



8ビット AVR マイクロ コントローラ

ATtiny102, ATtiny104

完全 データシート

序説

Atmel® ATtiny102/ATtiny104はAVR®強化RISC構造に基づく低電力CMOS 8ビット マイクロ コントローラです。単一クロック周期での強力な命令の実行により、ATtiny102/ATtiny104はMHz当たり1MIPS近くの単位処理量を達成します。これは消費電力対処理速度に対するデバイスの最適化をシステム設計者に許します。

本書は一般の方々の便宜のため有志により作成されたもので、Atmel社とは無関係であることを御承知ください。しおりの[はじめに]での内容にご注意ください。

特徴

- 高性能、低消費AVR® 8ビット マイクロ コントローラ
- 進化したRISC構造
 - 強力な112命令
 - 多くは1クロック周期実行
 - 16個の1バイト長汎用レジスタ
 - 完全なステイック動作
 - 12MHz時、最大12MIPSの単位処理量
- データ用メモリとプログラム用不揮発性メモリ
 - 実装書き換え可能な1Kバイト(512語)フラッシュメモリ内蔵
 - 32バイトの内蔵SRAM
 - フラッシュ書き込み/消去回数: 10,000
 - データ保持力: 20年/85°C, 100年/25°C
 - 全動作電圧範囲(1.8~5.5V)でのフラッシュメモリ自己プログラミング
- 内蔵周辺機能
 - 前置分周器、捕獲入力、2つの比較出力、2つのPWMチャネルを持つ1つの16ビットタイマ/カウンタ(TC0)
 - 独立したチップ上発振器を持つ設定可能なウォッチドッグタイマ(WDT)
 - 選択可能な内部基準電圧: 1.1V, 2.2V, 4.3V
 - 5(ATtiny102)/8(ATtiny104)チャネルを持つ10ビットA/D変換器(ADC)
 - チップ上のアナログ比較器(AC)
 - 直列通信部: USART
- 特殊マイクロコントローラ機能
 - 実装書き込み可能 (5Vでのみ)
 - 外部プログラミング (2.7~5.5V)
 - 自己プログラミング (1.8~5.5V)
 - 内部及び外部の割り込み
 - アイドル、A/D変換雑音低減、スタンバイ、パワーダウンの4つの低消費動作
 - 強化した電源ONリセット回路
 - 割り込みとリセットを持つ設定可能な供給電圧レベル監視器
 - 正確な校正付き内蔵RC発振器
 - 高速と標準の始動時間任意選択が利用可能
 - 固有IDを表すための個別通番
- I/Oと外圍器
 - 6(ATtiny102)/12(ATtiny104)ビットの設定変更可能なI/O
 - 8ビットUDFN (ATtiny102)
 - 8リードSOIC150 (ATtiny102)
 - 14リードSOIC150 (ATtiny104)
- 動作電圧
 - 1.8~5.5V
- 温度範囲
 - -40~+125°C
- 動作速度
 - 0~4MHz/1.8~5.5V
 - 0~8MHz/2.7~5.5V
 - 0~12MHz/4.5~5.5V

目次

| | | | |
|----------------------------------|----|--------------------------------|----|
| 序説 | 1 | 13.2. 割り込みベクタ | 32 |
| 特徴 | 2 | 13.3. 外部割り込み | 33 |
| 1. 概要 | 5 | 13.4. 割り込み用レジスタ | 34 |
| 2. 製品形態要約 | 5 | 14. 入出力ポート | 37 |
| 3. 注文情報 | 6 | 14.1. 概要 | 37 |
| 4. 構成図 | 6 | 14.2. 特徴 | 37 |
| 5. ピン配置 | 7 | 14.3. 入出力ピン等価回路 | 37 |
| 5.1. ピン説明 | 7 | 14.4. 標準デジタル入出力としてのポート | 38 |
| 6. 入出力多重化 | 8 | 14.5. 交換ポート機能 | 41 |
| 7. 全般情報 | 9 | 14.6. I/Oポート用レジスタ | 47 |
| 7.1. 資料 | 9 | 15. TC0 – PWM付き16ビット タイマ/カウンタ0 | 50 |
| 7.2. データ保持力 | 9 | 15.1. 概要 | 50 |
| 7.3. コード例について | 9 | 15.2. 特徴 | 50 |
| 8. AVR CPU コア | 10 | 15.3. 構成図 | 50 |
| 8.1. 概要 | 10 | 15.4. 定義 | 51 |
| 8.2. 特徴 | 10 | 15.5. 関係レジスタ | 51 |
| 8.3. 構成図 | 10 | 15.6. 16ビット タイマ/カウンタ レジスタのアクセス | 52 |
| 8.4. ALU (Arithmetic Logic Unit) | 11 | 15.7. タイマ/カウンタ クロック元 | 54 |
| 8.5. ステータス レジスタ | 11 | 15.8. 計数器部 | 55 |
| 8.6. 汎用レジスタ ファイル | 11 | 15.9. 捕獲入力部 | 56 |
| 8.7. Xレジスタ,Yレジスタ,Zレジスタ | 11 | 15.10. 比較出力部 | 57 |
| 8.8. スタック ポインタ | 11 | 15.11. 比較一致出力部 | 59 |
| 8.9. 16ビット レジスタのアクセス | 12 | 15.12. 動作種別 | 60 |
| 8.10. 命令実行タイミング | 12 | 15.13. タイマ/カウンタのタイミング | 64 |
| 8.11. リセットと割り込みの扱い | 13 | 15.14. 16ビット タイマ/カウンタ0用レジスタ | 65 |
| 8.12. コア関係レジスタ | 14 | 16. USART | 72 |
| 9. AVRのメモリ | 16 | 16.1. 概要 | 72 |
| 9.1. 概要 | 16 | 16.2. 特徴 | 72 |
| 9.2. 特徴 | 16 | 16.3. 構成図 | 72 |
| 9.3. 実装書き換え可能なプログラム用フラッシュ メモリ | 16 | 16.4. クロック生成 | 73 |
| 9.4. データ メモリ | 16 | 16.5. フレーム形式 | 74 |
| 9.5. I/Oメモリ (レジスタ) | 17 | 16.6. USARTの初期化 | 75 |
| 10. クロック体系 | 18 | 16.7. データ送信 – USART送信部 | 76 |
| 10.1. 概要 | 18 | 16.8. データ受信 – USART受信部 | 77 |
| 10.2. クロック分配 | 18 | 16.9. 非同期受信 | 80 |
| 10.3. クロック副系統 | 18 | 16.10. 複数プロセッサ通信動作 | 82 |
| 10.4. クロック元 | 18 | 16.11. ホーレート設定例 | 83 |
| 10.5. システム クロック前置分周器 | 19 | 16.12. USART用レジスタ | 85 |
| 10.6. 始動 | 20 | 17. USARTSPI – USARTでのSPI動作 | 90 |
| 10.7. クロック関係レジスタ | 21 | 17.1. 概要 | 90 |
| 11. 電力管理と休止形態動作 | 23 | 17.2. 特徴 | 90 |
| 11.1. 概要 | 23 | 17.3. クロック生成 | 90 |
| 11.2. 特徴 | 23 | 17.4. SPIデータ形態とタイミング | 90 |
| 11.3. 休止形態動作種別 | 23 | 17.5. フレーム形式 | 91 |
| 11.4. 電力削減レジスタ | 24 | 17.6. データ転送 | 92 |
| 11.5. 消費電力の最小化 | 24 | 17.7. USARTでのMSPIMとSPIの比較 | 93 |
| 11.6. 電力管理用レジスタ | 26 | 17.8. MSPIMでのUSART用レジスタ | 93 |
| 12. SCRST – システム制御とリセット | 27 | 18. AC – アナログ比較器 | 94 |
| 12.1. 概要 | 27 | 18.1. 概要 | 94 |
| 12.2. 特徴 | 27 | 18.2. 特徴 | 94 |
| 12.3. AVRのリセット | 27 | 18.3. 構成図 | 94 |
| 12.4. リセット元 | 27 | 18.4. アナログ比較器用レジスタ | 95 |
| 12.5. ウォッチドッグ タイマ | 28 | 19. ADC – A/D変換器 | 97 |
| 12.6. リセット関係レジスタ | 30 | 19.1. 概要 | 97 |
| 13. 割り込み | 32 | 19.2. 特徴 | 97 |
| 13.1. 概要 | 32 | 19.3. 構成図 | 97 |
| | | 19.4. 操作 | 98 |
| | | 19.5. 変換の開始 | 98 |

| | | |
|--------|----------------------------|-----|
| 19.6. | 前置分周と変換タイミング | 99 |
| 19.7. | チャネル変更と基準電圧選択 | 100 |
| 19.8. | 雑音低減機能 | 101 |
| 19.9. | アナログ入力回路 | 101 |
| 19.10. | アナログ雑音低減技術 | 101 |
| 19.11. | A/D変換の精度定義 | 102 |
| 19.12. | A/D変換の結果 | 102 |
| 19.13. | A/D変換用レジスタ | 103 |
| 20. | TPI – Tinyプログラミング インターフェース | 107 |
| 20.1. | 概要 | 107 |
| 20.2. | 特徴 | 107 |
| 20.3. | 構成図 | 107 |
| 20.4. | Tinyプログラミング インターフェースの物理層 | 107 |
| 20.5. | Tinyプログラミング インターフェースのアクセス層 | 110 |
| 20.6. | 命令一式 | 110 |
| 20.7. | 不揮発性メモリ制御器のアクセス | 112 |
| 20.8. | 制御/状態空間レジスタ説明 | 112 |
| 21. | MEMPROG – メモリ プログラミング | 114 |
| 21.1. | 概要 | 114 |
| 21.2. | 特徴 | 114 |
| 21.3. | 不揮発性メモリ (NVM) | 114 |
| 21.4. | NVMのアクセス | 117 |
| 21.5. | 自己プログラミング | 119 |
| 21.6. | 外部プログラミング | 120 |
| 21.7. | NVMプログラミング用レジスタ | 121 |
| 22. | 電気的特性 | 122 |
| 22.1. | 絶対最大定格 | 122 |
| 22.2. | DC特性 | 122 |
| 22.3. | 速度 | 123 |
| 22.4. | クロック特性 | 123 |
| 22.5. | システムとリセットの特性 | 124 |
| 22.6. | アナログ比較器特性 | 124 |
| 22.7. | A/D変換器特性 | 125 |
| 22.8. | 直列プログラミング特性 | 125 |
| 23. | 代表特性 | 126 |
| 23.1. | 活動動作消費電流 | 126 |
| 23.2. | アイドル動作消費電流 | 128 |
| 23.3. | 周辺機能部供給電流 | 129 |
| 23.4. | パワーダウン動作消費電流 | 129 |
| 23.5. | ピン駆動能力 | 130 |
| 23.6. | ピン 閾値とヒステリシス | 133 |
| 23.7. | アナログ比較器オフセット | 136 |
| 23.8. | ピン プルアップ | 136 |
| 23.9. | 内部発振器周波数 | 138 |
| 23.10. | VLM閾値 | 140 |
| 23.11. | 周辺機能部消費電流 | 141 |
| 23.12. | リセット消費電流とリセット パルス幅 | 143 |
| 24. | レジスタ要約 | 144 |
| 25. | 命令要約 | 145 |
| 26. | 外圍器情報 | 147 |
| 27. | 障害情報 | 148 |
| 28. | データシート改訂履歴 | 149 |

1. 概要

Atmel® AVR®コアは豊富な命令一式を16個の汎用作業レジスタに結合します。16個のレジスタの全ては算術論理演算部(ALU:Arithmetic Logic Unit)に直結され、1クロック周期で実行される単一命令でのアクセスを2つの独立したレジスタに許します。その結果としての基本構造は簡潔且つコード効率的で、同時に伝統的なCISC型マイクロコントローラに対して10倍程度までの単位処理量を達成します。

本デバイスは書き込み中の読み込み能力を持つ1024バイトの実装書き込み可能なフラッシュメモリ、32バイトのSRAM、6(ATtiny102)/12(ATtiny104)本の汎用入出力線、16個の汎用作業レジスタ、比較動作を持つ1つの16ビットタイマ/カウンタ(TC)、内部及び外部の割り込み、設定可能な1つの直列USART、5(ATtiny102)/8(ATtiny104)チャネルの10ビットA/D変換器(ADC)、内蔵発振器付きの設定変更可能なウォッチドッグタイマ、ソフトウェアで選択できる4つの節電動作を提供します。

アイドル動作はCPUを停止する一方でSRAM、TC、ADC、アナログ比較器(AC)、割り込み機構に機能の継続を許します。A/D変換雑音低減動作はADCを除く全入出力部とCPUを停止することによってA/D変換中の切り替え雑音を最小にします。パワーダウン動作はレジスタ内容を保ちますが、次の割り込みまたはハードウェアリセットまで、発振器を停止して他の全てのチップ機能を禁止します。スタンバイ動作ではデバイスの残りが休止する一方で発振器が走行し、低消費電力と組み合わせられた非常に速い始動を許します。

本デバイスはAtmelの高密度不揮発性メモリ技術を使って製造されます。チップ上の実装書き換え可能なフラッシュメモリは伝統的な不揮発性メモリ書き込み器による実装書き換えをプログラム用メモリに許します。

本デバイスはCコンパイラ、マクロアセンブラ、プログラムデバッガ/シミュレータ、インサーキットエミュレータ、評価キットを含む専用のプログラム及びシステム開発ツールで支援されます。

2. 製品形態要約

| 機能項目 | ATtiny102 | ATtiny104 |
|----------------------|--------------------|----------------|
| ピン数 | 8 | 14 |
| フラッシュメモリ容量 (バイト) | 1024 | 1024 |
| SRAM容量 (バイト) | 32 | 32 |
| EEPROM容量 (バイト) | – | – |
| 最大汎用入出力ピン数 | 6 | 12 |
| USART | 1 | 1 |
| A/D変換器 (ADC) / チャネル数 | 10ビットADC/5チャネル | 10ビットADC/8チャネル |
| アナログ比較器 (AC) チャネル数 | 1 | 1 |
| AC伝搬遅延 | 75~750ns | 75~750ns |
| 16ビットタイマ/カウンタ (TC) 数 | 1 | 1 |
| PWMチャネル数 | 2 | 2 |
| RC発振器精度 | ±2% | ±2% |
| 内部基準電圧 | 1.1V/2.2V/4.3V | 1.1V/2.2V/4.3V |
| 動作電圧 | 1.8~5.5V | |
| 最大動作周波数 (MHz) | 12 | |
| 動作温度範囲 | -40~+125°C | |
| 外囲器 | 8ピンUDFN/8ピンSOIC150 | 14ピンSOIC150 |

3. 注文情報

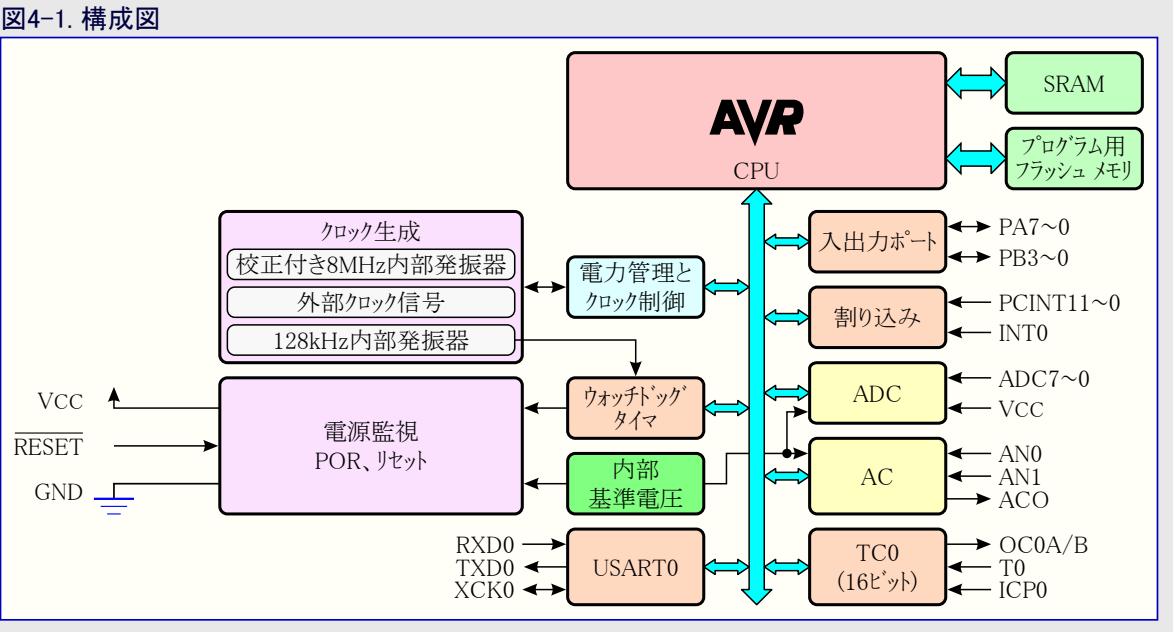
| デバイス | 速度(MHz) | 電源電圧 | 注文符号 | 外囲器 | 動作範囲 |
|-----------|---------|----------|---------------------|---------------|-----------------|
| ATtiny102 | 12 | 1.8～5.5V | ATtiny102-M7R | 8ﾊﾟｯﾄﾞUDFN | 工業用 (-40℃～105℃) |
| | | | ATtiny102F-M7R (注) | | |
| | | | ATtiny102-SSNR | 8ﾘｰﾄﾞSOIC150 | |
| | | | ATtiny102F-SSNR (注) | | |
| | | | ATtiny102-M8R | 8ﾊﾟｯﾄﾞUDFN | 工業用 (-40℃～125℃) |
| | | | ATtiny102F-M8R (注) | | |
| | | | ATtiny102-SSFR | 8ﾘｰﾄﾞSOIC150 | |
| | | | ATtiny102F-SSFR (注) | | |
| ATtiny104 | 12 | 1.8～5.5V | ATtiny104-SSNR | 14ﾘｰﾄﾞSOIC150 | 工業用 (-40℃～105℃) |
| | | | ATtiny104F-SSNR (注) | | 工業用 (-40℃～125℃) |
| | | | ATtiny104-SSFR | | |
| | | | ATtiny104F-SSFR (注) | | |

注: ATtiny102F-xxxとATtiny104F-xxxは高速始動時間任意選択を持ちます。

| 外囲器形式 | |
|--------------|---|
| 8ﾊﾟｯﾄﾞUDFN | 8ﾊﾟｯﾄﾞ 2×3×0.6mm本体 熱強化プラスチック極薄2列平板ﾘｰﾄﾞなし外囲器 (UDFN) |
| 8ﾋﾟﾝSOIC150 | 8ﾘｰﾄﾞ 0.150インチ幅本体 プラスティック ガルウイング小外形外囲器 (JEDEC SOIC) |
| 14ﾋﾟﾝSOIC150 | 14ﾘｰﾄﾞ 1.27mmﾋﾟｯﾁ 8.65×3.90×1.60mm本体 プラスティック小外形外囲器 (SOIC) |

関連リンク 20頁の「リセットからの始動」

4. 構成図



5. ピン配置

図5-1. ピン配置

| 8UDFN・8SOIC150 | | | | | | | | | |
|-----------------------------------|-----|---|----|--|--|--|--|--|---------------------------------------|
| | Vcc | 1 | 8 | | | | | | GND |
| (PCINT0/TPICLK/CLKI/T0/ADC0/AIN0) | PA0 | 2 | 7 | | | | | | PB3 (ADC7/ACO/T0*/RXD0/PCINT11) |
| (PCINT1/TPIDATA/OC0B/ADC1/AIN1) | PA1 | 3 | 6 | | | | | | PB2 (ADC6/ICP0/TXD0/PCINT10) |
| (PCINT2/RESET) | PA2 | 4 | 5 | | | | | | PB1 (ADC5/INT0/OC0A/XCK0/CLKO/PCINT9) |
| 14SOIC150 | | | | | | | | | |
| | Vcc | 1 | 14 | | | | | | GND |
| (PCINT0/TPICLK/CLKI/T0/ADC0/AIN0) | PA0 | 2 | 13 | | | | | | PB3 (ADC7/ACO/T0*/RXD0/PCINT11) |
| (PCINT1/TPIDATA/OC0B/ADC1/AIN1) | PA1 | 3 | 12 | | | | | | PB2 (ADC6/ICP0/TXD0/PCINT10) |
| (PCINT2/RESET) | PA2 | 4 | 11 | | | | | | PB1 (ADC5/INT0/OC0A/XCK0/CLKO/PCINT9) |
| (PCINT3/OC0A*) | PA3 | 5 | 10 | | | | | | PB0 (ADC4/PCINT8) |
| (PCINT4/ICP0*) | PA4 | 6 | 9 | | | | | | PA7 (PCINT7) |
| (PCINT5/OC0B*/ADC2) | PA5 | 7 | 8 | | | | | | PA6 (ADC3/PCINT6) |

注: T0*,ICP0*,OC0A*,OC0B*は再割り当て(GTCCRのREMAP=1)時の各々T0,ICP0,OC0A,OC0B

■ 電源 ■ 接地 □ デジタル機能 □ アナログ機能 □ クロック □ プログラミング



警告: UDFN外囲器裏面の放熱パッドは接続されるべきではありません。

5.1. ピン概要

5.1.1. VCC

電源ピン。

5.1.2. GND

接地ピン。

5.1.3. PA7～PA0 (ポートA)

これはビット毎に個別選択可能な内蔵プルアップ抵抗を持つ3(ATtiny102)/8(ATtiny104)ビット双方向入出力ポートです。出力緩衝部は共に高い吐き出し/吸い込み能力を持つ対称駆動特性を持ちます。入力時に外部的にLowへ引き込まれたポートピンはプルアップ抵抗が有効の場合に吐き出し電流が流れます。リセット条件が有効になると、例えクロックが動いていなくても、ポートピンはHi-Zにされます。

5.1.4. PB3～PB0 (ポートB)

これはビット毎に個別選択可能な内蔵プルアップ抵抗を持つ3(ATtiny102)/4(ATtiny104)ビット双方向入出力ポートです。出力緩衝部は共に高い吐き出し/吸い込み能力を持つ対称駆動特性を持ちます。入力時に外部的にLowへ引き込まれたポートピンはプルアップ抵抗が有効の場合に吐き出し電流が流れます。リセット条件が有効になると、例えクロックが動いていなくても、ポートピンはHi-Zにされます。

5.1.5. RESET

リセット入力。RESETピンが禁止されていなければ、例えクロックが走行していなくても、最小パルス幅より長いこのピンのLowレベルはリセットを生成します。最小パルス幅は「電気的特性」の「システムとリセットの特性」で与えられます。より短いパルスはリセットの生成が保証されません。

RESETピンは((駆動能力の)弱い)入出力ピンとしても使えます。

関連リンク [124頁の「システムとリセットの特性」](#)

6. 入出力多重化

各ピンは既定によって汎用入出力としてポートによって制御され、代わりに周辺機能の1つに割り当てすることもできます。下表はポート入出力ピンに多重化される周辺機能信号を記述します。

表6-1. ポート機能多重化

| ピン番号 | | ピン名 | 特殊機能 | 割り込み (注3) | ADC (注3) | AC | USART | TC | プログラミング (注8) |
|-----------|-----------|----------|-------|--------------|-------------|------|-------|-----------|-----------------|
| ATtiny102 | ATtiny104 | | | | | | | | |
| 1 (注1) | 1 | VCC | | | | | | | |
| 2 | 2 | PA0 (注2) | CLKI | PCINT0 | ADC0 | AIN0 | | T0 | TPICLK |
| 3 | 3 | PA1 (注5) | | PCINT1 | ADC1 | AIN1 | | OC0B | TPIDATA |
| 4 | 4 | PA2 | RESET | PCINT2 | | | | | RESET |
| – | 5 | PA3 | | PCINT3 | | | | OC0A (注9) | |
| – | 6 | PA4 | | PCINT4 | | | | ICP0 (注9) | |
| – | 7 | PA5 (注5) | | PCINT5 | ADC2 | | | OC0B (注9) | |
| – | 8 | PA6 | | PCINT6 | ADC3 | | | | |
| – | 9 | PA7 | | PCINT7 | | | | | |
| – | 10 | PB0 | | PCINT8 | ADC4 | | | | |
| 5 | 11 | PB1 (注6) | CLKO | INT0/PCINT9 | ADC5 | | XCK0 | OC0A | |
| 6 | 12 | PB2 (注7) | | PCINT10 | ADC6 | | TXD0 | ICP0 | |
| 7 | 13 | PB3 (注4) | | PCINT11 | ADC7 | ACO | RXD0 | T0 (注9) | |
| 8 | 14 | GND | | | | | | | |

注1: 8ピンUDFN外周器では放熱パッドもVCCで、これは接続されるべきではありません。

注2: CLKIの優先権はADC0よりも高くなります。外部クロック駆動(EXT_CLK)が許可されると、ADCチャネルは働かず、デジタル入力禁止レジスタ0(DIDR0)はデジタル入力緩衝部を禁止しません。

注3: PCINTと対応するADCチャネルの両方が許可されると、デジタル入力緩衝部は禁止されません。

注4: ACOが許可される時に、ADC、AC、USARTのRX入力は禁止されません。

注5: OC0Bが許可される時に、ADCとACは許可されていればそのチャネル上の入力を受け取り続けます。

注6: CLKOが許可されると、PB1のOC0Aはより低い優先権になります。

注7: USARTが許可される時に、使用者はTXD0ピンに対応するADCチャネルを使わないことを確実にしなければなりません。なぜならば、DIDR0レジスタは出力部ではなく入力緩衝部を制御するだけだからです。

注8: リセット/外部プログラミング中、全てのピンは入力として扱われ、出力は禁止されます。

注9: タイマ/カウンタの再割り当て許可(GTCCRのREMAP=1)時の代替位置です。

7. 全般情報

7.1. 資料

包括的なデータシート、応用記述、ドライバ群と開発ツールの説明は <http://www.atmel.com/avr> でのダウンロードで利用可能です。

7.2. データ保持力

信頼性証明結果はデータ保持誤り率の反映を示し、20年以上/85°Cまたは100年以上/25°Cで1PPMよりずっと小さな値です。

7.3. コード例について

この資料はデバイスの様々な部分の使用法を手短に示す簡単なコード例を含みます。これらのコード例はアセンブルまたはコンパイルに先立って、デバイス定義ヘッダファイルがインクルードされると仮定します。全てのCコンパイラ製造業者がヘッダファイル内にビット定義を含めるとは限らず、またCでの割り込みの扱いがコンパイラに依存することに注意してください。より多くの詳細についてはCコンパイラの資料で確認してください。

関連リンク [13頁の「リセットと割り込みの扱い」](#)
[29頁の「コード例」](#)

8. AVR CPU コア

8.1. 概要

Atmel® AVR®コアは豊富な命令一式を16個の汎用作業レジスタに結合します。16個のレジスタの全ては算術論理演算部(ALU:Arithmetic Logic Unit)に直結され、1クロック周期で実行される単一命令でのアクセスを2つの独立したレジスタに許します。その結果としての基本構造は簡潔且つコスト効率的で、同時に伝統的なCISC型マイクロコントローラに対して10倍程度までの単位処理量を達成します。

8.2. 特徴

- ・ 進化したRISC構造
- ・ 強力な112命令
- ・ 多くは1クロック周期実行
- ・ 16個の1バイト長汎用レジスタ
- ・ 完全なスタティック動作
- ・ 12MHz時、12MIPSの単位処理量

8.3. 構成図

本項はAVRコア構造を一般的に説明します。このCPUコアの主な機能は正しいプログラム実行を保証することです。従ってCPUはメモリアクセス、計算実行、周辺制御、割り込み操作ができなければなりません。

最大効率と平行処理のため、AVRはプログラムとデータに対してメモリとバスを分離するハーバード構造を使います。プログラムメモリ内の命令は単一段のパイプラインで実行されます。1命令の実行中に次の命令がプログラムメモリから事前取得されます。この概念は全部のクロック周期で命令実行を可能にします。プログラムメモリは実装書き換え可能なフラッシュメモリです。

高速レジスタファイルは1クロック周期アクセスの16個の8ビット長汎用レジスタを含みます。これは1クロック周期ALU(Arithmetic Logic Unit)操作を許します。代表的なALU操作では2つのオペランドがレジスタファイルからの出力で、1クロック周期内でその操作が実行されてその結果がレジスタファイルに書き戻されます。

16個中の6つのレジスタは効率的なアドレス計算ができるデータ空間アドレス指定用に3つの16ビット長間接アドレスポインタ用レジスタとして使われます。これらアドレスポインタの1つはプログラム用フラッシュメモリ内の定数表参照用アドレスポインタとしても使えます。これら16ビット長付加機能レジスタはX,Y,Zレジスタで、本項内で後述されます。

ALUはレジスタ間またはレジスタと定数間の算術及び論理操作を支援します。単一レジスタ操作もALUで実行できます。算術演算操作後、操作結果についての情報を反映するために、ステータスレジスタ(SREG)が更新されます。

プログラムの流れは条件/無条件分岐や呼び出し命令によって提供され、全アドレス空間を直接アドレス指定する能力があります。殆どのAVR命令は単一16ビット語(ワード)形式ですが、32ビット幅の命令も存在します。実際の命令一式はいくつかのデバイスが(全)命令一式の部分だけを実装するように変わります。

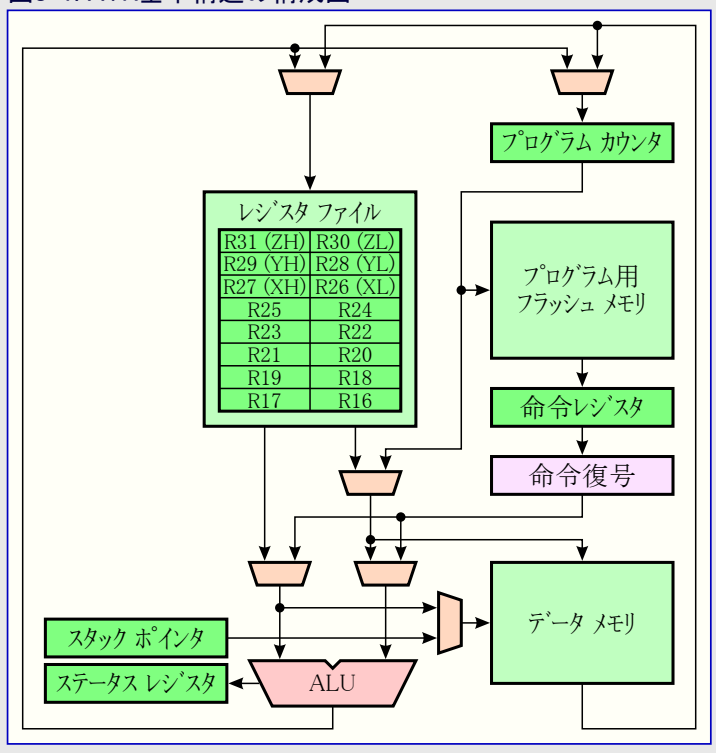
割り込みやサブルーチン呼び出し中、戻りアドレスを示すプログラムカウンタ(PC)はスタックに保存されます。スタックは一般的なデータ用SRAM上に実際には割り当てられ、従ってスタック容量はSRAM容量とSRAM使用量でのみ制限されます。全てのユーザープログラムはリセット処理ルーチンで(サブルーチン呼び出しや割り込みが実行される前に)、スタックポインタ(SP)を初期化しなければなりません。SPはI/O空間で読み書きアクセスが可能です。データ用SRAMはAVR構造で支援される4つの異なるアドレス指定種別を通して容易にアクセスできます。

AVR構造に於けるメモリ空間は全て直線的な普通のメモリ配置です。

柔軟な割り込み部にはI/O空間の各制御レジスタとステータスレジスタ(SREG)の特別な全割り込み許可(I)ビットがあります。全ての割り込みは割り込みベクタ表に個別の割り込みベクタを持ちます。割り込みには割り込みベクタ表の位置に従う優先順があります。下位側割り込みベクタアドレスが高い優先順位です。

I/Oメモリ空間は制御レジスタや他のI/O機能としてCPU周辺機能用の64アドレスを含みます。I/Oメモリはデータ空間位置\$0000~\$003Fとしてアクセスできます。詳細記述については「命令要約」章をご覧ください。

図8-1. AVR基本構造の構成図



8.4. ALU (Arithmetic Logic Unit)

高性能なAVRのALUは16個全ての汎用レジスタに直接接続され動作します。汎用レジスタ間または汎用レジスタと即値間の演算操作は単一クロック周期内で実行されます。ALU操作は算術演算、論理演算、ビット操作の3つの主要な種類に大別されます。符号付きと符号なし両方の乗算と固定小数点形式を支援する乗算器(乗算命令)も提供する構造の実装(製品)もあります。詳細記述については「命令要約」章をご覧ください。

8.5. ステータス レジスタ

ステータスレジスタは最も直前に実行した演算命令の結果についての情報を含みます。この情報は条件処理を行うためのプログラムの流れ変更に使えます。ステータスレジスタは「AVR命令一式」資料で詳述したように、全てのALU操作後、更新されることに注目してください。これは多くの場合でそれ用の比較命令使用の必要をなくし、高速でより少ないコードに帰着します。

ステータスレジスタは割り込み処理ルーチン移行時の保存と割り込みからの復帰時の回復(復帰)が自動的に行われません。これはソフトウェアによって扱われなければなりません。

8.6. 汎用レジスタ ファイル

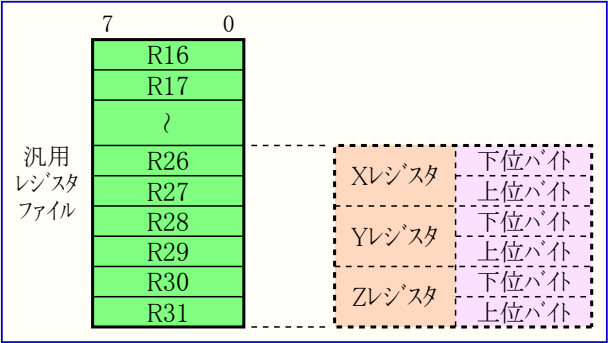
このレジスタファイルはAVRの増強したRISC命令群用に最適化されています。必要な効率と柔軟性を達成するために、次の入出力機構がレジスタファイルによって支援されます。

- 1つの8ビット出力オペランドと1つの8ビットの結果入力
- 2つの8ビット出力オペランドと1つの8ビットの結果入力
- 1つの16ビット出力オペランドと1つの16ビットの結果入力

注: AVRレジスタファイルの代表的な実装は32個の汎用レジスタを含みますが、ATtiny102/ATtiny104は16個のレジスタだけを実装します。互換性のため、レジスタはR0〜R15ではなく、R16〜R31で番号付けされます。

レジスタファイルを操作する殆どの命令は全てのレジスタに直接アクセスし、それらの殆どは単一クロック周期命令です。

図8-2. AVR CPU 汎用レジスタ構成図



8.7. Xレジスタ, Yレジスタ, Zレジスタ

R26〜R31レジスタには通常用途の使用にいくつかの追加機能があります。これらのレジスタはデータ空間の間接アドレス指定用の16ビットアドレスポインタです。3つのX,Y,Z間接アドレスレジスタは図で記載したように定義されます。

種々のアドレス指定種別で、これらのアドレスレジスタは自動増加、自動減少としての機能を持ちます(詳細については「命令要約」をご覧ください)。

図8-3. X,Y,Zレジスタ構成図

| | | | | | |
|--------|----|------------|---|---------|------------|
| | 15 | XH (上位) | | XL (下位) | 0 |
| X レジスタ | 7 | R27 (\$1B) | 0 | 7 | R26 (\$1A) |
| | 15 | YH (上位) | | YL (下位) | 0 |
| Y レジスタ | 7 | R29 (\$1D) | 0 | 7 | R28 (\$1C) |
| | 15 | ZH (上位) | | ZL (下位) | 0 |
| Z レジスタ | 7 | R31 (\$1F) | 0 | 7 | R30 (\$1E) |

8.8. スタック ポインタ

スタックは主に一時データの保存、局所変数の保存、割り込みとサブルーチン呼び出し後の戻りアドレスの保存に使われます。スタックは高位メモリから低位メモリへ伸長するように実行されます。スタックポインタレジスタは常にこのスタックの先頭(注:次に使われるべき位置)を指し示し、スタックポインタは\$40以上を指示するように設定されなければなりません。

スタックポインタはサブルーチンや割り込みのスタックが配置されるデータSRAMのスタック領域を指し示します。スタックPUSH命令はスタックポインタを減らします。データSRAM内のスタック空間はサブルーチン呼び出しの実行や割り込みの許可の何れにも先立ってプログラムによって定義されなければなりません。初期スタックポインタ値は内部SRAMの最終アドレスに等しく、スタックポインタはSRAMの先頭以上に設定されなければなりません。スタックポインタの詳細については表をご覧ください。

表8-1. スタックポインタ命令

| 命令 | スタックポインタ | 内容 |
|-------------|----------|---|
| PUSH | -1 | データがスタック上に押し込まれます。 |
| ICALL,RCALL | -2 | サブルーチン呼び出しまたは割り込みでの戻りアドレスがスタック上に押し込まれます。 |
| POP | +1 | データがスタックから引き出されます。 |
| RET,RETI | +2 | サブルーチンまたは割り込みからの復帰での戻りアドレスがスタックから引き出されます。 |

AVRのスタックポインタはI/O空間内の2つの8ビットレジスタとして実装されます。実際に使われるビット数は(そのデバイス)実装に依存します。SPLだけが必要とされる程に小さいAVR構造の実装(デバイス)のデータ空間もあることに注意してください。その場合、SPHレジスタは存在しません。

関連リンク 14頁の「SPH,SPL – スタックポインタレジスタ上位/下位バイト」
16頁の「データメモリ」

8.9. 16ビット レジスタのアクセス (脚注参照)

AVRのデータバスは8ビット幅で、故に16ビットレジスタのアクセスは非分断操作が必要です。これらのレジスタは2つの読みまたは書きの操作を用いてバイトアクセスされなければなりません。16ビットレジスタは16ビットバスを用いて8ビットバスと1つの一時レジスタに接続されます。

書き込み操作については、16ビットレジスタの上位バイトが下位バイトに先立って書かれなければなりません。そしてその上位バイトは一時レジスタに書かれます。16ビットレジスタの下位バイトが書かれる時に、同じクロック周期で一時レジスタが16ビットレジスタの上位バイトに複写されます。

読み込み操作については、16ビットレジスタの下位バイトが上位バイトに先立って読まれなければなりません。CPUによって下位バイトが読まれる時に、下位バイトが読まれるのと同じクロック周期で16ビットレジスタの上位バイトが一時レジスタに複写されます。その後上位バイトが読まれる時は、この一時レジスタから読まれます。

これはレジスタの読み書き時に16ビットレジスタの上位と下位のバイトが常に同時にアクセスされることを保証します。

非分断16ビットレジスタ読み書き操作中に割り込みが起動されて同じレジスタをアクセスする場合、この時間制限手順を不正にし得ます。これを防ぐために、16ビットレジスタの読み書き時に割り込みを禁止することができます。

8.10. 命令実行タイミング

本項は命令実行の一般的なアクセスタイミングの概念を記述します。AVR CPUはチップ(デバイス)用に選択したクロック元から直接的に生成したCPUクロック(clkCPU)によって駆動されます。内部クロック分周は使われません。

右図はハーバード構造と高速アクセスレジスタファイルの概念によって可能とされる並列の命令取得と命令実行を示します。これは機能対費用、機能対クロック、機能対電源部に関する好結果と対応するMHzあたり1MIPSを達成するための基本的なパイプラインの概念です。

右図はレジスタファイルに対する内部タイミングの概念を示します。単一クロック周期で2つのレジスタオペランドを使うALU操作が実行され、その結果が転送先レジスタへ書き戻されます。

図8-4. 命令の取得と実行の並列動作

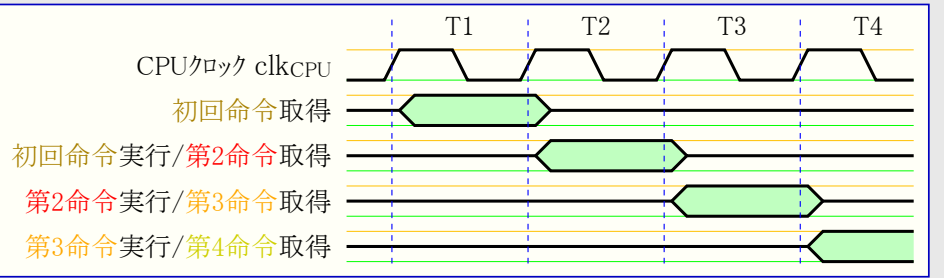
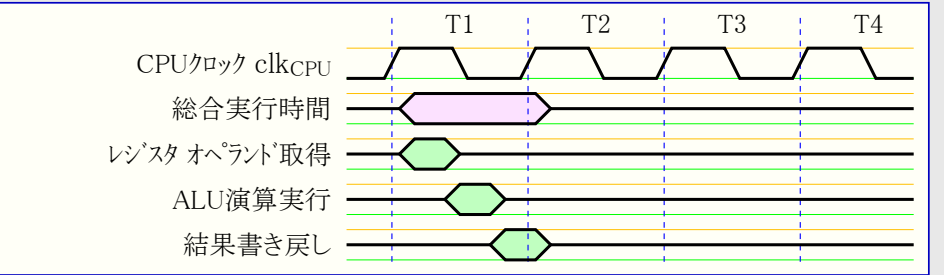


図8-5. 1クロック周期ALU命令



(訳注) 原書の本項の記述はXMEGA用の記述をそのまま複写したものとと思われます。そのため、書き込み操作の上位と下位のバイトの書き順が他のAVRと逆になっています。これによって、書き込み操作に関する他の殆どの部分での記述と矛盾が生じています。本書ではこの書き込み操作に関する書き順を矛盾が生じないように、本項と関連する他の一部の記述を修正しています。

8.11. リセットと割り込みの扱い

AVRは多くの異なる割り込み元を提供します。これらの割り込みと独立したリセット ベクタ各々はプログラム メモリ空間内に独立したプログラム ベクタを持ちます。全ての割り込みは割り込みを許可するために、**ステータスレジスタ(SREG)**の**全割り込み許可(I)ビット**と共に論理**1**が書かれなければならない個別の許可ビットを割り当てられます。

既定でのプログラム メモリ空間の最下位アドレスはリセットと割り込みのベクタとして定義されます。ベクタの完全な一覧は「**割り込み**」で示されます。この一覧は各種割り込みの優先順位も決めます。下位側アドレスがより高い優先順位です。リセットが最高優先順位で、次が外部割り込み要求0(INT0)です。

割り込みが起こると全割り込み許可(I)ビットが解除(0)され、全ての割り込みは禁止されます。使用者ソフトウェアは多重割り込みを許可するため、全割り込み許可(I)ビットへ論理**1**を書けます。その後全ての許可した割り込みが現在の割り込みルーチンで割り込めます。全割り込み許可(I)ビットは割り込みからの復帰(**RETI**)命令が実行されると、自動的に設定(**1**)されます。

根本的に2つの割り込み形式があります。

- 1つ目の形式は割り込み要求フラグを設定(**1**)する事象によって起動されます。これらの割り込みでは割り込み処理ルーチンを実行するために、プログラム カウンタは対応する現実の割り込みベクタを指示し、ハードウェアが対応する割り込み要求フラグを解除(0)します。割り込み要求フラグは解除(0)されるべきフラグのビット位置へ論理**1**を書くことによっても解除(0)できます。対応する割り込み許可ビットが解除(0)されている間に割り込み条件が起こると、割り込み要求フラグが設定(**1**)されて割り込みが許可されるか、またはこのフラグがソフトウェアによって解除(0)されるまで記憶(保持)されます。同様に、全割り込み許可(I)ビットが解除(0)されている間に1つまたはより多くの割り込み条件が起こると、対応する割り込み要求フラグが設定(**1**)されて全割り込み許可(I)ビットが設定(**1**)されるまで記憶され、その(I**1**)後で優先順に従って実行されます。
- 2つ目の割り込み形式は割り込み条件が存在する限り起動し(続け)ます。これらの割り込みは必ずしも割り込み要求フラグを持っているとは限りません。割り込みが許可される前に割り込み条件が消滅すると、この割り込みは起動されません。

AVRが割り込みから抜け出すと常に主プログラムへ戻り、何れかの保留割り込みが扱われる前に1つ以上の命令を実行します。

ステータスレジスタ(SREG)は割り込みルーチンへ移行時の保存も、復帰時の再設定も自動的に行われません。これはソフトウェアによって扱われなければならない。

割り込みを禁止する**CLI**命令を使うと、割り込みは直ちに禁止されます。**CLI**命令と同時に割り込みが起こっても、**CLI**命令後に割り込みは実行されません。

割り込みを許可する**SEI**命令を使うと、本例で示されるようにどの保留割り込みにも先立ち、**SEI**命令の次の命令が実行されます。

アセンブリ言語プログラム例

| | |
|-------|-------------------------|
| SEI | ; 全割り込み許可 |
| SLEEP | ; 休止形態動作移行 (移行後に割り込み待ち) |

注:「コード例について」をご覧ください。

関連リンク [32頁の「割り込み」](#)

8.11.1. 割り込み応答時間

許可した全てのAVR割り込みに対する割り込み実行応答は最小4クロック周期です。4クロック周期後、実際の割り込み処理ルーチンに対する**プログラム ベクタ アドレス**が実行されます。この4クロック周期間にプログラム カウンタ(PC)がスタック上に保存(プッシュ)されます。このベクタは標準的に割り込み処理ルーチンへの無条件分岐で、この分岐は2クロック周期(**訳注:**原文は3(**JMP**命令=3を想定、実際は**RJMP**命令=2))要します。複数周期命令実行中に割り込みが起こると、その割り込みが扱われる前に、この命令が完了されます。MCUが**休止形態動作**の時に割り込みが起こると、割り込み実行応答時間は4クロック周期、増やされます。この増加は選択した休止形態動作からの起動時間に加えてです。

割り込み処理ルーチンからの復帰は4クロック周期要します。これらの4クロック周期中、プログラム カウンタ(PC:2バイト)がスタックから取り戻され(ポップ)、スタック ポインタは増加され(+2)、**ステータスレジスタ(SREG)**の**全割り込み許可(I)ビット**が設定(**1**)されます。

8.12. コア関係レジスタ

8.12.1. CCP – 構成設定変更保護レジスタ (Configuration Change Protection Register)

名称 : CCP
変位 : \$3C
リセット : \$00
特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|--------|---|---|---|---|---|-----|-----|
| | CCP7~0 | | | | | | | |
| アクセス種別 | W | W | W | W | W | W | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

●ビット7~0 – CCP7~0 : 構成設定変更保護 (Configuration Change Protection)

保護されたI/Oレジスタの内容を変更するため、CCPレジスタは最初に正しい識票を書かれなければなりません。CCPが書かれた後、保護されたI/Oレジスタは次からの4CPU命令周期中に書けます。これらの周期中は全ての割り込みが無視されます。これらの周期後に割り込みはCPUによって自動的に再び取り扱われて、どの保留割り込みもそれらの優先権に従って実行されます。

保護されたI/Oレジスタの識票が書かれると、CCP0は保護機能が許可されている限り1として読めます。CCP7~2は常に0として読めます。

NVM自己プログラミング識票が書かれると、CCP1は4 CPU命令周期間、1として読み、他のビットは0として読み、CCP1は4周期後、自動的に解除(0)されます。ソフトウェアは自己プログラミングを実行するために、この4クロック周期内にフラッシュメモリの上位バイトにデータを書くべきです。

注: ビット0と1は読み書きアクセスです。他のビットは書き込みアクセス専用です。

表8-2. 構成設定変更保護レジスタによって認証される識票

| 識票 | 適用群 | 説明 |
|------|--------------------------------|--------------|
| \$D8 | IOREG: CLKMSR, CLKPSR, WDTC SR | 保護されたI/Oレジスタ |
| \$E7 | SPM | NVM自己プログラム許可 |

8.12.2. SPH,SPL – スタックポインタレジスタ上位/下位バイト (Stack Pointer Register Low and High byte)

SPHとSPLのレジスタ対は16ビット値のSPを表します。下位バイト[7~0](接尾辞L)は変位原点でアクセスできます。上位バイト[15~8](接尾辞H)は変位+1でアクセスすることができます。16ビットレジスタの読み書きのより多くの詳細については「16ビットレジスタのアクセス」をご覧ください。

I/O特定命令のINとOUTを使う時はI/Oアドレスの\$00~\$3Fが使われなければなりません。LD系とST系の命令を使ってデータ空間としてI/Oレジスタをアクセスする時はこれらの変位アドレスに\$20が加算されなければなりません。

名称 : SPHとSPL
変位 : \$3D
リセット : RAMEND (\$005F)
特質 : -

| ビット | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|--------|----|-------|-----|-----|-----|-----|-----|-----|
| | - | - | - | - | - | - | - | - |
| アクセス種別 | R | R | R | R | R | R | R | R |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | - | SP6~0 | | | | | | |
| アクセス種別 | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| リセット値 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |

●ビット6~0 – SP6~0 : スタックポインタ6~0 (Stack Pointer)

SPHとSPLはSPに組み合わされます。

8.12.3. SREG – ステータス レジスタ (Status Register)

I/O特定命令の**IN**と**OUT**を使う時はI/Oアドレスの\$00～\$3Fが使われなければなりません。**LD**系と**ST**系の命令を使ってデータ空間としてI/Oレジスタをアクセスする時はこれらの変位アドレスに\$20が加算されなければなりません。

名称 : SREG

変位 : \$3F

リセット : \$00

特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|-----|-----|-----|-----|-----|-----|-----|-----|
| | I | T | H | S | V | N | Z | C |
| アクセス種別 | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

● ビット7 – I : 全割り込み許可 (Global Interrupt Enable)

割り込みが許可されるには全割り込み許可ビットが設定(1)されなければなりません。その時の個別割り込み許可制御は独立した制御レジスタで行われます。全割り込み許可ビットが解除(0)されると、個別割り込み許可設定に拘らず、どの割り込みも許可されません。Iビットは割り込みが起こった後にハードウェアによって解除(0)され、後続の割り込みを許可するため、**RETI**命令によって設定(1)されます。Iビットは「**命令一式参考書**」で記述されるように**SEI**や**CLI**命令で応用(プログラム)によって設定(1)や解除(0)もできます。

● ビット6 – T : ビット変数 (Bit Copy Storage)

ビット複写命令、**BLD**(Bit Load)と**BST**(Bit Store)は操作したビットの転送元または転送先として、このTビットを使います。レジスタファイルのレジスタからのビットは**BST**命令によってTに複写でき、Tのビットは**BLD**命令によってレジスタファイルのレジスタ内のビットに複写できます。

● ビット5 – H : ハーフキャリー フラグ (Half Carry Flag)

ハーフキャリー(H)フラグはいくつかの算術操作でのハーフキャリーを示します。ハーフキャリーはBCD演算に有用です。詳細情報については「**命令要約**」記述をご覧ください。

● ビット4 – S : 符号 (Sign Bit, S= N Ex-OR V)

Sフラグは常に負(N)フラグと2の補数溢れ(V)フラグの排他的論理和です。詳細情報については「**命令要約**」記述をご覧ください。

● ビット3 – V : 2の補数溢れフラグ (2's Complement Overflow Flag)

2の補数溢れ(V)フラグは2の補数算術演算を支援します。詳細情報については「**命令要約**」記述をご覧ください。

● ビット2 – N : 負フラグ (Negative Flag)

負(N)フラグは算術及び論理操作での負の結果(MSB=1)を示します。詳細情報については「**命令要約**」記述をご覧ください。

● ビット1 – Z : ゼロフラグ (Zero Flag)

ゼロ(Z)フラグは算術及び論理操作でのゼロ(0)の結果を示します。詳細情報については「**命令要約**」記述をご覧ください。

● ビット0 – C : キャリー フラグ (Carry Flag)

キャリー(C)フラグは算術及び論理操作でのキャリー(または borrow)を示します。詳細情報については「**命令要約**」記述をご覧ください。

9. AVRのメモリ

9.1. 概要

本項は本デバイスの各種メモリを記述します。AVR構造にはプログラムメモリ空間とデータメモリ空間の2つの主なメモリ空間を持ちます。全てのメモリ空間は一般的な直線的アドレスです。

9.2. 特徴

- データ用メモリとプログラム用不揮発性メモリ
- 実装書き換え可能な1Kバイト(512語)フラッシュメモリ内蔵
- 32バイトの内蔵SRAM
- フラッシュ書き込み/消去回数: 10,000
- データ保持力: 20年/85°C, 100年/25°C
- 全動作電圧範囲(1.8~5.5V)でのフラッシュメモリ自己プログラミング

9.3. 実装書き換え可能なプログラム用フラッシュメモリ

ATtiny102/ATtiny104はプログラム保存用に実装書き換え可能な1Kバイトのフラッシュメモリをチップ上に含みます。全てのAVR命令が16または32ビット幅のため、このフラッシュメモリは512×16ビットとして構成されます。

フラッシュメモリは最低10,000回の消去/書き込み周回の耐久性があります。本デバイスのプログラムカウンタ(PC)は8/9ビット幅、従って\$0000から始まる256/512プログラムメモリ位置をアドレス指定する能力があります。「MEMPROG – メモリプログラミング」はフラッシュメモリの(TPI)直列プログラミングの詳細な記述を含みます。

定数表は読み書き命令を使うことによって全てのプログラムメモリアドレス空間に配置できます。プログラムメモリが直接アクセスできないため、これはデータメモリに割り当てられています。割り当てられたプログラムメモリはデータメモリのバイトアドレス\$4000から始まります。

命令の取得と実行のタイミング図は「命令実行タイミング」で示されます。

関連リンク [12頁の「命令実行タイミング」](#)
[114頁の「MEMPROG – メモリプログラミング」](#)

9.4. データメモリ

データメモリはI/Oメモリ、内部SRAM、不揮発性メモリ(NVM)施錠ビット、フラッシュメモリを含みます。右図はATtiny102/ATtiny104のメモリ空間がどう構成されるかを示します。

最初の64位置はI/Oメモリ用に予約され、同時に後続する32データメモリ位置はデータ用内部SRAMをアドレス指定します。

不揮発性メモリ施錠ビットと全てのフラッシュメモリ領域がデータメモリ空間に割り当てられています。これらの位置はデバイスのファームウェアに関して読み込み専用として見えます。

データメモリ用の4つの異なるアドレス指定形態は、直接、間接、事前減少付き間接、事後増加付き間接です。レジスタファイル内のレジスタR26~R31は間接アドレス指定ポイント用レジスタとして機能します。

INとOUTの命令はI/Oメモリの64位置全てをアクセスできます。LDSとSTSの命令を用いる直接アドレス指定は\$0040~\$00BF間の128位置に届きます。

間接アドレス指定はデータ空間全体に届きます。自動の事前減少付きと事後増加付きのレジスタ間接アドレス指定動作を使う時に(使われた)X,Y,Zアドレスレジスタが減少(-1)または増加(+1)されます。

図9-1. データメモリ配置図(バイトアドレス)

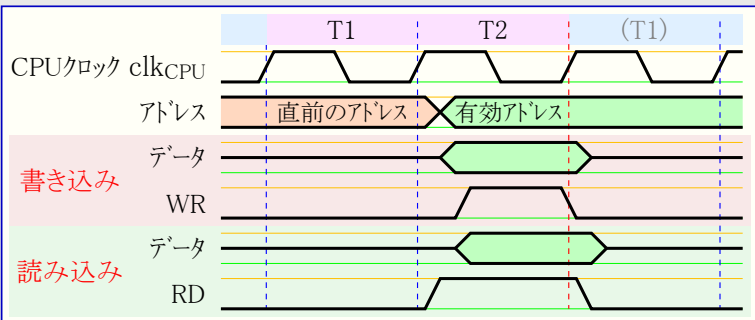
| | |
|----------------|----------------------|
| I/O空間 | \$0000~\$003F |
| SRAMデータメモリ | \$0040~\$005F |
| (予約) | \$0060~\$3EFF |
| NVM施錠ビット | \$3F00~\$3F01 |
| (予約) | \$3F02~\$3F3F |
| 構成設定ビット | \$3F40~\$3F41 |
| (予約) | \$3F42~\$3F7F |
| 校正ビット | \$3F80~\$3F81 |
| (予約) | \$3F82~\$3FBF |
| デバイス識別ビット | \$3FC0~\$3FC7 |
| (予約) | \$3FC8~\$3FFF |
| プログラム用フラッシュメモリ | \$4000~\$41FF/\$43FF |
| (予約) | \$4400~\$FFFF |

9.4.1. データメモリアクセスタイミング

データ用内蔵SRAMアクセスは右図で記載されるように2clkCPU周期で実行されます。

(**訳注**) 内蔵SRAMのアクセスを含む代表的な命令はT1,T2の2周期で実行され、T1で対象アドレスを取得/(算出)/確定し、T2で実際のアクセスが行われます。後続する(T1)は次の命令のT1です。

図9-2. データ用内蔵SRAMアクセス周期



9.5. I/O メモリ (レジスタ)

本デバイスのI/O空間定義は「[レジスタ要約](#)」で示されます。

ATtiny102/ATtiny104の全てのI/Oと周辺機能はI/O空間に配置されます。全てのI/O位置はI/O空間と16個の汎用作業レジスタ間のデータ転送を行うLDとSTの命令によってアクセスできます。USARTレジスタを除き、アドレス範囲\$00～\$1F内のI/OレジスタはSBI命令とCBI命令を使って直接的にビットアクセスが可能です。これらのレジスタではSBISとSBIC命令の使用によって単一ビット値が検査できます。より多くの詳細については「[命令要約](#)」章を参照してください。

将来のデバイスとの共通性を保つため、アクセスされる場合、予約ビットは0が書かれるべきです。予約済みI/Oメモリ アドレスは決して書かれるべきではありません。

状態フラグのいくつかはそれらへ論理1を書くことによって解除(0)されます。CBIとSBI命令は指定ビットだけを操作し、従って状態フラグのようなものを含むレジスタに使えることに注意してください。CBIとSBI命令は(I/Oアドレス)\$00～\$1Fのレジスタでのみ動作します。

I/Oと周辺制御レジスタは以降の項で説明されます。

関連リンク [144頁の「レジスタ要約」](#)

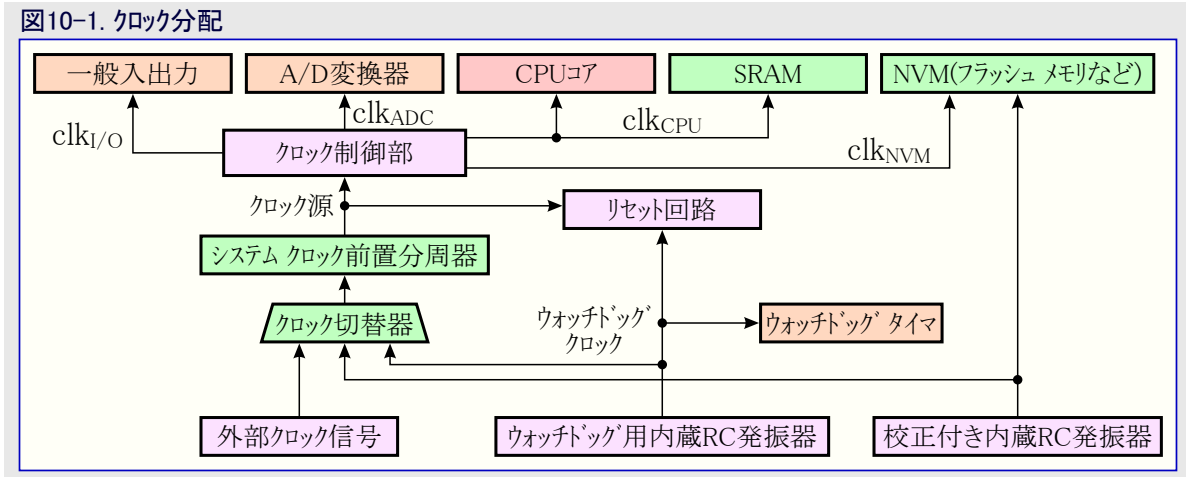
10. クロック体系

10.1. 概要

本章はATtiny102/ATtiny104に於けるクロックの分配と用語を要約します。

10.2. クロック分配

全てのクロックが与えられた時間有効である必要はありません。消費電力低減のため、「電力管理と休止形態動作」章で記述される各種休止形態動作と電力削減レジスタビットの使用により、使われていない部分のクロックを停止できます。クロック体系は下で詳述されます。



関連リンク [23頁の「電力管理と休止形態動作」](#)

10.3. クロック副系統

10.3.1. CPU クロック – clk_{CPU}

CPUクロックはAVRコアの動作と関係する系統部分に配給されます。このような部分の例はシステムレジスタとデータ用SRAMメモリです。CPUクロックの停止はコアが一般的な操作や計算を実行することを禁止します。

10.3.2. I/O クロック – clk_{I/O}

I/Oクロックはタイマ/カウンタのようなI/O部の大部分で使われます。I/Oクロックは外部割り込み部でも使われますが、いくつかの外部割り込みは例えばI/Oクロックが停止されても検出されることをこのような割り込みに許す非同期論理回路によって検出されることに注意してください。

10.3.3. NVMクロック – clk_{NVM}

NVMクロックは不揮発性メモリ(NVM)制御部の動作を制御します。NVMクロックは通常CPUクロックと同時に活動します。

10.3.4. A/D変換クロック – clk_{ADC}

A/D変換器には専用のクロック範囲が提供されます。これはデジタル回路によって生成される雑音を低減するためにCPUとI/Oクロックの停止を許します。これはより正確なA/D変換結果を与えます。

10.4. クロック元

本デバイスは選択可能なクロック主設定レジスタのクロック主選択(CLKMSR.CLKMS)ビットによる次のクロック元任意選択を持ちます。全ての同期クロック信号は主クロックから配給されます。主クロック用の3つの切り替え供給元は次のとおりです。

- ・ 校正付き8MHz内蔵発振器
- ・ 外部クロック信号
- ・ 128kHz内部発振器

活動するクロック元を変更して選択する方法は[クロック主設定レジスタのクロック主選択\(CLKMSR.CLKMS\)ビット](#)の記述を参照してください。

10.4.1. 校正付き8MHz内蔵発振器

校正された内蔵RC発振器は概ね8MHzのクロック信号を提供します。電圧と温度に依存しますが、このクロックは使用者によって高精度な校正ができます。

このクロックは**クロック主設定レジスタ**の**クロック主選択(CLKMSR.CLKMS)**ビットを**00**に設定することによって主クロックとして選択できます。一旦許可されると、発振器は外部部品なしで動作します。リセット中、ハードウェアが**発振校正(OSCCAL)**レジスタに校正値バイトを設定し、これによって発振器を自動的に校正します。この校正の精度は「**校正付き内蔵RC発振器の校正精度**」の工場校正として示されます。

特に応用が狭い範囲の温度と電圧を許す時に、既定の工場校正よりも高い精度に達することが可能です。ソフトウェアは始動または走行時のどちらかで校正データをOSCCALレジスタに再設定することができます。継続的な走行時校正法は電圧と温度を監視して検出されるどんな変化をも補償することをファームウェアに許します。

この発振器がチップ(システム)クロックとして使われる時に未だウォッチドッグ発振器がウォッチドッグ タイマとリセット時間経過に使われます。

関連リンク [117頁の「校正領域」](#)
 [123頁の「校正付き内蔵RC発振器の精度」](#)
 [138頁の「内部発振器周波数」](#)

10.4.2. 外部クロック信号

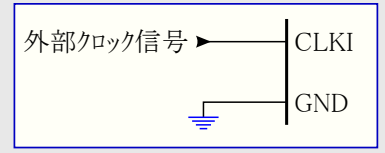
外部クロック元からデバイスを駆動するには、CLKIが右図で示されるように駆動されるべきです。デバイスを外部クロックで動かすには**クロック主設定レジスタ**の**クロック主選択(CLKMSR.CLKMS)**ビットが**10**にされなければなりません。

外部クロックを印加する時にMCUの安定な動作を保証するため、印加したクロック周波数の急な変化を避けることが必要とされます。或るクロック周期から次への2%より大きな周波数変化は予測されない事態を引き起こします。ご変化中、MCUはリセットに保たれるのを保証することが必要とされます。

表10-1. 外部クロック周波数

| 周波数 | CLKMSR.CLKMS |
|---------|--------------|
| 0~12MHz | 1 0 |

図10-2. 外部クロック駆動構成設定



10.4.3. 128kHz内部発振器

128kHz内部発振器は128kHzのクロックを提供する低電力発振器です。周波数は供給電圧、温度、1群の変量に依存します。このクロックは**クロック主設定レジスタ**の**クロック主選択(CLKMSR.CLKMS)**ビットを**01**に設定することによって主クロックとして選択できます。

10.4.4. クロック元切り替え

主クロック元は**クロック主設定レジスタ(CLKMSR)**を使って走行時に切り替えることができます。どのクロック元を切り替える時にもクロックシステムは主クロックに異常が起きないことを保証します。

10.4.5. 既定のクロック元

デバイスが電源投入またはリセットされる時に、主クロックとして常に校正付き8MHz内蔵発振器が選択されます。同期システム クロックはシステムクロック前置分周器によって8分周に制御された主クロックです。**クロック前置分周レジスタ**の**前置分周選択(CLKPSR.CLKPS)**ビットは後でシステムクロック周波数を変更するために書くことができます。次の「**システム クロック前置分周器**」項をご覧ください。

10.5. システム クロック前置分周器

システム クロックはシステム クロック前置分周器経由で主クロックから配給されます。システム クロックは「**CLKPSR – クロック前置分周レジスタ**」を設定することによって分周できます。システム クロック前置分周器は処理能力に対する必要条件が低い時の消費電力低減、またはシステム クロックを最大周波数の制限内に持って来るのに使うことができます。前置分周器は全ての主クロック元任意選択で使うことができ、CPUと全ての同期周辺機能のクロック周波数に影響を及ぼします。

システム クロック前置分周器は未だ安定動作を保証すると同時に内部クロック周波数の走行時変更の実行に使うことができます。

10.5.1. 切り替え時間

前置分周器設定を切り替える時に、システム クロック前置分周器は中間(経過途中)の周波数が直前の設定に対応するクロック周波数または新規設定に対応するクロック周波数のどちらよりも高くなく、システム クロックで異常が起きないことを保証します。

前置分周器として実行するリプル カウンタは主クロックの周波数で動き、それはCPUのクロック周波数よりも速いかもしれません。従って例えば(カウンタ値が)読めるとしても、前置分周器の状態を決めることはできず、1から他へのクロック分周値切り替えを行う正確な時間は必ずしも予測できません。

CLKPSR.CLKPS値が書かれる時から新規クロック周波数が活性(有効)になる前に $T1+T2 \sim T1+2 \times T2$ 間かかります。この間で2つの有効なクロック端が生成されます。ここでのT1は直前のクロック周期、T2は新規前置分周器設定に対応する周期です。

10.6. 始動

10.6.1. リセットからの始動

リセット元が活性(有効)になる時に内部リセットは直ちに有効設定にされます。内部リセットはリセット下が開放されて始動手順が完了されるまで有効設定にされ続けます。始動手順は次のような3つの段階を含みます。

- 1. リセット元が開放された後の最初の段階はリセット始動時間を計数するデバイスから成ります。このリセット始動時間の目的は供給電圧が十分な水準に達するのを保証することです。リセット始動時間は128KHz内部発振器を使って計数されます。
注: 実際の供給電圧は始動回路によって監視されません。例えばデバイスが早くに十分な供給電圧水準へ達したとしても、デバイスはリセット始動時間まで計数します。
- 2. 2つ目の段階は発振器始動時間を計数することで、これは校正付き内蔵発振器がシステムの他の部分によって使われる前にそれが安定状態に達することを保証します。校正付き内蔵発振器はそれが安定と見做され得る前に最低周期数の発振が必要です。
- 3. 内部リセットを開放する前の最後段階はデバイスを正しく構成設定するために不揮発性メモリから校正值と構成設定値を設定することです。構成設定時間は次表で一覧にされます。

以下のどちらかを支援する2つの始動時間任意選択があります。

- 標準始動時間 - 64ms
- 短縮始動時間 - 8ms

表10-2. 標準始動時間での校正付き内蔵発振器使用時の始動時間

| リセット | 発振器 | 構成設定 | 総始動時間 |
|------|-----|------|-------------------------------|
| 64ms | 6周期 | 21周期 | 64ms+6発振器周期+21システム クロック周期 (注) |

注: デバイスの電源投入後またはリセット後のシステム クロックは自動的に8分周された校正付き8MHz内蔵発振器に設定されます。

表10-3. 短縮始動時間での校正付き内蔵発振器使用時の始動時間

| リセット | 発振器 | 構成設定 | 総始動時間 |
|------|-----|------|------------------------------|
| 8ms | 6周期 | 21周期 | 8ms+6発振器周期+21システム クロック周期 (注) |

注: デバイスの電源投入後またはリセット後のシステム クロックは自動的に8分周された校正付き8MHz内蔵発振器に設定されます。

10.6.2. パワーダウン動作からの始動

パワーダウン休止動作からの起動時、供給電圧は十分な水準であると仮定され、発振器の安定動作を保証するために発振器始動時間だけが計数されます。発振器始動時間は選択されている主クロックで計数され、始動時間はその選択されたクロックに依存します。

表10-4. パワーダウン休止動作からの始動時間

| 発振器始動時間 | 総始動時間 |
|---------|------------|
| 6周期 | 6発振器周期 (注) |

注: 始動時間は主クロック発振器周期で測定されます。

10.6.3. アイドル,A/D変換雑音低減,スタンバイ動作からの始動

アイドル、A/D変換雑音低減、スタンバイ動作からの起動時、発振器は既に動いており、発振器始動時間は全く持ち込まれません。

10.7. クロック関係レジスタ

10.7.1. CLKMSR – クロック主設定レジスタ (Clock Main Settings Register)

名称 : CLKMSR
変位 : \$37
リセット : \$00
特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|---|---|---|---|---|---|----------|-----|
| | - | - | - | - | - | - | CLKMS1,0 | |
| アクセス種別 | R | R | R | R | R | R | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

●ビット1,0 – CLKMS1,0 : クロック主選択 (Clock Main Select Bits)

これらのビットはシステムの主クロック元を選びます。このビットは主クロックの供給元を切り替えるために走行時に書くことができます。クロックシステムは不具合なしでの主クロック元切り替えを保証します。

主クロック元の予期せぬ切り替えを避けるため、CLKMSビットを変更するには次のような保護された変更手順に従わなければなりません。

- 1. 保護されたI/Oレジスタの変更許可用の識票を構成設定変更保護 (CCP)レジスタに書いてください。
- 2. 4命令周期内に望む値でCLKMSビットを書いてください。

表10-5. 主クロックの選択

| CLKMS1 | CLKMS0 | 主クロック元 |
|--------|--------|---------------------|
| 0 | 0 | 校正付き8MHz内蔵発振器 |
| 0 | 1 | 128kHz内部発振器(WDT発振器) |
| 1 | 0 | 外部クロック信号 |
| 1 | 1 | (予約) |

10.7.2. OSCCAL – 発振校正レジスタ (Oscillator Calibration Register)

名称 : OSCCAL
変位 : \$39
リセット : \$xx (デバイス固有の校正值)
特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|------------|-----|-----|-----|-----|-----|-----|-----|
| | CAL7~0 | | | | | | | |
| アクセス種別 | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| リセット値 | デバイス固有の校正值 | | | | | | | |

●ビット7~0 – CAL7~0 : 発振校正值 (Oscillator Calibration Value)

発振校正レジスタは発振器周波数の偏差処理を省くための内蔵発振器の調整に使われます。チップのリセット中に予めプログラムされた値が自動的にこのレジスタへ書かれ、「電気的特性」章の「校正付き内蔵RC発振器の校正精度」の表で指定されるように工場校正された周波数を与えます。

応用ソフトウェアは周波数を変更するために、このレジスタに書くことができます。この発振器は「電気的特性」章の「校正付き内蔵RC発振器の校正精度」の表で指定されるように周波数を校正することができます。この範囲外への校正は保証されません。

最低発振器周波数はこれらのビットを0に設定することによって達せられます。レジスタ値の増加は発振器周波数を増します。

この発振器はフラッシュ メモリ書き込みアクセスに使われ、これによって書き込み時間が影響を及ぼされることに注意してください。さもなければ、フラッシュ メモリ書き込みが失敗するかもしれません。

MCUの安定な動作を保証するために、校正值は小さな段階で変更されるべきです。或る周回から次の周回へ2%よりも大きな周波数での段階変更は予期せぬ動きを引き起こし得ます。また、2つの連続するレジスタ値間の差は\$20を超えるべきではありません。この制限を超えた場合、MCUはクロック周波数を変更する間、リセットを保たれなければなりません。

10.7.3. CLKPSR – クロック前置分周レジスタ (Clock Prescale Register)

名称 : CLKPSR
変位 : \$36
リセット : \$03
特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|---|---|---|---|----------|-----|-----|-----|
| | - | - | - | - | CLKPS3~0 | | | |
| アクセス種別 | R | R | R | R | R/W | R/W | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

●ビット3~0 – CLKPS3~0 : クロック分周値選択 (Clock Prescaler Select Bits 3~0)

これらのビットは選択したクロック元と内部システムクロック間の分周係数を定義します。これらのビットは応用の必要条件に合わせるのにクロック周波数を変えるため、走行時に書くことができます。分周器はMCUへの主クロック入力を分周するため、分周係数が使われる時に全ての同期周辺機能の速度が減じられます。分周係数は下表で与えられます。

| 表10-6. クロック前置分周器選択 (注:既定値) | | | | | | | | | | | | | | | | |
|----------------------------|---|---|---|------|----|----|----|-----|-----|------|---|---|---|---|---|---|
| CLKPS3 | 0 | | | | | | | | 1 | | | | | | | |
| CLKPS2 | 0 | | | | 1 | | | | 0 | | | | 1 | | | |
| CLKPS1 | 0 | | 1 | | 0 | | 1 | | 0 | | 1 | | 0 | | 1 | |
| CLKPS0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 分周係数(数) | 1 | 2 | 4 | 8(注) | 16 | 32 | 64 | 128 | 256 | (予約) | | | | | | |

予期せぬクロック周波数の変更を防ぐため、CLKPSビットの変更するには保護された変更手順に従わなければなりません。

- 1. 保護されたI/Oレジスタの変更許可用の識票を構成設定変更保護(CCP)レジスタに書いてください。
- 2. (次からの)4命令周期内に望む値でCLKPSビットを書いてください。

始動でクロック分周係数8を選択するためにCLKPSビットは0011にリセットされます。応用ソフトウェアは選択したクロック元が現在の動作条件に於いてデバイスの最大周波数よりも高い周波数を持つ場合に、充分な分周係数が使われるのを保証しなければなりません。書き込み手続きが割り込まれないことを保証するため、前置分周器設定変更時に割り込みは禁止されなければなりません(訳注:本行の記述はCCP書き込みによって自動的に割り込みが禁止されるため不適切です。消し忘れと思われます)。

11. 電力管理と休止形態動作

11.1. 概要

高機能と産業的に先行するコスト効率は低電力の応用に対してAVRマイクロコントローラを理想的に選択させます。加えて休止形態動作は応用でMCU内の未使用部を一時停止することを可能にし、それによって節電します。AVRは応用で必要な消費電力に仕立てることを使用者に許す様々な休止形態動作を提供します。

11.2. 特徴

- 消費電力の最小化
- 休止形態動作
 - アイドル動作
 - A/D変換雑音低減動作
 - パワーダウン動作
 - スタンバイ動作

11.3. 休止形態動作種別

次表は異なる休止形態動作とそれらの起動元を示します。

表11-1. 各休止形態動作に於ける動作クロック範囲と復帰起動元

| 休止形態種別 | 動作クロック範囲 | | | | 発振器動作 | | 復帰起動元 (割り込み) | | | |
|-----------|----------|---------|--------|---------|----------|----------|--------------|--------|---------|----------------|
| | clk CPU | clk NVM | clk IO | clk ADC | 主クロック供給元 | INT0ピン変化 | A/D変換完了 | その他I/O | ウォッチドッグ | VCCレベル監視 (VLM) |
| アイドル | | | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| A/D変換雑音低減 | | | | ○ | ○ | ① | ○ | | ○ | ○ |
| スタンバイ | | | | | ○ | ① | | | ○ | |
| パワーダウン | | | | | | ① | | | ○ | |

① INT0についてはレベル割り込みのみです。

4つの休止形態(アイドル、A/D変換雑音低減、スタンバイ、パワーダウン)動作の何れかへ移行するには休止形態制御レジスタの**休止許可(SMCR.SE)ビット**が論理1を書かれ、**SLEEP**命令が実行されなければなりません。SMCRの**休止形態種別選択(SMCR.SM)ビット**は**SLEEP**命令によってどの休止形態が活性(有効)にされるのかを選びます。

MCUが休止形態動作中に許可した割り込みが起ると、MCUは起動します。その時にMCUは起動時間に加えて4周期停止され、割り込みルーチンを実行し、そして**SLEEP**命令の次の命令から実行を再開します。デバイスが休止状態から起動するとき、レジスタファイルとSRAMの内容は変えられません。休止形態動作中にリセットが起ると、MCUは起動し、リセットベクタから実行します。

注: レベルで起動した割り込みが起動復帰に使われる場合、MCUを起動(とMCUがその割り込み処理ルーチンへ移行)するには、変更したレベルが一定時間保持されなければならないことに注意してください。詳細については「**外部割り込み**」をご覧ください。

関連リンク [26頁の「SMCR - 休止形態制御レジスタ」](#)
[32頁の「割り込み」](#)

11.3.1. アイドル動作

休止形態種別選択(SMCR.SM)ビットが'000'を書かれると、**SLEEP**命令はMCUをアイドル動作へ移行させ、CPUを停止しますが、A/D変換器、アナログ比較器、タイマ/カウンタ、ウォッチドッグ、割り込み機構の継続動作を許します。この休止形態は基本的にclkCPUとclkNVMを停止する一方、他のクロックに走行を許します。

アイドル動作はMCUにタイマ溢れなどの内部割り込みだけでなく、外部で起動された割り込みからの起動も可能にします。アナログ比較器割り込みからの起動が必要とされないなら、アナログ比較器制御/状態レジスタAの**アナログ比較器禁止(ACSRA.ACD)ビット**を設定(1)することにより、アナログ比較器を電源断にできます。これはアイドル動作での消費電力を削減します。A/D変換が許可されるなら、この動作に移行すると変換が自動的に始まります。

関連リンク [95頁の「ACSRA - アナログ比較器制御/状態レジスタA」](#)

11.3.2. A/D変換雑音低減動作

SMCR.SMビットが'001'を書かれると、**SLEEP**命令はMCUをA/D変換雑音低減動作へ移行させ、CPUを停止しますが、A/D変換器、外部割り込み、ウォッチドッグの(許可されていれば)継続動作を許します。この休止形態は基本的にclkI/O、clkCPU、clkNVMを停止する一方、他のクロックに走行を許します。

これはA/D変換に対する雑音環境を改善し、より高い分解能の測定を可能にします。A/D変換器が許可されている場合、この動作に移行すると、変換が自動的に始まります。A/D変換完了割り込みからの他、外部リセット、ウォッチドッグシステムリセット、VCCレベル監視(VLM)割り込み、INT0の外部レベル割り込み、ピン変化割り込みだけが、A/D変換雑音低減動作からMCUを起動できます(**訳注:**本行追加)。

この動作種別はA/D変換器(ADC)を装備された全てのデバイスで利用可能です。

11.3.3. パワーダウソ動作

休止形態種別選択(SMCR.SM)ビットが'010'を書かれると、SLEEP命令はMCUをパワーダウソ動作へ移行させます。この動作では発振器が停止される一方、外部割り込み、ウォッチドック機能は(許可されていれば)継続して動作します。

次のこれらの事象だけがMCUを起動できます。

- ・ウォッチドック システム リセット、
- ・INT0の外部レベル割り込み
- ・ピン変化割り込み

この休止形態は基本的に生成した全てのクロックを停止し、非同期部の動作だけを許します。

11.3.4. スタンバイ動作

SMCR.SMビットが'100'を書かれると、SLEEP命令はMCUをスタンバイ動作へ移行させます。この動作は発振器が走行(動作)を保たれる例外を除いてパワーダウソ動作と同じです。発振器が既に走行していて始動が必要ないので、これは起動時間を減らします。

11.4. 電力削減レジスタ

電力削減レジスタ(PRR)は消費電力を削減するために個別周辺機能へのクロックを停止する方法を提供します。周辺機能へのクロックが停止されると、以下ようになります。

- ・周辺機能の現在の状態が固定化されます。
- ・関連するレジスタは読み書きすることができません。
- ・周辺機能によって使われる資源は専有されたままに留まります。

周辺機能は殆どの場合に於いてクロックを停止する前に禁止されるべきです。電力削減レジスタ(PRR)のビットを解除(0)することが周辺機能部を起し、停止前と同じ状態にします。

周辺機能単位部の停止は全体に亘る重要な消費電力の削減のために活動動作とアイドル動作で使えます。その他の休止形態動作ではクロックが予め停止されます。

関連リンク [26頁の「PRR – 電力削減レジスタ」](#)
[129頁の「周辺機能部供給電流」](#)

11.5. 消費電力の最小化

これらはAVRコアが制御するシステムで消費電力の最小化を試みる時に考慮するための様々な可能性です。一般的に休止形態動作は可能な限り多く使われるべきで、休止形態種別は動作するデバイスの機能が可能な限り少なくなるように選択されるべきです。必要とされない全ての機能は禁止されるべきです。特に次の機能部は最低可能消費電力の達成を試みるとき、特別な考慮を必要とするでしょう。

11.5.1. アナログ比較器

アイドル動作またはA/D変換雑音削減動作へ移行するとき、アナログ比較器は使われないなら、禁止されるべきです。その他の休止形態動作でのアナログ比較器は自動的に禁止されます。更なる詳細については「AC – アナログ比較器」をご覧ください。

A/D変換雑音低減動作移行時、アナログ比較器は禁止されるべきです。けれども、アナログ比較器が入力として内部基準電圧を使うように構成設定される場合、アナログ比較器は全ての休止動作形態で禁止されるべきです。さもなければ、休止動作形態と無関係に内部基準電圧が許可されます。

関連リンク [94頁の「AC – アナログ比較器」](#)

11.5.2. A/D変換器 (ADC)

許可したなら、A/D変換器は全ての休止形態動作で許可されます。節電するため、休止形態動作の何れかへ移行する前にA/D変換器は禁止されるべきです。A/D変換器がOFFそして再びONに切り替えられると、次の(最初の)変換は延長された(初回)変換になります。A/D変換器(ADC)はATtiny5/10でだけ利用可能です。

関連リンク [97頁の「ADC – A/D変換器」](#)

11.5.3. 内部基準電圧

内部基準電圧はアナログ比較器、A/D変換器によって必要とされる時に許可されます。これら単位部が上の項で記述されたように禁止されると、内部基準電圧は禁止され、電力を消費しません。再び許可する場合、この出力が使われる前に使用者は基準電圧へ起動(安定時間)を与えなければなりません。基準電圧が休止形態でON保持される場合、この出力は直ちに使えます。

11.5.4. ウォッチドック タイマ

ウォッチドック タイマが応用で必要とされないなら、この単位部はOFFされるべきです。ウォッチドック タイマが許可されていると全休止形態動作で許可され、故に常時電力を消費します。これはより深い休止形態動作での総消費電流にとって重要な一因になります。

関連リンク [29頁の「ウォッチドック タイマ」](#)

11.5.5. ポートピン

休止形態動作へ移行するとき、全てのポートピンは最小電力使用に設定されるべきです。最も重要なことはその時にピンが抵抗性負荷を駆動しないのを保証することです。I/Oクロック($\text{clk}_{\text{I/O}}$)が停止される休止形態動作ではデバイスの入力緩衝部が禁止されます。これは必要とされない時に入力論理回路によって電力が消費されないことを保証します。いくつかの場合で入力論理回路は起動条件を検出するために必要とされ、その時は許可されます。どのピンが許可されるかの詳細については「**デジタル入力許可と休止形態動作**」項を参照してください。入力緩衝部が許可され、入力信号が浮いている状態のままか、またはアナログ信号電圧がVCC/2付近の場合、入力緩衝部は過大な電力を消費するでしょう。

アナログ入力ピンに対するデジタル入力緩衝部は常に禁止されるべきです。入力ピンでのVCC/2付近のアナログ信号入力は活動動作でも重要な電流を引き起こし得ます。デジタル入力緩衝部はデジタル入力禁止レジスタ0(DIDR0)の書き込みによって禁止できます。

関連リンク [40頁の「デジタル入力許可と休止形態」](#)
 [106頁の「DIDR0 – デジタル入力禁止レジスタ0」](#)

11.6. 電力管理用レジスタ

11.6.1. SMCR – 休止形態制御レジスタ (Sleep Mode Control Register)

この休止形態制御レジスタは電力管理用の制御ビットを含みます。

名称 : SMCR
変位 : \$3A
リセット : \$00
特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|---|---|---|---|-----|-------|-----|-----|
| | - | - | - | - | | SM2~0 | | SE |
| アクセス種別 | R | R | R | R | R/W | R/W | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

●ビット3~1 – SM2~0 : 休止形態種別選択 (Sleep Mode Select Bit 2, 1 and 0)

SM2~0ビットは利用可能な4つの休止形態動作の1つを選択します。

表11-2. 休止形態動作種別選択

| SM2 | SM1 | SM0 | 休止形態動作種別 |
|-----|-----|-----|-------------|
| 0 | 0 | 0 | アイドル動作 |
| 0 | 0 | 1 | A/D変換雑音低減動作 |
| 0 | 1 | 0 | パワーダウン動作 |
| 0 | 1 | 1 | (予約) |
| 1 | 0 | 0 | スタンバイ動作 |
| 1 | 0 | 1 | (予約) |
| 1 | 1 | 0 | (予約) |
| 1 | 1 | 1 | (予約) |

●ビット0 – SE : 休止許可 (Sleep Enable)

SLEEP命令が実行される時にMCUを休止形態動作へ移行させるには、休止許可(SE)ビットが論理1を書かれなければなりません。MCUの目的外休止形態動作移行を避けるため、SLEEP命令実行直前に休止許可(SE)ビットを設定(1)し、起動後直ちに解除(0)することが推奨されます。

11.6.2. PRR – 電力削減レジスタ (Power Reduction Register)

名称 : PRR
変位 : \$35
リセット : \$00
特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|---|---|---|---|---|----------|-------|--------|
| | - | - | - | - | - | PRUSART0 | PRADC | PRTIM0 |
| アクセス種別 | R | R | R | R | R | R/W | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

●ビット2 – PRUSART0 : USART電力削減 (Power Reduction USART0)

このビットへの論理1書き込みはその単位部へのクロック停止によってUSARTを停止します。USARTの再起動時、USARTは正しい動作を保証するために再初期化されるべきです。

●ビット1 – PRADC : A/D変換器電力削減 (Power Reduction ADC)

このビットへの1書き込みはA/D変換器(ADC)を停止します。A/D変換器は停止前に禁止されなければなりません。ADCが停止されている時にアナログ比較器はADC入力多重器(MUX)を使えません。

●ビット0 – PRTIM0 : タイマ/カウンタ0電力削減 (Power Reduction Timer/Counter0)

このビットへの1書き込みはタイマ/カウンタ0部を停止します。タイマ/カウンタ0が許可されると、停止前と同様に動作は継続します。

注: アナログ比較器は「ACSR – アナログ比較器制御/状態レジスタ」のアナログ比較器禁止(ACD)ビットを使うことで禁止されます。(訳注: 共通性から本注追加)

12. SCRST – システム制御とリセット

12.1. 概要

リセット論理回路はマイクロ コントローラのリセットを管理します。これはマイクロ コントローラにリセットを発行してデバイスとその初期値に設定し、ソフトウェアによって識別されるリセット元を許します。

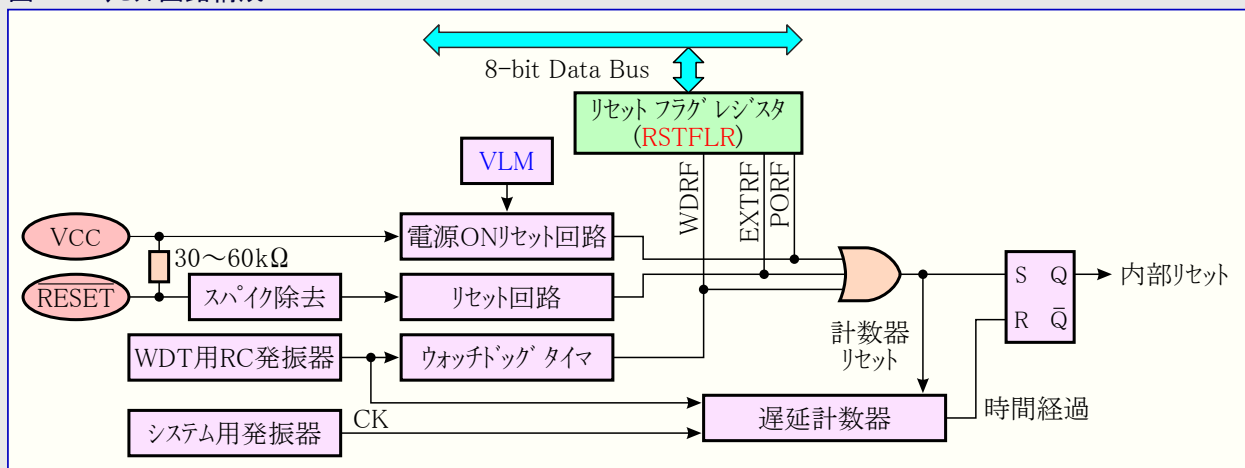
12.2. 特徴

- ・ マイクロ コントローラを初期状態にリセット
- ・ 多数のリセット元
 - 電源ONリセット
 - VCCレベル監視(VLM)リセット
 - 外部リセット
 - ウォッチドッグ システム リセット

12.3. AVRのリセット

リセット中、全てのI/Oレジスタはそれらの初期値に設定され、プログラムはリセット ベクタから実行を開始します。リセット ベクタに配置される命令は、きつとリセット処理ルーチンへの無条件相対分岐(RJMP)命令でしょう。プログラムが決して割り込み元を許可しないなら、割り込みベクタは使われず、これらの位置に通常のプログラム コードが配置できます。次の回路構成図はリセット論理回路を示します。リセット回路の電気的特性は「システムとリセットの特性」項で定義されます。

図12-1. リセット回路構成



AVRのI/Oポートはリセット元が有効になると直ちにそれらの初期状態にリセットされます。これはどのクロック元の走行も必要ありません。

全てのリセット元が無効にされてしまった後、遅延計数器(タイマ)が始動され、内部リセットを引き伸ばします。これは通常動作開始前に安定電圧へ達することを電源に許します。

関連リンク [20頁の「リセットからの始動」](#)
[124頁の「システムとリセットの特性」](#)

12.4. リセット元

本デバイスは次の4つのリセット元を持ちます。

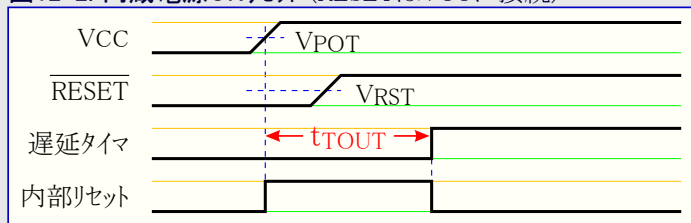
- ・ **電源ONリセット** 供給電圧が**電源ONリセット閾値電圧(V_{POA})**以下でMCUがリセットされます。
- ・ **VCCレベル監視(VLM)リセット** ... VCCピンの電圧が選択した起動レベル以下の時にMCUがリセットされます。
- ・ **外部リセット** RESETピンが**最小パルス幅**以上**Low**レベルに保たれると、MCUがリセットされます。
- ・ **ウォッチドッグ システム リセット** ウォッチドッグ システム リセット動作が許可され、ウォッチドッグ タイマが終了すると、MCUがリセットされます。

12.4.1. 電源ONリセット

電源ONリセット(POR)パルスはチップ上の検出回路によって生成されます。POR信号はVCCが検出電圧以下の時は必ず活性(有効)にされます。POR回路は供給電圧異常検出は勿論、始動リセットの起動にも使えます。

電源ONリセット回路はデバイスが電源投入でリセットされることを保証します。**電源ONリセット閾値電圧(V_{POR})**への到達はVCCの上昇後にデバイスがどのくらいリセットを保つかを決める遅延計数器(タイマ)を起動します。VCCがこの検出電圧以下に低下すると、リセット信号はどんな遅延もなく再び有効にされます。

図12-2. 内蔵電源ONリセット (RESETはVCCに接続)



関連リンク [124頁の「システムとリセットの特性」](#)

12.4.2. VCCレベル監視

ATtiny102/ATtiny104は固定の起動レベルに対してVCCピンの電圧レベルを比較するVCCレベル監視(VLM)回路を持ちます。起動レベルは**電圧レベル監視器起動レベル(VLM2~0)ビット**で設定されます。「**VLMCSR – VCCレベル監視制御/状態レジスタ**」をご覧ください。

VLM回路はVCCピンの電圧が選択した起動レベル以下かどうかを示す**状態フラグ(VLMF)**を提供します。このフラグはVLMCSRから読むことができますが、VLMF状態フラグが設定(1)された時に割り込みを生成することも可能です。この割り込みはVLMCSRレジスタの**VLM割り込み許可(VLMIE)ビット**によって許可されます。状態フラグは起動レベルを変更すること、またはそれに0を書くことによって解除(0)することができます。このフラグはVCCピンの電圧が選択した起動レベル以上に上がって戻ると自動で解除(0)されます。

VLMは供給低下でのリセット特性を改善するのに使うことができます。VLMなしでの電源ONリセット(POR)はMCUが必ずしももう機能的でないレベルへ供給電圧が落ちる前に活動しません。VLMではより早いリセットを生成することが可能です。

活動(有効)時、VLM回路は「**代表特性**」の「**VCCレベル監視器消費電流 対 動作電圧**」の表で図解されるように、或る程度の電力を消費します。節電のため、VLM回路は完全にOFF、または一定間隔でONとOFFを切り替えることができます。けれども、検出には或る程度の時間がかかり、従って安定な合図のために十分な長さで回路を許可することが推奨されます。「**VCCレベル監視器**」をご覧ください。

VLMが活性(有効)でVCCピンの電圧が選択した起動レベル以上の時が標準として動作で、VLMは短時間の間、停止することができます。VCCピンの電圧が選択した閾値以下に落ちた場合、VLMは構成設定に依存して割り込みで合図するか、またはリセットを生成するかのどちらかです。

VLMが低供給電圧でのリセット生成に構成設定されると、VCCがその**リセットレベル**以下である限り、デバイスがリセットを保ちます。供給電圧がリセットレベル以上に上昇した場合、その(リセット)条件が取り払われ、MCUはリセットを出て**電源投入始動手順**を始めます。

供給電圧がPORを起動するのに足るまで落ちた場合、供給電圧が回復された後で**電源ONリセットフラグ(PORF)**が設定(1)されます。

関連リンク [31頁の「VLMCSR – VCCレベル監視制御/状態レジスタ」](#)

[122頁の「電気的特性」](#)

[124頁の「VCCレベル監視器」](#)

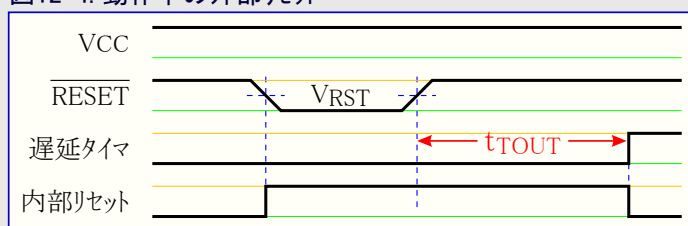
[126頁の「代表特性」](#)

12.4.3. 外部リセット

外部リセットはRESETピンのLowレベルによって生成されます。クロックが動いていなくても、最小パルス幅以上のリセットパルスはリセットを生成します。短すぎるパルスはリセット生成が保証されません。印加された信号の上昇が**リセット閾値電圧(VRST)**に達すると(遅延タイマを起動し)、遅延タイマは**遅延時間(t_TOUT)**経過後にMCUを始動します。外部リセットは**RSTDISBLヒューズ**によって禁止することができます。

関連リンク [124頁の「システムとリセットの特性」](#)

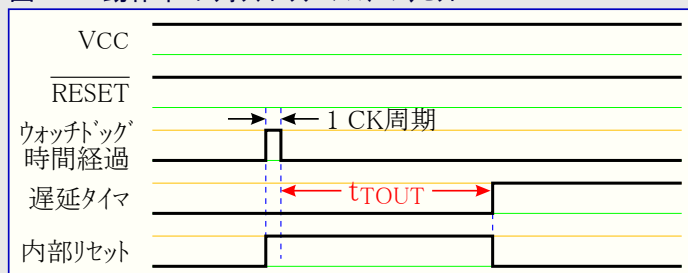
図12-4. 動作中の外部リセット



12.4.4. ウォッチドッグシステムリセット

ウォッチドッグ時間経過時、(内部的に)1CK周期幅の短いリセットパルスを生成します。遅延タイマはこのパルスの上昇端で**遅延時間(t_TOUT)**の計時を始めます。

図12-5. 動作中のウォッチドッグシステムリセット



12.5. ウォッチドッグ タイマ

ウォッチドッグ タイマが応用で必要とされないなら、この単位部はOFFされるべきです。ウォッチドッグ タイマが許可されていると全休止形態動作で許可され、故に常時電力を消費します。これはより深い休止形態動作での総消費電流にとって重要な一因になります。ウォッチドッグ リセットの詳細タイミングについては前の「ウォッチドッグ システム リセット」を参照してください。

12.5.1. 概要

ウォッチドッグ タイマは128kHzで動く独立したチップ上の発振器からクロック駆動されます。右図をご覧ください。ウォッチドッグ タイマ前置分周器の制御により、ウォッチドッグ リセット間隔は調整できます。ウォッチドッグ リセット(WDR)命令はウォッチドッグ タイマをリセットします。ウォッチドッグ タイマはそれが禁止される時とデバイス リセットが起こる時もリセットされます。10種の異なるクロック周期時間がこのリセット周期を決めるために選択できます。別のウォッチドッグ リセットなしにリセット周期が経過すると、デバイスはリセットしてリセット ベクタから実行します。

ウォッチドッグ タイマはリセットの代わりに割り込みを生成する設定にもできます。これは「[パワーダウン動作](#)から起動するのにウォッチドッグを使う時に変大有効となり得ます。

予期せぬウォッチドッグ 禁止や予期せぬ計時終了周期変更を防ぐため、WDTON構成設定ビットによって2つの異なる安全レベルが選択されます。詳細については次の「ウォッチドッグ タイマ構成設定変更手順」をご覧ください。

図12-6. ウォッチドッグ タイマ構成図 (訳注:内容に合せ修正)

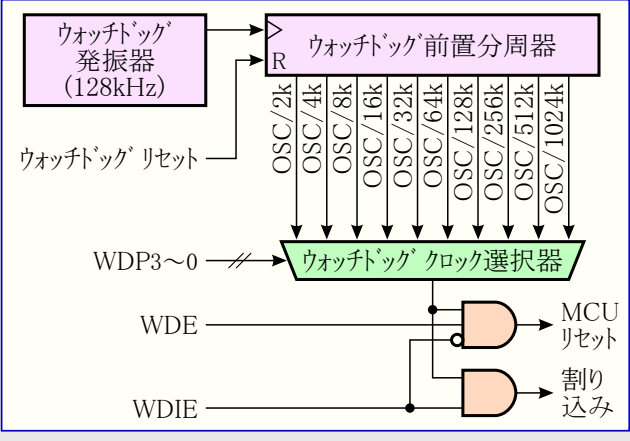


表12-1. WDTON構成設定ビットの設定によるウォッチドッグ機能設定

| WDTON構成設定ビット | 安全レベル | WDT初期状態 | WDT禁止方法 | 計時完了時間変更方法 |
|--------------|-------|---------|-----------|------------|
| 非プログラム(1) | 1 | 禁止 | 保護された変更手順 | なし(常時可) |
| プログラム(0) | 2 | 許可 | なし(常時許可) | 保護された変更手順 |

12.5.2. ウォッチドッグ タイマ構成設定変更手順

ウォッチドッグ 構成設定変更手順は安全レベル間で異なります。

12.5.2.1. 安全レベル1

この動作種別ではウォッチドッグ タイマが初めに禁止されますが、どんな制限もなくウォッチドッグ システム リセット許可(WDE)ビットに1を書くことによって許可できます。許可したウォッチドッグ タイマを禁止する時に特別な手順が必要とされます。許可したウォッチドッグ タイマを禁止するには次の手順に従わなければなりません。

- 1. 保護されたI/Oレジスタの変更許可用の識票を構成設定変更保護(CCP)レジスタに書いてください。
- 2. (次からの)4命令周期内に同じ操作(命令)で欲したWDEとウォッチドッグ タイマ前置分周選択(WDP3~0)ビットを書いてください。

12.5.2.2. 安全レベル2

この動作種別ではウォッチドッグ タイマが常に許可され、WDEビットは常に1として読めます。ウォッチドッグ 計時完了周期を変更する時に保護された変更が必要とされます。ウォッチドッグ 計時完了周期を変更するには次の手順に従わなければなりません。

- 1. 保護されたI/Oレジスタの変更許可用の識票を構成設定変更保護(CCP)レジスタに書いてください。
- 2. (次からの)4命令周期内に同じ操作(命令)でWDP2~0ビットを書いてください。WDEビットに書かれた値は無関係です。

12.5.3. コード例

次のコード例はウォッチドッグ(WDT)をOFFに切り替える方法を示します。本例は(例えば全割り込み禁止によって)割り込みが制御され、それ故この関数実行中に割り込みが起きない前提です。

| アセンブリ言語プログラム例 | | | |
|---------------|------|-----------------|--------------------|
| WDT_OFF: | WDR | | ;ウォッチドッグ タイマ リセット |
| | IN | R16, RSTFLR | ;現RSTFLR値を取得 |
| | ANDI | R16, ~(1<<WDRF) | ;WDRFビットのみ0値を取得 |
| | OUT | RSTFLR, R16 | ;RSTFLRのWDRFを解除(0) |
| | LDI | R16, \$D8 | ;変更許可識票値を取得 |
| | OUT | CCP, R16 | ;構成設定変更許可手順開始 |
| | LDI | R16, (0<<WDE) | ;WDE論理0値を取得 |
| | OUT | WDTCSR, R16 | ;ウォッチドッグ 禁止 |
| | RET | | ;呼び出し元へ復帰 |

注: 「コード例について」をご覧ください。

12.6. リセット関係レジスタ

12.6.1. WDTCSR – ウォッチドッグ タイマ制御/状態レジスタ (Watchdog Timer Control and Status Register)

名称 : WDTCSR
変位 : \$31
リセット : '0000x000'
特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|------|------|------|---|-----|-----|--------|-----|
| | WDIF | WDIE | WDP3 | — | WDE | | WDP2~0 | |
| アクセス種別 | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 不定 | 0 | 0 | 0 |

- ビット7 – WDIF : ウォッチドッグ割り込み要求フラグ (Watchdog Timer Interrupt Flag)
ウォッチドッグ タイマが割り込みに設定され、ウォッチドッグ タイマで計時完了が起こると、本ビットが設定(1)されます。対応する割り込み処理ベクタを実行すると、WDIFはハードウェアによって解除(0)されます。代わりにWDIFはこのフラグへの論理1書き込みによっても解除(0)されます。ステータスレジスタ(SREG)の全割り込み許可(I)ビットとウォッチドッグ割り込み許可(WDIE)が設定(1)されると、ウォッチドッグ計時完了割り込みが実行されます。
- ビット6 – WDIE : ウォッチドッグ割り込み許可 (Watchdog Timer Interrupt Enable)
このビットが1を書かれ、SREGの全割り込み許可(I)ビットが設定(1)されると、ウォッチドッグ割り込みが許可されます。この設定での組み合わせでウォッチドッグシステムリセット許可(WDE)ビットが解除(0)されると、ウォッチドッグ タイマは割り込み動作形態になり、ウォッチドッグ タイマで計時完了が起こると、対応する割り込みが実行されます。
WDEが設定(1)されると、ウォッチドッグ タイマは割り込み及びシステムリセット動作形態です。ウォッチドッグ タイマでの最初の計時完了がウォッチドッグ割り込み要求(WDIF)フラグを設定(1)します。対応する割り込みベクタの実行はハードウェアによってWDIEとWDIFを自動的に解除(0)します(ウォッチドッグはシステムリセット動作形態になります)。これは割り込みを使うのと同時にウォッチドッグ タイマ保護を維持するのに有用です。割り込み及びシステムリセット動作形態に留まるには、各割り込み後にWDIEが設定(1)されなければなりません。然しながら、これがウォッチドッグシステムリセット動作形態の安全機能を危険に晒すかもしれないため、これは割り込み処理ルーチン自身内で行われるべきではありません。次の計時完了に先立って割り込みが実行されない場合、システムリセットが適用(実行)されます。

| 表12-2. ウォッチドッグ タイマ構成設定 | | | | |
|------------------------|-----|------|----------------|----------------------|
| WDTON | WDE | WDIE | 動作種別 | 計時完了での動作 |
| 1 | 0 | 0 | 停止 | なし |
| 1 | 0 | 1 | 割り込み | 割り込み |
| 1 | 1 | 0 | システムリセット | リセット |
| 1 | 1 | 1 | 割り込み及びシステムリセット | 割り込み、その後システムリセット動作種別 |
| 0 | x | x | システムリセット | リセット |

注: WDTON構成設定ビットは0でプログラム、1で非プログラムに設定です。

- ビット3 – WDE : ウォッチドッグシステムリセット許可 (Watchdog System Reset Enable)
WDEはリセットフラグレジスタ(RSTFLR)のウォッチドッグリセットフラグ(WDRF)によって無効にされます。これはWDRFが設定(1)されると、WDEが常に設定(1)されることを意味します。WDEを解除(0)するにはWDRFが先に解除(0)されなければなりません。この特徴は失敗を引き起こす状態中の複数リセットと失敗後の安全な起動を保証します。
- ビット5,2~0 – WDP3~0 : ウォッチドッグタイマ前置分周選択 (Watchdog Timer Prescaler)
このWDP3~0ビットはウォッチドッグタイマが走行する時のウォッチドッグタイマの前置分周を決めます。各種前置分周値と対応する計時完了周期は下表で示されます。

| 表12-3. ウォッチドッグ前置分周選択 | | | | | | | | | | | | | | | | |
|----------------------|------|------|------|--------|-------|------|------|------|------|-------|------|---|---|---|---|---|
| WDP3 | 0 | | | | | | | | 1 | | | | | | | |
| WDP2 | 0 | | | | 1 | | | | 0 | | | | 1 | | | |
| WDP1 | 0 | | 1 | | 0 | | 1 | | 0 | | 1 | | 0 | | 1 | |
| WDP0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| WDT発振周期数 | 2k | 4k | 8k | 16k | 32k | 64k | 128k | 256k | 512k | 1024k | (予約) | | | | | |
| 代表的計時完了周期 (VCC=5V) | 16ms | 32ms | 64ms | 0.125s | 0.25s | 0.5s | 1.0s | 2.0s | 4.0s | 8.0s | | | | | | |

12.6.2. VLMCSR – VCCレベル監視制御/状態レジスタ (VCC Monitoring Control and Status Register)

名称 : VLMCSR
変位 : \$34
リセット : \$00
特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|------|-------|---|---|---|--------|-----|-----|
| | VLMF | VLMIE | - | - | - | VLM2~0 | | |
| アクセス種別 | R | R/W | R | R | R | R/W | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- **ビット7 – VLMF : VLMフラグ (VLM Flag)**
このビットは電圧レベル条件が起動されたことを示すためにVLM回路によって設定(1)されます。このビットは起動レベル選択が“禁止”に設定された時、またはVCCの電圧が選択した起動レベル以上へ上昇した時に解除(0)されます。
- **ビット6 – VLMIE : VLM割り込み許可 (VLM Interrupt Enable)**
このビットが設定(1)されると、VLM割り込みが許可されます。VLM割り込みはVLMフラグが設定(1)される時毎に生成されます。
- **ビット2~0 – VLM2~0 : 電圧レベル監視器起動レベル (Trigger Level of Voltage Level Monitor)**
これらのビットは電圧レベル監視器に対する起動レベルを設定します。

表12-4. 電圧レベル監視器の起動レベル設定

| VLM2~0 | ラベル表記 | 説明 |
|--------|-------|--|
| 0 0 0 | VLM0 | 電圧レベル監視器禁止 |
| 0 0 1 | VLM1L | この起動は通常の電源ONリセット(POR)を生成します。 |
| 0 1 0 | VLM1H | VLMフラグは設定(1)されません。 |
| 0 1 1 | VLM2 | この起動はVLMフラグ(VLMF)を設定(1)し、許可ならばVLM割り込みを生成します。 |
| 1 0 0 | VLM3 | |
| 1 0 1 | | (使用不可) |
| 1 1 0 | | |
| 1 1 1 | | |

12.6.3. RSTFLR – リセット フラグ レジスタ (Reset Flag Register)

名称 : RSTFLR
変位 : \$3B
リセット : ‘0000x0xx’
特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|---|---|---|---|------|---|-------|------|
| | - | - | - | - | WDRF | - | EXTRF | PORF |
| アクセス種別 | R | R | R | R | R/W | R | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 内容参照 | 0 | 内容参照 | 内容参照 |

- **ビット3 – WDRF : ウォッチドッグ リセット フラグ (Watchdog Reset Flag)**
このビットはウォッチドッグ システム リセットが起こると設定(1)されます。このビットは電源ONリセットまたは、このフラグへの論理0書き込みによってリセット(0)されます。
- **ビット1 – EXTRF : 外部リセット フラグ (External Reset Flag)**
このビットは外部リセットが起こると設定(1)されます。このビットは電源ONリセットまたは、このフラグへの論理0書き込みによってリセット(0)されます。
- **ビット0 – PORF : 電源ONリセット フラグ (Power-on Reset Flag)**
このビットは電源ONリセットが起こると設定(1)されます。このビットはこのフラグへの論理0書き込みによってのみリセット(0)されます。
リセット条件の確認にリセット フラグを使うため、使用者はプログラム内で可能な限り早くRSTFLRを読み、そして解除(0)すべきです。別のリセットが起こる前にこのレジスタが解除(0)されると、そのリセット元はリセット フラグを調べることによって得られます。

13. 割り込み

13.1. 概要

本項は本デバイスによって実行される割り込み操作の詳細を記述します。AVR割り込み操作の一般説明については「リセットと割り込みの扱い」の記述を参照してください。

関連リンク [13頁の「リセットと割り込みの扱い」](#)

13.2. 割り込みベクタ

割り込みベクタは下表で記述されます。

表13-1. リセットと割り込みのベクタ

| ベクタ番号 | プログラム アドレス | ラベル表記 | 割り込み元 |
|-------|------------|---------------|------------------------|
| 1 | \$0000 | RESET | 電源ON, WDT, VLM等の各種リセット |
| 2 | \$0001 | INT0 | 外部割り込み要求0 |
| 3 | \$0002 | PCINT0 (PCIO) | ピン変化0群割り込み要求 |
| 4 | \$0003 | PCINT1 (PCI1) | ピン変化1群割り込み要求 |
| 5 | \$0004 | TIM0_CAPT | タイマ/カウンタ0捕獲発生 |
| 6 | \$0005 | TIM0_OVF | タイマ/カウンタ0溢れ |
| 7 | \$0006 | TIM0_COMPA | タイマ/カウンタ0比較A一致 |
| 8 | \$0007 | TIM0_COMPB | タイマ/カウンタ0比較B一致 |
| 9 | \$0008 | ANA_COMP | アナログ比較器出力遷移 |
| 10 | \$0009 | WDT | ウォッチドッグ計時完了 |
| 11 | \$000A | VLM | VCC電圧レベル監視器(低電圧検出) |
| 12 | \$000B | ADC | A/D変換完了 |
| 13 | \$000C | USART0_RXS | USART受信開始 |
| 14 | \$000D | USART0_RXC | USART受信完了 |
| 15 | \$000E | USART0_DRE | USART送信データレジスタ空き |
| 16 | \$000F | USART0_TXC | USART送信完了 |

プログラムが決して割り込み元を許可しない条件では、割り込みベクタは使われず、結果としてこれらの位置に通常のプログラム コードを配置することができます。

本デバイスでの最も代表的且つ一般的なリセットと割り込みのベクタ アドレス用設定プログラムは次のとおりです。

| アドレス | ラベル | 命令 | 注釈 |
|--------|--------|-----------------------|---------------------|
| \$0000 | | RJMP RESET | ;各種リセット |
| \$0001 | | RJMP INT0 | ;外部割り込み要求0 |
| \$0002 | | RJMP PCINT0 | ;ピン変化0群割り込み要求 |
| \$0003 | | RJMP PCINT1 | ;ピン変化1群割り込み要求 |
| \$0004 | | RJMP TIM0_CAPT | ;タイマ/カウンタ0捕獲発生 |
| \$0005 | | RJMP TIM0_OVF | ;タイマ/カウンタ0溢れ |
| \$0006 | | RJMP TIM0_COMPA | ;タイマ/カウンタ0比較A一致 |
| \$0007 | | RJMP TIM0_COMPB | ;タイマ/カウンタ0比較B一致 |
| \$0008 | | RJMP ANA_COMP | ;アナログ比較器出力遷移 |
| \$0009 | | RJMP WDT | ;ウォッチドッグ計時完了 |
| \$000A | | RJMP VLM | ;VCC電圧レベル監視器(低電圧検出) |
| \$000B | | RJMP ADC | ;A/D変換完了 |
| \$000C | | RJMP USART0_RXS | ;USART受信開始 |
| \$000D | | RJMP USART0_RXC | ;USART受信完了 |
| \$000E | | RJMP USART0_DRE | ;USART送信データレジスタ空き |
| \$000F | | RJMP USART0_TXC | ;USART送信完了 |
| ; | | | |
| \$0010 | RESET: | LDI R16, LOW (RAMEND) | ;RAM最終アドレス下位を取得 |
| \$0011 | | OUT SPL, R16 | ;スタックポインタを初期化 |
| | | } | ;以下、I/O初期化など |

13.3. 外部割り込み

外部割り込みはINT0ピンまたはPCINT0～11ピンの何れかによって起動されます。許可したなら、例えばINT0またはPCINT0～11ピンが出力として設定されても、割り込みが起動することに注目してください。この特徴はソフトウェア割り込みを生成する方法を提供します。[ピン変化割り込みPCI0](#)は許可したPCINT0～7の何れかが切り替わると起動します。[ピン変化割り込みPCI1](#)は許可したPCINT8～11の何れかが切り替わると起動します。ピン変化割り込み許可レジスタ n (PCMSK0, PCMSK1)は、どのピンがピン変化割り込み要因となるかを制御します。PCINT0～11でのピン変化割り込みは非同期に検知され、そしてそれはそれらの割り込みがアイドル動作以外の休止形態動作からもデバイスを起動するのに使えることを意味します。

INT0割り込みは上昇端または下降端(含む両端)またはLowレベルによって起動できます。この構成設定は外部割り込み制御レジスタA (EICRA)の詳述で示されます。INT0割り込みがレベル起動として設定、且つ許可されると、そのピンがLowに保持される限り、割り込みは(継続的に)起動します。INT0の上昇端または下降端割り込みの認知は「[クロック体系](#)」で記述されるI/Oクロックの存在を必要とすることに注意してください。

関連リンク [18頁の「クロック体系」](#)
 [34頁の「EICRA – 外部割り込み制御レジスタA」](#)
 [35頁の「PCMSK0 – ピン変化割り込み0群許可レジスタ」](#)
 [36頁の「PCMSK1 – ピン変化割り込み1群許可レジスタ」](#)

13.3.1. Lowレベル割り込み

INT0のLowレベル割り込みは非同期に検知されます。これはそれらの割り込みが[アイドル動作](#)以外の[休止形態動作](#)からもデバイスを起動するのに使えることを意味します。I/Oクロックはアイドル動作を除く全休止形態動作で停止されます。

レベル起動割り込みが[パワーダウン動作](#)からの起動に使われる場合、この必要としたレベルはレベル割り込みを起動する完全な起動復帰のため、MCUに対して充分長く保持されなければならないことに注意してください。このレベルが起動時間の最後に先立って消滅すると、MCUは今までどおり起動しますが、割り込みが生成されません。起動時間は「[クロック体系](#)」で示されるように定義されます。

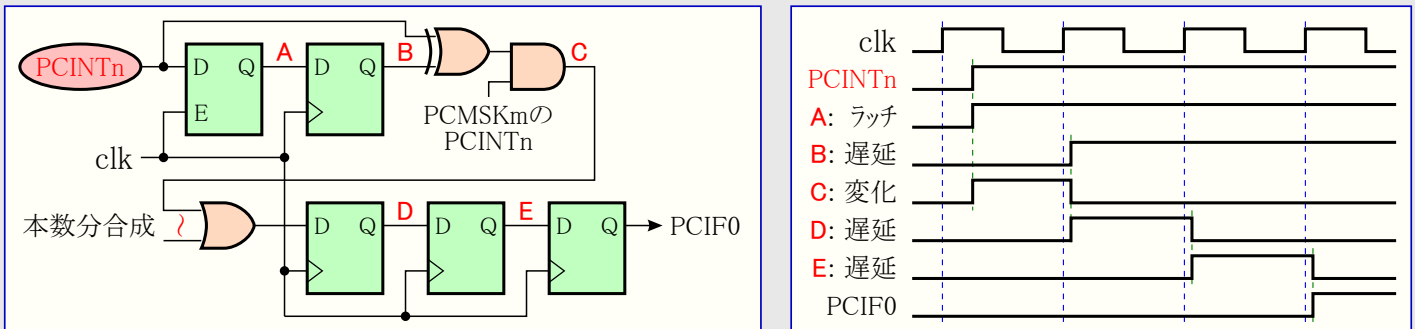
デバイスが起動復帰する前に割り込みピン上のLowレベルが取り去られると、プログラム実行は割り込み処理ルーチンへ転換されませんが、[SLEEP](#)命令に続く命令から継続します。

関連リンク [18頁の「クロック体系」](#)

13.3.2. ピン変化割り込みタイミング

ピン変化割り込みのタイミング例は次図で示されます。

図13-1. ピン変化割り込みタイミング



13.4. 割り込み用レジスタ

13.4.1. EICRA – 外部割り込み制御レジスタA (External Interrupt Control Register A)

外部割り込み制御レジスタAは割り込み条件制御ビットを含みます。

名称 : EICRA
変位 : \$15
リセット : \$00
特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|---|---|---|---|---|---|---------|-----|
| | - | - | - | - | - | - | ISC01,0 | |
| アクセス種別 | R | R | R | R | R | R | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

●ビット1,0 – ISC01,0 : 外部割り込み0条件制御 (Interrupt Sense Control 0)

外部割り込み0はステータスレジスタ(SREG)の全割り込み許可(I)ビットと外部割り込み許可レジスタ(EIMSK)の外部割り込み0許可(INT0)ビットが設定(1)される場合のINT0外部ピンによって活性(有効)にされます。割り込みを活性にする外部INT0ピンのエッジとレベルは右表で定義されます。INT0ピンの値はエッジ検出に先立って採取されます。エッジまたは論理変化割り込みが選択される場合、1クロック周期よりも長く留まるパルスは割り込みを生成します。より短いパルスは割り込み発生が保証されません。Lowレベル割り込みが選択される場合、そのLowレベルは割り込みを生成するのに現在実行している命令の完了まで保たれなければなりません。

表13-2. 外部割り込み0(INT0)割り込み条件

| ISC01 | ISC00 | 割り込み発生条件 |
|-------|-------|-----------------|
| 0 | 0 | INT0ピンのLowレベル |
| 0 | 1 | INT0ピンの論理変化(両端) |
| 1 | 0 | INT0ピンの下降端 |
| 1 | 1 | INT0ピンの上昇端 |

13.4.2. EIMSK – 外部割り込み許可レジスタ (External Interrupt Mask Register)

名称 : EIMSK
変位 : \$13
リセット : \$00
特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|---|---|---|---|---|---|---|------|
| | - | - | - | - | - | - | - | INT0 |
| アクセス種別 | R | R | R | R | R | R | R | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

●ビット0 – INT0 : 外部割り込み0許可 (External Interrupt Request 0 Enable)

ステータスレジスタ(SREG)の全割り込み許可(I)ビットが設定(1)され、INT0ビットが設定(1)されると、INT0外部ピン割り込みが許可されます。外部割り込み制御レジスタAの割り込み条件制御0(EICRA.ISC0)ビットは、この外部割り込みがINT0ピンの上昇端、下降端、両端、またはLowレベルのどれで活性(有効)にされるかを定義します。例えばINT0ピンが出力として設定されても、このピンの動きは割り込み要求を引き起こします。外部割り込み要求0に対応する割り込みはINT0割り込みベクタから実行されます。

13.4.3. EIFR – 外部割り込み要求フラグレジスタ (External Interrupt Flag Register)

名称 : EIFR
変位 : \$14
リセット : \$00
特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|---|---|---|---|---|---|---|-------|
| | - | - | - | - | - | - | - | INTF0 |
| アクセス種別 | R | R | R | R | R | R | R | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

●ビット0 – INTF0 : 外部割り込み0要求フラグ (External Interrupt Flag 0)

INT0ピン上のエッジまたは論理変化が割り込み要求を起動すると、INTF0が設定(1)になります。ステータスレジスタ(SREG)の全割り込み許可(I)ビットと外部割り込み許可レジスタ(EIMSK)の外部割り込み0許可(INT0)ビットが設定(1)なら、MCUは対応する割り込みベクタへ飛びます。このフラグは割り込みルーチンが実行されると解除(0)されます。代わりにこのフラグは論理1を書くことによっても解除(0)できます。INT0がレベル割り込みとして設定されると、このフラグは常に解除(0)されます。

13.4.4. PCICR – ピン変化割り込み制御レジスタ (Pin Change Interrupt Control Register)

名称 : PCICR
変位 : \$12
リセット : \$00
特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|---|---|---|---|---|---|-------|-------|
| | - | - | - | - | - | - | PCIE1 | PCIE0 |
| アクセス種別 | R | R | R | R | R | R | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

● ビット1 – PCIE1 : ピン変化1群割り込み許可 (Pin Change Interrupt Enable 1)

ステータスレジスタ(SREG)の全割り込み許可(I)ビットが設定(1)され、PCIE1ビットが設定(1)されると、ピン変化1群割り込みが許可されます。許可したPCINT8～11ピンの何れかの変化が割り込みを起こします。このピン変化割り込み要求に対応する割り込みはPCIE1割り込みベクタから実行されます。PCINT8～11ピンはピン変化割り込み許可レジスタ1(PCMSK1)によって個別に許可されます。

● ビット0 – PCIE0 : ピン変化0群割り込み許可 (Pin Change Interrupt Enable 0)

ステータスレジスタ(SREG)の全割り込み許可(I)ビットが設定(1)され、PCIE0ビットが設定(1)されると、ピン変化0群割り込みが許可されます。許可したPCINT0～7ピンの何れかの変化が割り込みを起こします。このピン変化割り込み要求に対応する割り込みはPCIE0割り込みベクタから実行されます。PCINT0～7ピンはピン変化割り込み許可レジスタ0(PCMSK0)によって個別に許可されます。

13.4.5. PCIFR – ピン変化割り込み要求フラグ レジスタ (Pin Change Interrupt Flag Register)

名称 : PCIFR
変位 : \$11
リセット : \$00
特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|---|---|---|---|---|---|-------|-------|
| | - | - | - | - | - | - | PCIF1 | PCIF0 |
| アクセス種別 | R | R | R | R | R | R | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

● ビット1 – PCIF1 : ピン変化1群割り込み要求フラグ (Pin Change Interrupt Flag 1)

PCINT8～11ピンの何れかの論理変化が割り込み要求を起動すると、PCIF1が設定(1)になります。ステータスレジスタ(SREG)の全割り込み許可(I)ビットとピン変化割り込み制御レジスタ(PCICR)のピン変化1群割り込み許可(PCIE1)ビットが設定(1)なら、MCUは対応する割り込みベクタへ飛びます。このフラグは割り込み処理ルーチンが開始されると解除(0)されます。代わりにこのフラグは論理1を書くことによっても解除(0)できます。

● ビット0 – PCIF0 : ピン変化0群割り込み要求フラグ (Pin Change Interrupt Flag 0)

PCINT0～7ピンの何れかの論理変化が割り込み要求を起動すると、PCIF0が設定(1)になります。ステータスレジスタ(SREG)の全割り込み許可(I)ビットとピン変化割り込み制御レジスタ(PCICR)のピン変化0群割り込み許可(PCIE0)ビットが設定(1)なら、MCUは対応する割り込みベクタへ飛びます。このフラグは割り込み処理ルーチンが開始されると解除(0)されます。代わりにこのフラグは論理1を書くことによっても解除(0)できます。

13.4.6. PCMSK0 – ピン変化割り込み許可レジスタ0 (Pin Change Mask Register 0)

名称 : PCMSK0
変位 : \$0F
リセット : \$00
特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|----------|----------|----------|----------|----------|--------|--------|--------|
| | (PCINT7) | (PCINT6) | (PCINT5) | (PCINT4) | (PCINT3) | PCINT2 | PCINT1 | PCINT0 |
| アクセス種別 | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

● ビット7～0 – PCINT7～PCINT0 : ピン変化割り込み7～0許可 (Pin Change Enable Mask 7～0)

各PCINT0～7ビットは対応するI/Oピンでピン変化割り込みが許可されるかどうかを選びます。PCINT0～7とピン変化割り込み制御レジスタ(PCICR)のPCIE0が設定(1)なら、対応するI/Oピンのピン変化割り込みが許可されます。PCINT0～7が解除(0)されると、対応するI/Oピンのピン変化割り込みは禁止されます。

13.4.7. PCMSK1 – ピン変化割り込み許可レジスタ1 (Pin Change Mask Register 1)

名称 : PCMSK1
変位 : \$10
リセット : \$00
特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|---|---|---|---|---------|---------|--------|----------|
| | - | - | - | - | PCINT11 | PCINT10 | PCINT9 | (PCINT8) |
| アクセス種別 | R | R | R | R | R/W | R/W | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

● ビット3～0 – PCINT11～PCINT8 : ピン変化割り込み11～8許可 (Pin Change Enable Mask 11～8)

各PCINT8～11ビットは対応するI/Oピンでピン変化割り込みが許可されるかどうかを選びます。PCINT8～11とピン変化割り込み制御レジスタ(PCICR)のPCIE1が設定(1)なら、対応するI/Oピンのピン変化割り込みが許可されます。PCINT8～11が解除(0)されると、対応するI/Oピンのピン変化割り込みは禁止されます。

14. 入出力ポート

14.1. 概要

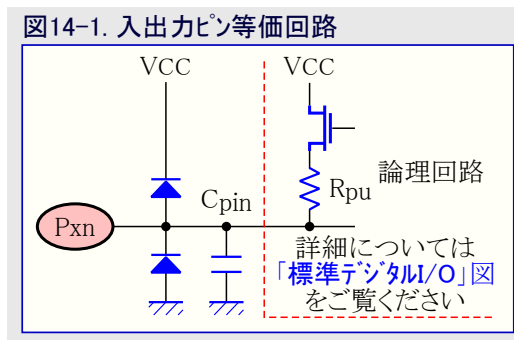
全てのAVRのポートは標準デジタルI/Oポートとして使われる時に真の読み-変更-書き(リード-モディファイ-ライト)を機能的に持ちます。これはSBIとCBI命令で他のどのピンの方角をも無意識に変更することなく、1つのポートピンの方角を変更できることを意味します。(出力として設定なら)駆動値を変更、または(入力として設定なら)プルアップ抵抗を許可/禁止する時にも同じく適用されます。各出力緩衝部は高い吐き出し(ソース)と吸い込み(シンク)の両能力で対称的な駆動特性を持ちます。

14.2. 特徴

- 全てのAVRポートは真の読み-変更-書き(リード-モディファイ-ライト)機能を持ちます。
- 専用レジスタを通した柔軟なピン構成設定
- 各出力緩衝部は高い吐き出し(ソース)と吸い込み(シンク)の両能力で対称的な駆動特性を持ちます。

14.3. 入出力ピン等価回路

全てのI/Oピンは右図で示されるようにVCCとGNDの両方に保護ダイオードを持ちます。



本章内の全てのレジスタとビットの参照は一般形で記されます。小文字の'*x*'はポート番号文字、小文字の'*n*'はビット番号を表します。けれどもプログラム内でレジスタやビット定義に使うとき、正確な形式(例えば、ここで一般に記されたPORT*xn*がポートBのビット3に対してはPORT B3)が使われなければなりません。

各々1つの出力レジスタ(PORT*x*)、方向レジスタ(DDR*x*)、プルアップ許可レジスタ(PUE*x*)、入力レジスタ(PIN*x*)の各ポートに対して、4つI/Oメモリアドレス位置が割り当てられます。入力レジスタのI/O位置は読むだけで、一方出力レジスタ、方向レジスタ、プルアップ許可レジスタは読み書き(両方)です。けれどもPIN*x*レジスタのビットへの論理1書き込みは、出力レジスタの対応ビット値を(1/0)反転する結果になります。

標準デジタルI/OとしてのI/Oポートの使用は次項で記述されます。多くのポートピンはデバイスの周辺機能用の交換機能と多重化されます。ポートピンとの各交換機能のインターフェース法は本章内の「交換ポート機能」で記述されます。交換機能の完全な記述については個別機能部項目を参照してください。

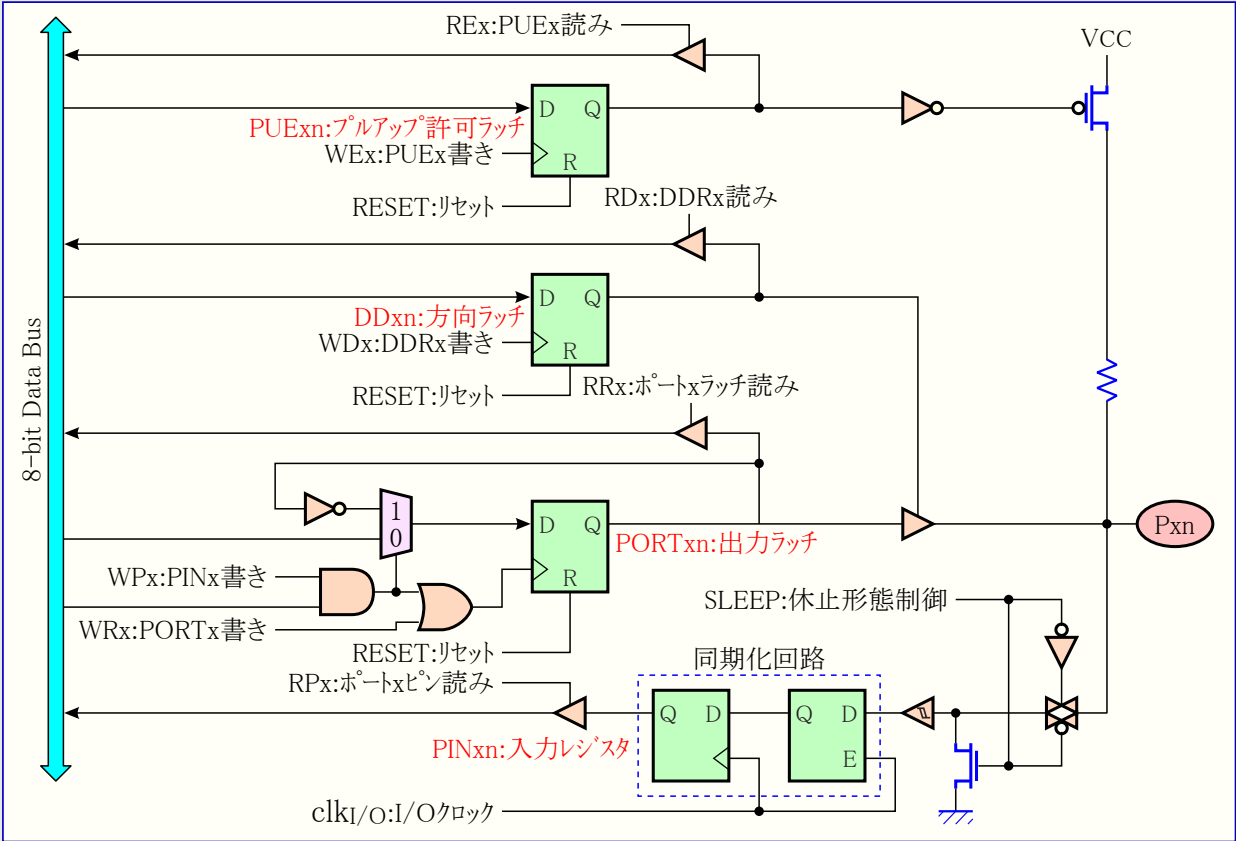
ポートピンのいくつかの交換機能の許可は、そのポート内の他のピンの標準デジタル入出力としての使用に影響しません。

関連リンク [122頁の「電気的特性」](#)

14.4. 標準デジタル入出力としてのポート

ポートは任意の内部プルアップ付き双方向I/Oポートです。次図はここで属にPxnと呼ばれるI/Oポートピンの1つの機能説明を示します。

図14-2. 標準デジタル入出力回路構成



注: WEx, WRx, WPx, WDx, REx, RRx, RPx, RDxは同一ポート内の全ピンで共通です。
clkI/OとSLEEPは全ポートで共通です。

14.4.1. ピンの構成設定

各ポートピンはDDxn、PUExn、PORTxn、PINxnの4つのレジスタビットから成ります。本章内の「I/Oポート用レジスタ」で示されるように、DDxnビットはDDRx I/O位置、PUExnビットはPUEx I/O位置、PORTxnビットはPORTx I/O位置、PINxnビットはPINx I/O位置でアクセスされます。

DDRxレジスタ内のDDxnビットはそのピンの方向を選択します。DDxnが'1'を書かれるとPxnは出力ピンとして設定されます。DDxnが'0'を書かれるとPxnは入力ピンとして設定されます。

そのピンが出力ピンとして設定される時にPORTxnが'1'を書かれると、そのポートピンはHigh(1)に駆動されます。そのピンが出力ピンとして設定される時にPORTxnが'0'を書かれると、そのポートピンはLow(0)に駆動されます。

PUExnが'1'を書かれた場合にプルアップ抵抗が活性(有効)にされます。プルアップ抵抗をOFFに切り替えるにはPUExnが'0'を書かれなければなりません。

表14-1. ポートピンの設定

| DDxn | PORTxn | PUExn | 入出力 | プルアップ抵抗 | 備考 |
|------|--------|-------|-----|---------|--|
| 0 | x | 0 | 入力 | なし | 高インピーダンス (Hi-Z) |
| 0 | x | 1 | 入力 | あり | 外部的にLowへ引かれた場合に電流を吐き出します。 |
| 1 | 0 | 0 | 出力 | なし | Low (吸い込み)出力 |
| 1 | 0 | 1 | 出力 | あり | 非推奨: Low (シンク)出力、内部プルアップ活性(有効) 内部プルアップ抵抗を通して電流を吐き出し、常に電力を消費します。 |
| 1 | 1 | 0 | 出力 | なし | High (吐き出し)出力 |
| 1 | 1 | 1 | 出力 | あり | High (吐き出し)出力、内部プルアップ活性(有効) |

ポートピンは例えクロックが動いていなくても、リセット条件が活性(有効)になるとHi-Zにされます。

14.4.2. ピンの出力交互切り替え

PINxnへの'1'書き込みはDDRxnの値に拘らず、PORTxnの値を反転切り替えます。SBI命令はポート内の1ビットの反転切り替えに使用します。

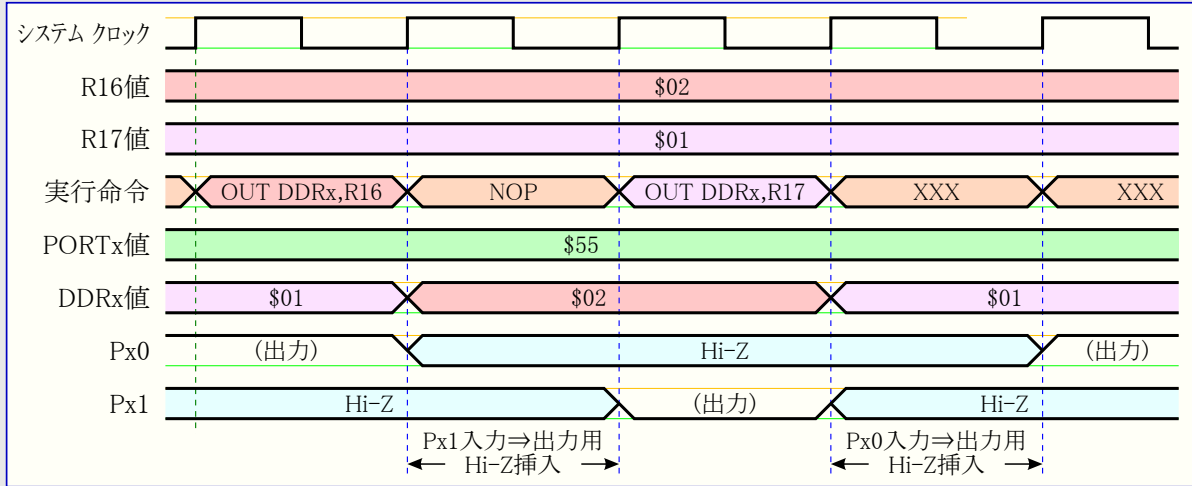
14.4.3. 接続前切断(Break-Before-Make)切り替え

接続前切断動作ではDDRxnを入力から出力へ切り換える時に1システム クロック持続する隣接Hi-Z区間が下図で示されるように導入されます。例えば、システム クロックが4MHzでDDRxnが出力にするように書かれた場合、PORTxnの値がポート ピンで見える前に250nsの隣接Hi-Z区間が導入されます。

異常を避けるため、DDRxn最大切り替え周波数は2システム クロックが推奨されます。この接続前切断はポート単位動作で、ポート単位の接続前切断許可(BBMx)ビットによって活性(有効)にされます。BBMxビットの詳細については「PORTCR – ポート制御レジスタ」をご覧ください。

DDRxnビットを出力から入力に切り替える時に隣接Hi-Z区間は導入されません。

図14-3. 接続前切断での入出力間切り替え



関連リンク 49頁の「PORTCR – ポート制御レジスタ」

14.4.4. ピン値の読み込み

DDxn方向ビットの設定に関係なく、ポートピンはPINxnレジスタビットを通して読めます。図14-2で示されるようにPINxnレジスタビットと先行するラッチは同期化回路を構成します。これは物理ピンが内部クロックのエッジ付近で値を変える場合の未定義状態(メタ ステープル)を避けるために必要とされますが、それは遅延も持ち込みます。右図は外部的に加えられたピン値を読む時の同期化タイミング図を示します。伝播遅延の最小と最大は各々 $t_{pd,min}$ と $t_{pd,max}$ で示されます。

(右図で)システム クロックの最初の上昇端直後から始まるクロック周期を考察してください。このラッチはクロックがLowの時に閉じ、クロックがHighの時に同期ラッチ信号の斜線部分で示されるように通過(トランスペアレント)となります。この信号値はシステム クロックがLowになる時に保持(ラッチ)されます。それが続くクロックの上昇端でPINxnレジスタに取り込まれます。2つの矢印 $t_{pd,min}$ と $t_{pd,max}$ によって示されるように、ピン上の単一信号遷移は出現時点に依存して0.5~1.5システムクロック周期遅られます。

ソフトウェアが指定したピン値を読み戻す時は、右図で示されるようにNOP命令が挿入されなければなりません。OUT命令はシステム クロックの上昇端で同期ラッチを設定します。この場合、同期化回路を通過する遅延時間(t_{pd})は1システムクロック周期です。

図14-4. 外部供給ピン値読み込み時の同期化

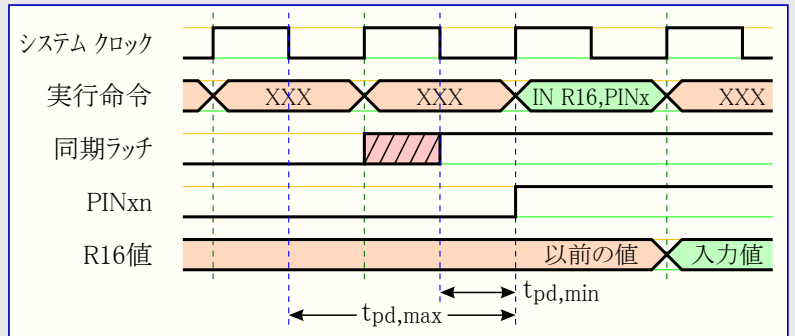
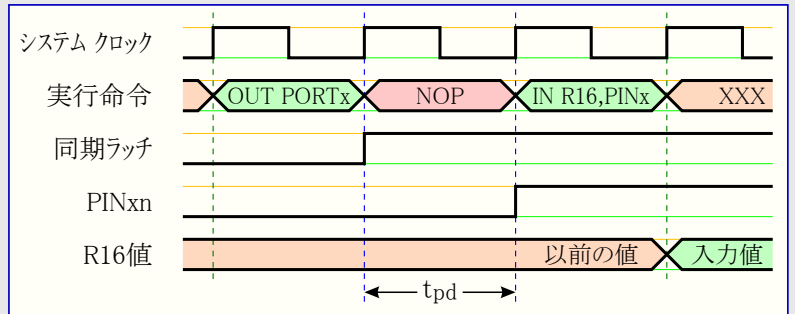


図14-5. プログラムで設定したピン値読み戻し時の同期化



14.4.5. デジタル入力許可と休止形態動作

図14-2.で示されるようにデジタル入力信号はシュミットトリガの入力をGNDにクランプできます。この図でSLEEPと印された信号は入力信号のいくつかが開放のまま、またはVCC/2付近のアナログ信号電圧を持つ場合の高消費電力を避けるため、パワーダウン動作とスタンバイ動作でMCU休止制御器によって設定(1)されます。

SLEEPは外部割り込みピンとして許可されたポートピンに対して無視されます。外部割り込み要求が許可されないなら、SLEEPはそれらのピンにも有効です。SLEEPは本章内の「交換ポート機能」で記載されるように様々な他の交換機能によっても無視されます。

外部割り込みが許可されていない”上昇端、下降端または論理変化(両端)割り込み”として設定された非同期外部割り込みピンに論理1が存在すると、上で言及した休止形態動作から(復帰)再開する時に、これらの休止形態動作に於けるクランプが要求された論理変化を生ずるので、対応する外部割り込み要求フラグが設定(=1)されます。

14.4.6. 未接続ピン

いくつかのピンが未使用にされる場合、それらのピンが定義されたレベルを持つことの保証が推奨されます。例え上記のような深い休止形態動作で多くのデジタル入力が禁止されるとしても、デジタル入力が許可される他の全ての動作(リセット、活動動作、アイドル動作)で消費電流削減のため、浮き状態入力は避けられるべきです。

未使用ピンの定義されたレベルを保証する最も簡単な方法は内部プルアップを許可することです。この場合、リセット中のプルアップは禁止されます。リセット中の低消費電力が重要なら、外部のプルアップまたはプルダウンを使うことが推奨されます。未使用ピンを直接GNDまたはVCCに接続することは、ピンが偶然に出力として設定されると過電流を引き起こす可能性があるため推奨されません。

14.4.7. プログラム例

次のコード例はポートBピンの0をHigh出力、1をLow出力、2をプルアップ指定として2と3を入力に設定する方法を示します。結果のピン値が再び読み戻されますが、前記で検討されたように、いくつかのピンへ直前に指定された値を読み戻すことができるようにNOP命令が挿入されます。

アセンブリ言語プログラム例

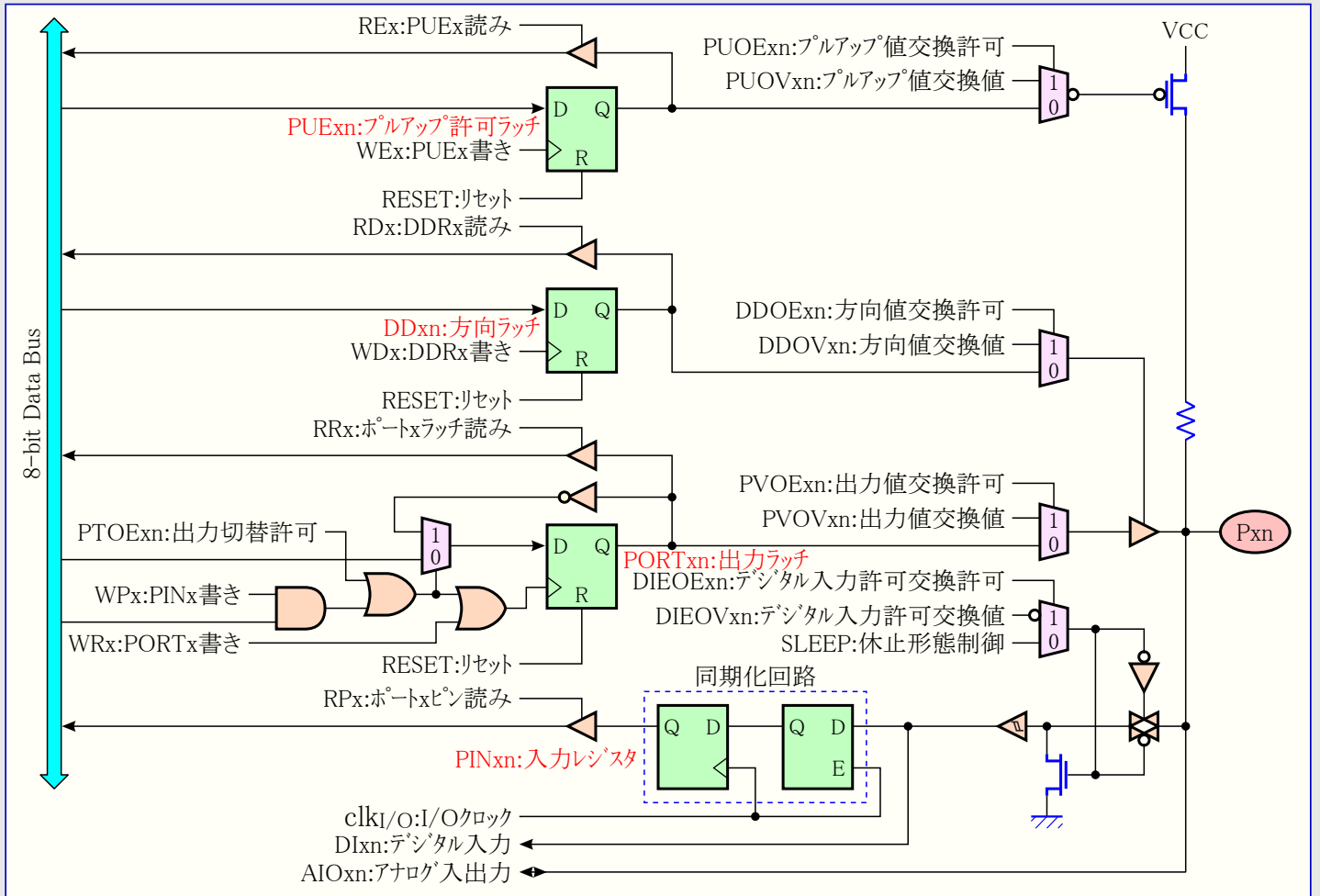
| | | |
|-----|----------------------------|--------------|
| LDI | R16, (1<<PUEB2) | ;プルアップ許可値を取得 |
| OUT | PUEB, R16 | ;プルアップを設定 |
| LDI | R16, (1<<PB0) | ;High値を取得 |
| OUT | PORTB, R16 | ;High値を設定 |
| LDI | R16, (1<<DDB1) (1<<DDB0) | ;出力ビット値を取得 |
| OUT | DDRB, R16 | ;入出力方向を設定 |
| NOP | | ;同期化遅延対処 |
| IN | R16, PINB | ;ピン値読み戻し |

注: 「コード例について」をご覧ください。

14.5. 交換ポート機能

多くのポートピンには標準デジタル入出力に加え交換機能があります。次図は単純化された図14-2.でのポートピン制御信号が交換機能によってどう重複できるかを示します。この重複信号は全てのポートピンに存在する訳ではありませんが、この図はAVRマイクロコントローラ系統の全ポートピンに適用できる一般的な記述として取り扱います。

図14-6. 交換ポート機能入出力回路構成



注: WEx, WRx, WPx, WDx, REx, RRx, RPx, RDxは同一ポート内の全ピンで共通です。
clkI/OとSLEEPは全ポートで共通です。他の信号は各ピン固有です。

次表は重複(交換)信号の機能一覧を示します。前の図で示すピンとポートは次表で示されません。重複(交換)信号は交換機能を持つ機能部で内部的に生成されます。

表14-2. 交換機能用交換信号の一般定義

| 信号略名 | 信号名 | 意味 |
|-------|------------------|---|
| PUOE | プルアップ値交換許可 | 1で、プルアップ許可はPUOV信号で制御され、0の場合、PUExn=1でプルアップが許可されます。 |
| PUOV | プルアップ値交換値 | PUOE=1時、PUExnレジスタビット設定に関係なく、プルアップの有(1)/無(0)を指定します。 |
| DDOE | 方向値交換許可 | 1で、出力駆動部はDDOV信号で制御され、0の場合、DDxnレジスタ値で制御されます。 |
| DDOV | 方向値交換値 | DDOE=1時、DDxnレジスタ値に関係なく、出力駆動部のON(1)/OFF(0)を制御します。 |
| PVOE | 出力値交換許可 | 1で出力駆動部がONなら、ポート値はPVOV信号で制御されます。出力駆動部がONで0なら、ポート値はPORTxnレジスタ値で制御されます。 |
| PVOV | 出力値交換値 | PVOE=1時、PORTxnレジスタ値に関係なく、ポート値を制御(1/0)します。 |
| PTOE | 出力切替許可 | PTOE=1時、PORTxnレジスタ値が反転します。 |
| DIEOE | デジタル入力許可 交換許可 | 1で、デジタル入力許可はDIEOV信号で制御され、0の場合、MCUの状態(活動動作、休止形態動作)によって決定されます。 |
| OIEOV | デジタル入力許可 交換値 | DIEOE=1時、MCUの状態(活動動作、休止形態動作)に関係なく、デジタル入力を許可(1)/禁止(0)します。 |
| DI | デジタル入力 | 交換機能用デジタル入力です。この信号は図上でシュミットトリガ出力に接続されていますが、これは同期化前となります。本信号はクロックとしての使用を除き、各交換機能自身が同期化します。 |
| AIO | アナログ入出力 | 交換機能用アナログ入出力です。この信号はピンに直接接続され、双方向使用ができます。 |

次節は交換機能に関連する重複(交換)信号と各ポートの交換機能を簡単に記述します。更に先の詳細については交換機能の記述を参照してください。

14.5.1. ポートAの交換機能

交換機能を持つポートAピンは下表で示されます。

| ポート ピン | 交換機能 | ポート ピン | 交換機能 |
|-----------|--|-----------|--|
| (PA7) | PCINT7 : ピン変化割り込み7入力 | PA1 | ADC1 : A/D変換チャネル0入力 |
| (PA6) | ADC3 : A/D変換チャネル3入力 PCINT6 : ピン変化割り込み6入力 | | AIN1 : アナログ比較器反転入力 |
| (PA5) | ADC2 : A/D変換チャネル3入力 OC0B : タイマ/カウンタ0 比較B一致出力 (代替位置) PCINT5 : ピン変化割り込み5入力 | | OC0B : タイマ/カウンタ0 比較B一致出力 (既定位置) PCINT1 : ピン変化割り込み1入力 TPIDATA : 直列プログラミング データ |
| (PA4) | ICP0 : タイマ/カウンタ0 捕獲起動入力 (代替位置) PCINT4 : ピン変化割り込み4入力 | PA0 | ADC0 : A/D変換チャネル0入力 |
| (PA3) | OC0A : タイマ/カウンタ0 比較A一致出力 (代替位置) PCINT3 : ピン変化割り込み3入力 | | AIN0 : アナログ比較器非反転入力 |
| PA2 | PCINT2 : ピン変化割り込み1入力 RESET : 外部リセット入力 | | T0 : タイマ/カウンタ0 外部クロック入力 (既定位置) PCINT0 : ピン変化割り込み0入力 CLKI : 外部クロック信号入力 TPICLK : 直列プログラミング クロック |

交換ピン構成設定は次のとおりです。

- **ポートA ビット7 : PA7 – PCINT7**
 - PCINT7 : ピン変化割り込み元7入力。PA7ピンはピン変化0群割り込みに対する外部割り込み元として扱えます。
- **ポートA ビット6 : PA6 – ADC3/PCINT6**
 - ADC3 : A/D変換器チャネル3入力。
 - PCINT6 : ピン変化割り込み元6入力。PA6ピンはピン変化0群割り込みに対する外部割り込み元として扱えます。
- **ポートA ビット5 : PA5 – ADC2/OC0B/PCINT5**
 - ADC2 : A/D変換器チャネル2入力。
 - OC0B : タイマ/カウンタ0の比較B一致出力。PA5ピンはタイマ/カウンタ0の比較B一致用外部出力として扱えます。この機能を扱うため、このピンは出力として構成設定(DDRA5=1)されなければなりません。OC0Bピンはタイマ機能のPWM動作出力ピンでもあります。
 - PCINT5 : ピン変化割り込み元5入力。PA5ピンはピン変化0群割り込みに対する外部割り込み元として扱えます。
- **ポートA ビット4 : PA4 – ICP0/PCINT4**
 - ICP0 : タイマ/カウンタ0の捕獲起動入力。PA4ピンはタイマ/カウンタ0用捕獲起動入力ピンとして動くことができます。
 - PCINT4 : ピン変化割り込み元4入力。PA4ピンはピン変化0群割り込みに対する外部割り込み元として扱えます。
- **ポートA ビット3 : PA3 – OC0A/PCINT3**
 - OC0A : タイマ/カウンタ0の比較A一致出力。PA3ピンはタイマ/カウンタ0の比較A一致用外部出力として扱えます。この機能を扱うため、このピンは出力として構成設定(DDRA3=1)されなければなりません。OC0Aピンはタイマ機能のPWM動作出力ピンでもあります。
 - PCINT3 : ピン変化割り込み元3入力。PA3ピンはピン変化0群割り込みに対する外部割り込み元として扱えます。
- **ポートA ビット2 : PA2 – PCINT2/RESET**
 - PCINT2 : ピン変化割り込み元2入力。PA2ピンはピン変化0群割り込みに対する外部割り込み元として扱えます。
 - RESET : 外部リセット入力。
- **ポートA ビット1 : PA1 – ADC1/AIN1/OC0B/PCINT1/TPIDATA**
 - ADC1 : A/D変換器チャネル1入力。
 - AIN1 : アナログ比較器の反転入力。アナログ比較器の機能を妨げるデジタル ポート機能を避けるため、内部プルアップをOFFにした入力としてポートピンを構成設定してください。
 - OC0B : タイマ/カウンタ0の比較B一致出力。PA1ピンはタイマ/カウンタ0の比較B一致用外部出力として扱えます。この機能を扱うため、このピンは出力として構成設定(DDRA1=1)されなければなりません。OC0Bピンはタイマ機能のPWM動作出力ピンでもあります。
 - PCINT1 : ピン変化割り込み元1入力。PA1ピンはピン変化0群割り込みに対する外部割り込み元として扱えます。
 - TPIDATA : 直列プログラミング データ。

• ホートAビット0 : PA0 – ADC0/AIN0/T0/PCINT0/CLKI/TPICLK

- ADC0 : A/D変換器チャンネル0入力。
- AIN0 : アナログ比較器の非反転入力。アナログ比較器の機能を妨げるデジタルポート機能为了避免のため、内部プルアップをOFFにした入力としてポートピンを構成設定してください。
- T0 : タイマ/カウンタ0の外部クロック入力。
- PCINT0 : ピン変化割り込み元0入力。PA0ピンはピン変化0群割り込みに対する外部割り込み元として扱えます。
- CLKI : 外部クロック元からのクロック信号入力。
- TPICLK : 直列プログラミングクロック。

次表はポートAの交換機能を「交換ポート機能」の図で示される交換信号に関連付けます。

表14-4. ホートA7～4の交換機能用交換信号

| 信号名 | PA7/PCINT7 | PA6/ADC3/PCINT6 | PA5/ADC2/OC0B/PCINT5 | PA4/ICP0/PCINT4 |
|------|--------------|----------------------|----------------------|-----------------|
| PUOE | 0 | 0 | 0 | 0 |
| PUOV | 0 | 0 | 0 | 0 |
| DDOE | 0 | 0 | 0 | 0 |
| DDOV | 0 | 0 | 0 | 0 |
| PVOE | 0 | 0 | OC0B許可・REMAP | 0 |
| PVOV | 0 | 0 | OC0B・REMAP | 0 |
| PTOE | 0 | 0 | 0 | 0 |
| DIOE | PCIE0・PCINT7 | (PCIE0・PCINT6)+ADC3D | (PCIE0・PCINT5)+ADC2D | PCIE0・PCINT4 |
| DIOV | PCIE0・PCINT7 | PCIE0・PCINT6 | PCIE0・PCINT5 | PCIE0・PCINT4 |
| DI | PCINT7入力 | PCINT6入力 | PCINT5入力 | ICP0/PCINT4入力 |
| AIO | – | ADC3入力 | ADC2入力 | – |

表14-5. ホートA3～0の交換機能用交換信号

| 信号名 | PA3/OC0A/PCINT3 | PA2/PCINT2/RESET | PA1/ADC1/AIN1/OC0B/PCINT1 | PA0/ADC0/AIN0/T0/PCINT0/CLKI |
|------|-----------------|---------------------------------|---------------------------|---|
| PUOE | 0 | RSTDISBL(注1) | 0 | EXT_CLOCK(注2) |
| PUOV | 0 | 1 | 0 | 0 |
| DDOE | 0 | RSTDISBL(注1) | 0 | EXT_CLOCK(注2) |
| DDOV | 0 | 0 | 0 | 0 |
| PVOE | OC0A許可・REMAP | 0 | OC0B許可・REMAP | EXT_CLOCK(注2) |
| PVOV | OC0A・REMAP | 0 | OC0B・REMAP | 0 |
| PTOE | 0 | 0 | 0 | 0 |
| DIOE | PCIE0・PCINT3 | RSTDISBL(注1) +(PCIE0・PCINT2) | (PCIE0・PCINT1)+ADC1D | EXT_CLOCK(注2) +(PCIE0・PCINT0)+ADC0D |
| DIOV | PCIE0・PCINT3 | RSTDISBL(注1) ・PCIE0・PCINT2 | PCIE0・PCINT1 | (EXT_CLOCK(注2)・PWR_DOWN) +(EXT_CLOCK(注2)・PCIE0・PCINT0) |
| DI | PCINT3入力 | PCINT2入力 | PCINT1入力 | T0/PCINT0/CLKI入力 |
| AIO | – | – | アナログ比較器反転/ ADC1入力 | アナログ比較器非反転/ ADC0入力 |

注1: RSTDISBLはその構成設定ビットがプログラム(0)の時に1です。

注2: EXT_CLOCKは主クロックとして外部クロック信号が選択された時に1です。

(訳注) 原書の表14-4.～7.は表14-4.と表14-5.に再合成しました。

14.5.2. ポートBの交換機能

交換機能を持つポートBピンは下表で示されます。

| 表14-8. ポートBピンの交換機能 | | | |
|--------------------|---|-----------|--|
| ポート ピン | 交換機能 | ポート ピン | 交換機能 |
| PB3 | ACO : アナログ比較器出力 ADC7 : A/D変換チャネル7入力 T0 : タイマ/カウンタ0 外部クロック入力 (代替位置) RXD0 : USART受信データ入力 PCINT11 : ピン変化割り込み11入力 | PB1 | ADC5 : A/D変換チャネル5入力 OC0A : タイマ/カウンタ0 比較A一致出力 (既定位置) INT0 : 外部割り込み0入力 XCK0 : USART転送クロック入出力 PCINT9 : ピン変化割り込み9入力 CLKO : システムクロック出力 |
| PB2 | ADC6 : A/D変換チャネル6入力 ICP0 : タイマ/カウンタ0 捕獲起動入力 (既定位置) TXD0 : USART送信データ出力 PCINT2 : ピン変化割り込み2入力 | (PB0) | ADC4 : A/D変換チャネル4入力 PCINT8 : ピン変化割り込み8入力 |

交換ピン構成設定は次のとおりです。

- ポートBビット3 : PB3 – ACO/ADC7/T0/RXD0/PCINT11
 - ACO : [アナログ比較器](#)出力。
 - ADC7 : A/D変換器チャネル7入力。
 - T0 : タイマ/カウンタ0の外部クロック入力。
 - RXD0 : USART受信データ入力。
 - PCINT11 : ピン変化割り込み元11入力。PB3ピンはピン変化1群割り込みに対する外部割り込み元として扱えます。
- ポートBビット2 : PB2 – ADC6/ICP0/TXD0/PCINT10
 - ADC6 : A/D変換器チャネル6入力。
 - ICP0 : タイマ/カウンタ0の捕獲起動入力。PB2ピンはタイマ/カウンタ0用捕獲起動入力ピンとして動くことができます。
 - TXD0 : USART送信データ入力。
 - PCINT10 : ピン変化割り込み元10入力。PB2ピンはピン変化1群割り込みに対する外部割り込み元として扱えます。
- ポートBビット1 : PB1 – ADC5/OC0A/INT0/PCINT9/CLKO
 - ADC5 : A/D変換器チャネル5入力。
 - OC0A : [タイマ/カウンタ0](#)の比較A一致出力。PB1ピンはタイマ/カウンタ0の比較A一致用外部出力として扱えます。この機能を扱うため、このピンは出力として構成設定([DDRB1=1](#))されなければなりません。OC0Aピンはタイマ機能のPWM動作出力ピンでもあります。
 - INT0 : 外部割り込み0入力。
 - XCK0 : [USART](#)転送クロック入出力。
 - PCINT9 : ピン変化割り込み元9入力。PB1ピンはピン変化1群割り込みに対する外部割り込み元として扱えます。
 - CLKO : システムクロック出力。システムクロックはPB1ピンに出力することができます。分周したシステムクロックは[CKOUT構成設定ビット](#)がプログラム([0](#))された場合に[PORTB1](#)と[DDRB1](#)設定に拘らず、出力されます。
- ポートBビット0 : PB0 – ADC4/PCINT8
 - ADC4 : [A/D変換器](#)チャネル4入力。
 - PCINT8 : ピン変化割り込み元8入力。PB0ピンは[ピン変化1群割り込み](#)に対する外部割り込み元として扱えます。

次表はポートBの交換機能を「交換ポート機能」の図で示される交換信号に関連付けます。

表14-9. ポートB3~0の交換機能用交換信号

| 信号名 | PB3/ACO/ADC7/T0 /RXD0/PCINT11 | PB2/ADC6/ICP0/TXD0 /PCINT10 | PB1/ADC5/OC0A/INT0/XCK0/PCINT9 /CLKO | PB0/ADC4 /PCINT8 |
|-------|----------------------------------|--------------------------------|--|--------------------------|
| PUOE | ACOE | TXEN0 | CKOUT(注) | 0 |
| PUOV | 0 | 0 | 0 | 0 |
| DDOE | RXEN0+(RXEN0・ACOE) | TXEN0 | CKOUT(注)+(OC0A許可・REMAP) +XCK0主装置 | 0 |
| DDOV | ACOE | TXEN0 | CLKO+(CKOUT(注)・OC0A許可・REMAP ・OC0A)+(CKOUT(注)・(OC0A許可 +REMAP)・XCK0主装置・XCK0出力) | 0 |
| PVOE | ACOE | TXEN0 | CKOUT(注) | 0 |
| PVOV | ACO・ACOE | TXEN0・TXD0出力 | (システム クロック) | 0 |
| PTOE | 0 | 0 | 0 | 0 |
| DIEOE | (PCIE1・PCINT11) +ADC7D | (PCIE1・PCINT10) +ADC6D | (PCIE1・PCINT9) +ADC5D+INT0 | (PCIE1・PCINT8) +ADC4D |
| DIEOV | PCIE0・PCINT11 | (PCIE1・PCINT10) | (PCIE1・PCINT9)+INT0 | PCIE1・PCINT8 |
| DI | T0/RXD0/PCINT11入力 | ICP0/PCINT10入力 | INT0/PCINT9入力 | PCINT8入力 |
| AIO | ADC3入力/ アナログ比較器出力 | ADC6入力 | ADC5入力 | ADC4入力 |

注: CKOUTはその構成設定ビットがプログラム(0)の時に1です。

(訳注) 原書の表14-9と表14-10は表14-9として纏めました。

14.6. I/Oポート用レジスタ

14.6.1. PINA – ポートA入力レジスタ (Port A Input Address)

名称 : PINA

変位 : \$00

リセット : \$xx

特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|---------|---------|---------|---------|---------|-------|-------|-------|
| | (PINA7) | (PINA6) | (PINA5) | (PINA4) | (PINA3) | PINA2 | PINA1 | PINA0 |
| アクセス種別 | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| リセット値 | 不定 | 不定 | 不定 | 不定 | 不定 | 不定 | 不定 | 不定 |

● ビット7~0 – PINA7~0 : ポートA入力 (Port A Input Pins)

14.6.2. DDRA – ポートA方向レジスタ (Port A Data Direction Register)

名称 : DDRA

変位 : \$01

リセット : \$00

特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|---------|---------|---------|---------|---------|-------|-------|-------|
| | (DDRA7) | (DDRA6) | (DDRA5) | (DDRA4) | (DDRA3) | DDRA2 | DDRA1 | DDRA0 |
| アクセス種別 | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

● ビット7~0 – DDRA7~0 : ポートAデータ方向 (Port A Data Direction)

14.6.3. PORTA – ポートA出力レジスタ (Port A Data Register)

名称 : PORTA

変位 : \$02

リセット : \$00

特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|----------|----------|----------|----------|----------|--------|--------|--------|
| | (PORTA7) | (PORTA6) | (PORTA5) | (PORTA4) | (PORTA3) | PORTA2 | PORTA1 | PORTA0 |
| アクセス種別 | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

● ビット7~0 – PORTA7~0 : ポートA出力データ (Port A Data)

14.6.4. PUEA – ポートAプルアップ許可制御レジスタ (Port A Pull-up Enable Control Register)

名称 : PUEA

変位 : \$03

リセット : \$00

特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|---------|---------|---------|---------|---------|-------|-------|-------|
| | (PUEA7) | (PUEA6) | (PUEA5) | (PUEA4) | (PUEA3) | PUEA2 | PUEA1 | PUEA0 |
| アクセス種別 | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

● ビット7~0 – PUEA7~0 : ポートAプルアップ許可 (Port A Pull-up Enable)

14.6.5. PINB – ポートB入力レジスタ (Port B Input Address)

名称 : PINB
変位 : \$04
リセット : \$xx
特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|---|---|---|---|-------|-------|-------|---------|
| | - | - | - | - | PINB3 | PINB2 | PINB1 | (PINB0) |
| アクセス種別 | R | R | R | R | R/W | R/W | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 不定 | 不定 | 不定 | 不定 |

● ビット3~0 – PINB3~0 : ポートB入力 (Port B Input Pins)

14.6.6. DDRB – ポートB方向レジスタ (Port B Data Direction Register)

名称 : DDRB
変位 : \$05
リセット : \$00
特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|---|---|---|---|-------|-------|-------|---------|
| | - | - | - | - | DDRB3 | DDRB2 | DDRB1 | (DDRB0) |
| アクセス種別 | R | R | R | R | R/W | R/W | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

● ビット3~0 – DDB3~0 : ポートBデータ方向 (Port B Data Direction)

14.6.7. PORTB – ポートB出力レジスタ (Port B Data Register)

名称 : PORTB
変位 : \$06
リセット : \$00
特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|---|---|---|---|--------|--------|--------|----------|
| | - | - | - | - | PORTB3 | PORTB2 | PORTB1 | (PORTB0) |
| アクセス種別 | R | R | R | R | R/W | R/W | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

● ビット3~0 – PORTB3~0 : ポートB出力データ (Port B Data)

14.6.8. PUEB – ポートBプルアップ許可制御レジスタ (Port B Pull-up Enable Control Register)

名称 : PUEB
変位 : \$07
リセット : \$00
特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|---|---|---|---|-------|-------|-------|---------|
| | - | - | - | - | PUEB3 | PUEB2 | PUEB1 | (PUEB0) |
| アクセス種別 | R | R | R | R | R/W | R/W | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

● ビット3~0 – PUEB3~0 : ポートBプルアップ許可 (Port B Pull-up Enable)

14.6.9. PORTCR – ポート制御レジスタ (Port Control Register)

名称 : PORTCR
変位 : \$16
リセット : \$00
特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|---|---|---|---|---|---|------|------|
| | - | - | - | - | - | - | BBMB | BBMA |
| アクセス種別 | R | R | R | R | R | R | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

● ビット1 – BBMB : ポートB接続前切断動作許可 (Break-Before-Make Mode Enable)

このビットが設定(1)されると、ポートB全体に対して接続前切断動作が活性(有効)にされます。そして出力にするためのDDRxn書き込み時に中間のHi-Z周期が挿入されます。

● ビット0 – BBMA : ポートA接続前切断動作許可 (Break-Before-Make Mode Enable)

このビットが設定(1)されると、ポートA全体に対して接続前切断動作が活性(有効)にされます。そして出力にするためのDDRxn書き込み時に中間のHi-Z周期が挿入されます。

15. TC0 – PWM付き16ビット タイマ/カウンタ0

15.1. 概要

16ビット タイマ/カウンタ部は正確なプログラム実行タイミング(事象管理)、波形生成、信号タイミング計測を許します。

16ビット タイマ/カウンタの簡単化した構成図は下で示されます。CPUがアクセス可能な(I/OビットとI/Oピンを含む)I/Oレジスタは赤文字(訳注: 原文太字)で示されます。デバイス仕様のI/Oレジスタとビット位置は65頁の「16ビット タイマ/カウンタ0用レジスタ」で一覧にされます。実際のI/Oピンの配置については「ピン配置」を参照してください。

関連リンク [7頁の「ピン配置」](#)

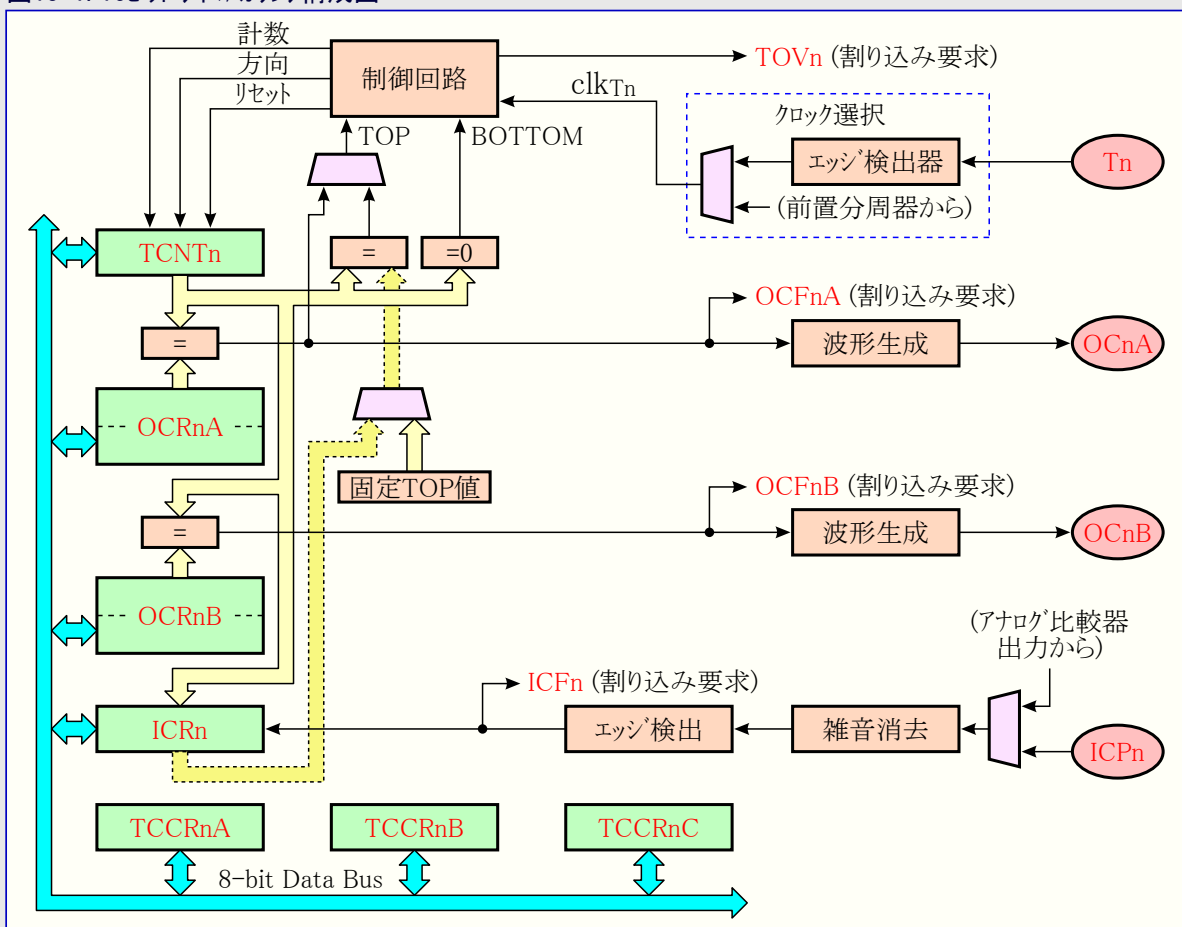
15.2. 特徴

- ・ 16ビットPWMを含む真の16ビット設計
- ・ 2つの独立した比較出力部
- ・ 2重緩衝の比較レジスタ
- ・ 1つの捕獲入力部
- ・ 捕獲入力雑音消去器
- ・ 比較一致でのタイマ/カウンタ解除(自動再設定)
- ・ 不具合なしで正しい位相のパルス幅変調器(PWM)
- ・ 可変PWM周期
- ・ 周波数発生器
- ・ 外部事象計数器
- ・ 独立した割り込み (TOV0, OCF0A, OCF0B, ICF0)

15.3. 構成図

タイマ/カウンタ0単位部を許可するには電力削減レジスタのタイマ/カウンタ0電力削減(PRR.PRTIM0)ビットが0を書かれなければなりません。

図15-1. 16ビット タイマ/カウンタ構成図



注: レジスタとビットの名前での'n'は装置番号(タイマ/カウンタ0についてはn=0)を表します。

実際のピン配置については関連リンクをご覧ください。

15.4. 定義

本章でのレジスタとビットの参照の多くは次のような一般形で書かれます。

- ・ 'n'=0はタイマ/カウンタ番号を表します。
- ・ 'x'=A,Bは比較出力部AまたはBを表します。

けれども、プログラムでレジスタまたはビット定義に使う時は正確な形式(例えばタイマ/カウンタ0のカウンタ値アクセスに対するTCNT0)が使われなければなりません。

次の定義は本章を通して使われます。

表15-1. 用語定義

| 用語 | 意味 |
|--------|--|
| BOTTOM | カウンタが\$0000に到達した時。 |
| MAX | カウンタが\$FFFF(65535)に到達した時。 |
| TOP | カウンタがTOP値に到達した時(計数動作での最大値と等しくなった時)。TOP値は固定値(\$00FF,\$01FF,\$03FF)、OCR0A値、ICR0値の何れか1つを指定できます。この指定は動作種別に依存します。 |

15.5. 関係レジスタ

タイマ/カウンタ(TCNT0)、比較レジスタ(OCR0A,OCR0B)、捕獲レジスタ(ICR0)は全て16ビットレジスタです。16ビットレジスタをアクセスする時は特別な手順に従わなければなりません。これらの手順は「[16ビット タイマ/カウンタ レジスタのアクセス](#)」項で記述されます。

タイマ/カウンタ0制御レジスタ(TCCR0A,TCCR0B,TCCR0C)は8ビットレジスタで、CPUアクセスの制限はありません。割り込み要求信号はタイマ/カウンタ0割り込み要求レジスタ(TIFR0)で全て見えます。全ての割り込みはタイマ/カウンタ0割り込み許可レジスタ(TIMSK0)で個別に遮蔽(禁止)されます。TIFR0とTIMSK0はこの図で示されません。

このタイマ/カウンタは内部的、前置分周器経由、またはT0ピンの外部クロック元によってクロック駆動されます。クロック選択論理部はタイマ/カウンタが値を増加(または減少)するのに使うクロック元とエッジを制御します。クロック元が選択されていないと、タイマ/カウンタは動きません。クロック選択論理部からの出力はタイマ/カウンタクロック(c_{clkT0})として参照されます。

2重緩衝された比較レジスタ(OCR0A,OCR0B)はタイマ/カウンタ値と常に比較されます。この比較結果は比較出力(OC0A,OC0B)ピンでPWMまたは可変周波数出力を生成するための波形生成器によって使えます。「[比較出力部](#)」をご覧ください。この比較一致発生は比較出力割り込み要求の発生に使える比較一致割り込み要求フラグ(OCF0A,OCF0B)も設定(1)します。

捕獲レジスタ(ICR0)は、捕獲起動(ICP0)ピンまたはアナログ比較器出力のどちらかの外部(エッジで起動された)事象でタイマ/カウンタ値を捕獲(複写)できます。捕獲入力部は尖頭雑音を捕らえる機会を軽減するためにデジタル濾波器(雑音消去器)を含みます。

TOP値または最大タイマ/カウンタ値は、いくつかの動作種別で、OCR0A、ICR0、または一群の固定値のどれかによって定義できます。PWM動作でTOP値としてOCR0Aを使うと、OCR0AはPWM出力生成用に使いません。けれどもこの場合、TOP値は動作中に変更されるのをTOP値に許す2重緩衝化します。固定的なTOP値が必要とされる場合、ICR0が代わりに見え、PWM出力として使われるべきOCR0Aを開放します。

関連リンク

- 67頁の「TCNT0H,TCNT0L – タイマ/カウンタ0計数値上位/下位バイト」
- 68頁の「OCR0AH,OCR0AL – タイマ/カウンタ0比較Aレジスタ上位/下位バイト」
- 68頁の「OCR0BH,OCR0BL – タイマ/カウンタ0比較Bレジスタ上位/下位バイト」
- 69頁の「ICR0H,ICR0L – タイマ/カウンタ0捕獲レジスタ上位/下位バイト」
- 65頁の「TCCR0A – タイマ/カウンタ0制御レジスタA」
- 66頁の「TCCR0B – タイマ/カウンタ0制御レジスタB」
- 67頁の「TCCR0C – タイマ/カウンタ0制御レジスタC」
- 70頁の「TIFR0 – タイマ/カウンタ0割り込み要求レジスタ」
- 69頁の「TIMSK0 – タイマ/カウンタ0割り込み許可レジスタ」
- 94頁の「AC – アナログ比較器」

15.6. 16ビット タイマ/カウンタ レジスタのアクセス

TCNT0, OCR0A, OCR0B, ICN0は8ビット バス経路でAVR CPUによってアクセスできる16ビット レジスタです。この16ビット レジスタは2回の読みまたは書き操作を使ってバイト単位でアクセスされなければなりません。16ビット タイマ/カウンタは16ビット アクセスの上位バイトの一時保存用に1つの8ビット レジスタを持ちます。16ビット タイマ/カウンタ内の全ての16ビット レジスタ間で、この同じ一時レジスタが共用されます。

下位バイト アクセスが16ビット読み書き動作を起動します。CPUによって16ビット レジスタの下位バイトが書かれると、現在一時レジスタ(TEMP)に保存された上位バイトと書かれた下位バイトの両方が同じクロック周期で16ビット レジスタに複写されます。16ビット レジスタの下位バイトがCPUによって読まれると、16ビット レジスタの上位バイトは下位バイトが読まれるのと同じクロック周期でTEMPレジスタに複写され、その後に読まれなければなりません。

注: 16ビット書き込みを実行するには下位バイトに先立って上位バイトが書かれなければなりません。16ビット読み込みについては下位バイトが上位バイトの前に読まれなければなりません。

全ての16ビット アクセスが上位バイトに対して一時レジスタを使う訳ではありません。OCR0AとOCR0Bの16ビット レジスタ読み込みは一時レジスタの使用に関係しません。

16ビット アクセス

次のコード例は割り込みが一時レジスタを更新しないことが前提の16ビット タイマ/カウンタ レジスタのアクセス法を示します。OCR0A, OCR0B, ICN0レジスタのアクセスに対して同じ原理が直接的に使えます。C言語使用時、コンパイラが16ビット アクセスを扱うことに注意してください。

アセンブリ言語プログラム例

```
~
;
;[16ビット($01FF)書き込み]
LDI    R17, $01    ;$01FFの上位バイト値取得
LDI    R16, $FF    ;$01FFの下位バイト値取得
OUT    TCNT0H, R17 ;上位バイト設定(一時レジスタ)
OUT    TCNT0L, R16 ;下位バイト設定(一時レジスタ⇒上位バイト)
;[16ビット読み込み]
IN     R16, TCNT0L ;下位バイト取得(上位バイト⇒一時レジスタ)
IN     R17, TCNT0H ;上位バイト取得(一時レジスタ)
~
;
```

C言語プログラム例

```
unsigned int i;
~
TCNT0 = 0x1FF;
i = TCNT0;
~
/* */
/* 16ビット($01FF)書き込み */
/* 16ビット読み込み */
/* */
```

注: 「コード例について」をご覧ください。

アセンブリ言語コード例はR17:R16レジスタ対にTCNT0値を戻します。

非分断読み込み

16ビット レジスタ アクセスが非分断操作であることに注意することが重要です。16ビット レジスタをアクセスする2命令間で割り込みが起き、割り込みコードがその16ビット タイマ/カウンタ レジスタの同じ若しくは他の何れかをアクセスすることによって一時レジスタを更新する場合、割り込み外のその後のアクセス結果は不正にされます。従って主コードと割り込みコードの両方が一時レジスタを更新するとき、主コードは16ビット アクセス中の割り込みを禁止しなければなりません。

次のコード例はTCNT0レジスタ内容の非分断読み込みを実行する方法を示します。OCR0A,OCR0B,ICR0も同じ原理を使うことによって実行することができます。

アセンブリ言語プログラム例

```
RD_TCNT0:    IN      R18, SREG          ; 現全割り込み許可フラグ(I)を保存
             CLI                      ; 全割り込み禁止
             IN      R16, TCNT0L        ; TCNT0下位バイト取得(上位バイト⇒一時レジスタ)
             IN      R17, TCNT0H        ; TCNT0上位バイト取得(一時レジスタ)
             OUT     SREG, R18          ; 全割り込み許可フラグ(I)を復帰
             RET                       ; 呼び出し元へ復帰
```

C言語プログラム例

```
unsigned int TIM16_Read_TCNT0(void)
{
    unsigned char sreg;                /* ステータスレジスタ一時保存変数定義 */
    unsigned int i;                   /* TCNT0読み出し変数定義 */
    sreg = SREG;                       /* 現全割り込み許可フラグ(I)を保存 */
    _CLI();                           /* 全割り込み禁止 */
    i = TCNT0;                         /* TCNT0値を取得 */
    SREG = sreg;                       /* 全割り込み許可フラグ(I)を復帰 */
    return i;                          /* TCNT0値で呼び出し元へ復帰 */
}
```

注: 「コード例について」をご覧ください。

アセンブリ言語コード例はR17:R16レジスタ対にTCNT0値を戻します。

非分断書き込み

次のコード例はTCNT0レジスタ内容の非分断書き込み法を示します。同じ原理を使うことにより、OCR0A,OCR0B,ICR0のどの書き込みも行えます。

アセンブリ言語プログラム例

```
WR_TCNT0:    IN      R18, SREG          ; 現全割り込み許可フラグ(I)を保存
             CLI                      ; 全割り込み禁止
             OUT     TCNT0H, R17        ; TCNT0上位バイト設定(一時レジスタ)
             OUT     TCNT0L, R16        ; TCNT0下位バイト設定(一時レジスタ⇒上位バイト)
             OUT     SREG, R18          ; 全割り込み許可フラグ(I)を復帰
             RET                       ; 呼び出し元へ復帰
```

C言語プログラム例

```
void TIM16_Write_TCNT0(unsigned int i)
{
    unsigned char sreg;                /* ステータスレジスタ一時保存変数定義 */
    unsigned int i;                   /* TCNT0書き込み変数定義 */
    sreg = SREG;                       /* 現全割り込み許可フラグ(I)を保存 */
    _CLI();                           /* 全割り込み禁止 */
    TCNT0 = i;                        /* TCNT0値を設定 */
    SREG = sreg;                       /* 全割り込み許可フラグ(I)を復帰 */
}
```

注: 「コード例について」をご覧ください。

アセンブリ言語コード例はR17:R16レジスタ対がTCNT0へ書かれるべき値を含むことが必要です。

15.6.1. 上位バイト一時レジスタの再使用

書かれる全レジスタについて上位バイトが同じ複数16ビットレジスタ書き込みなら、上位バイトは1度書かれることだけが必要です。けれども、直前で記述した非分断操作の同じ規則が、この場合にも適用されます。

15.7. タイマ/カウンタ クロック元

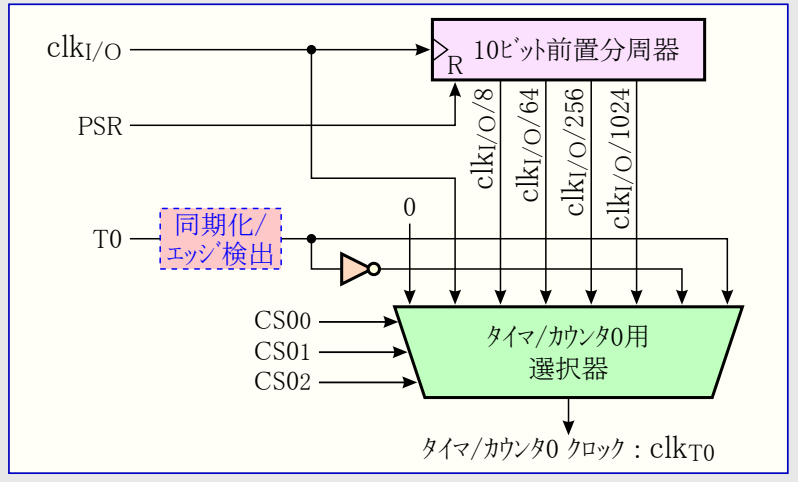
このタイマ/カウンタは内部または外部のクロック元によってクロック駆動できます。このクロック元は**タイマ/カウンタ制御レジスタB**に配置された**クロック選択(TCCR0B.CS02~0)ビット**によって制御されるクロック選択論理回路で選択されます。

15.7.1. 前置分周器

タイマ/カウンタはシステムクロック(TCCR0B.CS02~0=001設定)によって直接的にクロック駆動することができます。これはシステムクロック周波数($f_{clk_I/O}$)と等しいタイマ/カウンタ最大クロック周波数での最高速動作を提供します。選択で前置分周器からの4つの引き出し口の1つがクロック元として使えます。

前置分周されたクロックは $f_{clk_I/O}/8$, $f_{clk_I/O}/64$, $f_{clk_I/O}/256$, $f_{clk_I/O}/1024$ のどれかの周波数を持ちます。

図15-2. タイマ/カウンタ0の前置分周器



15.7.2. 前置分周器リセット

この前置分周器は自由走行で(換言するとタイマ/カウンタのクロック選択論理回路と無関係に動作し)、タイマ/カウンタ0(T0)と共用されます。前置分周器はタイマ/カウンタのクロック選択によって影響されないため、前置分周器の状態は前置分周したクロックが使われる状況に対して密接に関係します。タイマ/カウンタが許可され、前置分周器によってクロック駆動される(TCCR0B.CS02~0=5~2)とき、前置分周加工の一例が生じます。タイマ/カウンタが許可される時から最初の計数が起きるまでのシステムクロック周期数は、Nが前置分周値(8, 64, 256, 1024)とすると、 $1 \sim N+1$ システムクロック周期になり得ます。

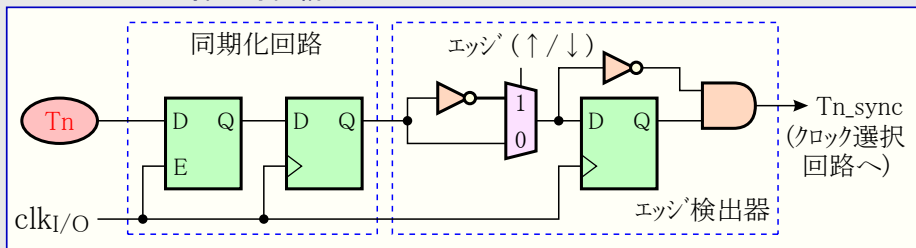
タイマ/カウンタをプログラム実行に同期するために前置分周器リセットを使うことが可能です。

15.7.3. 外部クロック

T0ピンに印加した外部クロック元はタイマ/カウンタクロック(f_{clk_T0})として使えます。このT0ピンはピン同期化論理回路によって全システムクロック周期に1回採取されます。この同期化(採取)された信号はその後にエッジ検出器を通して通過されます。下のT0同期化とエッジ検出器論理回路の構成図もご覧ください。レジスタは内部システムクロック($f_{clk_I/O}$)の上昇端でクロック駆動されます。ラッチは内部システムクロックのHigh区間で通過(Low区間で保持)です。

エッジ検出器は上昇端(CS02~0=111)または下降端(CS02~0=110)の検出毎に1つの clk_{T0} パルスを生成します。

図15-3. T0ピンの採取等価構成図



注: 'n'は装置番号(タイマ/カウンタ0についてはn=0)を表します。

同期化とエッジ検出器論理回路はT0ピンへ印加したエッジから計数器が更新されるまでに2.5~3.5システムクロック周期の遅延をもたらします。

クロック入力の許可と禁止はT0が最低1システムクロック周期に対して安定してしまっている時に行われなければならない、さもなければ不正なタイマ/カウンタクロックパルスが生成される危険があります。

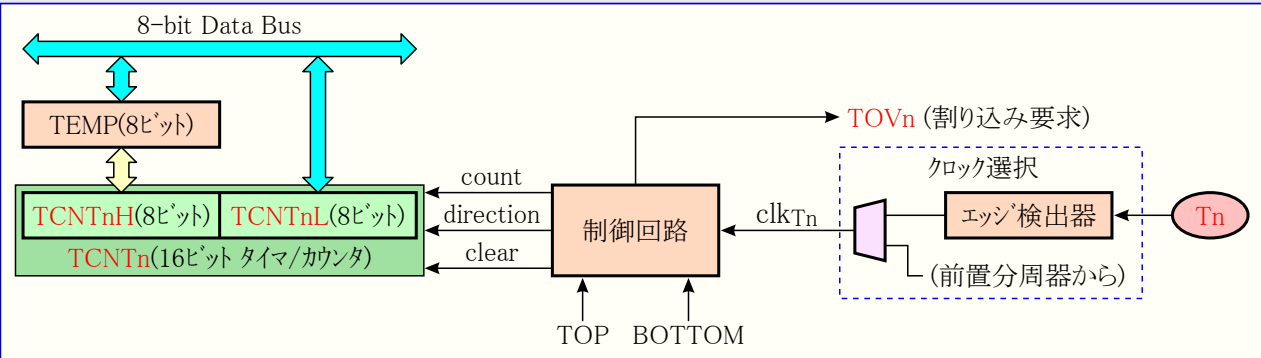
印加された外部クロックの各半周期は正しい採取を保証するために1システムクロック周期よりも長くなければなりません。この外部クロックは50%デューティ比で与えられるものとして、システムクロック周波数の半分未満($f_{EXTclk} < f_{clk_I/O}/2$)であることが保証されなければなりません。エッジ検出器が採取を使うため、検出できる外部クロックの最大周波数は採取周波数の半分です(ナイキストの標本化定理)。然しながら、発振元(水晶発振子、セラミック振動子、コンデンサ)の公差によって引き起こされたシステムクロック周波数やデューティ比の変動のため、外部クロック元の最大周波数は $f_{clk_I/O}/2.5$ 未満が推奨されます。

外部クロック元は前置分周できません。

15.8. 計数器部

16ビット タイマ/カウンタの主な部分は構成図で示されるような、設定可能な16ビット双方向カウンタ部です。

図15-4. カウンタ部構成図



注: レジスタとビットの名前での 'n' は装置番号(タイマ/カウンタ0についてはn=0)を表します。

| | | |
|----------------|-----------|----------------------------------|
| 信号説明 (内部信号) | count | TCNT0を1つ進めるまたは戻す信号。 |
| | direction | 進行方向(上昇または下降)選択信号。 |
| | clear | TCNT0のリセット(\$0000設定)信号。 |
| | clkTn | 以降でclkT0として参照されるタイマ/カウンタ クロック信号。 |
| | TOP | TCNT0が最大値に到達したことを示す信号。 |
| | BOTTOM | TCNT0が最小値(\$0000)に到達したことを示す信号。 |
| | TEMP | 一時レジスタ。 |

この16ビット カウンタはカウンタの上位8ビットを含むカウンタ上位(TCNT0H)と下位8ビットを含むカウンタ下位(TCNT0L)の2つの8ビット I/Oメモリ位置に配置されます。TCNT0HレジスタはCPUによる間接的なアクセスのみできます。CPUがTCNT0H I/O位置をアクセスする時に、CPUは上位バイト一時レジスタ(TEMP)をアクセスします。この一時レジスタはTCNT0Lが読まれる時にTCNT0H値で更新され、TCNT0Lが書かれる時にTCNT0Hは一時レジスタ値で更新されます。これは8ビット データ バス経由で1クロック周期内での16ビット カウンタ値全体の読み書きをCPUに許します。

注: カウンタ計数中と同時のTCNT0レジスタへの書き込みが予測不能な結果を生じる特別な場合があります。この特別な場合はそれらが重要となる項で記述されます。

選択した動作種別に依存して、カウンタは各タイマ/カウンタ クロック(clkT0)でリセット(\$0000)、増加(+1)、または減少(-1)されます。clkT0クロックはタイマ/カウンタ制御レジスタBのクロック選択(TCCR0B.CS02~0)ビットによって選択される、内部または外部のクロック元から生成できます。クロック元が選択されない(CS02~0=000)時にカウンタは停止されます。けれどもTCNT0値はタイマ/カウンタ クロック(clkT0)が存在するしないに拘らず、CPUによってアクセスできます。CPU書き込みは全てのカウンタ解除や計数動作を無視します(即ち、上位優先権を持ちます)。

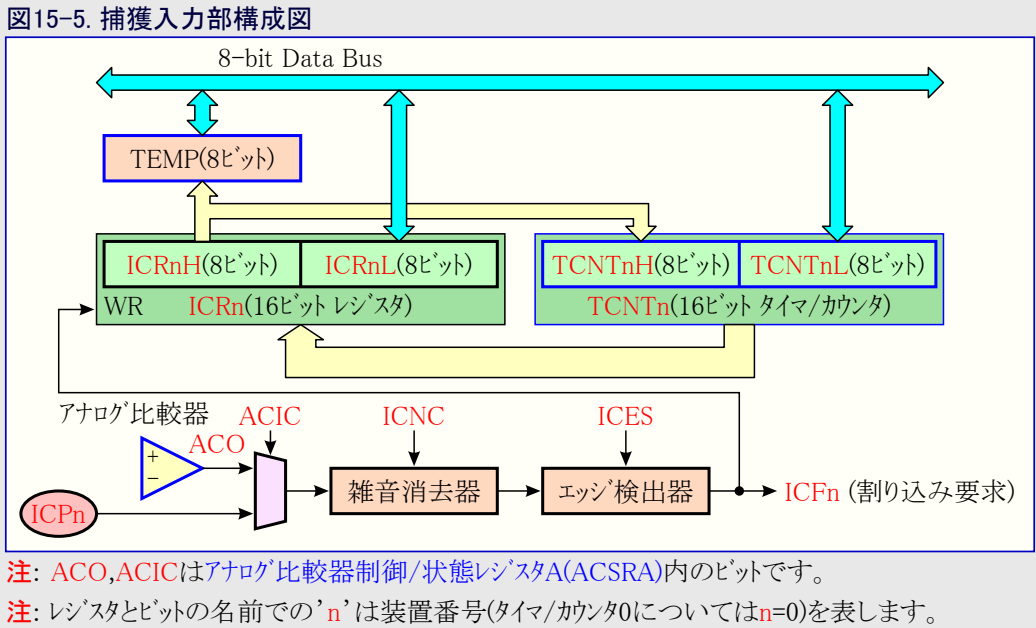
計数順序(方法)はタイマ/カウンタ制御レジスタA(TCCR0A.WGM01,0)とタイマ/カウンタ制御レジスタB(TCCR0B.WGM03,2)に配置された波形生成種別(WGM03~0)ビットの設定によって決定されます。これらはカウンタ動作(計数)方法と波形がOC0x比較出力に生成される方法間の接続に近いものです。進化した計数順序と波形生成についてより多くの詳細に対しては「動作種別」をご覧ください。

タイマ/カウンタ割り込み要求レジスタのタイマ/カウンタ0溢れ(TIFR0.TOV0)フラグはWGM03~0ビットによって選択された動作種別に従って設定(1)されます。TOV0はCPU割り込み発生に使えます。

15.9. 捕獲入力部

タイマ/カウンタ0は外部の出来事を捕獲でき、発生時間を示す時間印(タイマ/カウンタ値)を与える捕獲入力部と結合します。出来事または複数の出来事を示す外部信号はICP0ピンまたは代わりにアナログ比較器部経由で印加できます。時間印はその後、周波数、デューティ比、印加された信号の他の特性の計算に使えます。代わりに時間印は出来事の記録作成にも使えます。

捕獲入力部は下の構成図によって図解されます。直接的な捕獲入力部の部分でない構成図の要素は青枠(訳注:原文は灰色背景)で示されます。レジスタとビット名での小文字の'n'はタイマ/カウンタ番号を示します。



捕獲起動入力(ICP0)ピンまたは代わりにアナログ比較器出力(ACO)で論理レベルの変化(事象)が起き、その変化がエッジ検出器の設定を追認すると、捕獲が起動され、カウンタ(TCNT0)の16ビット値が捕獲レジスタ(ICR0)に書かれます。捕獲割り込み要求フラグ(ICF0)はTCNT0値がICR0に複写されるのと同じシステムクロック周期で設定(1)されます。許可(I=1, TMSK0.ICIE0=1)ならば捕獲割り込み要求フラグは捕獲割り込みを発生します。ICF0は割り込みが実行されると自動的に解除(0)されます。代わりにこのI/Oビット位置に論理1を書くことによってソフトウェアでも解除(0)できます。

捕獲レジスタ(ICR0)の16ビット値読み込みは、初めに下位バイト(ICR0L)、その後に上位バイト(ICR0H)を読むことによって行われます。ICR0Lから下位バイトが読まれる時に上位バイトが上位バイト一時レジスタ(TEMP)に複写されます。CPUがICR0H I/O位置を読むと、この一時レジスタをアクセスします。

ICR0はカウンタのTOP値定義にICR0を利用する波形生成種別を使う時にだけ書けます。これらの場合、TOP値がICR0に書かれ得る前に波形生成種別(WGM03~0)ビットが設定されなければなりません。ICR0に書く時は下位バイトがICR0Lに書かれる前に、上位バイトがICR0H I/O位置に書かれなければなりません。

関連リンク 52頁の「16ビット タイマ/カウンタ レジスタのアクセス」

15.9.1. 捕獲起動元

捕獲入力部用の主な起動元は捕獲起動入力(ICP0)ピンです。タイマ/カウンタ0は捕獲入力部用起動元として代わりにアナログ比較器出力を使えます。アナログ比較器はアナログ比較器制御/状態レジスタA(ACSRA)のアナログ比較器捕獲起動許可(ACIC)ビットの設定(=1)によって起動元として選択されます。起動元を変更することが捕獲を起動し得ることに気付いてください。従って捕獲割り込み要求フラグ(ICF0)は、その変更後に解除(0)されなければなりません。

捕獲起動入力(ICP0)ピンとアナログ比較器出力(ACO)の両入力は、T0ピンについてと同じ技法を使って採取されます。エッジ検出器も全く同じです。けれども雑音消去が許可されると、付加論理回路がエッジ検出器の前に挿入され、そして遅延を4システムクロック周期増します。タイマ/カウンタがTOP値定義にICR0を使う波形生成種別に設定されないなら、雑音消去器とエッジ検出器の入力が常に許可されます。

捕獲入力ICP0ピンのポートを制御することによってソフトウェアで起動できます。

関連リンク 95頁の「ACSRA - アナログ比較器制御/状態レジスタA」

15.9.2. 雑音消去器

雑音消去器は簡単なデジタル濾波器機構を使うことによって雑音耐性を改善します。雑音消去器入力は4採取に渡って監視され、エッジ検出器によって使われる交互出力を変更するには4つ全てが等しくなければなりません。

雑音消去器はタイマ/カウンタ制御レジスタBの捕獲入力雑音消去許可(TCCR0B.ICNC0)ビットの設定(1)によって許可されます。許可時、雑音消去器は入力に印加される変更とICR0が更新される間に4システムクロック周期の追加遅延をもたらします。

雑音消去器はシステムクロックを使い、従って前置分周器によって影響を及ぼされません。

15.9.3. 捕獲入力の使用

捕獲入力機能を使う主な要求(目的)は入って来る出来事に対して十分なプロセッサ能力を当てがうことです。2つの出来事間の時間が際どいと思います。次の出来事が起こる前に捕獲した**捕獲レジスタ(ICR0)**の値をプロセッサが読めなかった場合、ICR0は新しい値で上書きされます。この場合、捕獲の結果は不正にされます。

捕獲割り込みを使う時にICR0は割り込み処理ルーチンで可能な限り早く読まれるべきです。捕獲割り込みが相対的に高い優先順位であっても、最大割り込み応答時間は他の割り込み要求のどれかを扱うのに必要とされる最大クロック周期数に依存します。

動作中に**TOP値(分解能)**が積極的に変更されるとき、どの動作種別での捕獲入力部の使用も推奨されません。

外部信号のデューティ比測定は各捕獲後に起動端が変更されることを必要とします。検出端の変更はICR0が読まれてしまった後に可能な限り早く行われなければなりません。エッジの変更後、**捕獲割り込み要求フラグ(ICF0)**はソフトウェア(I/Oビット位置への論理1書き込み)によって解除(0)されなければなりません(訳補:エッジ変更によってICF0が設定(1)されることを想定)。周波数のみの測定について(割り込み処理が使われる場合)、ICF0の解除(0)は必要とされません。

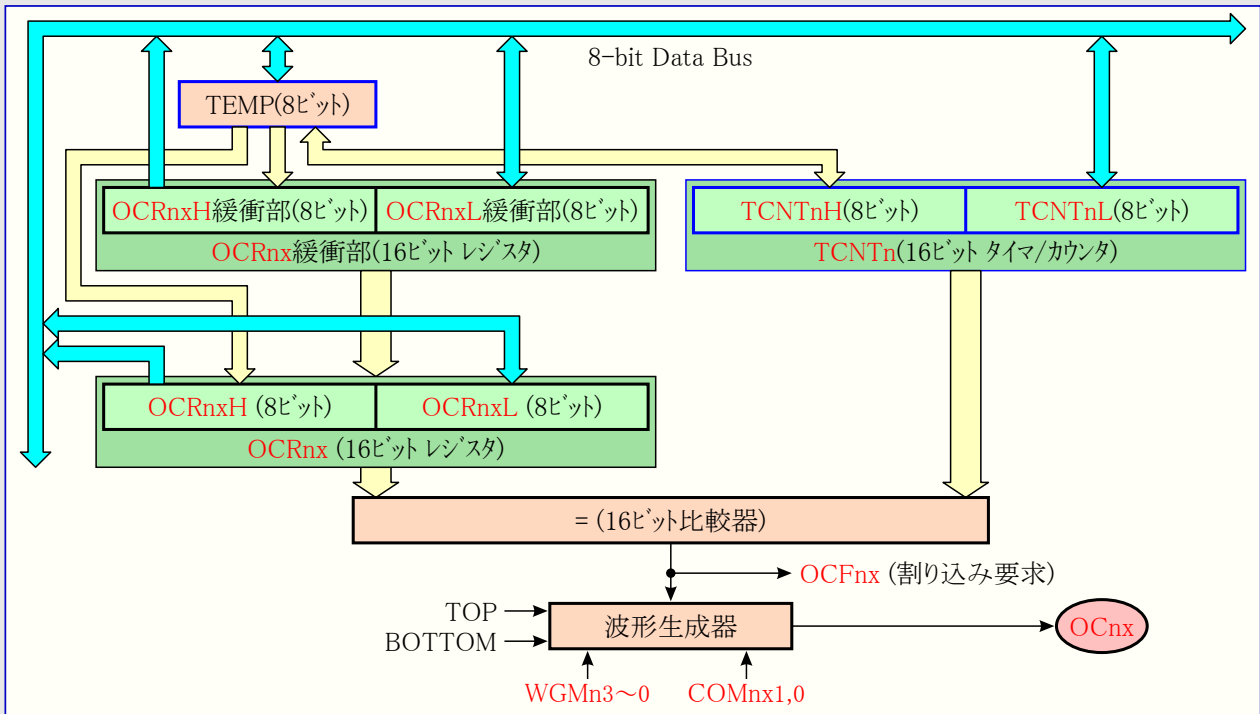
15.10. 比較出力部

この16ビット比較器は**TCNT0**と**比較レジスタ(OCR0x)**を継続的に比較します。TCNT0とOCR0xが等しければ、比較器は一致を指示します。この一致は**次のタイマ/カウンタクロック周期**で**比較割り込み要求フラグ(OCF0x)**を設定(1)します。許可(I=1, **TIMSK0.OCIE0x=1**)なら、この比較割り込み要求フラグは比較割り込みを発生します。OCF0xは割り込みが実行されると自動的に解除(0)されます。代わりにOCF0xはこのI/Oビット位置に論理1を書くことによってソフトウェアでも解除(0)できます。波形生成器は**波形生成種別(WGM03~0)ビット**と**比較出力選択(COM0x1,0)ビット**によって設定された動作種別に従った出力を生成するのにこの一致信号を使います。TOPとBOTTOM信号は動作種別のいくつかで両端値の特別な場合を扱うため、波形生成器によって使われます。「**動作種別**」をご覧ください。

比較A出力部の特殊な特性はタイマ/カウンタのTOP値(換言するとカウンタの分解能)定義を許します。カウンタの分解能に加え、TOP値は波形生成器によって生成された波形の周期時間を定義します。

下は比較出力部の構成図を示します。直接的な比較出力部の部分でない構成図の要素は青枠(訳注:原文は灰色背景)で示されます。

図15-6. 比較出力部構成図



注: レジスタとビットの名前での'n'はタイマ/カウンタ番号(タイマ/カウンタ0に対してはn=0)、『x'は比較出力部(AまたはB)を表します。

OCR0xは12種類のパルス幅変調(PWM)のどれかを使う時に2重緩衝されます。標準動作と比較一致タイマ/カウンタ解除(CTC)動作については2重緩衝動作が禁止されます。2重緩衝動作は計数の流れのTOPまたはBOTTOMのどちらかに対してOCR0xレジスタの更新を同期化します。この同期化は奇数長、非対称PWMパルスの発生を防ぎ、それによって不具合なしの出力を作成します。

2重緩衝動作が許可されると、CPUはOCR0x緩衝器をアクセスします。2重緩衝動作が禁止されるとOCR0xレジスタを直接アクセスします。

OCR0x(緩衝器またはレジスタ)の内容は書き込み操作によってのみ変更されます(タイマ/カウンタはTCNT0やICR0のようにOCR0xを自動的に更新しません)。従ってOCR0xは上位バイト一時レジスタ(TEMP)経由で読まれません。けれども他の16ビットレジスタをアクセスする時のように下位バイトを先に読むのは良い習慣です。OCR0x書き込みは16ビット全ての比較が継続的に行われるため、一時レジスタ経由で行われなければなりません。上位バイト(OCR0xH)が先に書かれなければなりません。上位バイトI/O位置がCPUによって書かれると、一時レジスタは書かれた値で更新されます。その後下位バイト(OCR0xL)が下位8ビットを書かれると、(一時レジスタ内の)上位バイトは(下位バイト書き込みと同じシステムクロック周期でOCR0x緩衝器またはOCR0xレジスタのどちらかに複写されます。

16ビットレジスタアクセス法のより多くの情報については「16ビットタイマ/カウンタレジスタのアクセス」を参照してください。

(訳注) ここでは比較nxレジスタ全体をOCRnx、OCRnxを構成する緩衝器部分をOCRnx緩衝器、実際の比較に使われるレジスタ本体部分をOCRnxレジスタとして記述しています。他の部分での記述でも特に必要がある場合はこの記述方法を適用します。

15.10.1. 強制比較出力

非PWM波形生成動作での比較器の一致出力は強制変更(TCCR0C.FOC0x)ビットに1を書くことによって強制(変更)できます。比較一致の強制は比較割り込み要求フラグ(OCF0x)の設定(1)やタイマ/カウンタの再設定/解除を行いませんが、OC0xピンは実際の比較一致が起きた場合と同様に更新されます(TCCR0A.COM0x1,0ビット設定がOC0xピンの設定(1)、解除(0)、1/0交互のどれかを定義)。

15.10.2. TCNT0書き込みによる比較一致妨害

TCNT0への全てのCPU書き込み操作は、例えばタイマ/カウンタが停止されていても、次のタイマ/カウンタクロック周期で起こるどんな比較一致をも妨げます。この特質はタイマ/カウンタクロックが許可されている時に、割り込みを起動することなく、TCNT0と同じ値に初期化されることをOCR0xに許します。

15.10.3. 比較一致部の使用

どの動作種別でのTCNT0書き込みでも1タイマ/カウンタクロック周期間、全ての比較一致を妨げるため、タイマ/カウンタが走行中であるか否かに拘らず、何れかの比較出力部を使う場合、TCNT0を変更する時に危険を伴います。TCNT0に書かれた値がOCR0x値と同じ場合、比較一致は失われ(一致が発生せず)、不正な波形生成に終わります。可変TOP値のPWM動作でTOPに等しいTCNT0を書いてはいけません。(行った場合)TOPに対する比較一致は無視され、カウンタは\$FFFFへ(計数を)続けます。同様にカウンタが下降計数のとき、BOTTOMに等しいTCNT0値を書いてはいけません。

OC0xの初期設定はポートピンに対するポート方向レジスタを出力に設定する前に行われるべきです。OC0x値を設定する一番簡単な方法は標準動作で強制変更(FOC0x)ストローブビットを使うことです。波形生成種別間を変更する時であっても、OC0x(内部)レジスタはそれらの値を保ちます。

比較出力選択(TCCR0A.COM0x1,0)ビットが比較値(OCR0x)と共に2重緩衝されないことに気付いてください。TCCR0A.COM0x1,0ビットの変更は直ちに有効となります。

15.11. 比較一致出力部

タイマ/カウンタ制御レジスタAの比較出力選択(TCCR0A.COM0x)ビットは次の2つの機能を持ちます。

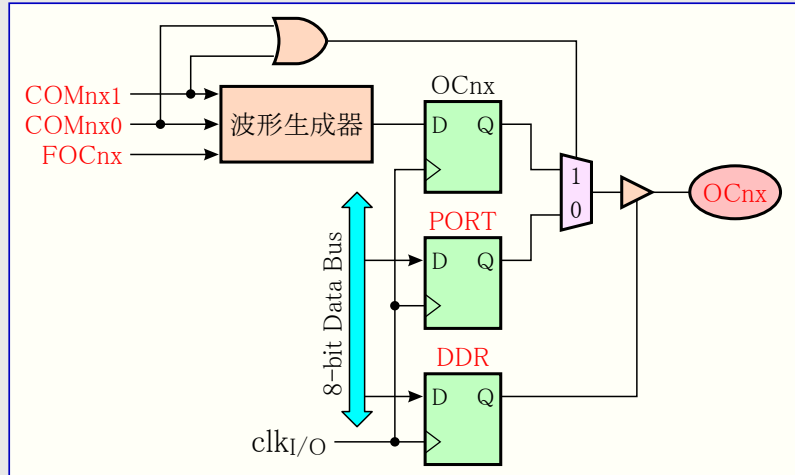
- ・ 波形生成器は次の比較一致での比較出力(OC0x)状態の定義にCOM0xビットを使います。
- ・ COM0xビットはOC0xピン出力元を制御します。

下図はCOM0xビット設定によって影響される論理回路の簡単化した図を示します。図のI/Oレジスタ、I/Oビット、I/Oピンは赤文字(訳注: 原文は太字)で示されます。COM0xビットによって影響を及ぼされる標準I/Oポート制御レジスタ、即ちPORTとDDRの部分だけが示されます。

システムリセットが起こると、OC0xレジスタは\$00にリセットされます。

注: 'OC0xの状態'は常にOC0xピンでなく内部OC0xレジスタを参照します。

図15-7. 比較一致出力回路図 (非PWM動作)



注: レジスタとビットの名前での'n'はタイマ/カウンタ番号(タイマ/カウンタ0に対してはn=0)、『x'は比較出力部(AまたはB)を表します。

COM0x1,0ビットのどちらかが設定(1)されると、標準I/Oポート機能は波形生成器からの比較出力(OC0x)によって無効にされます。けれどもOC0xピンの方向(入出力)はポートピンに対するポート方向レジスタ(DDR)によって未だ制御されます。ポート方向レジスタでは、OC0x値がピンで見えるのに先立って、OC0xピンに対するビット(DDR_OC0x)が出力として設定されなければなりません。このポートの交換機能は波形生成種別と無関係です。

比較出力ピン論理回路の設計は出力が許可される前のOC0x状態の初期化を許します。いくつかのCOM0x1,0ビット設定が或る種の動作種別に対して予約されます。

TCCR0A.COM0x1,0ビットは捕獲入力部で何の効果もありません。

15.11.1. 比較一致出力選択と波形生成

波形生成器は標準、CTC、PWM動作でTCCR0A.COM0x1,0ビットを違うふうに使います。全ての動作種別に対してTCCR0A.COM0x1,0=00設定は次の比較一致で実行すべきOC0xレジスタの動きがないことを波形生成器へ告げます。出力動作種別も参照してください。

TCCR0A.COM0x1,0ビットの状態変更はこのビットが書かれた後の最初の比較一致で有効になります。非PWM動作について、この動作は強制変更(TCCR0C.FOC0x)スローブビットを使うことによって直ちに効果を得ることを強制できます。

15.12. 動作種別

動作種別、換言するとタイマ/カウンタと比較出力ピンの動作は**波形生成種別(WGM03~0)**ビットと**比較出力選択(TCCR0A.COM0x1,0)**ビットの組み合わせによって定義されます。比較出力選択ビットは計数順序(動作)に影響を及ぼしませんが、一方波形生成種別ビットは影響を及ぼします。TCCR0A.COM0x1,0ビットは生成されたPWM出力が反転されるべきか否か(反転または非反転PWM)のどちらかを制御します。非PWM動作に対するTCCR0A.COM0x1,0ビットは比較一致で出力が解除(0)、設定(1)、1/0交互のどれにされるべきかを制御します。

15.12.1. 標準動作

最も単純な動作種別が標準動作(WGM03~0=0000)です。この動作種別での計数方向は常に上昇(+)で、カウンタの解除は実行されません。カウンタは16ビット最大値(MAX=\$FFFF)を通過すると単に範囲を超え、そしてBOTTOM=\$0000から再び始めます。通常動作での**タイマ/カウンタ溢れ(TOV0)フラグ**はTCNT0が\$0000になる時と同じタイマ/カウンタクロック周期で設定(1)されます。この場合のTOV0フラグは設定(1)のみで解除(0)されないことを除いて第17ビットのようになります。けれどもTOV0フラグを自動的に解除(0)するタイマ/カウンタ溢れ割り込みと組み合わせたタイマ/カウンタの分解能はソフトウェアによって増やせます。標準動作での考慮に特別な場合はなく、新しいカウンタ値は何時でも書けます。

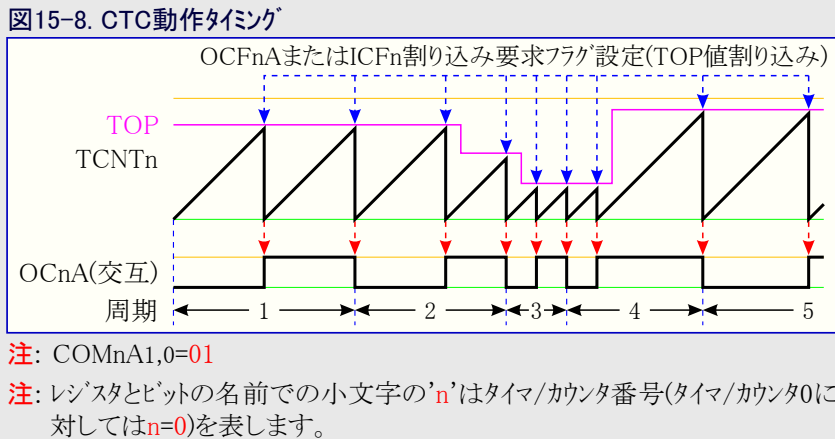
捕獲入力部は標準動作での使用が容易です。けれども外部の事象間の最大間隔がタイマ/カウンタの分解能(16ビット長)を越えてはならないことに気付いてください。事象間の間隔が長すぎる場合、捕獲部に対して分解能を拡張するために、タイマ/カウンタ溢れ割り込みまたは前置分周器が使われなければなりません。

比較出力部は与えられた或る時間に割り込みを生成するのに使えます。標準動作で波形を生成するのに比較出力を使うのは、それが大変多くのCPU時間を占有するため推奨されません。

15.12.2. 比較一致タイマ/カウンタ解除(CTC)動作

比較一致タイマ/カウンタ解除(CTC)動作(動作番号4または12、WGM03~0=0100または1100)ではOCR0AまたはICR0がカウンタの分解能を操作するのに使われます。カウンタ(TCNT0)値はOCR0A(WGM03~0=4)またはICR0(WGM03~0=12)のどちらかと一致する時にカウンタが\$0000に解除されます。OCR0AまたはICR0はカウンタに対する**TOP値**、従って分解能も定義します。この動作種別はより大きい比較一致出力周波数の制御を許します。それは外部の出来事の計数操作も簡単にします。

CTC動作についてのタイミング図は下で示されます。カウンタ(TCNT0)値はOCR0AまたはICR0のどちらかで比較一致が起こるまで増加し、そしてその後にTCNT0は解除(\$0000)されます。



実際のCTC動作に依存して、OCF0AまたはICF0のどちらかを使うことにより、カウンタ値がTOP値に到達する時毎に割り込みを生成できます。割り込みが許可されるなら、割り込み処理ルーチンはTOP値を更新するのに使えます。

注: カウンタが走行している間にBOTTOMと近い値にTOPを変更するのは、CTC動作が2重緩衝を提供しないために注意して行わなければなりません。OCR0AまたはICR0に書かれた新しい値がTCNT0の現在値よりも低い(小さい)場合、カウンタは(その回の)比較一致を失います。その後カウンタは比較一致が起こるのに先立って、最大値(\$FFFF)へそして\$0000から始まる循環を計数します。

多くの場合でこの特性は好ましくありません。OCR0Aが2重緩衝されるので、代替はTOPを定義するのにOCR0Aを用いる**高速PWM動作(WGM03~0=1111)**を使うことでしょう。

CTC動作で波形出力を生成するため、OC0A出力は**比較出力選択(COM0A1,0)**ビットを交互動作(=01)に設定することによって各比較一致での論理レベル交互切替に設定できます。OC0A値はそのピンに対するデータ方向が出力(DDR_OC0A=1)に設定されない限り、ポートピンで見えないでしょう。生成された波形はOCR0Aが0(\$0000)に設定される時に、 $f_{OC0A} = f_{clk_I/O} / 2$ の最大周波数を得ます。生成波形周波数は次式によって定義されます。

$$f_{OCnA} = \frac{f_{clk_I/O}}{2 \times N \times (1 + OCRnA)}$$

注: ・レジスタとビットの名前での小文字の'n'はタイマ/カウンタ番号(タイマ/カウンタ0に対してはn=0)を表します。
・Nは前置分周数(1,8,64,256,1024)を表します

標準動作と同じように、**タイマ/カウンタ溢れ割り込み要求(TOV0)フラグ**はカウンタがMAXから\$0000へ計数するのと同じタイマ/カウンタクロック周期で設定(1)されます。

15.12.3. 高速PWM動作

高速パルス幅変調(PWM)動作(動作番号5,6,7,14,15、WGM03~0=0101,0110,0111,1110,1111)は高周波数PWM波形生成選択を提供します。高速PWMはそれが単一傾斜(鋸波)動作であることによって他のPWM動作と異なります。カウンタはBOTTOMからTOPまで計数し、その後BOTTOMから再び始めます。

非反転比較出力動作(COM0x1,0=10)での比較出力(OC0x)はTCNT0とOCR0x間の比較一致で解除(0)され、BOTTOMで設定(1)されます。反転出力動作(COM0x1,0=11)の出力は比較一致で設定(1)され、BOTTOMで解除(0)されます。単一傾斜動作のため、高速PWM動作の動作周波数は両傾斜(三角波)動作を使う位相基準や位相/周波数基準PWM動作よりも2倍高くできます。この高い周波数は電力調節、整流、D/A変換に対して高速PWM動作を都合よく適合させます。高い周波数は物理的に小さな外部部品(コイルやコンデンサ)を許し、従ってシステム総費用を削減します。

高速PWMのPWM分解能は8,9,10ビットに固定、若しくはOCR0AかICR0のどちらかによって定義できます。許された最小分解能は2ビット(OCR0AまたはICR0が\$0003設定)、最大分解能は16ビット(OCR0AまたはICR0がMAX設定)です。ビットでのPWM分解能は右式を使うことによって計算できます。

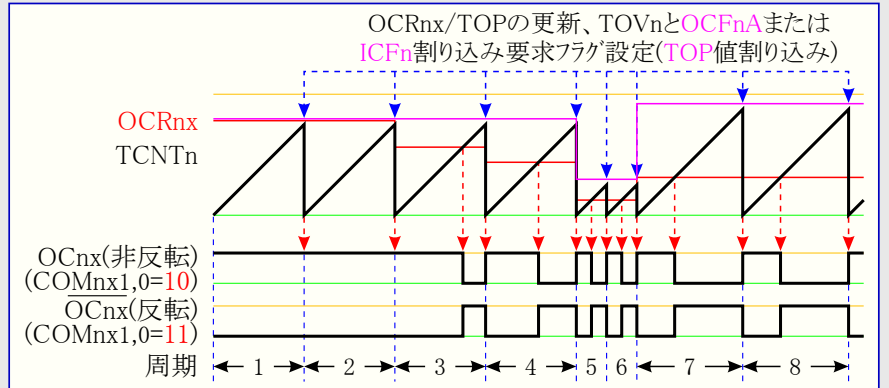
$$R_{FPWM} = \frac{\log(TOP+1)}{\log 2}$$

高速PWM動作でのカウンタはカウンタ値が固定値\$00FF,\$01FF,\$03FF(WGM03~0=0101,0110,0111)、ICR0値(WGM03~0=1110)またはOCR0A値(WGM03~0=1111)のどれかと一致するまで増加されます。そしてカウンタは(一致の)次のタイマ/カウンタクロック周期で解除(\$0000)されます。TOPを定義するのにOCR0AかICR0を使う高速PWM動作のタイミング図が下で示されます。TCNT0値はタイミング図で単一傾斜動作(鋸波)を表す折れ線グラフとして示されます。本図は非反転と反転のPWM出力を含みます。細い赤線はOCR0x値を示し、TCNT0値との交点(接点)がTCNT0とOCR0x間の比較一致を表します(注:図補正に伴い本行若干変更)。比較割り込み要求フラグ(OCF0x)は比較一致が起こると設定(1)されます。

タイマ/カウンタ溢れ割り込み要求(TOV0)フラグはカウンタがTOPに到達する時毎に設定(1)されます。加えて、OCR0AかICR0のどちらかがTOP値を定義するのに使われると、OCF0AまたはICF0割り込み要求フラグはTOV0が設定(1)されるのと同じタイマ/カウンタクロック周期で設定(1)されます。これらの割り込みの1つが許可されるなら、その割り込み処理ルーチンはTOPと比較値を更新するのに使えます。

TOP値を変更する時に、プログラムは新TOP値が全ての比較レジスタ値と等しいか大きいことを保証しなければなりません。TOP値が何れかの比較レジスタよりも小さな場合、TCNT0とそのOCR0x間で比較一致は決して起きません。固定TOP値を使う場合、どのOCR0xが書かれる時も、未使用ビットが0で隠(に置換)されることに注意してください。

図15-9. 高速PWM動作タイミング



注: レジスタとビットの名前での'n'はタイマ/カウンタ番号(タイマ/カウンタ0に対してはn=0)、『x'は比較出力部(AまたはB)を表します。

ICR0がTOP値を定義するのに使われる時はICR0を更新する手順がOCR0Aの更新と異なります。ICR0は2重緩衝されません。これは前置分周なしまたは低い前置分周値でカウンタが走行している時にICR0が小さな値に変更される場合、書かれた新しいICR0値がTCNT0の現在値よりも小さくなる危険を意味します。結果として、カウンタは(その回の)TOP値での比較一致を失います。その後のカウンタは比較一致を起こせるのに先立ってMAX値(\$FFFF)へそして次に\$0000から始める計数をしなければならないでしょう。しかし、OCR0Aは2重緩衝されます。この特徴はOCR0AのI/O位置に何時でも書かれることを許します。OCR0A I/O位置が書かれると、書かれた値はOCR0A緩衝器に置かれます。OCR0A(比較)レジスタはその後TCNT0がTOPと一致した次のタイマ/カウンタクロック周期にOCR0A緩衝器値で更新されます。この更新はTCNT0の解除(\$0000)やTOV0の設定(1)と同じタイマ/カウンタクロック周期で行われます。

TOPを定義するのにICR0を使うことは決まったTOP値を使う時に上手くいきます。ICR0を使うことにより、OC0AでのPWM出力を生成するためにOCR0Aが自由に使えます。けれども基準PWM周波数が(TOP値を変更することによって)動的に変更される場合、OCR0Aが2重緩衝機能のため、TOPとしてOCR0Aを使うことは明らかに良い選択です。

高速PWM動作での比較部はOC0xピンでのPWM波形の生成を許します。COM0x1,0ビットへの'10'書き込みは非反転PWM出力を作成し、反転PWM出力はCOM0x1,0への'11'書き込みで生成できます。実際のOC0x値はそのポートピンに対するデータ方向が出力(DDR_OC0x=1)として設定される場合にだけ見えるでしょう。PWM波形はTCNT0とOCR0x間の比較一致でOC0x(内部)レジスタを設定(1)(または解除(0))と、カウンタが解除(\$0000、TOPからBOTTOMへ変更)されるタイマ/カウンタクロック周期でのOC0xレジスタを解除(0)(または設定(1))することによって生成されます。

PWM出力周波数は次式によって計算できます。

$$f_{OCnxPWM} = \frac{f_{clk,I/O}}{N \times (1 + TOP)}$$

注: ・ レジスタとビットの名前での'n'はタイマ/カウンタ番号(タイマ/カウンタ0に対してはn=0)、『x'は比較出力部(AまたはB)を表します。
・ Nは前置分周数(1,8,64,256,1024)を表します

OCR0xの両端値は高速PWM動作でPWM波形出力を生成する時の特別な場合に当たります。OCR0xがBOTTOM(\$0000)に等しく設定されると、出力はTOP+1 タイマ/カウンタクロック周期毎の狭い尖頭(パルス)になるでしょう。TOPに等しいOCR0x設定は(COM0x1,0ビットによって制御される出力極性に依存して)定常的なLowまたはHigh出力に終わるでしょう。

50%デューティ比での周波数波形出力は高速PWM動作で比較一致毎にその論理レベルを交互反転するOC0A設定(COM0A1,0=01)によって達成できます。これはTOP値を定義するのにOCR0Aが使われる(WGM03~0=1111)の場合にだけ適用されます。生成された波形はOCR0Aが0(\$0000)に設定される時に $f_{OC0A} = f_{clk_I/O} / 2$ の最大周波数を持ちます。この特性は高速PWM動作で比較出力部の2重緩衝機能が許可されることを除いて、CTC動作でのOC0A交互出力(COM0A1,0=01)と同じです。

15.12.4. 位相基準PWM動作

位相基準パルス幅変調(PWM)動作(WGM03~0=0001,0010,0011,1010,1011)は高分解能で正しい位相のPWM波形生成選択を提供します。位相基準PWM動作は両傾斜(三角波)動作を基準とした位相/周波数基準PWMと似ています。カウンタはBOTTOM(\$0000)からTOPへ、そしてその後にTOPからBOTTOMへを繰り返し計数します。非反転比較出力動作(COM0x1,0=10)での比較出力(OC0x)は上昇計数中のTCNT0とOCR0x間の比較一致で解除(0)され、下降計数中の比較一致で設定(1)されます。反転出力動作(COM0x1,0=11)での動作は逆にされます。両傾斜(三角波)動作は単一傾斜(鋸波)動作よりも低い最大動作周波数になります。けれども両傾斜(三角波)PWM動作の対称特性のため、これらの動作種別は電動機制御の応用に好まれます。

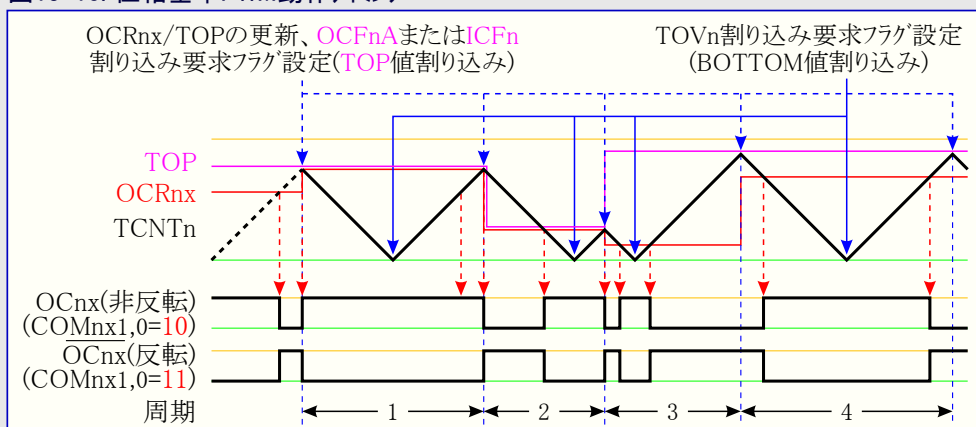
位相基準PWM動作のPWM分解能は8,9,10ビット固定、またはOCR0AかICR0のどちらかによって定義できます。許された最小分解能は2ビット(OCR0AまたはICR0が\$0003設定)、最大分解能は16ビット(OCR0AまたはICR0がMAX設定)です。ビットでのPWM分解能は右式を使うことによって計算できます。

$$R_{PCPWM} = \frac{\log(TOP+1)}{\log 2}$$

位相基準PWM動作でのカウンタはカウンタ値が固定値\$00FF,\$01FF,\$03FF(WGM03~0=0001,0010,0011)、ICR0値(WGM03~0=1010)またはOCR0A値(WGM03~0=1011)のどれかと一致するまで増やされます。カウンタはTOPに到達したその時に計数方向を変えます。このTCNT0値は1タイマ/カウンタクロック周期、TOPと等しくなります。TOPを定義するのにOCR0AまたはICR0が使われる位相基準PWM動作のタイミング図が下で示されます。TCNT0値はタイミング図で両傾斜動作(三角波)を表す折れ線グラフとして示されます。この図は非反転と反転のPWM出力を含みます。細い赤線はOCR0x値を示し、TCNT0値との交点(接点)がTCNT0とOCR0x間の比較一致を示します(訳注:図補正に伴い本行若干変更)。比較割り込み要求フラグ(OCF0x)は比較一致が起こると設定(1)されます。

タイマ/カウンタ溢れ(TOV0)フラグはカウンタがBOTTOMに到達する時毎に設定(1)されます。OCR0AかICR0のどちらかがTOP値を定義するのに使われると、OCF0AまたはICF0割り込み要求フラグはOCR0xレジスタが(TOPに於いて)2重緩衝器値で更新されるのと同じタイマ/カウンタクロック周期によって設定(1)されます。これらの割り込み要求フラグはカウンタがTOPまたはBOTTOM値に到達する毎に割り込みを発生するのに使えます。

図15-10. 位相基準PWM動作タイミング



注: レジスタとビットの名前での'n'はタイマ/カウンタ番号(タイマ/カウンタ0に対してはn=0)、『x'は比較出力部(AまたはB)を表します。

TOP値を変更する時にプログラムは新TOP値が全ての比較レジスタ値と等しいか大きいことを保証しなければなりません。TOP値が何れかの比較レジスタよりも小さな場合、TCNT0とそのOCR0x間で比較一致は決して起きません。固定TOP値を使う場合、どのOCR0xが書かれる時も、未使用ビットが0で隠(に置換)されることに注意してください。タイミング図の第3周期によって図解されるように、タイマ/カウンタが位相基準PWM動作で走行中にTOPを積極的に変更するのは、非対称出力で終わることが有り得ます。これに対する理由はOCR0xレジスタの更新時に見出せます。OCR0x更新はTOPで起きるので、PWM周期はTOPで始まりそして終わります。これは下降傾斜長が直前のTOP値によって決定され、一方上昇傾斜長は新しいTOP値で決定されることを意味します。これら2つの値(TOP)が違うとき、その周期の2つの傾斜長は異なるでしょう。この長さの相違が出力での非対称な結果を生じます。

タイマ/カウンタが走行中にTOP値を変更する場合、位相基準PWM動作の代わりに位相/周波数基準PWM動作を使うことが推奨されます。一定のTOP値を使う時に2つの動作種別間に現実的な違いはありません。

位相基準PWM動作での比較部はOC0xピンでのPWM波形の生成を許します。COM0x1,0ビットへの'10'書き込みは非反転PWM出力を作成し、反転PWM出力はCOM0x1,0への'11'書き込みで生成できます。実際のOC0x値はそのポートピンに対するデータ方向が出力(DDR_OC0x=1)として設定される場合にだけ見えるでしょう。PWM波形はカウンタが増加する時のTCNT0とOCR0x間の比較一致でOC0x(内部)レジスタを設定(1)(または解除(0))と、カウンタが減少する時のTCNT0とOCR0x間の比較一致でOC0xレジスタを解除(0)(または設定(1))することによって生成されます。位相基準PWMを使う時の出力に対するPWM周波数は次式によって計算できます。

$$f_{OCnxPCPWM} = \frac{f_{clk_I/O}}{2 \times N \times TOP}$$

注: ・レジスタとビットの名前での'n'はタイマ/カウンタ番号(タイマ/カウンタ0に対してはn=0)、『x'は比較出力部(AまたはB)を表します。

・Nは前置分周数(1,8,64,256,1024)を表します

OCR0xの両端値は位相基準PWM動作でPWM波形出力を生成する時の特別な場合に当たります。非反転PWM動作ではOCR0xがBOTTOMに等しく設定されると出力は定常的なLow、TOPに等しく設定されると定常的なHighになるでしょう。反転PWMに対する出力は逆の論理値になります。TOP値定義にOCR0Aが使われ(WGM03~0=1011)、COM0A1,0=01なら、OC0A出力はデューティ比50%で交互に変化します。

15.12.5. 位相/周波数基準PWM動作

位相/周波数基準パルス幅変調(PWM)動作(WGM03~0=1000,1001)は高分解能で正しい位相と周波数のPWM波形生成選択を提供します。位相/周波数基準PWM動作は両傾斜(三角波)動作を基準とした**位相基準PWM**と似ています。カウンタはBOTTOM(\$0000)からTOPへ、そしてその後にTOPからBOTTOMへを繰り返し計数します。非反転比較出力動作(COM0x1,0=10)での比較出力(OC0x)は上昇計数中のTCNT0とOCR0x間の比較一致で解除(0)され、下降計数中の比較一致で設定(1)されます。反転出力動作(COM0x1,0=11)での動作は逆にされます。両傾斜(三角波)動作は単一傾斜(鋸波)動作よりも低い最大動作周波数になります。けれども両傾斜(三角波)PWM動作の対称特性のため、これらの動作種別は電動機制御の応用に好まれます。

位相基準と位相/周波数基準PWM動作間の主な違いはOCR0xレジスタがOCR0x緩衝部によって更新される時(訳補:TOPとBOTTOM)です(図15-10と下のタイミング図をご覧ください)。

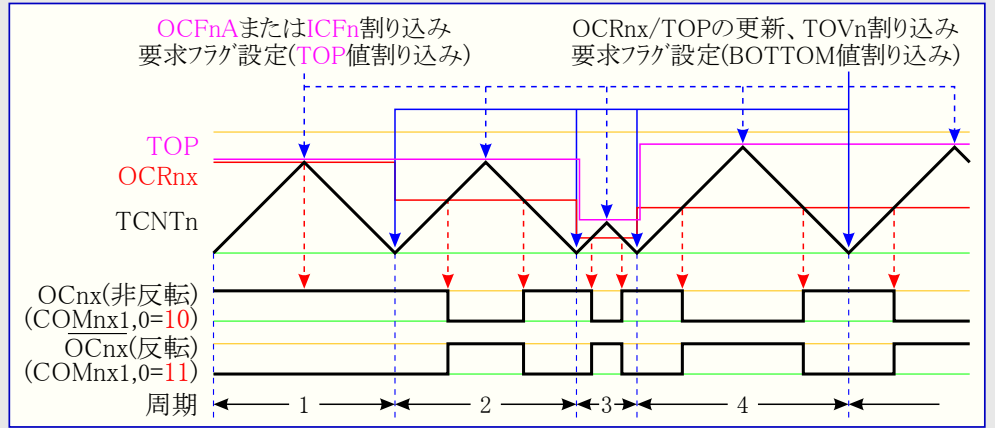
位相/周波数基準PWM動作のPWM分解能はOCR0AかICR0のどちらかで定義できます。許された最小分解能は2ビット(OCR0AまたはICR0が\$0003設定)、最大分解能は16ビット(OCR0AまたはICR0がMAX設定)です。ビットでのPWM分解能は右式を使うことによって計算できます。

$$R_{PFCPWM} = \frac{\log(TOP+1)}{\log 2}$$

位相/周波数基準PWM動作でのタイマ/カウンタはタイマ/カウンタ値がICR0値(WGM03~0=1000)かOCR0A値(WGM03~0=1001)のどちらかと一致するまで増加されます。カウンタはTOPに到達したその時に計数方向を変更します。このTCNT0値は1タイマ/カウンタクロック周期間、TOPと等価です。位相/周波数基準PWM動作のタイミング図が下で示されます。この図はOCR0AかICR0がTOPを定義するのに使われる時の位相/周波数基準PWM動作を示します。TCNT0値はタイミング図で両傾斜動作(三角波)を表す折れ線グラフとして示されます。この図は非反転と反転のPWM出力を含みます。細い赤線はOCR0x値を示し、TCNT0値との交点(接点)がTCNT0とOCR0x間の比較一致を示します(訳注:図補正に伴い本行若干変更)。比較割り込み要求フラグ(OCF0x)は比較一致が起こると設定(1)されます。

タイマ/カウンタ溢れ(TOV0)フラグはOCR0xレジスタが(BOTTOMで)2重緩衝器値で更新されるのと同じタイマ/カウンタクロック周期で設定(1)されます。TOP値を定義するのにOCR0AまたはICR0のどちらかが使われると、タイマ/カウンタがTOPに到達する時毎にOCF0AまたはICF0割り込み要求フラグが設定(1)されます。これらの割り込み要求フラグはカウンタがTOPまたはBOTTOM値に到達する毎に割り込みを発生するのに使えます。

図15-11. 位相/周波数基準PWM動作タイミング



注: レジスタとビットの名前での'n'はタイマ/カウンタ番号(タイマ/カウンタ0に対してはn=0)、『x'は比較出力部(AまたはB)を表します。

TOP値を変更する時にプログラムは新TOP値が全ての比較レジスタ値と等しいか大きいことを保証しなければなりません。TOP値が何れかの比較レジスタよりも小さな場合、TCNT0とそのOCR0x間で比較一致は決して起きません。

上のタイミング図で示されるように、生成された出力は位相基準PWM動作と異なり、全ての周期で対称です。OCR0xレジスタがBOTTOMで更新されるため、上昇と下降の傾斜長は常に等しくなります。これが対称出力パルス、従って正しい周波数を与えます。

TOPを定義するのにICR0を使うことは決まったTOP値を使う時に上手くなります。ICR0を使うことにより、OC0AでのPWM出力を生成するためにOCR0Aが自由に使えます。けれども基準PWM周波数が(TOP値を変更することによって)動的に変更される場合、OCR0Aが2重緩衝機能のため、TOPとしてOCR0Aを使うことは明らかに良い選択です。

位相/周波数基準PWM動作での比較部はOC0xピンでのPWM波形の生成を許します。COM0x1,0ビットを'10'に設定することは非反転PWM出力を作成し、反転PWM出力はCOM0x1,0を'11'に設定することで生成できます(TCCR0A.COM0xの説明をご覧ください)。実際のOC0x値はそのポートピンに対するデータ方向が出力(DDR_OC0x=1)として設定される場合にだけ見えるでしょう。PWM波形はカウンタが増加する時のTCNT0とOCR0x間の比較一致でOC0x(内部)レジスタを設定(1)(または解除(0))と、カウンタが減少する時のTCNT0とOCR0x間の比較一致でOC0xレジスタを解除(0)(または設定(1))することによって生成されます。位相/周波数基準PWMを使う時の出力に対するPWM周波数は次式によって計算できます。

$$f_{OCnxPFCPWM} = \frac{f_{clk_1/O}}{2 \times N \times TOP}$$

注: レジスタとビットの名前での'n'はタイマ/カウンタ番号(タイマ/カウンタ0に対してはn=0)、『x'は比較出力部(AまたはB)を表します。

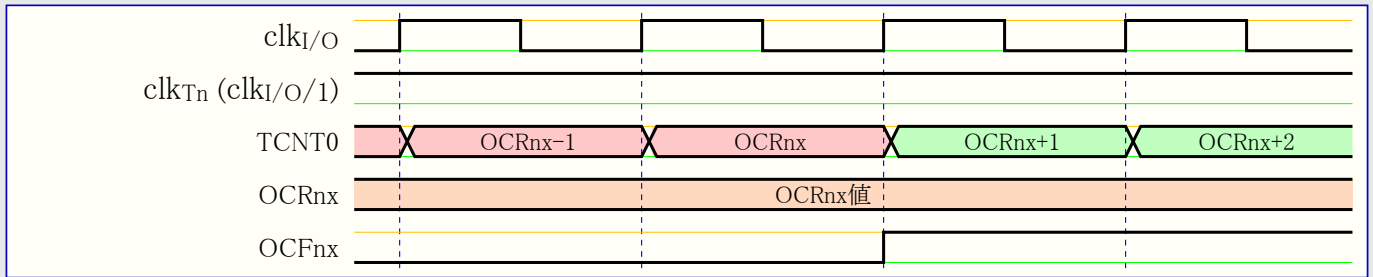
・Nは前置分周数(1,8,64,256,1024)を表します

OCR0xの両端値は位相/周波数基準PWM動作でPWM波形出力を生成する時の特別な場合に当たります。非反転PWM動作ではOCR0xがBOTTOMに等しく設定されると出力は定常的なLow、TOPに等しく設定されると定常的なHighになるでしょう。反転PWMに対する出力は逆の論理値になります。TOP値定義にOCR0Aが使われ(WGM03~0=1001)、COM0A1,0=01なら、OC0A出力はデューティ比50%で交互に変化します。

15.13. タイマ/カウンタのタイミング

このタイマ/カウンタは同期設計で、従ってタイマ/カウンタクロック(clk_{T0})が下図のクロック許可信号として示されます。この図は割り込みフラグが設定(1)される時、そして OCR0x レジスタが OCR0x 緩衝器値で更新される時(2重緩衝を使う動作種別のみ)の情報を含みます。

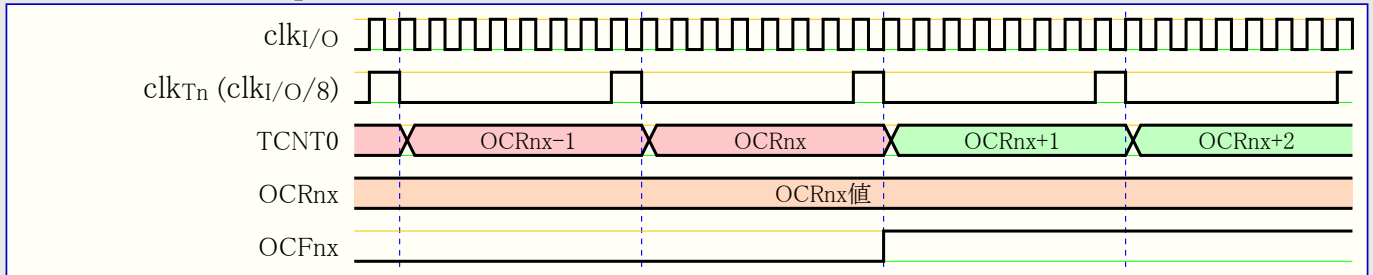
図15-12. 前置分周なし(1/1)のタイマ/カウンタ、 OCF0x 設定 タイミング



注: レジスタとビットの名前での'n'はタイマ/カウンタ番号(タイマ/カウンタ0に対しては $n=0$)、'x'は比較出力部(AまたはB)を表します。

次図は同じタイミング データを示しますが、前置分周器が許可されています。

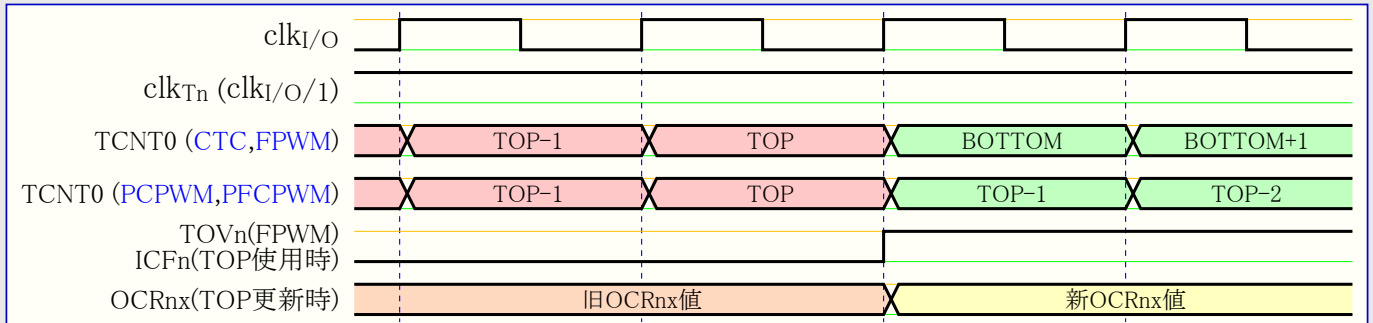
図15-13. 前置分周器($f_{\text{clk}_I/O}/8$)のタイマ/カウンタ、 OCF0x 設定 タイミング



注: レジスタとビットの名前での'n'はタイマ/カウンタ番号(タイマ/カウンタ0に対しては $n=0$)、'x'は比較出力部(AまたはB)を表します。

次図は各動作種別でのTOP近辺の計数手順を示します。位相/周波数基準PWM動作使用時の OCR_{nx} レジスタはBOTTOMで更新されます。タイミング図は同じになりますが、当然TOPはBOTTOMで、TOP-1はBOTTOM+1でどのように置き換えられます。BOTTOMで TOV0 を設定(1)する動作種別についても、同様な名称変更が適用されます。

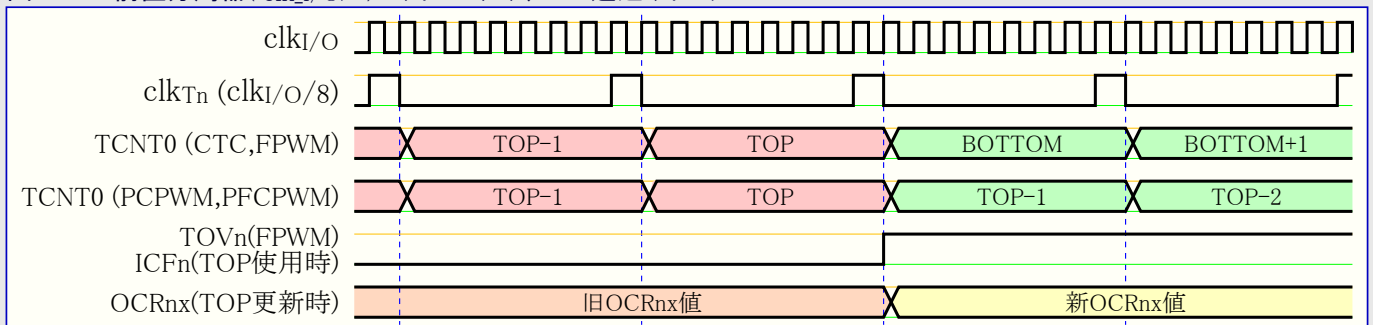
図15-14. 前置分周なし(1/1)のタイマ/カウンタ、TOP近辺 タイミング



注: レジスタとビットの名前での'n'はタイマ/カウンタ番号(タイマ/カウンタ0に対しては $n=0$)、'x'は比較出力部(AまたはB)を表します。

次図は同じタイミング データを示しますが、前置分周器が許可されています。

図15-15. 前置分周器($f_{\text{clk}_I/O}/8$)のタイマ/カウンタ、TOP近辺 タイミング



注: レジスタとビットの名前での'n'はタイマ/カウンタ番号(タイマ/カウンタ0に対しては $n=0$)、'x'は比較出力部(AまたはB)を表します。

15.14. 16ビット タイマ/カウンタ0用レジスタ

15.14.1. TCCR0A – タイマ/カウンタ0制御レジスタA (Timer/Counter 0 Control Register A)

名称 : TCCR0A
変位 : \$2E
リセット : \$00
特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|--------|--------|--------|--------|---|---|-------|-------|
| | COM0A1 | COM0A0 | COM0B1 | COM0B0 | - | - | WGM01 | WGM00 |
| アクセス種別 | R/W | R/W | R/W | R/W | R | R | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- ビット7,6 – COM0A1,0 : 比較1A出力選択 (Compare Output Mode for Channel A)
- ビット5,4 – COM0B1,0 : 比較1B出力選択 (Compare Output Mode for Channel B)

COM0A1,0とCOM0B1,0は各々OC0AとOC0B比較出力ピンの動作を制御します。COM0A1,0ビットの1つまたは両方が1を書かれると、OC0A出力はそのI/Oピンの通常ポート機能を無効にし、そのI/Oピンに接続されます。COM0B1,0ビットの1つまたは両方が1を書かれると、OC0B出力はそのI/Oピンの通常ポート機能を無効にし、そのI/Oピンに接続されます。けれども出力駆動部を許可するため、OC0AまたはOC0Bピンに対応するポート方向レジスタ(DDR)のビットが設定(1)されなければならないことに注意してください。

OC0AまたはOC0Bがピンに接続されるとき、COM0x1,0ビットの機能はWGM03~0ビット設定に依存します。

下表はWGM03~0ビットが標準動作またはCTC動作(つまり非PWM)に設定される時のCOM0x1,0ビット機能を示します。

| 表15-2. 非PWM動作での比較出力選択 (注: xはAまたはB) | | |
|------------------------------------|--------|-----------------------|
| COM0x1 | COM0x0 | 意味 |
| 0 | 0 | 標準ポート動作 (OC0x切断) |
| 0 | 1 | 比較一致でOC0xピン 交互切り替え出力 |
| 1 | 0 | 比較一致でOC0xピン Lowレベル出力 |
| 1 | 1 | 比較一致でOC0xピン Highレベル出力 |

下表はWGM03~0ビットが高速PWM動作に設定される時のCOM0x1,0ビット機能を示します。

| 表15-3. 高速PWM動作での比較出力選択 (注: xはAまたはB, Xは0または1) | | |
|--|--------|---|
| COM0x1 | COM0x0 | 意味 |
| 0 | 0 | 標準ポート動作 (OC0x切断) |
| 0 | 1 | WGM03~0=111X : 比較一致でOC0Aピン交互切り替え出力、OC0Bは標準ポート動作(OC0B切断) WGM03~0上記以外 : 標準ポート動作 (OC0x切断) |
| 1 | 0 | 比較一致でLow、BOTTOMでHighをOC0xピンへ出力 (非反転動作) |
| 1 | 1 | 比較一致でHigh、BOTTOMでLowをOC0xピンへ出力 (反転動作) |

注: COM0x1が設定(1)され、OCR0xがTOPと等しい時に特別な状態が起きます。この状態での比較一致は無視されますが、BOTTOMでの設定(1)または解除(0)は実行されます。詳細については「高速PWM動作」を参照してください。

下表はWGM03~0ビットが位相基準または位相/周波数基準PWM動作に設定される時のCOM0x1,0ビット機能を示します。

| 表15-4. 位相基準または位相/周波数基準PWM動作での比較出力選択 (注: xはAまたはB, Xは0または1) | | |
|---|--------|---|
| COM0x1 | COM0x0 | 意味 |
| 0 | 0 | 標準ポート動作 (OC0x切断) |
| 0 | 1 | WGM03~0=10X1 : 比較一致でOC0Aピン交互切り替え出力、OC0Bは標準ポート動作(OC1B切断) WGM03~0上記以外 : 標準ポート動作 (OC0x切断) |
| 1 | 0 | 上昇計数時の比較一致でLow、下降計数時の比較一致でHighをOC0xピンへ出力 |
| 1 | 1 | 上昇計数時の比較一致でHigh、下降計数時の比較一致でLowをOC0xピンへ出力 |

注: COM0x1が設定(1)され、OCR0xがTOPと等しい時に特別な状態が起きます。詳細については「位相基準PWM動作」を参照してください。

●ビット1,0 – WGM01,0 : 波形生成種別 (Waveform Generation Mode)

タイマ/カウンタ制御レジスタB(TCCR0B)で得られるWGM03,2ビットと組み合わせたこれらのビットはカウンタの計数順序(方向)、最大計数(TOP)値供給元、使われるべき波形生成のどの形式かを制御します。タイマ/カウンタ部によって支援される動作種別は標準動作(カウンタ)、比較一致タイマ/カウンタ解除(CTC)動作と3形式のパルス幅変調(PWM)動作です(「動作種別」をご覧ください)。

表15-5. 波形生成種別選択

| 番号 | WGM03~0 | タイマ/カウンタ動作種別 | TOP値 | OCR0x 更新時 | TOV0 設定時 |
|----|---------|-----------------------|--------|--------------|-------------|
| 0 | 0 0 0 0 | 標準動作 | \$FFFF | 即値 | MAX |
| 1 | 0 0 0 1 | 8ビット位相基準PWM動作 | \$00FF | TOP | BOTTOM |
| 2 | 0 0 1 0 | 9ビット位相基準PWM動作 | \$01FF | TOP | BOTTOM |
| 3 | 0 0 1 1 | 10ビット位相基準PWM動作 | \$03FF | TOP | BOTTOM |
| 4 | 0 1 0 0 | 比較一致タイマ/カウンタ解除(CTC)動作 | OCR0A | 即値 | MAX |
| 5 | 0 1 0 1 | 8ビット高速PWM動作 | \$00FF | TOP | TOP |
| 6 | 0 1 1 0 | 9ビット高速PWM動作 | \$01FF | TOP | TOP |
| 7 | 0 1 1 1 | 10ビット高速PWM動作 | \$03FF | TOP | TOP |
| 8 | 1 0 0 0 | 位相/周波数基準PWM動作 | ICR0 | BOTTOM | BOTTOM |
| 9 | 1 0 0 1 | 位相/周波数基準PWM動作 | OCR0A | BOTTOM | BOTTOM |
| 10 | 1 0 1 0 | 位相基準PWM動作 | ICR0 | TOP | BOTTOM |
| 11 | 1 0 1 1 | 位相基準PWM動作 | OCR0A | TOP | BOTTOM |
| 12 | 1 1 0 0 | 比較一致タイマ/カウンタ解除(CTC)動作 | ICR0 | 即値 | MAX |
| 13 | 1 1 0 1 | (予約) | - | - | - |
| 14 | 1 1 1 0 | 高速PWM動作 | ICR0 | TOP | TOP |
| 15 | 1 1 1 1 | 高速PWM動作 | OCR0A | TOP | TOP |

15.14.2. TCCR0B – タイマ/カウンタ0制御レジスタB (Timer/Counter 0 Control Register B)

名称 : TCCR0B
変位 : \$2D
リセット : \$00
特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|-------|-------|---|-------|-------|--------|-----|-----|
| | ICNC0 | ICES0 | - | WGM03 | WGM02 | CS02~0 | | |
| アクセス種別 | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

●ビット7 – ICNC0 : 捕獲起動入力0雑音消去許可 (Input Capture Noise Canceler)

このビットを(1)に設定することが捕獲起動入力雑音消去器を活性(有効)にします。雑音消去器が有効にされると、捕獲起動入力(ICP0)ピンからの入力が濾波されます。この濾波器機能はそれが出力を更新するのに、連続4回等しく評価されたICP0ピンの採取を必要とします。雑音消去器が許可されると、捕獲入力はそのによって4発振器(システムクロック)周期、遅らされます。

●ビット6 – ICES0 : 捕獲起動入力端選択 (Input Capture Edge Select)

このビットは出来事での捕獲を起動するのに使われる捕獲起動入力(ICP0)ピンのどちらかのエッジを選択します。ICES0ビットが0を書かれると起動動作として下降(負)端が使われ、ICES0ビットが1を書かれると上昇(正)端が捕獲を起動します。

捕獲がICES0設定に従って起動されると、カウンタ値が捕獲レジスタ(ICR0)に複写されます。この出来事は捕獲入力割り込み要求フラグ(ICF0)も設定(1)し、そしてこれは、この割り込みが許可されていれば捕獲入力割り込みを起こすのに使えます。

ICR0がTOP値として使われると(TCCR0AとTCCR0Bに配置されたWGM03~0ビットの記述をご覧ください)、ICP0が切り離され、従って捕獲入力機能は禁止されます。

●ビット4 – WGM03 : 波形生成種別 (Waveform Generation Mode)

タイマ/カウンタ制御レジスタA(TCCR0A)を参照してください。

●ビット3 – WGM02 : 波形生成種別 (Waveform Generation Mode)

タイマ/カウンタ制御レジスタA(TCCR0A)を参照してください。

●ビット2～0 – CS02～0 : クロック選択 (Clock Select)

3つのクロック選択ビットはタイマ/カウンタ(TCNT0)によって使われるべきクロック元を設定します。図15-12と図15-13を参照してください。

| 表15-6. タイマ/カウンタ0入力クロック選択 | | | |
|--------------------------|------|------|-----------------------------------|
| CS02 | CS01 | CS00 | 意味 |
| 0 | 0 | 0 | 停止 (タイマ/カウンタ0動作停止) |
| 0 | 0 | 1 | clk _{I/O} (前置分周なし) |
| 0 | 1 | 0 | clk _{I/O} /8 (8分周) |
| 0 | 1 | 1 | clk _{I/O} /64 (64分周) |
| 1 | 0 | 0 | clk _{I/O} /256 (256分周) |
| 1 | 0 | 1 | clk _{I/O} /1024 (1024分周) |
| 1 | 1 | 0 | T0ピンの下降端 (外部クロック) |
| 1 | 1 | 1 | T0ピンの上昇端 (外部クロック) |

タイマ/カウンタ0に対して外部ピン(クロック)動作が使われる場合、例えT0ピンが出力として設定されても、T0ピンの遷移はカウンタをクロック駆動します。この特性はソフトウェアに計数制御を許します。

15.14.3. TCCR0C – タイマ/カウンタ0制御レジスタC (Timer/Counter 0 Control Register C)

名称 : TCCR0C
変位 : \$2C
リセット : \$00
特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|-------|-------|---|---|---|---|---|---|
| | FOC0A | FOC0B | - | - | - | - | - | - |
| アクセス種別 | W | W | R | R | R | R | R | R |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

●ビット7 – FOC0A : OC0A強制変更 (Force Output Compare 0A)

●ビット6 – FOC0B : OC0B強制変更 (Force Output Compare 0B)

FOC0A/FOC0BビットはWGM03～0ビットが非PWM動作を指示する時だけ有効です。FOC0A/FOC0Bビットに論理1を書くと波形生成部で直ちに比較一致が強制されます。OC0x出力はCOM0x1,0ビット設定に従って変更されます。FOC0A/FOC0Bビットがストローブ(瞬間値)として実行されることに注意してください。それによって強制された比較の効果を決めるのはCOM0x1,0ビットに存在する値です。FOC0A/FOC0Bストローブは何れの割り込みの生成もTOPとしてOCR0Aを使う比較一致タイマ解除(CTC)動作でのタイマ/カウンタの解除(\$0000)も行いません。FOC0A/FOC0Bビットは常に0として読まれます。

15.14.4. TCNT0H,TCNT0L – タイマ/カウンタ0計数値上位/下位バイト (Timer/Counter 0 Counter Value Low and High byte)

TCNT0HとTCNT0Lのレジスタ対は16ビット値のTCNT0を表します。下位バイト[7～0](接尾辞L)は変位原点でアクセスできます。上位バイト[15～8](接尾辞H)は変位+1でアクセスすることができます。16ビットレジスタの読み書きのより多くの詳細については「16ビットレジスタのアクセス」をご覧ください。

名称 : TCNT0HとTCNT0L
変位 : \$28
リセット : \$0000
特質 : -

| ビット | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|--------|-----------|-----|-----|-----|-----|-----|-----|-----|
| | TCNT015~8 | | | | | | | |
| アクセス種別 | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TCNT07~0 | | | | | | | |
| アクセス種別 | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

●ビット15～0 – TCNT015～0 : タイマ/カウンタ0計数値 (Timer/Counter Counter Value)

TCNT0HとTCNT0LはTCNT0に結合されます。

カウンタが走行中にカウンタ(TCNT0)を変更することはOCR0xの1つとTCNT0間の比較一致消失の危険を誘発します。

TCNT0への書き込みは全ての比較部に対して次のタイマ/カウンタクロックでの比較一致を妨害(除去)します。

15.14.5. OCR0AH,OCR0AL – タイマ/カウンタ0比較レジスタ上位/下位バイト (Output Compare Register 0 A Low and High byte)

OCR0AHとOCR0ALのレジスタ対は16ビット値のOCR0Aを表します。下位バイト[7～0](接尾辞L)は変位原点でアクセスできます。上位バイト[15～8](接尾辞H)は変位+1でアクセスすることができます。16ビットレジスタの読み書きのより多くの詳細については「16ビットレジスタのアクセス」をご覧ください。

名称：OCR0AHとOCR0AL
変位：\$26
リセット：\$0000
特質：-

| ビット | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|--------|-----------|-----|-----|-----|-----|-----|-----|-----|
| | OCR0A15~8 | | | | | | | |
| アクセス種別 | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | OCR0A7~0 | | | | | | | |
| アクセス種別 | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

●ビット15～0 – OCR0A15～0：タイマ/カウンタ0比較A値 (Output Compare A Value)

OCR0AHとOCR0ALはOCR0Aに結合されます。

比較レジスタは継続的にカウンタ(TCNT0)値と比較される16ビット値を含みます。一致は比較一致割り込みやOC0Aピンでの波形出力を生成するのに使えます。

15.14.6. OCR0BH,OCR0BL – タイマ/カウンタ0比較Bレジスタ上位/下位バイト (Output Compare Register 0 B Low and High byte)

OCR0BHとOCR0BLのレジスタ対は16ビット値のOCR0Bを表します。下位バイト[7～0](接尾辞L)は変位原点でアクセスできます。上位バイト[15～8](接尾辞H)は変位+1でアクセスすることができます。16ビットレジスタの読み書きのより多くの詳細については「16ビットレジスタのアクセス」をご覧ください。

名称：OCR0BHとOCR0BL
変位：\$24
リセット：\$0000
特質：-

| ビット | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|--------|-----------|-----|-----|-----|-----|-----|-----|-----|
| | OCR0B15~8 | | | | | | | |
| アクセス種別 | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | OCR0B7~0 | | | | | | | |
| アクセス種別 | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

●ビット15～0 – OCR0B15～0：タイマ/カウンタ0比較B値 (Output Compare B Value)

OCR0BHとOCR0BLはOCR0Bに結合されます。

比較レジスタは継続的にカウンタ(TCNT0)値と比較される16ビット値を含みます。一致は比較一致割り込みやOC0Bピンでの波形出力を生成するのに使えます。

15.14.7. ICR0H,ICR0L – タイマ/カウンタ0捕獲レジスタ上位/下位バイト (Input Capture Register 0 Low and High byte)

ICR0HとICR0Lのレジスタ対は16ビット値のICR0を表します。下位バイト[7~0](接尾辞L)は変位原点でアクセスできます。上位バイト[15~8](接尾辞H)は変位+1でアクセスすることができます。16ビットレジスタの読み書きのより多くの詳細については「[16ビットレジスタのアクセス](#)」をご覧ください。

名称 : ICR0HとICR0L

変位 : \$22

リセット : \$0000

特質 : -

| ビット | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|--------|----------|-----|-----|-----|-----|-----|-----|-----|
| | ICR015~8 | | | | | | | |
| アクセス種別 | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ICR07~0 | | | | | | | |
| アクセス種別 | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

● ビット15~0 – ICR015~0 : タイマ/カウンタ0捕獲値 (Input Capture Value)

ICR0HとICR0LはICR0に結合されます。

捕獲レジスタはICP0ピン(またはタイマ/カウンタ0については任意の[アナログ比較器出力](#))で出来事が起こる毎にカウンタ(TCNT0)値で更新されます。この捕獲レジスタはタイマ/カウンタのTOP値を定義するのに使えます。

15.14.8. TIMSK0 – タイマ/カウンタ0割り込み許可レジスタ (Timer/Counter0 Interrupt Mask Register)

名称 : TIMSK0

変位 : \$2B

リセット : \$00

特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|---|---|-------|---|---|--------|--------|-------|
| | - | - | ICIE0 | - | - | OCIE0B | OCIE0A | TOIE0 |
| アクセス種別 | R | R | R/W | R | R | R/W | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

● ビット5 – ICIE0 : タイマ/カウンタ0捕獲割り込み許可 (Timer/Counter0 Input Capture Interrupt Enable)

このビットが1を書かれて、ステータスレジスタ(SREG)の全割り込み許可(I)ビットが設定(1)されると、タイマ/カウンタ0捕獲割り込みが許可されます。タイマ/カウンタ0割り込み要求フラグレジスタ(TIFR0)に配置された捕獲割り込み要求フラグ(ICF0)が設定(1)されると、対応する割り込みベクタが実行されます。

● ビット2 – OCIE0B : タイマ/カウンタ0比較B割り込み許可 (Timer/Counter0 Output Compare B Match Interrupt Enable)

このビットが1を書かれ、ステータスレジスタ(SREG)の全割り込み許可(I)ビットが設定(1)されると、タイマ/カウンタ0比較B一致割り込みが許可されます。タイマ/カウンタ0割り込み要求フラグレジスタ(TIFR0)に配置された比較0B割り込み要求フラグ(OCF0B)が設定(1)されると、対応する割り込みベクタが実行されます。

● ビット1 – OCIE0A : タイマ/カウンタ0比較A割り込み許可 (Timer/Counter0 Output Compare A Match Interrupt Enable)

このビットが1を書かれ、ステータスレジスタ(SREG)の全割り込み許可(I)ビットが設定(1)されると、タイマ/カウンタ0比較A一致割り込みが許可されます。タイマ/カウンタ0割り込み要求フラグレジスタ(TIFR0)に配置された比較0A割り込み要求フラグ(OCF0A)が設定(1)されると、対応する割り込みベクタが実行されます。

● ビット0 – TOIE0 : タイマ/カウンタ0溢れ割り込み許可 (Timer/Counter0 Overflow Interrupt Enable)

このビットが1を書かれて、ステータスレジスタ(SREG)の全割り込み許可(I)ビットが設定(1)されると、タイマ/カウンタ0溢れ割り込みが許可されます。タイマ/カウンタ0割り込み要求フラグレジスタ(TIFR0)に配置されたタイマ/カウンタ0溢れ割り込み要求フラグ(TOV0)が設定(1)されると、対応する割り込みベクタが実行されます。

15.14.9. TIFR0 – タイマ/カウンタ割り込み要求フラグレジスタ (Timer/Counter0 Interrupt Flag Register)

名称 : TIFR0
変位 : \$2A
リセット : \$00
特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|---|---|------|---|---|-------|-------|------|
| | - | - | ICF0 | - | - | OCF0B | OCF0A | TOV0 |
| アクセス種別 | R | R | R/W | R | R | R/W | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

●ビット5 – ICF0 : タイマ/カウンタ0捕獲割り込み要求フラグ (Timer/Counter0, Input Capture Flag)

ICP0ピンに捕獲の事象が起こると、このフラグが設定(1)されます。捕獲レジスタ(ICR0)がWGM03~0によってTOP値として設定されると、ICF0フラグはカウンタがTOP値に到達する時に設定(1)されます。

捕獲割り込みベクタが実行されると、ICF0は自動的に解除(0)されます。代わりにこのビット位置へ論理1を書くことによってもICF0は解除(0)できます。

●ビット2 – OCF0B : タイマ/カウンタ0比較B割り込み要求フラグ (Timer/Counter0, Output Compare B Match Flag)

このフラグはカウンタ(TCNT0)値が比較Bレジスタ(OCR0B)と一致した後(次)のタイマ/カウンタクロック周期で設定(1)されます。

強制的な比較出力(FOC0B)ストローブがOCF0Bフラグを設定(1)しないことに注意してください。

比較B一致割り込みベクタが実行されると、OCF0Bは自動的に解除(0)されます。代わりにこのビット位置へ論理1を書くことによってもOCF0Bは解除(0)できます。

●ビット1 – OCF0A : タイマ/カウンタ0比較A割り込み要求フラグ (Timer/Counter0, Output Compare A Match Flag)

このフラグはカウンタ(TCNT0)値が比較Aレジスタ(OCR0A)と一致した後(次)のタイマ/カウンタクロック周期で設定(1)されます。

強制的な比較出力(FOC0A)ストローブがOCF0Aフラグを設定(1)しないことに注意してください。

比較A一致割り込みベクタが実行されると、OCF0Aは自動的に解除(0)されます。代わりにこのビット位置へ論理1を書くことによってもOCF0Aは解除(0)できます。

●ビット0 – TOV0 : タイマ/カウンタ0溢れ割り込み要求フラグ (Timer/Counter0 Overflow Flag)

このフラグの(1)設定はWGM03~0ビット設定に依存します。標準またはCTC動作でのTOV0フラグはタイマ/カウンタ0溢れ時に設定(1)されます。他のWGM03~0ビット設定を使う時のTOV0フラグ動作については表15-5を参照してください。

タイマ/カウンタ0溢れ割り込みベクタが実行されると、TOV0は自動的に解除(0)されます。代わりにこのビット位置へ論理1を書くことによってもTOV0は解除(0)できます。

15.14.10. GTCCR – 一般タイマ/カウンタ制御レジスタ (General Timer/Counter Control Register)

名称 : GTCCR
変位 : \$2F
リセット : \$00
特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|-----|---|---|---|---|---|-------|-----|
| | TSM | - | - | - | - | - | REMAP | PSR |
| アクセス種別 | R/W | R | R | R | R | R | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

●ビット7 – TSM : タイマ/カウンタ同時動作 (Timer/Counter Synchronization Mode)

TSMビットへの1書き込みはタイマ/カウンタ同期化動作を活性(有効)にします。この動作でPSRへ書かれる値は保持され、従って対応する前置分周器リセット信号の有効を保持します。これはタイマ/カウンタが停止され、設定中に進行する危険なしに設定できることを保証します。TSMビットが0を書かれると、PSRビットはハードウェアによって解除(0)され、タイマ/カウンタが計数を始めます。

●ビット1 – REMAP : ピン再割り当て (Remap Pin)

このビットは表で示されるように、タイマ/カウンタ0ピンがピンにどう割り当てられるかを制御します。

表15-7. タイマ/カウンタ0ピン割り当て

| REMAP | T0 | OC0A | OC0B | ICP0 | 注 |
|-------|-----|-------|-------|-------|-----------|
| 0 | PA0 | PB1 | PA1 | PB2 | 既定 |
| 1 | PB3 | (PA3) | (PA5) | (PA4) | 再割り当て(代替) |

● ビット0 – PSR10 : タイマ/カウンタ前置分周器リセット (Prescaler Reset Timer/Counter)

このビットが1の時にタイマ/カウンタ0の前置分周器がリセットします。通常、TSMビットが設定(1)されている場合を除き、このビットはハードウェアによって直ちに解除(0)されます。

16. USART (Universal Synchronous Asynchronous Receiver Transceiver)

16.1. 概要

USART(Universal Synchronous and Asynchronous Receiver and Transmitter)は高い柔軟性をもつ直列通信機能です。

USARTは主装置SPI動作での使用もできます。電力削減レジスタ(PRR)のUSART電力削減(PRUSART0)ビットはUSARTnを許可するために0を書かれなければなりません。

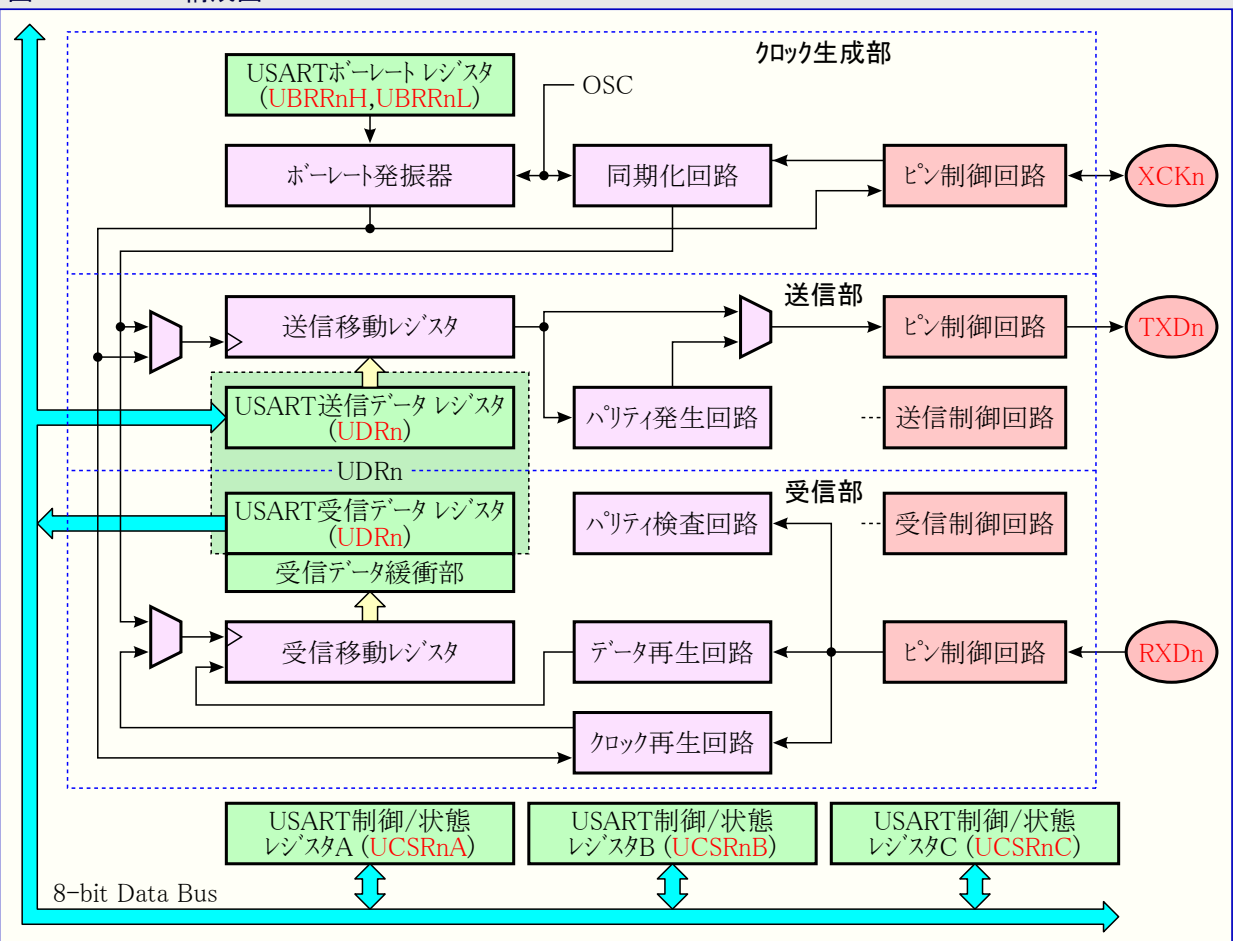
16.2. 特徴

- 全二重動作 (独立した送受信レジスタ)
- 同期または非同期動作
- 同期クロック駆動された主装置/従装置動作
- 高分解能ボーレート発振器
- 5, 6, 7, 8または9ビットデータと1または2停止ビットの直列フレームの支援
- ハードウェアによって支援された奇数または偶数パリティの生成と検査
- データ オーバーラン検出
- フレーミング異常検出
- フレーミング異常検出
- 不正開始ビット検出とデジタル低域通過濾波器を含む雑音濾波器
- 受信完了、送信完了、送信データレジスタ空きの3つの分離した割り込み
- 複数プロセッサ通信機能
- 倍速非同期通信動作

16.3. 構成図

USART構成図に於いてCPUがアクセス可能なレジスタとI/Oピンは赤文字(訳注:原文は太字)で示されます。構成図内の破線はUSARTの3つの主要部分、(上から)クロック生成部、送信部、受信部を分けます。制御レジスタは全部によって共用されます。クロック生成論理部はボーレート発振器と同期従装置動作によって使われる外部クロック入力に対する同期化論理回路から成ります。転送クロック(XCKn)ピンは同期転送動作だけで使われます。送信部は単一書き込み緩衝部(UDRn)、直列移動レジスタ、パリティ発生器、異なる直列フレーム形式を扱うための制御論理回路から成ります。書き込み緩衝部はどんなフレーム間の遅れもなしにデータの継続転送を許します。受信部はクロックとデータの再生部のため、USART部の最も複雑な部分です。再生部は非同期データ受信で使われます。再生部に加えて、受信部は2重の受信緩衝部(UDRn)、移動レジスタ、パリティ検査器、制御論理回路を含みます。受信部は送信部と同じフレーム形式を支援し、フレーミング異常、データ オーバーラン発生、パリティ誤りを検知できます。

図16-1. USART構成図



注: USARTピン配置については「ピン配置」記述と「入出力ポート」記述を参照してください。

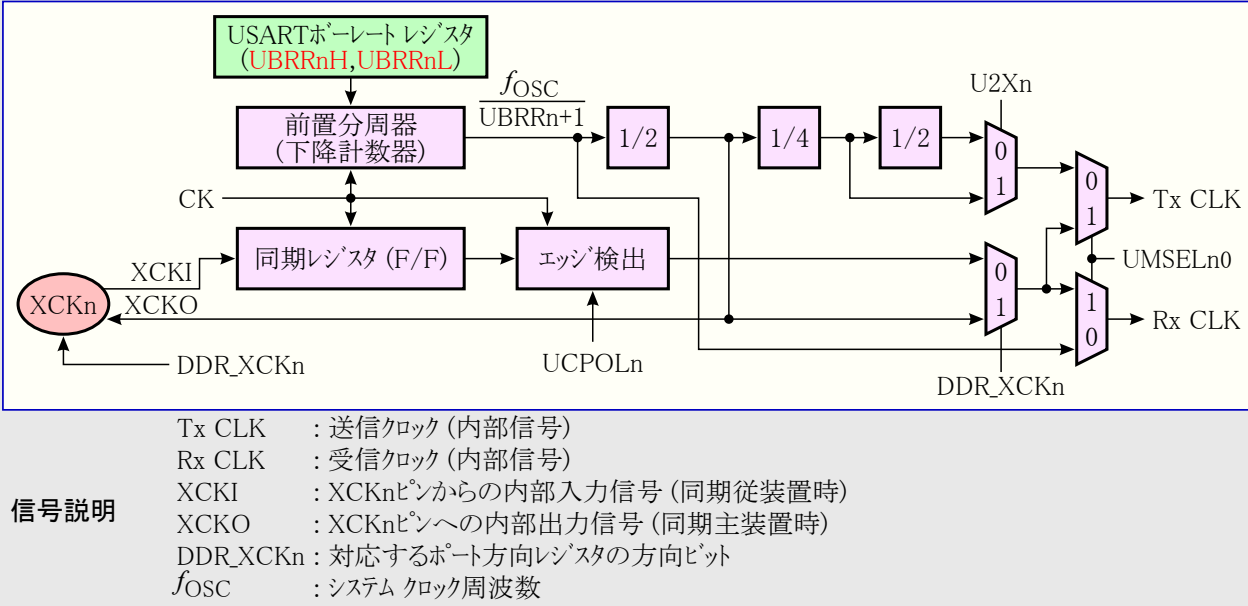
関連リンク [7頁の「ピン配置」](#)
[37頁の「入出力ポート」](#)

16.4. クロック生成

クロック生成論理回路は送受信部用基準クロックを生成します。USARTは標準非同期、倍速非同期、同期主装置、同期従装置の4つのクロック動作種別を支援します。USART制御/状態レジスタC(UCSRnC)のUSART動作種別選択(UMSELn0)ビットは同期動作と非同期動作のどちらかを選びます。倍速動作(非同期動作のみ)はUSART制御/状態レジスタA(UCSRnA)にある倍速許可(U2Xn)ビットによって制御されます。同期動作(UMSELn0=1)を使うとき、XCKnピンに対する方向制御ビット(DDR_XCKn)はクロック元が内部(主装置動作)または外部(従装置動作)のどちらかを制御します。このXCKnピンは同期動作を使う時だけ活性(有効)です。

下はクロック生成論理回路の構成図を示します。

図16-2. クロック生成部構成図



16.4.1. ボーレート発振器での内部クロック発生

内部クロック生成は非同期と同期主装置動作種別に対して使われます。本項の記述は前項のクロック生成部構成図を参照してください。

USARTボーレートレジスタ(UBRRn(UBRRnH:UBRRnL))と下降計数器は設定可能な前置分周器またはボーレート発振器として機能するように接続されます。システムクロック(f_{OSC})で走行する下降計数器は0への下降計数時毎またはUBRRnLレジスタが書かれる時にUBRRn値で設定されます。1クロックはカウンタが0に達する毎に生成されます。このクロックがボーレート発振器出力($=f_{OSC}/(UBRRn+1)$)です。送信部は動作種別に依存してボーレート発振器出力を2,8,16分周します。ボーレート発振器出力は受信部クロックとデータ再生部によって直接使われます。しかし、再生部はUSART動作種別選択(UMSELn0)、倍速許可(U2Xn)、DDR_XCKnビットの状態によって設定される動作種別に依存して2,8,16段を使う順次処理回路を使います。

下表は内部的に生成したクロック元を使う各動作種別に於けるボーレート(bps)とUBRRn値の計算式を含みます。

| 表16-1. ボーレートレジスタ(UBRRn)値計算式 | | |
|-----------------------------|--|--|
| 動作種別 | ボーレート計算式 | UBRRn値計算式 |
| 標準速非同期動作 (U2Xn=0) | $BAUD = \frac{f_{OSC}}{16 \times (UBRRn+1)}$ | $UBRRn = \frac{f_{OSC}}{16 \times BAUD} - 1$ |
| 倍速非同期動作 (U2Xn=1) | $BAUD = \frac{f_{OSC}}{8 \times (UBRRn+1)}$ | $UBRRn = \frac{f_{OSC}}{8 \times BAUD} - 1$ |
| 同期主装置動作 | $BAUD = \frac{f_{OSC}}{2 \times (UBRRn+1)}$ | $UBRRn = \frac{f_{OSC}}{2 \times BAUD} - 1$ |

注: ボーレートは転送速度(ビット/1秒)で定義されます。

BAUD : ボーレート (bps)
UBRRn : UBRRnHとUBRRnLレジスタ値 (0~4095)
 f_{OSC} : システム発振器クロック周波数

いくつかのシステムクロック周波数に対するいくつかのUBRRn値の例は「ボーレート設定例」で得られます。

16.4.2. 倍速動作 (U2Xn)

転送速度はUSART制御/状態レジスタA(UCSRnA)で倍速許可(U2Xn)ビットを設定(1)することによって倍にできます。このビットの設定は非同期動作に対してだけ有効です。同期動作を使うとき、このビットは0に設定してください。

このビットの設定(1)は事実上非同期通信に対する転送速度を倍にするボーレート分周器の分周数を16から8に減らします。けれども、この場合、受信部がデータ採取とクロック再生に対して半分の(16から8に減じた)採取数しか使わず、従ってこの動作種別が使われる時にシステムクロックとボーレート設定の精度がより必要とされます。

送信部についての低下要因はありません。

16.4.3. 外部クロック

外部クロックは同期従装置動作種別によって使われます。本項の記述は前項のクロック生成部構成図を参照します。

XCKnピンからの外部クロック入力是不確定レベル状態(メタステーブル)の機会を最少とするために同期化レジスタによって採取されます。同期化レジスタからの出力は送受信部で使い得るのに先立ってエッジ検出器を通過しなければなりません。この処理手順が2CPUクロック周期の遅延を持ち込み、このため最大外部XCKnクロック周波数は次式によって制限されます。

$$f_{XCKn} < \frac{f_{OSC}}{4}$$

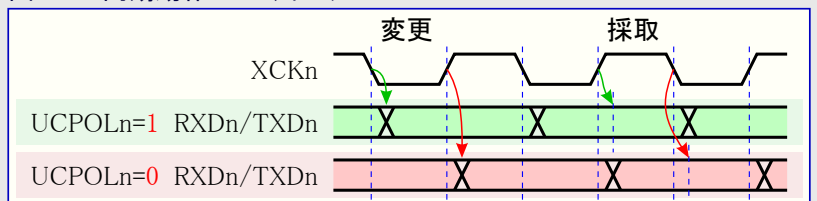
f_{OSC} の値がシステムクロック元の安定度に依存します。従って周波数変動によるデータ消失の可能性を避けるため、いくらかの余裕分を追加することが推奨されます。

16.4.4. 同期クロック動作

同期動作が使われる(UMSELn=1)とき、XCKnピンはクロック入力(従装置)またはクロック出力(主装置)のどちらかとして使われます。データ採取またはデータ変更とクロック端間の依存性は同じです。基本原則はデータ出力(TXDn)が変更されるエッジと反対のXCKnクロック端でデータ入力(RXDn)が採取されることです。

USART制御/状態レジスタC(UCSRnC)のXCKn極性(UCPOLn)ビットはデータ採取とデータ変更に対してどちらのクロック端が使われるのかを選びます。右のタイミング構成図で示されるようにUCPOLnが0の時にデータはXCKnの上昇端で変更され、下降端で採取されます。UCPOLnが設定(1)の場合、データはXCKnの下降端で変更され、上昇端で採取されます。

図16-3. 同期動作XCKnタイミング



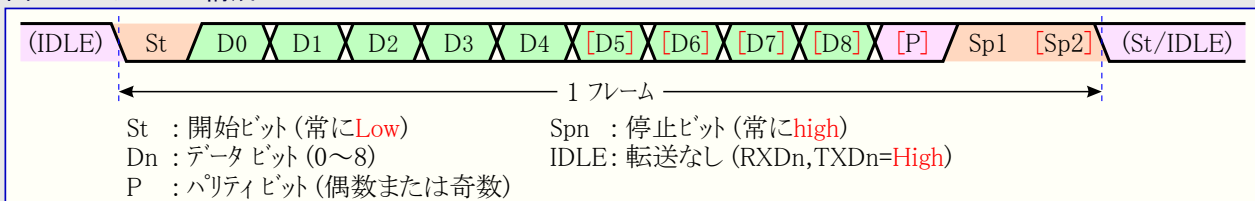
16.5. フレーム形式

1つの直列フレームは複数のデータビットと同期ビット(開始ビット、停止ビット)、任意の異常検査用パリティビットで定義されます。USARTは有効なフレーム形式として以下の組み合わせ30種全てを受け入れます。

- ・ 1 開始ビット
- ・ 5, 6, 7, 8, 9 ビット データ
- ・ 奇数または偶数パリティビット、またはなし
- ・ 1 または 2 停止ビット

フレームは最初に最下位データビット(LSB)、その後に最上位データビット(MSB)で終わる次データビットの、(合計で5～9ビットの)データビットが後続する開始ビットで始まります。許可したなら、パリティビットがデータビットの後、1つまたは2つの停止ビットの前に挿入されます。完全なフレームが送信されると、新規フレームによって直ちに後続されるか、または通信線をアイドル状態(high)に設定できます。下図は組み合わせ可能なフレーム形式を図解します。[] 付きビットは任意選択です。

図16-4. 17フレームの構成



St : 開始ビット (常にLow)
Dn : データビット (0～8)
P : パリティビット (偶数または奇数)
Spn : 停止ビット (常にhigh)
IDLE: 転送なし (RXDn, TXDn=High)

USARTによって使われるフレーム形式は以下によって設定されます。

- ・ フレーム内のデータビット数を選択するUSART制御/状態レジスタBとC(UCSRnB,UCSRnC)のデータ長選択(UCSZn2,UCSZn1,0)ビット
- ・ パリティを許可とパリティビットの種別(奇/偶)を設定するパリティ選択(UPMn1,0)ビット
- ・ 停止ビット数(1または2)を選択する停止ビット選択(USBSn)ビット (受信部は第2停止ビットを無視します。)

受信部と送信部は同じ設定を使います。これらのどのビットの設定変更も、送受信部両方に対して進行中の通信を不正とすることに注意してください。フレーミング異常(FEn)は最初の停止ビットが0(Low)の場合にだけ検出されます。

16.5.1. パリティビットの計算

パリティビットは全データビットの排他的論理和(Ex-OR)を行うことによって計算されます。奇数パリティが使われる場合は排他的論理和の結果が反転されます。パリティビットとデータビットの関係は次のとおりです。

偶数パリティビット = D0 Ex-OR D1 Ex-OR D2 Ex-OR ~ Ex-OR Dn-1
奇数パリティビット = D0 Ex-OR D1 Ex-OR D2 Ex-OR ~ Ex-OR Dn-1 Ex-OR 1 n : データビット長

使った場合、パリティビットは直列フレームの最後のデータビットと最初の停止ビット間に配置されます。

16.6. USARTの初期化

何れかの通信が行えるのに先立ってUSARTは初期化されなければなりません。標準的な初期化手順は使用方法に依存するボーレート設定、フレーム形式設定、送受信部許可から成ります。割り込み駆動USART操作に関して初期化を行うとき、[ステータスレジスタの全割り込み許可\(1\)ビット](#)は解除(0)される(そして全割り込みが禁止される)べきです。

ボーレートまたはフレーム形式の変更を伴う再初期化を行う前には、レジスタが変更される期間中に進行中の送信がないことを確実にしてください。[USART制御/状態レジスタA\(UCSRnA\)](#)の[送信完了\(TXCn\)フラグ](#)は送信部の全転送完了検査に使え、[受信完了\(RXCn\)フラグ](#)は受信緩衝部内の未読データ有無検査に使えます。この目的(次送信タイミング)にTXCnフラグが使われる場合、各々の送信([USARTデータレジスタ\(UDRn\)](#)が書かれる)前にTXCnフラグが解除(0)されなければなりません。

次の簡単なUSART初期化コード例が示すアセンブリ言語とC言語の関数は機能的に同じです。この例は固定フレーム形式でボーリングを使う(割り込み不許可)非同期動作と仮定します。ボーレート(UBRRn)値は関数の引数として与えられます。アセンブリ言語でのボーレート引数はR17:R16レジスタに格納されると仮定されます。

アセンブリ言語プログラム例

```
USART_Init:  OUT    UBRRnH, R17                ;ボーレート設定(上位バイト)
             OUT    UBRRnL, R16                ;ボーレート設定(下位バイト)
             LDI    R16, (1<<USBSn) | (3<<UCSZn0) ;フレーム形式値を取得
             OUT    UCSRnC, R16                ;フレーム形式設定(8ビット,2停止ビット)
             LDI    R16, (1<<RXENn) | (1<<TXENn) ;送受信許可値を取得
             OUT    UCSRnB, R16                ;送受信許可
             RET                                ;呼び出し元へ復帰
```

C言語プログラム例

```
#define FOSC 1843200                /* MCUクロック周波数 */
#define BAUD 9600                   /* 目的USARTボーレート速度 */
#define MYUBRR FOSC/16/BAUD-1      /* 目的UBRRn値 */

void main(void)
{
    ~
    USART_Init(MYUBRR);             /* USART初期化 */
    ~
}

void USART_Init(unsigned int baud)
{
    UBRRnH = (unsigned char) (baud>>8); /* ボーレート設定(上位バイト) */
    UBRRnL = (unsigned char) baud;       /* ボーレート設定(下位バイト) */
    UCSRnC = (1<<USBSn) | (3<<UCSZn0);  /* フレーム形式設定(8ビット,2停止ビット) */
    UCSRnB = (1<<RXENn) | (1<<TXENn);    /* 送受信許可 */
}
```

割り込みの禁止や引数としてフレーム形式を含めるなどで、より進化した初期化ルーチンが作成できます。けれども多くの応用はボーレートや制御レジスタの固定した設定が使用され、これらの応用形式での初期化コードは主ルーチンに直接置けるか、または他のI/Oの初期化コードと併せられます。

16.7. データ送信 – USART送信部

USART送信部はUSART制御/状態レジスタB(UCSRnB)で送信許可(TXENn)ビットを設定(1)することによって許可されます。送信部が許可されると、TXDnピンの標準ピン動作はUSARTによって無視され、送信部の直列出力としての機能を与えられます。何かの送信を行う前に一度はボーレート、フレーム形式、動作種別が設定されなければなりません。同期動作が使われる場合、XCKnピンの(受信)クロックは無視され、送信クロックとして使われます。

16.7.1. 5～8ビット データ フレーム送信

データ送信は送信されるべきデータを送信緩衝部に設定することによって開始されます。CPUはUSARTデータレジスタ(UDRn)I/O位置へ書くことによって送信緩衝部に設定できます。送信緩衝部内のデータは移動レジスタが新規フレームを送る準備が整った時に移動レジスタへ移されます。移動レジスタはアイドル状態(送信進行中以外)、または直前のフレームの最後の停止ビット送信後、直ちに新規データが設定されます。移動レジスタが新規データを設定されると、ボーレートレジスタ(UBRRnH:UBRRnL)と倍速許可(U2Xn)ビット、また動作種別によってはXCKnピンによって与えられる速度で1つの完全なフレームを転送します。8ビット未満のフレームを使うとき、UDRnに書かれた上位ビットは無視されます。

次のコード例はUSART制御/状態レジスタA(UCSRnA)の送信データレジスタ空き(UDREN)フラグのポーリングを基準とした簡単なUSART送信関数を示します。この関数が使われ得る前にUSARTが初期化されなければなりません。アセンブリ言語での送るべきデータはR16レジスタに格納されると仮定されます。

アセンブリ言語プログラム例

```
USART_Tx:    SBIS    UCSRnA, UDREN    ;送信緩衝部空きでスキップ
             RJMP    USART_Tx        ;送信緩衝部空き待機
;
             OUT     UDRn, R16        ;データ送信(送信開始)
             RET                     ;呼び出し元へ復帰
```

C言語プログラム例

```
void USART_Transmit(unsigned char data)
{
    while ( !(UCSRnA & (1<<UDREN)) );    /* 送信緩衝部空き待機 */
    UDRn = data;                          /* データ送信(送信開始) */
}
```

この関数は送信されるべき新規データを設定する前に、UDRENの検査によって送信緩衝部が空になるのを単純に待ちます。送信緩衝部空き割り込みが使われる場合、その割り込み処理ルーチンがデータを緩衝部内に書きます。

16.7.2. 9ビット データ フレーム送信

9ビット データが使われる場合(UCSZn2～0=111)、データの下位8ビットがUSARTデータレジスタ(UDRn)に書かれるのに先立って第9ビットがUSART制御/状態レジスタB(UCSRnB)の送信データビット8(TXB8n)ビットに書かれなければなりません。

第9ビットは複数プロセッサ通信使用時のアドレス フレーム識別、また例えば同期として扱う他の規約で使うことができます。

次のコード例は9ビット データを扱う送信関数を示します。アセンブリ言語での送るべきデータはR17:R16レジスタに格納されると仮定されます。

アセンブリ言語プログラム例

```
USART_Tx:    SBIS    UCSRnA, UDREN    ;送信緩衝部空きでスキップ
             RJMP    USART_Tx        ;送信緩衝部空き待機
;
             CBI     UCSRnB, TXB8n    ;第9ビットを0に仮設定
             SBRC    R17, 0          ;送信すべき第9ビットが0でスキップ
             SBI     UCSRnB, TXB8n    ;第9ビットを1に設定
             OUT     UDRn, R16        ;データ送信(送信開始)
             RET                     ;呼び出し元へ復帰
```

C言語プログラム例

```
void USART_Transmit(unsigned int data)
{
    while ( !(UCSRnA & (1<<UDREN)) );    /* 送信緩衝部空き待機 */
    UCSRnB &= ~(1<<TXB8n);              /* TXB8nを0に仮設定 */
    if (data & 0x0100) UCSRnB |= (1<<TXB8n); /* 第9ビットをR17からTXB8nへ複写 */
    UDRn = data;                          /* データ送信(送信開始) */
}
```

注: これらの送信関数は一般的な機能で書かれています。UCSRBnの内容が静的(換言すると、UCSRnBのTXB8nビットが初期化後に使われるだけ)ならば最適化できます。

16.7.3. 送信フラグと割り込み

USART送信部には状態を示す2つのフラグ、USARTデータレジスタ空き(UDREN)と送信完了(TXCn)があります。両フラグは割り込みを発生するのに使えます。

USARTデータレジスタ空き(UDREN)フラグは送信緩衝部が新規データを受け取る準備ができていないかどうかを示します。このビットは送信緩衝部が空の時に設定(1)され、送信緩衝部が移動レジスタに未だ移動されてしまっていない送信されるべきデータを含む時に解除(0)されます。将来のデバイスとの共通性のため、USART制御/状態レジスタA(UCSRnA)に書くとき、常にこのビットに0を書いてください。

USART制御/状態レジスタB(UCSRnB)でデータレジスタ空き割り込み許可(UDRIEn)ビットが1を書かれると、(全割り込みが許可されていれば)UDRENフラグが設定(1)されている限り、USARTデータレジスタ空き割り込みが実行されます。UDRENはUSARTデータレジスタ(UDRn)書き込みによって解除(0)されます。割り込み駆動データ送信が使われるとき、データレジスタ空き割り込みルーチンはUDRENを解除(0)するために新規データをUDRnに書くか、データレジスタ空き割り込みを禁止するかのどちらかを行わなければならない、さもなければ一旦割り込みルーチンを終了しても新しい割り込みが(継続的に)起こります。

送信完了(TXCn)フラグは送信移動レジスタ内の完全なフレームが移動出力されてしまい、送信緩衝部に新規データが現在存在しない時に設定(1)されます。TXCnフラグは送信完了割り込みが実行される時に自動的に解除(0)されるか、またはこのビット位置に1を書くことによって解除(0)できます。TXCnフラグは送信応用プログラムが送信完了後、直ちに通信回線を開放し、受信動作へ移行しなければならない(RS485規格のような)半二重(ハーフデュプレックス)通信インターフェースで有効です。

UCSRnBで送信完了割り込み許可(TXCIEEn)ビットが1を書かれ、(全割り込みが許可されていれば)TXCnフラグが設定(1)になる時にUSART送信完了割り込みが実行されます。送信完了割り込みが使われるとき、割り込み処理ルーチンはTXCnフラグを解除(0)しなくてもよく、これは割り込みが実行されると、自動的に行われます。

16.7.4. パリティ発生器

パリティ発生器は直列フレームデータに対するパリティビットを計算します。パリティビットが許可されると(UPMn1=1)、送信部制御論理回路は送られているフレームの最終データビットと最初の停止ビット間にパリティビットを挿入します。

16.7.5. 送信の禁止

USART制御/状態レジスタB(UCSRnB)のUSART送信許可(TXENn)ビットへの0書き込み時、送信部の禁止は進行中と保留中の送信が完了される(換言すると、送信移動レジスタと送信緩衝部レジスタが送信されるべきデータを含まない時)まで有効になりません。禁止されると、送信部はもはやTXDnピン(の標準ピン機能)を無効にしません。

16.8. データ受信 - USART受信部

USART受信部はUSART制御/状態レジスタB(UCSRnB)で受信許可(RXENn)ビットに1を書くことによって許可されます。受信部が許可されると、RXDnピンの標準ピン動作はUSARTによって無視され、受信部の直列入力としての機能を与えられます。何か直列受信が行われ得る前に一度はボーレート、フレーム形式、動作種別が設定されなければなりません。同期動作が使われる場合、XCKnピンのクロックは転送クロックとして使われます。

16.8.1. 5～8ビットデータフレーム受信

受信部は有効な開始ビットを検出する時にデータ受信を開始します。開始ビットに続く各ビットはボーレートまたはXCKnクロックで採取され、フレームの最初の停止ビットが受信されるまで受信移動レジスタ内へ移動されます。第2停止ビットは受信部によって無視されます。最初の停止ビットが受信されると(換言すると、受信移動レジスタに完全なフレームが存在すると)、この移動レジスタの内容は受信緩衝部内へ移されます。受信緩衝部はUSARTデータレジスタ(UDRn)I/O位置を読むことによって読めます。8ビット未満のフレームを使うとき、UDRnから読むデータの上位ビットは0で覆われます。

次のコード例はUSART制御/状態レジスタA(UCSRnA)の受信完了(RXCn)フラグのポーリングを基準とした簡単なUSART受信関数を示します。この関数が使われ得る前にUSARTが初期化されなければなりません。

| アセンブリ言語プログラム例 | | | |
|-----------------------------------|--------|----------------------------|---------------|
| USART_Rx: | SBIS | UCSRnA, RXCn | ;受信完了でスキップ |
| | RJMP | USART_Rx | ;受信完了待機 |
| ; | | | |
| | IN | R16, UDRn | ;受信データ取得 |
| | RET | | ;呼び出し元へ復帰 |
| C言語プログラム例 | | | |
| unsigned char USART_Receive(void) | | | |
| { | | | |
| | while | (!(UCSRnA & (1<<RXCn))); | /* 受信完了待機 */ |
| | return | UDRn; | /* 受信データ取得 */ |
| } | | | |

この関数は緩衝部を読んで値を戻す前に、RXCnフラグの検査によって受信緩衝部にデータが存在するのを単純に待ちます。

16.8.2. 9ビット データ フレーム受信

9ビット データが使用される場合(UCSZn2~0=111)、USARTデータレジスタ(UDRn)から下位バイトを読むのに先立って第9ビットがUSART制御/状態レジスタB(UCSRnB)の受信データビット8(RXB8n)ビットから読まれなければなりません。この規則はフレーミング異常(FEn)、オーバラン発生(DORn)、パリティ誤り(UPEn)状態フラグにも適用されます。USART制御/状態レジスタA(UCSRnA)から状態情報を読み、その後にUDRnからデータを読んでください。UDRn I/O位置を読むことが受信FIFO緩衝部の状態を切り替え、その結果、FIFO内に保管されるRXB8n,FEn,DORn,UPEnビット全てが切り替わります。

次のコード例は9ビット データと状態ビット両方を扱う簡単なUSART受信関数を示します。

アセンブリ言語プログラム例

```
USART_Rx:    SBIS    UCSRnA, RXCn        ;受信完了でスキップ
              RJMP    USART_Rx          ;受信完了待機
;
              IN      R18, UCSRnA        ;状態フラグ取得
              IN      R17, UCSRnB        ;受信第9ビット取得
              IN      R16, UDRn          ;受信データ取得
              ANDI     R18, (1<<FEn) | (1<<DORn) | (1<<UPEn) ;受信異常検査
              BREQ     USART_Rx_V        ;異常なしで分岐
;
              LDI     R17, -1            ;異常で-1値設定
              LDI     R16, -1            ;
USART_Rx_V:   LSR     R17                ;RXB8nビットをビット0位置へ移動
              ANDI     R17, $01          ;RXB8nビットのみ有効
              RET                        ;呼び出し元へ復帰
```

C言語プログラム例

```
unsigned int USART_Receive(void)
{
    unsigned char status, resh, resl;    /* 一時変数定義 */
    while ( !(UCSRnA & (1<<RXCn)) );    /* 受信完了待機 */
    status = UCSRnA;                     /* 状態フラグ取得 */
    resh = UCSRnB;                       /* 受信第9ビット取得 */
    resl = UDRn;                         /* 受信データ取得 */
    if ( status & ((1<<FEn) | (1<<DORn) | (1<<UPEn)) ) return -1; /* 受信異常で-1値設定/復帰 */
    resh = (resh>>1) & 0x01;             /* RXB8nビットのみ有効最下位へ */
    return ((resh<<8) | resl);           /* 結果9ビット データ取得/復帰 */
}
```

この受信関数例は何か評価を行う前に、全てのI/Oレジスタをレジスタ ファイルに読みます。これは読まれた緩衝部位置が可能な限り早く新規データを自由に受け入れられるため、最適な受信緩衝部利用になります。

16.8.3. 受信完了フラグと割り込み

USART受信部には受信部の状態を示す1つのフラグがあります。

USART受信完了(RXCn)フラグは未読データが受信緩衝部に存在するかを示します。このフラグは受信緩衝部に未読データが存在する時に1で、受信緩衝部が空の(換言すると、何も未読データを含まない)時に0です。受信部が禁止される場合(RXENn=0)、受信緩衝部が破棄され、その結果としてRXCnフラグは0になります。

USART制御/状態レジスタB(UCSRnB)でUSART受信完了割り込み許可(RXCIEn)ビットが設定(1)されると、(全割り込みが許可されていれば)RXCnフラグが設定(1)されている限り、USART受信完了割り込みが実行されます。割り込み駆動データ受信が使用されるとき、受信完了割り込みルーチンはRXCnフラグを解除(0)するためにUSARTデータレジスタ(UDRn)から受信したデータを読まなければならない、さもないと一旦割り込みルーチンを終了しても新しい割り込みが(継続的に)起きます。

16.8.4. 受信異常フラグ

USART受信にはフレーミング異常(FEn)、データ オーバーラン発生(DORn)、パリティ誤り(UPEn)の3つの異常フラグがあります。これら全ては**USART制御/状態レジスタA(UCSRnA)**を読むことでアクセスできます。異常フラグに対する共通点は異常状態を示しているフレームと共に受信緩衝部に配置されることです。異常フラグが緩衝されるため、**USARTデータレジスタ(UDRn)**I/O位置を読むことが緩衝部読み出し位置を切り替えるので、UCSRnAは受信緩衝部(UDRn)の前に読まれなければなりません。異常フラグに対するその他の共通点はソフトウェアがこのフラグ位置へ書き込みを行うことによって変更できないことです。しかし、将来のUSART(機能)実装の上位互換性のため、UCSRnAが書かれるとき、全てのフラグは**0**に設定されなければなりません。異常フラグはどれも割り込みを生成できません。

フレーミング異常(FEn)フラグは受信緩衝部に格納された、次に読み込み可能なフレームの第1停止ビットの状態を示します。FEnフラグは停止ビットが正しく(**High**として)読まれた時に**0**で、停止ビットが不正(**Low**)だった時にFEnフラグは**1**です。このビットは同期外れ状態の検出、中断状態検出、規約での操作に使えます。受信部が最初(第1)以外の全停止ビットを無視するため、FEnフラグは**USART制御/状態レジスタC(UCSRnC)**の**停止ビット選択(USBSn)ビット**設定によって影響を及ぼされません。将来のデバイスとの共通性のため、UCSRnAに書くとき、常にこのビットを**0**に設定してください。

データ オーバーラン発生(DORn)フラグは受信緩衝部が一杯状態のためのデータ消失を示します。データ オーバーランは受信緩衝部(2フレーム)が一杯で、(次の)新規フレーム データが受信移動レジスタで待っており、(更に次の)新規開始ビットが検出される時に起きます。DORnフラグが設定(**1**)なら、最後にUDRnから読んだフレームと次にUDRnから読むフレーム間で1つ以上の直列フレームが失われます。将来のデバイスとの共通性のため、UCSRnAに書く時は常にこのビットに**0**を書いてください。DORnフラグは受信されたデータが移動レジスタから受信緩衝部へ正常に移動された時に解除(**0**)されます。

パリティ誤り(UPEn)フラグは受信緩衝部内の次のフレームで受信時にパリティ異常があったことを示します。パリティ検査が許可されていない場合、UPEnフラグは常に**0**が読めます。将来のデバイスとの共通性のため、UCSRnAに書くとき、常にこのビットを**0**に設定してください。より多くの詳細については「**パリティビットの計算**」と下の「**パリティ検査器**」をご覧ください。

16.8.5. パリティ検査器

パリティ検査器は**USART制御/状態レジスタC(UCSRnC)**の**パリティ種別上位ビット(UPMn1)**が**1**を書かれる時に活性(有効)です。実行されるべきパリティ検査の形式(偶数または奇数)は**UCSRnC.UPMn0ビット**によって選択されます。許可されると、パリティ検査器は到着フレーム内のデータビットのパリティを計算し、その結果と(受信)直列フレーム内のパリティビットを比較します。検査の結果は受信データ、停止ビットと共に受信緩衝部に格納されます。その後**USART制御/状態レジスタA(UCSRnA)**の**パリティ誤り(UPEn)フラグ**はフレームにパリティ異常があるかを検査するため、ソフトウェアによって読むことができます。

UPEnフラグは受信緩衝部から読まれ得る次フレームで受信時にパリティ異常があり、その時点でパリティ検査が許可されていた(UPMn1=**1**)場合に設定(**1**)されます。このビットはUSARTデータレジスタ(UDRn)が読まれるまで有効です。

16.8.6. 受信の禁止

送信部と対照的に受信部の禁止は即時です。従って受信進行中のデータは失われます。禁止される(即ち、**USART制御/状態レジスタB(UCSRnB)**の**USART受信許可(RXENn)ビット**が**0**を書かれる)と、受信部はもはやRXDnポートピンの標準機能を無効にしません。受信FIFO緩衝部は受信部が禁止されると破棄されます。緩衝部内の残データは失われます。

16.8.7. 受信緩衝部の破棄

受信FIFO緩衝部は受信が禁止されると破棄(換言すると、緩衝部は内容を空に)されます。未読データは失われます。例えば異常状態のため、通常動作中に緩衝部が破棄されなければならない場合、**USART制御/状態レジスタA(UCSRnA)**の**受信完了(RXCn)フラグ**が解除(**0**)されるまでUSARTデータレジスタ(UDRn)I/O位置を読んでください。

次のコード例は受信緩衝部の破棄方法を示します。

アセンブリ言語プログラム例

```
USART_Flush:  SBIS      UCSRnA, RXCn      ;未読データありでスキップ
               RET                               ;未読データなしで復帰
;
               IN        R16, UDRn        ;データ受信
               RJMP      USART_Flush      ;未読データなしまで継続
```

C言語プログラム例

```
void USART_Flush(void)
{
    unsigned char dummy;                /* 一時変数定義 */
    while ( UCSRnA & (1<<RXCn) ) dummy=UDRn; /* 未読データ読み捨て */
}
```

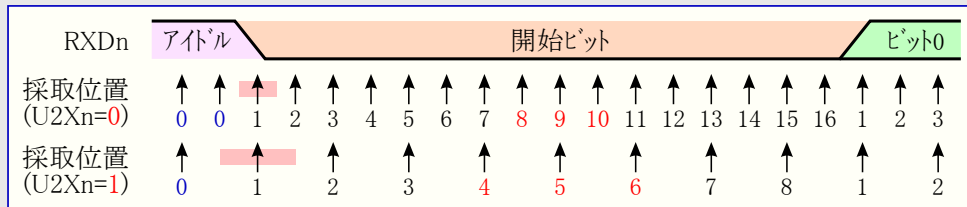
16.9. 非同期受信

USARTは非同期データ受信を扱うためのクロック再生とデータ再生部を含みます。クロック再生論理回路は内部的に生成したボーレートクロックをRXDnピンに到着する非同期直列フレームに同期化するのに使われます。データ再生論理回路は到着ビット毎に低域通過の濾波と採取をし、それによって受信部の雑音耐性を改善します。非同期受信動作範囲は内部ボーレートクロック精度、到着フレーム速度、フレーム長のビット数に依存します。

16.9.1. 非同期クロック再生

クロック再生論理回路は内部クロックを到着直列フレームに同期化します。下図は到着フレームの開始ビットの採取手順を図解します。採取速度は標準速動作でボーレートの16倍、倍速動作で8倍です。赤帯(※注:原文は水平矢印)は採取処理のための同期変量を図示します。倍速動作(U2Xn=1)を使う時の広い変量時間に注意してください。採取番号0はRXDn信号がアイドル(即ち、通信の動きなし)の間に行われる採取です。

図16-5. 開始ビットの採取

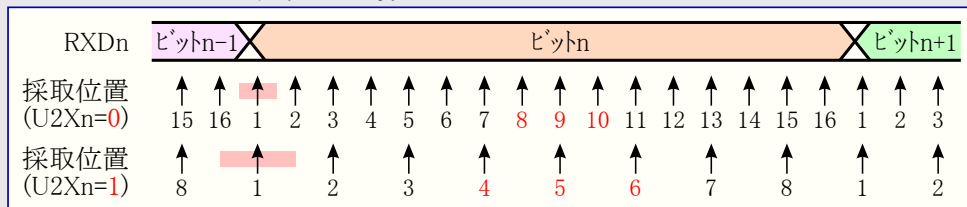


クロック再生論理回路がRXDn信号線で**High**(アイドル)から**Low**(開始)の遷移を検出すると、開始ビット検出手順が開始されます。図で示されるように採取1は最初の**Low**採取を意味します。その後クロック再生回路は有効な開始ビットが受信されるかを定めるために、標準速動作に対して採取8,9,10、倍速動作に対して採取4,5,6(図の赤数字(訳注:原文は箱枠内))で示された採取番号を使います。これら3回で2回以上の採取が論理**High**レベルの場合(多数決)、この開始ビットは尖頭雑音として捨てられ、受信部はRXDn上で次の**High**から**Low**の遷移を探し始めます。けれども、有効な開始ビットが検出された場合、クロック再生論理回路は同期化され、データ再生が開始されます。この同期化手順は各開始ビット毎に繰り返されます。

16.9.2. 非同期データ再生

受信部クロックが開始ビットに同期化されるとデータ再生が始まります。データ再生部は標準速動作で16段、倍速動作で8段の順列回路を使います。下図はデータビットとパリティビットの採取を示します。各採取は再生部の各段に等しい番号を与えられます。

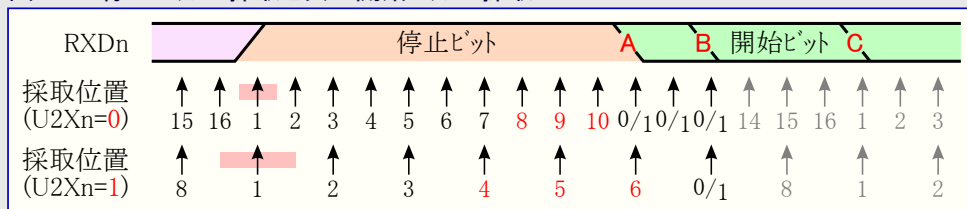
図16-6. データビットとパリティビットの採取



受信したビットの論理値の決定は受信したビット中央の3採取で論理値の多数決を取ることによって行われます。2つまたは全3つの中央採取(赤文字(訳注:原文は箱枠内)のそれらの採取番号によって記されるそれらがHighレベルならば受信したビットは論理1が記録されます。2つまたは全3つの採取がLowレベルならば受信したビットは論理0が記録されます。この多数決手順はRXDnピンの到着信号に対して低域通過濾波器(ローパスフィルタ)のように働きます。この再生手順はその後完全にフレームが受信されるまで繰り返されます。これには最初の(第1)停止ビットを含みます。受信部はフレームの最初の停止ビットだけを使います。

下図は停止ビットの採取と次フレームの最も早い開始ビット開始の可能性を示します。

図16-7. 停止ビットの採取と次の開始ビットの採取



フレーム内の他のビットに対して行われるのと同じ多数決が停止ビットにも行われます。停止ビットが論理0値と記録されると、**フレーミング異常 (FEn)フラグ**が設定(1)されます。

新規フレームの開始ビットを示すHighからLowへの遷移は多数決に使った最後のビット後に実現できます。標準速動作での最初のLowレベル採取は上図のA点で獲得され得ます。倍速動作での最初のLowレベルはB点に遅れます。C点は完全な長さの停止ビット(の終点)を示します。この早い開始ビット検出は受信部の動作範囲に影響します。

16.9.3. 非同期での動作範囲

受信部の動作範囲は受信したビット速度と内部的に生成したボーレート間の不一致に依存します。送信部が速すぎるまたは遅すぎるビット速度でフレームを送出したり、内部的に発生した受信部のボーレートが類似した基準周波数を持たない場合、受信部は開始ビットでフレームを同期できません(下の推奨をご覧ください)。

次式は到着データ速度と内部受信部ボーレート間の比率計算に使えます。

$$R_{\text{slow}} = \frac{(D+1) \times S}{S-1+D \times S+S_F} \quad R_{\text{fast}} = \frac{(D+2) \times S}{(D+1) \times S+S_M}$$

D : データとパリティのビット数 (5~10)
 S : ビットあたりの採取数 (標準速=16、倍速=8)
 S_F : 多数決に使う最初の採取番号 (標準速=8、倍速=4)
 S_M : 多数決に使う中心の採取番号 (標準速=9、倍速=5)
 R_{slow} : は受信側ボーレートに対して許容できる最低受信ビット速度の比率です。
 R_{fast} : は受信側ボーレートに対して許容できる最高受信ビット速度の比率です。

下表は許容できる最大受信部ボーレート誤差一覧です。標準速動作には、より高いボーレート変動許容能力があることに注目してください。

表16-2. 標準速と倍速での受信部ボーレート推奨最大許容誤差

| D | 標準速動作 (U2Xn=0) | | | | 倍速動作 (U2Xn=1) | | | |
|----|----------------|----------|-------------|-----------|---------------|----------|-------------|-----------|
| | Rslow(%) | Rfast(%) | 総合許容誤差(%) | 推奨許容誤差(%) | Rslow(%) | Rfast(%) | 総合許容誤差(%) | 推奨許容誤差(%) |
| 5 | 93.20 | 106.67 | -6.80~+6.67 | ±3.0 | 94.12 | 105.66 | -5.88~+5.66 | ±2.5 |
| 6 | 94.12 | 105.79 | -5.88~+5.79 | ±2.5 | 94.92 | 104.92 | -5.08~+4.92 | ±2.0 |
| 7 | 94.81 | 105.11 | -5.19~+5.11 | ±2.0 | 95.52 | 104.35 | -4.48~+4.35 | ±1.5 |
| 8 | 95.36 | 104.58 | -4.54~+4.58 | ±2.0 | 96.00 | 103.90 | -4.00~+3.90 | ±1.5 |
| 9 | 95.81 | 104.14 | -4.19~+4.14 | ±1.5 | 96.39 | 103.53 | -3.61~+3.53 | ±1.5 |
| 10 | 96.17 | 103.78 | -3.83~+3.78 | ±1.5 | 96.70 | 103.23 | -3.30~+3.23 | ±1.0 |

注: Dはデータビット数とパリティビットの合計ビット数です。

(訳注) 原書は表15-2に標準速、表15-3に倍速を記載していますが、比較が容易なように表16-2として纏めました。

受信部ボーレートの推奨最大許容誤差は最大総合許容誤差を送信部と受信部で等分割するという仮定の下で作られました。

受信部ボーレート誤差に対して2つの起こり得る原因があります。受信部のシステムクロック(OSC)は供給電圧範囲と温度範囲に関して常に若干の不安定性があります。システムクロックを生成するのにクリスタル発振子を使う時は殆ど問題ありませんが、(セラミック)振動子でのシステムクロックは振動子偏差に依存して2%を越えて異なるかもしれません。2つ目の誤り原因はより制御可能です。ボーレート発振器は欲したボーレートを得るためにシステム周波数の正確な分周を常に行うことはできません。この場合、可能ならば受け入れ可能な低い誤差を与えるUBRRn値が使えます。

16.9.4. フレーム開始検出

USARTフレーム開始検出器は開始ビット検出時にアイドル、A/D変換雑音低減、パワーダウン、スタンバイの休止動作形態からMCUを起こすことができます。

RXDnピンでHighからLowへの遷移が検出されると、内部8MHz発振器が給電されUSARTクロックが許可されます。ボーレートが内部8MHz発振器始動時間に関して充分遅ければ、始動後にデータフレームの残りを受信することができます。内部8MHz発振器の始動時間は供給電圧と温度で変化します。

USARTフレーム開始検出は非同期と同期の両動作形態で動きます。これはUSART制御/状態レジスタD(UCSRnD)のフレーム開始検出許可(SFDEn)ビットを(1)に書くことによって許可されます。USART開始割り込み許可(RXSIEn)ビットが設定(1)されるなら、開始検出時直ちにUSART受信開始割り込みが生成されます。

開始割り込みなしでこの機能使用時、開始検出論理回路は内部8MHz発振器とUSARTクロックを活性にし、同時にフレームが受信されつつあるだけです。他のクロックは受信完了割り込みがMCUを起こすまで停止されたままです。

同期動作形態での最大ボーレートは以下のようにデバイスが起こされる休止動作形態に依存します。

- ・ アイドルまたはA/D変換雑音低減動作 : システムクロック周波数/4
- ・ パワーダウンまたはスタンバイ動作 : 500kbps

非同期動作形態での最大ボーレートは以下のようにデバイスが起こされる休止動作形態に依存します。

- ・ アイドル動作 : 活動動作と同じ
- ・ 他の休止動作 : 次表をご覧ください。

表16-4. 標準動作での最大総ボーレート誤差

| ボーレート | フレームの大きさ | | | | | |
|------------|-------------|-------------|-------------|-------------|-------------|-------------|
| | 5ビット | 6ビット | 7ビット | 8ビット | 9ビット | 10ビット |
| 0~28.8kbps | +6.67~-5.88 | +5.79~-5.08 | +5.11~-4.48 | +4.58~-4.00 | +4.14~-3.61 | +3.78~-3.30 |
| 38.4kbps | +6.63~-5.88 | +5.75~-5.08 | +5.08~-4.48 | +4.55~-4.00 | +4.12~-3.61 | +3.76~-3.30 |
| 57.6kbps | +6.10~-5.88 | +5.30~-5.08 | +4.69~-4.48 | +4.20~-4.00 | +3.80~-3.61 | +3.47~-3.30 |
| 76.8kbps | +5.59~-5.88 | +4.85~-5.08 | +4.29~-4.48 | +3.85~-4.00 | +3.48~-3.61 | +3.18~-3.30 |
| 115.2kbps | +4.57~-5.88 | +3.97~-5.08 | +3.51~-4.48 | +3.15~-4.00 | +2.86~-3.61 | +2.61~-3.30 |

表16-5. 倍速動作での最大総ボーレート誤差

| ボーレート | フレームの大きさ | | | | | |
|------------|-------------|-------------|-------------|-------------|-------------|-------------|
| | 5ビット | 6ビット | 7ビット | 8ビット | 9ビット | 10ビット |
| 0~57.6kbps | +5.66~-4.00 | +4.92~-3.45 | +4.35~-3.03 | +3.90~-2.70 | +3.53~-2.44 | +3.23~-2.22 |
| 76.8kbps | +5.59~-4.00 | +4.85~-3.45 | +4.29~-3.03 | +3.85~-2.70 | +3.48~-2.44 | +3.18~-2.22 |
| 115.2kbps | +4.57~-4.00 | +3.97~-3.45 | +3.51~-3.03 | +3.15~-2.70 | +2.86~-2.44 | +2.61~-2.22 |

関連リンク [88頁の「UCSR0D – USART0制御/状態レジスタD」](#)

16.10. 複数プロセッサ通信動作

USART制御/状態レジスタA(UCSRnA)での**複数プロセッサ通信動作(MPCMn)ビット**の設定(1)はUSART受信部によって受信された到着フレームの選別機能を許可します。アドレス情報を含まないフレームは無視され、受信緩衝部に格納されません。これは同一直列バス経路で通信する複数MCUのシステムで、CPUによって扱われなければならない到着フレーム数を効果的に減らします。送信部はMPCMnビット設定によって影響されませんが、複数プロセッサ通信動作を利用するシステムの一部の時は違うふうに使われなければなりません。

受信部が5~8データビットを含むフレームを受信するように設定されるなら、最初の停止ビットはデータまたはアドレス情報を含むフレームかどうかを示します。受信部が9データビットのフレームに設定されるなら、**USART制御/状態レジスタB(UCSRnB)の受信第9(RXB8n)ビット**がアドレスとデータのフレームを識別するのに使われます。フレーム種別(最初の停止または第9)ビットが1の時にフレームはアドレスを含みます。フレーム種別ビットが0の時にそのフレームはデータフレームです。

複数プロセッサ通信動作は主MCUからのデータを多くの従MCUで受信することを可能にします。これはどのMCUがアドレス指定されるかを検出するため、最初にアドレスフレームを調べることによって行われます。特定の従MCUがアドレス指定されたなら、そのMCUは後続するデータフレームを通常のように受信し、一方その他の従MCUは他のアドレスフレームが受信されるまで受信したフレームを無視します。

16.10.1. 複数プロセッサ通信の使用法

主MCUとして動作するMCUは9ビットデータフレーム形式(**UCSZn=7**)を使えます。UCSRnBの**送信第9(TXB8n)ビット**はアドレスフレーム時に設定(1)、またはデータフレーム時に解除(0)されて送信されなければなりません。この場合、従MCUは9ビットデータフレーム形式の使用に設定されなければなりません。

複数プロセッサ通信動作でデータを交換するのに次の手順が使われるべきです。

1. 全ての従MCUは複数プロセッサ通信動作です(UCSRnAの**複数プロセッサ通信動作(MPCMn)ビット**が設定(1))。
2. 主MCUはアドレスフレームを送り、全ての従装置がこのフレームを受信し、これを読みます。従CPUでは通常のようにUCSRnAで**受信完了(RXCn)フラグ**が設定(1)されます。
3. 各従MCUは**USARTデータレジスタ(UDRn)**を読み、選択されたかを判定します。選択された場合はUCSRnAのMPCMnビットを解除(0)し、そうでなければ(非選択の場合は)MPCMnビット設定を保ち、次のアドレスフレームを待ちます。
4. アドレス指定されたMCUは新規アドレスフレームが受信されるまで全データフレームを受信します。MPCMnビットが未だ設定(1)されている他の従CPUはこのデータフレームを無視します。
5. 最後のデータフレームがアドレス指定されたMCUによって受信されると、アドレス指定されたMCUはMPCMnビットを設定(1)し、主装置からの新規アドレスフレームを待ちます。以降、処理は手順2.からを繰り返します。

5~8ビットデータフレーム形式のどの使用も可能ですが、受信側が使うnとn+1ビットデータフレーム形式間を切り替えなければならないため非実用的です。これは送信部と受信部が同じデータビット長設定を使うため、全二重(フルデュプレックス)動作を困難にします。5~8ビットデータフレームが使われる場合、最初の停止ビットがフレーム種別を示すのに使われるので、送信部は2停止ビット使用(**USBSn=1**)に設定されなければなりません。

MPCMnビットを設定(1)または解除(0)するのに読み-修正-書き(リッド モデファイ ライト)命令(**SBI**と**CBI**)を使ってはいけません。MPCMnビットは**送信完了(TXCn)フラグ**と同じI/O位置を共用しており、**SBI**または**CBI**命令を使うと偶然に解除(0)されるかもしれません。

16.11. ボーレート設定例

標準的なクリスタル発振子やセラミック振動子の周波数に対して最も共通して使われる非同期動作のボーレートは、下表で一覧されるようにUBRRn設定を使うことによって生成できます。目的のボーレートに対して設定誤差1.5%(標準速:U2Xn=0)、1.0%(倍速:U2Xn=1)以上を赤字で示します(訳注:原書の本文は0.5%未満を太字)。より高い誤差率でも受け入れ可能ですが、特に長い直列フレームで誤差率が高いと、受信部は雑音耐性が低下します(「非同期での動作範囲」項もご覧ください)。誤差率は次式を使って計算されます。

$$\text{誤差率(\%)} = \left(\frac{\text{UBRRn設定ボーレート(最近似値)}}{\text{目的のボーレート}} - 1 \right) \times 100(\%)$$

表16-6. Xtal、ボーレート対UBRRnH,UBRRnL設定 (UBRR=UBRRnH:UBRRnL)

| ボーレート (bps) | 1MHz | | | | 1.8432MHz | | | | 2MHz | | | | 2.4576MHz | | | |
|----------------|-------|-------|-------|-------|-----------|-------|--------|-------|-------|-------|-------|-------|-----------|-------|--------|-------|
| | U2X=0 | | U2X=1 | | U2X=0 | | U2X=1 | | U2X=0 | | U2X=1 | | U2X=0 | | U2X=1 | |
| | UBRR | 誤差(%) | UBRR | 誤差(%) | UBRR | 誤差(%) | UBRR | 誤差(%) | UBRR | 誤差(%) | UBRR | 誤差(%) | UBRR | 誤差(%) | UBRR | 誤差(%) |
| 1200 | 51 | 0.2 | 103 | 0.2 | 95 | 0.0 | 191 | 0.0 | 103 | 0.2 | 207 | 0.2 | 127 | 0.0 | 255 | 0.0 |
| 2400 | 25 | 0.2 | 51 | 0.2 | 47 | 0.0 | 95 | 0.0 | 51 | 0.2 | 103 | 0.2 | 63 | 0.0 | 127 | 0.0 |
| 4800 | 12 | 0.2 | 25 | 0.2 | 23 | 0.0 | 47 | 0.0 | 25 | 0.2 | 51 | 0.2 | 31 | 0.0 | 63 | 0.0 |
| 9600 | 6 | -7.0 | 12 | 0.2 | 11 | 0.0 | 23 | 0.0 | 12 | 0.2 | 25 | 0.2 | 15 | 0.0 | 31 | 0.0 |
| 14400 | 3 | 8.5 | 8 | -3.5 | 7 | 0.0 | 15 | 0.0 | 8 | -3.5 | 16 | 2.1 | 10 | -3.0 | 20 | 1.6 |
| 19200 | 2 | 8.5 | 6 | -7.0 | 5 | 0.0 | 11 | 0.0 | 6 | -7.0 | 12 | 0.2 | 7 | 0.0 | 15 | 0.0 |
| 28800 | 1 | 8.5 | 3 | 8.5 | 3 | 0.0 | 7 | 0.0 | 3 | 8.5 | 8 | -3.5 | 4 | 6.7 | 10 | -3.0 |
| 38400 | 1 | -18.6 | 2 | 8.5 | 2 | 0.0 | 5 | 0.0 | 2 | 8.5 | 6 | -7.0 | 3 | 0.0 | 7 | 0.0 |
| 57600 | 0 | 8.5 | 1 | 8.5 | 1 | 0.0 | 3 | 0.0 | 1 | 8.5 | 3 | 8.5 | 2 | -11.1 | 4 | 6.7 |
| 76800 | 0 | -18.6 | 1 | -18.6 | 1 | -25.0 | 2 | 0.0 | 1 | -18.6 | 2 | 8.5 | 1 | 0.0 | 3 | 0.0 |
| 115.2k | - | - | 0 | 8.5 | 0 | 0.0 | 1 | 0.0 | 0 | 8.5 | 1 | 8.5 | 0 | 33.3 | 2 | -11.1 |
| 230.4k | - | - | 0 | -45.7 | 0 | -50.0 | 0 | 0.0 | 0 | -45.7 | 0 | 8.5 | 0 | -33.3 | 0 | 33.3 |
| 250k | - | - | - | - | - | - | 0 | -7.8 | - | - | 0 | 0.0 | - | - | 0 | 22.9 |
| 最高速 | 62500 | | 125k | | 115.2k | | 230.4k | | 125k | | 250k | | 153.6k | | 307.2k | |

| ボーレート (bps) | 3.2768MHz | | | | 3.6864MHz | | | | 4MHz | | | | 4.608MHz | | | |
|----------------|-----------|-------|--------|-------|-----------|-------|--------|-------|-------|-------|-------|-------|----------|-------|-------|-------|
| | U2X=0 | | U2X=1 | | U2X=0 | | U2X=1 | | U2X=0 | | U2X=1 | | U2X=0 | | U2X=1 | |
| | UBRR | 誤差(%) | UBRR | 誤差(%) | UBRR | 誤差(%) | UBRR | 誤差(%) | UBRR | 誤差(%) | UBRR | 誤差(%) | UBRR | 誤差(%) | UBRR | 誤差(%) |
| 1200 | 170 | -0.2 | 340 | 0.1 | 191 | 0.0 | 384 | 0.0 | 207 | 0.2 | 416 | -0.1 | 239 | 0.0 | 479 | 0.0 |
| 2400 | 84 | 0.4 | 170 | -0.2 | 95 | 0.0 | 191 | 0.0 | 103 | 0.2 | 207 | 0.2 | 119 | 0.0 | 239 | 0.0 |
| 4800 | 42 | -0.8 | 84 | 0.4 | 47 | 0.0 | 95 | 0.0 | 51 | 0.2 | 103 | 0.2 | 59 | 0.0 | 119 | 0.0 |
| 9600 | 20 | 1.6 | 42 | -0.8 | 23 | 0.0 | 47 | 0.0 | 25 | 0.2 | 51 | 0.2 | 29 | 0.0 | 59 | 0.0 |
| 14400 | 13 | 1.6 | 27 | 1.6 | 15 | 0.0 | 31 | 0.0 | 16 | 2.1 | 34 | -0.8 | 19 | 0.0 | 39 | 0.0 |
| 19200 | 10 | -3.0 | 20 | 1.6 | 11 | 0.0 | 23 | 0.0 | 12 | 0.2 | 25 | 0.2 | 14 | 0.0 | 29 | 0.0 |
| 28800 | 6 | 1.6 | 13 | 1.6 | 7 | 0.0 | 15 | 0.0 | 8 | -3.5 | 16 | 2.1 | 9 | 0.0 | 19 | 0.0 |
| 38400 | 4 | 6.7 | 10 | -3.0 | 5 | 0.0 | 11 | 0.0 | 6 | -7.0 | 12 | 0.2 | 7 | -6.3 | 14 | 0.0 |
| 57600 | 3 | -11.1 | 6 | 1.6 | 3 | 0.0 | 7 | 0.0 | 3 | 8.5 | 8 | -3.5 | 4 | 0.0 | 9 | 0.0 |
| 76800 | 2 | -11.1 | 4 | 6.7 | 2 | 0.0 | 5 | 0.0 | 2 | 8.5 | 6 | -7.0 | 3 | -6.3 | 7 | -6.3 |
| 115.2k | 1 | -11.1 | 3 | -11.1 | 1 | 0.0 | 3 | 0.0 | 1 | 8.5 | 3 | 8.5 | 2 | -16.7 | 4 | 0.0 |
| 230.4k | 0 | -11.1 | 1 | -11.1 | 0 | 0.0 | 1 | 0.0 | 0 | 8.5 | 1 | 8.5 | 0 | 25.0 | 2 | -16.7 |
| 250k | - | - | 1 | -18.1 | 0 | -7.8 | 1 | -7.8 | 0 | 0.0 | 1 | 0.0 | 0 | 15.2 | 1 | 15.2 |
| 500k | - | - | 0 | -18.1 | - | - | 0 | -7.8 | - | - | 0 | 0.0 | 0 | -42.4 | 0 | 15.2 |
| 1M | - | - | - | - | - | - | - | - | - | - | - | - | - | - | 0 | -42.4 |
| 最高速 | 204.8k | | 409.6k | | 230.4k | | 460.8k | | 250k | | 500k | | 288k | | 576k | |

注: 最高速はUBRR=0、誤差=0.0%です。以降の周波数は次ページへ続く。

(訳注) 原書では頁割された表毎に表番号15-6.~9.となっていますが、共通性から纏めて表16-6.としました。
原書に対して数種の発振周波数を追加しました。

表16-6 (続き). Xtal、ボート対UBRRnH,UBRRnL設定 (UBRR=UBRRnH:UBRRnL)

| ボート (bps) | 4.9152MHz | | | | 6.144MHz | | | | 7.3728MHz | | | | 8MHz | | | |
|--------------|-----------|-------|--------|-------|----------|-------|-------|-------|-----------|-------|--------|-------|-------|-------|-------|-------|
| | U2X=0 | | U2X=1 | | U2X=0 | | U2X=1 | | U2X=0 | | U2X=1 | | U2X=0 | | U2X=1 | |
| | UBRR | 誤差(%) | UBRR | 誤差(%) | UBRR | 誤差(%) | UBRR | 誤差(%) | UBRR | 誤差(%) | UBRR | 誤差(%) | UBRR | 誤差(%) | UBRR | 誤差(%) |
| 1200 | 255 | 0.0 | 511 | 0.0 | 319 | 0.0 | 639 | 0.0 | 383 | 0.0 | 767 | 0.0 | 416 | -0.1 | 832 | 0.0 |
| 2400 | 127 | 0.0 | 255 | 0.0 | 159 | 0.0 | 319 | 0.0 | 191 | 0.0 | 383 | 0.0 | 207 | 0.2 | 416 | -0.1 |
| 4800 | 63 | 0.0 | 127 | 0.0 | 79 | 0.0 | 159 | 0.0 | 95 | 0.0 | 191 | 0.0 | 103 | 0.2 | 207 | 0.2 |
| 9600 | 31 | 0.0 | 63 | 0.0 | 39 | 0.0 | 79 | 0.0 | 47 | 0.0 | 95 | 0.0 | 51 | 0.2 | 103 | 0.2 |
| 14400 | 20 | 1.6 | 42 | -0.8 | 26 | -1.2 | 52 | 0.6 | 31 | 0.0 | 63 | 0.0 | 34 | -0.8 | 68 | 0.6 |
| 19200 | 15 | 0.0 | 31 | 0.0 | 19 | 0.0 | 39 | 0.0 | 23 | 0.0 | 47 | 0.0 | 25 | 0.2 | 51 | 0.2 |
| 28800 | 10 | -3.0 | 20 | 1.6 | 12 | 2.6 | 26 | -1.2 | 15 | 0.0 | 31 | 0.0 | 16 | 2.1 | 34 | -0.8 |
| 38400 | 7 | 0.0 | 15 | 0.0 | 9 | 0.0 | 19 | 0.0 | 11 | 0.0 | 23 | 0.0 | 12 | 0.2 | 25 | 0.2 |
| 57600 | 4 | 6.7 | 10 | -3.0 | 6 | -4.8 | 12 | 2.6 | 7 | 0.0 | 15 | 0.0 | 8 | -3.5 | 16 | 2.1 |
| 76800 | 3 | 0.0 | 7 | 0.0 | 4 | 0.0 | 9 | 0.0 | 5 | 0.0 | 11 | 0.0 | 6 | -7.0 | 12 | 0.2 |
| 115.2k | 2 | -11.1 | 4 | 6.7 | 2 | 11.1 | 6 | -4.8 | 3 | 0.0 | 7 | 0.0 | 3 | 8.5 | 8 | -3.5 |
| 230.4k | 1 | 33.3 | 2 | -11.1 | 1 | -16.7 | 2 | 11.1 | 1 | 0.0 | 3 | 0.0 | 1 | 8.5 | 3 | 8.5 |
| 250k | 1 | 22.9 | 2 | -18.1 | 1 | -23.2 | 2 | 2.4 | 1 | -7.8 | 3 | 0.0 | 1 | 0.0 | 3 | 0.0 |
| 500k | 0 | -38.6 | 0 | 22.9 | 0 | -23.2 | 1 | -23.2 | 0 | -7.8 | 1 | -7.8 | 0 | 0.0 | 1 | 0.0 |
| 1M | - | - | 0 | -38.6 | - | - | 0 | -23.2 | - | - | 0 | -7.8 | - | - | 0 | 0.0 |
| 最高速 | 307.2k | | 614.4k | | 384k | | 768k | | 460.8k | | 921.6k | | 500k | | 1M | |

| ボート (bps) | 9.216MHz | | | | 9.8304MHz | | | | 10MHz | | | | 11.0592MHz | | | |
|--------------|----------|-------|--------|-------|-----------|-------|---------|-------|-------|-------|-------|-------|------------|-------|---------|-------|
| | U2X=0 | | U2X=1 | | U2X=0 | | U2X=1 | | U2X=0 | | U2X=1 | | U2X=0 | | U2X=1 | |
| | UBRR | 誤差(%) | UBRR | 誤差(%) | UBRR | 誤差(%) | UBRR | 誤差(%) | UBRR | 誤差(%) | UBRR | 誤差(%) | UBRR | 誤差(%) | UBRR | 誤差(%) |
| 1200 | 479 | 0.0 | 959 | 0.0 | 511 | 0.0 | 1023 | 0.0 | 520 | -0.0 | 1041 | -0.0 | 575 | 0.0 | 1151 | 0.0 |
| 2400 | 239 | 0.0 | 479 | 0.0 | 255 | 0.0 | 511 | 0.0 | 259 | 0.2 | 520 | -0.0 | 287 | 0.0 | 575 | 0.0 |
| 4800 | 119 | 0.0 | 239 | 0.0 | 127 | 0.0 | 255 | 0.0 | 129 | 0.2 | 259 | 0.2 | 143 | 0.0 | 287 | 0.0 |
| 9600 | 59 | 0.0 | 119 | 0.0 | 63 | 0.0 | 127 | 0.0 | 64 | 0.2 | 129 | 0.2 | 71 | 0.0 | 143 | 0.0 |
| 14400 | 39 | 0.0 | 79 | 0.0 | 42 | -0.8 | 84 | 0.4 | 42 | 0.9 | 86 | -0.2 | 47 | 0.0 | 95 | 0.0 |
| 19200 | 29 | 0.0 | 59 | 0.0 | 31 | 0.0 | 63 | 0.0 | 32 | -1.4 | 64 | 0.2 | 35 | 0.0 | 71 | 0.0 |
| 28800 | 19 | 0.0 | 39 | 0.0 | 20 | 1.6 | 42 | -0.8 | 21 | -1.4 | 42 | 0.9 | 23 | 0.0 | 47 | 0.0 |
| 38400 | 14 | 0.0 | 29 | 0.0 | 15 | 0.0 | 31 | 0.0 | 15 | 1.8 | 32 | -1.4 | 17 | 0.0 | 35 | 0.0 |
| 57600 | 9 | 0.0 | 19 | 0.0 | 10 | -3.0 | 20 | 1.6 | 10 | -1.4 | 21 | -1.4 | 11 | 0.0 | 23 | 0.0 |
| 76800 | 7 | 6.7 | 14 | 0.0 | 7 | 0.0 | 15 | 0.0 | 7 | 1.8 | 15 | 1.8 | 8 | 0.0 | 17 | 0.0 |
| 115.2k | 4 | 0.0 | 9 | 0.0 | 4 | 6.7 | 10 | -3.0 | 4 | 8.5 | 10 | -1.4 | 5 | 0.0 | 11 | 0.0 |
| 230.4k | 2 | -16.7 | 4 | 0.0 | 2 | -11.1 | 4 | 6.7 | 2 | -9.6 | 4 | 8.5 | 2 | 0.0 | 5 | 0.0 |
| 250k | 1 | 15.2 | 4 | -7.8 | 1 | -23.2 | 4 | -1.7 | 2 | -16.7 | 4 | 0.0 | 2 | -7.8 | 5 | -7.8 |
| 500k | 0 | 15.2 | 1 | 15.2 | 0 | 22.9 | 1 | 22.9 | 0 | 25.0 | 2 | -16.7 | 0 | 38.2 | 2 | -7.8 |
| 1M | 0 | -42.4 | 0 | 15.2 | 0 | -38.6 | 0 | 22.9 | 0 | -37.5 | 0 | 25.0 | 0 | -30.9 | 0 | 38.2 |
| 最高速 | 576k | | 1.152M | | 614.4k | | 1.2288M | | 625k | | 1.25M | | 691.2k | | 1.3824M | |

| ボート (bps) | 14.7456MHz | | | | 16MHz | | | | 18.4320MHz | | | | 20MHz | | | |
|--------------|------------|-------|---------|-------|-------|-------|-------|-------|------------|-------|--------|-------|-------|-------|-------|-------|
| | U2X=0 | | U2X=1 | | U2X=0 | | U2X=1 | | U2X=0 | | U2X=1 | | U2X=0 | | U2X=1 | |
| | UBRR | 誤差(%) | UBRR | 誤差(%) | UBRR | 誤差(%) | UBRR | 誤差(%) | UBRR | 誤差(%) | UBRR | 誤差(%) | UBRR | 誤差(%) | UBRR | 誤差(%) |
| 1200 | 767 | 0.0 | 1533 | 0.0 | 832 | 0.0 | 1666 | 0.0 | 959 | 0.0 | 1919 | 0.0 | 1041 | 0.0 | 2082 | 0.0 |
| 2400 | 383 | 0.0 | 767 | 0.0 | 416 | -0.1 | 832 | 0.0 | 479 | 0.0 | 959 | 0.0 | 520 | 0.0 | 1041 | 0.0 |
| 4800 | 191 | 0.0 | 383 | 0.0 | 207 | 0.2 | 416 | -0.1 | 239 | 0.0 | 479 | 0.0 | 259 | 0.2 | 520 | 0.0 |
| 9600 | 95 | 0.0 | 191 | 0.0 | 103 | 0.2 | 207 | 0.2 | 119 | 0.0 | 239 | 0.0 | 129 | 0.2 | 259 | 0.2 |
| 14400 | 63 | 0.0 | 127 | 0.0 | 68 | 0.6 | 138 | -0.1 | 79 | 0.0 | 159 | 0.0 | 86 | -0.2 | 173 | -0.2 |
| 19200 | 47 | 0.0 | 95 | 0.0 | 51 | 0.2 | 103 | 0.2 | 59 | 0.0 | 119 | 0.0 | 64 | 0.2 | 129 | 0.2 |
| 28800 | 31 | 0.0 | 63 | 0.0 | 34 | -0.8 | 68 | 0.6 | 39 | 0.0 | 79 | 0.0 | 42 | 0.9 | 86 | -0.2 |
| 38400 | 23 | 0.0 | 47 | 0.0 | 25 | 0.2 | 51 | 0.2 | 29 | 0.0 | 59 | 0.0 | 32 | -1.4 | 64 | 0.2 |
| 57600 | 15 | 0.0 | 31 | 0.0 | 16 | 2.1 | 34 | -0.8 | 19 | 0.0 | 39 | 0.0 | 21 | -1.4 | 42 | 0.9 |
| 76800 | 11 | 0.0 | 23 | 0.0 | 12 | 0.2 | 25 | 0.2 | 14 | 0.0 | 29 | 0.0 | 15 | 1.7 | 32 | -1.4 |
| 115.2k | 7 | 0.0 | 15 | 0.0 | 8 | -3.5 | 16 | 2.1 | 9 | 0.0 | 19 | 0.0 | 10 | -1.4 | 21 | -1.4 |
| 230.4k | 3 | 0.0 | 7 | 0.0 | 3 | 8.5 | 8 | -3.5 | 4 | 0.0 | 9 | 0.0 | 4 | 8.5 | 10 | -1.4 |
| 250k | 3 | -7.8 | 6 | 5.3 | 3 | 0.0 | 7 | 0.0 | 4 | -7.8 | 8 | 2.4 | 4 | 0.0 | 9 | 0.0 |
| 500k | 1 | -7.8 | 3 | -7.8 | 1 | 0.0 | 3 | 0.0 | 1 | 15.2 | 4 | -7.8 | 1 | 25.0 | 4 | 0.0 |
| 1M | 0 | -7.8 | 1 | -7.8 | 0 | 0.0 | 1 | 0.0 | 0 | 15.2 | 1 | 15.2 | 0 | 25.0 | 1 | 25.0 |
| 最高速 | 921.6k | | 1.8432M | | 1M | | 2M | | 1.152M | | 2.304M | | 1.25M | | 2.5M | |

注: 最高速はUBRR=0、誤差=0.0%です。

16.12. USART用レジスタ

全てのUSARTレジスタはSBIとCBIの命令を使ってアクセスすることはできません。

16.12.1. UDR0 – USART0データレジスタ (USART I/O Data Register 0)

同じI/Oアドレスを共用するUSART受信データ緩衝レジスタとUSART送信データ緩衝レジスタはUSARTデータレジスタまたはUDR0として引用しました。送信データ緩衝レジスタ(TXB)はUDR0レジスタ位置に書かれるデータの転送先です。UDR0レジスタ位置読み込みは受信データ緩衝レジスタ(RXB)の内容を返します。

5～7ビットデータでの上位未使用ビットは送信部によって無視され、受信部によって0に設定されます。

送信緩衝部はUCSR0Aで送信データレジスタ空き(UDRE0)フラグが設定(1)される時にだけ書けます。UDRE0フラグが設定(1)されない時にUDR0へ書かれたデータはUSART送信部によって無視されます。送信部が許可されて送信緩衝部にデータが書かれると、送信部は送信移動レジスタが空きの時にそのデータを送信移動レジスタへ設定(移動)します。その後データはTXD0ピンで直列送信されます。

受信緩衝部は2段のFIFOから成ります。このFIFOは受信緩衝部がアクセスされる毎にその状態を切り替えます。この受信緩衝部の動きのため、この位置に読み-修正-書き(リード モデファイ ライト)命令(SBIとCBI)を使えません。ビット検査(SBICとSBIS)命令もFIFOの状態を換えるので、これらの命令を使う時は注意してください。

名称 : UDR0

変位 : \$08

リセット : \$00

特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|------------|-----|-----|-----|-----|-----|-----|-----|
| | TXB/RXB7~0 | | | | | | | |
| アクセス種別 | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

● ビット7～0 – TXB7～0/RXB7～0 : USART送受信データ (USART Transmit / Receive Data Buffer)

16.12.2. UCSR0A – USART0制御/状態レジスタA (USART Control and Status Register 0 A)

名称 : UCSR0A

変位 : \$0E

リセット : \$20

特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|------|------|-------|-----|------|------|------|-------|
| | RXC0 | TXC0 | UDRE0 | FE0 | DOR0 | UPE0 | U2X0 | MPCM0 |
| アクセス種別 | R | R/W | R | R | R | R | R/W | R/W |
| リセット値 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

● ビット7 – RXC0 : USART受信完了フラグ (USART Receive Complete)

このフラグは受信緩衝部に未読データがある時に設定(1)され、受信緩衝部が空の(換言すると、どんな未読データも含まない)時に解除(0)されます。受信部が禁止されると、受信緩衝部が破棄され、その結果RXC0フラグは0になります。RXC0フラグは受信完了割り込みを発生するのに使えます(USART制御/状態レジスタB(UCSR0B)の受信完了割り込み許可(RXCIE0)ビットをご覧ください)。

● ビット6 – TXC0 : USART送信完了フラグ (USART Transmit Complete)

このフラグは送信移動レジスタ内の完全なフレームが移動出力されてしまい、送信緩衝部(UDR0)に新規データが現存しない時に設定(1)されます。TXC0フラグは送信完了割り込みが実行されるとき、自動的に解除(0)されるか、またはこのビット位置に1を書くことによって解除(0)できます。TXC0フラグは送信完了割り込みを発生できます(UCSR0Bの送信完了割り込み許可(TXCIE0)ビットをご覧ください)。

● ビット5 – UDRE0 : USART送信データレジスタ空きフラグ (USART Data Register Empty)

UDRE0フラグは送信緩衝部(UDR0)が新規データを受け取る準備ができているかどうかを示します。UDRE0が1ならば緩衝部は空で、従って書かれる準備ができています。UDRE0フラグは送信緩衝部空き割り込みを発生できます(UCSR0Bの送信データレジスタ空き割り込み許可(UDRIE0)ビットをご覧ください)。送信部が準備できているのを示すため、リセット後のUDRE0は設定(1)です。

● ビット4 – FE0 : フレーミング異常フラグ (Framing Error)

受信緩衝部の次データが受信した時にフレーミング異常(換言すると、受信緩衝部で次データの最初の(第1)停止ビットがLow)だった場合、このビットが設定(1)されます。このフラグは受信緩衝部(UDR0)が読まれるまで有効です。受信したデータの停止ビットがHighの時にFE0フラグは0です。UCSR0Aに書くとき、常にこのビットを0に設定してください。

このビットは主装置SPI動作(MSPIM)で保留されています。

●ビット3 – DOR0 : データ オーバーラン発生フラグ (Data OverRun)

データ オーバーラン(DOR0)フラグは受信緩衝部満杯状況のためのデータ損失を示します。データ オーバーランは受信緩衝部が満杯(2文字)で、新規フレームが受信移動レジスタ内で待機中に新規開始ビットが検出されると起こります。

このビットが設定(1)された場合、UDR0から読んだ最後のフレームとUDR0から読む次のフレーム間に1つまたはそれ以上の直列フレームが失われます。将来のデバイスとの互換性のため、UCSR0Aを書く時には常にこのビットへ0を書いてください。このビットは受信したフレームが移動レジスタから受信緩衝部へ成功裏に移動された時に解除(0)されます。

このビットは主装置SPI動作(MSPIM)で保留されています。

●ビット2 – UPE0 : パリティ誤りフラグ (USART Parity Error)

受信緩衝部の次データが受信した時にパリティ異常があり、その時点でパリティ検査が許可されていれば(UPM01=1)、このビットが設定(1)されます。このフラグは受信緩衝部(UDR0)が読まれるまで有効です。UCSR0Aを書くとき、常にこのビットを0に設定してください。

このビットは主装置SPI動作(MSPIM)で保留されています。

●ビット1 – U2X0 : 倍速許可 (Double the USART Transmission Speed)

このビットは非同期動作でだけ有効です。同期動作を使うとき、このビットに0を書いてください。

このビットに1を書くことはボーレート分周器の分周値を16から8に減らして事実上、非同期通信の転送速度を倍にします。

このビットは主装置SPI動作(MSPIM)で保留されています。

●ビット0 – MPCM0 : 複数プロセッサ通信動作 (Multi-Processor Communication Mode)

このビットは複数プロセッサ通信動作を許可します。MPCM0ビットが1を書かれると、USART受信部によって受信したアドレス情報を含まない全到着フレームは無視されます。送信部はMPCM0設定に影響されません。

このビットは主装置SPI動作(MSPIM)で保留されています。

16.12.3. UCSR0B – USART0制御/状態レジスタB (USART Control and Status Register 0 B)

名称 : UCSR0B

変位 : \$0D

リセット : \$00

特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|--------|--------|--------|-------|-------|--------|-------|-------|
| | RXCIE0 | TXCIE0 | UDRIE0 | RXEN0 | TXEN0 | UCSZ02 | RXB80 | TXB80 |
| アクセス種別 | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

●ビット7 – RXCIE0 : 受信完了割り込み許可 (Receive Complete Interrupt Enable)

このビットへの1書き込みはUSART制御/状態レジスタA(UCSR0A)の受信完了(RXC0)フラグでの割り込みを許可します。USART受信完了割り込みはRXCIE0ビットが1を書かれ、ステータスレジスタ(SREG)で全割り込み許可(I)ビットが1を書かれ、UCSR0AでRXC0フラグが設定(1)される場合にだけ生成されます。

●ビット6 – TXCIE0 : 送信完了割り込み許可 (Transmit Complete Interrupt Enable)

このビットへの1書き込みはUSART制御/状態レジスタA(UCSR0A)の送信完了(TXC0)フラグでの割り込みを許可します。USART送信完了割り込みはTXCIE0ビットが1を書かれ、SREGで全割り込み許可(I)ビットが1を書かれ、UCSR0AでTXC0フラグが設定(1)される場合にだけ生成されます。

●ビット5 – UDRIE0 : 送信データレジスタ空き割り込み許可 (Transmit Data Register Empty Interrupt Enable)

このビットへの1書き込みはUSART制御/状態レジスタA(UCSR0A)の送信データレジスタ空き(UDRE0)フラグでの割り込みを許可します。USART送信データレジスタ空き割り込みはUDRIE0ビットが1を書かれ、SREGで全割り込み許可(I)ビットが1を書かれ、UCSR0AでUDRE0フラグが設定(1)される場合にだけ生成されます。

●ビット4 – RXEN0 : 受信許可 (Receiver Enable)

このビットへの1書き込みはUSART受信(部)を許可します。受信部は許可されるとRXD0ピンの標準ポート動作を無効にします。受信の禁止は受信緩衝部を破棄し、フレーミング異常(FE0)、オーバーラン(DOR0)、パリティ誤り(UPE0)のフラグを無効にします。

●ビット3 – TXEN0 : 送信許可 (Transmitter Enable)

このビットへの1書き込みはUSART送信(部)を許可します。送信部は許可されるとTXD0ピンの標準ポート動作を無効にします。送信の禁止(TXEN0=0書き込み)は進行中と保留中の送信が完了される(換言すると、送信移動レジスタと送信緩衝レジスタが送信されるべきデータを含まない)まで有効になりません。禁止したとき、送信部はもはやTXD0ポートの標準I/O機能を無効にしません。

●ビット2 – UCSZ02 : データビット長選択2 (Character Size)

USART制御/状態レジスタC(UCSR0C)のUCSZ01,0ビットと組み合わせたUCSZ02ビットは送受信部で使うフレームのデータビット数(Character size)を設定します。

このビットは主装置SPI動作(MSPIM)で保留されています。

●ビット1 – RXB80 : 受信データビット8 (Receive Data Bit 8)

RXB80は9ビットデータでの直列フレーム操作時に受信したフレームの第9データビット(ビット8)です。UDR0から下位ビットを読む前に読んでください。

このビットは主装置SPI動作(MSPIM)で保留されています。

●ビット0 – TXB80 : 送信データビット8 (Transmit Data Bit 8)

TXB80は9ビットデータでの直列フレーム操作時に送信されるべきデータの第9データビット(ビット8)です。UDR0へ下位ビットを書く前に書いてください。

このビットは主装置SPI動作(MSPIM)で保留されています。

16.12.4. UCSR0C – USART0制御/状態レジスタC (USART Control and Status Register 0 C)

名称 : UCSR0C

変位 : \$0C

リセット : \$06

特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|-----------|-----|---------|-----|-------|------------------|------------------|--------|
| | UMSEL01,0 | | UPM01,0 | | USBS0 | UCSZ01 UDORD0 | UCSZ00 UCPHA0 | UCPOL0 |
| アクセス種別 | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |

●ビット7,6 – UMSEL01,0 : USART動作選択 (USART Mode Select)

このビットはUSART動作種別を選びます。

表16-10. USART動作選択

| UMSEL01,0 | 動作種別 |
|-----------|--------------------|
| 0 0 | 非同期動作 |
| 0 1 | 同期動作 |
| 1 0 | (予約) |
| 1 1 | 主装置SPI (MSPIM) (注) |

注: UDORD0、UCPHA0、UCPOL0は主装置SPI動作(MSPIM)許可での同じ書き込み操作で設定することができます。

●ビット5,4 – UPM01,0 : パリティ選択 (Parity Mode)

これらのビットはパリティの発生と検査の許可と種別を設定します。許可した場合、送信部は各フレーム内での送信したデータビットのパリティを自動的に生成して送出します。受信部は到着データからパリティ値を生成し、UPM00設定と比較します。不一致が検出されると、USART制御/状態レジスタA(UCSR0A)でパリティ誤り(UPE0)フラグが設定(1)されます。

これらのビットは主装置SPI動作(MSPIM)で保留されています。

表16-11. パリティ選択

| UPM01,0 | パリティ動作 |
|---------|----------|
| 0 0 | 禁止 |
| 0 0 | (予約) |
| 1 0 | 偶数パリティ許可 |
| 1 1 | 奇数パリティ許可 |

●ビット3 – USBS0 : 停止ビット選択 (Stop Bit Select)

このビットは送信部によって挿入される停止ビット数を選びます。受信部はこの設定を無視します(訳補:常に第1停止ビットだけが有効)。

このビットは主装置SPI動作(MSPIM)で保留されています。

表16-12. 停止ビット選択

| USBS0 | 停止ビット数 |
|-------|--------|
| 0 | 1ビット |
| 1 | 2ビット |

●ビット2 – UCSZ01/UDORD0 : データビット長選択 (Character Size) / データ順選択 (Data Order)

UCSZ01:USART動作 : USART制御/状態レジスタB(UCSR0B)のUCSZ02ビットと組み合わせたUCSZ01,0ビットは送受信部で使うフレームのデータビット数(Character size)を設定します。

表16-13. データビット長選択

| UCSZ02~0 | 0 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1 0 0 | 1 0 1 | 1 1 0 | 1 1 1 |
|----------|-------|-------|-------|-------|-------|-------|-------|-------|
| データビット数 | 5ビット | 6ビット | 7ビット | 8ビット | (予約) | (予約) | (予約) | 9ビット |

UDORD0: 主装置SPI動作 : 1に設定されるとデータ語のLSBが最初に転送されます。0に設定されるとデータ語のMSBが最初に転送されます。詳細については「USARTでのSPI動作」の「フレーム形式」を参照してください。

●ビット1 – UCSZ00/UCPHA0 : データビット長選択 (Character Size) / クロック位相選択 (Clock Phase)

UCSZ00:USART動作 : UCSZ01を参照してください。

UDORD0: 主装置SPI動作 : クロック位相選択(UCPHA0)ビットの設定はデータがXCK0の先行(先)端または後行(後)端で採取/(設定)されるかを決めます。詳細については「USARTでのSPI動作」の「SPIデータ形態とタイミング」を参照してください。

●ビット0 – UCPOL0 : クロック極性選択 (Clock Polarity)

USART動作 : このビットは同期動作に対してだけ使われます。非同期動作が使われるとき、このビットに0を書いてください。UCPOL0ビットは同期クロック(XCK0)、データ出力変更、データ入力採取間の関係を設定します。

表16-14. XCKクロック極性選択

| UCPOL0 | 送信データ変更 (TXD0ピン出力) | 受信データ採取 (RXD0ピン入力) |
|--------|--------------------|--------------------|
| 0 | XCK0の上昇端 | XCK0の下降端 |
| 1 | XCK0の下降端 | XCK0の上昇端 |

主装置SPI動作 : UCPOL0ビットはXCK0クロックの極性を設定します。UCPOL0とクロック位相選択(UCPHA0)ビットの組み合わせ設定がデータ転送のタイミングを決めます。詳細については「USARTでのSPI動作」の「SPIデータ形態とタイミング」を参照してください。

16.12.5. UCSR0D – USART0制御/状態レジスタD (USART Control and Status Register 0 D)

このレジスタは主装置SPI動作(UCSR0C.UMSEL01,0=11)で使われません。

名称 : UCSR0D
変位 : \$0B
リセット : \$00
特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|--------|------|-------|---|---|---|---|---|
| | RXSIE0 | RXS0 | SFDE0 | - | - | - | - | - |
| アクセス種別 | R/W | R/W | R/W | R | R | R | R | R |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

●ビット7 – RXSIE0 : 受信開始割り込み許可 (USART RX Start Interrupt Enable)

このビットへの1書き込みは受信開始(RXS0)フラグでの割り込みを許可します。休止動作形態でこのビットは、RXD0ピンで開始条件が検出された時にMCUを起こすことができるフレーム開始割り込みを許可します。

USART受信開始割り込みはRXSIE0ビット、ステータスレジスタ(SREG)の全割り込み許可(I)ビットとUCSR0DのRXS0フラグが設定(1)される場合にだけ生成されます。

●ビット6 – RXS0 : 受信開始 (USART RX Start)

このフラグはRXD0ピンで開始条件が検出された時に設定(1)されます。受信開始割り込み許可(RXSIE0)ビットとSREGの全割り込み許可(I)ビットが設定(1)されていれば、このフラグが設定(1)される時に受信開始割り込みが生成されます。このフラグはRXS0ビット位置に論理1を書くことによって解除(0)することができます。

フレーム開始検出器が許可(RXSIE0=1)され、全割り込み許可(I)ビットが設定(1)されていれば、受信開始割り込みは全ての休止動作形態からMCUを起こします。

●ビット5 – SFDE0 : フレーム開始検出許可 (Start Frame Detection Enable)

このビットへの1書き込みはUSARTフレーム開始動作を許可します。フレーム開始検出器は開始条件、換言すると、RXDnピンでHigh(アイドル)からLow(開始)への遷移が検出される時に休止動作形態からMCUを起こすことができます。

表26-15. USARTフレーム開始検出動作

| SFDE | RXSIE | RXCIE | 説明 |
|------|-------|-------|---|
| 0 | x | x | フレーム開始検出禁止。 |
| 1 | 0 | 0 | (予約) |
| 1 | 0 | 1 | フレーム開始検出許可。RXCフラグは全ての休止動作形態からMCUを起こします。 |
| 1 | 1 | 0 | フレーム開始検出許可。RXSフラグは全ての休止動作形態からMCUを起こします。 |
| 1 | 1 | 1 | フレーム開始検出許可。RXCとRXSの両フラグは全ての休止動作形態からMCUを起こします。 |

16.12.6. UBRR0H,UBRR0L – USART0ボーレートレジスタ上位/下位バイト (USART Baud Rate Register 0 Low and High byte)

UBRR0HとUBRR0Lのレジスタ対は16ビット値のUBRR0を表します。下位バイト[7～0](接尾辞L)は変位原点でアクセスできます。上位バイト[15～8](接尾辞H)は変位+1でアクセスすることができます。16ビットレジスタの読み書きのより多くの詳細については「16ビットレジスタのアクセス」をご覧ください。

名称 : UBRR0HとUBRR0L
変位 : \$09
リセット : \$0000
特質 : -

| ビット | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|--------|----------|-----|-----|-----|-----------|-----|-----|-----|
| | - | | | | UBRR011~8 | | | |
| アクセス種別 | R | R | R | R | R/W | R/W | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | UBRR07~0 | | | | | | | |
| アクセス種別 | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

● ビット11～0 – UBRR011～0 : ボーレート分周値 (USART Baud Rate)

これはUSARTのボーレートを含む12ビットレジスタです。UBRR0HがUSARTボーレートの上位4ビットを含み、UBRR0Lが下位8ビットを含みます。ボーレートが変更されると、送受信部で進行中の転送は不正にされます。UBRR0L書き込みはボーレート前置分周器の更新を直ちに始めます。

17. USARTSPI – USARTでのSPI動作 (USART in SPI Mode)

17.1. 概要

USART(Universal Synchronous and Asynchronous Receiver and Transmitter)は主装置SPI互換動作に設定できます。USART動作選択(UMSELn1,0)ビットの11設定は主装置SPI(MSPIM)論理回路でのUSARTを許可します。この動作種別でのSPI主装置制御論理回路はUSART資源を直接制御します。これらの資源には送受信の移動レジスタと緩衝部、ボーレート発生器を含みます。パリティ発生/検査論理回路、データ/クロック再生論理回路、送受信制御論理回路は禁止されます。USART送受信論理回路は普通のSPI転送制御論理回路に置き換えられます。けれどもピン制御論理回路と割り込み生成論理回路は両動作種別で全く同じです。I/Oレジスタ位置は両動作種別で同じです。けれどもMSPIM使用時、制御レジスタの一部の機能が変わります。

17.2. 特徴

- ・ 全二重動作、3線同期データ転送
- ・ 主装置動作
- ・ 4つ全てのSPI転送形式(動作種別0,1,2,3)支援
- ・ LSBまたはMSB先行データ転送(データ順設定)
- ・ 順列動作(2重緩衝)
- ・ 高分解能ボーレート発生器
- ・ 高速動作(fXCKmax=fCK/2)
- ・ 柔軟な割り込み生成

17.3. クロック生成

クロック生成論理回路は送受信部に対する基準クロックを生成します。USARTのMSPIM動作種別については内部クロック生成(即ち主装置動作)だけが支援されます。従って、USARTでMSPIMを正しく動作するにはXCKnピンに対するデータ方向レジスタ(DDR_XCKn)が1(即ち出力)に設定されなければなりません。なるべくならDDR_XCKnはUSARTでのMSPIMが許可(換言するとTXENnとRXENnが1に設定)される前に設定されるべきです。

MSPIM動作で使われる内部クロック生成はUSART同期主装置動作と同一です。右表は同期主装置動作用のボーレートやUBRRn設定を計算する式を含みます。

表17-1. ボーレート レジスタ(UBRRn)値計算式

| 動作種別 | ボーレート計算式 | UBRRn値計算式 |
|---------|---|---|
| 同期主装置動作 | $BAUD = \frac{f_{OSC}}{2 \times (UBRRn + 1)}$ | $UBRRn = \frac{f_{OSC}}{2 \times BAUD} - 1$ |

注: ボーレートは転送速度(ビット/1秒)で定義されます。

BAUD : ボーレート (bps)
UBRRn : UBRRnHとUBRRnLレジスタ値 (0~4095)
fOSC : システム発振器クロック周波数

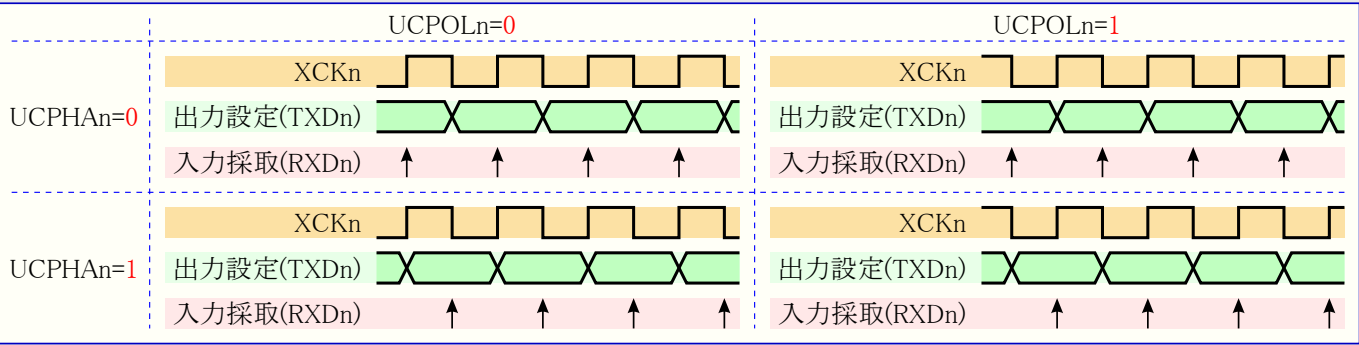
17.4. SPIデータ形態とタイミング

直列データに関してはクロック位相(UCPHAn)とクロック極性(UCPOLn)制御ビットによって決定されるXCKn(SCK)位相と極性で4つの組み合わせがあります。このデータ転送タイミング図は下図で示されます。データビットは安定のためにデータ信号に対して十分な時間を保証するXCKn信号の反対端で移動出力と(入力)ラッチが行われます。UCPOLnとUCPHAnの機能は下表で要約されます。これらビットのどの変更も送受信両方に対して実行中の通信を不正とすることに注意してください。

表17-2. UCPOL,UCPHA機能動作

| SPI動作種別番号 | UCPOLn | UCPHAn | XCKn(SCK)先行端 | XCKn(SCK)後行端 |
|-----------|--------|--------|--------------|--------------|
| 0 | 0 | 0 | 入力採取/上昇端 | 出力設定/下降端 |
| 1 | 0 | 1 | 出力設定/上昇端 | 入力採取/下降端 |
| 2 | 1 | 0 | 入力採取/下降端 | 出力設定/上昇端 |
| 3 | 1 | 1 | 出力設定/下降端 | 入力採取/上昇端 |

図17-1. UCPHAnとUCPOLnによるデータ転送タイミング



17.5. フレーム形式

MSPIMの直列フレームは8データビット/1フレームに固定されます。USARTでのMSPIM動作は2つの有効なフレーム形式を持ちます。

- MSB先行 8ビット データ
- LSB先行 8ビット データ

フレームは最下位(LSB)または最上位(MSB)のデータビットで始まります。その後次のデータビットが最後の最上位(MSB)または最下位(LSB)データビットまで合計8ビット続きます。完全なフレームが送信されると、新規フレームが直ちに後続するか、または通信線がアイドル状態(**high**)に設定されるでしょう。

USART制御/状態レジスタC(UCSRnC)のデータ順選択(UDORDn)ビットはUSARTでのMSPIMによって使われるフレーム形式を設定します。送受信部は同じ設定を使います。これらビットのどれかの設定変更が送受信部両方に対して実行中の通信を不正にすることに注意してください。

16ビット データ通信はUDRnに2バイト データを書くことによって達せられます。その後のUSART送信完了割り込み(TXCn)はこの16ビット値が移動出力されてしまったことを示します。

17.5.1. USART MSPIM初期化

USARTでのMSPIM動作はどれかの通信が行われ得るのに先立って初期化されなければなりません。標準的な初期化手順は使用方法に依存するボーレート設定、主装置動作操作(DDR_XCKnの1)設定、フレーム形式設定、送受信部許可から成ります。送信部だけが独立して操作できます。割り込み駆動USART操作に関して初期化を行う時にステータス レジスタの全割り込み許可(I)ビットが解除(0)される(そして全割り込みが禁止される)べきです。

注: XCKn出力の初期化を直ちに保証するため、ボーレート レジスタ(UBRRn)は送信部が許可される時に0でなければなりません。標準動作のUSART操作と逆に、UBRRnは送信部が許可されて最初の送信が開始される前に、希望する値を書かれなければなりません。この初期化がリセット後直ちに行なわれるなら、UBRRnが0にリセットされているので、送信部許可前にUBRRnを0に設定することは必要ありません。

ボーレート、データ転送形式またはフレーム形式の変更を伴う再初期化を行う前に、レジスタが変更される期間中に進行中の送信がないことを確実にしてください。USART制御/状態レジスタA(UCSRnA)の送信完了(TXCn)フラグは送信部の全転送完了検査に使え、受信完了(RXCn)フラグは受信緩衝部内の未読データ有無検査に使えます。この目的(次送信タイミング)にTXCnフラグが使われる場合、各々の送信(USARTデータ レジスタ(UDRn)が書かれる)前にTXCnフラグが解除(0)されなければならないことに注意してください。

次の簡単なUSART初期化コード例が示すアセンブリ言語とC言語の関数は機能的に同じです。この例はポーリングを使う(割り込み不許可)と仮定します。ボーレート(UBRRn)値は関数の引数として与えられます。アセンブリ言語でのボーレート引数はR17:R16レジスタに格納されると仮定されます。

アセンブリ言語プログラム例

```
USART_Init: CLR      R18                      ;0値取得
             OUT      UBRRnH, R18             ;ボーレート設定(上位バイト)=0
             OUT      UBRRnL, R18             ;ボーレート設定(下位バイト)=0 (クロック停止)
             SBI      XCKn_DDR, XCKn          ;XCKnポートピン出力設定
             LDI      R18, (1<<UMSELn1) | (1<<UMSELn0) | (0<<UCPHAn) | (0<<UCPOLn)
             ;動作種別値を取得
             OUT      UCSRnC, R18             ;MSPI,データ種別0設定
             LDI      R18, (1<<RXENn) | (1<<TXENn)
             ;送受信許可値を取得
             OUT      UCSRnB, R18             ;送受信許可
             OUT      UBRRnH, R17             ;ボーレート設定(上位バイト)
             OUT      UBRRnL, R16             ;ボーレート設定(下位バイト)
             RET                                ;呼び出し元へ復帰
```

C言語プログラム例

```
void USART_Init(unsigned int baud)
{
    UBRRn = 0;                                /* ボーレート設定(クロック停止) */
    XCKn_DDR |= (1<<XCKn);                    /* XCKnポートピン出力設定 */
    UCSRnC = (1<<UMSELn1) | (1<<UMSELn0) | (0<<UCPHAn) | (0<<UCPOLn);
    ; MSPI,データ種別0設定 */
    UCSRnB = (1<<RXENn) | (1<<TXENn);         /* 送受信許可 */
    UBRRn = baud;                             /* ボーレート設定 */
}
```


17.6. データ転送

USARTでのMSPI動作の使用は送信部が許可されること、換言するとUSART制御/状態レジスタB(UCSRnB)で送信許可(TXENn)ビットが1に設定される必要があります。送信部が許可されると、TXDnピンの標準ピン動作は無視され、送信部の直列出力としての機能を与えられます。受信部の許可は任意選択で、UCSRnBの受信許可(RXENn)ビットの設定(=1)によって行なわれます。受信部が許可されると、RXDnピンの標準ピン動作は無視され、受信部の直列入力としての機能を与えられます。XCKnは両方の場合で転送クロックとして使われます。

初期化後、USARTはデータ転送を行なう準備が整います。データ転送はUSARTデータレジスタ(UDRn)I/O位置に書くことによって開始されます。送信部が転送クロックを制御するため、これは送受信データ両方の状態についてです。UDRnに書かれたデータは移動レジスタが新規フレームを送る準備が整った時に送信緩衝部から移動レジスタへ移されます。

注: 入力緩衝部に於いて送信されたデータバイト数との同期を保つために、送信された各バイトに対して一度、UDRnが読まれなければなりません。入力緩衝操作は通常のUSART動作と同じで、換言すると、オーバーランが起きると、緩衝部内の先頭ではなく最後に受信したデータが失われます。これは第1バイト、第2、第3、第4バイトの順で4バイトのデータが送信され、全ての転送が完了される前にUDRnが読めない場合、第1バイトではなく、受信されるべき第3バイトが失われます。

次のコード例はUSART制御/状態レジスタA(UCSRnA)の送信データレジスタ空き(UDREN)フラグのポーリングに基いた、簡単なUSARTでのMSPI転送関数を示します。この関数が使われ得る前にUSARTが初期化されなければなりません。アセンブリ言語での送るべきデータはR16レジスタに格納されると仮定され、受信したデータは関数復帰後、同じR16レジスタで利用できます。

この関数は送信されるべき新規データを設定する前に、UDRENの検査によって送信緩衝部が空になるのを単純に待ちます。その後、緩衝部を読んで値を戻す前に、RXCnフラグの検査によって受信緩衝部にデータが存在するのを待ちます。

アセンブリ言語プログラム例

```
USART_MSPI: SBIS    UCSRnA, UDREN    ;送信緩衝部空きでスキップ
             RJMP    USART_MSPI     ;送信緩衝部空き待機
;
             OUT     UDRn, R16       ;データ送信(送信開始)
USART_SPIR: SBIS    UCSRnA, RXCn     ;受信完了でスキップ
             RJMP    USART_SPIR     ;受信完了待機
;
             IN      R16, UDRn       ;受信データ取得
             RET                     ;呼び出し元へ復帰
```

C言語プログラム例

```
void USART_MSPI_Transfer(unsigned int data)
{
    while ( !(UCSRnA & (1<<UDREN)) ); /* 送信緩衝部空き待機 */
    UDRn = data;                       /* データ送信(送信開始) */
    while ( !(UCSRnA & (1<<RXCn)) );  /* 受信完了待機 */
    return UDRn;                       /* 受信データ取得 */
}
```

17.6.1. 送受信フラグと割り込み

USARTのMSPI動作での受信完了(RXCn)、送信完了(TXCn)、送信データレジスタ空き(UDREN)フラグと対応する割り込みは通常のUSART操作と機能的に同一です。けれども受信異常状態フラグ(FE0,DOR0,UPE0)は使えず、常に0として読まれます。

17.6.2. 送受信の禁止

USARTのMSPI動作での送受信部の禁止は通常のUSART操作と機能的に同一です。

17.7. USARTでのMSPIMとSPIの比較

USARTでのMSPI動作は次に関してSPIと完全な互換性があります。

- 主装置動作タイミング図
- クロック極性選択(UCPOLn)ビットはSPIのSCK極性選択(CPOL)ビットと機能的に同じです。
- クロック位相選択(UCPHAn)ビットはSPIのSCK位相選択(CPHA)ビットと機能的に同じです。
- データ順選択(UDORDn)ビットはSPIのデータ順選択(DORD)ビットと機能的に同じです。

けれどもUSARTでのMSPI動作がUSART資源を再使用するため、USARTでのMSPI動作はSPIと比較して多少異なります。加えて制御レジスタビットの差異、主装置動作だけがUSARTでのMSPI動作によって支援されること、2つの部間で異なる次の特質があります。

- USARTでのMSPI動作は送信部の(2重)緩衝部を含みます。SPIは緩衝部を持ちません。
- USARTのMSPI動作での受信部は追加の緩衝段を含みます。
- SPIの上書き(WCOL)ビットはUSARTでのMSPI動作に含まれません。
- SPIの倍速許可(SPI2X)ビットは含まれません。しかし、対応するボーレートレジスタ(UBRRn)設定によって同じ効果が達せられます。
- 割り込みタイミングに互換性はありません。
- USARTでのMSPI動作が主装置動作だけのため、ピン制御が異なります。

USARTのMSPI動作とSPIでのピンは右表で示されます。

表17-3. USARTでのMSPIMとSPIのピン比較

| USART MSPIM | SPI | 備考 |
|-------------|------|------------------|
| TXDn | MOSI | 主装置出力のみ |
| RXDn | MISO | 主装置入力のみ |
| XCKn | SCK | (機能的に同一) |
| 該当なし | SS | USARTでのMSPIMで未支援 |

17.8. MSPIMでのUSART用レジスタ

USART用レジスタ記述を参照してください。

18. AC – アナログ比較器

18.1. 概要

アナログ比較器は非反転入力AIN0(PA0)ピンと反転入力AIN1(PA1)ピンの入力値を比較します。非反転AIN0ピンの電圧が反転AIN1ピンの電圧よりも高い時にACSRAのアナログ比較器出力(ACO)ビットが設定(1)されます。

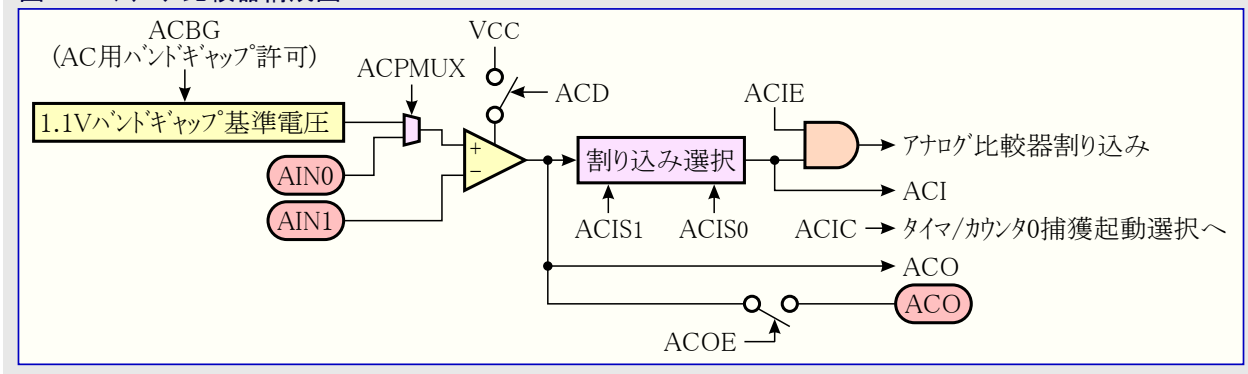
18.2. 特徴

- 柔軟な入力選択
 - 内部基準電圧
 - 非反転または反転の入力に対する2つのピン選択が可能 (訳注:他に該当記述なし)
- 以下での割り込み生成
 - 出力切り替わり
 - 出力下降端
 - 出力上昇端

18.3. 構成図

アナログ比較器(AC)の非反転入力には1つの内部基準(1.1V、バンドギャップ)電圧だけが接続されます。ACへの非反転入力としてバンドギャップ基準電圧の使用に関して、最初にアナログ比較器制御/状態レジスタAの基準電圧許可(ACSRA.ACBG)に'1'を書くことによって許可され、その後にアナログ比較器制御/状態レジスタBのAC非反転入力多重器(ACSRB.ACPMMUX)に'1'を書くことによって選択することが賢明です。比較器出力の出力はタイマ/カウンタ0の捕獲機能を起動するように設定できます。加えて、比較器はアナログ比較器専用の独立した割り込みを起動することができます。使用者は比較器出力の上昇端、下降端、またはその両方での割り込み起動を選ぶことができます。この比較器とその周辺論理回路の構成図は下で示されます。

図18-1. アナログ比較器構成図



注: アナログ比較器ピン配置については「ピン配置」と入出力ポート記述を参照してください。

関連リンク [7頁の「ピン配置」](#)

18.4. アナログ比較器用レジスタ

18.4.1. ACSRA – アナログ比較器 制御/状態レジスタA (Analog Comparator Control and Status Register A)

名称 : ACSRA

変位 : \$1F

リセット : '00x00000'

特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|-----|------|-----|-----|------|------|---------|-----|
| | ACD | ACBG | ACO | ACI | ACIE | ACIC | ACIS1,0 | |
| アクセス種別 | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W |
| リセット値 | 0 | 0 | 不定 | 0 | 0 | 0 | 0 | 0 |

● ビット7 – ACD : アナログ比較器禁止 (Analog Comparator Disable)

このビットが論理1を書かれると、アナログ比較器への電力がOFFにされます。このビットはアナログ比較器をOFFにするために何時でも設定(1)できます。これは活動動作やアイドル動作で電力消費を削減します。ACDビットを変更する時にACSRAの[アナログ比較器割り込み許可\(ACIE\)ビット](#)を解除(0)することによってアナログ比較器割り込みが禁止されなければなりません。さもなければ、このビットが変更される時に割り込みが起こり得ます。

● ビット6 – ACBG : アナログ比較器バンドギャップ許可 (Analog Comparator Bandgap Enable)

このビットが設定(1)されると、[内部基準\(1.1Vバンドギャップ\)電圧](#)が許可されます。[アナログ比較器制御/状態レジスタB\(ACSRB\)のアナログ比較器非反転入力多重器\(ACPMUX\)ビット](#)も設定(1)されると、バンドギャップ基準電圧がアナログ比較器の非反転入力に印加されます。電圧の安定化を許すために、ACBGビットへの1書き込みによってバンドギャップ基準電圧が最初に許可され、その後にACPMUXビットへの1書き込みによって選択されることが推奨されます。

● ビット5 – ACO : アナログ比較器出力 (Analog Comparator Output)

アナログ比較器の出力は同期化され、その後に直接ACOへ接続されます。この同期化は1～2クロック周期の遅延をもたらします。

● ビット4 – ACI : アナログ比較器割り込み要求フラグ (Analog Comparator Interrupt Flag)

このビットは比較器出力での出来事がACSRAの[アナログ比較器割り込み条件\(ACIS1,0\)ビット](#)によって定義した割り込み方法で起動する時に設定(1)されます。ACSRAのアナログ比較器割り込み許可(ACIE)ビットが設定(1)され、[ステータスレジスタ\(SREG\)の全割り込み許可\(I\)ビット](#)が設定(1)されていると、アナログ比較器割り込みルーチンが実行されます。対応する割り込み処理ベクタを実行すると、ACIはハードウェアによって解除(0)されます。代わりにこのフラグへ論理1を書くことによってもACIは解除(0)されます。

● ビット3 – ACIE : アナログ比較器割り込み許可 (Analog Comparator Interrupt Enable)

ACIEビットが論理1を書かれ、ステータスレジスタ(SREG)の全割り込み許可(I)ビットが設定(1)されると、アナログ比較器割り込みが活性(有効)にされます。論理0を書かれると、この割り込みは禁止されます。

● ビット2 – ACIC : アナログ比較器捕獲起動許可 (Analog Comparator Input Capture Enable)

論理1を書かれると、このビットはアナログ比較器によって起動されるタイマ/カウンタ0の捕獲機能を許可します。この場合、比較器出力は比較器にタイマ/カウンタ0捕獲割り込みの[雑音消去機能](#)と[エッジ選択機能](#)を利用させる[捕獲入力前置論理回路](#)へ直接的に接続されます。論理0を書かれると、アナログ比較器と捕獲機能間の接続は存在しません。比較器がタイマ/カウンタ0捕獲割り込みを起動するには、[タイマ/カウンタ0割り込み許可レジスタ\(TIMASK0\)の捕獲割り込み許可\(ICIE0\)ビット](#)が設定(1)されなければなりません。

● ビット1,0 – ACIS1,0 : アナログ比較器割り込み条件 (Analog Comparator Interrupt Mode Select)

これらのビットは比較器のどの事象がアナログ比較器割り込みを起動するのかを決めます。

ACIS1,0ビットを変更する時にACSRAのアナログ比較器割り込み許可(ACIE)ビットを解除(0)することによってアナログ比較器割り込みが禁止されなければなりません。さもなければ、これらのビットが変更される時に割り込みが起き得ます。

表18-1. アナログ比較器割り込み条件選択

| ACIS1 | ACIS0 | 割り込み発生条件 |
|-------|-------|-----------------|
| 0 | 0 | 比較器出力の変移 (切り替り) |
| 0 | 1 | (予約) |
| 1 | 0 | 比較器出力の下降端 |
| 1 | 1 | 比較器出力の上昇端 |

18.4.2. ACSRБ – アナログ比較器 制御/状態レジスタB (Analog Comparator Control and Status Register B)

名称 : ACSRБ
変位 : \$1E
リセット : \$00
特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|---|---|---|---|---|---|------|--------|
| | - | - | - | - | - | - | ACOE | ACPMUX |
| アクセス種別 | R | R | R | R | R | R | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

●ビット1 – ACOE : アナログ比較器出力許可 (Analog Comparator Output Enable)

このビットが設定(1)されると、アナログ比較器出力がACOピンに接続されます。

●ビット0 – ACPMUX : アナログ比較器非反転入力多重器 (Analog Comparator Positive Input Multiplexer)

このビットが設定(1)されると、固定のバンドギャップ基準電圧がアナログ比較器への非反転入力に置き換わります。このビットが解除(0)されると、アナログ比較器への非反転入力にAIN0が印加されます。アナログ比較器の入力としてバンドギャップ基準電圧が使われる時に、安定させるためにその電圧に対して或る時間がかかります。安定にされない場合、最初の変換は不正値を与えるかもしれません。

18.4.3. DIDR0 – デジタル入力禁止レジスタ0 (Digital Input Disable Register 0)

各々のビットが論理1を書かれると、対応するADCピンのデジタル入力緩衝部が禁止されます。このビットが設定(1)される時、対応するポート入力(PIN)レジスタのビットは常に0として読みます。ADC7~0ピンにアナログ信号が印加され、そのピンからのデジタル入力が必要ない時に、デジタル入力緩衝部での電力消費を減らすため、このビットは論理1を書かれるべきです。

名称 : DIDR0
変位 : \$17
リセット : \$00
特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|
| | ADC7D | ADC6D | ADC5D | ADC4D | ADC3D | ADC2D | ADC1D | ADC0D |
| アクセス種別 | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

●ビット1,0 – ADC1D,ADC0D : AIN1,AIN0 デジタル入力禁止 (AIN1,AIN0 Digital Input Disable)

A/D変換器:

- ADC1DまたはADC0Dが1に設定されると、対応するADCピンのデジタル入力緩衝部が禁止されます。このビットが設定(1)される時、対応するポート入力(PIN)レジスタのビットは常に0として読みます。ADC1,0ピンにアナログ信号が印加され、そのピンからのデジタル入力が必要ない時に、デジタル入力緩衝部での電力消費を減らすため、このビットは論理1を書かれるべきです。

アナログ比較器:

- ADC1DまたはADC0Dが1に設定されると、AIN1(ADC1)/AIN0(ADC0)ピンのデジタル入力緩衝部が禁止されます。このビットが設定(1)されると、対応するポート入力レジスタのビット(PINx)は常に0として読みます。アナログ入力として使うけれどもデジタル入力を必要としない時にこのビットに論理1を書くことによってデジタル入力緩衝部での消費電力を削減することができます。
- ADC7D~ADC2D: これらのビットはアナログ比較器に適用しません。

19. ADC - A/D變換器

19.1. 概要

ATtiny102/ATtiny104は10ビット逐次比較A/D変換器(ADC)が特徴です。ADCはポートAとBのピン、内部基準電圧、供給電圧から構成された5/8つのシングルエンド電圧入力を許す、8/14ピンデバイスに対する4チャネルのアナログ多重器に接続されます。シングルエンド電圧入力は0V(GND)を基準にします。

19.2. 特徴

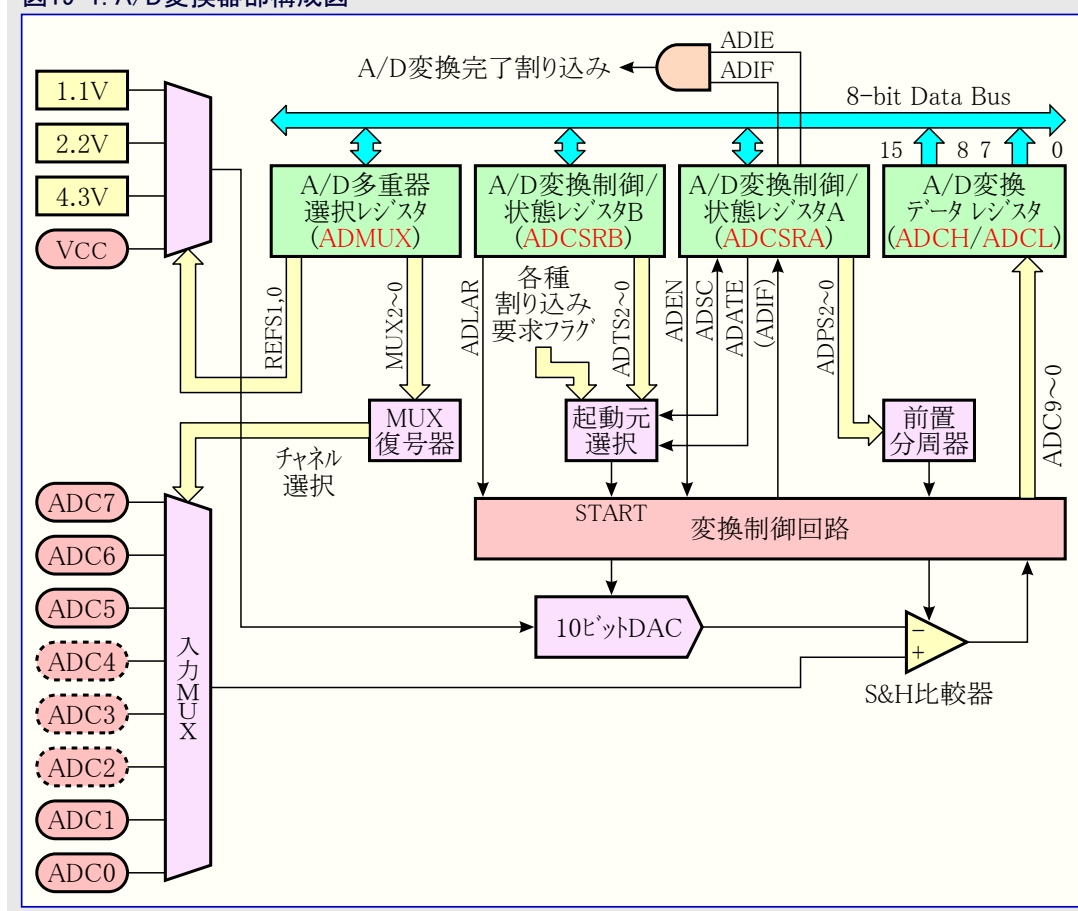
- ・10ビット分解能
- ・積分非直線性誤差1LSB
- ・絶対精度 ± 2 LSB
- ・変換時間15 μ s
- ・最大分解能で15kSPS(採取/s)
- ・8/14ピンでの5/8つの多重化されたシングル エント'入力チャネル
- ・A/D変換結果読み出しに対する任意の左揃え
- ・入力電圧範囲: 0 \sim VCC
- ・ADC基準電圧: 1.1V, 2.2V, 4.3V
- ・連続と単独の変換動作
- ・割り込み元の自動起動によるA/D変換開始
- ・A/D変換完了割り込み
- ・休止動作雑音低減機能

19.3. 構成図

A/D変換器はA/D変換器への入力電圧が変換中に一定の値で保持されることを保証する採取&保持(S/H)回路を含みます。

公称1.1V, 2.2V, 4.3Vの内蔵基準電圧がチップ上で提供されます。また、シングル エント' チャネルに対する基準電圧としてVCCを使うこともできます。

図19-1. A/D変換器部構成図



19.4. 操作

ADCを許可するために電力削減レジスタのA/D変換器電力削減(PRR.PRADC)ビットが'0'を書かれなければなりません。

A/D変換部はA/D変換制御/状態レジスタAのA/D許可(ADCSRA.ADEN)ビットに1を書くことによって(動作が)許可されます。基準電圧と入力チャネルの選択はADENが設定(1)されるまで実施しません。ADENが解除(0)されている時にA/D変換部は電力を消費しないので、節電をする休止形態へ移行する前にA/D変換部をOFFに切り替えることが推奨されます。

ADCは逐次比較を通してアナログ入力電圧を10ビット デジタル値に変換します。最小値はGNDを表し、最大値はVREF電圧を表します。ADC基準電圧はA/D多重器選択レジスタの基準電圧選択(REFS1,0)ビット書き込みによって選択されます。

アナログ入力チャネルはA/D変換多重器選択レジスタのチャネル選択(ADMUX.MUX)ビットを書くことによって選択されます。

A/D変換部はA/Dデータレジスタ(ADCH,ADCL)で示される10ビットの結果を生成します。既定では、結果が右揃えで表されますが、A/D変換制御/状態レジスタBの左揃え選択(ADCSRB.ADLAR)ビットを設定(1)することによって任意で左揃えで表せます。

結果が左揃え補正され、8ビットを越える精度が必要とされない場合はADCHを読むことだけで足ります。さもないければ、データレジスタの内容が同じ変換に属すること(からの結果)を保証するため、ADCLが初めに、次にADCHが読まれなければなりません。一度ADCLが読まれると、A/D変換器からのA/Dデータレジスタ(ADCH,ADCL)アクセスが阻止されます。これはADCLが読まれてしまってADCHが読まれる前に次の変換が完了すると、どちらのレジスタ(ADCH,ADCL)も更新されず、その変換からの結果が失われることを意味します。ADCHが読まれると、ADCH,ADCLへのA/D変換器アクセスが再び許可されます。

A/D変換部には変換完了時に起動できる自身の割り込みがあります。A/DデータレジスタへのA/D変換器アクセスがADCLとADCHの読み込み間で禁止されている場合、例えばその変換結果が失われても割り込みは起動します。

関連リンク 26頁の「PRR - 電力削減レジスタ」
103頁の「ADMUX - A/D変換多重器制御レジスタ」
103頁の「ADCSRA - A/D変換制御/状態レジスタA」
104頁の「ADCSRB - A/D変換制御/状態レジスタB」
105頁の「ADCH,ADCL - A/D変換データレジスタ上位/下位バイト [ADLAR=0]」
105頁の「ADCH,ADCL - A/D変換データレジスタ上位/下位バイト [ADLAR=1]」

19.5. 変換の開始

単独変換は電力削減レジスタのA/D変換器電力削減(PRR.PRADC)ビットに'0'を書いた状態で、A/D変換制御/状態レジスタAの変換開始(ADCSRA.ADSC)ビットに'1'を書くことによって開始されます。このビットは変換が進行中である限り、1に留まり、変換が完了されるとハードウェアによって解除(0)されます。変換が進行中に違う入力チャネルが選択されると、A/D変換部はそのチャネル変更を実行する前に現在の変換を済ませます。

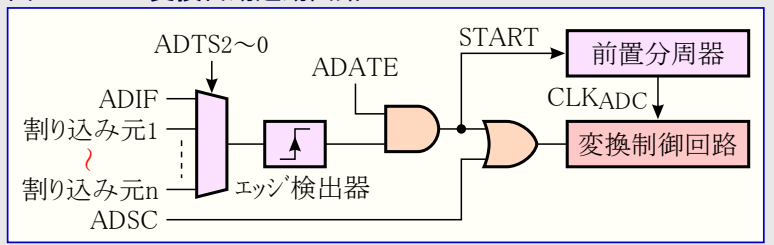
代わりに、変換は様々な起動元によって自動的に起動できます。自動起動はA/D変換自動起動許可(ADCSRA.ADATE)ビットの設定(1)によって許可されます。起動元はA/D変換制御/状態レジスタBのA/D変換起動元選択(ADCSRB.ADTS)ビットの設定によって選択されます。利用可能な起動元の一覧についてはADCSRB.ADTSの説明をご覧ください。選択した起動信号上に上昇端が起きると、A/D変換用前置分周器がリセットし、変換が開始されます。これは一定間隔で変換を開始する方法を提供します。変換完了時に起動信号が未だ設定(1)されている場合は新しい変換が開始されません。変換中にこの起動信号上で別の上昇端が起きると、そのエッジは無視されます。指定した割り込みが禁止またはステータスレジスタ(SREG)の全割り込み許可(I)ビットが解除(0)でも、割り込み要求フラグが設定(1)されることに注意してください。従って割り込みを起こさずに変換が起動できます。けれども次の割り込み要因で新規変換を起動するために、割り込み要求フラグは解除(0)されなければなりません。

起動元としてのA/D変換完了割り込み要求フラグ(ADIF)の使用は、実行中の変換が完了されると直ぐにA/D変換器に新規変換を開始させます。その時にA/D変換器は連続動作で動き、継続的な採取(変換)とA/Dデータレジスタを更新します。最初の変換はADCSRA.ADSCビットに'1'を書くことによって始めなければなりません。この動作でのA/D変換器はA/D変換完了割り込み要求フラグ(ADIF)が解除(0)されるかどうかに関らず、連続的な変換を実行します。

自動起動が許可されている場合は、ADCSRA.ADSCビットに'1'を書くことによって単独変換を開始できます。ADSCは変換が進行中かを判断するのに使えます。ADSCビットは変換がどう開始されたかに拘らず、変換中は'1'として読めます。

関連リンク 26頁の「PRR - 電力削減レジスタ」

図19-2. A/D変換自動起動回路



19.6. 前置分周と変換タイミング

既定での逐次比較回路は最大分解能を得るのに50～200kHzの入力クロック周波数を必要とします。

A/D変換部は100kHz以上のどんなCPUクロックからも受け入れ可能なA/D変換クロック周波数を生成する前置分周器を含みます。前置分周はA/D変換制御/状態レジスタAのA/Dクロック選択(ADCSRA.ADPS)ビットによって選択されます。前置分周器はA/D許可(ADCSRA.ADEN)ビットへの'1'書き込みによってA/D変換部がONにされた瞬間から計数を始めます。前置分周器はADEN=1である限り走行を維持し、ADEN=0の時は継続的にリセットします。

A/D変換開始(ADCSRA.ADSC)ビットの'1'書き込みによってシングルエントリ入力の変換を起動すると、その変換は直後の変換クロックの上昇端で始まります。

通常の変換は15変換クロック周期で行われます。下図で示されるように、A/D変換部がONされる(即ち、ADCSRA.ADENへの'1'書き込み)後の最初の変換はアナログ回路を初期化するために26変換クロック周期で行われます。

図19-3. A/D変換前置分周器部構成

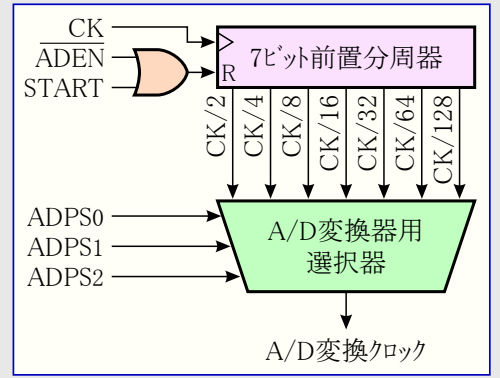
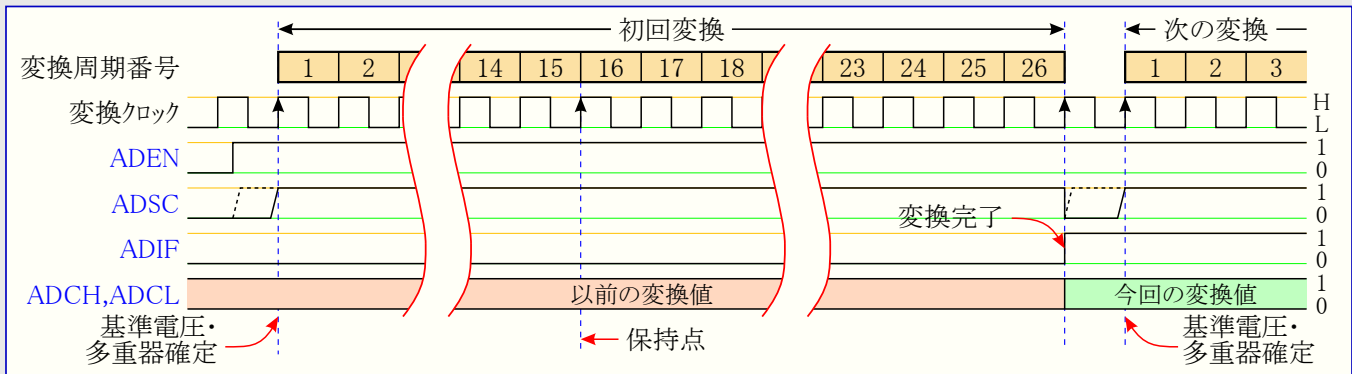
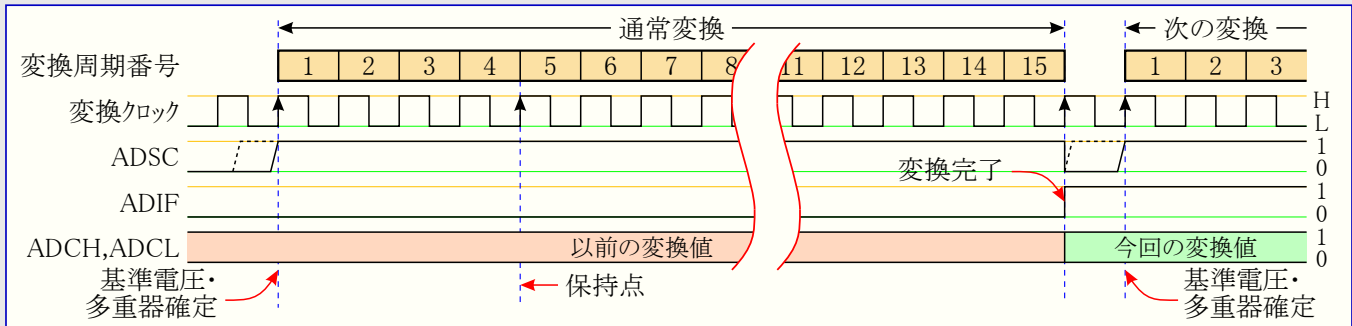


図19-4. 初回変換タイミング (単独変換動作)



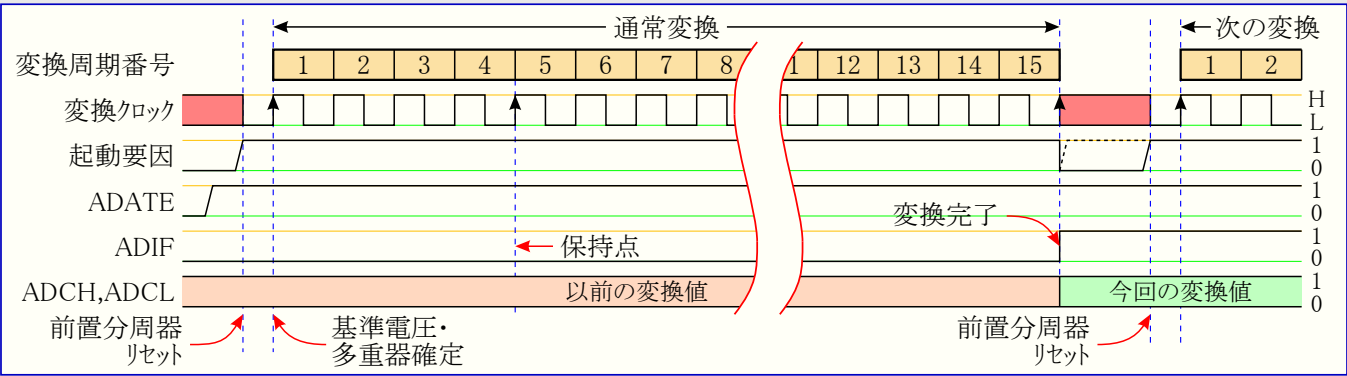
実際の採取&保持(保持開始点)は通常変換の開始後4変換クロック周期、初回変換の開始後15変換クロック周期で行われます。変換が完了すると、結果がA/Dデータレジスタ(ADCL)に書かれ、A/D変換完了割り込み要求フラグ(ADCSRA.ADIF)が設定(1)されます。単独変換動作(ADATE=0)では同時にA/D変換開始(ADCSRA.ADSC)ビットが解除(0)されます。その後にソフトウェアは再びADCSRA.ADSCを設定(1)でき、新規変換は変換クロックの最初の上昇端で開始されます。

図19-5. 通常変換タイミング (単独変換動作)



自動起動が使われると、次図で示されるように前置分周器は起動要因発生時にリセットされます。これは起動要因から変換開始までの一定の遅延を保証します。この動作での採取&保持は起動要因となる信号の上昇後、4.5変換クロック周期で採取が行われます。同期化論理回路(エッジ検出器)に対して追加の2CPUクロック周期が費やされます。

図19-6. 通常変換タイミング* (自動起動変換動作)



連続変換動作(ADATE=1)では変換完了後直ちに新規変換が開始され、一方ADCSRA.ADSCは1に留まります。下のA/D変換時間の表をご覧ください。

図19-7. 連続変換動作タイミング*

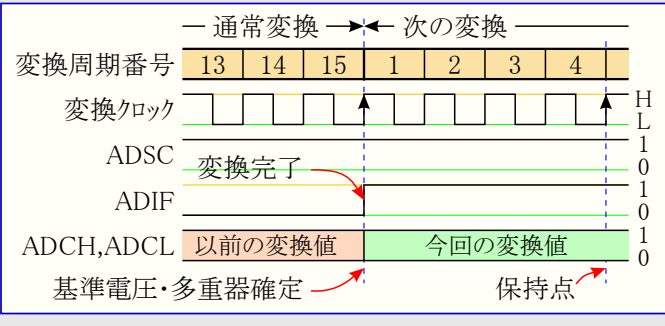


表19-1. A/D変換時間 (周期数)

| 変換種別 | 保持点 | 変換時間 |
|----------|-----|------|
| 初回変換 (注) | 15 | 26 |
| 通常変換 | 4 | 15 |
| 自動起動変換 | 4.5 | 15.5 |
| 連続変換 | 4 | 15 |

注: チャンネルや基準電圧の変更後の最初の変換も含まれます。

19.7. チャンネル変更と基準電圧

A/D多重器選択レジスタのチャンネル選択(ADMUX.MUX)ビットはCPUが順番にアクセスするための一時レジスタを通して単独緩衝されます。これはチャンネル選択が変換中の安全なところでだけ行うのを保証します。チャンネル選択は変換が開始されるまで継続的に更新されます。一旦変換が始まると、A/D変換器に対して十分な採取/変換時間を保証するためにチャンネル選択は固定されます。継続的な更新は(ADCSRA.ADIFの設定(1)によって示される)変換完了前の最後の変換クロック周期で再開します。変換開始(ADCSRA.ADSC)ビットが書かれた後の次の変換クロックの上昇端で変換が始まることに注意してください。従って使用者は変換開始(ADCSRA.ADSC)書き込み後、1変換クロック周期(経過)まで新しいチャンネル選択値をADMUXに書かないことが推奨されます。

自動起動が使われる場合、起動要因の正確な時間は確定できません。変換が新規設定によって影響されるように制御するにはADMUXの更新時に特別な注意が払われなければなりません。

A/D許可(ADCSRA.ADEN)とA/D変換自動起動許可(ADCSRA.ADATE)の両方が'1'を書かれると、何時でも割り込みが起き得ます。この期間でADMUXが変更されると、使用者は次の変換が旧設定または新設定どちらが基準にされるかを知ることができません。ADMUXは次の方法で安全に更新できます。

- ADENまたはADATEが解除(0)されている時。
- 変換開始後、最低1変換クロック周期経過後の変換中。
- 変換後から、変換起動元として使った割り込みフラグが解除(0)される直前まで。

これら条件の1つでADMUXを更新すると、新設定は次のA/D変換に影響を及ぼします。

19.7.1. A/D入力チャネル

チャンネル選択を変更する時に使用者は正しいチャンネルが選択されることを保証するために次の指針を守るべきです。

- 単独変換動作では常に変換を始める前にチャンネルを選択してください。チャンネル選択はADSCへの1書き込み後、1変換クロック周期で変更されるかもしれません。とは言え、最も簡単な方法はチャンネル選択を変更する前に変換が完了するまで待つことです。
- 連続変換動作では常に最初の変換を始める前にチャンネルを選択してください。チャンネル選択はADSCへの1書き込み後、1変換クロック周期で変更されるかもしれません。とは言え、最も簡単な方法は最初の変換が完了するまで待ち、その後にチャンネル選択を変更することです。既に次の変換が自動的に開始されているので、次の結果は直前のチャンネル選択を反映します。それに続く変換は新しいチャンネル選択を反映します。使用者は連続変換動作中に新しいチャンネルを書かないことを推奨されます。

19.7.2. A/D変換基準電圧

ADCの基準電圧(V_{REF})はA/D変換に対する変換範囲を示し、この場合は0V(V_{GND})と $V_{REF}=V_{CC}$ に制限されます。 V_{REF} を越えるシングルエンドチャンネルは\$3FFで飽和する符号に帰着します。

V_{REF} は V_{CC} または内部基準電圧から選ぶことができます。内部基準電圧は1.1V, 2.2V, 4.3Vに設定することができ、内部増幅器を通して内部バンドギャップ基準電圧(V_{BG})から生成されます。基準電圧元切り替え後の最初のA/D変換結果は不正確になるかもし、使用者はこの結果を破棄することが推奨されます。

19.8. 雑音低減機能

ADCはCPUコアと他の周辺I/Oが誘導した雑音を減らすために**休止形態動作**中の変換を可能にする雑音低減機能が特徴です。この機能は**A/D変換雑音低減動作**と**アイドル動作**で使うことができます。この機能を使うには次の手順が使われるべきです。

1. ADCが許可($ADEN=1$)され、変換中でない($ADSC=0$)ことを確認してください。単独変換動作が選択($ADATE=0$)され、且つA/D変換完了割り込みが許可($ADIE=1$)されていなければなりません。
2. A/D変換雑音低減(またはアイドル)動作に移行してください。一旦CPUが停止されてしまうと、ADCは変換を始めます。
3. A/D変換完了前に他の割り込みが起これなければ、A/D変換完了割り込みはCPUを起動してA/D変換完了割り込みルーチンを実行します。A/D変換完了前に他の割り込みがCPUを起動すると、その割り込みが実行され、A/D変換完了割り込み要求はA/D変換完了時に生成されます。CPUは新規**SLEEP**命令が実行されるまで通常動作に留まります。

注: アイドル動作とA/D変換雑音低減動作を除く他の休止動作へ移行する時にADCが自動的にOFFへ切り替えられないことに注意してください。使用者は余分な消費電力を避けるため、このような休止形態動作へ移行する前に $ADCSRA.ADEN \rightarrow 0$ を書くことが推奨されます。

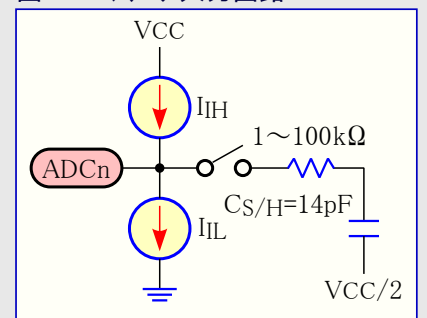
19.9. アナログ入力回路

シングルエンド入力チャンネルのアナログ回路は下で図解されます。 $ADCn$ に印加したアナログ(信号)源はそのチャンネルがADC入力として選択されているかどうかにかかわらず、ピン容量とそのピンの漏れ電流に左右されます。そのチャンネルが選択されると、(アナログ信号)源は直列抵抗(入力経路の合成抵抗)を通してS/Hコンデンサを駆動しなければなりません。

ADCは概ね10k Ω 若しくはそれ以下の出力インピーダンスのアナログ信号用に最適化されています。このような(アナログ信号)源が使われるなら、採取時間は無視してもよいでしょう。より高いインピーダンスの(アナログ信号)源が使われる場合、採取時間は広範囲に変化し得るS/Hコンデンサを充電するために(アナログ信号)源がどれくらいの時間を必要とするかに依存します。緩やかに変化する信号で、必要とされるS/Hコンデンサへの充放電を最小とすため、使用者は低インピーダンス信号源だけを使うことが推奨されます。

特定できない信号の渦からの歪を避けるため、どのチャンネルに対してもナイキスト周波数($f_{ADC}/2$)よりも高い信号成分が存在すべきではありません。使用者はADC入力として信号を印加する前に、低域通過濾波器で高い周波数成分を取り除くことが推奨されます。

図19-8. アナログ入力回路



19.10. アナログ雑音低減技術

デバイス内外のデジタル回路がアナログ測定の精度に影響を及ぼすかもしれないEMIを発生します。変換精度が重要な場合は次の技法を適用することによって雑音水準を低減できます。

- ・アナログ信号経路を可能な限り最短にしてください。
- ・アナログ信号経路がアナログGND面上を走ることを確認してください。
- ・アナログ信号経路を高速切り替えデジタル信号線から充分分離することを守ってください。
- ・何れかのADCポートピンがデジタル出力として使われる場合、これらを変換進行中に決して切り替えてはなりません。
- ・可能な限りVCCとGNDピンの近くにパスコンを配置してください。

高いA/D変換精度が必要とされるとき、「**雑音低減機能**」で記述されるようにA/D変換雑音低減動作の使用が推奨されます。正しい外部パスコン配置での良いシステム設計はA/D変換雑音低減動作の必要性を低減します。

19.11. A/D変換の精度定義

シングル エント 入力電圧のnビットA/D変換はGNDとVREF間を 2^n で直線的に変換します。最低値コードは0として読み、最高値コードは 2^n-1 として読みます。

以下のように各項目は理想状態からの偏差を表します。

• オフセット誤差 - 図19-9.

最初の遷移点(\$00から\$01)に於いて理想遷移点(差0.5LSB)と比較した偏差です。理想値は0LSBです。

• 利得誤差 - 図19-10.

オフセット誤差補正後の最後の遷移点(\$FEから\$FF)に於いて理想遷移点(最大差1.5LSB以下)と比較した偏差です。理想値は0LSBです。

• 積分非直線性誤差 (INL) - 図19-11.

オフセット誤差と利得誤差補正後の全ての遷移点に於いて理想遷移点と比較した最大偏差です。理想値は0LSBです。

• 微分非直線性誤差 (DNL) - 図19-12.

実際のコードの幅(隣接する2つの遷移点間)に於いて理想コード幅(1LSB)と比較した最大偏差です。理想値は0LSBです。

• 量子化誤差

有限数のコードで入力電圧を量子化するため、1LSB幅となる入力電圧範囲は同じ値のコードになります。この値は常に ± 0.5 LSBです。

• 絶対精度

補正しない全ての遷移点に於いて理想遷移点と比較した最大偏差です。これは、オフセット誤差、利得誤差、差動誤差、非直線誤差の影響の合成です。理想値は ± 0.5 LSBです。

図19-9. オフセット誤差

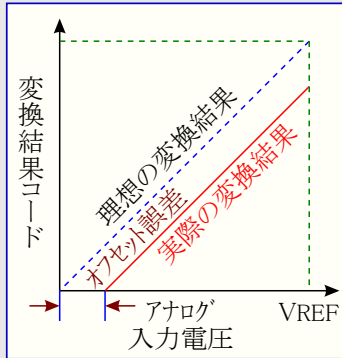


図19-10. 利得誤差

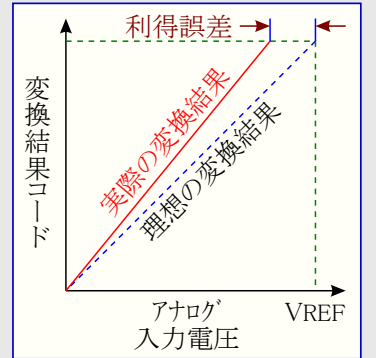


図19-11. 積分非直線性誤差

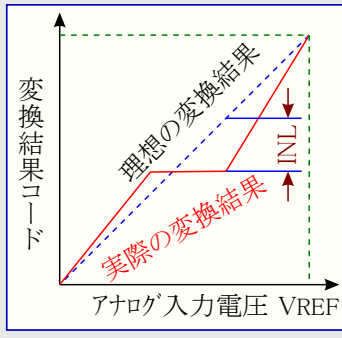
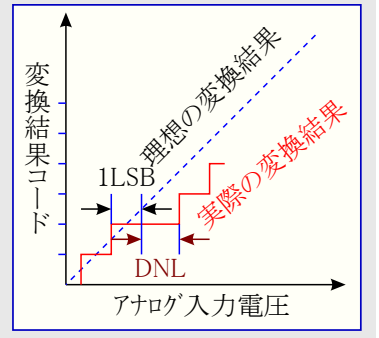


図19-12. 微分非直線性誤差



19.12. A/D変換の結果

変換完了(ADCSRA.ADIF=1)後、変換結果はA/Dデータレジスタ(ADCH,ADCL)で得られます。

次の変換結果による更新を避けてデータレジスタを施錠するために最初にADCLが読まれなければなりません。変換結果の形式は変換の形式に依存します。

19.12.1. シングル エント 変換

シングル エント 入力変換に対する結果は次のとおりです。

$$ADC = \frac{V_{IN} \times 1024}{V_{REF}}$$

ここでの V_{IN} は選択した入力ピンの電圧で、 V_{REF} は選択した基準電圧です(ADMUX.MUXの記述もご覧ください)。 $\$000$ はアナログGNDを表し、 $\$3FF$ は選択した基準電圧-1LSBを表します。結果は $\$000$ から $\$3FF$ までの一方向形式で表されます。

注: ADCLが読まれると、A/DデータレジスタはADCHが読まれるまで更新されません。その結果として、結果がが左揃えにされ、8ビットよりも高い精度が必要とされない場合、ADCHを読むことで充分です。さもなければ、先にADCLが読まれ、その後にADCHが読まれなければなりません。

19.13. A/D変換用レジスタ

19.13.1. ADMUX – A/D多重器選択レジスタ (ADC Multiplexer Select Register)

名称 : ADMUX

変位 : \$1B

リセット : \$00

特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|---------|-----|---|---|---|--------|-----|-----|
| | REFS1,0 | | - | - | - | MUX2~0 | | |
| アクセス種別 | R/W | R/W | R | R | R | R/W | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

● ビット7,6 – REFS1,0 : 基準電圧選択 (Reference Select)

これらのビットはA/D変換器の基準電圧を選びます。これらのビットが変換中に変更されると、その変更は変換が完了する(ADCSRAのADIFが設定(1)される)まで実施しません。

表19-2. ADC基準電圧選択

| REFS1,0 | 基準電圧選択 |
|---------|------------|
| 0 0 | VCC |
| 0 1 | 内部1.1V基準電圧 |
| 1 0 | 内部2.2V基準電圧 |
| 1 1 | 内部4.2V基準電圧 |

● ビット2~0 – MUX2~0 : A/Dチャネル選択 (Analog Channel Selection)

これらのビットの値はA/D変換器にどのアナログ入力に接続されるかを選びます。これらのビットが変換中に変更される場合、その変更は変換が完了する(ADCSRA.ADIFが設定(1)される)まで実施しません。

表19-3. アナログ入力チャネル選択

| MUX2~0 | 0 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1 0 0 | 1 0 1 | 1 1 0 | 1 1 1 |
|------------|-------|-------|--------|--------|--------|-------|-------|-------|
| アナログ入力チャネル | ADC0 | ADC1 | (ADC2) | (ADC3) | (ADC4) | ADC5 | ADC6 | ADC7 |
| ポートピン名 | PA0 | PA1 | PA5 | PA6 | PB0 | PB1 | PB2 | PB3 |

19.13.2. ADCSRA – A/D変換 制御/状態レジスタA (ADC Control and Status Register A)

名称 : ADCSRA

変位 : \$1D

リセット : \$00

特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|------|------|-------|------|------|---------|-----|-----|
| | ADEN | ADSC | ADATE | ADIF | ADIE | ADPS2~0 | | |
| アクセス種別 | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

● ビット7 – ADEN : A/D許可 (ADC Enable)

このビットに1を書くことがADC(動作)を許可します。0を書くことによってADCは(電源が)OFFされます。変換が進行中にADCをOFFにすることはその変換を(途中)終了します。

● ビット6 – ADSC : A/D変換開始 (ADC Start Conversion)

単独変換動作で各変換を始めるにはこのビットへ1を書いてください。連続変換動作で最初の変換を始めるにはこのビットへ1を書いてください。ADCが許可される(ADEN=1)と同時にADSCが書かれるか、またはADCが許可されてしまった後にADSCが書かれた後の初回変換は通常の15に代わって26変換クロック周期で行います。この初回変換はADCの初期化を実行します。

ADSCは変換が進行中である限り1として読めます。変換が完了すると0に戻ります。このビットへの0書き込みは無効です。

● ビット5 – ADATE : A/D変換自動起動許可 (ADC Auto Trigger Enable)

このビットが1を書かれると、A/D変換の自動起動が許可されます。ADCは選択した起動信号の上昇端で変換を開始します。この起動元はA/D変換制御/状態レジスタB(ADCSRB)のA/D変換起動要因選択(ADTS2~0)ビット設定によって選択されます。

●ビット4 – ADIF : A/D変換完了割り込み要求フラグ (ADC Interrupt Flag)

A/D変換が完了し、A/Dデータレジスタが更新されると、このフラグが設定(1)されます。A/D変換完了割り込み許可(ADIE)ビットとステータスレジスタ(SREG)の全割り込み許可(I)ビットが設定(1)されたなら、A/D変換完了割り込みが実行されます。対応する割り込み処理ベクタを実行する時にADIFはハードウェアによって解除(0)されます。代わりにこのフラグに論理1を書くことによってもADIFは解除(0)されます。ADCSRで読み-変更-書き(リード モディファイ ライト)を行うと、保留中の割り込みが禁止され得ることに注意してください。これはSBI,CBI命令が使われる場合にも適用されます。

●ビット3 – ADIE : A/D変換完了割り込み許可 (ADC Interrupt Enable)

このビットが1を書かれ、ステータスレジスタ(SREG)の全割り込み許可(I)ビットが設定(1)されると、A/D変換完了割り込みが活性になります。

●ビット2~0 – ADPS2~0 : A/D変換クロック選択 (ADC Prescaler Select Bits)

これらのビットはシステムクロック周波数とA/D変換部への入力クロック間の分周値を決めます。

| 表19-4. A/D変換クロック選択 (CK=システムクロック) | | | | | | | | |
|----------------------------------|-------|-------|-------|-------|-------|-------|-------|--------|
| ADPS2~0 | 0 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1 0 0 | 1 0 1 | 1 1 0 | 1 1 1 |
| A/D変換クロック | CK/2 | CK/2 | CK/4 | CK/8 | CK/16 | CK/32 | CK/64 | CK/128 |

19.13.3. ADCSRB – A/D変換 制御/状態レジスタB (ADC Control and Status Register B)

名称 : ADCSRB

変位 : \$1C

リセット : \$00

特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|-------|---|---|---|---|---------|-----|-----|
| | ADLAR | - | - | - | - | ADTS2~0 | | |
| アクセス種別 | R | R | R | R | R | R/W | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

●ビット5 – ADLAR : 左揃え選択 (ADC Left Adjust Result)

ADLARビットはA/Dデータレジスタ内の変換結果の配置に影響を及ぼします。結果を左揃えにするにはADLARに1を書いてください。さもなければ結果は右揃えです。ADLARビットの変更はどんな進行中の変換にも拘らず、直ちにA/D変換データレジスタの内容に影響を及ぼします。

| ビット | | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | ADCH |
|------------|------|------|------|------|------|------|------|------|------|------|
| \$1A | | - | - | - | - | - | - | ADC9 | ADC8 | |
| Read/Write | | R | R | R | R | R | R | R | R | |
| ADLAR=0時 | 初期値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ADCL |
| | ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| | \$19 | ADC7 | ADC6 | ADC5 | ADC4 | ADC3 | ADC2 | ADC1 | ADC0 | |
| Read/Write | | R | R | R | R | R | R | R | R | ADCH |
| 初期値 | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | |
| ADLAR=1時 | | ADC9 | ADC8 | ADC7 | ADC6 | ADC5 | ADC4 | ADC3 | ADC2 | ADCL |
| | ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| | | ADC1 | ADC0 | - | - | - | - | - | - | |

●ビット2~0 – ADTS2~0 : A/D変換自動起動要因選択 (ADC Auto Trigger Source)

A/D変換制御/状態レジスタ(ADCSRA)のA/D変換自動起動許可(ADATE)ビットが1を書かれると、これらのビットの値はどの起動元がA/D変換を起動するのを選択します。ADATEが解除(0)されると、ADTS2~0設定は無効です。変換は選択した割り込みフラグの上昇端によって起動されます。解除(0)されている起動元から設定(1)されている起動元への切り替えが起動信号上に上昇端を生成することに注意してください。ADCSRAのA/D許可(ADEN)ビットが設定(1)されているなら、これが変換を開始させます。連続変換動作(ADTS2~0=0)への切り替えは、例えばA/D変換完了割り込み要求フラグが設定(1)されていても、起動事象を引き起こしません。

| ADTS2~0 | 起動元 |
|---------|----------------|
| 0 0 0 | 連続変換動作 |
| 0 0 1 | アナログ比較器 |
| 0 1 0 | 外部割り込み要求0 |
| 0 1 1 | タイマ/カウンタ0比較A一致 |
| 1 0 0 | タイマ/カウンタ0溢れ |
| 1 0 1 | タイマ/カウンタ0比較B一致 |
| 1 1 0 | ピン変化割り込み |
| 1 1 1 | タイマ/カウンタ0捕獲発生 |

19.13.4. ADCH,ADCL – A/D変換データレジスタ上位/下位バイト (ADC Data Register Low and High byte) [ADLAR=0]

ADCHとADCLのレジスタ対は16ビット値のADCデータレジスタを表します。下位バイト[7~0](接尾辞L)は変位原点でアクセスできます。上位バイト[15~8](接尾辞H)は変位+1でアクセスすることができます。16ビットレジスタの読み書きのより多くの詳細については「16ビットレジスタのアクセス」をご覧ください。

A/D変換が完了すると、その結果がADCHとADCLの2つのレジスタで得られます。

ADCLが読まれると、A/DデータレジスタはADCHが読まれるまで更新されません。従ってこの結果が左揃えで且つ8ビットを越える精度が必要とされないなら、ADCHを読むことで用が足ります。さもなければADCLが先に、その後にADCHが読まれなければなりません。

A/D変換制御/状態レジスタB(ADCSRB)の左揃え選択(ADLAR)ビットとA/D多重器選択レジスタ(ADMUX)のA/Dチャネル選択(MUX2~0)ビットはこのレジスタから結果を読む方法に影響を及ぼします。ADLARが設定(1)ならば結果は左揃えにされます。ADLARが解除(0:既定)ならば結果は右揃えにされます。

名称 : ADCHとADCL
変位 : \$19
リセット : \$00
特質 : ADLAR=0

| ビット | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 0 |
|--------|--------|----|----|----|----|----|--------|---|
| | – | – | – | – | – | – | ADC9,8 | |
| アクセス種別 | R | R | R | R | R | R | R | R |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ADC7~0 | | | | | | | |
| アクセス種別 | R | R | R | R | R | R | R | R |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

● ビット9~0 – ADC9~0 : A/D変換結果 (ADC Conversion result)

これらのビットは変換での結果を表します。詳細については「A/D変換の結果」を参照してください。

19.13.5. ADCH,ADCL – A/D変換データレジスタ上位/下位バイト (ADC Data Register Low and High byte) [ADLAR=1]

ADCHとADCLのレジスタ対は16ビット値のADCデータレジスタを表します。下位バイト[7~0](接尾辞L)は変位原点でアクセスできます。上位バイト[15~8](接尾辞H)は変位+1でアクセスすることができます。16ビットレジスタの読み書きのより多くの詳細については「16ビットレジスタのアクセス」をご覧ください。

A/D変換が完了すると、その結果がADCHとADCLの2つのレジスタで得られます。

ADCLが読まれると、A/DデータレジスタはADCHが読まれるまで更新されません。従ってこの結果が左揃えで且つ8ビットを越える精度が必要とされないなら、ADCHを読むことで用が足ります。さもなければADCLが先に、その後にADCHが読まれなければなりません。

A/D変換制御/状態レジスタB(ADCSRB)の左揃え選択(ADLAR)ビットとA/D多重器選択レジスタ(ADMUX)のA/Dチャネル選択(MUX2~0)ビットはこのレジスタから結果を読む方法に影響を及ぼします。ADLARが設定(1)ならば結果は左揃えにされます。ADLARが解除(0:既定)ならば結果は右揃えにされます。

名称 : ADCHとADCL
変位 : \$19
リセット : \$00
特質 : ADLAR=1

| ビット | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|--------|--------|----|----|----|----|----|---|---|
| | ADC9~2 | | | | | | | |
| アクセス種別 | R | R | R | R | R | R | R | R |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ADC1,0 | | – | – | – | – | – | – |
| アクセス種別 | R | R | R | R | R | R | R | R |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

● ビット15~6 – ADC9~0 : A/D変換結果 (ADC Conversion result)

これらのビットは変換での結果を表します。詳細については「A/D変換の結果」を参照してください。

19.13.6. DIDR0 – デジタル入力禁止レジスタ0 (Digital Input Disable Register 0)

各々のビットは論理1を書かれると、対応するADCピンのデジタル入力緩衝部が禁止されます。このビットが設定(1)されると、対応するポート入力レジスタのビット(PINx)は常に0として読みます。アナログ信号がADC7～0ピンに印加され、そのピンからのデジタル入力が必要とされない時にデジタル入力緩衝部での消費電力を削減するため、そのビットは論理1を書かれるべきです。

名称 : DIDR0
変位 : \$17
リセット : \$00
特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|
| | ADC7D | ADC6D | ADC5D | ADC4D | ADC3D | ADC2D | ADC1D | ADC0D |
| アクセス種別 | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

● ビット7～0 – ADC7D～ADC0D : ADC7～ADC0デジタル入力禁止 (ADC7～0 Digital Input Disable)

A/D変換器:

- ADC7D～ADC0Dが1に設定されると、対応するADCピンのデジタル入力緩衝部が禁止されます。このビットが設定(1)される時、対応するポート入力(PIN)レジスタのビットは常に0として読みます。ADC7～0ピンにアナログ信号が印加され、そのピンからのデジタル入力が必要ない時に、デジタル入力緩衝部での電力消費を減らすため、このビットは論理1を書かれるべきです。

アナログ比較器:

- ADC1DまたはADC0Dが1に設定されると、AIN1(ADC1)/AIN0(ADC0)ピンのデジタル入力緩衝部が禁止されます。このビットが設定(1)されると、対応するポート入力レジスタのビット(PINx)は常に0として読みます。アナログ入力として使うけれどもデジタル入力を必要としない時にこのビットに論理1を書くことによってデジタル入力緩衝部での消費電力を削減することができます。
- ADC7D～ADC2D : これらのビットはアナログ比較器に適用しません。

20. TPI – Tiny フログラミング インターフェース (Tiny programming Interface)

20.1. 概要

Tinyフログラミング インターフェース(TPI)は全ての不揮発性メモリ(NVM)の外部フログラミングを支援します。メモリ フログラミングは「MEMPROG – メモリ フログラミング」で記述されるように、NVM制御器指令を実行することによってNVM制御器経由で行われます。

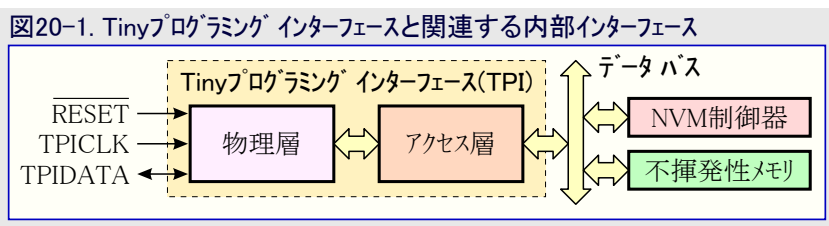
関連リンク 114頁の「MEMPROG – メモリ フログラミング」

20.2. 特徴

- 物理層:
 - 同期データ転送
 - 半二重双方向の送受信部
 - 1開始ビット、8ビット データ、1パリティビット、2停止ビットの固定構成形式
 - パリティ誤り検出、フレーミング誤り検出、中断キャラクタ検出
 - パリティ生成と衝突検出
 - データ送受信間への自動保護時間挿入
- アクセス層:
 - メッセージに基づく通信
 - 自動的な例外処理機構
 - 簡潔な命令一式
 - NVMフログラミング アクセス制御
 - Tinyフログラミング インターフェース制御/状態空間アクセス制御
 - データ空間アクセス制御

20.3. 構成図

Tinyフログラミング インターフェース(TPI)はフログラミング設備へのアクセスを提供します。このインターフェースは物理層とアクセス層の2つの層から成ります。



フログラミングは物理的なインターフェース経由で行われます。これは許可としてのRESETピン、クロック入力としてのTPICLKピン、データ入出力としてのTPIDATAピンを使う3ピン インターフェースです。NVMは1.8～5.5Vでフログラミングすることができます。

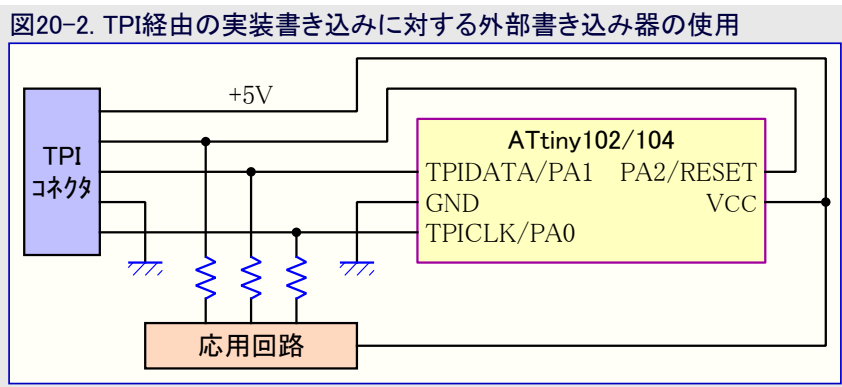
20.4. Tinyフログラミング インターフェースの物理層

TPI物理層は基本的な低位直列通信を扱います。TPI物理層は半二重双方向直列送受信部を使います。物理層は直列から並列へと並列から直列へのデータ変換、フレーム開始検出、フレーム異常検出、パリティ誤り検出、パリティ生成、衝突検出を含みます。

TPIは次のような3つのピン経由でアクセスされます。

- RESET : Tinyフログラミング インターフェース許可入力
- TPICLK : Tinyフログラミング インターフェース クロック入力
- TPIDATA : Tinyフログラミング インターフェース データ入出力

加えて、外部書き込み器とデバイス間でVCCとGNDが接続されなければなりません。



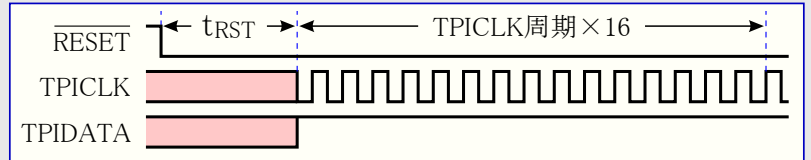
20.4.1. 許可

以下の手順がTinyプログラミング インターフェースを許可します。

1. VCCとGND間に5Vを印加してください。
2. 使われるリセットの方法に依存して、次のどちらかを実行してください。
 - t_{TOUT} （「システムとリセットの特性」参照）待つて \overline{RESET} ピンをLowに設定してください。これはデバイスのリセットしてTPI物理層を許可します。そしてプログラミング作業全体の間、 \overline{RESET} ピンはLowに保たなければなりません。
 - $RSTDISBL$ 構成設定ビットがプログラム(0)されている場合、 \overline{RESET} ピンに12Vを印加してください。 \overline{RESET} ピンはプログラミング作業全体の間、12Vに保たなければなりません。
3. t_{RST} （「システムとリセットの特性」参照）待つてください。
4. 16 TPICLK周期の間、TPIDATAピンをHighに保ってください。

関連リンク [124頁の「システムとリセットの特性」](#)

図20-3. Tinyプログラミング インターフェースを許可するための流れ



20.4.2. 禁止

NVM許可(NVMEN)ビットが解除(0)されていれば、 \overline{RESET} ピンが不活性のHigh状態に開放されるか、または代替としてのVHV(12V)がもはや \overline{RESET} ピンに印加されない場合に、TPIは自動的に禁止されます。

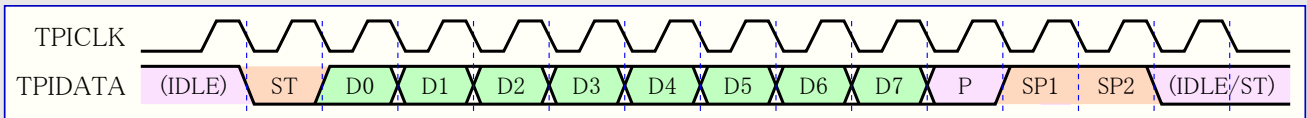
NVM許可ビットが解除(0)されていなければ、TPIプログラミング動作形態を抜け出すのに電源断が必要とされます。「TPISR – Tinyプログラミング インターフェース状態レジスタ」のNVMENビットをご覧ください。

関連リンク [112頁の「TPISR – Tinyプログラミング インターフェース状態レジスタ」](#)

20.4.3. フレーム形式

TPI物理層は固定フレーム形式を支援します。フレームは8ビット長の1つのキャラクタ、1つの開始ビット、パリティビット、2つの停止ビットから成ります。データは下位ビット先行で転送されます。

図20-4. 直列フレーム形式



上図で使った記号は次のとおりです。

- ST : 開始ビット (常にLow)
- D0～D7 : データビット (下位ビット先行送出)
- P : パリティビット (偶数パリティを使用)
- SP1 : 停止ビット1 (常にhigh)
- SP2 : 停止ビット2 (常にhigh)

20.4.4. パリティビット計算

パリティビットは常に偶数パリティを用いて計算されます。ビットの値は次のように全てのデータビットの排他的論理和(EOR)を行うことによって計算されます。

$$P = D0 \text{ EOR } D1 \text{ EOR } D2 \text{ EOR } D3 \text{ EOR } D4 \text{ EOR } D5 \text{ EOR } D6 \text{ EOR } D7 \text{ EOR } 0$$

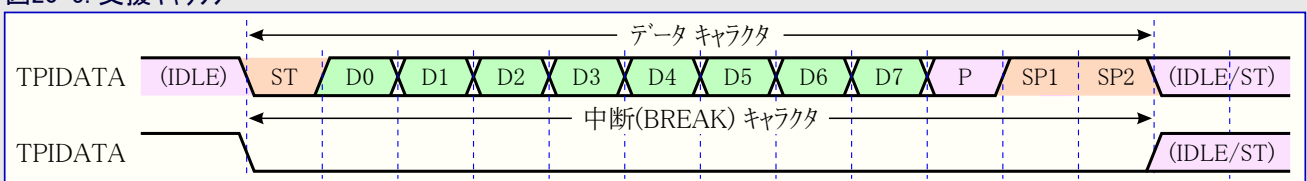
ここでの記号は次のとおりです。

- P : 偶数パリティを用いたパリティビット
- D0～D7 : キャラクタのデータビット

20.4.5. 支援キャラクタ

中断(BREAK)キャラクタは12ビット長のLowレベルと等価です。これは12ビット長を越えて延長することができます。

図20-5. 支援キャラクタ

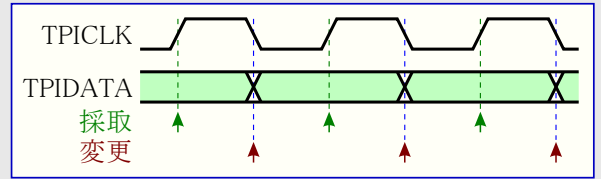


20.4.6. 動作

TPI物理層は外部書き込み器によって供給されるTPICLKに同期して動作します。データ採取やデータ変更とクロック端の間での依存性が右図で示されます。データは下降端で変更され、上昇端で採取されます。

TPI物理層は送信と受信の2つの動作形態を支援します。既定でのこの層は受信動作形態で開始ビットを持ちます。動作形態はアクセス層によって制御されます。

図20-6. データ変更とデータ採取



20.4.7. 直列データ受信

TPI物理層が受信動作形態の時は開始ビットが検出されると直ぐにデータ受信が開始されます。開始ビットに後続する各ビットがTPICLKの上昇端で採取され、第2停止ビットが受信されるまで移動レジスタに順次移動されます。移動レジスタに完全なフレームが存在する時に受信したデータはTPIアクセス層に対して利用可能です。

受信動作形態にはフレーム異常、パリティ誤り、中断検出の3つの例外が有り得ます。これらの全ての例外はTPIアクセス層に対して合図され、そして異常状態へ移行してTPI物理層を受信動作形態に置き、中断(BREAK)キャラクタを待ちます。

- ・フレーム異常例外。フレーム異常例外は停止ビットの状態を示します。フレーム異常例外は停止ビットが0として読まれた場合に設定されます。
- ・パリティ誤り例外。データビットのパリティはフレーム受信中に計算されます。フレームが完全に受信された後に結果がフレームのパリティビットと比較されます。比較失敗の場合にパリティ誤り例外が合図されます。
- ・中断検出例外。中断検出例外は全て0の完全なフレームが受信された時に生成されます。

20.4.8. 直列データ送信

TPI物理層が新しいフレームを送出する準備が整っている時に、送信するデータを移動レジスタに格納することによって、それがデータ送信を始めます。移動レジスタが新データを格納されると、送信部はTPICLKによって与えられる転送速度で完全なフレームをTPIDATAへ送り出します。

送信中に衝突が検出された場合は出力駆動部が禁止されます。TPIアクセス層は異常状態へ移行してTPI物理層が受信動作形態に置かれ、中断(BREAK)キャラクタを待ちます。

20.4.9. 衝突検出例外

TPI物理層はデータの送信と受信の両方に1つの双方向データ線を使います。外部書き込み器とTPI物理層が同時にTPIDATAを駆動した場合に有り得る駆動競合が起こるかもしれません。駆動競合の影響を減らすため、衝突検出機構が支援されます。衝突検出はTPI物理層が駆動するTPIDATA線の状態に基づきます。

TPIDATA線は内部プルアップ付きの3状態プッシュプル駆動部によって駆動されます。論理0が送出される時に出力駆動部は常に許可されます。連続する論理1を送出する時に、その出力は最初のクロック周期の間だけ活動的に駆動されます。その後、出力駆動部は自動的にHi-Zにされ、TPIDATA線は内部プルアップによってHighを保持されます。出力は次に論理0が送出される時に再び許可されます。

衝突検出は送信動作形態で出力駆動部が禁止される時に許可されます。データ線は内部プルアップによって直ぐにHighを保持されるべきで、そしてそれが外部書き込み器によってLowに駆動されるかを見るために監視されます。その出力がLowに読まれる場合に衝突が検出されます。

衝突検出が実行される方法に関連するいくつかの潜在的な落とし穴があります。例えば、TPI物理層が連続する論理0のビット列、または論理1と0を交互に切り替えるビット列を送信する時に衝突を検出することができません。これは全ての時で出力駆動部が活性でTPIDATA線のポーリングを妨げるためです。けれども、単一フレーム内では2つの停止ビットが常に論理1として送信されるべきで、(フレーム形式が停止ビットに関して違反されていない限り)フレームに対して最低1回の衝突検出を許可します。

TPI物理層はTPIDATA線で衝突を検出した時に送信を止めます。衝突はTPIアクセス層へ合図され、そして物理層を直ちに受信動作形態に変更して異常状態になります。中断(BREAK)キャラクタを送出することによってだけ、TPIアクセス層は異常状態から回復することができます。

20.4.10. 方向変更

半二重動作の正しいタイミングを保証するため、物理層に簡単な保護時間機構が付加されています。TPI物理層が受信から送信の動作形態へ変更する時に、開始ビットが送信されるのに先立って構成設定可能な追加IDLEビット数が挿入されます。送受信動作形態間の最低遷移時間は2 IDLEビットです。合計IDLE時間は指定した保護時間+2 IDLEビットです。

保護時間はTinyプログラミング インターフェース物理層制御レジスタ(TPIPCR)の専用ビットによって構成設定されます。物理層が初期化された後の既定の保護時間値は128ビットです。

外部書き込みはTPI目的対象が受信から送信へ動作形態を変更する時にTPIDATA線の制御権を失います。保護時間機能はこの通信の危険な段階を緩和します。外部書き込み器が受信から送信へ動作形態を変更する時に、開始ビットが送信されるのに先立って最低1つのIDLEビットが挿入されるべきです。

20.5. Tinyプログラミング インターフェースのアクセス層

TPIアクセス層には外部書き込み器との通信を処理する責任があります。通信はメッセージ形式に基き、各メッセージは1またはより多くのバイト単位のアペラントが後続する命令から成ります。命令は常に外部書き込み器によって送られますが、アペラントは発行された命令の形式に依存して、外部書き込み器またはTPIアクセス層のどちらかによって送られます。

TPIアクセス層はTPI物理層のキャラクタ転送方向を制御します。例外後の異常状態からの回復も処理します。

Tinyプログラミング インターフェースの**制御/状態空間(CSS)**はTPIアクセス層の制御と状態のレジスタに割り当てられます。CSSはTPI自身の動作に直接影響するレジスタから成ります。これらのレジスタは**SLDCS**と**SSTCS**の命令を用いてアクセス可能です。

アクセス層は直接的、またはアドレスポインタとして**ポインタレジスタ(PR)**を用いて間接的のどちらかでデータ空間もアクセスすることができます。データ空間は**SLD**,**SST**,**SIN**,**SOUT**の命令を用いてアクセス可能です。アドレスポインタは**SSTPR**命令を用いてポインタレジスタに格納することができます。

20.5.1. メッセージ形式

各メッセージは1またはより多くのバイト アペラントが後続する命令から成ります。命令は常に書き込み器によって送られます。命令に依存して後続するアペラントの全ては外部書き込み器またはTPIのどちらかによって送出されます。

メッセージは次のように、命令に基いて2つの形式に分類することができます。

- 書き込みメッセージ。書き込みメッセージはデータ書き込みを求められます。書き込みメッセージは全体的に外部書き込み器によって送られます。このメッセージ形式は**SSTCS**,**SST**,**SSTPR**,**SOUT**,**SKEY**の命令で使われます。
- 読み込みメッセージ。読み込みメッセージはデータ読み込みを求められます。TPIはバイト アペラントを送出することによって要求に反応します。このメッセージ形式は**SLDCS**,**SLD**,**SIN**の命令で使われます。

SKEY命令を除く全ての命令は命令に続く1バイトのアペラントが必要です。**SKEY**命令は8バイトのアペラントが必要です。より多くの情報については「**TPI命令**」をご覧ください。

20.5.2. 例外処理と同期

TPIの定常動作からの例外は多くの状況が考慮されます。TPI物理層が受信動作形態の時のこれらの例外は以下です。

- TPI物理層がパリティ誤りを検出
- TPI物理層がフレーム異常を検出
- TPI物理層が中断(BREAK)キャラクタを認証

TPI物理層が送信動作形態の時に有り得る例外は以下です。

- TPI物理層がデータ衝突を検出

これら全ての例外がTPIアクセス層に合図されます。アクセス層は進行中のどの操作も中止することによって例外に返答し、そして異常状態へ移行します。アクセス層は中断(BREAK)キャラクタが受信されるまで異常状態に留まり、その後に既定状態へ戻されます。結果として外部書き込み器は単に2つの中断(BREAK)キャラクタを送信することによって常に規約を同期することができます。

20.6. 命令一式

TPIは**TPI制御/状態空間(CSS)**とデータ空間をアクセスするのに使われる簡潔な命令一式を持ちます。命令はTPI、NVM制御器、NVMメモリのアクセスを外部書き込み器に許します。**SKEY**を除く全ての命令は命令に続く1バイトのアペラントが必要です。**SKEY**命令は8バイトのデータが後続します。全ての命令はバイトの大きさです。

表20-1. 命令一式要約

| ニモニック | アペラント | 説明 | 動作 |
|--------------|---------------|-------------------------------|---------------------|
| SLD | data,PR | 間接アドレス指定を使用してデータ空間から直列取得 | data←DS[PR] |
| | data,PR+ | 間接アドレス指定を使用してデータ空間から直列取得,事後増加 | data←DS[PR],PR←PR+1 |
| SST | PR,data | 間接アドレス指定を使用してデータ空間へ直列格納 | DS[PR]←data |
| | PR+,data | 間接アドレス指定を使用してデータ空間へ直列格納,事後増加 | DS[PR]←data,PR←PR+1 |
| SSTPR | PR,a | 直接アドレス指定を使用してポインタレジスタへ直列格納 | PR[a]←data |
| SIN | data,a | データ空間から直列入力 | I/O[a]←data |
| SOUT | a,data | データ空間へ直列出力 | data←I/O[a] |
| SLDCS | data,a | 直接アドレス指定を使用して制御/状態空間から直列取得 | CSS[a]←data |
| SSTCS | a,data | 直接アドレス指定を使用して制御/状態空間へ直列格納 | data←CSS[a] |
| SKEY | Key,{8{data}} | 鍵直列設定 | Key←{8{data}} |

20.6.1. SLD – 直列で間接アドレス指定を使ってデータ空間から取得

SLD命令は直列読み出しのためにデータ空間からTPI物理層移動レジスタへデータ バイトを格納するのに間接アドレス指定を用います。データ空間位置はポインタ レジスタ(PR)によって位置付けされ、このアドレスはデータがアクセスされる前に格納されなければなりません。ポインタ レジスタは、その操作によって無変化のままか、または事後増加されるかのどちらかです。

表20-2. 直列でデータ空間からの取得(SLD)命令

| 動作 | オペコード | 備考 | レジスタ |
|-------------|-----------|---------|------|
| data←DS[PR] | 0010 0000 | PR←PR | 無変化 |
| data←DS[PR] | 0010 0100 | PR←PR+1 | 事後増加 |

20.6.2. SST – 直列で間接アドレス指定を使ってデータ空間へ設定

SST命令は物理層移動レジスタへ移されたデータ バイトをデータ空間へ格納するのに間接アドレス指定を用います。データ空間位置はポインタ レジスタ(PR)によって位置付けされ、このアドレスはデータがアクセスされる前に格納されていなければなりません。ポインタ レジスタは、その操作によって無変化のままか、または事後増加されるかのどちらかにすることができます。

表20-3. 直列でデータ空間への格納(SST)命令

| 動作 | オペコード | 備考 | レジスタ |
|-------------|-----------|---------|------|
| DS[PR]←data | 0110 0000 | PR←PR | 無変化 |
| DS[PR]←data | 0110 0100 | PR←PR+1 | 事後増加 |

20.6.3. SSTPR – 直列でポインタ レジスタへ設定

SSTPR命令は物理層移動レジスタへ移されたデータ バイトをポインタ レジスタ(PR)へ格納します。命令のアドレス ビットはポインタ レジスタのどのバイトがアクセスされるのかを指定します。

表20-4. 直列でポインタ レジスタへの格納(SSTPR)命令

| 動作 | オペコード | 備考 |
|------------|-----------|-------------------------------|
| PR[a]←data | 0110 100a | ビット'a'はポインタ レジスタのバイト位置を指示します。 |

20.6.4. SIN – 直列で直接アドレス指定を使ってI/O空間から取得

SIN命令は直列読み出しのためにI/O空間からTPI物理層移動レジスタへデータ バイトを格納します。この命令は直接アドレス指定を用いて、アドレスは命令の6つのアドレス ビットから成ります。

表20-5. 直列で直接アドレス指定を使ってI/O空間から取得(SIN)命令

| 動作 | オペコード | 備考 |
|-------------|-----------|--------------------------------|
| data←I/O[a] | 0aa1 aaaa | 'a'と記されたビットは直接6ビットのアドレスを形成します。 |

20.6.5. SOUT – 直列で直接アドレス指定を使ってI/O空間へ設定

SOUT命令は物理層移動レジスタへ移されたデータ バイトをI/O空間へ格納します。この命令は直接アドレス指定を用い、アドレスは命令の6つのアドレス ビットから成ります。

表20-6. 直列で直接アドレス指定を使ってI/O空間へ格納(SIN)命令

| 動作 | オペコード | 備考 |
|-------------|-----------|--------------------------------|
| I/O[a]←data | 1aa1 aaaa | 'a'と記されたビットは直接6ビットのアドレスを形成します。 |

20.6.6. SLDCS – 直列で直接アドレス指定を使って制御/状態空間からデータ取得

SLDCS命令は直列読み出しのためにTPI制御/状態空間からTPI物理層移動レジスタへデータ バイトを格納します。SLDCS命令は直接アドレス指定を用い、直接アドレスは命令の4つのアドレス ビットから成ります。

表20-7. 直列で直接アドレス指定を使って制御/状態空間から取得(SLDCS)命令

| 動作 | オペコード | 備考 |
|-------------|-----------|--------------------------------|
| data←CSS[a] | 1000 aaaa | 'a'と記されたビットは直接4ビットのアドレスを形成します。 |

20.6.7. SSTCS – 直列で直接アドレス指定を使って制御/状態空間へデータ設定

SSTCS命令はTPI物理層移動レジスタに移されたデータバイトをTPI制御/状態空間へ格納します。表14-8.で示されるように、SSTCS命令は直接アドレス指定を用い、直接アドレスは命令の4つのアドレスビットから成ります。

表20-8. 直列で直接アドレス指定を使って制御/状態空間から取得(SLDCS)命令

| 動作 | オペコード | 備考 |
|-------------|-----------|--------------------------------|
| CSS[a]←data | 1100 aaaa | 'a'と記されたビットは直接4ビットのアドレスを形成します。 |

20.6.8. SKEY – 直列で鍵合図

SKEY命令はNVMプログラミングを許可する活性化鍵の合図に使われます。表14-9.で示されるように、SKEY命令は活性化鍵を含む8バイトのデータが後続します。

表20-9. 直列で鍵合図(KEY)命令

| 動作 | オペコード | 備考 |
|---------------|-----------|------------------|
| KEY←{8}data}} | 1110 0000 | データバイトが命令後に続きます。 |

20.7. 不揮発性メモリ制御器のアクセス

既定でのNVMプログラミングは許可されていません。NVM制御器をアクセスして不揮発性メモリをプログラミングできるようにするには、SKEY命令を使って固有鍵が送られなければなりません。

表20-10. 直列鍵合図(KEY)命令

| 鍵 | 値 |
|--------------|--------------------|
| NVMプログラミング許可 | \$1289AB45CDD888FF |

鍵が与えられた後、不揮発性メモリが許可されるまでTPI状態レジスタ(TPISR)の不揮発性メモリ許可(NVMEN)ビットがポーリングされなければなりません。

NVMプログラミングはTPISRのNVMENビットへ論理0を書くことによって禁止されます。

20.8. 制御/状態空間レジスタ説明

Tinyプログラミング インターフェースの制御と状態のレジスタはこのインターフェースの制御/状態空間(CSS)に割り当てられます。これらのレジスタはI/Oレジスタ割り当ての一部ではなく、SLDCSとSSTCSの命令経由でだけアクセス可能です。制御と状態のレジスタはTPIの構成設定と状態監視に直接的に関係します

表20-11. 制御と状態のレジスタ要約

| アドレス | 名称 | ビット7 | ビット6 | ビット5 | ビット4 | ビット3 | ビット2 | ビット1 | ビット0 |
|-----------|--------|--------------------------|------|------|------|------|------|-------|------|
| \$0F | TPIIR | Tinyプログラミング インターフェース識別符号 | | | | | | | |
| \$03～\$0E | (予約) | | | | | | | | |
| \$02 | TPIPCR | - | - | - | - | - | GT2 | GT1 | GT0 |
| \$01 | (予約) | | | | | | | | |
| \$00 | TPISR | - | - | - | - | - | - | NVMEN | - |

20.8.1. TPIIR – Tinyプログラミング インターフェース識別レジスタ (Tiny Programming Interface Identification Register)

名称 : TPIIR
変位 : -
リセット : \$00
特質 : CSS:\$0F

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|----------|-----|-----|-----|-----|-----|-----|-----|
| | TPIIC7~0 | | | | | | | |
| アクセス種別 | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

● ビット7～0 – TPIIC7～0 : Tinyプログラミング インターフェース識別符号 (Tiny Programming Interface Identification Code)

これらのビットはTinyプログラミング インターフェースに対する識別符号を与えます。この符号はTPIの識別に対して外部書き込み器で使うことができます。

表20-12. Tinyプログラミング インターフェース用識別符号

| 符号 | 値 |
|------------|------|
| インターフェース識別 | \$80 |

20.8.2. TPIPCR – Tinyプログラミング インターフェース物理層制御レジスタ (Tiny Programming Interface Physical Control Register)

名称 : TPIPCR
変位 : -
リセット : \$00
特質 : CSS:\$02

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|---|---|---|---|---|-------|-----|-----|
| | - | - | - | - | - | GT2~0 | | |
| アクセス種別 | R | R | R | R | R | R/W | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

● ビット2~0 – GT2~0 : 保護時間 (Guard Time)

これらのビットは受信動作形態から送信動作形態に変更する時にアイドル時間へ挿入される追加IDLEビット数を指定します。送信動作形態から受信への変更時に追加遅延は挿入されません。

受信から送信へ動作形態を変更する時の合計アイドル時間は保護時間+2 IDLEビットです。

既定保護時間は128 IDLEビットです。通信速度向上のため、保護時間は最短の安全な値に設定すべきです。

表20-13. 保護時間設定

| GT2 | GT1 | GT0 | 保護時間(IDLEビット数) |
|-----|-----|-----|----------------|
| 0 | 0 | 0 | +128 (既定値) |
| 0 | 0 | 1 | +64 |
| 0 | 1 | 0 | +32 |
| 0 | 1 | 1 | +16 |
| 1 | 0 | 0 | +8 |
| 1 | 0 | 1 | +4 |
| 1 | 1 | 0 | +2 |
| 1 | 1 | 1 | +0 |

20.8.3. TPISR – Tinyプログラミング インターフェース状態レジスタ (Tiny Programming Interface Status Register)

名称 : TPISR
変位 : -
リセット : \$00
特質 : CSS:\$00

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|---|---|---|---|---|---|-------|---|
| | - | - | - | - | - | - | NVMEN | - |
| アクセス種別 | R | R | R | R | R | R | R/W | R |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

● ビット1 – NVMEN : 不揮発性メモリプログラミング許可 (Non-Volatile Memory Programming Enabled)

NVMプログラミングはこのビットが設定(1)されている時に許可されています。外部書き込み器は成功裏に許可されたインターフェースを検証するのにこのビットをポーリングすることができます。

NVMプログラミングはこのビットに0を書くことによって禁止されます。

21. MEMPROG – メモリプログラミング

21.1. 概要

不揮発性メモリ(NVM)制御器は不揮発性メモリに対する全てのアクセスを管理します。NVM制御器はNVMタイミングとアクセス権を制御し、そしてNVMの状態を保持します。

通常実行中、CPUはフラッシュメモリ(プログラムメモリ)のコード領域からコードを実行します。**休止形態**に入ってプログラミング動作が活性でない時に、フラッシュメモリは消費電力を最小にするために禁止されます。

全てのNVMはデータメモリに割り当てられます。応用ソフトウェアは間接アドレス指定の取得命令を用いてデータメモリの割り当てられた位置からNVMを読むことができます。

NVMは1つの読み出し口しか持たず、従って次の命令とデータは同時に読むことができません。応用がデータ空間に割り当てられたNVM位置からデータを読む時は、次の命令が取得される前にデータが先行して読まれます。CPUの実行はここで1システムクロック周期遅らされます。

NVMに対する内部プログラミング(自己プログラミング)操作は禁止され、従ってNVMは応用ソフトウェアに対して読み込み専用として見えます。NVMの内部書き込みや消去は成功しません。

不揮発性メモリを書くために外部書き込み器によって用いられる方法は**外部プログラミング**として参照されます。外部プログラミングは実装または大量生産の両方で行われます。外部書き込み器は**Tinyプログラミングインターフェース(TPI)**経由でNVMを読み書きできます。

外部プログラミング動作形態では、読み込み専用の識票と校正の領域を除いて、全てのNVMを読み書きすることができます。

NVMは1.8～5.5Vでプログラミングすることができます。

21.2. 特徴

- 2種の組み込み不揮発性メモリ:
 - 不揮発性メモリ施錠ビット(NVM施錠ビット)
 - フラッシュメモリ
- フラッシュメモリ側の4つの独立した領域:
 - コード領域(プログラムメモリ)
 - 識票領域
 - 構成設定領域
 - 校正領域
- 応用ソフトウェアから全不揮発性メモリへの読み込みアクセス
- 外部書き込み器から不揮発性メモリへの読み書きアクセス:
 - 全不揮発性メモリに対する読み込みアクセス
 - NVM施錠ビット、フラッシュコード領域、フラッシュ構成設定領域に対する書き込みアクセス
- 外部プログラミング:
 - 実装書き換えと大量生産プログラミングを支援
 - Tinyプログラミングインターフェース(TPI)を通してのプログラミング
- NVM施錠ビットでの高い安全性

21.3. 不揮発性メモリ (NVM)

本デバイスは以下の組み込みNVMを持ちます。

- 不揮発性メモリ施錠ビット
- 4つの独立した領域を持つフラッシュメモリ
- 1Kバイトフラッシュメモリ
 - CPU実行は外部プログラミングを行っている間中停止されます。
- 付加列
 - フラッシュ – **固有ID**は追加が必要です。

21.3.1. 不揮発性メモリ施錠ビット

本デバイスは2つの施錠ビットを提供します。

表21-1. 施錠ビットバイト

| 名称 | ビット番号 | 説明 | 既定値 |
|-------|-------|--------------|------------|
| – | 7 | | 1 (非プログラム) |
| – | 6 | | 1 (非プログラム) |
| – | 5 | | 1 (非プログラム) |
| – | 4 | | 1 (非プログラム) |
| – | 3 | | 1 (非プログラム) |
| – | 2 | | 1 (非プログラム) |
| NVLB2 | 1 | 不揮発性メモリ施錠ビット | 1 (非プログラム) |
| NVLB1 | 0 | | 1 (非プログラム) |

施錠ビットは非プログラム(1)のままにしておくか、または追加の安全性を得るためにプログラム(0)することができます。施錠ビットはチップ消去指令でだけ、1に消去することができます。

表21-2. 施錠ビット保護種別

| 施錠種別 | メモリ施錠ビット (注) | | 保護形式 |
|------|--------------|-------|---|
| | NVLB2 | NVLB1 | |
| 1 | 1 | 1 | メモリ施錠機能が全く許可されません。 |
| 2 | 1 | 0 | フラッシュメモリの更なるプログラミング(書き込み)が外部プログラミング動作形態で禁止されます。構成設定領域ビットが外部プログラミング動作形態で施錠されます。 |
| 3 | 0 | 0 | フラッシュメモリの更なるプログラミング(書き込み)と照合(読み込み)が外部プログラミング動作形態で禁止されます。構成設定領域ビットが外部プログラミング動作形態で施錠されます。 |

注: • 0はプログラム、1は非プログラムを意味します。

• NVLB1とNVLB2をプログラム(0)する前に構成設定領域をプログラミングして(書いて)ください。

21.3.2. フラッシュメモリ

組み込みフラッシュメモリは4つの独立した領域を持ちます。

表21-3. フラッシュメモリ内のページ数とページの語数

| 領域 | 容量(バイト) | ページ容量(語) | ページ数 | PADDR | WADDR |
|----------------|-------------|----------|------|-------|-------|
| コード (プログラムメモリ) | 1024 (512語) | 8 | 64 | AP9~4 | AP3~1 |
| 構成設定 | 8 | 8 | 1 | – | AP3~1 |
| 識票 (注) | 16 | 8 | 2 | AP4 | AP3~1 |
| 校正 (注) | 8 | 8 | 1 | – | AP3~1 |

注: これらの領域は読み込み専用です。

21.3.3. 構成設定領域

ATtiny102/ATtiny104は構成設定領域に属する1つの構成設定バイトを持ちます。

表21-4. 構成設定バイト

| 構成設定バイト | 変位アドレス | 構成設定語データ |
|---------|--------|---|
| CONFW0 | \$04 | 構成設定語(ヒューズ値 – RSTDISBL, WDTON, CKOUT, SELFPROGEN) |

次表は全ての構成設定ビットの機能とそれらが構成設定バイトにどう割り当てられるかを簡単に記述します。

表21-5. 構成設定バイト0

| ビット名 | ビット | 説明 | 既定値 |
|------------|-----|-------------------------|------------|
| – | 7~4 | (予約) | 1 (非プログラム) |
| SELFPROGEN | 3 | 自己プログラム許可 (訳注:他に関連記述なし) | 1 (非プログラム) |
| CKOUT | 2 | システムクロック出力 | 1 (非プログラム) |
| WDTON | 1 | ウォッチドッグタイマ常時ON | 1 (非プログラム) |
| RSTDISBL | 0 | 外部リセット禁止 | 1 (非プログラム) |

構成設定ビットはチップ消去によって影響を及ぼされませんが、これらは構成設定領域消去指令を用いて解除することができます(本章内の「**構成設定領域消去**」をご覧ください)。**不揮発性施錠ビット1(NVLBI)**がプログラム(0)された場合に構成設定ビットが施錠されることに注意してください。

21.3.3.1. 構成設定ビットのラッチ

全ての構成設定ビットはデバイスがリセットされる時とデバイスが外部プログラミング動作形態を抜け出す時のどちらでもラッチされます。構成設定ビット値に対する変更はデバイスが外部プログラミング動作形態を去るまで無効です。

21.3.4. 識票領域

識票領域はデバイス識票のような種々雑多の情報を格納するのに用いられる専用のメモリ領域です。このメモリ領域の殆どは内部使用のために予約されています。

表21-6. 識票バイト

| 識票語アドレス | 識票語データ | |
|-----------|--------------|---------|
| | 上位バイト | 下位バイト |
| \$00 | デバイス識別1 | 製造者識別 |
| \$01 | (内部使用のために予約) | デバイス識別2 |
| \$02 | (内部使用のために予約) | |
| \$03~\$07 | 通番 | |

ATtiny102/ATtiny104はデバイスを識別するのに使うことができる3バイトの識票符号を持ちます。上の表で示されるようにこの3バイトは識票領域に属します。ATtiny102/ATtiny104用の識票データは次表で与えられます。

表21-7. 識票符号

| デバイス | 識票バイト | | |
|-----------|-------|---------|---------|
| | 製造者識別 | デバイス識別1 | デバイス識別2 |
| ATtiny102 | \$1E | \$90 | \$0C |
| ATtiny104 | \$1E | \$90 | \$0B |

21.3.4.1. 識票列要約

| バイト変位 | 名前 | ビット7 | ビット6 | ビット5 | ビット4 | ビット3 | ビット2 | ビット1 | ビット0 |
|-----------|------------------|------|------|------|------|--------------|------|------|------|
| \$00 | SIGROW_DEVICEID0 | | | | | DEVICEID07~0 | | | |
| \$01 | SIGROW_DEVICEID1 | | | | | DEVICEID17~0 | | | |
| \$02 | SIGROW_DEVICEID2 | | | | | DEVICEID27~0 | | | |
| \$03~\$05 | (予約) | | | | | | | | |
| \$06 | SIGROW_SERNUM0 | | | | | SERNUM07~0 | | | |
| \$07 | SIGROW_SERNUM1 | | | | | SERNUM17~0 | | | |
| \$08 | SIGROW_SERNUM2 | | | | | SERNUM27~0 | | | |
| \$09 | SIGROW_SERNUM3 | | | | | SERNUM37~0 | | | |
| \$0A | SIGROW_SERNUM4 | | | | | SERNUM47~0 | | | |
| \$0B | SIGROW_SERNUM5 | | | | | SERNUM57~0 | | | |
| \$0C | SIGROW_SERNUM6 | | | | | SERNUM67~0 | | | |
| \$0D | SIGROW_SERNUM7 | | | | | SERNUM77~0 | | | |
| \$0E | SIGROW_SERNUM8 | | | | | SERNUM87~0 | | | |
| \$0F | SIGROW_SERNUM9 | | | | | SERNUM97~0 | | | |

SIGROW_DEVICEIDn – デバイスIDバイトn (Device ID n)

名称 : SIGROW_DEVICEIDn
変位 : \$00 : \$01 : \$02 (n=0~2)
リセット : [デバイス識別値]
特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|--------------|---|---|---|---|---|---|---|
| | DEVICEIDn7~0 | | | | | | | |
| アクセス種別 | R | R | R | R | R | R | R | R |
| リセット値 | デバイス識別(ID)値 | | | | | | | |

●ビット7~0 – DEVICEIDn7~0 : デバイスIDバイトn値 (Byte n of the Device ID)

SIGROW_SERNUMn – 通番バイトn (Serial Number Byte n)

名称 : SIGROW_SERNUMn
変位 : \$06 : \$07 : \$08 : \$09 : \$0A : \$0B : \$0C : \$0D : \$0E : \$0F (n=0~9)
リセット : [デバイス通番値]
特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|------------|---|---|---|---|---|---|---|
| | SERNUMn7~0 | | | | | | | |
| アクセス種別 | R | R | R | R | R | R | R | R |
| リセット値 | デバイス通番値 | | | | | | | |

●ビット7~0 – SERNUMn7~0 : 通番バイトn値 (Serial Number n)

各デバイスは一意のIDで表される個別の通番を持ちます。これは現場で特定のデバイスを識別するのに使うことができます。通番は10バイトから成ります。

21.3.5. 校正領域

ATtiny102/ATtiny104は1つの校正バイトを持ちます。校正バイトは内蔵発振器用の校正データを含み、校正領域に属します。校正付き内蔵発振器の正しい周波数を保証するため、リセット中に校正バイトが自動的に発振校正(OSCCAL)レジスタ内に書かれます。

表21-8. 校正バイト

| 校正バイト | 変位アドレス | 校正語データ | |
|--------|-----------|--------|----------|
| | | 上位バイト | 下位バイト |
| OSCCAL | \$00 | (予約) | 内蔵発振器校正值 |
| (予約) | \$01~\$07 | (予約) | (予約) |

21.4. NVMのアクセス

NVM施錠ビットとフラッシュ メモリの領域は「データ メモリ」で示されるようにデータ空間に割り当てられます。NVMはこのデータ空間に割り当てられた位置経由で読み書きに関してアクセスすることができます。

NVM制御器は制御器にNVMで実行するのがどんな形式のプログラミング作業かを指示するのに使うことができる1式の指令を認証します。NVM制御器に対する指令はNVM指令レジスタ経由で発行されます。「NVMCMD – 不揮発性メモリ指令レジスタ」をご覧ください。選択した指令が格納された後、データ空間に割り当てられたNVM位置へデータを書くことによって操作が開始されます。

NVM制御器が操作実行中で多忙の時はNVM制御/状態レジスタのNVM多忙(NVMBSY)フラグ経由でこれを合図します。「NVMCSR – 不揮発性メモリ制御/状態レジスタ」をご覧ください。この多忙フラグが活性(1)である限り、NVM指令レジスタは書き込みアクセスに対して妨げられます。これは次の指令が開始され得る前に現在の指令が完全に実行されるのを保証します。

NVMのどの部分へのプログラミング(書き込み)も以下の操作を自動的に禁止します。

- NVMの他の何れかの部分に対する全てのプログラミング(書き込み)
- 何れかのNVM位置からの全ての読み込み

ATtiny102/ATtiny104は外部プログラミングと内部プログラミング(自己プログラミング)を支援します。

関連リンク [16 頁の「データ メモリ」](#)
[121 頁の「NVMCSR – 不揮発性メモリ制御/状態レジスタ」](#)
[121 頁の「NVMCMD – 不揮発性メモリ指令レジスタ」](#)

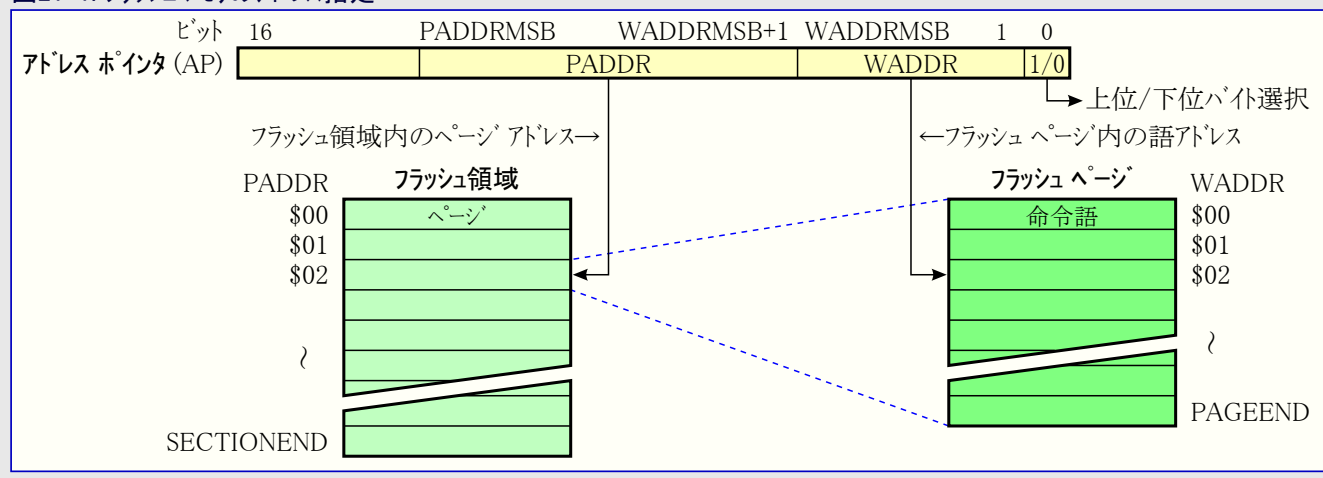
21.4.1. フラッシュのアドレス指定

データ空間はバイト アクセスを使いますが、フラッシュ領域がページで構成され、語としてアクセスされるため、データ空間のバイト アクセスはフラッシュ領域の語アドレスに変換されなければなりません。

データ空間アドレスの上位側ビットはデータ空間に割り当てられた**NVM施錠ビット**または**フラッシュ領域**を選択します。ページ内の語アドレス(WADDR)は[WADDRMSB~1]ビットにより、ページアドレス(PADDR)は[PADDRMSB~WADDRMSB+1]ビットによって保持されます。PADDRとWADDRは一緒にフラッシュ領域内の語の絶対アドレスを形成します。

フラッシュ領域アドレスの最下位ビットは語の上位または下位のバイトを選ぶのに用いられます。

図21-1. フラッシュメモリのアドレス指定



21.4.2. フラッシュ読み込み

フラッシュメモリはデータ空間割り当て位置から一度に1バイトずつ読むことができます。読み込み操作に対して語アドレス内の上位または下位のバイトを選ぶのに最下位ビット(ビット0)が使われます。このビットが**0**ならば下位バイトが読まれ、**1**ならば上位バイトが読まれます。

21.4.3. フラッシュ書き込み

フラッシュメモリは語単位で書くことができます。フラッシュ語を書く前に目的フラッシュ位置が消去されなければなりません。未消去フラッシュ語への書き込みはその内容を不正にします。

フラッシュメモリは書き込みに対して語でアクセスされ、データ空間はデータ空間に割り当てられたフラッシュメモリをアクセスするのにバイトアドレス指定を使います。従ってフラッシュメモリに対して正しい順序、即ち上位バイトの前の下位バイトで語を書くことが重要です。最初に下位バイトが一時緩衝器に書かれます。そして上位バイト書き込みがフラッシュ語緩衝器内に上位バイトと下位バイトの両方をラッチし、フラッシュメモリに対する書き込み操作を開始します。

フラッシュ消去操作はフラッシュ領域全体にだけ実行することができます。

フラッシュ書き込み手順は次のとおりです。

1. **フラッシュ領域消去**または**チップ消去**を実行してください。
2. 語単位でフラッシュ領域を書いてください。

21.4.3.1. チップ消去

チップ消去指令はフラッシュメモリのコード領域全体とNVM施錠ビットを消去します。安全性の理由のため、NVM施錠ビットはコード領域が完全に消去される前にリセットされません。構成設定、識票、校正の領域は変化しません。

チップ消去を始める前に、**NVM指令(NVMCMD)レジスタ**はチップ消去(**CHIP_ERASE**)指令を格納されなければなりません。消去操作を開始するには、フラッシュコード領域内に属する語位置の上位バイトに仮装バイトが書かれなければなりません。消去が完了されるまで**NVM多忙(NVMBSY)フラグ**が設定(**1**)に留まります。フラッシュメモリが消去されつつある間、フラッシュ緩衝器格納またはフラッシュメモリ読み込みのどちらも実行することができません。

チップ消去は次のようにして実行することができます。

1. NVMCMDレジスタに(**\$10,CHIP_ERASE**)を書いてください。
2. コード領域内のどれかの語位置の上位バイトに仮装バイトを書くことによって消去動作を開始してください。
3. NVMBYSYフラグが解除(**0**)されるまで待ってください。

21.4.3.2. コード領域消去

フラッシュコード領域の全ページを消去する方法は次のとおりです。

1. NVMCMDレジスタに**領域消去(\$14,SECTION_ERASE)**を書いてください。
2. コード領域内のどれかの語位置の上位バイトに偽装バイトを書くことによって消去動作を開始してください。
3. NVMBYSYフラグが解除(**0**)されるまで待ってください。

21.4.3.3. コード語書き込み

コード領域に語を書く方法は次のとおりです。

1. NVM指令(NVMCMD)レジスタに語書き込み(\$1D,WORD_WRITE)を書いてください。
2. 語位置の下位バイトにデータの下位バイトを書いてください。
3. 語位置の上位バイトにデータの上位バイトを書いてください。これがフラッシュ書き込み操作を始めます。
4. NVM多忙(NVMBSY)フラグが解除(0)されるまで待ってください。

21.4.3.4. 構成設定領域消去

構成設定領域を消去する方法は次のとおりです。

1. NVMCMDレジスタに領域消去(\$14,SECTION_ERASE)を書いてください。
2. 構成設定領域内のどれかの語位置の上位バイトに仮装バイトを書くことによって消去動作を開始してください。
3. NVMBSYフラグが解除(0)されるまで待ってください。

21.4.3.5. 構成設定領域書き込み

構成設定語を書く方法は次のとおりです。

1. NVMCMDレジスタに語書き込み(\$1D,WORD_WRITE)を書いてください。
2. 構成設定語位置の下位バイトにデータの下位バイトを書いてください。
3. 構成設定語位置の上位バイトにデータの上位バイトを書いてください。これがフラッシュ書き込み操作を始めます。
4. NVMBSYフラグが解除(0)されるまで待ってください。

21.4.4. NVM施錠ビット読み込み

不揮発性メモリ施錠バイトはデータ空間に割り当てられた位置から読むことができます。

21.4.5. NVM施錠ビット書き込み

施錠ビットを書く方法は次のとおりです。

1. NVMCMDレジスタに語書き込み(WORD_WRITE)指令を書いてください。
2. 不揮発性メモリ施錠バイト位置に施錠ビット値を書いてください。これは不揮発性メモリ施錠語の下位バイトです。
3. NVM施錠語位置の上位バイトに仮装バイトを書くことによってNVM施錠ビット書き込み動作を開始してください。
4. NVMBSYフラグが解除(0)されるまで待ってください。

21.5. 自己プログラミング

ATtiny102/ATtiny104のフラッシュメモリは書き込み中の読み込み(RWW:Read-While-Write)を支援せず、消去または書き込み中に読むことができません。故に、CPUは実行を停止します。

デバイスはMCU自身によるプログラムコードのダウンロード(書き込み)とアップロード(読み出し)に関する自己プログラミング機構を提供します。自己プログラミングでは語書き込み(WORD_WRITE)とページ消去(PAGE_ERASE)指令だけが支援されます。CPUはプログラミング操作を実行するためにNVMコードメモリ領域に”ページ消去”と”語書き込み”を実行することができます。

注: 使用者は正しいCPU動作を保証するために自己プログラミングを起動するST命令後2つのNOP命令を追加する必要があります。

アセンブリ言語プログラム例

自己プログラミング動作移行手順が以下で与えられます(R16はどのレジスタでもできます)。

```
LDI    R16, $E7                ;NVM自己プログラミング許可符号を取得
OUT    CCP, R16                ;自己プログラミング動作へ移行
```

プログラムはその後に4クロック周期内に望む自己プログラミング操作を実行しなければなりません。

ページ消去を実行するための完全なコードの例:

```
LDI    ZH, $43                 ;ページアドレスポインタ上位設定
LDI    ZL, $E0                 ;ページアドレスポインタ下位設定
LDI    temp, 0b011000          ;ページ消去指令を取得
OUT    NVMCMD, R16             ;ページ消去を指定
LDI    R16, $E7                ;NVM自己プログラミング許可符号を取得
OUT    CCP, R16                ;自己プログラミング動作へ移行
LDI    temp, $00               ;0値を取得
ST     Z+, temp                ;(4クロック周期内で)ページ消去実行起動
NOP                                     ;正しいCPU停止のために2つのNOPが必要
NOP                                     ;
```

21.6. 外部プログラミング

外部書き込み器を用いて不揮発性メモリをプログラミングするための方法が外部プログラミングとして参照されます。外部プログラミングは実装と大量生産の両方で行うことができます。

不揮発性メモリはTinyプログラミング インターフェース(TPI)経由で外部的にプログラミングすることができます。TPIの詳細については「[TPI – Tiny プログラミング インターフェース](#)」をご覧ください。TPIを用いる外部書き込み器はI/O空間に割り当てられたNVMの制御と状態のレジスタとデータメモリ空間に割り当てられたNVMメモリをアクセスすることができます。

関連リンク [107頁の「TPI – Tiny プログラミング インターフェース」](#)

21.6.1. 外部プログラミング動作形態移行

TPIは外部プログラミング動作形態へ移行され得る前に許可されなければなりません。以下の手順はTPIが許可された後で外部プログラミング動作形態へ移行する方法を記述します。

1. **SKEY命令**と共に**NVMメモリ アクセス鍵**を送ることによってNVMプログラミングの許可を要求してください。
2. それが設定(1)されるまで**TPI状態レジスタ(TPISR)**の**NVM許可(NVMEN)ビット**の状態をポーリングしてください。

TPIの許可とNVMプログラミングのより多くの詳細情報については「[TPI – Tiny プログラミング インターフェース](#)」の説明を参照してください。

関連リンク [107頁の「TPI – Tiny プログラミング インターフェース」](#)

21.6.2. 外部プログラミング動作形態抜け出し

NVMプログラミングを禁止するためにNVM許可(NVMEN)ビットを解除(0)し、そして $\overline{\text{RESET}}$ ピンを開放してください。

「**TPISR – Tinyプログラミング インターフェース状態レジスタ**」の**NVMENビット**をご覧ください。

関連リンク [113頁の「TPISR – Tinyプログラミング インターフェース状態レジスタ」](#)

21.7. NVMプログラミング用レジスタ

21.7.1. NVMCSR – 不揮発性メモリ制御/状態レジスタ (Non-Volatile Memory Control and Status Register)

名称 : NVMCSR

変位 : \$32

リセット : \$00

特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|--------|---|---|---|---|---|---|---|
| | NVMBSY | - | - | - | - | - | - | - |
| アクセス種別 | R/W | R | R | R | R | R | R | R |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

● ビット7 – NVMBSY : 不揮発性メモリ多忙 (Non-Volatile Memory Busy)

このビットはNVMメモリ(フラッシュメモリと施錠ビット)がプログラミング(書き込み)されつつあって多忙なことを示します。このビットはプログラミング(書き込み)動作が開始される時に設定(1)され、その操作が完了されるまで設定(1)に留まります。

21.7.2. NVMCMD – 不揮発性メモリ指令レジスタ (Non-Volatile Memory Command Register)

名称 : NVMCMD

変位 : \$33

リセット : \$00

特質 : -

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|---|---|-----------|-----|-----|-----|-----|-----|
| | - | - | NVMCMD5~0 | | | | | |
| アクセス種別 | R | R | R/W | R/W | R/W | R/W | R/W | R/W |
| リセット値 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

● ビット5~0 – NVMCMD5~0 : 不揮発性メモリ指令 (Non-Volatile Memory Command)

これらのビットはフラッシュメモリ用のプログラミング指令を定義します。

表21-9. NVMプログラミング指令

| 操作形式 | NVMCMD | | ニモニック | 説明 |
|------|---------|------|---------------|--------------|
| | 2進 | 16進 | | |
| 全般 | 00 0000 | \$00 | NO_OPERATION | 無操作 |
| | 01 0000 | \$10 | CHIP_ERASE | チップ消去 (注1) |
| | 01 0001 | \$11 | CHIP_WRITE | チップ書き込み (注2) |
| 領域 | 01 0100 | \$14 | SECTION_ERASE | 領域消去 |
| ページ | 01 1000 | \$18 | PAGE_ERASE | ページ消去 |
| 語 | 01 1101 | \$1D | WORD_WRITE | 語書き込み |

注: 自己プログラミングはNO_OPERATION, WORD_WRITE, PAGE_ERASEを支援します。

注1: コード領域と不揮発性メモリ(NVM)施錠ビットを消去。

注2: コード領域を書き込みますが、不揮発性メモリ(NVM)施錠ビットに影響を及ぼしません。

22. 電気的特性

22.1. 絶対最大定格

| | |
|----------------|----------------|
| 動作温度 | -55°C~125°C |
| 保存温度 | -65°C~150°C |
| RESETを除くピン許容電圧 | -0.5V~VCC+0.5V |
| RESETピン許容電圧 | -0.5V~13.0V |
| 最大動作電圧 | 6.0V |
| 入出力ピン毎のDC電流 | 40.0mA |
| VCCとGNDピンのDC電流 | 200.0mA |

注: 絶対最大定格を超える負担はデバイスに定常的な損傷を与えます。絶対最大定格は負担の定格を示すためだけのもので、この値または、この仕様書の動作特性で示された値を超える条件で動作することを示すものではありません。長時間の最大定格での使用はデバイスの信頼性を損なう場合があります。

22.2. DC特性

表22-1. DC特性 TA=-40°C~125°C, VCC=1.8V~5.5V (特記事項を除く)

| シンボル | 項目 | 条件 | 最小 | 代表 (注1) | 最大 | 単位 |
|-------------------|--------------------------------------|-------------------|-------------|---------|-------------|----|
| V _{IL} | Lowレベル入力電圧 (CLKI,RESETを除く) | VCC=1.8~2.4V | -0.5 | | 0.2VCC (注2) | V |
| | | VCC=2.4~5.5V | -0.5 | | 0.3VCC (注2) | |
| V _{IL1} | Lowレベル入力電圧 (CLKI) | VCC=1.8~5.5V | -0.5 | | 0.1VCC (注2) | |
| V _{IL2} | Lowレベル入力電圧 (RESET) | VCC=1.8~5.5V | -0.5 | | 0.1VCC (注2) | |
| V _{IL3} | Lowレベル入力電圧 (IOとしてのRESETピン) | VCC=1.8~2.4V | -0.5 | | 0.2VCC (注2) | |
| | | VCC=2.4~5.5V | -0.5 | | 0.3VCC (注2) | |
| V _{IH} | Highレベル入力電圧 (CLKI,RESETを除く) | VCC=1.8~2.4V | 0.7VCC (注3) | | VCC+0.5 | |
| | | VCC=2.4~5.5V | 0.6VCC (注3) | | VCC+0.5 | |
| V _{IH1} | Highレベル入力電圧 (XTAL1) | VCC=1.8~2.4V | 0.8VCC (注3) | | VCC+0.5 | |
| | | VCC=2.4~5.5V | 0.7VCC (注3) | | VCC+0.5 | |
| V _{IH2} | Highレベル入力電圧 (RESET) | VCC=1.8~5.5V | 0.9VCC (注3) | | VCC+0.5 | |
| V _{IH3} | Highレベル入力電圧 (IOとしてのRESETピン) | VCC=1.8~2.4V | 0.7VCC (注3) | | VCC+0.5 | |
| | | VCC=2.4~5.5V | 0.6VCC (注3) | | VCC+0.5 | |
| V _{OL} | Lowレベル出力電圧 (I/OとしてのRESETを除く) (注4,6) | IOL=10mA, VCC=5V | | | 0.6 | |
| | | IOL=5mA, VCC=3V | | | 0.5 | |
| V _{OH} | Highレベル出力電圧 (I/OとしてのRESETを除く) (注5,6) | IOH=-10mA, VCC=5V | 4.3 | | | |
| | | IOH=-5mA, VCC=3V | 2.5 | | | |
| I _{IL} | I/OピンLowレベル入力漏れ電流 | VCC=5.5V | | <0.05 | 1 | μA |
| I _{IH} | I/OピンHighレベル入力漏れ電流 | 確実なH/L範囲 | | <0.05 | 1 | |
| R _{RST} | RESETピンプルアップ抵抗 | | 30 | | 60 | kΩ |
| R _{PU} | I/Oピンプルアップ抵抗 | | 20 | | 50 | |
| I _{ACLK} | アナログ比較器入力漏れ電流 | VCC=5V, Vin=VCC/2 | -50 | | 50 | nA |
| I _{CC} | 活動動作消費電流 (注7) | VCC=2V, 1MHz | | 0.2 | 0.5 | mA |
| | | VCC=3V, 4MHz | | 1.1 | 1.2 | |
| | | VCC=5V, 8MHz | | 3.2 | 4 | |
| | アイドル動作消費電流 (注7) | VCC=2V, 1MHz | | 0.03 | 0.2 | |
| | | VCC=3V, 4MHz | | 0.2 | 0.5 | |
| | | VCC=5V, 8MHz | | 0.9 | 1.5 | |
| | パワーダウン動作消費電流 (注8) | VCC=3V, WDT有効 | TA=105°C | 5.5 | 10 | μA |
| | | | TA=125°C | 5.5 | 16 | |
| | | VCC=3V, WDT禁止 | TA=105°C | 0.11 | 2 | |
| | | | TA=125°C | 0.11 | 8 | |

注1: 25°Cでの代表値、特に断りのない限り最大値。

注2: “最大”はLowとして読まれることが保証されるピンでの最高値を意味します。

注3: “最小”はHighとして読まれることが保証されるピンでの最低値を意味します。

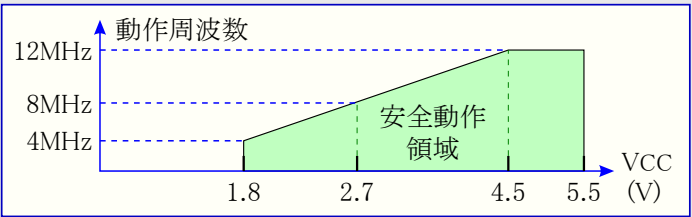
注4~8は次頁をご覧ください。

- 注4:** 各I/Oポートは安定状態(非過渡時)に於いて検査条件(VCC=3Vで5mA,VCC=5Vで10mA)よりも多くの吸い込み電流を流すことができますが、(全ポートに対する)全IOLの合計が60mAを超えるべきではありません。IOLが検査条件を超える場合、VOLは関連する仕様を超えるかもしれません。ピンは一覧にされた検査条件よりも大きな吸い込み電流を流すことを保証されません。
- 注5:** 各I/Oポートは安定状態(非過渡時)に於いて検査条件(VCC=3Vで5mA,VCC=5Vで10mA)よりも多くの吐き出し電流を流すことができますが、(全ポートに対する)全IOHの合計が60mAを超えるべきではありません。IOHが検査条件を超える場合、VOHは関連する仕様を超えるかもしれません。ピンは一覧にされた検査条件よりも大きな吐き出し電流を流すことを保証されません。
- 注6:** RESETピンはプログラミング動作へ移行して動作する時に高電圧を許容しなければならず、結果として、通常の入出力ピンと比べて弱い駆動能力を持ちます。
- 注7:** 「消費電力の最小化」で記述された方法を使う外部クロックでの値です。電力削減が許可(PRR=\$FF)され、入出力駆動はありません。
- 注8:** VCCレベル監視(VLM)禁止。

22.3. 速度

デバイスの最高動作周波数はVCCに依存します。最高周波数対VCC間の関連は1.8<VCC<4.5V間で直線です。

図22-1. 最高周波数対VCC



22.4. クロック特性

22.4.1. 校正付き内蔵RC発振器の精度

工場既定校正よりも高い精度に内蔵発振器を手動校正することが可能です。この発振器周波数が温度と電圧に依存することに注意してください。

表22-2. 校正付き内蔵RC発振器の校正精度

| 校正種別 | 周波数 | VCC | 温度 | 校正精度 (注) |
|-------|-------------------|----------------|---------------|----------|
| 工場校正 | 8.0MHz | 2.7～4.0V | 0～85℃ | ±2% |
| 使用者校正 | 7.3～8.1MHz内の固定周波数 | 1.8～5.5V内の固定電圧 | -40～85℃内の固定温度 | ±1% |

注: 校正点での発振器周波数精度(固定温度と固定電圧)

22.4.2. 外部クロック信号駆動

図22-2. 外部クロック駆動波形

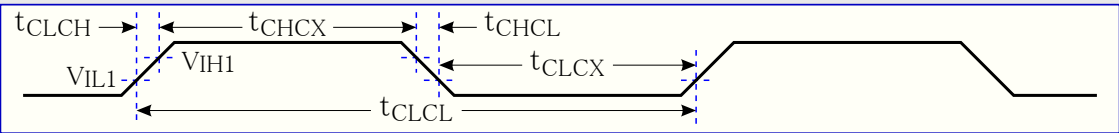


表22-3. 外部クロック特性

| シンボル | 項目 | VCC=1.8～5.5V | | VCC=2.7～5.5V | | VCC=4.5～5.5V | | 単位 |
|---------|---------------|--------------|-----|--------------|----|--------------|-----|-----|
| | | 最小 | 最大 | 最小 | 最大 | 最小 | 最大 | |
| 1/tCLCL | クロック周波数 | 0 | 4 | 0 | 8 | 0 | 12 | MHz |
| tCLCL | クロック周期 | 250 | | 125 | | 83 | | |
| tCHCX | Highレベル時間 | 100 | | 50 | | 33 | | ns |
| tCLCX | Lowレベル時間 | 100 | | 50 | | 33 | | |
| tCLCH | 上昇時間 | | 2.0 | | 1 | | 0.6 | μs |
| tCHCL | 下降時間 | | 2.0 | | 1 | | 0.6 | |
| ΔtCLCL | 隣接クロック周期間の変化率 | | 2 | | 2 | | 2 | % |

22.5. システムとリセットの特性

表22-4. リセットと内部電圧の電気的特性

| シンボル | 項目 | 条件 | 最小 | 代表 | 最大 | 単位 |
|------------|-------------------|--------------|--------|-----|--------|---------|
| V_{RST} | RESETピン閾値電圧 | | 0.2VCC | | 0.9VCC | V |
| V_{BG} | 内部バンドギャップ電圧 | VCC=1.8~5.5V | 1.0 | 1.1 | 1.2 | |
| t_{RST} | RESETピンでの最小パルス幅 | VCC=1.8V | | 2.1 | | μ s |
| | | VCC=3V | | 0.7 | | |
| | | VCC=5V | | 0.4 | | |
| t_{TOUT} | リセット起動後内部リセット保持時間 | | | 64 | 128 | ms |

注: 値は指針だけです。

22.5.1. 電源ONリセット

表22-5. 電源ONリセット特性 (TA=-40℃~125℃)

| シンボル | 項目 | 最小 | 代表 | 最大 | 単位 |
|-----------|---------------------|------|-----|-----|------|
| V_{POR} | 電源ONリセット開放閾値電圧 (注1) | 1.1 | 1.4 | 1.6 | V |
| V_{POA} | 電源ONリセット活性閾値電圧 (注2) | 0.6 | 1.2 | 1.6 | |
| SR_{ON} | 電源投入時上昇率 | 0.01 | | | V/ms |

注: 値は指針だけです。

注1: 電圧上昇時にデバイスがリセットから開放される閾値電圧です。

注2: 供給電圧がVPOA未満でなければ電源ONリセットは動作しません。

22.5.2. VCCレベル監視器 (VLM)

表22-6. 電圧レベル監視器閾値

| シンボル | 項目 | 最小 | 代表 (注) | 最大 | 単位 |
|-------------|-------------------------------|-----|--------|-----|---------|
| V_{VLM1L} | VLM1L起動電圧 | 1.1 | 1.4 | 1.6 | V |
| V_{VLM1H} | VLM1H起動電圧 | 1.4 | 1.6 | 1.8 | |
| V_{VLM2} | VLM2起動電圧 | 2.0 | 2.5 | 2.7 | |
| V_{VLM3} | VLM3起動電圧 | 3.2 | 3.7 | 4.5 | |
| - | VLM2, VLM3 (VLM1L, VLM1H)安定時間 | | 5 (50) | | μ s |

注: 値は室温に於いてです。

22.6. アナログ比較器特性

表22-7. アナログ比較器特性 (TA=-40℃~125℃)

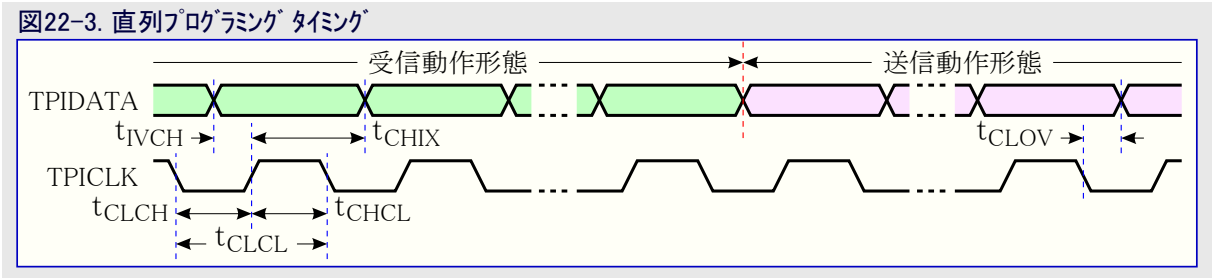
| シンボル | 項目 | 条件 | 最小 | 代表 | 最大 | 単位 |
|------------------|-----------------------------|-------------------|-----|-----|----|-----|
| V_{AIO} | 入力変位(オフセット)電圧 | VCC=5V, Vin=VCC/2 | | <10 | 40 | mV |
| I_{LAC} | 入力漏れ電流 | VCC=5V, Vin=VCC/2 | -50 | | 50 | nA |
| t_{APD} (注) | アナログ伝播遅延 (飽和から100mV過駆動へ) | VCC=2.7V | | 150 | | ns |
| | | VCC=4.0V | | 185 | | |
| | アナログ伝播遅延 (100mVの段階変化) | VCC=2.7V | | 135 | | |
| | | VCC=4.0V | | 160 | | |
| t_{DPD} | デジタル伝播遅延 | VCC=1.8~5.5V | | 1 | 2 | clk |

注: パットに対する15ns内部遅延を除く

22.7. A/D変換器特性

| 表22-8. A/D変換特性 (TA=-40℃～125℃,VCC=1.8～5.5V) | | | | | | | | |
|--|---|----------------------------|--------|---------------|-------|-------|------|-----|
| シンボル | 項目 | 条件 | | | 最小 | 代表 | 最大 | 単位 |
| | 分解能 | | | | | | 10 | ビット |
| | 絶対精度 (積分非直線性、微分非直線性、量子化、利得、オフセットの各誤差を含む) | VCC=4.3V, VREF=4.3V | 標準動作 | 変換クロック=200kHz | | 2 | | |
| | | | | 変換クロック=1MHz | | 3 | | |
| | | | 雑音低減動作 | 変換クロック=200kHz | | 1.5 | | |
| | | | | 変換クロック=1MHz | | 2.5 | | |
| INL | 積分非直線性誤差 | VCC=4.0V,VREF=4.0V | | 変換クロック=200kHz | 0.51 | 0.68 | 0.88 | |
| | | | | 変換クロック=1MHz | 0.39 | 0.62 | 0.92 | |
| DNL | 微分非直線性誤差 | | | 変換クロック=200kHz | 0.42 | 0.49 | 0.73 | |
| | | | | 変換クロック=1MHz | 0.22 | 0.48 | 0.55 | |
| | 利得誤差 | VCC=4.0V, 変換クロック=200kHz | | VREF=4.0V | -7.2 | -4 | -1 | LSB |
| | | | | VREF=内部1.1V | -41.3 | -13.3 | -1.9 | |
| | | | | VREF=内部2.2V | -38.3 | -8.7 | 3.1 | |
| | | | | VREF=内部4.3V | -80.4 | -3.2 | 9.9 | |
| | | | | | | | | |
| | オフセット(ゼロ)誤差 | | | VREF=4.0V | 3.0 | 5.1 | 8 | |
| | | | | VREF=内部1.1V | -54 | 9.1 | 2 | |
| | | | | VREF=内部2.2V | 2 | 5.4 | 11 | |
| | | | | VREF=内部4.3V | 1 | 3.4 | 5.4 | |
| | 変換時間 | 単独変換動作 | | | 13 | | 260 | μs |
| | 変換クロック周波数 | | | | 50 | | 1000 | kHz |
| VIN | 入力電圧 | | | | GND | | VREF | V |
| | 入力周波数帯域 | | | | | 38.5 | | kHz |
| RAIN | アナログ入力インピーダンス | | | | | 100 | | MΩ |
| | A/D変換出力 | | | | 0 | | 1023 | LSB |

22.8. 直列プログラミング特性



| 表22-9. 直列プログラミング特性 (TA=-40℃～125℃,VCC=5V±5%,特記事項を除く) | | | | | | |
|---|--------------------|-----|----|-----|-----|--|
| シンボル | 項目 | 最小 | 代表 | 最大 | 単位 | |
| 1/tCLCL | クロック周波数 | | | 2 | MHz | |
| tCLCL | クロック周期 | 500 | | | | |
| tCHCL | クロックパルスHレベル幅 | 200 | | | | |
| tCLCH | クロックパルスLレベル幅 | 200 | | | | |
| tIVCH | クロック↑に対するデータ入力準備時間 | 50 | | | ns | |
| tCHIX | クロック↑後のデータ入力保持時間 | 100 | | | | |
| tCLOV | クロック↓後のデータ出力有効時間 | | | 200 | | |

23. 代表特性

本項内に含まれたデータは主に同じ製法と設計法の類似デバイスの特徴付けとシミュレーションに基づいています。従って、このデータはデバイスがどう反応するかについての指標として扱われるべきです。

以下の図は代表的な特性を示します。これらの図は製造中に検査されていません。特性付けの間、デバイスは検査限界よりも高い周波数で動作していますが、それらは注文コードが示すよりも高い周波数での正しい機能が保証される訳ではありません。

全ての消費電流測定は全I/Oピンを入力として設定した内部プルアップ許可で行われています。消費電流は動作電圧、動作周波数、I/Oピンの負荷、I/Oピンの切り替え速度、命令実行、周囲温度のような様々な要素の関数です。支配的な要素は動作電圧と動作周波数です。

電源幅振幅の方形波発振器がクロック源として使われていますが、**パワーダウン動作**での消費電力はクロック選択と無関係です。**ウォッチドッグタイマ**許可の**パワーダウン動作**での消費電流とウォッチドッグタイマ禁止の**パワーダウン動作**での消費電流間の違いは、ウォッチドッグタイマによって引き込んだ(消費した)差電流を表します。

容量性負荷のピンの引き込み電流は(1つのピンに対して) $C_L(\text{負荷容量}) \times V_{CC}(\text{動作電圧}) \times f_{SW}(\text{I/Oピンの平均切り替え周波数})$ として推測できます。

23.1. 活動動作消費電流

図23-1. 活動動作消費電流 対 周波数 (100kHz~1MHz, PRR=\$FF)

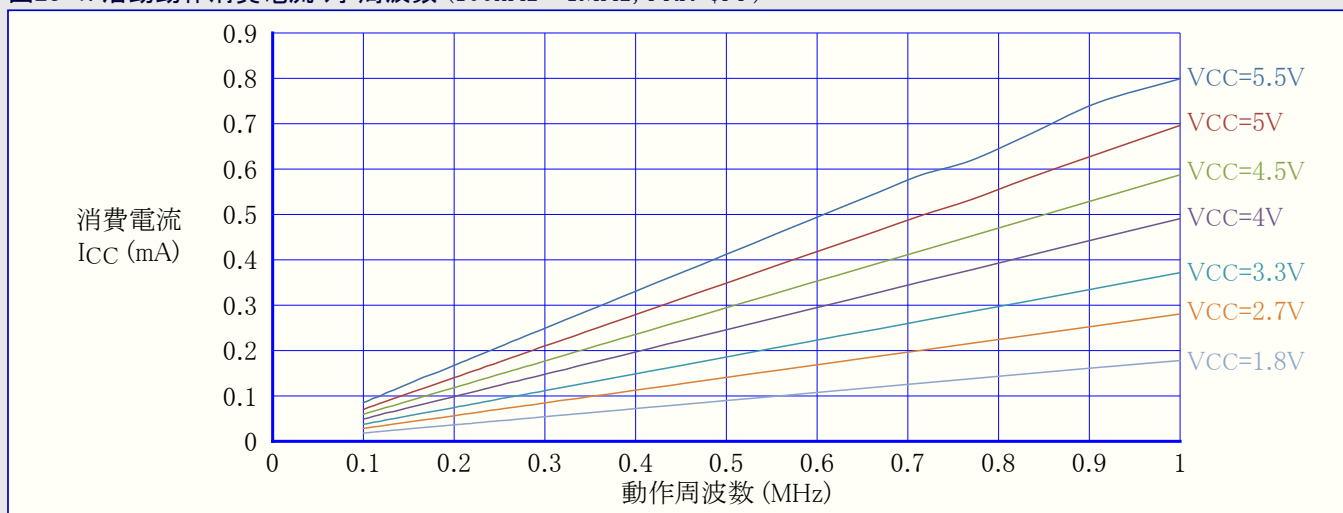


図23-2. 活動動作消費電流 対 周波数 (1MHz~12MHz, PRR=\$FF)

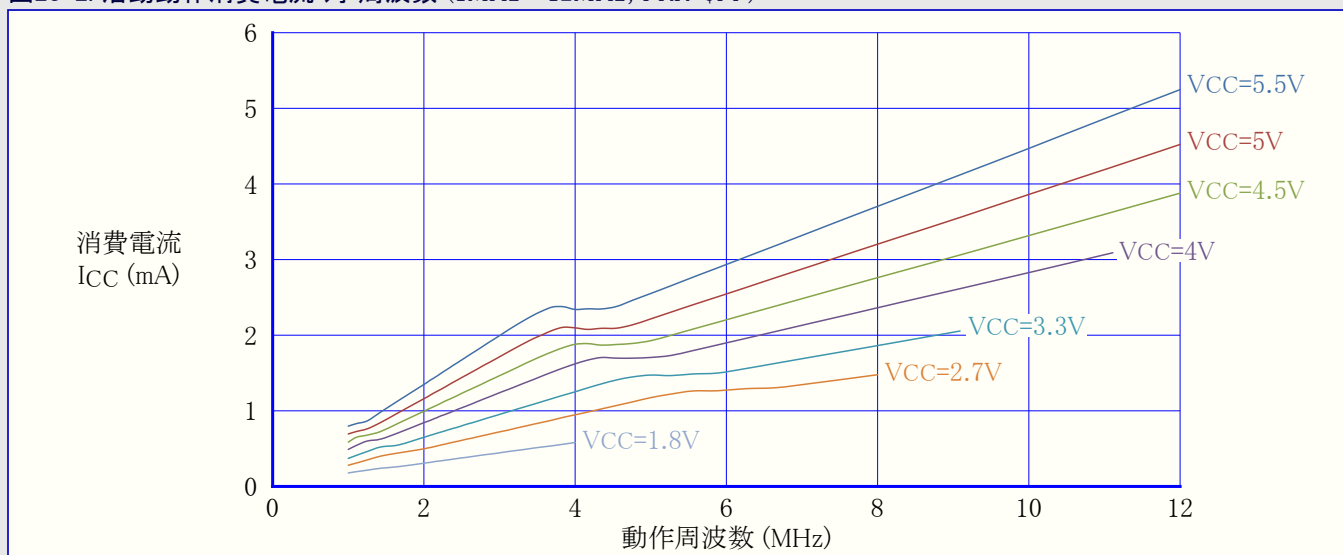


図23-3. 活動動作消費電流 対 動作電圧 (内蔵RC発振器,8MHz)

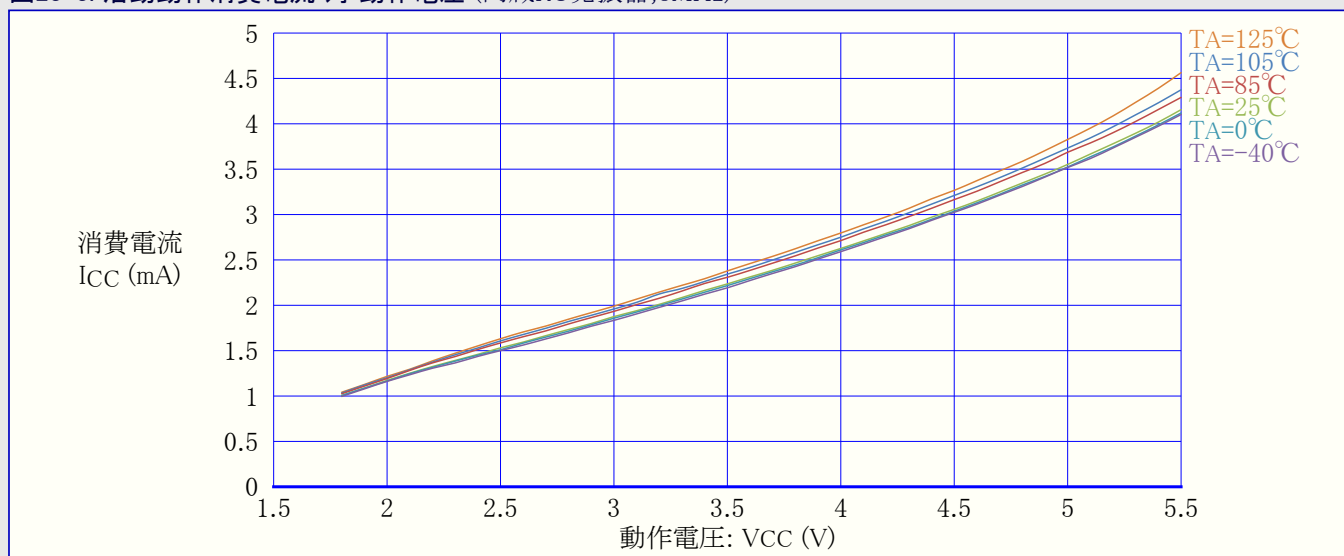


図23-4. 活動動作消費電流 対 動作電圧 (内蔵RC発振器,CKDIV8=プログラム(0),1MHz)

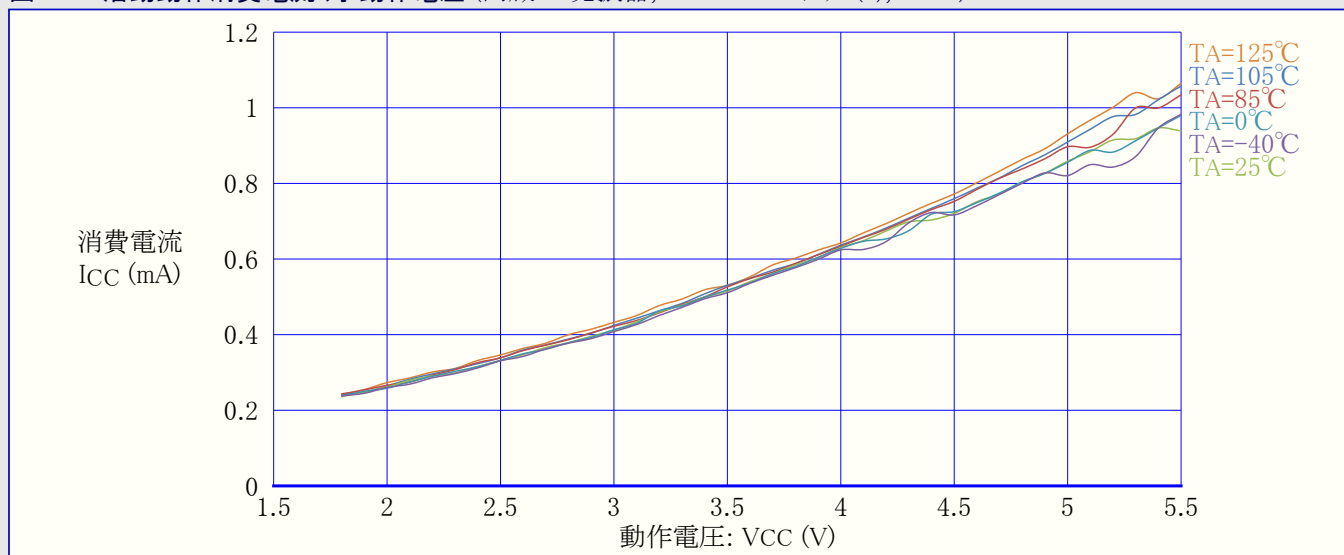
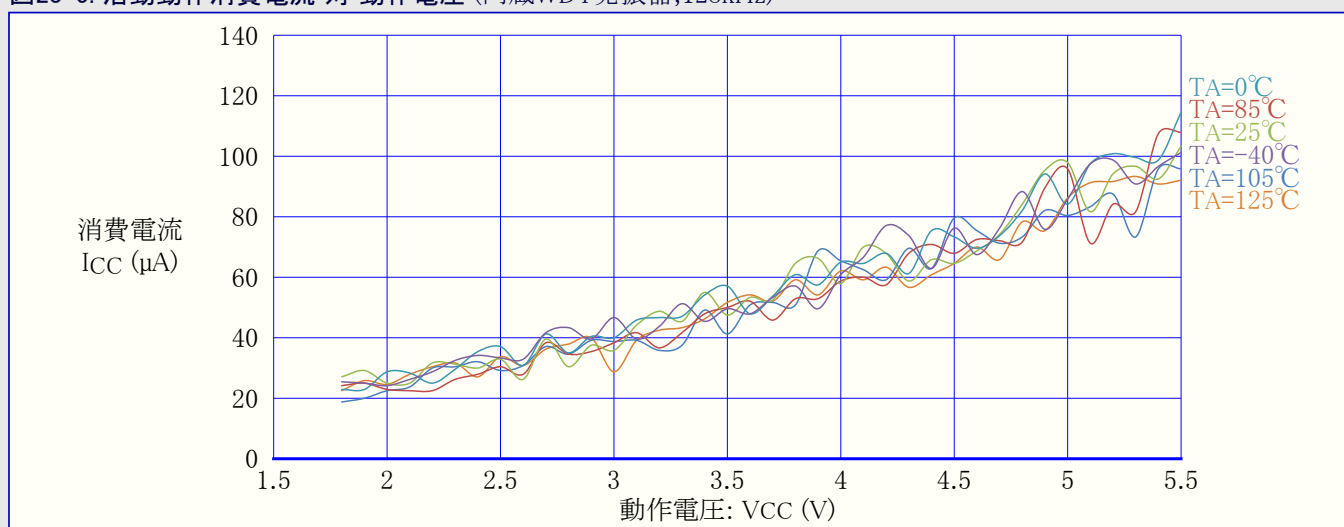


図23-5. 活動動作消費電流 対 動作電圧 (内蔵WDT発振器,128kHz)



23.2. アイドル動作消費電流

図23-6. アイドル動作消費電流 対 周波数 (100kHz~1MHz, PRR=\$FF)

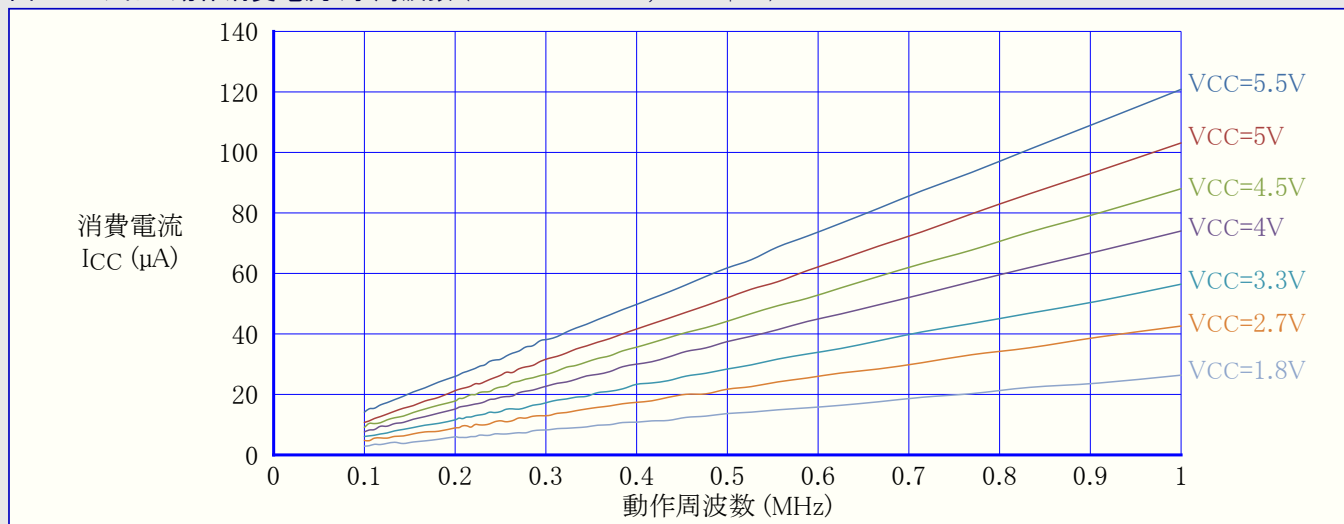


図23-7. アイドル動作消費電流 対 周波数 (1MHz~12MHz, PRR=\$FF)

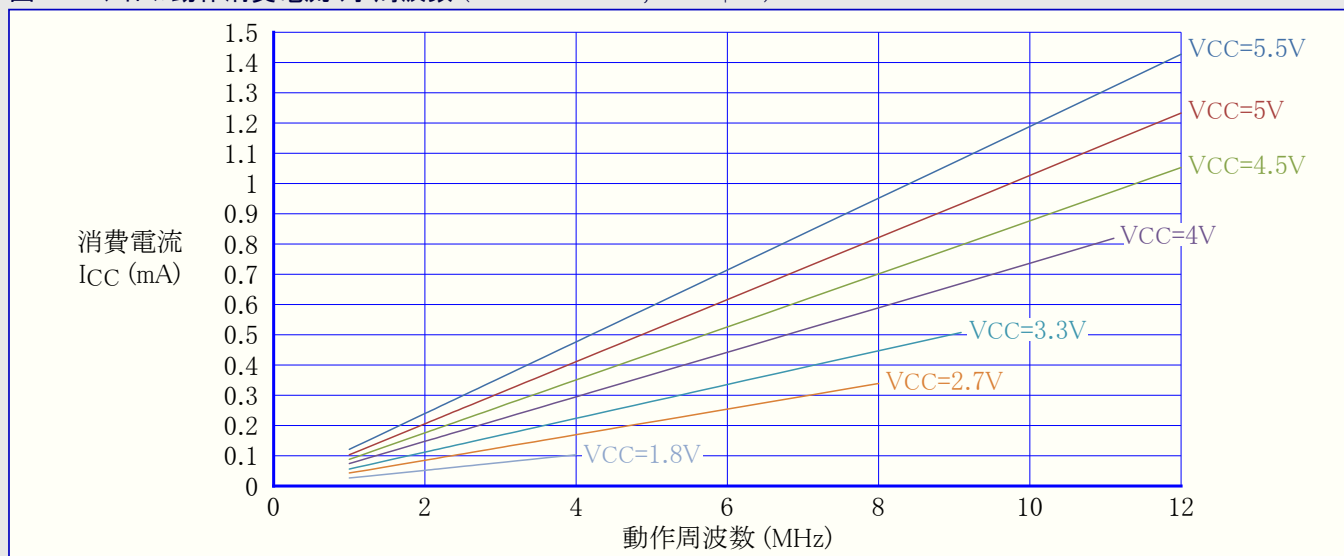


図23-8. アイドル動作消費電流 対 動作電圧 (内蔵RC発振器, 8MHz)

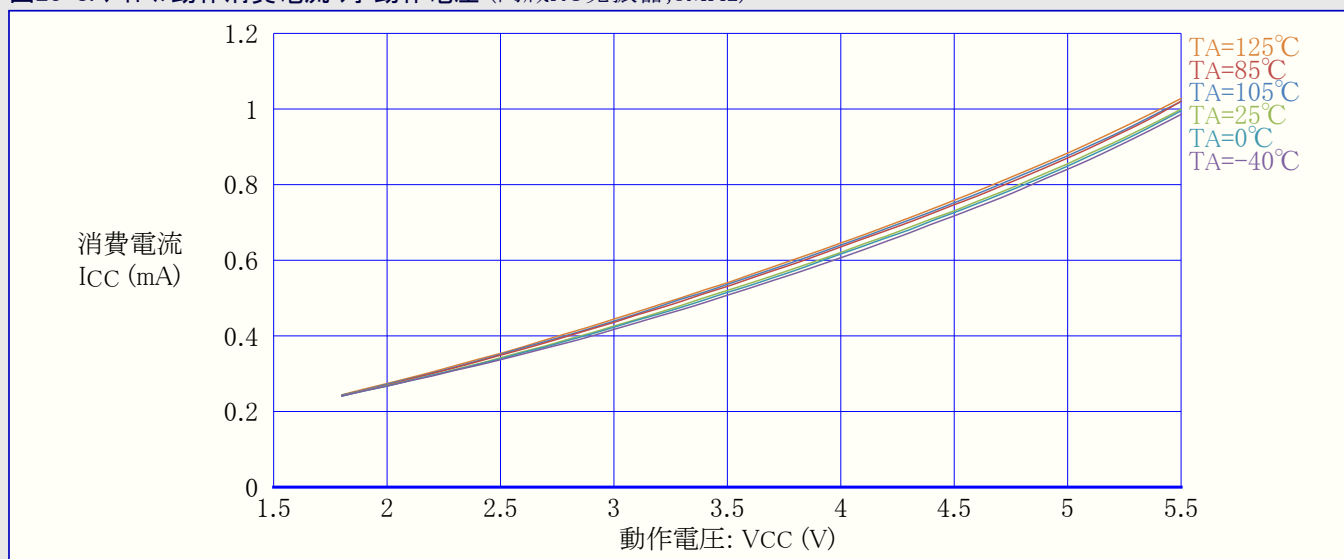
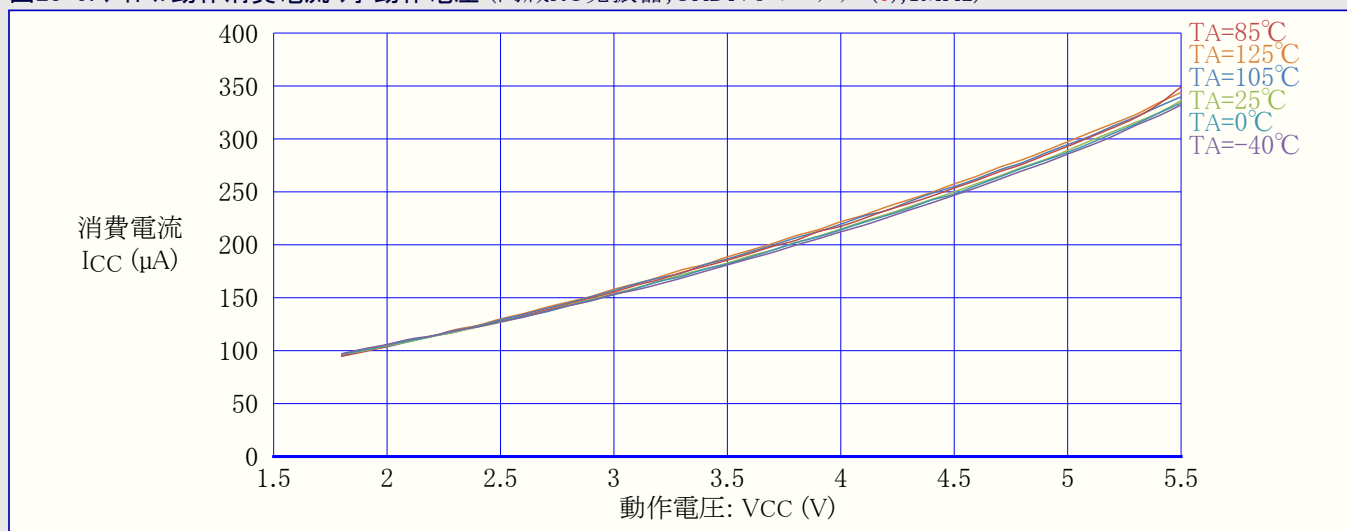


図23-9. アイドル動作消費電流 対 動作電圧 (内蔵RC発振器,CKDIV8=プログラム(0),1MHz)



23.3. 周辺機能部供給電流

以下の表と式は活動動作とアイドル動作で個別周辺機能部に対する追加消費電流の計算に使えます。周辺機能部の許可と禁止は電力削減レジスタによって制御されます。詳細については「[電力削減レジスタ](#)」をご覧ください。

右表は左表で言及したそれら以外の他の供給電圧と周波数に対して代表的な消費電流を計算するのに使うことができます。

表23-1. 各部追加消費電流 (絶対値:μA)

| PRR内ビット | 1MHz,2V | 4MHz,3V | 8MHz,5V |
|---------|---------|---------|---------|
| PRTIM0 | 6.6 | 40.0 | 153.0 |
| PRADC | 29.6 | 88.3 | 333.3 |

表23-2. 各部追加消費電流 (相対値:%)

| PRR内ビット | 活動動作(図23-1,図23-2) | アイドル動作(図23-6,図23-7) |
|---------|-------------------|---------------------|
| PRTIM0 | 2.3 | 10.4 |
| PRADC | 6.7 | 28.8 |

23.4. パワーダウン動作消費電流

図23-10. パワーダウン動作消費電流 対 動作電圧 (ウォッチドッグ タイマ禁止)

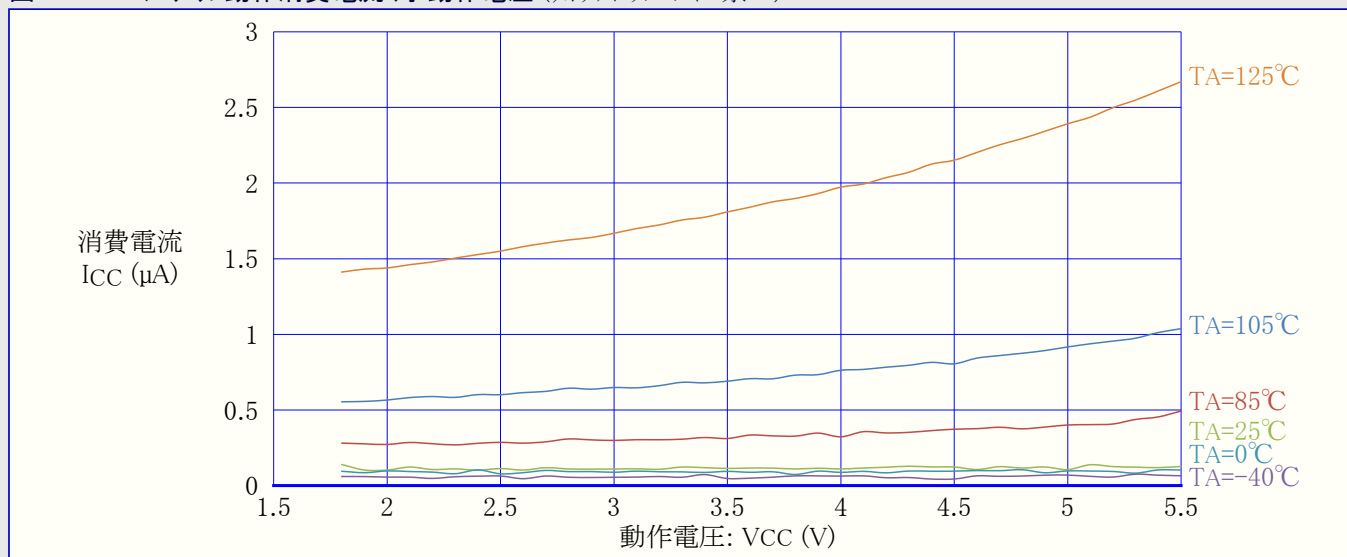
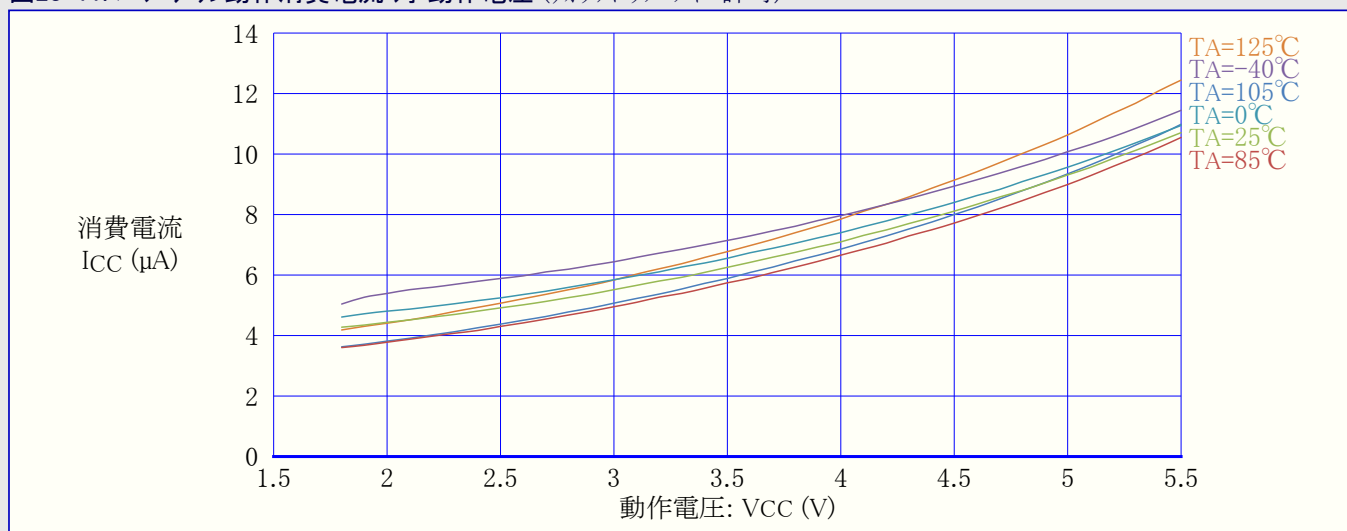


図23-11. パワーダウン動作消費電流 対 動作電圧 (ウォッチドッグ タイマ許可)



23.5. ピン駆動能力

図23-12. I/Oピン出力電圧 対 吸い込み電流 ($V_{CC}=1.8V$)

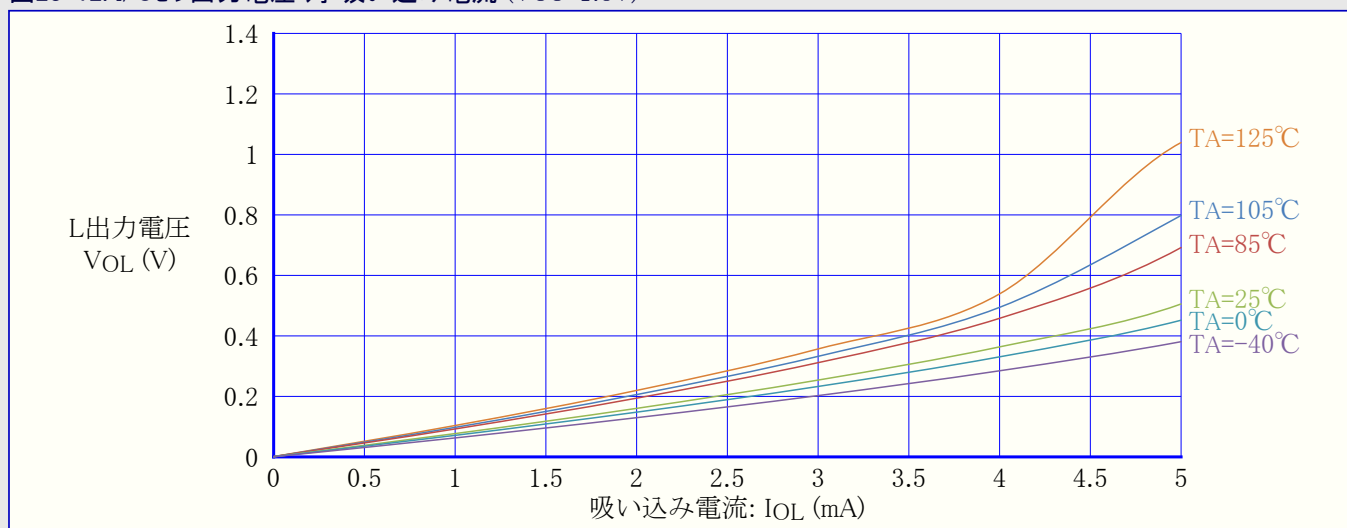


図23-13. I/Oピン出力電圧 対 吸い込み電流 ($V_{CC}=3V$)

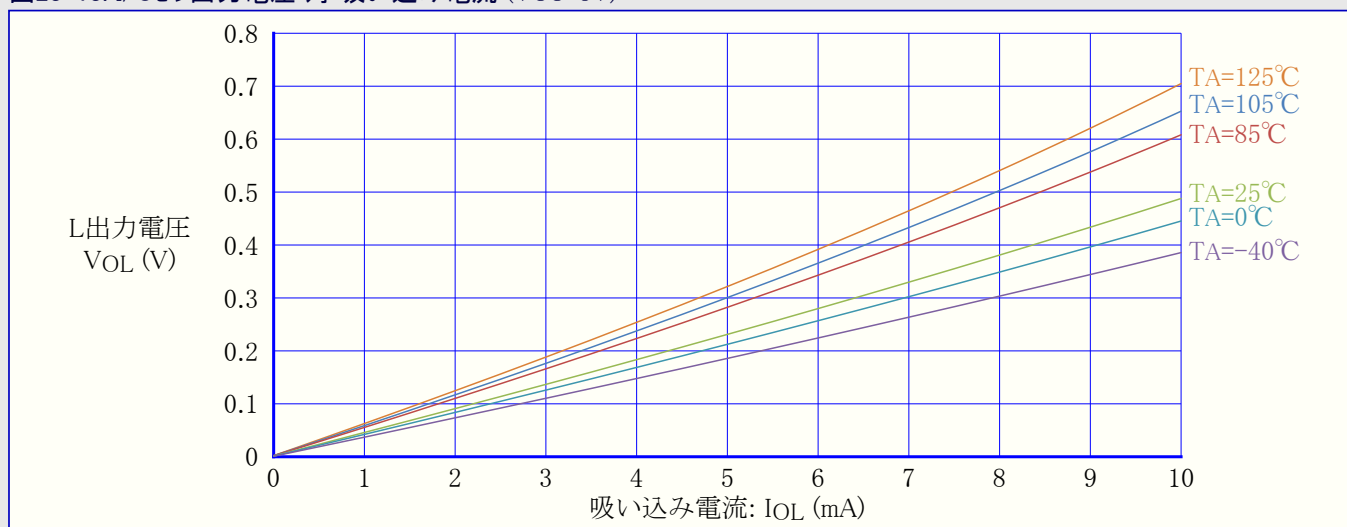


図23-14. I/Oピン出力電圧 対 吸い込み電流 ($V_{CC}=5V$)

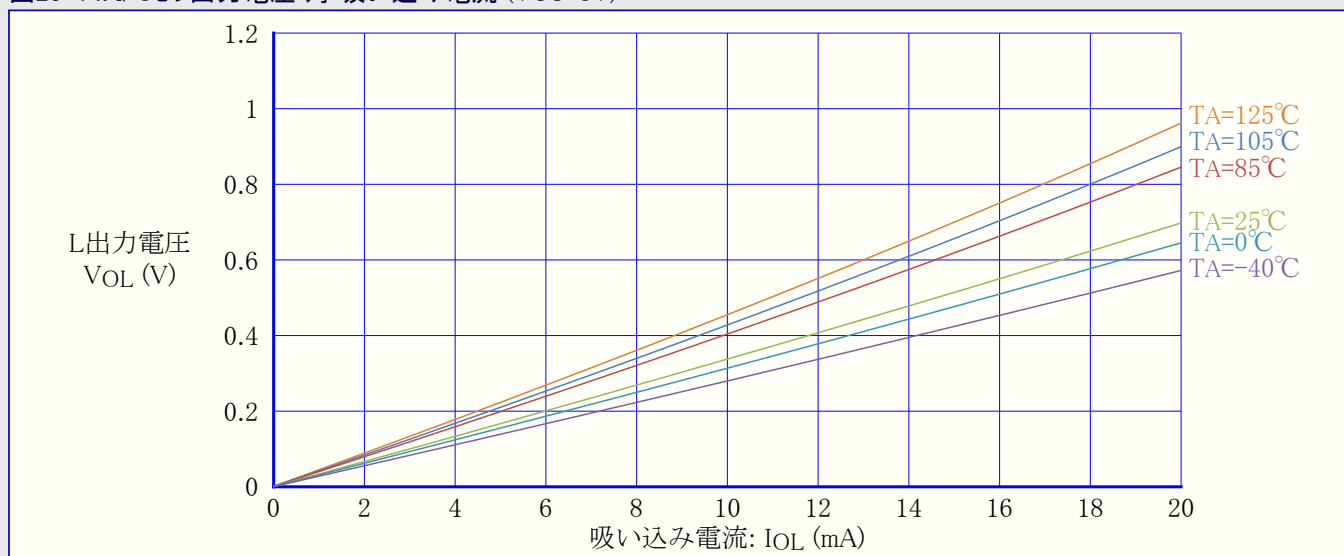


図23-15. I/Oピン出力電圧 対 吐き出し電流 ($V_{CC}=1.8V$)

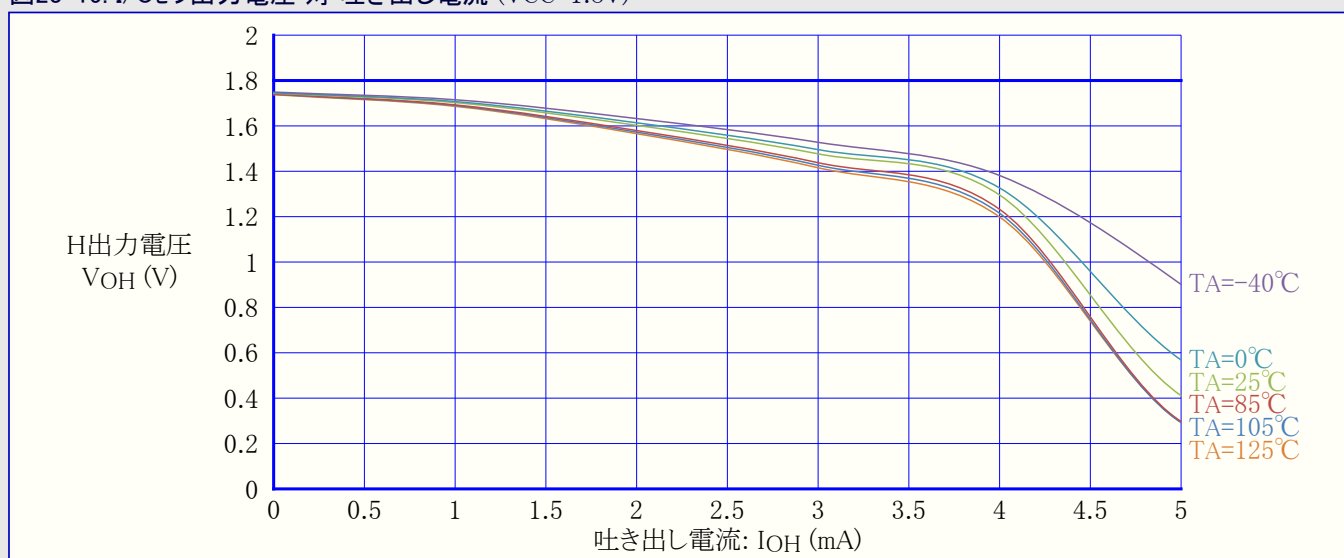


図23-16. I/Oピン出力電圧 対 吐き出し電流 ($V_{CC}=3V$)

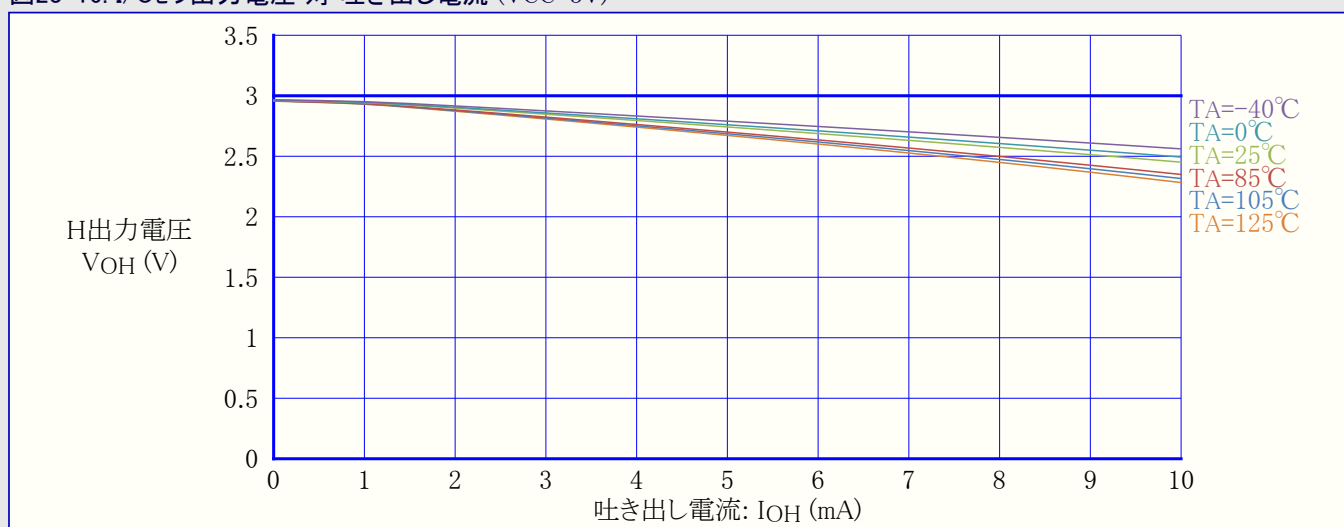


図23-17. I/Oピン出力電圧 対 吐き出し電流 (VCC=5V)

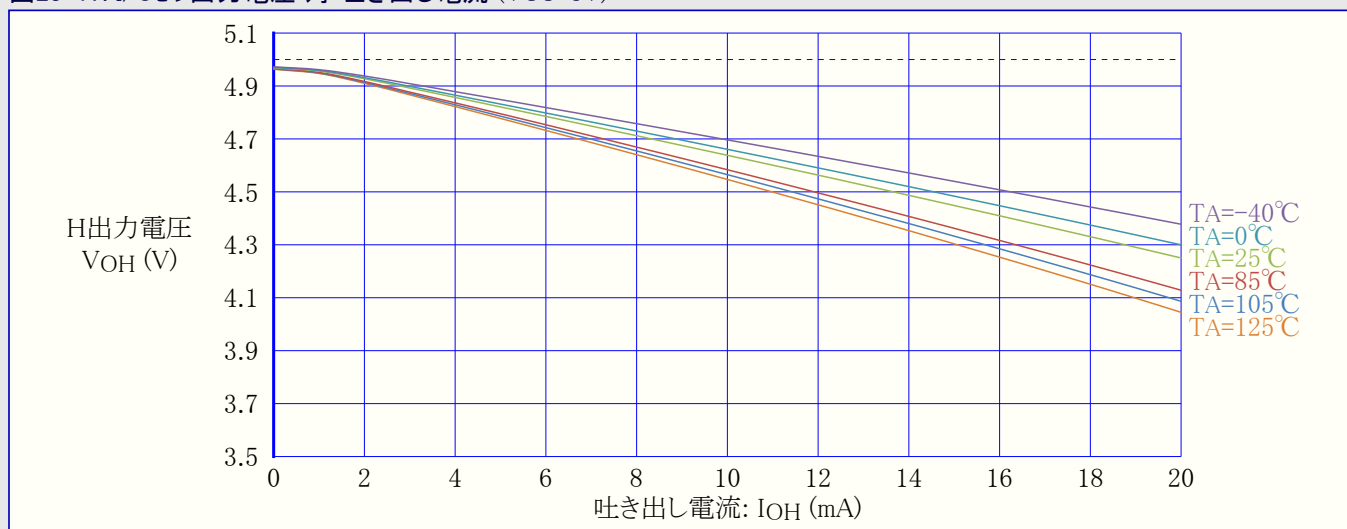


図23-18. 入出力としてのRESETピン出力電圧 対 吸い込み電流 ($T_A=25^{\circ}C$)

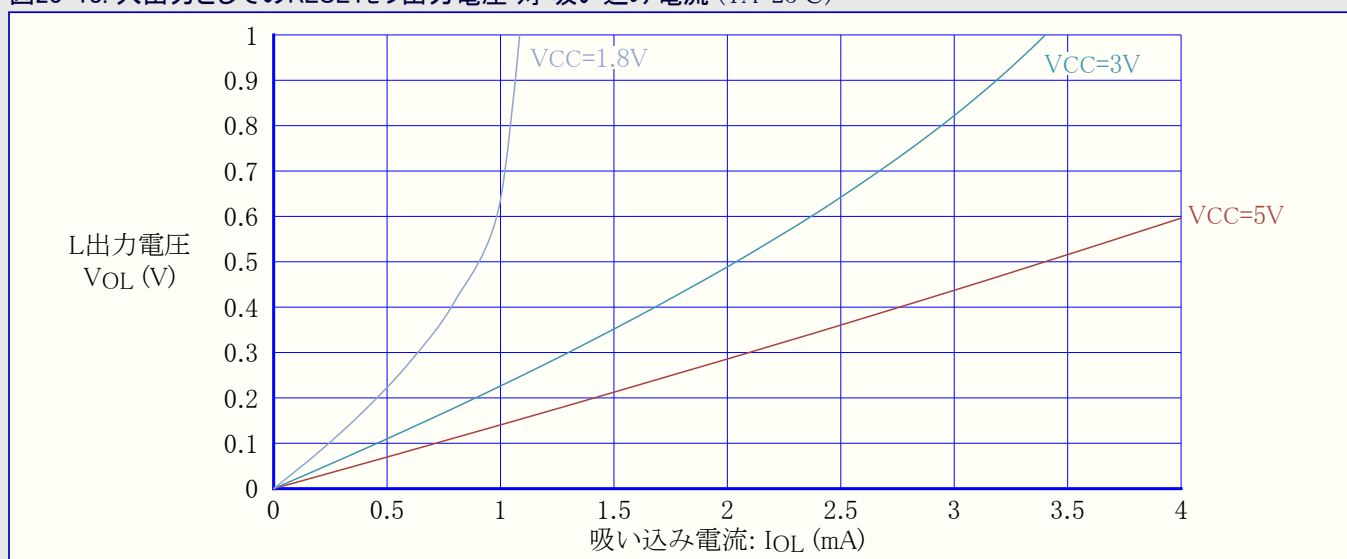
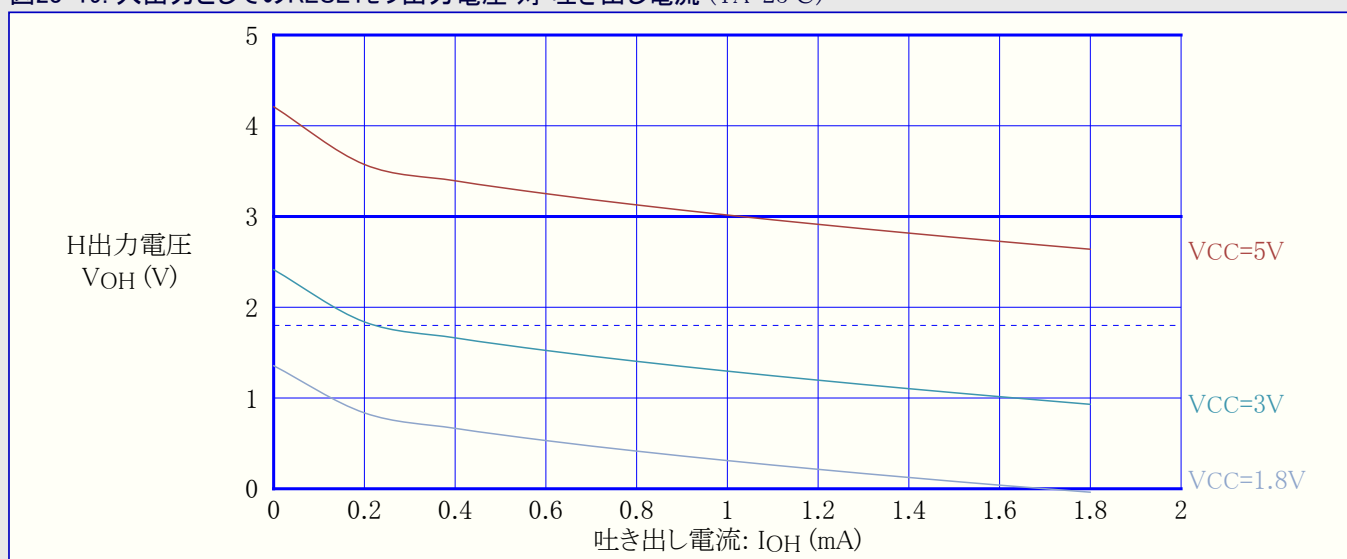


図23-19. 入出力としてのRESETピン出力電圧 対 吐き出し電流 ($T_A=25^{\circ}C$)



23.6. ピン 閾値とヒステリシス

図23-20. I/Oピン入力閾値(スレッショルド)電圧 対 動作電圧 (VIH, 1読み値)

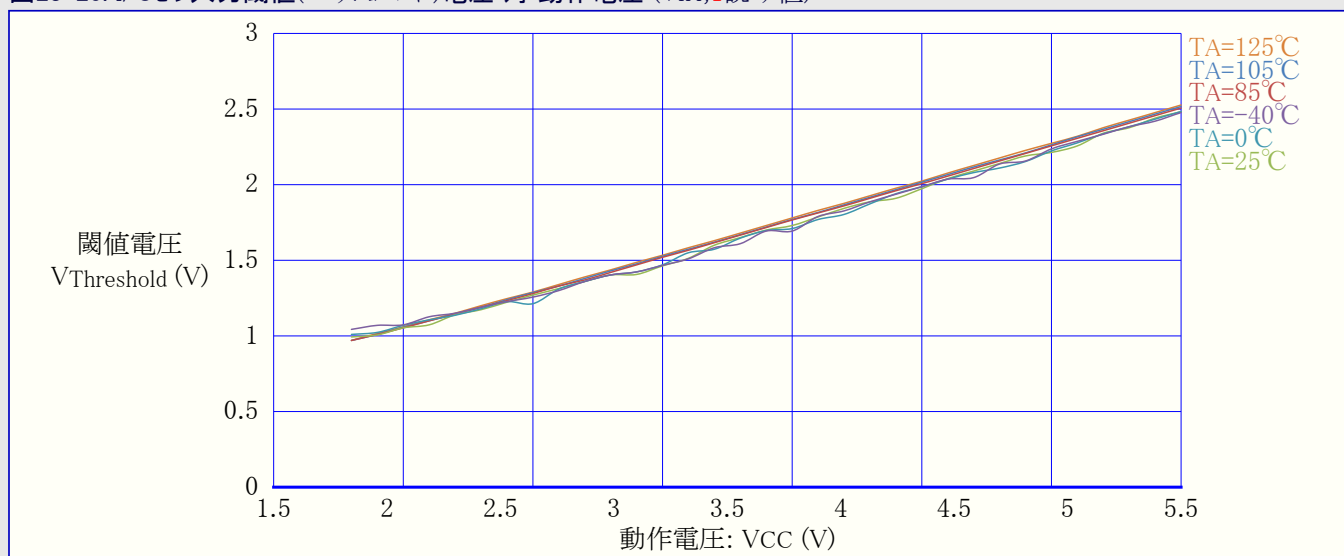


図23-21. I/Oピン入力閾値(スレッショルド)電圧 対 動作電圧 (VIL, 0読み値)

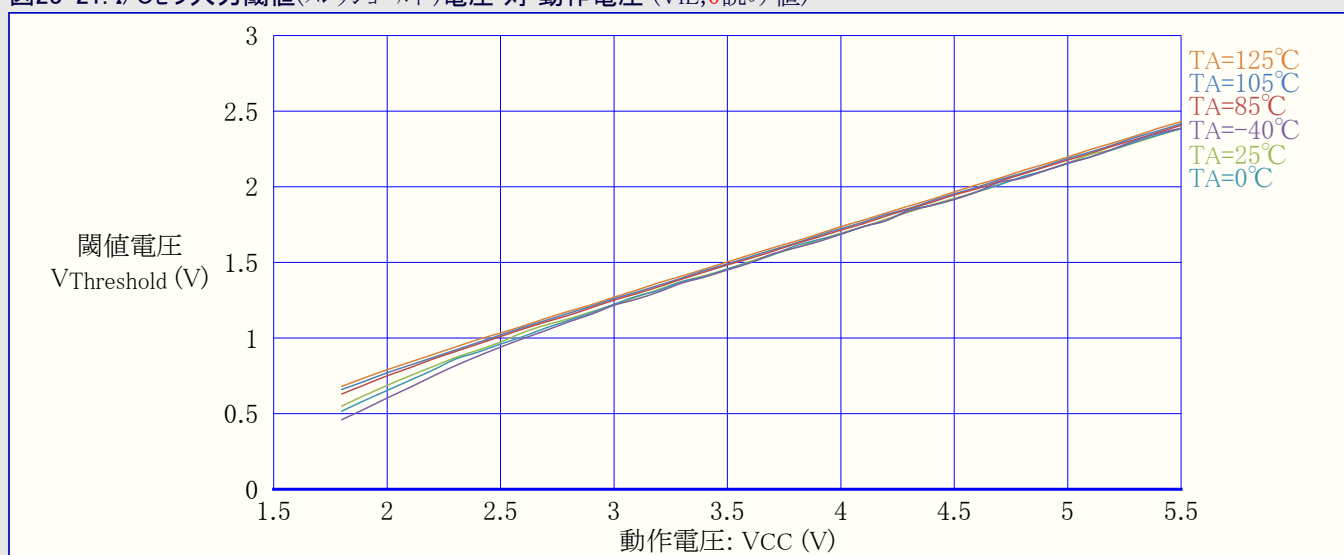


図23-22. I/Oピン入力ヒステリシス電圧 対 動作電圧

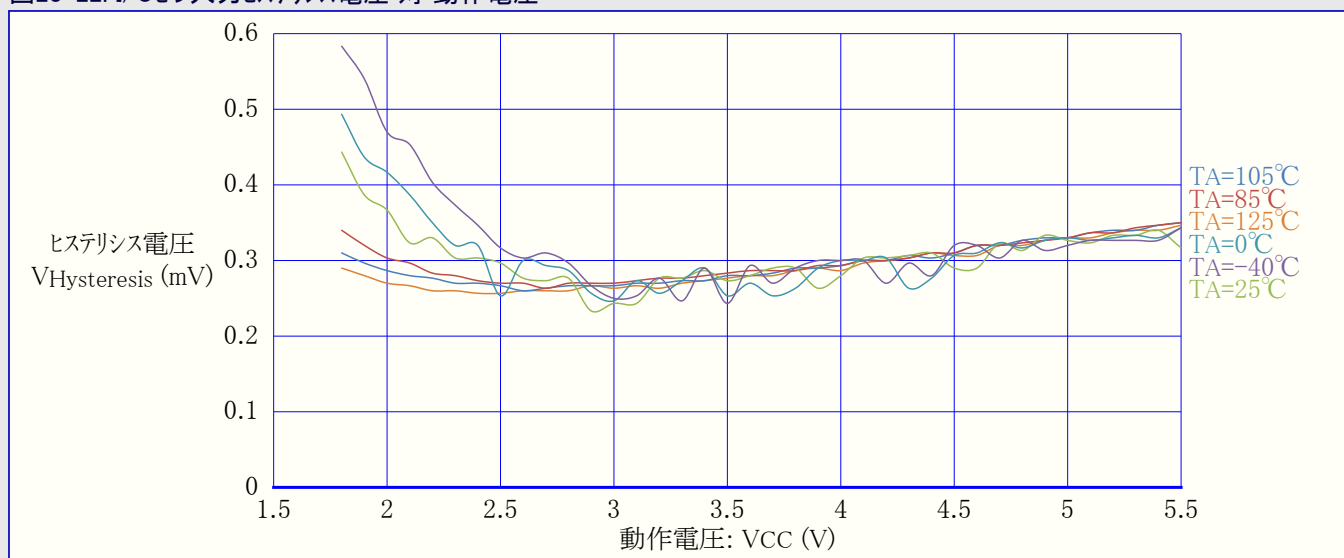


図23-23. 入出力としてのRESET入力閾値(スレッショルド)電圧 対 動作電圧 (V_{IH} , 1読み値)

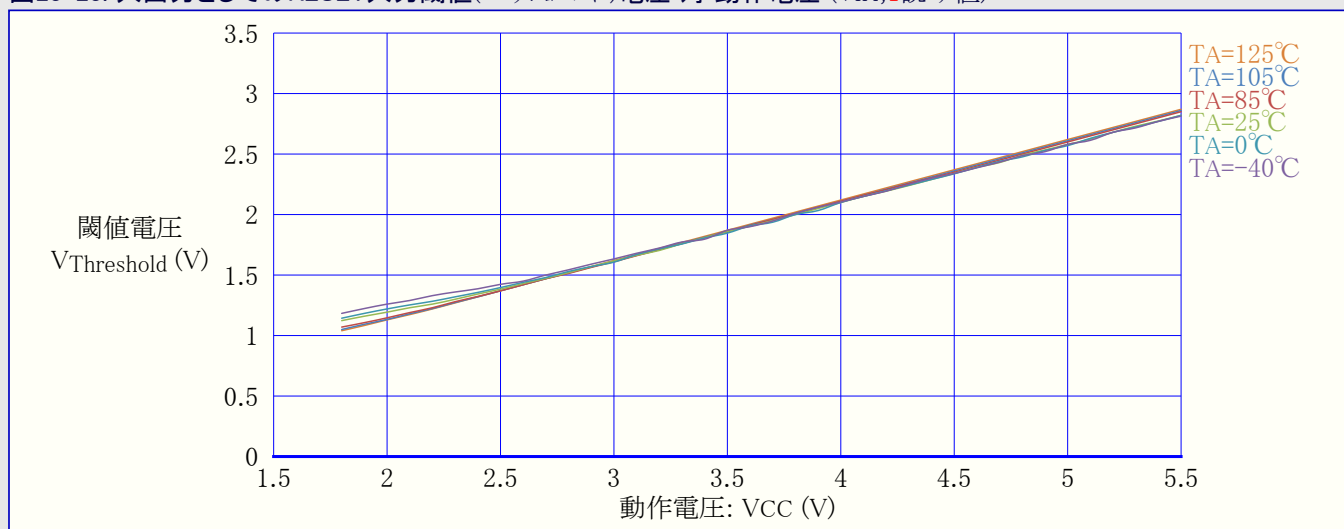


図23-24. 入出力としてのRESET入力閾値(スレッショルド)電圧 対 動作電圧 (V_{IL} , 0読み値)

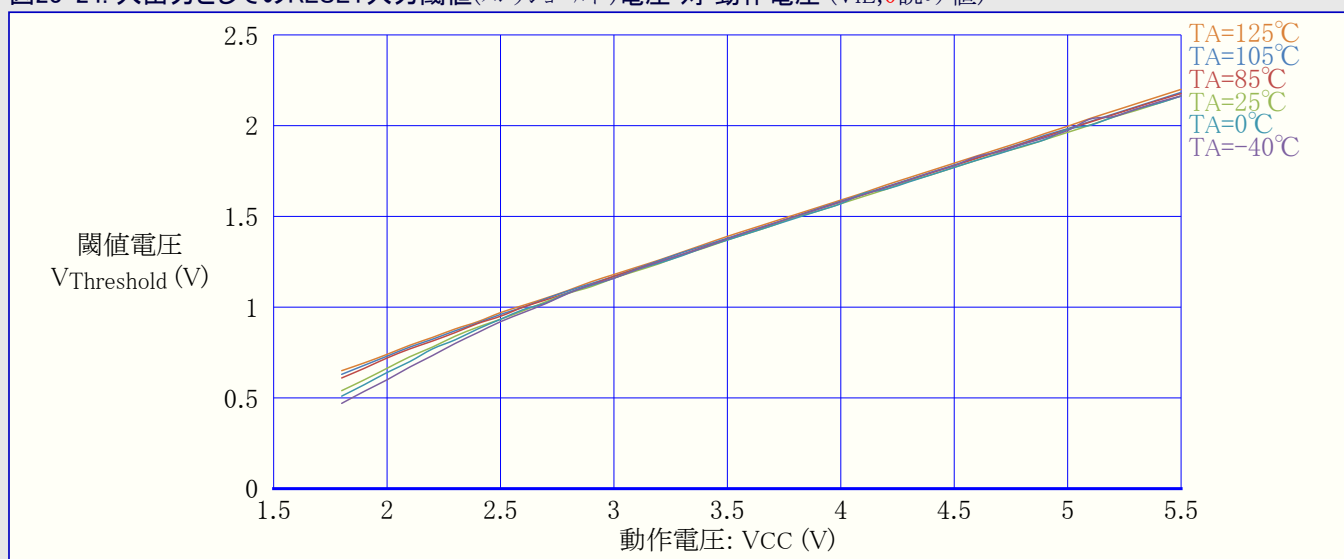


図23-25. 入出力としてのRESETピン入力ヒステリシス電圧 対 動作電圧 (I/Oとして使用時)

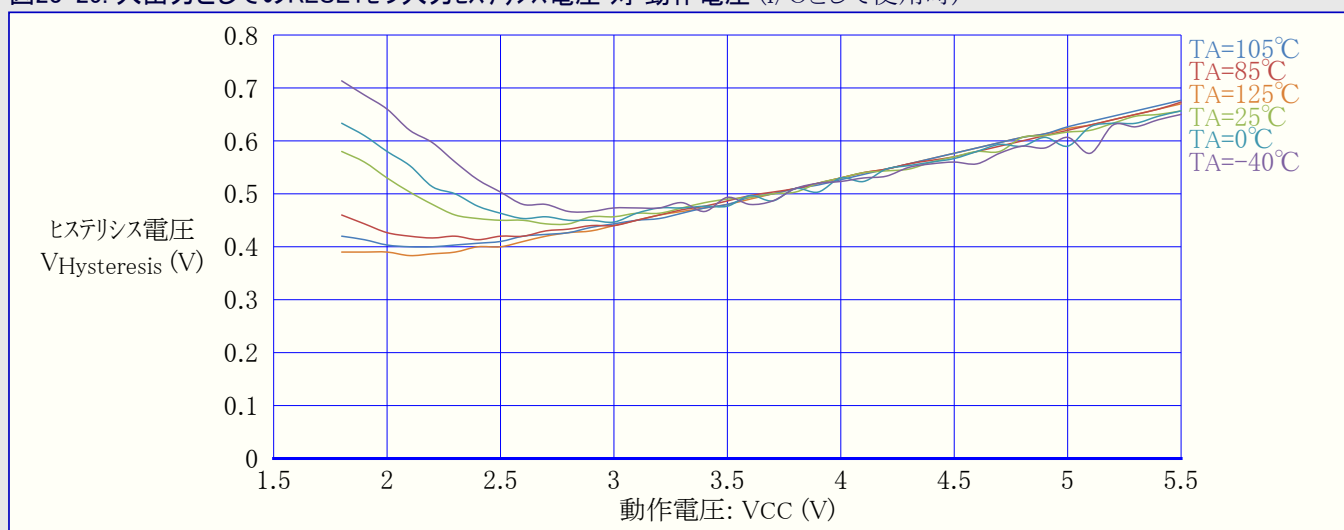


図23-26. RESETピン入力閾値(スレッショルド)電圧 対 動作電圧 ($V_{IH,I/O}$ ピン1読み値)

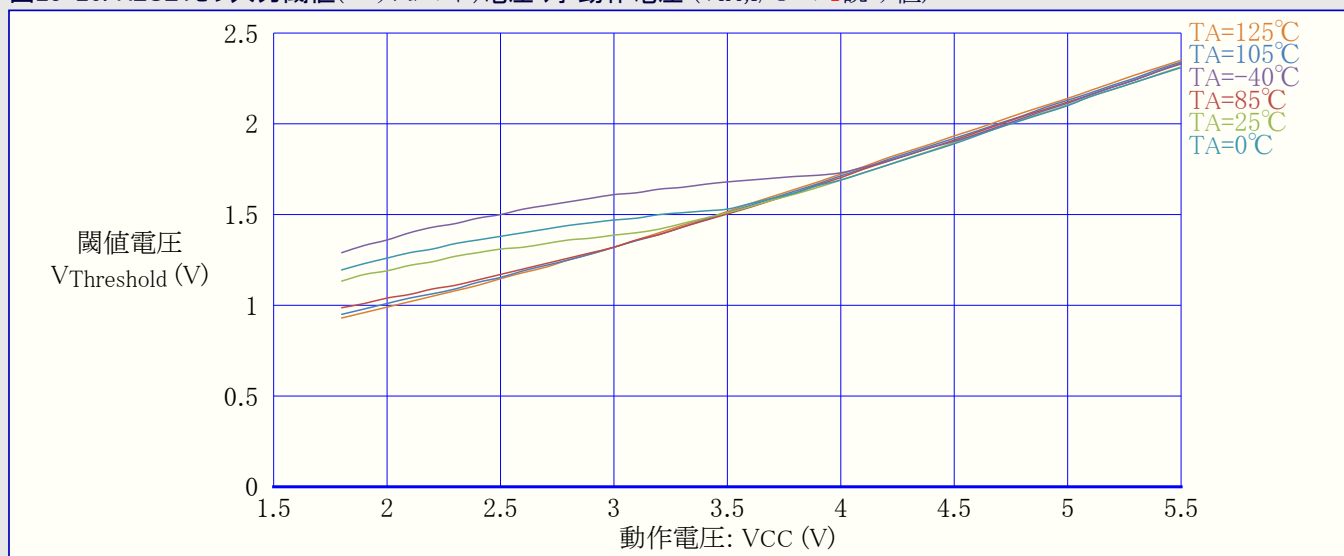


図23-27. RESETピン入力閾値(スレッショルド)電圧 対 動作電圧 ($V_{IL,I/O}$ ピン0読み値)

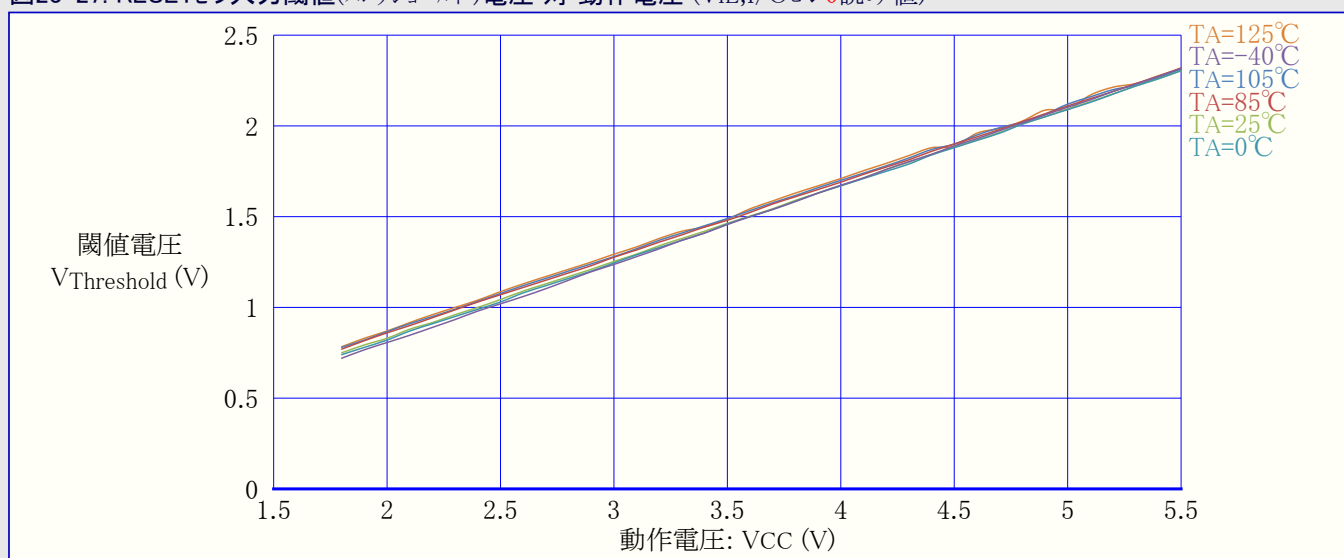
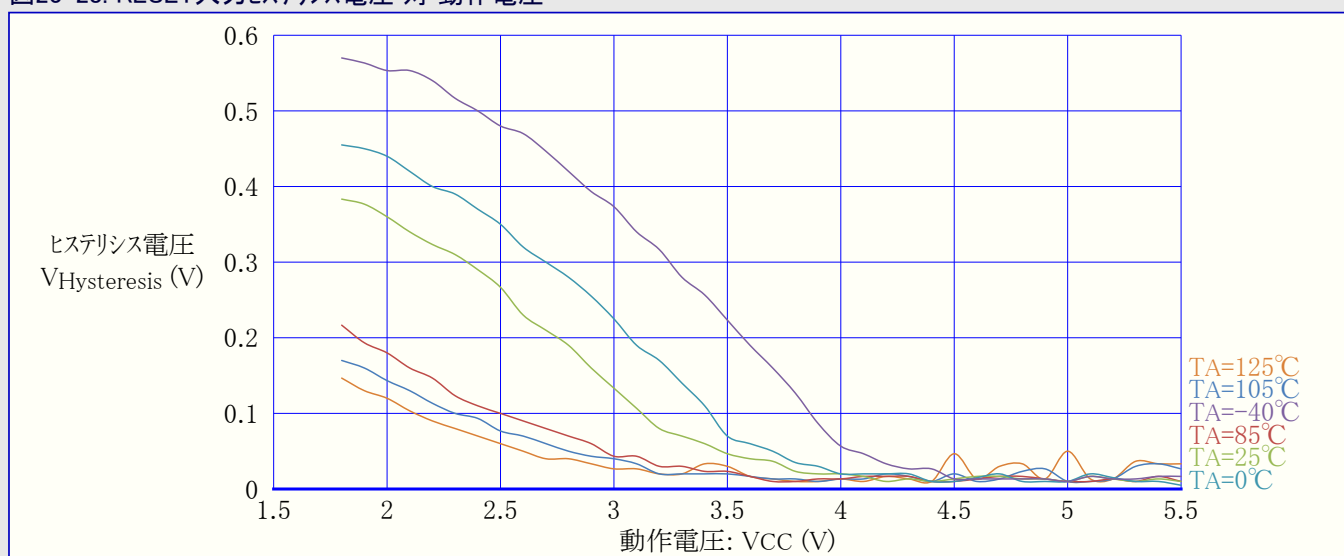
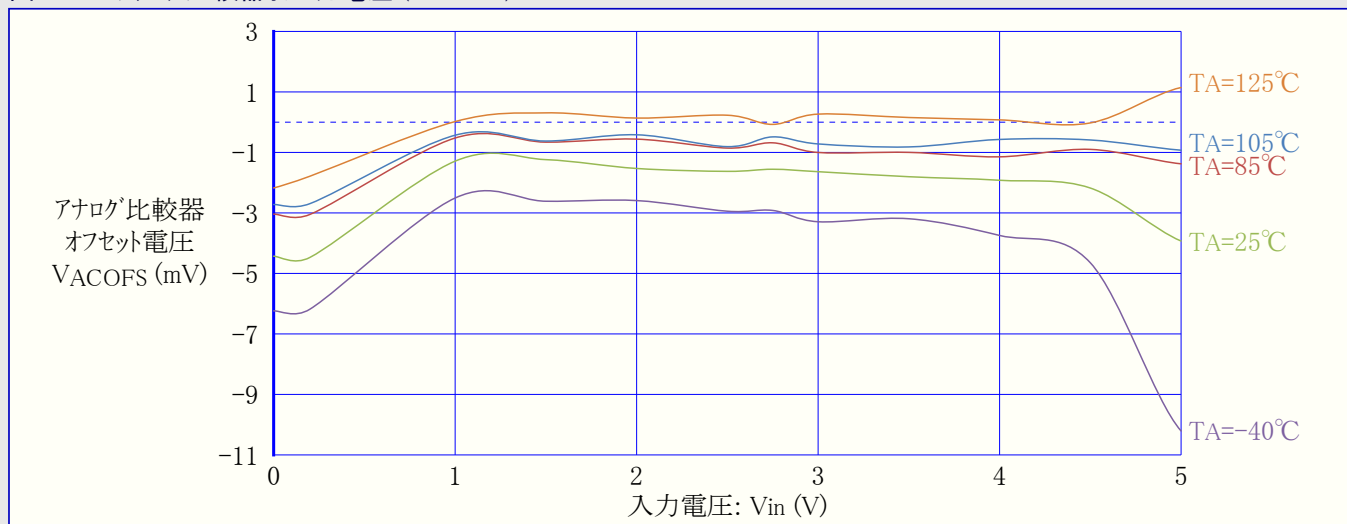


図23-28. RESET入力ヒステリシス電圧 対 動作電圧



23.7. アナログ比較器オフセット

図23-29. アナログ比較器オフセット電圧 (VCC=5V)



23.8. ピンプルアップ

図23-30. I/Oピンプルアップ抵抗電流 対 入力電圧 (VCC=1.8V)

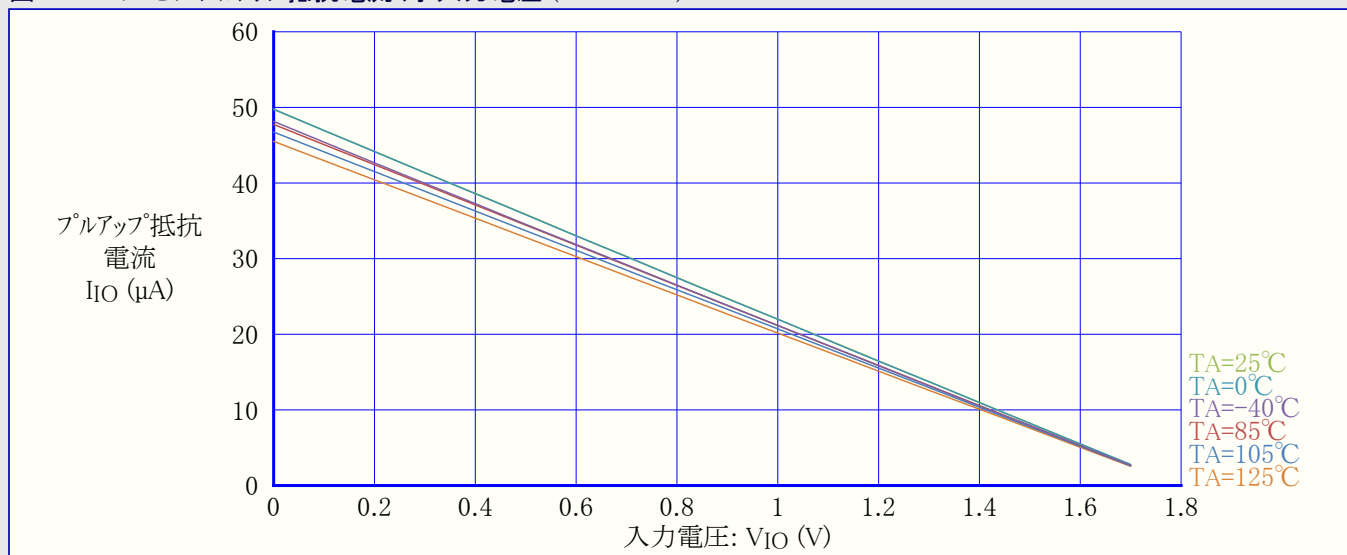


図23-31. I/Oピンプルアップ抵抗電流 対 入力電圧 (VCC=2.7V)

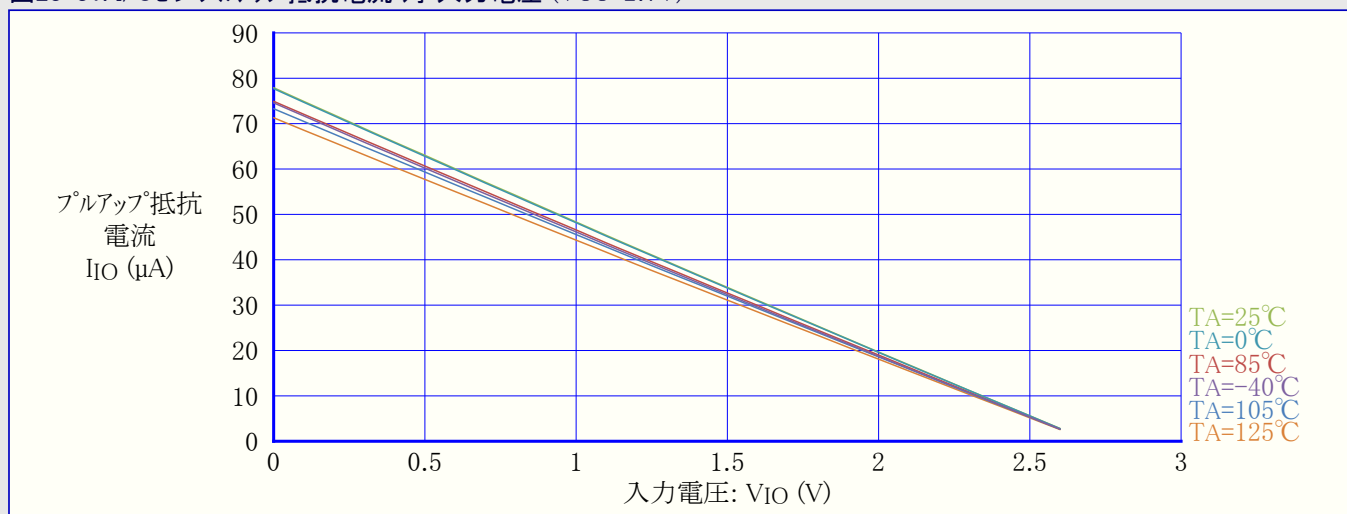


図23-32. I/Oピンプルアップ抵抗電流 対 入力電圧 (VCC=5V)

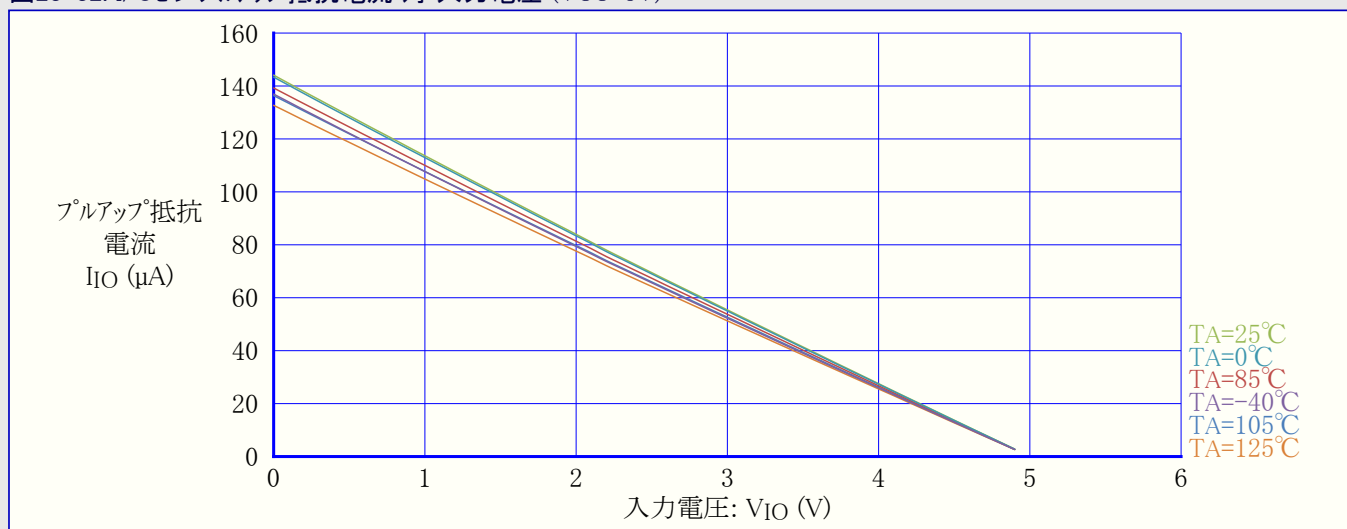


図23-33. RESETプルアップ抵抗電流 対 入力電圧 (VCC=1.8V)

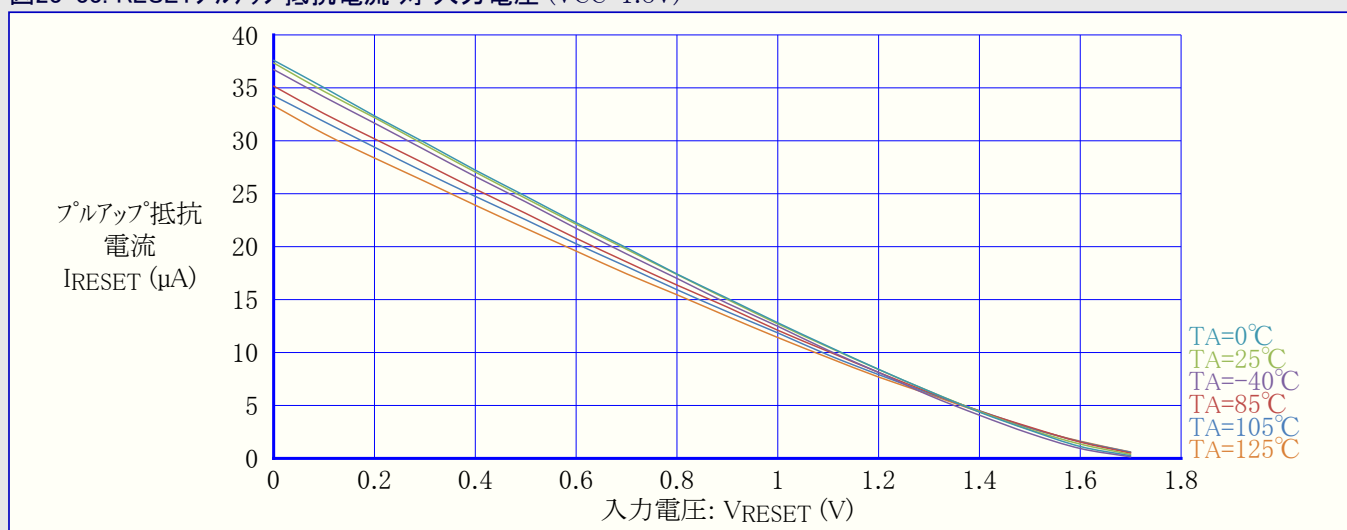


図23-34. RESETプルアップ抵抗電流 対 入力電圧 (VCC=2.7V)

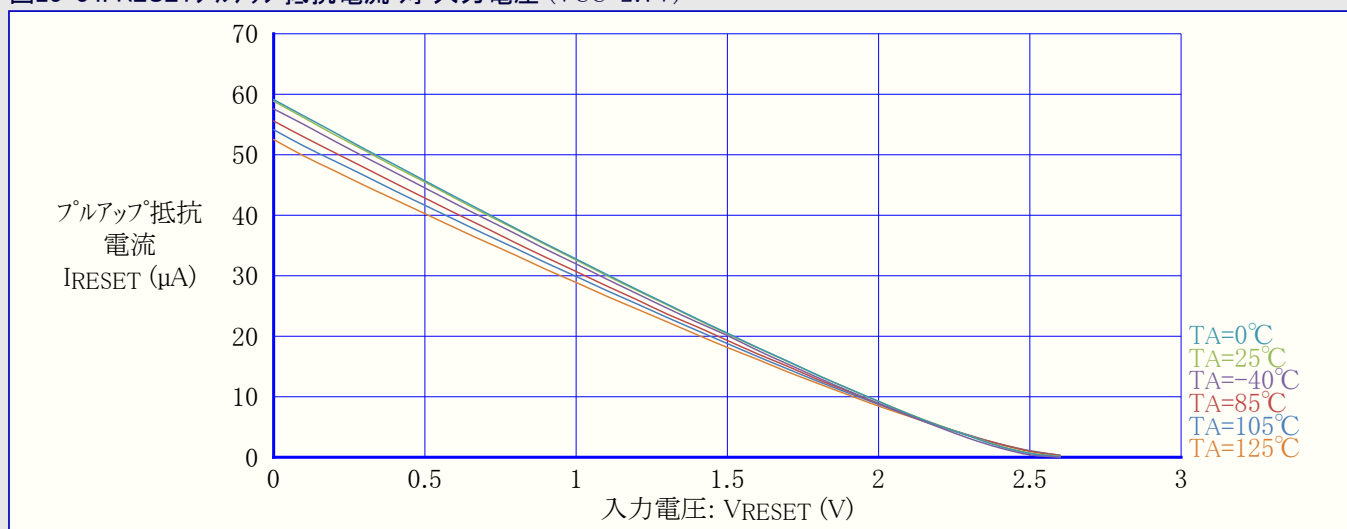
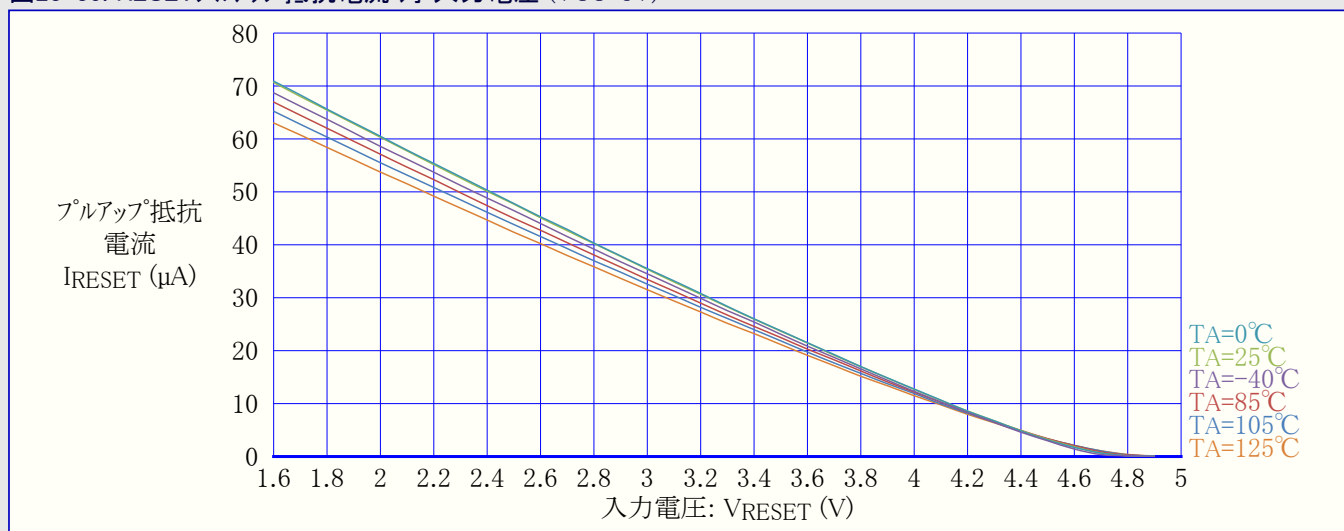


図23-35. RESETフルアップ抵抗電流 対 入力電圧 (VCC=5V)



23.9. 内部発振器周波数

図23-36. ウォッチドッグ発振器周波数 対 動作電圧

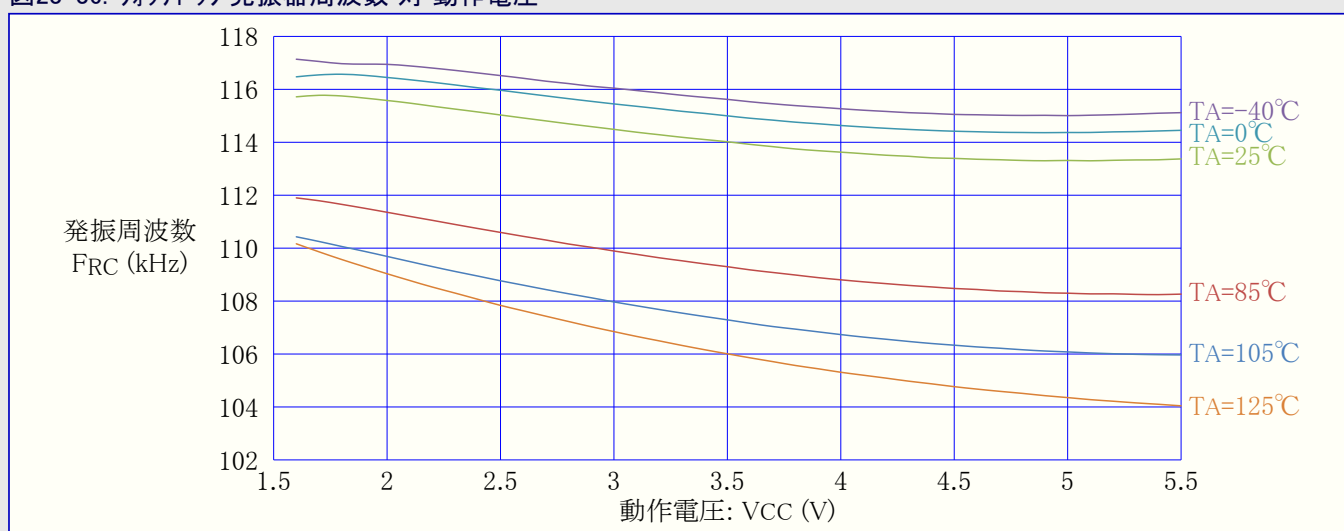


図23-37. ウォッチドッグ発振器周波数 対 動作温度

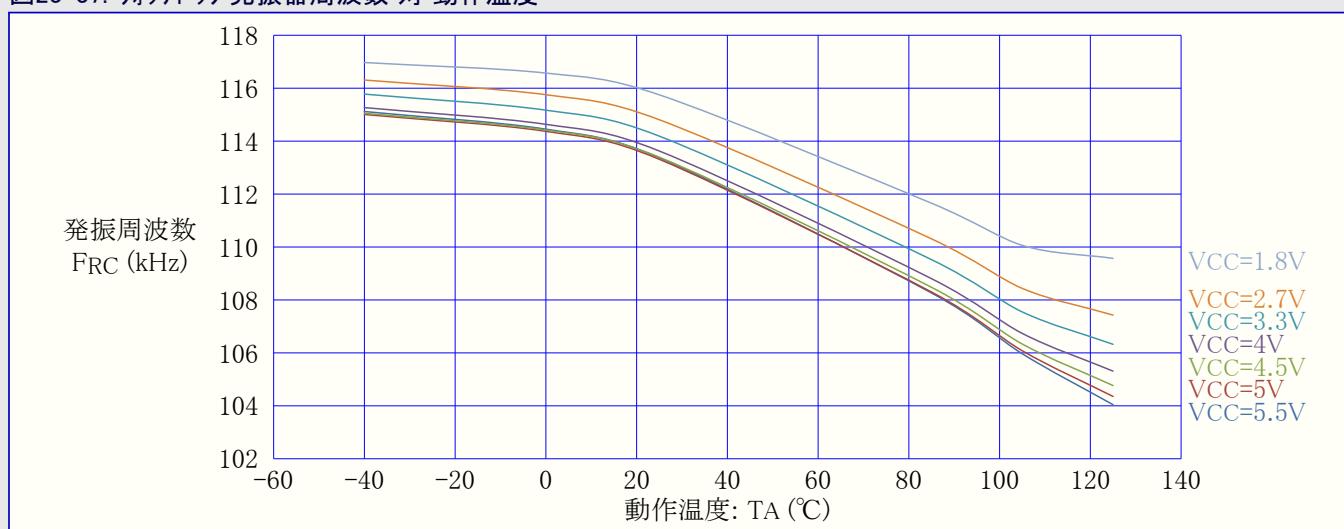


図23-38. 校正付き8MHz内蔵RC発振器周波数 対 動作電圧

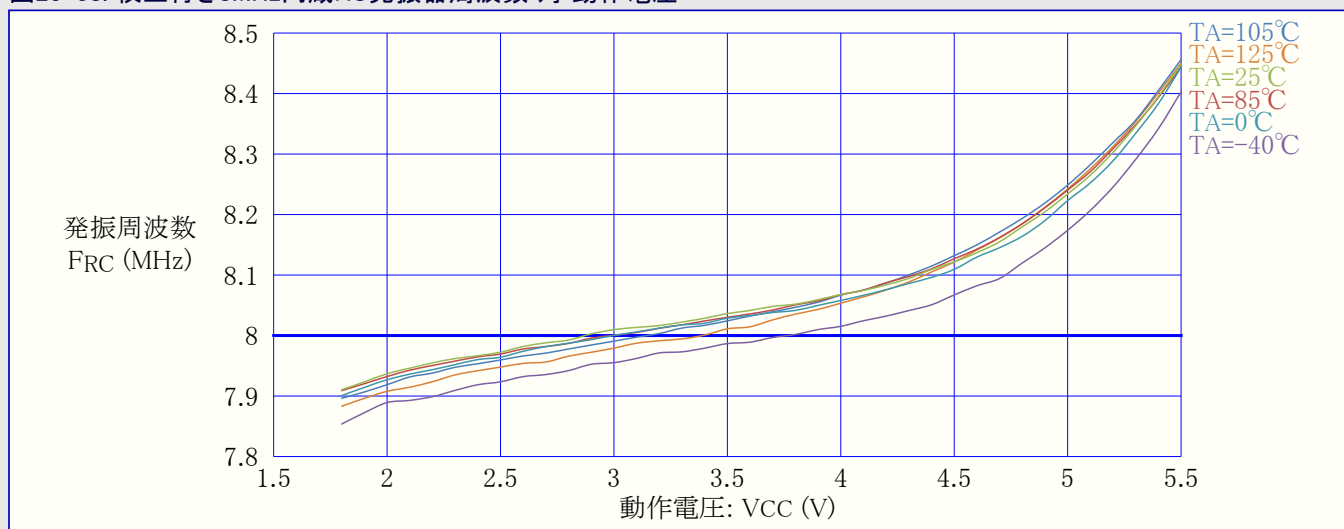


図23-39. 校正付き8MHz内蔵RC発振器周波数 対 動作温度

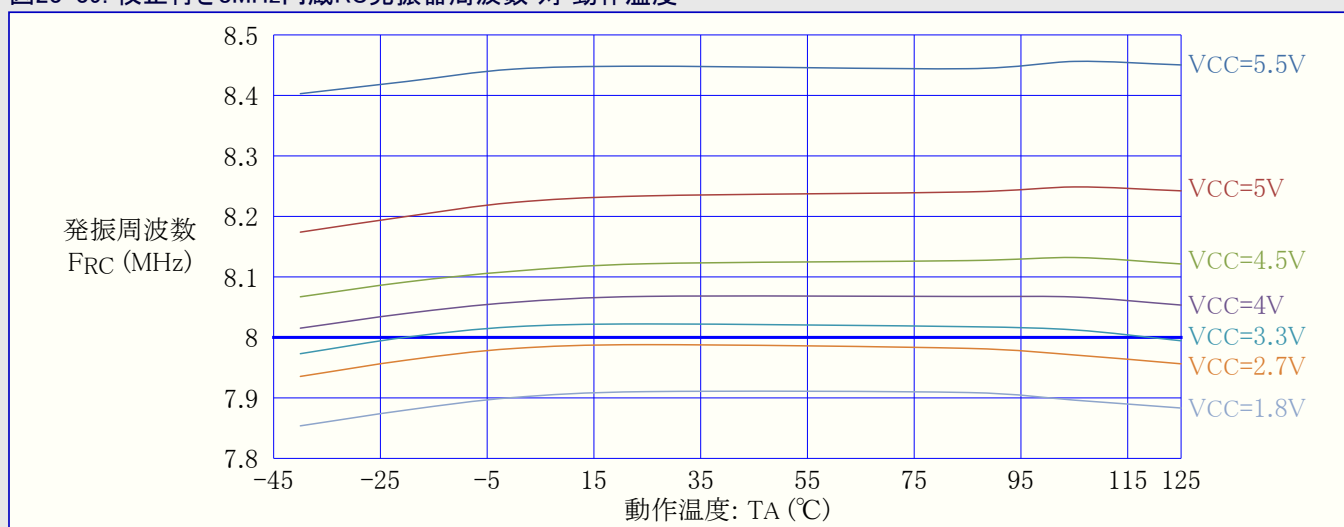
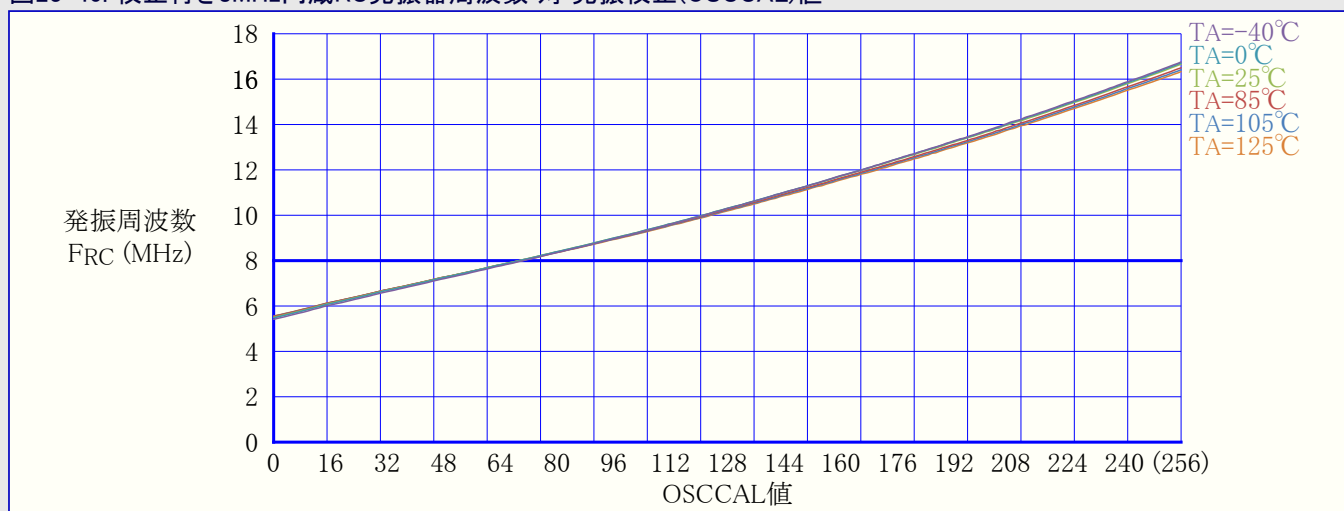


図23-40. 校正付き8MHz内蔵RC発振器周波数 対 発振校正(OSCCAL)値



23.10. VLM閾値

図23-41. VCCレベル監視器VLM1L閾値電圧 対 動作温度 (VLM2~0=001)

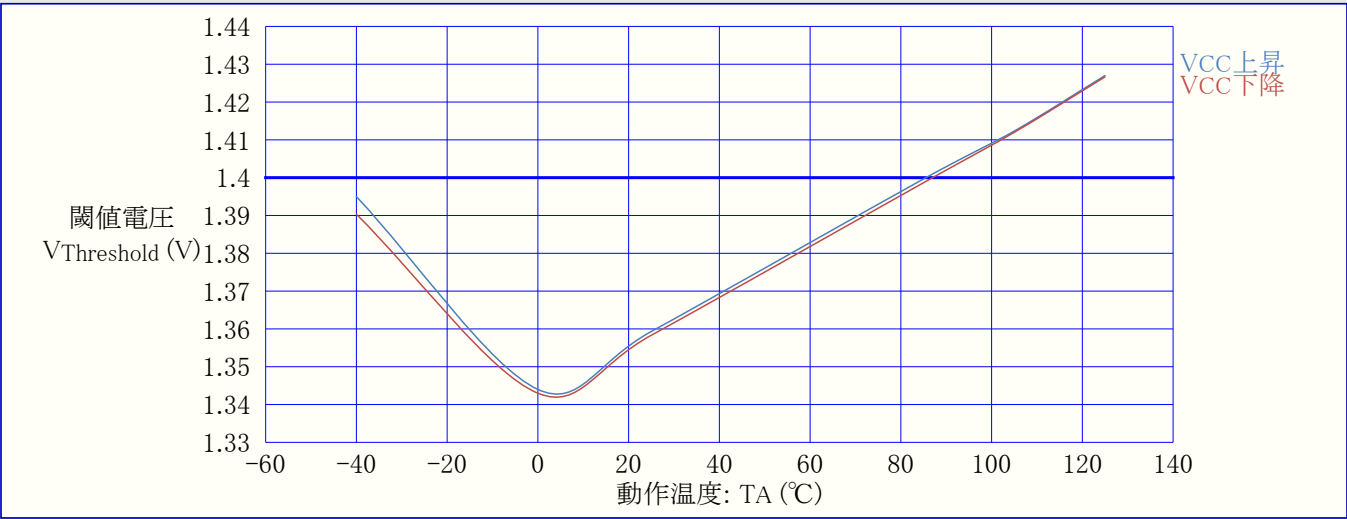


図23-42. VCCレベル監視器VLM1H閾値電圧 対 動作温度 (VLM2~0=010)

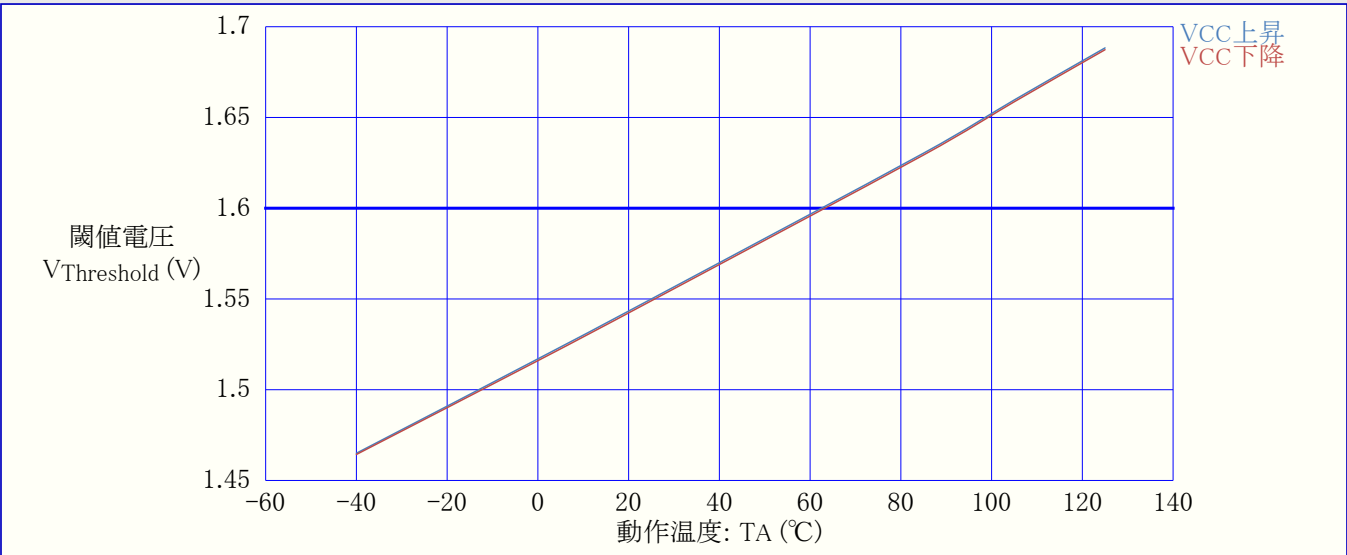


図23-43. VCCレベル監視器VLM2閾値電圧 対 動作温度 (VLM2~0=011)

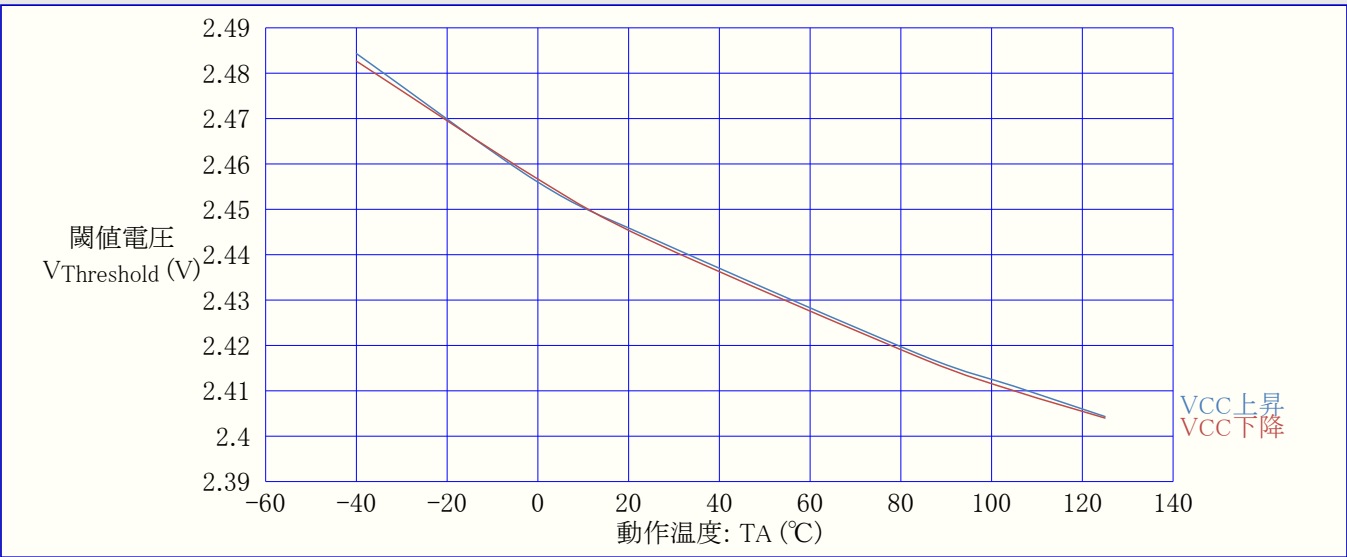
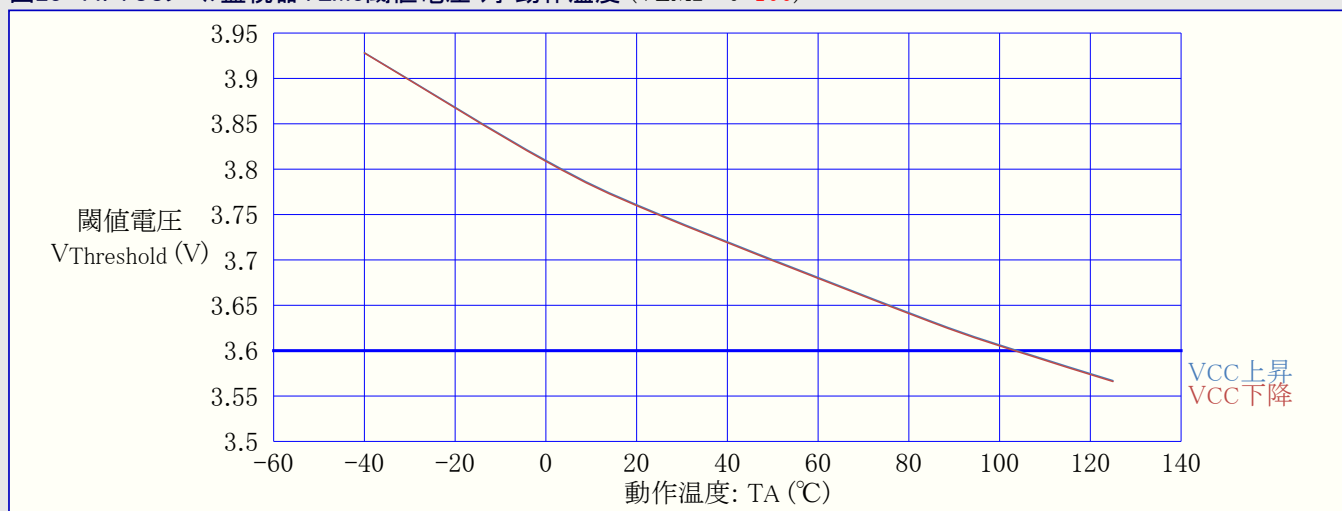


図23-44. VCCレベル監視器VLM3閾値電圧 対 動作温度 (VLM2~0=100)



23.11. 周辺機能部消費電流

図23-45. A/D変換器消費電流 対 動作電圧 (周波数4MHz)

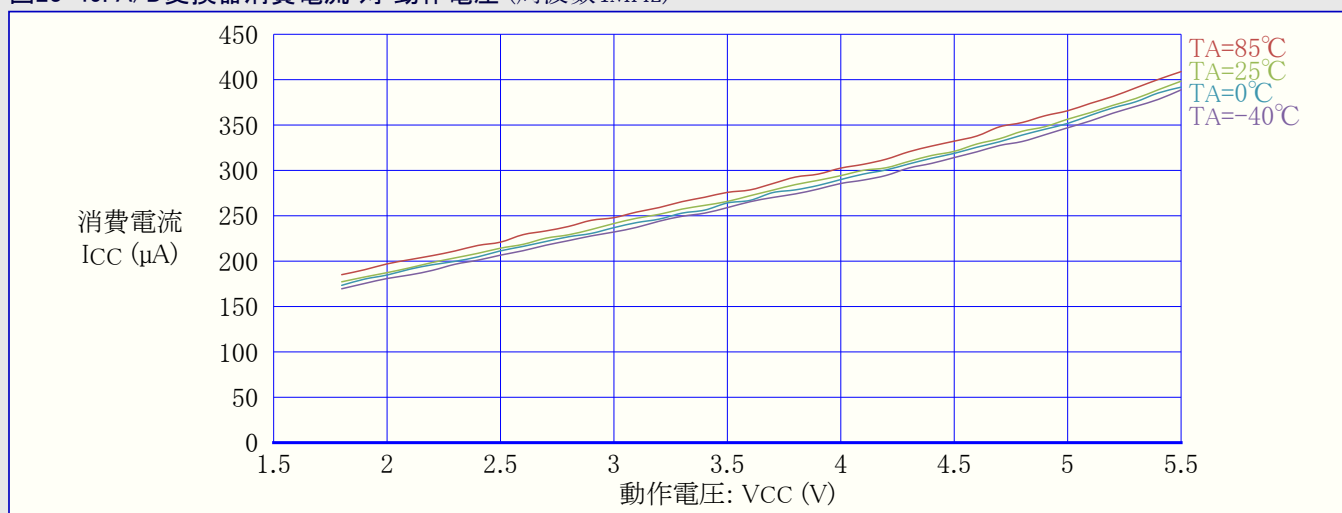


図23-46. アナログ比較器消費電流 対 動作電圧

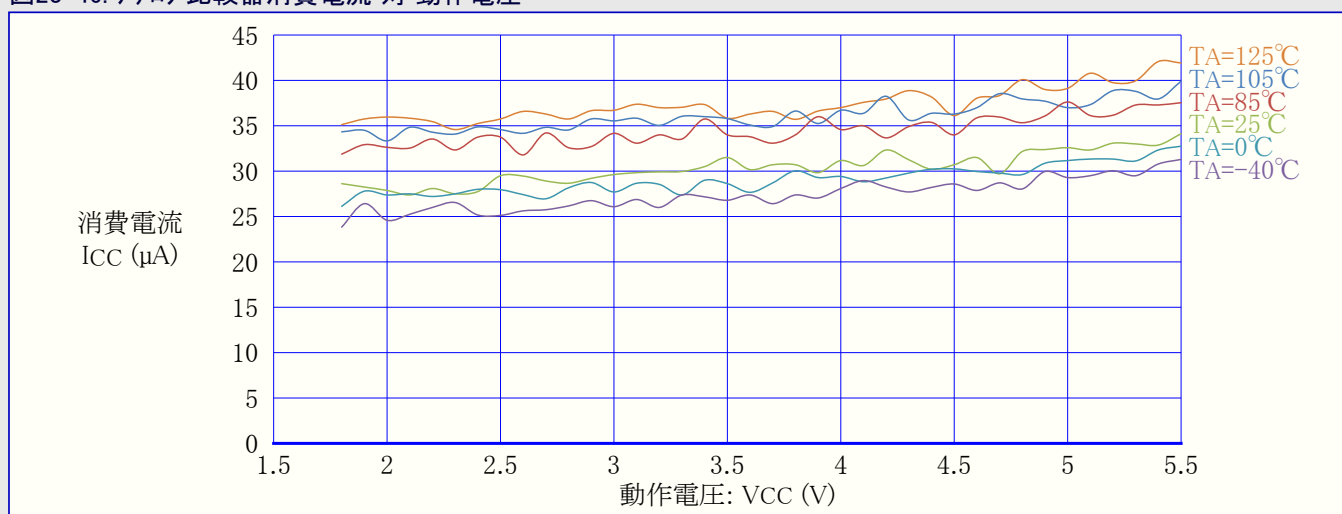


図23-47. VCCレベル監視器消費電流 対 動作電圧

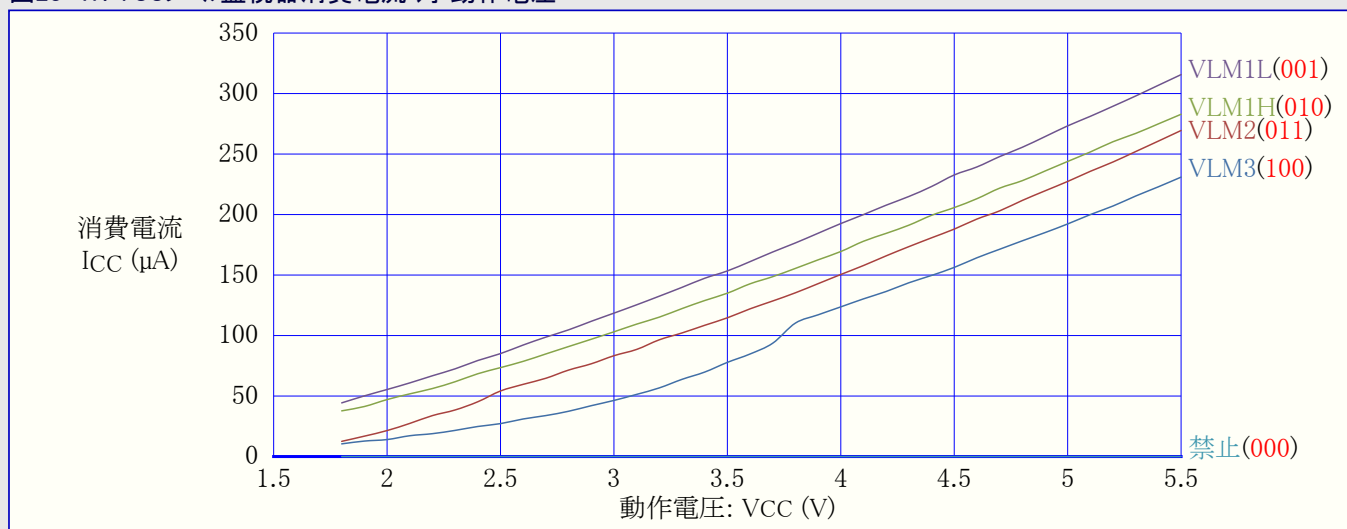


図23-48. VCCレベル監視器消費電流 対 動作電圧の温度依存性 (VLM1L(VLM2~0=001))

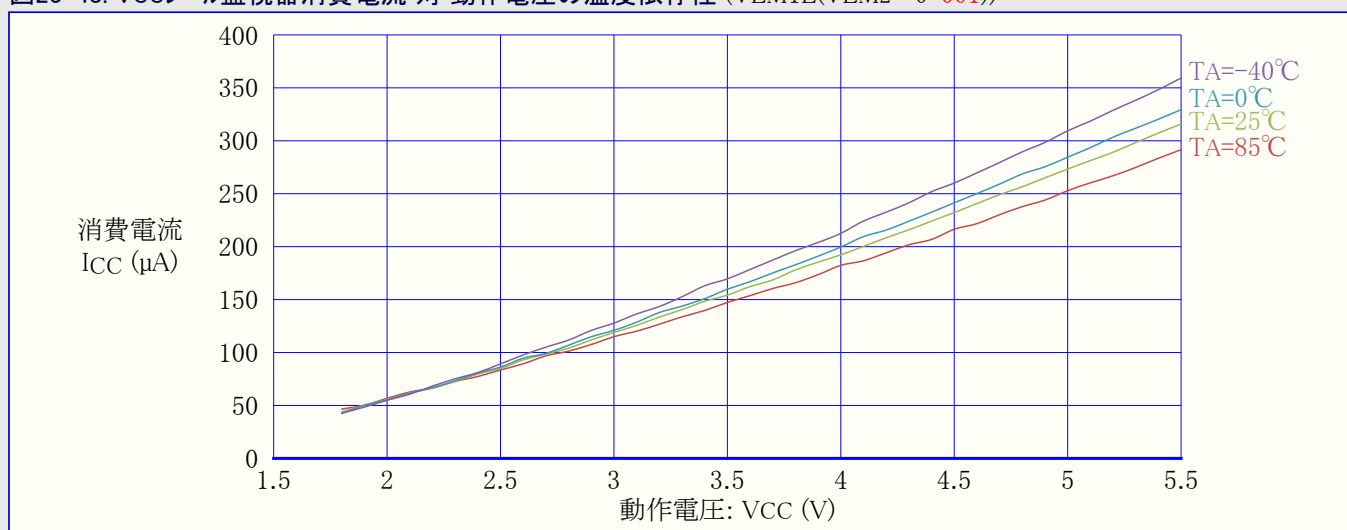
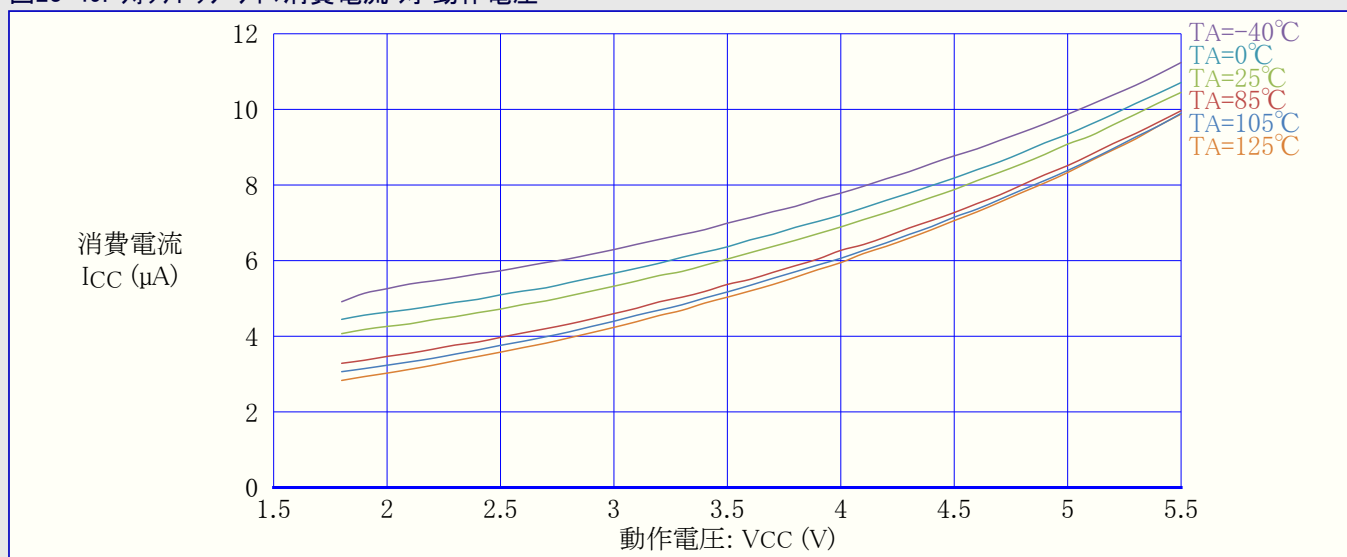
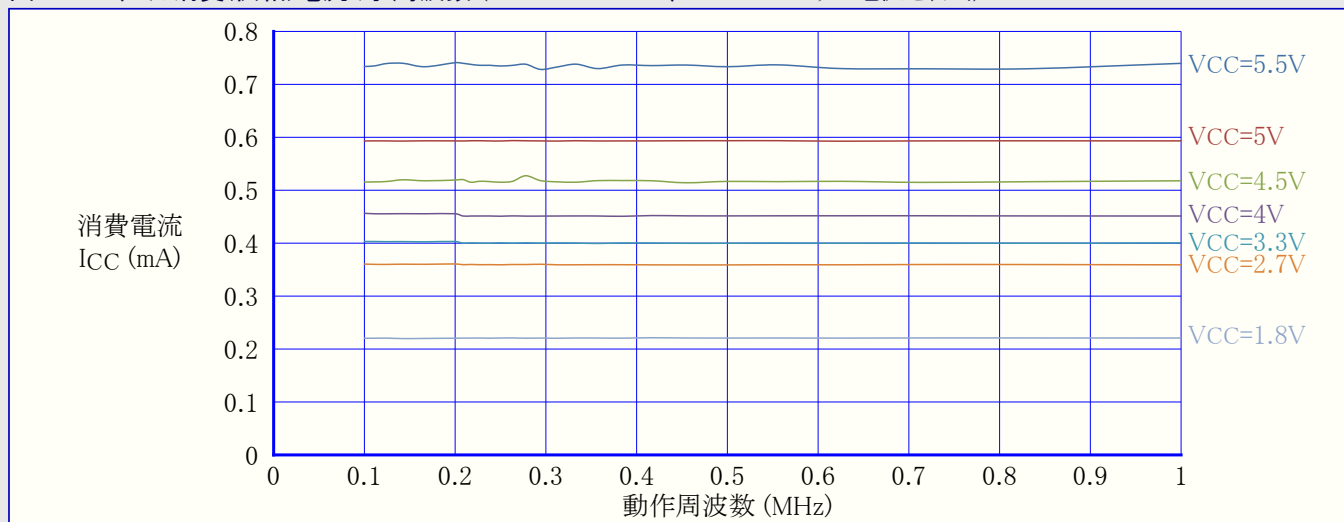


図23-49. ウォッチバック タイマ消費電流 対 動作電圧



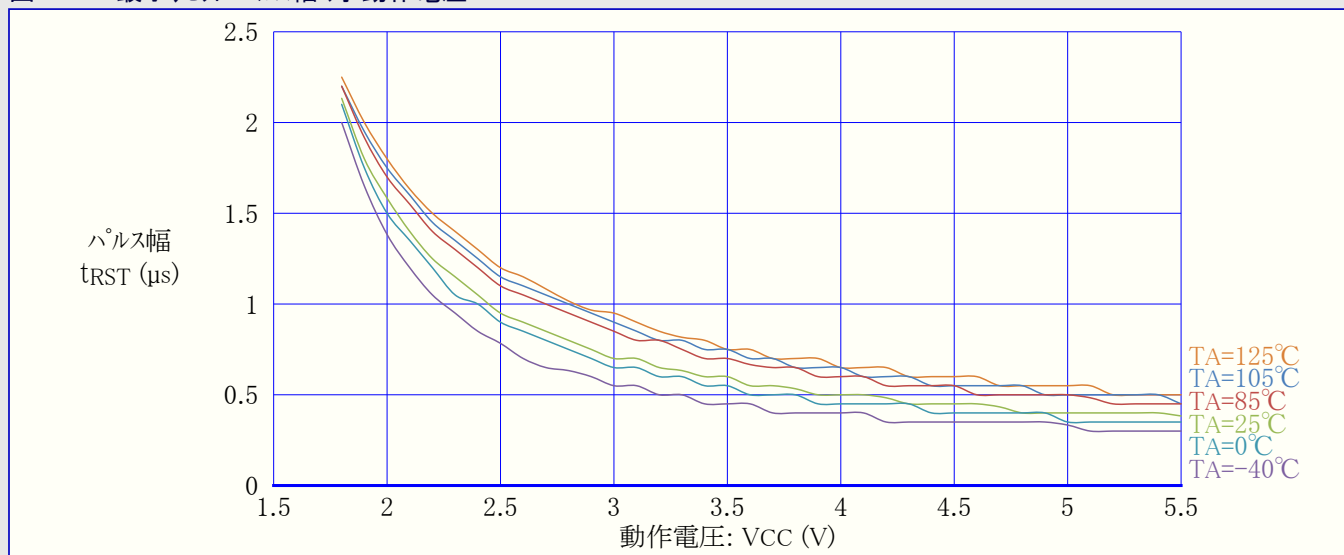
23.12. リセット消費電流とリセットパルス幅

図23-50. リセット消費(供給)電流 対 周波数 (100kHz～1MHz、RESETプルアップ電流を除く)



注: デバイスに対する既定クロック元は常に内部8MHz発振器です。従って、リセットに於ける電流消費は常に外部クロック信号によって影響を及ぼされません。

図23-51. 最小リセットパルス幅 対 動作電圧



24. レジスタ要約

| アドレス | レジスタ略称 | ビット7 | ビット6 | ビット5 | ビット4 | ビット3 | ビット2 | ビット1 | ビット0 |
|------|--------|------------------------------------|----------|-------------------|----------|--------------------------------|----------------|----------------|----------|
| \$3F | SREG | I | T | H | S | V | N | Z | C |
| \$3E | SPH | - | - | - | - | - | - | - | - |
| \$3D | SPL | - | - | SP5 | SP4 | SP3 | SP2 | SP1 | SP0 |
| \$3C | CCP | CCP7~0 (構成設定変更保護ビット) | | | | | | | |
| \$3B | RSTFLR | - | - | - | - | WDRF | - | EXTRF | PORF |
| \$3A | SMCR | - | - | - | - | SM2~0 | | | SE |
| \$39 | OSCCAL | CAL7~0 (内蔵RC発振器 発振校正値レジスタ) | | | | | | | |
| \$38 | 予約 | | | | | | | | |
| \$37 | CLKMSR | - | - | - | - | - | - | CLKMS1,0 | |
| \$36 | CLKPSR | - | - | - | - | CLKPS3~0 | | | |
| \$35 | PRR | - | - | - | - | - | PRUSART0 | PRADC | PRTIM0 |
| \$34 | VLMCSR | VLMF | VLMIE | - | - | - | VLM2~0 | | |
| \$33 | NVMCMD | - | - | NVMCMD5~0 (NVM指令) | | | | | |
| \$32 | NVMCSR | NVMBSY | - | - | - | - | - | - | - |
| \$31 | WDTCR | WDIF | WDIE | WDP3 | - | WDE | WDP2~0 | | |
| \$30 | 予約 | | | | | | | | |
| \$2F | GTCCR | TSM | - | - | - | - | - | REMAP | PSR |
| \$2E | TCCR0A | COM0A1 | COM0A0 | COM0B1 | COM0B0 | - | - | WGM01 | WGM00 |
| \$2D | TCCR0B | ICNC0 | ICES0 | - | WGM03 | WGM02 | CS02~0 | | |
| \$2C | TCCR0C | FOC0A | FOC0B | - | - | - | - | - | - |
| \$2B | TIMSK0 | - | - | ICIE0 | - | - | OCIE0B | OCIE0A | TOIE0 |
| \$2A | TIFR0 | - | - | ICF0 | - | - | OCF0B | OCF0A | TOV0 |
| \$29 | TCNT0H | TCNT15~8 (タイマ/カウンタ0 上位バイト) | | | | | | | |
| \$28 | TCNT0L | TCNT7~0 (タイマ/カウンタ0 下位バイト) | | | | | | | |
| \$27 | OCR0AH | OCR0A15~8 (タイマ/カウンタ0 比較Aレジスタ上位バイト) | | | | | | | |
| \$26 | OCR0AL | OCR0A7~0 (タイマ/カウンタ0 比較Aレジスタ下位バイト) | | | | | | | |
| \$25 | OCR0BH | OCR0B15~8 (タイマ/カウンタ0 比較Bレジスタ上位バイト) | | | | | | | |
| \$24 | OCR0BL | OCR0B7~0 (タイマ/カウンタ0 比較Bレジスタ下位バイト) | | | | | | | |
| \$23 | ICR0H | ICR015~8 (タイマ/カウンタ0 捕獲レジスタ上位バイト) | | | | | | | |
| \$22 | ICR0L | ICR07~0 (タイマ/カウンタ0 捕獲レジスタ下位バイト) | | | | | | | |
| \$21 | 予約 | | | | | | | | |
| \$20 | 予約 | | | | | | | | |
| \$1F | ACSR | ACD | ACBG | ACO | ACI | ACIE | ACIC | ACIS1 | ACIS0 |
| \$1E | ACSRB | - | - | - | - | - | - | ACOE | ACPMUX |
| \$1D | ADCSRA | ADEN | ADSC | ADATE | ADIF | ADIE | ADPS2~0 | | |
| \$1C | ADCSRB | ADLAR | - | - | - | - | ADTS2~0 | | |
| \$1B | ADMUX | REFS1 | REFS0 | - | - | - | MUX2~0 | | |
| \$1A | ADCH | ADC9~8またはADC9~2 (A/Dデータレジスタ上位バイト) | | | | | | | |
| \$19 | ADCL | ADC7~0またはADC1~0 (A/Dデータレジスタ下位バイト) | | | | | | | |
| \$18 | 予約 | | | | | | | | |
| \$17 | DIDR0 | ADC7D | ADC6D | ADC5D | ADC4D | ADC3D | ADC2D | ADC1D | ADC0D |
| \$16 | PORTCR | - | - | - | - | - | - | BBMB | BBMA |
| \$15 | EICRA | - | - | - | - | - | - | ISC01,0 | |
| \$14 | EIFR | - | - | - | - | - | - | - | INTF0 |
| \$13 | EIMSK | - | - | - | - | - | - | - | INT0 |
| \$12 | PCICR | - | - | - | - | - | - | PCIE1 | PCIE0 |
| \$11 | PCIFR | - | - | - | - | - | - | PCIF1 | PCIF0 |
| \$10 | PCMSK1 | - | - | - | - | PCINT11 | PCINT10 | PCINT9 | (PCINT8) |
| \$0F | PCMSK0 | (PCINT7) | (PCINT6) | (PCINT5) | (PCINT4) | (PCINT3) | PCINT2 | PCINT1 | PCINT0 |
| \$0E | UCSR0A | RXC0 | TXC0 | UDRE0 | FE0 | DOR0 | UPE0 | U2X0 | MPCM0 |
| \$0D | UCSR0B | RXCIE0 | TXCIE0 | UDRIE0 | RXEN0 | TXEN0 | UCSZ02 | RXB80 | TXB80 |
| \$0C | UCSR0C | UMSEL01,0 | | UPM01,0 | | USBS0 | UCSZ01 /UDORD0 | UCSZ00 /UCPHA0 | UCPOL0 |
| \$0B | UCSR0D | RXSIE0 | RXS0 | SFDE0 | - | - | - | - | - |
| \$0A | UBRR0H | - | - | - | - | UBRR011~8 (USART0 ボーレートレジスタ上位) | | | |
| \$09 | UBRR0L | UBRR07~0 (USART0 ボーレートレジスタ下位) | | | | | | | |
| \$08 | UDR0 | TXB/RXB7~0 (USART0 データレジスタ) | | | | | | | |
| \$07 | PUEB | - | - | - | - | PUEB3 | PUEB2 | PUEB1 | (PUEB0) |
| \$06 | PORTB | - | - | - | - | PORTB3 | PORTB2 | PORTB1 | (PORTB0) |
| \$05 | DDRB | - | - | - | - | DDRB3 | DDRB2 | DDRB1 | (DDRB0) |
| \$04 | PINB | - | - | - | - | PINB3 | PINB2 | PINB1 | (PINB0) |
| \$03 | PUEA | (PUEA7) | (PUEA6) | (PUEA5) | (PUEA4) | (PUEA3) | PUEA2 | PUEA1 | PUEA0 |
| \$02 | PORTA | (PORTA7) | (PORTA6) | (PORTA5) | (PORTA4) | (PORTA3) | PORTA2 | PORTA1 | PORTA0 |
| \$01 | DDRA | (DDRA7) | (DDRA6) | (DDRA5) | (DDRA4) | (DDRA3) | DDRA2 | DDRA1 | DDRA0 |
| \$00 | PINA | (PINA7) | (PINA6) | (PINA5) | (PINA4) | (PINA3) | PINA2 | PINA1 | PINA0 |

注: • USARTレジスタ(\$08~\$0E)はSBI/CBIの命令を使ってビットアクセスできません。

• ()のビットはATtiny102で利用できません。

25. 命令要約

| ニーモニック | オペランド | 意味 | 動作 | フラグ | クロック |
|-----------|-------|------------------------|---|-----------------|-------|
| 算術、論理演算命令 | | | | | |
| ADD | Rd,Rr | 汎用レジスタ間の加算 | $Rd \leftarrow Rd + Rr$ | I,T,H,S,V,N,Z,C | 1 |
| ADC | Rd,Rr | キャリーを含めた汎用レジスタ間の加算 | $Rd \leftarrow Rd + Rr + C$ | I,T,H,S,V,N,Z,C | 1 |
| SUB | Rd,Rr | 汎用レジスタ間の減算 | $Rd \leftarrow Rd - Rr$ | I,T,H,S,V,N,Z,C | 1 |
| SUBI | Rd,K | 汎用レジスタから即値の減算 | $Rd \leftarrow Rd - K$ | I,T,H,S,V,N,Z,C | 1 |
| SBC | Rd,Rr | キャリーを含めた汎用レジスタ間の減算 | $Rd \leftarrow Rd - Rr - C$ | I,T,H,S,V,N,Z,C | 1 |
| SBCI | Rd,K | 汎用レジスタからキャリーと即値の減算 | $Rd \leftarrow Rd - K - C$ | I,T,H,S,V,N,Z,C | 1 |
| AND | Rd,Rr | 汎用レジスタ間の論理積(AND) | $Rd \leftarrow Rd \text{ AND } Rr$ | I,T,H,S,0,N,Z,C | 1 |
| ANDI | Rd,K | 汎用レジスタと即値の論理積(AND) | $Rd \leftarrow Rd \text{ AND } K$ | I,T,H,S,0,N,Z,C | 1 |
| OR | Rd,Rr | 汎用レジスタ間の論理和(OR) | $Rd \leftarrow Rd \text{ OR } Rr$ | I,T,H,S,0,N,Z,C | 1 |
| ORI | Rd,K | 汎用レジスタと即値の論理和(OR) | $Rd \leftarrow Rd \text{ OR } K$ | I,T,H,S,0,N,Z,C | 1 |
| EOR | Rd,Rr | 汎用レジスタ間の排他的論理和(Ex-OR) | $Rd \leftarrow Rd \text{ EOR } Rr$ | I,T,H,S,0,N,Z,C | 1 |
| COM | Rd | 1の補数(論理反転) | $Rd \leftarrow \$FF - Rd$ | I,T,H,S,0,N,Z,C | 1 |
| NEG | Rd | 2の補数 | $Rd \leftarrow \$00 - Rd$ | I,T,H,S,V,N,Z,C | 1 |
| SBR | Rd,K | 汎用レジスタの(複数)ビット設定(1) | $Rd \leftarrow Rd \text{ OR } K$ | I,T,H,S,0,N,Z,C | 1 |
| CBR | Rd,K | 汎用レジスタの(複数)ビット解除(0) | $Rd \leftarrow Rd \text{ AND } (\$FF - K)$ | I,T,H,S,0,N,Z,C | 1 |
| INC | Rd | 汎用レジスタの増加(+1) | $Rd \leftarrow Rd + 1$ | I,T,H,S,V,N,Z,C | 1 |
| DEC | Rd | 汎用レジスタの減少(-1) | $Rd \leftarrow Rd - 1$ | I,T,H,S,V,N,Z,C | 1 |
| TST | Rd | 汎用レジスタのゼロとマイナス検査 | $Rd \leftarrow Rd \text{ AND } Rd$ | I,T,H,S,0,N,Z,C | 1 |
| CLR | Rd | 汎用レジスタの全0設定(=\$00) | $Rd \leftarrow Rd \text{ EOR } Rd$ | I,T,H,0,0,0,I,C | 1 |
| SER | Rd | 汎用レジスタの全1設定(=\$FF) | $Rd \leftarrow \$FF$ | I,T,H,S,V,N,Z,C | 1 |
| 分岐命令 | | | | | |
| RJMP | k | 相対分岐 | $PC \leftarrow PC + k + 1$ | I,T,H,S,V,N,Z,C | 2 |
| IJMP | | Zレジスタ間接分岐 | $PC \leftarrow Z$ | I,T,H,S,V,N,Z,C | 2 |
| RCALL | k | 相対サブルーチン呼び出し | $STACK \leftarrow PC, PC \leftarrow PC + k + 1$ | I,T,H,S,V,N,Z,C | 3(注) |
| ICALL | | Zレジスタ間接サブルーチン呼び出し | $STACK \leftarrow PC, PC \leftarrow Z$ | I,T,H,S,V,N,Z,C | 3(注) |
| RET | | サブルーチンからの復帰 | $PC \leftarrow STACK$ | I,T,H,S,V,N,Z,C | 6(注) |
| RETI | | 割り込みからの復帰 | $PC \leftarrow STACK$ | I,T,H,S,V,N,Z,C | 6(注) |
| CPSE | Rd,Rr | 汎用レジスタ間比較、一致でスキップ | $Rd=Rr$ なら, $PC \leftarrow PC + 2or3$ | I,T,H,S,V,N,Z,C | 1/2,3 |
| CP | Rd,Rr | 汎用レジスタ間の比較 | $Rd - Rr$ | I,T,H,S,V,N,Z,C | 1 |
| CPC | Rd,Rr | キャリーを含めた汎用レジスタ間の比較 | $Rd - Rr - C$ | I,T,H,S,V,N,Z,C | 1 |
| CPI | Rd,K | 汎用レジスタと即値の比較 | $Rd - K$ | I,T,H,S,V,N,Z,C | 1 |
| SBRC | Rr,b | 汎用レジスタのビットが解除(0)でスキップ | $Rr(b)=0$ なら, $PC \leftarrow PC + 2or3$ | I,T,H,S,V,N,Z,C | 1/2,3 |
| SBRs | Rr,b | 汎用レジスタのビットが設定(1)でスキップ | $Rr(b)=1$ なら, $PC \leftarrow PC + 2or3$ | I,T,H,S,V,N,Z,C | 1/2,3 |
| SBIC | P,b | I/Oレジスタのビットが解除(0)でスキップ | $P(b)=0$ なら, $PC \leftarrow PC + 2or3$ | I,T,H,S,V,N,Z,C | 1/2,3 |
| SBIS | P,b | I/Oレジスタのビットが設定(1)でスキップ | $P(b)=1$ なら, $PC \leftarrow PC + 2or3$ | I,T,H,S,V,N,Z,C | 1/2,3 |
| BRBS | s,k | ステータスフラグが設定(1)で分岐 | $SREG(s)=1$ なら, $PC \leftarrow PC + K + 1$ | I,T,H,S,V,N,Z,C | 1/2 |
| BRBC | s,k | ステータスフラグが解除(0)で分岐 | $SREG(s)=0$ なら, $PC \leftarrow PC + K + 1$ | I,T,H,S,V,N,Z,C | 1/2 |
| BREQ | k | 一致で分岐 | $Z=1$ なら, $PC \leftarrow PC + K + 1$ | I,T,H,S,V,N,Z,C | 1/2 |
| BRNE | k | 不一致で分岐 | $Z=0$ なら, $PC \leftarrow PC + K + 1$ | I,T,H,S,V,N,Z,C | 1/2 |
| BRCS | k | キャリーフラグが設定(1)で分岐 | $C=1$ なら, $PC \leftarrow PC + K + 1$ | I,T,H,S,V,N,Z,C | 1/2 |
| BRCC | k | キャリーフラグが解除(0)で分岐 | $C=0$ なら, $PC \leftarrow PC + K + 1$ | I,T,H,S,V,N,Z,C | 1/2 |
| BRSH | k | 符号なしの \geq で分岐 | $C=0$ なら, $PC \leftarrow PC + K + 1$ | I,T,H,S,V,N,Z,C | 1/2 |
| BRLO | k | 符号なしの $<$ で分岐 | $C=1$ なら, $PC \leftarrow PC + K + 1$ | I,T,H,S,V,N,Z,C | 1/2 |
| BRMI | k | -(マイナス)で分岐 | $N=1$ なら, $PC \leftarrow PC + K + 1$ | I,T,H,S,V,N,Z,C | 1/2 |
| BRPL | k | +(プラス)で分岐 | $N=0$ なら, $PC \leftarrow PC + K + 1$ | I,T,H,S,V,N,Z,C | 1/2 |
| BRGE | k | 符号付きの \geq で分岐 | $(N \text{ EOR } V)=0$ なら, $PC \leftarrow PC + K + 1$ | I,T,H,S,V,N,Z,C | 1/2 |
| BRLT | k | 符号付きの $<$ で分岐 | $(N \text{ EOR } V)=1$ なら, $PC \leftarrow PC + K + 1$ | I,T,H,S,V,N,Z,C | 1/2 |
| BRHS | k | ハーフキャリーフラグが設定(1)で分岐 | $H=1$ なら, $PC \leftarrow PC + K + 1$ | I,T,H,S,V,N,Z,C | 1/2 |
| BRHC | k | ハーフキャリーフラグが解除(0)で分岐 | $H=0$ なら, $PC \leftarrow PC + K + 1$ | I,T,H,S,V,N,Z,C | 1/2 |
| BRTS | k | 一時フラグが設定(1)で分岐 | $T=1$ なら, $PC \leftarrow PC + K + 1$ | I,T,H,S,V,N,Z,C | 1/2 |
| BRTC | k | 一時フラグが解除(0)で分岐 | $T=0$ なら, $PC \leftarrow PC + K + 1$ | I,T,H,S,V,N,Z,C | 1/2 |
| BRVS | k | 2の補数溢れフラグが設定(1)で分岐 | $V=1$ なら, $PC \leftarrow PC + K + 1$ | I,T,H,S,V,N,Z,C | 1/2 |
| BRVC | k | 2の補数溢れフラグが解除(0)で分岐 | $V=0$ なら, $PC \leftarrow PC + K + 1$ | I,T,H,S,V,N,Z,C | 1/2 |
| BRIE | k | 割り込み許可で分岐 | $I=1$ なら, $PC \leftarrow PC + K + 1$ | I,T,H,S,V,N,Z,C | 1/2 |
| BRID | k | 割り込み禁止で分岐 | $I=0$ なら, $PC \leftarrow PC + K + 1$ | I,T,H,S,V,N,Z,C | 1/2 |

K : 8ビット定数

P : I/Oレジスタ

Rd, Rr : 汎用レジスタ(R0~R31)

X, Y, Z : X, Y, Zレジスタ

b : ビット(0~7)

k : アドレス定数(7,12ビット)

s : ステータスフラグ(C,Z,N,V,X,H,T,I)

| ニーモニック | オペランド | 意味 | 動作 | フラグ | クロック |
|---------|-------|---------------------|--|-----------------|------|
| データ移動命令 | | | | | |
| MOV | Rd,Rr | 汎用レジスタ間の複写 | $Rd \leftarrow Rr$ | I,T,H,S,V,N,Z,C | 1 |
| LDI | Rd,K | 即値の取得 | $Rd \leftarrow K$ | I,T,H,S,V,N,Z,C | 1 |
| LD | Rd,X | Xレジスタ間接での取得 | $Rd \leftarrow (X)$ | I,T,H,S,V,N,Z,C | 1/2 |
| LD | Rd,X+ | 事後増加付きXレジスタ間接での取得 | $Rd \leftarrow (X), X \leftarrow X + 1$ | I,T,H,S,V,N,Z,C | 2/3 |
| LD | Rd,-X | 事前減少付きXレジスタ間接での取得 | $X \leftarrow X - 1, Rd \leftarrow (X)$ | I,T,H,S,V,N,Z,C | 2/3 |
| LD | Rd,Y | Yレジスタ間接での取得 | $Rd \leftarrow (Y)$ | I,T,H,S,V,N,Z,C | 1/2 |
| LD | Rd,Y+ | 事後増加付きYレジスタ間接での取得 | $Rd \leftarrow (Y), Y \leftarrow Y + 1$ | I,T,H,S,V,N,Z,C | 2/3 |
| LD | Rd,-Y | 事前減少付きYレジスタ間接での取得 | $Y \leftarrow Y - 1, Rd \leftarrow (Y)$ | I,T,H,S,V,N,Z,C | 2/3 |
| LD | Rd,Z | Zレジスタ間接での取得 | $Rd \leftarrow (Z)$ | I,T,H,S,V,N,Z,C | 1/2 |
| LD | Rd,Z+ | 事後増加付きZレジスタ間接での取得 | $Rd \leftarrow (Z), Z \leftarrow Z + 1$ | I,T,H,S,V,N,Z,C | 2/3 |
| LD | Rd,-Z | 事前減少付きZレジスタ間接での取得 | $Z \leftarrow Z - 1, Rd \leftarrow (Z)$ | I,T,H,S,V,N,Z,C | 2/3 |
| LDS | Rd,k | データ空間(SRAM)から直接取得 | $Rd \leftarrow (k)$ | I,T,H,S,V,N,Z,C | 2 |
| ST | X,Rr | Xレジスタ間接での設定 | $(X) \leftarrow Rr$ | I,T,H,S,V,N,Z,C | 1 |
| ST | X+,Rr | 事後増加付きXレジスタ間接での設定 | $(X) \leftarrow Rr, X \leftarrow X + 1$ | I,T,H,S,V,N,Z,C | 1 |
| ST | -X,Rr | 事前減少付きXレジスタ間接での設定 | $X \leftarrow X - 1, (X) \leftarrow Rr$ | I,T,H,S,V,N,Z,C | 2 |
| ST | Y,Rr | Yレジスタ間接での設定 | $(Y) \leftarrow Rr$ | I,T,H,S,V,N,Z,C | 1 |
| ST | Y+,Rr | 事後増加付きYレジスタ間接での設定 | $(Y) \leftarrow Rr, Y \leftarrow Y + 1$ | I,T,H,S,V,N,Z,C | 1 |
| ST | -Y,Rr | 事前減少付きYレジスタ間接での設定 | $Y \leftarrow Y - 1, (Y) \leftarrow Rr$ | I,T,H,S,V,N,Z,C | 2 |
| ST | Z,Rr | Zレジスタ間接での設定 | $(Z) \leftarrow Rr$ | I,T,H,S,V,N,Z,C | 1 |
| ST | Z+,Rr | 事後増加付きZレジスタ間接での設定 | $(Z) \leftarrow Rr, Z \leftarrow Z + 1$ | I,T,H,S,V,N,Z,C | 1 |
| ST | -Z,Rr | 事前減少付きZレジスタ間接での設定 | $Z \leftarrow Z - 1, (Z) \leftarrow Rr$ | I,T,H,S,V,N,Z,C | 2 |
| STS | k,Rr | データ空間(SRAM)へ直接設定 | $(k) \leftarrow Rr$ | I,T,H,S,V,N,Z,C | 1 |
| IN | Rd,P | I/Oレジスタからの入力 | $Rd \leftarrow P$ | I,T,H,S,V,N,Z,C | 1 |
| OUT | P,Rr | I/Oレジスタへの出力 | $P \leftarrow Rr$ | I,T,H,S,V,N,Z,C | 1 |
| PUSH | Rr | 汎用レジスタをスタックへ保存 | $STACK \leftarrow Rr$ | I,T,H,S,V,N,Z,C | 1 |
| POP | Rd | スタックから汎用レジスタへ復帰 | $Rd \leftarrow STACK$ | I,T,H,S,V,N,Z,C | 3(注) |
| ビット関係命令 | | | | | |
| SBI | P,b | I/Oレジスタのビット設定(1) | $I/O(P,b) \leftarrow 1$ | I,T,H,S,V,N,Z,C | 1 |
| CBI | P,b | I/Oレジスタのビット解除(0) | $I/O(P,b) \leftarrow 0$ | I,T,H,S,V,N,Z,C | 1 |
| LSL | Rd | 論理的左ビット移動 | $Rd(n+1) \leftarrow Rd(n), Rd(0) \leftarrow 0$ | I,T,H,S,V,N,Z,C | 1 |
| LSR | Rd | 論理的右ビット移動 | $Rd(n) \leftarrow Rd(n+1), Rd(7) \leftarrow 0$ | I,T,H,S,V,0,Z,C | 1 |
| ROL | Rd | キャリーを含めた左回転 | $Rd(0) \leftarrow C, Rd(n+1) \leftarrow Rd(n), C \leftarrow Rd(7)$ | I,T,H,S,V,N,Z,C | 1 |
| ROR | Rd | キャリーを含めた右回転 | $Rd(7) \leftarrow C, Rd(n) \leftarrow Rd(n+1), C \leftarrow Rd(0)$ | I,T,H,S,V,N,Z,C | 1 |
| ASR | Rd | 算術的右ビット移動 | $Rd(n) \leftarrow Rd(n+1), n=0 \sim 6$ | I,T,H,S,V,N,Z,C | 1 |
| SWAP | Rd | ニブル(4ビット)上位/下位交換 | $Rd(7 \sim 4) \leftrightarrow Rd(3 \sim 0)$ | I,T,H,S,V,N,Z,C | 1 |
| BSET | s | ステータスレジスタのビット設定(1) | $SREG(s) \leftarrow 1$ | 1,1,1,1,1,1,1,1 | 1 |
| BCLR | s | ステータスレジスタのビット解除(0) | $SREG(s) \leftarrow 0$ | 0,0,0,0,0,0,0,0 | 1 |
| BST | Rr,b | 汎用レジスタのビットを一時フラグへ移動 | $T \leftarrow Rr(b)$ | I,T,H,S,V,N,Z,C | 1 |
| BLD | Rd,b | 一時フラグを汎用レジスタのビットへ移動 | $Rd(b) \leftarrow T$ | I,T,H,S,V,N,Z,C | 1 |
| SEC | | キャリーフラグを設定(1) | $C \leftarrow 1$ | I,T,H,S,V,N,Z,C | 1 |
| CLC | | キャリーフラグを解除(0) | $C \leftarrow 0$ | I,T,H,S,V,N,Z,0 | 1 |
| SEN | | 負フラグを設定(1) | $N \leftarrow 1$ | I,T,H,S,V,1,Z,C | 1 |
| CLN | | 負フラグを解除(0) | $N \leftarrow 0$ | I,T,H,S,V,0,Z,C | 1 |
| SEZ | | ゼロフラグを設定(1) | $Z \leftarrow 1$ | I,T,H,S,V,N,1,C | 1 |
| CLZ | | ゼロフラグを解除(0) | $Z \leftarrow 0$ | I,T,H,S,V,N,0,C | 1 |
| SEI | | 全割り込み許可 | $I \leftarrow 1$ | I,T,H,S,V,N,Z,C | 1 |
| CLI | | 全割り込み禁止 | $I \leftarrow 0$ | 0,T,H,S,V,N,Z,C | 1 |
| SES | | 符号フラグを設定(1) | $S \leftarrow 1$ | I,T,H,1,V,N,Z,C | 1 |
| CLS | | 符号フラグを解除(0) | $S \leftarrow 0$ | I,T,H,0,V,N,Z,C | 1 |
| SEV | | 2の補数溢れフラグを設定(1) | $V \leftarrow 1$ | I,T,H,S,1,N,Z,C | 1 |
| CLV | | 2の補数溢れフラグを解除(0) | $V \leftarrow 0$ | I,T,H,S,0,N,Z,C | 1 |
| SET | | 一時フラグを設定(1) | $T \leftarrow 1$ | I,1,H,S,V,N,Z,C | 1 |
| CLT | | 一時フラグを解除(0) | $T \leftarrow 0$ | I,0,H,S,V,N,Z,C | 1 |
| SEH | | ハーフキャリーフラグを設定(1) | $H \leftarrow 1$ | I,T,1,S,V,N,Z,C | 1 |
| CLH | | ハーフキャリーフラグを解除(0) | $H \leftarrow 0$ | I,T,0,S,V,N,Z,C | 1 |
| MCU制御命令 | | | | | |
| NOP | | 無操作 | | I,T,H,S,V,N,Z,C | 1 |
| SLEEP | | 休止形態動作開始 | 休止形態動作参照 | I,T,H,S,V,N,Z,C | 1 |
| WDR | | ウォッチドッグタイマリセット | ウォッチドッグタイマ参照 | I,T,H,S,V,N,Z,C | 1 |
| BREAK | | 一時停止 | 対応デバッグ資料参照 | I,T,H,S,V,N,Z,C | 1 |

注: データメモリアクセスに対する周期時間は内部SRAMアクセスと仮定され、NVM制御器を通ずアクセスに対して有効ではありません。(フラッシュメモリとEEPROMのような)NVM制御器を通したメモリアクセス時、最低1つの追加周期が加えられなければならないが、他の主制御部またはVNM制御器の状態による同時アクセスに依存し、1つの追加周期以上になるかもしれません。

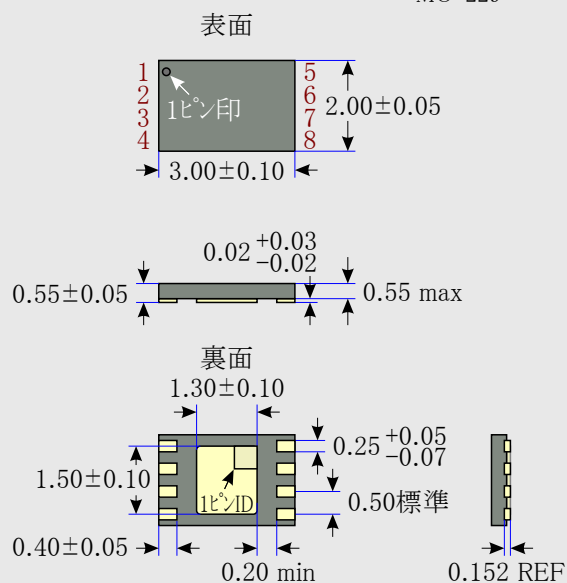
(訳補) 実行クロック数は表内説明と矛盾しない限り、基本的に原書に従っていますが、特にCALL,RET,LD,ST系の実行クロック数は本文説明や他の資料と矛盾する部分があるため誤っている可能性があります。

26. 外圍器情報

26.1. 8ピン[®]UDFN

8ピン[®] プラスティック極薄2列平板リードなし外圍器 (UDFN)

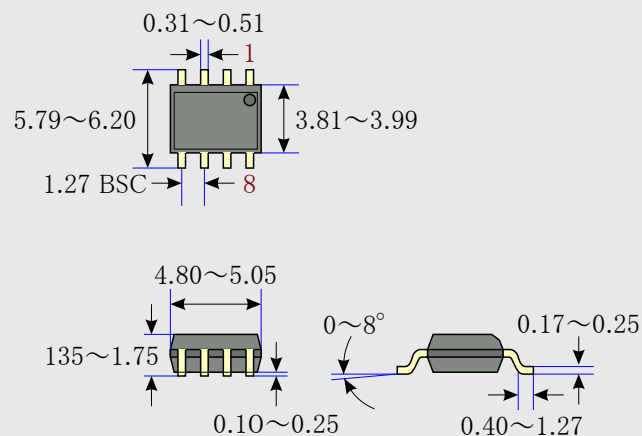
寸法: mm
MO-229



26.2. 8ピン[®]SOIC150

8ピン[®] 0.150インチ幅本体 プラスティック ガルウイング[®]小外形外圍器 (JEDEC SOIC)

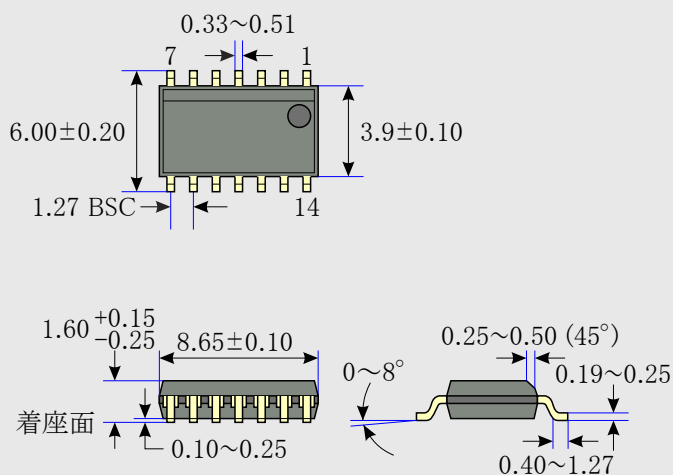
寸法: mm
JEDEC規格 MS-012



26.3. 14ピン[®]SOIC150

14ピン[®] プラスティック小外形外圍器 (SOIC)

寸法: mm
JEDEC規格 MS-012



27. 障害情報

ATtiny102 改訂A：既知の問題はありません。

ATtiny104 改訂A：既知の問題はありません。

28. データシート改訂履歴

28.1. 改訂A – 2016年2月

- 初版

28.2. 改訂B – 2016年6月

- 「電気的特性」を更新
- 「代表特性」を更新

28.3. 改訂C – 2016年7月

- 8ピンUDFN外囲器の放熱パッドはVCC

28.4. 改訂D – 2016年10月

- AVR CPUコア:
 - 新しい項「16ビットレジスタのアクセス」を追加
 - スタック ポインタ(SPHとSPL)レジスタ更新
- USART:
 - UCSR0D.SFDEビット追加、UCSR0D.RXIEビット名をRXSIEに修正
 - UBRR0HとUBRR0Lレジスタの表現を更新
- TC0 – 16ビット PWM付きタイマ/カウンタ:
 - TCNTHとTCNTL、OCR0AHとOCR0AL、OCR0BHとOCR0BL、ICR0HとICR0Lレジスタの表現を更新
- ADC – A/D変換器:
 - (ADLAR=0と1の両方に対する)ADCHとADCLレジスタを更新

Atmel®, Atmelロゴとそれらの組み合わせ、Enabling Unlimited Possibilities®, AVR®とその他は米国及び他の国に於けるAtmel Corporationの登録商標または商標です。他の用語と製品名は一般的に他の商標です。

お断り: 本資料内の情報はAtmel製品と関連して提供されています。本資料またはAtmel製品の販売と関連して承諾される何れの知的所有権も禁反言あるいはその逆によって明示的または暗示的に承諾されるものではありません。Atmelのウェブサイトに表示する販売の条件とAtmelの定義での詳しい説明を除いて、商品性、特定目的に関する適合性、または適法性の暗黙保証に制限せず、Atmelはそれらを含むその製品に関連する暗示的、明示的または法令による如何なる保証も否認し、何ら責任がないと認識します。たとえAtmelがそのような損害賠償の可能性を進言されたとしても、本資料を使用できない、または使用以外で発生する(情報の損失、事業中断、または利益と損失に関する制限なしの損害賠償を含み)直接、間接、必然、偶然、特別、または付随して起こる如何なる損害賠償に対しても決してAtmelに責任がないでしょう。Atmelは本資料の内容の正確さまたは完全性に関して断言または保証を行わず、予告なしでいつでも製品内容と仕様の変更を行う権利を保留します。Atmelはここに含まれた情報を更新することに対してどんな公約も行いません。特に別の方法で提供されなければ、Atmel製品は車載応用に対して適当ではなく、使用されるべきではありません。Atmel製品は延命または生命維持を意図した応用での部品としての使用に対して意図、認定、または保証されません。

安全重視、軍用、車載応用のお断り: Atmel製品はAtmelが提供する特別に書かれた承諾を除き、そのような製品の機能不全が著しく人に危害を加えたり死に至らしめることがかなり予期されるどんな応用(“安全重視応用”)に対しても設計されず、またそれらとの接続にも使用されません。安全重視応用は限定なしで、生命維持装置とシステム、核施設と武器システムの操作の装置やシステムを含みます。Atmelによって軍用等級として特に明確に示される以外、Atmel製品は軍用や航空宇宙の応用や環境のために設計も意図もされていません。Atmelによって車載等級として特に明確に示される以外、Atmel製品は車載応用での使用のために設計も意図もされていません。

© HERO 2022.

本データシートはAtmelのATtiny102/ATtiny104英語版データシート(改訂42505D-10/2016)の翻訳日本語版です。日本語では不自然となる重複する形容表現は省略されている場合があります。日本語では難解となる表現は大幅に意識されている部分もあります。必要に応じて一部加筆されています。頁割の変更により、原本より頁数が少なくなっています。

汎用入出力ポートの出力データレジスタとピン入力は、対応関係からの理解の容易さから出力レジスタと入力レジスタで統一表現されています。一部の用語がより適切と思われる名称に変更されています。必要と思われる部分には()内に英語表記や略称などを残す形で表記しています。

青字の部分はリンクとなっています。一般的に赤字の0,1は論理0,1を表します。その他の赤字は重要な部分を表します。

若干の構成変更により、一部の章、節、項番号が異なります。原書に於ける項図表番号の修正により、一部の図表番号が異なります。