OSCO ピン(外部クロック源を使用する場合)(2.6「外部オシレータピン」参照)

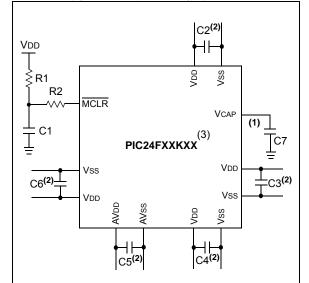
さらに、以下のピンの接続が必要な場合もあります。

 VREF+/VREF- ピン (アナログ モジュール用に外部参 照電圧を使用する場合)

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

最低限の必須ピン接続を図 2-1 に示します。

図 2-1: 推奨される最小限の接続



コンデンサと抵抗の推奨値

C1 ~ C6: 0.1 μF、20V、セラミック

C7: 10 μF、16V、タンタルまたはセラミック

R1: 10 kΩ

R2: 100 \sim 470 Ω

Note 1: VCAP ピンの接続につては 2.4「電圧レギュ レータ ピン (VCAP)」を参照してください。

- 2: 上図は、5 組の VDD/VSS および AVDD/AVSS ピンペアを持つ PIC24F の例です。これらの ピンペアの数はデバイスによって異なりま す。従って、必要なデカップリング コンデ ンサの数も異なります。
- 3: 一部のPIC24F Kデバイスはレギュレータを 備えません。

2.2 電源ピン

2.2.1 デカップリング コンデンサ

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

デカップリング コンデンサを使用する場合、以下の基準に注意してください。

- ・ コンデンサの値とタイプ: $0.1 \mu F$ (100 nF)、 $10 \sim 20 V$ のコンデンサを推奨します。このコンデンサには、共振周波数が 200 MHz 以上の低 ESR 品を使用する必要があります。セラミック コンデンサの使用を推奨します。
- ・プリント基板の配置: デカップリング コンデンサ は、できるだけピンの近くに配置する必要があります。コンデンサとデバイスを基板の同一面上に配置する事を推奨します。スペースに制約がある場合、ビアを使用してコンデンサを基板の別の層に配置する事もできますが、ピンからコンデンサまでのパターン長を1/4 インチ(6mm)以下にする必要があります。
- ・高周波ノイズへの対応: 基板が数 10 MHz 以上の高周波ノイズの影響を受ける場合、上記のデカップリングコンデンサと並列に補助コンデンサ(セラミックタイプ)を追加します。 $0.01 \sim 0.001 \mu F$ の補助コンデンサを、デカップリングコンデンサに近付けて配置します。高速回路の設計では、このようなコンデンサのペア (例: $0.1 \mu F$) を電源ピンとグランドピンのできるだけ近くに配置するよう配慮が必要です。
- ・性能の最適化:電源回路からの基板レイアウトの設計では、まずデカップリングコンデンサまでの電源の往復パターンを配置してからデバイスピンへのパターンを配置します。これにより、デカップリングコンデンサを電源系統の初段として、確実に配置します。コンデンサと電源ピン間のパターン長を最短にし、PCBの配線インダクタンスを低減する事も同様に重要です。

2.2.2 タンクコンデンサ

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

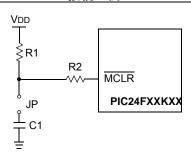
2.3 マスタクリア (MCLR) ピン

MCLR ピンは、2つのデバイス機能(デバイスリセットとデバイス プログラミング / デバッギング)を提供します。最終的なアプリケーションでプログラミング / デバッギング機能を使用しない場合、MCLR ピンをVDD に直接接続できます。ただし、部品を追加する事により、電圧サージによる予期せぬリセットに対するアプリケーションの耐性を改善できる場合があります。代表的な回路を図 2-1 に示します。アプリケーションの要件に応じて、これとは異なる回路も使用できます。

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

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

図 2-2: MCLR ピン接続の例



- **Note 1:** R1 ≤ 10 kΩ を推奨します。まず 10 kΩ を使用して、MCLR ピンの VIH と VIL が仕様を満たしているかどうかを確認する事を推奨します。
 - 2: R2 ≤ 470 Ω は、静電放電 (ESD) または電気的オーバーストレス (EOS) による MCLR ピンのブレークダウン時に、外付けコンデンサC から MCLR ピンに流れる電流を制限します。 MCLR ピンの VIH と VIL が仕様を満たすように R2 の値を選択する必要があります。

2.4 電圧レギュレータ ピン (VCAP)

Note: 本セクションの内容は、電圧レギュレー タを内蔵したPIC24F Kデバイスのみを対 象とします。

一部の PIC24F K デバイスだけが電圧レギュレータを内蔵しています。これらのデバイスは、VCAP ピンからの電圧レギュレータ出力を備えます。レギュレータ内蔵 PIC24F K デバイスでは、VCAP ピンに低 ESR (5 Ω 以下) のコンデンサを取り付けて、電圧レギュレータ出力を安定化する必要があります。VCAP ピンを VDD に直接接続しないでください。このピンは、 $10 \, \mu F$ のコンデンサを介してグランドに接続する必要があります。このコンデンサにはセラミックまたはタンタル型を使用できます。この用途に適したコンデンサの例を表 2-1 に示します。これらと同等の仕様を持つコンデ

使用するコンデンサの ESR 等価性を評価するために 図 2-3 を使用できます。

このコンデンサは VCAP ピンの近くに配置する必要があります。トレース長は 0.25 インチ (6 mm) 以下を推奨します。詳細は **29.0「電気的特性」**を参照してください。

VDD と VDDCORE については **29.0「電気的特性」**を参 照してください。

図 2-3: 推奨 VCAP コンデンサの 周波数 - ESR 性能

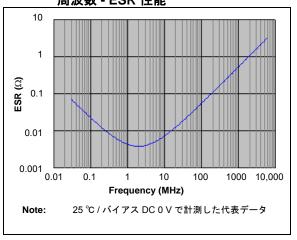


表 2-1: 推奨コンデンサの例

ンサであれば使用できます。

メーカー	製品番号	公称容量	基本許容誤差	定格電圧	温度レンジ
TDK	C3216X7R1C106K	10 μF	±10%	16 V	-55 ~ 125 ℃
TDK	C3216X5R1C106K	10 μF	±10%	16 V	-55 ~ 85 ℃
Panasonic	ECJ-3YX1C106K	10 μF	±10%	16 V	-55 ~ 125 ℃
Panasonic	ECJ-4YB1C106K	10 μF	±10%	16 V	-55 ~ 85 ℃
Murata	GRM32DR71C106KA01L	10 μF	±10%	16 V	-55 ~ 125 ℃
Murata	GRM31CR61C106KC31L	10 μF	±10%	16 V	-55 ~ 85 ℃

2.4.1 セラミック コンデンサに関する注意点

近年、数 10 µF までの表面実装型大容量 / 低電圧セラミック コンデンサのコスト効率が大幅に改善されています。低 ESR でサイズが小さく、その他の特性にも優れるセラミック コンデンサは、多くのアプリケーションにとって非常に魅力的です。

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

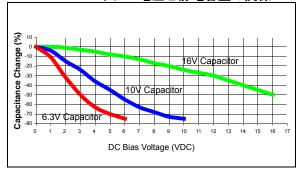
一般的に入手可能な低価格の $10 \, \mu F$ セラミック コンデンサの温度特性は X5R、X7R、Y5V です (その他のタイプも入手可能ですが、一般的ではありません)。これらのタイプのコンデンサの初期許容誤差の仕様値は、多くの場合 \pm $10 \sim 20\%$ (X5R、X7R) または \pm 20%/+80% (Y5V) です。しかし、これらのコンデンサをアプリケーション回路で使用した時の実効静電容量は、DC バイアス電圧や温度等の要因によって変化します。従って、回路内での総合的な変動幅は、仕様上の許容誤差よりも大幅に大きくなります。

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

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

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

図 2-4: セラミック コンデンサにおける DC バイアス電圧と静電容量の関係



内蔵電圧レギュレータ用にセラミック コンデンサを 使用する場合、動作電圧に対して最大定格電圧が十分 に高いコンデンサの使用を推奨します。例えば、3.3 V または 2.5 V のコア電圧に対しては、定格電圧 16 V の セラミック コンデンサを推奨します。推奨コンデンサを表 2-1 に示します。

2.5 ICSP ピン

PGC および PGD ピンは、In-Circuit Serial Programming[™] (ICSP[™]) とデバッグ用に使用します。ICSP コネクタとデバイスの ICSP ピン間のトレース長をできるだけ短くする事を推奨します。ICSP コネクタで ESD が予想される場合、数 $10\,\Omega$ の抵抗 ($100\,\Omega$ 以下) を直列に接続する事を推奨します。

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

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

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

2.6 外部オシレータピン

多くのマイクロコントローラでは、少なくとも2つのオシレータ(高速プライマリオシレータと低速セカンダリオシレータ)を選択できます(詳細は9.0「オシレータのコンフィグレーション」参照)。

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

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

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

アプリケーションの回路パターンと I/O 割り当てを検討する際に、オシレータに近い(オシレータピンと隣り合う)ポートピンと他の信号が影響を受けないように配慮する必要があります(高周波およびその他のノイズからの遮断、短い立ち上がり/立ち下がり時間の確保等)。

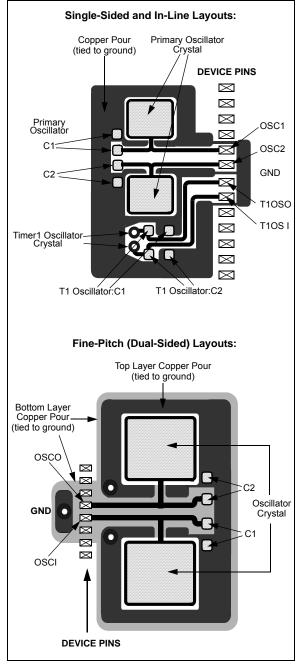
オシレータ回路の設計に関するその他の情報は、マイクロチップ社ウェブサイト (www.microchip.com) から入手可能な下記のアプリケーション ノートを参照してください。

- AN826 水晶振動子の基礎と rfPIC™ および PICmicro[®] デバイス向け水晶振動子の選択
- AN849 PICmicro® オシレータの基本設計
- AN943 PICmicro® オシレータの実用的な解析と設計
- AN949 オシレータを動作させるには

2.7 未使用 I/O ピン

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

図 2-5: オシレータ回路の推奨レイアウト



NOTE:

3.0 CPU

Note: 本書は、PIC24 デバイスの中の特定製品 グループを対象とし、それらの機能の概

グルーフを対象とし、それらの機能の概要説明を目的としています。従って本書は、包括的な参照資料の提供を意図したものではありません。CPU の詳細は「PIC24F ファミリ リファレンス マニュアル」の**セクション 2.「CPU」**(DS39703)を参照してください。

PIC24F CPU は、拡張命令セットと可変長オペコードフィールドの 24 ビット命令ワードを使用する 16 ビット(データ)改良型ハーバードアーキテクチャを採用しています。23 ビット幅のプログラム カウンタ (PC)は、最大 4M 命令のユーザプログラムメモリ空間をアドレッシングします。シングルサイクル命令プリフェッチ機構を使用する事により、スループットの維持を助け、予測可能な実行を提供します。プログラムフロー変更命令、ダブルワード移動(MOV.D)命令、テーブル命令を除く全ての命令は、1 サイクルで実行されます。REPEAT 命令を使用する事により、オーバーヘッドのないプログラムループ構造をサポートします。このループは任意の繰り返しで割り込み可能です。

PIC24Fはプログラマモデル内に16個の16ビットワーキング レジスタを備えます。各ワーキング レジスタはデータレジスタ、アドレスレジスタ、アドレスオフセット レジスタのいずれかとして使用できます。16番目のワーキング レジスタ (W15) は、割り込みおよび関数コール用のソフトウェア スタックポインタ (SSP)として動作します。

データ空間メモリマップの上位 32K バイトは、プログラムメモリまたはデータ EEPROM メモリの任意の16K ワード境界でプログラム空間に割り当てる事ができます。これは 8 ビットの PSV ページアドレス (PSVPAG) レジスタで定義します。プログラム空間からデータ空間へのマッピング機能により、全ての命令はプログラム空間に、あたかもそれがデータ空間であるかのようにして、アクセスできます。

命令セットアーキテクチャ(ISA) は PIC18 から大幅に拡張されていますが、下位互換性は満足なレベルで保たれています。PIC18 の全ての命令とアドレッシングモードは、直接サポートされるか、あるいは単純なマクロを使用してサポートされます。ISA 拡張の多くは、コンパイラ効率の観点から採用されています。

コアは命令内(オペランドなし)アドレッシング、相対アドレッシング、リテラルアドレッシング、メモリ直接アドレッシングと、アドレッシングモードの3種類のグループをサポートします。全てのモードは、レジスタ直接モードと各種のレジスタ間接モードをサポートします。各グループは、最大7種類のアドレッシングモードを提供します。各命令は、その機能的要求に応じて、事前に決められたアドレッシングモードに関連付けられています。

大部分の命令において、コアはデータ(またはプログラムデータ)メモリの読み出し、ワーキングレジスタ(データ)の読み出し、データメモリの書き込み、プログラム(命令)メモリの読み出しを1命令サイクルで実行できます。その結果、3パラメータ命令をサポート可能であるため、三項演算(A+B=C)を1サイクルで実行できます。

コアの算術演算能力とスループットを大幅に拡張するために、高速の17ビット×17ビット乗算器を内蔵しています。この乗算器は符号付き同士、符号なし同士、符合ありなし混合モードでの16ビット×16ビットまたは8×8ビット整数乗算をサポートします。全ての乗算命令は1サイクルで実行されます。

16 ビットALUは、繰り返し除算アルゴリズムをサポートする整数除算支援ハードウェアにより拡張されています。これは REPEAT 命令ループ機構と各種の繰り返し除算命令を使用して動作し、32 ビット (または 16 ビット)被除数と 16 ビット除数による符号付きおよび符号なし整数除算をサポートします。全ての除算演算は 19 サイクルを要しますが、任意のサイクル境界で割り込む事ができます。

PIC24F は、最大 8 個のノンマスカブル トラップ要因 と最大 118 個の割り込み要因を持つベクタ方式の例外 処理機構を採用しています。各割り込み要因には 7 段階の優先度を設定できます。

CPU のブロック図を図 3-1 に示します。

3.1 プログラマモデル

図 3-2 に、PIC24F のプログラマモデルを示します。プログラマモデル内の全てのレジスタはメモリに割り当てられ、命令による直接操作が可能です。

表 3-1 に各レジスタの概要を記載します。プログラマモデルに関連付けられた全てのレジスタはメモリに割り当てられます。

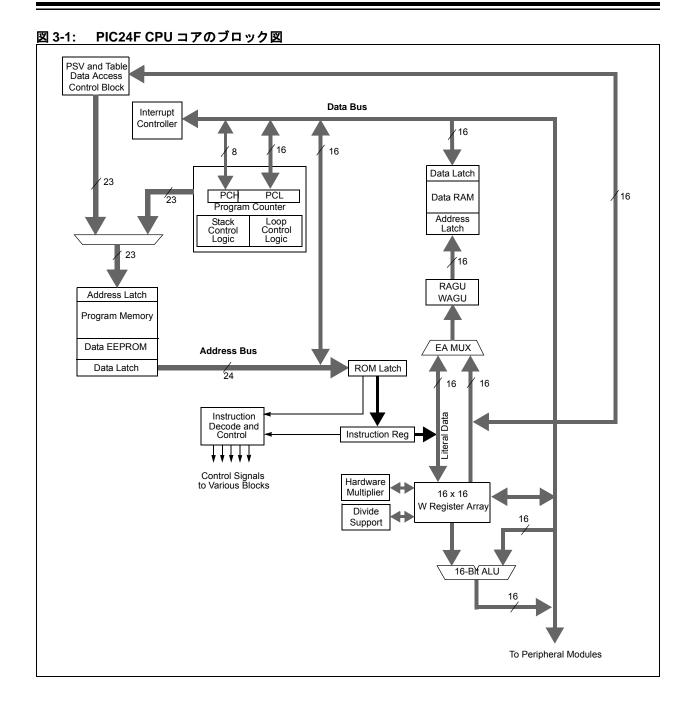
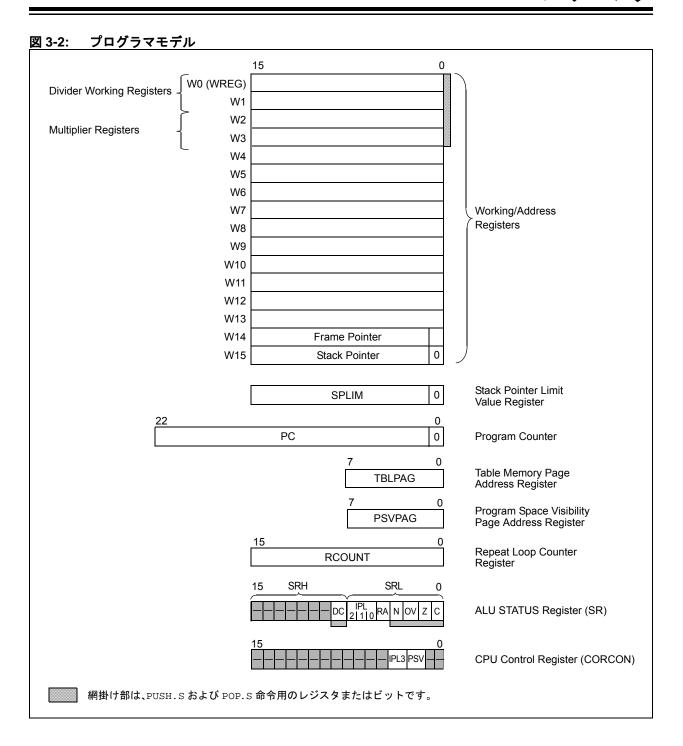


表 3-1: CPU コアレジスタ

レジスタ名	内容	
W0 ~ W15	ワーキング レジスタアレイ	
PC	23 ビット プログラムカウンタ	
SR	ALU ステータスレジスタ	
SPLIM	スタックポインタ リミット値レジスタ	
TBLPAG	テーブルメモリ ページアドレス レジスタ	
PSVPAG	PSV ページアドレス レジスタ	
RCOUNT	Repeat ループカウンタ レジスタ	
CORCON	CPU 制御レジスタ	



3.2 CPU 制御レジスタ

レジスタ 3-1: SR: ALU ステータス レジスタ

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0, HSC
_	_	_	_	_	_	_	DC
bit 15							bit 8

R/W-0, HSC ⁽¹⁾	R/W-0, HSC ⁽¹⁾	R/W-0, HSC ⁽¹⁾	R-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC
IPL2 ⁽²⁾	IPL1 ⁽²⁾	IPL0 ⁽²⁾	RA	N	OV	Z	С
bit 7							bit 0

凡例:HSC = ハードウェア セット/クリア可能ビット

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、(0) として読み出し (0) -n = POR 時の値 (0) 1 = ビットをセット (0) 0 = ビットをクリア (0) x = ビットは未知

bit 15-9 **未実装:**「0」として読み出し

bit 8 **DC**: ALU ハーフキャリー / ボロー ビット

1 = 結果の下位から4桁目のビット(バイトサイズデータの場合)または下位から8桁目のビット (ワードサイズデータの場合)からキャリーが発生した

0 = 結果の上記桁位置からキャリーは発生していない

bit 7-5 IPL<2:0>: CPU 割り込み優先度ステータスビット (1,2)

111 = CPU 割り込み優先度は 7 (15): ユーザ割り込みは無効

110 = CPU 割り込み優先度 6 (14)

101 = CPU 割り込み優先度は 5 (13)

100 = CPU 割り込み優先度 4 (12)

011 = CPU 割り込み優先度 3 (11)

010 = CPU 割り込み優先度 2 (10)

001 = CPU 割り込み優先度 1 (9)

000 = CPU 割り込み優先度 0 (8)

bit 4 RA: REPEAT ループ アクティブビット

1 = REPEAT ループを実行中

0 = REPEAT ループを実行中ではない

bit 3 N: ALU 負数結果ビット

1 = 結果は負数だった

0 = 結果は負数ではなかった (0 または正数)

OV: ALU オーバーフロー ビット

1 = 今回の符号付き (2 の補数) 算術演算でオーバーフローが発生した

0 = オーバーフローは発生していない

bit 1 **Z**: ALU ゼロビット

1=Zビットに影響する動作が過去にZビットをセットした後、Zビットはクリアされていない

0=Zビットに影響する直前の動作がZビットをクリアした(すなわち非ゼロ結果)

bit 0 **C**: ALU ハーフキャリー / ボロービット

1 = 結果の最上位ビット (MSb) からキャリーが発生した

0 = 結果の最上位ビット (MSb) からのキャリーは発生していない

Note 1: NSTDIS (INTCON1<15>) = 1 の場合、IPL ステータスビットは読み出し専用です。

2: IPL ステータスビットは IPL3 ビット (CORCON<3>) と連結して CPU 割り込み優先度 (IPL) を形成します。 カッコウの位は IPL3 = 1.0 時の IPL 値です。

す。カッコ内の値は IPL3 = 1 の時の IPL 値です。

レジスタ 3-2: CORCON: CPU 制御レジスタ

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
_	_	_	_	_	_	_	_
bit 15							bit 8

U-0	U-0	U-0	U-0	R/C-0, HSC	R/W-0	U-0	U-0
_	_	_	_	IPL3 ⁽¹⁾	PSV	_	_
bit 7							bit 0

凡例: $HSC = N - F \dot{D}_{x} T \dot{D}_{y} T \dot{D}_{y} T$

ビット

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-4 **未実装:**「0」として読み出し

bit 3 **IPL3:** CPU 割り込み優先度ステータスビット (1)

1 = CPU 優先度は 8 以上 0 = CPU 優先度は 7 以下

bit 2 **PSV**: データ空間内 PSV イネーブルビット

1 = データ空間内でプログラム空間は可視 0 = データ空間内でプログラム空間は不可視

bit 1-0 **未実装:**「0」として読み出し

Note 1: IPL3 = 1 の時、ユーザ割り込みは無効化されます。

3.3 算術論理演算ユニット (ALU)

PIC24F の ALU は 16 ビット幅を持ち、加減算、ビットシフト、論理演算の能力を備えます。特に明記がない限り、算術演算は 2 の補数に基づきます。演算によっては、ALU が SR レジスタのキャリー (C)、ゼロ(Z)、負数 (N)、オーバーフロー (OV)、ディジット キャリー (DC) ステータスビットの値に影響する場合があります。 減算では、C および DC ステータスビットはそれぞれボローおよびデジットボロー ビットとして機能します。

ALU は、使用する命令のモードに応じて 8 ビットまたは 16 ビット演算を実行できます。ALU 演算に使用するデータは、命令のアドレッシング モードに応じて、ワーキング レジスタアレイまたはデータメモリから読み出す事ができます。同様に、ALU からの出力データは、ワーキング レジスタアレイまたはデータメモリへ書き込む事ができます。

PIC24Fの CPU は、乗算と除算をハードウェアでサポートします。これには、専用のハードウェア乗算器と 16 ビット除数の除算用サポート ハードウェアが含まれます。

3.3.1 乗算器

ALU は高速の 17 x 17 ビット乗算器を備えています。 これは、下記の乗算モードをサポートします。

- 16 ビット x 16 ビット符号付き
- 16 ビットx16 ビット符号なし
- 16ビット符号付きx5ビット(リテラル)符号なし
- 16 ビット符号なしx16 ビット符号なし
- 16ビット符号なしx5ビット(リテラル)符号なし
- 16 ビット符号なし x 16 ビット符号付き
- 8 ビット符号なし x 8 ビット符号なし

3.3.2 除算器

除算ブロックは下記の除算モードをサポートします。

- 1. 32 ビット符号付き /16 ビット符号付き
- 2. 32 ビット符号なし /16 ビット符号なし
- 3. 16 ビット符号付き /16 ビット符号付き
- 4. 16 ビット符号なし /16 ビット符号なし

全ての除算命令は商を W0 に格納し、余りを W1 に格納します。16 ビット符号付きおよび符号なし DIV 命令は、16 ビットの除数および被除数用に任意のワーキング レジスタ (Wn)、32 ビットの被除数用に任意のワーキング レジスタペア (W(m + 1): Wm、m は偶数)を指定できます。除算アルゴリズムは除数 1 ビットあたり 1 サイクルを要するため、32/16 ビット命令と16/16 ビット命令の実行に必要なサイクル数は同じです。

3.3.3 マルチビット シフトのサポート

PIC24F の ALU は、シングル ビットシフトと、シング ルサイクルのマルチビット算術および論理シフトをサポートします。マルチビット シフトにはシフタブロックを使用し、最大 15 ビットの算術右シフトまたは最大 15 ビットの左シフトを 1 サイクルで実行できます。全てのマルチビット シフト命令は、オペランドソースと結果出力先の両方に対して、レジスタ直接アドレッシングのみをサポートします。

シフト操作を使用する全ての命令の要約を表 3-2 に示します。

表 3-2: シングルビットおよびマルチビット シフト操作を使用する命令

命令	内容		
ASR	ソースレジスタを1ビットまたは複数ビット右に算術シフトする		
SL	ソースレジスタを1ビットまたは複数ビット左にシフトする		
LSR	ソースレジスタを1ビットまたは複数ビット右に論理シフトする		

4.0 メモリ構成

ハーバードアーキテクチャを採用した PIC24F マイクロコントローラは、プログラムとデータ用に別々のメモリ空間とバスを備えます。このアーキテクチャでは、コード実行中にデータ空間からプログラムメモリに直接アクセスする事も可能です。

4.1 プログラム アドレス空間

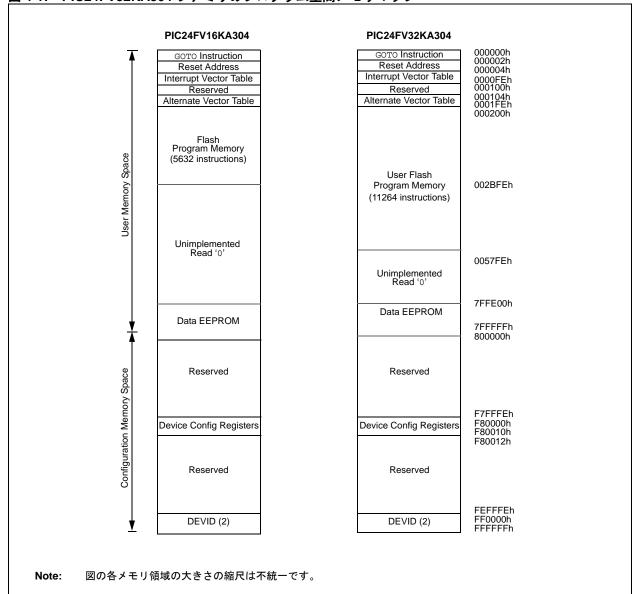
PIC24FV32KA304 ファミリは、4M 命令分のプログラム アドレスメモリ空間を備えます。この空間は24 ビット値によりアドレス指定でき、これはプログラム実行中に23 ビット プログラムカウンタから生成されるか、

あるいはテーブル操作またはデータ空間の再マッピングにより生成されます (4.3「プログラムメモリとデータメモリ間のインターフェイス」参照)。

プログラムメモリ空間に対するユーザアクセスは、アドレス領域の下位半分 (000000h ~ 7FFFFFh) に制限されます。例外として、TBLPAG<7> を使用するTBLRD/TBLWT 操作により、コンフィグレーションメモリ空間のコンフィグレーション ビットとデバイスID セクションにアクセスできます。

PIC24FV32KA304 ファミリデバイスのメモリマップ を図 4-1 に示します。

図 4-1: PIC24FV32KA304 ファミリのプログラム空間メモリマップ



4.1.1 プログラムメモリの構成

プログラムメモリ空間は、ワードアドレス指定が可能なブロックで構成されています。プログラムメモリ空間は 24 ビット幅として扱われますが、この空間の各アドレスを 2 ワード(上位ワードと下位ワード)と見なして、上位ワードの上位バイトが実装されていないと考えた方が適切です。下位ワードのアドレスは常に 偶数、上位ワードのアドレスは常に奇数です(図 4-2 参照)。

プログラムメモリ アドレスは常に下位ワードのアドレスを指し、コード実行中に2ずつインクリメントまたはデクリメントします。このような配置により、データメモリ空間のアドレス指定と互換性が保たれるため、プログラムメモリ空間内のデータへのアクセスが可能です。

4.1.2 ハードメモリ ベクタ

全てのPIC24Fデバイスのアドレス0000h~000200h は、ハードコードされたプログラムの実行ベクタ用に予約されています。デバイスリセット時にコード実行を PC の既定値から実際のコード先頭位置ヘリダイレクトするために、ハードウェア リセットベクタが用意されています。ユーザ アプリケーションは 000000h に GOTO 命令を書き込み、000002h に実際のコード先頭アドレスを書き込みます。

PIC24F は 2 つの割り込みベクタテーブルも備え、これらを 000004h \sim 0000FFh \geq 000104h \sim 0001FFh に配置します。これらのベクタテーブルにより、多数存在するデバイスの割り込み要因を個別の ISR で処理できます。割り込みベクタテーブルの詳細は 8.1 「割り込みベクタ (IVT) テーブル」を参照してください。

4.1.3 データ EEPROM

PIC24FV32KA304 ファミリでは、データ EEPROM は ユーザ プログラムメモリ空間の最後部に配置されます (アドレス 7FFE00 を先頭に、最大 7FFFF まで)。 データ EEPROM は 16 ビット幅 x 256 ワードのメモリとして構成されます。このメモリへのアクセスには、ユーザコード メモリへのアクセスと同様に、テーブル読み書き操作を使用します。

4.1.4 デバイス コンフィグレーション ワード

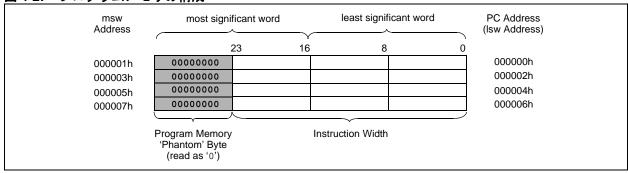
表 4-1 に、PIC24FV32KA304 ファミリのデバイス コンフィグレーション ワードのアドレスを示します。これらのアドレスのメモリマップ内の位置は図 4-1 を参照してください。

デバイス コンフィグレーション ワードの詳細は **26.0** 「その他の特殊な機能」を参照してください。

表 4-1: PIC24FV32KA304 ファミリのデバイス コンフィグレーション ワード

	/ /
コンフィグレーション ワード	アドレス
FBS	F80000
FGS	F80004
FOSCSEL	F80006
FOSC	F80008
FWDT	F8000A
FPOR	F8000C
FICD	F8000E
FDS	F80010





4.2 データアドレス空間

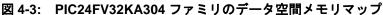
PIC24F コアは、単一の連続領域としてアクセス可能な 16 ビット幅のデータメモリ空間を備えます。データ空間へのアクセスには、読み出し用と書き込み用にそれ ぞれ 1 つずつ (計2つ)のアドレス生成ユニット (AGU)を使用します。データ空間のメモリマップを図 4-3 に示します。

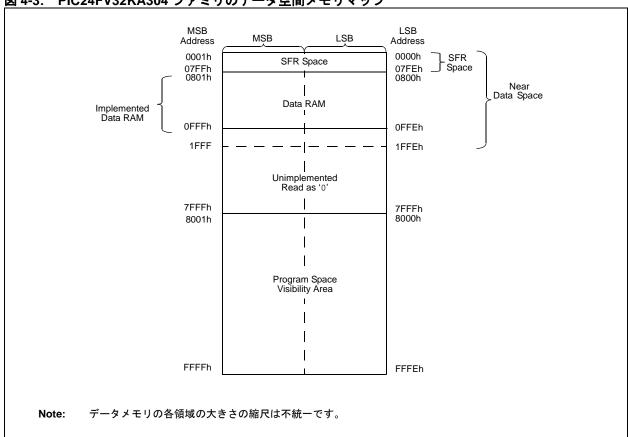
データメモリ空間内の全ての実効アドレス (EA) は 16 ビット幅を持ち、データ空間内の各バイト位置を指定します。これにより、64K バイト (32K ワード) のデータ空間アドレスレンジが得られます。データメモリ空間の下位半分 (EA<15>=0) は実装メモリアドレスに使用され、上位半分 (EA<15>=1) は PSV 領域用に予約されています (4.3.3 「PSV によるプログラムメモリからのデータ読み出し」参照)。

PIC24FV32KA304 ファミリデバイスは、1024 ワードの データメモリを実装しています。EA がこの領域外を指 した場合、全てゼロのデータが返されます。

4.2.1 データ空間の幅

データメモリ空間は、バイトアドレス指定が可能な 16 ビット幅ブロックにより構成されます。データは 16 ビットのワードとしてデータメモリとレジスタに 配置されますが、EA は全てのデータ空間をバイト単位で扱います。各ワードの下位バイト (LSB) のアドレスは偶数、上位バイト (MSB) のアドレスは奇数です。





4.2.2 データメモリの構成と配置

PIC® デバイスとの下位互換性を維持しながら、データ空間メモリの使用効率を改善するために、PIC24Fの命令セットはワード操作とバイト操作の両方をサポートします。このため、全ての実効アドレス (EA)の計算は、内部でステップスルーなワード配列メモリにスケーリングされます。例えば、ポストインクリメントするレジスタ間接アドレッシングモード [Ws++]の場合、コアはバイト操作なら Ws の値が +1 され、ワード操作なら +2 されるという事を認識します。

データバイト読み出しは、EA が指定するバイトを含む1ワードを読み出し、EAのLSBに従って上位または下位のいずれかのバイトを選択します。選択されたバイトは、データパスのLSBに置かれます。つまり、データメモリとレジスタは、共通の(ワード)アドレスを持ちながらも別々の書き込みラインを持つ並列した2バイトとして構成されています。データバイト書き込みは、バイトアドレスに一致する側のアレイまたはレジスタにのみ書き込みます。

全てのワードアクセスは、偶数アドレスに配置する必要があります。これに従わない配置のワードデータのフェッチはサポートされないため、バイト操作とワード操作を混在させる場合、または8ビットMCUコードを変換する場合は注意が必要です。不正な配置のワード読み書きを行うと、アドレスエラートラップが発生します。読み出し時にエラーが発生しても、実行中の命令は完全に実行されます。書き込み時にエラーが発生した場合、命令は実行されますが書き込みは、シーステムまたはユーザはアドレスフォルト発生前のマシンステートを分析できます。

ワーキング レジスタに対する全てのバイト書き込み は、下位バイト (LSB) に対して実行されます。上位バイト (MSB) は変更されません。

符号拡張命令 (SE) を使用すると、8 ビット符号付き データを 16 ビット符号付き値に変換できます。あるいは、16 ビット符号なしデータに変換する場合、ゼロ拡張 (\mathbb{Z} E) 命令を適切なアドレスで実行する事により、ワーキングレジスタの MSB をクリアできます。

大部分の命令はワードデータでもバイトデータでも処理可能ですが、ワードデータしか処理しない命令も一部存在する事に注意してください。

4.2.3 ニアデータ空間

0000h ~ 1FFFh の 8K バイト領域は二アデータ空間と呼びます。この空間内では、メモリ直接命令で 13 ビット絶対アドレス フィールドを介して直接アドレス指定できます。ニアデータ空間以外のデータ空間は間接的にアドレス指定できます。これとは別に、16 ビットアドレス フィールドによるメモリ直接アドレッシング (MDA) が可能な MOV 命令を使用してデータ空間全体 を ア ド レ ス 指 定 す る 事 も で き ま す。 PIC24FV32KA304 ファミリの場合、実装データメモリ全体が二アデータ空間 (NDS) 内に配置されます。

4.2.4 SFR 空間

ニアデータ空間の先頭 2Kバイト (0000h ~ 07FFh) は、 主として特殊機能レジスタ (SFR) 用に使用されます。 PIC24F コアと周辺モジュールは、これらの SFR を使 用してデバイスの動作を制御します。

SFR は制御するモジュールごとにグループ化されています。SFR 空間の多くの領域は未使用アドレスであり、これらは「0」として読み出されます。SFR 空間内で実際に SFR を実装している位置を表 4-2 に示します。表内の 1 つの実装領域は 32 バイト領域を表し、少なくとも 1 つのアドレスを SFR として実装します。実装している全ての SFR とそのアドレスを含む一覧を表 4-3 ~表 4-25 に示します。

表 4-2: SFR データ空間の実装領域

			SFR 空間アド	レス				
	xx00	xx20	xx40	xx60	xx80	xxA0	xxC0	xxE0
000h		= 7	7	ICN	害	り込み		_
100h	タイ	イマ	キャプチャ	_	コンペア	_		
200h	I2C™	UART	SPI		1	_	I/	0
300h		,	ADC/CMTU		-	_		
400h	_	_	_	_	_	_	_	_
500h	_	_	_	_	_	_	_	_
600h	_	RTC/Comp	CRC	_		_		
700h	_	_	System/DS/HLVD	NVM/PMD	_	_	_	_

凡例: --= SFR を実装していないブロック

表 4-3: CPU コア関連のレジス:	タマ	゙ップ
----------------------	----	-----

<u>秋 Ŧ⁻J.</u>	0.0-	47																
レジスタ 名	先頭 アドレス	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	全 リセット
WREG0	0000								WF	REG0								0000
WREG1	0002								WF	REG1								0000
WREG2	0004								WF	REG2								0000
WREG3	0006								WF	REG3								0000
WREG4	8000								WF	REG4								0000
WREG5	000A								WF	REG5								0000
WREG6	000C								WF	REG6								0000
WREG7	000E								WF	REG7								0000
WREG8	0010								WF	REG8								0000
WREG9	0012								WF	REG9								0000
WREG10	0014															0000		
WREG11	0016								WR	EG11								0000
WREG12	0018								WR	EG12								0000
WREG13	001A								WR	EG13								0000
WREG14	001C								WR	EG14								0000
WREG15	001E								WR	EG15								0000
SPLIM	0020								SF	PLIM								xxxx
PCL	002E								Р	CL								0000
PCH	0030	_	_	_	_	_	_	_	_	_				PCH				0000
TBLPAG	0032	_	_	_	_	_	_	_	_				TBL	_PAG				0000
PSVPAG	0034	_	PSVPAG												0000			
RCOUNT	0036	RCOUNT												xxxxx				
SR	0042	_	_	_	_	_	_	_	DC	IPL2	IPL1	IPL0	RA	N	OV	Z	С	0000
CORCON	0044	_	_	_	_	_	_	_	_	_	_	_	_	IPL3	PSV	_	_	0000
DISICNT	0052	_	_							DISIC	NT							xxxx

凡例: — = 未実装、「0」として読み出し、リセット値は 16 進数で表記

表 4-4: ICN 関連のレジスタマップ

レジス タ名	アドレス	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	全リ セット
CNPD1	0056	CN15PDE ⁽¹⁾	CN14PDE	CN13PDE	CN12PDE	CN11PDE	CN10PDE ^(1,2)	CN9PDE ⁽¹⁾	CN8PDE ⁽³⁾	CN7PDE ⁽¹⁾	CN6PDE	CN5PDE	CN4PDE	CN3PDE	CN2PDE	CN1PDE	CN0PDE	0000
CNPD2	0058	CN31PDE ^(1,2)	CN30PDE	CN29PDE	CN28PDE ^(1,2)	CN27PDE ⁽¹⁾	CN26PDE ^(1,2)	CN25PDE ^(1,2)	CN24PDE ⁽¹⁾	CN23PDE	CN22PDE	CN21PDE	CN20PDE ^(1,2)	CN19PDE ^(1,2)	CN18PDE ^(1,2)	CN17PDE ^(1,2)	CN16PDE ⁽¹⁾	0000
CNPD3	005A	_	1	_	_	_	_	_	_	_	_	_	CN36PDE ^(1,2)	CN35PDE ^(1,2)	CN34PDE ^(1,2)	CN33PDE ^(1,2)	CN32PDE ^(1,2)	0000
CNEN1	0062	CN15IE ⁽¹⁾	CN14IE	CN13IE	CN12IE	CN11IE	CN10IE ^(1,2)	CN9IE ⁽¹⁾	CN8IE ⁽³⁾	CN7IE ⁽¹⁾	CN6IE	CN5IE	CN4IE	CN3IE	CN2IE	CN1IE	CN0IE	0000
CNEN2	0064	CN31IE ^(1,2)	CN30IE	CN29IE	CN28IE ^(1,2)	CN27IE ⁽¹⁾	CN26IE ^(1,2)	CN25IE ^(1,2)	CN24IE ⁽¹⁾	CN23IE	CN22IE	CN21IE	CN20IE ^(1,2)	CN19IE ^(1,2)	CN18IE ^(1,2)	CN17IE ^(1,2)	CN16IE ⁽¹⁾	0000
CNEN3	0066	_	1	_	_	_	_	_	_	_	_	_	CN36IE ^(1,2)	CN35IE ^(1,2)	CN34IE ^(1,2)	CN33IE ^(1,2)	CN32IE ^(1,2)	0000
CNPU1	006E	CN15PUE ⁽¹⁾	CN14PUE	CN13PUE	CN12PUE	CN11PUE	CN10PUE ^(1,2)	CN9PUE ⁽¹⁾	CN8PUE ⁽³⁾	CN7PUE ⁽¹⁾	CN6PUE	CN5PUE	CN4PUE	CN3PUE	CN2PUE	CN1PUE	CN0PUE	0000
CNPU2	0070	CN31PUE ^(1,2)	CN30PUE	CN29PUE	CN28PUE ^(1,2)	CN27PUE ⁽¹⁾	CN26PUE ^(1,2)	CN25PUE ^(1,2)	CN24PUE ⁽¹⁾	CN23PUE	CN22PUE	CN21PUE	CN20PUE ^(1,2)	CN19PUE ^(1,2)	CN18PUE ^(1,2)	CN17PUE ^(1,2)	CN16PUE ⁽¹⁾	0000
CNPU3	0072	_	_	_	_	_	_	_	_	_	_	_	CN36PUE ^(1,2)	CN35PUE ^(1,2)	CN34PUE ^(1,2)	CN33PUE ^(1,2)	CN32PUE ^(1,2)	0000

PIC24FV32KA304 フ

凡例: ― = 未実装、「0」として読み出し、リセット値は 16 進数で表記

Note 1: 20 ピンデバイスはこれらのビットを実装していません。

2: 28 ピンデバイスはこれらのビットを実装していません。

3: 「FV」デバイスはこれらのビットを実装していません。

表 4-5: 割り込みコントローラ関連のレジスタマップ

双 4-3.	可り	~	ノトロ	7 1/1/1	主いレン													
レジスタ 名	アド レス	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	全 リセット
INTCON1	0080	NSTDIS	_	-	_	_	_	_	_	_	_	_	MATHERR	ADDRERR	STKERR	OSCFAIL	_	0000
INTCON2	0082	ALTIVT	DISI	_	_	_	_	_	_	_	_	_	_	_	INT2EP	INT1EP	INT0EP	0000
IFS0	0084	NVMIF	_	AD1IF	U1TXIF	U1RXIF	SPI1IF	SPF1IF	T3IF	T2IF	OC2IF	IC2IF	_	T1IF	OC1IF	IC1IF	INT0IF	0000
IFS1	0086	U2TXIF	U2RXIF	INT2IF	T5IF	T4IF	_	OC3IF	_	_	-	_	INT1IF	CNIF	CMIF	MI2C1IF	SI2C1IF	0000
IFS2	8800	_	-	-	_	_	_		_	_	-	IC3IF	_	_	-	SPI2IF	SPF2IF	0000
IFS3	A800	_	RTCIF	I	_	_	_	1		_	I	_	_		MI2C2IF	SI2C2IF	_	0000
IFS4	008C	_	_	CTMUIF	_	_	_	1	HLVDIF	_	I	_	_	CRCIF	U2ERIF	U1ERIF	_	0000
IFS5	008E	_	_	_	_	_	_	_	-		_	_	_	-	_	_	ULPWUIF	0000
IEC0	0094	NVMIE	-	AD1IE	U1TXIE	U1RXIE	SPI1IE	SPF1IE	T3IE	T2IE	OC2IE	IC2IE	_	T1IE	OC1IE	IC1IE	INT0IE	0000
IEC1	0096	U2TXIE	U2RXIE	INT2IE	T5IE	T4IE	_	OC3IE	-		_	_	INT1IE	CNIE	CMIE	MI2C1IE	SI2C1IE	0000
IEC2	0098	_	_	_	_	_	_	_	-		_	IC3IE	_	-	_	SPI2IE	SPF2IE	0000
IEC3	009A	_	RTCIE	_	_	_	_	_	-		_	_	_	-	MI2C2IE	SI2C2IE	_	0000
IEC4	009C	_	_	CTMUIE	_	_	_	_	HLVDIE		_	_	_	CRCIE	U2ERIE	U1ERIE	_	0000
IEC5	009E	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	ULPWUIE	0000
IPC0	00A4	_	T1IP2	T1IP1	T1IP0	_	OC1IP2	OC1IP1	OC1IP0	_	IC1IP2	IC1IP1	IC1IP0	_	INT0IP2	INT0IP1	INT0IP0	4444
IPC1	00A6	_	T2IP2	T2IP1	T2IP0	_	OC2IP2	OC2IP1	OC2IP0		IC2IP2	IC2IP1	IC2IP0	-	_	_	_	4444
IPC2	8A00	_	U1RXIP2	U1RXIP1	U1RXIP0	_	SPI1IP2	SPI1IP1	SPI1IP0		SPF1IP2	SPF1IP1	SPF1IP0	-	T3IP2	T3IP1	T3IP0	4444
IPC3	00AA	_	NVMIP2	NVMIP1	NVMIP0	_	_	_	-		AD1IP2	AD1IP1	AD1IP0	-	U1TXIP2	U1TXIP1	U1TXIP0	4044
IPC4	00AC	_	CNIP2	CNIP1	CNIP0	_	CMIP2	CMIP1	CMIP0	_	MI2C1P2	MI2C1P1	MI2C1P0	_	SI2C1P2	SI2C1P1	SI2C1P0	4444
IPC5	00AE	_	_	_	_	_	_	_	_	_	_	_	_	_	INT1IP2	INT1IP1	INT1IP0	0004
IPC6	00B0	_	T4IP2	T4IP1	T4IP0	_	_	_	_	_	OC3IP2	OC3IP1	OC3IP0	_	_	_	_	4040
IPC7	00B2	_	U2TXIP2	U2TXIP1	U2TXIP0	_	U2RXIP2	U2RXIP1	U2RXIP0	_	INT2IP2	INT2IP1	INT2IP0	_	T5IP2	T5IP1	T5IP0	4440
IPC8	00B4	_	_	_	_	_	_	_	-		SPI2IP2	SPI2IP1	SPI2IP0	-	SPF2IP2	SPF2IP1	SPF2IP0	0044
IPC9	00B6	_	_	_	_	_	_	_	_	_	IC3IP2	IC3IP1	IC3IP0	_	_	_	_	0040
IPC12	00BC	_	_	_	_	_	MI2C2IP2	MI2C2IP1	MI2C2IP0	_	SI2C2IP2	SI2C2IP1	SI2C2IP0	_	_	_	_	0440
IPC15	00C2	_	_	_	_	_	RTCIP2	RTCIP1	RTCIP0	_	_	_	_	_	_	_	_	0400
IPC16	00C4	_	CRCIP2	CRCIP1	CRCIP0	_	U2ERIP2	U2ERIP1	U2ERIP0	_	U1ERIP2	U1ERIP1	U1ERIP0	_	_	_	_	4440
IPC18	00C8	_	_	_	_	_	_	_	_	_	_	_	_	_	HLVDIP2	HLVDIP1	HLVDIP0	0004
IPC19	00CA	_	_	_	_	_	_	_	_	_	CTMUIP2	CTMUIP1	CTMUIP0	_	_	_	_	0040
IPC20	00CC	_	_	_	_	_	_	_	_	_	_	_	_	_	ULPWUIP2	ULPWUIP1	ULPWUIP0	0000
INTTREG	00E0	CPUIRQ	_	VHOLD	_	ILR3	ILR2	ILR1	ILR0	_	VECNUM6	VECNUM5	VECNUM4	VECNUM3	VECNUM2	VECNUM1	VECNUM0	0000

凡例: ― = 未実装、「0」として読み出し、リセット値は 16 進数で表記

-		0	
		_	
:	<	3	֡
1	2	3	
		Ś	
=	=	5	
	_		
	ä	Ś	
-		5	
		5	
1			
1		2	

表 4-6:	タイ	, <u>~</u>	関連	ரு I	パンフ	7 4	マ	· , –	Ĵ
AY TO.	<i></i>	•		UJ L	,,,	~ ~	•		,

レジスタ名	アド レス	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	全 リセット
TMR1	0100								TM	IR1								0000
PR1	0102								P	R1								FFFF
T1CON	0104	TON	I	TSIDL	1	_	_	T1ECS1	T1ECS0	1	TGATE	TCKPS1	TCKPS0	_	TSYNC	TCS	ı	0000
TMR2	0106								TM	IR2								0000
TMR3HLD	0108								TMR	3HLD								0000
TMR3	010A		TMR3											0000				
PR2	010C		PR2											0000				
PR3	010E		PR3												FFFF			
T2CON	0110	TON	_	TSIDL	_	_	_	_	_	_	TGATE	TCKPS1	TCKPS0	T32	_	TCS	_	FFFF
T3CON	0112	TON	_	TSIDL	_	_	_	_	_	_	TGATE	TCKPS1	TCKPS0	_	_	TCS	_	0000
TMR4	0114								TM	IR4								0000
TMR5HLD	0116								TMR	5HLD								0000
TMR5	0118								TM	IR5								0000
PR4	011A	PR4											FFFF					
PR5	011C	PR5											FFFF					
T4CON	011E	TON	_	TSIDL	_	_	_	_	_	_	TGATE	TCKPS1	TCKPS0	T45	_	TCS	_	0000
T5CON	0120	TON	_	TSIDL	_	_	_	_	_	_	TGATE	TCKPS1	TCKPS0	_	_	TCS	_	0000

―=未実装、「0」として読み出し、リセット値は 16 進数で表記

表 4-7: 入力キャプチャ関連のレジスタマップ

レジスタ 名	アド レス	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	全 リセット
IC1CON1	0140	_	_	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	_	_	_	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC1CON2	0142	_	_	_				_	IC32	ICTRIG	TRIGSTAT	_	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D
IC1BUF	0144									IC1BU	F							0000
IC1TMR	0146													xxxx				
IC2CON1	0148	_		ICSIDL	IC2TSEL2	IC2TSEL1	IC2TSEL0	_		_	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC2CON2	014A	_	_	_	_	_	_	_	IC32	ICTRIG	TRIGSTAT	_	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D
IC2BUF	014C									IC2BU	F							0000
IC2TMR	014E									IC2TM	R							xxxx
IC3CON1	0150	_	_	ICSIDL	IC3TSEL2	IC3TSEL1	IC3TSEL0	_	_	_	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC3CON2	0152	_	_	_				_	IC32	ICTRIG	TRIGSTAT	_	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D
IC3BUF	0154	D154 IC3BUF													0000			
IC3TMR	0156	100TUD												xxxx				

凡例: ― = 未実装、「0」として読み出し、リセット値は 16 進数で表記

表 4-8: 出力コンペア関連のレジスタマッ

レジスタ 名	アドレス	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	全 リセット
OC1CON1	0190	_		OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	ENFLT2	ENFLT1	ENFLT0	OCFLT2	OCFLT1	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OC1CON2	0192	FLTMD	FLTOUT	FLTTRIEN	OCINV	_	DCB1	DCB0	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000C
OC1RS	0194									OC1RS								0000
OC1R	0196									OC1R								0000
OC1TMR	0198													xxxx				
OC2CON1	019A		_	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	ENFLT2	ENFLT1	ENFLT0	OCFLT2	OCFLT1	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OC2CON2	019C	FLTMD	LTMD FLTOUT FLTTRIEN OCINV - DCB1 DCB0 OC32 OCTRIG TRIGSTAT OCTRIS SYNCSEL4 SYNCSEL3 SYNCSEL2 SYNCSEL1 SYNCSEL0 0										000C					
OC2RS	019E									OC2RS								0000
OC2R	01A0									OC2R								0000
OC2TMR	01A2									OC2TMR	1							xxxx
OC3CON1	01A4		_	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	ENFLT2	ENFLT1	ENFLT0	OCFLT2	OCFLT1	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OC3CON2	01A6	FLTMD	FLTOUT	FLTTRIEN	OCINV	_	DCB1	DCB0	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000C
OC3RS	01A8									OC3RS								0000
OC3R	01AA														0000			
OC3TMR	01AC		•		•	•		•		OC3TMR		•						xxxx

凡例: — = 未実装、「0」として読み出し、リセット値は 16 進数で表記

表 4-9: <u>I²C™ 関連のレジスタマップ</u>

レジスタ名	アドレス	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	全 リセット
I2C1RCV	0200	_	_	_	_	_	_	_	_				I2CR	RCV				0000
I2C1TRN	0202	_	_	_	_	_	_	_	_				I2CT	RN				00FF
I2C1BRG	0204	_	_	_	_	_	_	_	_				I2CB	RG				0000
I2C1CON	0206	I2CEN	_	I2CSIDL	SCLREL	IPMIEN	A10M	DISSLW	SMEN	GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	1000
I2C1STAT	0208	ACKSTAT	TRSTAT	_	_	_	BCL	GCSTAT	ADD10	IWCOL	I2COV	D/A	Р	S	R/W	RBF	TBF	0000
I2C1ADD	020A	_	_	_	_	_	_					I2CA	DD					0000
I2C1MSK	020C	_	_	_	_	_	_	AMSK9	AMSK8	AMSK7	AMSK6	AMSK5	AMSK4	AMSK3	AMSK2	AMSK1	AMSK0	0000
I2C2RCV	0210	_	_	_	_	_	_	_	_				I2CR	RCV				0000
I2C2TRN	0212	_	_	_	_	_	_	_	_				I2CT	RN				00FF
I2C2BRG	0214	_	_	_	_	_	_	_	_				I2CB	RG				0000
I2C2CON	0216	I2CEN	_	I2CSIDL	SCLREL	IPMIEN	A10M	DISSLW	SMEN	GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	1000
I2C2STAT	0218	ACKSTAT	TRSTAT	_	_	_	BCL	GCSTAT	ADD10	IWCOL	I2COV	D/A	Р	S	R/W	RBF	TBF	0000
I2C2ADD	021A	_	_	_	_	_	_					I2CA	DD					0000
I2C2MSK	021C	_	_	_	_	_	_	AMSK9	AMSK8	AMSK7	AMSK6	AMSK5	AMSK4	AMSK3	AMSK2	AMSK1	AMSK0	0000

凡例: — = 未実装、「0」として読み出し、リセット値は 16 進数で表記

表 4-10: UART 関連のレジスタマップ

<u>4X T-1U.</u>	OAI\ I	内足りに	<u> </u>	<u> </u>														
レジスタ 名	アド レス	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	全 リセット
U1MODE	0220	UARTEN	_	USIDL	IREN	RTSMD	_	UEN1	UEN0	WAKE	LPBACK	ABAUD	RXINV	BRGH	PDSEL1	PDSEL0	STSEL	0000
U1STA	0222	UTXISEL1	UTXINV	UTXISEL0	_	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA	0110
U1TXREG	0224	_	_	_	_	_	1	_				U1T	XREG					xxxx
U1RXREG	0226	_	_	_	_	_	1	_				U1R	XREG					0000
U1BRG	0228								ı	BRG								0000
U2MODE	0230	UARTEN	_	USIDL	IREN	RTSMD	1	UEN1	UEN0	WAKE	LPBACK	ABAUD	RXINV	BRGH	PDSEL1	PDSEL0	STSEL	0000
U2STA	0232	UTXISEL1	UTXINV	UTXISEL0	1	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA	0110
U2TXREG	0234	_	_	_	1	_	I	_				U2T	XREG					xxxx
U2RXREG	0236	_	_	_	_	_	1	_		•		U2F	XREG	•	•	•		0000
U2BRG	0238									BRG								0000

凡例: — = 未実装、「0」として読み出し、リセット値は 16 進数で表記

アド レス	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	全 リセット
0240	SPIEN	1	SPISIDL	-	_	SPIBEC2	SPIBEC1	SPIBEC0	SRMPT	SPIROV	SR1MPT	SISEL2	SISEL1	SISEL0	SPITBF	SPIRBF	0000
0242	_	_	_	DISSCK	DISSDO	MODE16	SMP	CKE	SSEN	CKP	MSTEN	SPRE2	SPRE1	SPRE0	PPRE1	PPRE0	0000
0244	FRMEN	SPIFSD	SPIFPOL	_	_	_	_	_	_	_	_		_	_	SPIFE	SPIBEN	0000
0248								SPI1	BUF								0000
0260	SPIEN	_	SPISIDL	_	_	SPIBEC2	SPIBEC1	SPIBEC0	SRMPT	SPIROV	SRXMPT	SISEL2	SISEL1	SISEL0	SPITBF	SPIRBF	0000
0262	_	_	_	DISSCK	DISSDO	MODE16	SMP	CKE	SSEN	CKP	MSTEN	SPRE2	SPRE1	SPRE0	PPRE1	PPRE0	0000
0264	FRMEN	SPIFSD	SPIFPOL	_	_	_	_	_	_	_	_		_	_	SPIFE	SPIBEN	0000
0268				•	•		•	SPI2	BUF		•	•		•	•		0000
	0240 0242 0244 0248 0260 0262 0264	0240 SPIEN 0242 — 0244 FRMEN 0248 0260 SPIEN 0262 — 0264 FRMEN	DET 19 BIT 14 0240 SPIEN — 0242 — — 0244 FRMEN SPIFSD 0248 — — 0260 SPIEN — 0262 — — 0264 FRMEN SPIFSD	DZ Bit 13 Bit 14 Bit 13 0240 SPIEN — SPISIDL 0242 — — — 0244 FRMEN SPIFSD SPIFPOL 0248 — — SPISIDL 0260 SPIEN — SPISIDL 0262 — — — 0264 FRMEN SPIFSD SPIFPOL	DET 13 BIT 13 BIT 13 BIT 12 0240 SPIEN — SPISIDL — 0242 — — — DISSCK 0244 FRMEN SPIFSD SPIFPOL — 0248 — SPISIDL — 0260 SPIEN — SPISIDL — 0262 — — DISSCK 0264 FRMEN SPIFSD SPIFPOL —	DIX BIT 13 BIT 13 BIT 12 BIT 11 0240 SPIEN — SPISIDL — — 0242 — — — DISSCK DISSDO 0244 FRMEN SPIFSD SPIFPOL — — 0248 — — SPISIDL — — 0260 SPIEN — SPISIDL — — 0262 — — — DISSCK DISSDO 0264 FRMEN SPIFSD SPIFPOL — —	DET IS BIT IS<	0240 SPIEN — SPISIDL — SPIBEC2 SPIBEC1 0242 — — — DISSCK DISSDO MODE16 SMP 0244 FRMEN SPIFSD SPIFPOL — — — — 0248 — SPIEN — SPISIDL — — SPIBEC2 SPIBEC1 0260 SPIEN — SPISIDL — — SPIBEC2 SPIBEC1 0262 — — — DISSCK DISSDO MODE16 SMP 0264 FRMEN SPIFSD SPIFPOL — — — — —	0240 SPIEN — SPISIDL — SPIBEC2 SPIBEC1 SPIBEC0 0242 — — DISSCK DISSDO MODE16 SMP CKE 0244 FRMEN SPIFSD SPIFPOL — — — — — 0248 SPIEN — SPISIDL — SPIBEC2 SPIBEC1 SPIBEC0 0260 SPIEN — SPISIDL — — SPIBEC2 SPIBEC1 SPIBEC0 0262 — — — DISSCK DISSDO MODE16 SMP CKE 0264 FRMEN SPIFSD SPIFPOL — — — — — — —	0240 SPIEN — SPISIDL — SPIBEC2 SPIBEC1 SPIBEC0 SRMPT 0242 — — — DISSCK DISSDO MODE16 SMP CKE SSEN 0244 FRMEN SPIFSD SPIFPOL — — — — — — 0248 SPIEN — SPISIDL — — SPIBEC2 SPIBEC1 SPIBEC0 SRMPT 0260 SPIEN — SPISIDL — — SPIBEC2 SPIBEC1 SPIBEC0 SRMPT 0262 — — — DISSCK DISSDO MODE16 SMP CKE SSEN 0264 FRMEN SPIFSD SPIFPOL —	0240 SPIEN — SPISIDL — SPIBEC2 SPIBEC1 SPIBEC0 SRMPT SPIROV 0242 — — — DISSCK DISSDO MODE16 SMP CKE SSEN CKP 0244 FRMEN SPIFSD SPIFPOL — — — — — — — 0248 SPIEN — SPISIDL — — SPIBEC2 SPIBEC1 SPIBEC0 SRMPT SPIROV 0262 — — — DISSCK DISSDO MODE16 SMP CKE SSEN CKP 0264 FRMEN SPIFSD SPIFPOL — <t< td=""><td>0240 SPIEN — SPISIDL — SPIBEC2 SPIBEC1 SPIBEC0 SRMPT SPIROV SR1MPT 0242 — — — DISSCK DISSDO MODE16 SMP CKE SSEN CKP MSTEN 0244 FRMEN SPIFSD SPIFPOL — — — — — — — — 0248 SPIEN — SPISIDL — — SPIBEC2 SPIBEC1 SPIBEC0 SRMPT SPIROV SRXMPT 0260 SPIEN — SPISIDL — — SPIBEC2 SPIBEC1 SPIBEC0 SRMPT SPIROV SRXMPT 0262 — — — DISSCK DISSDO MODE16 SMP CKE SSEN CKP MSTEN 0264 FRMEN SPIFSD SPIFPOL — — — — — — — — — — — — —</td><td>0240 SPIEN — SPISIDL — SPIBEC2 SPIBEC1 SPIBEC0 SRMPT SPIROV SR1MPT SISEL2 0242 — — — DISSCK DISSDO MODE16 SMP CKE SSEN CKP MSTEN SPRE2 0244 FRMEN SPIFSD SPIFPOL —</td><td>0240 SPIEN — SPISIDL — SPIBEC2 SPIBEC1 SPIBEC0 SRMPT SPIROV SR1MPT SISEL2 SISEL1 0240 SPIEN — — DISSCK DISSDO MODE16 SMP CKE SSEN CKP MSTEN SPRE2 SPRE1 0244 FRMEN SPIFSD SPIFPOL —</td><td>0240 SPIEN SPISIDL — SPIBEC2 SPIBEC1 SPIBEC0 SRMPT SPIROV SR1MPT SISEL2 SISEL1 SISEL0 0240 SPIEN — SPISIDL — — SPIBEC2 SPIBEC0 SRMPT SPIROV SR1MPT SISEL2 SISEL1 SISEL0 0242 — — DISSCK DISSOK DISSOK MODE16 SMP CKE SSEN CKP MSTEN SPRE2 SPRE1 SPRE0 0244 FRMEN SPIESD SPIFPOL —</td><td>0240 SPIEN — SPISIDL — SPIBEC2 SPIBEC1 SPIBEC0 SRMPT SPIROV SR1MPT SISEL2 SISEL1 SISEL0 SPITBF 0240 SPIEN — — DISSCK DISSDO MODE16 SMP CKE SSEN CKP MSTEN SPRE2 SPRE1 SPRE0 PPRE1 0244 FRMEN SPIFSD SPIFPOL — — — — — — — — — SPIE 0248 SPIEN — SPISIDL — — SPIBEC2 SPIBEC1 SPIBEC0 SRMPT SPIROV SRXMPT SISEL1 SISEL0 SPITBF 0260 SPIEN — SPISIDL — — SPIBEC2 SPIBEC0 SRMPT SPIROV SRXMPT SISEL2 SISEL1 SISEL0 SPITBF 0262 — — — DISSCK DISSDO MODE16 SMP CKE SSEN CKP MSTEN</td><td>UX Bit 13 Bit 14 Bit 13 Bit 12 Bit 11 Bit 10 Bit 9 Bit 0 Bit 7 Bit 0 Bit 7 Bit 0 Bit 3 Bit 3 Bit 2 Bit 1 Bit 0 0240 SPIEN — SPISIDL — — SPIBEC2 SPIBEC1 SPIBEC0 SRMPT SPIROV SR1MPT SISEL2 SISEL1 SISEL0 SPITBF SPIRBF 0242 — — — DISSOK DISSOK DISSOK DISSOK SMP CKE SSEN CKP MSTEN SPIED SPIED SPIEBE SPIBEN 0248 SPIEN — SPISIDL — — SPIBEC2 SPIBEC1 SPIBEC0 SRMPT SPIROV SRXMPT SISEL2 SISEL1 SISEL0 SPITBF SPIRBF 0260 SPIEN — SPISIDL — — SPIBEC2 SPIBEC0 SRMPT SPIROV SRXMPT SISEL2 SISEL1 SISEL0 SPITBF SPIRBF</td></t<>	0240 SPIEN — SPISIDL — SPIBEC2 SPIBEC1 SPIBEC0 SRMPT SPIROV SR1MPT 0242 — — — DISSCK DISSDO MODE16 SMP CKE SSEN CKP MSTEN 0244 FRMEN SPIFSD SPIFPOL — — — — — — — — 0248 SPIEN — SPISIDL — — SPIBEC2 SPIBEC1 SPIBEC0 SRMPT SPIROV SRXMPT 0260 SPIEN — SPISIDL — — SPIBEC2 SPIBEC1 SPIBEC0 SRMPT SPIROV SRXMPT 0262 — — — DISSCK DISSDO MODE16 SMP CKE SSEN CKP MSTEN 0264 FRMEN SPIFSD SPIFPOL — — — — — — — — — — — — —	0240 SPIEN — SPISIDL — SPIBEC2 SPIBEC1 SPIBEC0 SRMPT SPIROV SR1MPT SISEL2 0242 — — — DISSCK DISSDO MODE16 SMP CKE SSEN CKP MSTEN SPRE2 0244 FRMEN SPIFSD SPIFPOL —	0240 SPIEN — SPISIDL — SPIBEC2 SPIBEC1 SPIBEC0 SRMPT SPIROV SR1MPT SISEL2 SISEL1 0240 SPIEN — — DISSCK DISSDO MODE16 SMP CKE SSEN CKP MSTEN SPRE2 SPRE1 0244 FRMEN SPIFSD SPIFPOL —	0240 SPIEN SPISIDL — SPIBEC2 SPIBEC1 SPIBEC0 SRMPT SPIROV SR1MPT SISEL2 SISEL1 SISEL0 0240 SPIEN — SPISIDL — — SPIBEC2 SPIBEC0 SRMPT SPIROV SR1MPT SISEL2 SISEL1 SISEL0 0242 — — DISSCK DISSOK DISSOK MODE16 SMP CKE SSEN CKP MSTEN SPRE2 SPRE1 SPRE0 0244 FRMEN SPIESD SPIFPOL —	0240 SPIEN — SPISIDL — SPIBEC2 SPIBEC1 SPIBEC0 SRMPT SPIROV SR1MPT SISEL2 SISEL1 SISEL0 SPITBF 0240 SPIEN — — DISSCK DISSDO MODE16 SMP CKE SSEN CKP MSTEN SPRE2 SPRE1 SPRE0 PPRE1 0244 FRMEN SPIFSD SPIFPOL — — — — — — — — — SPIE 0248 SPIEN — SPISIDL — — SPIBEC2 SPIBEC1 SPIBEC0 SRMPT SPIROV SRXMPT SISEL1 SISEL0 SPITBF 0260 SPIEN — SPISIDL — — SPIBEC2 SPIBEC0 SRMPT SPIROV SRXMPT SISEL2 SISEL1 SISEL0 SPITBF 0262 — — — DISSCK DISSDO MODE16 SMP CKE SSEN CKP MSTEN	UX Bit 13 Bit 14 Bit 13 Bit 12 Bit 11 Bit 10 Bit 9 Bit 0 Bit 7 Bit 0 Bit 7 Bit 0 Bit 3 Bit 3 Bit 2 Bit 1 Bit 0 0240 SPIEN — SPISIDL — — SPIBEC2 SPIBEC1 SPIBEC0 SRMPT SPIROV SR1MPT SISEL2 SISEL1 SISEL0 SPITBF SPIRBF 0242 — — — DISSOK DISSOK DISSOK DISSOK SMP CKE SSEN CKP MSTEN SPIED SPIED SPIEBE SPIBEN 0248 SPIEN — SPISIDL — — SPIBEC2 SPIBEC1 SPIBEC0 SRMPT SPIROV SRXMPT SISEL2 SISEL1 SISEL0 SPITBF SPIRBF 0260 SPIEN — SPISIDL — — SPIBEC2 SPIBEC0 SRMPT SPIROV SRXMPT SISEL2 SISEL1 SISEL0 SPITBF SPIRBF

凡例: — = 未実装、「0」として読み出し、リセット値は 16 進数で表記

表 4-12: PORTA 関連のレジスタマップ

レジスタ 名	アド レス	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11 ^(2,3)	Bit 10 ^(2,3)	Bit 9 ^(2,3)	Bit 8 ^(2,3)	Bit 7 ⁽²⁾	Bit 6 ⁽⁴⁾	Bit 5 ⁽¹⁾	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	全 リセット
TRISA	02C0	_	_	_	_	TRISA11	TRISA10	TRISA9	TRISA8	TRISA7	TRISA6	_	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	00DF
PORTA	02C2	_	_	_	_	RA11	RA10	RA9	RA8	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	xxxx
LATA	02C4	_	_	_	_	LATA11	LATA10	LATA9	LATA8	LATA7	LATA6	_	LATA4	LATA3	LATA2	LATA1	LATA0	xxxx
ODCA	02C6	_	ı	-	-	ODA11	ODA10	ODA9	ODA8	ODA7	ODA6	_	ODA4	ODA3	ODA2	ODA1	ODA0	0000

凡例: — = 未実装、「0」として読み出し、リセット値は 16 進数で表記

Note 1: このビットは MCLRE = 1 の時にのみ利用できます。

2: 20 ピンデバイスはこれらのビットを実装していません。

3: 28 ピンデバイスはこれらのビットを実装していません。

4: 「FV」デバイスはこれらのビットを実装していません。

表 4-13: PORTB 関連のレジスタマップ

<u>奴 〒 1、</u>	<u> </u>		EVI V /	<u> </u>														
レジス タ名	アドレス	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11 ⁽¹⁾	Bit 10 ⁽¹⁾	Bit 9	Bit 8	Bit 7	Bit 6 ⁽¹⁾	Bit 5 ⁽¹⁾	Bit 4	Bit 3 ⁽¹⁾	Bit 2	Bit 1	Bit 0	全 リセット
TRISB	02C8	TRISB15	TRISB14	TRISB13	TRISB12	TRISB11	TRISB10	TRISB9	TRISB8	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	FFFF
PORTB	02CA	RB15	RB14	RB13	RB12	RB11	RB10	RB9	RB8	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx
LATB	02CC	LATB15	LATB14	LATB13	LATB12	LATB11	LATB10	LATB9	LATB8	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	xxxx
ODCB	02CE	ODB15	ODB14	ODB13	ODB12	ODB11	ODB10	ODB9	ODB8	ODB7	ODB6	ODB5	ODB4	ODB3	ODB2	ODB1	ODB0	0000

凡例: — = 未実装、「0」として読み出し、リセット値は 16 進数で表記

Note 1: 20 ピンデバイスはこれらのビットを実装していません。

© 2011 Microchip Technology Inc.

表 4-14: PORTC 関連のレジスタマップ ⁽¹⁾

レジス タ名	アドレス	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	全 リセット
TRISC	02D0	_	_	1	_	-	_	TRISC9	TRISC8	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	03FF
PORTC	02D2	1	_	-	-	_	_	RC9	RC8	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx
LATC	02D4	1	_	-	-	_	_	LATC9	LATC8	LATC7	LATC6	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	xxxx
ODCC	02D6	_	_		_		-	ODC9	ODC8	ODC7	ODC6	ODC5	ODC4	ODC3	ODC2	ODC1	ODC0	0000

凡例: ― = 未実装、「0」として読み出し、リセット値は 16 進数で表記

Note 1: 20/28 ピンデバイスは PORTC を実装していません。

表 4-15: パッド コンフィグレーション レジスタマップ

レジスタ 名	アドレス	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	全リセット
PADCFG1	02FC	_	_			-	_		_	-	_	SMBUSDEL2	SMBUSDEL1		_	_	_	0000

凡例: — = 未実装、「0」として読み出し、リセット値は 16 進数で表記

表 4-16: ADC 関連のジスタマップ

<u> 3X ∓⁻10. /</u>	700		<u> </u>															
レジスタ 名	アド レス	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	全 リセット
ADC1BUF0	0300								ADC1BUF	0								xxxx
ADC1BUF1	0302								ADC1BUF	1								xxxx
ADC1BUF2	0304								ADC1BUF	2								xxxx
ADC1BUF3	0306								ADC1BUF	3								xxxx
ADC1BUF4	0308								ADC1BUF	4								xxxx
ADC1BUF5	030A								ADC1BUF	5								xxxx
ADC1BUF6	030C								ADC1BUF	6								xxxx
ADC1BUF7	030E								ADC1BUF	7								xxxx
ADC1BUF8	0310								ADC1BUF	8								xxxx
ADC1BUF9	0312								ADC1BUF	9								xxxx
ADC1BUF10	0314								ADC1BUF1	0								xxxx
ADC1BUF11	0316								ADC1BUF	11								xxxx
ADC1BUF12	0318								ADC1BUF1	2								xxxx
ADC1BUF13	031A								ADC1BUF1	3								xxxx
ADC1BUF14	031C								ADC1BUF1	4								xxxx
ADC1BUF15	031E								ADC1BUF1	5								xxxx
ADC1BUF16	0320								ADC1BUF1	16								xxxx
ADC1BUF17	0322								ADC1BUF1	17								xxxx
AD1CON1	0340	ADON	_	ADSIDL	_	_	_	FORM1	FORM0	SSRC3	SSRC2	SSRC1	SSRC0	1	ASAM	SAMP	DONE	0000
AD1CON2	0342	PVCFG1	PVCFG0	NVCFG0	OFFCAL	BUFREGEN	CSCNA	-	_	BUFS	SMPI4	SMPI3	SMPI2	SMPI1	SMPI0	BUFM	ALTS	0000
AD1CON3	0344	ADRC	EXTSAM	_	SAMC4	SAMC3	SAMC2	SAMC1	SAMC0	ADCS7	ADCS6	ADCS5	ADCS4	ADCS3	ADCS2	ADCS1	ADCS0	0000
AD1CHS	0348	CH0NB2	CH0NB1	CH0NB0	CH0SB4	CH0SB3	CH0SB2	CH0SB1	CH0SB0	CH0NA2	CH0NA1	CH0NA0	CH0SA4	CH0SA3	CH0SA2	CH0SA1	CH0SA0	0000
AD1CSSH	034E	_	CSSL30	CSSL29	CSSL28	CSSL27	CSSL26	-	_	_	_	_	_	1	_	CSSL17	CSSL16	0000
AD1CSSL	0350	CSSL15	CSSL14	CSSL13	CSSL12	CSSL11	CSSL10	CSSL9	CSSL8	CSSL7	CSSL6	CSSL5	CSSL4	CSSL3	CSSL2	CSSL1	CSSL0	0000
AD1CON5	0354	ASEN	LPEN	CTMUREQ	BGREQ	VRSREQ		ASINT1	ASINT0	_	_	_	_	WM1	WMO	CM1	CM0	0000
AD1CHITH	0356	_	_	_	_	_	_	_	_	_	_	_	_	_	_	CHH17	CHH16	0000
AD1CHITL	0358	CHH15	CHH14	CHH13	CHH12	CHH11	CHH10	CHH9	CHH8	CHH7	CHH6	CHH5	CHH4	CHH3	CHH2	CHH1	CHH0	0000

PIC24FV32KA304 ファ

―=未実装、「0」として読み出し、リセット値は16進数で表記 凡例:

表 4-17: CTMU 関連のレジスタマップ

L	レジスタ名	アド レス	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	全 リセット
C-	TMUCON1	035A	CTMUEN	_	CTMUSIDL	TGEN	EDGEN	EDGSEQEN	IDISSEN	CTTRIG	_						_	_	0000
C-	TMUCON2	035C	EDG1EDGE	EDG1POL	EDG1SEL3	EDG1SEL2	EDG1SEL1	EDG1SEL0	EDG2	EDG1	EDG2EDGE	EDG2POL	EDG2SEL3	EDG2SEL2	EDG2SEL1	EDG2SEL0	_	_	0000
C-	TMUICON	035E	ITRIM5	ITRIM4	ITRIM3	ITRIM2	ITRIM1	ITRIM0	IRNG1	IRNG0	_	<u> </u>		I = I	I = I	<u> </u>		_	0000
АΓ	D1CTMUENH	0360									_					_	CTMEN17	CTMEN16	0000
АΓ)1CTMUENL	0362	CTMEN15	CTMEN14	CTMEN13	CTMEN12	CTMEN11	CTMEN10	CTMEN9	CTMEN8	CTMEN7	CTMEN6	CTMEN5	CTMEN4	CTMEN3	CTMEN2	CTMEN1	CTMEN0	0000

PIC24FV32KA304

1例: — = 未実装、「0」として読み出し、リセット値は 16 進数で表記

表 4-18: アナログ選択関連のレジスタマップ

レジスタ名	アド レス	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	全 リセット
ANSA	04E0	_	_	_	_	_	_	_	_	_	_	_	_	ANSA3	ANSA2	ANSA1	ANSA0	000F
ANSB	04E2	ANSB15	ANSB14	ANSB13	ANSB12	_	_	_	_	_	_	_	ANSB4	ANSB3 ⁽¹⁾	ANSB2	ANSB1	ANSB0	F01F
ANSC	04E4	_	_	_	_	_	_	_	_	_	_	_	_	_	ANSC2 ^(1,2)	ANSC1 ^(1,2)	ANSC0 ⁽¹⁾	0007

凡例: — = 未実装、「0」として読み出し、リセット値は 16 進数で表記

Note 1: 20 ピンデバイスはこれらのビットを実装していません。 2: 28 ピンデバイスはこれらのビットを実装していません。

表 4-19: リアルタイム クロック / カレンダ関連のレジスタマップ

	<u>, , , , , , , , , , , , , , , , , , , </u>		<u>, , , , , , , , , , , , , , , , , , , </u>	7 1 75 0 2	7 12/2	, , , , , , , , , , , , , , , , , , , 	<u> </u>											
レジスタ名	アドレス	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	全 リセット
ALRMVAL	0620							ALRI	ЛVAL									xxxx
ALCFGRPT	0622	ALRMEN	CHIME	AMASK3	AMASK2	AMASK1	AMASK0	ALRMPTR1	ALRMPTR0	ARPT7	ARPT6	ARPT5	ARPT4	ARPT3	ARPT2	ARPT1	ARPT0	0000
RTCVAL	0624							RTC	VAL									xxxx
RCFGCAL	0626	RTCEN	_	RTCWREN	RTCSYNC	HALFSEC	RTCOE	RTCPTR1	RTCPTR0	CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	0000
RTCPWC	0628	PWCEN	PWCPOL	PWCCPRE	PWCSPRE	RTCCLK1	RTCCLK0	RTCOUT1	RTCOUT0	_	_	_	_	_	_	_	_	xxxx

凡例: — = 未実装、「0」として読み出し、リセット値は 16 進数で表記

表 4-20: トリプル コンパレータ関連のレジスタマップ

	. , ,	,, ,, ,,	•••		~ ~ ~ ~ ~ ~													
レジスタ 名	アド レス	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	全 リセット
CMSTAT	0630	CMIDL	_	_	_	_	C3EVT	C2EVT	C1EVT	_	_	_	_	_	C3OUT	C2OUT	C1OUT	xxxx
CVRCON	0632	_	_	_	_	_	_	_	_	CVREN	CVROE	CVRSS	CVR4	CVR3	CVR2	CVR1	CVR0	0000
CM1CON	0634	CON	COE	CPOL	CLPWR	_	_	CEVT	COUT	EVPOL1	EVPOL0	_	CREF	_	_	CCH1	CCH0	xxxx
CM2CON	0636	CON	COE	CPOL	CLPWR	_	_	CEVT	COUT	EVPOL1	EVPOL0	_	CREF	_	_	CCH1	CCH0	0000
CM3CON	0638	CON	COE	CPOL	CLPWR	_	_	CEVT	COUT	EVPOL1	EVPOL0	_	CREF	_	_	CCH1	CCH0	0000

凡例: — = 未実装、「0」として読み出し、リセット値は 16 進数で表記

DS39995B_JP - p. 51

レジスタ 名	アドレス	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	全 リセット
CRCCON1	0640	CRCEN	_	CSIDL	VWORD4	VWORD3	VWORD2	VWORD1	VWORD0	CRCFUL	CRCMPT	CRCISEL	CRCGO	LENDIAN	_	_	_	0000
CRCCON2	0642	_	_	_	DWIDTH4	DWIDTH3	DWIDTH2	DWIDTH1	DWIDTH0	_	_	_	PLEN4	PLEN3	PLEN2	PLEN1	PLEN0	0000
CRCXORL	0644	X15	X14	X13	X12	X11	X10	Х9	X8	X7	X6	X5	X4	X3	X2	X1	_	0000
CRCXORH	0646	X31	X30	X29	X28	X27	X26	X25	X24	X23	X22	X21	X20	X19	X18	X17	X16	0000
CRCDATL	0648								CRCDA	TL								xxxx
CRCDATH	064A								CRCDA	ТН								xxxx
CRCWDATL	064C								CRCWD	ATL								xxxx
CRCWDATH	064E								CRCWDA	ATH								xxxx

凡例: — = 未実装、「0」として読み出し、リセット値は 16 進数で表記

表 4-22: クロック制御関連のレジスタマップ

レジスタ名	アド レス	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	全 リセット
RCON	0740	TRAPR	IOPUWR	SBOREN	LVREN	_	DPSLP	CM	PMSLP	EXTR	SWR	SWDTEN	WDTO	SLEEP	IDLE	BOR	POR	(Note 1)
OSCCON	0742	_	COSC2	COSC1	COSC0	_	NOSC2	NOSC1	NOSC0	CLKLOCK	_	LOCK	_	CF	SOSCDRV	SOSCEN	OSWEN	(Note 2)
CLKDIV	0744	ROI	DOZE2	DOZE1	DOZE0	DOZEN	RCDIV2	RCDIV1	RCDIV0	_	_	_	_	_	_	_	_	3140
OSCTUN	0748	_	_	_	_	_	_	_	_	_	_	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0	0000
REFOCON	074E	ROEN	_	ROSSLP	ROSEL	RODIV3	RODIV2	RODIV1	RODIV0	_	_	_	_	_	_	_	_	0000
HLVDCON	0756	HLVDEN	_	HLSIDL	_	_	_	_	_	VDIR	BGVST	IRVST	_	HLVDL3	HLVDL2	HLVDL1	HLVDL0	0000

凡例: — = 未実装、「0」として読み出し、リセット値は 16 進数で表記 **Note 1:** RCON レジスタのリセット値はリセットタイプによって異なります。

2: OSCCON レジスタのリセット値は、コンフィグレーション ヒューズとリセットのタイプによって異なります。

表 4-23: ディープスリープ関連のレジスタマップ

3X T 2U.	<i>,</i> ,		, ,	内をマノ	0 / /		_											
レジスタ 名	アドレス	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	全 リセット ⁽¹⁾
DSCON	0758	DSEN	_	_	_	_	_	_	RTCCWDIS	_	_	_		_	ULPWDIS	DSBOR	RELEASE	0000
DSWAKE	075A	_	_	_	_	_	_	_	DSINT0	DSFLT	_	_	DSWDT	DSRTCC	DSMCLR	-	DSPOR	0000
DSGPR0	075C									DSGPR0								0000
DSGPR1	075E									DSGPR1								0000

凡例: — = 未実装、「0」として読み出し、リセット値は 16 進数で表記

Note 1: ディープスリープ レジスタ DSGPR0 および DSGPR1 は、Vdd POR イベント時にのみリセットされます。

表 4-24: NVM 関連のレジスタマップ

レジスタ名	アド レス	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	全 リセット
NVMCON	0760	WR	WREN	WRERR	PGMONLY	_	-	-	-	1	ERASE	NVMOP5	NVMOP4	NVMOP3	NVMOP2	NVMOP1	NVMOP0	0000
NVMKEY	0766	_	_	-	_	_		_	_	NVMKEY							0000	

凡例: — = 未実装、「0」として読み出し、リセット値は 16 進数で表記

Note 1: 表内のリセット値は POR 時にのみ適用されます。その他のリセット状態でのリセット値は、リセット時のメモリ書き込み / 消去動作の状態によって異なります。

表 4-25: 超低消費電力復帰関連のレジスタマップ

レジス	タ名	アド レス	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	全 リセット
ULPW	CON	0768	ULPEN	_	ULPSIDL	_	_	_	_	ULPSINK	_	_	_	_	_	_	_	_	0000

凡例: — = 未実装、「0」として読み出し、リセット値は 16 進数で表記

表 4-26: PMD 関連のレジスタマップ

レジスタ名	アド レス	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	全 リセット
PMD1	0770	T5MD	T4MD	T3MD	T2MD	T1MD	_	1	1	I2C1MD	U2MD	U1MD	SPI2MD	SPI1MD	_	_	ADC1MD	0000
PMD2	0772		_	_	_	_	IC3MD	IC2MD	IC1MD	-	1	-	_	_	OC3MD	OC2MD	OC1MD	0000
PMD3	0774	1	_	I	_	_	CMPMD	RTCCMD		CRCPMD	_	-	1	-	_	I2C2MD	_	0000
PMD4	0776	_	_	-	_	_	_	_	_	ULPWUMD	_	_	EEMD	REFOMD	CTMUMD	HLVDMD	_	0000

凡例: — = 未実装、「0」として読み出し、リセット値は 16 進数で表記

4.2.5 ソフトウェア スタック

PIC24F の W15 レジスタは、通常のワーキング レジス タとしてだけでなく、ソフトウェア スタックポインタ としても使用されます。 スタックポインタは常に利用 可能な先頭空きワード位置を指し、下位アドレスから 上位アドレスに向かって移動します。 図 4-4 に示すように、スタックポインタはスタックポップでプリデクリメントし、スタックプッシュでポストインクリメントします。

CALL命令実行中のPC プッシュでは、プッシュ前にPC の MSB がゼロ拡張されるため、MSB は常にクリア状態です。

Note: 例外処理中の PC プッシュでは、プッシュ 前に SRL レジスタを PC の MSB に連結 します。

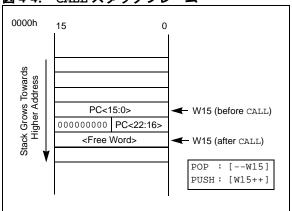
スタックポインタに関連付けられたスタックポインタ制限値 (SPLIM) レジスタは、スタックの上位側アドレス境界を設定します。SPLIM はリセット時に初期化つス境界を設定します。SPLIM はリセット時に初期化つ> と間様に、SPLIM<0> も強制的に「0」にクリアされます (全てのスタック動作はワード配列に対応する必要があるため)。W15をソースポインタまたはディスティネーション ポインタよして使用して EA を生成する場合、生成したアドレスと SPLIM の値が比較されます。スタックポインタ(W15) と SPLIM レジスタの内容が等しい場合、ツシュを実行してもスタックエラー トラップは後続のプッシュ動作で発生します。

例えばスタックが RAM 内のアドレス 0DF6 を超えた 時にスタックエラー トラップを発生させたい場合、 SPLIM の値を 0DF4 に初期化します。

同様に、スタックポインタ アドレスが 0800h を下まわると、スタックポインタ アンダーフロー (スタックエラー) トラップが発生します。これはスタックが特殊機能レジスタ (SFR) 空間に干渉する事を防ぎます。

Note: SPLIM レジスタへの書き込み直後は、W15 を使用する間接読み出しを避ける必要があります。

図 4-4: CALL スタックフレーム



4.3 プログラムメモリとデータメモリ間 のインターフェイス

PIC24Fアーキテクチャは24ビット幅のプログラム空間と16ビット幅のデータ空間を使用します。改良型ハーバードアーキテクチャを採用するため、プログラム空間にデータを格納する事も可能です。このデータを使用するには、両空間内のデータ配置を意識してアクセスする必要があります。

PIC24F アーキテクチャは、動作中にプログラム空間 へのアクセスを可能にするために、通常の動作とは異 なる下記の2つの方法を提供します。

- テーブル命令を使用して、プログラム空間の任意位置にある個々のバイトまたはワードにアクセスする
- プログラム空間の一部をデータ空間内に再マッピン グする (PSV)

テーブル命令は、プログラムメモリの小領域の読み書きに使用できます。このためテーブル命令は、時々更新が必要なデータテーブルへのアクセスに適します。この命令を使用すると、プログラムワードの全てのバイトにアクセスする事もできます。プログラム空間をデータ空間内に再マッピングする方法 (PSV) を使用すると、大きなデータブロックに読み出し専用でアクセスできます。従ってこの方法は、大きな定数テーブルの参照用に適します。この方法は、プログラムワードの下位ワード (Isw) にのみアクセスできます。

4.3.1 プログラム空間のアドレス指定

データ空間のアドレスレンジは 16 ビット、プログラム 空間のアドレスレンジは 24 ビットであるため、16 ビットのデータレジスタから 23 ビットまたは 24 ビットの プログラム アドレスを生成する方法が必要です。この方法は、プログラム空間へのアクセス方法によって異なります。

テーブル操作では、プログラム空間内の 32K ワード領域を定義するために、8 ビットのテーブル メモリページ アドレスレジスタ (TBLPAG) を使用します。これと16 ビット EA を連結する事より、24 ビットのプログラム空間アドレスを指定します。このフォーマットでは、TBLPAG の最上位ビット (MSb) を使用して、対象とするメモリ領域 (ユーザメモリ内 (TBLPAG7 = 0) またはコンフィグレーション メモリ内 (TBLPAG7 = 1)) を指定します。

プログラム空間の一部をデータ空間内に再マッピングする方法 (PSV) を使用する場合、8 ビットの PSV ページアドレス レジスタ (PSVPAG) を使用して、プログラム空間内の 16K ワードページを定義します。 EA のMSb が「1」の場合、PSVPAG と EA の下位 15 ビットを連結して 23 ビットのプログラム空間アドレスを形成します。テーブル命令を使用する方法とは異なり、再マッピング操作はユーザメモリ領域内に限定されます。

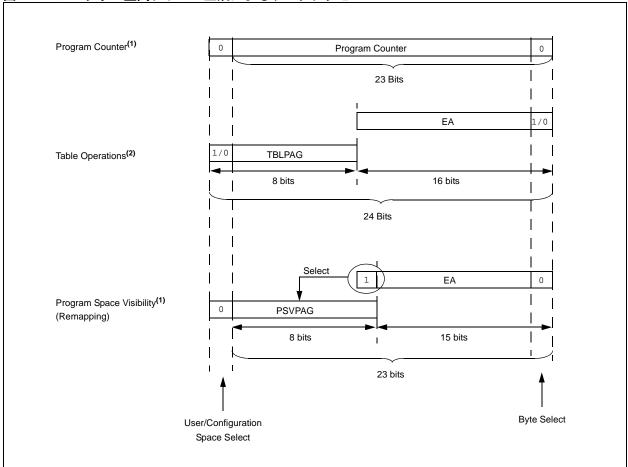
表 4-27 と図 4-5 に、テーブル操作と PSV を使用してデータ EA からプログラム EA を生成する方法を示します。ここでは、P<23:0> ビットはプログラム空間ワード、D<15:0> ビットはデータ空間ワードを参照します。

表 4-27: プログラム空間アドレスの構成

	アクセス		プログ	ラム空間で	アドレス	
アクセスタイプ	空間	<23>	<22:16>	<15>	<14:1>	<0>
命令アクセス		0		PC<22:1>	•	0
(コード実行)	ユーザ		0xx xxxx x	xxx xxx	xxxx xxx0	
	- 11°	TB	LPAG<7:0>		Data EA<15:0>	
TBLRD/TBLWT	ユーザ	0:	xxx xxxx	XXX	x xxxx xxxx x	xxx
(バイト/ワード読み書き)	-> - : : : : : : : : : : : : : : : : : :	ТВ	LPAG<7:0>		Data EA<15:0>	
	コンフィグレーション	1:	xxx xxxx	XXX	x xxxx xxxx x	xxx
PSV		0	PSVPAG<7:	0> ⁽²⁾	Data EA<14:	:0> ⁽¹⁾
(ブロック再マッピング / 読み出し)	ューザ	0	xxxx xxx	хх	xxx xxxx xxx	x xxxx

- **Note 1:** この場合、データ EA<15> は常に「1」ですが、プログラム空間アドレスの計算には使用しません。アドレスの bit 15 には PSVPAG<0> を使用します。
 - **2:** PIC24FV32KA304 ファミリの場合、PSVPAG の値は「00」(プログラムメモリにアクセス)または「FF」 (データ EEPROM にアクセス) のいずれかです。

図 4-5: プログラム空間アドレス生成によるデータアクセス



- Note 1: プログラム空間とデータ空間でデータのワード配列を維持するために、プログラム空間アドレスの LSb は常に「0」に固定されます。
 - 2: テーブル操作はワード配列に従う必要はありません。テーブル読み出しはコンフィグレーション メモリ空間にもアクセスできます。

4.3.2 テーブル命令によるプログラムメモリおよび EEPROM メモリからのデータアクセス

TBLRDL 命令とTBLWTL 命令を使用すると、データ空間を経由せずにプログラムメモリ内の任意アドレスの下位ワードを直接読み書きできます。また、データEEPROM メモリ内の任意アドレスの 1 ワードを直接読み書きする事もできます。TBLRDH 命令と TBLWTH命令は、プログラム空間内のワードの上位8ビットをデータとして読み書きできる唯一の方法です。

Note: TBLRDH 命令と TBLWTH 命令は、データ EEPROMメモリへのアクセスには使用しません。

PC は、24 ビットのプログラムワードごとに 2 ずつインクリメントします。これにより、プログラムメモリアドレスをデータ空間アドレスに直接マッピングできます。従ってプログラムメモリは、アドレスレンジが同じで隣り合う 2 つの 16 ビット幅ワードアドレス空間と見なす事ができます。TBLRDL 命令と TBLWTL 命令は、下位データワードを格納した空間にアクセスし、TBLRDH 命令と TBLWTH 命令は上位データバイトを格納した空間にアクセスします。

バイトサイズまたはワードサイズ (16 ビット)のデータをプログラム空間から (またはプログラム空間へ)移動するために、下記の2つのテーブル命令を使用できます。どちらの命令もバイト操作とワード操作が可能です。

- TBLRDL (テーブル LOW 読み出し): ワードモードでは、プログラム空間の下位ワード (P<15:0>) をデータアドレス (D<15:0>) にマッピングします。
 - バイトモードでは、プログラム空間の下位ワードの上位バイトまたは下位バイトのいずれかをデータアドレスの下位バイトにマッピングします。バイト選択が「1」であれば上位バイト、「0」であれば下位バイトを選択します。
- 2. TBLRDH (テーブル HIGH 読み出し): ワードモードでは、プログラムアドレスの上位 ワード全体(P<23:16>)をデータアドレスにマッピングします。D<15:8>(ファントムバイト)は常に「0」である事に注意してください。

バイトモードでは、上位のプログラムワードの上位バイトまたは下位バイトをデータアドレスの D<7:0> にマッピングします。上位の「ファントム」バイトを選択した場合 (バイト選択 = 1)、データは常に「0」です。

プログラム空間アドレスに個々のバイトまたはワードを書き込む場合も、2つのテーブル命令 TBLWTH と TBLWTL を、上記と同様の方法で使用します。これらの操作の詳細は 5.0「フラッシュ プログラムメモリ」で説明します。

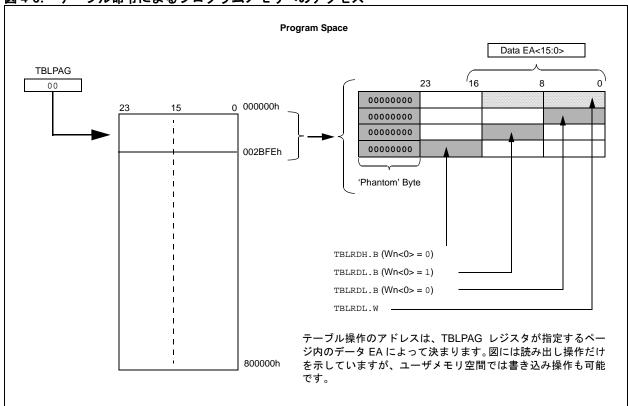
全てのテーブル操作において、テーブルページ レジスタ (TBLPAG) がプログラムメモリ空間内のアクセス先領域を指定します。TBLPAG は、ユーザ空間とコンフィグレーション空間を含むデバイスのプログラムメモリ空間全体をカバーします。TBLPAG<7>=0 の場

合、テーブルページはユーザメモリ空間内です。 TBLPAG<7>=1の場合、ページはコンフィグレーション空間内です。

Note:

コンフィグレーション メモリ空間に対しては、テーブル読み出し操作だけが可能です (デバイス ID 等の実装領域に対してのみ可能)。コンフィグレーション メモリ空間に対してテーブル書き込み操作はできません。

図 4-6: テーブル命令によるプログラムメモリへのアクセス



4.3.3 PSV によるプログラムメモリからの データ読み出し

データ空間の上位 32K バイトは、必要に応じてプログラム空間の 16K ワードページ (PIC24FV16KA3XX) または 32K ワードページ (PIC24FV32KA3XX) にマッピングできます。これにより、特別な命令 (TBLRDL/H)を使用しなくても、保存されている定数データにデータ空間から透過的にアクセスできます。

データ空間経由のプログラム空間へのアクセスは、データ空間 EA の MSb を「1」にセットし、CPU 制御 (CORCON<2>) レジスタの PSV ビットをセットして PSV を有効化した場合に発生します。データ空間に マッピングするプログラムメモリ空間の位置は、PSV ページアドレス (PSVPAG) レジスタで指定します。この 8 ビットレジスタは、プログラム空間内に配置された 256 ページの 16K ワードの中から任意の 1ページを指定します。つまり PSVPAG はプログラムメモリ アドレスの上位 8 ビットとして機能し、EA の 15 ビットが下位ビットとして機能します。

プログラムメモリ ワードごとに PC を 2 ずつインクリメントする事により、データ空間アドレスの下位 15 ビットを対応するプログラム空間アドレスの下位 15 ビットに直接マッピングします。

この領域からのデータ読み出しには2回のプログラム メモリ フェッチが必要であるため、命令実行には1サ イクルが追加されます。 8000 h 以上の各データ空間アドレスは、対応するプログラムメモリ アドレスに直接マッピングされますが (図 4-7 参照)、24 ビット プログラムワードの下位 16 ビットだけがデータ格納用に使用されます。データとして使用するプログラム空間アドレスの上位 8 ビットには、「1111 1111」または「0000 0000」を書き込んで、強制的に NOP とする必要があります。これにより、上位 8 ビットのコードが誤って実行されるという問題を回避できます。

Note: PSV アクセスは、テーブル読み / 書き時 に、一時的に無効化されます。

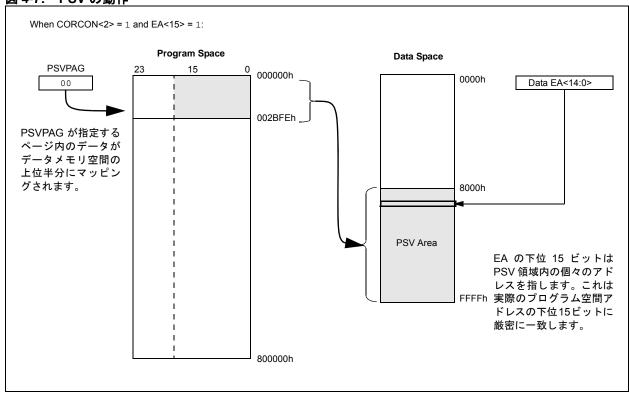
PSV を使用する動作を REPEAT ループの外で実行する場合、MOV 命令と MOV.D 命令の実行には、所定の実行時間に加えて 1 命令サイクルを要します。その他の全ての命令の実行には、所定の実行時間に加えて 2 命令サイクルを要します。

PSV を使用する動作を REPEAT ループの中で実行する場合、下記の条件では所定の命令実行時間に加えて 2 命令サイクルを要します。

- 最初の繰り返し実行時
- 最後の繰り返し実行時
- 割り込みによりループを終了する前の実行時
- 割り込みが完了してループを再開する際の実行時

REPEAT ループ内の上記以外の繰り返し実行では、 PSVを使用してデータにアクセスする命令を1サイク ルで実行できます。

図 4-7: PSV の動作



NOTE:

5.0 フラッシュ プログラムメモリ

Note:

本書は、PIC24 デバイスの中の特定製品グループを対象とし、それらの機能の概要説明を目的としています。従って本書は、包括的な参照資料の提供を意図したものではありません。フラッシュ プログラミングの詳細は「PIC24F ファミリリファレンス マニュアル」のセクション4.「プログラムメモリ」(DS39715)を参照してください。

PIC24FV32KA304 は、アプリケーション コードを保存して実行するためのフラッシュ プログラムメモリ を内蔵しています。このメモリは、1.8 V 以上の VDD で動作している時に読み書きおよび消去可能です。

フラッシュメモリは下記の3種類の方法でプログラミングできます。

- インサーキット シリアル プログラミング (ICSP™)
- ・ 実行中自己書き込み (RTSP)
- 拡張インサーキット シリアル プログラミング(拡張 ICSP)

ICSP を利用すると、アプリケーション回路に搭載済みの PIC24FV32KA304 をシリアルプログラミングできます。これは、プログラミング クロック用 (PGECx) とプログラミング データ用 (PGEDx) の2本のラインに加えて、電源 (VDD)、グランド (VSS)、マスタクリア/プログラムモード エントリ電圧 (MCLR/VPP) 用の3本のラインを使用して、簡単に行えます。この機能を利用すると、未プログラミングのデバイスを実装したボードを製造し、製品の出荷直前にマイクロコントローラをプログラミングできます。また、最新バージョンのファームウェアやカスタマイズしたファームウェアのプログラミングも可能です。

実行中自己書き込み (RTSP) の実行には、TBLRD (テーブル読み出し)命令と TBLWT (テーブル書き込み)命令を使用します。RTSP を使用すると、32 命令のブロック (96 byte) 内のプログラムメモリ データを一度に書き込む事ができ、32/64/128 命令のブロック (96/192/384 byte) 内のプログラムメモリを一度に消去できます。

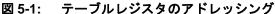
消去するブロックのサイズは、NVMOP<1:0>(NVMCON<1:0>) ビットで指定します。

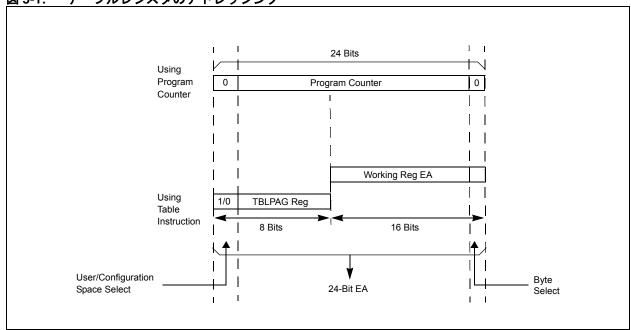
5.1 テーブル命令とフラッシュ プログラ ミング

使用する方法に関係なく、フラッシュメモリのプログラミングには常にテーブル読み書き命令を使用します。このため、デバイスが通常の動作モード中であれば、データメモリからプログラムメモリ空間に直接読み書きアクセスできます。プログラムメモリ内の24ビットターゲットアドレスは、TBLPAG<7:0>ビットと、テーブル命令内で指定されたワーキングレジスタからの実効アドレス(EA)によって形成されます(図5-1参照)。

プログラムメモリのビット <15:0> の読み書きには、 TBLRDL 命令と TBLWTL 命令を使用します。TBLRDL 命令と TBLWTL 命令は、ワードモードとバイトモードのいずれでもプログラムメモリにアクセスできます。

プログラムメモリのビット <23:16> の読み書きには、 TBLRDH 命令と TBLWTH 命令を使用します。TBLRDH と TBLWTH も、ワードモードとバイトモードのいずれで もプログラムメモリにアクセスできます。





5.2 RTSP の動作

PIC24Fのフラッシュ プログラムメモリ アレイは、複数の行 (1 行 = 32 命令、または 96 バイト) によって構成されます。RTSP を使用すると、1/2/4 行 (32/64/128 命令) のブロックを一度に消去でき、1 行 (32 命令) のブロックを一度にプログラミングできます。1 ワードずつ書き込む事もできます。

1/2/4行(96/192/384 byte)のブロック消去と1行(96 byte) のブロック書き込みは、プログラムメモリの先頭からエッジアラインされます。

TBLWT 命令を使用してプログラムメモリにデータを書き込む場合、データは直接メモリには書き込まれません。テーブル書き込み命令は、書き込みシーケンスが実行されるまで、書き込みデータをホールドラッチに保存します。

TBLWT 命令を何回実行しても、書き込みは正常に実行されます。しかし、メモリの1行を完全に書き込むには、TBLWT 命令を32回実行する必要があります。

RTSP プログラミングの基本シーケンスでは、テーブルポインタを設定してから、一連の TBLWT 命令を実行してバッファにデータを書き込みます。フラッシュメモリへのプログラミングは、NVMCON レジスタ内の制御ビットをセットする事によって行います。

プログラムメモリへの書き込み動作を実行する前に、ホールドレジスタにはデータを自由な順番で何度でも書き込めます。後続の書き込みは、以前の書き込みデータを上書きします。

Note: プログラムメモリ内のデータを消去せずに、同じ位置に複数回書き込む事は推奨しません。

全てのテーブル書き込み動作は、バッファにだけ書き込むため、1 ワード書き込み (2 命令サイクル) です。各行のプログラミングには、1 プログラミング サイクルが必要です。

5.3 拡張インサーキット シリアル プログラミング

拡張 ICSP は、プログラミング処理を管理するために、 内蔵ブートローダ(プログラム エグゼクティブとも呼ばれる)を使用します。プログラム エグゼクティブは、 SPI データフレーム フォーマットを使用して、プログラムメモリを消去/プログラミング/検証できます。拡張 ICSP の詳細は、各デバイスのプログラミング仕様を参照してください。

5.4 制御レジスタ

プログラム フラッシュメモリの読み書きには 2 つの SFR (NVMCON \geq NVMKEY) を使用します。

NVMCON レジスタ (レジスタ 5-1 参照) は、消去する ブロック、プログラミングするメモリのタイプ、プログラミング サイクルの開始を制御します。

NVMKEY は、書き込み保護用に使用する書き込み専用 レジスタです。プログラミングまたは消去シーケンス を開始するために、ユーザは NVMKEY レジスタに 55h と AAh を連続して書き込む必要があります。詳細は 5.5「プログラミング動作」を参照してください。

5.5 プログラミング動作

RTSP モードでの内蔵フラッシュのプログラミングまたは消去には、完全なプログラミング シーケンスが必要です。実行中のプログラミングまたは消去動作が終了するまで、プロセッサはストール (待機) します。プログラミングまたは消去動作を開始するには、WRビット (NVMCON<15>) をセットします。WRビットは動作終了時に自動的にクリアされます。

レジスタ 5-1: NVMCON: フラッシュメモリ制御レジスタ

R/SO-0, HC	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
WR	WREN	WRERR	PGMONLY ⁽⁴⁾	_	_	_	_
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
_	ERASE	NVMOP5 ⁽¹⁾	NVMOP4 ⁽¹⁾	NVMOP3 ⁽¹⁾	NVMOP2 ⁽¹⁾	NVMOP1 ⁽¹⁾	NVMOP0 ⁽¹⁾
bit 7							bit 0

凡例:SO = セットのみ可能ビットHC = ハードウェア クリア可能ビット-n = POR 時の値1 = ビットをセットR = 読み出し可能ビットW = 書き込み可能ビット0 = ビットをクリアx = ビットは未知U = 未実装ビット、「0」として読み出し

bit 15 WR: 書き込み制御ビット

1 = フラッシュメモリのプログラミングまたは消去動作を開始する(動作は内部タイマによって制御され、動作完了時にハードウェアがこのビットをクリアします)

0 = プログラミングまたは消去動作は完了した(現在非アクティブ)

bit 14 WREN: 書き込みイネーブルビット

1= フラッシュ プログラミング / 消去動作を有効にする

0 = フラッシュ プログラミング / 消去動作を禁止する

bit 13 WRERR: 書き込みシーケンス エラーフラグ ビット

1 = 不正なプログラミングまたは消去シーケンスの試行または中断が発生した(このビットはWRビットのセット試行時に自動的にセットされます)

0 = プログラミングまたは消去動作は正常に完了した

bit 12 **PGMONLY:** 消去なし書き込みイネーブルビット (4)

bit 11-7 **未実装:**「0」として読み出し

bit 6 **ERASE**: 消去 / プログラミング イネーブルビット

1 = 次の WR コマンドで、NVMOP<5:0> が指定する消去動作を実行する

0 = 次の WR コマンドで、NVMOP<5:0> が指定するプログラミング動作を実行する

bit 5-0 **NVMOP<5:0>:** プログラミング動作コマンドバイト ビット ⁽¹⁾

消去動作 (ERASE ビットが「1」の時):

1010xx = ブートブロックの全体を消去する (コード保護されたブートブロックを含む)(2)

1001xx =メモリの全体を消去する (ブートブロック、コンフィグレーション ブロック、汎用ブロック を含む) $^{(2)}$

011010 = フラッシュメモリの 4 行を消去する (3)

011001 = フラッシュメモリの 2 行を消去する (3)

011000 = フラッシュメモリの 1 行を消去する ⁽³⁾

0101xx = コンフィグレーション ブロックの全体を消去する (コード保護ビットを除く)

0100xx = データ EEPROM⁽⁴⁾ の全体を消去する

0011xx = 汎用メモリブロックの全体を消去する

0001xx = フラッシュメモリの 1 行を書き込む (ERASE ビットが「0」の時)⁽³⁾

Note 1: 上記以外の NVMOP<5:0> ビット設定は全て未実装です。

2: ICSP™ モードでのみ利用できます。デバイスのプログラミング仕様を参照してください。

3: テーブルポインタ内のアドレスによって消去する行が決まります。

4: このビットは、データ EEPROM へのアクセス時にのみ使用されます。

5.5.1 フラッシュ プログラムメモリのプロ グラミング アルゴリズム

フラッシュ プログラムメモリは、1 度に 1 行ずつ消去 しながら書き込む事ができます。一般的な手順は下記 の通りです。

- プログラムメモリの 1 行 (32 命令)を読み込んで、 データ RAM に格納する
- RAM内のプログラムデータを必要なデータに書き 換える
- 3. 下記の手順で1行を消去する(例 5-1 参照)
 - a) NVMOP ビット (NVMCON<5:0>) を「011000」 (1 行消去) に設定し、ERASE (NVMCON<6>) ビットと WREN (NVMCON<14>) ビットをセッ トする
 - b) TBLPAG レジスタとワーキング レジスタに、 消去するブロックの先頭アドレスを書き込む
 - c) NVMKEY に 55h を書き込む
 - d) NVMKEY に AAh を書き込む
 - e) WR ビット (NVMCON<15>) をセットすると 消去サイクルが始まり、完了するまで CPU が ストールし、 完了時に WR ビットが自動的に

クリアされる

- 4. 先頭の 32 命令をデータ RAM からプログラムメモリ バッファに書き込む (例 5-1 参照)
- 5. 下記の手順でフラッシュメモリにプログラム ブロックを書き込む
 - a) NVMOP ビットを「011000」(1 行プログラ ミング) に設定し、ERASE ビットをクリア して WREN ビットをセットする
 - b) NVMKEY に 55h を書き込む
 - c) NVMKEY に AAh を書き込む
 - d) WR ビットをセットするとフラッシュメモリ への書き込みサイクルが始まり、完了するま で CPU がストールし、完了すると WR ビッ トが自動的にクリアされる

不測の動作を回避するため、消去またはプログラミング動作を開始する前に、必ず NVMKEY の書き込み開始シーケンスを実行する必要があります。プログラミングコマンド実行後、ユーザアプリケーションはプログラミングが完了するまで待機する必要があります。プログラミングシーケンス開始直後に、2回のNOP命令が必要です(例 5-5参照)。

例 5-1: プログラムメモリ行の消去 – アセンブリコード

```
; Set up NVMCON for row erase operation
           #0x4058, W0
   MOV
   MOV
           W0, NVMCON
                                      ; Initialize NVMCON
; Init pointer to row to be ERASED
           #tblpage(PROG_ADDR), W0
   MOV
   MOV
           WO, TBLPAG
                                     ; Initialize PM Page Boundary SFR
           #tbloffset(PROG_ADDR), W0 ; Initialize in-page EA[15:0] pointer
   MOV
   TBLWTL W0, [W0]
                                     ; Set base address of erase block
           #5
                                      ; Block all interrupts
                                       for next 5 instructions
   MOV
           #0x55, W0
   MOV
           WO, NVMKEY
                                     ; Write the 55 key
   MOV
           #0xAA, W1
   MOV
           W1, NVMKEY
                                     ; Write the AA key
   BSET
           NVMCON, #WR
                                     ; Start the erase sequence
   MOP
                                      ; Insert two NOPs after the erase
   NOP
                                      ; command is asserted
```

例 5-2: プログラムメモリ行の消去 – C 言語コード

```
// C example using MPLAB C30
   int __attribute__ ((space(auto_psv))) progAddr = &progAddr;// Global variable located in Pgm Memory
   unsigned int offset;
//Set up pointer to the first memory location to be written
   TBLPAG = __builtin_tblpage(&progAddr);
                                                           // Initialize PM Page Boundary SFR
   offset = &progAddr & 0xFFFF;
                                                           // Initialize lower word of address
   __builtin_tblwtl(offset, 0x0000);
                                                           // Set base address of erase block
                                                           // with dummy latch write
   NVMCON = 0x4058;
                                                           // Initialize NVMCON
   asm("DISI #5");
                                                           // Block all interrupts for next 5
                                                           // instructions
    __builtin_write_NVM();
                                                           // C30 function to perform unlock
                                                       // sequence and set WR
```

例 5-3: 書き込みパッファへのデータ書き込み – アセンブリコード

```
; Set up NVMCON for row programming operations
   MOV
          #0x4004, W0
                                 ; Initialize NVMCON
   MOV
          WO, NVMCON
; Set up a pointer to the first program memory location to be written
; program memory selected, and writes enabled
          #0x0000, W0
   MOV
          W0, TBLPAG
                                  ; Initialize PM Page Boundary SFR
         #0x6000, W0
   MOV
                                  ; An example program memory address
; Perform the TBLWT instructions to write the latches
; 0th_program_word
       #LOW_WORD_0, W2
   MOV
         #HIGH_BYTE_0, W3
   TBLWTL W2, [W0]
                                 ; Write PM low word into program latch
   TBLWTH W3, [W0++]
                                  ; Write PM high byte into program latch
; 1st_program_word
        #LOW_WORD_1, W2
   MOV
   MOV
          #HIGH_BYTE_1, W3
   TBLWTL W2, [W0]
                                 ; Write PM low word into program latch
   TBLWTH W3, [W0++]
                                  ; Write PM high byte into program latch
; 2nd_program_word
        #LOW_WORD_2, W2
         #HIGH_BYTE_2, W3
   MOV
                                 ; Write PM low word into program latch
   TBLWTL W2, [W0]
   TBLWTH W3, [W0++]
                                  ; Write PM high byte into program latch
; 32nd_program_word
   MOV #LOW_WORD_31, W2
         #HIGH_BYTE_31, W3
   TBLWTL W2, [W0]
                                 ; Write PM low word into program latch
   TBLWTH W3, [W0]
                                  ; Write PM high byte into program latch
```

例 5-4: 書き込みパッファへのデータ書き込み – C 言語コード

```
// C example using MPLAB C30
  #define NUM_INSTRUCTION_PER_ROW 64
  int __attribute__ ((space(auto_psv))) progAddr = &progAddr;// Global variable located in Pgm
Memory
  unsigned int offset;
  unsigned int i;
  unsigned int progData[2*NUM_INSTRUCTION_PER_ROW]; // Buffer of data to write
  //Set up NVMCON for row programming
  NVMCON = 0x4001;
                                                           // Initialize NVMCON
  //Set up pointer to the first memory location to be written
  TBLPAG = __builtin_tblpage(&progAddr);
                                                          // Initialize PM Page Boundary SFR
  offset = &progAddr & 0xFFFF;
                                                           // Initialize lower word of address
  //Perform TBLWT instructions to write necessary number of latches
  for(i=0; i < 2*NUM_INSTRUCTION_PER_ROW; i++)</pre>
      __builtin_tblwtl(offset, progData[i++]);
                                                          // Write to address low word
      __builtin_tblwth(offset, progData[i]);
                                                          // Write to upper byte
      offset = offset + 2;
                                                           // Increment address
  }
```

例 5-5: プログラミング シーケンスの開始 – アセンブリコード

```
DISI
       #5
                             ; Block all interrupts
                              for next 5 instructions
MOV
       #0x55, W0
MOV
       WO, NVMKEY
                             ; Write the 55 key
MOV
       #0xAA, W1
MOV
       W1, NVMKEY
                            ; Write the AA kev
BSET
       NVMCON, #WR
                            ; Start the erase sequence
NOP
                            ; 2 NOPs required after setting WR
NOP
                            ; Wait for the sequence to be completed
BTSC NVMCON, #15
BRA
       $-2
```

例 5-6: プログラミング シーケンスの開始 – C 言語コード

例 5-7: フラッシュ プログラムメモリの 1 ワード プログラミング

```
; Setup a pointer to data Program Memory
MOV #tblpage(PROG_ADDR), W0
MOV W0, TBLPAG
                                 ;Initialize PM Page Boundary SFR
MOV #tbloffset(PROG_ADDR), W0
                                 ;Initialize a register with program memory address
MOV #LOW_WORD_N, W2
MOV #HIGH_BYTE_N, W3
TBLWTLW2, [W0]
                                 ; Write PM low word into program latch
TBLWTHW3, [W0++]
                                 ; Write PM high byte into program latch
; Setup NVMCON for programming one word to data Program Memory
MOV #0x4003, W0
MOV W0, NVMCON
                                 ; Set NVMOP bits to 0011
DISI#5
                                 ; Disable interrupts while the KEY sequence is written
MOV #0x55, W0
                                 ; Write the key sequence
MOV W0, NVMKEY
MOV #0xAA, W0
MOV W0, NVMKEY
BSETNVMCON, #WR
                                 ; Start the write cycle
```

NOTE:

6.0 データ EEPROM メモリ

Note: 本書は、PIC24 デバイスの中の特定製品 グループを対象とし、それらの機能の概 要説明を目的としています。従って本書 は、包括的な参照資料の提供を意図した

ものではありません。データ EEPROM の詳細は「PIC24F ファミリ リファレンスマニュアル」のセクション 5.「データ EEPROM」(DS39720) を参照してください。

データ EEPROM メモリは、プログラムメモリおよび 揮発性データ RAM とは別の不揮発性メモリ (NVM) で す。データ EEPROM メモリはプログラムメモリと同 じフラッシュ技術に基づき、より長期の記憶保持性と 高い書き換え耐性が得られるように最適化されていま す。

データ EEPROM はユーザ プログラムメモリ空間の最上位 (プログラムメモリーアドレス 7FFE00h ~ 7FFFFFh) に配置されます。PIC24FV32KA304 が実装するデータ EEPROM のサイズは 256 ワードです。

データ EEPROM は 16 ビット幅メモリを構成します。 通常動作時の VDD レンジ全域で、各ワードへ直接アクセスして読み書きできます。

フラッシュ プログラムメモリとは異なり、データ EEPROM のプログラミングまたは消去動作中に通常 のプログラム実行は停止しません。

データ EEPROM のプログラミング動作の制御には、 下記の3つの NVM 制御レジスタを使用します。

- NVMCON: 不揮発性メモリ制御レジスタ
- NVMKEY: 不揮発性メモリ キーレジスタ
- NVMADR: 不揮発性メモリ アドレスレジスタ

6.1 NVMCON レジスタ

NVMCON レジスタ(レジスタ 6-1 参照)は、データ EEPROM のプログラミング / 消去動作用の主要制御レジスタです。このレジスタの上位バイトは、プログラミングまたは消去サイクルを開始するための制御ビットと、動作が正常に完了したかどうかを示すフラグビットを格納します。NVMCOM レジスタの下位バイトは、実行する NVM 動作のタイプを設定します。

6.2 NVMKEY レジスタ

NVMKEY レジスタは、データ EEPROM 領域の偶発的な書き込み / 消去を回避するために使用する書き込み専用レジスタです。

プログラミングまたは消去シーケンスを開始する前に、下記の2つの命令を下記の順番で実行する必要があります。

- 1. NVMKEY に 55h を書き込む
- 2. NVMKEY に AAh を書き込む

NVMCON レジスタへの書き込みは、このシーケンスを実行した後の1命令サイクルで1回だけ実行できます。多くの場合、NVMCON レジスタ内の WR ビットをセットするだけで、書き込みまたは消去サイクルを開始できます。このロック解除シーケンスを実行する前に、割り込みを無効にする必要があります。

MPLAB[®] C30 C コンパイラは、ロック解除シーケンス を 実 行 す る 定 義 済 み ラ イ ブ ラ リ マ ク ロ 関 数 (builtin_write_NVM) を提供します。例 6-1 に、インライン アセンブリを使用してロック解除シーケンスを実行する方法を示します。

例 6-1: データ EEPROM のロック解除シーケンス

レジスタ 6-1: NVMCON: 不揮発性メモリ制御レジスタ

R/S-0, HC	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
WR	WREN	WRERR	PGMONLY	_	_	_	_
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
_	ERASE	NVMOP5	NVMOP4	NVMOP3	NVMOP2	NVMOP1	NVMOP0
bit 7							bit 0

凡例: HC=ハードウェア クリア可能 U=未実装ビット、「0」として読み出し

ビット

R=読み出し可能ビット S=セット可能ビット W=書き込み可能ビット

-n = POR 時の値 1=ビットをセット 0 = ビットをクリア x=ビットは未知

bit 15 WR: 書き込み制御ビット(プログラミングまたは消去)

> 1 = データ EEPROM の書き込みまたは消去サイクルを開始する(このビットはソフトウェアでセッ トできますがクリアはできません)

0= 書き込みサイクルは完了した(ハードウェアが自動的にこのビットをクリアします)

bit 14 WREN: 書き込みイネーブルビット (消去またはプログラミング)

1= プログラミングまたは消去動作を有効にする

0 = プログラミングおよび消去動作を禁止する(書き込み/消去動作完了時にデバイスはこのビット をクリアします)

bit 13 WRERR: フラッシュエラー フラグビット

1 = 書き込み動作は途中で終了した(プログラミング動作中にMCLRまたはWDTリセットが発生した)

0 = 書き込み動作は正常に完了した

bit 12 PGMONLY: 消去なし書き込みイネーブルビット

1 = 書き込み先アドレスの内容を消去せずに直接書き込みを実行する

0 = 書き込み前に書き込み先アドレスの内容を自動的に消去する

書き込み先アドレスの内容を消去すると、書き込み動作が自動的に開始されます。

bit 11-7 未実装:「0」として読み出し

bit 6 ERASE: 消去動作選択ビット

1 = WR がセットされた時に消去を実行する

0 = WR がセットされた時に書き込みを実行する

bit 5-0 NVMOP<5:0>: プログラミング動作コマンドバイト ビット

<u>消去動作 (ERASE ビットが「1」の時)</u> 011010 = **8** ワードを消去する

011001 = 4 ワードを消去する

011000 = 1 ワードを消去する

0100xx = データ EEPROM の全体を消去する

書き込み動作 (ERASE ビットが「0」の時):

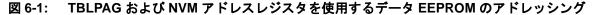
001xx = 1 ワードを書き込む

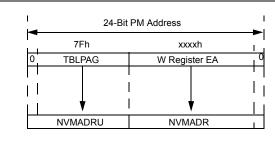
6.3 NVM アドレスレジスタ

フラッシュ プログラムメモリの場合と同様に、NVM アドレスレジスタ (NVMADRU と NVMADR) は、データEEPROM の書き込み / 消去動作を行う行またはワードの 24 ビット実効アドレス (EA) を形成します。NVMADRU レジスタは EA の上位 8 ビットを格納し、NVMADR レジスタは EA の下位 16 ビットを格納します。これらのレジスタは特殊機能レジスタ (SFR) 空間内には割り当てられず、直前に実行されたテーブル書き込み命令の EA<23:0> を直接キャプチャして消去するデータ EEPROM 行を選択します。プログラミング / 消去動作用に形成されるプログラムメモリEA を図 6-1に示します。

プログラムメモリの場合と同様に、NVMADR の最下位ビット (LSb) はゼロに限定されます (偶数アドレスに限定)。これは、データ EEPROM 空間内のアドレスはプログラムメモリ幅の下位ワードだけを格納するためです。従って最上位にファントム バイトを持つ上位ワードは存在しません。このため、データ EEPROMアドレスの最下位ビット (LSb) は常に「0」です。

全てのアドレスはユーザプログラム空間内に配置されるため、NVMADRU の最上位ビット (MSb) も常に「0」です。





6.4 データ EEPROM の動作

プログラムメモリの場合と同様に、データ EEPROM ブロックへのアクセスにもテーブル読み書き動作を使用します。データ EEPROM 動作の場合、メモリは 16 ビット幅しか持たないため、TBLWTH および TBLRDH命令は不要です。データ EEPROMでは下記のプログラミング動作を行えます。

- 1、4、8 ワードの消去
- データ EEPROM 全体のバルク消去
- 1ワード書き込み
- ・ 1 ワード読み出し

Note 1: プログラミング / 消去動作の実行中に EEPROMに対して読み出しを試みると、 予期せぬ結果が生じます。

2: C30 C コンパイラは、テーブル読み書き動作の自動実行、テーブルポインタと書き込みバッファの管理、ロック解除とメモリ書き込みシーケンスの開始を行うライブラリマクロ関数を提供します。これにより、アプリケーションごとにアセンブラマクロや実行速度が重視されるC言語ルーチンを作成する手間を省けます。

後述のサンプルコードでは、これらのライブラリマクロ関数を使用します。C30 コンパイラのライブラリを使用しないユーザのために、各プロセスの概要も記載します。

6.4.1 データ EEPROM の消去

データ EEPROM は、全体を一度に消去するか、3 種類のサイズ (1/4/8 ワード) で部分的に消去できます。消去するワード数は、NVMOP<1:0> ビット (NVMCON<1:0>)で指定します。データ EEPROM の一部だけを消去する場合、下記のシーケンスを実行する必要があります。

- NVMCON レジスタで消去するワード数(1/4/8のいずれか)を設定する
- 2. TBLPAG および WREG レジスタに、消去する EEPROM アドレスを書き込む
- 3. NVMIF ステータスビットをクリアし、(必要に 応じて)NVM割り込みを有効にする
- 4. キーシーケンスを NVMKEY に書き込む
- 5. WR ビットをセットして書き込みサイクルを開始する
- 6. WR ビットをポーリングするか、(NVMIF をセットした場合) NVM 割り込みを待機する

標準的な消去シーケンスを例 6-2 に示します。このサンプルコードは、1 ワード消去の方法を示しています。4 または 8 ワード消去も、同様の方法で行えます。このサンプルコードは、C ライブラリマクロ関数を使用してテーブルポインタ (builtin_tbloffset) と消去ページポインタ (builtin_tblwtl)を管理しています。メモリロック解除シーケンス (builtin_write_NVM) は、消去動作を開始するために WR ビットのセットも行い、動作終了時に制御を戻します。

例 6-2: 1 ワード消去

```
int __attribute__ ((space(eedata))) eeData = 0x1234; // Global variable located in EEPROM
   unsigned int offset;
   // Set up NVMCON to erase one word of data EEPROM
   NVMCON = 0x4058;
   // Set up a pointer to the EEPROM location to be erased
   TBLPAG = __builtin_tblpage(&eeData); // Initialize EE Data page pointer
   offset = __builtin_tbloffset(&eeData);
                                                   // Initizlize lower word of address
    __builtin_tblwtl(offset, 0);
                                                    // Write EEPROM data to write latch
   asm volatile ("disi #5");
                                                    // Disable Interrupts For 5 Instructions
                                      // Issue Unlock Sequence & Start Write Cycle
   builtin write NVM();
  while(NVMCONbits.WR=1);
                                      // Optional:Poll WR bit to wait for
                                      // write sequence to complete
```

6.4.1.1 データ EEPROM のバルク消去

データ EEPROM 全体の消去 (バルク消去)は、データ EEPROM の全体を対象とするため、アドレスレジス タを設定する必要はありません。下記の手順でバルク 消去を実行できます。

- 1. NVMCON レジスタでバルク消去モードを有効 にする
- NVMIF ステータスビットをクリアし、(必要に応じて)NVM 割り込みを有効にする
- 3. キーシーケンスを NVMKEY に書き込む
- 4. WR ビットをセットして書き込みサイクルを開始する
- 5. WR ビットをポーリングするか、NVM 割り込み (NVMIF のセット) を待機する

標準的なバルク消去シーケンスを例6-3に示します。

6.4.2 1ワード書き込み

データ EEPROM に 1 ワードを書き込む場合、下記の シーケンスを実行する必要があります。

- PGMONLY ビット (NVMCON<12>) が「1」に セットされている場合、データ EEPROM 内の 1 ワードを消去する(前セクション参照)
- 2. データワードをデータ EEPROM ラッチに書き 込む
- 下記の手順でデータワードを EEPROM に書き 込む
 - NVMCON レジスタで EEPROM の 1 ワード 書き込みを有効にする (NVMCON<5:0> = 11000)
 - NVMIF ステータスビットをクリアし、(必要に応じて)NVM 割り込みを有効にする
 - キーシーケンスを NVMKEY に書き込む
 - WRビットをセットして書き込みサイクルを 開始する
 - WR ビットをポーリングするか、(NVMIF を セットした場合) NVM 割り込みを待機する
 - シーケンスの完了を検出するために、NVMIF がセットされるまで待機する

標準的な 1 ワード書き込みシーケンスを例 6-4 に示します。

例 6-3: データ EEPROM のバルク消去

```
// Set up NVMCON to bulk erase the data EEPROM
NVMCON = 0x4050;

// Disable Interrupts For 5 Instructions
asm volatile ("disi #5");

// Issue Unlock Sequence and Start Erase Cycle
__builtin_write_NVM();
```

例 6-4: データ EEPROM への1ワード書き込み

```
int __attribute__ ((space(eedata))) eeData = 0x1234;// Global variable located in EEPROM
int newData;
                                         // New data to write to EEPROM
unsigned int offset;
// Set up NVMCON to erase one word of data EEPROM
NVMCON = 0x4004;
// Set up a pointer to the EEPROM location to be erased
// Initizlize lower word of address
offset = __builtin_tbloffset(&eeData);
__builtin_tblwtl(offset, newData);
                                         // Write EEPROM data to write latch
asm volatile ("disi #5");
                                         // Disable Interrupts For 5 Instructions
 __builtin_write_NVM();
                                         // Issue Unlock Sequence & Start Write Cycle
while(NVMCONbits.WR=1);
                                         // Optional:Poll WR bit to wait for
                                         // write sequence to complete
```

6.4.3 データ EEPROM の読み出し

データ EEPROM からのワード読み出しには、テーブル読み出し命令を使用します。EEPROM配列は16ビット幅しか持たないため、TBLRDL 命令だけが必要です。読み出し動作では、TBLPAGおよびWREGにEEPROM読み出し位置のアドレスを書き込んでからTBLRDL命令を実行します。

C30 コンパイラ ライブラリが提供するテーブルポインタ管理マクロ (builtin_tblpage と builtin_tbloffset) とテーブル読み出しマクロ(builtin_tblrdl)を使用した標準的な読み出しシーケンスを例 6-5 に示します。

PSV を使用してデータ EEPROM 内のアドレスを読み 出す事もできます。

例 6-5: TBLRD コマンドを使用するデータ EEPROM の読み出し

7.0 リセット

Note: 本書は、PIC24 デバイスの中の特定製品 グループを対象とし、それらの機能の概 要説明を目的としています。従って本書 は、包括的な参照資料の提供を意図した ものではありません。リセットの詳細は「PIC24Fファミリリファレンス マニュアル」のセクション 40.「ブラウンアウト リセット等のリセット」(DS39728) を参照

リセット モジュールは全てのリ<u>セット要</u>因を集約し、 デバイスのマスタリセット信号 (SYSRST) を制御しま す。下記はデバイスリセット要因の一覧です。

• POR: パワーオン リセット

• MCLR: ピンリセット

• SWR: RESET 命令

• WDTR: ウォッチドッグ タイマ リセット

してください。

• BOR: ブラウンアウト リセット

・ 低消費電力 BOR/ ディープスリープ BOR

• TRAPR: トラップ衝突リセット

• IOPUWR: 不正オペコード リセット

• UWR: 未初期化ワーキング レジスタ リセット

リセット モジュールの概略ブロック図を図 7-1 に示します。

全てのリセット要因は、発生時に SYSRST 信号をアクティブにします。この際、CPU と周辺モジュールに関連する各種レジスタは、既知の「リセット状態」に強制的に設定されます。ほとんどのレジスタはリセットに影響されません。すなわち、パワーオン リセット(POR)以外のリセットでは状態は変化せず、パワーオン リセット(POR)時の状態は未知です。

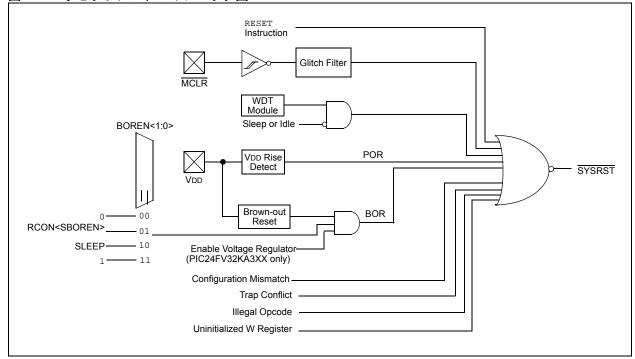
Note: レジスタのリセット状態の詳細については、ファミリ リファレンス マニュアルの各周辺モジュールまたはCPUのセクションを参照してください。

全てのタイプのデバイスリセットは、発生時にRCONレジスタ内の対応するステータスビットをセットし、発生したリセットのタイプを示します(レジスタ 7-1参照)。PORは、BORおよびPORビット(RCON<1:0>)以外の全てのビットをクリアします。ユーザアプリケーションは、コード実行中いつでも任意のビットをセット/クリアできます。RCONビットはステータスビットとしてのみ機能します。ソフトウェアで特定のリセットステータスビットをセットしても、デバイスリセットは発生しません。

RCON レジスタは、ウォッチドッグ タイマ (WDT) とデバイスの省電力状態に関連するビットも格納します。これらのビットの機能については、本書の別のセクションで説明します。

次回のデバイスリセットに備えるため に、RCON レジスタ内のステータスビットは読み出し後にクリアする必要があります

図 7-1: リセットシステムのブロック図



Note:

レジスタ 7-1: RCON: リセット制御レジスタ ⁽¹⁾

R/W-0, HS	R/W-0, HS	R/W-0	R/W-0	U-0	R/C-0, HS	R/W-0	R/W-0
TRAPR	IOPUWR	SBOREN	LVREN ⁽³⁾	_	DPSLP	СМ	PMSLP
bit 15							bit 8

R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-1, HS	R/W-1, HS
EXTR	SWR	SWDTEN ⁽²⁾	WDTO	SLEEP	IDLE	BOR	POR
bit 7							bit 0

凡例:	C = クリア可能ビット	HS = ハードウェア セット可能ビット
R = 読み出し可能ビット	W = 書き込み可能ビット	U = 未実装ビット、「0」として読み出し
-n = POR 時の値	1 = ビットをセット	0 = ビットをクリア x = ビットは未知

bit 15 TRAPR: トラップリセット フラグビット

1=トラップ衝突リセットが発生した

0 = トラップ衝突リセットは発生していない

bit 14 IOPUWR: 不正オペコード / 未初期化ワーキング レジスタ アクセス リセット フラグビット

1 = 不正オペコードを検出した、またはアドレスモードが不正、または未初期化ワーキング レジスタ

をアドレスポインタとして使用したためにリセットが発生した

0 = 上記のリセットは発生していない

bit 13 SBOREN: BOR ソフトウェア イネーブル / ディセーブル ビット

1 = BOR はソフトウェアで有効化されている

0 = BOR はソフトウェアで無効化されている

bit 12 LVREN: 低電圧スリープモード (3)

1 = スリープ中の電源電圧は低電圧レギュレータ (LVREG) だけから供給する

0 = スリープ中の電源電圧はメイン電圧レギュレータ (HVREG) から供給する ⁽³⁾

bit 11 未実装:「0」として読み出し

bit 10 DPSLP: ディープスリープ モード フラグビット

1 = ディープスリープが発生した

0 = ディープスリープは発生していない

bit 9 CM: コンフィグレーション ワード不整合リセット フラグビット

1=コンフィグレーション ワード不整合リセットが発生した

0=コンフィグレーション ワード不整合リセットは発生していない

bit 8 PMSLP: スリープ時プログラムメモリ電源ビット

1=スリープ中もプログラムメモリ バイアス電圧を維持する

0=スリープモード時および電圧レギュレータのスタンバイモード時に、プログラムメモリバイアス電

圧の供給を停止する

bit 7 EXTR: 外部リセット (MCLR) ピンビット

1=マスタクリア(ピン)リセットが発生した

0 = マスタクリア(ピン)リセットは発生していない

bit 6 SWR: ソフトウェア リセット(命令)フラグビット

1 = RESET 命令が実行された

0 = RESET 命令は実行されていない

bit 5

1 = WDT を有効にする

0 = WDT を無効にする

Note 1: 全てのリセット ステータスビットはソフトウェアでセット / クリアできます。これらのステータスビッ トをソフトウェアでセットしても、デバイスリセットは発生しません。

2: FWDTEN コンフィグレーション ビットが「1」(未プログラム状態)の場合、SWDTEN ビットの設定に 関係なく WDT は常に有効です。

3: PIC24FV32KA3xx でのみ使用します。PIC24F32KA3XX では使用しません。

レジスタ 7-1: RCON: リセット制御レジスタ (続き)⁽¹⁾

bit 4 WDTO: ウォッチドッグ タイマ タイムアウト フラグビット

1 = WDT タイムアウトが発生した

0 = WDT タイムアウトは発生していない

bit 3 SLEEP: スリープからの復帰フラグビット

1 = デバイスはスリープモードから復帰した

0 = デバイスはスリープモードから復帰したのではない

bit 2 IDLE: アイドルからの復帰フラグビット

1 = デバイスはアイドルモードから復帰した

0 = デバイスはアイドルモードから復帰したのではない

bit 1 BOR: ブラウンアウト リセット フラグビット

1 = ブラウンアウト リセットが発生した (BOR ビットは PORA 時にもセットされます)

0 = ブラウンアウト リセットは発生していない

bit 0 **POR**: パワーオン リセット フラグビット

1 = パワーオン リセットが発生した

0 = パワーオン リセットは発生していない

Note 1: 全てのリセット ステータスビットはソフトウェアでセット / クリアできます。これらのステータスビットをソフトウェアでセットしても、デバイスリセットは発生しません。

2: FWDTEN コンフィグレーション ビットが「1」(未プログラム状態)の場合、SWDTEN ビットの設定に関係なく WDT は常に有効です。

3: PIC24FV32KA3xx でのみ使用します。PIC24F32KA3XX では使用しません。

表 7-1: リセット フラグビットの動作

フラグビット	フラグをセットするイベント	フラグをクリアする イベント
TRAPR (RCON<15>)	トラップ衝突イベント	POR
IOPUWR (RCON<14>)	不正オペコード / 未初期化ワーキング レジスタ アクセス	POR
CM (RCON<9>)	コンフィグレーション不整合リセット	POR
EXTR (RCON<7>)	MCLR リセット	POR
SWR (RCON<6>)	RESET 命令	POR
WDTO (RCON<4>)	WDT タイムアウト	PWRSAV 命令、POR
SLEEP (RCON<3>)	PWRSAV #SLEEP 命令	POR
IDLE (RCON<2>)	PWRSAV #IDLE 命令	POR
BOR (RCON<1>)	POR, BOR	_
POR (RCON<0>)	POR	_
DPSLP (RCON<10>)	PWRSAV #SLEEP 命令 (DSCON <dsen> をセット)</dsen>	POR

Note: ユーザ ソフトウェアは全てのリセット フラグビットをセット / クリアできます。

7.1 リセット時クロック源の選択

クロック切り換えが有効な場合、デバイスリセット時のシステムクロック源の選択は表 7-2 に従います。クロック切り換えが無効な場合、システムクロック源は常にオシレータ コンフィグレーション ビットの設定に従って選択されます。詳細は 9.0「オシレータのコンフィグレーション」を参照してください。

表 7-2: 各種リセット時のクロック源の選択 (クロック切り換え有効時)

リセットタイプ	クロック源を指定するビット
POR	FNOSC コンフィグレーションビット
BOR	(FNOSC<10:8>)
MCLR	COSC 制御ビット
WDTO	(OSCCON<14:12>)
SWR	

7.2 デバイスリセット時間

各種タイプのデバイスリセットにおけるリセット時間 を表 7-3 に示します。システムリセット信号 SYSRST は、POR および PWRT 遅延時間経過後にリリースさ れる事に注意してください。

デバイスが実際にコード実行を開始するまでの遅延時間は、システムオシレータ遅延(オシレータ起動タイマ (OST) と PLL ロック時間を含む) にも影響されます。OST および PLL ロック遅延は、SYSRST 遅延と並列に発生します。

FSCM 遅延は、SYSRST 信号がリリースされてから FSCMがシステムクロック源の監視を開始するまでの 時間を決定します。

表 7-3: 各種デバイスリセットのリセット遅延時間

リセットタイプ	クロック源	SYSRST 遅延	システム クロック 遅延	Note
POR ⁽⁶⁾	EC	TPOR + TPWRT	_	1, 2
	FRC, FRCDIV	TPOR + TPWRT	TFRC	1, 2, 3
	LPRC	TPOR + TPWRT	TLPRC	1, 2, 3
	ECPLL	TPOR + TPWRT	TLOCK	1, 2, 4
	FRCPLL	TPOR + TPWRT	TFRC + TLOCK	1, 2, 3, 4
	XT, HS, SOSC	TPOR+ TPWRT	Tost	1, 2, 5
	XTPLL, HSPLL	TPOR + TPWRT	Tost + Tlock	1, 2, 4, 5
BOR	EC	TPWRT	_	2
	FRC, FRCDIV	TPWRT	TFRC	2, 3
	LPRC	TPWRT	TLPRC	2, 3
	ECPLL	TPWRT	TLOCK	2, 4
	FRCPLL	TPWRT	TFRC + TLOCK	2, 3, 4
	XT, HS, SOSC	TPWRT	Tost	2, 5
	XTPLL, HSPLL	Tpwrt	TFRC + TLOCK	2, 3, 4
その他	任意のクロック	_	_	なし

Note 1: TPOR: パワーオン リセット遅延

2: TPWRT = 起動タイマ有効時に 64 ms(公称値)、起動タイマ無効時にゼロ

3: TFRC、TLPRC: RC オシレータ起動時間

4: TLOCK: PLL ロック時間

5: TOST: オシレータ起動タイマ (OST): クロックをシステムに供給する前に、10 ビットカウンタが 1024 オシレータ サイクルをカウントします。

6: 2 段階起動が有効な場合、プライマリ オシレータの選択に関係なく、デバイスは FRC を使用して起動します。この場合 FRC 起動時間を適用します。

Note: 動作周波数とタイミングの詳細な仕様は、29.0「電気的特性」を参照してください。

7.2.1 起動の遅いオシレータの POR

オシレータ起動回路と、これに関連する遅延タイマは、パワーアップ時に発生するデバイスリセット遅延とはリンクしていません。一部の水晶振動子回路(特に低周波数水晶振動子)では、比較的長い起動時間が必要です。このため SYSRST 信号のリリース後に、下記状態が1つまたは複数発生する可能性があります。

- オシレータ回路が発振を開始していない
- オシレータ起動タイマ (OST) が終了していない (水晶振動子を使用する場合)
- PLL をロックできていない (PLL を使用する場合)

デバイスは、クロックがシステムに供給されるまでコード実行を開始しません。従って、リセット遅延時間を検討する場合、オシレータと PLL の起動遅延も考慮に入れる必要があります。

7.2.2 フェイルセーフ クロック監視機能 (FSCM) とデバイスリセット

FSCM を有効にすると、SYSRST がリリースされた時に、システムクロック源の監視を開始します。この時、有効なクロック源が存在しないと、デバイスはクロック源を自動的に FRC オシレータに切り換えます。ユーザはトラップ サービスルーチン (TSR) 内で、クロック源を必要な水晶振動子に切り換える事ができます。

7.3 特殊機能レジスタのリセット状態

PIC24Fの CPU と周辺モジュールに関連する特殊機能 レジスタ (SFR) の大部分は、デバイスリセット時に既 定値にリセットされます。SFR は周辺モジュールまたは CPU 機能ごとに分類されており、それらのリセット値は本書の関連セクションに記載しています。

後述の 4 つのレジスタを除き、各 SFR のリセット値は、どのタイプのリセットでも同じです。リセット制御レジスタ (RCON) のリセット値は、デバイスリセットのタイプによって異なります。オシレータ制御レジスタ (OSCCON) のリセット値は、リセットのタイプと、フラッシュ コンフィグレーション ワード (FOSCSEL) の FNOSC ビットの設定値によって異なります (表 7-2 参照)。RCFGCAL および NVMCON レジスタは、POR にのみ影響されます。

7.4 ディープスリープ BOR (DSBOR)

ディープスリープBORは、デバイスのディープスリープモード時に使用する非常に低消費電力のBOR回路です。消費電流が低いため、精度は変動する可能性があります。

DSBOR トリップポイントは約 $2.0\,V$ です。DSLPBOR (FDS<6>) を「1」にセットすると、DSBOR が有効化されます。DSLPBOR は、VDD が POR しきい値より低下した時にデバイスを確実にリセットするためにPORを補強します。

7.5 ブラウンアウト リセット (BOR)

PIC24FV32KA304 ファミリデバイスの内蔵 BOR 回路は、各種のコンフィグレーションと省電カオプションを備えます。BOR の制御には、BORV<1:0>および BOREN<1:0>コンフィグレーション ビット(FPOR<6:5,1:0>)を使用します。BOR は表 7-3 に示す合計 4 つのコンフィグレーションを備えます。

BOR しきい値は BORV<1:0> ビットで設定します。BOR が有効な場合 (BOREN<1:0> が「00」以外の場合)、VDD が設定されたしきい値を下回るとデバイスがリセットされます。デバイスは、VDD がしきい値を超えるまで BOR 状態を維持します。

パワーアップタイマが有効な場合、VDDがしきい値を超えた時にタイマが起動し、追加の遅延時間 (TPWRT)が経過するまでデバイスをリセット状態に維持します。パワーアップタイマの動作中に VDD がしきい値より低下した場合、デバイスは BOR に戻り、パワーアップタイマは初期化されます。VDD がしきい値を超えると、パワーアップタイマは追加の遅延時間を開始します。

BOR とパワーアップ タイマ (PWRT) は別々に設定します。BOR リセットを有効にしても、PWRT は自動的に有効化されません。

7.5.1 ソフトウェア制御による BOR

BOREN<1:0> = 01 の場合、BOR はソフトウェアで有効化 / 無効化できます。これには制御ビット SBOREN (RCON<13>)を使用します。SBOREN をセットすると、前述の BOR 機能が有効化されます。SBOREN をクリアすると、BOR は完全に無効化されます。SBOREN ビットはこのモードでのみ機能します。これ以外では「0」として読み出されます。

BOR をソフトウェアで制御する事により、デバイスを再プログラミングする事なく、環境条件に合わせて柔軟にアプリケーションを最適化できます。また、BORが消費する電流増加分も最適化できます。通常 BORの消費電流は微少ですが、低消費電力アプリケーションでは影響を無視できない場合があります。

Note: BOR をソフトウェアで制御する場合でも、BOR リセット電圧レベルはBORV<1:0> コンフィグレーション ビットで設定します。この設定をソフトウェアで変更する事はできません。

7.5.2 BOR の検出

BOR を有効にした場合、BOR だけでなく POR イベント発生時にも、BOR ビット (RCON<1>)が「1」にセットされます。このため、BOR の状態だけを読み出してBOR イベントが発生したかどうかを判断するのは困難です。POR と BOR の状態を同時に確認する事により、より確実に判断できます。この方法は、POR イベント直後にソフトウェアでPOR ビットとBOR ビットを「0」にリセットする事を前提とします。この場合、BOR ビットが「1」で POR ビットが「0」であれば、BOR イベントが発生したと確実に判定できます。

Note: デバイスがディープスリープ モードから 復帰した場合も、POR ビットと BOR ビッ トの両方がセットされます。

7.5.3 スリープモード時の BOR の無効化

BOREN<1:0>=10 の場合、BOR はハードウェアで制御され、前述のように動作します。この設定では、デバイスがスリープモードに移行すると、BOR は自動的に無効化されます。デバイスがスリープモードから他の動作モードへ戻る際に、BOR は自動的に有効化されます。

このモードを使用すると、デバイスが BOR 保護を最も必要とするコード実行時に、アプリケーションをブラウンアウト状態からの回復でき、スリープモード時には BOR による微小な電流消費を節約できます。

Note: BOR 電圧レベルはデバイスのタイプに よって異なります (PIC24FV32KA3XX と PIC24F32KA3XX では BOR レベルが異な ります)。BOR 電圧レベルは 29.0「電気的 特性」を参照してください。

8.0 割り込みコントローラ

Note: 本書は、PIC24 デバイスの中の特定製品 グループを対象とし、それらの機能の概 要説明を目的としています。従って本書 は、包括的な参照資料の提供を意図した ものではありません。割り込みコント ローラの詳細は「PIC24F ファミリリファ レンス マニュアル」の**セクション 8.「割**

PIC24Fの割り込みコントローラ モジュールは、多数の周辺モジュールの割り込み要求信号を、CPUへの1つの割り込み要求信号へ集約します。この割り込みコントローラは以下の特長を備えます。

り込み」(DS39707)を参照してください。

- 最大8つのプロセッサ例外とソフトウェア トラップ
- ・ ユーザ選択可能な7段階の優先度
- 最大 118 ベクタの割り込みベクタテーブル (IVT)
- 割り込みまたは例外要因の各々に一意のベクタ割り 当て
- 同一ユーザ指定優先度を持つ要因間の優先順位は自 然順序優先度に従う
- デバッグをサポートする代替割り込みベクタテーブル (AIVT)
- 一定の割り込みエントリとリターン レイテンシ

8.1 割り込みベクタ (IVT) テーブル

図 8-1 に割り込みベクタテーブル (IVT)を示します。IVT はプログラムメモリ内に配置され、先頭位置は000004hです。IVT は 126 ベクタを格納し、その内の8ベクタがノンマスカブルトラップベクタ用で、残りの最大 118 ベクタを割り込み要因用として使用できます。一般的に、各割り込み要因には固有のベクタが割り当てられます。各割り込みベクタは 24 ビット幅のアドレスを格納します。各割り込みベクタ位置に書き込まれた値は、対応する割り込みサービスルーチン(ISR)の開始アドレスを指します。

各割り込みベクタの優先度は、ベクタテーブル内の並び順で決まる自然順序優先度に従います。他の条件が同じであれば、アドレス値が低いほど自然順序優先度が高くなります。例えば、ベクタ 0 に関連付けられた割り込みは、他のベクタアドレスに関連付けられたどの割り込みよりも高い優先度を持ちます。

PIC24FV32KA304 ファミリデバイスは、ノンマスカブルトラップと一意の割り込みを実装しています。それらの要約を表 8-1 と表 8-2 に示します。

8.1.1 代替割り込みベクタテーブル (AIVT)

代替割り込みベクタテーブル (AIVT) はIVT の直後に配置されます(図 8-1 参照)。AIVT へのアクセスは、ALTIVT制御ビット (INTCON2<15>) により有効化されます。この ALTIVT ビットをセットすると、全ての割り込みと例外処理は既定値のベクタではなく代替ベクタを使用します。代替ベクタは既定値ベクタと同様の順番で配置されます。

AIVT は、割り込みベクタを書き換えずにアプリケーションとサポート環境間の切り換えを可能にする事によって、エミュレーションとデバッグをサポートします。この機能を使用すると、実行中にアプリケーションを切り換えながら複数の異なるソフトウェア アルゴリズムを評価できます。AIVT を使用しない場合、IVTで使用するアドレスと同じアドレスを AIVT に書き込む必要があります。

8.2 リセットシーケンス

割り込みコントローラはリセット処理には関与しないため、デバイスリセットは厳密には例外処理ではありません。PIC24Fは、リセット時にレジスタをクリアしてプログラムカウンタ (PC)を強制的にゼロに設定した後に、000000hからプログラムの実行を開始します。ユーザは、このリセットアドレスに GOTO 命令を書き込む事により、プログラムの実行を適切な起動ルーチンへリダイレクトできます。

Note: IVT と AIVT の未実装または未使用ベクタ 位置には、RESET 命令を実行する既定値 割り込みハンドラルーチンのアドレスを

書き込む必要があります。

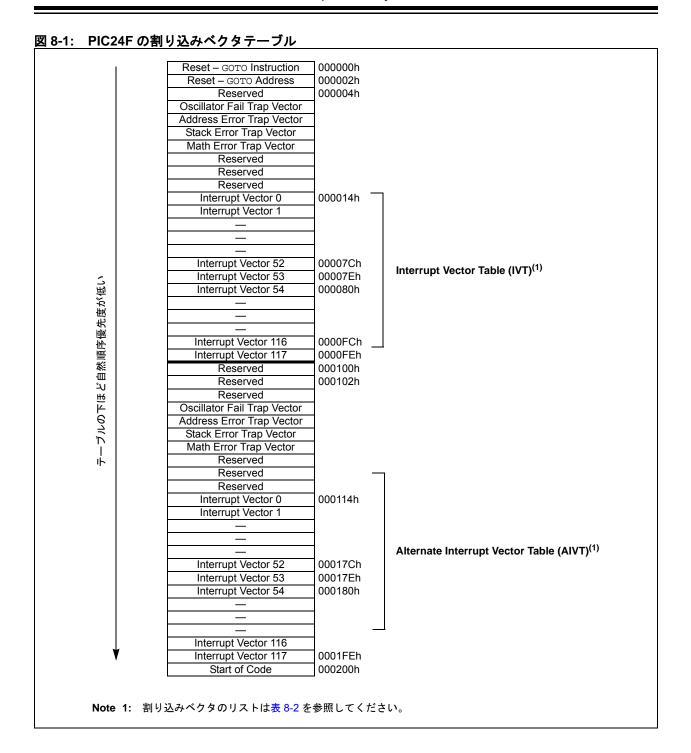


表 8-1: 割り込みベクタの詳細

ベクタ番号	IVT アドレス	AIVT アドレス	トラップ要因
0	000004h	000104h	予約済み
1	000006h	000106h	オシレータ障害
2	000008h	000108h	アドレスエラー
3	00000Ah	00010Ah	スタックエラー
4	00000Ch	00010Ch	算術エラー
5	00000Eh	00010Eh	予約済み
6	000010h	000110h	予約済み
7	000012h	000112h	予約済み

表 8-2: 実装されている割り込みベクタ

朝けつ か 悪口	ベクタ番号	N/T 77 (* 1. 7	AIVT		割り込み関連ビ	ット
割り込み要因	ハクラ田ち	IVT アドレス	アドレス	フラグ	イネーブル	優先度
ADC1 変換完了	13	00002Eh	00012Eh	IFS0<13>	IEC0<13>	IPC3<6:4>
コンパレータ イベント	18	000038h	000138h	IFS1<2>	IEC1<2>	IPC4<10:8>
CRC ジェネレータ	67	00009Ah	00019Ah	IFS4<3>	IEC4<3>	IPC16<14:12>
СТМU	77	0000AEh	0001AEh	IFS4<13>	IEC4<13>	IPC19<6:4>
外部割り込み 0	0	000014h	000114h	IFS0<0>	IEC0<0>	IPC0<2:0>
外部割り込み 1	20	00003Ch	00013Ch	IFS1<4>	IEC1<4>	IPC5<2:0>
外部割り込み 2	29	00004Eh	00014Eh	IFS1<13>	IEC1<13>	IPC7<6:4>
I2C1 マスタイベント	17	000036h	000136h	IFS1<1>	IEC1<1>	IPC4<6:4>
I2C1 スレーブイベント	16	000034h	000134h	IFS1<0>	IEC1<0>	IPC4<2:0>
I2C2 マスタイベント	50	000078h	000178h	IFS3<2>	IEC3<2>	IPC12<10:8>
I2C2 スレーブイベント	49	000076h	000176h	IFS3<1>	IEC3<1>	IPC12<6:4>
入力キャプチャ 1	1	000016h	000116h	IFS0<1>	IEC0<1>	IPC0<6:4>
入力キャプチャ 2	5	00001Eh	00011Eh	IFS0<5>	IEC0<5>	IPC1<6:4>
入力キャプチャ 3	37	00005Eh	00015Eh	IFS2<5>	IEC2<5>	IPC9<6:4>
状態変化通知	19	00003Ah	00013Ah	IFS1<3>	IEC1<3>	IPC4<14:12>
HLVD (HIGH/LOW 電圧検出)	72	0000A4h	0001A4h	IFS4<8>	IEC4<8>	IPC17<2:0>
NVM – NVM 書き込み完了	15	000032h	000132h	IFS0<15>	IEC0<15>	IPC3<14:12>
出カコンペア 1	2	000018h	000118h	IFS0<2>	IEC0<2>	IPC0<10:8>
出カコンペア 2	6	000020h	000120h	IFS0<6>	IEC0<6>	IPC1<10:8>
出カコンペア 3	25	000046h	000146h	IFS1<9>	IEC1<9>	IPC6<6:4>
リアルタイム クロック / カレンダ	62	000090h	000190h	IFS3<14>	IEC3<14>	IPC15<10:8>
SPI1 エラー	9	000026h	000126h	IFS0<9>	IEC0<9>	IPC2<6:4>
SPI1 イベント	10	000028h	000128h	IFS0<10>	IEC0<10>	IPC2<10:8>
SPI2 エラー	32	000054h	000154h	IFS2<0>	IEC2<2>	IPC8<2:0>
SPI2 イベント	33	000056h	000156h	IFS2<1>	IEC2<1>	IPC8<6:4>
Timer1	3	00001Ah	00011Ah	IFS0<3>	IEC0<3>	IPC0<14:12>
Timer2	7	000022h	000122h	IFS0<7>	IEC0<7>	IPC1<14:12>
Timer3	8	000024h	000124h	IFS0<8>	IEC0<8>	IPC2<2:0>
Timer4	27	00004Ah	00014Ah	IFS1<11>	IEC1<11>	IPC6<14:12>
Timer5	28	00004Ch	00015Ch	IFS1<12>	IEC1<12>	IPC7<2:0>
UART1 エラー	65	000096h	000196h	IFS4<1>	IEC4<1>	IPC16<6:4>
UART1 受信	11	00002Ah	00012Ah	IFS0<11>	IEC0<11>	IPC2<14:12>
UARTx 送信	12	00002Ch	00012Ch	IFS0<12>	IEC0<12>	IPC3<2:0>
UART2 エラー	66	000098h	000198h	IFS4<2>	IEC4<2>	IPC16<10:8>
UART2 受信	30	000050h	000150h	IFS1<14>	IEC1<14>	IPC7<10:8>
UARTx 送信	31	000052h	000152h	IFS1<15>	IEC1<15>	IPC7<14:12>
超低消費電力復帰	80	0000B4h	0001B4h	IFS5<0>	IEC5<0>	IPC20<2:0>

8.3 割り込み制御/ステータス レジスタ

PIC24FV32KA304 ファミリのデバイスは、割り込みコントローラ用に下記の22個のレジスタを使用します。

- INTCON1
- INTCON2
- IFS0、IFS1、IFS3、IFS4
- IEC0、IEC1、IEC3、IEC4
- IPC0 ~ IPC5、IPC7、IPC15 ~ IPC19
- INTTREG

グローバルな割り込み制御機能は、INTCON1 と INTCON2 で制御します。INTCON1 は、割り込みネスティング ディセーブル (NSTDIS) ビットと、プロセッサ トラップ要因の制御フラグとステータスフラグを格納します。INTCON2 レジスタは、外部割り込み要求信号の挙動と AIV テーブルの使用を制御します。

IFSx レジスタは全ての割り込み要求フラグを格納します。各割り込み要因はそれぞれ 1 つのステータスビットを持ちます。このビットは、周辺モジュールまたは外部信号によってセットされ、ソフトウェアによってクリアされます。

IECx レジスタは全ての割り込みイネーブルビットを格納します。これらの制御ビットを使用して、周辺モジュールまたは外部信号からの割り込みを個別に有効化できます。

IPCx レジスタは各割り込み要因の優先度を設定します。各ユーザ割り込み要因には7段階の優先度を割り当てる事ができます。

INTTREG レジスタは、対応する割り込みベクタ番号と新規 CPU 割り込み優先度を格納します。これらはINTTREG レジスタのベクタ番号 (VECNUM<6:0>) および割り込みレベル (ILR<3:0>) ビットフィールドにラッチされます。新規割り込み優先度は、保留中割り込みの優先度です。

各割り込み要因は、表 8-2 に従って IFSx、IECx、IPCx レジスタに割り当てられます。例えば、ベクタ番号と自然順序優先度が 0 の INTO (外部割り込み 0) の場合、ステータスビット INTOIF は IFSO<0>、イネーブルビット INTOIE は IECO<0>、優先度ビット INTOIP<2:0> は IPCO の最下位位置 (IPCO<2:0>) に格納されている事がわかります。

2 つの CPU 制御レジスタは特に割り込み制御ハードウェアの一部ではありませんが、割り込み機能を制御するビットを格納しています。ALU ステータス レジスタ (SR) は、IPL<2:0> ビット (SR<7:5>) を格納します。これらのビットは、現在の CPU 割り込み優先度を示します。ユーザは、IPL ビットに書き込む事によって、現在の CPU 優先度を変更できます。

CORCON レジスタは IPL3 ビットを格納します。この ビットも、IPL<2:0> と連結して現在の CPU 優先度を 示します。IPL3 は読み出し専用ビットであるため、 ユーザ ソフトウェアでトラップイベントをマスクする事はできません。

全ての割り込みレジスタの説明を、以降のレジスタ8-1からレジスタ8-3に記載します。

レジスタ 8-1: SR: ALU ステータス レジスタ

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R-0, HSC
_	_	_	_	_	_	_	DC ⁽¹⁾
bit 15							bit 8

R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC
IPL2 ^(2,3)	IPL1 ^(2,3)	IPL0 ^(2,3)	RA ⁽¹⁾	N ⁽¹⁾	OV ⁽¹⁾	Z ⁽¹⁾	C ⁽¹⁾
bit 7							bit 0

凡例: HSC = ハードウェア セット / クリア可能ビット

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「O」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-9 **未実装:**「0」として読み出し

bit 7-5 IPL<2:0>: CPU 割り込み優先度ステータスビット (2,3)

111 = CPU 割り込み優先度は 7 (15): ユーザ割り込みは無効

110 = CPU 割り込み優先度 6 (14)

101 = CPU 割り込み優先度 5 (13)

100 = CPU 割り込み優先度 4 (12)

011 = CPU 割り込み優先度 3 (11)

010 = CPU 割り込み優先度 2 (10)

001 = CPU 割り込み優先度 1 (9)

000 = CPU 割り込み優先度 0 (8)

Note 1: これらのビットに関してはレジスタ 3-1 を参照してください。これらのビットは割り込み制御機能専用ではありません。

2: IPL ビットは IPL3 ビット (CORCON<3>) と連結して CPU 割り込み優先度を形成します。カッコ内の値は IPL3 = 1 の時の割り込み優先度です。

3: NSTDIS (INTCON1<15>) = 1 の場合、IPL ステータスビットは読み出し専用です。

Note: bit 8 と bit 4 ~ 0 に関しては 3.0「CPU」を参照してください。

レジスタ 8-2: CORCON: CPU 制御レジスタ

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
_	_	_	_	_	_	_	_
bit 15							bit 8

U-0	U-0	U-0	U-0	R/C-0, HSC	R/W-0	U-0	U-0
_	_	_	_	IPL3 ⁽²⁾	PSV ⁽¹⁾	_	
bit 7							bit 0

凡例:C = クリア可能ビットHSC = ハードウェア セット / クリア可能ビットR = 読み出し可能ビットW = 書き込み可能ビットU = 未実装ビット、「0」として読み出し-n = POR 時の値1 = ビットをセット0 = ビットをクリアx = ビットは未知

bit 15-4 **未実装:**「0」として読み出し

bit 3 **IPL3**: CPU 割り込み優先度ステータスビット (2)

1 = CPU 優先度は 8 以上 0 = CPU 優先度は 7 以下 **未実装:**「0」として読み出し

Note 1: このビットに関してはレジスタ 3-2 を参照してください。このビットは割り込み制御機能専用ではあり

ません。

bit 1-0

2: IPL3 ビットは IPL<2:0> ビット (SR<7:5>) と連結して CPU 割り込み優先度を形成します。

Note: bit 2 については 3.0「CPU」に記載しています。

レジスタ 8-3: INTCON1: 割り込み制御レジスタ 1

R/W-0	U-0						
NSTDIS	_	_	_	_	_	_	_
bit 15							bit 8

U-0	U-0	U-0	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	U-0
_	_	_	MATHERR	ADDRERR	STKERR	OSCFAIL	_
bit 7							bit 0

凡例: HS = ハードウェア セット可能ビット

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15 NSTDIS: 割り込みネスティング ディセーブル ビット

1 = 割り込みネスティングを無効にする 0 = 割り込みネスティングを有効にする

bit 14-5 **未実装:**「0」として読み出し

bit 4 MATHERR: 算術エラートラップ ステータスビット

1 = オーバーフロー トラップが発生した 0 = オーバーフロー トラップは発生していない

bit 3 ADDRERR: アドレスエラー トラップ ステータスビット

1 = アドレスエラー トラップが発生した

0=アドレスエラー トラップは発生していない

bit 2 STKERR: スタックエラー トラップ ステータスビット

1 = スタックエラー トラップが発生した

0 = スタックエラー トラップは発生していない

bit 1 OSCFAIL: オシレータ障害トラップ ステータスビット

1=オシレータ障害トラップが発生した

0=オシレータ障害トラップは発生していない

bit 0 **未実装:**「0」として読み出し

レジスタ 8-4: INTCON2: 割り込み制御レジスタ 2

R/W-0	R-0, HSC	U-0	U-0	U-0	U-0	U-0	U-0
ALTIVT	DISI	_	_	_	_	_	_
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
_	_	_	_	_	INT2EP	INT1EP	INT0EP
bit 7							bit 0

凡例: HSC = ハードウェア セット / クリア可能ビット

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15 ALTIVT: 代替割り込みベクタテーブル イネーブルビット

1 = 代替割り込みベクタテーブルを使用する 0 = 標準(既定値)ベクタテーブルを使用する

bit 14 DISI: DISI 命令ステータスビット

1 = DISI 命令はアクティブ 0 = DISI 命令は非アクティブ

bit 13-3 **未実装:**「0」として読み出し

bit 2 INT2EP: 外部割り込み 2 エッジ検出極性選択ビット

1 = 立ち下がりエッジで割り込む 0 = 立ち上がりエッジで割り込む

bit 1 INT1EP: 外部割り込み 1 エッジ検出極性選択ビット

1 = 立ち下がりエッジで割り込む 0 = 立ち上がりエッジで割り込む

bit 0 **INTOEP**: 外部割り込み 0 エッジ検出極性選択ビット

1 = 立ち下がりエッジで割り込む 0 = 立ち上がりエッジで割り込む

レジスタ 8-5: IFSO: 割り込みフラグステータス レジスタ 0

R/W-0, HS	U-0	R/W-0, HS					
NVMIF	_	AD1IF	U1TXIF	U1RXIF	SPI1IF	SPF1IF	T3IF
bit 15							bit 8

R/W-0, HS	R/W-0, HS	R/W-0, HS	U-0	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS
T2IF	OC2IF	IC2IF	_	T1IF	OC1IF	IC1IF	INT0IF
bit 7							bit 0

凡例: HS = ハードウェア セット可能ビット

 R = 読み出し可能ビット
 W = 書き込み可能ビット
 U = 未実装ビット、「0」として読み出し

 -n = POR 時の値
 1 = ビットをセット
 0 = ビットをクリア
 x = ビットは未知

bit 15 NVMIF: NVM 割り込みフラグ ステータスビット

1 = 割り込み要求が発生した

0 = 割り込み要求は発生していない

bit 14 **未実装:**「0」として読み出し

bit 13 AD1IF: A/D 変換完了割り込みフラグ ステータスビット

1 = 割り込み要求が発生した 0 = 割り込み要求は発生していない

bit 12 **U1TXIF**: UART1 送信割り込みフラグ ステータスビット

1 = 割り込み要求が発生した

0 = 割り込み要求は発生していない

bit 11 **U1RXIF:** UART1 受信割り込みフラグ ステータスビット

1 = 割り込み要求が発生した

0 = 割り込み要求は発生していない

bit 10 SPI1IF: SPI1 イベント割り込みフラグ ステータスビット

1 = 割り込み要求が発生した

0 = 割り込み要求は発生していない

bit 9 SPF1IF: SPI1 フォルト割り込みフラグ ステータスビット

1 = 割り込み要求が発生した

0 = 割り込み要求は発生していない

bit 8 T3IF: Timer3 割り込みフラグ ステータスビット

1 = 割り込み要求が発生した

0 = 割り込み要求は発生していない

bit 7 T2IF: Timer2 割り込みフラグ ステータスビット

1 = 割り込み要求が発生した

0 = 割り込み要求は発生していない

bit 6 OC2IF: 出力コンペア チャンネル 2 割り込みフラグ ステータスビット

1 = 割り込み要求が発生した

0 = 割り込み要求は発生していない

bit 5 IC2IF: 入力キャプチャ チャンネル 2 割り込みフラグ ステータスビット

1 = 割り込み要求が発生した

0 = 割り込み要求は発生していない

bit 4 **未実装:**「0」として読み出し

bit 3 T1IF: Timer1 割り込みフラグ ステータスビット

1 = 割り込み要求が発生した

0 = 割り込み要求は発生していない

bit 2 OC1IF: 出力コンペア チャンネル 1 割り込みフラグ ステータスビット

1 = 割り込み要求が発生した

0 = 割り込み要求は発生していない

レジスタ 8-5: IFSO: 割り込みフラグステータス レジスタ O(続き)

bit 1 IC1IF: 入力キャプチャ チャンネル 1 割り込みフラグ ステータスビット

1=割り込み要求が発生した

0 = 割り込み要求は発生していない

bit 0 **INTOIF**: 外部割り込み 0 フラグ ステータスビット

1 = 割り込み要求が発生した

0 = 割り込み要求は発生していない

レジスタ 8-6: IFS1: 割り込みフラグステータス レジスタ 1

R/W-0, HS	U-0	R/W-0, HS	U-0				
U2TXIF	U2RXIF	INT2IF	T5IF	T4IF	_	OC3IF	_
bit 15							bit 8

U-0	U-0	U-0	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0	R/W-0
_	_	_	INT1IF	CNIF	CMIF	MI2C1IF	SI2C1IF
bit 7							bit 0

凡例: HS = ハードウェア セット可能ビット

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15 **U2TXIF:** UART2 送信割り込みフラグ ステータスビット

1 = 割り込み要求が発生した

0 = 割り込み要求は発生していない

bit 14 **U2RXIF**: UART2 受信割り込みフラグ ステータスビット

1=割り込み要求が発生した

0 = 割り込み要求は発生していない

bit 13 INT2IF: 外部割り込み 2 フラグ ステータスビット

1 = 割り込み要求が発生した

0 = 割り込み要求は発生していない

bit 12 T5IF: Timer5 割り込みフラグ ステータスビット

1=割り込み要求が発生した

0 = 割り込み要求は発生していない

bit 11 T4IF: Timer4 割り込みフラグ ステータスビット

1 = 割り込み要求が発生した

0 = 割り込み要求は発生していない

bit 10 **未実装:**「0」として読み出し

bit 9 OC3IF: 出力コンペア チャンネル 3 割り込みフラグ ステータスビット

1 = 割り込み要求が発生した

0 = 割り込み要求は発生していない

bit 8-5 **未実装:**「0」として読み出し

bit 4 **INT1IF**: 外部割り込み 1 フラグ ステータスビット

1 = 割り込み要求が発生した

0 = 割り込み要求は発生していない

bit 3 CNIF: 入力状態変化通知割り込みフラグ ステータスビット

1=割り込み要求が発生した

0 = 割り込み要求は発生していない

bit 2 CMIF: コンパレータ割り込みフラグ ステータスビット

1=割り込み要求が発生した

0 = 割り込み要求は発生していない

レジスタ 8-6: IFS1: 割り込みフラグステータス レジスタ 1 (続き)

bit 1 MI2C1IF: マスタ I2C1 イベント割り込みフラグ ステータスビット

1 = 割り込み要求が発生した 0 = 割り込み要求は発生していない

SI2C1IF: スレーブ I2C1 イベント割り込みフラグ ステータスビット

1 = 割り込み要求が発生した

bit 0

0=割り込み要求は発生していない

レジスタ 8-7: IFS2: 割り込みフラグステータス レジスタ 2

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
_	_	_	_	_	_	_	_
bit 15							bit 8

U-0	U-0	R/W-0, HS	U-0	U-0	U-0	R/W-0, HS	R/W-0, HS
_	_	IC3IF	_	_	_	SPI2IF	SPF2IF
bit 7							bit 0

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-6 **未実装:**「0」として読み出し

bit 5 IC3IF: 入力キャプチャ チャンネル 3 割り込みフラグ ステータスビット

1 = 割り込み要求が発生した 0 = 割り込み要求は発生していない

bit 4-2 **未実装:**「0」として読み出し

bit 1 SPI2IF: SPI2 イベント割り込みフラグ ステータスビット

1 = 割り込み要求が発生した

0 = 割り込み要求は発生していない

bit 0 SPF2IF: SPI2 フォルト割り込みフラグ ステータスビット

1 = 割り込み要求が発生した

0 = 割り込み要求は発生していない

レジスタ 8-8: IFS3: 割り込みフラグステータス レジスタ 3

U-0	R/W-0, HS	U-0	U-0	U-0	U-0	U-0	U-0
_	RTCIF	_	_	_	_	_	
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-0, HS	R/W-0, HS	U-0
_	_	_	_	_	MI2C2IF	SI2C2IF	_
bit 7							bit 0

凡例: HS = ハードウェア セット可能ビット

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15 **未実装:**「0」として読み出し

bit 14 RTCIF: リアルタイム クロック / カレンダ割り込みフラグ ステータスビット

1 = 割り込み要求が発生した

0=割り込み要求は発生していない

bit 13-3 **未実装:**「0」として読み出し

bit 2 MI2C2IF: マスタ I2C2 イベント割り込みフラグ ステータスビット

1 = 割り込み要求が発生した

0 = 割り込み要求は発生していない

bit 1 SI2C2IF: スレーブ I2C2 イベント割り込みフラグ ステータスビット

1 = 割り込み要求が発生した 0 = 割り込み要求は発生していない

bit 0 **未実装:**「0」として読み出し

レジスタ 8-9: IFS4: 割り込みフラグステータス レジスタ 4

U-0	U-0	R/W-0, HS	U-0	U-0	U-0	U-0	R/W-0, HS
_	_	CTMUIF	_	_	_	_	HLVDIF
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0, HS	R/W-0, HS	R/W-0, HS	U-0
_	_	_	_	CRCIF	U2ERIF	U1ERIF	_
bit 7							bit 0

凡例: HS = ハードウェア セット可能ビット

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-14 **未実装:**「0」として読み出し

bit 13 CTMUIF: CTMU 割り込みフラグ ステータスビット

1 = 割り込み要求が発生した 0 = 割り込み要求は発生していない

未実装:「0」として読み出し

bit 8 HLVDIF: HIGH/LOW 電圧検出割り込みフラグ ステータスビット

1 = 割り込み要求が発生した 0 = 割り込み要求は発生していない

bit 7-4 **未実装:**「0」として読み出し

bit 12-9

bit 3 CRCIF: CRC ジェネレータ割り込みフラグ ステータスビット

1 = 割り込み要求が発生した 0 = 割り込み要求は発生していない

bit 2 U2ERIF: UART2 エラー割り込みフラグ ステータスビット

1 = 割り込み要求が発生した 0 = 割り込み要求は発生していない

bit 1 U1ERIF: UART1 エラー割り込みフラグ ステータスビット

1 = 割り込み要求が発生した 0 = 割り込み要求は発生していない

bit 0 **未実装:**「0」として読み出し

レジスタ 8-10: IFS5: 割り込みフラグステータス レジスタ 5

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
_	_	_	_	_	_	_	_
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0, HS
_	_	_	_	_	_	_	ULPWUIF
bit 7							bit 0

凡例: HS = ハードウェア セット可能ビット

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「O」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-1 **未実装:**「0」として読み出し

bit 0 ULPWUIF: 超低消費電力復帰割り込みフラグ ステータスビット

1 = 割り込み要求が発生した 0 = 割り込み要求は発生していない

レジスタ 8-11: IECO: 割り込みイネーブル制御レジスタ 0

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
NVMIE	_	AD1IE	U1TXIE	U1RXIE	SPI1IE	SPF1IE	T3IE
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
T2IE	OC2IE	IC2IE	_	T1IE	OC1IE	IC1IE	INT0IE
bit 7							bit 0

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15 NVMIE: NVM 割り込みイネーブルビット

1=割り込み要求を有効にする

0 = 割り込み要求を無効にする

bit 14 **未実装:**「0」として読み出し

bit 13 AD1IE: A/D 変換完了割り込みイネーブルビット

1 = 割り込み要求を有効にする 0 = 割り込み要求を無効にする

bit 12 U1TXIE: UART1 送信割り込みイネーブルビット

1 = 割り込み要求を有効にする 0 = 割り込み要求を無効にする

bit 11 **U1RXIE:** UART1 受信割り込みイネーブルビット

1 = 割り込み要求を有効にする 0 = 割り込み要求を無効にする

bit 10 **SPI1IE**: SPI1 転送完了割り込みイネーブルビット

1 = 割り込み要求を有効にする 0 = 割り込み要求を無効にする

bit 9 SPF1IE: SPI1 フォルト割り込みイネーブルビット

1 = 割り込み要求を有効にする 0 = 割り込み要求を無効にする

bit 8 T3IE: Timer3 割り込みイネーブルビット

1 = 割り込み要求を有効にする 0 = 割り込み要求を無効にする

bit 7 **T2IE:** Timer2 割り込みイネーブルビット

1 = 割り込み要求を有効にする 0 = 割り込み要求を無効にする

bit 6 OC2IE: 出力コンペア チャンネル 2 割り込みイネーブルビット

1 = 割り込み要求を有効にする 0 = 割り込み要求を無効にする

bit 5 IC2IE: 入力キャプチャ チャンネル 2 割り込みイネーブルビット

1 = 割り込み要求を有効にする 0 = 割り込み要求を無効にする

bit 4 **未実装:**「0」として読み出し

bit 3 **T1IE**: Timer1 割り込みイネーブルビット

1 = 割り込み要求を有効にする 0 = 割り込み要求を無効にする

bit 2 OC1IE: 出力コンペア チャンネル 1 割り込みイネーブルビット

レジスタ 8-11: IECO: 割り込みイネーブル制御レジスタ 0

bit 1 IC1IE: 入力キャプチャ チャンネル 1 割り込みイネーブルビット

1 = 割り込み要求を有効にする 0 = 割り込み要求を無効にする

bit 0 **INTOIE**: 外部割り込み 0 イネーブルビット

1 = 割り込み要求を有効にする 0 = 割り込み要求を無効にする

レジスタ 8-12: IEC1: 割り込みイネーブル制御レジスタ 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	U-0
U2TXIE	U2RXIE	INT2IE	T5IE	T4IE	_	OC3IE	_
bit 15	•		•	•	•		bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
_	_	_	INT1IE	CNIE	CMIE	MI2C1IE	SI2C1IE
bit 7							bit 0

_		-	
N	ı٠	ш	
J	ь	171	

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15 **U2TXIE**: UART2 送信割り込みイネーブルビット

1 = 割り込み要求を有効にする

0 = 割り込み要求を無効にする

bit 14 U2RXIE: UART2 受信割り込みイネーブルビット

1 = 割り込み要求を有効にする 0 = 割り込み要求を無効にする

bit 13 **INT2IE**: 外部割り込み 2 イネーブルビット

1 = 割り込み要求を有効にする 0 = 割り込み要求を無効にする

bit 12 T5IE: Timer5 割り込みイネーブルビット

1 = 割り込み要求を有効にする 0 = 割り込み要求を無効にする

bit 11 **T4IE**: Timer4 割り込みイネーブルビット

1 = 割り込み要求を有効にする 0 = 割り込み要求を無効にする

bit 10 **未実装:**「0」として読み出し

bit 9 OC3IE: 出力コンペア 3 割り込みイネーブルビット

1 = 割り込み要求を有効にする 0 = 割り込み要求を無効にする

bit 8-5 **未実装:**「0」として読み出し

bit 4 INT1IE: 外部割り込み 1 イネーブルビット

1 = 割り込み要求を有効にする 0 = 割り込み要求を無効にする

bit 3 CNIE: 入力状態変化通知割り込みイネーブルビット

1 = 割り込み要求を有効にする 0 = 割り込み要求を無効にする

bit 2 CMIE: コンパレータ割り込みイネーブルビット

レジスタ 8-12: IEC1: 割り込みイネーブル制御レジスタ 1

bit 1 MI2C1IE: マスタ I2C1 イベント割り込みイネーブルビット

1 = 割り込み要求を有効にする 0 = 割り込み要求を無効にする

bit 0 SI2C1IE: スレーブ I2C1 イベント割り込みイネーブルビット

1 = 割り込み要求を有効にする 0 = 割り込み要求を無効にする

レジスタ 8-13: IEC2: 割り込みイネーブル制御レジスタ 2

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
_	_	_	_	_	_	_	_
bit 15							bit 8

U-0	U-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
_	_	IC3IE	_	_	_	SPI2IE	SPF2IE
bit 7							bit 0

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-6 **未実装:**「0」として読み出し

bit 5 IC3IE: 入力キャプチャ チャンネル 3 割り込みイネーブルビット

1 = 割り込み要求を有効にする 0 = 割り込み要求を無効にする

bit 4-2 **未実装:**「0」として読み出し

bit 1 SPI2IE: SPI2 イベント割り込みイネーブルビット

1 = 割り込み要求を有効にする 0 = 割り込み要求を無効にする

bit 0 SPF2IE: SPI2 フォルト割り込みイネーブルビット

レジスタ 8-14: IEC3: 割り込みイネーブル制御レジスタ 3

U-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
_	RTCIE	_	_	_	_	_	_
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	U-0
_	_	_	_	_	MI2C2IE	SI2C2IE	_
bit 7							bit 0

凡例:

bit 13-3

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「O」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15 **未実装:**「0」として読み出し

bit 14 RTCIE: リアルタイム クロック / カレンダ割り込みイネーブルビット

1 = 割り込み要求を有効にする 0 = 割り込み要求を無効にする **未実装:**「0」として読み出し

bit 2 MI2C2IE: マスタ I2C2 イベント割り込みイネーブルビット

1 = 割り込み要求を有効にする 0 = 割り込み要求を無効にする

bit 1 SI2C2IE: スレーブ I2C2 イベント割り込みイネーブルビット

1 = 割り込み要求を有効にする 0 = 割り込み要求を無効にする

bit 0 **未実装:**「0」として読み出し

レジスタ 8-15: IEC4: 割り込みイネーブル制御レジスタ 4

U-0	U-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0
_	_	CTMUIE	_	_	_	_	HLVDIE
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	U-0
_	_	_	_	CRCIE	U2ERIE	U1ERIE	_
bit 7							bit 0

凡例:

bit 0

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-14 **未実装:**「0」として読み出し

bit 13 CTMUIE: CTMU 割り込みイネーブルビット

1 = 割り込み要求を有効にする 0 = 割り込み要求を無効にする

bit 12-9 **未実装:**「0」として読み出し

bit 8 HLVDIE: HIGH/LOW 電圧検出割り込みイネーブルビット

1 = 割り込み要求を有効にする 0 = 割り込み要求を無効にする

bit 7-4 **未実装:**「0」として読み出し

bit 3 CRCIE: CRC ジェネレータ割り込みイネーブルビット

1 = 割り込み要求を有効にする 0 = 割り込み要求を無効にする

bit 2 U2ERIE: UART2 エラー割り込みイネーブルビット

1 = 割り込み要求を有効にする 0 = 割り込み要求を無効にする

bit 1 U1ERIE: UART1 エラー割り込みイネーブルビット

1 = 割り込み要求を有効にする 0 = 割り込み要求を無効にする **未実装:**「0」として読み出し

© 2011 Microchip Technology Inc.

レジスタ 8-16: IEC5: 割り込みイネーブル制御レジスタ 5

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
_	_	_	_	_	_	_	_
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
_	_	_	_	_	_	_	ULPWUIE
bit 7							bit 0

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「O」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-1 **未実装:**「0」として読み出し

bit 0 ULPWUIE: 超低消費電力復帰割り込みイネーブルビット

レジスタ 8-17: IPC0: 割り込み優先度制御レジスタ 0

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
_	T1IP2	T1IP1	T1IP0	_	OC1IP2	OC1IP1	OC1IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
_	IC1IP2	IC1IP1	IC1IP0	_	INT0IP2	INT0IP1	INT0IP0
bit 7							bit 0

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「O」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15 **未実装:**「0」として読み出し

bit 14-12 **T1IP<2:0>:** Timer1 割り込み優先度ビット

111 = 割り込み優先度7(最優先割り込み)

•

001 = 割り込み優先度 1

000 = 割り込み要因は無効

bit 11 **未実装:**「0」として読み出し

bit 10-8 OC1IP<2:0>: 出力コンペア チャンネル 1 割り込み優先度ビット

111 = 割り込み優先度7(最優先割り込み)

•

001 = 割り込み優先度 1

000 = 割り込み要因は無効

bit 7 **未実装:**「0」として読み出し

bit 6-4 IC1IP<2:0>: 入力キャプチャ チャンネル 1 割り込み優先度ビット

111 = 割り込み優先度7(最優先割り込み)

•

001 = 割り込み優先度1

000 = 割り込み要因は無効

bit 3 **未実装:**「0」として読み出し

bit 2-0 INT0IP<2:0>: 外部割り込み 0 優先度ビット

111 = 割り込み優先度7(最優先割り込み)

•

001 = 割り込み優先度 1

000 = 割り込み要因は無効

レジスタ 8-18: IPC1: 割り込み優先度制御レジスタ 1

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
_	T2IP2	T2IP1	T2IP0	_	OC2IP2	OC2IP1	OC2IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
_	IC2IP2	IC2IP1	IC2IP0	_	_	_	_
bit 7							bit 0

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「O」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15 **未実装:**「0」として読み出し

bit 14-12 **T2IP<2:0>:** Timer2 割り込み優先度ビット

111 = 割り込み優先度7(最優先割り込み)

•

001 = 割り込み優先度 1 000 = 割り込み要因は無効

bit 11 **未実装:**「0」として読み出し

bit 10-8 OC2IP<2:0>: 出力コンペア チャンネル 2 割り込み優先度ビット

111 = 割り込み優先度7(最優先割り込み)

•

001 = 割り込み優先度 1 000 = 割り込み要因は無効

bit 7 **未実装:**「0」として読み出し

bit 6-4 IC2IP: 入力キャプチャ チャンネル 2 割り込み優先度ビット

111 = 割り込み優先度7(最優先割り込み)

•

001 = 割り込み優先度 1 000 = 割り込み要因は無効

bit 3-0 **未実装:**「0」として読み出し

レジスタ 8-19: IPC2: 割り込み優先度制御レジスタ 2

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
_	U1RXIP2	U1RXIP1	U1RXIP0	_	SPI1IP2	SPI1IP1	SPI1IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
_	SPF1IP2	SPF1IP1	SPF1IP0	_	T3IP2	T3IP1	T3IP0
bit 7							bit 0

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15 **未実装:**「0」として読み出し

bit 14-12 U1RXIP<2:0>: UART1 受信割り込み優先度ビット

111 = 割り込み優先度7(最優先割り込み)

•

001 = 割り込み優先度 1

000 = 割り込み要因は無効

bit 11 **未実装:**「0」として読み出し

bit 10-8 SPI1IP<2:0>: SPI1 イベント割り込み優先度ビット

111 = 割り込み優先度7(最優先割り込み)

•

001 = 割り込み優先度1

000 = 割り込み要因は無効

bit 7 **未実装:**「0」として読み出し

bit 6-4 SPF1IP<2:0>: SPI1 フォルト割り込み優先度ビット

111 = 割り込み優先度7(最優先割り込み)

•

001 = 割り込み優先度 1

000 = 割り込み要因は無効

bit 3 **未実装:**「0」として読み出し

bit 2-0 T3IP<2:0>: Timer3 割り込み優先度ビット

111 = 割り込み優先度7(最優先割り込み)

•

001 = 割り込み優先度 1

000 = 割り込み要因は無効

レジスタ 8-20: IPC3: 割り込み優先度制御レジスタ 3

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
_	NVMIP2	NVMIP1	NVMIP0	_	_		_
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
_	AD1IP2	AD1IP1	AD1IP0	_	U1TXIP2	U1TXIP1	U1TXIP0
bit 7							bit 0

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「O」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15 **未実装:**「0」として読み出し

bit 14-12 NVMIP<2:0>: NVM 割り込み優先度ビット

111 = 割り込み優先度7(最優先割り込み)

•

001 = 割り込み優先度 1 000 = 割り込み要因は無効

bit 11-7 **未実装:**「0」として読み出し

bit 6-4 AD1IP<2:0>: A/D 変換完了割り込み優先度ビット

111 = 割り込み優先度7(最優先割り込み)

•

001 = 割り込み優先度 1 000 = 割り込み要因は無効

bit 3 **未実装:**「0」として読み出し

bit 2-0 **U1TXIP<2:0>:** UART1 送信割り込み優先度ビット

111 = 割り込み優先度7(最優先割り込み)

•

001 = 割り込み優先度 1 000 = 割り込み要因は無効

レジスタ 8-21: IPC4: 割り込み優先度制御レジスタ 4

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
_	CNIP2	CNIP1	CNIP0	_	CMIP2	CMIP1	CMIP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
_	MI2C1P2	MI2C1P1	MI2C1P0	_	SI2C1P2	SI2C1P1	SI2C1P0
bit 7							bit 0

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15 **未実装:**「0」として読み出し

bit 14-12 **CNIP<2:0>:** 入力状態変化通知割り込み優先度ビット

111 = 割り込み優先度7(最優先割り込み)

•

001 = 割り込み優先度 1

000 = 割り込み要因は無効

bit 11 **未実装:**「0」として読み出し

bit 10-8 CMIP<2:0>: コンパレータ割り込み優先度ビット

111 = 割り込み優先度7(最優先割り込み)

•

001 = 割り込み優先度1

000 = 割り込み要因は無効

bit 7 **未実装:**「0」として読み出し

bit 6-4 MI2C1P<2:0>: マスタ I2C1 イベント割り込み優先度ビット

111 = 割り込み優先度7(最優先割り込み)

•

001 = 割り込み優先度 1

000 = 割り込み要因は無効

bit 3 **未実装:**「0」として読み出し

bit 2-0 SI2C1P<2:0>: スレーブ I2C1 イベント割り込み優先度ビット

111 = 割り込み優先度7(最優先割り込み)

•

001 = 割り込み優先度1

000 = 割り込み要因は無効

レジスタ 8-22: IPC5: 割り込み優先度制御レジスタ 5

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
_	_	_	_	_	_	_	_
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
_	_	_	_	_	INT1IP2	INT1IP1	INT1IP0
bit 7							bit 0

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「O」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-3 **未実装:**「0」として読み出し

bit 2-0 **INT1IP<2:0>:** 外部割り込み 1 優先度ビット

111 = 割り込み優先度7(最優先割り込み)

:

001 = 割り込み優先度 1 000 = 割り込み要因は無効

レジスタ 8-23: IPC6: 割り込み優先度制御レジスタ 6

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
_	T4IP2	T4IP1	T4IP0	_	_	_	_
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
_	OC3IP2	OC3IP1	OC3IP0	_	_	_	_
bit 7							bit 0

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「O」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15 **未実装:**「0」として読み出し

bit 14-12 **T4IP<2:0>:** Timer4 割り込み優先度ビット

111 = 割り込み優先度7(最優先割り込み)

:

001 = 割り込み優先度1

000 = 割り込み要因は無効

bit 11-7 **未実装:**「0」として読み出し

bit 6-4 OC3IP: 出力コンペア チャンネル 3 割り込み優先度ビット

111 = 割り込み優先度7(最優先割り込み)

•

001 = 割り込み優先度1

000 = 割り込み要因は無効

bit 3-0 **未実装:**「0」として読み出し

レジスタ 8-24: IPC7: 割り込み優先度制御レジスタ 7

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
_	U2TXIP2	U2TXIP1	U2TXIP0	_	U2RXIP2	U2RXIP1	U2RXIP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
_	INT2IP2	INT2IP1	INT2IP0	_	T5IP2	T5IP1	T5IP0
bit 7							bit 0

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15 **未実装:**「0」として読み出し

bit 14-12 **U2TXIP<2:0>:** UART2 送信割り込み優先度ビット

111 = 割り込み優先度7(最優先割り込み)

•

001 = 割り込み優先度 1

000 = 割り込み要因は無効

bit 11 **未実装:**「0」として読み出し

bit 10-8 U2RXIP<2:0>: UART2 受信割り込み優先度ビット

111 = 割り込み優先度7(最優先割り込み)

•

001 = 割り込み優先度 1

000 = 割り込み要因は無効

bit 7 **未実装:**「0」として読み出し

bit 6-4 INT2IP<2:0>: 外部割り込み 2 優先度ビット

111 = 割り込み優先度7(最優先割り込み)

•

001 = 割り込み優先度 1 000 = 割り込み要因は無効

bit 3 **未実装:**「0」として読み出し

bit 2-0 T5IP: Timer5 割り込み優先度ビット

111 = 割り込み優先度7(最優先割り込み)

•

001 = 割り込み優先度 1 000 = 割り込み要因は無効

レジスタ 8-25: IPC8: 割り込み優先度制御レジスタ 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
_	_	_	_	_	_	_	_
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
_	SPI2IP2	SPI2IP1	SPI2IP0	_	SPF2IP2	SPF2IP1	SPF2IP0
bit 7							bit 0

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「O」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-7 **未実装:**「0」として読み出し

bit 6-4 SPI2IP<2:0>: SPI2 イベント割り込み優先度ビット

111 = 割り込み優先度7(最優先割り込み)

•

001 = 割り込み優先度 1

000=割り込み要因は無効

bit 3 **未実装:**「0」として読み出し

bit 2-0 SPF2IP<2:0>: SPI2 フォルト割り込み優先度ビット

111 = 割り込み優先度7(最優先割り込み)

•

001 = 割り込み優先度 1

000 = 割り込み要因は無効

レジスタ 8-26: IPC9: 割り込み優先度制御レジスタ 9

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
_	_	_	_	_	_	_	_
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
_	IC3IP2	IC3IP1	IC3IP0	_	_	_	_
bit 7							bit 0

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-7 **未実装:**「0」として読み出し

bit 6-4 IC3IP<2:0>: 入力キャプチャ チャンネル 3 イベント割り込み優先度ビット

111 = 割り込み優先度7(最優先割り込み)

•

001 = 割り込み優先度 1 000 = 割り込み要因は無効

bit 3-0 **未実装:**「0」として読み出し

レジスタ 8-27: IPC12: 割り込み優先度制御レジスタ 12

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
_	_	_	_	_	MI2C2IP2	MI2C2IP1	MI2C2IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
_	SI2C2IP2	SI2C2IP1	SI2C2IP0	_	_	_	_
bit 7							bit 0

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-11 **未実装:**「0」として読み出し

bit 10-8 MI2C2IP<2:0>: マスタ I2C2 イベント割り込み優先度ビット

111 = 割り込み優先度7(最優先割り込み)

•

001 = 割り込み優先度 1

000 = 割り込み要因は無効

bit 7 **未実装:**「0」として読み出し

bit 6-4 SI2C2IP<2:0>: スレーブ I2C2 イベント割り込み優先度ビット

111 = 割り込み優先度7(最優先割り込み)

•

001 = 割り込み優先度1

000 = 割り込み要因は無効

bit 3-0 **未実装:**「0」として読み出し

レジスタ 8-28: IPC15: 割り込み優先度制御レジスタ 15

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
_	_	_	_	_	RTCIP2	RTCIP1	RTCIP0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
_	_	_	_	_	_	_	_
bit 7							bit 0

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「O」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-11 **未実装:**「0」として読み出し

bit 10-8 RTCIP<2:0>: リアルタイム クロック / カレンダ割り込み優先度ビット

111 = 割り込み優先度7(最優先割り込み)

•

001 = 割り込み優先度 1 000 = 割り込み要因は無効

bit 7-0 **未実装:**「0」として読み出し

レジスタ 8-29: IPC16: 割り込み優先度制御レジスタ 16

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
_	CRCIP2	CRCIP1	CRCIP0	_	U2ERIP2	U2ERIP1	U2ERIP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
_	U1ERIP2	U1ERIP1	U1ERIP0	_	_	_	
bit 7							bit 0

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15 **未実装:**「0」として読み出し

bit 14-12 CRCIP<2:0>: CRC ジェネレータ エラー割り込み優先度ビット

111 = 割り込み優先度7(最優先割り込み)

•

001 = 割り込み優先度 1

000 = 割り込み要因は無効

bit 11 **未実装:**「0」として読み出し

bit 10-8 **U2ERIP<2:0>:** UART2 エラー割り込み優先度ビット

111 = 割り込み優先度7(最優先割り込み)

•

001 = 割り込み優先度1

000 = 割り込み要因は無効

bit 7 **未実装:**「0」として読み出し

bit 6-4 U1ERIP<2:0>: UART1 エラー割り込み優先度ビット

111 = 割り込み優先度7(最優先割り込み)

•

001 = 割り込み優先度 1

000 = 割り込み要因は無効

bit 3-0 **未実装:**「0」として読み出し

レジスタ 8-30: IPC18: 割り込み優先度制御レジスタ 18

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
_	_	_	_	_	_	_	_
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
_	_	_	_	_	HLVDIP2	HLVDIP1	HLVDIP0
bit 7							bit 0

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「O」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-3 **未実装:**「0」として読み出し

bit 2-0 **HLVDIP<2:0>:** HIGH/LOW 電圧検出割り込み優先度ビット

111 = 割り込み優先度7(最優先割り込み)

.

001 = 割り込み優先度 1 000 = 割り込み要因は無効

レジスタ 8-31: IPC19: 割り込み優先度制御レジスタ 19

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
_	_		_	_	_		_
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
_	CTMUIP2	CTMUIP1	CTMUIP0	_	_	_	_
bit 7							bit 0

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「O」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-7 **未実装:**「0」として読み出し

bit 6-4 CTMUIP<2:0>: CTMU 割り込み優先度ビット

111 = 割り込み優先度7(最優先割り込み)

:

001 = 割り込み優先度1

000 = 割り込み要因は無効

bit 3-0 **未実装:**「0」として読み出し

レジスタ 8-32: IPC20: 割り込み優先度制御レジスタ 20

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
_	_	_	_	_	_	_	_
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
_	_	_	_	_	ULPWUIP2	ULPWUIP1	ULPWUIP0
bit 7							bit 0

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「O」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-3 **未実装:**「0」として読み出し

bit 6-4 ULPWUIP<2:0>: 超低消費電力復帰割り込み優先度ビット

111 = 割り込み優先度7(最優先割り込み)

•

001 = 割り込み優先度 1 000 = 割り込み要因は無効

レジスタ 8-33: INTTREG: 割り込み制御/ステータス レジスタ

R-0	U-0	R/W-0	U-0	R-0	R-0	R-0	R-0
CPUIRQ	_	VHOLD	_	ILR3	ILR2	ILR1	ILR0
bit 15							bit 8

U-0	R-0						
_	VECNUM6	VECNUM5	VECNUM4	VECNUM3	VECNUM2	VECNUM1	VECNUM0
bit 7							bit 0

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15 CPUIRQ: 割り込みコントローラからの割り込み要求 CPU ビット

1 = 割り込み要求が発生したが、CPU はまだ承認していない

(これは、CPU 優先度の方が割り込み優先度よりも高い場合に発生します)

0 = 未承認の割り込み要求は存在しない

bit 14 **未実装:**「0」として読み出し

bit 13 VHOLD: ベクタホールド ビット

ベクタ番号のキャプチャを可能にし、VECNUMビットにどの割り込みを保存するのかを変更します。

1 = VECNUM は、現在処理中の割り込みではなく、最優先の保留中割り込みの番号を格納する

0 = VECNUMは、直前に承認された割り込みの番号を格納する(たとえ他の割り込みが保留中であっ

ても、直前に発生した CPU よりも高優先度の割り込みの番号を格納します)

bit 12 **未実装:**「0」として読み出し

bit 11-8 ILR<3:0>: 新規 CPU 割り込み優先度ビット

1111 = CPU 割り込み優先度 15

.

0001 = CPU 割り込み優先度 1

0000 = CPU 割り込み優先度 0

bit 7 **未実装:**「0」として読み出し

bit 6-0 **VECNUM<6:0>:** 保留中割り込みベクタ番号ビット

0111111 = 保留中の割り込みベクタ番号は 135

.

0000001 = 保留中の割り込みベクタ番号は9 0000000 = 保留中の割り込みベクタ番号は8

8.4 割り込みのセットアップ手順

8.4.1 初期化

下記の手順で割り込み要因を設定します。

- 割り込みをネストしない場合、NSTDIS 制御ビット (INTCON1<15>) をセットします。
- 2. 適切な IPCx レジスタの制御ビットで、割り込み 要因のユーザ割り当て優先度を選択します。優先 度は、アプリケーションと割り込み要因のタイプ によって決まります。複数の優先度を設定する必 要がない場合、全ての有効割り込み要因の IPCx レジスタ制御ビットをゼロ以外の同じ値に設定で きます。

Note: デバイスリセットで IPCx レジスタは初期 化され、全てのユーザ割り込み要因に優先 度 4 が割り当てられます。

- 3. 適切な IFSx レジスタで、その周辺モジュールに 対応する割り込みフラグ ステータスビットをク リアします。

8.4.2 割り込みサービスルーチン

ISR を宣言して IVT を正しいベクタアドレスに初期化する方法は、プログラミング言語 (C またはアセンブラ)とアプリケーションの開発に使用する言語開発ツールセットによって異なります。一般的にユーザは、ISRで処理する割り込み要因に対応する適切なIFSxレジスタで割り込みフラグをクリアする必要があります。クリアしないと、ISR はルーチン終了後即座に再実行されます。ISR をアセンブリ言語でコーディングする場合、保存した PC 値、SRL 値、以前の CPU 優先度をアンスタックするために、RETFIE 命令を使用して ISR を終了する必要があります。

8.4.3 トラップサービスルーチン (TSR)

トラップサービスルーチン(TSR)もISRと同様にコーディングできますが、TSR への再入を防ぐために、INTCON1 レジスタで適切なトラップ ステータスフラグをクリアする必要があります。

8.4.4 割り込みの無効化

全てのユーザ割り込みは、下記の手順で無効化できます。

- 1. PUSH 命令を使用して、現在の SR 値をソフトウェア スタックにプッシュします。
- SRL と値 E0h で論理 OR 演算を行って、CPU の 優先度を強制的に 7 に設定します。

ユーザ割り込みを有効にするには、POP 命令を使用して以前の SR 値を復元します。

この方法では、優先度7以下のユーザ割り込みだけを無効化できます。トラップ要因(優先度8~15)を無効にする事はできません。

DISI 命令を使用すると、優先度 1~6の割り込みを簡単に一定期間無効化できます。DISI 命令は優先度 7の割り込み要因を無効化しません。

NOTE:

9.0 オシレータのコンフィグレーション

Note: 本書は、PIC24 デバイスの中の特定製品 グループを対象とし、それらの機能の概要説明を目的としています。従って本書 は、包括的な参照資料の提供を意図したものではありません。オシレータ コンフィグレーションの詳細は「PIC24F ファミリ リファレンス マニュアル」のセクション 38.「500 kHz 低消費電力 FRC 等のオシレータ」(DS39726) を参照してください。

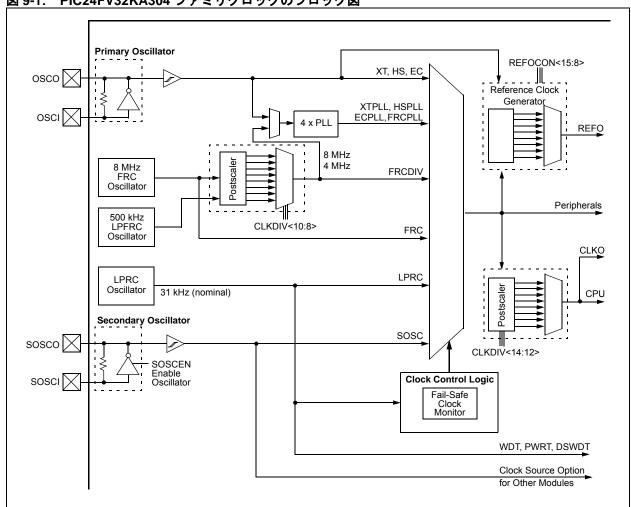
PIC24FV32KA304 ファミリのオシレータ システムは 下記の特長を備えます。

- 11種類のクロックモードを提供する合計5種類の外部/内部オシレータオプション
- 選択した内部および外部発振源に内蔵の4 逓倍位相 ロックループ (PLL) を併用して内部動作周波数を逓 倍可能

- ソフトウェア制御による各種クロック源の切り換え
- ・ソフトウェア制御可能なポストスケーラにより、 CPU のクロック周波数を変更してシステムの消費 電力を低減
- EC モード用のシステム周波数レンジ宣言ビット (外部クロック源を使用する際に、この宣言ビット を予測される周波数レンジに設定する事により、消 費電流を低減)
- クロック障害を検出し、アプリケーションを安全に 回復またはシャットダウンするフェイルセーフ ク ロック監視機能 (FSCM)

図 9-1 に、オシレータ システムの概略ブロック図を示します。





9.1 CPU クロッキング方式

システムクロック源は、下記の4つ発信源のいずれかによって供給できます。

- OSCIおよびOSCOピン上のプライマリオシレータ (POSC)
- SOSCIおよびSOSCOピン上のセカンダリ オシレータ (SOSC)

PIC24FV32KA304ファミリデバイスは下記の2種類のセカンダリ オシレータを備えます。

- 高消費電力セカンダリ オシレータ
- 低消費電力セカンダリ オシレータ

これらは SOSCSEL (FOSC<5>) ビットで選択できます。

- 高速内部 RC (FRC) オシレータ
 - 8 MHz FRC オシレータ
 - 500 kHz 低消費電力 FRC オシレータ
- 低消費電力内部 RC (LPRC) オシレータ (下記 2 モードが可能)
 - 高消費電力/高精度モード
 - 低消費電力/低精度モード

プライマリオシレータと8MHzFRCオシレータには、必要に応じて内部4xPLLを併用できます。FRCクロック源の周波数は、プログラマブルなクロック分周器を使用して低減できます。選択されたクロック源は、プロセッサ用と周辺モジュール用にクロックを生成します。

プロセッサクロックを2分周する事により、内部命令サイクルクロック FCY を生成します。本書では、命令サイクルクロックを FOSC/2 と表記する場合があります。プライマリオシレータの一部の動作モードでは、内部命令サイクルクロック FOSC/2をOSCO I/O ピンに出力できます。

9.2 POR 時の初期設定

デバイスのパワーオン リセット (POR) イベント時に使 用するクロック源と動作モードは、コンフィグレー ション ビットで選択します。オシレータ コンフィグ レーション ビットの設定は、プログラムメモリ内のコ ンフィグレーション レジスタに格納されます(詳細は 26.1 「コンフィグレーション ビット」参照)。POR 時に 使用するクロック源は、プライマリ オシレータ コン フィグレーション ビット POSCMD<1:0> と初期オシレータ選択ビット (FOSC<1:0>) FNOSC<2:0> (FOSCSEL<2:0>) で選択します。既定値 (未プログラミング状態)では、ポストスケーラ併用 FRCプライマリオシレータ(FRCDIV)が選択されます。 これらのビットを書き換える事により、セカンダリオ シレータまたは内部オシレータのいずれかを選択でき ます。EC モード周波数レンジ コンフィグレーション ビット POSCFREQ<1:0> (FOSC<4:3>) は、EC モー ドで動作する時の消費電力を最適化します。既定値設 定は「周波数レンジは8MHzより高い」です。

コンフィグレーション ビットを設定する事により、 ユーザは表 9-1 に示す各種クロックモードを選択でき ます。

9.2.1 クロック切り換えモード コンフィグレーション ビット

FCKSM コンフィグレーション ビット (FOSC<7:6>)では、デバイスのクロック切り換えと FSCM 機能を設定します。FCKSM1 を「0」にプログラミングした場合にのみ、クロック切り換えが有効化されます。FCKSM<1:0> を「00」にプログラミングした場合にのみ、FSCM 機能が有効化されます。

表 9-1: クロック選択用コンフィグレーション ビットの値

オシレータモード	クロック源	POSCMD<1:0>	FNOSC<2:0>	Note
ポストスケーラ併用 8 MHz FRC オシレータ (FRCDIV)	内部	11	111	1, 2
ポストスケーラ併用 500 kHz FRC オシレータ (LPFRCDIV)	内部	11	110	1
低消費電力 RC オシレータ (LPRC)	内部	11	101	1
セカンダリ (Timer1) オシレータ (SOSC)	セカンダリ	00	100	1
PLL モジュール併用プライマリ オシレータ (HSPLL)	プライマリ	10	011	
PLL モジュール併用プライマリ オシレータ (ECPLL)	プライマリ	00	011	
プライマリ オシレータ (HS)	プライマリ	10	010	
プライマリ オシレータ (XT)	プライマリ	01	010	
プライマリ オシレータ (EC)	プライマリ	0.0	010	
PLL モジュール併用 8 MHz FRC オシレータ (FRCPLL)	内部	11	001	1
8 MHz FRC オシレータ (FRC)	内部	11	000	1

Note 1: OSCO ピンの機能は、OSCIOFNC コンフィグレーション ビットによって決まります。

2: これは未プログラミング(消去後)デバイスの既定値オシレータモードです。

9.3 制御レジスタ

オシレータ動作の制御には、下記の3つの特殊機能レジスタ(SFR)を使用します。

- OSCCON
- CLKDIV
- OSCTUN

OSCCON レジスタ (レジスタ 9-1 参照)は、オシレータ用のメイン制御レジスタです。このレジスタはクロック源の切り換えを制御し、クロック源の監視を可能にします。

クロック分周レジスタ (レジスタ 9-2 参照) は、Doze モード関連の機能と FRC オシレータのポストスケー ラを制御します。

FRC オシレータ調整レジスタ (レジスタ 9-3 参照)では、FRC オシレータを約 \pm 5.25% のレンジで微調整できます。このレジスタの調整ビット値を増減すると、FRC オシレータの工場校正周波数が一定量変化します。

レジスタ 9-1: OSCCON: オシレータ制御レジスタ

U-0	R-0, HSC	R-0, HSC	R-0, HSC	U-0	R/W-x ⁽¹⁾	R/W-x ⁽¹⁾	R/W-x ⁽¹⁾
_	COSC2	COSC1	COSC0	_	NOSC2	NOSC1	NOSC0
bit 15							bit 8

R/SO-0, HSC	U-0	R-0, HSC ⁽²⁾	U-0	R/CO-0, HS	R/W-0 ⁽³⁾	R/W-0	R/W-0
CLKLOCK	_	LOCK	_	CF	SOSCDRV	SOSCEN	OSWEN
bit 7		•					bit 0

凡例: HSC = ハードウェア セット / クリア可能ビット

HS = ハードウェア セット可能ビット CO = クリアのみ可能ビット SO = セットのみ可能ビット

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「O」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15 **未実装:**「0」として読み出し

bit 14-12 COSC<2:0>: 切り換え前クロック源選択ビット

111 = ポストスケーラ併用 8 MHz FRC オシレータ (FRCDIV)

110 = ポストスケーラ併用 500 kHz 低消費電力 FRC オシレータ (LPFRCDIV)

101 = 低消費電力 RC オシレータ (LPRC)

100 = セカンダリ オシレータ (SOSC)

011 = PLL モジュール併用プライマリ オシレータ (XTPLL、HSPLL、ECPLL)

010 = プライマリ オシレータ (XT、HS、EC)

001 = ポストスケーラ /PLL モジュール併用 8 MHz FRC オシレータ (FRCPLL)

000 = 8 MHz FRC オシレータ (FRC)

bit 11 **未実装:**「0」として読み出し

bit 10-8 NOSC<2:0>: 切り換え後クロック源選択ビット (1)

111 = ポストスケーラ併用 8 MHz FRC オシレータ (FRCDIV)

110 = ポストスケーラ併用 500 kHz 低消費電力 FRC オシレータ (LPFRCDIV)

101 = 低消費電力 RC オシレータ (LPRC)

100 = セカンダリ オシレータ (SOSC)

011 = PLL モジュール併用プライマリ オシレータ (XTPLL、HSPLL、ECPLL)

010 = プライマリ オシレータ (XT、HS、EC)

001 = ポストスケーラ /PLL モジュール併用 8 MHz FRC オシレータ (FRCPLL)

000 = 8 MHz FRC オシレータ (FRC)

Note 1: これらのビットのリセット値は、FNOSC コンフィグレーション ビットによって決まります。

2: クロック切り換えの実行中または PLL を併用しないクロックモードを選択した場合にも「0」にリセットされます。

3: SOSC の発信源に外部水晶振動子ではなくデジタルクロック入力を選択した (SOSCSRC = 0) 場合、このビットは効果を持ちません。

レジスタ 9-1: OSCCON: オシレータ制御レジスタ (続き)

bit 7 CLKLOCK: クロック選択ロック イネーブルビット

FSCM が有効 (FCKSM1 = 1) の場合:

1 = クロックと PLL の選択をロックする

0 = クロックと PLL の選択をロックしない (OSWEN ビットのセットによる変更が可能)

FSCM が無効 (FCKSM1 = 0) の場合:

クロックと PLL の選択をロックしない (OSWEN ビットのセットによる変更が可能)

bit 6 **未実装:**「0」として読み出し

bit 5 **LOCK:** PLL ロック ステータスビット (2)

1 = PLL モジュールはロック状態、または PLL モジュール起動タイマが終了した

0 = PLL モジュールはロック解除状態、または PLL 起動タイマが動作中、または PLL は無効

bit 4 **未実装:**「0」として読み出し

bit 3 CF: クロック障害検出ビット

1 = FSCM がクロック障害を検出した

0= クロック障害は検出されていない

bit 2 **SOSCDRV**: セカンダリ オシレータ駆動強度ビット ⁽³⁾

1 = 高消費電力 SOSC 回路を選択する

0 = SOSCSRC コンフィグレーション ビットで低 / 高消費電力を選択する

bit 1 SOSCEN: 32kHz セカンダリ オシレータ (SOSC) イネーブルビット

L= セカンダリ オシレータを有効にする

0 = セカンダリ オシレータを無効にする OSWEN: オシレータ切り換えイネーブルビット

1 = NOSC<2:0> ビットで選択したクロック源へのオシレータ切り換えを開始する

0 = オシレータ切り換えは完了した

Note 1: これらのビットのリセット値は、FNOSC コンフィグレーション ビットによって決まります。

2: クロック切り換えの実行中または PLL を併用しないクロックモードを選択した場合にも「0」にリセット カルカカ

トされます。

bit 0

3: SOSC の発信源に外部水晶振動子ではなくデジタルクロック入力を選択した (SOSCSRC = 0) 場合、このビットは効果を持ちません。

レジスタ 9-2: CLKDIV: クロック分周レジスタ

R/W-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0	R/W-0	R/W-1
ROI	DOZE2	DOZE1	DOZE0	DOZEN ⁽¹⁾	RCDIV2	RCDIV1	RCDIV0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
_	_	_	_	_	_	_	
bit 7							bit 0

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15 ROI: 割り込み時復帰ビット

1 = 割り込みは DOZEN ビットをクリアし、CPU/ 周辺モジュール クロック比を 1:1 ヘリセットする

0 = 割り込みは DOZEN ビットに影響しない

bit 14-12 **DOZE<2:0>:** CPU/ 周辺モジュール クロック比選択ビット

111 = 1:128

110 = 1:64

101 = 1:32

100 = 1:16

011 = 1:8

010 = 1:4

001 = 1:2

000 = 1:1

bit 11 DOZEN: Doze イネーブルビット (1)

1 = DOZE<2:0> ビットで CPU/ 周辺モジュール クロック比を指定する

0 = CPU/ 周辺モジュール クロック比を 1:1 に設定する

bit 10-8 RCDIV<2:0>: FRC ポストスケーラ選択ビット

OSCCON (COSC<2:0>) = 111 の時:

111 = 31.25 kHz (256 分周)

110 = 125 kHz (64 分周)

101 = 250 kHz (32 分周)

100 = 500 kHz (16 分周)

011 = 1 MHz (8 分周)

010 = 2 MHz (4 分周)

001 = 4 MHz (2 分周)(既定值)

000 = 8 MHz (1 分周)

OSCCON (COSC<2:0>) = 110 の時:

111 = 1.95 kHz (256 分周)

111 = 7.83 kHz (230 分周 110 = 7.81 kHz (64 分周)

101 = 15.62 kHz (32 分周)

100 = 31.25 kHz (16 分周)

011 = 62.5 kHz (8 分周)

010 = 125 kHz (4 分周)

001 = 250 kHz (2 分周)(既定值)

000 = 500 kHz (1 分周)

bit 7-0 **未実装:**「0」として読み出し

Note 1: ROI ビットがセットされて割り込みが発生すると、このビットは自動的にクリアされます。

レジスタ 9-3: OSCTUN: FRC オシレータ調整レジスタ

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
_	_	_	_	_	_	_	_
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
_	_	TUN5 ⁽¹⁾	TUN4 ⁽¹⁾	TUN3 ⁽¹⁾	TUN2 ⁽¹⁾	TUN1 ⁽¹⁾	TUN0 ⁽¹⁾
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、[0] として読み出し [0] -n = POR 時の値 [0] 1 = ビットをセット [0] 0 = ビットをクリア [0] x = ビットは未知

bit 15-6 **未実装:**「0」として読み出し

bit 5-0 **TUN<5:0>:** FRC オシレータ調整ビット ⁽¹⁾

011111 = 高周波数側へ最大調整

011110

:

:

000001

000000 = 中心周波数、オシレータは工場校正周波数で動作

111111

:

100001

100000 = 低周波数側へ最大調整

Note 1: TUN<5:0> を1変化させた時の FRC 周波数の変化量は、FRC 調整レンジ全体で一定ではありません。

9.4 クロック切り換え動作

アプリケーションは、ソフトウェア制御によって、4種類のクロック源 (POSC、SOSC、FRC、LPRC) をいつでも自由に、ほぼ無制限に切り換える事ができます。このような柔軟な切り換え動作によって予期せぬ結果が生じる可能性もあるため、PIC24F は切り換えプロセスに保護ロック機能を組み込んでいます。

Note: プライマリ オシレータモードでは、 POSCMDx コンフィグレーション ビットで 3 種類のサブモード (XT、HS、EC) を選択できます。アプリケーションは、プライマリ オシレータモードから他のモードへの切り換え、あるいは他のモードへの切り換えをソフトウェアで実行できますが、プライマリ オシレータのサブモードを切り換えるにはデバイスを再プログラミングする必要があります。

9.4.1 クロック切り換えの有効化

クロック切り換えを有効にするには、FOSC コンフィグレーションレジスタのFCKSM1コンフィグレーション ビットに「0」をプログラミングする必要があります。詳細は 26.0 「その他の特殊な機能」を参照してください。FCKSM1コンフィグレーション ビットを未プログラム状態 (1) から書き換えない場合、クロック切り換え機能とフェイルセーフ クロック監視 (FSCM)機能は無効です (既定値設定)。

クロック切り換えが無効の場合、NOSCx 制御ビット (OSCCON<10:8>)でクロック選択を制御する事はできません。ただし、COSCx ビット (OSCCON<14:12>) には FNOSCx コンフィグレーション ビットで選択されたクロック源が反映されます。

クロック切り換えが無効の場合、OSWEN 制御ビット (OSCCON<0>) は効果を持たず、常に「0」を保持します。

9.4.2 オシレータ切り換えシーケンス

クロック切り換えの実行には少なくとも下記の基本 シーケンスが必要です。

- 1. 必要に応じて COSCx ビット (OSCCON<14:12>) を 読み出し、現在使用中のクロック源を特定します。
- 2. OSCCON レジスタの上位バイトに書き込むため のロック解除シーケンスを実行します。
- 3. 切り換え後のクロック源を指定する値をNOSC制 御ビット (OSCCON<10:8>) に書き込みます。
- 4. OSCCON レジスタの下位バイトに書き込むため のロック解除シーケンスを実行します。
- 5. OSWEN ビットをセットしてオシレータ切り換え を開始します。

この基本シーケンスが完了すると、システムクロック ハードウェアは自動的に下記のように応答します。

- 1. クロック切り換えハードウェアは、COSCxステータスビットと NOSCx ビットの新しい値を比較します。両者が一致する場合、クロック切り換えは冗長動作であるため、OSWEN ビットは自動的にクリアされ、クロック切り換えは中止します。
- クロック切り換えを開始した場合、LOCK (OSCCON<5>) および CF (OSCCON<3>) ビット がクリアされます。
- 3. 切り換え先のオシレータが動作中でない場合、ハードウェアはそのオシレータを起動します。水晶振動子を起動する必要がある場合、ハードウェアは OST 時間が経過するまで待機します。切り換え先のクロック源が PLL を併用する場合、ハードウェアは PLL ロック (LOCK = 1) を検出するまで待機します。
- ハードウェアは、切り換え後のクロック源からの 10 クロックサイクルを待ってからクロック切り 換えを実行します。
- 5. ハードウェアはOSWENビットをクリアし、クロック切り換えが正常に完了した事を示します。 さらに、NOSCx ビット値を COSCx ビットへ転送します。
- 6. 切り換え前のクロック源はこの時点で停止します。ただし例外として、LPRCをクロック源として使用するWDT、FSCM、RTCCのいずれかが有効である場合、LPRCは動作を継続します。また、SOSCENがセットされたままであれば、SOSCは動作を継続します。
 - Note 1: クロック切り換えシーケンスの間もプロセッサはコード実行を継続します。この間、タイミングが重要なコードを実行しないでください。
 - 2: PLL併用プライマリオシレータモードと FRCPLL モード間で直接クロックを切り 換える事はできません (どちらの方向に も切り換える事はできません)。このよ うに2つの PLL モード間で切り換えを行 う場合、FRC モードを一時的なクロック 源として経由してから切り換える必要 があります。

クロック切り換えには下記のコードシーケンスを推奨 します。

- 1. OSCCON レジスタのロック解除および書き込み シーケンス中の割り込みを無効にする。
- 連続する 2 つの命令で OSCCON<15:8> に 78h と 9Ah を書き込んで、OSCCON 上位バイトのロック 解除シーケンスを実行する。
- 3. ロック解除シーケンス直後の命令で、切り換え後のクロック源を指定する値を NOSCx ビットに書き込む。
- 連続する2つの命令でOSCCON<7:0> に46h と57h を書き込んで、OSCCON下位バイトのロック解除シーケンスを実行する。
- ロック解除シーケンス直後の命令で OSWEN ビットをセットする。
- 6. クロックに影響されにくいコードの実行を継続 する(必須ではない)。
- 適量のソフトウェア遅延(サイクルカウント)を 実行して、起動したオシレータと PLL を安定させる。
- 8. OSWEN が「0」である事を確認する(このビットが「0」であれば、クロック切り換えが成功した事を意味します。OSWEN がセットされたままであれば、エラーの原因を特定するために LOCK ビットを確認します)

例 9-1 に、OSCCON レジスタのロックを解除してクロック切り換えを開始するコードシーケンスを示します。

例 9-1: クロック切り換え用の基本コードシーケ ンス

;Place the new oscillator selection in WO ;OSCCONH (high byte) Unlock Sequence MOV #OSCCONH, w1 MOV #0x78, w2 MOV #0x9A, w3 MOV.b w2, [w1] MOV.b w3, [w1] ;Set new oscillator selection MOV.b WREG, OSCCONH ;OSCCONL (low byte) unlock sequence MOV #OSCCONL, w1 MOV #0x46, w2 MOV #0x57, w3 MOV.b w2, [w1] MOV.b w3, [w1] ;Start oscillator switch operation BSET OSCCON,#0

9.5 参照クロック出力

PIC24FV32KA304 ファミリデバイスが内蔵するデバイスクロックは、一部のオシレータモードで利用可能な CLKO 出力 (Fosc/2) 以外に、ポートピンで参照クロック信号を出力する事もできます。この機能は全てのオシレータ コンフィグレーションで利用できます。これにより、アプリケーションの外部デバイスを駆動するために、より広いレンジのクロック分周比を選択できます。

この参照クロック出力の制御には、REFOCON レジスタ (レジスタ 9-4 参照)を使用します。ROEN ビット (REFOCON<15>)をセットすると、クロック信号が REFO ピンに出力されます。RODIV ビット (REFOCON<11:8>)により、クロック分周比を 16 段階に選択できます。

ROSSLP および ROSEL ビット (REFOCON<13:12>)では、スリープモード時にも参照出力を供給するかどうかを指定します。ROSEL ビットにより、参照クロック出力のクロック源を選択します (OSC1/OSC2 上のオシレータまたは現在のシステムクロック源から選択)。ROSSLP ビットでは、デバイスがスリープモード中でも REFOで参照クロックを出力するかどうかを指定します。

スリープモード時にも参照クロック出力を使用するには、ROSSLP ビットと ROSEL ビットの両方をセットする必要があります。ROSEL ビットをセットしない場合、デバイスクロックをプライマリモード (EC、HS、XT) のいずれかに設定する必要があります。そうしないと、OSC1 と OSC2 上のオシレータは、デバイスがスリープモードに移行した時に停止します。ROSELビットをクリアした場合、クロック切り換えでシステムクロック周波数を変更した時に参照出力周波数も変化します。

レジスタ 9-4: REFOCON: 参照オシレータ制御レジスタ

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ROEN	_	ROSSLP	ROSEL	RODIV3	RODIV2	RODIV1	RODIV0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
_	_	_	_	_	_	_	_
bit 7							bit 0

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15 ROEN: 参照オシレータ出力イネーブルビット

1 = REFO ピンの参照オシレータ出力を有効にする

0=参照オシレータを無効にする

bit 14 **未実装:**「0」として読み出し

bit 13 ROSSLP: スリープ時参照オシレータ出力停止ビット

1=スリープ時にも参照オシレータの動作を継続する

0=スリープ時には参照オシレータの動作を停止する

bit 12 ROSEL: 参照クロック源選択ビット

1 = プライマリ オシレータをベースクロックとして使用する⁽¹⁾

0 = システムクロックをベースクロックとして使用する (デバイスのクロック切り換えはベースクロックに影響します)

bit 11-8 RODIV<3:0>: 参照 オシレータ分周比選択ビット

1111 = ベースクロックを 32,768 分周

1110 = ベースクロックを 16,384 分周

1101 = ベースクロックを 8,192 分周

1100 = ベースクロックを 4,096 分周

1011 = ベースクロックを 2,048 分周

1010 = ベースクロックを 1,024 分周

1001 = ベースクロックを 512 分周

1000 = ベースクロックを 256 分周

0111 = ベースクロックを 128 分周

0110 = ベースクロックを 64 分周

0101 = ベースクロックを 32 分周

0100 = ベースクロックを 16 分周

0011 = ベースクロックを8分周

0010 = ベースクロックを 4 分周

0001 = ベースクロックを2分周

0000 = ベースクロックのまま

bit 7-0 **未実装:**「0」として読み出し

Note 1: FOSC<1:0> ビットで水晶振動子を有効化しておく必要があります; 水晶振動子はスリープモード時も動作を継続します。

NOTE:

10.0 省電力機能

Note: 本書は、PIC24F デバイスの中の特定製品

グループを対象とし、それらの機能の概要説明を目的としています。従って本書は、包括的な参照資料の提供を意図したものではありません。省電力機能の詳細は「PIC24Fファミリリファレンスマニュアル」のセクション39.「ディープスリープ付き省電力機能」(DS39727)を参照してください。

PIC24FV32KA304 ファミリは、CPU と周辺モジュールに供給するクロックを別々に管理する事によって、消費電力を抑える事ができます。一般的に、クロック周波数が低いほど、また、クロックの供給を受ける回路の数が少ないほど、消費電力は低下します。全てのPIC24F デバイスは、下記の4つの方法で電力消費を管理します。

- クロック周波数
- 命令によるスリープ、アイドル、ディープスリープ モード
- ・ ソフトウェア制御 Doze モード
- ソフトウェアによる周辺モジュールの選択的制御

これらの方法を組み合わせる事で、タイミングが重要 な通信等のアプリケーション機能をそのまま維持しな がら、アプリケーション全体の消費電力を選択的に低 減できます。

10.1 クロック周波数とクロック切り換え

PIC24Fでは、幅広いレンジのクロック周波数をアプリケーション制御で選択できます。システムクロック設定がロックされていない場合、NOSC ビットを変更するだけで、低消費電力または高精度のオシレータを選択できます。動作中にシステムクロックを切り換える方法と制限事項の詳細は9.0「オシレータのコンフィグレーション」に記載しています。

10.2 命令ベースの省電力モード

PIC24F は、デバイスを特殊な省電力モードに移行させる2つの PWRSAV 命令を備えます。スリープモードはクロック動作を停止し、全てのコード実行を一時停止します。アイドルモードも CPU とコード実行を一時停止しますが、周辺モジュールは動作を継続できます。ディープスリープ モードはクロック動作、コード実行、周辺モジュール (RTCC と DSWDT を除く)の

全てを停止します。さらにこのモードは I/O ステータスを保持し、SRAM とフラッシュメモリへの電力供給も停止します。

PWRSAV 命令のアセンブリ構文を例 10-1 に示します。

Note: SLEEP_MODE と IDLE_MODE は、使用するデバイス用のアセンブラ インクルード ファイルで定義されている定数です。

スリープモードとアイドルモードは割り込み、WDTタイムアウト、デバイスリセットにより終了できます。 デバイスがこれらの省電力モードを終了する事を「復帰」と呼びます。

10.2.1 スリープモード

スリープモードは下記のように機能します。

- システムクロック源をシャットダウンします。内蔵 オシレータを使用している場合、これを停止します。
- I/O ピンから外部へ電流を供給していない場合、デバイスの消費電流を最小限に抑える事ができます。
- I/O ピンの方向と状態を保持します。
- スリープモード中はシステムクロック源が停止する ため、フェイルセーフ クロック監視 (FSCM) は機能 しません。
- LPRC をクロック源とする WDT または RTCC が有 効である場合、LPRC クロックはスリープモードで も動作を続けます。
- WDT が有効な場合、スリープモードに移行する前にWDT は自動的にクリアされます。
- ・ 一部のデバイス機能または周辺モジュールは、スリープモード中も動作を継続します。これには I/O ポートの入力状態変化通知や外部クロック入力を使用する周辺モジュール等が含まれます。動作にシステムクロック源を使用する周辺モジュールは全てスリープモード時に停止します。

以下のイベントのいずれかが発生すると、デバイスは スリープモードから復帰します。

- 有効化された割り込み要因のいずれか
- 全てのデバイスリセット
- WDT タイムアウト

スリープモードからの復帰時に、プロセッサはスリー プ移行前と同じクロック源を使用して再起動します。

例 10-1: PWRSAV 命令の構文

PWRSAV #SLEEP_MODE ; Put the device into SLEEP mode
PWRSAV #IDLE_MODE ; Put the device into IDLE mode

BSET DSCON, #DSEN ; Enable Deep Sleep

PWRSAV #SLEEP_MODE ; Put the device into Deep SLEEP mode

10.2.2 アイドルモード

アイドルモードは下記のように機能します。

- ・ CPU は命令の実行を停止します。
- WDT は自動的にクリアされます。
- ・ システムクロック源は動作を維持します。既定値では、全ての周辺モジュールがシステムクロック源を使用して通常動作を継続しますが、一部の周辺モジュールを選択的に無効にする事も可能です (10.6 「選択的な周辺モジュールの制御」参照)。
- WDT または FSCM が有効な場合、LPRC も動作を 維持します。

以下のイベントのいずれかが発生すると、デバイスは アイドルモードから復帰します。

- 有効化された割り込みのいずれか
- 全てのデバイスリセット
- ・ WDT タイムアウト

アイドルモードから復帰すると、CPU へのクロック供給が再開し、PWRSAV 命令の次の命令または ISR 内の先頭命令から命令実行が再開します。

10.2.3 省電力命令と割り込みの同時発生

PWRSAV 命令の実行と同時に発生した割り込みは、スリープまたはアイドルモードへの移行が完了するまで保留されます。移行後、デバイスはスリープまたはアイドルモードから復帰します。

10.2.4 ディープスリープ モード

PIC24FV32KA304 ファミリのディープスリープ モードは、外部スイッチを必要とせずにデバイスから全ての電力消費を排除する事によって、消費電力を最小限に抑える事を目的とします。ディープスリープ モードへの移行は完全にソフトウェアで制御されます。ディープスリープ モードからの復帰は以下のイベントのいずれかによってトリガできます。

- ・ POR イベント
- ・ MCLR イベント
- ・ RTCC アラーム (RTCC アラームが存在する場合)
- 外部割り込み 0
- ディープスリープ ウォッチドッグ タイマ (DSWDT) のタイムアウト
- ・ 超低消費電力復帰 (ULPWU) イベント

ディープスリープ モードでは、デバイスのリアルタイム クロック / カレンダ (RTCC) の動作を維持できます (クロックサイクルは失われません)。

デバイスは、電圧とタイムアウト イベントを監視するために、ディープスリープ専用のディープスリープ ブラウンアウト リセット (DSBOR) とディープスリープウォッチドッグ タイマ リセット (DSWDT) を備えます。 DSBOR と DSWDT は、他の電源管理モード (スリープ、アイドル、Doze) で使用する標準の BOR および WDT とは独立した機能です。

10.2.4.1 ディープスリープ モードへの移行

DSCON レジスタの DSEN ビットをセットしてからスリープコマンド (PWRSAV #SLEEP_MODE) を実行すると、デバイスはディープスリープ モードに移行します。 DSEN ビットをセットするにはロック解除シーケンスが必要です。 DSEN ビットをセットしてから SLEEPコマンドを実行するまでの時間に制限はありません。ディープスリープ モードが終了すると、DSEN ビットは自動的にクリアされます。

Note: ディープスリープから復帰した後に再び ディー プス リープに 移 行 す る 場 合、 RELEASE ビットをクリアした後に 3TCY 以上の遅延が必要です。

以下のシーケンスにより、ディープスリープ モードに 移行できます。

- 1. アプリケーションでディープスリープ WDT を使用する場合、これを有効化し、そのクロック源を設定します。ディープスリープ WDT の詳細は10.2.4.5「ディープスリープ WDT」を参照してください。
- 2. アプリケーションでディープスリープBORを使用 する場合、これを有効化します (DSLPBOR コン フィグレーション ビット(FDS<6>)をセットする)。
- 3. アプリケーションで RTCC アラームによるディー プスリープからの復帰を使用する場合、RTCC モジュールを有効化して設定します。RTCC の詳細は19.0「リアルタイムクロック/カレンダ(RTCC)」を参照してください。
- 4. 必要に応じて、重要なアプリケーション コンテクスト データを DSGPR0 および DSGPR1 レジスタに書き込んで保存します (必須ではありません)。
- 5. DSEN ビット (DSCON<15>) をセットしてディー プスリープ モードを有効化します。

Note: DSEN ビットをセットするには、ロック 解除シーケンスが必要です。

6. PWRSAV #0命令を実行してディープスリープモードに移行します。

DSEN ビットをセットするたびに、DSWAKE レジス タの全てのビットが自動的にクリアされます。

DSEN ビットをセットするには、例 10-2 に示すロック解除シーケンスが必要です。

例 10-2: ロック解除シーケンス

//Disable Interrupts For 5 instructions
asm volatile("disi #5");
//Issue Unlock Sequence
asm volatile
mov #0x55, W0;
mov W0, NVMKEY;
mov #0xAA, W1;
mov W1, NVMKEY;
bset DSCON, #DSEN

10.2.4.2 ディープスリープモードの終了

ディープスリープ モードは下記のイベントのいずれか の発生時に終了します。

- VDD 電源による POR リセット: VDD 電圧を監視する DSBOR 回路を使用しない場合、外部 VDD 電圧は、POR 回路の本来のリセット電圧まで下げる必要があります。
- DSWDT タイムアウト: DSWDT タイマがタイムア ウトした時に、デバイスはディープスリープを終了 します。
- RTCC アラーム (RTCEN = 1 の場合)
- MCLR ピンによるリセット
- INTOピン割り込み(ディープスリープモードへの移行前に割り込みが有効化されていた場合): 極性の設定により、ディープスリープモードの終了をトリガするピンのアサートレベル(0または1)を指定します。ディープスリープモードを終了するには、ディープスリープモード中にINTOピンの状態変化が必要です。

Note: ディープスリープ モードへの移行時に保留中の割り込みは全てクリアされます。

一般的にディープスリープ モードの終了はデバイスの 状態を保持しません (デバイスのパワーオン リセット (POR) と同じ)。例外として、復帰中も動作を維持する RTCC(使用する場合)と、DSGPRx レジスタおよび DSWDT は保持されます。

ディープスリープ終了後、POR シーケンスが完了する前に発生した復帰イベントは無視され、DSWAKE レジスタにキャプチャされません。

ディープスリープ モードの終了シーケンスは下記の通りです。

- 1. 復帰イベント発生後、デバイスはディープスリープ を終了して POR を実行します。DSEN ビットは 自動的にクリアされます。コード実行はリセット ベクタ位置から再開します。
- 2. デバイスがディープスリープを終了したかどうかを判定するために、ディープスリープ ビット DPSLP (RCON<10>) を読み出します。このビットは、ディープスリープ モードが終了した場合にセットされます。このビットがセットされていれば、クリアします。
- 3. 復帰要因を判定するために、DSWAKE レジスタ を参照します。
- 4. ディープスリープモード中にDSBORイベントが 発生したかどうかを判定するために、DSBOR ビット(DSCON<1>)を参照します。
- アプリケーション コンテクスト データが保存されている場合、これを DSGPR0 および DSGPR1 レジスタから読み戻します。
- 6. RELEASE ビット (DSCON<0>) をクリアします。

10.2.4.3 DSGPR0/DSGPR1 レジスタによる コンテクスト データの保存

ディープスリープ モードの終了によって POR が発生するため、ほとんどの特殊機能レジスタは POR 時の既定値にリセットされます。加えて、ディープスリープモード中に VCORE 電源は供給されないため、データ RAM 内の情報は失われます。

ディープスリープの前に重要なデータを保存する必要があるアプリケーションでは、ディープスリープ汎用レジスタ (DSGPR0 と DSGPR1) またはデータEEPROM (利用可能な場合)を使用できます。他のSFRとは異なり、DSGPR0 および DSGPR1 レジスタの内容は、デバイスがディープスリープ モード中でも保持されます。ディープスリープ終了後、ソフトウェアでこれらのレジスタを読み出してから RELEASE ビット (DSCON0) をクリアする事により、データを復元できます。

10.2.4.4 ディープスリープ中の I/O ピン

ディープスリープ中、汎用 I/O ピンは以前の状態を保持し、セカンダリオシレータ (SOSC) は動作を続けます (有効化されている場合)。ディープスリープへ移行する前に入力として設定 (TRISx ビットをセット)されたピンは、ディープスリープ中にハイインピーダンスを維持します。ディープスリープへ移行する前に出力として設定 (TRISx ビットをクリア)されたピンは、ディープスリープ中も出力ピンとして機能します。このモード中、これらの出力ピンは、ディープスリープへの移行時に対応する LATx ビットで設定されていた出力レベルを駆動し続けます。

デバイスが復帰し、POR シーケンス後にアプリケーションコードの実行を再開した後も、全ての I/O ピンは以前の状態を保持します。ディープスリープ中に入力として設定されていたピンはハイインピーダンスを維持し、出力として設定されていたピンは以前の値を引き続き駆動します。復帰後、TRIS および LAT レジスタと SOSCEN ビット (OSCCON<1>) はリセットされます。ファームウェアがこれらのビットまたはレジスタを変更しても、I/O ピンは即座に設定された状態に変化しません。ファームウェアが RELEASE ビット (DSCON<0>)をクリアすると I/O ピンが「リリース」されます。これにより、I/O ピンは、それぞれの TRIS および LAT ビット値が指定する状態に変化します。

従って復帰後もSOSCの動作を維持するには、RELEASE ビットをクリアする前に SOSCEN ビットをセットする 必要があります。

ディープスリープ BOR (DSBOR) を有効化した場合、ディープスリープ中に DSBOR イベントまたは通常の POR イベントが発生すると、I/O ピンは即座にリリース されます (RELEASE ビットをクリアした場合と同様)。以前の状態に関する情報は、汎用 DSGPR0 および DSGPR1 レジスタの内容も含め全て失われます。

ディープスリープ中にMCLRリセットイベントが発生した場合、DSGPRx、DSCON、DSWAKE レジスタは有効なままであり、RELEASE ビットはセットされた状態を保持します。SOSC の状態も保持されます。しかし I/Oピンは MCLR リセット状態にリセットされます。RELEASE ビットがセットされたままであるため、SOSCENビット(OSCCON<1>)を変更しても、RELEASEビットがクリアされるまで効果を持ちません。

他の全てのディープスリープ 復帰条件においても、アプリケーション ファームウェアは I/O ピンを再設定するために RELEASE ビットをクリアする必要があります。

10.2.4.5 ディープスリープ WDT

ディープスリープ モード中の DSWDT を有効にするには、コンフィグレーション ビット DSWDTEN (FDS<7>)をセットする必要があります。DSWDT は、デバイスのウォッチドッグ タイマ (WDT) が有効でなくても機能します。ディープスリープ モードに移行すると、DSWDTは自動的リセットされます。

DSWDT のクロック源は、DSWCKSEL コンフィグレーション ビット (FDS<4>) で選択します。ポストスケーラオプションは、DSWDTPS<3:0> コンフィグレーションビット (FDS<3:0>) で設定します。これにより、タイムアウト期間を 2.1 ms から 25.7 日の範囲で設定できます。FDSコンフィグレーション レジスタと DSWDTコンフィグレーション オプションの詳細は 26.0「その他の特殊な機能」を参照してください。

10.2.4.6 ディープスリープモード中のクロック 切り換え

RTCC と DSWDT は、いずれも SOSC または LPRC クロック源を使用して動作できます。LPRC と SOSC の両方を有効化しなくても、RTCC と DSWD の両方が動作できるため、消費電力を低減できます。

RTCC を LPRC で動作させると、RTCC の精度が $5 \sim 10\%$ 悪化します。これより高い精度の RTCC が必要な場合、SOSC クロック源を使用する必要があります。RTCC のクロック源は、RTCOSC コンフィグレーション ビット (FDS<5>) で選択します。

特定の状況では、ディープスリープモードへの移行時に DSWDT クロック源が無効化される可能性があります。このような場合でも、DSWDT が有効化されていれば、そのクロック源が自動的に有効化されます (ソフトウェアの介入は不要)。ただし、これが原因で DSWDT カウンタの開始が遅れる事があります。SOSC をクロック源として使用する場合、ディープスリープモードに移行する前にアプリケーションで SOSC を有効にする事により、この遅延を回避できます。

10.2.4.7 ディープスリープのステータスの チェックとクリア

ディープスリープ モードへの移行時に、ステータスビット DPSLP (RCON<10>) がセットされます。このビットはソフトウェアでクリアする必要があります。

ソフトウェアは、パワーアップ時にこのステータスビットを読み出して、リセットの原因がディープスリープモードの終了によるものかどうかを判定し、セットされていればクリアする必要があります。DPSLP ビットとPOR ビットの状態の組み合わせは4通り存在しますが、そのうち下記の3条件が考えられます。

- DPSLP ビットと POR ビットの両方がクリアされている場合、リセットはディープスリープモードの終了以外の原因によって発生した事を意味します。
- DPSLP ビットがクリアされ、POR ビットがセット されている場合、通常の POR が発生した事を意味 します。
- DPSLP ビットと POR ビットの両方がセットされている場合、ディープスリープ モードに移行した後に、デバイスがパワーダウンし、ディープスリープモードが終了した事を意味します。

10.2.4.8 パワーオン リセット (POR)

PORを生成するために VDD 電圧を監視します。ディープスリープの終了は POR と同様の挙動を示すため、10.2.4.7「ディープスリープのステータスのチェックとクリア」に記載した方法でディープスリープの終了イベントと通常の POR イベントを識別する必要があります。

通常の POR が発生した場合、全てのディープスリープロジック (ディープスリープレジスタ RTCC、DSWDT等)を含むデバイス全体がリセットされます。

10.2.4.9 ディープスリープ シーケンスのまとめ ディープスリープ モードへの移行と終了に必要な手順を以下に要約します。

- デバイスはリセット終了後アプリケーション コードの実行を開始します。
- 2. DSWDT機能が必要な場合、適切なコンフィグレーション ビットを設定します。
- 3. DSWDT と RTCC に対して適切なクロックを選択します (必須ではない)。
- 4. DSWDTを有効にして設定します(必須ではない)。
- 5. RTCC を有効にして設定します (必須ではない)。
- 6. コンテクスト データを DSGPRx レジスタに書き 込みます (必須ではない)。
- 7. INTO 割り込みを有効にします (必須ではない)。
- 8. DSCONレジスタのDSENビットをセットします。
- 9. PWRSV #SLEEP_MODE 命令を実行してディープス リープ モードに移行します。
- 10. 復帰イベントが発生すると、デバイスはディープ スリープを終了します。
- 11. DSEN ビットは自動的にクリアされます。
- 12. RCON の DPSLP ステータスビットと DSWAKE ステータスビットを読み出してクリアします。
- 13. DSGPRx レジスタを読み出します(必須ではない)。
- 14. コンフィグレーションに関連する全ての状態が完 全になったら RELEASE ビットをクリアします。
- 15. アプリケーションは通常動作を再開します。

レジスタ 10-1: DSCON: ディープスリープ制御レジスタ (1)

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
DSEN	_	_	_	_	_	_	RTCCWDIS
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/C-0, HS
	_	_	_	_	ULPWUDIS	DSBOR ⁽²⁾	RELEASE
bit 7							bit 0

凡例:C=クリア可能ビットHS=ハードウェア セット可能ビットR=読み出し可能ビットW=書き込み可能ビットU=未実装ビット、「0」として読み出し-n=POR 時の値1=ビットをセット0=ビットをクリアx=ビットは未知

bit 15 **DSEN**: ディープスリープ イネーブルビット

1 = PWRSAV #0 実行時にディープスリープに移行する 0 = PWRSAV #0 実行時に通常のスリープに移行する

bit 14-9 **未実装:**「0」として読み出し

bit 8 RTCCWDIS: RTCC 復帰ディセーブル ビット

1 = RTCC 無効でディープスリープから復帰する 0 = RTCC 有効でディープスリープから復帰する

bit 7-3 **未実装:**「0」として読み出し

bit 2 **ULPWUDIS:** ULPWU 復帰ディセーブル ビット

1 = ULPWU 無効でディープスリープから復帰する 0 = ULPWU 有効でディープスリープから復帰する

bit 1 **DSBOR**: ディープスリープ BOR イベントビット (2)

1 = ディープスリープ中に DSBOR はアクティブであり、BOR イベントが検出された

0 = ディープスリープ中に DSBOR はアクティブではなかった、または、アクティブであったが BOR は 検出されなかった

bit 0 RELEASE: I/O ピン状態リリースビット

1 = ディープスリープからの復帰時に I/O ピンは移行前の状態を保持する

0 = I/Oピンをディースリープ移行前の状態からリリースし、対応するTRISおよびLATビットでI/Oピンの状態を制御する

Note 1: ディープスリープ モードとは無関係の通常の POR イベントが発生すると、全てのレジスタビットがリセットされます。

2: 他のイベントとは異なり、ディープスリープ BOR イベントはディープスリープからの復帰をトリガしません。これは POR を生成します。

レジスタ 10-2: DSWAKE: ディープスリープ 復帰要因レジスタ ⁽¹⁾

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0, HS
_	_	_	_	_	_	_	DSINT0
bit 15							bit 8

R/W-0, HS	U-0	U-0	R/W-0, HS	R/W-0, HS	R/W-0, HS	U-0	R/W-0, HS
DSFLT	_	_	DSWDT	DSRTCC	DSMCLR	_	DSPOR ^(2,3)
bit 7							bit 0

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-9 **未実装:**「0」として読み出し

bit 8 DSINTO: チャージ割り込みビット

1=ディープスリープ中にチャージ割り込みがアサートされた

0 = ディープスリープ中にチャージ割り込みはアサートされなかった

bit 7 DSFLT: ディープスリープ フォルト検出ビット

1 = ディープスリープ中に異常が発生し、一部のディープスリープ コンフィグレーション設定が破損

した可能性がある

0 = ディープスリープ中に異常は発生しなかった

bit 6-5 **未実装:**「0」として読み出し

bit 4 **DSWDT**: ディープスリープ ウォッチドッグ タイマ タイムアウト ビット

1=ディープスリープ中にディープスリープ ウォッチドッグ タイマがタイムアウトした

0 = ディープスリープ中にディープスリープ ウォッチドッグ タイマはタイムアウトしなかった

bit 3 DSRTCC: リアルタイム クロック / カレンダ (RTCC) アラームビット

1 = ディープスリープ中にリアルタイム クロック / カレンダがアラームをトリガした

0 = ディープスリープ中にリアルタイム クロック / カレンダはアラームをトリガしなかった

bit 2 **DSMCLR**: MCLR イベントビット

1 =ディープスリープ中にアクティブであった $\frac{\overline{MCLR}}{\overline{MCLR}}$ ピンがアサートされた

0 = ディープスリープ中に MCLR ピンはアクティブではなかった、または、アクティブであったがア

サートされなかった

bit 1 **未実装:**「0」として読み出し

bit 0 **DSPOR:** パワーオンリセット イベントビット (2,3)

1 = アクティブであった VDD 電源 POR 回路が POR イベントを検出した

0 = VDD 電源 POR 回路がアクティブではなかった、または、アクティブであったが POR イベントを検出しなかった

Note 1: 全てのレジスタビットは、DSEN (DSCON<15>) ビットがセットされた時にクリアされます。

2: DSPOR ビット以外の全てのレジスタビットは、ディープスリープ モードとは無関係の通常の POR イベント時にリセットされます。 DSPOR ビットはディープスリープ中に発生する POR イベントではリセットされません。

3: このビットは、このレジスタ内の他のビットとは異なり、ディープスリープとは無関係の通常の POR 時にセットされます。

10.3 超低消費電力復帰

RBOピンによる超低消費電力復帰 (ULPWU) は、コンデンサの放電によるゆっくりとした電圧の低下を利用して、余分な電流を消費せずに割り込みを周期的に生成する事ができます。

この機能の使用方法は下記の通りです:

- 1. RBO ピンを出力として設定し、状態を「1」に設 定する事により、RBO ピンのコンデンサを充電し ます。
- 2. RB0 を入力として設定する事により、コンデンサ の充電を停止します。
- ULPWCON レジスタの ULPEN および ULPSINK ビットをセットする事により、コンデンサを放電 します。
- 4. スリープモードを設定します。
- 5. スリープモードに移行します。

RBOピンの電圧が VIL よりも低下すると、デバイスが復帰して次の命令を実行します。

この機能を使用すると、デバイスをスリープモードから周期的に復帰する場合の消費電力を低減できます。

タイムアウトは、RBO ピンの RC 回路の放電時間によって決まります。

ULPWU モジュールがデバイスをスリープモードから 復帰すると、ULPWUIF ビット (IFS5<0>) がセットさ れます。ソフトウェアは、復帰時にこのビットを読み 出する事によって、復帰要因を特定できます。

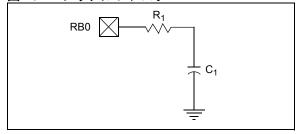
ULPWU モジュールを初期化するサンプルコードを 例 10-3 に示します。

例 10-3: 超低消費電力復帰の初期化

```
//**********
// 1.Charge the capacitor on RB0
//**************
   TRISBbits.TRISB0 = 0;
   LATBbits.LATB0 = 1;
   for(i = 0; i < 10000; i++) Nop();
//*********
//2.Stop Charging the capacitor
  on RB0
//*********
   TRISBbits.TRISB0 = 1;
//********
//3.Enable ULPWU Interrupt
//********
IFS5bits.ULPWUIF = 0;
IEC5bits.ULPWUIE = 1;
IPC21bits.ULPWUIP = 0x7;
//********
//4.Enable the Ultra Low Power
  Wakeup module and allow
   capacitor discharge
   ULPWCONbits.ULPEN = 1;
   ULPWCONbit.ULPSINK = 1;
//********
//5.Enter Sleep Mode
//***********
   Sleep();
//for sleep, execution will
//resume here
```

RBO ピンと外部コンデンサの間の直列抵抗は、 RB0/AN0/ULPWU ピンの過電流保護を提供するとと もに、タイムアウトのソフトウェア校正を可能にしま す(図 10-1 参照)。

図 10-1: シリアルレジスタ



タイマを使用してコンデンサの充放電時間を計測できます。計測結果に基づいて、目標とするスリープ遅延時間が得られるように充電時間を調整できます。この方法により温度、電圧、部品精度の影響を補正できます。この周辺モジュールは、簡単なプログラマブル低電圧検出器 (LVD) または温度センサとして構成する事もできます。

レジスタ 10-3: ULPWCON: ULPWU 制御レジスタ (1)

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0
ULPEN	_	ULPSIDL	_	_	_	_	ULPSINK
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
_	_	_	_	_	_	_	_
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し -n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15 ULPWUI モジュール イネーブルビット

1 = モジュールを有効にする 0 = モジュールを無効にする

bit 14 **未実装:**「0」として読み出し

bit 13 **ULPSIDL:** ULPWU アイドル時停止選択ビット

1 = デバイスがアイドルモードに移行した時にモジュールの動作を停止する

0=アイドルモード時もモジュールの動作を継続する

bit 12-9 **未実装:**「0」として読み出し

bit 8 ULPSINK: ULPWU 電流シンク イネーブルビット

1 = 電流シンクを有効にする
0 = 電流シンクを無効にする

bit 7-0 **未実装:**「0」として読み出し

10.4 電圧レギュレータの省電力機能

PIC24FV32KA304 シリーズのデバイスが内蔵する電圧レギュレータは、省電力化のために動作モードを変更できます。内蔵レギュレータは、高電圧レギュレータ (HVREG) と低電圧レギュレータ (LVREG) で構成されます。HVREG と LVREG の組み合わせにより、下記の電源モードを利用できます。

10.4.1 ランモード

ランモードでは、メイン HVREG が安定化電圧を供給します。このモードは、デバイスがスリープまたはディープスリープ モードではない時に、フルスピードで動作するのに十分な電流を供給します。

10.4.2 高速復帰スリープモード

高速復帰スリープモードでは、デバイスがスリープモードの時に、メイン HVREG がランモードと同じ高電流で安定化電圧を供給します。このモードでは、スリープ時の消費電力が最も高くなりますが、スリープから最も高速に復帰できます。

10.4.3 スリープ(スタンバイ)モード

スリープモードでは、デバイスがスリープモードの時に、メイン HVREG が低い(スタンバイ)電流で安定化電圧を供給します。このモードでは、供給電流が低いために機能が限定されます。このモードは高速復帰スリープモードよりも低消費電力ですが、スリープからの復帰により長い時間を要します。

10.4.4 低電圧スリープモード

低電圧スリープモードでは、デバイスがスリープモードの時に、LVREGが全ての安定化電圧を供給します。このモードでは、スリープ時の消費電力は最低となりますが、利用可能な機能も最も制限されます。低電圧スリープモードでは、復帰時に VCORE 電源レールが通常の電圧レベルに復帰するまでに追加の時間が必要であるため、通常のスリープモードよりも復帰に時間がかかります。

Note: PIC24F32KA30Xファミリのデバイスは電 圧レギュレータを内蔵しないため、低電圧 スリープモードをサポートしません。

10.4.5 ディープスリープ モード

ディープスリープ モードでは、メイン HVREG と LVREG の両方がシャットダウンするため、デバイスの消費電力を最低限に抑えられます。しかし、このモードではデバイスの機能を維持できず、復帰には最も長い時間を要します。

表 10-1: PIC24FV32KA304 デバイスの電圧管理コンフィグレーション設定

LVRCFG ビット (FPOR<2>)	LVREN ビット (RCON<12>)	PMSLP ビット (RCON<8>)	スリープ時の 電源モード	内容
0	0	1	高速復帰	スリープ時も通常動作時の HVREG モードから変更 しない
			スリープ	LVREG を使用しない
0	0	0	スリープ	スリープ時に HVREG は低消費電力スタンバイモー ドに移行する
			(スタンバイ)	LVREG を使用しない
0	1	0	低電圧	スリープ時に HVREG は OFF
			スリープ	LVREG がスリープ電圧を供給する
1	Х	1	高速復帰	スリープ時も通常動作時の HVREG モードから変更 しない
			スリープ	LVREG は常時 OFF
1	Х	0	スリープ	スリープ時に HVREG は低消費電力スタンバイモー ドに移行する
			(スタンバイ)	LVREG は常時 OFF

10.5 Doze モード

クロック周波数の低減と命令ベースの省電力モードは、一般的に使用できる効果的な省電力方式です。しかし、そのような方法が実用的ではない状況も存在します。例えば、他に何も実行していない時でも同期通信を常時継続する事が必要なアプリケーションも存在します。このようなアプリケーションでシステムクロックを下げると、通信エラーが発生する可能性があります。さらに、省電力モードを使用すると通信が完全に停止する可能性があります。

Doze モードは、コード実行を継続しながら消費電力を低減できる、単純かつ効果的な方法です。このモードでは、システムクロックは通常動作時と同じクロック源を使用し、同じ速度で動作を継続します。これにより、周辺モジュールのクロック速度を変更せずに、CPU のクロック速度だけを低減できます。2 つのクロックドメイン間の同期は維持されるため、CPU のコード実行は低速でも、周辺モジュールは SFR にアクセスできます。

Doze モードは、DOZEN ビット (CLKDIV11) をセット する事で有効にできます。周辺モジュールとコアのクロック速度比は、DOZE<2:0> ビット (CLKDIV<14:12>) で指定します。可能な設定は 8 通り (1:1 ~ 1:128) で、既定値は 1:1 です。

イベント駆動型アプリケーションでは、Doze モードを使用して選択的に消費電力を低減する事もできます。これにより、同期通信等のクロックが重要な機能を中断する事なく継続できます。この間、CPU はアイドリングし、何らかのイベントが割り込みルーチンを呼び出すまで待機します。割り込み発生時のフルスピードCPU 動作への自動復帰を有効にするには、ROI ビット(CLKDIV<15>)をセットします。既定値では、割り込みイベントは Doze モードの動作に影響しません。

10.6 選択的な周辺モジュールの制御

アイドルおよび Doze モードでは、CPU クロックの低速化または停止によって消費電力を大幅に低減できます。しかし、周辺モジュールにクロックを供給し続けるために電力を消費します。これらのモードでは提供できない機能をアプリケーションが要求する場合もあります(例:周辺モジュールの消費電力を最小限にするとともに CPU へ電源リソースを割り当てる)。

このような要求に応えるために、PIC24Fでは、個々の周辺モジュールを選択的に無効にする事により、周辺モジュールの消費電力を低減または排除する事ができます。これには下記の2つの制御ビットを使用します。

- 各周辺モジュールのメイン制御SFR内にある周辺モジュール イネーブルビット (一般的にビット名は「XXXEN」)
- いずれかの PMD 制御レジスタ内にある周辺モジュールディセーブル(PMD)ビット(一般的にビット名は「XXXMD」)

これらのビットは、対応するモジュールを有効化または無効化します。あるモジュールの PMD ビットをセットすると、そのモジュールへの全てのクロック源が無効化され、そのモジュールの消費電力は絶対的最小レベルまで低下します。この状態では、その周辺モジュールに関連する制御 / ステータス レジスタも無効化されるため、それらのレジスタへの書き込みは効果を持たず、読み出し値は無効です。多くの周辺モジュールがPMD ビットを備えています。

一方、XXXEN ビットをクリアする事によって周辺モジュールを無効化した場合、対応する周辺モジュールの機能は無効化されますが、そのモジュールのレジスタには読み書き可能です。この場合の消費電力低減効果は、PMD ビットを使用した場合よりも少なくなります。キャプチャ、コンペア、RTCC を除くほとんどの周辺モジュールはイネーブルビットを備えます。

また、デバイスがアイドルモードへ移行した時に、周辺モジュールを選択的に無効にする事により、さらに省電力化を図れます。これには「XXXIDL」の名前を持つ制御ビットを使用します。既定値では、アイドルモード時に動作可能な全てのモジュールが動作します。周辺モジュールのアイドルモード時無効化機能を使用すると、アイドルモード中の消費電力をさらに低減でき、低消費電力が重視されるアプリケーションの省電力機能を拡張できます。

NOTE:

12.0 TIMER1

Note: 本書は、

本書は、PIC24 デバイスの中の特定製品 グループを対象とし、それらの機能の概要説明を目的としています。従って本書は、包括的な参照資料の提供を意図したものではありません。タイマの詳細は「PIC24Fファミリリファレンス マニュアル」のセクション 14.「タイマ」(DS39704)を参照してください。

Timer1 モジュールは、リアルタイム クロック (RTC) 用のタイムカウンタまたはフリーランニングのインターバルタイマ/カウンタとして使用可能な 16 ビットタイマです。Timer1 は以下のいずれかのモードで動作できます。

- ・ 16 ビットタイマ
- ・ 16 ビット同期カウンタ
- ・ 16 ビット非同期カウンタ

Timer1 は下記の機能もサポートします。

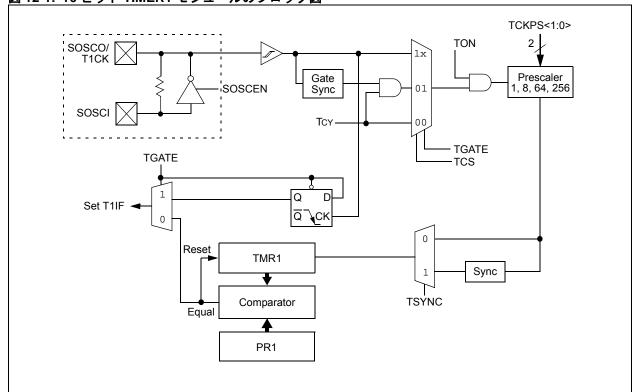
- タイマゲート動作
- 選択可能なプリスケーラ設定
- CPU のアイドルおよびスリープモード中のタイマ 動作
- 16 ビット周期レジスター致時の割り込み、または、 外部ゲート信号立ち下がりエッジでの割り込み

図 12-1 に 16 ビット Timer1 モジュールのブロック図を示します。

Timer1 動作の設定方法:

- 1. TON ビットをセット (= 1) する。
- TCKPS<1:0>ビットでタイマのプリスケーラ比を 選択する。
- 3. TCS ビットと TGATE ビットでクロックモードと ゲートモードを設定する。
- 4. TSYNCビットのセット/クリアにより同期または 非同期動作を設定する。
- 5. タイマ周期値を PR1 レジスタに書き込む。
- 6. 割り込みが必要な場合、割り込みイネーブルビット T1IE をセットし、優先度ビット T1IP<2:0> で割り込み優先度を設定する。

図 12-1: 16 ビット TIMER1 モジュールのブロック図



レジスタ 12-1: T1CON: TIMER1 制御レジスタ

R/W-0	U-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
TON	_	TSIDL	_	_	_	T1ECS1 ⁽¹⁾	T1ECS0 ⁽¹⁾
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	U-0
_	TGATE	TCKPS1	TCKPS0	_	TSYNC	TCS	_
bit 7							bit 0

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15 TON: Timer1 ON ビット

1 = 16 ビット Timer1 を起動する 0 = 16 ビット Timer1 を停止する

bit 14 **未実装:**「0」として読み出し

bit 13 TSIDL: アイドルモード時停止ビット

1 = デバイスがアイドルモードに移行した時にモジュールの動作を停止する

0=アイドルモード時もモジュールの動作を継続する

bit 12-10 **未実装:**「0」として読み出し

bit 9-8 T1ECS <1:0>: Timer1 拡張クロック選択ビット (1)

11 = 予約済み (使用せず)

10 = Timer1 のクロック源に LPRC を使用する

01 = Timer1 のクロック源に T1CK からの外部クロックを使用する

00 = Timer1 のクロック源にセカンダリ オシレータ (SOSC) を使用する

bit 7 **未実装:**「0」として読み出し

bit 6 TGATE: Timer1 ゲート時間積算イネーブルビット

TCS = 1 の時: このビットを無視 TCS = 0 の時:

1 = ゲート時間積算を有効にする 0 = ゲート時間積算を無効にする

bit 5-4 **TCKPS<1:0>:** Timer1 入力クロック プリスケール選択ビット

11 = 1:256 10 = 1:64 01 = 1:8 00 = 1:1

bit 3 **未実装:**「0」として読み出し

bit 2 TSYNC: Timer1 外部クロック入力同期選択ビット

TCS = 1 の時:

1 = 外部クロック入力を同期する 0 = 外部クロック入力を同期しない

<u>TCS = 0 の時 :</u> このビットを無視

bit 1 TCS: Timer1 クロック源選択ビット

1 = T1ECS<1:0> で選択した Timer1 クロック源を使用する

0 = 内部クロック (Fosc/2) を使用する

bit 0 **未実装:**「0」として読み出し

Note 1: T1ECS ビットは TCS = 1 の時にのみ有効です。

1/0 ポート 11.0

本書は、PIC24 デバイスの中の特定製品 Note: グループを対象とし、それらの機能の概 要説明を目的としています。従って本書 は、包括的な参照資料の提供を意図した ものではありません。I/O ポートの詳細 は「PIC24F ファミリ リファレンス マ ニュアル」のセクション 12.「ペリフェ ラル ピンセレクト (PPS) を備えた I/O **ポート」**(DS39711) を参照してください。

PIC24FV32KA304 ファミリのデバイス

はペリフェラル ピンセレクト機能をサ

ポートしない事に注意してください。

VDD と VSS を除く全てのデバイスピンは、周辺モ ジュールとパラレル I/O ポートの間で共有されます。 全ての I/O 入力ポートは、ノイズ耐性の向上のために シュミットトリガ入力を備えます。

パラレル I/O (PIO) ポート 11.1

周辺モジュールとピンを共有するパラレル I/O ポート では、一般的に周辺モジュールの方が優先されます。 周辺モジュールの出力バッファデータと制御信号は、 一対のマルチプレクサに供給されます。マルチプレク サは、周辺モジュールと関連ポートのどちらに I/O ピ ンの出力データと制御信号の所有権を持たせるのかを 選択します。このロジックは、ポートのデジタル出力 が同じピンを共有する周辺モジュールの入力を駆動し てしまう「ループスルー」も回避します。図 11-1 に、 ポートと周辺モジュールが 1 つの I/O ピンをどのよう に共有するのかを示しします。

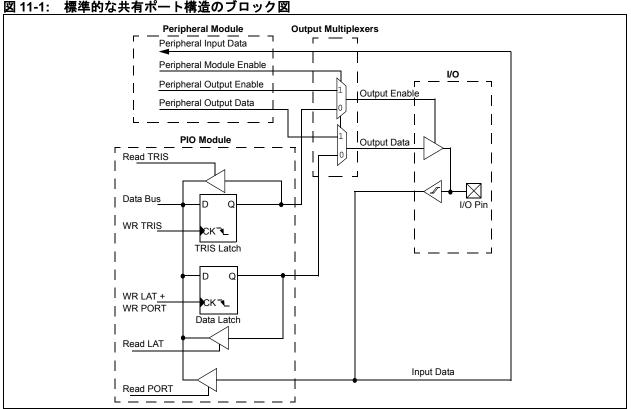
有効化された周辺モジュールが割り当てられたピンを アクティブに駆動している場合、そのピンの汎用出力 ピンとしての使用は無効化されます。I/O ピンの読み 出しは可能ですが、パラレルポート ビット用の出力ド ライバは無効化されます。周辺モジュールが有効化さ れていても、ピンをアクティブに駆動していない場合、 ポートはそのピンを駆動できます。

全てのポートピンは、デジタル I/O としての動作に直 接関連する3つのレジスタを備えます。データ方向レ ジスタ (TRISx) は、そのピンが入力なのか出力なのか を決定します。データ方向ビットが「1」の場合、そ のピンは入力です。リセット時に、全てのポートピン は入力ピンとして設定されます。 データラッチ レジス タ (LATx) を読み出すと、ラッチからデータが読み出さ れます。データラッチ レジスタに書き込むと、その データはラッチに書き込まれます。ポート (PORTx) か らの読み出しはポートピンを読み出しますが、ポート ピンへの書き込みはラッチに書き込まれます。

特定のデバイスで有効ではないビットと、そのビット に関連するデータおよび制御レジスタは全て無効化さ れます。つまり、対応する LATx レジスタ、TRISx レ ジスタ、ポートピンはゼロとして読み出されます。

あるピンが入力専用として定義された周辺モジュール または機能と共有されている場合、他に競合する出力 ソースが存在しないため、そのピンはポート専用であ るとみなされます。

ディープスリープ中、I/O ピンは状態を保 Note: 持します。この状態は、復帰時にソフト ウェア復元ビット(RELEASE)がクリアさ れるまで保持されます。



11.1.1 オープンドレイン コンフィグレーション

PORT、LAT、TRIS レジスタによるデータ制御に加えて、各ポートピンを個別にデジタル出力またはオープンドレイン出力用に設定する事もできます。この設定には、各ポートに割り当てられたオープンドレイン制御レジスタ (ODCx) を使用します。これらのビットをセットすると、対応するピンはオープンドレイン出力として動作します。

最大許容オープンドレイン電圧は、VIH の最大仕様値 と同じです。

11.2 アナログ ポートピンの設定

ANS および TRIS レジスタは、A/D ポートピンの動作を制御します。ポートピンをアナログ入力として動作させる場合、対応する TRIS ビットをセット(入力として設定)する必要があります。TRIS ビットがクリア(出力として設定)されている場合、デジタル出力レベル (VOH または VOL) が変換されます。

PORT レジスタを読み出すと、アナログ入力チャンネルとして設定されている全てのピンが、クリア状態(LOW レベル)とし読み出されます。デジタル入力と

して定義されたピン (ANx ピンを含む) にアナログレベル電圧を印加すると、入力バッファがデバイス仕様を超える電流を消費する原因になります。

11.2.1 アナログ選択レジスタ

ADC 入力やコンパレータ入力等のアナログ機能を共有する I/O ピンでアナログ機能を使用する場合、それらのピンのデジタル入力を遮断する必要があります。ピンをアナログ入力として使用する場合、そのピンには外部からアナログ電圧が印加されるという事に注意してください。

アナログ機能の制御には ANSx レジスタを使用します。各ポートにそれぞれ1つの ANS レジスタ (ANSA、ANSB、ANSC) が割り当てられています。各 ANSx レジスタは、アナログ機能とデジタル I/O 機能を共有する各ピンに対応するビットを格納します。

アナログ機能を持たないピンに対応するビットは未実装となります。レジスタ 11-1 ~レジスタ 11-3 を参照してください。

レジスタ 11-1: ANSA: アナログ選択 (PORTA)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
_	_		_	_		_	_
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1
_	_	_	_	ANSA3	ANSA2	ANSA1	ANSA0
bit 7							bit 0

凡例: U = 未実装ビット、「0」として読み出し

R=読み出し可能ビット W=書き込み可能ビット HSC=ハードウェア セット / クリア可能ビット -n = POR 時の値 1=ビットをセット 0=ビットをクリア x=ビットは未知

bit 15-4 **未実装:**「0」として読み出し

bit 3-0 ANSA<3:0>: アナログ選択制御ビット

1 = デジタル入力バッファを有効にしない(ピンをアナログ入力用に使用する)

0 = デジタル入力バッファを有効にする

レジスタ 11-2: ANSB: アナログ選択 (PORTB)

R/W-1	R/W-1	R/W-1	R/W-1	U-0	U-0	U-0	U-0
ANSB15	ANSB14	ANSB13	ANSB12	_	_	_	_
bit 15							bit 8

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
_	_	_	ANSB4	ANSB3 ⁽¹⁾	ANSB2	ANSB1	ANSB0
bit 7							bit 0

凡例: U = 未実装ビット、「0」として読み出し

R=読み出し可能ビット W=書き込み可能ビット HSC=ハードウェア セット / クリア可能ビット -n = POR 時の値 1= ビットをセット 0= ビットをクリア x= ビットは未知

bit 15-12 ANSB<15:12>: アナログ選択制御ビット

1 = デジタル入力バッファを有効にしない(ピンをアナログ入力用に使用する)

0 = デジタル入力バッファを有効にする

bit 11-5 **未実装:**「0」として読み出し

bit 4-0 ANSB<4:0>: アナログ選択制御ビット

1 = デジタル入力バッファを有効にしない(ピンをアナログ入力用に使用する)

0 = デジタル入力バッファを有効にする

Note 1: 20 ピンデバイスでは利用できません。

レジスタ 11-3: ANSC アナログ選択 (PORTC)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0		
_	_	_	_	_	_	_	_		
bit 15 bit 8									

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-1	R/W-1
_	_	_	_	_	ANSC2 ⁽¹⁾	ANSC1 ⁽¹⁾	ANSC0 ⁽¹⁾
bit 7							bit 0

凡例: U = 未実装ビット、「0」として読み出し

R=読み出し可能ビット W=書き込み可能ビット HSC=ハードウェア セット / クリア可能ビット -n = POR 時の値 1= ビットをセット 0= ビットをクリア x= ビットは未知

bit 15-3 **未実装:**「0」として読み出し

bit 2-0 ANSC<2:0>: アナログ選択制御ビット

1 = デジタル入力バッファを有効にしない(ピンをアナログ入力用に使用する)

0 = デジタル入力バッファを有効にする

Note 1: 20/28 ピンデバイスでは利用できません。

11.2.2 I/O ポートの読み書きタイミング

あるポートで方向を変更するか書き込みを行った直後に同一ポートを読み出す場合、間に1命令サイクルを 挿入する必要があります。これには通常 NOP 命令を使 用します。

11.3 状態変化通知

I/Oポートの状態変化通知機能により、PIC24FV32KA304ファミリは、選択した入力ピンの状態変化 (COS) に対応してプロセッサに割り込みを要求できます。この機能を使用する事により、クロックが停止するスリープモード中でも、入力の状態変化を検出できます。状態変化時の割り込み要求の生成には、デバイスのピン数に応じて最大 23 個の外部信号 (CN0 ~ CN22)を選択(有効化)できます。

CN モジュールには 6 個の制御レジスタが割り当てられています。CNEN1 レジスタと CNEN2 レジスタは、各 CN 入力ピンの割り込みイネーブル制御ビットを格納します。これらのビットをセットすると、対応するピンの CN 割り込みが有効化されます。

また、各 CN ピンには数 $10 k\Omega$ 相当のプルアップ / プルダウン抵抗が接続されています。プルアップは、そのピンの電流ソースとして機能します。プルダウンは

電流シンクとして機能し、押しボタンまたはキーパッド デバイスを接続する際の外付け抵抗を不要にします。

各ピンでは、プルアップまたはプルダウンのいずれか一方の抵抗だけを有効にする必要があります。VDD に押しボタンまたはキーパッドを接続する場合、プルダウン抵抗を有効にします。それらを VSS に接続する場合、プルアップ抵抗を有効にします。プルアップは、各 CN ピンの制御ビットを収めた CNPU1 レジスタとCNPU2 レジスタを使用して個別に有効にします。

これらの制御ビットをセットすると、対応するピンのプルアップ抵抗が有効化されます。プルダウンは、各CN ピンの制御ビットを収めた CNPD1 レジスタとCNPD2 レジスタを使用して個別に有効にします。これらの制御ビットをセットすると、対応するピンのプルダウン抵抗が有効化されます。

内部プルアップを選択した場合、そのピンは VDD をプルアップ ソース電圧として使用します。内部プルダウンを選択した場合、そのピンは内部抵抗によって VSS にプルダウンされます。内部プルアップ / プルダウンを有効にする場合、外部のプルアップ ソース / プルダウンシンクが存在しない事を確認してください。

Note: ポートピンをデジタル出力として設定する場合、状態変化通知ピンのプルアップおよびブルダウンを無効にする必要があります。

例 11-1: ポート読み書きのサンプルコード

```
0xFF00. W0;
MOV
                           //Configure PORTB<15:8> as inputs and PORTB<7:0> as outputs
      WO, TRISB;
MOV
NOP;
                            //Delay 1 cycle
BTSS
     PORTB, #13;
                            //Next Instruction
Equivalent 'C' Code
TRISB = 0xFF00;
                           //Configure PORTB<15:8> as inputs and PORTB<7:0> as outputs
NOP();
                            //Delay 1 cycle
if(PORTBbits.RB13 == 1)
                           // execute following code if PORTB pin 13 is set.
```

13.0 TIMER2/3 ≥ TIMER4/5

Note: 本書は、PIC24 デバイスの中の特定製品 グループを対象とし、それらの機能の概要説明を目的としています。従って本書は、包括的な参照資料の提供を意図したものではありません。タイマの詳細は「PIC24Fファミリリファレンス マニュアル」のセクション 14.「タイマ」(DS39704)を参照してください。

Timer2/3 モジュールと Timer4/5 モジュールは 2 個の 32 ビットタイマですが、4 個の 16 ビットタイマとしても構成でき、別々に動作モードを選択できます。

Timer2/3 または Timer4/5 は、32 ビットタイマとして下記の 3 種類のモードで動作します。

- 2個の16ビットタイマ (Timer2と Timer3): 非同期 カウンタモードを除く全ての16ビット動作モード を別々に選択可能
- 1個の32ビットタイマ
- 1 個の 32 ビット同期カウンタ

これらは下記の機能もサポートします。

- タイマゲート動作
- 選択可能なプリスケーラ設定
- アイドルおよびスリープモード時のタイマ動作
- ・ 32 ビット周期レジスター致時の割り込み
- ・ ADC イベントトリガ

4個の16ビットタイマは、別々に同期タイマまたはカウンタとして機能できます。16ビットタイマも上記の機能を備えますが、ADCイベントトリガ機能はTimer3にのみ実装されます。動作モードと有効にする機能はT2CON、T3CON、T4CON、T5CON レジスタの適切なビットで設定します。T2CON、T3CON、T4CON、T5CON の説明をレジスタ 13-1 ~レジスタ 13-2 に記載します。

32 ビットタイマ / カウンタ動作の場合、Timer2 と Timer4 が 32 ビットタイマの下位ワード (Isw)、Timer3 と Timer5 が上位ワード (msw) です。

Note: 32 ビット動作では、T3CONまたはT5CON 制御ビットを無視し、T2CON または T4CON制御ビットだけを設定と制御用に 使用します。32 ビットタイマ モジュール は Timer2 または Timer4 のクロックおよびゲート入力を使用し、割り込み生成用に Timer3 または Timer5 の割り込みフラグを 使用します。

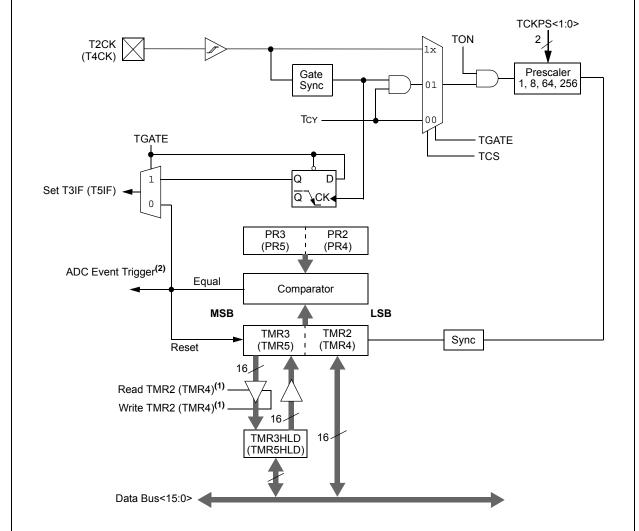
Timer2/3 または Timer4/5 を 32 ビット動作用に設定する手順は下記の通りです。

- T32 ビット (T2CON<3> または T4CON<3>) を「1」 にセットする。
- TCKPS<1:0> ビットで Timer2 または Timer4 のプリスケーラ比を選択する。
- 3. TCS ビットと TGATE ビットでクロックモードと ゲートモードを設定する。
- 4. タイマ周期値を書き込む: PR3 (または PR5) は 32 ビット値の上位ワードを格納し、PR2 (または PR4) は下位ワードを格納します。
- 5. 割り込みが必要な場合、割り込みイネーブルビット TxIE をセットし、優先度ビット TxIP<2:0> で割り込み優先度を設定する。
- 6. TON ビット (TxCON<15>) を「1」にセットする。 任意時点のタイマ値はレジスタペア TMR3:TMR2 (または TMR5:TMR4) に格納されます。TMR3 (TMR5) はカウント値の上位ワードを格納し、TMR2 (TMR4) は下位ワードを格納します。

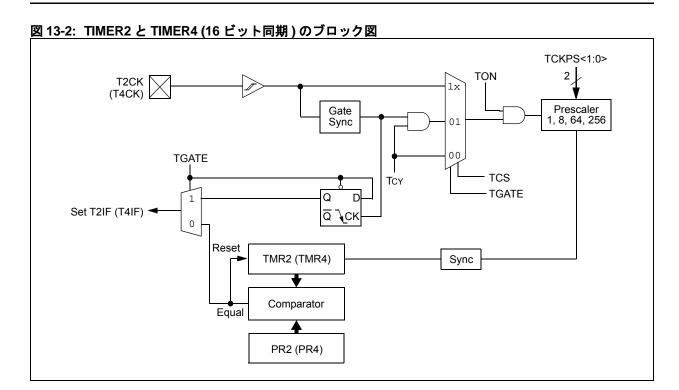
タイマを 16 ビット動作用に設定する手順は下記の通りです。

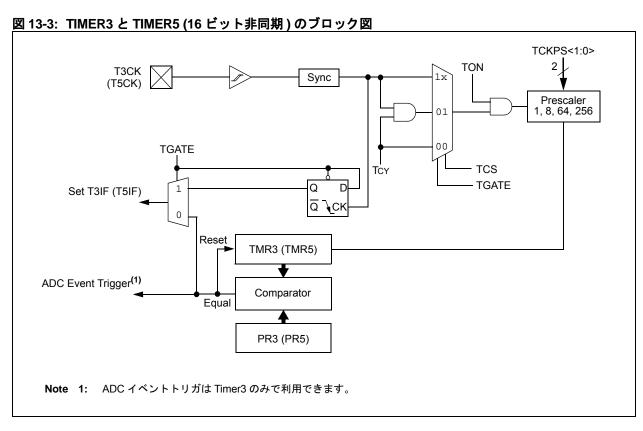
- そのタイマに対応する T32 ビット (Timer2 および Timer3 では T2CON<3>、Timer4 および Timer5 では T4CON<3>) をクリアする。
- 2. TCKPS<1:0>ビットでタイマのプリスケーラ比を 選択する。
- 3. TCS ビットと TGATE ビットでクロックモードと ゲートモードを設定する。
- 4. タイマ周期値を PRX レジスタに書き込む。
- 5. 割り込みが必要な場合、割り込みイネーブルビット TxIE をセットし、優先度ビット TxIP<2:0> で割り込み優先度を設定する。
- 6. TON ビット (TxCON<15>) を「1」にセットする。

図 13-1: TIMER2/3 と TIMER4/5 (32 ビット) のブロック図



- Note 1: 32 ビットタイマ / カウンタ動作を使用する場合、32 ビットタイマ コンフィグレーション ビット T32 をセットする必要があります。全ての制御ビットは T2CON レジスタと T4CON レジスタで別々に設定されます。
 - **2:** ADC イベントトリガは、32 ビットモードの場合 Timer2/3 のみ、16 ビットモードの場合 Timer3 のみで利用できます。





レジスタ 13-1: TxCON: TIMER2/TIMER4 制御レジスタ

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON	_	TSIDL	_	_	_	_	_
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	U-0
_	TGATE	TCKPS1	TCKPS0	T32 ⁽¹⁾	_	TCS	
bit 7							bit 0

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15 TON: Timerx ON ビット

TxCON<3> = 1 の時:

1 = 32 ビット Timerx/y を起動する 0 = 32 ビット Timerx/y を停止する

TxCON<3> = 0 の時:

1 = 16 ビット Timerx を起動する 0 = 16 ビット Timerx を停止する

bit 14 **未実装:**「0」として読み出し

bit 13 TSIDL: アイドルモード時停止ビット

1 = デバイスがアイドルモードに移行した時にモジュールの動作を停止する

0=アイドルモード時もモジュールの動作を継続する

bit 12-7 **未実装:**「0」として読み出し

bit 6 TGATE: Timerx ゲート時間積算イネーブルビット

TCS = 1 の時: このビットを無視 TCS = 0 の時:

1 = ゲート時間積算を有効にする 0 = ゲート時間積算を無効にする

bit 5-4 **TCKPS<1:0>:** Timerx 入力クロック プリスケール選択ビット

11 = 1:256 10 = 1:64 01 = 1:8 00 = 1:1

bit 3 **T32:** 32 ビット タイマモード選択ビット ⁽¹⁾

1 = Timer2/Timer3 または Timer4/Timer5 は 32 ビットタイマを形成する

0 = Timer2/Timer3 または Timer4/Timer5 は 2 個の 16 ビットタイマとして動作する

bit 2 **未実装:**「0」として読み出し

bit 1 TCS: Timerx クロック源選択ビット

1 = TxCK ピンからの外部クロック (立ち上がりエッジを使用)

0 = 内部クロック (Fosc/2)

bit 0 **未実装:**「0」として読み出し

Note 1: T3CON または T5CON 制御ビットの設定は、32 ビットモードのタイマ動作に影響しません。

レジスタ 13-2: TyCON: TIMER3/TIMER5 制御レジスタ

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON ⁽¹⁾	_	TSIDL ⁽¹⁾	_	_	_	_	_
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	U-0
_	TGATE ⁽¹⁾	TCKPS1 ⁽¹⁾	TCKPS0 ⁽¹⁾	_	_	TCS ⁽¹⁾	_
bit 7							bit 0

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「O」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15 **TON:** Timery ON ビット⁽¹⁾

1 = 16 ビット Timery を起動する

0 = 16 ビット Timery を停止する

bit 14 **未実装:**「0」として読み出し

bit 13 **TSIDL:** アイドルモード時停止ビット ⁽¹⁾

1 = デバイスがアイドルモードに移行した時にモジュールの動作を停止する

0 = アイドルモード時もモジュールの動作を継続する

bit 12-7 **未実装:**「0」として読み出し

bit 6 **TGATE**: タイマゲート時間積算イネーブルビット ⁽¹⁾

TCS = 1 の時: このビットを無視 TCS = 0 の時:

1 = ゲート時間積算を有効にする 0 = ゲート時間積算を無効にする

bit 5-4 TCKPS<1:0>: Timery 入力クロック プリスケール選択ビット (1)

11 = 1:256 10 = 1:64 01 = 1:8 00 = 1:1

bit 3-2 **未実装:**「0」として読み出し

bit 1 TCS: Timery クロック源選択ビット ⁽¹⁾

1 = T3CK ピンからの外部クロック (立ち上がりエッジを使用)

0 = 内部クロック (Fosc/2)

bit 0 **未実装:**「0」として読み出し

Note 1: 32 ビット動作が有効 (TxCON<3> = 1) の場合、これらのビットは Timery の動作に影響せず、全てのタイマ機能は TxCON レジスタによって設定されます。

NOTE:

14.0 専用タイマを備えた入力キャプチャ

Note:

本書は、PIC24F デバイスの中の特定製品 グループを対象とし、それらの機能の概 要説明を目的としています。従って本書 は、包括的な参照資料の提供を意図した ものではありません。詳細は「PIC24F ファミリ リファレンス マニュアル」のセ クション 34.「専用タイマを備えた入力 **キャプチャ」(DS39722) を参照してくだ**

PIC24FV32KA304 ファミリの全てのデバイスは、3 個 の独立した入力キャプチャ モジュールを実装してい ます。各モジュールは、外部パルスイベントのキャプ チャと割り込みの生成用に、豊富なコンフィグレー ションと動作オプションを提供します。

入力キャプチャ モジュールの主な特長には下記が含 まれます。

- 番号が連続する2個のモジュールをカスケード接続 する事により、全てのモードで 32 ビット動作用に ハードウェア コンフィグレーションが可能
- トリガモードと同期モードによる出力コンペア動作 (最大 20 種類のユーザ選択可能なトリガ/同期ソー ス)
- 各種イベントでタイマ値をキャプチャしてホールド する4段 FIFO
- 設定可能な割り込み生成
- 各モジュールで最大6種類のクロック源を使用して 別々の内部 16 ビットカウンタを駆動

モジュールは2つのレジスタ (ICxCON1 (レジスタ 14-1 参照)とICxCON2(レジスタ14-2参照))により制御さ れます。図 14-1 にモジュールの概略ブロック図を示し ます。

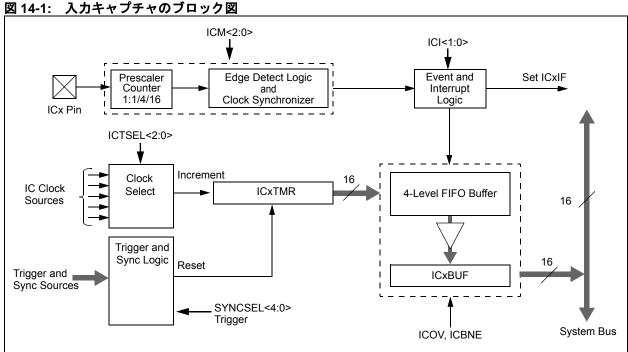
一般的動作モード 14.1

同期モードとトリガモード 14.1.1

既定値では、入力キャプチャ モジュールはフリーラン ニング モードで動作します。内部 16 ビットカウンタ ICxTMR は、選択された外部クロック源に同期した周期 で連続的にカウントアップし、オーバーフローするた びに FFFFh から 0000h ヘロールオーバーします。キャ プチャ イベントが発生すると、内部カウンタの 16 ビッ ト値が FIFO バッファに書き込まれます。

同期モードの場合、選択したクロック源が有効化され ると、モジュールは即座に ICx ピンでイベントのキャ プチャを開始します。選択した同期ソースでイベント が発生するたびに、内部カウンタがリセットされます。 トリガモードの場合、モジュールは、内部カウンタの 動作を開始する前に、別の内部モジュールからの同期 イベントが発生するまで待機します。

標準のフリーランニング動作を選択するには、 SYNCSEL ビットを「00000」に設定し、ICTRIG ビッ ト (ICxCON2<7>) をクリアします。SYNCSEL ビット を「00000」以外の値に設定した場合、同期モードま たはトリガモードが選択されます。同期モードとトリ ガモードのどちらを選択するかは、ICTRIG ビットで 指定します。このビットをセットすると、トリガモー ド動作が選択されます。どちらのモードでも、 SYNCSEL ビットで同期 / トリガソースを指定します。 SYNCSEL ビットを「00000」に設定し、かつ ICTRIG トリガをセットした場合、モジュールはソフトウェア トリガモードで動作します。この場合、TRIGSTAT ビッ ト (ICxCON2<6>) を手動でセットする事によってキャ プチャ動作を開始できます。



14.1.2 カスケード(32 ビット) モード

既定値では、各モジュールは、それぞれの 16 ビットタイマを使用して別々に動作します。分解能を高めるために、隣り合う奇数番号モジュールと偶数番号モジュールを組み合わせて、1 個の 32 ビット モジュールとして機能するように構成できます (例えば、モジュール 1 と 2 の組み合わせが可能)。奇数番号モジュール (ICx) は 32 ビット レジスタペアの下位 16 ビットを提供し、偶数番号モジュール (ICy) は上位 16 ビットを提供します。ICx レジスタがロールオーバーすると、対応する ICy レジスタが 1 つインクリメントします。

カスケード動作は、ペアで使用する両方のモジュールの IC32 ビット (ICxCON2<8>) をセットする事により、ハードウェアでコンフィグレーションされます。

14.2 キャプチャ動作

です。

入力キャプチャ モジュールは、ICx の立ち上がりエッジまたは ICx の全ての状態変化でタイマ値をキャプチャして割り込みを生成するように設定できます。キャプチャは、毎回の立ち上がりエッジあるいは 4回毎または16回毎の立ち上がりエッジで発生するように設定できます。割り込みは、毎回のイベントまたは一定回数のイベント毎に発生するように設定できます。モジュールのキャプチャ動作の設定手順は下記の通り

- 1. 同期モードを使用する場合、まず同期ソースを無効にします。
- ICBNE ビット(ICxCON1<3>)がクリアされるまで ICxBUF を読み出す事によって、FIFO から以前の データの全てを確実に削除します。
- SYNCSEL ビット (ICxCON2<4:0>) で必要な同期 / トリガソースを選択します。
- 4. ICTSEL ビット (ICxCON1<12:10>) で必要なクロック源を選択します。そのクロック源が動作中である場合、これに正しく同期するために、入カキャプチャモジュールを有効にする前に ICTSELビットを設定する必要があります。
- ICI ビット (ICxCON1<6:5>) で必要な割り込み頻度を設定します。
- 6. 動作モードとして同期モードまたはトリガモード を選択します。
 - a) SYNCSEL ビットが「00000」に設定されて いない事を確認します。
 - b) 同期モードの場合、ICTRIGビット(ICxCON2<7>) をクリアします。
 - c) トリガモードの場合、ICTRIG をセットし、 TRIGSTAT ビット (ICxCON2<6>) をクリアし ます。
- 7. ICM ビット (ICxCON1<2:0>) で必要な動作モード を設定します。
- 8. 選択したトリガ/同期ソースを有効化します。

32 ビット カスケード動作を使用する場合、設定手順は下記のように若干異なります。

- 1. ペアで使用する両方のモジュールの IC32 ビット ICyCON2<8> と ICxCON2<8> をセットします(先に偶数番号モジュールを有効化します)。これにより、両方のモジュールが一緒に機能を開始します。
- 2. 両方のモジュールの ICTSEL および SYNCSEL ビットで、同じ同期 / トリガソースとクロック源 を選択します。まず偶数番号モジュールを設定し てから、奇数番号モジュールを設定します。両方 のモジュールで同じ ICTSEL および SYNCSEL 設 定を使用する必要があります。
- 3. 偶数番号モジュールのICTRIGビット(ICyCON2<7>) をクリアします。これにより、偶数番号モジュー ルは、そのトリガ設定に関係なく、奇数番号モ ジュールと一緒に同期モードで動作します。
- 4. 奇数番号モジュールのICIビット(ICxCON1<6:5>) で必要な割り込み頻度を設定します。
- 5. 奇数番号モジュールのICTRIGビット(ICxCON2<7>) でトリガモード動作または同期モード動作のいず れかを選択します。

Note: 同期モード動作の場合、最後の手順として、同期ソースを有効化します。両方の入力キャプチャ モジュールは、同期ソースが有効化されるまで、リセット状態を保持します。

6. 奇数番号モジュールのICM ビット (ICxCON1<2:0>) で必要なキャプチャ モードを設定します。

タイムベースとトリガ/同期ソースが有効化されると、モジュールはイベントをキャプチャできる状態となります。FIFO 内に少なくとも 1 個のキャプチャ値が存在すると、ICBNE ビット (ICxCON1<3>) がセットされます。ICBNE が「0」にクリアされるまで、FIFO から入力キャプチャ値を読み出す必要があります。

32 ビット動作の場合、ICxBUF と ICyBUF の両方を読み出す事によって完全な 32 ビットタイマ値が得られます (ICxBUF が Isw、ICyBUF が msw を格納)。奇数番号モジュールのICBNE ビット (ICxCON1<3>)がセットされている場合、FIFO バッファ内に少なくとも 1 個のキャプチャ値が存在します。ICBNE ビットがハードウェアによって自動的にクリアされるまで、バッファレジスタを繰り返し読み出す必要があります。

レジスタ 14-1: ICxCON1: 入力キャプチャ x 制御レジスタ 1

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
_	_	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	_	_
bit 15							bit 8

U-0	R/W-0	R/W-0	R-0, HCS	R-0, HCS	R/W-0	R/W-0	R/W-0
_	ICI1	ICI0	ICOV	ICBNE	ICM2 ⁽¹⁾	ICM1 ⁽¹⁾	ICM0 ⁽¹⁾
bit 7							bit 0

凡例: HS = ハードウェア クリア / セット可能ビット

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-14 **未実装:**「0」として読み出し

bit 13 ICSIDL: アイドル時入力キャプチャ x モジュール停止制御ビット

1 = CPU がアイドルモード時に入力キャプチャ モジュールを停止する

0 = CPU がアイドルモード時でも入力キャプチャ モジュールの動作を継続する

bit 12-10 ICTSEL<2:0>: 入力キャプチャタイマ選択ビット

111 = システムクロック (Fosc/2)

110 = 予約済み

101 = 予約済み

100 = Timer1

011 = Timer5

010 = Timer4

001 = Timer2

000 = Timer3

bit 9-7 **未実装:**「0」として読み出し

bit 6-5 **ICI<1:0>**: 割り込みあたりキャプチャ数選択ビット

11 = キャプチャ イベント4回毎に割り込む

10 = キャプチャ イベント3回毎に割り込む

01 = キャプチャ イベント2回毎に割り込む

00 = 毎回のキャプチャ イベントで割り込む

bit 4 ICOV: 入力キャプチャx オーバーフロー ステータスフラグ ビット(読み出し専用)

1 = 入力キャプチャ オーバーフローが発生した

0 = 入力キャプチャ オーバーフローは発生していない

bit 3 ICBNE: 入力キャプチャ x バッファ エンプティ ステータスビット (読み出し専用)

1=入力キャプチャバッファはエンプティではない(少なくとも1個以上のキャプチャ値を読み出し可能)

0 = 入力キャプチャ バッファはエンプティである

bit 2-0 ICM<2:0>: 入力キャプチャモード選択ビット (1)

111 = 割り込みモード:デバイスのスリープまたはアイドルモード時にのみ、入力キャプチャは割り込みピンとして機能する(立ち上がりエッジのみ検出、他の制御ビットを全て無視)

110 = 未使用(モジュール無効)

101 = プリスケーラ キャプチャモード: 立ち上がりエッジ 16 回毎にキャプチャする

100 = プリスケーラ キャプチャモード: 立ち上がりエッジ4回毎にキャプチャする

011 = 単純キャプチャモード: 立ち上がりエッジで毎回キャプチャする

010 = 単純キャプチャモード: 立ち下がりエッジで毎回キャプチャする

001 = エッジ検出キャプチャモード: 立ち上がりエッジと立ち下がりエッジの両方で毎回キャプチャする (ICI<1:0> ビットはこのモードでの割り込み生成を制御しません)

000 = 入力キャプチャ モジュールを OFF にする

レジスタ 14-2: ICxCON2: 入力キャプチャ x 制御レジスタ 2

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
_	_	_	_	_	_	_	IC32
bit 15							bit 8

R/W-0	R/W-0, HS	U-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-1
ICTRIG	TRIGSTAT	_	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0
bit 7							bit 0

凡例: HS = ハードウェア セット可能ビット

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-9 **未実装:**「0」として読み出し

bit 8 IC32: IC モジュール カスケード イネーブルビット (32 ビット動作)

1 = ICxとICvは32ビットモジュールとしてカスケードで動作する(このビットを両方のモジュールで

セットする必要があります)

0 = ICx は別々に 16 ビット モジュールとして機能する

bit 7 ICTRIG: ICx トリガ / 同期選択ビット

1 = SYNCSELx ビットが指定するソースから ICx をトリガする

0 = SYNCSELx ビットが指定するソースに ICx を同期する

bit 6 TRIGSTAT: タイマトリガ ステータスビット

1 = タイマソースはトリガされ、現在動作している(このビットはハードウェアでセットされます、ソ

フトウェアでもセット可能です)

0 = タイマソースはトリガされておらず、クリア状態を保持している

bit 5 **未実装:**「0」として読み出し

bit 4-0 **SYNCSEL<4:0>:** トリガ / 同期ソース選択ビット

11111 = 予約済み

11110 = 予約済み

11101 = 予約済み

11100 = CTMU(1)

 $11011 = A/D^{(1)}$

11010 = コンパレータ 3⁽¹⁾

11001 = コンパレータ 2(1)

11000 = コンパレータ 1⁽¹⁾

10111 = 入力キャプチャ4

10110 = 入力キャプチャ 3

10101 = 入力キャプチャ2

10100 = 入力キャプチャ1

10011 = 予約済み

10010 = 予約済み

1000x = 予約済み

01111 = Timer5

01110 = Timer4

01101 = Timer3

01100 = Timer2

01000 = Timer1

01010 = 入力キャプチャ5

01001 = 予約済み

01000 = 予約済み

00111 = 予約済み

00110 = 予約済み

00101 = 出力コンペア 5

00100 = 出力コンペア4

00011 = 出力コンペア3

00010 = 出力コンペア 2 00001 = 出力コンペア 1

00000 = 他のモジュールに同期しない

Note 1: これらの入力はトリガ源としてのみ使用し、同期源として使用しないでください。

15.0 専用タイマを備えた出力コンペア

Note:

本書は、PIC24F デバイスの中の特定製品 グループを対象とし、それらの機能の概要説明を目的としています。従って本書は、包括的な参照資料の提供を意図したものではありません。詳細は「PIC24F ファミリ リファレンス マニュアル」のセクション 35.「専用タイマを備えた入力キャプチャ」(DS39722) を参照してください。

PIC24FV32KA304 ファミリの全てのデバイスは、3 個の独立した出力コンペア モジュールを実装しています。各モジュールは、内部デバイスイベントでパルス列を生成するための豊富なコンフィグレーションと動作オプションを提供します。また、これらのモジュールは、電源アプリケーション用にパルス幅変調 (PWM) 波形を生成する事もできます。

出力コンペア モジュールの主な特長には下記が含まれます。

- 番号が隣り合う2個のモジュールをカスケード接続 する事により、全てのモードで32ビット動作用の ハードウェアコンフィグレーションが可能
- ・ 同期モードとトリガモードによる出力コンペア動作 (最大 21 種類のユーザ選択可能なトリガ/同期ソース)
- 2個の周期レジスタ(メインレジスタ OCxR とセカンダリレジスタOCxRS)を使用して可変幅のパルスを柔軟に生成
- モジュールのコンフィグレーションにより、出力イベント時の単発または連続パルス生成、あるいは PWM 波形の連続生成が可能
- 各モジュールで最大 6 種類のクロック源を使用して、別々の内部 16 ビットカウンタを駆動

15.1 一般的動作モード

15.1.1 同期モードとトリガモード

既定値では、出力コンペア モジュールはフリーランニング モードで動作します。内部 16 ビットカウンタ OCxTMR は、選択された外部クロック源に同期した周期で連続的にカウントアップし、オーバーフローするたびに FFFFh から 0000h ヘロールオーバーします。内部カウンタといずれかの周期レジスタが一致するたびに、コンペアまたは PWM イベントが生成されます。

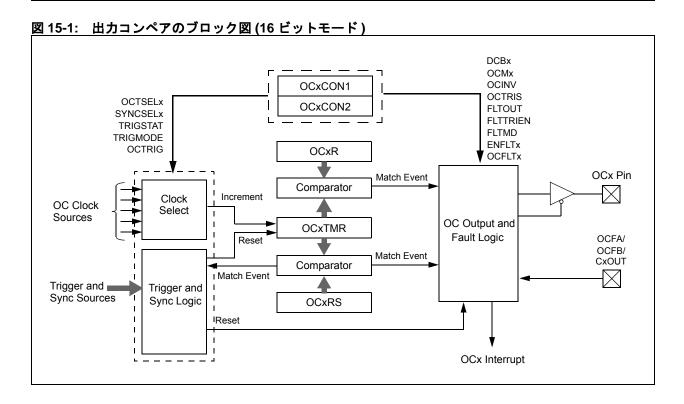
同期モードの場合、選択したクロック源が有効化されると、モジュールは即座にコンペアまたは PWM 動作を開始します。選択した同期ソースでイベントが発生するたびに、モジュールの内部カウンタがリセットされます。トリガモードの場合、モジュールは、カウンタの動作を開始する前に、別の内部モジュールからの同期イベントを待機します。

フリーランニングモードは、既定値として選択されるか、あるいは SYNCSEL ビット (OCxCON2<4:0>)が「00000」に設定されている場合に選択されます。 SYNCSEL ビットを「00000」以外の値に設定した場合、同期モードまたはトリガモードが選択されます。同期モードとトリガモードのどちらを選択するかは、ICTRIG ビット (OCxCON2<7>)で指定します。このビットをセットすると、トリガモード動作が選択されます。どちらのモードでも、SYNCSEL ビットで同期/トリガソースを指定します。

15.1.2 カスケード (32 ビット) モード

既定値では、各モジュールは、それぞれの 16 ビットタイマとデューティ サイクル レジスタを使用して別々に動作します。レンジを拡大するために、隣り合う奇数番号のモジュールと偶数番号のモジュールを組み合わせて、1 個の 32 ビット モジュールとして機能するように構成できます。例えばモジュール 1 と 2 の組み合わせが可能です。奇数番号モジュール (OCx)は32 ビット レジスタペアの下位 16 ビットを提供し、偶数番号モジュール (OCy) は上位 16 ビットを提供します。OCx レジスタがロールオーバーすると、対応するOCy レジスタがインクリメントします。

カスケード動作は、ペアで使用する両方のモジュール の OC32 ビット (OCxCON2<8>) をセットする事によ り、ハードウェアでコンフィグレーションされます。



15.2 コンペア動作

コンペアモード(図 15-1)では、出力コンペアモジュールを単発または連続パルス生成用に設定できます。毎回のタイマイベントで、出力ピンを繰り返しトグルする事もできます。

モジュールをコンペア動作用に設定する手順は下記の 通りです。

- OCxR および OCxRS(ダブルコンペア モード用) デューティ サイクル レジスタの値を下記のよう に計算します:
 - a) 命令クロックサイクル時間を決定します。この際、タイマソースへの外部クロック周波数 (使用する場合)と、タイマ プリスケーラ設 定を考慮する必要があります。
 - b) タイマ開始値 (0000h) に対する出力パルス立 ち上がりエッジまでの相対的な時間を計算し ます。
 - c) 目標パルス幅とパルス立ち上がりエッジまで の時間に基づいて、パルス立ち下がりエッジ までの時間を計算します。
- 立ち上がりエッジ値を OCxR に書き込み、立ち下がりエッジ値を OCxRS に書き込みます。
- 3. トリガモード動作の場合、OCTRIGをセットしてトリガモードを有効にします。TRIGMODE ビットでトリガ動作を選択します(ハードウェアとソフトウェアのどちらが TRIGSTAT をクリアするのかを選択)。同期モードの場合、OCTRIGをクリアします。
- 4. SYNCSEL<4:0> ビットでトリガまたは同期ソースを選択します。フリーランニングタイマ動作が必要な場合、SYNCSELビットを「00000」(同期/トリガソースなし)に設定します。
- 5. OCTSEL<2:0> ビットで、タイムベース ソースを 選択します。使用するクロック源が動作中である 場合、これに正しく同期するために、出力コンペ ア モジュールを有効にする前に OCTSEL<2:0> ビットを設定する必要があります。必要に応じて、 選択したタイマの TON ビットをセットして、コン ペア タイムベースのカウントを有効にします。同 期モード動作は、同期ソースが有効化されると即 座に開始します。トリガモード動作は、トリガソー ス イベント発生後に開始します。
- 6. OCM<2:0> ビットを適切なコンペア動作用に設定 します (0xx)。

32 ビットのカスケード動作の場合、下記の手順も必要です。

- 1. OCyCON2<8> レジスタと OCxCON2<8> レジスタの両方の OC32 ビットをセットします。先に偶数番号モジュールを有効にします。これにより、両方のモジュールが一緒に機能を開始します。
- 2. 偶数番号モジュールを同期モードで動作させるために、偶数番号モジュールの OCTRIG ビット (OCyCON2)をクリアします。
- OCyモジュールの出力とフォルト設定を設定します。
- 4. OCxモジュールの出力ピンを出力状態に強制する ために、OCTRIS ビットをクリアします。
- トリガモード動作が必要な場合、OCx モジュールの OCTRIG ビット (OCxCON2<7>)、TRIGSTAT ビット (OCxCON2<6>)、SYNCSEL ビット(OCxCON2<4:0>) でトリガオプションを設定します。
- 先に OCy モジュールの OCM<2:0> で動作モード (コンペアまたは PWM) を設定し、次に OCx モ ジュールを設定します。

選択した出力モードに応じて、モジュールは OCx ピンを既定値状態に保持し、OCxR がタイマに一致した時に、OCx ピンを既定値状態とは反対の状態へ遷移させます。ダブルコンペア モードの場合、OCx ピンは OCxRS の一致時に既定値状態に戻されます。 OCxIF 割り込みフラグは、シングルコンペア モードの場合 OCxR 一致後にセットされ、ダブルコンペア モードの場合各 OCxRS 一致後にセットされます。

単発パルスイベントは 1 回だけ発生しますが、OCxCON1レジスタの値を単純に再書き込みして繰り返す事ができます。連続パルスイベントは、終了されるまで繰り返しパルスを生成し続けます。

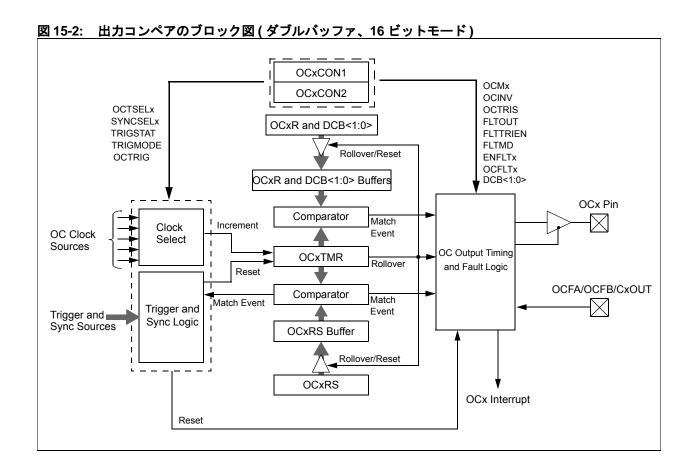
15.3 パルス幅変調 (PWM) モード

PWM モードでは、出力コンペア モジュールをエッジ アラインまたはセンターアラインされたパルス波形生成用に設定できます。全てのPWM動作は二重バッファリングされます(バッファレジスタはモジュール内部に置かれ、SFR 空間に割り当てられません)。

出力コンペア モジュールをエッジアライン PWM 動作用に設定する手順は下記の通りです。

- 1. 目標オンタイムを計算し、その値を OCxR レジス タに書き込みます。
- 2. 目標周期を計算し、その値を OCxRS レジスタに 書き込みます。
- この OCx モジュールを同期ソースとして選択するために、0x1F を SYNCSEL<4:0> (OCxCON2<4:0>) に書き込み、「0」を OCTRIG (OCxCON2<7>) に書き込みます。

- OCTSEL2<2:0> (OCxCON<12:10>) ビットでクロック源を選択します。
- 5. 必要に応じて、タイマと出力コンペア モジュール の割り込みを有効にします。出力コンペア割り込みは、PWM フォルトピンを使用する場合に必要です。
- 6. OCM<2:0> (OCxCON1<2:0>) ビットで、必要な PWM モードを選択します。
- 7. タイマをクロック源として選択した場合、TMRy プリスケール値を設定し、タイムベースを有効にするために TON (TxCON<15>) ビットをセットします。



15.3.1 PWM 周期

エッジアライン PWM モードでは、OCxRS レジスタの値によって周期が決まります。センターアライン PWM モードでは、同期ソースの周期 (タイマの PRy 等)によって周期が決まります。いずれのモードでも、周期の計算には式 15-1 を使用できます。

式 15-1: PWM 周期の計算 ⁽¹⁾

PWM 周期 = [Value + 1] x TCY x (プリスケーラ値)

Value = OCxRS(xyyyr = OCxRS(xyyyr = OCxRS(xyyyr = OCxRS))

TMRy が同期ソースである場合)

Note 1: Tcy = Tosc * 2 に基づく (Doze モードと PLL は無効)

15.3.2 PWM デューティ サイクル

PWM デューティ サイクルは、OCxRS および OCxR レジスタで指定します。OCxRS および OCxR レジスタはいつでも書き込み可能ですが、デューティ サイクル値は 1 周期が終了するまでラッチされません。これにより、PWM デューティ サイクルにダブルバッファを提供し、PWM 動作のグリッチを回避しています。

PWM デューティ サイクルの重要境界条件として下記が挙げられます。

- エッジアライン PWM モード:
 - OCxR と OCxRS に 0000h を書き込んだ場合、OCx ピンはLOWのままです(0% デューティ サイクル)。
 - OCxRS の値が OCxR よりも大きい場合、OCx ピンはHIGHのままです(100%デューティ サイクル)。
- センターアライン PWM モード (TMRy を同期ソースとして使用):
 - OCxR、OCxRS、PRy の全てに 0000h を書き込ん だ場合、OCx ピンは LOW のままです (0% デュー ティ サイクル)。
 - OCxRS の値が PRy よりも大きい場合、OCx ピンは HIGH です (100% デューティ サイクル)。

例 15-3 に PWM モードの詳細なタイミングを示します。表 15-1 と表 15-2 に、それぞれ 4 MIPS と 10 MIPS で動作するデバイスの PWM 周波数と分解能の例を示します。

式 15-2: 最大 PWM 分解能の計算 ⁽¹⁾

最大 PWM 分解能 (bit) =

$$\frac{\log_{10}\left(\frac{FCY}{FPWM}\bullet(プリスケール値)}{\log_{10}(2)}$$
bit

Note 1: Fcy = Fosc/2 に基づく、Doze モードと PLL は無効

式 15-3: PWM 周期とデューティ サイクルの計算 ⁽¹⁾

1. 目標 PWM 周波数 52.08 kHz に対応する OCxRS レジスタ値を求めます。ここで、Fosc = 8 MHz/PLL 併用 (デバイスクロック周波 = 32 MHz)、エッジアライン PWM モードを使用、プリスケーラ設定は 1:1 とします。

TCY = 2 * TOSC = 62.5 ns

PWM 周期 = 1/PWM 周波数 = 1/52.08 kHz = 19.2 μs PWM 周期 = (OCxRS + 1) • TCY • (OCx プリスケール値)

19.2 μ s = $(OCxRS + 1) \cdot 62.5 \text{ ns} \cdot 1$

OCxRS = 306

2. PWM 周波数 52.08 kHz/ デバイス周波数 32 MHz で使用可能なデューティ サイクルの最大分解能を求めます。

PWM 分解能 = $(\log_{10}(FCY/FPWM)/\log_{10}2)$ bit

 $= (\log_{10}(16 \text{ MHz}/52.08 \text{ kHz})/\log_{10}2) \text{ bit}$

= 8.3 bit

Note 1: Tcy = 2 * Tosc に基づく、Doze モードと PLL は無効

15.4 サブサイクル分解能

DCB ビット (OCxCON2<10:9>) は、1 命令サイクルよりも細かい分解能を提供します。DCB ビットを使用すると、一致イベントから生成される立ち下がりエッジのタイミングを、1 命令サイクルの 1/4、1/2、3/4 だけ遅らせる事ができます。

例えば、DCB<1:0>=10 に設定した場合、立ち下がりエッジは、一致イベントが発生した命令サイクルの開始時ではなく、そこから 1/2 サイクル遅れたタイミングで発生します。OCM<2:0>=001 の場合、これらのビットは使用できません。モジュールが PWM モードで動作している (OCM<2:0>=110 または 111) 場合、DCB ビットは二重バッファリングされます。

DCB ビットは、システムクロックと同じクロック源で使用する事を前提としています。プリスケーラを有効にした OCx モジュールを使用する場合、DCB ビットによって生じる立ち下がりエッジの遅延は、OCx モジュールの周期ではなくシステムクロックの周期を基準とします。

表 15-1: PWM 周波数と分解能の例 (4 MIPS、FCY = 4 MHz)⁽¹⁾

PWM 周波数	7.6 Hz	61 Hz	122 Hz	977 Hz	3.9 kHz	31.3 kHz	125 kHz
プリスケーラ比	8	1	1	1	1	1	1
周期値	FFFFh	FFFFh	7FFFh	0FFFh	03FFh	007Fh	001Fh
分解能(ビット)	16	16	15	12	10	7	5

Note 1: Fcy = Fosc/2 に基づく、Doze モードと PLL は無効

表 15-2: PWM 周波数と分解能の例 (16 MIPS、FcY = 16 MHz)⁽¹⁾

PWM 周波数	30.5 Hz	244 Hz	488 Hz	3.9 kHz	15.6 kHz	125 kHz	500 kHz
プリスケーラ比	8	1	1	1	1	1	1
周期値	FFFFh	FFFFh	7FFFh	0FFFh	03FFh	007Fh	001Fh
分解能(ビット)	16	16	15	12	10	7	5

Note 1: Fcy = Fosc/2 に基づく、Doze モードと PLL は無効

レジスタ 15-1: OCxCON1: 出力コンペア x 制御レジスタ 1

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
_	_	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	ENFLT2	ENFLT1
bit 15							bit 8

R/W-0	R/W-0, HCS	R/W-0, HCS	R/W-0, HCS	R/W-0	R/W-0	R/W-0	R/W-0
ENFLT0	OCFLT2	OCFLT1	OCFLT0	TRIGMODE	OCM2 ⁽¹⁾	OCM1 ⁽¹⁾	OCM0 ⁽¹⁾
bit 7							bit 0

凡例: $HS = N - F \dot{D}_{x} T \dot{D}_{y} T / D \dot{D}$

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-14 **未実装:**「0」として読み出し

bit 13 OCSIDL: 出力コンペア x アイドルモード時停止制御ビット

1 = CPU アイドルモード時は出力コンペア x を停止する

0 = CPU アイドルモード時でも出力コンペアxの動作を継続する

bit 12-10 OCTSEL<2:0>: 出力コンペア x タイマ選択ビット

111 = システムクロック

110 = 予約済み

101 = 予約済み

100 = Timer1

011 = Timer5

010 = Timer4

001 = Timer3

000 = Timer2

bit 9 **ENFLT2**: コンパレータ フォルト入力イネーブルビット ⁽²⁾

1= コンパレータ フォルト入力を有効にする

0 = コンパレータ フォルト入力を無効にする

bit 8 ENFLT1: OCFB フォルト入力イネーブルビット

1 = OCFB フォルト入力を有効にする

0 = OCFB フォルト入力を無効にする

bit 7 ENFLTO: OCFA フォルト入力イネーブルビット

1 = OCFA フォルト入力を有効にする

0 = OCFA フォルト入力を無効にする

bit 6 **OCFLT2:** PWM コンパレータ フォルト条件ステータスビット ⁽²⁾

- 1 = PWM コンパレータ フォルト条件が発生した (このビットはハードウェアウェアのみでクリアされます)
- 0 = PWM コンパレータ フォルト条件は発生していない (このビットは、OCM<2:0>=111 の時にのみ 使用されます)

bit 5 OCFLT1: PWM OCFB フォルト条件ステータスビット

- 1 = PWM OCFB フォルト条件が発生した(このビットはハードウェアウェアのみでクリアされます)
- 0 = PWM OCFB フォルト条件は発生していない (このビットは、OCM<2:0> = 111 の時にのみ使用されます)
- bit 4 OCFLT0: PWM OCFA フォルト条件ステータスビット
 - 1 = PWM OCFA フォルト条件が発生した(このビットはハードウェアウェアのみでクリアされます)
 - 0 = PWM OCFA フォルト条件は発生していない (このビットは、OCM<2:0> = 111 の時にのみ使用されます)
- bit 3 TRIGMODE: トリガステータス モード選択ビット
 - 1 = TRIGSTAT (OCxCON2<6>) は、OCxRS = OCxTMR の時にクリアされる、またはソフトウェアでクリアされる
 - 0 = TRIGSTAT はソフトウェアのみでクリアされる
- **Note 1:** OCx モジュールによって、フォルト入力に使用するコンパレータ モジュールが異なります (OC1 と OC2 はコンパレータ 1 を使用、OC3 と OC4 はコンパレータ 2 を使用、OC5 はコンパレータ 3 を使用)。

レジスタ 15-1: OCxCON1: 出力コンペア x 制御レジスタ 1 (続き)

- bit 2-0 OCM<2:0>: 出力コンペア x モード選択ビット (1)
 - 111 = OCx でセンターアライン PWM モード
 - 110 = OCx でエッジアライン PWM モード
 - 101 = ダブルコンペア連続パルスモード: OCx ピンをLOWに初期化し、OCxR およびOCxRSの一致でOCx の状態を繰り返しトグルする
 - 100 = ダブルコンペア単発パルスモード: OCx ピンを LOW に初期化し、1 サイクル中の OCxR および OCxRS 一致で OCx の状態をトグルする
 - 011 = シングルコンペア連続パルスモード: コンペアイベントは OCx ピンを繰り返しトグルする
 - 010 = シングルコンペア単発パルスモード: OCx ピンをHIGHに初期化し、コンペアイベントでOCx ピンを LOW に変更する
 - 001 = シングルコンペア単発パルスモード: OCxピンをLOWに初期化し、コンペアイベントでOCxピンを HIGH に変更する
 - 000 = 出力コンペア チャンネルを無効にする
- **Note 1:** OCx モジュールによって、フォルト入力に使用するコンパレータ モジュールが異なります (OC1 と OC2 はコンパレータ 1 を使用、OC3 と OC4 はコンパレータ 2 を使用、OC5 はコンパレータ 3 を使用)。

レジスタ 15-2: OCxCON2: 出力コンペア x 制御レジスタ 2

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
FLTMD	FLTOUT	FLTTRIEN	OCINV	_	DCB1 ⁽³⁾	DCB0 ⁽³⁾	OC32
bit 15							bit 8

R/W-0	R/W-0, HS	R/W-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0
OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0
bit 7							bit 0

凡例:HS = ハードウェア セット可能ビットR = 読み出し可能ビットW = 書き込み可能ビットU = 未実装ビット、「0」として読み出し-n = POR 時の値1 = ビットをセット0 = ビットをクリアx = ビットは未知

bit 15 **FLTMD:** フォルトモード選択ビット

1 = フォルト要因が排除され、対応する OCFLTO ビットがソフトウェアでクリアされるまでフォルト モードを維持する

0 = フォルト要因が排除され、次の PWM 周期が開始されるまでフォルトモードを維持する

bit 14 **FLTOUT**: フォルト出力ビット

1 = フォルト時に PWM 出力を HIGH に駆動する

0 = フォルト時に PWM 出力を LOW に駆動する

bit 13 FLTTRIEN: フォルト出力状態選択ビット

1 = フォルト条件時に、ピンを「出力」にする

0 = フォルトはピンの I/O 状態に影響しない

bit 12 OCINV: OCMP 反転ビット

1 = OCx 出力を反転する

0 = OCx 出力を反転しない

bit 11 **未実装:**「0」として読み出し

bit 10-9 **DCB<1:0>:** OC パルス幅最下位ビット (3)

11 = OCx 立ち下がりエッジを 3/4 命令サイクル遅延する

10 = OCx 立ち下がりエッジを 1/2 命令サイクル遅延する

01 = OCx 立ち下がりエッジを 1/4 命令サイクル遅延する

00 = OCx 立ち下がりエッジは命令サイクル開始時に発生する

bit 8 OC32: OC モジュール カスケード イネーブルビット (32 ビット動作)

1 = カスケード モジュール動作を有効にする

0 = カスケードモジュール動作を無効にする

bit 7 OCTRIG: OCx トリガ / 同期選択ビット

1 = SYNCSELx ビットが指定するソースから OCx をトリガする

0 = SYNCSELx ビットが指定するソースに OCx を同期する

bit 6 TRIGSTAT: タイマトリガ ステータスビット

1 = タイマソースはトリガされ、動作している

0 = タイマソースはトリガされておらず、クリア状態を保持している

bit 5 OCx 出力ピン方向選択ビット

1 = OCx ピンを 3 ステートにする

0 = OCx ピンに出力コンペア周辺モジュール x を接続する

Note 1: このモードまたは別の等価な SYNCSEL 設定を選択して OC モジュールをその OC モジュール自身のトリガソースとして使用しないでください。

2: これらの入力は、トリガソースとしてのみ使用し、同期ソースとして使用しないでください。

3: これらのビットは、OCINV = 1 の時に立ち上がりエッジに影響します。これらのビットは、OCM ビット (OCxCON1<2:0>) = 001 の時に効果を持ちません。

レジスタ 15-2: OCxCON2: 出力コンペア x 制御レジスタ 2 (続き)

```
bit 4-0
             SYNCSEL<4:0>: トリガ / 同期ソース選択ビット
             11111 = この OC モジュール ^{(1)}
             11110 = 予約済み
             11101 = 予約済み
             11100 = CTMU<sup>(2)</sup>
             11011 = A/D^{(2)}
             11010 = コンパレータ 3^{(2)}
             11001 = コンパレータ 2<sup>(2)</sup>
             11000 = コンパレータ 1<sup>(2)</sup>
             10111 = 入力キャプチャ 4<sup>(2)</sup>
             10110 = 入力キャプチャ 3(2)
             10101 = 入力キャプチャ 2<sup>(2)</sup>
             10100 = 入力キャプチャ 1<sup>(2)</sup>
             1000x = 予約済み
             01111 = Timer5
             01110 = Timer4
             01101 = Timer3
             01100 = Timer2
             01011 = Timer1
             01010 = 入力キャプチャ 5<sup>(2)</sup>
             01001 = 予約済み
             01000 = 予約済み
             00111 = 予約済み
             00110 = 予約済み
             00101 = 出力コンペア 5<sup>(1)</sup>
             00100 = 出力コンペア 4<sup>(1)</sup>
             00011 = 出力コンペア 3<sup>(1)</sup>
             00010 = 出力コンペア 2<sup>(1)</sup>
             00001 = 出力コンペア 1<sup>(1)</sup>
             00000 = 他のモジュールに同期しない
```

- Note 1: このモードまたは別の等価な SYNCSEL 設定を選択して OC モジュールをその OC モジュール自身のトリガソースとして使用しないでください。
 - 2: これらの入力は、トリガソースとしてのみ使用し、同期ソースとして使用しないでください。
 - 3: これらのビットは、OCINV = 1 の時に立ち上がりエッジに影響します。これらのビットは、OCM ビット (OCxCON1<2:0>) = 001 の時に効果を持ちません。

16.0 シリアル ペリフェラル インターフェイス (SPI)

Note: 本書は、PIC24 デバイスの中の特定製品グループを対象とし、それらの機能の概要説明を目的としています。従って本書は、包括的な参照資料の提供を意図したものではありません。シリアルペリフェラルインターフェイスの詳細は「PIC24F ファミリリファレンスマニュアル」のセクション 23.「シリアルペリフェラルインターフェイス (SPI)」(DS39699)を参照してください。

シリアル ペリフェラル インターフェイス (SPI) モジュールは、他の周辺モジュールまたはマイクロコントローラ デバイスとの通信に使用する同期シリアルインターフェイスです。通信相手の周辺モジュールにはデータ EEPROM、シフトレジスタ、ディスプレイドライバ、A/D コンバータ等が含まれます。SPI モジュールは Motorola® SPI および SIOP インターフェイスと互換性を持ちます。

本モジュールは2種類のバッファモードによる動作をサポートします。標準モードでは、単一のシリアルバッファを使用してデータをシフトします。拡張バッファモードでは、8段 FIFO バッファを使用してデータをシフトします。

Note: 標準モードでも拡張バッファモードでも、 SPIxBUF レジスタに対する「読み出し - 変 更 - 書き込み」動作(ビット指向命令等) の実行を避ける必要があります。

SPI モジュールは、マスタモードまたはスレーブモード動作において基本フレーム化 SPI プロトコルもサポートします(全部で4種類のフレーム化SPIコンフィグレーションをサポート)。

SPI シリアルインターフェイスは下記の4本のピンで 構成されます。

- SDI1: シリアルデータ入力
- SDO1: シリアルデータ出力
- SCK1: シフトクロック入力または出力
- SS1: アクティブ LOW スレーブ選択またはフレー ム同期 I/O パルス

SPI モジュールは、2 ピン、3 ピ<u>ン、</u>4 ピンを使用して 動作できます。3 ピンモード<u>では</u> SS1 を使用しません。 2 ピンモードでは SDO1 と SS1 を使用しません。

標準モードと拡張バッファモードにおけるモジュールのブロック図を、それぞれ図16-1と図16-2に示します。

PIC24FV32KA304ファミリのデバイスは2個のSPIモジュールを内蔵しています。

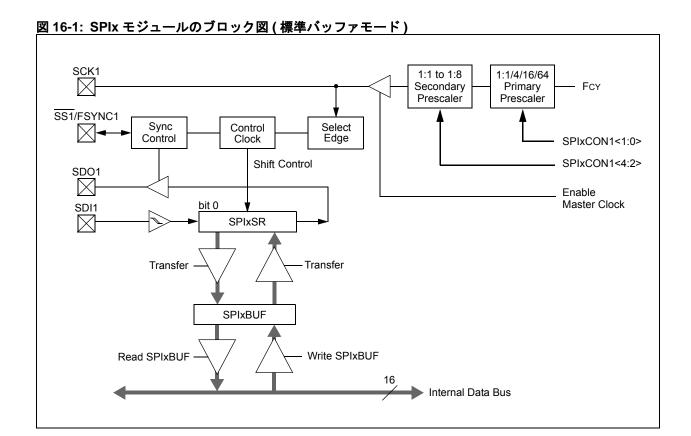
ote: 本セクションでは、これらのSPIモジュールを「SPIx」と表記します。特殊機能レジスタ (SFR) の表記もこれに従います。例えばSPI1モジュール用の制御レジスタの名前は、SPI1CON1 または SPI1CON2です。

SPIx モジュールの動作を標準マスタモードにセット アップする手順は下記の通りです。

- 1. 割り込みを使用する場合:
 - a) IFSO レジスタの各 SPIxIF ビットをクリアし ます。
 - b) IECO レジスタの各 SPIxIE ビットをセットします。
 - c) IPC2 レジスタの各 SPIxIPx ビットに割り込み優先度設定を書き込みます。
- SPIxCON1 および SPIxCON2 レジスタに必要な 設定を書き込み、MSTEN ビット (SPIxCON1<5>) を「1」にセットします。
- 3. SPIROV ビット (SPIxSTAT<6>) をクリアします。
- SPIEN ビット (SPIxSTAT<15>) をセットして SPI 動作を有効にします。
- 5. 送信するデータを SPIxBUF レジスタに書き込み ます。データを SPIxBUF レジスタに書き込むと、 即座に送信 (および受信) が開始します。

SPIx モジュールの動作を標準スレーブモードにセットアップする手順は下記の通りです。

- 1. SPIxBUF レジスタをクリアします。
- 2. 割り込みを使用する場合:
 - a) IFSO レジスタの各 SPIxIF ビットをクリアし ます。
 - b) IEC0 レジスタの各 SPIxIE ビットをセットします。
 - c) IPC2 レジスタの各 SPIxIP ビットに割り込み 優先度設定を書き込みます。
- SPIxCON1 および SPIxCON2 レジスタに必要な 設定を書き込み、MSTEN ビット (SPIxCON1<5>) を「0」にクリアします。
- 4. SMP ビットをクリアします。
- CKE ビットをセットした場合、SSEN ビット (SPIxCON1<7>) をセットして SS1 ピンを有効 にする必要があります。
- 6. SPIROV ビット (SPIxSTAT<6>) をクリアします。
- 7. SPIEN ビット (SPIxSTAT<15>) をセットして SPI 動作を有効にします。



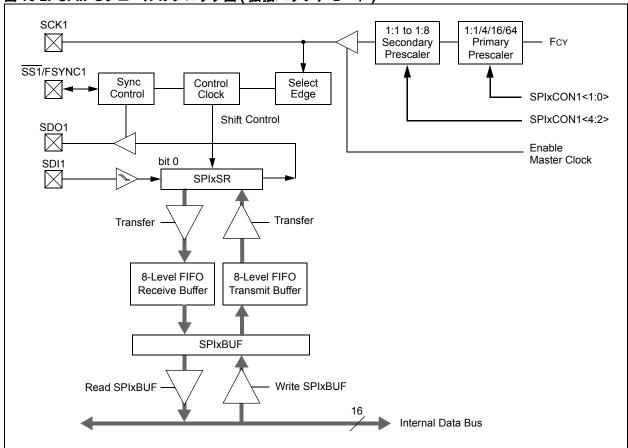
SPIx モジュールの動作を拡張バッファ マスタ (EBM) モードにセットアップする手順は下記の通りです。

- 1. 割り込みを使用する場合:
 - a) IFSO レジスタの各 SPIxIF ビットをクリアします。
 - b) IECO レジスタの各 SPIxIE ビットをセットします。
 - c) IPC2 レジスタの各 SPIxIPx ビットに割り込 み優先度設定を書き込みます。
- SPIxCON1 および SPIxCON2 レジスタに必要な 設定を書き込み、MSTEN ビット (SPIxCON1<5>) を「1」にセットします。
- 3. SPIROV ビット (SPIxSTAT<6>) をクリアします。
- 4. SPIBEN ビット (SPIxCON2<0>) をセットして拡張バッファモードを選択します。
- SPIEN ビット (SPIxSTAT<15>) をセットして SPI 動作を有効化します。
- 6. 送信するデータを SPIxBUF レジスタに書き込み ます。データを SPIxBUF レジスタに書き込むと、 即座に送信(および受信)が開始します。

SPIx モジュールの動作を拡張バッファ スレーブモードにセットアップする手順は下記の通りです。

- 1. SPIxBUF レジスタをクリアします。
- 2. 割り込みを使用する場合:
 - a) IFSO レジスタの各 SPIxIF ビットをクリアします。
 - b) IECO レジスタの各 SPIxIE ビットをセットします。
 - c) IPC2 レジスタの各 SPIxIPx ビットに割り込み優先度設定を書き込みます。
- 3. SPIxCON1 および SPIxCON2 レジスタに必要な 設定を書き込み、MSTEN ビット (SPIxCON1<5>) を「0」にクリアします。
- 4. SMP ビットをクリアします。
- CKE ビット<u>をセ</u>ットした場合、SSEN ビットを セットして SS1 ピンを有効にする必要があります。
- 6. SPIROV ビット (SPIxSTAT<6>) をクリアします。
- SPIBEN ビット (SPIxCON2<0>) をセットして拡張バッファモードを選択します。
- 8. SPIEN ビット (SPIxSTAT<15>) をセットして SPI 動作を有効にします。

図 16-2: SPIx モジュールのブロック図 (拡張バッファモード)



レジスタ 16-1: SPIxSTAT: SPIx ステータス / 制御レジスタ

R/W-0	U-0	R/W-0	U-0	U-0	R-0, HSC	R-0, HSC	R-0, HSC
SPIEN	_	SPISIDL	_	_	SPIBEC2	SPIBEC1	SPIBEC0
bit 15							bit 8

R-0,HSC	R/C-0,	R/W-0, HSC	R/W-0	R/W-0	R/W-0	R-0, HSC	R-0, HSC
	HS						
SRMPT	SPIROV	SRXMPT	SISEL2	SISEL1	SISEL0	SPITBF	SPIRBF
bit 7							bit 0

凡例: C=クリア可能ビット HS=ハードウェア セット可能ビット HSC=ハードウェアセット/クリア可能ビット R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し -n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15 SPIEN: SPIx イネーブルビット

1 =モジュールを有効にし、SCK1、SDO1、SDI1、 $\overline{SS1}$ をシリアルポート ピンとして設定する

0 = モジュールを無効にする

bit 14 **未実装:**「0」として読み出し

bit 13 SPISIDL: アイドルモード時停止ビット

1 = デバイスがアイドルモードに移行した時にモジュールの動作を停止する

0 = アイドルモード時もモジュールの動作を継続する

bit 12-11 **未実装:**「0」として読み出し

bit 10-8 SPIBEC<2:0>: SPIx バッファ エレメント カウントビット(拡張バッファモード時に有効)

<u>マスタモード時 :</u> 保留中 SPI 転送の数 <u>スレーブモード時 :</u>

未読み出し SPI 転送の数

bit 7 SRMPT: シフトレジスタ (SPIxSR) エンプティビット (拡張バッファモード時に有効)

1 = SPIx シフトレジスタはエンプティであり送受信可能

0 = SPIx シフトレジスタはエンプティではない

bit 6 SPIROV: 受信オーバーフロー フラグビット

1=新たなバイト/ワードを完全に受信したが破棄した(ユーザソフトウェアはSPIxBUFレジスタ内の既存データをまだ読み出していない)

0 = オーバーフローは発生していない

bit 5 SRXMPT: 受信 FIFO エンプティビット (拡張バッファモード時に有効)

1 = 受信 FIFO はエンプティ

0 = 受信 FIFO はエンプティではない

bit 4-2 SISEL<2:0>: SPIx バッファ割り込みモードビット(拡張バッファモード時に有効)

111 = SPIx 送信バッファがフルになった時 (SPITBF ビットがセットされた時) に割り込む

110 = 最終ビットを SPIxSR にシフトインして TX FIFO がエンプティになった時に割り込む

101 = 最終ビットを SPIxSR からシフトアウトして送信が完了した時に割り込む

100 = 1 データバイトを SPIxSR にシフトインして TX FIFO に1 データバイトの空きができた時に割り込む

011 = SPIx 受信バッファがフルになった時 (SPIRBF ビットがセットされた時) に割り込む

010 = SPIx 受信バッファが 3/4 フル以上になった時に割り込む

001 = 受信バッファでデータが利用可能となった時 (SRMPT ビットがセットされた時) に割り込む

000 = 受信バッファの最終データを読み出してバッファがエンプティになった時 (SRXMPT ビットが セットされた時)に割り込む

レジスタ 16-1: SPIxSTAT: SPIx ステータス / 制御レジスタ (続き)

bit 1 SPITBF: SPIx 送信バッファフル ステータスビット

1 = 送信をまだ開始していない (SPIxTXB はフル)

0 = 送信を開始した (SPIxTXB はエンプティ)

標準バッファモード時:

このビットは、CPU が SPIxBUF に書き込んで、データ SPIxTXB に転送された時に、ハードウェアで自動的にセットされます。

このビットは、SPIx モジュールがデータを SPIxTXB から SPIxSR へ転送した時に、自動的にクリアされます。

拡張バッファモード時:

このビットは、CPU が SPIxBUF に書き込んで、データがバッファの最後の空き位置に転送された時に、 ハードウェアで自動的にセットされます。

このビットは、バッファに 1 データの空きができて CPU が書き込めるようになった時に、ハードウェアで自動的にクリアされます。

bit 0 SPIRBF: SPIx 受信バッファフル ステータスビット

1 = 受信を完了した (SPIxRXB はフル)

0 = 受信を完了していない (SPIxRXB はエンプティ)

標準バッファモード時:

このビットは、SPIx がデータを SPIxSR から SPIxRXB へ転送した時に、自動的にセットされます。 このビットは、CPU が SPIxBUF を読み出した時 (SPIxRXB を読み出した時) に、ハードウェアで自動 的にセットされます。

拡張バッファモード時:

このビットは、SPIx がデータを SPIxSR からバッファへ転送してバッファの最後の未読み出し位置が埋まった時に、ハードウェアで自動的にセットされます。

このビットは、バッファに 1 データの空きができて SPIxSR から 1 データを転送できるようになった時に、ハードウェアで自動的にクリアされます。

レジスタ 16-2: SPIxCON1: SPIx 制御レジスタ 1

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
_	_	_	DISSCK	DISSDO	MODE16	SMP	CKE ⁽¹⁾
bit 15							bit 8

| R/W-0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| SSEN | CKP | MSTEN | SPRE2 | SPRE1 | SPRE0 | PPRE1 | PPRE0 |
| bit 7 | | | | | | | bit 0 |

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-13 **未実装:**「0」として読み出し

bit 12 **DISSCK**: SCK1 ピン ディセーブル ビット (SPI マスタモード時のみ)

1 = 内部 SPI クロックを無効にする(ピンは I/O として機能する)

0 = 内部 SPI クロックを有効にする

bit 11 **DISSDO**: SDO1 ピン ディセーブル ビット

1 = モジュールは SDO1 ピンを使用しない (ピンは I/O として機能する)

0 = モジュールは SDO1 ピンを制御する

bit 10 MODE16: ワード / バイト通信選択ビット

1 = ワード幅 (16 ビット) で通信する

0 = バイト幅 (8 ビット) で通信する

bit 9 SMP: SPIx データ入力サンプル タイミングビット

マスタモード時:

1 = データ出力時間の最後に入力データをサンプリングする

0 = データ出力時間の中間で入力データをサンプリングする

<u>スレーブモード時:</u>

SPIx をスレーブモードで使用する場合、この SMP ビットをクリアする必要があります。

bit 8 **CKE:** SPIx クロックエッジ選択ビット ⁽¹⁾

1 = クロックがアクティブ状態からアイドル状態へ遷移した時にシリアル出力データが変化する (bit 6 参照)

○ = クロックがアイドル状態からアクティブ状態へ遷移した時にシリアル出力データが変化する (bit 6 参照)

bit 7 SSEN: スレーブ選択イネーブルビット(スレーブモード)

1 = SS1 ピンをスレーブモード用に使用する

0 =モジュールは \overline{SSX} ピンを使用しない (ポート機能がこのピンを制御する)

bit 6 CKP: クロック極性選択ビット

1 = クロックのアイドル状態は HIGH レベル、アクティブ状態は LOW レベル

0 = クロックのアイドル状態は LOW レベル、アクティブ状態は HIGH レベル

bit 5 MSTEN: マスタモード イネーブルビット

1=マスタモード

0=スレーブモード

bit 4-2 **SPRE<2:0>:** セカンダリ プリスケール ビット(マスタモード)

111 = セカンダリ プリスケールは 1:1

110 = セカンダリ プリスケールは 2:1

•

•

000 = セカンダリ プリスケールは 8:1

Note 1: フレーム化 SPI モードでは CKE ビットを使用しません。フレーム化 SPI モード (FRMEN = 1) の場合、 ユーザはこのビットを「0」に設定する必要があります。

レジスタ 16-2: SPIxCON1: SPIx 制御レジスタ 1 (続き)

bit 1-0 **PPRE<1:0>:** プライマリ プリスケール ビット(マスタモード)

11 = プライマリ プリスケールは 1:1 10 = プライマリ プリスケールは 4:1 01 = プライマリ プリスケールは 16:1 00 = プライマリ プリスケールは 64:1

Note 1: フレーム化 SPI モードでは CKE ビットを使用しません。フレーム化 SPI モード (FRMEN = 1) の場合、 ユーザはこのビットを「0」に設定する必要があります。

レジスタ 16-3: SPIxCON2: SPIX 制御レジスタ 2

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0
FRMEN	SPIFSD	SPIFPOL	_	_	_	_	_
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
_	_	_	_	_	_	SPIFE	SPIBEN
bit 7							bit 0

凡.例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15 FRMEN: フレーム化 SPIx サポートビット

1 = フレーム化 SPIx サポートを有効にする 0 = フレーム化 SPIx サポートを無効にする

bit 14 SPIFSD: SS1 ピン フレーム同期パルス方向制御ビット

1 = フレーム同期パルス入力(スレーブ) 0 = フレーム同期パルス出力(マスタ)

bit 13 SPIFPOL: フレーム同期パルス極性ビット(フレームモード時のみ)

1 = フレーム同期パルスはアクティブ HIGH 0 = フレーム同期パルスはアクティブ LOW

bit 12-2 **未実装:**「0」として読み出し

bit 1 SPIFE: フレーム同期パルスエッジ選択ビット

1 = フレーム同期パルスは先頭ビットクロックと同時に発生する 0 = フレーム同期パルスは先頭ビットクロックに先行して発生する

bit 0 SPIBEN: 拡張バッファ イネーブルビット

1 = 拡張バッファを有効にする

0 = 拡張バッファを無効にする(レガシーモード)

式 16-1: デバイスクロック周波数と SPI クロック周波数の関係 ⁽¹⁾

FSCK = $\frac{}{ \mathcal{J} \ni A = \mathcal{J} \cup \mathcal{J}$

Note 1: Fcy = Fosc/2 に基づく (Doze モードと PLL は無効)

表 16-1: SCK サンプリング周波数 ^(1,2)

Fcy = 16 MHz			セカンタ	ダ リ プリスケー	-ラ設定	
FCY = 10 MINZ	1:1	2:1	4:1	6:1	8:1	
プライマリ プリスケーラ設定	1:1	無効	8000	4000	2667	2000
	4:1	4000	2000	1000	667	500
	16:1	1000	500	250	167	125
	64:1	250	125	63	42	31
Fcy = 5 MHz						
プライマリ プリスケーラ設定	1:1	5000	2500	1250	833	625
	4:1	1250	625	313	208	156
	16:1	313	156	78	52	39
	64:1	78	39	20	13	10

Note 1: Fcy = Fosc/2 に基づく (Doze モードと PLL は無効)

2: SCK1 周波数の単位は kHz

17.0 I²C™ (INTER-INTEGRATED CIRCUIT™)

Note: 本書は、PIC24 デバイスの中の特定製品グループを対象とし、それらの機能の概要説明を目的としています。従って本書は、包括的な参照資料の提供を意図したものではありません。I²Cの詳細は「PIC24Fファミリーリファレンスマニュアル」のセクション 24.「I²C™ (Inter-Integrated Circuit™)」(DS39702)を参照してください。

 I^2C^{TM} (Inter-Integrated Circuit) モジュールは、他の周辺モジュールまたはマイクロコントローラ デバイスとの通信に使用するシリアル インターフェイスです。通信相手の周辺モジュールには、シリアルデータEEPROM、ディスプレイ ドライバ、A/D コンバータ等が含まれます。

I²C モジュールは下記の機能をサポートします。

- 独立したマスタロジックとスレーブロジック
- ・ 7 ビットと 10 ビットのデバイス アドレッシング
- I²C プロトコルで定義されたジェネラルコール アドレス
- スレーブデータ要求に対するプロセッサの応答を遅延するクロックストレッチング
- 100 kHz と 400 kHz のバス仕様
- ・ 設定可能なアドレス マスキング
- 調停によるメッセージの喪失を防ぐマルチマスタ モード
- アドレスに関係なく全てのメッセージにスレーブを 応答させるバスリピータ モード
- 自動 SCL

図 17-1 に本モジュールのブロック図を示します。

17.1 ピン割り当て変更オプション

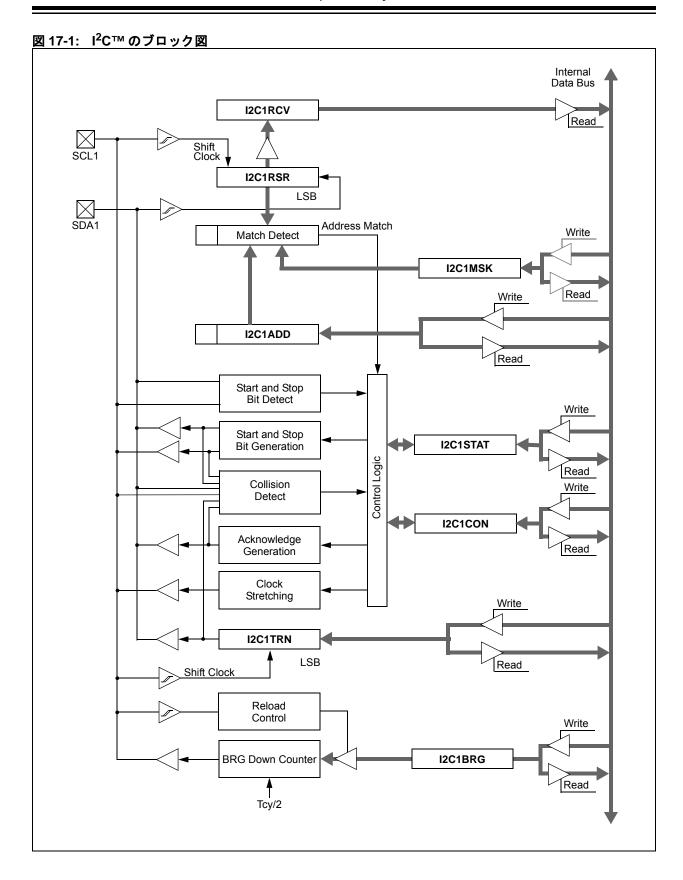
I²C モジュールは、特定のピンに接続されています。 周辺モジュールの柔軟な多重化を可能にするために、 28 ピンデバイスでは、I2C1 モジュールを別のピンに 割り当て変更できます。これらの代替ピンは、デバイ スコンフィグレーション時に SCL1 および SDA1 ピン として割り当てられます。

ピン割り当ては I2C1SEL コンフィグレーション ビットで設定します。このビットを「0」に設定すると、モジュールは SCL1 おおよび SDA1 ピンに多重化されます。

17.2 シングルマスタ環境でマスタとして 通信する

マスタモードでのメッセージ送信の詳細は、通信先デバイスの通信プロトコルによって異なります。標準的なイベントのシーケンスは下記の通りです。

- 1. SDA1 と SCL1 で START 条件を生成する
- 書き込み指示を含む I²C アドレスバイトをスレー ブへ送信する
- 3. スレーブからの肯定応答を待機/確認する
- 4. 先頭データバイト (時に応じコマンドとも呼ばれる)をスレーブへ送信する
- 5. スレーブからの肯定応答を待機/確認する
- 6. シリアルメモリ アドレスの下位バイトをスレーブ へ送信する
- 7. 全てのデータバイトを送信するまでステップ 4 と 5 を繰り返す
- 8. SDA1とSCL1でリピートスタート条件を生成する
- 読み出し指示を含むデバイスアドレス バイトをスレーブへ送信する
- 10. スレーブからの肯定応答を待機 / 確認する
- 11. マスタ受信を有効にしてシリアルメモリ データを 受信する
- 12. データバイトの受信終了時にACKまたはNACK条件を生成する
- 13. SDA1 と SCL1 で STOP 条件を生成する



17.3 バスマスタとして動作する時の baud レートを設定する

baud レート ジェネレータ (BRG) のリロード値の計算には式 17-1 を使用します。

式 17-1: BAUD レート リロード値の計算 ⁽¹⁾

$$FSCL = \frac{FCY}{I2C1BRG + 1 + \frac{FCY}{10,000,000}}$$

または

I2C1BRG =
$$\left(\frac{FCY}{FSCL} - \frac{FCY}{10,000,000}\right) - 1$$

Note 1: Fcy = Fosc/2 に基づく (Doze モードと PLL は無効)

17.4 スレーブアドレスのマスキング

I2C1MSK レジスタ (レジスタ 17-3 参照)は、「ドントケア」として扱うアドレスビット位置を指定します (7 ビットおよび 10 ビットのどちらのアドレッシングモードにも対応)。I2C1MSK レジスタ内のいずれかのビット位置をセット (= 1) すると、スレーブ モジュールは、対応するアドレスビットの値が「0」であっても「1」であっても応答します。例えば、I2C1MSK を「00100000」に設定した場合、スレーブ モジュールはアドレス「0000000」と「00100000」の両方を検出します。

アドレス マスキングを有効にする場合、IPMIEN ビット (I2C1CON<11>) をクリアする事によって、IPMI (Intelligent Platform Management Interface)を無効にする必要があります。

Note:

ピンプロトコルが改変された結果、表 17-2 に示すアドレスが予約済みとなっています。スレーブモードでは、これらのアドレスに応答しません。これには、アドレスマスク設定によってこれらのアドレスが一致する場合も含まれます。

表 17-1: I²C™ クロック周波数 ⁽¹⁾

要求		I2C	1BRG	実際の
システム FSCL	Fcy	(10 進数)	(16 進数)	FSCL
100 kHz	16 MHz	157	9D	100 kHz
100 kHz	8 MHz	78	4E	100 kHz
100 kHz	4 MHz	39	27	99 kHz
400 kHz	16 MHz	37	25	404 kHz
400 kHz	8 MHz	18	12	404 kHz
400 kHz	4 MHz	9	9	385 kHz
400 kHz	2 MHz	4	4	385 kHz
1 MHz	16 MHz	13	D	1.026 MHz
1 MHz	8 MHz	6	6	1.026 MHz
1 MHz	4 MHz	3	3	0.909 MHz

Note 1: Fcy = Fosc/2 に基づく (Doze モードと PLL は無効)

表 17-2· I²C™ の予約済みアドレス ⁽¹⁾

双 17 2. 1	<u> </u>	J: 中3254でアン 1 レス
スレーブ アドレス	R/W ビット	内容
0000 000	0	ジェネラルコール アドレス ⁽²⁾
0000 000	1	START バイト
0000 001	х	Cbus アドレス
0000 010	х	予約済み
0000 011	х	予約済み
0000 1xx	х	HS モード マスタコード
1111 1xx	х	予約済み
1111 0xx	х	10 ビットスレーブ上位バイト ⁽³⁾

Note 1: この表に記載したアドレスビットは、アドレスマスク設定に関係なく、アドレス一致を生成しません。

2: このアドレスには、GCEN = 1 の場合にのみ応答します。

3: このアドレスとの一致は、10 ビット アドレッシング モードの上位バイトに対してのみ発生可能です。

レジスタ 17-1: I2CxCON: I2Cx 制御レジスタ

R/W-0	U-0	R/W-0	R/W-1 HC	R/W-0	R/W-0	R/W-0	R/W-0
I2CEN	_	I2CSIDL	SCLREL	IPMIEN	A10M	DISSLW	SMEN
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0, HC				
GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
bit 7							bit 0

凡例: HC = ハードウェア クリア可能ビット

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15 **I2CEN:** I2Cx イネーブルビット

1 = I2Cx モジュールを有効化し、SDAx および SCLx ピンをシリアルポート ピンとして設定する

0 = I2Cx モジュールを無効化する (ポート機能が全ての I^2C^{TM} ピンを制御する)

bit 14 **未実装:**「0」として読み出し

bit 13 I2CSIDL: アイドルモード時停止ビット

1 = デバイスがアイドルモードに移行するとモジュールは動作を停止する

0=アイドルモード時もモジュールの動作を継続する

bit 12 **SCLREL**: SCLx リリース制御ビット (I²C スレーブとして動作時)

1 = SCLx クロックをリリースする

0 = SCLx クロックを LOW に保持する (クロック ストレッチング)

STREN = 1 の場合:

このビットは読み書き可能です (ソフトウェアは「0」を書き込んでクロック ストレッチングを開始し、「1」を書き込んでクロックをリリースできます)。ハードウェアは、スレーブ送信開始時と、スレーブ受信終了時にこのビットをクリアします。

STREN = 0 の場合:

このビットは読み出しとリセットのみ可能です(ソフトウェアは「1」を書き込んでクロックをリリースする事しかできません)。ハードウェアは、スレーブ送信開始時にこのビットをクリアします。

bit 11 IPMIEN: IPMI (Intelligent Peripheral Management Interface) イネーブルビット

1 = IPMI サポートモードを有効にする(全てのアドレスに応答する)

0 = IPMI サポートモードを無効にする

bit 10 **A10M:** 10 ビット スレーブ アドレッシング ビット

1 = I2CxADD は 10 ビット スレーブアドレス

0 = I2CxADD は 7 ビット スレーブアドレス

bit 9 DISSLW: スルーレート制御ディセーブル ビット

1=スルーレート制御を無効にする

0=スルーレート制御を有効にする

bit 8 SMEN: SMBus 入力レベルビット

1 = SMBus 仕様に準拠する I/O ピンしきい値を有効にする

0 = SMBus 用入力しきい値を無効にする

bit 7 GCEN: ジェネラルコール イネーブルビット (I^2C スレーブとして動作時)

1 = I2CxRSRにジェネラルコールアドレスを受信した時の割り込みを有効にする(モジュールを受信用に有効にする)

0 = ジェネラルコールアドレスを無効にする

bit 6 STREN: SCLx クロック ストレッチ イネーブルビット (I^2 C スレーブとして動作時)

このビットは SCLREL ビットと組み合わせて使用します。

1 = ソフトウェアまたは受信クロック ストレッチングを有効にする

0 = ソフトウェアまたは受信クロック ストレッチングを無効にする

レジスタ 17-1: I2CxCON: I2Cx 制御レジスタ (続き)

bit 5 **ACKDT:** 肯定応答データビット (I²C マスタとして動作時; マスタ受信時に適用)

ソフトウェアが肯定応答シーケンスを開始した時に送信する値です。

1 = 肯定応答時に NACK を送信する

0 = 肯定応答時に ACK を送信する

bit 4 ACKEN: 肯定応答シーケンス イネーブルビット (I²C マスタとして動作時、マスタ受信時に適用)

1 = SDAx および SCLx ピンで肯定応答シーケンスを開始し、ACKDT データビットを送信する (マスタ肯定応答シーケンスの終了時にハードウェアがこのビットをクリアします)

0 = 肯定応答シーケンスを実行していない

bit 3 RCEN: 受信イネーブルビット (I^2C マスタとして動作時)

 $1 = I^2C$ の受信モードを有効にする

(マスタ受信データバイトの第8ビット終了時にハードウェアがこのビットをクリアします)

0 = 受信シーケンスを実行していない

bit 2 **PEN**: STOP 条件イネーブルビット (I²C マスタとして動作時)

1 = SDAx および SCLx ピンで STOP 条件を開始する

(マスタ STOP シーケンス終了時にハードウェアがこのビットをクリアします)

0 = STOP 条件を実行していない

bit 1 RSEN: リピートスタート条件イネーブルビット (I^2C マスタとして動作時)

1 = SDAx および SCLx ピンでリピートスタート条件を開始する

(マスタ リピートスタート シーケンス終了時にハードウェアがこのビットをクリアします)

0 = リピートスタート条件を実行していない

bit 0 **SEN:** START 条件イネーブルビット (I^2C マスタとして動作時)

1 = SDAx および SCLx ピンで START 条件を開始する

(マスタ START シーケンス終了時にハードウェアがこのビットをクリアします)

0 = START 条件を実行していない

レジスタ 17-2: I2CxSTAT: I2Cx ステータス レジスタ

R-0, HSC	R-0, HSC	U-0	U-0	U-0	R/C-0, HS	R-0, HSC	R-0, HSC
ACKSTAT	TRSTAT	_	_	_	BCL	GCSTAT	ADD10
bit 15							bit 8

R/C-0, HS	R/C-0, HS	R-0, HSC	R/C-0, HSC	R/C-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC
IWCOL	I2COV	D/A	Р	S	R/W	RBF	TBF
bit 7							bit 0

凡例: C=クリア可能ビット HS=ハードウェア セット HSC=ハードウェア セット/クリ

可能ビット ア可能ビット

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「O」として読み出し

-n=POR 時の値 1=ビットをセット 0=ビットをクリア x=ビットは未知

bit 15 ACKSTAT: 肯定応答ステータスビット

1 = NACK が検出された 0 = ACK が検出された

肯定応答終了時に、ハードウェアがこのビットをセットまたはクリアします。

bit 14 TRSTAT: 送信ステータスビット (I²C マスタとして動作時、マスタ送信動作に適用)

1 = マスタ送信を実行中 (8 ビット + ACK)

0 = マスタ送信を実行中ではない

ハードウェアは、マスタ送信開始時にこのビットをセットし、スレーブ肯定応答終了時にこのビットを クリアします。

bit 13-11 未実装:「0」として読み出し

bit 10 BCL: マスタバス衝突検出ビット

1 = マスタ動作中にバス衝突を検出した

0 = 衝突を検出していない

バス衝突検出時にハードウェアがこのビットをセットします。

bit 9 GCSTAT: ジェネラルコール ステータスビット

1 = ジェネラルコールアドレスを受信した

0 = ジェネラルコールアドレスを受信しなかった

ハードウェアは、アドレスがジェネラルコール アドレスに一致した時にこのビットをセットし、STOP 検出時にこのビットをクリアします。

bit 8 **ADD10:** 10 ビットアドレス ステータスビット

1 = 10 ビットアドレスが一致した

0 = 10 ビットアドレスは一致しなかった

ハードウェアは一致した 10 ビットアドレスの第 2 バイトが一致した時にこのビットをセットし、STOP条件でこのビットをクリアします。

bit 7 IWCOL: 書き込み衝突検出ビット

 $1 = I^2C$ モジュールがビジーのため、I2CxTRN レジスタへの書き込みに失敗した

0 = 衝突は発生していない

ビジー中に I2CxTRN への書き込みが発生すると、ハードウェアがこのビットをセットします (クリアはソフトウェアによる)。

bit 6 **I2COV**: 受信オーバーフロー フラグビット

1 = I2CxRCV レジスタが先に受信したバイトをまだ保持している時に次のバイトを受信した

0=オーバーフローは発生していない

I2CxRSR から I2CxRCV への転送が発生した時に、ハードウェアがこのビットをセットします (クリアはソフトウェアによる)。

bit 5 D/A: データ / アドレスビット (I^2C スレーブとして動作時)

1 = 直前に受信したバイトはデータであった

0 = 直前に受信したバイトはデバイスアドレスであった

ハードウェアは、デバイスアドレス一致時にこのビットをクリアし、I2CxTRN への書き込みまたはスレーブバイトの受信時にこのビットをセットします。

bit 4 P: STOP ビット

1 = STOP ビットを検出した

0 = STOP ビットを検出しなかった

START、リピートスタート、STOP を検出した時に、ハードウェアがこのビットをセットまたはクリアします。

レジスタ 17-2: I2CxSTAT: I2Cx ステータス レジスタ (続き)

bit 3 **S**: START ビット

1 = START (またはリピートスタート)ビットを検出した

0 = START ビットを検出しなかった

START、リピートスタート、STOP を検出した時に、ハードウェアがこのビットをセットまたはクリアします。

bit 2 **R/W**: 読み出し / 書き込み情報ビット (I²C スレーブとして動作時)

1 = 読み出し - データ転送はスレーブからの出力 0 = 書き込み - データ転送はスレーブへの入力

I²C デバイスアドレス バイトを受信した後に、ハードウェアがこのビットをセットまたはクリアします。

bit 1 RBF: 受信バッファフル ステータスビット

1 = 受信を完了した (I2CxRCV はフル)

0 = 受信を完了していない (I2CxRCV はエンプティ)

ハードウェアは、I2CxRCV に受信バイトが書き込まれた時にこのビットをセットし、ソフトウェアが I2CxRCV を読み出した時にこのビットをクリアします。

bit 0 TBF: 送信バッファフル ステータスビット

1 = 送信中 (I2CxTRN はフル)

0 = 送信完了 (I2CxTRN はエンプティ)

ハードウェアは、ソフトウェアが I2CxTRN レジスタに書き込んだ時にこのビットをセットし、データ送信完了時にこのビットをクリアします。

レジスタ 17-3: I2CxMSK: I2Cx スレーブモード アドレスマスク レジスタ

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
_	_	_	_	_	_	AMSK9	AMSK8
bit 15							bit 8

| R/W-0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| AMSK7 | AMSK6 | AMSK5 | AMSK4 | AMSK3 | AMSK2 | AMSK1 | AMSK0 |
| bit 7 | | | | | | | bit 0 |

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-10 **未実装:**「0」として読み出し

bit 9-0 AMSK<9:0>: アドレスビットx用マスク選択ビット

1 = 受信メッセージ アドレスのビットxのマスキングを有効にする(このビット位置のビット一致を要

求しない)

0 = ビットxのマスキングを無効にする(このビット位置のビット一致を要求する)

レジスタ 17-4: PADCFG1: ピン コンフィグレーション制御レジスタ

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
_	_	_	_	_	_	_	_
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
_	_	SMBUSDEL2	SMBUSDEL1	_	_	_	_
bit 7							bit 0

凡例:

R=読み出し可能ビッ W=書き込み可能ビット U=未実装ビット、「0」として読み出し

卜

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-6 **未実装:**「0」として読み出し

bit 5 SMBUSDEL2: SMBus SDAx 入力遅延選択ビット

1 = I2C2 モジュールを長い SMBus 入力遅延 (公称値 300 ns の遅延) 用に設定する 0 = I2C2 モジュールをレガシーの入力遅延 (公称値 150 ns の遅延) 用に設定する

bit 4 SMBUSDEL1: SMBus SDAx 入力遅延選択ビット

1 = I2C1 モジュールを長い SMBus 入力遅延 (公称値 300 ns の遅延) 用に設定する 0 = I2C1 モジュールをレガシーの入力遅延 (公称値 150 ns の遅延) 用に設定する

bit 3-0 **未実装:**「0」として読み出し

18.0 UART (UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER)

Note: 本書は、PIC24 デバイスの中の特定製品 グループを対象とし、それらの機能の概 要説明を目的としています。従って本書 は、包括的な参照資料の提供を意図した ものではありません。UART の詳細は 「PIC24Fファミリリファレンス マニュア ル」のセクション 21.「UART」(DS39708)

UART モジュールは、この PIC24F デバイスファミリで利用可能なシリアルI/Oモジュールの1つです。UARTはパソコン、LIN/J2602、RS-232、RS-485 インターフェイス等の周辺デバイスと通信可能 \underline{v} 全工重非同期システムです。また、本モジュールは UxCTS ピンとUxRTS ピンによるハードウェア フロー制御オプションをサポートし、IrDA® エンコーダ / デコーダも備えます。UART モジュールの主な特長は以下の通りです。

を参照してください。

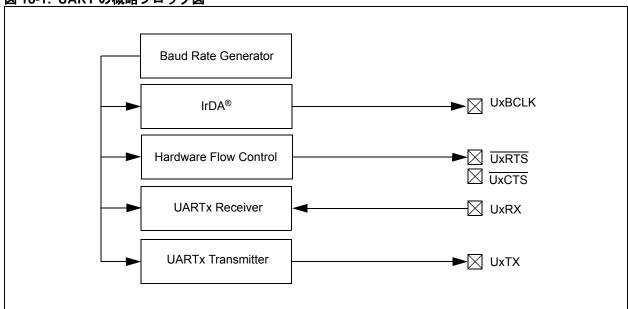
- UxTX および UxRX ピンを介する全二重 8 ビットまたは 9 ビット データ転送
- 偶数、奇数、「なし」のパリティ オプション (8 ビットデータ用)
- 1個または2個のストップビット
- UxCTS および UxRTS ピンを使用するハードウェアフロー制御オプション

- 16ビット プリスケーラ付きbaud レート ジェネレータ (IBRG) を内蔵
- 16 MIPS で 1 Mbps ~ 15 bps の baud レートレンジ
- 4 段先入れ先出し (FIFO) 送信データバッファ
- 4段 FIFO 受信データバッファ
- パリティエラー、フレーミングエラー、バッファ オーバーランエラーの検出
- 9ビットモード アドレスの検出をサポート (第9ビット=1)
- ・ 送信および受信割り込み
- 診断用ループバック モードをサポート
- 同期およびブレーク キャラクタをサポート
- baud レート自動検出をサポート
- IrDA[®] エンコーダ / デコーダロジック
- IrDA サポート用の 16x baud クロック出力

UART の概略ブロック図を図 18-1 に示します。UART モジュールは、以下の主要ハードウェア要素で構成さ れます。

- baud レート ジェネレータ
- ・ 非同期トランスミッタ
- ・ 非同期レシーバ





18.1 UART の baud レート ジェネレータ (BRG)

UARTモジュールは専用の16ビットbaudレートジェネレータ (BRG) を備えます。UxBRG レジスタはフリーランニング16ビットタイマの周期を制御します。式 18-1 に、BRGH = 0 の時の baud レートの計算式を示します。

式 18-1: UART の BAUD レート (BRGH = 0)⁽¹⁾

Baud Rate =
$$\frac{FCY}{16 \cdot (UxBRG + 1)}$$

$$UxBRG = \frac{FCY}{16 \cdot Baud Rate} - 1$$

Note 1: Fcy = Fosc/2 に基づく (Doze モード と PLL は無効)

例 18-1 に、下記条件における baud レート誤差の計算式を示します。

- Fcy = 4 MHz
- 目標 baud レート = 9600

BRGH = 0 とした時、可能な最大 baud レートは Fcy/16 (UxBRG = 0)、可能な最小 baud レートは Fcy/(16 * 65536) です。

式 18-2 に、BRGH = 1 の時の baud レートの計算式を示します。

式 18-2: UART BAUD レート (BRGH = 1)⁽¹⁾

Baud Rate =
$$\frac{FCY}{4 \cdot (UxBRG + 1)}$$

$$UxBRG = \frac{FCY}{4 \cdot Baud Rate} - 1$$

Note 1: Fcy = Fosc/2 に基づく (Doze モード と PLL は無効)

BRGH = 1 とした時、可能な最大 baud レートは Fcy/4 (UxBRG = 0)、可能な最小 baud レートは Fcy/(4 * 65536) です。

UxBRG レジスタに新たな値を書き込むと、BRG タイマがリセット (クリア) されます。これにより、BRG はタイマのオーバーフローを待たずに新たな baud レートで生成を開始します。

式 18-1: BAUD レート誤差の計算 (BRGH = 0)⁽¹⁾

Desired Baud Rate = FCY/(16 (UxBRG + 1))

Solving for UxBRG value:

UxBRG = ((FCY/Desired Baud Rate)/16) - 1

UxBRG = ((4000000/9600)/16) - 1

UxBRG = 25

Calculated Baud Rate = 4000000/(16(25+1))

= 9615

Error = (Calculated Baud Rate – Desired Baud Rate)

Desired Baud Rate = (9615 – 9600)/9600

= 0.16%

Note 1: Fcy = Fosc/2 に基づく (Doze モードと PLL は無効)

18.2 8 ビット データモードの送信方法

- 1. UAR を下記のようにセットアップします。
 - a) データ、パリティ、ストップビットを適切に 設定する
 - b) UxBRG レジスタに適切な baud レート値を書き込む
 - c) 送信割り込みと受信割り込みのイネーブル ビットと優先度ビットを設定する
- 2. UART を有効化します。
- 3. UTXEN ビットをセットします (セット後2サイクルで送信割り込みが発生します)。
- 4. UxTXREG ワードの下位バイトにデータバイトを 書き込みます。値は即座に送信シフトレジスタ (TSR) へ転送され、baud クロックの次の立ち上が りエッジでシリアル ビットストリームがシフト アウトを開始します。
- 5. あるいは、UTXEN = 0 の状態でデータバイトを転送してから UTXEN ビットをセットする事もできます。この場合、UTXEN ビットのセット後即座にシリアル ビットストリームが開始します (baud クロックはクリア状態から開始するため)。
- 6. 割り込み制御ビットUTXISELxの設定に従って送 信割り込みが生成されます。

18.3 9 ビット データモードの送信方法

- 1. 18.2「8 ビット データモードの送信方法」と同様に UART をセットアップします。
- 2. UART を有効化します。
- 3. UTXEN ビットをセットします (セット後2サイクルで送信割り込みが発生します)。
- 4. UxTXREG には必ず 16 ビット値を書き込みます。
- 5. UxTXREG へのワード書き込みにより、9 ビット データの TSR への転送が開始します。シリアル ビットストリームは、baud クロックの最初の立ち 上がりエッジでシフトアウトを開始します。
- 6. 割り込み制御ビット (UTXISELx) の設定に従って 送信割り込みが生成されます。

18.4 ブレークおよび同期キャラクタの送信シーケンス

下記のシーケンスは、ブレークとこれに続く baud レート自動検出同期バイトで構成されたメッセージフレーム ヘッダを送信します。

- 1. UART を必要なモードに設定します。
- 2. UTXEN および UTXBRK ビットをセットします (ブレーク キャラクタの送信をセットアップする)。
- 3. UxTXREG にダミーキャラクタを書き込んで送信 を開始します (この値は無視されます)。
- 4. UxTXREG に 0x55 を書き込みます (送信 FIFO に 同期キャラクタを書き込む)。
- 5. ブレーク送信後に、ハードウェアが UTXBRK ビットをリセットします。この時点で同期キャラクタが送信されます。

18.5 8 ビットまたは 9 ビットモードの データを受信方法

- 18.2「8 ビット データモードの送信方法」と同様に UART をセットアップします。
- 2. UART を有効化します。
- 3. 受信割り込みは、割り込み制御ビット(URXISELx) の設定に従って、1 個または複数個のキャラクタを受信した時に生成されます。
- 4. OERR ビットを読み出して、オーバーラン エラー が発生したかどうかを判定します。 OERR ビット はソフトウェアでリセットする必要があります。
- 5. UxRXREG を読み出します。

UxRXREG キャラクタの読み出しにより、PERR および FERR ビットを含む次のキャラクタが受信 FIFO の 先頭位置へ移動します。

18.6 UxCTS および UxRTS 制御ピンの動作

UARTx 送信クリア (UXCTS) および送信要求 (UXRTS) ピンは、UART モジュールに割り当てられた 2 本のハードウェア制御ピンです。UART は、この 2 本のピンを使用して、片方向モードとフロー制御モードで動作できます。これらのピンは、UART とデータ端末装置 (DTE) 間の送受信を制御します。これらのピンの設定には UXMODE レジスタの UEN<1:0> ビットを使用します。

18.7 赤外線サポート

UART モジュールは、2 種類の赤外線 UART をサポートします。1 つは、外部 IrDA エンコーダ / デコーダ デバイスをサポートする IrDA クロック出力です (レガシーのモジュール サポート)。もう1 つは、IrDA エンコーダ / デコーダの完全な内蔵です。

IrDA モードは 16x baud クロックを要求するため、BRGH ビット (UxMODE<3>) が「0」の時にのみ機能します。

18.7.1 外部 IrDA サポートの IrDA クロック出力

外部 IrDA エンコーダ / デューダ デバイスをサポート するために、UxBCLK ピン (UxRTS ピンと同様) を 16x baud クロック生成用に設定できます。UEN<1:0> = 11 の時、UART モジュールを有効にすると UxBCLK ピンは 16x baud クロックを出力します。これを使用して IrDA コーデックチップをサポートできます。

18.7.2 内蔵 IrDA エンコーダ / デコーダ

UART モジュールは、機能を完備した IrDA エンコーダ / デコーダを、モジュールの一部として実装しています。 内蔵 IrDA エンコーダ / デコーダ機能の有効化には IREN ビット (UxMODE<12>) を使用します。これを有効化 (IREN = 1) した場合、受信ピン (UxRX) は赤外線レシーバからの入力として機能し、送信ピン (UxTX) は赤外線トランスミッタへの出力として機能します。

レジスタ 18-1: UxMODE: UARTx モードレジスタ

R/W-0	U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0 ⁽²⁾	R/W-0 ⁽²⁾
UARTEN	_	USIDL	IREN ⁽¹⁾	RTSMD	_	UEN1	UEN0
bit 15							bit 8

R/C-0, HC	R/W-0	R/W-0, HC	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WAKE	LPBACK	ABAUD	RXINV	BRGH	PDSEL1	PDSEL0	STSEL
bit 7							bit 0

凡例:C = クリア可能ビットHC = ハードウェア クリア可能ビットR = 読み出し可能ビットW = 書き込み可能ビットU = 未実装ビット、「0」として読み出し-n = POR 時の値1 = ビットをセット0 = ビットをクリアx = ビットは未知

bit 15 **UARTEN:** UARTx イネーブルビット

1 =UARTx を有効にする; UARTx は UEN<1:0> の定義に従って全ての UARTx ピンを制御する

0 =UARTx を無効にする; ポートラッチで全ての UARTx ピンを制御する; UARTx の消費電力は最小

bit 14 **未実装:**「0」として読み出し

bit 13 USIDL: アイドルモード時停止ビット

1 = デバイスがアイドルモードに移行した時にモジュールの動作を停止する

0=アイドルモード時もモジュールの動作を継続する

bit 12 **IREN:** IrDA[®] エンコーダ / デコーダ イネーブルビット ⁽¹⁾

1 = IrDA エンコーダ / デコーダを有効にする

0 = IrDA エンコーダ / デコーダを無効にする

bit 11 RTSMD: UxRTS ピンモード選択ビット

1 = UxRTS ピンを片方向モードにする

0 = UxRTS ピンをフロー制御モードにする

bit 10 **未実装:**「0」として読み出し

bit 9-8 **UEN<1:0>:** UARTx イネーブルビット (2)

11 = UxTX、UxRX、UxBCLK ピンを有効にして使用する; UxCTS ピンをポートラッチで制御する

10 = UxTX、UxRX、UxCTS、UxRTS ピンを有効にして使用する

01 = UxTX、UxRX、UxRTS ピンを有効にして使用する: UxCTS ピンをポートラッチで制御する

00 = UxTX ピンと UxRX ピンを有効にして使用する ; UxCTS ピンと UxRTS/UxBCLK ピンをポートラッチで制御する

bit 7 WAKE: スリープモード中スタートビット検出時復帰イネーブルビット

1 = UARTx はUxRX ピンのサンプリングを継続する; 割り込みは立ち下がりエッジで発生する; 次の立ち上がりエッジでこのビットはハードウェアでクリアされる

0 = 復帰は無効

bit 6 LPBACK: UARTx ループバック モード選択ビット

1=ループバック モードを有効にする

0 = ループバック モードを無効にする

bit 5 ABAUD: baud レート自動検出イネーブルビット

1 = 次のキャラクタでbaud レート計測を有効にする(同期フィールド(55h)の受信が必要); 完了時にこのビットはハードウェアでクリアされる

0 = baud レート自動検出を無効にする、または自動検出は完了した

bit 4 RXINV: 受信極性反転ビット

1 = UxRX のアイドル状態は「0」

0 = UxRX のアイドル状態は「1」

Note 1: この機能は 16x BRG モード (BRGH = 0) のみで使用可能です。

2: このビットが利用可能かどうかは、ピンが利用可能かどうかによって決まります。

レジスタ 18-1: UxMODE: UARTx モードレジスタ (続き)

bit 3 **BRGH**: 高 baud レート イネーブルビット

1 = BRG はビット周期あたり 4 クロックを生成する (4x baud クロック、高速モード) 0 = BRG はビット周期あたり 16 クロックを生成する (16x baud クロック、標準モード)

bit 2-1 **PDSEL<1:0>:** パリティ / データ選択ビット

11 = 9 ビットデータ、パリティなし 10 = 8 ビットデータ、奇数パリティ 01 = 8 ビットデータ、偶数パリティ 00 = 8 ビットデータ、パリティなし

bit 0 STSEL: ストップビット選択ビット

1 = 2 個のストップビット 0 = 1 個のストップビット

Note 1: この機能は 16x BRG モード (BRGH = 0) のみで使用可能です。

2: このビットが利用可能かどうかは、ピンが利用可能かどうかによって決まります。

レジスタ 18-2: UxSTA: UARTx ステータス / 制御レジスタ

R/W-0	R/W-0	R/W-0	U-0	R/W-0, HC	R/W-0	R-0, HSC	R-1, HSC
UTXISEL1	UTXINV	UTXISEL0	_	UTXBRK	UTXEN	UTXBF	TRMT
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R-1, HSC	R-0, HSC	R-0, HSC	R/C-0, HS	R-0, HSC
URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA
bit 7							bit 0

凡例: HC = ハードウェア クリア可能ビット

HS = ハードウェア セット可能ビット C = クリア可能ビット HSC = ハードウェア セット/クリア可能ビット

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15,13 UTXISEL<1:0>: 送信割り込みモード選択ビット

11 = 予約済み(使用しない)

10 = キャラクタを送信シフトレジスタ (TSR) に転送した結果、送信バッファがエンプティになった時に割り込む

01 = 最後のキャラクタを送信シフトレジスタからシフトアウトして全ての送信動作が完了した時に割り込む

00=1キャラクタを送信シフトレジスタに転送した時(送信バッファに少なくとも1キャラクタの空きができた時)に割り込む

bit 14 UTXINV: IrDA® エンコーダ極性反転ビット

IREN = 0 の場合:

1 = UxTX アイドル時に「0」

0 = UxTX アイドル時に「1」

<u>IREN = 1 の場合:</u>

1 = UxTX アイドル時に「1」

0 = UxTX アイドル時に「0」

bit 12 **未実装:**「0」として読み出し

bit 11 UTXBRK: 送信ブレークビット

1 =次の送信で同期ブレークを送信する (スタートビット、12 個の「0」ビット、ストップビットの順); このビットは完了時にハードウェアでクリアされる

0 = 同期ブレーク送信は無効または完了した

bit 10 UTXEN: 送信イネーブルビット

1 = 送信を有効にする; UARTx が UxTX ピンを制御する

0 = 送信を無効にする;保留中の送信を全て中止してバッファをリセットする;PORT レジスタが UxTX ピンを制御する

bit 9 UTXBF: 送信バッファフル ステータスビット (読み出し専用)

1 = 送信バッファはフル

0 = 送信バッファはフルではない;少なくともあと1個のキャラクタを書き込める

bit 8 TRMT: 送信シフトレジスタ エンプティビット (読み出し専用)

1 = 送信シフトレジスタも送信バッファもエンプティ(直前の送信は完了した)

0 = 送信シフトレジスタはエンプティではない;送信は進行中または待機中

bit 7-6 **URXISEL<1:0>:** 受信割り込みモード選択ビット

11 =RSR転送で受信バッファがフル(4個のデータ キャラクタを格納)になった時に割り込みをセット する

10 =RSR 転送で受信バッファが 3/4 フル (3 個のデータ キャラクタを格納) になった時に割り込みをセットする

0x =1個のキャラクタを受信してRSRから受信バッファに転送した時(受信バッファに1個または複数個のキャラクタが格納されている時)に割り込みをセットする

レジスタ 18-2: UxSTA: UARTx ステータス / 制御レジスタ (続き)

- bit 5 **ADDEN**: アドレス キャラクタ検出ビット (受信データの bit 8 = 1)
 - 1 = アドレス検出モードを有効にする;9ビットモードが選択されていない場合、このビットは効果を 持たない
 - 0 = アドレス検出モードを無効にする
- bit 4 RIDLE: レシーバアイドル ビット (読み出し専用)
 - 1 = レシーバはアイドル中
 - 0 = レシーバはアクティブ
- bit 3 PERR: パリティエラー ステータスビット (読み出し専用)
 - 1 = 現在のキャラクタ (受信 FIFO の先頭キャラクタ) でパリティエラーを検出した
 - 0 = パリティエラーを検出していない
- bit 2 FERR: フレーミング エラー ステータスビット (読み出し専用)
 - 1 = 現在のキャラクタ (受信 FIFO の先頭キャラクタ)でフレーミング エラーを検出した
 - 0 = フレーミング エラーを検出していない
- bit 1 OERR: 受信バッファ オーバーラン エラー ステータスビット (クリアと読み出しのみ可能)
 - 1 = 受信バッファはオーバーフローした
 - 0 = 受信バッファはオーバーフローしていない (セットされている OERR ビットをクリアすると $(1 \rightarrow 0)$ 、受信バッファと RSR がエンプティ状態にリセットされます)
- bit 0 URXDA: 受信バッファ データあり / なしステータスビット (読み出し専用)
 - 1 = 受信バッファはデータを格納している;少なくとも1個のキャラクタを読み出せる
 - 0 = 受信バッファはエンプティ

レジスタ 18-3: UxTXREG: UARTx 送信レジスタ

U-x	U-x	U-x	U-x	U-x	U-x	U-x	W-x
_	_	_	_	_	_	_	UTX8
bit 15							bit 8

W-x	W-x	W-x	W-x	W-x	W-x	W-x	W-x
UTX7	UTX6	UTX5	UTX4	UTX3	UTX2	UTX1	UTX0
bit 7							bit 0

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-9 **未実装:**「0」として読み出し

bit 8 UTX8: 送信キャラクタ データビット (9 ビットモード時)

bit 7-0 UTX<7:0>: 送信キャラクタ データビット

レジスタ 18-4: UxRXREG: UARTx 受信レジスタ

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R-0, HSC
_	_	_	_	_	_	_	URX8
bit 15							bit 8

| R-0, HSC |
|----------|----------|----------|----------|----------|----------|----------|----------|
| URX7 | URX6 | URX5 | URX4 | URX3 | URX2 | URX1 | URX0 |
| bit 7 | | | | | | | bit 0 |

凡例: HSC = ハードウェア セット / クリア可能ビット

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「O」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-9 **未実装:**「0」として読み出し

bit 8 **URX8:** 受信キャラクタ データビット (9 ビットモード時)

bit 7-0 URX<7:0>: 受信キャラクタ データビット

19.0 リアルタイム クロック / カレンダ (RTCC)

Note: 本書は、PIC24 デバイスの中の特定製品グループを対象とし、それらの機能の概要説明を目的としています。従って本書は、包括的な参照資料の提供を意図したものではありません。リアルタイム クロック / カレンダの詳細は「PIC24F ファミリ リファレンス マニュアル」のセクション 29.「リアルタイム クロック / カレンダ (RTCC)」(DS39696) を参照してください。

RTCC は、校正可能なリアルタイム クロック / カレンダ (RTCC) 機能を提供します。

RTCC の主な特長は以下の通りです。

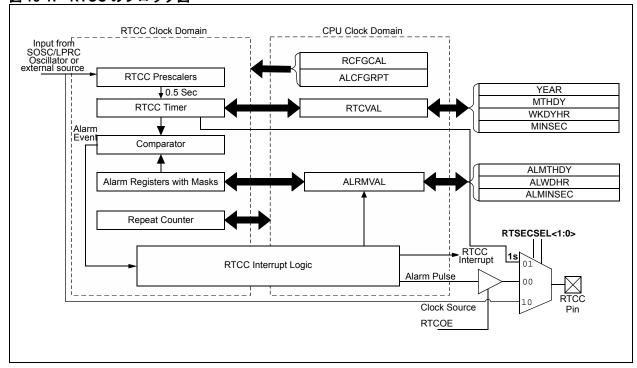
- ディープスリープモード中の動作
- ・ クロック源を選択可能
- 時刻(時/分/秒)を24時間フォーマットで出力
- 0.5 秒分解能
- ・ 日付(曜日/日/月/年)を出力
- アラームの設定単位: 1/2 秒、1 秒、10 秒、1 分、10 分、1 時間、1 日、1 週間、1 月、1 年
- カウンタのデクリメントによるアラームのリピート
- 無制限にアラームを繰り返すチャイム

- ・ 2000 ~ 2099 年のうるう年補正
- BCD 表記によるソフトウェア オーバーヘッドの軽減
- バッテリ駆動による長期動作用に最適化
- 32.768 kHz クロック水晶振動子 (32K INTRC) 周波数のユーザ校正 (周期的に自動調整)
- ・ 秒未満の同期
- 精度± 2.64 秒 / 月以内の校正
- 水晶振動子を 260 ppm の精度まで校正
- CPU に負荷をかけずに外部デバイスを周期的に復帰 (外部電源制御)
- 外部回路を制御するための電源制御出力
- 校正を 15 秒毎に実施
- 下記のいずれかのクロック源を使用して動作
 - 外部リアルタイムクロック (32.768 kHz)
 - 内部 LPRC クロック (31.25 kHz)
 - 50 Hz または 60 Hz の外部入力

19.1 RTCC の源クロック

RTCC モジュールの参照クロックには、SOSC 水晶振動子、LPRC 内部オシレータ、外部 50 Hz/60 Hz 電源ライン入力のいずれかを選択できます。これにより、システム要件に応じてコスト、精度、消費電力を最適にバランスさせる事ができます。

図 19-1: RTCC のブロック図



19.2 RTCC モジュール レジスタ

RTCC モジュールのレジスタは、下記の3種類に分類されます。

- RTCC 制御レジスタ
- ・ RTCC 値レジスタ
- アラーム値レジスタ

19.2.1 レジスタの割り当て

レジスタ インターフェイスを限定するために、RTCC タイマレジスタとアラーム時間レジスタは対応するレジスタポインタを介してアクセスされます。RTCC値レジスタウィンドウ(RTCVALHとRTCVALL)は、RTCPTR ビット (RCFGCAL<9:8>)を使用して必要なタイマレジスタペアを選択します(表 19-1参照)。RTCVALH バイトを書き込むと、RTCC ポインタ値(RTCPTR<1:0> ビット)は1つデクリメントします(「00」に達するまで)。ポインタ値が「00」に達すると、RTCVALHとRTCVALを介してLMINUTES値とSECONDS値にアクセス可能となります。ポインタ値を手動で変更すると、このアクセスはできなくなります。

表 19-1: RTCVAL レジスタの割り当て

RTCPTR<1:0>	RTCC 値レジス	くタ ウィンドウ
KICFIK(1.0)	RTCVAL<15:8>	RTCVAL<7:0>
00	MINUTES	SECONDS
01	WEEKDAY	HOURS
10	MONTH	DAY
11		YEAR

アラーム値レジスタ ウィンドウ (ALRMVALH と ALRMVALL)は、ALRMPTR ビット (ALCFGRPT<9:8>) を使用して必要なアラームレジスタ ペアを選択します (表 19-2 参照)。

ALRMVALH バイトを書き込むと、アラームポインタ値 (ALRMPTR<1:0> ビット) は1つデクリメントします (「00」に達するまで)。ポインタ値が「00」に達すると、ALRMVALHとALRMVALLを介して ALRMMIN

値と ALRMSEC 値にアクセス可能となります。ポイン タ値を手動で変更すると、このアクセスはできなくな ります。

表 19-2: ALRMVAL レジスタの割り当て

ALRMPTR	アラーム値レジスタ ウィンドウ				
<1:0>	ALRMVAL<15:8>	ALRMVAL<7:0>			
0.0	ALRMMIN	ALRMSEC			
01	ALRMWD	ALRMHR			
10	ALRMMNTH	ALRMDAY			
11	PWCSTAB	PWCSAMP			

16 ビットコアは、8 ビット読み出し動作と 16 ビット読み出し動作を区別しないため、ALRMVALH またはALRMVALL をバイト読み出しすると、ALRMPTR<1:0>値がデクリメントする事に注意が必要です。同様に、RTCVALHバイトまたはRTCVALLバイトを読み出すと、RTCPTR<1:0> がデクリメントします。

Note: この注意は読み出し時にのみ必要です。書き込み動作には関係しません。

19.2.2 書き込みロック

RTCC タイマレジスタのいずれかに書き込みを実行するには、RTCWREN ビット (RTCPWC<13>) をセットする必要があります (例 19-1 参照)。

Note: タイマへの偶発的な書き込みを防ぐために、必要な時以外は RTCWREN ビット (RCFGCAL<13>) を常時クリア状態に維持する事を推奨します。RTCWREN は、55h/AA シーケンス後 1 命令サイクルの時間ウィンドウ内でセットする必要があります。従って、コードは例 19-1 の手順に従う事を推奨します。

19.2.3 RTCC クロック源の選択

RTCCには4種類の参照源クロック オプションを選択できます。この選択には RTCCSEL<1:0> ビットを使用します (00 = セカンダリ オシレータ、01 = LPRC、10 = 50 Hz 外部クロック、11 = 60 Hz 外部クロック)。

例 19-1: RTCWREN ビットのセット

```
asm volatile("push w7");
asm volatile("push w8");
asm volatile("disi #5");
asm volatile("mov #0x55, w7");
asm volatile("mov #0x55, w7");
asm volatile("mov #0xAA, w8");
asm volatile("mov w8, _NVMKEY");
asm volatile("bset _RCFGCAL, #13"); //set the RTCWREN bit
asm volatile("pop w8");
asm volatile("pop w8");
asm volatile("pop w8");
```

19.2.4 RTCC 制御レジスタ

レジスタ 19-1: RCFGCAL: RTCC 校正 / コンフィグレーションレジスタ ⁽¹⁾

R/W-0	U-0	R/W-0	R-0, HSC	R-0, HSC	R/W-0	R/W-0	R/W-0
RTCEN ⁽²⁾	_	RTCWREN	RTCSYNC	HALFSEC ⁽³⁾	RTCOE	RTCPTR1	RTCPTR0
bit 15		_					bit 8

| R/W-0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| CAL7 | CAL6 | CAL5 | CAL4 | CAL3 | CAL2 | CAL1 | CAL0 |
| bit 7 | | | | | | | bit 0 |

凡例 HSC = ハードウェア セット / クリア可能ビット

 R = 読み出し可能ビット
 W = 書き込み可能ビット
 U = 未実装ビット、「0」として読み出し

 -n = POR 時の値
 1 = ビットをセット
 0 = ビットをクリア
 x = ビットは未知

bit 15 RTCEN: RTCC イネーブルビット (2)

1 = RTCC モジュールを有効にする

0 = RTCC モジュールを無効にする

bit 14 **未実装:**「0」として読み出し

bit 13 RTCWREN: RTCC 値レジスタ書き込みイネーブルビット

1 = ユーザは RTCVALH および RTCVALL レジスタに書き込める 0 = ユーザは RTCVALH および RTCVALL レジスタに書き込めない

bit 12 RTCSYNC: RTCC 値レジスタ読み出し同期ビット

1 = RTCVALH、RTCVALL、ALCFGRPT レジスタの読み出し中にロールオーバー リップルによって 値が変化する可能性がある(結果として無効なデータ読み出しとなる可能性がある)

このレジスタを2回読み出してデータが同じであれば、そのデータは有効であるとみなせます。

0 = RTCVALH、RTCVALL、ALCFGRPT レジスタは、ロールオーバー リップルを気にせずに読み出せる

bit 11 **HALFSEC**: 1/2 秒ステータスビット (3)

1 = 1 秒の後半

0 = 1 秒の前半

bit 10 RTCOE: RTCC 出力イネーブルビット

1 = RTCC モジュールを有効にする

0 = RTCC モジュールを無効にする

bit 9-8 RTCPTR<1:0>: RTCC 値レジスタ ウィンドウ ポインタビット

このポインタビットは、RTCVALH および RTCVALL レジスタの読み出し時に、対応する RTCC 値レジスタを指します。RTCPTR<1:0> の値は、RTCVALH を読み書きするたびにデクリメントします (「00」に達するまで)。

RTCVAL<15:8>:

00 = 分

01 = 曜日

10 = 月

11 = 予約済み

RTCVAL<7:0>:

00 = 秒

01 = 時

10 = 日

11 **= 年**

Note 1: RCFGCAL レジスタは POR にのみ影響されます。

2: RTCEN ビットへの書き込みは、RTCWREN = 1 の時にのみ可能です。

3: このビットは読み出し専用です。このビットは、MINSEC レジスタの下位バイトへの書き込み時に「0」 にクリアされます。

レジスタ 19-1: RCFGCAL: RTCC 校正 / コンフィグレーションレジスタ ⁽¹⁾ (続き)

bit 7-0 **CAL<7:0>:** RTCC ドリフト校正ビット

011111111 = 正方向の最大調整値; 毎分 508 個の RTC クロックパルスを追加する

.

01111111 = 正方向の最小調整値; 毎分 4 個の RTC クロックパルスを追加する

00000000 = 調整しない

11111111 = 負方向の最小調整値: 毎分4個の RTC クロックパルスを削減する

.

10000000 = 負方向の最大調整値;毎分 512 個の RTC クロックパルスを削減する

Note 1: RCFGCAL レジスタは POR にのみ影響されます。

2: RTCEN ビットへの書き込みは、RTCWREN = 1 の時にのみ可能です。

3: このビットは読み出し専用です。このビットは、MINSEC レジスタの下位バイトへの書き込み時に「0」にクリアされます。

レジスタ 19-2: RTCPWC: RTCC コンフィグレーション レジスタ 2⁽¹⁾

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PWCEN	PWCPOL	PWCCPRE	PWCSPRE	RTCCLK1 ⁽²⁾	RTCCLK0 ⁽²⁾	RTCOUT1	RTCOUT0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	_	_	_	_	_	_	_
bit 7							bit 0

凡例

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15 PWCEN: 電源制御イネーブルビット

1=電源制御を有効にする

0=電源制御を無効にする

bit 14 PWCPOL: 電源制御極性ビット

1 = 電源制御出力をアクティブ HIGH にする

0 = 電源制御出力をアクティブ LOW にする

bit 13 **PWCCPRE:** 電源制御の制御 / 安定性プリスケーラ ビット

1 = PWC 安定性ウィンドウ クロックに 2 分周した RTCC 源クロックを使用する

0 = PWC 安定性ウィンドウ クロックに RTCC 源クロックをそのまま使用する

bit 12 PWCSPRE: 電源制御サンプル プリスケーラ ビット

1 = PWC サンプル ウィンドウ クロックに 2 分周した RTCC 源クロックを使用する

0 = PWC サンプル ウィンドウ クロックに RTCC 源クロックをそのまま使用する

bit 11-10 RTCCLK<1:0>: RTCC クロック選択ビット (2)

このビットは、内部 RTCC クロックの源クロックを選択します。これは全ての RTCC タイマ動作に使用されます。

00 = 外部セカンダリ オシレータ (SOSC)

01 = 内部 LPRC オシレータ

10 = 外部電源ライン - 50 Hz

11 = 外部電源ライン - 60 Hz

bit 9-8 RTCOUT<1:0>: RTCC 出力選択ビット

このビットは、RTCC ピン出力のソースを指定します。

00 = RTCC アラームパルス

01 = RTCC 秒クロック

10 = RTCC クロック

11 = 電源制御

bit 7-0 **未実装:**「0」として読み出し

Note 1: RTCPWC レジスタは POR にのみ影響されます。

2: これらのレジスタビットの値を変更した時、RTCC内のクロックプリスケーラを正しくリセットするために、秒値レジスタにも書き込む必要があります。

レジスタ 19-3: ALCFGRPT: アラーム コンフィグレーション レジスタ

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ALRMEN	CHIME	AMASK3	AMASK2	AMASK1	AMASK0	ALRMPTR1	ALRMPTR0
bit 15							bit 8

| R/W-0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| ARPT7 | ARPT6 | ARPT5 | ARPT4 | ARPT3 | ARPT2 | ARPT1 | ARPT0 |
| bit 7 | | | | | | | bit 0 |

凡例

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、[0] として読み出し [-n] = POR 時の値 [1] = ビットをセット [0] = ビットをクリア [x] = ビットは未知

bit 15 ALRMEN: アラーム イネーブルビット

1 = アラームを有効にする (ARPT<7:0> = 00h かつ CHIME = 0 の場合、このビットはアラームイベント後に自動的にクリアされます)

0 = アラームを無効にする

bit 14 CHIME: チャイム イネーブルビット

1 = チャイムを有効にする (ARPT<7:0> ビットは 00h から FFh ヘロールオーバーする)

0 = チャイムを無効にする (ARPT<7:0> ビットは 00h で停止)

bit 13-10 AMASK<3:0>: アラームマスク コンフィグレーション ビット

0000 = 毎 1/2 秒

0001 = 毎秒

0010 = 毎10秒

0011 = 毎分

0100 = 毎10分

0101 = 毎時

0110 = 毎日

0111 = 毎週

1000 = 毎月

1001 = 毎年(2月29日に設定した場合は毎4年)

101x = 予約済み - 未使用

11xx = 予約済み - 未使用

bit 9-8 ALRMPTR<1:0>: アラーム値レジスタ ウィンドウ ポインタビット

このポインタビットは、ALRMVALH および ALRMVALL レジスタの読み出し時に、対応するアラーム値レジスタを指します。ALRMPTR<1:0> の値は、ALRMVALH を読み書きするたびにデクリメントします(「00」に達するまで)。

ALRMVAL<15:8>:

00 = ALRMMIN

01 = ALRMWD

10 = ALRMMNTH

11 = 未実装

ALRMVAL<7:0>:

00 = ALRMSEC

01 = ALRMHR

10 = ALRMDAY

11 = 未実装

bit 7-0 ARPT<7:0>: アラーム リピートカウンタ値ビット

111111111 = アラームを 255 回リピートする

.

00000000 = アラームをリピートしない

カウンタは毎回のアラームイベントでデクリメントします。CHIME = 1 ではない場合、カウンタ値は

00h から FFh ヘロールオーバーしません。

19.2.5 RTCVAL レジスタの割り当て

レジスタ 19-4: YEAR: 年値レジスタ ⁽¹⁾

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
_	_	_	_	_	_	_	
bit 15							bit 8

| R/W-x |
|--------|--------|--------|--------|--------|--------|--------|--------|
| YRTEN3 | YRTEN2 | YRTEN2 | YRTEN1 | YRONE3 | YRONE2 | YRONE1 | YRONE0 |
| bit 7 | | | | | | | bit 0 |

凡例

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し -n=POR 時の値 1=ビットをセット 0=ビットをクリア x=ビットは未知

bit 15-8 **未実装:**「0」として読み出し

bit 7-4 YRTEN<3:0>: 十進表記「年」の 10 の桁のバイナリ値ビット

0~9の値を格納します。

bit 3-0 YRONE<3:0>: 十進表記「年」の1の桁のバイナリ値ビット

0~9の値を格納します。

Note 1: 年値レジスタへの書き込みは、RTCWREN = 1 の時にのみ可能です。

レジスタ 19-5: MTHDY: 月 / 日値レジスタ ⁽¹⁾

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
_	_	_	MTHTEN0	MTHONE3	MTHONE2	MTHONE1	MTHONE0
bit 15							bit 8

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
_	_	DAYTEN1	DAYTEN0	DAYONE3	DAYONE2	DAYONE1	DAYONE0
bit 7							bit 0

凡例

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し -n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-13 **未実装:**「0」として読み出し

bit 12 MTHTENO: 十進表記「月」の 10 の桁のバイナリ値ビット

「0」または「1」を格納します。

bit 11-8 MTHONE<3:0>: 十進表記「月」の1の桁のバイナリ値ビット

0~9の値を格納します。

bit 7-6 **未実装:**「0」として読み出し

bit 5-4 DAYTEN<1:0>: 十進表記「日」の 10 の桁のバイナリ値ビット

0~3の値を格納します。

bit 3-0 **DAYONE<3:0>:** 十進表記「日」の1の桁のバイナリ値ビット

0~9の値を格納します。

Note 1: このレジスタへの書き込みは、RTCWREN = 1 の時にのみ可能です。

レジスタ 19-6: WKDYHR: 曜日 / 時値レジスタ ⁽¹⁾

U-0	U-0	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x
_	_	_	_	_	WDAY2	WDAY1	WDAY0
bit 15							bit 8

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
_	_	HRTEN1	HRTEN0	HRONE3	HRONE2	HRONE1	HRONE0
bit 7							bit 0

凡例

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「O」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-11 **未実装:**「0」として読み出し

bit 10-8 WDAY<2:0>: 十進表記「曜日」のバイナリ値ビット

0~6の値を格納します。

bit 7-6 **未実装:**「0」として読み出し

bit 5-4 **HRTEN<1:0>:** 十進表記「時」の 10 の桁のバイナリ値ビット

0~2の値を格納します。

bit 3-0 **HRONE<3:0>:** 十進表記「時」の1の桁のバイナリ値ビット

0~9の値を格納します。

Note 1: このレジスタへの書き込みは、RTCWREN = 1 の時にのみ可能です。

レジスタ 19-7: MINSEC: 分 / 秒値レジスタ

U-0	R/W-x						
_	MINTEN2	MINTEN1	MINTEN0	MINONE3	MINONE2	MINONE1	MINONE0
bit 15							bit 8

U-0	R/W-x						
_	SECTEN2	SECTEN1	SECTEN0	SECONE3	SECONE2	SECONE1	SECONE0
bit 7							bit 0

凡例

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15 **未実装:**「0」として読み出し

bit 14-12 MINTEN<2:0>: 十進表記「分」の 10 の桁のバイナリ値ビット

0~5の値を格納します。

bit 11-8 MINONE<3:0>: 十進表記「分」の1の桁のバイナリ値ビット

0~9の値を格納します。

bit 7 **未実装:**「0」として読み出し

bit 6-4 **SECTEN<2:0>:** 十進表記「秒」の 10 の桁のバイナリ値ビット

0~5の値を格納します。

bit 3-0 **SECONE<3:0>:** 十進表記「秒」の1の桁のバイナリ値ビット

0~9の値を格納します。

19.2.6 ALRMVAL レジスタの割り当て

レジスタ 19-8: ALMTHDY: アラーム月 / 日値レジスタ ⁽¹⁾

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
_	_		MTHTEN0	MTHONE3	MTHONE2	MTHONE1	MTHONE0
bit 15							bit 8

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
_	_	DAYTEN1	DAYTEN0	DAYONE3	DAYONE2	DAYONE1	DAYONE0
bit 7							bit 0

凡例

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-13 **未実装:**「0」として読み出し

bit 12 MTHTENO: 十進表記「月」の 10 の桁のバイナリ値ビット

「0」または「1」を格納します。

bit 11-8 MTHONE<3:0>: 十進表記「月」の1の桁のバイナリ値ビット

0~9の値を格納します。

bit 7-6 **未実装:**「0」として読み出し

bit 5-4 DAYTEN<1:0>: 十進表記「日」の 10 の桁のバイナリ値ビット

0~3の値を格納します。

bit 3-0 **DAYONE<3:0>:** 十進表記「日」の1の桁のバイナリ値ビット

0~9の値を格納します。

Note 1: このレジスタへの書き込みは、RTCWREN = 1 の時にのみ可能です。

レジスタ 19-9: ALWDHR: アラーム曜日 / 時値レジスタ ⁽¹⁾

U-0	U-0	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x
_	_	_	_	_	WDAY2	WDAY1	WDAY0
bit 15							bit 8

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
_	_	HRTEN1	HRTEN0	HRONE3	HRONE2	HRONE1	HRONE0
bit 7							bit 0

凡例

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-11 **未実装:**「0」として読み出し

bit 10-8 WDAY<2:0>: 十進表記「曜日」のバイナリ値ビット

0~6の値を格納します。

bit 7-6 **未実装:**「0」として読み出し

bit 5-4 HRTEN<1:0>: 十進表記「時」の 10 の桁のバイナリ値ビット

0~2の値を格納します。

bit 3-0 **HRONE<3:0>:** 十進表記「時」の1の桁のバイナリ値ビット

0~9の値を格納します。

Note 1: このレジスタへの書き込みは、RTCWREN = 1 の時にのみ可能です。

レジスタ 19-10: ALMINSEC: アラーム分 / 秒値レジスタ

U-0	R/W-x						
_	MINTEN2	MINTEN1	MINTEN0	MINONE3	MINONE2	MINONE1	MINONE0
bit 15							bit 8

U-0	R/W-x						
_	SECTEN2	SECTEN1	SECTEN0	SECONE3	SECONE2	SECONE1	SECONE0
bit 7							bit 0

凡例

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「O」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15 **未実装:**「0」として読み出し

bit 14-12 MINTEN<2:0>: 十進表記「分」の 10 の桁のバイナリ値ビット

0~5の値を格納します。

bit 11-8 MINONE<3:0>: 十進表記「分」の1の桁のバイナリ値ビット

0~9の値を格納します。

bit 7 **未実装:**「0」として読み出し

bit 6-4 **SECTEN<2:0>:** 十進表記「秒」の 10 の桁のバイナリ値ビット

0~5の値を格納します。

bit 3-0 **SECONE<3:0>:** 十進表記「秒」の1の桁のバイナリ値ビット

0~9の値を格納します。

レジスタ 19-11: RTCCSWT: 制御 / サンプル ウィンドウ タイマレジスタ ⁽¹⁾

| R/W-x |
|----------|----------|----------|----------|----------|----------|----------|----------|
| PWCSTAB7 | PWCSTAB6 | PWCSTAB5 | PWCSTAB4 | PWCSTAB3 | PWCSTAB2 | PWCSTAB1 | PWCSTAB0 |
| bit 15 | | | | | | | bit 8 |

| R/W-x |
|----------|----------|----------|----------|----------|----------|----------|----------|
| PWCSAMP7 | PWCSAMP6 | PWCSAMP5 | PWCSAMP4 | PWCSAMP3 | PWCSAMP2 | PWCSAMP1 | PWCSAMP0 |
| bit 7 | | | | | | | bit 0 |

凡例

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、 $\begin{bmatrix} 0 \end{bmatrix}$ として読み出し $\begin{bmatrix} -n = POR \\ \end{bmatrix}$ 明の値 $\begin{bmatrix} 1 = ビットをセット \\ 0 = ビットをクリア \\ \end{bmatrix}$ $\begin{bmatrix} x = ビットは未知 \\ \end{bmatrix}$

bit 15-8 PWCSTAB<7:0>: PWM 安定性ウィンドウ タイマビット

11111111 = 安定性ウィンドウは 255 TPWCCLK クロック周期

. 00000000 = 安定性ウィンドウは 0 TPWCCLK クロック周期 サンプル ウィンドウはアラームイベント トリガ時に開始します。

PWCEN = 1 の時、安定性ウィンドウ タイマは、毎回のアラームイベントでカウントを開始します。

bit 7-0 **PWCSAMP<7:0>:** PWM サンプル ウィンドウ タイマビット

11111111 = サンプル ウィンドウを常時有効にする (PWCEN = 0 でも有効)

111111110 = サンプル ウィンドウは 254 TPWCCLK クロック周期

.

00000000 = サンプル ウィンドウは 0 TPWCCLK クロック周期

PWCEN = 1 の時、サンプル ウィンドウ タイマは安定性ウィンドウ終了時にカウントを開始します。 PWCEN = 1 の時、PWCSTAB<7:0> = 0 であれば、サンプル ウィンドウ タイマは毎回のアラームイベントでカウントを開始します。

Note 1: このレジスタへの書き込みは、RTCWREN = 1 の時にのみ可能です。

19.3 校正

リアルタイム水晶振動子入力は、周期的な自動調整機能を使用して校正できます。正しく校正する事により、RTCC の誤差を 3 秒 / 月未満に抑える事ができます。校正を正しく行うには、クロック周波数誤差をパルス数誤差として求め、その値を RCFGCAL レジスタの下位バイトに書き込む必要があります。 RCFGCAL の下位バイトに書き込まれた 8 ビット符号付き値を 4 倍した値が、1 分間に 1 回の頻度で RTCC タイマ値に加算または減算されます。 RTCC 校正の手順は下記の通りです。

- デバイスの別のタイマリソースを使用して、 32.768 kHz 水晶振動子の周波数誤差を特定する 必要があります。
- 2. この周波数誤差を、毎分のクロックパルス数誤差に換算します。
- 3. a) オシレータ周波数が理想値よりも高い場合、ステップ2の結果は負数となり、RCFGCAL レジスタには負の値を書き込む必要があります。これにより、指定したクロックパルス数が、タイマカウンタから毎分1回減算されます。

b) オシレータ周波数が理想値よりも低い場合、ステップ2の結果は正数となり、RCFGCAL レジスタには正の値を書き込む必要があります。これにより、指定したクロックパルス数が、タイマカウンタに毎分1回加算されます。

式 19-1:

(理想周波数 † - 実測周波数)* 60 = 毎分クロック数

† 理想周波数 = 32,768 Hz

RCFGCAL レジスタの下位バイトへの書き込みは、タイマが OFF の時、または、秒バルスの立ち上がりエッジ直後 (ただし SECONDS = 00、15、30、45 の時を除く)に発生する必要があります。これは、RTCC の自動調整が 15 秒間隔で発生するためです。

Note: 誤差値に水晶振動子の初期誤差(温度ドリフト、経時変化ドリフト)を含めるかどうかは、ユーザの自由です。

19.4 アラーム

- 1/2 秒から 1 年まで設定可能
- ALRMEN ビット (ALCFGRPT<15>) で有効化
- 単発アラームとリピート アラームを選択可能

19.4.1 アラームの設定

アラーム機能は ALRMEN ビットで有効化します。このビットはアラーム発生時にクリアされます。 ALRMVAL への書き込みは、ALRMEN = 0 の時に実行する必要があります。

アラームのインターバルは、図 19-2 に示すように、AMASK ビット (ALCFGRPT<13:10>) で設定します。これらのビットにより、アラーム値のどの桁(複数指定可)がクロック値に一致した時にアラームを発生させるのかを指定します。

アラームを事前に決められたインターバルでリピートさせる事も可能です。アラームが有効化された後のリピート回数は、ARPT<7:0> ビット (ALCFGRPT<7:0>) で指定します。ARPT ビットが 00h で、かつ CHIME ビット (ALCFGRPT<14>) がクリアされている場合、リピート機能は無効化され、アラームは1回だけ発生します。アラームは最大 255 回リピートできます (ARPT<7:0> を FFh に設定)。

アラームが1回発生するたびに、ARPT ビットの値は1つデクリメントします。この値が00hに達すると最後のアラームが生成され、その後 ALRMEN ビットが自動的にクリアされてアラームが無効化されます。

CHIME ビットを「1」にセットした場合、アラームは無限回数リピートします。この場合、ARPT ビットが00hに達してもアラームは無効化されず、CHIME ビットがセットされている限り FFh にロールオーバーして無限にカウントを続けます。

19.4.2 アラーム割り込み

毎回のアラームイベントで割り込みが生成されます。割り込み以外に、アラームの2倍の周期で動作するアラームパルス出力を利用できます。この出力はRTCCクロックに完全に同期し、他の周辺モジュールへのトリガクロックとして使用できます。

Note: アラームが有効 (ALRMEN = 1) な時に、RCFGCAL およびALCFGRPT レジスタと CHIME ビット以外のレジスタを変更した場合、不正なアラームイベントによって不正なアラーム割り込みが発生する可能性があります。このような不正アラームイベントを回避するために、タイマ値とアラーム値はアラーム無効時(ALRMEN = 0)に変更する必要があります。ALCFGRPT レジスタと CHIME ビットは、RTCSYNC = 0 の時に変更する事を推奨します。

図 19-2: アラームマスク設定

m ; s s
h h m m s s s
h h • m m • s s
h h : m m : s s
h h m m s s s

19.5 電源制御

RTCC は電源制御機能を備えています。この機能を使用すると、外部デバイスを周期的に復帰し、そのデバイスが安定してから復帰イベントをサンプリングした後に、その外部デバイスをシャットダウンできます。RTCC は、省電力モード(スリープ、ディープスリープ等)から復帰する事なく、これを完全に自動実行できます。

この機能を有効にするには、RTCC を有効化 (RTCEN = 1) し、PWCEN レジスタビットをセットし、RTCC ピンで PWC 制御信号を駆動する (RTCOE = 1、RTCSECSEL<1:0> = 11) 必要があります。

PWC 制御信号の極性は、PWCP レジスタビットで選択できます。適切な外付けスイッチに合わせてアクティブ LOW またはアクティブ HIGH を選択する事により、1 個または複数個の外部デバイスへの電源をON/OFF できます。アクティブ LOW 設定は、RTCCピンのオープンドレイン設定と組み合わせて使用する事もできます。この設定では、外付けスイッチを必要とせずに外部デバイスの GND ピンを直接駆動できます(適切な外付け VDD プルアップ デバイスを使用)。最後に、PWC を周期的に有効にするために、CHIMEビットをセットする必要があります。

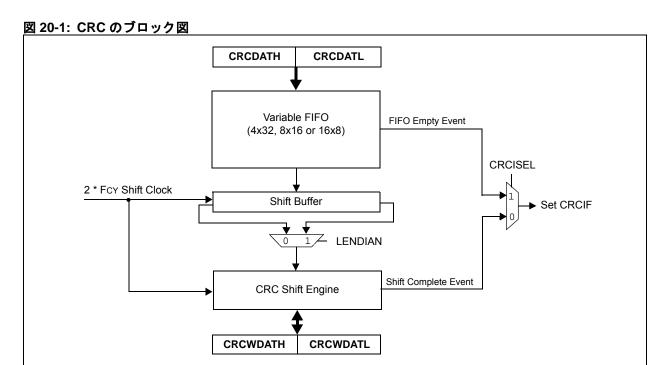
NOTE:

20.0 32 ビット プログラマブル巡回冗 長検査 (CRC)

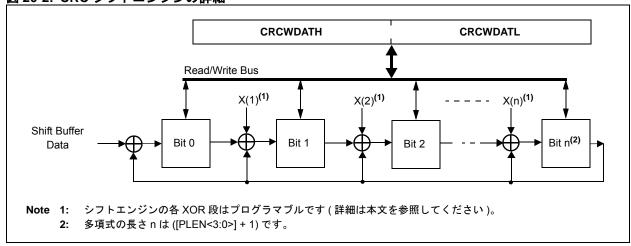
Note: 本書は、PIC24F デバイスの中の特定製品 グループを対象とし、それらの機能の概要 説明を目的としています。従って本書は、 包括的な参照資料の提供を意図したもの ではありません。詳細は「PIC24F ファミ リ リファレンス マニュアル」のセクショ ン 41.「32 ビット プログラマブル巡回冗 長検査 (CRC)」(DS39729) を参照してく ださい。 プログラマブル CRC ジェネレータは、各種のネットワーキングおよびセキュリティ アプリケーション向けにチェックサムを高速に生成するハードウェア実装手段を提供します。CRC ジェネレータは以下の特長を備えます。

- ユーザプログラマブルな最大32ビットのCRC多項式
- 選択可能なシフト方向 (リトルエンディアン/ビッグエンディアン)
- データ幅と多項式長を別々に設定可能
- ・ 設定可能な割り込み出力
- ・ データ FIFO

CRC ジェネレータの概略ブロック図を図 20-1 に示します。単純な構成の CRC シフトエンジンのブロック図を図 20-2 に示します。







20.1 ユーザ インターフェイス

20.1.1 多項式インターフェイス

CRC モジュールには、32 ビットを使用して最大で32 次の CRC 多項式をプログラミングできます。多項式の次数(最高指数)を表す多項式の長さは、PLEN<4:0>ビット(CRCCON2<4:0>)で選択します。

CRCXORL および CRCXORH レジスタは、多項式に 含める指数項を指定します。いずれかのビットをセットすると、そのビットに対応する指数項が多項式に含まれます。これにより、CRC エンジン内の対応するビットで XOR 演算が行われます。ビットをクリアすると、対応する XOR 演算は無効化されます。

16 ビットと 32 ビットの CRC 多項式の例を下に示します。

$$x16 + x12 + x5 + 1$$

x32 + x26 + x23 + x22 + x16 + x12 + x11 + x10 + x8 + x7 + x5 + x4 + x2 + x + 1

これらの多項式を CRC ジェネレータにプログラミン グするには、レジスタの各ビットを表 20-1 のように設定します。

多項式で使用する項に対応するビットを「1」にセットしている事に注目してください (例: X26 と X23)。多項式は常に bit 0 の XOR 演算を必要とします。従って X0 はドントケアです。長さ N の多項式の場合、N番目のビットは、対応するビットがセットされているかどうかに関係なく、常に使用されます。従って、CRCxOR レジスタには、長さ 32 の多項式用の 32 番目のビットは存在しません。

20.1.2 データ インターフェイス

CRC モジュールは可変データ幅の FIFO を備えます。入力データ幅は、DWIDTH<4:0> ビット (CRCCON2<12:8>) を使用して 1 \sim 32 の任意のビット数に設定できます。データ幅 (DWIDTH) が 15 より大きい場合、FIFO は 4 段です。データ幅 (DWIDTH) が 8 \sim 15 の場合、FIFO は 8 段です。データ幅 (DWIDTH) が 7 以下の場合、FIFO は 16 段です。

CRC 計算を開始する前に、データを FIFO に書き込む 必要があります。 FIFO に書き込める最小データエレメントは 1 バイトです。従って、データ幅が 7 以下であっても、 FIFO には 1 バイトを書き込む必要があります。 例えば DWIDTH の値が 5 である場合、データサイズは 6 ビット (DWIDTH + 1) です。このデータは 1 バイト (8 ビット) データとして書き込まれ、モジュールは使用しない上位 2 ビットを無視します。

データを CRCDAT レジスタの MSb (データ幅で決まる MSb) に書き込むたびに、VWORD<4:0> ビット (CRCCON1< 12:8>) の値が 1 つインクリメントします。例えば DWIDTH の値が 24 である場合、CRCDATH の bit 7 を書き込んだ時に VWORD ビットがインクリメントします。従って、常に CRCDATH より先に CRCDATL レジスタに書き込む必要があります。

CRC エンジンは、CRCGO ビットがセットされ、かつ VWORD の値がゼロより大きい時に、データのシフトを開始します。各ワードは FIFO からバッファレジスタにコピーされ、この時 VWORD がデクリメントします。次にデータがバッファからシフトアウトされます。CRC エンジンは、VWORD 値がゼロに達するまで、1 命令サイクルあたり 2 ビットをシフトし続けます。これは、各ワードの計算がデータ幅の 1/2 の命令サイクル数で完了する事を意味します。例えば、32 ビットデータの 1 ワードの CRC 計算は 16 命令サイクルで完了します。

VWORD 値が DWIDTH の設定値 (4、8、16 のいずれか)に達すると、CRCFUL ビットがセットされます。 VWORD 値がゼロに達すると、CRCMPT ビットがセットされます。 CRCEN が「0」の場合、FIFO はエンプティにされ、VWORD<4:0> ビットは「00000」にセットされます。

CRCDAT に書き込んだ後、少なくとも1命令サイクルが経過しないと、VWORD ビットの値を読み出す事はできません。

表 20-1: CRC のセットアップ例: 16 ビットと 32 ビットの多項式

CDC 生物に	ビット値					
CRC 制御ビット	16 ビット多項式	32 ビット多項式				
PLEN<4:0>	01111	11111				
X<31:16>	0000 0000 0000 000x	0000 0100 1100 0001				
X<15:0>	0001 0000 0010 000x	0001 1101 1011 011x				

20.1.3 データのシフト方向

シフト方向の選択にはLENDIAN ビット (CRCCON1<3>)を使用します。既定値では、CRC は MSb を先頭に CRC エンジンを通してデータをシフトします。LENDIAN を「1」にセットすると、CRC は LSb を先頭にデータをシフトします。この設定により、各種通信方式の組み込みがより容易となり、ソフトウェアによるビット順反転処理のオーバーヘッドを排除できます。この機能は、データを CRC エンジンヘシフトする時にだけ方向を変更するため、CRC の計算結果は反転されない事に注意してください。

20.1.4 割り込み動作

CRC モジュールは、ユーザの設定に従って下記のいずれかの条件で割り込みを生成します。CRCISEL が「0」の場合、VWORD<4:0> ビットが「1」から「0」へ変化した時に割り込みが発生します。CRCISEL が「1」の場合、CRC 計算終了後にモジュールが CRCGO ビットを「0」にクリアした時に割り込みが発生します。ユーザが手動で CRCGO ビットを「0」にクリアしても、割り込みは発生しません。

20.1.5 標準的な手順

CRC 計算の標準的な実行手順は下記の通りです。

- CRCEN ビットをセットして CRC モジュールを有効にします。
- 2. 下記のモジュール動作を設定します。
 - a) CRCXORL および CRCXORH レジスタと PLEN<4:0> ビットで多項式を設定します。
 - b) DWIDTH および LENDIAN ビットでデータ幅 とシフト方向を設定します。
 - c) CRCISEL ビットで割り込みモードを選択します。
- CRCFUL ビットがセットされるか、書き込むデータがなくなるまで、CRCDATL および CRCDATHレジスタにデータを書き込んで FIFO にデータを格納します。
- 4. CRCWDATL および CRCWDATH レジスタに 00h を書き込んで、古い結果をクリアします。中断した計算を再開する場合、CRCWDAT は変更しなくてもかまいません。
- 5. CRCGO ビットをセットして計算を開始します。
- 6. FIFO に空きができるにつれて、残りのデータを FIFO に書き込みます。
- 7. 計算が終了すると、CRCGO ビットが自動的にクリアされます。CRCISEL = 1 であれば、割り込みが発生します。
- 8. CRCWDATL および CRCWDATH レジスタから計 算結果を読み出します。

20.2 レジスタ

CRCモジュールには下記の8個のレジスタが割り当てられています。

- CRCCON1
- CRCCON2
- CRCXORL
- CRCXORH
- CRCDATL
- CRCDATH
- CRCWDATL
- CRCWDATH

CRCCON1 (レジスタ 20-1 参照) および CRCCON2 (レジスタ 20-2 参照) レジスタは、モジュール動作の制御と各種設定に使用します。CRCXOR レジスタ (レジスタ 20-3 とレジスタ 20-4 参照) は、CRC 計算に含める多項式項の選択に使用します。CRCDAT と CRCWDAT のレジスタペアは、データ入力と CRC 結果出力に使用する 2 ワードバッファとして機能します。

レジスタ 20-1: CRCCON1: CRC 制御レジスタ 1

R/W-0	U-0	R/W-0	R-0	R-0	R-0	R-0	R-0
CRCEN	_	CSIDL	VWORD4	VWORD3	VWORD2	VWORD1	VWORD0
bit 15							bit 8

R-0, HCS	R-1, HCS	R/W-0	R/W-0, HC	R/W-0	U-0	U-0	U-0
CRCFUL	CRCMPT	CRCISEL	CRCGO	LENDIAN	_	_	_
bit 7							bit 0

凡例:HC = ハードウェア クリア可能ビットHS = ハードウェア クリア / セット可能ビットR = 読み出し可能ビットW = 書き込み可能ビットU = 未実装ビット、「0」として読み出し-n = POR 時の値1 = ビットをセット0 = ビットをクリアx = ビットは未知

bit 15 CRCEN: CRC イネーブルビット

1 = CRC モジュールを有効にする

0 = CRC モジュールを無効にする

全てのステートマシン、ポインタ、CRCWDAT/CRCDAT はリセットされます。 その他の SFR はリセットされません。

bit 14 **未実装:**「0」として読み出し

bit 13 **CSIDL**: アイドルモード時 CRC 停止ビット

1 = デバイスがアイドルモードに移行した時にモジュールの動作を停止する

0 = アイドルモード時もモジュールの動作を継続する

bit 12-8 VWORD<4:0>: ポインタ値ビット

FIFO 内の有効ワード数を示します。PLEN<3:0> > 7 の時の最大値は 8、PLEN<3:0> \leq 7 の時の最大値

は16です。

bit 7 CRCFUL: FIFO フルビット

1 = FIFO はフルである

0 = FIFO はフルではない

bit 6 CRCMPT: FIFO エンプティビット

1 = FIFO はエンプティである

0 = FIFO はエンプティではない

bit 5 CRCISEL: CRC 割り込み選択ビット

1 = FIFO がエンプティ (CRC 計算未完了) の時に割り込む

0 = シフトが完了して CRCWDAT 結果の準備完了時に割り込む

bit 4 CRCGO: CRC 開始ビット

1 = CRC シリアルシフトを開始する

0 = CRC シリアルシフトを停止する

bit 3 **LENDIAN**: データシフト方向選択ビット

1 = LSb を先頭にデータワードを CRC にシフトする (リトルエンディアン)

0 = MSb を先頭にデータワードを CRC にシフトする (ビッグエンディアン)

bit 2-0 **未実装:**「0」として読み出し

レジスタ 20-2: CRCCON2: CRC 制御レジスタ 2

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
_	_	_	DWIDTH4	DWIDTH3	DWIDTH2	DWIDTH1	DWIDTH0
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
_	_	_	PLEN4	PLEN3	PLEN2	PLEN1	PLEN0
bit 7							bit 0

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「O」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-13 **未実装:**「0」として読み出し

bit 12-8 **DWIDTH<4:0>:** データ幅選択ビット

データワードの幅を指定します (データワード幅 = (DWIDTH<4:0>) + 1)。

bit 7-5 **未実装:**「0」として読み出しbit 4-0 **PI FN<4-0>:** 多項式長選択ビッ

PLEN<4:0>: 多項式長選択ビット CRC 多項式の長さを定義します (多項式の長さ = (PLEN<4:0>) + 1)。

レジスタ 20-3: CRCXORL: CRC XOR 多項式レジスタ、下位パイト

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
X15	X14	X13	X12	X11	X10	X9	X8
bit 15							bit 8

R/W-0	U-0						
X7	X6	X5	X4	Х3	X2	X1	_
bit 7							bit 0

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-1 X<15:1>: 多項式項 Xⁿ XOR イネーブルビット

bit 0 **未実装:**「0」として読み出し

レジスタ 20-4: CRCXORH: CRC XOR 多項式レジスタ、上位バイト

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
X31	X30	X29	X28	X27	X26	X25	X24
bit 15							bit 8

| R/W-0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| X23 | X22 | X21 | X20 | X19 | X18 | X17 | X16 |
| bit 7 | | | • | | | | bit 0 |

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「O」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-0 X<31:16>: 多項式項 Xⁿ XOR イネーブルビット

21.0 HIGH/LOW 電圧検出 (HLVD) 等 の全体に関わる内蔵機能

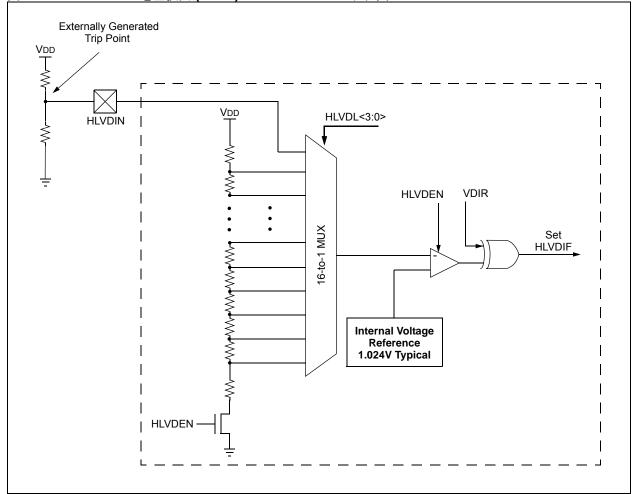
Note: 本書は、PIC24 デバイスの中の特定製品 グループを対象とし、それらの機能の概要説明を目的としています。従って本書 は、包括的な参照資料の提供を意図したものではありません。HIGH/LOW 電圧検 出の詳細は「PIC24F ファミリ リファレンス マニュアル」のセクション 36.「プログラマブル HIGH/LOW 電圧検出 (HLVD) 等の全体に関わる内蔵機能」 (DS39725) を参照してください。

HIGH/LOW 電圧検出モジュール (HLVD) は、デバイス電圧のトリップポイントと変化方向の両方をユーザが指定できるプログラマブルな回路です。

デバイス電圧が指定された方向にトリップポイントをまたいで変化すると、割り込みフラグがセットされます。割り込みが有効化されていれば、プログラム実行は割り込みベクタアドレスに分岐し、ソフトウェアは割り込みに応答する事ができます。

HLVD 制御レジスタ (レジスタ 21-1 参照)は、HLVD モジュールの全ての動作を制御します。これを使用してソフトウェア制御で回路を OFF する事により、デバイスの消費電流を最小限に抑える事ができます。

図 21-1: HIGH/LOW 電圧検出 (HLVD) モジュールのブロック図



レジスタ 21-1: HLVDCON: HIGH/LOW 電圧検出制御レジスタ

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
HLVDEN	_	HLSIDL	_	_	_	_	_
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
VDIR	BGVST	IRVST	_	HLVDL3	HLVDL2	HLVDL1	HLVDL0
bit 7							bit 0

凡例:

bit 6

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15 **HLVDEN:** HIGH/LOW 電圧検出電源イネーブルビット

1 = HLVD を有効にする 0 = HLVD を無効にする

bit 14 **未実装:**「0」として読み出し

bit 13 **HLSIDL**: アイドルモード時 HLVD 停止ビット

1 = デバイスがアイドルモードに移行した時にモジュールの動作を停止する

0 = アイドルモード時もモジュールの動作を継続する

bit 12-8 **未実装:**「0」として読み出し

bit 7 VDIR: 電圧変化方向選択ビット

1 = 電圧がトリップポイント (HLVDL<3:0>) 以上に上昇した時にイベントを生成する

0 = 電圧がトリップポイント (HLVDL<3:0>) 以下に下降した時にイベントを生成する

BGVST: バンドギャップ電圧安定フラグビット

1 = バンドギャップ電圧は安定している 0 = バンドギャップ電圧は安定していない

bit 5 IRVST: 内部参照電圧安定フラグビット

1 = 内部参照電圧は安定しており、HIGH電圧検出ロジックは指定電圧レンジで割り込みフラグをセットする

0 = 内部参照電圧は不安定であるため、電圧検出ロジックは指定電圧レンジで割り込みフラグをセッ

トしない (HLDV 割り込みを生成させない)

bit 4 **未実装:**「0」として読み出し

bit 3-0 **HLVDL<3:0>:** HIGH/LOW 電圧検出リミットビット

1111 = 外部アナログ入力を使用する (HLVDIN ピンから入力)

1110 = トリップポイント 1⁽¹⁾

1101 = トリップポイント 2(1)

1100 = トリップポイント 3⁽¹⁾

.

0000 = トリップポイント 15⁽¹⁾

Note 1: 実際のトリップポイントについては 29.0「電気的特性」を参照してください。

22.0 しきい値検出機能を備えた 12 ビット A/D コンバータ

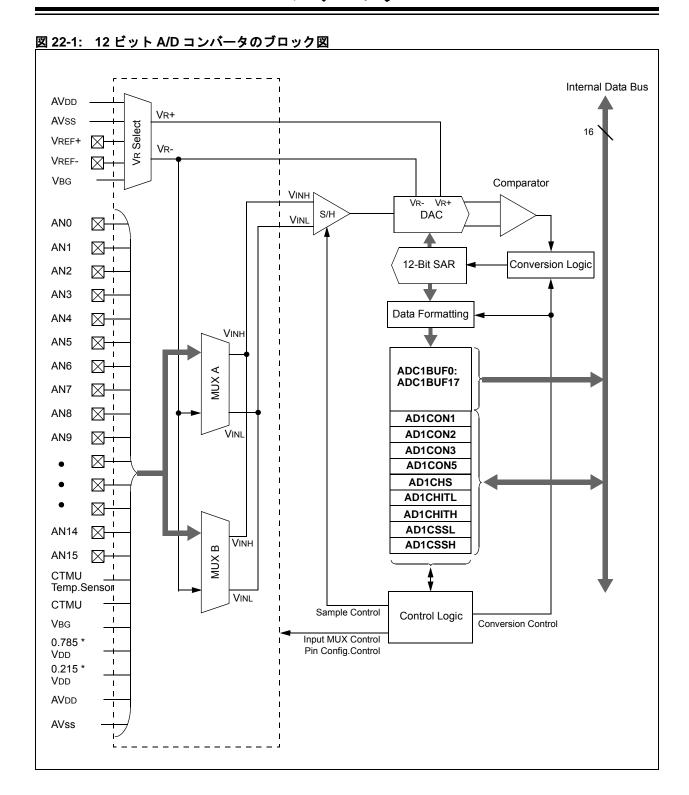
Note: 本書は、PIC24 デバイスの中の特定製品 グループを対象とし、それらの機能の概要説明を目的としています。従って本書 は、包括的な参照資料の提供を意図したものではありません。しきい値検出機能を備えた 12 ビット A/D コンバータのマニュアル」のセクション 51.「しきい値検出機能を備えた 12 ビット A/D コンバータ」(DS39739) を参照してください。

PIC24F の 12 ビット A/D コンバータは、下記の特長を備えます。

- 逐次比較型レジスタ (SAR) 変換
- 最大 100 ksps の変換速度
- 最大32チャンネルのアナログ入力(内部および外部)
- ・ 複数の内部参照入力チャンネル
- 外部電圧参照入力ピン
- ユニポーラ差動サンプル / ホールド (S/H) アンプ
- 変換結果を事前評価する自動しきい値スキャンと コンペア動作
- ・ 選択可能な変換トリガ源
- 固定長 (1 ワード / チャンネル) の設定可能な変換 結果バッファ
- ・ 変換結果整列用の 4 つのオプション
- 設定可能な割り込み生成
- CPU スリープ / アイドルモード時の動作

12 ビット A/D 変換モジュールは、一部の PIC24 が内蔵する 10 ビット モジュールの拡張バージョンです。両モジュールとも、基本的には同じ逐次比較型レジスタ (SAR) コンバータであり、柔軟な設定を可能にする豊富なハードウェア機能を備えます。本バージョンのモジュールは、12 ビット分解能、より充実した自動サンプリング オプション、他のアナログモジュール(CTMU、設定可能な結果バッファ等)とのより緊密な統合によって機能を拡張しています。本モジュールなんさいで行えます。これにより、変換結果に基づく簡単な判定をモジュール内部で行えます。

本モジュールの概略ブロック図を図22-1に示します。



A/D 変換の実行手順は下記の通りです。

- 1. A/D モジュールの設定
 - a) ポートピンをアナログ入力として設定する、 および/または、バンドギャップリファレン ス入力を選択する(ANS<12:10>、ANS<5:0>)
 - b) アナログ入力の予測電圧レンジに適合する参 照電圧源を選択する(AD1CON2<15:13>)
 - c) 必要なデータ収集レートに適合するアナログ 変換クロック (プロセッサクロックを何分周 するか)を選択する (AD1CON3<7:0>)
 - d) 適切なサンプリング/変換シーケンスを選択 する(AD1CON1<7:5>、AD1CON3<12:8>)
 - e) バッファに格納する変換結果のフォーマット を選択する (AD1CON1<9:8>)
 - f) 割り込み頻度を選択する (AD1CON2<6:2>)
 - g) A/Dモジュールを有効にする(AD1CON1<15>)
- 2. A/D 割り込みの設定(必須ではない)
 - a) AD1IF ビットをクリアする
 - b) A/D 割り込み優先度を選択する

しきい値検出スキャンを使用して A/D サンプリング / 変換を実行する手順は下記の通りです。

- 1. A/D モジュールの設定
 - a) ポートピンをアナログ入力として設定する (ANS<12:10>、ANS<5,0>)
 - b) アナログ入力の予測電圧レンジに適合する 参照電圧源を選択する (AD1CON2<15:13>)
 - c) 必要なデータ収集レートに適合するアナログ 変換クロック (プロセッサクロックを何分周 するか)を選択する (AD1CON3<7:0>)
 - d) 適切なサンプリング/変換シーケンスを選択する (AD1CON1<7:5>、AD1CON3<12:8>)
 - e) バッファに格納する変換結果のフォーマット を選択する (AD1CON1<9:8>)
 - f) 割り込み頻度を選択する (AD1CON2<6:2>)

- 2. しきい値コンペア チャンネルの設定
 - a) 自動スキャンを有効にする (ASEN ビット (AD1CON<15>))
 - b) コンペアモード (「より大きい」、「より小さい」、「ウィンドウ」) を選択する (CM ビット (AD1CON5<1:0>))
 - c) スキャンするしきい値コンペア チャンネル を選択する (ADCSSH、ADCSSL)
 - d) しきい値コンペア チャンネルの電流ソース として CTMU が必要な場合、対応する CTMU チャンネルを有効にする (ADCCTMUENH、 ADCCTMUENL)
 - e) しきい値を対応する ADC1BUFn レジスタに 書き込む
 - f) A/Dモジュールを有効にする(AD1CON1<15>)

Note: スリープモード時にしきい値検出を使用して A/D サンプリング / 変換を実行する場合、スリープモードに移行する前にRC A/D クロック源を選択する必要があります。

- 3. A/D 割り込みを設定する(必須ではない)
 - a) AD1IF ビットをクリアする
 - b) A/D 割り込み優先度を選択する

22.1 A/D 制御レジスタ

12 ビット A/D コンバータ モジュールは、動作用に最大 43 個のレジスタを使用します。全てのレジスタは データメモリ空間に配置されます。

22.1.1 制御レジスタ

本モジュールは、下記の最大 11 個の制御 / ステータス レジスタを備えます(どのレジスタを備えるかはデバ イスによって異なります)。

- AD1CON1: A/D 制御レジスタ 1
- AD1CON2: A/D 制御レジスタ 2
- AD1CON3: A/D 制御レジスタ 3
- AD1CON5: A/D 制御レジスタ 5
- AD1CHS: A/D サンプル選択レジスタ
- AD1CHITH と AD1CHITL: A/D スキャン コンペア 一致レジスタ
- AD1CSSL と AD1CSSH: A/D 入力スキャン選択レジスタ
- AD1CTMENH と AD1CTMENL: CTMUイネーブル レジスタ

AD1CON1、AD1CON2、AD1CON3 レジスタ (レジスタ 22-1、レジスタ 22-2、レジスタ 22-3 参照)は、A/D モジュールの全体的な動作を制御します。これにはモジュールの有効化、変換クロックと参照電圧源の設定、サンプリングおよび変換トリガの選択、サンプリング/変換シーケンスの手動制御が含まれます。AD1CON5 レジスタ (レジスタ 22-4 参照)は、しきい値検出動作の機能(省電力モード時の機能も含む)を制御します。

AD1CHS レジスタ (レジスタ 22-5 参照)は、S/H アンプに接続する入力チャンネルを選択します。また、このレジスタを使用して、入力マルチプレクサの選択と、差動サンプリング用の参照源の選択も行えます。

AD1CHITHおよびAD1CHITLレジスタ(レジスタ22-6、レジスタ22-7参照)は、しきい値検出動作で使用するセマフォレジスタです。個々のビット(場合によってはビットペア)のステータスは、一致条件が発生した

かどうかを示します。全てのデバイスは AD1CHITL を 実装していますが、16 チャンネル以下のデバイスには AD1CHITH を実装していないものも存在します。

AD1CSSH/L レジスタ (レジスタ 22-8 とレジスタ 22-9 参照)は、シーケンシャル スキャンに含めるチャンネルを選択します。

AD1CTMENH/L レジスタ (レジスタ 22-10 とレジスタ 22-11 参照) は、変換中に CTMU が使用するチャンネルを選択します。いずれかのチャンネルを選択する事により、A/D コンバータはそのチャンネルを介して CTMU(特にその電流ソース)を制御し、そのデータを読み出す事ができます。全てのデバイスは AD1CTMENL を実装していますが、16 チャンネル以下のデバイスには AD1CTMENH を実装していないものも存在します。

22.1.2 A/D 結果バッファ

本モジュールは、ADC1BUF と呼ばれるマルチワードのデュアルポート RAM を実装しています。このバッファは、少なくともそのデバイスの外部アナログチャンネル数と同数 (最大 32)のワードアドレスで構成されます。バッファアドレス数は常に偶数です。各アドレスはデータメモリ空間に配置され、個々にアドレッシング可能です。バッファアドレスは、ADC1BUF0~ADC1BUFn (n は最大 31)として参照されます。

A/D 結果バッファは読み書き可能です。モジュールがアクティブ (AD1CON<15>=1) の時、バッファは読み出し専用となり、A/D 変換結果を保存します。モジュールが非アクティブ (AD1CON<15>=0) の時、バッファは読み書き可能です。この場合、バッファアドレスへの書き込みによって、しきい値検出動作の変換しきい値が設定されます。

ADON ビット (AD1CON1<15>) でモジュールを無効化しても、バッファの内容はクリアされません。変換結果と設定されたしきい値は、ADON ビットをセットまたはクリアしても保持されます。

レジスタ 22-1: AD1CON1: A/D 制御レジスタ 1

R/W-0	U-0	R/W-0	U-0	U-0	r-0	R/W-0	R/W-0
ADON	_	ADSIDL	_	_	_	FORM1	FORM0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0 HSC	R/C-0 HSC
SSRC3	SSRC2	SSRC1	SSRC0	_	ASAM	SAMP	DONE
bit 7							bit 0

凡例: U = 未実装ビット、「0」として読み出し

C=クリア可能ビット r=予約済みビット

R=読み出し可能ビット W=書き込み可能ビット HSC=ハードウェア セット / クリア可能ビット -n = POR 時の値 1= ビットをセット 0= ビットをクリア x = ビットは未知

bit 15 ADON: A/D 動作モードビット

1 = A/D コンバータ モジュールは動作中

0 = A/D コンバータは停止中

bit 14 **未実装:**「0」として読み出し

bit 13 ADSIDL: アイドルモード時停止ビット

1 = デバイスがアイドルモードに移行した時にモジュール動作を中止する

0 = アイドルモード中もモジュール動作を継続する

bit 12-11 **未実装:**「0」として読み出し

bit 10 **予約済み:**「1」として読み出し

bit 9-8 **FORM<1:0>:** データ出力フォーマット ビット

11 = 符号付き小数、左揃え

10 = 符合なし小数、左揃え

01 = 符号付き整数、右揃え00 = 符号なし整数、右揃え

bit 7-4 SSRC<3:0>: サンプルクロック源選択ビット

1111 =利用不可(使用禁止)

•

1000 =利用不可(使用禁止)

0111 =内部カウンタでサンプリング終了/変換開始をトリガする(自動変換)

0110 =利用不可(使用禁止)

0101 =Timer1 イベントでサンプリング終了 / 変換開始をトリガする

0100 =CTMU イベントでサンプリング終了 / 変換開始をトリガする

0011 =Timer5 イベントでサンプリング終了 / 変換開始をトリガする

0010 =Timer3 イベントでサンプリング終了 / 変換開始をトリガする

0001 =INTO イベントでサンプリング終了 / 変換開始をトリガする 0000 =ソフトウェアによる SMAP ビットのクリアでサンプリング終了 / 変換開始をトリガする

bit 3 **未実装:**「0」として読み出し

bit 2 ASAM: A/D サンプル自動開始ビット

1 = 直前の変換終了後即座にサンプリングを開始する; SAMP ビットは自動的にセットされる

0 = SAMP ビットの手動セットによってサンプリングを開始する

bit 1 **SAMP:** A/D サンプル イネーブルビット

1 = A/D サンプル / ホールド アンプをサンプリング状態にする

0 = A/D サンプル / ホールド アンプをホールド状態にする

bit 0 **DONE**: A/D 変換ステータスビット

1 = A/D 変換サイクルは完了した

0 = A/D 変換サイクルは開始していない、または実行中

レジスタ 22-2: AD1CON2: A/D 制御レジスタ 2

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
PVCFG1	PVCFG0	NVCFG0	OFFCAL	BUFREGEN	CSCNA	_	_
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BUFS ⁽¹⁾	SMPI4	SMPI3	SMPI2	SMPI1	SMPI0	BUFM ⁽¹⁾	ALTS
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-14 **PVCFG<1:0>:** コンバータ正極性参照電圧コンフィグレーション ビット

11 = 内部 VRH2

10 = 内部 VRH1

01 = 外部 VREF+

00 = AVDD

bit 13 NVCFG0: コンバータ負極性参照電圧コンフィグレーション ビット

1 = 外部 VREF-

0 = AVss

bit 12 OFFCAL: オフセット校正モード選択ビット

1 = チャンネル サンプル / ホールド アンプの反転および非反転入力を AVss に接続する

0 = チャンネル サンプル / ホールド アンプの反転および非反転入力を通常の入力に接続する

bit 11 BUFREGEN: A/D バッファレジスタ イネーブルビット

1 = 変換結果を変換チャンネルによって決められたバッファ位置に書き込む

0 = A/D 結果バッファを FIFO として扱う

bit 10 **CSCNA**: CH0+ サンプル A 時スキャン入力選択ビット

1= 入力をスキャンする

0= 入力をスキャンしない

bit 9-8 **未実装:**「0」として読み出し

bit 7 **BUFS**: バッファ書き込みステータスビット ⁽¹⁾

1 = A/D はバッファの上位半分に書き込んでいる; ユーザはバッファの下位半分のデータにアクセス する必要がある

0 = A/Dはバッファの下位半分に書き込んでいる;ユーザはバッファの上位半分のデータにアクセス

する必要がある

bit 6-2 **SMPI<4:0>**: 割り込み頻度選択ビット

11111 =32 個目のサンプルの変換が完了するたびに割り込む

11110 =31 個目のサンプルの変換が完了するたびに割り込む

•

00001 =2 個目のサンプルの変換が完了するたびに割り込む

00000 =各サンプルの変換が完了するたびに割り込む

bit 1 **BUFM**: バッファ書き込みモード選択ビット ⁽¹⁾

1 = 最初の割り込み発生時に AD1BUF0 でバッファの書き込みを開始し、次の割り込み発生時に AD1BUF(n/2)で書き込みを開始する(分割バッファモード)

0 = アドレス ADCBUF0でバッファの書き込みを開始し、後続の割り込み発生時に次のアドレスに順次書き込む (FIFO モード)

bit 0 ALTS: 交互入力サンプルモード選択ビット

1 = 最初のサンプリングでサンプルA用チャンネル入力選択を使用し、次のサンプリングでサンプル B用チャンネル入力選択を使用する

0 = 常にサンプルA用チャンネル入力選択を使用する

Note 1: バッファを FIFO モードで使用する場合 (BUFREGEN = 0) にのみ、適用可能です。また、BUFS は BUFM = 1 の時にのみ使用されます。

レジスタ 22-3: AD1CON3: A/D 制御レジスタ 3

R/W-0	R-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADRC	EXTSAM	_	SAMC4	SAMC3	SAMC2	SAMC1	SAMC0
bit 15							bit 8

| R/W-0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| ADCS7 | ADCS6 | ADCS5 | ADCS4 | ADCS3 | ADCS2 | ADCS1 | ADCS0 |
| bit 7 | | | | | | | bit 0 |

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15 ADRC: A/D 変換クロック源ビット

1 = RC クロック

0= システムクロックから分周したクロック

bit 14 EXTSAM: サンプリング時間延長ビット

1 = SAMP = 0 後も A/D はサンプリングし続けている

0 = A/D はサンプリングを終了した

bit 13 **予約済み:**「0」として読み出し

bit 12-8 SAMC<4:0>: 自動サンプリング時間選択ビット

11111 **=31** TAD

:

00001 =1 TAD

00000 **=0** TAD

bit 7-0 **ADCS<7:0>:** A/D 変換クロック選択ビット

11111111-01000000 = 予約済み

00111111 = 64 · TCY = TAD

•

00000001 = 2 · Tcy = TaD 00000000 = Tcy = TaD

レジスタ 22-4: AD1CON5: A/D 制御レジスタ 5

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
ASEN	LPEN	CTMREQ	BGREQ	VRSREQ	_	ASINT1	ASINT0
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
_	_	_	_	WM1	WM0	CM1	СМ0
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15 ASEN: 自動スキャン イネーブルビット

1= 自動スキャンを有効にする

0 = 自動スキャンを無効にする

bit 14 LPEN: 低消費電力イネーブルビット

1= スキャン後に低消費電力モードに戻る

0 = スキャン後にフルパワーモードに戻る

bit 13 CTMREQ: CTMU 要求ビット

1 = ADC が有効化されてアクティブな時に CTMU を有効にする

0 = ADC は CTMU を有効にしない

bit 12 BGREQ: バンドギャップ要求ビット

1 = ADC が有効化されてアクティブな時にバンドギャップを有効にする

0 = ADC はバンドギャップを有効にしない

bit 11 VRSREQ: VREG スキャン要求ビット

1 = ADC が有効化されてアクティブな時に内蔵レギュレータを有効にする

0 = ADC は内蔵レギュレータを有効にしない

bit 10 **未実装:**「0」として読み出し

bit 9-8 ASINT<1:0>: 自動スキャン(しきい値検出)割り込みモードビット

11 = しきい値検出シーケンスが完了して有効なコンペアが発生した後に割り込む

10 = 有効なコンペアが発生した後に割り込む

01 = しきい値検出シーケンスが完了した後に割り込む

00 = 割り込みなし

bit 7-4 **未実装:**「0」として読み出し

bit 3-2 WM<1:0>: 書き込みモードビット

11 = 予約済み

10 = 自動コンペアのみ(変換結果を保存しないが、CMおよびASINTビットの定義に従って、有効な一 致が発生した時に割り込みを生成する)

01 = 変換と保存(CMビットで定義された一致が発生した時に、レジスタビットによって決められた位置に変換結果を保存する)

00 = レガシー動作(バッファレジスタ ビットによって決められた位置に変換データを保存する)

bit 1-0 **CM<1:0>:** コンペアモード ビット

11 = ウィンドウ外側モード(変換結果が対応するバッファペアによって定義されたウィンドウの外側にある場合に有効な一致が発生する)

10 = ウィンドウ内側モード(変換結果が対応するバッファペアによって定義されたウィンドウの内側にある場合に有効な一致が発生する)

01 = より大きいモード(変換結果が対応するバッファレジスタの値よりも大きい場合に有効な一致が 発生する)

00 = より小さいモード(変換結果が対応するバッファレジスタの値よりも小さい場合に有効な一致が発生する)

レジスタ 22-5: AD1CHS: A/D サンプル選択レジスタ

| R/W-0 |
|--------|--------|--------|--------|--------|--------|--------|--------|
| CH0NB2 | CH0NB1 | CH0NB0 | CH0SB4 | CH0SB3 | CH0SB2 | CH0SB1 | CH0SB0 |
| bit 15 | | | | | | | bit 8 |

| R/W-0 |
|--------|--------|--------|--------|--------|--------|--------|--------|
| CH0NA2 | CH0NA1 | CH0NA0 | CH0SA4 | CH0SA3 | CH0SA2 | CH0SA1 | CH0SA0 |
| bit 7 | | | | | | | bit 0 |

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「O」として読み出し

0 = ビットをクリア x=ビットは未知 -n = POR 時の値 1=ビットをセット

bit 15-13 CH0NB<2:0>: サンプル B チャンネル 0 負極性入力選択ビット

 $111 = AN6^{(1)}$

 $110 = AN5^{(2)}$

101 = AN4

100 = AN3

011 = AN2

010 = AN1001 = AN0

000 = AVss

bit 12-8 **CH0SB<4:0>:** MUX B マルチプレクサ設定用 S/H アンプ正極性入力選択ビット

11111 = 未実装 (使用禁止) 11101 = AVDD(3)

11101 = AVss⁽³⁾

11100 = 上側ガードバンド レール (0.785 * VDD)

11011 = 下側ガードバンド レール (0.215 * VDD)

11010 = 内部バンドギャップ リファレンス (VBG)(3)

11001-10010 = 未実装(使用禁止)

10001 = チャンネルを接続しない、全ての入力はフローティング (CTMU 用に使用)

10000 = チャンネルを接続しない、全ての入力はフローティング (CTMU 温度センサ入力用に使用)

01111 = AN15

01110 = AN14

01101 = AN13

01100 = AN12

01011 **= AN11**

01010 = AN10

01001 = AN9

01000 = AN8⁽¹⁾

00111 = AN7⁽¹⁾

00110 = AN6⁽¹⁾

 $00101 = AN5^{(2)}$

00100 = AN4

00011 = AN3

00010 = AN2

00001 = AN1

00000 = AN0

bit 7-5 CH0NA<2:0>: サンプル A チャンネル 0 負極性入力選択ビット

定義は CHONB<2:0> と同じです。

bit 4-0 CH0SA<4:0>: サンプル A チャンネル 0 正極性入力選択ビット

定義は CHONA<4:0> と同じです。

Note 1: 44 ピンデバイスだけがこのビットを実装します。

2: 28/44 ピンデバイスだけがこのビットを実装します。

3: この入力に使用する実際のバンドギャップ値は、PVCFG ビット (AD1CON2<15:14>) で選択します。

レジスタ 22-6: AD1CHITH: A/D スキャン コンペアー致レジスタ (上位ワード)⁽¹⁾

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
_	_	_	_	_	_	_	_
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
_	_	_	_	_	_	CHH17	CHH16
bit 7							bit 0

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-2 **未実装:**「0」として読み出し

bit 1-0 CHH<17:16>: A/D コンペアー致ビット

CM<1:0> = 11 の場合:

1 = A/D 結果バッファ x にデータが書き込まれた、または、一致が発生した

0 = A/D 結果バッファ x にデータは書き込まれていない

CM<1:0> が上記以外の値の場合:

1 = A/D 結果チャンネル x で一致が発生した

0 = A/D 結果チャンネル x で一致は発生していない

Note 1: 未実装チャンネルは「0」として読み出されます。

レジスタ 22-7: AD1CHITL: A/D スキャン コンペアー致レジスタ (下位ワード)⁽¹⁾

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CHH15	CHH14	CHH13	CHH12	CHH11	CHH10	CHH9	CHH8
bit 15							bit 8

| R/W-0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| CHH7 | CHH6 | CHH5 | CHH4 | CHH3 | CHH2 | CHH1 | CHH0 |
| bit 7 | | | | | | | bit 0 |

凡.例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-0 **CHH<15:0>:** A/D コンペアー致ビット

CM<1:0> = 11 の場合:

1 = A/D 結果バッファ×にデータが書き込まれた、または、一致が発生した

0 = A/D 結果バッファ x にデータは書き込まれていない

<u>CM<1:0> が上記以外の値の場合:</u>

1 = A/D 結果チャンネル n で一致が発生した

0 = A/D 結果チャンネル n で一致は発生していない

Note 1: 未実装チャンネルは「0」として読み出されます。

レジスタ 22-8: AD1CSSH: A/D 入力スキャン選択レジスタ (上位ワード)⁽¹⁾

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
_	CSS30	CSS29	CSS28	CSS27	CSS26	_	_
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
_	_	_	_	_	_	CSS17	CSS16
bit 7							bit 0

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15 **未実装:**「0」として読み出し

bit 14-10 CSS<30:26>: A/D 入力スキャン選択ビット

1 = 対応するチャンネルを入力スキャンに含める

0 = 対応するチャンネルを入力スキャンから除外する

bit 9-2 **未実装:**「0」として読み出し

bit 1-0 CSS<17:16>: A/D 入力スキャン選択ビット

1 = 対応するチャンネルを入力スキャンに含める 0 = 対応するチャンネルを入力スキャンから除外する

Note 1: 未実装チャンネルは「0」として読み出されます。未実装チャンネルをサンプリング用に選択しないでください。選択すると不確定な結果が生成される可能性があります。

レジスタ 22-9: AD1CSSL: A/D 入力スキャン選択レジスタ (下位ワード)⁽¹⁾

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CSS15	CSS14	CSS13	CSS12	CSS11	CSS10	CSS9	CSS8
bit 15							bit 8

| R/W-0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| CSS7 | CSS6 | CSS5 | CSS4 | CSS3 | CSS2 | CSS1 | CSS0 |
| bit 7 | | | | | | | bit 0 |

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-0 **CSS<15:0>:** A/D 入力スキャン選択ビット

1 = 対応する ANx 入力をスキャンに含める

0= チャンネルを入力スキャンから除外する

Note 1: 未実装チャンネルは「0」として読み出されます。未実装チャンネルをサンプリング用に選択しないでください。選択すると不確定な結果が生成される可能性があります。

レジスタ 22-10: AD1CTMENH: CTMU イネーブル レジスタ (上位ワード)⁽¹⁾

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
_	_	_	_	_	_	_	_
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
_	_	_	_	_	-	CTMEN17	CTMEN16
bit 7							bit 0

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-2 **未実装:**「0」として読み出し

bit 1-0 **CTMEN<17:16>:** 変換中 CTMU イネーブルビット

1 = 変換中に CTMU を有効にして選択したチャンネルに接続する

0 = CTMU をこのチャンネルに接続しない

Note 1: 未実装チャンネルは「0」として読み出されます。

レジスタ 22-11: AD1CTMENL: CTMU イネーブル レジスタ (下位ワード)⁽¹⁾

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CTMEN15	CTMEN14	CTMEN13	CTMEN12	CTMUEN11	CTMEN10	CTMEN9	CTMEN8
bit 15							bit 8

| R/W-0 |
|--------|--------|--------|--------|--------|--------|--------|--------|
| CTMEN7 | CTMEN6 | CTMEN5 | CTMEN4 | CTMEN3 | CTMEN2 | CTMEN1 | CTMEN0 |
| bit 7 | | | | | | | bit 0 |

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「O」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-0 **CTMEN<15:0>:** 変換中 CTMU イネーブルビット

1 = 変換中に CTMU を有効にして選択したチャンネルに接続する

0 = CTMU をこのチャンネルに接続しない

Note 1: 未実装チャンネルは「0」として読み出されます。

22.2 A/D サンプリング要件

12 ビット A/D コンバータのアナログ入力モデルを 図 22-2 に示します。A/D の総サンプリング時間は、 ホールド コンデンサの充電時間によって決まりま す。

A/D コンバータの仕様精度を達成するには、ホールドコンデンサ(CHOLD)をアナログ入力ピンの電圧レベルまで完全に充電できる必要があります。ソースインピーダンス(RIC)、内国サンプリングスイッチのインピーダンス(RSS)による複合インピーダンスは、CHOLDの充電に要する時間内にホールドコンデンサを完全に充電するために、アナログソースの複合インピーダンスを十分に小さくする必要があります。ピンリーク電流によるA/DコスインピーダンスRSを2.5kΩ以下に抑える事を推奨します。アナログ入力チャンネルを選択(変更)した後、

変換を開始する前に、サンプリング動作が完了する必要があります。内部ホールド コンデンサは、毎回のサンプリング動作の前に放電された状態となります。

変換から次の変換までの間に、1 TAD 以上の時間周期を確保する必要があります。詳細は 29.0「電気的特性」を参照してください。

式 22-1: A/D 変換クロック周期

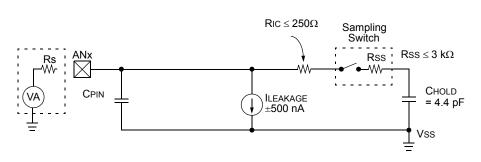
$$Tad = Tcy(ADCS + 1)$$

$$ADCS = \frac{TAD}{TCY} - 1$$

Note: Tcy = 2/Fosc に基づく; Doze モードと

PLL は無効

図 22-2: 12 ビット A/D コンバータのアナログ入力モデル



凡例: CPIN = 入力静電容量

VT = しきい値電圧

ILEAKAGE = 各種接合によるピン部位のリーク電流

RIC = 配線抵抗

Rss = サンプリング スイッチ抵抗

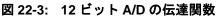
CHOLD = サンプル / ホールド静電容量 (DAC から)

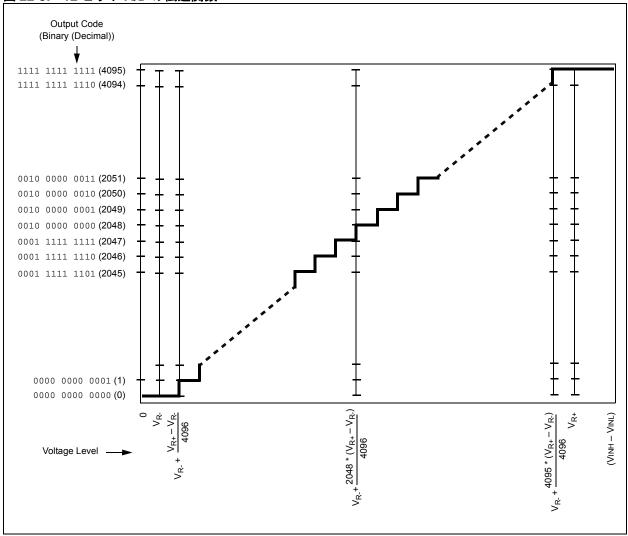
Note: CPIN の値はデバイスのパッケージによって異なり、検査されていません。Rs \leq 5 k Ω の場合、CPIN の影響は無視できます。

22.3 伝達関数

12 ビット分解能 A/D コンバータの伝達関数を図 22-3 に示します。入力電圧差(VINH – VINL)は参照電圧((VR+) – (VR-)) と比較されます。

- 最初のコード遷移は、 入力電圧が ((VR+) – (VR-))/4096 または 1.0 LSb の時に発生します。
- コード「0000 0000 0001」の中心電圧は VR-+ (1.5 * ((VR+) - (VR-))/4096)です。
- コード「0010 0000 0000」の中心電圧は VREFL + (2048.5 * ((VR+) – (VR-))/4096)です。
- VR-+(((VR-)-(VR-))/4096)より低い入力電圧は「0000 0000 0000」として変換されます。
- (VR-) + (1023((VR+) (VR-))/4096) より高い入力電圧 は、「1111 1111 1111」として変換されます。





23.0 コンパレータ モジュール

Note: 本

本書は、PIC24 デバイスの中の特定製品 グループを対象とし、それらの機能の概要説明を目的としています。従って本書は、包括的な参照資料の提供を意図したものではありません。コンパレータ モジュールの詳細は「PIC24F ファミリ リファレンス マニュアル」のセクション 46.「スケーラブル コンパレータ モジュール」(DS39734) を参照してください。

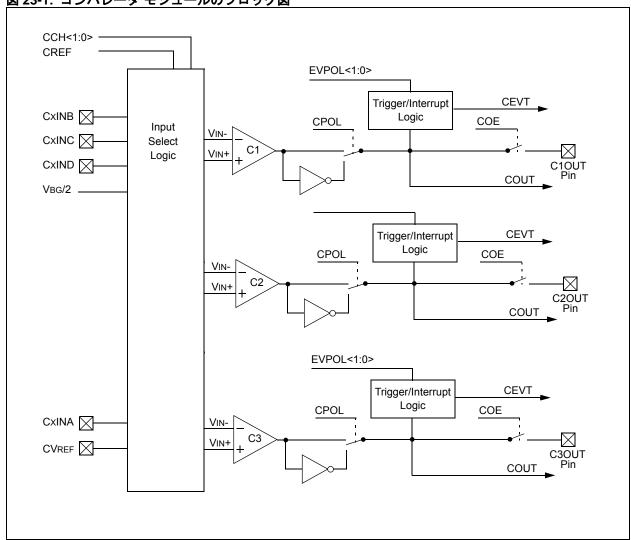
コンパレータ モジュールは、3個の2入力コンパレータを提供します。コンパレータへの入力には、4つの外部アナログ入力のいずれか1つと、内部バンドギャップ リファレンスの1/2(VBG/2)またはコンパレータ参照電圧ジェネレータからの参照電圧入力を使用できます。

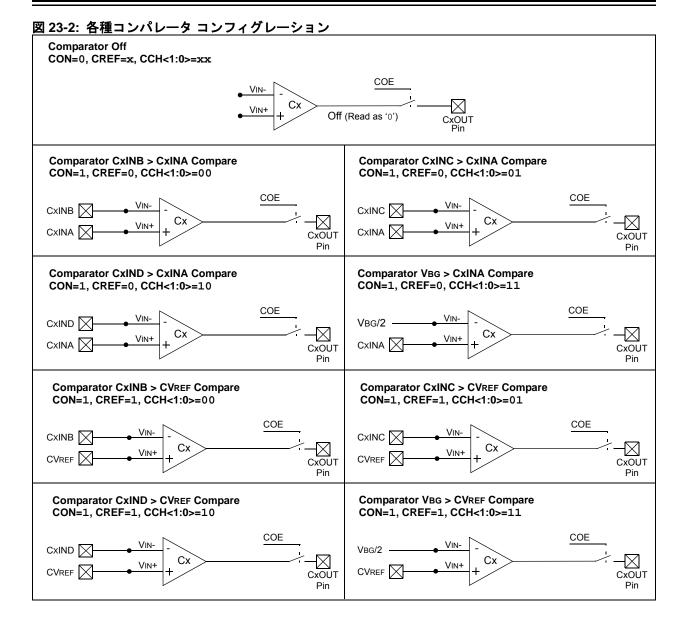
コンパレータ出力はCxOUTピンに直接接続できます。 COE ビットが「1」の時、I/O ピンロジックは対応するピンにコンパレータの非同期出力を供給します。

図 23-1 にコンパレータ モジュールの概略ブロック図を示します。使用可能な各種コンパレータ コンフィグレーションを図 23-2 に示します。

各コンパレータは、そのコンパレータ専用の制御レジスタ CMxCON (レジスタ 23-1 参照)を持ち、別々に動作を有効化して設定できます。3 個のコンパレータの出力とイベントステータスは、全て1つの CMSTATレジスタ(レジスタ 23-2 参照)に格納されます。

図 23-1: コンパレータ モジュールのブロック図





レジスタ 23-1: CMxCON: コンパレータ x 制御レジスタ

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R-0
CON	COE	CPOL	CLPWR	_	_	CEVT	COUT
bit 15							bit 8

R/W-0	R/W-0	U-0	R/W-0	U-0	U-0	R/W-0	R/W-0
EVPOL1	EVPOL0	_	CREF	_	_	CCH1	CCH0
bit 7							bit 0

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15 **CON:** コンパレータ イネーブルビット

1 = コンパレータを有効にする

0 = コンパレータを無効にする

bit 14 **COE**: コンパレータ出力イネーブルビット

1 = コンパレータ出力を CxOUT ピンで出力する

0 = コンパレータ出力を内部でのみ使用する

bit 13 CPOL: コンパレータ出力極性選択ビット

1 = コンパレータ出力を反転する

0 = コンパレータ出力を反転しない

bit 12 CLPWR: コンパレータ低消費電力モード選択ビット

1=コンパレータを低消費電力モードで動作させる

0=コンパレータを低消費電力モードで動作させない

bit 11-10 **未実装:**「0」として読み出し

bit 9 CEVT: コンパレータ イベントビット

1 = EVPOL<1:0> で定義されたコンパレータ イベントが発生した; このビットがクリアされるまで、

後続のトリガと割り込みを無効化する

0 = コンパレータ イベントは発生していない

bit 8 COUT: コンパレータ出力ビット

<u>CPOL = 0 の場合:</u>

1 = VIN+ > VIN-

0 = VIN+ < VIN-

<u>CPOL = 1 の場合:</u>

1 = VIN+ < VIN-

0 = VIN+ > VIN-

bit 7-6 **EVPOL<1:0>:** トリガ / イベント / 割り込み極性選択ビット

11 = コンパレータ出力が変化するたびにトリガ / イベント / 割り込みを生成する (CEVT = 0 の時)

10 = コンパレータ出力で下記の状態遷移が発生した時にトリガ / イベント / 割り込みを生成する:

<u>CPOL = 0 (極性反転しない)の場合:</u>

HIGH から LOW への遷移時のみ

<u>CPOL = 1 (極性反転する)の場合:</u>

LOW から HIGH への遷移時のみ

01 = コンパレータ出力で下記の状態遷移が発生した時にトリガ / イベント / 割り込みを生成する:

<u>CPOL = 0 (極性反転しない)の場合:</u>

LOW から HIGH への遷移時のみ

<u>CPOL = 1 (極性反転する)の場合:</u>

HIGH から LOW への遷移時のみ

00 = トリガ / イベント / 割り込みの生成を無効にする

bit 5 **未実装:**「0」として読み出し

レジスタ 23-1: CMxCON: コンパレータ x 制御レジスタ (続き)

bit 4 CREF: コンパレータ参照電圧選択ビット(非反転入力)

1 = 非反転入力を内部 CVREF 電圧に接続する 0 = 非反転入力を CxINA ピンに接続する

bit 3-2 **未実装:**「0」として読み出し

bit 1-0 **CCH<1:0>:** コンパレータ チャンネル選択ビット

11 = コンパレータの反転入力を VBG/2 に接続する

10 = コンパレータの反転入力を CxIND ピンに接続する 01 = コンパレータの反転入力を CxINC ピンに接続する 00 = コンパレータの反転入力を CxINB ピンに接続する

レジスタ 23-2: CMSTAT: コンパレータ モジュール ステータス レジスタ

R/W-0	U-0	U-0	U-0	U-0	R-0, HSC	R-0, HSC	R-0, HSC
CMIDL	_	_	_	_	C3EVT	C2EVT	C1EVT
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R-0, HSC	R-0, HSC	R-0, HSC
_	_	_	_	_	C3OUT	C2OUT	C1OUT
bit 7							bit 0

凡例: HSC = N - F + D = N

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15 CMIDL: アイドルモード時コンパレータ停止ビット

1 = デバイスがアイドルモードに移行した時に全てのコンパレータの動作を停止する

0 = アイドルモード中も有効化された全てのコンパレータの動作を継続する

bit 14-11 **未実装:**「0」として読み出し

bit 10 C3EVT: コンパレータ 3 イベント ステータスビット (読み出し専用)

このステータスビットは、コンパレータ 3 の現在のイベント ステータス (CM3CON<9>) を示します。

bit 9 **C2EVT:** コンパレータ 2 イベント ステータスビット (読み出し専用)

このステータスビットは、コンパレータ 2 の現在のイベント ステータス (CM2CON<9>) を示します。

bit 8 **C1EVT:** コンパレータ 1 イベント ステータスビット (読み出し専用)

このステータスビットは、コンパレータ 1 の現在のイベント ステータス (CM1CON<9>) を示します。

bit 7-3 **未実装:**「0」として読み出し

bit 2 C3OUT: コンパレータ 3 出力ステータスビット (読み出し専用)

このステータスビットは、コンパレータ 3 の現在の出力 (CM3CON<8>) を示します。

bit 1 **C2OUT**: コンパレータ 2 出力ステータスビット (読み出し専用)

このステータスビットは、コンパレータ 2 の現在の出力 (CM2CON<8>) を示します。

bit 0 **C1OUT:** コンパレータ 1 出力ステータスビット (読み出し専用)

このステータスビットは、コンパレータ 1 の現在の出力 (CM1CON<8>) を示します。

24.0 コンパレータ参照電圧

Note:

本書は、PIC24 デバイスの中の特定製品グループを対象とし、それらの機能の概要説明を目的としています。従って本書は、包括的な参照資料の提供を意図したものではありません。コンパレータ参照電圧の詳細は「PIC24Fファミリリファレンスマニュアル」のセクション 20.「コンパレータ参照電圧モジュール」(DS39709)を参照してください。

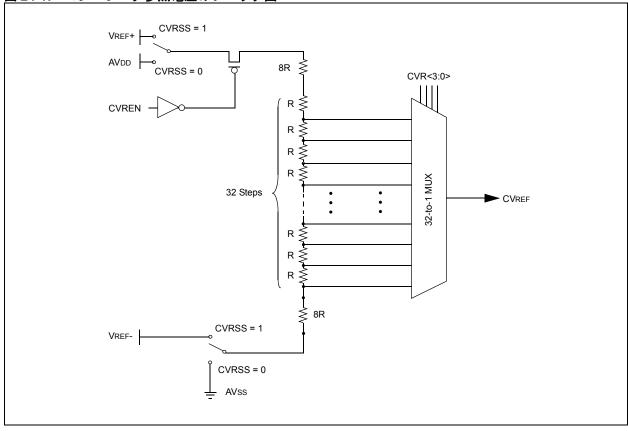
24.1 コンパレータ参照電圧のコンフィ グレーション

コンパレータ参照電圧モジュールは、CVRCON レジスタ (レジスタ 24-1 参照)によって制御されます。コンパレータ参照電圧は、幅広い電圧レンジで 32 段階の出力電圧を供給します。

コンパレータ参照電圧には、VDD / VSS または外部 VREF+/VREF-から電圧を供給できます。この電圧源は、 CVRSS ビット (CVRCON<5>) で選択します。

CVREF 出力を変更する際は、コンパレータ参照電圧のセトリングタイムを考慮する必要があります。

図 24-1: コンパレータ参照電圧のブロック図



レジスタ 24-1: CVRCON: コンパレータ参照電圧制御レジスタ

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
_	_	_	_	_	_	_	_
bit 15							bit 8

| R/W-0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| CVREN | CVROE | CVRSS | CVR4 | CVR3 | CVR2 | CVR1 | CVR0 |
| bit 7 | | | | | | | bit 0 |

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-8 **未実装:**「0」として読み出し

bit 7 CVREN: コンパレータ参照電圧イネーブルビット

1 = CVREF 回路の電源を ON にする 0 = CVREF 回路の電源を OFF にする

bit 6 CVROE: コンパレータ VREF 出力イネーブルビット

1 = CVREF 電圧レベルを CVREF ピンで出力する 0 = CVREF 電圧レベルを CVREF ピンから切断する

bit 5 CVRSS: コンパレータ VREF 源選択ビット

1 = コンパレータの参照電圧源: CVRSRC = VREF+ - VREF-0 = コンパレータの参照電圧源: CVRSRC = AVDD - AVSS

bit 4-0 CVR<4:0>: コンパレータ VREF 値選択 0 ≤ CVR<4:0> ≤ 31 ビット

<u>CVRSS = 1 の時:</u>

CVREF = (VREF-) + (CVR<4:0>/32) • (VREF+ - VREF-)

<u>CVRSS = 0 の時:</u>

 $\overline{\text{CVREF}} = (\overline{\text{AVSS}}) + (\overline{\text{CVR}} < 4:0 > /32) \cdot (\overline{\text{AVDD}} - \overline{\text{AVSS}})$

25.0 充電時間計測ユニット (CTMU)

Note:

本書は、PIC24 デバイスの中の特定製品グループを対象とし、それらの機能の概要説明を目的としています。従って本書は、包括的な参照資料の提供を意図したものではありません。充電時間計測ユニットの詳細は PIC24F ファミリ リファレンス マニュアルのセクション53.「しきい値検出機能を備えた充電時間計測ユニット (CTMU)」(DS39743) を参照してください。

充電時間計測ユニット (CTMU) は、充電時間の計測と、パルス源と非同期パルス生成間の正確な時間差計測を可能にする柔軟なアナログ モジュールです。本モジュールの主な特長は以下の通りです。

- ・ 13 個の外部エッジ入力ソース
- 各エッジソースの極性の制御
- エッジシーケンスの制御
- エッジレベルまたはエッジ遷移に対する応答の制御
- 1 ns の時間計測分解能
- 静電容量計測に適した精密な電流ソース

CTMU を他の内蔵アナログ モジュールと組み合わせ て使用する事により、時間、静電容量、静電容量の相対的変化を高い精度で計測できます。さらに、システムクロックとは同期しない出力パルスを生成する事もできます。CTMU モジュールは、静電容量式タッチセンサとのインターフェイ用に最適です。

CTMU モジュールは、3 個のレジスタ (CTMUCON1、CTMUCON2、CTMUICON)を介して制御されます。CTMUCON1は、CTMUモジュールの有効化と動作モードの制御、およびエッジシーケンスの制御に使用します。CTMUCON2は、エッジソースの選択とエッジソースの極性選択に使用します。CTMUICON レジスタは、電流ソースの電流レンジの選択と、電流の調整に使用します。

25.1 静電容量計測

CTMU モジュールは、2 つの入力チャンネル間のエッジイベントの時間差に等しい幅を持つ出力パルスを生成する事によって、静電容量を計測します。両入力チャンネルに入力するパルスエッジイベントは、各種の内蔵周辺モジュール (OC1、Timer1、任意の入力キャプチャまたはコンパレータモジュール)と最大13本の外部ピン(CTED1~CTED13)から選択できます。このパルスをモジュールの高精度電流ソースと併せて使用し、下式に基づいて静電容量を計算します。

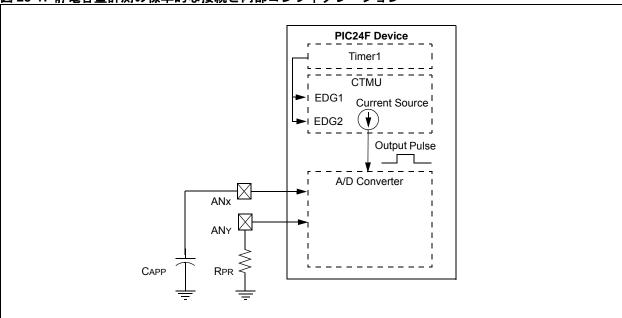
式 25-1:

$$I = C \cdot \frac{dV}{dT}$$

静電容量計測では、CTMU 出力のパルス発生後、A/D コンバータが入力チャンネルのいずれかで外付けコンデンサ (CAPP) をサンプリングします。これとは別のA/D チャンネルでは、電流ソースの校正用に高精度抵抗 (RPR) を使用します。パルス終了後、コンバータはコンデンサの電圧を計測します。静電容量の実際の計算は、アプリケーションによってソフトウェアで実行します。

図 25-1 に、静電容量計測に使用する外付け接続と、この用法における CTMU と A/D モジュールの関係を示します。この例では Timer1 からエッジイベントを供給していますが、Timer1 のかわりに外部エッジソースを使用するコンフィグレーションも可能です。 CTMU モジュールを使用した静電容量計測と時間計測の詳細は、関連する「PIC24F ファミリ リファレンス マニュアル」に記載しています。

図 25-1: 静電容量計測の標準的な接続と内部コンフィグレーション



25.2 時間計測

パルス幅の時間計測は、A/D モジュールの内部コンデンサ (CAD) と電流校正用の高精度抵抗を使用して、静電容量計測と同様に行えます。図 25-2 に、時間計測に使用する外付け接続と、この用法における CTMU と A/D モジュールの関係を示します。この例では、2 本の CTEDピンから外部エッジイベントを供給していますが、内部エッジソースを使用するコンフィグレーションも可能です。

25.3 パルスの生成と遅延

CTMU モジュールは、デバイスのシステムクロックとは同期しないパルスを生成する事もできます。 すなわち CTMU モジュールは、モジュールに入力されるエッジイベントに対して遅延したパルスを生成する事ができます (遅延量はプログラマブル)。

モジュールをパルス生成用に構成する場合、遅延量をTGEN ビット (CTMUCON<12>) で設定し、内部電流ソースをコンパレータ2のB入力に接続します。また、コンデンサ (CDELAY) をコンパレータ2ピン (C2INB)に接続し、コンパレータ参照電圧 (CVREF)をC2INAに接続します。さらに、CVREFに特定のトリップポイントを設定します。モジュールは、エッジイベント検出時にコンデンサ CDELAY の充電を開始します。CDELAYの充電電圧がCVREFトリップポイントを超えた時に、パルスがCTPLSに出力されます。パルスの遅延時間は、CDELAYとCVREFトリップポイントの値によって決まります。

図 25-3 に、パルス生成用の外付け接続と、使用する各アナログ モジュール間の関係を示します。図では入力パルスソースとして CTED1 を使用していますが、他のソースを選択する事もできます。CTMU モジュールを使用したパルス生成の詳細は、関連する「PIC24Fファミリ リファレンス マニュアル」に記載しています。

図 25-2: 時間計測の標準的な接続と内部コンフィグレーション

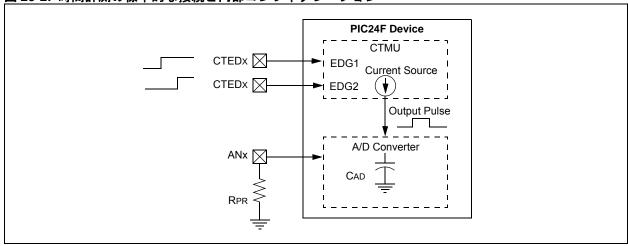
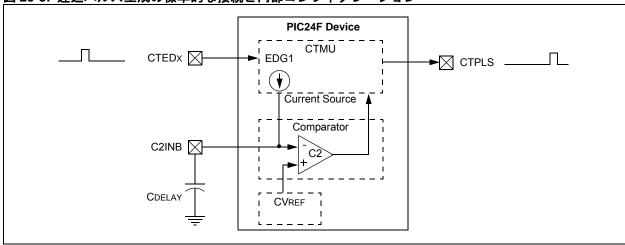


図 25-3: 遅延パルス生成の標準的な接続と内部コンフィグレーション



レジスタ 25-1: CTMUCON1: CTMU 制御レジスタ 1

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CTMUEN	_	CTMUSIDL	TGEN	EDGEN	EDGSEQEN	IDISSEN	CTTRIG
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
_	_	_	_	_	_	_	_
bit 7							bit 0

凡例:

bit 14

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「O」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15 CTMUEN: CTMU イネーブルビット

1 = モジュールを有効にする

0 = モジュールを無効にする **未実装:**「0」として読み出し

bit 13 CTMUSIDL: アイドルモード時停止ビット

1 = デバイスがアイドルモードに移行した時にモジュールの動作を停止する

0 = アイドルモード中もモジュールの動作を継続する

bit 12 TGEN: 時間生成イネーブルビット

1 = エッジ遅延生成を有効にする

0 = エッジ遅延生成を無効にする

bit 11 EDGEN: エッジ イネーブルビット

1 = エッジをブロックしない

0 = エッジをブロックする

bit 10 EDGSEQEN: エッジシーケンス イネーブルビット

1 = エッジ1イベントはエッジ2イベントよりも先に発生する必要がある

0 = エッジの順番を要求しない

bit 9 IDISSEN: アナログ電流ソース制御ビット

1 = アナログ電流ソース出力をグランドに接続する

0 = アナログ電流ソース出力をグランドに接続しない

bit 8 CTTRIG: トリガ制御ビット

1 = トリガ出力を有効にする

0 = トリガ出力を無効にする

bit 7-0 **未実装:**「0」として読み出し

レジスタ 25-2: CTMUCON2: CTMU 制御レジスタ 2

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EDG1EDGE	EDG1POL	EDG1SEL3	EDG1SEL2	EDG1SEL1	EDG1SEL0	EDG2	EDG1
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
EDG2EDGE	EDG2POL	EDG2SEL3	EDG2SEL2	EDG2SEL1	EDG2SEL0	_	_
bit 7							bit 0

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「O」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15 EDG1EDGE: エッジ1エッジセンシティブ選択ビット

1 = 入力でエッジを検出する 0 = 入力でレベルを検出する

bit 14 EDG1POL: エッジ 1 極性選択ビット

1 = エッジ1を立ち上がりエッジに応答させる 0 = エッジ1を立ち下がりエッジに応答させる

bit 13-10 **EDG1SEL<3:0>:** エッジ1ソース選択ビット

1111 = コンパレータ 3 出力をエッジ 1 のソースにする

1110 = コンパレータ2出力をエッジ1のソースにする

1101 = コンパレータ1出力をエッジ1のソースにする

1100 = IC3 をエッジ1のソースにする

1011 = IC2 をエッジ1のソースにする

1010 = IC1 をエッジ1のソースにする

1001 = CTED8 をエッジ1のソースにする

1000 = CTED7 をエッジ1のソースにする

0111 = CTED6 をエッジ1のソースにする

0110 = CTED5 をエッジ1のソースにする

0101 = CTED4 をエッジ1のソースにする

0100 = CTED3 をエッジ1のソースにする(2)

0011 = CTED1 をエッジ1のソースにする

0010 = CTED2 をエッジ1のソースにする

0001 **= OC1** をエッジ**1のソー**スにする

0000 = Timer1 をエッジ1のソースにする

bit 9 **EDG2**: エッジ 2 ステータスビット

このビットはエッジ2のステータスを示し、電流ソースを制御するために書き込みも可能です。

1 = エッジ2が発生した

0 = エッジ2は発生していない

bit 8 EDG1: エッジ1ステータスビット

このビットはエッジ1のステータスを示し、電流ソースを制御するために書き込みも可能です。

1 = エッジ1が発生した

0 = エッジ1は発生していない

bit 7 EDG2EDGE: エッジ2エッジセンシティブ選択ビット

1 = 入力をエッジセンシティブにする

0 = 入力をレベルセンシティブにする

Note 1: PIC24FV32KA302 では、エッジソース CTED11 と CTED12 を利用できません。

2: PIC24FV32KA301 では、エッジソース CTED3、CTED11、CTED12、CTED13 を利用できません。

レジスタ 25-2: CTMUCON2: CTMU 制御レジスタ 2 (続き)

bit 6 EDG2POL: エッジ2極性選択ビット 1=エッジ2を立ち上がりエッジに応答させる 0 = エッジ2を立ち下がりエッジに応答させる bit 5-2 EDG2SEL<3:0>: エッジ2ソース選択ビット 1111 = コンパレータ3出力をエッジ2のソースにする 1110 = コンパレータ2出力をエッジ2のソースにする 1101 = コンパレータ1出力をエッジ2のソースにする 1100 = 未実装(使用せず) 1011 = IC3 をエッジ2のソースにする 1010 = IC2 をエッジ2のソースにする 1001 = IC1 をエッジ2のソースにする 1000 = CTED13 をエッジ2のソースにする(2) 0111 = CTED12 をエッジ2のソースにする (1,2) 0110 = CTED11 をエッジ2のソースにする (1,2) 0101 = CTED10 をエッジ2のソースにする 0100 = CTED9 をエッジ2のソースにする 0011 = CTED1 をエッジ2のソースにする 0010 = CTED2 をエッジ2のソースにする 0001 = OC1 をエッジ2のソースにする

0000 = Timer1 をエッジ2のソースにする

bit 1-0 **未実装:**「0」として読み出し

Note 1: PIC24FV32KA302では、エッジソース CTED11と CTED12 を利用できません。

2: PIC24FV32KA301 では、エッジソース CTED3、CTED11、CTED12、CTED13 を利用できません。

レジスタ 25-3: CTMUICON: CTMU 電流制御レジスタ

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ITRIM5	ITRIM4	ITRIM3	ITRIM2	ITRIM1	ITRIM0	IRNG1	IRNG0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
_	_	_	_	_	_	_	_
bit 7							bit 0

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-10 ITRIM<5:0>: 電流ソース調整ビット

011111 = 公称電流に対する正方向の最大調整

011110

•

•

000001 = 公称電流に対する正方向の最小調整

000000 = IRNG<1:0> で指定された公称電流出力

111111 = 公称電流に対する負方向の最小調整

•

٠

100010

100001 = 公称電流に対する負方向の最大調整

bit 9-8 IRNG<1:0>: 電流ソース レンジ選択ビット

11 = 100 x ベース電流

10 = 10 x ベース電流

01 = ベース電流レベル (公称値: 0.55 µA)

00 = 1000 x ベース電流

bit 7-0 **未実装:**「0」として読み出し

NOTE:

26.0 その他の特殊な機能

Note:

本書は、PIC24 デバイスの中の特定製品グループを対象とし、それらの機能の概要説明を目的としています。従って本書は、包括的な参照資料の提供を意図したものではありません。ウォッチドッグタイマ、デバイス全体に関わる内蔵機能、プログラミングと診断の詳細は、「PIC24Fファミリリファレンスマニュアル」の下記の関連セクションを参照してください。

- ・セクション 9.「ウォッチドッグ タイマ (WDT)」(DS39697)
- セクション 36.「プログラマブル HIGH/LOW 電圧検出 (HLVD) 等の全体 に関わる内蔵機能」 (DS39725)
- ・セクション 33.「プログラミングと診 断」(DS39716)

PIC24FV32KA304 ファミリデバイスは、アプリケーションの柔軟性と信頼性を最大限に高め、外付け部品を少なくしてコストを最小に抑えるために、下記の各種機能を備えています。

- ・ 柔軟性の高いコンフィグレーション
- ウォッチドッグ タイマ (WDT)
- ・ コード保護
- インサーキット シリアル プログラミング (ICSP™)
- ・ インサーキット エミュレーション

26.1 コンフィグレーション ビット

各コンフィグレーション ビットの既定値状態を変更する(「0」に変更する)か、またはそのまま変更しない(「1」のままにする)かによって、各種の設定オプションを選択できます。これらのビットは、プログラムメモリ内のアドレス F80000h を先頭に配置されます。全てのコンフィグレーション レジスタのアドレスを表 26-1 に示します。各ビット機能の詳細な説明をレジスタ 26-1 ~レジスタ 26-8 に記載します。

アドレス F80000h はユーザ プログラムメモリ空間の外側である事に注意してください。このアドレスはコンフィグレーションメモリ空間 (800000h~FFFFFFh)に属し、アクセスにはテーブル読み書き操作が必要です。

表 26-1: コンフィグレーション レジスタのアドレス

コンフィグレーション レジスタ	アドレス
FBS	F80000
FGS	F80004
FOSCSEL	F80006
FOSC	F80008
FWDT	F8000A
FPOR	F8000C
FICD	F8000E
FDS	F80010

レジスタ 26-1: FBS: ブートセグメント コンフィグレーション レジスタ

<u> </u>					* * * *		
U-0	U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1
_	_	_	_	BSS2	BSS1	BSS0	BWRP
bit 7							bit 0

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「0」として読み出し

│-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 7-4 **未実装:**「0」として読み出し

bit 3-1 BSS<2:0>: ブートセグメント プログラム フラッシュ コード保護ビット

X11 = ブートプログラム フラッシュ セグメントなし

011 = 予約済み

110 = 標準セキュリティのブートプログラム フラッシュ セグメントは200hで始まり000AFEhで終わる

010 = 高セキュリティのブートプログラム フラッシュ セグメントは200hで始まり000AFEhで終わる

101 = 標準セキュリティのブートプログラム フラッシュ セグメントは 200h で始まり 0015FEh で終わる (1)

001 = 高セキュリティのブートプログラム フラッシュ セグメントは200hで始まり0015FEhで終わる $^{(1)}$ 100 = 標準セキュリティのブートプログラム フラッシュ セグメントは200hで始まり002BFEhで終わる $^{(1)}$

000=高セキュリティのブートプログラム フラッシュ セグメントは200hで始まり002BFEhで終わる⁽¹⁾

bit 0 BWRP: ブートセグメントのプログラム フラッシュ書き込み保護ビット

1 = ブートセグメントは書き込み可能

0 = ブートセグメントを書き込み保護する

Note 1: PIC24FV16KA3XXでは、この設定を使用しないでください。

レジスタ 26-2: FGS: 汎用セグメント コンフィグレーション レジスタ

U-0	U-0	U-0	U-0	U-0	U-0	R/C-1	R/C-1
_	_	_	_	_	_	GSS0	GWRP
bit 7							bit 0

凡.例:

R=読み出し可能ビット C=クリア可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 7-2 **未実装:**「0」として読み出し

bit 1 GSS0: 汎用セグメントコード フラッシュコード保護ビット

1 = 保護しない

0 = 標準セキュリティを有効にする

bit 0 GWRP: 汎用セグメントコード フラッシュ書き込み保護ビット

1 = 汎用セグメントは書き込み可能 0 = 汎用セグメントを書き込み保護する

レジスタ 26-3: FOSCSEL: オシレータ選択コンフィグレーション レジスタ

R/P-1	R/P-1	R/P-1	U-0	U-0	R/P-1	R/P-1	R/P-1
IESO	LPRCSEL	SOSCSRC	_	_	FNOSC2	FNOSC1	FNOSC0
bit 7							bit 0

凡例:

R=読み出し可能ビット P=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 7 IESO: 内部 / 外部切り換えビット

1 = 内部 / 外部切り換えモードを有効にする (2 段階起動を有効にする)

0 = 内部 / 外部切り換えモードを無効にする (2 段階起動を無効にする)

bit 6 LPRCSEL: 内部 LPRC オシレータ電源選択ビット

1=高消費電力/高精度モード

0 = 低消費電力 / 低精度モード

bit 5 SOSCSRC: セカンダリ オシレータ クロック源コンフィグレーション ビット

1 = SOSCI/SOSCO ピンで SOSC アナログ水晶振動子機能を利用可能にする

0 = SOSC 水晶振動子を無効にする; SOSCO ピンでデジタル SCLKI 機能を選択する

bit 4-3 **未実装:**「0」として読み出し

bit 2-0 **FNOSC<2:0>:** オシレータ選択ビット

000 = 高速 RC オシレータ (FRC)

001 = N 分周 PLL 併用 FRC オシレータ (FRCDIV+PLL)

010 = プライマリ オシレータ (XT、HS、EC)

011 = PLL 併用プライマリ オシレータ (HS+PLL、EC+PLL)

100 = セカンダリ オシレータ (SOSC)

101 = 低消費電力 RC オシレータ (LPRC)

110 = N 分周 500 kHz 低消費電力 FRC オシレータ (LPFRCDIV)

111 = N 分周 8 MHz FRC オシレータ (FRCDIV)

レジスタ 26-4: FOSC: オシレータ コンフィグレーション レジスタ

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
FCKSM1	FCKSM0	SOSCSEL	POSCFREQ1	POSCFREQ0	OSCIOFNC	POSCMD1	POSCMD0
bit 7							bit 0

凡例:

R=読み出し可能ビット P=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 7-6 FCKSM<1:0>: クロック切り換え / 監視選択コンフィグレーション ビット

1x = クロック切り換えとフェイルセーフ クロック監視機能を無効にする

01 = クロック切り換えを有効にし、ファイルセーフ クロック監視機能を無効にする00 = クロック切り換えを無効にし、フェイルセーフ クロック監視機能を有効にする

bit 5 SOSCSEL: セカンダリ オシレータ消費電力選択コンフィグレーション ビット

1 = セカンダリ オシレータを高消費電力動作に設定する 0 = セカンダリ オシレータを低消費電力動作に設定する

bit 4-3 POSCFREQ<1:0>: プライマリ オシレータ周波数レンジ コンフィグレーション ビット

11 = プライマリ オシレータ / 外部クロック入力周波数は 8 MHz より高い

10 = プライマリ オシレータ / 外部クロック入力周波数は 100 kHz ~ 8 MHz

01 = プライマリ オシレータ / 外部クロック入力周波数は 100 kHz より低い

00 = 予約済み(使用せず)

bit 2 OSCIOFNC: CLKO イネーブル コンフィグレーション ビット

1 = OSCO ピンで CLKO 出力信号をアクティブにする (CLKO をアクティブにするために、プライマリオシレータを無効化 (POSCMD<1:0> = 11) するか、外部クロックモード (EC)(POSCMD<1:0> = 00) に設定する必要があります)

0 = CLKO 出力を無効にする

bit 1-0 **POSCMD<1:0>:** プライマリ オシレータ コンフィグレーション ビット

11 = プライマリ オシレータモードを無効にする

10 = HS オシレータモードを選択する

01 = XT オシレータモードを選択する

00 = 外部クロックモードを選択する

レジスタ 26-5: FWDT: ウォッチドッグ タイマ コンフィグレーション レジスタ

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
FWDTEN1	WINDIS	FWDTEN0	FWPSA	WDTPS3	WDTPS2	WDTPS1	WDTPS0
bit 7							bit 0

凡例:

R=読み出し可能ビット P=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 7,5 **FWDTEN<1:0>:** ウォッチドッグ タイマ イネーブルビット

11 = WDT をハードウェアで有効にする

10 = WDT を SWDTEN ビットの設定で制御する

01 = WDTをデバイス動作時にだけ有効にする; スリープ時にWDTを無効にする; SWDTENビットを無効にする

00 = WDT をハードウェアで無効にする; SWDTEN ビットを無効にする

bit 6 WINDIS: ウィンドウ付きウォッチドッグ タイマ ディセーブル ビット

1 = 標準 WDT を選択する; ウィンドウ付き WDT を無効にする

0 = ウィンドウ付き WDT を有効にする

Note: WDT をハードウェアでもソフトウェアでも無効化 (FWDTEN<1:0> = 00 かつ RCON レジスタの SWDTEN = 0) している場合、CLRWDT 命令を実行してもデバイスリセットは発生しません。

bit 4 FWPSA: WDT プリスケーラ ビット

1 = WDT プリスケーラ比は 1:128

0 = WDT プリスケーラ比は 1:32

bit 3-0 **WDTPS<3:0>:** ウォッチドッグ タイマ ポストスケール選択ビット

1111 = 1:32,768

1110 = 1:16,384

1101 = 1:8,192

1100 = 1:4,096

1011 = 1:2,048

1010 = 1:1,024

1001 = 1:512

1000 = 1:256

0111 = 1:1280110 = 1:64

0110 - 1:32

0101 - 1.320100 = 1:16

0011 = 1:8

0010 = 1:4

0001 = 1:2

0000 = 1:1

レジスタ 26-6: FPOR: リセット コンフィグレーション レジスタ

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
MCLRE ⁽²⁾	BORV1 ⁽³⁾	BORV0 ⁽³⁾	I2C1SEL ⁽¹⁾	PWRTEN	LVRCFG ⁽¹⁾	BOREN1	BOREN0
bit 7							bit 0

凡例:

R=読み出し可能ビット P=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 7 MCLRE: MCLR ピン イネーブルビット (2)

1 = MCLR ピンを有効にする; RA5 入力ピンを無効にする

0 = RA5 入力ピンを有効にする; MCLR を無効にする

bit 6-5 **BORV<1:0>:** ブラウンアウト リセット イネーブルビット ⁽³⁾

11 = 最低電圧に設定されたブラウンアウト リセット

10 = ブラウンアウト リセット

01 = 最高電圧に設定されたブラウンアウト リセット

00 = POR 時のダウンサイド保護を有効にする - 「ゼロパワー」を選択する

bit 4 I2C1SEL: 代替 I2C1 ピン割り当てビット (1)

1 = SCL1/SDA1 ピンは既定値配置 0 = SCL1/SDA1 ピンは代替配置

bit 3 **PWRTEN:** パワーアップタイマ イネーブルビット

1 = PWRT を有効にする

0 = PWRT を無効にする

bit 2 LVRCFG: 低電圧レギュレータ コンフィグレーション ビット (1)

1 = 低電圧レギュレータは利用不可

0 = 低電圧レギュレータは利用可能; スリープ時に LVREN ビット (RCON<12>) で制御

bit 1-0 **BOREN<1:0>:** ブラウンアウト リセット イネーブルビット

11 = ブラウンアウト リセットをハードウェアのみで有効にする: SBOREN ビットは無効

10 = ブラウンアウト リセットをデバイス動作時にのみ有効にし、スリープ時に無効にする; SBOREN ビットは無効

01 = ブラウンアウト リセットを SBOREN ビットの設定で制御する

00 = ブラウンアウト リセットをハードウェアで無効にする: SBOREN ビットは無効

Note 1: この設定は「FV」デバイスにのみ適用されます。「F」デバイスでは、このビットは予約済みです。「1」 から変更しないでください。

2: MCLRE ヒューズは、Vpp ベースの ICSP™ モードエントリを使用している時にのみ変更できます。これ はユーザが誤って低電圧テストエントリからデバイスをロックアウトしてしまう事を防ぎます。

3: BOR 電圧の詳細は 29.0 「電気的特性」を参照してください。

レジスタ 26-7: FICD: インサーキット デバッガ コンフィグレーション レジスタ

R/P-1	U-0	U-0	U-0	U-0	U-0	R/P-1	R/P-1
DEBUG	_	_	_	_	_	FICD1	FICD0
bit 7							bit 0

凡例:

R=読み出し可能ビット P=書き込み可能ビット U=未実装ビット、「0」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 7 **DEBUG**: バックグラウンド デバッガ イネーブルビット

1 = バックグラウンド デバッガ機能を無効にする 0 = バックグラウンド デバッガ機能を有効にする

bit 6-2 **未実装:**「0」として読み出し bit 1-0 **FICD<1:0:>** ICD ピン選択ビット

11 = PGEC1/PGED1 をデバイスのプログラミング / デバッギングに使用する

10 = PGEC2/PGED2 をデバイスのプログラミング / デバッギングに使用する

01 = PGEC3/PGED3 をデバイスのプログラミング / デバッギングに使用する

00 = 予約済み(使用せず)

レジスタ 26-8: FDS: ディープスリープ コンフィグレーション レジスタ

R/P-1	R/P-1	U-0	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
DSWDTEN	DSBOREN	_	DSWDTOSC	DSWDTPS3	DSWDTPS2	DSWDTPS1	DSWDTPS0
bit 7							bit 0

凡.例:

R=読み出し可能ビット P=書き込み可能ビット U=未実装ビット、「O」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 7 DSWDTEN: ディープスリープ ウォッチドッグ タイマ イネーブルビット

> 1 = DSWDT を有効にする 0 = DSWDT を無効にする

bit 6 DSBOREN: ディープスリープ / 低消費電力 BOR イネーブルビット

(ディープスリープモード以外の動作には影響しません)

1 = ディープスリープ時にディープスリープ BOR を有効にする 0 = ディープスリープ時にディープスリープ BOR を無効にする

bit 5 未実装:「0」として読み出し

bit 4 **DSWDTOSC:** DSWDT 参照クロック選択ビット

> 1 = DSWDT は参照クロックとして LPRC を使用する 0 = DSWDT は参照クロックとして SOSC を使用する

bit 3-0 DSWDTPS<3:0>: ディープスリープ ウォッチドッグ タイマ ポストスケール選択ビット

DSWDT のプリスケーラ値は 32 です。これにより約1 ms のベースタイムが生成されます。

1111 = 1:2,147,483,648 (25.7 日) 公称值

1110 = 1:536,870,912 (6.4 日) 公称値

1101 = 1:134,217,728 (38.5 時間) 公称値

1100 = 1:33,554,432 (9.6 時間) 公称値

1011 = 1:8,388,608 (2.4 時間) 公称値

1010 = 1:2,097,152 (36 分) 公称值

1001 = 1:524,288 (9 分) 公称值

1000 = 1:131,072 (135 秒) 公称值

0111 = 1:32,768 (34 秒) 公称值

0110 = 1:8,192 (8.5 秒) 公称值

0101 = 1:2,048 (2.1 秒) 公称值

0100 = 1:512 (528 ms) 公称值 0011 = 1:128 (132 ms) 公称值

0010 = 1:32 (33 ms) 公称值

0001 = 1:8 (8.3 ms) 公称值

0000 = 1:2 (2.1 ms) 公称值

レジスタ 26-9: DEVID: デバイス ID レジスタ

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
_	_	_	_	_	_	_	_
bit 23							bit 16

R	R	R	R	R	R	R	R
FAMID7	FAMID6	FAMID5	FAMID4	FAMID3	FAMID2	FAMID1	FAMID0
bit 15							bit 8

R	R	R	R	R	R	R	R
DEV7	DEV6	DEV5	DEV4	DEV3	DEV2	DEV1	DEV0
bit 7							bit 0

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「O」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 23-16 **未実装:**「0」として読み出し

bit 15-8 **FAMID<7:0>:** デバイスファミリ ID ビット

01000101 = PIC24FV32KA304 ファミリ

bit 7-0 **DEV<7:0>:** デバイス ID ビット

00010111 = PIC24FV32KA304

00000111 = PIC24FV16KA304

00010011 = PIC24FV32KA302

00000011 = PIC24FV16KA302

00000011 - FIC24FV 10KA302

00011001 = PIC24FV32KA301

00001001 = PIC24FV16KA301

00010110 = PIC24F32KA304

00000110 = PIC24F16KA304

00010010 **= PIC24F32KA302**

00000010 = PIC24F16KA302

00011000 = PIC24F32KA301

00001000 = PIC24F16KA301

レジスタ 26-10: DEVREV: デバイス リビジョン レジスタ

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
_	_	_	_	_	_	_	_
bit 23							bit 16

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
_	_	_	_	_	_	_	
bit 15							bit 8

U-0	U-0	U-0	U-0	R	R	R	R
_	_	_	_	REV3	REV2	REV1	REV0
bit 7							bit 0

凡例:

R=読み出し可能ビット W=書き込み可能ビット U=未実装ビット、「O」として読み出し

-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 23-4 **未実装:**「0」として読み出し

bit 3-0 **REV<3:0>:** マイナー リビジョン ID ビット

26.2 内蔵電圧レギュレータ

PIC24FV32KA304 ファミリの全てのデバイスは、コアデジタルロジックに公称値3.0 Vの電圧を供給します。このため、これより高い typical 電圧 (最大 5.0 V)で動作する必要がある回路では問題が生じる可能性があります。システム設計を容易にするために、「FV」ファミリの全てのデバイスはレギュレータを内蔵しており、VDD からコア ロジックを駆動する事ができます。

このレギュレータは常時動作し、他の VDD ピンからコアへ電源を供給します。 VCAP ピンには低 ESR コンデンサ(セラミック コンデンサ等)を接続する必要があります(図 26-1 参照)。これは、レギュレータの安定性維持に役立ちます。このフィルタ コンデンサの推奨値は 29.1「DC 特性」に記載しています。

PIC24FJ64GA ファミリの場合、全てのデバイスでレギュレータが無効化されています。

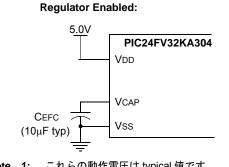
「F」デバイスでは VDDCORE と VDD ピンが内部で接続されており、動作許容電圧レンジは「FV」デバイスよりも全体的に低くなります (1.8 ~ 3.6 V)。図 26-1 に、使用可能なコンフィグレーションを示します。

26.2.1 電圧レギュレータのトラッキング モード と低電圧検出

全ての PIC24FV32KA304 では、内蔵電圧レギュレータが公称値 3.0 V の定電圧をデジタル コアロジックに 供給します。このレギュレータは、約 3.0 V ~デバイスの VDDMAX の VDD レンジで、公称値 3.0 V の電圧を供給できます。このレギュレータは、3.0 V よりも低い VDD レベルを昇圧する事はできません。レギュレータに供給される電圧が大きく低下した時、「ブラウンアウト」条件を回避するために、レギュレータはトラッキング モードに移行します。トラッキング モードでは、レギュレータの出力電圧は VDD に追従します。レギュレータによる電圧降下は 100mV (typical) です。

トラッキングモードでは、デバイスはフルスピードで動作できません。デバイスがトラッキングモードに移行した事を検出するために、内蔵レギュレータは単純な HIGH/LOW 電圧検出 (HLVD) 回路を備えています。VDD がフルスピード動作に必要なレンジより低下すると、この回路は HIGH/LOW 電圧検出割り込みフラグHLVDIF (IFS4<8>)をセットします。このフラグを使用して割り込みを生成し、アプリケーションを低消費電力動作モードに移行させるか、あるいは正常なシャットダウン処理を開始できます。HIGH/LOW電圧検出回路を備えるのは「FV」デバイスだけです。

図 26-1: 内蔵レギュレータの接続



Note 1: これらの動作電圧は typical 値です。 VDD と VDDCORE の動作レンジについては 29.0「電気的特性」を参照してください。

26.2.2 内蔵レギュレータと POR

PIC24FV32KA304 では、出力を生成するまでに約 $1 \mu s$ を要します。この遅延時間は TPM と呼ばれ、この間のコード実行は無効化されます。スリープモードを含むパワーダウン モード後にデバイスが動作を再開するたびに、毎回 TPM が適用されます。

26.3 ウォッチドッグ タイマ (WDT)

PIC24FV32KA304 ファミリのデバイスでは、LPRC オシレータを使用して WDT を駆動します。WDT を有効にすると、そのクロック源も有効化されます。

LPRC から供給される WDT クロック源の公称周波数は 31 kHz です。このクロックは、5 ビット (32 分周)動作または7 ビット (128 分周)動作用に設定可能なプリスケーラに供給されます。このプリスケーラはFWPSA コンフィグレーション ビットで設定します。31 kHz 入力の場合、プリスケーラの公称 WDT タイムアウト期間 (TWDT) は、5 ビットモードで 1 ms、7 ビットモードで 4 ms です。

可変ポストスケーラで WDT プリスケーラ出力を分周する事により、タイムアウト期間を延長できます。コンフィグレーションビット WDTPS<3:0>(FWDT<3:0>)を使用して、プリスケーラの分周比を 1:1 ~ 1:32,768 の範囲で 16 段階に選択できます。プリスケーラとポストスケーラを使用する事により、タイムアウト期間を 1 ms ~ 131 s の範囲で設定できます。

WDT、プリスケーラ、ポストスケーラは下記の条件でリセットされます。

- 全てのデバイスリセット時
- クロック切り換え完了時 (ソフトウェアによるクロック切り換え(NOSCビット変更後にOSWENビットをセット)またはハードウェア(フェイルセーフクロック監視機能)によるクロック切り換え)
- PWRSAV 命令実行時(スリープまたはアイドルモードへの移行時)
- デバイスがスリープまたはアイドルモードを終了して通常動作を再開する時
- 通常動作中に CLRWDT 命令を実行した時

WDT がハードウェアで有効化されている場合 (FWDTEN<1:0> = 11)、スリープまたはアイドルモード中も WDT は動作を継続します。WDT タイムアウトが発生するとデバイスは復帰し、PWRSAV 命令の次の命令からコード実行を再開します。デバイスが復帰した後に、対応する SLEEP または IDLE ビット (RCON<3:2>) をソフトウェアでクリアする必要があります。

WDT フラグビット WDTO (RCON4) は、WDT タイム アウト後に自動的にクリアされません。後続の WDT イベントを検出するために、このフラグをソフトウェ アでクリアする必要があります。

Note: CLRWDT 命令と PWRSAV 命令は、実行時にプリスケーラとポストスケーラのカウントをクリアします。

26.3.1 ウィンドウ付き動作

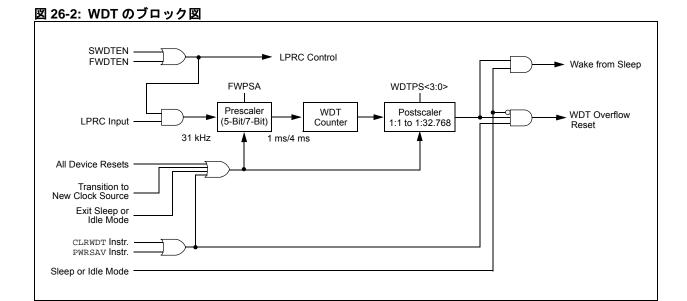
ウォッチドッグ タイマでは、動作オプションとして固定ウィンドウ モードを選択できます。このウィンドウ付きモードの場合、CLRWDT命令は設定されたWDT周期の最後の1/4期間中にのみ、WDTをリセットできます。このウィンドウより前にCLRWDT命令を実行した場合、WDTタイムアウトに似たWDTリセットが発生します。

ウィンドウ付き WDT モードは、コンフィグレーション ビット WINDIS (FWDT<6>) に「0」を書き込むと有効化されます。

26.3.2 制御レジスタ

WDT は、FWDTEN<1:0> コンフィグレーション ビットで有効化 / 無効化します。FWDTEN<1:0> コンフィグレーション ビットの両方のビットをセットすると、WDT は常時有効化されます。

FWDTEN<1:0> コンフィグレーション ビットに「10」を書き込むと、WDT をソフトウェアで制御できます。この場合、SWDTEN 制御ビット (RCON<5>) をセットする事により、WDT はソフトウェアで有効化されます。SWDTEN 制御ビットは、全てのデバイスリセットでクリアされます。ソフトウェア WDT オプションを使用する事により、ユーザ アプリケーションは最大限の省電力化を図るために、重要なコードセグメントで WDT を有効にし、重要ではないセグメントで WDT を無効にできます。FWTEN<1:0> ビットを「01」に設定した場合、WDT は動作モードおよびアイドルモード時にのみ有効化され、スリープモードでは無効化されます。この設定では、SWDTEN ビット (RCON<5>)による WDT のソフトウェア制御は無効化されます。



26.4 ディープスリープ ウォッチドッグ タイマ (DSWDT)

PIC24FV32KA304 ファミリのデバイスは、WDT モジュールに加えて、デバイスがディープスリープ中であっても動作する DSWDT モジュールを備えます。このモジュールの駆動には、SOSC または LPRC オシレータを使用します。クロック源はコンフィグレーション ビット DSWCKSEL(FDS<4>) で選択します。

ポストスケーラを選択する事により、DSWDT は 2.1 ms ~ 25.7 日の範囲でタイムアウトを生成できます。ポストスケーラは、コンフィグレーション ビット DSWDTPS<3:0>(FDS<3:0>) で選択します。DSWDT を有効にすると、そのクロック源も有効化されます。

DSWDT は、デバイスをディープスリープ モードから 復帰できる要因の 1 つです。

26.5 プログラムの検証と コード保護

PIC24FV32KA304 ファミリの全てのデバイスでは、コンフィグレーションビットBSS0 がブートセグメントのコード保護を制御し、コンフィグレーション ビットGSS0 が汎用セグメントのコード保護を制御します。これらのビットは、プログラムメモリ空間に対する外部読み書きを禁止します。これは通常の実行モードに直接影響しません。

BWRPビットはブートセグメントの書き込み保護を制御し、GWRP ビットはコンフィグレーション ワード内の汎用セグメントの書き込み保護を制御します。これらのビットに「0」を書き込むと、プログラムメモリに対する内部書き込み / 消去動作がブロックされます。

26.6 インサーキット シリアル プログラミング

PIC24FV32KA304 ファミリのマイクロコントローラは、完成品アプリケーション回路に実装した状態でシリアルプログラミングできます。このプログラミングは、クロック用 (PGECx) とデータ用 (PGEDx) の 2 本のラインと、電源、グランド、プログラミング電圧用の3本のラインを使用して簡単に行えます。この機能を利用すると、未プログラミングデバイスを実装したボードを製造し、製品の出荷直前にマイクロコントローラをプログラミングできます。また、最新ファームウェアやカスタマイズしたファームウェアをプログラミングする事もできます。

26.7 インサーキット デバッガ

MPLAB® ICD 3、MPLAB REAL ICE™、PICkit™ 3のいずれかをデバッガとして選択すると、インサーキットデバッグ機能が有効となります。この機能により、MPLAB IDE を使用して簡単にデバッグを行えます。デバッグ機能は、PGECx および PGEDx ピンを介して制御します。

デバイスのインサーキット デバッガ機能を使用するには、MCLR、VDD、VSS、PGECx、PGEDx とピンペアへの接続用に ICSP コネクタを回路に実装しておく必要があります。この機能を有効にすると、一部のリソースを一般用途に使用できなくなります。このようなリソースには、データ RAM の先頭 80 バイトと 2本の I/O ピンが含まれます。

27.0 開発サポート

PIC® マイクロコントローラと dsPIC® デジタル シグナル コントローラには、以下に示す豊富なソフトウェア およびハードウェア開発ツールによるサポートが用意されています。

- 統合開発環境
 - MPLAB® IDE ソフトウェア
- ・ コンパイラ/アセンブラ/リンカ
 - 各種デバイスファミリ用 MPLAB C コンパイラ
 - 各種デバイスファミリ用 HI-TECH C
 - MPASM™ アセンブラ
 - MPLINK™ オブジェクト リンカ /
 MPLIB™ オブジェクト ライブラリアン
 - 各種デバイスファミリ用 MPLAB アセンブラ / リンカ / ライブラリアン
- ・シミュレータ
 - MPLAB SIM ソフトウェア シミュレータ
- ・エミュレータ
 - MPLAB REAL ICE™ インサーキット エミュレータ
- インサーキット デバッガ
 - MPLAB ICD 3
 - PICkit™ 3 Debug Express
- ・ デバイス プログラマ
 - PICkit™2プログラマ
 - MPLAB PM3 デバイス プログラマ
- 低コストのデモボード、開発ボード、評価キット、 スタータキット

27.1 MPLAB 統合開発環境ソフトウェア

MPLAB IDE ソフトウェアを使用すると、従来の 8/16/32 ビット マイクロコントローラ市場では考えられないほど、ソフトウェアを容易に開発できます。 MPLAB IDE は下記の機能を備え、Windows® オペレーティング システム上で動作します。

- 全てのデバッグツールで共通のグラフィカル イン ターフェイス
 - シミュレータ
 - プログラマ(別売り)
 - インサーキットエミュレータ(別売り)
 - インサーキットデバッガ(別売り)
- コンテキスト色分け表示のフル機能エディタ
- ・ 複数プロジェクト管理機能
- 値を直接編集できるカスタマイズ可能なデータウィンドウ
- 高度なソースコード デバッグ
- マウスオーバーで変数の現在値を表示
- ソースウィンドウからウォッチ ウィンドウへの変数のドラッグ&ドロップ
- 充実したオンラインヘルプ
- IAR C コンパイラ等、一部サードパーティー製ツールの統合もサポート

MPLAB IDE を使用すると、下記の作業が可能です。

- ソースファイル (C またはアセンブリ)の編集
- コンパイル/アセンブルからエミュレータ/シミュレータ ツールへのダウンロードまでをワンタッチで実行(プロジェクトの全情報を自動更新)
- ・ 以下を使用したデバッグ
 - ソースファイル (C またはアセンブリ)
 - Cとアセンブリの混在使用
 - マシンコード

MPLAB IDE は、コスト効率の高いシミュレータから、低コストのインサーキット デバッガ、高機能なエミュレータまで、各種デバッグツールを共通の開発パラダイムでサポートします。このため、より高機能で強力なツールにアップグレードした場合でも、短期間で使用方法を習得できます。

27.2 各種デバイスファミリ用 MPLAB C コンパイラ

MPLAB C コンパイラ コード開発システムは、マイクロチップ社製 PIC18/PIC24/PIC32 ファミリ マイクロコントローラと dsPIC30/dsPIC33 ファミリ デジタルシグナル コントローラに対応する、ANSI 完全準拠のC コンパイラです。これらのコンパイラは、強力な統合機能と優れたコード最適化機能を備えながらも、容易に使用できます。

また、MPLAB IDE デバッガ用に最適化されたシンボル情報を出力できるため、ソースレベルのデバッグも容易です。

27.3 各種デバイスファミリ用 HI-TECH C

HI-TECH C コンパイラ コード開発システムは、マイクロチップ社製PICファミリ マイクロコントローラとdsPIC ファミリ デジタルシグナル コントローラに対応する、ANSI 完全準拠の C コンパイラです。これらのコンパイラは、強力な統合機能とインテリジェントなコード生成機能を備えながらも、容易に使用できます。

また、MPLAB IDE デバッガ用に最適化されたシンボル情報を出力できるため、ソースレベルのデバッグも容易です。

このコンパイラはマクロアセンブラ、リンカ、プリプロセッサ、ワンステップ ドライバを備え、複数のプラットフォーム上で動作します。

27.4 MPASM アセンブラ

MPASM アセンブラは、PIC10/12/16/18 MCU に対応したフル機能の汎用マクロアセンブラです。

MPASM アセンブラは、MPLINK オブジェクト リンカ 用の再配置可能なオブジェクト ファイル、Intel® 標準 HEX ファイル、メモリ使用量とシンボル参照の詳細を記述した MAP ファイル、ソースラインと生成されたマシンコードを含む絶対 LST ファイル、デバッグ用 COFF ファイルを生成します。

MPASM アセンブラの機能には下記が含まれます。

- MPLAB IDE プロジェクトへの統合
- ユーザ定義マクロによるアセンブリコードの最適化
- 多用途ソースファイルに対応する条件付きアセンブリ
- アセンブリ プロセスを完全に制御できるディレク ティブ

27.5 MPLINK オブジェクト リンカ / MPLIB オブジェクト ライブラリアン

MPLINK オブジェクト リンカは、MPASM アセンブラと MPLAB C18 C コンパイラが作成した再配置可能なオブジェクトを結合します。このオブジェクト リンカは、リンカスクリプトからのディレクティブを使用して、プリコンパイル済みライブラリから再配置可能なオブジェクトをリンクできます。

MPLIB オブジェクト ライブラリアンは、プリコンパイル済みコードのライブラリ ファイルの作成と変更を管理します。ライブラリのルーチンをソースファイルから呼び出すと、そのルーチンが含まれているモジュールのみがアプリケーションとリンクします。これにより、大きなライブラリを各種アプリケーションで効率的に使用できます。

オブジェクト リンカ / ライブラリの機能は以下の通りです。

- 多数の小さいファイルをリンクするのではなく、 1つのライブラリを効果的にリンクする
- 関連するモジュールをグループ化する事により、 コードの保守性が向上する
- モジュールのリスト表示、置換、削除、抽出を容易 に行え、ライブラリを柔軟に作成できる

27.6 各種デバイスファミリ用 MPLAB アセンブラ、リンカ、ライブラリアン

MPLAB アセンブラは、PIC24、PIC32、dsPIC デバイス用のシンボリック アセンブリ言語から再配置可能なマシンコードを生成します。MPLAB C コンパイラはこのアセンブラを使用してオブジェクト ファイルを生成します。このアセンブラが生成した再配置可能なオブジェクト ファイルをアーカイブまたは他の再配置可能なオブジェクト ファイルとリンクして、実行ファイルを生成します。アセンブラの主な機能は以下の通りです。

- デバイスの命令セットを完全にサポート
- 固定小数点と浮動小数点データをサポート
- ・ コマンドライン インターフェイス
- ・ 豊富なディレクティブ セット
- 柔軟なマクロ言語
- MPLAB IDE との互換性

27.7 MPLAB SIM ソフトウェア シミュレータ

MPLAB SIM ソフトウェア シミュレータは、PIC MCU と dsPIC® DSC を命令レベルでシミュレートする事により、PC ホスト環境でのコード開発を可能にします。任意の命令でデータ領域を検証または変更でき、総合的なスティミュラス コントローラから外的刺激を加える事ができます。レジスタのログをファイルに出力して、さらに詳細なランタイム分析も行えます。また、トレースバッファとロジック アナライザ ディスプレイを使用すると、シミュレータの機能を拡張してプログラムの実行、I/O の動作、大部分の周辺モジュール、内部レジスタを記録 / 追跡できます。

MPLAB SIM ソフトウェア シミュレータは、MPLAB C コンパイラ、MPASM/MPLAB アセンブラを使用したシンボリック デバッグを完全サポートしています。このソフトウェア シミュレータは、ハードウェアラボ環境外での柔軟なコード開発とデバッグを可能にする経済的で優れたソフトウェア開発ツールです。

27.8 MPLAB REAL ICE インサーキット エミュレータ システム

MPLAB REAL ICE インサーキット エミュレータ システムは、マイクロチップ社製フラッシュ DSC/MCU デバイス用の次世代型高速エミュレータです。このエミュレータを使用すると、MPLAB 統合開発環境 (IDE) の優れたグラフィカル ユーザインターフェイスを使用して、PIC® フラッシュ MCU と dsPIC® フラッシュ DSC のデバッグとプログラミングを行えます。

このエミュレータと PC との接続にはハイスピード USB 2.0 インターフェイスを使用し、対象デバイスとの接続にはインサーキット デバッガ システムと共通の RJ-11 コネクタか、高速で耐ノイズ性に優れる最新の LVDS インターフェイス (CAT5) を使用します。

エミュレータの更新用ファームウェアは、MPLAB IDE からダウンロードできます。MPLAB IDE の最新リリースに伴って、サポートするデバイスと新機能が追加されます。MPLAB REAL ICE は、低コスト/高速エミュレーション、リアルタイム変数ウォッチ、トレース解析、複雑なブレークポイント、高耐久性のプローブインターフェイス、接続ケーブルの長尺対応(最長3m)等、他社製エミュレータに比べて多くの利点を持ちます。

27.9 MPLAB ICD 3 インサーキット デバッガシステム

MPLAB ICD 3 インサーキット デバッガシステムは、マイクロチップ社製フラッシュ デジタルシグナル コントローラ (DSC) とマイクロコントローラ (MCU) に対応した、非常に対費用効果の高い高速ハードウェア デバッガ / プログラマです。このデバッガを使用すると、MPLAB 統合開発環境 (IDE) の使いやすい強力なグラフィカル ユーザ インターフェイスを使用して、PIC®フラッシュ マイクロコントローラと dsPIC® DSC のデバッグとプログラミングを行えます。

MPLAB ICD 3 インサーキット デバッガのプローブは、PC との接続にハイスピード USB 2.0 インターフェイスを使用し、対象デバイスとの接続に MPLAB ICD 2/MPLAB REAL ICE システムと互換のコネクタ (RJ-11)を使用します。MPLAB ICD 3 は全ての MPLAB ICD 2 ヘッダをサポートしています。

27.10 PICkit 3 インサーキット デバッガ/ プログラマと PICkit 3 Debug Express

MPLAB PICkit 3 は、MPLAB 統合開発環境 (IDE) の強力な GUI を使用して PIC® および dsPIC® フラッシュマイクロコントローラをデバッグ / プログラミングできる低価格なツールです。MPLAB PICkit 3 と PC の接続にはフルスピード USB インターフェイスを使用します。対象デバイスとの接続には、MPLAB ICD 3/MPAB REAL ICE と互換のマイクロチップ デバッグコネクタ (RJ-11) を使用します。このコネクタは、2本のデバイス I/O ピンとリセットラインを使用して、インサーキット デバッグとインサーキット シリアル プログラミングを実現します。

PICkit 3 Debug Express は、PICkit 3、デモボードとマイクロコントローラ、フックアップ ケーブル、CD-ROM (ユーザガイド、レッスン、チュートリアル、コンパイラ、MPLAB IDE ソフトウェア収録)を含みます。

27.11 PICkit 2 開発用プログラマ / デバッガと PICkit 2 Debug Express

PICkit™ 2 開発用プログラマ / デバッガは、使いやすい インターフェイスを使用してマイクロチップ社のフ ラッシュ マイクロコントローラ ファミリのプログラ ミングとデバッグを行える低コストの開発ツールで す。高機能な Windows® プログラミング インターフェ イスは、8/16/32 ビット マイクロコントローラのベー スライン (PIC10F、PIC12F5xx、PIC16F5xx)、ミッ ドレンジ (PIC12F6xx、PIC16F)、PIC18F、PIC24、 dsPIC30、dsPIC33、PIC32 ファミリと、多くのマイ クロチップ社製シリアル EEPROM 製品をサポートし ています。マイクロチップ社の強力な MPLAB 統合開 発環境 (IDE) を使用する事により、PICkit™ 2 はほと んどの PIC® マイクロコントローラでインサーキット デバッグを行えます。インサーキット デバッグでは、 PIC マイクロコントローラをアプリケーションに組み 込んだままの状態で、プログラムの実行 / 停止とシン グルステップ実行を行えます。ブレークポイントで停 止させてファイル レジスタを確認 / 変更する事もでき ます。

PICkit 2 Debug Express は、PICkit 2、デモボードとマイクロコントローラ、フックアップ ケーブル、CD-ROM (ユーザガイド、レッスン、チュートリアル、コンパイラ、MPLAB IDE ソフトウェア収録) を含みます。

27.12 MPLAB PM3 デバイス プログラマ

MPLAB PM3 デバイス プログラマは CE 準拠の汎用デ バイス プログラマであり、VDDMIN と VDDMAX でのプ ログラマブル電圧検証によって高い信頼性を確保しま す。このデバイス プログラマは、メニューとエラー メッセージを表示する大型LCD (128×64)と、各種パッ ケージ タイプに対応するための脱着可能なモジュー ル式ソケット アセンブリを備えます。ICSP™ ケーブ ルは標準で付属しています。 スタンドアロン モードで は、MPLAB PM3 デバイス プログラマを PC へ接続せ ずに、PIC デバイスの読み出し、検証、プログラム実 行を行えます。また、このモードでコード保護も設定 できます。MPLAB PM3 とホスト PC との接続には、 RS-232 または USB ケーブルを使用します。さらに、 大容量メモリデバイスの高速プログラミングを可能に する高速通信と最適化アルゴリズムを備え、ファイル 保存とデータ アプリケーションのための MMC カード を内蔵しています。

27.13 デモボード、開発ボード、 評価キット、スタータキット

各種 PIC MCU と dsPIC DSC には多彩なデモボード、開発ボード、評価ボードを使用でき、完全に機能するシステム上でアプリケーションを迅速に開発できます。ほとんどのボードは、カスタム回路を追加するためのプロトタイプ領域を備えています。また、付属のアプリケーション ファームウェアとソースコードを使用して動作を評価できます。これらを編集して使用する事もできます。

これらのボードは、LED、温度センサ、スイッチ、スピーカ、RS-232 インターフェイス、LCD、ポテンショメータ、増設 EEPROM メモリ等、幅広い機能をサポートしています。

デモボードと開発ボードは、カスタム回路の試作と各種マイクロコントローラ アプリケーションの学習教材として使用できます。

PICDEM[™] およびdsPICDEM[™] デモ / 開発ボード シリーズの回路の他に、アナログフィルタ設計、KeeLoq[®] セキュリティ IC、CAN IrDA[®]、PowerSmart バッテリ管理、SEEVAL[®]評価システム、 Δ Σ ADC、流量検出等に対応する評価キットとデモソフトも各種取り揃えています。

また、特定のデバイスの評価に必要なツールー式を備えたスタータキットも用意しています。通常、スタータキットには、1つのアプリケーションとデバッグ機能を全て組み込んだ1枚のボードが含まれます。

デモボード、開発ボード、評価キットの一覧は、マイクロチップ社のウェブページ (www.microchip.com) でご覧ください。

28.0 命令セットの要約

Note: 本セクションには、PIC24F 命令セットアーキテクチャの簡単な要約を記載します。これらは、包括的な参照資料を提供するものではありません。

PIC24Fの命令セットは、従来のPIC® MCUの命令セットから大幅に拡張されていますが、これらのPIC MCUから容易に移行できるように互換性も維持しています。命令のほとんどはシングルプログラムメモリワードです。3つの命令だけが2プログラムメモリワードを必要とします。

各シングルワード命令は24 ビット ワードであり、これは命令のタイプを指定する8 ビットオペコードと、命令の動作を詳しく指定する1つまたは複数のオペランドにより構成されます。この命令セットは直交性が高く、下記の5つの基本カテゴリに分類されます。

- ワード指向またはバイト指向の演算
- ・ ビット指向の演算
- ・ リテラル演算
- 制御演算

表 28-1 に、命令の説明に使用する記号の凡例を示します。表 28-2 に、PIC24F 命令セットの全ての命令とその命令に影響を受けるステータスフラグを示します。

バレルシフト命令を含むバイト指向Wレジスタ命令のほとんどは、以下の3つのオペランドを使用します。

- 最初のソースオペランド(一般的にアドレス修飾子なしのレジスタ「Wb」)
- 2番目のソースオペランド(一般的にアドレス修飾 子ありまたはなしのレジスタ「Ws」)
- 結果の格納先(一般的にアドレス修飾子ありまたはなしのレジスタ「Wd」)

これに対し、ワード指向またはバイト指向のファイル レジスタ命令は、以下の2つのオペランドを使用しま す。

- ファイルレジスタ (値「f」で指定)
- 格納先(ファイルレジスタ「f」または「WREG」で表されるW0レジスタのいずれか)

単純なローテート/シフト命令を含むビット指向命令 のほとんどは、以下の2つのオペランドを使用します。

- W レジスタ (アドレス修飾子「あり」または「なし」)、 またはファイルレジスタ (「Ws」または「f」の値で指定)
- W レジスタまたはファイルレジスタのビット位置 (リテラル値またはレジスタ「Wb」の内容で間接的 に指定)

データの移動を伴うリテラル命令は、以下のオペランドの一部を使用できます。

- Wレジスタまたはファイルレジスタに書き込むリテラル値(「k」の値で指定)
- リテラル値書き込み先のWレジスタまたはファイルレジスタ(「Wb」または「f」で指定)

これに対し、算術演算または論理演算を伴うリテラル 命令は、以下のオペランドの一部を使用します。

- 最初のソースオペランド(一般的にアドレス修飾子なしのレジスタ「Wb」)
- 2番目のソースオペランド(リテラル値)
- 結果の格納先(最初のソースオペランドと異なる場合のみ、一般的にアドレス修飾子ありまたはなしのレジスタ「Wd」)

制御命令に使用できるオペランドには下記が含まれます。

- ・ プログラムメモリ アドレス
- テーブル読み出し/書き込み命令のモード

ダブルワード命令以外の全ての命令はシングルワード命令です。ダブルワード命令は、48 ビットで全ての必要な情報が得られるように2ワードで構成されています。この場合、2ワード目の8 MSb は「0」です。このため、2 ワード目を単独の命令として実行すると、NOP として実行されます。

シングルワード命令のほとんどは 1 命令サイクルで実行されます。ただし、条件付きテストの結果が「真」である場合、または命令実行の結果としてプログラムカウンタ (PC) が変化する場合はこの限りではありません。これらの場合、NOP 命令サイクルが追加されるため、実行には 2 命令サイクルを要します。注意すべき例外は、BRA (無条件/計算分岐)、間接CALL/GOTO、全てのテーブル読み出し/書き込み、RETURN/RETFIE 命令です。これらは1ワード命令ですが、実行には 2 サイクルまたは 3 サイクルを要します。

後続命令のスキップを伴う命令は、スキップされる命令がシングルワード命令かダブルワード命令かによって、スキップ実行時に2サイクルまたは3サイクルを要します。また、ダブルワード移動には2サイクルを要します。ダブルワード命令は2命令サイクルで実行されます。

表 28-1: オペコードの説明に使用する記号

フィールド	内容
#text	text で定義されるリテラル
(text)	text の内容
[text]	text が指すアドレス
{ }	オプションのフィールドまたは演算
<n:m></n:m>	レジスタビット フィールド
b	バイトモード選択
d	ダブルワードモード選択
S	シャドウレジスタ選択
W	ワードモード選択(既定値)
bit4	4 ビット選択フィールド (ワードアドレス命令で使用) ∈ {015}
C, DC, N, OV, Z	MCU ステータスビット: キャリー、デジットキャリー、負数、オーバーフロー、ゼロ
Expr	絶対アドレス、ラベル、式(リンカで解決)
f	ファイルレジスタ アドレス ∈ {0000h1FFFh}
lit1	1 ビット符号なしリテラル ∈ {0,1}
lit4	4 ビット符号なしリテラル ∈ {015}
lit5	5 ビット符号なしリテラル ∈ {031}
lit8	8 ビット符号なしリテラル ∈ {0255}
lit10	10 ビット符号なしリテラル ∈ 、バイトモードでは {0255}、ワードモードでは {0:1023}
lit14	14 ビット符号なしリテラル ∈ {016384}
lit16	16 ビット符号なしリテラル ∈ {065535}
lit23	23 ビット符号なしリテラル ∈ {08388608}; LSB は「0」である事が必要
なし	フィールドへの入力は必須ではなく、空白のままでも良い
PC	プログラム カウンタ
Slit10	10 ビット符号付きリテラル ∈ {-512511}
Slit16	16 ビット符号付きリテラル ∈ {-3276832767}
Slit6	6 ビット符号付きリテラル ∈ {-1616}
Wb	ベース W レジスタ ∈ {W0W15}
Wd	格納先 W レジスタ ∈ { Wd, [Wd], [Wd++], [Wd], [++Wd], [Wd] }
Wdo	格納先 W レジスタ ∈ { Wnd, [Wnd], [Wnd++], [Wnd], [++Wnd], [Wnd], [Wnd+Wb] }
Wm,Wn	被除数/除数ワーキング レジスタ ペア(直接アドレッシング)
Wn	16 個のワーキング レジスタの内の 1 つ ∈ {W0W15}
Wnd	16 個の格納先ワーキング レジスタの内の 1 つ ∈ {W0W15}
Wns	16 個の格納元ワーキング レジスタの内の 1 つ ∈ {W0W15}
WREG	WO(ファイルレジスタ命令で使用するワーキング レジスタ)
Ws	格納元 W レジスタ ∈ { Ws, [Ws], [Ws++], [Ws], [++Ws], [Ws] }
Wso	格納元 W レジスタ ∈ { Wns, [Wns], [Wns++], [Wns], [++Wns], [Wns], [Wns+Wb] }

表 28-2: 命令セットの概要

アセンブリ ニーモニック		アセンブリ構文	内容	ワード 数	サイクル 数	影響を受ける ステータスフラグ
ADD	ADD	f	f = f + WREG	1	1	C, DC, N, OV, Z
	ADD	f,WREG	WREG = f + WREG	1	1	C, DC, N, OV, Z
	ADD	#lit10,Wn	Wd = lit10 + Wd	1	1	C, DC, N, OV, Z
	ADD	Wb,Ws,Wd	Wd = Wb + Ws	1	1	C, DC, N, OV, Z
	ADD	Wb,#lit5,Wd	Wd = Wb + lit5	1	1	C, DC, N, OV, Z
ADDC	ADDC	f	f = f + WREG + (C)	1	1	C, DC, N, OV, Z
	ADDC	f,WREG	WREG = f + WREG + (C)	1	1	C, DC, N, OV, Z
	ADDC	#lit10,Wn	Wd = lit10 + Wd + (C)	1	1	C, DC, N, OV, Z
	ADDC	Wb,Ws,Wd	Wd = Wb + Ws + (C)	1	1	C, DC, N, OV, Z
	ADDC	Wb,#lit5,Wd	Wd = Wb + lit5 + (C)	1	1	C, DC, N, OV, Z
AND	AND	f	f = f .AND.WREG	1	1	N, Z
	AND	f,WREG	WREG = f .AND.WREG	1	1	N, Z
	AND	#lit10,Wn	Wd = lit10 .AND.Wd	1	1	N, Z
	AND	Wb,Ws,Wd	Wd = Wb .AND.Ws	1	1	N, Z
	AND	Wb,#lit5,Wd	Wd = Wb .AND. lit5	1	1	N, Z
ASR	ASR	f	f = f を算術右シフト	1	1	C, N, OV, Z
	ASR	f,WREG	WREG = f を算術右シフト	1	1	C, N, OV, Z
	ASR	Ws,Wd	Wd = Ws を算術右シフト	1	1	C, N, OV, Z
	ASR	Wb, Wns, Wnd	Wnd = Wns 分だけ Wb を算術右シフト	1	1	N, Z
	ASR	Wb,#lit5,Wnd	Wnd = lit5 分だけ Wb を算術右シフト	1	1	N, Z
BCLR	BCLR	f,#bit4	Bit Clear f	1	1	なし
	BCLR	Ws,#bit4	Ws をビットクリア	1	1	なし
BRA	BRA	C,Expr	キャリーの場合は分岐	1	1 (2)	なし
	BRA	GE,Expr	等しいか、より大きい場合は分岐	1	1 (2)	なし
	BRA	GEU, Expr	符号なしで等しいか、より大きい場合は分岐	1	1 (2)	なし
	BRA	GT,Expr	より大きい場合は分岐	1	1 (2)	なし
	BRA	GTU, Expr	符号なしでより大きい場合は分岐	1	1(2)	なし
	BRA	LE,Expr	等しいか、より小さい場合は分岐	1	1(2)	なし
	BRA	LEU, Expr	符号なしで等しいか、より小さい場合は分岐	1	1 (2)	なし
	BRA	LT,Expr	より小さい場合は分岐	1	1 (2)	なし
	BRA	LTU, Expr	符号なしでより小さい場合は分岐	1	1 (2)	なし
	BRA	N,Expr	負の場合は分岐	1	1 (2)	なし
	BRA	NC,Expr	キャリーなしの場合は分岐	1	1 (2)	なし
	BRA	NN,Expr	負でない場合は分岐	1	1 (2)	なし
	BRA	NOV,Expr	オーバーフローでない場合は分岐	1	1 (2)	なし
	BRA	NZ,Expr	ゼロでない場合は分岐	1	1 (2)	なし
	BRA	OV,Expr	オーバーフローの場合は分岐	1	1 (2)	なし
	BRA	Expr	無条件で分岐	1	2	なし
	BRA	Z,Expr	ゼロの場合は分岐	1	1 (2)	なし
	BRA	Wn	計算分岐	1	2	なし
BSET	BSET	f,#bit4	Bit Set f	1	1	なし
	BSET	Ws,#bit4	Ws をビットセット	1	1	なし
BSW	BSW.C	Ws,Wb	Ws <wb> に C ビット書き込み</wb>	1	1	なし
	BSW.Z	Ws,Wb	Ws <wb> に Z ビット書き込み</wb>	1	1	なし
BTG	BTG	f,#bit4	fをビットトグル	1	1	なし
	BTG	Ws,#bit4	Ws をビットトグル	1	1	なし
BTSC	BTSC	f,#bit4	fをビットテスト、クリアの場合はスキップ	1	1 (2 or 3)	なし
	BTSC	Ws,#bit4	Ws をビットテスト、クリアの場合はスキップ	1	1 (2 or 3)	なし

アセンブリ ニーモニック		アセンブリ構文	内容	ワード 数	サイクル 数	影響を受ける ステータスフラグ
BTSS	BTSS	f,#bit4	fをビットテスト、セットの場合はスキップ	1	1 (2 or 3)	なし
	BTSS	Ws,#bit4	Ws をビットテスト、セットの場合はスキップ	1	1 (2 or 3)	なし
BTST	BTST	f,#bit4	fをビットテスト	1	1	Z
	BTST.C	Ws,#bit4	Ws をビットテストして C へ	1	1	С
	BTST.Z	Ws,#bit4	Ws をビットテストして Z へ	1	1	Z
	BTST.C	Ws,Wb	Ws <wb> をビットテストして C へ</wb>	1	1	С
	BTST.Z	Ws,Wb	Ws <wb> をビットテストして Z へ</wb>	1	1	Z
BTSTS	BTSTS	f,#bit4	fをビットテストして、その後セット	1	1	Z
	BTSTS.C	Ws,#bit4	Ws をビットテストして C へ、その後セット	1	1	С
	BTSTS.Z	Ws,#bit4	Ws をビットテストして Z へ、その後セット	1	1	Z
CALL	CALL	lit23	サブルーチン呼び出し	2	2	なし
	CALL	Wn	間接サブルーチン呼び出し	1	2	なし
CLR	CLR	f	f = 0x0000	1	1	なし
	CLR	WREG	WREG = 0x0000	1	1	なし
	CLR	Ws	Ws = 0x0000	1	1	なし
CLRWDT	CLRWDT		ウォッチドッグ タイマをクリア	1	1	WDTO, Sleep
COM	COM	f	$f = \overline{f}$	1	1	N, Z
	COM	f,WREG	WREG = \overline{f}	1	1	N, Z
	COM	Ws,Wd	$Wd = \overline{Ws}$	1	1	N, Z
CP	CP	f	fをWREG とコンペア	1	1	C, DC, N, OV, Z
	CP	Wb,#lit5	Wb を lit5 とコンペア	1	1	C, DC, N, OV, Z
	CP	Wb,Ws	Wb を Ws とコンペア (Wb – Ws)	1	1	C, DC, N, OV, Z
CP0	CP0	f	f を 0x0000 とコンペア	1	1	C, DC, N, OV, Z
	CP0	Ws	Ws を 0x0000 とコンペア	1	1	C, DC, N, OV, Z
CPB	CPB	f	ボロー付きで f を WREG とコンペア	1	1	C, DC, N, OV, Z
	CPB	Wb,#lit5	ボロー付きで Wb を lit5 とコンペア	1	1	C, DC, N, OV, Z
	CPB	Wb,Ws	ボロー付きで_Wb を Ws とコンペア (Wb – Ws – C)	1	1	C, DC, N, OV, Z
CPSEQ	CPSEQ	Wb,Wn	Wb を Wn とコンペア、等しい場合はスキップ	1	1 (2 or 3)	なし
CPSGT	CPSGT	Wb,Wn	Wb を Wn とコンペア、より大きい場合はスキップ	1	1 (2 or 3)	なし
CPSLT	CPSLT	Wb,Wn	Wb を Wn とコンペア、より小さい場合はスキップ	1	1 (2 or 3)	なし
CPSNE	CPSNE	Wb,Wn	Wb を Wn とコンペア、等しくない場合はスキップ	1	1 (2 or 3)	なし
DAW	DAW	Wn	Wn = 10 進数調整 Wn	1	1	С
DEC	DEC	f	f = f –1	1	1	C, DC, N, OV, Z
	DEC	f,WREG	WREG = f –1	1	1	C, DC, N, OV, Z
	DEC	Ws,Wd	Wd = Ws - 1	1	1	C, DC, N, OV, Z
DEC2	DEC2	f	f = f - 2	1	1	C, DC, N, OV, Z
	DEC2	f,WREG	WREG = f - 2	1	1	C, DC, N, OV, Z
	DEC2	Ws,Wd	Wd = Ws - 2	1	1	C, DC, N, OV, Z
DISI	DISI	#lit14	k 命令サイクルの間割り込み禁止	1	1	なし
DIV	DIV.SW	Wm,Wn	符号付き 16 ビット /16 ビット整数除算	1	18	N, Z, C, OV
	DIV.SD	Wm,Wn	符号付き 32 ビット /16 ビット整数除算	1	18	N, Z, C, OV
	DIV.UW	Wm,Wn	符号なし 16 ビット /16 ビット整数除算	1	18	N, Z, C, OV
	DIV.UD	Wm,Wn	符号なし 32 ビット /16 ビット整数除算	1	18	N, Z, C, OV
EXCH	EXCH	Wns,Wnd	Wns と Wnd を交換	1	1	なし
FF1L	FF1L	Ws,Wnd	左 (MSb) 側から先頭を見つける	1	1	С
FF1R	FF1R	Ws,Wnd	右 (LSb) 側から先頭を見つける	1	1	С

アセンブリ ニーモニック		トの概要(続き)	内容	ワード 数	サイクル 数	影響を受ける ステータスフラグ
GOTO	GOTO	Expr	アドレスへ移動	2	2	なし
	GOTO	Wn	間接で移動	1	2	なし
INC	INC	f	f = f + 1	1	1	C, DC, N, OV, Z
	INC	f,WREG	WREG = f + 1	1	1	C, DC, N, OV, Z
	INC	Ws,Wd	Wd = Ws + 1	1	1	C, DC, N, OV, Z
INC2	INC2	f	f = f + 2	1	1	C, DC, N, OV, Z
	INC2	f,WREG	WREG = f + 2	1	1	C, DC, N, OV, Z
	INC2	Ws,Wd	Wd = Ws + 2	1	1	C, DC, N, OV, Z
IOR	IOR	f	f = f.IOR.WREG	1	1	N, Z
	IOR	f,WREG	WREG = f.IOR.WREG	1	1	N, Z
	IOR	#lit10,Wn	Wd = lit10 .IOR.Wd	1	1	N, Z
	IOR	Wb,Ws,Wd	Wd = Wb .IOR.Ws	1	1	N, Z
	IOR	Wb,#lit5,Wd	Wd = Wb .IOR. lit5	1	1	N, Z
LNK	LNK	#lit14	フレームポインタをリンク	1	1	なし
LSR	LSR	f	f=fを論理右シフト	1	1	C, N, OV, Z
	LSR	f,WREG	WREG = f を論理右シフト	1	1	C, N, OV, Z
	LSR	Ws,Wd	Wd = Ws を論理右シフト	1	1	C, N, OV, Z
	LSR	Wb,Wns,Wnd	Wnd = Wns 分だけ Wb を論理右シフト	1	1	N, Z
	LSR	Wb,#lit5,Wnd	Wnd = lit5 分だけ Wb を論理右シフト	1	1	N, Z
MOV	MOV	f,Wn	f を Wn へ移動	1	1	なし
	MOV	[Wns+Slit10],Wnd	[Wns+Slit10] を Wnd へ移動	1	1	なし
	MOV	f	fをfへ移動	1	1	N, Z
	MOV	f,WREG	f を WREG へ移動	1	1	N, Z
	MOV	#lit16,Wn	16 ビットリテラルを Wn へ移動	1	1	なし
	MOV.b	#lit8,Wn	8 ビットリテラルを Wn へ移動	1	1	なし
	MOV	Wn,f	Wn を f へ移動	1	1	なし
	MOV	Wns,[Wns+Slit10]	Wns を [Wns+Slit10] へ移動	1	1	なし
	MOV	Wso,Wdo	Ws を Wd へ移動	1	1	なし
	MOV	WREG, f	WREG を f へ移動	1	1	N, Z
	MOV.D	Wns,Wd	ダブルを W(ns):W(ns+1) から Wd へ移動	1	2	なし
	MOV.D	Ws,Wnd	ダブルを Ws から W(nd+1):W(nd) へ移動	1	2	なし
MUL	MUL.SS	Wb, Ws, Wnd	{Wnd + +1, Wnd} = 符号付き (Wb) * 符号付き (Ws)	1	1	なし
	MUL.SU	Wb,Ws,Wnd	{Wnd + +1, Wnd} = 符号付き (Wb) * 符号なし (Ws)	1	1	なし
	MUL.US	Wb,Ws,Wnd	{Wnd + +1, Wnd} = 符号なし (Wb) * 符号付き (Ws)	1	1	なし
	MUL.UU	Wb, Ws, Wnd	{Wnd + +1, Wnd} = 符号なし (Wb) * 符号なし (Ws)		1	なし
	MUL.SU	Wb,#lit5,Wnd	{Wnd + +1, Wnd} = 符号付き (Wb) * 符号なし (lit5)	1	1	なし
	MUL.UU	Wb,#lit5,Wnd	{Wnd + +1, Wnd} = 符号なし (Wb) * 符号なし (lit5)	1	1	なし
	MUL	f	W3:W2 = f * WREG	1	1	なし
NEG	NEG	f	f = f + 1	1	1	C, DC, N, OV, Z
	NEG	f,WREG	WREG = f + 1	1	1	C, DC, N, OV, Z
	NEG	Ws,Wd	$Wd = \overline{Ws} + 1$	1	1	C, DC, N, OV, Z
NOP	NOP		NOP	1	1	なし
	NOPR		NOP	1	1	なし
POP	POP	f	Top-of-Stack (TOS) から f をポップ	1	1	なし
	POP	Wdo	Top-of-Stack (TOS) から Wdo ヘポップ	1	1	なし
	POP.D	Wnd	Top-of-Stack (TOS) から W(nd):W(nd+1) へポップ	1	2	なし
	POP.S		シャドウレジスタをポップ	1	1	全て
PUSH	PUSH	f	Top-of-Stack (TOS) へ f をプッシュ	1	1	なし
	PUSH	Wso	Top-of-Stack (TOS) へ Wso をプッシュ	1	1	なし
	PUSH.D	Wns	Top-of-Stack (TOS) へ W(ns):W(ns+1) をプッシュ	1	2	なし
	PUSH.S		シャドウレジスタをプッシュ	1	1	なし

アセンブリ ニーモニック		アセンブリ構文	内容	ワード 数	サイクル 数	影響を受ける ステータスフラグ
PWRSAV	PWRSAV	#lit1	スリープまたはアイドルモードへ移行	1	1	WDTO, スリープ
RCALL	RCALL	Expr	相対コール	1	2	なし
	RCALL	Wn	加減算コール	1	2	なし
REPEAT	REPEAT	#lit14	次の命令を lit14 + 1 回繰り返す	1	1	なし
	REPEAT	Wn	次の命令を (Wn) + 1 回繰り返す	1	1	なし
RESET	RESET		ソフトウェア デバイス リセット	1	1	なし
RETFIE	RETFIE		割り込みからの復帰	1	3 (2)	なし
RETLW	RETLW	#lit10,Wn	Wn にリテラルを持って戻る	1	3 (2)	なし
RETURN	RETURN		サブルーチンから戻る	1	3 (2)	なし
RLC	RLC	f	f = f をキャリー経由で左へローテート	1	1	C, N, Z
	RLC	f,WREG	WREG = f をキャリー経由で左へローテート	1	1	C, N, Z
	RLC	Ws,Wd	Wd = Ws をキャリー経由で左へローテート	1	1	C, N, Z
RLNC	RLNC	f	f = f を左へローテート (キャリーなし)	1	1	N, Z
	RLNC	f,WREG	WREG = f を左へローテート (キャリーなし)	1	1	N, Z
	RLNC	Ws,Wd	Wd = Ws を左へローテート(キャリーなし)	1	1	N, Z
RRC	RRC	f	f=fをキャリー経由で右へローテート	1	1	C, N, Z
	RRC	f,WREG	WREG = f をキャリー経由で右へローテート	1	1	C, N, Z
	RRC	Ws,Wd	Wd = Ws をキャリー経由で右へローテート	1	1	C, N, Z
RRNC	RRNC	f	f = f を右へローテート (キャリーなし)	1	1	N, Z
	RRNC	f,WREG	WREG = f を右へローテート (キャリーなし)	1	1	N, Z
	RRNC	Ws,Wd	Wd = Ws を右へローテート (キャリーなし)	1	1	N, Z
SE	SE	Ws, Wnd	Wnd = Ws を符号拡張	1	1	C, N, Z
SETM	SETM	f	f = FFFFh	1	1	なし
	SETM	WREG	WREG = FFFFh	1	1	なし
	SETM	Ws	Ws = FFFFh	1	1	なし
SL	SL	f	f = f を左シフト	1	1	C, N, OV, Z
	SL	f,WREG	WREG = f を左シフト	1	1	C, N, OV, Z
	SL	Ws,Wd	Wd = Ws を左シフト	1	1	C, N, OV, Z
	SL	Wb,Wns,Wnd	Wnd = Wns 分だけ Wb を左シフト	1	1	N, Z
	SL	Wb,#lit5,Wnd	Wnd = lit5 分だけ Wb を左シフト	1	1	N, Z
SUB	SUB	f	f = f – WREG	1	1	C, DC, N, OV, Z
	SUB	f,WREG	WREG = f – WREG	1	1	C, DC, N, OV, Z
	SUB	#lit10,Wn	Wn = Wn – lit10	1	1	C, DC, N, OV, Z
	SUB	Wb,Ws,Wd	Wd = Wb – Ws	1	1	C, DC, N, OV, Z
	SUB	Wb,#lit5,Wd	Wd = Wb – lit5	1	1	C, DC, N, OV, Z
SUBB	SUBB	f	$f = f - WREG - (\overline{C})$	1	1	C, DC, N, OV, Z
	SUBB	f,WREG	WREG = $f - WREG - (\overline{C})$	1	1	C, DC, N, OV, Z
	SUBB	#lit10,Wn	$Wn = Wn - lit10 - (\overline{C})$	1	1	C, DC, N, OV, Z
	SUBB		$Wd = Wb - Ws - (\overline{C})$	1	1	C, DC, N, OV, Z
		Wb, Ws, Wd				
GHDD	SUBB	Wb,#lit5,Wd	Wd = Wb - Iit5 - (C) $f = WREG - f$	1	1	C, DC, N, OV, Z
SUBR	SUBR	f wppg		1	1	C, DC, N, OV, Z
	SUBR	f,WREG	WREG = WREG - f	1	1	C, DC, N, OV, Z
	SUBR	Wb, Ws, Wd	Wd = Ws – Wb	1	1	C, DC, N, OV, Z
	SUBR	Wb,#lit5,Wd	Wd = lit5 – Wb	1	1	C, DC, N, OV, Z
SUBBR	SUBBR	f	$f = WREG - f - (\overline{C})$	1	1	C, DC, N, OV, Z
	SUBBR	f,WREG	WREG = WREG - f - (C)	1	1	C, DC, N, OV, Z
	SUBBR	Wb,Ws,Wd	$Wd = Ws - Wb - (\overline{C})$	1	1	C, DC, N, OV, Z
	SUBBR	Wb,#lit5,Wd	$Wd = lit5 - Wb - (\overline{C})$	1	1	C, DC, N, OV, Z
SWAP	SWAP.b	Wn	Wn = Wn ニブル入れ換え	1	1	なし
	SWAP	Wn	Wn = Wn バイト入れ換え	1	1	なし
TBLRDH	TBLRDH	Ws,Wd	Prog<23:16> を Wd<7:0> へ読み出し	1	2	なし

アセンブリ ニーモニック		アセンブリ構文	内容	ワード 数	サイクル 数	影響を受ける ステータスフラグ
TBLRDL	TBLRDL	Ws,Wd	Prog<15:0> を Wd へ読み出し	1	2	なし
TBLWTH	TBLWTH	Ws,Wd	Ws<7:0> を Prog<23:16> へ書き込み	1	2	なし
TBLWTL	TBLWTL	Ws,Wd	Ws を Prog<15:0> へ書き込み	1	2	なし
ULNK	ULNK		フレームポインタをリンク解除	1	1	なし
XOR	XOR	f	f = f .XOR.WREG	1	1	N, Z
	XOR	f,WREG	WREG = f .XOR.WREG	1	1	N, Z
	XOR	#lit10,Wn	Wd = lit10 .XOR.Wd	1	1	N, Z
	XOR	Wb,Ws,Wd	Wd = Wb .XOR.Ws	1	1	N, Z
	XOR	Wb,#lit5,Wd	Wd = Wb .XOR. lit5	1	1	N, Z
ZE	ZE	Ws, Wnd	Wnd = Ws をゼロ拡張	1	1	C, Z, N

NOTE:

29.0 電気的特性

本セクションには、PIC24FV32KA304 ファミリの電気的特性の概要を記載しています。今後新たに追加される情報は、本書の改訂版に記載します。

以下には、PIC24FV32KA304ファミリの絶対最大定格の一覧を掲載します。長期間にわたる最大定格条件での動作や保管は、デバイスの信頼性に影響する可能性があります。本書の動作表に示す条件または上記から外れた条件でのデバイスの運用は想定していません。

絶対最大定格^(†)

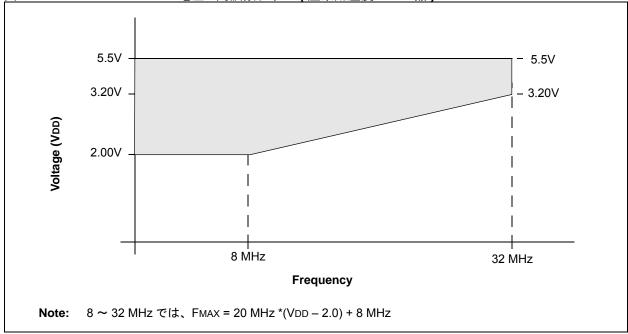
バイアス下での周囲温度	40 ∼ +125 °C
保管温度	65 ~ +150 °C
VDD 電圧 (PIC24FVXXKA30X); Vss 基準	0.3 ~ +6.5 V
VDD 電圧 (PIC24FXXKA30X); Vss 基準	0.3 ~ +4.5 V
アナログ / デジタル兼用ピンの電圧 ; Vss 基準	0.3 V ~ (VDD + 0.3 V)
デジタル専用ピンの電圧 ; Vss 基準	0.3 V ~ (VDD + 0.3 V)
MCLR/VPP ピンの電圧 ; Vss 基準	0.3 ~ +9.0 ∨
Vss ピンの最大出力電流	300 mA
VDD ピンの最大入力電流 ⁽¹⁾	
I/O ピンの出力最大シンク電流	25 mA
I/O ピンの出力最大ソース電流	25 mA
全てのポートの最大シンク電流	200 mA
全てのポートの最大ソース電流(1)	200 mA

Note 1: 最大許容電流は、デバイスの消費電力によって決まります (表 29-1 参照)。

NOTE: 上記の「絶対最大定格」を超える条件は、デバイスに恒久的な損傷を生じる可能性があります。これはストレス定格です。本仕様書の動作表に示す条件または上記から外れた条件でのデバイスの運用は想定していません。長期間にわたる最大定格条件での動作や保管は、デバイスの信頼性に影響する可能性があります。

29.1 DC 特性







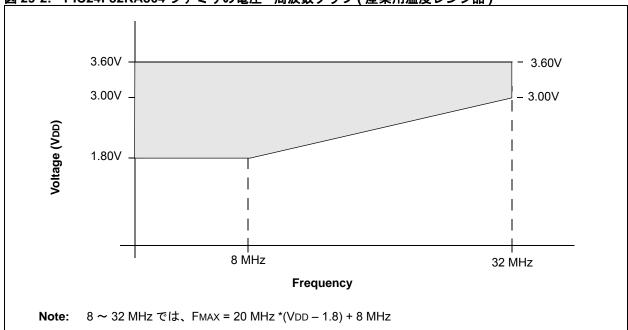


表 29-1: 動作温度条件

定格	記号	Min	Тур	Max	単位
動作時接合部温度レンジ	TJ	-40	_	+125	°C
動作時周囲温度レンジ	TA	-40	_	+85	°C
電力損失: チップ内部の電力損失: PINT = VDD x (IDD – Σ IOH) I/O ピンの電力損失: PI/O = Σ ({VDD – VOH} x IOH) + Σ (VOL x IOL)	Pb	I	PINT + PI/0)	W
最大許容電力損失	PDMAX	(TJ – TA)/θ.	IA	W

表 29-2: パッケージ熱特性

特性	記号	Тур	Max	単位	Note
パッケージ熱抵抗、20 ピン SPDIP	θЈА	62.4	_	°C /W	1
パッケージ熱抵抗、28 ピン SPDIP	θЈА	60	_	°C /W	1
パッケージ熱抵抗、20 ピン SSOP	θЈА	108	_	°C /W	1
パッケージ熱抵抗、28 ピン SSOP	θЈА	71	_	°C /W	1
パッケージ熱抵抗、20 ピン SOIC	θЈА	75	_	°C /W	1
パッケージ熱抵抗、28 ピン SOIC	θЈА	80.2	_	°C /W	1
パッケージ熱抵抗、20 ピン QFN	θЈА	43	_	°C /W	1
パッケージ熱抵抗、28 ピン QFN	θЈА	32	_	°C /W	1
パッケージ熱抵抗、44 ピン QFN	θЈА	29	_	°C /W	1
パッケージ熱抵抗、48 ピン UQFN	θЈА	_	_	°C /W	1

Note 1: 接合部~周囲の熱抵抗である 0JA は、パッケージのシミュレーションで求めた数値です。

表 29-3: DC 特性: 温度/電圧仕様

DC 特性	<u>±</u>		標準動作条件: 1.8 ~ 3.6 V: PIC24F32KA 2.0 ~ 5.5 V: PIC24FV32K 動作温度:-40 ℃ ≤ TA ≤ +85 ℃ (産業用温度レンジ品)					
パラ メータ 番号	記号	特性	Min	Typ ⁽¹⁾	Max	単位	条件	
DC10	VDD	電源電圧	1.8	_	3.6	V	「F」デバイスの場合	
			2.0	_	5.5 V	V	「FV」デバイスの場合	
DC12	VDR	RAM データ保持電圧 ⁽²⁾	1.5	_	_	V		
DC16	VPOR	VDD 起動電圧 内部パワーオン リセット信号を保証する電圧	Vss	_	0.7	V		
DC17	SVDD	VDD 立ち上がり速度 内部パワーオンリセット 信号を保証する立ち上がり	0.05	_	_	V/ms	0.1 s 以内に 0 - 3.3 V 60 ms 以内に 0 - 2.5 V	

Note 1: 「Typ」列のデータは、特に明記のない限り 3.3 V、25 ℃での値です。各パラメータ値はあくまでも設計 指標値であり、テストで確認した値ではありません。

2: この限界値よりも VDD が低下すると RAM のデータが失われます。

表 29-4: HIGH/LOW 電圧検出の特性

標準動作条件: 1.8 ~ 3.6 V: PIC24F32KA3XX

2.0 ~ 5.5 V: PIC24FV32KA3XX

動作温度 : -40 °C ≤ TA ≤ +85 °C (産業用温度レンジ品)

パラ メータ 番号	記号		特性		Тур	Max	単位	条件
DC18	VHLVD	VDD 推移時の	HLVDL<3:0> = 0000 ⁽²⁾	1		1.90	٧	
		HLVD 電圧	HLVDL<3:0> = 0001	1.88	_	2.13	V	
			HLVDL<3:0> = 0010	2.09	_	2.35	٧	
			HLVDL<3:0> = 0011	2.25	_	2.53	٧	
			HLVDL<3:0> = 0100	2.35	_	2.62	V	
			HLVDL<3:0> = 0101	2.55	_	2.84	V	
			HLVDL<3:0> = 0110	2.80	_	3.10	V	
			HLVDL<3:0> = 0111	2.95	_	3.25	٧	
			HLVDL<3:0> = 1000	3.09	_	3.41	V	
			HLVDL<3:0> = 1001	3.27	_	3.59	V	
			HLVDL<3:0> = 1010 ⁽¹⁾	3.46	_	3.79	V	
			HLVDL<3:0> = 1011 ⁽¹⁾	3.62	_	4.01	V	
			HLVDL<3:0> = 1100 ⁽¹⁾	3.91		4.26	V	
			HLVDL<3:0> = 1101 ⁽¹⁾	4.18		4.55	V	
			HLVDL<3:0> = 1110 ⁽¹⁾	4.49		4.87	V	

Note 1: これらのトリップポイントを PIC24F32KA304 デバイスで使用しないでください。

2: このトリップポイントを PIC24FVXXKA30X デバイスで使用しないでください。

<u>表 29-5: BOR トリ</u>ップポイント

標準動作条件: 1.8 ~ 3.6 V: PIC24F32KA3XX

2.0 ~ 5.5 V: PIC24FV32KA3XX

動作温度:-40 °C ≤ TA ≤ +85 °C (産業用温度レンジ品)

パラ メータ 番号	記号	特性		Min	Тур	Max	単位	条件
DC19		VDD 低下検出時の BOR しきい値電圧	BORV = 00	_	_			LPBOR と DSBOR に対して有効、 Note 1
			BORV = 01	2.90	3	3.38	V	
			BORV = 10	2.53	2.7	3.07	V	
			BORV = 11	1.75	1.85	2.05	٧	Note 2
			BORV = 11	1.95	2.05	2.16	V	Note 3

Note 1: LPBOR は POR 回路をリアームしますが、BOR を生成しません。

2: PIC24F (3.3 V) デバイスで有効

3: PIC24FV (5 V) デバイスで有効

表 29-6: DC 特性: 動作電流 (IDD)

<u>表 29-6: D</u> DC 特性	C 特性:劉作電流 (ID	標準動作条件動作温度:-4	KA3XX PKA3XX				
パラメータ No.	デバイス	Typical	Max	単位		条件	
IDD 電流							
DC20			_		-40 °C		
DC20a		269.00	_	μA	+25 ℃	2.0 V	
DC20b		209.00	_	μΛ	+60 °C	2.0 V	
DC20c	PIC24FV32KA3XX		450.00		+85 ℃		
DC20d	FIG24FV3ZNASAX		_		-40 ℃		
DC20e		465.00	_	μA	+25 ℃	5.0 V	
DC20f		405.00	_	μΑ	+60 °C	3.0 V	
DC20g			830.00		+85 ℃		0.5 MIPS、
DC20h			_		-40 ℃		FOSC = 1 MHz
DC20i		200.00	_		+25 ℃	1.8 V	
DC20j		200.00	μΑ		+60 °C	1.0 V	
DC20k	PIC24F32KA3XX	330.00			+85 °C		
DC20I	FIG24F32NA3AA		_		-40 °C		
DC20m		410.00	_	μA	+25 ℃	3.3 V	
DC20n		410.00	_	μΛ	+60 °C	3.3 V	
DC20o			750.00		+85 °C		
DC22			_		-40 ℃		
DC22a		490.00	_	μA	+25 ℃	2.0 V	
DC22b		490.00	_	μΑ	+60 °C	2.0 V	
DC22c	PIC24FV32KA3XX		_		+85 ℃		
DC22d	FIG24FV3ZNASAX		_		-40 °C		
DC22e		880.00	_	μA	+25 ℃	5.0 V	
DC22f		880.00	_	μΑ	+60 °C	3.0 V	
DC22g			_		+85 ℃		1 MIPS、
DC22h			_		-40 °C		FOSC = 2 MHz
DC22i		407.00	_	μA	+25 ℃	1.8 V	
DC22j		407.00		μΑ	+60 °C	1.0 V	
DC22k	PIC24F32KA3XX				+85 °C		
DC22I	FIGZ4F3ZNA3AX				-40 °C		
DC22m		800.00		^	+25 ℃	3.3 V	
DC22n		000.00	_	μA	+60 °C	3.3 V	
DC22o			_		+85 °C		

表 29-6: DC 特性:動作電流 (IDD) (続き) 標準動作条件: 1.8 ~ 3.6 V: PIC24F32KA3XX DC 特性 動作温度: -40 °C ≤ TA ≤ +85 °C (産業用温度レンジ品)

DC 特性		2.0 ~ 5.5 V: PIC24FV32KA3XX 動作温度:-40 °C ≤ TA ≤ +85 °C (産業用温度レンジ品)						
パラメータ No.	デバイス	Typical	Max	単位		条件		
IDD 電流 (続き	<u>*</u>)	_						
DC24	_		_		-40 °C			
DC24a	PIC24FV32KA3XX	13.00	_	mA	+25 ℃	5.0 V		
DC24b	1110241 1021040701	13.00	_	ША	+60 °C	3.0 V		
DC24c			20.00		+85 ℃		16 MIPS、	
DC24d			_		-40 °C		FOSC = 32 MHz	
DC24e	PIC24F32KA3XX	12.00		mA	+25 ℃	3.3 V		
DC24f	1 10241 321043///	12.00	_	- MA	+60 °C	0.0 V		
DC24g			18.00		+85 °C			
DC26			_		-40 ℃			
DC26a		2.00	_	mΛ	+25 ℃	2.0 V		
DC26b		2.00 mA		+60 °C	2.0 V			
DC26c	DICO4EVOOKAOVV		_		+85 °C			
DC26d	PIC24FV32KA3XX		_		-40 °C]	
DC26e	-	2.50	_	A	+25 ℃	501/	FRC (4 MIPS), FOSC = 8 MHz	
DC26f		3.50	_	mA	+60 °C	5.0 V		
DC26g			_		+85 °C			
DC26h			_	mA	-40 °C	1.8 V		
DC26i		4.00	_		+25 °C			
DC26j	1	1.80	_		+60 °C			
DC26k	DIOCATOOKA OVY		_	1	+85 °C			
DC26I	PIC24F32KA3XX		_		-40 °C	3.3 V		
DC26m		0.40	_	mA	+25 °C			
DC26n		3.40	_		+60 °C			
DC26o	1		_	-	+85 °C			
DC30			_		-40 ℃			
DC30a	-		_		+25 °C	1		
DC30b		48.00	_	μA	+60 °C	2.0 V		
DC30c			250.00	-	+85 °C	_		
DC30d	PIC24FV32KA3XX		_		-40 °C			
DC30e			_		+25 °C	1		
DC30f		75.00	_	μA	+60 °C	5.0 V		
DC30g			275.00		+85 °C		LPRC	
DC30h			_		-40 °C		(15.5 KIPS), FOSC = 31 kHz	
DC30i	1	0.10	_	1 .	+25 °C	1	FUSU - 31 KMZ	
DC30j	1	8.10	_	μA	+60 °C	1.8 V		
DC30k	DIO04500111		28.00	1	+85 °C	1		
DC30I	PIC24F32KA3XX		_		-40 °C			
DC30m	1	-	_	_	+25 °C	1		
DC30n	1	13.50	_	μA	+60 °C	3.3 V		
DC30o	1		55.00	1	+85 °C	1		
		1	55.00	1		l	ı	

表 29-7:	DC 特性 : アイドル	電流 (l idle)									
		標準動作領	€件:		1.8 ~ 3.6 V: F	PIC24F32KA	зхх				
	DC 特性	* 11-10-t	2.0 ~ 5.5 V: PIC24FV32KA3XX 動作温度 : -40 ℃ ≤ TA ≤ +85 ℃ (産業用温度レンジ品)								
	1	動作温度:	-40 °C ≤ TA :	≤ +85 ℃(產 T	€業用温度レンシ 	〉品)					
パラメータ No.	デバイス	Typical	Max	単位		条件					
アイドル電流	荒 (IIDLE)	•		•							
DC40			_		-40 °C						
DC40a	1	100.00			+25 °C	201/					
DC40b		120.00	_	μΑ	+60 °C	2.0 V					
DC40c	DICOAEVOOKA OVV		200.00		+85 °C						
DC40d	PIC24FV32KA3XX		<u>—</u>		-40 °C						
DC40e	1	160.00	_		+25 °C	501/					
DC40f	1	160.00		μA	+60 °C	5.0 V					
DC40g	1		430.00		+85 °C		0.5 MIPS				
DC40h			_		-40 °C		FOSC = 1 MHz				
DC40i		50.00	_		+25 °C	101/					
DC40j		50.00	_	μΑ	+60 °C	1.8 V					
DC40k	PIC24F32KA3XX		100.00		+85 °C						
DC40I	PIC24F32NA3AA		_		-40 °C						
DC40m		00.00			+25 °C	3.3 V					
DC40n		90.00 µA		+60 °C	3.3 V						
DC40o			370.00		+85 ℃						
DC42					-40 °C						
DC42a		165.00	_	μA	+25 ℃	2.0 V					
DC42b		105.00	165.00	_	μΛ	+60 °C	2.0 V				
DC42c	PIC24FV32KA3XX		_		+85 °C						
DC42d	1 10241 732104377		_		-40 °C						
DC42e		260.00	_	μΑ	+25 ℃	5.0 V					
DC42f		200.00	_	μΛ	+60 °C	3.0 V					
DC42g			_		+85 ℃		1 MIPS、				
DC42h					-40 °C		FOSC = 2 MHz				
DC42i		95.00	_	μΑ	+25 ℃	1.8 V					
DC42j		33.00	_	μΛ	+60 °C	1.0 V					
DC42k	PIC24F32KA3XX		_		+85 ℃						
DC42I	1 10241 0210 10701		_		-40 °C						
DC42m		180.00	_	μΑ	+25 ℃	3.3 V					
DC42n		130.00	_	μ, ,	+60 °C	J.5 V					
DC42o			_		+85 °C						
DC44			_		-40 °C						
DC44a	PIC24FV32KA3XX	3.10	_	mA	+25 °C	5.0 V					
DC44b		0.10	_	, \	+60 °C						
DC44c			6.50		+85 °C		16 MIPS、				
DC44d			_		-40 °C		FOSC = 32 MHz				
DC44e	PIC24F32KA3XX	2.90	_	mA	+25 °C	3.3 V					
DC44f	. 102 11 021010701	2.50	_	,	+60 °C] 3.5 v					
DC44g			6.00		+85 °C						

表 29-7: DC 特性: アイドル電流 (lidle) (続き)

表 29-7: L	ひ 特性:アイトル)	標準動作条			1.8 ~ 3.6 V: F	PIC24F32KA3	BXX					
	DC 特性		2.0 ~ 5.5 V: PIC24FV32KA3XX									
		動作温度:	-40 °C ≤ TA ≤	≤ +85 °C(產	産業用温度レンシ	シ品)						
パラメータ No.	デバイス	Typical	Max	単位		条件						
アイドル電流	た(liDLE)(続き)											
DC46			ı		-40 °C							
DC46a		0.65	_	mA	+25 ℃	2.0 V						
DC46b		0.05		IIIA	+60 °C	2.0 V						
DC46c	PIC24FV32KA3XX		ı		+85 °C							
DC46d	FICZ4FVJZNAJAA		ı		-40 °C							
DC46e		1.00 — mA		mA	+25 °C	5.0 V						
DC46f		1.00	_	ША	+60 °C	J.0 V						
DC46g			ı		+85 °C		FRC (4 MIPS)、					
DC46h					-40 °C		FOSC = 8 MHz					
DC46i		0.55		mA	+25 ℃	1.8 V						
DC46j		0.55	_	ША	+60 °C	1.0 V						
DC46k	PIC24F32KA3XX	_		+85 °C								
DC46I	1 10241 321043700			mA	-40 °C	3.3 V						
DC46m		1.00	1		+25 ℃							
DC46n		1.00	_	IIIA	+60 °C							
DC46o					+85 °C							
DC50			-		-40 °C							
DC50a		60.00	_	μA	+25 ℃	2.0 V						
DC50b		00.00	-	μΛ	+60 °C	2.0 V						
DC50c	PIC24FV32KA3XX		200.00		+85 °C							
DC50d	110241 1021140/01		_		-40 °C							
DC50e		70.00	_	μA	+25 ℃	5.0 V						
DC50f		70.00	_	μΛ	+60 °C	3.0 V	1,000					
DC50g			225.00		+85 °C		LPRC (15.5 KIPS)、					
DC50h			_		-40 °C		FOSC = 31 kHz					
DC50i		2.20	_	μA	+25 ℃	1.8 V						
DC50j				μ/	+60 °C	1.0 V						
DC50k	PIC24F32KA3XX		18.00		+85 °C]					
DC50I	1 10241 02104000				-40 ℃							
DC50m		4.00	_	μA	+25 °C	3.3 V						
DC50n		7.00	_	μ	+60 °C	0.5 V						
DC50o			40.00		+85 °C							

DC 特性: 省電力時の電流 (IPD) 表 29-8:

表 29-8: DC 特性	DC 特性:省電力時	標準動作象	·件:		.0 ~ 5.5 V	: PIC24F32KA3 : PIC24FV32KA ンジ品)		
パラメータ No.	デバイス	Typical ⁽¹⁾	Max	単位	条件			
省電力時の	電流 (IPD)							
DC60			_		-40 °C			
DC60a		6.00	8.00	пΛ	+25 ℃	2.0 V		
DC60b		6.00 HA		+60 °C	2.0 V			
DC60c	PIC24FV32KA3XX		9.00		+85 °C			
DC60d	1 10241 10211/10/07		_		-40 ℃			
DC60e		6.00	8.00	μA	+25 ℃	5.0 V		
DC60f		0.00	9.00	μΑ	+60 °C	5.0 V		
DC60g			10.00		+85 ℃		スリープモード ⁽²⁾	
DC60h					-40 ℃	1.8 V		
DC60i		0.025	0.80	μA	+25 °C			
DC60j		0.025	1.50	μΛ	+60 °C	1.0 V		
DC60k	PIC24F32KA3XX		2.00		+85 °C			
DC60I	1 102 11 0210 1070		_		-40 °C			
DC60m		0.040	1.00	μA	+25 ℃	3.3 V		
DC60n		0.040	2.00	μΛ	+60 °C	0.0 1		
DC60o			3.00		+85 °C			
DC61			_		-40 °C			
DC61a		0.25	_	μA	+25 ℃	2.0 V		
DC61b		0.25		μΛ	+60 °C	2.0 V		
DC61c	PIC24FV32KA3XX		_		+85 ℃		低電圧	
DC61d	1.02 11 00210 0000		_		-40 °C		スリープモード ⁽²⁾	
DC61e		0.35	_		+25 ℃			
DC61f		0.55	_	μA	+60 °C	0.0 V		
DC61g			3.00		+85 °C		ļ	

- Note 1: 「Typical」列の値は、特に明記のない限り、3.3 V/25 ℃ (PIC24F32KA3XX); 5.0 V/25 ℃ (PIC24FV32KA3XX) での値です。各パラメータ値はあくまでも設計指標値であり、テストで確認した値ではありません。
 2: ベース IPD は、全ての周辺モジュールとクロックを停止させた時の計測値です(全ての I/O を出力として設定して LOW に設定、PMSLP を「0」に設定、WDT 等は全て OFF)。
 3: △電流はモジュールを有効化した時に消費される電流増加量です。この電流をベース IPD 電流に加算する必要がある。

 - 要があります。

 - 4: この電流はスリープ時にのみ適用されます。5: この電流はスリープおよびディープスリープ時に適用されます。
 - 6: この電流はディープスリープ時にのみ適用されます。

表 29-8: DC 特性 : 省電力時の電流 (IPD) (続き)

DC 特性		標準動作第		1.8 ~ 3.6 V: PIC24F32KA3XX 2.0 ~ 5.5 V: PIC24FV32KA3XX ≦ TA ≤ +85 ℃ (産業用温度レンジ品)					
パラメータ No.	デバイス	Typical ⁽¹⁾	Max	単位	条件				
省電力時の	電流 (IPD) (続き)								
DC70			-		-40 ℃				
DC70a		0.03	_	пΔ	+25 ℃	2.0 V			
DC70b	0.03 μΑ			+60 °C	2.0 V				
DC70c	PIC24FV32KA3XX				+85 °C				
DC70d	1 10241 V32NA3AA		_	μA	-40 °C	5.0 V			
DC70e		0.10	_		+25 ℃				
DC70f		0.10	_		+60 ℃				
DC70g			1.20		+85 ℃		ディープスリープ		
DC70h			_		-40 °C		モード		
DC70i		0.02		μΑ	+25 ℃	1.8 V			
DC70j		0.02		μΑ	+60 °C	1.0 V			
DC70k	PIC24F32KA3XX				+85 °C				
DC70I	1 10241 0211/40///				-40 °C				
DC70m		0.08		пΔ	+25 ℃	33 \/			
DC70n		0.08	_	μΑ	+60 °C	3.3 V			
DC70o			1.20		+85 °C				

- 「Typical」列の値は、特に明記のない限り、3.3 V/25 °C (PIC24F32KA3XX); 5.0 V/25 °C (PIC24FV32KA3XX) Note 1: での値です。各パラメータ値はあくまでも設計指標値であり、テストで確認した値ではありません。
 - ベース IPD は、全ての周辺モジュールとクロックを停止させた時の計測値です (全ての I/O を出力として設 定して LOW に設定、PMSLP を「0」に設定、WDT 等は全て OFF)。
 - 3: △電流はモジュールを有効化した時に消費される電流増加量です。 この電流をベース IPD 電流に加算する必 要があります。
 - 4:
 - この電流はスリープ時にのみ適用されます。 この電流はスリープおよびディープスリープ時に適用されます。 この電流はディープスリープ時にのみ適用されます。

表 29-8:	DC 特性: 省電力時	持の電流 (Ⅱ	PD)(続き)						
		標準動作条		1	.8 ~ 3.6 V	: PIC24F32KA3	XX		
DC 特性				2	2.0 ~ 5.5 V	: PIC24FV32KA	3XX		
	·	動作温度:	-40 °C ≤ TA ≤	:+85 ℃ (産業	美用温度レ	ンジ品)			
パラメータ No.	デバイス	Typical ⁽¹⁾	Max	単位		条件			
	<u> </u> 電流 (IPD) (続き)				<u> </u>				
DC71					-40 °C				
DC71a					+25 °C				
DC71b		0.50	_	μΑ	+60 °C	2.0 V			
DC71c			_		+85 °C				
DC71d	PIC24FV32KA3XX				-40 °C				
DC71e			_	μΑ	+25 °C				
DC71f		0.70	_		+60 °C	5.0 V			
DC71g	1.5 +85 °C			ウォッチドッグ					
DC71h			_		-40 °C		タイマ 電流:		
DC71i			_	_	+25 °C	-	∆WDT ^(3,4)		
DC71j		0.50	_	μΑ	+60 °C	1.8 V			
DC71k	PIOO (FOOL(AD)()		_		+85 °C				
DC71I	PIC24F32KA3XX		_		-40 °C				
DC71m		0.70	_	πΔ	+25 °C	221			
DC71n	0.		0.70	_	μΑ	+60 °C	3.3 V		
DC71o			1.5	-	+85 °C				
DC72			_		-40 ℃				
DC72a		0.00	_		+25 ℃	201/			
DC72b		0.80	_	μΑ	+60 °C	2.0 V			
DC72c	PIC24FV32KA3XX		_		+85 °C				
DC72d	PICZ4FV3ZNASAA		_		-40 °C				
DC72e		1.50	_	μA	+25 ℃	5.0 V			
DC72f		1.50	_	μΑ	+60 °C	5.0 V	32 kHz 水晶振動子		
DC72g			2.0		+85 °C		(RTCC、DSWDT、 Timer1 で使用):		
DC72h			_		-40 ℃		ΔSOSC;		
DC72i		0.70	_	μΑ	+25 ℃	1.8 V	(SOSCSEL = 0) ^(3,5)		
DC72j		0.70	_	μΛ	+60 °C	1.0 V	,		
DC72k	PIC24F32KA3XX		_		+85 °C				
DC72I	1 10241 321/43//		_		-40 ℃				
DC72m		1.00		μA	+25 ℃	331/			
DC72n]	1.00	_	μΑ	+60 °C	 3.3 ∨	
DC72o			1.5		+85 °C				

凡例: 網掛けしていない行は PIC24F32KA3XX デバイス用、網掛けした行は PIC24FV32KA3XX デバイス用です。

「Typical」列の値は、特に明記のない限り、3.3 V/25 °C (PIC24F32KA3XX); 5.0 V/25 °C (PIC24FV32KA3XX) Note 1: での値です。各パラメータ値はあくまでも設計指標値であり、テストで確認した値ではありません。

- ベース IPD は、全ての周辺モジュールとクロックを停止させた時の計測値です (全ての I/O を出力として設 定して LOW に設定、PMSLP を「0」に設定、WDT 等は全て OFF)。
- 3: Δ電流はモジュールを有効化した時に消費される電流増加量です。この電流をベース IPD 電流に加算する必 要があります。
- 4:
- この電流はスリープ時にのみ適用されます。 この電流はスリープおよびディープスリープ時に適用されます。 この電流はディープスリープ時にのみ適用されます。

DC 特性: 省電力時の電流 (IPD) (続き) 標準動作条件: 1.8 ~ 3.6 V: PIC24F32KA3XX DC 特性 2.0 ~ 5.5 V: PIC24FV32KA3XX 動作温度:-40 °C ≤ TA ≤ +85 °C (産業用温度レンジ品) パラメータ デバイス Typical⁽¹⁾ Max 単位 条件 No. 省電力時の電流 (IPD) (続き) DC75 -40 °C DC75a +25 °C 5.40 μΑ 2.0 V DC75b +60 °C +85 °C DC75c PIC24FV32KA3XX DC75d -40 °C DC75e +25 °C 8.10 μΑ 5.0 V DC75f +60 °C DC75g 14.00 +85 °C ∆HLVD^(3,4) DC75h -40 °C DC75i +25 °C 4.90 μΑ 1.8 V DC75j +60 °C DC75k +85 °C PIC24F32KA3XX DC75I -40 °C DC75m +25 °C 7.50 μΑ 3.3 V DC75n +60 °C DC75o 14.00 +85 °C DC76 -40 °C DC76a +25 °C 5.60 2.0 V μΑ DC76b +60 °C DC76c +85 °C PIC24FV32KA3XX DC76d -40 °C DC76e +25 °C 6.50 μΑ 5.0 V DC76f +60 °C DC76q 11.20 +85 °C ∆BOR^(3,4) DC76h -40 °C DC76i +25 °C 5.60 μA 1.8 V DC76i +60 °C DC76k +85 °C PIC24F32KA3XX DC76I -40 °C DC76m +25 °C 6.00 3.3 V μΑ

凡例:

11.20

- 網掛けしていない行は PIC24F32KA3XX デバイス用、網掛けした行は PIC24FV32KA3XX デバイス用です。「Typical」列の値は、特に明記のない限り、3.3 V/25 °C (PIC24F32KA3XX); 5.0 V/25 °C (PIC24FV32KA3XX) Note 1: での値です。各パラメータ値はあくまでも設計指標値であり、テストで確認した値ではありません。
 - 2: ベース IPD は、全ての周辺モジュールとクロックを停止させた時の計測値です (全ての I/O を出力として設 定して LOW に設定、PMSLP を「0」に設定、WDT 等は全て OFF)。

+60°C

+85 °C

- 3: △電流はモジュールを有効化した時に消費される電流増加量です。この電流をベース IPD 電流に加算する必 要があります。
- この電流はスリープ時にのみ適用されます。 4:
- この電流はスリープおよびディープスリープ時に適用されます。 この電流はディープスリープ時にのみ適用されます。 5:

DC76n

DC76o

表 29-8:	DC 特性: 省電力問	持の電流 (Ⅱ	 PD) (続き)				
		標準動作第	⊱件:			: PIC24F32KA3	
DC 特性		動作温度:	-40 °C ≤ TA ≤			: PIC24FV32KA シジ品)	3XX
パラメータ No.	デバイス	Typical ⁽¹⁾	Max	単位		条件	
省電力時の	電流 (IPD) (続き)						
DC78			_		-40 °C		
DC78a		0.00	_		+25 ℃	2.0 V	
DC78b		0.03	_	μA	+60 °C	2.0 V	
DC78c	DICOAEVOOKAOVV		_		+85 °C		
DC78d	PIC24FV32KA3XX		_		-40 °C		
DC78e		0.05	_		+25 ℃	5 O V	
DC78f		0.05	_	μA	+60 °C	5.0 V	
DC78g			0.20		+85 °C		ΔLPBOR/ ディープ
DC78h			_		-40 °C		スリープ BOR ^(3,5)
DC78i		0.02	_		+25 ℃	4.0.7	
DC78j		0.03 µA —		+60 °C	1.8 V		
DC78k	DICOAFOOKAOVY		_		+85 °C		
DC78I	PIC24F32KA3XX		_	μA	-40 °C	3.3 V	
DC78m		0.05	_		+25 ℃		
DC78n		0.05	_	μΑ	+60 °C	3.3 V	
DC78o			0.20		+85 °C		
DC80			_		-40 °C		
DC80a		0.20	_		+25 ℃	2.0 V	
DC80b		0.20	_	μA	+60 °C	2.0 V	
DC80c	PIC24FV32KA3XX		_		+85 °C		
DC80d	PICZ4FV3ZNASAA		_		-40 °C		
DC80e		0.70	-		+25 ℃	5.0 V	
DC80f		0.70	_	μA	+60 °C	5.0 V	
DC80g			1.5		+85 ℃		ディープスリープ WDT: ∆DSWDT
DC80h			_		-40 °C		(LPRC) ^(3,6)
DC80i		0.20	_		+25 ℃	1.8 V	(=: : : :)
DC80j		0.20 μA		+60 °C	1.0 V		
DC80k	DICOAFOOKAOVV		_		+85 °C		
DC80I	PIC24F32KA3XX		_		-40 ℃		
DC80m		0.25	_		+25 ℃	221/	
DC80n		0.35	_	μΑ	+60 °C	3.3 V	
DC80o			0.8		+85 °C		

凡例: 網掛けしていない行は PIC24F32KA3XX デバイス用、網掛けした行は PIC24FV32KA3XX デバイス用です。 Note 1:

「Typical」列の値は、特に明記のない限り、3.3 V/25 ℃ (PIC24F32KA3XX); 5.0 V/25 ℃ (PIC24FV32KA3XX) での値です。各パラメータ値はあくまでも設計指標値であり、テストで確認した値ではありません。

- ベース IPD は、全ての周辺モジュールとクロックを停止させた時の計測値です (全ての I/O を出力として設 定して LOW に設定、PMSLP を「0」に設定、WDT 等は全て OFF)。
- Δ電流はモジュールを有効化した時に消費される電流増加量です。この電流をベース IPD 電流に加算する必 要があります。
- 4:
- この電流はスリープ時にのみ適用されます。 この電流はスリープおよびディープスリープ時に適用されます。 この電流はディープスリープ時にのみ適用されます。 5:
- 6:

表 29-9: DC 特性: I/O ピン入力仕様

 標準動作条件:
 1.8 ~ 3.6 V: PIC24F32KA3XX

 DC 特性
 2.0 ~ 5.5 V: PIC24FV32KA3XX

動作温度:-40 °C ≤ TA ≤ +85 °C (産業用温度レンジ品)

	動作温度: -40 ℃ ≤ IA ≤ +85 ℃ (産業用温度レンシ品)							
パラ メータ No.	記号	特性	Min	Typ ⁽¹⁾	Max	単位	条件	
	VIL	入力 LOW 電圧 ⁽⁴⁾	_	_	_	_		
DI10		l/O ピン	Vss	_	0.2 VDD	V		
DI15		MCLR	Vss	_	0.2 VDD	V		
DI16		OSCI (XT モード)	Vss	_	0.2 VDD	V		
DI17		OSC1 (HS モード)	Vss	_	0.2 VDD	V		
DI18		I ² C™ バッファ付き I/O ピン	Vss	_	0.3 VDD	V	SMBus 無効時	
DI19		SMBus バッファ付き I/O ピン	Vss	_	0.8	V	SMBus 有効時	
	VIH	入力 HIGH 電圧 ⁽⁴⁾	_	_	_	_		
DI20		I/O ピン : アナログ機能あり デジタル専用	0.8 VDD 0.8 VDD		VDD VDD	V V		
DI25		MCLR	0.8 VDD	_	VDD	V		
DI26		OSCI (XTモード)	0.7 VDD	_	VDD	V		
DI27		OSC1 (HS モード)	0.7 VDD	_	VDD	V		
DI28		I ² C バッファ付き I/O ピン アナログ機能あり デジタル専用	0.7 VDD 0.7 VDD	_ _	VDD VDD	V V		
DI29		SMBus 付き I/O ピン	2.1	_	VDD	V	$2.5V \le VPIN \le VDD$	
DI30	ICNPU	CNx プルアップ電流	50	250	500	μΑ	VDD = 3.3 V, VPIN = VSS	
DI50	IIL	入力リーク電流 ^(2,3) I/O ポート	_	0.05	0.1	μΑ	Vss ≤ Vpin ≤ Vdd 高インピーダンス状態のピン	
DI55		MCLR	_	_	0.1	μΑ	VSS ≤ VPIN ≤ VDD	
DI56		OSCI	_	_	5	μΑ	Vss≤VPIN≤VDD XT および HS モード	

- Note 1: 「Typ」列のデータは、特に明記のない限り3.3 V、25 ℃での値です。各パラメータ値はあくまでも設計指標値であり、テストで確認した値ではありません。
 - 2: MCLR ピンのリーク電流は、印加する電圧レベルによって大きく変化します。仕様値のレベルは、通常の動作条件での値を表します。入力電圧によっては、これより大きなリーク電流が計測される場合があります。
 - 3: 負の電流値は、ピンによるソース電流として定義しています。
 - 4: I/O ピンのバッファタイプについては表 1-3 を参照してください。

表 29-10: DC 特性: I/O ピン出力仕様

DC 特性			標準動作	条件:			1.8 ~ 3.6 V: PIC: 2.0 ~ 5.5 V: PIC:	-
			動作温度	1)				
パラ メータ No.	記号	特性	Min	Typ ⁽¹⁾	Max	単位	ś	条件
	Vol	出力 LOW 電圧	_	_				
DO10		全ての I/O ピン		_	0.4	V	IOL = 8.0 mA	VDD = 4.5 V
			_	_	0.4	V	IOL = 4.0 mA	VDD = 3.6 V
			_	_	0.4	V	IOL = 3.5 mA	VDD = 2.0 V
DO16		OSC2/CLKO	_	_	0.4	V	IOL = 2.0 mA	Vdd = 4.5 V
			_	_	0.4	V	IOL = 1.2 mA	Vdd = 3.6 V
			_	_	0.4	V	IOL = 0.4 mA	V _{DD} = 2.0 V
	VOH	出力 HIGH 電圧						
DO20		全ての I/O ピン	3.8	_	_	V	Iон = -3.5 mA	VDD = 4.5 V
			3	_	_	V	Iон = -3.0 mA	VDD = 3.6 V
			1.6	_	_	V	Iон = -1.0 mA	VDD = 2.0 V
DO26		OSC2/CLKO	3.8	_	_	V	IOH = -2.0 mA	VDD = 4.5 V
			3	_	_	V	IOH = -1.0 mA	VDD = 3.6 V
			1.6	_	_	V	IOH = -0.5 mA	V _{DD} = 2.0 V

Note 1: 「Typ」列のデータは、特に明記のない限り 25 ℃での値です。各パラメータ値はあくまでも設計指標値であり、テストで確認した値ではありません。

表 29-11: DC 特性: プログラム メモリ

DC 特性		WIE. ZEZZZE	標準動作第	標準動作条件: 1.8 ~ 3.6 V: PIC24F32KA3X 2.0 ~ 5.5 V: PIC24FV32KA3 動作温度:-40 ℃ ≤ TA ≤ +85 ℃(産業用温度レンジ品)				
パラ メータ No.	記 号	特性	Min	Typ ⁽¹⁾	Max	単位	条件	
		プログラム フラッシュメ モリ						
D130	ЕР	セル書き込み耐性	10,000 ⁽²⁾	_	_	E/W		
D131	VPR	読み出し用 Vdd	VMIN	_	3.6	V	Vmin = 最小動作電圧	
D133A	Tıw	自己処理書き込みサイク ル時間	_	2	_	ms		
D134	TRETD	特性保持期間	40	_	_	年	他の仕様値に違反していない場合	
D135	IDDP	プログラミング中の消費 電流	_	10		mA		

Note 1: 「Typ」列のデータは、特に明記のない限り3.3 V、25 ℃での値です。

2: 自己書き込みとブロック消去

表 29-12: DC 特性: データ EEPROM メモリ

DC 特性			標準動作条件: 1.8 ~ 3.6 V: PIC24F32KA3 2.0 ~ 5.5 V: PIC24FV32KA 動作温度: -40 ℃ ≤ TA ≤ +85 ℃ (産業用温度レンジ品)				
パラ メータ No.	記号	特性	Min	Typ ⁽¹⁾	Max	単位	条件
		データ EEPROM メモリ					
D140	EPD	セル書き込み耐性	100,000	_	_	E/W	
D141	VPRD	読み出し用 Vdd	VMIN	_	3.6	V	Vmin = 最小動作電圧
D143A	Tiwd	自己処理書き込みサイク ル時間	_	4	_	ms	
D143B	TREF	リフレッシュ前の書き込 み / 消去サイクル総数	_	10M	_	E/W	
D144	TRETDD	特性保持期間	40	_	_	年	他の仕様値に違反していない場合
D145	IDDPD	プログラミング山の沿弗	_	7		mΑ	

Note 1: 「Typ」列のデータは、特に明記のない限り3.3 V、25 ℃での値です。

表 29-13: コンパレータの DC 仕様

動作条件	動作条件:2.0 V < VDD < 3.6 V、-40 ℃ < TA < +85 ℃ (特に明記のない場合)									
パラ メータ No.	メータ 記号 特性 Min Typ Max 単位 注釈									
D300	VIOFF	入力オフセット電圧 *		20	40	mV				
D301	VICM	入力コモンモード電圧 *	0		Vdd	V				
D302	CMRR	コモンモード除去率 *	55		_	dB				

^{*} これらのパラメータは特性データであり、テストはしていません。

表 29-14: コンパレータ参照電圧の DC 仕様

動作条件:	動作条件: 2.0 V < VDD < 3.6 V、-40 ℃ < TA < +85 ℃ (特に明記のない場合)									
パラメー タ No.	記号	特性	Min	Тур	Max	単位	注釈			
VRD310	CVRES	分解能	_	_	Vdd/32	LSb				
VRD311	CVRAA	絶対精度	_	_	AVdd – 1.5	LSb				
VRD312	CVRur	ユニット抵抗値 (R)	_	2k	_	Ω				

表 29-15: 内部電圧レギュレータの仕様

動作条件	動作条件 :-40 ℃ < Ta < +85 ℃ (特に明記のない場合)										
パラ メータ No.	記号	特性	Min	Тур	Max	単位	注釈				
	Vbg	バンドギャップ リファレンス 電圧	0.973	1.024	1.075	>					
	TBG	バンドギャップ リファレンス 起動時間		1	_	ms					
	Vrgout	レギュレータ出力電圧	3.1	3.3	3.6	>					
	Cefc	外部フィルタ コンデンサ値	4.7	10	_	μF	直列抵抗として 3 Ω 未満を 推奨、5 Ω 未満を要求				
	VLVR	低電圧レギュレータ出力電圧	_	2.6	_	V					

表 29-16: CTMU 電流ソースの仕様

DC 特性	ŧ			作条件 l度:-40		A ≤ +85 °	1.8 ~ 3.6 V: PIC24F32KA3XX 2.0 ~ 5.5 V: PIC24FV32KA3XX ℃ (産業用温度レンジ品)			
パラ メータ No.	記号	特性	Min Typ ⁽¹⁾ Ma		Max	単位	注釈	条件		
	Iout1	CTMU 電流ソース、 ベースレンジ	_	550	_	nA	CTMUICON<1:0> = 00			
	IOUT2	CTMU 電流ソース、 10x レンジ	_	5.5	_	μА	CTMUICON<1:0> = 01	- 2.5V < VDD < VDDMAX		
	Іоит3	CTMU 電流ソース、 100x レンジ	_	55	_	μА	CTMUICON<1:0> = 10	- 2.5V < VDD < VDDMAX		
	Iout4	CTMU 電流ソース、 1000x レンジ	_	550	_	μА	CTMUICON<1:0> = 11 Note 2			
	VF	温度ダイオード 順方向電圧	_	.76	_	V				
	VΔ	1 °Cあたりの 電圧変化	_	3	_	mV/ °C				

Note 1: 電流調整レンジの中央 (CTMUICON<7:2> = 000000) における公称電圧です。PIC24F32KA では、IOT4 を 選択した時の電流出力は typical 電流値以下に制限されます。

2: 温度検出ダイオードには、この電流レンジを使用しないでください。

29.2 AC 特性とタイミング パラメータ

以下に PIC24FV32KA304 ファミリの AC 特性とタイミング パラメータの定義を記載します。

表 29-17: 温度および電圧仕様 - AC

標準動作条件: 1.8 ~ 3.6 V: PIC24F32KA3XX 2.0 ~ 5.5 V: PIC24FV32KA3XX 動作温度: -40 ℃ ≤ TA ≤ +85 ℃ (産業用温度レンジ品) 動作電圧 (VDD) レンジは 29.1 「DC 特性」に記載しています。

図 29-3: デバイスのタイミング仕様に対応する負荷条件

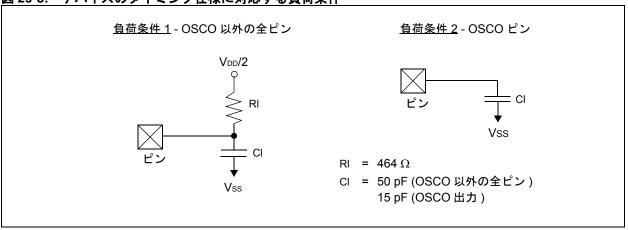


表 29-18: 出力ピンの容量性負荷要件

パラ メータ No.	記号	特性	Min	Typ ⁽¹⁾	Max	単位	条件
DO50	Cosc2	OSCO/CLKO ピン	_	_	15		XT および HS モードで、外部 クロックを使用して OSCI を 駆動する時
DO56	Сю	全 I/O ピンと OSCO	_	_	50	pF	EC モード
DO58	Св	SCLx, SDAx			400	pF	I ² C™ モード

Note 1: 「Typ」列のデータは、特に明記のない限り 3.3 V、25 ℃での値です。各パラメータ値はあくまでも設計指標値であり、テストで確認した値ではありません。

図 29-4: 外部クロックのタイミング

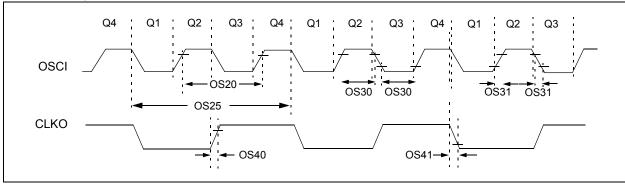


表 29-19: 外部クロックのタイミング要件

AC 特性		<u> </u>	標準動作条件		0.5 °0	• 3.6 V: PIC24F32KA3XX • 5.5 V: PIC24FV32KA3XX	
パラ メータ No.	記号	特性	Min	Typ ⁽¹⁾	≤ +85 C Max	(産業用 単位	温度レンジ品) 条件
OS10	Fosc	外部 CLKI 周波数 (外部クロックは EC モー ドでのみ使用可能)	DC 4	1 1	32 8	MHz MHz	EC ECPLL
		オシレータ周波数	0.2 4 4 31	 - -	4 25 8 33	MHz MHz MHz kHz	XT HS XTPLL SOSC
OS20	Tosc	Tosc = 1/FOSC	_	_	_	_	Fosc 値に関してはパラメータ OS10 を参照してください。
OS25	Tcy	命令サイクル時間 ⁽²⁾	62.5	_	DC	ns	
OS30	TosL、 TosH	外部クロック入力 (OSCI) HIGH または LOW 時間	0.45 x Tosc	_	_	ns	EC
OS31	TosR、 TosF	外部クロック入力 (OSCI) 立ち上がりまたは立ち下 がり時間	_		20	ns	EC
OS40	TckR	CLKO 立ち上がり時間 ⁽³⁾	_	6	10	ns	
OS41	TckF	CLKO 立ち下がり時間 ⁽³⁾	_	6	10	ns	

- Note 1: 「Typ」列のデータは、特に明記のない限り 3.3 V、25 ℃での値です。各パラメータ値はあくまでも設計指標値であり、テストで確認した値ではありません。
 - 2: 命令サイクル周期 (TCY) は、入力オシレータのタイムベース周期の 2 倍です。全ての仕様値は、標準動作条件下でデバイスにコードを実行させた時の、特定オシレータタイプの特性データに基づいています。これらの仕様限界を守らないと、オシレータの不安定な動作や消費電流の予期せぬ増加が生じる可能性があります。全てのデバイスは、OSCI/CLKI ピンに外部クロックを供給した状態で、「Min」値の条件で動作する事をテスト済みです。全てのデバイスにおいて、外部クロック入力を使用する場合のサイクル時間の「Max」値は「DC」(すなわちクロックなし)です。
 - 3: 計測は EC モードで実施しています。CLKO 信号は OSCO ピンで計測しています。CLKO は Q1 ~ Q2 周期 (1/2 Tcy) で LOW、Q3 ~ Q4 周期 (1/2 Tcy) で HIGH です。

表 29-20: PLL クロック タイミングの仕様

AC 特性			標準動作第	条件 :		1.8 ~ 3.6 V: PIC24F32KA3XX 2.0 ~ 5.5 V: PIC24FV32KA3XX			
			動作温度:	動作温度 : -40 ℃ ≤ TA ≤ +85 ℃ (産業用温度レンジ品)					
パラ メータ No.	記号	特性 ⁽¹⁾	Min	Typ ⁽²⁾	Max	単位	条件		
OS50	FPLLI	PLL 入力周波数レンジ	4	_	8	MHz	ECPLL、HSPLL モード、 -40°C ≤ TA ≤ +85°C		
OS51	Fsys	PLL 出力周波数レンジ	16	_	32	MHz	-40 °C ≤ TA ≤ +85 °C		
OS52	TLOCK	PLL 起動時間 (ロック時間)	_	1	2	ms			
OS53	DCLK	CLKO 安定性(ジッタ)	-2	1	2	%	100 ms の期間で計測		

Note 1: これらのパラメータは特性評価済みですが、製品によるテストは実施していません。

2: 「Typ」列のデータは、特に明記のない限り 3.3 V、25 ℃での値です。各パラメータ値はあくまでも設計 指標値であり、テストで確認した値ではありません。

表 29-21: AC 特性: 内部 FRC の精度

<u>女 ZJ-Z I</u>	· AC 技压·对即FK	O VITEI	ζ.							
		標準動作	乍条件:			1.8 ~ 3.6 V: PIC24F32KA3XX				
AC 特性						2.0 ~ 5.5 V: PIC24FV32KA3XX				
		動作温度 : -40 ℃ ≤ TA ≤ +85 ℃ (産業用温度レンジ品)								
パラ										
メータ	特性	Min	Тур	Max	単位	条件				
番号										
F20	内部 FRC の精度 (8 M	Hz) ⁽¹⁾								
	FRC	-2	_	+2	%	+25 ℃	3.0 V ≤ VDD ≤ 3.6 V (「F」デ			
							バイス)、3.2 V ≤ VDD ≤ 5.5 V (
							「FV」デバイス)			
		-5	_	+5	%	-40 °C ≤ TA ≤ +85 °C	1.8 V ≤ VDD ≤ 3.6 V (「F」デ			
							バイス)、2.0 V ≤ VDD ≤ 5.5 V (
							「FV」デバイス)			

Note 1: 周波数は 25 °C /3.3 V の条件で校正されています。OSCTUN ビットを使用して温度ドリフトを補正できます。

表 29-22: AC 特性: 内部 RC の精度

	110 4 - 1 1	11111						
		標準動作条件:				1.8 ~ 3.6 V: PIC24F32KA3XX		
AC 特性	ŧ	2.0 ~ 5.5 V: PIC24FV32KA3XX						
		動作温度:-40 °C ≤ TA ≤ +85 °C (産業用温度レンジ品)						
パラ メータ No.	特性	Min	Тур	Max	単位	条件		
	LPRC (31 kHz) ⁽¹⁾							
F21		-15	_	15	%			

Note 1: LPRC 周波数は VDD によって変動します。

表 29-23: 内部 RC オシレータの仕様

<u> </u>	О. РЭЦ	S N C カ クレータの114米	描绘制作名	<i></i>		10	2 6 V. DIC24E22KA2VV		
AC 特性			標準動作条件: 1.8 ~ 3.6 V: PIC24F32KA3XX 2.0 ~ 5.5 V: PIC24FV32KA3XX 動作温度: -40 °C ≤ TA ≤ +85 °C (産業用温度レンジ品) -40 °C ≤ TA ≤ +125 °C (拡張温度レンジ品)						
パラ メータ No.	記号	特性 ⁽¹⁾	Min	Тур	Max	単位	条件		
	TFRC	FRC 起動時間	_	5	_	μS			
	TLPRC	LPRC 起動時間	_	70	_	μS			

図 29-5: CLKO と I/O のタイミング特性

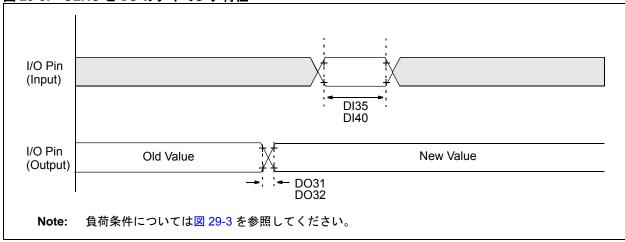


表 29-24: CLKO と I/O のタイミング要件

AC 特性	Ē		標準動作条動作条動作温度:-		: +85 °C (産	1.8 ~ 3.6 V: PIC24F32KA3XX 2.0 ~ 5.5 V: PIC24FV32KA3XX 業用温度レンジ品)		
パラ メータ 記号 特性 No.			Min	Typ ⁽¹⁾	Max	単位	条件	
DO31	TioR	ポート出力立ち上がり時間	_	10	25	ns		
DO32	TioF	ポート出力立ち下がり時間	_	10	25	ns		
DI35	TINP	INTx ピンの HIGH または LOW 時間 (出力)	20	_	_	ns		
DI40	TRBP	CNx の HIGH または LOW 時間 (入力)	2	_	_	Tcy		

Note 1: 特に明記のない限り、「Typ」列のデータは 3.3 V/25 °C (PIC24F32KA3XX); 5.0 V/25 °C (PIC24FV32KA3XX) での値です。

表 29-25: コンパレータのタイミング

パラ メータ No.	記号	特性	Min	Тур	Max	単位	注釈
300	TRESP	応答時間 * ⁽¹⁾	_	150	400	ns	
301	TMC2OV	コンパレータのモード変更から出力確定までの時間 *	_	_	10	μS	

^{*} これらのパラメータは特性データであり、テストはしていません。

Note 1: 応答時間は、片方のコンパレータ入力を(VDD-1.5)/2 に、もう一方の入力を Vss から VDD に変化させて計測しています。

表 29-26: コンパレータ参照電圧設定時間の仕様

パラ メータ No.	記号	特性	Min	Тур	Max	単位	注釈
VR310	TSET	セトリングタイム ⁽¹⁾			10	μS	

Note 1: セトリングタイムは、CVRSS = 1 の時に CVR<3:0> ビットが「0000」から「1111」に変化するまでの時間として計測しています。

表 29-27: ADC モジュールの仕様

<u> </u>			標準動作条件		F	1.8 ~ 3.6 V: PIC24F32KA3XX 2.0 ~ 5.5 V: PIC24FV32KA3XX		
パラ メータ 番号	記号	特性	Min.	<u>Typ</u>	Max.	座兼用》 単位	<u> </u>	
デバイ	<u></u> ス電源						1	
AD01	AVDD	モジュールの VDD 電源	VDD - 0.3 か 1.8 の 大きい方	_	VDD + 0.3 か 3.6 の 小さい方	V		
AD02	AVss	モジュールの Vss 電源	Vss - 0.3	_	Vss + 0.3	V		
参照入:	<u></u> カ							
AD05	VREFH	参照電圧 HIGH	AVss + 1.7	_	AVDD	V		
AD06	VREFL	参照電圧 LOW	AVss	_	AVDD - 1.7	V		
AD07	VREF	絶対参照電圧	AVss - 0.3	_	AVDD + 0.3	V		
アナロ	グ入力							
AD10	VINH-VINL	フルスケール入力スパン	VREFL	_	VREFH	V	(Note 2)	
AD11	VIN	絶対入力電圧	AVss - 0.3	_	AVDD + 0.3	V		
AD12	VINL	絶対 VINL 入力電圧	AVss - 0.3		AVDD/2	V		
AD17	Rin	アナログ電圧ソースの 推奨インピーダンス		_	2.5K	Ω	12 ビット	
ADC の	精度							
AD20b	NR	分解能		12	_	bit		
AD21b	INL	積分非直線性	_	± 1	± 9	LSb	VINL = AVSS = VREFL = 0 V AVDD = VREFH = 5 V	
AD22b	DNL	微分非直線性	_	± 1	± 5	LSb	VINL = AVSS = VREFL = 0 V AVDD = VREFH = 5 V	
AD23b	GERR	ゲイン誤差	_	± 1	± 9	LSb	VINL = AVSS = VREFL = 0 V AVDD = VREFH = 5 V	
AD24b	EOFF	オフセット誤差	_	± 1	± 5	LSb	VINL = AVSS = VREFL = 0 V AVDD = VREFH = 5 V	
AD25b		単調性 (1)	_	_	-	_	保証	

Note 1: 入力電圧の増加に対して ADC 結果が減少しない事が保証されます。

2: 計測には、ADC 参照電圧として、外部 VREF+ と VREF- を使用しています。

表 29-28: ADC のタイミング要件 (1)

3X 23-Z	U. ADU	のダイミング要件い	標準動作	上冬州,		10	~ 3.6 V: PIC24F32KA3XX
AC 特性			保牛乳T	F宋什:		?5.5 V: PIC24FV32KA3XX	
			動作温度: -40 °C ≤ TA ≤ +85 °C (産業用温度レンジ品)				
パラ メータ 番号	記号	特性	Min.	Тур	Max.	単位	条件
クロック	ク パラメー	·\$					
AD50	TAD	ADC クロック周期	75	_	_	ns	Tcy = 75 ns、AD1CON3 は 既定値状態
AD51	TRC	ADC 内部 RC オシレータ周期	_	250	_	ns	
変換レ-	- -						
AD55	TCONV	変換時間	_	12	_	TAD	
AD56	FCNV	スループット レート	_	_	100	ksps	$AVDD \ge 2.7 V$
AD57	TSAMP	サンプリング時間	_	1	_	TAD	
AD58	TACQ	アクイジション時間	750	_	_	ns	(Note 2)
AD59	Tswc	変換からサンプリングへの切 り換え時間	_	_	(Note 3)		
AD60	TDIS	放電時間	0.5	_	_	TAD	
クロック	ク パラメー	·タ					
AD61	TPSS	サンプリング ビット (SAMP) のセットからサンプリング開 始までの遅延時間	2	_	3	TAD	

- **Note 1:** サンプリング コンデンサは時間とともに放電するため、クロック周波数が 10 kHz よりも低いと、特に高温条件で線形性に影響する可能性があります。
 - **2:** 変換後に電圧がフルスケールで変化 (VDD から VSS または VSS から VDD に変化) した時に、ホールドコンデンサが「新しい」入力電圧で充電されるまでの時間
 - 3: デバイスクロックの次のサイクルで切り換わります。

表 29-29: リセット、ウォッチドッグ タイマ、オシレータ起動タイマ、パワーアップタイマ、ブラウン アウト リセットのタイミング要件

AC 特性			標準動作		C ≤ TA ≤ +8:	1.8 ~ 3.6 V: PIC24F32KA3XX 2.0?5.5 V: PIC24FV32KA3XX 用温度レンジ品)	
パラ メータ 番号	記号	特性	Min.	Typ ⁽¹⁾	Max.	単位	条件
SY10	TmcL	MCLR パルス幅 (LOW)	2	_	_	μS	
SY11	TPWRT	パワーアップタイマ時間	50	64	90	ms	
SY12	TPOR	パワーオン リセット遅延	1	5	10	μS	
SY13	Tioz	MCLR = LOW またはウォッチドッグ タイマリセット後に I/O ピンが高インピーダンスになるまでの時間	_	_	100	ns	
SY20	TWDT	ウォッチドッグ タイマ	0.85	1.0	1.15	ms	1.32 プリスケーラ
		タイムアウト時間	3.4	4.0	4.6	ms	1:128 プリスケーラ
SY25	TBOR	ブラウンアウト リセット パルス幅	1	_	_	μS	
SY45	TRST	内部状態リセット時間	_	5	_	μS	
SY55	TLOCK	PLL 起動時間	_	100	_	μS	
SY65	Tost	オシレータ起動時間	_	1024	_	Tosc	
SY70	TDSWU	ディープスリープからの復 帰時間	_	100	_	μS	VCAP で充電された 10 μF コン デンサの完全放電に基づく; TPOR と TRST を含む
SY71	ТРМ	プログラムメモリの 復帰時間	_	1	_	μS	PMSLP = 0 によるスリープ復帰
SY72	TLVR	低電圧レギュレータの 復帰時間	_	250	_	μ\$	

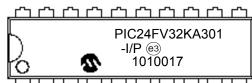
Note 1: 「Typ」列のデータは、特に明記のない限り3.3 V、25 ℃での値です。

NOTE:

30.0 パッケージ情報

30.1 パッケージのマーキング情報

20-Lead PDIP (300 mil)

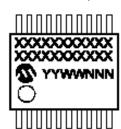


Example

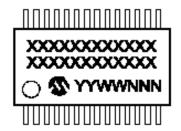
28-Lead SPDIP (.300")



20-Lead SSOP (5.30 mm)



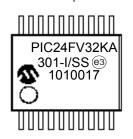
28-Lead SSOP (5.30 mm)



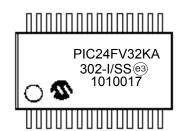
Example



Example



Example



凡例: XX...X お客様固有情報

Y 年コード(西暦の下1桁) YY 年コード(西暦の下2桁)

WW 週コード (1月の第1週を「01」とする) NNN 英数字のトレーサビリティ コード

e3 つや消し錫 (Sn) の使用を示す鉛フリー JEDEC マーク

このパッケージは鉛フリーです。鉛フリー JEDC マーク(e3)

は外箱に表記しています。

Note: マイクロチップ社の製品番号が1行に収まりきらない場合、複数行を使用します。この場合、お客様固有情報に使用できる文字数が制限されます。

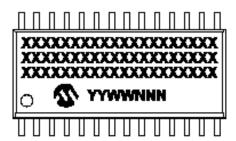
20-Lead SOIC (7.50 mm)



Example



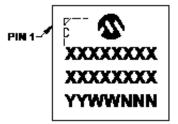
28-Lead SOIC (7.50 mm)



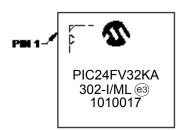
Example



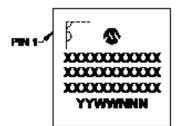
28-Lead QFN (6x6 mm)



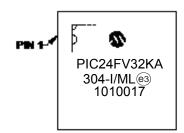
Example



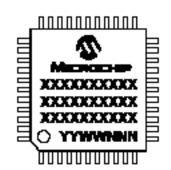
44-Lead QFN (8x8x0.9 mm)



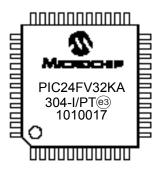
Example



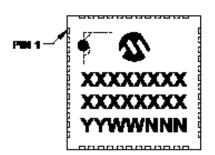
44-Lead TQFP (10x10x1 mm)



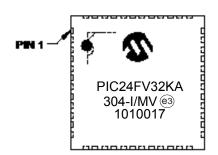
Example



48-Lead UQFN (6x6x0.5 mm)



Example

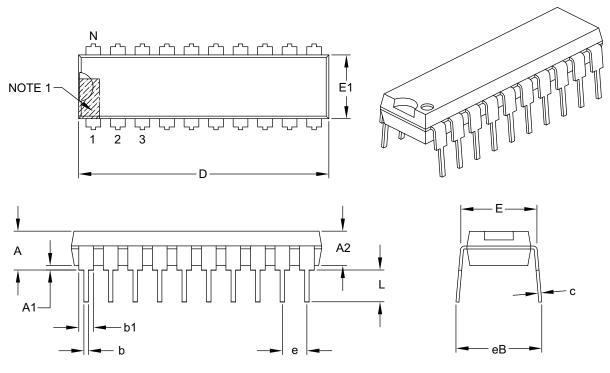


30.2 パッケージの詳細

以下に、各パッケージの技術的詳細を記載します。

20-Lead Plastic Dual In-Line (P) – 300 mil Body [PDIP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



	Units		INCHES	
	Dimension Limits	MIN	NOM	MAX
Number of Pins	N		20	
Pitch	е		.100 BSC	
Top to Seating Plane	A	_	_	.210
Molded Package Thickness	A2	.115	.130	.195
Base to Seating Plane	A1	.015	_	_
Shoulder to Shoulder Width	Е	.300	.310	.325
Molded Package Width	E1	.240	.250	.280
Overall Length	D	.980	1.030	1.060
Tip to Seating Plane	L	.115	.130	.150
Lead Thickness	С	.008	.010	.015
Upper Lead Width	b1	.045	.060	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	_	_	.430

Notes:

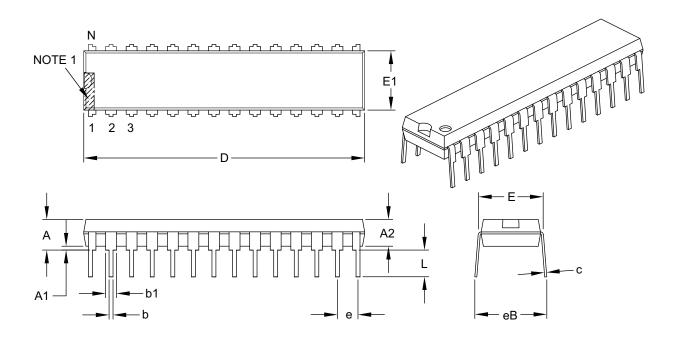
- 1. Pin 1 visual index feature may vary, but must be located within the hatched area.
- 2. § Significant Characteristic.
- 3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
- 4. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-019B

28-Lead Skinny Plastic Dual In-Line (SP) - 300 mil Body [SPDIP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



	Units		INCHES		
Dimensio	n Limits	MIN	NOM	MAX	
Number of Pins	N		28		
Pitch	е		.100 BSC		
Top to Seating Plane	Α	_	_	.200	
Molded Package Thickness	A2	.120	.135	.150	
Base to Seating Plane	A1	.015	_	_	
Shoulder to Shoulder Width	Е	.290	.310	.335	
Molded Package Width	E1	.240	.285	.295	
Overall Length	D	1.345	1.365	1.400	
Tip to Seating Plane	L	.110	.130	.150	
Lead Thickness	С	.008	.010	.015	
Upper Lead Width	b1	.040	.050	.070	
Lower Lead Width	b	.014	.018	.022	
Overall Row Spacing §	eВ	_	_	.430	

Notes:

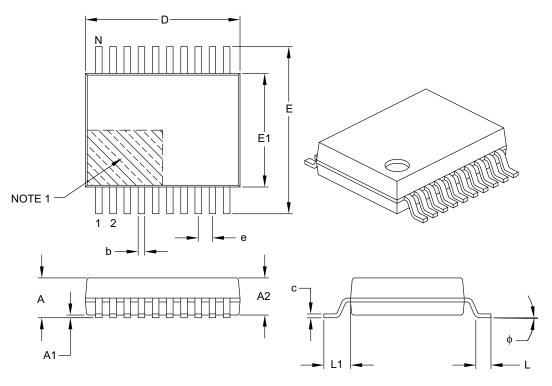
- 1. Pin 1 visual index feature may vary, but must be located within the hatched area.
- 2. § Significant Characteristic.
- 3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
- 4. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-070B

20-Lead Plastic Shrink Small Outline (SS) – 5.30 mm Body [SSOP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



	Units		MILLIMETERS		
	Dimension Limits		NOM	MAX	
Number of Pins	N		20		
Pitch	е		0.65 BSC		
Overall Height	А	_	_	2.00	
Molded Package Thickness	A2	1.65	1.75	1.85	
Standoff	A1	0.05	_	_	
Overall Width	E	7.40	7.80	8.20	
Molded Package Width	E1	5.00	5.30	5.60	
Overall Length	D	6.90	7.20	7.50	
Foot Length	L	0.55	0.75	0.95	
Footprint	L1	1.25 REF			
Lead Thickness	С	0.09	_	0.25	
Foot Angle	ф	0°	4°	8°	
Lead Width	b	0.22	_	0.38	

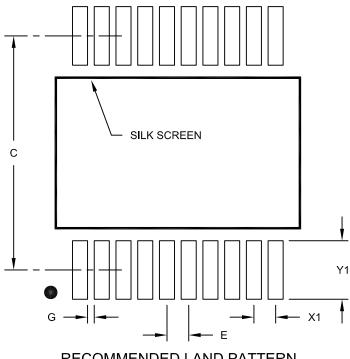
Notes:

- 1. Pin 1 visual index feature may vary, but must be located within the hatched area.
- 2. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.20 mm per side.
- 3. Dimensioning and tolerancing per ASME Y14.5M.
 - BSC: Basic Dimension. Theoretically exact value shown without tolerances.
 - REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-072B

20-Lead Plastic Shrink Small Outline (SS) - 5.30 mm Body [SSOP]

For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

	MILLIMETERS			
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Contact Pad Spacing	С		7.20	
Contact Pad Width (X20)	X1			0.45
Contact Pad Length (X20)	Y1			1.75
Distance Between Pads	G	0.20		

Notes:

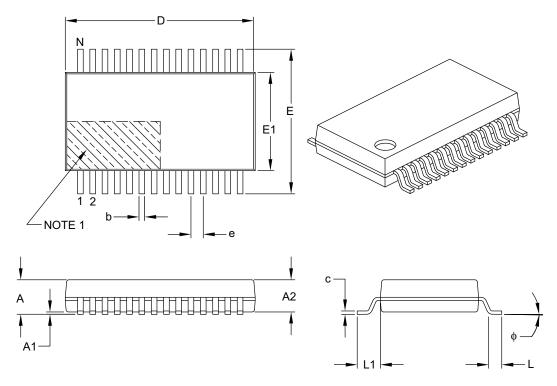
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2072A

28-Lead Plastic Shrink Small Outline (SS) - 5.30 mm Body [SSOP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



LISTO MILLIMETERS					
	Units		MILLIMETERS		
	Dimension Limits		NOM	MAX	
Number of Pins	N		28		
Pitch	е		0.65 BSC		
Overall Height	A	-	_	2.00	
Molded Package Thickness	A2	1.65	1.75	1.85	
Standoff	A1	0.05	_	-	
Overall Width	E	7.40	7.80	8.20	
Molded Package Width	E1	5.00	5.30	5.60	
Overall Length	D	9.90	10.20	10.50	
Foot Length	L	0.55	0.75	0.95	
Footprint	L1	1.25 REF			
Lead Thickness	С	0.09	_	0.25	
Foot Angle	ф	0°	4°	8°	
Lead Width	b	0.22	_	0.38	

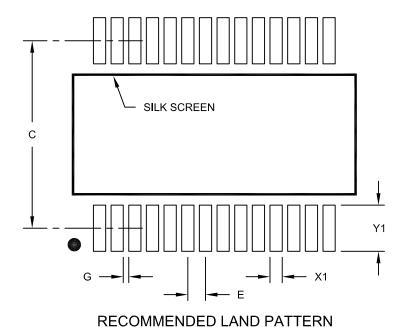
Notes:

- 1. Pin 1 visual index feature may vary, but must be located within the hatched area.
- 2. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.20 mm per side.
- 3. Dimensioning and tolerancing per ASME Y14.5M.
 - BSC: Basic Dimension. Theoretically exact value shown without tolerances.
 - REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-073B

28-Lead Plastic Shrink Small Outline (SS) - 5.30 mm Body [SSOP]

Ste: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Units MILLIMETERS **Dimension Limits** MIN NOM MAX Contact Pitch 0.65 BSC Ε Contact Pad Spacing С 7.20 Contact Pad Width (X28) X1 0.45 Contact Pad Length (X28) Y1 1.75 Distance Between Pads G 0.20

Notes:

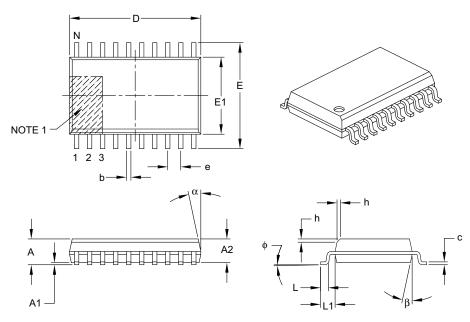
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2073A

20-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Units		MILLIMETERS		
	Dimension Limits		NOM	MAX
Number of Pins	N		20	
Pitch	е		1.27 BSC	
Overall Height	A	_	_	2.65
Molded Package Thickness	A2	2.05	_	_
Standoff §	A1	0.10	_	0.30
Overall Width	E	10.30 BSC		
Molded Package Width	E1	7.50 BSC		
Overall Length	D	12.80 BSC		
Chamfer (optional)	h	0.25	_	0.75
Foot Length	L	0.40	_	1.27
Footprint	L1		1.40 REF	
Foot Angle	ф	0°	_	8°
Lead Thickness	С	0.20	_	0.33
Lead Width	b	0.31	_	0.51
Mold Draft Angle Top	α	5°	_	15°
Mold Draft Angle Bottom	β	5°	_	15°

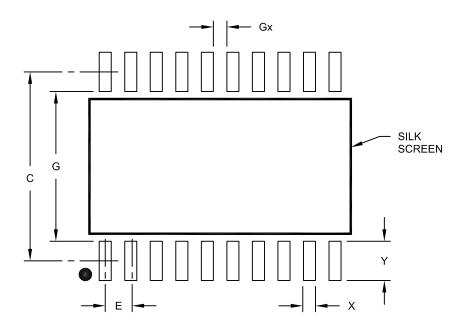
Notes:

- 1. Pin 1 visual index feature may vary, but must be located within the hatched area.
- 2. § Significant Characteristic.
- 3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15 mm per side.
- 4. Dimensioning and tolerancing per ASME Y14.5M.
 - BSC: Basic Dimension. Theoretically exact value shown without tolerances.
 - REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-094B

20-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

	Units	N	/ILLIMETERS		
Dimension Limits		MIN	NOM	MAX	
Contact Pitch	E	1.27 BSC			
Contact Pad Spacing	С		9.40		
Contact Pad Width (X20)	Х			0.60	
Contact Pad Length (X20)	Υ			1.95	
Distance Between Pads	Gx	0.67			
Distance Between Pads	G	7.45			

Notes:

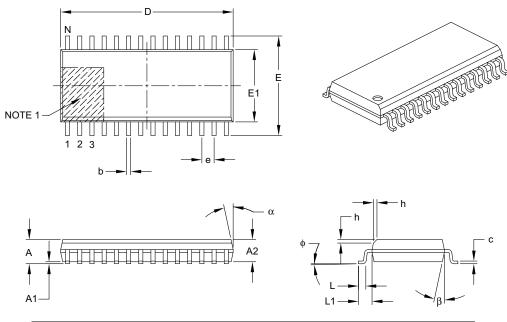
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2094A

28-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



	Units	MILLIMETERS		
	Dimension Limits	MIN	NOM	MAX
Number of Pins	N		28	
Pitch	е		1.27 BSC	
Overall Height	A	_	_	2.65
Molded Package Thickness	A2	2.05	-	_
Standoff §	A1	0.10	-	0.30
Overall Width	E	10.30 BSC		
Molded Package Width	E1	7.50 BSC		
Overall Length	D	17.90 BSC		
Chamfer (optional)	h	0.25	_	0.75
Foot Length	L	0.40	_	1.27
Footprint	L1		1.40 REF	
Foot Angle Top	ф	0°	_	8°
Lead Thickness	С	0.18	_	0.33
Lead Width	b	0.31	_	0.51
Mold Draft Angle Top	α	5°	_	15°
Mold Draft Angle Bottom	β	5°	_	15°

Notes:

- 1. Pin 1 visual index feature may vary, but must be located within the hatched area.
- 2. § Significant Characteristic.
- 3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15 mm per side.
- 4. Dimensioning and tolerancing per ASME Y14.5M.

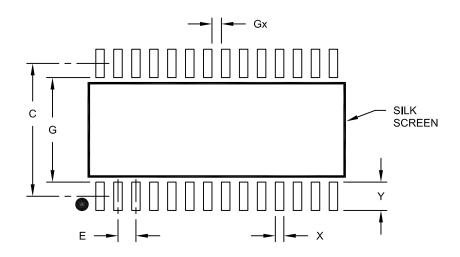
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-052B

28-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

	Units	N	IILLIMETER	S
Dimension Limits		MIN	NOM	MAX
Contact Pitch	Е	1.27 BSC		
Contact Pad Spacing	С		9.40	
Contact Pad Width (X28)	Х			0.60
Contact Pad Length (X28)	Υ			2.00
Distance Between Pads	Gx	0.67		
Distance Between Pads	G	7.40		

Notes:

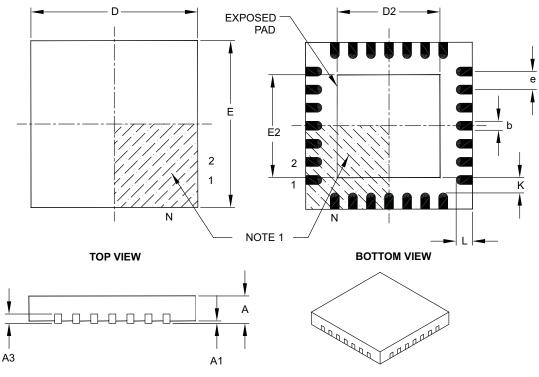
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2052A

28-Lead Plastic Quad Flat, No Lead Package (ML) – 6x6 mm Body [QFN] with 0.55 mm Contact Length

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



	Units		MILLIMETERS		
Dimension	Dimension Limits		NOM	MAX	
Number of Pins	N		28		
Pitch	е		0.65 BSC		
Overall Height	Α	0.80	0.90	1.00	
Standoff	A1	0.00	0.02	0.05	
Contact Thickness	A3		0.20 REF		
Overall Width	Е		6.00 BSC		
Exposed Pad Width	E2	3.65	3.70	4.20	
Overall Length	D		6.00 BSC		
Exposed Pad Length	D2	3.65	3.70	4.20	
Contact Width	b	0.23	0.30	0.35	
Contact Length	L	0.50	0.55	0.70	
Contact-to-Exposed Pad	K	0.20	_	_	

Notes:

- 1. Pin 1 visual index feature may vary, but must be located within the hatched area.
- 2. Package is saw singulated.
- 3. Dimensioning and tolerancing per ASME Y14.5M.

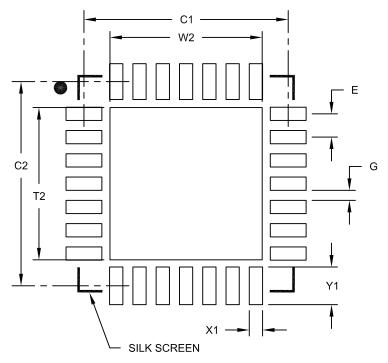
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-105B

28-Lead Plastic Quad Flat, No Lead Package (ML) – 6x6 mm Body [QFN] with 0.55 mm Contact Length

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.65 BSC	
Optional Center Pad Width	W2			4.25
Optional Center Pad Length	T2			4.25
Contact Pad Spacing	C1		5.70	
Contact Pad Spacing	C2		5.70	
Contact Pad Width (X28)	X1			0.37
Contact Pad Length (X28)	Y1			1.00
Distance Between Pads	G	0.20		

Notes:

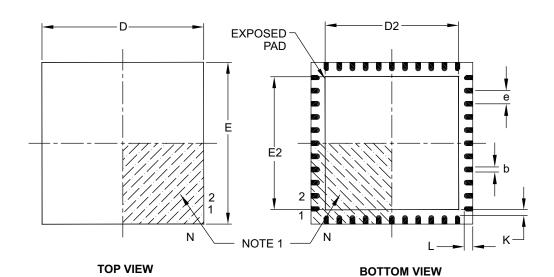
1. Dimensioning and tolerancing per ASME Y14.5M

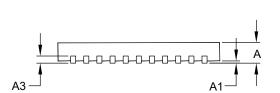
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

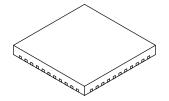
Microchip Technology Drawing No. C04-2105A

44-Lead Plastic Quad Flat, No Lead Package (ML) - 8x8 mm Body [QFN]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging







	Units		MILLIMETERS		
	Dimension Limits	MIN	NOM	MAX	
Number of Pins	N		44		
Pitch	е		0.65 BSC		
Overall Height	A	0.80	0.90	1.00	
Standoff	A1	0.00	0.02	0.05	
Contact Thickness	A3	0.20 REF			
Overall Width	E		8.00 BSC		
Exposed Pad Width	E2	6.30	6.45	6.80	
Overall Length	D		8.00 BSC		
Exposed Pad Length	D2	6.30	6.45	6.80	
Contact Width	b	0.25	0.30	0.38	
Contact Length	L	0.30	0.40	0.50	
Contact-to-Exposed Pad	K	0.20	_	_	

Notes:

- 1. Pin 1 visual index feature may vary, but must be located within the hatched area.
- 2. Package is saw singulated.
- 3. Dimensioning and tolerancing per ASME Y14.5M.

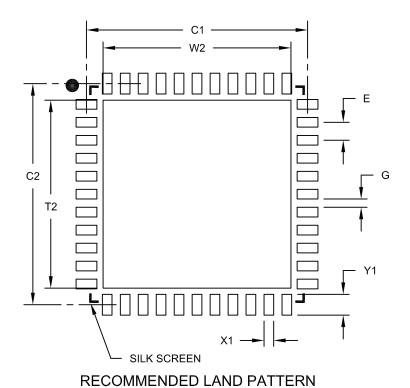
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-103B

44-Lead Plastic Quad Flat, No Lead Package (ML) - 8x8 mm Body [QFN]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Units **MILLIMETERS Dimension Limits** MIN MOM MAX Contact Pitch Е 0.65 BSC Optional Center Pad Width W2 6.80 Optional Center Pad Length T2 6.80 Contact Pad Spacing C1 8.00 Contact Pad Spacing C2 8.00 Contact Pad Width (X44) X1 0.35 Contact Pad Length (X44) 0.80 Υ1 Distance Between Pads G 0.25

Notes:

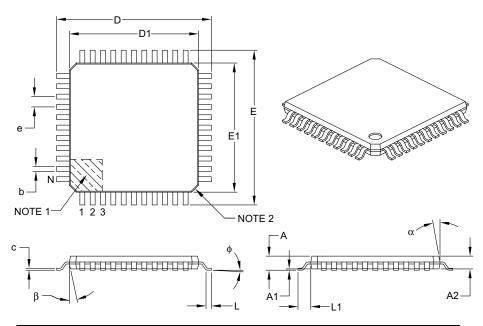
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2103A

44-Lead Plastic Thin Quad Flatpack (PT) - 10x10x1 mm Body, 2.00 mm [TQFP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



	Units		MILLIMETERS	
Dimens	sion Limits	MIN	NOM	MAX
Number of Leads	N	44		
Lead Pitch	е		0.80 BSC	
Overall Height	А	-	_	1.20
Molded Package Thickness	A2	0.95	1.00	1.05
Standoff	A1	0.05	_	0.15
Foot Length	L	0.45	0.60	0.75
Footprint	L1		1.00 REF	
Foot Angle	ф	0°	3.5°	7°
Overall Width	E		12.00 BSC	•
Overall Length	D		12.00 BSC	
Molded Package Width	E1		10.00 BSC	
Molded Package Length	D1	10.00 BSC		
Lead Thickness	С	0.09	_	0.20
Lead Width	b	0.30	0.37	0.45
Mold Draft Angle Top	α	11°	12°	13°
Mold Draft Angle Bottom	β	11°	12°	13°

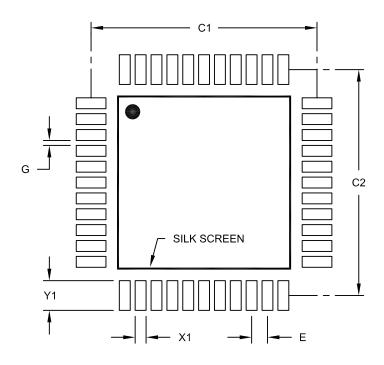
Notes:

- 1. Pin 1 visual index feature may vary, but must be located within the hatched area.
- 2. Chamfers at corners are optional; size may vary.
- 3. Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.25 mm per side.
- 4. Dimensioning and tolerancing per ASME Y14.5M.
 - BSC: Basic Dimension. Theoretically exact value shown without tolerances.
 - REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-076B

44-Lead Plastic Thin Quad Flatpack (PT) – 10x10x1 mm Body, 2.00 mm [TQFP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIM	ETERS	
Dimension Limits		MIN	NOM	MAX
Contact Pitch	Е		0.80 BSC	
Contact Pad Spacing	C1		11.40	
Contact Pad Spacing	C2		11.40	
Contact Pad Width (X44)	X1			0.55
Contact Pad Length (X44)	Y1			1.50
Distance Between Pads	G	0.25		

Notes:

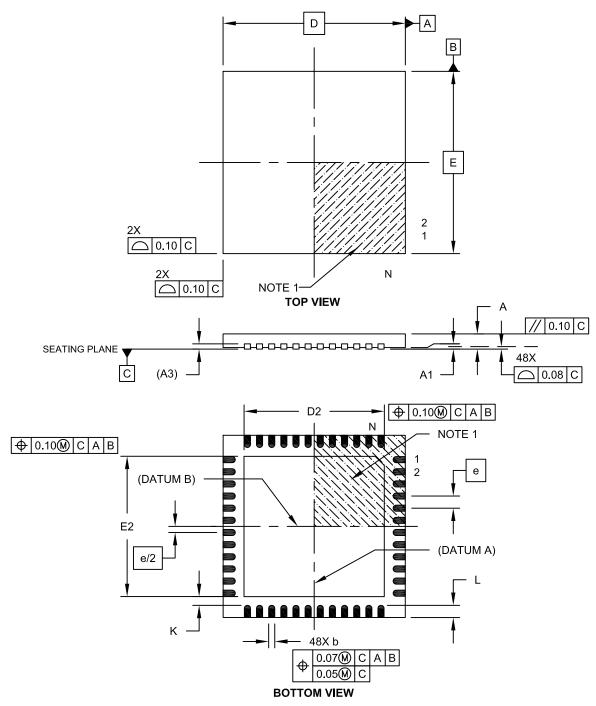
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2076A

48-Lead Plastic Ultra Thin Quad Flat, No Lead Package (MV) – 6x6x0.5 mm Body [UQFN]

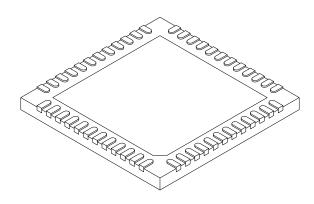
Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Microchip Technology Drawing C04-153A Sheet 1 of 2

48-Lead Plastic Ultra Thin Quad Flat, No Lead Package (MV) - 6x6x0.5 mm Body [UQFN]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



	Units	N	IILLIMETER	S
Dimension	Limits	MIN	NOM	MAX
Number of Pins	Ν	48		
Pitch	е	0.40 BSC		
Overall Height	Α	0.45	0.50	0.55
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3		0.127 REF	
Overall Width	Ш	6.00 BSC		
Exposed Pad Width	E2	4.45	4.60	4.75
Overall Length	Д	6.00 BSC		
Exposed Pad Length	D2	4.45	4.60	4.75
Contact Width	b	0.15	0.20	0.25
Contact Length	L	0.30	0.40	0.50
Contact-to-Exposed Pad	K	0.20	-	-

Notes:

- 1. Pin 1 visual index feature may vary, but must be located within the hatched area.
- 2. Package is saw singulated.
- 3. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-153A Sheet 2 of 2

NOTE:

補遺 A: 改訂履歴

リビジョン A (2011年3月)

PIC24FV32KA304 ファミリデバイス用データシート の初版

リビジョンB(2011年10月)

25.0「充電時間計測ユニット(CTMU)」内のCTMUICON レジスタ (レジスタ 25-3 参照) の IRNG ビットに関す る記述を下記のように変更 ;

「01」: ベース電流レベル (公称値 0.55 μA)、 「00」: ベース電流を 1000 倍する

29.0 「電気的特性」内の IPD 仕様値 (typical) を下記のように変更 ;

- DC20h/i/j/k 204 μA から 200 μA に変更
- DC60h/i/j/k 0.15 μA から 0.025 μA に変更
- DC60l/m/n/o 0.25 μA から 0.040 μA に変更
- DC72h/i/j/k 0.80 μA から 0.70 μA に変更

NOTE:

マイクロチップ社ウェブサイト

マイクロチップ社は、自社が運営する WWW サイト (www.microchip.com) を通してオンライン サポートを提供しています。当ウェブサイトでは、お客様に役立つ情報やファイルを簡単に見つけ出せます。お好みのインターネット ブラウザを使用して、下記の内容をご覧になれます。

- 製品サポート データシートとエラッタ、アプリケーションノート、サンプルコード、設計リソース、ユーザガイドとハードウェアサポート文書、最新ソフトウェアリリース、ソフトウェアアーカイブ
- 一般的技術サポート よく寄せられる質問 (FAQ)、技術サポートのご依頼、オンライン ディスカッション グループ、マイクロチップ社のコンサルタント プログラムおよびメンバーリスト
- **ご注文とお問い合わせ** 製品セレクタと注文ガイド、最新プレスリリース、セミナー/イベントの一覧、お問い合わせ先(営業所/販売代理店/工場)の一覧

顧客変更通知サービス

マイクロチップ社の顧客変更通知サービスは、お客様にマイクロチップ社製品の最新情報をお届けする配信サービスです。ご興味のある製品ファミリまたは開発ツールに関する変更、更新、エラッタ情報をいち早くメールにてお知らせします。

当サービスをご希望のお客様は、マイクロチップ社 ウェブサイト(www.microchip.com)でご登録ください。 [サポート]枠から[変更通知]をクリックし、画面の 指示に従ってください。

お客様サポート

マイクロチップ社製品のお客様は、下記のチャンネルからサポートをご利用頂けます。

- 代理店または販売担当者
- 各地の営業所
- ・ フィールド アプリケーション エンジニア (FAE)
- 技術サポート
- 開発システム情報ライン

サポートについては代理店、販売担当者、フィールドアプリケーションエンジニア(FAE)までお問い合わせください。各地の営業所もご利用頂けます。本書の末尾には各国の営業所の一覧を記載しています。

技術サポートは下記のウェブサイトからもご利用頂けます: http://microchip.com/support

お客様アンケート

マイクロチップ社は、弊社製品を存分にご活用頂くために、文書の作成に最善の努力を尽くしています。本書の構成、明確さ、内容等に関するご意見を下記まで FAX にてお寄せください。

宛先:マイクロチップ・テクノロジー・ジャパン株式会社: FAX 番号: 045-471-6122

下記のアンケートフォームにお客様情報と本書に関するご意見をご記入ください。

TO: RE:	マイクロチップ・テクノロジー・ジャパン株式会社 お客様アンケート	送信ページ数
Fron	n: ご氏名 会社名 住所	
	郵便番号 電話番号 :()	
アプ	リケーション (任意):	
	をご希望ですか?はい いいえ	
デバ	イス:PIC24FV32KA304 ファミリ	文書番号:DS39995B_JP
アン	ケート	
1. ;	本書で最も良かった点は何ですか?	
-		
2. ;	本書はお客様のハードウェア / ソフトウェア開発に役立っ	ちますか?
-		
3. ;	本書の構成はわかりやすいですか?わかりにくい場合は	その理由もご記入ください。
-		
4.	本書の構成と内容を改善するには何を追加すべきですか	?
-		
5. ;	本書から省略しても構わない内容は何ですか?	
-		
6.	不正確であったり誤解を招きやすい箇所がございました	らご指摘ください。
-		
7.	その他、本書の改善に向けてご意見がございましたらご	記入ください。
-		

製品識別システム

ご注文または製品の価格や納期につきましては、弊社または各地の営業所までお問い合わせください。

_			001:3 · H
	テープ&リールフラ 温度レンジ パッケージ	アミリ	例: a) PIC 拡引 44 b) PIC 標準 28
	アーキテクチャ	24 = 16 ビット改良型ハーバード (DSP なし)	
	フラッシュメモリ ファミリ	F = 標準電圧レンジ、フラッシュ プログラムメモリ FV = 拡張電圧レンジ、フラッシュ プログラムメモリ	
	製品グループ	FV - 孤張竜圧レンフ、フラッシュ フロップムアモッ KA3 = 汎用マイクロコントローラ	
	ピン数	01 = 20 ピン 02 = 28 ピン 04 = 44 ピン	
	温度レンジ	I = -40 ~ +85 ℃ (産業用)	
	パッケージ	SP = SPDIP SO = SOIC SS = SSOP ML = QFN P = PDIP PT = TQFP	
	パターン	3 桁の QTP、SQTP、その他のコード等 (あるいは空白のまま) ES = エンジニアリング サンプル	

- C24FV32KA304-I/ML:
 - R電圧レンジ、汎用、32 Kbyte プログラムメモリ、 ピン、産業用温度、QFN パッケージ
- C24F16KA302-I/SS: 準電圧レンジ、汎用、16 Kbyte プログラムメモリ、 3 ピン、産業用温度、SSOP パッケージ



各国の営業所とサービス

北米

本社 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel:480-792-7200 Fax:480-792-7277

技術サポート:

http://www.microchip.com/ support

URL:

www.microchip.com

アトランタ

Duluth, GA Tel:678-957-9614 Fax:678-957-1455

ボストン

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

シカゴ

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

クリーブランド Independence, OH

Tel:216-447-0464 Fax:216-447-0643

ダラス

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

デトロイト

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

インディアナポリス

Noblesville, IN Tel:317-773-8323 Fax:317-773-5453

ロサンゼルス

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

サンタクララ Santa Clara, CA Tel:408-961-6444 Fax:408-961-6445

トロント

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

アジア / 太平洋

アジア太平洋支社 Suites 3707-14, 37th Floor Tower 6, The Gateway Harbour City, Kowloon Hong Kong Tel:852-2401-1200 Fax:852-2401-3431

オーストラリア - シドニー

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

中国 - 北京

Tel:86-10-8569-7000 Fax:86-10-8528-2104

中国 - 成都

Tel:86-28-8665-5511 Fax:86-28-8665-7889

中国 - 重慶

Tel:86-23-8980-9588 Fax:86-23-8980-9500

中国 - 武漢 Tel:86-571-2819-3180 Fax:86-571-2819-3189

中国 - 香港 SAR

Tel:852-2401-1200 Fax:852-2401-3431

中国 - 南京

Tel:86-25-8473-2460 Fax:86-25-8473-2470

中国 - 青島 Tel:86-532-8502-7355 Fax:86-532-8502-7205

中国 - 上海

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

中国 - 瀋陽

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

中国 - 深圳

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

中国 - 武漢

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

中国 - 西安

Tel:86-29-8833-7252 Fax:86-29-8833-7256

中国 - 厦門

Tel:86-592-2388138 Fax:86-592-2388130

中国 - 珠海

Tel:86-756-3210040 Fax:86-756-3210049

アジア / 太平洋

インド - バンガロール Tel:91-80-3090-4444 Fax:91-80-3090-4123

インド - ニューデリー Tel:91-11-4160-8631 Fax:91-11-4160-8632

インド - プネ

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

日本 - 横浜

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

韓国 - 大邱

Tel:82-53-744-4301 Fax:82-53-744-4302

韓国 - ソウル

Tel:82-2-554-7200 Fax:82-2-558-5932 または 82-2-558-5934

マレーシア - クアラルンプー

Tel:60-3-6201-9857 Fax:60-3-6201-9859

マレーシア - ペナン Tel:60-4-227-8870

Fax:60-4-227-4068 フィリピン - マニラ

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

シンガポール Tel:65-6334-8870 Fax:65-6334-8850

台湾 - 新竹

Tel:886-3-6578-300 Fax:886-3-6578-370

台湾 - 高雄

Tel:886-7-213-7830 Fax:886-7-330-9305

台湾 - 台北

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

タイ・パンコク

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

ヨーロッパ

オーストリア - ヴェルス Tel:43-7242-2244-39 Fax:43-7242-2244-393

デンマーク - コペンハーゲン Tel:45-4450-2828 Fax:45-4485-2829

フランス - パリ

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

ドイツ - ミュンヘン Tel:49-89-627-144-0 Fax:49-89-627-144-44

イタリア - ミラノ Tel:39-0331-742611

Fax:39-0331-466781

オランダ・ドリューネン Tel:31-416-690399

Fax:31-416-690340

スペイン - マドリッド Tel:34-91-708-08-90 Fax:34-91-708-08-91

イギリス - ウォーキンガム Tel:44-118-921-5869 Fax:44-118-921-5820

05/02/11