

## 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134命令(多くは1周期で実行)
  - 32個の1バイト長汎用レジスタ+周辺制御レジスタ
  - 完全なステティック動作
  - 16MHz時、16MIPSに達する高速動作
  - 2周期実行の乗算命令
- 高耐久不揮発性メモリ部
  - 実装自己書き換え可能な128Kバイト(64K語)フラッシュメモリ内蔵
  - 4KバイトのEEPROM
  - 4Kバイトの内蔵SRAM
  - 書き換え回数: 10,000/フラッシュ, 100,000/EEPROM
  - データ保持力: 20年/85°C, 100年/25°C
  - 個別施錠ビットを持つ任意のプログラム領域  
チップ内プログラムによる実装書き換え  
真の書き込み中の読み出し動作
  - 64Kバイトまでの任意外部メモリ空間
  - ソフトウェア保護用の設定可能な施錠機能
  - 実装書き込み用SPIインターフェース
- QTouch® ライフライ支援
  - 容量性接触釦、滑動部、輪
  - QTouchとQMatrix®での取得
  - 最大64の感知チャネル
- JTAG (IEEE 1149.1準拠) インターフェース
  - JTAG規格に従った境界走査(Boundary-Scan)能力
  - 広範囲な内蔵デバッグ機能
  - JTAGインターフェース経由でのフラッシュ、EEPROM、ヒューズ、施錠ビットのプログラミング
- 内蔵周辺機能
  - 独立した前置分周器、比較機能付き2つの8ビットタイマ/カウンタ
  - 独立した前置分周器、比較、捕獲機能付き2つの拡張した16ビットタイマ/カウンタ
  - 専用発振器と8ビットタイマ/カウンタによる実時間計数器(RTC)
  - 2つの8ビットPWM出力と6つの1~16ビットPWM出力
  - 比較出力変調器
  - 8チャネルの10ビットA/D変換器  
シングルエンド入力 8チャネル、差動入力 7チャネル  
可変利得(×1, ×10, ×200)差動入力 2チャネル
  - バイト対応の2線直列インターフェース
  - 設定可能な2つの直列USART
  - 主装置/従装置動作SPI直列インターフェース
  - 設定可能な専用発振器付きウォッチドッグタイマ
  - アナログ比較器
- 特殊マイクロコントローラ機能
  - 電源ONリセット回路と設定可能な低電圧検出器(BOD)
  - 校正可能な内蔵RC発振器
  - 外部及び内部の割り込み
  - アイドル、A/D変換雑音低減、パワーセーブ、パワーダウン、スタンバイ、拡張スタンバイの6つの低消費動作
  - ソフトウェア選択可能なクロック周波数
  - ヒューズでのATmega103互換動作
  - 全プルアップ禁止機能
- I/Oと外圍器
  - 53ビットの設定可能なI/O
  - 64リードTQFP、64パッドQFN/MLF
- 動作電圧
  - 2.7~5.5V (ATmega128L)
  - 4.5~5.5V (ATmega128)
- 動作速度
  - 0~8MHz (ATmega128L)
  - 0~16MHz (ATmega128)



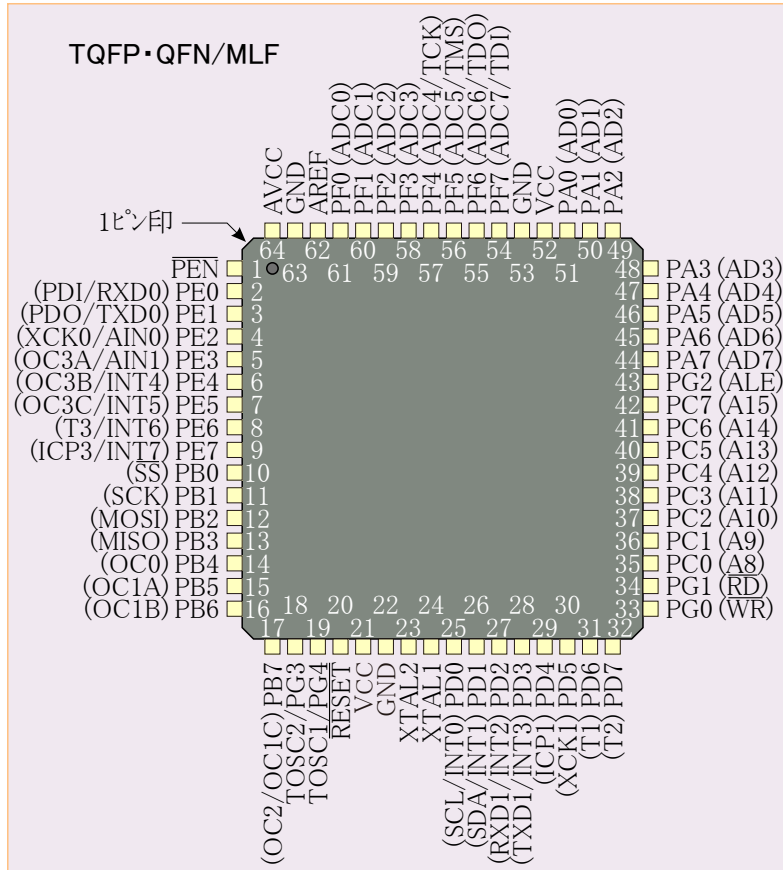
8ビット Atmel  
マイクロコントローラ  
実装書き換え可能な  
128Kバイト  
フラッシュメモリ内蔵

ATmega128  
ATmega128L

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

Rev. 2467X-06/11, 2467XJ7-09/22

## ピン配置



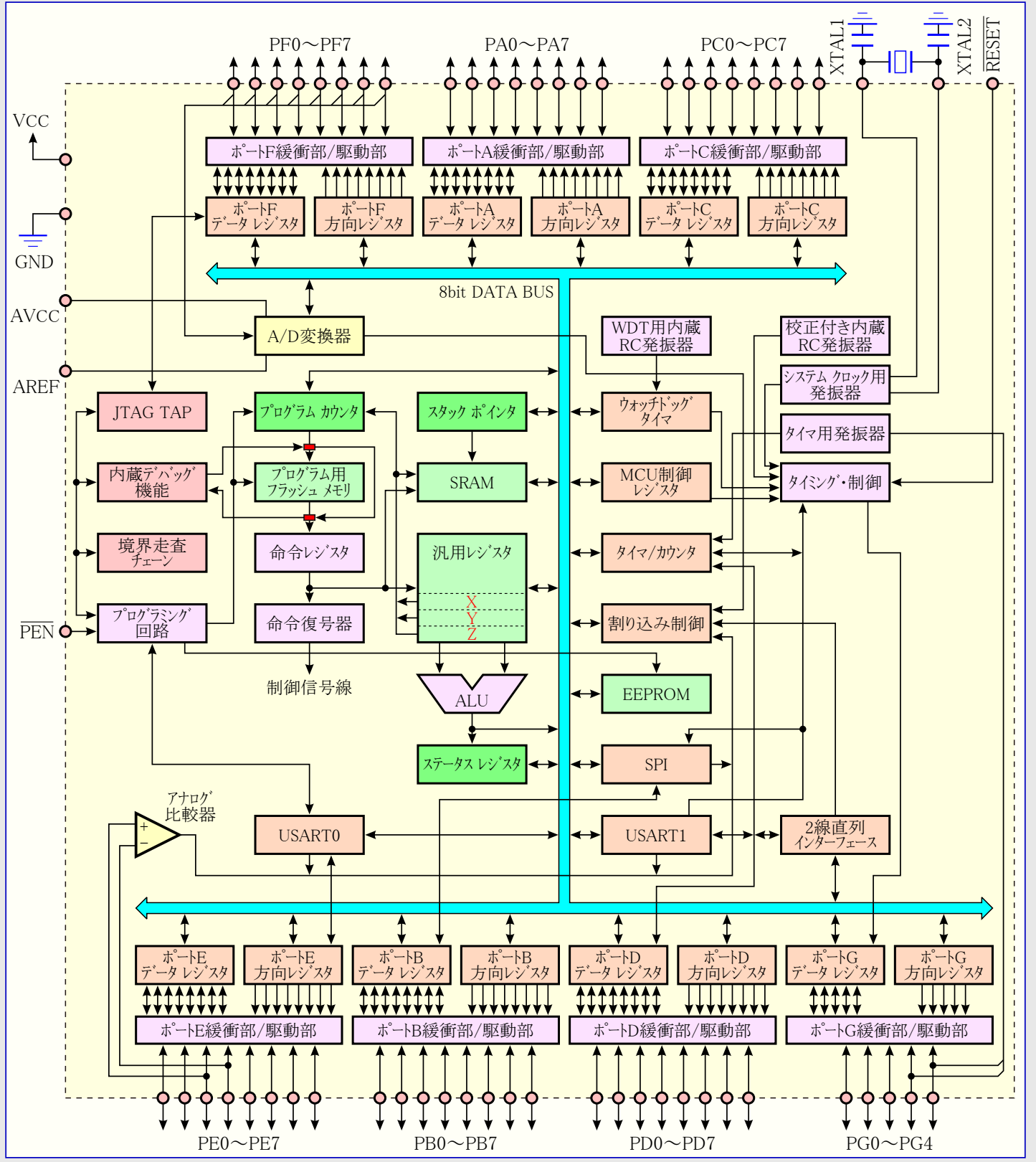
**注:** このピン配置図はTQFPとQFN/MLF両方に適用します。  
QFN/MLFの底面パッドはGNDに半田付けされるべきです。

## 概要

Atmel® AVR® ATmega128はAVR RISC構造の低消費CMOS 8ビットマイクロコントローラです。1周期で実行する強力な命令は、MHzあたり1 MIPSにも達し、実行速度対電力消費の最適化が容易に行えます。

## 構成図

図2. ATmega128構成図



Atmel® AVR® AVRは32個の汎用レジスタと豊富な命令群を兼ね備えています。32個の全レジスタはALU(Arithmetic Logic Unit)に直結され、レジスタ間命令は1クロック周期で実行されます。AVR構造は現状のCISC型マイクロコントローラに対し、最大10倍の単位処理量向上効果があります。



ATmega128は次の特徴、書き込み中読める能力を持つ128Kバイトの実装書き換え可能なフラッシュメモリと4KバイトのEEPROM、4KバイトのSRAM、53本の汎用入出力線、32個の汎用作業レジスタ、実時間計数器(RTC)、比較動作やPWMを含む柔軟性のある4つのタイマ/カウンタ、2つのUSART、バイト対応の2線直列インターフェース、設定変更可能な増幅器を持つ任意選択差動入力付き8チャンネルの10ビットA/D変換器、設定変更可能な内部発振器付きウォッチドッグタイマ、SPI直列ポート、内蔵デバッグとプログラミング機能にも使われるIEEE標準1149.1準拠JTAG検査インターフェース、ソフトウェア選択可能な6つの低消費動作機能を提供します。アイドル動作では動作を停止しますが、SRAM、タイマ/カウンタ、SPIポート、割り込み機能は有効で、動作を継続します。パワーダウン動作ではレジスタの内容は保護されますが、発振器が停止するため、以降のハードウェアリセットか外部割り込みまで他の全機能を禁止(無効に)します。パワーセーブ動作では非同期タイマ用発振器が動作を継続し、デバイスのその他が停止中であっても基準タイマの継続が許されます。A/D変換雑音低減動作ではA/D変換中の切り替え雑音を最小とするために、非同期タイマとA/D変換器を除く全ての周辺機能とCPUが停止します。スタンバイ動作ではクリスタル発振器/セラミック振動子用発振器が動作し、一方デバイスのその他は休止します。これは低消費電力と非常に速い起動の組み合わせを許します。拡張スタンバイ動作では主発振器と非同期タイマの両方が動作を継続します。

AtmelはAVRマイクロコントローラに容量性接触釦、滑動器、輪の機能を組み込むためのQTouch®ライブラリを提供します。特許権を持つ充電転移信号採取は強力な感知を提供し、接触キーの完全な反発運動報告を含み、そしてキー事象の明白な検出のための隣接キー抑制®(AKS™)技術を含みます。簡単に使えるQTouch Suiteツールチェーンはあなた自身の接触応用に対して調査、開発、そしてデバッグを許します。

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ています。内蔵の実装書き換え(ISP)可能なプログラム用フラッシュメモリは規定の不揮発性メモリ書き込み器、SPI直列インターフェース経由、AVRコア上ポートプログラムの実行によって再書き込みができます。ポートプログラムは応用領域フラッシュメモリ内の応用プログラムの読み込みにもどのインターフェースでも使えます。ポート領域フラッシュメモリ内のプログラムは真の「書き込み中の読み出し可」動作により、応用領域フラッシュメモリ更新中も実行を継続します。モリチックチップ上の実装書き換え可能なフラッシュメモリと、8ビットRISC型CPUの組み合わせによるAtmel ATmega128は多くの組み込み制御の応用に対して高度な柔軟性と対費用効果をもたらす強力なマイクロコントローラです。

ATmega128デバイスはCコンパイラ、マクロアセンブラ、デバッグ、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。

## ATmega128とATmega103の互換性

ATmega128はAVR命令一式で予約した64 I/O位置に取って代わる多数のI/O位置を持つ高複合マイクロコントローラです。ATmega103に対して過去との互換性を保証するため、ATmega103に存在する全てのI/O位置はATmega128で同じ位置です。多くの追加I/O位置は\$60から始まり\$FFまでの拡張I/O空間(ATmega103では内蔵SRAM空間)に追加されます。これらの位置はINやOUT命令の使用では届かず、LD/LDS/LDDやST/STS/STD命令の使用でのみ届くことができます。内蔵RAMの再配置はATmega103使用者にとって未だ問題となるかもしれません。同様に割り込みベクタ数の増加はコード(プログラム)が絶対アドレスを使う場合に問題となるでしょう。これらの問題を解決するため、M103Cヒューズをプログラム(0)することにより、ATmega103互換動作が選べます。この動作では拡張I/O領域内のどの機能も使わず、従って内蔵RAMはATmega103のように配置されます。また、拡張した割り込みベクタは取り去られます。

ATmega128はATmega103と100%ピン互換で、現状のプリント基板上のATmega103を置換できます。「ATmega128によるATmega103の置換」応用記述はATmega128によるATmega103置換で使用者が承知すべき事項を記載します。

### ATmega103互換動作 (訳注:本項内容は原文に対して一部補正)

M103Cヒューズのプログラム(0)によって、Atmel® ATmega128は上記のようにRAM、I/Oピン、割り込みベクタに関してATmega103と互換です。けれども、この互換動作でATmega128のいくつかの新機能が利用できません。これらの機能は以下で一覧されます。

- 2つに代わり1つのUSARTで、非同期動作だけです。ポーレートレジスタの下位8ビットだけが利用可能です。
- 3つの比較レジスタ付きの2つの16ビットタイマ/カウンタに代わり、2つの比較レジスタ付きの1つの16ビットタイマ/カウンタです。
- 2線直列インターフェース(TWI)は支援されません。
- ポートCは出力専用です。
- ポートGは交換機能だけを扱います(標準入出力ポート機能なし)。
- ポートFはA/D変換のアナログ入力に加えデジタル入力だけを扱います。
- ポートローダ能力(機能)は支援されません。
- 校正付き内蔵RC発振器の周波数調整ができません。
- 外部メモリインターフェースで、どのアドレスピンも標準入出力ピンに開放できず、異なる外部メモリアドレス領域にどの異なる待ち状態設定もできません。

加えて、よりATmega103と互換にするため、他にいくつかの小規模な相違点があります。

- MCU制御/状態レジスタ(MCUCSR)に外部リセットフラグ(EXTRF)と電源ONリセットフラグ(PORF)だけが存在します。
- ウォッチドッグの計時終了値変更について時間制限による手順が必要ありません。
- 外部割り込み0~3ピンはLowレベル割り込みとして扱います(だけが使えます)。
- USARTはFIFO緩衝部を持たず、従ってより速くデータオーバーランになります。

ATmega128で同じ動作を保証するため、ATmega103での未使用I/Oビットは0を書かれるべきです。

## ピン概要

### VCC

デジタル電源ピン。

### GND

接地ピン。

### PA7～PA0 – (ポートA)

ポートAは(ビット毎に選ばれる)内蔵プルアップ抵抗付き8ビット双方向入出力ポートです。ポートA出力緩衝部は高い吐き出しと吸い込み両方の能力の対称駆動特性です。入力としてプルアップ抵抗が活性(有効)なら、外部的にLowへ引かれたポートAピンは電流を吐き出します。リセット条件が有効になるとクロックが動いていなくてもポートAピンはHi-Zにされます。

ポートAは[45頁](#)で一覧されるATmega128の様々な特殊機能も扱います。

### PB7～PB0 – (ポートB)

ポートBは(ビット毎に選ばれる)内蔵プルアップ抵抗付き8ビット双方向入出力ポートです。ポートB出力緩衝部は高い吐き出しと吸い込み両方の能力の対称駆動特性です。入力としてプルアップ抵抗が活性(有効)なら、外部的にLowへ引かれたポートBピンは電流を吐き出します。リセット条件が有効になるとクロックが動いていなくてもポートBピンはHi-Zにされます。

ポートBは[46頁](#)で一覧されるATmega128の様々な特殊機能も扱います。

### PC7～PC0 – (ポートC)

ポートCは(ビット毎に選ばれる)内蔵プルアップ抵抗付き8ビット双方向入出力ポートです。ポートC出力緩衝部は高い吐き出しと吸い込み両方の能力の対称駆動特性です。入力としてプルアップ抵抗が活性(有効)なら、外部的にLowへ引かれたポートCピンは電流を吐き出します。リセット条件が有効になるとクロックが動いていなくてもポートCピンはHi-Zにされます。

ポートCは[48頁](#)で一覧されるAtmel® AVR® ATmega128の様々な特殊機能も扱います。ATmega103互換動作でのポートCは出力専用でリセット条件が有効になると、Hi-Zにされません。

**注:** ATmega128は既定でATmega103互換動作で出荷されます。従ってPCBへ設置される前に(ATmega103互換禁止に)プログラミングされない場合、ATmega103互換動作が禁止されるまで、ポートCは初回通電で出力になります。

### PD7～PD0 – (ポートD)

ポートDは(ビット毎に選ばれる)内蔵プルアップ抵抗付き8ビット双方向入出力ポートです。ポートD出力緩衝部は高い吐き出しと吸い込み両方の能力の対称駆動特性です。入力としてプルアップ抵抗が活性(有効)なら、外部的にLowへ引かれたポートDピンは電流を吐き出します。リセット条件が有効になるとクロックが動いていなくてもポートDピンはHi-Zにされます。

ポートDは[49頁](#)で一覧されるATmega128の様々な特殊機能も扱います。

### PE7～PE0 – (ポートE)

ポートEは(ビット毎に選ばれる)内蔵プルアップ抵抗付き8ビット双方向入出力ポートです。ポートE出力緩衝部は高い吐き出しと吸い込み両方の能力の対称駆動特性です。入力としてプルアップ抵抗が活性(有効)なら、外部的にLowへ引かれたポートEピンは電流を吐き出します。リセット条件が有効になるとクロックが動いていなくてもポートEピンはHi-Zにされます。

ポートEは[51頁](#)で一覧されるATmega128の様々な特殊機能も扱います。

### PF7～PF0 – (ポートF)

ポートFはA/D変換器へのアナログ入力として扱います。

A/D変換器が使われない場合、ポートFは8ビット双方向入出力ポートとしても扱います。ポートピンは(ビット毎に選ばれる)内蔵プルアップ抵抗を提供できます。ポートF出力緩衝部は高い吐き出しと吸い込みの両能力の対称駆動特性です。入力としてプルアップ抵抗が活性(有効)なら、外部的にLowへ引かれたポートFピンは電流を吐き出します。リセット条件が有効になるとクロックが動いていなくてもポートFピンはHi-Zにされます。JTAGインターフェースが許可されると、リセットが起きてもPF7(TDI)、PF5(TMS)、PF4(TCK)ピンのプルアップ抵抗は活性(有効)にされます。

データを移動出力するTAP状態移行以外、TDOピンはHi-Zにされます。

ポートFはJTAGインターフェース機能も扱います。

ATmega103互換動作でのポートFは入力ポートだけです。

### PG4～PG0 – (ポートG)

ポートGは(ビット毎に選ばれる)内蔵プルアップ抵抗付き5ビット双方向入出力ポートです。ポートG出力緩衝部は高い吐き出しと吸い込み両方の能力の対称駆動特性です。入力としてプルアップ抵抗が活性(有効)なら、外部的にLowへ引かれたポートGピンは電流を吐き出します。リセット条件が有効になるとクロックが動いていなくてもポートGピンはHi-Zにされます。

ポートGは[55頁](#)で一覧されるATmega128の様々な特殊機能も扱います。



ATmega103互換動作でのこれらのピンは外部メモリへのスロープ信号や32kHz発振器への入出力としてだけを扱い、リセット条件が有効になるとクロックが動いていなくても非同期にPG0=1,PG1=1,PG2=0に初期化されます。PG3とPG4は発振器ピンです。

## RESET

リセット入力。最小パルス幅(32頁の表19参照)より長いこのピンのLowレベルはクロックが動いていなくてもリセットを生成します。より短いパルスはリセットの生成が保証されません。

## XTAL1

発振器反転増幅器への入力と内部クロック操作回路への入力。

## XTAL2

発振器反転増幅器からの出力。

## AVCC

AVCCはポートFとA/D変換器用供給電圧(電源)ピンです。例えばA/D変換が使われなくても、外部的にVCCへ接続されるべきです。A/D変換が使われる場合、VCCから低域通過濾波器を通して接続されるべきです。

## AREF

AREFはA/D変換器用アナログ基準(電圧)ピンです。

## PEN

SPI直列プログラミング動作用プログラミング許可ピンで、内部的にHighへプルアップされています。電源ONリセット中、このピンのLow保持によってデバイスはSPI直列プログラミング動作へ移行します。通常動作中、このピンに機能はありません。

## 資料

包括的なデータシート、応用記述、開発ツール群は<http://www.atmel.com/avr>でのダウンロードで利用可能です。

## 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。

## コード例について

この文書はデバイスの様々な部分の使用法を手短か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。

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ファイルがインクルードされることが前提です。拡張I/O領域に配置したI/Oレジスタに対し、IN、OUT、SBIS、SBIC、CBI、SBI命令は拡張I/O領域へのアクセスを許す命令に置き換えられなければなりません。代表的にはSBRS、SBRC、SBR、CBR命令と組み合わせたLDS、STS命令です。

## 容量性接触感知

Atmel QTouchライブラリは殆どのAtmel AVRマイクロコントローラ上の接触感知インターフェースを実現するための使い易い解決策を提供します。QTouchライブラリはQTouchとQMatrix採取法用の支援を含みます。

接触感知は適切なAtmel AVRマイクロコントローラ用QTouchライブラリをリンクすることによってどの応用にも追加することができます。これは接触チャンネルと感知器を定義するのに簡単なAPIの組を使い、その後チャンネル情報を取得して接触感知器の状態を判断するために接触感知APIを呼ぶことによって行われます。

QTouchライブラリは無料で、次の場所、[www.atmel.com/qtouchlibrary](http://www.atmel.com/qtouchlibrary)のAtmelウェブサイトからダウンロードすることができます。実装の詳細と他の情報についてはAtmelウェブサイトからも入手可能なAtmel QTouchライブラリ使用者の手引きを参照してください。

## AVR CPU コア

## 序説

ここでは一般的なAtmel® AVR®コア構造について説明します。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。

## 構造概要

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32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、その結果がレジスタファイルに書き戻されます。

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、32個中の6つのレジスタが3つの16ビット長間接アドレスポイント用レジスタとして使用されます。これらアドレスポイントの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トとしても使えます。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、本章内で後述されます。

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できます。AVR命令の多くは16ビット語(ワード)形式です。全てのプログラムメモリのアドレスは(**訳注**:定数のみを除き)16または32ビット長命令を含みます。

プログラム用フラッシュメモリ空間はブートプログラム領域と応用プログラム領域の2つに分けられます。どちらの領域にも書き込み禁止や読み書き防止用の専用施錠ビットがあります。応用フラッシュメモリ領域内に書き込むSPM命令はブートプログラム領域内に属さ(存在し)なければなりません。

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全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)、スタックポイント(SP)を初期化しなければなりません。SPはI/O空間で読み書き可能です。データ用SRAMはAVR構造で支援される5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。

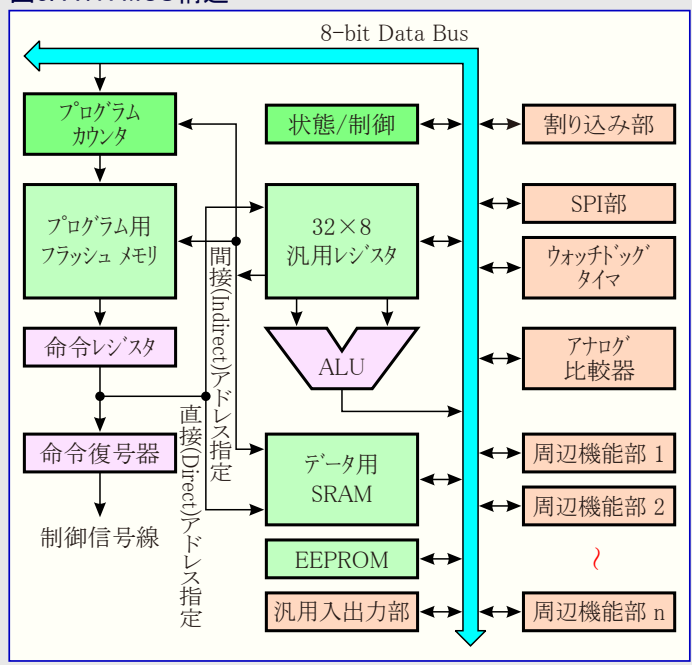
柔軟な割り込み部にはI/O空間の各制御レジスタとステータスレジスタ(SREG)の特別な全割り込み許可(I)ビットがあります。全ての割り込みは割り込みベクタ表に個別の割り込みベクタを持ちます。割り込みには割り込みベクタ表の位置に従う優先順があります。下位側割り込みベクタアドレスが高い優先順位です。

I/Oメモリ空間は制御レジスタ、SPI、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メモリは直接またはレジスタファイルの次のデータ空間位置\$20~\$5Fとしてアクセスできます。加えてATmega128にはST/STS/STDとLD/LDS/LDD命令だけ使えるSRAM内の\$60~\$FFに拡張I/O空間があります。

## ALU (Arithmetic Logic Unit)

高性能なAVRのALUは32個の全汎用レジスタとの直結で動作します。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。

図3. AVR MCU構造



## ステータスレジスタ (Status Register) SREG

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「[命令一式手引書](#)」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。

AVRのステータスレジスタ(SREG)は次のように定義されます。

| ビット         | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |      |
|-------------|-----|-----|-----|-----|-----|-----|-----|-----|------|
| \$3F (\$5F) | I   | T   | H   | S   | V   | N   | Z   | C   | SREG |
| Read/Write  | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |      |
| 初期値         | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |      |

### • ビット7 - I: 全割り込み許可 (Global Interrupt Enable)

全割り込み許可ビットは割り込みが許可されるために設定(1)されなければなりません。その時に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に、**RETI**命令によって設定(1)されます。Iビットは「[命令一式手引書](#)」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。



## 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
- 2つの8ビット出力オペランドと1つの16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

図4.はCPU内の32個の汎用作業レジスタの構造を示します。

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周期命令です。

図4.で示されるように各レジスタは使用者データ空間の最初の32位置へ直接的に配置することで、それらはデータ メモリ アドレスも割り当てられます。例えば物理的にSRAM位置として実装されていなくてもX,Y,Zレジスタ(ポインタ)がレジスタ ファイル内のどのレジスタの指示にも設定できるように、このメモリ構成は非常に柔軟なレジスタのアクセスを提供します。

図4. AVR CPU 汎用レジスタ構成図

|                    |     | 7 | 0 アドレス |      |
|--------------------|-----|---|--------|------|
| 汎用<br>レジスタ<br>ファイル | R0  |   |        | \$00 |
|                    | R1  |   |        | \$01 |
|                    | R2  |   |        | \$02 |
|                    | ⋮   |   |        |      |
|                    | R13 |   |        | \$0D |
|                    | R14 |   |        | \$0E |
|                    | R15 |   |        | \$0F |
|                    | R16 |   |        | \$10 |
|                    | R17 |   |        | \$11 |
|                    | ⋮   |   |        |      |
|                    | R26 |   |        | \$1A |
|                    | R27 |   |        | \$1B |
|                    | R28 |   |        | \$1C |
|                    | R29 |   |        | \$1D |
|                    | R30 |   |        | \$1E |
|                    | R31 |   |        | \$1F |

|       |        |            |
|-------|--------|------------|
| Xレジスタ | 下位8ビット | R26 (\$1A) |
|       | 上位8ビット | R27 (\$1B) |
| Yレジスタ | 下位8ビット | R28 (\$1C) |
|       | 上位8ビット | R29 (\$1D) |
| Zレジスタ | 下位8ビット | R30 (\$1E) |
|       | 上位8ビット | R31 (\$1F) |

## 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5.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固定変位、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「命令一式手引書」をご覧ください)。

図5. 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) | 0 |
|        | 15 | YH (上位)    |   | YL (下位) |            | 0 |
| Y レジスタ | 7  | R29 (\$1D) | 0 | 7       | R28 (\$1C) | 0 |
|        | 15 | ZH (上位)    |   | ZL (下位) |            | 0 |
| Z レジスタ | 7  | R31 (\$1F) | 0 | 7       | R30 (\$1E) | 0 |

## スタック ポインタ (Stack Pointer) SPH,SPL (SP)

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 ポインタレジスタは常にこのスタックの先頭(訳注:次に使われるべき位置)を指し示します。スタックが高位メモリから低位メモリへ伸長するように実行されることに注意してください。これはスタックへのPUSH命令はスタック ポインタを減少するという意味です。

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。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。スタック ポインタは\$100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PUSH命令でデータがスタックに格納されると-1され、サブルーチン呼び出しや割り込みで戻りアドレスがスタックに格納されると-2されます。スタック ポインタはPOP命令でデータがスタックから引き出されると+1され、サブルーチンからの復帰(RET)命令や割り込みからの復帰(RETI)命令でアドレスがスタックから引き出されると+2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。

| ビット         | 15   | 14   | 13   | 12   | 11   | 10   | 9   | 8   |     |
|-------------|------|------|------|------|------|------|-----|-----|-----|
| \$3E (\$5E) | SP15 | SP14 | SP13 | SP12 | SP11 | SP10 | SP9 | SP8 | SPH |
| Read/Write  | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  | R/W | R/W |     |
| 初期値         | 0    | 0    | 0    | 0    | 0    | 0    | 0   | 0   |     |
| ビット         | 7    | 6    | 5    | 4    | 3    | 2    | 1   | 0   |     |
| \$3D (\$5D) | SP7  | SP6  | SP5  | SP4  | SP3  | SP2  | SP1 | SP0 | SPL |
| Read/Write  | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  | R/W | R/W |     |
| 初期値         | 0    | 0    | 0    | 0    | 0    | 0    | 0   | 0   |     |

## RAMページ選択レジスタ (RAM Page Z Select Register) RAMPZ

| ビット         | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0      |       |
|-------------|---|---|---|---|---|---|---|--------|-------|
| \$3B (\$5B) | - | - | - | - | - | - | - | RAMPZ0 | RAMPZ |
| Read/Write  | R | R | R | R | R | R | R | R/W    |       |
| 初期値         | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0      |       |

### • ビット7～1 – Res : 予約 (Reserved)

これらのビットは予約されており、常に0として読みます。将来のデバイスとの互換性のため、このアドレスへの書き込み時、これらのビットに0を書いてください。

### • ビット0 – RAMPZ0 : 拡張RAMページ Zポインタ (Extended RAM Page Z-pointer)

通常、RAMページ選択レジスタ(RAMPZ)はZポインタ(レジスタ)によって(間接)アクセスされる64Kバイト(単位の)RAMページを選ぶのに使われます。ATmega128が64Kバイトより多くのSRAMメモリ(データメモリ空間)を支援しないので、この(RAMPZ)レジスタはELPM/SPM命令が使われる時にアクセスされるプログラムメモリのページを選ぶためだけに使われます。RAMPZ0ビット設定は次の効果があります。

RAMPZ0=0 : プログラムメモリアドレス\$0000～\$7FFF(下位64Kバイト)がELPM/SPM命令によってアクセスされます。

RAMPZ0=1 : プログラムメモリアドレス\$8000～\$FFFF(上位64Kバイト)がELPM/SPM命令によってアクセスされます。

LPM命令がRAMPZ設定によって影響を及ぼされないことに注意してください。

## 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)用に選んだ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。

図6.はハ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、機能対電源部についての好結果と対応するMHzあたり1 MIPSを達成するための基本的なパイプラインの概念です。

図7.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6. 命令の取得と実行の並列動作

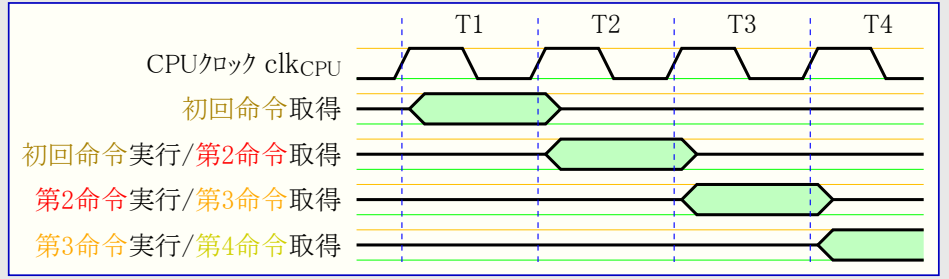
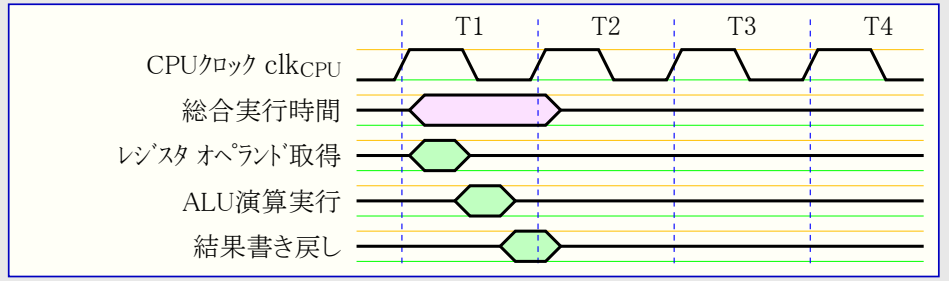


図7. 1周期ALU命令



## 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。BLB02またはBLB12ブート施錠ビットがプログラム(0)されると、プログラムカウンタ値によっては割り込みが自動的に禁止されるかもしれません。この特質はソフトウェア保護を改善します。詳細については187頁の「**メモリプログラミング**」章をご覧ください。

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37頁の「**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。割り込みベクタは**MCU制御レジスタ(MCUCR)の割り込みベクタ選択(IVSEL)ビット**の設定(1)によって**ブートフラッシュ領域先頭**へ移動できます。より多くの情報については37頁の「**割り込み**」を参照してください。リセットベクタも**BOOTRSTヒューズのプログラム(0)**によってブートフラッシュ領域先頭へ移動できます。178頁の「**ブートローダ支援 - 書き込み中読み出し可能な自己プログラミング**」をご覧ください。

割り込みが起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ると、割り込み要求フラグが設定(1)され、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なことに注意してください。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。

割り込みを禁止するために**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っても、**CLI**命令後に割り込みは実行されません。次例は時間制限EEPROM書き込み手順中に割り込みを無効とするために、これがどう使えるかを示します。

### アセンブリ言語プログラム例

```
IN      R16, SREG          ;ステータスレジスタを保存
CLI     ;EEPROM書き込み手順中割り込み禁止
SBI     EECR, EEMWE       ;EEPROM主書き込み許可
SBI     EECR, EEWE        ;EEPROM書き込み開始
OUT     SREG, R16        ;ステータスレジスタを復帰
```

### C言語プログラム例

```
char cSREG;                /* ステータスレジスタ保存変数定義 */
cSREG = SREG;              /* ステータスレジスタを保存 */
__disable_interrupt();    /* EEPROM書き込み手順中割り込み禁止 */
EECR |= (1<<EEMWE);       /* EEPROM主書き込み許可 */
EECR |= (1<<EEWE);        /* EEPROM書き込み開始 */
SREG = cSREG;             /* ステータスレジスタを復帰 */
```

割り込みを許可するために**SEI**命令を使うと、次例で示されるようにどの保留割り込みにも先立って**SEI**命令の次の命令が実行されません。

### アセンブリ言語プログラム例

```
SEI     ;全割り込み許可
SLEEP  ;休止形態移行(割り込み待ち)
```

### C言語プログラム例

```
__enable_interrupt();     /* 全割り込み許可 */
__sleep();                /* 休止形態移行(割り込み待ち) */
```

**注:** SLEEP命令までは割り込み禁止、保留割り込み実行前に休止形態へ移行します。

## 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クロック周期時間中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3クロック周期要します。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。この増加は選んだ休止形態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。

## AVR ATmega128のメモリ

この項はAtmel® AVR® ATmega128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があります。加えてATmega128はデータ保存用EEPROMメモリが特徴です。3つのメモリ空間全ては一般的な直線的アドレスです。

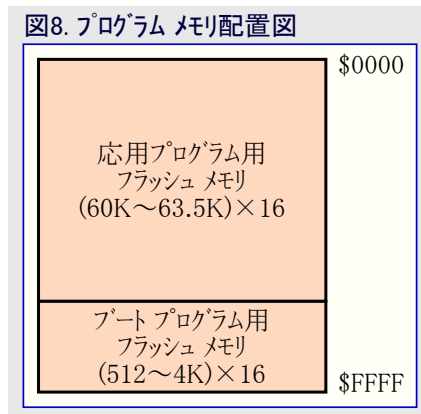
### 実装自己書き換え(ISP: In-System Program)可能なプログラム用フラッシュメモリ

ATmega128はプログラム保存用に実装書き換え可能な128K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64K×16ビットとして構成されます。ソフトウェア保護のため、フラッシュプログラムメモリ空間はブートプログラム領域と応用プログラム領域の2つに分けられます。

フラッシュメモリは最低10,000回の消去/書き込み回数の耐久性があります。ATmega128のプログラムカウンタ(PC)は16ビット幅で、従って64Kプログラムメモリ位置のアドレス指定です。ブートプログラム領域の操作と関係するソフトウェア保護用ブート施錠ビットは178頁の「ブートローダ支援 - 書き込み中読み出し可能な自己プログラミング」の詳細で記述されます。187頁の「メモリプログラミング」はSPI、JTAG、並列プログラミング動作でのフラッシュメモリプログラミングの詳細な記述を含みます。

定数表は全てのプログラムメモリアドレス空間に配置できます。(LPM命令とELPM命令記述参照)

命令の取得と実行のタイミング図は10頁の「命令実行タイミング」で示されます。





## データ用SRAM メモリ

Atmel® AVR® ATmega128は表1.で一覧されるように2つの異なるSRAMデータメモリ用設定を支援します。

図9.はATmega128のSRAM(データ空間)メモリ構成方法を示します。

ATmega128はINやOUT命令で予約した64位置で支援されるより多くの周辺機能部を持つ複合マイクロコントローラです。SRAM(データ空間)内\$60~\$FFの拡張I/O空間に対してLD/LDS/LDDとST/STS/STD命令だけが使えます。ATmega128がATmega103互換動作のとき、この拡張I/O領域は存在しません。

標準動作で、最初の4352データメモリ位置はレジスタファイル、I/Oメモリ、拡張I/Oメモリ、データ用内蔵SRAMに充てます。先頭の32位置はレジスタファイル、次の64位置は標準I/Oメモリ、その次の160位置は拡張I/Oメモリ、そして次の4096位置はデータ用内蔵SRAMに充てます。

ATmega103互換動作で、最初の4096データメモリ位置はレジスタファイル、I/Oメモリと内蔵SRAMに充てます。先頭の32位置はレジスタファイル、次の64位置は標準I/Oメモリ、そして次の4000位置はデータ用内蔵SRAMに充てます。

任意の外部データSRAMがATmega128で使えます。このSRAMは64Kアドレス空間内の残りのアドレス領域を占有します。この領域は内蔵SRAMに続くアドレスから始まります。レジスタファイル、I/O領域、拡張I/O領域、内蔵SRAMが標準動作で最下位4352バイトを、ATmega103互換動作(拡張I/O領域なし)で最下位4096バイトを占有するので、64K(65536)バイトの外部メモリ使用時、標準動作で61184バイト、ATmega103互換動作で61440バイトの外部メモリが利用可能です。外部メモリ配置の利用法の詳細については18頁の「外部メモリインターフェース」をご覧ください。

アドレスが内部データメモリ領域を越えるSRAMメモリ空間をアクセスするとき、外部データSRAMは内部データメモリアクセスと同じ命令を使ってアクセスされます。内部データメモリがアクセスされる時、読みと書きのストロブピン(PG1とPG0)はそのアクセス周期中全てで無効です。外部SRAM操作はMCU制御レジスタ(MCUCR)の外部SRAM許可(SRE)ビットの設定(1)によって許可されます。

外部SRAMのアクセスは内蔵SRAMのアクセスに比べ1バイト毎に1追加クロック周期を要します。これは(外部SRAMへの)LD,ST,LDS,STS,LDD,STD,PUSH,POP命令が1追加クロック周期を要することを意味します。スタックが外部SRAMに配置されると、割り込みやサブルーチンの呼び出し/復帰時、2バイトのプログラムカウンタ(PC)が保存(プッシュ)/復帰(ポップ)され、そして内部メモリアクセスでのパイプラインの優位性がなくなるため、2+1クロック周期の追加を要します。待ち状態付きで外部メモリインターフェースが使われると、1バイトの外部アクセスは1,2,3待ち状態に対し、各々(内部SRAMアクセスに対して)2,3,4追加クロック周期を要します。割り込みやサブルーチンの呼び出しと復帰は1,2,3待ち状態に対し、命令一式手引書で示される値よりも5,7,9クロック周期多く必要です。

直接、間接、変位付き間接、事前減少付き間接、事後増加付き間接の5つの異なるアドレス指定種別でデータメモリ(空間)を網羅します。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が特徴です。

直接アドレス指定はデータ空間全体に届きます。

変位付き間接動作はYまたはZレジスタで与えられる基準アドレスからの63アドレス位置に届きます。

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(使われる)X,Y,Zアドレスレジスタは減少(-1)または増加(+1)されます。

ATmega128の32個の汎用レジスタ、64個のI/Oレジスタ、160個の拡張I/Oレジスタ、4096バイトのデータ用内蔵SRAMは、これら全てのアドレス指定種別を通して全部アクセスできます。レジスタファイルは9頁の「汎用レジスタファイル」で記述されます。

## データメモリアクセスタイミング

この項は内部メモリアクセスに対する一般的なアクセスタイミングの概念を記述します。データ用内蔵SRAMアクセスは図10.に記載されるように2 clkCPU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。

表1. SRAM/データメモリ設定

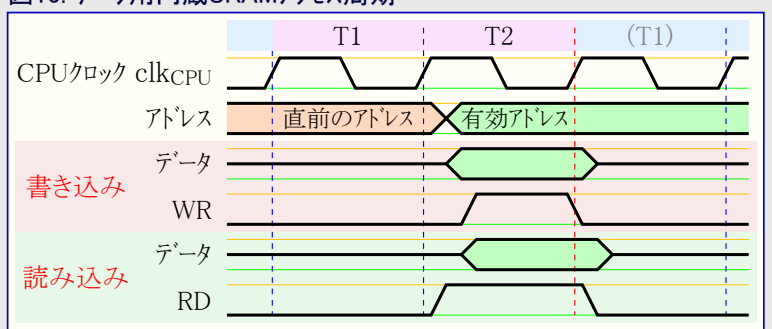
| 設定          | 内蔵SRAM | 外部メモリ   |
|-------------|--------|---------|
| 標準          | 4096   | ~59.75K |
| ATmega103互換 | 4000   | ~60K    |

図9. データ空間とSRAMの配置

| 標準動作                   |        |        | アドレス                  | ATmega103互換動作       |        |        | アドレス                  |
|------------------------|--------|--------|-----------------------|---------------------|--------|--------|-----------------------|
| レジスタファイル<br>(32×8)     | R0     | \$0000 | \$0000<br>~<br>\$001F | レジスタファイル<br>(32×8)  | R0     | \$0000 | \$0000<br>~<br>\$001F |
|                        | R31    | \$001F |                       |                     | R31    | \$001F |                       |
| I/Oレジスタ<br>(64×8)      | \$00   | \$0020 | \$0020<br>~<br>\$005F | I/Oレジスタ<br>(64×8)   | \$00   | \$0020 | \$0020<br>~<br>\$005F |
|                        | \$3F   | \$005F |                       |                     | \$3F   | \$005F |                       |
| 拡張I/Oレジスタ<br>(160×8)   | \$0060 | \$0060 | \$0060<br>~<br>\$00FF | 内蔵SRAM<br>(4000×8)  | \$0060 | \$0060 | \$0060<br>~<br>\$0FFF |
|                        | \$00FF | \$00FF |                       |                     | \$0FFF | \$0FFF |                       |
| 内蔵SRAM<br>(4096×8)     | \$0100 | \$0100 | \$0100<br>~<br>\$10FF | 外部SRAM<br>(0~60K×8) | \$1000 | \$1000 | \$1000<br>~<br>\$FFFF |
|                        | \$10FF | \$10FF |                       |                     | \$FFFF | \$FFFF |                       |
| 外部SRAM<br>(0~59.75K×8) | \$1100 | \$1100 | \$1100<br>~<br>\$FFFF |                     |        |        |                       |
|                        | \$FFFF | \$FFFF |                       |                     |        |        |                       |

注: 赤字はI/Oアドレス

図10. データ用内蔵SRAMアクセス周期





## データ用EEPROMメモリ

Atmel® AVR® ATmega128は4KバイトのデータEEPROMを含みます。それは単一バイトが読み書きできる分離したデータ空間として構成されます。EEPROMは最低100,000回の消去/書き込み回数の耐久性があります。CPUとEEPROM間のアクセスは以降のEEPROMアドレスレジスタ、EEPROMデータレジスタ、EEPROM制御レジスタで詳細に記述されます。

187頁の「メモリプログラミング」はSPI、JTAG、並列プログラミング動作でのEEPROMプログラミングの詳細な記述を含みます。

### EEPROMアクセス

EEPROMアクセスレジスタはI/O空間でアクセス可能です。

EEPROMの書き込み(訳注:原文はアクセス)時間は15頁の表2.で与えられます。(書き込みは)自己タイミング機能ですが、使用者ソフトウェアは**次バイトが書ける時を検知**してください。使用者コードがEEPROMに書く命令を含む場合、いくつかの予防処置が取られなければなりません。嚴重に濾波した電源では電源投入/切断でVCCが緩やかに上昇または下降しそうです。これはデバイスが何周期かの間、使われるクック周波数に於いて最小として示されるより低い電圧で走行する原因になります。これらの状態で問題を避ける方法の詳細については16頁の「EEPROMデータ化けの防止」をご覧ください。

予期せぬEEPROM書き込みを防止するため、特別な書き込み手順に従わなければなりません。この詳細については「EEPROM制御レジスタ」の記述を参照してください。

EEPROMが読まれると、CPUは次の命令が実行される前に4クロック周期停止されます。EEPROMが書かれると、CPUは次の命令が実行される前に2クロック周期停止されます。

### EEPROMアドレスレジスタ (EEPROM Address Register) EEARH,EEARL (EEAR)

| ビット         | 15    | 14    | 13    | 12    | 11     | 10     | 9     | 8     |       |
|-------------|-------|-------|-------|-------|--------|--------|-------|-------|-------|
| \$1F (\$3F) | –     | –     | –     | –     | EEAR11 | EEAR10 | EEAR9 | EEAR8 | EEARH |
| Read/Write  | R     | R     | R     | R     | R/W    | R/W    | R/W   | R/W   |       |
| 初期値         | 0     | 0     | 0     | 0     | 不定     | 不定     | 不定    | 不定    |       |
| ビット         | 7     | 6     | 5     | 4     | 3      | 2      | 1     | 0     |       |
| \$1E (\$3E) | EEAR7 | EEAR6 | EEAR5 | EEAR4 | EEAR3  | EEAR2  | EEAR1 | EEAR0 | EEARL |
| Read/Write  | R/W   | R/W   | R/W   | R/W   | R/W    | R/W    | R/W   | R/W   |       |
| 初期値         | 不定    | 不定    | 不定    | 不定    | 不定     | 不定     | 不定    | 不定    |       |

#### • ビット15～12 – Res : 予約 (Reserved)

これらのビットは予約されており、常に0として読みます。このアドレス位置へ書く時、将来のデバイスとの共通性のため、これらのビットに0を書いてください。

#### • ビット11～0 – EEAR11～0 : EEPROMアドレス (EEPROM Address)

EEPROMアドレスレジスタ(EEARHとEEARL)は4096バイトEEPROM空間のEEPROMアドレスを指定します。EEPROMデータバイトは0～4095間で直線的に配されます。EEARの初期値は不定です。EEPROMがアクセスされるであろう前に適切な値が書かれねばなりません。

### EEPROMデータレジスタ (EEPROM Data Register) EEDR

| ビット         | 7     | 6   | 5   | 4   | 3   | 2   | 1   | 0     |      |
|-------------|-------|-----|-----|-----|-----|-----|-----|-------|------|
| \$1D (\$3D) | (MSB) |     |     |     |     |     |     | (LSB) | EEDR |
| Read/Write  | R/W   | R/W | R/W | R/W | R/W | R/W | R/W | R/W   |      |
| 初期値         | 0     | 0   | 0   | 0   | 0   | 0   | 0   | 0     |      |

#### • ビット7～0 – EEDR7～0 : EEPROMデータ (EEPROM Data)

EEPROM書き込み操作に対してEEDRはEEPROMアドレスレジスタ(EEAR)で与えたアドレスのEEPROMへ書かれるべきデータを含みます。EEPROM読み込み操作に対してEEDRはEEARで与えたアドレスのEEPROMから読み出したデータを含みます。

### EEPROM制御レジスタ (EEPROM Control Register) EECR

| ビット         | 7 | 6 | 5 | 4 | 3     | 2     | 1    | 0    |      |
|-------------|---|---|---|---|-------|-------|------|------|------|
| \$1C (\$3C) | – | – | – | – | EERIE | EEMWE | EEWE | EERE | EECR |
| Read/Write  | R | R | R | R | R/W   | R/W   | R/W  | R/W  |      |
| 初期値         | 0 | 0 | 0 | 0 | 0     | 0     | 不定   | 0    |      |

#### • ビット7～4 – Res : 予約 (Reserved)

これらのビットは予約されており、常に0として読みます。

#### • ビット3 – EERIE : EEPROM操作可割り込み許可 (EEPROM Ready Interrupt Enable)

EERIEの1書き込みは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)されているなら、EEPROM操作可割り込みを許可します。EERIEの0書き込みはこの割り込みを禁止します。EEPROM操作可割り込みはEEWEが解除(0)されていると継続する割り込みを発生します。

### •ビット2 – EEMWE : EEPROM主書き込み許可 (EEPROM Master Write Enable)

EEMWEビットはEEPROM書き込み許可(EEMWE)ビットの1設定がEEPROM書き込みの原因となるかどうかを決めます。EEMWEが設定(1)されるとき、EEMWEの1設定は選ばれたアドレスのEEPROMにデータを書きます。EEMWEが0の場合、EEMWEの1設定は無効です。EEMWEがソフトウェアによって設定(1)されてしまうと、4クロック周期後にハードウェアがこのビットを0に解除します。EEPROM書き込み手順については「書き込み許可(EEMWE)ビット」の記述をご覧ください。

### •ビット1 – EEWWE : EEPROM書き込み許可 (EEPROM Write Enable)

EEPROM書き込み許可信号(EEMWE)はEEPROMへの書き込みスローブです。アドレスとデータが正確に設定されると、EEPROMへその値を書くためにEEWEビットは1を書かれなければなりません。論理1がEEWEへ書かれる前にEEPROM主書き込み許可(EEMWE)ビットは1を書かれなければならず、さもなければEEPROM書き込みは行われません。EEPROMを書くとき、次の手順に従うべきです(手順3.と4.の順番は重要ではありません)。

1. EEPROM書き込み許可(EEMWE)ビットが0になるまで待ちます。
2. SPM制御/状態レジスタ(SPMCSR)のSPM操作許可(SPMEN)ビットが0になるまで待ちます。
3. 今回のEEPROMアドレスをEEPROMアドレスレジスタ(EEAR)に書きます。(任意、省略可)
4. 今回のEEPROMデータをEEPROMデータレジスタ(EEDR)に書きます。(任意、省略可)
5. EEPROM制御レジスタ(EECR)のEEPROM主書き込み許可(EEMWE)ビットに1を、EEPROM書き込み許可(EEMWE)ビットに0を同時に書きます。
6. EEMWEビット設定後4クロック周期内にEEPROM書き込み許可(EEMWE)ビットへ論理1を書きます。

CPUがフラッシュメモリ書き込み中、EEPROMはプログラム(書き込み)ができません。ソフトウェアは新規EEPROM書き込みを始める前にフラッシュメモリのプログラミングが完了されていることを確認しなければなりません。2.はソフトウェアがフラッシュメモリをプログラム(書き込み)することをCPUに許すブートローダを含む場合だけ関係します。フラッシュメモリが決してCPUによって更新されないなら、2.は省略できます。ブートプログラミングについての詳細に関しては178頁の「ブートローダ支援 – 書き込み中読み出し可能な自己プログラミング」をご覧ください。

**警告:** 手順5.と6.間の割り込みはEEPROM主書き込み許可が時間超過するため、書き込み周期失敗になります。EEPROMをアクセスする割り込みルーチンが他のEEPROMアクセスを中断し、EEARかEEDRが変更されると、中断したEEPROMアクセスを失敗させます。これらの問題を避けるため、全ての手順中、ステータスレジスタ(SREG)の全割り込み許可(I)ビットは解除(0)されていることが推奨されます。

書き込みアクセス時間が経過されると、EEPROM書き込み許可(EEMWE)ビットはハードウェアによって解除(0)されます。ユーザーソフトウェアは次バトを書く前にこのビットをポーリングし、0を待つことができます。EEMWEが設定(1)されてしまうと、次の命令が実行される前にCPUは2周期停止されます。

### •ビット0 – EERE : EEPROM読み込み許可 (EEPROM Read Enable)

EEPROM読み込み許可信号(EERE)はEEPROMへの読み込みスローブです。EEARに正しいアドレスが設定されると、EEPROM読み出しを起動するためにEEREビットは論理1を書かれなければなりません。EEPROM読み出しアクセスは(その)1命令で行われ、要求したデータは直ちに利用できます。EEPROMが読まれるとき、次の命令が実行される前にCPUは4周期停止されます。

ユーザーは読み込み操作を始める前にEEMWEビットをポーリングすべきです。書き込み操作が実行中の場合、EEPROMアドレスレジスタ(EEAR)の変更もEEPROM読み込みもできません。

EEPROMアクセスの時間に校正済み内蔵RC発振器が使われません。表2.はCPUからのEEPROMアクセスに対する代表的な書き込み時間を示します。

表2. EEPROM書き込み時間

| 項目              | 校正付き内蔵RC発振器周期数 (注) | 代表    |
|-----------------|--------------------|-------|
| EEPROM書き込み(CPU) | 8448               | 8.5ms |

注: CKSELヒューズ設定と無関係に1MHzが使われます。

次のコード例はアセンブリ言語とC言語でのEEPROM書き込み関数を示します。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、これらの関数実行中に割り込みが起きない前提です。本例はソフトウェア内にフラッシュブートローダが無い前提でもあります。そのようなコードが存在する場合、EEPROM書き込み関数は何れかが実行するSPM命令の完了も待たねばなりません。

#### アセンブリ言語プログラム例

```
EEPROM_WR:  SBIC    EECR, EEWE          ;EEPROM書き込み完了ならばスキップ
            RJMP   EEPROM_WR        ;以前のEEPROM書き込み完了まで待機
;
            OUT    EEARH, R18        ;EEPROMアドレス上位バイト設定
            OUT    EEARL, R17        ;EEPROMアドレス下位バイト設定
            OUT    EEDR, R16         ;EEPROM書き込み値を設定
            SBI    EECR, EEMWE       ;EEPROM主書き込み許可ビット設定
            SBI    EECR, EEWE        ;EEPROM書き込み開始(書き込み許可ビット設定)
            RET                       ;呼び出し元へ復帰
```

#### C言語プログラム例

```
void EEPROM_write(unsigned int uiAddress, unsigned char ucData)
{
    while(EECR & (1<<EEWE));          /* 以前のEEPROM書き込み完了まで待機 */
    EEAR = uiAddress;                 /* EEPROMアドレス設定 */
    EEDR = ucData;                    /* EEPROM書き込み値を設定 */
    EECR |= (1<<EEMWE);               /* EEPROM主書き込み許可 */
    EECR |= (1<<EEWE);                /* EEPROM書き込み開始 */
}
```

次のコード例はアセンブリ言語とC言語でのEEPROM読み込み関数を示します。本例は割り込みが制御され、これら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

```
EEPROM_RD:  SBIC    EECR, EEWE          ;EEPROM書き込み完了ならばスキップ
            RJMP   EEPROM_RD        ;以前のEEPROM書き込み完了まで待機
;
            OUT    EEARH, R18        ;EEPROMアドレス上位バイト設定
            OUT    EEARL, R17        ;EEPROMアドレス下位バイト設定
            SBI    EECR, EERE        ;EEPROM読み出し開始(読み込み許可ビット設定)
            IN     R16, EEDR         ;EEPROM読み出し値を取得
            RET                       ;呼び出し元へ復帰
```

#### C言語プログラム例

```
unsigned char EEPROM_read(unsigned int uiAddress)
{
    while(EECR & (1<<EEWE));          /* 以前のEEPROM書き込み完了まで待機 */
    EEAR = uiAddress;                 /* EEPROMアドレス設定 */
    EECR |= (1<<EERE);                /* EEPROM読み出し開始 */
    return EEDR;                      /* EEPROM読み出し値を取得,復帰 */
}
```

## パワーダウン休止動作中のEEPROM書き込み

EEPROM書き込み動作が活動中にパワーダウン休止動作へ移行すると、EEPROM書き込み動作が継続し、EEPROM書き込み時間が過ぎ去ってしまう前に完了します。しかし、書き込み動作が完了されると、発振器が動作を継続し、結果としてデバイスはパワーダウン動作へ完全に移行しません。従ってパワーダウンへ移行する前に、EEPROM書き込み動作が完了される(EEWE=0)ことの確認が推奨されます。(訳補: パワーダウン移行後もEEPROM書き込みは正常に完了するが、その後発振器が止まらないことへの注意)

## EEPROMデータ化けの防止

低VCCの期間中、正しく動作するための供給電圧がCPUとEEPROMに対して低すぎるためにEEPROMデータが化け得ます。これらの問題はEEPROMを使う基板段階の装置と同じで、同じ設計上の解決策が適用されるべきです。

EEPROMデータ化けは電圧が低すぎる時の2つの状態によって起こされ得ます。1つ目として、EEPROMへの通常の書き込み手順は正しく動作するための最低電圧が必要です。2つ目として、供給電圧が低すぎると、CPU自身が命令を間違って実行し得ます。

EEPROMデータ化けは次の推奨設計によって容易に避けられます。

不十分な供給電源電圧の期間中、AVRのRESETを活性(Low)に保ってください。これは内蔵低電圧検出器(BOD)を許可することによって行えます。書き込み動作実行中にリセットが起こると、この書き込み操作は供給電源電圧が充分ならば(継続)完了されます。

## I/O メモリ (レジスタ)

ATmega128のI/O空間定義は234頁の「[レジスタ要約](#)」で示されます。

Atmel® AVR® ATmega128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32個の汎用作業レジスタ間のデータ転送を行うLD/LDS/LDD命令とST/STS/STD命令によってアクセスされます。アドレス範囲\$00～\$1F内のI/OレジスタはSBI命令とCBI命令の使用で直接的にビットアクセス可能です。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。I/O指定命令INとOUTを使う時はI/Oアドレス\$00～\$3Fが使われなければなりません。LD命令とST命令を使い、データ空間としてI/Oレジスタをアクセスするとき、これらのアドレスに\$20が加算されなければなりません。

ATmega128はINやOUT命令で予約された64位置で支援されるより多くの周辺機能部を持つ複合マイクロコントローラです。SRAM(データ空間)内\$60～\$FFの拡張I/O領域に対してはLD/LDS/LDDとST/STS/STD命令だけが使えます。ATmega128がATmega103互換動作の時は拡張I/O領域がSRAM領域に置き換えられます。

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I/Oレジスタ内の全ビットを操作し、設定(1)として読むどのフラグにも1を書き戻し、従ってフラグを解除(0)す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。

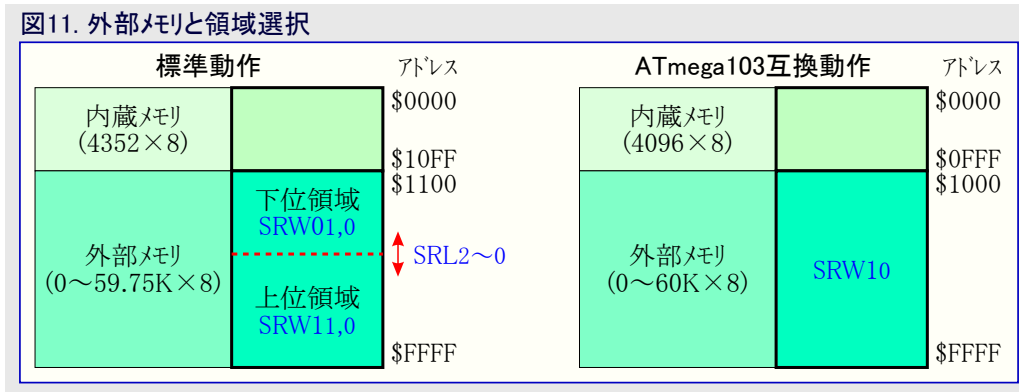
## 外部メモリ インターフェース

外部メモリ インターフェースが提供する全ての機能で、LCD表示器やA/D、D/Aのような周辺装置や、外部SRAMやフラッシュメモリのようなメモリデバイスへのインターフェースとしての動作に上手く仕立てられます。主な特徴を次に示します。

- 4つの異なる待ち状態設定(含む待ち状態なし)
- 異なる外部メモリ領域(領域容量設定可)に対する個別の待ち状態設定
- アドレス上位バイトに割り当てるビット数が選択可能
- 消費電流を最小とするため、データ線のバス保持機能(任意選択)

### 概要

外部メモリ(XMEM)が許可されると、内蔵SRAMを越えるアドレス空間は割り当てた外部メモリピンを使うことが利用可能になります(2頁の「ピン配置」、45頁の表27、48頁の表33、55頁の表45をご覧ください)。メモリ設定は図11.で示されます。



### ATmega103互換動作

両方の外部メモリ制御レジスタ(XMCRAとXMCRB)は拡張I/O空間に配置されます。ATmega103互換動作で、これらのレジスタは利用できず、これらのレジスタによって選ばれる特徴(機能)は利用できません。これらの特徴(機能)がATmega103に存在しないので、デバイスはATmega103互換です。ATmega103互換動作での制限を次に示します。

- 2待ち状態設定のみ利用可能 (SRW1n=00とSRW1n=01)
- アドレス上位バイトへの割り当てビット数固定
- 外部メモリ空間は異なる待ち状態の領域分割不可
- バス保持機能利用不可
- RD, WR, ALEピンは出力専用 (ATmega128のポートG)

### 外部メモリ インターフェースの使用

このインターフェースは以下から成ります。

- AD7~0 : 交互切り替えされた下位アドレスバスとデータバス
- A15~8 : 上位アドレスバス(ビット数設定可能)
- ALE : アドレスラッチ許可信号
- RD : 読み出しストロブ信号
- WR : 書き込みストロブ信号

外部メモリ(XMEM)インターフェースに関する制御ビットは3つのレジスタ、MCU制御レジスタ(MCUCR)、外部メモリ制御レジスタA(XMCRA)、外部メモリ制御レジスタB(XMCRB)に配置されます。

XMEMインターフェースが許可されると、このインターフェースに割り当てたポートに対応するデータ方向レジスタ設定を無効にします。このポートの無効化についての詳細に関しては41頁の「入出力ポート」項の交換機能をご覧ください。XMEMインターフェースはアクセスが内部または外部どちらかを自動的に検知します。アクセスが外部の場合、XMEMインターフェースは図13.(本図は待ち状態なしの波形を示す)に従ったアドレス、データ、制御信号を出力します。ALEがHigh→Low(下降)になるとAD7~0上のアドレスが有効です。データ転送中、ALEはLowです。XMEMインターフェースが許可されると、内部アクセスでも同様にアドレス、データ、ALEポートの動きを引き起こしますが、内部アクセス中、RDとWRストロブは切り替わりません。外部メモリインターフェースが禁止されると、標準ピンとデータ方向設定が使われます。XMEMインターフェースが禁止される時に内蔵SRAM境界以上のアドレス空間が内蔵SRAMに割り当てられないことに注意してください。図12.はGがHighの時に透過(通過:トランスペアレント)となる8ビットのラッチ(一般的には74xx573または同等品)を使うAVRと外部SRAMの接続法を図解します。



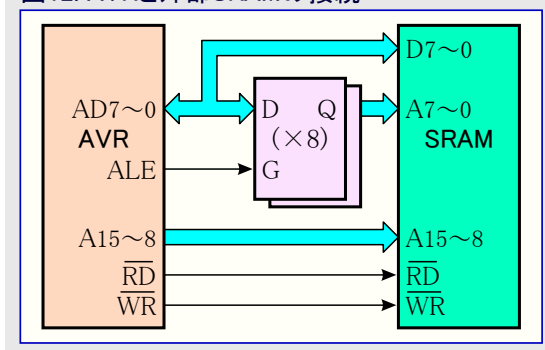
## アドレス ラッチの必要性

XMEMインターフェースの高速動作のため、アドレス ラッチは4MHz/2.7V、8MHz/4V以上のシステム周波数に対し、注意して選ばれなければなりません。これらの周波数以上の条件で動作するとき、代表的な旧デバイス74HCシリーズのラッチは不適切になります。外部メモリ インターフェースは74AHC系のラッチに適應するよう設計されています。けれども主なタイミング パラメータに対応してさえいけば、多くのラッチが使えます。アドレス ラッチについての主なパラメータは次のとおりです。

- DからQへの伝播遅延時間 ( $t_{PD}$ ) (訳注) 記号はAVR側ではなく、一般的なラッチ デバイス側での記号です。
- G下降端に対するデータ準備時間 ( $t_{SU}$ )
- G下降後のデータ(アドレス)保持遅延時間 ( $t_H$ )

外部メモリ インターフェースはGがLowにされた後、 $t_H=5ns$ の最小アドレス保持時間を保証するよう設計されています(211～213頁、表137.～144.の $t_{LAXX\_LD}$ と $t_{LAXX\_ST}$ を参照してください)。DからQへの伝播遅延時間( $t_{PD}$ )は外部部品のアクセス時間必要条件を計算する時に考慮されなければなりません。G=Low(下降)前のデータ準備時間( $t_{SU}$ )はALE=Low(下降)前のアドレス有効時間( $t_{AVLLC}$ )－PCB配線遅延(容量性負荷に依存)を越えてはいけません。

図12. AVRと外部SRAMの接続



## プルアップとバス保持機能

AD7～0ポートのプルアップ抵抗は対応するポートレジスタが1を書かれると、活性(有効)にできます。休止形態での消費電力を削減するため、休止形態移行前にポートレジスタへ0を書くことによってプルアップを禁止することが推奨されます。

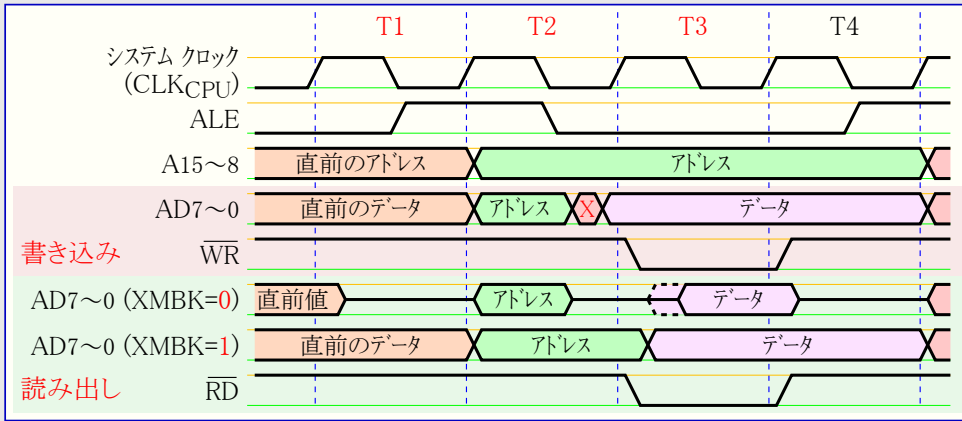
XMEMインターフェースはAD7～0線のバス保持機能も提供します。バス保持機能は22頁の「外部メモリ制御レジスタB(XMCRB)」で記述されるようにプログラムで許可と禁止ができます。許可されると、これらの線がXMEMインターフェースにより、その他の状態であるHi-Zにされる時にバス保持機能はAD7～0バスでの論理レベル(0または1)固定を保証します。

## タイミング

外部メモリ デバイスは様々なタイミング 必要条件を持ちます。これらの必要条件に合わせるため、ATmega128のXMEMインターフェースは表4.に示される4つの異なる待ち状態を提供します。この待ち状態選択前に外部メモリ デバイスのタイミング特性を考慮することが大事です。重要なパラメータの多くはATmega128の必要条件設定に関連する外部メモリのアクセス時間です。外部メモリのアクセス時間はチップ選択とアドレスを受け取ってから、そのアドレスのデータがバス上で実際に駆動されるまでの時間で定義されます。このアクセス時間はALEパルスがLowにされてから読み出し手順中にデータが安定にならなければならないまでの時間(211～212頁の表137.～144.の $t_{LLRL}+t_{RLRH}-t_{DVRH}$ )を超えることはできません。各種待ち状態はソフトウェアで設定されます。付加機能として、外部メモリ空間を個別に待ち状態設定できる2つの領域に分割することが可能です。これは異なるタイミング 必要条件の2つの異なるメモリ デバイスを同じXMEMインターフェースに接続することを可能にします。XMEMインターフェースの詳細タイミングについては211頁からの表137.～144.と図157.～160.を参照してください。

XMEMインターフェースは非同期で、次の図の波形が内部システム クロックに関連することに注意してください。内部と外部(XTAL1)のクロック間のズレは保証されません(デバイス、温度、供給電圧間で一様でない)。従ってXMEMインターフェースは同期動作用ではありません。

図13. 待ち状態なし 外部データメモリアクセス周期 (SRWn1=0, SRWn0=0) [T1~T3]



**共通注意事項**

SRWn1はSRW11(上位領域)またはSRW01(下位領域)、SRWn0はSRW10(上位領域)またはSRW00(下位領域)です。

[T1~Tn]は図での命令実行周期範囲です。

Xは値変更区間を示します。

最終周期のALEは次命令がRAM(内部または外部)をアクセスする場合だけ存在します。また、最後の周期のアドレスとデータの変更区間の有無も同様です。

図14. 1待ち状態 外部データメモリアクセス周期 (SRWn1=0, SRWn0=1) [T1~T4]

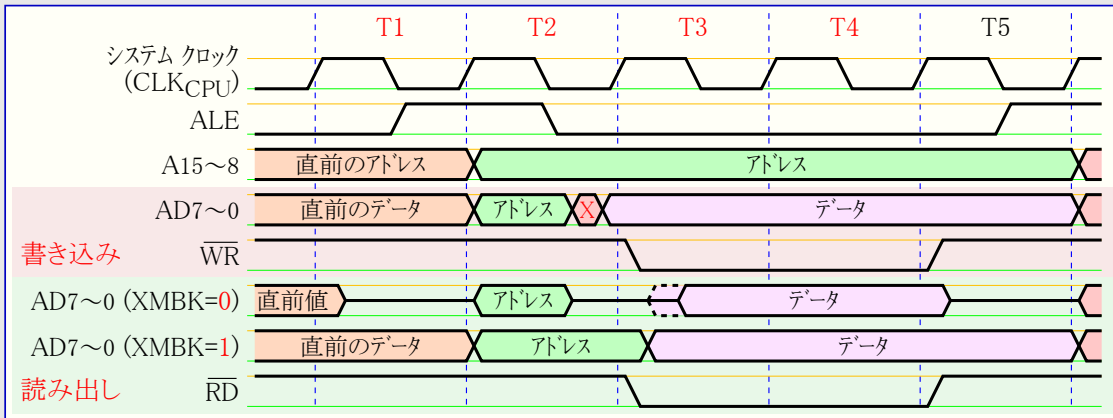


図15. 2待ち状態 外部データメモリアクセス周期 (SRWn1=1, SRWn0=0) [T1~T5]

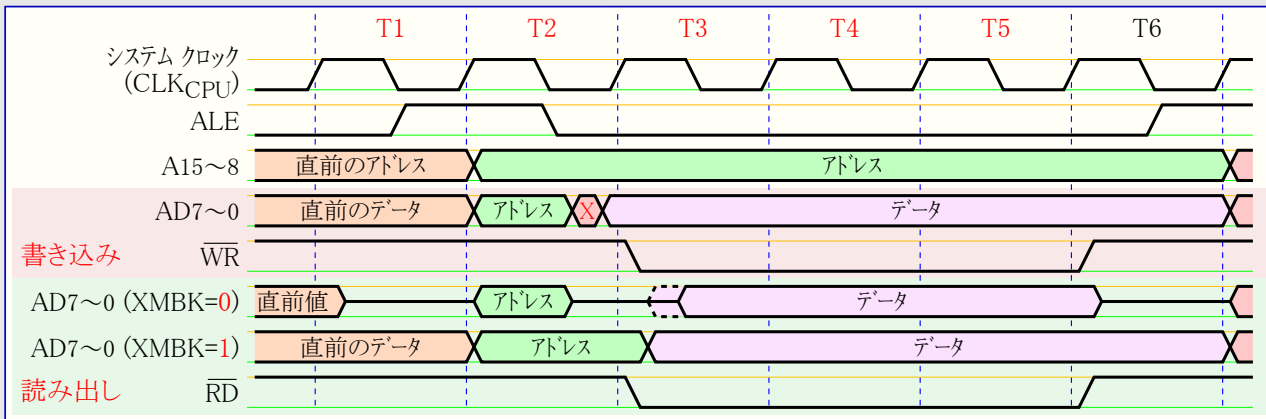
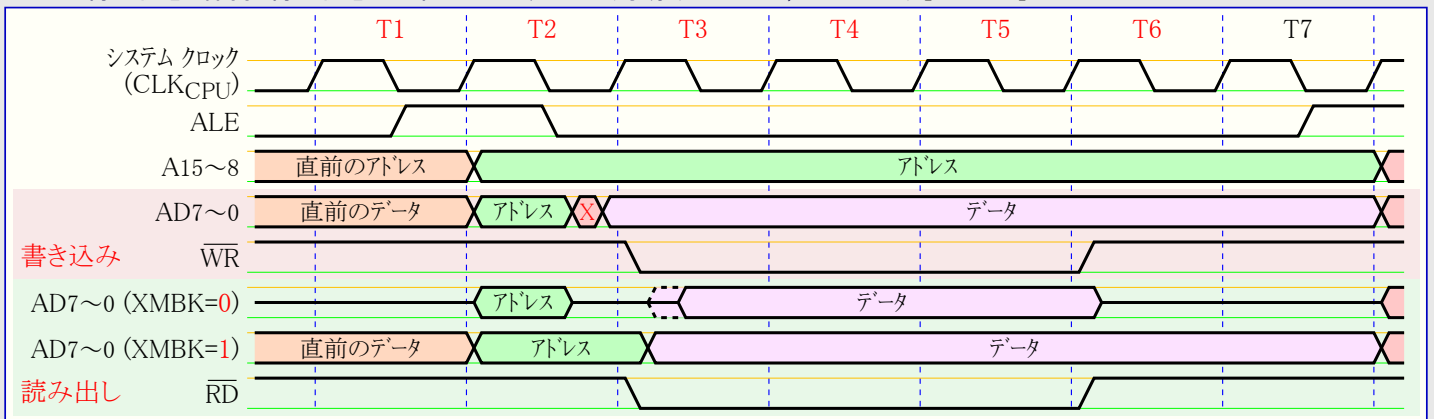


図16. 2待ち状態+保持1待ち状態 外部データメモリアクセス周期 (SRWn1=1, SRWn0=1) [T1~T6]



## XMEMインターフェース用レジスタ

### MCU制御レジスタ (MCU Control Register) MCUCR

| ビット         | 7   | 6     | 5   | 4   | 3   | 2   | 1     | 0    |       |
|-------------|-----|-------|-----|-----|-----|-----|-------|------|-------|
| \$35 (\$55) | SRE | SRW10 | SE  | SM1 | SM0 | SM2 | IVSEL | IVCE | MCUCR |
| Read/Write  | R/W | R/W   | R/W | R/W | R/W | R/W | R/W   | R/W  |       |
| 初期値         | 0   | 0     | 0   | 0   | 0   | 0   | 0     | 0    |       |

• **ビット7 – SRE : 外部メモリ許可 (External SRAM/XMEM Enable)**

SREに1を書くことが外部メモリ インターフェースを許可します。A15~8, AD7~0, ALE,  $\overline{RD}$ ,  $\overline{WR}$ ピン機能は交換ピン機能として活性(有効)にされます。SREビットはそれぞれのデータ方向レジスタ内の何れのピン方向設定も無効にします。SREに0を書くことは外部メモリ インターフェースを禁止し、通常ピンとデータ方向設定が使われます。

• **ビット6 – SRW10 : 待ち状態選択ビット (Wait-state Select Bit)**

非ATmega103互換動作(標準動作)での詳細な記述については以降(XMCRA内容)のSRWnビットの共通記述をご覧ください。ATmega103互換動作で、SRW10の1書き込みは待ち状態を許可し、図14.で示されるように読み/書きスロープ中に1つの延長周期が追加されます。

### 外部メモリ制御レジスタA (External Memory Control Register A) XMCRA

| ビット        | 7 | 6    | 5    | 4    | 3     | 2     | 1     | 0 |       |
|------------|---|------|------|------|-------|-------|-------|---|-------|
| (\$6D)     | - | SRL2 | SRL1 | SRL0 | SRW01 | SRW00 | SRW11 | - | XMCRA |
| Read/Write | R | R/W  | R/W  | R/W  | R/W   | R/W   | R/W   | R |       |
| 初期値        | 0 | 0    | 0    | 0    | 0     | 0     | 0     | 0 |       |

• **ビット7 – Res : 予約 (Reserved)**

このビットは予約されており、常に0として読みます。このアドレス位置へ書く時、将来のデバイスとの共通性のため、このビットに0を書いてください。

• **ビット6~4 – SRL2,SRL1,SRL0 : 外部SRAM範囲選択 (Wait-state Sector Limit)**

異なる外部メモリ領域に対して異なる待ち状態を指定することが可能です。外部メモリ アドレス空間は個別の待ち状態ビットを持つ2つの領域に分割できます。SRL2~0ビットはこれらの領域の分岐点を選びます。表3.と図11.をご覧ください。既定ではSRL2~0が0に設定され、外部メモリ アドレス空間全体が1つの領域として扱われます。SRAM(外部メモリ)アドレス空間全体が1つの領域として設定される場合、待ち状態はSRW11とSRW10ビットによって設定されます。

表3. 上位/下位領域選択

| SRL2 | SRL1 | SRL0 | 下位領域          | 上位領域          |
|------|------|------|---------------|---------------|
| 0    | 0    | 0    | なし            | \$1100~\$FFFF |
| 0    | 0    | 1    | \$1100~\$1FFF | \$2000~\$FFFF |
| 0    | 1    | 0    | \$1100~\$3FFF | \$4000~\$FFFF |
| 0    | 1    | 1    | \$1100~\$5FFF | \$6000~\$FFFF |
| 1    | 0    | 0    | \$1100~\$7FFF | \$8000~\$FFFF |
| 1    | 0    | 1    | \$1100~\$9FFF | \$A000~\$FFFF |
| 1    | 1    | 0    | \$1100~\$BFFF | \$C000~\$FFFF |
| 1    | 1    | 1    | \$1100~\$DFFF | \$E000~\$FFFF |

• **ビット1,MCUCRのビット6 – SRW11,SRW10 : 上位領域待ち状態選択ビット1,0 (Wait-state Select Bits for Upper Sector)**

SRW11とSRW10ビットは外部メモリ アドレス空間の上位領域に対する待ち状態数を制御します。表4.をご覧ください。

• **ビット3,2 – SRW01,SRW00 : 下位領域待ち状態選択ビット1,0 (Wait-state Select Bits for Lower Sector)**

SRW01とSRW00ビットは外部メモリ アドレス空間の下位領域に対する待ち状態数を制御します。表4.をご覧ください。

表4. 待ち状態選択

| SRWn1 | SRWn0 | 待ち状態                               |
|-------|-------|------------------------------------|
| 0     | 0     | 待ち状態なし                             |
| 0     | 1     | 読み/書きスロープ中に1待ち状態挿入                 |
| 1     | 0     | 読み/書きスロープ中に2待ち状態挿入                 |
| 1     | 1     | 読み/書きスロープ中に2、アドレス、データ出力保持中に1待ち状態挿入 |

注: nは0(下位領域)または1(上位領域)です。外部メモリ インターフェースの待ち状態とタイミングのより多くの詳細については図13.~16.のSRWビット設定がタイミングへ与える影響をご覧ください。

• **ビット0 – Res : 予約 (Reserved)**

このビットは予約されており、常に0として読みます。このアドレス位置へ書く時、将来のデバイスとの共通性のため、このビットに0を書いてください。



## 外部メモリ制御レジスタB (External Memory Control Register B) XMCRB

| ビット (\$6C) | 7    | 6 | 5 | 4 | 3 | 2    | 1    | 0    |       |
|------------|------|---|---|---|---|------|------|------|-------|
|            | XMBK | - | - | - | - | XMM2 | XMM1 | XMM0 | XMCRB |
| Read/Write | R/W  | R | R | R | R | R/W  | R/W  | R/W  |       |
| 初期値        | 0    | 0 | 0 | 0 | 0 | 0    | 0    | 0    |       |

### • ビット7 - XMBK : バス保持許可 (External Memory Bus-keeper Enable)

XMBKに1を書くことがAD7~0線のバス保持機能を許可します。バス保持機能が許可されると、その他の状態であるHi-Zにされる時にAD7~0バスでの論理レベル(0または1)固定を保証します。XMBKに0を書くことがバス保持機能を禁止します。XMBKはSREの制限を受けませんので、XMEMインターフェースが禁止されても、バス保持機能はXMBKが1である限り未だ活性(有効)にされます。

### • ビット6~3 - Res : 予約 (Reserved)

これらのビットは予約されており、常に0として読みます。このアドレス位置へ書く時、将来のデバイスとの共通性のため、これらのビットに0を書いてください。

### • ビット2~0 - XMM2,XMM1,XMM0 : 上位アドレス遮蔽 (External Memory High Mask)

外部メモリインターフェースが許可されると、既定では全てのポートピンが上位アドレスバイトに使われます。外部メモリアクセスのために全60Kバイトアドレス空間が必要とされない場合、表5.で記述されるように、いくつかまたは全てのポートピンは標準ポートピン機能用に開放できます。23頁の「外部メモリの全64Kバイト位置の使用」で記述されるように、外部メモリの全64Kバイト位置のアクセスのためにXMMnビットの使用が可能です。

表5. 外部メモリ許可時に開放するポートピン

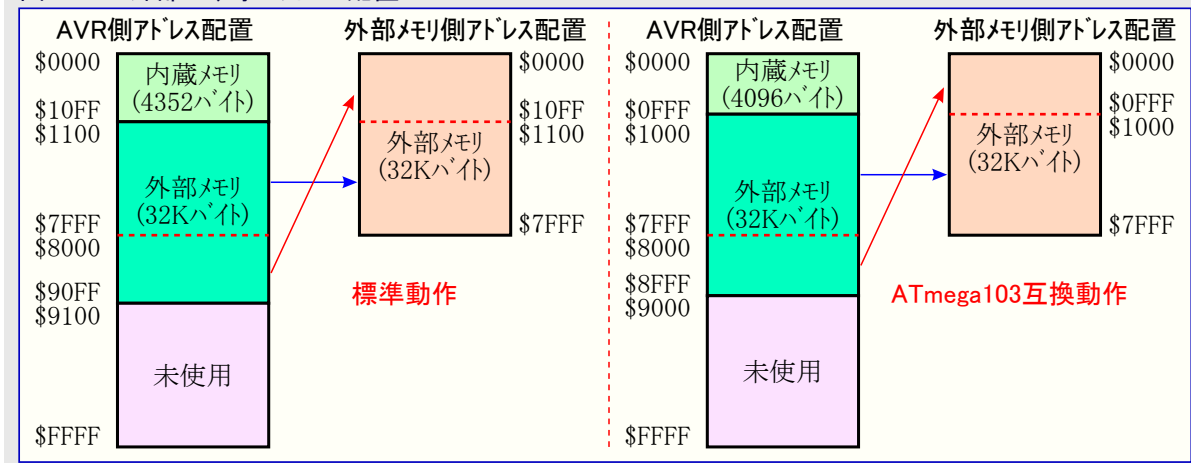
| XMM2 | XMM1 | XMM0 | アドレス上位バイトビット数 | 開放ポートピン  |
|------|------|------|---------------|----------|
| 0    | 0    | 0    | 8 (最大約60Kバイト) | なし       |
| 0    | 0    | 1    | 7             | PORTC7   |
| 0    | 1    | 0    | 6             | PORTC7~6 |
| 0    | 1    | 1    | 5             | PORTC7~5 |
| 1    | 0    | 0    | 4             | PORTC7~4 |
| 1    | 0    | 1    | 3             | PORTC7~3 |
| 1    | 1    | 0    | 2             | PORTC7~2 |
| 1    | 1    | 1    | 0 (上位バイトなし)   | PORTC7~0 |

## 64Kバイト未満外部メモリの全域使用

外部メモリは図11.で示されるように内部メモリの後に配置されるため、データ空間の最初の4352バイトをアドレス指定するとき、外部メモリは指定されません。外部メモリの最初の4352バイト(アドレス\$0000~\$10FFF)がアクセスできないと思われるかもしれませんが、けれども64Kバイトより小さな外部メモリを接続するとき、例えば32Kバイトで、これらの位置は単純なアドレス\$8000~\$90FFF指定によって容易にアクセスされます。外部メモリアドレスA15ビットは外部メモリに接続されず、アドレス\$8000~\$90FFFは外部メモリに対するアドレス\$0000~\$10FFFとして見えます。\$90FFFを越えるアドレス指定はこれが他の(下位側)アドレスによって既にアクセスされる外部メモリ位置のアドレス指定のため推奨されません。応用ソフトウェアにとって、この外部32Kバイトメモリは\$1100~\$90FFFの1つの直線的な32Kバイトアドレス空間として見えます。これは図17.で図解されます。(訳注:標準/互換動作直接記述のため、原書の本位置の1行を削除)

デバイスがATmega103互換動作に設定されると、内部アドレス空間は4096バイトです。これは外部メモリの先頭4096バイトがアドレス\$8000~\$8FFFでアクセスできることを意味します。応用ソフトウェアには\$1000~\$8FFFの1つの直線的な32Kバイトアドレス空間として見えます。

図17. 32K外部メモリ時のアドレス配置



## 外部メモリの全64Kバイト位置の使用

外部メモリは図11.で示されるように内部メモリの後に配置されるため、既定では外部メモリの約60Kバイトだけが利用可能です(アドレス空間\$0000～\$10FFFは内部メモリ用に予約されます)。けれども上位アドレスビットを0で遮蔽することにより、外部メモリ全体を利用することが可能です。これはXMMnビットを使い、アドレスの最上位側ビットをソフトウェアによって制御することで行えます。ポートCを\$00出力に設定し、上位側ビットを標準ポートピン動作作用に開放することにより、メモリインターフェースは\$0000～\$1FFFをアドレス指定します。以下のコード例をご覧ください。

## アセンブリ言語プログラム例

```

.EQU    OFFSET=$2000                ;切り替え点アドレス定義
                                           ;[ポートC初期化]
LDI     R16, $00                    ;ポート出力値を取得
OUT     PORTC, R16                  ;アドレス最上位側ビット値全0設定
LDI     R16, $FF                    ;ポート方向全出力値を取得
OUT     DDRC, R16                   ;アドレス上位バイト用ポート出力設定
;                                     ;[$0000～$1FFFアクセス]
LDI     R16, (1<<XMM1) | (1<<XMM0)  ;PORTC7～5開放値を取得
STS     XMCRB, R16                  ;PORTC7～5開放(~$1FFF範囲設定)
LDI     R16, $AA                    ;書き込み値を取得
STS     $0001+OFFSET, R16           ;外部メモリの$0001番地に$AA書き込み
;                                     ;[$2000～$FFFFアクセス]
LDI     R16, $00                    ;最大外部メモリ(16ビット幅)値を取得
STS     XMCRB, R16                  ;PORTC開放なし、(~$FFFF範囲設定)
LDI     R16, $55                    ;書き込み値を取得
STS     $0001+OFFSET, R16           ;外部メモリのOFFSET+$0001番地に$55書き込み

```

## C言語プログラム例

```

#define OFFSET 0x2000                /* 切り替え点アドレス定義 */
void XRAM_example(void)
{
    unsigned char *p = (unsigned char *) (OFFSET + 1); /* 書き込みアドレス(ポインタ)定義 */
    PORTC = 0x00; /* アドレス最上位側ビット値全0設定 */
    DDRC = 0xFF; /* アドレス上位バイト用ポート出力設定 */
    XMCRB = (1<<XMM1) | (1<<XMM0); /* PORTC7～5開放(~$1FFF範囲設定) */
    *p = 0xAA; /* 外部メモリの$0001番地に$AA書き込み */
    XMCRB = 0x00; /* PORTC開放なし、(~$FFFF範囲設定) */
    *p = 0x55; /* 外部メモリの$2001番地に$55書き込み */
}

```

注: 6ページの「コード例について」をご覧ください。

メモリの多くが遮蔽され隠されているとき、この選択手段の使用は注意して用いられなければなりません。

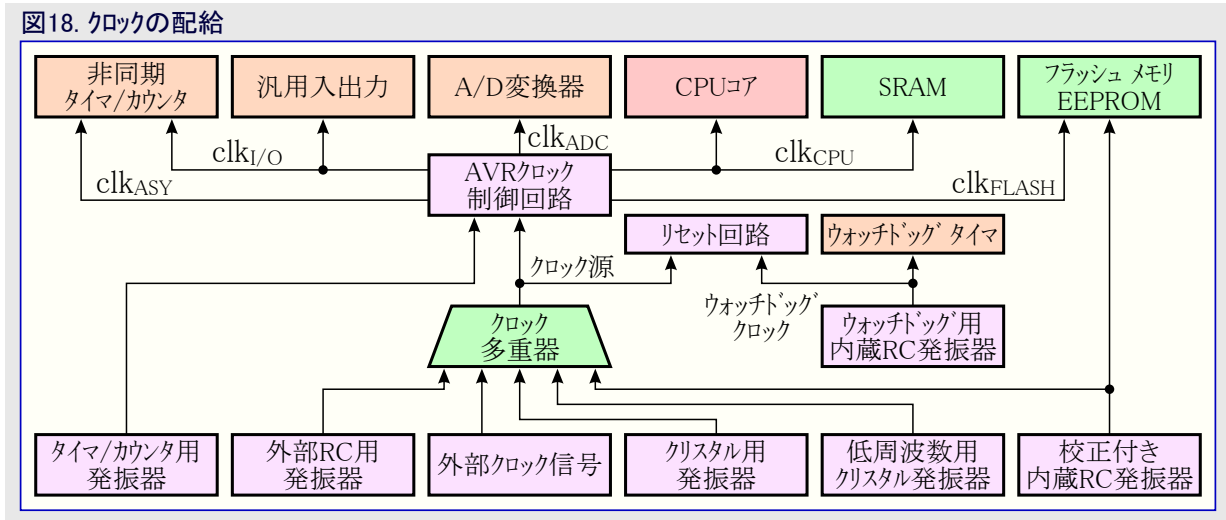
(訳補) 上の記述はXMMn設定によりバンク化動作となることを示しています。



## システム クロックとクロック選択

### クロック系統とその配給

図18はAVR内の主要なクロック系統とその配給を示します。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、29頁の「電力管理と休止形態」で記述される各種休止形態の使用によって、使われない部分のクロックを停止することができます。クロック系統は以下で詳述されます。



#### 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の部分に配給されます。このような部分の例は汎用レジスタファイル、ステータスレジスタ、スタックポインタを保持するデータメモリです。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。

#### I/O クロック - $clk_{I/O}$

I/Oクロックはタイマ/カウンタ、SPI、USART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。2線直列I/F(TWI)部のアドレス認証は $clk_{I/O}$ が停止される時非同期に実行され、全休止形態でTWIアドレス受信を可能とすることにも注意してください。

#### フラッシュ クロック - $clk_{FLASH}$

フラッシュクロックはフラッシュメモリインターフェースの動作を制御します。このフラッシュクロックは常にCPUクロックと同時に活動します。

#### 非同期タイマ クロック - $clk_{ASY}$

非同期タイマクロックは外部32kHzクロック用クリスタルから直接的にクロック駆動されるのを非同期タイマ/カウンタに許します。この専用クロック範囲はデバイスが休止形態の時でも、このタイマ/カウンタの実時間計数器としての使用を許します。

#### 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た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。

### クロック元

このデバイスには右で示されるようにフラッシュヒューズビットによって選択可能な後続のクロック元選択があります。選んだクロック元からのクロックはAVRクロック発生器への入力で、適切な単位部へ配給されます。

各クロック選択に対する各種選択は次項で得られます。CPUがパワーダウンまたはパワーセーブから起動するとき、選んだクロック元は命令実行開始前に安定な発振器動作を保証する起動時間に使われます。CPUがリセットから始まる時、これらは通常動作開始前に安定電圧へ達するのを電源に許す付加遅延です。ウォッチドッグ発振器はこの起動時間の実時間部のタイミングに使われます。各計時完了に使われるウォッチドッグ(WDT)発振器の周期数は表7で示されます。214頁の「代表特性」で示されるようにウォッチドッグ発振器の周波数は電圧に依存します。

表6. クロック種別選択

| クロック種別           | CKSEL3~0  |
|------------------|-----------|
| 外部クリスタル/セラミック発振器 | 1111~1010 |
| 外部低周波数クリスタル発振器   | 1001      |
| 外部RC発振           | 1000~0101 |
| 校正付き内蔵RC発振器      | 0100~0001 |
| 外部クロック信号         | 0000      |

注: 1=非プログラム、0=プログラム

表7. WDT発振器の代表的計時完了値、周期数

| VCC=3.0V | VCC=5.0V | 周期数         |
|----------|----------|-------------|
| 4.3ms    | 4.1ms    | 4K (4096)   |
| 69ms     | 65ms     | 64K (65536) |

## 既定のクロック元

このデバイスはCKSEL=0001, SUT=10で出荷されます。従って既定クロック元設定は最長起動時間の1MHz内蔵RC発振器です。この既定設定は全ての使用者が実装または並列書き込み器を使って、それらを希望したクロック元設定にできることを保証します。

## クリスタル用発振器

XTAL1とXTAL2は図19.で示されるように、チップ上の発振器としての使用に設定できる反転増幅器の各々、入力と出力です。クリスタル発振器またはセラミック振動子のどちらでも使えます。CKOPTヒューズは2つの異なる発振増幅器動作の1つを選びます。CKOPTヒューズがプログラム(0)されると、発振器出力は完全な供給電圧端振幅出力で発振します。この動作は雑音が多い環境やXTAL2出力が他のクロック緩衝器を駆動する時に適します。この動作は広い周波数範囲を持ちます。CKOPTヒューズが非プログラム(1)にされると、発振器はより小さな出力振幅になります。これはかなり消費電力を削減します。この動作は制限された周波数範囲を持ち、他のクロック緩衝器を駆動するのに使えません。

セラミック振動子での最高周波数はCKOPTヒューズが非プログラム(1)で8MHz、CKOPTヒューズがプログラム(0)で16MHzです。C1とC2はクリスタル発振器とセラミック振動子の両方について常に等しくすべきです。このコンデンサの最適値は使うクリスタル発振器やセラミック振動子、浮遊容量の量、その環境の電磁雑音に依存します。クリスタル発振器使用に対するコンデンサ選択について初期の指針のいくつかは表8.で与えられます。セラミック振動子については製造業者によって与えられたコンデンサ値が使われるべきです。

この発振器は示された周波数範囲で各々最適化された3つの異なる種別で動作できます。この動作種別は表8.で示されるようにCKSEL3~1ヒューズによって選ばれます。

CKSEL0ヒューズはSUT1,0ヒューズと共に表9.で示されるように起動時間を選びます。

図19. クリスタル発振器接続図

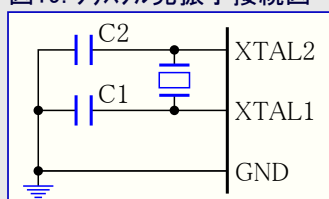


表8. クリスタル発振器動作

| CKOPT | CKSEL3~1 | 周波数範囲      | 推奨C1,2容量 |
|-------|----------|------------|----------|
| 1     | 101 (注1) | 0.4~0.9MHz | -        |
|       | 110      | 0.9~3.0MHz | 12~22pF  |
|       | 111      | 3.0~8.0MHz | 12~22pF  |
| 0     | 101~111  | 1.0~ MHz   | 12~22pF  |

注1: この選択はクリスタル発振器ではなく、セラミック振動子でのみ使われるべきです。

表9. クリスタル発振器/セラミック振動子用起動遅延時間選択表

| CKSEL0 | SUT1,0 | パワーダウン、パワーセーブからの起動遅延時間 | リセットからの付加遅延時間 (VCC=5.0V) | 推奨使用法                       |
|--------|--------|------------------------|--------------------------|-----------------------------|
| 0      | 0 0    | 258 × CK (注1)          | 4.1ms                    | 外部セラミック振動子、高速上昇電源           |
|        | 0 1    | 258 × CK (注1)          | 65ms                     | 外部セラミック振動子、低速上昇電源           |
|        | 1 0    | 1K × CK (注2)           | -                        | 外部セラミック振動子、低電圧検出(BOD)リセット許可 |
|        | 1 1    | 1K × CK (注2)           | 4.1ms                    | 外部セラミック振動子、高速上昇電源           |
| 1      | 0 0    | 1K × CK (注2)           | 65ms                     | 外部セラミック振動子、低速上昇電源           |
|        | 0 1    | 16K × CK               | -                        | 外部クリスタル発振器、低電圧検出(BOD)リセット許可 |
|        | 1 0    | 16K × CK               | 4.1ms                    | 外部クリスタル発振器、高速上昇電源           |
|        | 1 1    | 16K × CK               | 65ms                     | 外部クリスタル発振器、低速上昇電源           |

注1: これらの選択はデバイスの最高周波数付近での動作でないとき、応用にとって起動での周波数安定性が重要でない場合だけ使われるべきです。これらの選択はクリスタル発振器用ではありません。

注2: これらの選択はセラミック振動子での使用を意図され、起動での周波数安定性を保証します。デバイスの最高周波数付近での動作でないとき、応用にとって起動での周波数安定性が重要でない場合はクリスタル発振器も使えます。

## 低周波数クリスタル用発振器

デバイスに対するクロック元として時計用32.768kHzクリスタルを使うには、CKSEL3~0ヒューズを'1001'に設定することによって低周波数クリスタル発振器が選ばれなければなりません。クリスタルは図19.で示されるように接続されるべきです。CKOPTヒューズのプログラム(0)により、使用者はXTAL1とXTAL2の内部容量(コンデンサ)を許可でき、それによって外部コンデンサの必要がなくなります。内部容量は36pFの公称値です。

この発振器が選ばれると、起動時間は表10.で示されるようにSUTヒューズによって決定されます。

表10. 低周波数クリスタル発振器用起動遅延時間選択表

| SUT1,0 | パワーダウン、パワーセーブからの起動遅延時間 | リセットからの付加遅延時間 (VCC=5.0V) | 推奨使用法                     |
|--------|------------------------|--------------------------|---------------------------|
| 0 0    | 1K × CK (注1)           | 4.1ms                    | 高速上昇電源または低電圧検出(BOD)リセット許可 |
| 0 1    | 1K × CK (注1)           | 65ms                     | 低速上昇電源                    |
| 1 0    | 32K × CK               | 65ms                     | 起動時周波数の安定重視               |
| 1 1    |                        |                          | (予約)                      |

注1: これらの選択は応用にとって起動での周波数安定性が重要でない場合だけ使われるべきです。

## 外部RC発振器

タイミングに鈍感な応用に対しては図20.で示される外部RC設定が使えます。周波数は式 $f=1/(3RC)$ によって大まかに推測されます。Cは最低22pFであるべきです。CKOPTヒューズのプログラム(0)により、使用者はXTAL1とGND間の36pF内部容量を許可でき、それによって外部コンデンサの必要がなくなります。RとCの選択法の詳細と発振器操作のより多くの情報については「外部RC発振器(External RC Oscillator)」応用記述を参照してください(訳注:本応用記述は存在しません)。

この発振器は示された周波数範囲で各々最適化された4つの異なる種別で動作できます。この動作は表11.で示されるようにCKSEL3~0ヒューズによって選ばれます。

この発振器が選ばれると、起動時間は表12.で示されるようにSUTヒューズによって決定されます。

図20. 外部RC接続図

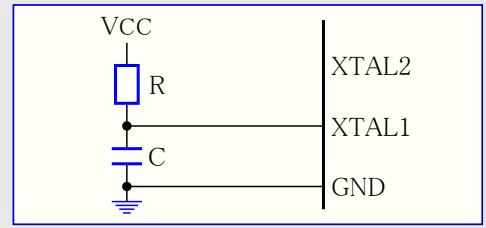


表11. 外部RC発振器動作

| CKSEL3~0 | 周波数範囲 (MHz) |
|----------|-------------|
| 0 1 0 1  | 0.1~0.9     |
| 0 1 1 0  | 0.9~3.0     |
| 0 1 1 1  | 3.0~8.0     |
| 1 0 0 0  | 8.0~12.0    |

表12. 外部RC発振器用起動遅延時間選択表

| SUT1.0 | パワーダウン、パワーセーブからの起動遅延時間 | リセットからの付加遅延時間 (VCC=5.0V) | 推奨使用法                     |
|--------|------------------------|--------------------------|---------------------------|
| 0 0    | 18×CK                  | -                        | 低電圧検出(BOD)リセット許可          |
| 0 1    | 18×CK                  | 4.1ms                    | 高速上昇電源                    |
| 1 0    | 18×CK                  | 65ms                     | 低速上昇電源                    |
| 1 1    | 6×CK (注1)              | 4.1ms                    | 高速上昇電源または低電圧検出(BOD)リセット許可 |

注1: この選択はデバイスの最高周波数付近で動作する時に使われるべきではありません。

## 校正付き内蔵RC発振器

校正された内蔵RC発振器は決められた1.0, 2.0, 4.0, 8.0MHzクロックを供給します。全ての周波数は5V, 25°Cでの公称値です。このクロックは表13.で示されるようにCKSELヒューズのプログラミングによってシステムクロックとして選べます。選んだなら、外部部品なしで動作します。このクロック選択を使うとき、CKOPTヒューズは常に非プログラム(1)にされるべきです。リセット中、ハードウェアが発振校正レジスタ(OSCCAL)に1MHz校正値レジスタを設定し、これによってRC発振器を自動的に校正します。5V, 25°Cで1.0MHz発振器周波数が選ばれ、この校正は公称周波数±3%以内の周波数を与えます。atmel.com/avrで利用可能な応用記述に記載された実行時校正法の使用で、与えられたどのVCCと温度でも±1%の精度を達成することができます。この発振器がチップ(システム)クロックとして使われるとき、ウォッチドッグ発振器は未だウォッチドッグタイマとリセット付加遅延タイマに使われます。予め設定された校正值のより多くの情報については187頁の「校正バイト」項をご覧ください。

この発振器が選ばれると、起動時間は表14.で示されるようにSUTヒューズによって決定されます。XTAL1とXTAL2は未接続(NC)のままにされるべきです。

表13. 校正付き内蔵RC発振器動作

| CKSEL3~0     | 公称周波数 (MHz) |
|--------------|-------------|
| 0 0 0 1 (注1) | 1.0         |
| 0 0 1 0      | 2.0         |
| 0 0 1 1      | 4.0         |
| 0 1 0 0      | 8.0         |

注1: デバイスはこの選択で出荷されます。

表14. 校正付き内蔵RC発振器用起動遅延時間選択表

| SUT1.0   | パワーダウン、パワーセーブからの起動遅延時間 | リセットからの付加遅延時間 (VCC=5.0V) | 推奨使用法            |
|----------|------------------------|--------------------------|------------------|
| 0 0      | 6×CK                   | -                        | 低電圧検出リセット(BOD)許可 |
| 0 1      | 6×CK                   | 4.1ms                    | 高速上昇電源           |
| 1 0 (注1) | 6×CK                   | 65ms                     | 低速上昇電源           |
| 1 1      |                        |                          | (予約)             |

注1: デバイスはこの選択で出荷されます。

## 発振校正レジスタ (Oscillator Calibration Register) OSCCAL

|            |            |      |      |      |      |      |      |      |        |
|------------|------------|------|------|------|------|------|------|------|--------|
| ビット (\$6F) | 7          | 6    | 5    | 4    | 3    | 2    | 1    | 0    | OSCCAL |
|            | CAL7       | CAL6 | CAL5 | CAL4 | CAL3 | CAL2 | CAL1 | CAL0 |        |
| Read/Write | R/W        | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  |        |
| 初期値        | デバイス固有の校正値 |      |      |      |      |      |      |      |        |

注: OSCCALレジスタはATmega103互換動作で利用できません。

### •ビット7~0 - CAL7~0: 発振校正値 (Oscillator Calibration Value)

このアドレスへの校正ビット書き込みは発振器周波数の偏差処理を省くために内蔵発振器を調整します。リセット中、識別列上位バイト(アドレス\$00)に配置される1MHz校正値が発振校正レジスタ(OSCCAL)内へ自動的に設定されます。内蔵RC発振器が他の周波数で使われる場合、校正値は手動で設定されなければなりません。これは初めに書き込み器によって識別列を読み、そしてその後フラッシュメモリまたはEEPROMに校正値を保存することによって行えます。その後この値はソフトウェアによって読まれ、OSCCALに設定できます。OSCCALが0の時に最低利用可能周波数が選ばれます。このレジスタへ0以外の値を書くことは内蔵発振器の周波数を増加します。このレジスタへの\$FF書き込みは最高使用可能周波数にします。校正付き発振器はフラッシュメモリとEEPROMのアクセス時間に使われます。フラッシュメモリまたはEEPROMが書かれる場合、公称周波数より上へ10%を越えて校正してはいけません。そうでなければ、フラッシュメモリまたはEEPROM書き込みは失敗するかもしれません。この発振器は1.0, 2.0, 4.0, 8.0MHzへの校正が意図されることに注意してください。表15.で示されるような他の値への調整は保証されません。

表15. 内蔵RC発振器周波数範囲

| OSCCAL値 | 公称周波数に対する割合 |       |
|---------|-------------|-------|
|         | 最小          | 最大    |
| \$00    | 50 %        | 100 % |
| \$7F    | 75 %        | 150 % |
| \$FF    | 100 %       | 200 % |

## 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ためにXTAL1は図21.で示されるように駆動されるべきです。外部クロックでデバイスを走行するためにCKSELピュースは'0000'にプログラム(設定)されなければなりません。CKOPTピュースのプログラム(0)により、使用者はXTAL1とGND間の36pF内部容量を許可できます。

このクロック元が選ばれると、起動時間は表16.で示されるようにSUTピュースによって決定されます。

図21. 外部クロック信号駆動接続図

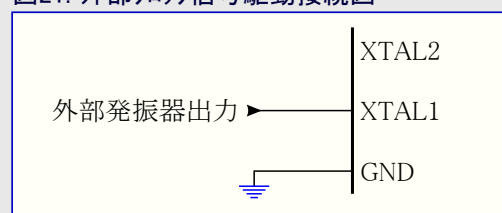


表16. 外部クロック信号駆動用起動遅延時間選択表

| SUT1,0 | パワーダウン、パワーセーブからの起動遅延時間 | リセットからの付加遅延時間 (VCC=5.0V) | 推奨使用法            |
|--------|------------------------|--------------------------|------------------|
| 0 0    | 6 × CK                 | -                        | 低電圧検出リセット(BOD)許可 |
| 0 1    | 6 × CK                 | 4.1ms                    | 高速上昇電源           |
| 1 0    | 6 × CK                 | 65ms                     | 低速上昇電源           |
| 1 1    | (予約)                   |                          |                  |

外部クロックを供給するとき、MCUの安定な動作を保証するために供給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。このようなクロック周波数での変化中、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。

## タイマ/カウンタ用発振器

AVR マイクロコントローラのタイマ/カウンタ用発振器(TOSC1とTOSC2)ピンに対して、クリスタル発振器はこのピン間に直接的に接続されます。外部コンデンサは必要とされません。この発振器は時計用32.768kHzクリスタルでの使用に最適化されています。外部クロック元をTOSC1に供給することは推奨されません。

注: タイマ/カウンタ用発振器は低周波数クリスタル用発振器と同じ形式のクリスタル用発振器を使い、内部容量(C)は同じ36pFの公称値を持ちます。



## クロック分周制御レジスタ (XTAL Divide Control Register) XDIV

クロック分周制御レジスタは元となるクロック周波数を2～129範囲の数で分周するために使われます。この機能は処理能力に対する必要条件が低いとき、消費電力低減のために使えます。

| ビット         | 7      | 6     | 5     | 4     | 3     | 2     | 1     | 0     |      |
|-------------|--------|-------|-------|-------|-------|-------|-------|-------|------|
| \$3C (\$5C) | XDIVEN | XDIV6 | XDIV5 | XDIV4 | XDIV3 | XDIV2 | XDIV1 | XDIV0 | XDIV |
| Read/Write  | R/W    | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   |      |
| 初期値         | 0      | 0     | 0     | 0     | 0     | 0     | 0     | 0     |      |

### • ビット7 - XDIVEN : クロック分周許可 (XTAL Divide Enable)

XDIVENビットが1を書かれると、CPUと全周辺機能のクロック( $clk_{CPU}$ ,  $clk_{I/O}$ ,  $clk_{FLASH}$ ,  $clk_{ADC}$ )周波数は、XDIV6～0の設定で定義した要素で分周されます。このビットは応用目的に合せたクロック周波数へ変えるため、実行時に書くことができます。

### • ビット6～0 - XDIV6～0 : クロック分周値選択 (XTAL Divide Select Bits 6～0)

これらのビットはクロック分周許可(XDIVEN)ビットが設定(1)された場合に適用する分周要素を定義します。これらのビット値がdで示される場合、次式はCPUと全周辺機能のクロック周波数 $f_{CLK}$ の結果を定義します。

$$f_{CLK} = \frac{\text{元となるクロック}}{129-d}$$

これらのビット値はクロック分周許可(XDIVEN)ビットが解除(0)の時だけ変更できます。XDIVENが1を書かれるとき、同時にXDIV6～0内に書かれた値は分周要素として採用されます。XDIVENが0を書かれるとき、同時にXDIV6～0内に書かれた値は却下されます。この分周器がMCUへの主クロック入力を分周するため、分周要素が使われると、全周辺(機能)の速度が低下されます。

システムクロックが分周されると、タイマ/カウンタ0は非同期クロックでだけ使えます。非同期クロック周波数は(主クロック用)元クロックを分周した周波数の1/4よりも低くなければなりません。さもなければ割り込みが失われたり、タイマ/カウンタ0のレジスタアクセスが失敗するかもしれません。



## 電力管理と休止形態

休止形態は応用で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を提供します。

6つの休止形態の何れかへ移行するにはMCU制御レジスタ(MCUCR)の**休止許可(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。MCUCRの**休止種別選択(SM2~0)ビット**は**SLEEP**命令によって活性(有効)にされる休止形態(アイドル、A/D変換雑音低減、パワーダウン、パワーセーブ、スタンバイ、拡張スタンバイ)のどれかを選びます。一覧については表17をご覧ください。MCUが休止形態中に許可した割り込みが起こると、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。休止形態中にリセットが起こると、MCUは起動し、リセットベクタから実行します。

24頁の図18はATmega128の各種クロック系統とその配給を示します。この図は適切な休止形態を選ぶ助けになります。

### MCU制御レジスタ (MCU Control Register) MCUCR

このMCU制御レジスタは電力管理用の制御ビットを含みます。

| ビット         | 7          | 6            | 5         | 4          | 3          | 2          | 1            | 0           |       |
|-------------|------------|--------------|-----------|------------|------------|------------|--------------|-------------|-------|
| \$35 (\$55) | <b>SRE</b> | <b>SRW10</b> | <b>SE</b> | <b>SM1</b> | <b>SM0</b> | <b>SM2</b> | <b>IVSEL</b> | <b>IVCE</b> | MCUCR |
| Read/Write  | R/W        | R/W          | R/W       | R/W        | R/W        | R/W        | R/W          | R/W         |       |
| 初期値         | 0          | 0            | 0         | 0          | 0          | 0          | 0            | 0           |       |

#### • ビット5 - SE : 休止許可 (Sleep Enable)

**SLEEP**命令が実行される時にMCUを休止形態へ移行させるには、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。

#### • ビット2,4,3 - SM2,SM1,SM0 : 休止種別選択 (Sleep Mode Select Bit 2, 1 and 0)

これらのビットは表17で示される利用可能な6つの休止形態の1つを選びます。

表17. 休止形態種別選択

| SM2 | SM1 | SM0 | 休止形態種別        |
|-----|-----|-----|---------------|
| 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   | パワーセーブ動作      |
| 1   | 0   | 0   | (予約)          |
| 1   | 0   | 1   | (予約)          |
| 1   | 1   | 0   | スタンバイ動作 (注)   |
| 1   | 1   | 1   | 拡張スタンバイ動作 (注) |

注: (拡張)スタンバイ動作は外部クリスタル発振子またはセラミック振動子でだけ利用できます。

### アイドル動作

休止種別選択(SM2~0)ビットが'000'を書かれるとき、**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、SPI、USART、アナログ比較器、A/D変換器、2線直列インターフェース、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clkFLASH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にタイマ溢れやUSARTの送信完了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(ACSR)のアナログ比較器禁止(ACD)ビット**を設定(1)することによって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。

### A/D変換雑音低減動作

SM2~0ビットが'001'を書かれるとき、**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、2線直列インターフェースのアドレス監視、タイマ/カウンタ0、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、clkFLASH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、ウォッチドッグリセット、低電圧検出(BOD)リセット、2線直列インターフェースのアドレス一致割り込み、タイマ/カウンタ0の割り込み、SPM/EEPROM操作可割り込み、INT7~4の外部レベル割り込みとINT3~0の外部割り込みだけが、A/D変換雑音低減動作からMCUを起動できます。

## パワーダウン動作

休止種別選択(SM2~0)ビットが'010'を書かれると、SLEEP命令はMCUをパワーダウン動作へ移行させます。この動作では外部発振器が停止される一方、外部割り込み、2線直列インターフェースのアドレス監視、ウォッチドッグ機能は(許可されていれば)継続して動作します。外部リセット、ウォッチドッグリセット、低電圧検出(BOD)リセット、2線直列インターフェースのアドレス一致割り込み、INT7~4の外部レベル割り込み、INT3~0の外部割り込みだけがMCUを起動できます。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。

レベルで起動した割り込みがパワーダウン動作からの復帰に使われる場合、MCUを起動するため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59ページの「外部割り込み」を参照してください。

パワーダウン動作から起動するとき、起動条件が起きてから起動の効果が現れるまで遅延があります。これは停止されてしまっている後の再始動と安定になることをクロックに許します。この起動(遅延)時間は24ページの「クロック元」で記述されるように、リセット遅延時間を定義するのと同じCKSELヒューズによって定義されます。

## パワーセーブ動作

SM2~0ビットが'011'を書かれると、SLEEP命令はMCUをパワーセーブ動作へ移行させます。この動作は(次の)1つの例外を除いてパワーダウン動作と同じです。

タイマ/カウンタ0が非同期にクロック駆動される、換言するとタイマ/カウンタ0非同期状態レジスタ(ASSR)の非同期クロック(AS0)ビットが設定(1)されると、タイマ/カウンタ0は休止中に走行(動作)します。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、タイマ/カウンタ割り込み許可レジスタ(TIMSK)のタイマ/カウンタ0溢れ割り込み許可(TOIE0)ビットまたは比較0割り込み許可(OCIE0)ビットが設定(1)されるなら、デバイスは対応するどちらの割り込みからでも起動できます。

タイマ/カウンタ0が非同期にクロック駆動されないなら、パワーセーブ動作での起動復帰後のタイマ/カウンタ0のレジスタ内容はASSRのAS0=0でも不定と見做すべきなので、パワーダウン動作をパワーセーブ動作の代わりにすることが推奨されます。

この休止形態は基本的にclkASY以外の全てのクロックを停止し、非同期にクロック駆動されるならタイマ/カウンタ0を含め、非同期部の動作だけを許します。

## スタンバイ動作

外部クリスタル発振子/セラミック振動子クロック種別が選ばれ、SM2~0ビットが'110'のとき、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(外部クリスタル用)発振器が走行(動作)を保たれる例外を除いてパワーダウン動作と同じです。デバイスはスタンバイ動作から6クロック周期で起動します。

## 拡張スタンバイ動作

外部クリスタル発振子/セラミック振動子クロック種別が選ばれ、SM2~0ビットが'111'のとき、SLEEP命令はMCUを拡張スタンバイ動作へ移行させます。この動作は(外部クリスタル用)発振器が走行(動作)を保たれる例外を除いてパワーセーブ動作と同じです。デバイスは拡張スタンバイ動作から6クロック周期で起動します。

表18. 各休止形態に於ける動作クロック範囲と復帰起動要因

| 休止種別        | 動作クロック範囲 |           |        |         |         | 動作発振器    |         | 復帰起動要因 (割り込み) |            |           |                |          |         |
|-------------|----------|-----------|--------|---------|---------|----------|---------|---------------|------------|-----------|----------------|----------|---------|
|             | clk CPU  | clk FLASH | clk IO | clk ADC | clk ASY | 主クロック供給元 | タイマ用発振器 | INTn          | TWI アドレス一致 | タイマ/カウンタ0 | SPM EEPROM 操作可 | A/D 変換完了 | その他 I/O |
| アイドル        |          |           | ○      | ○       | ○       | ○        | ②       | ○             | ○          | ○         | ○              | ○        | ○       |
| A/D変換雑音低減   |          |           |        | ○       | ○       | ○        | ②       | ③             | ○          | ○         | ○              | ○        |         |
| パワーダウン      |          |           |        |         |         |          |         | ③             | ○          |           |                |          |         |
| パワーセーブ      |          |           |        |         | ②       |          | ②       | ③             | ○          | ②         |                |          |         |
| スタンバイ(注1)   |          |           |        |         |         | ○        |         | ③             | ○          |           |                |          |         |
| 拡張スタンバイ(注1) |          |           |        |         | ②       | ○        | ②       | ③             | ○          | ②         |                |          |         |

注1: クロック元として外部クリスタル発振子またはセラミック振動子が選ばれた場合です。

② タイマ/カウンタ0非同期状態レジスタ(ASSR)の非同期クロック(AS0)ビットが設定(1)された場合です。

③ INT3~0の割り込み、またはINT7~4のレベル割り込みだけです。

## 消費電力の最小化

これらはAVR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それぞれの検討点です。一般的に休止形態は可能な限り多く使われるべきで、休止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ばれるべきです。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。24頁の「[システム クロックとクロック選択](#)」もご覧ください。

### A/D変換器 (ADC)

許可したなら、A/D変換器は全休止形態で許可されます。節電のため、休止形態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変換器操作の詳細については154頁の「[A/D変換器](#)」を参照してください。

### アナログ比較器

アイドル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。A/D変換雑音削減動作へ移行するとき、アナログ比較器は禁止されるべきです。その他の休止形態でのアナログ比較器は自動的に禁止されます。しかしアナログ比較器が入力として内部基準電圧を使う設定の場合、全休止形態でアナログ比較器は禁止されるべきです。さもなければ内部基準電圧は休止形態と無関係に許可されます。アナログ比較器設定法の詳細については152頁の「[アナログ比較器](#)」を参照してください。

### 低電圧検出器 (BOD)

低電圧検出器(BOD)が応用で必要とされないなら、この単位部はOFFにされるべきです。低電圧検出器がBODENヒューズによって許可されていると全休止形態で許可され、故に常時電力を消費します。これはより深い休止形態での総消費電流にとって重要な一因になります。低電圧検出器(BOD)設定法の詳細については33頁の「[低電圧検出 \(BOD\)](#)」を参照してください。

### 内部基準電圧

内部基準電圧は低電圧検出器(BOD)、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。起動時間の詳細については34頁の「[内部基準電圧](#)」を参照してください。

### 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にされるべきです。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 タイマ設定法の詳細については35頁の「[ウォッチドッグ タイマ](#)」を参照してください。

### ポートピン

休止形態へ移行するとき、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(clk<sub>I/O</sub>)とA/D変換クロック(clk<sub>ADC</sub>)の両方が停止される休止形態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43頁の「[デジタル入力許可と休止形態](#)」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。

### JTAGインターフェースと内蔵デバッグ機能 (OCD)

内蔵デバッグ機能がOCDENヒューズによって許可され、チップがパワーダウンまたはパワーセーブの休止形態へ移行すると、主クロック元は許可に留まります。これらの休止形態では、これが総消費電流にとって重要な一因になります。これを避けるには3通りの方法があります。

- OCDENヒューズを禁止 (非プログラム(1)設定)
- JTAGENヒューズを禁止 (非プログラム(1)設定)
- MCU制御/状態レジスタ(MCUCSR)のJTAG禁止(JTD)ビットへの1書き込み

JTAGインターフェースが許可され、JTAG TAP制御器がデータを移動していないと、TDOピンは浮き状態のままにされます。TDOピンに接続したハードウェアが論理レベルをプルアップしないなら、消費電力が増加するでしょう。走査チェーン内の次のデバイスのTDIピンがこの問題を避けるプルアップを含むことに注意してください。MCUCSRのJTDビットに1を書くか、またはJTAGENヒューズを非プログラムのままにすることがJTAGインターフェースを禁止します。

## システム制御とリセット

### 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JMP(絶対分岐)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。これはリセットベクタが応用領域の一方、割り込みベクタがポート領域の場合やその逆も同様です。図22.の回路構成図はリセット論理回路を示します。表19.はリセット回路の電気的特性を定義します。

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。遅延タイマの遅延時間はCKSELヒューズを通して使用者によって定義されます。この遅延時間についての各種選択は24頁の「クロック元」で示されます。

### リセット元

Atmel® AVR® ATmega128には次の5つのリセット元があります。

- **電源ONリセット** …… 供給電圧が電源ONリセット閾値電圧(V<sub>POT</sub>)以下で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。
- **ウォッチドッグリセット** …… ウォッチドッグが許可され、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。
- **低電圧リセット** …… 低電圧検出器(BOD)が許可され、供給電圧(V<sub>CC</sub>)が低電圧検出電圧(V<sub>BOT</sub>)以下でMCUがリセットされます。
- **JTAG AVRリセット** …… JTAGシステムの走査チェーンの1つとしてリセットレジスタ内に論理1がある間中、MCUがリセットされます。詳細については167頁の「IEEE 1149.1 (JTAG) 境界走査」を参照してください。

図22. リセット回路構成

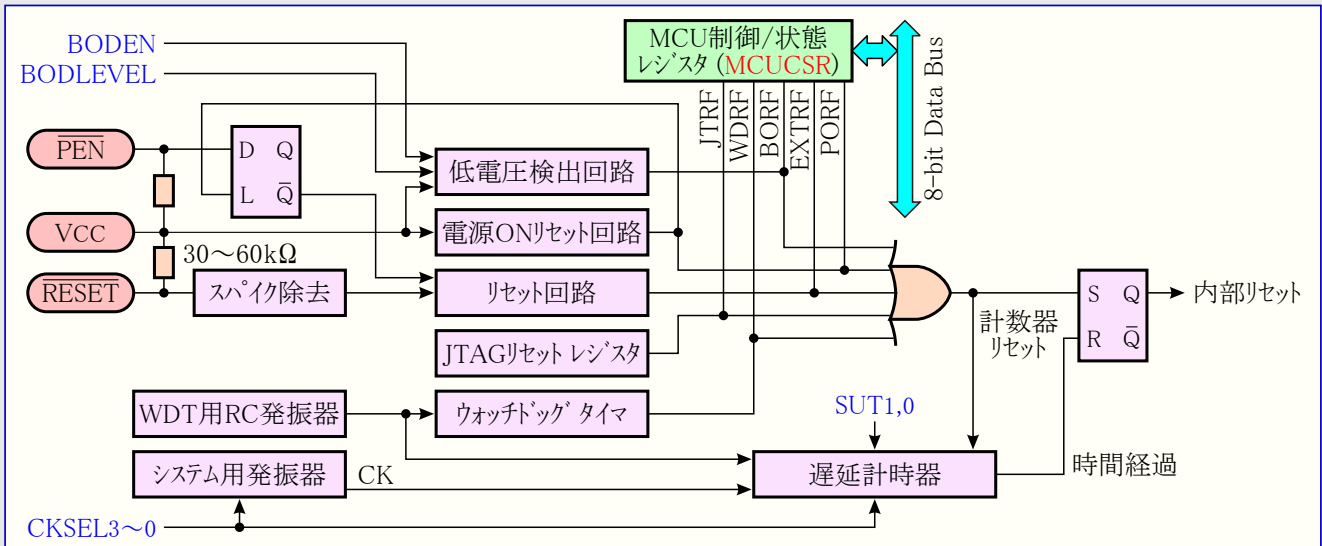


表19. リセット電気的特性

| シンボル              | 項目                   | 最小                 | 代表  | 最大                  | 単位  |    |
|-------------------|----------------------|--------------------|-----|---------------------|-----|----|
| V <sub>POT</sub>  | 上昇時電源ONリセット閾値電圧      |                    | 1.4 | 2.3                 | V   |    |
|                   | 下降時電源ONリセット閾値電圧 (注1) |                    | 1.3 | 2.3                 |     |    |
| V <sub>RST</sub>  | RESETピン閾値電圧          | 0.2V <sub>CC</sub> |     | 0.85V <sub>CC</sub> |     |    |
| t <sub>RST</sub>  | RESETピンでの最小パルス幅      | 1.5                |     |                     | μs  |    |
| V <sub>BOT</sub>  | 低電圧検出閾値電圧 (注2)       | BODLEVEL=非プログラム(1) | 2.4 | 2.6                 | 2.9 | V  |
|                   |                      | BODLEVEL=プログラム(0)  | 3.7 | 4.0                 | 4.5 |    |
| t <sub>BOT</sub>  | 最小低電圧検出時間            | BODLEVEL=非プログラム(1) |     | 2                   |     | μs |
|                   |                      | BODLEVEL=プログラム(0)  |     | 2                   |     |    |
| V <sub>HYST</sub> | 低電圧検出ヒステシス電圧         |                    | 100 |                     | mV  |    |

注1: 供給電圧がこの電圧以下にならないと、上昇時の電源ONリセットは動作しません。

注2: V<sub>BOT</sub>はいくつかのデバイスについて公称最小動作電圧以下かもしれません。この状態のデバイスについて、そのデバイスは製造検査中にV<sub>CC</sub>=V<sub>BOT</sub>へ落として検査されます。これはV<sub>CC</sub>がマイクロコントローラの正しい動作がもはや保証されない電圧へ落ちる前に低電圧(BOD)リセットが起きることを保証します。この検査はATmega128LについてはBODLEVEL=1、ATmega128についてはBODLEVEL=0を使って実行されます。ATmega128に対してBODLEVEL=1は適用できません。



## 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。検出電圧は表19で定義されます。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( $V_{POT}$ )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23. 内蔵電源ONリセット (RESETはVCCに接続)

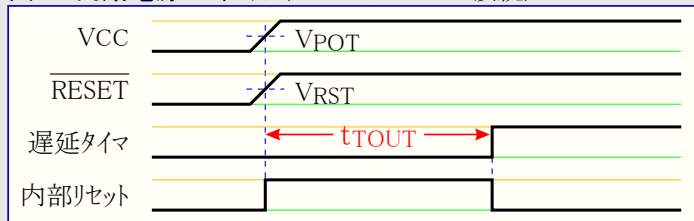
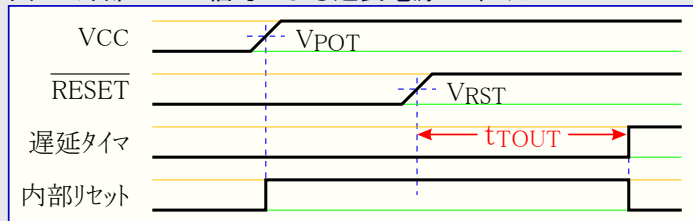


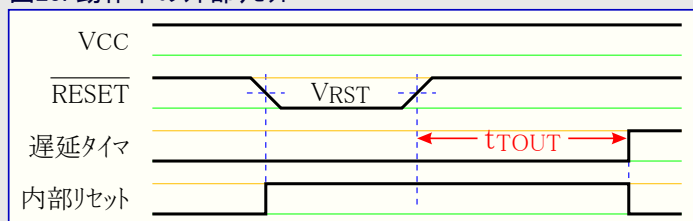
図24. 外部RESET信号による延長電源ONリセット



## 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。例えばクロックが動いていなくても、最小パルス幅(表19参照)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の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( $V_{RST}$ )に達すると(遅延タイマが起動され)、遅延タイマは遅延時間( $t_{TOUT}$ )経過後にMCUを始動します。

図25. 動作中の外部リセット



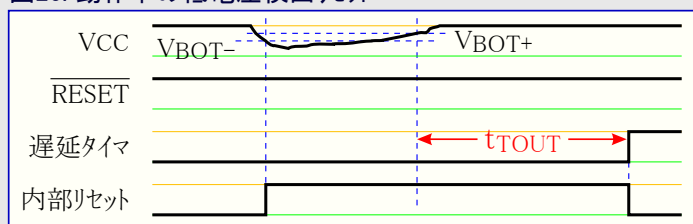
## 低電圧(ブラウンアウト)検出

ATmega128には固定化された起動(検出)電圧と比較することによって動作中のVCCを監視するチップ上の低電圧検出(BOD)回路があります。BODの起動電圧はBODLEVELヒューズにより、2.7V(非プログラム(1))または4.0V(プログラム(0))を選べます。この起動電圧はスパイク対策BODを保証するためにヒステリシスを持ちます。検出電圧のヒステリシスは $V_{BOT+}=V_{BOT+}+V_{HYST}/2$ 、 $V_{BOT-}=V_{BOT-}-V_{HYST}/2$ と解釈すべきです。

BOD回路はBODENヒューズによって許可/禁止ができます。BODが許可(BODEN=プログラム(0))され、VCCが起動電圧以下の値に下降すると(図26の $V_{BOT-}$ )、低電圧リセットが直ちに有効とされます。VCCが起動電圧以上に上昇すると(図26の $V_{BOT+}$ )、(遅延タイマが起動され、)遅延タイマは遅延時間( $t_{TOUT}$ )経過後にMCUを始動します。

BOD回路は電圧が表19で与えられる $t_{BOD}$ 時間より長く起動電圧以下に留まる場合だけ、VCCでの低下を検出します。

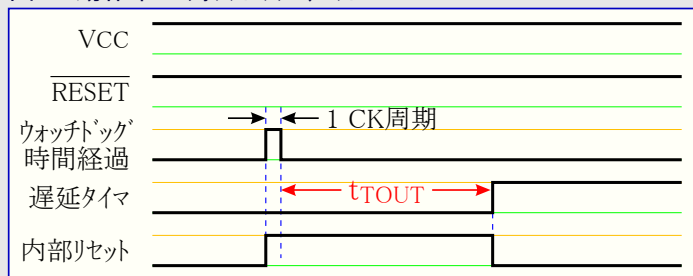
図26. 動作中の低電圧検出リセット



## ウォッチドッグリセット

ウォッチドッグ時間経過時、(内部的に)1 CK周期幅の短いリセットパルスを生成します。本パルスの下降端で遅延タイマが遅延時間( $t_{TOUT}$ )の計時を始めます。ウォッチドッグタイマ操作の詳細については35頁を参照してください。

図27. 動作中のウォッチドッグリセット





## MCU制御/状態レジスタ (MCU Control and Status Register) MCUCSR

MCU制御/状態レジスタはどのリセット元がMCUリセットを起こしたかの情報を提供します。

| ビット         | 7   | 6 | 5 | 4    | 3    | 2    | 1     | 0    |        |
|-------------|-----|---|---|------|------|------|-------|------|--------|
| \$34 (\$54) | JTD | - | - | JTRF | WDRF | BORF | EXTRF | PORF | MCUCSR |
| Read/Write  | R/W | R | R | R/W  | R/W  | R/W  | R/W   | R/W  |        |
| 初期値         | 0   | 0 | 0 | 内容参照 | 内容参照 | 内容参照 | 内容参照  | 内容参照 |        |

注: Atmel® AVR® ATmega103互換動作ではEXTRFとPORFだけ利用可能です。

### • ビット4 – JTRF : JTAG リセット フラグ (JTAG Reset Flag)

このビットはリセットがJTAG命令AVR\_RESETで選ばれたJTAGリセットレジスタ内の論理1によって引き起こされると設定(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2 – BORF : 低電圧リセット フラグ (Brown-Out Reset Flag)

このビットは低電圧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MCUCSR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、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。

## 内部基準電圧

ATmega128は内部基準電圧が特徴です。この基準電圧は低電圧検出(BOD)に使われ、A/D変換やアナログ比較器の入力としても使えます。A/D変換器への2.56V基準電圧は内部基準電圧(1.23V)から生成されます。

### 基準電圧許可信号と起動時間

この基準電圧には使われるべき方法に影響を及ぼすかもしれない起動時間があります。この起動時間は表20.で与えられます。節電のために、この基準電圧は常にONではありません。この基準電圧は次の状態中ONです。

1. 低電圧検出リセット許可時 (BODENヒューズ=プログラム(0))
2. アナログ比較器基準電圧接続時 (アナログ比較器 制御/状態レジスタ(ACSR)の基準電圧入力選択(ACBG)=1)
3. A/D変換部動作許可時 (A/D変換制御/状態レジスタ(ADCSRA)のA/D動作許可(ADEN)=1)

従って低電圧検出(BOD)が許可されていないと、ACBGの設定(=1)またはA/D変換部許可(ADEN=1)後、使用者はアナログ比較器またはA/D変換器出力が使われる前に基準電圧へ起動時間を与えなければなりません。パワーダウン動作での消費電力を減らすため、使用者はパワーダウン動作へ移行する前に基準電圧がOFFされるのを保証することで上の3つの状態を避けられます。

表20. 内部基準電圧特性

| シンボル     | 項目   | 最小   | 代表   | 最大   | 単位      |
|----------|------|------|------|------|---------|
| $V_{BG}$ | 基準電圧 | 1.15 | 1.23 | 1.40 | V       |
| $t_{BG}$ | 起動時間 |      | 40   | 70   | $\mu$ s |
| $I_{BG}$ | 消費電流 |      | 10   |      | $\mu$ A |

## ウォッチドッグ タイマ

ウォッチドッグ タイマは1MHz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。これはVCC=5Vでの代表値です。他のVCC電圧での代表値については「[特性データ](#)」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36頁の表22.で示されるように調整できます。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チップ リセットが起こる時もリセットされます。このリセット周期を決めるために8つの異なるクロック周期を選べます。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、ATmega128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 リセットの詳細タイミングについては33頁を参照してください。

予期せぬウォッチドッグ禁止や予期せぬ計時周期変更を防ぐため、3つの異なる安全基準が表21.で示されるWDTONとM103Cヒューズによって選ばれます。安全基準0はATmega103での設定に相当します。どの安全基準でもWDTの許可に制限はありません。詳細については36頁の「[ウォッチドッグ タイマ設定変更の時間制限手順](#)」を参照してください。

図28. ウォッチドッグ タイマ構成図

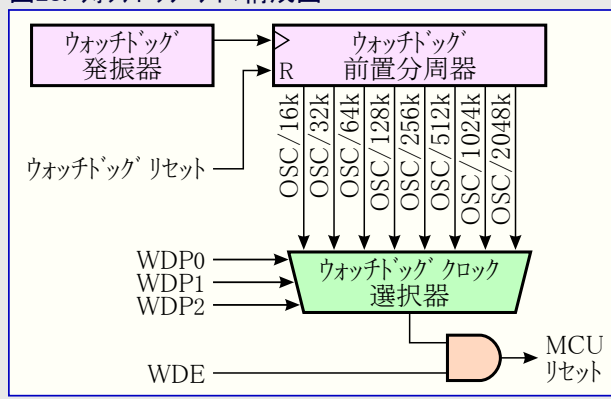


表21. M103CとWDTONヒューズ設定によるウォッチドッグ機能設定

| M103C ヒューズ | WDTON ヒューズ | 安全基準 | WDT初期状態 | WDT禁止方法  | 計時時間変更方法 |
|------------|------------|------|---------|----------|----------|
| 非プログラム(1)  | 非プログラム(1)  | 1    | 禁止      | 時間制限     | 時間制限     |
| 非プログラム(1)  | プログラム(0)   | 2    | 許可      | なし(常時許可) | 時間制限     |
| プログラム(0)   | 非プログラム(1)  | 0    | 禁止      | 時間制限     | 制限なし     |
| プログラム(0)   | プログラム(0)   | 2    | 許可      | なし(常時許可) | 時間制限     |

## ウォッチドッグ タイマ制御レジスタ (Watchdog Timer Control Register) WDTCR

| ビット         | 7 | 6 | 5 | 4    | 3   | 2    | 1    | 0    |       |
|-------------|---|---|---|------|-----|------|------|------|-------|
| \$21 (\$41) | - | - | - | WDCE | WDE | WDP2 | WDP1 | WDP0 | WDTCR |
| Read/Write  | R | R | R | R/W  | R/W | R/W  | R/W  | R/W  |       |
| 初期値         | 0 | 0 | 0 | 0    | 0   | 0    | 0    | 0    |       |

• ビット7～5 – Res : 予約 (Reserved)

これらのビットは予約されており、常に0として読みます。

• ビット4 – WDCE : ウォッチドッグ変更許可 (Watchdog Change Enable)

ウォッチドッグ許可(WDE)ビットが論理0を書かれるとき、このビットは設定(1)されなければなりません。さもなければ、ウォッチドッグは禁止されません。一旦1を書かれると、ハードウェアが4クロック周期後、このビットを解除(0)します。ウォッチドッグ禁止手順についてはWDEビットの記述を参照してください。安全基準1と2では前置分周選択ビットを変更する時も、本ビットが設定(1)されなければなりません。36頁の「[ウォッチドッグ タイマ設定変更の時間制限手順](#)」をご覧ください。

• ビット3 – WDE : ウォッチドッグ許可 (Watchdog Enable)

ウォッチドッグ許可(WDE)が論理1を書かれるとウォッチドッグ タイマが許可され、WDEが論理0を書かれるとウォッチドッグ タイマ機能が禁止されます。WDEはウォッチドッグ変更許可(WDCE)ビットが論理レベル1を持つ場合だけ解除(0)できます。許可されているウォッチドッグ 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. 同じ操作(命令)内で、ウォッチドッグ変更許可(WDCE)ビットとウォッチドッグ許可(WDE)ビットに論理1を書きます。例え禁止操作を始める前のWDEが1に設定されていても、論理1がWDEに書かれなければなりません。
2. 次からの4クロック周期内で、論理0をWDEに書きます。これがウォッチドッグを禁止します。

安全基準2では上記の手順でもウォッチドッグ タイマを禁止することができません。36頁の「[ウォッチドッグ タイマ設定変更の時間制限手順](#)」をご覧ください。

•ビット2~0 – WDP2,WDP1,WDP0 : ウォッチドッグ タイマ前置分周選択 (Watchdog Timer Prescaler 2,1 and 0)

このWDP2~0ビットはウォッチドッグ タイマが許可される時のウォッチドッグ タイマの前置分周を決めます。異なる前置分周値と対応する計時完了周期は表22.に示されます。

表22. ウォッチドッグ前置分周選択

| WDP2 | WDP1 | WDP0 | WDT発振周期数          | 代表的な計時完了周期 |          |
|------|------|------|-------------------|------------|----------|
|      |      |      |                   | VCC=3.0V   | VCC=5.0V |
| 0    | 0    | 0    | 16k (16,384)      | 14.8ms     | 14.0ms   |
| 0    | 0    | 1    | 32k (32,768)      | 29.6ms     | 28.1ms   |
| 0    | 1    | 0    | 64k (65,536)      | 59.1ms     | 56.2ms   |
| 0    | 1    | 1    | 128k (131,072)    | 0.12s      | 0.11s    |
| 1    | 0    | 0    | 256k (262,144)    | 0.24s      | 0.22s    |
| 1    | 0    | 1    | 512k (524,288)    | 0.47s      | 0.45s    |
| 1    | 1    | 0    | 1024k (1,048,576) | 0.95s      | 0.9s     |
| 1    | 1    | 1    | 2048k (2,097,152) | 1.9s       | 1.8s     |

次のコード例はウォッチドッグ(WDT)をOFFに切り替えるアセンブリ言語とC言語の関数を示します。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、それ故これら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, WDTCR   ;現WDTCR値を取得
              ORI     R16, (1<<WDCE) | (1<<WDE) ;WDCEとWDE論理1値を設定
              OUT     WDTCR, R16   ;WDCEとWDEに論理1書き込み
              LDI     R16, (0<<WDE) ;WDE論理0値を取得
              OUT     WDTCR, R16   ;ウォッチドッグ禁止
              RET                ;呼び出し元へ復帰
    
```

**C言語プログラム例**

```

void WDT_off(void)
{
    __watchdog_reset();           /* ウォッチドッグ タイマリセット */
    WDTCR |= (1<<WDCE) | (1<<WDE); /* WDCEとWDEに論理1書き込み */
    WDTCR = 0x00;                 /* ウォッチドッグ禁止 */
}
    
```

**ウォッチドッグ タイマ設定変更の時間制限手順**

ウォッチドッグ設定変更手順は安全基準間で僅かに異なります。各レベルに対して独立した手順が記述されます。

**安全基準0**

この動作種別はATmega103にみられるウォッチドッグ操作と互換性があります。ウォッチドッグ タイマは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許可(WDE)ビットに1を書くことによって許可できます。ウォッチドッグ計時完了周期は制限なしに何時でも変更できます。許可したウォッチドッグを禁止するには35頁のWDEビットの記述で示された手順に従わなければなりません。

**安全基準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許可(WDE)ビットに1を書くことにより許可できます。ウォッチドッグ計時完了周期を変更または許可したウォッチドッグ タイマを禁止する時に時間制限手順が必要とされます。許可したウォッチドッグ タイマを禁止や、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. 同じ操作(命令)でウォッチドッグ変更許可(WDCE)とWDEに論理1を書きます。WDEビットの直前の値に拘らず、論理1がWDEに書かれなければなりません。
2. 次からの4クロック周期内に同じ操作(命令)で欲したWDEとウォッチドッグ タイマ前置分周選択(WDP2~0)ビットを書きますが、WDCEビットは解除(0)されてです。

**安全基準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として読みます。時間制限手順はウォッチドッグ計時完了周期を変更する時に必要とされます。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. 同じ操作(命令)でWDCEとWDEに論理1を書きます。WDEビットが常に設定(1)されていても、WDEは時間制限手順を開始するために1を書かれなければなりません。
2. 次からの4クロック周期内に同じ操作(命令)で欲したWDP2~0ビットを書きますが、WDCEビットは解除(0)されてです。WDEビットに書かれた値は無関係です。

## 割り込み

本章はAtmel® AVR® ATmega128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11頁の「リセットと割り込みの扱い」を参照してください。

### ATmega128の割り込みベクタ

表23. 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 (注2) | 発生元              | 備考                     |
|-------|-----------------|------------------|------------------------|
| 1     | \$0000 (注1)     | リセット             | 電源ON, WDT, BOD等の各種リセット |
| 2     | \$0002          | INT0             | 外部割り込み要求0              |
| 3     | \$0004          | INT1             | 外部割り込み要求1              |
| 4     | \$0006          | INT2             | 外部割り込み要求2              |
| 5     | \$0008          | INT3             | 外部割り込み要求3              |
| 6     | \$000A          | INT4             | 外部割り込み要求4              |
| 7     | \$000C          | INT5             | 外部割り込み要求5              |
| 8     | \$000E          | INT6             | 外部割り込み要求6              |
| 9     | \$0010          | INT7             | 外部割り込み要求7              |
| 10    | \$0012          | タイマ/カウンタ2 COMP2  | タイマ/カウンタ2比較一致          |
| 11    | \$0014          | タイマ/カウンタ2 OVF2   | タイマ/カウンタ2溢れ            |
| 12    | \$0016          | タイマ/カウンタ1 CAPT1  | タイマ/カウンタ1捕獲発生          |
| 13    | \$0018          | タイマ/カウンタ1 COMP1A | タイマ/カウンタ1比較A一致         |
| 14    | \$001A          | タイマ/カウンタ1 COMP1B | タイマ/カウンタ1比較B一致         |
| 15    | \$001C          | タイマ/カウンタ1 OVF1   | タイマ/カウンタ1溢れ            |
| 16    | \$001E          | タイマ/カウンタ0 COMP0  | タイマ/カウンタ0比較一致          |
| 17    | \$0020          | タイマ/カウンタ0 OVF0   | タイマ/カウンタ0溢れ            |
| 18    | \$0022          | SPI STC          | SPI 転送完了               |
| 19    | \$0024          | USART0 RX        | USART0 受信完了            |
| 20    | \$0026          | USART0 UDRE      | USART0 送信緩衝部空き         |
| 21    | \$0028          | USART0 TX        | USART0 送信完了            |
| 22    | \$002A          | A/D変換器 ADC       | A/D変換完了                |
| 23    | \$002C          | EEPROM EE_RDY    | EEPROM 操作可             |
| 24    | \$002E          | アナログ比較器 ANA_COMP | アナログ比較器出力遷移            |
| 25    | \$0030 (注3)     | タイマ/カウンタ1 COMP1C | タイマ/カウンタ1比較C一致         |
| 26    | \$0032 (注3)     | タイマ/カウンタ3 CAPT3  | タイマ/カウンタ3捕獲発生          |
| 27    | \$0034 (注3)     | タイマ/カウンタ3 COMP3A | タイマ/カウンタ3比較A一致         |
| 28    | \$0036 (注3)     | タイマ/カウンタ3 COMP3B | タイマ/カウンタ3比較B一致         |
| 29    | \$0038 (注3)     | タイマ/カウンタ3 COMP3C | タイマ/カウンタ3比較C一致         |
| 30    | \$003A (注3)     | タイマ/カウンタ3 OVF3   | タイマ/カウンタ3溢れ            |
| 31    | \$003C (注3)     | USART1 RX        | USART1 受信完了            |
| 32    | \$003E (注3)     | USART1 UDRE      | USART1 送信緩衝部空き         |
| 33    | \$0040 (注3)     | USART1 TX        | USART1 送信完了            |
| 34    | \$0042 (注3)     | 2線直列インターフェース TWI | 2線直列インターフェース状態変化       |
| 35    | \$0044 (注3)     | SPM命令 SPM_RDY    | SPM命令操作可               |

注1: BOOTRSTヒューズがプログラム(0)されると、デバイスはリセットでブートローダアドレスへ飛びます。178頁の「ブートローダ支援 - 書き込み中読み出し可能な自己プログラミング」をご覧ください。

注2: MCU制御レジスタ(MCUCR)の割り込みベクタ選択(IVSEL)ビットが設定(1)されると、割り込みベクタはブートフラッシュ領域先頭(部)へ移動されます。そして各割り込みベクタのアドレスは、この表のアドレスがブートフラッシュ領域の先頭アドレスに加算されます。

注3: アドレス\$0030～\$0044の割り込みはATmega103互換動作では存在しません。

表24. はBOOTRST(ヒューズ)とIVSEL(割り込みベクタ選択ビット)の様々な組み合わせに対するリセットと割り込みベクタの配置を示します。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置けます。これはリセットベクタが応用領域の一方、割り込みベクタがブート領域、またはその逆の場合でも同様です。

表24. リセットと割り込みベクタの配置

| BOOTRST   | IVSEL | リセットベクタアドレス | 割り込みベクタ先頭アドレス      |
|-----------|-------|-------------|--------------------|
| 非プログラム(1) | 0     | \$0000      | \$0002             |
|           | 1     | \$0000      | ブート領域先頭アドレス+\$0002 |
| プログラム(0)  | 0     | ブート領域先頭アドレス | \$0002             |
|           | 1     | ブート領域先頭アドレス | ブート領域先頭アドレス+\$0002 |

注: ブート領域先頭アドレスは186頁の表112. で示されます。



ATmega128での最も代表的且つ一般的なりセットと割り込みのベクタ アドレス用設定を次に示します。

| アドレス   | ラベル    | 命令                     | 注釈                |
|--------|--------|------------------------|-------------------|
| \$0000 |        | JMP RESET              | ;各種リセット           |
| \$0002 |        | JMP EXT_INT0           | ;外部割り込み要求0        |
| \$0004 |        | JMP EXT_INT1           | ;外部割り込み要求1        |
| \$0006 |        | JMP EXT_INT2           | ;外部割り込み要求2        |
| \$0008 |        | JMP EXT_INT3           | ;外部割り込み要求3        |
| \$000A |        | JMP EXT_INT4           | ;外部割り込み要求4        |
| \$000C |        | JMP EXT_INT5           | ;外部割り込み要求5        |
| \$000E |        | JMP EXT_INT6           | ;外部割り込み要求6        |
| \$0010 |        | JMP EXT_INT7           | ;外部割り込み要求7        |
| \$0012 |        | JMP TIM2_COMP          | ;タイマ/カウンタ2比較一致    |
| \$0014 |        | JMP TIM2_OVF           | ;タイマ/カウンタ2溢れ      |
| \$0016 |        | JMP TIM1_CAPT          | ;タイマ/カウンタ1捕獲発生    |
| \$0018 |        | JMP TIM1_COMPA         | ;タイマ/カウンタ1比較A一致   |
| \$001A |        | JMP TIM1_COMPB         | ;タイマ/カウンタ1比較B一致   |
| \$001C |        | JMP TIM1_OVF           | ;タイマ/カウンタ1溢れ      |
| \$001E |        | JMP TIM0_COMP          | ;タイマ/カウンタ0比較一致    |
| \$0020 |        | JMP TIM0_OVF           | ;タイマ/カウンタ0溢れ      |
| \$0022 |        | JMP SPI_STC            | ;SPI転送完了          |
| \$0024 |        | JMP USART0_RXC         | ;USART0 受信完了      |
| \$0026 |        | JMP USART0_DRE         | ;USART0 送信緩衝部空    |
| \$0028 |        | JMP USART0_TXC         | ;USART0 送信完了      |
| \$002A |        | JMP ADC                | ;A/D変換完了          |
| \$002C |        | JMP EE_RDY             | ;EEPROM操作可        |
| \$002E |        | JMP ANA_COMP           | ;アナログ比較器出力遷移      |
| \$0030 |        | JMP TIM1_COMPC         | ;タイマ/カウンタ1比較C一致   |
| \$0032 |        | JMP TIM3_CAPT          | ;タイマ/カウンタ3捕獲発生    |
| \$0034 |        | JMP TIM3_COMPA         | ;タイマ/カウンタ3比較A一致   |
| \$0036 |        | JMP TIM3_COMPB         | ;タイマ/カウンタ3比較B一致   |
| \$0038 |        | JMP TIM3_COMPC         | ;タイマ/カウンタ3比較C一致   |
| \$003A |        | JMP TIM3_OVF           | ;タイマ/カウンタ3溢れ      |
| \$003C |        | JMP USART1_RXC         | ;USART1 受信完了      |
| \$003E |        | JMP USART1_DRE         | ;USART1 送信緩衝部空    |
| \$0040 |        | JMP USART1_TXC         | ;USART1 送信完了      |
| \$0042 |        | JMP TWI                | ;2線直列インターフェース状態変化 |
| \$0044 |        | JMP SPM_RDY            | ;SPM命令操作可         |
| ;      |        |                        |                   |
| \$0046 | RESET: | LDI R16, HIGH (RAMEND) | ;RAM最終アドレス上位を取得   |
| \$0047 |        | OUT SPH, R16           | ;スタック ポインタ上位を初期化  |
| \$0048 |        | LDI R16, LOW (RAMEND)  | ;RAM最終アドレス下位を取得   |
| \$0049 |        | OUT SPL, R16           | ;スタック ポインタ下位を初期化  |
|        |        | }                      | ;以下、I/O初期化など      |

BOOTRSTヒューズが非プログラム(1)、ブート領域容量が8Kバイトに設定され、どの割り込みが許可されるのにも先立ってMCU制御レジスタ(MCUCR)の割り込みベクタ選択(IVSEL)ビットが設定(1)される時の最も代表的且つ一般的なりセットと割り込みのベクタ アドレス用設定を次に示します。

| アドレス   | ラベル    | 命令                     | 注釈                          |
|--------|--------|------------------------|-----------------------------|
| \$0000 | RESET: | LDI R16, HIGH (RAMEND) | ;RAM最終アドレス上位を取得 (応用プログラム開始) |
| \$0001 |        | OUT SPH, R16           | ;スタック ポインタ上位を初期化            |
| \$0002 |        | LDI R16, LOW (RAMEND)  | ;RAM最終アドレス下位を取得             |
| \$0003 |        | OUT SPL, R16           | ;スタック ポインタ下位を初期化            |
|        |        | }                      | ;以下、I/O初期化など                |
|        | . ORG  | \$F002                 | ;ブート プログラム領域が8Kバイトの場合       |
| \$F002 |        | JMP EXT_INT0           | ;外部割り込み要求0                  |
| \$F004 |        | JMP EXT_INT1           | ;外部割り込み要求1                  |
|        |        | }                      |                             |
| \$F044 |        | JMP SPM_RDY            | ;SPM命令操作可                   |



BOOTRSTヒューズがプログラム(0)、ブート領域容量が8Kバイトに設定される時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を次に示します。

| アドレス   | ラベル    | 命令                     | 注釈                              |
|--------|--------|------------------------|---------------------------------|
|        |        | . ORG \$0002           | ;割り込みベクタ先頭                      |
| \$0002 |        | JMP EXT_INT0           | ;外部割り込み要求0                      |
| \$0004 |        | JMP EXT_INT1           | ;外部割り込み要求1                      |
| \$0044 |        | JMP SPM_RDY            | ;SPM命令操作可<br>;以下、プログラムなど        |
|        |        | . ORG \$F000           | ;ブートプログラム領域が8Kバイトの場合            |
| \$F000 | RESET: | LDI R16, HIGH (RAMEND) | ;RAM最終アドレス上位を取得 (プログラム開始)       |
| \$F001 |        | OUT SPH, R16           | ;スタックポインタ上位を初期化                 |
| \$F002 |        | LDI R16, LOW (RAMEND)  | ;RAM最終アドレス下位を取得                 |
| \$F003 |        | OUT SPL, R16           | ;スタックポインタ下位を初期化<br>;以下、I/O初期化など |

BOOTRSTヒューズがプログラム(0)、ブート領域容量が8Kバイトに設定され、どの割り込みが許可されるのにも先立ってMCU制御レジスタ(MCUCR)の割り込みベクタ選択(IVSEL)ビットが設定(1)される時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を次に示します。

| アドレス   | ラベル    | 命令                     | 注釈                              |
|--------|--------|------------------------|---------------------------------|
|        |        | . ORG \$F000           | ;ブートプログラム領域が8Kバイトの場合            |
| \$F000 |        | JMP RESET              | ;各種リセット (BOOTRSTヒューズ=0)         |
| \$F002 |        | JMP EXT_INT0           | ;外部割り込み要求0                      |
| \$F004 |        | JMP EXT_INT1           | ;外部割り込み要求1                      |
| \$F044 |        | JMP SPM_RDY            | ;SPM命令操作可                       |
| \$F046 | RESET: | LDI R16, HIGH (RAMEND) | ;RAM最終アドレス上位を取得 (プログラム開始)       |
| \$F047 |        | OUT SPH, R16           | ;スタックポインタ上位を初期化                 |
| \$F048 |        | LDI R16, LOW (RAMEND)  | ;RAM最終アドレス下位を取得                 |
| \$F049 |        | OUT SPL, R16           | ;スタックポインタ下位を初期化<br>;以下、I/O初期化など |

## 応用領域とブート領域間の割り込みベクタ移動

MCU制御レジスタ(MCUCR)は割り込みベクタ表の配置を制御します。

## MCU制御レジスタ (MCU Control Register) MCUCR

| ビット         | 7   | 6     | 5   | 4   | 3   | 2   | 1     | 0    |       |
|-------------|-----|-------|-----|-----|-----|-----|-------|------|-------|
| \$35 (\$55) | SRE | SRW10 | SE  | SM1 | SM0 | SM2 | IVSEL | IVCE | MCUCR |
| Read/Write  | R/W | R/W   | R/W | R/W | R/W | R/W | R/W   | R/W  |       |
| 初期値         | 0   | 0     | 0   | 0   | 0   | 0   | 0     | 0    |       |

### • ビット1 – IVSEL : 割り込みベクタ選択 (Interrupt Vector Select)

IVSELビットが解除(0)されると、割り込みベクタはフラッシュメモリの先頭に配置されます。このビットが設定(1)されると、割り込みベクタはフラッシュメモリのブートローダ領域の始まりへ移動されます。ブートフラッシュ領域先頭の実際のアドレスはBOOTSZヒューズによって決定されます。詳細については178頁の「ブートローダ支援 – 書き込み中読み出し可能な自己プログラミング」章を参照してください。割り込みベクタ表の予期せぬ変更を防ぐため、IVSELビットの変更は特別な書き込み手順に従わなければなりません。

1. 割り込みベクタ変更許可(IVCE)ビットに1を書きます。
2. 4周期内に、欲した値をIVSELに書き、同時に0をIVCEへ書きます。

この手順が実行される間、割り込みは自動的に禁止されます。割り込みはIVCEが設定(1)される周期で禁止され、後続のIVSELに書く命令の後まで禁止されたままです。IVSELが書かれなければ、割り込みは4周期間禁止されたままです。ステータスレジスタ(SREG)の全割り込み許可(I)ビットはこの自動禁止によって影響されません。

**注:** 割り込みベクタがブートローダ領域に配置され、BLB02ブート施錠ビットがプログラム(0)されると、応用領域から実行する間中、割り込みが禁止されます。割り込みベクタが応用領域に配置され、BLB12ブート施錠ビットがプログラム(0)されると、ブートローダ領域から実行する間中、割り込みが禁止されます。ブート施錠ビットの詳細については178頁の「ブートローダ支援 – 書き込み中読み出し可能な自己プログラミング」を参照してください。

### • ビット0 – IVCE : 割り込みベクタ変更許可 (Interrupt Vector Change Enable)

IVCEビットは割り込みベクタ選択(IVSEL)ビットの変更を許可するために論理1を書かれなければなりません。IVCEはIVSELが書かれる時、またはIVCEが書かれた後の4周期後、ハードウェアによって解除(0)されます。上記IVSELで説明されるようにIVCEビットの設定(1)は割り込みを(一時的に)禁止します。以下のコード例をご覧ください。

#### アセンブリ言語プログラム例

```

MOVE_IVT:   IN      R16, MCUCR           ; 現MCUCR値取得
            MOV     R17, R16           ; 現MCUCR値複写
            ORI    R16, (1<<IVCE)     ; IVCE論理1値を取得
            OUT    MCUCR, R16         ; IVCEに論理1書き込み
            ORI    R17, (1<<IVSEL)     ; IVSEL論理1値を取得
            OUT    MCUCR, R17         ; ブート領域へ割り込みベクタを移動
            RET                          ; 呼び出し元へ復帰
    
```

#### C言語プログラム例

```

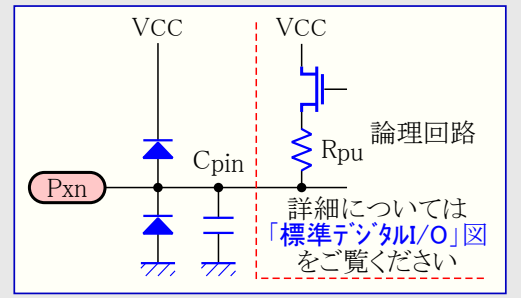
void Move_interrupts(void)
{
    uchr temp;                          /* 一時定数定義 */
    temp = MCUCR;                       /* 現MCUCR値取得 */
    MCUCR = temp | (1<<IVCE);           /* IVCEに論理1書き込み */
    MCUCR = temp | (1<<IVSEL);         /* ブート領域へ割り込みベクタを移動 */
}
    
```

## 入出力ポート

### 序説

全てのAtmel® 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向をも無意識に変更することなく、1つのポートピンの方向を変更できることを意味します。(出力として設定されていれば)駆動値を変更、または(入力として設定されていれば)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。このピン駆動部はLED(表示器)を直接駆動するのに十分な強さです。全てのポートピンは個別に選択可能な、供給電圧で抵抗値が変化しないプルアップ抵抗を持っています。全てのI/Oピンは図29.で示されるようにVCCとGNDの両方に保護ダイオードを持っています。各値の完全な一覧については206頁の「電気的特性」を参照してください。

図29. 入出力ピン等価回路



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時は正確な形式(例えば、ここで一般に記されたPORTxnがポートBのビット3に対してはPORTB3)が使われなければなりません。物理的なI/Oレジスタとビット位置は56頁の「I/Oポート用レジスタ」で一覧されます。

各々1つの出力レジスタ(PORTx)、方向レジスタ(DDRx)、入力レジスタ(PINx)の各ポートに対して、3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、一方出力レジスタと方向レジスタは読み書き(両方)です。加えて特殊I/O機能レジスタ(SFIOR)のプルアップ禁止(PUD)ビットが設定(1)されると、全ポートで全ビットに対してプルアップ機能を禁止します。

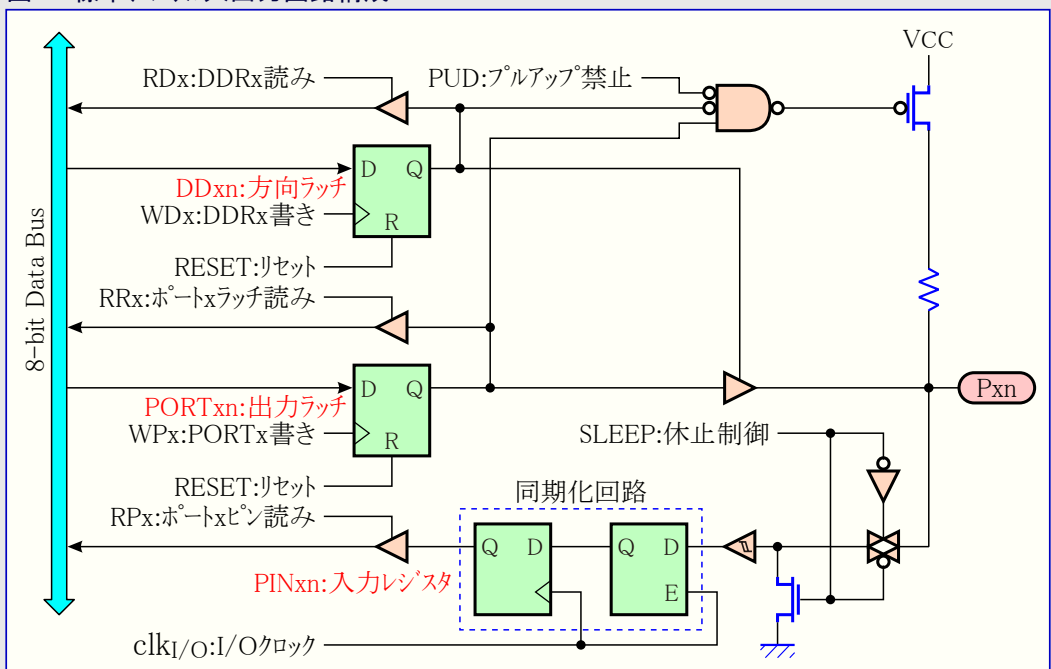
標準デジタルI/OとしてのI/Oポートの使用は次の「標準デジタル入出力としてのポート」で記述されます。多くのポートピンはデバイスの周辺機能用の交換機能と多重化されます。ポートピンとの各交換機能のインターフェース法は44頁の「交換ポート機能」で記述されます。交換機能の完全な記述については個別機能部項目を参照してください。

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ないことに注意してください。

### 標準デジタル入出力としてのポート

この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。図30.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30. 標準デジタル入出力回路構成



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

## ピンの設定

各ポートピンは3つのレジスタビット、DDxn、PORTxn、PINxnから成ります。56ページの「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アドレス、PORTxnビットはPORTx I/Oアドレス、PINxnビットはPINx I/Oアドレス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びます。DDxnが論理1を書かれるとPxnは出力ピンとして設定されます。DDxnが論理0を書かれるとPxnは入力ピンとして設定されます。

そのピンが入力ピンとして設定される時にPORTxnが論理1を書かれると、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PORTxnが論理0を書かれるか、またはそのピンが出力ピンとして設定されなければなりません。ポートピンは例えば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。

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

Hi-Z入力(DDxn=0, PORTxn=0)とHigh出力(DDxn=1, PORTxn=1)間の切り替え時、プルアップ許可入力(DDxn=0, PORTxn=1)またはLow出力(DDxn=1, PORTxn=0)のどちらかの中間状態が生じるに違いありません。通常、高インピーダンス環境は強力なHigh(吐き出し)駆動部とプルアップ間の違いに気付かないので、プルアップが許可された状態は十分受け入れられます。この事例でないなら、全ポートの全プルアップを禁止するのに特殊I/O機能レジスタ(SFIOR)のプルアップ禁止(PUD)ビットが設定(1)できます。

プルアップ入力とLow出力間の切り替えは同じ問題を発生します。使用者は中間状態としてHi-Z入力(DDxn=0, PORTxn=0)またはHigh出力(DDxn=1, PORTxn=1)のどちらかを使わなければなりません。

表25.はピン値に対する制御信号の一覧を示します。

表25. ポートピンの設定

| DDxn | PORTxn | PUD (SFIOR) | 入出力 | プルアップ抵抗 | 備考                            |
|------|--------|-------------|-----|---------|-------------------------------|
| 0    | 0      | X           | 入力  | なし      | 高インピーダンス (Hi-Z)               |
| 0    | 1      | 0           | 入力  | あり      | Pxnに外部からLowを入力すると吐き出し電流が流れます。 |
| 0    | 1      | 1           | 入力  | なし      | 高インピーダンス (Hi-Z)               |
| 1    | 0      | X           | 出力  | なし      | Low (吸い込み)出力                  |
| 1    | 1      | X           | 出力  | なし      | High (吐き出し)出力                 |

## 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30.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端(エッジ)付近で値を変える場合の未定義状態(メタステーブル)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。図31.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。

(図31.で)システムクロックの最初の上昇端の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されます。

ソフトウェアが指定したピン値を読み戻す時は、図32.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31. 外部供給ピン値読み込み時の同期化

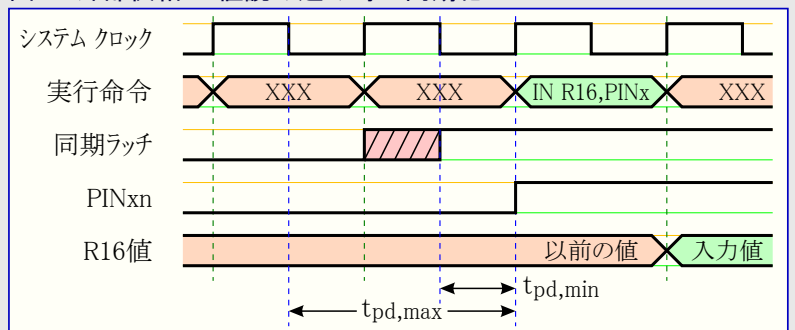
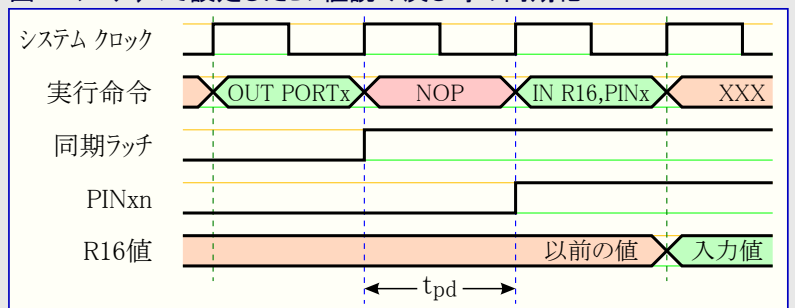


図32. プログラムで設定したピン値読み戻し時の同期化



次のコード例はポートピンの0と1をHigh出力、2と3をLow出力、6と7をプルアップ指定として4～7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<<PB7) | (1<<PB6) | (1<<PB1) | (1<<PB0)    ;プルアップとHigh値を取得
LDI    R17, (1<<DDB3) | (1<<DDB2) | (1<<DDB1) | (1<<DDB0) ;出力ビット値を取得
OUT    PORTB, R16    ;プルアップとHigh値を設定
OUT    DDRB, R17    ;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
~
;
```

#### C言語プログラム例

```

unsigned char i;
~
PORTB = (1<<PB7) | (1<<PB6) | (1<<PB1) | (1<<PB0);    /* */
DDRB = (1<<DDB3) | (1<<DDB2) | (1<<DDB1) | (1<<DDB0); /* プルアップとHigh値を設定 */
__no_operation(); /* 入出力方向を設定 */
i = PINB; /* 同期化遅延対処 */
~
/* ピン値読み戻し */
/* */
```

**注:** アセンブリ言語プログラムについてはプルアップがピン0,1,6,7に設定されてから、ビット0と1の強力なHigh駆動部としての再定義、ビット2と3のLow駆動部としての定義、方向ビットが正しく設定されるまでの時間を最小とするために2つの一時レジスタが使われます。

#### デジタル入力許可と休止形態

図30.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**、**パワーセーブ動作**、**スタンバイ動作**、**拡張スタンバイ動作**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は無視されます。外部割り込み要求が許可されないなら、SLEEPは他のピンについてと同様に有効です。SLEEPは44頁の「**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せん。

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から(復帰)再開する時に、これらの休止形態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。

#### 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のを保証することが推奨されます。例え上記のような深い休止形態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。

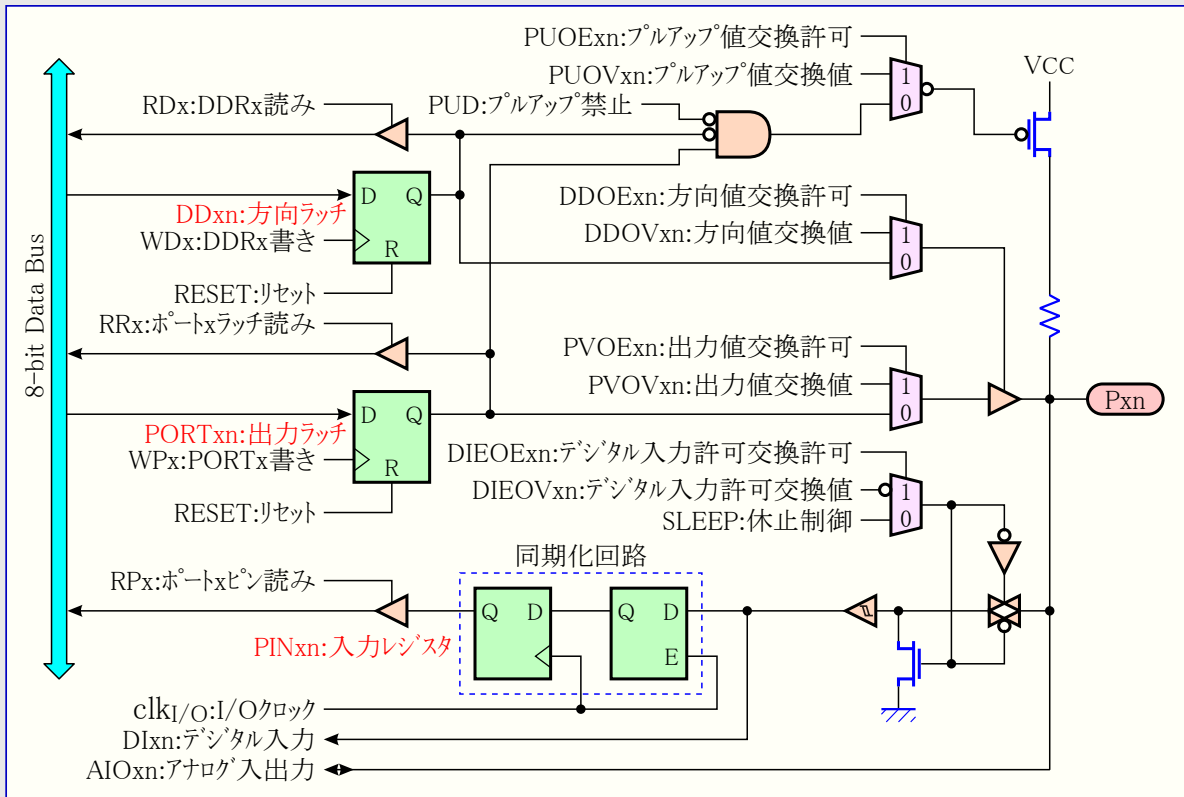
未使用ピンの定義されたレベルを保証する最も簡単な方法は内部プルアップを許可することです。この場合、リセット中のプルアップは禁止されます。リセット中の低消費電力が重要なら、外部プルアップまたはプルダウンを使うことが推奨されます。未使用ピンを直接GNDまたはVCCに接続することは、ピンが偶然に出力として設定されると過電流を引き起こす可能性があるため推奨されません。



## 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。図33.は単純化された図30.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 コントローラシステム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33. 交換ポート機能入出力回路構成



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

表26.は重複(交換)信号の機能一覧を示します。図33.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26. 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、DDxn=0, PORTxn=1, PUD=0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、DDxn, PORTxn, PUDの値に関係なく、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の場合、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。  |
| 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)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)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。この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。  |

以降の項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。

## 特殊I/O機能レジスタ (Special Function I/O Register) SFIOR

|             |     |   |   |   |      |     |      |        |       |
|-------------|-----|---|---|---|------|-----|------|--------|-------|
| ビット         | 7   | 6 | 5 | 4 | 3    | 2   | 1    | 0      |       |
| \$20 (\$40) | TSM | - | - | - | ACME | PUD | PSR0 | PSR321 | SFIOR |
| Read/Write  | R/W | R | R | R | R/W  | R/W | R/W  | R/W    |       |
| 初期値         | 0   | 0 | 0 | 0 | 0    | 0   | 0    | 0      |       |

### •ビット2 - PUD : プルアップ禁止 (Pull-up Disable)

このビットが1を書かれると、例えDDxnとPORTxnレジスタがプルアップを許可(DDxn=0, PORTxn=1)に設定されていても、I/Oポートのプルアップは禁止されます。この特徴についてより多くの詳細に関しては42頁の「[ピンの設定](#)」をご覧ください。

### ポートAの交換機能

ポートAには外部メモリ インターフェース用のアドレス下位ビットとデータ信号線としての交換機能があります。

表27. ポートAピンの交換機能

| ポートピン | 交換機能                              |
|-------|-----------------------------------|
| PA7   | AD7 (外部メモリ インターフェース アドレス/データビット7) |
| PA6   | AD6 (外部メモリ インターフェース アドレス/データビット6) |
| PA5   | AD5 (外部メモリ インターフェース アドレス/データビット5) |
| PA4   | AD4 (外部メモリ インターフェース アドレス/データビット4) |
| PA3   | AD3 (外部メモリ インターフェース アドレス/データビット3) |
| PA2   | AD2 (外部メモリ インターフェース アドレス/データビット2) |
| PA1   | AD1 (外部メモリ インターフェース アドレス/データビット1) |
| PA0   | AD0 (外部メモリ インターフェース アドレス/データビット0) |

表28.と表29.はポートAの交換機能を44頁の図33.で示される交換信号に関連付けます。

表28. ポートA7~4の交換機能用交換信号

| 信号名   | PA7/AD7  | PA6/AD6  | PA5/AD5  | PA4/AD4  |
|-------|--|--|--|--|
| PUOE  | SRE  | SRE  | SRE  | SRE  |
| PUOV  | $(\overline{WR} ADA) \cdot \text{PORTA7} \cdot \text{PUD}$ | $(\overline{WR} ADA) \cdot \text{PORTA6} \cdot \text{PUD}$ | $(\overline{WR} ADA) \cdot \text{PORTA5} \cdot \text{PUD}$ | $(\overline{WR} ADA) \cdot \text{PORTA4} \cdot \text{PUD}$ |
| DDOE  | SRE  | SRE  | SRE  | SRE  |
| DDOV  | $\overline{WR} ADA$  | $\overline{WR} ADA$  | $\overline{WR} ADA$  | $\overline{WR} ADA$  |
| PVOE  | SRE  | SRE  | SRE  | SRE  |
| PVOV  | A7·ADA D7出力· $\overline{WR}$                               | A6·ADA D6出力· $\overline{WR}$                               | A5·ADA D5出力· $\overline{WR}$                               | A4·ADA D4出力· $\overline{WR}$                               |
| DIEOE | 0  | 0  | 0  | 0  |
| DIEOV | 0  | 0  | 0  | 0  |
| DI    | D7入力   | D6入力   | D5入力   | D4入力   |
| AIO   | -  | -  | -  | -  |

表29. ポートA3~0の交換機能用交換信号

| 信号名   | PA3/AD3  | PA2/AD2  | PA1/AD1  | PA0/AD0  |
|-------|--|--|--|--|
| PUOE  | SRE  | SRE  | SRE  | SRE  |
| PUOV  | $(\overline{WR} ADA) \cdot \text{PORTA3} \cdot \text{PUD}$ | $(\overline{WR} ADA) \cdot \text{PORTA2} \cdot \text{PUD}$ | $(\overline{WR} ADA) \cdot \text{PORTA1} \cdot \text{PUD}$ | $(\overline{WR} ADA) \cdot \text{PORTA0} \cdot \text{PUD}$ |
| DDOE  | SRE  | SRE  | SRE  | SRE  |
| DDOV  | $\overline{WR} ADA$  | $\overline{WR} ADA$  | $\overline{WR} ADA$  | $\overline{WR} ADA$  |
| PVOE  | SRE  | SRE  | SRE  | SRE  |
| PVOV  | A3·ADA D3出力· $\overline{WR}$                               | A2·ADA D2出力· $\overline{WR}$                               | A1·ADA D1出力· $\overline{WR}$                               | A0·ADA D0出力· $\overline{WR}$                               |
| DIEOE | 0  | 0  | 0  | 0  |
| DIEOV | 0  | 0  | 0  | 0  |
| DI    | D3入力   | D2入力   | D1入力   | D0入力   |
| AIO   | -  | -  | -  | -  |

注: ADAはアドレス有効(ADdress Active)の略でアドレスが出力される時を表します。18頁の「[外部メモリ インターフェース](#)」をご覧ください。

## ポートBの交換機能

ポートBピンの交換機能は表30.で示されます。

表30. ポートBピンの交換機能

| ポートピン | 交換機能  |
|-------|---|
| PB7   | OC2 (タイマ/カウンタ2 比較一致出力)<br>OC1C (タイマ/カウンタ1 比較C一致出力) (注1) |
| PB6   | OC1B (タイマ/カウンタ1 比較B一致出力)                                |
| PB5   | OC1A (タイマ/カウンタ1 比較A一致出力)                                |
| PB4   | OC0 (タイマ/カウンタ0 比較一致出力)                                  |
| PB3   | MISO (SPI 主装置側データ入力/従装置側データ出力)                          |
| PB2   | MOSI (SPI 主装置側データ出力/従装置側データ入力)                          |
| PB1   | SCK (SPI 直列クロック 主装置側出力/従装置側入力)                          |
| PB0   | $\overline{SS}$ (SPI 従装置選択入力)                           |

注1: OC1CはATmega103互換動作で利用できません。

交換ピンの設定は次のとおりです。

### • OC2/OC1C – ポートB ビット7 : PB7

**OC2** : タイマ/カウンタ2の比較一致出力。PB7ピンはタイマ/カウンタ2の比較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設定(DDB7=1)されなければなりません。このOC2ピンはタイマ機能のPWM動作用出力ピンでもあります。

**OC1C** : タイマ/カウンタ1の比較C一致出力。PB7ピンはタイマ/カウンタ1の比較C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設定(DDB7=1)されなければなりません。このOC1Cピンはタイマ機能のPWM動作用出力ピンでもあります。

### • OC1B – ポートB ビット6 : PB6

**OC1B** : タイマ/カウンタ1の比較B一致出力。PB6ピンはタイマ/カウンタ1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設定(DDB6=1)されなければなりません。このOC1Bピンはタイマ機能のPWM動作用出力ピンでもあります。

### • OC1A – ポートB ビット5 : PB5

**OC1A** : タイマ/カウンタ1の比較A一致出力。PB5ピンはタイマ/カウンタ1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設定(DDB5=1)されなければなりません。このOC1Aピンはタイマ機能のPWM動作用出力ピンでもあります。

### • OC0 – ポートB ビット4 : PB4

**OC0** : タイマ/カウンタ0の比較一致出力。PB4ピンはタイマ/カウンタ0の比較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設定(DDB4=1)されなければなりません。このOC0ピンはタイマ機能のPWM動作用出力ピンでもあります。

### • MISO – ポートB ビット3 : PB3

**MISO** : SPIチャンネル用の主装置データ入力、従装置データ出力。SPIが主装置として許可されると、このピンはポートB方向レジスタ(DDRB)のDDB3の設定に拘らず、入力として設定されます。SPIが従装置として許可されると、このピンのデータ方向はDDB3によって制御されます。このピンがSPIによって入力を強制される時、プルアップは未だポートB出力レジスタ(PORTB)のPORTB3によって制御できます。

### • MOSI – ポートB ビット2 : PB2

**MOSI** : SPIチャンネル用の主装置データ出力、従装置データ入力。SPIが従装置として許可されると、このピンはポートB方向レジスタ(DDRB)のDDB2設定に拘らず、入力として設定されます。SPIが主装置として許可されると、このピンのデータ方向はDDB2によって制御されます。このピンがSPIによって入力を強制される時、プルアップは未だポートB出力レジスタ(PORTB)のPORTB2によって制御できます。

### • SCK – ポートB ビット1 : PB1

**SCK** : SPIチャンネル用の主装置クロック出力、従装置クロック入力。SPIが従装置として許可されると、このピンはポートB方向レジスタ(DDRB)のDDB1設定に拘らず、入力として設定されます。SPIが主装置として許可されると、このピンのデータ方向はDDB1によって制御されます。このピンがSPIによって入力を強制される時、プルアップは未だポートB出力レジスタ(PORTB)のPORTB1によって制御できます。

### • $\overline{SS}$ – ポートB ビット0 : PB0

**$\overline{SS}$**  : SPI従装置選択入力。SPIが従装置として許可されると、このピンはポートB方向レジスタ(DDRB)のDDB0の設定に拘らず、入力として設定されます。従装置としてこのピンがLowに駆動されるとSPI(機能)が活性(有効)にされます。SPIが主装置として許可されると、このピンのデータ方向はDDB0によって制御されます。このピンがSPIによって入力を強制される時、プルアップは未だポートB出力レジスタ(PORTB)のPORTB0によって制御できます。

表31と表32は、ポートBの交換機能を44頁の図33.で示される交換信号に関連付けます。SPI主装置入力とSPI従装置出力がMISOを構成し、一方MOSIはSPI主装置出力とSPI従装置入力に分けられます。

表31. ポートB7~4の交換機能用交換信号

| 信号名   | PB7/OC2/OC1C    | PB6/OC1B | PB5/OC1A | PB4/OC0 |
|-------|-----------------|----------|----------|---------|
| PUOE  | 0               | 0        | 0        | 0       |
| PUOV  | 0               | 0        | 0        | 0       |
| DDOE  | 0               | 0        | 0        | 0       |
| DDOV  | 0               | 0        | 0        | 0       |
| PVOE  | OC2/OC1C許可 (注1) | OC1B許可   | OC1A許可   | OC0許可   |
| PVOV  | OC2/OC1C (注1)   | OC1B     | OC1A     | OC0     |
| DIEOE | 0               | 0        | 0        | 0       |
| DIEOV | 0               | 0        | 0        | 0       |
| DI    | -               | -        | -        | -       |
| AIO   | -               | -        | -        | -       |

注1: 詳細については110頁の「比較出力変調器(COM1C2)」をご覧ください。ATmega103互換動作にOC1Cは存在しません。

表32. ポートB3~0の交換機能用交換信号

| 信号名   | PB3/MISO   | PB2/MOSI   | PB1/SCK    | PB0/SS     |
|-------|------------|------------|------------|------------|
| PUOE  | SPE·MSTR   | SPE·MSTR   | SPE·MSTR   | SPE·MSTR   |
| PUOV  | PORTB3·PUD | PORTB2·PUD | PORTB1·PUD | PORTB0·PUD |
| DDOE  | SPE·MSTR   | SPE·MSTR   | SPE·MSTR   | SPE·MSTR   |
| DDOV  | 0          | 0          | 0          | 0          |
| PVOE  | SPE·MSTR   | SPE·MSTR   | SPE·MSTR   | 0          |
| PVOV  | SPI従装置出力   | SPI主装置出力   | SCK出力      | 0          |
| DIEOE | 0          | 0          | 0          | 0          |
| DIEOV | 0          | 0          | 0          | 0          |
| DI    | SPI主装置入力   | SPI従装置入力   | SCK入力      | SPI SS     |
| AIO   | -          | -          | -          | -          |

## ポートCの交換機能

ATmega103互換動作でのポートCは出力専用です。ATmega128は既定のATmega103互換動作で出荷されます。従ってPCBへ設置される前に(ATmega103互換禁止に)プログラミングされない場合、ATmega103互換動作が禁止されるまで、ポートCは初回通電で出力になります。ポートCには外部メモリ インターフェース用のアドレス上位バイトとしての交換機能があります。

表33. ポートCピンの交換機能

| ポートピン | 交換機能                  |
|-------|-----------------------|
| PC7   | A15 (外部メモリ アドレスビット15) |
| PC6   | A14 (外部メモリ アドレスビット14) |
| PC5   | A13 (外部メモリ アドレスビット13) |
| PC4   | A12 (外部メモリ アドレスビット12) |
| PC3   | A11 (外部メモリ アドレスビット11) |
| PC2   | A10 (外部メモリ アドレスビット10) |
| PC1   | A9 (外部メモリ アドレスビット9)   |
| PC0   | A8 (外部メモリ アドレスビット8)   |

表34.と表35.はポートCの交換機能を44頁の図33.で示される交換信号に関連付けます。

表34. ポートC7~4の交換機能用交換信号

| 信号名   | PC7/A15     | PC6/A14     | PC5/A13     | PC4/A12     |
|-------|-------------|-------------|-------------|-------------|
| PUOE  | SRE・(XMM<1) | SRE・(XMM<2) | SRE・(XMM<3) | SRE・(XMM<4) |
| PUOV  | 0           | 0           | 0           | 0           |
| DDOE  | SRE・(XMM<1) | SRE・(XMM<2) | SRE・(XMM<3) | SRE・(XMM<4) |
| DDOV  | 1           | 1           | 1           | 1           |
| PVOE  | SRE・(XMM<1) | SRE・(XMM<2) | SRE・(XMM<3) | SRE・(XMM<4) |
| PVOV  | A15         | A14         | A13         | A12         |
| DIEOE | 0           | 0           | 0           | 0           |
| DIEOV | 0           | 0           | 0           | 0           |
| DI    | -           | -           | -           | -           |
| AIO   | -           | -           | -           | -           |

注1: ATmega103互換動作ではXMM=0です。

表35. ポートC3~0の交換機能用交換信号

| 信号名   | PC3/A11     | PC2/A10     | PC1/A9      | PC0/A8      |
|-------|-------------|-------------|-------------|-------------|
| PUOE  | SRE・(XMM<5) | SRE・(XMM<6) | SRE・(XMM<7) | SRE・(XMM<7) |
| PUOV  | 0           | 0           | 0           | 0           |
| DDOE  | SRE・(XMM<5) | SRE・(XMM<6) | SRE・(XMM<7) | SRE・(XMM<7) |
| DDOV  | 1           | 1           | 1           | 1           |
| PVOE  | SRE・(XMM<5) | SRE・(XMM<6) | SRE・(XMM<7) | SRE・(XMM<7) |
| PVOV  | A11         | A10         | A9          | A8          |
| DIEOE | 0           | 0           | 0           | 0           |
| DIEOV | 0           | 0           | 0           | 0           |
| DI    | -           | -           | -           | -           |
| AIO   | -           | -           | -           | -           |

注1: ATmega103互換動作ではXMM=0です。



## ポートDの交換機能

ポートDピンの交換機能は表36.で示されます。

表36. ポートDピンの交換機能

| ポートピン | 交換機能  |
|-------|---|
| PD7   | T2 (タイマ/カウンタ2 外部クロック入力)                         |
| PD6   | T1 (タイマ/カウンタ1 外部クロック入力)                         |
| PD5   | XCK1 (USART1 外部クロック入出力) (注1)                    |
| PD4   | ICP1 (タイマ/カウンタ1 捕獲起動入力)                         |
| PD3   | INT3 (外部割り込み3 入力)<br>TXD1 (USART1 送信データ出力) (注1) |
| PD2   | INT2 (外部割り込み2 入力)<br>RXD1 (USART1 受信データ入力) (注1) |
| PD1   | INT1 (外部割り込み1 入力)<br>SDA (2線直列バス データ入出力) (注1)   |
| PD0   | INT0 (外部割り込み0 入力)<br>SCL (2線直列バス クロック入出力) (注1)  |

注1: XCK1, TXD1, RXD1, SDA, SCLはATmega103互換動作では利用できません。

交換ピンの設定は次のとおりです。

• T2 - ポートD ビット7 : PD7

T2 : タイマ/カウンタ2の外部クロック入力ピンです。

• T1 - ポートD ビット6 : PD6

T1 : タイマ/カウンタ1の外部クロック入力ピンです。

• XCK1 - ポートD ビット5 : PD5

XCK1 : USART1の外部クロック入出力。ポートD方向レジスタ(DDRD)のDDD5は、このクロックが入力(DDD5=0)または出力(DDD5=1)のどちらかを制御します。XCK1ピンはUSART1が同期種別で動作する時だけ活動します(有効です)。

• ICP1 - ポートD ビット4 : PD4

ICP1 : タイマ/カウンタ1の捕獲起動入力。PD4ピンはタイマ/カウンタ1用捕獲起動入力ピンとして動作できます。

• INT3/TXD1 - ポートD ビット3 : PD3

INT3 : 外部割り込み3入力。PD3ピンはMCUへの外部割り込み元として扱えます。

TXD1 : 送信データ(USART1用データ出力ピン)。USART1送信部が許可されると、このピンはポートD方向レジスタ(DDRD)のDDD3の値に拘らず、出力として設定されます。

• INT2/RXD1 - ポートD ビット2 : PD2

INT2 : 外部割り込み2入力。PD2ピンはMCUへの外部割り込み元として扱えます。

RXD1 : 受信データ(USART1用データ入力ピン)。USART1受信部が許可されると、このピンはDDRDのDDD2の値に拘らず、入力として設定されます。USART1がこのピンを入力に強制するとき、プルアップは未だPORTD2ビットによって制御できます。

• INT1/SDA - ポートD ビット1 : PD1

INT1 : 外部割り込み1入力。PD1ピンはMCUへの外部割り込み元として扱えます。

SDA : 2線直列インターフェース データ。2線直列インターフェースを許可するために2線直列インターフェース制御レジスタ(TWCR)の2線直列インターフェース動作許可(TWEN)ビットが設定(1)されると、PD1は(通常の)ポートから切り離されて、2線直列インターフェース用直列データ入出力ピンになります。この動作では入力信号上の50nsより短いスパイク(瞬間雑音)を消去するためのスパイク除去器があり、スlewレート(上昇/下降速度)制限付きオープンドレイン駆動部によって駆動されます。このピンが2線直列インターフェースによって使われるとき、プルアップは未だPORTD1によって制御できます(訳注:共通性から本行追加)。

• INT0/SCL - ポートD ビット0 : PD0

INT0 : 外部割り込み0入力。PD0ピンはMCUへの外部割り込み元として扱えます。

SCL : 2線直列インターフェース クロック。2線直列インターフェースを許可するために2線直列インターフェース制御レジスタ(TWCR)の2線直列インターフェース動作許可(TWEN)ビットが設定(1)されると、PD0は(通常の)ポートから切り離されて、2線直列インターフェース用直列クロック入出力ピンになります。この動作では入力信号上の50nsより短いスパイク(瞬間雑音)を消去するためのスパイク除去器があり、スlewレート(上昇/下降速度)制限付きオープンドレイン駆動部によって駆動されます。このピンが2線直列インターフェースによって使われるとき、プルアップは未だPORTD0ビットによって制御できます(訳注:共通性から本行追加)。

表37.と表38.はポートDの交換機能を44頁の図33.で示される交換信号に関連付けます。

表37. ポートD7~4の交換機能用交換信号

| 信号名   | PD7/T2 | PD6/T1 | PD5/XCK1 | PD4/ICP1 |
|-------|--------|--------|----------|----------|
| PUOE  | 0      | 0      | 0        | 0        |
| PUOV  | 0      | 0      | 0        | 0        |
| DDOE  | 0      | 0      | 0        | 0        |
| DDOV  | 0      | 0      | 0        | 0        |
| PVOE  | 0      | 0      | UMSEL1   | 0        |
| PVOV  | 0      | 0      | XCK1出力   | 0        |
| DIEOE | 0      | 0      | 0        | 0        |
| DIEOV | 0      | 0      | 0        | 0        |
| DI    | T2入力   | T1入力   | XCK1入力   | ICP1入力   |
| AIO   | -      | -      | -        | -        |

表38. ポートD3~0の交換機能用交換信号

| 信号名   | PD3/INT3/TXD1 | PD2/INT2/RXD1 | PD1/INT1/SDA | PD0/INT0/SCL |
|-------|---------------|---------------|--------------|--------------|
| PUOE  | TXEN1         | RXEN1         | TWEN         | TWEN         |
| PUOV  | 0             | PORTD2・PUD    | PORTD1・PUD   | PORTD0・PUD   |
| DDOE  | TXEN1         | RXEN1         | TWEN         | TWEN         |
| DDOV  | 1             | 0             | SDA_出力       | SCL_出力       |
| PVOE  | TXEN1         | 0             | TWEN         | TWEN         |
| PVOV  | TXD1          | 0             | 0            | 0            |
| DIEOE | INT3許可        | INT2許可        | INT1許可       | INT0許可       |
| DIEOV | 1             | 1             | 1            | 1            |
| DI    | INT3入力        | INT2入力/RXD1   | INT1入力       | INT0入力       |
| AIO   | -             | -             | SDA入力        | SCL入力        |

**注:** 許可されると、2線直列インターフェースはPD0とPD1ピン出力のスレーブ制御を許可します。これは図で示されていません。  
加えて、ポート図で示されるAIO出力と2線直列インターフェース(TWI)部間にスパイク除去器が接続されます。

## ポートEの交換機能

ポートEピンの交換機能は表39.で示されます。

表39. ポートE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  |
|-------|--|-------|---|
| PE7   | INT7 (外部割り込み7 入力)<br>ICP3 (タイマ/カウンタ3 捕獲起動入力) (注1)  | PE3   | AIN1 (アナログ比較器反転入力)<br>OC3A (タイマ/カウンタ3 比較A一致出力) (注1) |
| PE6   | INT6 (外部割り込み6 入力)<br>T3 (タイマ/カウンタ3 外部クロック入力) (注1)  | PE2   | AIN0 (アナログ比較器非反転入力)<br>XCK0 (USART0 外部クロック入出力) (注1) |
| PE5   | INT5 (外部割り込み5 入力)<br>OC3C (タイマ/カウンタ3 比較C一致出力) (注1) | PE1   | TXD0 (USART0 送信データ出力)<br>PDO (直列プログラミング データ出力)      |
| PE4   | INT4 (外部割り込み4 入力)<br>OC3B (タイマ/カウンタ3 比較B一致出力) (注1) | PE0   | RXD0 (USART0 受信データ入力)<br>PDI (直列プログラミング データ入力)      |

注1: ICP3, T3, OC3C, OC3B, OC3A, XCK0はATmega103互換動作では利用できません。

交換ピンの設定は次のとおりです。

### • INT7/ICP3 – ポートE ビット7 : PE7

INT7 : 外部割り込み7入力。PE7ピンはMCUへの外部割り込み元として扱えます。

ICP3 : タイマ/カウンタ3の捕獲起動入力。PE7ピンはタイマ/カウンタ3用捕獲起動入力ピンとして動作できます。

### • INT6/T3 – ポートE ビット6 : PE6

INT6 : 外部割り込み6入力。PE6ピンはMCUへの外部割り込み元として扱えます。

T3 : タイマ/カウンタ3の外部クロック入力ピンです。

### • INT5/OC3C – ポートE ビット5 : PE5

INT5 : 外部割り込み5入力。PE5ピンはMCUへの外部割り込み元として扱えます。

OC3C : タイマ/カウンタ3の比較C一致出力。PE5ピンはタイマ/カウンタ3の比較C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設定(DDE5=1)されなければなりません。このOC3Cピンはタイマ機能のPWM動作用出力ピンでもあります。

### • INT4/OC3B – ポートE ビット4 : PE4

INT4 : 外部割り込み4入力。PE4ピンはMCUへの外部割り込み元として扱えます。

OC3B : タイマ/カウンタ3の比較B一致出力。PE4ピンはタイマ/カウンタ3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設定(DDE4=1)されなければなりません。このOC3Bピンはタイマ機能のPWM動作用出力ピンでもあります。

### • AIN1/OC3A – ポートE ビット3 : PE3

AIN1 : アナログ比較器の反転入力。このピンはアナログ比較器の反転入力に直接接続されます。

OC3A : タイマ/カウンタ3の比較A一致出力。PE3ピンはタイマ/カウンタ3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設定(DDE3=1)されなければなりません。このOC3Aピンはタイマ機能のPWM動作用出力ピンでもあります。

### • AIN0/XCK0 – ポートE ビット2 : PE2

AIN0 : アナログ比較器の非反転入力。このピンはアナログ比較器の非反転入力に直接接続されます。

XCK0 : USART0の外部クロック入出力。ポートE方向レジスタ(DDRE)のDDE2は、このクロックが入力(DDE2=0)または出力(DDE2=1)のどちらかを制御します。XCK0ピンはUSART0が同期種別で動作する時だけ活動します(有効です)。

### • TXD0/PDO – ポートE ビット1 : PE1

TXD0 : 送信データ(USART0用データ出力ピン)。USART0送信部が許可されると、このピンはポートE方向レジスタ(DDRE)のDDE1の値に拘らず、出力として設定されます。

PDO : 直列プログラミング データ出力。直列プログラミング中、このピンはATmega128のデータ出力線として使われます。

### • RXD0/PDI – ポートE ビット0 : PE0

RXD0 : 受信データ(USART0用データ入力ピン)。USART0受信部が許可されると、このピンはDDREのDDE0の値に拘らず、入力として設定されます。USART0がこのピンを入力に強制するとき、プルアップは未だPORTE0ビットによって制御できます。

PDI : 直列プログラミング データ入力。直列プログラミング中、このピンはATmega128のデータ入力線として使われます。

表40.と表41.はポートEの交換機能を44頁の図33.で示される交換信号に関連付けます。

表40. ポートE7~4の交換機能用交換信号

| 信号名  | PE7/INT7/ICP3 | PE6/INT6/T3 | PE5/INT5/OC3C | PE4/INT4/OC3B |
|------|---------------|-------------|---------------|---------------|
| PUOE | 0             | 0           | 0             | 0             |
| PUOV | 0             | 0           | 0             | 0             |
| DDOE | 0             | 0           | 0             | 0             |
| DDOV | 0             | 0           | 0             | 0             |
| PVOE | 0             | 0           | OC3C許可        | OC3B許可        |
| PVOV | 0             | 0           | OC3C          | OC3B          |
| DIOE | INT7許可        | INT6許可      | INT5許可        | INT4許可        |
| DIOV | 1             | 1           | 1             | 1             |
| DI   | INT7入力/ICP3入力 | INT6入力/T3入力 | INT5入力        | INT4入力        |
| AIO  | -             | -           | -             | -             |

表41. ポートE3~0の交換機能用交換信号

| 信号名  | PE3/AIN1/OC3A | PE2/AIN0/XCK0 | PE1/TXD0/PDO | PE0/RXD0/PDI |
|------|---------------|---------------|--------------|--------------|
| PUOE | 0             | 0             | TXEN0        | RXEN0        |
| PUOV | 0             | 0             | 0            | PORTE0・PUD   |
| DDOE | 0             | 0             | TXEN0        | RXEN0        |
| DDOV | 0             | 0             | 1            | 0            |
| PVOE | OC3A許可        | UMSEL0        | TXEN0        | 0            |
| PVOV | OC3A          | XCK0出力        | TXD0         | 0            |
| DIOE | 0             | 0             | 0            | 0            |
| DIOV | 0             | 0             | 0            | 0            |
| DI   | -             | XCK0入力        | -            | RXD0         |
| AIO  | AIN1入力        | AIN0入力        | -            | -            |



## ポートFの交換機能

ポートFピンには表42.に示されるようにA/D変換用アナログ入力としての交換機能があります。ポートFピンのいくつかが出力として設定される場合、A/D変換実行中の時にそれらを切り替えないことが重要です。これは変換結果を不正にするかもしれません。ATmega103互換動作でのポートFは入力専用です。JTAGインターフェースが許可されると、リセットが起きてもPF7(TDI)、PF5(TMS)、PF4(TCK)ピンのプルアップ抵抗は活性(有効)にされます。

表42. ポートFピンの交換機能

| ポートピン | 交換機能  |
|-------|---|
| PF7   | ADC7 (A/D変換アナログ入力チャンネル7)<br>TDI (JTAG 検査データ入力)  |
| PF6   | ADC6 (A/D変換アナログ入力チャンネル6)<br>TDO (JTAG 検査データ出力)  |
| PF5   | ADC5 (A/D変換アナログ入力チャンネル5)<br>TMS (JTAG 検査種別選択入力) |
| PF4   | ADC4 (A/D変換アナログ入力チャンネル4)<br>TCK (JTAG クロック入力)   |
| PF3   | ADC3 (A/D変換アナログ入力チャンネル3)                        |
| PF2   | ADC2 (A/D変換アナログ入力チャンネル2)                        |
| PF1   | ADC1 (A/D変換アナログ入力チャンネル1)                        |
| PF0   | ADC0 (A/D変換アナログ入力チャンネル0)                        |

交換ピンの設定は次のとおりです。

- ADC7/TDI - ポートF ビット7 : PF7

ADC7 : PF7はA/D変換チャンネル7入力としても使われます。

TDI : JTAG検査データ入力。命令レジスタまたはデータレジスタ(走査チェーン)に移動入力される直列入力データです。JTAGインターフェースが許可されると、このピンはI/Oピンとして使えません。

- ADC6/TDO - ポートF ビット6 : PF6

ADC6 : PF6はA/D変換チャンネル6入力としても使われます。

TDO : JTAG検査データ出力。命令レジスタまたはデータレジスタから移動出力される直列出力データです。JTAGインターフェースが許可されたなら、このピンはI/Oピンとして使えません。TDOピンはデータが移動出力されるTAP状態への移行を除いてHi-Zです。

- ADC5/TMS - ポートF ビット5 : PF5

ADC5 : PF5はA/D変換チャンネル5入力としても使われます。

TMS : JTAG検査種別選択。このピンはTAP(検査入出力ポート)制御器状態機構を通しての操作に使われます。JTAGインターフェースが許可されると、このピンはI/Oピンとして使えません。

- ADC4/TCK - ポートF ビット4 : PF4

ADC4 : PF4はA/D変換チャンネル4入力としても使われます。

TCK : JTAGクロック。JTAG操作はTCKに同期します。JTAGインターフェースが許可されると、このピンはI/Oピンとして使えません。

- ADC3 - ポートF ビット3 : PF3

ADC3 : PF3はA/D変換チャンネル3入力としても使われます。

- ADC2 - ポートF ビット2 : PF2

ADC2 : PF2はA/D変換チャンネル2入力としても使われます。

- ADC1 - ポートF ビット1 : PF1

ADC1 : PF1はA/D変換チャンネル1入力としても使われます。

- ADC0 - ポートF ビット0 : PF0

ADC0 : PF0はA/D変換チャンネル0入力としても使われます。

表43.と表44.はポートFの交換機能を44頁の図33.で示される交換信号に関連付けます。

表43. ポートF7~4の交換機能用交換信号

| 信号名  | PF7/ADC7/TDI | PF6/ADC/TDO | PF5/ADC5/TMS | PF4/ADC4/TCK |
|------|--------------|-------------|--------------|--------------|
| PUOE | JTAGEN       | JTAGEN      | JTAGEN       | JTAGEN       |
| PUOV | 1            | 0           | 1            | 1            |
| DDOE | JTAGEN       | JTAGEN      | JTAGEN       | JTAGEN       |
| DDOV | 0            | IR移動+DR移動   | 0            | 0            |
| PVOE | 0            | JTAGEN      | 0            | 0            |
| PVOV | 0            | TDO         | 0            | 0            |
| DIOE | JTAGEN       | JTAGEN      | JTAGEN       | JTAGEN       |
| DIOV | 0            | 0           | 0            | 0            |
| DI   | -            | -           | -            | -            |
| AIO  | ADC7入力/TDI入力 | ADC6入力      | ADC5入力/TMS入力 | ADC4入力/TCK入力 |

表44. ポートF3~0の交換機能用交換信号

| 信号名  | PF3/ADC3 | PF2/ADC2 | PF1/ADC1 | PF0/ADC0 |
|------|----------|----------|----------|----------|
| PUOE | 0        | 0        | 0        | 0        |
| PUOV | 0        | 0        | 0        | 0        |
| DDOE | 0        | 0        | 0        | 0        |
| DDOV | 0        | 0        | 0        | 0        |
| PVOE | 0        | 0        | 0        | 0        |
| PVOV | 0        | 0        | 0        | 0        |
| DIOE | 0        | 0        | 0        | 0        |
| DIOV | 0        | 0        | 0        | 0        |
| DI   | -        | -        | -        | -        |
| AIO  | ADC3入力   | ADC2入力   | ADC1入力   | ADC0入力   |

## ポートGの交換機能

ATmega103互換動作では交換機能だけがポートGの既定で、ポートGは標準デジタルポートピンとして使えません。交換ピン設定は次のとおりです。

表45. ポートGピンの交換機能

| ポートピン | 交換機能                                 |
|-------|--------------------------------------|
| PG4   | TOSC1 (タイマ用発振増幅器入力)                  |
| PG3   | TOSC2 (タイマ用発振増幅器出力)                  |
| PG2   | ALE (外部メモリ用アドレスラッチ許可信号出力)            |
| PG1   | $\overline{RD}$ (外部メモリ用読み出しスローブ信号出力) |
| PG0   | $\overline{WR}$ (外部メモリ用書き込みスローブ信号出力) |

交換ピンの設定は次のとおりです。

- TOSC1 - ポートGビット4 : PG4

TOSC1 : タイマ発振器ピン1。タイマ/カウンタ0の非同期クロック動作を許可するためにASSRの非同期許可(AS0)ビットが設定(1)されると、PG4ピンはポートから切り離され、発振用反転増幅器の入力になります。この動作では、このピンに水晶発振器が接続され、このピンはI/Oピンとして使えません。

- TOSC2 - ポートGビット3 : PG3

TOSC2 : タイマ発振器ピン2。タイマ/カウンタ0の非同期クロック動作を許可するためにASSRの非同期許可(AS0)ビットが設定(1)されると、PG3ピンはポートから切り離され、発振用反転増幅器の出力になります。この動作では、このピンに水晶発振器が接続され、このピンはI/Oピンとして使えません。

- ALE - ポートGビット2 : PG2

ALE : 外部メモリ用アドレスラッチ許可信号出力ピンです。

- $\overline{RD}$  - ポートGビット1 : PG1

$\overline{RD}$  : 外部メモリ用読み出し制御スローブ信号出力ピンです。

- $\overline{WR}$  - ポートGビット0 : PG0

$\overline{WR}$  : 外部メモリ用書き込み制御スローブ信号出力ピンです。

表46.はポートGの交換機能を44頁の図33.で示される交換信号に関連付けます。

表46. ポートG4~0の交換機能用交換信号

| 信号名   | PG4/TOSC1        | PG3/TOSC2        | PG2/ALE | PG1/ $\overline{RD}$ | PG0/ $\overline{WR}$ |
|-------|------------------|------------------|---------|----------------------|----------------------|
| PUOE  | AS0              | AS0              | SRE     | SRE                  | SRE                  |
| PUOV  | 0                | 0                | 0       | 0                    | 0                    |
| DDOE  | AS0              | AS0              | SRE     | SRE                  | SRE                  |
| DDOV  | 0                | 0                | 1       | 1                    | 1                    |
| PVOE  | 0                | 0                | SRE     | SRE                  | SRE                  |
| PVOV  | 0                | 0                | ALE     | RD                   | WR                   |
| DIEOE | AS0              | AS0              | 0       | 0                    | 0                    |
| DIEOV | 0                | 0                | 0       | 0                    | 0                    |
| DI    | -                | -                | -       | -                    | -                    |
| AIO   | T/C0用<br>発振増幅器入力 | T/C0用<br>発振増幅器出力 | -       | -                    | -                    |

(訳注) 原書の表46.と表47.は表46.として統合しました。

## I/Oポート用レジスタ

### ポートA出力レジスタ (Port A Data Register) PORTA

| ビット         | 7      | 6      | 5      | 4      | 3      | 2      | 1      | 0      |       |
|-------------|--------|--------|--------|--------|--------|--------|--------|--------|-------|
| \$1B (\$3B) | PORTA7 | PORTA6 | PORTA5 | PORTA4 | PORTA3 | PORTA2 | PORTA1 | PORTA0 | PORTA |
| Read/Write  | R/W    | R/W    | R/W    | R/W    | R/W    | R/W    | R/W    | R/W    |       |
| 初期値         | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      |       |

### ポートA方向レジスタ (Port A Data Direction Register) DDRA

| ビット         | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |      |
|-------------|------|------|------|------|------|------|------|------|------|
| \$1A (\$3A) | DDA7 | DDA6 | DDA5 | DDA4 | DDA3 | DDA2 | DDA1 | DDA0 | DDRA |
| Read/Write  | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  |      |
| 初期値         | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |      |

### ポートA入力レジスタ (Port A Input Address) PINA

| ビット         | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |      |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|------|
| \$19 (\$39) | PINA7 | PINA6 | PINA5 | PINA4 | PINA3 | PINA2 | PINA1 | PINA0 | PINA |
| Read/Write  | R     | R     | R     | R     | R     | R     | R     | R     |      |
| 初期値         | 不定    | 不定    | 不定    | 不定    | 不定    | 不定    | 不定    | 不定    |      |

### ポートB出力レジスタ (Port B Data Register) PORTB

| ビット         | 7      | 6      | 5      | 4      | 3      | 2      | 1      | 0      |       |
|-------------|--------|--------|--------|--------|--------|--------|--------|--------|-------|
| \$18 (\$38) | PORTB7 | PORTB6 | PORTB5 | PORTB4 | PORTB3 | PORTB2 | PORTB1 | PORTB0 | PORTB |
| Read/Write  | R/W    | R/W    | R/W    | R/W    | R/W    | R/W    | R/W    | R/W    |       |
| 初期値         | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      |       |

### ポートB方向レジスタ (Port B Data Direction Register) DDRB

| ビット         | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |      |
|-------------|------|------|------|------|------|------|------|------|------|
| \$17 (\$37) | DDB7 | DDB6 | DDB5 | DDB4 | DDB3 | DDB2 | DDB1 | DDB0 | DDRB |
| Read/Write  | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  |      |
| 初期値         | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |      |

### ポートB入力レジスタ (Port B Input Address) PINB

| ビット         | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |      |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|------|
| \$16 (\$36) | PINB7 | PINB6 | PINB5 | PINB4 | PINB3 | PINB2 | PINB1 | PINB0 | PINB |
| Read/Write  | R     | R     | R     | R     | R     | R     | R     | R     |      |
| 初期値         | 不定    | 不定    | 不定    | 不定    | 不定    | 不定    | 不定    | 不定    |      |

### ポートC出力レジスタ (Port C Data Register) PORTC

| ビット         | 7      | 6      | 5      | 4      | 3      | 2      | 1      | 0      |       |
|-------------|--------|--------|--------|--------|--------|--------|--------|--------|-------|
| \$15 (\$35) | PORTC7 | PORTC6 | PORTC5 | PORTC4 | PORTC3 | PORTC2 | PORTC1 | PORTC0 | PORTC |
| Read/Write  | R/W    | R/W    | R/W    | R/W    | R/W    | R/W    | R/W    | R/W    |       |
| 初期値         | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      |       |

### ポートC方向レジスタ (Port C Data Direction Register) DDRC

| ビット         | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |      |
|-------------|------|------|------|------|------|------|------|------|------|
| \$14 (\$34) | DDC7 | DDC6 | DDC5 | DDC4 | DDC3 | DDC2 | DDC1 | DDC0 | DDRC |
| Read/Write  | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  |      |
| 初期値         | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |      |



## ポートC入力レジスタ (Port C Input Address) PINC

| ビット         | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |      |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|------|
| \$13 (\$33) | PINC7 | PINC6 | PINC5 | PINC4 | PINC3 | PINC2 | PINC1 | PINC0 | PINC |
| Read/Write  | R     | R     | R     | R     | R     | R     | R     | R     |      |
| 初期値         | 不定    | 不定    | 不定    | 不定    | 不定    | 不定    | 不定    | 不定    |      |

Atmel® AVR® ATmega103互換動作でのDDRCとPINCレジスタはプッシュプル0(Low)出力に初期化されます。ポートピンは例えクロックが走行していなくてもそれらの初期値になります。DDRCとPINCレジスタはATmega103互換動作でも利用可能ですが、過去との100%互換のためには使われるべきでないことに注意してください。

## ポートD出力レジスタ (Port D Data Register) PORTD

| ビット         | 7      | 6      | 5      | 4      | 3      | 2      | 1      | 0      |       |
|-------------|--------|--------|--------|--------|--------|--------|--------|--------|-------|
| \$12 (\$32) | PORTD7 | PORTD6 | PORTD5 | PORTD4 | PORTD3 | PORTD2 | PORTD1 | PORTD0 | PORTD |
| Read/Write  | R/W    | R/W    | R/W    | R/W    | R/W    | R/W    | R/W    | R/W    |       |
| 初期値         | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      |       |

## ポートD方向レジスタ (Port D Data Direction Register) DDRD

| ビット         | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |      |
|-------------|------|------|------|------|------|------|------|------|------|
| \$11 (\$31) | DDD7 | DDD6 | DDD5 | DDD4 | DDD3 | DDD2 | DDD1 | DDD0 | DDRD |
| Read/Write  | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  |      |
| 初期値         | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |      |

## ポートD入力レジスタ (Port D Input Address) PIND

| ビット         | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |      |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|------|
| \$10 (\$30) | PIND7 | PIND6 | PIND5 | PIND4 | PIND3 | PIND2 | PIND1 | PIND0 | PIND |
| Read/Write  | R     | R     | R     | R     | R     | R     | R     | R     |      |
| 初期値         | 不定    | 不定    | 不定    | 不定    | 不定    | 不定    | 不定    | 不定    |      |

## ポートE出力レジスタ (Port E Data Register) PORTE

| ビット         | 7      | 6      | 5      | 4      | 3      | 2      | 1      | 0      |       |
|-------------|--------|--------|--------|--------|--------|--------|--------|--------|-------|
| \$03 (\$23) | PORTE7 | PORTE6 | PORTE5 | PORTE4 | PORTE3 | PORTE2 | PORTE1 | PORTE0 | PORTE |
| Read/Write  | R/W    | R/W    | R/W    | R/W    | R/W    | R/W    | R/W    | R/W    |       |
| 初期値         | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      |       |

## ポートE方向レジスタ (Port E Data Direction Register) DDRE

| ビット         | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |      |
|-------------|------|------|------|------|------|------|------|------|------|
| \$02 (\$22) | DDE7 | DDE6 | DDE5 | DDE4 | DDE3 | DDE2 | DDE1 | DDE0 | DDRE |
| Read/Write  | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  |      |
| 初期値         | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |      |

## ポートE入力レジスタ (Port E Input Address) PINE

| ビット         | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |      |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|------|
| \$01 (\$21) | PINE7 | PINE6 | PINE5 | PINE4 | PINE3 | PINE2 | PINE1 | PINE0 | PINE |
| Read/Write  | R     | R     | R     | R     | R     | R     | R     | R     |      |
| 初期値         | 不定    | 不定    | 不定    | 不定    | 不定    | 不定    | 不定    | 不定    |      |

## ポートF出力レジスタ (Port F Data Register) PORTF

| ビット        | 7      | 6      | 5      | 4      | 3      | 2      | 1      | 0      |       |
|------------|--------|--------|--------|--------|--------|--------|--------|--------|-------|
| (\$62)     | PORTF7 | PORTF6 | PORTF5 | PORTF4 | PORTF3 | PORTF2 | PORTF1 | PORTF0 | PORTF |
| Read/Write | R/W    | R/W    | R/W    | R/W    | R/W    | R/W    | R/W    | R/W    |       |
| 初期値        | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      |       |

### ポートF方向レジスタ (Port F Data Direction Register) DDRF

| ビット (\$61) | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |      |
|------------|------|------|------|------|------|------|------|------|------|
|            | DDF7 | DDF6 | DDF5 | DDF4 | DDF3 | DDF2 | DDF1 | DDF0 | DDRF |
| Read/Write | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  |      |
| 初期値        | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |      |

### ポートF入力レジスタ (Port F Input Address) PINF

| ビット (\$00 (\$20)) | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |      |
|-------------------|-------|-------|-------|-------|-------|-------|-------|-------|------|
|                   | PINF7 | PINF6 | PINF5 | PINF4 | PINF3 | PINF2 | PINF1 | PINF0 | PINF |
| Read/Write        | R     | R     | R     | R     | R     | R     | R     | R     |      |
| 初期値               | 不定    | 不定    | 不定    | 不定    | 不定    | 不定    | 不定    | 不定    |      |

ポートFはデジタル入力のみとして扱うATmega103互換動作で、PORTFとDDRFレジスタが利用できないことに注意してください。

### ポートG出力レジスタ (Port G Data Register) PORTG

| ビット (\$65) | 7 | 6 | 5 | 4      | 3      | 2      | 1      | 0      |       |
|------------|---|---|---|--------|--------|--------|--------|--------|-------|
|            | - | - | - | PORTG4 | PORTG3 | PORTG2 | PORTG1 | PORTG0 | PORTG |
| Read/Write | R | R | R | R/W    | R/W    | R/W    | R/W    | R/W    |       |
| 初期値        | 0 | 0 | 0 | 0      | 0      | 0      | 0      | 0      |       |

### ポートG方向レジスタ (Port G Data Direction Register) DDRG

| ビット (\$64) | 7 | 6 | 5 | 4    | 3    | 2    | 1    | 0    |      |
|------------|---|---|---|------|------|------|------|------|------|
|            | - | - | - | DDG4 | DDG3 | DDG2 | DDG1 | DDG0 | DDRG |
| Read/Write | R | R | R | R/W  | R/W  | R/W  | R/W  | R/W  |      |
| 初期値        | 0 | 0 | 0 | 0    | 0    | 0    | 0    | 0    |      |

### ポートG入力レジスタ (Port G Input Address) PING

| ビット (\$63) | 7 | 6 | 5 | 4     | 3     | 2     | 1     | 0     |      |
|------------|---|---|---|-------|-------|-------|-------|-------|------|
|            | - | - | - | PING4 | PING3 | PING2 | PING1 | PING0 | PING |
| Read/Write | R | R | R | R     | R     | R     | R     | R     |      |
| 初期値        | 0 | 0 | 0 | 不定    | 不定    | 不定    | 不定    | 不定    |      |

PORTG, DDRG, PINGレジスタがATmega103互換動作で利用できないことに注意してください。ATmega103互換動作でのポートGは交換機能(TOSC1, TOSC2, ALE,  $\overline{RD}$ ,  $\overline{WR}$ )だけを扱います。

## 外部割り込み

外部割り込みはINT7~0ピンによって起動されます。許可したなら、例えINT7~0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。外部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。これは外部割り込み制御レジスタ(EICRA(INT3~0)とEICRB(INT7~4))の詳述で示される設定です。外部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。INT7~4の上昇端または下降端割り込みの認知は24頁の「クロックシステムとその配給」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。INT7~4のLowレベル割り込みとINT3~0の端(エッジ)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、変化したレベルはMCUを起動するために一定時間保たれなければならないことに注意してください。これはMCUの雑音不安定性を減らします。変化したレベルはウォッチドッグ発振器クロックによって2度採取されます。ウォッチドッグ発振器の周期は5.0V, 25°Cで(公称)1μsです。ウォッチドッグ発振器の周波数は214頁の「代表特性」で示されるように電圧依存です。この採取の間、入力が必要としたレベルであるか、または起動時間の最後まで保たれれば、MCUは起動します。起動時間は24頁の「システムクロックとクロック選択」で示されるようにSUTヒューズで定義されます。このレベルがウォッチドッグ発振器クロックによって2度採取されるけれど、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。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たれなければなりません。

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

| ビット (\$6A) | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     | EICRA |
|------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
|            | ISC31 | ISC30 | ISC21 | ISC20 | ISC11 | ISC10 | ISC01 | ISC00 |       |
| Read/Write | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   |       |
| 初期値        | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |       |

このレジスタはAtmel® AVR® ATmega103互換動作でやり取り(アクセス)できませんが、初期値はATmega103と(同じ)Lowレベル割り込みとしてINT3~0を定義(設定)します。

#### • ビット7~0 - ISC31, ISC30~ISC01, ISC00 : 外部割り込み3~0条件制御 (External Interrupt3~0 Sense Control Bits)

外部割り込み3~0はステータスレジスタ(SREG)の全割り込み許可(D)ビットと外部割り込み許可レジスタ(EIMSK)の対応する割り込み許可ビットが設定(1)される場合のINT3~0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INT3~0ピンの端(エッジ)とレベルは表48.で定義されます。INT3~0の端は非同期に記録されます。表49.で与えられた最小パルス幅より広いINT3~0の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の発生が保証されません。Lowレベル割り込みが選ばれる場合、Lowレベルは割り込みを生成するため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。許可したなら、レベルで起動する割り込みはそのピンがLowを保持される限り、割り込み要求を(連続的に)生成します。ISCnビット変更時、割り込みが起き得ます。従って最初にEIMSKの割り込み許可ビットを解除(0)することでINTn割り込みを禁止することが推奨されます。その後ISCnビットは変更できます。最後にINTn割り込み要求フラグは割り込みが再び許可される前に外部割り込み要求フラグレジスタ(EIFR)の外部割り込みn要求(INTFn)フラグに論理1を書くことによって解除(0)されるべきです。

表48. 外部割り込み3~0 割り込み条件

| ISCn1 | ISCn0 | 割り込み発生条件      |
|-------|-------|---------------|
| 0     | 0     | INTnピンのLowレベル |
| 0     | 1     | (予約)          |
| 1     | 0     | INTnピンの下降端    |
| 1     | 1     | INTnピンの上昇端    |

注: n=3~0

ISCn1,0ビット変更時、割り込みはEIMSKで割り込み許可ビットの解除(0)によって禁止されなければなりません。さもなければビットが変更される時に割り込みが起き得ます。

表49. 非同期外部割り込み特性

| シンボル             | 項目              | 最小 | 代表 | 最大 | 単位 |
|------------------|-----------------|----|----|----|----|
| t <sub>INT</sub> | 非同期外部割り込み最小パルス幅 |    | 50 |    | ns |

## 外部割り込み制御レジスタB (External Interrupt Control Register B) EICRB

| ビット         | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |       |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| \$3A (\$5A) | ISC71 | ISC70 | ISC61 | ISC60 | ISC51 | ISC50 | ISC41 | ISC40 | EICRB |
| Read/Write  | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   |       |
| 初期値         | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |       |

### •ビット7～0 – ISC71, ISC70～ISC41, ISC40 : 外部割り込み7～4条件制御 (External Interrupt7～4 Sense Control Bits)

外部割り込み7～4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)の対応する割り込み許可ビットが設定(1)される場合のINT7～4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INT7～4ピンの端(エッジ)とレベルは表50.で定義されます。INT7～4ピンの値は端検知前に採取されています。端または論理変化割り込みが選ばれる場合、1クロック周期より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の発生が保証されません。XTAL分周器が許可される場合、CPUクロック周波数がXTAL周波数よりも低くできることを考慮してください。Lowレベル割り込みが選ばれる場合、Lowレベルは割り込みを生成するために現在実行している命令の完了まで保たなければなりません。許可したなら、レベルで起動する割り込みはそのピンがLowを保持される限り割り込み要求を(連続的に)生成します。

表50. 外部割り込み7～4 割り込み条件

| ISCn1 | ISCn0 | 割り込み発生条件        |
|-------|-------|-----------------|
| 0     | 0     | INTnピンのLowレベル   |
| 0     | 1     | INTnピンの論理変化(両端) |
| 1     | 0     | INTnピンの下降端      |
| 1     | 1     | INTnピンの上昇端      |

注: n=7～4

ISCn1,0ビット変更時、割り込みはEIMSKで割り込み許可ビットの解除(0)によって禁止されなければなりません。さもなければビットが変更される時に割り込みが起き得ます。

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

| ビット         | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |       |
|-------------|------|------|------|------|------|------|------|------|-------|
| \$39 (\$59) | INT7 | INT6 | INT5 | INT4 | INT3 | INT2 | INT1 | INT0 | EIMSK |
| Read/Write  | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  |       |
| 初期値         | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |       |

### •ビット7～0 – INT7～INT0 : 外部割り込み7～0 許可 (External Interrupt Request 7～0 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、INT7～0ビットが1を書かれると、対応する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(EICRAとEICRB)の割り込み条件制御n(ISCn1,ISCn0)ビットは、その外部割り込みが上昇端、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。例えそのピンが出力として設定されても、そのピンの動きは割り込み要求を引き起こします。これはソフトウェア割り込みを生成する方法を提供します。

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

| ビット         | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |      |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|------|
| \$38 (\$58) | INTF7 | INTF6 | INTF5 | INTF4 | INTF3 | INTF2 | INTF1 | INTF0 | EIFR |
| Read/Write  | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   |      |
| 初期値         | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |      |

### •ビット7～0 – INTF7～INTF0 : 外部割り込み7～0要求フラグ (External Interrupt Flag7～0)

INT7～0ピン上の端(エッジ)または論理変化が割り込み要求を起動すると、INTF7～0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)の対応する割り込み許可(INT7～0)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。INT7～0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。INT3～0割り込み禁止で休止形態へ移行すると、これらのピンの入力緩衝部が禁止されることに注意してください。これはINTF3～0フラグを設定(1)する内部信号の論理変化を起こすかもしれません。より多くの情報については43頁の「デジタル入力許可と休止形態」をご覧ください。

## 8ビット タイマ/カウンタ0 (PWM, 非同期動作付き)

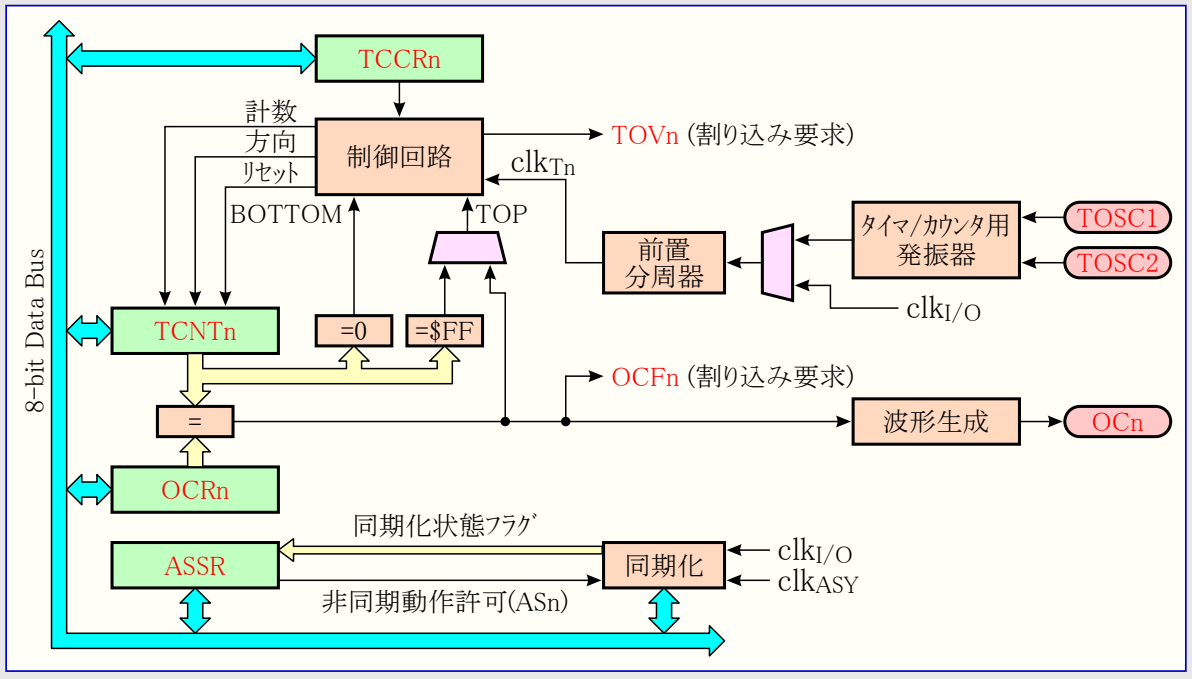
タイマ/カウンタ0は単一比較部付きの汎用8ビット タイマ/カウンタ部です。主な特徴は次のとおりです。

- 単一比較部付き計数器
- 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)
- 周波数発生器
- クロック用10ビット前置分周器
- 溢れと比較一致割り込み (TOV0とOCF0)
- I/O(システム)クロックに依存しない時計用外部32kHzクリスタルからのクロック駆動可能

### 概要

この8ビット タイマ/カウンタの簡便化した構成図は図34.で示されます。I/Oピンの実際の配置については2頁の「ピン配置」を参照してください。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:原文は太字)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69頁の「8ビット タイマ/カウンタ用レジスタ」で一覧されます。

図34. 8ビット タイマ/カウンタ構成図



### 関係レジスタ

タイマ/カウンタ(TCNT0)と比較レジスタ(OCR0)は8ビットのレジスタです。割り込み要求信号はタイマ/カウンタ割り込み要求レジスタ(TIFR)で全て見えます。全ての割り込みはタイマ/カウンタ割り込み許可レジスタ(TIMSK)で個別に遮蔽(禁止)されます。TIFRとTIMSKはこれらのレジスタが他のタイマ/カウンタ部によって共用されるため、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、または本章内後ろで詳述されるようにTOSC1/2ピンから非同期にクロック駆動されます。非同期動作は非同同期状態レジスタ(ASSR)によって制御されます。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を制御します。クロック元が選ばれないと、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(clkT0)として参照されます。

2重緩衝化した比較レジスタ(OCR0)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(OC0)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。詳細については63頁の「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(OCF0)も設定(1)します。

### 定義

本文書でのレジスタとビット参照の多くは一般形で書かれます。小文字の'n'はタイマ/カウンタ番号、この場合は0で置き換えます。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が使われなければなりません(例えばタイマ/カウンタ0のカウンタ値のアクセスに対してのTCNT0のように)。

表51.の定義は本文書を通して広範囲に渡って使われます。

表51. 用語定義

| 用語     | 意味   |
|--------|--|
| BOTTOM | タイマ/カウンタが\$00に到達した時。   |
| MAX    | タイマ/カウンタが\$FF(255)に到達した時。                                    |
| TOP    | タイマ/カウンタが指定された固定値(\$FF)またはOCR0値に到達した時。この指定(TOP)値は動作種別に依存します。 |

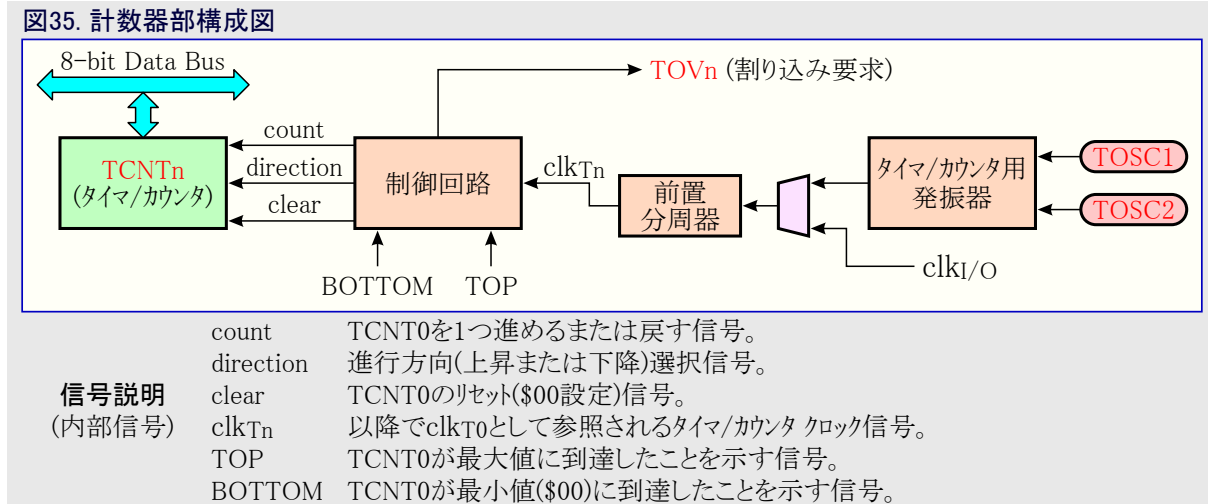


## タイマ/カウンタのクロック

このタイマ/カウンタは内部同期または外部非同期クロック元によってクロック駆動できます。既定のクロック元( $clk_{T0}$ )はMCUクロック( $clk_{I/O}$ )と同じです。非同期状態レジスタ(ASSR)の**非同期動作許可(AS0)ビット**が論理1を書かれると、クロック元はTOSC1とTOSC2に接続されたタイマ/カウンタ用発振器から取得されます。非同期動作の詳細については71頁の「**非同期状態レジスタ(ASSR)**」をご覧ください。クロック元と前置分周器の詳細については73頁の「**タイマ/カウンタの前置分周器**」をご覧ください。

## 計数器部

8ビットタイマ/カウンタの主な部分は設定可能な双方向カウンタ部です。図35.は、このカウンタとその周辺環境の構成図を示します。



使っ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( $clk_{T0}$ )で解除(\$00)、増加(+1)、または減少(-1)されます。 $clk_{T0}$ はクロック選択(CS02~0)ビットによって選ばれた内部または外部のクロック元から生成できます。クロック元が選ばれない(CS02~0=000)時にタイマ/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( $clk_{T0}$ )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(上位優先順位を持ちます)。

計数順序(方法)は**タイマ/カウンタ制御レジスタ(TCCR0)**に配置された**波形生成種別(WGM01,0)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OC0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関しては65頁の「**動作種別**」をご覧ください。

**タイマ/カウンタ溢れ(TOV0)フラグ**はWGM01,0ビットによって選ば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。

## 比較出力部

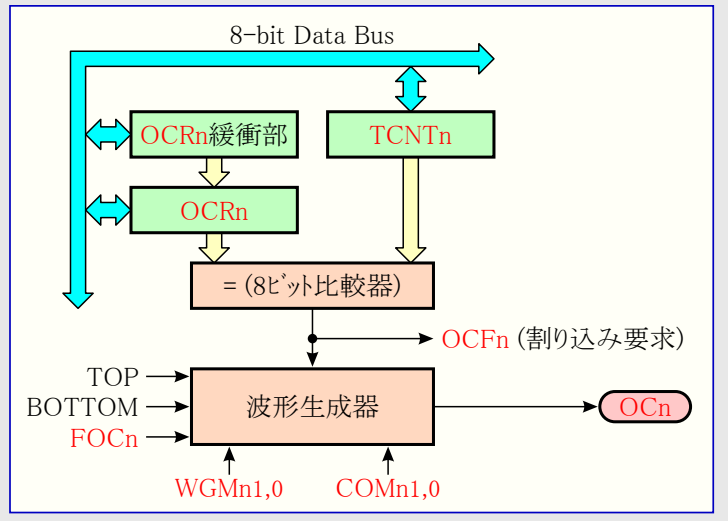
この8ビット比較器はTCNT0と比較レジスタ(OCR0)を継続的に比較します。TCNT0とOCR0が等しければ比較器は一致を指示します。この一致は次のタイマ/カウンタクロック周期で比較割り込み要求フラグ(OCF0)を設定(1)します。許可(I=1, OCIE0=1)なら、この比較割り込み要求フラグは比較割り込みを発生します。OCF0は割り込みが実行されると自動的に解除(0)されます。代わりにOCF0はこのビット位置に論理1を書くことによってソフトウェアでも解除(0)できます。波形生成器は波形生成種別(WGM01,0)ビットと比較出力選択(COM01,0)ビットによって設定された動作種別に従った出力を生成するのにこの一致信号を使います。MAXとBOTTOM信号は動作種別(65頁の「動作種別」参照)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。

図36.は比較出力部の構成図を示します。

OCR0はパルス幅変調(PWM)の何れかを使う時に2重緩衝化されます。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OCR0レジスタ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。

OCR0のアクセスは複雑なように思えますが決してそんなことはありません。2重緩衝動作が許可されるとCPUはOCR0緩衝部をアクセスし、禁止されるとOCR0レジスタを直接アクセスします。

図36. 比較出力部構成図



## 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(FOC0)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(OCF0)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、OC0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(COM01,0ビット設定がOC0ピンの設定(1)、解除(0)、1/0交互のどれかを定義)。

## TCNT0書き込みによる比較一致妨害

TCNT0への全てのCPU書き込みは、例えタイマ/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OCR0に許します。

## 比較一致部の使用

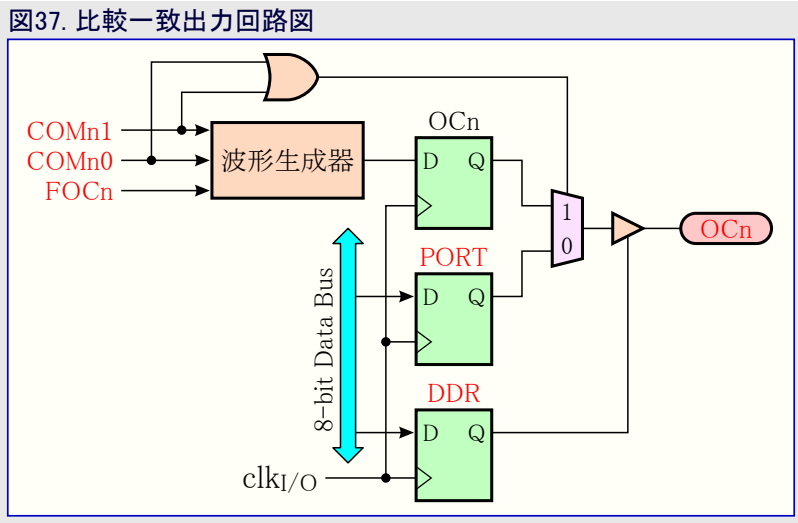
どの動作種別でのTCNT0書き込みでも1タイマ/カウンタクロック周期間、全ての比較一致を妨げるため、タイマ/カウンタが走行中であるか否かに拘らず比較出力部を使う場合、TCNT0を変更する時に危険を伴います。TCNT0に書かれた値がOCR0値と同じ場合、比較一致は失われ(一致が発生せず)、不正な波形生成に終わります。同様にタイマ/カウンタが下降計数のとき、BOTTOMに等しいTCNT0値を書いてはいけません。

OC0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。OC0値を設定する一番簡単な方法は標準動作で強制変更(FOC0)ストローブビットを使うことです。波形生成動作種別間を変更する時であっても、OC0(内部)レジスタはその値を保ちます。

比較出力選択(COM01,0)ビットが比較値(OCR0)と共に2重緩衝されないことに気付いてください。COM01,0ビットの変更は直ちに有効となります。

## 比較一致出力部

比較出力選択(COM01,0)ビットは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(OC0)状態の定義にCOM01,0ビットを使います。またCOM01,0ビットはOC0ピン出力元を制御します。図37はCOM01,0ビット設定によって影響される論理回路の単純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。COM01,0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(PORTとDDR)の部分だけが示されます。OC0の状態を参照するとき、その参照はOC0ピンでなく内部OC0レジスタに対してです。



COM01,0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(OC0)によって無効にされます。けれどもOC0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。OC0ピンに対するポート方向レジスタのビット(DDR\_OC0)はOC0値がピンで見えるのに先立って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OC0状態の初期化を許します。いくつかのCOM01,0ビット設定が或る種の動作種別に対して予約されることに注意してください。69頁の「8ビットタイマ/カウンタ0用レジスタ」をご覧ください。

### 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COM01,0ビットを違うふうに使います。全動作種別に対してCOM01,0=00設定は次の比較一致で実行すべきOC0レジスタの動きがないことを波形生成器へ告げます。非PWM動作での比較出力動作については69頁の表53を参照してください。高速PWM動作については69頁の表54、位相基準PWMについては69頁の表55を参照してください。

COM01,0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(FOC0)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。

## 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(WGM01,0)ビット**と**比較出力選択(COM01,0)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。COM01,0ビットは生成されるPWM出力が反転されるべきか、されないべきか(反転または非反転PWM)どちらかを制御します。非PWM動作に対するCOM01,0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(64頁の「**比較一致出力部**」をご覧ください)。

タイミング情報の詳細については68頁の「**タイマ/カウンタのタイミング**」を参照してください。

## 標準動作

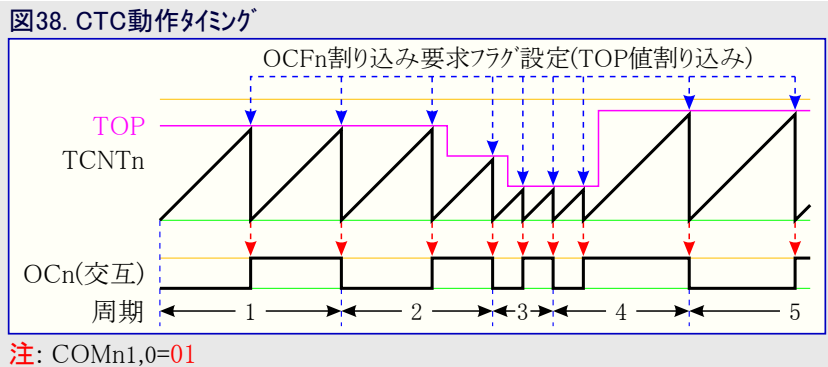
最も単純な動作種別が標準動作(WGM01,0=00)です。本動作種別での計数方向は常に上昇(+)で、カウンタ解除は実行されません。カウンタは8ビット最大値(TOP=\$FF)を通過すると単に範囲を超え、そして\$00(BOTTOM)から再び始めます。通常動作での**タイマ/カウンタ溢れ(TOV0)フラグ**はTCNT0が\$00になる時と同じタイマ/カウンタクロック周期で設定(1)されます。この場合のTOV0フラグは設定(1)のみで解除(0)されないことを除いて第9ビットのようになります。けれどもTOV0フラグを自動的に解除(0)するタイマ/カウンタ溢れ割り込みと組み合わせたタイマ/カウンタの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。

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。

## 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(WGM01,0=10)ではOCR0がカウンタ分解能を操作するのに使われます。CTC動作ではカウンタ(TCNT0)値がOCR0と一致すると、カウンタは\$00に解除されます。OCR0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図38.で示されます。カウンタ(TCNT0)値はTCNT0とOCR0間で比較一致が起こるまで増加し、そしてその後にカウンタ(TCNT0)は解除(\$00)されます。



OCF0フラグを使うことにより、タイマ/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。けれども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BOTTOMと近い値にTOPを変更することは、CTC動作が2重緩衝機能を持たないために注意して行わなければなりません。OCR0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、タイマ/カウンタは(その回の)比較一致を失います。その後のカウンタは比較一致が起こるのに先立って、最大値(\$FF)へそして次に\$00から始める計数をしなければならないでしょう。

CTC動作で波形出力を生成するため、OC0出力は**比較出力選択(COM01,0)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り替えに設定できます。OC0値はそのピンに対するデータ方向が出力(DDR\_OC0=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OCR0が0(\$00)に設定される時に $f_{OC0} = f_{clk\_I/O} / 2$ 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。

$$f_{OCn} = \frac{f_{clk\_I/O}}{2 \times N \times (1 + OCRn)}$$

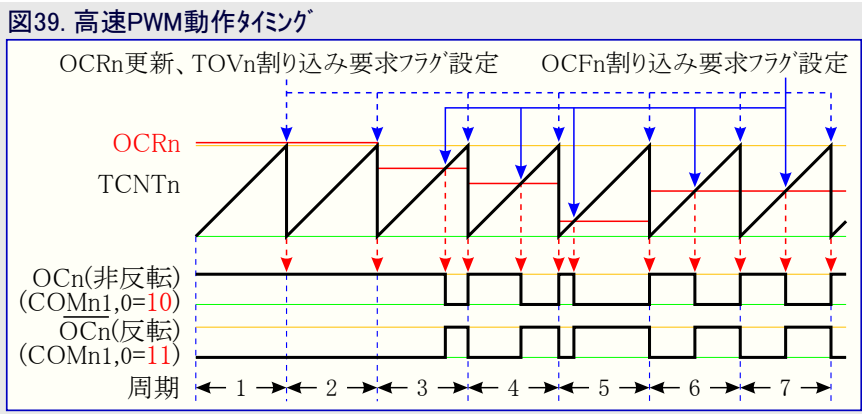
変数Nは前置分周数(1,8,32,64,128,256,1024)を表します。

標準動作と同じように、**タイマ/カウンタ溢れ(TOV0)フラグ**はカウンタがMAXから\$00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。

## 高速PWM動作

高速パルス幅変調(PWM)動作(WGM01,0=11)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MAXまで計数し、その後にBOTTOMから再び始めます。非反転比較出力動作(COM01,0=10)での比較出力(OC0)は、TCNT0とOCR0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(COM01,0=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動作でのタイマ/カウンタはタイマ/カウンタ値がMAX値と一致するまで増加されます。そしてタイマ/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(\$00)されます。高速PWM動作のタイミング図は図39.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。この図は非反転と反転のPWM出力を含みます。赤細線はOCR0値を示し、TCNT0値との交点(接点)がTCNT0とOCR0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(OCF0)はOCR0=MAXを除いて比較一致が起こると設定(1)されます(訳注:共通性のため本行追加)。



タイマ/カウンタ溢れ(TOV0)フラグはカウンタがMAXに到達する時毎に設定(1)されます。割り込みが許可されるなら、その割り込み処理ルーチンは比較値を更新するのに使えます。

高速PWM動作での比較部はOC0ピンでのPWM波形の生成を許します。COM01,0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COM01,0を'11'に設定することで生成できます(69頁の表54.をご覧ください)。実際のOC0値はポートピンに対するデータ方向(DDR\_OC0)が出力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OCR0間の比較一致でOC0(内部)レジスタを設定(1)(または解除(0))と、カウンタが解除(\$00、MAXからBOTTOMへ変更)されるタイマ/カウンタクロック周期でOC0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。

$$f_{OCnPWM} = \frac{f_{clk\_I/O}}{N \times 256}$$

変数Nは前置分周数(1,8,32,64,128,256,1024)を表します。

OCR0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あたります。OCR0がBOTTOM(\$00)と等しく設定されると、出力はMAX+1 タイマ/カウンタクロック周期毎の狭いスパイク(パルス)になるでしょう。OCR0がMAXに等しく設定されると、(COM01,0ビットによって設定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。

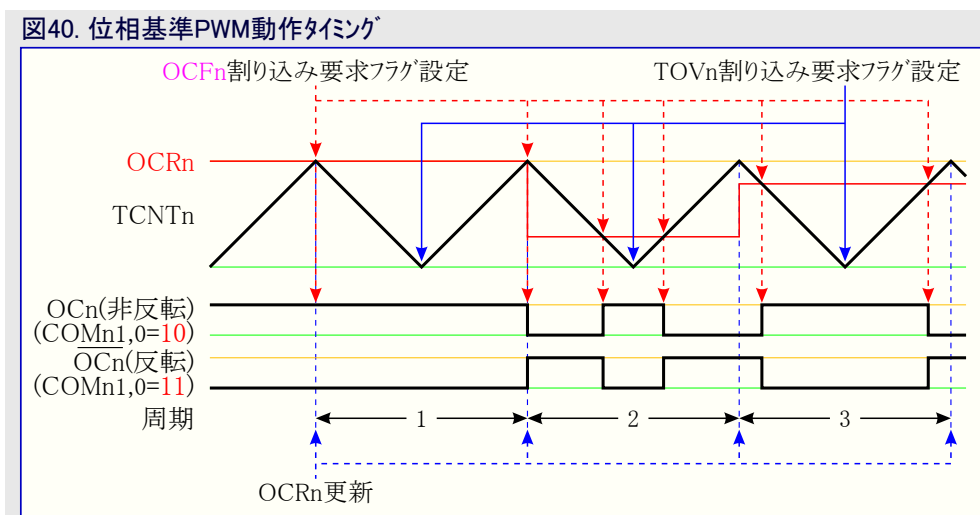
(訳注:以下は本動作種別がTOP=MAX(\$FF)で固定のため基本的には不適切です。) 高速PWM動作で(デューティ比50%)周波数の波形出力は比較一致毎に論理反転するOC0設定(COM01,0=01)によって達成できます。生成された波形はOCR0が0(\$00)に設定される時に $f_{OC0} = f_{clk\_I/O} / 2$ の最大周波数でしょう。この特性は高速PWM動作で比較出力部の2重緩衝機能が許可されることを除いて、CTC動作でのOC0交互出力(COM01,0=01)と同じです。



## 位相基準PWM動作

位相基準パルス幅変調(PWM)動作(WGM01,0=01)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)動作に基づきます。カウンタはBOTTOMからMAXへそして次にMAXからBOTTOMへを繰り返し計数します。非反転比較出力動作(COM01,0=10)での比較出力(OC0)は上昇計数中のTCNT0とOCR0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(COM01,0=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です。けれども両傾斜(三角波)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ビットに固定されます。位相基準PWM動作でのカウンタはカウンタ値がMAXと一致するまで増加されます。カウンタはMAXに到達すると、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間MAXと等しくなります。位相基準PWM動作のタイミング図は図40.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。赤細線はOCR0値を示し、TCNT0値との交点(接点)がTCNT0とOCR0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)。



タイマ/カウンタ溢れ(TOV0)フラグはタイマ/カウンタがBOTTOMに到達する時毎に設定(1)されます。この割り込み要求フラグはカウンタがBOTTOM値に到達する毎に割り込みを発生するのに使えます。

位相基準PWM動作での比較部はOC0ピンでのPWM波形の生成を許します。COM01,0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COM01,0ビットを'11'に設定することで生成できます(69頁の表55.をご覧ください)。実際のOC0値はそのポートピンに対するデータ方向(DDR\_OC0)が出力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OCR0間の比較一致でOC0(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OCR0間の比較一致でOC0レジスタを解除(0)(または設定(1))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。

$$f_{OCnPCPWM} = \frac{f_{clk\_I/O}}{N \times 510}$$

変数Nは前置分周数(1,8,32,64,128,256,1024)を表します。

OCR0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OCR0がBOTTOM(\$00)に等しく設定されると出力は定常的なLow、MAX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。

図40.の第2周期のその出発点に於いて、例え比較一致がないとしても、OCnにはHighからLowへの遷移があります。この遷移点はBOTTOMを挟む対称を保証するためです。比較一致なしに遷移を生ずるのは2つの場合です。

- 図40.のようにOCR0はMAXからその値を変更します。OCR0値がMAXのとき、OCnピン値は下降計数での比較一致の結果と同じです(訳補: L→H、直前がHのため、常にH)。BOTTOMを挟む対称を保証するため、(変更直後の)MAX(位置)でのOCn値は上昇計数での比較一致の結果(H→L)と一致しなければなりません。
- タイマ/カウンタがOCR0値よりも高い値から数え始め、そしてその理屈のために比較一致、それ故上昇途中で起こされるであろうOCnの変更を逃します。(訳補: 従って上記同様、MAX位置で(直前がHならば)H→L遷移が生じます。)

## タイマ/カウンタのタイミング

図41.と図42.はタイマ/カウンタ動作についてのタイミング データを含みます。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( $clk_{T0}$ )がクロック許可信号として示されます。図はMAX値近辺の計数の流れを示します。図43.と図44.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。この図は割り込み要求フラグが設定(1)される時を図解します。

以下の図は同期動作でのタイマ/カウンタを示し、従ってタイマ/カウンタ クロック( $clk_{T0}$ )が計数許可信号として示されます。非同期動作では $clk_{I/O}$ がタイマ/カウンタ用発振器(TOSC)クロックによって置換されるべきです。この図は割り込みフラグが設定(1)される時の情報を含みます。図41.は基本的なタイマ/カウンタ動作についてのタイミング データを示します。この図は位相基準PWM動作以外の全ての動作種別でのMAX値近辺の計数の流れを示します。

図41. 前置分周なし(1/1)のタイマ/カウンタ タイミング

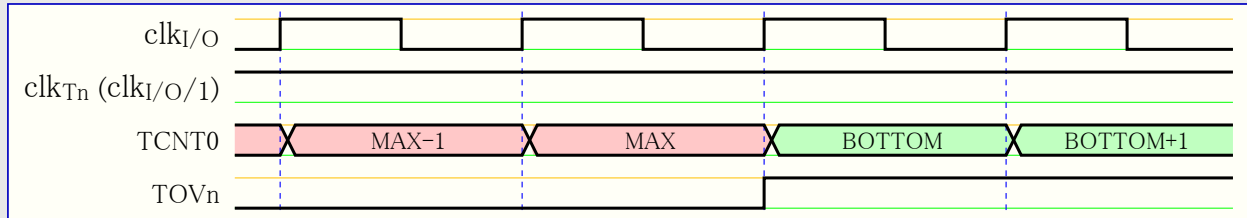


図42.は同じタイミング データを示しますが、前置分周器が許可されています。

図42. 前置分周器( $f_{clk_{I/O}/8}$ )のタイマ/カウンタ タイミング

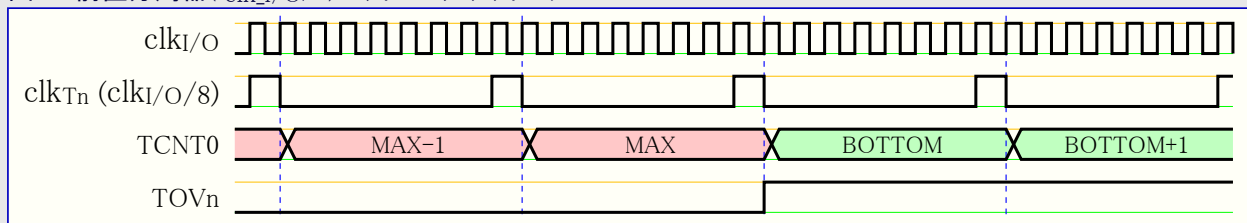


図43.はCTC動作を除く全ての動作種別でのOCF0の設定を示します。

図43. 前置分周器( $f_{clk_{I/O}/8}$ )のタイマ/カウンタ、OCF0設定 タイミング

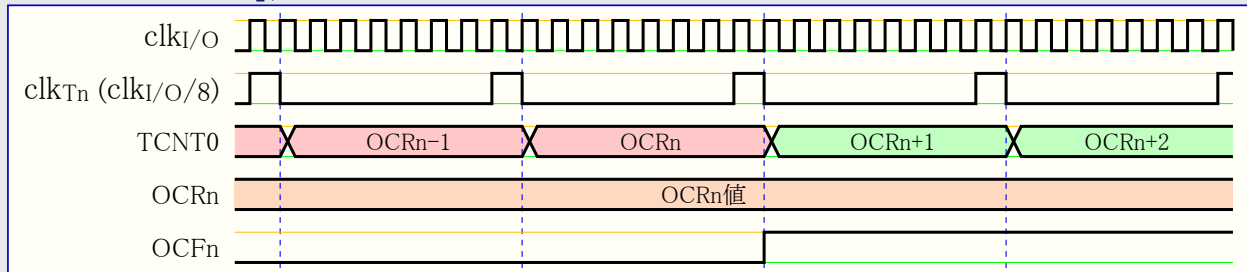
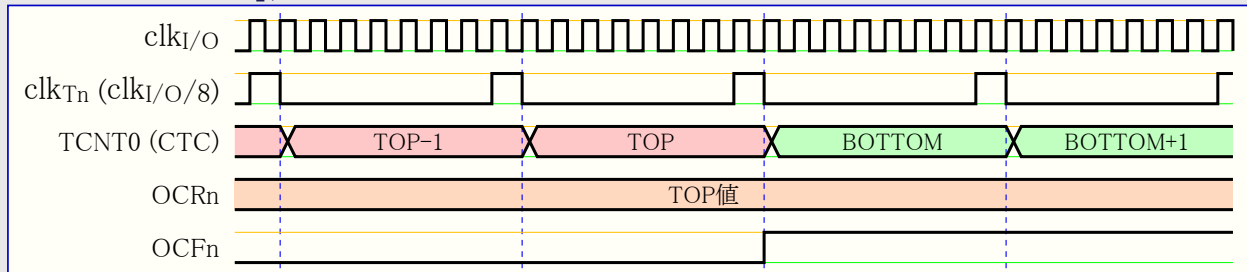


図44.はCTC動作でのTCNT0の解除とOCF0の設定を示します。

図44. 前置分周器( $f_{clk_{I/O}/8}$ )のタイマ/カウンタ、OCF0設定 タイミング



## 8ビット タイマ/カウンタ0用レジスタ

### タイマ/カウンタ0制御レジスタ (Timer/Counter0 Control Register) TCCR0

|             |      |       |       |       |       |      |      |      |       |
|-------------|------|-------|-------|-------|-------|------|------|------|-------|
| ビット         | 7    | 6     | 5     | 4     | 3     | 2    | 1    | 0    |       |
| \$33 (\$53) | FOC0 | WGM00 | COM01 | COM00 | WGM01 | CS02 | CS01 | CS00 | TCCR0 |
| Read/Write  | W    | R/W   | R/W   | R/W   | R/W   | R/W  | R/W  | R/W  |       |
| 初期値         | 0    | 0     | 0     | 0     | 0     | 0    | 0    | 0    |       |

#### •ビット7 – FOC0 : OC0強制変更 (Force Output Compare 0)

FOC0ビットはWGM01,0ビットが非PWM動作を指示する時だけ有効です。けれども将来のデバイスとの共通性を保証するため、PWM動作で扱う時にTCCR0が書かれる場合、このビットは0に設定されなければなりません。FOC0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。OC0出力はCOM01,0ビット設定に従って変更されます。FOC0ビットがストローブとして実行されることに注意してください。従って強制した比較の効果を決めるのはCOM01,0ビットに存在する値です。

FOC0ストローブは何れの割り込みの生成もTOPとしてOCR0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(\$00)も行いません。

FOC0ビットは常に0として読みます。

#### •ビット3,6 – WGM01,WGM00 : 波形生成種別 (Waveform Generation Mode bit 1 and 0)

これらのビットはカウンタの計数順序(方向)、最大(TOP)カウンタ値の供給元、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/カウンタ部により支援される動作種別は標準動作、比較一致タイマ/カウンタ解除(CTC)動作と、2形式のパルス幅変調(PWM)動作です。表52.と65頁の「動作種別」をご覧ください。

表52. 波形生成種別選択

| 番号 | WGM01 (CTC0) | WGM00 (PWM0) | タイマ/カウンタ動作種別          | TOP値 | OCR0 更新時 | TOV0 設定時 |
|----|--------------|--------------|-----------------------|------|----------|----------|
| 0  | 0            | 0            | 標準動作                  | \$FF | 即時       | MAX      |
| 1  | 0            | 1            | 位相基準PWM動作             | \$FF | TOP      | BOTTOM   |
| 2  | 1            | 0            | 比較一致タイマ/カウンタ解除(CTC)動作 | OCR0 | 即時       | MAX      |
| 3  | 1            | 1            | 高速PWM動作               | \$FF | BOTTOM   | MAX      |

注: CTC0とPWM0ビット定義名は旧名です。WGM01,0定義名を使ってください。しかし、これらのビットの機能と位置は旧版のタイマ/カウンタと一致します。

#### •ビット5,4 – COM01,COM00 : 比較0出力選択 (Compare Match 0 Output Mode bit 1 and 0)

これらのビットはOC0比較出力ピンの動作を制御します。COM01,0ビットの1つまたは両方が1を書かれると、OC0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、OC0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。

OC0がピンに接続されるとき、COM01,0ビットの機能はWGM01,0ビット設定に依存します。

表53.はWGM01,0ビットが標準動作またはCTC動作(つまりPWM以外)に設定される時のCOM01,0ビット機能を示します。

表54.はWGM01,0ビットが高速PWM動作に設定される時のCOM01,0ビットの機能を示します。

表55.はWGM01,0ビットが位相基準PWM動作に設定される時のCOM01,0ビットの機能を示します。

表53. 非PWM動作での比較出力選択

| COM01 | COM00 | 意味                   |
|-------|-------|----------------------|
| 0     | 0     | 標準ポート動作 (OC0切断)      |
| 0     | 1     | 比較一致でOC0ピン(交互)出力     |
| 1     | 0     | 比較一致でOC0ピン Lowレベル出力  |
| 1     | 1     | 比較一致でOC0ピン Highレベル出力 |

表54. 高速PWM動作での比較出力選択

| COM01 | COM00 | 意味                                    |
|-------|-------|---------------------------------------|
| 0     | 0     | 標準ポート動作 (OC0切断)                       |
| 0     | 1     | (予約)                                  |
| 1     | 0     | 比較一致でLow、BOTTOMでHighをOC0ピンへ出力 (非反転動作) |
| 1     | 1     | 比較一致でHigh、BOTTOMでLowをOC0ピンへ出力 (反転動作)  |

注: COM01が設定(1)され、OCR0がTOPと等しいとき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)や解除(0)は行われます。より多くの詳細については66頁の「高速PWM動作」をご覧ください。

表55. 位相基準PWM動作での比較出力選択

| COM01 | COM00 | 意味                                      |
|-------|-------|---|
| 0     | 0     | 標準ポート動作 (OC0切断)                         |
| 0     | 1     | (予約)                                    |
| 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OC0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OC0ピンへ出力 |

注: COM01が設定(1)され、OCR0がTOPと等しいとき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、TOPでの設定(1)や解除(0)は行われます。より多くの詳細については67頁の「位相基準PWM動作」をご覧ください。

•ビット2~0 - CS02,CS01,CS00 : クロック選択0 (Clock Select0, bit 2,1 and 0)

この3つのクロック選択ビットはタイマ/カウンタ(TCNT0)によって使われるクロック元を選びます。表56をご覧ください。

表56. 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 <sub>T0S</sub> (前置分周なし)       |
| 0    | 1    | 0    | clk <sub>T0S</sub> /8 (8分周)       |
| 0    | 1    | 1    | clk <sub>T0S</sub> /32 (32分周)     |
| 1    | 0    | 0    | clk <sub>T0S</sub> /64 (64分周)     |
| 1    | 0    | 1    | clk <sub>T0S</sub> /128 (128分周)   |
| 1    | 1    | 0    | clk <sub>T0S</sub> /256 (256分周)   |
| 1    | 1    | 1    | clk <sub>T0S</sub> /1024 (1024分周) |

### タイマ/カウンタ0 (Timer/Counter0) TCNT0

| ビット         | 7     | 6   | 5   | 4   | 3   | 2   | 1   | 0     |       |
|-------------|-------|-----|-----|-----|-----|-----|-----|-------|-------|
| \$32 (\$52) | (MSB) |     |     |     |     |     |     | (LSB) | TCNT0 |
| Read/Write  | R/W   | R/W | R/W | R/W | R/W | R/W | R/W | R/W   |       |
| 初期値         | 0     | 0   | 0   | 0   | 0   | 0   | 0   | 0     |       |

このタイマ/カウンタレジスタは読み書き両方の操作について、タイマ/カウンタ部の8ビットカウンタに直接アクセスします。TCNT0への書き込みは次のタイマ/カウンタクロックでの比較一致を妨害(除去)します。カウンタが走行中にカウンタ(TCNT0)を変更することは、TCNT0とOCR0間の比較一致消失の危険を誘発します。

### タイマ/カウンタ0 比較レジスタ (Timer/Counter0 Output Compare Register) OCR0

| ビット         | 7     | 6   | 5   | 4   | 3   | 2   | 1   | 0     |      |
|-------------|-------|-----|-----|-----|-----|-----|-----|-------|------|
| \$31 (\$51) | (MSB) |     |     |     |     |     |     | (LSB) | OCR0 |
| Read/Write  | R/W   | R/W | R/W | R/W | R/W | R/W | R/W | R/W   |      |
| 初期値         | 0     | 0   | 0   | 0   | 0   | 0   | 0   | 0     |      |

この比較レジスタは継続的にカウンタ(TCNT0)値と比較される8ビットの値を含みます。一致は比較一致割り込みやOC0ピンでの波形出力を生成するのに使えます。

### タイマ/カウンタ割り込み許可レジスタ (Timer/Counter Interrupt Mask Register) TIMSK

| ビット         | 7     | 6     | 5      | 4      | 3      | 2     | 1     | 0     |       |
|-------------|-------|-------|--------|--------|--------|-------|-------|-------|-------|
| \$37 (\$57) | OCIE2 | TOIE2 | TICIE1 | OCIE1A | OCIE1B | TOIE1 | OCIE0 | TOIE0 | TIMSK |
| Read/Write  | R/W   | R/W   | R/W    | R/W    | R/W    | R/W   | R/W   | R/W   |       |
| 初期値         | 0     | 0     | 0      | 0      | 0      | 0     | 0     | 0     |       |

•ビット1 - OCIE0 : タイマ/カウンタ0比較割り込み許可 (Timer/Counter0 Output Compare Match Interrupt Enable)

OCIE0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比較一致割り込みが許可されます。タイマ/カウンタ0で比較一致が起こる(換言すると、タイマ/カウンタ割り込み要求フラグレジスタ(TIFR)で比較0割り込み要求フラグ(OCF0)が設定(1)されると、対応する割り込みが実行されます。

•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)

TOIE0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溢れ割り込みが許可されます。タイマ/カウンタ0溢れが起こる(換言すると、タイマ/カウンタ割り込み要求フラグレジスタ(TIFR)でタイマ/カウンタ0溢れ割り込み要求(TOV0)フラグが設定(1)されると、対応する割り込みが実行されます。

## タイマ/カウンタ割り込み要求フラグ レジスタ (Timer/Counter Interrupt Flag Register) TIFR

| ビット         | 7    | 6    | 5    | 4     | 3     | 2    | 1    | 0    |      |
|-------------|------|------|------|-------|-------|------|------|------|------|
| \$36 (\$56) | OCF2 | TOV2 | ICF1 | OCF1A | OCF1B | TOV1 | OCF0 | TOV0 | TIFR |
| Read/Write  | R/W  | R/W  | R/W  | R/W   | R/W   | R/W  | R/W  | R/W  |      |
| 初期値         | 0    | 0    | 0    | 0     | 0     | 0    | 0    | 0    |      |

### • ビット1 – OCF0 : タイマ/カウンタ0比較割り込み要求フラグ (Timer/Counter0, Output Compare Match Flag)

OCF0ビットは比較一致がタイマ/カウンタ(TCNT0)と比較レジスタ(OCR0)間で起こる時に設定(1)されます。対応する割り込み処理ベクタを実行すると、OCF0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OCF0は解除(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、タイマ/カウンタ割り込み許可レジスタ(TIMSK)のタイマ/カウンタ0比較一致割り込み許可(OCIE0)ビット、OCF0が設定(1)されると、タイマ/カウンタ0比較一致割り込みが実行されます。

### • 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)

TOV0ビットはタイマ/カウンタ(TCNT0)溢れが起こる時に設定(1)されます。対応する割り込み処理ベクタを実行すると、TOV0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TOV0は解除(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、タイマ/カウンタ割り込み許可レジスタ(TIMSK)のタイマ/カウンタ0溢れ割り込み許可(TOIE0)ビット、TOV0が設定(1)されると、タイマ/カウンタ0溢れ割り込みが実行されます。PWM動作ではタイマ/カウンタ0が\$00で計数方向を変える時にこのビットが設定(1)されます。

## タイマ/カウンタ0の非同期動作

### タイマ/カウンタ0非同期状態レジスタ (Timer/Counter0 Asynchronous Status Register) ASSR

| ビット         | 7 | 6 | 5 | 4 | 3   | 2      | 1      | 0      |      |
|-------------|---|---|---|---|-----|--------|--------|--------|------|
| \$30 (\$50) | - | - | - | - | AS0 | TCN0UB | OCR0UB | TCR0UB | ASSR |
| Read/Write  | R | R | R | R | R/W | R      | R      | R      |      |
| 初期値         | 0 | 0 | 0 | 0 | 0   | 0      | 0      | 0      |      |

### • ビット3 – AS0 : タイマ/カウンタ0非同期動作許可 (Asynchronous Timer/Counter0)

AS0が0を書かれると、タイマ/カウンタ0はI/Oクロック( $clk_{I/O}$ )からクロック駆動されます。AS0が1を書かれると、タイマ/カウンタ0はタイマ発振器(TOSC1, TOSC2)ピンに接続されたクリスタル発振器からクロック駆動されます。AS0の値が変更されると、タイマ/カウンタ0(TCNT0)、比較0レジスタ(OCR0)、タイマ/カウンタ0制御レジスタ(TCCR0)の内容は不正にされるかもしれません。

### • ビット2 – TCN0UB : タイマ/カウンタ0更新中フラグ (Timer/Counter0 Update Busy)

タイマ/カウンタ0が非同期に動き、タイマ/カウンタ0(TCNT0)が書かれると、このビットが設定(1)になります。TCNT0が一時保存レジスタから更新されてしまうと、このビットはハードウェアによって解除(0)されます。このビットの論理0はTCNT0が新しい値で更新される用意ができたことを示します。

### • ビット1 – OCR0UB : 比較0レジスタ更新中フラグ (Output Compare Register0 Update Busy)

タイマ/カウンタ0が非同期に動き、比較0レジスタ(OCR0)が書かれると、このビットが設定(1)になります。OCR0が一時保存レジスタから更新されてしまうと、このビットはハードウェアによって解除(0)されます。このビットの論理0は比較0レジスタ(OCR0)が新しい値で更新される用意ができたことを示します。

### • ビット0 – TCR0UB : タイマ/カウンタ0制御レジスタ更新中フラグ (Timer/Counter0 Control Register Update Busy)

タイマ/カウンタ0が非同期に動き、タイマ/カウンタ0制御レジスタ(TCCR0)が書かれると、このビットが設定(1)になります。TCCR0が一時保存レジスタから更新されてしまうと、このビットはハードウェアによって解除(0)されます。このビットの論理0はタイマ/カウンタ0制御レジスタ(TCCR0)が新しい値で更新される用意ができたことを示します。

更新中フラグが設定(1)中に3つのタイマ/カウンタ0レジスタのどれかに書き込みが実行されると、更新された値は不正にされ、予期せぬ割り込みを起こす原因になるかもしれません。

TCNT0、OCR0、TCCR0読み込みについての機構は異なります。TCNT0を読む時は実際のタイマ/カウンタ値が読まれ、OCR0またはTCCR0を読む時は一時保存レジスタの値が読まれます。



## タイマ/カウンタ0 非同期動作

タイマ/カウンタ0が非同期で動くとき、いくつかの考慮が成されなければなりません。

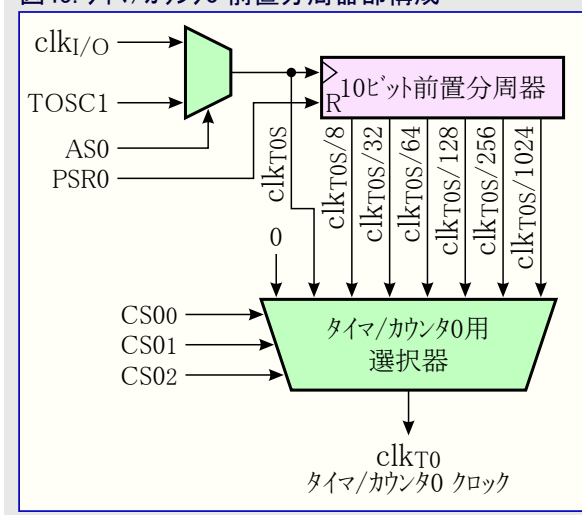
- **警告:** タイマ/カウンタ0の同期/非同期クロック駆動間を切り替えると、**タイマ/カウンタ0(TCNT0)**、**タイマ/カウンタ0比較レジスタ(OCR0)**、**タイマ/カウンタ0制御レジスタ(TCCR0)**が不正となるかもしれません。クロック元を切り替える安全な手順を次に示します。
  1. **タイマ/カウンタ割り込み許可レジスタ(TIMSK)**の**OCIE0**と**TOIE0**の解除(0)により、タイマ/カウンタ0割り込みを禁止します。
  2. **非同期状態レジスタ(ASSR)**の**非同期動作許可(AS0)**設定によってクロック元を適切に選びます。
  3. **TCNT0**, **OCR0**, **TCCR0**に新しい値を書きます。
  4. 非同期動作へ切り替えるには、**TCN0UB**, **OCR0UB**, **TCR0UB**について(=0まで)待機します。
  5. **タイマ/カウンタ割り込み要求フラグ レジスタ(TIFR)**の**OCF0**と**TOV0**フラグを解除(0)します。
  6. 必要とされるなら、割り込みを許可します。
- 発振器は時計用32.768kHzクリスタルで使うために最適化されています。TOSC1ピンに外部クロックを印加することは不正なタイマ/カウンタ0動作に終わるかもしれません。CPU主クロック周波数は、この発振器周波数の4倍よりも高くなければなりません。
- **TCNT0**, **OCR0**, **TCCR0**レジスタの1つに書くとき、その値は一時レジスタへ転送され、TOSC1で2つの上昇端後、(実レジスタ)に設定されます。一時レジスタの内容がそれらの転送先へ転送されてしまう前に新しい値を書くべきではありません。記載された3つのレジスタの各々がそれら個別の一時レジスタを持ちます。それは、例えば**TCNT0**書き込みが**OCR0**書き込みの実行を妨げないことを意味します。転送先レジスタへの転送が起きたことを検知するために**非同期状態レジスタ(ASSR)**は実装されました。
- **TCNT0**, **OCR0**, **TCCR0**に書いた後で**パワーセーブ**または**拡張スタンバイ**動作へ移行するとき、デバイスを起動するのに**タイマ/カウンタ0**が使われる場合、使用者は書かれたレジスタが更新されてしまうまで待たなければなりません。さもなければMCUは変更が有効になる前に**休止形態**へ移行するでしょう。**タイマ/カウンタ0**比較一致割り込みがデバイスを起動するのに使われると、**TCNT0**または**OCR0**書き込み中の比較機能が禁止されるため、これは非常に重要です。書き込み周期が完了されず、**OCR0UB**が0に戻る前にMCUが**休止形態**へ移行すると、デバイスは決して比較一致割り込みを受け取らず、そしてMCUは起動しないでしよう。
- **パワーセーブ**または**拡張スタンバイ**動作からデバイスを起動するのに**タイマ/カウンタ0**が使われる場合、使用者がこれら動作へ(復帰後)再移行することを欲するなら、予防処置を講じなければなりません。割り込み論理回路はリセットするのに1 TOSC1周期必要です。起動と**休止形態**再移行間の時間が1 TOSC1周期未満の場合に割り込みが起きず、デバイスは起動に失敗するでしょう。**パワーセーブ**または**拡張スタンバイ**動作再移行前の時間が充分であるかどうか迷うなら、1 TOSC1周期が経過されることを保証するのに次の手順が使えます。
  1. **TCNT0**, **OCR0**, **TCCR0**に値を書きます。
  2. **非同期状態レジスタ(ASSR)**の対応する更新中フラグが0に戻るまで待ちます。
  3. **パワーセーブ**または**拡張スタンバイ**動作へ移行します。
- **非同期動作が選ばれる**と、**タイマ/カウンタ0**用32.768kHz発振器は**パワーダウン**と**スタンバイ**動作を除いて常に動作します。電源投入リセット、**パワーダウン**または**スタンバイ**動作から起動後、この発振器が安定するのに1秒程度かかるかもしれないことを使用者は承知すべきです。電源投入、**パワーダウン**または**スタンバイ**動作から起動後、**タイマ/カウンタ0**を使う前に少なくとも1秒待つことが推奨されます。この発振器の使用またはクロック信号がTOSCピンに印加されるかのどちらかに拘らず、起動時の不安定なクロック信号のため、**パワーダウン**または**スタンバイ**動作からの起動復帰後、**タイマ/カウンタ0**の全レジスタの内容が失われたと見做されなければなりません。
- **タイマ/カウンタ0**が非同期でクロック駆動される時の**パワーセーブ**または**拡張スタンバイ**動作から起動の説明。割り込み条件が合致すると、**タイマ/カウンタ**クロックの次の周期で起動処理が開始され、プロセッサがカウンタ値を読めるのに先立って**タイマ/カウンタ**は常に最低1、進行されます。起動後MCUは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。
- **パワーセーブ**動作から起動直後の**TCNT0**の読み込みは不正な結果を得るかもしれません。**TCNT0**が非同期TOSCクロックでクロック駆動されるため、**TCNT0**読み込みは内部I/Oクロック領域に同期したレジスタを通して行われなければなりません。同期化はTOSCクロックの全上昇端で行われます。**パワーセーブ**動作から起動し、I/Oクロック(**clk<sub>I/O</sub>**)が再び活性(有効)になると、**TCNT0**はTOSCクロックの次の上昇端まで以前(**休止形態**移行前)の値を読むでしょう。**パワーセーブ**動作から起動後のTOSCクロックの位相は起動時間に依存するため本質的に特定できません。従って**TCNT0**読み込みに対する推奨手順は次のとおりです。
  1. **OCR0**または**TCCR0**のどちらかに何か値を書きます。
  2. **非同期状態レジスタ(ASSR)**の対応する更新中フラグが解除(0)されるまで待ちます。
  3. **TCNT0**を読みます。
- **非同期動作の間中**、**非同期タイマ**用割り込み要求フラグの同期化は3プロセッサ周期+1タイマ周期かかります。従ってプロセッサが割り込み要求フラグ設定の原因となったタイマ値を読めるのに先立って、このタイマは最低1、進行されます。比較出力ピンは**タイマ**クロックで変更され、**プロセッサ**クロックに同期されません。

## タイマ/カウンタ0の前置分周器

タイマ/カウンタ0用クロック元の名前は $clk_{T0S}$ です。既定での $clk_{T0S}$ は主システムI/Oクロック( $clk_{I/O}$ )に接続されます。**タイマ/カウンタ0非同期状態レジスタ(ASSR)の非同期クロック(AS0)ビットの設定(1)**により、タイマ/カウンタ0はTOSC1ピンから非同期にクロック駆動されます。これは実時間計数器(RTC)としてのタイマ/カウンタ0使用を可能にします。AS0が設定(1)されると、TOSC1とTOSC2ピンはポートGから切り離されます。水晶発振子はタイマ/カウンタ0用の独立したクロック元として扱うため、TOSC1とTOSC2ピン間に接続できます。この発振器は32.768kHz水晶発振子で使うために最適化されています。TOSC1に外部クロック元を印加することは推奨されません。

タイマ/カウンタ0に対して可能な前置分周済み選択は $clk_{T0S}/8$ ,  $clk_{T0S}/32$ ,  $clk_{T0S}/64$ ,  $clk_{T0S}/128$ ,  $clk_{T0S}/256$ ,  $clk_{T0S}/1024$ です。加えて0(停止)は勿論 $clk_{T0S}$ も選択可能です。**特殊I/O機能レジスタ(SFIOR)のタイマ/カウンタ0前置分周器リセット(PSR0)ビットの設定(1)**は前置分周器をリセットします。これは予測可能な前置分周器での操作を使用者に許します。

図45. タイマ/カウンタ0 前置分周器部構成



## 特殊I/O機能レジスタ (Special Function I/O Register) SFIOR

| ビット         | 7   | 6 | 5 | 4 | 3    | 2   | 1    | 0      |       |
|-------------|-----|---|---|---|------|-----|------|--------|-------|
| \$20 (\$40) | TSM | - | - | - | ACME | PUD | PSR0 | PSR321 | SFIOR |
| Read/Write  | R/W | R | R | R | R/W  | R/W | R/W  | R/W    |       |
| 初期値         | 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ビットに1を書くことがタイマ/カウンタ同期化動作を活性(有効)にします。この動作でPSR0とPSR321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。これは対応するタイマ/カウンタを停止し、設定中にそれらの1つが進行する危険なしに同じ値に設定できるのを保証します。TSMビットが0を書かれると、PSR0とPSR321ビットはハードウェアによって解除(0)され、同時にタイマ/カウンタが計数を始めます。

### • ビット1 – PSR0 : タイマ/カウンタ0 前置分周器リセット (Prescaler Reset Timer/Counter0)

このビットが1の時にタイマ/カウンタ0の前置分周器はリセットします。通常、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。タイマ/カウンタ0が非同期動作の時にこのビットが(1)を書かれると、このビットは前置分周器がリセットされてしまうまで1に留まります。TSMビットが設定(1)される場合、このビットはハードウェアによって解除(0)されません。

(訳補) SFIORのビット4は元のADHSMが削除されました。この結果、次の項目が必要と思われます。

### • ビット4 – Res : 予約 (Reserved Bit)

このビットは将来の使用に対して予約されています。将来のデバイスとの共通性のため、SFIORが書かれる時にこのビットは0が書かれなければなりません。

## 16ビット タイマ/カウンタ (タイマ/カウンタ1とタイマ/カウンタ3)

この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。主な特徴は次のとおりです。

- 真の16ビット設計 (換言すれば16ビットPWMの許容)
- 3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
- 10種の独立した割り込み (TOV1,OCF1A,OCF1B,OCF1C,ICF1,TOV3,OCF3A,OCF3B,OCF3C,ICF3)

### ATmega103互換動作での制限

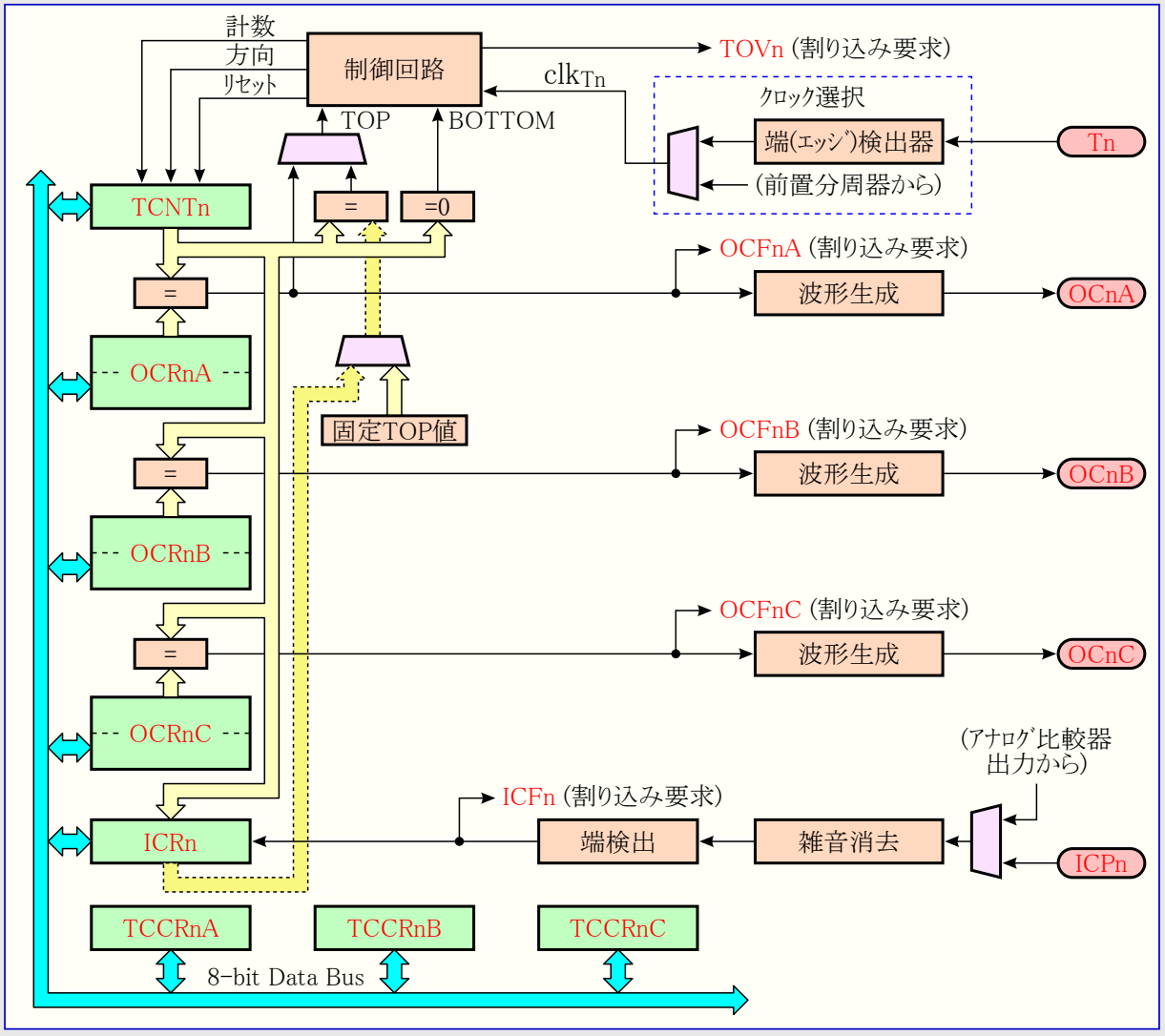
Atmel® AVR® ATmega103互換動作では1つの16ビット タイマ/カウンタ(タイマ/カウンタ1)だけが利用可能なことに注意してください。同様にATmega103互換動作では2つの比較レジスタ(比較Aと比較B)だけであることにも注意してください。

### 概要

本文書でのレジスタとビット参照の多くは一般形で書かれます。小文字の'n'はタイマ/カウンタ番号、小文字の'x'は比較出力部の置き換えです。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が使われなければなりません(例えばタイマ/カウンタ1のカウンタ値に対するアクセスのTCNT1のように)。

この16ビット タイマ/カウンタの簡単化した構成図は図46.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88頁の「16ビット タイマ/カウンタ1,3用レジスタ」で示されます。

図46. 16ビット タイマ/カウンタ構成図



注: タイマ/カウンタ1と3のピン配置と説明については2頁の「ピン配置」、46頁の表30、51頁の表39を参照してください。

## 関係レジスタ

タイマ/カウンタ(TCNTn)、比較レジスタ(OCRnA,OCRnB,OCRnC)、捕獲レジスタ(ICRn)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とき、特別な手順に従わなければなりません。これらの手順は76頁の「16ビットレジスタのアクセス」項で記述されます。タイマ/カウンタ制御レジスタ(TCCRnA,TCCRnB,TCCRnC)は8ビットレジスタで、CPUアクセス制限はありません。割り込み要求信号はタイマ/カウンタ割り込み要求レジスタ(TIFR)と拡張タイマ/カウンタ割り込み要求レジスタ(ETIFR)で全て見えます。全ての割り込みはタイマ/カウンタ割り込み許可レジスタ(TIMSK)と拡張タイマ/カウンタ割り込み許可レジスタ(ETIMSK)で個別に遮蔽(禁止)されます。(E)TIFRと(E)TIMSKはこれらのレジスタが他のタイマ/カウンタ部によって共用されるため、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Tn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端(エッジ)を制御します。クロック元が選ばれないと、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(clkTn)として参照されます。

2重緩衝化した比較レジスタ(OCRnA,OCRnB,OCRnC)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(OCnA,OCnB,OCnC)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。80頁の「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(OCFnA,OCFnB,OCFnC)も設定(1)します。

捕獲レジスタ(ICRn)は捕獲起動(ICPn)ピンまたはアナログ比較器出力(152頁の「アナログ比較器」参照)のどちらかの外部(端で起動された)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部(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、OCRnA、ICRn、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OCRnAを使うと、OCRnAはPWM出力生成用に使えません。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、ICRnが代わりに使え、PWM出力として使われるべきOCRnAを開放します。

## 定義

次の定義は本文書を通して広範囲に使われます。

表57. 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)、OCRnA値、ICRn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。 |

## 互換性

この16ビットタイマ/カウンタは旧版の16ビットAVRタイマ/カウンタから改良更新されてしまっています。この16ビットタイマ/カウンタは次の点に関して以前の版と完全な互換性があります。

- タイマ/カウンタ割り込み関係レジスタを含め、16ビットタイマ/カウンタに関連する全てのI/Oレジスタアドレスの位置。
- タイマ/カウンタ割り込み関係レジスタを含め、16ビットタイマ/カウンタに関連する全てのI/Oレジスタ内のビット位置。
- 割り込みベクタ。

次の制御ビットは変更された名前を持ちますが、同じ機能とレジスタ位置を持ちます。

- PWMn0はWGMn0に変更。
- PWMn1はWGMn1に変更。
- CTCnはWGMn2に変更。

次のレジスタが16ビットタイマ/カウンタに追加されます。

- タイマ/カウンタ制御レジスタC(TCCRnC)
- OCRnCHとOCRnCLを組み合わせた比較nCレジスタ(OCRnC)

次の制御ビットが16ビットタイマ/カウンタ制御レジスタに追加されます。

- COMn1,0がTCCRnAに追加。
- FOCnA, FOCnB, FOCnCが新規TCCRnCに追加。
- WGMn3がTCCRnBに追加。

この比較C出力用の割り込み要求フラグと割り込み許可ビットが追加されます。

この16ビットタイマ/カウンタにはいくつかの特別な状況で互換性に影響を及ぼす改良点があります。



## 16ビットレジスタのアクセス

TCNTn, OCRnA, OCRnB, OCRnC, ICRnは8ビットバス経由で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アクセスされなければなりません。各々の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、この同一一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。16ビットレジスタの下位バイトがCPUによって書かれると、一時レジスタに保存した上位バイトと書かれた下位バイトは同じクロック周期で両方が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一時レジスタに複写されます。

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。OCRnA, OCRnB, OCRnC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ビット書き込みを行うために上位バイトは下位バイトに先立って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。

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。OCRnA, OCRnB, OCRnC, ICRn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言語を使う時はコンパイラが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

```

~
LDI    R17, $01           ;
LDI    R16, $FF          ;[16ビット($01FF)書き込み]
OUT    TCNTnH, R17       ;$01FFの上位バイト値取得
OUT    TCNTnL, R16       ;$01FFの下位バイト値取得
~                               ;上位バイト設定(一時レジスタ)
~                               ;下位バイト設定(一時レジスタ⇒上位バイト)
IN     R16, TCNTnL       ;[16ビット読み込み]
IN     R17, TCNTnH       ;下位バイト取得(上位バイト⇒一時レジスタ)
~                               ;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;
~
TCNTn = 0x1FF;           /* */
i = TCNTn;               /* 16ビット($01FF)書き込み */
~                         /* 16ビット読み込み */
~                         /* */

```

**注:** 6頁の「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TCNTn値を戻します。

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TCNTnレジスタ内容の非分断読み込み法を示します。同じ原理を使うことにより、OCRnA,OCRnB,OCRnC,ICRnのどの読み込みも行えます。

#### アセンブリ言語プログラム例

```
RD_TCNTn:   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, TCNTnL       ; TCNTn下位バイト取得(上位バイト⇒一時レジスタ)
            IN      R17, TCNTnH       ; TCNTn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_TCNTn(void)
{
    unsigned char sreg;                /* ステータスレジスタ一時保存変数定義 */
    unsigned int i;                   /* TCNTn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 */
    __disable_interrupt();            /* 全割り込み禁止 */
    i = TCNTn;                         /* TCNTn値を取得 */
    SREG = sreg;                      /* 全割り込み許可フラグ(I)を復帰 */
    return i;                         /* TCNTn値で呼び出し元へ復帰 */
}
```

注: 6頁の「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TCNTn値を戻します。

次のコード例はTCNTn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、OCRnA,OCRnB,OCRnC,ICRn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_TCNTn:   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     TCNTnH, R17       ; TCNTn上位バイト設定(一時レジスタ)
            OUT     TCNTnL, R16       ; TCNTn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_TCNTn(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;                   /* TCNTn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 */
    __disable_interrupt();            /* 全割り込み禁止 */
    TCNTn = i;                        /* TCNTn値を設定 */
    SREG = sreg;                      /* 全割り込み許可フラグ(I)を復帰 */
}
```

注: 6頁の「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TCNTnへ書かれるべき値を含むことが必要です。

### 上位バイト一時レジスタの再使用

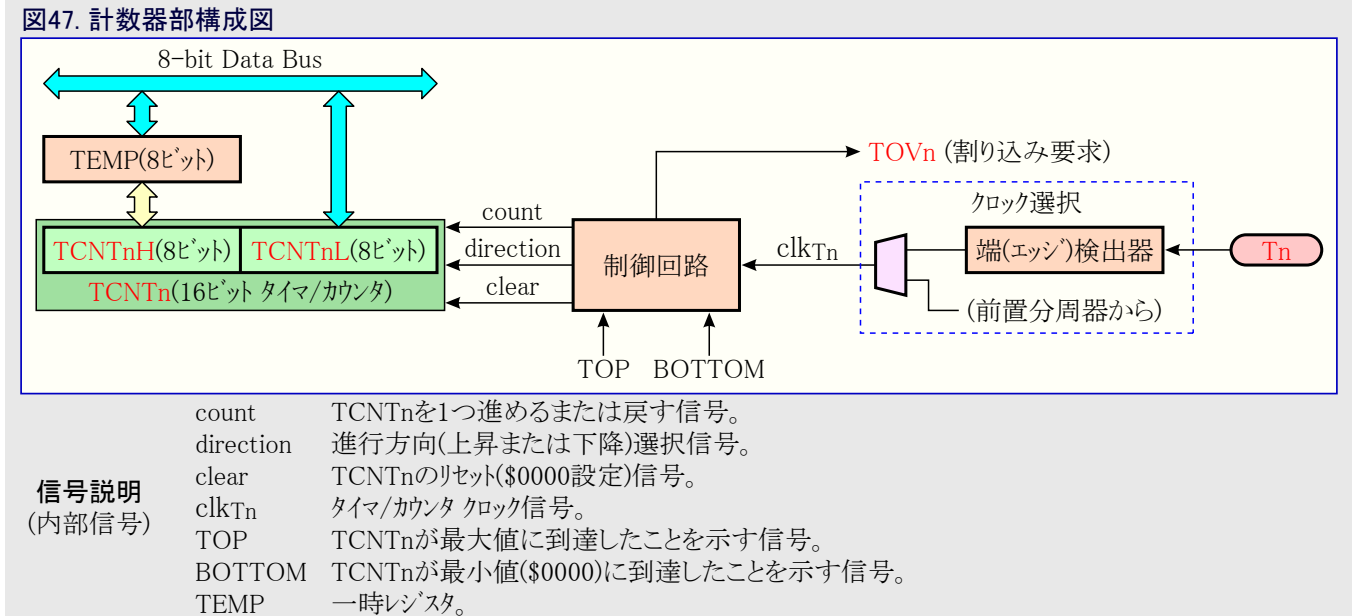
書かれる全レジスタについて上位バイトが同じ複数16ビットレジスタ書き込みなら、上位バイトは1度書かれることだけが必要です。けれども直前で記述した非分断操作の同じ規則が、この場合にも適用されることに注意してください。

## タイマ/カウンタのクロック

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(TCCRnB)**に配置された**クロック選択(CSn2~0)**ビットによって制御されるクロック選択論理回路で選ばれます。クロック元と前置分周器の詳細については97頁の「**タイマ/カウンタ1、タイマ/カウンタ2、タイマ/カウンタ3の前置分周器**」をご覧ください。

## 計数器部

16ビットタイマ/カウンタの主な部分は設定可能な16ビット双方向カウンタ部です。図47はこのカウンタとその周辺の構成図を示します。



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(TCNTnH)と下位8ビットを含むカウンタ下位(TCNTnL)の2つの8ビットI/Oメモリ位置に配置されます。TCNTnH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TCNTnH I/O位置をアクセスするとき、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TCNTnLが読まれる時にTCNTnH値で更新され、TCNTnLが書かれる時にTCNTnH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クロック周期内の16ビットカウンタ値全体の読み書きをCPUに許します。予測不能な結果を生じる、カウンタが計数中の時のTCNTn書き込みの特別な場合に注意することが重要です。この特別な場合はそれらが重要となる項目で記述されます。

使っ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(clkTn)でリセット(\$0000)、増加(+1)、または減少(-1)されます。clkTnはクロック選択(CSn2~0)ビットによって選ばれた内部または外部のクロック元から生成できます。クロック元が選ばれない(CSn2~0=000)時にカウンタは停止されます。けれどもTCNTn値はタイマ/カウンタクロック(clkTn)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(上位優先権を持ちます)。

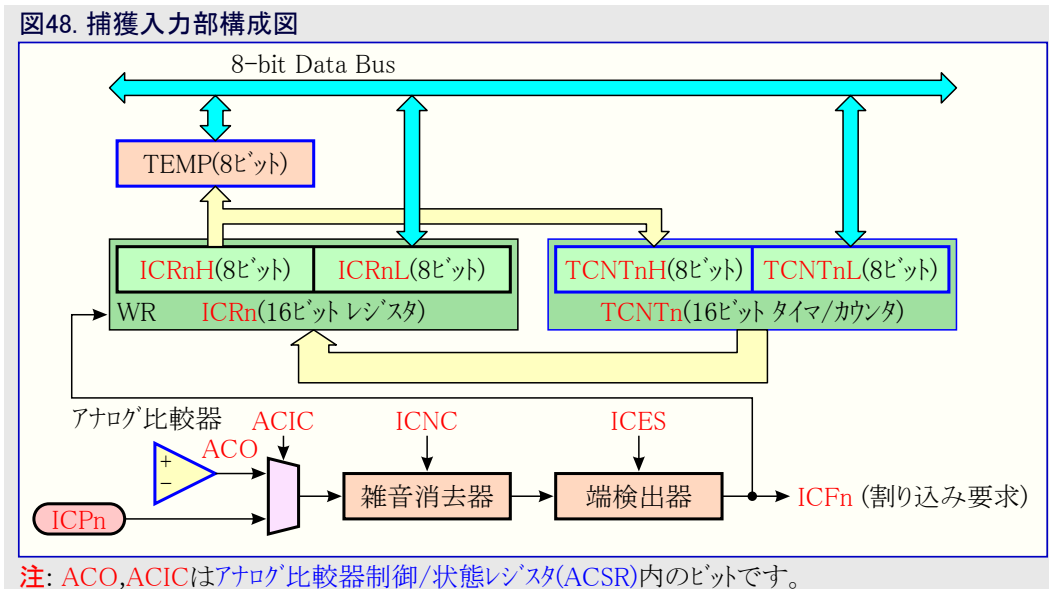
計数順序(方法)は**タイマ/カウンタ制御レジスタA(TCCRnA)**と**タイマ/カウンタ制御レジスタB(TCCRnB)**に配置された**波形生成種別(WGMn3~0)**ビットの設定によって決定されます。これらはカウンタ動作(計数)方法と波形がOCnx比較出力に生成される方法間の接続に近いものです。進化した計数順序と波形生成についてより多くの詳細に対しては83頁の「**動作種別**」をご覧ください。

**タイマ/カウンタ溢れ(TOVn)フラグ**はWGMn3~0ビットによって選ばれた動作種別に従って設定(1)されます。TOVnはCPU割り込み発生に使えます。

## 捕獲入力部

タイマ/カウンタ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合体します。出来事または複数の出来事を示す外部信号はICPnピンまたは代わりにアナログ比較器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。

捕獲入力部は図48.で示される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。



注: アナログ比較器出力(ACO)はタイマ/カウンタ1のみ起動でき、タイマ/カウンタ3では使えません。

捕獲起動入力(ICPn)ピン若しくは代わりにアナログ比較器出力(ACO)で論理レベルの変化(出来事)が起き、その変化が端(エッジ)検出器の設定を追認すると、捕獲が起動されます。捕獲が起動されると、カウンタ(TCNTn)の16ビット値が捕獲レジスタ(ICRn)に書かれます。捕獲割り込み要求フラグ(ICFn)はTCNTn値がICRnに複写されるのと同じシステムクロックで設定(1)されます。許可(I=1, TICIE<sub>n</sub>=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。ICFn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(ICRn)の16ビット値読み込みは、初めに下位8ビット(ICRnL)、その後に上位8ビット(ICRnH)を読むことによって行われます。下位8ビットが読まれる時に上位8ビットが上位8ビット一時レジスタ(TEMP)に複写されます。CPUがICRnH I/O位置を読むと、この一時レジスタをアクセスします。

ICRnはカウンタのTOP値定義にICRn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ICRnに書かれ得る前に波形生成種別(WGM<sub>n</sub>3~0)ビットが設定されなければなりません。ICRnに書く時は下位8ビットがICRnLに書かれる前に、上位8ビットがICRnH I/O位置に書かれなければなりません。

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76頁の「16ビットレジスタのアクセス」を参照してください。

## 捕獲起動元

捕獲入力部用の主な起動元は捕獲起動入力(ICPn)ピンです。タイマ/カウンタ1は捕獲入力部用起動元としてアナログ比較器出力を代わりに使えます。アナログ比較器はアナログ比較器制御/状態レジスタ(ACSR)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ばれます。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(ICFn)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(ICPn)ピンとアナログ比較器出力(ACO)の両入力は、Tnピン(97頁の図59.参照)についてと同じ技法を使って採取されます。端検出器も全く同じです。けれども雑音除去が許可されると、付加論理回路が端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やします。タイマ/カウンタがTOP値定義にICRnを使う波形生成種別に設定されないなら、雑音除去器と端検出器の入力が常に許可されることに注意してください。

捕獲入力ICPn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。

## 雑音除去器

雑音除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。雑音除去器の入力は4採取に渡って監視され、端検出器によって使われる方向転換となる出力を変更するためには4回全てが同じでなければなりません。

雑音除去器はタイマ/カウンタ制御レジスタB(TCCRnB)の捕獲入力雑音除去許可(ICNC<sub>n</sub>)ビットの設定(1)によって許可されます。許可したとき、雑音除去器は入力に印加した変更からICRnの更新までに4システムクロック周期の追加遅延をもたらします。雑音除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されません。

## 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して、次の出来事が起こる前に捕獲した**捕獲レジスタ(ICRn)**の値をプロセッサが読めなかった場合、ICRn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ICRn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。

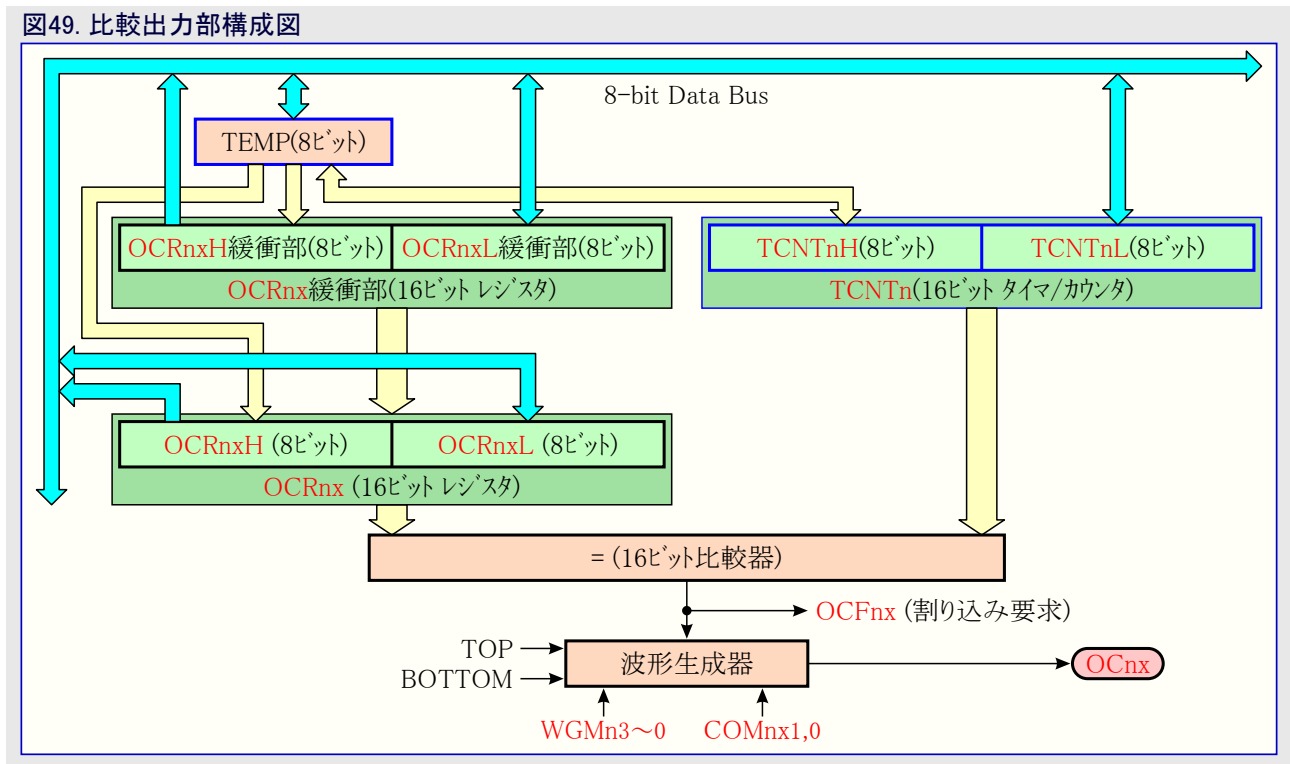
外部信号のデューティ比測定は各捕獲後に起動端が変更されることを必要とします。検出端の変更はICRnが読まれてしまった後に可能な限り早く行われなければなりません。端(エッジ)の変更後、**捕獲割り込み要求フラグ(ICFn)**はソフトウェア(I/Oビット位置への論理1書き込み)によって解除(0)されなければなりません(訳補:端変更によってICFnが設定(1)されることを想定)。周波数のみの測定について(割り込み処理が使われる場合)、ICFnの解除(0)は必要とされません。

## 比較出力部

この16ビット比較器は**TCNTn**と**比較レジスタ(OCRnx)**を継続的に比較します。TCNTnとOCRnx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(OCFn)**を設定(1)します。許可(I=1, OCIE<sub>n</sub>=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。OCFn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OCFn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(WGMn3~0)ビット**と**比較出力選択(COMnx1,0)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(83頁の「動作種別」参照)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。

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。

図49.は比較出力部の構成図を示します。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(タイマ/カウンタ1に対してはn=1)、小文字の'x'は比較出力部(A、BまたはC)を表します。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。



OCRnxは12種類のパルス幅変調(PWM)のどれかをう時に2重緩衝化されます。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OCRnx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。

OCRnxのアクセスは複雑なように思えますが決してそんなことはありません。2重緩衝動作が許可されるとCPUはOCRnx緩衝部をアクセスし、禁止されるとOCRnxレジスタを直接アクセスします。OCRnx(緩衝部またはレジスタ)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TCNTnやICRnのようにOCRnxを自動的に更新しません)。従ってOCRnx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。OCRnx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(OCRnxH)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(OCRnxL)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OCRnx緩衝部またはOCRnx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76頁の「16ビット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を構成する緩衝部分をOCRnx緩衝部、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。

## 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(FOCnx)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(OCFnx)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、OCnx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(COMnx1,0ビット設定がOCnxピンの設定(1)、解除(0)、1/0交互のどれかを定義)。

## TCNTn書き込みによる比較一致妨害

TCNTnへの全てのCPU書き込みは、例えタイマ/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、TCNTnと同じ値に初期化されることをOCRnxに許します。

## 比較一致部の使用

どの動作種別でのTCNTn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、TCNTnを変更する時に危険を伴います。TCNTnに書かれた値がOCRnx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TCNTn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TCNTn値を書いてはいけません。

OCnx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。OCnx値を設定する一番簡単な方法は標準動作で強制変更(FOCnx)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、OCnx(内部)レジスタはその値を保ちます。

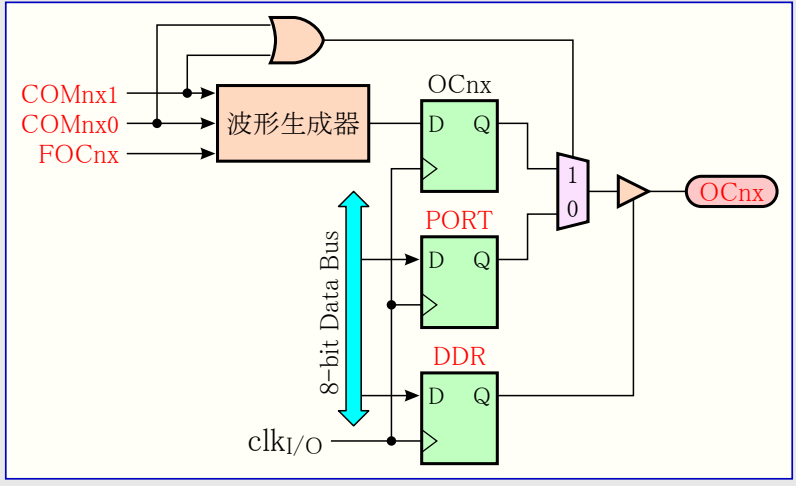
比較出力選択(COMnx1,0)ビットが比較値(OCRnx)と共に2重緩衝されないことに気付いてください。COMnx1,0ビットの変更は直ちに有効となります。



## 比較一致出力部

比較出力選択(COMnx1,0)ビットは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(OCnx)状態の定義にCOMnx1,0ビットを使います。次にCOMnx1,0ビットはOCnxピン出力元を制御します。図50.はCOMnx1,0ビット設定によって影響される論理回路の簡略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。COMnx1,0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(PORTとDDR)の部分だけが示されます。OCnxの状態を参照するとき、その参照はOCnxピンでなく内部OCnxレジスタに対してです。システムリセットが起ると、OCnxレジスタは0にリセットされます。

図50. 比較一致出力回路図



COMnx1,0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(OCnx)によって無効にされます。けれどもOCnx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。OCnxピンに対するポート方向レジスタのビット(DDR\_OCnx)はOCnx値がピンで見えるのに先立って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一般的に波形生成種別と無関係ですが、いくつかの例外があります。詳細については表58、表59、表60を参照してください。

比較出力ピン論理回路の設計は出力が許可される前のOCnx状態の初期化を許します。いくつかのCOMnx1,0ビット設定が或る種の動作種別に対して予約されることに注意してください。88頁の「16ビットタイマ/カウンタ1,3用レジスタ」をご覧ください。

COMnx1,0ビットは捕獲入力部での何の効果もありません。

### 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COMnx1,0ビットを違うふうに使います。全動作種別に対してCOMnx1,0=00設定は次の比較一致で実行すべきOCnxレジスタの動きがないことを波形生成器へ告げます。非PWM動作での比較出力動作については88頁の表58を参照してください。高速PWM動作については88頁の表59、位相基準PWMと位相/周波数基準PWMについては88頁の表60を参照してください。

COMnx1,0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(FOCnx)ストップ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。

## 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(WGMn3~0)ビット**と**比較出力選択(COMnx1,0)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。COMnx1,0ビットは生成されたPWM出力が反転されるべきか、されないべきか(反転または非反転PWM)どちらかを制御します。非PWM動作に対してのCOMnx1,0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。82頁の「**比較一致出力部**」をご覧ください。

タイミング情報の詳細については87頁の「**タイマ/カウンタ1,3のタイミング**」を参照してください。

## 標準動作

最も単純な動作種別が標準動作(WGMn3~0=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\$0000(BOTTOM)から再び始めます。通常動作での**タイマ/カウンタ溢れ(TOVn)フラグ**はTCNTn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TOVn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TOVn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。

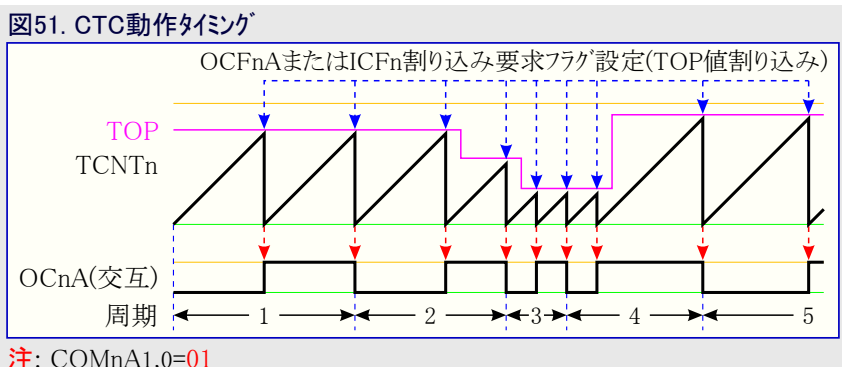
捕獲入力部は標準動作での使用が容易です。けれども外部の事象間の最大間隔がタイマ/カウンタの分解能(16ビット長)を越えてはならないことに気付いてください。事象間の間隔が長すぎる場合、捕獲部に対して分解能を拡張するために、タイマ/カウンタ溢れ割り込みまたは前置分周器が使われなければなりません。

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。

## 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(WGMn3~0=0100または1100)ではOCRnAまたはICRnがカウンタの分解能を操作するのに使われます。CTC動作ではカウンタ(TCNTn)値がOCRnA(WGMn3~0=4)またはICRn(WGMn3~0=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。OCRnAまたはICRn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図51.で示されます。カウンタ(TCNTn)値はOCRnAまたはICRnのどちらかで比較一致が起こるまで増加し、そしてその後にカウンタ(TCNTn)は解除(\$0000)されます。



TOP値を定義するのに使われるレジスタに対してOCFnAまたはICFn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。けれども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BOTTOMと近い値にTOPを変更するのは、CTC動作が2重緩衝機能を持たないために注意して行わなければなりません。OCRnAまたはICRnに書かれた新しい値がTCNTnの現在値よりも低(小さい)場合、カウンタは(その回の)比較一致を失います。その後のカウンタは比較一致が起こせるのに先立って、最大値(\$FFFF)へそして次に\$0000から始める計数をしなければなりません。多くの場合でこの特性は好ましくありません。OCRnAが2重緩衝されるので、代替はTOPを定義するのにOCRnAを用いる**高速PWM動作(WGMn3~0=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、OCnA出力は**比較出力選択(COMnA1,0)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。OCnA値はそのピンに対するデータ方向が出力(DDR\_OCnA=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OCRnAが0(\$0000)に設定される時に $f_{OCnA} = f_{clk\_I/O} / 2$ 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。

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

変数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(TOVn)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。

## 高速PWM動作

高速パルス幅変調(PWM)動作(WGMn3~0=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(COMnx1,0=10)での比較出力(OCnx)はTCNTnとOCRnx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(COMnx1,0=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OCRnAかICRn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(OCRnAまたはICRnが\$0003設定)、最大分解能は16ビット(OCRnAまたはICRnがMAX設定)です。ビットでのPWM分解能は次式を使うことによって計算できます。

高速PWM動作でのカウンタはカウンタ値が固定値\$00FF,\$01FF,\$03FF(WGMn3~0=0101,0110,0111)、ICRn値(WGMn3~0=1110)またはOCRnA値(WGMn3~0=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。高速PWM動作のタイミング図は図52.

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

で示されます。本図はOCRnAかICRnがTOPを定義するのに使われる時の高速PWM動作を示します。TCNTn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OCRnx値を示し、TCNTn値との交点(接点)がTCNTnとOCRnx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(OCFnx)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(TOVn)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、OCRnAかICRnのどちらかがTOP値を定義するのに使われると、OCFnAまたはICFn割り込み要求フラグはTOVn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値を変更するとき、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、TCNTnとそのOCRnx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OCRnx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。

ICRnがTOP値を定義するのに使われるとき、ICRnを更新する手順はOCRnAの更新と異なります。ICRn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ICRnが小さな値に変更される場合、書かれた新しいICRn値がTCNTnの現在値よりも小さくなる危険を意味します。その結果はカウンタが(その回の)TOP値での比較一致を失うことです。その後のカウンタは比較一致が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りません。けれども、OCRnAは2重緩衝されます。この特徴は何時でも書かれることをOCRnAのI/O位置に許します。OCRnA I/O位置が書かれると、書かれた値はOCRnA緩衝部に置かれます。OCRnA(比較レジスタ)はその後にTCNTnがTOPと一致した次のタイマ/カウンタクロック周期にOCRnA緩衝部の値で更新されます。この更新はTCNTnの解除(\$0000)やTOVn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ICRnを使うことは決まったTOP値を使う時に上手いきます。ICRnを使うことにより、OCnAでのPWM出力を生成するためにOCRnA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、OCRnAが2重緩衝機能のため、TOPとしてOCRnA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OCnxピンでのPWM波形の生成を許します。COMnx1,0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COMnx1,0を'11'に設定することで生成できます。88頁の表59をご覧ください。実際のOCnx値はそのポートピンに対するデータ方向が出力(DDR\_OCnx=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TCNTnとOCRnx間の比較一致でOCnx(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OCnx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。

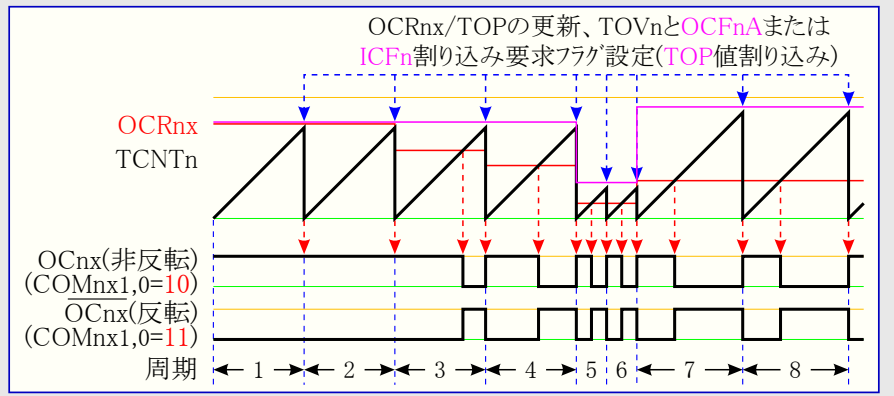
PWM出力周波数は次式によって計算できます。変数Nは前置分周数(1,8,64,256,1024)を表します。

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

OCRnx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。OCRnx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スパイク(パルス)になるでしょう。TOPに等しいOCRnx設定は(COMnx1,0ビットによって設定される出力極性に依存して)定期的なLowまたはHigh出力に終わるでしょう。

高速PWM動作で(デューティ比50%)周波数の波形出力は比較一致毎に論理反転するOCnA設定(COMnA1,0=01)によって達成できます。これはTOP値を定義するのにOCRnAが使われる(WGMn3~0=1111)の場合にだけ適用されます。生成された波形はOCRnAが0(\$0000)に設定される時に $f_{OCnA} = f_{clk\_I/O} / 2$ の最大周波数でしょう。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OCnA交互出力(COMnA1,0=01)と同じです。

図52. 高速PWM動作タイミング





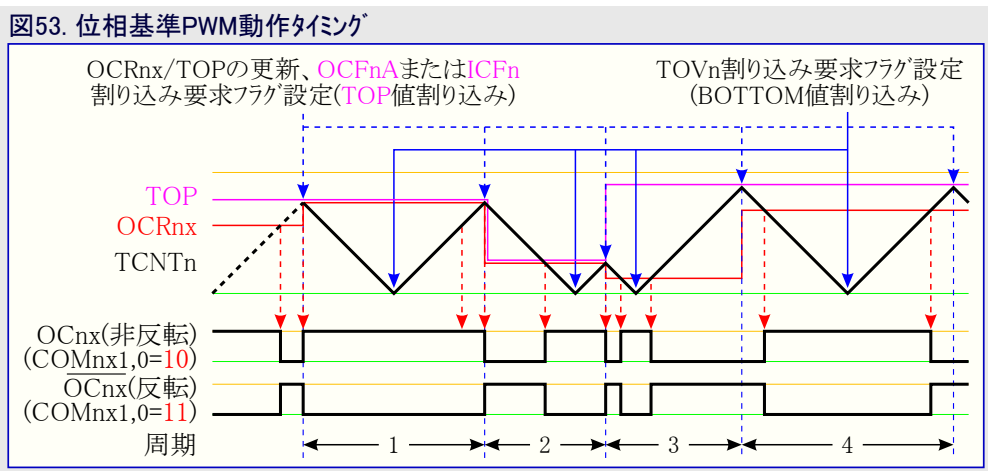
## 位相基準PWM動作

位相基準パルス幅変調(PWM)動作(WGMn3~0=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(COMnx1,0=10)での比較出力(OCnx)は上昇計数中のTCNTnとOCRnx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(COMnx1,0=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ビットに固定、若しくはOCRnAかICRn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(OCRnAまたはICRnが\$0003設定)、最大分解能は16ビット(OCRnAまたはICRnがMAX設定)です。ビットでのPWM分解能は次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(WGMn3~0=0001,0010,0011)、ICRn値(WGMn3~0=1010)またはOCRnA値(WGMn3~0=1011)のどれ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TCNTn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。位相基準PWM動作のタイミング図は図53.で示されます。本図はOCRnAかICRnがTOPを定義するのに使われる時の位相基準PWM動作を示します。TCNTn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OCRnx値を示し、TCNTn値との交点(接点)がTCNTnとOCRnx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(OCFnx)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(TOVn)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。OCRnAかICRnのどちらかがTOP値を定義するのに使われるとき、OCFnAまたはICFn割り込み要求フラグはOCRnxレジスタが(TOPに於いて)2重緩衝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。

TOP値を変更するとき、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、TCNTnとそのOCRnx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OCRnx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53.で示される第3周期が図解するようにタイマ/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OCRnxレジスタの更新時に見出せます。OCRnx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)が違う時にその周期の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OCnxピンでのPWM波形の生成を許します。COMnx1,0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COMnx1,0を'11'に設定することで生成できます(88頁の表60.をご覧ください)。実際のOCnx値はそのポートピンに対するデータ方向が出力(DDR\_OCnx=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TCNTnとOCRnx間の比較一致でOCnx(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TCNTnとOCRnx間の比較一致でOCnx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。変数Nは前置分周数(1,8,64,256,1024)を表します。

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

OCRnx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OCRnx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OCRnAが使われ(WGMn3~0=1011)、COMnA1,0=01なら、OCnA出力はデューティ比50%で交互に変化します。

## 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(WGMn3~0=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(COMnx1,0=10)での比較出力(OCnx)は上昇計数中のTCNTnとOCRnx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(COMnx1,0=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。

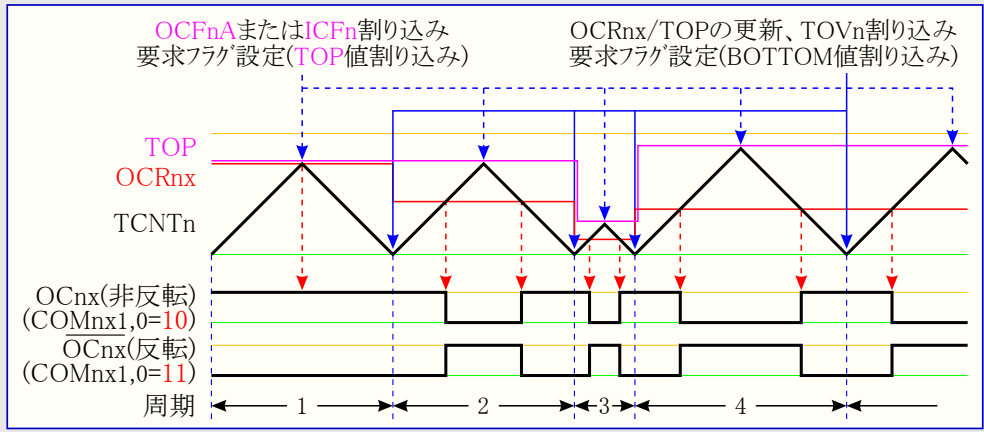
位相基準と位相/周波数基準PWM動作間の主な違いはOCRnxレジスタがOCRnx緩衝部によって更新される時(訳補:TOPとBOTTOM)です(図53と図54参照)。

位相/周波数基準PWM動作のPWM分解能はOCRnAかICRn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(OCRnAまたはICRnが\$0003設定)、最大分解能は16ビット(OCRnAまたはICRnがMAX設定)です。ビットでのPWM分解能は次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ICRn値(WGMn3~0=1000)かOCRnA値(WGMn3~0=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TCNTn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。位相/周波数基準PWM動作のタイミング図は図54.で示されます。この図はOCRnAかICRn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。TCNTn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OCRnx値を示し、TCNTn値との交点(接点)がTCNTnとOCRnx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(OCFnx)は比較一致が起こると設定(1)されます。

図54. 位相/周波数基準PWM動作タイミング



タイマ/カウンタ溢れ(TOVn)フラグはOCRnxレジスタが(BOTTOMに於いて)2重緩衝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。OCRnAかICRnのどちらかがTOP値を定義するのに使われるとき、OCFnxまたはICFnx割り込み要求フラグはタイマ/カウンタがTOPに到達する時毎に設定(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。

TOP値を変更するとき、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、TCNTnとそのOCRnx間で比較一致は決して起きません。

図54.が示すように、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。OCRnx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ICRnを使うことは決まったTOP値を使う時に上手くいきます。ICRnを使うことにより、OCnAでのPWM出力を生成するためにOCRnA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、OCRnAが2重緩衝機能のため、TOPとしてOCRnA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OCnxピンでのPWM波形の生成を許します。COMnx1,0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COMnx1,0を'11'に設定することで生成できます(88頁の表60.をご覧ください)。実際のOCnx値はそのポートピンに対するデータ方向が出力(DDR\_OCnx=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TCNTnとOCRnx間の比較一致でOCnx(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TCNTnとOCRnx間の比較一致でOCnx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。変数Nは前置分周数(1,8,64,256,1024)を表します。

$$f_{OCnxPFCPWM} = \frac{f_{clk\_1/O}}{2 \times N \times TOP}$$

OCRnx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OCRnx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OCRnAが使われ(WGMn3~0=1001)、COMnA1,0=01なら、OCnA出力はデューティ比50%で交互に変化します。



## 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( $clk_{Tn}$ )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 $OCR_{nx}$ レジスタが $OCR_{nx}$ 緩衝値で更新される時(2重緩衝を使う動作種別のみ)の情報を含まず、図55は $OCF_{nx}$ の設定についてのタイミング図を示します。

図55. 前置分周なし(1/1)のタイマ/カウンタ、 $OCF_{nx}$ 設定 タイミング

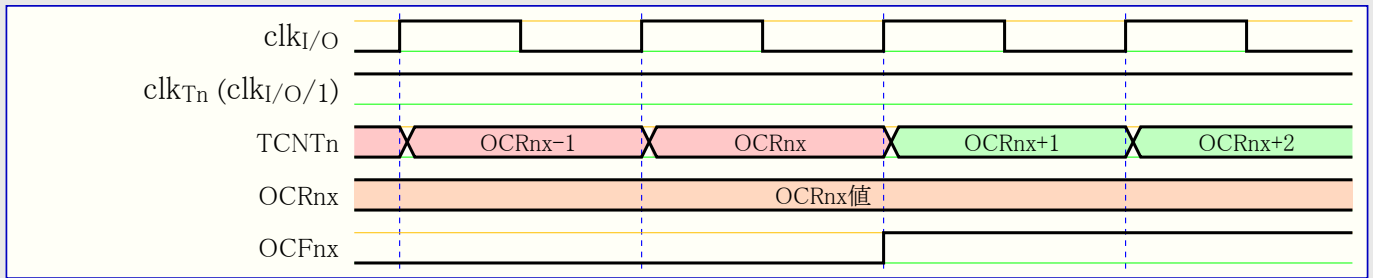


図56. は同じタイミング データを示しますが、前置分周器が許可されています。

図56. 前置分周器( $f_{clk_{I/O}/8}$ )のタイマ/カウンタ、 $OCF_{nx}$ 設定 タイミング

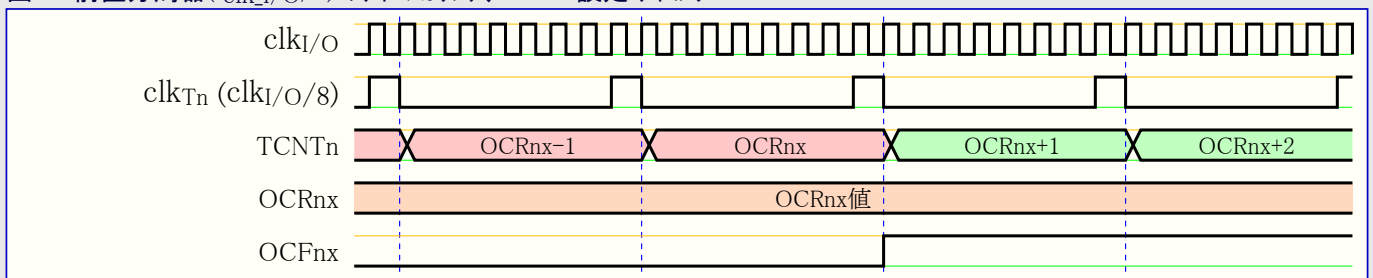


図57. は各動作種別でのTOP近辺の計数手順を示します。位相/周波数基準PWM動作使用時の $OCR_{1x}$ レジスタはBOTTOMで更新されます。タイミング図は同じになりますが、当然TOPはBOTTOMで、TOP-1はBOTTOM+1でなどのように置き換えられます。BOTTOMで $TOV_n$ を設定(1)する動作種別についても、同様な名称変更が適用されます。

図57. 前置分周器なし(1/1)のタイマ/カウンタ、TOP近辺 タイミング

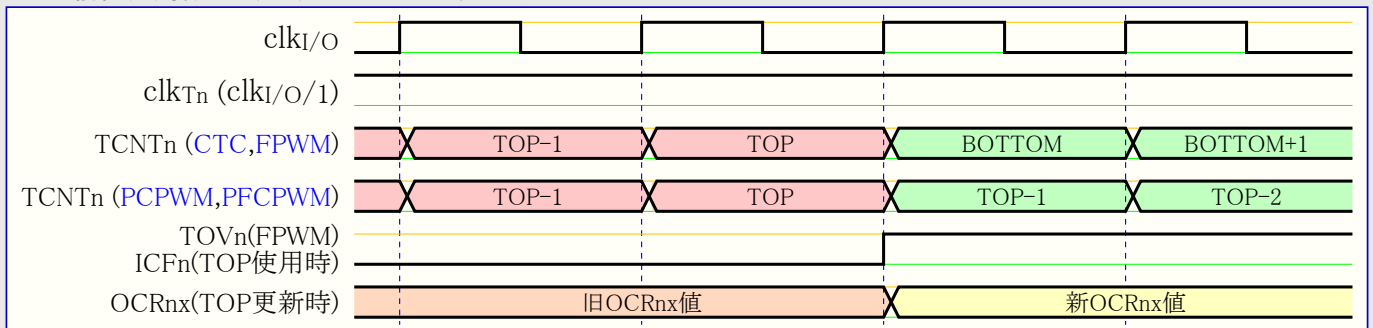
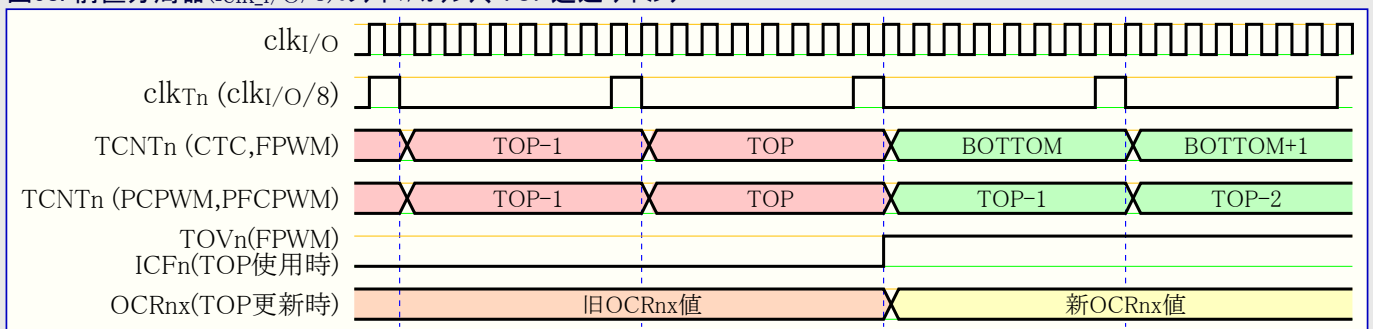


図58. は同じタイミング データを示しますが、前置分周器が許可されています。

図58. 前置分周器( $f_{clk_{I/O}/8}$ )のタイマ/カウンタ、TOP近辺 タイミング



## 16ビット タイマ/カウンタ1,3用レジスタ

### タイマ/カウンタ1制御レジスタA (Timer/Counter1 Control Register A) TCCR1A

| ビット         | 7      | 6      | 5      | 4      | 3      | 2      | 1     | 0     |        |
|-------------|--------|--------|--------|--------|--------|--------|-------|-------|--------|
| \$2F (\$4F) | COM1A1 | COM1A0 | COM1B1 | COM1B0 | COM1C1 | COM1C0 | WGM11 | WGM10 | TCCR1A |
| Read/Write  | R/W    | R/W    | R/W    | R/W    | R/W    | R/W    | R/W   | R/W   |        |
| 初期値         | 0      | 0      | 0      | 0      | 0      | 0      | 0     | 0     |        |

### タイマ/カウンタ3制御レジスタA (Timer/Counter3 Control Register A) TCCR3A

| ビット        | 7      | 6      | 5      | 4      | 3      | 2      | 1     | 0     |        |
|------------|--------|--------|--------|--------|--------|--------|-------|-------|--------|
| (\$8B)     | COM3A1 | COM3A0 | COM3B1 | COM3B0 | COM3C1 | COM3C0 | WGM31 | WGM30 | TCCR3A |
| Read/Write | R/W    | R/W    | R/W    | R/W    | R/W    | R/W    | R/W   | R/W   |        |
| 初期値        | 0      | 0      | 0      | 0      | 0      | 0      | 0     | 0     |        |

- ビット7,6 – COMnA1,COMnA0 : 比較nA出力選択 (Compare Output Mode A bit 1 and 0)
- ビット5,4 – COMnB1,COMnB0 : 比較nB出力選択 (Compare Output Mode B bit 1 and 0)
- ビット3,2 – COMnC1,COMnC0 : 比較nC出力選択 (Compare Output Mode C bit 1 and 0)

COMnA1,0, COMnB1,0, COMnC1,0は各々OCnA,OCnB,OCnC比較出力ピンの動作を制御します。COMnA1,0ビットの1つまたは両方が1を書かれると、OCnA出力はそのI/Oピンの標準ポート機能を無効にし、そのI/Oピンに接続されます。COMnB1,0ビットの1つまたは両方が1を書かれると、OCnB出力はそのI/Oピンの標準ポート機能を無効にし、そのI/Oピンに接続されます。COMnC1,0ビットの1つまたは両方が1を書かれると、OCnC出力はそのI/Oピンの標準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、OCnA,OCnB,OCnC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。

OCnA,OCnB,OCnCがピンに接続されるとき、COMnx1,0ビットの機能はWGMn3~0ビット設定に依存します。表58.はWGMn3~0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COMnx1,0ビット機能を示します。

表58. 非PWM動作での比較出力選択 (注: nは1または3, xはA,BまたはC)

| COMnx1 | COMnx0 | 意味                    |
|--------|--------|-----------------------|
| 0      | 0      | 標準ポート動作 (OCnx切断)      |
| 0      | 1      | 比較一致でOCnxピントグル(交互)出力  |
| 1      | 0      | 比較一致でOCnxピン Lowレベル出力  |
| 1      | 1      | 比較一致でOCnxピン Highレベル出力 |

表59.はWGMn3~0ビットが高速PWM動作に設定される時のCOMnx1,0ビット機能を示します。

表59. 高速PWM動作での比較出力選択 (注: nは1または3, xはA,BまたはC)

| COMnx1 | COMnx0 | 意味   |
|--------|--------|--|
| 0      | 0      | 標準ポート動作 (OCnx切断)   |
| 0      | 1      | WGMn3~0=1111 : 比較一致でOCnAピントグル(交互)出力、OCnB/OCnC切断(標準ポート動作)<br>WGMn3~0上記以外 : 標準ポート動作 (OCnx切断) |
| 1      | 0      | 比較一致でLow、BOTTOMでHighをOCnxピンへ出力 (非反転動作)   |
| 1      | 1      | 比較一致でHigh、BOTTOMでLowをOCnxピンへ出力 (反転動作)  |

注: COMnx1が設定(1)され、OCRnx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。より多くの詳細については84頁の「高速PWM動作」をご覧ください。

表60.はWGMn3~0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COMnx1,0ビット機能を示します。

表60. 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: nは1または3, xはA,BまたはC, Xは0または1)

| COMnx1 | COMnx0 | 意味   |
|--------|--------|--|
| 0      | 0      | 標準ポート動作 (OCnx切断)   |
| 0      | 1      | WGMn3~0=10X1 : 比較一致でOCnAピントグル(交互)出力、OCnB/OCnC切断(標準ポート動作)<br>WGMn3~0上記以外 : 標準ポート動作 (OCnx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OCnx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OCnxピンへ出力   |

注: COMnx1が設定(1)され、OCRnxがTOPと等しい時に特別な状態が起きます。より多くの詳細については85頁の「位相基準PWM動作」をご覧ください。

•ビット1,0 – WGMn1,WGMn0 : 波形生成種別 (Waveform Generation Mode bit 1 and 0)

タイマ/カウンタ制御レジスタB(TCCRnB)で得られるWGMn3,2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(表61.参照)。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。83頁の「動作種別」をご覧ください。

表61. 波形生成種別選択

| 番号 | WGMn3 | WGMn2<br>(CTCn) | WGMn1<br>(PWMn1) | WGMn0<br>(PWMn0) | タイマ/カウンタ動作種別          | TOP値   | OCRnx<br>更新時 | TOVn<br>設定時 |
|----|-------|-----------------|------------------|------------------|-----------------------|--------|--------------|-------------|
| 0  | 0     | 0               | 0                | 0                | 標準動作                  | \$FFFF | 即時           | 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 | OCRnA  | 即時           | MAX         |
| 5  | 0     | 1               | 0                | 1                | 8ビット高速PWM動作           | \$00FF | BOTTOM       | TOP         |
| 6  | 0     | 1               | 1                | 0                | 9ビット高速PWM動作           | \$01FF | BOTTOM       | TOP         |
| 7  | 0     | 1               | 1                | 1                | 10ビット高速PWM動作          | \$03FF | BOTTOM       | TOP         |
| 8  | 1     | 0               | 0                | 0                | 位相/周波数基準PWM動作         | ICRn   | BOTTOM       | BOTTOM      |
| 9  | 1     | 0               | 0                | 1                | 位相/周波数基準PWM動作         | OCRnA  | BOTTOM       | BOTTOM      |
| 10 | 1     | 0               | 1                | 0                | 位相基準PWM動作             | ICRn   | TOP          | BOTTOM      |
| 11 | 1     | 0               | 1                | 1                | 位相基準PWM動作             | OCRnA  | TOP          | BOTTOM      |
| 12 | 1     | 1               | 0                | 0                | 比較一致タイマ/カウンタ解除(CTC)動作 | ICRn   | 即時           | MAX         |
| 13 | 1     | 1               | 0                | 1                | (予約)                  | -      | -            | -           |
| 14 | 1     | 1               | 1                | 0                | 高速PWM動作               | ICRn   | BOTTOM       | TOP         |
| 15 | 1     | 1               | 1                | 1                | 高速PWM動作               | OCRnA  | BOTTOM       | TOP         |

注: CTCnとPWMn1,0ビット定義名は旧名です。WGMn2~0定義名を使ってください。しかし、これらのビットの機能と位置は旧版のタイマ/カウンタと一致します。

### タイマ/カウンタ1制御レジスタB (Timer/Counter1 Control Register B) TCCR1B

| ビット         | 7     | 6     | 5 | 4     | 3     | 2    | 1    | 0    |        |
|-------------|-------|-------|---|-------|-------|------|------|------|--------|
| \$2E (\$4E) | ICNC1 | ICES1 | – | WGM13 | WGM12 | CS12 | CS11 | CS10 | TCCR1B |
| Read/Write  | R/W   | R/W   | R | R/W   | R/W   | R/W  | R/W  | R/W  |        |
| 初期値         | 0     | 0     | 0 | 0     | 0     | 0    | 0    | 0    |        |

### タイマ/カウンタ3制御レジスタB (Timer/Counter3 Control Register B) TCCR3B

| ビット        | 7     | 6     | 5 | 4     | 3     | 2    | 1    | 0    |        |
|------------|-------|-------|---|-------|-------|------|------|------|--------|
| (\$8A)     | ICNC3 | ICES3 | – | WGM33 | WGM32 | CS32 | CS31 | CS30 | TCCR3B |
| Read/Write | R/W   | R/W   | R | R/W   | R/W   | R/W  | R/W  | R/W  |        |
| 初期値        | 0     | 0     | 0 | 0     | 0     | 0    | 0    | 0    |        |

• **ビット7 – ICNCn : 捕獲起動入力n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(ICPn)ピンからの入力が濾波されます。この濾波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ICPnピンの採取を必要とします。雑音消去器が許可されると、捕獲入力はそれによって4発振器(システムクロック)周期遅らされます。

• **ビット6 – ICESn : 捕獲起動入力n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(ICPn)ピンのどちらかの端(エッジ)を選びます。ICESn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、ICESn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ICESn設定に従って起動されると、カウンタ値が**捕獲レジスタ(ICRn)**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(ICFn)**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。

ICRnがTOP値として使われると(TCCRnAとTCCRnBに配置された**WGMn3~0ビット**の記述をご覧ください)、ICPn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5 – Res : 予約 (Reserved)**

このビットは将来の使用に対して予約されています。将来のデバイスとの共通性を保つため、TCCRnBが書かれるとき、このビットは0を書かれなければなりません。

• **ビット4,3 – WGMn3,WGMn2 : 波形生成種別 (Waveform Generation Mode bit 3 and 2)**

TCCRnAの**WGMn1,0ビット**記述をご覧ください。

• **ビット2~0 – CSn2,CSn1,CSn0 : クロック選択n (Clock Select, bit 2,1 and 0)**

この3つのクロック選択ビットはタイマ/カウンタ(TCNTn)によって使われるべきクロック元を選びます。[図55](#)と[図56](#)をご覧ください。

タイマ/カウンタnに対して外部ピン(クロック)動作が使われる場合、例えばTnピンが出力として設定されても、Tn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の制御を許します。

表62. タイマ/カウンタn入力クロック選択

| CS12 | CS11 | CS10 | 意味                                |
|------|------|------|-----------------------------------|
| 0    | 0    | 0    | 停止 (タイマ/カウンタn動作停止)                |
| 0    | 0    | 1    | clk <sub>I/O</sub> (前置分周なし)       |
| 0    | 1    | 0    | clk <sub>I/O</sub> /8 (8分周)       |
| 0    | 1    | 1    | clk <sub>I/O</sub> /64 (64分周)     |
| 1    | 0    | 0    | clk <sub>I/O</sub> /256 (256分周)   |
| 1    | 0    | 1    | clk <sub>I/O</sub> /1024 (1024分周) |
| 1    | 1    | 0    | Tnピンの下降端 (外部クロック)                 |
| 1    | 1    | 1    | Tnピンの上昇端 (外部クロック)                 |

## タイマ/カウンタ1制御レジスタC (Timer/Counter1 Control Register C) TCCR1C

| ビット (\$7A) | 7     | 6     | 5     | 4 | 3 | 2 | 1 | 0 | TCCR1C |
|------------|-------|-------|-------|---|---|---|---|---|--------|
|            | FOC1A | FOC1B | FOC1C | - | - | - | - | - |        |
| Read/Write | W     | W     | W     | R | R | R | R | R |        |
| 初期値        | 0     | 0     | 0     | 0 | 0 | 0 | 0 | 0 |        |

## タイマ/カウンタ3制御レジスタC (Timer/Counter3 Control Register C) TCCR3C

| ビット (\$8C) | 7     | 6     | 5     | 4 | 3 | 2 | 1 | 0 | TCCR3C |
|------------|-------|-------|-------|---|---|---|---|---|--------|
|            | FOC3A | FOC3B | FOC3C | - | - | - | - | - |        |
| Read/Write | W     | W     | W     | R | R | R | R | R |        |
| 初期値        | 0     | 0     | 0     | 0 | 0 | 0 | 0 | 0 |        |

- ビット7 - FOCnA : OCnA強制変更 (Force Output Compare A)
- ビット6 - FOCnB : OCnB強制変更 (Force Output Compare B)
- ビット5 - FOCnC : OCnC強制変更 (Force Output Compare C)

FOCnA/FOCnB/FOCnCビットはWGMn3~0ビットが非PWM動作を指示する時だけ有効です。FOCnA/FOCnB/FOCnC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。OCnx出力はCOMnx1,0ビット設定に従って変更されます。FOCnA/FOCnB/FOCnCビットがストロープ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COMnx1,0ビットに存在する値です。

FOCnA/FOCnB/FOCnCストロープは何れの割り込みの生成もTOPとしてOCRnA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。

FOCnA/FOCnB/FOCnCビットは常に0として読みます。

- ビット4~0 - Res : 予約 (Reserved)

これらのビットは将来の使用に対して予約されています。将来のデバイスとの共通性を保つため、TCCRnCが書かれるとき、これらのビットは0を書かれなければなりません。

## タイマ/カウンタ1 (Timer/Counter1) TCNT1H,TCNT1L (TCNT1)

| ビット \$2D (\$4D) | 15    | 14  | 13  | 12  | 11  | 10  | 9   | 8     | TCNT1H |
|-----------------|-------|-----|-----|-----|-----|-----|-----|-------|--------|
|                 | (MSB) |     |     |     |     |     |     |       |        |
| Read/Write      | R/W   | R/W | R/W | R/W | R/W | R/W | R/W | R/W   |        |
| 初期値             | 0     | 0   | 0   | 0   | 0   | 0   | 0   | 0     |        |
| ビット \$2C (\$4C) | 7     | 6   | 5   | 4   | 3   | 2   | 1   | 0     | TCNT1L |
|                 |       |     |     |     |     |     |     | (LSB) |        |
| Read/Write      | R/W   | R/W | R/W | R/W | R/W | R/W | R/W | R/W   |        |
| 初期値             | 0     | 0   | 0   | 0   | 0   | 0   | 0   | 0     |        |

## タイマ/カウンタ3 (Timer/Counter3) TCNT3H,TCNT3L (TCNT3)

| ビット (\$89) | 15    | 14  | 13  | 12  | 11  | 10  | 9   | 8     | TCNT3H |
|------------|-------|-----|-----|-----|-----|-----|-----|-------|--------|
|            | (MSB) |     |     |     |     |     |     |       |        |
| Read/Write | R/W   | R/W | R/W | R/W | R/W | R/W | R/W | R/W   |        |
| 初期値        | 0     | 0   | 0   | 0   | 0   | 0   | 0   | 0     |        |
| ビット (\$88) | 7     | 6   | 5   | 4   | 3   | 2   | 1   | 0     | TCNT3L |
|            |       |     |     |     |     |     |     | (LSB) |        |
| Read/Write | R/W   | R/W | R/W | R/W | R/W | R/W | R/W | R/W   |        |
| 初期値        | 0     | 0   | 0   | 0   | 0   | 0   | 0   | 0     |        |

この2つのタイマ/カウンタ I/O位置(TCNTnHとTCNTnLを合わせたTCNTn)は、読み書き両方に対してタイマ/カウンタ部の16ビットカウンタに直接アクセスします。CPUがこれらのレジスタをアクセスする時に上位と下位の両バイトが同時に読み書きされるのを保証するため、このアクセスは8ビット上位バイト一時レジスタ(TEMP)を使って実行されます。この一時レジスタは他の全ての16ビットレジスタによって共用されます。[76頁の「16ビットレジスタのアクセス」](#)をご覧ください。

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

TCNTn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。



タイマ/カウンタ1 比較レジスタ (Timer/Counter1 Output Compare Register A) OCR1AH,OCR1AL (OCR1A)

|             |       |     |     |     |     |     |     |       |        |
|-------------|-------|-----|-----|-----|-----|-----|-----|-------|--------|
| ビット         | 15    | 14  | 13  | 12  | 11  | 10  | 9   | 8     |        |
| \$2B (\$4B) | (MSB) |     |     |     |     |     |     |       | OCR1AH |
| Read/Write  | R/W   | R/W | R/W | R/W | R/W | R/W | R/W | R/W   |        |
| 初期値         | 0     | 0   | 0   | 0   | 0   | 0   | 0   | 0     |        |
| ビット         | 7     | 6   | 5   | 4   | 3   | 2   | 1   | 0     |        |
| \$2A (\$4A) |       |     |     |     |     |     |     | (LSB) | OCR1AL |
| Read/Write  | R/W   | R/W | R/W | R/W | R/W | R/W | R/W | R/W   |        |
| 初期値         | 0     | 0   | 0   | 0   | 0   | 0   | 0   | 0     |        |

タイマ/カウンタ1 比較レジスタ (Timer/Counter1 Output Compare Register B) OCR1BH,OCR1BL (OCR1B)

|             |       |     |     |     |     |     |     |       |        |
|-------------|-------|-----|-----|-----|-----|-----|-----|-------|--------|
| ビット         | 15    | 14  | 13  | 12  | 11  | 10  | 9   | 8     |        |
| \$29 (\$49) | (MSB) |     |     |     |     |     |     |       | OCR1BH |
| Read/Write  | R/W   | R/W | R/W | R/W | R/W | R/W | R/W | R/W   |        |
| 初期値         | 0     | 0   | 0   | 0   | 0   | 0   | 0   | 0     |        |
| ビット         | 7     | 6   | 5   | 4   | 3   | 2   | 1   | 0     |        |
| \$28 (\$48) |       |     |     |     |     |     |     | (LSB) | OCR1BL |
| Read/Write  | R/W   | R/W | R/W | R/W | R/W | R/W | R/W | R/W   |        |
| 初期値         | 0     | 0   | 0   | 0   | 0   | 0   | 0   | 0     |        |

タイマ/カウンタ1 比較レジスタ (Timer/Counter1 Output Compare Register C) OCR1CH,OCR1CL (OCR1C)

|            |       |     |     |     |     |     |     |       |        |
|------------|-------|-----|-----|-----|-----|-----|-----|-------|--------|
| ビット        | 15    | 14  | 13  | 12  | 11  | 10  | 9   | 8     |        |
| (\$79)     | (MSB) |     |     |     |     |     |     |       | OCR1CH |
| Read/Write | R/W   | R/W | R/W | R/W | R/W | R/W | R/W | R/W   |        |
| 初期値        | 0     | 0   | 0   | 0   | 0   | 0   | 0   | 0     |        |
| ビット        | 7     | 6   | 5   | 4   | 3   | 2   | 1   | 0     |        |
| (\$78)     |       |     |     |     |     |     |     | (LSB) | OCR1CL |
| Read/Write | R/W   | R/W | R/W | R/W | R/W | R/W | R/W | R/W   |        |
| 初期値        | 0     | 0   | 0   | 0   | 0   | 0   | 0   | 0     |        |

タイマ/カウンタ3 比較レジスタ (Timer/Counter3 Output Compare Register A) OCR3AH,OCR3AL (OCR3A)

|            |       |     |     |     |     |     |     |       |        |
|------------|-------|-----|-----|-----|-----|-----|-----|-------|--------|
| ビット        | 15    | 14  | 13  | 12  | 11  | 10  | 9   | 8     |        |
| (\$87)     | (MSB) |     |     |     |     |     |     |       | OCR3AH |
| Read/Write | R/W   | R/W | R/W | R/W | R/W | R/W | R/W | R/W   |        |
| 初期値        | 0     | 0   | 0   | 0   | 0   | 0   | 0   | 0     |        |
| ビット        | 7     | 6   | 5   | 4   | 3   | 2   | 1   | 0     |        |
| (\$86)     |       |     |     |     |     |     |     | (LSB) | OCR3AL |
| Read/Write | R/W   | R/W | R/W | R/W | R/W | R/W | R/W | R/W   |        |
| 初期値        | 0     | 0   | 0   | 0   | 0   | 0   | 0   | 0     |        |

タイマ/カウンタ3 比較レジスタ (Timer/Counter3 Output Compare Register B) OCR3BH,OCR3BL (OCR3B)

|            |       |     |     |     |     |     |     |       |        |
|------------|-------|-----|-----|-----|-----|-----|-----|-------|--------|
| ビット        | 15    | 14  | 13  | 12  | 11  | 10  | 9   | 8     |        |
| (\$85)     | (MSB) |     |     |     |     |     |     |       | OCR3BH |
| Read/Write | R/W   | R/W | R/W | R/W | R/W | R/W | R/W | R/W   |        |
| 初期値        | 0     | 0   | 0   | 0   | 0   | 0   | 0   | 0     |        |
| ビット        | 7     | 6   | 5   | 4   | 3   | 2   | 1   | 0     |        |
| (\$84)     |       |     |     |     |     |     |     | (LSB) | OCR3BL |
| Read/Write | R/W   | R/W | R/W | R/W | R/W | R/W | R/W | R/W   |        |
| 初期値        | 0     | 0   | 0   | 0   | 0   | 0   | 0   | 0     |        |

## タイマ/カウンタ3 比較レジスタ (Timer/Counter3 Output Compare Register C) OCR3CH,OCR3CL (OCR3C)

| ビット (\$83) | 15    | 14  | 13  | 12  | 11  | 10  | 9   | 8     |        |
|------------|-------|-----|-----|-----|-----|-----|-----|-------|--------|
|            | (MSB) |     |     |     |     |     |     |       | OCR3CH |
| Read/Write | R/W   | R/W | R/W | R/W | R/W | R/W | R/W | R/W   |        |
| 初期値        | 0     | 0   | 0   | 0   | 0   | 0   | 0   | 0     |        |
| ビット (\$82) | 7     | 6   | 5   | 4   | 3   | 2   | 1   | 0     |        |
|            |       |     |     |     |     |     |     | (LSB) | OCR3CL |
| Read/Write | R/W   | R/W | R/W | R/W | R/W | R/W | R/W | R/W   |        |
| 初期値        | 0     | 0   | 0   | 0   | 0   | 0   | 0   | 0     |        |

この比較レジスタは継続的にカウンタ(TCNTn)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OCnxピンでの波形出力を生成するのに使えます。

この比較レジスタは容量が16ビットです。CPUがこれらのレジスタへ書く時に上位と下位の両バイトが同時に書かれるのを保証するため、このアクセスは8ビット上位バイト一時レジスタ(TEMP)を使って実行されます。この一時レジスタは他の全ての16ビットレジスタによって共用されます。76頁の「16ビットレジスタのアクセス」をご覧ください。

## タイマ/カウンタ1 捕獲レジスタ (Timer/Counter1 Input Capture Register) ICR1H,ICR1L (ICR1)

| ビット \$27 (\$47) | 15    | 14  | 13  | 12  | 11  | 10  | 9   | 8     |       |
|-----------------|-------|-----|-----|-----|-----|-----|-----|-------|-------|
|                 | (MSB) |     |     |     |     |     |     |       | ICR1H |
| Read/Write      | R/W   | R/W | R/W | R/W | R/W | R/W | R/W | R/W   |       |
| 初期値             | 0     | 0   | 0   | 0   | 0   | 0   | 0   | 0     |       |
| ビット \$26 (\$46) | 7     | 6   | 5   | 4   | 3   | 2   | 1   | 0     |       |
|                 |       |     |     |     |     |     |     | (LSB) | ICR1L |
| Read/Write      | R/W   | R/W | R/W | R/W | R/W | R/W | R/W | R/W   |       |
| 初期値             | 0     | 0   | 0   | 0   | 0   | 0   | 0   | 0     |       |

## タイマ/カウンタ3 捕獲レジスタ (Timer/Counter3 Input Capture Register) ICR3H,ICR3L (ICR3)

| ビット (\$81) | 15    | 14  | 13  | 12  | 11  | 10  | 9   | 8     |       |
|------------|-------|-----|-----|-----|-----|-----|-----|-------|-------|
|            | (MSB) |     |     |     |     |     |     |       | ICR3H |
| Read/Write | R/W   | R/W | R/W | R/W | R/W | R/W | R/W | R/W   |       |
| 初期値        | 0     | 0   | 0   | 0   | 0   | 0   | 0   | 0     |       |
| ビット (\$80) | 7     | 6   | 5   | 4   | 3   | 2   | 1   | 0     |       |
|            |       |     |     |     |     |     |     | (LSB) | ICR3L |
| Read/Write | R/W   | R/W | R/W | R/W | R/W | R/W | R/W | R/W   |       |
| 初期値        | 0     | 0   | 0   | 0   | 0   | 0   | 0   | 0     |       |

この捕獲レジスタはICPnピン(またはタイマ/カウンタ1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(TCNTn)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。

この捕獲レジスタは容量が16ビットです。CPUがこれらのレジスタをアクセスする時に上位と下位の両バイトが同時に読まれるのを保証するため、このアクセスは8ビット上位バイト一時レジスタ(TEMP)を使って実行されます。この一時レジスタは他の全ての16ビットレジスタによって共用されます。76頁の「16ビットレジスタのアクセス」をご覧ください。

## タイマ/カウンタ割り込み許可レジスタ (Timer/Counter Interrupt Mask Register) TIMSK

| ビット         | 7     | 6     | 5      | 4      | 3      | 2     | 1     | 0     |       |
|-------------|-------|-------|--------|--------|--------|-------|-------|-------|-------|
| \$37 (\$57) | OCIE2 | TOIE2 | TICIE1 | OCIE1A | OCIE1B | TOIE1 | OCIE0 | TOIE0 | TIMSK |
| Read/Write  | R/W   | R/W   | R/W    | R/W    | R/W    | R/W   | R/W   | R/W   |       |
| 初期値         | 0     | 0     | 0      | 0      | 0      | 0     | 0     | 0     |       |

**注:** このレジスタは様々なタイマ/カウンタ用割り込み制御ビットを含みますが、タイマ/カウンタ1だけがこの項で記述されます。残りのビットは各々のタイマ/カウンタ項で記述されます。

### • ビット5 – TICIE1 : タイマ/カウンタ1捕獲割り込み許可 (Timer/Counter1 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1捕獲割り込みが許可されます。タイマ/カウンタ割り込み要求フラグレジスタ(TIFR)に配置された捕獲1割り込み要求フラグ(ICF1)が設定(1)されると、対応する割り込みベクタ(37頁の「割り込み」参照)が実行されます。

### • ビット4 – OCIE1A : タイマ/カウンタ1比較A割り込み許可 (Timer/Counter1 Output Compare A Match 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1比較A一致割り込みが許可されます。タイマ/カウンタ割り込み要求フラグレジスタ(TIFR)に配置された比較1A割り込み要求フラグ(OCF1A)が設定(1)されると、対応する割り込みベクタ(37頁の「割り込み」参照)が実行されます。

### • ビット3 – OCIE1B : タイマ/カウンタ1比較B割り込み許可 (Timer/Counter1 Output Compare B Match 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1比較B一致割り込みが許可されます。タイマ/カウンタ割り込み要求フラグレジスタ(TIFR)に配置された比較1B割り込み要求フラグ(OCF1B)が設定(1)されると、対応する割り込みベクタ(37頁の「割り込み」参照)が実行されます。

### • ビット2 – TOIE1 : タイマ/カウンタ1溢れ割り込み許可 (Timer/Counter1 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1溢れ割り込みが許可されます。タイマ/カウンタ割り込み要求フラグレジスタ(TIFR)に配置されたタイマ/カウンタ1溢れ割り込み要求フラグ(TOV1)が設定(1)されると、対応する割り込みベクタ(37頁の「割り込み」参照)が実行されます。

## 拡張タイマ/カウンタ割り込み許可レジスタ (Extended Timer/Counter Interrupt Mask Register) ETIMSK

| ビット        | 7 | 6 | 5      | 4      | 3      | 2     | 1      | 0      |        |
|------------|---|---|--------|--------|--------|-------|--------|--------|--------|
| (\$7D)     | - | - | TICIE3 | OCIE3A | OCIE3B | TOIE3 | OCIE3C | OCIE1C | ETIMSK |
| Read/Write | R | R | R/W    | R/W    | R/W    | R/W   | R/W    | R/W    |        |
| 初期値        | 0 | 0 | 0      | 0      | 0      | 0     | 0      | 0      |        |

**注:** このレジスタはATmega103互換動作では利用できません。

### • ビット7,6 – Res : 予約 (Reserved)

これらのビットは将来使用するために予約されています。将来のデバイスとの共通性を保つため、ETIMSK書き込み時、これらのビットは0に設定されなければなりません。

### • ビット5 – TICIE3 : タイマ/カウンタ3捕獲割り込み許可 (Timer/Counter3 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3捕獲割り込みが許可されます。拡張タイマ/カウンタ割り込み要求フラグレジスタ(ETIFR)に配置された捕獲3割り込み要求フラグ(ICF3)が設定(1)されると、対応する割り込みベクタ(37頁の「割り込み」参照)が実行されます。

### • ビット4 – OCIE3A : タイマ/カウンタ3比較A割り込み許可 (Timer/Counter3 Output Compare A Match 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3比較A一致割り込みが許可されます。拡張タイマ/カウンタ割り込み要求フラグレジスタ(ETIFR)に配置された比較3A割り込み要求フラグ(OCF3A)が設定(1)されると、対応する割り込みベクタ(37頁の「割り込み」参照)が実行されます。

### • ビット3 – OCIE3B : タイマ/カウンタ3比較B割り込み許可 (Timer/Counter3 Output Compare B Match 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3比較B一致割り込みが許可されます。拡張タイマ/カウンタ割り込み要求フラグレジスタ(ETIFR)に配置された比較3B割り込み要求フラグ(OCF3B)が設定(1)されると、対応する割り込みベクタ(37頁の「割り込み」参照)が実行されます。

### • ビット2 – TOIE3 : タイマ/カウンタ3溢れ割り込み許可 (Timer/Counter3 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3溢れ割り込みが許可されます。拡張タイマ/カウンタ割り込み要求フラグレジスタ(ETIFR)に配置されたタイマ/カウンタ3溢れ割り込み要求フラグ(TOV3)が設定(1)されると、対応する割り込みベクタ(37頁の「割り込み」参照)が実行されます。

• **ビット1 – OCIE3C : タイマ/カウンタ3比較C割り込み許可** (Timer/Counter3 Output Compare C 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3比較C一致割り込みが許可されます。拡張タイマ/カウンタ割り込み要求フラグレジスタ(ETIFR)に配置された**比較3C割り込み要求フラグ(OCF3C)**が設定(**1**)されると、対応する割り込みベクタ(37頁の「割り込み」参照)が実行されます。

• **ビット0 – OCIE1C : タイマ/カウンタ1比較C割り込み許可** (Timer/Counter1 Output Compare C 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1比較C一致割り込みが許可されます。拡張タイマ/カウンタ割り込み要求フラグレジスタ(ETIFR)に配置された**比較1C割り込み要求フラグ(OCF1C)**が設定(**1**)されると、対応する割り込みベクタ(37頁の「割り込み」参照)が実行されます。

**タイマ/カウンタ割り込み要求フラグレジスタ (Timer/Counter Interrupt Flag Register) TIFR**

| ビット         | 7    | 6    | 5    | 4     | 3     | 2    | 1    | 0    |      |
|-------------|------|------|------|-------|-------|------|------|------|------|
| \$36 (\$56) | OCF2 | TOV2 | ICF1 | OCF1A | OCF1B | TOV1 | OCF0 | TOV0 | TIFR |
| Read/Write  | R/W  | R/W  | R/W  | R/W   | R/W   | R/W  | R/W  | R/W  |      |
| 初期値         | 0    | 0    | 0    | 0     | 0     | 0    | 0    | 0    |      |

**注:** このレジスタは様々なタイマ/カウンタ用割り込み制御ビットを含みますが、タイマ/カウンタ1だけがこの項で記述されます。残りのビットは各々のタイマ/カウンタ項で記述されます。

• **ビット5 – ICF1 : タイマ/カウンタ1捕獲割り込み要求フラグ** (Timer/Counter1, Input Capture Flag)

ICP1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(ICR1)が**WGM13~0**によって**TOP値**として設定されると、ICF1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1割り込みベクタが実行されると、ICF1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ICF1は解除(**0**)できます。

• **ビット4 – OCF1A : タイマ/カウンタ1比較A割り込み要求フラグ** (Timer/Counter1, Output Compare A Match Flag)

このフラグはカウンタ(TCNT1)値が**比較Aレジスタ(OCR1A)**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(FOC1A)ストロブがOCF1Aフラグを設定(**1**)しないことに注意してください。

比較1A一致割り込みベクタが実行されると、OCF1A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OCF1Aは解除(**0**)できます。

• **ビット3 – OCF1B : タイマ/カウンタ1比較B割り込み要求フラグ** (Timer/Counter1, Output Compare B Match Flag)

このフラグはカウンタ(TCNT1)値が**比較Bレジスタ(OCR1B)**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(FOC1B)ストロブがOCF1Bフラグを設定(**1**)しないことに注意してください。

比較1B一致割り込みベクタが実行されると、OCF1B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OCF1Bは解除(**0**)できます。

• **ビット2 – TOV1 : タイマ/カウンタ1溢れ割り込み要求フラグ** (Timer/Counter1 Overflow Flag)

このフラグの(**1**)設定は**WGM13~0**ビット設定に依存します。標準または**CTC動作**でのTOV1フラグはタイマ/カウンタ1溢れ時に設定(**1**)されます。他の**WGM13~0**ビット設定を使う時のTOV1フラグ動作については**89頁の表61**を参照してください。

タイマ/カウンタ1溢れ割り込みベクタが実行されると、TOV1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TOV1は解除(**0**)できます。

**拡張タイマ/カウンタ割り込み要求フラグレジスタ (Extended Timer/Counter Interrupt Flag Register) ETIFR**

| ビット        | 7 | 6 | 5    | 4     | 3     | 2    | 1     | 0     |       |
|------------|---|---|------|-------|-------|------|-------|-------|-------|
| (\$7C)     | - | - | ICF3 | OCF3A | OCF3B | TOV3 | OCF3C | OCF1C | ETIFR |
| Read/Write | R | R | R/W  | R/W   | R/W   | R/W  | R/W   | R/W   |       |
| 初期値        | 0 | 0 | 0    | 0     | 0     | 0    | 0     | 0     |       |

• **ビット7,6 – Res : 予約 (Reserved)**

これらのビットは将来使用するために予約されています。将来のデバイスとの共通性を保つため、ETIFR書き込み時、これらのビットは**0**に設定されなければなりません。

• **ビット5 – ICF3 : タイマ/カウンタ3捕獲割り込み要求フラグ** (Timer/Counter3, Input Capture Flag)

ICP3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(ICR3)が**WGM33~0**によって**TOP値**として設定されると、ICF3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3割り込みベクタが実行されると、ICF3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ICF3は解除(**0**)できます。

• **ビット4 – OCF3A : タイマ/カウンタ3比較A割り込み要求フラグ** (Timer/Counter3, Output Compare A Match Flag)

このフラグはカウンタ(TCNT3)値が**比較Aレジスタ(OCR3A)**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(FOC3A)ストロブがOCF3Aフラグを設定(1)しないことに注意してください。

比較3A一致割り込みベクタが実行されると、OCF3A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OCF3Aは解除(0)できます。

• **ビット3 – OCF3B : タイマ/カウンタ3比較B割り込み要求フラグ** (Timer/Counter3, Output Compare B Match Flag)

このフラグはカウンタ(TCNT3)値が**比較Bレジスタ(OCR3B)**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(FOC3B)ストロブがOCF3Bフラグを設定(1)しないことに注意してください。

比較3B一致割り込みベクタが実行されると、OCF3B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OCF3Bは解除(0)できます。

• **ビット2 – TOV3 : タイマ/カウンタ3溢れ割り込み要求フラグ** (Timer/Counter3 Overflow Flag)

このフラグの(1)設定はWGM33~0ビット設定に依存します。**標準**または**CTC動作**でのTOV3フラグはタイマ/カウンタ3溢れ時に設定(1)されず。他のWGM33~0ビット設定を使う時のTOV3フラグ動作については**89頁の表61**を参照してください。

タイマ/カウンタ3溢れ割り込みベクタが実行されると、TOV3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TOV3は解除(0)できます。

• **ビット1 – OCF3C : タイマ/カウンタ3比較C割り込み要求フラグ** (Timer/Counter3, Output Compare C Match Flag)

このフラグはカウンタ(TCNT3)値が**比較Cレジスタ(OCR3C)**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(FOC3C)ストロブがOCF3Cフラグを設定(1)しないことに注意してください。

比較3C一致割り込みベクタが実行されると、OCF3C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OCF3Cは解除(0)できます。

• **ビット0 – OCF1C : タイマ/カウンタ1比較C割り込み要求フラグ** (Timer/Counter1, Output Compare C Match Flag)

このフラグはカウンタ(TCNT1)値が**比較Cレジスタ(OCR1C)**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(FOC1C)ストロブがOCF1Cフラグを設定(1)しないことに注意してください。

比較1C一致割り込みベクタが実行されると、OCF1C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OCF1Cは解除(0)できます。



## タイマ/カウンタ1、タイマ/カウンタ2、タイマ/カウンタ3の前置分周器

タイマ/カウンタ1、タイマ/カウンタ2、タイマ/カウンタ3は同じ前置分周器部を共有しますが、タイマ/カウンタは異なる前置分周器設定ができます。以下の記述は言及した全てのタイマ/カウンタに適用されます。

### 内部クロック元

タイマ/カウンタはシステムクロック(CSn2~0=001設定)によって直接的にクロック駆動できます。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つのタップの1つがクロック元として使えます。この前置分周したクロックは $f_{clk\_I/O}/8$ ,  $f_{clk\_I/O}/64$ ,  $f_{clk\_I/O}/256$ ,  $f_{clk\_I/O}/1024$ の何れかの周波数です。

### 前置分周器リセット

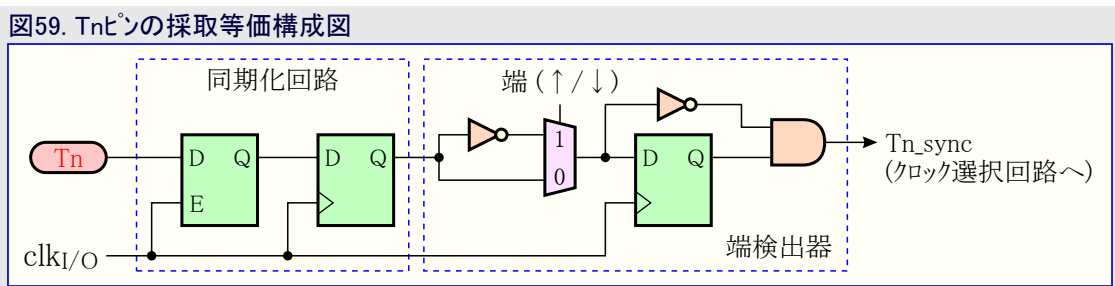
この前置分周器は自由走行で(換言するとタイマ/カウンタのクロック選択論理回路と無関係に動作する)、タイマ/カウンタ1、タイマ/カウンタ2、タイマ/カウンタ3によって共有されます。前置分周器はタイマ/カウンタのクロック選択によって影響を及ぼされないため、前置分周器の状態は前置分周したクロックが使われる状況に対して密接に関係します。タイマ/カウンタが許可され、前置分周器によってクロック駆動される(CSn2~0=5~2)とき、前置分周加工の一例が生じます。タイマ/カウンタが許可される時から最初の計数が起きるまでのシステムクロック周期数はNが前置分周値(8, 64, 256, 1024)とすると、 $1 \sim N+1$  システムクロック周期になり得ます。

プログラム実行にタイマ/カウンタを同期することに対して前置分周器リセットを使うことが可能です。しかし、同じ前置分周器を共有する他のタイマ/カウンタも前置分周を使う場合、注意が必要とされなければなりません。前置分周器リセットはそれが接続される全タイマ/カウンタに関する前置分周器周期に影響を及ぼします。

### 外部クロック元

Tnピンに印加された外部クロック元はタイマ/カウンタクロック( $f_{clk\_T1}/f_{clk\_T2}/f_{clk\_T3}$ )として使えます。このTnピンはピン同期化論理回路によって全てのシステムクロック周期に一度採取されます。この同期化(採取)された信号はその後に端(エッジ)検出器を通して通過されます。図59はTn同期化と端検出器論理回路の機能等価構成図を示します。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。

端検出器は上昇端(CSn2~0=111)または下降端(CSn2~0=110)の検出毎に1つの $clk_{T1}/clk_{T2}/clk_{T3}$ パルスを生成します。



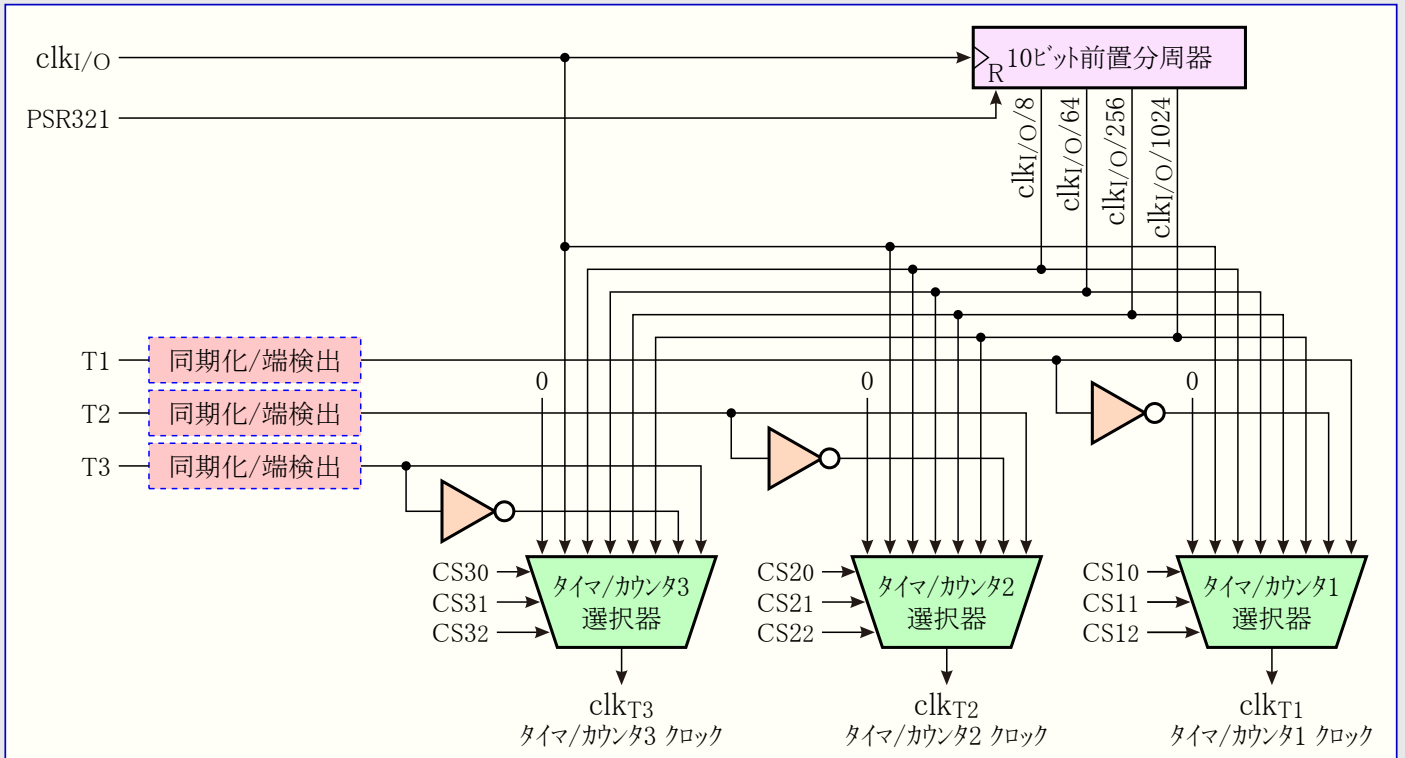
同期化と端検出器論理回路はTnピンへ印加された端からカウンタが更新されるまでに2.5~3.5システムクロック周期の遅延をもたらします。

クロック入力の許可と禁止はTn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長くなければなりません。この外部クロックは50%/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。端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quistの標本化定理)。然しながら、発振元(クリスタル発振子、セラミック振動子、コンデンサ)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。

図60. タイマ/カウンタ1、タイマ/カウンタ2、タイマ/カウンタ3の前置分周器部構成図



注: 入力ピン(T1/T2/T3)の同期化/端(エッジ)検出論理回路は図59.で示されます。

### 特殊I/O機能レジスタ (Special Function I/O Register) SFIOR

| ビット         | 7   | 6 | 5 | 4 | 3    | 2   | 1    | 0      |       |
|-------------|-----|---|---|---|------|-----|------|--------|-------|
| \$20 (\$40) | TSM | - | - | - | ACME | PUD | PSR0 | PSR321 | SFIOR |
| Read/Write  | R/W | R | R | R | R/W  | R/W | R/W  | R/W    |       |
| 初期値         | 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ビットに1を書くことはタイマ/カウンタ同期化動作を活性(有効)にします。この動作でPSR0とPSR321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。これは対応するタイマ/カウンタを停止し、設定中にそれらの1つが進行する危険なしに同じ値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、PSR0とPSR321ビットはハードウェアによって解除(0)され、同時にタイマ/カウンタが計数を始めます。

#### •ビット0 - PSR321 : タイマ/カウンタ3,2,1 前置分周器リセット (Prescaler Reset Timer/Counter 3,2,1)

このビットが1のとき、タイマ/カウンタ1、タイマ/カウンタ2、タイマ/カウンタ3の前置分周器はリセットします。TSMビットが設定(1)されている場合を除いて通常、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。タイマ/カウンタ1、タイマ/カウンタ2、タイマ/カウンタ3は同じ前置分周器を共用し、この前置分周器のリセットが、これら3つのタイマ/カウンタ全てに影響を及ぼすことに注意してください。

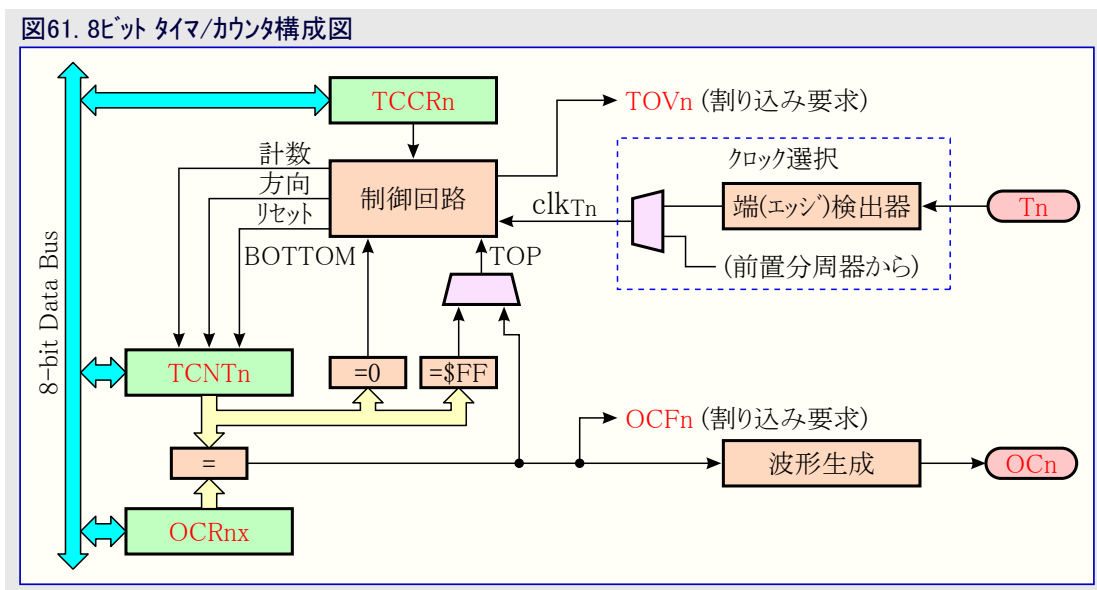
## 8ビット タイマ/カウンタ2 (PWM付き)

タイマ/カウンタ2は単一比較部付きの汎用8ビット タイマ/カウンタ部です。主な特徴は次のとおりです。

- 単一比較部付き計数器
- 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)
- 周波数発生器
- クロック用10ビット前置分周器
- 溢れと比較一致割り込み (TOV2とOCF2)

### 概要

この8ビット タイマ/カウンタの簡易化した構成図は図61.で示されます。I/Oピンの実際の配置については2頁の「ピン配置」を参照してください。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:原文は太字)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107頁の「8ビット タイマ/カウンタ2用レジスタ」で一覧されます。



### 関係レジスタ

タイマ/カウンタ(TCNT2)と比較レジスタ(OCR2)は8ビットのレジスタです。割り込み要求信号はタイマ/カウンタ割り込み要求レジスタ(TIFR)で全て見えます。全ての割り込みはタイマ/カウンタ割り込み許可レジスタ(TIMSK)で個別に遮蔽(禁止)されます。TIFRとTIMSKはこれらのレジスタが他のタイマ/カウンタと共用されるため、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T2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を制御します。クロック元が選ばれないと、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(clkT2)として参照されます。

2重緩衝化した比較レジスタ(OCR2)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(OC2)ピンでPWMまたは可変周波数出力を作成するための波形生成器によって使えます。101頁の「比較出力部」をご覧ください。この比較一致発生は比較一致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(OCF2)も設定(1)します。

### 定義

本文書でのレジスタとビット参照の多くは一般形で書かれます。小文字の'n'はタイマ/カウンタ番号、この場合は2で置き換えます。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が使われなければなりません(例えばタイマ/カウンタ2のカウンタ値のアクセスに対してのTCNT2のように)。

表63.の定義は本章を通して広範囲に渡って使われます。

表63. 用語定義

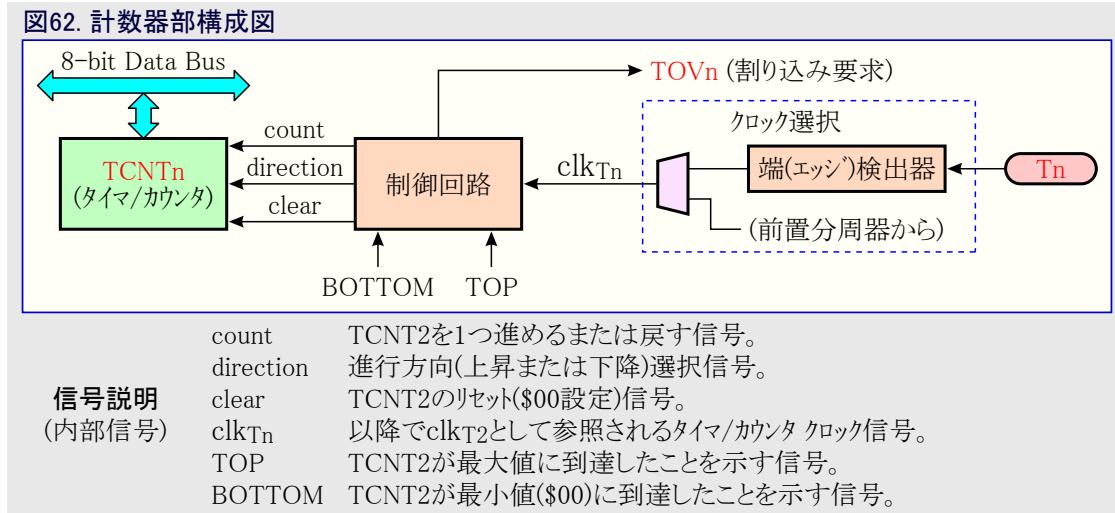
| 用語     | 意味   |
|--------|--|
| BOTTOM | タイマ/カウンタが\$00に到達した時。   |
| MAX    | タイマ/カウンタが\$FF(255)に到達した時。                                    |
| TOP    | タイマ/カウンタが指定された固定値(\$FF)またはOCR2値に到達した時。この指定(TOP)値は動作種別に依存します。 |

## タイマ/カウンタのクロック

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(TCCR2)に配置されたクロック選択(CS22~0)ビットによって制御されるクロック選択論理回路で選ばれます。クロック元と前置分周器の詳細については97頁の「タイマ/カウンタ1、タイマ/カウンタ2、タイマ/カウンタ3の前置分周器」をご覧ください。

## 計数器部

8ビットタイマ/カウンタの主な部分は設定可能な双方向カウンタ部です。図62.は、このカウンタとその周辺環境の構成図を示します。



使った動作種別に依って、カウンタは各タイマ/カウンタ クロック(clk<sub>T2</sub>)で解除(\$00)、増加(+1)、または減少(-1)されます。clk<sub>T2</sub>はクロック選択(CS22~0)ビットによって選ばれた内部または外部のクロック元から生成できます。クロック元が選ばれない(CS22~0=000)と、タイマ/カウンタは停止されます。けれどもTCNT2値はタイマ/カウンタ クロック(clk<sub>T2</sub>)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(上位優先順位を持ちます)。

計数順序(方法)はタイマ/カウンタ制御レジスタ(TCCR2)に配置された波形生成種別(WGM21,0)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OC2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関しては103頁の「動作種別」をご覧ください。

タイマ/カウンタ溢れ(TOV2)フラグはWGM21,0ビットによって選ばれた動作種別に従って設定(1)されます。TOV2はCPU割り込み発生に使えます。

## 比較出力部

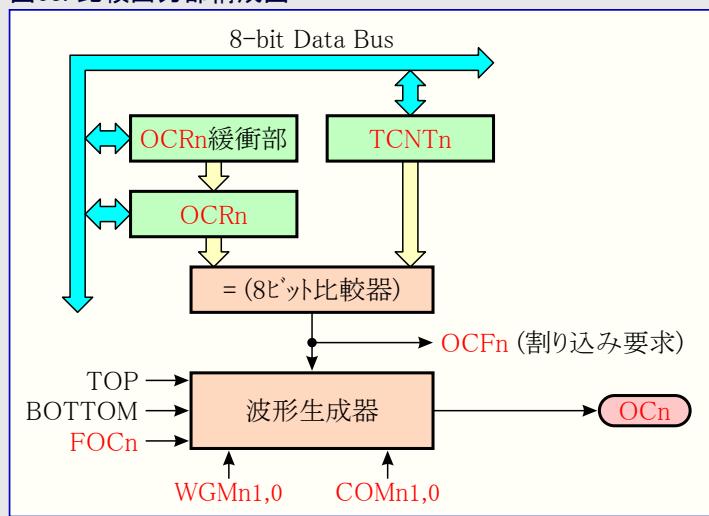
この8ビット比較器はTCNT2と比較レジスタ(OCR2)を継続的に比較します。TCNT2とOCR2が等しければ比較器は一致を指示します。この一致は次のタイマ/カウンタクロック周期で比較割り込み要求フラグ(OCF2)を設定(1)します。許可(I=1, OCIE2=1)なら、この比較割り込み要求フラグは比較割り込みを発生します。OCF2は割り込みが実行されると自動的に解除(0)されます。代わりにOCF2はこのビット位置に論理1を書くことによってソフトウェアでも解除(0)できます。波形生成器は波形生成種別(WGM21,0)ビットと比較出力選択(COM21,0)ビットによって設定された動作種別に従った出力を生成するのにこの一致信号を使います。MAXとBOTTOM信号は動作種別(103頁の「動作種別」参照)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。

図63.は比較出力部の構成図を示します。

OCR2はパルス幅変調(PWM)のどれかを使う時に2重緩衝化されます。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OCR2レジスタ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。

OCR2のアクセスは複雑なように思えますが決してそんなことはありません。2重緩衝動作が許可されるとCPUはOCR2緩衝部をアクセスし、禁止されるとOCR2レジスタを直接アクセスします。

図63. 比較出力部構成図



## 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(FOC2)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(OCF2)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、OC2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(COM21,0ビット設定がOC2ピンの設定(1)、解除(0)、1/0交互のどれかを定義)。

## TCNT2書き込みによる比較一致妨害

TCNT2への全てのCPU書き込みは、例えタイマ/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、TCNT2と同じ値に初期化されることをOCR2に許します。

## 比較一致部の使用

どの動作種別でのTCNT2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、比較出力部を使う場合、TCNT2を変更する時に危険を伴います。TCNT2に書かれた値がOCR2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。同様にタイマ/カウンタが下降計数のとき、BOTTOMに等しいTCNT2値を書いてはいけません。

OC2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。OC2値を設定する一番簡単な方法は標準動作で強制変更(FOC2)スローブビットを使うことです。波形生成動作種別間を変更する時であっても、OC2(内部)レジスタはその値を保ちます。

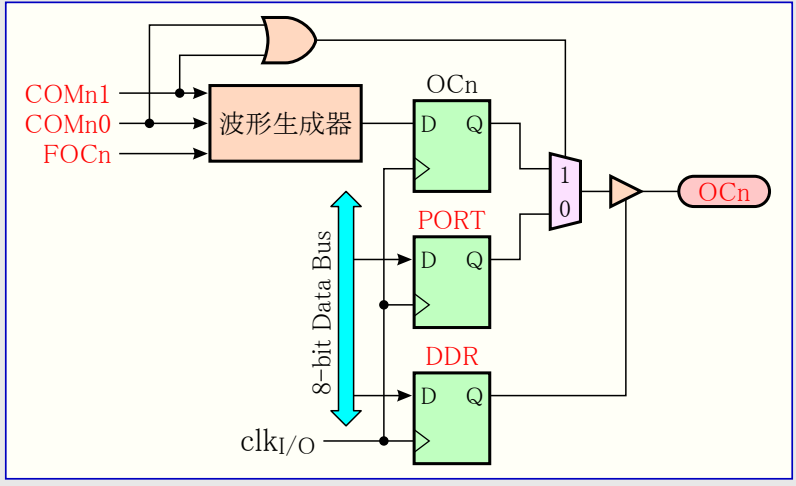
比較出力選択(COM21,0)ビットが比較値(OCR2)と共に2重緩衝されないことに気付いてください。COM21,0ビットの変更は直ちに有効となります。



## 比較一致出力部

比較出力選択(COM21,0)ビットは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(OC2)状態の定義にCOM21,0ビットを使います。またCOM21,0ビットはOC2ピン出力元を制御します。図64.はCOM21,0ビット設定によって影響を及ぼされる論理回路の単純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。COM21,0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(PORTとDDR)の部分だけが示されます。OC2の状態を参照するとき、その参照はOC2ピンでなく内部OC2レジスタに対してです。システムリセットが起ると、OC2レジスタは'0'にリセットされます。

図64. 比較一致出力回路図



COM21,0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(OC2)によって無効にされます。けれどもOC2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。OC2ピンに対するポート方向レジスタのビット(DDR\_OC2)はOC2値がピンで見えるのに先立って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OC2状態の初期化を許します。いくつかのCOM21,0ビット設定が或る種の動作種別に対して予約されることに注意してください。107頁の「8ビットタイマ/カウンタ用レジスタ」をご覧ください。

### 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COM21,0ビットを違うふうに使います。全動作種別に対してCOM21,0=00設定は次の比較一致で実行すべきOC2レジスタの動きがないことを波形生成器へ告げます。非PWM動作での比較出力動作については107頁の表65を参照してください。高速PWM動作については107頁の表66、位相基準PWMについては107頁の表67を参照してください。

COM21,0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(FOC2)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。

## 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(WGM21,0)ビットと比較出力選択(COM21,0)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。COM21,0ビットは生成されるPWM出力が反転されるべきか否か(反転または非反転PWM)どちらかを制御します。非PWM動作に対するCOM21,0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(102頁の「比較一致出力部」をご覧ください)。

タイミング情報の詳細については106頁の「タイマ/カウンタ2のタイミング」を参照してください(訳注:原文中の図番号省略)。

## 標準動作

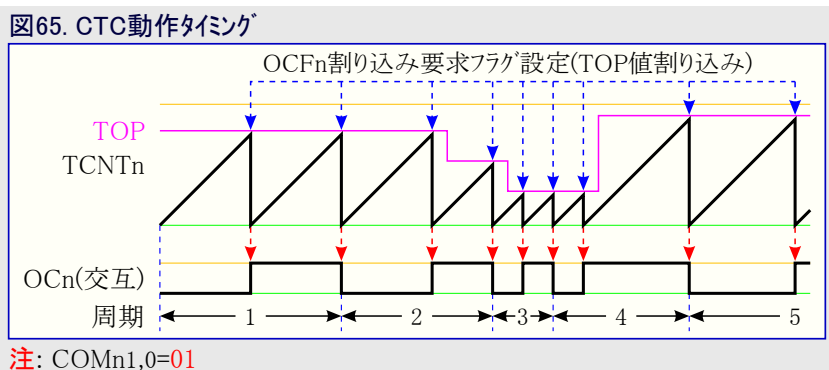
最も単純な動作種別が標準動作(WGM21,0=00)です。本動作種別での計数方向は常に上昇(+)で、カウンタ解除は実行されません。カウンタは8ビット最大値(TOP=\$FF)を通過すると単に範囲を超え、そして\$00(BOTTOM)から再び始めます。通常動作でのタイマ/カウンタ溢れ(TOV2)フラグはTCNT2が\$00になる時と同じタイマ/カウンタクロック周期で設定(1)されます。この場合のTOV2フラグは設定(1)のみで解除(0)されないことを除いて第9ビットのようになります。けれどもTOV2フラグを自動的に解除(0)するタイマ/カウンタ2溢れ割り込みと組み合わせたタイマ/カウンタの分解能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。

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。

## 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(WGM21,0=10)ではOCR2がカウンタ分解能を操作するのに使われます。CTC動作ではカウンタ(TCNT2)値がOCR2と一致すると、カウンタは\$00に解除されます。OCR2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図65.で示されます。カウンタ(TCNT2)値はTCNT2とOCR2間で比較一致が起こるまで増加し、そしてその後にカウンタ(TCNT2)は解除(\$00)されます。



OCF2フラグを使うことにより、タイマ/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。けれども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BOTTOMと近い値にTOPを変更することは、CTC動作が2重緩衝機能を持たないために注意して行わなければなりません。OCR2に書かれた新しい値がTCNT2の現在値よりも低い(小さい)場合、タイマ/カウンタは(その回の)比較一致を失います。その後のカウンタは比較一致が起こるのに先立って最大値(\$FF)へ、そして次に\$00から始める計数をしなければならないでしょう。

CTC動作で波形出力を生成するため、OC2出力は比較出力選択(COM21,0)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り替えに設定できます。OC2値はそのピンに対するデータ方向が出力(DDR\_OC2=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OCR2が0(\$00)に設定される時に $f_{OC2} = f_{clk\_I/O} / 2$ 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。

$$f_{OCn} = \frac{f_{clk\_I/O}}{2 \times N \times (1 + OCRn)}$$

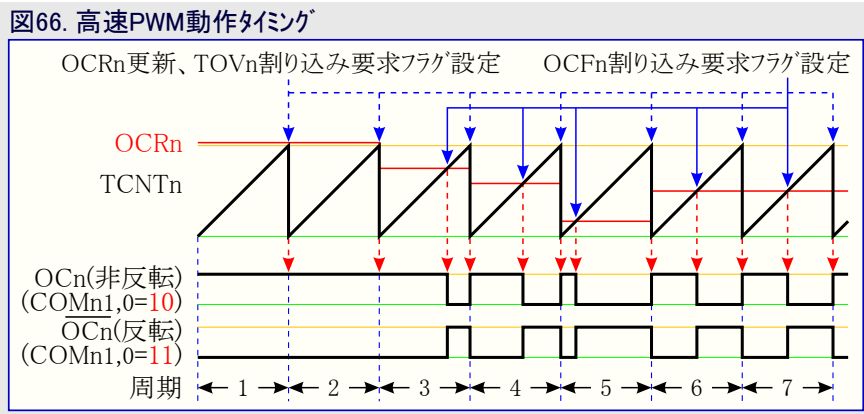
変数Nは前置分周数(1,8,64,256,1024)を表します。

標準動作と同じように、タイマ/カウンタ溢れ(TOV2)フラグはカウンタがMAXから\$00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。

## 高速PWM動作

高速パルス幅変調(PWM)動作(WGM21,0=11)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MAX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(COM21,0=10)での比較出力(OC2)はTCNT2とOCR2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(COM21,0=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PWM動作より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動作でのタイマ/カウンタはタイマ/カウンタ値がMAX値と一致するまで増加されます。そしてタイマ/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(\$00)されます。高速PWM動作のタイミング図は図66.で示されます。TCNT2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。赤細線はOCR2値を示し、TCNT2値との交点(接点)がTCNT2とOCR2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(OCF2)はOCR2=MAXを除いて比較一致が起こると設定(1)されます(訳注:共通性のため本行追加)。



タイマ/カウンタ溢れ(TOV2)フラグはカウンタがMAXに到達する時毎に設定(1)されます。割り込みが許可されるなら、その割り込み処理ルーチンは比較値を更新するのに使えます。

高速PWM動作での比較部はOC2ピンでのPWM波形の生成を許します。COM21,0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COM21,0を'11'に設定することで生成できます(107頁の表66をご覧ください)。実際のOC2値はポートピンに対するデータ方向(DDR\_OC2)が出力として設定される場合にだけ見えるでしょう。PWM波形はTCNT2とOCR2間の比較一致でOC2(内部)レジスタを設定(1)(または解除(0))と、カウンタが解除(\$00、MAXからBOTTOMへ変更)されるタイマ/カウンタクロック周期でOC2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。

$$f_{OCnPWM} = \frac{f_{clk\_I/O}}{N \times 256}$$

変数Nは前置分周数(1,8,64,256,1024)を表します。

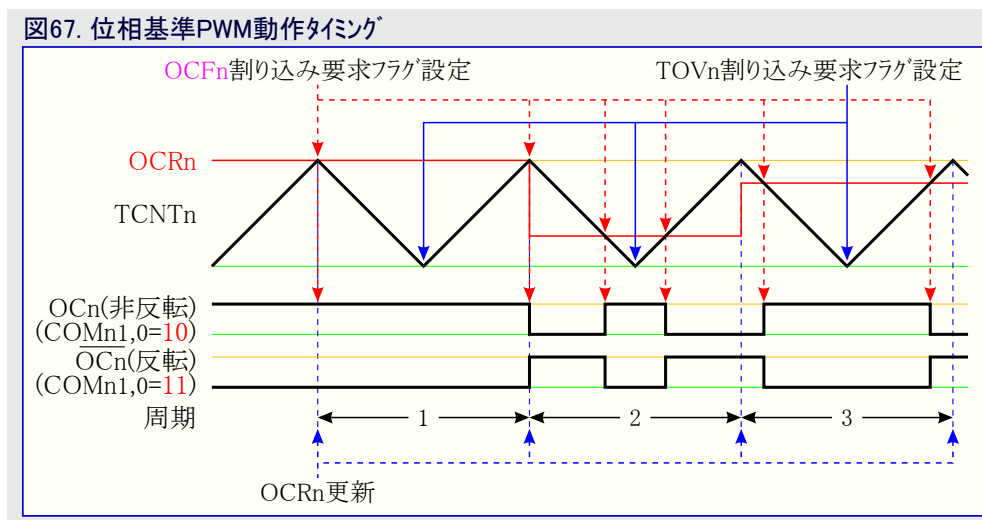
OCR2の両端値は高速PWM動作でPWM波形出力を生成する時の特別な場合に当たります。OCR2がBOTTOM(\$00)と等しく設定されると、出力はMAX+1 タイマ/カウンタクロック周期毎の狭いスパイク(パルス)になるでしょう。OCR2がMAXに等しく設定されると、(COM21,0ビットによって設定される出力極性に依存して)定常的なLowまたはHigh出力に終わるでしょう。

(訳注:以下は本動作種別がTOP=MAX(\$FF)で固定のため基本的には不適切です。) 高速PWM動作で(デューティ比50%)周波数の波形出力は比較一致毎に論理反転するOC2設定(COM21,0=01)によって達成できます。生成された波形はOCR2が0(\$00)に設定される時に $f_{OC2} = f_{clk\_I/O} / 2$ の最大周波数でしょう。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OC2交互出力(COM21,0=01)と同じです。

## 位相基準PWM動作

位相基準パルス幅変調(PWM)動作(WGM21,0=01)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)動作を基準とします。カウンタはBOTTOMからMAXへそして次にMAXからBOTTOMへを繰り返し計数します。非反転比較出力動作(COM21,0=10)での比較出力(OC2)は上昇計数中のTCNT2とOCR2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(COM21,0=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)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ビットに固定されます。位相基準PWM動作でのカウンタはカウンタ値がMAXと一致するまで増加されます。カウンタはMAXに達すると計数方向を変更します。このTCNT2値は1タイマ/カウンタクロック周期MAXと等しくなります。位相基準PWM動作のタイミング図は図67.で示されます。TCNT2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OCR2値を示し、TCNT2値との交点(接点)がTCNT2とOCR2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)。



タイマ/カウンタ溢れ(TOV2)フラグはタイマ/カウンタがBOTTOMに到達する時毎に設定(1)されます。この割り込み要求フラグはカウンタがBOTTOM値に到達する毎に割り込みを発生するために使えます。

位相基準PWM動作での比較部はOC2ピンでのPWM波形の生成を許します。COM21,0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COM21,0ビットを'11'に設定することで生成できます(107頁の表67.をご覧ください)。実際のOC2値はそのポートピンに対するデータ方向(DDR\_OC2)が出力として設定される場合だけ見えるでしょう。PWM波形はカウンタが増加する時のTCNT2とOCR2間の比較一致でOC2(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TCNT2とOCR2間の比較一致でOC2レジスタを解除(0)(または設定(1))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。

$$f_{OCnPCPWM} = \frac{f_{clk\_I/O}}{N \times 510}$$

変数Nは前置分周数(1,8,64,256,1024)を表します。

OCR2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あたります。非反転PWM動作ではOCR2がBOTTOM(\$00)に等しく設定されると出力は定常的なLow、MAX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。

図67.の第2周期のその出発点に於いて、例え比較一致がないとしても、OCnにはHighからLowへの遷移があります。この遷移点はBOTTOMを挟む対称を保証するためです。比較一致なしに遷移を生ずるのは2つの場合です。

- 図67.のようにOCR2はMAXからその値を変更します。OCR2値がMAXのとき、OCnピン値は下降計数での比較一致の結果と同じです(訳補: L→H、直前がHのため、常にH)。BOTTOMを挟む対称を保証するため、(変更直後の)MAX(位置)でのOCn値は上昇計数での比較一致の結果(H→L)と一致しなければなりません。
- タイマ/カウンタがOCR2値よりも高い値から数え始め、そしてその理屈のために比較一致、それ故上昇途中で起こされるであろうOCnの変更を逃します。(訳補: 従って上記同様、MAX位置で(直前がHならば)H→L遷移が生じます。)

## 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、従って以下の図でタイマ/カウンタクロック( $clk_{T2}$ )が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)される時の情報を含みます。図68は基本的なタイマ/カウンタ動作についてのタイミング データを含みます。この図は位相基準 PWM動作以外の全ての動作種別でのMAX値近辺の計数の流れを示します。

図68. 前置分周なし(1/1)のタイマ/カウンタ タイミング

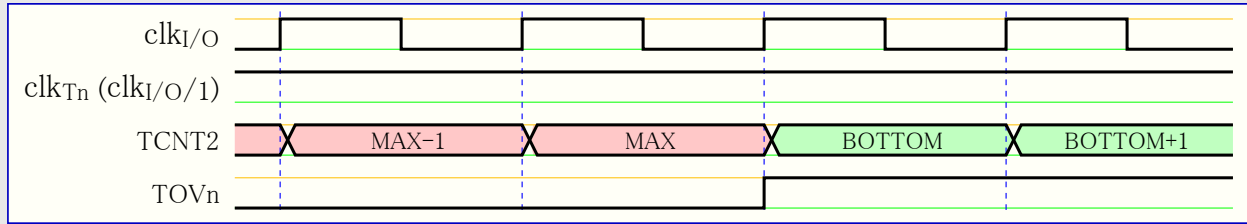


図69. は同じタイミング データを示しますが、前置分周器が許可されています。

図69. 前置分周器( $f_{clk_{I/O}/8}$ )のタイマ/カウンタ タイミング

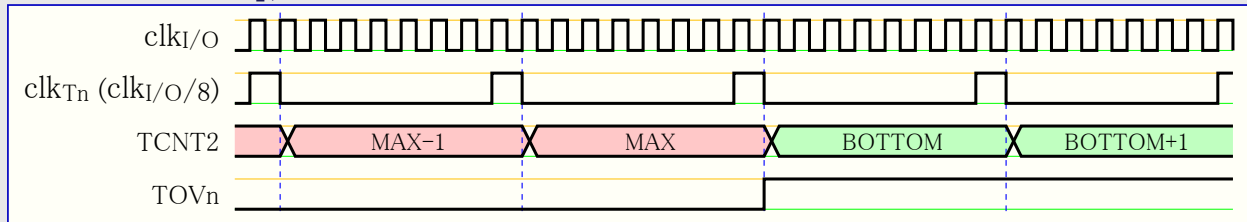


図70. はCTC動作を除く全ての動作種別でのOCF2の設定を示します。

図70. 前置分周器( $f_{clk_{I/O}/8}$ )のタイマ/カウンタ、OCF2設定 タイミング

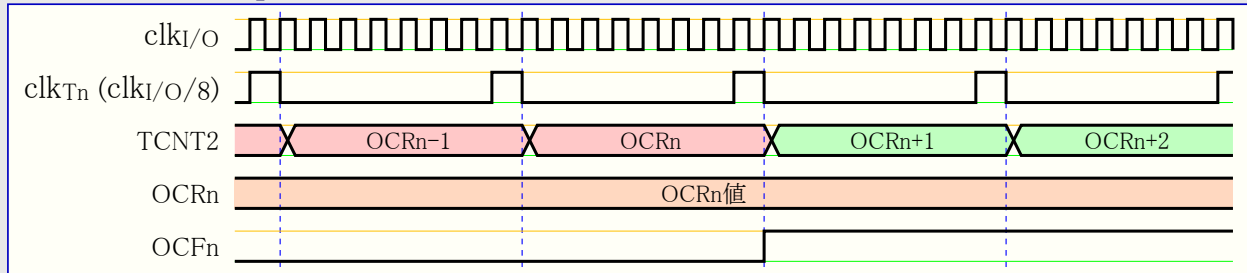
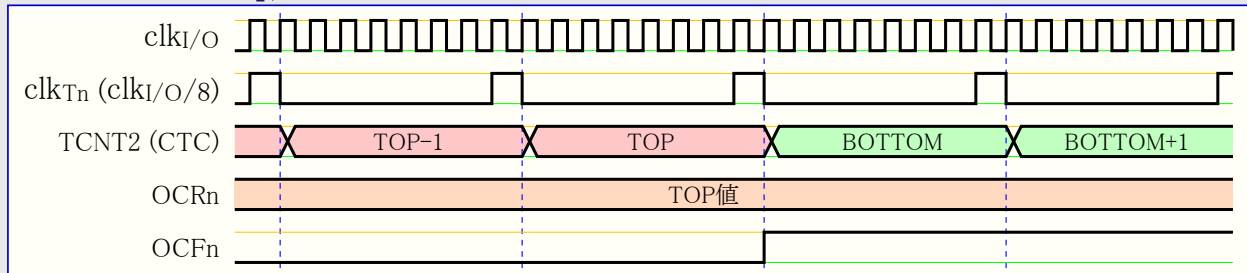


図71. はCTC動作でのTCNT2の解除とOCF2の設定を示します。

図71. 前置分周器( $f_{clk_{I/O}/8}$ )のタイマ/カウンタ、OCF2設定 タイミング





## 8ビット タイマ/カウンタ2用レジスタ

### タイマ/カウンタ2制御レジスタ (Timer/Counter2 Control Register) TCCR2

| ビット         | 7    | 6     | 5     | 4     | 3     | 2    | 1    | 0    |       |
|-------------|------|-------|-------|-------|-------|------|------|------|-------|
| \$25 (\$45) | FOC2 | WGM20 | COM21 | COM20 | WGM21 | CS22 | CS21 | CS20 | TCCR2 |
| Read/Write  | W    | R/W   | R/W   | R/W   | R/W   | R/W  | R/W  | R/W  |       |
| 初期値         | 0    | 0     | 0     | 0     | 0     | 0    | 0    | 0    |       |

#### • ビット7 – FOC2 : OC2強制変更 (Force Output Compare 2)

FOC2ビットはWGM21,0ビットが非PWM動作を指示する時だけ有効です。けれども将来のデバイスとの共通性を保証するため、PWM動作で扱う時にTCCR2が書かれる場合、このビットは0に設定されなければなりません。FOC2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。OC2出力はCOM21,0ビット設定に従って変更されます。FOC2ビットがストローブとして実行されることに注意してください。従って強制した比較の効果を決めるのはCOM21,0ビットに存在する値です。

FOC2ストローブは何れの割り込みの生成もTOPとしてOCR2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(\$00)も行いません。

FOC2ビットは常に0として読みます。

#### • ビット3,6 – WGM21,WGM20 : 波形生成種別 (Waveform Generation Mode bit 1 and 0)

これらのビットはカウンタの計数順序(方向)、最大(TOP)カウンタ値の供給元、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、比較一致タイマ/カウンタ解除(CTC)動作と、2形式のパルス幅変調(PWM)動作です。表64.と103頁の「動作種別」をご覧ください。

表64. 波形生成種別選択

| 番号 | WGM21 (CTC2) | WGM20 (PWM2) | タイマ/カウンタ動作種別          | TOP値 | OCR2更新時 | TOV2設定時 |
|----|--------------|--------------|-----------------------|------|---------|---------|
| 0  | 0            | 0            | 標準動作                  | \$FF | 即値      | MAX     |
| 1  | 0            | 1            | 位相基準PWM動作             | \$FF | TOP     | BOTTOM  |
| 2  | 1            | 0            | 比較一致タイマ/カウンタ解除(CTC)動作 | OCR2 | 即値      | MAX     |
| 3  | 1            | 1            | 高速PWM動作               | \$FF | BOTTOM  | MAX     |

注: CTC2とPWM2ビット定義名は旧名です。WGM21,0定義名を使ってください。しかし、これらのビットの機能と位置は旧版のタイマ/カウンタと一致します。

#### • ビット5,4 – COM21,COM20 : 比較2出力選択 (Compare Match 2 Output Mode bit 1 and 0)

これらのビットはOC2比較出力ピンの動作を制御します。COM21,0ビットの1つまたは両方が1を書かれると、OC2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、OC2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。

OC2がピンに接続されるとき、COM21,0ビットの機能はWGM21,0ビット設定に依存します。

表65.はWGM21,0ビットが標準動作またはCTC動作(つまりPWM以外)に設定される時のCOM21,0ビット機能を示します。

表66.はWGM21,0ビットが高速PWM動作に設定される時のCOM21,0ビットの機能を示します。

表67.はWGM21,0ビットが位相基準PWM動作に設定される時のCOM21,0ビットの機能を示します。

表65. 非PWM動作での比較出力選択

| COM21 | COM20 | 意味                   |
|-------|-------|----------------------|
| 0     | 0     | 標準ポート動作 (OC2切断)      |
| 0     | 1     | 比較一致でOC2ピントグル(交互)出力  |
| 1     | 0     | 比較一致でOC2ピン Lowレベル出力  |
| 1     | 1     | 比較一致でOC2ピン Highレベル出力 |

表66. 高速PWM動作での比較出力選択

| COM21 | COM20 | 意味                                    |
|-------|-------|---------------------------------------|
| 0     | 0     | 標準ポート動作 (OC2切断)                       |
| 0     | 1     | (予約)                                  |
| 1     | 0     | 比較一致でLow、BOTTOMでHighをOC2ピンへ出力 (非反転動作) |
| 1     | 1     | 比較一致でHigh、BOTTOMでLowをOC2ピンへ出力 (反転動作)  |

注: COM21が設定(1)され、OCR2がTOPと等しいとき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)や解除(0)は行われます。より多くの詳細については104頁の「高速PWM動作」をご覧ください。

表67. 位相基準PWM動作での比較出力選択

| COM21 | COM20 | 意味                                      |
|-------|-------|---|
| 0     | 0     | 標準ポート動作 (OC2切断)                         |
| 0     | 1     | (予約)                                    |
| 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OC2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OC2ピンへ出力 |

注: COM21が設定(1)され、OCR2がTOPと等しいとき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、TOPでの設定(1)や解除(0)は行われます。より多くの詳細については105頁の「位相基準PWM動作」をご覧ください。

•ビット2~0 - CS22,CS21,CS20 : クロック選択2 (Clock Select2, bit 2,1 and 0)

この3つのクロック選択ビットはタイマ/カウンタ(TCNT2)によって使われるクロック元を選びます。

表68. タイマ/カウンタ2入力クロック選択

| CS22 | CS21 | CS20 | 意味                                |
|------|------|------|-----------------------------------|
| 0    | 0    | 0    | 停止 (タイマ/カウンタ2動作停止)                |
| 0    | 0    | 1    | clk <sub>I/O</sub> (前置分周なし)       |
| 0    | 1    | 0    | clk <sub>I/O</sub> /8 (8分周)       |
| 0    | 1    | 1    | clk <sub>I/O</sub> /64 (64分周)     |
| 1    | 0    | 0    | clk <sub>I/O</sub> /256 (256分周)   |
| 1    | 0    | 1    | clk <sub>I/O</sub> /1024 (1024分周) |
| 1    | 1    | 0    | T2ピンの下降端 (外部クロック)                 |
| 1    | 1    | 1    | T2ピンの上昇端 (外部クロック)                 |

外部ピン(クロック)動作がタイマ/カウンタ2に対して使われる場合、例えT2ピンが出力として設定されても、T2ピンの遷移はカウンタをクロック駆動します。この特徴がソフトウェアに計数制御を許します。

タイマ/カウンタ2 (Timer/Counter2) TCNT2

| ビット         | 7     | 6   | 5   | 4   | 3   | 2   | 1   | 0     |       |
|-------------|-------|-----|-----|-----|-----|-----|-----|-------|-------|
| \$24 (\$44) | (MSB) |     |     |     |     |     |     | (LSB) | TCNT2 |
| Read/Write  | R/W   | R/W | R/W | R/W | R/W | R/W | R/W | R/W   |       |
| 初期値         | 0     | 0   | 0   | 0   | 0   | 0   | 0   | 0     |       |

このタイマ/カウンタレジスタは読み書き両方の操作について、タイマ/カウンタ部の8ビットカウンタに直接アクセスします。TCNT2への書き込みは次のタイマ/カウンタクロックでの比較一致を妨害(除去)します。カウンタが走行中にカウンタ(TCNT2)を変更することはTCNT2とOCR2間の比較一致消失の危険を誘発します。

タイマ/カウンタ2 比較レジスタ (Timer/Counter2 Output Compare Register) OCR2

| ビット         | 7     | 6   | 5   | 4   | 3   | 2   | 1   | 0     |      |
|-------------|-------|-----|-----|-----|-----|-----|-----|-------|------|
| \$23 (\$43) | (MSB) |     |     |     |     |     |     | (LSB) | OCR2 |
| Read/Write  | R/W   | R/W | R/W | R/W | R/W | R/W | R/W | R/W   |      |
| 初期値         | 0     | 0   | 0   | 0   | 0   | 0   | 0   | 0     |      |

この比較レジスタは継続的にカウンタ(TCNT2)値と比較される8ビットの値を含みます。一致は比較一致割り込みやOC2ピンでの波形出力を生成するのに使えます。

## タイマ/カウンタ割り込み許可レジスタ (Timer/Counter Interrupt Mask Register) TIMSK

| ビット         | 7     | 6     | 5      | 4      | 3      | 2     | 1     | 0     |       |
|-------------|-------|-------|--------|--------|--------|-------|-------|-------|-------|
| \$37 (\$57) | OCIE2 | TOIE2 | TICIE1 | OCIE1A | OCIE1B | TOIE1 | OCIE0 | TOIE0 | TIMSK |
| Read/Write  | R/W   | R/W   | R/W    | R/W    | R/W    | R/W   | R/W   | R/W   |       |
| 初期値         | 0     | 0     | 0      | 0      | 0      | 0     | 0     | 0     |       |

## • ビット7 – OCIE2 : タイマ/カウンタ2比較割り込み許可 (Timer/Counter2 Output Compare Match Interrupt Enable)

OCIE2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2比較一致割り込みが許可されます。タイマ/カウンタ2で比較一致が起こる、例えば、タイマ/カウンタ割り込み要求フラグレジスタ(TIFR)で比較2割り込み要求フラグ(OCF2)が設定(1)されると、対応する割り込みが実行されます。

## • ビット6 – TOIE2 : タイマ/カウンタ2溢れ割り込み許可 (Timer/Counter2 Overflow Interrupt Enable)

TOIE2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2溢れ割り込みが許可されます。タイマ/カウンタ2溢れが起こる、例えば、タイマ/カウンタ割り込み要求フラグレジスタ(TIFR)でタイマ/カウンタ2溢れ割り込み要求(TOV2)フラグが設定(1)されると、対応する割り込みが実行されます。

## タイマ/カウンタ割り込み要求フラグレジスタ (Timer/Counter Interrupt Flag Register) TIFR

| ビット         | 7    | 6    | 5    | 4     | 3     | 2    | 1    | 0    |      |
|-------------|------|------|------|-------|-------|------|------|------|------|
| \$36 (\$56) | OCF2 | TOV2 | ICF1 | OCF1A | OCF1B | TOV1 | OCF0 | TOV0 | TIFR |
| Read/Write  | R/W  | R/W  | R/W  | R/W   | R/W   | R/W  | R/W  | R/W  |      |
| 初期値         | 0    | 0    | 0    | 0     | 0     | 0    | 0    | 0    |      |

## • ビット7 – OCF2 : タイマ/カウンタ2比較割り込み要求フラグ (Timer/Counter2, Output Compare Match Flag)

OCF2ビットは比較一致がタイマ/カウンタ(TCNT2)と比較レジスタ(OCR2)間で起こる時に設定(1)されます。対応する割り込み処理ベクタを実行する時にOCF2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OCF2は解除(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、タイマ/カウンタ割り込み許可レジスタ(TIMSK)のタイマ/カウンタ2比較一致割り込み許可(OCIE2)ビット、OCF2が設定(1)されると、タイマ/カウンタ2比較一致割り込みが実行されます。

## • ビット6 – TOV2 : タイマ/カウンタ2溢れ割り込み要求フラグ (Timer/Counter2 Overflow Flag)

TOV2ビットはタイマ/カウンタ(TCNT2)溢れが起こる時に設定(1)されます。対応する割り込み処理ベクタを実行する時にTOV2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TOV2は解除(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、タイマ/カウンタ割り込み許可レジスタ(TIMSK)のタイマ/カウンタ2溢れ割り込み許可(TOIE2)ビット、TOV2が設定(1)されると、タイマ/カウンタ2溢れ割り込みが実行されます。PWM動作ではタイマ/カウンタ2が\$00で計数方向を変える時にこのビットが設定(1)されます。

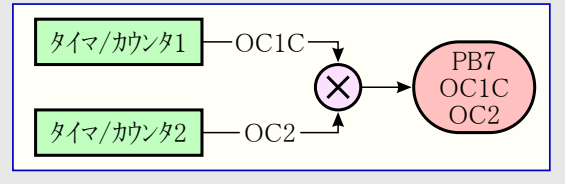
## 比較出力変調器 (OCM1C2)

### 概要

比較出力変調器(OCM)は搬送波を変調した波形の生成を許します。この変調器は16ビットタイマ/カウンタ1の比較C出力部と8ビットタイマ/カウンタ2の比較出力部からの出力を使います。これらのタイマ/カウンタについてのより多くの詳細に関しては、74頁の「16ビットタイマ/カウンタ(タイマ/カウンタ1とタイマ/カウンタ3)」と99頁の「8ビットタイマ/カウンタ2(PWM付き)」をご覧ください。この特徴(機能)がATmega103互換動作で利用できないことに注意してください。

この変調器が許可されると、構成図(図72.)で示されるように2つの比較出力が共に変調されます。

図72. 比較出力変調器構成図

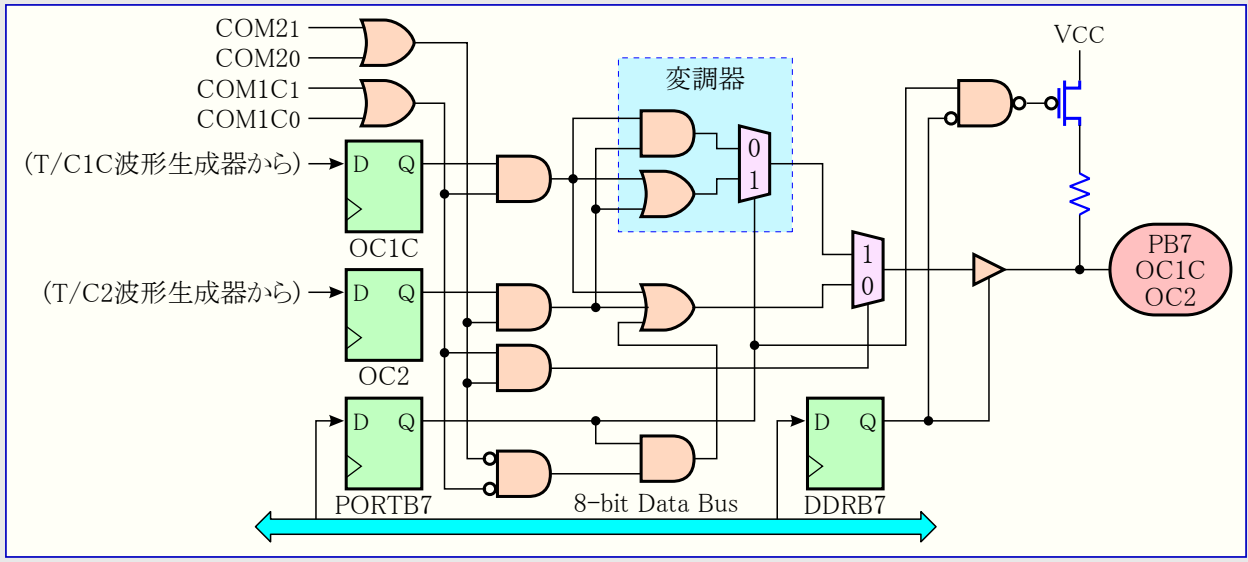


### 説明

比較1C出力部と比較2出力部は出力に関してPB7ポートピンを共用します。比較出力部の出力(OC1CとOC2)はこれらの1つが許可(換言すると、COMnx1,0≠00)にされると、標準PORTB7レジスタを無効にします。OC1CとOC2の両方が同時に許可されると、この変調器が自動的に許可されます。

この変調器の機能等価回路図は図73.で示されます。この図はタイマ/カウンタ部とポートBのビット7ピン出力駆動回路部分を含みます。

図73. 比較出力変調器等価回路図

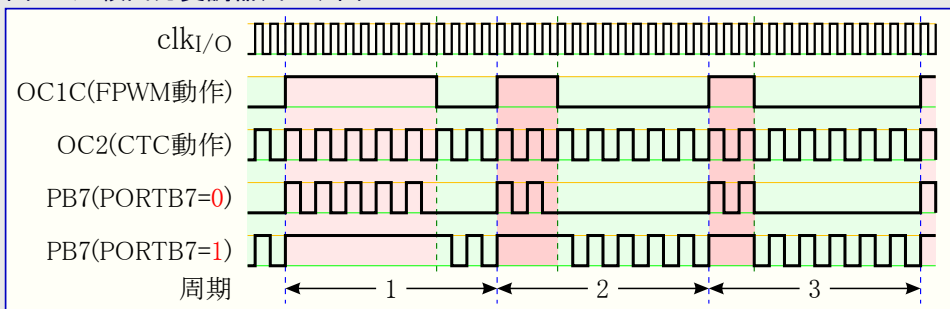


変調器が許可されると、変調形式(論理ANDまたはOR)がPORTB7レジスタによって選ばれます。COMnx1,0ビット設定に拘らず、DDB7がポートの方向を制御することに注意してください。

### タイミング例

図74.は変調器の動作を図解します。この例でのタイマ/カウンタ1は(非反転)高速PWM動作に設定され、タイマ/カウンタ2は比較交互出力(COM21,0=01)のCTC波形動作を使います。

図74. 比較出力変調器タイミング図



本例ではタイマ/カウンタ2が搬送波を供給し、一方変調する信号(変調波)はタイマ/カウンタ1の比較C出力によって生成されます。

PWM信号(OC1C)の分解能は変調によって減らされます。この減少係数は搬送波(OC2)1周期のシステムクロック周期数と等価です。この例の分解能は係数2で減らされます(即ち1/2)。この減少理由は図74.のPORTB7=0時のPB7出力の第2周期と第3周期で図解されます。第2周期のHigh区間は第3周期のHigh区間よりも1周期長いけれども、PB7出力の結果は両周期で等しくなります。

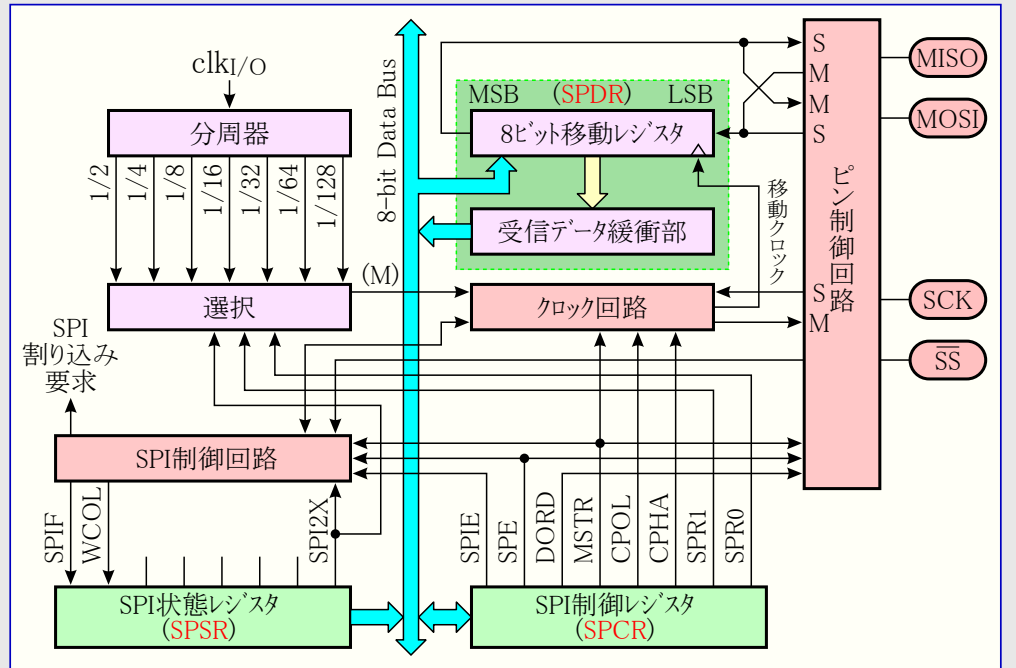
## 直列周辺インターフェース (SPI: Serial Peripheral Interface)

直列周辺インターフェースはAtmel® AVR® ATmega128と様々なAVRデバイスや周辺デバイス間の高速同期データ転送を許します。ATmega128のSPIは次の特徴を含みます。

- 全二重3線同期データ転送
- 主装置/従装置動作
- LSB/MSB先行データ転送
- 設定変更可能な7つのビット速度
- 送信完了割り込み要求フラグ
- 送信上書きフラグ保護
- アイドル動作からの起動
- 倍速(CK/2)主装置SPI動作

SPIでの主装置と従装置のCPU間相互接続は図76.で示されます。このシステムは2つの移動レジスタと主装置クロック発生器から成ります。SPI主装置は希望した従装置のSS(従装置選択)ピンをLowへ引き込む時に一群の通信を始めます。主装置と従装置は各々の移動レジスタで送るべきデータを用意し、主装置はデータを交換するのに必要なクロックパルスをSCK信号線に生成します。データは常にMOSI(Master Out Slave In)信号線を主装置から従装置へ、MISO(Master In Slave Out)信号線を従装置から主装置へ移動されます。各データパケット後、主装置はSS(従装置選択)ピンをHighへ引き上げることによって従装置と同期を取ります。

図75. SPI構成図



注: SPIピン配置については2頁の「ピン配置」と46頁の表30.を参照してください。

主装置として設定されると、SPIインターフェースにはSS信号線の自動制御がありません。これは通信を開始できるのに先立って使用者ソフトウェアによって操作されなければなりません。これが行われると、SPIデータレジスタ(SPDR)へのバイト書き込みがSPIクロック発生器を開始し、ハードウェアが従装置内へ8ビットを移動します。1バイトの移動後、SPIクロック発生器は停止し、SPI状態レジスタ(SPSR)の転送完了フラグ(SPIF)を設定(1)します。SPI制御レジスタ(SPCR)でSPI割り込み許可(SPIE)ビットが設定(1)されていれば割り込みが要求されます。主装置はSPDR内へ次バイトを書くことによって次バイトの移動を継続、またはSS(従装置選択)信号線をHighへ引き上げることによってパケットの終了を指示することができます。最後の到着バイトはその後の使用のため、緩衝レジスタ内に保持されます。

従装置として設定されると、SPIインターフェースはSSピンがHighに駆動される限り、MISOをHi-Zにした休止状態に留まります。この状態でプログラムはSPIデータレジスタ(SPDR)の内容を更新できますが、そのデータはSSピンがLowに駆動されるまでSCKピンでの到着クロックパルスによって移動出力されません。1バイトが完全に移動されてしまうと転送完了フラグ(SPIF)が設定(1)されます。SPCRでSPI割り込み許可(SPIE)ビットが設定(1)されていれば割り込みが要求されます。従装置は受信データを読む前にSPDR内へ送られるべき次のデータの配置を続けられます。最後の到着バイトはその後の使用のため、緩衝レジスタ内に保持されます。

このシステムは送信側で単一緩衝、受信側で2重緩衝です。これは一連の移動全体が完了される前に送信されるべきバイトがSPIデータレジスタ(SPDR)へ書けないことを意味します。しかし、データを受信するとき、次のデータが完全に移動入力される前に受信したデータがSPIデータレジスタ(SPDR)から読まなければなりません。さもなければ最初のバイトは失われます。

SPI従装置動作では制御論理回路がSCKピンの到着信号を採取します。このクロック信号の正しい採取を保証するため、LowとHighの最小周期は以下であるべきです。

- Low周期 : 2 CPUクロック周期より長い
- High周期 : 2 CPUクロック周期より長い

SPIが許可されると、MOSI, MISO, SCK, SSピンのデータ方向は表69.に従って無視されます。自動的なポート無視のより多くの詳細については、44頁の「交換ポート機能」を参照してください。

図76. SPI 主装置/従装置の接続

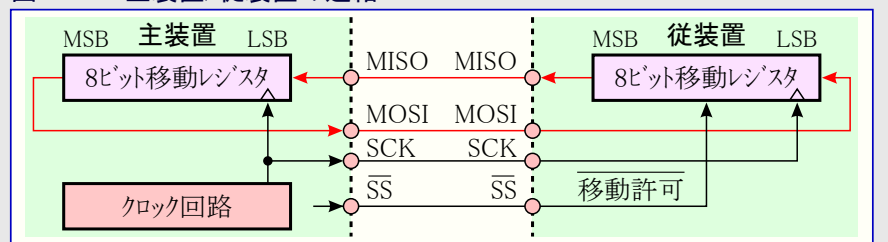


表69. SPIピン方向規定

| ピン名  | 主装置時の方向規定           | 従装置時の方向規定           |
|------|---------------------|---------------------|
| SCK  | ポートB方向レジスタ(DDRB)の指定 | 入力                  |
| MISO | 入力                  | ポートB方向レジスタ(DDRB)の指定 |
| MOSI | ポートB方向レジスタ(DDRB)の指定 | 入力                  |
| SS   | ポートB方向レジスタ(DDRB)の指定 | 入力                  |

注: 使用者定義SPIピンの方向定義方法の詳細記述については46頁の「ポートBの交換機能」を参照してください。



次のコード例は主装置としてSPIを初期化する方法と簡単な送信を実行する方法を示します。例でのDDR\_SPIはSPIピンを制御する実際のポート方向レジスタに置き換えられなければなりません。DD\_MOSI, DD\_MISO, DD\_SCKはこれらのピンに対する実際のポート方向ビットに置き換えられなければなりません。例えばMOSIがPB5ピンに配置されるなら、DD\_MOSIはDDB5、DDR\_SPIはDDRBに置き換ええます。

#### アセンブリ言語プログラム例

```
SPI_M_Init: LDI    R17, (1<<DD_MOSI) | (1<<DD_SCK)      ;MOSI, SCK=出力、他は入力値を取得
            OUT    DDR_SPI, R17                        ;MOSI, SCK=出力、他は入力に設定
            LDI    R17, (1<<SPE) | (1<<MSTR) | (1<<SPRO) ;SPI許可、主装置、16分周値を取得
            OUT    SPCR, R17                          ;SPI許可、主装置、16分周に設定
            RET                                         ;呼び出し元へ復帰

SPI_M_Tx:   OUT    SPDR, R16                          ;データ(R16)送信開始
SPI_M_Tx_W: SBIS   SPSR, SPIF                        ;転送完了ならばスキップ
            RJMP   SPI_M_Tx_W                        ;転送完了まで待機
;
            RET                                         ;呼び出し元へ復帰
```

#### C言語プログラム例

```
void SPI_MasterInit(void)
{
    DDR_SPI = (1<<DD_MOSI) | (1<<DD_SCK);          /* MOSI, SCK=出力、他は入力に設定 */
    SPCR = (1<<SPE) | (1<<MSTR) | (1<<SPRO);        /* SPI許可、主装置、16分周に設定 */
}

void SPI_MasterTransmit(char cData)
{
    SPDR = cData;                                  /* データ送信開始 */
    while(!(SPSR & (1<<SPIF)));                   /* 転送完了まで待機 */
}
```

注: 6頁の「コード例について」をご覧ください。

次のコード例は従装置としてSPIを初期化する方法と簡単な受信を実行する方法を示します。

#### アセンブリ言語プログラム例

```
SPI_S_Init: LDI    R17, (1<<DD_MISO)                 ;MISO出力、他は入力値を取得
            OUT    DDR_SPI, R17                     ;MISO出力、他は入力に設定
            LDI    R17, (1<<SPE)                   ;SPI許可値を取得
            OUT    SPCR, R17                       ;SPI許可設定
            RET                                         ;呼び出し元へ復帰

SPI_S_Rx:   SBIS   SPSR, SPIF                       ;受信(転送)完了ならばスキップ
            RJMP   SPI_S_Rx                         ;受信(転送)完了まで待機
;
            IN     R16, SPDR                         ;受信データを取得
            RET                                         ;呼び出し元へ復帰
```

#### C言語プログラム例

```
void SPI_SlaveInit(void)
{
    DDR_SPI = (1<<DD_MISO);                          /* MISO出力、他は入力に設定 */
    SPCR = (1<<SPE);                                  /* SPI許可設定 */
}

char SPI_SlaveReceive(void)
{
    while(!(SPSR & (1<<SPIF)));                       /* 受信(転送)完了まで待機 */
    return SPDR;                                      /* 受信データと共に復帰 */
}
```

注: 6頁の「コード例について」をご覧ください。

## SSピンの機能

### 従装置動作

SPIが従装置として設定されると、従装置選択(SS)ピンは常に入力です。SSがLowに保たれるとSPIは活性に(作動)され、使用者によってそのように設定されていればMISOは出力になります。他の全てのピンは入力です。SSがHighに駆動されると、出力として使用者設定され得るMISOを除く全てのピンは入力、SPIは非活動で、それは到着データを受信しないことを意味します。一旦SSピンがHighに駆動されると、SPI論理回路がリセットすることに注意してください。

このSSピンはパケット/バイト同期に対して、従装置ビット計数器が主装置クロック発生器との同期を保つのに有用です。SSピンがHighに駆動されると、SPI従装置は直ちに送受信論理回路をリセットし、それは移動レジスタ内で部分的に受信したどのデータも取り落とします。

### 主装置動作

SPIが主装置(SPI制御レジスタ(SPCR)の主装置許可(MSTR)ビット=1)として設定されると、SSピンの方向は使用者が決められます。

SSが出力として設定されると、このピンはSPIシステムに影響を及ぼされない標準出力ピンです。代表的にはこのピンがSPI従装置のSSピンを駆動するでしょう。

SSが入力として設定されると、SPI主装置動作を保証するため、それはHighに保持されなければなりません。SSピンが入力として定義されたSPI主装置として設定される時に周辺回路によってSSピンがLowに駆動されると、SPIシステムは他の主装置が従装置として選んでデータ送信を始めると解釈します。バスの衝突を避けるためにSPIシステムは次の動作を行います。

1. SPCRで主/従選択(MSTR)ビットが解除(0)され、SPIシステムは従装置になります。SPIシステムが従装置になる結果としてMOSIとSCKピンが入力になります。
2. SPI状態レジスタ(SPSR)でSPI割り込み要求フラグ(SPIF)が設定(1)され、そしてSPI割り込みが許可(SPCRのSPIE=1)され、且つ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)なら、割り込みルーチンが実行されます。

従って割り込み駆動SPI送信が主装置動作で使われ、SSがLowに駆動される可能性があるとき、その割り込み(処理)はMSTRビットが未だ設定(1)されていることを常に検査すべきです。MSTRビットが従装置選択によって解除(0)されてしまっていると、それはSPI主装置動作を再び許可するため、使用者によって設定(1)されなければなりません。

### SPI制御レジスタ (SPI Control Register) SPCR

| ビット         | 7    | 6   | 5    | 4    | 3    | 2    | 1    | 0    |      |
|-------------|------|-----|------|------|------|------|------|------|------|
| \$0D (\$2D) | SPIE | SPE | DORD | MSTR | CPOL | CPHA | SPR1 | SPR0 | SPCR |
| Read/Write  | R/W  | R/W | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  |      |
| 初期値         | 0    | 0   | 0    | 0    | 0    | 0    | 0    | 0    |      |

#### • ビット7 – SPIE : SPI割り込み許可 (SPI Interrupt 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)されて、SPI状態レジスタ(SPSR)でSPI割り込み要求フラグ(SPIF)が設定(1)されると、このビットがSPI割り込みを実行させます。

#### • ビット6 – SPE : SPI許可 (SPI Enable)

SPEビットが1を書かれるとSPIが許可されます。どのSPI操作を許可するにも、このビットが設定(1)されなければなりません。

#### • ビット5 – DORD : データ順選択 (Data Order)

DORDビットが1を書かれるとデータ語のLSBが最初に転送されます。DORDビットが0を書かれるとMSBが最初に転送されます。

#### • ビット4 – MSTR : 主装置/従装置選択 (Master/Slave Select)

このビットは1を書かれると主装置動作、論理0を書かれると従装置動作を選びます。SSが入力として設定され、MSTRが設定(1)の間にLowへ駆動されると、MSTRが解除(0)され、SPI状態レジスタ(SPSR)でSPI割り込み要求フラグ(SPIF)が設定(1)になります。その後使用者はSPI主装置動作を再び許可するためにMSTRを設定(1)しなければなりません。

#### • ビット3 – CPOL : SCK極性選択 (Clock Polarity)

このビットが1を書かれると、アイドル時にSCKはHighです。CPOLが0を書かれると、アイドル時にSCKはLowです。例については図77と図78を参照してください。CPOL機能は右で要約されます。

表70. CPOL機能動作

| CPOL | SCK先行端 | SCK後行端 |
|------|--------|--------|
| 0    | 上昇端    | 下降端    |
| 1    | 下降端    | 上昇端    |

#### • ビット2 – CPHA : SCK位相選択 (Clock Phase)

このSCK位相選択(CPHA)ビットの設定はデータがSCK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。例については図77と図78を参照してください。CPHA機能は右で要約されます。

表71. CPHA機能動作

| CPHA | SCK先行端 | SCK後行端 |
|------|--------|--------|
| 0    | 入力採取   | 出力設定   |
| 1    | 出力設定   | 入力採取   |

•ビット1,0 – SPR1,SPR0 : SPIクロック選択 (SPI Clock Rate Select 1 and 0)

これら2ビットは主装置として設定されたデバイスのSCK速度を制御します。従装置でのSPR1とSPR0は無効です。SCKと(システム)発振器クロック周波数fosc間の関連は次表で示されます。

表72. SCK速度選択 (fosc=CPUクロック周波数)

|        |        |        |        |         |         |         |   |          |
|--------|--------|--------|--------|---------|---------|---------|---|----------|
| SPR1   | 0      |        | 0      |         | 1       |         | 1 |          |
| SPR0   | 0      |        | 1      |         | 0       |         | 1 |          |
| SPI2X  | 1      | 0      | 1      | 0       | 1       | 0       | 1 | 0        |
| SCK周波数 | fosc/2 | fosc/4 | fosc/8 | fosc/16 | fosc/32 | fosc/64 |   | fosc/128 |

SPI状態レジスタ (SPI Status Register) SPSR

|             |      |      |   |   |   |   |   |       |      |
|-------------|------|------|---|---|---|---|---|-------|------|
| ビット         | 7    | 6    | 5 | 4 | 3 | 2 | 1 | 0     |      |
| \$0E (\$2E) | SPIF | WCOL | - | - | - | - | - | SPI2X | SPSR |
| Read/Write  | R    | R    | R | R | R | R | R | R/W   |      |
| 初期値         | 0    | 0    | 0 | 0 | 0 | 0 | 0 | 0     |      |

•ビット7 – SPIF : SPI割り込み要求フラグ (SPI Interrupt Flag)

直列転送が完了すると、このSPIFフラグが設定(1)されます。全割り込みが許可(ステータスレジスタ(SREG)の全割り込み許可(I)ビット=1)されて、SPI制御レジスタ(SPCR)でSPI割り込み許可(SPIE)ビットが設定(1)されるなら、割り込みが生成されます。SPIが主装置動作の時にSSピンが入力でLowに駆動されるなら、これもこのSPIFフラグを同様に設定(1)します。対応する割り込み処理ベクタを実行する時にSPIFはハードウェアによって解除(0)されます。代わりにSPIFが設定(1)されたSPI状態レジスタ(SPSR)を始めに読み、その後にSPIデータレジスタ(SPCR)をアクセスすることによってもSPIFフラグは解除(0)されます。

•ビット6 – WCOL : 上書き発生フラグ (Write Collision Flag)

データ転送中にSPIデータレジスタ(SPDR)が書かれると、このWCOLビットが設定(1)されます。WCOLビットとSPIFビットはWCOLが設定(1)されたSPI状態レジスタ(SPSR)を始めに読み、その後にSPIデータレジスタ(SPDR)をアクセスすることによって解除(0)されます。

•ビット5~1 – Res : 予約 (Reserved Bit)

これらのビットは予約されており、常に0として読みます。

•ビット0 – SPI2X : SPI倍速許可 (Double SPI Speed Bit)

このビットが論理1を書かれると、SPIが主装置動作の時にSCK速度(SCK周波数)が倍にされます(表72.参照)。これは最小SCK周期が2CPUクロック周期であることを意味します。SPIが従装置として設定される時に、SPIはfosc(CPUクロック周波数)/4またはそれ以下の動作のみ保証されます。

ATmega128のSPIインターフェースはフラッシュメモリやEEPROMの書き換え(読み書き)にも使われます。直列プログラミングと照合については197頁をご覧ください。

SPIデータレジスタ (SPI Data Register) SPDR

|             |       |     |     |     |     |     |     |       |      |
|-------------|-------|-----|-----|-----|-----|-----|-----|-------|------|
| ビット         | 7     | 6   | 5   | 4   | 3   | 2   | 1   | 0     |      |
| \$0F (\$2F) | (MSB) |     |     |     |     |     |     | (LSB) | SPDR |
| Read/Write  | R/W   | R/W | R/W | R/W | R/W | R/W | R/W | R/W   |      |
| 初期値         | 不定    | 不定  | 不定  | 不定  | 不定  | 不定  | 不定  | 不定    |      |

SPIデータレジスタはSPI移動レジスタとレジスタファイル(汎用レジスタ)間のデータ転送に使われる読み書き可能なレジスタです。このレジスタへの書き込みがデータ送信を開始します。このレジスタの読み込みは移動レジスタの受信緩衝部読み出しを引き起こします。

## データ転送形式

直列データに関してはSPI制御レジスタ(SPCR)のSCK位相(CPHA)とSCK極性(CPOL)制御ビットによって決定されるSCK位相と極性で4つの組み合わせがあります。このSPIデータ転送形式は図77と図78.で示されます。データビットは安定のためデータ信号に対して十分な時間を保証するSCK信号の反対端で移動出力と(入力)ラッチが行われます。これは以下で行われるように表70と表71.を要約することによって明解にされます。

表73. CPOL,CPHA機能動作

| SPI動作種別番号 | CPOL | CPHA | SCK先行端   | 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77. SPI<sup>™</sup>-データ転送形式 (CPHA=0)

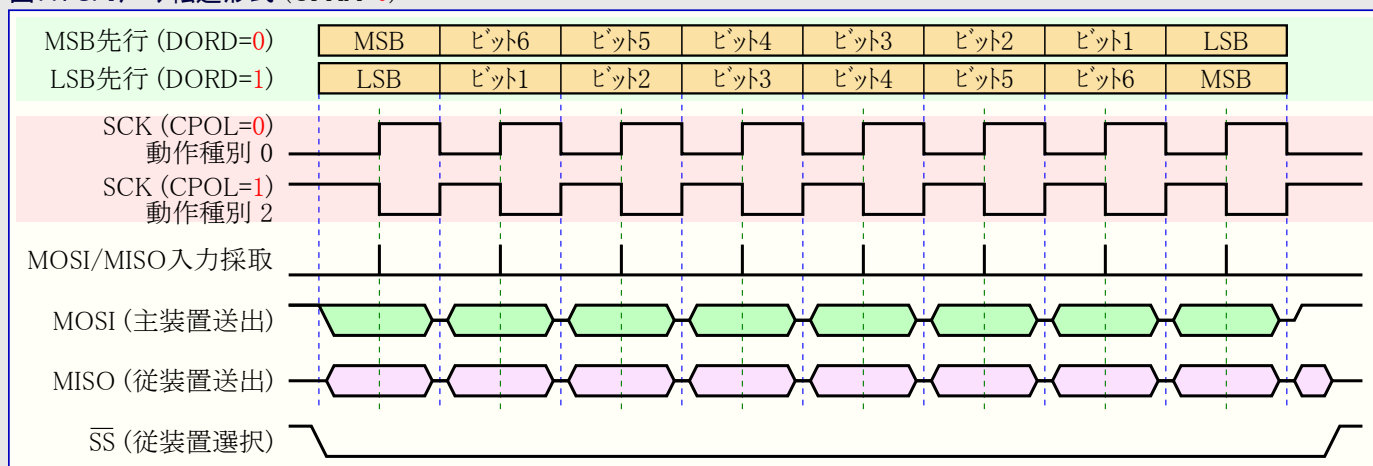
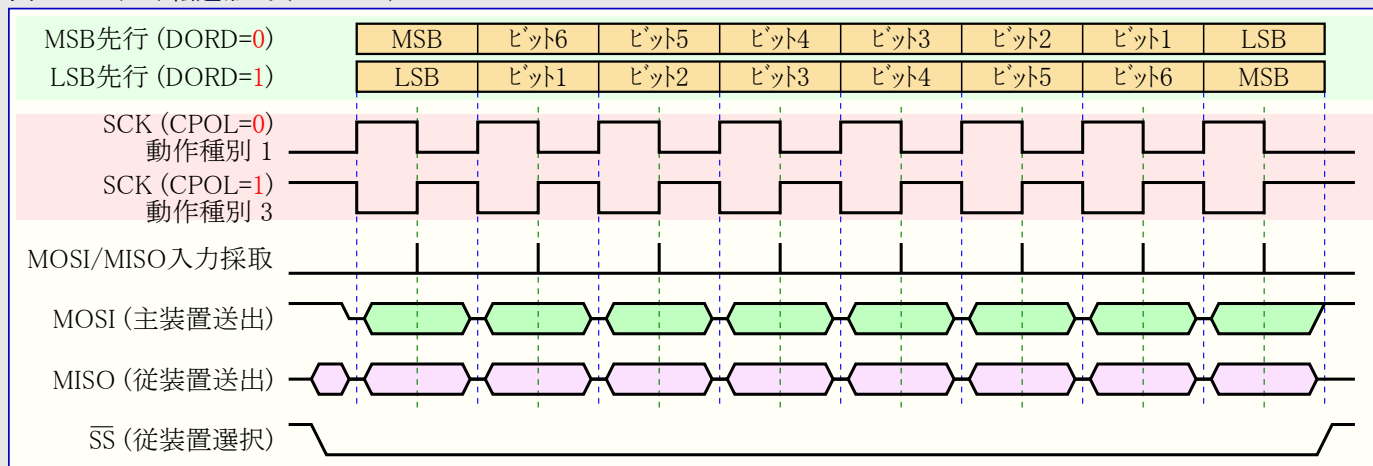


図78. SPI<sup>™</sup>-データ転送形式 (CPHA=1)



## USART (USART0, USART1)

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。主な特徴を次に示します。

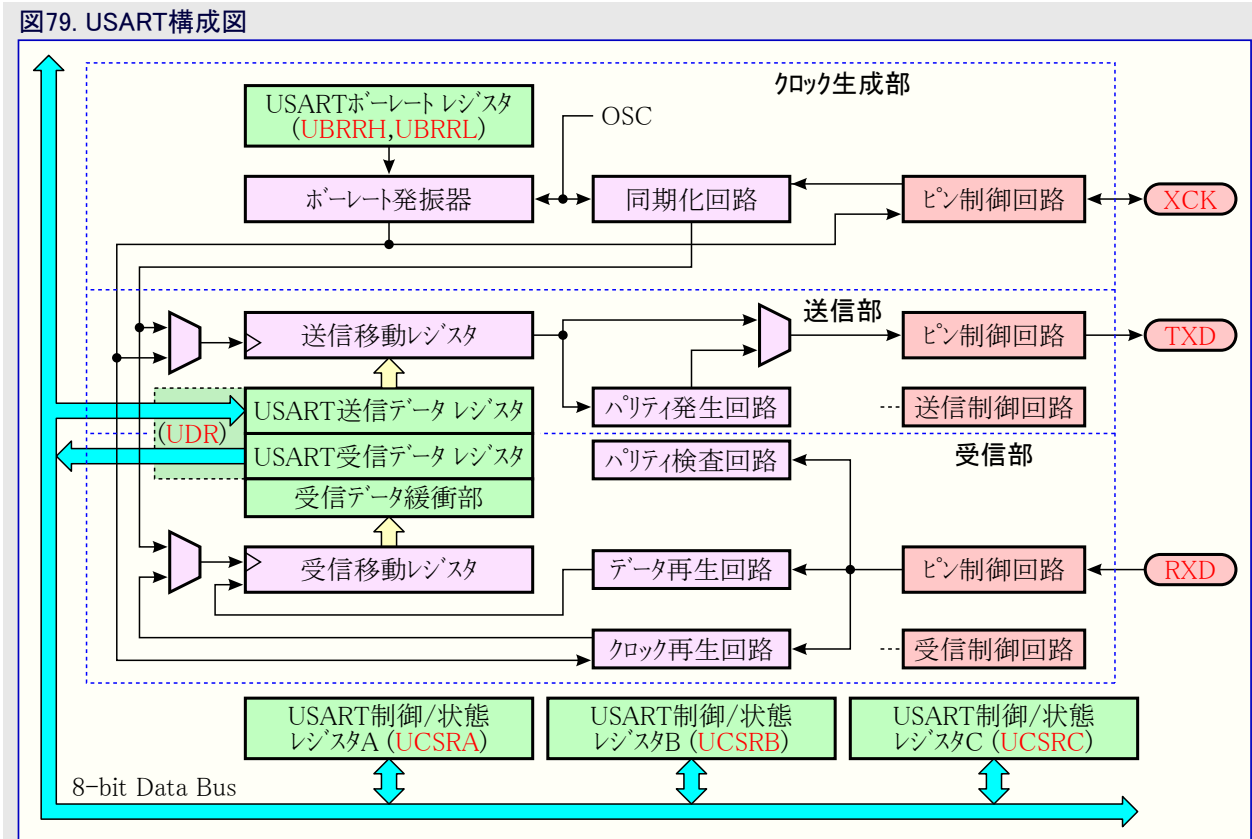
- 全二重動作 (独立した送受信レジスタ)
- 同期または非同期動作
- 同期クロック駆動された主装置/従装置動作
- 高分解能ボーレート発振器
- 5, 6, 7, 8または9ビットデータと1または2停止ビットの直列フレームの支援
- ハードウェアによって支援された奇数または偶数パリティの生成と検査
- データオーバーラン検出
- フレーミング異常検出
- 不正開始ビット検出とデジタル低域通過濾波器を含む雑音除去器
- 受信完了、送信完了、送信データレジスタ空きの3つの分離した割り込み
- 複数プロセッサ通信機能
- 倍速非同期通信動作

### 2つのUSART

ATmega128にはUSART0とUSART1の2つのUSARTがあります。両USARTに関する機能は以下で記述されます。USART0とUSART1は234頁の「レジスタ要約」で示されるように個別のI/Oレジスタを持ちます。ATmega103互換動作ではUSART1が利用できず、UBRR0HとUCSR0Cのどちらのレジスタもありません。これはATmega103互換動作でATmega128がUSART0の非同期動作だけ支援することを意味します。

### 概要

USARTの簡便化した構成図は図79.で示されます。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有されます。**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(XCK)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(UDR)、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く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(UDR)、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。



## UARTとの互換性

USARTはAVRのUARTと次の項目に関して完全な互換性があります。

- 全てのUSART側レジスタでのビット位置
- ボーレート生成
- 送信操作
- 送信緩衝の動作
- 受信操作

けれども受信緩衝動作にはいくつかの特別な場合で互換性に影響を及ぼす2つの改良点があります。

- 第2受信緩衝部が追加されました。2つの緩衝レジスタは循環型FIFO緩衝部として動作します。従ってUSARTデータレジスタ(UDRn)は到着データ毎に一度だけ読まれなければなりません。より重要なのは異常フラグ(FEnとDORn)と第9データビット(RXB8n)が受信緩衝部内のデータと共に緩衝されることの実事です。従って状態ビットは常にUSARTデータレジスタ(UDRn)が読まれる前に読まれなければなりません。さもなければ緩衝部の状態が失われるため、異常情報も失われます。
- 受信部移動レジスタは第3緩衝段のように動けます。これは緩衝レジスタが一杯の場合、新規開始ビットが検出されるまで直列移動レジスタ(図79参照)内に留まるのを受信したデータに許すことによって行われます。従ってUSARTはデータオーバーラン(DORn)異常条件により耐えます。

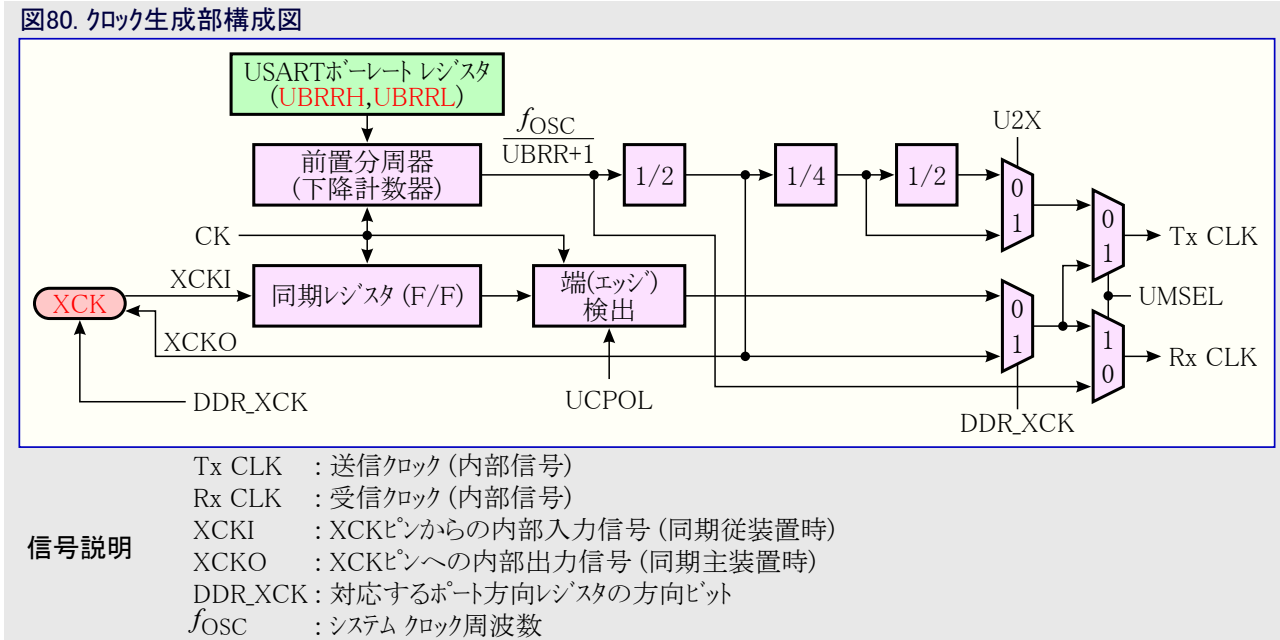
次の制御ビットは名称変更されましたが、機能とレジスタ(ビット)位置は同じです。

- 9ビット選択(CHR9)ビットはデータ長選択ビット2(UCSZn2)へ変更
- オーバーラン発生(OR)フラグはオーバーラン発生(DORn)フラグへ変更
- パリティ誤り(PE)フラグはパリティ誤り(UPEn)フラグへ変更 (訳注:原書で欠落のため追加)

## 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(UCSRC)のUSART動作種別選択(UMSEL)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(UCSRA)にある倍速許可(U2X)ビットによって制御されます。同期動作(UMSEL=1)を使うとき、XCKピンに対する方向制御ビット(DDR\_XCK)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XCKピンは同期動作を使う時だけ活性(有効)です。

図80.はクロック生成論理回路の構成図を示します。



### 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。本項の記述は図80.を参照してください。

USARTボーレートレジスタ(UBRR(UBRRH:UBRRL))と下降カウンタは設定変更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( $f_{OSC}$ )で走行する下降カウンタは0への下降計数時毎またはUBRRLレジスタが書かれる時にUBRR値で設定されます。1クロック(パルス)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( $=f_{OSC}/(UBRR+1)$ )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(UMSEL)、倍速許可(U2X)、DDR\_XCK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。

表74.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UBRR値の計算式を含みます。

表74. ボーレートレジスタ(UBRR)値計算式

| 動作種別             | ボーレート計算式                                    | UBRR値計算式                                    |
|------------------|---|---|
| 標準速非同期動作 (U2X=0) | $BAUD = \frac{f_{OSC}}{16 \times (UBRR+1)}$ | $UBRR = \frac{f_{OSC}}{16 \times BAUD} - 1$ |
| 倍速非同期動作 (U2X=1)  | $BAUD = \frac{f_{OSC}}{8 \times (UBRR+1)}$  | $UBRR = \frac{f_{OSC}}{8 \times BAUD} - 1$  |
| 同期主装置動作          | $BAUD = \frac{f_{OSC}}{2 \times (UBRR+1)}$  | $UBRR = \frac{f_{OSC}}{2 \times BAUD} - 1$  |

**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)  
 UBRR : UBRRHとUBRRL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UBRR値の例は表82.で得られます(131頁をご覧ください)。

## 倍速動作 (U2X)

転送速度はUSART制御/状態レジスタA(UCSRA)で倍速許可(U2X)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に減じます。けれども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)採取数をこの場合だけ使い、従ってこの動作種別が使われる時は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ることに注意してください。送信部についての低下要因はありません。

## 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。本項の記述での詳細については図80を参照してください。

XCKピンからの外部クロック入力是不確定レベル状態(マスタペブル)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端(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2 CPUクロック周期の遅延を持ち込み、このため最大外部XCKクロック周波数は次式によって制限されます。

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

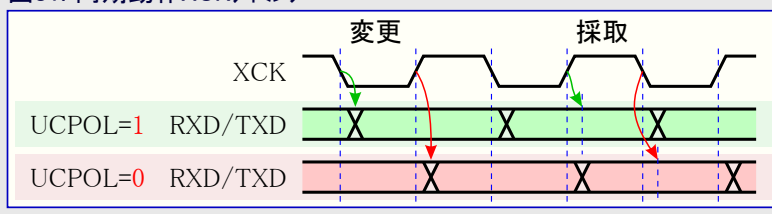
$f_{OSC}$ がシステムクロック元の安定度に依存することに注意してください。従って周波数変動によるデータ消失の可能性を避けるため、いくらかの余裕分を追加することが推奨されます。

## 同期クロック動作

同期動作が使われる(UMSEL=1)とき、XCK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(TXD)が変更される端と反対のXCKクロック端でデータ入力(RXD)が採取されることです。

USART制御/状態レジスタC(UCSRC)のXCK極性(UCPOL)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かを選びます。図81.で示されるようにUCPOLが0のとき、データはXCKの上昇端で変更され、下降端で採取されます。UCPOLが設定(1)の場合、データはXCKの下降端で変更され、上昇端で採取されます。

図81. 同期動作XCKタイミング



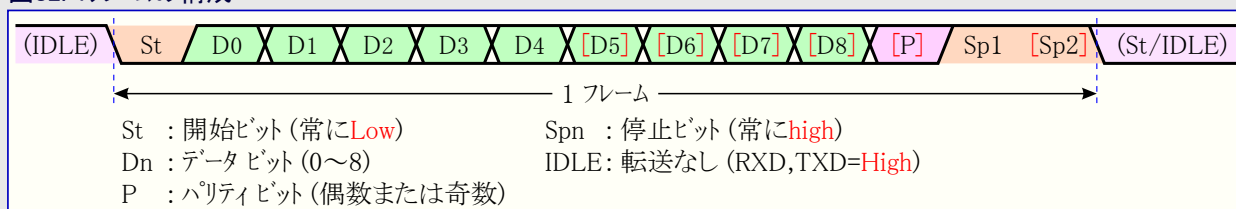
## フレーム形式

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

フレームは最下位データビット(LSB)が次に続く開始ビットで始まります。その後に次データビットが最後の最上位データビット(MSB)まで(最大)合計9ビット続きます。許可したなら、パリティビットがデータビットの後、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。図82.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82. 1フレームの構成



USARTによって使われるフレーム形式はUSART制御/状態レジスタBとC(UCSRB, UCSRC)でデータ長選択(UCSZ2~0)ビット、パリティ選択(UPM1,0)ビット、停止ビット選択(USBS)ビットによって設定されます。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。

データ長選択(UCSZ2~0)ビットはフレーム内のデータビット数を選びます。パリティ選択(UPM1,0)ビットはパリティビットの許可と種別(奇/偶)を設定します。1または2停止ビットのどちらかの選択は停止ビット選択(USBS)ビットによって行います。受信部は第2停止ビットを無視します。従ってフレーミング異常(FE)は最初の停止ビットが0(Low)の場合にだけ検出されます。

## 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。

$$\begin{aligned} \text{偶数パリティビット} &= D_0 \text{ Ex-OR } D_1 \text{ Ex-OR } D_2 \text{ Ex-OR } \sim \text{ Ex-OR } D_{n-1} \\ \text{奇数パリティビット} &= D_0 \text{ Ex-OR } D_1 \text{ Ex-OR } D_2 \text{ Ex-OR } \sim \text{ Ex-OR } D_{n-1} \text{ Ex-OR } 1 \end{aligned} \quad n: \text{データビット長}$$

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。

## 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\(0\)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\(UCSRA\)の送信完了\(TXC\)フラグ](#)は送信部の全転送完了検査に使い、[受信完了\(RXC\)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TXCフラグが使われる場合、各々の送信([USARTデータレジスタ\(UDR\)](#)が書かれる)前にTXC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(UBRR)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    UBRRH, R17                ;ボーレート設定(上位バイト)
             OUT    UBRRL, R16             ;ボーレート設定(下位バイト)
             LDI    R16, (1<<USBS) | (3<<UCSZ0) ;フレーム形式値を取得
             OUT    UCSRC, R16             ;フレーム形式設定(8ビット,2停止ビット)
             LDI    R16, (1<<RXEN) | (1<<TXEN) ;送受信許可値を取得
             OUT    UCSRB, 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UBRR値 */

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

void USART_Init(unsigned int baud)
{
    UBRRH = (unsigned char) (baud>>8); /* ボーレート設定(上位バイト) */
    UBRRL = (unsigned char) baud; /* ボーレート設定(下位バイト) */
    UCSRC = (1<<USBS) | (3<<UCSZ0); /* フレーム形式設定(8ビット,2停止ビット) */
    UCSRB = (1<<RXEN) | (1<<TXEN); /* 送受信許可 */
}
```

**注:** 6ページの「[コード例について](#)」をご覧ください。

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われ、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。

## USARTのデータ送信

USART送信部はUSART制御/状態レジスタB(UCSRB)で送信許可(TXEN)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、TXD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。何か送信を行う前に一度はポーレート、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、XCK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。

### 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(UDR)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、ポーレートレジスタ(UBRRH:UBRRL)と倍速許可(U2X)ビット、また動作種別によってはXCK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、UDR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(UCSRA)の送信データレジスタ空き(UDRE)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 UCSRA, UDRE ;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 UDR, 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 ( !(UCSRA & (1<<UDRE)) ); /* 送信緩衝部空き待機 */
    UDR = data; /* データ送信(送信開始) */
}
```

注: 6頁の「コード例について」をご覧ください。

この関数は送信されるべき新規データを設定する前に、UDRE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。

### 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(UCSZ2~0=111)、データの下位バイトがUSARTデータレジスタ(UDR)に書かれる前に、第9ビットがUSART制御/状態レジスタB(UCSRB)の送信データビット8(TXB8)ビットに書かれなければなりません。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 UCSRA, UDRE ;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 UCSRB, TXB8 ;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 UCSRB, TXB8 ;第9ビットを1に設定
          OUT UDR, 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 ( !(UCSRA & (1<<UDRE)) ); /* 送信緩衝部空き待機 */
    UCSRB &= ~(1<<TXB8); /* TXB8を0に仮設定 */
    if (data & 0x0100) UCSRB |= (1<<TXB8); /* 第9ビットをR17からTXB8へ複写 */
    UDR = 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。UCSRBの内容が静的(換言すると、UCSRBのTXB8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。

6頁の「コード例について」をご覧ください。

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。



## 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(UDRE)と送信完了(TXC)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(UDRE)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(UCSRA)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(UCSRB)でデータレジスタ空き割り込み許可(UDRIE)ビットが1を書かれると、(全割り込みが許可されているならば)UDRE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。UDREはUSARTデータレジスタ(UDR)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UDREを解除(0)するために新規データをUDR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(TXC)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。TXCフラグは送信完了割り込みが実行される時、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。TXC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用です。

UCSRBで送信完了割り込み許可(TXCIE)ビットが設定(1)され、(全割り込みが許可されていれば)TXC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TXC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時、自動的に行われます。

## 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(UPM1=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。

## 送信の禁止

送信部の禁止(UCSRBのUSART送信許可(TXEN)ビット=0)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時)まで有効になりません。禁止されると、送信部はもはやTXDピン(の標準ピン機能)を無効にしません。

## USARTのデータ受信

USART受信部はUSART制御/状態レジスタB(UCSRB)で受信許可(RXEN)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、RXD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ポーレート、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、XCKピンのクロックは転送クロックとして使われます。

## 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ポーレートまたはXCK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(UDR)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、UDR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(UCSRA)の受信完了(RXC)フラグのポーリングを基にした簡単な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    UCSRA, RXC          ;受信完了でスキップ
           RJMP    USART_Rx         ;受信完了待機
;
           IN     R16, UDR           ;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 ( !(UCSRA & (1<<RXC)) ); /* 受信完了待機 */
    return UDR;                    /* 受信データ取得 */
}
```

注: 6頁の「コード例について」をご覧ください。

この関数は緩衝部を読んで値を戻す前に、RXC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。

## 9ビット データ フレーム受信

9ビット データが使われる場合(UCSZ2~0=111)、USARTデータレジスタ(UDR)から下位バイトを読む前に、第9ビットがUSART制御/状態レジスタB(UCSRB)の受信データビット8(RXB8)ビットから読まなければならない。この規則はフレーミング異常(FE)、オーバーラン発生(DOR)、パリティ誤り(UPE)状態フラグにも適用します。USART制御/状態レジスタA(UCSRA)から状態情報を読み、その後UDRからデータを読んでください。UDR 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RXB8,FE,DOR,UPE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    UCSRA, RXC          ;受信完了でスキップ
           RJMP    USART_Rx       ;受信完了待機
;
           IN     R18, UCSRA       ;状態フラグ取得
           IN     R17, UCSRB       ;受信第9ビット取得
           IN     R16, UDR         ;受信データ取得
           ANDI   R18, (1<<FE) | (1<<DOR) | (1<<UPE) ;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            ;RXB8ビットをビット0位置へ移動
           ANDI  R17, $01         ;RXB8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 ( !(UCSRA & (1<<RXC)) ); /* 受信完了待機 */
    status = UCSRA; /* 状態フラグ取得 */
    resh = UCSRB; /* 受信第9ビット取得 */
    resl = UDR; /* 受信データ取得 */
    if ( status & ((1<<FE) | (1<<DOR) | (1<<UPE)) ) return -1; /* 受信異常で-1値設定/復帰 */
    resh = (resh>>1) & 0x01; /* RXB8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 */
}
    
```

**注:** 6頁の「コード例について」をご覧ください。

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。

## 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(RXC)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(RXEN=0)、受信緩衝部が破棄され、その結果としてRXCフラグは0になります。

USART制御/状態レジスタB(UCSRB)で**USART受信完了割り込み許可(RXCIE)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)RXC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われるとき、受信完了割り込みルーチンはRXCフラグを解除(0)するためにUSARTデータレジスタ(UDR)から受信したデータを読まなければならない、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)起きます。

## 受信異常フラグ

USART受信には3つの異常フラグ、フレーミング異常(FE)、データオーバーラン発生(DOR)、パリティ誤り(UPE)があります。全てが**USART制御/状態レジスタA(UCSRA)**を読むことによってアクセスできます。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(UDR)** 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、UCSRAは受信緩衝部(UDR)の前に読まなければならない。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、UCSRA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らない。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(FE)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。FE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FE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、FEフラグは**USART制御/状態レジスタC(UCSRC)の停止ビット選択(USBS)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、UCSRA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(DOR)フラグ**は受信部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。DORフラグが設定(**1**)なら、最後にUDRから読んだフレームと次にUDR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ています。将来のデバイスとの共通性のため、UCSRAに書くとき、常にこのビットに**0**を書いてください。DOR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(UPE)フラグ**は受信緩衝部内の次フレームで、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、UPE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、UCSRA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**120頁の「パリティビットの計算」と次の「パリティ検査器」**をご覧ください。

## パリティ検査器

パリティ検査器はパリティ種別上位ビット(**UPM1**)が設定(**1**)されると活性(有効)になります。実行されるべきパリティ検査の形式(偶数または奇数)は**UPM0**ビットによって選ばれます。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。その後パリティ誤り(UPE)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。

UPE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(**UPM1=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(UDR)が読まれるまで有効です。

## 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。禁止されると(換言すると、**USART制御/状態レジスタB(UCSRB)のUSART受信許可(RXEN)ビット**が**0**に設定)、受信部はもはやRXD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。

## 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(UCSRA)の受信完了(RXC)フラグ**が解除(**0**)されるまでUSARTデータレジスタ(UDR) 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    UCSRA, RXC          ;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, UDR           ;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 ( UCSRA & (1<<RXC) ) dummy=UDR; /* 未読データ読み捨て */
}
```

注: 6頁の「コード例について」をご覧ください。

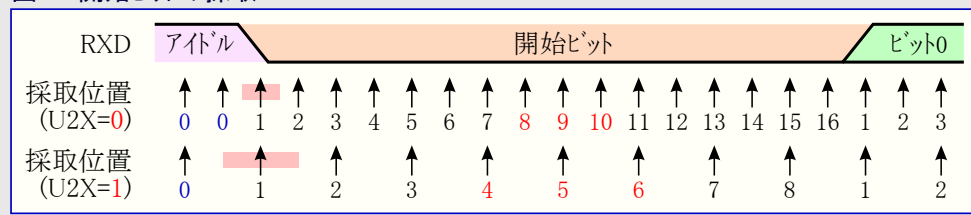
## 非同期受信

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RXD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。

### 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。図83は到着フレームの開始ビット採取手順を図解します。採取速度は標準動作でボーレートの16倍、倍速動作で8倍です。赤帯(訳注: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(U2X=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RXD信号がアイドル(換言すると、通信の動きなし)の時にされる採取です。

図83. 開始ビットの採取

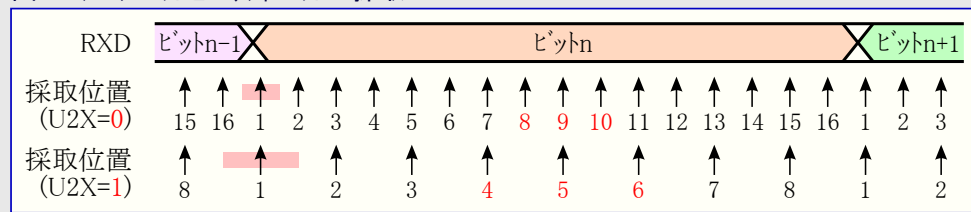


クロック再生論理回路がRXD信号線でHigh(アイドル)からLow(開始)の遷移を検出すると、開始ビット検出手順が開始されます。図で示されるように採取1は最初のLow採取を意味します。その後にクロック再生回路は有効な開始ビットが受信されるかを定めるために、標準動作に対して採取8,9,10、倍速動作に対して採取4,5,6(図の赤字(訳注:原文は箱枠内))で示された採取番号)を使います。これら3回で2回以上の採取が論理Highレベルの場合(多数決)、この開始ビットは尖頭雑音として捨てられ、受信部は次のHighからLowの遷移を探し始めます。けれども有効な開始ビットが検出されると、クロック再生論理回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

### 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められます。データ再生部は標準動作で16段、倍速動作で8段の順列回路を使います。図84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。

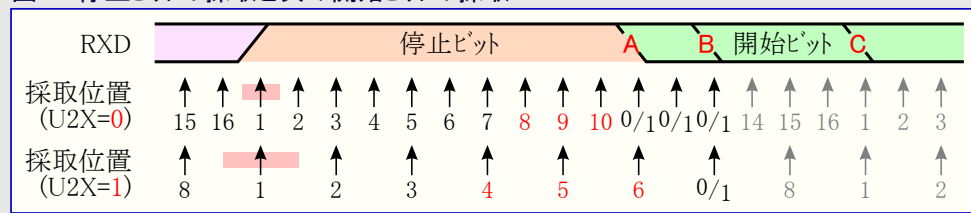
図84. データビットとパリティビットの採取



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。この中央の3採取は図上の赤字(訳注:原文は箱枠内)の採取番号によって強調されます。多数決の手順は次のように行われます。2または全3採取が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または全3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RXD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に完全な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。受信部がフレームの最初の停止ビットだけを使うことに注意してください。

図85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85. 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(FE)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。標準動作での最初のLowレベル採取は図85のA点で有り得ます。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。



## 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(表75.参照)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。

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。

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

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<sub>F</sub> : 多数決に使う最初の採取番号 (標準速=8、倍速=4)  
 S<sub>M</sub> : 多数決に使う中心の採取番号 (標準速=9、倍速=5)  
 R<sub>slow</sub> : 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<sub>fast</sub> : 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。

表75.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力があることに注目してください。

表75. 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 (U2X=0) |          |             |           | 倍速動作 (U2X=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75.に標準速、表76.に倍速を記載していますが、比較が容易なように表75.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(XTAL)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水晶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い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UBRR値が使えます。



## 複数プロセッサ通信動作

USART制御/状態レジスタA(UCSRA)での**複数プロセッサ通信動作(MPCM)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由で通信する複数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MPCM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(UCSRB)の受信第9(RXB8)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。

## 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(UCSZ=7)を使えます。UCSRBの**送信第9(TXB8)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(UCSRAの**複数プロセッサ通信動作(MPCM)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UCSRAで**受信完了(RXC)フラグ**が設定(1)されます。
3. 各従MCUは**USARTデータレジスタ(UDR)**を読み、選ばれたかを判定します。選ばれた場合はUCSRAのMPCMビットを解除(0)し、そうでなければ(非選択の場合は)MPCM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。MPCM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MPCM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、手順は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(USBS=1)に設定されなければなりません。

MPCMビットを設定(1)または解除(0)するのに読み-修正-書き(リード モデファイライト)命令(SBIとCBI)を使ってはいけません。MPCMビットは**送信完了(TXC)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。

## USART用レジスタ

### USARTnデータレジスタ (USARTn I/O Data Register) UDRn

| ビット                 | 7       | 6       | 5       | 4       | 3       | 2       | 1       | 0       |      |
|---------------------|---------|---------|---------|---------|---------|---------|---------|---------|------|
| \$0C(\$2C) : (\$9C) | R/TXBn7 | R/TXBn6 | R/TXBn5 | R/TXBn4 | R/TXBn3 | R/TXBn2 | R/TXBn1 | R/TXBn0 | UDRn |
| Read/Write          | R/W     | R/W     | R/W     | R/W     | R/W     | R/W     | R/W     | R/W     |      |
| 初期値                 | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 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UDRnとして引用しました。送信データ緩衝レジスタ(TXB)はUDRnレジスタ位置に書かれるデータの転送先です。UDRn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UCSRnAで送信データレジスタ空き(UDREN)フラグが設定(1)される時にだけ書けます。UDRENフラグが設定(1)されない時にUDRn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TXDn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。

### USARTn制御/状態レジスタA (USARTn Control and Status Register A) UCSRnA

| ビット                 | 7    | 6    | 5     | 4   | 3    | 2    | 1    | 0     |        |
|---------------------|------|------|-------|-----|------|------|------|-------|--------|
| \$0B(\$2B) : (\$9B) | RXCn | TXCn | UDREN | FEn | DORn | UPEn | U2Xn | MPCMn | UCSRnA |
| Read/Write          | R    | R/W  | R     | R   | R    | R    | R/W  | R/W   |        |
| 初期値                 | 0    | 0    | 1     | 0   | 0    | 0    | 0    | 0     |        |

#### • ビット7 – RXCn : USART受信完了フラグ (USARTn 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RXCnフラグは0になります。RXCn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(UCSRnB)の受信完了割り込み許可(RXCIE)ビットをご覧ください)。

#### • ビット6 – TXCn : USART送信完了フラグ (USARTn 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(UDRn)に新規データが現存しない時に設定(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、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フラグは送信完了割り込みを発生できます(UCSRnBの送信完了割り込み許可(TXCIE)ビットをご覧ください)。

#### • ビット5 – UDREN : USART送信データレジスタ空きフラグ (USARTn Data Register Empty)

UDRENフラグは送信緩衝部(UDRn)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。UDREN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。UDRENフラグは送信緩衝部空き割り込みを発生できます(UCSRnBの送信データレジスタ空き割り込み許可(UDRIE)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UDRENは設定(1)です。

#### • ビット4 – FEn : 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(UDRn)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FEnフラグは0です。UCSRnAに書くとき、常にこのビットを0に設定してください。

#### • ビット3 – DORn : データオーバラン発生フラグ (Data OverRun)

このビットはオーバラン状態が検出されると設定(1)されます。受信緩衝部(2フレーム分)が一杯で、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データオーバランが起こります。UCSRnAに書くとき、常にこのビットを0に設定してください。

#### • ビット2 – UPEn : 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誤りがあり、その時点でパリティ検査が許可されていれば(UPMn1=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(UDRn)が読まれるまで有効です。UCSRnAに書くとき、常にこのビットを0に設定してください。

#### • ビット1 – U2Xn : 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。

#### • ビット0 – MPCMn : 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。MPCMn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MPCMn設定に影響されません。より多くの詳細情報については127頁の「複数プロセッサ通信動作」をご覧ください。

## USARTn制御/状態レジスタB (USARTn Control and Status Register B) UCSRnB

| ビット                 | 7                  | 6                  | 5                  | 4                 | 3                 | 2      | 1                 | 0                 |        |
|---------------------|--------------------|--------------------|--------------------|-------------------|-------------------|--------|-------------------|-------------------|--------|
| \$0A(\$2A) : (\$9A) | RXCIE <sub>n</sub> | TXCIE <sub>n</sub> | UDRIE <sub>n</sub> | RXEN <sub>n</sub> | TXEN <sub>n</sub> | UCSZn2 | RXB8 <sub>n</sub> | TXB8 <sub>n</sub> | UCSRnB |
| Read/Write          | R/W                | R/W                | R/W                | R/W               | R/W               | R/W    | R                 | R/W               |        |
| 初期値                 | 0                  | 0                  | 0                  | 0                 | 0                 | 0      | 0                 | 0                 |        |

### •ビット7 – RXCIE<sub>n</sub> : 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(UCSRnA)の受信完了(RXC<sub>n</sub>)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RXCIE<sub>n</sub>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、UCSRnAでRXC<sub>n</sub>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 – TXCIE<sub>n</sub> : 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(UCSRnA)の送信完了(TXC<sub>n</sub>)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TXCIE<sub>n</sub>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、UCSRnAでTXC<sub>n</sub>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 – UDRIE<sub>n</sub> : 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(UCSRnA)の送信データレジスタ空き(UDRE<sub>n</sub>)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UDRIE<sub>n</sub>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、UCSRnAでUDRE<sub>n</sub>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 – RXEN<sub>n</sub> : 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RXD<sub>n</sub>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(FE<sub>n</sub>)、オーバラン(DOR<sub>n</sub>)、パリティ誤り(UPE<sub>n</sub>)のフラグを無効にします。

### •ビット3 – TXEN<sub>n</sub> : 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TXD<sub>n</sub>ピンの標準ポート動作を無効にします。送信の禁止(TXEN<sub>n</sub>=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TXD<sub>n</sub>ポートの標準I/O機能を無効にしません。

### •ビット2 – UCSZn2 : データビット長選択2 (Character Size)

USART制御/状態レジスタC(UCSRnC)のUCSZn1,0ビットと組み合わせたUCSZn2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。

### •ビット1 – RXB8<sub>n</sub> : 受信データビット8 (Receive Data Bit 8)

RXB8<sub>n</sub>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。UDR<sub>n</sub>から下位ビットを読む前に読んでください。

### •ビット0 – TXB8<sub>n</sub> : 送信データビット8 (Transmit Data Bit 8)

TXB8<sub>n</sub>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。UDR<sub>n</sub>へ下位ビットを書く前に書いてください。

## USARTn制御/状態レジスタC (USARTn Control and Status Register C) UCSRnC

| ビット             | 7 | 6                  | 5     | 4     | 3                 | 2      | 1      | 0                  |        |
|-----------------|---|--------------------|-------|-------|-------------------|--------|--------|--------------------|--------|
| (\$95) : (\$9D) | – | UMSEL <sub>n</sub> | UPMn1 | UPMn0 | USBS <sub>n</sub> | UCSZn1 | UCSZn0 | UCPOL <sub>n</sub> | UCSRnC |
| Read/Write      | R | R/W                | R/W   | R/W   | R/W               | R/W    | R/W    | R/W                |        |
| 初期値             | 0 | 0                  | 0     | 0     | 0                 | 1      | 1      | 0                  |        |

**注:** このレジスタはATmega103互換動作で利用できません。

### •ビット7 – Res : 予約 (Reserved)

このビットは将来使用するために予約されています。将来のデバイスとの共通性を保つため、UCSRnCが書かれるとき、このビットは0が書かれなければなりません。

### •ビット6 – UMSEL<sub>n</sub> : USARTn動作選択 (USARTn Mode Select)

このビットは非同期と同期の動作種別のどちらかを選びます。

表77. USART動作選択

| UMSEL <sub>n</sub> | 動作種別  |
|--------------------|-------|
| 0                  | 非同期動作 |
| 1                  | 同期動作  |

•ビット5,4 – UPMn1,0 : 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、UPMn0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(UCSRnA)でパリティ誤り(UPEn)フラグが設定(1)されます。

表78. パリティ選択

| UPMn1 | UPMn0 | パリティ動作   |
|-------|-------|----------|
| 0     | 0     | 禁止       |
| 0     | 1     | (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 – USBSn : 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)。

表79. 停止ビット選択

| USBSn | 停止ビット数 |
|-------|--------|
| 0     | 1ビット   |
| 1     | 2ビット   |

•ビット2,1 – UCSZn1,0 : データビット長選択 (Character Size)

USART制御/状態レジスタB(UCSRnB)のUCSZn2ビットと組み合わせたUCSZn1,0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80. データビット長選択

| UCSZn2~0 | 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  |

•ビット0 – UCPOLn : クロック極性選択 (Clock Polarity)

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。UCPOLnビットは同期クロック(XCKn)、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81. XCKクロック極性選択

| UCPOLn | 送信データ変更 (TXDnピン出力) | 受信データ採取 (RXDnピン入力) |
|--------|--------------------|--------------------|
| 0      | XCKnの上昇端           | XCKnの下降端           |
| 1      | XCKnの下降端           | XCKnの上昇端           |

USARTnボーレートレジスタ (USARTn Baud Rate Register) UBRRnH, UBRRnL (UBRRn)

| ビット                 | 15     | 14     | 13     | 12     | 11      | 10      | 9      | 8      |        |
|---------------------|--------|--------|--------|--------|---------|---------|--------|--------|--------|
| (\$90) : (\$98)     | -      | -      | -      | -      | UBRRn11 | UBRRn10 | UBRRn9 | UBRRn8 | UBRRnH |
| Read/Write          | R      | R      | R      | R      | R/W     | R/W     | R/W    | R/W    |        |
| 初期値                 | 0      | 0      | 0      | 0      | 0       | 0       | 0      | 0      |        |
| ビット                 | 7      | 6      | 5      | 4      | 3       | 2       | 1      | 0      |        |
| \$09(\$29) : (\$99) | UBRRn7 | UBRRn6 | UBRRn5 | UBRRn4 | UBRRn3  | UBRRn2  | UBRRn1 | UBRRn0 | UBRRnL |
| Read/Write          | R/W    | R/W    | R/W    | R/W    | R/W     | R/W     | R/W    | R/W    |        |
| 初期値                 | 0      | 0      | 0      | 0      | 0       | 0       | 0      | 0      |        |

注: UBRRnHレジスタはATmega103互換動作で利用できません。

•ビット15~12 – Res : 予約 (Reserved Bits)

これらのビットは将来使用するために予約されています。将来のデバイスとの共通性のため、UBRRnHが書かれるとき、これらのビットは0が書かれなければなりません。

•ビット11~0 – UBRRn11~0 : ボーレート分周値 (USARTn Baud Rate Register)

これはUSARTnのボーレートを含む12ビットレジスタです。UBRRnHがUSARTnボーレートの上位4ビットを含み、UBRRnL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。UBRRnL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。



## ボーレート設定例

標準的な水晶発振子やセラミック振動子の周波数に対して最も一般的に使われる非同期動作のボーレートは表82のUBRR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:U2X=0)、1.0%(倍速:U2X=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(126頁の「非同期での動作範囲」参照)で誤差率が高いと、受信部は雑音耐性が低下します。誤差率は次式を使って計算されます。

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

表82. Xtal、ボーレート対UBRRH,UBRRL設定 (UBRR=UBRRH:UBRRL)

| ボーレート<br>(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<br>(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。以降の周波数は次頁へ続く。

(訳注) 原書では頁割された表毎に表番号82.~85.となっていますが、共通性から纏めて表82.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82 (続き). Xtal、ホーレート対UBRRH,UBRRL設定 (UBRR=UBRRH:UBRRL)

| ホーレート<br>(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    |       |

| ホーレート<br>(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 |       |

| ホーレート<br>(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。

## 2線直列インターフェース (TWI:Two-wire Serial Interface, I<sup>2</sup>C)

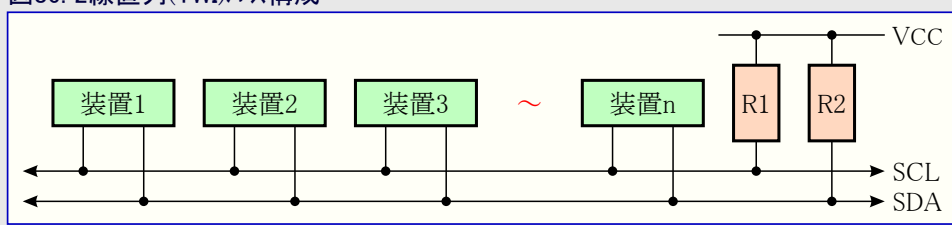
### 特徴

- 2本のバス信号線のみ必要な、単純ながら強力な柔軟な通信インターフェース
- 主装置動作と従装置動作の両方を支援
- 送信装置または受信装置として動作可能
- 7ビットのアドレス空間が128までの異なる従装置アドレスを許容
- 複数主装置の調停支援
- 400kHzまでのデータ転送速度
- 上昇/下降(スリュート)制限された出力駆動回路
- バス信号線のスパイクを排除する雑音除去回路
- 一斉呼び出しを含む完全に設定変更可能な従装置アドレスの支援
- AVRが休止形態の時のアドレス認証(一致)起動

### 2線直列インターフェースバスの定義

2線直列インターフェース(TWI)は代表的なマイクロコントローラ応用に対して理想的に適応されています。TWI通信規約は2本の双方向バス信号線、データ用1本(SDA)とクロック用1本(SCL)だけを使って128個までの異なる装置の相互接続をシステム設計者に許します。バスを実現するのに必要とされる外部ハードウェアはTWIバス信号線各々に1本ずつのプルアップ抵抗だけです。バスに接続した全ての装置は個別のアドレスを持ち、バス衝突を解決する機構は本質的にTWI通信規約で行います。

図86. 2線直列(TWI)バス構成



### TWI用語定義

次の定義は本章で度々使われます。

表86. TWI用語定義

| 用語   | 意味                               |
|------|----------------------------------|
| 主装置  | 送信の開始と終了を行う装置。主装置はSCLクロックも生成します。 |
| 従装置  | 主装置によって指定された装置。                  |
| 送信装置 | バス上にデータを送り出す装置。                  |
| 受信装置 | バスからデータを読み込む装置。                  |

### 電気的な相互接続

図86.で描かれたように両方のバス信号線はプルアップ抵抗を通して正供給電圧に接続されます。全てのTWI準拠装置のバス駆動部はオープンドレインかオープンコレクタです。これはインターフェースの動作のために重要なワイアードAND機能を実現します。TWIバス信号線のLowレベルは1つまたはより多くのTWI装置の0出力時に生成されます。Highレベルは全TWI装置がHi-Z出力時の出力で、プルアップ抵抗に信号線をHighへ引き上げさせます。どんなバス動作を許すのにも、TWIバスに接続した全てのAVRデバイスが電力供給されなければならないことに注意してください。

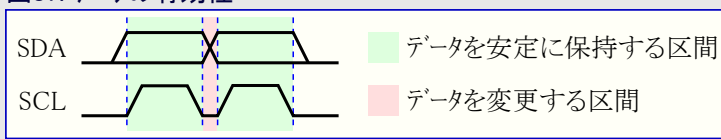
このバスに接続できる装置数は7ビットの従装置アドレス空間と400pFのバス容量制限によってのみ制限されます。TWIの電気的特性の詳細仕様は208頁の「2線直列インターフェース特性」で与えられます。そこで与えられる2組の異なる仕様は、1つがバス速度100kHz以下に関するもので、もう1つはバス速度400kHzまでに関して有効です。

### データ転送とフレーム形式

#### ビット転送

TWIバスに転送される各データビットはクロック信号線のパルスを伴います。データ信号線のレベルはクロック信号線がHighの時に安定していなければなりません。この規則の例外は開始条件と停止条件の生成だけです。

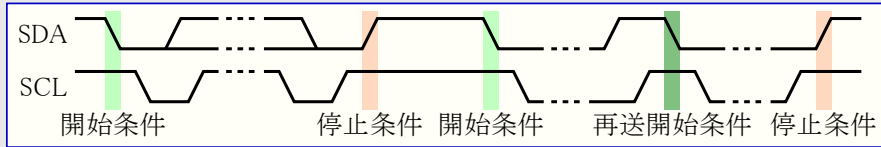
図87. データの有効性



## 開始条件と停止条件

主装置がデータ転送の開始と終了を行います。転送は主装置がバスに**開始条件**を起こすと開始され、主装置が**停止条件**を起こすと終了されます。**開始条件**と**停止条件**間はバスが使用中と考えられ、他の主装置はバスの制御獲得を試みるべきではありません。**開始条件**と**停止条件**間で新規**開始条件**が起こされると特別な状態が起きます。これは**再送開始条件**として引用され、主装置がバスの制御を手放さずに新規転送を始めたい時に使われます。**再送開始条件**後、バスは次の**停止条件**まで使用中と考えられます。これは開始動作についてと全く同じで、従って特記事項を除いて本データシートの残りに対して**開始条件**と**再送開始条件**の両方の記述に**開始条件**が使われます。下で描かれるように、**開始条件**と**停止条件**はSCL信号線がHighの時のSDA信号線のレベル変更によって指示されます。

図88. 開始条件、再送開始条件、停止条件 バス タイミング



## アドレス パケット形式

TWIバスに送信した全てのアドレス パケットは7ビットのアドレスビット、1ビットの方向(Read/Write)制御ビット、1ビットの応答ビットから成る9ビットです。方向(R/W)ビットが設定(1)されると読み出し操作が実行され、さもなければ書き込み操作が実行されるべきです。従装置がアドレス指定されたことを認証すると、9番目のSCL(ACK)周期でSDAをLowへ引くことによって確認応答すべきです。アドレス指定された従装置が忙しいまたはその他の理由で主装置の要求を扱えない場合、**確認応答(ACK)**クロック周期でSDA信号線をHighのままにすべきです。主装置はその後に**停止条件**または新規転送を始めるために**再送開始条件**を送出できます。従装置アドレスと方向(R/W)ビットから成るアドレス パケットは各々、SLA+RまたはSLA+Wと呼ばれます。

アドレスビットの最上位ビット(MSB)が最初に送信されます。従装置アドレスは設計者によって自由に割り当てられますが、アドレス0000 000は一斉呼び出し用に予約されています。

一斉呼び出しが起こされると、全従装置は**確認応答(ACK)**周期でSDA信号線をLowにすることによって応答すべきです。一斉呼び出しは主装置がシステム内のそれぞれの従装置に同じ通信内容を送信したい時に使われます。一斉呼び出しアドレスに続きW(方向が書き込み)ビットがバスに送信されると、一斉呼び出しに応答する設定の全ての従装置は、ACK周期でSDA信号線をLowに引き込みます。そして後続のデータ パケットは一斉呼び出しに確認応答した全従装置によって受信されます。一斉呼び出しアドレスに続くR(方向が読み出し)ビットの送信は、従装置それぞれが異なるデータの送信を始めた場合の衝突の原因となるので意味がないことに注意してください。

1111 xxx形式の全アドレスは将来の目的のために予約されるべきです(訳補: I<sup>2</sup>C規格のアドレス拡張他)。

## データ パケット形式

TWIバスに送信した全てのデータ パケットは1バイトのデータと1ビットの応答ビットから成る9ビットです。データ転送中、主装置はクロックと**開始条件**、**停止条件**を生成し、一方受信装置は受信に応答する責任があります。確認応答(ACK)は受信装置が9番目のSCL周期中にSDA信号線をLowに引き込むことによって示されます。受信装置がSDA信号線をHighのままにするとNACKを示します。受信装置が最終バイトを受信したとき、または何らかの理由でこれ以上のバイトを受信できないとき、最終バイト後に**NACK**を送ることによって送信装置へ通知すべきです。データバイトの最上位(MSB)ビットが最初に送信されます。

図89. アドレス パケット形式

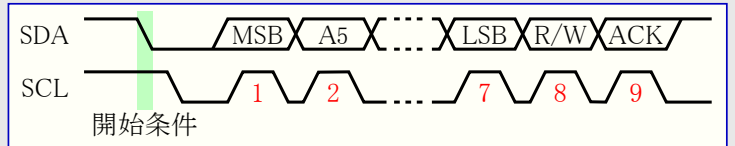
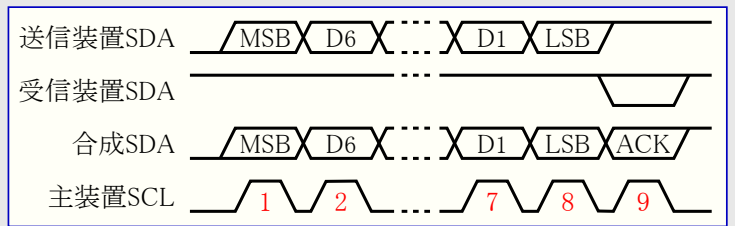


図90. データ パケット形式

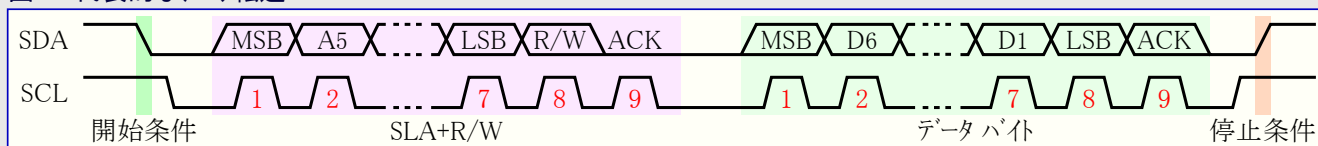


## 転送内でのアドレスパケットとデータパケットの組み合わせ

転送は基本的に開始条件、SLA+R/W、1つ以上のデータパケット、停止条件から成ります。開始条件に続く停止条件から成る空の通信内容は規則違反です。SCL信号線のワイアードANDが主装置と従装置間のハンドシェイクに使えることに注目してください。従装置はSCL信号線をLowに引き込むことによってSCLのLow期間を引き伸ばせます。これは主装置が従装置に対して速すぎるクロック速度設定、または従装置がデータ送信間の処理に追加時間を必要とする場合に有用です。従装置がSCLのLow期間を延長することは、主装置によって決められるSCLのHigh期間に影響しません。同様に従装置はSCLのデューティ比(Low期間)を延長することによってTWIデータ転送速度を落とせます。

図91.は代表的なデータ転送を示します。応用ソフトウェアによって実装されたソフトウェア規約に依存して、様々なデータがSLA+R/Wと停止条件間に送信できることに注意してください。

図91. 代表的なデータ転送



## 複数主装置バスシステムの調停と同期

TWI規約は多数主装置のバスシステムを許します。例え2つ以上の主装置が同時に送信を始めても、送信が通常のように続行することを保証するために特別な手段が講じられます。複数主装置のシステムでは2つの問題が起こります。

- 送信を完了するために1つの主装置だけを許す方法が実現されなければなりません。他の全ての主装置は(自身が行っている従装置)選択手順を失った(失敗した)ことに気付く時に送信を止めるべきです。この選択手順は調停(アビレーション)と呼ばれます。競合する主装置は調停(従装置選択)手順を失ったことに気付くと、勝ち残った主装置によってアドレス指定されるかどうかを調べるため、直ちに従装置動作へ切り替えるべきです。複数の主装置が同時に送信を始めた事実は従装置で検知できるべきではありません。換言すると、バスに転送されているデータが不正にされてはなりません。
- 違う主装置が異なるSCL周波数を使うかもしれません。同期確定手順で送信が続行するために、全主装置からの直列クロックを同期化する方法が考案されなければなりません。これは調停手順を容易にします。

バス信号線のワイアードANDはこれらの問題の両方の解決に使われます。全ての主装置からの直列クロックはワイアードANDされ、最短High期間の主装置の1つからに等しいHigh期間の合成クロックを生成します。合成クロックのLow期間は最長Low期間の主装置のLow期間に等しくなります。全ての主装置がSCL信号線を監視する、実際には合成SCL信号線がHighまたはLowになる時に各々SCLのHighとLow経過時間の計時を始めることに注意してください。

調停は全ての主装置がデータ出力後にSDA信号線を継続的に監視することによって実行されます。SDA信号線から読んだ値がその主装置の出力した値と一致しない場合、調停に敗れます。主装置がSDAにHigh値を出力し、同時に他の主装置がLow値を出力する時のみ調停に敗れるかもしれないことに注意してください。敗れた主装置は直ちに従装置動作へ移行し、勝ち残った主装置によってアドレス指定されるかを検査すべきです。SDA信号線はHighのままにすべきですが、敗れた主装置は現在のデータ若しくはアドレスパケットの最後までクロック信号を生成することを許されます。調停は唯一の主装置が残るまで継続され、多くのビットを必要とするかもしれません。多くの主装置が同じ従装置をアドレス指定しようとすると、調停はデータパケットに続くでしょう。

図92. 複数主装置間でのSCL同期化

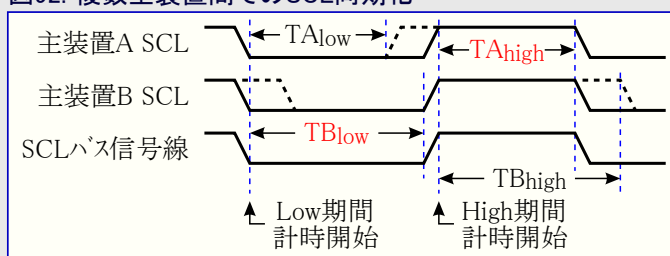
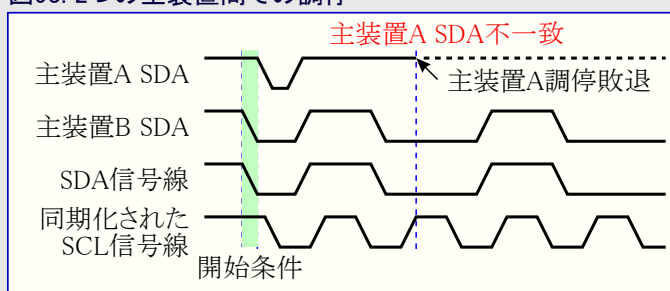


図93. 2つの主装置間での調停



調停が次の状態間で許されないことに注意してください。

- 再送開始条件とデータビット間
- 停止条件とデータビット間
- 再送開始条件と停止条件間

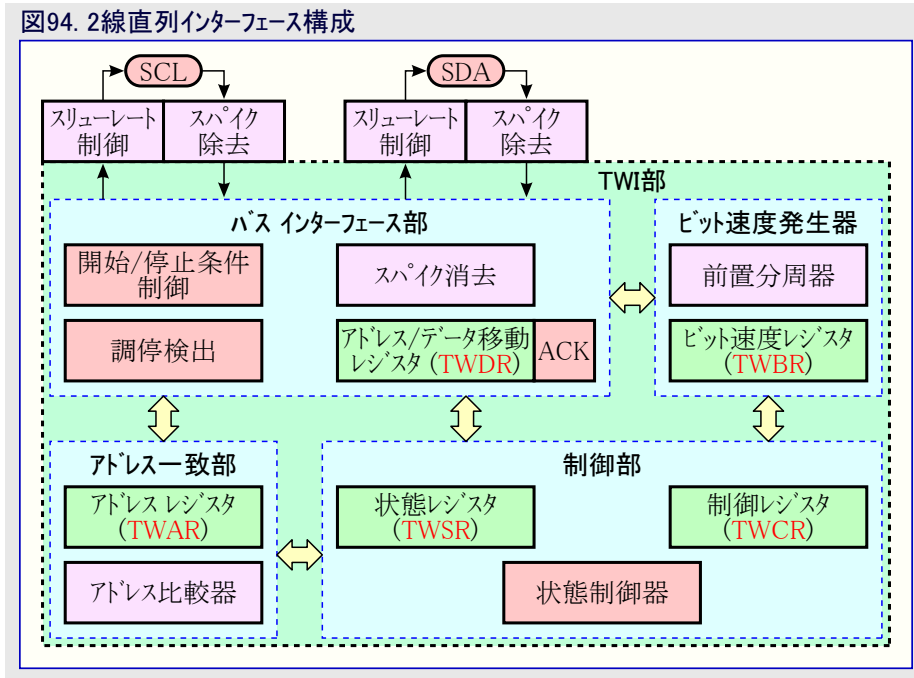
これらの違法な調停状態を決して起こさないように保証するのは使用者ソフトウェアの責任です。これは複数主装置システムでの全てのデータ転送は同じ構成、SLA+R/Wとデータパケットを使わなければならないことを意味します。言葉を変えると、全ての送信は同じデータパケット数を含まなければならない、さもなければ調停の結果は不定にされます。

(訳補) 同じデータパケット数とは、或る主装置が最後まで調停を継続し、他の主装置がパケットを残している場合を想定しています。



## TWI部の概要

図94.で示されるようにTWI部は様々な部分から成ります。赤文字で示された(訳注:原文は太線で描かれた)全てのレジスタはAVR データバスを通してアクセス可能です。



## SCLとSDAピン

これらのピンはAVR TWIをMCUシステムのその他とインターフェースします。出力駆動部はTWI仕様に適合させるためのスlewレート(上昇/下降)制限器を含みます。入力段は50nsよりも短いスパイクを除去するスパイク除去部を含みます。「入出力ポート」章で説明したようにAVRパッドの内部プルアップはSCLとSDAピンに対応するポートのビットを設定(=1)することによって許可できることに注目してください。内部プルアップはいくつかのシステムで外部抵抗の必要をなくせます。

## ビット速度発生器

この部分は主装置動作で動く時のSCL周期を制御します。SCL周期はTWIビット速度レジスタ(TWBR)とTWI状態レジスタ(TWSR)の前置分周器ビットの設定によって制御されます。従装置動作はビット速度や前置分周器設定と関係ありませんが、従装置でのCPUクロック周波数はSCL周波数よりも最低16倍高くなければなりません。従装置がSCLのLow期間を延長するかもしれず、これによって平均TWIバスクロック周波数が減少することに注意してください。SCL周波数は次式に従って生成されます。

$$\text{SCL周波数} = \frac{\text{CPUクロック周波数}}{16 + 2 \times (\text{TWBR}) \times \text{前置分周値}}$$

TWBR : TWIビット速度レジスタ値  
前置分周値 : TWI状態レジスタ内TWPSで指定(139頁の表87.参照)

注: プルアップ抵抗値はSCL周波数とバス信号線の容量性負荷に応じて選ばれるべきです。プルアップ抵抗の値については208頁の表133.をご覧ください。

## バスインターフェース部

この部分はデータとアドレスの移動レジスタ(TWDR)、開始条件/停止条件制御器、調停検出回路を含みます。TWDRは送信されるべきアドレスまたはデータバイト、若しくは受信したアドレスまたはデータバイトを含みます。8ビットのTWDRに加えてバスインターフェース部は送信されるべきまたは受信した(N)ACKビットを含むレジスタも含みます。この(N)ACKレジスタは応用ソフトウェアによって直接的にアクセスできません。けれどもTWI制御レジスタ(TWCR)を操作することにより、受信時に設定(1)または解除(0)できます。送信装置動作時、受信した(N)ACKビットの値はTWSRの値によって判定できます。

開始条件/停止条件制御器は開始条件、再送開始条件、停止条件の生成と検出に対して責任があります。開始条件/停止条件制御器はAVR MCUが主装置によってアドレス指定された場合にMCUを起動できる休止形態の1つの時でも、開始条件または停止条件を検出できます。

TWIが主装置として送信を始めると、調停検出ハードウェアは調停が進行中かを定めるために送信の試行を継続的に監視します。TWIが調停に敗れた場合、制御部に通知されます。その後正しい処置が行われ、適切な状態符号が生成されます。



## アドレス一致部

アドレス一致部は受信したアドレスバイトが**TWI アドレスレジスタ(TWAR)**の7ビットアドレスと一致するかを検査します。TWARで**一斉呼び出し検出許可(TWGCE)ビット**が1を書かれると、全ての到着アドレスビットは一斉呼び出しアドレスに対しても比較されます。アドレス一致で制御部は通知され、正しい処置を行うことを許します。TWIは**TWI制御レジスタ(TWCR)**の設定によって、そのアドレスへの応答をするかもしれないし、しないかもしれません。アドレス一致部はAVR MCUが主装置によってアドレス指定された場合にMCUを起動できる**休止形態**の1つの時でも、アドレスを比較できます。TWIがパワーダウン動作でのアドレス一致でCPUを起動中に他の割り込み(例えばINT0)が起こると、TWIは動作を停止し、アイドル状態で復帰します。これが何らかの問題の原因なら、パワーダウン動作へ移行する時にTWIアドレス一致だけが割り込みを許可されることを保証してください。

## 制御部

制御部はTWIバスを監視し、TWI制御レジスタ(TWCR)の設定に従った応答を生成します。応用に注意を要求する事象がTWIバスで起こると、**TWI割り込み要求フラグ(TWINT)**が有効にされます。次のクロック周期で、TWI状態レジスタ(TWSR)は事象を示す状態符号で更新されます。TWI割り込み要求フラグが有効にされる時にだけ、TWSRは適切な状態情報を含みます。他の全てのとき、TWSRは適切な状態情報が利用できないことを示す特別な状態符号を含みます。TWINTフラグが設定(1)されている限り、SCL信号線はLowに保たれます。これは続くTWI送信を許す前の(現状)処理完了を応用ソフトウェアに許します。

TWI割り込み要求フラグ(TWINT)は次の場合に設定(1)されます。

- **開始条件**または**再送開始条件**送信後
- **SLA+R/W**送信後
- アドレスバイト送信後
- 調停に敗れた後
- 自身の従装置アドレスまたは一斉呼び出しによってアドレス指定された後
- データバイト受信後
- 従装置として未だアドレス指定されている間の**停止条件**または**再送開始条件**受信後
- 不正な**開始条件**または**停止条件**のためバス異常が起きた時

## TWI用レジスタ

### TWI ビット速度レジスタ (TWI Bit Rate Register) TWBR

| ビット<br>(\$70) | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |      |
|---------------|-------|-------|-------|-------|-------|-------|-------|-------|------|
|               | TWBR7 | TWBR6 | TWBR5 | TWBR4 | TWBR3 | TWBR2 | TWBR1 | TWBR0 | TWBR |
| Read/Write    | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   |      |
| 初期値           | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |      |

#### • ビット7～0 – TWBR7～0 : TWI ビット速度選択 (TWI Bit Rate Register)

TWBRはビット速度発生器用の分周値を選びます。ビット速度発生器は主装置動作でのSCLクロック周波数を生成する周波数分周器です。ビット速度の計算については136頁の「ビット速度発生器」をご覧ください。

### TWI制御レジスタ (TWI Control Register) TWCR

| ビット<br>(\$74) | 7     | 6    | 5     | 4     | 3    | 2    | 1 | 0    |      |
|---------------|-------|------|-------|-------|------|------|---|------|------|
|               | TWINT | TWEA | TWSTA | TWSTO | TWWC | TWEN | – | TWIE | TWCR |
| Read/Write    | R/W   | R/W  | R/W   | R/W   | R    | R/W  | R | R/W  |      |
| 初期値           | 0     | 0    | 0     | 0     | 0    | 0    | 0 | 0    |      |

TWCRはTWI動作の制御に使われます。TWIの許可、バス上に開始条件を印加することによる主装置のアクセス開始、受信装置の応答生成、停止条件の生成、バスにデータを送出するためのTWIデータレジスタ(TWDR)への書き込み中のバスの一時停止制御に使われます。TWDRがアクセス不能の間にTWDRへ書き込もうとする場合の上書き発生も示します。

#### • ビット7 – TWINT : TWI割り込み要求フラグ (TWI Interrupt Flag)

このビットはTWIが現在の作業を終了し、応用ソフトウェアの応答が予測されるとき、ハードウェアによって設定(1)されます。TWI制御レジスタ(TWCR)のTWI割り込み許可(TWIE)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)されていると、MCUはTWI割り込みベクタへ飛びます。TWINTフラグが設定(1)の間中、SCLのLow期間は引き伸ばされます。TWINTフラグは論理1書き込みによってソフトウェアで解除(0)されなければなりません。このフラグが割り込みルーチンを実行する時に自動的に解除(0)されないことに注意してください。このフラグの解除(0)がTWI動作を始めるので、このフラグを解除(0)する前にTWIアドレスレジスタ(TWAR)、TWIデータレジスタ(TWDR)、TWI状態レジスタ(TWSR)への全てのアクセスが完了していなければならないことにも注意してください。

#### • ビット6 – TWEA : 確認応答(ACK)許可 (TWI Enable Acknowledge Bit)

TWEAビットは確認応答(ACKパルス)の生成を制御します。TWEAビットが1を書かれ、次の条件に合致すると、TWIバスにACKパルスが生成されます。

- 装置が自分用の従装置アドレスを受信した場合。
- TWIアドレスレジスタ(TWAR)の一斉呼び出し検出許可(TWGCE)ビットが設定(1)されている時に一斉呼び出しを受信した場合。
- 主受信装置または従受信装置動作でデータバイトを受信した場合。

TWEAビットに0を書くことによって一時的かつ仮想的に装置を2線直列バスから切り離すことができます。アドレス認証はその後に再びTWEAビットへ1を書くことによって再開できます。

#### • ビット5 – TWSTA : 開始(START)条件生成許可 (TWI START Condition Bit)

2線直列バスの主装置になることを欲する時に応用はTWSTAビットに1を書きます。TWIハードウェアはバスが利用可能かを検査し、開放ならばバスに開始条件を生成します。しかし、バスが未開放の場合、TWIは停止条件が検出されるまで待ち、その後にバス主権を要求する新規開始条件を生成します。TWSTAは開始条件が送出されてしまった時にソフトウェアで解除(0)されなければなりません。

#### • ビット4 – TWSTO : 停止(STOP)条件生成許可 (TWI STOP Condition Bit)

主装置動作でTWSTOビットに1を書くことが2線直列バスに停止条件を生成します。停止条件がバスで実行されると、TWSTOビットは自動的に解除(0)されます。従装置動作でのTWSTOビットの設定(1)は異常状態からの回復に使えます。これは停止条件を生成しますが、TWIは明確に指定されていない従装置動作に戻り、SCL、SDA信号線をHi-Z状態に開放します。

#### • ビット3 – TWWC : TWI上書き発生フラグ (TWI Write Collision Flag)

TWI割り込み要求フラグ(TWINT)が0の時にTWIデータレジスタ(TWDR)への書き込みを試みると、このTWWCフラグが設定(1)されます。このフラグはTWINTが1の時のTWDR書き込みによって解除(0)されます。

#### • ビット2 – TWEN : TWI動作許可 (TWI Enable Bit)

TWENビットはTWI動作を許可し、TWIインターフェースを活性(有効)にします。TWENが1を書かれると、TWIはSCL、SDAピンに接続したI/Oピンを制御できるようになり、スパイク濾波器とスルーレート制限器を許可します。このビットが0を書かれると、TWIがOFFにされ、どんな進行中の動作にも関係なく、全てのTWI送信が終了されます。

#### • ビット1 – Res : 予約 (Reserved Bit)

このビットは予約されており、常に0として読みます。

• **ビット0 – TWIE : TWI割り込み許可 (TWI 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)されていると、TWI割り込み要求フラグ(TWINT)が1である限り、TWI割り込み要求が活性に(発生)されます。

**TWI状態レジスタ (TWI Status Register) TWSR**

| ビット (\$71) | 7    | 6    | 5    | 4    | 3    | 2 | 1     | 0     | TWSR |
|------------|------|------|------|------|------|---|-------|-------|------|
|            | TWS7 | TWS6 | TWS5 | TWS4 | TWS3 | – | TWPS1 | TWPS0 |      |
| Read/Write | R    | R    | R    | R    | R    | R | R/W   | R/W   |      |
| 初期値        | 1    | 1    | 1    | 1    | 1    | 0 | 0     | 0     |      |

• **ビット7～3 – TWS7～3 : TWI状態 (TWI Status)**

これら5ビットはTWI論理回路と2線直列バスの状態を反映します。各種状態符号は本章の後ろで記述されます。TWSRから読む値が5ビットの状態符号と2ビットの前置分周値の両方を含むことに注意してください。応用設計者は状態ビットを検査する時に前置分周器ビットを0で隠すべきです。これは前置分周器設定に関係なく状態検査を行います。この手法は特記事項を除いてこのデータシート内で使われます。

• **ビット2 – Res : 予約 (Reserved Bit)**

このビットは予約されており、常に0として読みます。

• **ビット1,0 – TWPS1,TWPS0 : TWI前置分周器選択 (TWI Prescaler Bits)**

これらのビットは読み書きでき、ビット速度の前置分周器を制御します。

ビット速度を計算するには136頁の「ビット速度発生器」をご覧ください。TWPS1,0の値はこの式で使われます。

表87. TWIビット速度前置分周器選択

|       |   |   |    |    |
|-------|---|---|----|----|
| TWPS1 | 0 | 0 | 1  | 1  |
| TWPS0 | 0 | 1 | 0  | 1  |
| 分周値   | 1 | 4 | 16 | 64 |

**TWIデータレジスタ (TWI Data Register) TWDR**

| ビット (\$73) | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    | TWDR |
|------------|------|------|------|------|------|------|------|------|------|
|            | TWD7 | TWD6 | TWD5 | TWD4 | TWD3 | TWD2 | TWD1 | TWD0 |      |
| Read/Write | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  |      |
| 初期値        | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1    |      |

送信動作でのTWDRは送信されるべき次のデータバイトを含みます。受信動作でのTWDRは最後に受信したバイトを含みます。TWDRはTWIがバイトを移動する手順でない間に書き込み可能です。これはTWI制御レジスタ(TWCR)のTWI割り込み要求フラグ(TWINT)がハードウェアによって設定(1)されると起きます。最初のTWI割り込みが起こる前にデータレジスタ(TWDR)は使用者によって初期化できないことに注意してください。TWDRのデータはTWINTが安定して設定(1)されている限り存続します。データが移動出力される間、バスのデータが同時に移動入力されます。TWI割り込みによる休止形態からの起動後を除いて、TWDRは常にバスに現れる最後のバイトを含みます。この例外の場合のTWDRの内容は不定です。バス調停に敗れた場合の主装置から従装置への移行でもデータは失われません。確認応答(ACK)ビットの扱いはTWI論理回路によって自動的に制御され、CPUはACKビットを直接的にアクセスできません。

• **ビット7～0 – TWD7～0 : TWIデータ (TWI Data)**

これら8ビットは送信されるべき次のデータバイト、または2線直列バスで最後に受信したデータバイトを構成します。

**TWI(従装置)アドレスレジスタ (TWI (Slave) Address Register) TWAR**

| ビット (\$72) | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0     | TWAR |
|------------|------|------|------|------|------|------|------|-------|------|
|            | TWA6 | TWA5 | TWA4 | TWA3 | TWA2 | TWA1 | TWA0 | TWGCE |      |
| Read/Write | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  | R/W   |      |
| 初期値        | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 0     |      |

TWARは従装置の送受信装置として設定した時にTWIが応答する7ビット従装置アドレスを(TWAR上位7ビットに)設定されるべきで、主装置動作では必要とされません。複数主装置のシステムでは、他の主装置によって従装置としてアドレス指定され得る主装置に於いて、TWARは設定されなければなりません。

TWARの最下位ビット(TWGCE)は一斉呼び出しアドレス(\$00)認証の許可に使われます。これらは受信した直列アドレスで従装置アドレス(と許可ならば一斉呼び出しアドレス)を捜す関連アドレス比較器です。一致が見つかるとう割り込み要求が生成されます。

• **ビット7～1 – TWA6～0 : TWI従装置アドレス (TWI (Slave) Address)**

これら7ビットはTWI部の従装置アドレスを構成します。

• **ビット0 – TWGCE : 一斉呼び出し検出許可 (TWI General Call Recognition Enable Bit)**

設定(1)なら、このビットは2線直列バスを伝って与えられる一斉呼び出しの認証(検出)を許可します。



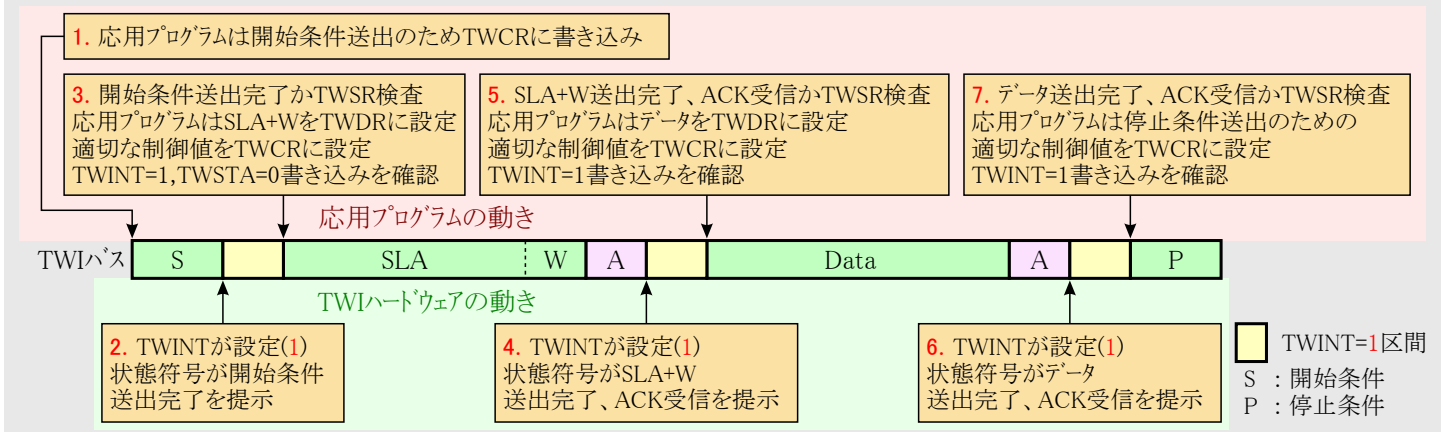
## TWIの使用法

AVR TWIはバト志向で割り込みが基本です。割り込みはバトの受信や**開始条件**の送出のような全てのバスの事象後に起こります。TWIは割り込みが基本のため、アプリケーションはTWIバト転送中に他の操作を続行するために開放されます。**ステータスレジスタ(SREG)の全割り込み許可(0)ビット**と共に**TWI制御レジスタ(TWCR)のTWI割り込み許可(TWIE)ビット**は、TWCRのTWI割り込み要求フラグ(TWINT)の設定(1)が割り込み要求を発生すべきかどうか決めることをアプリケーションに許します。TWIEビットが解除(0)されると、アプリケーションはTWIバスの動きを検知するためにTWINTフラグをポーリングしなければなりません。

TWINTフラグが設定(1)されると、TWIは動作を終え、アプリケーションの応答を待ちます。この場合、TWI状態レジスタ(TWSR)はTWIバスの現在の状態を示す値を含みます。そしてアプリケーションはTWCRとTWDRの操作により、TWIが次のTWIバス周期で何を行うべきかを決定できます。

図95.はアプリケーションがTWIハードウェアにどうインターフェースできるかの簡単な例です。この例では主装置が単一データバトを従装置に送信しようとしています。この内容はかなり大雑把ですので、より詳細な説明が本項の後に続きます。希望した動きを実現する簡単なコード例も示されます。

図95. 代表的な送信でのアプリケーションとTWIのインターフェース



1. TWI送信の最初の段階は**開始条件**を送出することです。これはTWIハードウェアに**開始条件**送出を命じる特別な値をTWCR内に書くことによって行います。どんな値を書くかは後で記述されます。けれども、書かれる値でTWINTビットが設定(1)されることが重要です。TWINTへの1書き込みは、このフラグを解除(0)します。TWCRでTWINTビットが設定(1)されている限り、TWIはどんな動作も始めません。アプリケーションがTWINTを解除(0)した後、TWIは直ちに**開始条件**の送出を始めます。
2. 開始条件が送出されてしまうと、TWCRでTWINTフラグが設定(1)され、TWSRは**開始条件**が正常に送出されてしまったことを示す状態符号に更新されます。
3. アプリケーションは**開始条件**が正常に送信されたのを確認するためにTWSRの値を直ぐに検査すべきです。TWSRが他を示している場合、アプリケーションは異常ルーチン呼び出すような或る特別な動きを講じるかもしれません。期待した状態符号だと仮定すると、アプリケーションはTWDRに**SLA+W**を設定しなければなりません。TWDRがアドレスとデータの両方に使われることを思い出してください。TWDRが希望した**SLA+W**に設定されてしまった後、TWDRにある**SLA+W**の送信をTWIハードウェアへ命じる特別な値がTWCRに書かれなければなりません。どんな値を書くかは後で記述されます。けれども書かれる値でTWINTビットが設定(1)されることが重要です。TWINTへの1書き込みがこのフラグを解除(0)します。TWCRでTWINTビットが設定(1)されている限り、TWIはどんな動作も始めません。アプリケーションがTWINTを解除(0)した後、TWIは直ちにアドレスパケットの送信を始めます。
4. アドレスパケットが送信されてしまうと、TWCRでTWINTフラグが設定(1)され、TWSRはアドレスパケットが正常に送信されたことを示す状態符号に更新されます。この状態符号は従装置がパケットに応答したかどうかも反映します。
5. アプリケーションはアドレスパケットが正常に送信され、期待された**ACK**ビット値であることを確認するためにTWSRの値を直ぐに検査すべきです。TWSRが他を示している場合、アプリケーションは異常ルーチン呼び出すような或る特別な動きを講じるかもしれません。期待した状態符号だと仮定すると、アプリケーションはTWDRにデータを設定しなければなりません。その後、TWDRにあるデータパケットの送信をTWIハードウェアへ命じる特別な値がTWCRに書かれなければなりません。どんな値を書くかは後で記述されます。けれども書かれる値でTWINTビットが設定(1)されることが重要です。TWINTへの1書き込みがこのフラグを解除(0)します。TWCRでTWINTビットが設定(1)されている限り、TWIはどんな動作も始めません。アプリケーションがTWINTを解除(0)した後、TWIは直ちにデータパケットの送信を始めます。
6. データパケットが送信されてしまうと、TWCR内のTWINTフラグが設定(1)され、TWSRはデータパケットが正常に送信されたことを示す状態符号に更新されます。この状態符号は従装置がパケットに応答したかどうかも反映します。
7. アプリケーションはデータパケットが正常に送信され、期待された**ACK**ビットの値であることを確認するためにTWSRの値を直ぐに検査すべきです。TWSRが他を示している場合、アプリケーションは異常ルーチン呼び出すような或る特別な動きを講じるかもしれません。期待した状態符号だと仮定すると、アプリケーションは**停止条件**の送出をTWIハードウェアへ命じる特別な値をTWCRに書かなければなりません。どんな値を書くかは後で記述されます。けれども書かれる値でTWINTビットが設定(1)されることが重要です。TWINTへの1書き込みがこのフラグを解除(0)します。TWCRでTWINTビットが設定(1)されている限り、TWIはどんな動作も始めません。アプリケーションがTWINTを解除(0)した後、TWIは直ちに**停止条件**の送出を始めます。**停止条件**が送出されてしまった後にTWINTが設定(1)されないことに注意してください。



この例は簡単とはいえ、全てのTWI送信に関係した原理を示しています。これらは次のように要約できます。

- TWIが動作を終了して応用(ソフトウェア)の反応を予想する時に**TWINTフラグ**が設定(1)されます。SCL信号線はTWINTが解除(0)されるまでLowに引き込まれます。
- TWINTフラグが設定(1)されると、使用者は次のTWIバス周期に関連した値で、(必要な)全てのTWIレジスタを更新しなければなりません。例で示されるように**TWDR**は次のTWIバス周期で送信されるべき値を設定されなければなりません。
- (必要な)全てのTWIレジスタを更新し、その他保留中の応用ソフトウェアの処理が完了してしまった後に**TWCR**が書かれます。TWCR書き込み時、TWINTビットが設定(1)されるべきです。TWINTへの1書き込みはこのフラグを解除(0)します。TWCR設定によってどの動作が指定されても、TWIはその(TWINT=0)後に実行を始めます。

次にアセンブリ言語とC言語の実装例が与えられます。以下のコードは例えばインクルードファイルの使用により、様々な定義が作成されている前提であることに注意してください。

|    | アセンブリ言語プログラム例   | C言語プログラム例                                       | 注釈   |
|----|---|---|--|
| 1. | LDI R16, (1<<TWINT)   (1<<TWSTA)<br>  (1<<TWEN)<br>OUT TWCR, R16                    | TWCR = (1<<TWINT)   (1<<TWSTA)<br>  (1<<TWEN);  | ;開始条件送出  |
| 2. | WAIT1: IN R16, TWCR<br>SBRS R16, TWINT<br>RJMP WAIT1                                | while (!(TWCR & (1<<TWINT)));                   | ;TWINT=1まで待機<br>;(開始条件送出完了待機)                                    |
| 3. | IN R16, TWSR<br>ANDI R16, \$F8<br>CPI R16, START<br>BRNE ERROR                      | if ((TWSR & 0xF8) != START)<br>ERROR();         | ;TWI状態レジスタ値検査<br>;前置分周選択ビットの遮蔽<br>;STARTと異なる状態符号で<br>;異常処理へ      |
|    | LDI R16, SLA_W<br>OUT TWDR, R16<br>LDI R16, (1<<TWINT)   (1<<TWEN)<br>OUT TWCR, R16 | TWDR = SLA_W;<br>TWCR = (1<<TWINT)   (1<<TWEN); | ;TWDRにSLA+W設定<br>;アドレス送信開始のため<br>;TWCRのTWINTを解除(0)               |
| 4. | WAIT2: IN R16, TWCR<br>SBRS R16, TWINT<br>RJMP WAIT2                                | while (!(TWCR & (1<<TWINT)));                   | ;TWINT=1まで待機<br>;(SLA+W送出完了と<br>;ACK/NACK受信完了待機)                 |
| 5. | IN R16, TWSR<br>ANDI R16, \$F8<br>CPI R16, MT_SLA_ACK<br>BRNE ERROR                 | if ((TWSR & 0xF8) != MT_SLA_ACK)<br>ERROR();    | ;TWI状態レジスタ値検査<br>;前置分周選択ビットの遮蔽<br>;MT_SLA_ACKと違う状態符号で<br>;異常処理へ  |
|    | LDI R16, DATA<br>OUT TWDR, R16<br>LDI R16, (1<<TWINT)   (1<<TWEN)<br>OUT TWCR, R16  | TWDR = DATA;<br>TWCR = (1<<TWINT)   (1<<TWEN);  | ;TWDRにデータ設定<br>;データ送信開始のため<br>;TWCRのTWINTを解除(0)                  |
| 6. | WAIT3: IN R16, TWCR<br>SBRS R16, TWINT<br>RJMP WAIT3                                | while (!(TWCR & (1<<TWINT)));                   | ;TWINT=1まで待機<br>;(データ送出完了と<br>;ACK/NACK受信完了待機)                   |
| 7. | IN R16, TWSR<br>ANDI R16, \$F8<br>CPI R16, MT_DATA_ACK<br>BRNE ERROR                | if ((TWSR & 0xF8) != MT_DATA_ACK)<br>ERROR();   | ;TWI状態レジスタ値検査<br>;前置分周選択ビットの遮蔽<br>;MT_DATA_ACKと違う状態符号で<br>;異常処理へ |
|    | LDI R16, (1<<TWINT)   (1<<TWSTO)<br>  (1<<TWEN)<br>OUT TWCR, R16                    | TWCR = (1<<TWINT)   (1<<TWSTO)<br>  (1<<TWEN);  | ;停止条件送出  |

注: 6頁の「コード例について」をご覧ください。



## 転送種別

TWIは4つの主な動作種別の1つで動けます。これらは送信主装置(MT)、受信主装置(MR)、送信従装置(ST)、受信従装置(SR)と名付けられます。これら種別の多くは同じ応用に使えます。例えば、TWI方式のEEPROM内にデータを書くのにTWIはMT動作を、EEPROMからデータを読み戻すのにMR動作を使えます。システム内に他の主装置が存在する場合、それらのいくつかがTWIにデータを送信するかもしれず、するとSR動作が使われるでしょう。どの動作種別が適正かを決めるのは応用ソフトウェアです。

次項はこれら動作種別の各々を記述します。起こり得る状態符号は各動作種別のデータ伝送詳細図に沿って示されます。これらの図は次の略号を含みます。

|           |                           |
|-----------|---------------------------|
| S         | 開始(START)条件               |
| Rs        | 再送開始(REPEATED START)条件    |
| R         | 読み出し指定ビット (SDA=High)      |
| W         | 書き込み指定ビット (SDA=Low)       |
| A         | 確認応答(ACK)ビット (SDA=Low)    |
| $\bar{A}$ | 非確認応答(NACK)ビット (SDA=High) |
| Data      | 8ビット データ バイト              |
| P         | 停止(STOP)条件                |
| SLA       | 従装置アドレス                   |

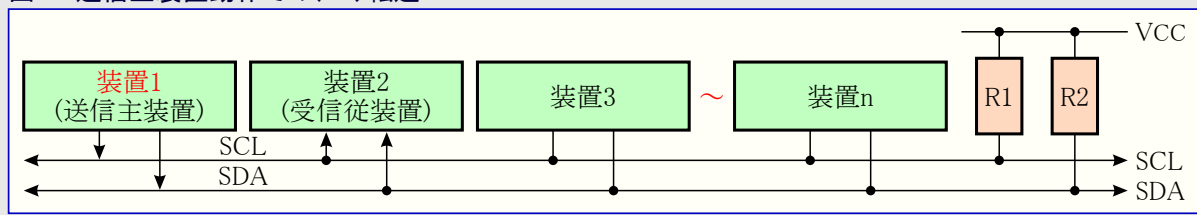
**図97.~103.**内の楕円(訳注:原文は円)はTWI制御レジスタ(TWCR)のTWI割り込み要求フラグ(TWINT)が設定(1)されたことを示すために使われます。この楕円内の番号は前置分周選択ビットが0で遮蔽されたTWI状態レジスタ(TWSR)に保持した状態符号を表します。これら位置での動きはTWI転送の継続または完了が応用(ソフトウェア)によって行われなければなりません。TWI転送はソフトウェアによってTWINTフラグが解除(0)されるまで一時停止されます。

TWI割り込み要求フラグ(TWINT)が設定(1)される時のTWI状態レジスタ(TWSR)の状態符号は適切なソフトウェア動作を決めるのに使われます。各状態符号に対する必要なソフトウェア動作や後続の直列転送の詳細は表88.~91.で与えられます。これらの表に於いて前置分周選択ビットが0で遮蔽されていることに注意してください。

## 送信主装置動作

送信主装置動作では何バイトかのデータが受信従装置へ送信されます(図96参照)。主装置動作へ移行するには**開始条件**が送出されなければなりません。それに続くアドレスパケットの形式が送信主装置または受信主装置のどちらへ移行すべきかを決めます。**SLA+W**が送信されると送信主装置(MT)へ移行し、**SLA+R**が送信されると受信主装置(MR)へ移行します。本項で言及する全ての状態符号は**前置分周選択ビット**が0か、または0で遮蔽されることが前提です。

図96. 送信主装置動作でのデータ転送



**開始条件**はTWCRに次の値を書くことによって送出されます。

| ビット  | 7     | 6    | 5     | 4     | 3    | 2    | 1 | 0    |
|------|-------|------|-------|-------|------|------|---|------|
| TWCR | TWINT | TWEA | TWSTA | TWSTO | TWWC | TWEN | - | TWIE |
| 設定値  | 1     | X    | 1     | 0     | X    | 1    | 0 | X    |

TWENは2線直列インターフェース(TWI)を許可するために設定(1)されなければなりません。TWSTAは**開始条件**を送出するために1を書かれねばならず、TWINTはTWINTフラグを解除(0)するために1を書かれなければなりません。そしてTWIは2線直列バスを検査し、バスが開放になると直ぐに**開始条件**を生成します。**開始条件**が送出されてしまった後、TWINTフラグがハードウェアによって設定(1)され、TWSRの状態符号が\$08(表88参照)になります。送信主装置へ移行するには**SLA+W**が送信されなければなりません。これはTWDRに**SLA+W**を書くことによって行います。その後、転送を継続するためにTWINTビットは(1の書き込みによって)解除(0)されるべきです。これはTWCRに次の値を書くことによって成し遂げられます。

| ビット  | 7     | 6    | 5     | 4     | 3    | 2    | 1 | 0    |
|------|-------|------|-------|-------|------|------|---|------|
| TWCR | TWINT | TWEA | TWSTA | TWSTO | TWWC | TWEN | - | TWIE |
| 設定値  | 1     | X    | 0     | 0     | X    | 1    | 0 | X    |

**SLA+W**が送信されて応答ビットが受信されてしまうと、TWINTが再び設定(1)され、TWSRの状態符号の数値が利用可能になります。主装置動作で可能性のある状態符号は\$18,\$20,\$38です。これら状態符号の各々に対する適切な動作は表88で詳述されます。

**SLA+W**が正常に送信されてしまうと、データパケットが送信されるべきです。これはTWDRにデータバイトを書くことによって行われます。TWDRはTWINTが1の時にだけ書かれなければなりません。さもなければ、そのアクセスは破棄され、TWCRで**上書き発生(TWWC)フラグ**が設定(1)されます。TWDR更新後、転送を継続するためにTWINTビットは(1の書き込みによって)解除(0)されるべきです。これはTWCRに次の値を書くことによって成し遂げられます。

| ビット  | 7     | 6    | 5     | 4     | 3    | 2    | 1 | 0    |
|------|-------|------|-------|-------|------|------|---|------|
| TWCR | TWINT | TWEA | TWSTA | TWSTO | TWWC | TWEN | - | TWIE |
| 設定値  | 1     | X    | 0     | 0     | X    | 1    | 0 | X    |

最後のバイトが送られてしまうまでこの手順が繰り返され、この転送は**停止条件**または**再送開始条件**を生成することによって終了されます。**停止条件**はTWCRに次の値を書くことによって生成されます。

| ビット  | 7     | 6    | 5     | 4     | 3    | 2    | 1 | 0    |
|------|-------|------|-------|-------|------|------|---|------|
| TWCR | TWINT | TWEA | TWSTA | TWSTO | TWWC | TWEN | - | TWIE |
| 設定値  | 1     | X    | 0     | 1     | X    | 1    | 0 | X    |

**再送開始条件**はTWCRに次の値を書くことによって生成されます。

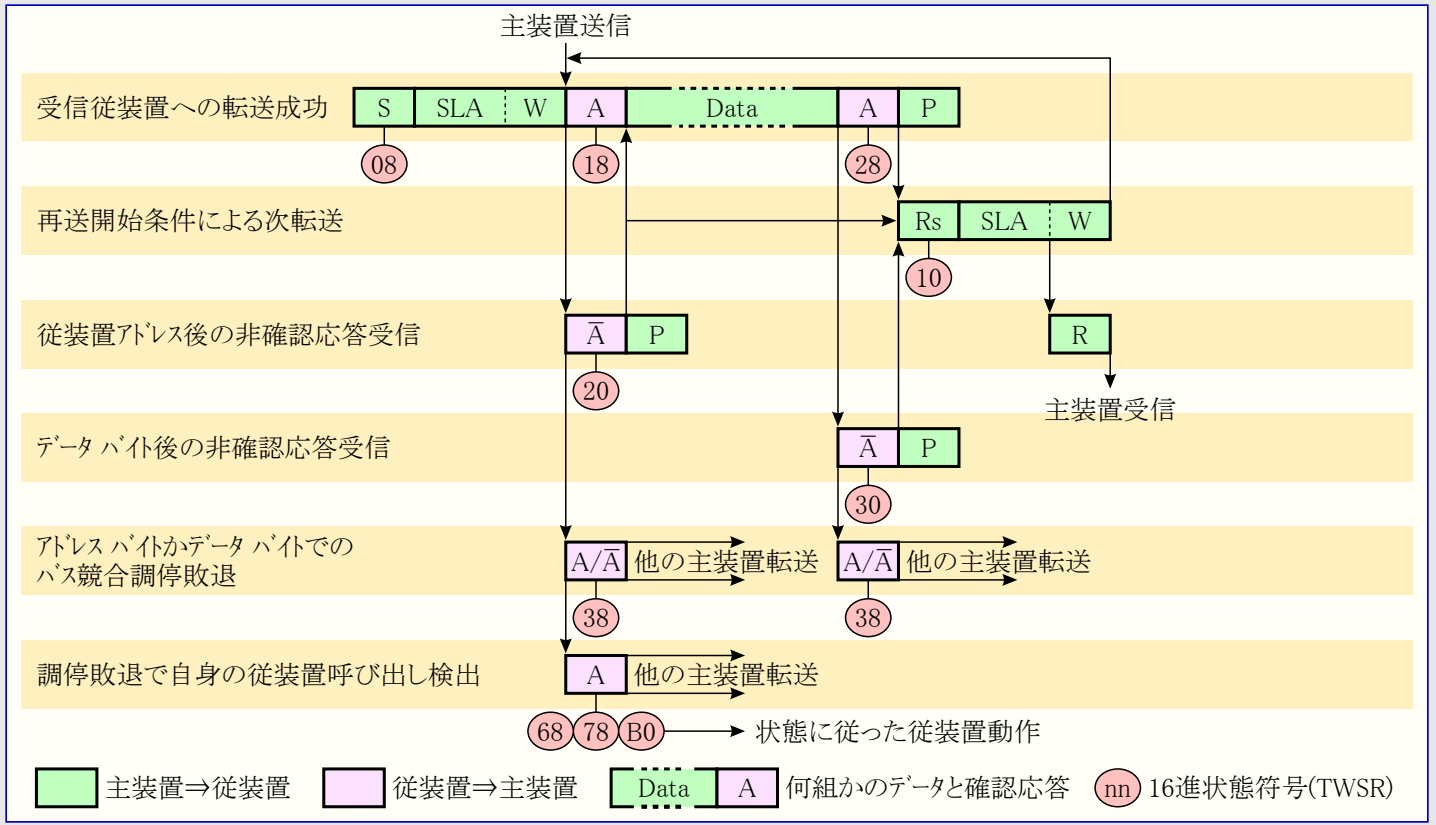
| ビット  | 7     | 6    | 5     | 4     | 3    | 2    | 1 | 0    |
|------|-------|------|-------|-------|------|------|---|------|
| TWCR | TWINT | TWEA | TWSTA | TWSTO | TWWC | TWEN | - | TWIE |
| 設定値  | 1     | X    | 1     | 0     | X    | 1    | 0 | X    |

**再送開始条件**(状態符号\$10)後、2線直列インターフェースは**停止条件**を送出せずに再び同じ従装置または新しい従装置にアクセスできます。**再送開始条件**は主装置がバスの制御を失わずに送信主装置、受信主装置間の切り替えを可能にします(訳注:原文では従装置も含まれていますが、基本動作に対して不適切なため削除しました)。

表88. 送信主装置動作の状態符号 (注: TWSRの前置分周選択ビットは0の前提)

| 状態符号 (TWSR) | 直前の動作とバスの状態                | ソフトウェアの対応 |        |       |       |      | TWCR設定によるハードウェア動作   |
|-------------|----------------------------|-----------|--------|-------|-------|------|---------------------|
|             |                            | TWDR操作    | TWCR設定 |       |       |      |                     |
|             |                            |           | TWSTA  | TWSTO | TWINT | TWEA |                     |
| \$08        | 開始条件送信                     | SLA+W設定   | 0      | 0     | 1     | X    | SLA+W送信、ACKかNACK受信  |
| \$10        | 再送開始条件送信                   | SLA+W設定   | 0      | 0     | 1     | X    | SLA+W送信、ACKかNACK受信  |
|             |                            | SLA+R設定   | 0      | 0     | 1     | X    | SLA+R送信、受信主装置動作へ移行  |
| \$18        | SLA+W送信<br>ACK受信           | データ設定     | 0      | 0     | 1     | X    | データ送信、ACKかNACK受信    |
|             |                            | なし        | 1      | 0     | 1     | X    | 再送開始条件送信            |
|             |                            |           | 0      | 1     | 1     | X    | 停止条件送信、TWSTO=0      |
|             |                            |           | 1      | 1     | 1     | X    | 停止条件→開始条件送信、TWSTO=0 |
| \$20        | SLA+W送信<br>NACK受信          | データ設定     | 0      | 0     | 1     | X    | データ送信、ACKかNACK受信    |
|             |                            | なし        | 1      | 0     | 1     | X    | 再送開始条件送信            |
|             |                            |           | 0      | 1     | 1     | X    | 停止条件送信、TWSTO=0      |
|             |                            |           | 1      | 1     | 1     | X    | 停止条件→開始条件送信、TWSTO=0 |
| \$28        | データバイト送信<br>ACK受信          | データ設定     | 0      | 0     | 1     | X    | データ送信、ACKかNACK受信    |
|             |                            | なし        | 1      | 0     | 1     | X    | 再送開始条件送信            |
|             |                            |           | 0      | 1     | 1     | X    | 停止条件送信、TWSTO=0      |
|             |                            |           | 1      | 1     | 1     | X    | 停止条件→開始条件送信、TWSTO=0 |
| \$30        | データバイト送信<br>NACK受信         | データ設定     | 0      | 0     | 1     | X    | データ送信、ACKかNACK受信    |
|             |                            | なし        | 1      | 0     | 1     | X    | 再送開始条件送信            |
|             |                            |           | 0      | 1     | 1     | X    | 停止条件送信、TWSTO=0      |
|             |                            |           | 1      | 1     | 1     | X    | 停止条件→開始条件送信、TWSTO=0 |
| \$38        | SLA+W, データバイトで<br>バス競合調停敗退 | なし        | 0      | 0     | 1     | X    | バス開放、未指定従装置動作へ移行    |
|             |                            |           | 1      | 0     | 1     | X    | バス開放時に開始条件送信        |

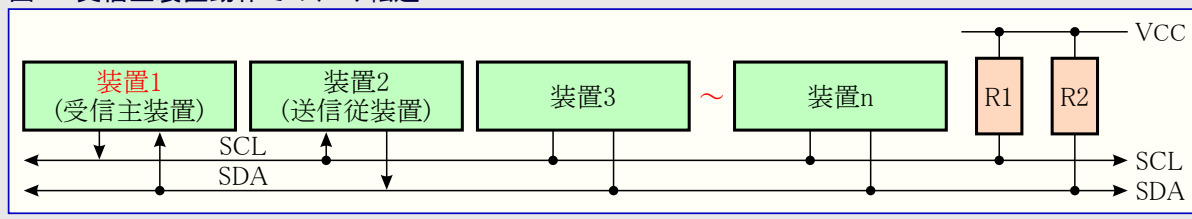
図97. 送信主装置動作の形式と状態



## 受信主装置動作

受信主装置動作では何バイトかのデータが送信従装置から受信されます(図98参照)。主装置動作へ移行するには**開始条件**が送出されなければなりません。それに続くアドレスパケットの形式が送信主装置または受信主装置のどちらへ移行すべきかを決めます。**SLA+W**が送信されると送信主装置(MT)へ移行し、**SLA+R**が送信されると受信主装置(MR)へ移行します。本項で言及する全ての状態符号は**前置分周選択ビット**が0か、または0で遮蔽されることが前提です。

図98. 受信主装置動作でのデータ転送



**開始条件**はTWCRに次の値を書くことによって送出されます。

| ビット  | 7     | 6    | 5     | 4     | 3    | 2    | 1 | 0    |
|------|-------|------|-------|-------|------|------|---|------|
| TWCR | TWINT | TWEA | TWSTA | TWSTO | TWWC | TWEN | - | TWIE |
| 設定値  | 1     | X    | 1     | 0     | X    | 1    | 0 | X    |

TWENは2線直列インターフェース(TWI)を許可するために設定(1)されなければなりません。TWSTAは**開始条件**を送出するために1を書かれねばならず、TWINTはTWINTフラグを解除(0)するために1を書かれなければなりません。そしてTWIは2線直列バスを検査し、バスが開放になると直ぐに**開始条件**を生成します。**開始条件**が送出されてしまった後、TWINTフラグがハードウェアによって設定(1)され、TWSRの状態符号が\$08(表89参照)になります。受信主装置へ移行するには**SLA+R**が送信されなければなりません。これは**TWDR**に**SLA+R**を書くことによって行います。その後、転送を継続するためにTWINTビットは(1)の書き込みによって解除(0)されるべきです。これはTWCRに次の値を書くことによって成し遂げられます。

| ビット  | 7     | 6    | 5     | 4     | 3    | 2    | 1 | 0    |
|------|-------|------|-------|-------|------|------|---|------|
| TWCR | TWINT | TWEA | TWSTA | TWSTO | TWWC | TWEN | - | TWIE |
| 設定値  | 1     | X    | 0     | 0     | X    | 1    | 0 | X    |

**SLA+R**が送信されて応答ビットが受信されてしまうと、TWINTが再び設定(1)され、TWSRの状態符号の数値が利用可能になります。主装置動作で可能性のある状態符号は\$38,\$40,\$48です。これら状態符号の各々に対する適切な動作は表89で詳述されます。

ハードウェアによってTWINTフラグが設定(1)されると、受信したデータがTWDRから読めます。この手順は最後のバイトが受信されてしまうまで繰り返されます。最後のバイトが受信されてしまった後、受信主装置は最後に受信したデータバイト後の**NACK**送信によって送信従装置へ通知すべきです。この転送は**停止条件**または**再送開始条件**を生成することによって終了されます。**停止条件**はTWCRに次の値を書くことによって生成されます。

| ビット  | 7     | 6    | 5     | 4     | 3    | 2    | 1 | 0    |
|------|-------|------|-------|-------|------|------|---|------|
| TWCR | TWINT | TWEA | TWSTA | TWSTO | TWWC | TWEN | - | TWIE |
| 設定値  | 1     | X    | 0     | 1     | X    | 1    | 0 | X    |

**再送開始条件**はTWCRに次の値を書くことによって生成されます。

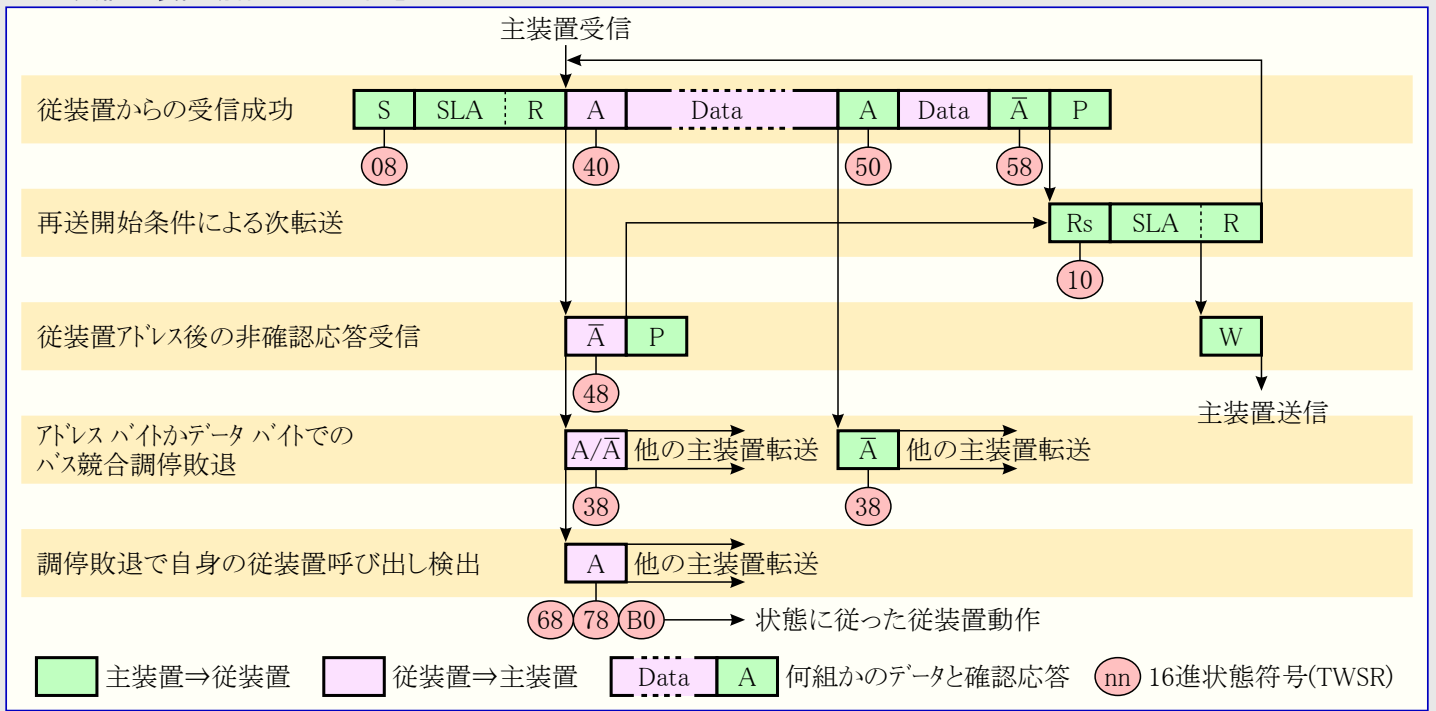
| ビット  | 7     | 6    | 5     | 4     | 3    | 2    | 1 | 0    |
|------|-------|------|-------|-------|------|------|---|------|
| TWCR | TWINT | TWEA | TWSTA | TWSTO | TWWC | TWEN | - | TWIE |
| 設定値  | 1     | X    | 1     | 0     | X    | 1    | 0 | X    |

**再送開始条件**(状態符号\$10)後、2線直列インターフェースは**停止条件**を送出せずに再び同じ従装置または新しい従装置にアクセスできます。**再送開始条件**は主装置がバスの制御を失わずに送信主装置、受信主装置間の切り替えを可能にします(訳注:原文では従装置も含まれていますが、基本動作に対して不適切なため削除しました)。

表89. 受信主装置動作の状態符号 (注: TWSRの前置分周選択ビットは0の前提)

| 状態符号 (TWSR) | 直前の動作とバスの状態             | ソフトウェアの対応 |        |       |       |      | TWCR設定によるハードウェア動作   |
|-------------|-------------------------|-----------|--------|-------|-------|------|---------------------|
|             |                         | TWDR操作    | TWCR設定 |       |       |      |                     |
|             |                         |           | TWSTA  | TWSTO | TWINT | TWEA |                     |
| \$08        | 開始条件送信                  | SLA+R設定   | 0      | 0     | 1     | X    | SLA+R送信、ACKかNACK受信  |
| \$10        | 再送開始条件送信                | SLA+R設定   | 0      | 0     | 1     | X    | SLA+R送信、ACKかNACK受信  |
|             |                         | SLA+W設定   | 0      | 0     | 1     | X    | SLA+W送信、送信主装置動作へ移行  |
| \$38        | SLA+Rで調停敗退<br>またはNACK受信 | なし        | 0      | 0     | 1     | X    | バス開放、未指定従装置動作へ移行    |
|             |                         |           | 1      | 0     | 1     | X    | バス開放時に開始条件送信        |
| \$40        | SLA+R送信<br>ACK受信        | なし        | 0      | 0     | 1     | 0    | データ受信、NACK応答        |
|             |                         |           | 0      | 0     | 1     | 1    | データ受信、ACK応答         |
| \$48        | SLA+R送信<br>NACK受信       | なし        | 1      | 0     | 1     | X    | 再送開始条件送信            |
|             |                         |           | 0      | 1     | 1     | X    | 停止条件送信、TWSTO=0      |
|             |                         |           | 1      | 1     | 1     | X    | 停止条件→開始条件送信、TWSTO=0 |
| \$50        | データバイト受信<br>ACK応答       | データ取得     | 0      | 0     | 1     | 0    | データ受信、NACK応答        |
|             |                         |           | 0      | 0     | 1     | 1    | データ受信、ACK応答         |
| \$58        | データバイト受信<br>NACK応答      | データ取得     | 1      | 0     | 1     | X    | 再送開始条件送信            |
|             |                         |           | 0      | 1     | 1     | X    | 停止条件送信、TWSTO=0      |
|             |                         |           | 1      | 1     | 1     | X    | 停止条件→開始条件送信、TWSTO=0 |

図99. 受信主装置動作の形式と状態

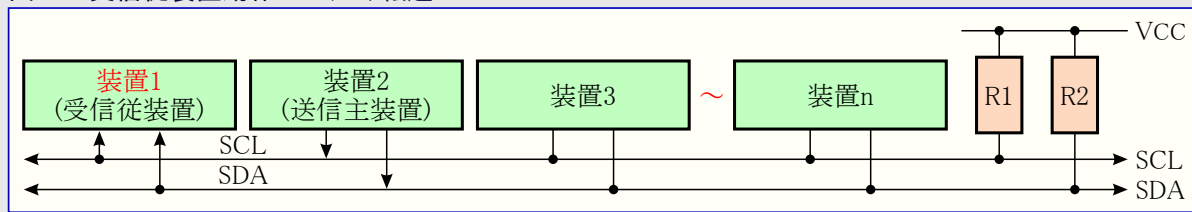




## 受信従装置動作

受信従装置動作では何バイトかのデータが送信主装置から受信されます(図100.参照)。本項で言及する全ての状態符号は前置分周選択ビットが0か、または0で遮蔽されることが前提です。

図100. 受信従装置動作でのデータ転送



受信従装置動作を始めるにはTWARとTWCRが次のように初期化されなければなりません。

| ビット  | 7            | 6    | 5    | 4    | 3    | 2    | 1    | 0     |
|------|--------------|------|------|------|------|------|------|-------|
| TWAR | TWA6         | TWA5 | TWA4 | TWA3 | TWA2 | TWA1 | TWA0 | TWGCE |
| 設定値  | 装置自身の従装置アドレス |      |      |      |      |      |      | 1/0   |

上位7ビットは主装置によってアドレス指定される時に2線直列インターフェースが応答するアドレスです。最下位(TWGCE)ビットが設定(1)されるなら、TWIは一斉呼び出し(\$00)に応答し、さもなければ一斉呼び出しアドレスを無視します。

| ビット  | 7     | 6    | 5     | 4     | 3    | 2    | 1 | 0    |
|------|-------|------|-------|-------|------|------|---|------|
| TWCR | TWINT | TWEA | TWSTA | TWSTO | TWWC | TWEN | - | TWIE |
| 設定値  | 0     | 1    | 0     | 0     | 0    | 1    | 0 | X    |

TWENは2線直列インターフェース(TWI)を許可するために1を書かれなければなりません。TWEAは装置自身の従装置アドレスまたは一斉呼び出しアドレスの確認応答(ACK)を許可するために1を書かれなければなりません。TWSTAとTWSTOは0を書かれなければなりません。

TWARとTWCRが初期化されてしまうと、TWIは自身の従装置アドレス(または許可なら、一斉呼び出しアドレス)とそれに続くデータ方向ビットによってアドレス指定されるまで待機します。方向ビットが0(W)ならばTWIは受信従装置で動作し、さもなければ(1(R)ならば)送信従装置へ移行されます。自身の従装置アドレスとWビットが受信されてしまった後にTWINTフラグが設定(1)され、TWSRから有効な状態符号が読めます。この状態符号は適切なソフトウェア動作を決めるのに使われます。各状態符号に対して行うべき適切な動作は表90.で詳述されます。受信従装置動作はTWIが主装置動作の間で調停に敗れた場合にも移行されるかもしれません。(状態符号\$68,\$78参照)

転送中にTWEAビットがリセット(0)されると、TWIは次に受信したデータバイト後のSDAに非確認応答(NACK)(SDA=High)を返します。これは従装置がこれ以上受信できないことを示すのに使えます。TWEAが0の間中、TWIは自身の従装置アドレスに応答しませんが、2線直列バスは未だ監視され、アドレス認証はTWEAの設定(1)によって何時でも再開できます。これはTWEAビットがTWIを2線直列バスから一時的に隔離するのに使えることを意味します。

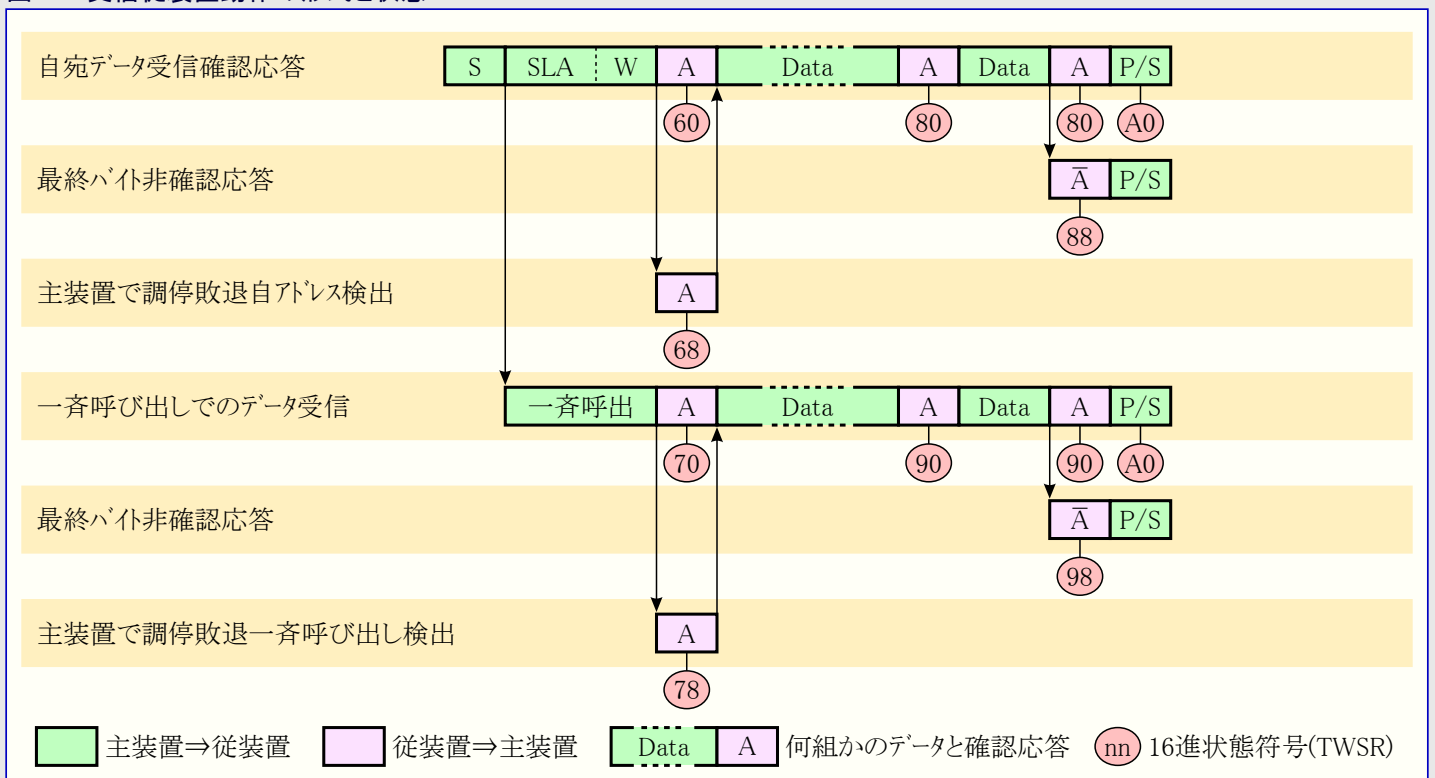
アイドル動作を除く休止形態ではTWIへのクロック系がOFFにされます。TWEAビットが設定(1)されていると、このインターフェースはクロック元として2線直列バスクロック(SCL)を使うことにより、自身の従装置アドレスと一斉呼び出しに未だ確認応答できます。その後デバイスが休止形態から起動し、TWIは起動中からTWINTフラグが1書き込みによって解除(0)されるまでSCLクロックをLowに保ちます。その後のデータ受信はAVRクロックが通常通り走行することで通常通りに行われます。AVRが長い起動時間に設定されていると、SCL信号線が長時間Lowに保持され、他のデータ送信を阻止するかもしれないことに気付いてください。

これらの(アイドル動作を除く)休止形態から起動すると、2線直列インターフェースデータレジスタ(TWDR)がバスで渡す最後のバイトを反映しないことに注意してください。

表90. 受信従装置動作の状態符号 (注: TWSRの前置分周選択ビットは0の前提)

| 状態符号 (TWSR) | 直前の動作とバスの状態                              | ソフトウェアの応答 |        |       |       |      | TWCR設定によるハードウェア動作               |
|-------------|--|-----------|--------|-------|-------|------|---------------------------------|
|             |  | TWDR操作    | TWCR設定 |       |       |      |                                 |
|             |  |           | TWSTA  | TWSTO | TWINT | TWEA |                                 |
| \$60        | 自宛SLA+W受信<br>ACK応答                       | なし        | X      | 0     | 1     | 0    | データ受信、NACK応答                    |
|             |  |           | X      | 0     | 1     | 1    | データ受信、ACK応答                     |
| \$68        | 主装置のSLA+R/Wで<br>調停敗退/自宛SLA+W<br>受信/ACK応答 | なし        | X      | 0     | 1     | 0    | データ受信、NACK応答                    |
|             |  |           | X      | 0     | 1     | 1    | データ受信、ACK応答                     |
| \$70        | 一斉呼び出し受信<br>ACK応答                        | なし        | X      | 0     | 1     | 0    | データ受信、NACK応答                    |
|             |  |           | X      | 0     | 1     | 1    | データ受信、ACK応答                     |
| \$78        | 主装置のSLA+R/Wで<br>調停敗退/一斉呼び<br>出し受信/ACK応答  | なし        | X      | 0     | 1     | 0    | データ受信、NACK応答                    |
|             |  |           | X      | 0     | 1     | 1    | データ受信、ACK応答                     |
| \$80        | 自宛データバイト受信<br>ACK応答                      | データ取得     | X      | 0     | 1     | 0    | データ受信、NACK応答                    |
|             |  |           | X      | 0     | 1     | 1    | データ受信、ACK応答                     |
| \$88        | 自宛データバイト受信<br>NACK応答                     | データ取得     | 0      | 0     | 1     | 0    | 未指定従装置動作へ移行、応答禁止                |
|             |  |           | 0      | 0     | 1     | 1    | 未指定従装置動作へ移行、応答対応                |
|             |  |           | 1      | 0     | 1     | 0    | 未指定従装置動作へ移行、応答禁止<br>バス開放で開始条件送信 |
|             |  |           | 1      | 0     | 1     | 1    | 未指定従装置動作へ移行、応答対応<br>バス開放で開始条件送信 |
| \$90        | 一斉呼び出しのデータ<br>バイト受信/ACK応答                | データ取得     | X      | 0     | 1     | 0    | データ受信、NACK応答                    |
|             |  |           | X      | 0     | 1     | 1    | データ受信、ACK応答                     |
| \$98        | 一斉呼び出しのデータ<br>バイト受信<br>NACK応答            | データ取得     | 0      | 0     | 1     | 0    | 未指定従装置動作へ移行、応答禁止                |
|             |  |           | 0      | 0     | 1     | 1    | 未指定従装置動作へ移行、応答対応                |
|             |  |           | 1      | 0     | 1     | 0    | 未指定従装置動作へ移行、応答禁止<br>バス開放で開始条件送信 |
|             |  |           | 1      | 0     | 1     | 1    | 未指定従装置動作へ移行、応答対応<br>バス開放で開始条件送信 |
| \$A0        | 自指定中の<br>停止条件または<br>再送開始条件検出             | なし        | 0      | 0     | 1     | 0    | 未指定従装置動作へ移行、応答禁止                |
|             |  |           | 0      | 0     | 1     | 1    | 未指定従装置動作へ移行、応答対応                |
|             |  |           | 1      | 0     | 1     | 0    | 未指定従装置動作へ移行、応答禁止<br>バス開放で開始条件送信 |
|             |  |           | 1      | 0     | 1     | 1    | 未指定従装置動作へ移行、応答対応<br>バス開放で開始条件送信 |

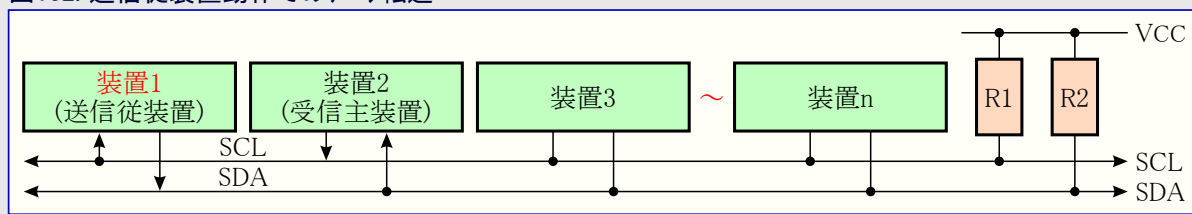
図101. 受信従装置動作の形式と状態



## 送信従装置動作

送信従装置動作では何バイトかのデータが送信主装置へ送信されます(図102参照)。本項で言及する全ての状態符号は前置分周選択ビットが0か、または0で遮蔽されることが前提です。

図102. 送信従装置動作でのデータ転送



送信従装置動作を始めるにはTWARとTWCRが次のように初期化されなければなりません。

| ビット  | 7            | 6    | 5    | 4    | 3    | 2    | 1    | 0     |
|------|--------------|------|------|------|------|------|------|-------|
| TWAR | TWA6         | TWA5 | TWA4 | TWA3 | TWA2 | TWA1 | TWA0 | TWGCE |
| 設定値  | 装置自身の従装置アドレス |      |      |      |      |      |      | 1/0   |

上位7ビットは主装置によってアドレス指定される時に2線直列インターフェースが応答するアドレスです。最下位(TWGCE)ビットが設定(1)されるなら、TWIは一斉呼び出し(\$00)に応答し、さもなければ一斉呼び出しアドレスを無視します。

| ビット  | 7     | 6    | 5     | 4     | 3    | 2    | 1 | 0    |
|------|-------|------|-------|-------|------|------|---|------|
| TWCR | TWINT | TWEA | TWSTA | TWSTO | TWWC | TWEN | - | TWIE |
| 設定値  | 0     | 1    | 0     | 0     | 0    | 1    | 0 | X    |

TWENは2線直列インターフェース(TWI)を許可するために1を書かれなければなりません。TWEAは装置自身の従装置アドレスまたは一斉呼び出しアドレスの確認応答(ACK)を許可するために1を書かれなければなりません。TWSTAとTWSTOは0を書かれなければなりません。

TWARとTWCRが初期化されてしまうと、TWIは自身の従装置アドレス(または許可ならば一斉呼び出しアドレス)とそれに続くデータ方向ビットによってアドレス指定されるまで待機します。方向ビットが1(R)ならばTWIは送信従装置で動作し、さもなくば0(W)ならば受信従装置へ移行されます。自身の従装置アドレスとRビットが受信されてしまった後、TWINTフラグが設定(1)され、TWSRから有効な状態符号が読めます。この状態符号は適切なソフトウェア動作を決めるのに使われます。各状態符号に対して行うべき適切な動作は表91で詳述されます。送信従装置動作はTWIが主装置動作の間で調停に敗れた場合にも移行されるかもしれません。(状態符号\$B0参照)

転送中にTWEAビットが0を書かれると、TWIは転送の最後のバイトを送信します。受信主装置が最終バイト後にACKまたはNACKのどちらを送信するかによって状態\$C0か\$C8へ移行します。TWIはアドレス指定されていない従装置動作に切り替えられ、主装置が転送を続ける場合、その主装置を無視します。従って受信主装置は直列データとして全て1を受信します。従装置が最後のバイトを送信(TWEAが0で主装置からのNACKを予測したとしても、主装置が(ACK送信によって)追加データバイトを要求すると状態\$C8へ移行します。

TWEAが0の間中、TWIは自身の従装置アドレスに応答しませんが、2線直列バスは未だ監視され、アドレス認証はTWEAの設定(1)によって何時でも再開できます。これはTWEAビットがTWIを2線直列バスから一時的に隔離するのに使えることを意味します。

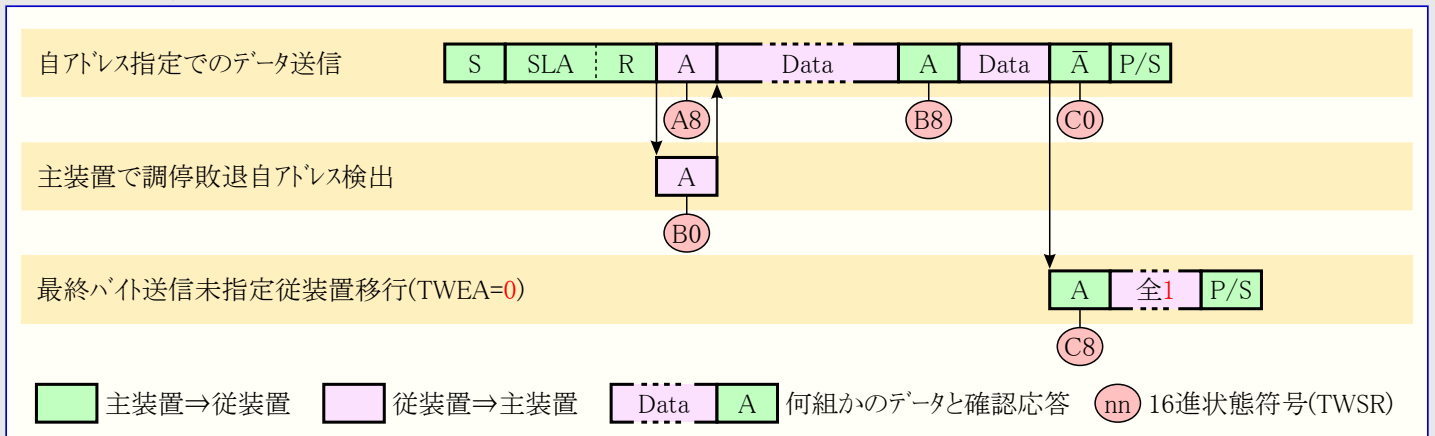
アイドル動作を除く休止形態ではTWIへのクロック系がOFFにされます。TWEAビットが設定(1)されていると、このインターフェースはクロック元として2線直列バスクロック(SCL)を使うことにより、自身の従装置アドレスと一斉呼び出しに未だ確認応答できます。その後デバイスが休止形態から起動し、TWIは起動中からTWINTフラグが(1書き込みによって)解除(0)されるまでSCLクロックをLowに保ちます。その後のデータ受信はAVRクロックが通常通り走行することで通常通りに行われます。AVRが長い起動時間に設定されていると、SCL信号線が長時間Lowに保持され、他のデータ送信を阻止するかもしれないことに気付いてください。

これらの(アイドル動作を除く)休止形態から起動すると、2線直列インターフェースデータレジスタ(TWDR)がバスで渡す最後のバイトを反映しないことに注意してください。

表91. 送信従装置動作の状態符号 (注: TWSRの前置分周選択ビットは0の前提)

| 状態符号<br>(TWSR) | 直前の動作と<br>バスの状態                          | ソフトウェアの応答 |        |       |       |      | TWCR設定によるハードウェア動作               |
|----------------|--|-----------|--------|-------|-------|------|---------------------------------|
|                |  | TWDR操作    | TWCR設定 |       |       |      |                                 |
|                |  |           | TWSTA  | TWSTO | TWINT | TWEA |                                 |
| \$A8           | 自宛SLA+R受信<br>ACK応答                       | データ設定     | X      | 0     | 1     | 0    | 最終データバイト送信、NACK受信予定             |
|                |  |           | X      | 0     | 1     | 1    | データバイト送信、ACK受信予定                |
| \$B0           | 主装置のSLA+R/Wで<br>調停敗退/自宛SLA+<br>R受信/ACK応答 | データ設定     | X      | 0     | 1     | 0    | 最終データバイト送信、NACK受信予定             |
|                |  |           | X      | 0     | 1     | 1    | データバイト送信、ACK受信予定                |
| \$B8           | データバイト送信<br>ACK受信                        | データ設定     | X      | 0     | 1     | 0    | 最終データバイト送信、NACK受信予定             |
|                |  |           | X      | 0     | 1     | 1    | データバイト送信、ACK受信予定                |
| \$C0           | データバイト送信<br>NACK受信                       | なし        | 0      | 0     | 1     | 0    | 未指定従装置動作へ移行、応答禁止                |
|                |  |           | 0      | 0     | 1     | 1    | 未指定従装置動作へ移行、応答対応                |
|                |  |           | 1      | 0     | 1     | 0    | 未指定従装置動作へ移行、応答禁止<br>バス開放で開始条件送信 |
|                |  |           | 1      | 0     | 1     | 1    | 未指定従装置動作へ移行、応答対応<br>バス開放で開始条件送信 |
| \$C8           | 最終データバイト送信<br>(TWEA=0)<br>ACK受信          | なし        | 0      | 0     | 1     | 0    | 未指定従装置動作へ移行、応答禁止                |
|                |  |           | 0      | 0     | 1     | 1    | 未指定従装置動作へ移行、応答対応                |
|                |  |           | 1      | 0     | 1     | 0    | 未指定従装置動作へ移行、応答禁止<br>バス開放で開始条件送信 |
|                |  |           | 1      | 0     | 1     | 1    | 未指定従装置動作へ移行、応答対応<br>バス開放で開始条件送信 |

図103. 送信従装置動作の形式と状態



### その他の状態

定義したTWI状態に従わない2つの状態符号があります。表92をご覧ください。

状態\$F8はTWI割り込み要求フラグ(TWINT)が設定(1)されないの適切な情報が利用できないことを示します。これは他の状態間でTWIが直列転送に関係しない時に起きます。

状態\$00は2線直列バス転送中にバス異常が起きたことを示します。バス異常はフレーム形式の不正な位置で開始(START)条件または停止(STOP)条件が起きる時に発生します。このような不正位置の例はアドレスバイト、データバイト、確認応答(ACK)ビットの直列転送中です。バス異常が起きるとTWINTが設定(1)されます。バス異常から回復するには、停止(STOP)条件生成許可(TWSTO)ビットが設定(1)され、TWINTが論理1書き込みによって解除(0)されなければなりません。これはTWIをアドレス指定されていない従装置動作にさせ、TWSTOビットを解除(0)させます(TWCRの他のビットは影響されません)。SDAとSCL信号進は開放され、停止条件は送出されません。

表92. その他の状態符号 (注: TWSRの前置分周選択ビットは0の前提)

| 状態符号<br>(TWSR) | 直前の動作と<br>バスの状態        | ソフトウェアの応答 |        |       |       |      | TWCR設定によるハードウェア動作          |
|----------------|------------------------|-----------|--------|-------|-------|------|----------------------------|
|                |                        | TWDR操作    | TWCR設定 |       |       |      |                            |
|                |                        |           | TWSTA  | TWSTO | TWINT | TWEA |                            |
| \$F8           | 適切な状態情報なし<br>TWINT=0   | なし        | -      | -     | -     | -    | 待機または現在の転送続行               |
| \$00           | 不正な開始条件/停止<br>条件でのバス異常 | なし        | 0      | 1     | 1     | X    | 停止条件を送出せずにバスを開放<br>TWSTO=0 |



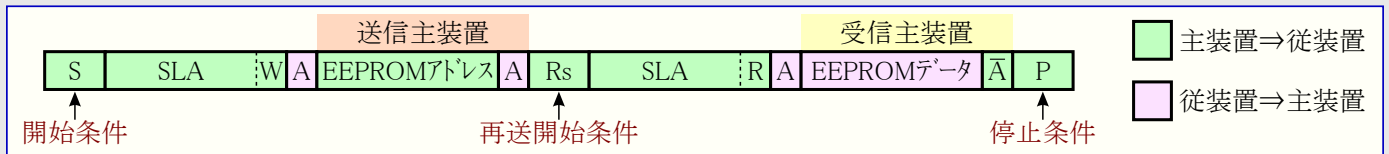
## 各種TWI動作種別の組み合わせ

いくつかの場合で望んだ動作を満たすために各々のTWI動作種別は組み合わせられなければなりません。例えば直列EEPROMからのデータ読み出しを考えてください。一般的にこのような転送は次の段階を含みます。

1. 転送が開始されなければなりません。
2. EEPROMは読み出すべき場所を指示されなければなりません。
3. 読み出しが実行されなければなりません。
4. 転送が終了されなければなりません。

データが主装置から従装置へとその逆の両方向へ転送されることに注意してください。主装置はどの場所を読みたいかを従装置に指示しなければならず、送信主装置動作の使用を必要とします。その後データに従装置から読まねばならず、受信主装置動作の使用を意味します。従って転送方向が切り替えられなければなりません。主装置はこれら全ての段階中にバスの制御を保持しなければならず、この手順は排他的(非分断)操作として行われるべきです。複数主装置システムでこの原則に違反すると、他の主装置が②と③段階間でEEPROM内のデータポインタを変更するかもしれず、(元の)主装置は不正なデータ位置を読むでしょう。このような転送方向の切り替えはアドレスバイトの送信とデータの受信間で再送開始条件を送出することによって成し遂げられます。再送開始条件後も主装置はバスの占有権を保持します。次の図はこの転送の流れを示します。

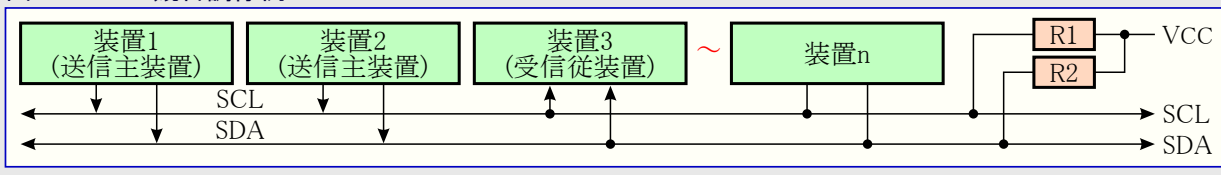
図104. 直列EEPROMアクセスでの各種TWI動作種別の組み合わせ



## 複数主装置システムでのバス競合と調停

複数の主装置が同じバスに接続されると、それらの1つまたはそれ以上によって同時に送信が開始されるかもしれません。TWIは主装置の1つが転送を続けることを許され、手順内でデータが失われないような方法でこのような状態が扱われることを標準で保証します。2つの主装置が受信従装置へデータを送信することを試みる場合の調停状況の例は以下で図示されます。

図105. バスの競合調停例

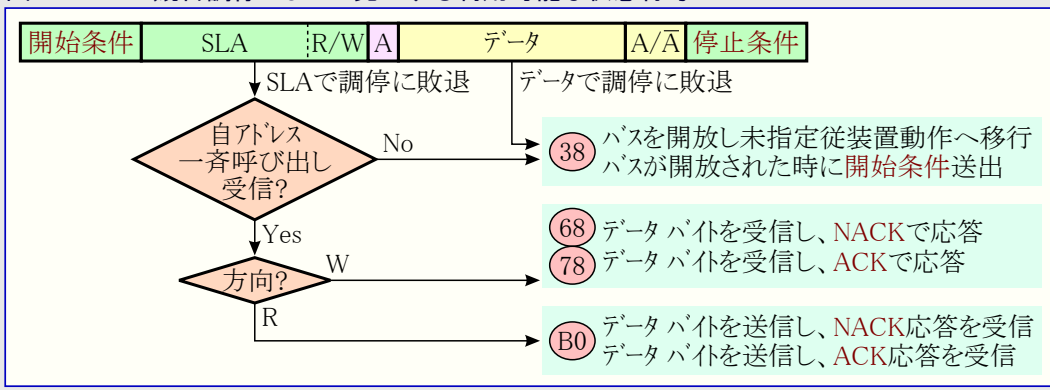


以下で示されるように様々な異なる状況が調停中に起こるかもしれません。

- 複数の主装置が同じ従装置に全く同じ通信を実行する場合。この場合、主/従装置のどれもがバスの衝突について知りません。
- 複数の主装置が異なるデータまたは方向ビット(R/W)で同じ従装置をアクセスする場合。この場合、R/Wビットまたはデータビットのどちらかで調停が起きます。他の主装置がSDAに0を出力する間に1を出力しようとする主装置が調停に敗れます。敗れた主装置はアプリケーションソフトウェアの処置によって未指定従装置動作に切り替えるか、またはバスが開放になるまで待って新規開始条件を送出します。
- 複数の主装置が異なる従装置をアクセスする場合。この場合、SLAビット内で調停が起きます。他の主装置がSDAに0を出力する間に1を出力しようとする主装置が調停に敗れます。SLA内で調停に敗れた主装置は勝った主装置によってアドレス指定されるかを検査するために従装置動作へ切り替えます。アドレス指定されると、R/Wビットの値によって受信従装置(SR)動作または送信従装置(ST)動作へ切り替えます。アドレス指定されないなら、アプリケーションソフトウェアの処置によって未指定従装置動作に切り替えるか、またはバスが開放になるまで待って新規開始条件を送出します。

これは図106.で要約されます。利用可能な状態符号は楕円(訳注:原文は円)で与えられます。

図106. バスの競合調停によって発生する利用可能な状態符号

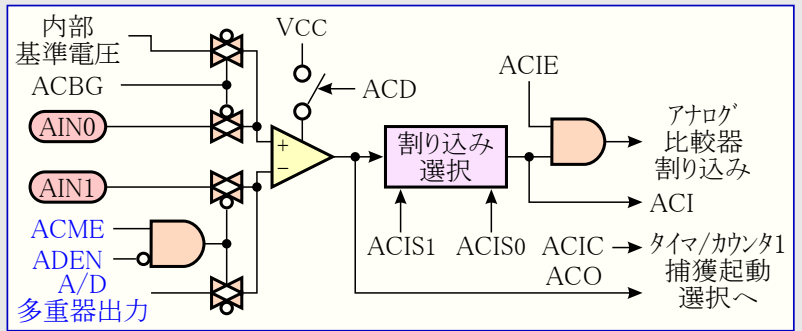




## アナログ比較器

アナログ比較器は非反転入力AIN0ピンと反転入力AIN1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**ACSRのアナログ比較器出力(ACO)ビット**が設定(1)されます。この比較器出力はタイマ/カウンタ1の捕獲機能を起動するように設定できます。加えて、この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、またはその両方で割り込み起動を選べます。この比較器とその周辺論理回路の構成図は**図107**で示されます。

図107. アナログ比較器部構成図



注: A/D多重器出力については153頁の**表94**をご覧ください。  
アナログ比較器ピン配置については2頁の「**ピン配置**」と51頁の**表39**を参照してください。

### アナログ比較器 制御/状態レジスタ (Analog Comparator Control and Status Register) ACSR

| ビット         | 7   | 6    | 5   | 4   | 3    | 2    | 1     | 0     |      |
|-------------|-----|------|-----|-----|------|------|-------|-------|------|
| \$08 (\$28) | ACD | ACBG | ACO | ACI | ACIE | ACIC | ACIS1 | ACIS0 | ACSR |
| Read/Write  | R/W | R/W  | R   | R/W | R/W  | R/W  | R/W   | R/W   |      |
| 初期値         | 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ACSR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り得ます。

#### • ビット6 – ACBG : 基準電圧選択 (Analog Comparator Bandgap Select)

このビットが設定(1)されると、内部基準電圧(公称1.23V)がアナログ比較器への非反転入力に取って代わります。このビットが解除(0)されると、AIN0がアナログ比較器の非反転入力に印加されます。**34頁の「内部基準電圧」**をご覧ください。

#### • 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ACSR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。ACSR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)されていると、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1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1捕獲割り込みの**雑音消去機能**と**端(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1捕獲割り込みを起動するには、**タイマ/カウンタ割り込み許可レジスタ(TIMSK)の捕獲割り込み許可(TICIE1)ビット**が設定(1)されなければなりません。

#### • ビット1,0 – ACIS1, ACIS0 : 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。各種設定は**表93**で示されます。

ACIS1, ACIS0ビットを変更する時にACSR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93. 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(トグル) |
| 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     |

## 特殊I/O機能レジスタ (Special Function I/O Register) SFIOR

| ビット         | 7   | 6 | 5 | 4 | 3    | 2   | 1    | 0      |       |
|-------------|-----|---|---|---|------|-----|------|--------|-------|
| \$20 (\$40) | TSM | - | - | - | ACME | PUD | PSR0 | PSR321 | SFIOR |
| Read/Write  | R/W | R | R | R | R/W  | R/W | R/W  | R/W    |       |
| 初期値         | 0   | 0 | 0 | 0 | 0    | 0   | 0    | 0      |       |

### •ビット3 – ACME : アナログ比較器多重器許可 (Analog Comparator Multiplexer Enable)

このビットが論理1を書かれ、A/D変換部がOFF(ADCSRAのADENビットが0)にされると、A/D変換の多重器がアナログ比較器への反転入力を選びます。このビットが論理0を書かれると、AIN1がアナログ比較器の反転入力に印加されます。このビットの詳細な記述については次の「アナログ比較器入力選択」をご覧ください。

### アナログ比較器入力選択

アナログ比較器への反転入力を取り替えるのにADC7~0のどれかを選ぶことができます。A/D変換の多重器がこの入力選択に使われ、従ってこの機能を利用するためにA/D変換部がOFF(動作禁止)にされなければなりません。SFIORのアナログ比較器多重器許可(ACME)ビットが設定(1)され、A/D変換部がOFF(ADCSRAのADENビットが0)にされていれば、表94.で示されるようにADMUXのチャンネル選択(MUX2~0)ビットがアナログ比較器への反転入力を取り替えるための入力ピンを選びます。ACMEが解除(0)、またはADENが設定(1)されると、AIN1がアナログ比較器への反転入力に印加されます。

表94. アナログ比較器反転入力選択

| ACME | ADEN | MUX2~0 | アナログ比較器反転入力 | 備考 |
|------|------|--------|-------------|----|
| 0    | x    | x x x  | AIN1        |    |
|      | 1    | x x x  |             |    |
| 1    | 0    | 0 0 0  | ADC0        |    |
|      |      | 0 0 1  | ADC1        |    |
|      |      | 0 1 0  | ADC2        |    |
|      |      | 0 1 1  | ADC3        |    |
|      |      | 1 0 0  | ADC4        |    |
|      |      | 1 0 1  | ADC5        |    |
|      |      | 1 1 0  | ADC6        |    |
|      |      | 1 1 1  | ADC7        |    |

## A/D変換器

A/D変換部の主な機能を次に示します。

- 10ビット分解能
- 積分非直線性誤差0.75 LSB
- 絶対精度±1.5 LSB
- 変換時間13~260μs
- 76.9kSPS(採取/s)まで(最大分解能で15kSPSまで)
- 8チャンネルのシングルエンド入力多重器内蔵
- 7チャンネルの差動入力チャンネル
- ×10、×200の任意利得付き2つの差動入力チャンネル
- A/D変換結果読み出しに対する任意の左揃え
- 0~VCC A/D変換入力電圧範囲
- 選択可能な2.56V A/D変換基準電圧
- 連続と単独の変換動作
- A/D変換完了割り込み
- 休止形態雑音低減機能

Atmel® AVR® ATmega128は10ビット逐次比較A/D変換器が特徴です。このA/D変換器はポートFのピンから構成された8つのシングルエンド電圧入力を許す8チャンネルアナログ多重器に接続されます。このシングルエンド電圧入力は0V(GND)が基準です。

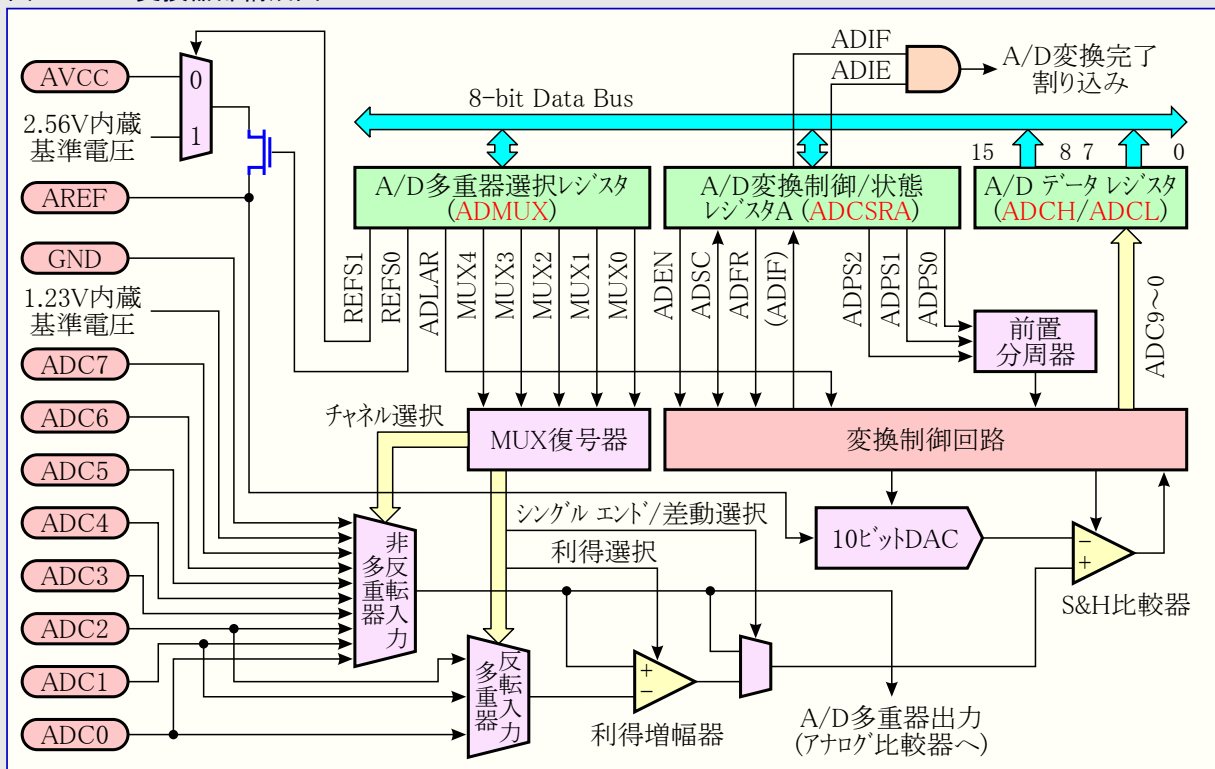
デバイスは16通りの差動電圧入力も支援します。差動入力の2つ(ADC1,ADC0とADC3,ADC2)は、A/D変換前の差動入力電圧で0dB(×1)、20dB(×10)、46dB(×200)の増幅段を提供する設定変更可能な利得段が装備されます。7つの差動アナログ入力は共通反転端子(ADC1)を共用し、一方他のADC入力は非反転入力端子として選べます。利得×1または×10が使われる場合は8ビット分解能が期待でき、利得×200が使われる場合は7ビット分解能が期待できます。

この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。A/D変換部の構成図は図108.で示されます。

A/D変換部には分離されたアナログ電源供給ピン(AVCC)があります。AVCCはVCCから±0.3Vよりも多く違ってはなりません。このピンの接続方法は158頁の「雑音低減技術」項をご覧ください。

公称2.56Vの内蔵基準電圧またはAVCCがチップ上で提供されます。この基準電圧は雑音特性向上のため、コンデンサによってAREFピンで外部的にデカップ(雑音分離)できます。

図108. A/D変換器部構成図



## 操作

A/D変換部は逐次比較を通してアナログ入力電圧を10ビットのデジタル値に変換します。最小値はGNDを表し、最大値はAREFピンの電圧-1 LSBを表します。A/D多重器選択レジスタ(ADMUX)の基準電圧選択(REFS1,REFS0)ビットへの書き込みにより、任意でAVCCまたは内部2.56V基準電圧がAREFピンに接続できます。この内部基準電圧は雑音耐性を改善するためにAREFピンで外部コンデンサによってデカップ(雑音結合減少)のようにできます。

アナログ入力チャネルと差動利得はADMUXのチャネル選択(MUX4~0)ビットへの書き込みによって選ばれます。GNDと固定基準電圧(1.23V内蔵基準電圧(VBG))だけでなく、どのADC入力ピン(ADC7~0)もがA/D変換器のシングルエンド入力として選べます。ADC入力ピンの選択は差動利得増幅器への反転と非反転入力として選べます。

差動チャネルが選ばれると、差動利得段は選んだ入力チャネル間の差電圧を選んだ増幅率で増幅します。それからこの増幅した値はA/D変換器の入力になります。シングルエンドチャネルが使われると、利得増幅器全体が迂回(無視)されます。

A/D変換部はA/D変換制御/状態レジスタ(ADCSRA)のA/D許可(ADEN)ビットの設定(1)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)されていると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。

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、この結果は右揃え(16ビットのビット0側10ビット)で示されますが、ADMUXで左揃え選択(ADLAR)ビットを設定(1)することにより、任意で左揃え(16ビットのビット15側10ビット)で示せます。

この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を読むことで足りる。さもな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が読まれてしまい、ADCHが読まれる前に変換が完了すると、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。

## 変換の開始

単独変換はADCSRAで変換開始(ADSC)ビットに論理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ばれると、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。

連続変換動作でのA/D変換部は継続的に採取/変換とA/Dデータレジスタを更新します。連続変換動作はADCSRAでA/D動作選択(ADFR)ビットに1を書くことによって選ばれます。最初の変換はADCSRAでADSCビットに論理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)されるかどうかにかかわらず、連続的な変換を実行します。

変換はA/D変換雑音低減機能の使用によっても開始され得ます。この機能はアイドル休止動作とA/D変換雑音低減休止動作中に変換を許可します。詳細については158頁の「雑音低減機能」をご覧ください。(訳注:共通性から2行追加)



## 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。10ビットよりも低い分解能が必要とされるなら、A/D変換器への入力クロック周波数はより高い採取速度を得るために200kHzよりも高くできます。

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。この前置分周器はA/D変換制御/状態レジスタ(ADCSRA)のA/Dクロック選択(ADPS2~0)ビットによって設定されます。前置分周器はADCSRAでA/D許可(ADEN)ビットの設定(1)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ビットが設定(1)される限り走行を維持し、ADENが0の時は継続的にリセットします。

ADCSRAのA/D変換開始(ADSC)ビットの設定(1)によってシングルエンド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。差動入力変換タイミングの詳細については157頁の「差動増幅チャネル」を参照してください。

通常の変換は13変換クロック周期で行われます。A/D変換部がONされる(ADCSRAのADEN=1)後の最初の変換はアナログ回路を初期化するために25変換クロック周期で行われます。

実際の採取&保持(保持開始点)は通常変換の開始後1.5変換クロック周期、初回変換の開始後13.5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(ADCH,ADCL)に書かれ、ADCSRAのA/D変換完了割り込み要求フラグ(ADIF)が設定(1)されます。単独変換動作(ADFR=0)では同時にADCSRAのA/D変換開始(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。

連続変換動作(ADFR=1)では変換完了後直ちに新規変換が開始され、一方ADSCは1に留まります。変換時間の概要については表95をご覧ください。

図109. A/D変換前置分周器部構成

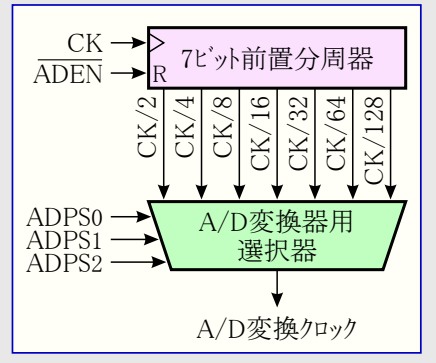


図110. 初回変換タイミング (単独変換動作)

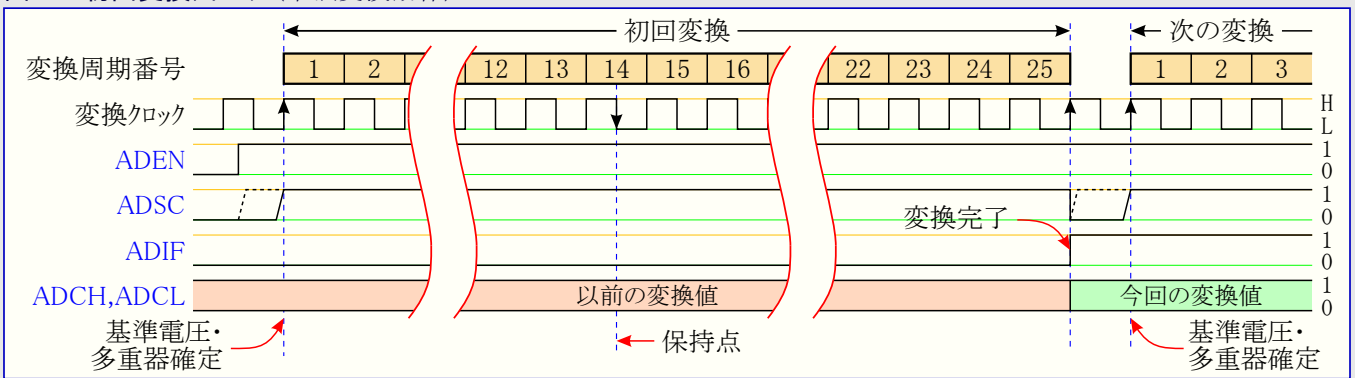


図111. 通常変換タイミング (単独変換動作)

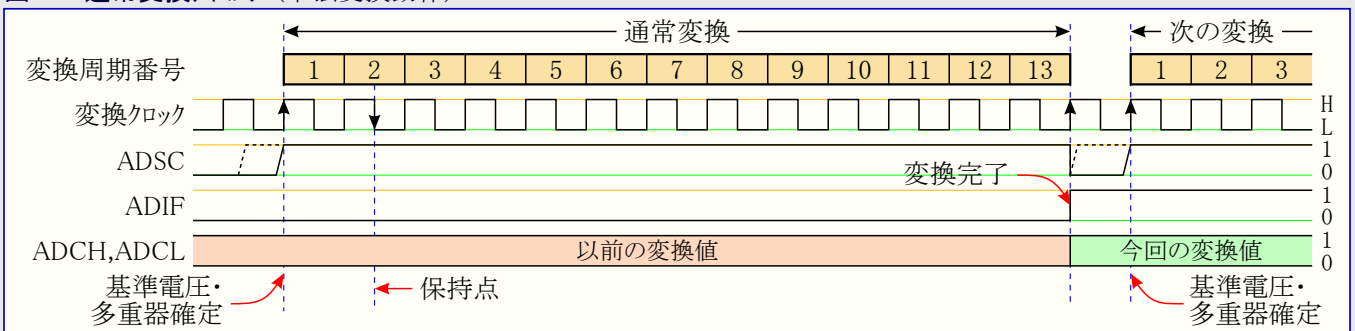


図112. 連続変換動作タイミング

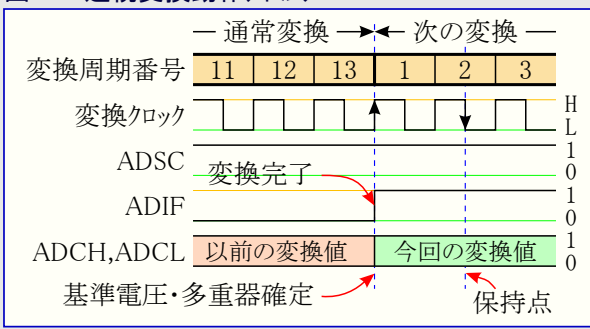


表95. A/D変換時間

| 変換種別          | 保持点          | 変換時間       |
|---------------|--------------|------------|
| 初回変換          | 13.5         | 25         |
| シングルエンド入力通常変換 | 1.5          | 13         |
| 差動入力通常変換      | 1.5/2.5 (注1) | 13/14 (注1) |

注: 変換時間を除く各値は変換開始からの変換クロック数です。

注1: CK<sub>ADC2</sub>の状態に依存(訳注:共通性から追加)。



## 差動増幅チャネル

差動増幅チャネルを使うとき、変換の±方向を考慮に入れる必要があります。

差動変換は変換クロックの半分に等しい内部クロックCKADC2に同期化されます。この同期化は採取&保持がCKADC2の特定位相で起きるといった方法でA/D変換器インターフェースによって自動的に行われます。CKADC2がLowの時に使用者によって変換が開始されると(即ち全ての単独変換と連続変換の最初)、シングルエンド入力変換(次の前置分周したクロック周期から13変換クロック周期)と同じ変換時間になります。CKADC2がHighの時に使用者によって変換が開始されると、同期化機構のために14変換クロック周期になります。連続変換動作では直前の変換完了後、直ちに次の変換が開始され、この時にCKADC2がHighのため、自動的に開始される全ての連続変換(即ち、初回を除く全て)は14変換クロック周期になります。

利得段は全利得設定に於いて帯域幅4kHzで最適化されています。より高い周波数は非直線増幅に陥るかもしれません。利得段帯域幅よりも高い周波数成分を含む入力信号の場合、外部低域通過濾波器(ローパスフィルタ)が使われるべきです。A/D変換クロック周波数が利得段帯域幅制限と無関係なことに注意してください。例えばA/D変換クロック周期が6 $\mu$ sでは、このチャネルの帯域幅とは無関係に12k採取/sでの採取をチャネルに許せます。

## チャネル変更と基準電圧選択

A/D多重器選択レジスタ(ADMUX)のチャネル選択(MUX4~0)ビットと基準電圧選択(REFS1,0)ビットはCPUが乱順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。これはチャネルと基準電圧の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。チャネルと基準電圧の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ネルと基準電圧の選択は固定されます。継続的な更新は変換完了(ADCSRAのADIF=1)前の最後の変換クロック周期で再開します。ADCSRAの変換開始(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ADSC書き込み後、1変換クロック周期(経過)まで新しいチャネルまたは基準電圧選択値をADMUXに書かないことを推奨されます。

差動入力チャネルを変更するとき、特別な注意が求められるべきです。一旦差動入力チャネルが選ばれてしまうと、利得段は新しい値に安定するのに125 $\mu$ s程度かかるかもしれません。従って変換は新規差動入力チャネル選択後の最初の125 $\mu$ s内に開始されるべきではありません。または、この期間内に得た変換結果は破棄されるべきです。

(ADMUXのREFS1,0ビットの変更による)A/D変換器基準電圧変更後の最初の差動入力変換に対して、同じ設定(安定)時間が厳守されるべきです。

JTAGインターフェースが許可されると、PORTF7~4のADC入力チャネル機能が無効にされます。53頁の表42を参照してください。

## A/D入力チャネル

チャネル選択を変更する時に使用者は正しいチャネルが選ば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ネルを選んでください。チャネル選択は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ネルを選んでください。チャネル選択は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、その後チャ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ネル選択を反映します。それに続く変換は新しいチャネル選択を反映します。

差動増幅チャネルへ切り替える時に自動変位(オフセット)消去回路用の設定時間が必要なので、最初の変換結果は貧弱な正確さとなってしまうかもしれません。使用者は最初の変換結果をなるべくなら無視すべきです。

## A/D変換基準電圧

このA/D変換用の基準電圧(VREF)はA/D変換に対する変換範囲を示します。VREFを越えるシングルエンド入力チャネルは\$3FFで打ち切る符号に帰着します。VREFはAVCC、内部2.56V基準電圧、外部AREFピンのどれかとして選べます。

AVCCは受動型スイッチを通してA/D変換部に接続されます。内部2.56V基準電圧は内蔵基準(バンドギャップ)電圧(VBG)から内部増幅器を通して生成されます。どちらの場合でも外部AREFピンは直接的にA/D変換部へ接続され、AREFピンとGND間にコンデンサを接続することにより、基準電圧は雑音耐性をより高められます。VREF(電圧)は高入力インピーダンス電圧計とAREFピンで測定することもできます。VREFは高インピーダンス出力で、容量性負荷のみがシステム内で接続されるべきであることに注意してください。

使用者がAREFピンに接続された固定電圧源にするなら、この外部電圧がその他の内部基準電圧と短絡してしまうため、使用者はこの応用内で他の基準電圧選択を使ってはなりません。外部電圧がAREFピンに印加されないなら、使用者は基準電圧選択としてAVCCと内部2.56V基準電圧間の切り替えができます。基準電圧源切り替え後の最初のA/D変換結果は不正確かもしれず、使用者はこの結果を破棄することが推奨されます。

差動入力チャネルが使われる場合、選んだ基準電圧は210頁の表135で示されるよりもAVCCに近くすべきではありません。

## 雑音低減機能

このA/D変換部はCPUコアと他の周辺I/Oが誘導した雑音を削減するために**休止形態**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**で使えます。この機能を使うには次の手順が使われるべきです。

1. A/D変換部が許可( $ADEN=1$ )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( $ADFR=0$ )され、且つA/D変換完了割り込みが許可( $ADIE=1$ )されていないとなりません。
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、A/D変換部は変換を始めます。
3. A/D変換完了前に他の割り込みが起こらなければ、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活動動作に留まります。

アイドル動作とA/D変換雑音低減動作を除く他の休止形態へ移行する時にA/D変換部が自動的に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へ移行する前にADENへ0を書くことが推奨されます。このような休止形態でA/D変換が許可され、使用者が差動変換の実行を欲する場合、使用者は有効な結果を得るための延長した(初回)変換を指示するために、休止形態から起動後にA/D変換部をOFF→ON( $ADEN=0 \rightarrow 1$ )に切り替えることが推奨されます。

## アナログ入力回路

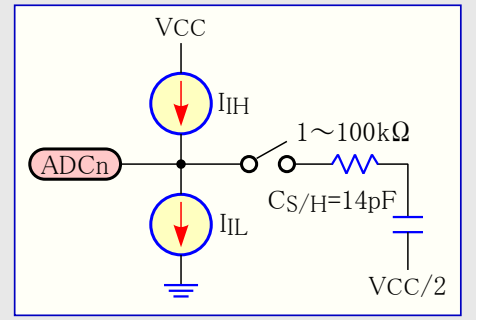
シングルエンド入力チャネルのアナログ回路は図113.で図示されます。ADCnに印加したアナログ(信号)源はそのチャネルがADC入力として選ばれているかどうかにかかわらず、ピン容量とそのピンの漏れ電流に左右されます。そのチャネルが選ばれると、(アナログ)信号源は直列抵抗(入力経路の合成抵抗)を通してS/Hコンデンサを駆動しなければなりません。

A/D変換部は概ね10kΩ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。必要とされるS/Hコンデンサへの充放電を最小とするため、使用者は緩やかに変化する低インピーダンス(アナログ)信号源だけを使うことが推奨されます。

差動増幅チャネルが使われる場合、この入力回路は多少違って見えるので、数100kΩまたはそれ以下の供給元インピーダンスが推奨されます。

特定できない信号の渦からの歪を避けるために、どのチャ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(ローパスフィルタ)で高い周波数成分を取り除くことが推奨されます。

図113. アナログ入力回路

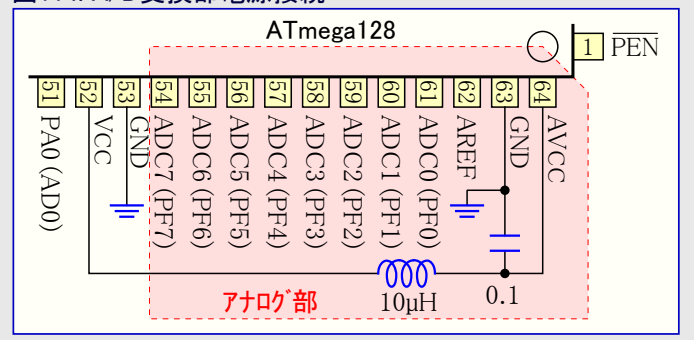


## アナログ雑音低減技術

デバイス内外のデジタル回路がアナログ測定に影響を及ぼすかもしれないEMIを発生します。精密な変換精度が必要な場合、次の技法を適用することによって雑音レベルを低減できます。

1. アナログ信号経路を可能な限り最短にしてください。アナログ信号線がGND面上を走ることに注意し、高速切り替えデジタル信号線から充分離すことを守ってください。
2. デバイスのAVCCピンは図114.で示されるようにLC濾波器を経由してデジタル供給電圧(VCC)に接続されるべきです。
3. CPUからの誘導雑音を低減するためにA/D変換の**雑音低減機能**を使ってください。
4. どれかのADCポートピンがデジタル出力として使われる場合、これらが変換進行中に切り替わらないことが重要です。

図114. A/D変換部電源接続



## 変位(オフセット)補償の仕組み

利得段には差動測定の変位(オフセット)をできるだけ無効にする組み込み変位補正回路があります。アナログ経路内の残留変位は両差動入力に対して同じチャネルを選ぶことによって直接的に計測できます。その後この残留変位は計測結果からソフトウェアで減算できます。この変位補正に基いたソフトウェア手法の使用はどのチャネルの変位も1 LSB以下に減少できます。

## A/D変換の精度定義

シングルエンド入力電圧のnビットA/D変換はGNDとVREF間を2<sup>n</sup>で直線的に変換します。最低値符号は0として読み、最高値符号は2<sup>n</sup>-1として読みます。以下の各種パラメータは理想状態からの偏差を表します。

### • 変位(オフセット)誤差 - 図115.

最初の遷移点(\$000から\$001)で理想遷移点(差0.5 LSB)と比べた偏差です。理想値は0 LSBです。

### • 利得誤差 - 図116.

変位誤差補正後の最後の遷移点(\$3FEから\$3FF)で理想遷移点(最大差1.5 LSB以下)と比べた偏差です。理想値は0 LSBです。

### • 積分非直線性誤差 (INL) - 図117.

変位誤差と利得誤差補正後の全ての遷移点で理想遷移点と比べた最大偏差です。理想値は0 LSBです。

### • 微分非直線性誤差 (DNL) - 図118.

実際の符号の幅(隣接する2つの遷移点間)で理想符号幅(1 LSB)と比べた最大偏差です。理想値は0 LSBです。

### • 量子化誤差

有限数の符号で入力電圧を量子化するため、1 LSB幅となる入力電圧範囲は同じ値の符号になります。この値は常に±0.5LSBです。

### • 絶対精度

補正しない全ての遷移点で理想遷移点と比べた最大偏差です。これは、変位誤差、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115. 変位(オフセット)誤差

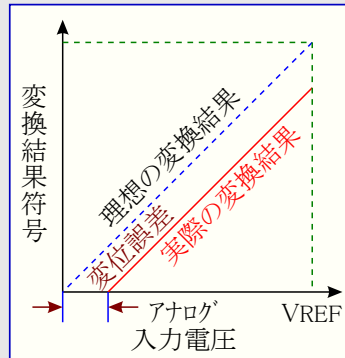


図116. 利得誤差

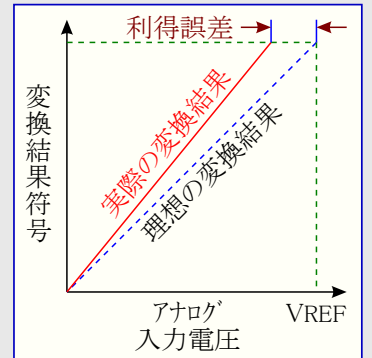


図117. 積分非直線性誤差

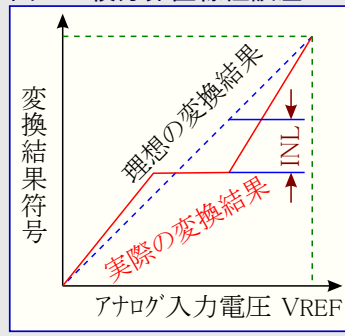
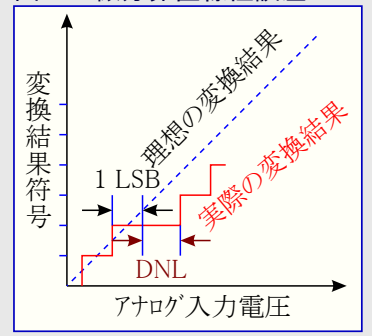


図118. 微分非直線性誤差



## A/D変換の結果

変換完了(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。

シングルエンド入力変換の結果は右式で示されます。VINは選んだ入力ピンの電圧、VREFは選んだ基準電圧です(160頁の表97と表98をご覧ください)。\$000はGNDを表し、\$3FFは選んだ基準電圧-1 LSBを表します。

$$ADC = \frac{VIN \times 1024}{VREF}$$

差動チャンネルが使われると、その結果は右式で示されます。VPOSは非反転入力ピンの電圧、VNEGは反転入力ピンの電圧、GAINは選んだ倍率、VREFは選んだ基準電圧です。この結果は-512(\$200)~+511 (\$1FF)の2の補数形式で示されます。使用者が結果の符号を素早く検査したいなら、結果のMSB(ADCHのADC9)を読むことで充分なことに注目してください。このビットが1ならばその結果は-、0ならばその結果は+です。図119.は差動入力範囲の符号化を示します。

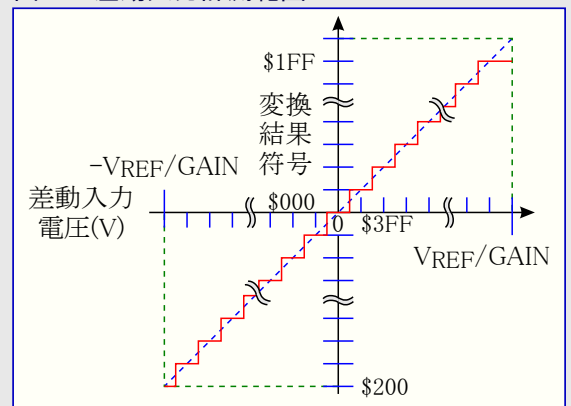
$$ADC = \frac{(VPOS - VNEG) \times GAIN \times 512}{VREF}$$

表96.は差動入力チャンネルの組(ADCn-ADCm)が基準電圧(VREF)で選ばれた場合の出力符号の結果を示します。

表96. 入力電圧と出力符号の関係

| VADCn                    | 読み出し符号 | 対応する10進値 |
|--------------------------|--------|----------|
| VADCm+VREF/GAIN          | \$1FF  | 511      |
| VADCm+(511/512)VREF/GAIN | \$1FF  | 511      |
| VADCm+(510/512)VREF/GAIN | \$1FE  | 510      |
| }                        | }      | }        |
| VADCm+(1/512)VREF/GAIN   | \$001  | 1        |
| VADCm                    | \$000  | 0        |
| VADCm-(1/512)VREF/GAIN   | \$3FF  | -1       |
| }                        | }      | }        |
| VADCm-(511/512)VREF/GAIN | \$201  | -511     |
| VADCm-VREF/GAIN          | \$200  | -512     |

図119. 差動入力計測範囲



例: ADMUX=\$ED(ADC3とADC2差動10倍入力、VREF=2.56V、左揃え)で、ADC3が300mV、ADC2が500mVの場合、A/D変換値=512×10×(300-500)÷2560=-400=\$270  
従って、ADCH=\$9C、ADCL=\$00、ADLAR=0なら、ADCH=\$02、ADCL=\$70。



## A/D変換用レジスタ

### A/D多重器選択レジスタ (ADC Multiplexer Select Register) ADMUX

|             |       |       |       |      |      |      |      |      |       |
|-------------|-------|-------|-------|------|------|------|------|------|-------|
| ビット         | 7     | 6     | 5     | 4    | 3    | 2    | 1    | 0    |       |
| \$07 (\$27) | REFS1 | REFS0 | ADLAR | MUX4 | MUX3 | MUX2 | MUX1 | MUX0 | ADMUX |
| Read/Write  | R/W   | R/W   | R/W   | R/W  | R/W  | R/W  | R/W  | R/W  |       |
| 初期値         | 0     | 0     | 0     | 0    | 0    | 0    | 0    | 0    |       |

#### •ビット7,6 – REFS1,0 : 基準電圧選択 (Reference Select Bits 1, 0)

これらのビットは表97.で示されるように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=1)まで実施しません。AREFピンに外部基準電圧が印加される場合、内部の基準電圧が使われてはなりません。

表97. A/D変換部の基準電圧選択

| REFS1 | REFS0 | 基準電圧  |
|-------|-------|---|
| 0     | 0     | AREFピンの外部基準電圧 (AVCCと内部2.56V基準電圧は切り離されます。)           |
| 0     | 1     | AVCC (内部2.56V基準電圧は切り離されますが、AREFにデカップ用コンデンサが接続できます。) |
| 1     | 0     | (予約)  |
| 1     | 1     | 内部2.56V基準電圧 (AVCCは切り離されますが、AREFにデカップ用コンデンサが接続できます。) |

#### •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データレジスタの内容に影響を及ぼします。このビットの完全な記述については162頁の「A/Dデータレジスタ」をご覧ください。

#### •ビット4~0 – MUX4~0 : A/Dチャネル選択 (Analog Channel Select Bits 4~0)

これらのビットの値はA/D変換器にどのアナログ入力のリ組み合わせが接続されるかを選びます。これらのビットは差動チャネルに対する利得も選びます。詳細については表98.をご覧ください。これらのビットが変換中に変更される場合、その変更は変換完了(ADCSRAのADIF=1)まで実施しません。

表98. アナログ入力チャネル選択

| MUX4~0     | シングルエント<br>入力 | 差動入力  |      |      | MUX4~0     | シングルエント<br>入力 | 差動入力  |      |    |
|------------|---------------|-------|------|------|------------|---------------|-------|------|----|
|            |               | 非反転入力 | 反転入力 | 利得   |            |               | 非反転入力 | 反転入力 | 利得 |
| 0000       | ADC0          |       |      |      | 10000      |               | ADC0  | ADC1 | ×1 |
| 00001      | ADC1          |       |      |      | 10001 (注1) |               | ADC1  | ADC1 | ×1 |
| 00010      | ADC2          |       |      |      | 10010      |               | ADC2  | ADC1 | ×1 |
| 00011      | ADC3          |       |      |      | 10011      |               | ADC3  | ADC1 | ×1 |
| 00100      | ADC4          |       |      |      | 10100      |               | ADC4  | ADC1 | ×1 |
| 00101      | ADC5          |       |      |      | 10101      |               | ADC5  | ADC1 | ×1 |
| 00110      | ADC6          |       |      |      | 10110      |               | ADC6  | ADC1 | ×1 |
| 00111      | ADC7          |       |      |      | 10111      |               | ADC7  | ADC1 | ×1 |
| 01000 (注1) |               | ADC0  | ADC0 | ×10  | 11000      |               | ADC0  | ADC2 | ×1 |
| 01001      |               | ADC1  | ADC0 | ×10  | 11001      |               | ADC1  | ADC2 | ×1 |
| 01010 (注1) |               | ADC0  | ADC0 | ×200 | 11010 (注1) |               | ADC2  | ADC2 | ×1 |
| 01011      |               | ADC1  | ADC0 | ×200 | 11011      |               | ADC3  | ADC2 | ×1 |
| 01100 (注1) |               | ADC2  | ADC2 | ×10  | 11100      |               | ADC4  | ADC2 | ×1 |
| 01101      |               | ADC3  | ADC2 | ×10  | 11101      |               | ADC5  | ADC2 | ×1 |
| 01110 (注1) |               | ADC2  | ADC2 | ×200 | 11110      | 1.23V(VBG)    |       |      |    |
| 01111      |               | ADC3  | ADC2 | ×200 | 11111      | 0V(GND)       |       |      |    |

注1: 変位(オフセット)校正に使えます。

## A/D変換 制御/状態レジスタA (ADC Control and Status Register A) ADCSRA

| ビット         | 7    | 6    | 5    | 4    | 3    | 2     | 1     | 0     |        |
|-------------|------|------|------|------|------|-------|-------|-------|--------|
| \$06 (\$26) | ADEN | ADSC | ADFR | ADIF | ADIE | ADPS2 | ADPS1 | ADPS0 | ADCSRA |
| Read/Write  | R/W  | R/W  | R/W  | R/W  | R/W  | R/W   | R/W   | R/W   |        |
| 初期値         | 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A/D変換部(動作)を許可します。**0**を書くことによってA/D変換部は(電源が)OFFされます。変換が進行中にA/D変換部を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。A/D変換部が許可される(ADEN=**1**)と同時にADSCが書かれるか、またはA/D変換部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13に代わって25変換クロック周期で行います。この初回変換はA/D変換部の初期化を実行します。

ADSCは変換が進行中である限り**1**として読みます。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 – ADFR : 連続/単独変換動作選択 (ADC Free Running Select)

このビットが設定(**1**)されるとA/D変換器は連続変換動作で動作します。この動作種別では連続的にA/D変換の採取/変換とA/Dデータレジスタを更新します。このビットを解除(**0**)することは連続変換動作を終了します(単独変換動作になります)。

### • 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。**ステータスレジスタ(SREG)の全割り込み許可(I)ビット**とA/D変換完了割り込み許可(ADIE)ビットが設定(**1**)されていれば、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。ADCSRA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、SREGの全割り込み許可(I)ビットが設定(**1**)されていると、A/D変換完了割り込みが活性に(許可)されます。このビットが解除(**0**)されると、この割り込みは禁止されます(**訳注**:共通性のため本行追加)。

### • 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XTAL(システム)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99. A/D変換クロック選択 (CK=システムクロック)

|           |      |      |      |      |       |       |       |        |
|-----------|------|------|------|------|-------|-------|-------|--------|
| ADPS2     | 0    | 0    | 0    | 0    | 1     | 1     | 1     | 1      |
| ADPS1     | 0    | 0    | 1    | 1    | 0     | 0     | 1     | 1      |
| ADPS0     | 0    | 1    | 0    | 1    | 0     | 1     | 0     | 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 |



## A/Dデータレジスタ (ADC Data Register) ADCH,ADCL

| ADLAR=0時    |      |      |      |      |      |      |      |      |      |
|-------------|------|------|------|------|------|------|------|------|------|
| ビット         | 15   | 14   | 13   | 12   | 11   | 10   | 9    | 8    |      |
| \$05 (\$25) | -    | -    | -    | -    | -    | -    | ADC9 | ADC8 | ADCH |
| Read/Write  | R    | R    | R    | R    | R    | R    | R    | R    |      |
| 初期値         | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |      |
| ビット         | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |      |
| \$04 (\$24) | ADC7 | ADC6 | ADC5 | ADC4 | ADC3 | ADC2 | ADC1 | ADC0 | ADCL |
| Read/Write  | R    | R    | R    | R    | R    | R    | R    | R    |      |
| 初期値         | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |      |
| ADLAR=1時    |      |      |      |      |      |      |      |      |      |
| ビット         | 15   | 14   | 13   | 12   | 11   | 10   | 9    | 8    |      |
|             | ADC9 | ADC8 | ADC7 | ADC6 | ADC5 | ADC4 | ADC3 | ADC2 | ADCH |
| ビット         | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |      |
|             | ADC1 | ADC0 | -    | -    | -    | -    | -    | -    | ADCL |

A/D変換が完了すると、その結果がこの2つのレジスタで得られます。差動チャンネルが使われると、結果は2の補数形式で表されます。

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が先で、その後にADCHが読まれなければなりません。

A/D多重器選択レジスタ(ADMUX)の左揃え選択(ADLAR)ビットとA/Dチャンネル選択(MUX4~0)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。

- ADC9~0 : A/D変換結果 (ADC Conversion result)

これらのビットは159頁の「A/D変換の結果」で詳述されるように変換での結果を表します。

## JTAGインターフェースと内蔵デバッグ機能

### 特徴

- JTAGインターフェース (IEEE標準1149.1準拠)
- JTAG規格に従った境界走査(Boundary-Scan)能力
- デバッグは各部のアクセスが可能
  - 全ての内蔵周辺機能
  - 内部及び外部RAM
  - 内蔵レジスタファイル
  - プログラムカウンタ
  - EEPROM及びフラッシュメモリ
- 中断(BREAK)によって支援される広範囲な内蔵デバッグ機能
  - AVRのBREAK命令
  - プログラムの流れ変更での停止
  - 1命令実行(シングルステップ)停止
  - プログラムメモリ上の単一アドレスまたはアドレス範囲による中断点(ブレイクポイント)
  - データメモリ上の単一アドレスまたはアドレス範囲による中断点
- JTAGインターフェースを介してのフラッシュメモリ、EEPROM、ヒューズビット、施錠ビットの読み書き(プログラミング)
- AVR Studioによる内蔵デバッグ機能の支援

### 概要

AVRのIEEE標準1149.1準拠JTAGインターフェースは以下について使えます。

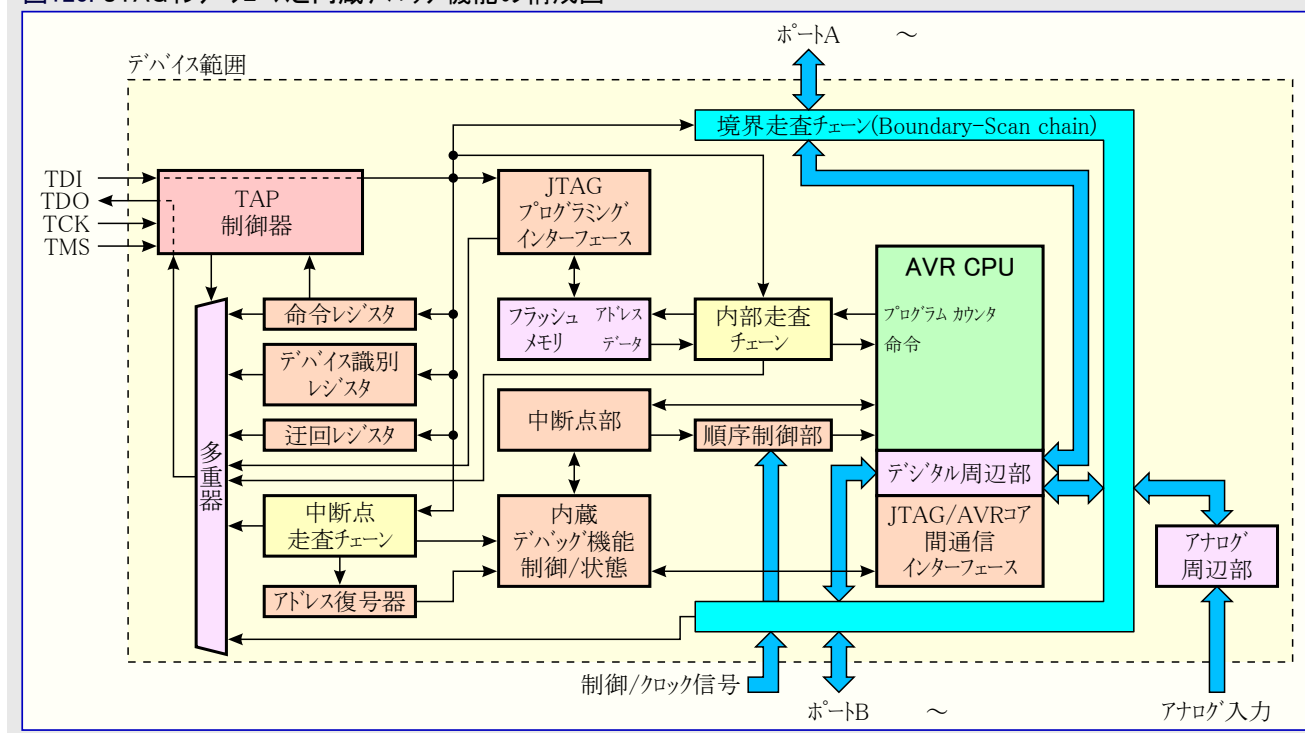
- JTAG境界走査(Boundary-Scan)能力による基板などの試験
- 不揮発性メモリ、ヒューズビット、施錠ビットの読み書き(プログラミング)
- 内蔵デバッグ機能による実装デバッグ

簡単な記述は次節で与えられます。JTAGインターフェース経由のプログラミングと境界走査(Boundary-Scan)チェーンに関する詳細な記述は、各々199頁の「JTAGインターフェース経由のプログラミング」と167頁の「IEEE 1149.1(JTAG)境界走査(Boundary-Scan)」で得られます。内蔵デバッグ機能の支援は独自JTAG命令で考慮されており、Atmelと選ばれた業者のみに配布されます。

図120はJTAGインターフェースと内蔵デバッグ機能を示します。TAP(検査入出力ポート)制御器はTMSとTCKによって制御される順次回路です。TAP制御器はTDI入力とTDO出力間の走査チェーン(移動レジスタ接続)として、JTAG命令レジスタまたは各種データレジスタの1つのどれかを選びます。JTAG命令レジスタはデータレジスタの動きを制御するJTAG命令を保持します。

デバイス識別(Identification)レジスタ、迂回(Bypass)レジスタ、境界走査チェーン(Boundary-Scan chain)の各データレジスタは基板段階の試験に使われます。(実際には現実と仮定の各種データレジスタで構成する)JTAGプログラミングインターフェースはJTAGインターフェース経由での直列プログラミングに使われます。内部走査チェーン(Scan chain)と中断点(ブレイクポイント)走査チェーンは内部デバッグ機能だけで使われます。

図120. JTAGインターフェースと内蔵デバッグ機能の構成図



## 検査入出力ポート (TAP:Test Access Port)

JTAGインターフェースはAVRの4つのピンを通してアクセスされます。JTAG用語ではこれらのピンが検査入出力ポート(TAP)を構成します。

- **TMS** : 検査種別選択。このピンはTAP制御器順次回路を通しての指示に使われます。
- **TCK** : 検査クロック。JTAG操作はTCKに同期します。
- **TDI** : 検査データ入力。命令レジスタまたはデータレジスタ(走査チェーン)内に移されるべき直列入力データです。
- **TDO** : 検査データ出力。命令レジスタまたはデータレジスタからの直列出力データです。

IEEE標準1149.1では任意TAP信号の検査リセット(TRST)も記載されていますが、この信号は提供されません。

JTAGエンハンスが非プログラム(1)の時にこれら4つのTAPピンは標準ポートピンで、TAP制御器はリセットです。プログラム(0)され、MCU制御/状態レジスタ(MCUCSR)のJTAGインターフェース禁止(JTD)ビットが解除(0)されると、TAP信号入力は内部的にHighへ引かれ、JTAGは境界走査(Boundary-Scan)とプログラミングに関して許可されます。この場合、TAP出力(TDO)ピンはJTAG TAP制御器がデータを移動していない状態で浮き状態(フローティング)のままにされ、従ってプルアップ抵抗が接続されるか、または他のハードウェア(例えば走査チェーン内の次のデバイスのTDI入力)がプルアップを持たなければなりません。このデバイスはこのヒューズがプログラム(0)されて出荷されます。

内蔵デバッグ機能ではJTAGインターフェースピンに加え、外部リセット元が検知できるため、RESETピンがデバッグによって監視されます。応用でリセット信号線にオープンコレクタ(トレイン)だけが使われるとすれば、デバッグは全システムをリセットするためにRESETピンをLowにすることもできます。

## TAP制御器

TAP制御器は境界走査(Boundary-Scan)回路、JTAGプログラミング回路、内蔵デバッグ機能の働きを制御する16段の無限順次回路です。図121.に描かれた状態遷移はTCKクロックの上昇時の(各状態遷移付近で示される)TMS上に存在する信号に依存します。電源投入リセット後の初期状態は検査回路リセットです。

この文書内の定義として、全ての移動レジスタに関して入出力ともLSBが最初に移動されます。

検査実行/アイドル状態が現在の状態と仮定した、JTAGインターフェースを使うための典型的な手順を次に示します。

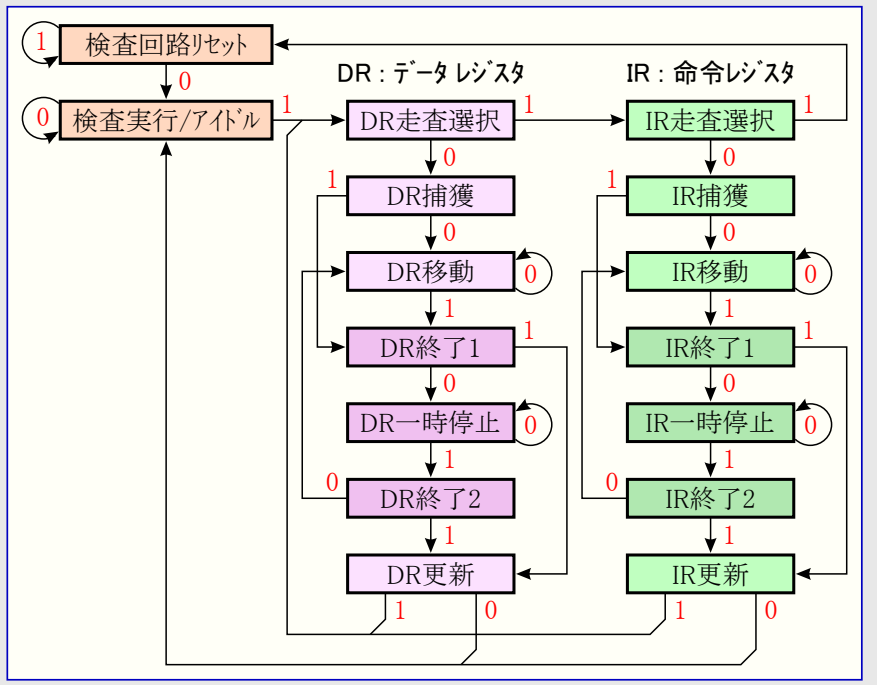
- 命令レジスタ(IR)移動状態へ移行するためにTCKの上昇でTMSへ順次1,1,0,0を与えます。この状態中、TCKの上昇端でTDI入力からJTAG命令レジスタ内に4ビットのJTAG命令を移動します。TMS入力はIR移動状態に留まるために3 LSBの入力中、Lowに保持されなければなりません。命令のMSBはTMS入力のHigh設定によってこの状態を抜ける時に移動入力されます。命令がTDIピンから移動されている間、捕獲されたIRの状態(\$01)がTDOピンへ移動出力されます。JTAG命令はTDIとTDO間に通ず(接続する)特定のデータレジスタを選び、選んだデータレジスタ周辺回路を制御します。
- 検査実行/アイドル状態へ再度移行するためにTMSへ順次1,1,0を与えます。この(JTAG)命令はIR更新状態通過で移動レジスタから並列出力にラッチされます。IR終了1、IR一時停止、IR終了2の各状態は順次回路の誘導操作のためだけに使われます。
- データレジスタ(DR)移動状態へ移行するためにTCKの上昇端でTMSへ順次1,0,0を与えます。この状態中、TCKの上昇端でTDI入力から(JTAG命令レジスタ内の現在のJTAG命令で)選んだデータレジスタに値を移動入力します。DR移動状態に留まるためにMSBを除く全ビットの入力中、TMS入力はLowに保持されなければなりません。データのMSBはTMS入力のHigh設定によってこの状態を抜ける時に移動入力されます。データレジスタがTDIピンから移動入力される間、DR捕獲状態(1,0)でデータレジスタに捕獲した並列入力がTDOピンへ移動出力されます。
- 検査実行/アイドル状態へ再移行するためにTMSへ順次1,1,0を与えます。選んだデータレジスタがラッチした並列出力を持つ場合、DR更新状態でラッチを行います。DR終了1、DR一時停止、DR終了2の各状態は順次回路の誘導操作のためだけに使われます。

状態遷移図で示されるように検査実行/アイドル状態はJTAG命令の選択とデータレジスタの使用間で移行される必要はなく、また、いくつかのJTAG命令は検査実行/アイドル状態で実行されるべき必然的な機能を選ぶかもしれず、それはアイドル状態として不適当になります。

**注:** TAP制御器の初期状態に関係なく、5 TCKクロック周期に対してTMSをHighに保持することにより、常に検査回路リセット状態へ移行することができます。

JTAG仕様の詳細情報については166頁の「参考文献」に記載された文献を参照してください。

図121. TAP制御器状態遷移図



## 境界走査チェーン(Boundary-Scan Chain)の使用

境界走査能力の完全な説明は167頁の「IEEE 1149.1(JTAG)境界走査(Boundary-Scan)」章で与えられます。

## 内蔵デバッグ機能の使用

図120.で示されるように内蔵デバッグに関するハードウェア支援は主に次の部分から成ります。

- 内部AVR CPUコアと内部周辺機能部間インターフェースの走査チェーン(Scan chain)
- 中断点(ブレイクポイント)部
- CPUとJTAGシステム間の通信インターフェース

デバッグの実行で必要とされる全ての読み込みまたは変更/書き込みは内部AVR CPU走査チェーン(Scan chain)経由のAVR命令によって行われます。CPUはCPUとJTAGシステム間の通信インターフェース部分が配置されるI/Oメモリ位置に結果を送ります。

中断点部はプログラムの流れ変更での一時停止(Break)、1命令実行(Single Step Break)、2つのプログラムメモリ用中断点、2つの複合中断点を実行します。合わせて4つ中断点は次のどれかとして設定ができます。

- 4つのプログラムメモリ単一中断点
- 3つのプログラムメモリ単一中断点 + 1つのデータメモリ単一中断点
- 2つのプログラムメモリ単一中断点 + 2つのデータメモリ単一中断点
- 2つのプログラムメモリ単一中断点 + 1つのプログラムメモリ範囲中断点(中断点と遮蔽)
- 2つのプログラムメモリ単一中断点 + 1つのデータメモリ範囲中断点(中断点と遮蔽)

けれどもAtmel® AVR Studio®のようなデバッグは最終使用者への柔軟性をより少なくする内部目的にこれら資源の1つまたはそれ以上を使うかもしれません。

内蔵デバッグ特殊JTAG命令の一覧は「内蔵デバッグ特殊JTAG命令」で与えられます。

JTAG検査入出力ポート(TAP)を許可するにはJTAGエンハンスがプログラム(0)されなければなりません。加えて内蔵デバッグ機能で作業するには更にOCDENヒューズがプログラム(0)され、施錠ビットが施錠なしに設定されなければなりません。何れかの施錠ビットが設定(0)されると、保護機能で内蔵デバッグ機能が禁止されます。そうでなければ内蔵デバッグ機能が保護されたデバイスへの裏口を提供することになっていでしょう(訳補:保護の意味がなくなるの意)。

AVR Studioは使用者にチップ内蔵デバッグ能力(機能)、AVRインサーキットエミュレータ、または(AVR Studio)組み込みAVR命令一式シミュレータでAVRデバイスのプログラム実行を完全に制御することを可能にします。AVR StudioはAtmel AVRアセンブラでアセンブルしたアセンブリ言語プログラムと他業者製コンパイラでコンパイルしたC言語プログラムのソースレベル実行を支援します。

AVR StudioはMicrosoft Windows® 98/2000、Windows NT®, Windows XP®, Windows Vista®下で走行します。

AVR Studioの完全な記述についてはAVR Studio使用者の手引きを参照してください。要点だけが本文書で示されます。

全ての必要な実行指令はAVR Studio内のソースレベルと逆アセンブリレベルの両方で利用できます。使用者はプログラムの実行、1命令実行(Single Step)、内側(関数内)実行、外側(関数外)実行、命令行へのカーソル配置と命令行が(カーソル行へ)到達されるまでの実行、一時停止、実行対象のリセットができます。加えて使用者は無制限数のコードの中断点(BREAK命令使用)と、2つまでのデータメモリの中断点か、代わりに遮蔽による範囲指定(複合)中断点を持てます。

## 内蔵デバッグ特殊JTAG命令

内蔵デバッグの支援は独自JTAG命令で考慮され、Atmelと選ばれた業者だけに配布されます。参考のため、命令コードの一覧が示されます。

- 独自命令0 : \$8 (内蔵デバッグ機能アクセス用独自JTAG命令)
- 独自命令1 : \$9 (内蔵デバッグ機能アクセス用独自JTAG命令)
- 独自命令2 : \$A (内蔵デバッグ機能アクセス用独自JTAG命令)
- 独自命令3 : \$B (内蔵デバッグ機能アクセス用独自JTAG命令)

## 内蔵デバッグに関連するI/Oメモリ内のレジスタ

### 内蔵デバッグレジスタ(On-chip Debug Register) OCDR

| ビット         | 7        | 6   | 5   | 4   | 3   | 2   | 1   | 0   |      |
|-------------|----------|-----|-----|-----|-----|-----|-----|-----|------|
| \$22 (\$42) | IDRD/MSB |     |     |     |     |     |     | LSB | OCDR |
| Read/Write  | R/W      | R/W | R/W | R/W | R/W | R/W | R/W | R/W |      |
| 初期値         | 0        | 0   | 0   | 0   | 0   | 0   | 0   | 0   |      |

OCDRはマイクロコントローラ内の実行プログラムからデバッグへの通信チャネルを提供します。CPUはこの位置へ書くことによってデバッグへバイトを転送できます。このレジスタが書かれてしまっているのをデバッグに示すため、同時に内部フラグIDRD(I/Oデバッグレジスタ更新)が設定(1)されます。CPUがOCDRを読むとき、LSB 7ビットがOCDRからで、一方MSBはIDRDビットです。デバッグはこの情報を読む時にIDRDビットを解除(0)します。

いくつかのAVRデバイスに於いてこのレジスタは標準I/O位置で共用されます。この場合、OCDRは内蔵デバッグ許可(OCDEN)ヒューズがプログラム(0)された場合にだけアクセスでき、デバッグがOCDRへアクセスできます。他の全ての場合、標準I/O位置がアクセスされます。

このレジスタの使用法のより多くの情報はデバッグの資料を参照してください。

## JTAGプログラミング能力の使用

JTAG経由Atmel® AVR®デバイスプログラミングは4ピンのJTAGポート、TCK、TMS、TDI、TDO経由で実行されます。これらはJTAGプログラミング実行のために(電源ピンに加えて)制御/監視される必要があるピンだけです。外部的な12V印加は必要とされません。JTAG検査入出力ポート (TAP)を許可するにはJTAGENヒューズがプログラム(0)され、MCU制御/状態レジスタ(MCUCSR)のJTAGインターフェース禁止(JTD)ビットが解除(0)されなければなりません。

JTAGプログラミング能力は以下を支援します。

- フラッシュメモリのプログラミングと照合
- EEPROMのプログラミングと照合
- ヒューズビットのプログラミングと照合
- 施錠ビットのプログラミングと照合

施錠ビット保護は丁度並列プログラミング動作でのようです。施錠ビットLB1またはLB2がプログラム(0)されると、先にチップ消去を行わない限り、OCDENヒューズはプログラム(0)できません。これは保護されたデバイスの内容を読み出す裏口が存在しないことを保証する保護機能です。

プログラミング特殊JTAG命令の記述は199頁の「[プログラミング特殊JTAG命令](#)」で与えられます。JTAGインターフェースを通すプログラミングの詳細は199頁の「[JTAGインターフェース経由のプログラミング](#)」項で与えられます。

## 参考文献

一般的な境界走査(Boundary-Scan)についての詳細は次の文献を参照してください。

- IEEE : IEEE Std 1149.1-1990. IEEE Standard Test Access Port and Boundary-Scan Architecture, IEEE, 1993
- Colin Maunder : The Board Designers Guide to Testable Logic Circuits, Addison Wesley, 1992



## IEEE 1149.1 (JTAG) 境界走査(Boundary-Scan)

### 特徴

- JTAGインターフェース (IEEE標準1149.1準拠)
- JTAG規格に従った境界走査(Boundary-Scan)能力
- チップ外接線を持つアナログ回路は勿論、全てのポート機能の完全な走査(Scan)
- 任意IDCODE命令の支援
- AVRリセット用公開AVR\_RESET命令の追加

### 概要

境界走査チェーン(Boundary-Scan chain)にはデジタルI/Oピンは勿論、チップ外接線を持つアナログ回路のアナログとデジタル間境界の論理値を駆動、監視する能力があります。システム上ではJTAG能力を持つ全てのICがTDIとTDOの接続により長い移動レジスタの形態で接続されます。外部制御器がこれらの出力ピンで値を駆動するようにデバイスを設定し、他のデバイスから受け取った入力値を監視します。この制御器は予測される結果と受信した値を比較します。この方法で境界走査(Boundary-Scan)は4つのTAP信号だけを使うことにより、基板上の部品全般と内部結線の検査機構を提供します。

IEEE 1149.1で定義された4つの必須JTAG命令、IDCODE, BYPASS, SAMPLE/PRELOAD, EXTESTだけでなく、AVR特殊公開JTAG命令のAVR\_RESETも基板検査に使えます。既定JTAG命令がIDCODEなので、データレジスタ経路の初期走査はデバイスの識別符号(ID-code)を示します。検査動作中にリセット(状態)のAVRデバイスがあることは好都合かもしれませんが、リセットでない場合、デバイスの入力は走査動作によって決定されるかもしれず、検査動作を終了する時に内部ソフトウェアが未確定状態になってしまうかもしれません。リセットへの移行で、どのポートピンの出力も直ちにHi-Z状態に移行し、HIGHZ命令が不要になります。必要とされるなら、デバイスを通過する走査チェーンを可能な限り最短とするのにBYPASS命令が発行できます。外部RESETピンをLowに引き込むか、またはリセットデータレジスタの適切な設定によるAVR\_RESET命令の発行でデバイスをリセット状態に設定できます。

EXTEST命令は外部ピンの採取と出力ピンの値設定に使われます。EXTEST命令がJTAG命令レジスタ(IR)に設定されると、直ちに出力ラッチからの値がピンで出力駆動されます。このため、最初にEXTEST命令を発行する時に基板の損傷を避けるため、走査チェーン内に初期値を設定するのにSAMPLE/PRELOAD命令も使われるべきです。SAMPLE/PRELOAD命令は通常動作中デバイスの外部ピン的高速監視にも使えます。

JTAG検査入出力ポート(TAP)を許可するにはJTAGENヒューズがプログラム(0)され、MCU制御/状態レジスタ(MCUCSR)のJTAGインターフェース禁止(JTD)ビットが解除(0)されなければなりません。

境界走査(Boundary-Scan)にJTAGインターフェースを使う場合、内部チップ周波数よりも高いJTAG TCKクロック周波数の使用が可能です。チップのクロックは走行に必要なとされません。

### データレジスタ

境界走査(Boundary-Scan)に関連するデータレジスタ(DR)を次に示します。

- 迂回(Bypass)レジスタ
- デバイス識別(Device Identification)レジスタ
- リセット(Reset)レジスタ
- 境界走査チェーン(Boundary-Scan chain)

#### 迂回(Bypass)レジスタ

迂回レジスタは1段の移動レジスタから成ります。TDIとTDO間の経路として迂回レジスタが選ばれると、このレジスタはTAP制御器がデータレジスタ(DR)捕獲状態を抜ける時に0へリセットされます。迂回レジスタは他のデバイスが検査される時にシステム上の走査チェーン(Scan chain)を短くするのに使えます。

#### デバイス識別(Device Identification)レジスタ

図122.はデバイス識別レジスタの構造を示します。

図122. デバイス識別レジスタの形式

|        |     |    |    |    |      |   |       |
|--------|-----|----|----|----|------|---|-------|
| ビット    | 31  | 28 | 27 | 12 | 11   | 1 | 0     |
| デバイス識別 | 版番号 |    |    |    | 部品番号 |   | 製造者識別 |
| ビット数   | 4   |    |    |    | 16   |   | 1     |

**版番号** 版番号は部品(デバイス)の改訂を示す4ビットです。JTAG版番号はデバイスの改訂版に従い、改訂P(\$0F)で丸められます。改訂AとQは\$00, 改訂BとRは\$01, 以下同様です。

**部品番号** 部品番号は部品(名)を示す16ビットです。Atmel® AVR® ATmega128のJTAG部品番号は表100.で一覧されます。

表100. AVR JTAG 部品番号

| 部品番号      | JTAG部品番号(Hex) |
|-----------|---------------|
| ATmega128 | \$9702        |

**製造者識別** 製造者識別は製造業者を示す11ビットです。AtmelのJTAG製造者識別は表101.で示されます。

表101. 製造者ID

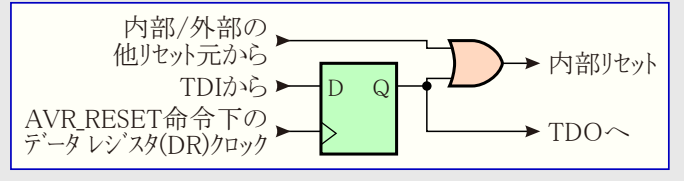
| 製造業者  | JTAG製造者ID番号(Hex) |
|-------|------------------|
| Atmel | \$01F            |

## リセット (Reset) レジスタ

リセットレジスタはデバイスのリセットに使われる検査データレジスタです。リセットによってAVRのポートピンがHi-Z状態となるので、未実装の任意JTAG命令HIGHZ機能の代わりにできます。

リセットレジスタ内の値1は外部RESETピンをLowに引っ張るのと同じです。デバイスはリセットレジスタ内の値が1である限り、リセットです。リセットレジスタの開放後、デバイスはクロック選択用ヒューズビットの設定に従ってリセット起動遅延時間(24ページの「クロック元」参照)、リセットに留まります。図123.で示されるようにこのデータレジスタからの出力はラッチされず、直ちにリセットが行われます。

図123. リセットレジスタ



## 境界走査チェーン(Boundary-Scan Chain)

境界走査チェーンにはデジタル/Oピンは勿論、チップ外の接続を持つアナログ回路のアナログとデジタル間境界の論理値上の設定と監視能力があります。

完全な記述については170ページの「境界走査チェーン(Boundary-Scan chain)」をご覧ください。

## 境界走査(Boundary-Scan)用JTAG命令

命令レジスタ(IR)は16種までの命令を支援する4ビット幅です。以下に示した一覧は境界走査(Boundary-Scan)操作に有用なJTAG命令です。全ポートピンをHi-Z状態にするための任意HIGHZ命令は実装されませんが、AVR\_RESET命令を使うことによって全出力がHi-Z状態に設定できることに注目してください。

このデータシート内の定義として、全ての移動レジスタに関して入出力ともLSBが最初に移動されます。

各命令の命令コードは命令名下の16進形式で示されます。本文は各命令についてTDIとTDO間の経路として選ばれるデータレジスタを記述します。(訳注:配置構成変更により、内容を一部変更)

### EXTEST - \$0

AVRデバイスに外部回路的検査を行うためにデータレジスタ(DR)として境界走査チェーン(Boundary-Scan chain)を選ぶための必須JTAG命令です。ポートピンは方向、プルアップ、出力値、入力値は走査チェーン(Scan chain)内で全てアクセス可能です。チップ外接線を持つアナログ回路についてはアナログとデジタル論理間のインターフェースが走査チェーン内になります。境界走査チェーンのラッチした出力値はJTAG命令レジスタ(IR)にEXTEST命令が設定されると直ぐに駆動されます。本命令が有効な状態を次に示します。

- DR捕獲：外部ピン上の値が境界走査チェーン内に採取されます。
- DR移動：内部走査チェーンがTCKクロック入力によって移されます。
- DR更新：走査チェーンからの値が出力ピンに印加(出力)されます。

### IDCODE - \$1

データレジスタ(DR)として32ビットのデバイス識別(ID)レジスタを選ぶ任意JTAG命令です。デバイス識別レジスタは版番号、デバイス番号と、JED ECによって決められた製造者符号から成ります。これは電源投入後の既定命令です。本命令が有効な状態を次に示します。

- DR捕獲：デバイス識別レジスタ内の値が境界走査チェーン内に採取されます。
- DR移動：内部走査チェーン(デバイス識別レジスタ値)がTCKクロック入力によって移されます。

### SAMPLE\_PRELOAD - \$2

システム動作に影響を与えずに入出力ピンの状態採取、出力ラッチの事前設定を行うための必須JTAG命令です。けれども出力ラッチはピンに接続されません。データレジスタ(DR)として境界走査チェーン(Boundary-Scan chain)が選ばれます。本命令が有効な状態を次に示します。

- DR捕獲：外部ピン上の値が境界走査チェーン内に採取されます。
- DR移動：境界走査チェーンがTCKクロック入力によって移されます。
- DR更新：境界走査チェーンからの値が出力ラッチに印加(設定)されます。けれども出力ラッチはピンに接続されません。

### AVR\_RESET - \$C

AVRデバイスを強制的にリセット状態、またはJTAGリセット元から開放するためのAVR特殊公開JTAG命令です。この命令で検査入出力ポート(TAP)制御器はリセットされません。データレジスタ(DR)として1ビットのリセットレジスタが選ばれます。リセットチェーン(レジスタ)が論理1である限り、リセットが活性(有効)なことに注意してください。このチェーンからの出力はラッチされません。本命令が有効な状態を次に示します。

- DR移動：走査チェーン(リセットレジスタ)がTCKクロック入力によって移されます。

### BYPASS - \$F

データレジスタ(DR)に迂回(Bypass)レジスタを選ぶ必須JTAG命令です。本命令が有効な状態を次に示します。

- DR捕獲：迂回レジスタに論理0を設定します。
- DR移動：TDIとTDO間の迂回レジスタ(セル)が移されます。

## 境界走査(Boundary-Scan)に関連するI/Oメモリ内のレジスタ

## MCU制御/状態レジスタ (MCU Control and Status Register) MCUCSR

MCU制御/状態レジスタは一般MCU機能制御ビットを含み、MCUリセットを起こしたリセット元の情報を提供します。

| ビット         | 7   | 6 | 5 | 4    | 3    | 2    | 1     | 0    |        |
|-------------|-----|---|---|------|------|------|-------|------|--------|
| \$34 (\$54) | JTD | - | - | JTRF | WDRF | BORF | EXTRF | PORF | MCUCSR |
| Read/Write  | R/W | R | R | R/W  | R/W  | R/W  | R/W   | R/W  |        |
| 初期値         | 0   | 0 | 0 | 内容参照 | 内容参照 | 内容参照 | 内容参照  | 内容参照 |        |

- ビット7 - JTD : JTAGインターフェース禁止 (JTAG interface disable)

このビットが0の時にJTAGENヒューズがプログラム(0)されていれば、JTAGインターフェースが許可されます。このビットが1の場合、JTAGインターフェースは禁止されます。JTAGインターフェースの予期せぬ許可や禁止を避けるためにこのビットを変更する時は時間制限手順に従わなければなりません。この値を変更するために応用ソフトウェアは4周期以内に希望する値をこのビットへ2回書かなければなりません。

JTAGインターフェースが他のJTAG回路に接続されないままにされる場合、このJTDビットは1に設定されるべきです。この理由はJTAGインターフェースのTDOピンの静止時電流を避けるためです。

- ビット4 - JTRF : JTAG リセット フラグ (JTAG Reset Flag)

このビットはJTAG命令AVR\_RESETによって選んだJTAG リセットレジスタ内の論理1でリセットが起こされると設定(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。

## 境界走査チェーン(Boundary-Scan Chain)

境界走査チェーンにはデジタルI/Oピンは勿論、チップ外の接続を持つアナログ回路のアナログとデジタル間境界の論理値の設定と監視の能力があります。

### デジタルポートピンの走査

図124.はプルアップ機能付き双方向ポートピン用境界走査(Boundary-Scan)セルを示します。このセルはプルアップ許可(PUExn)用標準境界走査(Boundary-Scan)セルと3つの信号、出力(方向)制御(OCxn)、出力データ(ODxn)、入力データ(IDxn)の組み合わせの双方向ピン用セルの2段の移動レジスタだけから成ります。ポートとピンの添え字は続く記述で使われません。

境界走査(Boundary-Scan)論理回路はデータシート内の図に含まれません。図125.は41頁の「入出力ポート」章で記載される単純なデジタルポートピンを示します。図124.からの境界走査セルの詳細が図125.上の破線内に置き換わります。

交換ポート機能が存在しない場合、入力データ(ID)はPINxnレジスタ値(しかしIDは同期化回路を持たない)、出力データ(OD)はPORTxnレジスタ、出力制御(OC)はDDxn方向レジスタ、プルアップ許可(PUE)は論理合成値(PUD AND DDxn AND PORTxn)に各々対応します。

デジタル交換ポート(ピン)機能は走査チェーン(Scan chain)が実際のピン値を読むために図125.内の破線外側に接続されます。アナログ機能について、それらは外部ピンからアナログ回路へ直結で、走査チェーンはアナログ回路とデジタル論理回路間のインターフェースに挿入されます。

図124. プルアップ機能付き双方向ポート用境界走査(Boundary-Scan)セル構成図

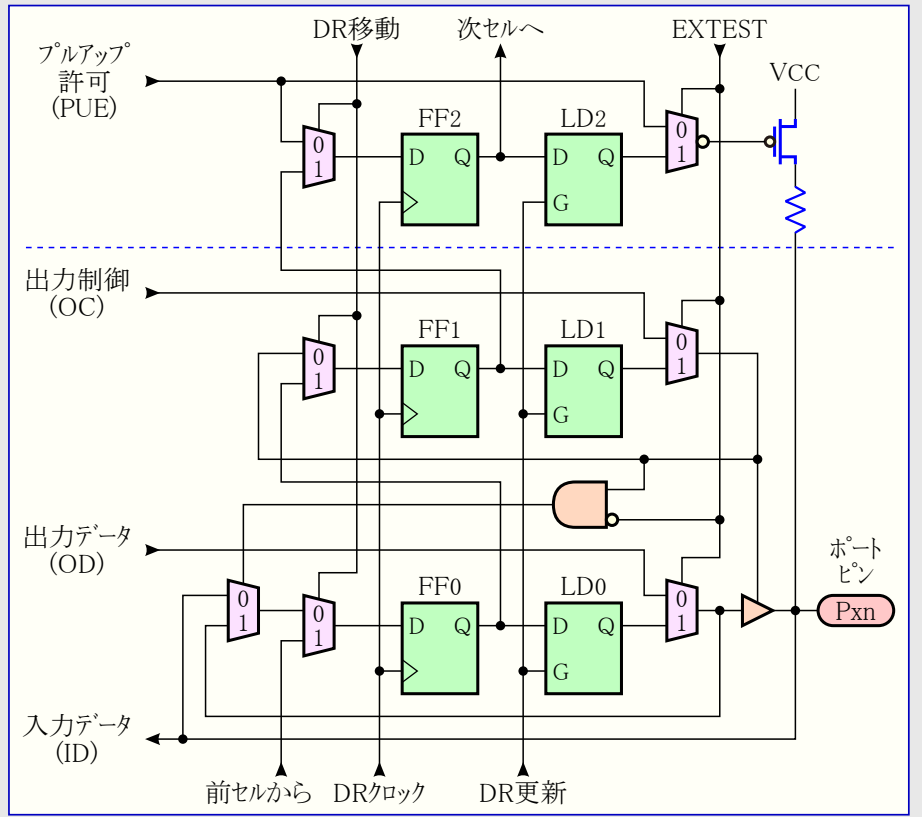
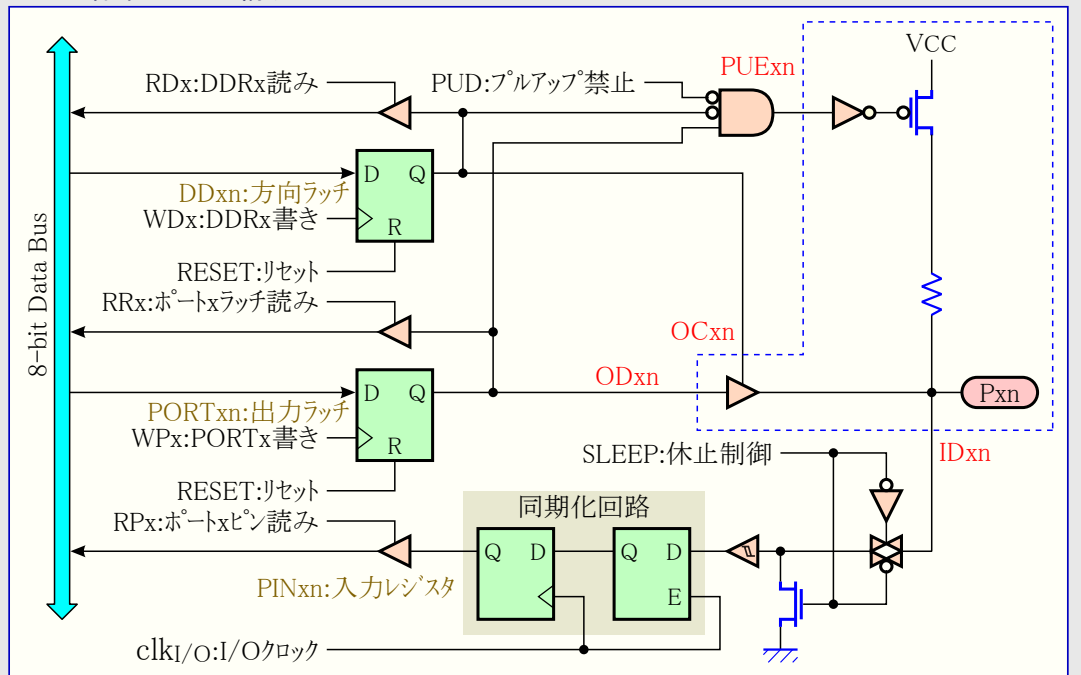


図125. 標準ポートピン構成図



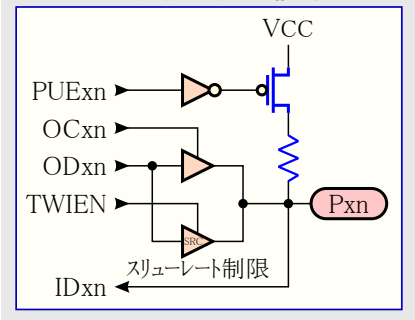
注: 詳細については境界走査(Boundary-Scan)をご覧ください。

## 境界走査と2線(直列)インターフェース

2線インターフェースピンのSCLとSDAは走査チェーン内に1つの追加制御信号TWIEN(2線インターフェース許可)を持ちます。図126.で示されるようにこのTWIEN信号は通常のデジタルポートピンと並列のスレーブ(上昇/下降)制御付き3状態緩衝器を許可します。図130.で示される標準走査セルがTWIEN信号に付随します。

- 注:** 1. 入力50nsスパイク除去用の独立した走査チェーンは提供されません。デジタルポートピンに対する標準走査の支援は接続検査に対して充分です。走査経路内にTWIENを持っている理由は境界走査を行う時にスレーブ制御緩衝器を切断できることだけです。
2. 駆動出力の衝突を引き起こさないよう、OCとTWIEN信号は同時に有効とされないように注意してください。

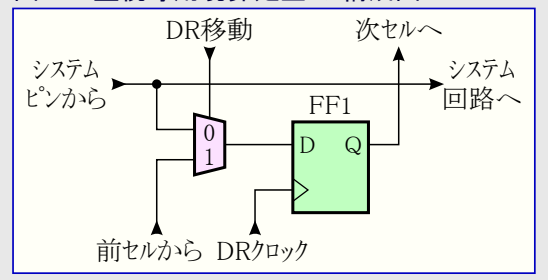
図126. TWI用追加走査信号



## RESETピンの走査

RESETピンは標準リセット動作について5V負論理(Low有効)、高電圧並列プログラミングについて12V正論理(High有効)を受け入れます。図127.で示される監視専用セルが5Vリセット信号(RSTT)と12Vリセット信号(RSTHV)の両方に挿入されます。

図127. 監視専用境界走査セル構成図



## クロックピンの走査

Atmel® AVR®デバイスにはヒューズによって選択可能な多くのクロック種別があります。これらは校正付き内蔵RC発振器、外部RC発振器、外部クロック信号、(高周波数)水晶発振子、低周波数水晶発振子、セラミック振動子です。

図128.は走査チェーン内で支援される各発振器と外部接続を示します。許可信号は標準境界走査セルで支援され、一方発振器/クロック出力には監視専用セルが付随されます。主クロックに加え、タイマ用発振器も同じ方法で走査されます。校正付き内蔵RC発振器からの出力はこの発振器が外部接続を持たないので走査されません。

図128. クロック機能と発振器用境界走査セル構成図

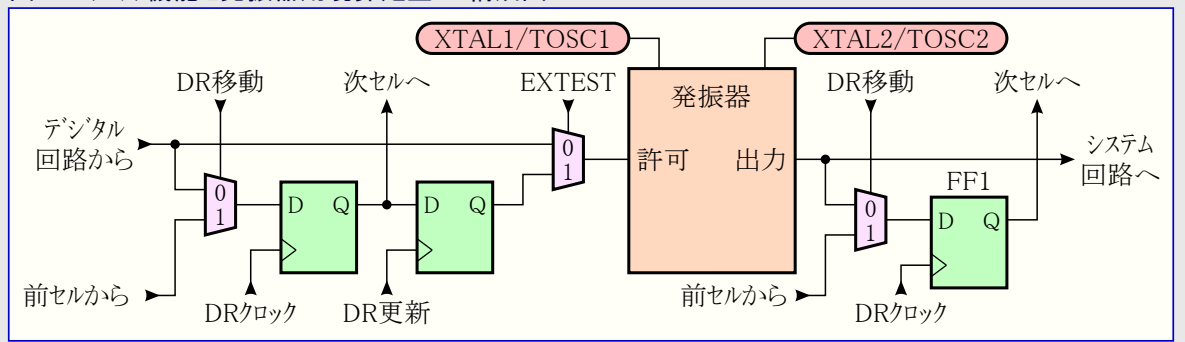


表102.はタイマ用32kHz発振器は勿論、外部クロックピンXTAL1、XTAL1/XTAL2接続の発振器に対する走査レジスタの一覧です。



表102. 発振器用走査信号

| 許可信号     | 走査クロック信号線     | クロック種別                   | 未使用時値 |
|----------|---------------|--------------------------|-------|
| EXTCLKEN | EXTCLK(XTAL1) | 外部クロック信号                 | 0     |
| OSCON    | OSCK          | 外部クリスタル発振子<br>外部セラミック振動子 | 0     |
| RCOSCEN  | RCCK          | 外部RC発振                   | 1     |
| OSC32EN  | OSC32CK       | 低周波数外部クリスタル              | 0     |
| TOSKON   | TOSCK         | タイマ用32kHz発振器             | 0     |

- 注: 1. 主クロックとして同時に1つよりも多くのクロック元を許可してはいけません。
2. 発振器出力の走査は内部発振器とJTAGのTCKクロック間の周波数偏差のため、予期せぬ結果を与えます。可能なら、外部クロックでの走査が望まれます。
3. クロック設定はヒューズによってプログラミングされます。ヒューズは実行動作時に変更されないため、クロック設定は与えられた応用について考慮して決定されます。使用者は最終システムで使われるのと同じクロック選択で走査するのを推奨されます。システム論理回路が休止形態でクロック選択(元)を禁止できるため、許可信号は走査チェーン内で支援され、それ(その禁止)によって(クロックが)供給されない場合、発振器ピンを走査経路から切り離します。CKOPTヒューズ(訳注:原書のINTCAPは誤り)は走査チェーン内で支援されないため、このヒューズが正しくプログラム(0)される場合を除いて、境界走査チェーンは内部コンデンサを必要とするXTAL発振器を形成できません。

### アナログ比較器の走査

境界走査に関する比較器関連の信号は図129.で示されます。図130.の境界走査セルがこれら信号の各々に付随します。この信号は表103.で記述されます。

この比較器は全てのアナログ入力がデジタルポートピンとしても兼用されるため、純粋な接続試験用に使用される必要がありません。

図129. アナログ比較器部構成図

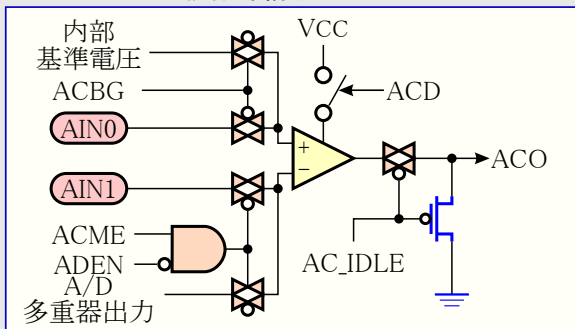


図130. 比較器、A/D変換用標準境界走査セル構成図

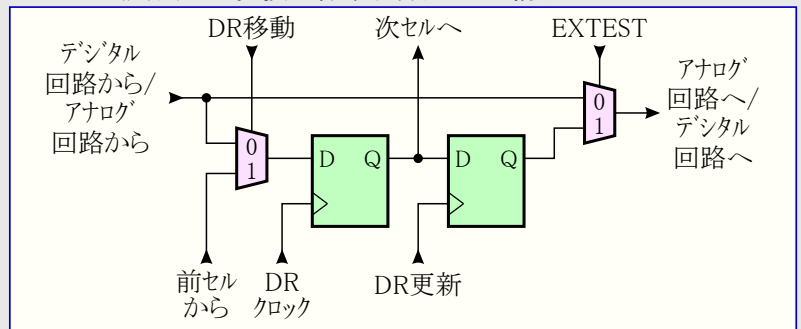


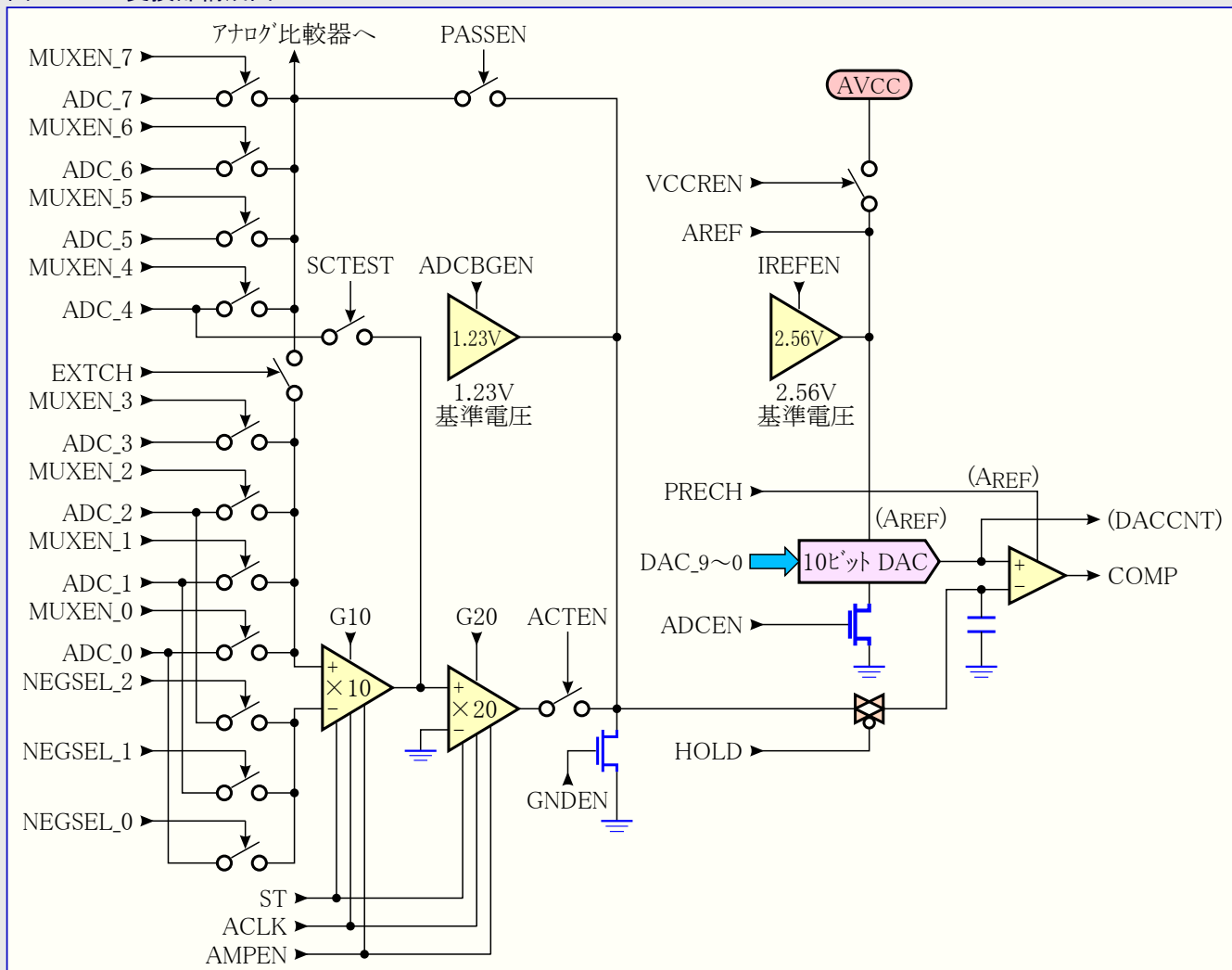
表103. アナログ比較器用境界走査信号

| 信号名     | 比較器側での方向 | 意味             | 未使用時推奨入力値 | 推奨入力時出力値 |
|---------|----------|----------------|-----------|----------|
| AC_IDLE | 入力       | 真の時にアナログ比較器OFF | 1         | CPU実行に依存 |
| ACO     | 出力       | アナログ比較器出力      | CPU実行時入力  | 0        |
| ACME    | 入力       | 真の時にA/D多重器出力使用 | 0         | CPU実行に依存 |
| ACBG    | 入力       | 内蔵基準電圧許可       | 0         | CPU実行に依存 |

## A/D変換器の走査

図131.はA/D変換部と関連する制御と監視信号の構成図を示します。図130.の境界走査セルがこれら信号の各々に付随されます。このA/D変換器は、全てのアナログ入力デジタルポートピンとしても兼用されるため、純粋な接続試験に使われる必要がありません。

図131. A/D変換部構成図



各信号の概要は表104.で示されます。

表104. A/D変換用境界走査信号

| 信号名      | A/D変換器側での方向 | 意味  | 未使用時推奨入力値 | 推奨入力または未使用時出力値 |
|----------|-------------|---|-----------|----------------|
| COMP     | 出力          | 比較器出力   | 0         | 0              |
| ACLK     | 入力          | 利得段のスイッチ キャパシタのクロック                             | 0         | 0              |
| ACTEN    | 入力          | 利得段⇒比較器接続許可                                     | 0         | 0              |
| ADCBGEN  | 入力          | 比較器反転入力へ内部1.23V基準電圧接続                           | 0         | 0              |
| ADCEN    | 入力          | A/D変換器電源ON信号                                    | 0         | 0              |
| AMPEN    | 入力          | 利得段電源ON信号                                       | 0         | 0              |
| DAC_9    | 入力          | D/A変換器デジタル入力ビット9                                | 1         | 1              |
| DAC_8    | 入力          | D/A変換器デジタル入力ビット8                                | 0         | 0              |
| DAC_7    | 入力          | D/A変換器デジタル入力ビット7                                | 0         | 0              |
| DAC_6    | 入力          | D/A変換器デジタル入力ビット6                                | 0         | 0              |
| DAC_5    | 入力          | D/A変換器デジタル入力ビット5                                | 0         | 0              |
| DAC_4    | 入力          | D/A変換器デジタル入力ビット4                                | 0         | 0              |
| DAC_3    | 入力          | D/A変換器デジタル入力ビット3                                | 0         | 0              |
| DAC_2    | 入力          | D/A変換器デジタル入力ビット2                                | 0         | 0              |
| DAC_1    | 入力          | D/A変換器デジタル入力ビット1                                | 0         | 0              |
| DAC_0    | 入力          | D/A変換器デジタル入力ビット0                                | 0         | 0              |
| EXTCH    | 入力          | ADC3~0の利得段迂回接続                                  | 1         | 1              |
| G10      | 入力          | 10倍利得許可   | 0         | 0              |
| G20      | 入力          | 20倍利得許可   | 0         | 0              |
| GNDEN    | 入力          | 比較器反転入力GND接続                                    | 0         | 0              |
| HOLD     | 入力          | S&H信号。0=採取、1=保持。利得段使用の場合ACLK=1の時に切り替えなければなりません。 | 1         | 1              |
| IREFEN   | 入力          | D/A変換器へのAREFとして内部2.56V基準電圧許可                    | 0         | 0              |
| MUXEN_7  | 入力          | ADC7 アナログ入力チャネル接続                               | 0         | 0              |
| MUXEN_6  | 入力          | ADC6 アナログ入力チャネル接続                               | 0         | 0              |
| MUXEN_5  | 入力          | ADC5 アナログ入力チャネル接続                               | 0         | 0              |
| MUXEN_4  | 入力          | ADC4 アナログ入力チャネル接続                               | 0         | 0              |
| MUXEN_3  | 入力          | ADC3 アナログ入力チャネル接続                               | 0         | 0              |
| MUXEN_2  | 入力          | ADC2 アナログ入力チャネル接続                               | 0         | 0              |
| MUXEN_1  | 入力          | ADC1 アナログ入力チャネル接続                               | 0         | 0              |
| MUXEN_0  | 入力          | ADC0 アナログ入力チャネル接続                               | 1         | 1              |
| NEGSEL_2 | 入力          | ADC2⇒反転差動入力接続                                   | 0         | 0              |
| NEGSEL_1 | 入力          | ADC1⇒反転差動入力接続                                   | 0         | 0              |
| NEGSEL_0 | 入力          | ADC0⇒反転差動入力接続                                   | 0         | 0              |
| PASSEN   | 入力          | 利得段迂回路許可  | 1         | 1              |
| PRECH    | 入力          | 比較器出力ラッチのプリチャージ(負論理)                            | 1         | 1              |
| SCTEST   | 入力          | スイッチ キャパシタ検査 (10倍利得段出力⇒ADC4)                    | 0         | 0              |
| ST       | 入力          | AMPEN=1後の最初の2ACLK周期間1ならば、利得段出力がより早く安定           | 0         | 0              |
| VCCREN   | 入力          | D/A変換器へのAREFとしてAVCC許可                           | 0         | 0              |

注: 図131.内の不正なスイッチ設定は信号衝突の原因になり、デバイスを損傷するかもしれません。それらは多くの入力を図131.で出力比較器の反転入力のS&Hへ選びます。1つのADCピン、内部基準電圧、GNDのどれか1つの経路だけが選ばれることを確認してください。

走査中にA/D変換器が使われないなら、表104.の推奨入力値が使われるべきです。使用者は走査中に差動段を使わないことが推奨されます。スイッチ キャパシタを基にした差動段は走査チェーン内で使われる時に達成が難しい高速な操作とタイミング精度が必要です。従って差動段の操作に関する詳細は提供されません。

Atmel® AVR®のA/D変換部はデジタル論理回路内に実装した逐次比較方式と図131.で示したアナログ回路が基です。境界走査使用時、問題は印加したアナログ電圧が或る範囲内で計測されるのを常に保証することです。これは逐次比較動作を行わずして容易に実行でき、DAC\_9~0デジタル(信号)線に下限値を加え、比較器からの出力がLow(0)であることを確認し、その後DAC\_9~0デジタル(信号)線に上限値を加え、比較器からの出力がHigh(1)であることを検証します。

このA/D変換器は全てのアナログ入力デジタルポートピンとしても兼用されるため、純粋な接続試験に使われる必要がありません。

A/D変換を使う時に次のことに留意してください。

- ADCチャンネルで使うポートピンは信号の衝突を避けるため、プルアップを禁止した入力に設定されなければなりません。
- 通常動作ではA/D変換部許可時に(10回の比較で構成される)擬似変換が実行されます。使用者はA/D変換許可後、A/D変換信号の制御/監視をする前に最低200ns待機するか、または最初の変換結果を使う前に擬似変換の実行を推奨されます。
- DAC値はHOLD信号がLow(0:採取)の時に中間値(\$200)で安定していなければなりません。

例として、電源電圧5VでAREFが外部的にVCCへ接続される時のADCチャンネル3で1.5V±5%の入力信号を検証する手順を考察してください。

$$\begin{aligned} \text{上限} &: 1024 \times 1.5V \times 1.05 \div 5V = 323 = \$143 \\ \text{下限} &: 1024 \times 1.5V \times 0.95 \div 5V = 291 = \$123 \end{aligned}$$

表105.の手順内で他の値が与えられる以外は表104.の推奨値が使われます。(表105.には)走査チェーンのDAC値とポートピン値だけが示されます。「動作」欄は続く行内の値で境界走査チェーンレジスタを満たす前にどのJTAG命令を使うべきかを記述します。表の同一列上のデータを走査入力する時に走査出力のデータで検証が行われるべきです。

表105. A/D変換使用手順

| 手順番号 | 動作                | ADGEN | DAC_9~0 | MUXEN_7~0 | HOLD | PRECH | PF3データ | PF3制御 | PF3プルアップ許可 |
|------|-------------------|-------|---------|-----------|------|-------|--------|-------|------------|
| 1    | SAMPLE_PRELOAD    | 1     | \$200   | \$08      | 1    | 1     | 0      | 0     | 0          |
| 2    | EXTEST            | 1     | \$200   | \$08      | 0    | 1     | 0      | 0     | 0          |
| 3    |                   | 1     | \$200   | \$08      | 1    | 1     | 0      | 0     | 0          |
| 4    |                   | 1     | \$123   | \$08      | 1    | 1     | 0      | 0     | 0          |
| 5    |                   | 1     | \$123   | \$08      | 1    | 0     | 0      | 0     | 0          |
| 6    | 0となるべきCOMP走査出力の検証 | 1     | \$200   | \$08      | 1    | 1     | 0      | 0     | 0          |
| 7    |                   | 1     | \$200   | \$08      | 0    | 1     | 0      | 0     | 0          |
| 8    |                   | 1     | \$200   | \$08      | 1    | 1     | 0      | 0     | 0          |
| 9    |                   | 1     | \$143   | \$08      | 1    | 1     | 0      | 0     | 0          |
| 10   |                   | 1     | \$143   | \$08      | 1    | 0     | 0      | 0     | 0          |
| 11   | 1となるべきCOMP走査出力の検証 | 1     | \$200   | \$08      | 1    | 1     | 0      | 0     | 0          |

この手法の使用ではHOLD信号の束縛タイミングがTCKクロック周波数を押さえます。手順の第5段階でのHOLDのHigh(1)に保持で、TCKクロック周波数は最大保持時間(thold\_max)で割られた走査ビット数の最低5倍でなければなりません。

## ATmega128の境界走査(Boundary-Scan)順

表106.は境界走査チェーンがデータ経路として選ばれる時のTDIとTDO間の走査順を示します。ビット0が最下位ビットで走査の最初に入出力されます。走査順は可能な限りピン配置順に従います。このためにポートAのビットは他のポートと違うビット順で走査されます。この規則の例外はアナログ回路が接続される物理ピンに関係なく走査チェーンの最上位を構成するアナログ回路用走査チェーンです。図124.上で、Pxn出力はFF0、Pxn制御はFF1、Pxnプルアップ許可はFF2に各々対応します。ポートFのビット4,5,6,7はJTAG許可時にTAPピンを構成するので走査チェーンには存在しません。

表106. ATmega128境界走査(Boundary-Scan)順

| ビット | 信号名                 | 基本単位                | ビット               | 信号名               | 基本単位              | ビット               | 信号名               | 基本単位                         |      |
|-----|---------------------|---------------------|-------------------|-------------------|-------------------|-------------------|-------------------|------------------------------|------|
| 204 | AC_IDLE             | アナログ<br>比較器         | 152               | PE2.Control       | ポートE              | 99                | RCOSCEN           | 主クロック/発振器<br>用許可信号           |      |
| 203 | ACO                 |                     | 151               | PE2.Pullup_Enable |                   | 98                | OSC32EN           |                              |      |
| 202 | ACME                |                     | 150               | PE3.Data          |                   | 97                | EXTCLK(XTAL1)     | 主クロック用<br>入力/発振器<br>(監視専用セル) |      |
| 201 | ACBG                |                     | 149               | PE3.Control       |                   | 96                | OSCK              |                              |      |
| 200 | COMP                |                     | 148               | PE3.Pullup_Enable |                   | 95                | RCCK              |                              |      |
| 199 | PRIVATE_SIGNAL1 (注) |                     | A/D変換             | 147               |                   | PE4.Data          | 94                | OSC32CK                      | TWI  |
| 198 | ACLK                |                     |                   | 146               |                   | PE4.Control       | 93                | TWIEN                        |      |
| 197 | ACTEN               |                     |                   | 145               |                   | PE4.Pullup_Enable | 92                | PD0.Data                     | ポートD |
| 196 | PRIVATE_SIGNAL2 (注) |                     |                   | 144               |                   | PE5.Data          | 91                | PD0.Control                  |      |
| 195 | ADCBGEN             |                     |                   | 143               |                   | PE5.Control       | 90                | PD0.Pullup_Enable            |      |
| 194 | ADCEN               | 142                 |                   | PE5.Pullup_Enable |                   | 89                | PD1.Data          |                              |      |
| 193 | AMPEN               | 141                 |                   | PE6.Data          |                   | 88                | PD1.Control       |                              |      |
| 192 | DAC_9               | 140                 |                   | PE6.Control       |                   | 87                | PD1.Pullup_Enable |                              |      |
| 191 | DAC_8               | 139                 |                   | PE6.Pullup_Enable |                   | 86                | PD2.Data          |                              |      |
| 190 | DAC_7               | 138                 |                   | PE7.Data          | 85                | PD2.Control       |                   |                              |      |
| 189 | DAC_6               | 137                 | PE7.Control       | 84                | PD2.Pullup_Enable |                   |                   |                              |      |
| 188 | DAC_5               | 136                 | PE7.Pullup_Enable | 83                | PD3.Data          |                   |                   |                              |      |
| 187 | DAC_4               | 135                 | PB0.Data          | 82                | PD3.Control       |                   |                   |                              |      |
| 186 | DAC_3               | 134                 | PB0.Control       | 81                | PD3.Pullup_Enable |                   |                   |                              |      |
| 185 | DAC_2               | 133                 | PB0.Pullup_Enable | 80                | PD4.Data          |                   |                   |                              |      |
| 184 | DAC_1               | 132                 | PB1.Data          | 79                | PD4.Control       |                   |                   |                              |      |
| 183 | DAC_0               | 131                 | PB1.Control       | 78                | PD4.Pullup_Enable |                   |                   |                              |      |
| 182 | EXTCH               | 130                 | PB1.Pullup_Enable | 77                | PD5.Data          |                   |                   |                              |      |
| 181 | G10                 | 129                 | PB2.Data          | 76                | PD5.Control       |                   |                   |                              |      |
| 180 | G20                 | 128                 | PB2.Control       | 75                | PD5.Pullup_Enable |                   |                   |                              |      |
| 179 | GNDEN               | 127                 | PB2.Pullup_Enable | 74                | PD6.Data          |                   |                   |                              |      |
| 178 | HOLD                | 126                 | PB3.Data          | 73                | PD6.Control       |                   |                   |                              |      |
| 177 | IREFEN              | 125                 | PB3.Control       | 72                | PD6.Pullup_Enable |                   |                   |                              |      |
| 176 | MUXEN_7             | 124                 | PB3.Pullup_Enable | 71                | PD7.Data          |                   |                   |                              |      |
| 175 | MUXEN_6             | 123                 | PB4.Data          | 70                | PD7.Control       |                   |                   |                              |      |
| 174 | MUXEN_5             | 122                 | PB4.Control       | 69                | PD7.Pullup_Enable |                   |                   |                              |      |
| 173 | MUXEN_4             | 121                 | PB4.Pullup_Enable | 68                | PG0.Data          | ポートG              |                   |                              |      |
| 172 | MUXEN_3             | 120                 | PB5.Data          | 67                | PG0.Control       |                   |                   |                              |      |
| 171 | MUXEN_2             | 119                 | PB5.Control       | 66                | PG0.Pullup_Enable |                   |                   |                              |      |
| 170 | MUXEN_1             | 118                 | PB5.Pullup_Enable | 65                | PG1.Data          |                   |                   |                              |      |
| 169 | MUXEN_0             | 117                 | PB6.Data          | 64                | PG1.Control       |                   |                   |                              |      |
| 168 | NEGSEL_2            | 116                 | PB6.Control       | 63                | PG1.Pullup_Enable |                   |                   |                              |      |
| 167 | NEGSEL_1            | 115                 | PB6.Pullup_Enable | 62                | PC0.Data          | ポートC              |                   |                              |      |
| 166 | NEGSEL_0            | 114                 | PB7.Data          | 61                | PC0.Control       |                   |                   |                              |      |
| 165 | PASSEN              | 113                 | PB7.Control       | 60                | PC0.Pullup_Enable |                   |                   |                              |      |
| 164 | PRECH               | 112                 | PB7.Pullup_Enable | 59                | PC1.Data          |                   |                   |                              |      |
| 163 | SCTEST              | 111                 | PG3.Data          | 58                | PC1.Control       |                   |                   |                              |      |
| 162 | ST                  | 110                 | PG3.Control       | 57                | PC1.Pullup_Enable |                   |                   |                              |      |
| 161 | VCCREN              | 109                 | PG3.Pullup_Enable | 56                | PC2.Data          |                   |                   |                              |      |
| 160 | PEN                 | プログラム許可<br>(監視専用セル) | 108               | PG4.Data          | 55                |                   | PC2.Control       |                              |      |
| 159 | PE0.Data            | ポートE                | 107               | PG4.Control       | 54                |                   | PC2.Pullup_Enable |                              |      |
| 158 | PE0.Control         |                     | 106               | PG4.Pullup_Enable | 53                |                   | PC3.Data          |                              |      |
| 157 | PE0.Pullup_Enable   |                     | 105               | TOSC              | 52                |                   | PC3.Control       |                              |      |
| 156 | PE1.Data            |                     | 104               | TOSCON            | 51                |                   | PC3.Pullup_Enable |                              |      |
| 155 | PE1.Control         |                     | 103               | RSTT              | 50                |                   | PC4.Data          |                              |      |
| 154 | PE1.Pullup_Enable   |                     | 102               | RSTHV             | 49                |                   | PC4.Control       |                              |      |
| 153 | PE2.Data            | 101                 | EXTCLKEN          | 48                | PC4.Pullup_Enable |                   |                   |                              |      |
|     |                     |                     | 100               | OSCON             | 47                | PC5.Data          |                   |                              |      |

注: PRIVATE\_SIGNAL1とPRIVATE\_SIGNAL2は常に0として走査されるべきです。 [次頁](#)へ続く



表106 (続き). ATmega128境界走査(Boundary-Scan)順

| ビット | 信号名               | 基本単位 | ビット               | 信号名               | 基本単位 | ビット               | 信号名               | 基本単位 |
|-----|-------------------|------|-------------------|-------------------|------|-------------------|-------------------|------|
| 46  | PC5.Control       | ポートC | 30                | PA6.Pullup_Enable | ポートA | 14                | PA0.Data          | ポートA |
| 45  | PC5.Pullup_Enable |      | 29                | PA5.Data          |      | 13                | PA0.Control       |      |
| 44  | PC6.Data          |      | 28                | PA5.Control       |      | 12                | PA0.Pullup_Enable |      |
| 43  | PC6.Control       |      | 27                | PA5.Pullup_Enable |      | 11                | PF3.Data          | ポートF |
| 42  | PC6.Pullup_Enable |      | 26                | PA4.Data          |      | 10                | PF3.Control       |      |
| 41  | PC7.Data          |      | 25                | PA4.Control       |      | 9                 | PF3.Pullup_Enable |      |
| 40  | PC7.Control       |      | 24                | PA4.Pullup_Enable |      | 8                 | PF2.Data          |      |
| 39  | PC7.Pullup_Enable | 23   | PA3.Data          | 7                 |      | PF2.Control       |                   |      |
| 38  | PG2.Data          | 22   | PA3.Control       | 6                 |      | PF2.Pullup_Enable |                   |      |
| 37  | PG2.Control       | 21   | PA3.Pullup_Enable | 5                 |      | PF1.Data          |                   |      |
| 36  | PG2.Pullup_Enable | 20   | PA2.Data          | 4                 |      | PF1.Control       |                   |      |
| 35  | PA7.Data          | 19   | PA2.Control       | 3                 |      | PF1.Pullup_Enable |                   |      |
| 34  | PA7.Control       | 18   | PA2.Pullup_Enable | 2                 |      | PF0.Data          |                   |      |
| 33  | PA7.Pullup_Enable | 17   | PA1.Data          | 1                 |      | PF0.Control       |                   |      |
| 32  | PA6.Data          | 16   | PA1.Control       | 0                 |      | PF0.Pullup_Enable |                   |      |
| 31  | PA6.Control       | 15   | PA1.Pullup_Enable |                   |      |                   |                   |      |

### 境界走査記述言語(Boundary-Scan Description Language)ファイル

境界走査記述言語(BSDL)ファイルは境界走査が可能なデバイスを自動検査生成ソフトウェアによって使われる標準形式で記述します。境界走査データレジスタ内のビットの順番と機能がこの記述内に含まれます。

## ブートローダ支援 – 書き込み中読み出し可能な自己プログラミング

ブートローダ支援はMCU自身によるプログラムコードのダウンロードとアップロード用の真の書き込み中の読み出しが可能な自己プログラミング機構を提供します。この特徴はフラッシュメモリに常駐するブートローダプログラムを使うMCUによって制御される柔軟な応用ソフトウェア更新を可能にします。ブートローダプログラムはフラッシュメモリ内にコードを書き(プログラム)、コードを読み、またはプログラムメモリからコードを読むのに、利用可能なデータインターフェースと関連する規約のどれもが使えます。ブートローダ領域内のプログラムコードはブートローダメモリを含むフラッシュメモリ全体を書く能力を持ちます。従ってブートローダは自身をも変更でき、この機能がそれ以上必要とされないなら、そのコードから自身を消去することもできます。ブートローダメモリの容量はヒューズで設定可能で、ブートローダは個別に設定可能な2組の独立したブート施錠ビットを持ちます。これは異なる保護基準を選ぶ独特な柔軟性を使用者に与えます。

### 特徴

- 書き込み中読める(Read-While-Write)自己プログラミング
- 柔軟性のあるブートローダメモリ容量
- 高い安全性 (柔軟な保護用の独立したブート施錠ビット)
- リセットベクタ選択用の独立したヒューズ
- 最適化されたページ容量 (注1)
- 効率的なコード手法
- 効率的な読み-変更-書き(リード モデファイライト)支援

**注1:** ページはプログラミング中に使われる多数のバイトから成るフラッシュメモリの区画です(189頁の表124参照)。このページ構成は通常動作に影響を及ぼしません。

### フラッシュメモリの応用領域とブートローダ領域

フラッシュメモリは応用領域とブートローダ領域の2つの主な領域で構成されます(図133参照)。各領域の容量は186頁の表112と図133で示されるようにBOOTSZヒューズによって設定されます。これら2つの領域は個別の施錠ビットの組を持つため、異なる保護基準を持てます。

#### 応用領域

応用領域は応用コードを格納するのに使われるフラッシュメモリの領域です。応用領域用保護基準は応用ブート施錠ビット(ブート施錠ビット0)によって選べます(180頁の表108参照)。応用領域から実行される時にSPM命令が禁止されるので、応用領域はどんなブートローダコードも決して格納し得ません。

#### ブートローダ領域 (BLS)

応用領域が応用コード格納用に使われるのに対して、SPM命令はBLSから実行する時にだけプログラミングを始められるので、ブートローダソフトウェアはBLSに格納されなければなりません。SPM命令はBLS自身を含む全てのフラッシュメモリをアクセスできます。ブートローダ領域用保護基準はブートローダ施錠ビット(ブート施錠ビット1)によって選べます(180頁の表109参照)。

### フラッシュメモリの書き込み中に読み出し可能な領域と不能な領域

どちらのアドレスがプログラミングされるかによって、CPUが書き込み中の読み出しを支援するか、ブートローダソフトウェアが更新中にCPUが停止されるかのどちらです。上で記述されるようなBOOTSZヒューズによって設定可能な2つの領域に加え、フラッシュメモリは書き込み中読み出し可能な(RWW)領域と書き込み中読み出し不能な(NRWW)領域の2つの固定領域にも分けられます。RWWとNRWW領域間の境界は179頁の図133と186頁の表113で与えられます。この2つの領域間の主な違いを次に示します。

- RWW領域側に配置されたページを消去または書くとき、NRWW領域はその動作中に読むことができます。
- NRWW領域側に配置されたページを消去または書くとき、その全ての動作中にCPUは停止されます。

ブートローダソフトウェア動作中、使用者ソフトウェアがRWW領域側に配置されるどのコードも決して読めないことに注意してください。「書き込み中読み出し可能領域」という記述はプログラミング(消去または書き込み)される領域としての引用で、ブートローダソフトウェアが更新中に実際に読まれる領域ではありません。

**訳補:** 上の記述はNRWW領域からRWW領域をプログラミングするという前提で、消去または書き込みを行う側ではなく、行われる側でこの名称が定義されていることを意味します。即ち、NRWW領域からRWW領域をプログラミングすると、NRWW領域のプログラムは通常通り動作する(即ち読める)ので、プログラミングされる側はRWW領域と名付けられ、この逆ではCPUが停止する(即ち読めない)ので、NRWW領域と名付けられているという意味です。

## RWW - 書き込み中読み出し可能領域

ブートローダソフトウェア更新がRWW領域側のページをプログラミングする場合、フラッシュメモリからコードを読むことが可能ですが、NRWW領域に配置されるコードだけです。プログラミング実行中、そのソフトウェアはRWW領域が決して読まれないことを保証しなければなりません。使用者ソフトウェアがプログラミング中に(例えば、CALL, JMP, LPM系命令または割り込みによって)RWW領域側に配置されるコードを読むと、そのソフトウェアは未知の状態へ行き着くかもしれません。これを避けるために割り込みは禁止、またはブートローダ領域へ移動のどちらかにされるべきです。ブートローダ領域は常にNRWW領域に配置されます。RWW領域が読み出しに対して妨げられている限り、SPM命令制御レジスタ(SPMCR)のRWW領域多忙(RWWSB)ビットは論理1として読みます。プログラミングが完了した後、RWW領域に配置したコードを読む前にRWWSBはソフトウェアによって解除(0)されなければなりません。RWWSBを解除(0)する方法の詳細については181頁の「SPM命令制御/状態レジスタ(SPMCSR)」をご覧ください。

## NRWW - 書き込み中読み出し不能領域

NRWW領域に配置したコードはブートローダソフトウェアがRWW領域内のページを更新する時に読めます。ブートローダコードがNRWW領域を更新するとき、全てのページ消去またはページ書き込み動作中にCPUが停止されます。

表107. 書き込み中読み出し可能機能

| プログラミング中にZポイントで指定される領域 | プログラミング中に読める領域 | CPU動作 | RWW機能支援 |
|------------------------|----------------|-------|---------|
| RWW領域                  | NRWW領域         | 通常動作  | あり      |
| NRWW領域                 | なし             | 停止    | なし      |

図132. RWW領域とNRWW領域の関係

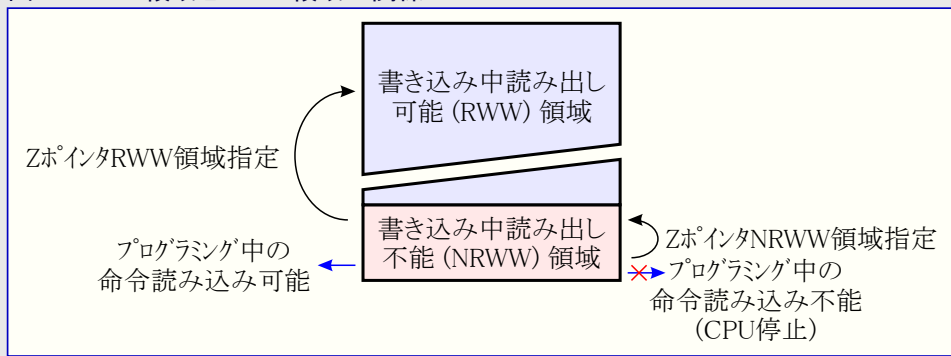
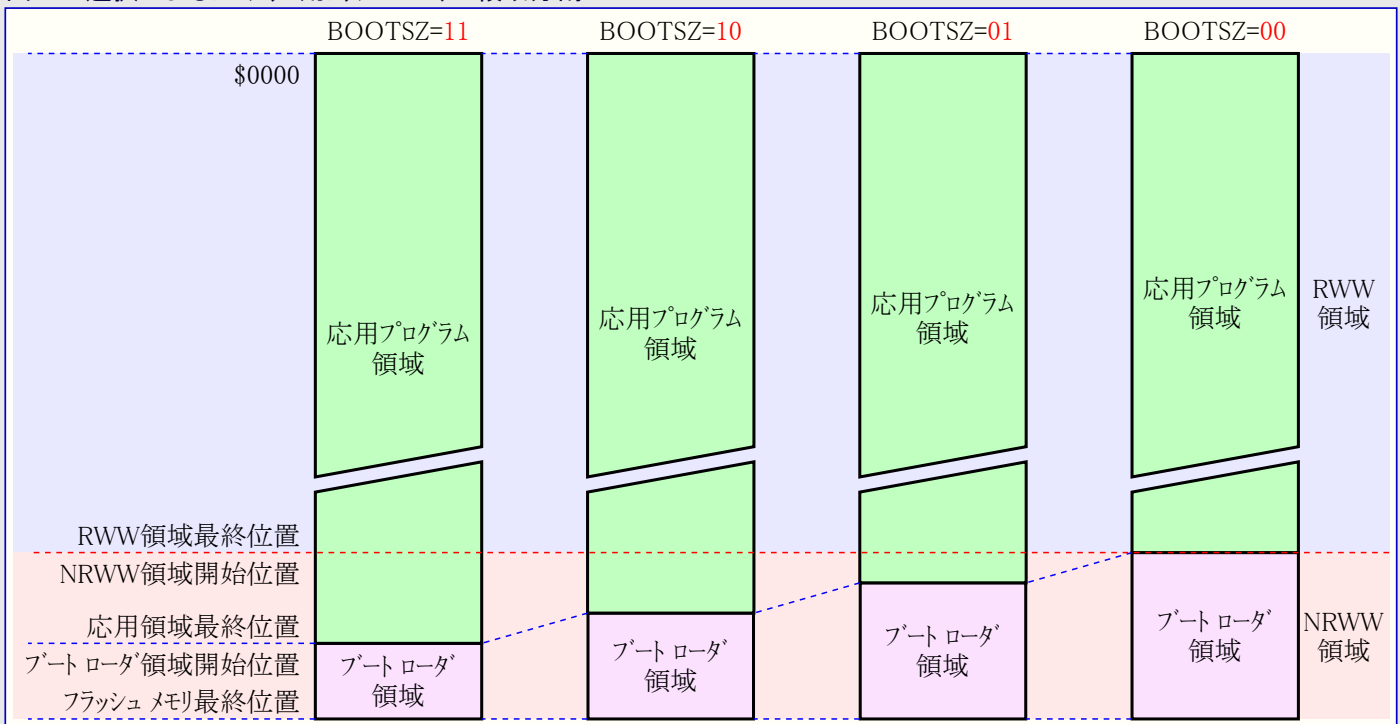


図133. 選択によるプログラム用フラッシュメモリの領域分割



注: 図内の各パラメータについては186頁の表112.で与えられます。

## ブートローダ施錠ビット

ブートローダ能力が必要とされないなら、フラッシュメモリ全体が応用コード用に利用可能です。ブートローダは個別に設定可能な独立した2組のブート施錠ビットを持ちます。これは異なる保護基準を選ぶ独特な柔軟性を使用者に与えます。

使用者は以下を選べます。

- MCUによって更新するソフトウェアからフラッシュメモリ全体を保護
- MCUによって更新するソフトウェアからフラッシュメモリのブートローダ領域だけを保護
- MCUによって更新するソフトウェアからフラッシュメモリの応用領域だけを保護
- フラッシュメモリ全体で更新するソフトウェアを許可

より多くの詳細については表108と表109をご覧ください。ブート施錠ビットはソフトウェアと直列または並列のプログラミング動作で設定(0)できますが、これらのビットはチップ消去指令によってのみ解除(1)できます。一般書き込み禁止(LB動作種別2)はSPM命令によるフラッシュメモリのプログラミングを制御しません。同様に、試みられたなら、一般読み書き禁止(LB動作種別3)はLPM命令とSPM命令による読み込みも書き込みも制御しません。(訳補:一般LBはLPM/SPM命令に関して無関係の意)

表108. 応用領域に対する保護種別 (0=プログラム、1=非プログラム)

| BLB0 動作種別 | BLB02 | BLB01 | 保護種別   |
|-----------|-------|-------|--|
| 1         | 1     | 1     | LPM, SPM命令が応用領域をアクセスすることに対して制限はありません。                        |
| 2         | 1     | 0     | SPM命令は応用領域に書くことを許されません。                                      |
| 3         | 0     | 0     | SPM命令による応用領域への書き込みと、ブートローダ領域でのLPM命令による応用領域からの読み込みが許されません。(注) |
| 4         | 0     | 1     | ブートローダ領域でのLPM命令による応用領域からの読み込みが許されません。(注)                     |

注: BLB02=0で、割り込みベクタがブートローダ領域に配置されていると、応用領域での実行時に割り込みが禁止されます。

表109. ブートローダ領域に対する保護種別 (0=プログラム、1=非プログラム)

| BLB1 動作種別 | BLB12 | BLB11 | 保護種別   |
|-----------|-------|-------|--|
| 1         | 1     | 1     | LPM, SPM命令がブートローダ領域をアクセスすることに対して制限はありません。                        |
| 2         | 1     | 0     | SPM命令はブートローダ領域に書くことを許されません。                                      |
| 3         | 0     | 0     | SPM命令によるブートローダ領域への書き込みと、応用領域でのLPM命令によるブートローダ領域からの読み込みが許されません。(注) |
| 4         | 0     | 1     | 応用領域でのLPM命令によるブートローダ領域からの読み込みが許されません。(注)                         |

注: BLB12=0で、割り込みベクタが応用領域に配置されていると、ブートローダ領域での実行時に割り込みが禁止されます。

## ブートローダプログラムへの移行

ブートローダへの移行は応用プログラムから分岐(Jump)または呼び出し(Call)によって行います。これはUSARTやSPIインターフェース経由で受信した指令のような起点によって始められるかもしれませんが、代わりに、リセット後にリセットベクタがブートローダ領域開始アドレスを指示するようにブートリセット(BOOTRST)ヒューズをプログラム(0)できます。この場合、ブートローダがリセット後に開始されます。応用コードが設定された(書かれた)後、そのプログラム(ブートローダ)は応用コードの実行を始めることができます。このヒューズがMCU自身によって変更できないことに注意してください。これは一旦ブートリセットヒューズがプログラム(0)されると、リセットベクタは常にブートローダリセットを指示し、このヒューズが直列プログラミング、並列プログラミングまたはJTAGインターフェースを通してのみ変更できることを意味します。

表110. ブートリセットヒューズ (0=プログラム、1=非プログラム)

| BOOTRST | リセット後実行開始アドレス (リセットベクタ) |                           |
|---------|-------------------------|---------------------------|
| 0       | ブートローダリセット              | ブートローダ開始アドレス(186頁の表112参照) |
| 1       | 応用リセット                  | \$0000                    |

## SPM命令制御/状態レジスタ (Store Program Memory Control and Status Register) SPMCSR

このレジスタはブートローダ動作を制御するために必要とされる制御ビットを含みます。

| ビット<br>(\$68) | 7     | 6     | 5 | 4      | 3      | 2     | 1     | 0     | SPMCSR |
|---------------|-------|-------|---|--------|--------|-------|-------|-------|--------|
|               | SPMIE | RWWSB | – | RWWSRE | BLBSET | PGWRT | PGERS | SPMEN |        |
| Read/Write    | R/W   | R     | R | R/W    | R/W    | R/W   | R/W   | R/W   |        |
| 初期値           | 0     | 0     | 0 | 0      | 0      | 0     | 0     | 0     |        |

• **ビット7 – SPMIE : SPM操作可割り込み許可 (SPM Interrupt Enable)**

SPMIE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(0)ビット**が設定(**1**)されているなら、SPM操作可割り込みが許可されます。SPM操作可割り込みはSPMCSRのSPM操作許可(SPMEN)ビットが解除(**0**)されている限り実行されます。

• **ビット6 – RWWSB : RWW領域多忙 (Read-While-Write Section Busy)**

RWW領域に自己プログラミング(ページ消去またはページ書き込み)操作が開始されると、RWWSBがハードウェアによって設定(**1**)されます。RWWSBビットが設定(**1**)されると、RWW領域はアクセスできません。自己プログラミングが完了された後にRWWSREビットが**1**を書かれると、このRWWSBビットは解除(**0**)されます。同様に、ページ設定操作が開始されると、RWWSBビットは自動的に解除(**0**)されます。

• **ビット5 – Res : 予約 (Reserved)**

このビットは予約されており、常に0として読みます。

• **ビット4 – RWWSRE : RWW領域読み出し許可 (Read-While-Write Section Read Enable)**

RWW領域にプログラミング(ページ消去またはページ書き込み)すると、RWW領域は(RWWSBがハードウェアによって設定(**1**)され)読み出しに対して防がれます。RWW領域を再許可するために使用者ソフトウェアはプログラミングが完了(SPMENが解除(**0**))されるまで待たねばなりません。その後、RWWSREビットがSPMENと同時に**1**を書かれると、次の4クロック周期内の**SPM**命令がRWW領域を再許可します。ページ消去またはページ書き込みでフラッシュメモリが多忙(SPMEN=**1**)の間、RWW領域は再許可できません。フラッシュメモリが設定されている間にRWWSREビットが書かれると、フラッシュメモリ設定操作は失敗し、設定したデータが失われます。

• **ビット3 – BLBSET : ブート施錠ビット設定 (Boot Lock Bits Set)**

このビットがSPMENと同時に**1**を書かれると、次の4クロック周期内の**SPM**命令はR0のデータに従ったブート施錠ビットを設定します。R1のデータとZポインタのアドレスは無視されます。BLBSETビットは施錠ビットの設定完了で、または4クロック周期内に**SPM**命令が実行されない場合、自動的に解除(**0**)されます。

SPMCSRでBLBSETとSPMENが設定(**1**)された後の3クロック周期内の**LPM**命令は(ZポインタのZ0によって)ヒューズビットまたは施錠ビットのどちらかを転送先レジスタに読みます。詳細については184頁の「ソフトウェアからのヒューズビットと施錠ビットの読み出し」をご覧ください。

• **ビット2 – PGWRT : ページ書き込み (Page Write)**

このビットがSPMENと同時に**1**を書かれると、次の4クロック周期内の**SPM**命令は一時緩衝部に格納したデータでページ書き込みを実行します。ページアドレスはZポインタの上位部から取得されます。R1とR0のデータは無視されます。PGWRTビットはページ書き込みの完了で、または4クロック周期内に**SPM**命令が実行されない場合、自動的に解除(**0**)されます。NRWW領域がアドレス指定されると、ページ全体の書き込み動作中にCPUは停止されます。

• **ビット1 – PGERE : ページ消去 (Page Erase)**

このビットがSPMENと同時に**1**を書かれると、次の4クロック周期内の**SPM**命令はページ消去を実行します。ページアドレスはZポインタの上位部から取得されます。R1とR0のデータは無視されます。PGEREビットはページ消去の完了で、または4クロック周期内に**SPM**命令が実行されない場合、自動的に解除(**0**)されます。NRWW領域がアドレス指定されると、ページ全体の消去中にCPUは停止されます。

• **ビット0 – SPMEN : SPM操作許可 (Store Program Memory Enable)**

このビットは次の4クロック周期間の**SPM**命令を許可します。このビットがRWWSRE、BLBSET、PGWRT、PGEREのどれかと共に**1**を書かれると、続く**SPM**命令は特別な意味を持ちます(上の記述をご覧ください)。SPMENだけが書かれると、続く**SPM**命令はZポインタによってアドレス指定したページ一時緩衝部へR1:R0の値を格納します。Zレジスタの最下位ビットは無視されます。SPMENビットは**SPM**命令の完了で、または4クロック周期内に**SPM**命令が実行されない場合、自動的に解除(**0**)されます。ページ消去とページ書き込み中、SPMENビットはその動作が完了されるまで**1**に留まります。

下位5ビットに**10001**、**01001**、**00101**、**00011**、**00001**以外のどんな組み合わせを書いても無効です。



## 自己プログラミング中のフラッシュメモリのアドレス指定

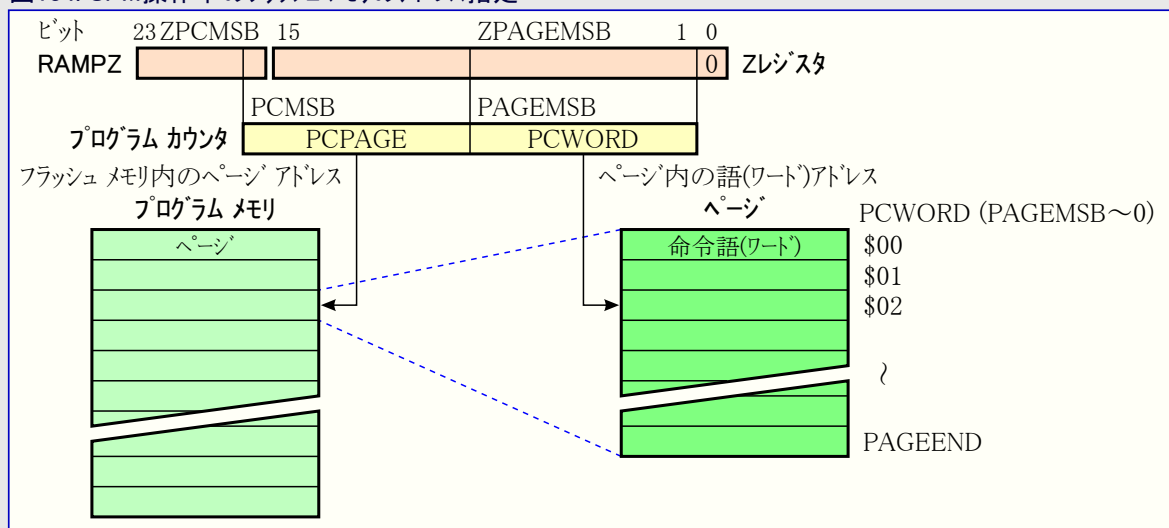
RAMPZと共にZポインタ(レジスタ)がSPM命令でのアドレス指定に使われます。RAMPZ使用法の詳細については10頁の「RAMページ選択レジスタ(RAMPZ)」をご覧ください。

|          |     |     |     |     |     |     |    |    |
|----------|-----|-----|-----|-----|-----|-----|----|----|
| ビット      | 15  | 14  | 13  | 12  | 11  | 10  | 9  | 8  |
| ZH (R31) | Z15 | Z14 | Z13 | Z12 | Z11 | Z10 | Z9 | Z8 |
| ビット      | 7   | 6   | 5   | 4   | 3   | 2   | 1  | 0  |
| ZL (R30) | Z7  | Z6  | Z5  | Z4  | Z3  | Z2  | Z1 | Z0 |

フラッシュメモリがページで構成されるため(189頁の表124参照)、プログラムカウンタ(アドレスポインタ)は2つの違う領域を持つように取り扱われます。1つの領域は下位側ビットから成り、ページ内の語(ワード)をアドレス指定し、一方上位側ビットはそのページをアドレス指定します。これは図134で示されます。ページ消去とページ書き込み操作が個別にアドレス指定されることに注意してください。従ってブートローダソフトウェアはページ消去とページ書き込み操作の両方で同じページをアドレス指定することが最も重要です。一旦プログラミング操作が開始されると、このアドレスはラッチされ、RAMPZ/Zポインタは他の操作に使えます。

RAMPZ/Zポインタを使わないSPM操作はブートローダ施錠ビット設定だけです。この操作でRAMPZ/Zレジスタの内容は無視され、無効です。(ELPM命令もアドレスを格納するのにRAMPZ/Zポインタを使います。この命令はフラッシュメモリのバイト単位をアドレス指定するのでZポインタの最下位ビット(Z0)も使われます。

図134. SPM操作中のフラッシュメモリのアドレス指定



注: 図内で使った各変数は186頁の表114で一覧されます。  
PCPAGEとPCWORDは189頁の表124で一覧されます。

## フラッシュメモリの自己プログラミング

プログラムメモリはページ単位形式で更新されます。ページ一時緩衝部へ格納したデータでページを書く前にそのページが消去されなければなりません。ページ一時緩衝部はSPM命令使用時毎の1語(ワード)で満たされ、この緩衝部はページ消去命令前、またはページ消去とページ書き込み操作間のどちらかで満たすことができます。

### 手段1 (ページ消去前の一時緩衝部格納)

- ページ一時緩衝部を満たしてください。
- ページ消去を実行してください。
- ページ書き込みを実行してください。

### 手段2 (ページ消去後の一時緩衝部格納)

- ページ消去を実行してください。
- ページ一時緩衝部を満たしてください。
- ページ書き込みを実行してください。

ページの一部の変更だけが必要な場合、消去前にページの残す部分は(例えばページ一時緩衝部に)保存されなければならない、その後改めて書かれます。手段1を使う場合、初めにページを読んで必要な変更を行い、その後に変更したデータを書き戻すことをユーザーソフトウェアに許す効率的な読み-修正-書き(リード-モディファイ-ライト)機能をブートローダが提供します。手段2が使われる場合、ページが既に消去されているため、格納中の旧データを読むことができません。ページ一時緩衝部は乱順でアクセスできます。ページ消去とページ書き込み操作の両方で使われるページアドレスは同じページをアドレス指定することが非常に重要です。アセンブリ言語でのコード例については185頁の「アセンブリ言語による簡単なブートローダ例」をご覧ください。

## SPM命令によるページ消去の実行

ページ消去を実行するにはRAMPZとZポイントにアドレスを設定してSPM命令制御/状態レジスタ(SPMCSR)に'X0000011'を書き、SPMCSR書き込み後4クロック周期内にSPM命令を実行してください。R1とR0のデータは無視されます。ページアドレスはRAMPZとZポイントのPCPAGEに書かれなければなりません。この操作中、Zポイントの他のビットは0を書かれなければなりません。

- RWW領域のページ消去 : ページ消去中、NRWW領域は読めます。
- NRWW領域のページ消去 : ページ消去中、CPUは停止されます。

## ページ一時緩衝部の設定 (ページ設定)

命令語(ワード)を(ページ一時緩衝部に)書くにはZポイントにアドレス、R1:R0にデータを設定してSPMCSRに'00000001'を書き、SPMCSR書き込み後4クロック周期内にSPM命令を実行してください。ZポイントのPCWORDの内容は一時緩衝部のデータのアドレスに使われます。一時緩衝部はページ書き込み操作後、またはSPMCSRのRWWSREビット書き込みによって自動的に消去されます。システムリセット後も消去されています。一時緩衝部を消去せずに各アドレスへ複数回書くことができないことに注意してください。

**注:** SPMページ設定操作の途中でEEPROMが書かれると、設定した全データが失われます。

## ページ書き込みの実行

ページ書き込みを行うにはRAMPZとZポイントにアドレスを設定してSPMCSRに'X0000101'を書き、SPMCSR書き込み後4クロック周期内にSPM命令を実行してください。R1とR0のデータは無視されます。ページアドレスは(RAMPZとZポイントの)PCPAGEに書かれなければなりません。この操作中、Zポイントの他のビットは0を書かれなければなりません。

- RWW領域のページ書き込み : ページ書き込み中、NRWW領域は読めます。
- NRWW領域のページ書き込み : ページ書き込み中、CPUは停止されます。

## SPM操作可割り込みの使用法

SPM操作可割り込みが許可されると、SPMCSRのSPMENビットが解除(0)されている時にSPM操作可割り込みが継続的に発生します。これはソフトウェアでSPMCSRをポーリングする代わりにこの割り込みが使えることを意味します。SPM操作可割り込みを使うとき、割り込みが読み出しに対して防がれる時にRWW領域をアクセスするのを避けるために、割り込みベクタはブートローダ領域(BLS)へ移動されるべきです。割り込み(ベクタ)の移動法は37頁の「割り込み」で記述されます。

## ブートローダ領域更新中の考慮

ブート施錠ビット11(BLB11)が非プログラム(1)にされたままとすることによって使用者がブートローダ領域に更新を許す場合、特別な注意が祓われなければなりません。ブートローダ自身への予期せぬ書き込みはブートローダ全体を不正にし得て、更にソフトウェアの更新が不可能になるかもしれません。ブートローダ自体の変更が必要ないなら、内部ソフトウェアのどんな変更からもブートローダを保護するためにブート施錠ビット11(BLB11)をプログラム(0)することが推奨されます。

## 自己プログラミング中のRWW領域読み込みの防止

自己プログラミング中(ページ消去もページ書き込みも)、RWW領域は読み出しに対して常に防がれます。使用者ソフトウェアそれ自身が自己プログラミング操作中にこの領域がアドレス指定されるのを防止しなければなりません。SPMCSRのRWWSBはRWW領域が多忙である限り設定(1)されます。自己プログラミング中の割り込みベクタ表は37頁の「割り込み」で記述されるようにブートローダ領域(BLS)へ移動されるべきか、または割り込みが禁止されなければなりません。プログラミングが完了した後にRWW領域をアドレス指定する前に、使用者ソフトウェアはRWWSREの書き込みによってRWWSBを解除(0)しなければなりません。例については185頁の「アセンブリ言語による簡単なブートローダ例」をご覧ください。

## SPM命令によるブートローダ施錠ビットの設定

ブートローダ施錠ビットを解除(0)するには希望したデータをR0に設定してSPMCSRに'X0001001'を書き、SPMCSR書き込み後4クロック周期内にSPM命令を実行してください。アクセス可能な施錠ビットはMCUによるどのソフトウェア更新からも応用とブートローダ領域を保護できるブート施錠ビットだけです。

| ビット | 7 | 6 | 5     | 4     | 3     | 2     | 1 | 0 |
|-----|---|---|-------|-------|-------|-------|---|---|
| R0  | 1 | 1 | BLB12 | BLB11 | BLB02 | BLB01 | 1 | 1 |

フラッシュメモリのアクセスに影響を及ぼすブートローダ施錠ビットの各種設定法については表108と表109をご覧ください。

R0のビット5~2が解除(0)される場合、SPMCSRでSPMENとブート施錠ビット設定(BLBSET)ビットが設定(1)された後4クロック周期内にSPM命令が実行されると、対応するブート施錠ビットがプログラム(0)されます。この操作中、Zポイントは関係ありませんが、将来との共通性のために(施錠ビット読み出しに使われるのと同じ)00001でZポイントを設定することが推奨されます。将来との共通性のため、施錠ビット書き込み時、R0のビット7,6,1,0は1に設定することも推奨されます。施錠ビットをプログラミングするとき、この操作中に全てのフラッシュメモリは読むことができます。

## SPM命令での書き込み時のEEPROM書き込みによる妨害

EEPROM書き込み動作がフラッシュメモリへの全ソフトウェアプログラミングを妨げることに注意してください。ソフトウェアからのヒューズと施錠ビット読み出しもEEPROM書き込み動作中、妨げられます。使用者はEEPROM制御レジスタ(EECR)のEEPROM書き込み許可(EWE)ビットを検査し、SPM命令制御/状態レジスタ(SPMCSR)へ書く前にこのビットが解除(0)されているのを確認することが推奨されます。

### ソフトウェアからのヒューズビットと施錠ビットの読み出し

ソフトウェアからヒューズと施錠ビットの両方を読むことが可能です。施錠ビットを読むにはZポイントに\$0001を設定してSPMCSRのSPMENとブート施錠ビット設定(BLBSET)ビットを設定(1)してください。SPMENとBLBSETビットがSPMCSRに設定された後、3 CPU周期内にLPM命令が実行されると、施錠ビットの値が転送先レジスタに格納されます。SPMENとBLBSETビットは施錠ビット読み出しの完了で、または3 CPU周期内にLPM命令が実行されないか、または4 CPU周期内にSPM命令が実行されない場合、自動的に解除(0)されます。SPMENとBLBSETビットが解除(0)されると、LPMは命令一式手引書で記述されるように動作します。

| ビット | 7 | 6 | 5     | 4     | 3     | 2     | 1   | 0   |
|-----|---|---|-------|-------|-------|-------|-----|-----|
| Rd  | - | - | BLB12 | BLB11 | BLB02 | BLB01 | LB2 | LB1 |

ヒューズ下位ビットを読む手順は上記の施錠ビット読み出しと同様です。ヒューズ下位ビットを読み出すにはZポイントに\$0000を設定してSPMCSRのSPMENとBLBSETビットを設定(1)してください。SPMENとBLBSETビットがSPMCSRに設定された後、3 CPU周期内にLPM命令が実行されると、以下で示されるようにヒューズ下位ビット(FLB)の値が転送先レジスタに格納されます。ヒューズ下位ビットの配置と詳細な記述については188頁の表119を参照してください。

| ビット | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
|-----|------|------|------|------|------|------|------|------|
| Rd  | FLB7 | FLB6 | FLB5 | FLB4 | FLB3 | FLB2 | FLB1 | FLB0 |

同様に、ヒューズ上位ビットを読むにはZポイントに\$0003を設定してください。SPMENとBLBSETビットがSPMCSRで設定(1)された後、3周期内にLPM命令が実行されると、以下で示されるようにヒューズ上位ビット(FHB)の値が転送先レジスタに格納されます。ヒューズ上位ビットの配置と詳細な記述については188頁の表118を参照してください。

| ビット | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
|-----|------|------|------|------|------|------|------|------|
| Rd  | FHB7 | FHB6 | FHB5 | FHB4 | FHB3 | FHB2 | FHB1 | FHB0 |

拡張ヒューズビットを読む時はZポイントに\$0002を設定してください。SPMENとBLBSETビットがSPMCSRで設定(1)された後、3周期内にLPM命令が実行されると、以下で示されるように拡張ヒューズビット(EFB)の値が転送先レジスタに格納されます。拡張ヒューズビットの配置と詳細な記述については188頁の表117を参照してください。

| ビット | 7 | 6 | 5 | 4 | 3 | 2 | 1    | 0    |
|-----|---|---|---|---|---|---|------|------|
| Rd  | - | - | - | - | - | - | EFB1 | EFB0 |

プログラム(0)されたヒューズと施錠ビットは0として読みます。非プログラム(1)にされたヒューズと施錠ビットは1として読みます。

### フラッシュメモリデータ化けの防止

低VCCの期間中、CPUとフラッシュメモリの正しい動作に対して供給電圧が低すぎるためにフラッシュメモリのプログラムが不正にされ得ます。これらの問題はフラッシュメモリを使う基板段階の装置と同じで、同じ設計上の解決策が適用されるべきです。

フラッシュメモリのプログラム化けは電圧が低すぎる時の2つの状態によって起こされます。1つ目としてフラッシュメモリへの通常の書き込み手順は正しく動作するための最低電圧が必要です。2つ目として供給電圧が低すぎると、CPU自身が命令を間違えて実行し得ます。

フラッシュメモリ化けは次の推奨設計によって容易に避けられます(1つは必須)。

- そのシステムでブートローダ更新が必要ない場合、どんなブートローダソフトウェア更新をも防ぐためにブートローダ施錠ビットをプログラム(0)してください。
- 不十分な供給電源電圧の期間中、AVR RESETを活性(Low)に保ってください。これは動作電圧が検出電圧と一致するなら、内部低電圧検出器(BOD)を許可することによって行えます。そうでなければ外部低VCCリセット保護回路が使えます。書き込み操作進行中にリセットが起こると、その書き込み動作は供給電源電圧が充分であれば完了されます。
- 低VCCの期間中、AVRコアをパワーダウン休止動作に保ってください。これはCPUが命令の復号と実行を試みるのを防ぎ、SPMCSR従ってフラッシュメモリを予期せぬ書き込みから効果的に保護します。

### SPM命令使用時のフラッシュメモリ用プログラミング(書き込み)時間

校正された内蔵RC発振器がフラッシュメモリアクセス時間に使われます。表111はCPUからのフラッシュメモリアクセスに対する代表的なプログラミング時間を示します。

表111. SPM命令によるフラッシュメモリのプログラミング時間

| 項目   | 最小    | 最大    |
|--|-------|-------|
| SPM命令によるフラッシュ書き込み(ページ消去、ページ書き込み、施錠ビット書き込み) | 3.7ms | 4.5ms |

## アセンブリ言語による簡単なブートローダ例

このルーチンはRAMからフラッシュメモリへ1ページのデータを書きます。RAM内の最初のデータ位置はYレジスタによって指示され、フラッシュメモリ内の最初のデータ位置はZレジスタによって指示されます。異常処理は含まれません。このルーチン(少なくともSPMJサブルーチン)はブートローダ領域側に配置されなければなりません。NRWW領域側のコードだけが自己プログラミング(ページ消去とページ書き込み)中に読めます。使用レジスタはR0,R1,TMP,CNTL,CNTH,SPMCで、レジスタの保存と復帰はこのルーチン内に含まれず、使用レジスタはコード量を犠牲にすれば最適化できます。割り込み表がブートローダ領域に移動されるか、割り込みが禁止されるかのどちらかが前提です。

ページ内データが256バイト以下の場合には計数器上位が不要になります。また関連する命令も変更になります。これらの部分を赤字で示します(訳注:本行は以下のプログラム補正に対応して追加しました)。RAMPZは予め設定されている前提です。

| ラベル   | 命令                               | 注釈                                     |
|-------|----------------------------------|--|
|       | . EQU PGSZB = PAGESIZE*2         | ; PGSZBはページ内のバイト数です。(PAGESIZEは語(ワード)数) |
|       | . ORG SMALLBOOTSTART             | ;                                      |
| WRPG: | LDI SPMC, (1<<PGERS)+(1<<SPMEN)  | ; [ページ消去]                              |
|       | CALL SPMJ                        | ; ページ消去SPMCSR値を取得                      |
|       |                                  | ; ページ消去                                |
|       |                                  | ; [RWW領域読み出し再許可]                       |
|       | LDI SPMC, (1<<RWWSRE)+(1<<SPMEN) | ; RWW領域読み出し許可SPMCSR値を取得                |
|       | CALL SPMJ                        | ; RWW領域読み出し許可                          |
|       |                                  | ; [RAMからフラッシュページ一時緩衝部へ転送]              |
|       | LDI CNTL, LOW(PGSZB)             | ; バイト計数器を初期化                           |
|       | LDI CNTH, HIGH(PGSZB)            | ; (削除)                                 |
| WLP:  | LD R0, Y+                        | ; RAM上の下位データを取得(ポインタ進行)                |
|       | LD R1, Y+                        | ; RAM上の上位データを取得(ポインタ進行)                |
|       | LDI SPMC, (1<<SPMEN)             | ; ページ一時緩衝部書き込みSPMCSR値を取得               |
|       | CALL SPMJ                        | ; 対応語(ワード)データをページ一時緩衝部に設定              |
|       | ADIW ZH: ZL, 2                   | ; ページ一時緩衝部ポインタ進行                       |
|       | SBIW CNTH: CNTL, 2               | ; 計数器を減数 (SUBI)                        |
|       | BRNE WLP                         | ; 指定バイト数分継続                            |
|       |                                  | ; [ページ書き込み]                            |
|       | SUBI ZL, LOW(PGSZB)              | ; ページ一時緩衝部先頭にポインタを復帰                   |
|       | SBCI ZH, HIGH(PGSZB)             | ; (削除)                                 |
|       | LDI SPMC, (1<<PGWRT)+(1<<SPMEN)  | ; フラッシュ書き込みSPMCSR値を取得                  |
|       | CALL SPMJ                        | ; フラッシュメモリページ書き込み                      |
|       |                                  | ; [RWW領域読み出し再許可]                       |
|       | LDI SPMC, (1<<RWWSRE)+(1<<SPMEN) | ; RWW領域読み出し許可SPMCSR値を取得                |
|       | CALL SPMJ                        | ; RWW領域読み出し許可                          |
|       |                                  | ; [読み戻し照合(任意)]                         |
|       | LDI CNTL, LOW(PGSZB)             | ; バイト計数器を初期化                           |
|       | LDI CNTH, HIGH(PGSZB)            | ; (削除)                                 |
|       | SUBI YL, LOW(PGSZB)              | ; RAMデータ先頭にポインタを復帰                     |
|       | SBCI YH, HIGH(PGSZB)             | ;                                      |
| RLP:  | ELPM R0, Z+                      | ; フラッシュメモリから1バイト取得(ポインタ進行)             |
|       | LD R1, Y+                        | ; RAMから1バイトデータを取得(ポインタ進行)              |
|       | CPSE R0, R1                      | ; 値一致でスキップ                             |
|       | JMP ERROR                        | ; 不一致で異常処理へ                            |
| ;     |                                  |  |
|       | SBIW CNTH: CNTL, 1               | ; 計数器を減数 (SUBI)                        |
|       | BRNE RLP                         | ; 指定バイト数分継続                            |
|       |                                  | ; [RWW領域へ復帰]                           |
| RTN:  | LDS TMP, SPMCSR                  | ; SPM命令制御/状態レジスタ値を取得                   |
|       | SBRS TMP, RWWSB                  | ; RWW領域多忙でスキップ                         |
|       | RET                              | ; 準備可で呼び出し元へ復帰                         |
| ;     |                                  | ; [RWW領域読み出し再許可]                       |
|       | LDI SPMC, (1<<RWWSRE)+(1<<SPMEN) | ; RWW領域読み出し許可SPMCSR値を取得                |
|       | CALL SPMJ                        | ; RWW領域読み出し許可                          |
|       | RJMP RTN                         | ; RWW領域準備可まで待機へ                        |
|       |                                  | ; [SPM命令実行サブルーチン]                      |
| SPMJ: | LDS TMP, SPMCSR                  | ; SPM命令制御/状態レジスタ値を取得                   |
|       | SBRC TMP, SPMEN                  | ; 操作可能(直前のSPM完了)でスキップ                  |
|       | RJMP SPMJ                        | ; 操作可まで待機                              |
| ;     |                                  |  |
|       | IN TMP, SREG                     | ; ステータスレジスタ値を保存                        |
|       | CLI                              | ; 全割り込み禁止                              |
| WAIT: | SBIC EECR, EEWEIF                | ; EEPROM書き込み中以外でスキップ                   |
|       | RJMP WAIT                        | ; EEPROM書き込み完了まで待機                     |
| ;     |                                  |  |
|       | STS SPMC, SPMC                   | ; SPM動作指定                              |
|       | SPM                              | ; 対応SPM動作実行                            |
|       | OUT SREG, TMP                    | ; ステータスレジスタ値を復帰                        |
|       | RET                              | ; 呼び出し元へ復帰                             |



## ATmega128用ブートローダのパラメータ

自己プログラミングの記述で使ったパラメータは表112.~114.で与えられます。

表112. 応用領域とブートローダ領域の分割設定

| BOOTSZ1 | BOOTSZ0 | ブートローダ領域 |      | アドレス範囲        |               |             |
|---------|---------|----------|------|---------------|---------------|-------------|
|         |         | 容量 (語)   | ページ数 | 応用プログラム領域     | ブートローダプログラム領域 | ブートリセットアドレス |
| 1       | 1       | 512      | 4    | \$0000~\$FDFF | \$FE00~\$FFFF | \$FE00      |
| 1       | 0       | 1024     | 8    | \$0000~\$FBFF | \$FC00~\$FFFF | \$FC00      |
| 0       | 1       | 2048     | 16   | \$0000~\$F7FF | \$F800~\$FFFF | \$F800      |
| 0       | 0       | 4096     | 32   | \$0000~\$EFFF | \$F000~\$FFFF | \$F000      |

注: 各種BOOTSZヒューズ設定は図133.で示されます。

表113. RWW領域とNRWW領域の範囲

| 領域                  | ページ数 | アドレス範囲        |
|---------------------|------|---------------|
| 書き込み中読み出し可能(RWW)領域  | 480  | \$0000~\$EFFF |
| 書き込み中読み出し不能(NRWW)領域 | 32   | \$F000~\$FFFF |

注: これら2つの領域についての詳細に関しては179頁の「RWW - 書き込み中読み出し可能領域」と「NRWW - 書き込み中読み出し不能領域」をご覧ください。

表114. 図134.で使った各変数説明とZポイントの配置

| 変数名      | 対応値    |           | 意味  |
|----------|--------|-----------|---|
|          | PC     | Zポイント (注) |   |
| PCMSB    | PC15   |           | プログラムカウンタの最上位ビット。(プログラムカウンタは16ビット、PC15~0)                           |
| PAGEMSB  | PC6    |           | 1ページ内に使われる語(ワード)アドレスの最上位ビット。<br>(ページ内の128語(ワード)には7ビット PC6~0が必要)     |
| ZPCMSB   |        | Z16       | PCMSBに配置される(対応する)Zレジスタ内のビット。<br>(Z0が使われないため、ZPCMSB=PCMSB+1)         |
| ZPAGEMSB |        | Z7        | PAGEMSBに配置される(対応する)Zレジスタ内のビット。<br>(Z0が使われないため、ZPAGEMSB=PAGEMSB+1)   |
| PCPAGE   | PC15~7 | Z16~8     | プログラムカウンタ ページ アドレス: ページ消去とページ書き込み用のページ選択                            |
| PCWORD   | PC6~0  | Z7~1      | プログラムカウンタ 語(ワード)アドレス: 一時緩衝部格納用語(ワード)選択<br>(ページ書き込み操作中は0でなければなりません。) |

注: ・ Zレジスタは16ビット幅だけです。ビット16はI/O領域のRAMPZレジスタに配置されます。

・ Z0: 全てのSPM命令に対して0であるべきで、(E)LPM命令に対するバイト選択です。

・ 自己プログラミング中のZポイントの使用に対する詳細に関しては182頁の「自己プログラミング中のフラッシュメモリのアドレス指定」をご覧ください。



## メモリプログラミング

## プログラムメモリとデータメモリ用施錠ビット

Atmel® AVR® ATmega128は非プログラム(1)のままか、表116.で一覧される付加機能を得るためにプログラム(0)できる6つの施錠ビットを提供します。この施錠ビットはチップ消去指令でのみ1に消去できます。

表115. 施錠ビットバイトの内容

| 名称    | ビット番号 | 意味                            | 既定値(注)     |
|-------|-------|-------------------------------|------------|
| -     | 7     |                               | 1 (非プログラム) |
| -     | 6     |                               | 1 (非プログラム) |
| BLB12 | 5     | ブートローダ領域に対する保護用ブート施錠ビット       | 1 (非プログラム) |
| BLB11 | 4     |                               | 1 (非プログラム) |
| BLB02 | 3     | 応用領域に対する保護用ブート施錠ビット           | 1 (非プログラム) |
| BLB01 | 2     |                               | 1 (非プログラム) |
| LB2   | 1     | フラッシュとEEPROMメモリに対する一般保護用施錠ビット | 1 (非プログラム) |
| LB1   | 0     |                               | 1 (非プログラム) |

注: 0はプログラム、1は非プログラムを意味します。

表116. 施錠ビットの保護種別

| メモリ施錠ビット |       |       | 保護種別   |
|----------|-------|-------|--|
| LB 種別    | LB2   | LB1   | 直列、並列、またはJTAG経由でプログラミングに対する保護  |
| 1        | 1     | 1     | メモリ施錠機能は機能しません。  |
| 2        | 1     | 0     | フラッシュ、EEPROMの更なるプログラミング(書き込み)が禁止されます。ヒューズビットが固定されます。(注1)                   |
| 3        | 0     | 0     | フラッシュ、EEPROMの更なるプログラミング(書き込み)と照合(読み出し)が禁止されます。ヒューズビットとブート施錠ビットが固定されます。(注1) |
| LB0 種別   | BLB02 | BLB01 | フラッシュメモリの応用プログラム領域に対する保護   |
| 1        | 1     | 1     | (E)LPM, SPM命令が応用領域をアクセスすることに対して制限はありません。                                   |
| 2        | 1     | 0     | SPM命令は応用領域に書くことを許されません。  |
| 3        | 0     | 0     | SPM命令による応用領域への書き込みと、ブートローダ領域での(E)LPM命令による応用領域からの読み込みが許されません。(注2)           |
| 4        | 0     | 1     | ブートローダ領域での(E)LPM命令による応用領域からの読み込みが許されません。(注2)                               |
| LB1 種別   | BLB12 | BLB11 | フラッシュメモリのブートローダプログラム領域に対する保護   |
| 1        | 1     | 1     | (E)LPM, SPM命令がブートローダ領域をアクセスすることに対して制限はありません。                               |
| 2        | 1     | 0     | SPM命令はブートローダ領域に書くことを許されません。  |
| 3        | 0     | 0     | SPM命令によるブートローダ領域への書き込みと、応用領域での(E)LPM命令によるブートローダ領域からの読み込みが許されません。(注3)       |
| 4        | 0     | 1     | 応用領域での(E)LPM命令によるブートローダ領域からの読み込みが許されません。(注3)                               |

注: 0はプログラム、1は非プログラムを意味します。

注1: 施錠ビットを書く前にヒューズビットとブート施錠ビットを書いてください。

注2: BLB02=0で、割り込みベクタがブートローダ領域に配置されていると、応用領域での実行時に割り込みが禁止されます。

注3: BLB12=0で、割り込みベクタが応用領域に配置されていると、ブートローダ領域での実行時に割り込みが禁止されます。

## 識票バイト

全てのAtmelマイクロコントローラはデバイスを識別する3バイトの識票符号を持ちます。この符号は直列と並列の両プログラミング動作で、またデバイスが施錠されていても読めます。この3バイトは分離された空間に存在します。Atmel® AVR® ATmega128の識票バイトを右に示します。

- ① \$000 : \$1E 製造業者Atmelを示します。
- ② \$001 : \$97 フラッシュメモリ容量128Kバイトを示します。
- ③ \$002 : \$02 ②値\$97と合せ、ATmega128を示します。

## 校正バイト

ATmega128は内蔵RC発振器用に4つの異なる校正値を備えています。これらのバイトは1,2,4,8MHzの各々に対して識票列のアドレス\$0000,\$0001,\$0002,\$0003の上位バイトにあります。リセット中に1MHz値がOSCCALレジスタへ自動的に設定されます。他の周波数が使われる場合、その校正値は手動で設定されなければなりません。詳細については27頁の「発振校正レジスタ(OSCCAL)」をご覧ください。

## ヒューズ ビット

Atmel® AVR® ATmega128には3つのヒューズ バイトがあります。表117.~119.は全てのヒューズの概略機能とヒューズ バイト内でどのように配置されるかを示します。ヒューズはプログラムされると、論理0として読まれることに注意してください。

表117. 拡張ヒューズ バイト一覧

| 名称         | ビット | 意味               | 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)              |
| M103C (注1) | 1   | ATmega103互換動作。   | 0 (プログラム) ATmega103互換動作 |
| WDTON (注2) | 0   | ウォッチドッグ タイマ常時有効。 | 1 (非プログラム) WDTはWDTCRで許可 |

注1: 詳細については4頁の「ATmega128とATmega103の互換性」をご覧ください。

注2: 詳細については35頁の「ウォッチドッグ タイマ制御レジスタ(WDTCR)」をご覧ください。

表118. ヒューズ 上位バイト一覧

| 名称          | ビット | 意味                            | 既定値                   |
|-------------|-----|-------------------------------|-----------------------|
| OCDEN (注4)  | 7   | 内蔵デバッグ機能(OCD)許可。              | 1 (非プログラム) OCD禁止      |
| JTAGEN (注5) | 6   | JTAGインターフェース許可。               | 0 (プログラム) JTAG許可      |
| SPIEN (注1)  | 5   | 直列プログラミング許可。                  | 0 (プログラム) 直列プログラミング許可 |
| CKOPT (注2)  | 4   | クロック発振器任意選択機能。                | 1 (非プログラム)            |
| EESAVE      | 3   | チップ消去からEEPROM内容を保護。           | 1 (非プログラム) EEPROMは未保護 |
| BOOTSZ1     | 2   | ブート ロータ容量選択。(表112.参照) (注3)    | 0 (プログラム)             |
| BOOTSZ0     | 1   |                               | 0 (プログラム)             |
| BOOTRST     | 0   | リセット ベクタ(応用領域またはブート ロータ領域)選択。 | 1 (非プログラム) 応用領域       |

注1: SPIENヒューズは直列プログラミングでアクセスできません。

注2: CKOPTヒューズの機能はCKSELヒューズ設定に依存します。詳細は24頁の「クロック元」をご覧ください。

注3: BOOTSZ1,0既定値は最大ブート ロータ容量になります。186頁の表112.をご覧ください。

注4: 施錠ビットやJTAGENヒューズの設定に拘らず、OCDENヒューズがプログラム(0)された製品の出荷は決してありません。プログラム(0)されたOCDENヒューズは全休止形態でクロック系の一部を許可します。これは消費電力の増加になってしまいます。

注5: JTAGインターフェースが未接続のままにされる場合、JTAGENヒューズは可能ならば禁止されるべきです。これはJTAGインターフェースのTDOピンの静止時電流を避けるためです。

表119. ヒューズ 下位バイト一覧

| 名称       | ビット | 意味                 | 既定値              |
|----------|-----|--------------------|------------------|
| BODLEVEL | 7   | 低電圧検出器(BOD)検出電圧選択。 | 1 (非プログラム)       |
| BODEN    | 6   | 低電圧検出器(BOD)許可。     | 1 (非プログラム) BOD禁止 |
| SUT1     | 5   | 起動時間選択。 (注1)       | 1 (非プログラム)       |
| SUT0     | 4   |                    | 0 (プログラム)        |
| CKSEL3   | 3   | クロック種別選択。 (注2)     | 0 (プログラム)        |
| CKSEL2   | 2   |                    | 0 (プログラム)        |
| CKSEL1   | 1   |                    | 0 (プログラム)        |
| CKSEL0   | 0   |                    | 1 (非プログラム)       |

注1: SUT1,0の既定値は最大起動時間になります。詳細については26頁の表14.をご覧ください。

注2: CKSEL3~0の既定設定は1MHz校正付き内蔵RC発振器になります。詳細については24頁の表6.をご覧ください。

ヒューズ ビットの状態はチップ消去によって影響されません。施錠ビット1(LB1)がプログラム(0)されると、ヒューズ ビットが固定されることに注意してください。施錠ビットをプログラム(0)する前にヒューズ ビットをプログラミング(書き込み)してください。

## ヒューズのラッチ

ヒューズ値はデバイスがプログラミング動作へ移行する時にラッチされ、ヒューズ値の変更はデバイスがプログラミング動作を去るまで無効です。これは一旦プログラム(0)されると直ぐに効果があるEESAVEヒューズには適用されません。ヒューズは通常動作での電源投入でもラッチされません。

## 並列プログラミング

この項はAtmel® AVR® ATmega128のプログラム用フラッシュメモリ、データ用EEPROM、メモリの**施錠ビット**、**ヒューズビット**の並列プログラミングと照合の方法を記述します。特記事項を除いて、パルス幅は最低250nsと仮定されます。

### 信号名

本章でATmega128のいくつかのピンは並列プログラミング中の機能を表す信号名で参照されます。**図135**と**表120**をご覧ください。後続の表で示されないピンはピン名によって参照されます。

XA0とXA1ピンはXTAL1ピンが正パルスを与えられる時に実行される動作を決めます。このビット符号化は**表122**で示されます。

WRまたはOEパルス送出時、設定された指令が決める動作が実行されます。各種指令は**表123**で示されます。

図135. 並列プログラミング構成図

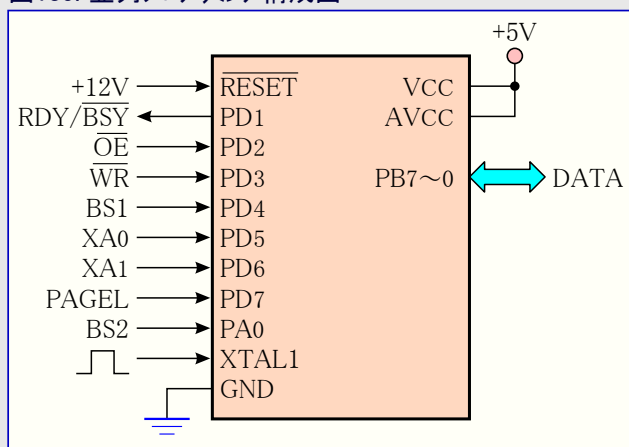


表120. 信号名とピン名の関係

| 信号名     | ピン名   | 入出力 | 機能  |
|---------|-------|-----|---|
| RDY/BSY | PD1   | 出力  | 0(Low) : 多忙(プログラミング中)    1(High) : 準備可(指令受付可) |
| OE      | PD2   | 入力  | 出力許可(負論理)                                     |
| WR      | PD3   | 入力  | 書き込みパルス(負論理)                                  |
| BS1     | PD4   | 入力  | 上位/下位バイト選択1 (0:下位, 1:上位) (一般用)                |
| XA0     | PD5   | 入力  | XTAL動作ビット0                                    |
| XA1     | PD6   | 入力  | XTAL動作ビット1                                    |
| PAGEL   | PD7   | 入力  | プログラムメモリとEEPROMデータをページ一時緩衝部に設定                |
| BS2     | PA0   | 入力  | 上位/下位バイト選択2 (0:下位, 1:上位) (ヒューズビット用)           |
| DATA    | PB7~0 | 入出力 | 双方向データバス (OE=Low時出力)                          |

表121. プログラミング動作移行時のピン値

| ピン名   | シンボル           | 値 |
|-------|----------------|---|
| PAGEL | Prog_enable[3] | 0 |
| XA1   | Prog_enable[2] | 0 |
| XA0   | Prog_enable[1] | 0 |
| BS1   | Prog_enable[0] | 0 |

表122. XA0とXA1の符号化(機能)

| XA1 | XA0 | XTAL1パルス時の動作                         |
|-----|-----|--------------------------------------|
| 0   | 0   | フラッシュまたはEEPROMのアドレス設定 (上位/下位はBS1で指定) |
| 0   | 1   | データ設定 (フラッシュの上位/下位はBS1で指定)           |
| 1   | 0   | 指令設定                                 |
| 1   | 1   | アイドル (動作なし)                          |

表123. 指令バイトのビット符号化

| 指令バイト            | 指令の機能             |
|------------------|-------------------|
| \$80 (1000 0000) | チップ消去             |
| \$40 (0100 0000) | ヒューズビット書き込み       |
| \$20 (0010 0000) | 施錠ビット書き込み         |
| \$10 (0001 0000) | フラッシュメモリ書き込み      |
| \$11 (0001 0001) | EEPROM書き込み        |
| \$08 (0000 1000) | 識票バイト、校正バイト読み出し   |
| \$04 (0000 0100) | ヒューズビット、施錠ビット読み出し |
| \$02 (0000 0010) | フラッシュメモリ読み出し      |
| \$03 (0000 0011) | EEPROM読み出し        |

表124. フラッシュメモリのページ数とページの語数

| 全容量               | ページ容量 | PCWORD | ページ数 | PCPAGE | PCMSB |
|-------------------|-------|--------|------|--------|-------|
| (64K語)<br>128Kバイト | 128語  | PC6~0  | 512  | PC15~7 | 15    |

表125. EEPROMメモリのページ数とページの語数

| 全容量   | ページ容量 | PCWORD | ページ数 | PCPAGE  | EEAMSB |
|-------|-------|--------|------|---------|--------|
| 4Kバイト | 8バイト  | EEA2~0 | 512  | EEA11~3 | 11     |

## 並列プログラミング手順

### 並列プログラミング動作への移行

次に示す方法がデバイスを並列プログラミング動作にします。

- ① VCCとGND間に4.5～5.5Vを印加し、最低100 $\mu$ s待ちます。
- ②  $\overline{\text{RESET}}$ をLow(0)にし、XTAL1を少なくとも6回、交互に(High/Low)切り替えます。
- ③ 189頁の表121.で一覧されるProg\_enableピンを全てLow(0)に設定し、最低100ns待ちます。
- ④  $\overline{\text{RESET}}$ に11.5～12.5Vを印加します。+12Vが $\overline{\text{RESET}}$ に印加されてしまった後100ns以内のProg\_enableピンのどんな動きも、デバイスのプログラミング動作への移行を失敗させます。

**注意:** 外部クリスタルや外部RC発振器設定が選ばれていると、条件付きのXTAL1パルスを印加できないため、同じことが当てはまるかもしれません。このような場合、次の方法に従うべきです。

- ① 189頁の表121.で一覧されるProg\_enableピンを全てLow(0)に設定します。
- ② VCCとGND間に4.5～5.5Vを印加すると同時に $\overline{\text{RESET}}$ へ11.5～12.5Vを印加します。
- ③ 100ns待ちます。
- ④ クロック元として外部クロックが選ばれる(CKSEL3～0=0000)ことを保証するためにヒューズを再プログラミング(再書き込み)します。施錠ビットがプログラム(0)されている場合、ヒューズを変更するのに先立ってチップ消去指令が実行されなければなりません。
- ⑤ デバイスの電源を落とすか、 $\overline{\text{RESET}}$ ピンをLow(0)に持つてくることによってプログラミング動作を抜けます。
- ⑥ 上で記述したように元の方法でプログラミング動作へ移行します。

### 効率的なプログラミングへの考慮

設定した指令とアドレスはプログラミング中、維持されます。効率的なプログラミングを行うために次を考慮すべきです。

- 複数のメモリ領域を読み書きする時に指令設定は一度だけ必要です。
- チップ消去後のフラッシュメモリと(EESAVEヒューズがプログラム(0)されている場合を除き)EEPROM全体の内容は\$FFなので、値が\$FFのデータ書き込みを飛ばします。
- アドレス上位バイトはフラッシュメモリで新規256語(ワード)枠、EEPROMで新規256バイト枠の読み書き前に一度だけ必要です。この考慮は識票バイト読み出しにも適用されます。

### チップ消去

チップ消去はフラッシュメモリ、EEPROM(注1)、施錠ビットを消去します。施錠ビットはプログラムメモリが完全に消去されてしまうまでリセット(消去)されません。ヒューズビットは変更されません。チップ消去はフラッシュメモリやEEPROMが再書き込みされる前に実行されなければなりません。

**注1:** EESAVEヒューズがプログラム(0)されていると、EEPROMはチップ消去中、保護されます。

「チップ消去」指令設定

- ① XA1をHigh(1)、XA0をLow(0)に設定します。これは指令設定を許可します。
- ② BS1をLow(0)に設定します。
- ③ DATAを\$80(1000 0000)に設定します。これはチップ消去指令です。
- ④ XTAL1に正パルスを与えます。これはチップ消去指令を設定します。
- ⑤  $\overline{\text{WR}}$ に負パルスを与えます。これはチップ消去を開始します。RDY/ $\overline{\text{BSY}}$ がLow(0)になります。
- ⑥ 次の指令を設定する前にRDY/ $\overline{\text{BSY}}$ がHigh(1)になるまで待ちます。

## フラッシュメモリ書き込み (図137.タイミングを参照)

フラッシュメモリはページで構成されます(189頁の表124.参照)。フラッシュメモリへ書く時にプログラムデータはページ緩衝部にラッチされます。これは同時に書かれることをプログラムデータの1ページに許します。次の手順は完全なフラッシュメモリの書き込み方法を記述します。

### A. 「フラッシュメモリ書き込み」指令設定

- ① XA1をHigh(1)、XA0をLow(0)に設定します。これは指令設定を許可します。
- ② BS1をLow(0)に設定します。
- ③ DATAを\$10(0001 0000)に設定します。これはフラッシュメモリ書き込み指令です。
- ④ XTAL1に正パルスを与えます。これはフラッシュメモリ書き込み指令を設定します。

### B. アドレス下位バイト設定

- ① XA1をLow(0)、XA0をLow(0)に設定します。これはアドレス設定を許可します。
- ② BS1をLow(0)に設定します。これは下位アドレス(バイト)を選びます。
- ③ DATAにアドレス下位バイト(\$00~\$FF)を設定します。
- ④ XTAL1に正パルスを与えます。これはアドレス下位バイトを設定します。

### C. データ下位バイト設定

- ① XA1をLow(0)、XA0をHigh(1)に設定します。これはデータ設定を許可します。
- ② DATAにデータ下位バイト(\$00~\$FF)を設定します。
- ③ XTAL1に正パルスを与えます。これはデータ下位バイトを設定します。

### D. データ上位バイト設定

- ① BS1をHigh(1)に設定します。これは上位バイトを選びます。
- ② XA1をLow(0)、XA0をHigh(1)に設定します。これはデータ設定を許可します。
- ③ DATAにデータ上位バイト(\$00~\$FF)を設定します。
- ④ XTAL1に正パルスを与えます。これはデータ上位バイトを設定します。

### E. 語(ワード)データをページ一時緩衝部に設定

- ① BS1をHigh(1)にします。これは上位バイトを選びます。
- ② PAGELに正パルスを与えます。これは語データをページ一時緩衝部にラッチ(設定)します。

### F. 緩衝部全体が満たされるか、または(必要な)全てのデータが緩衝部内に設定されるまで、B~Eを繰り返す

アドレス内の下位ビットがページ内の語(ワード)位置を指示する一方、上位ビットがフラッシュメモリ内のページをアドレス指定します。これは図136.で図示されます。ページ内の語アドレスに8ビット未満が必要とされる場合(ページ容量<256)、アドレス下位バイトの最上位(側)ビットがページ書き込み実行時のページアドレスに使われることに注意してください。

### G. アドレス上位バイト設定

- ① XA1をLow(0)、XA0をLow(0)に設定します。これはアドレス設定を許可します。
- ② BS1をHigh(1)に設定します。これは上位アドレス(バイト)を選びます。
- ③ DATAにアドレス上位バイト(\$00~\$FF)を設定します。
- ④ XTAL1に正パルスを与えます。これはアドレス上位バイトを設定します。

### H. ページ書き込み

- ① BS1をLow(0)に設定します。
- ② WRに負パルスを与えます。これはデータのページ全体の書き込みを開始します。RDY/ $\overline{BSY}$ がLow(0)になります。
- ③ RDY/ $\overline{BSY}$ がHigh(1)になるまで待ちます。

### I. フラッシュメモリ全部が書かれるか、または(必要な)全データが書かれてしまうまで、B~Hを繰り返す

### J. ページ書き込み終了

- ① XA1をHigh(1)、XA0をLow(0)に設定します。これは指令設定を許可します。
- ② DATAを\$00(0000 0000)にします。これは無操作指令です。
- ③ XTAL1に正パルスを与えます。これは無操作指令を設定し、内部書き込み信号がリセットされます。

図136. ページで構成されたフラッシュメモリのアドレス指定

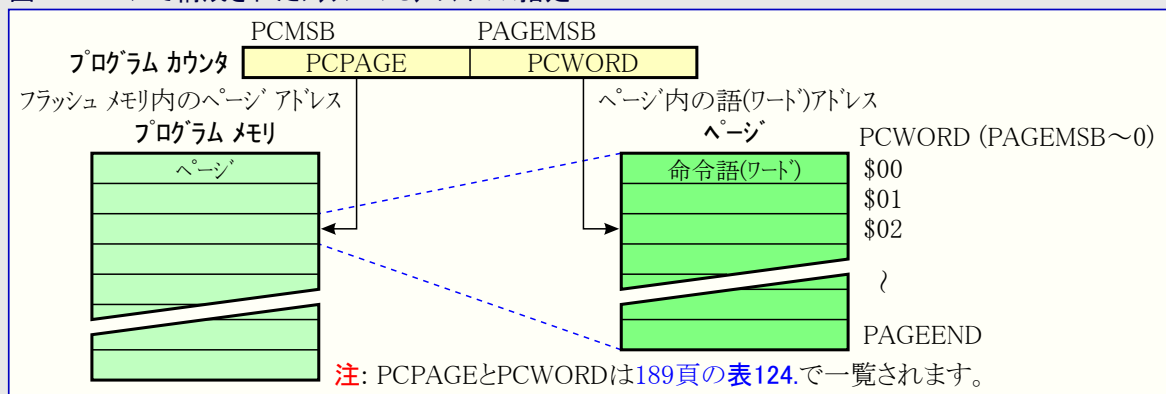
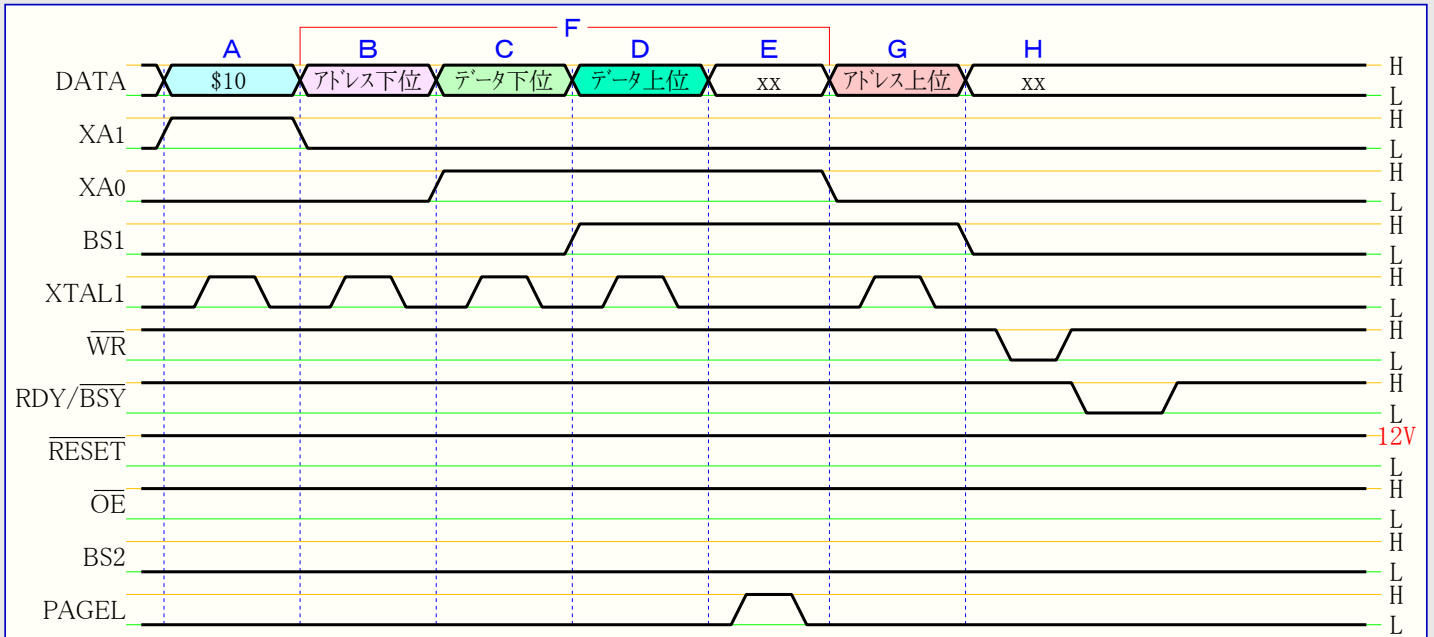




図137. フラッシュ メリ書き込みタイミング



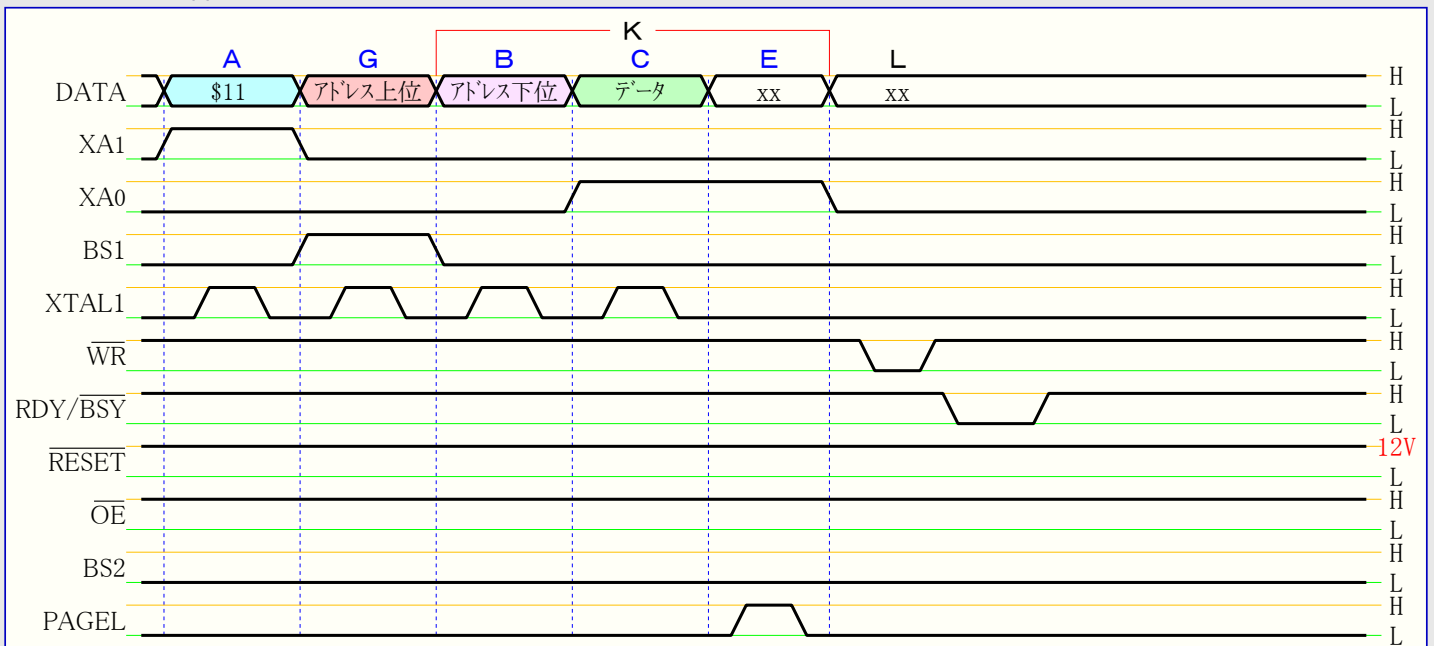
注: xx値は無関係です。A～Hは前記プログラミングを参照してください。

### EEPROM書き込み

EEPROMはページで構成されます(189頁の表125参照)。EEPROMを書く時にデータはページ緩衝部にラッチされます。これは同時に書かれることをデータの1ページに許します。データ用EEPROMメモリの書き込み方法は次のとおりです。(指令、アドレス、データ設定の詳細については191頁の「フラッシュメモリの書き込み」を参照。図138タイミング参照。)

1. EEPROM書き込み指令\$11(0001 0001)を設定します。(「フラッシュメモリ書き込み」のAを参照)
2. アドレス上位バイト(\$00～\$0F)を設定します。(「フラッシュメモリ書き込み」のGを参照)
3. アドレス下位バイト(\$00～\$FF)を設定します。(「フラッシュメモリ書き込み」のBを参照)
4. データバイト(\$00～\$FF)を設定します。(「フラッシュメモリ書き込み」のCを参照)
5. データをラッチします(PAGELに正パルスを与えます)。(「フラッシュメモリ書き込み」のEを参照)
- K. 緩衝部全体が満たされるまで3～5を繰り返します。
- L. EEPROMページ書き込み
  - ① BS1をLow(0)に設定します。
  - ② WRに負パルスを与えます。これはEEPROMページ書き込みを開始します。RDY/BSYがLow(0)になります。
  - ③ 次のページを書く前に、RDY/BSYがHigh(1)になるまで待ちます。

図138. EEPROM書き込みタイミング



## フラッシュメモリ読み出し

フラッシュメモリの読み出し方法は次のとおりです。(指令とアドレス設定の詳細については191頁の「フラッシュメモリの書き込み」を参照)

1. フラッシュメモリ読み出し指令\$02(0000 0010)を設定します。(「フラッシュメモリ書き込み」のAを参照)
2. アドレス上位バイト(\$00~\$FF)を設定します。(「フラッシュメモリ書き込み」のGを参照)
3. アドレス下位バイト(\$00~\$FF)を設定します。(「フラッシュメモリ書き込み」のBを参照)
4. BS1をLow(0)、OEをLow(0)に設定します。フラッシュメモリ語(ワード)の下位バイトが直ぐにDATAで読めます。
5. BS1をHigh(1)に設定します。フラッシュメモリ語(ワード)の上位バイトが直ぐにDATAで読めます。
6. OEをHigh(1)に設定します。DATAはHi-Zになります。

## EEPROM読み出し

データ用EEPROMの読み出し方法は次のとおりです。(指令とアドレス設定の詳細については191頁の「フラッシュメモリの書き込み」を参照)

1. EEPROM読み出し指令\$03(0000 0011)を設定します。(「フラッシュメモリ書き込み」のAを参照)
2. アドレス上位バイト(\$00~\$0F)を設定します。(「フラッシュメモリ書き込み」のGを参照)
3. アドレス下位バイト(\$00~\$FF)を設定します。(「フラッシュメモリ書き込み」のBを参照)
4. BS1をLow(0)、OEをLow(0)に設定します。EEPROMのバイトデータが直ぐにDATAで読めます。
5. OEをHigh(1)に設定します。DATAはHi-Zになります。

**ヒューズビット書き込み** (訳注:原書での拡張/上位/下位ヒューズ項を1つに纏めました。)

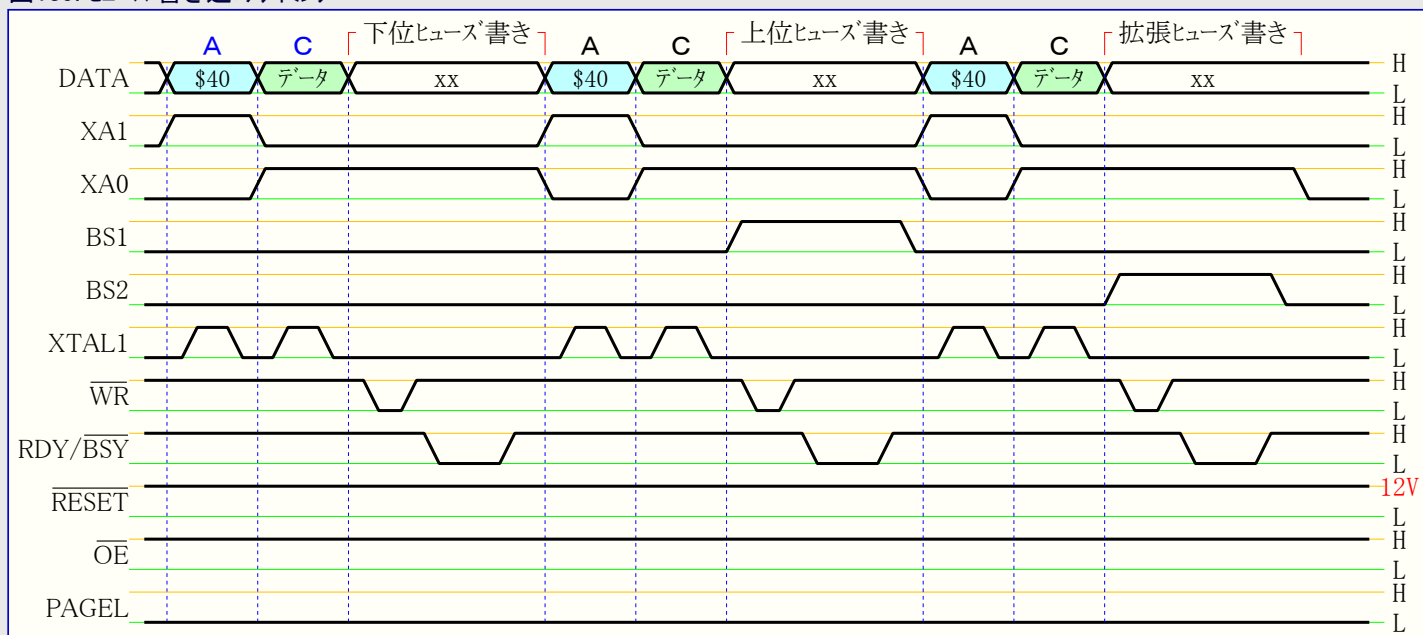
各ヒューズバイトの書き込み方法は次のとおりです。(指令とデータ設定の詳細については191頁の「フラッシュメモリの書き込み」を参照)

1. ヒューズビット書き込み指令\$40(0100 0000)を設定します。(「フラッシュメモリ書き込み」のAを参照)
2. データ下位バイトを設定します。0=プログラム,1=非プログラム(消去)です。(「フラッシュメモリ書き込み」のCを参照)
3. BS1とBS2を右表の目的バイトに対応する設定にします。
4. WRに負ハルスを与え、RDY/BSYがHighになるまで待ちます。
5. 3.でHighに設定したBS1、BS2をLow(0)に戻します。これはデータ下位バイトを選びます。

表A. ヒューズバイト対応BS1,BS2設定

| ヒューズバイト | BS1     | BS2     |
|---------|---------|---------|
| 拡張バイト   | Low(0)  | High(1) |
| 上位バイト   | High(1) | Low(0)  |
| 下位バイト   | Low(0)  | Low(0)  |

図139. ヒューズ書き込みタイミング



## 施錠ビット書き込み

施錠ビットの書き込み方法は次のとおりです。(指令とデータ設定の詳細については191頁の「フラッシュメモリの書き込み」を参照)

1. 施錠ビット書き込み指令\$20(0010 0000)を設定します。(「フラッシュメモリ書き込み」のAを参照)
2. データ下位バイト(としてデータ)を設定します。0=プログラム,1=無変化です。(「フラッシュメモリ書き込み」のCを参照)
3.  $\overline{WR}$ に負パルスを与え、RDY/BSYがHighになるまで待ちます。

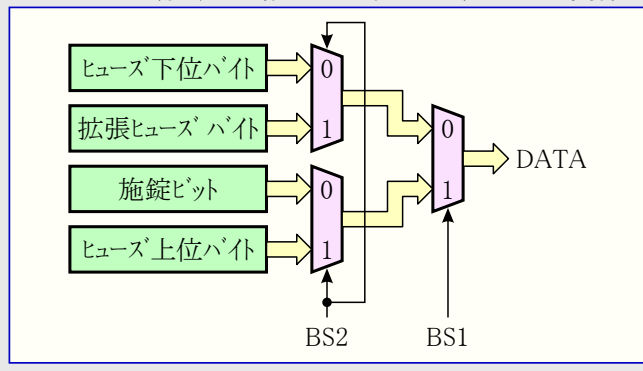
施錠ビットはチップ消去の実行によってのみ解除(1)できます。

## ヒューズビットと施錠ビットの読み出し

ヒューズビットと施錠ビットの読み出し方法は次のとおりです。(指令設定の詳細については191頁の「フラッシュメモリの書き込み」を参照)

1. ヒューズビットと施錠ビットの読み出し指令\$04(0000 0100)を設定します。(「フラッシュメモリ書き込み」のAを参照)
2. BS1とBS2をLow(0)、 $\overline{OE}$ をLow(0)に設定します。ヒューズ下位ビットの状態が直ぐにDATAで読めます。(0=プログラム)
3. BS1とBS2をHigh(1)、 $\overline{OE}$ をLow(0)に設定します。ヒューズ上位ビットの状態が直ぐにDATAで読めます。(0=プログラム)
4. BS1をLow(0)、BS2をHigh(1)、 $\overline{OE}$ をLow(0)に設定します。拡張ヒューズビットの状態が直ぐにDATAで読めます。(0=プログラム)
5. BS1をhigh(1)、BS2をLow(0)、 $\overline{OE}$ をLow(0)に設定します。施錠ビットの状態が直ぐにDATAで読めます。(0=プログラム)
6.  $\overline{OE}$ をHigh(1)に設定します。DATAはHi-Zになります。

図140. ヒューズ、施錠ビット読み出し中のBS1, BS2との関係



## 識票バイト読み出し

識票バイトの読み出し方法は次のとおりです。(指令とアドレス設定の詳細については191頁の「フラッシュメモリの書き込み」を参照)

1. 識票バイト読み出し指令\$08(0000 1000)を設定します。(「フラッシュメモリ書き込み」のAを参照)
2. アドレス下位バイト(\$00~\$02)を設定します。(「フラッシュメモリ書き込み」のBを参照)
3. BS1をLow(0)、 $\overline{OE}$ をLow(0)に設定します。選んだ識票バイトが直ぐにDATAで読めます。
4.  $\overline{OE}$ をHigh(1)に設定します。DATAはHi-Zになります。

## 校正バイト読み出し

校正バイトの読み出し方法は次のとおりです。(指令とアドレス設定の詳細については191頁の「フラッシュメモリの書き込み」を参照)

1. 校正バイト読み出し指令\$08(0000 1000)を設定します。(「フラッシュメモリ書き込み」のAを参照)
2. アドレス下位バイトに(\$00~\$03)を設定します。(「フラッシュメモリ書き込み」のBを参照)
3. BS1をHigh(1)、 $\overline{OE}$ をLow(0)に設定します。校正バイトが直ぐにDATAで読めます。
4.  $\overline{OE}$ をHigh(1)に設定します。DATAはHi-Zになります。

並列プログラミング特性

図141. 並列プログラミング タイミング (一般的な必要条件)

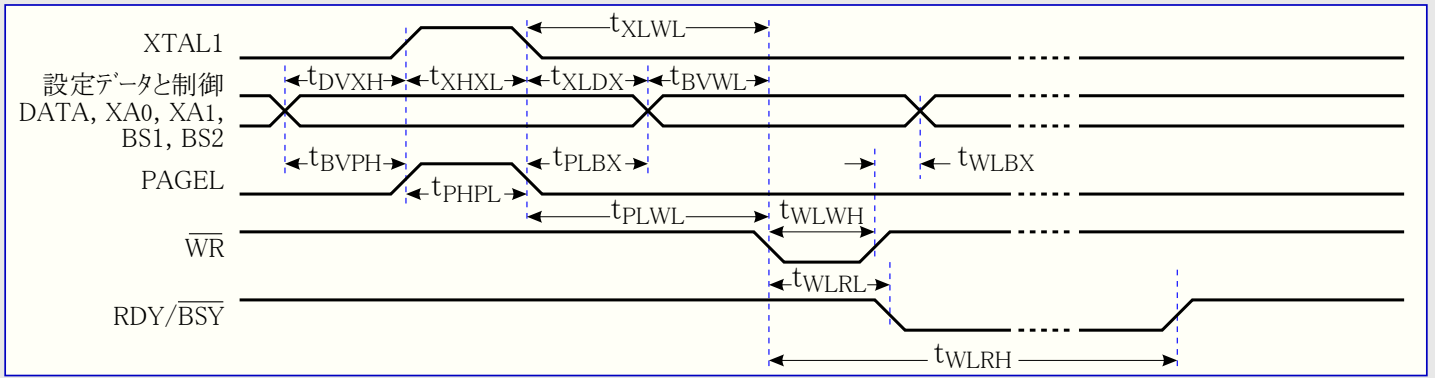
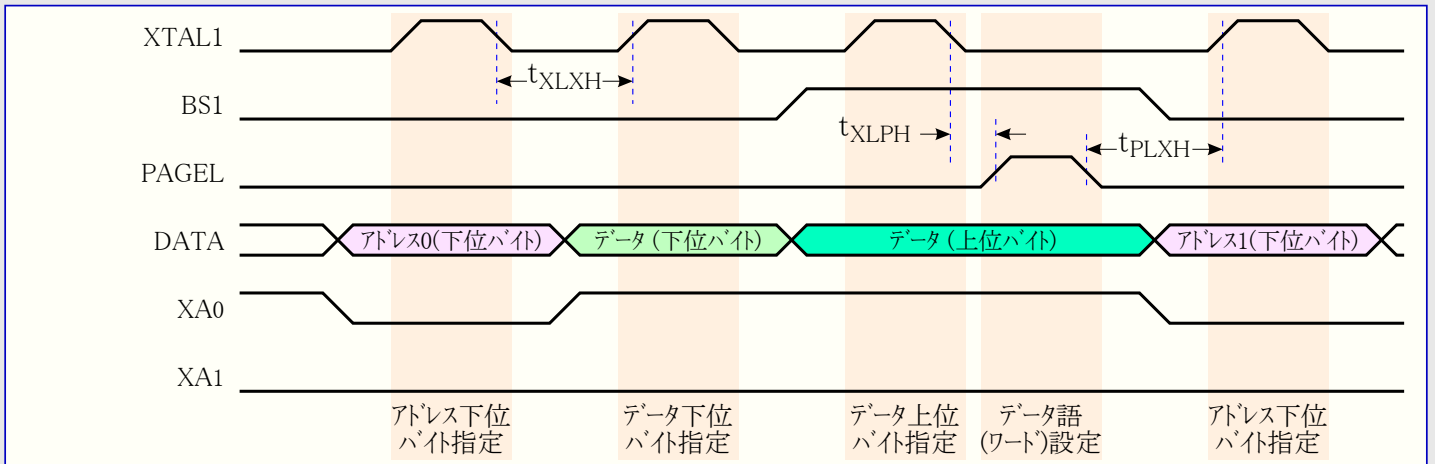
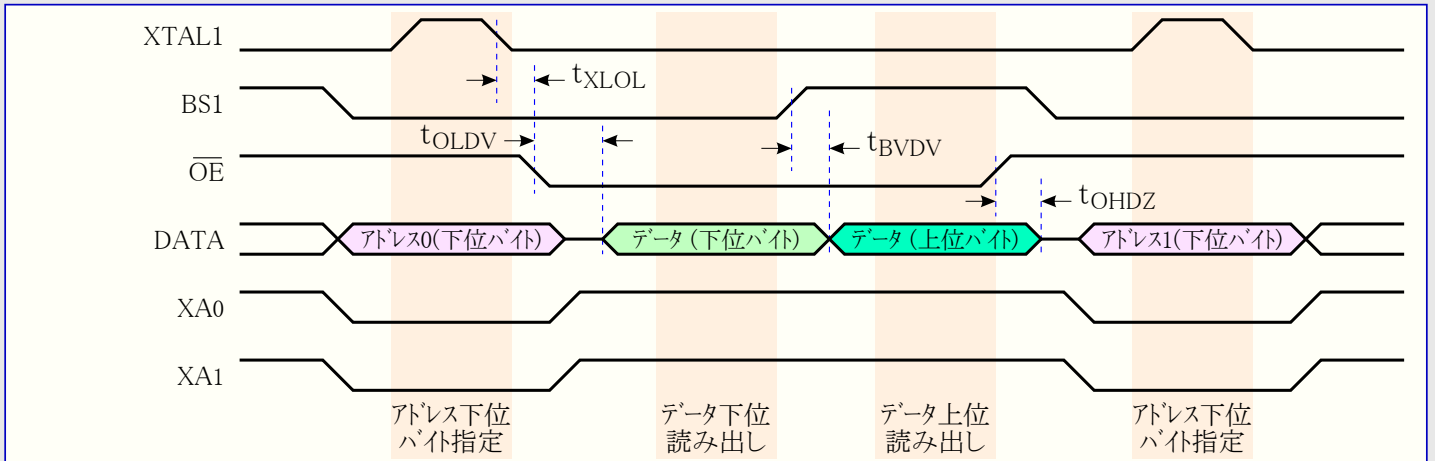


図142. 並列プログラミング タイミング (ページ設定での必要条件)



注: 図141.で示されるタイミング必要条件(即ち、 $t_{DVXH}$ 、 $t_{XHXL}$ 、 $t_{XLDX}$ )は設定操作にも適用されます。

図143. 並列プログラミング タイミング (同一ページ読み出しでの必要条件)



注: 図141.で示されるタイミング必要条件(即ち、 $t_{DVXH}$ 、 $t_{XHXL}$ 、 $t_{XLDX}$ )は読み出し操作にも適用されます。

表126. 並列プログラミング特性 (VCC=5V±10%)

| シンボル     | 項目                              | 最小   | 代表 | 最大   | 単位 |
|----------|---------------------------------|------|----|------|----|
| VPP      | プログラミング許可電圧                     | 11.5 |    | 12.5 | V  |
| IPP      | プログラミング許可電流                     |      |    | 250  | μA |
| tDVXH    | XTAL1 ↑ に対するデータと制御の準備時間         | 67   |    |      | ns |
| tXLXH    | XTAL1 ↓ から次XTAL1 ↑ までの待機時間      | 200  |    |      |    |
| tXHXL    | XTAL1 Highパルス幅                  | 150  |    |      |    |
| tXLDX    | XTAL1パルス ↓ 後のデータと制御の保持時間        | 67   |    |      |    |
| tXLWL    | XTAL1パルス ↓ 後のWR ↓ 待機時間          | 0    |    |      |    |
| tXLPH    | XTAL1パルス ↓ 後のPAGELパルス ↑ 待機時間    | 0    |    |      |    |
| tPLXH    | PAGELパルス ↓ 後のXTAL1パルス ↑ 待機時間    | 150  |    |      |    |
| tBVPH    | PAGELパルス ↑ に対するBS1準備時間          | 67   |    |      |    |
| tPHPL    | PAGEL Highパルス幅                  | 150  |    |      |    |
| tPLBX    | PAGELパルス ↓ 後のBS1保持時間            | 67   |    |      |    |
| tWLBX    | WRパルス ↓ 後のBS1,BS2保持時間           | 67   |    |      |    |
| tPLWL    | PAGELパルス ↓ 後のWRパルス ↓ 待機時間       | 67   |    |      |    |
| tBVWL    | WRパルス ↓ に対するBS1準備時間             | 67   |    |      |    |
| tWLWH    | WR Lowパルス幅                      | 150  |    |      |    |
| tWLRH    | WRパルス ↓ 後のRDY/BSY ↓ 遅延時間        | 0    |    | 1    |    |
| tWLRH    | 書き込み時間 (WR ↓ からRDY/BSY ↑) (注1)  | 3.7  |    | 5    | ms |
| tWLRH_CE | チップ消去時間 (WR ↓ からRDY/BSY ↑) (注2) | 7.5  |    | 10   |    |
| tXLCL    | XTAL1パルス ↓ 後のOE ↓ 待機時間          | 0    |    |      | ns |
| tBVDV    | BS1有効からのDATA遅延時間                | 0    |    | 250  |    |
| tOLDV    | OE ↓ 後のDATA出力遅延時間               |      |    | 250  |    |
| tOHDZ    | OE ↑ 後のDATA Hi-Z遅延時間            |      |    | 250  |    |

注1: フラッシュメモリ、EEPROM、ヒューズビット、施錠ビット書き込み指令に対して有効です。

注2: チップ消去指令に対して有効です。



## 直列プログラミング

フラッシュメモリとEEPROMの両方はRESETがGNDに引かれている間に直列SPIバスを使ってプログラミングできます。この直列インターフェースはSCK入力、MOSI入力、MISO出力から成ります。RESETがLowに設定された後、書き込み/消去操作が実行されるのに先立って**プログラミング許可命令**が初めに実行されるのを必要とします。**注意**、表127.でSPIプログラミング用のピン配置が一覧されます。全てのデバイスが内部SPIインターフェースに対する専用SPIピンを使うとは限りません。直列プログラミングについての記述全てでMOSIとMISOが各々直列入力と直列出力の記述に使われます。Atmel® AVR® ATmega128で、これらのピンはPDIとPDOに割り当てられます。

### 直列プログラミング用ピン配置

SPIプログラミングインターフェースがSPI I/O部を流用すると言え、1つの重要な違い、SPI I/O部でPB2とPB3に配置されるMOSI/MISOピンがこのプログラミングインターフェースで使われないことがあります。代わりに表127.で示されるようにPE0とPE1がSPIプログラミング動作のデータに使われます。

表127. 直列プログラミング用ピン配置

| 信号名        | ピン名 | 入出力 | 機能      |
|------------|-----|-----|---------|
| MOSI (PDI) | PE0 | 入力  | 直列データ入力 |
| MISO (PDO) | PE1 | 出力  | 直列データ出力 |
| SCK        | PB1 | 入力  | 直列クロック  |

EEPROMをプログラミングする時に自動消去周期が自動書き込み動作内に組み入れられ(直列プログラミングのみ)、**チップ消去命令**を初めに実行する必要がありません。チップ消去操作はプログラム(フラッシュメモリ)とEEPROM両方の全てのメモリ位置の内容を\$FFにします。

CKSELヒューズによる有効なクロックが与えられなければなりません。直列クロック(SCK)入力のLowとHighの区間の最小値は次のように定義されます。

$$\begin{array}{ll}
 f_{CK} < 12\text{MHz} : \text{Low区間} > 2 \text{ CPUクロック周期} & f_{CK} < 12\text{MHz} : \text{High区間} > 2 \text{ CPUクロック周期} \\
 f_{CK} \geq 12\text{MHz} : \text{Low区間} > 3 \text{ CPUクロック周期} & f_{CK} \geq 12\text{MHz} : \text{High区間} > 3 \text{ CPUクロック周期}
 \end{array}$$

### 直列プログラミング手順

ATmega128に直列データを書く時にデータはSCKの上昇端で行われ、ATmega128から読む時にデータはSCKの下降端で行われます。タイミングの詳細については図145.をご覧ください。

直列プログラミング動作でのATmega128のプログラミングと照合は次の手順が推奨されます(4バイト命令形式は表129.を参照)。

#### 1. 電源投入手順:

RESETとSCKがLow(0)に設定されている間にVCCとGND間へ電源を印加します。いくつかのシステムで電源投入中、SCKがLowに保持されるのを書き込み器が保証できません。この場合、SCKがLow(0)に設定されてしまった後、RESETは最低2 CPUクロック周期幅の正パルスを与えられなければなりません。

RESET信号を使う代わりに、SCKがLow(0)に設定される間の電源ONリセット中、PENをLowに保持できます。この場合、電源ONリセットでのPEN値だけが重要です。電源投入中、SCKがLowに保持されるのを書き込み器が保証できない場合、このPEN手法は使えません。この方法を使うと、通常動作を始めるためにデバイスは電源を落とされなければなりません。

#### 2. 最低20ms待ち、MOSIピンにプログラミング許可命令を送ることによって直列プログラミングを許可してください。

#### 3. 通信が同期を外していると、直列プログラミング命令は動作しません。同期していると、プログラミング許可命令の第3バイト送出時に第2バイト(\$53)が戻ります。この戻りが正しいかどうかによらず、命令の4バイト全てが送信されなければなりません。\$53が戻らない場合、RESETに正パルスを与え、新規プログラミング許可命令を行ってください。

#### 4. フラッシュメモリは1ページ単位で書かれます。ページ容量は189ページの表124.で得られます。このメモリページはページ設定命令と共にアドレスの下位7+1ビットとデータを供給することによって1バイトずつ設定されます。ページが正しく設定されるのを保証するため、与えられたアドレスにデータ上位バイトが適用される前にデータ下位バイトが設定されなければなりません。(プログラムメモリ)のページはアドレスの上位9ビットを含むページ書き込み命令の設定によって(フラッシュメモリ)に格納されます。ホーリングが使われない場合、使用者は次のページを行う前に最低tWD\_FLASH(表128.参照)待たなければなりません。

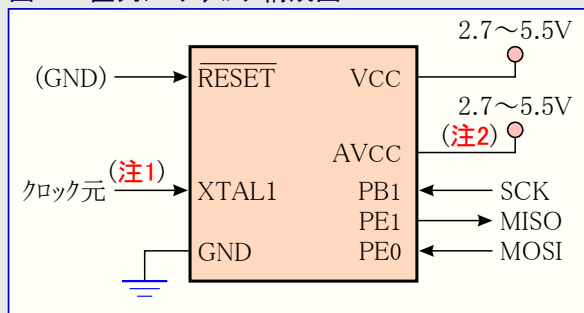
**注:** 何れの(フラッシュ、EEPROM、施錠ビット、ヒューズ)書き込み操作が完了される前にホーリング(読み)以外の命令が加えられると、不正な書き込みに終わるかもしれません(訳注:共通性のため原書の重複行削除、後部Note行移動)。

#### 5. EEPROMは適切なEEPROM書き込み命令と共にアドレスとデータを供給することによって1バイト単位で書かれます。EEPROMのメモリ位置は新規データが書かれるのに先立って始めて自動的に消去されます。ホーリングが使われない場合、次のバイトを行う前に最低tWD\_EEPROM(表128.参照)待たなければなりません。

#### 6. どのメモリ位置も選んだアドレスの内容を直列出力MISOに読み戻す読み出し命令を使うことによって照合できます。

#### 7. プログラミング作業終了時、RESETは通常動作を開始するため、High(1)に設定できます。

図144. 直列プログラミング構成図



**注1:** デバイスが内蔵発振器で動作する場合、XTAL1ピンにクロック元を接続する必要はありません。

**注2:** VCC-0.3V < AVCC < VCC+0.3Vですが、AVCCは常に2.7~5.5V内にすべきです。

### 8. 電源OFF手順 (必要とされるならば)

- ・ RESETをHigh(1)に設定します。
- ・ VCC電源をOFFにします。

### フラッシュメモリのデータポーリング

フラッシュメモリにページが書かれつつある時の書かれているページ内のアドレス位置読み出しは値\$FFを与えます。デバイスが新規ページに対する準備ができると、書いた値が正しく読めます。これは次ページが書ける時を決めるのに使われます。ページ全体が同時に書かれ、ページ内のどのアドレスでもポーリングに使えることに留意してください。フラッシュメモリのデータポーリングは値\$FFに対して行えないので、この値を書く時に使用者は次ページを書く前に最低tWD\_FLASH待たなければなりません。全ての場所が\$FFのチップ消去されたデバイスは\$FFデータのアドレスの書き込みを飛ばせます。tWD\_FLASH値については表128をご覧ください。

### EEPROMのデータポーリング

新規バイト(書き込み命令)が書かれてEEPROM内に書かれつつある時のその書かれているアドレス位置読み出しは値\$FFを与えます。デバイスが新規バイトに対する準備ができると、書いた値が正しく読めます。これは次バイトが書ける時を決めるのに使われます。これは値\$FFに対して行えませんが、使用者は次のことを覚えておくべきです。全ての場所が\$FFのチップ消去されたデバイスは\$FFデータのアドレスの書き込みを飛ばせます。これはデバイスをチップ消去せずにEEPROMが書き換えられる場合には適用しません。この場合、データポーリングは値\$FFに対して使えず、次バイト書き込み前に最低tWD\_EEPROM待たなければなりません。tWD\_EEPROM値については表128をご覧ください。

表128. ヒューズ, フラッシュ, EEPROM次位置書き込み前の待機時間

| シンボル       | 最低待機時間 | 備考           |
|------------|--------|--------------|
| tWD_FUSE   | 4.5ms  | ヒューズ書き込み     |
| tWD_FLASH  | 5ms    | フラッシュメモリ書き込み |
| tWD_EEPROM | 10ms   | EEPROM書き込み   |
| tWD_ERASE  | 10ms   | チップ消去        |

図145. 直列プログラミングバイト通信波形

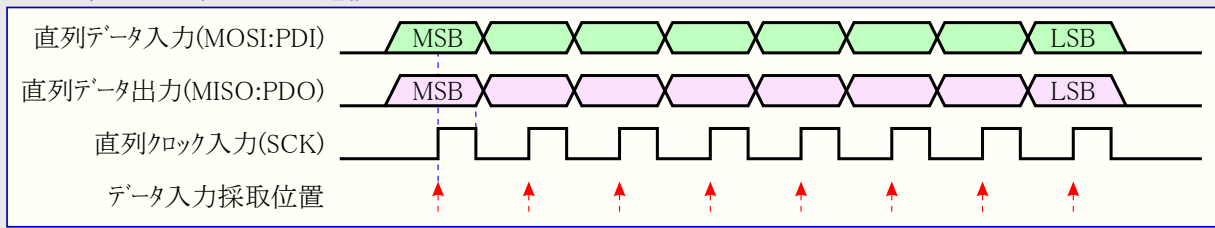


表129. 直列プログラミング命令一表

| 命令           | 命令形式      |           |           |           | 動作                             |          |
|--------------|-----------|-----------|-----------|-----------|--------------------------------|----------|
|              | 第1バイト     | 第2バイト     | 第3バイト     | 第4バイト     |                                |          |
| プログラミング許可    | 1010 1100 | 0101 0011 | xxxx xxxx | xxxx xxxx | RESET=Low後のプログラミングを許可します。      |          |
| チップ消去        | 1010 1100 | 100x xxxx | xxxx xxxx | xxxx xxxx | フラッシュメモリ, EEPROM, 施錠ビットを消去します。 |          |
| フラッシュメモリ読み出し | 0010 P000 | HHHH HHHH | LLLL LLLL | RRRR RRRR | アドレスH:LのP(H/L)バイトを読み出します。      |          |
| ページ一時緩衝部設定   | 0100 P000 | 0000 xxxx | xLLL LLLL | WWWW WWWW | 緩衝部アドレスLのP(H/L)バイトに書き込みます。     |          |
| フラッシュページ書き込み | 0100 1100 | HHHH HHHH | Lxxx xxxx | xxxx xxxx | アドレスH:Lのページに書き込みます。            |          |
| EEPROM読み出し   | 1010 0000 | 00xx HHHH | LLLL LLLL | RRRR RRRR | アドレスH:Lのバイトを読み出します。            |          |
| EEPROM書き込み   | 1100 0000 | 00xx HHHH | LLLL LLLL | WWWW WWWW | アドレスH:Lのバイトに書き込みます。            |          |
| 施錠ビット読み出し    | 0101 1000 | 0000 0000 | xxxx xxxx | xxRR RRRR | 施錠ビット                          | を読み出します。 |
| 施錠ビット書き込み    | 1010 1100 | 111x xxxx | xxxx xxxx | 11WW WWWW | (187頁の表115.参照)                 | に書き込みます。 |
| ヒューズ下位読み出し   | 0101 0000 | 0000 0000 | xxxx xxxx | RRRR RRRR | ヒューズ下位ビット                      | を読み出します。 |
| ヒューズ下位書き込み   | 1010 1100 | 1010 0000 | xxxx xxxx | WWWW WWWW | (188頁の表119.参照)                 | に書き込みます。 |
| ヒューズ上位読み出し   | 0101 1000 | 0000 1000 | xxxx xxxx | RRRR RRRR | ヒューズ上位ビット                      | を読み出します。 |
| ヒューズ上位書き込み   | 1010 1100 | 1010 1000 | xxxx xxxx | WWWW WWWW | (188頁の表118.参照)                 | に書き込みます。 |
| 拡張ヒューズ読み出し   | 0101 0000 | 0000 1000 | xxxx xxxx | RRRR RRRR | 拡張ヒューズビット                      | を読み出します。 |
| 拡張ヒューズ書き込み   | 1010 1100 | 1010 0100 | xxxx xxxx | WWWW WWWW | (188頁の表117.参照)                 | に書き込みます。 |
| 識票バイト読み出し    | 0011 0000 | 00xx xxxx | xxxx xxLL | RRRR RRRR | アドレスLの識票バイトを読み出します。            |          |
| 校正バイト読み出し    | 0011 1000 | 00xx xxxx | 0000 00LL | RRRR RRRR | アドレスLの校正バイトを読み出します。            |          |

H = アドレス上位バイトのビット      L = アドレス下位バイトのビット      P = 0=下位バイト, 1=上位バイト  
R = 読み出しデータ (MCU出力)      W = 書き込みデータ (MCU入力)      x = 0または1 (無意味/不定)

### 直列プログラミング特性

SPI部の特性については209頁の「SPIタイミング特性」を参照してください。

## JTAGインターフェース経由のプログラミング

JTAGインターフェースを通すプログラミングは4つのJTAG仕様ピン、TCK,TMS,TDI,TDOの制御が必要です。RESETピンとクロックピンの制御は必要とされません。

JTAGインターフェースを使い得るにはJTAGENヒューズがプログラム(0)されなければなりません。本デバイスは既定でこのヒューズがプログラム(0)されて出荷されます。更にMCU制御/状態レジスタ(MCUCSR)のJTAGインターフェース禁止(JTD)ビットが解除(0)されなければなりません。JTDビットが設定(1)の場合、代わりに外部リセットを強制的なLowにできます。その後、2チップ(CPU)クロック後にJTDビットが解除(0)され、JTAGピンはプログラミングに利用できます。これはJTAGインターフェース経由実装書き込みを許すにも拘らず、一方、実行動作で標準ポートピンとしてJTAGピンを使う手段を提供します。境界走査(Boundary-Scan)や内蔵デバッグ機能にJTAGピンを使う時にこの手法が使えないことに注意してください。これらの場合のJTAGピンはその目的専用にならなければなりません。

この文書内の定義では全ての移動レジスタについて入出力ともLSBが最初に移動されます。

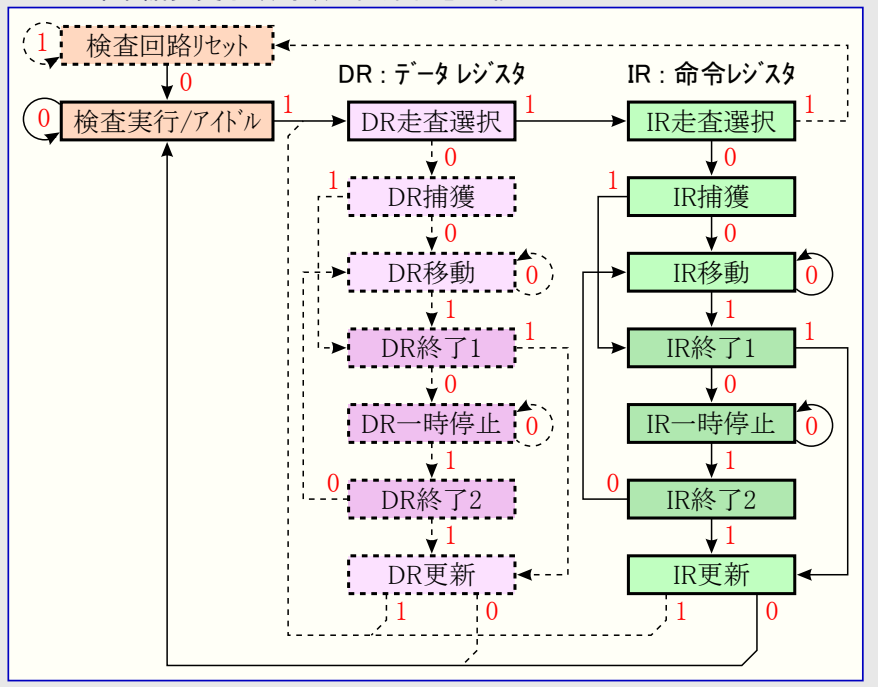
## プログラミング特殊JTAG命令

命令レジスタ(IR)は4ビット長で、16種類までの命令を支援します。以下に示される一覧はプログラミングに有用なJTAG命令です。

各命令の命令コードは命令名後の16進形式で示されます。本文は各命令についてTDIとTDO間の経路として選ばれるデータレジスタを記述します。

検査入出力ポート(TAP)制御器の検査実行/アイドル状態は内部クロックの発生に使われます。また、JTAG手順間のアイドル状態としても使えます。命令を切り替えるための順次回路の順番は図146.で示されます。

図146. 命令語変更手順時順次回路状態遷移図



## AVR\_RESET (\$C)

AVRデバイスをリセット状態へ、またはリセット状態から開放するためのAVR固有公開JTAG命令です。この命令によって検査入出力ポート(TAP)制御器はリセットされません。1ビットのリセットレジスタがデータレジスタ(DR)として選ばれます。リセットチェーンに論理1がある限り、リセットが有効であることに注意してください。このチェーンからの出力はラッチされません。本命令が有効な状態を次に示します。

- DR移動 : リセットレジスタがTCKクロック入力によって移動されます。

## PROG\_ENABLE (\$4)

JTAGポート経由のプログラミングを許可するためのAVR固有公開JTAG命令です。16ビット長のプログラミング許可レジスタがデータレジスタ(DR)として選ばれます。本命令が有効な状態を次に示します。

- DR移動 : プログラミング許可識別子がデータレジスタに移動入力されます。
- DR更新 : プログラミング許可識別子が正規値と比較され、識別子が有効なら、プログラミング動作に移行されます。

## PROG\_COMMANDS (\$5)

JTAGポート経由のプログラミング命令移行用のAVR固有公開JTAG命令です。15ビット長のプログラミング命令レジスタがデータレジスタ(DR)として選ばれます。本命令が有効な状態を次に示します。

- DR捕獲 : 直前の命令の結果がデータレジスタに設定されます。
- DR移動 : データレジスタがTCK入力により移動され、直前の命令の結果を移動出力し、新規命令を移動入力します。
- DR更新 : プログラミング命令がフラッシュメモリ入力に適用されます。
- 検査実行/アイドル : 1つのクロック周期が生成され、適用された命令を実行します。



## PROG\_PAGELOAD (\$6)

JTAGポート経由でフラッシュメモリのページデータを直接設定するためのAVR固有公開JTAG命令です。2048ビット長の仮想フラッシュページ設定レジスタがデータレジスタ(DR)として選ばれます。これはフラッシュメモリ1ページ分のビット数と同じ長さの仮想走査チェーン(Scan chain)です。内部的な移動レジスタは8ビットです。多くのJTAG命令と異なり、移動レジスタからのデータ転送に**DR更新**状態が使われません。データは内部順次回路により、**DR移動**状態でフラッシュメモリページ緩衝部へバイト単位で自動的に転送されます。これは(この命令が)活性(有効)な場合だけです。

- **DR移動** : フラッシュメモリページデータがTCK入力によってTDIから移動入力され、バイト毎にフラッシュメモリページ緩衝部へ自動的に設定されます。

**注:** JTAG命令**PROG\_PAGELOAD**はAVRデバイスがJTAG走査チェーンの最初のデバイスの場合にだけ使えます。AVRが走査チェーンの最初のデバイスでない場合、バイト単位のプログラミング法が使われなければなりません。

## PROG\_PAGEREAD (\$7)

JTAGポート経由でフラッシュメモリの1ページの全データを取得するためのAVR固有公開JTAG命令です。2056ビット長の仮想フラッシュページ取得レジスタがデータレジスタ(DR)として選ばれます。これはフラッシュメモリ1ページ+8ビット分のビット数と同じ長さの仮想走査チェーン(Scan chain)です。内部的な移動レジスタは8ビットです。多くのJTAG命令と異なり、移動レジスタからのデータ転送に**DR捕獲**状態が使われません。データは内部順次回路により、**DR移動**状態でフラッシュメモリページ緩衝部からバイト単位で自動的に転送されます。これは(この命令が)活性(有効)な場合だけです。

- **DR移動** : フラッシュメモリのデータが自動的に(フラッシュメモリページ緩衝部から)バイト毎に読まれ、TCK入力によってTDOへ移動出力されます。TDI入力は無視されます。

**注:** JTAG命令**PROG\_PAGEREAD**はAVRデバイスがJTAG走査チェーンの最初のデバイスの場合にだけ使えます。AVRが走査チェーンの最初のデバイスでない場合、バイト単位のプログラミング法が使われなければなりません。

## データレジスタ

データレジスタ(DR)は199頁の「**プログラミング特殊JTAG命令**」項で記載されたJTAG命令レジスタ(IR)によって選ばれます。プログラミング操作に関連するデータレジスタを次に示します。

- リセット(Reset)レジスタ
- プログラミング許可(Programming Enable)レジスタ
- プログラミング命令(Programming Command)レジスタ
- 仮想フラッシュページ設定(Virtual Flash Page Load)レジスタ
- 仮想フラッシュページ取得(Virtual Flash Page Read)レジスタ

## リセット (Reset) レジスタ

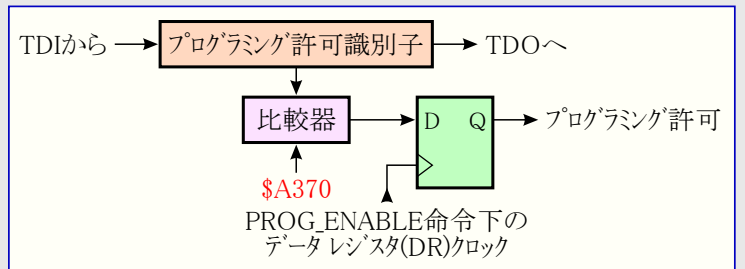
リセットレジスタはプログラミング中、デバイスをリセットするのに使われる検査データレジスタです。プログラミング動作への移行に先立ってデバイスをリセットするのに必要とされます。

リセットレジスタ内の値**1**は外部RESETを**Low**に引き込むことに相当します。リセットレジスタ内の値が**1**である限り、デバイスはリセットされます。リセットレジスタの開放後、デバイスは**クロック種別ヒューズ**設定に従ってリセット起動遅延時間(24頁の「**クロック元**」参照)分リセットを維持します。168頁の**図123**で示されるようにこのデータレジスタからの出力はラッチされず、それ故リセットが直ちに起こります。

## プログラミング許可 (Programming Enable) レジスタ

プログラミング許可レジスタは16ビット長のレジスタです。このレジスタの内容はプログラミング許可識別子**\$A370(1010 0011 0111 0000)**と比較されます。このレジスタの内容がプログラミング許可識別子と一致すると、JTAGポート経由のプログラミングが許可されます。このレジスタは電源ONリセットで**0**にリセットされ、プログラミング動作を抜ける時には常にリセットされるべきです。

図147. プログラミング許可レジスタ



## プログラミング命令 (Programming Command) レジスタ

プログラミング命令レジスタは15ビット長のレジスタです。このレジスタはプログラミング命令を連続的に移動入力し、直前のプログラミング命令の結果を連続的に移動出力するのに使われます。JTAGプログラミング命令一式は**表130**で示されます。プログラミング命令を移動入力する時の状態順は**図149**で図解されます。

図148. プログラミング命令レジスタ

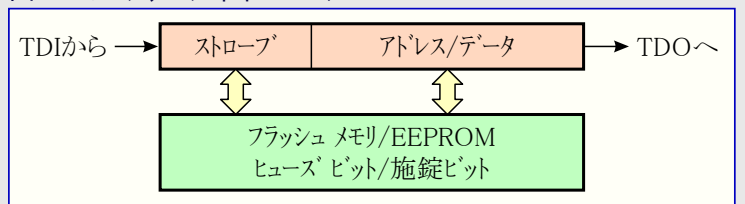


表130. JTAGのラミング命令一式 (手順)

| 命令                 |                | TDI入力              | TDO出力              | 備考                     |                        |
|--------------------|----------------|--------------------|--------------------|------------------------|------------------------|
| チップ消去              | 1 ① チップ消去      | 010 0011 1000 0000 | xxx xxxx xxxx xxxx |                        |                        |
|                    |                | 011 0001 1000 0000 | xxx xxxx xxxx xxxx |                        |                        |
|                    |                | 011 0011 1000 0000 | xxx xxxx xxxx xxxx |                        |                        |
|                    |                | 011 0011 1000 0000 | xxx xxxx xxxx xxxx |                        |                        |
| 1 ②                | チップ消去完了検査      | 011 0011 1000 0000 | xxx xxSx xxxx xxxx | (注2)                   |                        |
| フラッシュメモリ書き込み       | 2 ①            | フラッシュ書き込み移行        | 010 0011 0001 0000 | xxx xxxx xxxx xxxx     |                        |
|                    | 2 ②            | アドレス上位バイト設定        | 000 0111 HHHH HHHH | xxx xxxx xxxx xxxx     | (注10)                  |
|                    | 2 ③            | アドレス下位バイト設定        | 000 0011 LLLL LLLL | xxx xxxx xxxx xxxx     |                        |
|                    | 2 ④            | データ下位バイト設定         | 001 0011 WWWW WWWW | xxx xxxx xxxx xxxx     |                        |
|                    | 2 ⑤            | データ上位バイト設定         | 001 0111 WWWW WWWW | xxx xxxx xxxx xxxx     |                        |
|                    | 2 ⑥            | データラッチ             | 011 0111 0000 0000 | xxx xxxx xxxx xxxx     | (注1)                   |
|                    |                |                    | 111 0111 0000 0000 | xxx xxxx xxxx xxxx     |                        |
|                    |                |                    | 011 0111 0000 0000 | xxx xxxx xxxx xxxx     |                        |
|                    | 2 ⑦            | ページ書き込み            | 011 0111 0000 0000 | xxx xxxx xxxx xxxx     | (注1)                   |
|                    |                |                    | 011 0101 0000 0000 | xxx xxxx xxxx xxxx     |                        |
| 011 0111 0000 0000 |                |                    | xxx xxxx xxxx xxxx |                        |                        |
| 011 0111 0000 0000 |                |                    | xxx xxxx xxxx xxxx |                        |                        |
| 2 ⑧                | ページ書き込み完了検査    | 011 0111 0000 0000 | xxx xxSx xxxx xxxx | (注2)                   |                        |
| フラッシュ読み出し          | 3 ①            | フラッシュ読み出し移行        | 010 0011 0000 0010 | xxx xxxx xxxx xxxx     |                        |
|                    | 3 ②            | アドレス上位バイト設定        | 000 0111 HHHH HHHH | xxx xxxx xxxx xxxx     | (注10)                  |
|                    | 3 ③            | アドレス下位バイト設定        | 000 0011 LLLL LLLL | xxx xxxx xxxx xxxx     |                        |
|                    | 3 ④            | データ下位/上位バイト取得      | 011 0010 0000 0000 | xxx xxxx xxxx xxxx     |                        |
| 011 0110 0000 0000 |                |                    | xxx xxxx RRRR RRRR | 下位バイト                  |                        |
| 011 0111 0000 0000 |                |                    | xxx xxxx RRRR RRRR | 上位バイト                  |                        |
| EEPROM書き込み         | 4 ①            | EEPROM書き込み移行       | 010 0011 0001 0001 | xxx xxxx xxxx xxxx     |                        |
|                    | 4 ②            | アドレス上位バイト設定        | 000 0111 HHHH HHHH | xxx xxxx xxxx xxxx     | (注10)                  |
|                    | 4 ③            | アドレス下位バイト設定        | 000 0011 LLLL LLLL | xxx xxxx xxxx xxxx     |                        |
|                    | 4 ④            | データバイト設定           | 001 0011 WWWW WWWW | xxx xxxx xxxx xxxx     |                        |
|                    | 4 ⑤            | データラッチ             | 011 0111 0000 0000 | xxx xxxx xxxx xxxx     | (注1)                   |
|                    |                |                    | 111 0111 0000 0000 | xxx xxxx xxxx xxxx     |                        |
|                    |                |                    | 011 0111 0000 0000 | xxx xxxx xxxx xxxx     |                        |
|                    | 4 ⑥            | EEPROM書き込み         | 011 0011 0000 0000 | xxx xxxx xxxx xxxx     | (注1)                   |
| 011 0001 0000 0000 |                |                    | xxx xxxx xxxx xxxx |                        |                        |
| 011 0011 0000 0000 |                |                    | xxx xxxx xxxx xxxx |                        |                        |
| 011 0011 0000 0000 |                |                    | xxx xxxx xxxx xxxx |                        |                        |
| 4 ⑦                | EEPROM書き込み完了検査 | 011 0011 0000 0000 | xxx xxSx xxxx xxxx | (注2)                   |                        |
| EEPROM読み出し         | 5 ①            | EEPROM読み出し移行       | 010 0011 0000 0011 | xxx xxxx xxxx xxxx     |                        |
|                    | 5 ②            | アドレス上位バイト設定        | 000 0111 HHHH HHHH | xxx xxxx xxxx xxxx     | (注10)                  |
|                    | 5 ③            | アドレス下位バイト設定        | 000 0011 LLLL LLLL | xxx xxxx xxxx xxxx     |                        |
|                    | 5 ④            | EEPROM読み出し         | 011 0011 LLLL LLLL | xxx xxxx xxxx xxxx     |                        |
| 011 0010 0000 0000 |                |                    | xxx xxxx xxxx xxxx |                        |                        |
| 011 0011 0000 0000 |                |                    | xxx xxxx RRRR RRRR |                        |                        |
| ヒューズビット書き込み        | 6 ①            | ヒューズ書き込み移行         | 010 0011 0100 0000 | xxx xxxx xxxx xxxx     |                        |
|                    | 6 ②            | データ下位バイト設定         | 001 0011 WWWW WWWW | xxx xxxx xxxx xxxx     | (注3) ビット配置は188頁の表117参照 |
|                    | 6 ③            | 拡張ヒューズ書き込み         | 011 1011 0000 0000 | xxx xxxx xxxx xxxx     | (注1)                   |
|                    |                |                    | 011 1001 0000 0000 | xxx xxxx xxxx xxxx     |                        |
|                    |                |                    | 011 1011 0000 0000 | xxx xxxx xxxx xxxx     |                        |
|                    |                |                    | 011 1011 0000 0000 | xxx xxxx xxxx xxxx     |                        |
|                    | 6 ④            | 拡張ヒューズ書き込み完了検査     | 011 1011 0000 0000 | xxx xxSx xxxx xxxx     | (注2)                   |
|                    | 6 ⑤            | データ下位バイト設定         | 001 0011 WWWW WWWW | xxx xxxx xxxx xxxx     | (注3) ビット配置は188頁の表118参照 |
|                    | 6 ⑥            | 上位側ヒューズ書き込み        | 011 0111 0000 0000 | xxx xxxx xxxx xxxx     | (注1)                   |
|                    |                |                    | 011 0101 0000 0000 | xxx xxxx xxxx xxxx     |                        |
| 011 0111 0000 0000 |                |                    | xxx xxxx xxxx xxxx |                        |                        |
| 011 0111 0000 0000 |                |                    | xxx xxxx xxxx xxxx |                        |                        |
| 6 ⑦                | 上位ヒューズ書き込み完了検査 | 011 0111 0000 0000 | xxx xxSx xxxx xxxx | (注2)                   |                        |
| 6 ⑧                | データ下位バイト設定     | 001 0011 WWWW WWWW | xxx xxxx xxxx xxxx | (注3) ビット配置は188頁の表119参照 |                        |
| 6 ⑨                | 下位側ヒューズ書き込み    | 011 0011 0000 0000 | xxx xxxx xxxx xxxx | (注1)                   |                        |
|                    |                | 011 0001 0000 0000 | xxx xxxx xxxx xxxx |                        |                        |
|                    |                | 011 0011 0000 0000 | xxx xxxx xxxx xxxx |                        |                        |
| 6 ⑩                | 下位ヒューズ書き込み完了検査 | 011 0011 0000 0000 | xxx xxSx xxxx xxxx | (注2)                   |                        |

次頁へ続く





表130 (続き). JTAGプログラミング命令一式

| 命令                 |                     | TDI入力              | TDO出力              | 備考                     |
|--------------------|---------------------|--------------------|--------------------|------------------------|
| 施錠ビット書き込み          | 7① 施錠ビット書き込み移行      | 010 0011 0010 0000 | xxx xxxx xxxx xxxx |                        |
|                    | 7② データ下位バイト設定       | 001 0011 11WW WWW  | xxx xxxx xxxx xxxx | (注4) ビット配置は187頁の表115参照 |
|                    | 7③ 施錠ビット書き込み        | 011 0011 0000 0000 | xxx xxxx xxxx xxxx | (注1)                   |
|                    |                     | 011 0001 0000 0000 | xxx xxxx xxxx xxxx |                        |
|                    |                     | 011 0011 0000 0000 | xxx xxxx xxxx xxxx |                        |
| 7④ 施錠ビット書き込み完了検査   | 011 0011 0000 0000  | xxx xxSx xxxx xxxx | (注2)               |                        |
| ヒューズ/施錠ビット読み出し     | 8① ヒューズ/施錠ビット読み出し移行 | 010 0011 0000 0100 | xxx xxxx xxxx xxxx |                        |
|                    | 8② 拡張ヒューズ読み出し       | 011 1010 0000 0000 | xxx xxxx xxxx xxxx | (注5) ビット配置は188頁の表117参照 |
|                    |                     | 011 1011 0000 0000 | xxx xxxx RRRR RRRR |                        |
|                    | 8③ 上位側ヒューズ読み出し      | 011 1110 0000 0000 | xxx xxxx xxxx xxxx | (注5) ビット配置は188頁の表118参照 |
|                    |                     | 011 1111 0000 0000 | xxx xxxx RRRR RRRR |                        |
|                    | 8④ 下位側ヒューズ読み出し      | 011 0010 0000 0000 | xxx xxxx xxxx xxxx | (注5) ビット配置は188頁の表119参照 |
|                    |                     | 011 0011 0000 0000 | xxx xxxx RRRR RRRR |                        |
|                    | 8⑤ 施錠ビット読み出し        | 011 0110 0000 0000 | xxx xxxx xxxx xxxx | (注5) ビット配置は187頁の表115参照 |
|                    |                     | 011 0111 0000 0000 | xxx xxxx xxRR RRRR |                        |
|                    | 8⑥ ヒューズ/施錠ビット読み出し   | 011 1010 0000 0000 | xxx xxxx xxxx xxxx | (注5)                   |
| 011 1110 0000 0000 |                     | xxx xxxx RRRR RRRR | 拡張ヒューズ             |                        |
| 011 0010 0000 0000 |                     | xxx xxxx RRRR RRRR | 上位ヒューズ             |                        |
| 011 0110 0000 0000 |                     | xxx xxxx RRRR RRRR | 下位ヒューズ             |                        |
| 011 0111 0000 0000 |                     | xxx xxxx RRRR RRRR | 施錠ビット              |                        |
| 識票バイト              | 9① 識票バイト読み出し移行      | 010 0011 0000 1000 | xxx xxxx xxxx xxxx |                        |
|                    | 9② アドレスバイト設定        | 000 0011 LLLL LLLL | xxx xxxx xxxx xxxx |                        |
|                    | 9③ 識票バイト読み出し        | 011 0010 0000 0000 | xxx xxxx xxxx xxxx |                        |
| 011 0011 0000 0000 |                     | xxx xxxx RRRR RRRR |                    |                        |
| 校正バイト              | 10① 校正バイト読み出し移行     | 010 0011 0000 1000 | xxx xxxx xxxx xxxx |                        |
|                    | 10② アドレスバイト設定       | 000 0011 LLLL LLLL | xxx xxxx xxxx xxxx |                        |
|                    | 10③ 校正バイト読み出し       | 011 0110 0000 0000 | xxx xxxx xxxx xxxx |                        |
| 011 0111 0000 0000 |                     | xxx xxxx RRRR RRRR |                    |                        |
| 11① 無操作設定          | 010 0011 0000 0000  | xxx xxxx xxxx xxxx |                    |                        |
|                    | 011 0011 0000 0000  | xxx xxxx xxxx xxxx |                    |                        |

H = アドレス上位バイトのビット      L = アドレス下位バイトのビット      S = 動作完了フラグ  
R = 読み出しデータ (MCU出力)      W = 書き込みデータ (MCU入力)      x = 0か1 (無効/無意味)

注1 : この命令は直前の命令手順によって上位7ビットが正しく設定されている(通常の場合、必要ではありません)。

注2 : Sが1になるまで繰り返します。(待機)

注3 : 設定値0でプログラム、1で非プログラムです。

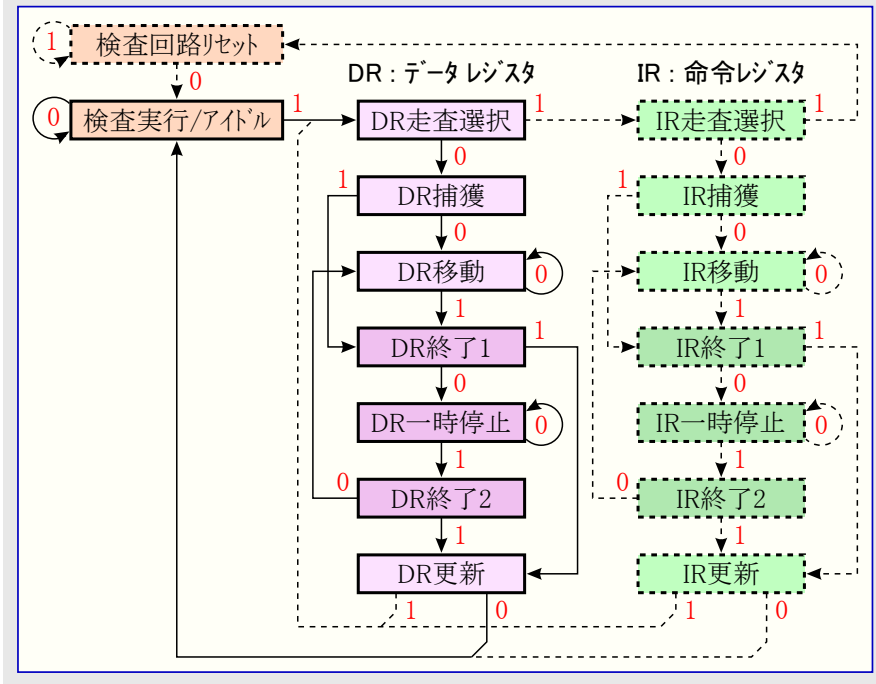
注4 : 設定値0でプログラム、1では施錠ビットが変化しません。

注5 : 読み出し値0でプログラム、1で非プログラムです。

(訳注) 原書での注6~9は表内備考欄に直接記載しました。

注10 : PCMSB(表124.参照)とEEAMSB(表125.参照)を越えるアドレスビットは無効です。

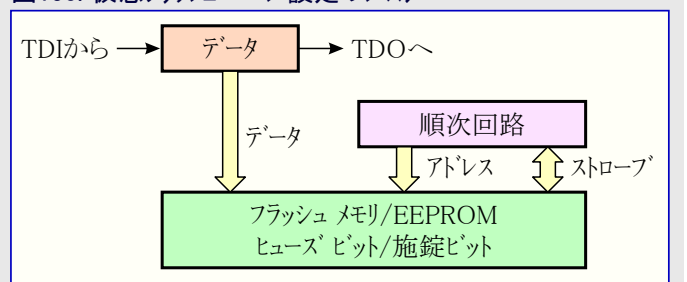
図149. テータ語変更/読み出し手順時順次回路状態遷移図



仮想フラッシュ ページ設定 (Virtual Flash Page Load) レジスタ

仮想フラッシュ ページ設定レジスタはフラッシュ メモリの1ページと等しいビット長の仮想走査チェーン(Scan chain)です。内部的な移動レジスタは8ビットで、データはバイト単位でフラッシュ メモリ ページ緩衝部へ自動的に転送されます。ページ内への全命令語(データ)の移動入力はページ内の最初のコードのLSBから始まり、ページ内の最終コードのMSBで終了します。これはページ書き込みを実行する前にフラッシュ メモリ ページ緩衝部全体を設定する効率的な方法を提供します。

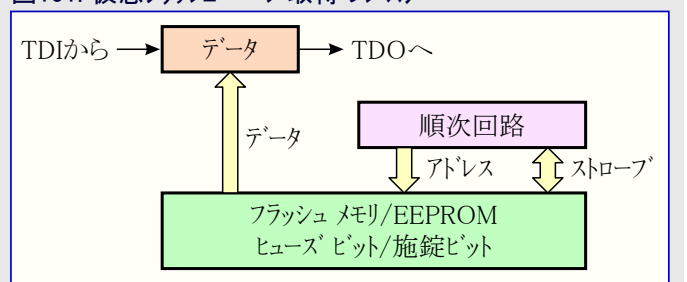
図150. 仮想フラッシュ ページ設定 レジスタ



仮想フラッシュ ページ取得 (Virtual Flash Page Read) レジスタ

仮想フラッシュ ページ取得レジスタはフラッシュ メモリの1ページと等しいビット数+8ビットの仮想走査チェーン(Scan chain)です。内部的な移動レジスタは8ビットで、データはフラッシュ メモリ ページ緩衝部からバイト単位で自動的に転送されます。最初の8クロックは先頭バイトを内部移動レジスタへ転送するのに使われ、この8クロック中に移動出力されたビットは無視されるべきです。この初期化に続いて、ページ内の最初のコードのLSBから始まり、ページ内の最後のコードのMSBで終了するデータが移動出力されます。これは書き込み確認のためにフラッシュ メモリの1ページ全体を読み出す効率的な方法を提供します。

図151. 仮想フラッシュ ページ取得 レジスタ



プログラミング手法

1①、1②形式のような以下の全ての参照は表130を参照してください。

プログラミング動作への移行

1. JTAG命令AVR\_RESETを入力し、リセットレジスタに1を移動します。
2. PROG\_ENABLE命令を入力し、プログラミング許可レジスタに\$A370(1010 0011 0111 0000)を移動します。

プログラミング動作からの抜け出し

1. JTAG命令PROG\_COMMANDSを入力します。
2. 無操作命令(11①参照)により、全てのプログラミング命令を禁止します。
3. PROG\_ENABLE命令を入力し、プログラミング許可レジスタに\$0000(0000 0000 0000 0000)を移動します。
4. JTAG命令AVR\_RESETを入力し、リセットレジスタに0を移動します。



## チップ消去の実行

1. JTAG命令**PROG\_COMMANDS**を入力します。
2. チップ消去命令(1①参照)を使い、チップ消去を開始します。
3. チップ消去完了検査(1②参照)を使い、完了までポーリングするか、**tWLRH\_CE**(196頁の表126.参照)時間待ちます。

## フラッシュメモリの書き込み

フラッシュメモリへ書き込む前にチップ消去が実行されなければなりません。上記の「**チップ消去の実行**」をご覧ください。

1. JTAG命令**PROG\_COMMANDS**を入力します。
2. フラッシュ書き込み移行命令(2①参照)を使い、フラッシュメモリ書き込みを許可します。
3. アドレス上位設定命令(2②参照)を使い、書き込みアドレス上位バイトを設定します。
4. アドレス下位設定命令(2③参照)を使い、書き込みアドレス下位バイトを設定します。
5. データ設定命令(2④, 2⑤, 2⑥参照)を使い、書き込みデータを設定します。
6. ページ内の全語(ワード)数分4.と5.を繰り返します。
7. ページ書き込み命令(2⑦参照)を使い、ページをフラッシュメモリに書き込みます。
8. ページ書き込み完了検査(2⑧参照)を使い、完了までポーリングするか、**tWLRH\_FLASH**(196頁の表126.参照)時間待ちます。
9. 全データ書き込みまで3.~8.を繰り返します。

より効率的なデータ転送は**PROG\_PAGeload**命令を使うことで達せられます。

1. JTAG命令**PROG\_COMMANDS**を入力します。
2. フラッシュ書き込み移行命令(2①参照)を使い、フラッシュメモリ書き込みを許可します。
3. アドレス設定命令(2②, 2③参照)を使い、ページアドレスを設定します。ページ内アドレスには**PCWORD**(189頁の表124.参照)が使われ、これらのビットは0として書かれなければなりません。
4. JTAG命令**PROG\_PAGeload**を入力します。
5. ページ内の最初のコードのLSBから始まって最後のコードのMSBで終了するページ内に全コードデータを移動してページ全体を設定します。
6. JTAG命令**PROG\_COMMANDS**を入力します。
7. ページ書き込み命令(2⑦参照)を使い、ページをフラッシュメモリに書き込みます。
8. ページ書き込み完了検査(2⑧参照)を使い、完了までポーリングするか、**tWLRH\_FLASH**(196頁の表126.参照)時間待ち機します。
9. 全データ書き込みまで3.~8.を繰り返します。

## フラッシュメモリの読み出し

1. JTAG命令**PROG\_COMMANDS**を入力します。
2. フラッシュ読み出し移行命令(3①参照)を使い、フラッシュメモリ読み出しを許可します。
3. アドレス設定命令(3②, 3③参照)を使い、読み出しアドレスを設定します。
4. データ読み出し命令(3④参照)を使い、データを読み出します。
5. 全データ読み出しまで3.~4.を繰り返します。

より効率的なデータ転送は**PROG\_PAGERead**命令を使うことで達せられます。

1. JTAG命令**PROG\_COMMANDS**を入力します。
2. フラッシュ読み出し移行命令(3①参照)を使い、フラッシュメモリ読み出しを許可します。
3. アドレス設定命令(3②, 3③参照)を使い、ページアドレスを設定します。ページ内アドレスには**PCWORD**(189頁の表124.参照)が使われ、これらのビットは0として書かれなければなりません。
4. JTAG命令**PROG\_PAGERead**を入力します。
5. ページ内の最初のコードのLSBから始まって最後のコードのMSBで終了するページ内の全コードデータを移動し、ページ全体を読み出します。最初に移動出力される8ビットが無視されるべきことに留意してください。
6. JTAG命令**PROG\_COMMANDS**を入力します。
7. 全データ読み出しまで3.~6.を繰り返します。

## EEPROMの書き込み

EEPROMへ書き込む前にチップ消去が実行されなければなりません。上記の「**チップ消去の実行**」をご覧ください。

1. JTAG命令**PROG\_COMMANDS**を入力します。
2. EEPROM書き込み移行命令(4①参照)を使い、EEPROMメモリ書き込みを許可します。
3. アドレス上位設定命令(4②参照)を使い、書き込みアドレス上位バイトを設定します。
4. アドレス下位設定命令(4③参照)を使い、書き込みアドレス下位バイトを設定します。
5. データ設定命令(4④, 4⑤参照)を使い、書き込みデータを設定します。
6. ページ内の全バイト数分4.と5.を繰り返します。
7. EEPROM書き込み命令(4⑥参照)を使い、データをEEPROMメモリに書き込みます。
8. EEPROM書き込み完了検査(4⑦参照)を使い、完了までポーリングするか、**tWLRH**(196頁の表126.参照)時間待ちます。
9. 全データ書き込みまで3.~8.を繰り返します。

**PROG\_PAGeload**命令がEEPROM書き込み時に使えないことに注意してください。

## EEPROMの読み出し

1. JTAG命令PROG\_COMMANDSを入力します。
2. EEPROM読み出し移行命令(5①参照)を使い、EEPROMメモリ読み出しを許可します。
3. アドレス設定命令(5②, 5③参照)を使い、読み出しアドレスを設定します。
4. データ読み出し命令(5④参照)を使い、データを読み出します。
5. 全データ読み出しまで3.~4.を繰り返します。

PROG\_PAGEREAD命令がEEPROM読み出し時に使えないことに注意してください。

## ヒューズビットの書き込み

1. JTAG命令PROG\_COMMANDSを入力します。
2. ヒューズ書き込み移行命令(6①参照)を使い、ヒューズ書き込みを許可します。
3. 拡張データ設定命令(6②参照)を使い、拡張ヒューズ値を設定します。(0=プログラム、1=非プログラム)
4. 拡張ヒューズ書き込み命令(6③参照)を使い、拡張ヒューズを書き込みます。
5. ヒューズ書き込み完了検査(6④参照)を使い、完了までポーリングするか、tWLRH(196頁の表126.参照)時間待ちます。
6. 上位データ設定命令(6⑤参照)を使い、上位側ヒューズ値を設定します。(0=プログラム、1=非プログラム)
7. 上位ヒューズ書き込み命令(6⑥参照)を使い、上位側ヒューズを書き込みます。
8. ヒューズ書き込み完了検査(6⑦参照)を使い、完了までポーリングするか、tWLRH(196頁の表126.参照)時間待ちます。
9. 下位データ設定命令(6⑧参照)を使い、下位側ヒューズ値を設定します。(0=プログラム、1=非プログラム)
10. 下位ヒューズ書き込み命令(6⑨参照)を使い、下位側ヒューズを書き込みます。
11. ヒューズ書き込み完了検査(6⑩参照)を使い、完了までポーリングするか、tWLRH(196頁の表126.参照)時間待ちます。

## 施錠ビットの書き込み

1. JTAG命令PROG\_COMMANDSを入力します。
2. 施錠ビット書き込み移行命令(7①参照)を使い、施錠ビット書き込みを許可します。
3. データ設定命令(7②参照)を使い、施錠ビット値を設定します。(0=プログラム、1=無変化)
4. 施錠ビット書き込み命令(7③参照)を使い、施錠ビットに書き込みます。
5. 施錠ビット書き込み完了検査(7④参照)を使い、完了までポーリングするか、tWLRH(196頁の表126.参照)時間待ちます。

## ヒューズ/施錠ビットの読み出し

1. JTAG命令PROG\_COMMANDSを入力します。
2. ヒューズ/施錠ビット読み出し移行命令(8①参照)を使い、ヒューズ/施錠ビット読み出しを許可します。
3. 全てのヒューズと施錠ビットを読むにはヒューズ/施錠ビット読み出し命令(8⑥参照)を使います。  
拡張ヒューズビットのみを読み出す場合は拡張ヒューズ読み出し命令(8②参照)を使います。  
上位側ヒューズビットだけを読むには上位ヒューズ読み出し命令(8③参照)を使います。  
下位側ヒューズビットだけを読むには下位ヒューズ読み出し命令(8④参照)を使います。  
施錠ビットだけを読むには施錠ビット読み出し命令(8⑤参照)を使います。

## 識票バイトの読み出し

1. JTAG命令PROG\_COMMANDSを入力します。
2. 識票読み出し移行命令(9①参照)を使い、識票読み出しを許可します。
3. アドレス設定命令(9②参照)を使い、読み出しアドレス\$00を設定します。
4. データ読み出し命令(9③参照)を使い、識票バイトを読み出します。
5. 第2、第3バイトを読むためにアドレスを\$01,\$02として各々3.~4.を繰り返します。

## 校正バイトの読み出し

1. JTAG命令PROG\_COMMANDSを入力します。
2. 校正バイト読み出し移行命令(10①参照)を使い、校正バイト読み出しを許可します。
3. アドレス設定命令(10②参照)を使い、読み出しアドレス\$00を設定します。
4. データ読み出し命令(10③参照)を使い、校正バイトを読み出します。

(訳注) 校正バイトは4バイトですので上記説明は不適切です。識票バイトと同様な手法が予測されます(未確認)。

## 電気的特性

### 絶対最大定格 (警告)

|                |                  |
|----------------|------------------|
| 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             |
| 入出力ピン出力電流      | 40.0mA           |
| 消費電流           | 200.0 ~ 400.0mA  |

(注意) 本データシート内で示された代表値はシミュレーションと同じ製法技術で製造された他のAVRマイクロコントローラの特性に基いています。最小と最大の値はデバイスの特性が記載された後に利用可能になります。

### (警告)

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す。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。

## DC特性

TA=-40°C~85°C, VCC=2.7V~5.5V (特記事項を除く)

| シンボル              | 項目                           | 条件                             | 最小           | 代表  | 最大          | 単位 |
|-------------------|------------------------------|--------------------------------|--------------|-----|-------------|----|
| V <sub>IL</sub>   | Lowレベル入力電圧 (XTAL1,RESETを除く)  | VCC=2.7~5.5V                   | -0.5         |     | 0.2VCC (注1) | V  |
| V <sub>IL1</sub>  | Lowレベル入力電圧 (XTAL1)           | VCC=2.7~5.5V                   | -0.5         |     | 0.1VCC (注1) |    |
| V <sub>IL2</sub>  | Lowレベル入力電圧 (RESET)           | VCC=2.7~5.5V                   | -0.5         |     | 0.2VCC (注1) |    |
| V <sub>IH</sub>   | Highレベル入力電圧 (XTAL1,RESETを除く) | VCC=2.7~5.5V                   | 0.6VCC (注2)  |     | VCC+0.5     |    |
| V <sub>IH1</sub>  | Highレベル入力電圧 (XTAL1)          | VCC=2.7~5.5V                   | 0.7VCC (注2)  |     | VCC+0.5     |    |
| V <sub>IH2</sub>  | Highレベル入力電圧 (RESET)          | VCC=2.7~5.5V                   | 0.85VCC (注2) |     | VCC+0.5     |    |
| V <sub>OL</sub>   | Lレベル出力電圧 (注3)                | IOL=20mA, VCC=5V               |              |     | 0.7         |    |
|                   |                              | IOL=10mA, VCC=3V               |              |     | 0.5         |    |
| V <sub>OH</sub>   | Hレベル出力電圧 (注4)                | IOH=-20mA, VCC=5V              | 4.2          |     |             |    |
|                   |                              | IOH=-10mA, VCC=3V              | 2.2          |     |             |    |
| I <sub>IL</sub>   | I/OピンLowレベル入力漏れ電流            | VCC=5.5V                       |              |     | 1           | μA |
| I <sub>IH</sub>   | I/OピンHighレベル入力漏れ電流           | 確実なH/L範囲                       |              |     | 1           |    |
| RRST              | RESETピンプルアップ抵抗               |                                | 30           |     | 85          | kΩ |
| R <sub>PEN</sub>  | PENピンプルアップ抵抗                 |                                | 30           |     | 60          |    |
| R <sub>PU</sub>   | I/Oピンプルアップ抵抗                 |                                | 20           |     | 50          |    |
|                   |                              |                                |              |     |             |    |
| I <sub>CC</sub>   | 活動動作消費電流                     | ATmega128L VCC=3V, 4MHz        |              | 5   | 5.5         | mA |
|                   |                              | ATmega128 VCC=5V, 8MHz         |              | 17  | 19          |    |
|                   | アイドル動作消費電流                   | ATmega128L VCC=3V, 4MHz        |              | 2   | 2.5         |    |
|                   |                              | ATmega128 VCC=5V, 8MHz         |              | 8   | 11          |    |
| パワーダウン動作消費電流      | VCC=3V, WDT有効                |                                | <15          | 25  | μA          |    |
|                   | VCC=3V, WDT禁止                |                                | <5           | 10  |             |    |
| V <sub>ACIO</sub> | アナログ比較器入力変位(オフセット)電圧         | VCC=5V, V <sub>in</sub> =VCC/2 |              |     | 40          | mV |
| I <sub>ACLK</sub> | アナログ比較器入力漏れ電流                |                                | -50          |     | 50          | nA |
| t <sub>ACPD</sub> | アナログ比較器伝播遅延時間                | VCC=2.7V                       |              | 750 |             | ns |
|                   |                              | VCC=5.0V                       |              | 500 |             |    |

注1: Lowレベルの認識が保証される最高電圧です。

注2: Highレベルの認識が保証される最低電圧です。

(注3)~(注4)は次頁を参照してください。



**注3:** 各I/Oポートは安定状態(非過渡時)に於いて検査条件(VCC=5Vで20mA、VCC=3V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、次の条件を厳守しなければなりません。

1. 全ポートのIOLの合計が400mAを超えるべきではありません。
2. ポートA0~A7、C3~C7、G2のIOLの合計が100mAを超えるべきではありません。
3. ポートB0~B7、E0~E7、G3~G4のIOLの合計が100mAを超えるべきではありません。
4. ポートC0~C2、D0~D7、G0~G1とXTAL2のIOLの合計が100mAを超えるべきではありません。
5. ポートF0~F7のIOLの合計が100mA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も仕様書での値を超えます。表の検査条件よりも大きな吸い込み電流を流すことは保証されません。

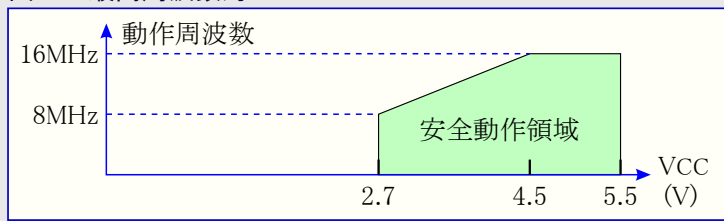
**注4:** 各I/Oポートは安定状態(非過渡時)に於いて検査条件(VCC=5Vで20mA、VCC=3Vで10mA)よりも多くの吐き出し電流を流すことができますが、次の条件を厳守しなければなりません。

1. 全ポートのIOHの合計が400mAを超えるべきではありません。
2. ポートA0~A7、C3~C7、G2のIOHの合計が100mAを超えるべきではありません。
3. ポートB0~B7、E0~E7、G3~G4のIOHの合計が100mAを超えるべきではありません。
4. ポートC0~C2、D0~D7、G0~G1とXTAL2のIOHの合計が100mAを超えるべきではありません。
5. ポートF0~F7のIOHの合計が100mA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も仕様書での値を超えます。表の検査条件よりも大きな吐き出し電流を流すことは保証されません。

## 速度勾配

図152. 最高周波数対VCC



## 外部クロック特性

図153. 外部クロック駆動波形

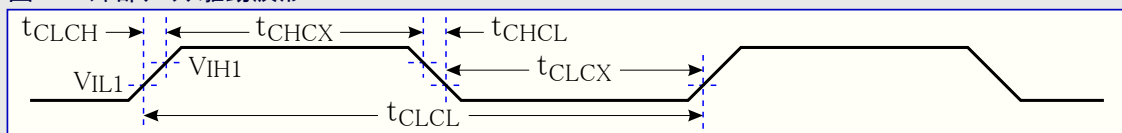


表131. 外部クロック特性

| シンボル              | 項目            | VCC=2.7~5.5V |     | VCC=4.5~5.5V |     | 単位  |
|-------------------|---------------|--------------|-----|--------------|-----|-----|
|                   |               | 最小           | 最大  | 最小           | 最大  |     |
| $1/t_{CLCL}$      | クロック周波数       | 0            | 8   | 0            | 16  | MHz |
| $t_{CLCL}$        | クロック周期        | 125          |     | 62.5         |     |     |
| $t_{CHCX}$        | Highレベル時間     | 50           |     | 25           |     | ns  |
| $t_{CLCX}$        | Lowレベル時間      | 50           |     | 25           |     |     |
| $t_{CLCH}$        | 上昇時間          |              | 1.6 |              | 0.5 | μs  |
| $t_{CHCL}$        | 下降時間          |              | 1.6 |              | 0.5 |     |
| $\Delta t_{CLCL}$ | 隣接クロック周期間の変化率 |              | 2   |              | 2   | %   |

**注:** 詳細については27頁の「外部クロック信号」を参照してください。

表132. 代表的な外部RC発振周波数 (VCC=5V)

| 周波数 f  | 抵抗 R (kΩ) | 容量 C (pF) |
|--------|-----------|-----------|
| 650kHz | 33        | 22        |
| 2.0MHz | 10        | 22        |

**注:** Rの範囲は3~100kΩ、Cは最小20pFとすべきです。表で与えられたCの値はピン容量を含みます。これは外圍器形状や基板配置で変化します。

## 2線直列インターフェース特性

表133.は2線直列バスに接続した装置に対する必要条件を記述します。ATmega128の2線直列インターフェースは記載条件下で、これらの必要条件を越えるか、または合致します。

図154. 2線直列バス タイミング

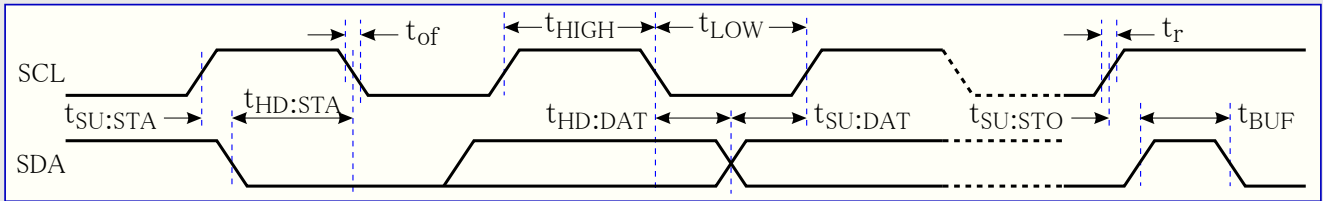


表133. 2線直列インターフェース必要条件

| シンボル         | 項目  | 条件                                    | 最小                  | 最大           | 単位      |          |
|--------------|---|---------------------------------------|---------------------|--------------|---------|----------|
| $V_{IL}$     | Lowレベル入力電圧                                  |                                       | -0.5                | $0.3V_{CC}$  | V       |          |
| $V_{IH}$     | Highレベル入力電圧                                 |                                       | $0.7V_{CC}$         | $V_{CC}+0.5$ |         |          |
| $V_{hys}$ ①  | シュミットトリガ入力ヒステリシス電圧                          |                                       | ② $0.05V_{CC}$      |              |         |          |
| $V_{OL}$ ①   | Lowレベル出力電圧                                  | $I_{OL}=3mA$                          | 0                   | 0.4          | ns      |          |
| $t_r$ ①      | 出力上昇時間( $V_{ILmin} \rightarrow V_{IHmax}$ ) |                                       | ② $20+0.1C_b$ ③     | 300          |         |          |
| $t_{of}$ ①   | 出力下降時間( $V_{IHmin} \rightarrow V_{ILmax}$ ) | $10pF < C_b < 400pF$                  | ② $20+0.1C_b$ ③     | 250          |         |          |
| $t_{SP}$ ①   | 入力パルス最小幅(尖頭消去濾波)                            |                                       | 0                   | ② 50         | $\mu A$ |          |
| $I_i$        | 入力電流(ピン単位)                                  | $0.1V_{CC} < V_i < 0.9V_{CC}$         | -10                 | 10           |         |          |
| $C_i$ ①      | ピン入力容量                                      |                                       |                     | 10           |         |          |
| $f_{SCL}$    | SCLクロック周波数                                  | ④⑤ $f_{CK} > \max(16f_{SCL}, 250kHz)$ | 0                   | 400          | kHz     |          |
| $R_p$        | プルアップ抵抗値                                    | $f_{SCL} \leq 100kHz$                 | $(V_{CC}-0.4V)/3mA$ | $1000ns/C_b$ |         | $\Omega$ |
|              |   | $f_{SCL} > 100kHz$                    | $(V_{CC}-0.4V)/3mA$ | $300ns/C_b$  |         |          |
| $t_{HD:STA}$ | (再送)開始条件保持時間                                | $f_{SCL} \leq 100kHz$                 | 4.0                 |              | $\mu s$ |          |
|              |   | $f_{SCL} > 100kHz$                    | 0.6                 |              |         |          |
| $t_{LOW}$    | SCLクロックLowレベル時間                             | $f_{SCL} \leq 100kHz$                 | 4.7                 |              |         |          |
|              |   | $f_{SCL} > 100kHz$                    | 1.3                 |              |         |          |
| $t_{HIGH}$   | SCLクロックHighレベル時間                            | $f_{SCL} \leq 100kHz$                 | 4.0                 |              | $\mu s$ |          |
|              |   | $f_{SCL} > 100kHz$                    | 0.6                 |              |         |          |
| $t_{SU:STA}$ | 再送開始条件準備時間                                  | $f_{SCL} \leq 100kHz$                 | 4.7                 |              | $\mu s$ |          |
|              |   | $f_{SCL} > 100kHz$                    | 0.6                 |              |         |          |
| $t_{HD:DAT}$ | データ保持時間                                     | $f_{SCL} \leq 100kHz$                 | 0                   | 3.45         | ns      |          |
|              |   | $f_{SCL} > 100kHz$                    | 0                   | 0.9          |         |          |
| $t_{SU:DAT}$ | データ準備時間                                     | $f_{SCL} \leq 100kHz$                 | 250                 |              | ns      |          |
|              |   | $f_{SCL} > 100kHz$                    | 100                 |              |         |          |
| $t_{SU:STO}$ | 停止条件準備時間                                    | $f_{SCL} \leq 100kHz$                 | 4.0                 |              | $\mu s$ |          |
|              |   | $f_{SCL} > 100kHz$                    | 0.6                 |              |         |          |
| $t_{BUF}$    | 停止条件→開始条件間バス開放時間                            | $f_{SCL} \leq 100kHz$                 | 4.7                 |              | $\mu s$ |          |
|              |   | $f_{SCL} > 100kHz$                    | 1.3                 |              |         |          |

① ATmega128で、この項目は特性が記載されていますが、100%検査はされていません。

②  $f_{SCL} > 100kHz$ についてのみ必要とされます。

③  $C_b$ は1つのバス信号線の容量(pF)です。

④  $f_{CK}$ はCPU(システム)クロック周波数です。

⑤ この必要条件はAtmel® AVR® ATmega128の全ての2線直列インターフェース動作に適用します。2線直列バスに接続した他の装置は一般的な $f_{SCL}$ 必要条件に従うことだけを必要とします。

## SPIタイミング特性

図155. SPI タイミング必要条件 (主装置動作)

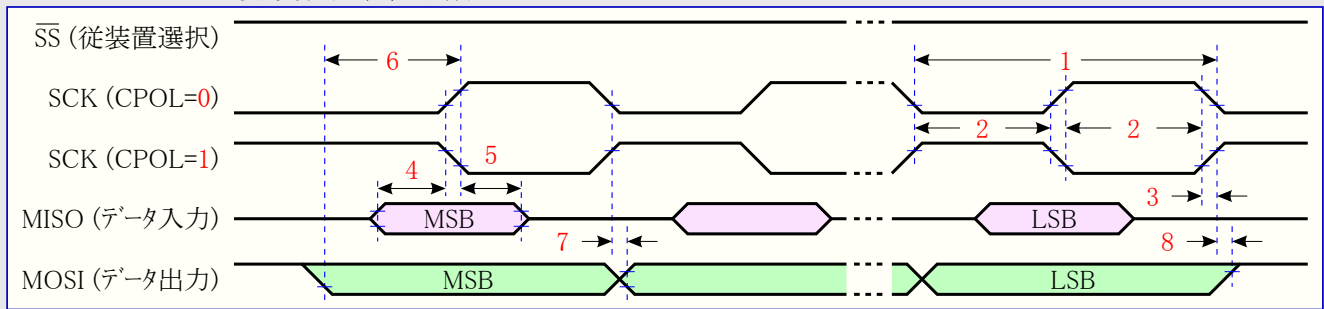


図156. SPI タイミング必要条件 (従装置動作)

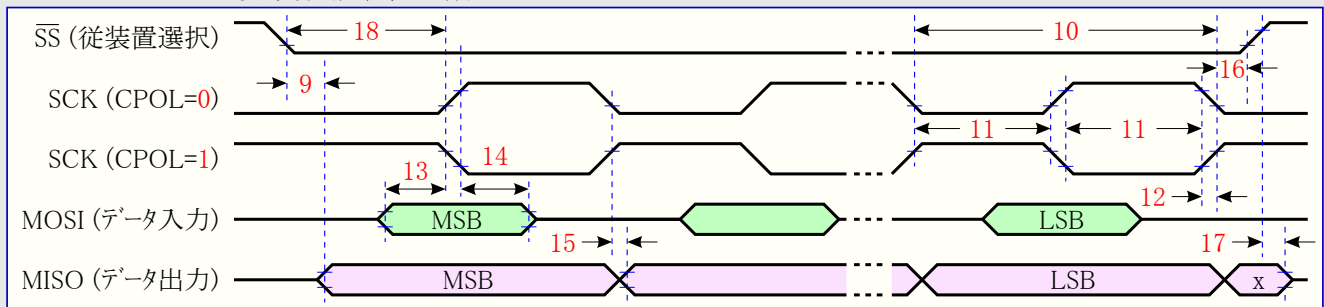


表134. SPI タイミング特性

| 番号 | 項目                 | 動作種別 | 最小               | 代表                  | 最大  | 単位 |
|----|--------------------|------|------------------|---------------------|-----|----|
| 1  | SCK周期              | 主装置  |                  | 表72.参照              |     | ns |
| 2  | SCK High/Low期間     | 主装置  |                  | 50%デューティ比           |     |    |
| 3  | SCK上昇/下降時間         | 主装置  |                  | 3.6                 |     |    |
| 4  | 入力データ 準備時間         | 主装置  |                  | 10                  |     |    |
| 5  | 入力データ 保持時間         | 主装置  |                  | 10                  |     |    |
| 6  | 出力からSCK変移時間        | 主装置  |                  | 0.5t <sub>SCK</sub> |     |    |
| 7  | SCKからの出力遅延時間       | 主装置  |                  | 10                  |     |    |
| 8  | SCKからのHigh出力時間     | 主装置  |                  | 10                  |     |    |
| 9  | SS ↓からの出力遅延時間      | 従装置  |                  | 15                  |     | μs |
| 10 | SCK周期              | 従装置  | 4t <sub>CK</sub> |                     |     |    |
| 11 | SCK High/Low期間 (注) | 従装置  | 2t <sub>CK</sub> |                     |     | ns |
| 12 | SCK上昇/下降時間         | 従装置  |                  |                     | 1.6 |    |
| 13 | 入力データ 準備時間         | 従装置  | 10               |                     |     |    |
| 14 | 入力データ 保持時間         | 従装置  | 10               |                     |     |    |
| 15 | SCKからの出力遅延時間       | 従装置  |                  | 15                  |     |    |
| 16 | SCKからのSS ↑遅延時間     | 従装置  | 20               |                     |     |    |
| 17 | SS ↑からの出力Hi-Z遅延時間  | 従装置  |                  | 10                  |     |    |
| 18 | SS ↓からのSCK遅延時間     | 従装置  | 2t <sub>CK</sub> |                     |     |    |

注: SPIプログラミングでの最小SCK High/Low期間は、2t<sub>CLCL</sub>(t<sub>CK</sub> < 12MHz)、3t<sub>CLCL</sub>(t<sub>CK</sub> ≥ 12MHz)です。

## A/D変換器特性

表135. A/D変換特性

| シンボル                    | 項目  | 条件                                    | 最小       | 代表          | 最大       | 単位          |     |
|-------------------------|---|---------------------------------------|----------|-------------|----------|-------------|-----|
| シングル<br>エンド<br>入力<br>変換 | 分解能   |                                       |          |             | 10       | ビット         |     |
|                         | 絶対精度<br>(INL,DNL,利得,<br>変位(オフセット),量子化<br>誤差を含む) | VCC=4V 変換クロック=200kHz                  |          | 1.5         |          |             | LSB |
|                         |   | VREF=4V 変換クロック=1MHz                   |          | 3.25        |          |             |     |
|                         |   | 雑音低減動作 変換クロック=200kHz                  |          | 1.5         |          |             |     |
|                         |   | 雑音低減動作 変換クロック=1MHz                    |          | 3.75        |          |             |     |
|                         | 積分非直線性誤差(INL)                                   | VCC=4V<br>VREF=4V<br>変換クロック=200kHz    |          |             | 0.75     |             | LSB |
|                         | 微分非直線性誤差(DNL)                                   |                                       |          |             | 0.5      |             |     |
|                         | 変位(オフセット)(ゼロ)誤差                                 |                                       |          |             | 1        |             |     |
|                         | 利得誤差  |                                       |          |             | 1        |             |     |
|                         | 変換クロック周波数                                       |                                       |          | 50          |          | 1000        | kHz |
|                         | 変換時間  | 連続変換動作                                |          | 13          |          | 260         | μs  |
|                         | VREF  | 基準電圧                                  |          | 2.0         |          | AVCC        | V   |
|                         | VIN   | 入力電圧                                  |          | GND         |          | VREF        |     |
|                         |   | A/D変換出力                               |          | 0           |          | 1023        | LSB |
|                         | 入力周波数帯域   |                                       |          | 38.5        |          | kHz         |     |
| 差動<br>入力<br>変換          | 分解能   |                                       | ×1       |             |          | 10          | ビット |
|                         |   |                                       | ×10      |             |          | 10          |     |
|                         |   |                                       | ×200     |             |          | 10          |     |
|                         | 絶対精度 (変位(オフセット),<br>利得誤差校正後)                    |                                       | ×1       |             | 17       |             | LSB |
|                         |   |                                       | ×10      |             | 17       |             |     |
|                         |   |                                       | ×200     |             | 7        |             |     |
|                         | 積分非直線性誤差(INL)                                   | VCC=5V<br>VREF=4V<br>変換クロック=50~200kHz | ×1       |             | 1.5      |             | LSB |
|                         |   |                                       | ×10      |             | 2        |             |     |
|                         |   |                                       | ×200     |             | 5        |             |     |
|                         | 変位(オフセット)(ゼロ)誤差                                 |                                       | ×1       |             | 2        |             | LSB |
|                         |   |                                       | ×10      |             | 3        |             |     |
|                         |   |                                       | ×200     |             | 4        |             |     |
|                         | 利得誤差  |                                       | ×1       |             | 1.5      |             | %   |
|                         |   |                                       | ×10      |             | 1.5      |             |     |
| ×200                    |   |                                       |          | 0.5         |          |             |     |
| 変換クロック周波数               |   |                                       | 50       |             | 200      | kHz         |     |
| 変換時間                    |   |                                       | 65       |             | 260      | μs          |     |
| VREF                    | 基準電圧  |                                       | 2.0      |             | AVCC-0.5 | V           |     |
| VIN                     | 入力電圧  |                                       | GND      |             | VCC      |             |     |
| VDIFF                   | 差動入力電圧差   |                                       | -VREF/利得 |             | VREF/利得  |             |     |
|                         | A/D変換出力   |                                       | -511     |             | 511      | LSB         |     |
|                         | 入力周波数帯域   |                                       |          | 4           |          | kHz         |     |
| 共通                      | AVCC  | アナログ供給電圧                              |          | VCC-0.3(注1) |          | VCC+0.3(注2) | V   |
|                         | VINT  | 内蔵2.56V基準電圧                           |          | 2.3         | 2.56     | 2.7         |     |
|                         | RREF  | 基準電圧入力インピーダンス                         |          |             | 32       |             | kΩ  |
|                         | RAIN  | アナログ入力インピーダンス                         |          | 55          | 100      |             | MΩ  |

注1: AVCCの最小値は2.7Vです。

注2: AVCCの最大値は5.5Vです。

(訳注) 原書の表135.と表136.は表135.として統合しました。

## 外部メモリ タイミング特性

表137. 外部データメモリ特性 (VCC=4.5V~5.5V、待ちなし)

| シンボル | 項目       | 8MHz時  |      | 一般式         |             |   | 単位  |
|------|----------|--------|------|-------------|-------------|---|-----|
|      |          | 最小     | 最大   | 最小          | 最大          | 注 |     |
| 0    | 1/tCLCL  | 発振器周波数 |      | 0.0         | 16          |   | MHz |
| 1    | tLHLL    | 115    |      | 1.0tCLCL-10 |             |   | ns  |
| 2    | tAVLL    | 57.5   |      | 0.5tCLCL-5  |             | 1 |     |
| 3A   | tLLAX,ST | 5      |      | 5           |             |   |     |
| 3B   | tLLAX,LD | 5      |      | 5           |             |   |     |
| 4    | tAVLLC   | 57.5   |      | 0.5tCLCL-5  |             | 1 |     |
| 5    | tAVRL    | 115    |      | 1.0tCLCL-10 |             |   |     |
| 6    | tAVWL    | 115    |      | 1.0tCLCL-10 |             |   |     |
| 7    | tLLWL    | 47.5   | 67.5 | 0.5tCLCL-15 | 0.5tCLCL+5  | 2 |     |
| 8    | tLLRL    | 47.5   | 67.5 | 0.5tCLCL-15 | 0.5tCLCL+5  | 2 |     |
| 9    | tDVRH    | 40     |      | 40          |             |   |     |
| 10   | tRLDV    |        | 75   |             | 1.0tCLCL-50 |   |     |
| 11   | tRHDX    | 0      |      | 0           |             |   |     |
| 12   | tRLRH    | 115    |      | 1.0tCLCL-10 |             |   |     |
| 13   | tOVWL    | 42.5   |      | 0.5tCLCL-20 |             | 1 |     |
| 14   | tWHDX    | 115    |      | 1.0tCLCL-10 |             |   |     |
| 15   | tDVWH    | 125    |      | 1.0tCLCL    |             |   |     |
| 16   | tWLWH    | 115    |      | 1.0tCLCL-10 |             |   |     |

注1: 一般式の定数はデューティサイクル=50%(XTAL1の外部クロックのHigh時間は半周期)と仮定した値です。

注2: 一般式の定数はデューティサイクル=50%(XTAL1の外部クロックのLow時間は半周期)と仮定した値です。

表138. 外部データメモリ特性 (VCC=4.5V~5.5V、SRWn1=0, SRWn0=1 (1待ち周期))

| シンボル | 項目      | 8MHz時  |     | 一般式         |             |   | 単位  |
|------|---------|--------|-----|-------------|-------------|---|-----|
|      |         | 最小     | 最大  | 最小          | 最大          | 注 |     |
| 0    | 1/tCLCL | 発振器周波数 |     | 0.0         | 16          |   | MHz |
| 10   | tRLDV   |        | 200 |             | 2.0tCLCL-50 |   | ns  |
| 12   | tRLRH   | 240    |     | 2.0tCLCL-10 |             |   |     |
| 15   | tDVWH   | 250    |     | 2.0tCLCL    |             |   |     |
| 16   | tWLWH   | 240    |     | 2.0tCLCL-10 |             |   |     |

表139. 外部データメモリ特性 (VCC=4.5V~5.5V、SRWn1=1, SRWn0=0 (2待ち周期))

| シンボル | 項目      | 8MHz時  |     | 一般式         |             |   | 単位  |
|------|---------|--------|-----|-------------|-------------|---|-----|
|      |         | 最小     | 最大  | 最小          | 最大          | 注 |     |
| 0    | 1/tCLCL | 発振器周波数 |     | 0.0         | 16          |   | MHz |
| 10   | tRLDV   |        | 325 |             | 3.0tCLCL-50 |   | ns  |
| 12   | tRLRH   | 365    |     | 3.0tCLCL-10 |             |   |     |
| 15   | tDVWH   | 375    |     | 3.0tCLCL    |             |   |     |
| 16   | tWLWH   | 365    |     | 3.0tCLCL-10 |             |   |     |

表140. 外部データメモリ特性 (VCC=4.5V~5.5V、SRWn1=1, SRWn0=1 (2&1待ち周期))

| シンボル | 項目      | 8MHz時  |     | 一般式         |             |   | 単位  |
|------|---------|--------|-----|-------------|-------------|---|-----|
|      |         | 最小     | 最大  | 最小          | 最大          | 注 |     |
| 0    | 1/tCLCL | 発振器周波数 |     | 0.0         | 16          |   | MHz |
| 10   | tRLDV   |        | 325 |             | 3.0tCLCL-50 |   | ns  |
| 12   | tRLRH   | 365    |     | 3.0tCLCL-10 |             |   |     |
| 14   | tWHDX   | 240    |     | 2.0tCLCL-10 |             |   |     |
| 15   | tDVWH   | 375    |     | 3.0tCLCL    |             |   |     |
| 16   | tWLWH   | 365    |     | 3.0tCLCL-10 |             |   |     |





表141. 外部データメモリ特性 (VCC=2.7V~5.5V、待ちなし)

|    | シンボル     | 項目                    | 4MHz時 |     | 一般式         |             |   | 単位  |
|----|----------|-----------------------|-------|-----|-------------|-------------|---|-----|
|    |          |                       | 最小    | 最大  | 最小          | 最大          | 注 |     |
| 0  | 1/tCLCL  | 発振器周波数                |       |     | 0.0         | 8           |   | MHz |
| 1  | tLHLL    | ALE Highパルス幅          | 235   |     | 1.0tCLCL-15 |             |   | ns  |
| 2  | tAVLL    | ALE↓前 下位アドレス準備時間      | 115   |     | 0.5tCLCL-10 |             | 1 |     |
| 3A | tLLAX,ST | ライト時 ALE↓後 下位アドレス保持時間 | 5     |     | 5           |             |   |     |
| 3B | tLLAX,LD | リード時 ALE↓後 下位アドレス保持時間 | 5     |     | 5           |             |   |     |
| 4  | tAVLLC   | ALE↓前 上位アドレス準備時間      | 115   |     | 0.5tCLCL-10 |             | 1 |     |
| 5  | tAVRL    | リード時RD↓前 下位アドレス有効時間   | 235   |     | 1.0tCLCL-15 |             |   |     |
| 6  | tAVWL    | ライト時WR↓前 下位アドレス有効時間   | 235   |     | 1.0tCLCL-15 |             |   |     |
| 7  | tLLWL    | ALE↓後 WR↓遅延時間         | 115   | 130 | 0.5tCLCL-10 | 0.5tCLCL+5  | 2 |     |
| 8  | tLLRL    | ALE↓後 RD↓遅延時間         | 115   | 130 | 0.5tCLCL-10 | 0.5tCLCL+5  | 2 |     |
| 9  | tDVRH    | RD↑前 データ準備時間          | 45    |     | 45          |             |   |     |
| 10 | tRLDV    | RD↓後 データ出力遅延時間        |       | 190 |             | 1.0tCLCL-60 |   |     |
| 11 | tRHDX    | RD↑後 データ保持時間          | 0     |     | 0           |             |   |     |
| 12 | tRLRH    | RD Lowパルス幅            | 235   |     | 1.0tCLCL-15 |             |   |     |
| 13 | tOVWL    | WR↓前 データ準備時間          | 105   |     | 0.5tCLCL-20 |             | 1 |     |
| 14 | tWHDX    | WR↑後 データ保持時間          | 235   |     | 1.0tCLCL-15 |             |   |     |
| 15 | tDVWH    | WR↑前 データ有効時間          | 250   |     | 1.0tCLCL    |             |   |     |
| 16 | tWLWH    | WR Lowパルス幅            | 235   |     | 1.0tCLCL-15 |             |   |     |

注1: 一般式の定数はデューティサイクル=50%(XTAL1の外部クロックのHigh時間は半周期)と仮定した値です。

注2: 一般式の定数はデューティサイクル=50%(XTAL1の外部クロックのLow時間は半周期)と仮定した値です。

表142. 外部データメモリ特性 (VCC=2.7V~5.5V、SRWn1=0, SRWn0=1 (1待ち周期))

|    | シンボル    | 項目             | 4MHz時 |     | 一般式         |             |   | 単位  |
|----|---------|----------------|-------|-----|-------------|-------------|---|-----|
|    |         |                | 最小    | 最大  | 最小          | 最大          | 注 |     |
| 0  | 1/tCLCL | 発振器周波数         |       |     | 0.0         | 8           |   | MHz |
| 10 | tRLDV   | RD↓後 データ出力遅延時間 |       | 440 |             | 2.0tCLCL-60 |   | ns  |
| 12 | tRLRH   | RD Lowパルス幅     | 485   |     | 2.0tCLCL-15 |             |   |     |
| 15 | tDVWH   | WR↑前 データ有効時間   | 500   |     | 2.0tCLCL    |             |   |     |
| 16 | tWLWH   | WR Lowパルス幅     | 485   |     | 2.0tCLCL-15 |             |   |     |

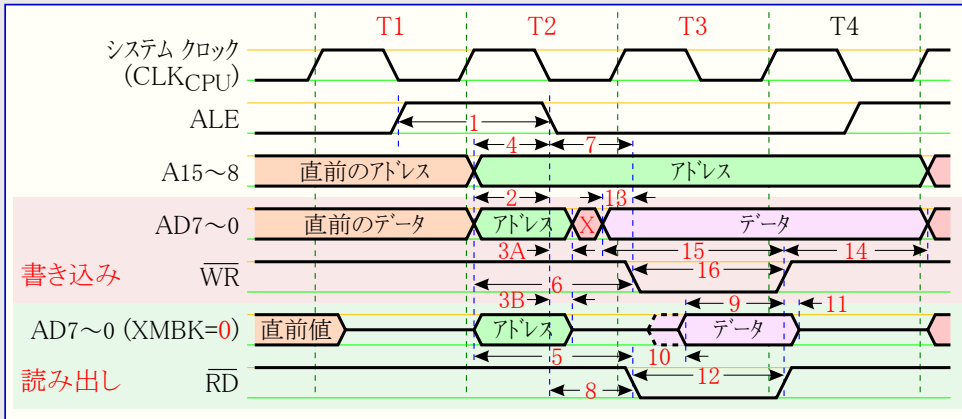
表143. 外部データメモリ特性 (VCC=2.7V~5.5V、SRWn1=1, SRWn0=0 (2待ち周期))

|    | シンボル    | 項目             | 4MHz時 |     | 一般式         |             |   | 単位  |
|----|---------|----------------|-------|-----|-------------|-------------|---|-----|
|    |         |                | 最小    | 最大  | 最小          | 最大          | 注 |     |
| 0  | 1/tCLCL | 発振器周波数         |       |     | 0.0         | 8           |   | MHz |
| 10 | tRLDV   | RD↓後 データ出力遅延時間 |       | 690 |             | 3.0tCLCL-60 |   | ns  |
| 12 | tRLRH   | RD Lowパルス幅     | 735   |     | 3.0tCLCL-15 |             |   |     |
| 15 | tDVWH   | WR↑前 データ有効時間   | 750   |     | 3.0tCLCL    |             |   |     |
| 16 | tWLWH   | WR Lowパルス幅     | 735   |     | 3.0tCLCL-15 |             |   |     |

表144. 外部データメモリ特性 (VCC=2.7V~5.5V、SRWn1=1, SRWn0=1 (2&1待ち周期))

|    | シンボル    | 項目             | 4MHz時 |     | 一般式         |             |   | 単位  |
|----|---------|----------------|-------|-----|-------------|-------------|---|-----|
|    |         |                | 最小    | 最大  | 最小          | 最大          | 注 |     |
| 0  | 1/tCLCL | 発振器周波数         |       |     | 0.0         | 8           |   | MHz |
| 10 | tRLDV   | RD↓後 データ出力遅延時間 |       | 690 |             | 3.0tCLCL-60 |   | ns  |
| 12 | tRLRH   | RD Lowパルス幅     | 735   |     | 3.0tCLCL-15 |             |   |     |
| 14 | tWHDX   | WR↑後 データ保持時間   | 485   |     | 2.0tCLCL-15 |             |   |     |
| 15 | tDVWH   | WR↑前 データ有効時間   | 750   |     | 3.0tCLCL    |             |   |     |
| 16 | tWLWH   | WR Lowパルス幅     | 735   |     | 3.0tCLCL-15 |             |   |     |

図157. 待ち状態なし外部データメモリアクセス周期 (SRWn1=0, SRWn0=0) [T1~T3]



**共通注意事項**

SRWn1はSRW11(上位領域)またはSRW01(下位領域)、SRWn0はSRW10(上位領域)またはSRW00(下位領域)です。

[T1~Tn]は図での命令実行周期範囲です。

Xは値変更区間を示します。

最終周期のALEは次命令がRAM(内部または外部)をアクセスする場合だけ存在します。また、最後の周期のアドレスとデータの変更区間の有無も同様です。

図158. 1待ち状態 外部データメモリアクセス周期 (SRWn1=0, SRWn0=1) [T1~T4]

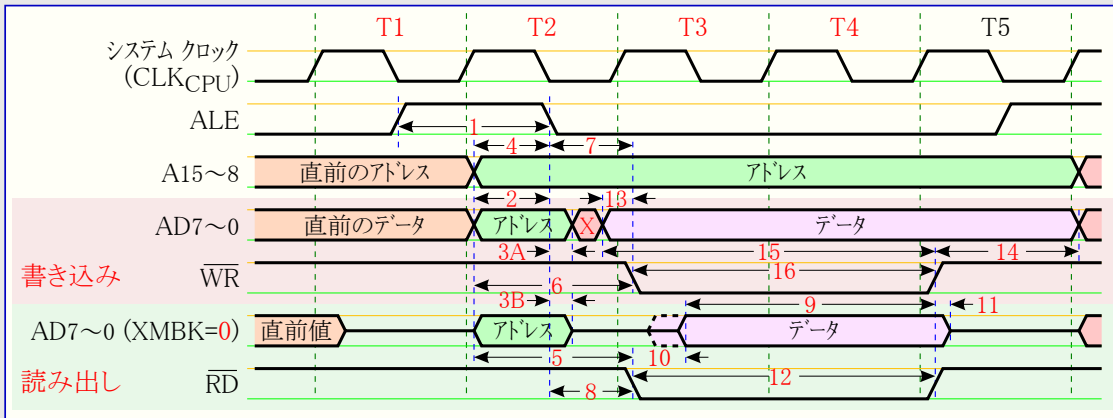


図159. 2待ち状態 外部データメモリアクセス周期 (SRWn1=1, SRWn0=0) [T1~T5]

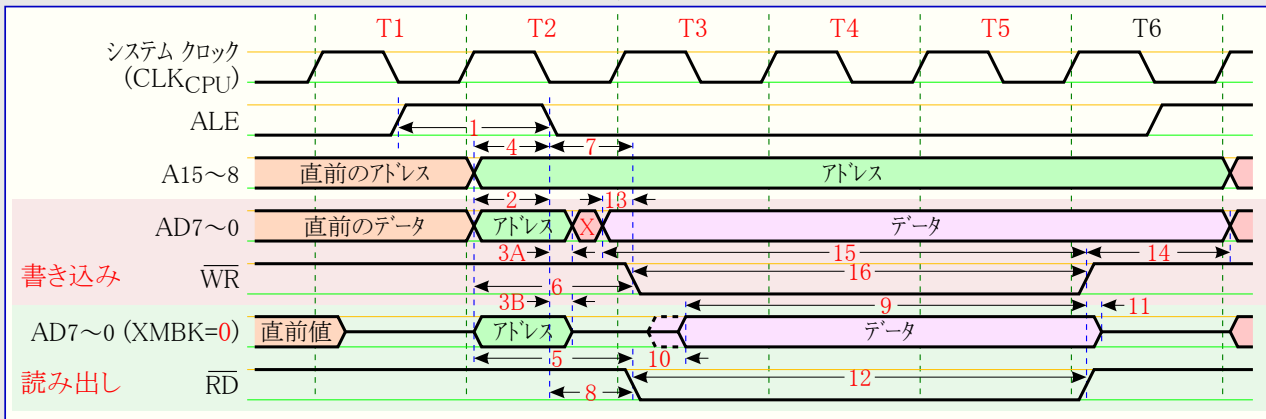
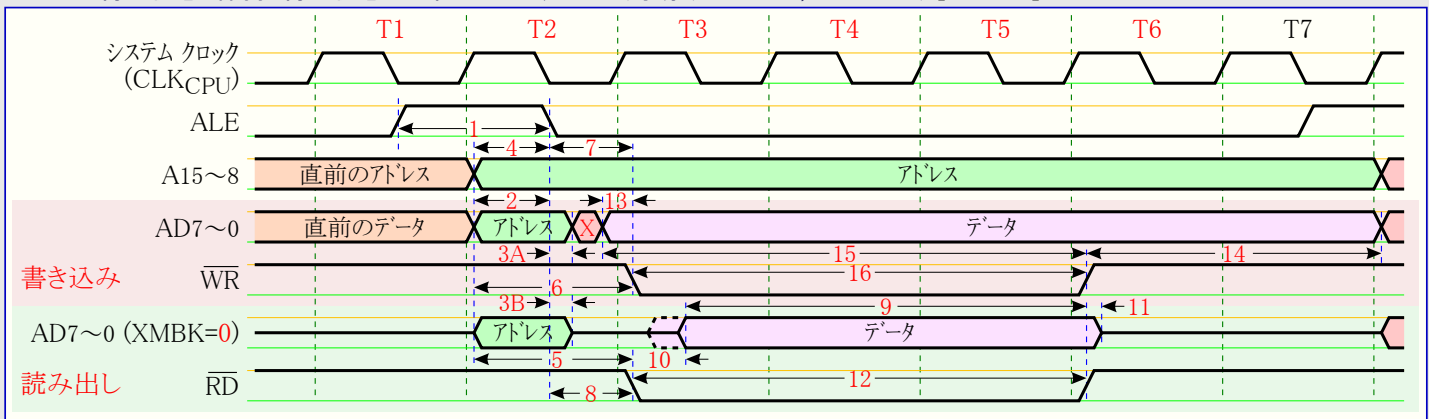


図160. 2待ち状態+保持1待ち状態 外部データメモリアクセス周期 (SRWn1=1, SRWn0=1) [T1~T6]



## 代表特性

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。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。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。

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。

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。

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)  $CL(\text{負荷容量}) \times VCC(\text{動作電圧}) \times f(\text{I/Oピンの平均切り替え周波数})$  として推測できます。

デバイスは検査範囲よりも高い周波数特性を示します。デバイスは注文番号が示す周波数よりも高い周波数での機能特性を保証されません。

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。

## 活動動作消費電流

図161. 活動動作消費電流 対 周波数 (100kHz~1MHz)

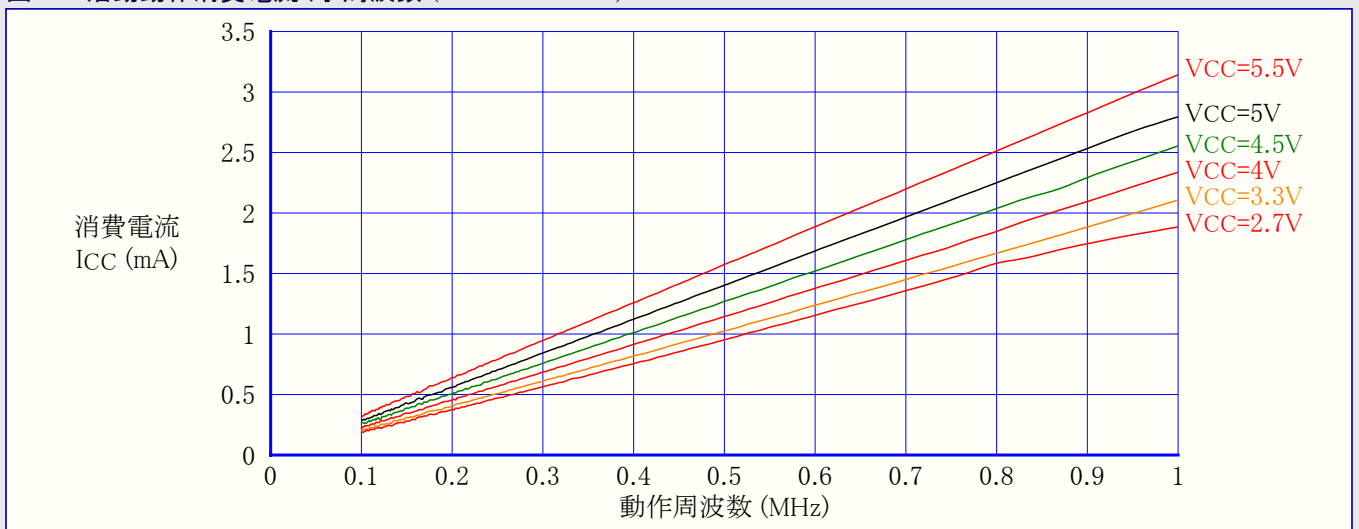


図162. 活動動作消費電流 対 周波数 (1MHz~20MHz)

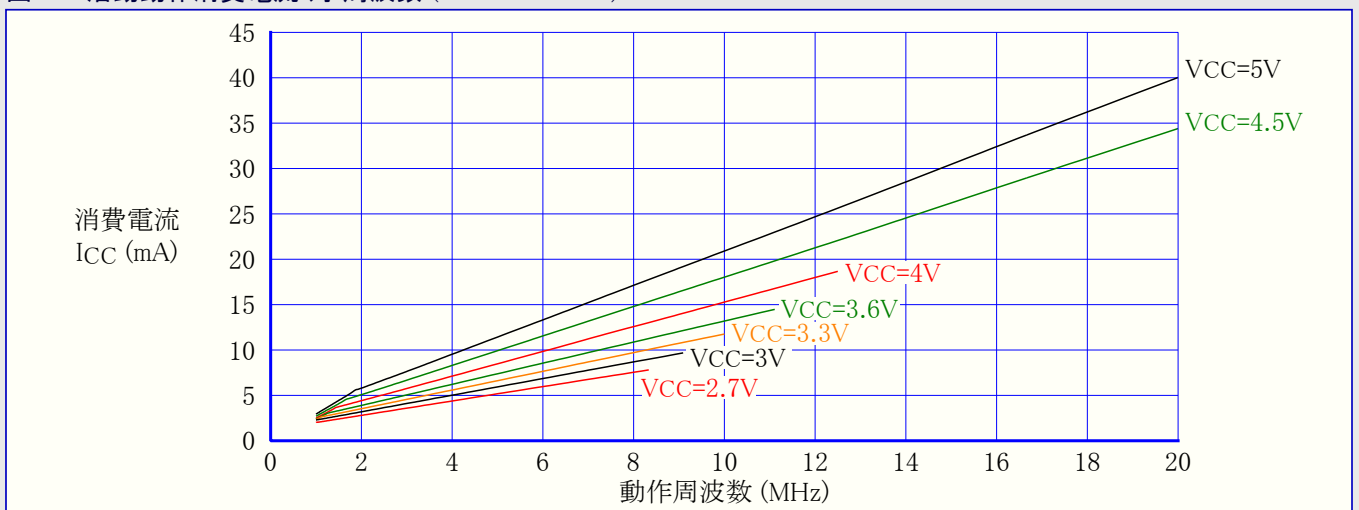


図163. 活動動作消費電流 対 動作電圧 (内蔵RC発振器, 1MHz)

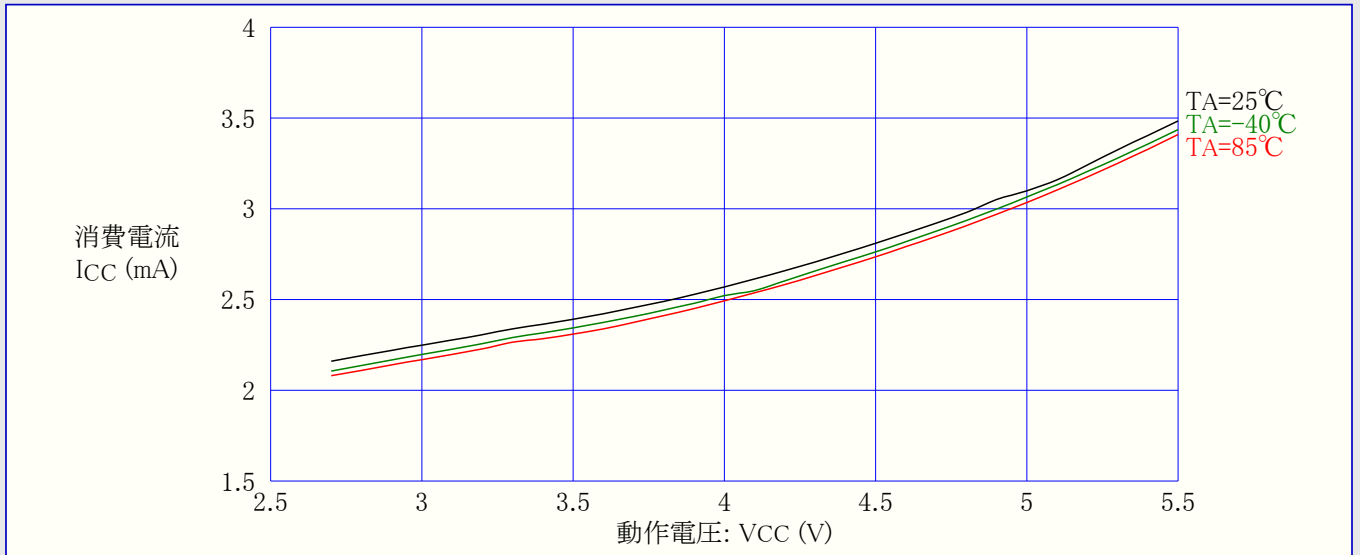


図164. 活動動作消費電流 対 動作電圧 (内蔵RC発振器, 2MHz)

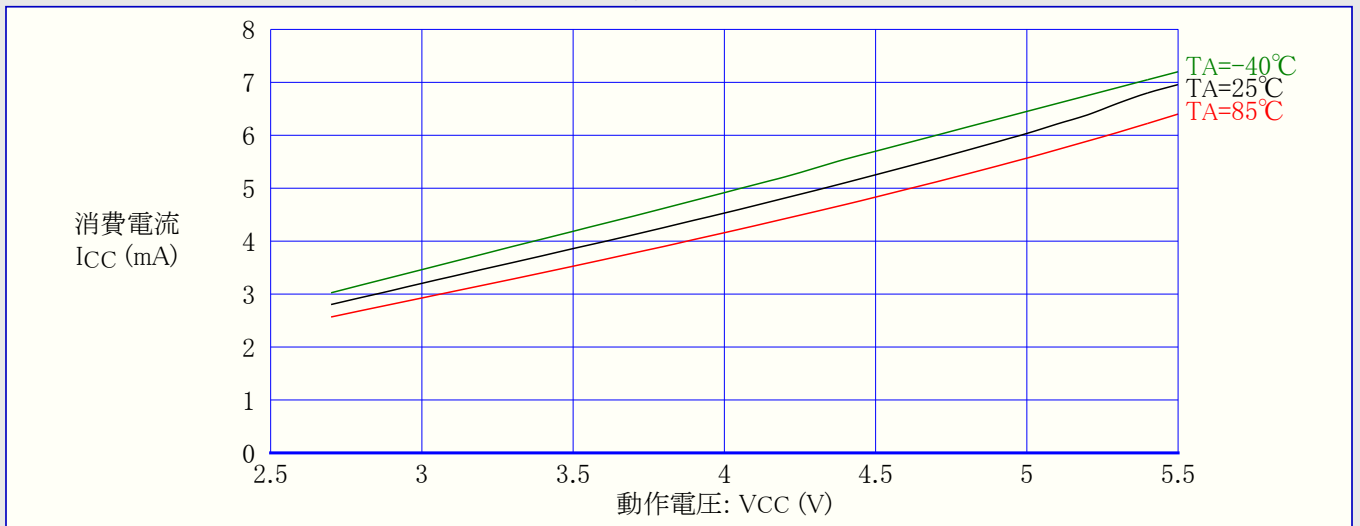


図165. 活動動作消費電流 対 動作電圧 (内蔵RC発振器, 4MHz)

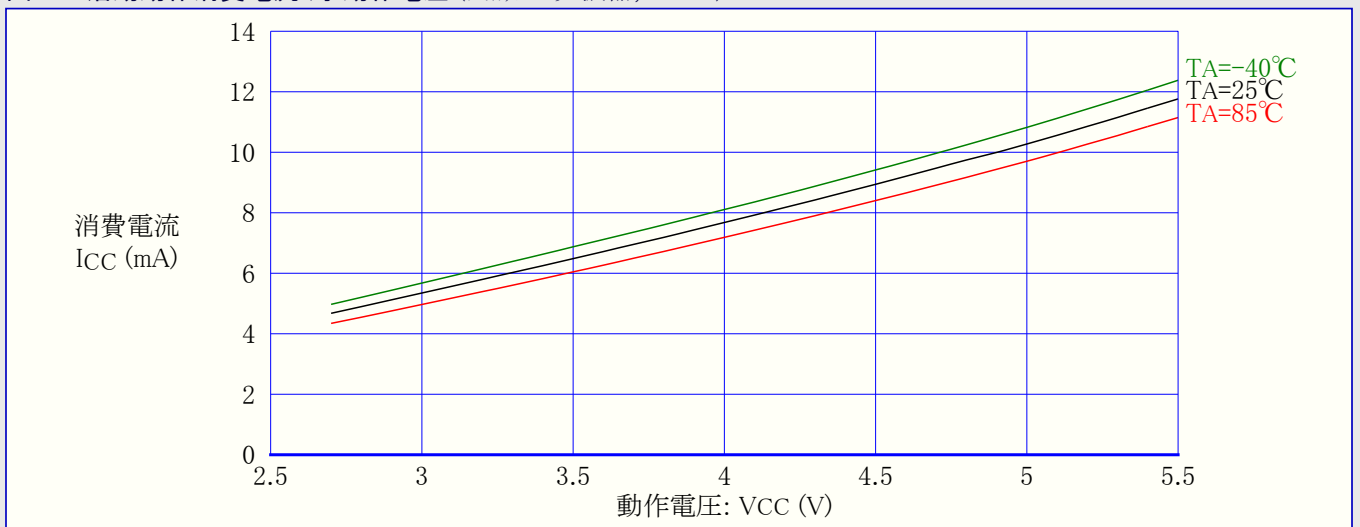


図166. 活動動作消費電流 対 動作電圧 (内蔵RC発振器, 8MHz)

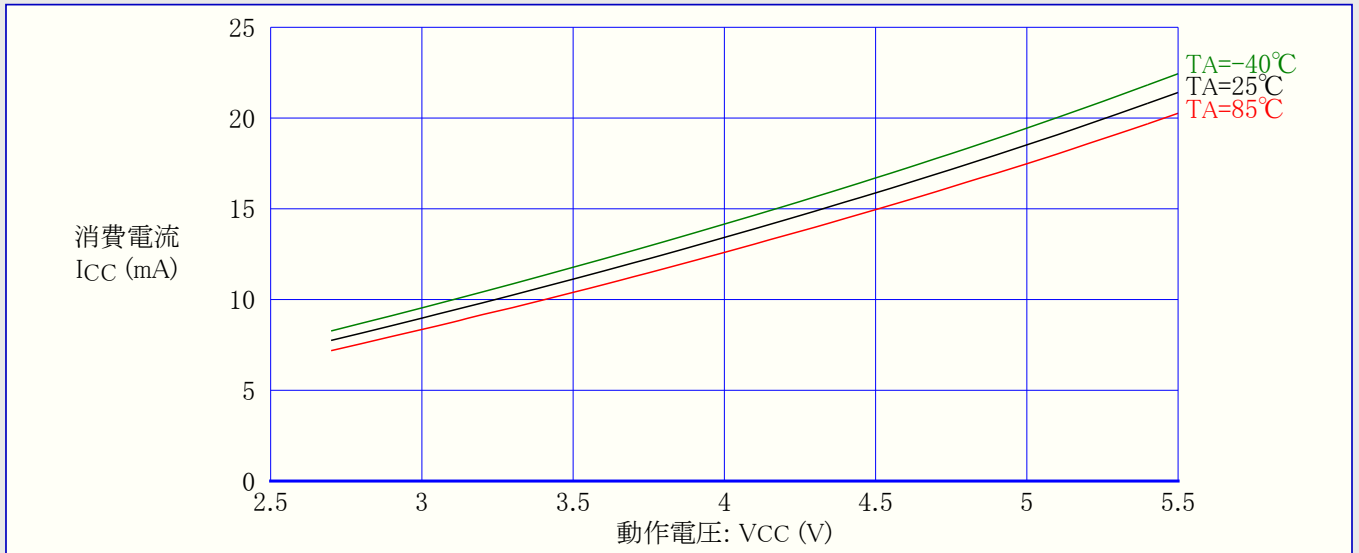
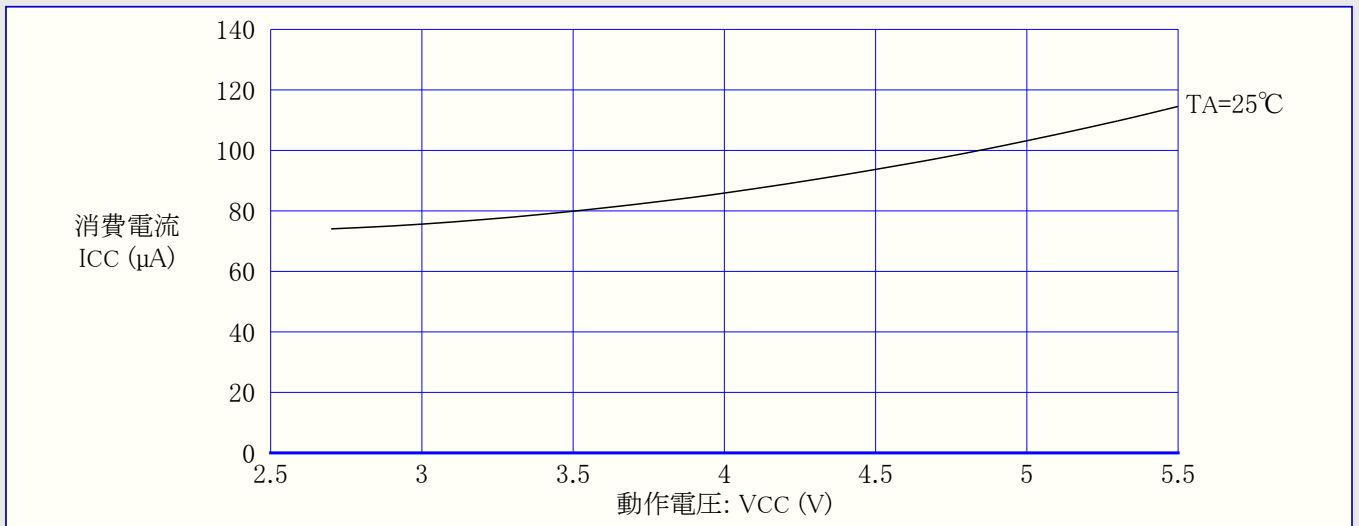


図167. 活動動作消費電流 対 動作電圧 (32kHz外部発振器)



アイドル動作消費電流

図168. アイドル動作消費電流 対 周波数 (100kHz~1MHz)

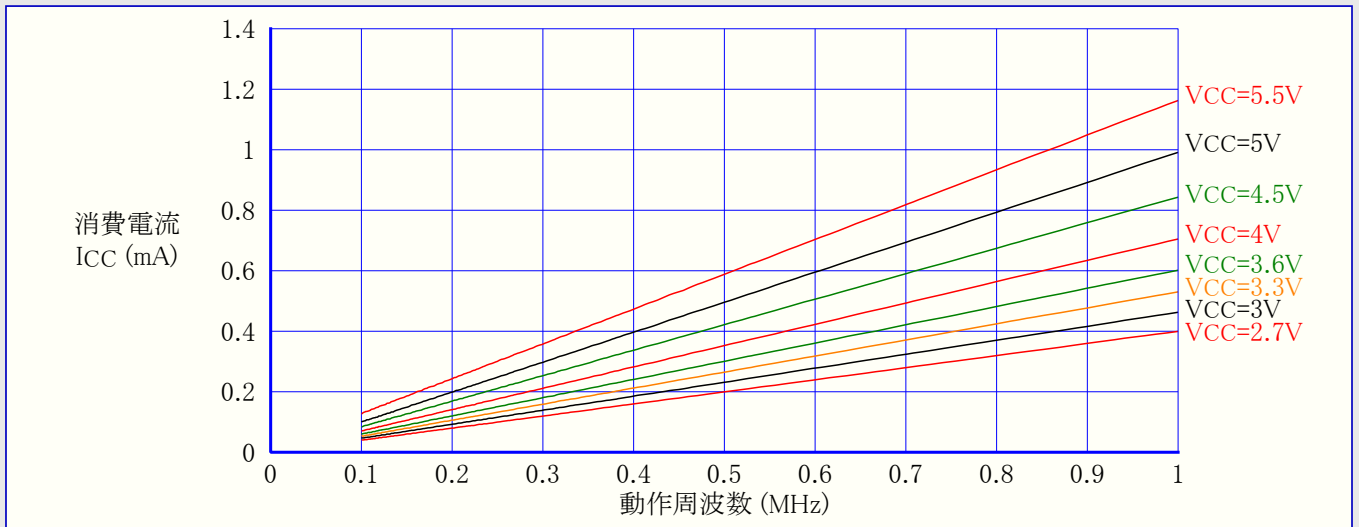




図169. アイドル動作消費電流 対 周波数 (1MHz~20MHz)

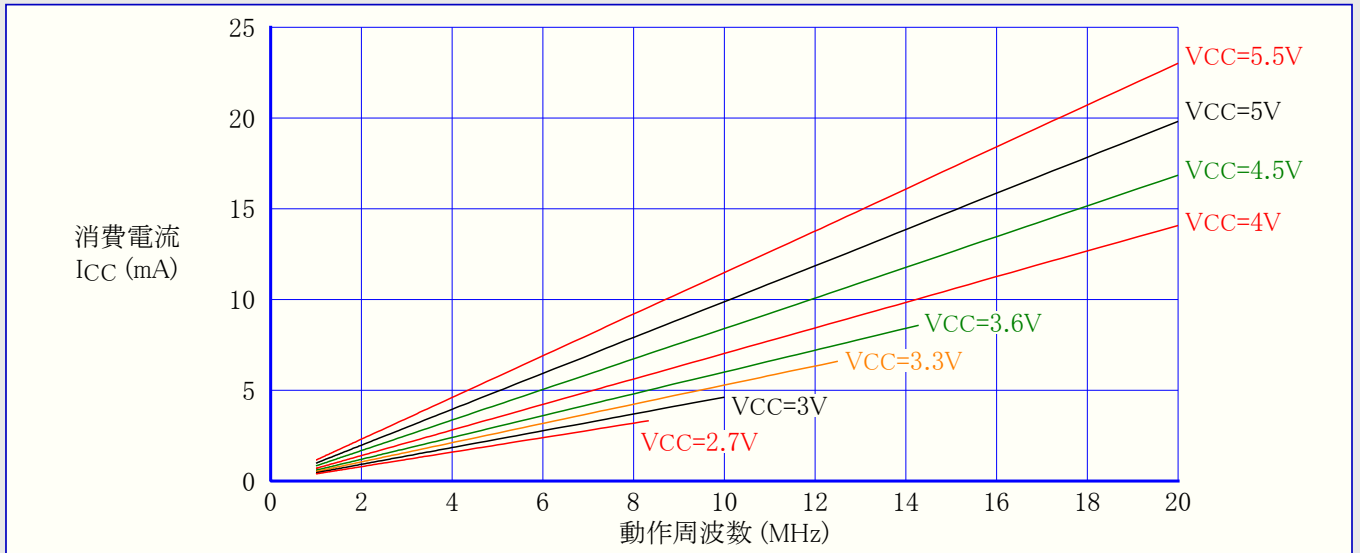


図170. アイドル動作消費電流 対 動作電圧 (内蔵RC発振器, 1MHz)

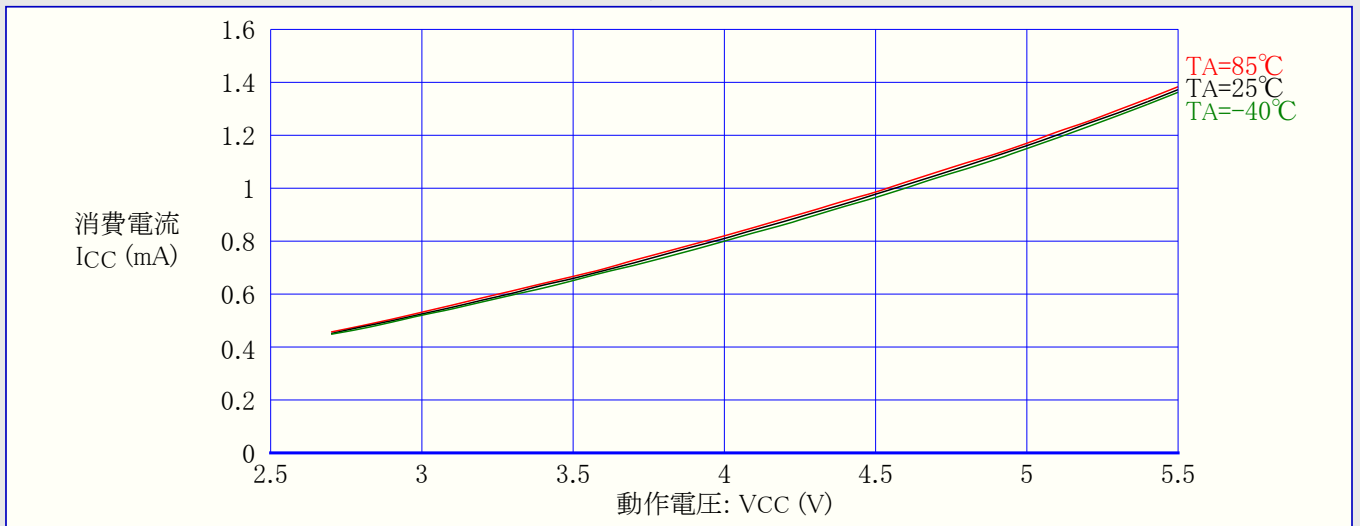


図171. アイドル動作消費電流 対 動作電圧 (内蔵RC発振器, 2MHz)

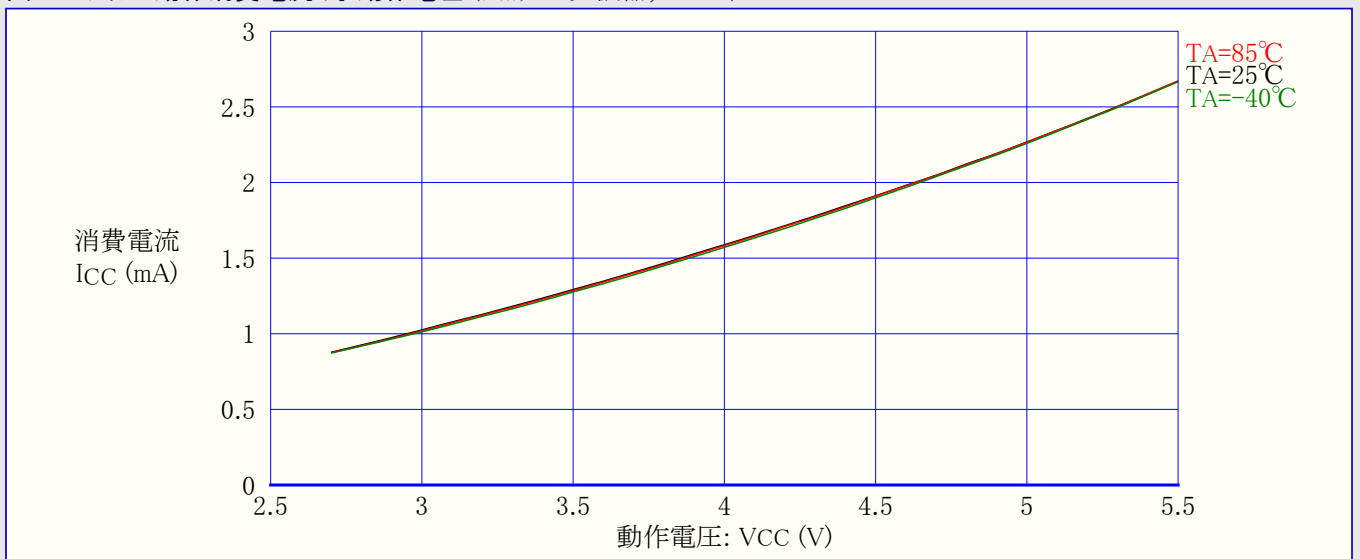


図172. アイドル動作消費電流 対 動作電圧 (内蔵RC発振器, 4MHz)

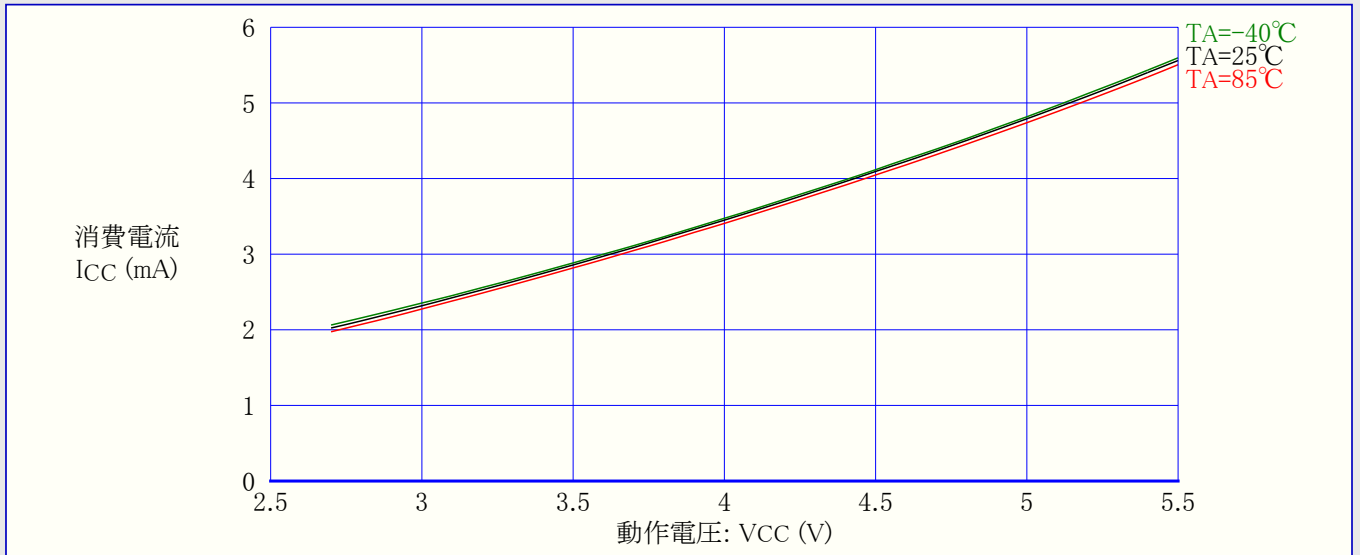


図173. アイドル動作消費電流 対 動作電圧 (内蔵RC発振器, 8MHz)

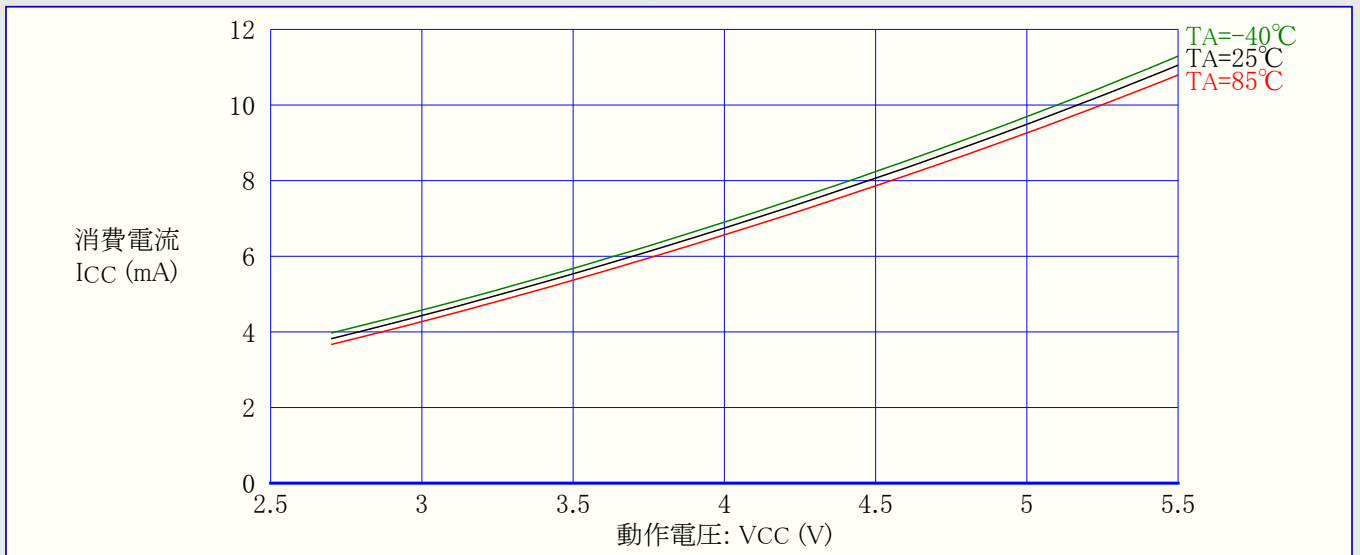
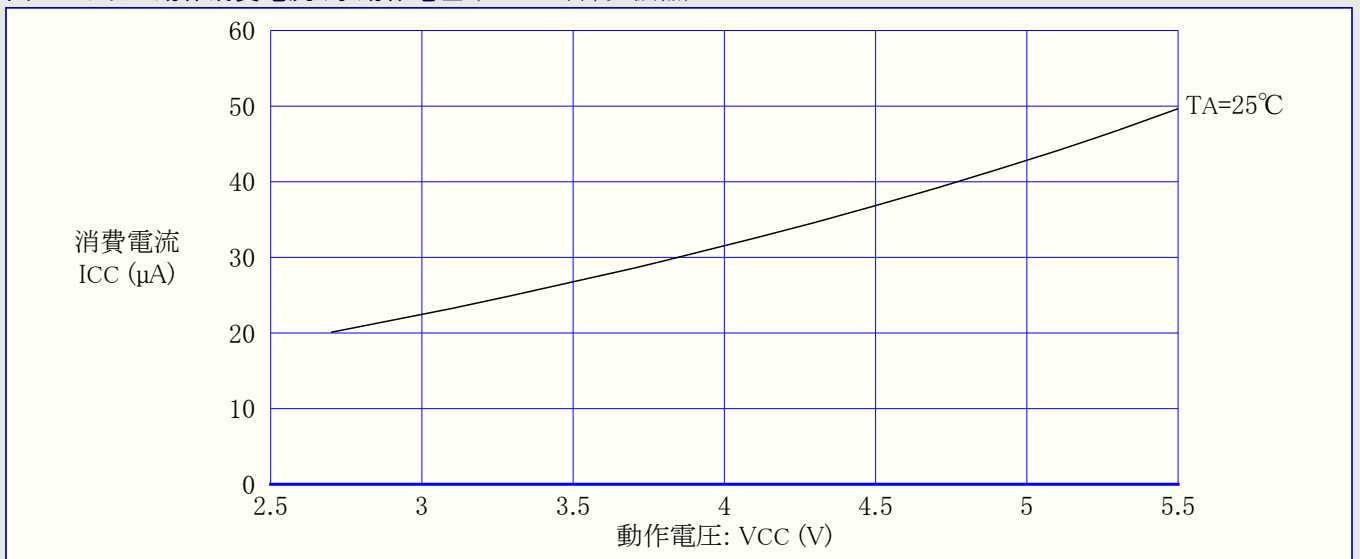


図174. アイドル動作消費電流 対 動作電圧 (32kHz外部発振器)



パワーダウン動作消費電流

図175. 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)

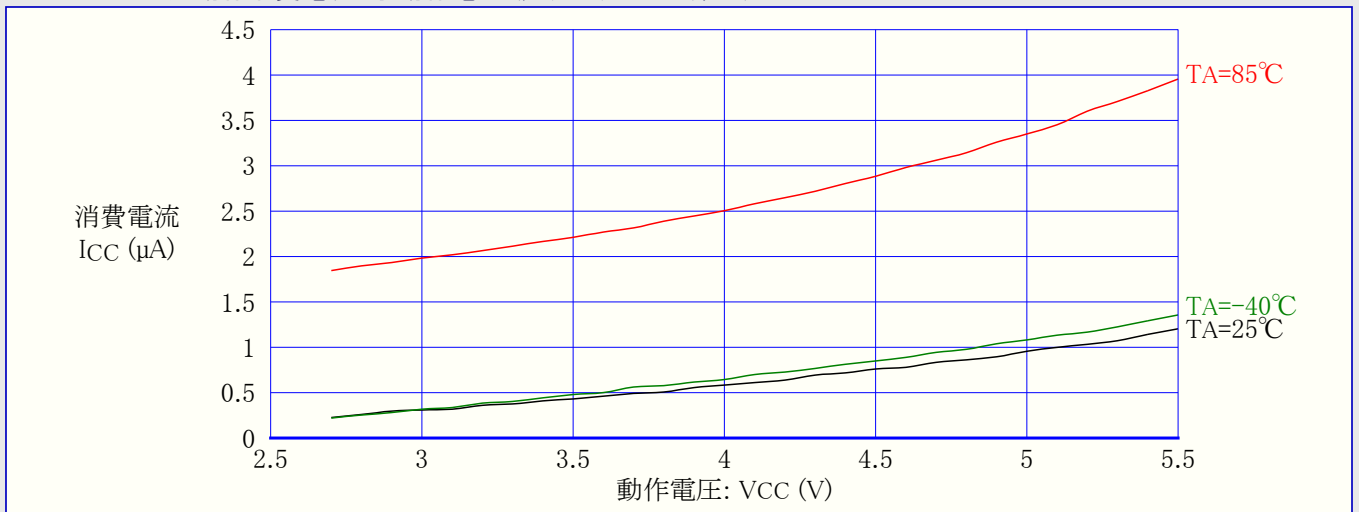
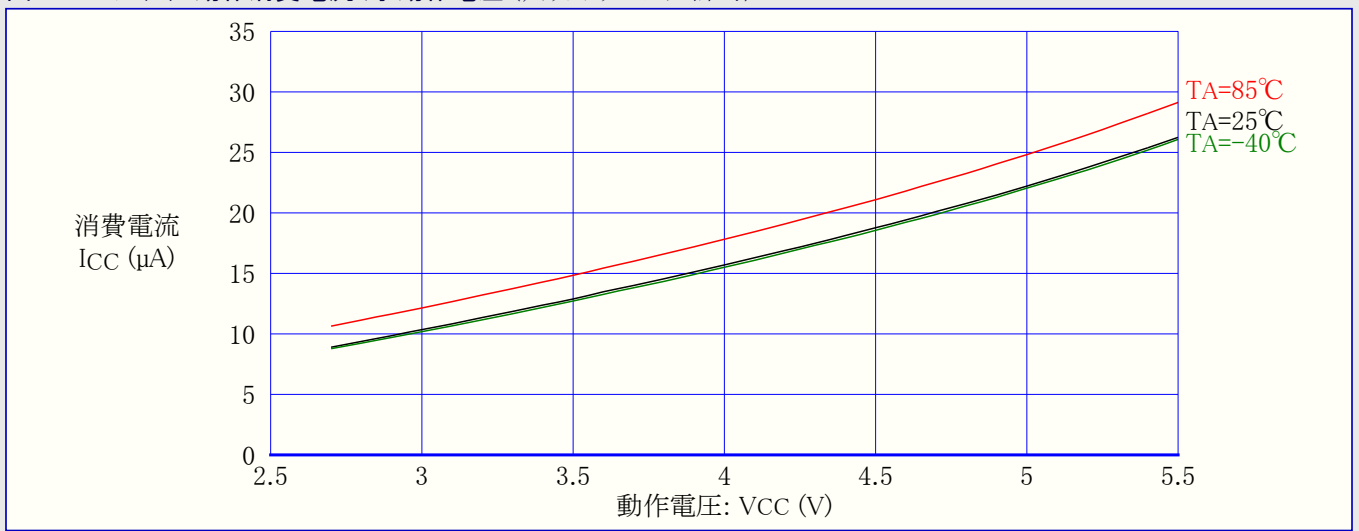
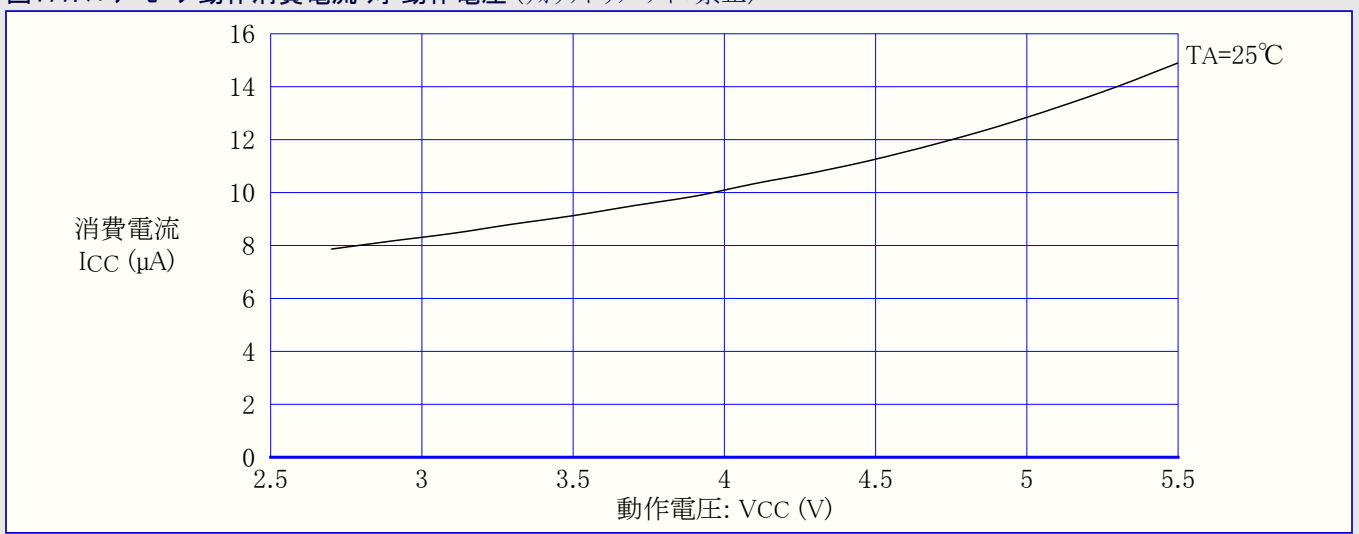


図176. パワーダウン動作消費電流 対 動作電圧 (ウォッチドッグ タイマ許可)



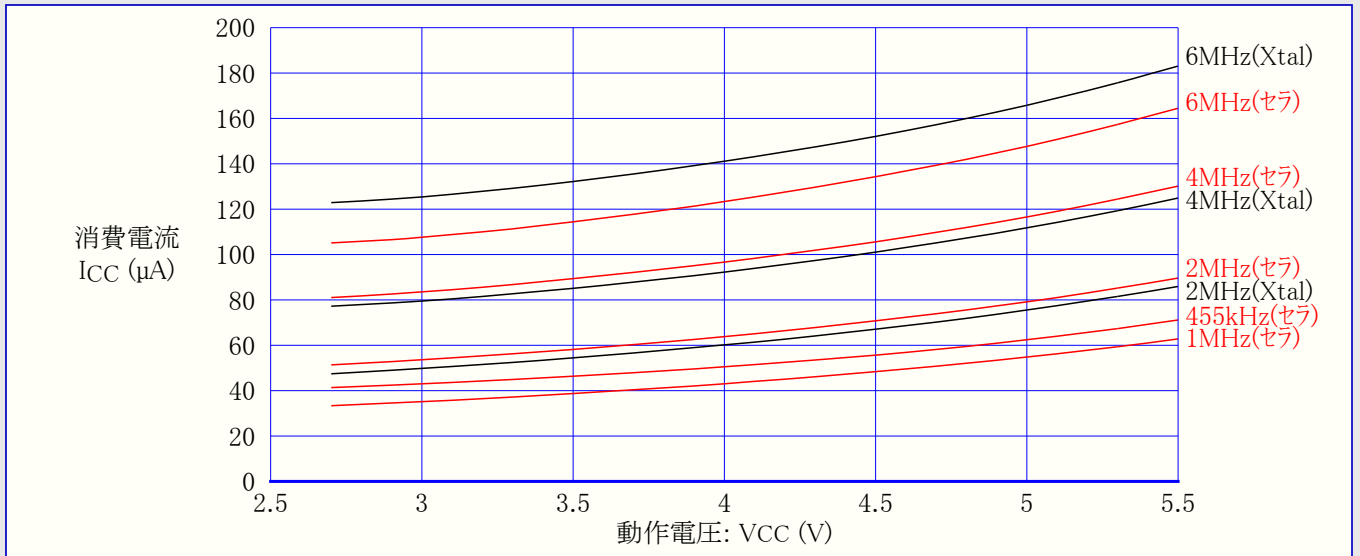
パワーセーブ動作消費電流

図177. パワーセーブ動作消費電流 対 動作電圧 (ウォッチドッグ タイマ禁止)



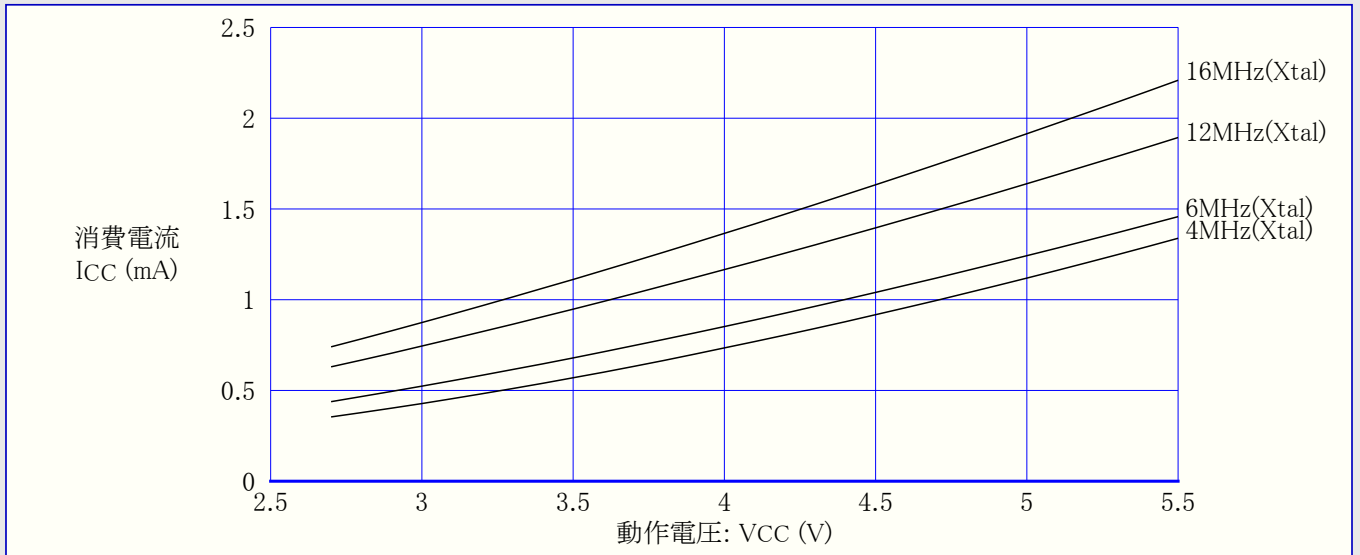
## スタンバイ動作消費電流

図178. スタンバイ動作消費電流 対 動作電圧 (CKOPT=1)



注: セラはセラミック振動子

図179. スタンバイ動作消費電流 対 動作電圧 (CKOPT=0)



## ピンプルアップ

図180. I/Oピンプルアップ抵抗電流 対 入力電圧 (VCC=5V)

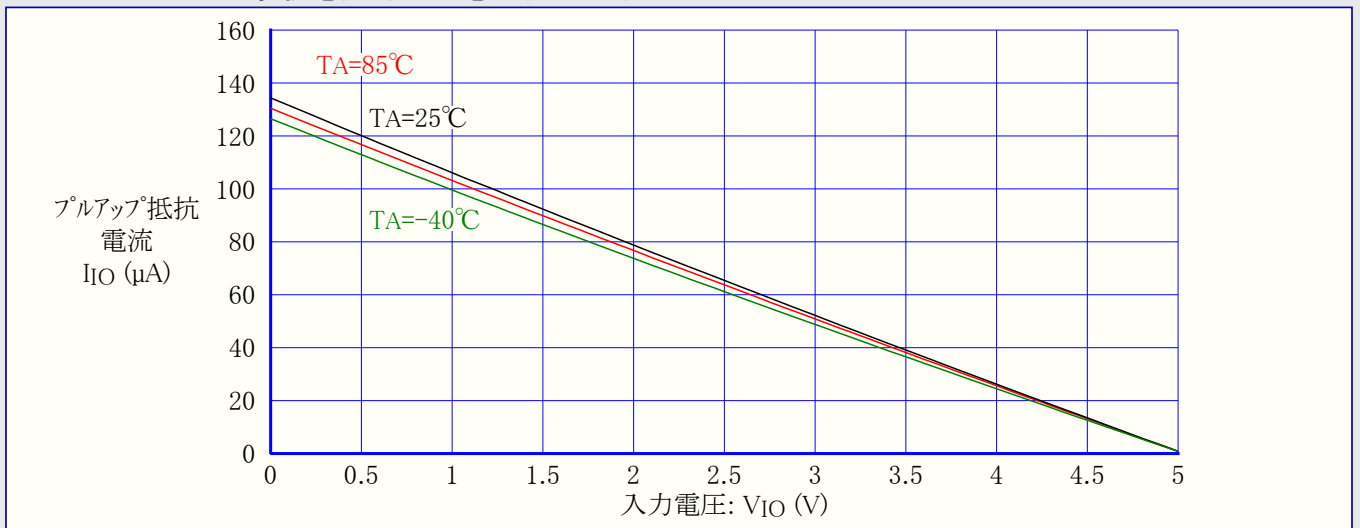
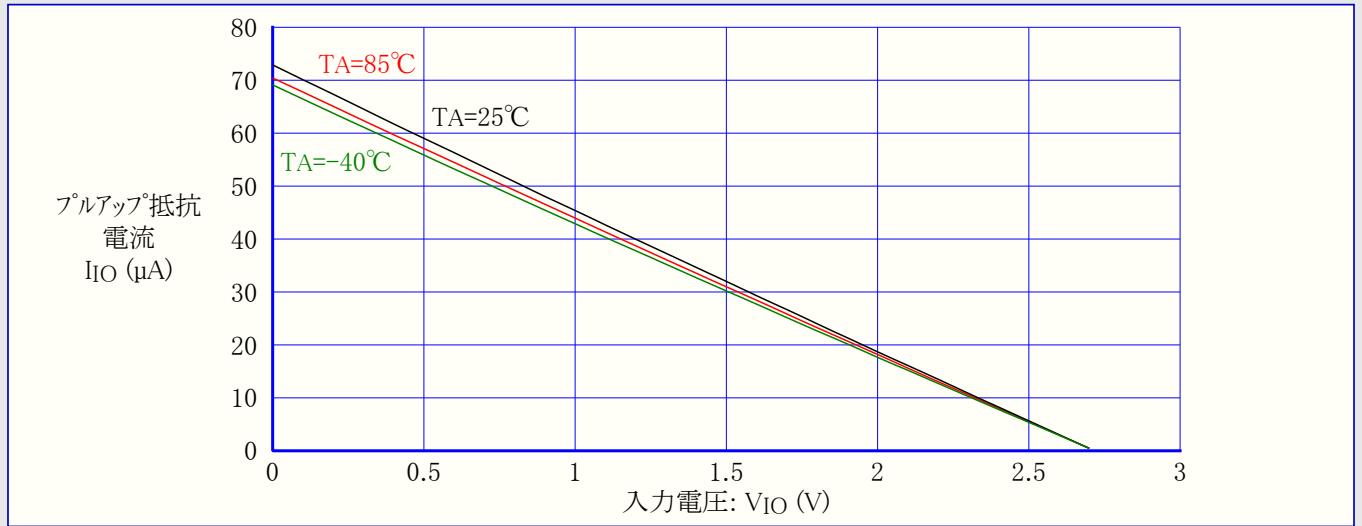


図181. I/Oピンプルアップ抵抗電流 対 入力電圧 (VCC=2.7V)



ピン駆動能力

図182. I/Oピン吐き出し電流 対 出力電圧 (VCC=5V)

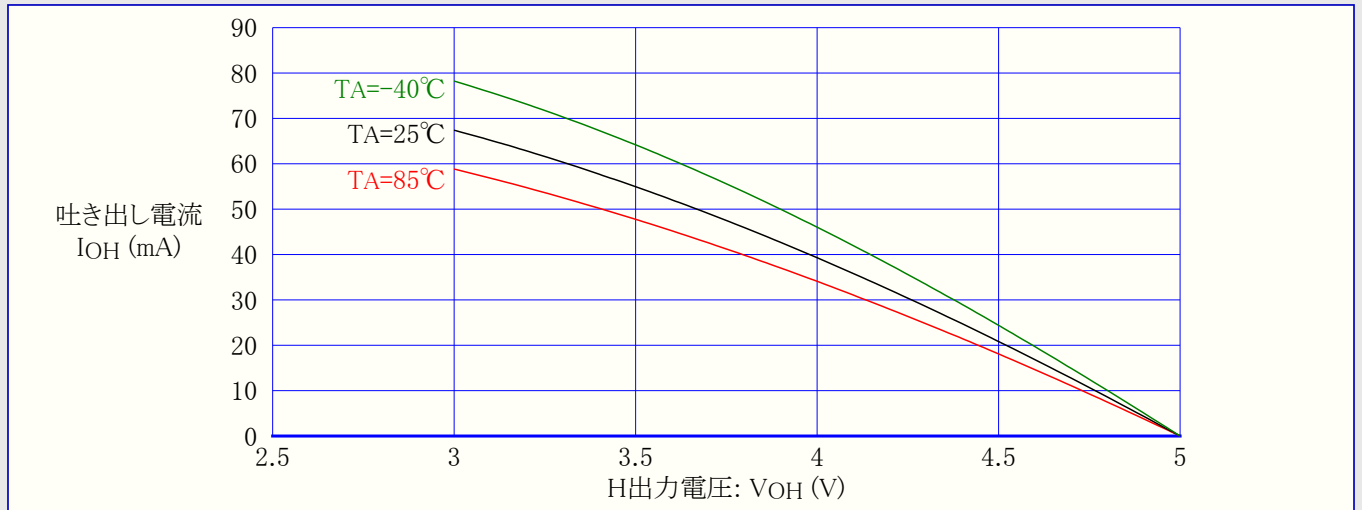


図183. I/Oピン吐き出し電流 対 出力電圧 (VCC=2.7V)

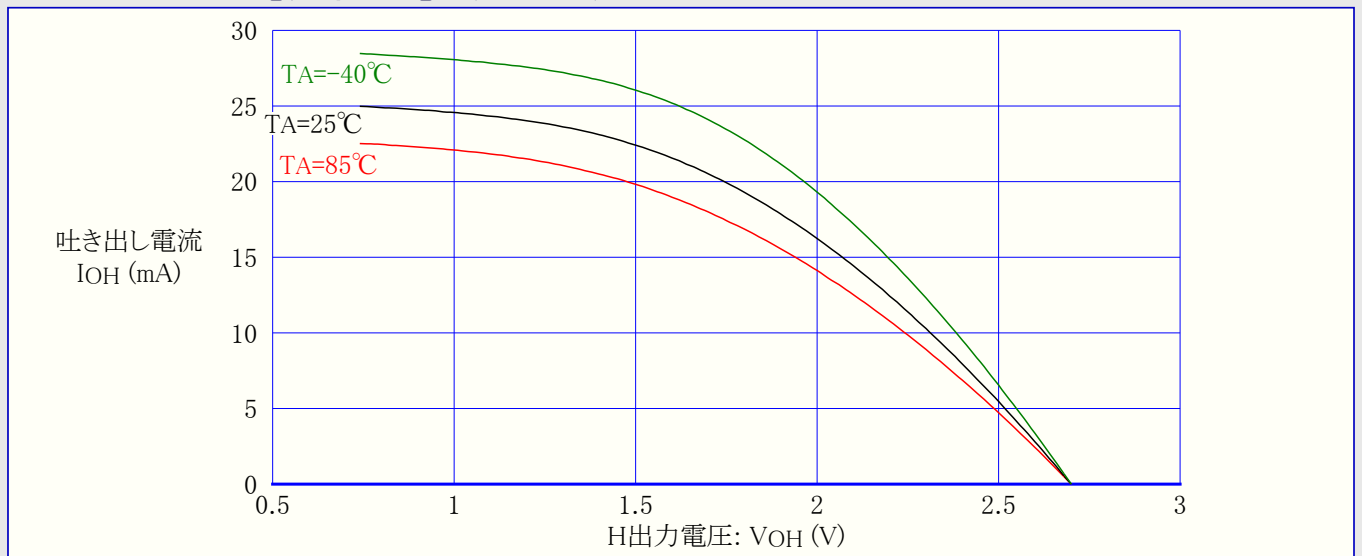




図184. I/Oピン吸い込み電流 対 出力電圧 (VCC=5V)

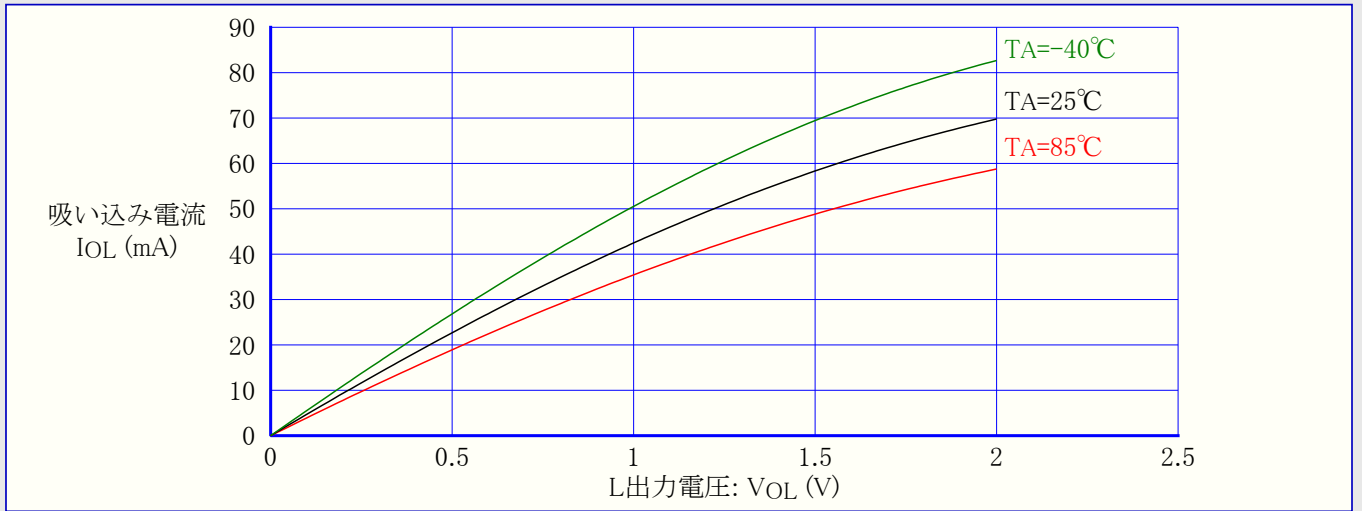
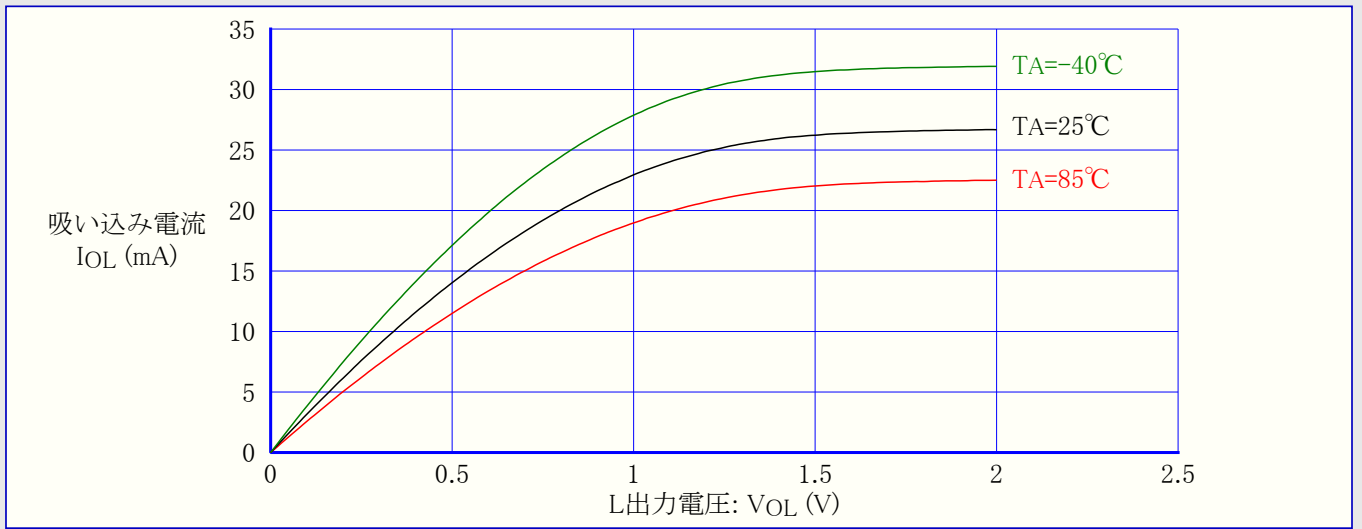


図185. I/Oピン吸い込み電流 対 出力電圧 (VCC=2.7V)



ピン 閾値とヒステリシス

図186. 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)

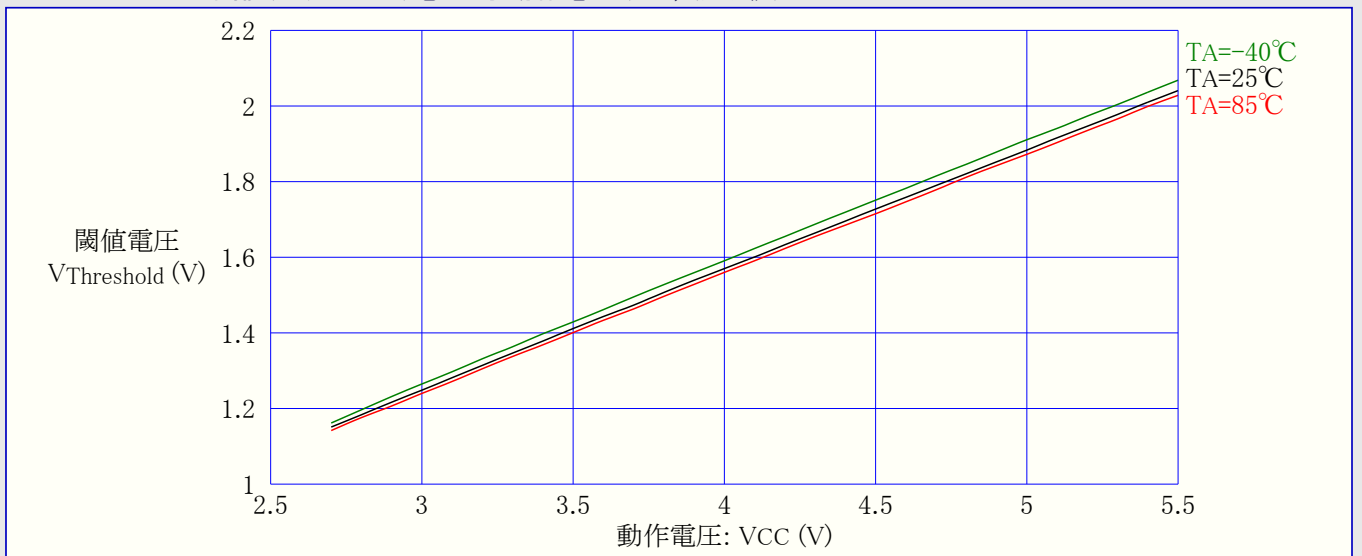


図187. I/Oピン入力閾値(スレッシュホールド)電圧 対 動作電圧 (VIL,0読み値)

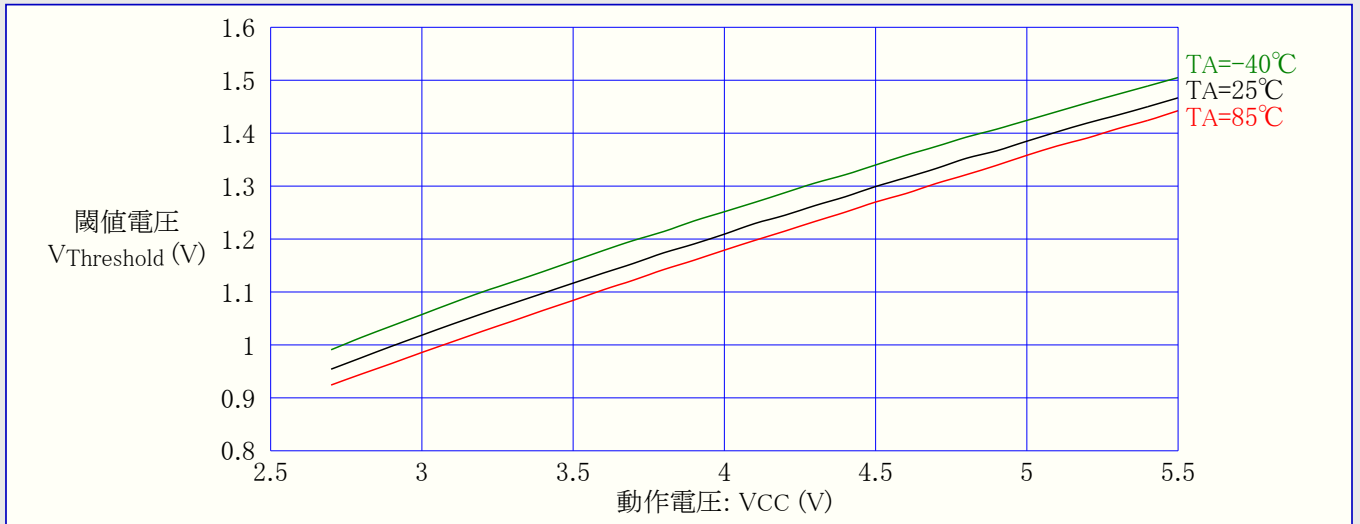
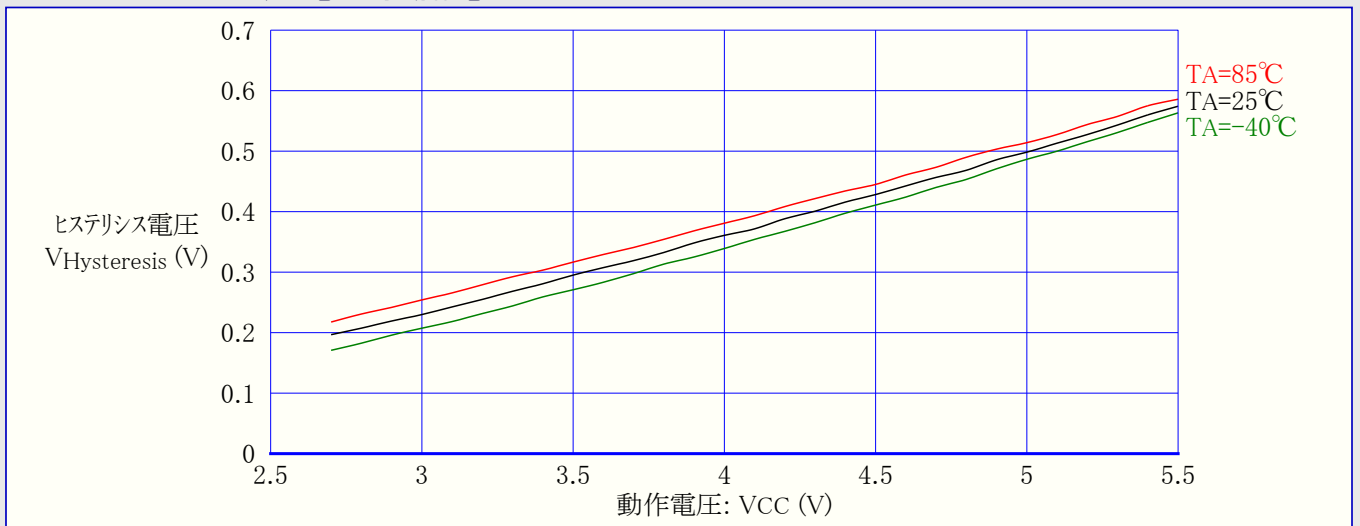


図188. I/Oピン入力ヒステリシス電圧 対 動作電圧



低電圧検出器(BOD)閾値とアナログ比較器変位(オフセット)

図189. 低電圧検出器(BOD)閾値(スレッシュホールド)電圧 対 動作温度 (検出電圧4.0V)

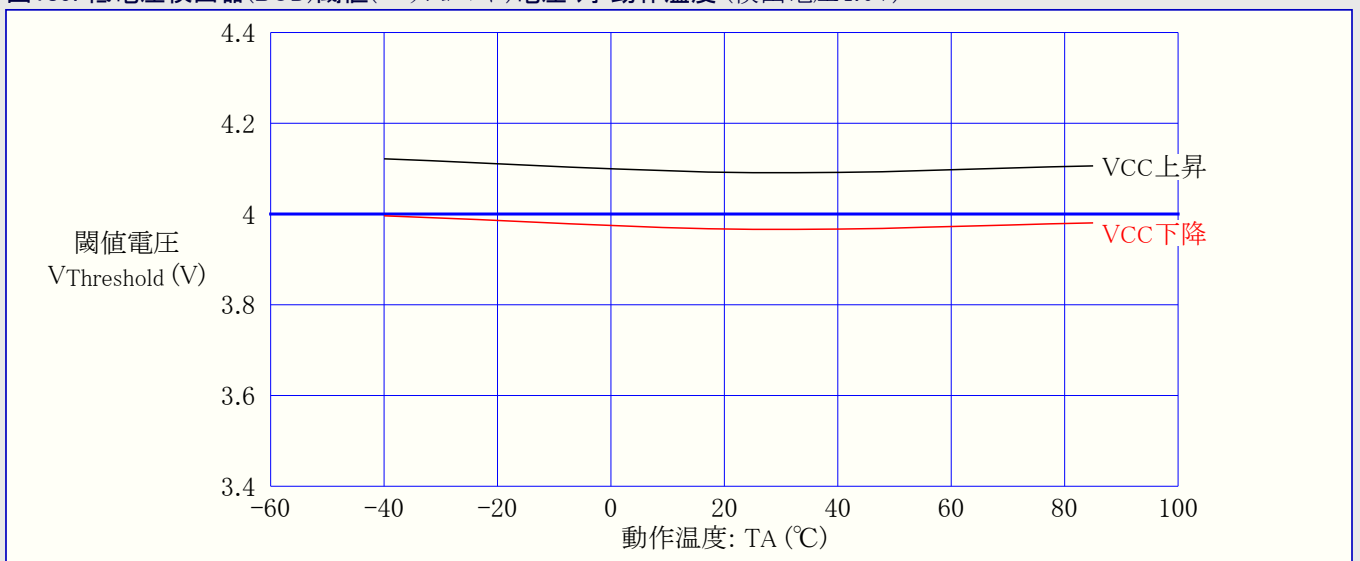


図190. 低電圧検出器(BOD)閾値(スレッシュホールド)電圧 対 動作温度 (検出電圧2.7V)

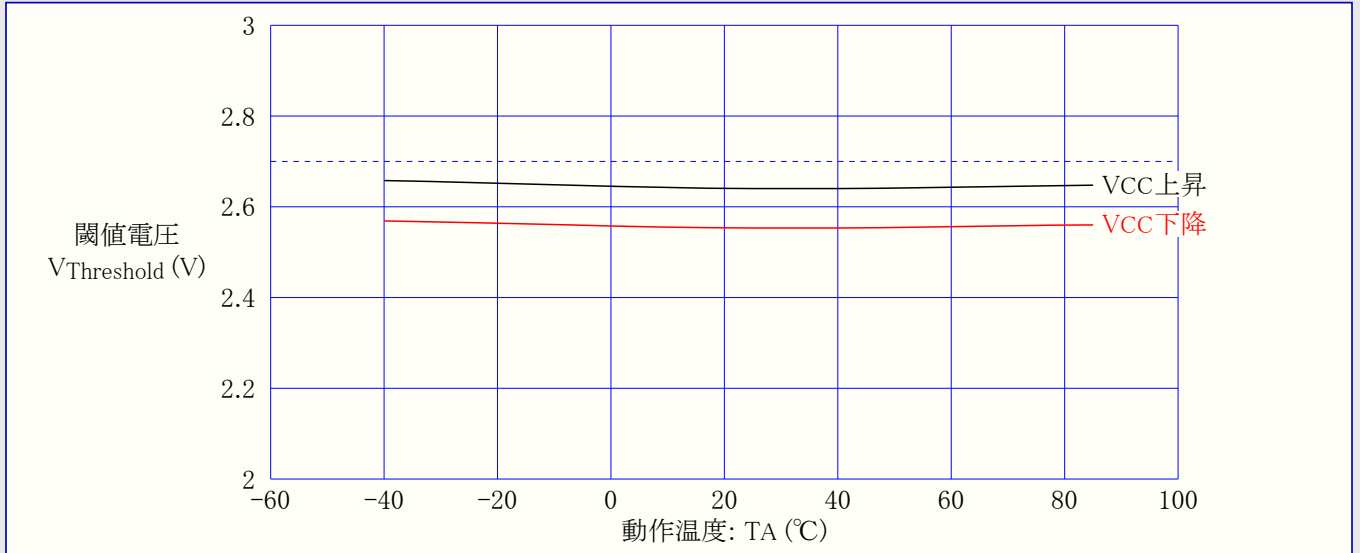
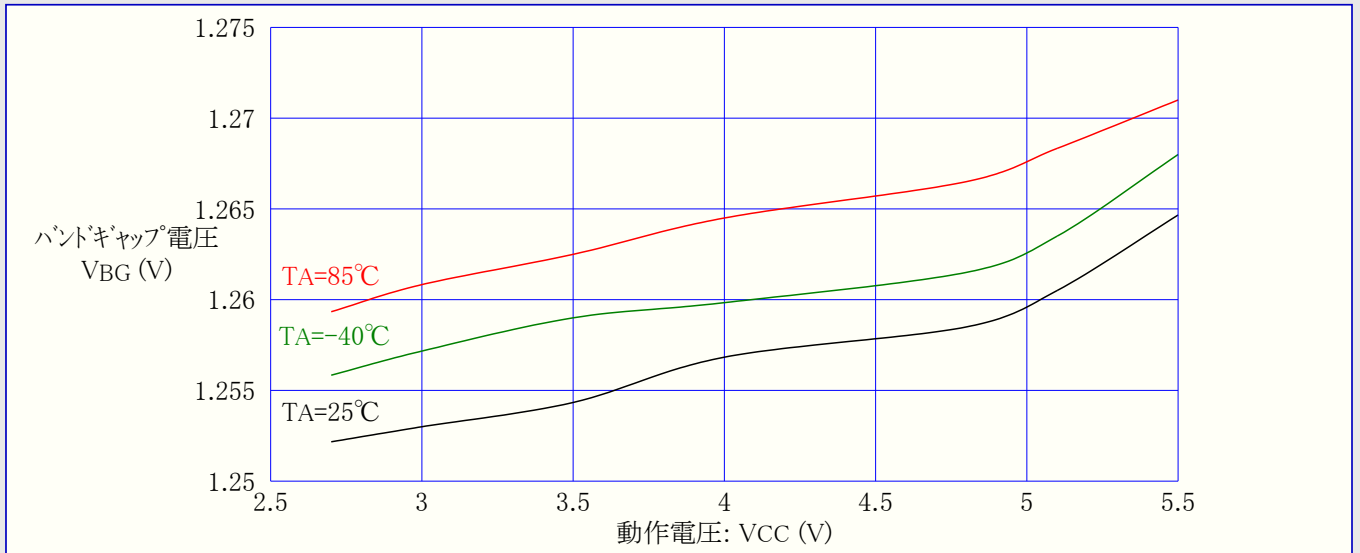


図191. 内部基準(バンドギャップ)電圧 対 動作電圧



## 内部発振器周波数

図192. ウォッチドッグ発振器周波数 対 動作電圧

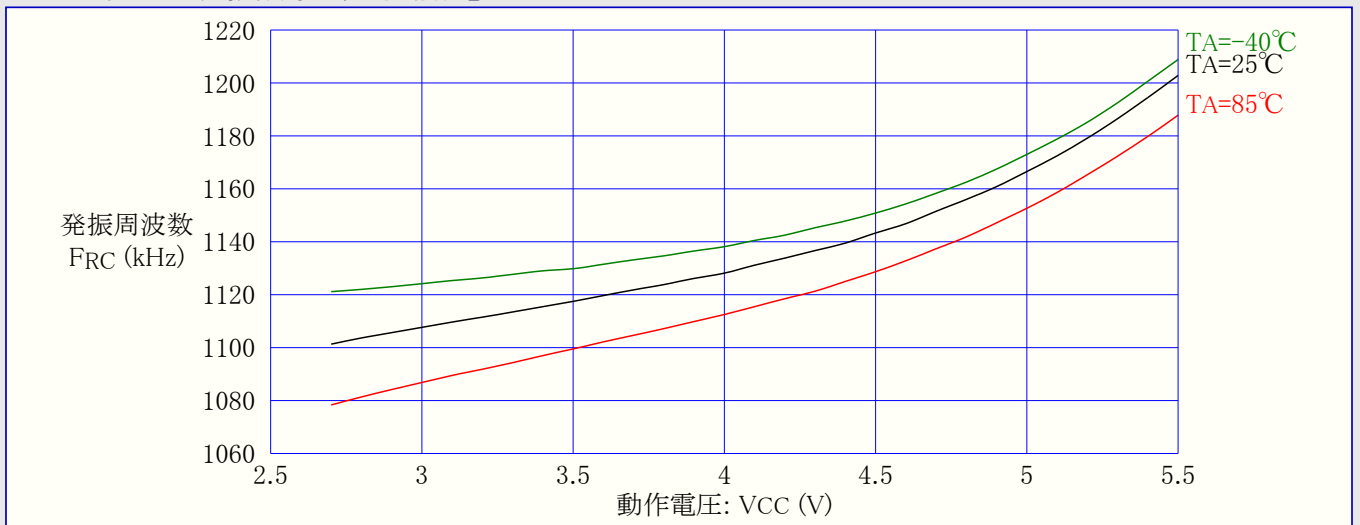


図193. 校正済み1MHz内蔵RC発振器周波数 対 動作温度

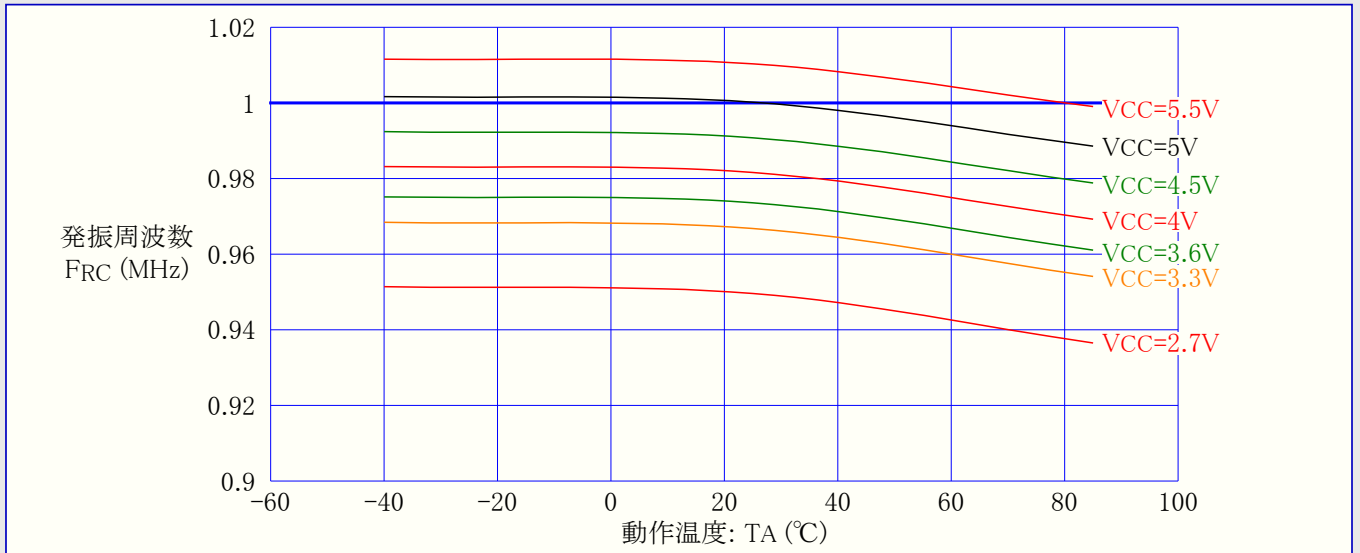


図194. 校正済み1MHz内蔵RC発振器周波数 対 動作電圧

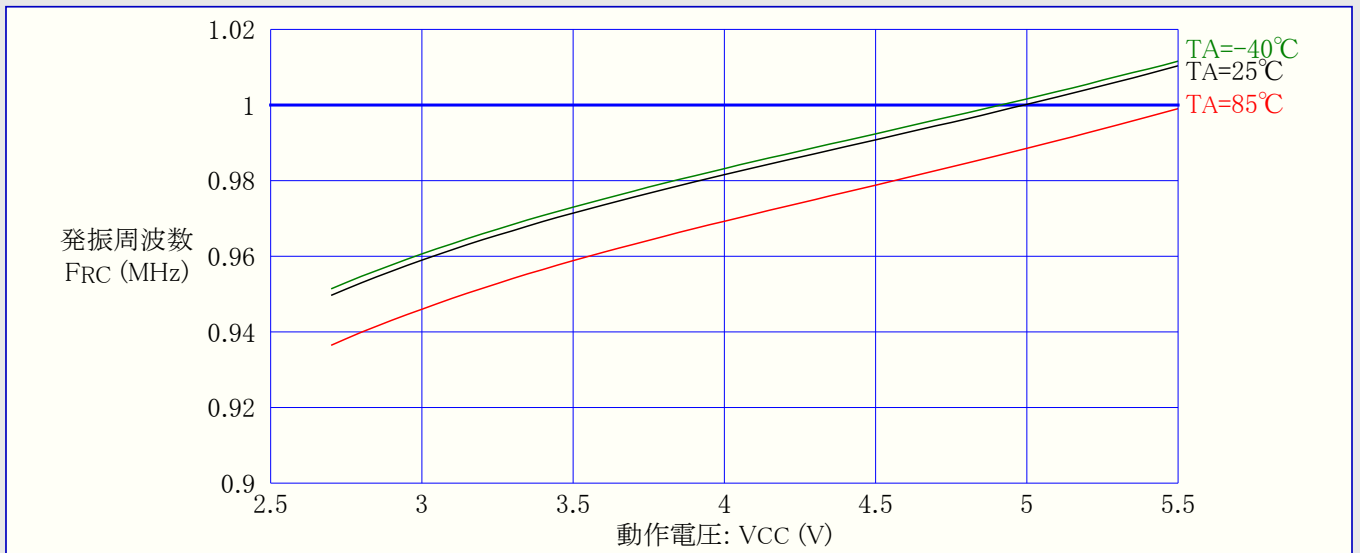


図195. 校正付き1MHz内蔵RC発振器周波数 対 発振校正(OSCCAL)値

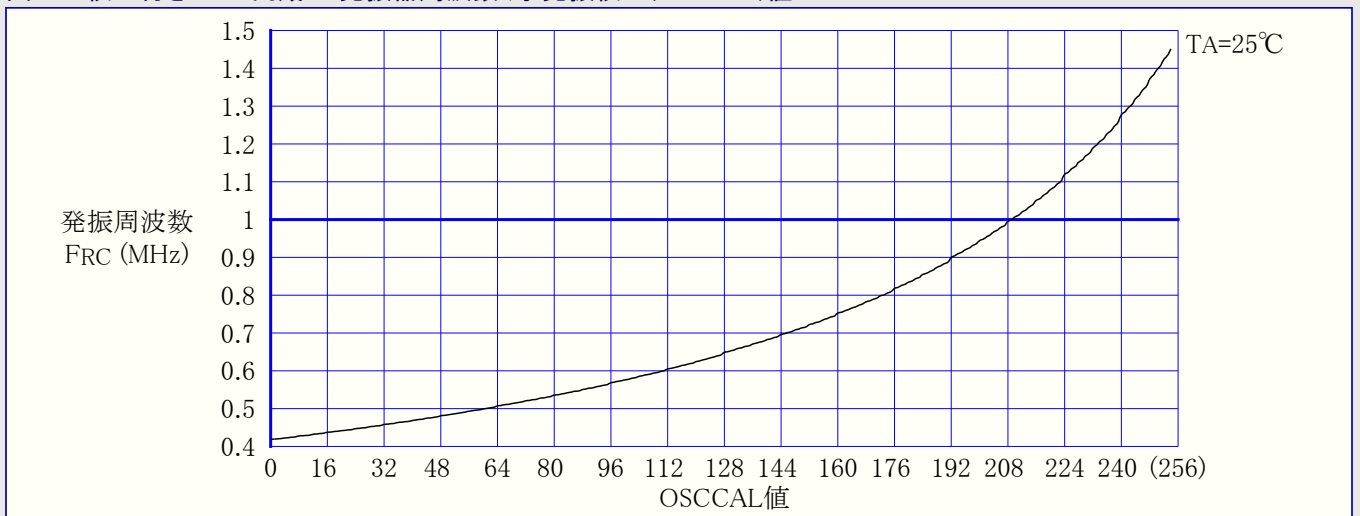


図196. 校正済み2MHz内蔵RC発振器周波数 対 動作温度

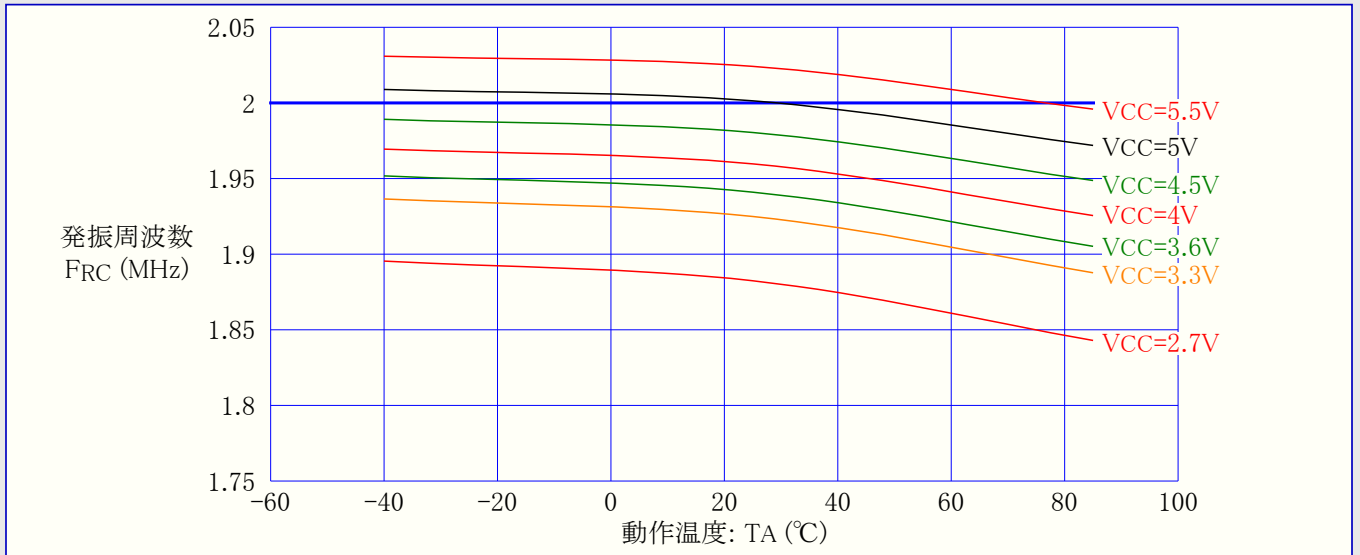


図197. 校正済み2MHz内蔵RC発振器周波数 対 動作電圧

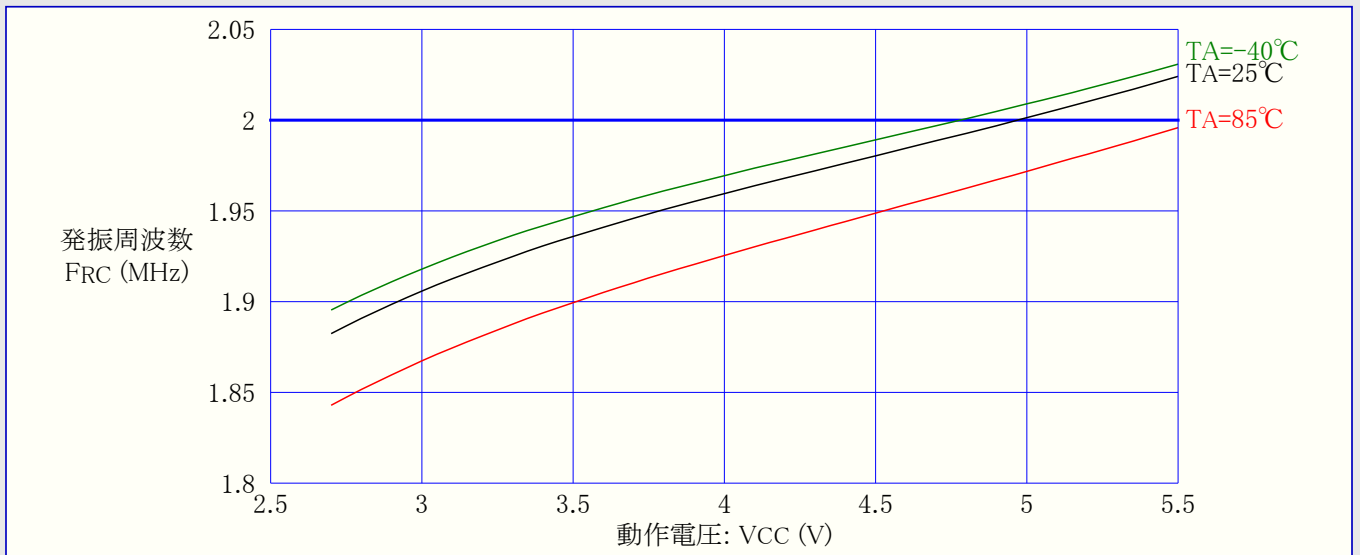


図198. 校正付き2MHz内蔵RC発振器周波数 対 発振校正(OSCCAL)値

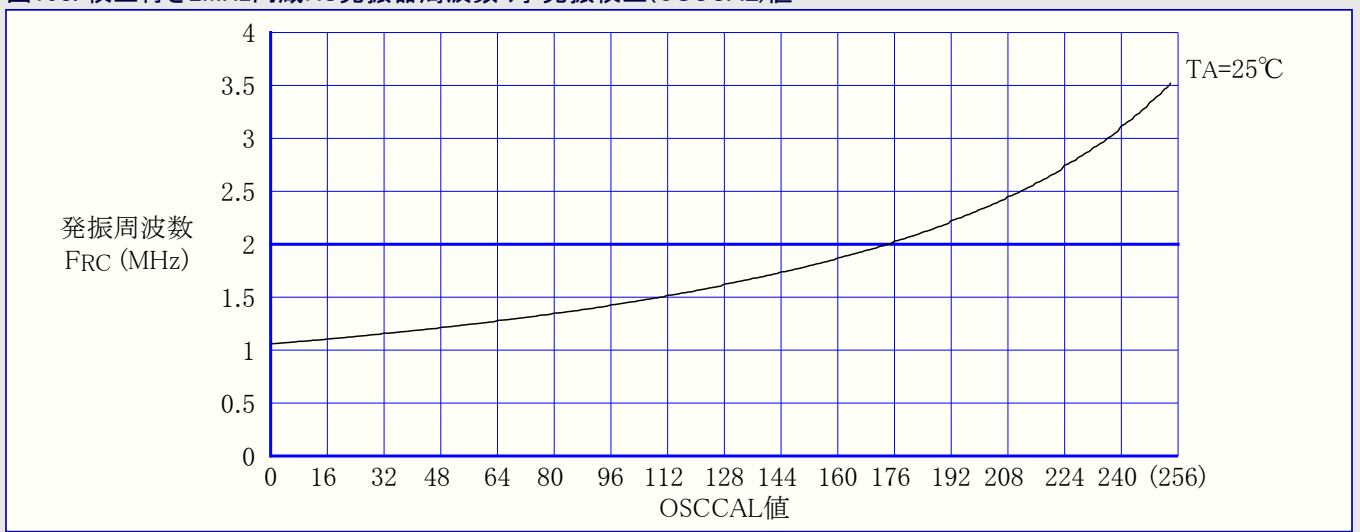




図199. 校正済み4MHz内蔵RC発振器周波数 対 動作温度

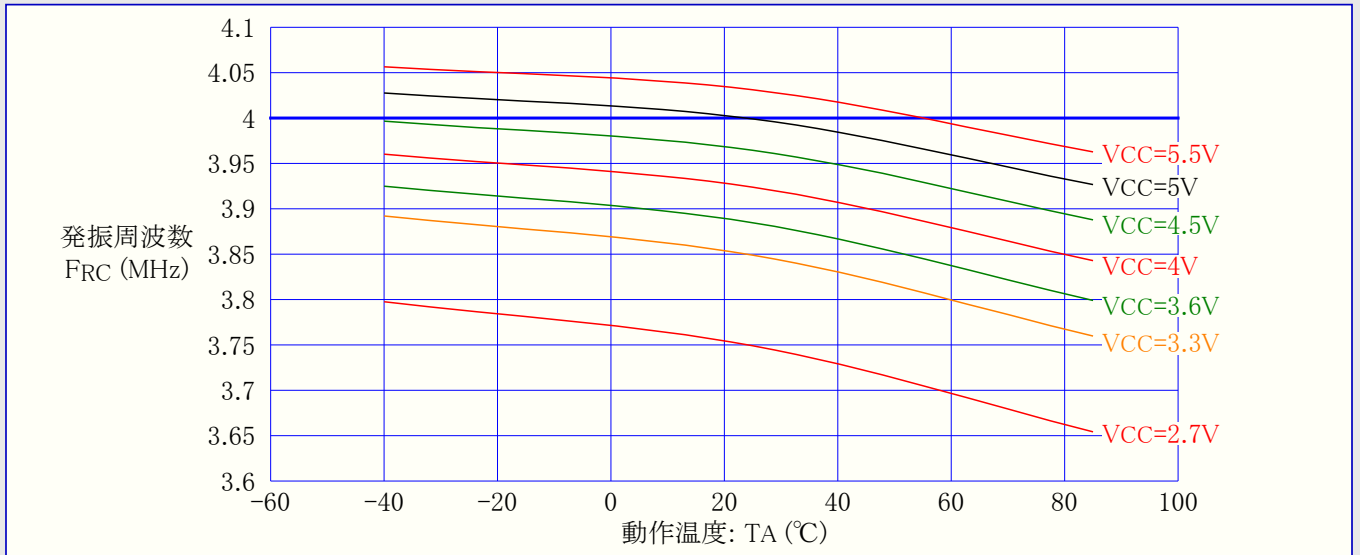


図200. 校正済み4MHz内蔵RC発振器周波数 対 動作電圧

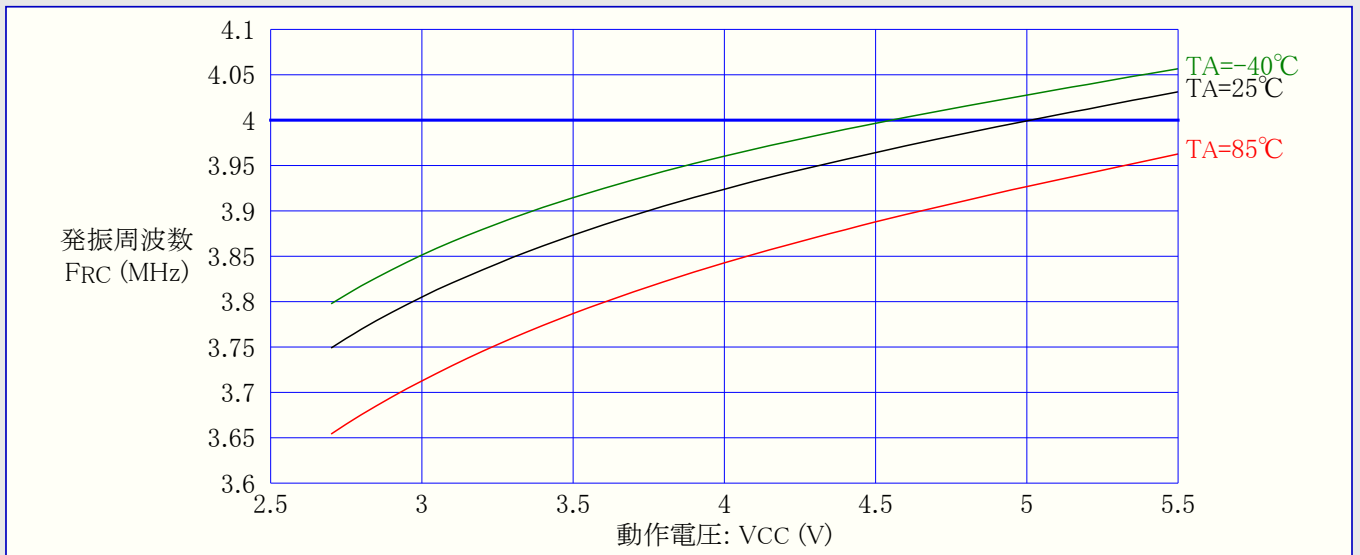


図201. 校正付き4MHz内蔵RC発振器周波数 対 発振校正(OSCCAL)値

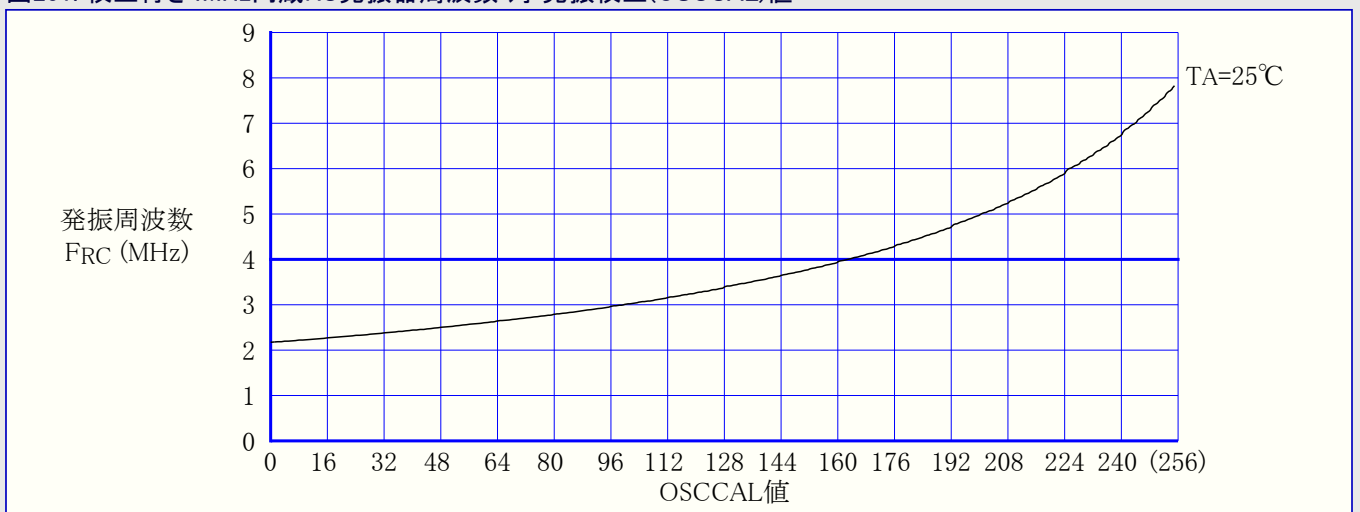


図202. 校正済み8MHz内蔵RC発振器周波数 対 動作温度

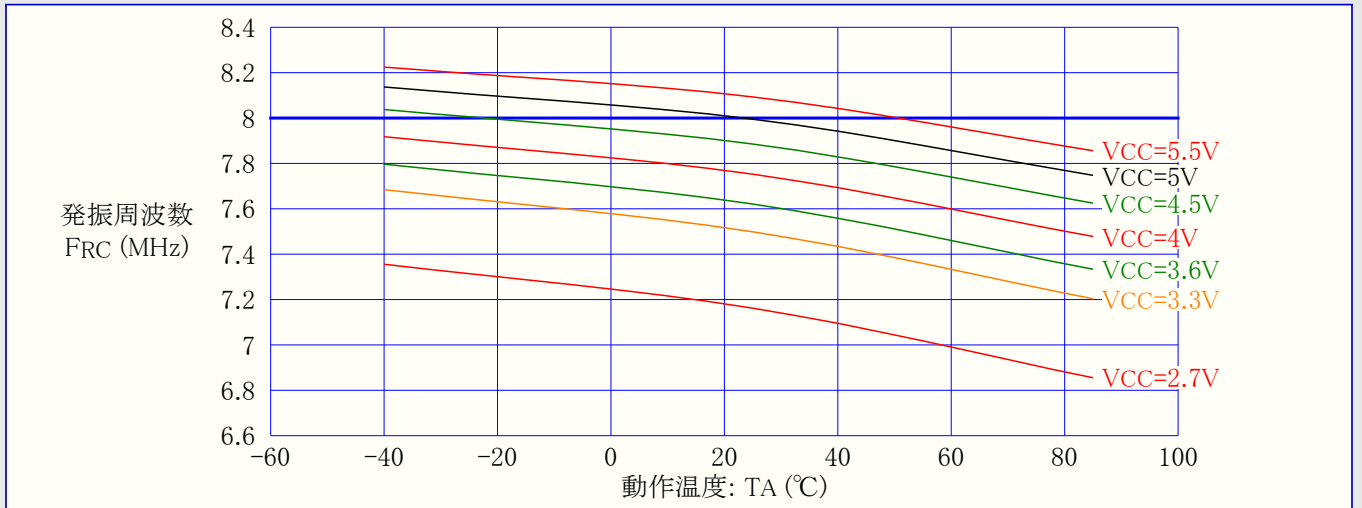


図203. 校正済み8MHz内蔵RC発振器周波数 対 動作電圧

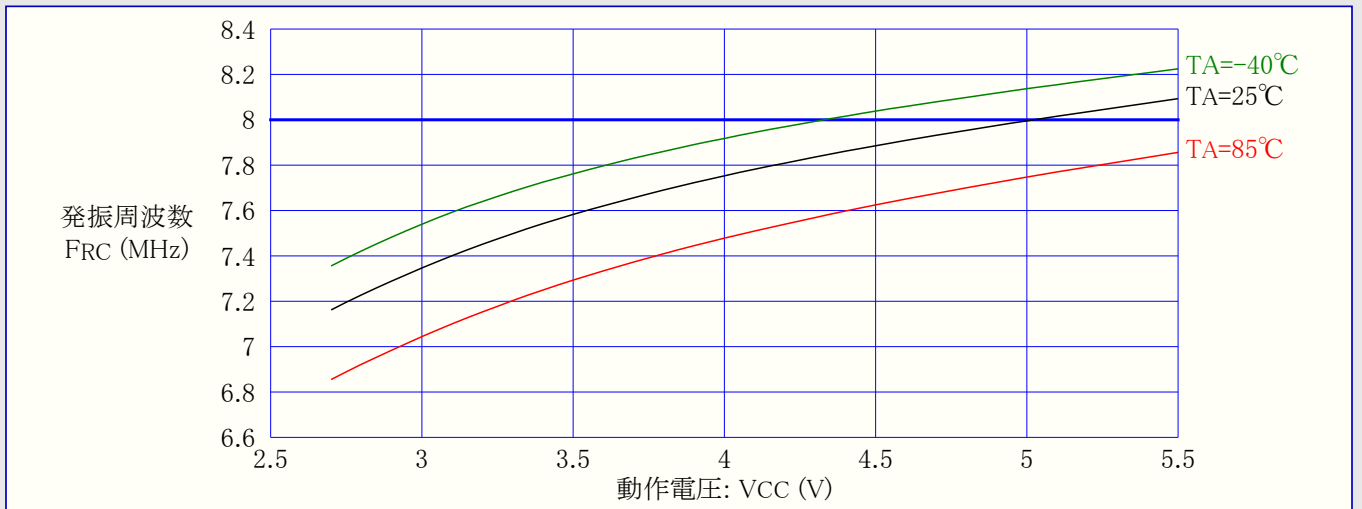
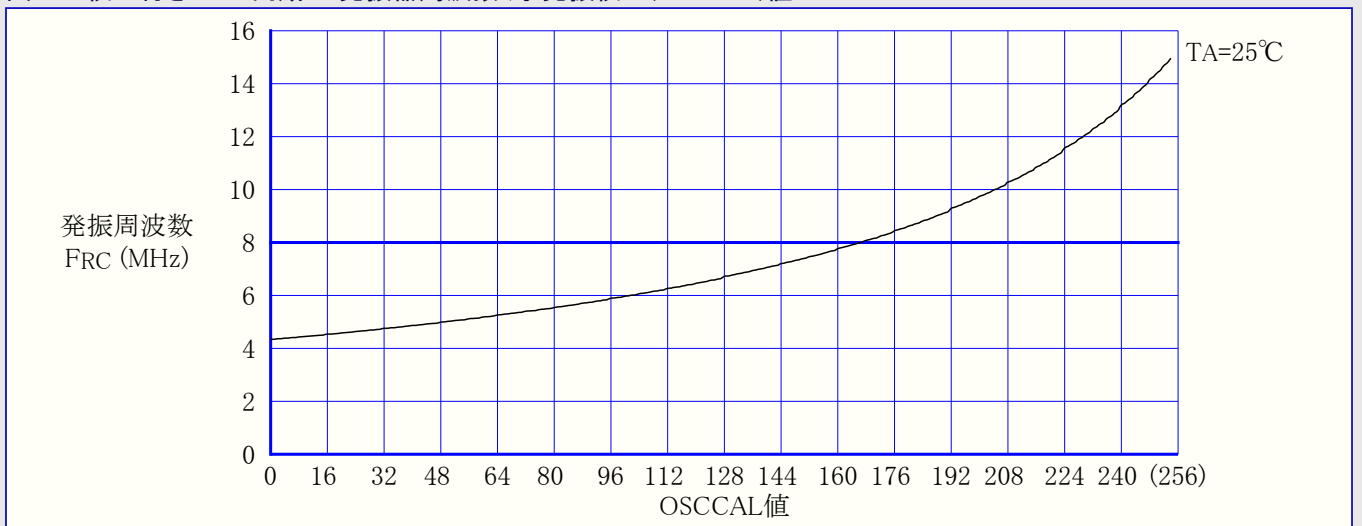


図204. 校正付き8MHz内蔵RC発振器周波数 対 発振校正(OSCCAL)値



周辺機能部消費電流

図205. 低電圧検出器(BOD)消費電流 対 動作電圧

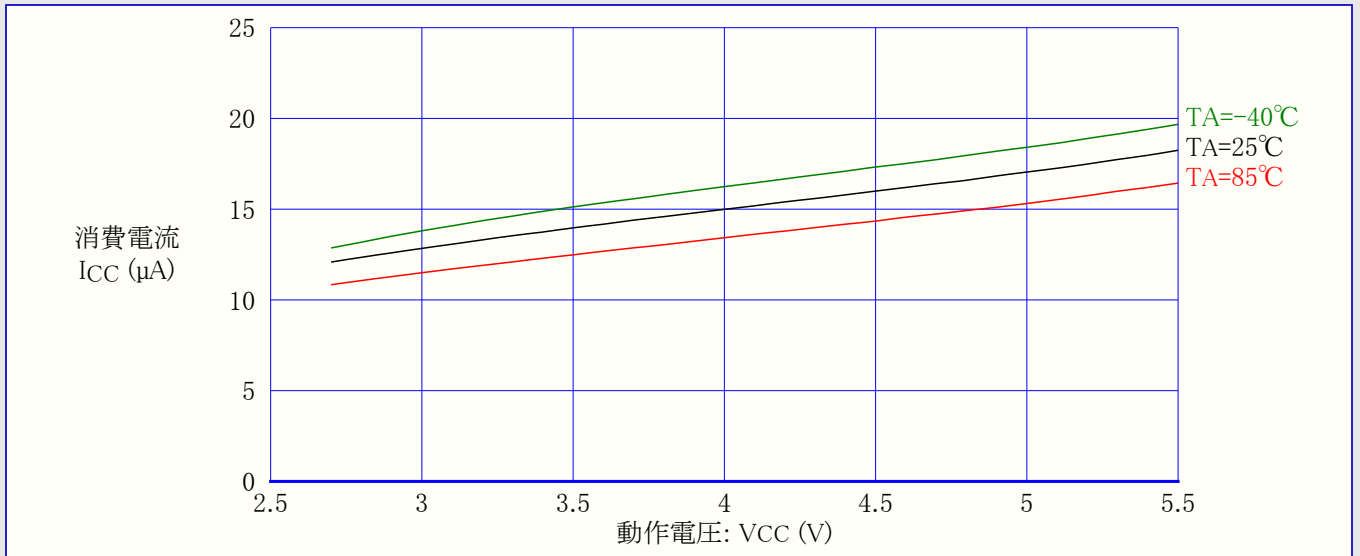


図206. A/D変換器消費電流 対 動作電圧 (変換クロック=50kHz)

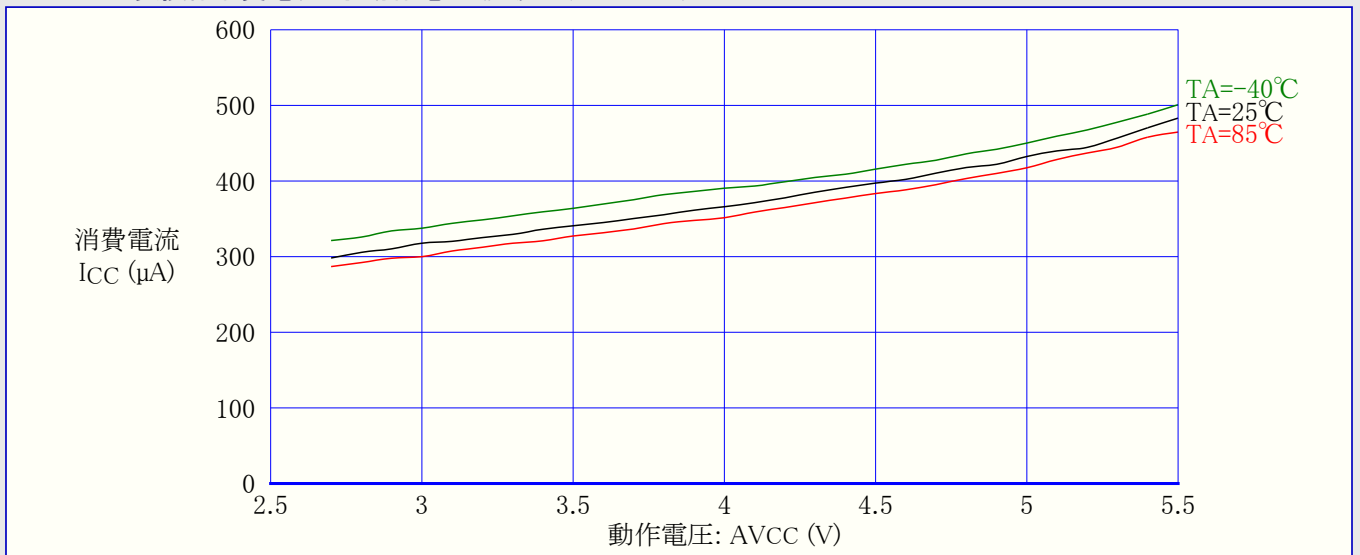


図207. 外部基準電圧(AREF)電流 対 動作電圧

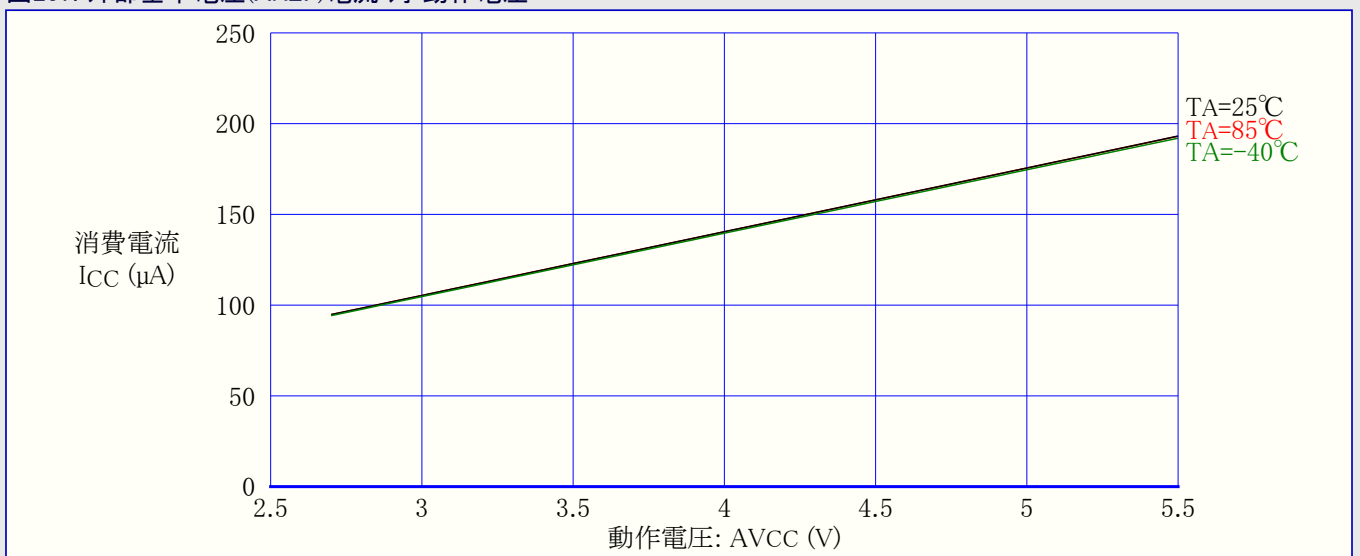


図208. アナログ比較器消費電流 対 動作電圧

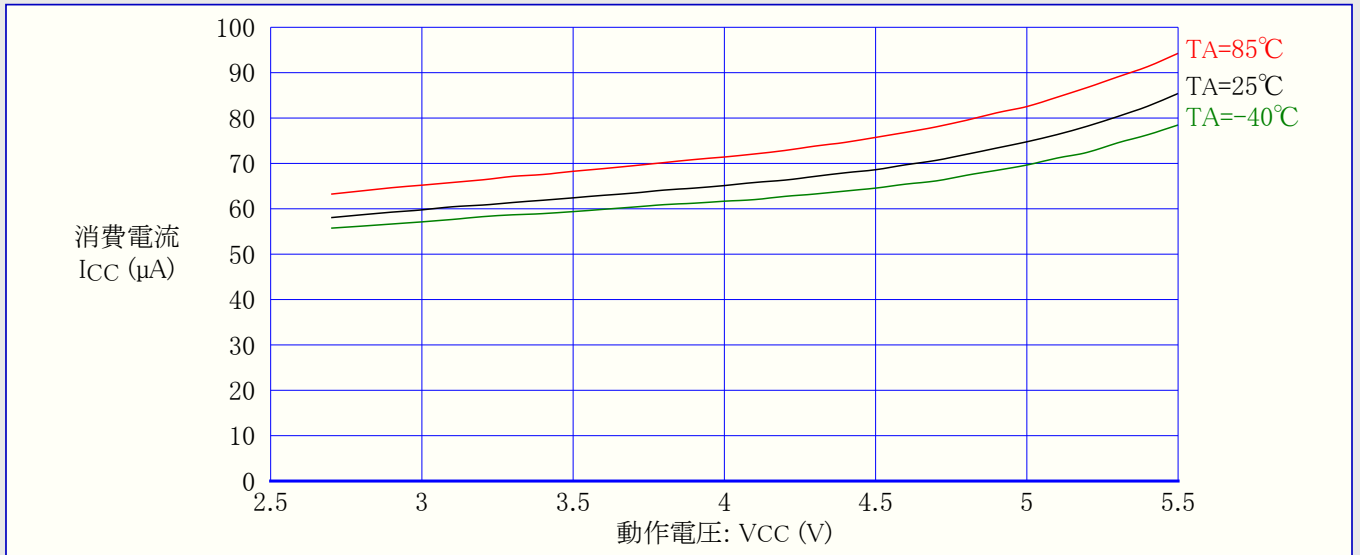
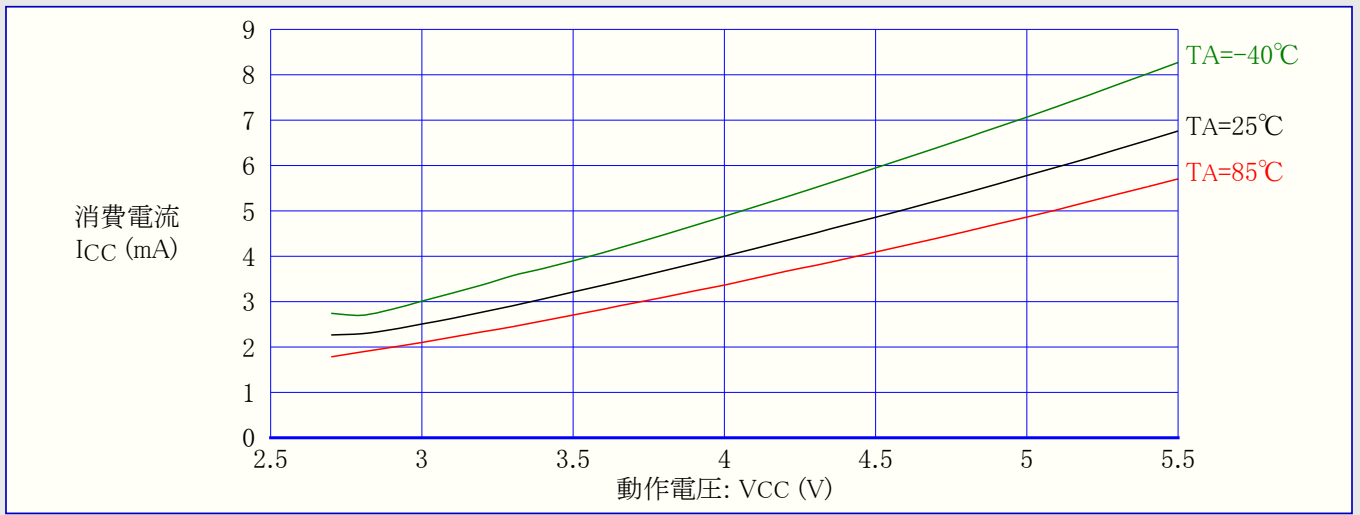


図209. プログラミング電流 対 動作電圧



リセット消費電流とリセットパルス幅

図210. 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)

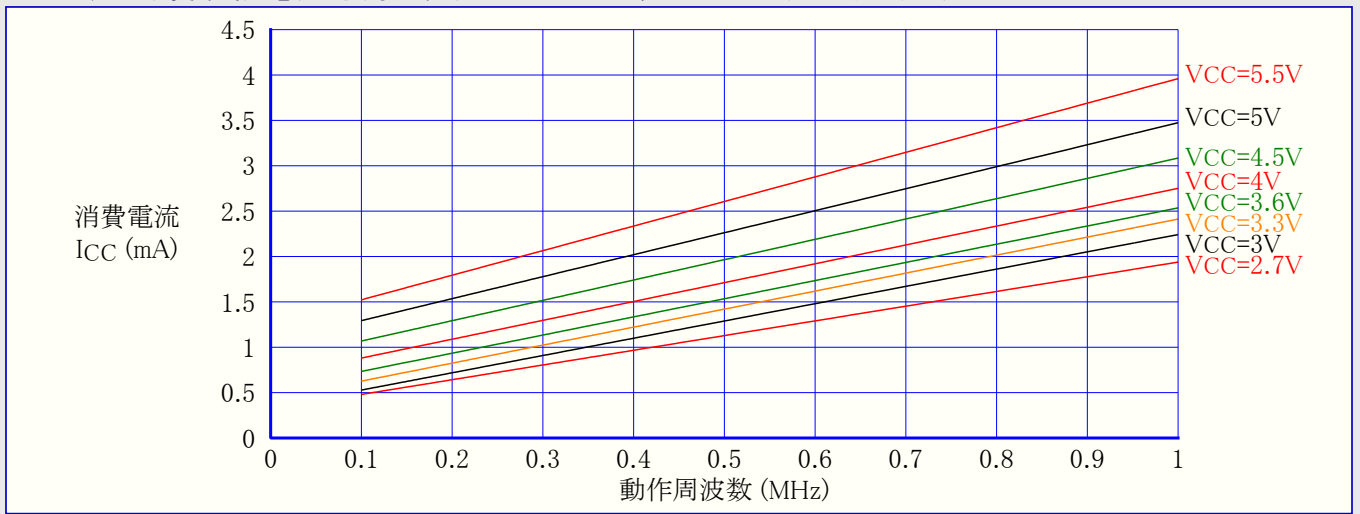


図211. リセット消費(供給)電流 対 周波数 (1MHz~20MHz、RESETプルアップ電流を除く)

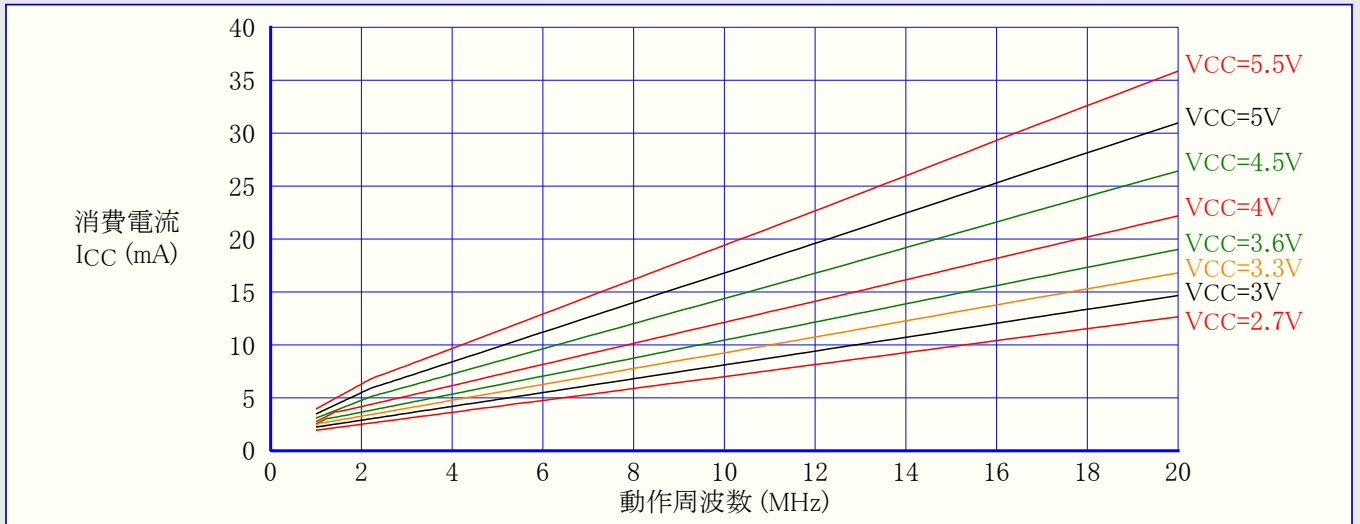


図212. RESETプルアップ抵抗電流 対 入力電圧 (VCC=5V)

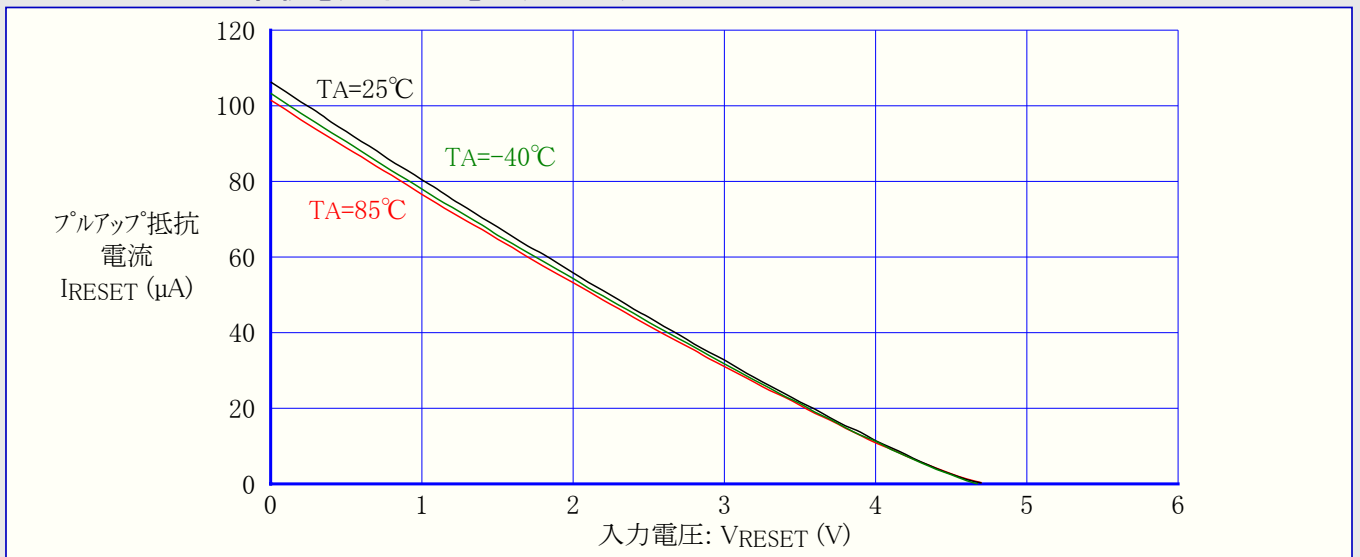


図213. RESETプルアップ抵抗電流 対 入力電圧 (VCC=2.7V)

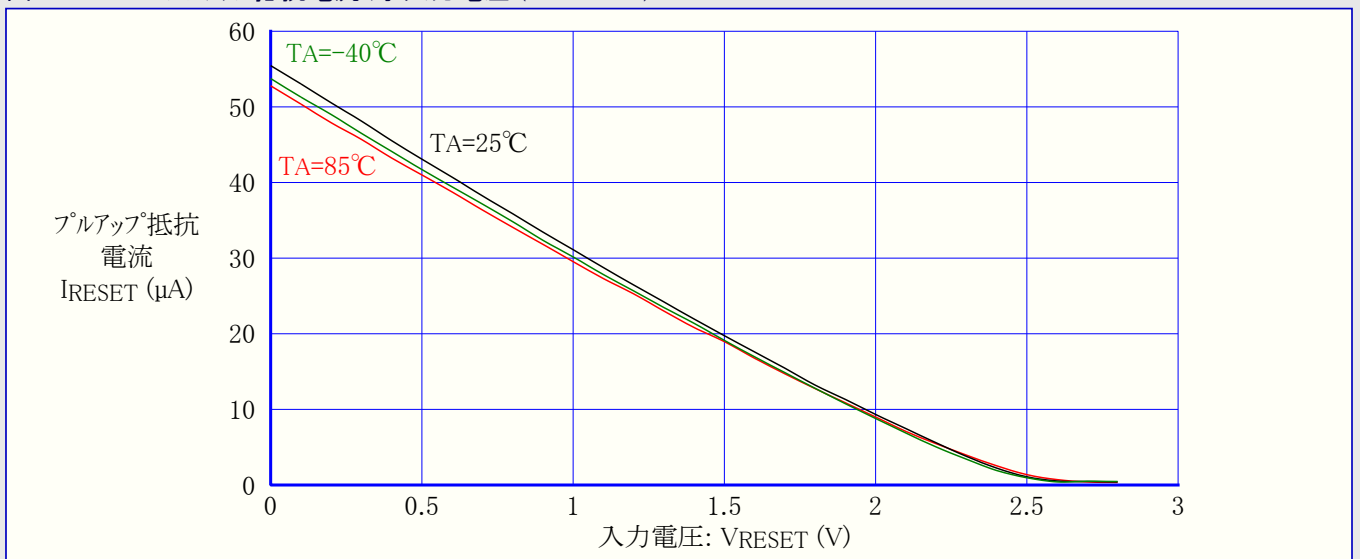




図214. RESET入力閾値(スレッショルド)電圧 対 動作電圧 ( $V_{IH}$ , 1読み値)

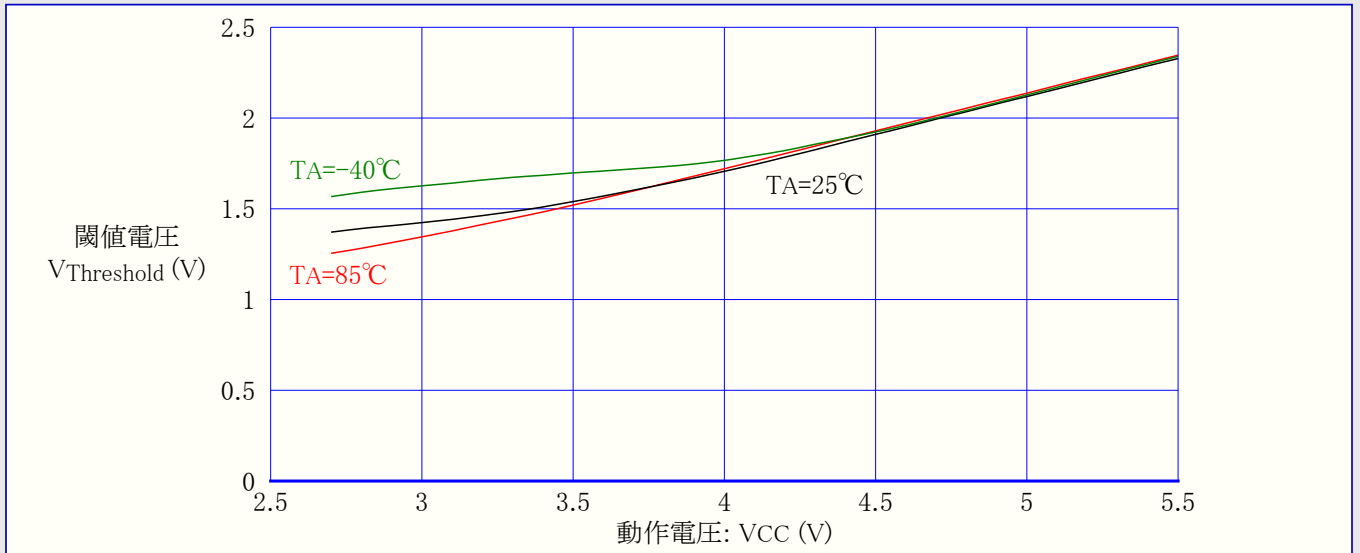


図215. RESET入力閾値(スレッショルド)電圧 対 動作電圧 ( $V_{IL}$ , 0読み値)

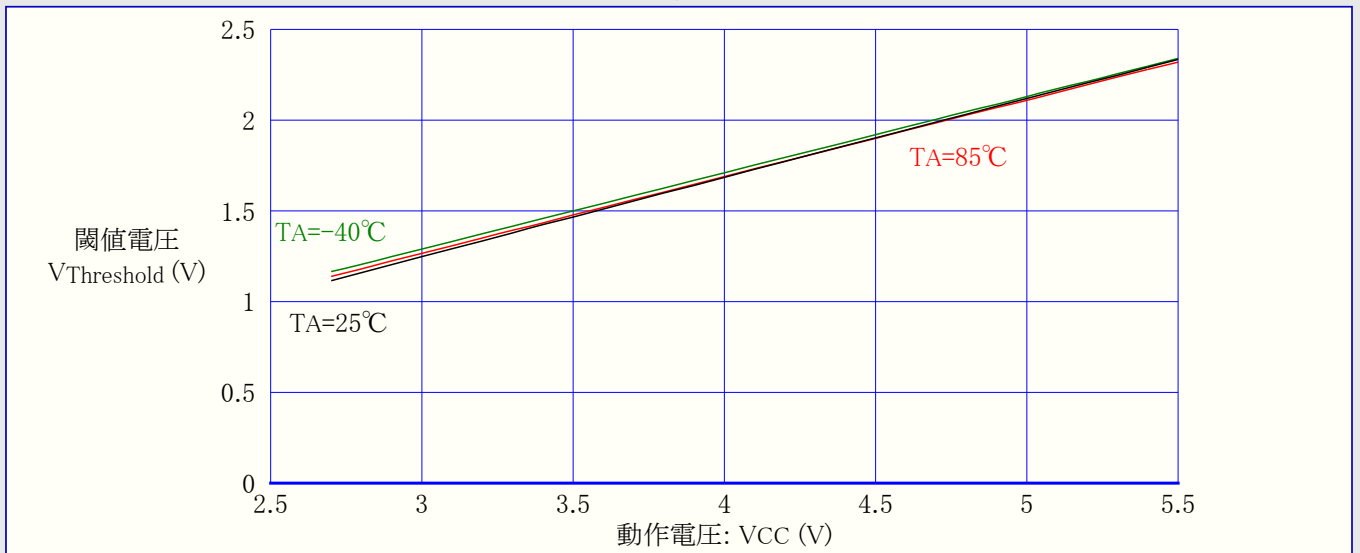


図216. RESET入力ヒステリシス電圧 対 動作電圧

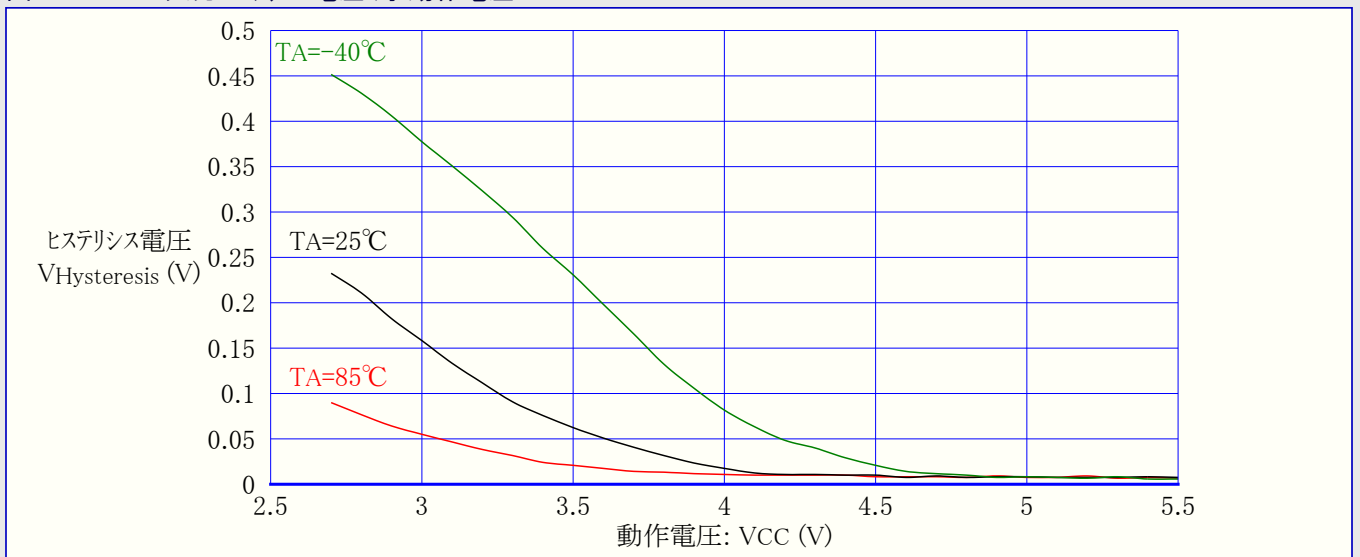
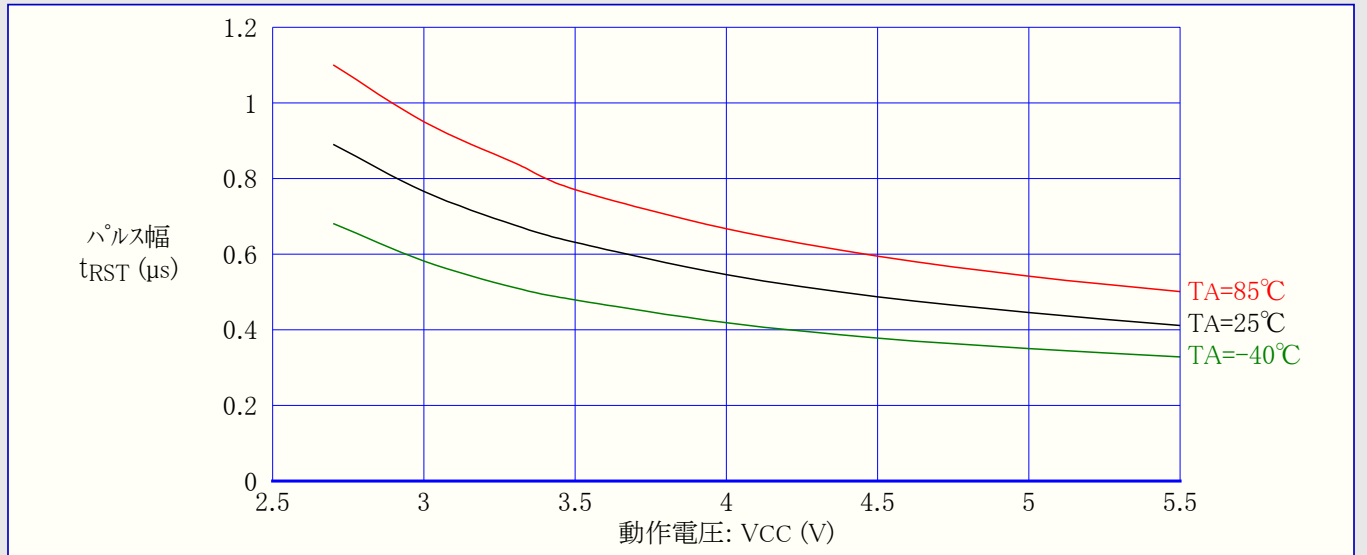


図217. リセット パルス幅 対 動作電圧 (1MHz外部クロック)



# レジスタ要約

## 拡張I/Oレジスタ領域

| 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   | 頁   |     |
|---------|--------|-----------------------------------|--------|--------|--------|---------------------------------|--------|--------|--------|-----|-----|
| (~\$FF) | 予約     |                                   |        |        |        |                                 |        |        |        |     |     |
| (\$9F)  | 予約     |                                   |        |        |        |                                 |        |        |        |     |     |
| (\$9E)  | 予約     |                                   |        |        |        |                                 |        |        |        |     |     |
| (\$9D)  | UCSR1C | -                                 | UMSEL1 | UPM11  | UPM10  | USBS1                           | UCSZ11 | UCSZ10 | UCPOL1 | 129 |     |
| (\$9C)  | UDR1   | USART1 データレジスタ                    |        |        |        |                                 |        |        |        |     | 128 |
| (\$9B)  | UCSR1A | RXC1                              | TXC1   | UDRE1  | FE1    | DOR1                            | UPE1   | U2X1   | MPCM1  | 128 |     |
| (\$9A)  | UCSR1B | RXCIE1                            | TXCIE1 | UDRIE1 | RXEN1  | TXEN1                           | UCSZ12 | RXB81  | TXB81  | 129 |     |
| (\$99)  | UBRR1L | USART1 ホールレイトレジスタ下位バイト (UBRR17~0) |        |        |        |                                 |        |        |        |     |     |
| (\$98)  | UBRR1H | -                                 | -      | -      | -      | USART1 ホールレイトレジスタ上位 (UBRR111~8) |        |        |        | 130 |     |
| (\$97)  | 予約     |                                   |        |        |        |                                 |        |        |        |     |     |
| (\$96)  | 予約     |                                   |        |        |        |                                 |        |        |        |     |     |
| (\$95)  | UCSR0C | -                                 | UMSEL0 | UPM01  | UPM00  | USBS0                           | UCSZ01 | UCSZ00 | UCPOL0 | 129 |     |
| (\$94)  | 予約     |                                   |        |        |        |                                 |        |        |        |     |     |
| (\$93)  | 予約     |                                   |        |        |        |                                 |        |        |        |     |     |
| (\$92)  | 予約     |                                   |        |        |        |                                 |        |        |        |     |     |
| (\$91)  | 予約     |                                   |        |        |        |                                 |        |        |        |     |     |
| (\$90)  | UBRR0H | -                                 | -      | -      | -      | USART0 ホールレイトレジスタ上位 (UBRR011~8) |        |        |        | 130 |     |
| (\$8F)  | 予約     |                                   |        |        |        |                                 |        |        |        |     |     |
| (\$8E)  | 予約     |                                   |        |        |        |                                 |        |        |        |     |     |
| (\$8D)  | 予約     |                                   |        |        |        |                                 |        |        |        |     |     |
| (\$8C)  | TCCR3C | FOC3A                             | FOC3B  | FOC3C  | -      | -                               | -      | -      | -      | 91  |     |
| (\$8B)  | TCCR3A | COM3A1                            | COM3A0 | COM3B1 | COM3B0 | COM3C1                          | COM3C0 | WGM31  | WGM30  | 88  |     |
| (\$8A)  | TCCR3B | ICNC3                             | ICES3  | -      | WGM33  | WGM32                           | CS32   | CS31   | CS30   | 90  |     |
| (\$89)  | TCNT3H | タイマ/カウンタ3 上位バイト                   |        |        |        |                                 |        |        |        |     |     |
| (\$88)  | TCNT3L | タイマ/カウンタ3 下位バイト                   |        |        |        |                                 |        |        |        |     | 91  |
| (\$87)  | OCR3AH | タイマ/カウンタ3 比較Aレジスタ上位バイト            |        |        |        |                                 |        |        |        |     |     |
| (\$86)  | OCR3AL | タイマ/カウンタ3 比較Aレジスタ下位バイト            |        |        |        |                                 |        |        |        |     | 92  |
| (\$85)  | OCR3BH | タイマ/カウンタ3 比較Bレジスタ上位バイト            |        |        |        |                                 |        |        |        |     |     |
| (\$84)  | OCR3BL | タイマ/カウンタ3 比較Bレジスタ下位バイト            |        |        |        |                                 |        |        |        |     | 92  |
| (\$83)  | OCR3CH | タイマ/カウンタ3 比較Cレジスタ上位バイト            |        |        |        |                                 |        |        |        |     |     |
| (\$82)  | OCR3CL | タイマ/カウンタ3 比較Cレジスタ下位バイト            |        |        |        |                                 |        |        |        |     | 93  |
| (\$81)  | ICR3H  | タイマ/カウンタ3 捕獲レジスタ上位バイト             |        |        |        |                                 |        |        |        |     |     |
| (\$80)  | ICR3L  | タイマ/カウンタ3 捕獲レジスタ下位バイト             |        |        |        |                                 |        |        |        |     | 93  |
| (\$7F)  | 予約     |                                   |        |        |        |                                 |        |        |        |     |     |
| (\$7E)  | 予約     |                                   |        |        |        |                                 |        |        |        |     |     |
| (\$7D)  | ETIMSK | -                                 | -      | TICIE3 | OCIE3A | OCIE3B                          | TOIE3  | OCIE3C | OCIE1C | 94  |     |
| (\$7C)  | ETIFR  | -                                 | -      | ICF3   | OCF3A  | OCF3B                           | TOV3   | OCF3C  | OCF1C  | 95  |     |
| (\$7B)  | 予約     |                                   |        |        |        |                                 |        |        |        |     |     |
| (\$7A)  | TCCR1C | FOC1A                             | FOC1B  | FOC1C  | -      | -                               | -      | -      | -      | 91  |     |
| (\$79)  | OCR1CH | タイマ/カウンタ1 比較Cレジスタ上位バイト            |        |        |        |                                 |        |        |        |     |     |
| (\$78)  | OCR1CL | タイマ/カウンタ1 比較Cレジスタ下位バイト            |        |        |        |                                 |        |        |        |     | 92  |
| (\$77)  | 予約     |                                   |        |        |        |                                 |        |        |        |     |     |
| (\$76)  | 予約     |                                   |        |        |        |                                 |        |        |        |     |     |
| (\$75)  | 予約     |                                   |        |        |        |                                 |        |        |        |     |     |
| (\$74)  | TWCR   | TWINT                             | TWEA   | TWSTA  | TWSTO  | TWWC                            | TWEN   | -      | TWIE   | 138 |     |
| (\$73)  | TWDR   | 2線直列インターフェースデータレジスタ               |        |        |        |                                 |        |        |        |     | 139 |
| (\$72)  | TWAR   | TWA6                              | TWA5   | TWA4   | TWA3   | TWA2                            | TWA1   | TWA0   | TWGCE  | 139 |     |
| (\$71)  | TWSR   | TWS7                              | TWS6   | TWS5   | TWS4   | TWS3                            | -      | TWPS1  | TWPS0  | 139 |     |
| (\$70)  | TWBR   | 2線直列インターフェースビット速度レジスタ             |        |        |        |                                 |        |        |        |     | 138 |
| (\$6F)  | OSCCAL | 内蔵RC発振器 発振校正値レジスタ                 |        |        |        |                                 |        |        |        |     | 27  |
| (\$6E)  | 予約     |                                   |        |        |        |                                 |        |        |        |     |     |
| (\$6D)  | XMCRA  | -                                 | SRL2   | SRL1   | SRL0   | SRW01                           | SRW00  | SRW11  | -      | 21  |     |
| (\$6C)  | XMCRB  | XMBK                              | -      | -      | -      | -                               | XMM2   | XMM1   | XMM0   | 22  |     |
| (\$6B)  | 予約     |                                   |        |        |        |                                 |        |        |        |     |     |
| (\$6A)  | EICRA  | ISC31                             | ISC30  | ISC21  | ISC20  | ISC11                           | ISC10  | ISC01  | ISC00  | 59  |     |
| (\$69)  | 予約     |                                   |        |        |        |                                 |        |        |        |     |     |
| (\$68)  | SPMCSR | SPMIE                             | RWWSB  | -      | RWWSRE | BLBSET                          | PGWRT  | PGERS  | SPMEN  | 181 |     |
| (\$67)  | 予約     |                                   |        |        |        |                                 |        |        |        |     |     |
| (\$66)  | 予約     |                                   |        |        |        |                                 |        |        |        |     |     |
| (\$65)  | PORTG  | -                                 | -      | -      | PORTG4 | PORTG3                          | PORTG2 | PORTG1 | PORTG0 | 58  |     |
| (\$64)  | DDRG   | -                                 | -      | -      | DDF4   | DDF3                            | DDF2   | DDF1   | DDF0   | 58  |     |
| (\$63)  | PING   | -                                 | -      | -      | PINF4  | PINF3                           | PINF2  | PINF1  | PINF0  | 58  |     |
| (\$62)  | PORTF  | PORTF7                            | PORTF6 | PORTF5 | PORTF4 | PORTF3                          | PORTF2 | PORTF1 | PORTF0 | 57  |     |
| (\$61)  | DDRF   | DDF7                              | DDF6   | DDF5   | DDF4   | DDF3                            | DDF2   | DDF1   | DDF0   | 58  |     |
| (\$60)  | 予約     |                                   |        |        |        |                                 |        |        |        |     |     |

## I/Oレジスタ領域

| 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   | 頁            |
|-------------|--------|------------------------------------|--------|--------|--------|--------------------------------|--------|--------|--------|--------------|
| \$3F (\$5F) | SREG   | I                                  | T      | H      | S      | V                              | N      | Z      | C      | 8            |
| \$3E (\$5E) | SPH    | SP15                               | SP14   | SP13   | SP12   | SP11                           | SP10   | SP9    | SP8    | 9            |
| \$3D (\$5D) | SPL    | SP7                                | SP6    | SP5    | SP4    | SP3                            | SP2    | SP1    | SP0    |              |
| \$3C (\$5C) | XDIV   | XDIVEN                             | XDIV6  | XDIV5  | XDIV4  | XDIV3                          | XDIV2  | XDIV1  | XDIV0  | 28           |
| \$3B (\$5B) | RAMPZ  | -                                  | -      | -      | -      | -                              | -      | -      | RAMPZ0 | 10           |
| \$3A (\$5A) | EICRB  | ISC71                              | ISC70  | ISC61  | ISC60  | ISC51                          | ISC50  | ISC41  | ISC40  | 60           |
| \$39 (\$59) | EIMSK  | INT7                               | INT6   | INT5   | INT4   | INT3                           | INT2   | INT1   | INT0   | 60           |
| \$38 (\$58) | EIFR   | INTF7                              | INTF6  | INTF5  | INTF4  | INTF3                          | INTF2  | INTF1  | INTF0  | 60           |
| \$37 (\$57) | TIMSK  | OCIE2                              | TOIE2  | TICIE1 | OCIE1A | OCIE1B                         | TOIE1  | OCIE0  | TOIE0  | 109,94,70    |
| \$36 (\$56) | TIFR   | OCF2                               | TOV2   | ICF1   | OCF1A  | OCF1B                          | TOV1   | OCF0   | TOV0   | 109,95,71    |
| \$35 (\$55) | MCUCR  | SRE                                | SRW10  | SE     | SM1    | SM0                            | SM2    | IVSEL  | IVCE   | 21,29,40     |
| \$34 (\$54) | MCUCSR | JTD                                | -      | -      | JTRF   | WDRF                           | BORF   | EXTRF  | PORF   | 169,34       |
| \$33 (\$53) | TCCR0  | FOC0                               | WGM00  | COM01  | COM00  | WGM01                          | CS02   | CS01   | CS00   | 69           |
| \$32 (\$52) | TCNT0  | タイマ/カウンタ0                          |        |        |        |                                |        |        |        | 70           |
| \$31 (\$51) | OCR0   | タイマ/カウンタ0 比較レジスタ                   |        |        |        |                                |        |        |        | 70           |
| \$30 (\$50) | ASSR   | -                                  | -      | -      | -      | AS0                            | TCN0UB | OCR0UB | TCR0UB | 71           |
| \$2F (\$4F) | TCCR1A | COM1A1                             | COM1A0 | COM1B1 | COM1B0 | FOC1A                          | FOC1B  | WGM11  | WGM10  | 88           |
| \$2E (\$4E) | TCCR1B | ICNC1                              | ICES1  | -      | WGM13  | WGM12                          | CS12   | CS11   | CS10   | 90           |
| \$2D (\$4D) | TCNT1H | タイマ/カウンタ1 上位バイト                    |        |        |        |                                |        |        |        | 91           |
| \$2C (\$4C) | TCNT1L | タイマ/カウンタ1 下位バイト                    |        |        |        |                                |        |        |        |              |
| \$2B (\$4B) | OCR1AH | タイマ/カウンタ1 比較レジスタ 上位バイト             |        |        |        |                                |        |        |        | 92           |
| \$2A (\$4A) | OCR1AL | タイマ/カウンタ1 比較レジスタ 下位バイト             |        |        |        |                                |        |        |        |              |
| \$29 (\$49) | OCR1BH | タイマ/カウンタ1 比較レジスタ 上位バイト             |        |        |        |                                |        |        |        | 92           |
| \$28 (\$48) | OCR1BL | タイマ/カウンタ1 比較レジスタ 下位バイト             |        |        |        |                                |        |        |        |              |
| \$27 (\$47) | ICR1H  | タイマ/カウンタ1 捕獲レジスタ 上位バイト             |        |        |        |                                |        |        |        | 93           |
| \$26 (\$46) | ICR1L  | タイマ/カウンタ1 捕獲レジスタ 下位バイト             |        |        |        |                                |        |        |        |              |
| \$25 (\$45) | TCCR2  | FOC2                               | WGM20  | COM21  | COM20  | WGM21                          | CS22   | CS21   | CS20   | 107          |
| \$24 (\$44) | TCNT2  | タイマ/カウンタ2                          |        |        |        |                                |        |        |        | 108          |
| \$23 (\$43) | OCR2   | タイマ/カウンタ2 比較レジスタ                   |        |        |        |                                |        |        |        | 108          |
| \$22 (\$42) | OCDR   | IDRD/<br>OCDR7                     | OCDR6  | OCDR5  | OCDR4  | OCDR3                          | OCDR2  | OCDR1  | OCDR0  | 167          |
| \$21 (\$41) | WDTCSR | -                                  | -      | -      | WDCE   | WDE                            | WDP2   | WDP1   | WDP0   | 35           |
| \$20 (\$40) | SFIOR  | TSM                                | -      | -      | -      | ACME                           | PUD    | PSR0   | PSR321 | 153,45,73,98 |
| \$1F (\$3F) | EEARH  | -                                  | -      | -      | -      | EEAR11                         | EEAR10 | EEAR9  | EEAR8  | 14           |
| \$1E (\$3E) | EEARL  | -                                  | -      | -      | -      | EEPROMアドレスレジスタ 下位バイト (EEAR7~0) |        |        |        |              |
| \$1D (\$3D) | EEDR   | EEPROMデータレジスタ                      |        |        |        |                                |        |        |        | 14           |
| \$1C (\$3C) | EEDR   | -                                  | -      | -      | -      | EERIE                          | EEMWE  | EWE    | EERE   | 14           |
| \$1B (\$3B) | PORTA  | PORTA7                             | PORTA6 | PORTA5 | PORTA4 | PORTA3                         | PORTA2 | PORTA1 | PORTA0 | 56           |
| \$1A (\$3A) | DDRA   | DDA7                               | DDA6   | DDA5   | DDA4   | DDA3                           | DDA2   | DDA1   | DDA0   | 56           |
| \$19 (\$39) | PINA   | PINA7                              | PINA6  | PINA5  | PINA4  | PINA3                          | PINA2  | PINA1  | PINA0  | 56           |
| \$18 (\$38) | PORTB  | PORTB7                             | PORTB6 | PORTB5 | PORTB4 | PORTB3                         | PORTB2 | PORTB1 | PORTB0 | 56           |
| \$17 (\$37) | DDRB   | DDB7                               | DDB6   | DDB5   | DDB4   | DDB3                           | DDB2   | DDB1   | DDB0   | 56           |
| \$16 (\$36) | PINB   | PINB7                              | PINB6  | PINB5  | PINB4  | PINB3                          | PINB2  | PINB1  | PINB0  | 56           |
| \$15 (\$35) | PORTC  | PORTC7                             | PORTC6 | PORTC5 | PORTC4 | PORTC3                         | PORTC2 | PORTC1 | PORTC0 | 56           |
| \$14 (\$34) | DDRC   | DDC7                               | DDC6   | DDC5   | DDC4   | DDC3                           | DDC2   | DDC1   | DDC0   | 56           |
| \$13 (\$33) | PINC   | PINC7                              | PINC6  | PINC5  | PINC4  | PINC3                          | PINC2  | PINC1  | PINC0  | 57           |
| \$12 (\$32) | PORTD  | PORTD7                             | PORTD6 | PORTD5 | PORTD4 | PORTD3                         | PORTD2 | PORTD1 | PORTD0 | 57           |
| \$11 (\$31) | DDRD   | DDD7                               | DDD6   | DDD5   | DDD4   | DDD3                           | DDD2   | DDD1   | DDD0   | 57           |
| \$10 (\$30) | PIND   | PIND7                              | PIND6  | PIND5  | PIND4  | PIND3                          | PIND2  | PIND1  | PIND0  | 57           |
| \$0F (\$2F) | SPDR   | SPI データレジスタ                        |        |        |        |                                |        |        |        | 114          |
| \$0E (\$2E) | SPSR   | SPIF                               | WCOL   | -      | -      | -                              | -      | -      | SPI2X  | 114          |
| \$0D (\$2D) | SPCR   | SPIE                               | SPE    | DORD   | MSTR   | CPOL                           | CPHA   | SPR1   | SPR0   | 113          |
| \$0C (\$2C) | UDR0   | USART0 データレジスタ                     |        |        |        |                                |        |        |        | 128          |
| \$0B (\$2B) | UCSR0A | RXC0                               | TXC0   | UDRE0  | FE0    | DOR0                           | UPE0   | U2X0   | MPCM0  | 128          |
| \$0A (\$2A) | UCSR0B | RXCIE0                             | TXCIE0 | UDRIE0 | RXEN0  | TXEN0                          | UCSZ02 | RXB80  | TXB80  | 129          |
| \$09 (\$29) | UBRR0L | USART0 ボーレートレジスタ 下位バイト (UBRR07~0)  |        |        |        |                                |        |        |        | 130          |
| \$08 (\$28) | ACSR   | ACD                                | ACBG   | ACO    | ACI    | ACIE                           | ACIC   | ACIS1  | ACIS0  | 152          |
| \$07 (\$27) | ADMUX  | REFS1                              | REFS0  | ADLAR  | MUX4   | MUX3                           | MUX2   | MUX1   | MUX0   | 160          |
| \$06 (\$26) | ADCSRA | ADEN                               | ADSC   | ADFR   | ADIF   | ADIE                           | ADPS2  | ADPS1  | ADPS0  | 161          |
| \$05 (\$25) | ADCH   | A/Dデータレジスタ 上位バイト (ADC9~8またはADC9~2) |        |        |        |                                |        |        |        | 162          |
| \$04 (\$24) | ADCL   | A/Dデータレジスタ 下位バイト (ADC7~0またはADC1~0) |        |        |        |                                |        |        |        |              |
| \$03 (\$23) | PORTE  | PORTE7                             | PORTE6 | PORTE5 | PORTE4 | PORTE3                         | PORTE2 | PORTE1 | PORTE0 | 57           |
| \$02 (\$22) | DDRE   | DDE7                               | DDE6   | DDE5   | DDE4   | DDE3                           | DDE2   | DDE1   | DDE0   | 57           |
| \$01 (\$21) | PINE   | PINE7                              | PINE6  | PINE5  | PINE4  | PINE3                          | PINE2  | PINE1  | PINE0  | 57           |
| \$00 (\$20) | PINF   | PINF7                              | PINF6  | PINF5  | PINF4  | PINF3                          | PINF2  | PINF1  | PINF0  | 58           |

注：将来のデバイスとの共通性のため、予約ビットへ書く場合は0を書くべきです。予約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。

- いくつかの状態ビットはそれらに論理1を書くことによって解除(0)されます。CBI,SBI命令はそのI/Oレジスタ内の全ビットを操作し、設定(1)として読んだどのフラグにも1を書き戻します。従ってフラグを解除(0)します。CBI,SBI命令は\$00~\$1FのI/Oレジスタだけで動作します。



# 命令要約

| ニーモニック           | オペラント* | 意味                      | 動作  | フラグ             | クロック  |
|------------------|--------|-------------------------|---|-----------------|-------|
| <b>算術、論理演算命令</b> |        |                         |   |                 |       |
| 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     |
| ADIW             | Rd,K6  | 即値の語(ワード)長加算            | $RdH:RdL \leftarrow RdH:RdL + K6$               | I,T,H,S,V,N,Z,C | 2     |
| 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     |
| SBIW             | Rd,K6  | 即値の語(ワード)長減算            | $RdH:RdL \leftarrow RdH:RdL - K6$               | I,T,H,S,V,N,Z,C | 2     |
| 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,I | 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     |
| MUL              | Rd,Rr  | 符号なし間の乗算                | $R1:R0 \leftarrow Rd \times Rr$ (U×U)           | I,T,H,S,V,N,Z,C | 2     |
| MULS             | Rd,Rr  | 符号付き間の乗算                | $R1:R0 \leftarrow Rd \times Rr$ (S×S)           | I,T,H,S,V,N,Z,C | 2     |
| MULSU            | Rd,Rr  | 符号付きと符号なしの乗算            | $R1:R0 \leftarrow Rd \times Rr$ (S×U)           | I,T,H,S,V,N,Z,C | 2     |
| FMUL             | Rd,Rr  | 符号なし間の固定小数点乗算           | $R1:R0 \leftarrow (Rd \times Rr) \ll 1$ (U×U)   | I,T,H,S,V,N,Z,C | 2     |
| FMULS            | Rd,Rr  | 符号付き間の固定小数点乗算           | $R1:R0 \leftarrow (Rd \times Rr) \ll 1$ (S×S)   | I,T,H,S,V,N,Z,C | 2     |
| FMULSU           | Rd,Rr  | 符号付きと符号なしの固定小数点乗算       | $R1:R0 \leftarrow (Rd \times Rr) \ll 1$ (S×U)   | I,T,H,S,V,N,Z,C | 2     |
| <b>分岐命令</b>      |        |                         |   |                 |       |
| RJMP             | k      | 相対無条件分岐                 | $PC \leftarrow PC + k + 1$                      | I,T,H,S,V,N,Z,C | 2     |
| IJMP             |        | Zレジスタ間接無条件分岐            | $PC \leftarrow Z$                               | I,T,H,S,V,N,Z,C | 2     |
| JMP              | k      | 絶対無条件分岐                 | $PC \leftarrow k$                               | I,T,H,S,V,N,Z,C | 3     |
| 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     |
| 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     |
| CALL             | k      | 絶対サブルーチン呼び出し            | $STACK \leftarrow PC, PC \leftarrow k$          | I,T,H,S,V,N,Z,C | 4     |
| RET              |        | サブルーチンからの復帰             | $PC \leftarrow STACK$                           | I,T,H,S,V,N,Z,C | 4     |
| RETI             |        | 割り込みからの復帰               | $PC \leftarrow STACK$                           | I,T,H,S,V,N,Z,C | 4     |
| 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 |
| SBRB             | 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 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 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   |

K6, K : 6, 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(7,12,16ビット) q : 符号なし6ビット定数(変位) 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 |
|----------------|--------|-----------------------|--|------------------------|------|
| <b>データ移動命令</b> |        |                       |  |                        |      |
| 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    |
| MOVW           | Rd,Rr  | 汎用レジスタ対間の複写           | $Rd+1:Rd \leftarrow Rr+1:Rr$                                       | 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        | 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    |
| 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    |
| 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        | 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    |
| 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    |
| LDD            | Rd,Y+q | 変位付きYレジスタ間接での取得       | $Rd \leftarrow (Y + q)$  | I,T,H,S,V,N,Z,C        | 2    |
| 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        | 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    |
| 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    |
| LDD            | Rd,Z+q | 変位付きZレジスタ間接での取得       | $Rd \leftarrow (Z + q)$  | I,T,H,S,V,N,Z,C        | 2    |
| 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        | 2    |
| 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        | 2    |
| 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        | 2    |
| 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        | 2    |
| 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    |
| STD            | Y+q,Rr | 変位付きYレジスタ間接での設定       | $(Y + q) \leftarrow Rr$  | 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        | 2    |
| 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        | 2    |
| 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    |
| STD            | Z+q,Rr | 変位付きZレジスタ間接での設定       | $(Z + q) \leftarrow Rr$  | 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        | 2    |
| LPM            |        | プログラム領域からZレジスタ間接での取得  | $R0 \leftarrow (Z)$  | I,T,H,S,V,N,Z,C        | 3    |
| LPM            | Rd,Z   | 同上(任意のレジスタへ)          | $Rd \leftarrow (Z)$  | I,T,H,S,V,N,Z,C        | 3    |
| LPM            | Rd,Z+  | 同上(事後増加付き)            | $Rd \leftarrow (Z), Z \leftarrow Z + 1$                            | I,T,H,S,V,N,Z,C        | 3    |
| ELPM           |        | プログラム領域から拡張Zレジスタ間接で取得 | $R0 \leftarrow (RAMPZ:Z)$  | I,T,H,S,V,N,Z,C        | 3    |
| ELPM           | Rd,Z   | 同上(任意のレジスタへ)          | $Rd \leftarrow (RAMPZ:Z)$  | I,T,H,S,V,N,Z,C        | 3    |
| ELPM           | Rd,Z+  | 同上(事後増加付き)            | $Rd \leftarrow (RAMPZ:Z), RAMPZ:Z \leftarrow RAMPZ:Z + 1$          | I,T,H,S,V,N,Z,C        | 3    |
| SPM            |        | プログラム領域へZレジスタ間接での設定   | $(Z) \leftarrow R1:R0$   | I,T,H,S,V,N,Z,C        | -    |
| 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        | 2    |
| 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        | 2    |
| <b>ビット関係命令</b> |        |                       |  |                        |      |
| 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        | 2    |
| 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        | 2    |
| 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             | $Rd(n) \leftarrow Rd(n+1), n=0\sim6$                               | I,T,H,S,V,N,Z,C        | 1    |
| SWAP           | Rd     | ニブル(4ビット)上位/下位交換      | $Rd(7\sim4) \leftrightarrow Rd(3\sim0)$                            | 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)    | $SREG(s) \leftarrow 1$   | <b>I,T,H,I,Y,I,I,I</b> | 1    |
| BCLR           | s      | ステータスレジスタのビット解除(0)    | $SREG(s) \leftarrow 0$   | <b>0,0,0,0,0,0,0</b>   | 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,I,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,I,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,I,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,H,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    |

| ニーモニック  | オペラント | 意味               | 動作            | フラグ             | クロック |
|---------|-------|------------------|---------------|-----------------|------|
| MCU制御命令 |       |                  |               |                 |      |
| NOP     |       | 無操作              |               | I,T,H,S,V,N,Z,C | 1    |
| SLEEP   |       | 休止形態開始           | 休止形態参照        | 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             | 内蔵デバッグ機能専用    | I,T,H,S,V,N,Z,C | N/A  |

## 注文情報

| 速度(MHz) | 電源電圧     | 注文符号 (注1)            | 外囲器 (注2) | 動作範囲             |
|---------|----------|----------------------|----------|------------------|
| 8       | 2.7~5.5V | ATmega128L-8AU       | 64A      | 工業用 (-40°C~85°C) |
|         |          | ATmega128L-8AUR (注3) |          |                  |
|         |          | ATmega128L-8MU       | 64M1     |                  |
|         |          | ATmega128L-8MUR (注3) |          |                  |
| 16      | 4.5~5.5V | ATmega128-16AU       | 64A      |                  |
|         |          | ATmega128-16AUR (注3) | 64M1     |                  |
|         |          | ATmega128-16MU       |          |                  |
|         |          | ATmega128-16MUR (注3) |          |                  |
| 8       | 3.0~5.5V | ATmega128L-8AN       | 64A      | 拡張 (-40°C~105°C) |
|         |          | ATmega128L-8ANR (注3) | 64M1     |                  |
|         |          | ATmega128L-8MN       |          |                  |
|         |          | ATmega128L-8MNR (注3) |          |                  |
| 16      | 4.5~5.5V | ATmega128-16AN       | 64A      |                  |
|         |          | ATmega128-16ANR (注3) | 64M1     |                  |
|         |          | ATmega128-16MN       |          |                  |
|         |          | ATmega128-16MNR (注3) |          |                  |

**注1:** 有害物質使用制限に関する欧州指令(RoHS指令)適合の鉛フリー製品。またハロゲン化合物フリーで完全に安全です。

**注2:** このデバイスはウェハー(チップ単体)形状でも供給できます。最低数量と詳細な注文情報については最寄のAtmel営業所へお問い合わせください。

**注3:** テープとリール

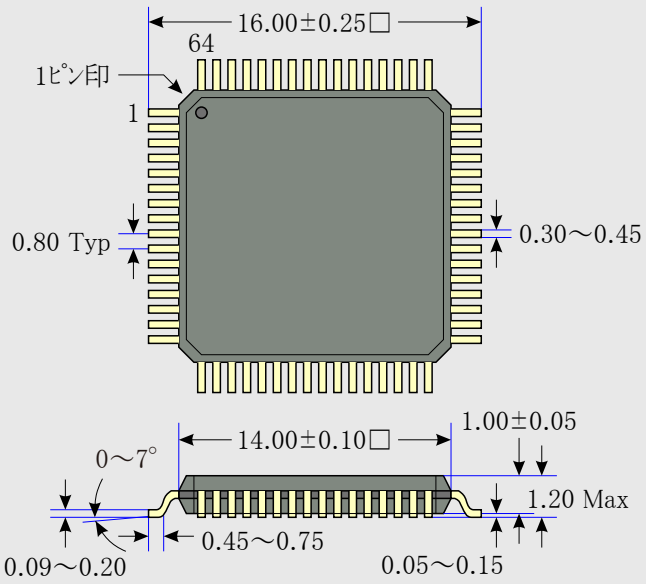
| 外囲器形式 |   |
|-------|---|
| 64A   | 64リード、14×14×1.0mm厚 プラスチック4方向平板外囲器 (TQFP)              |
| 64M1  | 64パッド、9×9×1mm 0.5mmピッチ 4方向平板リードなし/極小リード枠外囲器 (QFN/MLF) |

## 外圍器情報

### 64A

64リード、0.8mmピッチ プラスティック4方向平板外圍器 (TQFP)

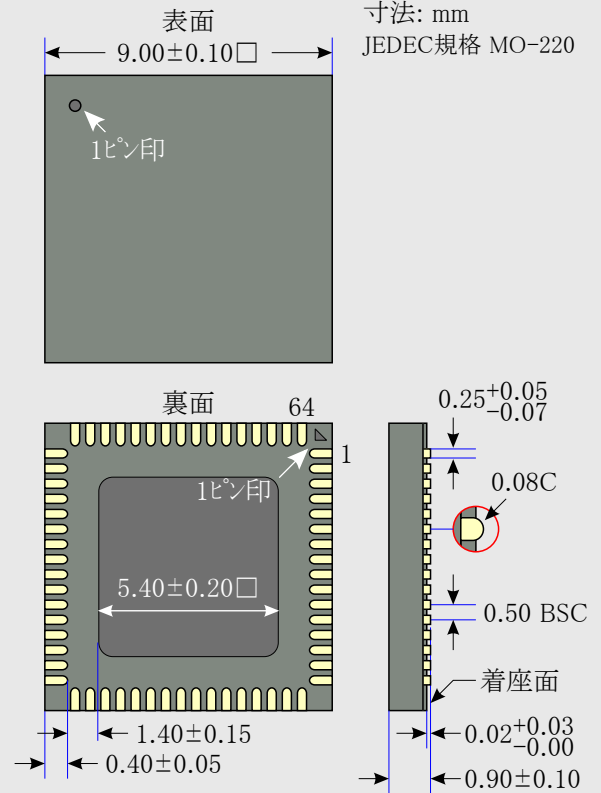
寸法: mm  
JEDEC規格 MS-026 AEB



### 64M1

64パッド、0.5mmピッチ 4方向平板リードなし/極小リード枠外圍器 (QFN/MLF)

寸法: mm  
JEDEC規格 MO-220



## データシート改訂履歴

この章内の参照頁番号はこの文書が参照されていることに注意してください。この章内の改訂番号は文書の改訂番号を参照してください。

### 改訂2467C – 02/02

1. ホートG交換機能説明の訂正  
55頁の「ホートGの交換機能」内のTOSC1とTOSC2の説明が訂正されました。
2. 改訂Fと改訂G用JTAG版番号の追加  
167頁の表100を更新しました。(訳補:改訂L版以降表削除)
3. いくつかの試験条件と特性値の追加  
次の頁の表内のいくつかのTBDを削除しました  
32頁の表19、34頁の表20、206頁の「DC特性」、207頁の表131、209頁の表134、210頁の表135。
4. 234頁の「注文情報」を訂正
5. 214頁からの「代表特性」にいくつかの特性データを追加
6. JTAGプログラミング動作からの代替抜け出し手順の削除  
203頁の「プログラミング動作からの抜け出し」を参照。
7. JTAGプログラミング経路での拡張ヒューズ<sup>1</sup> バイト アクセス法の説明追加  
205頁の「ヒューズの書き込み」と「ヒューズと施錠ビットの読み出し」を参照。

### 改訂2467D – 03/02

1. 4頁の「ATmega103互換動作」について、多くの情報を追加
2. 15頁の表2「EEPROMプログラミング時間」を更新
3. 24頁の表7、25頁の表9、表10、26頁の表12、表14、27頁の表16の起動待機時間代表値を更新
4. 36頁の表22「WDT計時終了代表値」を更新
5. 161頁の「A/D変換制御/状態レジスタ(ADCSRA)」内のADSCビットの説明を訂正
6. 159頁の「A/D変換結果」の差動結果の極性検査法の説明を改善
7. 167頁の「JTAG Ver番号」内のJTAG Ver番号を訂正
8. 182頁の「自己プログラミング時のフラッシュのアドレス指定」と183頁の「ページ消去の実行」と「ページ書き込みの実行」で、(RAMPZ使用の)SPM中のアドレス指定の説明を改善
9. 188頁の表118.下にOCDENヒューズ<sup>1</sup>に関する注意を追加
10. プログrammingの図の更新  
AVCCがプログラミング中に接続しなければならないことを反映し、189頁の図135と197頁の図144.が更新されました。193頁の図139.ヒューズ書き込みタイミングが追加されました。
11. 200頁のPROG\_PAGELOADとPROG\_PAGEREADの使用に関する注意を追加
12. 214頁からの「代表特性」に校正付きRC発振器特性を追加
13. 2線直列インターフェース(TWI)章の更新  
パワーダウン動作でのTWIの使用、低TWBRR値での主装置としてのTWI使用に関する多くの詳細がデータシートに追加されました。  
136頁の「ビット速度生成部」の最後に注意事項が追加されました。137頁の「アドレス比較部」の最後に説明が追加されました。
14. タイマ/カウンタ0と組み合わされるクロックの使用に関する注意事項の追加  
28頁の「クロック分周制御レジスタ(XDIV)」を参照。

### 改訂2467E – 05/02

1. 214頁からの「代表特性」の特性データを更新
2. 次の表を更新  
32頁の表19、34頁の表20、109頁の表68、172頁の表102、210頁の表135。
3. 発振校正値(OSCCAL)の説明更新  
2,4,8MHz発振選択用の校正バイトの利用法が説明されていませんでした。これを次の章に追加しました。  
27頁の「発振校正値レジスタ-OSCCAL」と187頁の「校正バイト」の記載が改善されました。

### 改訂2467F – 09/02

1. 64QFN/MLF外囲器の追加と234頁の注文情報更新
2. 「64Kバイト未満外部メモリの全域使用法」章の追加





3. 24頁の「既定のクロック元」章の追加
4. 文書内全てでSPMCRをSPMCSRに改名
5. 外部クロック信号使用時、周波数変更に関していくつかの制限があります。これは27頁の「外部クロック信号」と207頁の表131.で説明されます。
6. 31頁の「消費電力の最小化」章内にOCDと消費電力の関係を追加
7. WGMビット設定に関する誤植修正
  - 66頁の「高速PWM動作」(タイマ/カウンタ0)
  - 67頁の「位相基準PWM動作」(タイマ/カウンタ0)
  - 104頁の「高速PWM動作」(タイマ/カウンタ2)
  - 105頁の「位相基準PWM動作」(タイマ/カウンタ2)
8. 130頁の表81.(USART)を修正
9. 172頁の表102.(境界走査)を修正
10. 206頁の「DC特性」内のVILを更新

### 改訂2467G – 09/02

1. フラッシュメモリの書き換え可能回数(寿命)を1,000から10,000に変更

### 改訂2467H – 02/03

1. SFIORレジスタで2つの前置分周器制御ビット名を修正
2. 204頁の「フラッシュメモリ書き込み」と「EEPROM書き込み」に最初の手順としてチップ消去を追加
3. 存在しない「多目的発振器」、「32kHzクリスタル発振器」応用記述の参照を削除
4. 85頁の図52.でOCn波形を修正
5. 「タイマ/カウンタ1」で各種の微小修正
6. タイマ/カウンタ0と2にPWMの対称性についての情報を追加
7. 「2線直列インターフェース(TWI)」で各種の微小修正
8. フラッシュページ容量を知らせるため、SPI直列プログラミングと自己プログラミング両方に189頁の表124.への参照を追加
9. 183頁の「ページ一時緩衝部の設定(ページ設定)」にSPMページ設定中のEEPROM書き込みについての注意を追加
10. ADHSMを完全に削除
11. 16頁に「パワーダウン休止動作中のEEPROM書き込み」項目を追加
12. 240頁の「外圍器情報」を更新

### 改訂2467I – 09/03

1. 28頁の「クロック分周制御レジスタ(XDIV)」の注意を更新
2. 31頁の「JTAGインターフェースと内蔵デバッグ機能(OCD)」を更新
3. 32頁の表19.でV<sub>BOT</sub>(BODLEVEL=1)に対する値を更新
4. JTAGENに関する164頁の「検査入出力ポート(TAP)」を更新
5. 169頁のJTDビットに対する説明を更新
6. 188頁の表118.にJTAGENヒューズに関する注意を追加
7. 206頁の「DC特性」でR<sub>PU</sub>値を更新
8. 244頁の「障害情報」でJTAG命令IDCODEに関する問題解決案を追加

### 改訂2467J – 12/03

1. 26頁の「校正付き内蔵RC発振器」を更新

### 改訂2467K – 03/04

1. 244頁の「障害情報」を更新

### 改訂2467L – 05/04

1. データシートから「暫定」と「TBD」を削除、ピン名ICnをICPnに改名
2. 25頁の表8.、32頁の表19.、36頁の表22.、159頁の表96.、196頁の表126.、198頁の表128.、207頁の表132.、209頁の表134.を更新
3. 18頁の「外部メモリインターフェース」を更新
4. 167頁の「デバイス識別レジスタ」を更新

5. 206頁の「電気的特性」を更新
6. 210頁の「A/D変換特性」を更新
7. 214頁からの「代表特性」を更新
8. 239頁の「注文情報」を更新

#### 改訂2467M – 11/04

1. 「アナログGND」を削除し、「GND」に置換
2. 26頁の表11、186頁の表114、198頁の表128、207頁の表132、158頁の図114を更新
3. 5頁の「ポートC (PC7~0)」に注:を追加
4. 239頁の「注文情報」を更新

#### 改訂2467N – 03/06

1. 「ピン配置」図に対する注:を更新
2. 49頁の「ポートDの交換機能」を更新
3. 55頁の「ポートGの交換機能」を更新
4. 67頁の「位相基準PWM動作」を更新
5. 88頁の表59と表60を更新
6. 96頁で「ビット2 – TOV3 : タイマ/カウンタ3割り込み要求フラグ」を更新
7. 111頁の「直列周辺インターフェース (SPI)」を更新
8. 154頁で「A/D変換器 (ADC)」内の特徴を更新
9. 160頁の「アナログ入力/利得選択」表内に注:を追加
10. 244頁の「障害情報」を更新

#### 改訂2467O – 10/06

1. 27頁の「タイマ/カウンタ用発振器」に注を追加
2. 84頁の「高速PWM動作」を更新
3. 69頁の表52、表54、88頁の表59、89頁の表61、107頁の表64、表66を更新
4. 244頁の「障害情報」を更新

#### 改訂2467P – 08/07

1. 1頁の「特徴」を更新
2. 6頁に「データ保持力」を追加
3. 88頁の表60と156頁の表95を更新
4. 120頁の「アセンブリ言語コード例」を更新
5. 158頁の図114を更新
6. 27頁の「クロック分周制御レジスタ (XDIV)」を更新
7. 244頁の「障害情報」を更新
8. 48頁の表34を更新
9. 113頁の「従装置動作」を更新

#### 改訂2467Q – 05/08

1. 16頁の「EEPROMデータ化けの防止」を更新  
文「内蔵BODの検出電圧が必要とした検出電圧と一致しない場合、外部低VCCリセット保護回路が使えます。」を削除
2. 131頁の「ポート設定例」で132頁の表82.(続き)を更新  
16MHzを越える周波数例を削除(訳注:本書では参考表記で残留)
3. 158頁の表114を更新  
インダクタ値を10mHから10μHに修正
4. 167頁の「版」の記述を更新
5. 206頁の「DC特性」からATmega128L削除
6. 207頁に「速度勾配」を追加
7. 239頁の「注文情報」を更新  
鉛板外囲器はもはや供給されず、これらの外囲器に関する注文情報を削除

8. 244頁の「障害情報」を更新  
「ATmega128改訂F～M」で改訂F～Mの障害内容を合併

#### 改訂2467R – 06/08

1. 表紙から“新規設計で推奨されません。”を削除

#### 改訂2467S – 07/09

1. 244頁の「障害情報」を更新
2. 新雛形(5.10版)で目次を更新
3. 表紙に“新規設計で推奨されません。”を追加
4. 206頁の「DC特性」で活動動作とアイドル動作の代表ICC値を追加

#### 改訂2467T – 07/10

1. 31頁の「消費電力の最小化」から24頁の「システム クロックとクロック選択」へのリンクを追加
2. 129頁の「USARTn制御/状態レジスタ (UCSRnB)」を更新
3. 208頁の表133「2線直列インターフェース必要条件」で式を修正
4. 208頁の表133「2線直列インターフェース必要条件」で下の注6と7を削除
5. データシートで技術用語の使い方を更新

#### 改訂2467U – 08/10

1. 追補A ATmega128/L 105°Cに関するコード情報で239頁の「注文情報」を更新

#### 改訂2467V – 02/11

1. 改訂Uで思いがけず変更された文書番号(2467)を更新
2. Atmelの新形式の指針に従って編集上の更新。数値とそれらの単位間にもはや空白はありません。
3. 改訂Uでの8ビットタイマ/カウンタ0、16ビットT/C1と3、PWM付き8ビットT/C2の章入れ替え再構成

#### 改訂2467W – 05/11

1. Atmel QTouchライブラリ支援とQTouch感知能力の特徴を追加
2. 206頁の「DC特性」を更新、RRST最大値が60kΩから85kΩに変更
3. テープとリールの製品を含むように239頁の「注文情報」を更新

#### 改訂2467X – 06/11

1. 239頁の「注文情報」で誤植を修正

## 障害情報

この章の改訂番号はATmega128デバイスの改訂版を参照してください。

### 改訂F～M

- 初回アナログ比較器変換が遅らされるかもしれない F～M
- 非同期タイマでタイマレジスタ書き込み時に割り込みが失われる可能性 F～M
- XDIVレジスタ変更時に安定時間が必要 F～M
- OSCCALレジスタ変更時に安定時間が必要 F～M
- IDCODEがTDI入力からのデータを遮断 F～M
- EEREビットを設定(1)するためのSTまたはSTS使用によるEEPROM読み込みが予期せぬ割り込み要求を起動する F～M

#### 1. 初回アナログ比較器変換が遅らされるかもしれない (F～M)

デバイスが低速上昇VCCで給電される場合、最初のアナログ比較器変換はいくつかのデバイスで予想よりも長くなるでしょう。

##### 対策/対処

デバイスが給電またはリセットされた時は最初の変換前にアナログ比較器を**禁止し、そして許可**してください。

#### 2. 非同期タイマ/カウンタでタイマ/カウンタレジスタ書き込み時に割り込みが失われる可能性 (F～M)

非同期タイマ/カウンタレジスタ(TCNTx)が\$00の時に同期タイマ/カウンタクロックのタイマ/カウンタレジスタが書かれる場合に割り込みが失われるでしょう。

##### 対策/対処

非同期タイマ/カウンタ制御レジスタ(TCCRx)、非同期タイマ/カウンタ(TCNTx)レジスタまたは非同期タイマ/カウンタ比較レジスタ(OCRx)を書く前に、常に非同期タイマ/カウンタレジスタが\$FFまたは\$00のどちらの値でもないことを調べてください。

#### 3. XDIVレジスタ変更時に安定時間が必要 (F～M)

XDIVレジスタでクロック元周波数を2%よりも多く上昇設定した後、デバイスは後続するいくつかの命令を正しく実行しないかもしれません。

##### 対策/対処

**NOP**命令は周波数変更後も常に正しく実行されます。従って周波数変更後の次の8命令は**NOP**命令であるべきです。これを保証するには次の手順に従ってください。

1. SREGの**全割り込み許可(I)ビット**を解除(0)します。(訳補:割り込み許可ならば)
2. XDIVレジスタに新しい分周値を設定します。
3. 8つの**NOP**命令を実行します。
4. SREGの**全割り込み許可(I)ビット**を設定(1)します。(訳補:割り込み許可ならば)

これは後続する全命令の正しい実行を保証します。

##### アセンブリ言語コード例

```

CLI          ; 全割り込み禁止
OUT         XDIV, R16 ; 新規分周値設定
NOP         ; 無操作
NOP         ; 無操作
NOP         ; 無操作
NOP         ; 無操作
NOP         ; 無操作
NOP         ; 無操作
NOP         ; 無操作
NOP         ; 無操作
NOP         ; 無操作
NOP         ; 無操作
SEI         ; 全割り込み許可

```

#### 4. OSCCALレジスタ変更時に安定時間が必要 (F～M)

OSCCALレジスタでクロック元周波数を2%よりも多く上昇設定した後、デバイスは後続するいくつかの命令を正しく実行しないかもしれません。

##### 対策/対処

動きは障害番号3.に従うので、同じ対策が利用できます。

## 5. IDCODEがTDI入力からのデータを遮蔽 (F~M)

IDCODEは正しく動作しません。DR更新の間中、後続するデバイスへのデータは全て1に置換されます。

### 問題の修正と対策

- ATmega128が走査チェーン内で唯一のデバイスなら、この問題は見られません。
- ATmega128のデバイスIDレジスタとおそらくは走査チェーンの後続するデバイスからのデータ内容を読むために、IDCODE命令を実行するか、またはTAP制御器の検査回路リセット状態へ移行することのどちらかにより、ATmega128のデバイスIDレジスタを選んでください。境界走査チェーンの先行デバイスのデバイスIDレジスタを読む間中、ATmega128へBYPASS命令を実行してください。
- 境界走査チェーン内の全デバイスのデバイスIDが同時に捕獲されなければならない場合、ATmega128はチェーンの先頭デバイスでなければなりません。

## 6. EEREビットを設定(1)するためのSTまたはSTS使用によるEEPROM読み込みが予期せぬ割り込み要求を起動する (F~M)

EEPROM制御レジスタ(EECR)のEEPROM読み込み許可(EERE)ビットを設定(1)するためのSTまたはSTS命令使用によるEEPROM読み込みが予期せぬEEPROM割り込み要求を起動します。

### 対策/対処

EECR内のEEREを設定(1)するのにOUTまたはSBIを常に使ってください。



## 目次

|                             |    |                            |     |
|-----------------------------|----|----------------------------|-----|
| 特徴                          | 1  | 8ビット タイマ/カウンタ0 (PWM,非同期動作) | 61  |
| ピン配置                        | 2  | 概要                         | 61  |
| 概要                          | 3  | タイマ/カウンタのクロック              | 62  |
| 構成図                         | 3  | 計数器部                       | 62  |
| ATmega128とATmega103の互換性     | 4  | 比較出力部                      | 63  |
| ピン説明                        | 5  | 比較一致出力部                    | 64  |
| 資料                          | 6  | 動作種別                       | 65  |
| データ保持力                      | 6  | タイマ/カウンタのタイミング             | 68  |
| コード例について                    | 6  | 8ビット タイマ/カウンタ0用レジスタ        | 69  |
| 容量性接触感知                     | 6  | タイマ/カウンタ0の非同期動作            | 71  |
| AVR CPU コア                  | 7  | タイマ/カウンタ0の前置分周器            | 73  |
| 序説                          | 7  | 16ビット タイマ/カウンタ1とタイマ/カウンタ3  | 74  |
| 構造概要                        | 7  | 概要                         | 74  |
| ALU (Arithmetic Logic Unit) | 7  | 16ビット レジスタのアクセス            | 76  |
| ステータス レジスタ                  | 8  | タイマ/カウンタのクロック              | 78  |
| 汎用レジスタ ファイル                 | 9  | 計数器部                       | 78  |
| スタック ポインタ                   | 9  | 捕獲入力部                      | 79  |
| 命令実行タイミング                   | 10 | 比較出力部                      | 80  |
| リセットと割り込みの扱い                | 11 | 比較一致出力部                    | 82  |
| AVR ATmega128のメモリ           | 12 | 動作種別                       | 83  |
| 実装書き換え可能なプログラム用フラッシュ メモリ    | 12 | タイマ/カウンタのタイミング             | 87  |
| データ用SRAMメモリ                 | 13 | 16ビット タイマ/カウンタ1,3用レジスタ     | 88  |
| データ用EEPROMメモリ               | 14 | タイマ/カウンタ1,2,3の前置分周器        | 97  |
| I/Oメモリ (レジスタ)               | 17 | 8ビット タイマ/カウンタ2 (PWM)       | 99  |
| 外部メモリ インターフェース              | 18 | 概要                         | 99  |
| システム クロックとクロック選択            | 24 | タイマ/カウンタのクロック              | 100 |
| クロック系統とその配給                 | 24 | 計数器部                       | 100 |
| クロック元                       | 24 | 比較出力部                      | 101 |
| 既定のクロック元                    | 24 | 比較一致出力部                    | 102 |
| クリスタル用発振器                   | 25 | 動作種別                       | 103 |
| 低周波数クリスタル用発振器               | 25 | タイマ/カウンタのタイミング             | 106 |
| 外部RC発振器                     | 26 | 8ビット タイマ/カウンタ2用レジスタ        | 107 |
| 校正付き内蔵RC発振器                 | 26 | 比較出力変調器 (COM1C2)           | 110 |
| 外部クロック信号                    | 27 | 概要                         | 110 |
| タイマ/カウンタ用発振器                | 27 | 説明                         | 110 |
| 電力管理と休止形態                   | 29 | SPI (直列周辺インターフェース)         | 111 |
| アイドル動作                      | 29 | SSピンの機能                    | 113 |
| A/D変換雑音低減動作                 | 29 | データ転送形式                    | 115 |
| パワーダウン動作                    | 30 | USART0とUSART1              | 116 |
| パワーセーブ動作                    | 30 | 概要                         | 116 |
| スタンバイ動作                     | 30 | クロック生成                     | 118 |
| 拡張スタンバイ動作                   | 30 | フレーム形式                     | 119 |
| 消費電力の最小化                    | 31 | USARTの初期化                  | 120 |
| システム制御とリセット                 | 32 | USARTのデータ送信                | 121 |
| 内部基準電圧                      | 34 | USARTのデータ受信                | 122 |
| ウォッチドッグ タイマ                 | 35 | 非同期受信                      | 125 |
| ウォッチドッグ タイマ設定変更の時間制限手順      | 36 | 複数プロセッサ通信動作                | 127 |
| 割り込み                        | 37 | USART0,1用レジスタ              | 128 |
| ATmega128の割り込みベクタ           | 37 | ポーレート設定例                   | 131 |
| 入出力ポート                      | 41 | 2線直列インターフェース (TWI)         | 133 |
| 序説                          | 41 | 特徴                         | 133 |
| 標準デジタル入出力としてのポート            | 41 | 2線直列インターフェース バスの定義         | 133 |
| 交換ポート機能                     | 44 | データ転送とフレーム形式               | 133 |
| I/Oポート用レジスタ                 | 56 | 複数主装置バス システムの調停と同期         | 135 |
| 外部割り込み                      | 59 | TWI部の概要                    | 136 |
|                             |    | TWI用レジスタ                   | 138 |
|                             |    | TWIの使用法                    | 140 |

|                                      |     |                                    |     |
|--------------------------------------|-----|------------------------------------|-----|
| 転送種別                                 | 142 | A/D変換器特性                           | 210 |
| 複数主装置システムでのバス競合と調停                   | 151 | データ用外部メモリ タイミング特性                  | 211 |
| <b>アナログ比較器</b>                       | 152 | <b>代表特性</b>                        | 214 |
| アナログ比較器入力選択                          | 153 | 活動動作消費電流                           | 214 |
| <b>A/D変換器</b>                        | 154 | アイドル動作消費電流                         | 216 |
| 特徴                                   | 154 | パワーダウン動作消費電流                       | 219 |
| 操作                                   | 155 | パワーセーフ動作消費電流                       | 219 |
| 変換の開始                                | 155 | スタンバイ動作消費電流                        | 220 |
| 前置分周と変換タイミング                         | 156 | ピンプルアップ                            | 220 |
| チャンネル変更と基準電圧選択                       | 157 | ピン駆動能力                             | 221 |
| 雑音低減機能                               | 158 | ピン 閾値とヒステリシス                       | 222 |
| A/D変換の結果                             | 159 | 低電圧検出器(BOD)閾値と<br>アナログ比較器変位(オフセット) | 223 |
| A/D変換用レジスタ                           | 160 | 内部発振器周波数                           | 224 |
| <b>JTAGインターフェースと内蔵デバッグ機能</b>         | 163 | 周辺機能部消費電流                          | 229 |
| 特徴                                   | 163 | リセット消費電流とリセットパルス幅                  | 230 |
| 概要                                   | 163 | <b>レジスタ要約</b>                      | 234 |
| 検査入出力ポート (TAP:Test Access Port)      | 164 | <b>命令要約</b>                        | 236 |
| TAP制御器                               | 164 | <b>注文情報</b>                        | 239 |
| 境界走査チェーン(Boundary-Scan Chain)<br>の使用 | 165 | <b>外圍器情報</b>                       | 240 |
| 内蔵デバッグ機能の使用                          | 165 | <b>データシート改訂履歴</b>                  | 241 |
| 内蔵デバッグ特殊JTAG命令                       | 165 | <b>障害情報</b>                        | 245 |
| 内蔵デバッグ関連I/Oメモリ内のレジスタ                 | 165 |                                    |     |
| JTAGプログラミング能力の使用                     | 166 |                                    |     |
| 参考文献                                 | 166 |                                    |     |
| <b>IEEE1149.1(JTAG)境界走査</b>          | 167 |                                    |     |
| 特徴                                   | 167 |                                    |     |
| 概要                                   | 167 |                                    |     |
| データレジスタ                              | 167 |                                    |     |
| 境界走査(Boundary-Scan)用JTAG命令           | 168 |                                    |     |
| 境界走査関連I/Oメモリ内のレジスタ                   | 169 |                                    |     |
| 境界走査チェーン(Boundary-Scan Chain)        | 170 |                                    |     |
| ATmega128の境界走査順                      | 176 |                                    |     |
| 境界走査記述言語(BSDL)ファイル                   | 177 |                                    |     |
| <b>ブートローダ支援 – RWW自己プログラミング</b>       | 178 |                                    |     |
| 特徴                                   | 178 |                                    |     |
| フラッシュメモリの応用領域とブートローダ領域               | 178 |                                    |     |
| 書き中に読みが可能な領域と不能な領域                   | 178 |                                    |     |
| ブートローダ施錠ビット                          | 180 |                                    |     |
| ブートローダプログラムへの移行                      | 180 |                                    |     |
| 自己プログラミングでのフラッシュアドレス指定               | 182 |                                    |     |
| フラッシュメモリの自己プログラミング                   | 182 |                                    |     |
| <b>メモリプログラミング</b>                    | 187 |                                    |     |
| プログラムメモリとデータメモリ用施錠ビット                | 187 |                                    |     |
| 識票バイト                                | 187 |                                    |     |
| 校正バイト                                | 187 |                                    |     |
| ヒューズビット                              | 188 |                                    |     |
| 並列プログラミング                            | 189 |                                    |     |
| 並列プログラミング手順                          | 190 |                                    |     |
| 直列プログラミング                            | 197 |                                    |     |
| 直列プログラミング手順                          | 197 |                                    |     |
| JTAGインターフェース経由プログラミング                | 199 |                                    |     |
| <b>電気的特性</b>                         | 206 |                                    |     |
| 絶対最大定格                               | 206 |                                    |     |
| DC特性                                 | 206 |                                    |     |
| 速度勾配                                 | 207 |                                    |     |
| 外部クロック特性                             | 207 |                                    |     |
| 2線直列インターフェース特性                       | 208 |                                    |     |
| SPI タイミング特性                          | 209 |                                    |     |



#### *Atmel Corporation*

2325 Orchard Parkway  
San Jose, CA 95131  
USA  
TEL (+1)(408) 441-0311  
FAX (+1)(408) 487-2600  
[www.atmel.com](http://www.atmel.com)

#### *Atmel Asia Limited*

Unit 01-5 & 16, 19F  
BEA Tower, Millennium City 5  
418 Kwun Tong Road  
Kwun Tong, Kowloon  
HONG KONG  
TEL (+852) 2245-6100  
FAX (+852) 2722-1369

#### *Atmel Munich GmbH*

Business Campus  
Parking 4  
D-85748 Garching b. Munich  
GERMANY  
TEL (+49) 89-31970-0  
FAX (+49) 89-3194621

#### *Atmel Japan*

141-0032 東京都品川区  
大崎1-6-4  
新大崎勸業ビル 16F  
アトメル ジャパン合同会社  
TEL (+81)(3)-6417-0300  
FAX (+81)(3)-6417-0370

#### © 2011 Atmel Corporation. 不許複製

Atmel<sup>®</sup>、ロゴとそれらの組み合わせ、AVR<sup>®</sup>、QTouch<sup>®</sup>、QMatrix<sup>®</sup>、AVR Studio<sup>®</sup>とその他はAtmel Corporationの登録商標または商標またはその付属物です。Windows<sup>®</sup>とその他は米国とその他の国に於けるMicrosoft Corporationの登録商標です。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のウェブサイトにある販売の条件と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。

#### © HERO 2022.

本データシートはAtmelのATmega128英語版データシート(改訂2467X-06/11)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、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。