

16/32Kバイト実装書き込み可能フラッシュメモリ、USB制御器付き
Atmel 8ビット マイクロコントローラ

データシート

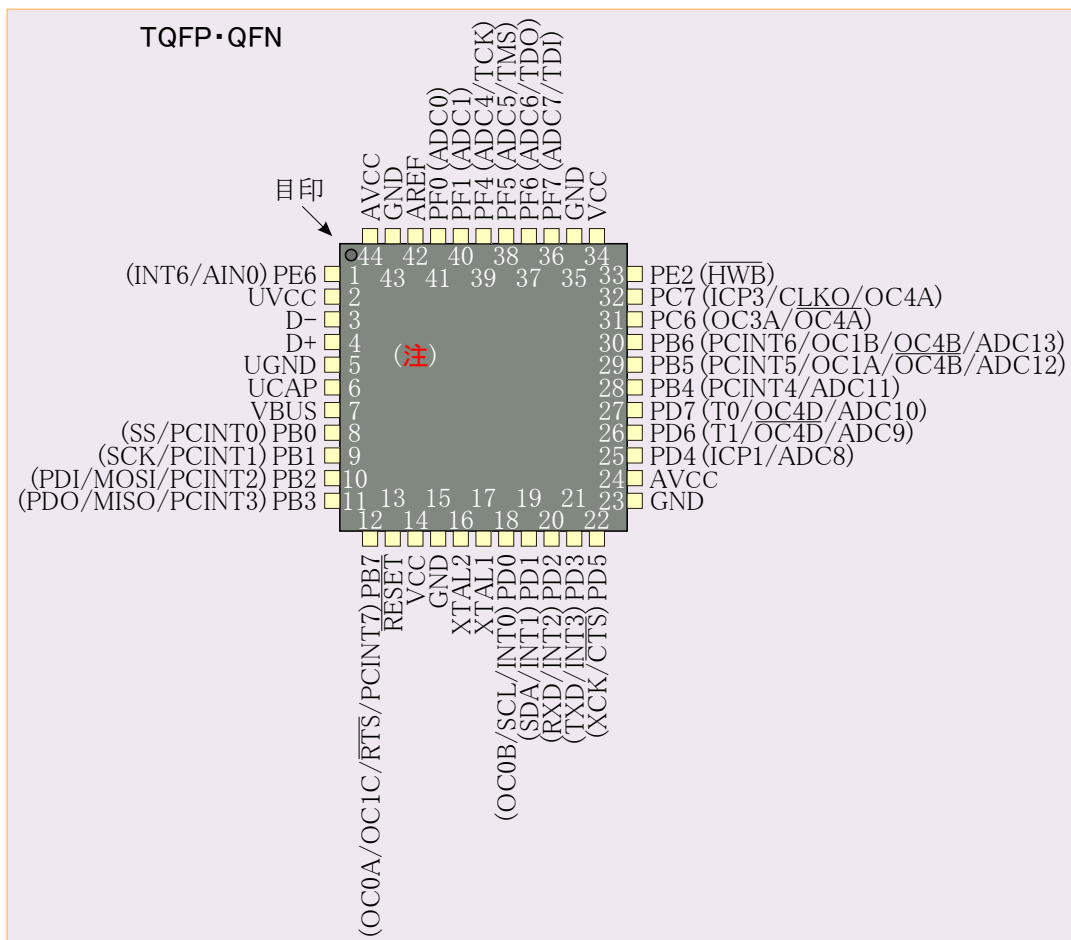
特徴

- 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131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
 - 完全なスタティック動作
 - 16MHz時、16MIPSに達する高速動作
 - 2周期実行の乗算命令
- データメモリと不揮発性プログラムメモリ
 - 実装自己書き換え可能な16K/32Kバイト(8K/16K語)フラッシュメモリ内蔵
 - 1.25K/2.5Kバイトの内蔵SRAM
 - 512/1KバイトのEEPROM
 - 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ポートコード領域
 - チップ内ポート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
 - 出荷時に外部XTALクロック使用デバイスは既定ポートローダが予め書き込み済み
 - ソフトウェア保護用の設定可能な施錠機能
- 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
- 転送完了での割り込み付きUSB全速(Full-speed)/低速(Low-speed)装置部
 - USB仕様改訂2.0(Universal Serial Bus Specification Rev 2.0)完全適合
 - 1.5Mbpsと12Mbpsまでのデータ転送速度を支援
 - 64バイトまでの制御転送用エンドポイント0
 - 大量(Bulk),割り込み(Interrupt),等時(Isochronous)転送と入出力方向が設定可能な6つのエンドポイント
 - 二重バンク動作で256バイトまで設定可能なエンドポイント容量
 - 完全に独立したエンドポイントメモリ割付用832バイトUSBデュアルポートRAM(DPRAM)
 - 休止(Suspend:サスペンド)/再開(Resume:リジューム)割り込み
 - USBバスリセットでCPUリセット可能
 - 全速バス動作48MHz PLL
 - マイクロコントローラ要求でのUSBバス接続/切断
 - 低速(Low-speed)動作に対するクリスタルなし動作
- 内蔵周辺機能
 - USBと高速タイマ用チップ上PLL: 32~96MHz動作
 - 独立した前置分周器、比較機能付き、1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16ビットタイマ/カウンタ
 - PLL(64MHz)と比較機能付き、1つの10ビット高速タイマ/カウンタ
 - 2つの8ビットPWM
 - 2~16ビット分解能が設定可能な4つのPWM
 - 2~11ビット分解能が設定可能な高速動作の6つのPWM
 - 比較出力変調器
 - 12チャネルの10ビットA/D変換器(設定可能な利得付き差動チャネル機能)
 - ハードウェア流れ制御付きの設定可能な直列USART
 - 主装置/従装置動作SPI直列インターフェース
 - バイト対応の2線直列インターフェース

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

- 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
- ピン変化での割り込みと起動復帰
- チップ上の温度感知器
- 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8MHz発振器
 - 内蔵クロック前置分周器とクロック急速切り換え(内蔵RC/外部発振器)
 - 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
- I/Oと外圍器
 - CMOS出力とLVTTL入力兼備の全I/O
 - 26ビットの設定可能なI/O
 - 44ピンTQFP(10×10mm)、44ピンQFN(7×7mm)
- 動作電圧
 - 2.7～5.5V
- 温度範囲
 - -40～85℃ (工業用)
- 最大速度
 - 8MHz/2.7V (全温度範囲)
 - 16MHz/4.5V (全温度範囲)

1. ピン配置

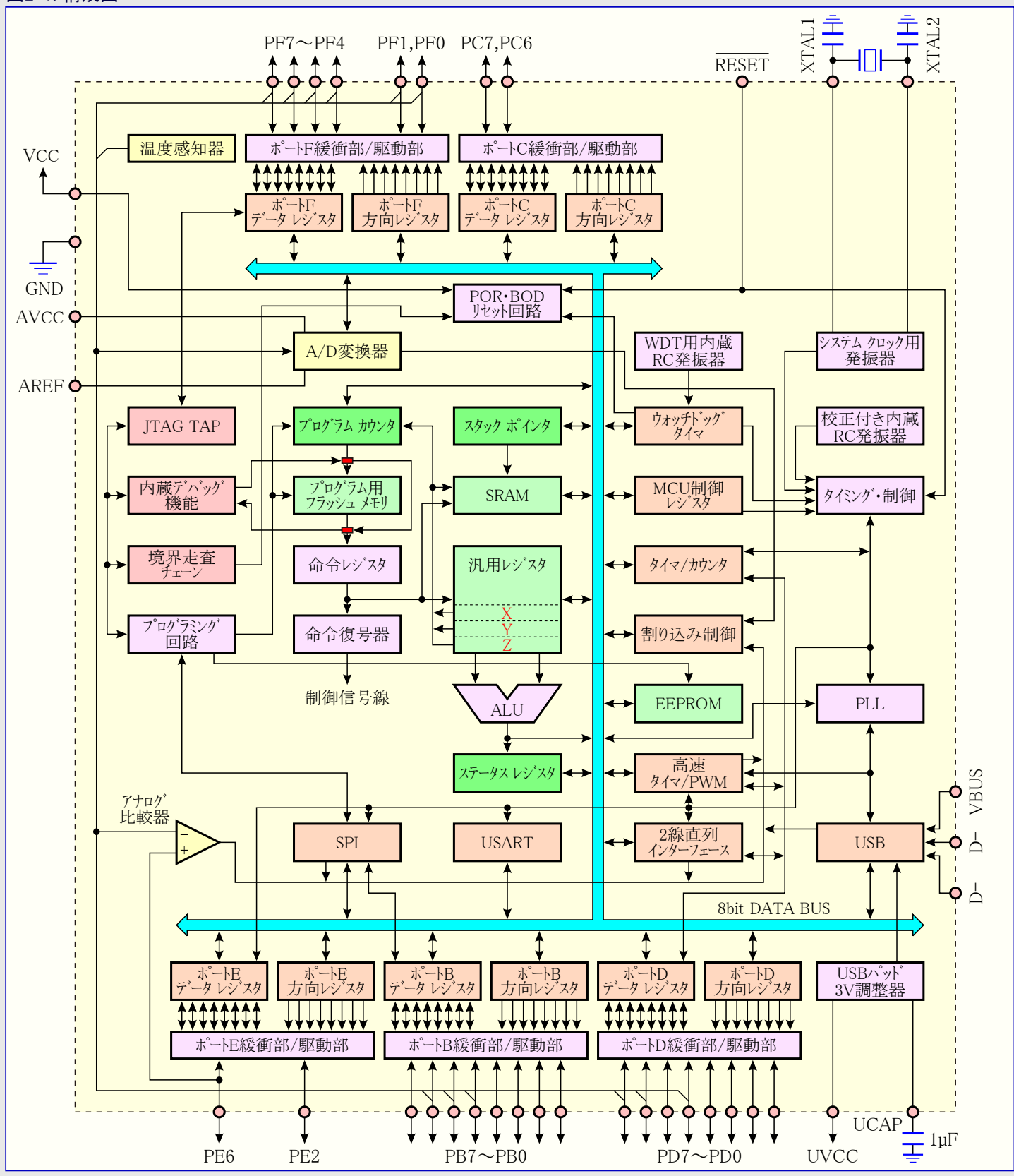


2. 概要

ATmega16U4/32U4はAVR強化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。

2.1. 構成図

図2-1. 構成図



AVRコアは32個の汎用作業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て10倍程度までの単位処理量向上効果があります。

このデバイスは次の特徴、書き込み中読める能力を持つ16K/32Kバイトの実装書き換え可能なフラッシュメモリ、512/1KバイトのEEPROM、1.25K/2.5KバイトのSRAM、26本の汎用入出力線(CMOS出力とLVTTTL入力)、32個の汎用作業レジスタ、比較動作やPWMを持つ柔軟な3つのタイマ/カウンタ、比較動作と調整可能なPLL供給元を持つ1つのより高速なタイマ/カウンタ、(CTS/RTS流れ制御信号を含む)1つのUSART、バイト対応の2線直列インターフェース、設定可能な増幅器付き任意選択差動段と校正された温度感知器付き12チャンネルの10ビットA/D変換器、設定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。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、A/D変換器を除く全ての周辺機能とCPUを停止します。スタンバイ動作ではクリスタル発振子/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。

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。モリシ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本デバイス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。

ATmega16U4/32U4 AVRはコンパイラ、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。

2.2. ピン概要

2.2.1. VCC

デジタル電源ピン。

2.2.2. GND

接地ピン。

2.2.3. 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は他のポートより優れた駆動能力を持ちます(訳注:該当記述が他にないため、消し忘れかもしれません)。

ポートBは45頁で一覧されるように本デバイスの様々な特殊機能

2.2.4. PC7,PC6 (ポートC)扱います。

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。

製品のピン出力にはビット7と6だけが存在します。

ポートCは47頁で一覧される本デバイスの様々な特殊機能も扱います。

2.2.5. 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48頁で一覧されるATmega16U4/32U4の様々な特殊機能も扱います。

2.2.6. PE6,PE2 (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。

製品のピン出力にはビット6と2だけが存在します。

ポートEは50頁で一覧されるATmega16U4/32U4の様々な特殊機能も扱います。

2.2.7. PF7~PF4, PF1,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)ならば、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。

製品のピン出力にはビット3と2が存在しません。

ポートFはJTAGインターフェース機能も扱います。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。

2.2.8. D-

USB全速(Full-speed)/低速(Low-speed)負極性データ上側ポート。22Ω直列抵抗でUSBコネクタD-ピンに接続されるべきです。

2.2.9. D+

USB全速(Full-speed)/低速(Low-speed)正極性データ上側ポート。22Ω直列抵抗でUSBコネクタD+ピンに接続されるべきです。

2.2.10. UGND

USBハット接地。

2.2.11. UVCC

USBハット用内蔵電圧調整器の供給電圧入力。

2.2.12. UCAP

USBハット用内蔵電圧調整器の供給電圧出力。外部コンデンサ(1μF)に接続されるべきです。

2.2.13. VBUS

USB VBUS監視入。

2.2.14. RESET

リセット入力。最小パルス幅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。最小パルス幅は249頁の表29-3. で与えられます。より短いパルスはリセットの生成が保証されません。

2.2.15. 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。

2.2.16. XTAL2

発振器反転増幅器からの出力。

2.2.17. 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。

2.2.18. AREF

AREFはA/D変換器用アナログ基準(電圧)ピンです。

3. 諸注意

3.1. お断り

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を基にしています。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。

3.2. 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。

3.3. コード例

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

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SBR, SBR, SBR, CBR命令と組み合わせたLDS, STS命令です。

3.4. 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℃または100年以上/25℃で1PPMよりずっと小さな値です。

4. AVR CPU コア

4.1. 序説

ここでは一般的な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。

4.2. 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ト構造を使います。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。

32個中の6つのレジスタは効率的なアドレス計算ができるデータ空間アドレス指定用に、3つの16ビット長間接アドレスポインタ用レジスタとして使われます。これらアドレスポインタの1つはプログラム用フラッシュメモリ内の定数表参照用アドレスポインタとしても使えます。これら16ビット長付加機能レジスタはX,Y,Zレジスタで、本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)、スタックポインタ(SP)を初期化しなければなりません。SPはI/O空間で読み書きアクセスが可能です。データ用SRAMはAVR構造で支援される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。

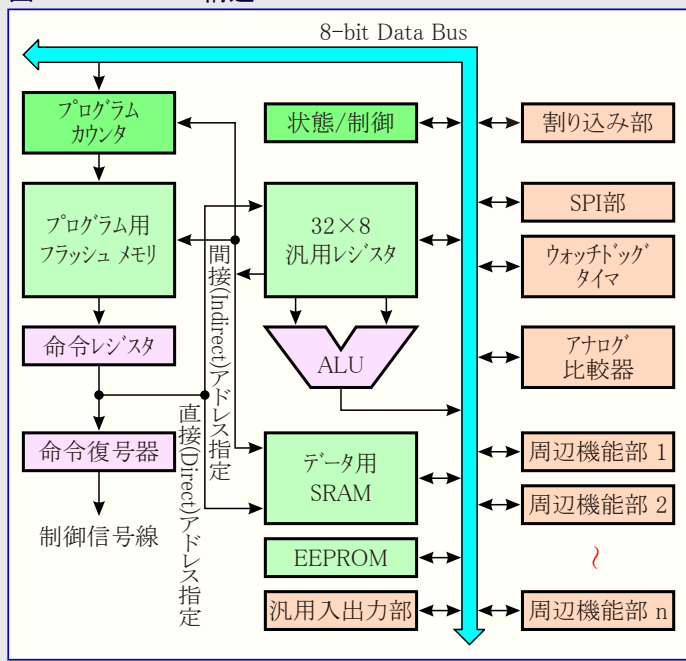
柔軟な割り込み部にはI/O空間の各制御レジスタとステータスレジスタ(SREG)の特別な全割り込み許可(I)ビットがあります。全ての割り込みは割り込みベクタ表に個別の割り込みベクタを持ちます。割り込みには割り込みベクタ表の位置に従う優先順があります。下位側割り込みベクタアドレスが高い優先順位です。

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ATmega16U4/32U4にはST/STS/STDとLD/LDS/LDD命令だけ使えるSRAM内の\$60～\$0FFに拡張I/O空間があります。

4.3. 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4-1. AVR MCU構造



4.4. 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。そのとき、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。

4.5. 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4-2.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4-2.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4-2. AVR CPU 汎用レジスタ構成図

7		0 アドレス	
R0		\$00	
R1		\$01	
R2		\$02	
⋮			
R13		\$0D	
R14		\$0E	
R15		\$0F	
R16		\$10	
R17		\$11	
⋮			
R26		\$1A	Xレジスタ
R27		\$1B	
R28		\$1C	Yレジスタ
R29		\$1D	
R30		\$1E	Zレジスタ
R31		\$1F	

4.5.1. 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4-3.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4-3. X,Y,Zレジスタ構成図

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

4.6. 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 ポインタの初期値は内蔵SRAMの最終アドレスです。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)	—	—	—	—	(SP11)	SP10	SP9	SP8	SPH
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
初期値	0	0	1	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	1	1	1	1	1	1	1	1	

(訳注) 上記の初期値は修正漏れで、RAMENDの場合はATmega16U4=\$05FF,ATmega32U4=\$0AFFと思われます。

(訳注) 原書本位置のRAMPZ記述は存在しないため、削除しました。

4.7. 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4-5.はハーバード構造と高速アクセスレジスタファイルの概念によって可能とされる並列の命令取得と命令実行を示します。これは機能対費用、機能対クロック、機能対電源部に関する好結果と対応するMHzあたり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4-6.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4-5. 命令の取得と実行の並列動作

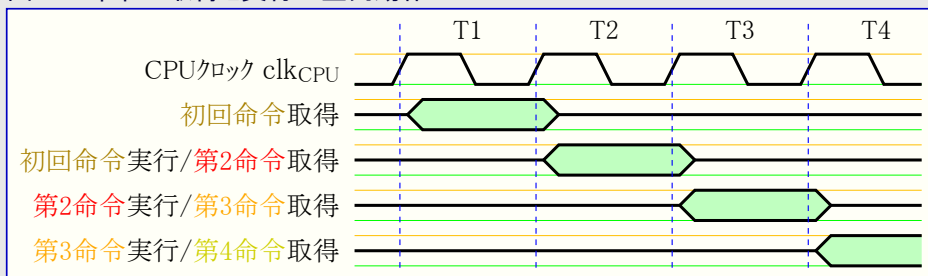
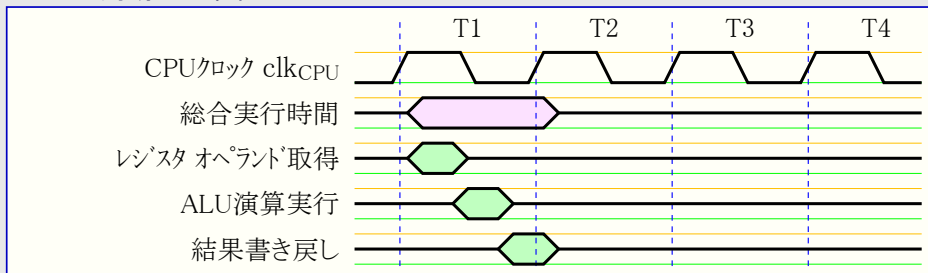


図4-6. 1周期ALU命令



4.8. 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 プ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226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36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36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。215頁の「ブート ロード支援 - 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。

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

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

2つ目の割り込み形式は割り込み条件が存在する限り起動し(続け)ます。これらの割り込みは必ずしも割り込み要求フラグを持っているとは限りません。割り込みが許可される前に割り込み条件が消滅すると、この割り込みは起動されません。

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

ステータスレジスタ(SREG)は割り込みルーチンへ移行時の保存も、復帰時の回復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こっても、**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, EEMPE        ;EEPROM主書き込み許可
SBI     EECR, EEPE         ;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<<EEMPE);        /* EEPROM主書き込み許可 */
EECR |= (1<<EEPE);         /* 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す。

アセンブリ言語プログラム例

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

4.8.1. 割り込み応答時間

許可した全てのAVR割り込みに対する割り込み実行応答は最小4クロック周期です。4クロック周期後、実際の割り込み処理ルーチンに対する**プログラム ベクタ アドレス**が実行されます。この4クロック周期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。

5. AVRのメモリ

この項は本デバイスの各種メモリを記述します。AVR構造にはプログラムメモリ空間とデータメモリ空間の2つの主なメモリ空間があります。加えて本デバイスはデータ保存用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。

表5-1. メモリ割付

メモリ種別		ニモニック	ATmega16U4	ATmega32U4
フラッシュメモリ	容量	Flash size	16Kバイト	32Kバイト
	開始アドレス	–	\$00000	
	終了アドレス	Flash end	\$3FFF(注1)/\$1FFF(注2)	\$7FFF(注1)/\$3FFF(注2)
汎用レジスタ	容量	–	32バイト	
	開始アドレス	–	\$0000	
	終了アドレス	–	\$001F	
(標準)I/Oレジスタ	容量	–	64バイト	
	開始アドレス	–	\$0020	
	終了アドレス	–	\$005F	
拡張I/Oレジスタ	容量	–	160バイト	
	開始アドレス	–	\$0060	
	終了アドレス	–	\$00FF	
内蔵SRAM	容量	ISRAM size	1.25Kバイト	2.5Kバイト
	開始アドレス	ISRAM start	\$0100	
	終了アドレス	ISRAM end	\$05FF	\$0AFF
外部メモリ	容量	XMem size	(存在しません。)	
	開始アドレス	XMem start		
	終了アドレス	XMem end		
内蔵EEPROM	容量	E2 size	512バイト	1Kバイト
	開始アドレス	–	\$0000	
	終了アドレス	E2 end	\$01FF	\$03FF

注1: バイトアドレスです。

注2: 語(ワード=16ビット)アドレスです。

5.1. 実装書き換え(ISP: In-System Program)可能なプログラム用フラッシュ メモリ

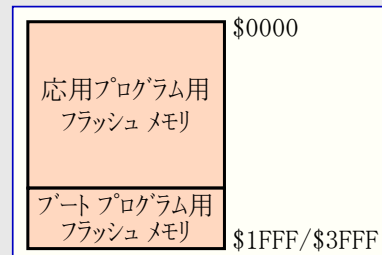
本デバイスはプログラム保存用に実装書き換え可能な16/32Kバイトのフラッシュ メモリをチップ上に含みます。全てのAVR命令が16または32ビット幅のため、このフラッシュ メモリは8/16K×16ビットとして構成されます。ソフトウェア保護のため、フラッシュ プログラム メモリ空間はブート プログラム領域と応用プログラム領域の2つに分けられます。

フラッシュ メモリは最低100,000回の消去/書き込み回数耐久性があります。本デバイスのプログラム カウンタ(PC)は13/14ビット幅で、故に8/16K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215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。226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命令記述参照)

命令の取得と実行のタイミング図は9頁の「[命令実行タイミング](#)」で示されます。

図5-1. プログラム メモリ配置図



5.2. データ用SRAM メモリ

図5-2は本デバイスのSRAMメモリ構成方法を示します。

本デバイスは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～\$FFF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。

下位1536/2816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1280/2560位置はデータ用内蔵SRAMに充てます。

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タ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バイト位置に届きます。

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。

本デバイスの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、1.25K(1280)/2.5K(2560)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8頁の「[汎用レジスタ ファイル](#)」で記述されます。

図5-2. データ空間とSRAMの配置

		アドレス
レジスタ ファイル (32×8)	R0	\$0000
	～ R31	～ \$001F
I/O レジスタ (64×8)	\$00	\$0020
	～ \$3F	～ \$005F
拡張I/O レジスタ (160×8)	\$0060	\$0060
	～ \$00FF	～ \$00FF
内蔵 SRAM (1280/2560×8)	\$0100	\$0100
	～ \$05FF/\$0AFF	～ \$05FF/\$0AFF

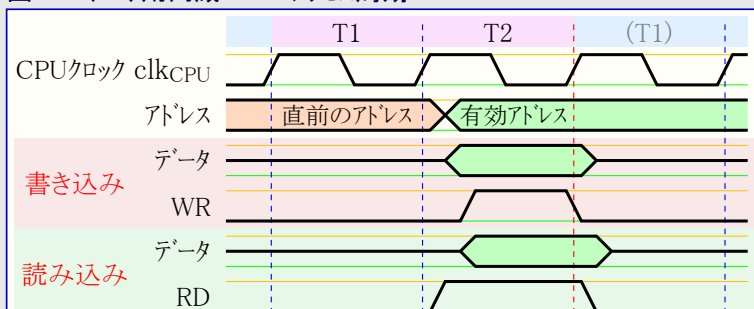
注: 赤字はI/Oアドレス

5.2.1. 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5-3で記載されるように2 clk_{CPU}周期で実行されます。

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

図5-3. データ用内蔵SRAMアクセス周期



5.3. データ用EEPROMメモリ

本デバイスは512/1Kバイトのデータ用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。

EEPROMのSPI、JTAG、並列プログラミングの詳細記述については各々 [238頁](#)、[241頁](#)、[229頁](#)をご覧ください。

5.3.1. 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[14頁の表5-2](#)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時間、使われるクロ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[15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。

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

ビット	15	14	13	12	11	10	9	8	
\$22 (\$42)	–	–	–	–	–	–	(EEAR9)	EEAR8	EEARH
Read/Write	R	R	R	R	R	R	R/W	R/W	
初期値	0	0	0	0	0	0	不定	不定	
ビット	7	6	5	4	3	2	1	0	
\$21 (\$41)	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15~10 – 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9~0 – EEAR9~0 : EEPROMアドレス (EEPROM Address)

EEPROMアドレスレジスタ(EEARHとEEARL)は512/1K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0~511/1023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なければなりません。

5.3.3. EEDR – EEPROMデータレジスタ (EEPROM Data Register)

ビット	7	6	5	4	3	2	1	0	
\$20 (\$40)	(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。

5.3.4. EECR – EEPROM制御レジスタ (EEPROM Control Register)

ビット	7	6	5	4	3	2	1	0	
\$1F (\$3F)	–	–	EEPM1	EEPM0	EERIE	EEMPE	EEPE	EERE	EECR
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	不定	不定	0	0	不定	0	

● ビット7,6 – 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5,4 – EEPM1,0 : EEPROMプログラミング種別 (EEPROM Programming Mode Bits)

EEPROMプログラミング種別ビット設定はEEPROMプログラミング許可(EEPE)書き込み時にどのプログラミング動作が起動されるかを定義します。1つの非分離操作(旧値消去と新値書き込み)、または2つの異なる操作として消去と書き込み操作を分離してデータをプログラムする(書く)ことが可能です。各動作に対するプログラミング時間は表5-2で示されます。EEPEが設定(1)されている間はEEPMMnへのどの書き込みも無視されます。リセット中、EEPMMnビットはEEPROMがプログラミング作業中を除いて'00'にリセットされます。

表5-2. EEPROMプログラミング種別			
EEPMM1	EEPMM0	プログラミング時間	動作
0	0	3.4ms	1操作での消去と書き込み(非分離操作)
0	1	1.8ms	消去のみ
1	0	1.8ms	書き込みのみ
1	1	–	将来使用に予約

●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EEPROMプログラミング許可(EEPE)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 – EEMPE : EEPROM主プログラム許可 (EEPROM Master Program Enable)

EEMPEビットはEEPROMプログラミング許可(EEPE)ビットの1書き込みが有効か無効かどうかを決めます。EEMPEが設定(1)されると、4クロック周期内のEEPE設定(1)は選んだアドレスのEEPROMをプログラムします。EEMPEが0なら、EEPE設定(1)は無効です。EEMPE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次のEEPE記述をご覧ください。

●ビット1 – EEPE : EEPROMプログラム許可 (EEPROM Program Enable)

EEPROMプログラム許可信号(EEPE)はEEPROMへのプログラミング許可信号です。EEPEが(1)を書かれると、EEPROMはEEPMMnビット設定に従ってプログラムされます。論理1がEEPEへ書かれる前にEEPROM主プログラム許可(EEMPE)ビットは1を書かれなければならない、さもなければEEPROM書き込み(消去)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プログラム許可(EEPE)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主プログラム許可(EEMPE)ビットに1を、EEPROMプログラム許可(EEPE)ビットに0を同時に書きます。
6. EEMPEビット設定後4クロック周期内にEEPROMプログラム許可(EEPE)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215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主プログラム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(プログラミング)アクセス時間が経過されると、EEPROMプログラム許可(EEPE)ビットはハードウェアによって解除(0)されます。使用者ソフトウェアは次バイ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。EEPE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への読み込みストローブです。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。

使用者は読み込み操作を始める前にEEPEビットをポーリングすべきです。書き込み(プログラム)操作実行中の場合、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発振器が使われます。右表は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5-3. EEPROM書き込み時間		
項目	校正付き内蔵RC発振器周期数	代表
EEPROM書き込み(CPU)	26,368	3.3ms

次のコード例はアセンブリ言語とC言語でのEEPROM消去、書き込み、または非分離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。(訳注:共通性から次例は補足修正しています。)

アセンブリ言語プログラム例

```
EEPROM_WR:  SBIC    EECR, EEPE                ;EEPROMプログラミング完了ならばスキップ
            RJMP    EEPROM_WR                ;以前のEEPROMプログラミング完了まで待機
;
            LDI     R19, (0<<EEPROM1) | (0<<EEPROM0) ;プログラミング種別値取得(本例は非分離)
            OUT     EECR, R19                ;対応プログラミング種別設定
            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, EEMPE              ;EEPROM主プログラム許可ビット設定
            SBI     EECR, EEPE              ;EEPROMプログラミング開始(プログラム許可ビット設定)
            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<<EEPE)));                /* 以前のEEPROMプログラミング完了まで待機 */
    EECR = (0<<EEPROM1) | (0<<EEPROM0);        /* 対応プログラミング種別設定 */
    EEAR = uiAddress;                          /* EEPROMアドレス設定 */
    EEDR = ucData;                            /* EEPROM書き込み値を設定 */
    EECR |= (1<<EEMPE);                       /* EEPROM主プログラム許可 */
    EECR |= (1<<EEPE);                       /* EEPROMプログラミング開始 */
}
```

注: 5頁の「コード例」をご覧ください。

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, EEPE                ;EEPROMプログラミング完了ならばスキップ
            RJMP    EEPROM_RD                ;以前のEEPROMプログラミング完了まで待機
;
            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<<EEPE)));                /* 以前のEEPROMプログラミング完了まで待機 */
    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 */
}
```

注: 5頁の「コード例」をご覧ください。

5.3.5. 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自身が命令を間違えて実行し得ます。

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。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。書き込み動作実行中にリセットが起ると、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。

5.4. I/O メモリ (レジスタ)

本デバイスのI/O空間定義は269頁の「レジスタ要約」で示されます。

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時はこれらのアドレスに\$20が加算されなければなりません。

本デバイスは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～\$0FF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。

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命令は他の多くのAVRの様ではなく、指定ビットだけを操作し、従って状態フラグのようなものを含むレジスタに使えることに注意してください。CBIとSBI命令は(I/Oアドレス)\$00～\$1Fのレジスタでのみ動作します。

I/Oと周辺制御レジスタは以降の項で説明されます。

5.4.1. 汎用I/Oレジスタ

本デバイスは3つの汎用I/Oレジスタを含みます。これらのレジスタはどの情報の格納にも使え、特に全体変数や状態フラグの格納に有用です。(I/O)アドレス範囲\$00～\$1Fの汎用I/OレジスタはSBI, CBI, SBIS, SBIC命令の使用で直接ビットアクセスが可能です。

5.4.2. GPIOR2 – 汎用I/Oレジスタ2 (General Purpose I/O Register 2)

ビット	7	6	5	4	3	2	1	0	
\$2B (\$4B)	(MSB)							(LSB)	GPIOR2
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	

5.4.3. GPIOR1 – 汎用I/Oレジスタ1 (General Purpose I/O Register 1)

ビット	7	6	5	4	3	2	1	0	
\$2A (\$4A)	(MSB)							(LSB)	GPIOR1
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	

5.4.4. GPIOR0 – 汎用I/Oレジスタ0 (General Purpose I/O Register 0)

ビット	7	6	5	4	3	2	1	0	
\$1E (\$3E)	(MSB)							(LSB)	GPIOR0
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	

6.2. 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。

表6-1. クロック元選択

クロック元	CKSEL3~0またはEXCKSEL3~0
外部クリスタル低電力発振器	1111~1000
(予約)	0111~0110
外部低周波数クリスタル発振器	0101~0100
(予約)	0011
校正付き内蔵RC発振器	0010
外部クロック信号	0000
(予約)	0001

注: 1=非プログラム、0=プログラム

6.2.1. 既定のクロック元 (ATmega16U4とATmega32U4)

このデバイスはクリスタル用低電力発振器(8MHz~16MHz)が許可で、CKDIV8ヒューズがプログラム(0)され、結果として8MHzクリスタルで1.0MHzのシステムクロックで出荷されます。既定クロック選択ヒューズ設定の概要については228頁の表28-5をご覧ください。

6.2.2. 既定のクロック元 (ATmega16U4RCとATmega32U4RC)

このデバイスは校正付き内蔵RC発振器(8MHz)が許可で、CKDIV8ヒューズがプログラム(0)され、結果として1.0MHzのシステムクロックで出荷されます。既定クロック選択ヒューズ設定の概要については227頁の表28-5をご覧ください。

6.2.3. クロック起動手順

何れのクロック元も発振を開始するための十分なVCCと、それが安定であると考えられるのに先立つ、最低発振周期数が必要です。

十分なVCCを保証するため、その他全てのリセット元によってデバイスリセットが開放された後、デバイスは起動遅延時間(t_{OUT})の内部リセットを発生します。30頁の「システム制御とリセット」はこの内部リセットに対する起動条件を記述します。この遅延(t_{OUT})はウォッチドッグ発振器で計時され、遅延周期数はSUTとCKSELのヒューズによって設定されます。選択可能な遅延右表で示されます。この表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。

遅延の主な目的はAVRが最小VCCを印加されるまでAVRをリセットに保つことです。この遅延は実電圧を監視しませんので、VCC上昇時間より長い遅延を選ぶことが必要とされるべきです。これが不可能な場合、内部または外部の低電圧検出回路(BOD)が使われるべきです。BOD回路がリセットを開放する前に十分なVCCを保証するでしょうから、起動遅延時間は禁止され得ます。低電圧検出回路(BOD)なしでの起動遅延時間の禁止は推奨されません。

この発振器はクロックが安定と考えられるのに先立って最低クロック数を発振することを必要とされます。内部リプルカウンタは発振器の出力クロックを監視し、与えられたクロック周期数に対して内部リセットを活性(有効)に保ちます。このリセットはその後に開放され、デバイスが実行を開始します。推奨発振器起動時間はクロック種別に依存し、外部的に印加されたクロックに対する6周期から、低周波数クリスタルに対する32K周期まで変化します。

クロックについての起動手順は計時完了遅延とデバイスがリセットから起動する時の起動時間の両方を含みます。パワーダウんまたはパワーセーブから起動するとき、VCCは十分な電圧であると認識され、起動時間だけが含められます。

表6-2. WDT発振器の代表計時完了値、周期数

VCC=3.0V	VCC=5.0V	周期数
0ms	0ms	0
4.3ms	4.1ms	512
69ms	65ms	8K (8192)

6.3. クリスタル用低電力発振器

XTAL1とXTAL2は図6-2.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。クリスタル発振子またはセラミック振動子のどちらでも使えます。

この発振器はXTAL2出力上の電圧振幅を減少した低電力発振器です。これは最低消費電力を与えますが、他のクロック入力を駆動する能力はありません。

C1とC2はクリスタル発振子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子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。クリスタル発振子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下表で与えられます。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。

低電力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。この動作種別はこの表で示されるようにCKSEL3~1ヒューズによって選ばれます。

図6-2. クリスタル発振子接続図

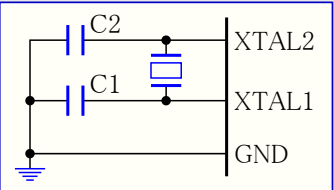


表6-3. クリスタル用低電力発振器動作種別

CKSEL3~1	周波数範囲	推奨C1,2容量
100 (注1)	0.4~0.9MHz	-
101	0.9~3.0MHz	12~22pF
110	3.0~8.0MHz	12~22pF
111	8.0~16MHz	12~22pF

注: 周波数が仕様(VCC依存)超の場合、CKDIV8=0が可能です。分周後クロックが仕様内であることを保証せねばなりません。

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

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

表6-4. クリスタル発振子/セラミック振動子用低電力発振器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)	14×CK+4.1ms	外部セラミック振動子、高速上昇電源
	0 1	258×CK (注1)	14×CK+65ms	外部セラミック振動子、低速上昇電源
	1 0	1K×CK (注2)	14×CK	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)	14×CK+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)	14×CK+65ms	外部セラミック振動子、低速上昇電源
	0 1	16K×CK	14×CK	外部クリスタル発振子、低電圧検出(BOD)リセット許可
	1 0	16K×CK	14×CK+4.1ms	外部クリスタル発振子、高速上昇電源
	1 1	16K×CK	14×CK+65ms	外部クリスタル発振子、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子用ではありません。

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子も使えます。

6.4. 低周波数クリスタル用発振器

このデバイスは専用の低周波数発振器によってクロック元として時計用32.768kHzクリスタルを使えます。クリスタル発振子は図6-2.で示されるように接続されるべきです。この発振器が選ばれると、起動時間は下表で示されるようにSUTヒューズとCKSEL0ヒューズによって決定されます。

表6-5. 低周波数クリスタル用発振器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	1K×CK (注1)	14×CK	低電圧検出(BOD)リセット許可
	0 1	1K×CK (注1)	14×CK+4.1ms	高速上昇電源
	1 0	1K×CK (注1)	14×CK+65ms	低速上昇電源
	1 1	(予約)		
1	0 0	32K×CK	14×CK	低電圧検出(BOD)リセット許可
	0 1	32K×CK	14×CK+4.1ms	高速上昇電源
	1 0	32K×CK	14×CK+65ms	低速上昇電源
	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。

6.5. 校正付き内蔵RC発振器

既定による校正された内蔵RC発振器は8.0MHzのクロックを供給します。この周波数は3V/25℃での公称値です。デバイスはCKDIV8ヒューズがプログラム(0)で出荷されます。より多くの詳細については22頁の「システム クロック前置分周器」をご覧ください。このクロックは下表で示されるように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。選ばれたなら、外部部品なしで動作します。リセット中、ハードウェアが発振校正(OSCCAL)レジスタに予めプログラムされた校正値バイトを設定し、これによってRC発振器を自動的に校正します。3V/25℃で、この校正は8MHz±1%の周波数を与えます。この発振器は発振校正(OSCCAL)レジスタの変更により、1%の精度内で7.3～8.1MHz範囲のどの周波数にも校正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時に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値のより多くの情報については228頁の「校正バイト」項をご覧ください。

表6-6. 校正付き内蔵RC発振器動作種別

CKSEL3~0	周波数範囲 (MHz)	注: この8MHz周波数がデバイス仕様(VCCに依存)を越える場合、内部周波数を8分周するためにCKDIV8ヒューズをプログラム(0)にできます。
0 0 1 0	7.3~8.1	

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

表6-7. 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	14×CK	低電圧検出リセット(BOD)許可
0 1	6×CK	14×CK+4.1ms	高速上昇電源
1 0	6×CK	14×CK+65ms	低速上昇電源
1 1	(予約)		

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

ビット (\$66)	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								

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

発振校正レジスタは発振器周波数の偏差処理を省くための内蔵発振器の調整に使われます。チップのリセット中に、25℃で8MHzの発振器周波数を与える工場校正値がこのレジスタへ自動的に書かれます。応用ソフトウェアは発振器周波数を変更するために、このレジスタに書くことができます。校正範囲は±40%で直線(校正段階:~0.4%)です。25℃での代表的な製造工程で、符号(値)は8MHzに対して127であるべきで、入力値\$00は最低、\$FFは最高の周波数を与えます。

温度感度は殆ど直線ですが、前述のように製造工程に依存します。この傾斜を決めるため、周波数は2つの温度で測定されなければなりません。本デバイスの温度感知器が上記の操作を可能にし、これは199頁の「温度感知器校正」で詳述されます。この後に測定した温度の関数で発振器周波数を校正することが可能です。

この発振器はフラッシュ メモリとEEPROMの書き込みアクセス時間に使われ、これらの書き込み時間はそれに応じて影響されることに注意してください。フラッシュ メモリまたはEEPROMが書かれる場合、8.8MHzより高く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の書き込みは失敗するかもしれません。

6.5.2. RCCTRL – 発振器制御レジスタ (Oscillator Control Register)

ビット (\$67)	7	6	5	4	3	2	1	0	RCCTRL
	–	–	–	–	–	–	–	RCFREQ	
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)

これらのビットを設定(1)してはいけません。これらのビットは常に0として読みます。

● ビット0 – RCFREQ : RC発振器周波数選択 (RC Oscillator Frequency Select)

このビットが解除(0:既定値)されると、RC発振器出力周波数は8MHzに設定されます。このビットが設定(1)されると、RC発振器出力周波数は1MHzです。発振校正(OSCCAL)レジスタが8MHzと1MHzの両方の出力動作で同じ効果(~0.4%/段階)を持つことに注意してください。

6.6. 外部クロック信号

外部クロック元からデバイスを駆動するには、XTAL1が図6-3.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は、表6-1.で示されるようにCKSELヒューズが'0000'にプログラム(設定)されなければなりません。

このクロック元が選ばれると、起動時間は下表で示されるようにSUTヒューズによって決定されず。

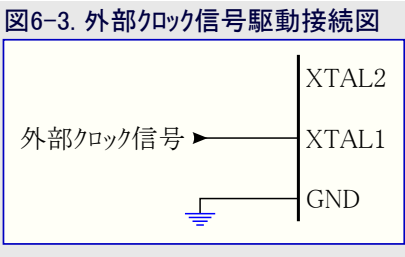


表6-8. 外部クロック信号駆動用起動遅延時間選択表

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	14×CK	低電圧検出(BOD)リセット許可
0 1	6×CK	14×CK+4.1ms	高速上昇電源
1 0	6×CK	14×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。

システムクロック前置分周器が安定な動作を保証しながら、内部クロック周波数の実行時変更の実現に使えることに注意してください。詳細については22頁の「システムクロック前置分周器」を参照してください。

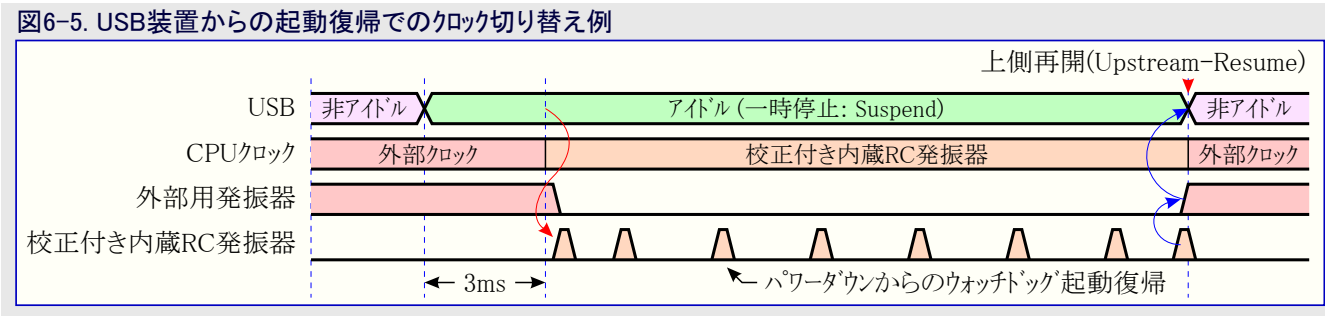
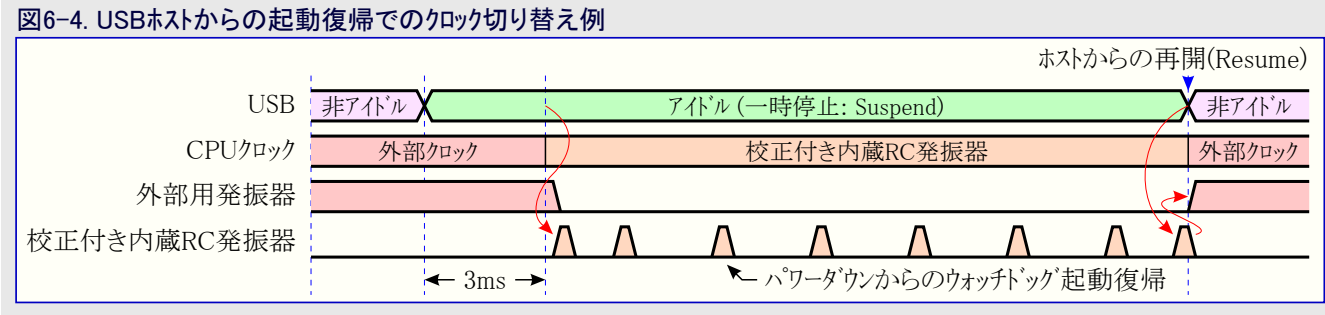
6.7. クロック切り替え

本デバイスはより正確に応用電力と実行時間を制御するために、ソフトウェアによって或るクロック元から別の1つに切り換えることを使用者に許す、クロック切り換え制御器を含みます。

6.7.1. 使用例

この変更はデバイスがUSB休止(サスペンド)動作へ移行する時に必要とされるかもしれません。そして(電力)消費と起動遅延を減らすために外部クロックから校正付き内蔵RC発振器へ切り替えます。このような構成設定では外部クロックが禁止されます。その後ファームウェアは応用での事象があるかを検査するために、パワーダウンから起動復帰するのにウォッチドッグタイマを使えます。応用での事象発生またはUSB信号線での非アイドル状態をUSB制御器が示す(例えば再開(Resume))場合、ファームウェアはUSB操作を再開するために、クロック多重器を校正付き内蔵RC発振器から外部クロックへ切り替えます。

この機能は校正付き8MHz内蔵RC発振器、外部クロック信号、クリスタル用低電力発振器間の切り換えだけに使えます。低周波数クリスタル用発振器はこの機能で使われてはなりません。



6.8. クロック出力緩衝部 (外部クロック出力)

このデバイスはシステム クロックをCLKOピンに出力できます。この出力を許可するには、CKOUTヒューズがプログラム(0)されなければなりません。この動作はチップのクロックがシステム上の他の回路を駆動する時用です。このヒューズがプログラム(0)されると、I/Oピンの標準動作は無視され、このクロックはリセット中も出力されます。CLKOがクロック出力を扱うとき、校正付き内蔵RC発振器を含む何れのクロック元も選べます。システム クロック前置分周器が使われる場合、出力は分周したシステム クロックです。

6.9. システム クロック前置分周器

このデバイスはシステム クロック前置分周器を持ち、システム クロックはクロック前置分周レジスタ(CLKPR)の設定によって分周できます。この特徴(機能)は必要とされる処理能力が低い時の消費電力削減に使えます。これは全クロック種別で使え、CPUと全同期周辺機能のクロック周波数に影響を及ぼします。clkCPU, clkFLASH, clkI/O, clkADCは表6-9.で示された値によって分周されます。

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

前置分周器として実行するリップル カウンタは分周されないクロック周波数で走行し、CPUのクロック周波数より速いかもしれません。従って例え(カウンタ値が)読めるとしても、前置分周器の状態を決めることはできず、1から他へのクロック分周値切り替えを行う正確な時間は必ずしも予測できません。CLKPS値が書かれる時から新規クロック周波数が活性(有効)になる前にT1+T2~T1+2×T2間かかります。この間で2つの有効なクロック端が生成されます。ここでのT1は直前のクロック周期、T2は新規前置分周器設定に対応する周期です。

予期せぬクロック周波数の変更を防ぐため、CLKPSビットの変更は次の特別な書き込み手順に従わなければなりません。

1. クロック分周値変更許可(CLKPCE)ビットに1、CLKPR内の他の全ビットに0を書きます。
2. (次からの)4周期以内にCLKPCEビットを0とする欲したCLKPS値を書きます。

前置分周器設定変更時、書き込み手続きが割り込まれないことを保証するため、割り込みは禁止されなければなりません。

6.10. PLL

このPLLはUSBインターフェースと高速タイマ単位部用の(96MHzまでの)内部高周波数クロックを生成するのに使われ、PLL入力は外部低周波数クロック(XTAL1ピンからの外部クロック入力ピンまたはクリスタル発振器)から供給されます。

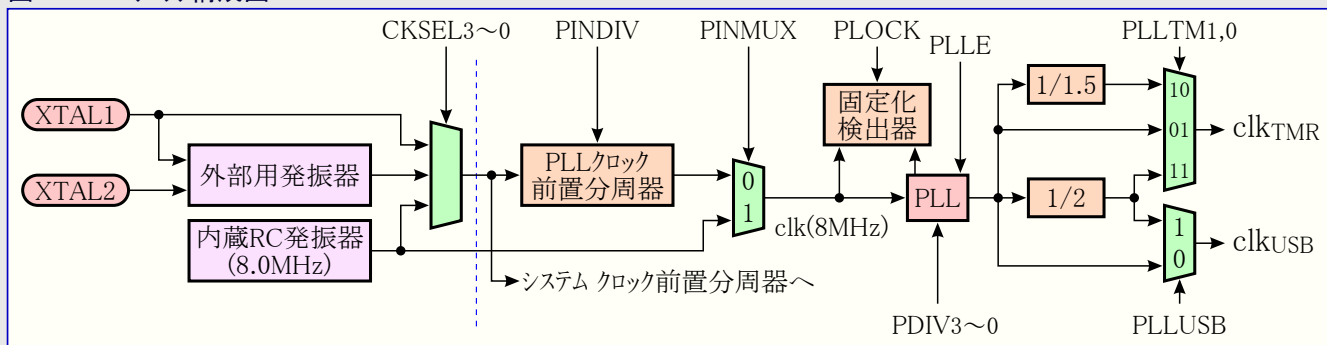
6.10.1. 内部PLL

このデバイスの内部PLLは公称8MHz入力から40MHzと96MHz間のクロック周波数を生成します。

8MHzのPLL入力クロック元は、クロック元多重器出力から8MHzを生成する内部PLLクロック前置分周器の出力です(PLLインターフェースについては「PLLCSR - PLL制御/状態レジスタ」をご覧ください)。

PLL出力信号はUSBと高速タイマ単位部へ配給される前にPLL後置分周器段に入ります。これらの単位部の各々は独立した分周比を選ぶことができます。

図6-6. PLLクロック構成図



6.11. クロック切り替え方法

6.11.1. 外部クロックから校正付き内蔵RC発振器への切り替え

```
if (Usb_suspend_detected())           // if (UDINT.SUSPI==1)
(
    Usb_ack_suspend();                 // UDINT.SUPI=0;
    Usb_freeze_clock();                // USBCON.FRZCLK=1;
    Disable_pll();                     // PLLCSR.PLLE=0;
    Enable_RC_clock();                 // CLKSELO.RCE=1;
    while (!RC_clock_ready());         // while (CLKSTA.RCON!=1);
    Select_RC_clock();                 // CLKSELO.CLKS=0;
    Disable_external_clock();          // CLKSELO.EXT=0;
)
```


6.11.2. 校正付き内蔵RC発振器から外部クロックへの切り替え

```
if (Usb_wake_up_detected())           // if (UDINT.WAKEUPI==1)
(
    Usb_ack_wake_up();                 // UDINT.WAKEUPI=0;
    Enable_external_clock();           // CLKSEL0.EXTE=1;
    while (!External_clock_ready());   // while (CLKSTA.EXTON!=1);
    Select_external_clock();           // CLKSEL0.CLKS=1;
    Enable_pll();                      // PLLCSR.PLLE=1;
    Disable_RC_clock();                // CLKSEL0.RCE=0;
    while (!Pll_ready());              // while (PLLCSR.PLOCK!=1);
    Usb_unfreeze_clock();              // USBCON.FRZCLK=0;
)
```

6.12. クロック関係用レジスタ

6.12.1. CLKSEL0 – クロック選択レジスタ0 (Clock selection Register 0)

ビット (\$C5)	7	6	5	4	3	2	1	0	
	RCSUT1	RCSUT0	EXSUT1	EXSUT0	RCE	EXTE	–	CLKS	CLKSEL0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
初期値	0	0	0	0	内容参照	内容参照	0	内容参照	

- **ビット7,6 – RCSUT1,0 : 校正付き内蔵RC発振器用SUT値 (SUT for RC oscillator)**
これら2ビットは校正付き内蔵RC発振器用SUT値です。ヒューズ[※]ビットによってRC発振器が選ばれている場合、そのSUTヒューズ[※]がこれらのビット内に複写されます。この付加起動時間はリセット後だけ使われ、クロック切り替え後に使われないため、ファームウェアでの変更は無効です。
- **ビット5,4 – EXSUT1,0 : 外部クロック信号/クリスタル用低電力発振器用SUT値 (SUT for External Clock/Low Power Crystal Oscillator)**
これら2ビットは外部クロック信号/クリスタル用低電力発振器用SUT値です。ヒューズ[※]ビットによって外部クロック信号/クリスタル用低電力発振器が選ばれている場合、そのSUTヒューズ[※]がこれらのビット内に複写されます。ファームウェアは新しい値を書くことによって、これらのビットを変更できます。この値は外部クロック信号/クリスタル用低電力発振器の次の開始で使われます。
- **ビット3 – RCE : 校正付き内蔵RC発振器許可 (Enable RC Oscillator)**
校正付き内蔵RC発振器を許可するにはRCEビットが論理1を書かれなければなりません。校正付き内蔵RC発振器を禁止するにはRCEビットが論理0を書かれなければなりません。
- **ビット2 – EXTE : 外部クロック信号/クリスタル用低電力発振器許可 (Enable External Clock/Low Power Crystal Oscillator)**
外部クロック信号/クリスタル用低電力発振器を許可するにはEXTEビットが論理1を書かれなければなりません。外部クロック信号/クリスタル用低電力発振器を禁止するにはEXTEビットが論理0を書かれなければなりません。
- **ビット0 – CLKS : クロック選択 (Clock Selector)**
CPUクロックとして外部クロック信号/クリスタル用低電力発振器を選ぶにはCLKSビットが論理1を書かれなければなりません。CPUクロックとして校正付き内蔵RC発振器を選ぶにはCLKSビットが論理0を書かれなければなりません。リセット後、ヒューズ[※]ビット設定によって外部クロック信号/クリスタル用低電力発振器が選ばれているなら、CLKSビットがハードウェアによって設定(1)されます。ファームウェアは選択に先立ってクロックが正しく開始されているかを調べなければなりません。

6.12.2. CLKSEL1 – クロック選択レジスタ1 (Clock selection Register 1)

ビット (\$C6)	7	6	5	4	3	2	1	0	
	RCCKSEL3	RCCKSEL2	RCCKSEL1	RCCKSEL0	EXCKSEL3	EXCKSEL2	EXCKSEL1	EXCKSEL0	CLKSEL1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	1	0	0	0	0	0	

- **ビット7～4 – RCCKSEL3～0 : 校正付き内蔵RC発振器用CKSEL値 (CKSEL for RC oscillator)**
校正付き内蔵RC発振器用クロック設定。リセット後、本レジスタ内のこの部分は校正付き内蔵RC発振器に対応する値'0010'が格納されます。校正付き内蔵RC発振器への切り替え前のファームウェアによるこの値の変更は、校正付き内蔵RC発振器が開始しなくなるため禁止されます。
 - **ビット3～0 – EXCKSEL3～0 : 外部クロック信号/クリスタル用低電力発振器用CKSEL値 (CKSEL for External Clock/Low Power Crystal Oscillator)**
外部クロック信号/クリスタル用低電力発振器用クロック設定。リセット後、ヒューズ[※]ビットによって外部クロック信号/クリスタル用低電力発振器が選ばれているなら、本レジスタ内のこの部分はそのヒューズ[※]設定が格納されます。ファームウェアはクロック切り替え後、起動時間を変更するためにこれを修正できます。
- EXCKSEL3～0構成設定については18頁の表6-1. クロック元選択をご覧ください。校正付き内蔵RC発振器、クリスタル用低電力発振器、外部クロック信号だけが許されます。

6.12.3. CLKSTA – クロック状態レジスタ (Clock Status Register)

ビット (\$C7)	7	6	5	4	3	2	1	0	
	–	–	–	–	–	–	RCON	EXTON	CLKSTA
Read/Write	R	R	R	R	R	R	R	R	
初期値	0	0	0	0	0	0	内容参照	内容参照	

●ビット7～2 – 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1 – RCON : 校正付き内蔵RC発振器ON (RC Oscillator On)

校正付き内蔵RC発振器が走行中なら、本ビットはハードウェアによって設定(1)されます。校正付き内蔵RC発振器が停止なら、本ビットはハードウェアによって解除(0)されます。

●ビット0 – EXTON : 外部クロック信号/クリスタル用低電力発振器ON (External Clock/Low Power Crystal Oscillator On)

外部クロック信号/クリスタル用低電力発振器が走行中なら、本ビットはハードウェアによって設定(1)されます。外部クロック信号/クリスタル用低電力発振器が停止なら、本ビットはハードウェアによって解除(0)されます。

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

ビット (\$61)	7	6	5	4	3	2	1	0	
	CLKPCE	–	–	–	CLKPS3	CLKPS2	CLKPS1	CLKPS0	CLKPR
Read/Write	R/W	R	R	R	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	内容参照	内容参照	内容参照	内容参照	

●ビット7 – CLKPCE : クロック分周値変更許可 (Clock Prescaler Change Enable)

CLKPSビットの変更を許可するためにCLKPCEビットは論理1を書かれなければなりません。CLKPCEビットは同時にCLKPRの他の全ビットが0を書かれる時だけ更新されます。CLKPCEは書き込み後4クロック周期またはCLKPSビット書き込み時、ハードウェアによって解除(0)されます。この制限時間(4クロック周期)内のCLKPCEビット再書き込みは制限時間の延長もCLKPCEビットの解除(0)も行いません。

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

これらのビットは選択したクロック元と内部システムクロック間の分周値を定義します。これらのビットは応用の必要条件に合わせた各種クロック周波数を実行時に書けます。分周値が使われると、分周器はMCUへの主クロックを分周し、全ての同期周辺機能の速度が減じられます。分周値は下表で与えられます。

CKDIV8ヒューズがCLKPSビットの初期値を決めます。CKDIV8が非プログラム(1)にされると、CLKPSビットは'0000'にリセットされます。CKDIV8がプログラム(0)されると、CLKPSビットは起動時に8分周を与える'0011'にリセットされます。現在の動作条件でデバイスの最高周波数より高い周波数のクロック元を選択した場合、この機能が使われるべきです。CKDIV8ヒューズ設定に拘らず、どの値もCLKPSビットへ書けることに注意してください。応用ソフトウェアは現在の動作条件でデバイスの最高周波数より高い周波数のクロック元を選択した場合、十分な分周値が選択されることを保証しなければなりません。このデバイスはCKDIV8ヒューズがプログラム(0)で出荷されます。

表6-9. クロック前置分周器選択														
CLKPS3	0								1					
CLKPS2	0				1				0				1	
CLKPS1	0		1		0		1		0		1		0	
CLKPS0	0	1	0	1	0	1	0	1	0	1	0	1	0	1
分周値(数)	1	2	4	8	16	32	64	128	256	(予約)				

6.12.5. PLLCSR – PLL制御/状態レジスタ (PLL Control and Status Register)

ビット \$29 (\$49)	7	6	5	4	3	2	1	0	
	–	–	–	PINDIV	–	–	PLLE	PLOCK	PLLCSR
Read/Write	R	R	R	R/W	R	R	R/W	R	
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 – PINDIV : PLL入力前置分周器選択(1:1,1:2) (PLL Input Prescaler)

このビットは8または16MHz入力のどちらかからPLL用の8MHz入力クロックを生成するためのPLL入力前置分周器の構成設定を可能にします。

8MHzクロック元使用時、このビットはPLLを許可する前に0を設定しなければなりません(1:1)。

16MHzクロック元使用時、このビットはPLLを許可する前に1を設定しなければなりません(1:2)。

●ビット3,2 – 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1 – PLLE : PLL許可 (PLL Enable)

PLLEが設定(1)されると、PLLが開始されます。PLLEビットがPINMUX(PLL周波数制御(PLLFRQ)レジスタ参照)と共に設定(1)されると、校正付き内蔵RC発振器が自動的に許可されることに注意してください。パワーダウン動作へ移行する前に、内蔵RC発振器を停止して余分な消費を避けるため、PLLは禁止されなければなりません。

●ビット0 – PLOCK : PLL固定化検出 (PLL Lock Detector)

PLOCKビットが設定(1)されると、PLLは基準クロックに固定化されます。PLL許可後、PLLが固定化するのにms単位でかなり多くの時間がかかります。PLOCKを解除(0)するにはPLL許可(PLLE)ビットを解除(0)してください。

6.12.6. PLLFRQ – PLL周波数制御レジスタ (PLL Frequency Control Register)

ビット	7	6	5	4	3	2	1	0	
\$32 (\$52)	PINMUX	PLLUSB	PLLTm1	PLLTm0	PDIV3	PDIV2	PDIV1	PDIV0	PLLFRQ
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	1	0	0	

●ビット7 – PINMUX : PLL入力多重器 (PLL Input Multiplexer)

このビットはPLLのクロック入力を選びます。

- PINMUX=0: PLL入力は供給元として主システムクロックを持つPLL前置分周器に接続されます。
- PINMUX=1: PLL入力は校正付き8MHz内蔵RC発振器に直接接続されます。この動作はクリスタルなし、または8/16MHzと異なる値のクリスタルの使用に於いて、USB低速(Low-speed)動作種別での動作を可能にします。

●ビット6 – PLLUSB : USB周辺機能用PLL後置分周器選択 (PLL Postscaler for USB Peripheral)

このビットはPLL出力周波数とUSB部入力周波数間の分周係数を選びます。

- PLLUSB=0: 分周なし、直結。(PLL出力=48MHzの場合)
- PLLUSB=1: PLL周波数は2分周され、USB部へ送られます。(PLL出力=96MHzの場合)

●ビット5,4 – PLLTm1,0 : 高速タイマ用PLL後置分周器選択 (PLL Postscaler for High Speed Timer)

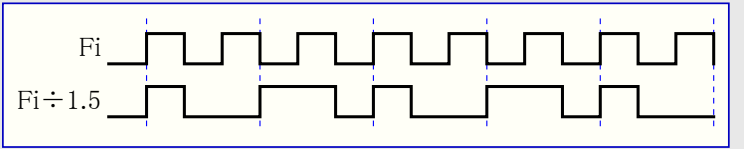
これらのビットはPLL出力周波数と高速タイマ入力周波数間の分周係数に関して符号化します。

分周係数1.5はクロック内にいくつかの震え(ジッタ)を誘引しますが、平均デューティサイクルが50%なので誤差を無効に保つことに注意してください。より多くの詳細については図6-7をご覧ください。

表6-10. 高速タイマ用後置分周器選択

PLLTm1	PLLTm0	分周数	高速タイマ クロック種別
0	0	0 (切断)	同期クロック(CK)
0	1	1	
1	0	1.5	非同期クロック(PCK)
1	1	2	

図6-7. 分周係数=1.5でのPLL後置分周器動作



●ビット3~0 – PDIV3~0 : PLL固定化周波数 (PLL Lock Frequency)

これらのビットは必要とする出力周波数値に従ってPLL内部VCO基準クロックの構成設定を行います。

表6-11. PLL出力周波数選択

PDIV3	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
PDIV2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1
PDIV1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1
PDIV0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
出力周波数(MHz)	不可	不可	不可	40	48	56	不可	72	80	88	96	不可	不可	不可	不可
備考					既定										

5Vでの最適なPLL構成設定はPLL出力周波数=96MHzで、64MHzの高速タイマクロックを生成するために1.5分周し、48MHzのUSBクロックを生成するために2分周します。

7. 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**休止形態制御レジスタ(SMCR)**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。SMCR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7-2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。

17頁の図6-1はATmega16U4/32U4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手助けになります。

7.1. 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、USB、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。

7.2. 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、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、(clkUSBを含む)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、本動作に移行すると、変換が自動的に始まります。A/D変換完了割り込みからの他、外部リセット、ウォッチドッグシステムリセット、ウォッチドッグ割り込み、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、SPM/EEPROM操作可割り込み、INT6の外部レベル割り込み、INT3~0の外部割り込み、ピン変化割り込みだけが、A/D変換雑音低減動作からMCUを起動できます。

7.3. 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、INT6の外部レベル割り込み、INT3~0の外部割り込み、ピン変化割り込み、USB非同期割り込み元(VBUSTIとWAKEUP)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38頁の「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18頁の「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。

7.4. 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AT90USB64/128との共通性のために未だ存在しますが、本デバイスにはタイマ/カウンタ2の非同期動作が存在しないので、この動作はパワーダウン動作と同じです。

7.5. 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。

7.6. 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AT90USB64/128との共通性のために未だ存在しますが、本デバイスにはタイマ/カウンタ2の非同期動作が存在しないので、この動作はスタンバイ動作と同じです。

表7-1. 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	主クロック 供給元	INT _n , ピン変化 割り込み	TWI アドレス 一致	SPM EEPROM 操作可	A/D変換 完了	ウォッチ ドッグ	USB 同期系	USB 非同期系	その他 I/O
アイドル			○	○	○	○	○	○	○	○	○	③	○
A/D変換雑音低減				○	○	②	○	○	○	○	○	③	
パワーダウン						②	○			○		③	
パワーセーブ						②	○			○		③	
スタンバイ(注1)					○	②	○			○		③	
拡張スタンバイ(注1)					○	②	○			○		③	

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だけの推奨です。

② INT6についてはレベル割り込みだけです。

③ USB非同期系割り込みはVBUSTI、WAKEUPIです。

7.7. 電力削減

電力削減レジスタ(PRR_n)は消費電力を削減するために個別周辺機能へのクロックを停止する方法を提供します。周辺機能は現状で固定化され、I/Oレジスタは読み込みも書き込みもできません。クロックを停止している時に周辺機能によって使われていた資源は占有されたままなので、その周辺機能は殆どの場合、クロックを停止する前に禁止されるべきです。周辺機能部の起動は電力削減レジスタ(PRR_n)のビットを解除(0)することによって行い、その周辺機能部を停止前と同じ状態にします。

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

7.8. 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。

7.8.1. 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。節電するため、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194頁の「A/D変換器」を参照してください。

7.8.2. 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。さもないと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192頁の「アナログ比較器」を参照してください。

7.8.3. 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BODLEVEL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31頁の「低電圧検出 (BOD)」を参照してください。

7.8.4. 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目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32頁の「内部基準電圧」を参照してください。

7.8.5. ウォッチドッグ タイマ

ウォッチドッグ タイマが応用で必要とされないなら、この単位部はOFF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32頁の「ウォッチドッグ タイマ」を参照してください。

7.8.6. 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(clkI/O)とA/D変換クロック(clkADC)の両方が停止される休止形態ではデバイスの入力緩衝器が禁止されます。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。入力緩衝器が許可され、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、入力緩衝器は過大な電力を消費するでしょう。

アナログ入力ピンに対するデジタル入力緩衝器は常に禁止されるべきです。入力ピンでのVCC/2付近のアナログ信号入力では活動動作でも重要な電流を引き起こし得ます。デジタル入力緩衝器はデジタル入力禁止レジスタ(DIDR0,DIDR1,DIDR2)の書き込みによって禁止できます。詳細については193頁の「DIDR1 – デジタル入力禁止レジスタ1」、205頁の「DIDR0 – デジタル入力禁止レジスタ0」と「DIDR2 – デジタル入力禁止レジスタ2」を参照してください。

7.8.7. 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OCDENBUS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。

- OCDENBUSを禁止 (非プログラム(1)設定)
- JTAGENBUSを禁止 (非プログラム(1)設定)
- MCU制御レジスタ(MCUCR)のJTAG禁止(JTD)ビットへの1書き込み

7.9. 電力管理用レジスタ

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

この休止形態制御レジスタは電力管理用の制御ビットを含みます。

ビット	7	6	5	4	3	2	1	0	
\$33 (\$53)	–	–	–	–	SM2	SM1	SM0	SE	SMCR
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3~1 – SM2~0 : 休止種別選択 (Sleep Mode Select Bit 2, 1 and 0)

これらのビットは表7-2.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7-2. 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の使用だけが推奨されます。

- ビット0 – SE : 休止許可 (Sleep Enable)

SLEEP命令が実行される時にMCUを休止形態へ移行させるには、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。

7.9.2. PRR0 – 電力削減レジスタ0 (Power Reduction Register 0)

ビット	7	6	5	4	3	2	1	0	
(\$64)	PRTWI	–	PRTIM0	–	PRTIM1	PRSPI	–	PRADC	PRR0
Read/Write	R/W	R	R/W	R	R/W	R/W	R	R/W	
初期値	0	0	0	0	0	0	0	0	

- ビット7 – PRTWI : 2線直列インターフェース電力削減 (Power Reduction TWI)

このビットへの1書き込みはその部分へのクロック停止によって2線直列インターフェース(TWI)を停止します。TWIの再起動時、TWIは正しい動作を保証するために再初期化されるべきです。

- ビット6 – 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5 – PRTIM0 : タイマ/カウンタ0電力削減 (Power Reduction Timer/Counter0)**

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

● **ビット4 – 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3 – PRTIM1 : タイマ/カウンタ1電力削減 (Power Reduction Timer/Counter1)**

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

● **ビット2 – PRSPI : 直列周辺インターフェース電力削減 (Power Reduction Serial Peripheral Interface)**

このビットへの1書き込みはその部分へのクロック停止によって**直列周辺インターフェース(SPI)**を停止します。SPIの再起動時、SPIは正しい動作を保証するために再初期化されるべきです。

● **ビット1 – 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0 – PRADC : A/D変換器電力削減 (Power Reduction ADC)**

このビットへの1書き込みは**A/D変換器(ADC)**を停止します。A/D変換器は停止前に**禁止**されなければなりません。A/D変換器停止時、**アナログ比較器**は**ADC入力切替器**を使えません。

注: アナログ比較器は193頁の「**ACSR – アナログ比較器制御/状態レジスタ**」での**アナログ比較器禁止(ACD)**ビットの使用で禁止されます。(誤注: 共通性から本注追加)

7.9.3. PRR1 – 電力削減レジスタ1 (Power Reduction Register 1)

ビット (\$65)	7	6	5	4	3	2	1	0	
	PRUSB	–	–	PRTIM4	PRTIM3	–	–	PRUSART	PRR1
Read/Write	R/W	R	R	R/W	R/W	R	R	R/W	
初期値	0	0	0	0	0	0	0	0	

● **ビット7 – PRUSB : USB電力削減 (Power Reduction USB)**

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

● **ビット6,5 – 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 – PRTIM4 : タイマ/カウンタ4電力削減 (Power Reduction Timer/Counter4)**

このビットへの1書き込みは**タイマ/カウンタ4**部を停止します。タイマ/カウンタ4が許可されると、停止前と同様に動作は継続します。

● **ビット3 – PRTIM3 : タイマ/カウンタ3電力削減 (Power Reduction Timer/Counter3)**

このビットへの1書き込みは**タイマ/カウンタ3**部を停止します。タイマ/カウンタ3が許可されると、停止前と同様に動作は継続します。

● **ビット2,1 – 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。

● **ビット0 – PRUSART : USART電力削減 (Power Reduction USART)**

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

8. システム制御とリセット

8.1. 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、きっ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、割り込みベクタがフート領域の場合やその逆も同様です。図8-1の回路構成図はリセット論理回路を示します。249頁の表29-3はリセット回路の電気的特性を定義します。

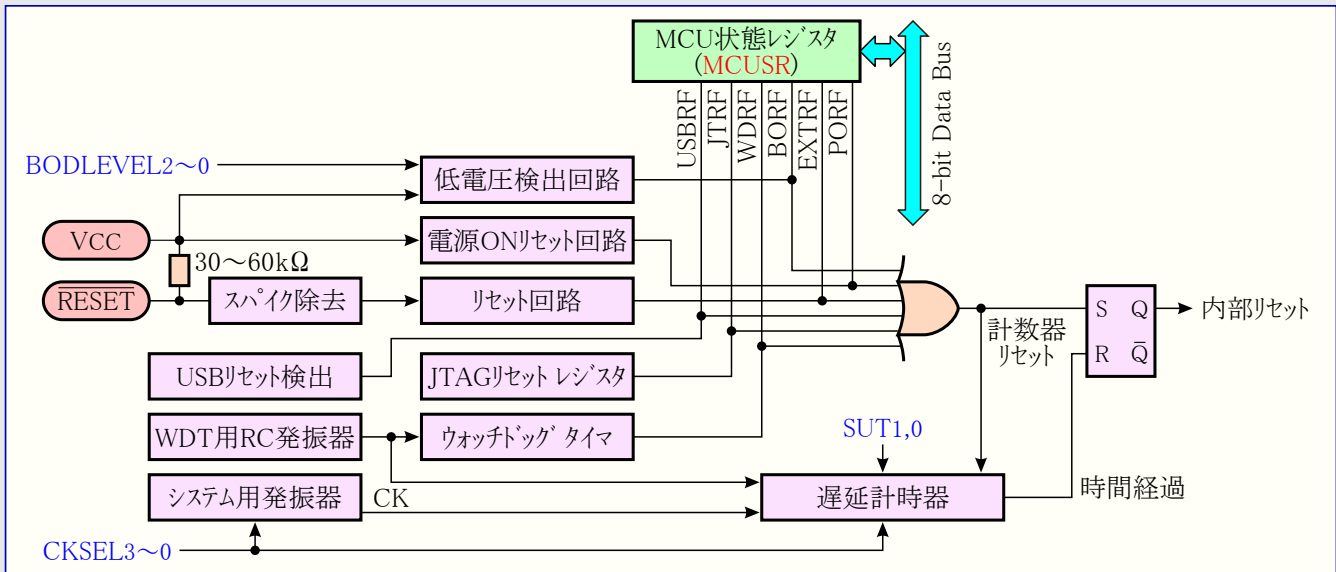
AVRのI/Oポートはリセット元が有効になると直ちにそれらの初期状態にリセットされます。これはどのクロック元の走行も必要ありません。全てのリセット元が無効にされてしまった後、遅延計数器(タイマ)が始動され、内部リセットを引き伸ばします。これは通常動作開始前に安定電圧へ達することを電源に許します。遅延タイマの遅延時間はSUTヒューズとCKSELヒューズを通して使用者によって定義されます。この遅延時間についての各種選択は18頁の「クロック元」で示されます。

8.2. リセット元

ATmega16U4/32U4には次の6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_{POT})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** 低電圧検出が許可され、供給電圧(V_{CC})が低電圧検出電圧(V_{BOT})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210頁の「IEEE 1149.1 (JTAG) 境界走査」を参照してください。
- **USBリセット** 使用者によってこの機能が許可されているなら、USBリセット条件の終了検出で、(USB制御器が許可で接続状態に留まることを除いて)MCUはリセットされます。

図8-1. リセット回路構成



8.3. 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249頁の表29-3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から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8-2. 内蔵電源ONリセット (RESETはVCCに接続)

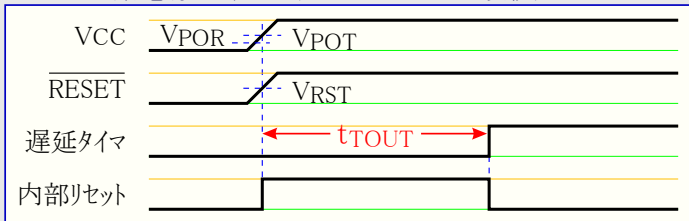
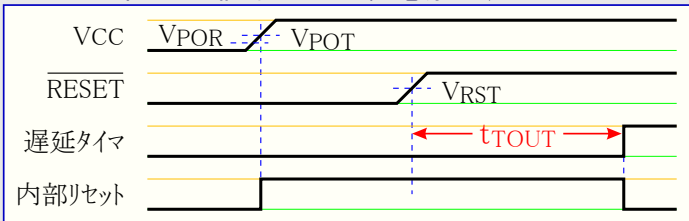


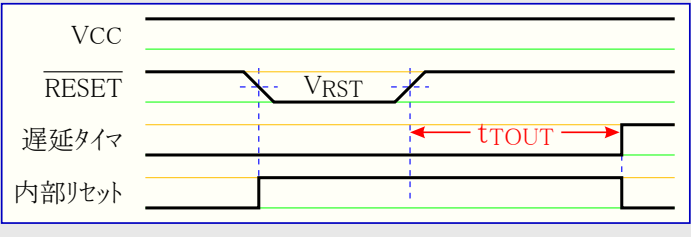
図8-3. 外部RESET信号による延長電源ONリセット



8.4. 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(249頁の表29-3を参照)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(VRST)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(tTOUT)経過後にMCUを始動します。

図8-4. 動作中の外部リセット



8.5. 低電圧(ブラウンアウト)検出リセット

本デバイス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ヒューズによって選べます。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VBOT+=VBOT+VHYST/2、VBOT-=VBOT-VHYST/2と解釈すべきです。

BODが許可され、VCCが起動電圧以下の値に下降すると(図8-5のVBOT-)、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8-5のVBOT+)、(遅延タイマが起動されて)遅延タイマは遅延時間(tTOUT)経過後にMCUを始動します。

BOD回路は電圧が表8-2.で与えられるtBOD時間より長く起動電圧以下に留まる場合のみ、VCCでの低下を検出します。

図8-5. 動作中の低電圧検出リセット

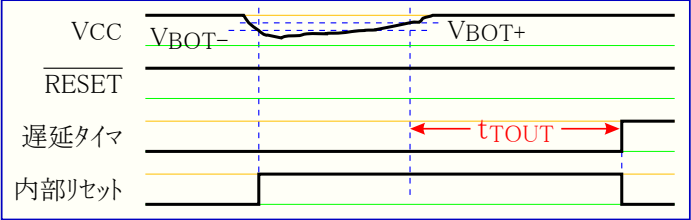


表8-1. BODLEVELヒューズ (VBOT) 設定

BODLEVEL2~0	000	001	010	011	100	101	110	111
最大	4.5	3.7	3.6	2.8	2.6	2.4	2.2	低電圧検出 (BOD) リセット禁止
代表	4.3	3.5	3.4	2.6	2.4	2.2	2.0	
最小	4.0	3.3	3.2	2.4	2.2	2.0	1.8	
単位	V							

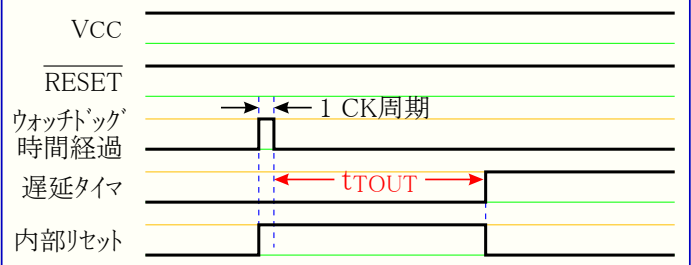
表8-2. 低電圧検出(BOD)特性

シンボル	項目	最小	代表	最大	単位
VHYST	低電圧検出ヒステリシス電圧		50		mV
tBOD	最小低電圧検出時間		TBD		μs

8.6. 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)1CK周期幅の短いリセットパルスを生成します。このパルスの下降端で遅延タイマは遅延時間(tTOUT)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32頁の「ウォッチドッグ タイマ」を参照してください。

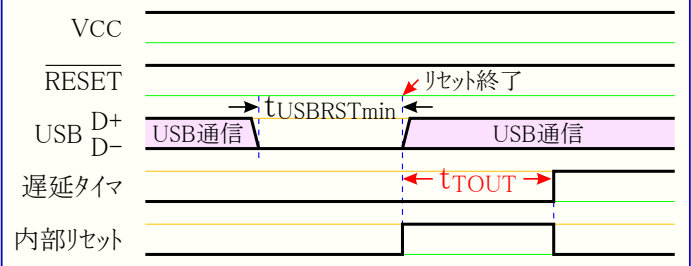
図8-6. 動作中のウォッチドッグ リセット



8.7. USBリセット

USB部が許可され、USBリセットCPU機能許可で構成設定されている時にバス上で有効なUSBリセット信号が検出された場合にCPUコアはリセットされますが、USB制御器は許可で且つ接続状態に留まります。これはデバイスの信頼性強化に使われるかもしれません。

図8-7. 動作中のUSBリセット



8.8. 内部基準電圧

ATmega16U4/32U4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。

8.8.1. 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8-3.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 (BODLEVELヒューズ'のプログラム(0)により)
- 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A(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8-3. 内部基準電圧特性						
シンボル	項目	条件	最小	代表	最大	単位
V _{BG}	基準電圧	VCC=2.7V TA=25°C	1.0	1.1	1.2	V
t _{BG}	起動時間			40	70	μs
I _{BG}	消費電流			10		μA

8.9. ウォッチドッグ タイマ

ATmega16U4/32U4には強化されたウォッチドッグ タイマ(WDT)があります。主な機能を次に示します。

- 独立した内蔵発振器からのクロック駆動
- 3つの動作種別
 - 割り込み
 - システム リセット
 - 割り込みとシステム リセット
- 選択可能な16ms〜8sの計時完了時間
- 安全動作用のウォッチドッグ常時ONハートウェア ヒューズ

ウォッチドッグ タイマ(WDT)はチップ上の独立した128kHz発振器の間隔で計時するタイマです。WDTは計数器が与えられた計時完了値に達した時に割り込みまたはシステム リセットを与えます。通常動作では計時完了値へ達する前に計数器を再始動するために、システムがウォッチドッグ リセット(WDR)命令を使う必要があります。システムが計数器を再始動しなければ、割り込みまたはシステム リセットが起こるでしょう。

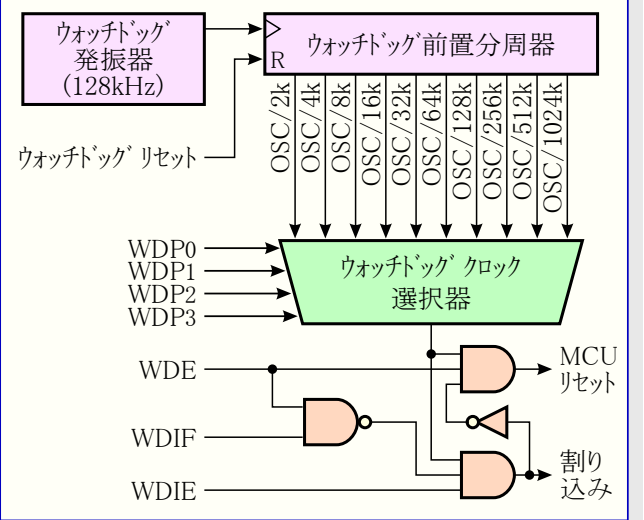
割り込み動作種別では、タイマ計時完了時にWDTが割り込みを生じます。この割り込みは休止形態からデバイスを起動するためや、一般的なシステム タイマとしても使えます。1つの例は或る動作に対して許された最大時間を制限することで、その動作が予測されたより長く走行する時に割り込みを生じます。システム リセット動作種別ではタイマ計時完了時にWDTがリセットを生じます。これは一般的にコード外走行の場合の中断を防止するのに使われます。3つ目の動作種別は先に割り込みを生じ、その後にシステム リセット動作種別に切り替えることで、他の2つの動作種別の組み合わせとなる、割り込み及びシステム リセット動作種別です。この動作種別は例えばシステム リセットに先立って重要なパラメータを保存することによって安全な停止を許します。

ウォッチドッグ常時ON(WDTON)ヒューズ'のプログラム(0)はウォッチドッグ タイマをシステム リセット動作種別に強制します。このヒューズ'のプログラム(0)でシステム リセット動作(WDE)ビットと割り込み動作(WDIE)ビットは各々、'1'と'0'に固定されます。

更にプログラム保護を保証するためにウォッチドッグ設定の変更は時間制限手順に従わなければなりません。システム リセット許可(WDE)の解除と計時完了時間設定の変更についての手順は次のとおりです。

- 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(WDP3~0)ビットを書きますが、WDCEビットは解除(0)されてです。これは1操作(命令)で行わなければなりません。

図8-8. ウォッチドッグ タイマ構成図



次のコード例はウォッチドッグ タイマを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:    CLI                ;全割り込み禁止
            WDR                ;ウォッチドッグ タイマ リセット
            IN     R16, MCUSR   ;MCUSR値を取得
            ANDI   R16, ~(1<<WDRF) ;WDRF論理0値を取得
            OUT    MCUSR, R16   ;ウォッチドッグ リセット フラグ(WDRF)解除
            LDS    R16, WDTCSR  ;現WDTCSR値を取得(他ビット保護用)
            ORI    R16, (1<<WDCE) | (1<<WDE) ;WDCEとWDE論理1値を設定
            STS    WDTCSR, R16  ;WDCEとWDEに論理1書き込み
            LDI    R16, (0<<WDE) ;WDE論理0値を取得
            STS    WDTCSR, R16  ;ウォッチドッグ禁止
            SEI                ;全割り込み許可
            RET                ;呼び出し元へ復帰
```

C言語プログラム例

```
void WDT_off(void)
{
    __disable_interrupt(); /* 全割り込み禁止 */
    __watchdog_reset();    /* ウォッチドッグ タイマ リセット */
    MCUSR &= ~(1<<WDRF);   /* ウォッチドッグ リセット フラグ(WDRF)解除 */
    WDTCSR |= (1<<WDCE) | (1<<WDE); /* WDCEとWDEに論理1書き込み */
    WDTCSR = 0x00;        /* ウォッチドッグ禁止 */
    __enable_interrupt();  /* 全割り込み許可 */
}
```

注: 5頁の「コード例」をご覧ください。

注: ウォッチドッグが偶然に許可されると(例えばポインタの逸脱や低電圧(ブラウンアウト)状態)、デバイスはリセットし、ウォッチドッグは許可に留まります。コードがウォッチドッグ操作の初期設定をしなければ、これは計時完了の無限繰り返しを引き起こすかもしれません。この状態を避けるため、応用ソフトウェアは例えウォッチドッグが使われなくても、初期化ルーチンで**WDRFフラグ**と**WDE制御ビット**を常に解除(0)すべきです。

次のコード例はウォッチドッグ タイマの計時完了値変更用のアセンブリ言語とC言語の関数を示します。

アセンブリ言語プログラム例

```
WDT_PRS:    CLI                ;全割り込み禁止
            WDR                ;ウォッチドッグ タイマ リセット
            LDS    R16, WDTCSR  ;現WDTCSR値を取得(他ビット保護用)
            ORI    R16, (1<<WDCE) | (1<<WDE) ;WDCEとWDE論理1値を設定
            STS    WDTCSR, R16  ;WDCEとWDEに論理1書き込み
            LDI    R16, (1<<WDE) | (1<<WDP2) | (1<<WDPO) ;WDE=1,計時間隔=0.5s値を取得
            STS    WDTCSR, R16  ;0.5s監視間隔リセット動作開始
            SEI                ;全割り込み許可
            RET                ;呼び出し元へ復帰
```

C言語プログラム例

```
void WDT_off(void)
{
    __disable_interrupt(); /* 全割り込み禁止 */
    __watchdog_reset();    /* ウォッチドッグ タイマ リセット */
    WDTCSR |= (1<<WDCE) | (1<<WDE); /* WDCEとWDEに論理1書き込み */
    WDTCSR = (1<<WDE) | (1<<WDP2) | (1<<WDPO); /* 0.5s監視間隔リセット動作開始 */
    __enable_interrupt();  /* 全割り込み許可 */
}
```

注: 5頁の「コード例」をご覧ください。

注: ウォッチドッグ タイマ前置分周選択(WDP3~0)ビットの変更がより短い計時完了周期に変わってしまう結果になり得るため、ウォッチドッグ タイマはWDPビットのどんな変更にも先立ってリセット(WDR命令)されるべきです。

8.10. リセット関係用レジスタ

8.10.1. MCUSR – MCU状態レジスタ (MCU Status Register)

MCU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)	–	–	USBRF	JTRF	WDRF	BORF	EXTRF	PORF	MCUSR
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	内容参照	内容参照	内容参照	内容参照	内容参照	内容参照	

● ビット7,6 – 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。これらのビットを設定(1)してはなりません。

● ビット5 – USBRF : USBリセット フラグ (USB Reset Flag)

このビットはUSB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MCUSRを読み、そして解除(0)すべきです。別のリセットが起こる前にこのレジスタが解除(0)されると、そのリセット元はリセット フラグを調べることによって得られます。

8.10.2. WDTCSR – ウォッチドッグ タイマ制御レジスタ (Watchdog Timer Control Register)

ビット (\$60)	7	6	5	4	3	2	1	0	
	WDIF	WDIE	WDP3	WDCE	WDE	WDP2	WDP1	WDP0	WDTCSR
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	不定	0	0	0	

● ビット7 – WDIF : ウォッチドッグ割り込み要求フラグ (Watchdog Interrupt Flag)

ウォッチドッグ タイマが割り込みに設定され、ウォッチドッグ タイマで計時完了が起こると、本ビットが設定(1)されます。対応する割り込み処理ベクタを実行すると、WDIFはハードウェアによって解除(0)されます。代わりにWDIFはこのフラグへの論理1書き込みによっても解除(0)されます。ステータスレジスタ(SREG)の全割り込み許可(I)ビットとウォッチドッグ割り込み許可(WDIE)が設定(1)されていれば、ウォッチドッグ計時完了割り込みが実行されます。

● ビット6 – WDIE : ウォッチドッグ割り込み許可 (Watchdog 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、ウォッチドッグ割り込みが許可されます。この設定(=1)との組み合わせでウォッチドッグ リセット許可(WDE)ビットが解除(0)されると、割り込み動作種別になり、ウォッチドッグ タイマで計時完了が起こると、対応する割り込みが実行されます。

WDEが設定(1)されると、ウォッチドッグ タイマは割り込み及びシステム リセット動作種別になります。ウォッチドッグ タイマでの最初の計時完了がウォッチドッグ割り込み要求(WDIF)フラグを設定(1)します。対応する割り込みベクタの実行はハードウェアによってWDIEとWDIFを自動的に解除(0)します(ウォッチドッグはシステム リセット動作になります)。これは割り込みを使う間のウォッチドッグ リセット保護を維持するのに有用です。割り込み及びシステム リセット動作種別に留まるには、各割り込み後にWDIEが設定(1)されなければなりません。然しながら、ウォッチドッグ システム リセット動作種別の安全機能に危険を及ぼすかもしれないため、これは割り込み処理ルーチン自身内で行われるべきではありません。次の計時完了に先立って割り込みが実行されない場合、システム リセットが適用(実行)されます。

表8-4. ウォッチドッグ タイマ設定				
WDTON	WDE	WDIE	動作種別	計時完了での動作
1	0	0	停止	なし
1	0	1	割り込み	割り込み
1	1	0	システム リセット	リセット
1	1	1	割り込み及びシステム リセット	割り込み、その後システム リセット動作種別
0	x	x	システム リセット	リセット

(訳注)
左表のWDTONヒューズ値はプログラム=0、非プログラム=1で表記しています。従って原書の表と逆論理になっています。

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

このビットはウォッチドッグ リセット許可(WDE)と前置分周器ビットの変更用の時間制限手順で使われます。WDEビットの解除(0)や前置分周器ビット変更のため、WDCEは設定(1)されなければなりません。一旦1を書かれると、4クロック周期後にハードウェアがWDCEを解除(0)します。

● ビット3 – WDE : ウォッチドッグ リセット許可 (Watchdog System Reset Enable)

WDEはMCU状態レジスタ(MCUSR)のウォッチドッグ リセット フラグ(WDRF)によって無効にされます。これはWDRFが設定(1)されると、WDEが常に設定(1)されることを意味します。WDEを解除(0)するにはWDRFが先に解除(0)されなければなりません。この特徴は失敗を引き起こす状態中の複数リセットと失敗後の安全な起動を保証します。

● ビット5,2~0 – WDP3~0 : ウォッチドッグ タイマ前置分周選択 (Watchdog Timer Prescaler 3,2,1 and 0)

このWDP3~0ビットはウォッチドッグ タイマが走行する時のウォッチドッグ タイマの前置分周を決めます。各種前置分周値と対応する計時完了周期は下表で示されます。

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

9. 割り込み

この項はATmega16U4/32U4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9頁の「リセットと割り込みの扱い」を参照してください。

9.1. ATmega16U4/32U4の割り込みベクタ

表9-1. リセットと割り込みのベクタ

ベクタ番号	プログラム アドレス (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	Reserve	(予約)
7	\$000C	Reserve	(予約)
8	\$000E	INT6	外部割り込み要求6
9	\$0010	Reserve	(予約)
10	\$0012	PCINT0 (PCIO)	ピン変化0群割り込み要求
11	\$0014	USB一般	USB一般割り込み要求
12	\$0016	USBエンドポイント	USBエンドポイント割り込み要求
13	\$0018	ウォッチドッグ WDT	ウォッチドッグ計時完了
14	\$001A	Reserve	(予約)
15	\$001C	Reserve	(予約)
16	\$001E	Reserve	(予約)
17	\$0020	タイマ/カウンタ1 CAPT	タイマ/カウンタ1捕獲発生
18	\$0022	タイマ/カウンタ1 COMPA	タイマ/カウンタ1比較A一致
19	\$0024	タイマ/カウンタ1 COMPB	タイマ/カウンタ1比較B一致
20	\$0026	タイマ/カウンタ1 COMPC	タイマ/カウンタ1比較C一致
21	\$0028	タイマ/カウンタ1 OVF	タイマ/カウンタ1溢れ
22	\$002A	タイマ/カウンタ0 COMPA	タイマ/カウンタ0比較A一致
23	\$002C	タイマ/カウンタ0 COMPB	タイマ/カウンタ0比較B一致
24	\$002E	タイマ/カウンタ0 OVF	タイマ/カウンタ0溢れ
25	\$0030	SPI STC	SPI 転送完了
26	\$0032	USART RX	USART 受信完了
27	\$0034	USART UDRE	USART 送信緩衝部空き
28	\$0036	USART TX	USART 送信完了
29	\$0038	アナログ比較器 ANA_COMP	アナログ比較器出力遷移
30	\$003A	A/D変換器 ADC	A/D変換完了
31	\$003C	EEPROM EE_RDY	EEPROM 操作可
32	\$003E	タイマ/カウンタ3 CAPT	タイマ/カウンタ3捕獲発生
33	\$0040	タイマ/カウンタ3 COMPA	タイマ/カウンタ3比較A一致
34	\$0042	タイマ/カウンタ3 COMPB	タイマ/カウンタ3比較B一致
35	\$0044	タイマ/カウンタ3 COMPC	タイマ/カウンタ3比較C一致
36	\$0046	タイマ/カウンタ3 OVF	タイマ/カウンタ3溢れ
37	\$0048	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
38	\$004A	SPM命令 SPM_RDY	SPM命令操作可
39	\$004C	タイマ/カウンタ4 COMPA	タイマ/カウンタ4比較A一致
40	\$004E	タイマ/カウンタ4 COMPB	タイマ/カウンタ4比較B一致
41	\$0050	タイマ/カウンタ4 COMPD	タイマ/カウンタ4比較D一致
42	\$0052	タイマ/カウンタ4 OVF	タイマ/カウンタ4溢れ
43	\$0054	タイマ/カウンタ4 FPF	タイマ/カウンタ4障害保護

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。215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。

本表は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のなら、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9-2. 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225頁の表27-8で示されます。

9.1.1. 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。

9.2. ベクタ移動用レジスタ

9.2.1. MCUCR – MCU制御レジスタ (MCU Control Register)

ビット	7	6	5	4	3	2	1	0	
\$35 (\$55)	JTD	–	–	PUD	–	–	IVSEL	IVCE	MCUCR
Read/Write	R/W	R	R	R/W	R	R	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217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215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 */
}
```

10. 外部割り込み

外部割り込みはINT6,3~0ピンまたはPCINT0~7ピンの何れかによって起動されます。許可したなら、例えINT6,3~0またはPCINT0~7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の生成方法を提供します。

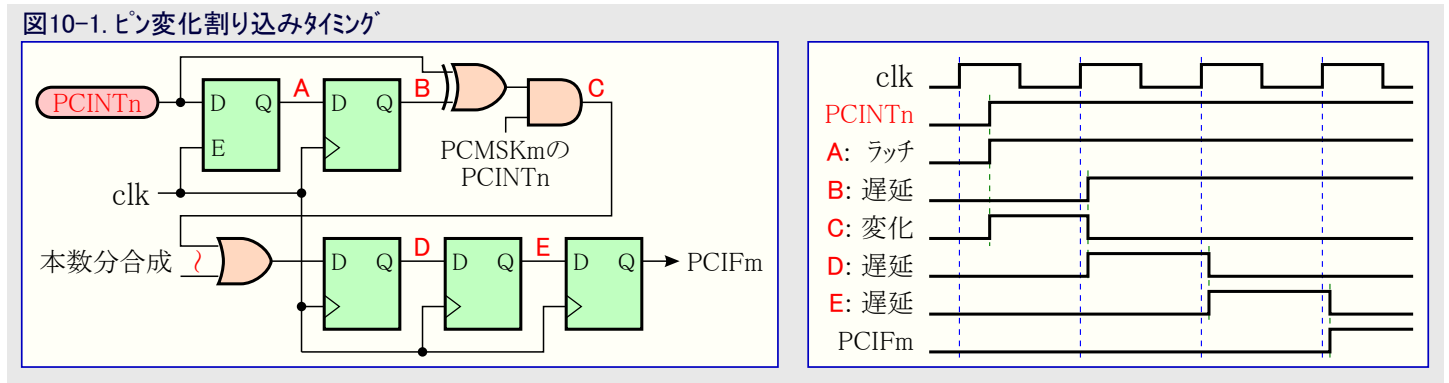
ピン変化割り込みPCIOは許可したPCINT0~7の何れかが切り替わると起動します。ピン変化割り込み許可レジスタ0(PCMSK0)は、どのピンがピン変化割り込み要因となるかを制御します。PCINT0~7のピン変化割り込みは非同期に検知されます。これはそれらの割り込みがアイドル動作以外の休止形態からもデバイスを起動するのに使えることを意味します。

INT7~0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(INT6))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かつ許可されると、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。INT6の上昇端または下降端割り込みの認知は17ページの「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。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、この必要としたレベルはレベル割り込みを起動する完全な起動復帰のため、MCUに対して充分長く保持されなければならないことに注意してください。このレベルが起動時間の最後に先立って消滅すると、MCUは今までどおり起動しますが、割り込みが生成されません。起動時間は17ページの「システムクロックとクロック選択」で示されるようにSUTヒューズとCKSELヒューズで定義されます。

10.1. ピン変化割り込みタイミング (訳注: 共通性から本項追加)

ピン変化割り込みの例は図10-1.で示されます。



10.2. 外部割り込み用レジスタ

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

外部割り込み制御レジスタAは割り込み条件制御ビットを含みます。

ビット (\$69)	7	6	5	4	3	2	1	0	
	ISC31	ISC30	ISC21	ISC20	ISC11	ISC10	ISC01	ISC00	EICRA
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 – ISC31,0~ISC01,0 : 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(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)とレベルは右表で定義されます。INT3~0の端は非同期に記録されます。下表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(継続的に)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を書くことにより解除(0)されるべきです。

表10-1. 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	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=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10-2. 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 _{INT}	非同期外部割り込み最小パルス幅		50		ns

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

ビット (\$6A)	7	6	5	4	3	2	1	0	
	–	–	ISC61	ISC60	–	–	–	–	EICRB
Read/Write	R	R	R/W	R/W	R	R	R	R	
初期値	0	0	0	0	0	0	0	0	

● ビット7,6 – Res : 予約 (Reserved)

これらのビットは予約されており、常に0として読みます。

● ビット5,4 – ISC61,0 : 外部割り込み6条件制御 (External Interrupt6 Sense Control Bits)

外部割り込み6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INT6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INT6ピンの端(エッジ)とレベルは右表で定義されます。INT6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クロック周期よりも長く留まるパルスは割り込みを生成します。より短いパルスは割り込みの発生が保証されません。**システムクロック前置分周器**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10-3. 外部割り込み6 割り込み条件

ISC61	ISC60	割り込み発生条件
0	0	INT6ピンのLowレベル
0	1	INT6ピンの論理変化(両端)
1	0	INT6ピンの下降端
1	1	INT6ピンの上昇端

注: ISC61,0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)により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。

● ビット3~0 – 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。

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

ビット \$1D (\$3D)	7	6	5	4	3	2	1	0	
	–	INT6	–	–	INT3	INT2	INT1	INT0	EIMSK
Read/Write	R	R/W	R	R	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

● ビット6,3~0 – INT6,INT3~INT0 : 外部割り込み6,3~0 許可 (External Interrupt Request 6,3~0 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、INT6,3~0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。例えば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。

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

ビット \$1C (\$3C)	7	6	5	4	3	2	1	0	
	–	INTF6	–	–	INTF3	INTF2	INTF1	INTF0	EIFR
Read/Write	R	R/W	R	R	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

● ビット6,3~0 – INTF6,INTF3~INTF0 : 外部割り込み6,3~0要求フラグ (External Interrupt Flag6,3~0)

INT6,3~0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、INTF6,3~0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(INT6,3~0)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解除(0)できます。INT6,3~0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。

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

ビット (\$68)	7	6	5	4	3	2	1	0	
	–	–	–	–	–	–	–	PCIE0	PCICR
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) (訳注: 共通性から本項追加)

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

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

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

ビット \$1B (\$3B)	7	6	5	4	3	2	1	0	
	–	–	–	–	–	–	–	PCIF0	PCIFR
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) (訳注: 共通性から本項追加)

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

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

10.2.7. PCMSK0 – ピン変化割り込み許可レジスタ0 (Pin Change Enable Mask 0～7)

ビット (\$6B)	7	6	5	4	3	2	1	0	
	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0	PCMSK0
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 – PCINT7～PCINT0 : ピン変化割り込み7～0許可 (Pin Change Enable Mask 7～0)

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

11. 入出力ポート

11.1. 序説

全てのAVRのポートは標準デジタルI/Oポートとして使われる時に真の読み-変更-書き(リード-モディファイライト)を機能的に持ちます。これはSBIとCBI命令で他のどのピンの方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11-1.で示されるようにVCCとGNDの両方に保護ダイオードがあります。各値の完全な一覧については248頁の「電気的特性」を参照してください。

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PORTx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53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。けれどもPINxレジスタのビットへの論理1書き込みは、出力レジスタの対応ビット値を(1/0)反転する結果になります。加えてMCU制御レジスタ(MCUCR)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。

11.2. 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11-2.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11-1. 入出力ピン等価回路

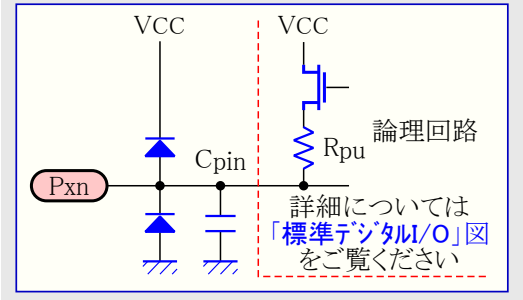
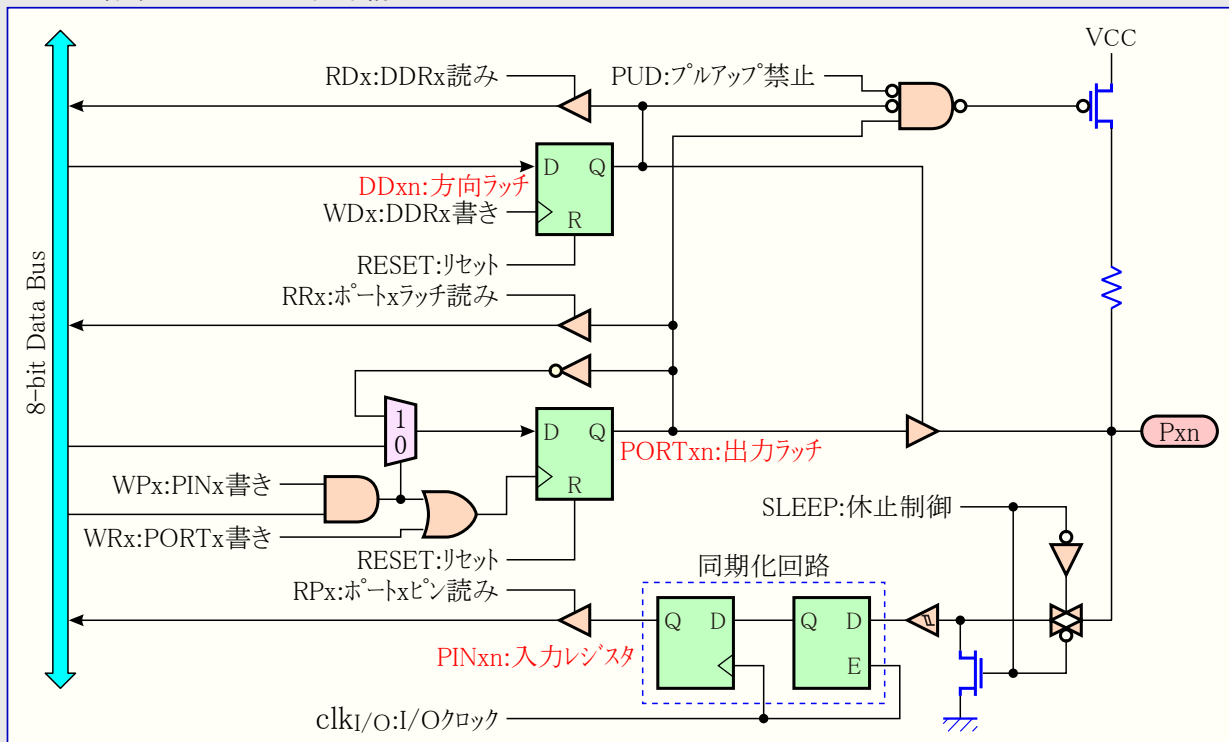


図11-2. 標準デジタル入出力回路構成



注: WRx, 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。

11.2.1. 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。53頁の「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。

11.2.2. ピンの出力交互切り替え

PINxnへの論理1書き込みはDDRxnの値に拘らず、PORTxnの値を反転切り替えます。SBI命令がポート内の1ビットの反転切り替えに使えることに注目してください。

11.2.3. 入出力間の切り替え

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ために、MCU制御レジスタ(MCUCR)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11-1. はピン値に対する制御信号の一覧を示します。

表11-1. ポートピンの設定					
DDxn	PORTxn	PUD (MCUCR)	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

11.2.4. ピン値の読み込み

DDxn方向ビットの設定に関係なく、ポートピンはPINxnレジスタビットを通して読めます。図11-2. で示されるようにPINxnレジスタビットと先行するラッチは同期化回路を構成します。これは物理ピンが内部クロックの端(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11-3. 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tpd,minとtpd,maxで示されます。

(図11-3. で)システムクロックの最初の上昇端直後から始まるクロック周期を考察してください。このラッチはクロックがLowの時に閉じ、クロックがHighの時に同期ラッチ信号の斜線部分で示されるように通過(トランスペアレント)となります。この信号値はシステムクロックがLowになる時に保持(ラッチ)されます。それが続くクロックの上昇端でPINxnレジスタに取り込まれます。2つの矢印tpd,minとtpd,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11-4. 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(tpd)は1システムクロック周期です。

図11-3. 外部供給ピン値読み込み時の同期化

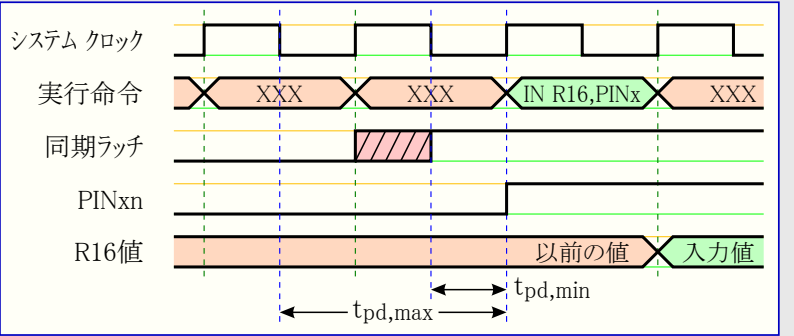
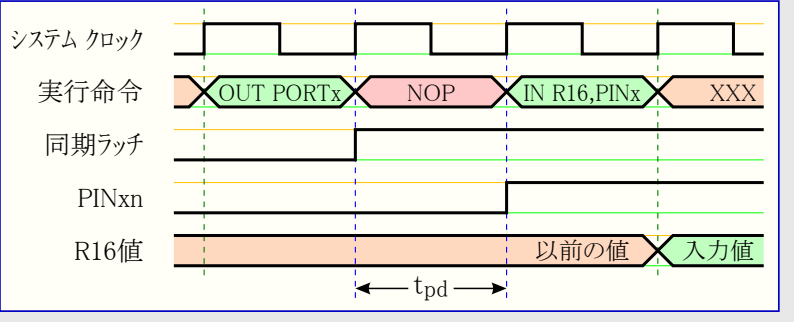


図11-4. プログラムで設定したピン値読み戻し時の同期化



次のコード例はポートB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);
__no_operation();
i = PINB;
~
/* */
/* プルアップとHigh値を設定 */
/* 入出力方向を設定 */
/* 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。

11.2.5. デジタル入力許可と休止形態

図11-2.で示されるようにデジタル入力信号はシュミットトリガの入力をGNDにクランプできます。この図で**SLEEP**と印された信号は入力信号のいくつかが開放のまま、またはVCC/2付近のアナログ信号電圧を持つ場合の高消費電力を避けるため、**パワーダウン動作**、**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す。

外部割り込みが許可されていない”上昇端、下降端または論理変化(両端)割り込み”として設定された非同期外部割り込みピンに論理1が存在すると、上で言及した休止形態から(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。

11.2.6. 未接続ピン

いくつかのピンが未使用にされる場合、それらのピンが定義されたレベルを持つ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。

11.3. 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11-5は単純化された図11-2.でのポートピン制御信号が交換機能により、どう重複できるかを示します。この重複信号は全てのポートピンに存在する訳ではありませんが、この図はAVRマイクロコントローラ系統の全ポートピンに適用できる一般的な記述として取り扱います。

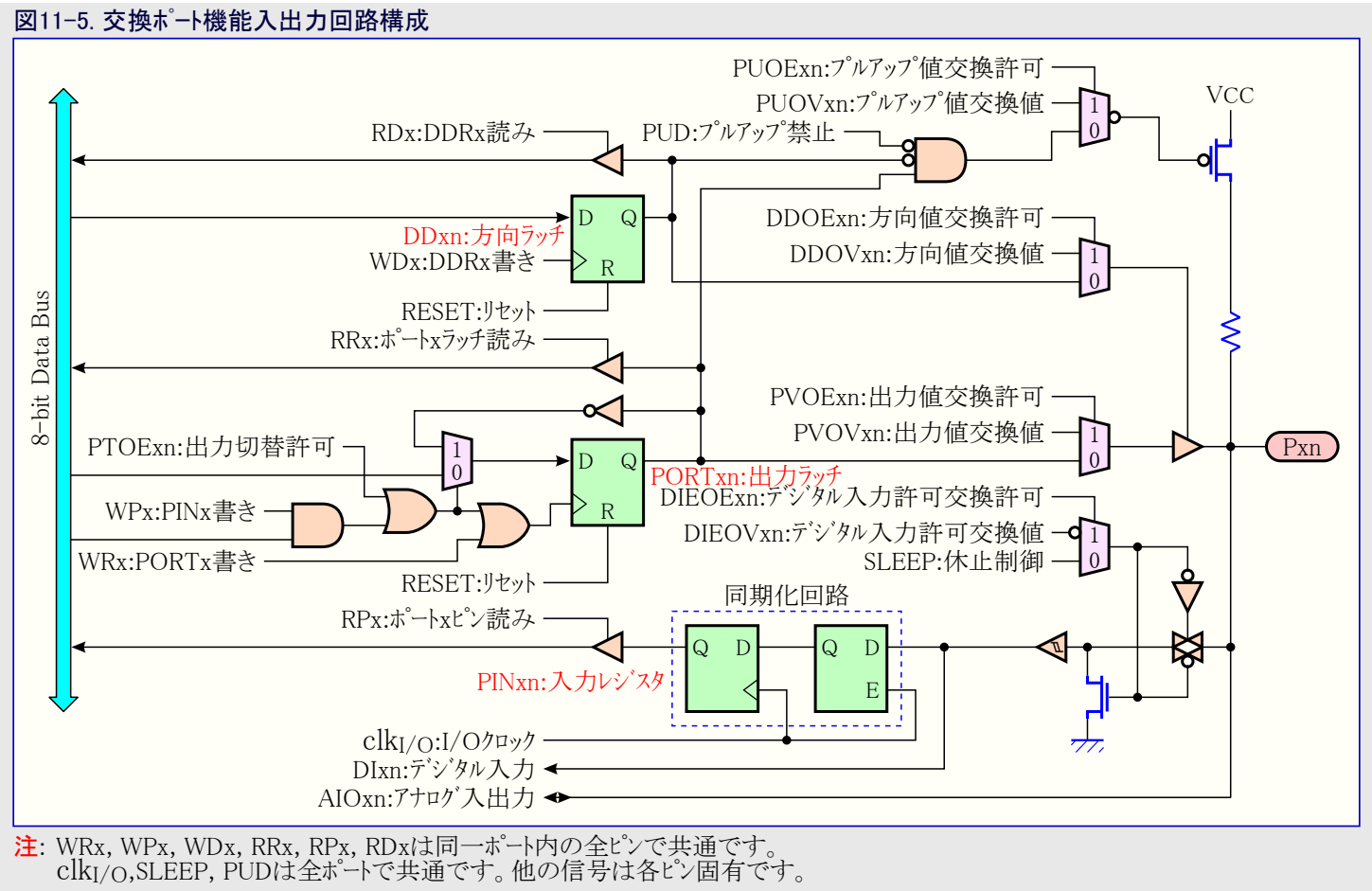


表11-2は重複(交換)信号の機能一覧を示します。図11-5.で示すピンとポートは次表で示されません。重複(交換)信号は交換機能を持つ機能部で内部的に生成されます。

表11-2. 交換機能用交換信号の一般定義		
信号略名	信号名	意味
PUOE	プルアップ値交換許可	1で、プルアップ許可はPUOV信号で制御され、0の場合、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。
PTOE	出力切替許可	PTOE=1時、PORTxnレジスタ値が反転します。
DIEOE	デジタル入力許可交換許可	1で、デジタル入力許可はDIEOV信号で制御され、0の場合、MCUの状態(活動動作、休止形態)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。

11.3.1. ポートBの交換機能

ポートBピンの交換機能は下で示されます。

表11-3. ポートBピンの交換機能

ポートピン	交換機能	ポートピン	交換機能
PB7	OC0A (タイマ/カウンタ0 比較A一致/PWM出力) OC1C (タイマ/カウンタ1 比較C一致/PWM出力) RTS (USART送信要求出力) PCINT7 (ピン変化割り込み7入力)	PB3	PDO (直列プログラミング データ出力) MISO (SPI 主側データ入力/従側データ出力) PCINT3 (ピン変化割り込み3入力)
PB6	OC1B (タイマ/カウンタ1 比較B一致/PWM出力) OC4B (タイマ/カウンタ4 比較B一致/PWM出力) ADC13 (A/D変換アナログ入力チャネル13) PCINT6 (ピン変化割り込み6入力)	PB2	PDI (直列プログラミング データ入力) MOSI (SPI 主側データ出力/従側データ入力) PCINT2 (ピン変化割り込み2入力)
PB5	OC1A (タイマ/カウンタ1 比較A一致/PWM出力) OC4B (タイマ/カウンタ4 比較B一致/PWM補完出力) ADC12 (A/D変換アナログ入力チャネル12) PCINT5 (ピン変化割り込み5入力)	PB1	SCK (SPI 直列クロック 主側出力/従側入力) PCINT1 (ピン変化割り込み1入力)
PB4	ADC11 (A/D変換アナログ入力チャネル11) PCINT4 (ピン変化割り込み4入力)	PB0	\overline{SS} (SPI 従装置選択入力) PCINT0 (ピン変化割り込み0入力)

交換ピンの設定は次のとおりです。

• OC0A/OC1C/ \overline{RTS} /PCINT7 – ポートB ビット7 : PB7

OC0A : タイマ/カウンタ0の比較A一致出力。PB7ピンはタイマ/カウンタ0の比較A一致用外部出力として扱えます。この機能を扱うため、このピンは出力として設定(**DDB7=1**)されなければなりません。このOC0A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。

\overline{RTS} : 強化されたUSARTによって使われるRTS(Request To Send)受信部流れ制御出力信号。

PCINT7 : ピン変化割り込み7入力。PB7ピンは外部割り込み元としても扱えます。

• OC1B/OC4B/ADC13/PCINT6 – 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。

OC4B : タイマ/カウンタ4の比較B一致出力。このピンはタイマ/カウンタ4単位部からの高速PWM信号生成に使うことができます。この機能を扱うには、このピンが出力として設定(**DDB6=1**)されなければなりません。

ADC13 : A/D変換チャネル13入力。

PCINT6 : ピン変化割り込み6入力。PB6ピンは外部割り込み元としても扱えます。

• OC1A/OC4B/ADC12/PCINT5 – 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。

OC4B : タイマ/カウンタ4の比較B一致出力。このピンはタイマ/カウンタ4単位部からのOC4B(PB6)補完用高速PWM信号生成に使うことができます。この機能を扱うには、このピンが出力として設定(**DDB5=1**)されなければなりません。

ADC12 : A/D変換チャネル12入力。

PCINT5 : ピン変化割り込み5入力。PB5ピンは外部割り込み元としても扱えます。

• ADC11/PCINT4 – ポートB ビット4 : PB4

ADC11 : A/D変換チャネル11入力。

PCINT4 : ピン変化割り込み4入力。PB4ピンは外部割り込み元としても扱えます。

• PDO/MISO/PCINT3 – ポートB ビット3 : PB3

PDO : 直列プログラミング データ出力。直列プログラミングの間、このピンはATmega16U4/32U4からのデータ出力線として使われます。

MISO : SPI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(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とき、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。

PCINT3 : ピン変化割り込み3入力。PB3ピンは外部割り込み元としても扱えます。

• PDI/MOSI/PCINT2 – ポートB ビット2 : PB2

PDI : [直列プログラミング](#) データ入力。直列プログラミングの間、このピンはATmega16U4/32U4へのデータ入力線として使われます。

MOSI : SPI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とき、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せん。

PCINT2 : ピン変化割り込み2入力。PB2ピンは外部割り込み元としても扱えます。

• SCK/PCINT1 – ポートB ビット1 : PB1

SCK : SPI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とき、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せん。

PCINT1 : ピン変化割り込み1入力。PB1ピンは外部割り込み元としても扱えます。

• \overline{SS} /PCINT0 – 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DDRBのDDB0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)が活性化(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にプルアップは未だPORTBのPORTB0によって制御できます。

PCINT0 : ピン変化割り込み0入力。PB0ピンは外部割り込み元としても扱えます。

表11-4.と表11-5.はポートBの交換機能を44頁の図11-5.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11-4. ポートB7~4の交換機能用交換信号

信号名	PB7/OC0A/OC1C/ \overline{RTS} /PCINT7	PB6/OC1B/OC4B/ ADC13/PCINT6	PB5/OC1A/ $\overline{OC4B}$ / ADC12/PCINT5	PB4/ADC11/PCINT4
PUOE	0	0	0	0
PUOV	0	0	0	0
DDOE	0	0	0	0
DDOV	0	0	0	0
PVOE	OC0A許可 OC1C許可	OC1B許可 OC4B許可	OC1A許可 OC4B許可	0
PVOV	OC0A/OC1C	OC1B/OC4B	OC1A/ $\overline{OC4B}$	0
PTOE	-	-	-	-
DIOE	PCIE0・PCINT7	PCIE0・PCINT6	PCIE0・PCINT5	PCIE0・PCINT4
DIOV	1	1	1	1
DI	PCINT7入力	PCINT6入力	PCINT5入力	PCINT4入力
AIO	-	ADC13入力	ADC12入力	ADC11入力

表11-5. ポートB3~0の交換機能用交換信号

信号名	PB3/PDO/MISO/PCINT3	PB2/PDI/MOSI/PCINT2	PB1/SCK/PCINT1	PB0/ \overline{SS} /PCINT0
PUOE	SPE・MSTR	SPE・ \overline{MSTR}	SPE・ \overline{MSTR}	SPE・ \overline{MSTR}
PUOV	PORTB3・ \overline{PUD}	PORTB2・ \overline{PUD}	PORTB1・ \overline{PUD}	PORTB0・ \overline{PUD}
DDOE	SPE・MSTR	SPE・MSTR	SPE・MSTR	SPE・MSTR
DDOV	0	0	0	0
PVOE	SPE・ \overline{MSTR}	SPE・MSTR	SPE・MSTR	0
PVOV	SPI従装置出力	SPI主装置出力	SCK出力	0
PTOE	-	-	-	-
DIOE	PCIE0・PCINT3	PCIE0・PCINT2	PCIE0・PCINT1	PCIE0・PCINT0
DIOV	1	1	1	1
DI	SPI主装置/PCINT3入力	SPI従装置/PCINT2入力	SCK/PCINT1入力	SPI \overline{SS} /PCINT0入力
AIO	-	-	-	-

11.3.2. ポートCの交換機能

ポートCの交換機能は下で示されます。

表11-6. ポートC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
PC7	ICP3 (タイマ/カウンタ3 捕獲起動入力) OC4A (タイマ/カウンタ4 比較A一致/PWM出力) CLKO (システムクロック出力)	PC6	OC3A (タイマ/カウンタ3 比較A一致/PWM出力) OC4A (タイマ/カウンタ4 比較A一致/PWM補完出力)

交換ピンの設定は次のとおりです。

• ICP3/OC4A/CLKO – ポートCビット7 : PC7

ICP3 : タイマ/カウンタ3が正しく構成設定されたなら、このピンは捕獲起動入力機能として扱うことができます。

OC4A : タイマ/カウンタ4の比較A一致出力。このピンはタイマ/カウンタ4単位部からの高速PWM信号生成に使うことができます。この機能を扱うには、このピンが出力として設定(DDC7=1)されなければなりません。

CLKO : 対応する(CKOUT)ピンスが許可(プログラム(0))されると、このピンは内部マイクロコントローラ動作周波数を出力します。システムクロック前置分周器が使われた場合、それはこの出力周波数に影響を及ぼします。

• OC3A/OC4A – ポートCビット6 : PC6

OC3A : タイマ/カウンタ3の比較A一致出力。このピンはタイマ/カウンタ3単位部からのPWM信号生成に使うことができます。この機能を扱うには、このピンが出力として設定(DDC6=1)されなければなりません。

OC4A : タイマ/カウンタ4の比較A一致出力。このピンはタイマ/カウンタ4部署からのOC4A(PC7)補完用高速PWM信号生成に使うことができます。この機能を扱うには、このピンが出力として設定(DDC6=1)されなければなりません。

下表はポートCの交換機能を44頁の図11-5.で示される交換信号に関連付けます。

表11-7. ポートC7,6の交換機能用交換信号

信号名	PC7/ICP3/OC4A/CLKO	PC6/OC3A/OC4A
PUOE	0	0
PUOV	0	0
DDOE	0	0
DDOV	0	0
PVOE	OC4A許可	OC3A許可 OC4A許可
PVOV	OC4A	OC3A/OC4A
PTOE	–	–
DIOE	0	0
DIOV	0	0
DI	ICP3入力	–
AIO	–	–

11.3.3. ポートDの交換機能

ポートDピンの交換機能は下で示されます。

表11-8. ポートD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
PD7	T0 (タイマ/カウンタ0 外部クロック入力) OC4D (タイマ/カウンタ4 比較D一致/PWM出力) ADC10 (A/D変換アナログ入力チャネル10)	PD3	INT3 (外部割り込み3 入力) TXD (USART 送信データ出力)
PD6	T1 (タイマ/カウンタ1 外部クロック入力) OC4D (タイマ/カウンタ4 比較D一致/PWM補完出力) ADC9 (A/D変換アナログ入力チャネル9)	PD2	INT2 (外部割り込み2 入力) RXD (USART 受信データ入力)
PD5	XCK (USART 外部クロック入出力) CTS (USART 送信許可入力)	PD1	INT1 (外部割り込み1 入力) SDA (2線直列バス データ入出力)
PD4	ICP1 (タイマ/カウンタ1 捕獲起動入力) ADC8 (A/D変換アナログ入力チャネル8)	PD0	INT0 (外部割り込み0 入力) SCL (2線直列バス クロック入出力) OC0B (タイマ/カウンタ0 比較B一致/PWM出力)

交換ピンの設定は次のとおりです。

- T0/OC4D/ADC10 – ポートD ビット7 : PD7

T0 : タイマ/カウンタ0の外部クロック入力ピンです。

OC4D : タイマ/カウンタ4の比較D一致出力。このピンはタイマ/カウンタ4単位部からの高速PWM信号生成に使うことができます。この機能を扱うには、このピンが出力として設定(DDD7=1)されなければなりません。

ADC10 : A/D変換チャネル10入力。

- T1/OC4D/ADC9 – 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。

OC4D : タイマ/カウンタ4の比較D一致出力。このピンはタイマ/カウンタ4単位部からのOC4D(PD7)補完用高速PWM信号生成に使うことができます。この機能を扱うには、このピンが出力として設定(DDD6=1)されなければなりません。

ADC9 : A/D変換チャネル9入力。

- XCK/CTS – ポートD ビット5 : PD5

XCK : USART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。XCKピンはUSARTが同期種別で動作する時だけ活動します(有効です)。

CTS : 強化されたUSARTによって使われるCTS(Clear To Send)送信部流れ制御入力信号。

- 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。

ADC8 : A/D変換チャネル8入力。

- INT3/TXD – 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。

TXD : 送信データ(USART用データ出力ピン)。USART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/RXD – 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。

RXD : 受信データ(USART用データ入力ピン)。USART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。USART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、スレーブ(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/OC0B – 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、スレーブ(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)。

OC0B: タイマ/カウンタ0の比較B一致出力。このピンはタイマ/カウンタ0単位部からのPWM信号生成に使うことができます。この機能を扱うには、このピンが出力として設定(DDD0=1)されなければなりません。

2つの下表はポートDの交換機能を44頁の図11-5.で示される交換信号に関連付けます。

表11-9. ポートD7~4の交換機能用交換信号

信号名	PD7/T0/OC4D/ADC10	PD6/T1/OC4D/ADC9	PD5/XCK/CTS	PD4/ICP1/ADC8
PUOE	0	0	0	0
PUOV	0	0	0	0
DDOE	0	0	XCK出力許可	0
DDOV	0	0	1	0
PVOE	OC4D許可	OC4D許可	XCK出力許可	0
PVOV	OC4D	OC4D	XCK出力	0
PTOE	-	-	-	-
DIOE	0	0	0	0
DIOV	0	0	0	0
DI	T0入力	T1入力	XCK/CTS入力	ICP1入力
AIO	ADC10入力	ADC9入力	-	ADC8入力

表11-10. ポートD3~0の交換機能用交換信号

信号名	PD3/INT3/TXD	PD2/INT2/RXD	PD1/INT1/SDA	PD0/INT0/SCL/OC0B
PUOE	TXEN	RXEN	TWEN	TWEN
PUOV	0	PORTD2・PUD	PORTD1・PUD	PORTD0・PUD
DDOE	TXEN	RXEN	TWEN	TWEN
DDOV	1	0	SDA_出力	SCL_出力
PVOE	TXEN	0	TWEN	TWEN OC0B許可
PVOV	TXD	0	0	OC0B
PTOE	-	-	-	-
DIOE	INT3許可	INT2許可	INT1許可	INT0許可
DIOV	1	1	1	1
DI	INT3入力	INT2入力/RXD	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。

11.3.4. ポートEの交換機能

ポートEピンの交換機能は下で示されます。

表11-11. 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	
PE6		INT6 (外部割り込み6 入力) AIN0 (アナログ比較器非反転入力)		PE2		$\overline{\text{HWB}}$ (フート ロータ強制実行入力)	

交換ピンの設定は次のとおりです。

- INT6/AIN0 – 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。
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。
- $\overline{\text{HWB}}$ – ポートE ビット2 : PE2
 $\overline{\text{HWB}}$: このピンが外部リセット パルス間にGNDに接続されると、リセット後のポート ロータ部実行を許します。このピンのHWB動作はHWBEヒューズが許可されている時だけ活性(有効)です。(リセットを除き、)通常動作中、このピンは汎用入出力として働きます。

下表はポートEの交換機能を44頁の図11-5.で示される交換信号に関連付けます。

表11-12. ポートE6.2の交換機能用交換信号		
信号名	PE6/INT6/AIN0	PE2/ $\overline{\text{HWB}}$
PUOE	0	0
PUOV	0	0
DDOE	0	0
DDOV	0	1
PVOE	0	0
PVOV	0	0
PTOE	–	–
DIEOE	INT6許可	0
DIEOV	1	0
DI	INT6入力	HWB入力
AIO	AIN0入力	–

11.3.5. ポートFの交換機能

ポートFピンには表11-13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。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11-13. 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ネル7) TDI (JTAG 検査データ入力)
PF6	ADC6 (A/D変換アナログ入力チャネル6) TDO (JTAG 検査データ出力)
PF5	ADC5 (A/D変換アナログ入力チャネル5) TMS (JTAG 検査種別選択入力)
PF4	ADC4 (A/D変換アナログ入力チャネル4) TCK (JTAG クロック入力)
PF1	ADC1 (A/D変換アナログ入力チャネル1)
PF0	ADC0 (A/D変換アナログ入力チャネル0)

交換ピンの設定は次のとおりです。

- ADC7/TDI – ポートF ピット7 : PF7

ADC7 : A/D変換チャネル7入力。

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 ピット6 : PF6

ADC6 : A/D変換チャネル6入力。

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他場合、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にされます。

- ADC5/TMS – ポートF ピット5 : PF5

ADC5 : A/D変換チャネル5入力。

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 ピット4 : PF4

ADC4 : A/D変換チャネル4入力。

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。

- ADC1 – ポートF ピット1 : PF1

ADC1 : A/D変換チャネル1入力。

- ADC0 – ポートF ピット0 : PF0

ADC0 : A/D変換チャネル0入力。

表11-14.と表11-15.はポートFの交換機能を44頁の図11-5.で示される交換信号に関連付けます。

表11-14. 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
PTOE	-	-	-	-
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11-15. ポートF1,0の交換機能用交換信号		
信号名	PF1/ADC1	PF0/ADC0
PUOE	0	0
PUOV	0	0
DDOE	0	0
DDOV	0	0
PVOE	0	0
PVOV	0	0
PTOE	-	-
DIOE	0	0
DIOV	0	0
DI	-	-
AIO	ADC1入力	ADC0入力

11.4. I/Oポート用レジスタ

11.4.1. MCUCR – MCU制御レジスタ (MCU Control Register)

ビット	7	6	5	4	3	2	1	0	
\$35 (\$55)	JTD	–	–	PUD	–	–	IVSEL	IVCE	MCUCR
Read/Write	R/W	R	R	R/W	R	R	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

● ビット4 – PUD : プルアップ禁止 (Pull-up Disable)

このビットが1を書かれると、例えば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。

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

ビット	7	6	5	4	3	2	1	0	
\$05 (\$25)	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	

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

ビット	7	6	5	4	3	2	1	0	
\$04 (\$24)	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	

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

ビット	7	6	5	4	3	2	1	0	
\$03 (\$23)	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	PINB
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	

11.4.5. PORTC – ポートC出力レジスタ (Port C Data Register)

ビット	7	6	5	4	3	2	1	0	
\$08 (\$28)	PORTC7	PORTC6	–	–	–	–	–	–	PORTC
Read/Write	R/W	R/W	R	R	R	R	R	R	
初期値	0	0	0	0	0	0	0	0	

11.4.6. DDRC – ポートC方向レジスタ (Port C Data Direction Register)

ビット	7	6	5	4	3	2	1	0	
\$07 (\$27)	DDC7	DDC6	–	–	–	–	–	–	DDRC
Read/Write	R/W	R/W	R	R	R	R	R	R	
初期値	0	0	0	0	0	0	0	0	

11.4.7. PINC – ポートC入力レジスタ (Port C Input Address)

ビット	7	6	5	4	3	2	1	0	
\$06 (\$26)	PINC7	PINC6	–	–	–	–	–	–	PINC
Read/Write	R/W	R/W	R	R	R	R	R	R	
初期値	不定	不定	0	0	0	0	0	0	

11.4.8. PORTD – ポートD出力レジスタ (Port D Data Register)

ビット	7	6	5	4	3	2	1	0	
\$0B (\$2B)	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	

11.4.9. DDRD – ポートD方向レジスタ (Port D Data Direction Register)

ビット	7	6	5	4	3	2	1	0	
\$0A (\$2A)	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	

11.4.10. PIND – ポートD入力レジスタ (Port D Input Address)

ビット	7	6	5	4	3	2	1	0	
\$09 (\$29)	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	PIND
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	

11.4.11. PORTE – ポートE出力レジスタ (Port E Data Register)

ビット	7	6	5	4	3	2	1	0	
\$0E (\$2E)	–	PORTE6	–	–	–	PORTE2	–	–	PORTE
Read/Write	R	R/W	R	R	R	R/W	R	R	
初期値	0	0	0	0	0	0	0	0	

11.4.12. DDRE – ポートE方向レジスタ (Port E Data Direction Register)

ビット	7	6	5	4	3	2	1	0	
\$0D (\$2D)	–	DDE6	–	–	–	DDE2	–	–	DDRE
Read/Write	R	R/W	R	R	R	R/W	R	R	
初期値	0	0	0	0	0	0	0	0	

11.4.13. PINE – ポートE入力レジスタ (Port E Input Address)

ビット	7	6	5	4	3	2	1	0	
\$0C (\$2C)	–	PINE6	–	–	–	PINE2	–	–	PINE
Read/Write	R/W	R/W	R	R	R	R/W	R	R	
初期値	0	不定	0	0	0	不定	0	0	

11.4.14. PORTF – ポートF出力レジスタ (Port F Data Register)

ビット	7	6	5	4	3	2	1	0	
\$11 (\$31)	PORTF7	PORTF6	PORTF5	PORTF4	–	–	PORTF1	PORTF0	PORTF
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

11.4.15. DDRF – ポートF方向レジスタ (Port F Data Direction Register)

ビット	7	6	5	4	3	2	1	0	
\$10 (\$30)	DDF7	DDF6	DDF5	DDF4	–	–	DDF1	DDF0	DDRF
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

11.4.16. PINF – ポートF入力レジスタ (Port F Input Address)

ビット	7	6	5	4	3	2	1	0	
\$0F (\$2F)	PINF7	PINF6	PINF5	PINF4	–	–	PINF1	PINF0	PINF
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
初期値	不定	不定	不定	不定	0	0	不定	不定	

12. タイマ/カウンタ0、タイマ/カウンタ1、タイマ/カウンタ3の前置分周器

タイマ/カウンタ0、1、3は同じ前置分周器部を共用しますが、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。Tn (n=0,1,3)は一般名として使われます。

12.1. 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。

12.2. 前置分周器リセット

この前置分周器は自由走行で(換言するとタイマ/カウンタのクロック選択論理回路と無関係に動作する)、Tnによって共用されます。前置分周器はタイマ/カウンタのクロック選択によって影響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~N+1 システム クロック周期になり得ます。

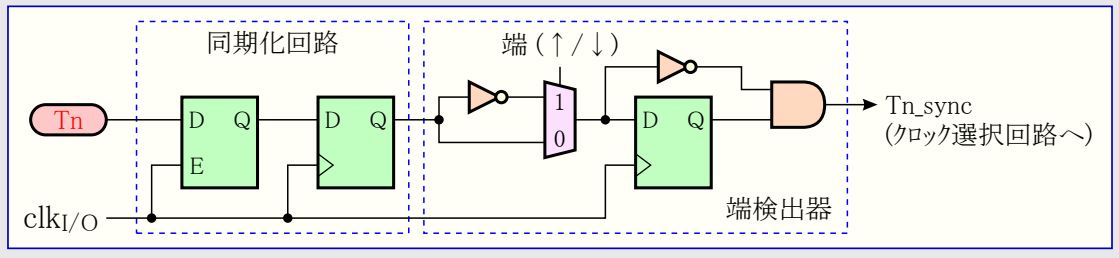
プログラム実行にタイマ/カウンタを同期することに対して前置分周器リセットを使うことが可能です。しかし、同じ前置分周器を共用する他のタイマ/カウンタも前置分周を使う場合、注意が必要とされなければなりません。前置分周器リセットはそれが接続される全タイマ/カウンタについての前置分周器周期に影響を及ぼします。

12.3. 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(clk_{Tn})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12-1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($clk_{I/O}$)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_{Tn} パルスを生成します。

図12-1. Tnピンの採取等価構成図

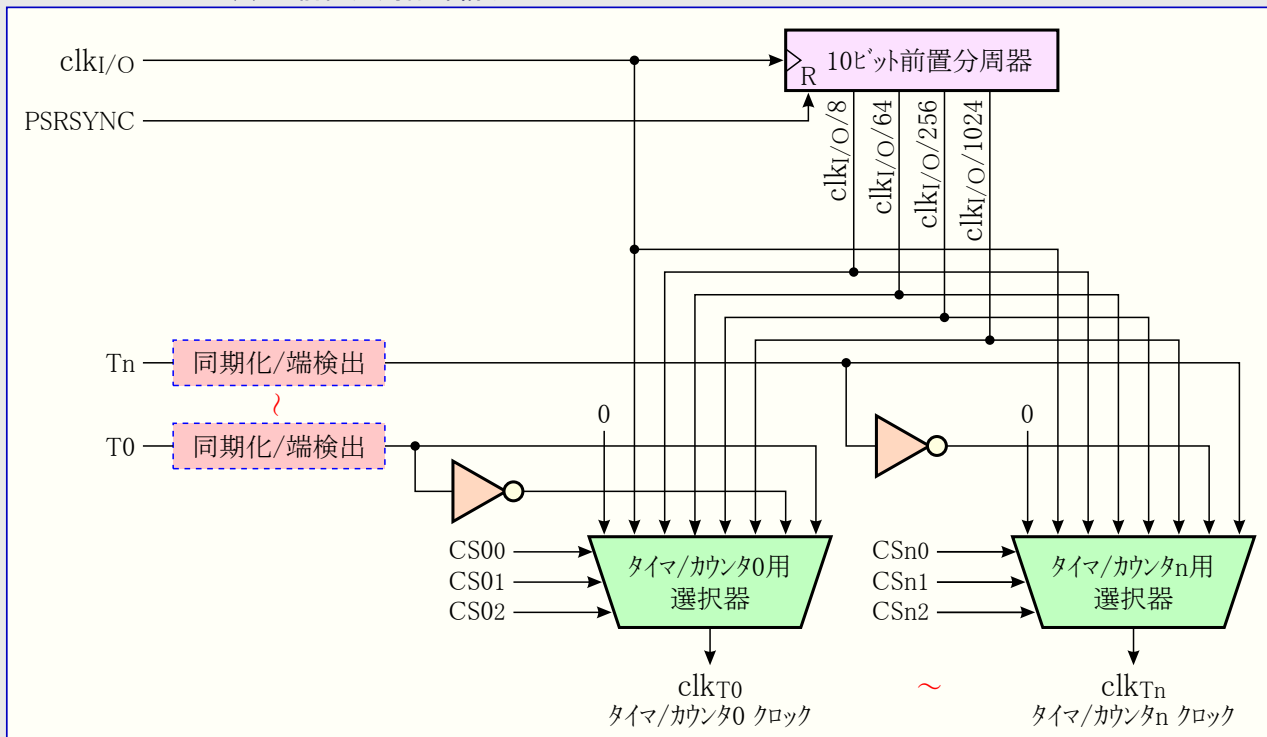


同期化と端検出器論理回路はTnピンへ印加された端からカウンタが更新されるまでに2.5~3.5システム クロック周期の遅延をもたらします。クロック入力の許可と禁止はTnが最低1システム クロック周期に対して安定してしまっている時に行われなければならない、さもなければ不正なタイマ/カウンタ クロック パルスが生成される危険があります。

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(ナイキストの標本化定理)。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12-2. タイマ/カウンタ0,1,3の前置分周器部構成図



注: 入力ピンの同期化/端(エッジ)検出論理回路は図12-1.で示されます。

12.4. 同期系タイマ/カウンタ前置分周器用レジスタ

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

ビット	7	6	5	4	3	2	1	0	
\$23 (\$43)	TSM	–	–	–	–	–	–	PSRSYNC	GTCCR
Read/Write	R/W	R	R	R	R	R	R	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ビットへの1書き込みはタイマ/カウンタ同期化動作を活性(有効)にします。この動作でPSRSYNC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書かれると、PSRSYNC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 – PSRSYNC : 同期系タイマ/カウンタ前置分周器リセット (Prescaler Reset Timer/Counter 1,0)

このビットが1のとき、タイマ/カウンタ0、タイマ/カウンタ1、タイマ/カウンタ3の前置分周器はリセットします。通常、TSMビットが設定(1)されている場合を除き、このビットはハードウェアによって直ちに解除(0)されます。タイマ/カウンタ0、タイマ/カウンタ1、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全てのタイマ/カウンタに影響を及ぼすことに注意してください。

13. 8ビット タイマ/カウンタ0 (PWM付き)

タイマ/カウンタ0は2つの独立した比較出力部とPWM支援付きの汎用8ビット タイマ/カウンタ部です。それは正確なプログラム実行タイミング(事象管理)、波形生成を許します。主な特徴は次に示されます。

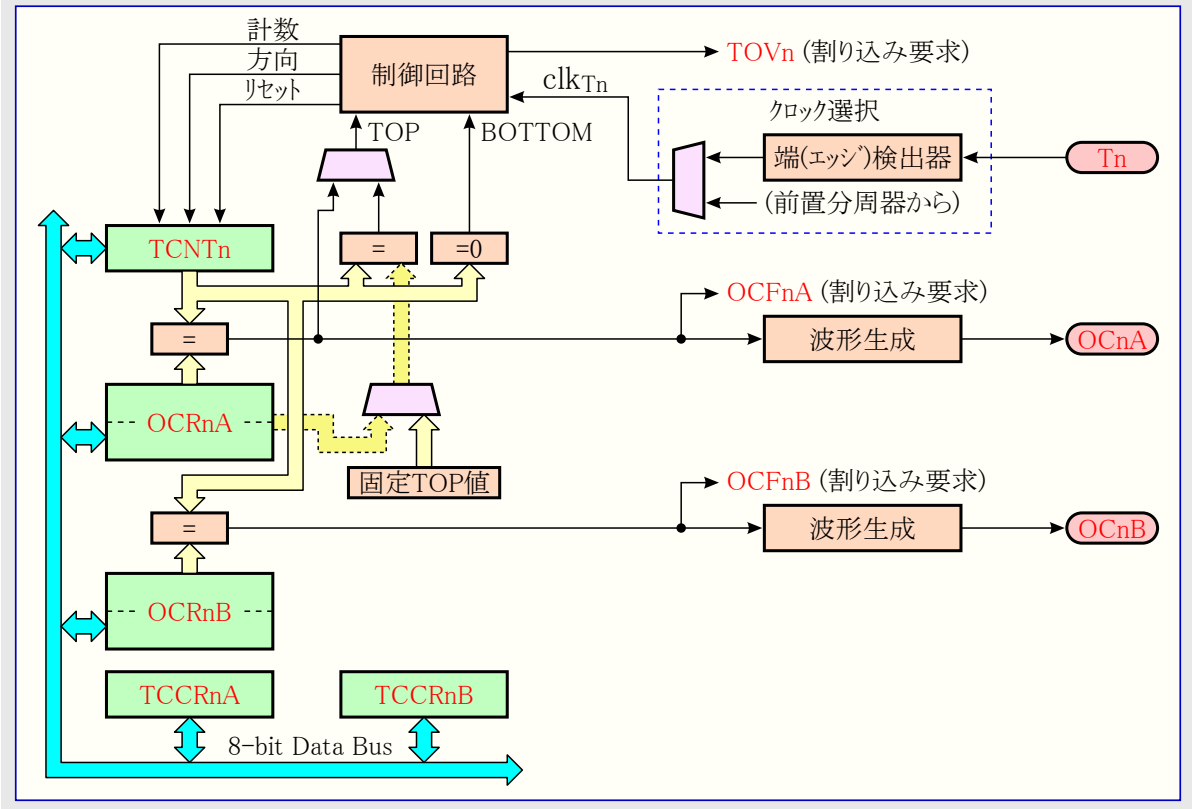
- 2つの独立した比較出力部
- 2重緩衝の比較レジスタ
- 比較一致でのタイマ/カウンタ解除(自動再設定)
- 不具合なしで正しい位相のパルス幅変調器 (PWM)
- 可変PWM周期
- 周波数発生器
- 3つの独立した割り込み (TOV0,OCF0A,OCF0B)

13.1. 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13-1.で示されます。I/Oピン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65頁の「8ビット タイマ/カウンタ0用レジスタ」で一覧されます。

28頁の「PRR0 - 電力削減レジスタ」のPRTIM0ビットはタイマ/カウンタ0部を許可するために0を書かれなければなりません(訳注:本行追加)。

図13-1. 8ビット タイマ/カウンタ構成図



13.1.1. 関係レジスタ

タイマ/カウンタ(TCNT0)と比較レジスタ(OCR0AとOCR0B)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0割り込み要求レジスタ(TIFR0)で全て見えます。全ての割り込みはタイマ/カウンタ0割り込み許可レジスタ(TIMSK0)で個別に遮蔽(禁止)されます。TIFR0とTIMSK0はこの図で示されません。

このタイマ/カウンタは内部的、前置分周器経由、またはT0ピンの外部クロック元によってクロック駆動されます。クロック選択論理部はタイマ/カウンタが値を増加(または減少)するため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(OCR0AとOCR0B)はタイマ/カウンタ値と常に比較されます。この比較結果は比較出力(OC0AとOC0B)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。詳細については59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(OCF0AとOCF0B)も設定(1)します。

13.1.2. 定義

本章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。小文字のxは比較出力部のチャンネル名を表し、この場合はAまたはBです。然しながらプログラムでレジスタまたはビット定義に使用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のカウント値のアクセスに対してのTCNT0のように)。

表の定義は本文書を通して広範囲に渡って使われます。

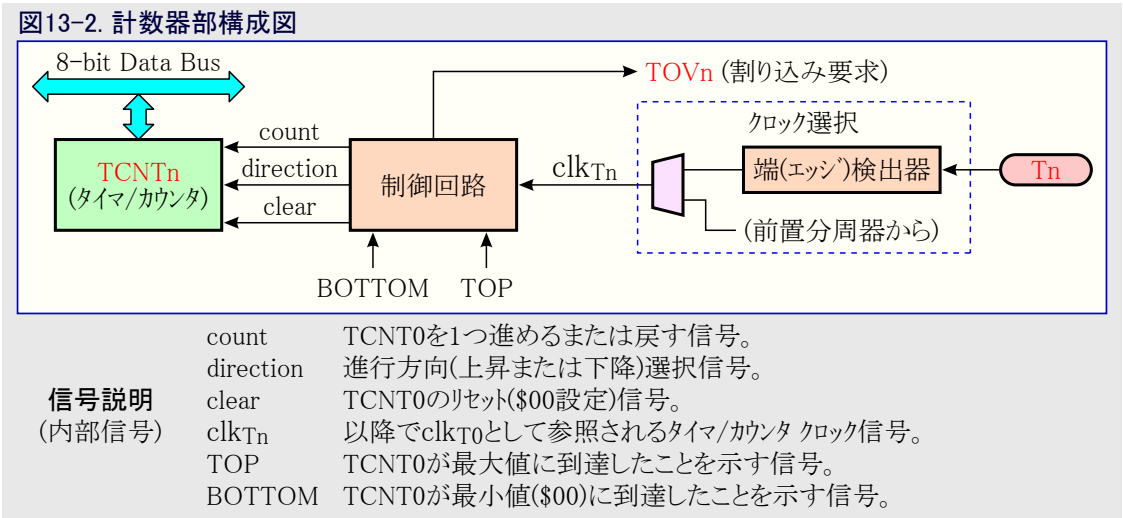
用語	意味
BOTTOM	タイマ/カウンタが\$00に到達した時。
MAX	タイマ/カウンタが\$FF(255)に到達した時。
TOP	タイマ/カウンタが指定された固定値(\$FF)またはOCR0A値に到達した時。この指定(TOP)値は動作種別に依存します。

13.2. 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(TCCR0B)に配置されたクロック選択(CS02~0)ビットによって制御されるクロック選択論理回路で選ばれます。クロック元と前置分周器の詳細については55頁の「タイマ/カウンタ0、タイマ/カウンタ1、タイマ/カウンタ3の前置分周器」をご覧ください。

13.3. 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13-2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(clkT0)で解除(\$00)、増加(+1)、または減少(-1)されます。clkT0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)と、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(clkT0)が存在するしないに拘らず、CPUによってアクセスできます。CPU書き込みは全てのカウンタ解除や計数動作を無視します(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0制御レジスタA(TCCR0A)に配置された波形生成種別(WGM01,0)ビットとタイマ/カウンタ0制御レジスタB(TCCR0B)に配置された波形生成種別(WGM02)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OC0A/OC0B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61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WGM02~0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。

13.4. 比較出力部

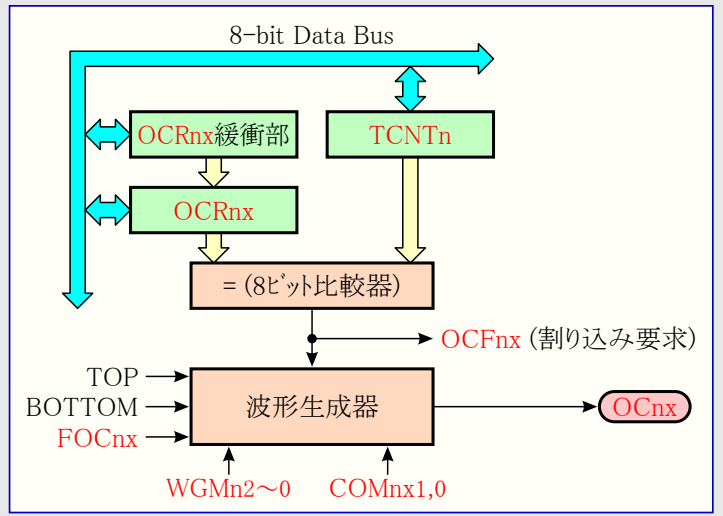
この8ビット比較器はTCNT0と比較レジスタ(OCR0AとOCR0B)を継続的に比較します。TCNT0がOCR0AまたはOCR0Bと等しければ比較器は一致を指示します。この一致は次のタイマ/カウンタクロック周期で比較割り込み要求フラグ(OCF0AまたはOCF0B)を設定(1)します。対応する割り込みが許可(I=1, OCIE0AまたはOCIE0B=1)されているならば、その比較割り込み要求フラグは比較割り込みを発生します。比較割り込み要求フラグは割り込みが実行されると自動的に解除(0)されます。代わりにこのフラグはこのI/Oビット位置に論理1を書くことによってソフトウェアでも解除(0)できます。波形生成器は波形生成種別(WGM02~0)ビットと比較出力選択(COM0x1,0)ビットによって設定された動作種別に従った出力を生成するのにこの一致信号を使います。MAXとBOTTOM信号は動作種別(61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13-3. は比較出力部の構成図を示します。

OCR0x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OCR0xレジスタの更新を同期化します。この同期化は奇数長、非対称PWMパルスの発生を防ぎ、それによって不具合なしの出力を作成します。

OCR0x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OCR0x緩衝部をアクセスし、禁止されるとOCR0xレジスタを直接アクセスします。

図13-3. 比較出力部構成図



(訳注) ここでは比較nxレジスタ全体をOCR0x、OCR0xを構成する緩衝部部分をOCR0x緩衝部、実際の比較に使われるレジスタ本体部分をOCR0x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。

13.4.1. 強制比較出力

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

13.4.2. TCNT0書き込みによる比較一致妨害

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

13.4.3. 比較一致部の使用

どの動作種別でのTCNT0書き込みでも1タイマ/カウンタクロック周期間、全ての比較一致を妨げるため、タイマ/カウンタが走行中であるか否かに拘らず、比較出力部を使う場合、TCNT0を変更する時に危険を伴います。TCNT0に書かれた値がOCR0x値と同じ場合に比較一致は失われ(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。

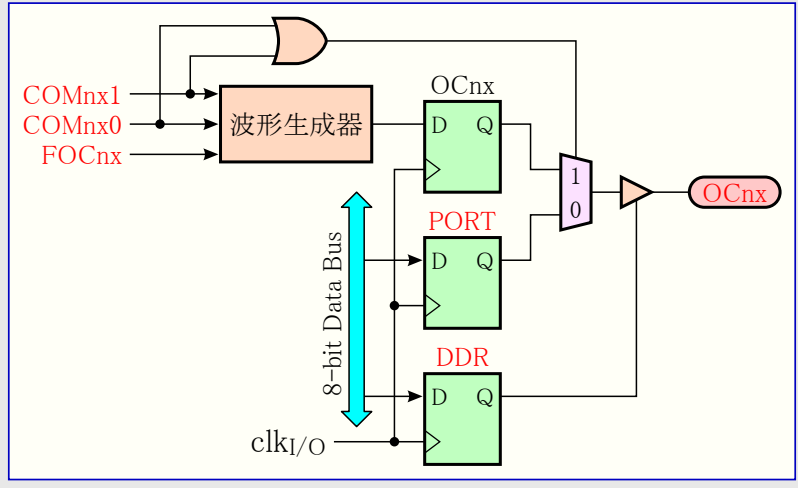
OC0xの初期設定はポートピンに対するポート方向レジスタを出力に設定する前に行われるべきです。OC0x値を設定する一番簡単な方法は標準動作で強制変更(FOC0x)スローブビットを使うことです。波形生成動作種別間を変更する時でも、OC0x(内部)レジスタはその値を保ちます。

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

13.5. 比較一致出力部

比較出力選択($COM0x1,0$)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($OC0x$)状態の定義に $COM0x1,0$ ビットを使います。また $COM0x1,0$ ビットは $OC0x$ ピン出力元を制御します。図13-4は $COM0x1,0$ ビット設定によって影響を及ぼ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。 $COM0x1,0$ 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。 $OC0x$ の状態を参照するとき、その参照は $OC0x$ ピンでなく内部 $OC0x$ レジスタに対してです。システムリセットが起ると、 $OC0x$ レジスタは'0'にリセットされます。

図13-4. 比較一致出力回路図



$COM0x1,0$ ビットのどちらかが設定(1)されると、標準I/Oポート機能は波形生成器からの比較出力($OC0x$)によって無効にされます。けれども $OC0x$ ピンの方向(入出力)はポートピンに対するポート方向レジスタ(DDR)によって未だ制御されます。 $OC0x$ ピンに対するポート方向レジスタのビット(DDR_OC0x)は $OC0x$ 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 $OC0x$ 状態の初期化を許します。いくつかの $COM0x1,0$ ビット設定が或る種の動作種別に対して予約されることに注意してください。65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。

13.5.1. 比較一致出力選択と波形生成

波形生成器は標準、CTC、PWM動作で $COM0x1,0$ ビットを違うふうに使います。全ての動作種別に対して $COM0x1,0=00$ 設定は次の比較一致で実行すべき $OC0x$ レジスタの動きがないことを波形生成器へ告げます。非PWM動作での比較出力動作については65頁の表13-2と表13-5を参照してください。高速PWM動作については65頁の表13-3と表13-6、位相基準PWMについては65頁の表13-4と表13-7を参照してください。

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

13.6. 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(WGM02~0)ビットと比較出力選択(COM0x1,0)ビットの組み合わせによって定義されます。比較出力選択ビットは計数順序(動作)に影響を及ぼしませんが、一方波形生成種別ビットは影響を及ぼします。COM0x1,0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COM0x1,0ビットは比較一致で出力が解除(0)、設定(1)、1/0交互のどれにされるべきかを制御します(60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64頁の「タイマ/カウンタ0のタイミング」を参照してください。

13.6.1. 標準動作

最も単純な動作種別が標準動作(WGM02~0=000)です。この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。

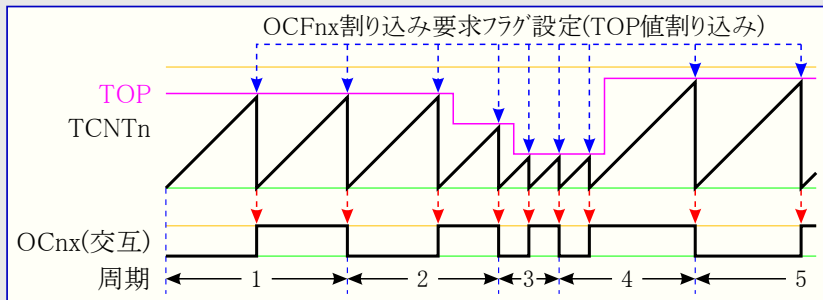
比較出力部は与えられた或る時間に割り込みを生成するのに使えます。標準動作で波形を生成するのに比較出力を使うのは、それが大変多くのCPU時間を占有するため推奨されません。

13.6.2. 比較一致タイマ/カウンタ解除(CTC)動作

比較一致タイマ/カウンタ解除(CTC)動作(WGM02~0=010)ではOCR0A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(TCNT0)値がOCR0Aと一致する時にカウンタは\$00に解除されます。OCR0A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13-5.で示されます。カウンタ(TCNT0)値はTCNT0とOCR0A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。

図13-5. CTC動作タイミング



注: COMnx1,0=01

OCF0Aフラグを使うことにより、タイマ/カウンタ値がTOP値に達する時毎に割り込みが生成できます。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。OCR0A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。その後も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、OC0A出力は比較出力選択(COM0A1,0)ビットを交互動作(=01)に設定することによって各比較一致での論理レベル交互切り替えに設定できます。OC0A値はそのピンに対するデータ方向が出力(DDR_OC0A=1)に設定されない限りポートピンで見えないでしょう。生成された波形はOCR0Aが0(\$00)に設定される時に $f_{OC0A} = f_{clk_I/O} / 2$ の最大周波数を得ます。生成波形周波数は次式によって定義されます。

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

変数Nは前置分周数(1,8,64,256,1024)を表します。

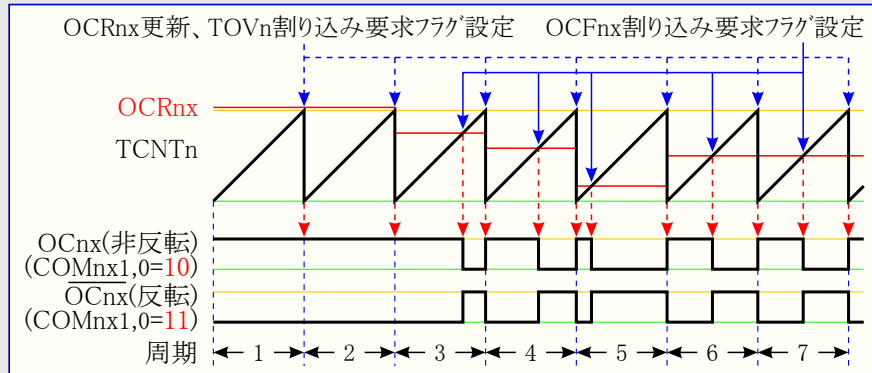
標準動作と同じように、タイマ/カウンタ溢れ(TOV0)フラグはカウンタがMAXから\$00へ計数するのと同じタイマ/カウンタクロック周期で設定(1)されます。

13.6.3. 高速PWM動作

高速パルス幅変調(PWM)動作(WGM02~0=011または111)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。TOPはWGM02~0=011時に\$FF、WGM02~0=111時にOCR0Aとして定義されます。非反転比較出力動作(COM0x1,0=10)での比較出力(OC0x)はTCNT0とOCR0x間の比較一致で解除(0)され、BOTTOMで設定(1)されます。反転出力動作(COM0x1,0=11)の出力は比較一致で設定(1)され、BOTTOMで解除(0)されます。この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TOP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13-6.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。細い赤線はOCR0x値を示し、TCNT0値との交点(接点)がTCNT0とOCR0x間の比較一致を示します(訳注:図補正に伴い本行若干変更)。比較割り込み要求フラグ(OCF0x)はOCR0x=TOP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)。

図13-6. 高速PWM動作タイミング



タイマ/カウンタ溢れ(TOV0)フラグはカウンタがTOPに到達する時毎に設定(1)されます。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OC0xピンでのPWM波形の生成を許します。COM0x1,0ビットを'10'に設定することは非反転PWM出力を作成し、反転PWM出力はCOM0x1,0を'11'に設定することで生成できます。WGM02ビットが設定(1)ならば、COM0A1,0ビットの'01'設定は比較一致での交互反転をOC0Aピンに許します。この任意選択はOC0Bピンに対して利用できません(65頁の表13-3と表13-6をご覧ください)。実際のOC0x値はポートピンに対するデータ方向(DDR_OC0x)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TCNT0とOCR0x間の比較一致で、OC0x(内部)レジスタを設定(1)(または解除(0))と、カウンタが解除(\$00,TOPからBOTTOMへ変更)されるタイマ/カウンタクロック周期でOC0xレジスタを解除(0)または設定(1)することによって生成されます。

PWM出力周波数は次式によって計算できます。

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

変数Nは前置分周数(1,8,64,256,1024)を表します。

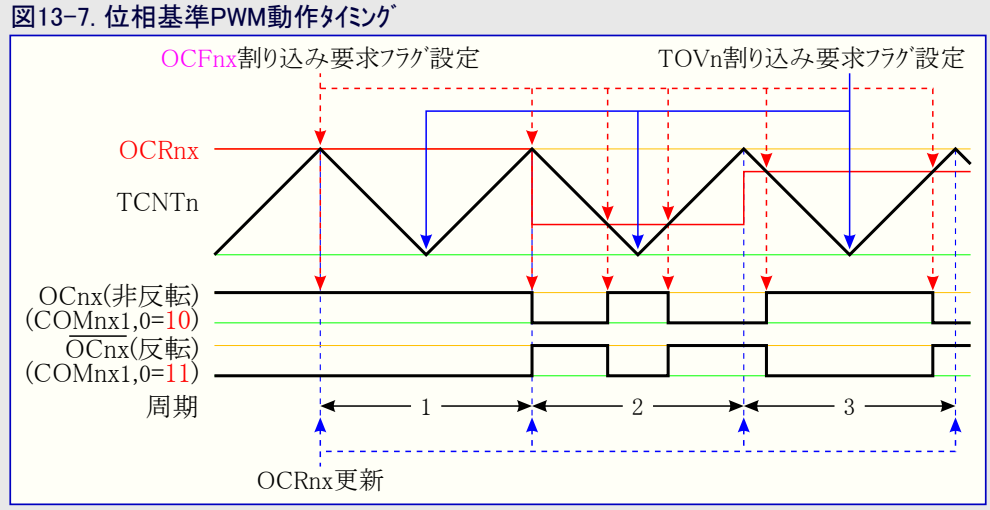
OCR0xの両端値は高速PWM動作でPWM波形出力を生成する時の特別な場合に当たります。OCR0xがBOTTOM(\$00)と等しく設定されると、出力はTOP+1 タイマ/カウンタクロック周期毎の狭いスパイク(パルス)になるでしょう。OCR0xがTOPに等しく設定されると、(COM0x1,0ビットによって設定される出力極性に依存して)定常的なLowまたはHigh出力に終わるでしょう。

(訳補:WGM02~0=111の場合については、)高速PWM動作での(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OC0A設定(COM0A1,0=01)によって達成できます。生成された波形はOCR0Aが0(\$00)に設定される時に $f_{OC0x} = f_{clk_I/O} / 2$ 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OC0A交互出力(COM0A1,0=01)と同じです。

13.6.4. 位相基準PWM動作

位相基準パルス幅変調(PWM)動作(WGM02~0=001または101)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TOPへそして次にTOPからBOTTOMへを繰り返し計数します。TOPはWGM02~0=001時に\$FF、WGM02~0=101時にOCR0Aとして定義されます。非反転比較出力動作(COM0x1,0=10)での比較出力(OC0x)は上昇計数中のTCNT0とOCR0xの比較一致で解除(0)され、下降計数中の比較一致で設定(1)されます。反転出力動作(COM0x1,0=11)での動作は逆にされます。両傾斜(三角波)動作は単一傾斜(鋸波)動作よりも低い最大動作周波数になります。けれども両傾斜(三角波)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動作でのカウンタはカウンタ値がTOPと一致するまで増加されます。カウンタはTOPに達すると計数方向を変更します。このTCNT0値は1タイマ/カウンタクロック周期間TOPと等しくなります。位相基準PWM動作のタイミング図は図13-7.で示されます。TCNT0値はタイミング図で両傾斜動作(三角波)を表す折れ線グラフとして示されます。この図は非反転と反転のPWM出力を含みます。細い赤線はOCR0x値を示し、TCNT0値との交点(接点)がTCNT0とOCR0x間の比較一致を示します(訳注:図補正に伴い本行若干変更)。



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ために使えます。

位相基準PWM動作での比較部はOC0xピンでのPWM波形の生成を許します。COM0x1,0ビットを'10'に設定することは非反転PWM出力を作成し、反転PWM出力はCOM0x1,0ビットを'11'に設定することで生成できます。WGM02ビットが設定(1)なら、COM0A1,0ビットの'01'設定は比較一致での交互反転をOC0Aピンに許します。この任意選択はOC0Bピンに対して利用できません(65頁の表13-4.と表13-7.をご覧ください)。実際のOC0x値はそのポートピンに対するデータ方向(DDR_OC0x)が出力として設定される場合だけ見えてでしょう。PWM波形はカウンタが増加する時のTCNT0とOCR0x間の比較一致でOC0x(内部)レジスタを設定(1)(または解除(0))と、カウンタが減少する時のTCNT0とOCR0x間の比較一致でOC0xレジスタを解除(0)(または設定(1))によって生成されます。位相基準PWMを使う時の出力に対するPWM周波数は次式によって計算できます。

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

変数Nは前置分周数(1,8,64,256,1024)を表します。

OCR0xの両端値は位相基準PWM動作でPWM波形出力を生成する時の特別な場合にあたります。非反転PWM動作ではOCR0xがBOTTOM(\$00)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。

図13-7.の第2周期のその出発点に於いて、例えば比較一致がないとしても、OCnx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13-7.のようにOCR0xはTOPからその値を変更します。OCR0x値がTOPのとき、OCnx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)TOP(位置)でのOCnx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OCR0x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OCnxの変更を逃します。(訳補: 従って上記同様、TOP位置で(直前がHならば)H→L遷移が生じます。)

13.7. 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_{Tn})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13-8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13-8. 前置分周なし(1/1)のタイマ/カウンタ タイミング

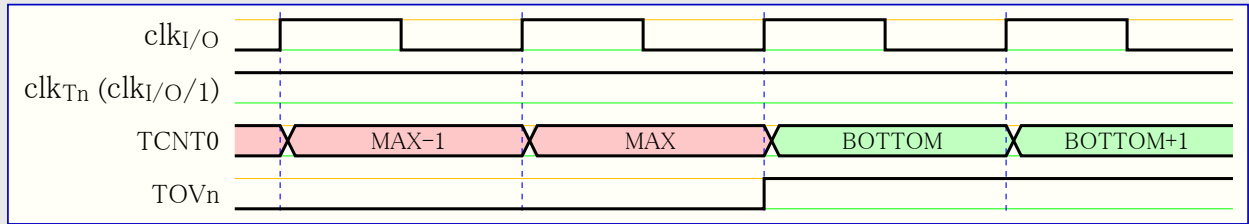


図13-9. は同じタイミング データを示しますが、前置分周器が許可されています。

図13-9. 前置分周器($f_{\text{clk}_{I/O}}/8$)のタイマ/カウンタ タイミング

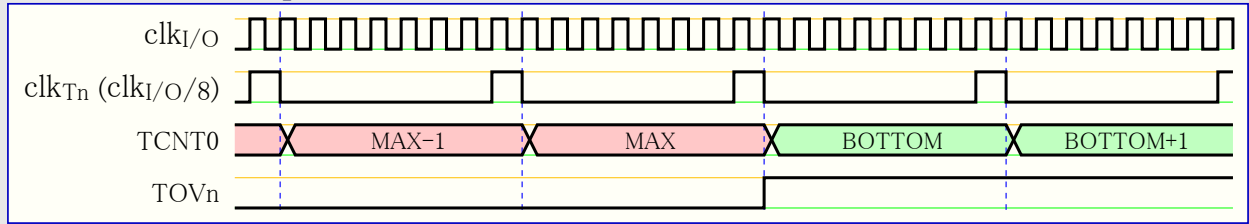


図13-10. はCTC動作とOCR0AがTOPのPWM動作を除く全動作種別でのOCF0Aと全動作種別でのOCF0Bの設定を示します。

図13-10. 前置分周器($f_{\text{clk}_{I/O}}/8$)のタイマ/カウンタ、OCF0x設定 タイミング

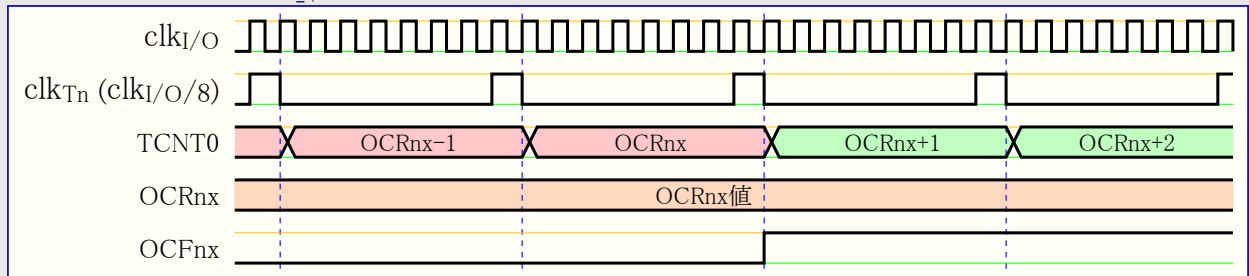
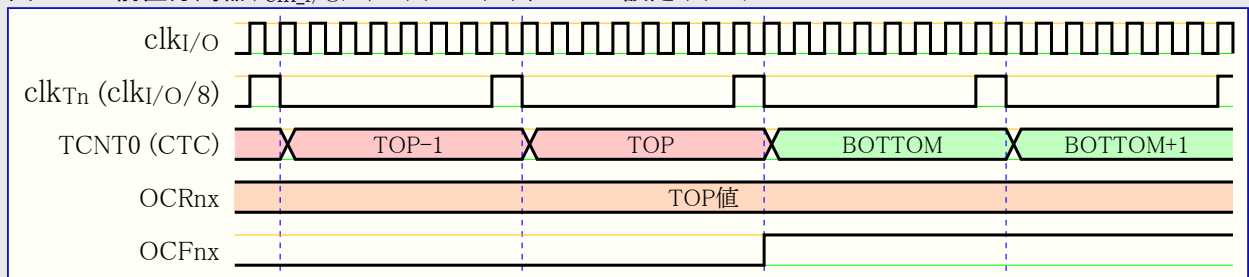


図13-11. はOCR0AがTOPの高速PWM動作と、CTC動作でのTCNT0の解除とOCF0Aの設定を示します。

図13-11. 前置分周器($f_{\text{clk}_{I/O}}/8$)のタイマ/カウンタ、OCF0A設定 タイミング



13.8. 8ビット タイマ/カウンタ用レジスタ

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

ビット	7	6	5	4	3	2	1	0	
\$24 (\$44)	COM0A1	COM0A0	COM0B1	COM0B0	–	–	WGM01	WGM00	TCCR0A
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

●ビット7,6 – COM0A1,0 : 比較A出力選択 (Compare Match A Output Mode bit 1 and 0)

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

OC0Aがピンに接続されるとき、COM0A1,0ビットの機能はWGM02~0ビット設定に依存します。

表13-2.はWGM02~0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COM0A1,0ビット機能を示します。

表13-3.はWGM02~0ビットが高速PWM動作に設定される時のCOM0A1,0ビットの機能を示します。

表13-4.はWGM02~0ビットが位相基準PWM動作に設定される時のCOM0A1,0ビットの機能を示します。

表13-2. 非PWM動作比較A出力選択

COM0A1	COM0A0	意味
0	0	標準ポート動作 (OC0A切断)
0	1	比較一致でOC0Aピントグル(交互)出力
1	0	比較一致でOC0Aピン Lowレベル出力
1	1	比較一致でOC0Aピン Highレベル出力

表13-4. 位相基準PWM動作比較A出力選択 (共通注意参照)

COM0A1	COM0A0	意味
0	0	標準ポート動作 (OC0A切断)
0	1	WGM02=0 : 標準ポート動作 (OC0A切断) WGM02=1 : 比較一致でOC0Aピントグル(交互)出力
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OC0A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OC0Aピンへ出力

表13-3. 高速PWM動作比較A出力選択 (共通注意参照)

COM0A1	COM0A0	意味
0	0	標準ポート動作 (OC0A切断)
0	1	WGM02=0 : 標準ポート動作 (OC0A切断) WGM02=1 : 比較一致でOC0Aピントグル(交互)出力
1	0	比較一致でLow、BOTTOMでHighをOC0Aピンへ出力 (非反転動作)
1	1	比較一致でHigh、BOTTOMでLowをOC0Aピンへ出力 (反転動作)

●ビット5,4 – COM0B1,0 : 比較B出力選択 (Compare Match B Output Mode bit 1 and 0)

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

OC0Bがピンに接続されるとき、COM0B1,0ビットの機能はWGM02~0ビット設定に依存します。

表13-5.はWGM02~0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COM0B1,0ビット機能を示します。

表13-6.はWGM02~0ビットが高速PWM動作に設定される時のCOM0B1,0ビットの機能を示します。

表13-7.はWGM02~0ビットが位相基準PWM動作に設定される時のCOM0B1,0ビットの機能を示します。

表13-5. 非PWM動作比較B出力選択

COM0B1	COM0B0	意味
0	0	標準ポート動作 (OC0B切断)
0	1	比較一致でOC0Bピントグル(交互)出力
1	0	比較一致でOC0Bピン Lowレベル出力
1	1	比較一致でOC0Bピン Highレベル出力

表13-7. 位相基準PWM動作比較B出力選択 (共通注意参照)

COM0B1	COM0B0	意味
0	0	標準ポート動作 (OC0B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OC0B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OC0Bピンへ出力

表13-6. 高速PWM動作比較B出力選択 (共通注意参照)

COM0B1	COM0B0	意味
0	0	標準ポート動作 (OC0B切断)
0	1	(予約)
1	0	比較一致でLow、BOTTOMでHighをOC0Bピンへ出力 (非反転動作)
1	1	比較一致でHigh、BOTTOMでLowをOC0Bピンへ出力 (反転動作)

共通注意: COM0x1が設定(1)され、対応するOCR0xがTOPと等しい時に特別な状態が起きます。この状態での比較一致は無視されますが、BOTTOMまたはTOPでの設定(1)または解除(0)は行われます。より多くの詳細については62頁の「高速PWM動作」または63頁の「位相基準PWM動作」をご覧ください。(表13-3,4,6,7.各々での注を纏めました。)

●ビット3,2 – 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1,0 – WGM01,0 : 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(TCCR0B)で得られるWGM02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)、最大カウンタ(TOP)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(下表参照)。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2形式のパルス幅変調(PWM)動作です。61頁の「動作種別」をご覧ください。

表13-8. 波形生成種別選択							
番号	WGM02	WGM01	WGM00	タイマ/カウンタ動作種別	TOP値	OCR0x更新時	TOV0設定時
0	0	0	0	標準動作	\$FF	即時	MAX
1	0	0	1	8ビット位相基準PWM動作	\$FF	TOP	BOTTOM
2	0	1	0	比較一致タイマ/カウンタ解除(CTC)動作	OCR0A	即時	MAX
3	0	1	1	8ビット高速PWM動作	\$FF	BOTTOM	MAX
4	1	0	0	(予約)	-	-	-
5	1	0	1	位相基準PWM動作	OCR0A	TOP	BOTTOM
6	1	1	0	(予約)	-	-	-
7	1	1	1	高速PWM動作	OCR0A	BOTTOM	TOP

注: MAX=\$FF、BOTTOM=\$00です。

13.8.2. TCCR0B – タイマ/カウンタ0制御レジスタB (Timer/Counter0 Control Register B)

ビット	7	6	5	4	3	2	1	0	
\$25 (\$45)	FOC0A	FOC0B	-	-	WGM02	CS02	CS01	CS00	TCCR0B
Read/Write	W	W	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 – FOC0A : OC0A強制変更 (Force Output Compare A)

FOC0AビットはWGM02~0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TCCR0B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。FOC0A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。OC0A出力はCOM0A1,0ビット設定に従って変更されます。FOC0Aビットがスロープ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COM0A1,0ビットに存在する値です。

FOC0Aスロープは何れの割り込みの生成もTOPとしてOCR0Aを使う比較一致タイマ解除(CTC)動作でのタイマ/カウンタの解除(\$00)も行いません。

FOC0Aビットは常に0として読みます。

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

FOC0BビットはWGM02~0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TCCR0B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。FOC0Bビットに論理1を書くと、波形生成部で直ちに比較一致が強制されます。OC0B出力はCOM0B1,0ビット設定に従って変更されます。FOC0Bビットがスロープ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COM0B1,0ビットに存在する値です。

FOC0Bスロープは何れの割り込みの生成も行いません。

FOC0Bビットは常に0として読みます。

●ビット5,4 – 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 – WGM02 : 波形生成種別 (Waveform Generation Mode bit 2)

65頁の「TCCR0A – タイマ/カウンタ制御レジスタA」のWGM01,0ビット記述をご覧ください。

●ビット2~0 – CS02~0 : 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13-9. タイマ/カウンタ0入力クロック選択			
CS02	CS01	CS00	意味
0	0	0	停止 (タイマ/カウンタ0動作停止)
0	0	1	clk _{I/O} (前置分周なし)
0	1	0	clk _{I/O} /8 (8分周)
0	1	1	clk _{I/O} /64 (64分周)
1	0	0	clk _{I/O} /256 (256分周)
1	0	1	clk _{I/O} /1024 (1024分周)
1	1	0	T0ピンの下降端 (外部クロック)
1	1	1	T0ピンの上昇端 (外部クロック)

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

13.8.3. TCNT0 – タイマ/カウンタ0 (Timer/Counter0)

ビット	7	6	5	4	3	2	1	0	
\$26 (\$46)	(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OCR0x間の比較一致消失の危険を誘発します。

13.8.4. OCR0A – タイマ/カウンタ0 比較Aレジスタ (Timer/Counter0 Output Compare A Register)

ビット	7	6	5	4	3	2	1	0	
\$27 (\$47)	(MSB)							(LSB)	OCR0A
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OC0Aピンでの波形出力を生成するのに使えます。

13.8.5. OCR0B – タイマ/カウンタ0 比較Bレジスタ (Timer/Counter0 Output Compare B Register)

ビット	7	6	5	4	3	2	1	0	
\$28 (\$48)	(MSB)							(LSB)	OCR0B
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OC0Bピンでの波形出力を生成するのに使えます。

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

ビット (\$6E)	7	6	5	4	3	2	1	0	
	–	–	–	–	–	OCIE0B	OCIE0A	TOIE0	TIMSK0
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

● ビット7～3 – 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。

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

OCIE0Bビットが1を書かれ、ステータスレジスタ(SREG)の全割り込み許可(I)ビットが設定(1)されると、タイマ/カウンタ0比較B一致割り込みが許可されます。タイマ/カウンタ0で比較B一致が起こる、換言するとタイマ/カウンタ0割り込み要求フラグレジスタ(TIFR0)で比較B割り込み要求フラグ(OCF0B)が設定(1)されると、対応する割り込みが実行されます。

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

OCIE0Aビットが1を書かれ、ステータスレジスタ(SREG)の全割り込み許可(I)ビットが設定(1)されると、タイマ/カウンタ0比較A一致割り込みが許可されます。タイマ/カウンタ0で比較A一致が起こる、換言するとタイマ/カウンタ0割り込み要求フラグレジスタ(TIFR0)で比較A割り込み要求フラグ(OCF0A)が設定(1)されると、対応する割り込みが実行されます。

● 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0割り込み要求フラグレジスタ(TIFR0)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。

13.8.7. TIFR0 – タイマ/カウンタ0割り込み要求フラグレジスタ (Timer/Counter 0 Interrupt Flag Register)

ビット \$15 (\$35)	7	6	5	4	3	2	1	0	
	–	–	–	–	–	OCF0B	OCF0A	TOV0	TIFR0
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

● ビット7～3 – 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。

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

OCF0Bビットは比較一致がタイマ/カウンタ(TCNT0)と比較レジスタ(OCR0B)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、OCF0B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OCF0B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0割り込み許可レジスタ(TIMSK0)のタイマ/カウンタ0比較B一致割り込み許可(OCIE0B)ビット、OCF0Bが設定(1)されると、タイマ/カウンタ0比較B一致割り込みが実行されます。

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

OCF0Aビットは比較一致がタイマ/カウンタ(TCNT0)と比較レジスタ(OCR0A)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、OCF0A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OCF0A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0割り込み許可レジスタ(TIMSK0)のタイマ/カウンタ0比較A一致割り込み許可(OCIE0A)ビット、OCF0Aが設定(1)されると、タイマ/カウンタ0比較A一致割り込みが実行されます。

● 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0割り込み許可レジスタ(TIMSK0)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。位相基準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。

これらフラグの設定はWGM02～0ビット設定に依存します。66ページの波形生成種別ビット記述の表13-8を参照してください。

14. 16ビット タイマ/カウンタ (タイマ/カウンタ1,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:

- 真の16ビット設計 (換言すれば16ビットPWMの許可)
- 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)

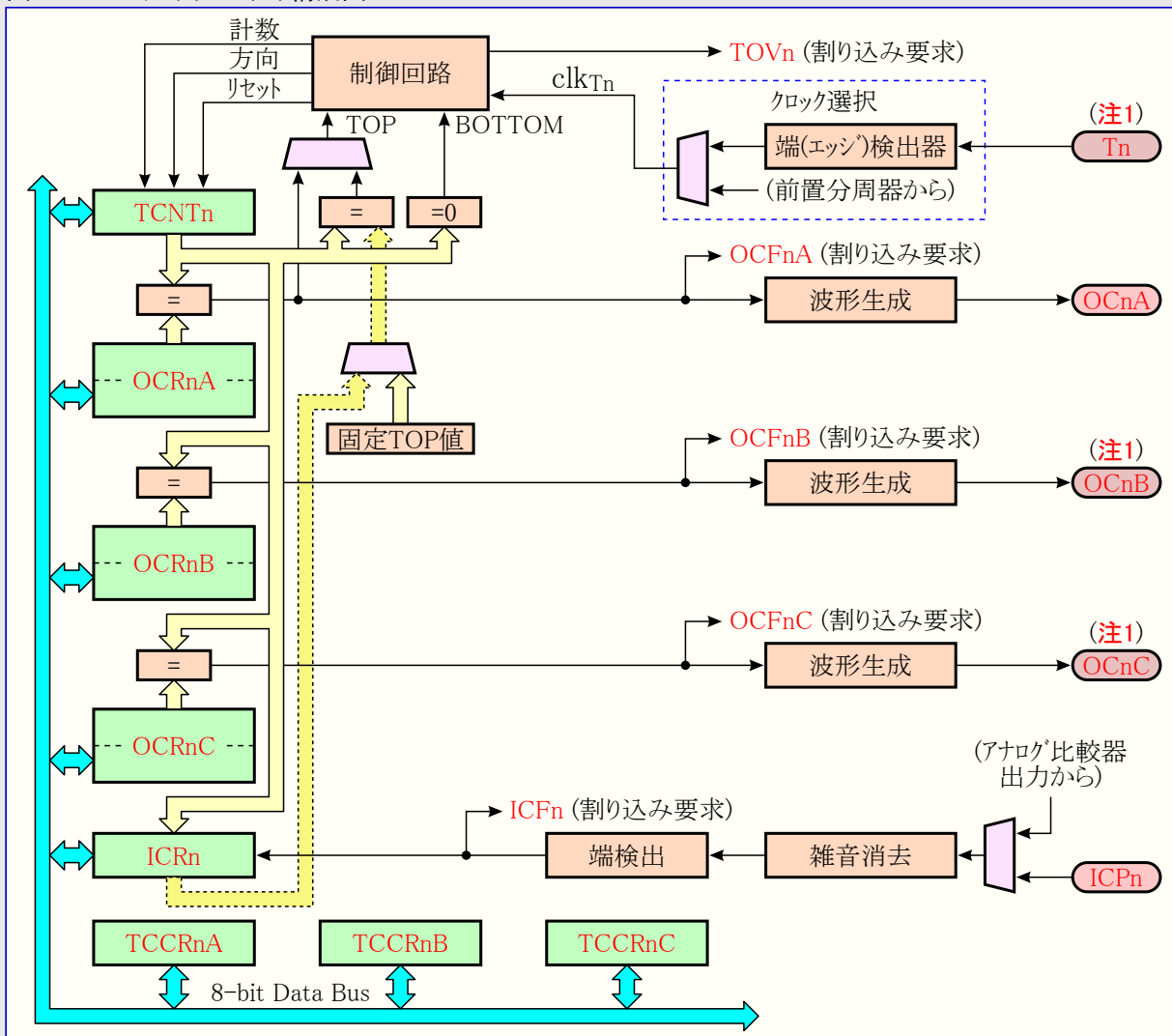
14.1. 概要

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/カウンタの簡略化構成図は図14-1.で示されます。CPUがアクセス可能な(I/OビットとI/Oピンを含む)I/Oレジスタは、赤文字(訳注:原文太字)で示されます。デバイス仕様のI/Oレジスタとビット位置は83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。

28頁の「PRR0 - 電力削減レジスタ」のPRTIM1ビットはタイマ/カウンタ1部を許可するために0を書かれなければなりません。29頁の「PRR1 - 電力削減レジスタ」のPRTIM3ビットはタイマ/カウンタ3部を許可するために0を書かれなければなりません。

図14-1. 16ビット タイマ/カウンタ構成図



注: タイマ/カウンタ1,3のピン配置と説明については2頁の「ピン配置」、45頁の表11-3、47頁の表11-6、48頁の表11-8を参照してください。

注1: T3入力、OC3B出力、OC3C出力が製品で利用できないため、Tn,OCnB,OCnCは各々、T1,OC1B,OC1Cだけを参照します。

14.1.1. 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71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(TIFRn)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(TIMSKn)で個別に遮蔽(禁止)されます。TIFRnとTIMSKnはこの図で示されません。

このタイマ/カウンタは内部的、前置分周器経由、または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。75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(192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出力生成用に使いません。けれどもこの場合、TOP値は動作中に変更されるのをTOP値に許す2重緩衝化します。固定的なTOP値が必要とされる場合、ICRnが代わりに使い、PWM出力として使われるべきOCRnAを開放します。

14.1.2. 定義

次の定義は本文書を通して広範囲に使われます。

表14-1. 用語定義	
用語	意味
BOTTOM	カウンタが\$0000に到達した時。
MAX	カウンタが\$FFFF(65535)に到達した時。
TOP	カウンタがTOP値に到達した時(計数動作での最大値と等しくなった時)。TOP値は固定値(\$00FF,\$01FF,\$03FF)、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。

14.2. 16ビットレジスタのアクセス

TCNTn, OCRnA, OCRnB, OCRnC, ICnRn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、この同じ一時レジスタが共用されます。下位バイトアクセスが16ビット読み書き動作を起動します。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, ICnRn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言語を使う時にコンパイラが16ビットアクセスを扱うことに注意してください。

アセンブリ言語プログラム例

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

注: 5頁の「コード例」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: 5頁の「コード例」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: 5頁の「コード例」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必要があります。

14.2.1. 上位バイト一時レジスタの再使用

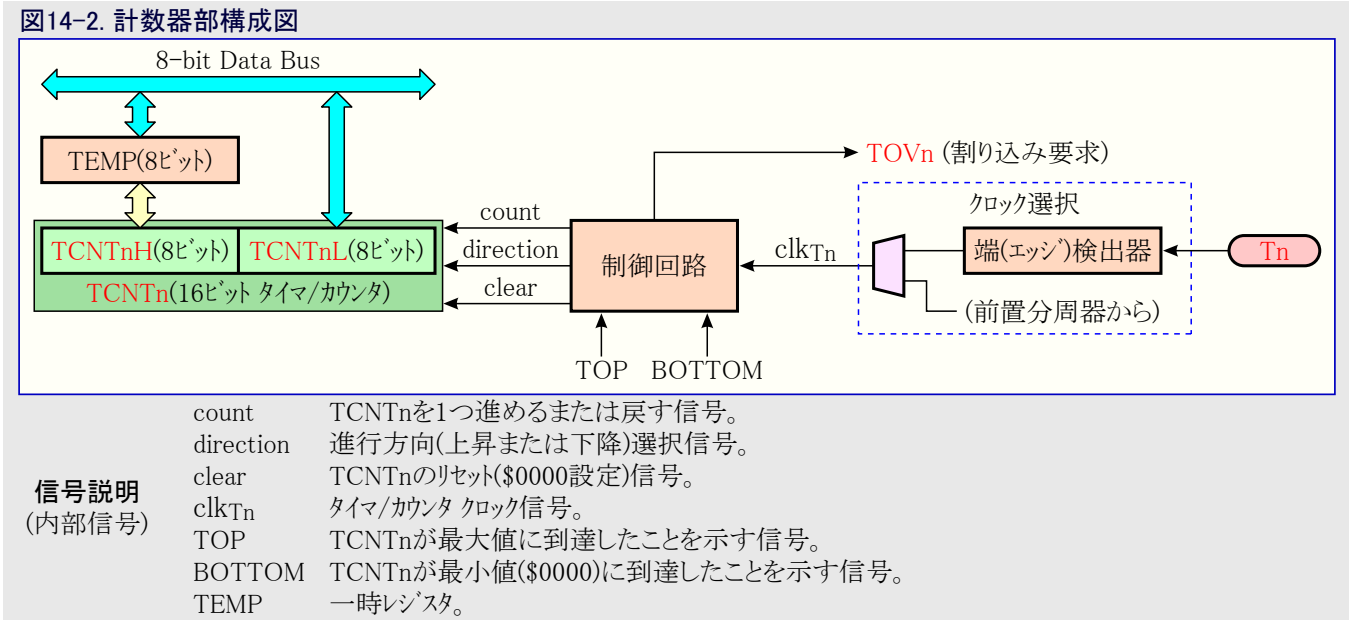
書かれる全レジスタについて上位バイトが同じ複数16ビットレジスタ書き込みなら、上位バイトは1度書かれることだけが必要です。けれども直前で記述した非分断操作の同じ規則が、この場合にも適用されることに注意してください。

14.3. 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55頁の「タイマ/カウンタ0、タイマ/カウンタ1、タイマ/カウンタ3の前置分周器」をご覧ください。

14.4. 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14-2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)。

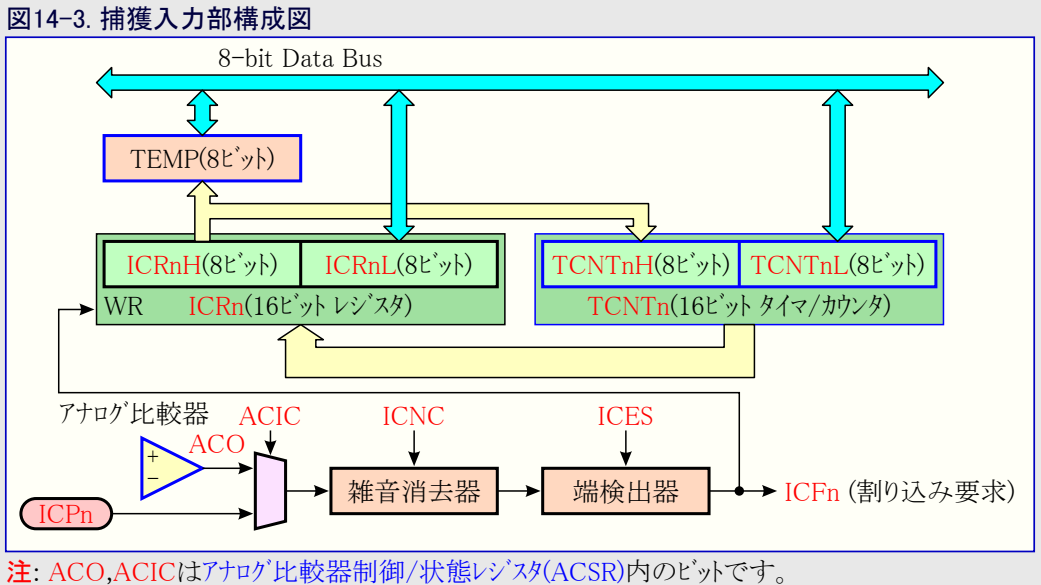
計数順序(方法)はタイマ/カウンタ制御レジスタA(TCCRnA)とタイマ/カウンタ制御レジスタB(TCCRnB)に配置された波形生成種別(WGMn3~0)ビットの設定によって決定されます。これらはカウンタ動作(計数)方法と波形がOCnx比較出力に生成される方法間の接続に近いものです。進化した計数順序と波形生成についてより多くの詳細に対しては78頁の「動作種別」をご覧ください。

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。

14.5. 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ICP_n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14-3.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(ICP_n)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(TCNT_n)の16ビット値が捕獲レジスタ(ICR_n)に書かれます。捕獲割り込み要求フラグ(ICF_n)はTCNT_n値がICR_n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, ICLEN=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。ICF_n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(ICR_n)の16ビット値読み込みは、初めに下位バイト(ICR_nL)、その後に上位バイト(ICR_nH)を読むことによって行われます。下位バイトが読まれる時に上位バイトが上位バイト一時レジスタ(TEMP)に複写されます。CPUがICR_nH I/O位置を読むと、この一時レジスタをアクセスします。

ICR_nはカウンタのTOP値定義にICR_n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ICR_nに書かれ得る前に波形生成種別(WGMn3~0)ビットが設定されなければなりません。ICR_nに書く時は下位バイトがICR_nLに書かれる前に、上位バイトがICR_nH 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71頁の「16ビットレジスタのアクセス」を参照してください。

14.5.1. 捕獲起動元

捕獲入力部用の主な起動元は捕獲起動入力(ICP_n)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(ICF_n)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(ICP_n)ピンとアナログ比較器出力(ACO)の両入力、T_nピン(55頁の図12-1.参照)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。けれども雑音消去が許可されると、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ICR_nを使う波形生成種別に設定されないなら、雑音消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ICP_n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。

14.5.2. 雑音消去器

雑音消去器は簡単なデジタル濾波器機構を使うことによって雑音耐性を改善します。雑音消去器の入力は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。

雑音消去器はタイマ/カウンタ制御レジスタB(TCCRnB)の捕獲入力雑音消去許可(ICNCn)ビットの設定(1)によって許可されます。許可したとき、雑音消去器は入力に印加した変更からICR_n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。雑音消去器はシステムクロックを使い、従って前置分周器によって影響されません。

14.5.3. 捕獲入力の使用

捕獲入力機能を使う主な要求(目的)は入って来る出来事に対して十分なプロセッサ能力を当てがうことです。2つの出来事間の時間が際どいとして、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。

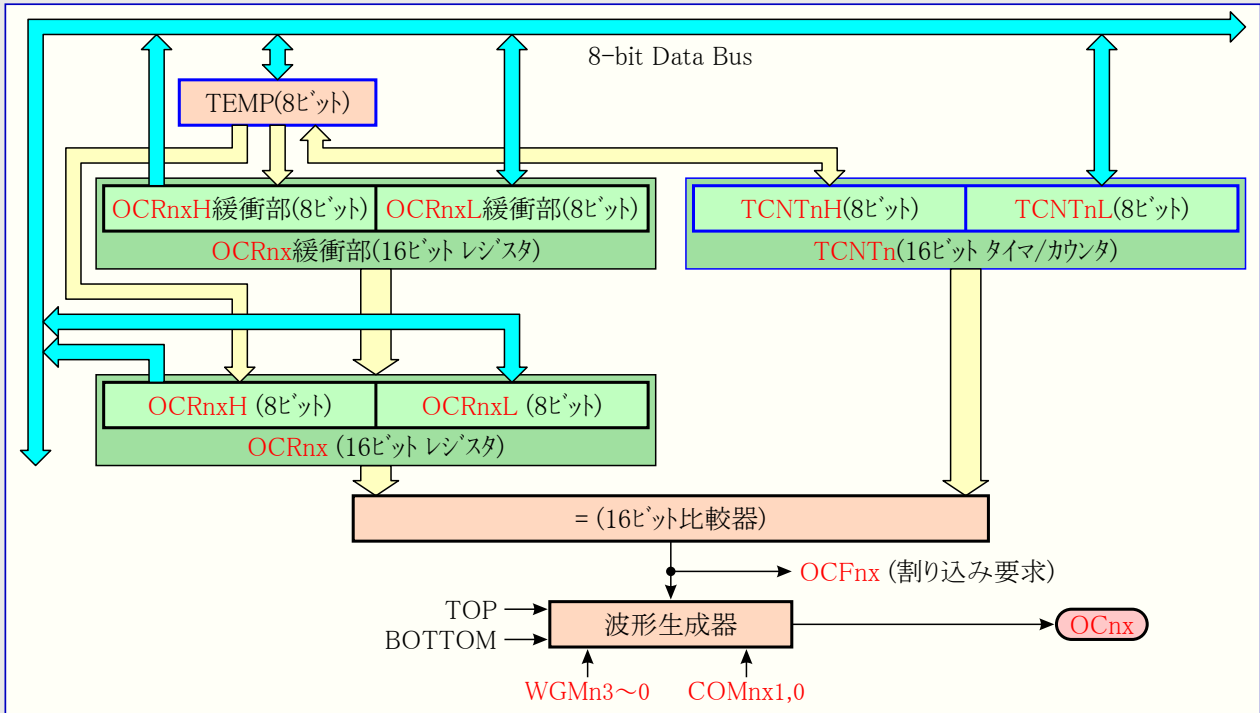
14.6. 比較出力部

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, **OCIEnx=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(78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14-4.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。

図14-4. 比較出力部構成図



OCRnxは12種類のパルス幅変調(PWM)のどれかを使う時に2重緩衝化されます。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。その後に下位バイト(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71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。

14.6.1. 強制比較出力

非PWM波形生成動作での比較器の一致出力は強制変更(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)。

14.6.2. 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。

14.6.3. 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かंनाいかに拘らず、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。

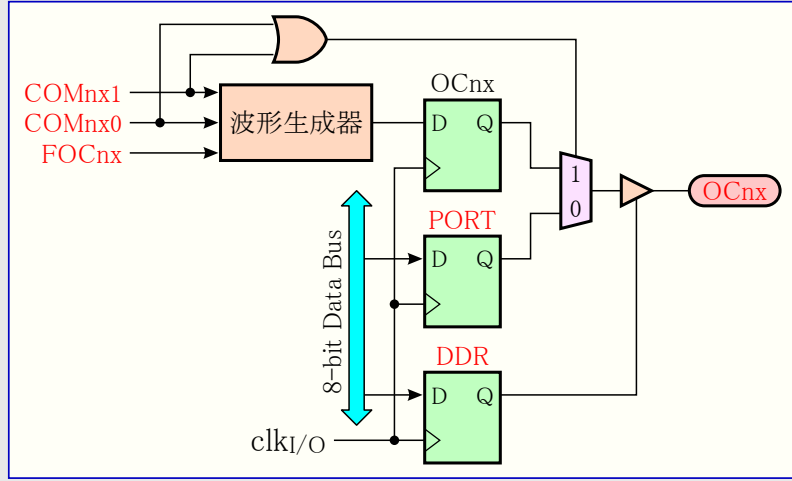
OCnxの初期設定はポートピンに対するポート方向レジスタを出力に設定する前に行われるべきです。OCnx値を設定する一番簡単な方法は標準動作で強制変更(FOCnx)ストローブビットを使うことです。波形生成種別間を変更する時であっても、OCnx(内部)レジスタはその値を保ちます。

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。

14.7. 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14-5はCOMnx1,0ビット設定によって影響される論理回路の簡単化した図を示します。図のI/Oレジスタ、I/Oビット、I/Oピンは赤文字(訳注:原文は太字)で示されます。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14-5. 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14-2、表14-3、表14-4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。83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。

14.7.1. 比較一致出力選択と波形生成

波形生成器は標準、CTC、PWM動作で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83頁の表14-2を参照してください。高速PWM動作については83頁の表14-3、位相基準PWMと位相/周波数基準PWMについては83頁の表14-4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。

14.8. 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。77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82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。

14.8.1. 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。

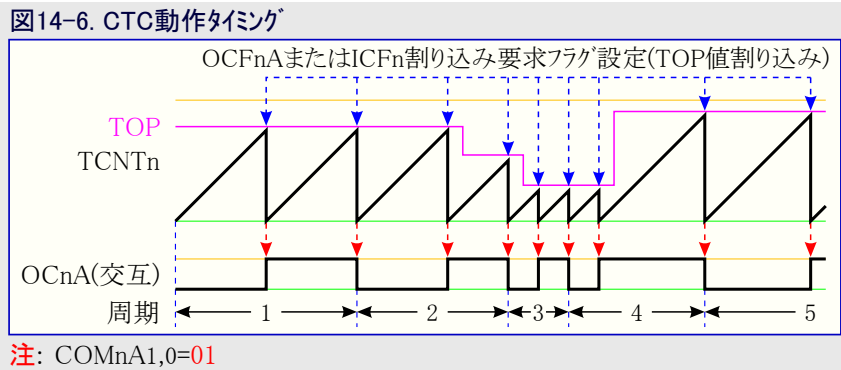
捕獲入力部は標準動作で使用が容易です。けれども外部の事象間の最大間隔がタイマ/カウンタの分解能(16ビット長)を越えてはならないことに気付いてください。事象間の間隔が長すぎる場合、捕獲部に対して分解能を拡張するために、タイマ/カウンタ溢れ割り込みまたは前置分周器が使われなければなりません。

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。

14.8.2. 比較一致タイマ/カウンタ解除(CTC)動作

比較一致タイマ/カウンタ解除(CTC)動作(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14-6.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が生成できます。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らないでしょう。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。

14.8.3. 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。

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

高速PWM動作でのカウンタはカウンタ値が固定値\$00FF,\$01FF,\$03FF(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14-7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、OCFnxまたはICFnx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時に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。その後の結果はカウンタが(その回の)TOP値での比較一致を失う事です。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らないでしょう。しかし、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値を使う時に上手くなります。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。83頁の表14-3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。

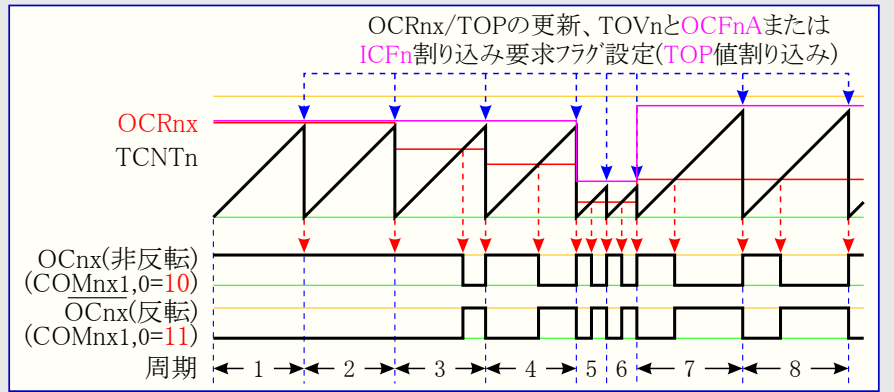
PWM出力周波数は次式によって計算できます。変数Nは前置分周数(1,8,64,256,1024)を表します。

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

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)定常的な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 $f_{\text{OCnA}} = f_{\text{clk_I/O}} / 2$ 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14-7. 高速PWM動作タイミング



14.8.4. 位相基準PWM動作

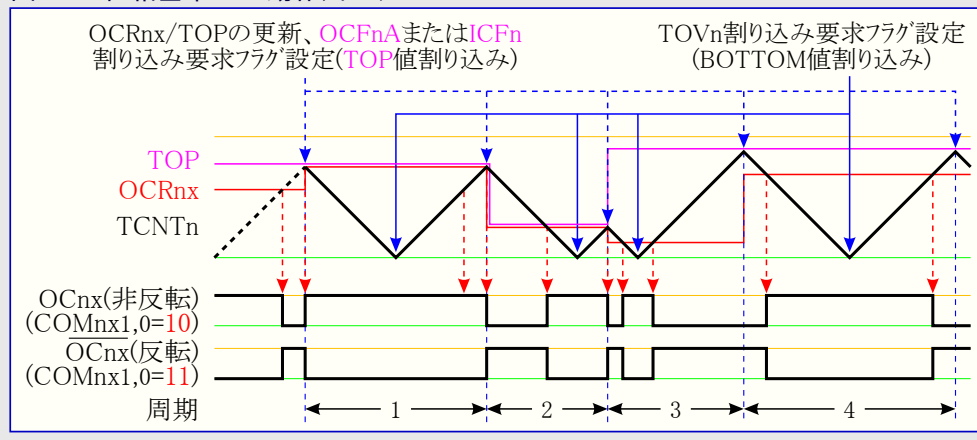
位相基準パルス幅変調(PWM)動作(WGMn3~0=0001,0010,0011,1010,1011)は高分解能で正しい位相のPWM波形生成選択を提供します。位相基準PWM動作は両傾斜(三角波)動作を基準とした位相/周波数基準PWMと似ています。カウンタはBOTTOM(\$0000)からTOPへ、そしてその後にTOPからBOTTOMへを繰り返し計数します。非反転比較出力動作(COMnx1,0=10)での比較出力(OCnx)は上昇計数中のTCNTnとOCRnx間の比較一致で解除(0)され、下降計数中の比較一致で設定(1)されます。反転出力動作(COMnx1,0=11)での動作は逆にされます。両傾斜(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14-8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14-8. 位相基準PWM動作タイミング



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14-8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)が違うとき、その周期の2つの傾斜長は異なるでしょう。この長さの相違が出力での非対称な結果を生じます。

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

位相基準PWM動作での比較部は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(83頁の表14-4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。

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

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。

14.8.5. 位相/周波数基準PWM動作

位相/周波数基準パルス幅変調(PWM)動作(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。

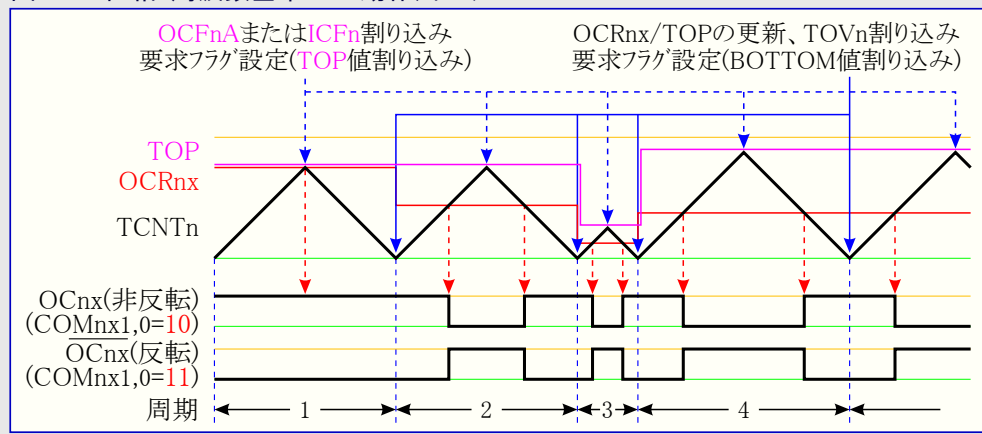
位相基準と位相/周波数基準PWM動作間の主な違いはOCRnxレジスタがOCRnx緩衝部によって更新される時(訳補:TOPとBOTTOM)です(図14-8と図14-9参照)。

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間、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14-9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14-9. 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14-9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(83頁の表14-4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。

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

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。

14.9. 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} 緩衝部値で更新される時(2重緩衝を使う動作種別のみ)の情報を含みます。図14-10は OCF_{nx} の設定についてのタイミング図を示します。

図14-10. 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

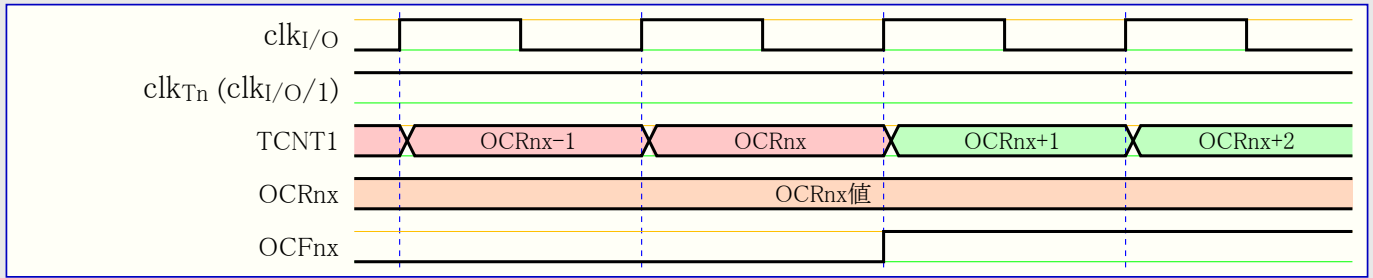


図14-11は同じタイミング データを示しますが、前置分周器が許可されています。

図14-11. 前置分周器($\text{f}_{\text{clk}_{\text{I/O}}/8}$)のタイマ/カウンタ、 OCF_{nx} 設定 タイミング

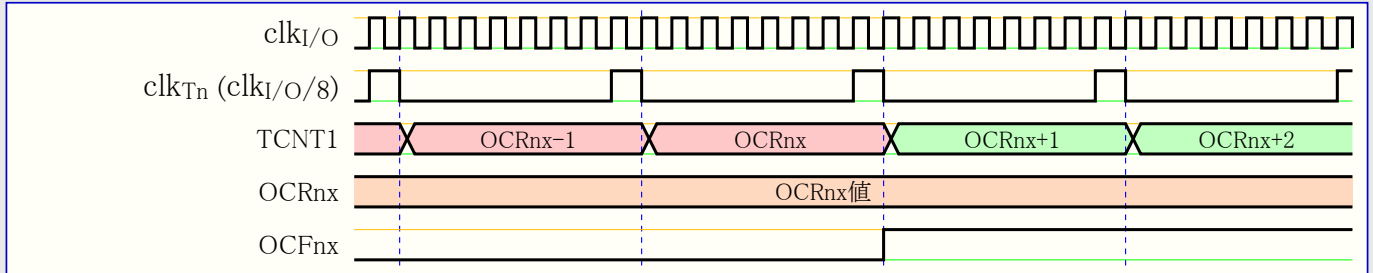


図14-12は各動作種別でのTOP近辺の計数手順を示します。位相/周波数基準PWM動作使用時の OCR_{nx} レジスタはBOTTOMで更新されます。タイミング図は同じになりますが、当然TOPはBOTTOMで、TOP-1はBOTTOM+1でなどのように置き換えられます。BOTTOMで TOV_{n} を設定(1)する動作種別についても、同様な名称変更が適用されます。

図14-12. 前置分周なし(1/1)のタイマ/カウンタ、TOP近辺 タイミング

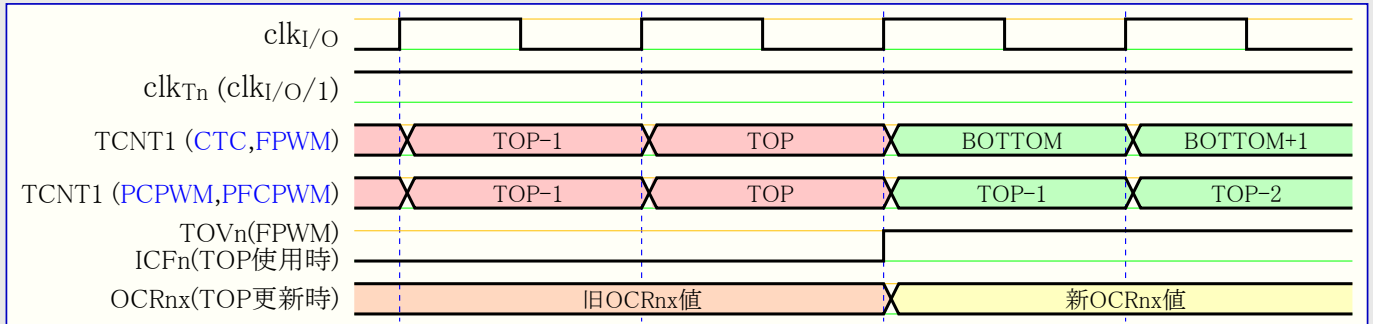
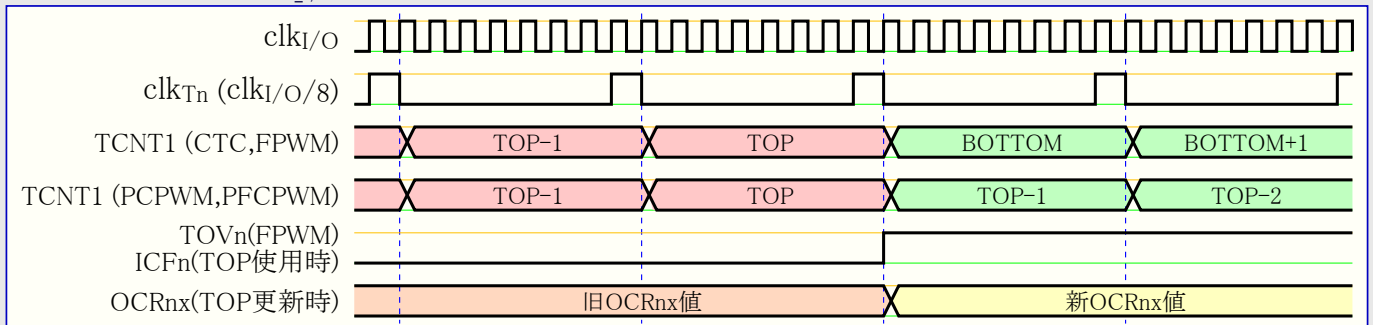


図14-13は同じタイミング データを示しますが、前置分周器が許可されています。

図14-13. 前置分周器($\text{f}_{\text{clk}_{\text{I/O}}/8}$)のタイマ/カウンタ、TOP近辺 タイミング



14.10. 16ビット タイマ/カウンタ1,3用レジスタ

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

ビット (\$80)	7	6	5	4	3	2	1	0	
	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	

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

ビット (\$90)	7	6	5	4	3	2	1	0	
	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 – COMnA1,0 : 比較nA出力選択 (Compare Output Mode A bit 1 and 0)
- ビット5,4 – COMnB1,0 : 比較nB出力選択 (Compare Output Mode B bit 1 and 0)
- ビット3,2 – COMnC1,0 : 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。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。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、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14-2. 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14-3. 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, 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=111X : 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) 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79頁の「高速PWM動作」をご覧ください。

表は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14-4. 位相基準または位相/周波数基準PWM動作での比較出力選択 (注: 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) 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80頁の「位相基準PWM動作」をご覧ください。

●ビット1,0 – WGMn1,0 : 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(下表参照)。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。78頁の「動作種別」をご覧ください。

表14-5. 波形生成種別選択								
番号	WGMn3	WGMn2 (CTCn)	WGMn1 (PWMn1)	WGMn0 (PWMn0)	タイマ/カウンタ動作種別	TOP値	OCRnx 更新時	TOVn 設定時
0	0	0	0	0	標準動作	\$FFFF	即値	MAX
1	0	0	0	1	8ビット位相基準PWM動作	\$00FF	TOP	BOTTOM
2	0	0	1	0	9ビット位相基準PWM動作	\$01FF	TOP	BOTTOM
3	0	0	1	1	10ビット位相基準PWM動作	\$03FF	TOP	BOTTOM
4	0	1	0	0	比較一致タイマ/カウンタ解除(CTC)動作	OCRnA	即値	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	即値	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。

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

ビット (\$81)	7	6	5	4	3	2	1	0	
	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	

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

ビット (\$91)	7	6	5	4	3	2	1	0	
	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)ピンからの入力が増幅されます。この増幅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、捕獲入力はそのによって4発振器(システムクロック)周期遅らされます。
- ビット6 – 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(TCCRNbA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 – WGMn3,2 : 波形生成種別 (Waveform Generation Mode bit 3 and 2)

TCCRNAnのWGMn1,0ビット記述をご覧ください。

●ビット2~0 – CSn2~0 : 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14-10と図14-11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14-6. タイマ/カウンタn入力クロック選択

CSn2	CSn1	CSn0	意味
0	0	0	停止 (タイマ/カウンタ動作停止)
0	0	1	clkI/O (前置分周なし)
0	1	0	clkI/O/8 (8分周)
0	1	1	clkI/O/64 (64分周)
1	0	0	clkI/O/256 (256分周)
1	0	1	clkI/O/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)

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

ビット (\$82)	7	6	5	4	3	2	1	0	
	FOC1A	FOC1B	FOC1C	–	–	–	–	–	TCCR1C
Read/Write	W	W	W	R	R	R	R	R	
初期値	0	0	0	0	0	0	0	0	

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

ビット (\$92)	7	6	5	4	3	2	1	0	
	FOC3A	–	–	–	–	–	–	–	TCCR3C
Read/Write	W	R	R	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。

14.10.7. TCNT1H,TCNT1L (TCNT1) – タイマ/カウンタ1 (Timer/Counter1)

ビット (\$85)	15	14	13	12	11	10	9	8	
	(MSB)								TCNT1H
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 (\$84)	7	6	5	4	3	2	1	0	
								(LSB)	TCNT1L
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	

14.10.8. TCNT3H,TCNT3L (TCNT3) – タイマ/カウンタ3 (Timer/Counter3)

ビット (\$95)	15	14	13	12	11	10	9	8	
	(MSB)								TCNT3H
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 (\$94)	7	6	5	4	3	2	1	0	
								(LSB)	TCNT3L
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、読み書き両方についてタイマ/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。71 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。

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

ビット (\$89)	15	14	13	12	11	10	9	8	
	(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 (\$88)	7	6	5	4	3	2	1	0	
								(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	

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

ビット (\$8B)	15	14	13	12	11	10	9	8	
	(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 (\$8A)	7	6	5	4	3	2	1	0	
								(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	

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

ビット (\$8D)	15	14	13	12	11	10	9	8	
	(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 (\$8C)	7	6	5	4	3	2	1	0	
								(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	

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

ビット (\$99)	15	14	13	12	11	10	9	8	
	(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 (\$98)	7	6	5	4	3	2	1	0	
								(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	

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

ビット (\$9B)	15	14	13	12	11	10	9	8	
	(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 (\$9A)	7	6	5	4	3	2	1	0	
								(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	

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

ビット (\$9D)	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 (\$9C)	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。71頁の「16ビットレジスタのアクセス」をご覧ください。

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

ビット (\$87)	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 (\$86)	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	

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

ビット (\$97)	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 (\$96)	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。**71頁の「16ビットレジスタのアクセス」**をご覧ください。

14.10.17. TIMSK1 – タイマ/カウンタ1割り込み許可レジスタ (Timer/Counter1 Interrupt Mask Register)

ビット (\$6F)	7	6	5	4	3	2	1	0	
	–	–	ICIE1	–	OCIE1C	OCIE1B	OCIE1A	TOIE1	TIMSK1
Read/Write	R	R	R/W	R	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

14.10.18. TIMSK3 – タイマ/カウンタ3割り込み許可レジスタ (Timer/Counter3 Interrupt Mask Register)

ビット (\$71)	7	6	5	4	3	2	1	0	
	–	–	ICIE3	–	OCIE3C	OCIE3B	OCIE3A	TOIE3	TIMSK3
Read/Write	R	R	R/W	R	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

● ビット5 – ICIE_n : タイマ/カウンタ_n捕獲割り込み許可 (Timer/Counter n 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_n捕獲割り込みが許可されます。タイマ/カウンタ_n割り込み要求フラグレジスタ(TIFR_n)に配置された**捕獲割り込み要求フラグ(ICFn)**が設定(**1**)されると、対応する割り込みベクタ(36頁の「割り込み」参照)が実行されます。

● ビット3 – OCIE_nC : タイマ/カウンタ_n比較C割り込み許可 (Timer/Counter n Output Compare C 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_n比較C一致割り込みが許可されます。タイマ/カウンタ_n割り込み要求フラグレジスタ(TIFR_n)に配置された**比較_nC割り込み要求フラグ(OCFnC)**が設定(**1**)されると、対応する割り込みベクタ(36頁の「割り込み」参照)が実行されます。

● ビット2 – OCIE_nB : タイマ/カウンタ_n比較B割り込み許可 (Timer/Counter n 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_n比較B一致割り込みが許可されます。タイマ/カウンタ_n割り込み要求フラグレジスタ(TIFR_n)に配置された**比較_nB割り込み要求フラグ(OCFnB)**が設定(**1**)されると、対応する割り込みベクタ(36頁の「割り込み」参照)が実行されます。

● ビット1 – OCIE_nA : タイマ/カウンタ_n比較A割り込み許可 (Timer/Counter n 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_n比較A一致割り込みが許可されます。タイマ/カウンタ_n割り込み要求フラグレジスタ(TIFR_n)に配置された**比較_nA割り込み要求フラグ(OCFnA)**が設定(**1**)されると、対応する割り込みベクタ(36頁の「割り込み」参照)が実行されます。

● **ビット0 – TOIE_n : タイマ/カウンタ_n溢れ割り込み許可** (Timer/Counter *n* Overflow Interrupt Enable)

このビットが**1**が書かれて、**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_n溢れ割り込みが許可されます。タイマ/カウンタ_n割り込み要求フラグレジスタ(TIFR_n)に配置された**タイマ/カウンタ_n溢れ割り込み要求フラグ(TOV_n)**が設定(**1**)されると、対応する割り込みベクタ(**36**頁の「**割り込み**」参照)が実行されます。

14.10.19. TIFR1 – タイマ/カウンタ1割り込み要求フラグレジスタ (Timer/Counter1 Interrupt Flag Register)

ビット	7	6	5	4	3	2	1	0	
\$16 (\$36)	–	–	ICF1	–	OCF1C	OCF1B	OCF1A	TOV1	TIFR1
Read/Write	R	R	R/W	R	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

14.10.20. TIFR3 – タイマ/カウンタ3割り込み要求フラグレジスタ (Timer/Counter3 Interrupt Flag Register)

ビット	7	6	5	4	3	2	1	0	
\$18 (\$38)	–	–	ICF3	–	OCF3C	OCF3B	OCF3A	TOV3	TIFR3
Read/Write	R	R	R/W	R	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

● **ビット5 – ICF_n : タイマ/カウンタ_n捕獲割り込み要求フラグ** (Timer/Counter *n*, Input Capture Flag)

ICP_n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(ICR_n)が**WGM_n3~0**によって**TOP値**として設定されると、ICF_nフラグはカウンタがTOP値に到達する時に設定(**1**)されます。

捕獲割り込みベクタが実行されると、ICF_n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ICF_nは解除(**0**)できます。

● **ビット3 – OCF_nC : タイマ/カウンタ_n比較C割り込み要求フラグ** (Timer/Counter *n*, Output Compare C Match Flag)

このフラグはカウンタ(TCNT_n)値が**比較Cレジスタ(OCR_nC)**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(FOC_nC)ストローブがOCF_nCフラグを設定(**1**)しないことに注意してください。

比較C一致割り込みベクタが実行されると、OCF_nC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OCF_nCは解除(**0**)できます。

● **ビット2 – OCF_nB : タイマ/カウンタ_n比較B割り込み要求フラグ** (Timer/Counter *n*, Output Compare B Match Flag)

このフラグはカウンタ(TCNT_n)値が**比較Bレジスタ(OCR_nB)**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(FOC_nB)ストローブがOCF_nBフラグを設定(**1**)しないことに注意してください。

比較B一致割り込みベクタが実行されると、OCF_nB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OCF_nBは解除(**0**)できます。

● **ビット1 – OCF_nA : タイマ/カウンタ_n比較A割り込み要求フラグ** (Timer/Counter *n*, Output Compare A Match Flag)

このフラグはカウンタ(TCNT_n)値が**比較Aレジスタ(OCR_nA)**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(FOC_nA)ストローブがOCF_nAフラグを設定(**1**)しないことに注意してください。

比較A一致割り込みベクタが実行されると、OCF_nA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OCF_nAは解除(**0**)できます。

● **ビット0 – TOV_n : タイマ/カウンタ_n溢れ割り込み要求フラグ** (Timer/Counter *n* Overflow Flag)

このフラグの設定(**1**)は**WGM_n3~0**ビット設定に依存します。**標準**または**CTC動作**でのTOV_nフラグはタイマ/カウンタ_n溢れ時に設定(**1**)されます。他のWGM_n3~0ビット設定を使う時のTOV_nフラグ動作については**84**頁の**表14-5**を参照してください。

タイマ/カウンタ_n溢れ割り込みベクタが実行されると、TOV_n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TOV_nは解除(**0**)できます。

15.2.1. 速度

タイマ/カウンタ4の最高速は64MHzです。けれども4V以下の供給電圧が使われる場合、タイマ/カウンタ4が低電圧状態で充分速く動作しないため、入力周波数を下げることが推奨されます。

15.2.2. 精度

タイマ/カウンタ4は8ビットタイマ/カウンタとして代替使用できる10ビットタイマ/カウンタ部です。タイマ/カウンタ4レジスタは基本的に8ビットレジスタですが、10ビット精度が使われる場合に8ビットデータバス経由でAVR CPUによって10ビットタイマ/カウンタ4の上位2ビットをアクセスするための共通一時緩衝部として使える、2ビット上位バイト(TC4H)レジスタが上位にあります。然るに、この10ビットレジスタの上位2ビットが0を書かれるなら、タイマ/カウンタ4は8ビットタイマ/カウンタとして働きます。何れかの8ビットレジスタの下位バイト読み込み時、上位2ビットがTC4Hレジスタへ書かれ、何れかの8ビットレジスタの下位バイト書き込み時、上位2ビットがTC4Hレジスタから書かれます。8ビットバス経由で10ビットタイマ/カウンタ値をアクセスする時は特別な手順に従わなければなりません。これらの手順は104頁の「10ビットレジスタアクセス」項で記述されます。

増強PWM動作は出力周波数が通常動作と同じに留まる一方、各比較A/B/Dレジスタに分解能ビット追加を可能にします。これはTC4Hレジスタが11ビット増強PWM動作でのMSBとなる、もう1ビットを含むことを意味します。この機能についての詳細とその使用法に関しては96頁の「増強された比較/PWM動作」項をご覧ください。

15.2.3. 関係レジスタ

タイマ/カウンタ4(TCNT4)と比較レジスタ(OCR4A, OCR4B, OCR4C, OCR4D)はTCNT4内容と比較されるべきデータ元として使われる8ビットレジスタです。OCR4A, OCR4B, OCR4DはOC4A, OC4B, OC4Dピンの動作を決め、それらは比較一致割り込みも生成できます。OCR4Cはタイマ/カウンタ4のTOP値、換言すると比較一致での解除値を保持します。タイマ/カウンタ4上位バイト(TC4H)レジスタは10ビット精度が使われる場合に、タイマ/カウンタ4レジスタの上位ビットへのアクセスに共通一時緩衝部として使われる2(+1)ビットレジスタです。

割り込み要求(TOV4, OCF4A, OCF4B, OCF4D)と障害保護のFPF4)信号はタイマ/カウンタ4割り込み要求フラグレジスタ(TIFR4)とタイマ/カウンタ4制御レジスタD(TCCR4D)で見えます。この割り込みはタイマ/カウンタ4割り込み許可レジスタ(TIMSK4)とTCCR4Dの障害保護割り込許可(FPIE4)ビットとで個別に遮蔽(許可/禁止)されます。

制御信号はタイマ/カウンタ制御レジスタ、TCCR4A, TCCR4B, TCCR4C, TCCR4D, TCCR4Eで見つかります。

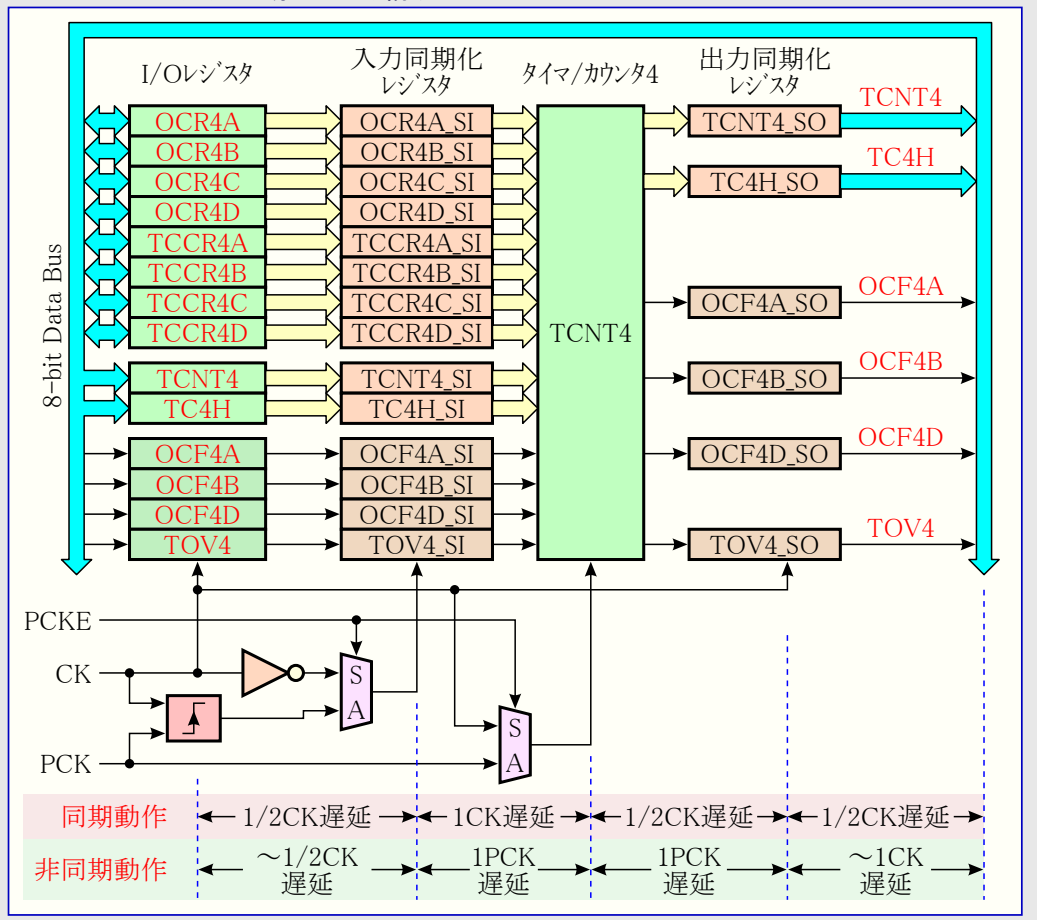
15.2.4. 同期化

同期クロック動作でのタイマ/カウンタ4と前置分周器は前置分周器が非同期動作で高速64MHzのPCKクロック動作をする一方、どのクロック元からもCPUの走行(動作)を許します。これはCPUクロック範囲と高速周辺クロック範囲間の境界同期化があるので可能です。図15-2はタイマ/カウンタ4同期化レジスタ構成図を示し、レジスタ間の遅延を記載します。全てのクロック開閉の詳細はこの図で示されません。

タイマ/カウンタ4レジスタ値は計数動作に影響を及ぼす前に入力同期化遅延を引き起こす内部同期化レジスタを通ります。TCCR4A, TCCR4B, TCCR4C, TCCR4D, OCR4A, OCR4B, OCR4C, OCR4Dのレジスタはレジスタ書き込み後に正しく読み戻せます。読み戻し値はタイマ/カウンタ4(TCNT4)、タイマ/カウンタ4上位バイト(TC4H)レジスタとフラグ(TOV4, OCF4A, OCF4B, OCF4D)に関して入力と出力の同期化のために遅延されます。

非同期タイマ/カウンタ4の同期化機構はシステムクロックがHighの時に最低2つのPCK端を必要とするため、システムクロック周波数はPCK周波数の1/2よりも低くなければなりません。システムクロック周波数が高すぎる場合にデータまたは制御値を失う危険があります。

図15-2. タイマ/カウンタ4同期化レジスタ構成図



15.2.5. 定義

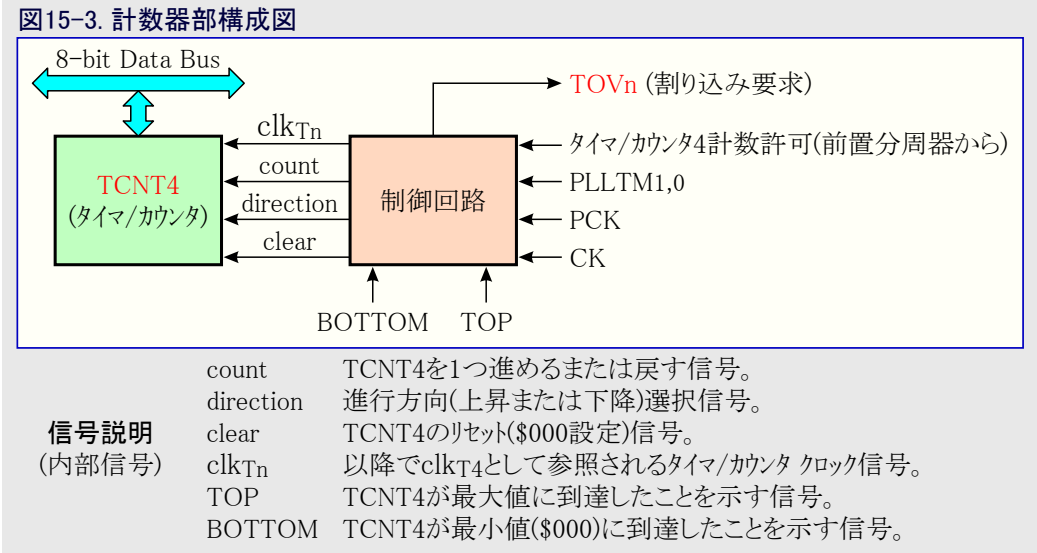
本章でのレジスタとビット参照の多くは一般形で書かれます。小文字の'n'はタイマ/カウンタ番号、この場合は4で置き換えます。小文字のxは比較出力部のチャンネル名を表し、この場合はA,B,CまたはDです。然しながらプログラムでレジスタまたはビット定義に使う時は正確な形式が使われなければなりません(例えばタイマ/カウンタ4のカウンタ値のアクセスに対してのTCNT4のように)。

表の定義は本文書を通して広範囲に渡って使われます。

用語	意味
BOTTOM	タイマ/カウンタが\$000に到達した時。
MAX	タイマ/カウンタが\$3FF(1023)に到達した時。
TOP	タイマ/カウンタが計数中に最高値と等しくなる時の(OCR1Cに格納された)TOP値。TOPはリセット後の既定として値\$FFを持ちます。

15.3. 計数器部

タイマ/カウンタ4の主な部分は設定可能な双方向カウンタ部です。図15-3は、このカウンタとその周辺環境の構成図を示します。



使用した動作種別に依存してカウンタは各タイマ/カウンタクロック(clkT4)で解除(\$000)、増加(+1)、または減少(-1)されます。タイマ/カウンタクロック(clkT4)はクロック選択(CS43~0)ビットと高速タイマ用PLL後置分周器(PLLTM1,0)ビットを使って同期クロックまたは非同期PLLクロックから生成されます。クロック元が選ばれない(CS43~0=0000)時にタイマ/カウンタは停止されます。けれどもTCNT4値はタイマ/カウンタクロック(clkT4)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(TCCR4A,TCCR4C,TCCR4D)に配置された波形生成種別(WGM41,0)ビットとパルス幅変調許可(PWM4x)ビットの設定によって決定されます。進化した計数順序と波形生成については97頁の「動作種別」をご覧ください。

タイマ/カウンタ溢れ(TOV4)フラグはWGM41,0とPWM4x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。この溢れフラグはCPU割り込み発生に使えます。

15.3.1. 非同期動作に対するタイマ/カウンタ4の初期化

タイマ/カウンタ4を非同期動作へ変更するには以下の手順に従ってください。

- 1. PLLを許可してください。
- 2. PLLの安定のために100µs待機してください。
- 3. PLOCKビットが設定(1)されるまでポーリングしてください。
- 4. 非同期動作を許可するためにPLL周波数制御(PLLFRQ)レジスタで高速タイマ用後置分周器選択(PLLTM1,0)ビットを(00以外の値に)設定してください。

15.4. 比較出力部

この比較器はTCNT4と比較レジスタ(OCR4A,OCR4B,OCR4CとOCR4D)を継続的に比較します。TCNT4が比較レジスタと等しければ比較器は一致を指示します。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(OCF4A,OCF4BまたはOCF4D)を設定(1)します。対応する割り込みが許可(I=1,OCIE4x=1)されているなら、そ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。比較割り込み要求フラグは割り込みが実行されると、自動的に解除(0)されます。代わりにこのフラグはこの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(WGM41,0)ビットとPWMx許可(PWM4x)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(97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に波形生成器によって使われます。

図15-4.は比較出力部の構成図を示します。

図15-4. 比較出力部構成図

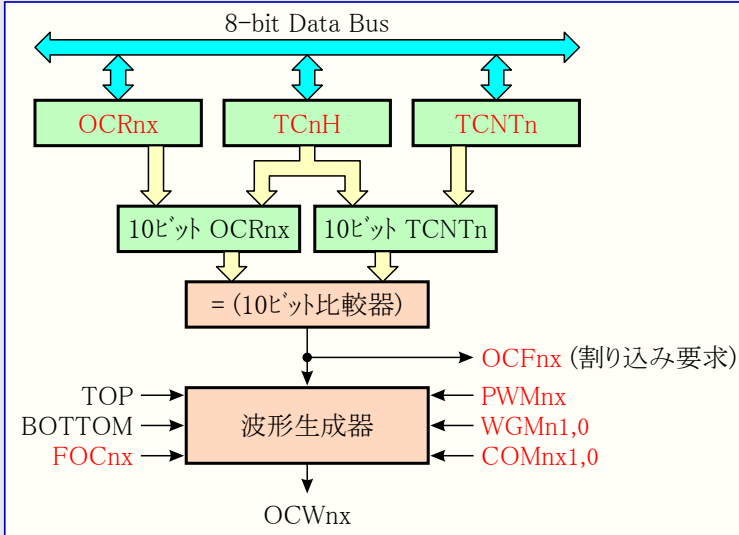
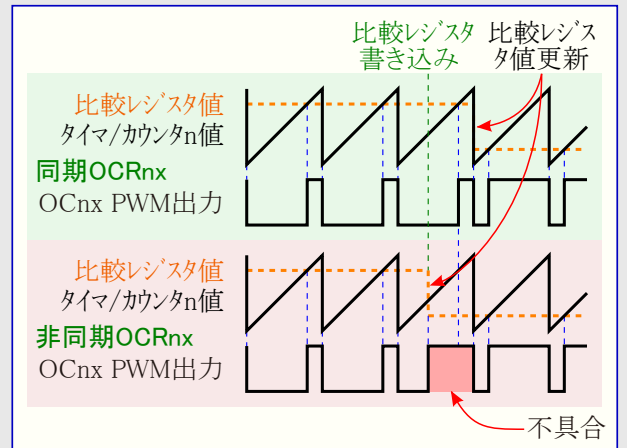


図15-5. 同期/非同期OCR4xの効果



OCR4x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OCR4xの更新を同期化します。この同期化は奇数長や非対称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。例については図15-5をご覧ください。書き込みと更新動作の間の時間中、OCR4A,OCR4B,OCR4C,OCR4Dからの読み込みは一時位置の内容を読みます。これは最も最近書かれた値が常にOCR4A,OCR4B,OCR4C,OCR4Dの読み出しであることを意味します。

15.4.1. 強制比較出力

非PWM波形生成動作での比較器の一致出力は強制変更(FOC4x)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(OCF4x)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、波形出力(OCW4x)は実際の比較一致が起きた場合と同様に更新されます(COM4x1,0ビット設定が波形出力(OCW4x)の設定(1)、解除(0)、1/0交互のどれかを定義)。

15.4.2. TCNT4書き込みによる比較一致妨害

TCNT4への全てのCPU書き込みは、例えばタイマ/カウンタが停止されていても、次のタイマ/カウンタクロック周期で起こるどんな比較一致をも妨げます。この特質はタイマ/カウンタクロックが許可されている時に割り込みを起動することなしにTCNT4と同じ値に初期化されるのをOCR4xに許します。

15.4.3. 比較一致部の使用

どの動作種別でのTCNT4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ので、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、TCNT4を変更する時に危険を伴います。TCNT4に書かれた値がOCR4x値と同じ場合に、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時はBOTTOMに等しいTCNT4値を書き込むべきではありません。

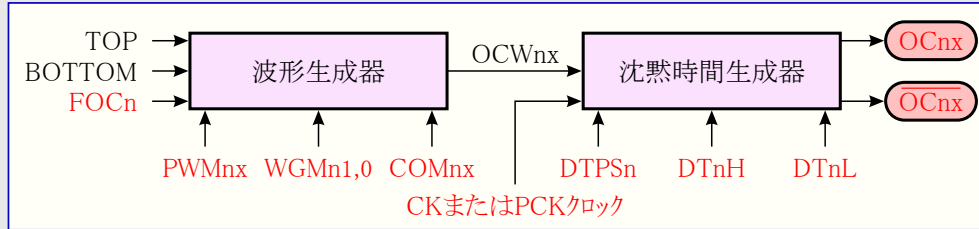
波形出力(OCW4x)の初期設定はポートピンに対するポート方向レジスタを出力に設定する前に行われるべきです。OCW4x値を設定する一番簡単な方法は標準動作で強制変更(FOC4x)ストローブビットを使うことです。波形生成動作種別間を変更する時でも、OCR4x(内部)レジスタはその値を保ちます。

比較出力選択(COM4x1,0)ビットが比較値(OCR4x)と共に2重緩衝されないことに気付いてください。COM4x1,0ビットの変更は直ちに有効となります。

15.5. 沈黙時間生成器 (Dead Time Generator)

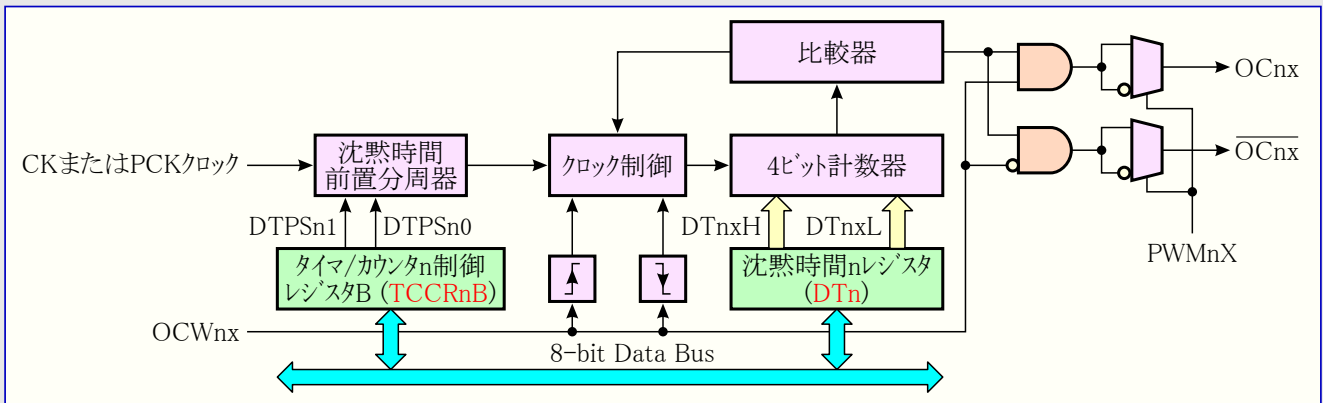
沈黙時間生成器は外部電力制御スイッチの安全な駆動を可能とするためにタイマ/カウンタ4のPWM出力対に対して提供されます。この沈黙時間生成器はPWM動作が許可され、比較出力選択(COM4x1,0)ビットが'01'に設定される時のタイマ/カウンタ4相互補完出力対(OC4xとOC4x)に対して沈黙時間(非重複時間)の挿入に使える独立部です。作業の分担は次のとおりです。波形生成器が波形出力(OCW4x)を生成し、沈黙時間生成器がこの波形出力から非重複PWM出力を生成します。各PWM出力に対して1つで、3つの沈黙時間生成器が提供されます。非重複時間は調整可能で、PWM出力とその補完出力は独立して調整でき、両PWM出力に対して無関係です。

図15-6. タイマ/カウンタ1と沈黙時間生成器



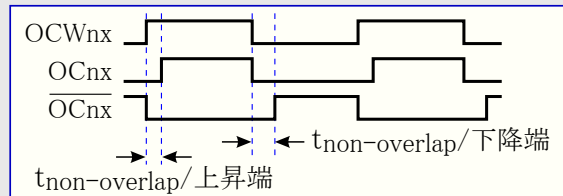
沈黙時間生成器は図15-7で示されるように沈黙時間を計数する4ビット下降カウンタを基にします。これはタイマ/カウンタ4クロック(CKまたはPCK)を1,2,4,8分周できる沈黙時間生成器前の専用前置分周器です。これは生成可能な広範囲の沈黙時間を提供します。この前置分周器はDTPS41,DTPS40の2つの制御ビットによって制御されます。本単位部には沈黙時間計時区間を開始するのに使われる上昇端と下降端の検出器もあります。この端(エッジ)に対応してOC4xまたはOC4xの上昇端の遷移は、この(4ビット)計数器が0へ計数されるまで遅らされます。比較器は計数器と0との比較に使われ、0に達した時に沈黙時間挿入を止めます。この計数器は沈黙時間挿入を開始する時に波形出力(OCW4x)の端に応じてDT4 I/OレジスタからDT4xHまたはDT4xLの4ビット値が読み込まれます。比較出力は沈黙時間が0に調節される時に波形出力から最小で1タイマ/カウンタクロック周期遅延されます。PWM反転動作(PWM4X)ビットが設定(1)なら、OC4xとOC4xは反転されます。これは両出力を沈黙時間中、Highにもさせます。

図15-7. 沈黙時間生成器



この計数時間長はDTPS41,0ビットを使うことによる沈黙時間前置分周器設定の選択と、そして沈黙時間(DT4I/Oレジスタ)の沈黙時間値選択による使用者調整が可能です。DT4レジスタはPWM出力とその補完出力を前置分周された沈黙時間生成器クロック周期数に換算し、独立して制御するDT4xHとDT4xLの2つの4ビット領域から成ります。従ってOC4xとOC4xの上昇端は4ビットのDT4H値によって調節される $t_{non-overlap}$ /上昇端と、4ビットのDT4L値によって調節される $t_{non-overlap}$ /下降端として異なる沈黙時間区間を持ちます。

図15-8. 非重複出力対 (COM4x1,0=01)



15.6. 比較一致出力部

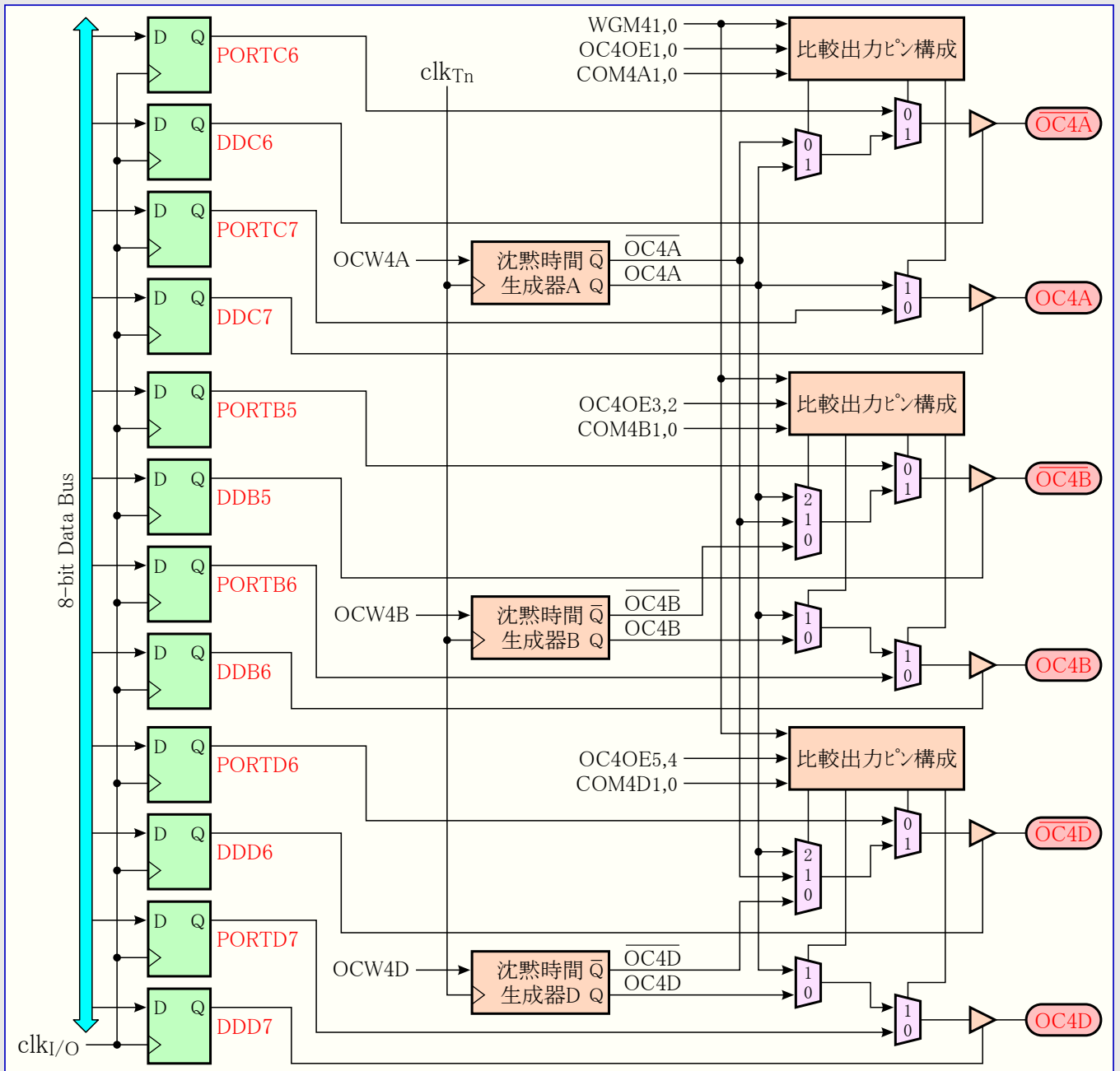
比較出力選択(COM4x1,0)ビットは2つの機能を持ちます。波形生成器は次の比較一致での非反転または反転の波形出力(OCW4x)を定義するのにCOM4x1,0ビットを使います。またCOM4x1,0ビットはOC4xとOC4xのピン出力元を制御します。図15-9はCOM4x1,0ビット設定によって影響を及ぼされる論理回路の簡略化回路図を示します。図のI/Oレジスタ、I/Oビット、I/Oピンは赤文字(訳注:原文太字)で示されます。COM4x1,0ビットによって影響を及ぼされる標準I/Oポート制御レジスタ(PORTとDDR)の部分だけが示されます。

標準動作(非PWM)での沈黙時間生成器は禁止され、比較出力(OC4x)は波形出力(OCW4x)から1タイマ/カウンタクロック周期遅延される同期化器のように働きます。ところがCOM4x1,0ビットが'01'に設定される時の高速PWM動作と位相/周波数基準PWM動作では、非反転と反転の両比較出力が生成され、使用者が設定可能な沈黙時間がそれらの相互補完出力対(OC4xとOC4x)に対して挿入されます。その他のCOM4x1,0ビット設定が使われる時のPWM動作での機能は標準動作と同じです。OC4xの状態を参照する時のその参照はOC4xピンでなく沈黙時間生成器からの比較出力(OC4x)に対してです。システムリセットが起こると、OC4xは'0'にリセットされます。

COM4x1,0ビットのどちらかが設定(1)されると、標準I/Oポート機能は沈黙時間生成器からの比較出力(OC4x/OC4x)によって無効にされます。けれどもOC4x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。OC4xとOC4xピンに対するポート方向レジスタのビット(DDR_OC4xとDDR_OC4x)はOC4xとOC4x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比較出力動作種別と無関係です。

比較出力ピン論理回路の設計は出力が許可される前のOC4x状態の初期化を許します。いくつかのCOM4x1,0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。比較出力ピン構成については97頁の表15-2、98頁の表15-3、99頁の表15-4、と104頁の表15-5を参照してください。

図15-9. 比較一致出力回路図



15.6.1. 比較一致出力選択と波形生成

波形生成器は標準とPWM動作でCOM4x1,0ビットを違うふうに使います。全ての動作種別に対してCOM4x1,0='00'設定は次の比較一致で実行すべきOCW4x出力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106頁の表15-6を参照してください。高速PWM動作については106頁の表15-7、位相基準PWMについては106頁の表15-8を参照してください。

COM4x1,0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動作に対して、この動作は強制変更(FOC4x)スローブビットを使うことによって直ちに効果を得ることを強制できます。

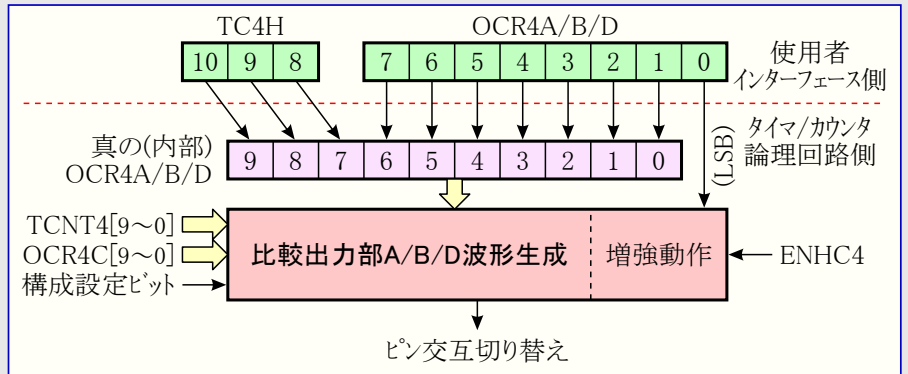
15.6.2. 増強された比較/PWM動作

タイマ/カウンタ4制御レジスタ(TCCR4E)の増強化比較/PWM動作許可(ENHC4)ビットが設定(1)されると、増強化比較/PWM動作が許可されます。この動作は使用者に比較レジスタ(OCR4A, OCR4B, OCR4D)への精度ビット追加を許します。前で説明したように、比較条件は3つの比較レジスタ(OCR4A/B/D)の1つがTCNT4の(10ビット分解能)値と一致する時に現れます。基本PWM動作では対応する許可された出力が比較一致で交互に切り替わります。増強化比較/PWM動作は比較一致状態がどちらの内部クロック端で実際に合図されるのかを決めるビットを導入します。これは対応する出力がOCR4A/B/DのLSBが0の場合に通常クロック端で、またはそのLSBが1の場合に逆(次)端で交互に切り替わることを意味します。

使用者は通常と増強のPWM動作間で、出力周波数が同一である一方、PWM分解能が後者でより良くなることに気付くでしょう。

比較レジスタ(OCR4A/B/D)への書き込みまたはそれらの読み込みは両(PWM)動作種別で同じです。増強動作に於いて、使用者はタイマ/カウンタ4上位バイト(TC4H)レジスタが3ビット幅まででできる(そしてそれが2ビット動作中と同じ動きを持つ)ことを充分考慮しなければなりません。これはOCR4A, OCR4B, OCR4Dのレジスタアクセスだけに限らず、これは出力信号周期を決める分解能に留まります。

図15-10. 増強動作でのレジスタアクセスの働き方



この図は最下位ビットを波形生成部へ直接転送するために、使用者によって格納された値に対応する真(内部)のOCR4A/B/D値が右移動されることを示します。

利用可能な最大分解能は11ビットですが、(それ以下の)他のどの分解能も指定することができます。例えば、(増強動作での)8ビット分解能は7ビット分解能での通常動作と同じ周波数を得ることを可能にします。

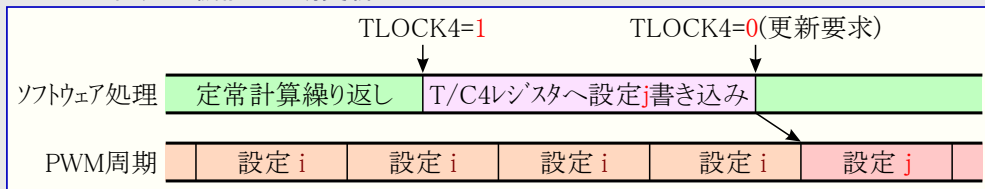
例:

- PLL後置分周器出力=64MHz、タイマ/カウンタ4の前置分周なし。
- OCR4C=\$7F設定は理論上の完全7ビット分解能、故に500kHz出力周波数を決めます。
- OCR4A=\$85('10000101')設定は比較Aレジスタの真の値が\$42('01000010')で、且つ増強ビットの設定(1)を指定します。これは得られるデューティサイクル(51.95%)が、(通常動作に於ける)\$42と\$43の比較値によって得られるデューティサイクル(51.56%, 52.34%)間の中間値であることを意味します。

15.7. 同期更新

周期での同時性のない矛盾した値を避けるため、多くの値の1つの同期更新が必要な場合、タイマ/カウンタ制御器によってPWM周期の最後で全ての値を同時に更新することができます。新しい値一式はソフトウェアによって計算され、ソフトウェアによって効率的な更新を開始することができます。

図15-11. 固定化機能と同期更新



通常動作では各書き込みが現在の周期の最後で有効になります。しかし、或る場合には複数の比較レジスタの同時更新を必要とし、それは高速PWM周波数で大抵、常に可能ではないかもしれません。これは矛盾値でのいくつかのPWM周期に終わるかもしれません。

固定化機能(TLOCK4=1)使用時、比較レジスタに書かれる値は有効にならず、一時的に緩衝されます。TLOCK4ビット開放(=0)時、更新が開始され、新しい値一式全体が現在のPWM周期の最後で格納されます。

111頁の「TCCR4E - タイマ/カウンタ4制御レジスタ」を参照してください。

15.8. 動作種別

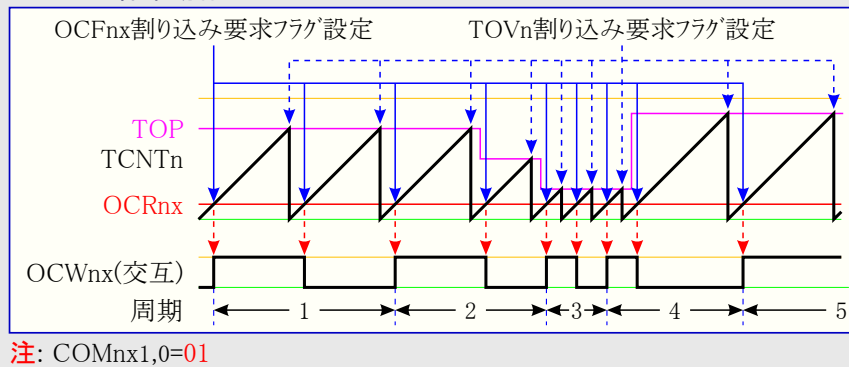
動作種別、換言するとタイマ/カウンタと比較出力ピンの動作は波形生成種別(WGM4x1,0とPWM4xビット)と比較出力選択(COM4x1,0)ビットの組み合わせによって定義されます。比較出力選択ビットは計数順序(動作)に影響を及ぼしませんが、一方波形生成種別ビットは影響を及ぼします。COM4x1,0ビットは生成されるPWM出力が反転されるべきか、されないべきか、または相互補完かのどれかを制御します。非PWM動作に対するCOM4x1,0ビットは比較一致で出力が解除(0)、設定(1)、1/0交互のどれにされるべきかを制御します。

15.8.1. 標準動作

最も単純な動作種別が標準動作(PWM4x=0)で、カウンタはBOTTOMから(OCR4Cとして定義した)TOPへ計数し、そしてBOTTOMから再び始めます。OCR4Cはカウンタに対するTOP値、従って分解能も定義し、比較一致出力周波数の制御を許します。交互切り替え比較出力動作での波形出力(OCW4x)はTCNT4とOCR4x間の比較一致で交互切り替えされます。非反転比較出力動作(COM4x1,0=10)での波形出力は比較一致で解除(0)されます。反転比較出力動作(COM4x1,0=11)での波形出力は比較一致で設定(1)されます。

標準動作に対するタイミング図は図15-12.で示されます。このタイミング図で折れ線として示されるカウンタ値(TCNT4)はカウンタ値はTOP値に達するまで増加(+1)されます。そしてカウンタは次のクロック周期で解除されます。この図は交互切り替え比較出力動作での波形出力(OCW4x)を含みます。細い赤線はOCR4x値を示し、TCNT4値との交点(接点)がTCNT4とOCR4x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)。

図15-12. 標準動作タイミング



注: COMnx1,0=01

タイマ/カウンタ溢れ(TOV4)フラグはTCNT4が\$0000になるのと同じタイマ/カウンタクロック周期で設定(1)されます。この場合のTOV4フラグは設定(1)のみで解除(0)されないことを除いて第11ビットのようになります。けれどもTOV4フラグを自動的に解除(0)するタイマ/カウンタ4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に推奨されません。波形生成に関して、OCW4x出力は比較出力選択(COM4x1,0)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。OC4x値はそのピンに対するデータ方向が出力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OCR4Cが0(\$000)に設定される時に $f_{OC4x} = f_{clkT4} / 2$ 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。

$$f_{OC4x} = \frac{f_{clkT4}}{2 \times (1 + OCR4C)}$$

分解能はOCR4C内の表現するのに何ビット必要とされるかを示します。これは次式によって計算されます。

$$\text{分解能 PWM} = \log_2(OCR4C + 1)$$

標準動作での比較出力ピン構成は下表で記述されます。

表15-2. 標準動作比較出力ピン構成

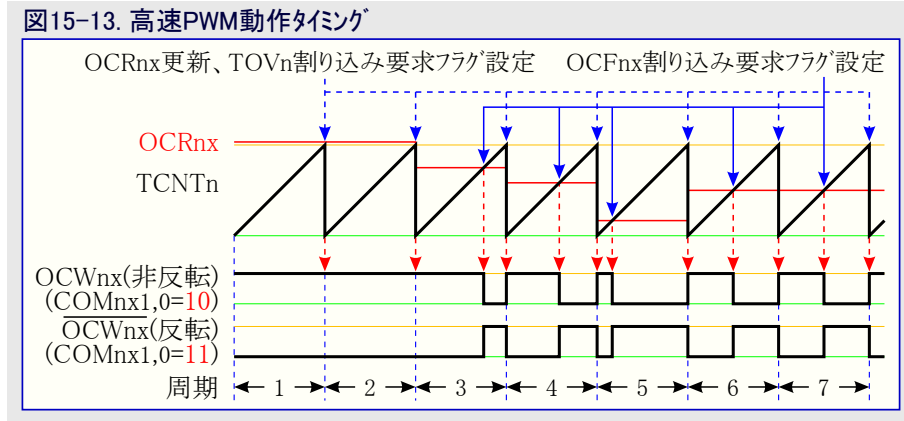
COM4x1	COM4x0	OC4xピン	$\overline{OC4x}$ ピン
0	0	標準ポート動作 (OC4x切断)	標準ポート動作 ($\overline{OC4x}$ 切断)
0	1	OC4x	標準ポート動作 ($\overline{OC4x}$ 切断)
1	0	OC4x	標準ポート動作 ($\overline{OC4x}$ 切断)
1	1	OC4x	標準ポート動作 ($\overline{OC4x}$ 切断)

15.8.2. 高速PWM動作

高速パルス幅変調または高速PWM動作(WGM41,0=00とPWM4x=1)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)動作であることで他のPWM動作と異なります。カウンタはBOTTOMから(OCR4Cとして定義された)TOP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(COM4x1,0=10)での波形出力(OCW4x)はTCNT4とOCR4x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(COM4x1,0=11)の出力は比較一致で設定(1)され、BOTTOMで解除(0)されます。相互補完比較出力動作(COM4x1,0=01)での波形出力は比較一致で解除(0)され、BOTTOMで設定(1)されます。

この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/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動作に関するタイミング図は図15-13.で示されます。カウンタはカウンタ値がTOP値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(\$000)されます。TCNT4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波形出力を含みます。細い赤線はOCR4x値を示し、TCNT4値との交点(接点)がTCNT4とOCR4x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(TOV4)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。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OC4xピンでのPWM波形の生成を許します。COM4x1,0ビットの'10'設定は非反転PWM出力を作成し、COM4x1,0の'11'設定は反転PWM出力を生成します。COM4x1,0ビットの'01'設定は相互補完比較出力動作を許可し、非反転(OC4x)と反転(OC4x)の両出力を生成します。実際の値はポートピンに対するデータ方向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TCNT4とOCR4x間の比較一致で、OCW4xを設定(1)(または解除(0))することによって生成されます。

PWM出力周波数は次式によって計算できます。

$$f_{OCnxPWM} = \frac{f_{clkT4}}{N}$$

変数Nは単一傾斜動作での段数を表します。Nの値はTOP値と同じです。

OCR4C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。OCR4CがBOTTOM(\$000)に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。OCR4CがMAXに等しく設定されると、(COM4x1,0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。

高速PWM動作での(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波形出力(OCW4x)設定(COM4x1,0=01)によって達成できます。生成された波形はOCR4Cが3(\$003)に設定される時に $f_{OC4x} = f_{clkT4}/4$ の最大周波数でしょう。

COM4x1,0ビットのどちらかが設定(1)され、OC4xとOC4xのピンに対するデータ方向レジスタビットが出力として設定されている場合、標準I/Oポート機能は沈黙時間生成器からの比較出力値(OC4x/OC4x)によって無効にされます。COM4x1,0ビットが解除(00)なら、ポート出力レジスタからの実際の値がポートピンで見えるでしょう。比較出力ピン構成は下表で記述されます。

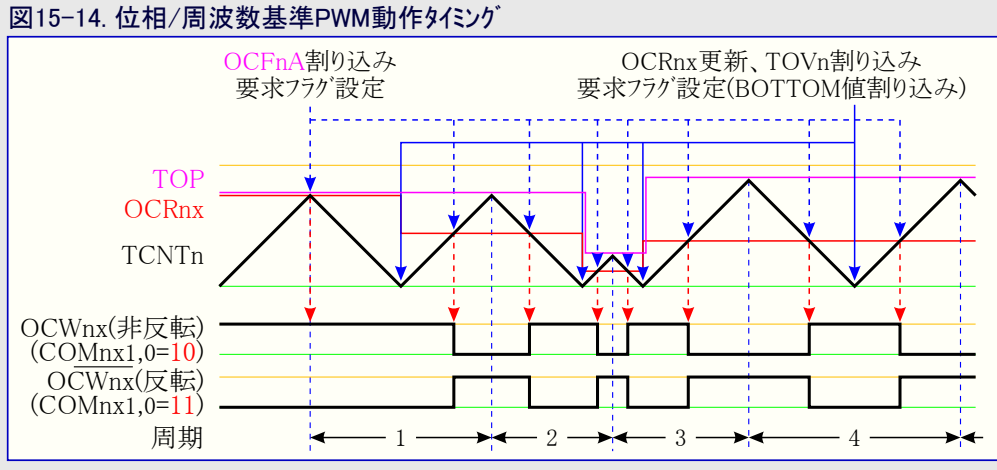
表15-3. 高速PWM動作比較出力ピン構成

COM4x1	COM4x0	OC4xピン	OC4xピン
0	0	標準ポート動作 (OC4x切断)	標準ポート動作 (OC4x切断)
0	1	OC4x	OC4x
1	0	OC4x	標準ポート動作 (OC4x切断)
1	1	OC4x	標準ポート動作 (OC4x切断)

15.8.3. 位相/周波数基準PWM動作

位相/周波数基準PWM動作(WGM41,0=01とPWM4x=1)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)動作が基準です。カウンタはBOTTOM(\$000)から(OCR4Cとして定義した)TOPへ、そしてその後TOPからBOTTOMへを繰り返し計数します。非反転比較出力動作(COM4x1,0=10)と相互補完比較出力動作(COM4x1,0=01)での波形出力(OCW4x)は上昇計数中のTCNT4とOCR4x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(COM4x1,0=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動作の対称特性なのでこれらの動作種別は電動機制御の応用に好まれます。TCNT4値は1タイマ/カウンタクロック周期間TOPと等しくなります。

TCNT4値が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る、位相/周波数基準PWM動作のタイミング図は図15-14.で示されます。カウンタはカウンタ値がTOPと一致するまで増加されます。カウンタがTOPに到達した時に計数方向を変更します。この図は非反転と反転での波形出力(OCW4x)を含みます。細い赤線はOCR4x値を示し、TCNT4値との交点(接点)がTCNT4とOCR4x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(TOV4)フラグはカウンタがBOTTOMに達する時毎に設定(1)されます。割り込み要求フラグはカウンタがBOTTOMに達する時毎に割り込みを発生するのに使えます。

位相/周波数基準PWM動作での比較部はOC4xピンでのPWM波形の生成を許します。COM4x1,0ビットの'10'設定は非反転PWMを作成し、COM4x1,0の'11'設定は反転PWM出力を生成します。COM4x1,0ビットの'01'設定は相互補完比較出力動作を許可し、非反転(OC4x)と反転(OC4x)の両出力を生成します。実際の値はポートピンに対するデータ方向レジスタビットが出力として設定される場合だけ見えます。PWM波形はカウンタが増加する時のTCNT4とOCR4x間の比較一致で波形出力(OCW4x)を設定(1)(または解除(0))と、カウンタが減少する時のTCNT4とOCR4x間の比較一致で波形出力を解除(0)(または設定(1))することによって生成されます。位相/周波数基準PWMを使う時の出力に対するPWM周波数は次式によって計算できます。

$$f_{OCnxPFCPWM} = \frac{f_{clkT4}}{N}$$

変数Nは両傾斜動作での段数を表します。Nの値はTOP値と同じです。

OCR4Cに対する両端値は位相/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OCR4x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。

COM4x1,0ビットのどちらかが設定(1)され、OC4xとOC4xのピンに対するデータ方向レジスタビットが出力として設定されている場合、標準I/Oポート機能は沈黙時間生成器からの比較出力値(OC4x/OC4x)によって無効にされます。COM4x1,0ビットが解除(00)なら、ポート出力レジスタからの実際の値がポートピンで見えます。比較出力ピン構成は下表で記述されます。

表15-4. 位相/周波数基準PWM動作比較出力ピン構成

COM4x1	COM4x0	OC4xピン	OC4xピン
0	0	標準ポート動作 (OC4x切断)	標準ポート動作 (OC4x切断)
0	1	OC4x	OC4x
1	0	OC4x	標準ポート動作 (OC4x切断)
1	1	OC4x	標準ポート動作 (OC4x切断)

15.8.4. PWM6動作

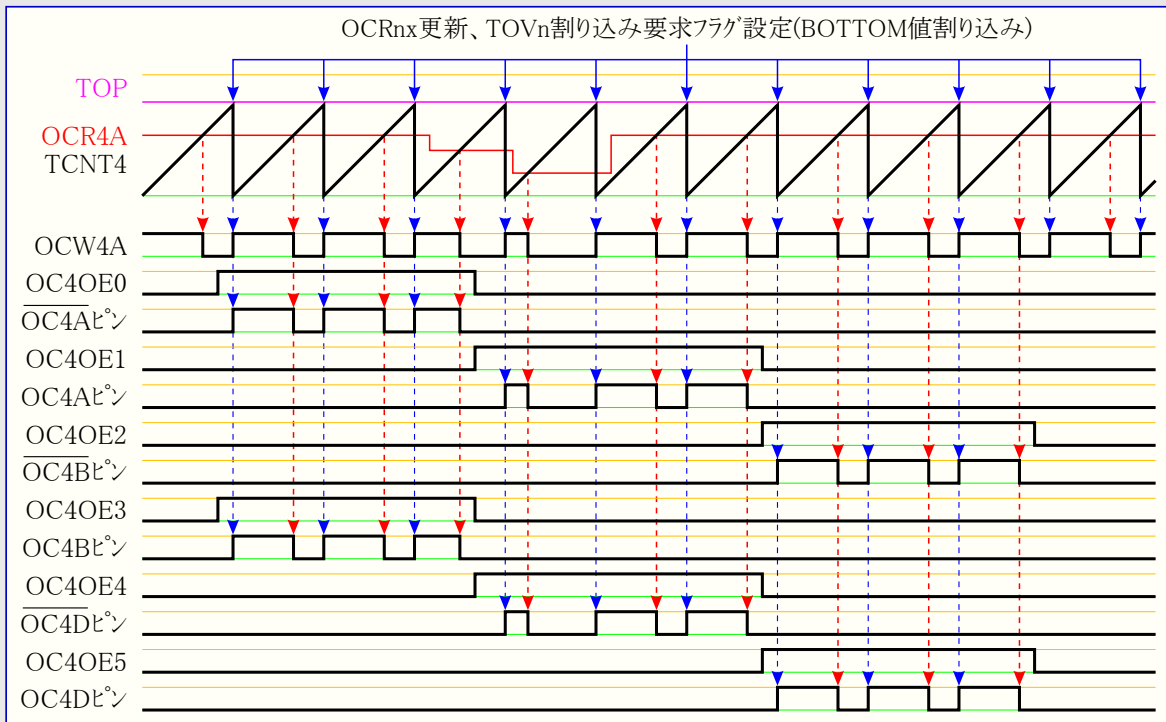
PWM6動作(WGM41,0=1xとPWM4A=1)は例えばブラシレスDC電動機用のPWM波形生成選択を提供します。WPM6動作でのOCR4Aは全波形を生成するのに使われる波形生成器からの同じ波形出力(OCW4A)として6つ全ての比較出力波形を制御します。PWM6動作は比較出力ピンを許可/禁止するための瞬時応答で使える**比較出力での無効化許可(OC4OEm:m=0~5)ビット**も提供します。比較出力での無効化許可ビットが解除(0)なら、ポート出力レジスタからの実際の値がポートピンで見えるでしょう。

PWM6動作は単一傾斜動作と両傾斜動作の2つのカウンタ操作種別を提供します。単一傾斜動作が選択(WGM40=0)されるなら、カウンタは**高速PWM動作**のようにBOTTOMから(OCR4Cとして定義された)TOPまで計数し、その後BOTTOMから再び始めます。PWM波形はTCNT4とOCR4A間の比較一致で波形出力(OCW4A)を解除(0)(または設定(1))と、カウンタが解除(\$000、TOPからBOTTOMへ変更)されるタイマ/カウンタクロック周期で波形出力を設定(1)(または解除(0))することによって生成されます。**タイマ/カウンタ溢れ(TOV4)フラグ**はカウンタがTOPに到達する時毎に設定(1)され、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。

ところが両傾斜動作が選択(WGM40=1)される場合、カウンタは位相/周波数基準PWM動作のようにBOTTOM(\$000)から(OCR4Cとして定義した)TOPへ、そしてその後TOPからBOTTOMへを繰り返して計数します。PWM波形は上昇計数中のTCNT4とOCR4A間の比較一致で波形出力(OCW4A)を解除(0)(または設定(1))と、下降計数中の比較一致で波形出力を設定(1)(または解除(0))することによって生成されます。タイマ/カウンタ溢れ(TOV4)フラグはカウンタがBOTTOMに到達する時毎に設定(1)され、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。

COM4x1,0ビットが'10'に設定される時の単一傾斜動作(WGM40=0)でのPWM6動作に関するタイミング図は図15-15で示されます。カウンタはカウンタ値がTOP値と一致するまで増加(+1)されます。そしてカウンタは次のタイマ/カウンタクロック周期で解除(\$000)されます。TCNT4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OC4AとOC4Aの比較出力ピンと、対応する比較出力有効許可(OC4OE1,OC4OE0)ビットを含みます。

図15-15. PWM6単一傾斜動作タイミング



注: COM4x1,0=10,非反転動作

COM4x1,0ビットのどちらかが設定(1)されている場合、標準I/Oポート機能は**沈黙時間生成器**からの比較出力値(OC4x/OC4x)によって無効にされます。比較出力ピンは比較出力での無効化許可(OC4OE5~OC4OE0)ビットによっても無効にできます。無効化許可ビットが解除(0)なら、ポート出力レジスタからの実際の値がポートピンで見え、無効化許可ビットが設定(1)なら、比較出力ピンはポートピンへの接続が許されます。比較出力ピン構成は表15-5で記述されます。

表15-5. PWM6動作比較出力ピン構成

COM4A1	COM4A0	OC4Aピン (PC7)	$\overline{\text{OC4A}}$ ピン (PC6)
0	0	標準ポート動作 (OC4A切断)	標準ポート動作 (OC4A切断)
0	1	OC4A・OC4OE1	$\overline{\text{OC4A}}$ ・OC4OE0
1	0	OC4A・OC4OE1	OC4A・OC4OE0
1	1	OC4A・OC4OE1	OC4A・OC4OE0
COM4B1	COM4B0	OC4Bピン (PB6)	$\overline{\text{OC4B}}$ ピン (PB5)
0	0	標準ポート動作 (OC4A切断)	標準ポート動作 (OC4A切断)
0	1	OC4A・OC4OE3	$\overline{\text{OC4A}}$ ・OC4OE2
1	0	OC4A・OC4OE3	OC4A・OC4OE2
1	1	OC4A・OC4OE3	OC4A・OC4OE2
COM4D1	COM4D0	OC4Dピン (PD7)	$\overline{\text{OC4D}}$ ピン (PD6)
0	0	標準ポート動作 (OC4A切断)	標準ポート動作 (OC4A切断)
0	1	OC4A・OC4OE5	$\overline{\text{OC4A}}$ ・OC4OE4
1	0	OC4A・OC4OE5	OC4A・OC4OE4
1	1	OC4A・OC4OE5	OC4A・OC4OE4

(訳補) PWM6動作は基本的に高速PWM動作と位相/周波数基準PWM動作に対して波形出力にOCW4Aを共通で使い、各比較出力ピンを比較出力での無効化許可ビットによって個別に許可/禁止できるようにしたものと考えられます。

15.9. 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_{T4})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15-16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/周波数基準PWM動作以外の全ての動作種別でのTOP値近辺の計数の流れを示します。図15-17は位相/周波数基準PWM動作と両傾斜PWM6動作以外の全ての動作種別での同じタイミング データを示しますが、前置分周器が許可されています。図15-18は全動作種別でのOCF4A, OCF4B, OCF4Dの設定を示します。図15-19は位相/周波数基準PWM動作と両傾斜PWM6動作でのTOV4の設定を示します。

図15-16. 前置分周なし($1/1$)のタイマ/カウンタ タイミング

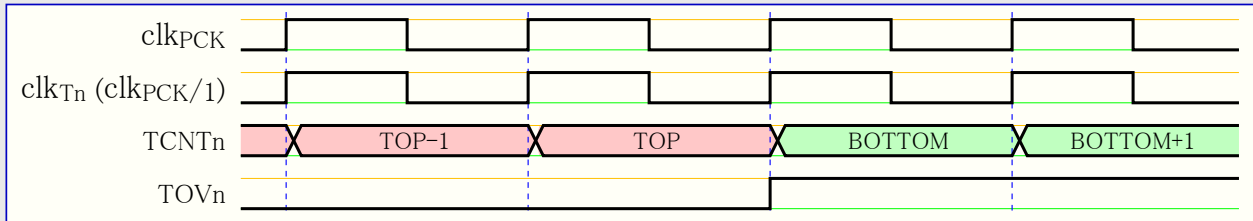


図15-17. 前置分周器($f_{\text{clk}_{PCK}}/8$)のタイマ/カウンタ タイミング (位相/周波数基準PWM動作と両傾斜PWM6動作以外)

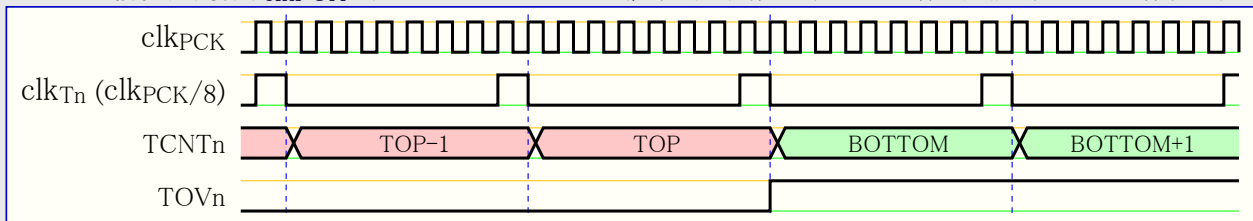


図15-18. 前置分周器($f_{\text{clk}_{PCK}}/8$)のタイマ/カウンタ、OCF1x設定 タイミング

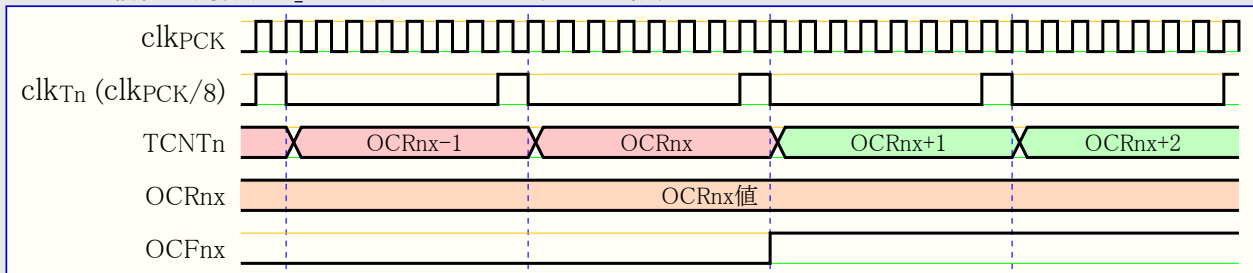
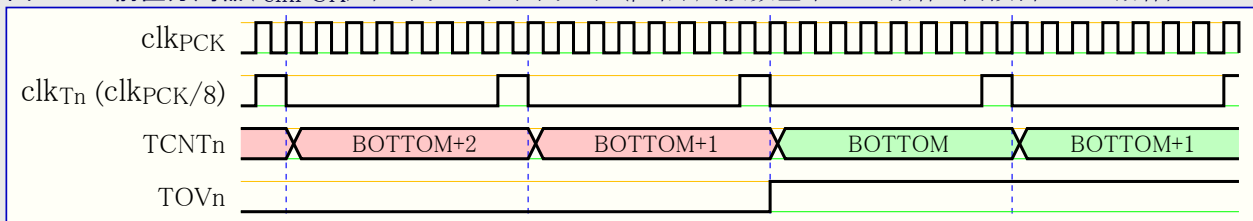


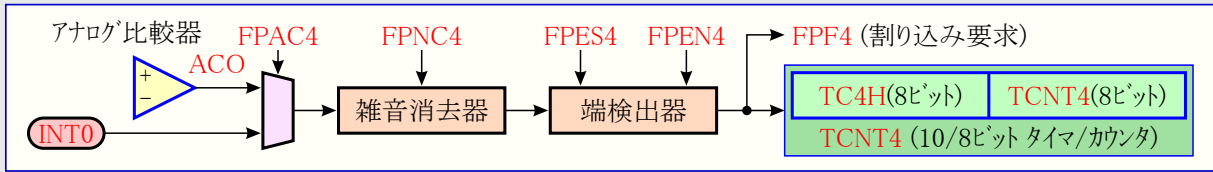
図15-19. 前置分周器($f_{\text{clk}_{PCK}}/8$)のタイマ/カウンタ タイミング (位相/周波数基準PWM動作と両傾斜PWM6動作)



15.10. 障害保護部

タイマ/カウンタ4は外部事象が起動される場合にPWM出力ピンを禁止にできる障害保護部を併設します。出来事を指示する外部信号は外部割り込みINT0経由、または代わりにアナログ比較器部経由で印加できます。障害保護部は図15-20.で示される構成図によって表されます。障害保護部の直接の部分でない構成図の要素は青枠(訳注:原文は灰色背景)で示されます。

図15-20. 障害保護部構成図



障害保護動作は障害保護許可(FPEN4)ビットと外部割り込み(INT0)ピン、または代わりにアナログ比較器出力で論理レベル(事象)の変化によって許可され、この変化が端(エッジ)検出器の設定に通知して障害保護が起動されます。障害保護が起動されると、比較出力選択(COM4x1.0)ビットが解除(0)され、比較出力はPWM出力ピンから切断されてPORTxレジスタビットがPWM出力ピンに接続されます。障害保護許可(FPEN4)はCOM4xビットが解除(0)されるのと同じシステムクロックで自動的に解除(0)されます。障害保護割り込み許可(FPIE4)が設定(1)なら、障害保護割り込みが生成され、FPEN4ビットが解除(0)されます。代わりにFPEN4はタイマ/カウンタが障害保護動作へ移行されてしまった時を理解するのにソフトウェアによってポーリングすることもできます。

15.10.1. 障害保護起動元

障害保護部に対する主な起動元は外部割り込み(INT0)ピンです。代わりにアナログ比較器出力が障害保護部に対する起動元として使えます。アナログ比較器はタイマ/カウンタ4制御レジスタD(TCCR4D)の障害保護アナログ比較器(FPAC4)ビットの設定(1)によって選ばれます。起動元変更が障害保護動作を起動し得ることに注意してください。従って起動元変更、端検出器設定、または障害保護許可後に障害保護割り込み要求フラグ(FPF4)を解除(0)することが推奨されます。

外部割り込み(INT0)ピンとアナログ比較器出力(ACO)入力はTnピン(55ページの図12-1.参照)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。けれども雑音消去が許可されると、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。障害保護動作はINT0ピンのポートを制御することによってソフトウェアでも起動できます。

15.10.2. 雑音消去器

雑音消去器は簡単なデジタル濾波器機構を使うことによって雑音耐性を改善します。雑音消去器の入力は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には4回全てが同じでなければなりません。

雑音消去器はタイマ/カウンタ4制御レジスタD(TCCR4D)の障害保護雑音消去許可(FPNC4)ビットの設定(1)によって許可されます。許可した時に雑音消去器は入力に印加した変更から4システムクロック周期の追加遅延をもたらします。雑音消去器はシステムクロックを使い、従って前置分周器によって影響を及ぼされません。

15.11. 10ビットレジスタのアクセス

10ビット値がTCNT4とOCR4A/B/C/Dレジスタに書かれる場合、その10ビットレジスタは2回の読みまたは書き操作を使って8ビットバス経由でAVR CPUによってアクセスできます。10ビットレジスタは10ビットアクセスの上位2ビットの一時保存用に共通2ビットのタイマ/カウンタ1上位バイト(TC4H)レジスタを持ちます。全ての10ビットレジスタ間で、この同じTC4Hレジスタが共用されます。下位バイトアクセスが10ビット読み書き動作を起動します。10ビットレジスタの下位バイトがCPUによって書かれると、TC4Hレジスタに格納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10ビットレジスタに両方が複写されます。10ビットレジスタの下位バイトがCPUによって読まれると、10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TC4Hレジスタに複写されます。

10ビット書き込みを行うために上位バイトは下位バイトが書かれるのに先立ってTC4Hレジスタへ書かれなければなりません。10ビット読み込みについては下位バイトが上位バイトに先立って読まれなければなりません。

次のコード例は割り込みがTC4Hレジスタを更新しないことが前提の10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。OCR4A/B/C/D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。

アセンブリ言語プログラム例

```
~                               ;
                                ;[10ビット($1FF)書き込み]
LDI    R17, $01                ;$1FFの上位バイト値取得
LDI    R16, $FF                ;$1FFの下位バイト値取得
OUT     TCnH, R17              ;上位バイト設定(TCnHレジスタ)
OUT     TCNTn, R16             ;下位バイト設定(TCnH⇒上位バイト)
~                               ;[10ビット読み込み]
IN      R16, TCNTn              ;下位バイト取得(上位バイト⇒一時レジスタ)
IN      R17, TCnH              ;上位バイト取得(TCnHレジスタ)
~                               ;
```

C言語プログラム例

```
unsigned int i;
~                               /**/
                                /* [10ビット($1FF)書き込み] */
TCnH = 0x01;                   /* 上位バイト設定(TCnHレジスタ) */
TCNTn = 0xFF;                  /* 下位バイト設定(TCnHレジスタ⇒上位バイト) */
~                               /**/
i = TCNTn;                     /* [10ビット読み込み] */
i |= ((unsigned int)TCnH<<8);  /* 下位バイト取得(上位バイト⇒TCnHレジスタ) */
~                               /* 上位バイト取得(TCnHレジスタ)合成 */
                                /**/
```

注: 5頁の「コード例」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TCNT4値を戻します。

10ビットレジスタアクセスが非分断操作であるのに注意することが重要です。10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10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TC4H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TC4Hレジスタを更新する時に主コードは10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。

訳補: 増強化比較/PWM動作時のOCR4A/B/Dに対する11ビットアクセスも同様で、TC4Hの下位3ビットが11ビット値の上位3ビットを含みます。従って、その場合の本章に於いては10ビットを11ビット、TC4Hの2ビットを3ビットと読み換えてください。

次のコード例はTCNT4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、OCR4A/B/C/Dのどんな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, TCNTn        ; TCNTn下位バイト取得(上位バイト⇒TCnHレジスタ)
             IN      R17, TCnH        ; TCNTn上位バイト取得(TCnH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 TIM10_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 */
    _CLI();                           /* 全割り込み禁止 */
    i = TCNTn;                        /* 下位バイト取得(上位バイト⇒TCnHレジスタ) */
    i |= ((unsigned int)TCnH<<8);     /* 上位バイト取得(TCnHレジスタ)合成 */
    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: 5頁の「コード例」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TCNT4値を戻します。

次のコード例はTCNT4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、OCR4A/B/C/Dのどんな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     TCnH, R17         ; TCNTn上位バイト設定(TCnHレジスタ)
             OUT     TCNTn, R16        ; TCNTn下位バイト設定(TCnH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 TIM10_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 */
    _CLI();                           /* 全割り込み禁止 */
    TCnH = (i>>8);                   /* 上位バイト設定(TCnHレジスタ) */
    TCNTn = (unsigned char)i;         /* 下位バイト設定(TCnHレジスタ⇒上位バイト) */
    SREG = sreg;                      /* 全割り込み許可フラグ(I)を復帰 */
}
```

注: 5頁の「コード例」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TCNT4へ書かれるべき値を含む必要があります。

15.11.1. 上位バイト一時レジスタの再使用

書かれる全レジスタについて上位バイトが同じ複数10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。

15.12. タイマ/カウンタ4用レジスタ

15.12.1. TCCR4A – タイマ/カウンタ4制御レジスタA (Timer/Counter4 Control Register A)

ビット (\$C0)	7	6	5	4	3	2	1	0	
	COM4A1	COM4A0	COM4B1	COM4B0	FOC4A	FOC4B	PWM4A	PWM4B	TCCR4A
Read/Write	R/W	R/W	R/W	R/W	W	W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

● ビット7,6 – COM4A1,0 : 比較A出力選択 (Compare A Output Mode bits 1 and 0)

これらのビットは波形出力(OCW4A)の動きとOC4A比較出力ピンの接続を制御します。COM4A1,0ビットの1つまたは両方が1を書かれると、OC4A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。相互補間OC4A出力はCOM4A1,0ビットが'01'に設定される時のPWM動作でだけ接続されます。けれども出力駆動部を許可するのにOC4AとOC4A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。

COM4A1,0ビットの機能はPWM4AとWGM41,0ビット設定に依存します。

右表はPWM4Aビットが標準動作(つまり非PWM)に設定される時のCOM4A1,0ビットの機能を示します。

表15-6. 標準(非PWM)動作比較A出力選択

COM4A1,0	OCW4Aの動き	OC4Aピン	OC4Aピン
0 0	標準ポート動作	切断	切断
0 1	比較一致でトグル(交互切り替え)出力	接続	切断
1 0	比較一致で解除(0)	接続	切断
1 1	比較一致で設定(1)	接続	切断

右表はPWM4AとWGM41,0ビットが高速PWM動作に設定される時のCOM4A1,0ビットの機能を示します。

表15-7. 高速PWM動作比較A出力選択

COM4A1,0	OCW4Aの動き	OC4Aピン	OC4Aピン
0 0	標準ポート動作	切断	切断
0 1	比較一致で解除(0)、TCNT4=\$000で設定(1)	接続	接続
1 0	比較一致で解除(0)、TCNT4=\$000で設定(1)	接続	切断
1 1	比較一致で設定(1)、TCNT4=\$000で解除(0)	接続	切断

右表はPWM4AとWGM41,0ビットが位相/周波数基準PWM動作に設定される時のCOM4A1,0ビットの機能を示します。

表15-8. 位相/周波数基準PWM動作比較A出力選択

COM4A1,0	OCW4Aの動き	OC4Aピン	OC4Aピン
0 0	標準ポート動作	切断	切断
0 1	上昇計数時の比較一致で解除(0)、下降計数時の比較一致で設定(1)	接続	接続
1 0	上昇計数時の比較一致で解除(0)、下降計数時の比較一致で設定(1)	接続	切断
1 1	上昇計数時の比較一致で設定(1)、下降計数時の比較一致で解除(0)	接続	切断

下表はPWM4AとWGM41,0ビットが単一傾斜PWM6動作に設定される時のCOM4A1,0ビットの機能を示します。PWM6動作では全ての波形生成に波形出力(OCW4A)が使われ、以下で記述されるようにOC4AとOC4Aが全てのOC4xとOC4xピンに接続されます。

表15-9. 単一傾斜PWM6動作比較x出力選択

COM4A1,0	OCW4Aの動き	OC4xピン	OC4xピン
0 0	標準ポート動作	切断	切断
0 1	比較一致で解除(0)、TCNT4=\$000で設定(1)	OC4A	OC4A
1 0	比較一致で解除(0)、TCNT4=\$000で設定(1)	OC4A	OC4A
1 1	比較一致で設定(1)、TCNT4=\$000で解除(0)	OC4A	OC4A

右表はPWM4AとWGM41,0ビットが両傾斜PWM6動作に設定される時のCOM4A1,0ビットの機能を示します。

表15-10. 両傾斜PWM6動作比較x出力選択

COM4A1,0	OCW4Aの動き	OC4xピン	OC4xピン
0 0	標準ポート動作	切断	切断
0 1	上昇計数時の比較一致で解除(0)、下降計数時の比較一致で設定(1)	OC4A	OC4A
1 0	上昇計数時の比較一致で解除(0)、下降計数時の比較一致で設定(1)	OC4A	OC4A
1 1	上昇計数時の比較一致で設定(1)、下降計数時の比較一致で解除(0)	OC4A	OC4A

●ビット5,4 – COM4B1,0 : 比較B出力選択 (Compare B Output Mode bits 1 and 0)

これらのビットは波形出力(OCW4B)の動きとOC4B比較出力ピンの接続を制御します。COM4B1,0ビットの1つまたは両方が1を書かれると、OC4B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。相互補間OC4B出力はCOM4B1,0ビットが'01'に設定される時のPWM動作でだけ接続されます。けれども出力駆動部を許可するのにOC4AとOC4B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。

COM4B1,0ビットの機能はPWM4BとWGM41,0ビット設定に依存します。

右表はPWM4Bビットが標準動作(つまり非PWM)に設定される時のCOM4B1,0ビットの機能を示します。

表15-11. 標準(非PWM)動作比較B出力選択

COM4B1,0	OCW4Bの動き	OC4Bピン	OC4Bピン
0 0	標準ポート動作	切断	切断
0 1	比較一致でトル(交互切り替え)出力	接続	切断
1 0	比較一致で解除(0)	接続	切断
1 1	比較一致で設定(1)	接続	切断

右表はPWM4BとWGM41,0ビットが高速PWM動作に設定される時のCOM4B1,0ビットの機能を示します。

表15-12. 高速PWM動作比較B出力選択

COM4B1,0	OCW4Bの動き	OC4Bピン	OC4Bピン
0 0	標準ポート動作	切断	切断
0 1	比較一致で解除(0)、TCNT4=\$000で設定(1)	接続	接続
1 0	比較一致で解除(0)、TCNT4=\$000で設定(1)	接続	切断
1 1	比較一致で設定(1)、TCNT4=\$000で解除(0)	接続	切断

右表はPWM4BとWGM41,0ビットが位相/周波数基準PWM動作に設定される時のCOM4B1,0ビットの機能を示します。

表15-13. 位相/周波数基準PWM動作比較B出力選択

COM4B1,0	OCW4Bの動き	OC4Bピン	OC4Bピン
0 0	標準ポート動作	切断	切断
0 1	上昇計数時の比較一致で解除(0)、下降計数時の比較一致で設定(1)	接続	接続
1 0	上昇計数時の比較一致で解除(0)、下降計数時の比較一致で設定(1)	接続	切断
1 1	上昇計数時の比較一致で設定(1)、下降計数時の比較一致で解除(0)	接続	切断

●ビット3 – FOC4A : 比較A強制変更 (Force Output Compare Match 4A)

FOC4AビットはPWM4Aビットが非PWM動作を指示する時だけ有効です。

このビットへの1書き込みは波形出力(OCW4A)と既にCOM4A1,0ビットで設定した値に従って比較出力(OC4A, OC4A)ピンの変更を強制します。COM4A1,0をFOC4Aと同じ周期で書いた場合、新しい設定が使われます。比較出力強制ビットはタイマ/カウンタ値と無関係に出力ピンを変更するのに使えます。COM4A1,0で設定した自動動作は比較一致が起こった場合のように実行しますが、割り込みは生成されません。

FOC4Aビットは常に0として読みます。

●ビット2 – FOC4B : 比較B強制変更 (Force Output Compare Match 4B)

FOC4BビットはPWM4Bビットが非PWM動作を指示する時だけ有効です。

このビットへの1書き込みは波形出力(OCW4B)と既にCOM4B1,0ビットで設定した値に従って比較出力(OC4B, OC4B)ピンの変更を強制します。COM4B1,0をFOC4Bと同じ周期で書いた場合、新しい設定が使われます。比較出力強制ビットはタイマ/カウンタ値と無関係に出力ピンを変更するのに使えます。COM4B1,0で設定した自動動作は比較一致が起こった場合のように実行しますが、割り込みは生成されません。

FOC4Bビットは常に0として読みます。

●ビット1 – PWM4A : PWM-A許可 (Pulse width Moduration A Enable)

設定(1)時、このビットはOCR4A比較器を基にしたPWM動作を許可します。

●ビット0 – PWM4B : PWM-B許可 (Pulse width Moduration B Enable)

設定(1)時、このビットはOCR4B比較器を基にしたPWM動作を許可します。

15.12.2. TCCR4B – タイマ/カウンタ4制御レジスタB (Timer/Counter4 Control Register B)

ビット (\$C1)	7	6	5	4	3	2	1	0	TCCR4B
	PWM4X	PSR4	DTPS41	DTPS40	CS43	CS42	CS41	CS40	
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 – PWM4X : PWM反転動作 (PWM Inversion Mode)

このビットが設定(1)されると、PWM反転動作が選ばれ、沈黙時間生成器出力、OC4xとOC4xが反転されます。

●ビット6 – PSR4 : タイマ/カウンタ4前置分周器リセット (Prescaler Reset Timer/Counter4)

このビットが設定(1)されると、タイマ/カウンタ4の前置分周器がリセットされます(TCNT4は影響されません)。このビットはこの動作完了後にハードウェアによって解除(0)されます。このビットへの0書き込みは無効です。このビットは常に0として読みます。

●ビット5,4 – DTPS41,0 : 沈黙時間前置分周器選択 (Dead Time Prescaler Bits)

沈黙時間生成器前の沈黙時間専用前置分周器はタイマ/カウンタ4 クロック(CKまたはPCK)を1,2,4,8分周でき、生成され得る広範囲な沈黙時間を提供します。この前置分周器はタイマ/カウンタ4制御レジスタB(TCCR4B)からのDTPS41,DTPS40の2つの制御ビットによって制御されます。これらのビットは沈黙時間前置分周器の分周値を定義します。この分周値は下表で与えられます。

表15-14. 沈黙時間前置分周器の分周値

DTPS41	DTPS40	タイマ/カウンタ4 クロック分周数
0	0	1 (分周なし)
0	1	2分周
1	0	4分周
1	1	8分周

●ビット3~0 – CS43~0 : クロック選択4 (Clock Select Bits 3,2,1 and 0)

このクロック選択ビット3~0はタイマ/カウンタ4の前置分周元を定義します。

表15-15. タイマ/カウンタ4 前置分周クロック選択

CS43	CS42	CS41	CS40	意味		CS43	CS42	CS41	CS40	意味	
				非同期動作	同期動作					非同期動作	同期動作
0	0	0	0	停止 (タイマ/カウンタ4動作停止)		1	0	0	0	PCK/128	CK/128
0	0	0	1	PCK	CK(CPUクロック)	1	0	0	1	PCK/256	CK/256
0	0	1	0	PCK/2	CK/2	1	0	1	0	PCK/512	CK/512
0	0	1	1	PCK/4	CK/4	1	0	1	1	PCK/1024	CK/1024
0	1	0	0	PCK/8	CK/8	1	1	0	0	PCK/2048	CK/2048
0	1	0	1	PCK/16	CK/16	1	1	0	1	PCK/4096	CK/4096
0	1	1	0	PCK/32	CK/32	1	1	1	0	PCK/8192	CK/8192
0	1	1	1	PCK/64	CK/64	1	1	1	1	PCK/16384	CK/16384

停止状態はタイマ/カウンタの許可/禁止機能を提供します。

15.12.3. TCCR4C – タイマ/カウンタ4制御レジスタC (Timer/Counter4 Control Register C)

ビット (\$C2)	7	6	5	4	3	2	1	0	
	COM4A1S	COM4A0S	COM4B1S	COM4B0S	COM4D1	COM4D0	FOC4D	PWM4D	TCCR4C
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	W	R/W	
初期値	0	0	0	0	0	0	0	0	

●ビット7,6 – COM4A1S,COM4A0S : 比較A出力選択 (Compare A Output Mode bits 1 and 0)

これらのビットは73頁の「タイマ/カウンタ4制御レジスタA(TCCR4A)」項で記述されたCOM4A1,0ビットの投影ビットです。

●ビット5,4 – COM4B1S,COM4B0S : 比較B出力選択 (Compare B Output Mode bits 1 and 0)

これらのビットは106頁の「TCCR4A – タイマ/カウンタ4制御レジスタA」項で記述されたCOM4B1,0ビットの投影ビットです。

●ビット3,2 – COM4D1,0 : 比較D出力選択 (Compare D Output Mode bits 1 and 0)

これらのビットは波形出力(OCW4D)の動きとOC4D比較出力ピンの接続を制御します。COM4D1,0ビットの1つまたは両方が1を書かれると、OC4D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。相互補間OC4D出力はCOM4D1,0ビットが'01'に設定される時のPWM動作でだけ接続されます。けれども出力駆動部を許可するのにOC4DとOC4D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。

COM4D1,0ビットの機能はPWM4DとWGM41,0ビット設定に依存します。

右表はPWM4Dビットが標準動作(つまり非PWM)に設定される時のCOM4D1,0ビットの機能を示します。

表15-16. 標準(非PWM)動作比較D出力選択

COM4D1,0	OCW4Dの動き	OC4Dピン	OC4Dピン
0 0	標準ポート動作	切断	切断
0 1	比較一致でトル(交互切り替え)出力	接続	切断
1 0	比較一致で解除(0)	接続	切断
1 1	比較一致で設定(1)	接続	切断

右表はPWM4DとWGM41,0ビットが高速PWM動作に設定される時のCOM4D1,0ビットの機能を示します。

表15-17. 高速PWM動作比較D出力選択

COM4D1,0	OCW4Dの動き	OC4Dピン	OC4Dピン
0 0	標準ポート動作	切断	切断
0 1	比較一致で解除(0)、TCNT4=\$000で設定(1)	接続	接続
1 0	比較一致で解除(0)、TCNT4=\$000で設定(1)	接続	切断
1 1	比較一致で設定(1)、TCNT4=\$000で解除(0)	接続	切断

右表はPWM4DとWGM41,0ビットが位相/周波数基準PWM動作に設定される時のCOM4D1,0ビットの機能を示します。

表15-18. 位相/周波数基準PWM動作比較D出力選択

COM4D1,0	OCW4Dの動き	OC4Dピン	OC4Dピン
0 0	標準ポート動作	切断	切断
0 1	上昇計数時の比較一致で解除(0)、下降計数時の比較一致で設定(1)	接続	接続
1 0	上昇計数時の比較一致で解除(0)、下降計数時の比較一致で設定(1)	接続	切断
1 1	上昇計数時の比較一致で設定(1)、下降計数時の比較一致で解除(0)	接続	切断

●ビット1 – FOC4D : 比較D強制変更 (Force Output Compare Match 4D)

FOC4DビットはPWM4Dビットが非PWM動作を指示する時だけ有効です。

このビットへの1書き込みは波形出力(OCW4D)と既にCOM4D1,0ビットで設定した値に従って比較出力(OC4D,OC4D)ピンの変更を強制します。COM4D1,0をFOC4Dと同じ周期で書いた場合、新しい設定が使われます。比較出力強制ビットはタイマ/カウンタ値と無関係に出力ピンを変更するのに使えます。COM4D1,0で設定した自動動作は比較一致が起こった場合のように実行しますが、割り込みは生成されません。

FOC4Dビットは常に0として読みます。

●ビット0 – PWM4D : PWM-D許可 (Pulse width Moduration D Enable)

設定(1)時、このビットはOCR4D比較器を基にしたPWM動作を許可します。

15.12.4. TCCR4D – タイマ/カウンタ4制御レジスタD (Timer/Counter4 Control Register D)

ビット (\$C3)	7	6	5	4	3	2	1	0	
	FPIE4	FPEN4	FPNC4	FPES4	FPAC4	FPF4	WGM41	WGM40	TCCR4D
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 – FPIE4 : 障害保護割り込み許可 (Fault Protection Interrupt Enable)

このビットの(1への)設定が障害保護割り込みを許可します。

● ビット6 – FPEN4 : 障害保護動作許可 (Fault Protection Mode Enable)

このビットの(1への)設定が障害保護動作を活性(有効)にします。

● ビット5 – FPNC4 : 障害保護雑音消去許可 (Fault Protection Noise Canceler)

このビットの(1)設定が障害保護雑音消去器を活性(有効)にします。雑音消去器が有効にされると、障害保護入力(INT0)ピンからの入力が濾波されます。この濾波器機能はそれが出力を更新するのにに対して連続4回等しく評価されたINT0ピンの採取を必要とします。従って障害保護は雑音消去器が許可される時に4発振器(システム クロック)周期遅らされます。

● ビット4 – FPES4 : 障害保護起動入力端選択 (Fault Protection Edge Select)

このビットは障害事象を起動するのに使われる障害保護入力(INT0)ピンのどちらかの端(エッジ)を選びます。FPES4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、FPES4ビットが1を書かれると上昇(正)端が障害を起動します。

● ビット3 – FPAC4 : 障害保護アナログ比較器許可 (Fault Protection Analog Comparator 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4の障害保護機能を許可します。この場合、比較器出力は比較器にタイマ/カウンタ4障害保護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障害保護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、アナログ比較器と障害保護機能間の接続は存在しません。比較器がタイマ/カウンタ4障害保護割り込みを起動するにはタイマ/カウンタ4制御レジスタD(TCCR1D)の障害保護割り込み許可(FPIE4)ビットが設定(1)されなければなりません。

● ビット2 – FPF4 : 障害保護割り込み要求フラグ (Fault Protection Interrupt Flag)

FPIE4ビットが設定(1)されると、障害保護割り込みが許可されます。ピン上の動きは例え障害保護ピンが出力として設定されていても割り込み要求を引き起こします。障害保護割り込み要求に対応する割り込みは障害保護割り込みベクタから実行されます。FPF4ビットは対応する割り込み処理ベクタを実行するとき、ハードウェアによって解除(0)されます。代わりにFPF4はこのフラグへの論理1書き込みによっても同期化クロック周期後に解除(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、FP1E4とFPF4が設定(1)される時に障害保護割り込みが実行されます。

● ビット1,0 – WGM41,0 : 波形生成種別 (Waveform Generation Mode Bits)

PWM4x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)と使われるべき波形生成のどの形式かを制御します(下表参照)。タイマ/カウンタ4によって支援される動作種別は標準動作(カウンタ)、高速PWM動作、位相/周波数基準PWM動作とPWM6動作です。

表15-19. 波形生成種別選択					
PWM4x	WGM41,0	タイマ/カウンタ動作種別	TOP値	OCR4x更新時	TOV4設定時
0	x x	標準動作	OCR4C	即時	TOP
1	0 0	高速PWM動作	OCR4C	TOP	TOP
1	0 1	位相/周波数基準PWM動作	OCR4C	BOTTOM	BOTTOM
1	1 0	単一傾斜PWM6動作	OCR4C	TOP	TOP
1	1 1	両傾斜PWM6動作	OCR4C	BOTTOM	BOTTOM

15.12.5. TCCR4E – タイマ/カウンタ4制御レジスタE (Timer/Counter4 Control Register E)

ビット (\$C4)	7	6	5	4	3	2	1	0	
	TLOCK4	ENHC4	OC4OE5	OC4OE4	OC4OE3	OC4OE2	OC4OE1	OC4OE0	TCCR4E
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 – TLOCK4 : レジスタ更新固定化 (Register Update Lock)

このビットは比較レジスタの更新を制御します。このビットが設定(1)されると、比較レジスタへの書き込みは出力に影響を及ぼしませんが、その値は格納され、TLOCK4ビットが解除(0)される時に比較レジスタが(その値で)更新されます。

より多くの詳細については96頁の「同期更新」項を参照してください。

●ビット6 – ENHC4 : 増強化比較/PWM動作許可 (Enhanced Compare/PWM Mode)

このビットが設定(1)されると、波形生成部は増強化動作で働きます。増強化動作ではOCR4A/B/D比較レジスタが更なる1つの精度ビットを受け入れることができ、一方LSBは比較条件が合図されると出力ピンのレベルが更新されるのがどちらのクロック端かを決めます。

●ビット5~0 – OC4OE5~OC4OE0 : 比較出力での無効化許可 (Output Compare Override Enable Bits)

これらのビットは対応する比較出力ピンでの瞬時応答を持つPWM6動作で比較出力ピンを接続または切断する、比較出力での(標準ポート機能)無効化許可ビットです。比較出力での無効化許可ビットが解除(0)されると、ポート出力レジスタからの実際の値がポートピンで見えるでしょう。この表は比較出力での無効化許可ビットとそれらに対応する比較出力ピンを示します。

表15-20. 比較出力での無効化許可ビット 対 比較出力ピン

OC4OE5	OC4OE4	OC4OE3	OC4OE2	OC4OE1	OC4OE0
OC4D (PD7)	$\overline{\text{OC4D}}$ (PD6)	OC4B (PB6)	$\overline{\text{OC4B}}$ (PB5)	OC4A (PC7)	$\overline{\text{OC4A}}$ (PC6)

15.12.6. TCNT4 – タイマ/カウンタ4 (Timer/Counter4)

ビット (\$BE)	7	6	5	4	3	2	1	0	
	(MSB)							(LSB)	TCNT4
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8ビットレジスタはタイマ/カウンタ4の下位バイトを含みます。

タイマ/カウンタ4は読み書きアクセスで10ビット昇降カウンタとして実現されます。CPUの同期化のため、タイマ/カウンタ4に書かれたタイマ/カウンタ4データは同期動作で1.5 CPUクロック周期、非同期動作で更に1 CPUクロック周期遅延されます。10ビット精度が好まれる時は8ビット AVR データバス経由での10ビットTCNT4アクセスに関して特別な手順に従わなければなりません。これらの手順は104頁の「10ビットレジスタのアクセス」項で記述されます。代わりにタイマ/カウンタ4は8ビットタイマ/カウンタとしても使えます。タイマ/カウンタ4が常にTCNT4レジスタ書き込み後に計数を開始することに注意してください。

15.12.7. TC4H – タイマ/カウンタ4上位バイトレジスタ (Timer/Counter4 High Byte)

ビット (\$BF)	7	6	5	4	3	2	1	0	
	–	–	–	–	–	TC410	TC49	TC48	TC4H
Read/Write	R	R	R	R	R	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4一時レジスタは3ビット読み書きレジスタです。

●ビット7~3 – 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。

●ビット2 – TC410 : 増強化動作11ビットアクセス用追加最上位ビット (Additional MSB bits for 11-bit accesses in Enhanced PWM mode)

増強化動作で11ビットを扱う場合のOCR4A/B/D比較レジスタ用の最上位ビットです。これらレジスタへのアクセスは107頁の「10ビットレジスタのアクセス」項の手順に準じなければなりません。増強化動作の詳細については96頁の「増強化された比較/PWM動作」をご覧ください。(訳注: 本ビット記述全体を変更)

●ビット1,0 – TC49,8 : 10ビットアクセスの上位2ビット (Two MSB bits of the 10-bit accesses)

通常動作で10ビット精度が使われる場合、タイマ/カウンタ4上位バイトレジスタは10ビットアクセスのMSB(TC49,TC48)ビットを一時的に格納するのに使われます。タイマ/カウンタ4内の全10ビットレジスタ間で同じTC4Hレジスタが共用されます。10ビットTCNT4を8ビットAVRデータバス経由でアクセスする時に特別な手順に従わなければならないことに注意してください。これらの手順は104頁の「10ビットレジスタのアクセス」項で記述されます。

15.12.8. OCR4A – タイマ/カウンタ4 比較レジスタ (Timer/Counter4 Output Compare Register A)

ビット (\$CF)	7 (MSB)	6	5	4	3	2	1	0 (LSB)	OCR4A
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8ビット読み書きレジスタです。

比較レジスタは継続的にタイマ/カウンタ(TCNT4)と比較されるデータを含みます。比較一致での動作はタイマ/カウンタ4制御レジスタA(TCCR4A)で指定されます。比較一致はタイマ/カウンタ4がOCR4A値へ計数する場合にだけ起こります。TCNT4とOCR4Aを同じ値に設定するソフトウェア書き込みは比較一致を生成しません。

比較一致は比較事象に続く同期化遅延後に比較A一致割り込み要求フラグ(OCF4A)を設定(1)します。

10(11)ビット精度が使われる場合、8ビット AVRデータバス経由で10ビット比較A内部レジスタをアクセスする時に特別な手順に従わなければならないことに注意してください。これらの手順は104頁の「10ビット レジスタのアクセス」項で記述されます。

15.12.9. OCR4B – タイマ/カウンタ4 比較Bレジスタ (Timer/Counter4 Output Compare Register B)

ビット (\$D0)	7 (MSB)	6	5	4	3	2	1	0 (LSB)	OCR4B
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8ビット読み書きレジスタです。

比較レジスタは継続的にタイマ/カウンタ(TCNT4)と比較されるデータを含みます。比較一致での動作はタイマ/カウンタ4制御レジスタA(TCCR4A)で指定されます。比較一致はタイマ/カウンタ4がOCR4B値へ計数する場合にだけ起こります。TCNT4とOCR4Bを同じ値に設定するソフトウェア書き込みは比較一致を生成しません。

比較一致は比較事象に続く同期化遅延後に比較B一致割り込み要求フラグ(OCF4B)を設定(1)します。

10(11)ビット精度が使われる場合、8ビット AVRデータバス経由で10ビット比較B内部レジスタをアクセスする時に特別な手順に従わなければならないことに注意してください。これらの手順は104頁の「10ビット レジスタのアクセス」項で記述されます。

15.12.10. OCR4C – タイマ/カウンタ4 比較Cレジスタ (Timer/Counter4 Output Compare Register C)

ビット (\$D1)	7 (MSB)	6	5	4	3	2	1	0 (LSB)	OCR4C
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8ビット読み書きレジスタです。

比較レジスタは継続的にタイマ/カウンタ(TCNT4)と比較されるデータを含み、比較一致はTCNT4を解除(=\$000)します。このレジスタは標準動作とPWM動作で同じ機能を持ちます。

3より小さな値が比較Cレジスタに書かれると、このレジスタへ書き込むことが許された最小値である3によって置換されることに注意してください。

10ビット精度が使われる場合、8ビット AVRデータバス経由で10ビット比較C内部レジスタをアクセスする時に特別な手順に従わなければならないことに注意してください。これらの手順は104頁の「10ビット レジスタのアクセス」項で記述されます。

15.12.11. OCR4D – タイマ/カウンタ4 比較Dレジスタ (Timer/Counter4 Output Compare Register D)

ビット (\$D2)	7 (MSB)	6	5	4	3	2	1	0 (LSB)	OCR4D
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8ビット読み書きレジスタです。

比較Dレジスタは継続的にタイマ/カウンタ(TCNT4)と比較されるデータを含みます。比較一致での動作はタイマ/カウンタ1制御レジスタC(TCCR4C)で指定されます。比較一致はタイマ/カウンタ4がOCR4D値へ計数する場合にだけ起こります。TCNT4とOCR4Dを同じ値に設定するソフトウェア書き込みは比較一致を生成しません。

比較一致は比較事象に続く同期化遅延後に比較D一致割り込み要求フラグ(OCF4D)を設定(1)します。

10(11)ビット精度が使われる場合、8ビット AVRデータバス経由で10ビット比較D内部レジスタをアクセスする時に特別な手順に従わなければならないことに注意してください。これらの手順は104頁の「10ビット レジスタのアクセス」項で記述されます。

15.12.12. TIMSK4 – タイマ/カウンタ4 割り込み許可レジスタ (Timer/Counter4 Interrupt Mask Register)

ビット (\$72)	7	6	5	4	3	2	1	0	
	OCIE4D	OCIE4A	OCIE4B	–	–	TOIE4	–	–	TIMSK4
Read/Write	R/W	R/W	R/W	R	R	R/W	R	R	
初期値	0	0	0	0	0	0	0	0	

● ビット7 – OCIE4D : タイマ/カウンタ4比較D割り込み許可 (Timer/Counter4 Output Compare Match D Interrupt Enable)

OCIE4D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)されると、タイマ/カウンタ4比較D一致割り込みが許可されます。比較D一致が起きると、ベクタ\$0050の対応する割り込みが実行されます。タイマ/カウンタ4の比較D一致フラグ(OCF4D)がタイマ/カウンタ4割り込み要求フラグレジスタ(TIFR4)で設定(1)されます。

● ビット6 – OCIE4A : タイマ/カウンタ4比較A割り込み許可 (Timer/Counter4 Output Compare Match A Interrupt Enable)

OCIE4A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)されると、タイマ/カウンタ4比較A一致割り込みが許可されます。比較A一致が起きると、ベクタ\$004Cの対応する割り込みが実行されます。タイマ/カウンタ4の比較A一致フラグ(OCF4A)がタイマ/カウンタ4割り込み要求フラグレジスタ(TIFR4)で設定(1)されます。

● ビット5 – OCIE4B : タイマ/カウンタ4比較B割り込み許可 (Timer/Counter4 Output Compare Match B Interrupt Enable)

OCIE4B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)されると、タイマ/カウンタ4比較B一致割り込みが許可されます。比較B一致が起きると、ベクタ\$004Aの対応する割り込みが実行されます。タイマ/カウンタ4の比較B一致フラグ(OCF4B)がタイマ/カウンタ4割り込み要求フラグレジスタ(TIFR4)で設定(1)されます。

● ビット2 – TOIE4 : タイマ/カウンタ4溢れ割り込み許可 (Timer/Counter4 Overflow Interrupt Enable)

TOIE4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)されると、タイマ/カウンタ4溢れ割り込みが許可されます。タイマ/カウンタ4溢れが起きると、(ベクタ\$0052の)対応する割り込みが実行されます。(タイマ/カウンタ4)溢れ割り込み要求(TOV4)フラグはタイマ/カウンタ4割り込み要求フラグレジスタ(TIFR4)で設定(1)されます。

15.12.13. TIFR4 – タイマ/カウンタ4 割り込み要求フラグレジスタ (Timer/Counter4 Interrupt Flag Register)

ビット \$19 (\$39)	7	6	5	4	3	2	1	0	
	OCF4D	OCF4A	OCF4B	–	–	TOV4	–	–	TIFR4
Read/Write	R/W	R/W	R/W	R	R	R/W	R	R	
初期値	0	0	0	0	0	0	0	0	

● ビット7 – OCF4D : タイマ/カウンタ4比較D割り込み要求フラグ (Timer/Counter4, Output Compare D Match Flag)

OCF4Dビットはタイマ/カウンタ4(TCNT4)と比較Dレジスタ(OCR4D)のデータ値間で比較一致が起こる時に設定(1)されます。対応する割り込み処理ベクタ実行時にOCF4D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の論理1書き込みによっても同期化クロック周期後にOCF4Dが解除(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4割り込み許可レジスタ(TIMSK4)のタイマ/カウンタ4比較D一致割り込み許可(OCIE4D)ビット、OCF4Dが設定(1)されると、タイマ/カウンタ4比較D一致割り込みが実行されます。

● ビット6 – OCF4A : タイマ/カウンタ4比較A割り込み要求フラグ (Timer/Counter4, Output Compare A Match Flag)

OCF4Aビットはタイマ/カウンタ4(TCNT4)と比較Aレジスタ(OCR4A)のデータ値間で比較一致が起こる時に設定(1)されます。対応する割り込み処理ベクタを実行すると、OCF4A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の論理1書き込みによっても同期化クロック周期後にOCF4Aが解除(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4割り込み許可レジスタ(TIMSK4)のタイマ/カウンタ4比較A一致割り込み許可(OCIE4A)ビット、OCF4Aが設定(1)されると、タイマ/カウンタ4比較A一致割り込みが実行されます。

● ビット5 – OCF4B : タイマ/カウンタ4比較B割り込み要求フラグ (Timer/Counter4, Output Compare B Match Flag)

OCF4Bビットはタイマ/カウンタ4(TCNT4)と比較Bレジスタ(OCR4B)のデータ値間で比較一致が起こる時に設定(1)されます。対応する割り込み処理ベクタを実行すると、OCF4B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の論理1書き込みによっても同期化クロック周期後にOCF4Bが解除(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4割り込み許可レジスタ(TIMSK4)のタイマ/カウンタ4比較B一致割り込み許可(OCIE4B)ビット、OCF4Bが設定(1)されると、タイマ/カウンタ4比較B一致割り込みが実行されます。

● ビット2 – TOV4 : タイマ/カウンタ4溢れ割り込み要求フラグ (Timer/Counter4 Overflow Flag)

標準動作と高速PWM動作でのTOV4ビットはカウンタがTOPに達する時毎で、カウンタがBOTTOMにリセットされる時と同じクロック周期で設定(1)されます。位相/周波数基準PWM動作でのTOV4ビットはBOTTOMに達する時毎で、\$000がカウンタへ計数される時と同じクロック周期で設定(1)されます。

対応する割り込み処理ベクタを実行すると、TOV4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の論理1書き込みによっても同期化クロック周期後にTOV4が解除(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4割り込み許可レジスタ(TIMSK4)のタイマ/カウンタ4溢れ割り込み許可(TOIE4)ビット、TOV4が設定(1)されると、タイマ/カウンタ4溢れ割り込みが実行されます。

15.12.14. DT4 – タイマ/カウンタ4 沈黙時間値レジスタ (Timer/Counter4 Dead Time Value)

ビット (\$D4)	7	6	5	4	3	2	1	0	
	DT4H3	DT4H2	DT4H1	DT4H0	DT4L3	DT4L2	DT4L1	DT4L0	DT4
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	

この沈黙時間値レジスタは8ビット読み書きレジスタです。

タイマ/カウンタ4の全チャネルの沈黙時間遅延は沈黙時間値レジスタ(DT4)によって調節されます。このレジスタは各相互補完出力に対して1組のDT4H3~0とDT4L3~0の2つの領域から成ります。従ってOC4xの上昇端と $\overline{\text{OC4x}}$ の上昇端に対して異なる沈黙時間遅延に調節できます。

- ビット7~4 – DT4H3~0 : OC4x沈黙時間 (Dead Time Value for OC4x Output)

OC4x出力に対する沈黙時間値です。沈黙時間遅延は前置分周したタイマ/カウンタ クロック数として設定されます。最小沈黙時間は0で、最大沈黙時間は前置分周したタイマ/カウンタ4 クロック周期×15です。

- ビット3~0 – DT4L3~0 : $\overline{\text{OC4x}}$ 沈黙時間 (Dead Time Value for $\overline{\text{OC4x}}$ Output)

$\overline{\text{OC4x}}$ 出力に対する沈黙時間値です。沈黙時間遅延は前置分周したタイマ/カウンタ クロック数として設定されます。最小沈黙時間は0で、最大沈黙時間は前置分周したタイマ/カウンタ4 クロック周期×15です。

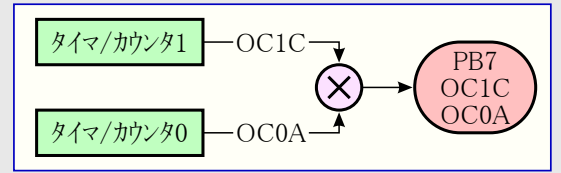
16. 比較出力変調器 (OCM1C0A)

16.1. 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0の比較A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69頁の「16ビット タイマ/カウンタ(タイマ/カウンタ1,タイマ/カウンタ3)」と57頁の「8ビット タイマ/カウンタ0(PWM付き)」をご覧ください。

この変調器が許可されると、構成図(図16-1.)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16-1. 比較出力変調器構成図

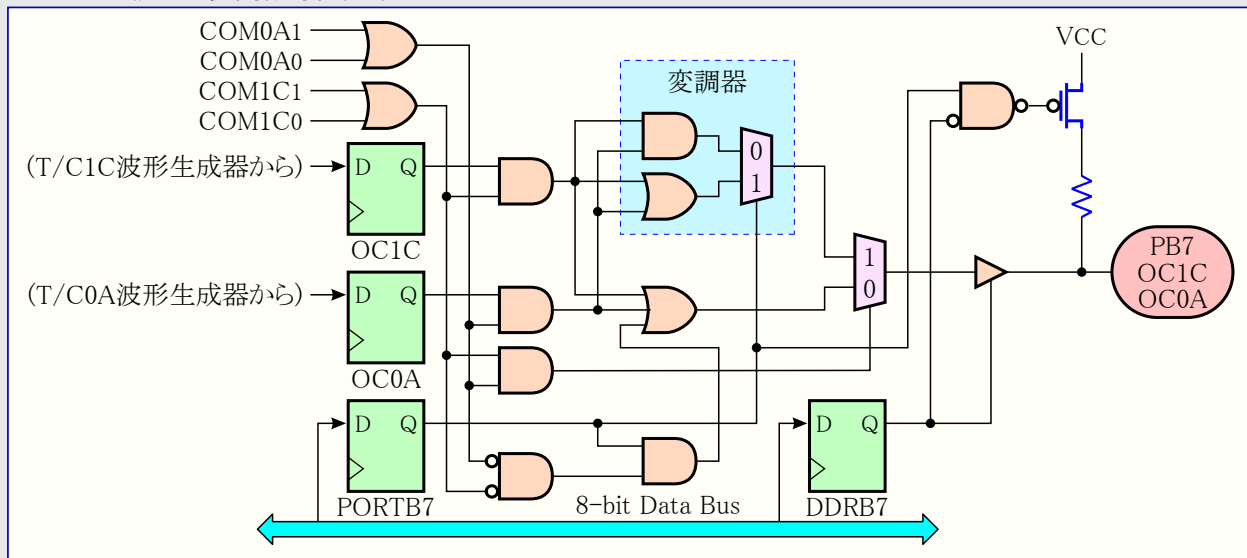


16.2. 説明

比較1C出力部と比較0A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OC0A)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OC0A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16-2.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16-2. 比較出力変調器等価回路図



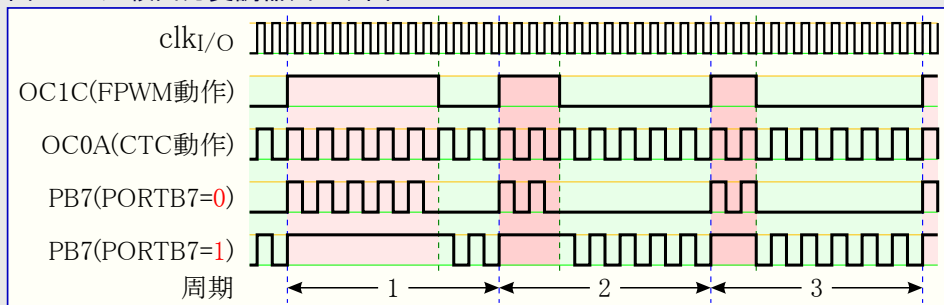
変調器が許可されると、PORTB7レジスタによって変調形式(論理積(AND)または論理和(OR))が選べます。PORTB7=0の時は論理積(AND)が実行され、PORTB7=1の時は論理和(OR)が実行されます(図16-3.)。

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。

16.2.1. タイミング例

図16-3.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0は比較交互出力(COM01,0=01)のCTC波形動作を使用します。

図16-3. 比較出力変調器タイミング図



この例ではタイマ/カウンタ0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(OC0A)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16-3.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。

17. 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ATmega16U4/32U4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。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

USARTはSPI主装置動作も使えます。140頁の「USARTでのSPI動作」をご覧ください。28頁の「PRR0 - 電力削減レジスタ」のPRSPIビットはSPI部を許可するために0を書かれなければなりません。

SPIでの主装置と従装置のCPU間相互連結は図17-2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開始します。主装置と従装置は各々の移動レジスタに送出す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。

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(PCR)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。

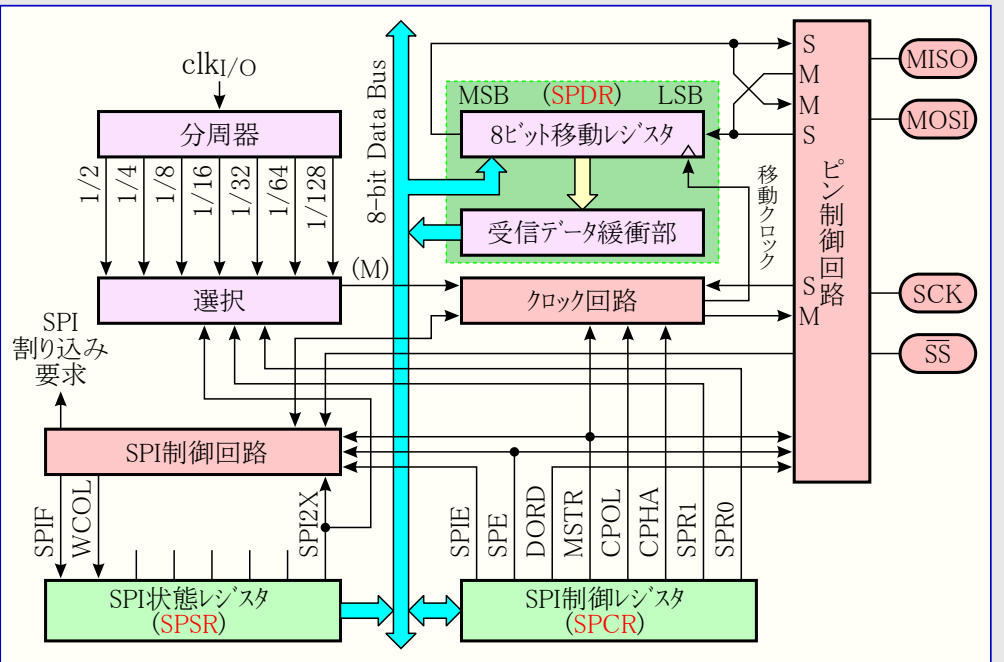
従装置として設定されると、SPIインターフェースはSSピンがHighに駆動される限り、MISOをHi-Zにした休止状態に留まります。この状態でプログラムはSPIデータレジスタ(SPDR)の内容を更新できますが、そのデータはSSピンがLowに駆動されるまでSCKピンでの到着クロックパルスによって移動出力されません。1バイトが完全に移動されてしまうと転送完了フラグ(SPIF)が設定(1)されます。PCRでSPI割り込み許可(SPIE)ビットが設定(1)されていれば割り込みが要求されます。従装置は受信データを読む前にSPDR内へ送られるべき次のデータの配置を続けられます。最後の到着バイトはその後の使用のため、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。けれども、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)から読まれなければなりません。さもなければ始め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、High/Lowの最小時間は各々2 CPUクロック周期よりも長くあるべきです。

SPIが許可されると、MOSI, MISO, SCK, SSピンのデータ方向は右表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17-1. SPI構成図



注: SPIピン配置については2頁の「ピン配置」と45頁の表11-3を参照してください。

図17-2. SPI 主装置/従装置の連結

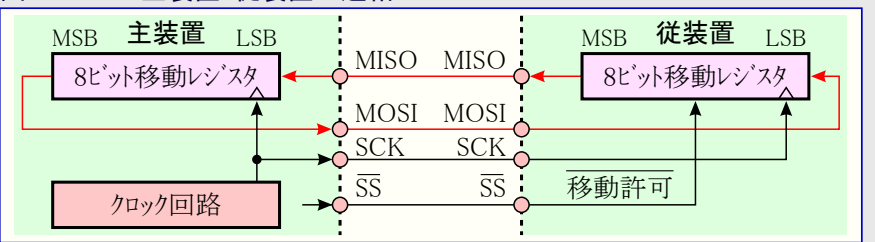


表17-1. SPIピン方向規定

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45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<<SPR0) ;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<<SPR0);    /* 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: 5頁の「コード例」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: 5頁の「コード例」をご覧ください。

17.1. \overline{SS} ピンの機能

17.1.1. 従装置動作

SPIが従装置として設定されると、従装置選択(\overline{SS})ピンは常に入力です。 \overline{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。 \overline{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 \overline{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 \overline{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。 \overline{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。

17.1.2. 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、 \overline{SS} ピンの方向は使用者が決められます。

\overline{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 \overline{SS} ピンを駆動するでしょう。

\overline{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。 \overline{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 \overline{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、 \overline{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。

17.1.3. 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17-3と図17-4.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17-3.と表17-4.を要約することによって明解にされます。

表17-2. 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17-3. SPIデータ転送形式 (CPHA=0)

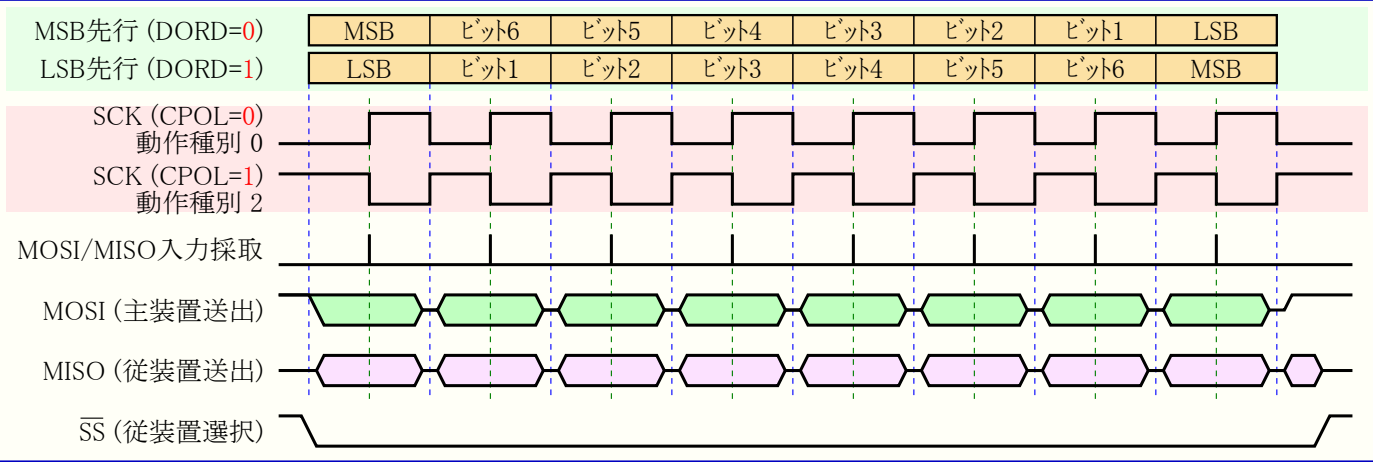
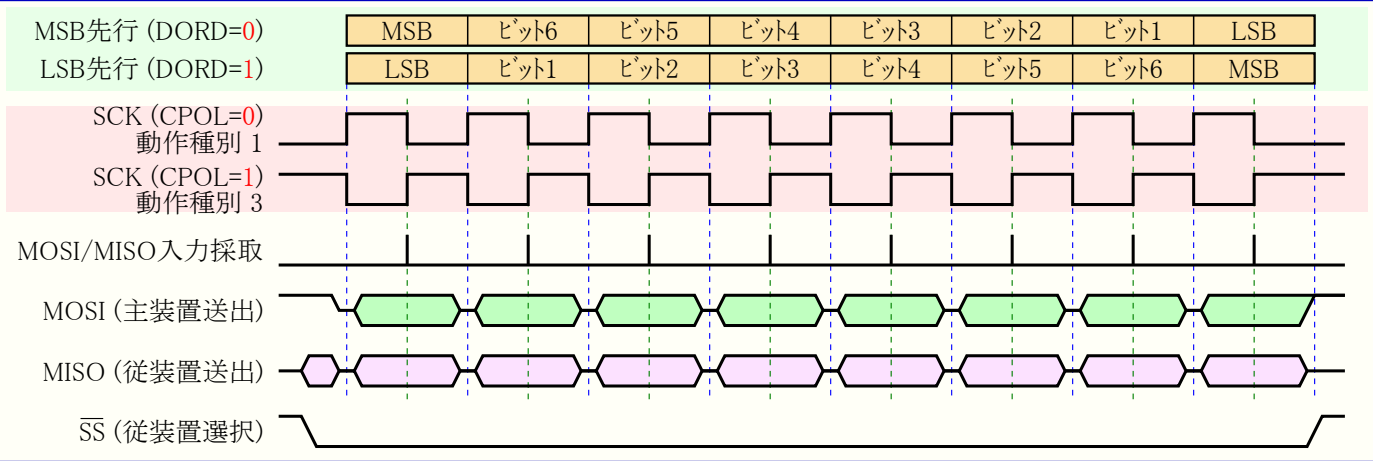


図17-4. SPIデータ転送形式 (CPHA=1)



17.3. SPI用レジスタ

17.3.1. SPCR – SPI制御レジスタ (SPI Control Register)

ビット	7	6	5	4	3	2	1	0	
\$2C (\$4C)	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。 \overline{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17-3.と図17-4.を参照してください。CPOL機能は右で要約されます。

表17-3. 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17-3.と図17-4.を参照してください。CPHA機能は右で要約されます。

表17-4. 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 – SPR1,0 : 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間の関連は下で示されます。

表17-5. 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

17.3.2. SPSR – SPI状態レジスタ (SPI Status Register)

ビット	7	6	5	4	3	2	1	0	
\$2D (\$4D)	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 \overline{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SPIデータレジスタ(SPD R)に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17-5参照)。これは最小SCK周期が2 CPU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またはそれ以下での動作のみ保証されます。

本デバイスの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238頁をご覧ください。

17.3.3. SPDR – SPIデータレジスタ (SPI Data Register)

ビット	7	6	5	4	3	2	1	0	
\$2E (\$4E)	(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。

18. USART

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。

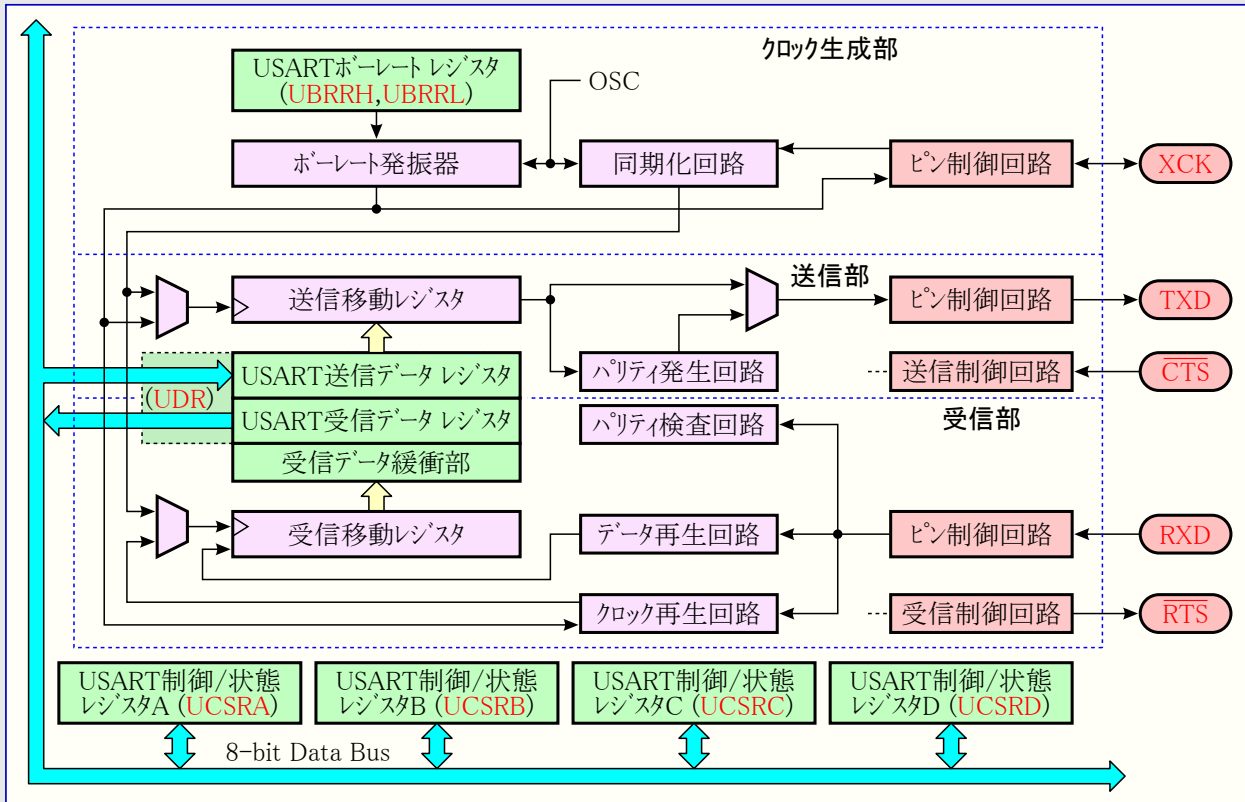
- 全二重動作 (独立した送受信レジスタ)
- 同期または非同期動作
- CTS/RTS信号ハードウェア管理流れ制御
- 同期クロック駆動された主装置/従装置動作
- 高分解能ボーレート発振器
- 5, 6, 7, 8または9ビットデータと1または2停止ビットの直列フレームの支援
- ハードウェアによって支援された奇数または偶数パリティの生成と検査
- データ オーバーラン検出
- フレーミング異常検出
- 不正開始ビット検出とデジタル低域通過濾波器を含む雑音濾波器
- 受信完了、送信完了、送信データレジスタ空きの3つの分離した割り込み
- 複数プロセッサ通信機能
- 倍速非同期通信動作

18.1. 概要

USARTの簡便化した構成図は図18-1.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:原文太字)で示されます。

USARTは主装置SPI動作での使用もできます。140頁の「USARTでのSPI動作」をご覧ください。29頁の「PRR1 – 電力削減レジスタ」のUSART電力削減(PRUSART)ビットはUSART部を許可するために0を書かれなければなりません。(訳注:共通性から前2行追加)

図18-1. USART構成図



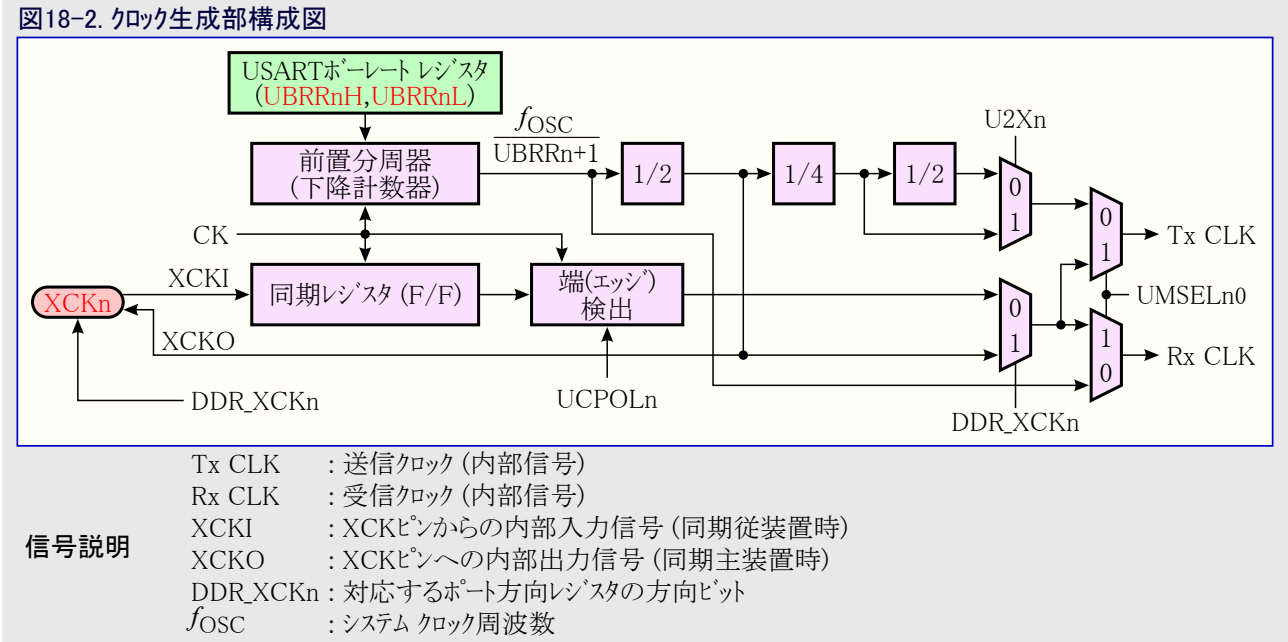
注: USARTピン配置については2頁の「ピン配置」、48頁の表11-8をご覧ください。

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(XCKn)ピンは同期転送動作だけで使われます。送信部は単一書き込み緩衝部(UDRn)、直列移動レジスタ、パリティ発生器、異なる直列フレーム形式を扱うための制御論理回路から成ります。書き込み緩衝部はどんなフレーム間の遅れもなしにデータの継続転送を許します。受信部はクロックとデータの再生部のため、USART部の最も複雑な部分です。再生部は非同期データ受信で使われます。再生部に加えて、受信部は2重の受信緩衝部(UDRn)、移動レジスタ、パリティ検査器、制御論理回路を含みます。受信部は送信部と同じフレーム形式を支援し、フレーミング異常、データ オーバーラン発生、パリティ誤りを検知できます。

18.2. クロック生成

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

図18-2. はクロック生成論理回路の構成図を示します。



18.2.1. ボーレート発振器での内部クロック発生

内部クロック生成は非同期と同期主装置動作種別に対して使われます。本項の記述は図18-2.を参照してください。

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

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

表18-1. ボーレート レジスタ(UBRRn)値計算式

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

注: ボーレートは転送速度(ビット/1秒)で定義されます。

BAUD : ボーレート (bps)

UBRRn : UBRRnHとUBRRnLレジスタ値 (0~4095)

f_{OSC} : システム発振器クロック周波数

いくつかのシステム クロック周波数に対するいくつかのUBRRn値の例は134~135頁の表18-4.で得られます。

18.2.2. 倍速動作 (U2Xn)

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

18.2.3. 外部クロック

外部クロックは同期従装置動作種別によって使われます。本項の記述での詳細については図18-2を参照してください。

XCKnピンからの外部クロック入力是不確定レベル状態(メタステーブル)の機会を最少とするために同期化レジスタによって採取されます。同期化レジスタからの出力は送受信部で使い得るのに先立って端(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XCKnクロック周波数は次式によって制限されます。

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

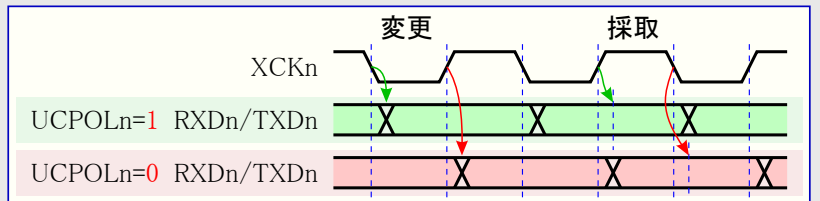
f_{OSC} がシステムクロック元の安定度に依存することに注意してください。従って周波数変動によるデータ消失の可能性を避けるため、いくらかの余裕分を追加することが推奨されます。

18.2.4. 同期クロック動作

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

USART制御/状態レジスタC(UCSRnC)のXCKn極性(UCPOLn)ビットはデータ採取とデータ変更に対してどちらのクロック端が使われるのかを選びます。図18-3.で示されるようにUCPOLnが0のとき、データはXCKnの上昇端で変更され、下降端で採取されます。UCPOLnが設定(1)の場合、データはXCKnの下降端で変更され、上昇端で採取されます。

図18-3. 同期動作XCKnタイミング



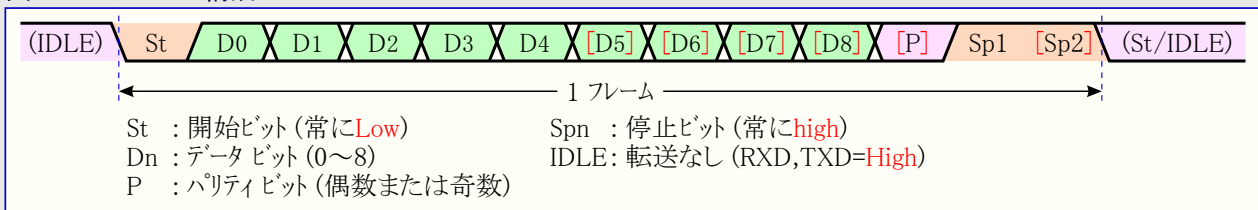
18.3. フレーム形式

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18-4.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18-4. 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(UCSRnB, UCSRnC)でデータ長選択(UCSZn2~0)ビット、パリティ選択(UPMn1,0)ビット、停止ビット選択(USBSn)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(UCSZn2~0)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(UPMn1,0)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(USBSn)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(FEN)は最初の停止ビットが0(Low)の場合にだけ検出されます。

18.3.1. パリティビットの計算

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

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

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

18.4. USARTの初期化

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

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

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

アセンブリ言語プログラム例

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

C言語プログラム例

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

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

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

注: 5頁の「コード例」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。

18.5. USARTのデータ送信

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

18.5.1. 5～8ビット データ フレーム送信

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

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

アセンブリ言語プログラム例

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

C言語プログラム例

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

注: 5頁の「コード例」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、UDRENの検査によって送信緩衝部が空になるのを単純に待ちます。送信緩衝部空き割り込みが使われる場合、その割り込み処理ルーチンがデータを緩衝部内に書きます。

18.5.2. 9ビット データ フレーム送信

9ビットデータが使われる場合(UCSZn2=0=111)、データの下位8ビットがUSARTデータレジスタ(UDRn)に書かれるのに先立って第9ビットがUSART制御/状態レジスタB(UCSRnB)の送信データビット8(TXB8n)ビットに書かれなければなりません。次のコード例は9ビットデータを扱う送信関数を示します。アセンブリ言語での送るべきデータはR17:R16レジスタに格納されると仮定されます。

アセンブリ言語プログラム例

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

C言語プログラム例

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

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

5頁の「コード例」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。

18.5.3. 送信フラグと割り込み

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

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

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

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

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

18.5.4. パリティ発生器

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

18.5.5. 送信の禁止

送信部の禁止(UCSRnBのUSART送信許可(TXENn)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TXDnピン(の標準ピン機能)を無効にしません。

18.6. USARTのデータ受信

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

18.6.1. 5～8ビットデータフレーム受信

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

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

アセンブリ言語プログラム例

```
USART_Rx:    SBIS      UCSRnA, RXCn          ;受信完了でスキップ
              RJMP      USART_Rx            ;受信完了待機
;
              IN        R16, UDRn           ;受信データ取得
              RET                          ;呼び出し元へ復帰
```

C言語プログラム例

```
unsigned char USART_Receive(void)
{
    while ( !(UCSRnA & (1<<RXCn)) );        /* 受信完了待機 */
    return UDRn;                             /* 受信データ取得 */
}
```

注: 5頁の「コード例」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、RXCnフラグの検査によって受信緩衝部にデータが存在するのを単純に待ちます。

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

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

アセンブリ言語プログラム例

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

C言語プログラム例

```
unsigned int USART_Receive(void)
{
    unsigned char status, resh, resl;
    while ( !(UCSRnA & (1<<RxCn)) );
    status = UCSRnA;
    resh = UCSRnB;
    resl = UDRn;
    if ( status & ((1<<FEn) | (1<<DORn) | (1<<UPEn)) ) return -1;
    resh = (resh>>1) & 0x01;
    return ((resh<<8) | resl);
}
```

注: 5頁の「コード例」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。

18.6.3. 受信完了フラグと割り込み

USART受信部には受信部の状態を示す1つのフラグがあります。

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

USART制御/状態レジスタB(UCSRnB)でUSART受信完了割り込み許可(RXCIEn)ビットが設定(1)されると、(全割り込みが許可されていれば)RxCnフラグが設定(1)されている限り、USART受信完了割り込みが実行されます。割り込み駆動データ受信が使われるとき、受信完了割り込みルーチンはRxCnフラグを解除(0)するためにUSARTデータレジスタ(UDRn)から受信したデータを読まなければならない、さもな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。

18.6.4. 受信異常フラグ

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

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

データ オーバーラン発生(DORn)フラグは受信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。DORnフラグが設定(1)なら、最後にUDRnから読んだフレームと次にUDRnから読むフレーム間で1つ以上の直列フレームが失われています。将来のデバイスとの共通性のため、UCSRnAに書くとき、常にこのビットに0を書いてください。DORnフラグは受信されたデータが移動レジスタから受信緩衝部へ正常に移動された時に解除(0)されます。

パリティ誤り(UPEn)フラグは受信緩衝部内の次のフレームで受信時にパリティ異常があったことを示します。パリティ検査が許可されていない場合、UPEnフラグは常に0が読めます。将来のデバイスとの共通性のため、UCSRnAに書くとき、常にこのビットを0に設定してください。より多くの詳細については125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。

18.6.5. 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(UPMn1)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UPMn0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(UPEn)フラグはフレームにパリティ異常があるかを検査するため、ソフトウェアによって読むことができます。

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

18.6.6. 受信の禁止

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

18.6.7. 受信緩衝部の破棄

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

アセンブリ言語プログラム例

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

C言語プログラム例

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

注: 5頁の「コード例」をご覧ください。

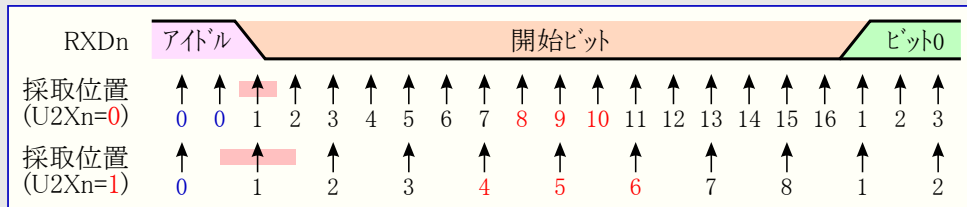
18.7. 非同期受信

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

18.7.1. 非同期クロック再生

クロック再生論理回路は内部クロックを到着直列フレームに同期化します。図18-5は到着フレームの開始ビットの採取手順を図解します。採取速度は標準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(U2Xn=1)を使う時の広い変量時間に注意してください。採取番号0はRXDn信号がアイドル(換言すると、通信の動きなし)の時に実行される採取です。

図18-5. 開始ビットの採取

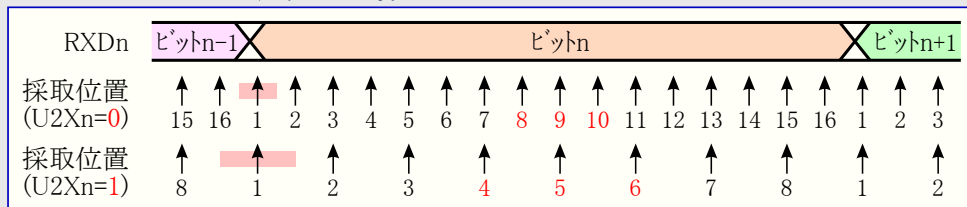


クロック再生論理回路がRXDn信号線でHigh(アイドル)からLow(開始)の遷移を検出すると、開始ビット検出手順が開始されます。図で示されるように採取1は最初のLow採取を意味します。その後にクロック再生回路は有効な開始ビットが受信されるかを決定するために、標準動作に対して採取8,9,10、倍速動作に対して採取4,5,6(図の赤字(訳注:原文は箱枠内)で示された採取番号)を使います。これら3回で2回以上の採取が論理Highレベルの場合(多数決)、この開始ビットは尖頭雑音として捨てられ、受信部は次のHighからLowの遷移を探し始めます。けれども有効な開始ビットが検出されると、クロック再生論理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

18.7.2. 非同期データ再生

受信部クロックが開始ビットに同期化されるとデータ再生が始まります。データ再生部は標準動作で16段、倍速動作で8段の順列回路を使います。図18-6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18-6. データビットとパリティビットの採取



受信したビットの論理値の決定は受信したビット中央の3採取で論理値の多数決を取る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RXDnピンの到着信号に対して低域通過濾波器(ローパスフィルタ)のように働きます。この再生手順はその後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18-7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18-7. 停止ビットの採取と次の開始ビットの採取



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

新規フレームの開始ビットを示すHighからLowへの遷移は多数決に使った最後のビット後に実現できます。標準動作での最初のLowレベル採取は図18-7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。

18.7.3. 非同期での動作範囲

受信部の動作範囲は受信したビット速度と内部的に生成したボーレート間の不一致に依存します。送信部が速すぎるまたは遅すぎるビット速度でフレームを送出したり、内部的に発生した受信部のボーレートが類似した(表18-2.参照)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。

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

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

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

表18-2. 標準速と倍速での受信部ボーレート推奨最大許容誤差

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

注: Dはデータビット数とパリティビットの合計ビット数です。

(訳注) 原書は表18-2.に標準速、表18-3.に倍速を記載していますが、比較が容易なように表18-2.として纏めました。

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

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

18.8. 複数プロセッサ通信動作

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

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

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

18.8.1. 複数プロセッサ通信の使用法

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

複数プロセッサ通信動作でデータを交換するのに次の手順が使われるべきです。

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

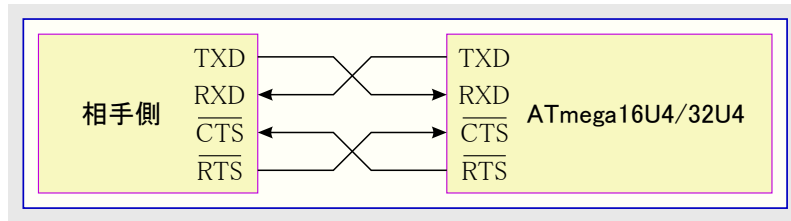
MPCMnビットを設定(1)または解除(0)するのに読み-修正-書き(リド・モデファイライト)命令(SBIとCBI)を使ってはいけません。MPCMnビットは送信完了(TXCn)フラグと同じI/O位置を共用しており、SBIまたはCBI命令を使うと偶然に解除(0)されるかもしれません。

(訳注) ATmega16U4/32U4ではUCSRnAのI/OアドレスがSBI,CBI命令適用範囲外のため、上記記述は不適切です。但し、命令の組み合わせによって同様処理を行う場合に対して、上記注意の本意は適切(有効)です。

18.9. ハードウェア流れ(フロー)制御

ハードウェア流れ制御はソフトウェアによって許可できます。

$\overline{\text{CTS}}$: Clear to Send (相手側からの送信要求/許可信号入力)
 $\overline{\text{RTS}}$: Request to Send (相手側への送信要求/許可信号出力)



18.9.1. 受信部流れ制御

受信の流れは $\overline{\text{RTS}}$ ピンを使うハードウェアによって制御されます。この流れ制御の狙いは内部受信FIFOが満杯の時に外部の送信部へ通知することです。従ってその送信部は文字送信を停止できます。 $\overline{\text{RTS}}$ 使用と関連する流れ制御はUSART制御/状態レジスタD(UCSR_D)のRTS許可(RTSEN)ビットを使うことで許可されます。

図18-8.は受信例を示します。

図18-8. 受信流れ制御波形例

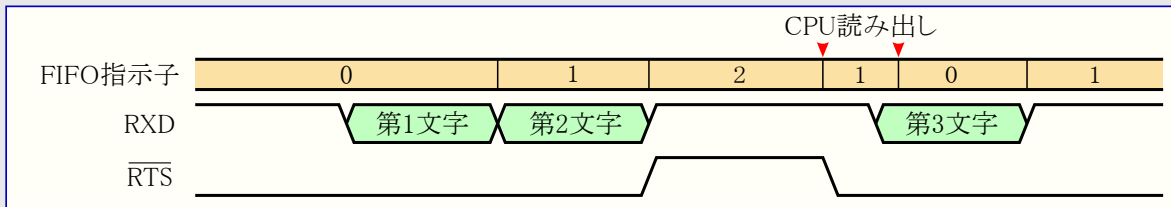
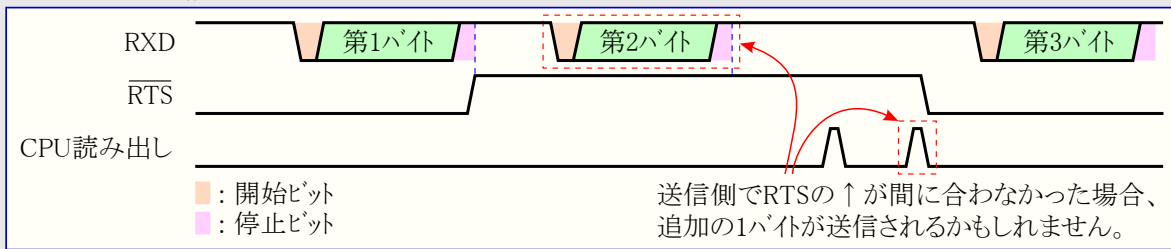


図18-9. RTSの動き



$\overline{\text{RTS}}$ は受信FIFOが満杯の場合、最終受信停止ビットの2/3で上昇します。

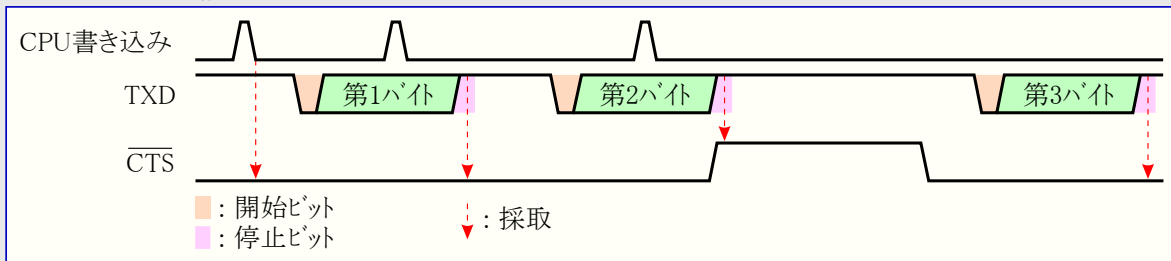
確かな送信を保証するため、 $\overline{\text{RTS}}$ 上昇後も追加データを未だ受信でき、受信移動レジスタ内に格納されます。

18.9.2. 送信部流れ制御

送信の流れは外部の受信部によって制御される $\overline{\text{CTS}}$ ピンを使うハードウェアによって制御されます。この流れ制御の狙いは(外部の)受信部がデータ満杯($\text{CTS}=1$)の時に(こちらからの)送信を停止することです。 $\overline{\text{CTS}}$ 使用と関連する流れ制御はUSART制御/状態レジスタD(UCSR_D)のCTS許可(CTSEN)ビットを使うことで許可されます。

$\overline{\text{CTS}}$ ピンは各CPU書き込みと現状送信中の最終停止ビットの中央で採取されます。

図18-10. CTSの動き



18.10. ボーレート設定例

標準的なクリスタル発振子やセラミック振動子の周波数に対して最も一般的に使われる非同期動作のボーレートは表18-4のUBRRn設定を使うことによって生成できます。目的のボーレートに対して設定誤差1.5%(標準速: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(131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。

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

表18-4. Xtal、ボーレート対UBRRnH,UBRRnL設定 (UBRR=UBRRnH:UBRRnL)

ボーレート (bps)	1MHz				1.8432MHz				2MHz				2.4576MHz			
	U2Xn=0		U2Xn=1		U2Xn=0		U2Xn=1		U2Xn=0		U2Xn=1		U2Xn=0		U2Xn=1	
	UBRR	誤差(%)	UBRR	誤差(%)	UBRR	誤差(%)	UBRR	誤差(%)	UBRR	誤差(%)	UBRR	誤差(%)	UBRR	誤差(%)	UBRR	誤差(%)
1200	51	0.2	103	0.2	95	0.0	191	0.0	103	0.2	207	0.2	127	0.0	255	0.0
2400	25	0.2	51	0.2	47	0.0	95	0.0	51	0.2	103	0.2	63	0.0	127	0.0
4800	12	0.2	25	0.2	23	0.0	47	0.0	25	0.2	51	0.2	31	0.0	63	0.0
9600	6	-7.0	12	0.2	11	0.0	23	0.0	12	0.2	25	0.2	15	0.0	31	0.0
14400	3	8.5	8	-3.5	7	0.0	15	0.0	8	-3.5	16	2.1	10	-3.0	20	1.6
19200	2	8.5	6	-7.0	5	0.0	11	0.0	6	-7.0	12	0.2	7	0.0	15	0.0
28800	1	8.5	3	8.5	3	0.0	7	0.0	3	8.5	8	-3.5	4	6.7	10	-3.0
38400	1	-18.6	2	8.5	2	0.0	5	0.0	2	8.5	6	-7.0	3	0.0	7	0.0
57600	0	8.5	1	8.5	1	0.0	3	0.0	1	8.5	3	8.5	2	-11.1	4	6.7
76800	0	-18.6	1	-18.6	1	-25.0	2	0.0	1	-18.6	2	8.5	1	0.0	3	0.0
115.2k	-	-	0	8.5	0	0.0	1	0.0	0	8.5	1	8.5	0	33.3	2	-11.1
230.4k	-	-	0	-45.7	0	-50.0	0	0.0	0	-45.7	0	8.5	0	-33.3	0	33.3
250k	-	-	-	-	-	-	0	-7.8	-	-	0	0.0	-	-	0	22.9
最高速	62500		125k		115.2k		230.4k		125k		250k		153.6k		307.2k	

ボーレート (bps)	3.2768MHz				3.6864MHz				4MHz				4.608MHz			
	U2Xn=0		U2Xn=1		U2Xn=0		U2Xn=1		U2Xn=0		U2Xn=1		U2Xn=0		U2Xn=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4.~6.となっていますが、共通性から纏めて表18-4.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18-4 (続き). Xtal、ホーレート対UBRRnH,UBRRnL設定 (UBRR=UBRRnH:UBRRnL)

ホーレート (bps)	4.9152MHz				6.144MHz				7.3728MHz				8MHz			
	U2Xn=0		U2Xn=1		U2Xn=0		U2Xn=1		U2Xn=0		U2Xn=1		U2Xn=0		U2Xn=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 (bps)	9.216MHz				9.8304MHz				10MHz				11.0592MHz			
	U2Xn=0		U2Xn=1		U2Xn=0		U2Xn=1		U2Xn=0		U2Xn=1		U2Xn=0		U2Xn=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 (bps)	14.7456MHz				16MHz				18.4320MHz				20MHz			
	U2Xn=0		U2Xn=1		U2Xn=0		U2Xn=1		U2Xn=0		U2Xn=1		U2Xn=0		U2Xn=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。

18.11. USART用レジスタ

18.11.1. UDRn – USARTデータレジスタ (USART I/O Data Register)

ビット (\$CE)	7	6	5	4	3	2	1	0	
	R/TXB7	R/TXB6	R/TXB5	R/TXB4	R/TXB3	R/TXB2	R/TXB1	R/TXB0	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。(訳注:適用範囲外命令のため、この注意は不適切です。)

18.11.2. UCSRnA – USART制御/状態レジスタA (USART Control and Status Register A)

ビット (\$C8)	7	6	5	4	3	2	1	0	
	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 (USART Receive Complete)

このフラグは受信緩衝部に未読データがある時に設定(1)され、受信緩衝部が空の(換言すると、どんな未読データも含まない)時に解除(0)されます。受信部が禁止されると、受信緩衝部が破棄され、その結果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(RXCIEn)ビット**をご覧ください)。

● ビット6 – TXCn : USART送信完了フラグ (USART Transmit Complete)

このフラグは送信移動レジスタ内の完全なフレームが移動出力されてしまい、送信緩衝部(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と、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(TXCIEn)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 (USART 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(UDRIEn)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異常があり、その時点でパリティ検査が許可されていれば(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132頁の「**複数プロセッサ通信動作**」をご覧ください。

18.11.3. UCSRnB – USART制御/状態レジスタB (USART Control and Status Register B)

ビット (\$C9)	7	6	5	4	3	2	1	0	
	RXCIE _n	TXCIE _n	UDRIE _n	RXEN _n	TXEN _n	UCSZ _{n2}	RXB8 _n	TXB8 _n	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_n : 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_n)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_n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_n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_n : 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_n)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_n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_n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_n : 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_n)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_n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_n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_n : 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_n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_n)、オーバラン(DOR_n)、パリティ誤り(UPEN_n)のフラグを無効にします。

●ビット3 – TXEN_n : 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_nピンの標準ポート動作を無効にします。送信の禁止(TXEN_n=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_nポート(の標準I/O機能)を無効にしません。

●ビット2 – UCSZ_{n2} : データビット長選択2 (Character Size)

USART制御/状態レジスタC(UCSRnC)のUCSZ_{n1,0}ビットと組み合わせたUCSZ_{n2}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_n : 受信データビット8 (Receive Data Bit 8)

RXB8_n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_nから下位ビットを読む前に読んでください。

●ビット0 – TXB8_n : 送信データビット8 (Transmit Data Bit 8)

TXB8_n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_nへ下位ビットを書く前に書いてください。

18.11.4. UCSRnC – USART制御/状態レジスタC (USART Control and Status Register C)

ビット (\$CA)	7	6	5	4	3	2	1	0	
	UMSEL _{n1}	UMSEL _{n0}	UPM _{n1}	UPM _{n0}	USBS _n	UCSZ _{n1}	UCSZ _{n0}	UCPOL _n	UCSRnC
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	1	1	0	

●ビット7,6 – UMSEL_{n1,0} : USART動作選択 (USART Mode Select)

このビットは右で示されるようにUSART動作種別を選びます。

表18-7. USART動作選択

UMSEL _{n1}	UMSEL _{n0}	動作種別
0	0	非同期動作
	1	同期動作
1	0	(予約)
	1	主装置SPI (MSPIM) (注)

注: 主装置SPI動作(MSPIM)操作の完全な記述については140頁の「USARTでのSPI動作」をご覧ください。

●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18-8. 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18-9. 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18-10. 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18-11. 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

18.11.5. UCSRnD – USART制御/状態レジスタD (USART Control and Status Register D)

ビット (\$CB)	7	6	5	4	3	2	1	0	UCSRnD
	–	–	–	–	–	–	CTSENn	RTSENn	
Read/Write	R	R	R	R	R	R	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

●ビット7~2 – Res : 予約 (Reserved Bits)

これらのビットは予約されており、0として読みます。これらのビットを設定(1)してはなりません。

●ビット1 – CTSENn : CTS許可 (USART CTS Enable)

送信流れ(フロー)制御(CTS)信号を許可するにはファームウェアによってこのビットを設定(1)してください。送信はCTS入力が0の場合にだけ許可されます。送信流れ制御信号を禁止するにはこのビットを解除(0)してください。送信はハードウェア(制御)条件なしに起きます。この(CTS)ピンを入力として許可するには、データ方向レジスタの(対応)ビットが正しく解除(0)されなければなりません。

●ビット0 – RTSENn : RTS許可 (USART RTS Enable)

受信流れ(フロー)制御(RTS)信号を許可するにはファームウェアによってこのビットを設定(1)してください。この場合、RTS線はFIFOが一杯の時に自動的に上がり(Highになります)。受信流れ制御信号を禁止するにはこのビットを解除(0)してください。この(RTS)ピンを出力として許可するには、データ方向レジスタの(対応)ビットが正しく設定(1)されなければなりません。

18.11.6. UBRRnH, UBRRnL (UBRRn) – USARTボーレートレジスタ (USART Baud Rate Register)

ビット (\$CD)	15	14	13	12	11	10	9	8	UBRRnH
	–	–	–	–	UBRR11	UBRR10	UBRR9	UBRR8	
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 (\$CC)	7	6	5	4	3	2	1	0	UBRRnL
	UBRR7	UBRR6	UBRR5	UBRR4	UBRR3	UBRR2	UBRR1	UBRR0	
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 – UBRR11～0 : ボーレート分周値 (USART Baud Rate Register)

USARTのボーレートを含む12ビットレジスタです。UBRRnHがUSART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。

19. USARTでのSPI動作

USART(Universal Synchronous and Asynchronous Receiver and Transmitter)は主装置SPI互換動作に設定できます。主装置SPI(MSPIM)は次の特徴を持ちます。

- ・全二重動作、3線同期データ転送
- ・主装置動作
- ・4つ全てのSPI転送形式(動作種別0,1,2,3)支援
- ・LSBまたはMSB先行データ転送(データ順設定)
- ・順列動作(2重緩衝)
- ・高分解能ボーレート発振器
- ・高速動作($f_{XCK_{max}} = f_{CK}/2$)
- ・柔軟な割り込み生成

19.1. 概要

USART動作選択(UMSELn1,0)ビットの11設定は主装置SPI(MSPIM)論理回路でのUSARTを許可します。この動作種別でのSPI主装置制御論理回路はUSART資源を直接制御します。これらの資源には送受信の移動レジスタと緩衝部、ボーレート発生器を含みます。パリティ発生/検査論理回路、データ/クロック再生論理回路、送受信制御論理回路は禁止されます。USART送受信論理回路は普通のSPI転送制御論理回路に置き換えられます。けれどもピン制御論理回路と割り込み生成論理回路は両動作種別で全く同じです。

I/Oレジスタ位置は両動作種別で同じです。けれどもMSPIM使用時、制御レジスタの一部の機能が変わります。

19.2. クロック生成

クロック生成論理回路は送受信部に対する基準クロックを生成します。USARTのMSPIM動作種別については内部クロック生成(換言すると主装置動作)だけが支援されます。従って、USARTでMSPIMを正しく動作するにはXCKnピンに対するデータ方向レジスタ(DDR_XCKn)が1(換言すると出力)に設定されなければなりません。なるべくならDDR_XCKnはUSARTでのMSPIMが許可(換言するとTXENnとRXENnが1に設定)される前に設定されるべきです。

MSPIM動作で使われる内部クロック生成はUSART同期主装置動作と同一です。従ってボーレートやUBRRn設定は右表での同じ式を使って計算できます。

表19-1. ボーレートレジスタ(UBRRn)値計算式

動作種別	ボーレート計算式	UBRRn値計算式
同期主装置動作	$BAUD = \frac{f_{OSC}}{2 \times (UBRRn + 1)}$	$UBRRn = \frac{f_{OSC}}{2 \times BAUD} - 1$

注: ボーレートは転送速度(ビット/1秒)で定義されます。

BAUD : ボーレート (bps)

UBRRn : UBRRnHとUBRRnLレジスタ値 (0~4095)

f_{OSC} : システム発振器クロック周波数

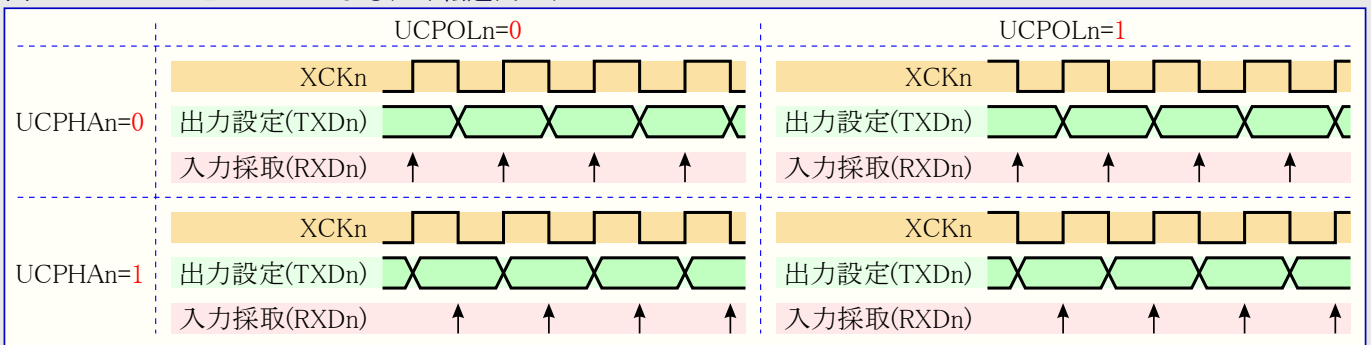
19.3. データ転送形式

直列データに関してはクロック位相(UCPHAn)とクロック極性(UCPOLn)制御ビットによって決定されるXCKn(SCK)位相と極性で4つの組み合わせがあります。このデータ転送タイミング図は図19-1.で示されます。データビットは安定のためにデータ信号に対して十分な時間を保証するXCKn信号の反対端で移動出力と(入力)ラッチが行われます。UCPOLnとUCPHAnの機能は下表で要約されます。これらビットのどの変更も送受信両方に対して実行中の通信を不正とすることに注意してください。

表19-2. UCPOLn,UCPHAn機能動作

SPI動作種別番号	UCPOLn	UCPHAn	XCKn(SCK)先行端	XCKn(SCK)後行端
0	0	0	入力採取/上昇端	出力設定/下降端
1	0	1	出力設定/上昇端	入力採取/下降端
2	1	0	入力採取/下降端	出力設定/上昇端
3	1	1	出力設定/下降端	入力採取/上昇端

図19-1. UCPHAnとUCPOLnによるデータ転送タイミング



19.4. フレーム形式

MSPIMの直列フレームは8データビット/1フレームに固定されます。USARTでのMSPIM動作は2つの有効なフレーム形式を持ちます。

- MSB先行 8ビット データ
- LSB先行 8ビット データ

フレームは最下位(LSB)または最上位(MSB)のデータビットで始まります。その後次のデータビットが最後の最上位(MSB)または最下位(LSB)データビットまで合計8ビット続きます。完全なフレームが送信されると、新規フレームが直ちに後続するか、または通信線がアイドル状態(**high**)に設定されるでしょう。

USART制御/状態レジスタC(UCSRnC)のデータ順選択(UDORDn)ビットはUSARTでのMSPIMによって使われるフレーム形式を設定します。送受信部は同じ設定を使います。これらビットのどれかの設定変更が送受信部両方に対して実行中の通信を不正にすることに注意してください。

16ビット データ通信はUDRnに2バイト データを書くことによって達せられます。その後のUSART送信完了割り込み(TXCn)はこの16ビット値が移動出力されてしまったことを示します。

19.4.1. USART MSPIM初期化

USARTでのMSPIM動作はどれかの通信が行われ得るのに先立って初期化されなければなりません。標準的な初期化手順は使用方法に依存するボーレート設定、主装置動作操作(DDR_XCKnの1)設定、フレーム形式設定、送受信部許可から成ります。送信部だけが独立して操作できます。割り込み駆動USART操作に関して初期化を行う時にステータスレジスタの全割り込み許可(I)ビットが解除(0)され、そして全割り込みが禁止されるべきです。

注: XCKn出力の初期化を直ちに保証するため、ボーレートレジスタ(UBRRn)は送信部が許可される時に0でなければなりません。標準動作のUSART操作と逆に、UBRRnは送信部が許可されて最初の送信が開始される前に、希望する値を書かれなければなりません。この初期化がリセット後直ちに行なわれるなら、UBRRnが0にリセットされているので、送信部許可前にUBRRnを0に設定することは必要ありません。

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

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

アセンブリ言語プログラム例

```
USART_Init: CLR      R18                      ;0値取得
              OUT      UBRRnH, R18            ;ボーレート設定(上位バイト)=0
              OUT      UBRRnL, R18            ;ボーレート設定(下位バイト)=0 (クロック停止)
              SBI      XCKn_DDR, XCKn         ;XCKnポートピン出力設定
              LDI      R18, (1<<UMSELn1) | (1<<UMSELn0) | (0<<UCPHAn) | (0<<UCPOLn)
              ;動作種別値を取得
              OUT      UCSRnC, R18            ;MSPI,データ種別0設定
              LDI      R18, (1<<RXENn) | (1<<TXENn)
              ;送受信許可値を取得
              OUT      UCSRnB, R18            ;送受信許可
              OUT      UBRRnH, R17            ;ボーレート設定(上位バイト)
              OUT      UBRRnL, R16            ;ボーレート設定(下位バイト)
              RET                               ;呼び出し元へ復帰
```

C言語プログラム例

```
void USART_Init(unsigned int baud)
{
    UBRRn = 0;                               /* ボーレート設定(クロック停止) */
    XCKn_DDR |= (1<<XCKn);                   /* XCKnポートピン出力設定 */
    UCSRnC = (1<<UMSELn1) | (1<<UMSELn0) | (0<<UCPHAn) | (0<<UCPOLn);
    ; MSPI,データ種別0設定 */
    UCSRnB = (1<<RXENn) | (1<<TXENn);        /* 送受信許可 */
    UBRRn = baud;                             /* ボーレート設定 */
}
```

注: 5頁の「コード例」をご覧ください。

19.5. データ転送

USARTでのMSPI動作の使用は送信部が許可されること、換言すると**USART制御/状態レジスタB(UCSRnB)**で**送信許可(TXENn)**ビットが**1**に設定される必要があります。送信部が許可されると、TXDnピンの標準ピン動作は無視され、送信部の直列出力としての機能を与えられます。受信部の許可は任意選択で、UCSRnBの**受信許可(RXENn)**ビットの設定(=1)によって行なわれます。受信部が許可されると、RXDnピンの標準ピン動作は無視され、受信部の直列入力としての機能を与えられます。XCKnは両方の場合で転送クロックとして使われます。

初期化後、USARTはデータ転送を行なう準備が整います。データ転送は**USARTデータレジスタ(UDRn)**I/O位置に書くことによって開始されます。送信部が転送クロックを制御するため、これは送受信データ両方の状態についてです。UDRnに書かれたデータは移動レジスタが新規フレームを送る準備が整った時に送信緩衝部から移動レジスタへ移されます。

注: 入力緩衝部に於いて送信されたデータバイト数との同期を保つために、送信された各バイトに対して一度、UDRnが読まれなければなりません。入力緩衝操作は通常のUSART動作と同じで、換言すると、オーバーランが起きると、緩衝部内の先頭ではなく最後に受信したデータが失われます。これは第1バイト、第2、第3、第4バイトの順で4バイトのデータが送信され、全ての転送が完了される前にUDRnが読めない場合、第1バイトではなく、受信されるべき第3バイトが失われます。

次のコード例は**USART制御/状態レジスタA(UCSRnA)**の**送信データレジスタ空き(UDREN)**フラグのポーリングに基いた、簡単なUSARTでのMSPI転送関数を示します。この関数が使われ得る前にUSARTが初期化されなければなりません。アセンブリ言語での送るべきデータはR16レジスタに格納されると仮定され、受信したデータは関数復帰後、同じR16レジスタで利用できます。

この関数は送信されるべき新規データを設定する前に、UDRENの検査によって送信緩衝部が空になるのを単純に待ちます。その後、緩衝部を読んで値を戻す前に、RXCnフラグの検査によって受信緩衝部にデータが存在するのを待ちます。

アセンブリ言語プログラム例

```
USART_MSPI: SBIS    UCSRnA, UDREN    ;送信緩衝部空きでスキップ
             RJMP    USART_MSPI     ;送信緩衝部空き待機
;
             OUT     UDRn, R16       ;データ送信(送信開始)
USART_SPIR: SBIS    UCSRnA, RXCn     ;受信完了でスキップ
             RJMP    USART_SPIR     ;受信完了待機
;
             IN      R16, UDRn       ;受信データ取得
             RET                     ;呼び出し元へ復帰
```

C言語プログラム例

```
void USART_MSPI_Transfer(unsigned int data)
{
    while ( !(UCSRnA & (1<<UDREN)) ); /* 送信緩衝部空き待機 */
    UDRn = data;                       /* データ送信(送信開始) */
    while ( !(UCSRnA & (1<<RXCn)) );  /* 受信完了待機 */
    return UDRn;                       /* 受信データ取得 */
}
```

注: 5ページの「コード例」をご覧ください。

19.5.1. 送受信フラグと割り込み

USARTのMSPI動作での**受信完了(RXCn)**、**送信完了(TXCn)**、**送信データレジスタ空き(UDREN)**フラグと対応する割り込みは通常のUSART操作と機能的に同一です。けれども受信異常状態フラグ(**FEn**, **DORn**, **UPEn**)は使えず、常に**0**として読みます。

19.5.2. 送受信の禁止

USARTのMSPI動作での送受信部の禁止は通常のUSART操作と機能的に同一です。

19.6. USARTでのMSPIMとSPIの比較

USARTでのMSPI動作は次に関してSPIと完全な互換性があります。

- 主装置動作タイミング図
- クロック極性選択(UCPOLn)ビットはSPIのSCK極性選択(CPOL)ビットと機能的に同じです。
- クロック位相選択(UCPHAn)ビットはSPIのSCK位相選択(CPHA)ビットと機能的に同じです。
- データ順選択(UDORDn)ビットはSPIのデータ順選択(DORD)ビットと機能的に同じです。

けれどもUSARTでのMSPI動作がUSART資源を再使用するため、USARTでのMSPI動作はSPIと比較して多少異なります。加えて制御レジスタビットの差異、主装置動作だけがUSARTでのMSPI動作によって支援されること、2つの部間で異なる次の特質があります。

- USARTでのMSPI動作は送信部の(2重)緩衝部を含みます。SPIは緩衝部を持ちません。
- USARTのMSPI動作での受信部は追加の緩衝段を含みます。
- SPIの上書き(WCOL)ビットはUSARTでのMSPI動作に含まれません。
- SPIの倍速許可(SPI2X)ビットは含まれません。しかし、対応するボーレートレジスタ(UBRRn)設定によって同じ効果が達せられます。
- 割り込みタイミングに互換性はありません。
- USARTでのMSPI動作が主装置動作だけのため、ピン制御が異なります。

USARTのMSPI動作とSPIでのピンは右表で示されます。

表19-3. USARTでのMSPIMとSPIのピン比較

USART MSPIM	SPI	備考
TXDn	MOSI	主装置出力のみ
RXDn	MISO	主装置入力のみ
XCKn	SCK	(機能的に同一)
該当なし	SS	USARTでのMSPIMで未支援

19.7. MSPIMでのUSART用レジスタ

以下の節はUSARTを使うSPI操作で使われるレジスタを記述します。

19.7.1. UDRn – USART MSPIMデータレジスタ (USART I/O Data Register)

MSPIMでのUSARTデータレジスタ(UDRn)の機能と説明は通常のUSART操作と同一です。136頁の「UDRn – USARTデータレジスタ」をご覧ください。

19.7.2. UCSRnA – USART MSPIM制御/状態レジスタA (USART Control and Status Register A)

ビット (\$C8)	7	6	5	4	3	2	1	0	
	RXCn	TXCn	UDREN	–	–	–	–	–	UCSRnA
Read/Write	R	R/W	R	R	R	R	R	R	
初期値	0	0	1	0	0	0	1	0	

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

このフラグは受信緩衝部に未読データがある時に設定(1)され、受信緩衝部が空の(換言すると、どんな未読データも含まない)時に解除(0)されます。受信部が禁止されると、受信緩衝部が破棄され、その結果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(RXCIEn)ビットをご覧ください)。

●ビット6 – TXCn : USART送信完了フラグ (USART Transmit Complete)

このフラグは送信移動レジスタ内の完全なフレームが移動出力されてしまい、送信緩衝部(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とき、自動的に解除(0)されるか、またはこのビット位置に1を書くことによって解除(0)できます。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(TXCIEn)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 (USART 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の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(UDRIEn)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4~0 – Res : 予約 (Reserved Bits in MSPI mode)

MSPIM動作時、これらのビットは将来の使用に予約されています。将来のデバイスとの共通性のため、これらのビットはUCSRnAが書かれるとき、0が書かれなければなりません。

19.7.3. UCSRnB – USART MSPIM制御/状態レジスタB (USART Control nad Status Register B)

ビット (\$C9)	7	6	5	4	3	2	1	0	
	RXCIEn	TXCIEn	UDRIEn	RXENn	TXENn	–	–	–	UCSRnB
Read/Write	R/W	R/W	R/W	R/W	R/W	R	R	R	
初期値	0	0	0	0	0	1	1	0	

●ビット7 – RXCIEn : 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(RXCn)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RXCIEn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RXCn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 – TXCIEn : 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(TXCn)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TXCIEn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TXCn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 – UDRIEn : 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(UDREN)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UDRIEn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UDREN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 – RXENn : 受信許可 (Receiver Enable)

このビットへの1書き込みはMSPIMでのUSART受信(部)を許可します。受信部は許可されるとRXDn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ます。MSPIMで受信(部)だけの許可(換言するとRXENn=1とTXENn=0)は、主装置動作だけが支援されて送信(部)が転送クロックを制御するので、意味を持ちません。

●ビット3 – TXENn : 送信許可 (Transmitter Enable)

このビットへの1書き込みがUSART送信(部)を許可します。送信部は許可された時にTXDnピンの標準ポート動作を無効にします。送信の禁止(TXENn=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TXDnポート(の標準I/O機能)を無効にしません。

●ビット2~0 – Res : 予約 (Reserved Bits in MSPI mode)

MSPI動作時、これらのビットは将来の使用に予約されています。将来のデバイスとの共通性のため、これらのビットはUCSRnBが書かれるとき、0が書かれなければなりません。

19.7.4. UCSRnC – USART MSPIM制御/状態レジスタC (USART Control nad Status Register C)

ビット (\$CA)	7	6	5	4	3	2	1	0	UCSRnC
	UMSELn1	UMSELn0	–	–	–	UDORDn	UCPHAn	UCPOLn	
Read/Write	R/W	R/W	R	R	R	R/W	R/W	R/W	
初期値	0	0	0	0	0	1	1	0	

●ビット7,6 – UMSELn1,0 : USART動作選択 (USART Mode Select)

このビットは表で示されるようにUSART動作種別を選びます。通常USART操作の完全な記述については、137頁の「UCSRnC – USART制御/状態レジスタC」をご覧ください。主装置SPI動作(MSPIM)は両UMSELnビットが1に設定される時に許可されます。データ順選択(UDORDn)、クロック位相選択(UCPHAn)、クロック極性選択(UCPOLn)はMSPIMが許可されるのと同じ書き込み操作で設定できます。

表19-4. USART動作選択

UMSELn1	UMSELn0	動作種別
0	0	非同期動作
	1	同期動作
1	0	(予約)
	1	主装置SPI (MSPIM)

●ビット5~3 – Res : 予約 (Reserved Bits in MSPI mode)

MSPI動作時、これらのビットは将来の使用に予約されています。将来のデバイスとの共通性のため、これらのビットはUCSRnCが書かれるとき、0が書かれなければなりません。

●ビット2 – UDORDn : データ順選択 (Data Order)

UDORDn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。UDORDnビットが0を書かれるとMSBが最初に転送されます。詳細については141頁の「フレーム形式」節をご覧ください。

●ビット1 – UCPHAn : クロック位相選択 (Clock Phase)

このクロック位相選択(UCPHAn)ビットの設定はデータがXCKnの先行(先)端または後行(後)端で採取/(設定)されるかを決めます。詳細については140頁の「データ転送形式」節をご覧ください。

●ビット0 – UCPOLn : クロック極性選択 (Clock Polarity)

クロック極性選択(UCPOLn)ビットはXCKnクロックの極性を設定します。UCPOLnとクロック位相選択(UCPHAn)ビットの組み合わせがデータ転送のタイミングを決めます。詳細については140頁の「データ転送形式」節をご覧ください。

19.7.5. UBRRnH, UBRRnL (UBRRn) – USART MSPIMボーレートレジスタ (USART Baud Rate Register)

MSPIMでのボーレートレジスタ(UBRRn)の機能と説明は通常のUSART操作と同一です。139頁の「UBRRnH,UBRRnL – USARTボーレートレジスタ」をご覧ください。

20. 2線直列インターフェース (TWI:Two-wire Serial Interface, I²C)

20.1. 特徴

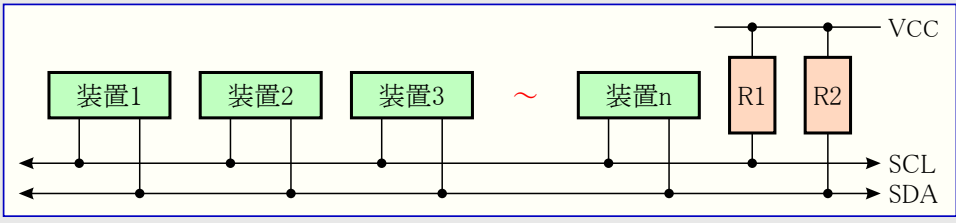
- 2本のバス信号線のみ必要な、単純ながら強力な柔軟な通信インターフェース
- 主装置動作と従装置動作の両方を支援
- 送信装置または受信装置として動作可能
- 7ビットのアドレス空間が128までの異なる従装置アドレスを許容
- 複数主装置の調停支援
- 400kHzまでのデータ転送速度
- 上昇/下降(スレーブ)制限された出力駆動回路
- バス信号線のスパイクを排除する雑音除去回路
- 一斉呼び出しを含む完全に設定変更可能な従装置アドレスの支援
- AVRが休止形態の時のアドレス認証(一致)起動

28頁の「PRR0 - 電力削減レジスタ」のPRTWIビットは2線直列インターフェース部を許可するために0を書かれなければなりません。

20.2. 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20-1. 2線直列(TWI)バス構成



20.2.1. TWI用語定義

次の定義は本章で度々使われます。

表20-1. 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。

20.2.2. 電氣的な相互接続

図20-1.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。

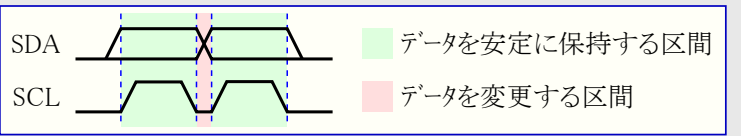
このバスに接続できる装置数は7ビットの従装置アドレス空間と400pFのバス容量制限によってのみ制限されます。TWIの電氣的特性の詳細仕様は251頁の「2線直列インターフェース特性」で与えられます。そこで与えられる2組の異なる仕様は、1つがバス速度100kHz以下に関するもので、もう1つはバス速度400kHzまでに関して有効です。

20.3. データ転送とフレーム形式

20.3.1. 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に伴います。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。

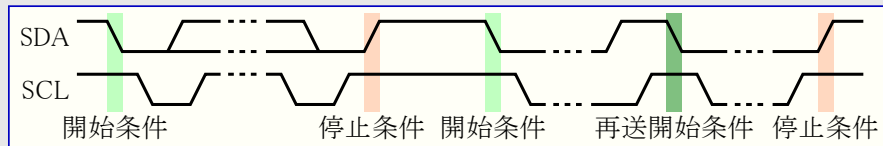
図20-2. データの有効性



20.3.2. 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20-3. 開始条件、再送開始条件、停止条件 バス タイミング



20.3.3. 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: I²C規格のアドレス拡張他)。

20.3.4. 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ができないとき、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20-4. アドレス パケット形式

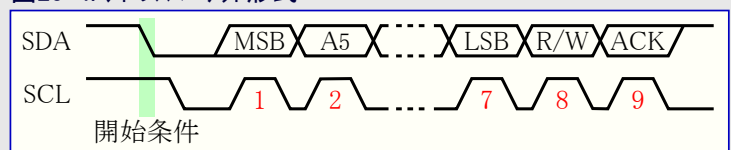
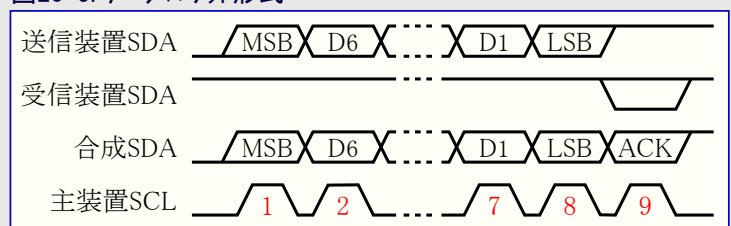


図20-5. データ パケット形式

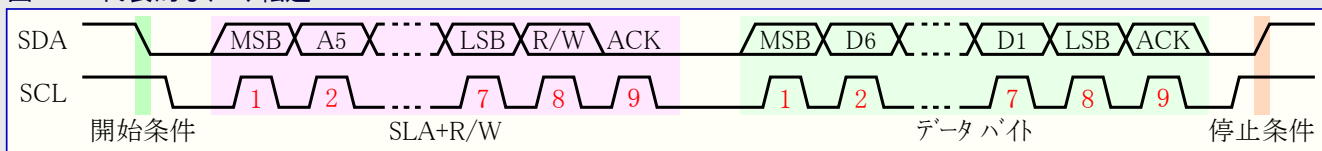


20.3.5. 転送内でのアドレス パケットとデータ パケットの組み合わせ

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20-6.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20-6. 代表的なデータ転送



20.4. 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。例えば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20-7. 複数主装置間でのSCL同期化

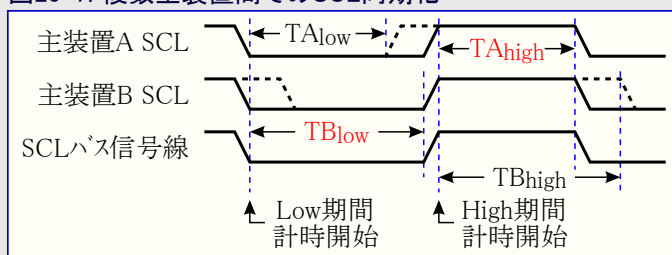
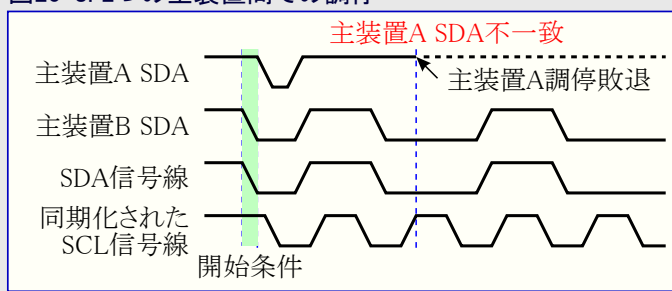


図20-8. 2つの主装置間での調停



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。

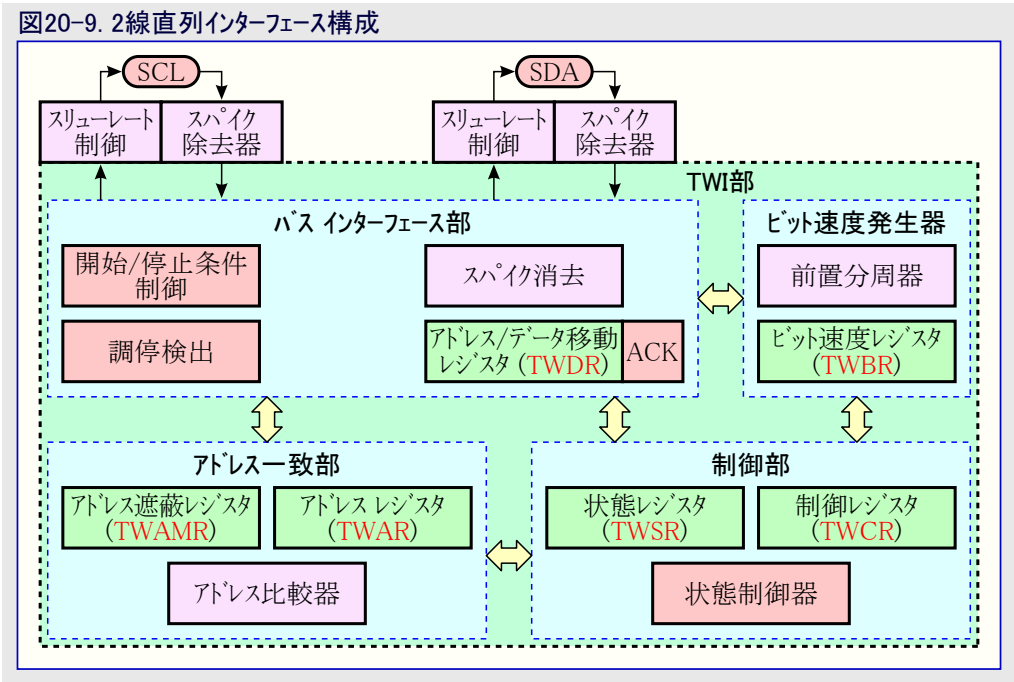
- 再送開始条件とデータ ビット間
- 停止条件とデータ ビット間
- 再送開始条件と停止条件間

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。

20.5. TWI部の概要

図20-9.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。



20.5.1. 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レーブ(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消去部を含みます。「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。

20.5.2. 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ない、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(164頁の表20-7.参照)

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251頁の表29-4.をご覧ください。

20.5.3. 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。

20.5.4. 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)が起これば、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。

20.5.5. 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バスで起これば、**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

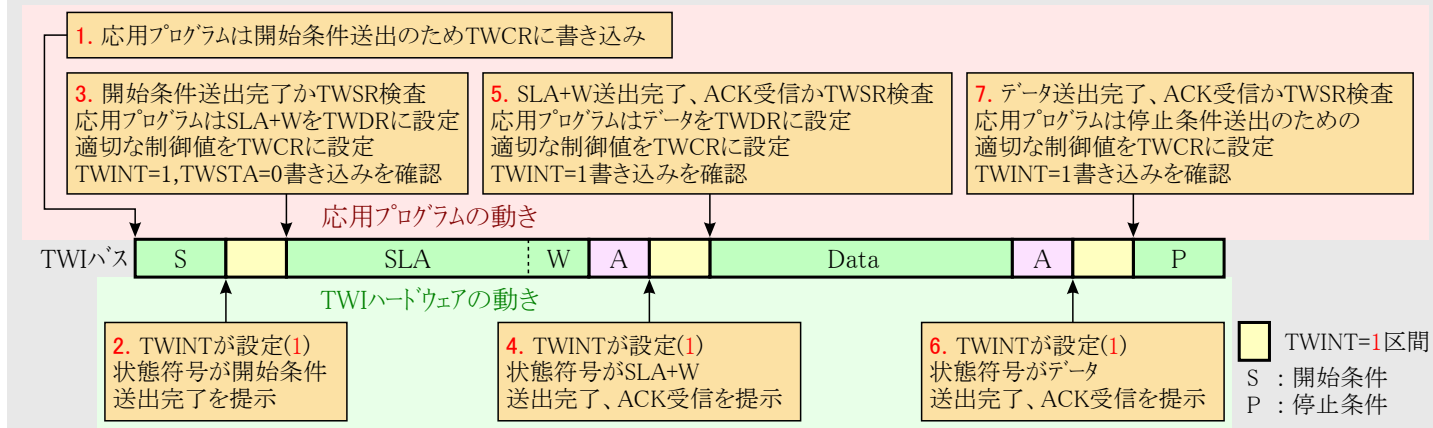
20.6. TWIの使用法

AVR TWIはパルソ志向で割り込みが基本です。割り込みはパルソの受信や**開始条件**の送出のような全てのパルソの事象後に起こります。TWIは割り込みが基本のため、アプリケーションソフトウェアはTWIパルソ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を決めることをアプリケーションソフトウェアに許します。TWIEビットが解除(0)されると、アプリケーションソフトウェアはTWIパルソ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、アプリケーションソフトウェアの応答を待ちます。この場合、TWI状態レジスタ(TWSR)はTWIパルソ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ソフトウェアはTWCRとTWDRの操作により、TWIが次のTWIパルソ周期で何を行うべきかを決定できます。

図20-10はアプリケーションソフトウェアが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パルソ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20-10. 代表的な送信でのアプリケーションプログラムと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内に書くことで行います。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。アプリケーションソフトウェアが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. アプリケーションソフトウェアは**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、アプリケーションソフトウェア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、アプリケーションソフトウェアは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。アプリケーションソフトウェアが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. アプリケーションソフトウェア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、アプリケーションソフトウェア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、アプリケーションソフトウェアは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。アプリケーションソフトウェアが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. アプリケーションソフトウェア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、アプリケーションソフトウェア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、アプリケーションソフトウェアは**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。アプリケーションソフトウェアが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しまっ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) (1<<TWEN) OUT TWCR, R16	TWCR = (1<<TWINT) (1<<TWSTA) (1<<TWEN);	; 開始条件送出
2.	WAIT1: IN R16, TWCR SBRS R16, TWINT RJMP WAIT1	while (!(TWCR & (1<<TWINT)));	; TWINT=1まで待機 ; (開始条件送出完了待機)
3.	IN R16, TWSR ANDI R16, \$F8 CPI R16, START BRNE ERROR	if ((TWSR & 0xF8) != START) ERROR();	; TWI状態レジスタ値検査 ; 前置分周選択ビットの遮蔽 ; STARTと異なる状態符号で ; 異常処理へ
	LDI R16, SLA_W OUT TWDR, R16 LDI R16, (1<<TWINT) (1<<TWEN) OUT TWCR, R16	TWDR = SLA_W; TWCR = (1<<TWINT) (1<<TWEN);	; TWDRにSLA+W設定 ; アドレス送信開始のため ; TWCRのTWINTを解除(0)
4.	WAIT2: IN R16, TWCR SBRS R16, TWINT RJMP WAIT2	while (!(TWCR & (1<<TWINT)));	; TWINT=1まで待機 ; (SLA+W送出完了と ; ACK/NACK受信完了待機)
5.	IN R16, TWSR ANDI R16, \$F8 CPI R16, MT_SLA_ACK BRNE ERROR	if ((TWSR & 0xF8) != MT_SLA_ACK) ERROR();	; TWI状態レジスタ値検査 ; 前置分周選択ビットの遮蔽 ; MT_SLA_ACKと違う状態符号で ; 異常処理へ
	LDI R16, DATA OUT TWDR, R16 LDI R16, (1<<TWINT) (1<<TWEN) OUT TWCR, R16	TWDR = DATA; TWCR = (1<<TWINT) (1<<TWEN);	; TWDRにデータ設定 ; データ送信開始のため ; TWCRのTWINTを解除(0)
6.	WAIT3: IN R16, TWCR SBRS R16, TWINT RJMP WAIT3	while (!(TWCR & (1<<TWINT)));	; TWINT=1まで待機 ; (データ送出完了と ; ACK/NACK受信完了待機)
7.	IN R16, TWSR ANDI R16, \$F8 CPI R16, MT_DATA_ACK BRNE ERROR	if ((TWSR & 0xF8) != MT_DATA_ACK) ERROR();	; TWI状態レジスタ値検査 ; 前置分周選択ビットの遮蔽 ; MT_DATA_ACKと違う状態符号で ; 異常処理へ
	LDI R16, (1<<TWINT) (1<<TWSTO) (1<<TWEN) OUT TWCR, R16	TWCR = (1<<TWINT) (1<<TWSTO) (1<<TWEN);	; 停止条件送出

注: 5頁の「コード例」をご覧ください。

20.7. 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

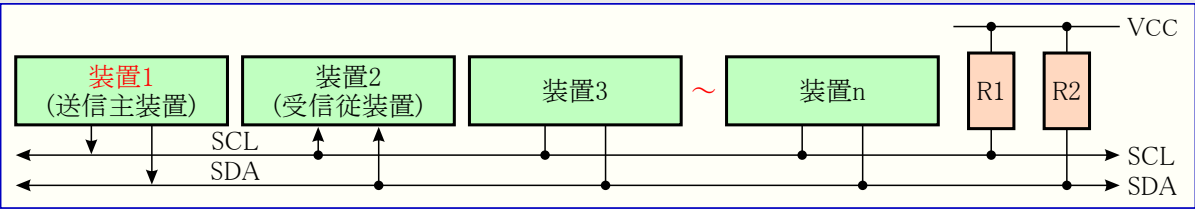
図20-12. ~18.内の楕円(訳注:原文は円)はTWI制御レジスタ(TWCR)のTWI割り込み要求フラグ(TWINT)が設定(1)されたことを示すために使われます。この楕円内の番号は前置分周選択ビットが0で遮蔽された、TWI状態レジスタ(TWSR)に保持した状態符号を表します。これら位置での動きはTWI転送の継続または完了が応用(ソフトウェア)によって行われなければなりません。TWI転送はソフトウェアによってTWINTフラグが解除(0)されるまで一時停止されます。

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20-2. ~5.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。

20.7.1. 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20-11.参照)。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20-11. 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20-2.参照)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20-2.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。

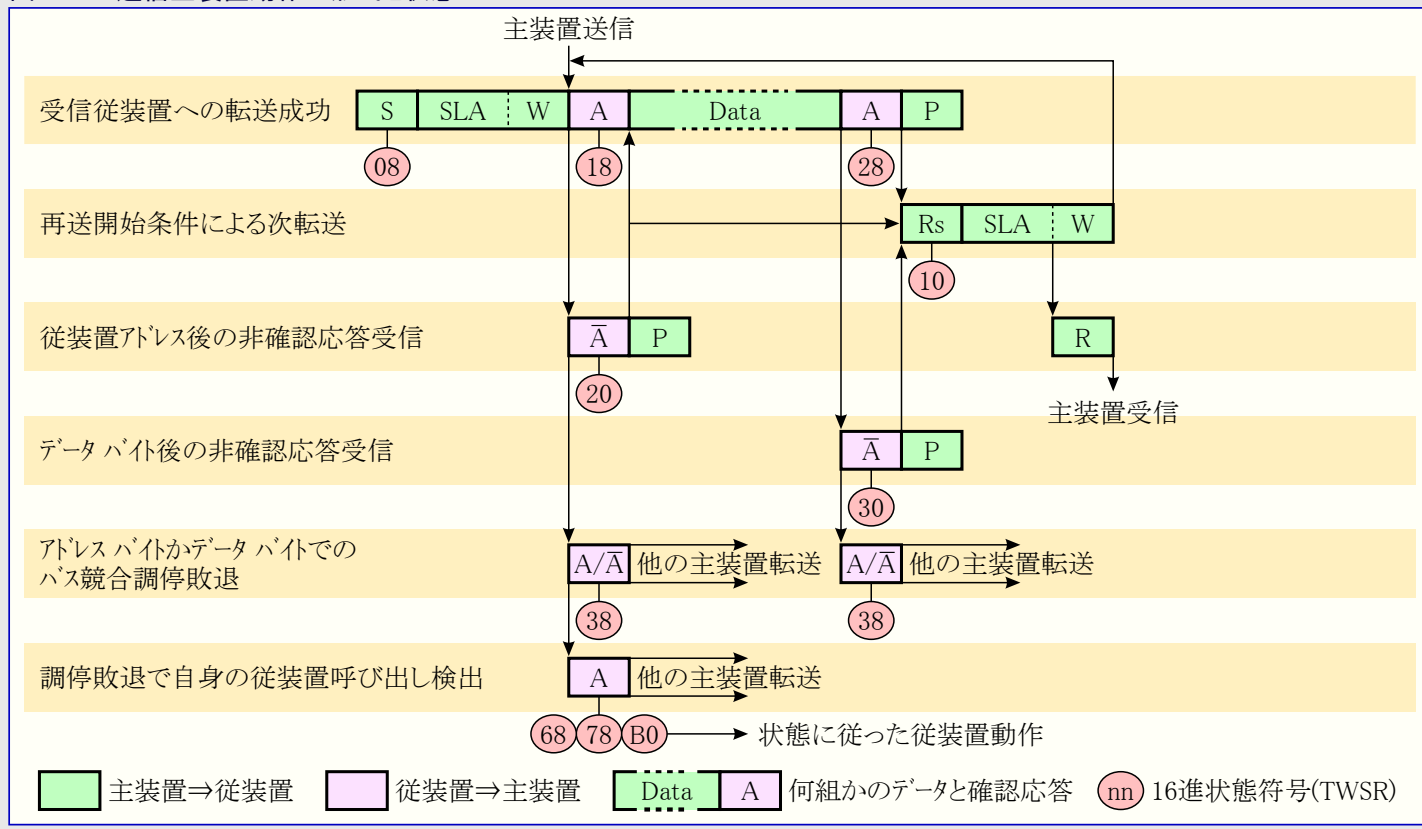
ビット	7	6	5	4	3	2	1	0
TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
設定値	1	X	1	0	X	1	0	X

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20-2. 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 ACK受信	データ設定	0	0	1	X	データ送信、ACKかNACK受信
			1	0	1	X	再送開始条件送信
		なし	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 NACK受信	データ設定	0	0	1	X	データ送信、ACKかNACK受信
			1	0	1	X	再送開始条件送信
		なし	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 ACK受信	データ設定	0	0	1	X	データ送信、ACKかNACK受信
			1	0	1	X	再送開始条件送信
		なし	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 NACK受信	データ設定	0	0	1	X	データ送信、ACKかNACK受信
			1	0	1	X	再送開始条件送信
		なし	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 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

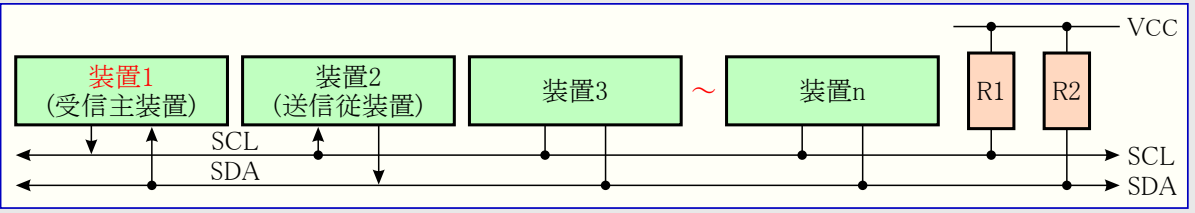
図20-12. 送信主装置動作の形式と状態



20.7.2. 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20-13.参照)。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20-13. 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20-3.参照)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20-3.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。

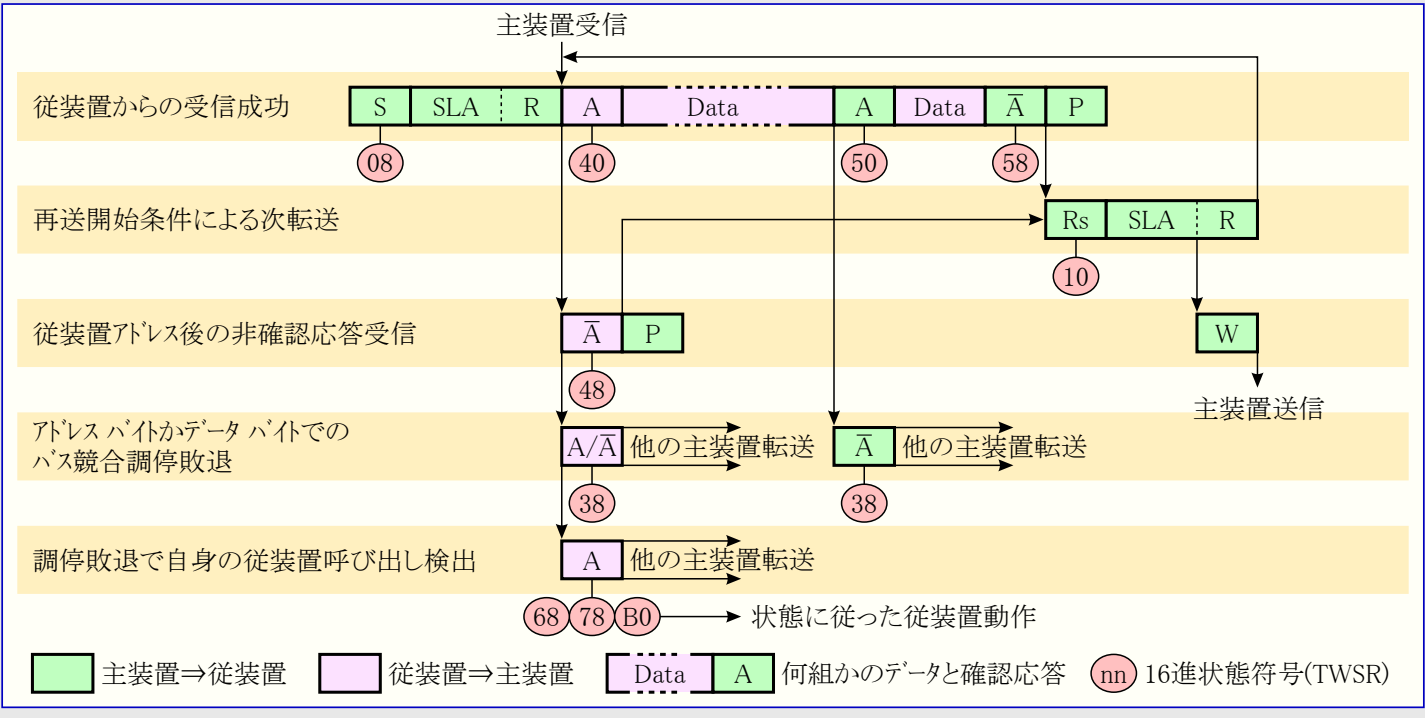
ビット	7	6	5	4	3	2	1	0
TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
設定値	1	X	1	0	X	1	0	X

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20-3. 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)

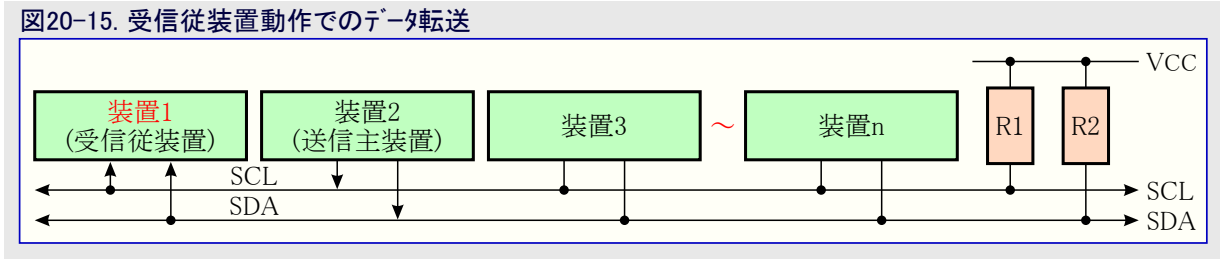
状態符号 (TWSR)	直前の動作と バスの状態	ソフトウェアの対応					TWCR設定によるハードウェア動作
		TWDR操作	TWCR設定				
			TWSTA	TWSTO	TWINT	TWEA	
\$08	開始条件送信	SLA+R設定	0	0	1	X	SLA+R送信、ACKかNACK受信
\$10	再送開始条件送信	SLA+R設定	0	0	1	X	SLA+R送信、ACKかNACK受信
		SLA+W設定	0	0	1	X	SLA+W送信、送信主装置動作へ移行
\$38	SLA+Rで調停敗退 またはNACK受信	なし	0	0	1	X	バス開放、未指定従装置動作へ移行
			1	0	1	X	バス開放時に開始条件送信
\$40	SLA+R送信 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 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 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 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20-14. 受信主装置動作の形式と状態



20.7.3. 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20-15.参照)。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。



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り、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、さもなくば(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20-4.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。

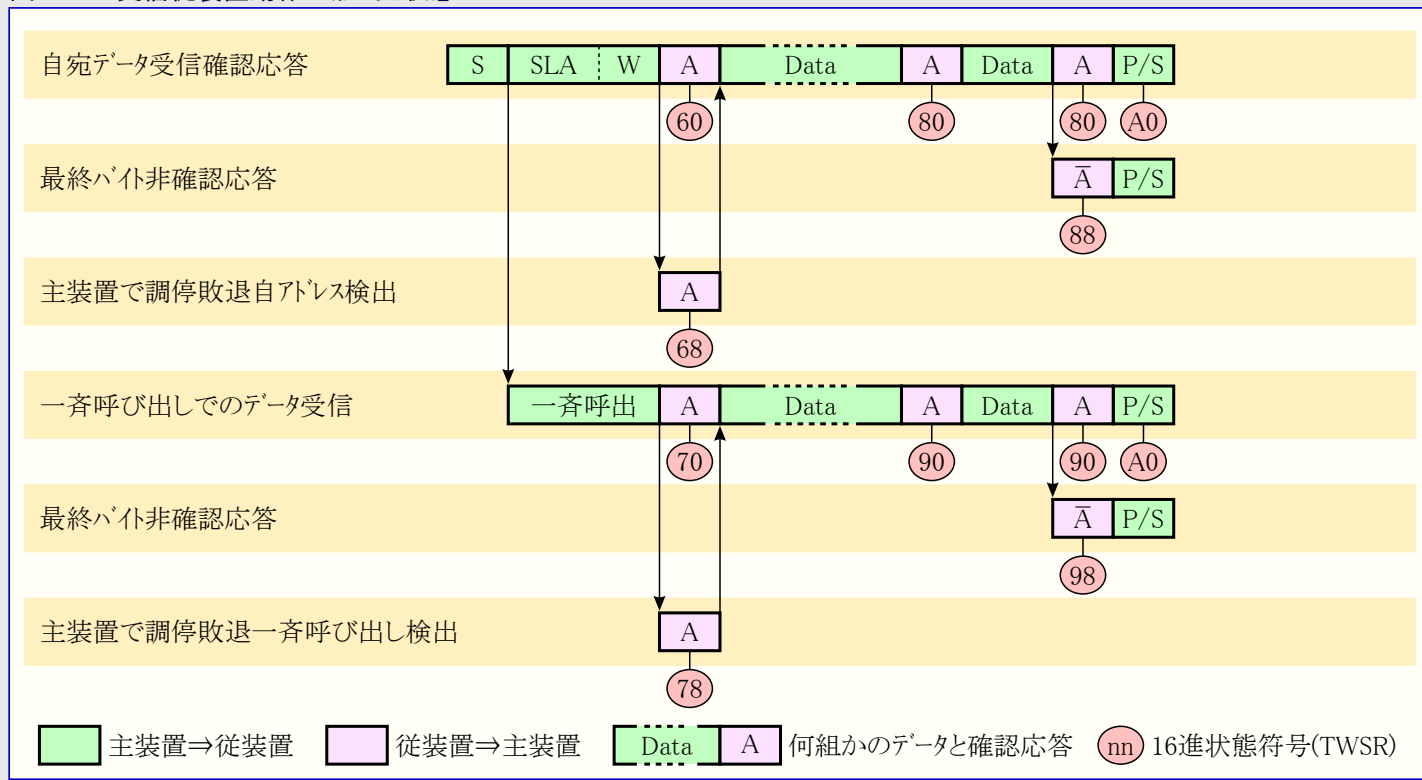
アイドル動作を除く休止形態ではTWIへのクロック系がOFFにされます。TWEAビットが設定(1)されていると、このインターフェースはクロック元として2線直列バスクロック(SCL)を使うことにより、自身の従装置アドレスと一斉呼び出しに未だ確認応答できます。その後デバイスが休止形態から起動し、TWIは起動中からTWINTフラグが(1書き込みによって)解除(0)されるまでSCLクロックをLowに保ちます。その後のデータ受信はAVRクロックが通常通り走行することで通常通りに行われます。AVRが長い起動時間に設定されていると、SCL信号線が長時間Lowに保持され、他のデータ送信を阻止するかもしれないこと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20-4. 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 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 調停敗退/自宛SLA+W 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 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 調停敗退/一斉呼び 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 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 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 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 バス開放で開始条件送信
\$90	一斉呼び出しのデータ 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 バイト受信 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 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 バス開放で開始条件送信
\$A0	自指定中の 停止条件または 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 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 バス開放で開始条件送信

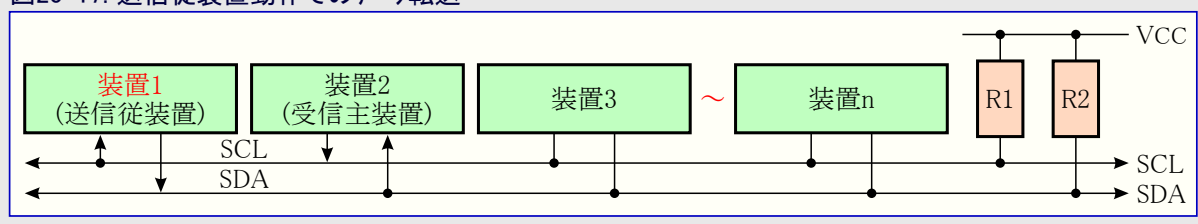
図20-16. 受信従装置動作の形式と状態



20.7.4. 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20-17参照)。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20-17. 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TWCRC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20-5**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から\$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。

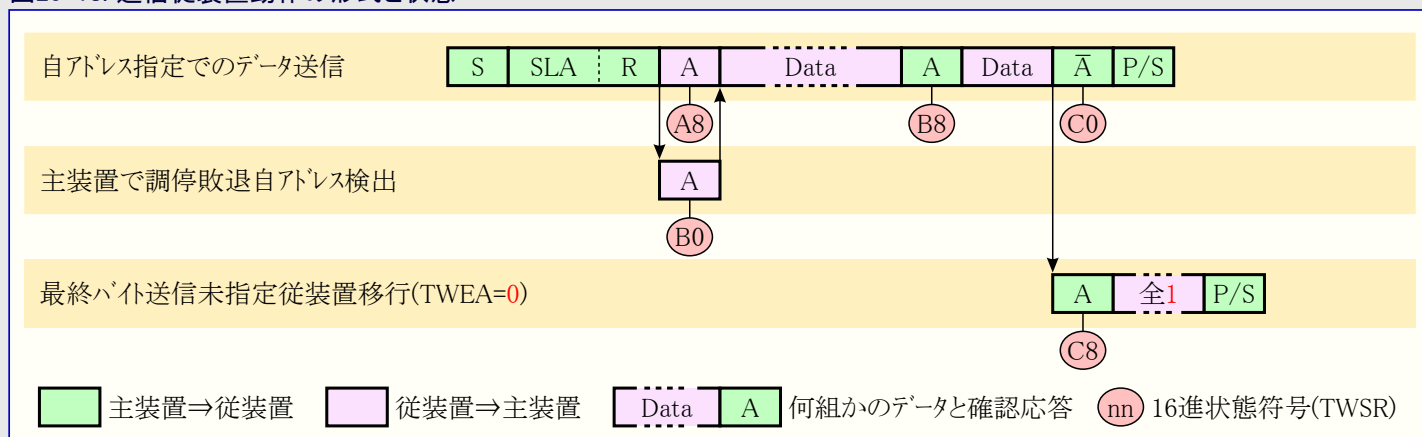
アイドル動作を除く**休止形態**ではTWIへのクロック系がOFFにされます。TWEAビットが設定**(1)**されていると、このインターフェースはクロック元として2線直列バスクロック(SCL)を使うことにより、自身の従装置アドレスと一斉呼び出しに未だ**確認応答**できます。その後デバイスが休止形態から起動し、TWIは起動中からTWINTフラグが**(1)**書き込みによって解除**(0)**されるまでSCLクロックを**Low**に保ちます。その後のデータ受信はAVRクロックが通常通り走行することで通常通りに行われます。AVRが長い起動時間に設定されていると、SCL信号線が長時間**Low**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20-5. 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 (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
\$A8	自宛SLA+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
\$B0	主装置のSLA+R/Wで 調停敗退/自宛SLA+ 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 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 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 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 バス開放で開始条件送信
\$C8	最終データバイト送信 (TWEA=0) 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 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 バス開放で開始条件送信

図20-18. 送信従装置動作の形式と状態



20.7.5. 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。下表をご覧ください。

状態\$F8はTWI割り込み要求フラグ(TWINT)が設定(1)されないなので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)されて、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20-6. 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 (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	
\$F8	適切な状態情報なし 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 TWSTO=0

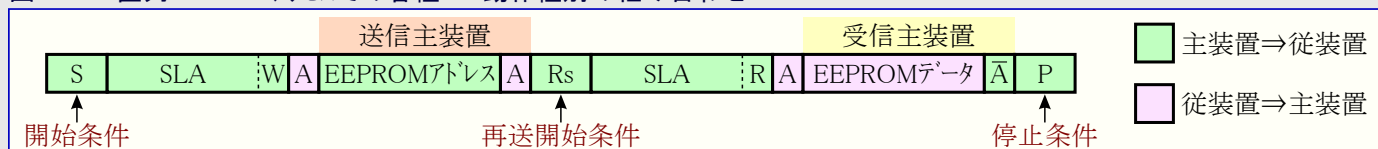
20.7.6. 各種TWI動作種別の組み合わせ

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。その後にデータを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。

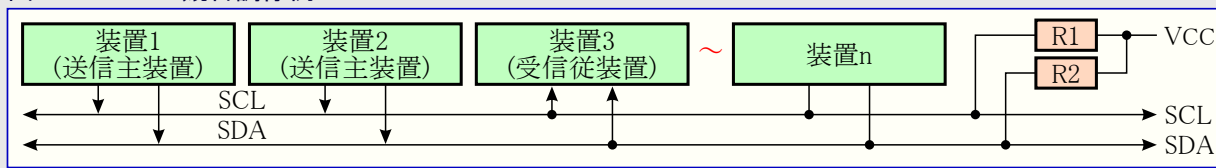
図20-19. 直列EEPROMアクセスでの各種TWI動作種別の組み合わせ



20.8. 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20-20. バスの競合調停例

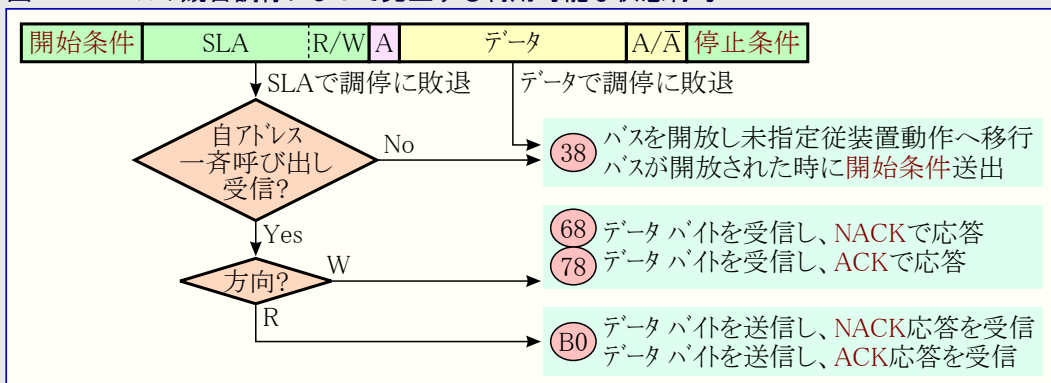


以下で示されるように様々な異なる状況が調停中に起こるかもしれません。

- 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/従装置のどちら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。敗れた主装置は応用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つ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、応用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つて新規開始条件を送出します。

これは図20-21.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20-21. バスの競合調停によって発生する利用可能な状態符号



20.9. TWI用レジスタ

20.9.1. TWBR – TWIビット速度レジスタ (TWI Bit Rate Register)

ビット (\$B8)	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149頁の「[ビット速度発生器](#)」をご覧ください。

20.9.2. TWCR – TWI制御レジスタ (TWI Control Register)

ビット (\$BC)	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、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とき、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**を生成しませんが、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レーブ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。

20.9.3. TWSR – TWI状態レジスタ (TWI Status Register)

ビット (\$B9)	7	6	5	4	3	2	1	0	
	TWS7	TWS6	TWS5	TWS4	TWS3	–	TWPS1	TWPS0	TWSR
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 – TWPS1,0 : 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149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20-7. 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

20.9.4. TWDR – TWIデータレジスタ (TWI Data Register)

ビット (\$BB)	7	6	5	4	3	2	1	0	
	TWD7	TWD6	TWD5	TWD4	TWD3	TWD2	TWD1	TWD0	TWDR
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。

20.9.5. TWAR – TWI(従装置)アドレスレジスタ (TWI (Slave) Address Register)

ビット (\$BA)	7	6	5	4	3	2	1	0	
	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE	TWAR
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。

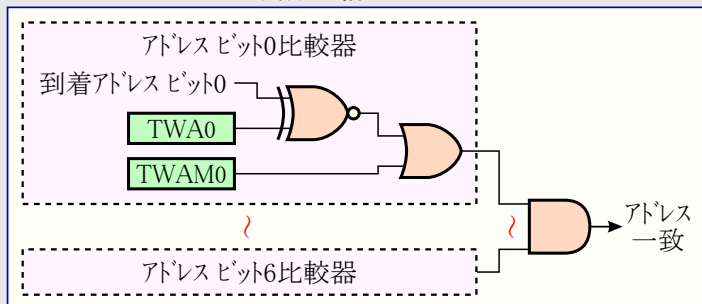
20.9.6. TWAMR – TWI(従装置)アドレス遮蔽レジスタ (TWI (Slave) Address Mask Register)

ビット (\$BD)	7	6	5	4	3	2	1	0	
	TWAM6	TWAM5	TWAM4	TWAM3	TWAM2	TWAM1	TWAM0	–	TWAMR
Read/Write	R/W	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7～1 – TWAM6～0 : TWI従装置アドレス遮蔽 (TWI (Slave) Address Mask)

TWAMRは7ビットの従装置アドレス遮蔽値を格納できます。TWAMR内の各ビットはTWI(従装置)アドレスレジスタ(TWAR)内の対応するアドレスビットを遮蔽(禁止)します。遮蔽ビットが1に設定されると、その後のアドレス一致論理回路は到着アドレスビットとTWAR内の対応ビット間の比較を無視します。図20-22はアドレス一致論理回路を詳細に示します。

図20-22. TWIアドレス一致論理 構成図



● ビット0 – Res : 予約 (Reserved Bit)

このビットは使われず、常に0として読みます。

21.3.1. バス給電装置

図21-3. 5V入出力での代表的バス給電応用

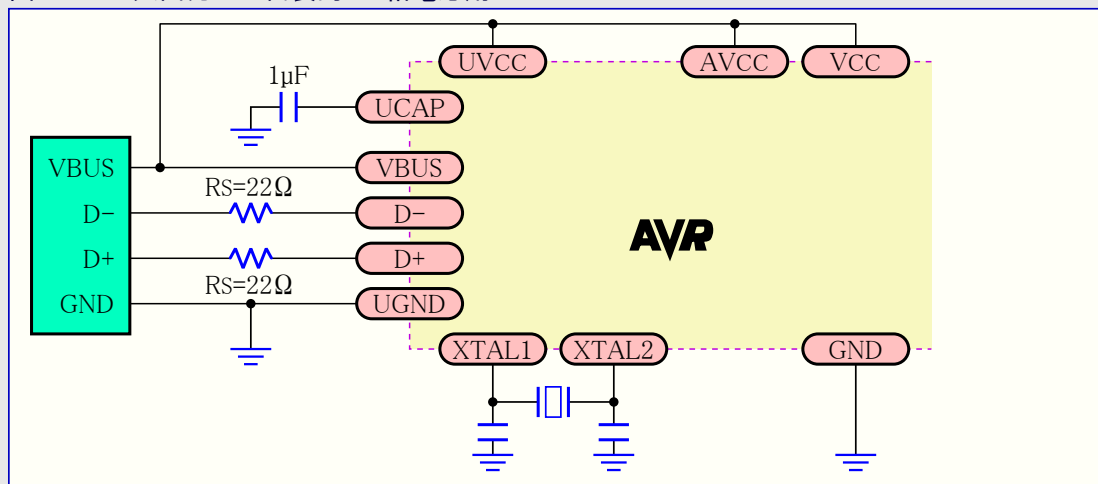
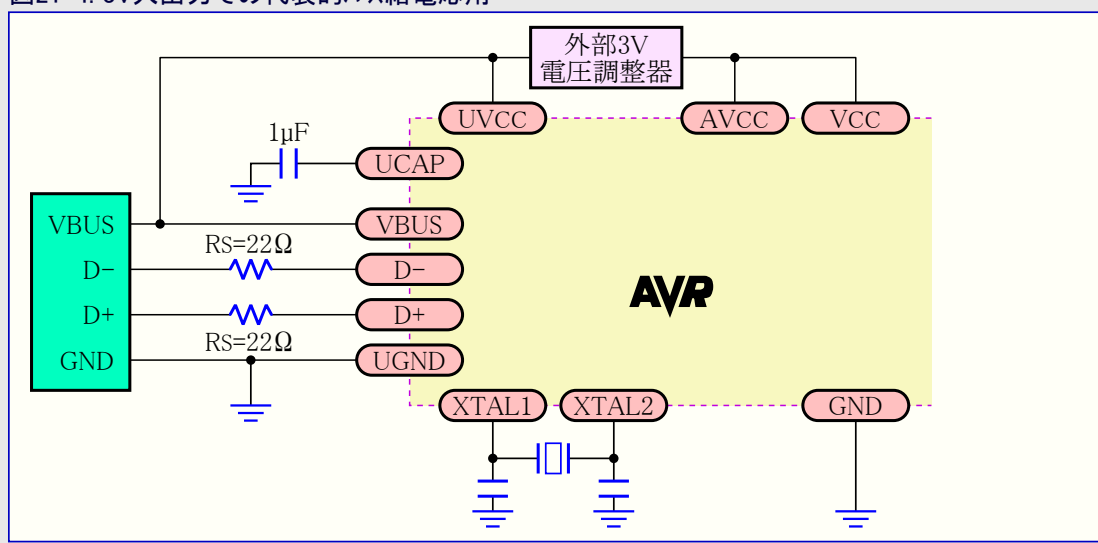


図21-4. 3V入出力での代表的バス給電応用



21.3.2. 自己給電装置

図21-5. 3.4~5.5V入出力での代表的自己給電応用

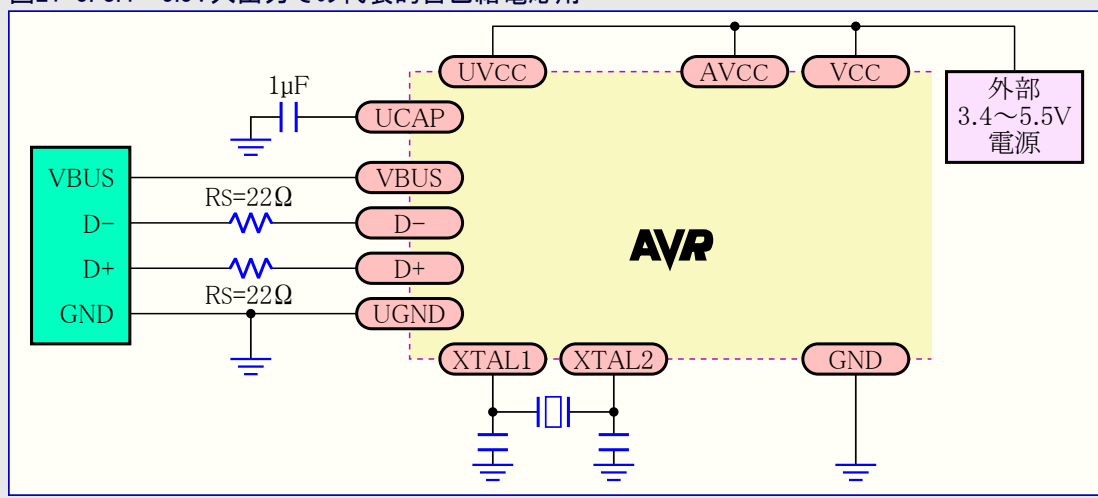
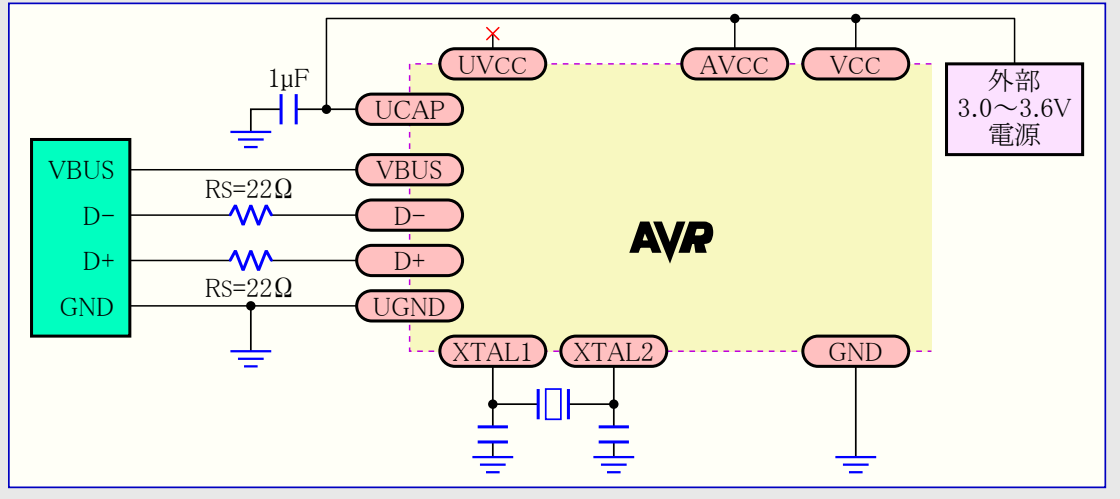


図21-6. 3.0～3.6V入出力での代表的自己給電応用



21.4. クリスタルなし動作

外部部品点数と部品費用を減らすため、USB単位部は低速(Low-speed)動作に於いてPLL用の入力クロック元として内蔵RC発振器で動作するように構成設定することができます。内蔵RC発振器は0～40℃の温度範囲内でUSB低速周波数精度を満足するように工場校正されています。

USB全速(Full-speed)については外部クリスタル用発振器または外部クロック入力だけが使えます。

21.5. 設計の指針

- USBデータ線の直列抵抗器は値が22Ω(±5%)であるべきです。
- USB入力コネクタ(またはケーブル付き装置の場合、ケーブル接続)からUSBマイクロコントローラへの配線は可能な限り短く、差動配線規則(斜めの累積を避け、可能な限り近く、同じ長さ)に従うべきです。
- 外部の妨害によるUSBパット[®]の損傷を防ぐために、電圧瞬間変動/ESD消去器も使われ得ます。
- UCAPコンデンサは正しい動作のために値が1μF(±10%)であるべきです。
- VBUS線上には10μFコンデンサが強く推奨されます。

21.6. 一般的な動作

21.6.1. 序説

USB制御器は以下によって生成されたハードウェアリセット後にリセットされ、禁止されます。

- 電源ONリセット
- 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)
- JTAGリセット

しかし、別に利用可能な任意のリセット元は以下です。

- **USBリセット終了(EOR: End Of Reset)**

この場合、USB制御器はリセットされますが、**禁止されません**(故に装置は接続を保ちます)。

21.6.2. 電源ONとリセット

次の構成図は電源ONでのUSB制御器主状態を説明します。

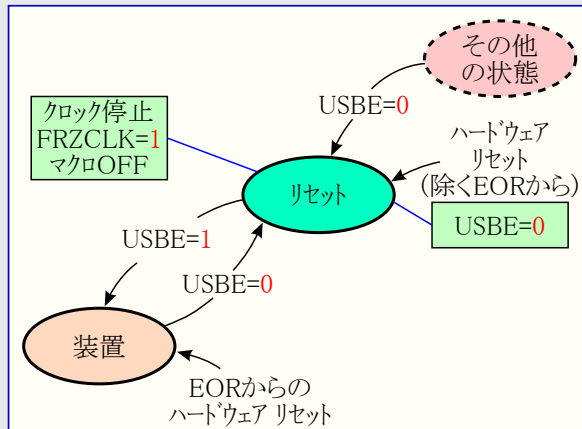
ハードウェア リセット後のUSB制御器状態は「リセット」です。この状態は以下です。

- USB制御(USBCON)レジスタのUSB全体許可(USBE)は設定(1)されません。(=解除(0))
- 電力消費を最小とするためにUSB制御器クロックは停止されます(USBCONレジスタのUSBクロック停止(FRZCLK)=1)。
- USB制御器は禁止されます。
- USBハットは休止(Suspend)動作です。
- 装置のUSB制御器内部状態はリセットです。

USB設定(=1)後、USB制御器は装置状態へ移行します。制御器は‘アイドル’です。

USB制御器はUSB Eの解除(0)によって何時でも「停止」にできます。実際にはUSB Eの解除(0)がUSB制御器上でハードウェアリセットとして働きます。

図21-7. リセット後のUSB制御器状態



21.6.3. 割り込み

2つの割り込みベクタがUSBインターフェースに割り当てられます。

USB制御器は各エンドポイントへの関係データ転送に関連するUSBエンドポイント事象との違いでUSB一般事象を区別します。

図21-8. USB割り込み概要

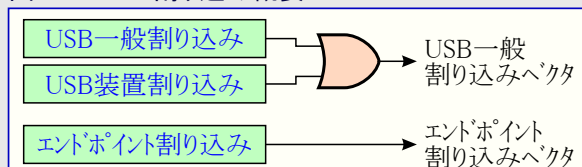
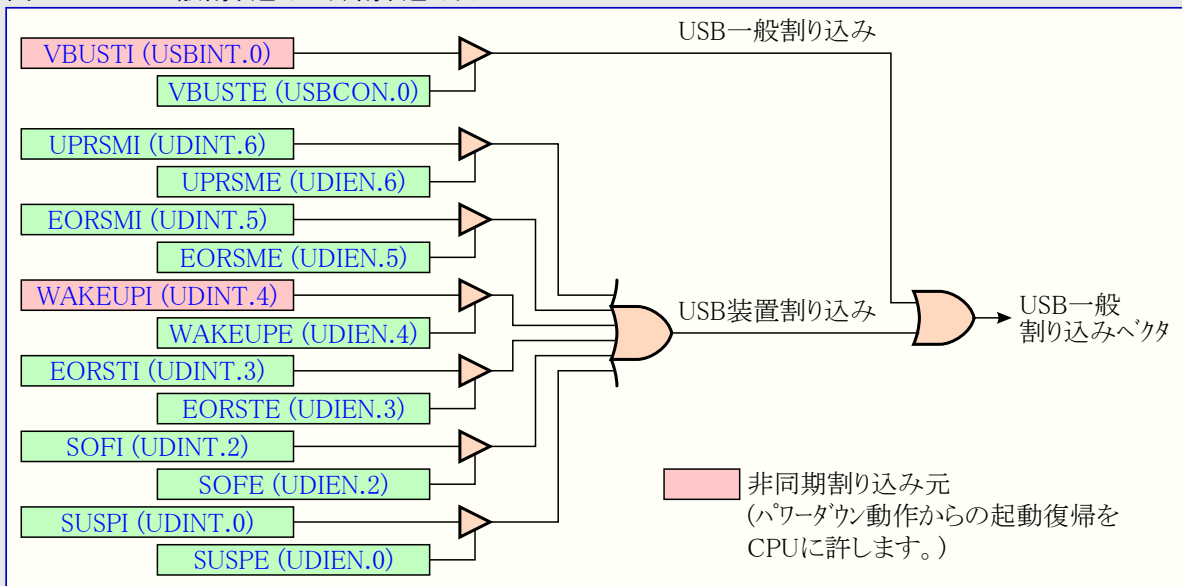


図21-9. USB一般割り込みベクタ割り込み元

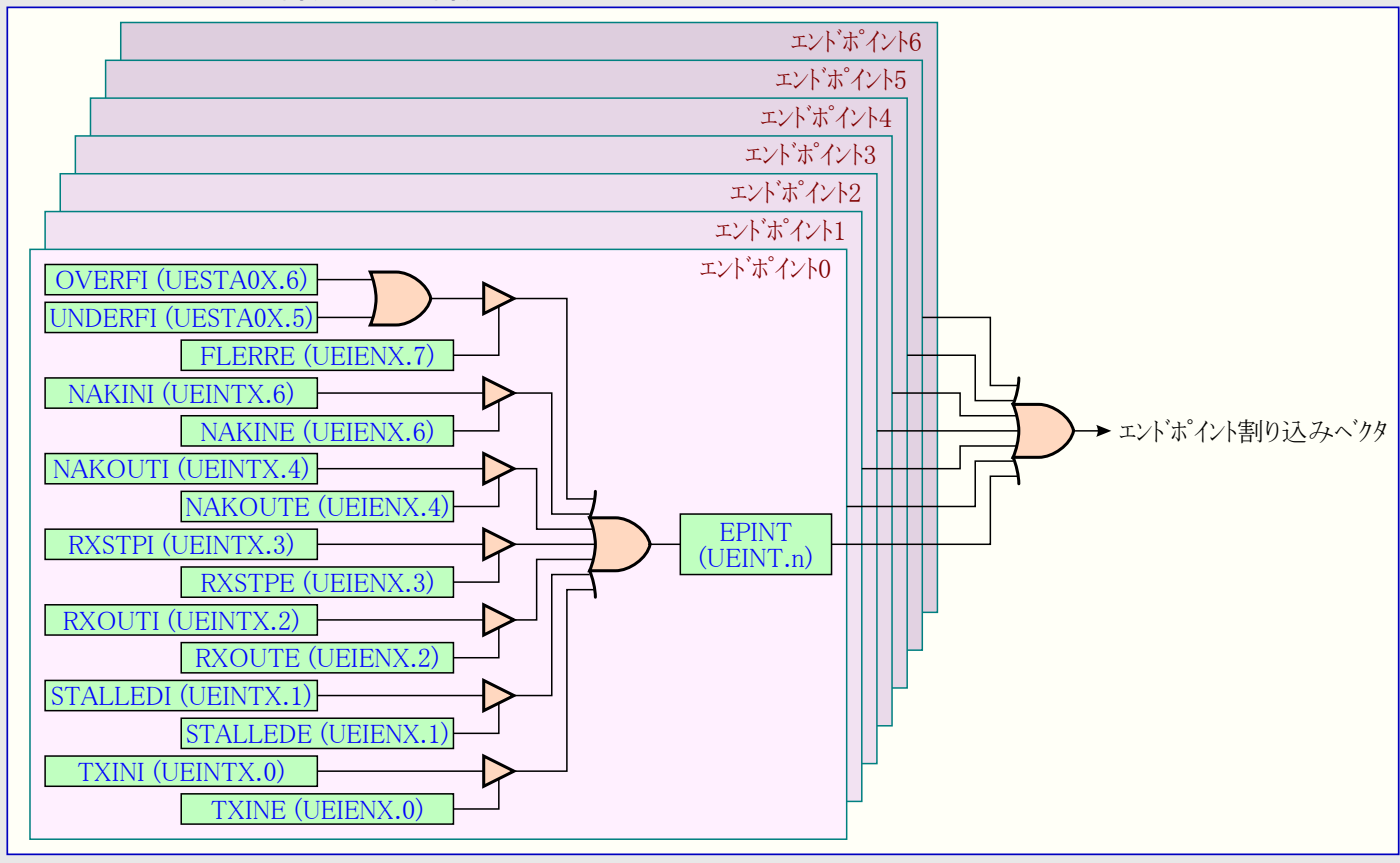


以下を除き、これら割り込みの殆ど全てはUSBクロックが許可される(FRZCLK=0)場合にだけ検出される時間相対事象です。

- VBUS挿抜検出(挿入、取り外し)
- データ線上で状態変化が検出されるといつも起動されるWAKEUP割り込み

このWAKEUP非同期割り込みは一般的にUSBが休止(Suspend)状態へ移行された後に、パワーダウン動作でのデバイスの起動復帰を許します。

図21-10. USBエンドポイント割り込みベクタ割り込み元



各エンドポイントはフラグに関して8つの割り込み元を持ち、各源は対応するエンドポイント割り込み起動を許可または不許可にすることができます。エンドポイントに対して最低1つの(割り込み)元が割り込み起動を許可されるなら、対応する事象はプログラムをUSBエンドポイント割り込みベクタへ分岐させます。使用者は**エンドポイント割り込み要求(UEINT)レジスタ**を読むことで割り込みの元(エンドポイント)を判定し、そして様々なフラグのポーリングによって検知される事象を扱います。

21.7. 電力動作種別

21.7.1. アイドル動作

この動作ではCPUコアが停止されます(CPUクロック停止)。アイドル動作はUSB制御器が走行中、非走行中のどちらでも使われます。CPUは何れかのUSB割り込みで起動復帰します。

21.7.2. パワーダウ動作

この動作では発振器が停止され、全てのクロック(CPUと周辺機能部)が停止します。USB制御器は以下の時に起動復帰します。

- CPU起動復帰割り込み(WAKEUPI)起動
- VBUS遷移割り込み(VBUSTI)起動

21.7.3. USBクロック停止

ファームウェアはUSB制御器のクロックを停止するUSBCONレジスタの**USBクロック停止(FRZCLK)ビット**の設定(1)によって電力消費を低減することができます。FRZCLKが設定(1)されている時は次のレジスタが未だアクセス可能です。

- USBCON, USBSTA, USBINT
- UDCON (分離(DETACH), ...)
- UDINT
- UDIEN

更にFRZCLKが設定(1)されている時は次の割り込みだけが起動され得ます。

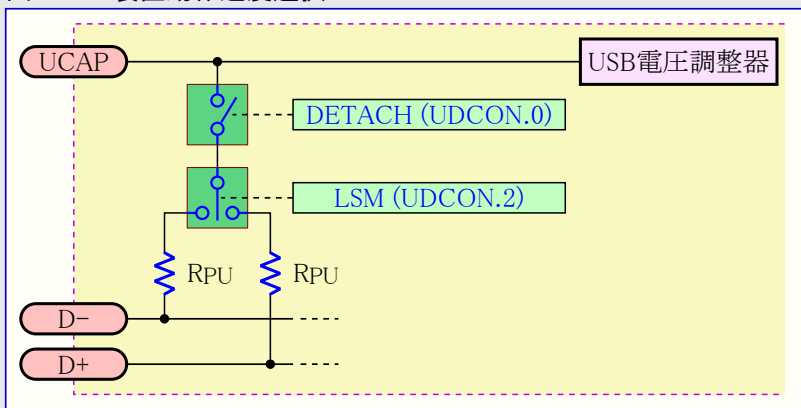
- CPU起動復帰割り込み(WAKEUPI)
- VBUS遷移割り込み(VBUSTI)

21.8. 速度制御

速度選択(全速(Full-speed)または低速(Low-speed))はD+/D-のプルアップに依存します。

USB装置制御(UDCON)レジスタの低速/全速装置動作(LSM)ビットはD- (低速動作)またはD+ (全速動作)データ線上の内部プルアップ選択を許します。

図21-11. 装置動作速度選択



21.9. メリ管理

本制御器は以下のメモリ割り当て管理だけを支援します。

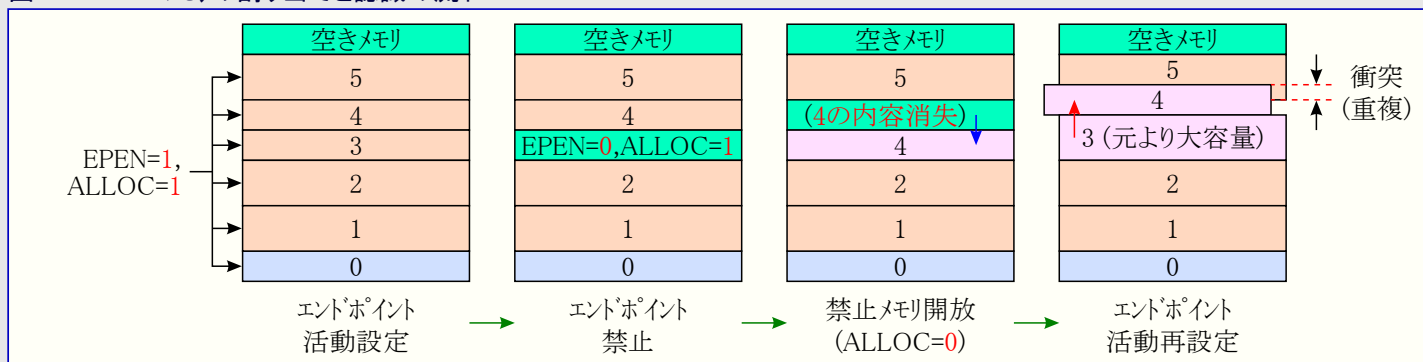
エンドポイントの予約は(エンドポイント0から最終エンドポイントへ)順に増やす場合にだけ行えます。従ってファームウェアはこれと同じ順で設定すべきです。

エンドポイント k の予約はそれ(エンドポイントX構成1(UECFG1X)レジスタ)のエンドポイントメモリ割り当て(ALLOC)ビットが設定(1)される時に行われます。そしてハードウェアがメモリを割り当て、それをエンドポイント $k-1$ と $k+1$ 間に挿入します。エンドポイント $k+1$ のメモリは上へ移動してそのデータは失われます。 $k+2$ とそれよりも上のエンドポイントのメモリが移動しないことに注意してください。

エンドポイント許可(EPEN)の解除(0)はそれの設定(EPsize, EPBK)やALLOCビットのどちらも解除(0)しません。メモリを開放するためにファームウェアはALLOCビットを解除(0)すべきです。するとエンドポイント $k+1$ のメモリは自動的に下へ移動します。 $k+2$ とそれよりも上のエンドポイントのメモリが移動しないことに注意してください。

下図は代表例でのUSBメモリの割り当てと認識を図示します。

図21-12. USBメモリの割り当てと認識の流れ



- 最初にエンドポイント0～5は昇順で設定されます。各メモリはDPRAM内に予約されます。
- そしてエンドポイント3が禁止(EPEN=0)されますが、メモリの予約は制御器によって内部的に保持されます。
- ALLOCビットが解除(0)され、エンドポイント4が下へ移動しますが、エンドポイント5は移動しません。
- 最後はファームウェアがより大きな容量でエンドポイント3の再設定を選ぶ場合です。制御器はエンドポイント2の後にメモリを予約し、エンドポイント4を自動的に移動します。エンドポイント5は動かず、エンドポイント4と5の共通の領域を使うことで衝突(重複)が出現します。これらのエンドポイントのデータは潜在的に失われます。

以下に注意してください。

- エンドポイント0のデータは、より高位のエンドポイントの活動設定または非活動設定でも決して失われません。このデータはそれが非活性設定にされた場合に失われます。
- 同じ設定での同一エンドポイントの非活動設定と再活動設定は、より高位のエンドポイントの移動を引き起こしません。これらのエンドポイントに対してデータは保護されます。
- エンドポイントX状態0(UESTA0X)レジスタのエンドポイント構成状態(CFGOK)フラグはメモリ割り当てで衝突(重複)した場合でもハードウェアによって設定(1)されます。

21.10. ハット休止

次図はハットの動きを図示します。

- アイドル動作でハットは低電力消費動作に置かれます。
- 活動動作でハットは動作します。

休止状態を示すUSB装置割り込み要求(UDINT)レジスタの休止(Suspend)割り込み要求(SUSPI)フラグはUSBバス上で検出されます。このフラグは自動的にUSBハットをアイドルにします。非アイドル事象の検出はUDINTレジスタのCPU起動復帰割り込み要求(WAKEUPI)フラグを設定(1)し、USBハットを起動復帰します。

更にハットはUSB装置制御(UDCON)レジスタの分離(DETACH)ビットが設定(1)されている場合にもアイドル動作にして置けます。これはDETACHビットが解除(0)されると、活動動作に復帰します。

図21-13. ハットの動き

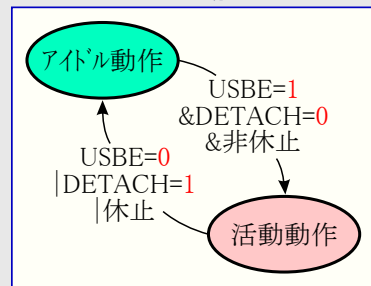
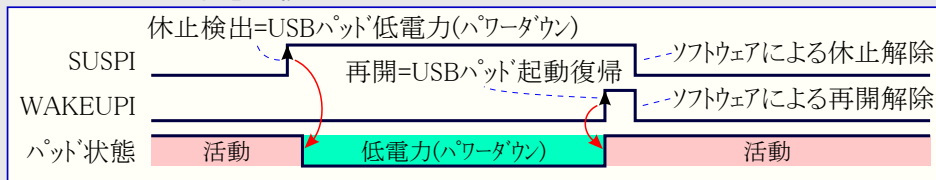


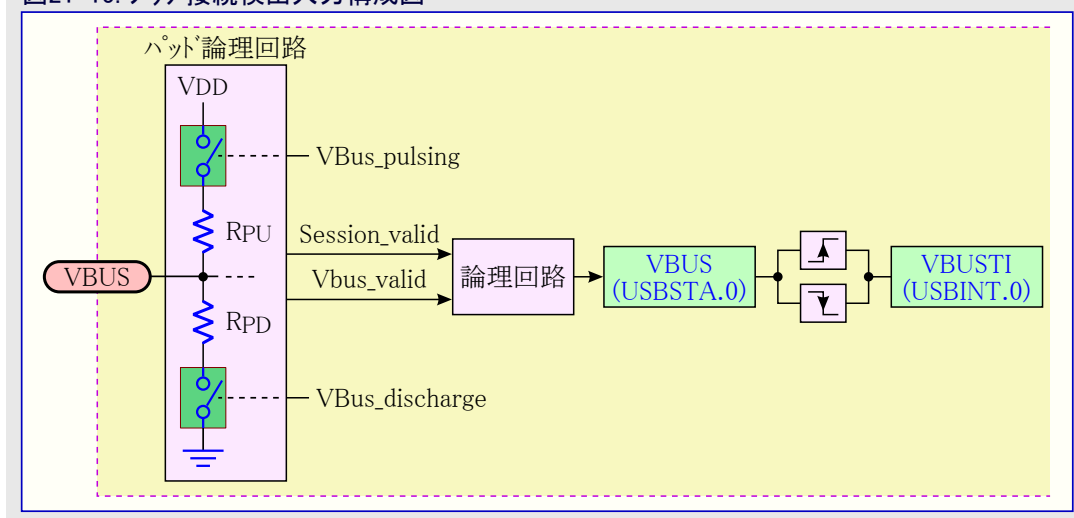
図21-14. ハットの状態遷移



21.11. プラグ接続検出

USB接続は以下の構成でVBUSハットによって検出されます。

図21-15. プラグ接続検出入力構成図



VBUSハットの制御回路はVBUS電圧水準に関する信号を出力します。

- Session_valid信号はVBUSハット上の電圧が1.4V以上の時にHigh活性です。1.4Vより低いなら、信号は非活性です。
- VBUS状態ビットはSession_valid信号が活性(VBUS>1.4V)の時に設定(1)されます。
- USB割り込み要求(USBINT)レジスタのVBUS遷移割り込み(VBUSTI)フラグはVBUS状態変化毎に設定(1)されます。
- USB周辺機能はVBUSビットが設定(1)されていない間、バスへ接続することができません。

21.12. USBソフトウェア操作種別

USB動作種別に応じて、ソフトウェアは以下の操作のいくつかを実行しなければなりません。

USBインターフェース電源ON

- USBハット電圧調整器電源ON
- PLLインターフェース設定
- PLL許可
- PLL固定化(ロック)検査
- USBインターフェース許可
- USBインターフェース設定 (USB速度、エンドポイント設定、...)
- USB VBUS接続情報待機
- USB装置取り付け(接続)

USBインターフェース電源OFF

- USBインターフェース分離
- USBインターフェース禁止
- PLL禁止
- USBハット電圧調整器禁止

USBインターフェース休止

- 休止ビット解除(=0)
- USBクロック休止設定
- PLL禁止
- 休止形態を抜けるための割り込みを確実に許可してください。
- MCUを休止形態へ移行します。

USBインターフェース再開

- PLL許可
- PLL固定化(ロック)待機
- USBクロック休止解除
- 再開情報解除

21.13. USB一般用レジスタ

21.13.1. UHWCON – USBハードウェア制御レジスタ (USB Hardware Control Register)

ビット (\$D7)	7	6	5	4	3	2	1	0	
	–	–	–	–	–	–	–	UVREGE	UHWCON
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)

これらのビットからの読み込み値は常に0です。これらのビットを変更してはいけません。

- ビット0 – UVREGE : USBパッド電圧調整器許可 (USB pad regulator Enable)

USBパッド用電圧調整器を許可するには設定(1)してください。USBパッド用電圧調整器を禁止するには解除(0)してください。

21.13.2. USBCON – USB制御レジスタ (USB Control Register)

ビット (\$D8)	7	6	5	4	3	2	1	0	
	USBE	–	FRZCLK	OTGPADE	–	–	–	VBUSTE	USBCON
Read/Write	R/W	R	R/W	R/W	R	R	R	R/W	
初期値	0	0	1	0	0	0	0	0	

- ビット7 – USBE : USB全体許可 (USB macro Enable Bit)

USB制御器を許可するには設定(1)してください。USBクロック入力とUSB送受信器を禁止し、USB制御器をリセットするには解除(0)してください。

- ビット6 – Res : 予約 (Reserved)

このビットからの読み込み値は常に0です。このビットを設定(1)してはいけません。

- ビット5 – FRZCLK : USBクロック停止 (Freeze USB Clock Bit)

クロック入力を禁止(再開検出は未だ活動)するには設定(1)してください。これは電力消費を低減します。クロック入力を許可するには解除(0)してください。

- ビット4 – OTGPADE : OTGパッド許可 (OTG Pad Enable)

OTGパッドを許可するには設定(1)してください。OTGパッドを禁止するには解除(0)してください。現実のOTGパッドはVBUSパッドです。例えUSB全体許可(USBE)=0でも、このビットが設定(1)/解除(0)できることに留意してください、それは例えUSB制御器が禁止でもVBUS検出を許します。USB動作(接続、送信など)を許すには本パッドが許可されなければなりません(訳注:共通性から本行追加)。

- ビット3～1 – Res : 予約 (Reserved)

これらのビットからの読み込み値は常に0です。これらのビットを設定(1)してはいけません。

- ビット0 – VBUSTE : VBUS遷移割り込み許可 (VBUS Transition Interrupt Enable Bit)

VBUS遷移割り込み生成を許可するには、このビットを設定(1)してください。VBUS遷移割り込み生成を禁止するには、このビットを解除(0)してください。

21.13.3. USBSTA – USB状態レジスタ (USB Status Register)

ビット (\$D9)	7	6	5	4	3	2	1	0	
	–	–	–	–	–	–	ID	VBUS	USBSTA
Read/Write	R	R	R	R	R	R	R	R	
初期値	0	0	0	0	0	0	1	0	

- ビット7～2 – Res : 予約 (Reserved)

これらのビットからの読み込み値は常に0です。これらのビットを設定(1)してはいけません。

- ビット1 – ID : UIDピンフラグ (UID pin Flag)

このビットは常に1として読み、(UIDピンの値を示す)AT90USB64/128との共通性が維持されます。

- ビット0 – VBUS : VBUSフラグ (VBus Flag)

このビットからの読み込み値はVBUSピンの状態を示します。本ビットは装置動作で応用でのUSBバス接続状態の監視に使えます。より多くの詳細については172頁の「フラグ接続検出」項をご覧ください。

21.13.4. USBINT – USB割り込み要求レジスタ (USB Interrupt Request Register)

ビット (\$DA)	7	6	5	4	3	2	1	0	
	–	–	–	–	–	–	–	VBUSTI	USBINT
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)

これらのビットからの読み込み値は常に0です。これらのビットを設定(1)してはいけません。

- ビット0 – VBUSTI : VBUS遷移割り込み要求フラグ (VBUS Transition Interrupt Flag)

VBUSパッド上で遷移(HighからLow、またはLowからHigh)が検出された時にハードウェアによって設定(1)されます。

これはソフトウェアによって解除(0)されるべきです。

22. USB装置動作

22.1. 序説

USB装置制御器は全速(Full-speed)と低速(Low-speed)のデータ転送を支援します。既定の制御(Control)エンドポイントに加えて別に、制御(Control)、大量(Bulk:バルク)、割り込み(Interrupt)、等時(Isochronous:アイソクロノス)動作に設定できる6つのエンドポイントを提供します。

- エンドポイント0 : 既定制御エンドポイント、64バイトまでのFIFO容量設定可能
- エンドポイント1 : 交互動作で256バイトまでのFIFO容量設定可能
- エンドポイント2～6 : 交互動作で64バイトまでのFIFO容量設定可能

制御器はアイドル動作で始まります。この動作ではパットの電力消費が最小に低減されます。

22.2. 電源ONとリセット

次の構成図は電源ONでのUSB装置制御器の主な状態を説明します。

装置制御器のリセット状態は次のとおりです。

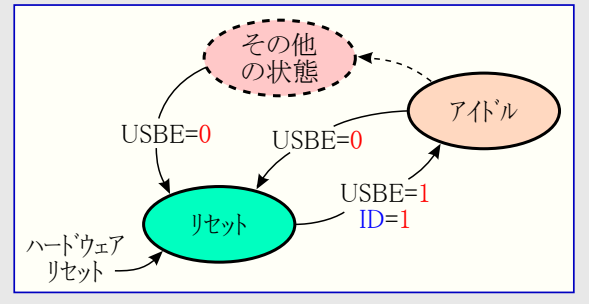
- 電力消費を最小とするために全体クロックが停止されます(USB制御(USBCON)レジスタのUSBクロック停止(FRZCLK)=1)。
- USB装置制御器の内部状態はリセットです(全てのレジスタがそれらの既定値でリセットされます。USB装置制御(UDCON)レジスタの分離(DETACH)ビットが設定(1)されることに注意してください)。
- エンドポイントのバンクはリセットされます。
- D+とD-のプルアップは非活性(分離(Detach)動作)にされます。

D+とD-のプルアップは分離(DETACH)ビットが解除(0)されると直ぐに活性(有効)にされ、VBUSが現れます。

リセット後の制御器は**最小電力消費**でのアイドル状態で、この状態への移行にPLLが動作されている必要はありません。

USB装置制御器はUSB制御(USBCON)レジスタのUSB全体許可(USBE)ビットの解除(0)によって何時でもリセットにできます(USBインターフェース禁止)。

図22-1. リセット後のUSB装置制御器状態



22.3. エンドポイント リセット

エンドポイントは**エンドポイント リセット(UERST)**レジスタ内でエンドポイントに対応するビット(EPRSTn)の設定(1)によって何時でもリセットできます。このリセットは以下を行います。

- そのエンドポイントの内部状態機構をリセット
- RxとTxのバンクが消去され、それらの内部ポインタが復元(初期化)されます。
- エンドポイントX割り込み要求(UEINTX)レジスタ、エンドポイントX状態0(UESTA0X)レジスタ、エンドポイントX状態1(UESTA1X)レジスタはそれらのリセット値に復元(初期化)されます。

交互データ番号領域は無変化に留まります。

その他のレジスタは無変化に留まります。

エンドポイント設定は活性(有効)に留まり、エンドポイントは未だ許可されています。

エンドポイントリセットは**CLEAR FEATURE** USB命令への応答としてデータ番号交互命令の解除(エンドポイントX制御(UECONX)レジスタの交互データ番号リセット(RSTDT)ビット)と連携して行えます。

22.4. USB リセット

USB線上でUSBリセット(最小2.5μs幅でのSEO状態)が検出されると、制御器によって次の操作が行われます。

- 全てのエンドポイントが禁止されます。
- 既定制御エンドポイントは構成(設定)状態に留まります(より多くの詳細については「**エンドポイント リセット**」をご覧ください)。

CPUハードウェア リセット機能が活性(USB装置制御(UDCON)レジスタのCPUリセット(RSTCPU)ビットが設定(1))にされた場合、USB制御器の禁止を除いてCPUコアへリセットが生成されます(必然的に一般的なUSBリセット後で接続状態に留まるのと同じ動きになります)。この機能は装置の信頼性強化に使えます。

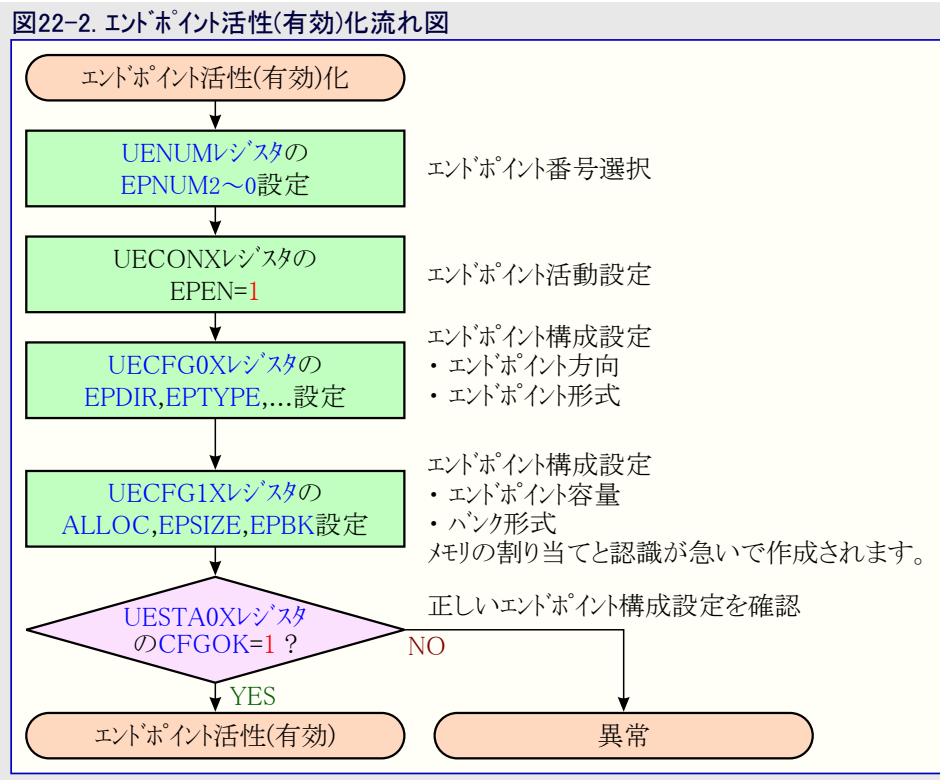
22.5. エンドポイント選択

CPUによって何らかの操作が行われる前にエンドポイントが先に選ばれなければなりません。これはCPUによって管理されるエンドポイント番号での**エンドポイント番号(UENUM)レジスタのエンドポイント番号(EPNUM2～0)ビット**設定によって行われます。

その後CPUは各種エンドポイントレジスタとデータへアクセスすることができます。

22.6. エンドポイントの活性化(有効設定)

エンドポイントは**エンドポイントX制御(UECONX)レジスタ**の**エンドポイント許可(EPEN)ビット**が**設定(1)**されない限り、リセット下に保たれます。以下の流れはエンドポイントを活性(有効)にするために尊重されなければなりません。



エンドポイントが不正に設定(CFGOK=0)されている限り、ハードウェアはホストによって送られたパケットに応答しません。

構成状況(CFGOK)フラグはエンドポイント容量指定がDPRAM容量よりも大きいなら、**設定(1)**されません。

エンドポイント許可(EPEN)ビットの解除(0)はエンドポイントリセットとして働きます(より多くの詳細については176頁の「**エンドポイントリセット**」をご覧ください)。これは次の操作も行います。

- エンドポイントの構成設定は保持されます(EPSIZE, EPNK, ALLOC保持)。
- 交互データ番号領域をリセットします。
- エンドポイントに関連するDPRAMは未だ予約されています。

メモリ割り当てと認識についてのより多くの詳細に関しては171頁の「**メモリ管理**」をご覧ください。

22.7. アドレス設定

USB装置アドレスはUSB規約に従って設定されます。

- 電源投入後、USB装置はアドレス0で応答します。
- ホストは**設定(SETUP)命令**を送ります(SET_ADDRESS(addr))。
- ファームウェアはこの要求を扱い、**装置アドレス(UDADDR)レジスタ**の**USB装置アドレス(UADD6~0)**内にそのアドレスを記録しますが、**アドレス許可(ADDEN)ビット**は解除(0)を保持してください。
- USB装置ファームウェアは0バイトの**IN命令**を送ります(IN 0 : 0長パケット)。
- その後、ファームウェアはADDENビットの**設定(1)**によってUSB装置アドレスを許可にできます。制御器によって受け付けられるアドレスはUADD6~0に格納された1つ(のアドレス)だけです。

ADDENとUADD6~0ビットは同時に書かれるべきではありません。

UADD6~0ビットは電源投入またはリセット後、既定アドレス\$00を内包します。

ADDENビットは以下の状態でハードウェアによって解除(0)されます。

- 電源投入リセット後
- USBリセットが受信された時
- 制御器が禁止された時(USB制御(USBCON)レジスタの**USB全体許可(USBE)=0**)

この(ADDEN)ビットが解除(0)されると、既定装置アドレス\$00が使われます。

22.8. 休止、起動復帰と再開

USB線が非活性で3ms間後、制御器は全速(Full-speed)動作に切り替え、(許可されていれば)休止(Suspend)割り込み(SUSPI)を起動します。そしてファームウェアはUSB制御(USBCON)レジスタのUSBクロック停止(FRZCLK)ビットを設定(1)できます。

CPUは更により低い電力消費にさせるのにアイドル動作へ移行することもできます。

休止(Suspend)動作からの回復には2つの方法があります。

- 1つ目はFRZCLKビットを解除(0)することです。これはCPUがアイドル動作でない場合に可能です。
- 2つ目の方法はCPUがアイドル動作の場合で、CPU起動復帰割り込み(WAKEUPI)を許可(WAKEUPE=1)することです。そして制御器によって非アイドル信号が見つかると直ぐに、WAKEUPI割り込みが起動されます。ファームウェアはその後に転送を再開するためにFRZCLKビットを解除(0)すべきです。

休止割り込み(SUSPI)とCPU起動復帰割り込み(WAKEUPI)間に関連はなく、WAKEUPI割り込みはデータ線上の非アイドル形態で直ぐに起動されます。従ってWAKEUPI割り込みは例えば制御器が休止動作でなくても起き得ます。

WAKEUPI割り込みが起動されるとき、SUSPI割り込みが既に設定(1)されている場合、それはハードウェアによって解除(0)されます。

SUSPI割り込みが起動されるとき、WAKEUPI割り込みが既に設定(1)されている場合、それはハードウェアによって解除(0)されます。

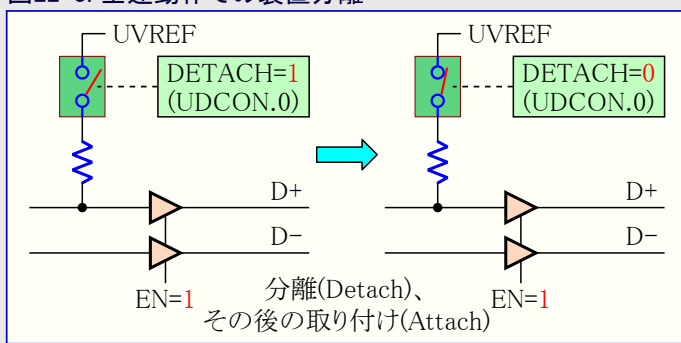
22.9. 分離 (切り離し)

USB装置制御(UDCON)レジスタの分離(DETACH)ビットのリセット値は1です。

単にDETACHビットを設定(1)し、そして解除(0)することによって装置を再列挙(再認識)することが可能です(しかし、ファームウェアはms単位での反発除去遅延を考慮しなければなりません)。

- DETACHビットの設定(1)は(選ばれた全速(Full-speed)または低速(Low-speed)動作に応じて)、D+またはD-パッド上のプルアップを切断します。その後のDETACHビットの解除(0)がD+またはD-パッド上のプルアップを接続します。

図22-3. 全速動作での装置分離



22.10. 遠隔起動復帰

遠隔起動復帰(Remote Wake-up: または上方向再開(Upstream resume))機能は、装置によって自発的に送ることが許された唯一の操作です。とにかく、それを行うため、装置は最初にホストからDEVICE_REMOTE_WAKEUP要求が受信されるべきです。

- 最初にUSB制御器は信号線の休止(Suspend)状態を検知しなければなりません。遠隔起動復帰は休止(Suspend)割り込み(SUSPI)フラグが設定(1)されている時にだけ送ることができます。
- ファームウェアはその後に上方向再開(Upstream resume)列を送るためにUSB装置制御(UDCON)レジスタの遠隔起動(RMWKUP)ビットを設定(1)できます。
- 制御器が上方向再開(Upstream resume)送信を始めると、(許可されていれば)上方向再開割り込み(UPRSMI)が起動されます。SUSPIはハードウェアによって解除(0)されます。
- 上方向再開(Upstream resume)の最後でRMWKUPビットがハードウェアによって解除(0)されます。
- 制御器がホストからの良好な再開終了(End Of Resume)を検知すると、(許可されていれば)再開終了割り込み(EORSMI)が起動されます。

22.11. 不能応答(STALL)要求

各エンドポイントに対する不能応答(STALL)管理は2ビット(と1フラグ)を使って行われます。

- STALLRQ (不能応答要求(STALL Request)許可)
- STALLRQC (不能応答要求(STALL Request)禁止)
- STALLEDI (不能応答送信(STALL sent)割り込み)

次の要求で不能応答(STALL)ハンドシェイクを送るにはエンドポイントX制御(UECONX)レジスタのSTALLRQビットが設定(1)されなければなりません。後続する全ての要求はSTALLRQCビットが設定(1)されるまで不能応答(STALL)でハンドシェイクされます。

STALLRQCの設定(1)はSTALLRQビットを自動的に解除(0)します。STALLRQCビットもソフトウェアによって設定(1)された後、ハードウェアによって直ちに解除(0)されます。従ってファームウェアはこのビットを決して設定(1)として読まないでしう。

不能応答(STALL)ハンドシェイクが送られる度にエンドポイントX割り込み要求(UEINTX)レジスタのSTALLEDIフラグがUSB制御器によって設定(1)され、(許可されていれば)エンドポイントn(X)割り込み(EPINTn)が起動されます。

到着パケットは破棄されます(OUTデータ受信割り込み要求(RXOUTI)と読み書き可(RWAL)フラグは設定(1)されません)。

ホストはその後に不能応答(STALL)をリセットするための命令を送信し、ファームウェアは直ぐにSTALLRQCビットを設定(1)してエンドポイントをリセットしなければなりません。

22.11.1. 制御エンドポイントに対する特別な考慮

SETUP要求は常に肯定応答(ACK)で応答されます。

不能応答(STALL)要求が制御エンドポイントで設定され、SETUP要求が起きると、そのSETUP要求はACKで応答されなければならない、**不能応答(STALL)ハンドシェーク要求(STALLRQ)ビット**と**不能応答(STALL)ハンドシェーク送信完了割り込み要求(STALLEDI)フラグ**は自動的にリセット(=0)されます(**SETUP受信割り込み要求(RXSTPI)=1**,**送信可割り込み要求(TXINI)=0**,**不能応答(STALL)ハンドシェーク送信完了割り込み要求(STALLEDI)=0**,...)。

この管理は列挙管理処理を簡単にします。命令が未支援または異常を含む場合、ファームウェアは不能応答(STALLRQ)ビットを設定(1)して次のSETUP要求を待つ主作業へ戻ることができます。

この機能は**GET_DESCRIPTOR**に対する追加状態送信をし得る第8節検査に適合します。ファームウェアは状態受信後、直ちに不能応答要求(STALLRQ)ビットを設定(1)します。全ての追加状態は以降のSETUP要求まで、自動的に不能応答(STALL)にされます。

22.11.2. 不能応答(STALL)ハンドシェークと再試行機構

再試行機構は不能応答(STALL)ハンドシェークよりも高い優先順位を持ちます。不能応答(STALL)ハンドシェークは不能応答(STALL)ハンドシェーク要求(STALLRQ)ビットが設定(1)され、且つ再試行の必要がない場合に送信されます。

22.12. 制御エンドポイント管理

SETUP要求は常に肯定応答(ACK)で応答されます。新規のSETUPパケットが受信されると、(許可されているならば)**SETUP受信割り込み要求(RXSTPI)**が起動されます。**OUT受信割り込み要求(RXOUTI)**は起動されません。

エンドポイントX割り込み要求(UEINTX)レジスタのFIFO制御(FIFOCON)ビットと**読み書き可(RWAL)フラグ**は制御エンドポイントに無関係です。従ってファームウェアはこのエンドポイントでそれらを決して使ってはなりません。読むとそれらの値は常に0です。

制御エンドポイントは以下のビットによって管理されます。

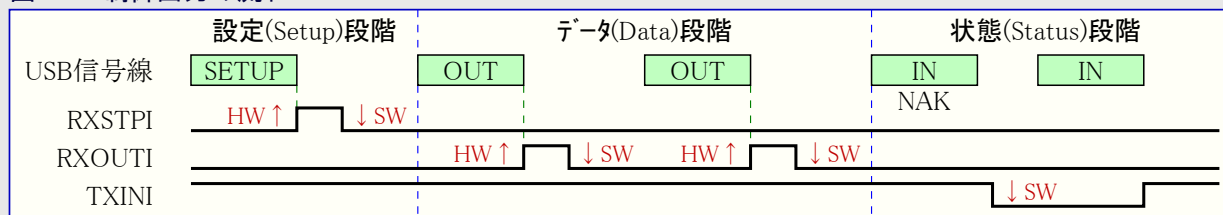
- 新規SETUPパケットが受信されると、RXSTPIが設定(1)されます。パケットの応答と**エンドポイント バンク**の解除のため、これはファームウェアによって解除(0)されるべきです。
- 新規OUTデータが受信されると、RXOUTIが設定(1)されます。パケットの応答と**エンドポイント バンク**の解除のため、これはファームウェアによって解除(0)されるべきです。
- バンクが新規INパケット受け入れ準備可の時に**送信可割り込み要求(TXINI)**が設定(1)されます。パケットの送信と**エンドポイント バンク**の解除のため、これはファームウェアによって解除(0)されるべきです。

22.12.1. 制御出力

次図は制御出力転送を示します。状態段階中、制御器は最初のIN指示票(トークン)で否定応答(NAK)を送信する必要はありません。

- ファームウェアが読まなければならない記述子の正確なバイト数を知っているなら、その後の状態段階が予測でき、次のIN指示子に対して0長パケット(ZLP)を送信できます。
- またはバイトを読み、全バイトがホストによって送信されてしまったことを知らせる、**NAK(否定応答)INパケット受信割り込み要求(NAKINI)**のポーリングができ、処理単位(Transaction)は既に状態段階です。

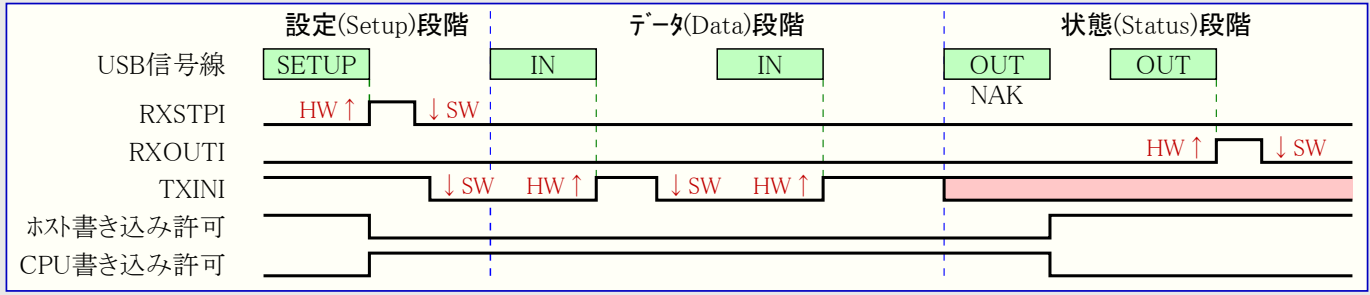
図22-4. 制御出力の流れ



22.12.2. 制御入力

次図は制御入力転送を示します。USB制御器はCPUとホストから同時に起こる書き込み要求を管理しなければなりません。

図22-5. 制御出力の流れ



否定応答(NAK)ハンドシェイクが状態段階命令の最初で常に生成されます。
制御器が状態段階を検知すると、CPUによって書かれた全データが消去され、送信可割り込み要求(TXINI)の解除(0)は無効です。
ファームウェアは送信が完了か、または受信が完了かを調べます。

OUT再試行は常に肯定応答(ACK)で応答されます。この受信は次のとおりです。

- OUT受信割り込み要求(RXOUTI)フラグが設定(1)されます。(OUTデータ受信)
- 送信可割り込み要求(TXINI)フラグが設定(1)されます。(データ送信、新規データ受け入れ可)

ソフトウェアでの手順は次のとおりです。

- 送信可状態設定
- 待機 (送信完了または受信完了)
- 受信完了ならフラグ解除後復帰、送信完了なら継続へ

一旦OUT状態段階が受信されてしまうと、USB制御器はSETUP要求を待ちます。SETUP要求は他の何れの要求よりも高い優先順位を持ち、肯定応答(ACK)で応答しなければなりません。これはSETUPが受信された時にFIFOがリセットされ、他のどのフラグも解除(0)されるべきことを意味します。

警告: バイト計数器は0長OUTパケットが受信された時にリセットされます。ファームウェアはこれに充分注意しなければなりません。

22.13. OUTエンドポイント管理

OUTパケットはホストによって送られます。受け取りを知らせる、または空でないバンクの全てのデータはCPUによって読むことができます。

22.13.1. 概要

エンドポイントが先に構成設定されなければなりません。

現在のバンクが一杯になる毎にOUT受信割り込み要求(RXOUTI)とFIFO制御(FIFOCON)ビットが設定(1)されます。これはOUT受信割り込み許可(RXOUTE)ビットが設定(1)なら、割り込みを起動します。ファームウェアはRXOUTIビットの解除(0)によってUSB割り込みに応答できます。ファームウェアは現在のバンクを開放するためにデータを読み、FIFOCONビットを解除(0)してください。OUTエンドポイントが2重バンクで構成されている場合、FIFOCONビットの解除(0)は次のバンクに切り替えます。そしてRXOUTIとFIFOCONビットは新しいバンクの状態に応じてハードウェアによって更新されます。

RXOUTIは常にFIFOCONの解除(0)に先立って解除(0)されるべきです。

読み書き可(RWAL)フラグは常に現在のバンクの状態を反映します。このフラグはファームウェアがバンクからデータを読める時に設定(1)され、バンクが空の場合にハードウェアによって解除(0)されます。

図22-6. 1つのOUTデータバンク例

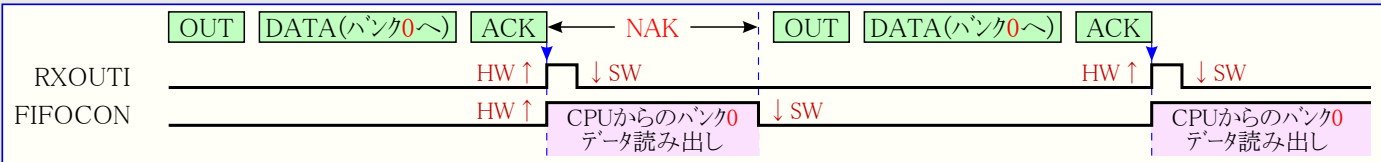
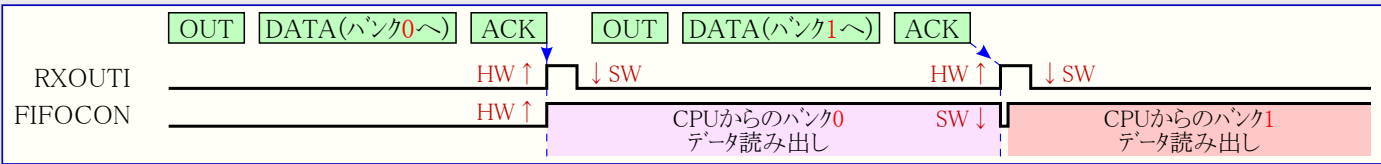


図22-7. 2つのOUTデータバンク例



22.13.2. 詳細内容

データは以下の流れに従い、CPUによって読まれます。

- ホストによってバンクが満たされると、**エンドポイント $n(=X)$ 割り込み(EPINT n)**が起動され、許可(**OUT受信割り込み許可(RXOUTE)=1**)されていれば**OUT受信割り込み要求(RXOUTI)フラグ**が設定(**1**)されます。CPUはソフトウェア構造に応じてRXOUTIフラグまたはFIFO制御(FIFOCON)ビットのポーリングもできます。
- CPUはRXOUTIの解除(**0**)によって割り込みに応答します。
- CPUは現在のバンク内のバイト数(N)を読めます(N=BYCT)。
- CPUは現在のバンクからデータを読めます(UEDATXのN回読み出し)。
- 全データが読まれると、CPUはFIFOCONビットの解除(**0**)によってバンクを開放できます。
- UEDATXのN回読み出し後、**読み書き可(RWAL)フラグ**はハードウェアによって直ちに解除(**0**)されます。

エンドポイントが2つのバンクを使う場合、現在のバンクがCPUによって読まれる一方、2つ目のバンクがホストによって満たされ得ます。その後、CPUがFIFOCONを解除(**0**)すると、次(2つ目)のバンクが既に(読み出し)の準備が整っているかもしれず、(その場合、)RXOUT Iが直ちに設定(**1**)されます。

22.14. INエンドポイント管理

INパケットはホストからのIN要求でUSB装置制御器によって送られます。受け取りを知らせる、または一杯でないバンクの全てのデータはCPUによって書くことができます。

22.14.1. 概要

エンドポイントが先に構成設定されなければなりません。

現在のバンクが自由になると、ハードウェアによって**送信可割り込み要求(TXINI)**が設定(**1**)されます。これは**送信可割り込み許可(TXINE)ビット**が設定(**1**)なら、割り込みを起動します。**FIFO制御(FIFOCON)ビット**が同時に設定(**1**)されます。CPUはFIFO内にデータを書き、データの送信をUSB制御器に許すためにFIFOCONビットを解除(**0**)してください。INエンドポイントが2重バンクで構成されている場合、これは次のバンクへの切り替えも行います。TXINIとFIFOCONビットは次のバンクの状態に関してハードウェアによって更新されます。

TXINIは常にFIFOCONの解除(**0**)に先立って解除(**0**)されるべきです。

読み書き可(RWAL)フラグは常に現在のバンクの状態を反映します。このビットはファームウェアがバンクへデータを書ける時に設定(**1**)され、バンクが一杯の場合にハードウェアによって解除(**0**)されます。

図22-8. 1つのINデータバンク例

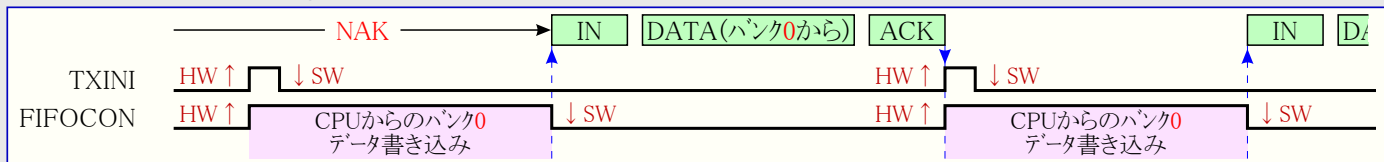
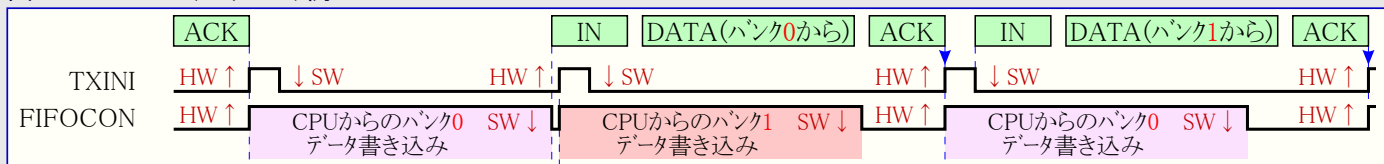


図22-9. 2つのINデータバンク例



22.14.2. 詳細内容

データは以下の流れに従い、CPUによって書かれます。

- バンクが空の時に**エンドポイント $n(=X)$ 割り込み(EPINT n)**が起動され、許可(**送信可割り込み許可(TXINE)=1**)されていれば**送信可割り込み要求(TXINI)**が設定(**1**)されます。CPUはソフトウェア構造の選択に応じてTXINIまたはFIFO制御(FIFOCON)のポーリングもできます。
- CPUはTXINIの解除(**0**)によって割り込みに応答します。
- CPUは現在のバンク内にデータを書けます(UEDATXの書き込み)。
- 全データが書かれると、CPUはFIFOCONの解除(**0**)によってバンクを開放できます。
- UEDATXのN回書き込み後、**読み書き可(RWAL)フラグ**はハードウェアによって直ちに解除(**0**)されます。

エンドポイントが2つのバンクを使う場合、現在のバンクがCPUによって書かれる一方、2つ目のバンクがホストによって読まれ得ます。その後、CPUがFIFOCONを解除(**0**)すると、次(2つ目)のバンクが既に準備が整っている(空)かもしれず、(その場合、)TXINIが直ちに設定(**1**)されます。

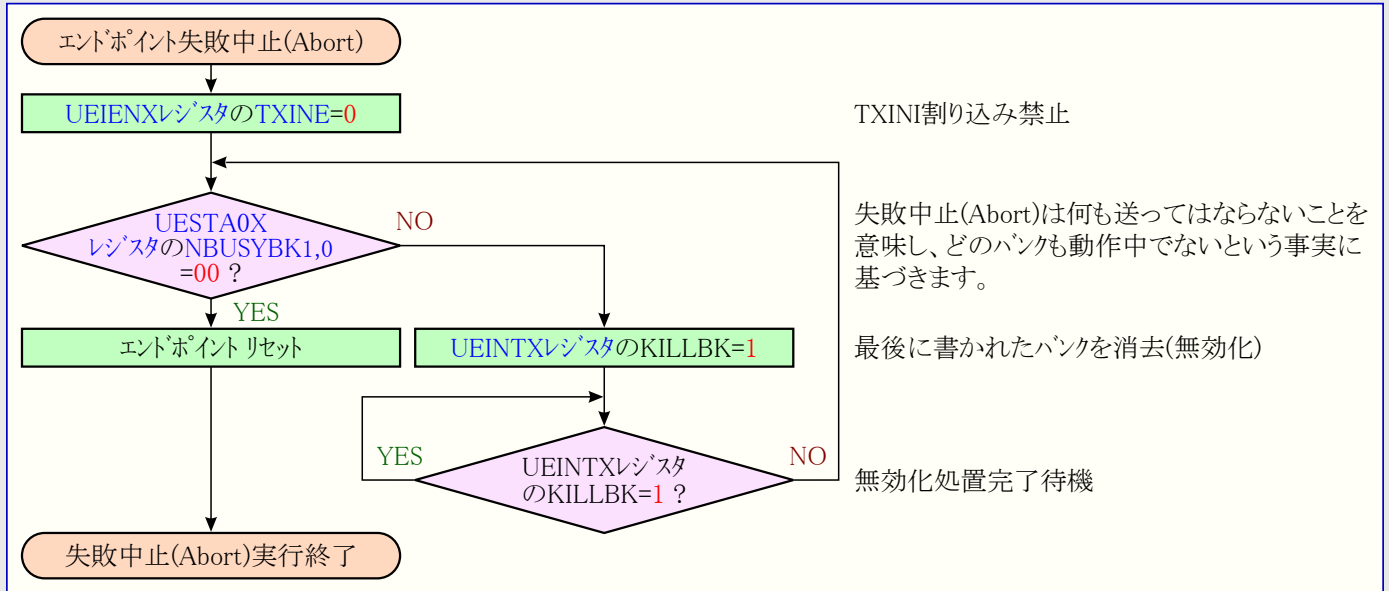
22.14.2.1. 失敗中止 (Abort)

いくつかの状態ではホストによって失敗中止(Abort)段階が生成され得ます。

- 制御転送でIN段階中に0長データOUTパケットが受信された場合
- 等時(Isochronous)転送に於いてINエンドポイントでのIN段階中にOUTエンドポイントでの0長データOUTパケットが受信された場合
- その他 ...

バンク消去(KILLBK)ビットは最後に書かれたバンクを消す(無効にする)のに使われます。この失敗中止を管理する最良の方法は次の操作を実行することです。

図22-10. 失敗中止の流れ図



22.15. 等時(Isochronous)動作

22.15.1. アンダーフロー (Underflow)

アンダーフローはIN段階中にホストが空のバンクを読もうと試みる場合に起き得ます。この場合、アンダーフロー異常割り込み要求(UNDERFI)が起動されます。

アンダーフローはOUT段階中にバンクが既に一杯であるのにホストがパケットを送る場合に起き得ます。代表的にはCPUが充分速くない場合です。そのパケットは失われます。

バンクがデータを与える準備ができていない場合(OUT受信割り込み要求(RXOUTI)=1または読み書き可(RWAL)=1)でだけCPUが読むのが当然なので、CPU側でOUT段階中にアンダーフロー異常になることは不可能です。

22.15.2. CRC異常

CRC異常はOUT段階中にUSB制御器が不正な受信パケットを検出した場合に起き得ます。この場合、不能応答(STALL)ハンドシェイク送信完了割り込み要求(STALLEDI)が起動されます。これはOUT受信割り込み要求(RXOUTI)の起動を妨げません。

22.16. オーバーフロー

制御(Control)、等時(Isochronous)、大量(Bulk)または割り込み(Interrupt)エンドポイントでのオーバーフローは、OUT段階中にホストがパケットに対して小さすぎるバンクに書くことを試みる場合に起き得ます。この場合、(許可なら)オーバーフロー異常割り込み要求(OVERFI)が起動されます。そのパケットは応答され、(許可なら)OUT受信割り込み要求(RXOUTI)が起動されます。バンクはパケットの先頭からのバイトで満たされます。

バンクがデータアクセスの準備ができていない場合(送信可(TXINI)=1または読み書き可(RWAL)=1)でだけCPUが書くのが当然なので、CPU側でIN段階中にオーバーフロー異常になることは不可能です。

22.17. 装置割り込み

図22-11.は全割り込み元を示します。

割り込みは割り込み処理(即ちこれらの発生は通常処理の一部です)と例外(異常)の2種類です。

通常処理中割り込みは以下の時に生成されます。

- VBUS接続検出(挿抜) (VBUSTI)
- 上方向再開(Upstream resume) (UPRSMI)
- 再開終了 (EORSMI)
- 起動復帰 (WAKEUPI)
- リセット終了(速度初期化) (EORSTI)
- フレーム開始 (SOFI, フレーム番号CRC異常(FNCERR)フラグ=0の場合)
- 不活性の3ms後の休止(Suspend)検出 (SUSPI)

例外割り込みは以下の時に生成されます。

- フレーム開始(SOF)のフレーム番号でのCRC異常 (SOFI, FNCERR=1の場合)

図22-11. USB装置制御器割り込み元

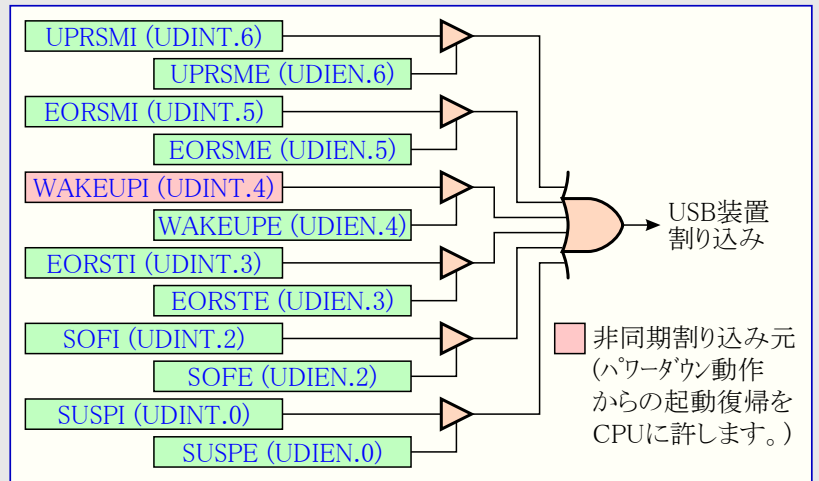
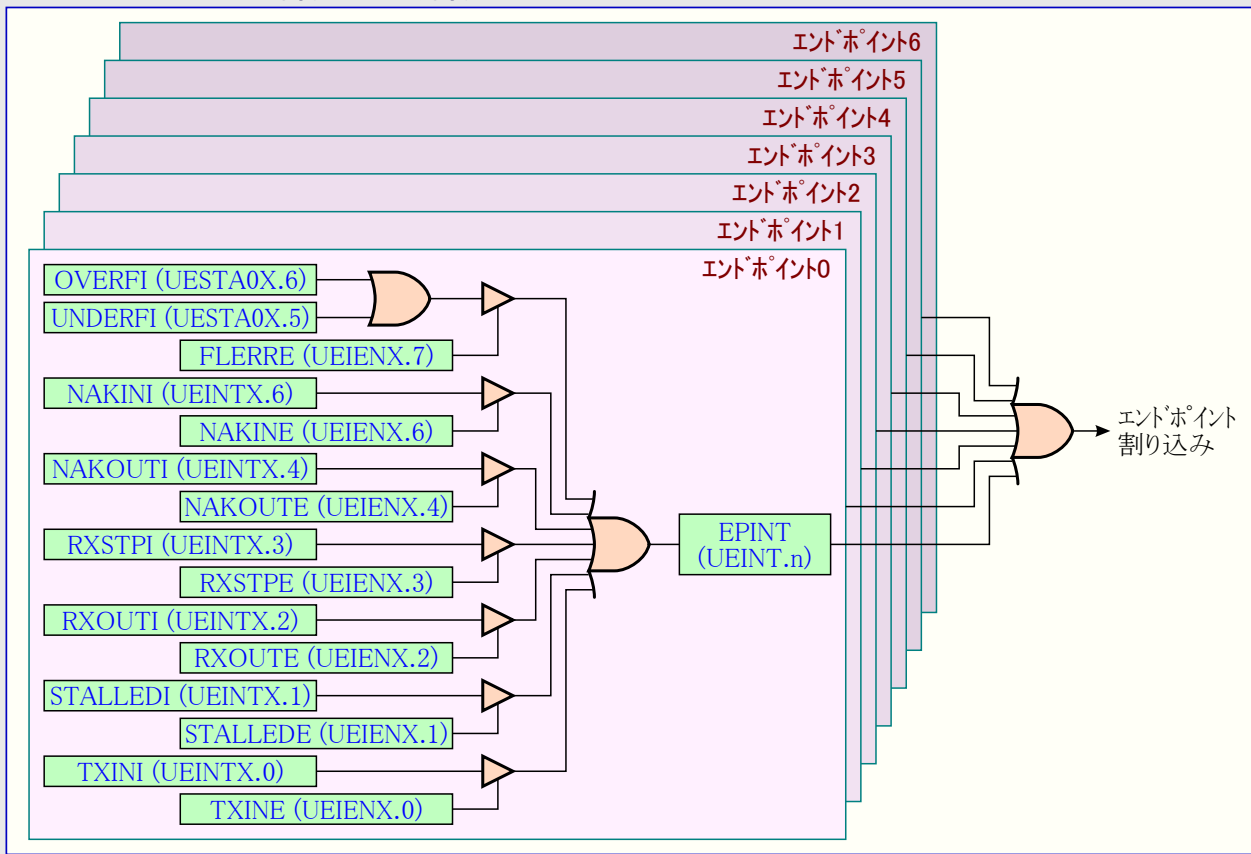


図22-12. USBエンドポイント割り込みベクタ割り込み元



通常処理中割り込みは以下の時に生成されます。

- INデータ受け入れ可 (EPINT_n, TXINI=1)
- OUTデータ受信 (EPINT_n, RXOUTI=1)
- 設定(SETUP)受信 (EPINT_n, RXSTPI=1)

例外割り込みは以下の時に生成されます。

- 不能パケット (EPINT_n, STALLEDI=1)
- 等時(Isochronous)動作のOUTでのCRC異常 (EPINT_n, OVERFI=1)
- 等時動作でのオーバーフロー (EPINT_n, OVERFI=1)
- 等時動作でのアンダーフロー (EPINT_n, UNDERFI=1)
- INでの否定応答(NAK) (EPINT_n, NAKINI=1)
- OUTでの否定応答(NAK) (EPINT_n, NAKOUTI=1)

22.18. USB装置一般用レジスタ

22.18.1. UDCON – USB装置制御レジスタ (USB Device Control Register)

ビット (\$E0)	7	6	5	4	3	2	1	0	
	–	–	–	–	RSTCPU	LSM	RMWKUP	DETACH	UDCON
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	1	

● ビット7～4 – Res : 予約 (Reserved)

これらのビットからの読み込み値は常に0です。これらのビットを設定(1)してはいけません。

● ビット3 – RSTCPU : CPUリセット (USB Reset CPU Bit)

(USB制御器の禁止と接続状態を除いて)USBリセット終了検出でCPUをリセットするには、ファームウェアによってこのビットを設定(1)してください。このビットはUSB制御器が禁止される時にリセット(0)されますが、[リセット終了\(End Of Reset\)](#)によって生成されたCPUリセットによって影響を及ぼされません(許可(1)に留まります)。

● ビット2 – LSM : 低速/全速装置動作選択 (USB Device Low Speed Mode Selection)

このビットはUSBの低速(Low-speed)または全速(Full-speed)動作選択を許します。

全速動作(取り付け(DETACHビット=0)とでD+内部プルアップ活性(有効))を選ぶには解除(0)してください。低速動作(取り付け(DETACHビット=0)とでD-内部プルアップ活性(有効))を選ぶには設定(1)してください。(訳注:原書本項の無効なホスト動作関係記述削除)

● ビット1 – RMWKUP : 遠隔起動復帰 (Remote Wake-up Bit)

遠隔起動復帰に関してホストへ上方向再開(Upstream-resume)を送信するには設定(1)してください(休止割り込み要求(SUSPI)フラグが設定(1)でなければなりません)。

信号通信完了時、ハードウェアによって解除(0)されます。ソフトウェアでの解除(0)は無効です。より多くの詳細については178頁の「[遠隔起動復帰](#)」をご覧ください。

● ビット0 – DETACH : 分離 (Detach Bit)

装置を物理的に分離(D+またはD-上の内部プルアップ切断)するには設定(1)してください。

装置を再接続するには解除(0)してください。より多くの詳細については178頁の「[分離\(切り離し\)](#)」をご覧ください。

22.18.2. UDINT – USB装置割り込み要求レジスタ (USB Device Interrupt Register)

ビット (\$E1)	7	6	5	4	3	2	1	0	
	–	UPRSMI	EORSMI	WAKEUPI	EORSTI	SOFI	–	SUSPI	UDINT
Read/Write	R	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 – Res : 予約 (Reserved)

このビットからの読み込み値は常に0です。このビットを設定(1)してはいけません。

● ビット6 – UPRSMI : 上方向再開割り込み要求フラグ (Upstream Resume Interrupt Flag)

USB制御器が上方向再開(Upstream Resume)と呼ばれる再開信号を送る時にハードウェアによって設定(1)されます。これは[上方向再開割り込み許可\(UPRSME\)](#)が設定(1)されているなら、USB割り込みを起動します。

ソフトウェアで解除(0)されるべきです(その前にUSBクロック入力が許可されていなければなりません)。ソフトウェアによる設定(1)は無効です。

● ビット5 – EORSMI : 再開終了割り込み要求フラグ (End Of Resume Interrupt Flag)

USB制御器がホストによって開始された良好な再開終了(End Of Resume)を検知すると、ハードウェアによって設定(1)されます。これは[再開終了割り込み許可\(EORSME\)](#)が設定(1)されているなら、USB割り込みを起動します。

ソフトウェアによって解除(0)されるべきでしょう。ソフトウェアによる設定(1)は無効です。

● ビット4 – WAKEUPI : CPU起動復帰割り込み要求フラグ (Wake-up CPU Interrupt Flag)

USB制御器が信号線から(上方向再開(Upstream resume)でない)非アイドル信号濾出によって再活性化されると、ハードウェアによって設定(1)されます。これは[CPU起動復帰割り込み許可\(WAKEUPE\)](#)が設定(1)されているなら、USB割り込みを起動します。

ソフトウェアで解除(0)されるべきです(その前にUSBクロック入力が許可されていなければなりません)。ソフトウェアによる設定(1)は無効です。

より多くの詳細については178頁の「[休止、起動復帰と再開](#)」をご覧ください。

● ビット3 – EORSTI : リセット終了割り込み要求フラグ (End Of Reset Interrupt Flag)

リセット終了(End Of Reset)がUSB制御器によって検出されると、ハードウェアによって設定(1)されます。これは[リセット終了割り込み許可\(EORSTE\)](#)が設定(1)されているなら、USB割り込みを起動します。

ソフトウェアによって解除(0)されるべきでしょう。ソフトウェアによる設定(1)は無効です。

● **ビット2 – SOFI : フレーム開始割り込み要求フラグ** (Start Of Frame Interrupt Flag)

フレーム開始(SOF)PIDが(毎1msで)検知されると、ハードウェアによって設定(1)されます。これは**フレーム開始割り込み許可(SOFE)**が設定(1)されているなら、USB割り込みを起動します。

● **ビット1 – Res : 予約** (Reserved)

このビットからの読み込み値は常に0です。このビットを設定(1)してはいけません。

● **ビット0 – SUSPI : 休止割り込み要求フラグ** (Suspend Interrupt Flag)

USB休止(Suspend) (3フレーム期間のバスアイドル、3msのJ状態)が検出されると、ハードウェアによって設定(1)されます。これは**休止割り込み許可(SUSPE)**が設定(1)されているなら、USB割り込みを起動します。

ソフトウェアによって解除(0)されるべきでしょう。ソフトウェアによる設定(1)は無効です。

より多くの詳細については178頁の「**休止、起動復帰と再開**」をご覧ください。

これらの割り込み要求フラグは例え対応する割り込み許可ビットが設定(1)されていなくても設定(1)されます。

22.18.3. UDIEN – USB装置割り込み許可レジスタ (USB Device Interrupt Enable Register)

ビット (\$E2)	7	6	5	4	3	2	1	0	
	–	UPRSME	EORSME	WAKEUPE	EORSTE	SOFE	–	SUSPE	UDIEN
Read/Write	R	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 – Res : 予約** (Reserved)

このビットからの読み込み値は常に0です。このビットを設定(1)してはいけません。

● **ビット6 – UPRSME : 上方向再開割り込み許可** (Upstream Resume Enable Bit)

上方向再開(Upstream Resume)割り込み(UPRSMI)を許可するには設定(1)してください。UPRSMIを禁止するには解除(0)してください。

● **ビット5 – EORSME : 再開終了割り込み許可** (End Of Resume Enable Bit)

再開終了(End Of Resume)割り込み(EORSMI)を許可するには設定(1)してください。EORSMIを禁止するには解除(0)してください。

● **ビット4 – WAKEUPE : CPU起動復帰割り込み許可** (Wake-up CPU Enable Bit)

CPU起動復帰割り込み(WAKEUPI)を許可するには設定(1)してください。WAKEUPIを禁止するには解除(0)してください。

● **ビット3 – EORSTE : リセット終了割り込み許可** (End Of Reset Enable Bit)

リセット終了(End Of Reset)割り込み(EORSTI)を許可するには設定(1)してください。EORSTIを禁止するには解除(0)してください。

● **ビット2 – SOFE : フレーム開始割り込み許可** (Start Of Frame Enable Bit)

フレーム開始割り込み(SOFI)を許可するには設定(1)してください。SOFIを禁止するには解除(0)してください。

● **ビット1 – Res : 予約** (Reserved)

このビットからの読み込み値は常に0です。このビットを設定(1)してはいけません。

● **ビット0 – SUSPE : 休止割り込み許可** (Suspend Interrupt Enable Bit)

USB休止(Suspend)割り込み(SUSPI)を許可するには設定(1)してください。SUSPIを禁止するには解除(0)してください。

22.18.4. UDADDR – USB装置アドレスレジスタ (USB Device Address Register)

ビット (\$E3)	7	6	5	4	3	2	1	0	
	ADDEN	UADD6	UADD5	UADD4	UADD3	UADD2	UADD1	UADD0	UDADDR
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 – ADDEN : アドレス許可** (Address Enable Bit)

UADD6~0を活性(有効)にするには設定(1)してください。

このビットはハードウェアによって解除(0)されます。ソフトウェアによる解除(0)は無効です。

より多くの詳細については177頁の「**アドレス設定**」をご覧ください。

● **ビット6~0 – UADD6~0 : USB装置アドレス** (USB Address Bits)

装置アドレスを指定するためにソフトウェアで設定してください。

22.18.5. UDFNUMH, UDFNUML (UDFNUM) – USB装置フレーム番号レジスタ (USB Device Frame Number Register)

ビット (\$E5)	15	14	13	12	11	10	9	8	
	–	–	–	–	–	FNUM10	FNUM9	FNUM8	UDFNUMH
Read/Write	R	R	R	R	R	R	R	R	
初期値	0	0	0	0	0	0	0	0	
ビット (\$E4)	7	6	5	4	3	2	1	0	
	FNUM7	FNUM6	FNUM5	FNUM4	FNUM3	FNUM2	FNUM1	FNUM0	UDFNURL
Read/Write	R	R	R	R	R	R	R	R	
初期値	0	0	0	0	0	0	0	0	

● ビット15～11 – Res : 予約 (Reserved)

これらのビットからの読み込み値は常に0です。これらのビットを設定(1)してはいけません。

● ビット10～0 – FNUM10～0 : フレーム番号 (Frame Number Flags)

ハードウェアによって設定されます。これらは11ビットのフレーム番号情報です。これらは最後に受信したフレーム開始(SOF)パケットで供給されます。FNUM10～0は不正なSOFが受信された場合も更新されます。

(訳注) UDFNUMHとUDFNURLは原書での個別表記を統一表記に纏めました。

22.18.6. UDMFN – USB装置フレーム番号状態レジスタ (USB Device Maintained Frame Number Register)

ビット (\$E6)	7	6	5	4	3	2	1	0	
	–	–	–	FNCERR	–	–	–	–	UDMFN
Read/Write	R	R	R	R	R	R	R	R	
初期値	0	0	0	0	0	0	0	0	

● ビット7～5 – Res : 予約 (Reserved)

これらのビットからの読み込み値は常に0です。これらのビットを設定(1)してはいけません。

● ビット4 – FNCERR : フレーム番号CRC異常フラグ (Frame Number CRC Error Flag)

フレーム開始(SOF)パケットで不正なフレーム番号が受信されると、ハードウェアによって設定(1)されます。

このビットとフレーム開始割り込み要求(SOFI)は同時に更新されます。

● ビット3～0 – Res : 予約 (Reserved)

これらのビットからの読み込み値は常に0です。これらのビットを設定(1)してはいけません。

22.19. USB装置エンドポイント用レジスタ

22.19.1. UENUM – エンドポイント番号レジスタ (USB Endpoint Number Register)

ビット (\$E9)	7	6	5	4	3	2	1	0	
	–	–	–	–	–	EPNUM2	EPNUM1	EPNUM0	UENUM
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

● ビット7～3 – Res : 予約 (Reserved)

これらのビットからの読み込み値は常に0です。これらのビットを設定(1)してはいけません。

● ビット2～0 – EPNUM2～0 : エンドポイント番号 (Endpoint Number Bits)

CPUによってアクセスされるべきエンドポイントの番号を選ぶためにソフトウェアで設定してください。Xで終わるUSB装置レジスタのXがこの番号に対応します(訳注: 共通性から本行追加)。より多くの詳細については176頁の「エンドポイント選択」項をご覧ください。

EPNUM2～0=111は許されません。

22.19.2. UERST – エンドポイントリセットレジスタ (USB Endpoint Reset Register)

ビット (\$EA)	7	6	5	4	3	2	1	0	
	–	EPRST6	EPRST5	EPRST4	EPRST3	EPRST2	EPRST1	EPRST0	UERST
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	0	0	0	

● ビット7 – Res : 予約 (Reserved)

このビットからの読み込み値は常に0です。このビットを設定(1)してはいけません。

● ビット6~0 – EPRST6~EPRST0 : エンドポイントFIFOリセット (Endpoint FIFO Reset Bits)

ハードウェアリセットまたはUSBバスリセットが受信された時に選んだエンドポイントFIFOをリセットするために、他のどの操作にも先立って設定(1)してください。より多くの情報については176頁の「[エンドポイントリセット](#)」項をご覧ください。

リセット動作を完了するためにソフトウェアで解除(0)された後にエンドポイントの使用を始めてください。

22.19.3. UECONX – エンドポイントX制御レジスタ (USB Endpoint X Control Register)

ビット (\$EB)	7	6	5	4	3	2	1	0	
	–	–	STALLRQ	STALLRQC	RSTDT	–	–	EPEN	UECONX
Read/Write	R	R	W	W	W	R	R	R/W	
初期値	0	0	0	0	0	0	0	0	

● ビット7,6 – Res : 予約 (Reserved)

これらのビットからの読み込み値は常に0です。これらのビットを設定(1)してはいけません。

● ビット5 – STALLRQ : 不能応答ハンドシェイク要求 (STALL Request Handshake Bit)

次のハンドシェイクに対してホストへ不能応答(STALL)を要求するには設定(1)してください。

次の(新規)SETUPが受信されると、ハードウェアによって解除(0)されます。ソフトウェアによる解除(0)は無効です。

より多くの詳細については179頁の「[不能応答\(STALL\)ハンドシェイクと再試行機構](#)」項をご覧ください。

● ビット4 – STALLRQC : 不能応答ハンドシェイク解除 (STALL Request Clear Handshake Bit)

不能応答(STALL)ハンドシェイク機構を禁止するには設定(1)してください。

設定(1)後、ハードウェアによって直ちに解除(0)されます。ソフトウェアによる解除(0)は無効です。

より多くの詳細については179頁の「[不能応答\(STALL\)ハンドシェイクと再試行機構](#)」項をご覧ください。

● ビット3 – RSTDT : 交互データ番号リセット (Reset Data Toggle Bit)

交互データ番号処理を自動的に解除するには設定(1)してください。

OUTエンドポイントに対して次に受信したパケットは交互データ番号=0でしょう。

INエンドポイントに対して次に送るべきパケットは交互データ番号=0でしょう。

このビットはハードウェアによって即時に解除(0)されます。ファームウェアはこのビットが解除(0)されるのを待つ必要はありません。ソフトウェアによる解除(0)は無効です。

● ビット2,1 – Res : 予約 (Reserved)

これらのビットからの読み込み値は常に0です。これらのビットを設定(1)してはいけません。

● ビット0 – EPEN : エンドポイント許可 (Endpoint Enable Bit)

装置構成に従ってエンドポイントを許可するには設定(1)してください。エンドポイント0は装置構成設定に関係するので、ハードウェアまたはUSBリセット後、常に許可されるべきです。

エンドポイントを禁止するには、このビットを解除(0)してください。より多くの詳細については177頁の「[エンドポイントの活性化\(有効設定\)](#)」項をご覧ください。

22.19.4. UECFG0X – エンドポイントX構成0レジスタ (USB Endpoint X Configuration 0 Register)

ビット (\$EC)	7	6	5	4	3	2	1	0	
	EPTYPE1	EPTYPE0	–	–	–	–	–	EPDIR	UECFG0X
Read/Write	R/W	R/W	R	R	R	R	R	R/W	
初期値	0	0	0	0	0	0	0	0	

● ビット7,6 – EPTYPE1,0 : エンドポイント形式 (Endpoint Type Bits)

エンドポイント構成設定に従ってこのビットを設定してください。

00=制御(Control)、01=等時(Isochronous)、10=大量(Bulk)、11=割り込み(Interrupt)

● ビット5~1 – Res : 予約 (Reserved)

これらのビットからの読み込み値は常に0です。これらのビットを設定(1)してはいけません。

● ビット0 – EPDIR : エンドポイント方向 (Endpoint Direction Bit)

大量(Bulk)、割り込み(Interrupt)、等時(Isochronous)エンドポイントに対してIN方向を設定するには設定(1)してください。

大量、割り込み、等時、制御(Control)エンドポイントに対してOUT方向を設定するには解除(0)してください。

22.19.5. UECFG1X – エンドポイントX構成1レジスタ (USB Endpoint X Configuration 1 Register)

ビット (\$ED)	7	6	5	4	3	2	1	0	
	–	EPSIZE2	EPSIZE1	EPSIZE0	EPBK1	EPBK0	ALLOC	–	UECFG1X
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)

このビットからの読み込み値は常に0です。このビットを設定(1)してはいけません。

● ビット6~4 – EPSIZE2~0 : エンドポイント容量 (Endpoint Size Bits)

エンドポイント容量に従ってこのビットを設定してください。

000=8バイト、001=16バイト、010=32バイト、011=64バイト、100=128バイト、101=256バイト、110=512バイト、111=予約(使用禁止)

● ビット3,2 – EPBK1,0 : エンドポイント バンク形式 (Endpoint Bank Bits)

エンドポイント容量に従ってこの領域を設定してください。

00=1バンク、01=2バンク、1x=予約(本設定使用禁止)

● ビット1 – ALLOC : エンドポイント メモリ割り当て (Endpoint Allocation Bit)

エンドポイント メモリを割り当てるにはこのビットを設定(1)してください。

エンドポイント メモリを開放するには解除(0)してください。

より多くの詳細については177頁の「[エンドポイントの活性化\(有効設定\)](#)」項をご覧ください。

● ビット0 – Res : 予約 (Reserved)

このビットからの読み込み値は常に0です。このビットを設定(1)してはいけません。

22.19.6. UESTA0X – エンドポイントX状態0レジスタ (USB Endpoint X Status 0 Register)

ビット (\$EE)	7	6	5	4	3	2	1	0	
	CFGOK	OVERFI	UNDERFI	–	DTSEQ1	DTSEQ0	NBUSYBK1	NBUSYBK0	UESTA0X
Read/Write	R	R/W	R/W	R	R	R	R	R	
初期値	0	0	0	0	0	0	0	0	

● ビット7 – CFGOK : 構成状態フラグ (Configuration Status Flag)

エンドポイントX容量(EPSIZE)とバンク形式(EPBK)が許された最大バンク数と最大FIFOと比較して正しい時にハードウェアによって設定(1)されます。このビットは割り当て(ALLOC)ビットが設定(1)される時に更新されます。

このビットが解除(0)されている場合、使用者は正しいEPSIZEとEPBK値でエンドポイント構成1(UECFG1X)レジスタを再設定すべきです。

● ビット6 – OVERFI : オーバーフロー異常割り込み要求フラグ (Overflow Error Interrupt Flag)

等時(Isochronous)エンドポイントでオーバーフロー異常が起これば、ハードウェアによって設定(1)されます。(許可なら)割り込み(EPINTn)が起動されます。

より多くの詳細については183頁の「[等時\(Isochronous\)動作](#)」項をご覧ください。

このビットはソフトウェアによって解除(0)されるべきでしょう。ソフトウェアによる設定(1)は無効です。

● ビット5 – UNDERFI : アンダーフロー異常割り込み要求フラグ (Underflow Error Interrupt Flag)

等時(Isochronous)エンドポイントでアンダーフロー異常が起これば、ハードウェアによって設定(1)されます。(許可なら)割り込み(EPINTn)が起動されます。

より多くの詳細については182頁の「[等時\(Isochronous\)動作](#)」項をご覧ください。

このビットはソフトウェアによって解除(0)されるべきでしょう。ソフトウェアによる設定(1)は無効です。

● ビット4 – Res : 予約 (Reserved)

このビットからの読み込み値は常に0です。このビットを設定(1)してはいけません。

● ビット3,2 – DTSEQ1,0 : 交互データ番号フラグ (Data Toggle Sequencing Flag)

現在のバンクのPIDデータを示すためにハードウェアによって設定されます。

00=Data0、01=Data1、1x=予約

OUT転送に於ける本値は現在のバンクで最後に受信した交互データ番号を示します。

IN転送に於ける本値は送るべき次パケットに対して使われるべき交互データ番号を示します。これは現在のバンクと無関係です。

● **ビット1,0 – NBUSYBK1,0 : バンク内容状況フラグ** (Busy Bank Flag)

内容が設定されているバンク数を示すためにハードウェアによって設定されます。

INエンドポイントに対しては使用者によって満たされ、IN転送の準備ができていないバンク数を示します。

OUTエンドポイントに対してはホストからのOUT転送単位(Transaction)によって満たされたバンク数を示します。

00=全バンク空き、01=1バンク設定済み、10=2バンク設定済み、11=予約

22.19.7. UESTA1X – エンドポイントX状態レジスタ (USB Endpoint X Status 1 Register)

ビット (\$EF)	7	6	5	4	3	2	1	0	
	–	–	–	–	–	CTRLDIR	CURRBK1	CURRBK0	UESTA1X
Read/Write	R	R	R	R	R	R	R	R	
初期値	0	0	0	0	0	0	0	0	

● **ビット7~3 – Res : 予約** (Reserved)

これらのビットからの読み込み値は常に0です。これらのビットを設定(1)してはいけません。

● **ビット2 – CTRLDIR : 制御転送方向** (Control Direction Flag)

設定(SETUP)パケット後にハードウェアによって設定され、後続パケットの方向を与えます。

0=OUTエンドポイント、1=INエンドポイント。ソフトウェアによる設定(1)や解除(0)はできません。

● **ビット1,0 – CURRBK1,0 : (除く制御エンドポイント)現バンク番号** (Current Bnak (endpoints except Control endpoint) Flag)

現在のバンク番号を示すためにハードウェアによって設定されます。

00=バンク0、01=バンク1、1x=予約。ソフトウェアによる設定(1)や解除(0)はできません。

22.19.8. UEINTX – エンドポイントX割り込み要求レジスタ (USB Endpoint X Interrupt Register)

ビット (\$E8)	7	6	5	4	3	2	1	0	
	FIFOCON	NAKINI	RWAL	NAKOUTI	RXSTPI	RXOUTI	STALLEDI	TXINI	UEINTX
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 – FIFOCON : FIFO制御** (FIFO Control Bit)

OUTとSETUPエンドポイントに対して：

新規OUTメッセージが現在のバンク内に格納されると、ハードウェアによってOUTデータ受信割り込み要求(RXOUTI)またはSETUP受信割り込み要求(RXSTPI)と同時に設定(1)されます。

現在のバンクを開放するために解除(0)して次のバンクに切り替えてください。ソフトウェアによる設定(1)は無効です。

INエンドポイントに対して：

現在のバンクが空の時にハードウェアによって送信可割り込み要求(TXINI)と同時に設定(1)されます。

FIFOデータを送信してバンクを切り替えるために解除(0)してください。ソフトウェアによる設定(1)は無効です。

● **ビット6 – NAKINI : IN受信否定応答割り込み要求フラグ** (NAK IN Received Interrupt Flag)

ホストからのIN要求の応答で否定応答(NAK)ハンドシェイクが送られた時にハードウェアによって設定(1)されます。これはIN受信否定応答割り込み許可(NAKINI)が設定(1)されているなら、USB割り込みを起動します。

このフラグはソフトウェアによって解除(0)されるべきでしょう。ソフトウェアによる設定(1)は無効です。

● **ビット5 – RWAL : 読み書き可フラグ** (Read/Write Allowed Flag)

以下を示すためにハードウェアによって設定(1)されます。

- ・ INエンドポイントについては、現バンクが一杯でない、換言するとファームウェアがFIFOにデータを書ける
- ・ OUTエンドポイントについては、現バンクが空でない、換言するとファームウェアがFIFOからデータを読める

このビットは不能応答(STALL)ハンドシェイク要求(STALLRQ)が設定(1)、または異常の場合に決して設定(1)されません。

その他の場合、ハードウェアによって解除(0)されます。

このビットは制御(Control)エンドポイントに使われるべきではありません。

● **ビット4 – NAKOUTI : OUT受信否定応答割り込み要求フラグ** (NAK OUT Received Interrupt Flag)

ホストからのOUT/PING要求の応答で否定応答(NAK)ハンドシェイクが送られた時にハードウェアによって設定(1)されます。これはOUT受信否定応答割り込み許可(NAKOUTI)が設定(1)されているなら、USB割り込みを起動します。

このフラグはソフトウェアによって解除(0)されるべきでしょう。ソフトウェアによる設定(1)は無効です。

● **ビット3 – RXSTPI : SETUP受信割り込み要求フラグ** (Received SETUP Interrupt Flag)

現バンクが新規の有効設定(SETUP)パケットを含むことを示すためにハードウェアによって設定(1)されます。(許可なら、)割り込み(EPINTn)が起動されます。

割り込みでのやり取りを行うためにソフトウェアで解除(0)してください。ソフトウェアによる設定(1)は無効です。

エンドポイントがINエンドポイントの場合、このビットは非活性(解除(0))です。

● **ビット2 – RXOUTI/KILLBK**

RXOUTI : OUTデータ受信割り込み要求フラグ (Received OUT Data Interrupt Flag)

現バンクが新規パケットを含むことを示すためにハードウェアによって設定(1)されます。(許可なら、)割り込み(EPINTn)が起動されます。

割り込みでのやり取りを行うためにソフトウェアで解除(0)してください。ソフトウェアによる設定(1)は無効です。

KILLBK : 最終書き込みバンク無効化 (Kill Bank IN Bit)

最後に書かれたバンクを消去(無効化)するにはこのビットを設定(1)してください。

そのバンクが消去(無効化)されると、ハードウェアによって解除(0)されます。ソフトウェアによる解除(0)は無効です。

失敗中止(Abort)でのより多くの詳細については182頁の「失敗中止(Abort)」をご覧ください。

● **ビット1 – STALLEDI : 不能応答送信完了/異常検出割り込み要求フラグ** (STALLEDI Interrupt Flag)

不能応答(STALL)ハンドシェイクが送信された、または等時(Isochronous)OUTエンドポイントでCRC異常が検出されたことを示すためにハードウェアによって設定(1)されます。

このフラグはソフトウェアによって解除(0)されるべきでしょう。ソフトウェアによる設定(1)は無効です。

● **ビット0 – TXINI : 送信可割り込み要求フラグ** (Transmitter Ready Flag)

現バンクが空で、満たされ得ることを示すためにハードウェアによって設定(1)されます。(許可なら、)割り込み(EPINTn)が起動されます。

割り込みでのやり取りを行うためにソフトウェアで解除(0)してください。ソフトウェアによる設定(1)は無効です。

エンドポイントがOUTエンドポイントの場合、このビットは非活性(解除(0))です。

22.19.9. UEIENX – エンドポイントX割り込み許可レジスタ (USB Endpoint X Interrupt Enable Register)

ビット (\$F0)	7	6	5	4	3	2	1	0	
	FLERRE	NAKINE	–	NAKOUTE	RXSTPE	RXOUTE	STALLEDE	TXINE	UEIENX
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 – FLERRE : 流れ異常割り込み許可** (Flow Error Interrupt Enable Bit)

オーバーフロー割り込み要求(OVERFI)またはアンダーフロー割り込み要求(UNDERFI)が設定(1)された時にエンドポイント割り込み(EPINTn)を許可するには設定(1)してください。

OVERFIまたはUNDERFIが設定(1)された時にエンドポイント割り込み(EPINTn)を禁止するには解除(0)してください。

● **ビット6 – NAKINE : IN受信否定応答割り込み許可** (NAK IN Interrupt Enable Bit)

IN受信否定応答割り込み要求(NAKINI)が設定(1)された時にエンドポイント割り込み(EPINTn)を許可するには設定(1)してください。

NAKINIが設定(1)された時にエンドポイント割り込み(EPINTn)を禁止するには解除(0)してください。

● **ビット5 – Res : 予約** (Reserved)

このビットからの読み込み値は常に0です。このビットを設定(1)してはいけません。

● **ビット4 – NAKOUTE : OUT受信否定応答割り込み許可** (NAK OUT Interrupt Enable Bit)

OUT受信否定応答割り込み要求(NAKOUTI)が設定(1)された時にエンドポイント割り込み(EPINTn)を許可するには設定(1)してください。

NAKOUTIが設定(1)された時にエンドポイント割り込み(EPINTn)を禁止するには解除(0)してください。

● **ビット3 – RXSTPE : SETUP受信割り込み許可** (Received SETUP Interrupt Enable Bit)

SETUP受信割り込み要求(RXSTPI)が設定(1)された時にエンドポイント割り込み(EPINTn)を許可するには設定(1)してください。

RXSTPIが設定(1)された時にエンドポイント割り込み(EPINTn)を禁止するには解除(0)してください。

● **ビット2 – RXOUTE : OUTデータ受信割り込み許可** (Received OUT Data Interrupt Enable Bit)

OUTデータ受信割り込み要求(RXOUTI)が設定(1)された時にエンドポイント割り込み(EPINTn)を許可するには設定(1)してください。

RXOUTIが設定(1)された時にエンドポイント割り込み(EPINTn)を禁止するには解除(0)してください。

●ビット1 – STALLEDE : 不能応答/異常検出割り込み許可 (Stalled Interrupt Enable Bit)

不能応答/異常検出割り込み要求(STALLEDI)が設定(1)された時に、エンドポイント割り込み(EPINTn)を許可するには設定(1)してください。

STALLEDIが設定(1)された時にエンドポイント割り込み(EPINTn)を禁止するには解除(0)してください。

●ビット0 – TXINE : 送信可割り込み許可 (Transmitter Ready Interrupt Enable Bit)

送信可割り込み要求(TXINI)が設定(1)された時にエンドポイント割り込み(EPINTn)を許可するには設定(1)してください。

TXINIが設定(1)された時にエンドポイント割り込み(EPINTn)を禁止するには解除(0)してください。

22.19.10. UEDATX – エンドポイントX データレジスタ (USB Endpoint X Data Register)

ビット (\$F1)	7	6	5	4	3	2	1	0	
	DAT7	DAT6	DAT5	DAT4	DAT3	DAT2	DAT1	DAT0	UEDATX
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 – DAT7~0 : データ (Data Bits)

エンドポイント番号(EPNUM)によって選んだエンドポイントFIFOとのバイト読み書きをソフトウェアによって行ってください。

22.19.11. UEBCHX, UEBCLX (UEBCX) – エンドポイントX バイト数レジスタ (USB Endpoint X Byte Count Register)

ビット (\$F3)	15	14	13	12	11	10	9	8	
	–	–	–	–	–	BYCT10	BYCT9	BYCT8	UEBCHX
Read/Write	R	R	R	R	R	R	R	R	
初期値	0	0	0	0	0	0	0	0	
ビット (\$F2)	7	6	5	4	3	2	1	0	
	BYCT7	BYCT6	BYCT5	BYCT4	BYCT3	BYCT2	BYCT1	BYCT0	UEBCLX
Read/Write	R	R	R	R	R	R	R	R	
初期値	0	0	0	0	0	0	0	0	

●ビット15~11 – Res : 予約 (Reserved)

これらのビットからの読み込み値は常に0です。これらのビットを設定(1)してはいけません。

●ビット10~0 – BYCT10~0 : バイト数 (Byte count Bits)

ハードウェアによって設定されます。これらはFIFOエンドポイントのバイト数です。

- ・(INエンドポイントについては)、エンドポイント内への毎書き込み後に増加(+1)され、毎バイト送信後に減少(-1)されます。
- ・(OUTエンドポイントについては)、ホストによる毎バイト送信後に増加(+1)され、ソフトウェアによる毎バイト読み込み後に減少(-1)されます。

(訳注) UEBCHXとUEBCLXは原書での個別表記を統一表記に纏めました。

22.19.12. UEINT – エンドポイント割り込み要求レジスタ (USB Endpoint Interrupt Register)

ビット (\$F4)	7	6	5	4	3	2	1	0	
	–	EPINT6	EPINT5	EPINT4	EPINT3	EPINT2	EPINT1	EPINT0	UEINT
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 – Res : 予約 (Reserved)

このビットからの読み込み値は常に0です。このビットを設定(1)してはいけません。

●ビット6~0 – EPINT6~EPINT0 : エンドポイント割り込み要求フラグ (Endpoint Interrupts Bits)

エンドポイントX割り込み要求(UEINTX)レジスタに対応するエンドポイント割り込み許可ビットが設定(1)され、UEINTXによって割り込みが起動されると、ハードウェアによって設定(1)されます。

その割り込み元が取り扱われる(処理される)と、ハードウェアによって解除(0)されます。

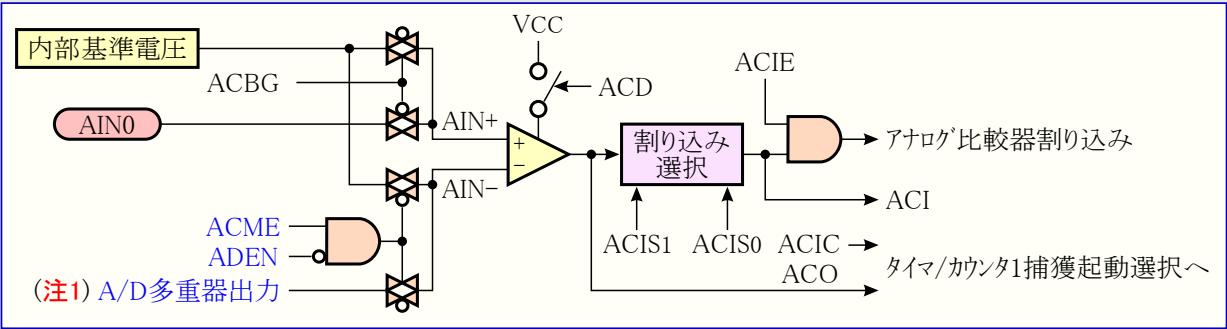
23. アナログ比較器

アナログ比較器は非反転入力AIN+と反転入力AIN-の入力値を比較します。非反転入力AIN+の電圧が反転入力AIN-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23-1.で示されます。

AIN+はAIN0(PE6)ピンまたは内部(ハントギャップ)基準電圧のどちらかに接続することができます。AIN-はA/D多重器だけに接続することができます。

ADC入力(A/D多重器出力)の使用を可能とするには、28頁の「PRR0 – 電力削減レジスタ」のPRADCビットが論理0を書かれることで(電力削減を)禁止されなければなりません。

図23-1. アナログ比較器部構成図



注1: A/D多重器出力については表23-1.をご覧ください。
注: アナログ比較器ピン配置については2頁の「ピン配置」と50頁の表11-11.を参照してください。

23.1. 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ADC13~0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。ADCSR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、下表で示されるように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、内部(ハントギャップ)基準電圧がアナログ比較器への反転入力に印加されます。

表23-1. 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	内部(ハントギャップ)基準電圧	
	1	x x x		
		0 0 0		
		0 0 1	ADC0	
		0 1 0	(利用不可)	
		0 1 1	(利用不可)	
1	0	1 0 0	ADC4	
		1 0 1	ADC5	
		1 1 0	ADC6	
		1 1 1	ADC7	

23.2. アナログ比較器用レジスタ

23.2.1. ADCSRB – A/D変換制御/状態レジスタB (ADC Control and Status Register B)

ビット (\$7B)	7	6	5	4	3	2	1	0	
	ADHSM	ACME	MUX5	-	ADTS3	ADTS2	ADTS1	ADTS0	ADCSR B
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

● ビット6 – 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に接続されます。このビットが論理0を書かれると、内部(ハントギャップ)基準電圧がアナログ比較器の反転入力に接続されます(32頁の「内部基準電圧」をご覧ください)。このビットの詳細な記述については「アナログ比較器入力選択」をご覧ください。

23.2.2. ACSR – アナログ比較器 制御/状態レジスタ (Analog Comparator Control and Status Register)

ビット	7	6	5	4	3	2	1	0	
\$30 (\$50)	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1.1V)がアナログ比較器への非反転入力に置き換わります。本ビットが解除(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。32ページの「[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1割り込み許可レジスタ\(TIMSK1\)の捕獲割り込み許可\(ICIE1\)ビット](#)が設定(1)されなければなりません。

● ビット1,0 – ACIS1,0 : アナログ比較器割り込み条件 (Analog Comparator Interrupt Mode Select)

これらのビットは比較器のどの事象がアナログ比較器割り込みを起動するのかを決めます。各種設定は右表で示されます。

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23-2. 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

23.2.3. DIDR1 – デジタル入力禁止レジスタ1 (Digital Input Disable Register 1)

ビット	7	6	5	4	3	2	1	0	
(\$7F)	–	–	–	–	–	–	–	AIN0D	DIDR1
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) (訳注:共通性から本項追加)

これらのビットは使われず、常に0として読みます。

● ビット0 – AIN0D : AIN0 デジタル入力禁止 (AIN0 Digital Input Disable)

このビットが論理1を書かれると、AIN0ピンのデジタル入力緩衝部が禁止されます。このビットが設定(1)されると、対応するポート入力レジスタのビット(PINx)は常に0として読みます。AIN0ピンにアナログ信号が印加され、そのピンからのデジタル入力が必要とされない時にデジタル入力緩衝部での消費電力を削減するため、このビットは論理1を書かれるべきです。

24. A/D変換器

24.1. 特徴

- 10/8ビット分解能
 - 積分非直線性誤差0.5 LSB
 - 絶対精度±2 LSB
 - 変換時間13～260μs (50kHz～1MHz変換クロック)
 - 最大分解能で15kSPS(採取/s)まで
 - 12チャンネルのシングルエンド入力多重器内蔵
 - 1,10,40,200倍の利得を提供する1つの差動増幅器
 - 温度感知器
- 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
 - 割り込み元の自動起動によるA/D変換開始
 - A/D変換完了割り込み
 - 休止形態雑音低減機能

ATmega16U4/32U4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B,D,Fの多数のピンピンから構成されたシングルエンド電圧入力を許す12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。

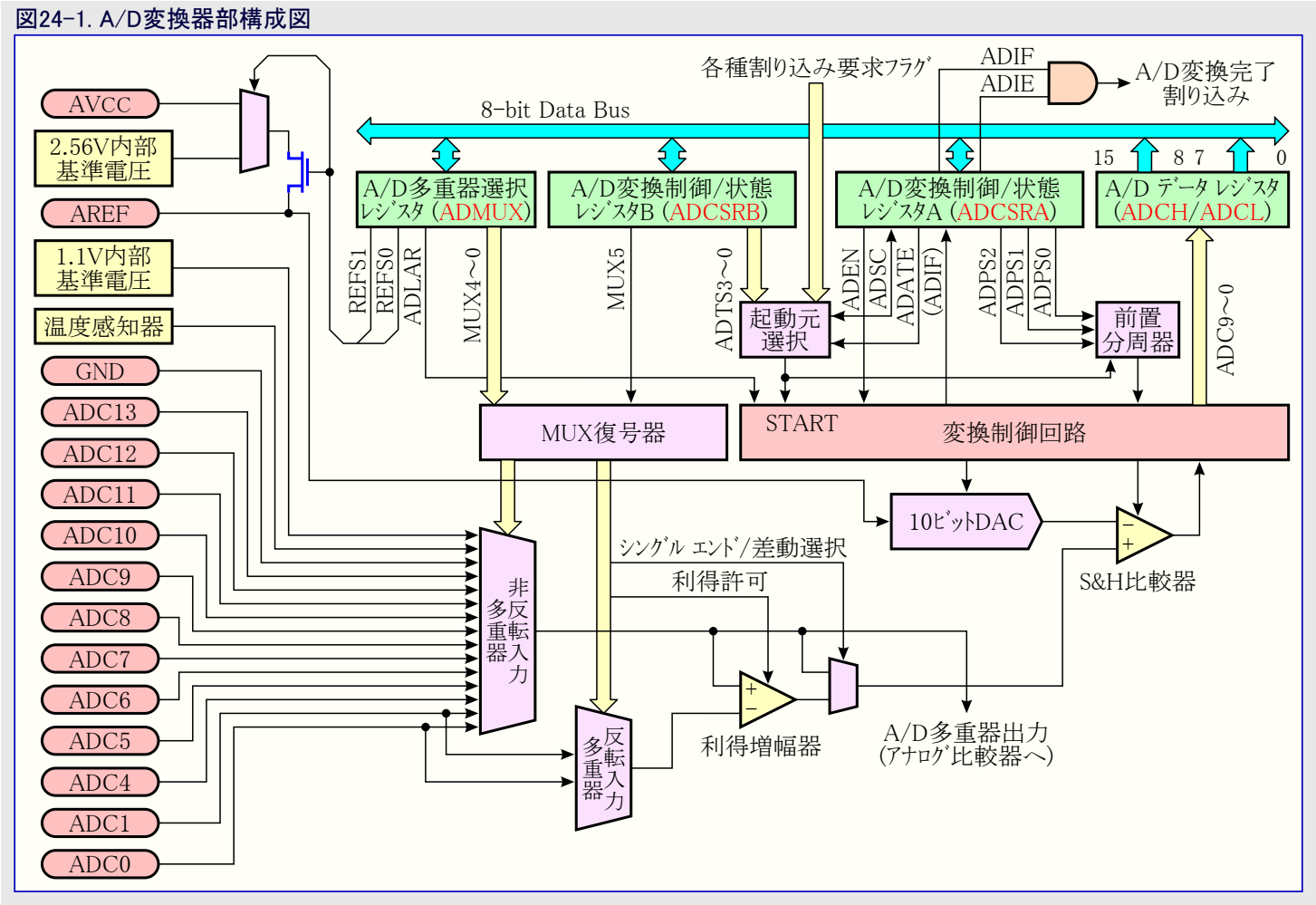
差動増幅器はA/D変換前の差動入力電圧で0dB(×1),20dB(×10),32dB(×40),46dB(×200)の増幅段を提供する設定変更可能な利得段を装備することによって、デバイスには32通りの差動電圧入力も支援します。2つの差動アナログ入力が共通反転端子(ADC0/ADC1)を共用し、一方他のADC入力が非反転入力端子として選べます。利得×1,×10,×40が使われる場合は8ビット分解能が期待でき、利得×200が使われる場合は7ビット分解能が期待できます。

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24-1.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200ページの「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。

28ページの「PRR0 - 電力削減レジスタ」のPRADCBITはA/D変換部を許可するために0を書かれなければなりません。(訳注:本行追加)



24.2. 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(REFS1,0)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。

アナログ入力チャンネルと差動利得はA/D変換制御/状態レジスタB(ADCSRB)のチャンネル選択(MUX5)ビットとADMUXのチャンネル選択(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(1.1V内蔵基準電圧(V_{BG}))だけでなく、どのADC入力ピン(ADC13~0)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。

差動チャンネルが選ばれると、差動利得段は選んだ入力チャン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ンネルが使われると、利得増幅器全体が迂回(無視)されます。(訳注:3行追加)

A/D変換部はA/D変換制御/状態レジスタA(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ンネルの選択は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、この結果が右揃え(16ビットのビット0側10ビット)で表されますが、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)で表せます。

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を読むことで足ります。さもないと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。

24.3. 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ンネルが選ばれると、A/D変換部はそのチャンネル変更を実行する前に現在の変換を済ませます。

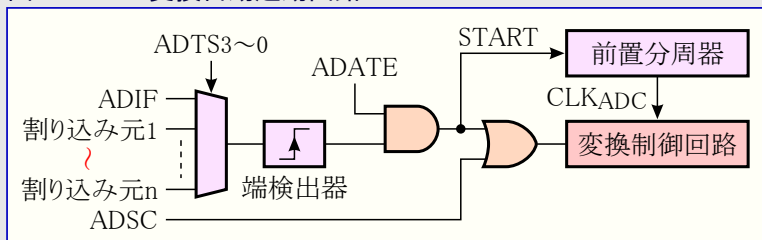
また、変換は様々な起動元によって自動的に起動できます。自動起動はA/D変換制御/状態レジスタA(ADCSRA)のA/D変換自動起動許可(ADATE)ビットの設定(1)によって許可されます。起動元はA/D変換制御/状態レジスタB(ADCSRB)のA/D変換起動元選択(ADTS3~0)ビットの設定によって選ばれます(起動元の一覧についてはADTSビットの記述をご覧ください)。選んだ起動信号上に上昇端が起きると、A/D変換用前置分周器がリセットし、変換が開始されます。これは一定間隔での変換開始の方法を提供します。変換完了時、起動信号が未だ設定(1)されている場合、新規の変換は開始されません。変換中にこの起動信号上で別の上昇端が起きると、その端(エッジ)は無視されます。指定した割り込みが禁止またはステータスレジスタ(SREG)の全割り込み許可(I)ビットが解除(0)でも、割り込み要求フラグが設定(1)されることに注意してください。従って割り込みを起こさずに変換が起動できます。けれども次の割り込み要因で新規変換を起動するために、割り込み要求フラグは解除(0)されなければなりません。

起動元としてA/D変換完了割り込み要求フラグ(ADIF)を使うことは、A/D変換器に実行中の変換が完了されると直ぐに新規変換を開始させます。そのためA/D変換器は連続動作で動き、継続的な採取(変換)とA/Dデータレジスタを更新します。最初の変換はADCSRA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)されるかどうかに関らず、連続的な変換を実行します。

自動起動が許可されている場合、ADCSRAのADSCビットに論理1を書くことによって単独変換を開始できます。ADSCは変換が進行中かを決めるためにも使えます。ADSCビットは変換がどう開始されたかに拘らず、変換中は1として読みます。

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200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)

図24-2. A/D変換自動起動回路



24.4. 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200kHzよりも高くできます。また、A/D変換制御/状態レジスタ(ADCSRB)の高速A/D変換動作(ADHSM)ビットの設定(1)が、より高い電力消費の犠牲でA/D変換クロック周波数の増加を許します。

A/D変換部は100kHz以上のどんなCPUクロックからも受け入れ可能なA/D変換クロック周波数を生成する前置分周器を含みます。この前置分周はA/D変換制御/状態レジスタA(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)される限り走行を保ち、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。差動変換タイミングでの詳細については次頁の「差動増幅チャネル」をご覧ください。

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(ADATE=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。

自動起動が使われると、前置分周器は起動要因発生時にリセットされます。これは起動要因から変換開始までの一定の遅延を保証します。この動作での採取&保持は起動要因となる信号の上昇後、2変換クロック周期で採取が行われます。同期化論理回路(端(エン)検出器)に対して、追加の3 CPUクロック周期が費やされます。

連続変換動作(ADATE=1)では変換完了後直ちに新規変換が開始され、一方ADSCは1に留まります。変換時間の概要については次頁の表24-1.をご覧ください。

図24-3. A/D変換前置分周器部構成

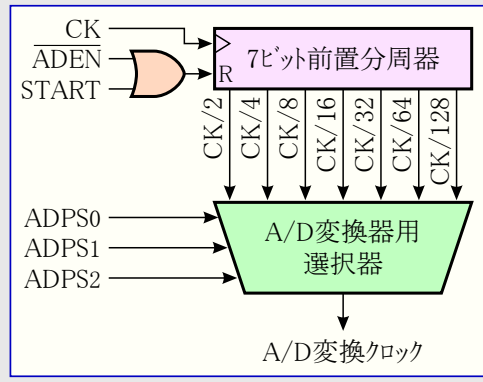


図24-4. 初回変換タイミング (単独変換動作)

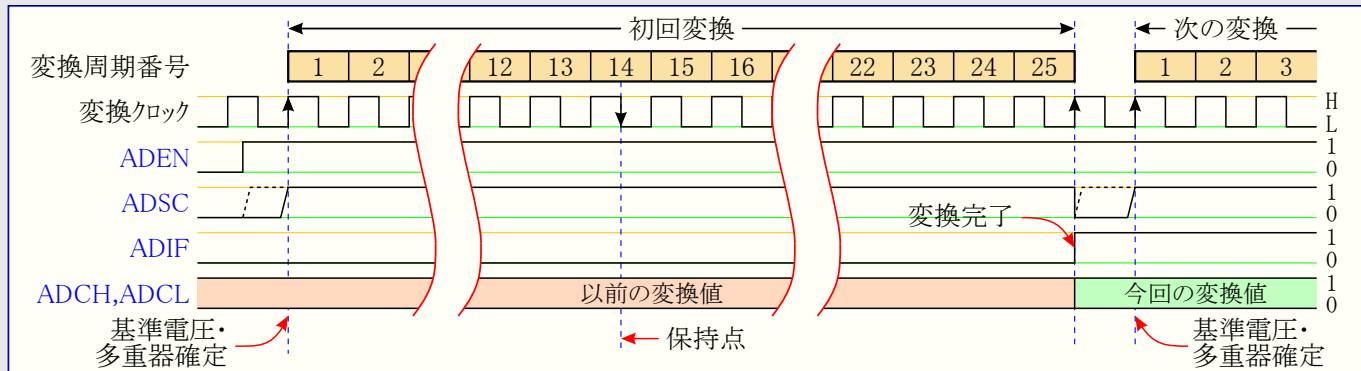


図24-5. 通常変換タイミング (単独変換動作)

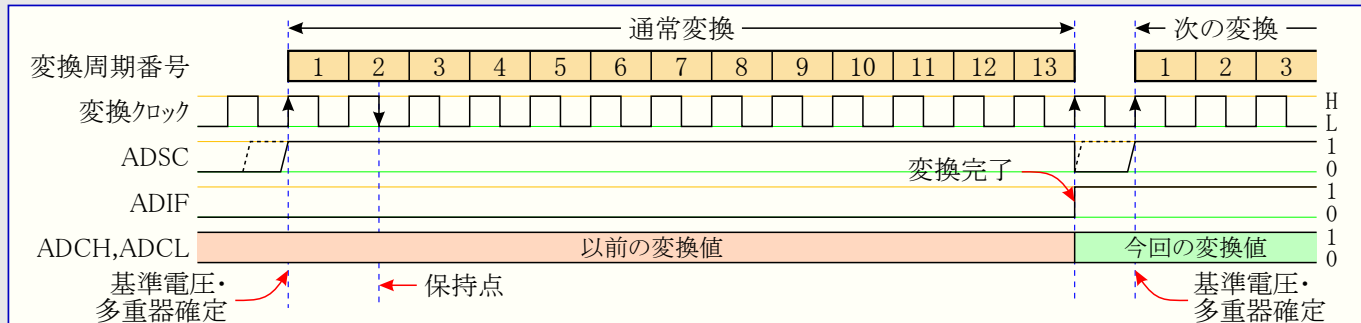


図24-6. 通常変換タイミング* (自動起動変換動作)

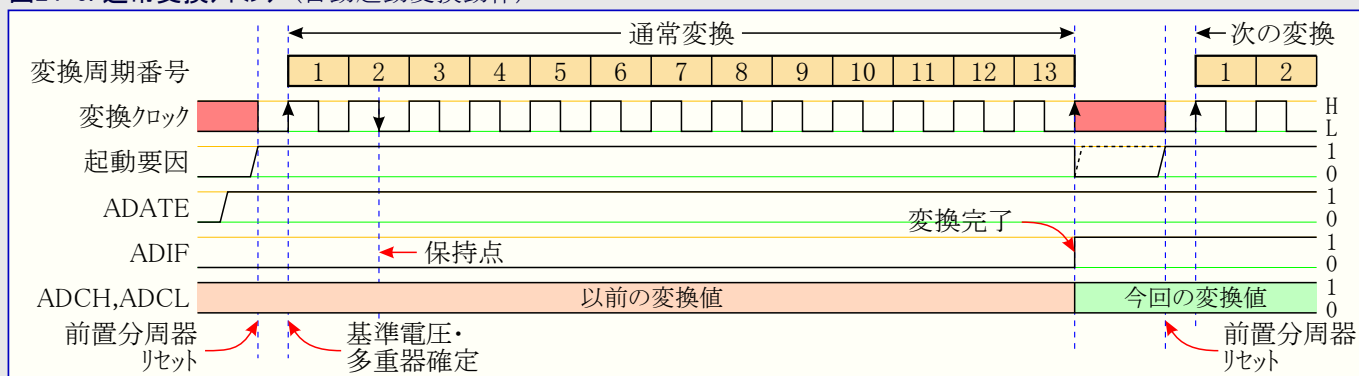


図24-7. 連続変換動作タイミング*

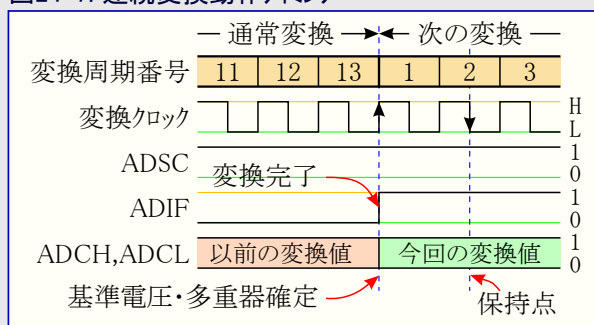


表24-1. A/D変換時間

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)
自動起動変換	1.5 (2)	13.5

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: CKADC2の状態に依存(訳注: 共通性から追加)。

24.4.1. 差動増幅チャネル

差動増幅チャネルを使う時は変換の±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う方法でA/D変換器インターフェースによって自動的行われます。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。

差増増幅チャネルが使われ、自動起動によって変換が開始される場合、A/D変換部は変換の間、OFFに切り替えられなければなりません。自動起動が使われるとき、A/D用前置分周器は変換が開始される前にリセットされます。利得段は変換以前の安定したA/D変換クロックに関係するため、この変換は有効ではありません。各変換間のA/D変換部の動作禁止とその後の許可(ADCSRAのADENへの0書き込み後の1書き込み)では、延長した変換(初回変換)だけが実行されます。この延長した変換からの結果は有効です。詳細タイミングについては196頁の「前置分周と変換タイミング」をご覧ください。

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6μs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。

24.5. 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)ビット及びA/D変換制御/状態レジスタB(ADCSRB)のチャネル選択(MUX5)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われることを保証します。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(A/D変換制御/状態レジスタA(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ADCSRBとADMUXに書かないことが推奨されます。

自動起動が使われる場合、起動要因の正確な時間は確定できません。変換が新規設定によって影響されるように制御するにはADMUXレジスタの更新時に特別な注意が払われなければなりません。

ADCSRAのA/D許可(ADEN)とA/D変換自動起動許可(ADATE)の両方が1を書かれると、何時でも割り込みが起き得ます。この期間でADMUXが変更されると、使用者は次の変換が旧設定または新設定どちらが基準にされるかを知ることができません。ADMUXは次の方法で安全に更新できます。

- ADENまたはADATEが解除(0)されているとき。
- 変換開始後、最低1変換クロック周期経過後の変換中。
- 変換後から、変換起動元として使った割り込みフラグが解除(0)される直前まで。

これら条件の1つでADMUXを更新すると、新設定は次のA/D変換に影響を及ぼします。

差動入力チャネルを変更する時に特別な注意が払われるべきです。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125µs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125µs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。

(入力信号の)設定時間と利得段帯域は高速A/D変換動作(ADHSM)ビット設定と無関係です。

JTAGインターフェース許可の場合、PORTF7~4のADC入力チャネル機能が無効にされます(51頁の表11-13を参照)。(訳注:本行追加)

24.5.1. 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、その後に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ふかもしれません。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。

24.5.2. 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 エント入力チャネルは\$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2.56V内部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253頁の表29-7.で示されるよりもAVCCに近くすべきではありません。

24.6. 温度感知器

ATmega16U4/32U4はA/D変換器を通して値を読むことができるチップ上の温度感知器を含みます。

温度測定はA/D変換器のシングルエンド入力と対にされたチップ上の温度感知器に基づきます。A/D変換制御/状態レジスタB(ADCSRB)とA/D多重器選択レジスタ(ADMUX)のチャネル選択(MUX5~0)ビットがこの温度感知器を許可します。温度感知器測定ではA/D変換基準元に対して2.56V内部基準電圧も選ばなければなりません。温度感知器が許可されると、A/D変換器は温度感知器上の電圧を測定するのに単独変換動作(ADATE=0)で使うことができます。

温度感知器とその駆動部はADCSRBとADMUXの値がA/D変換入力として温度感知器を選ぶ時に許可されます。この駆動部の伝播遅延は概ね2 μ sです。従って2回の逐次比較が必要とされます。正しい温度測定は2回目のそれです。

直前の変換中にADCSRBとADMUXを設定することにより、1変換に対してこのタイミングを減らすこともできます。実際に直前の変換開始事象の開始直後にADCSRBとADMUXは温度感知器を選ぶように設定することができ、その後、駆動部は温度感知器測定の採取/保持段階の2 μ s前には許可されます。

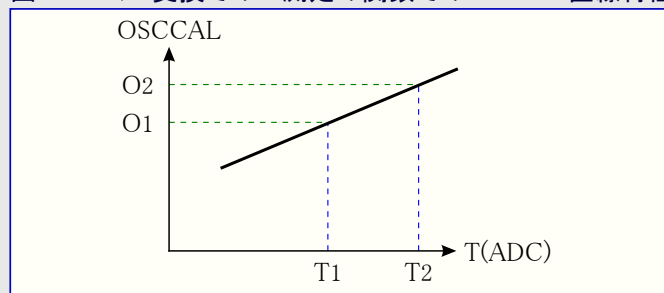
24.6.1. 感知器校正

感知器初期公差は大きい($\pm 10^{\circ}\text{C}$)けれど、その特性は直線です。従って、応用が精度を必要とするなら、ファームウェアは直接温度測定にこの感知器を使うのに校正段階を含めなければなりません。

この感知器の他の応用はOSCCALレジスタ(20頁の「OSCCAL – 発振校正レジスタ」参照)を通して使用者によって周波数を調整することができる校正付き内蔵RC発振器に関係するかもしれません。製造中に2つの温度(許容誤差 $\pm 10^{\circ}\text{C}$ (注1)で $+25^{\circ}\text{C}$ と $+85^{\circ}\text{C}$)で校正が行われます。各々の温度に於いて、温度感知器値(T_i)が測定されてEEPROM内に格納され(注2)、そしてOSCCAL校正值 O_i (換言すると、正確な8MHz出力を持つためにこの温度でOSCCALレジスタに設定されるべき値)が別のメモリ領域に格納されます。

これら4つの値と温度感知器の直線特性と内蔵RC発振器により、ファームウェアは実行中に温度感知器測定の関数でRC発振器を容易に再校正することができます(注3)(応用記述がこの操作を記述します)。

図24-8. A/D変換での T° 測定の関数でのOSCCAL直線特性



注1: 製造中の校正温度が正確でない($\pm 10^{\circ}\text{C}$)ので、温度感知器校正值は正確な温度測定を行うのに使うことができません。

注2: EESAVEヒューズが非プログラム(1)のままの場合、どのチップ消去操作もEEPROM内に含まれる温度感知器校正值を解除することに注意してください。

注3: 温度の関数でのOSCCALのソフトウェア再校正後の正確な結果は特性付けによって与えられます。

24.7. 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(**ADATE=0**)され、且つA/D変換完了割り込みが許可(**ADIE=1**)されていなければなりません。
2. A/D変換雑音低減(またはアイドル)動作に移行してください。一旦CPUが停止されてしまうと、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れなければ、A/D変換完了割り込みはCPUを起動してA/D変換完了割り込みルーチンを実行します。A/D変換完了前に他の割り込みがCPUを起動すると、その割り込みが実行され、A/D変換完了割り込み要求はA/D変換完了時に生成されます。CPUは新規**SLEEP**命令が実行されるまで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**→**1**)に切り替えることが推奨されます。

24.7.1. アナログ入力回路

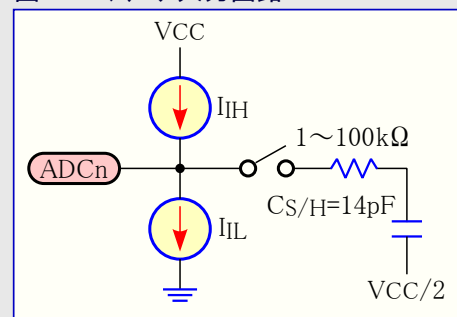
シングルエンド入力チャネルのアナログ回路は**図24-9**で図示されます。ADCnに印加したアナログ(信号)源はそのチャネルがADC入力として選ばれているかどうかにかかわらず、ピン容量とそのピンの漏れ電流に左右されます。そのチャネルが選ばれると、(アナログ信号)源は直列抵抗(入力経路の合成抵抗)を通してS/Hコンデンサを駆動しなければなりません。

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いインピーダンス(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24-9. アナログ入力回路



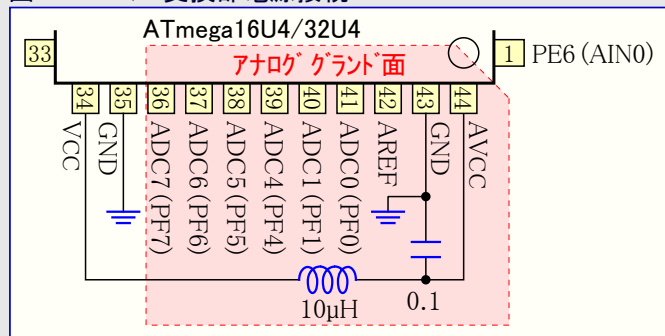
24.7.2. アナログ雑音低減技術

デバイス内外のデジタル回路がアナログ測定の精度に影響を及ぼすかもしれないEMIが発生します。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も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。アナログ信号線がアナログ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24-10**で示されるようにLC濾波器経由でデジタル供給電圧(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、切り替わらない事が重要です。

(訳注) **図24-10**のピン名が一部省略されています。

図24-10. A/D変換部電源接続



注: 同じ回路がADC13~8側のAVCC濾波に使われるべきです。

24.7.3. 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。

24.7.4. 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2ⁿ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2ⁿ-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24-11.

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24-12.

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24-13.

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24-14.

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±0.5 LSB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24-11. 変位(オフセット)誤差

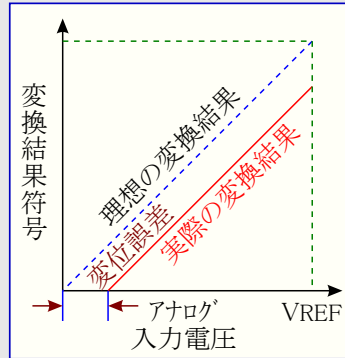


図24-12. 利得誤差

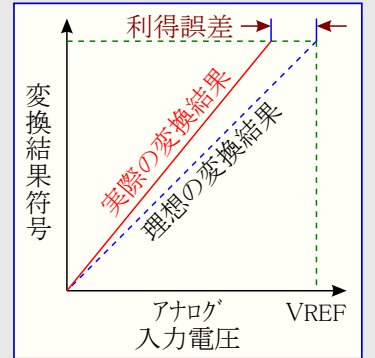


図24-13. 積分非直線性誤差

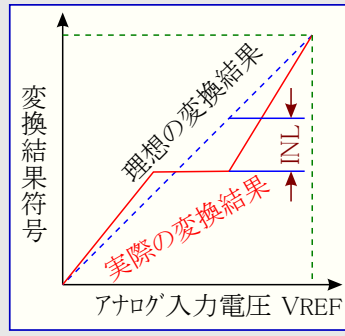
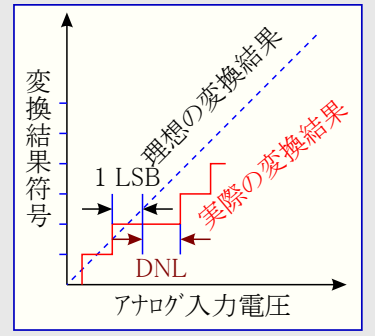


図24-14. 微分非直線性誤差



24.8. 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(202頁の表24-3と表24-4をご覧ください)。\$000はアナログGNDを表し、\$3FFは選んだ基準電圧-1 LSBを表します。

$$ADC = \frac{V_{IN} \times 1023}{V_{REF}}$$

差動チャネルが使われると、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24-15は差動入力範囲の符号化を示します。

$$ADC = \frac{(V_{POS} - V_{NEG}) \times GAIN \times 512}{V_{REF}}$$

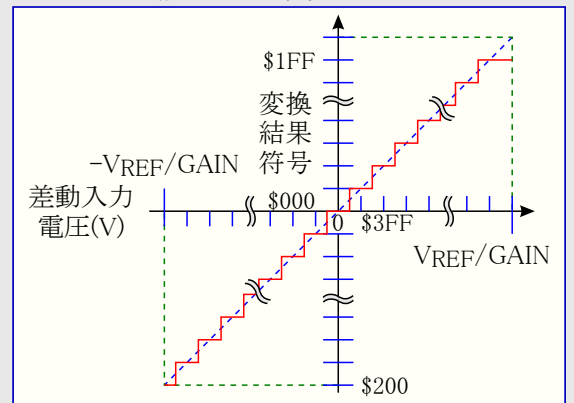
表24-2は差動入力チャネルの組(ADCn-ADCm)が基準電圧(VREF)で選ばれた場合の出力符号の結果を示します。

表24-2. 入力電圧と出力符号の関係

V _{ADCn}	読み出し符号	対応する10進値
V _{ADCm} +V _{REF} /GAIN	\$1FF	511
V _{ADCm} +(511/512)V _{REF} /GAIN	\$1FF	511
V _{ADCm} +(510/512)V _{REF} /GAIN	\$1FE	510
}	}	}
V _{ADCm} +(1/512)V _{REF} /GAIN	\$001	1
V _{ADCm}	\$000	0
V _{ADCm} -(1/512)V _{REF} /GAIN	\$3FF	-1
}	}	}
V _{ADCm} -(511/512)V _{REF} /GAIN	\$201	-511
V _{ADCm} -V _{REF} /GAIN	\$200	-512

例: ADMUX=\$E9, MUX5=0(ADC1とADC0差動10倍入力、VREF=2.56V、左揃え)で、ADC1が300mV、ADC0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。

図24-15. 差動入力計測範囲



24.9. A/D変換用レジスタ

24.9.1. ADMUX – A/D多重器選択レジスタ (ADC Multiplexer Select Register)

ビット (\$7C)	7	6	5	4	3	2	1	0	
	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24-3.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24-3. 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と内部基準電圧は切り離されます。)
0	1	AVCC (内部基準電圧は切り離され、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、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204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変換制御/状態レジスタB(ADCSRB)のMUX5と組み合わせられ、A/D変換器にどのアナログ入力との組み合わせが接続されるかを選びます。これらのビットは表で示されるように差動チャネルに対する利得も選びます。これらのビットが変換中に変更される場合、その変更は変換が完了する(ADCSR AのADIF=1)まで実施しません。

表24-4. アナログ入力チャネルと利得の選択

MUX5~0	差動入力時			S/D	MUX5~0	差動入力時			S/D	MUX5~0	差動入力時			S/D
	非反転	反転	利得			非反転	反転	利得			非反転	反転	利得	
0 00000	ADC0			シングル エンド 入力	0 10110	ADC6	ADC1	×1	差動 入力	1 01100	ADC4	ADC1	×10	差動 入力
0 00001	ADC1				0 10111	ADC7	ADC1	×1		1 01101	ADC5	ADC1	×10	
0 00010	(利用不可)				0 11000	(利用不可)				1 01110	ADC6	ADC1	×10	
0 00011	(利用不可)				0 11001	(利用不可)				1 01111	ADC7	ADC1	×10	
0 00100	ADC4				0 11010	(利用不可)				1 10000	ADC4	ADC0	×40	
0 00101	ADC5				0 11011	(利用不可)				1 10001	ADC5	ADC0	×40	
0 00110	ADC6				0 11100	(利用不可)				1 10010	ADC6	ADC0	×40	
0 00111	ADC7				0 11101	(利用不可)				1 10011	ADC7	ADC0	×40	
0 01000	(利用不可)			差動 入力	0 11110	1.1V(VBG)			シングル エンド 入力	1 10100	ADC4	ADC1	×40	差動 入力
0 01001	ADC1	ADC0	×10		0 11111	0V(GND)				1 10101	ADC5	ADC1	×40	
0 01010	(利用不可)				1 00000	ADC8				1 10110	ADC6	ADC1	×40	
0 01011	ADC1	ADC0	×200		1 00001	ADC9				1 10111	ADC7	ADC1	×40	
0 01100	(利用不可)				1 00010	ADC10				1 11000	ADC4	ADC0	×200	
0 01101	(利用不可)				1 00011	ADC11				1 11001	ADC5	ADC0	×200	
0 01110	(利用不可)				1 00100	ADC12				1 11010	ADC6	ADC0	×200	
0 01111	(利用不可)				1 00101	ADC13				1 11011	ADC7	ADC0	×200	
0 10000	ADC0	ADC1	×1		1 00110	ADC1	ADC0	×40	D	1 11100	ADC4	ADC1	×200	
0 10001	(利用不可)				1 00111	温度感知器			S	1 11101	ADC5	ADC1	×200	
0 10010	(利用不可)				1 01000	ADC4	ADC0	×10	差動 入力	1 11110	ADC6	ADC1	×200	
0 10011	(利用不可)				1 01001	ADC5	ADC0	×10		1 11111	ADC7	ADC1	×200	
0 10100	ADC4	ADC1	×1		1 01010	ADC6	ADC0	×10						
0 10101	ADC5	ADC1	×1		1 01011	ADC7	ADC0	×10						

注: MUX5ビットはADCSRBレジスタの一部になります。

24.9.2. ADCSRA – A/D変換 制御/状態レジスタA (ADC Control and Status Register A)

ビット (\$7A)	7	6	5	4	3	2	1	0	
	ADEN	ADSC	ADATE	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にされます。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 – ADATE : A/D変換自動起動許可 (ADC Auto Trigger Enable)

このビットが**1**を書かれると、A/D変換の自動起動が許可されます。A/D変換器は選んだ起動信号の上昇端で変換を開始します。この起動元はA/D変換制御/状態レジスタB(ADCSR)のA/D変換起動要因選択(ADTS3~0)ビット設定によって選ばれます。

● ビット4 – ADIF : A/D変換完了割り込み要求フラグ (ADC Interrupt Flag)

A/D変換が完了し、A/Dデータレジスタが更新されると、このフラグが設定(**1**)されます。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(訳注:アドレス範囲外のため、本行は不適切です)。

● 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24-5. 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

24.9.3. ADCSRB – A/D変換 制御/状態レジスタB (ADC Control and Status Register B)

ビット (\$7B)	7	6	5	4	3	2	1	0	
	ADHSM	ACME	MUX5	–	ADTS3	ADTS2	ADTS1	ADTS0	ADCSR
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

● ビット7 – ADHSM : 高速A/D変換動作 (ADC High Speed Mode)

このビットへの**1**書き込みは高速A/D変換動作を許可します。この動作は、より高い電力消費の犠牲で、より高い変換速度を可能にします。

● ビット5 – MUX5 : アナログ チャネル追加選択ビット (Analog Channel Additional Selection Bit)

このビットはA/D変換器へ接続されるアナログ入力(の差動増幅器構成設定を含む)組み合わせを選ぶ、ADCSRとA/D多重器選択(ADMUX)レジスタのMUX5~0ビットの一部になります。

●ビット3～0 – ADTS3～0 : A/D変換自動起動要因選択 (ADC Auto Trigger Source)

A/D変換制御/状態レジスタA(ADCSRA)のA/D変換自動起動許可(ADATE)ビットが1を書かれると、これらのビット値はどの起動元がA/D変換を起動するのかわを選びます。ADATEが解除(0)されると、ADTS3～0設定は無効です。変換は選んだ割り込みフラグの上昇端によって起動されます。解除(0)されている起動元から設定(1)されている起動元への切り替えが、起動信号上に上昇端を生成することに注意してください。ADCSRAのA/D許可(ADEN)ビットが設定(1)されているなら、これが変換を開始させます。連続変換動作(ADTS3～0=0)への切り替えは、例えばA/D変換完了割り込み要求フラグが設定(1)されていても、起動事象を引き起こしません。

表24-6. A/D変換自動起動元選択				
ADTS3	ADTS2	ADTS1	ADTS0	起動元
0	0	0	0	連続変換動作
0	0	0	1	アナログ比較器
0	0	1	0	外部割り込み要求0
0	0	1	1	タイマ/カウンタ0比較A一致
0	1	0	0	タイマ/カウンタ0溢れ
0	1	0	1	タイマ/カウンタ1比較B一致
0	1	1	0	タイマ/カウンタ1溢れ
0	1	1	1	タイマ/カウンタ1捕獲要求
1	0	0	0	タイマ/カウンタ4溢れ
1	0	0	1	タイマ/カウンタ4比較A一致
1	0	1	0	タイマ/カウンタ4比較B一致
1	0	1	1	タイマ/カウンタ4比較D一致

24.9.4. ADCH,ADCL – A/Dデータレジスタ (ADC Data Register)

ADLAR=0時								
ビット (\$79)	15	14	13	12	11	10	9	8
	–	–	–	–	–	–	ADC9	ADC8
Read/Write	R	R	R	R	R	R	R	R
初期値	0	0	0	0	0	0	0	0
ビット (\$78)	7	6	5	4	3	2	1	0
	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0
Read/Write	R	R	R	R	R	R	R	R
初期値	0	0	0	0	0	0	0	0
ADLAR=1時								
	15	14	13	12	11	10	9	8
	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2
	7	6	5	4	3	2	1	0
	ADC1	ADC0	–	–	–	–	–	–

A/D変換が完了すると、その結果がこれらの2レジスタで得られます。差動チャネルが使われると、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(差動入力チャネルについては符号+7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。さもないとADCLが初めに、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チャネル選択(MUX4～0)ビットとA/D制御/状態レジスタB(ADCSRB)のMUX5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201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。

24.9.5. DIDR0 – デジタル入力禁止レジスタ0 (Digital Input Disable Register 0)

ビット (\$7E)	7	6	5	4	3	2	1	0	
	ADC7D	ADC6D	ADC5D	ADC4D	–	–	ADC1D	ADC0D	DIDR0
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

● ビット7～4,1,0 – ADC7D～ADC4D,ADC1D,ADC0D : ADC7～4,1,0 デジタル入力禁止 (ADC7～4,1,0 Digital Input Disable)

このビットが論理1を書かれると、対応するADCnピンのデジタル入力緩衝部が禁止されます。このビットが設定(1)されると、対応するポート入力レジスタのビット(PINx)は常に0として読みます。アナログ信号がADC7～4,1,0ピンに印加され、そのピンからのデジタル入力が必要とされない時にデジタル入力緩衝部での消費電力を削減するため、そのビットは論理1を書かれるべきです。

24.9.6. DIDR2 – デジタル入力禁止レジスタ2 (Digital Input Disable Register 2)

ビット (\$7D)	7	6	5	4	3	2	1	0	
	–	–	ADC13D	ADC12D	ADC11D	ADC10D	ADC9D	ADC8D	DIDR2
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5～0 – ADC13D～ADC8D : ADC13～8 デジタル入力禁止 (ADC13～8 Digital Input Disable)

このビットが論理1を書かれると、対応するADCnピンのデジタル入力緩衝部が禁止されます。このビットが設定(1)されると、対応するポート入力レジスタのビット(PINx)は常に0として読みます。アナログ信号がADC13～8ピンに印加され、そのピンからのデジタル入力が必要とされない時にデジタル入力緩衝部での消費電力を削減するため、そのビットは論理1を書かれるべきです。

25. JTAGインターフェースと内蔵デバッグ機能

25.1. 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

25.2. 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。

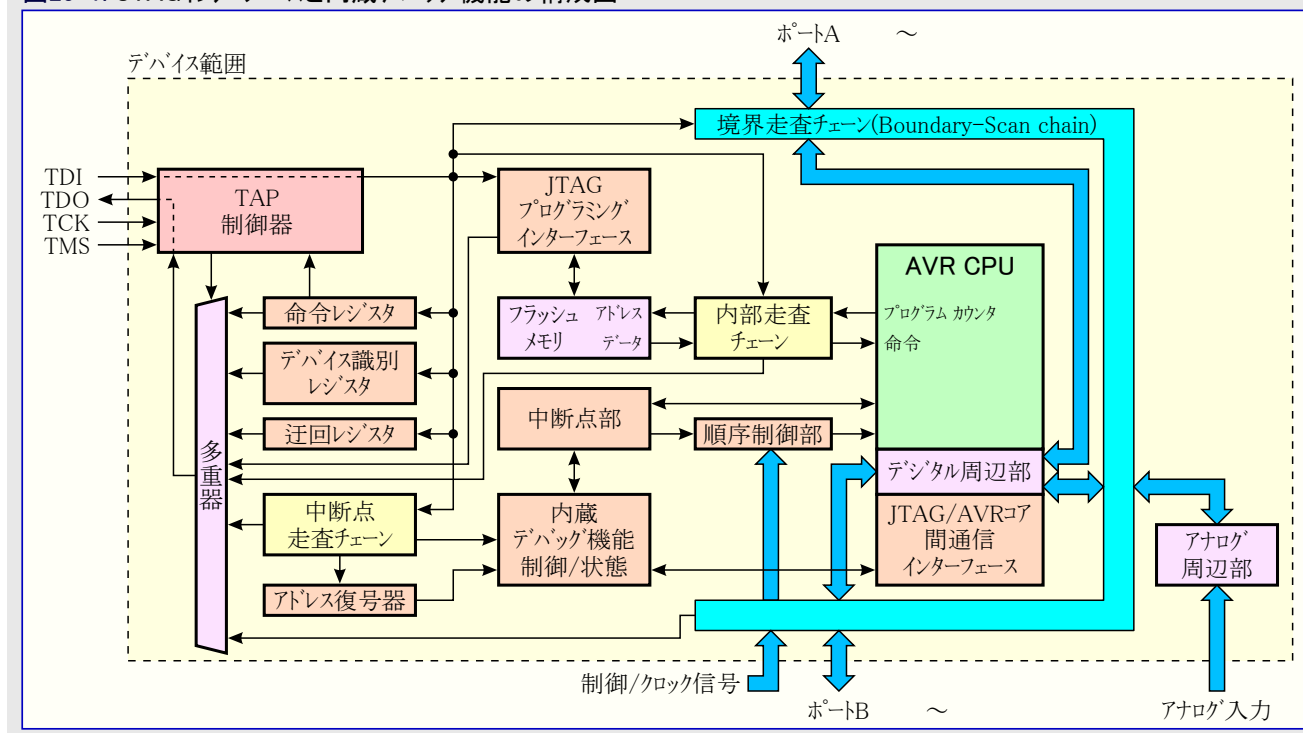
- JTAG境界走査(Boundary-Scan)能力による基板などの試験
- 不揮発性メモリ、ヒューズ ビット、施錠ビットの読み書き(プログラミング)
- 内蔵デバッグ機能による実装デバッグ

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241頁の「JTAGインターフェース経由のプログラミング」と210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27-1はJTAGインターフェースと内蔵デバッグ機能の構成図を示します。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25-1. JTAGインターフェースと内蔵デバッグ機能の構成図



25.3. 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制御レジスタ(MCUCR)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。

25.4. 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25-2.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。

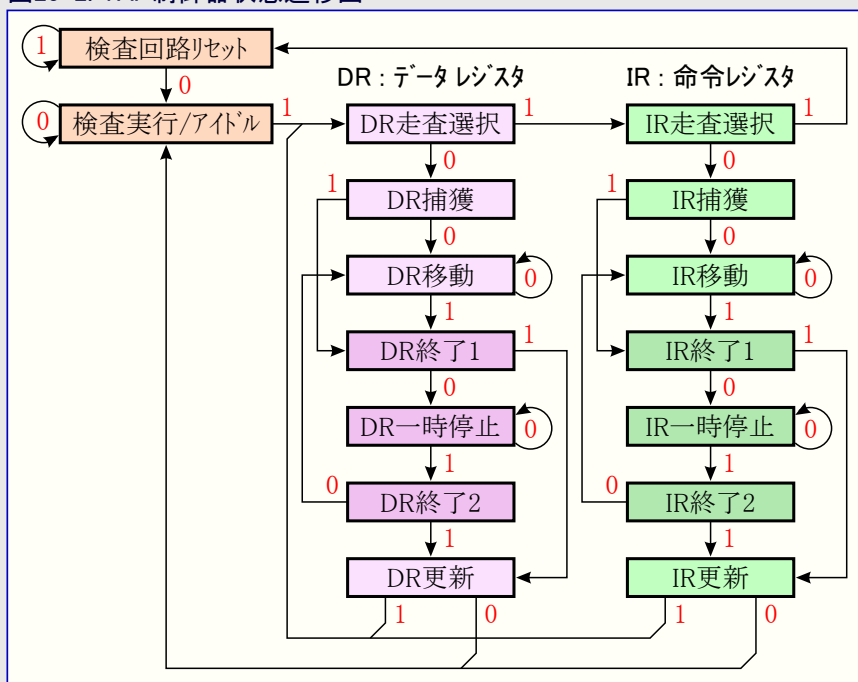
- 命令レジスタ(IR)移動状態へ移行するためにTCKの上昇でTMSへ順次1,1,0,0を与えます。この状態中、TCKの上昇端でTDI入力からJTAG命令レジスタ内に4ビットのJTAG命令を移動します。TMS入力はIR移動状態に留まるために3LSBの入力中、Lowに保持されなければなりません。命令のMSBはTMS入力のHigh設定によってこの状態を抜ける時に移動入力されます。命令がTDIピンから移されている間、捕獲されたIRの状態(\$01)がTDOピンへ移動出力されます。JTAG命令はTDIとTDO間に通ず(接続する)特定のデータレジスタを選び、選んだデータレジスタ周辺回路を制御します。
- 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。
- データレジスタ移動(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209頁の「参考文献」に記載された文献を参照してください。

図25-2. TAP制御器状態遷移図



25.5. 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210頁の「IEEE 1149.1(JTAG)境界走査(Boundary-Scan)」項で与えられます。

25.6. 内蔵デバッグ機能の使用

図25-1.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(ブレークポイント)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。

25.7. 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)

25.8. 内蔵デバッグに関連するI/Oメモリ内のレジスタ

25.8.1. 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	
\$31 (\$51)	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。

25.9. JTAGプログラミング能力の使用

JTAG経由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制御レジスタ(MCUCR)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。

JTAGインターフェースを通すプログラミングとプログラミング特殊JTAG命令の記述は241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。

25.10. 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

26. IEEE 1149.1 (JTAG) 境界走査(Boundary-Scan)

26.1. 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

26.2. 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制御レジスタ(MCUCR)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とされません。

26.3. 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)

26.3.1. 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。

26.3.2. デバイス識別 (Device Identification) レジスタ

図26-1. はデバイス識別レジスタの構造を示します。

図26-1. 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ます。改訂Aは\$00, 改訂B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。ATmega16U4/32U4のJTAG部品番号は右表で一覧されます。

表26-1. AVR JTAG 部品番号

部品番号	JTAG部品番号(16進数)
ATmega16U4	\$9488
ATmega32U4	\$9587

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製造者識別は右で示されます。

表26-2. 製造者ID

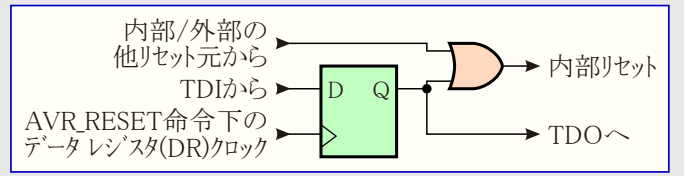
製造業者	JTAG製造者ID番号(16進数)
Atmel	\$01F

26.3.3. 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(18ページの「クロック元」参照)、リセットに留まります。図23-2.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26-2. リセットレジスタ



26.3.4. 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上の設定と監視能力があります。

完全な記述については213ページの「境界走査チェーン(Boundary-Scan chain)」をご覧ください。

26.4. 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)

26.4.1. 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。

26.4.2. 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。

26.4.3. 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。

26.4.4. 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。

26.4.5. 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。

26.5. 境界走査(Boundary-Scan)に関連するI/Oメモリ内のレジスタ

26.5.1. MCUCR – MCU制御レジスタ (MCU Control Register)

MCU制御レジスタは一般MCU機能制御ビットを含みます。

ビット	7	6	5	4	3	2	1	0	
\$35 (\$55)	JTD	–	–	PUD	–	–	IVSEL	IVCE	MCUCR
Read/Write	R/W	R	R	R/W	R	R	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

● 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回書き込まなければなりません。内蔵デバッグシステム使用時、このビットが切り替えられてはならないことに注意してください。

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。(訳注:共通性から前2行追加)

26.5.2. MCUSR – MCU状態レジスタ (MCU Status Register)

MCU状態レジスタは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)	–	–	USBRF	JTRF	WDRF	BORF	EXTRF	PORF	MCUSR
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	内容参照	内容参照	内容参照	内容参照	内容参照	内容参照	

● 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。

26.6. 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。

26.6.1. デジタルポートピンの走査

図26-3.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26-4.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26-3.からの境界走査セルの詳細が図26-4.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(けれどもIDは同期化回路を持ず)、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26-4.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。

JTAG IRがEXTESTまたはSAMPLE_PRELOADを内包すると、クロックは例えばCKOUTヒューズがプログラム(0)されていてもポートピンに送出されません。例えばクロックが出力でも、JTAG IRがSAMPLE_PRELOADを内包するとき、クロックは境界走査によって採取されません。

図26-3. 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

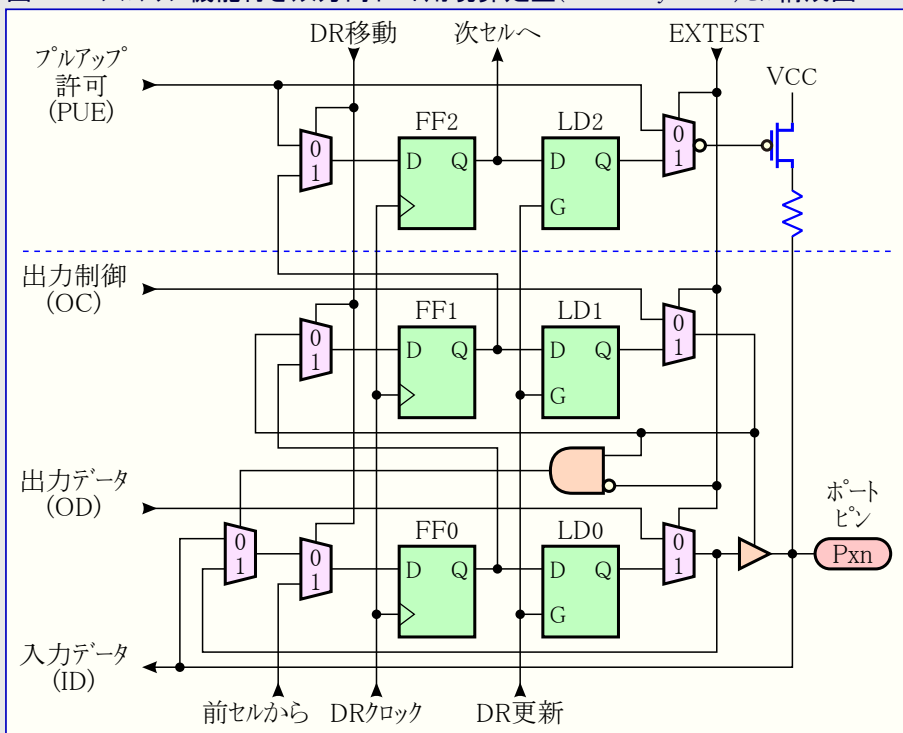
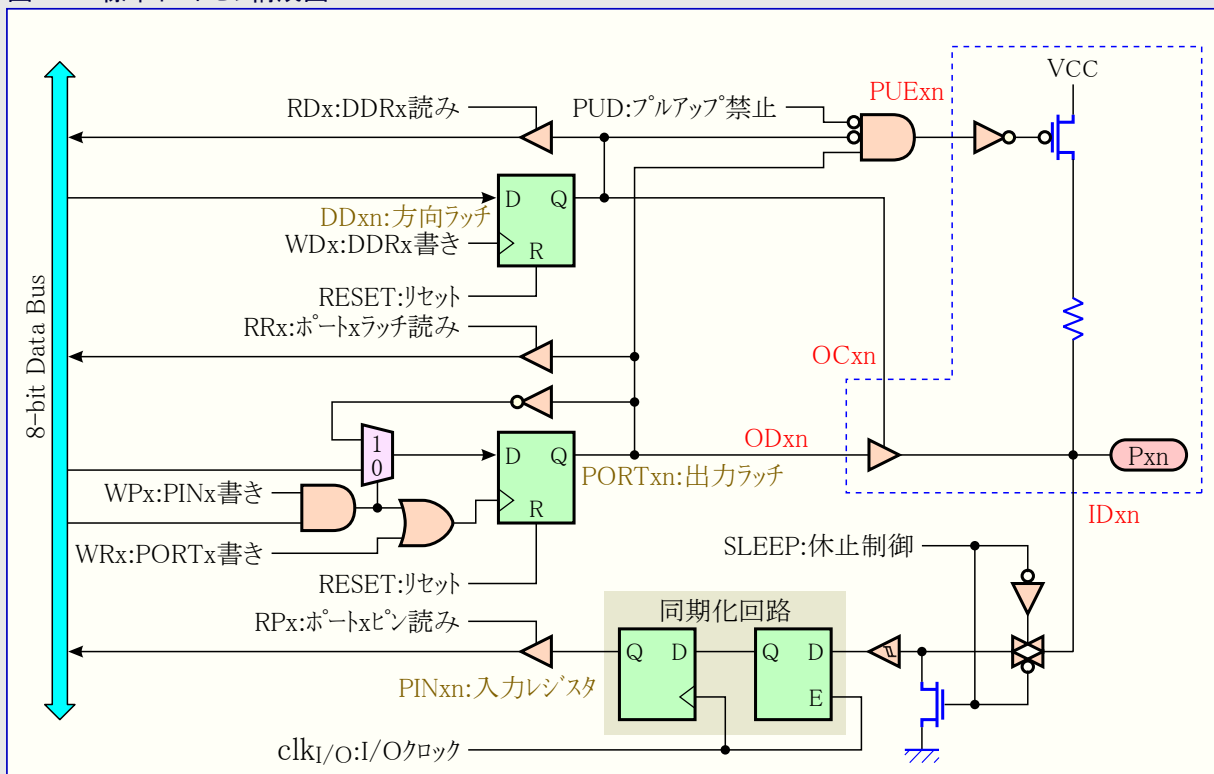


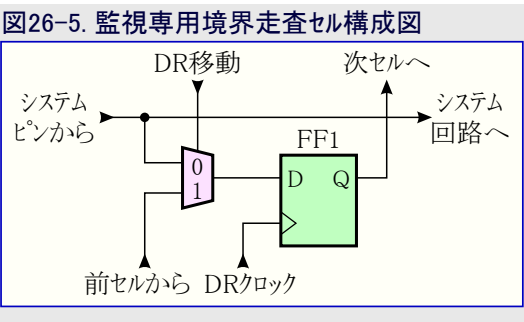
図26-4. 標準ポートピン構成図



注: 詳細については境界走査(Boundary-Scan)をご覧ください。

26.6.2. 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26-5.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。



26.7. ATmega16U4/32U4の境界走査(Boundary-Scan)順

下表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26-3.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26-3. ATmega16U4/32U4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
88	PE6.Data	ポートE	58	PD1.Control	ポートD	28	(予約)	(予約)
87	PE6.Control		57	PD2.Data		27	(予約)	
86	(予約)		56	PD2.Control		26	(予約)	
85	(予約)		55	PD3.Data		25	PE2.Data	ポートE
84	(予約)		54	PD3.Control		24	PE2.Control	
83	(予約)	ポートB	53	PD4.Data	(予約)	23	(予約)	(予約)
82	PB0.Data		52	PD4.Control		22	(予約)	
81	PB0.Control		51	PD5.Data		21	(予約)	
80	PB1.Data		50	PD5.Control		20	(予約)	
79	PB1.Control		49	PD6.Data		19	(予約)	
78	PB2.Data	ポートE	48	PD6.Control	(予約)	18	(予約)	(予約)
77	PB2.Control		47	PD7.Data		17	(予約)	
76	PB3.Data		46	PD7.Control		16	(予約)	
75	PB3.Control		45	(予約)		15	(予約)	
74	PB4.Data		44	(予約)		14	(予約)	
73	PB4.Control	ポートF	43	(予約)	(予約)	13	(予約)	(予約)
72	PB5.Data		42	(予約)		12	(予約)	
71	PB5.Control		41	(予約)		11	(予約)	
70	PB6.Data		40	(予約)		10	(予約)	
69	PB6.Control		39	(予約)		9	(予約)	
68	PB7.Data	ポートD	38	(予約)	(予約)	8	(予約)	(予約)
67	PB7.Control		37	(予約)		7	(予約)	
66	(予約)		36	(予約)		6	(予約)	
65	(予約)		35	(予約)		5	(予約)	
64	(予約)		34	(予約)		4	PF2.Control	ポートF
63	(予約)	リセット(監視専用)	33	(予約)	(予約)	3	PF1.Data	
62	RSTT		32	(予約)		2	PF1.Control	
61	PD0.Data		31	(予約)		1	PF0.Data	
60	PD0.Control		30	(予約)		0	PF0.Control	
59	PD1.Data		29	(予約)				

26.8. 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。ATmega16U4/32U4用のBSDLファイルは入手可能です。

27. 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。SPMとLPMの一般情報は6頁の「AVR CPUコア」で提供されます。

27.1. 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-モディファイライト)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(230頁の表28-11.参照)。このページ構成は通常動作に影響を及ぼしません。

27.2. 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27-2.参照)。各領域の容量は225頁の表27-8.と図27-2.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。

27.2.1. 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。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。217頁の表27-2.参照してください。

27.2.2. 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にアクセスできます。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。217頁の表27-3.参照してください。

27.3. 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216頁の図27-2.と225頁の表27-9.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。

27.3.1. 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/状態レジスタ(SPMCSR)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219頁の「SPMCSR – SPM命令制御/状態レジスタ」を参照してください。

27.3.2. 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27-1. 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27-1. RWW領域とNRWW領域の関係

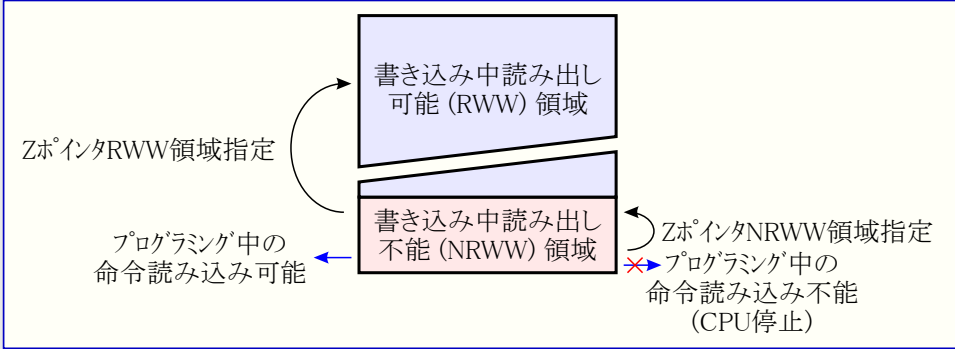
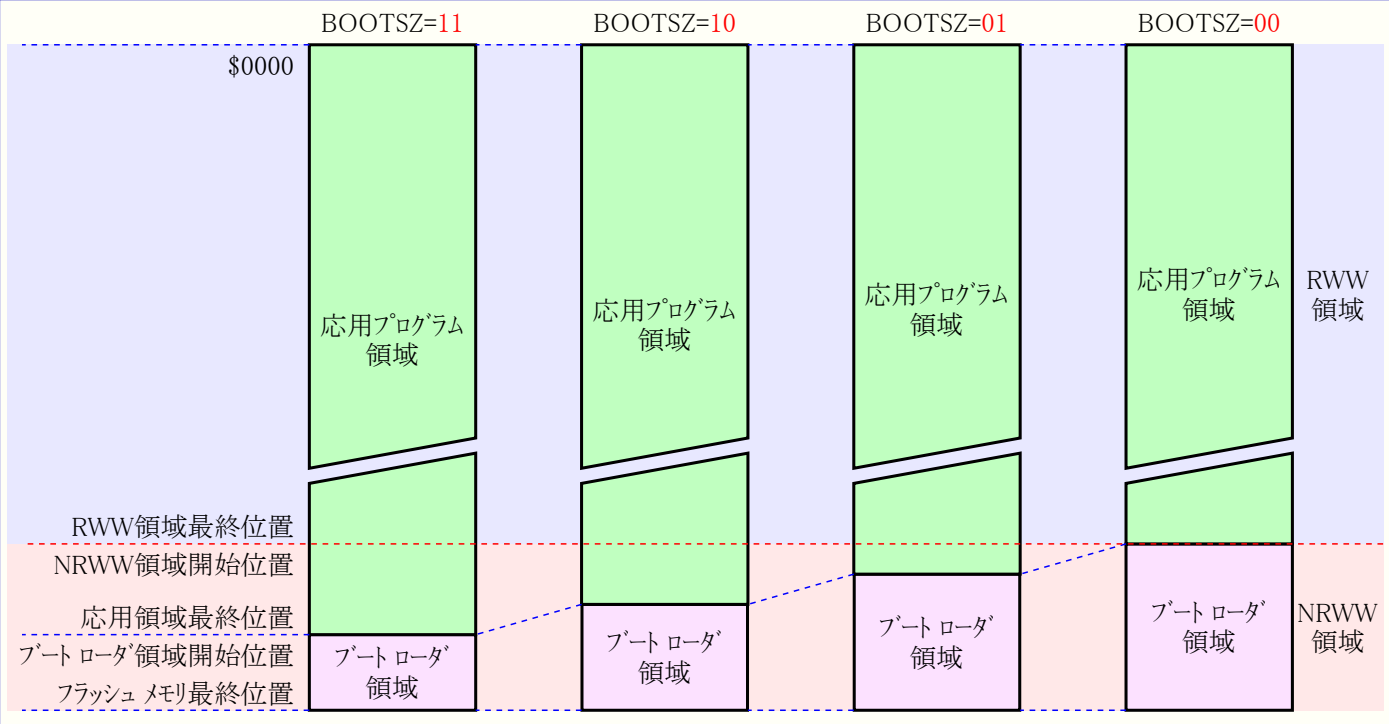


図27-2. 選択によるプログラム用フラッシュメモリの領域分割



注: 図内の各パラメータについては225頁の表27-8.で与えられます。

27.4. 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以下の2つの表をご覧ください。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27-2. 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27-3. 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。

27.5. ブート ロータ プログラムへの移行

ブート ロータは3つの異なる状況で実行できます。

27.5.1. 通常応用状況

応用プログラムから無条件分岐(Jump)またはサブルーチン呼び出し(Call)を行ってください。これはUSART、SPIまたはUSB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。

27.5.2. ブート リセット ヒューズ

リセット後にリセット ベクタがブート ロータ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 ロータ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 ロータ)は応用コードの実行を開始することができます。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 ロータ リセットを指示し、このヒューズが**直列プログラミング**または**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27-4. 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 ロータ リセット	ブート ロータ開始アドレス (225頁の表27-8.参照)
1	応用リセット	\$0000

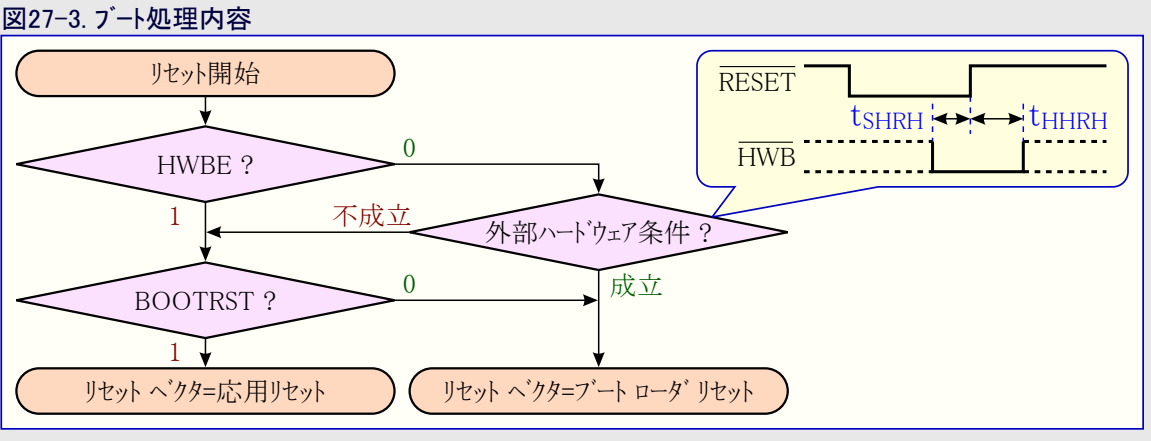
27.5.3. 外部ハードウェア状況

リセット下の特別なハードウェア条件でブート ロータの実行がリセット後に強制されるように**ハードウェア ブート許可(HWBE)ヒューズ**をプログラム(0)にすることができます(下表をご覧ください)。

HWBEヒューズ許可時、PE2/ $\overline{\text{HWB}}$ ピンはリセット間に入力として設定され、リセットの上昇端中に採取されます。リセットの上昇端の間中でPE2/ $\overline{\text{HWB}}$ ピンが0のとき、リセット ベクタはブート ロータ リセット アドレスとして設定され、ブート ロータが実行されます(図27-3.をご覧ください)。

表27-5. ハードウェア ブート許可ヒューズ (0=プログラム、1=非プログラム)

HWBE	リセット後実行開始アドレス (リセット ベクタ)
0	PE2/ $\overline{\text{HWB}}$ ピンはリセット後の強制ブート ロータ実行に使えません。
1	PE2/ $\overline{\text{HWB}}$ ピンはリセット後の強制ブート ロータ実行のために、リセットの間に使われます。



27.5.4. SPMCSR – SPM命令制御/状態レジスタ (Store Program Memory Control and Status Register)

このレジスタはブートローダー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	7	6	5	4	3	2	1	0	
\$37 (\$57)	SPMIE	RWWSB	SIGRD	RWWSRE	BLBSET	PGWRT	PGERS	SPMEN	SPMCSR
Read/Write	R/W	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 – 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(I)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 – SIGRD : 識票列読み出し (Signature Row Read)

このビットがSPMENと同時に1を書かれると、次の3クロック周期内のLPM命令は識票列からバイトを転送先レジスタに読みます。詳細については222頁の「**ソフトウェアからの識票列読み出し**」をご覧ください。SIGRDとSPMENが設定(1)された後の4クロック周期内のSPM命令は無効です。この操作は将来の使用に対して予約されており、使われるべきではありません。

● 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222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 – PGERS : 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。PGERS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、PGERS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6ビットに100001, 010001, 001001, 000101, 000011, 000001以外のどんな組み合わせを書いても無効です。

注: 常に1つのSPM命令だけが活動すべきです。

27.6. 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

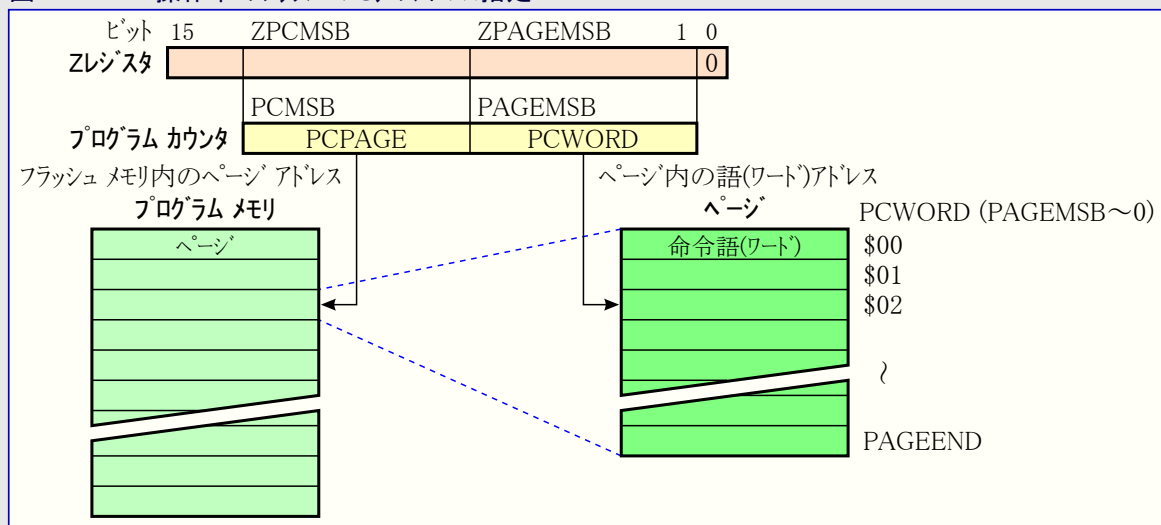
Zポインタ(レジスタ)はSPM命令でのアドレス指定に使われます。

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、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27-4.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、Zポインタは他の操作に使えます。

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Zレジスタの内容は無視され、無効です。LPM命令もアドレスを格納するのに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27-4. 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225頁の表27-10.で一覧されます。
PCPAGEとPCWORDは230頁の表28-11.で一覧されます。(訳注:共通性から追加)

27.7. 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224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。

27.7.1. 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Zポインタ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ZポインタのPCPAGEに書かれなければなりません。この操作中、Zポインタの他のビットは無視されます。

- 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。

27.7.2. 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ポインタにアドレス、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ポインタの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。

27.7.3. ページ書き込みの実行

ページ書き込みを行うにはZポインタ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ZポインタのPCPAGEに書かれなければなりません。この操作中、Zポインタの他のビットは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。

27.7.4. 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36ページの「割り込み」で記述されます。

27.7.5. 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。

27.7.6. 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36ページの「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224ページの「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。

27.7.7. 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27-2と表27-3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ポインタ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)\$0001でZポインタ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。

27.7.8. 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プログラム許可(EEPE)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。

27.7.9. 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めます。施錠ビットを読むにはZポインタに\$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ポインタに\$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227頁の表28-5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ポインタに\$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227頁の表28-4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ポインタに\$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227頁の表28-3を参照してください。

ビット	7	6	5	4	3	2	1	0
Rd	-	-	-	-	EFB3	EFB2	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。

27.7.10. ソフトウェアからの識票列読み出し

ソフトウェアから識票列を読むには右表で与えられる識票バイトアドレスをZポインタに設定してSPMCSRのSPMENと識票列読み出し(SIGRD)ビットを設定(1)してください。SPMENとSIGRD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、識票バイト値が転送先レジスタに格納されます。SPMENとSIGRDビットは識票バイ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場合、自動的に解除(0)されます。SPMENとSIGRDビットが解除(0)されると、LPMはAVR命令一式説明で記述されるように動作します。

表27-6. 識票列アドレス一覧

識票バイト	Zポインタアドレス
デバイス識票バイト 1	\$0000
デバイス識票バイト 2	\$0002
デバイス識票バイト 3	\$0004
RC発振器校正値	\$0001

注: 他の全てのアドレスは将来の使用に
対して予約されています。

27.7.11. フラッシュ メリ データ化けの防止

低VCCの期間中、CPUとフラッシュ メ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 メリへの通常の手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 ロータ更新が必要ない場合、どんなブート ロータ ソフトウェア更新をも防ぐためにブート ロータ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 メリを予期せぬ書き込みから効果的に保護します。

27.7.12. SPM命令使用時のフラッシュ メ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 メリ アクセス時間に使われます。この表はCPUからのフラッシュ メリ 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27-7. SPM命令によるフラッシュ メ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

27.7.13. 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
	LDI SPMC, (1<<RWWSRE) + (1<<SPMEN)	; [RWW領域読み出し再許可]
	CALL SPMJ	; RWW領域読み出し許可SPMCSR値を取得
		; RWW領域読み出し許可
	LDI CNTL, LOW(PGSZB)	; [RAMからフラッシュページ一時緩衝部へ転送]
	LDI CNTH, HIGH(PGSZB)	; バイト計数器を初期化
WLP:	LD R0, Y+	; (削除)
	LD R1, Y+	; RAM上の下位データを取得(ポインタ進行)
	LDI SPMC, (1<<SPMEN)	; RAM上の上位データを取得(ポインタ進行)
	CALL SPMJ	; ページ一時緩衝部書き込みSPMCSR値を取得
	ADIW ZH: ZL, 2	; 対応語(ワード)データをページ一時緩衝部に設定
	SBIW CNTH: CNTL, 2	; ページ一時緩衝部ポインタ進行
	BRNE WLP	; 計数器を減数 (SUBI)
		; 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:	LPM 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	; カウンタを減数 (SUBI)
	BRNE RLP	; 指定バイト数分継続
		; [RWW領域へ復帰]
RTN:	IN 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:	IN 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
;		
	OUT SPMCSR, 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

27.7.14. ATmega16U4/32U4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以下の表を通して与えられます。

表27-8. 応用領域とブートローダ領域の分割設定 (上段:ATmega16U4, 下段:ATmega32U4)

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	256	4	\$0000~\$1EFF \$0000~\$3EFF	\$1F00~\$1FFF \$3F00~\$3FFF	\$1F00 \$3F00
1	0	512	8	\$0000~\$1DFF \$0000~\$3DFF	\$1E00~\$1FFF \$3E00~\$3FFF	\$1E00 \$3E00
0	1	1024	16	\$0000~\$1BFF \$0000~\$3BFF	\$1C00~\$1FFF \$3C00~\$3FFF	\$1C00 \$3C00
0	0	2048	32	\$0000~\$17FF \$0000~\$37FF	\$1800~\$1FFF \$3800~\$3FFF	\$1800 \$3800

注: 各種BOOTSZヒューズ設定は図27-2.で示されます。

表27-9. RWW領域とNRWW領域の範囲 (上段:ATmega16U4, 下段:ATmega32U4)

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	96	\$0000~\$17FF
	224	\$0000~\$37FF
書き込み中読み出し不能(NRWW)領域	32	\$1800~\$1FFF \$3800~\$3FFF

注: これら2つの領域についての詳細に関しては216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27-10. 図27-4.で使った各変数説明とZポイントの配置 (上段:ATmega16U4, 下段:ATmega32U4)

変数名	対応値		意味 (注1)
	PC	Zポイント (注2)	
PCMSB	PC12 PC13		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13ビット、PC12~0) (プログラムカウンタは14ビット、PC13~0)
PAGEMS	PC5		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。 (ページ内の64語(ワード)には6ビット PC5~0が必要)
ZPCMSB		Z13 Z14	PCMSBに配置される(対応する)Zレジスタ内のビット。 (Z0が使われないため、ZPCMSB=PCMSB+1)
ZPAGEMS		Z6	PAGEMSに配置される(対応する)Zレジスタ内のビット。 (Z0が使われないため、ZPAGEMS=PAGEMS+1)
PCPAGE	PC12~6 PC13~6	Z13~7 Z14~7	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	PC5~0	Z6~1	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 (ページ書き込み操作中は0でなければなりません。)

注1: Z0 : 全てのSPM命令に対して0であるべきで、LPM命令に対するバイト選択です。

注2: 自己プログラミング中のZポイントの使用についての詳細に関しては220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。

28. メモリプログラミング

28.1. プログラム メモリとデータ メモリ用施錠ビット

本デバイスは非プログラム(1)のままか、表28-2.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28-1. 施錠ビット バイトの内容

名称	ビット番号	意味	既定値 (注)	
			ATmega16U4/32U4	ATmega16U4RC/32U4RC
—	7	ブート ロータ領域に対する保護用ブート施錠ビット	1 (非プログラム)	
—	6		1 (非プログラム)	
BLB12	5		1 (非プログラム)	
BLB11	4		0 (プログラム)	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	0 (プログラム)	1 (非プログラム)
LB1	0		0 (プログラム)	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28-2. 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	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、ブートロータ領域での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2)
4	0	1	ブートロータ領域での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2)
LB1 種別	BLB12	BLB11	フラッシュメモリのブートロータプログラム領域に対する保護
1	1	1	LPM, SPM命令がブートロータ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命令はブートロータ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タ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タ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LPM命令によるブートロータ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タ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、ブートロータ領域での実行時に割り込みが禁止されます。

28.2. ヒューズ ビット

本デバイスには3つのヒューズ バイトがあります。表28-3.~5.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28-3. 拡張ヒューズ バイト一覧				
名称	ビット	意味	既定値	
			ATmega16U4/32U4	ATmega16U4RC/32U4RC
—	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)	
HWBE	3	ハードウェア ブート許可(HWBピン有効)。	0 (プログラム) 許可	1 (非プログラム) 不許可
BODLEVEL2	2		0 (プログラム)	
BODLEVEL1	1	低電圧検出(BOD)リセットの制御と検出電圧選択。 (注1)	1 (非プログラム)	
BODLEVEL0	0		1 (非プログラム)	

注1: BODLEVELヒューズの符号化については31頁の表8-1.をご覧ください。

表28-4. 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(OCDF)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	
WDTON (注3)	4	ウォッチドッグ タイム常時有効。	1 (非プログラム) WDTはWDTCSRで許可	
EESAVE	3	チップ消去からEEPROM内容を保護。	1 (非プログラム) EEPROM内容保護なし	
BOOTSZ1	2	ブート ロード容量選択。(表27-8.参照) (注2)	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 ロード領域)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: BOOTSZ1,0既定値は最大ブート ロード容量になります。225頁の表27-8.をご覧ください。
注3: 詳細については35頁の「WDTCSR – ウォッチドッグ タイム制御レジスタ」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28-5. ヒューズ 下位バイト一覧				
名称	ビット	意味	既定値	
			ATmega16U4/32U4	ATmega16U4RC/32U4RC
CKDIV8 (注3)	7	システム クロック 8分周選択。	0 (プログラム) 8分周	
CKOUT (注2)	6	システム クロック出力許可。	1 (非プログラム) 不許可	
SUT1	5	起動時間選択。	1 (プログラム)	
SUT0	4		0 (非プログラム)	
CKSEL3	3	クロック種別選択。 (注1)	1 (非プログラム)	0 (プログラム)
CKSEL2	2		1 (非プログラム)	0 (プログラム)
CKSEL1	1		1 (非プログラム)	1 (非プログラム)
CKSEL0	0		0 (プログラム)	0 (プログラム)

注1: CKSEL3~0の既定設定はATmega16U4とATmega32U4に関してクリスタル用低電力発振器、ATmega16U4RCとATmega32U4RCに関して内蔵RC発振器になります。詳細については18頁の表6-1.をご覧ください。
注2: CKOUTはポートC7に出力することをシステム クロックに許します。詳細については22頁の「クロック出力緩衝部」をご覧ください。
注3: 詳細については22頁の「システム クロック前置分周器」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。

28.2.1. 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ず。

28.3. 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。

ATmega16U4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 : \$94 フラッシュ メモリ容量16Kバイトを示します。
- ③ \$002 : \$88 ②値\$94と合せ、ATmega16U4を示します。

ATmega32U4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 : \$95 フラッシュ メモリ容量32Kバイトを示します。
- ③ \$002 : \$87 ②値\$95と合せ、ATmega32U4を示します。

28.4. 校正バイト

本デバイスには内蔵RC発振器用に1バイトの校正値があります。このバイトは識票アドレス空間でアドレス\$0000の上位バイトにあります。リセット中、[校正付き内蔵RC発振器](#)の正しい周波数を保証するために、このバイトが[発振校正\(OSCCAL\)レジスタ](#)へ自動的に書かれます。

28.5. 並列プログラミング

本項は本デバイス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。

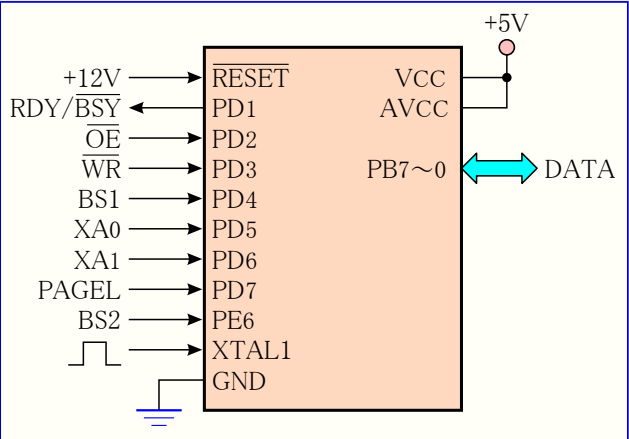
28.5.1. 信号名

本章で本デバイス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28-1**と**表28-6**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28-9**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28-10**で示されます。

図28-1. 並列プログラミング構成図



注: 未使用ピンは浮き状態のままにすべきです。

表28-6. 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	バイト選択1 (表28-7参照)
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	PE6	入力	バイト選択2 (表28-7参照)
DATA	PB7~0	入出力	双方向データバス (OE=Low時出力)

表28-7. BS2とBS1の符号化(機能)

BS2	1	1	0	0
BS1	1	0	1	0
フラッシュ/EEPROMアドレス設定	(予約)	拡張上位バイト	上位バイト	下位バイト
フラッシュメモリデータ設定/読み出し	(予約)	(予約)	上位バイト	下位バイト
ヒューズ書き込み	(予約)	拡張ヒューズバイト	ヒューズ上位バイト	ヒューズ下位バイト
ヒューズ/施錠ビット読み出し	ヒューズ上位バイト	拡張ヒューズバイト	施錠ビット	ヒューズ下位バイト

表28-8. 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28-9. 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BS1,2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28-10. 指令バイトのビット符号化

指令バイト	指令の機能	指令バイト	指令の機能
\$80 (1000 0000)	チップ消去	\$08 (0000 1000)	識票バイト、校正バイト読み出し
\$40 (0100 0000)	ヒューズビット書き込み	\$04 (0000 0100)	ヒューズビット、施錠ビット読み出し
\$20 (0010 0000)	施錠ビット書き込み	\$02 (0000 0010)	フラッシュメモリ読み出し
\$10 (0001 0000)	フラッシュメモリ書き込み	\$03 (0000 0011)	EEPROM読み出し
\$11 (0001 0001)	EEPROM書き込み		

表28-11. フラッシュメモリのページ数とページの語数

デバイス	全容量	ページ容量	PCWORD	ページ数	PCPAGE	PCMSB
ATmega16U4	8K語 (16Kバイト)	64語	PC5~0	128	PC12~6	12
ATmega32U4	16K語 (32Kバイト)			256	PC13~6	13

表28-12. EEPROMメモリのページ数とページの語数

デバイス	全容量	ページ容量	PCWORD	ページ数	PCPAGE	EEAMSB
ATmega16U4	512バイト	4バイト	EEA1~0	128	EEA8~2	8
ATmega32U4	1Kバイト			256	EEA9~2	9

28.6. 並列プログラミング手順

28.6.1. 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を印加します。
- ② 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。
- ③ 229頁の表28-8.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 RESETに11.5~12.5Vを印加します。+12Vが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。
- ⑤ 新規指令送出前に少なくとも50μs間待ちます。

28.6.2. 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。

28.6.3. 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 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/BSYがLow(0)になります。
- ⑥ 次の指令を設定する前にRDY/BSYがHigh(1)になるまで待ちます。

(訳注) 原書に於いて、フラッシュメモリプログラミング時のアドレス指定が拡張アドレスまで指定するようになっていました。然しながらプログラミング時のアドレス指定は語(ワード)単位で行うため、64K語(128Kバイト)までの場合は16ビット幅以内なので、基本的に拡張アドレス指定は必要ない筈です。これは128K語(256Kバイト)用のデバイス記述を複写し、修正を忘れたものと思われます。けれども、将来の展開を予想し、64K語(128Kバイト)を越えるデバイス用の回路を採用したため、拡張アドレスの明示的な指定が必要になった、と考えることもできます。

本書では原則的に原書に従いますが、独自追加部分は拡張アドレス指定なし、として記述しました。

28.6.4. フラッシュ メモリ書き込み (図28-3.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(230頁の表28-11.参照)。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 (アドレス ビット7~0) 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 BS2をLow(0)、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を繰り返し

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。これは次頁の図28-2.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 (アドレス ビット15~8) 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 BS2をLow(0)、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~\$1F/\$3F)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. アドレス上位拡張バイト (アドレス ビット23~16) 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 BS2をHigh(1)、BS1をLow(0)に設定します。これは上位拡張アドレス(バイト)を選びます。
- ③ DATAにアドレス上位バイト(\$00)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拡張バイトを設定します。

I. ページ書き込み

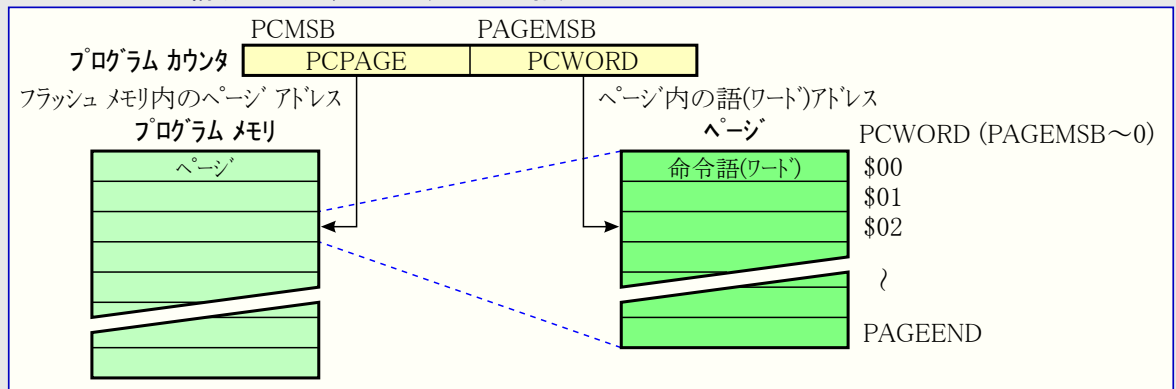
- ① BS1をLow(0)に設定します。
- ② $\overline{\text{WR}}$ に負パルスを与えます。これはデータのページ全体の書き込みを開始します。RDY/ $\overline{\text{BSY}}$ がLow(0)になります。
- ③ RDY/ $\overline{\text{BSY}}$ がHigh(1)になるまで待ちます。

J. 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~Iを繰り返し

K. ページ書き込み終了

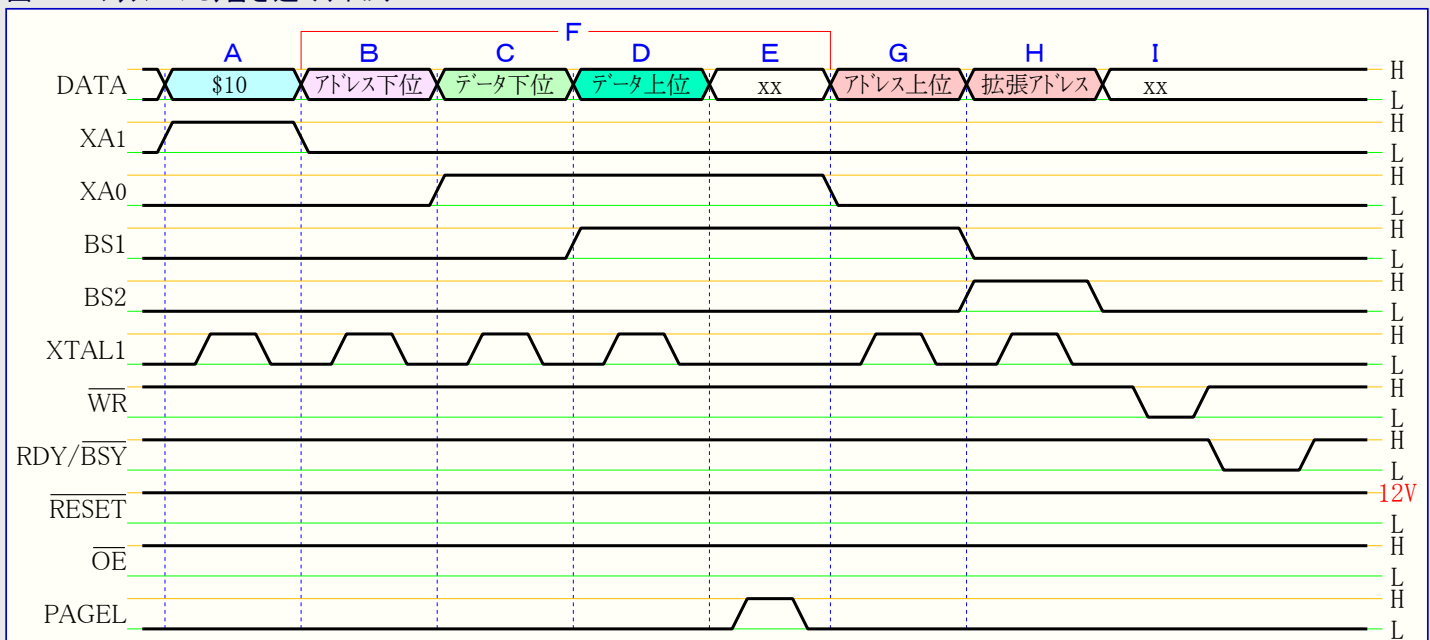
- ① XA1をHigh(1)、XA0をLow(0)に設定します。これは指令設定を許可します。
- ② DATAを\$00(0000 0000)にします。これは無操作指令です。
- ③ XTAL1に正パルスを与えます。これは無操作指令を設定し、内部書き込み信号がリセットされます。

図28-2. ページで構成されたフラッシュメモリのアドレス指定



注: PCPAGEとPCWORDは230頁の表28-11.で一覧されます。

図28-3. フラッシュメモリ書き込みタイミング



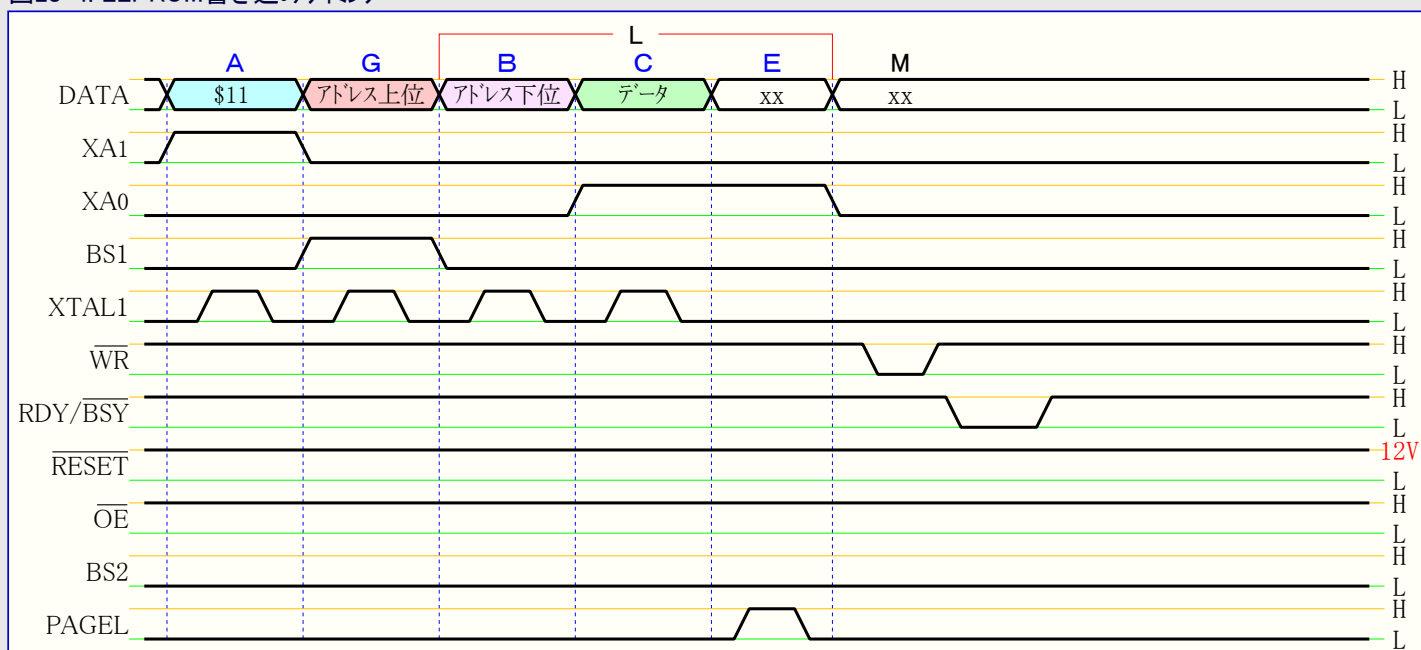
注: xx値は無関係です。A~Iは前記プログラミングを参照してください。

28.6.5. EEPROM書き込み

EEPROMはページで構成されます(230頁の表28-12.参照)。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231頁の「フラッシュメモリの書き込み」を参照。図28-4.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~\$01/\$03)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)
- L. 緩衝部全体が満たされるまで3~5を繰り返します。
- M. EEPROMページ書き込み
 - ① BS2をLow(0)、BS1をLow(0)に設定します。
 - ② \overline{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/ \overline{BSY} がLow(0)になります。
 - ③ 次のページを書く前に、RDY/ \overline{BSY} がHigh(1)になるまで待ちます。

図28-4. EEPROM書き込みタイミング



28.6.6. 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231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. アドレス拡張バイト(\$00)を設定します。(「フラッシュメモリ書き込み」のHを参照)
3. アドレス上位バイト(\$00~\$1F/\$3F)を設定します。(「フラッシュメモリ書き込み」のGを参照)
4. 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)
5. BS1をLow(0)、 \overline{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。
6. 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。
7. \overline{OE} をHigh(1)に設定します。DATAはHi-Zになります。

28.6.7. EEPROM読み出し

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231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。(「フラッシュメモリの書き込み」のAを参照)
2. アドレス上位バイト(\$00~\$01/\$03)を設定します。(「フラッシュメモリの書き込み」のGを参照)
3. アドレス下位バイト(\$00~\$FF)を設定します。(「フラッシュメモリの書き込み」のBを参照)
4. BS1をLow(0)、 \overline{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. \overline{OE} をHigh(1)に設定します。DATAはHi-Zになります。

28.6.8. 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。)

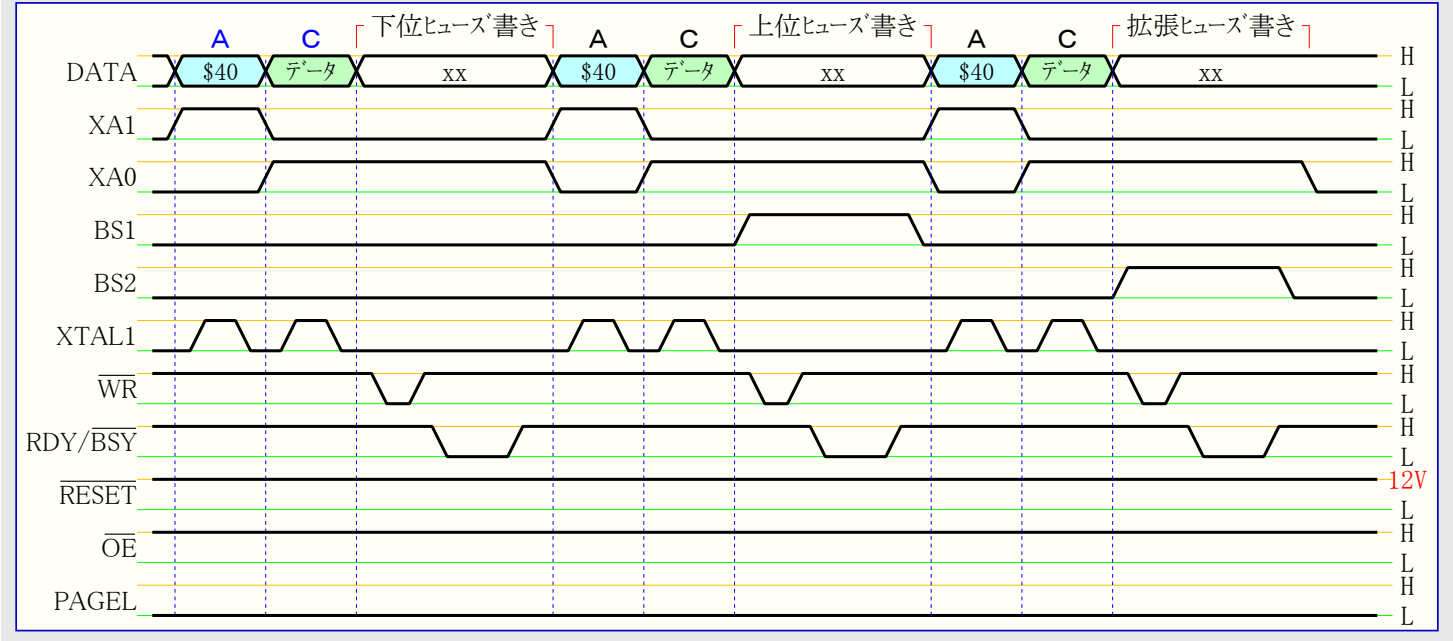
各ヒューズ バイトの書き込み方法は次のとおりです。(指令とデータ設定の詳細については231頁の「フラッシュ メモリの書き込み」を参照)

- 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. \overline{WR} に負パルスを与え、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28-5. ヒューズ書き込みタイミング



28.6.9. 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231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 メモリの書き込み」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。LB保護種別3が設定(LB1とLB2がプログラム(0))されると、どの外部的なプログラミング動作種別によってもブート施錠ビットはプログラミングできません。(「フラッシュ メモリの書き込み」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。

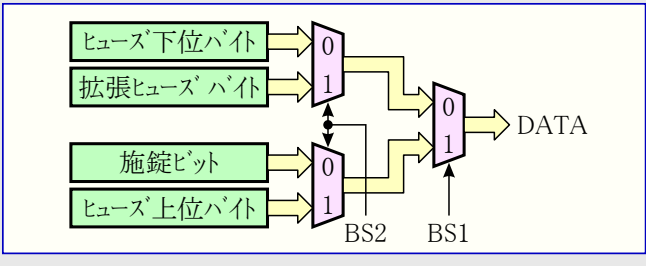
施錠ビットはチップ消去の実行によってのみ解除(1)できます。

28.6.10. 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231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 メモリの書き込み」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28-6. ヒューズ、施錠ビット読み出し中のBS1、BS2との関係



28.6.11. 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231頁の「フラッシュメモリの書き込み」を参照)

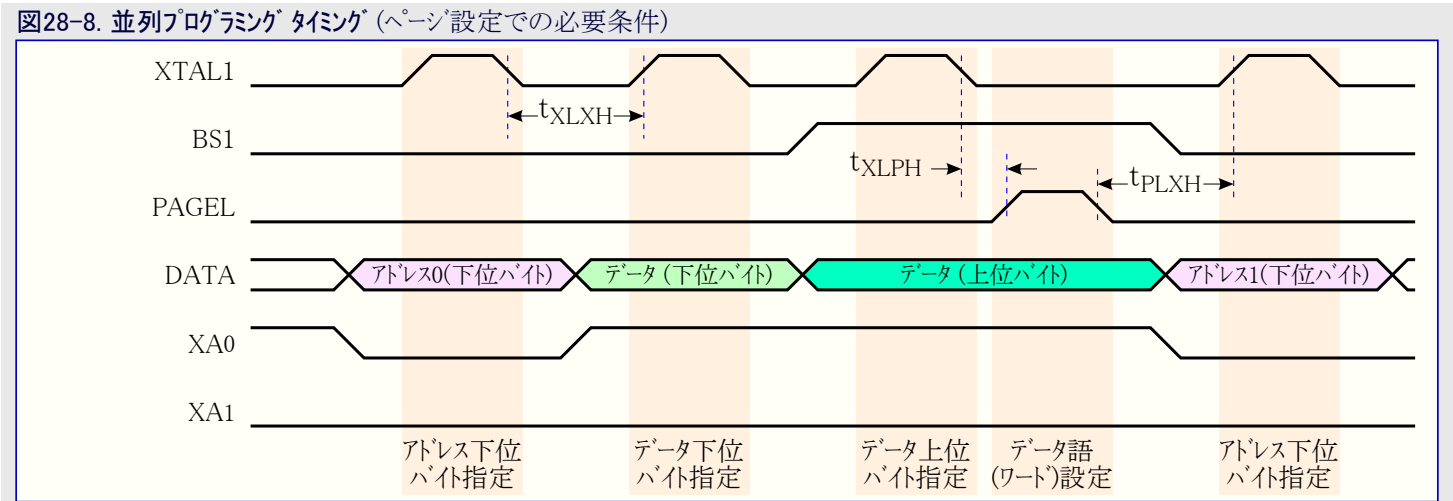
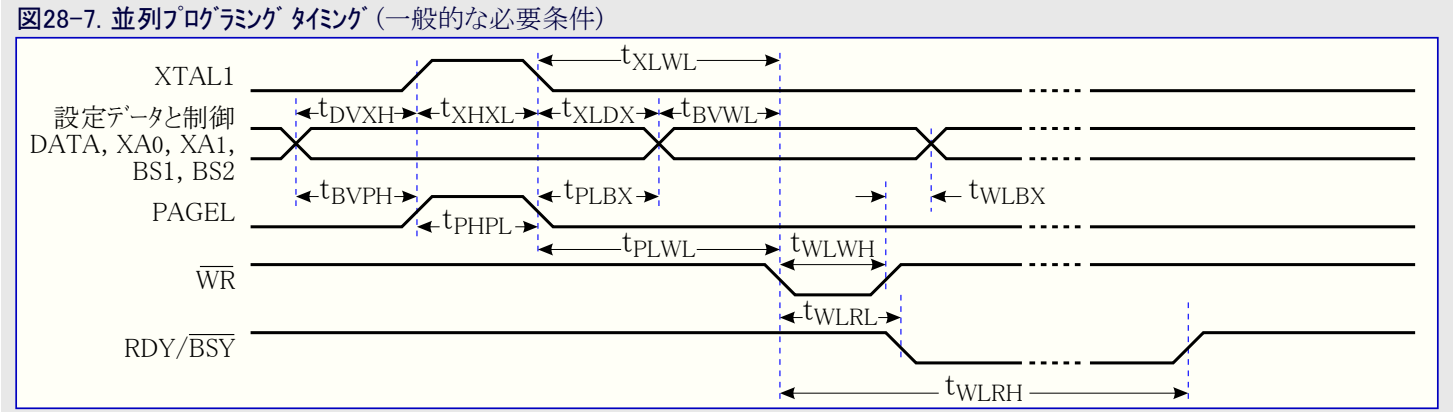
1. 識票バイト読み出し指令\$08(0000 1000)を設定します。(「フラッシュメモリの書き込み」のAを参照)
2. アドレス下位バイト(\$00～\$02)を設定します。(「フラッシュメモリの書き込み」のBを参照)
3. BS1をLow(0)、 $\overline{\text{OE}}$ をLow(0)に設定します。選んだ識票バイトが直ぐにDATAで読めます。
4. $\overline{\text{OE}}$ をHigh(1)に設定します。DATAはHi-Zになります。

28.6.12. 校正バイト読み出し

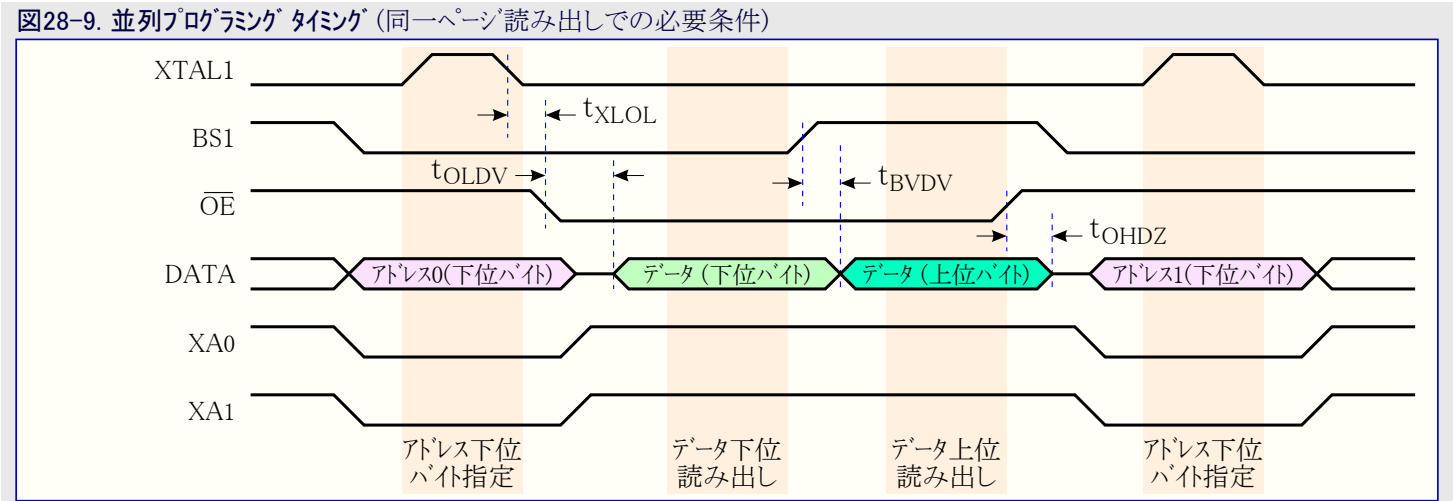
校正バイトの読み出し方法は次のとおりです。(指令とアドレス設定の詳細については231頁の「フラッシュメモリの書き込み」を参照)

1. 校正バイト読み出し指令\$08(0000 1000)を設定します。(「フラッシュメモリの書き込み」のAを参照)
2. アドレス下位バイトに\$00を設定します。(「フラッシュメモリの書き込み」のBを参照)
3. BS1をHigh(1)、 $\overline{\text{OE}}$ 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. $\overline{\text{OE}}$ をHigh(1)に設定します。DATAはHi-Zになります。

28.6.13. 並列プログラミング特性



注: 図28-7.で示されるタイミング必要条件(即ち、 t_{DVXH} 、 t_{XHL} 、 t_{XLDX})は設定操作にも適用されます。



注: 図28-7.で示されるタイミング必要条件(即ち、 t_{DVXH} 、 t_{XHL} 、 t_{XLDX})は読み出し操作にも適用されます。

表28-13. 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
V _{PP}	プログラミング許可電圧	11.5		12.5	V
I _{PP}	プログラミング許可電流			250	μA
t _{DVXH}	XTAL1 ↑ に対するデータと制御の準備時間	67			
t _{XLXH}	XTAL1 ↓ から次XTAL1 ↑ までの待機時間	200			
t _{XHXL}	XTAL1 Highパルス幅	150			
t _{XLDX}	XTAL1パルス ↓ 後のデータと制御の保持時間	67			
t _{XLWL}	XTAL1パルス ↓ 後の \overline{WR} ↓ 待機時間	0			
t _{XLPH}	XTAL1パルス ↓ 後のPAGELパルス ↑ 待機時間	0			
t _{PLXH}	PAGELパルス ↓ 後のXTAL1パルス ↑ 待機時間	150			ns
t _{BVPH}	PAGELパルス ↑ に対するBS1準備時間	67			
t _{PHPL}	PAGEL Highパルス幅	150			
t _{PLBX}	PAGELパルス ↓ 後のBS1保持時間	67			
t _{WLBX}	\overline{WR} パルス ↓ 後のBS1,BS2保持時間	67			
t _{PLWL}	PAGELパルス ↓ 後の \overline{WR} パルス ↓ 待機時間	67			
t _{BVWL}	\overline{WR} パルス ↓ に対するBS1準備時間	67			
t _{WLWH}	\overline{WR} Lowパルス幅	150			
t _{WLRL}	\overline{WR} パルス ↓ 後のRDY/ \overline{BSY} ↓ 遅延時間	0		1	μs
t _{WLRH}	書き込み時間 (\overline{WR} ↓ からRDY/ \overline{BSY} ↑) (注1)	3.7		4.5	ms
t _{WLRH_CE}	チップ消去時間 (\overline{WR} ↓ からRDY/ \overline{BSY} ↑) (注2)	7.5		9	
t _{XLCL}	XTAL1パルス ↓ 後の \overline{OE} ↓ 待機時間	0			
t _{BVDV}	BS1有効からのDATA遅延時間	0		250	ns
t _{OLDV}	\overline{OE} ↓ 後のDATA出力遅延時間			250	
t _{OHDZ}	\overline{OE} ↑ 後のDATAフローティング遅延時間			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。

28.7. 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、PDI入力、PDO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得るのに先立って**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28-14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。

28.7.1. 直列プログラミング用ピン配置

表28-14. 直列プログラミング用ピン配置

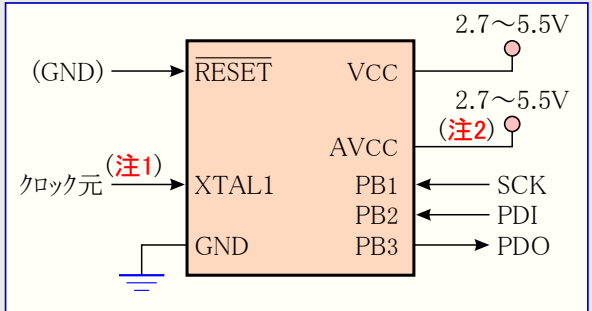
信号名	ピン名	入出力	機能
SCK	PB1	入力	直列クロック
PDI/MOSI	PB2	入力	直列データ入力
PDO/MISO	PB3	出力	直列データ出力

EEPROMをプログラミングする時に自動消去期間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す。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。

$f_{CK} < 12\text{MHz}$: Low区間 > 2 CPUクロック周期	$f_{CK} < 12\text{MHz}$: High区間 > 2 CPUクロック周期
$f_{CK} \geq 12\text{MHz}$: Low区間 > 3 CPUクロック周期	$f_{CK} \geq 12\text{MHz}$: High区間 > 3 CPUクロック周期

図28-10. 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。

28.8. 直列プログラミング手順 (訳注:本項の一部を矛盾回避のため修正しました。)

本デバイスに直列データを書く時にデータはSCKの上昇端で行われ、本デバイス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28-11をご覧ください。

直列プログラミング動作での本デバイス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28-16を参照)。

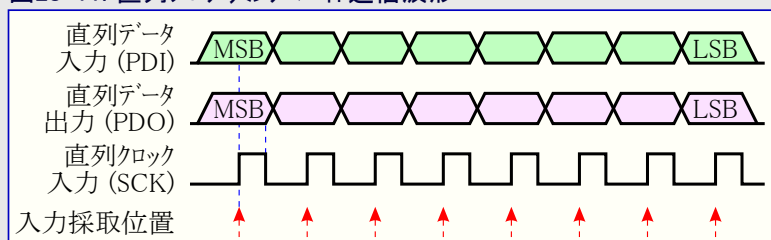
- 電源投入手順:
RESETとSCKがLow(0)に設定されている間にVCCとGND間へ電源を印加します。いくつかのシステムで電源投入中、SCKがLowに保持されるのを書き込み器が保証できません。この場合、SCKがLow(0)に設定されてしまった後、RESETは最低2 CPUクロック周期幅の正パルスを与えられなければなりません。
- 最低20ms待ち、PDIピンに**プログラミング許可命令**を送ることによって直列プログラミングを許可してください。
- 通信が同期を外していると、直列プログラミング命令は動作しません。同期していると、プログラミング許可命令の第3バイト送出時に第2バイト(\$53)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。
- 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230ページの表28-11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6+1ビットとデータを提供することによって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でのデータ上位バイト適用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7/8ビット(PC12/13~6)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。この命令が行なわれる前に拡張アドレスバイト設定命令がアドレスのMSB定義に使われていることに注意してください。拡張アドレスバイトは本命令が再実行されるまで保存されます。即ち、本命令は初回ページと64K語(ワード)境界を過ぎる時にだけ実行される必要があります。ポーリング(BSY/RDY)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28-15参照)待たなければなりません。フラッシュ書き込み操作完了前の(ポーリング以外の)直列プログラミングインターフェースでのアクセスは不正な書き込み結果になり得ます。
- EEPROMはページ単位かバイト単位のどちらかでプログラミングできます。
バイト単位: EEPROMは適切な**EEPROM書き込み命令**と共にアドレスとデータを提供することによって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。ポーリング(BSY/RDY)が使われない場合、次のバイトを行う前に最低tWD_EEPROM(表28-15参照)待たなければなりません。
ページ単位: EEPROMの1ページは**EEPROMページ設定命令**と共にアドレスの下位2ビットとデータを提供することによって1バイトずつ設定されます。EEPROMページはアドレスの上位7/8ビットを含む**EEPROMページ書き込み命令**によって(EEPROMに)格納されます。EEPROMページアクセス使用時、EEPROMページ設定命令で設定したバイト位置だけが変更されます。残りの位置は無変化で留まります。ポーリング(BSY/RDY)が使われない場合、次ページ(表28-12参照)を行う前に最低tWD_EEPROM(表28-15参照)待たなければなりません。チップ消去されたデバイスでの\$FFデータは書かれる必要がありません。
- どのメモリ位置も選んだアドレスの内容を直列出力PDOに読み戻す**読み出し命令**を使うことで照合できます。フラッシュメモリ読み出し時、フラッシュメモリ読み出し命令内に含まれない上位アドレスバイト定義に拡張アドレスバイト設定命令を使ってください。拡張アドレスバイトはこの命令が再実行されるまで保存されます。換言すると、本命令は初回ページと64K語(ワード)境界を過ぎる時にだけ実行される必要があります。

7. プログラミング作業終了時、 $\overline{\text{RESET}}$ は通常動作を開始するため、High(1)に設定できます。
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。

表28-15. ヒューズ、フラッシュ、EEPROM書き込み待機時間

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	4.5ms	フラッシュメモリ書き込み
tWD_EEPROM	9.0ms	EEPROM書き込み
tWD_ERASE	9.0ms	チップ消去

図28-11. 直列プログラミングバイト通信波形



28.8.1. 直列プログラミング命令一式 (訳注: 命令一覧は新記述形式に変更しました。)

表28-16.と次頁の図28-12.は命令一式を記述します。

表28-16. 直列プログラミング命令一式

命令	命令形式				備考
	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	\$AC	\$53	\$00	\$00	
チップ消去	\$AC	\$80	\$00	\$00	
多忙/準備可検査	\$F0	\$00	\$00	状態値	最下位ビットが多忙フラグ。
設定系命令					
拡張アドレス設定 (注1)	\$4D	\$00	拡張アドレス	\$00	
フラッシュ ページ内上位バイト設定	\$48	アドレス上位	アドレス下位	上位バイト	
フラッシュ ページ内下位バイト設定	\$40	アドレス上位	アドレス下位	下位バイト	
EEPROMページ内バイト設定	\$C1	\$00	ページ内位置	バイト	注: ページ内指示以外のビットは0。
読み出し命令					
フラッシュ メモリ上位バイト読み出し	\$28	アドレス上位	アドレス下位	上位バイト	
フラッシュ メモリ下位バイト読み出し	\$20	アドレス上位	アドレス下位	下位バイト	
EEPROM読み出し	\$A0	アドレス上位	アドレス下位	バイト	
施錠ビット読み出し	\$58	\$00	\$00	施錠ビット値	
識票バイト読み出し	\$30	\$00	アドレス	識票バイト	
ヒューズ下位読み出し	\$50	\$00	\$00	ヒューズ下位	
ヒューズ上位読み出し	\$58	\$08	\$00	ヒューズ上位	
拡張ヒューズ読み出し	\$50	\$08	\$00	拡張ヒューズ	
校正バイト読み出し	\$38	\$00	\$00	校正バイト	
書き込み命令					
フラッシュ ページ書き込み	\$4C	アドレス上位	アドレス下位	\$00	
EEPROMバイト書き込み	\$C0	アドレス上位	アドレス下位	バイト	
EEPROMページ書き込み	\$C2	アドレス上位	アドレス下位	\$00	注: ページ指示以外のアドレスビットは0。
施錠ビット書き込み	\$AC	\$E0	\$00	施錠ビット値	
ヒューズ下位書き込み	\$AC	\$A0	\$00	ヒューズ下位	
ヒューズ上位書き込み	\$AC	\$A8	\$00	ヒューズ上位	
拡張ヒューズ書き込み	\$AC	\$A4	\$00	拡張ヒューズ	

注1: 全命令が全デバイスで利用可能な訳ではありません。

注: 施錠ビットとヒューズ値はプログラムが0、非プログラムが1です。将来との互換性のため、未使用のヒューズと施錠ビットは非プログラム(1)にすべきです。

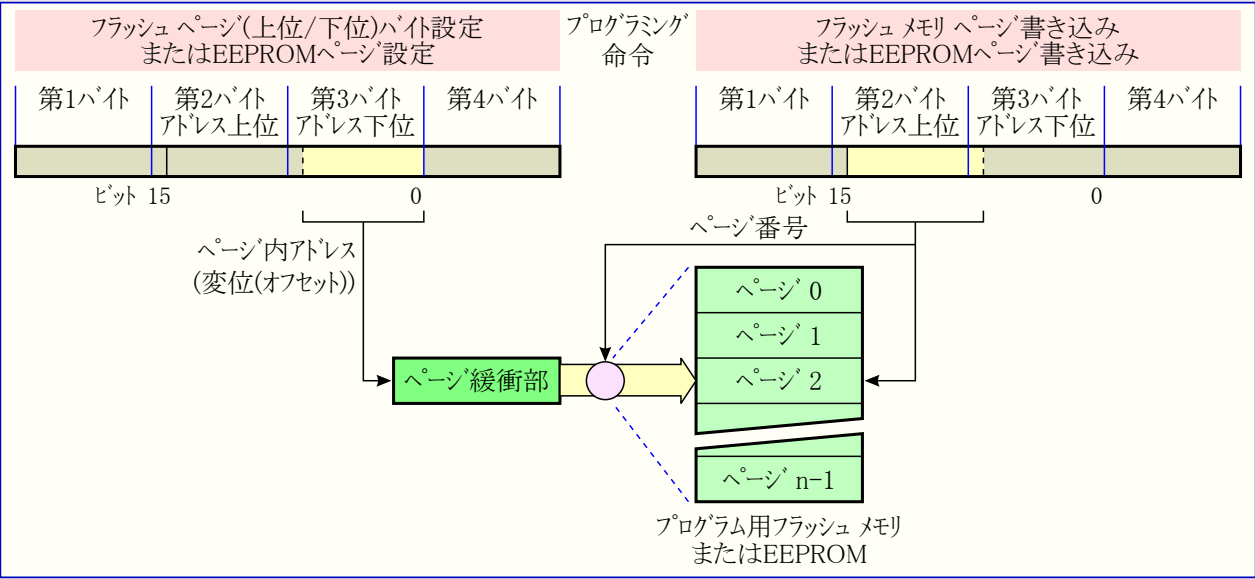
- ヒューズ、施錠ビット、識票バイト、校正バイト、ページ容量については対応項目を参照してください。
- プログラミングと書き込み器に関する応用記述については <http://www.atmel.com/avr> をご覧ください。
- 第4バイトの赤背景はホスト読み込み(デバイス出力)を示します。

多忙/準備可検査バイトデータ出力のLSBが1なら、プログラミング操作が未だ保留(動作中)です。次の命令が実行される前に本ビットが0に戻るまで待ってください。

同じページ内で、下位バイトデータは上位バイトデータに先行して格納されなければなりません。

データがページ緩衝部に格納された後にEEPROMページをプログラムしてください。図28-12をご覧ください。

図28-12. 直列プログラミング命令例



(訳補) フラッシュメモリ、EEPROM、ページ緩衝部内のアドレス(位置)指定に使うビットはメモリ容量とページ構成に依存します。ATmega16U4/32U4でのこれらの指定方法は次表で要約されます。

表B. アドレス(第2,3バイト)指定法

命令	第2バイト	第3バイト	備考
拡張アドレス設定	—	—	該当命令なし
フラッシュページ内バイト設定	0000 0000	00LL LLLL	ATmega16U4/32U4 : L=PC5~0
EEPROMページ内バイト設定	0000 0000	0000 00LL	ATmega16U4/32U4 : L=EEA1~0
フラッシュメモリ読み出し	000H HHHH 00HH HHHH	LLLL LLLL	ATmega16U4 : H=PC12~8, L=PC7~0 ATmega32U4 : H=PC13~8, L=PC7~0
EEPROM読み出し	0000 000H 0000 00HH	LLLL LLLL	ATmega16U4 : H=EEA8, L=EEA7~0 ATmega32U4 : H=EEA9~8, L=EEA7~0
フラッシュページ書き込み	000H HHHH 00HH HHHH	LL00 0000	ATmega16U4 : H=PC12~8, L=PC7~6 ATmega32U4 : H=PC13~8, L=PC7~6
EEPROMバイト書き込み	0000 000H 0000 00HH	LLLL LLLL	ATmega16U4 : H=EEA8, L=EEA7~0 ATmega32U4 : H=EEA9~8, L=EEA7~0
EEPROMページ書き込み	0000 000H 0000 00HH	LLLL LL00	ATmega16U4 : H=EEA8, L=EEA7~2 ATmega32U4 : H=EEA9~8, L=EEA7~2

28.8.2. 直列プログラミング特性

SPI部の特性については252頁の「SPIタイミング特性」を参照してください。

28.9. 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インターフェースが使われ得るにはJTAGENヒューズがプログラム(0)されなければなりません。この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制御レジスタ(MCUCR)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されなければなりません。

プログラミング中、TCK入力のクロック周波数はチップの最高周波数よりも低くなければなりません。システムクロック前置分周器はTCKクロック入力を十分に低い周波数へ分周するのには使えません。

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。

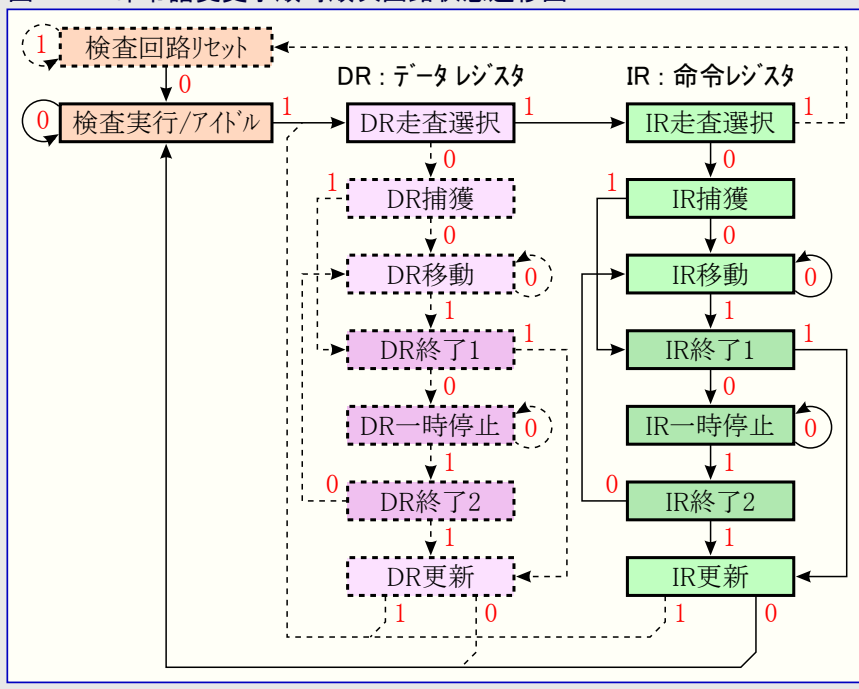
28.9.1. プログラミング特殊JTAG命令

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28-13.で示されます。

図28-13. 命令語変更手順時順次回路状態遷移図



28.9.2. 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。

28.9.3. 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。

28.9.4. 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)として選択されます。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

28.9.5. 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。8ビットフラッシュバイトデータレジスタがデータレジスタ(DR)として選ばれます。これは物理的にプログラミング命令レジスタの下位8ビットです。活性(有効)状態は以下です。

- DR移動 : フラッシュバイトデータレジスタがTCKクロック入力によって移動されます。
- DR更新 : フラッシュバイトデータレジスタの内容が一時レジスタに複写されます。11 TCK周期で開始される書き込み(設定)手順が一時レジスタの内容をフラッシュページ緩衝部に設定します。AVRはPROG_PAGELOAD命令移行後出会った最初のDR更新に対して下位バイトで始め、新規DR更新状態毎に上位/下位バイト書き込み間を自動的に切り替えます。プログラムカウンタは最初の書き込みバイトを除き、下位バイト書き込み前に予め増加(+1)されます。これは最初のデータがPROG_COMMANDSによって設定したアドレスに書かれ、ページ緩衝部の最終位置設定がプログラムカウンタを次ページへ増加しないことを保証します。

28.9.6. PROG_PAGEREAD (\$7)

JTAGポート経由でフラッシュメモリ内容を直接捕獲するためのAVR固有公開JTAG命令です。8ビットフラッシュバイトデータレジスタがデータレジスタ(DR)として選ばれます。これは物理的にプログラミング命令レジスタの下位8ビットです。活性(有効)状態は以下です。

- DR捕獲 : 選んだフラッシュメモリバイトの内容がフラッシュバイトデータレジスタに捕獲されます。AVRはPROG_PAGEREAD命令移行後に出会った最初のDR捕獲に対して下位バイトで始め、新規DR捕獲状態毎に上位/下位バイト読み込み間を自動的に切り替えます。プログラムカウンタは最初の読み込みバイトを含め、各上位バイト読み込み後に増加(+1)されます。これは最初のデータがPROG_COMMANDSによって設定した先頭アドレスから捕獲され、ページの最終位置読み込みがプログラムカウンタを次ページへ増加(進行)することを保証します。
- DR移動 : フラッシュバイトデータレジスタがTCKクロック入力によって移動されます。

28.9.7. データレジスタ

データレジスタ(DR)は241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
- フラッシュバイトデータ(Flash Data Byte)レジスタ

28.9.8. リセット (Reset) レジスタ

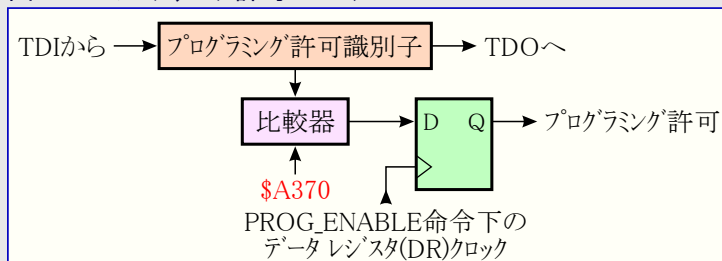
リセットレジスタはプログラミング中、デバイスをリセットするのに使われる検査データレジスタです。プログラミング動作への移行に先立ってデバイスをリセットするのに必要とされます。

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(18頁の「クロック元」参照)分リセットを維持します。211頁の図26-2.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。

28.9.9. 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28-14. プログラミング許可 レジスタ



28.9.10. 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28-17.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28-16.で図解されます。

図28-15. プログラミング命令 レジスタ



表28-17. JTAGプログラミング命令一式 (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		
フラッシュメモリ書き込み	1 ②	チップ消去完了検査	011 0011 1000 0000	xxx xxSx xxxx xxxx	(注2)	
	2 ①	フラッシュ書き込み移行	010 0011 0001 0000	xxx xxxx xxxx xxxx		
	2 ②	アドレス拡張上位バイト設定	000 0111 EEEE EEEE	xxx xxxx xxxx xxxx	(注10)	
	2 ③	アドレス上位バイト設定	000 0111 HHHH HHHH	xxx xxxx xxxx xxxx	(注10)	
	2 ④	アドレス下位バイト設定	000 0011 LLLL LLLL	xxx xxxx xxxx xxxx		
	2 ⑤	データ下位バイト設定	001 0011 WWWW WWWW	xxx xxxx xxxx xxxx		
	2 ⑥	データ上位バイト設定	001 0111 WWWW WWWW	xxx xxxx xxxx xxxx		
	2 ⑦	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 ⑧	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 ⑨	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	アドレス拡張上位バイト設定	000 0111 EEEE EEEE	xxx xxxx xxxx xxxx	(注10)	
	3 ③	アドレス上位バイト設定	000 0111 HHHH HHHH	xxx xxxx xxxx xxxx	(注10)	
	3 ④	アドレス下位バイト設定	000 0011 LLLL LLLL	xxx xxxx xxxx xxxx		
	3 ⑤	データ下位/上位バイト取得	011 0010 0000 0000	xxx xxxx xxxx xxxx		
EEPROM書き込み	4 ①	EEPROM書き込み移行	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	
			011 0111 0000 0000	xxx xxxx RRRR RRRR		
	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		

[次ページへ続く](#)

表28-17 (続き). JTAGプログラミング命令一式

命令		TDI入力	TDO出力	備考
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 WWW WWW	xxx xxxx xxxx xxxx	(注3) ビット配置は227頁の表28-3.参照
	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 xxS x xxxx xxxx	(注2)
	6 ⑤ データ下位バイト設定	001 0011 WWW WWW	xxx xxxx xxxx xxxx	(注3) ビット配置は227頁の表28-4.参照
	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 xxS x xxxx xxxx	(注2)
	6 ⑧ データ下位バイト設定	001 0011 WWW WWW	xxx xxxx xxxx xxxx	(注3) ビット配置は227頁の表28-5.参照
	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	
		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 xxS x xxxx xxxx	(注2)
施錠ビット書き込み	7 ① 施錠ビット書き込み移行	010 0011 0010 0000	xxx xxxx xxxx xxxx	
	7 ② 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226頁の表28-1.参照
	7 ③ 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	
		011 0011 0000 0000	xxx xxxx xxxx xxxx	
	7 ④ 施錠ビット書き込み完了検査	011 0011 0000 0000	xxx xxS x xxxx xxxx	(注2)
ヒューズ/施錠ビット読み出し	8 ① ヒューズ/施錠ビット読み出し移行	010 0011 0000 0100	xxx xxxx xxxx xxxx	
	8 ② 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227頁の表28-3.参照
		011 1011 0000 0000	xxx xxxx RRRR RRRR	
	8 ③ 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227頁の表28-4.参照
		011 1111 0000 0000	xxx xxxx RRRR RRRR	
	8 ④ 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227頁の表28-5.参照
		011 0011 0000 0000	xxx xxxx RRRR RRRR	
	8 ⑤ 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226頁の表28-1.参照
		011 0111 0000 0000	xxx xxxx xxRR RRRR	
	8 ⑥ 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	
		011 0010 0000 0000	xxx xxxx RRRR RRRR	
		011 0110 0000 0000	xxx xxxx RRRR RRRR	
		011 0111 0000 0000	xxx xxxx RRRR RRRR	
識票バイト	9 ① 識票バイト読み出し移行	010 0011 0000 1000	xxx xxxx xxxx xxxx	
	9 ② アドレスバイト設定	000 0011 LLLL LLLL	xxx xxxx xxxx xxxx	
	9 ③ 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	10 ① 校正バイト読み出し移行	010 0011 0000 1000	xxx xxxx xxxx xxxx	
	10 ② アドレスバイト設定	000 0011 LLLL LLLL	xxx xxxx xxxx xxxx	
	10 ③ 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	
	11 ① 無操作設定	010 0011 0000 0000	xxx xxxx xxxx xxxx	
		011 0011 0000 0000	xxx xxxx xxxx xxxx	

E = アドレス上位拡張バイトのビット

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28-11.参照)とEEAMSB(表28-12.参照)を越えるアドレスビットは無効です。

注11 : 全てのTDIとTDO手順は2進数で示されています。

28.9.11. フラッシュ バイト データ (Flash Data Byte) レジスタ

フラッシュ バイト データ レジスタはページ書き込み実行前のフラッシュ メモリ ページ緩衝部全体設定、またはフラッシュ メモリ内容の読み出しや照合の効率的な方法を提供します。順次回路はフラッシュ メモリへの制御信号を設定し、フラッシュ メモリからのストローブ信号を判断し、従って移動入出力されるためのデータ語だけが必要です。

実際のフラッシュ バイト データ レジスタは8ビットの走査チェーンと8ビットの一時レジスタから成ります。ページ設定中、**DR更新**状態は走査チェーンによる内容を一時レジスタへ複写し、11 TCK周期内で一時レジスタの内容をフラッシュ メモリ ページ緩衝部へ設定する書き込み(設定)手順を始めます。AVRは**PROG_PAGELoad**命令移行後に会った最初の**DR更新**に対して下位バイトで始め、新規**DR更新**毎に上位/下位バイト書き込み間を自動的に切り替えます。プログラム カウンタは最初の書き込みバイトを除き、下位バイト書き込み前に予め増加(+1)されます。これは**PROG_COMMANDS**によって設定したアドレスに最初のデータが書かれ、ページ緩衝部の最終位置設定がプログラム カウンタを次ページに増加しないことを保証します。

ページ読み込み中、選んだフラッシュ メモリのバイトの内容は**DR捕獲**状態中にフラッシュ バイト データ レジスタ内に捕獲されます。AVRは**PROG_PAGEREAD**命令移行後に会った最初の**DR捕獲**に対して下位バイトで始め、新規**DR捕獲**毎に上位/下位バイト読み込み間を自動的に切り替えます。プログラム カウンタは最初の読み込みバイトを含め、各上位バイト読み込み後に増加(+1)されます。これは最初のデータが**PROG_COMMANDS**によって設定した先頭アドレスから捕獲され、ページの最終位置読み込みがプログラム カウンタを次ページに増加(進行)することを保証します。

図28-16. データ語変更/読み出し手順時順次回路状態遷移図

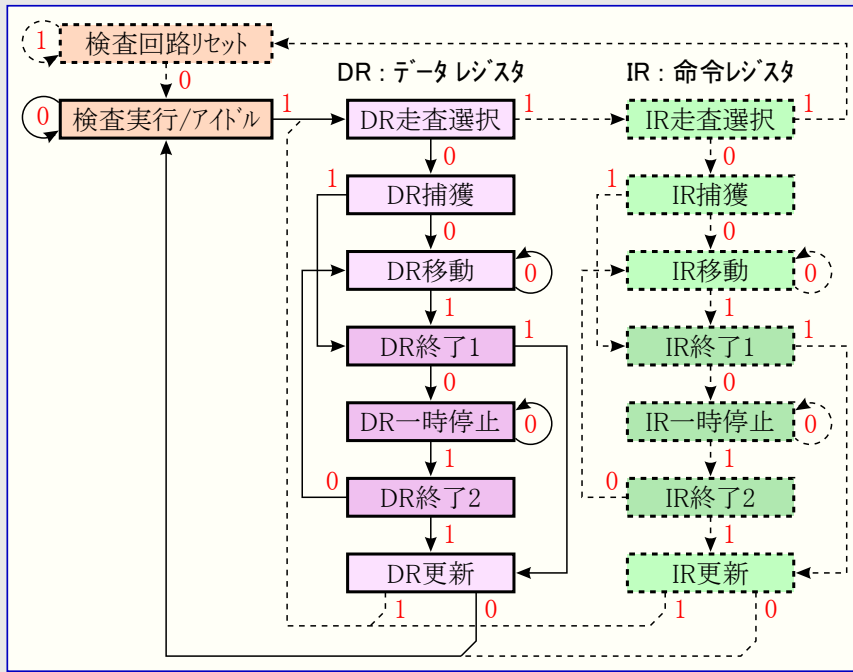
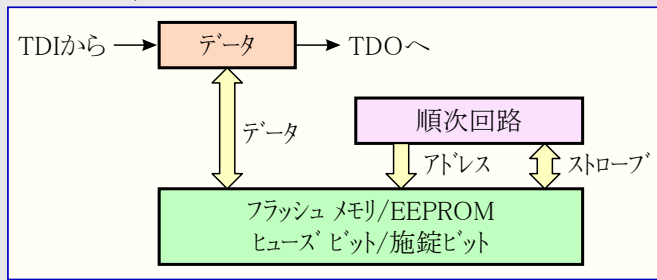


図28-17. フラッシュ バイト データ レジスタ



フラッシュ バイト データ レジスタを制御する順次回路はTCKによってクロック駆動されます。通常動作中内でフラッシュ メモリの各バイトに対して8ビットが移動され、TAP制御器を通して誘導するために必要としたこのクロック周期はフラッシュ バイト データ レジスタに対し、それが使用者にとって明白に操作を完了するための十分なクロック パルス数で順次回路を自動的に送ります。けれどもページ設定中に各**DR更新**状態間で更に数ビットが移動される場合、各**DR更新**状態間の最小11 TCK周期を保証するためのいくつかのTCK周期に対し、TAP制御器は検査実行/アイドル状態に留まるべきです。

28.9.12. 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28-17を参照してください。

28.9.13. 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。

28.9.14. 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(1①参照)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。

28.9.15. 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**(237頁の表28-13参照)時間待ちます。

28.9.16. 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。245頁の「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. 上位拡張アドレス設定命令(2②参照)を使い、書き込みアドレス上位拡張バイトを設定します。
4. アドレス上位設定命令(2③参照)を使い、書き込みアドレス上位バイトを設定します。
5. アドレス下位設定命令(2④参照)を使い、書き込みアドレス下位バイトを設定します。
6. データ設定命令(2⑤, 2⑥, 2⑦参照)を使い、書き込みデータを設定します。
7. ページ内の全語(ワード)数分5.と6.を繰り返します。
8. ページ書き込み命令(2⑧参照)を使い、ページをフラッシュ メモリに書き込みます。
9. ページ書き込み完了検査(2⑨参照)を使い、完了までポーリングするか、**tWLRH_FLASH**(237頁の表28-13.参照)時間待ちます。
10. 全データ書き込みまで3.~9.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, 2④参照)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**(230頁の表28-11.参照)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. 先頭命令のLSBから始まって最終命令のMSBで終わるページ内にバイト単位で全命令語を移動入力することによってページ全体を設定します。フラッシュ バイト データ レジスタの内容をフラッシュ ページ位置に複写し、各新規ワード前にプログラム カウンタを自動増加するのに**DR更新**を使います。
6. JTAG命令**PROG_COMMANDS**を入力します。
7. ページ書き込み命令(2⑧参照)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(2⑨参照)を使い、完了までポーリングするか、**tWLRH_FLASH**(237頁の表28-13.参照)時間待機します。
9. 全データ書き込みまで3.~8.を繰り返します。

28.9.17. 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, 3④参照)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(3⑤参照)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, 3④参照)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**(230頁の表28-11.参照)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. 先頭命令のLSBから始まり最終命令のMSBで終わるページ(またはフラッシュ)内で全命令語を移動出力することにより、ページ(またはフラッシュ)全体を読みます。**DR捕獲**状態はフラッシュ メモリからのデータ捕獲と各語(ワード)が読まれた後にプログラム カウンタも自動増加します。**DR移動**に先立って**DR捕獲**が起きることに注意してください。従って移動出力される先頭バイトは有効なデータを含みます。
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。

28.9.18. 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。245頁の「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**(237頁の表28-13.参照)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。

28.9.19. 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. 全データが読み出されてしまうまで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。

28.9.20. ヒューズビットの書き込み (訳注:本項は修正してあります。)

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(237頁の表28-13参照)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(237頁の表28-13参照)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(237頁の表28-13参照)時間待ちます。

28.9.21. 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(237頁の表28-13参照)時間待ちます。

28.9.22. ヒューズ/施錠ビットの読み出し (訳注:本項は修正してあります。)

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。

28.9.23. 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。

28.9.24. 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。

29. 電気的特性

29.1. 絶対最大定格 (警告)

動作温度	-40°C ~ +85°C
保存温度	-65°C ~ +150°C
RESET,VBUS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
VBUSピン許容電圧	-0.5V ~ +6.0V
最大動作電圧	6.0V
入出力ピン出力電流	40.0mA
消費電流	200.0mA

(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、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。

29.2. DC特性

表29-1. 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 _{IL}	Lowレベル入力電圧 (XTAL1,RESETを除く)	VCC=2.7~5.5V	-0.5		0.2VCC-0.1 (LVTTL) (注1)	V
V _{IL1}	Lowレベル入力電圧 (XTAL1)	VCC=2.7~5.5V	-0.5		0.1VCC (注1)	
V _{IL2}	Lowレベル入力電圧 (RESET)	VCC=2.7~5.5V	-0.5		0.1VCC (注1)	
V _{IH}	Highレベル入力電圧 (XTAL1,RESETを除く)	VCC=2.7~5.5V	0.2VCC+0.9 (LVTTL) (注2)		VCC+0.5	
V _{IH1}	Highレベル入力電圧 (XTAL1)	VCC=2.7~5.5V	0.7VCC (注2)		VCC+0.5	
V _{IH2}	Highレベル入力電圧 (RESET)	VCC=2.7~5.5V	0.9VCC (注2)		VCC+0.5	
V _{OL}	Lレベル出力電圧 (注3)	IOL=10mA, VCC=5V			0.7	
		IOL=5mA, VCC=3V			0.5	
V _{OH}	Hレベル出力電圧 (注4)	IOH=-10mA, VCC=5V	4.2			
		IOH=-5mA, VCC=3V	2.3			
I _{IL}	I/OピンLowレベル入力漏れ電流	VCC=5.5V			1	μA
I _{IH}	I/OピンHighレベル入力漏れ電流	確実なH/L範囲			1	
R _{RST}	RESETピンプルアップ抵抗		30		60	kΩ
R _{PU}	I/Oピンプルアップ抵抗		20		50	
I _{CC}	活動動作消費電流 (電力削減レジスタ(PRR0,1)全ビット=0)	VCC=3V, 4MHz			5	mA
		VCC=5V, 8MHz		10	15	
		VCC=5V, 16MHz			27	
	アイドル動作消費電流 (電力削減レジスタ(PRR0,1)全ビット=0)	VCC=3V, 4MHz			2	
		VCC=5V, 8MHz			6	
	パワーダウン動作消費電流	VCC=3V, WDT許可		<10	12	μA
		VCC=3V, WDT禁止		1	5	
V _{ACIO}	アナログ比較器入力変位(オフセット)電圧	VCC=5V, V _{in} =VCC/2		<10	40	mV
I _{ACLK}	アナログ比較器入力漏れ電流		-50		50	nA
t _{ACPD}	アナログ比較器伝播遅延時間	VCC=2.7V		750		ns
		VCC=4.0V		500		
R _{USB}	外部USB直列抵抗			22±5%		Ω
V _{REG}	電圧調整器出力電圧 (注5)	CUCAP=1μF±20%, VCC≥4.0V, I≤80mA または VCC≥3.4V, I≤55mA	3.0	3.3	3.6	V

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6)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=3Vで10mA,VCC=5Vで20mA)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。(訳注:本項の現状はテンプレートです。)

1. ポートC7,6のIOLの合計が100mAを超えるべきではありません。
2. ポートB7~0、E6,2のIOLの合計が100mAを超えるべきではありません。
3. ポートD7~0のIOLの合計が100mAを超えるべきではありません。
4. ポートF7~4,1,0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=3Vで10mA,VCC=5Vで20mA)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。(訳注:本項の現状はテンプレートです。)

1. ポートC7,6のIOHの合計が100mAを超えるべきではありません。
2. ポートB7~0、E6,2のIOHの合計が100mAを超えるべきではありません。
3. ポートD7~0のIOHの合計が100mAを超えるべきではありません。
4. ポートF7~4,1,0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。

注5: 調整器最大出力電流はUSB活動時に必要とされるUSB緩衝部電流(約25mA)で減じられるべきです。残りの調整器出力電流は外部応用に使うことができます。

注6: USB電氣的特性節で詳述されるようにD+/D-ポートは39Ωの直列抵抗器を通して印加する-1Vへの電圧低下に耐えます。

29.3. 外部クロック特性

図29-1. 外部クロック駆動波形

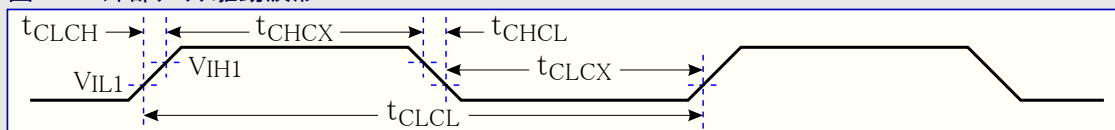


表29-2. 外部クロック特性

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	
Δt_{CLCL}	隣接クロック周期間の変化率		2		2	%

注: 本書内の全DC特性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づいています。

29.4. システムとリセットの特性

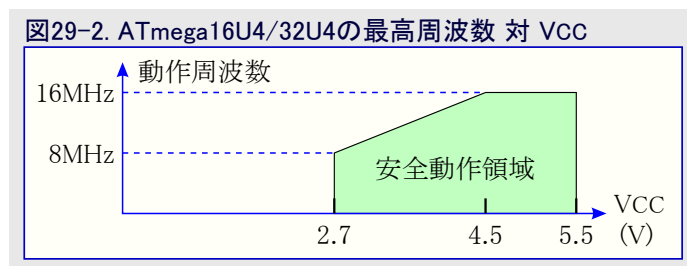
表29-3. リセット電氣的特性

シンボル	項目	条件	最小	代表	最大	単位
V_{POT}	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_{POR}	内部電源ONリセット信号保証のVCC開始電圧		-0.1		+0.1	
V_{CCR}	内部電源ONリセット信号保証のVCC上昇率		0.3			V/ms
V_{RST}	RESETピン閾値電圧		0.2 VCC		0.85 VCC	V
t_{RST}	RESETピンでの最小パルス幅	VCC=5V, 25°C		400		ns

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。

29.5. 最高速と動作電圧

最高周波数は動作電圧に依存します。図29-2.で示されるように最高周波数対動作電圧曲線は2.7～4.5V間で直線です。



29.6. 2線直列インターフェース特性

下表は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。本デバイスの2線直列インターフェースは記載条件下に於いて、これらの必要条件を越えるか、または合致します。

図29-3. 2線直列バス タイミング

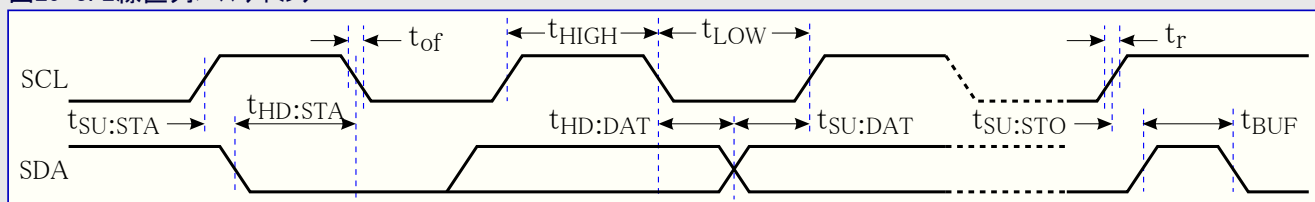


表29-4. 2線直列インターフェース必要条件

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($V_{IHmin} \rightarrow V_{ILmax}$)	$10pF < C_b < 400pF$ ③	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	μA
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	Ω
R_p	プルアップ抵抗値	$f_{SCL} \leq 100kHz$	$(V_{CC}-0.4V)/3mA$	$1000ns/C_b$	
		$f_{SCL} > 100kHz$	$(V_{CC}-0.4V)/3mA$	$300ns/C_b$	
$t_{HD:STA}$	(再送)開始条件保持時間	$f_{SCL} \leq 100kHz$	4.0		μs
t_{LOW}	SCLクロックLowレベル時間	$f_{SCL} > 100kHz$	0.6		
		$f_{SCL} \leq 100kHz$	4.7		
t_{HIGH}	SCLクロックHighレベル時間	$f_{SCL} > 100kHz$	0.6		ns
		$f_{SCL} \leq 100kHz$	4.7		
$t_{SU:STA}$	再送開始条件準備時間	$f_{SCL} > 100kHz$	0.6		
		$f_{SCL} \leq 100kHz$	4.0		μs
$t_{HD:DAT}$	データ保持時間	$f_{SCL} > 100kHz$	0	3.45	
		$f_{SCL} \leq 100kHz$	0	0.9	
$t_{SU:DAT}$	データ準備時間	$f_{SCL} > 100kHz$	250		ns
		$f_{SCL} \leq 100kHz$	100		
$t_{SU:STO}$	停止条件準備時間	$f_{SCL} > 100kHz$	4.0		
		$f_{SCL} \leq 100kHz$	0.6		μs
t_{BUF}	停止条件→開始条件間バス開放時間	$f_{SCL} > 100kHz$	4.7		
		$f_{SCL} \leq 100kHz$	1.3		

① ATmega16U4/32U4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ATmega16U4/32U4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。

⑥ ATmega16U4/32U4の2線直列インターフェースによって生成した実際のLow区間は $(1/f_{SCL}-2/f_{CK})$ で、従ってLow時間の必要条件に対して $f_{SCL}=100kHz$ で厳密に満たされるには f_{CK} が(概ね)6MHz以上でなければなりません。

⑦ ATmega16U4/32U4の2線直列インターフェースによって生成した実際のLow区間は $(1/f_{SCL}-2/f_{CK})$ で、従って $f_{CK}=8MHz$ 時、厳密には $f_{SCL} > 308kHz$ でLow時間の必要条件が満たされません。それにも拘らず、バスに接続されたAT90mega16U4/32U4装置は相応な t_{LOW} 許容余地のある他の装置だけでなく他のATmega16U4/32U4装置と最高速(400kHz)で通信できるでしょう。

29.7. SPIタイミング特性

図29-4. SPI タイミング必要条件 (主装置動作)

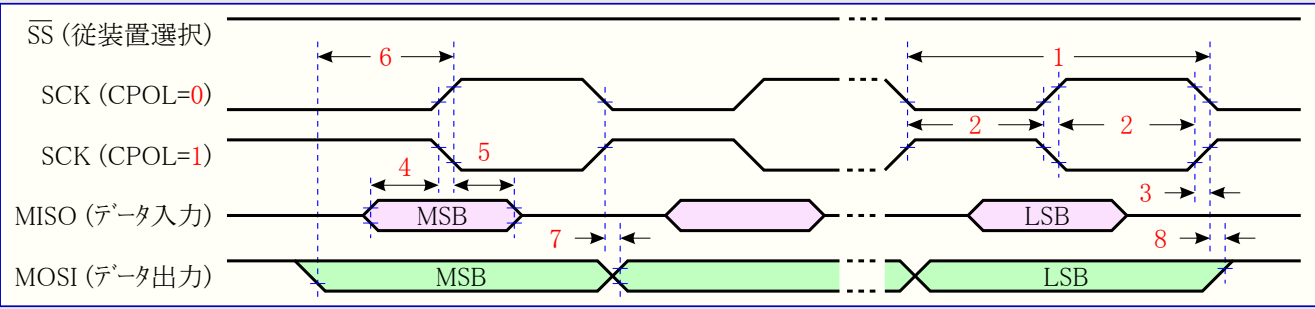


図29-5. SPI タイミング必要条件 (従装置動作)

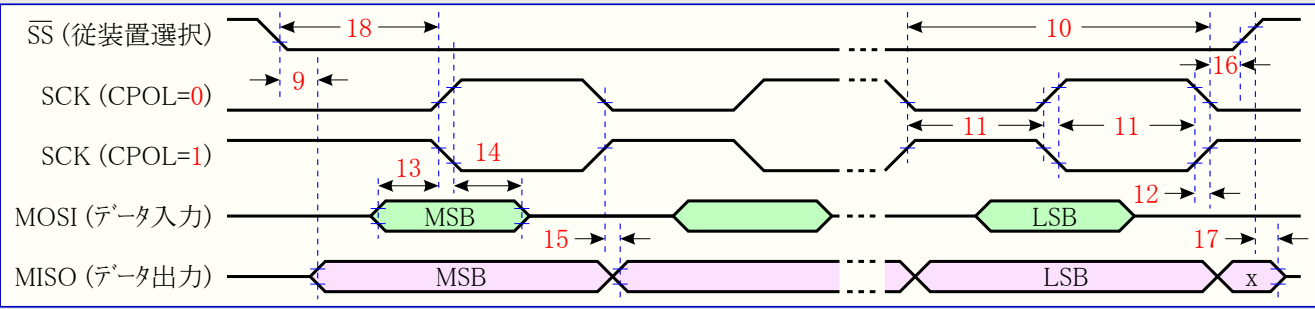


表29-5. SPI タイミング特性

番号	項目	動作種別	最小	代表	最大	単位
1	SCK周期	主装置		表17-5.参照		
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			1600	
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 _{SCK}		
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		ns
10	SCK周期	従装置	4t _{CK}			
11	SCK High/Low期間 (注)	従装置	2t _{CK}			
12	SCK上昇/下降時間	従装置			1600	
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	t _{CK}			
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	20			

注: SPIプログラミングでの最小SCK High/Low期間は、2t_{CLCL}(t_{CK}<12MHz)、3t_{CLCL}(t_{CK}≥12MHz)です。

29.8. ハードウェア ブート受付タイミング特性

図29-6. ハードウェア ブート タイミング必要条件

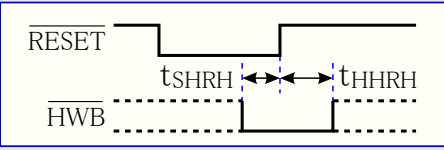


表29-6. ハードウェア ブート タイミング

シンボル	項目	最小	最大	単位
t _{SHRH}	RESET ↑ 前Low確保時間	0		ns
t _{HHRH}	RESET ↑ 後Low保持時間	付加起動遅延時間 (SUTによるCK計数+WDT計数)		

29.9. A/D変換器特性

表29-7. A/D変換特性

シンボル	項目	条件	最小	代表	最大	単位
	分解能	シングル エント' 入力変換		10		ビット
		×1/×10/×40差動入力変換		8		
		×200差動入力変換		8		
TUE	絶対精度	VREF=4V, 変換クロック =200kHz	VCC=4V, シングル エント'	2.0	3.0	
			VCC=5V, ×1/×10/×40差動	2.0	3.0	
			VCC=5V, ×200差動	2.0	4.0	
INL	積分非直線性誤差	VREF=4V, 変換クロック =200kHz	VCC=4V, シングル エント'	0.5	1.5	
			VCC=5V, ×1/×10/×40差動	0.3	1.5	
			VCC=5V, ×200差動	0.5	1.5	
DNL	微分非直線性誤差	VREF=4V, 変換クロック =200kHz	VCC=4V, シングル エント'	0.4	0.7	LSB
			VCC=5V, ×1/×10/×40差動	0.3	1.0	
			VCC=5V, ×200差動	0.6	1.0	
	利得誤差	VREF=4V, 変換クロック =200kHz	VCC=4V, シングル エント'	-2.5	-1.0	
			VCC=5V, ×1/×10/×40差動	0.0	-1.5	
			VCC=5V, ×200差動	0.0	-1.8	
	変位(オフセット)(ゼロ)誤差	VREF=4V, 変換クロック =200kHz	VCC=4V, シングル エント'	-2.5	-1.0	
			VCC=5V, 差動	0.0	-1.8	
AVCC	アナログ供給電圧		VCC-0.3		VCC+0.3	
VREF	基準電圧	シングル エント' 入力変換	2.56		AVCC	V
		差動入力変換	2.56		AVCC-0.5	
VIN	入力電圧	シングル エント' 入力変換	GND		VREF	
		差動入力変換	0		AVCC	
	入力周波数帯域	シングル エント' 入力変換		38.5		kHz
		差動入力変換		4		
VINT	内蔵2.56V基準電圧		2.4	2.56	2.8	V
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			100		MΩ

29.10. 校正付き内蔵RC発振器の精度

表29-8. 校正付き内蔵RC発振器の校正精度

校正種別	周波数	VCC	温度	校正精度
工場校正	8.0MHz	3V	25°C	±10%
使用者校正	7.3~8.1MHz	2.7~5.5V	-40~85°C	±1%

30. 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。

活動動作とアイドル動作の消費電流測定は電力削減レジスタ(PRR)で全ビットが設定(1)で行なわれ、従って関係周辺機能部はOFFにされます。アナログ比較器もこれらの測定中に禁止されます。詳細については27頁の「電力削減(レジスタ)」をご覧ください。

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) $C_L(\text{負荷容量}) \times V_{CC}(\text{動作電圧}) \times f(\text{I/Oピンの平均切り替え周波数})$ 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。

30.1. 活動動作消費電流

図30-1. 活動動作消費電流 対 周波数 (100kHz~1MHz, TA=25°C)

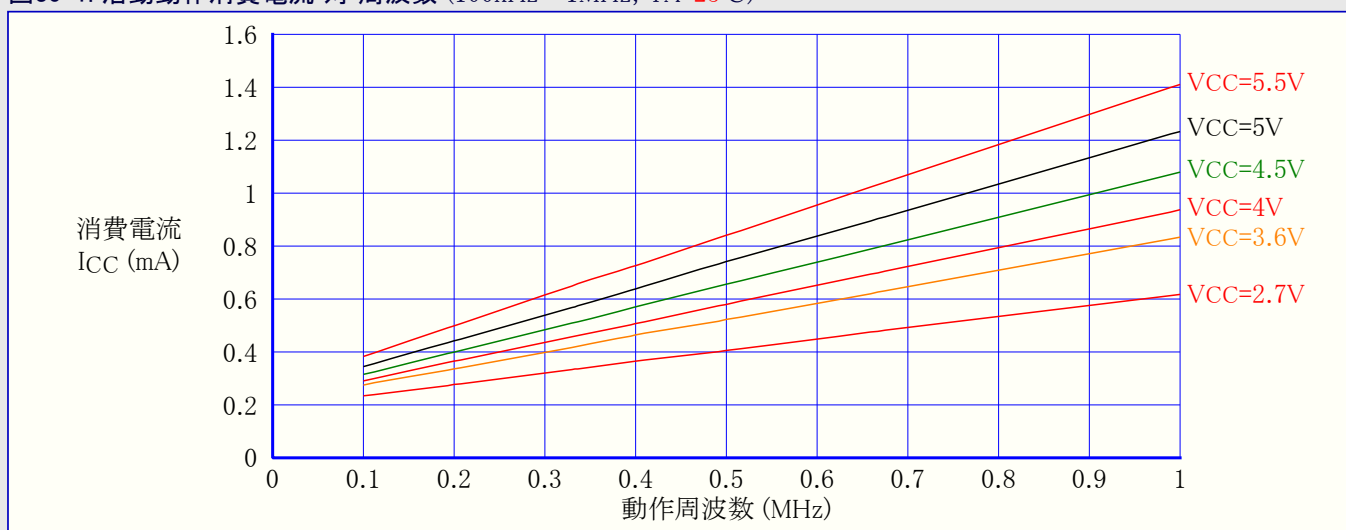


図30-2. 活動動作消費電流 対 周波数 (100kHz~1MHz, TA=85°C)

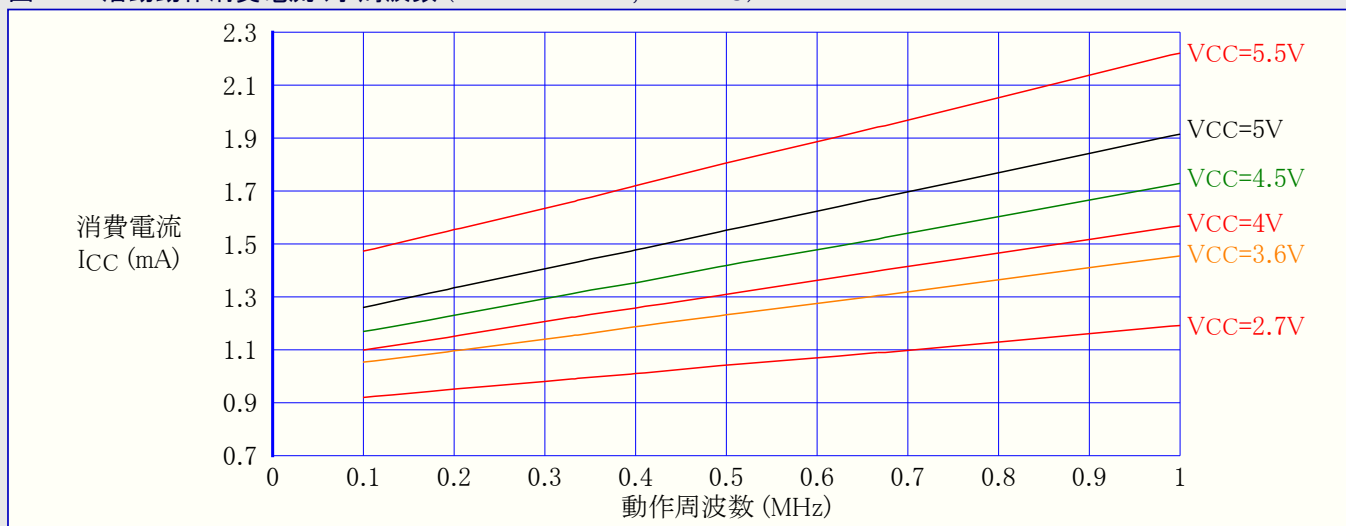


図30-3. 活動動作消費電流 対 周波数 (2MHz~16MHz, TA=-40°C)

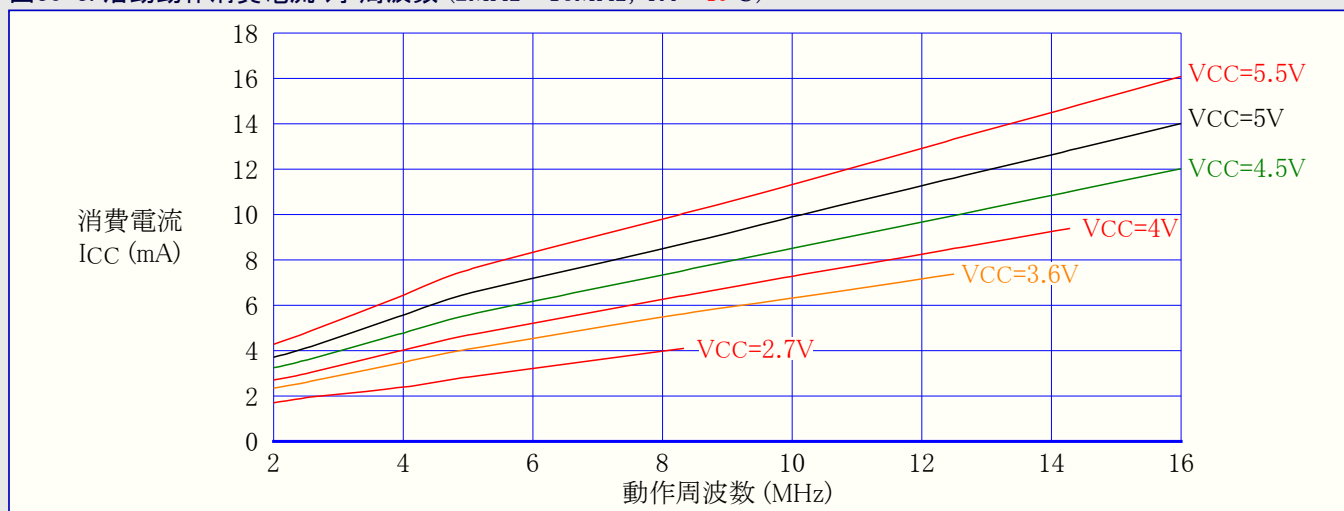


図30-4. 活動動作消費電流 対 周波数 (2MHz~16MHz, TA=25°C)

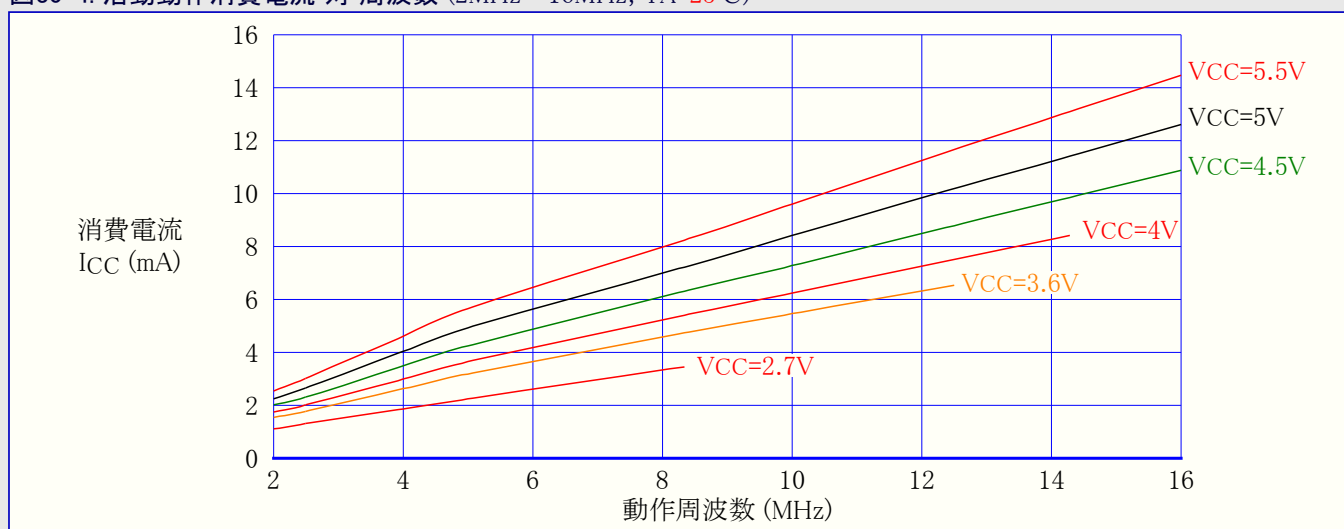
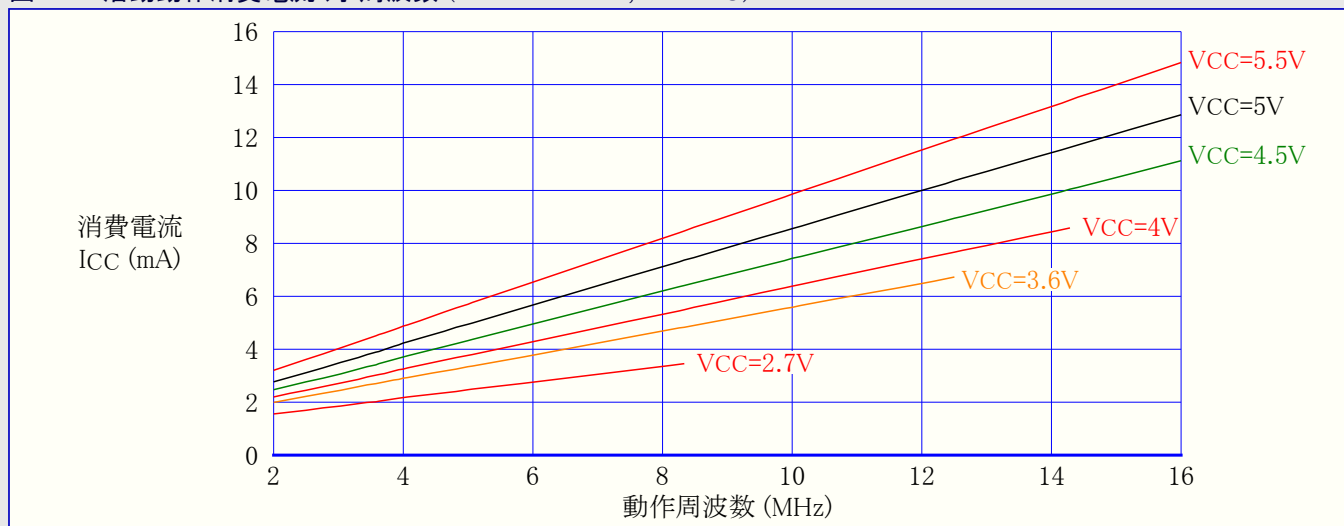


図30-5. 活動動作消費電流 対 周波数 (2MHz~16MHz, TA=85°C)



30.2. アイドル動作消費電流

図30-6. アイドル動作消費電流 対 周波数 (100kHz~1MHz, TA=25°C)

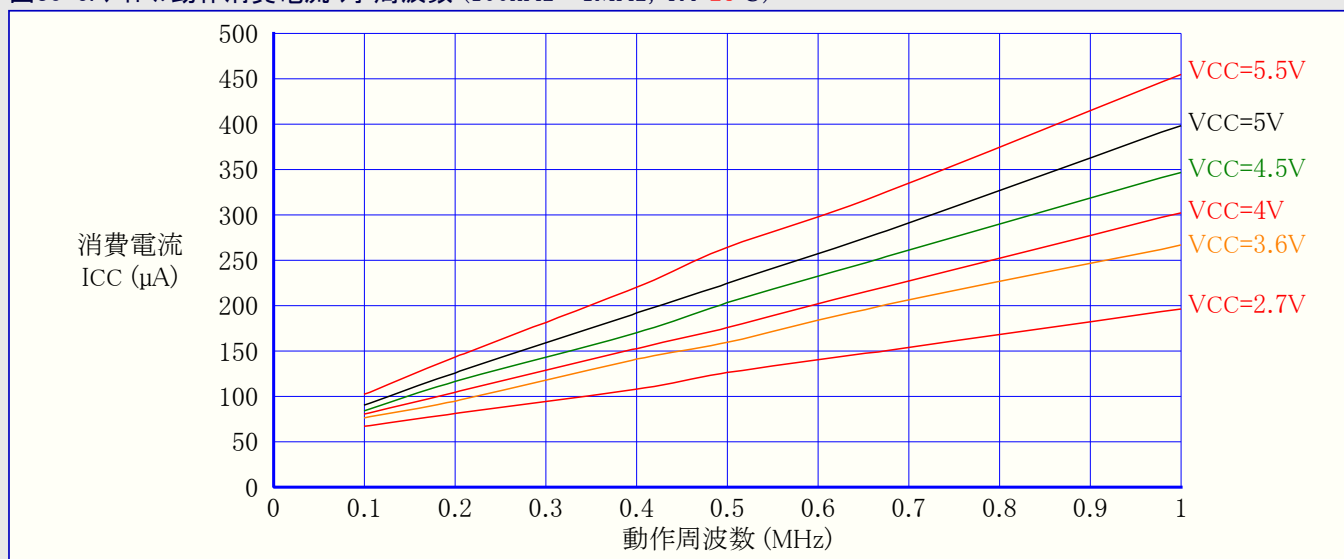


図30-7. アイドル動作消費電流 対 周波数 (100kHz~1MHz, TA=85°C)

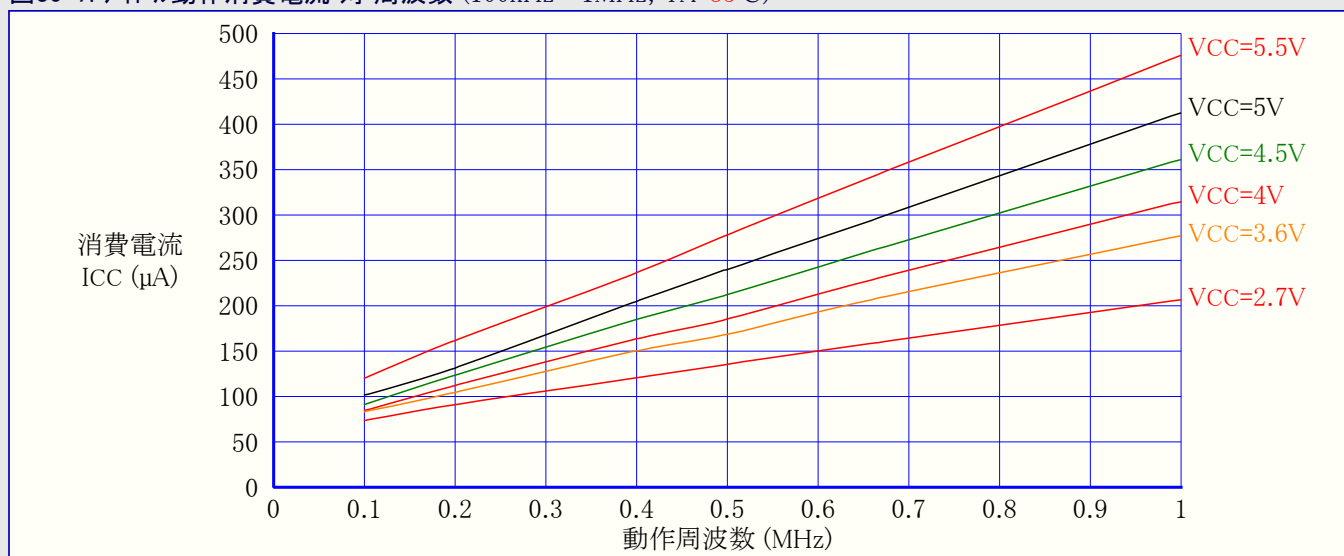


図30-8. アイドル動作消費電流 対 周波数 (2MHz~16MHz, TA=25°C)

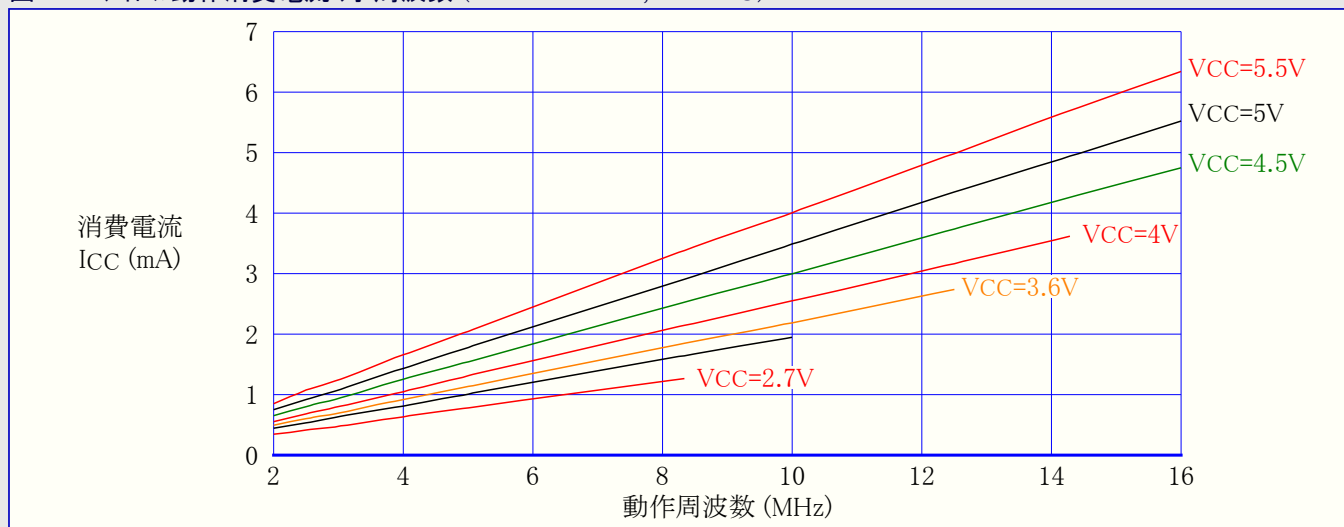
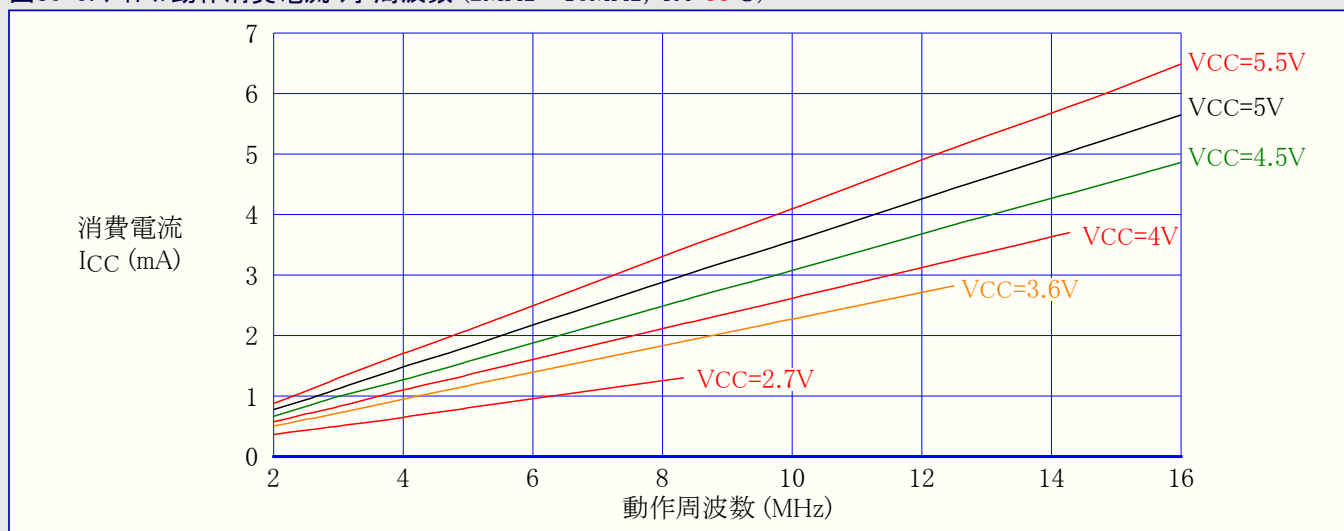


図30-9. アイドル動作消費電流 対 周波数 (2MHz~16MHz, TA=85°C)



30.3. パワーダウン動作消費電流

図30-10. パワーダウン動作消費電流 対 動作電圧 (ウォッチドッグ タイマ禁止)

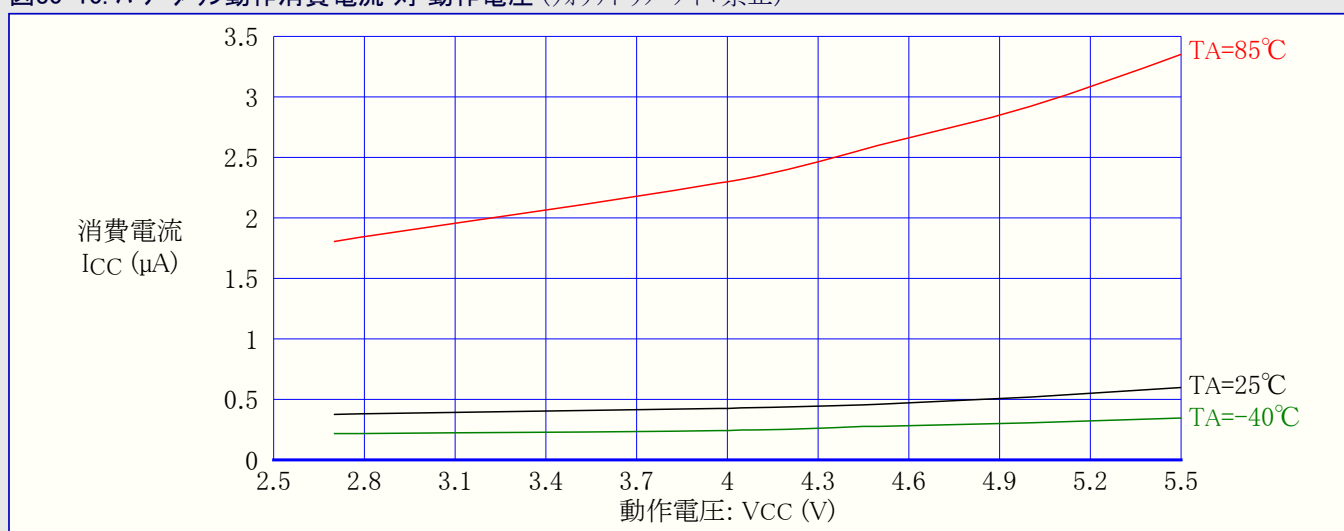


図30-11. パワーダウン動作消費電流 対 動作電圧 (ウォッチドッグ タイマ許可)

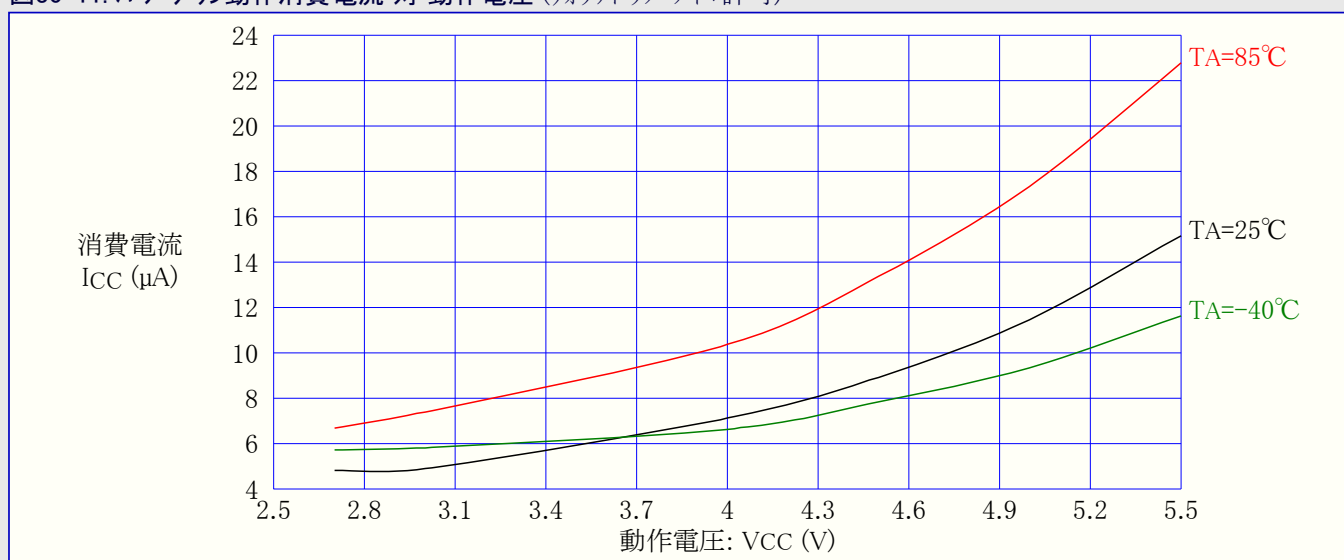
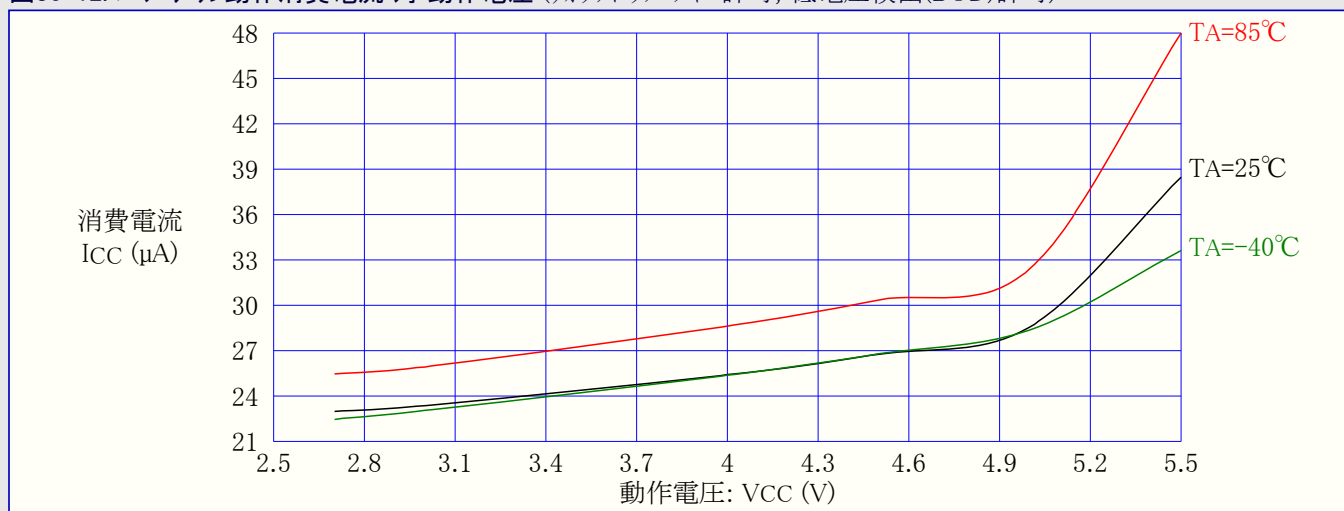
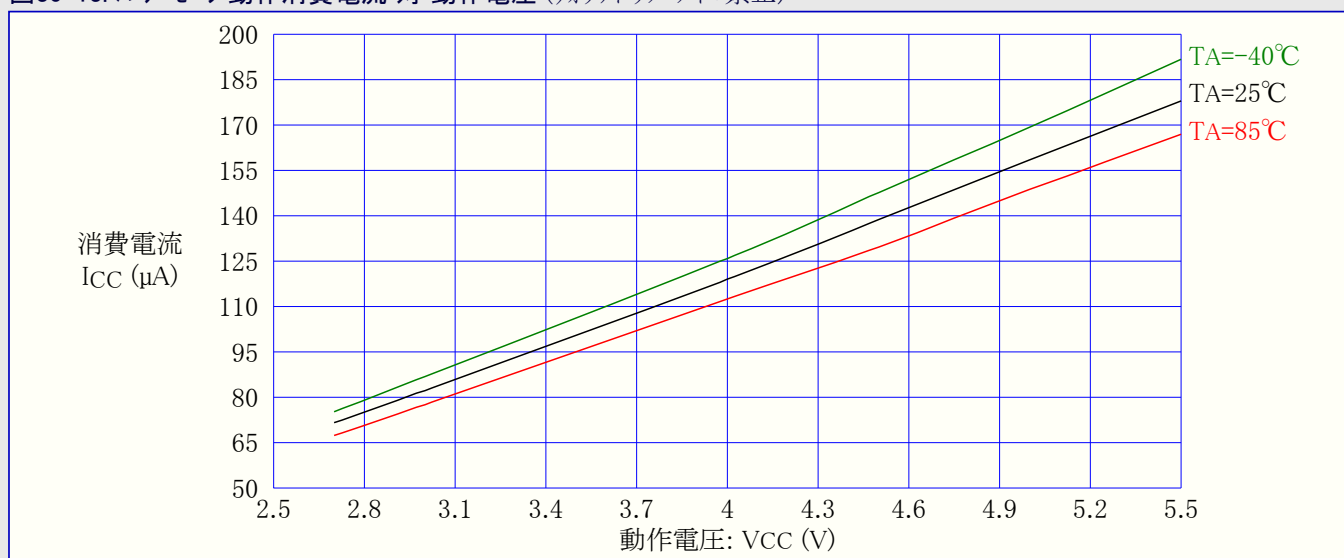


図30-12. パワーダウン動作消費電流 対 動作電圧 (ウォッチドッグ タイマ許可, 低電圧検出(BOD)許可)



30.4. パワーセーブ動作消費電流

図30-13. 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)



30.5. ピン プルアップ

図30-14. I/Oピンプルアップ抵抗電流 対 入力電圧 (VCC=2.7V)

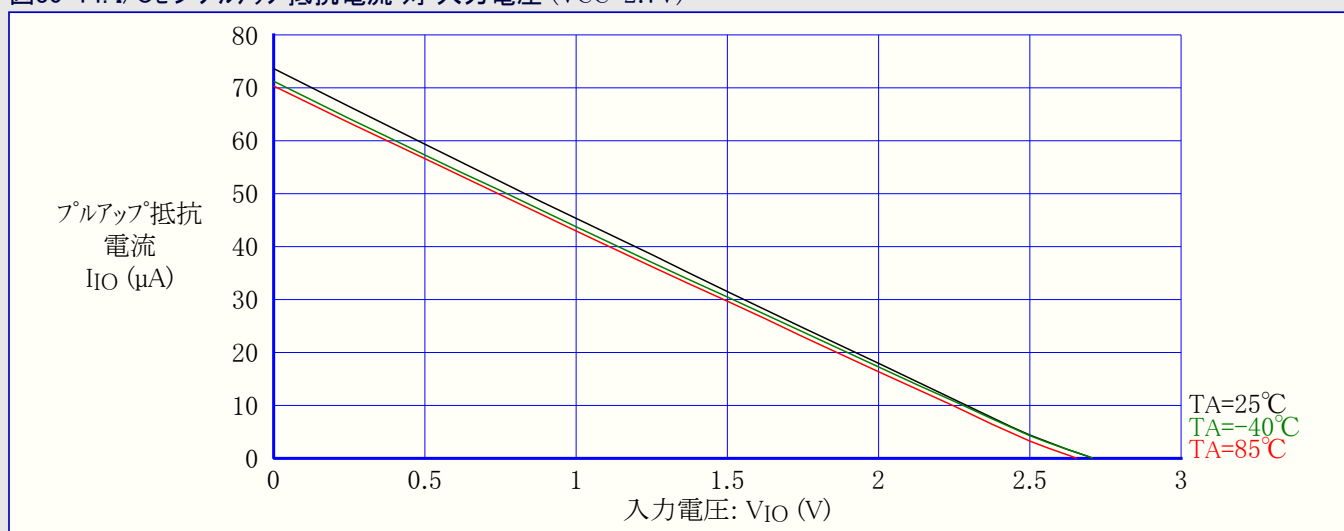


図30-15. I/Oピンプルアップ抵抗電流 対 入力電圧 (VCC=5V)

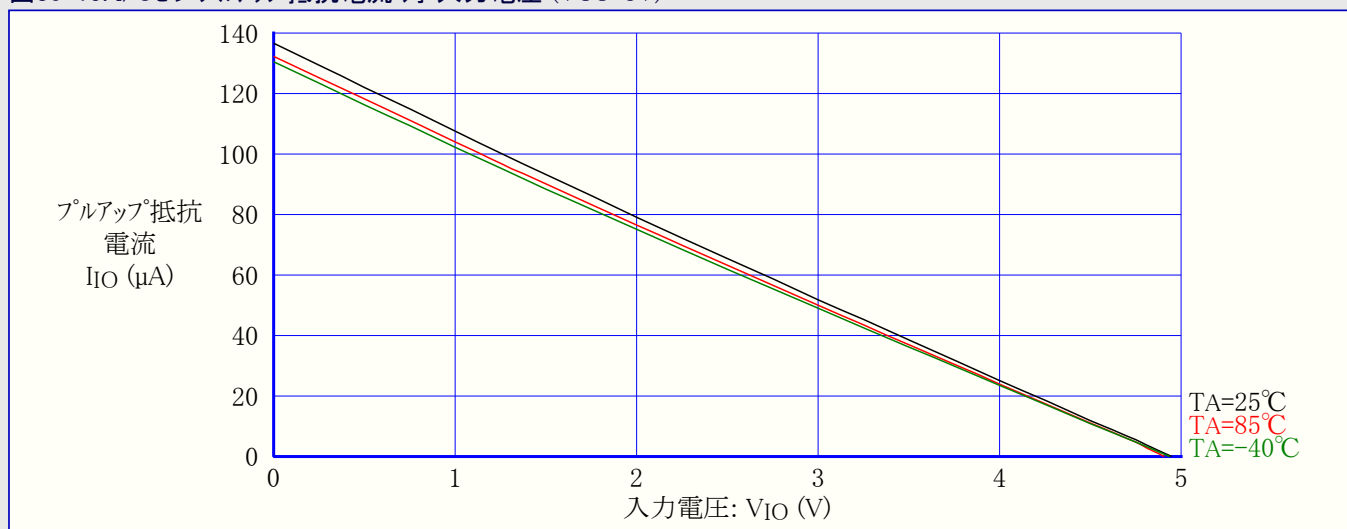
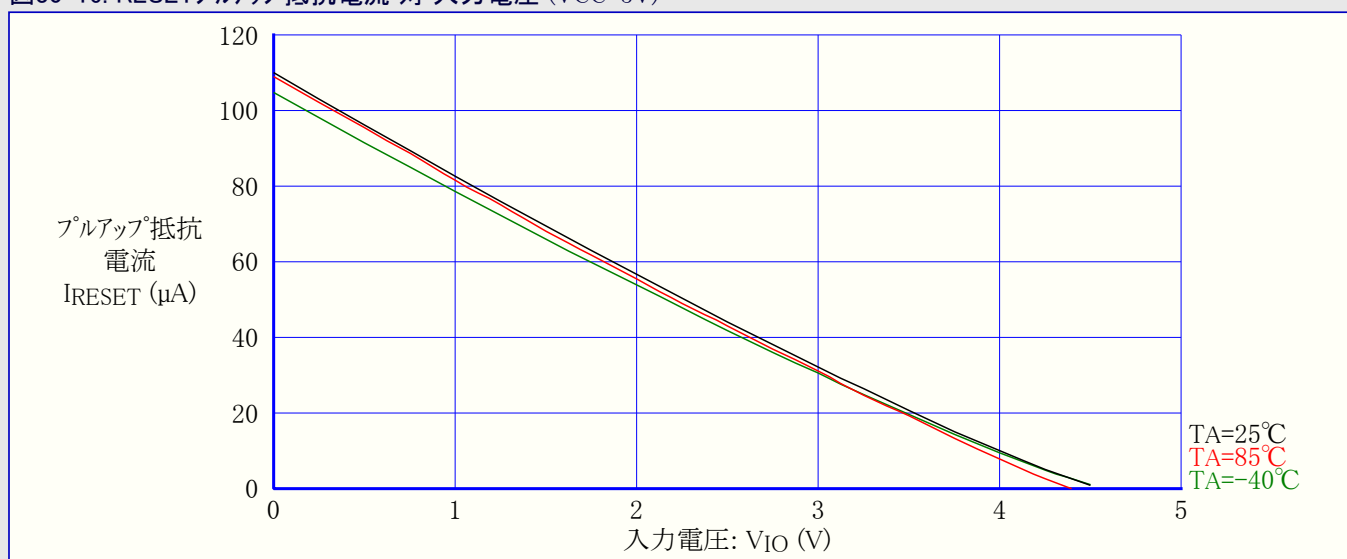


図30-16. RESETプルアップ抵抗電流 対 入力電圧 (VCC=5V)



30.6. ピン駆動能力

図30-17. I/Oピン出力電圧 対 吸い込み電流 (VCC=3V)

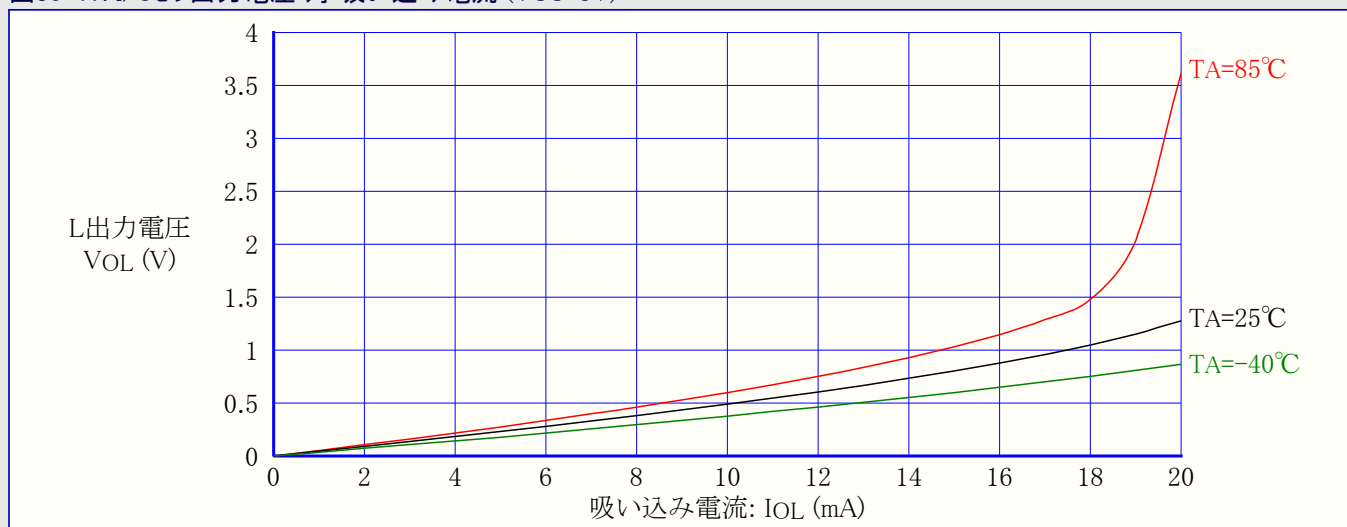


図30-18. I/Oピン出力電圧 対 吸い込み電流 (VCC=5V)

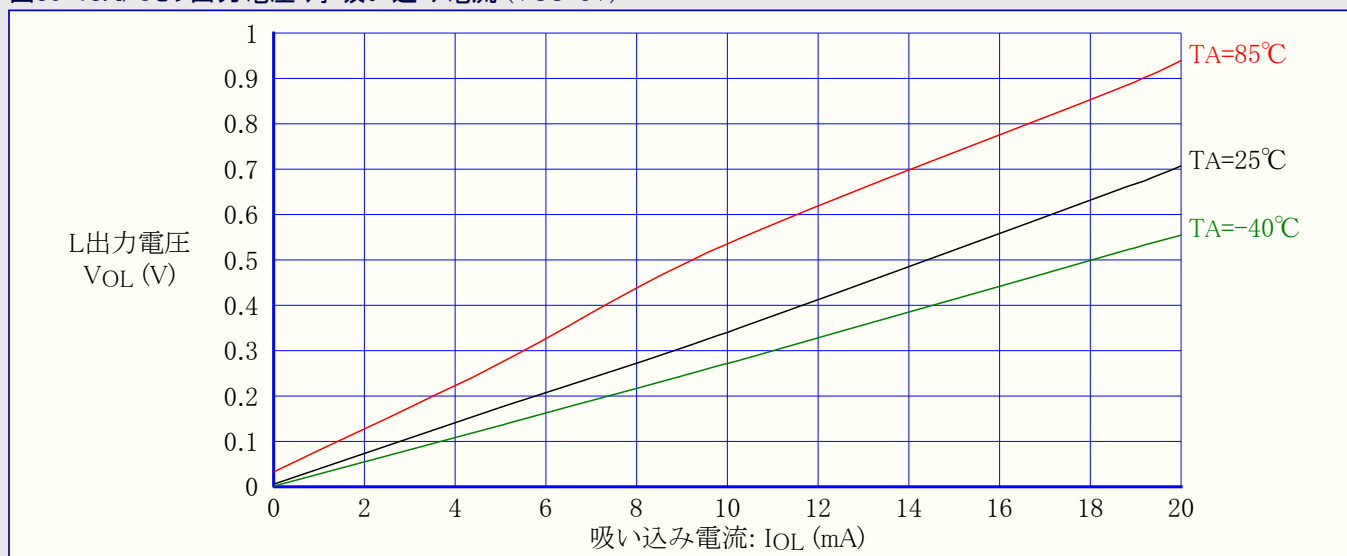


図30-19. I/Oピン出力電圧 対 吐き出し電流 (VCC=3V)

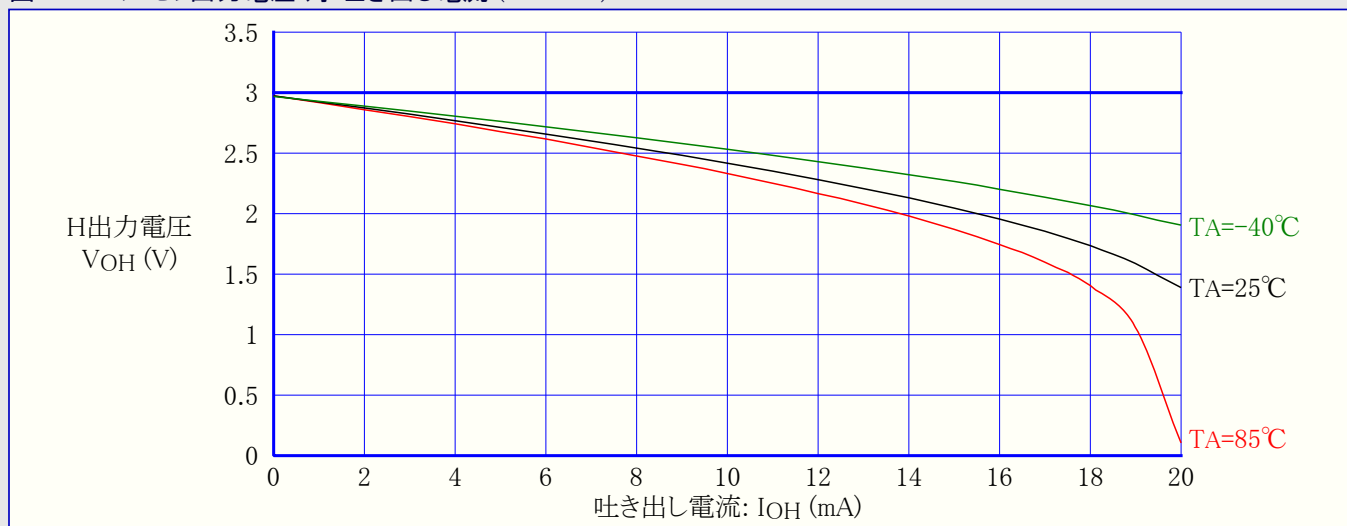


図30-20. I/Oピン出力電圧 対 吐き出し電流 (VCC=5V)

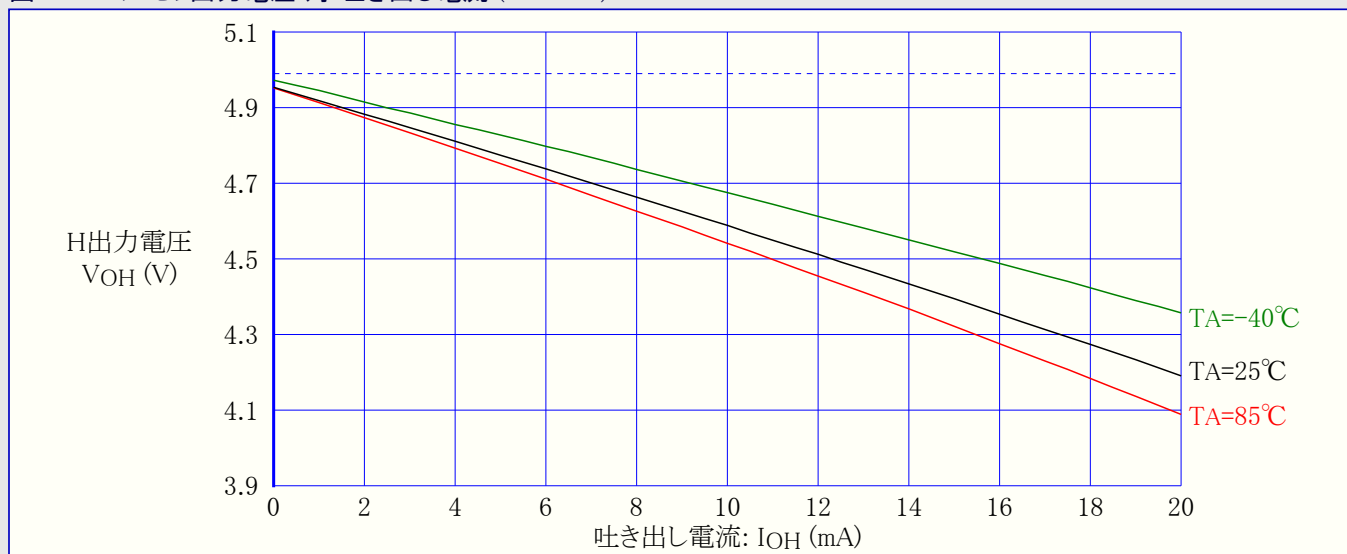
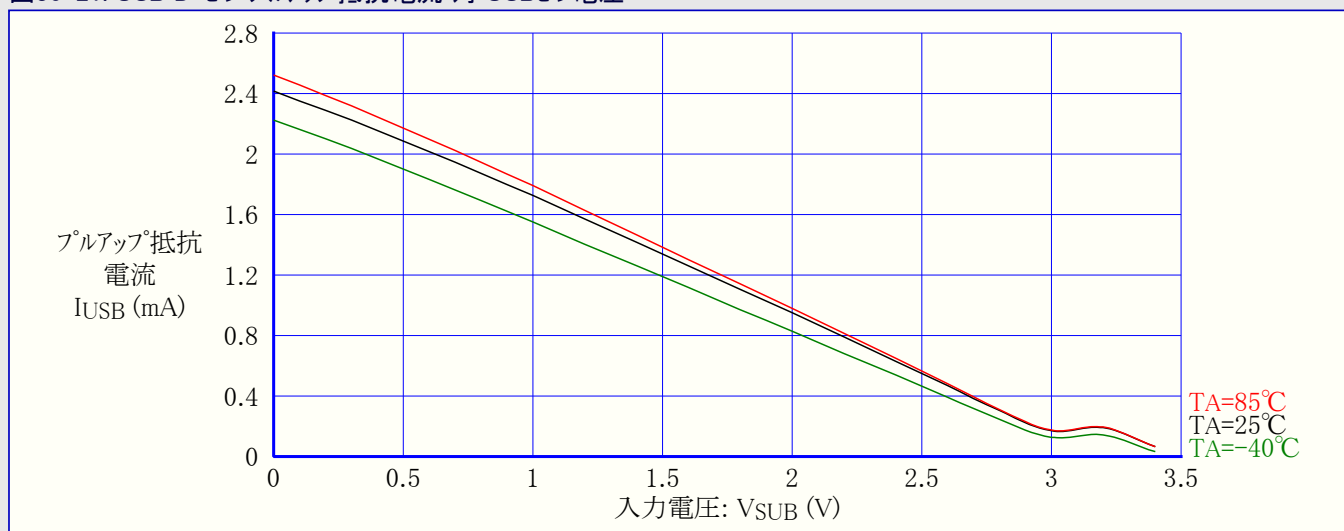


図30-21. USB D+ピンプルアップ抵抗電流 対 USBピン電圧



30.7. ピン 閾値とヒステリシス

図30-22. I/Oピン入力閾値(スレッショルド)電圧 対 動作電圧 (V_{IH} , 1読み値)

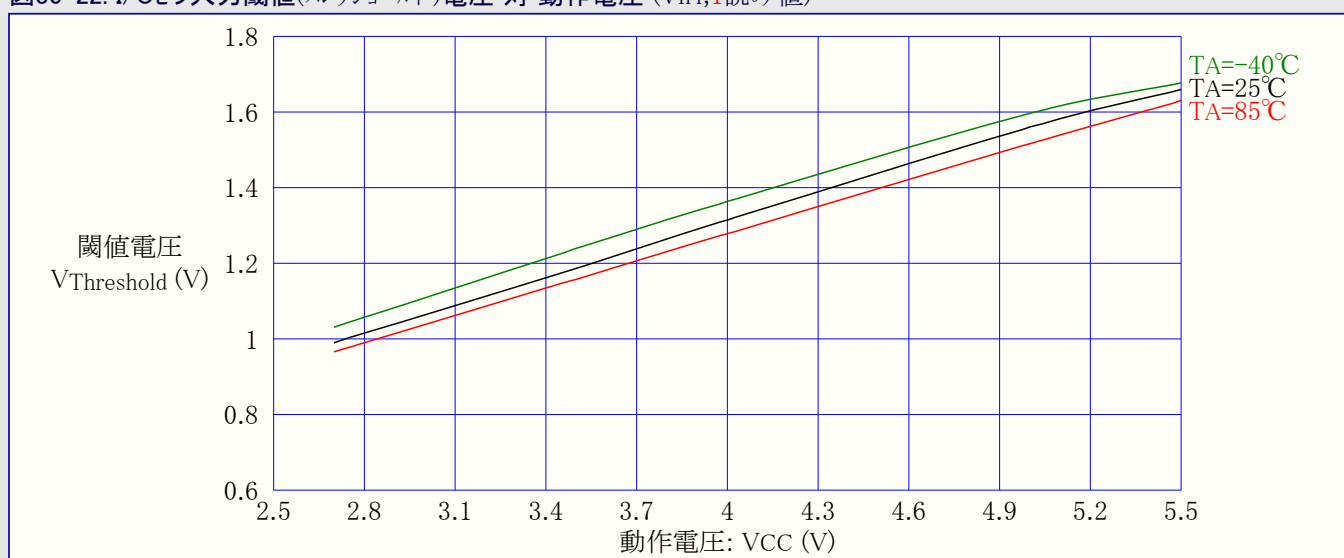


図30-23. I/Oピン入力閾値(スレッショルド)電圧 対 動作電圧 (V_{IL} , 0読み値)

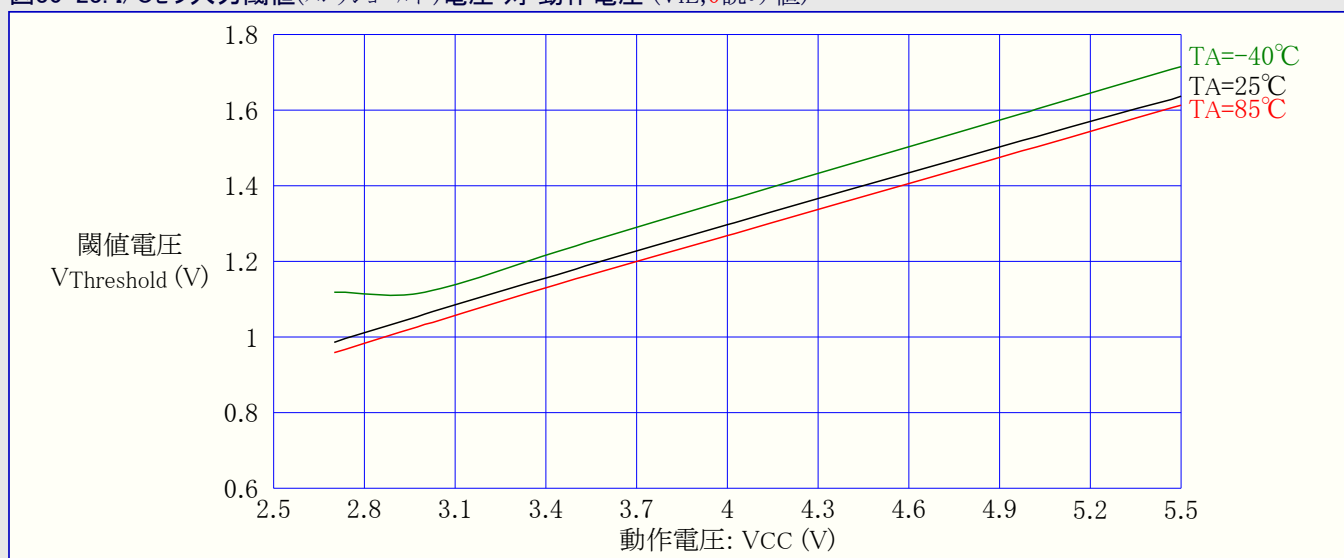


図30-24. USBピン入力閾値(スレッショルド)電圧 対 動作電圧 (V_{IH} , 1読み値)

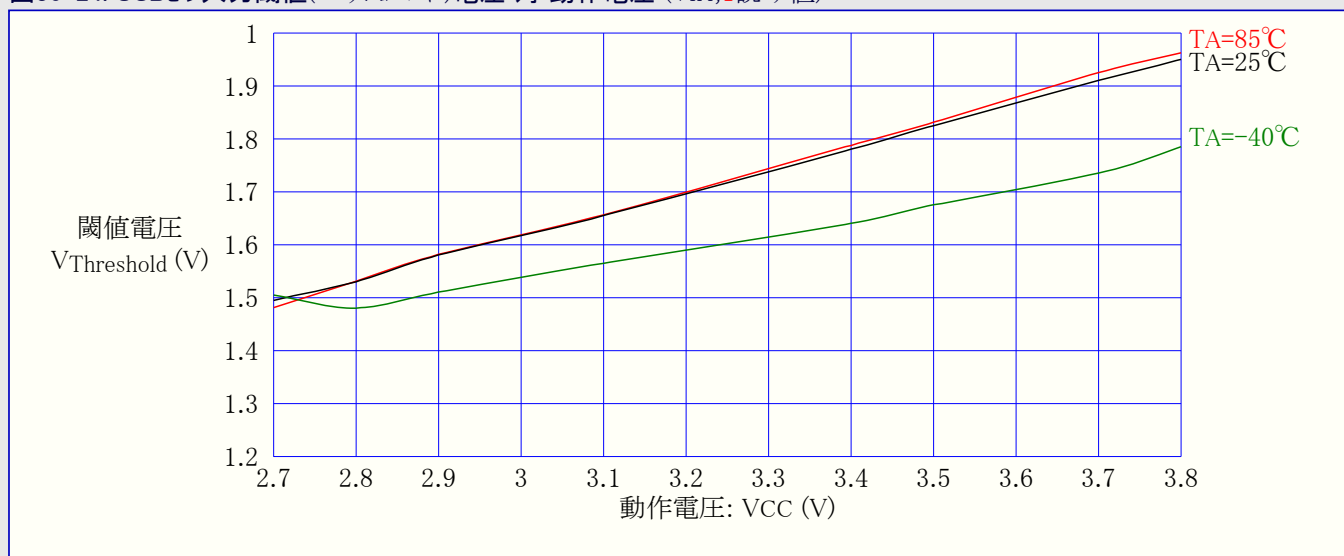


図30-25. USBピン入力閾値(スレッショルド)電圧 対 動作電圧 (V_{IL} , 0読み値)

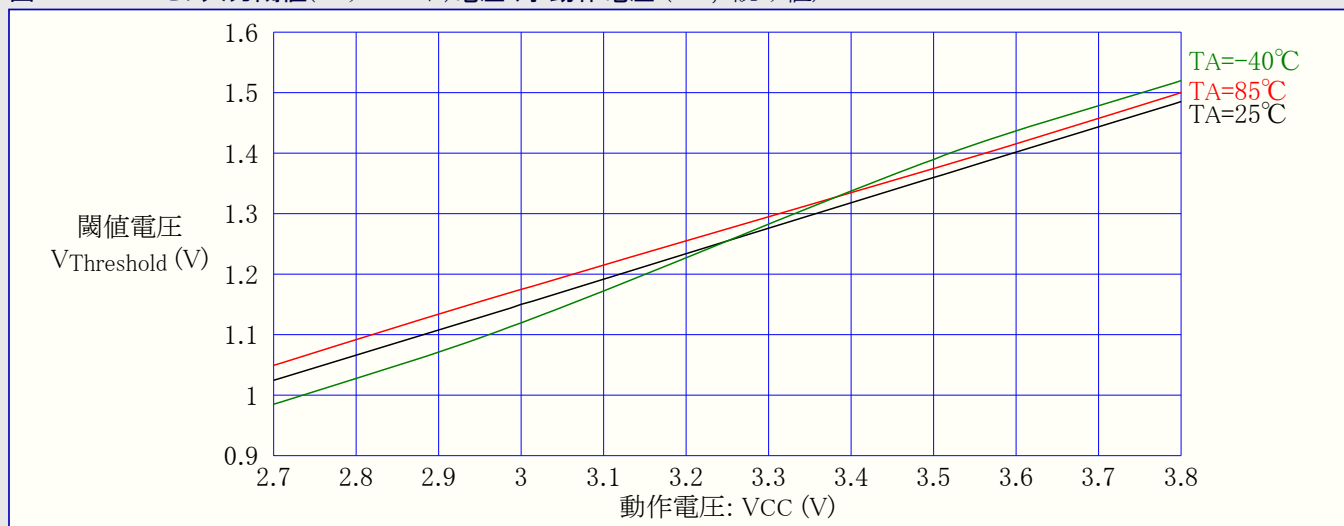


図30-26. VBUSピン入力閾値(スレッショルド)電圧 対 動作電圧 (V_{IH} , 1読み値)

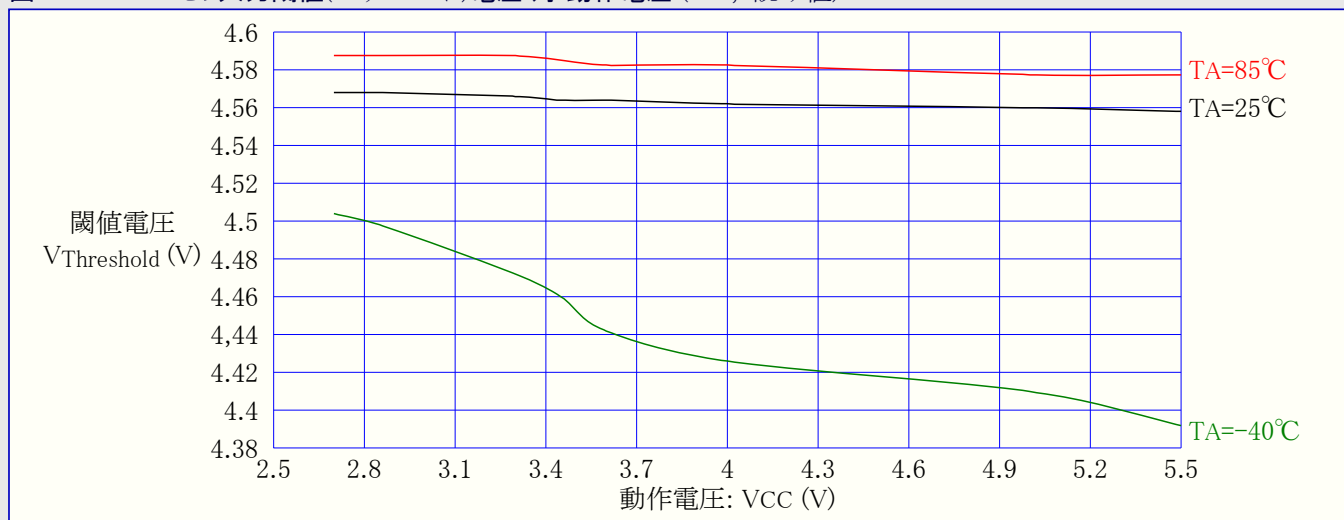
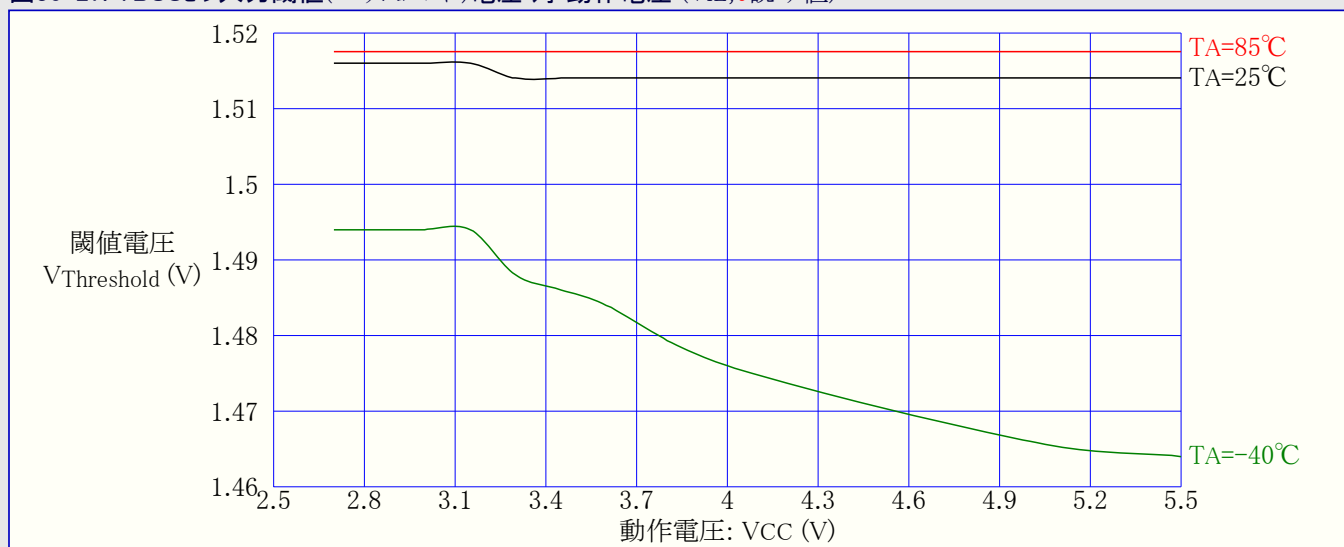


図30-27. VBUS⁺入力閾値(スレッショルド)電圧 対 動作電圧 (VIL,0読み値)



30.8. 低電圧検出器(BOD)閾値

図30-28. 低電圧検出器(BOD)閾値(スレッショルド)電圧 対 動作温度 (検出電圧2.6V)

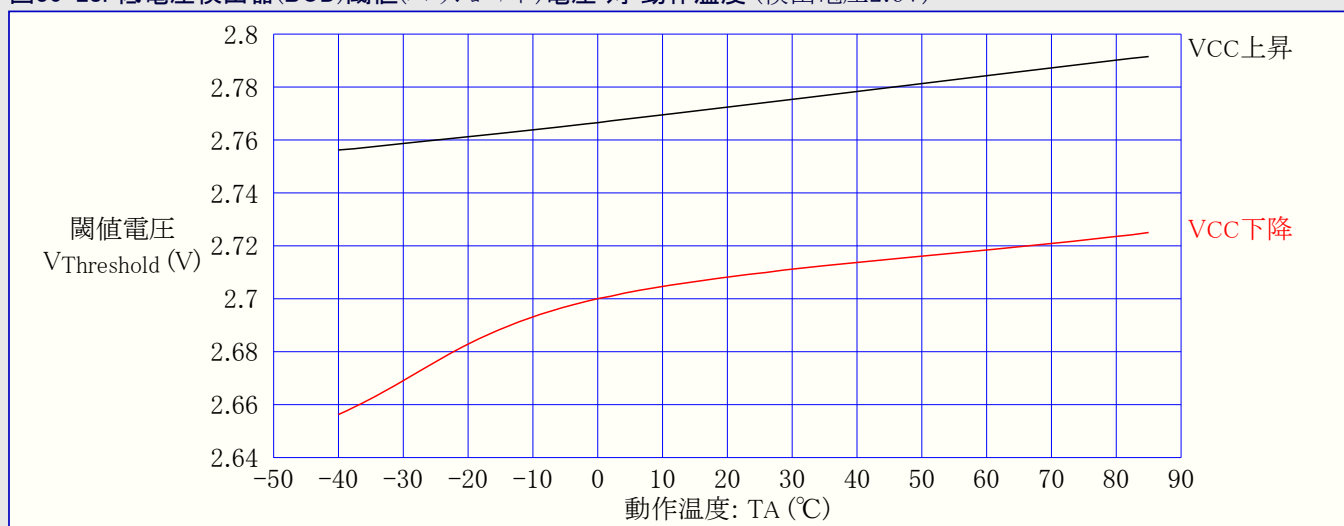


図30-29. 低電圧検出器(BOD)閾値(スレッショルド)電圧 対 動作温度 (検出電圧3.5V)

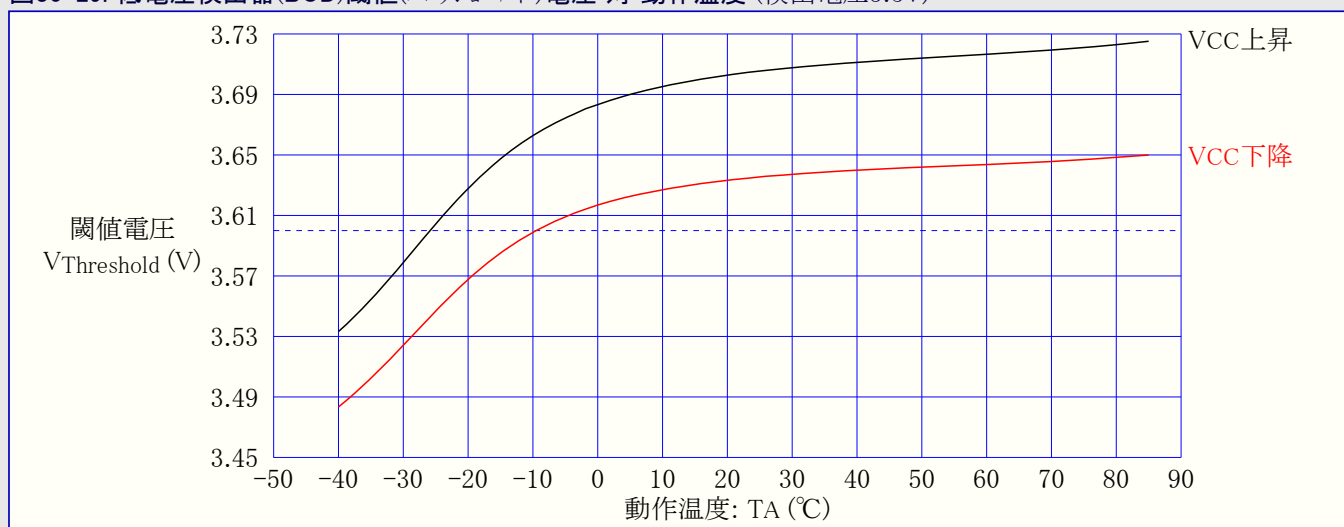


図30-30. 低電圧検出器(BOD)閾値(スレッショルド)電圧 対 動作温度 (検出電圧4.3V)

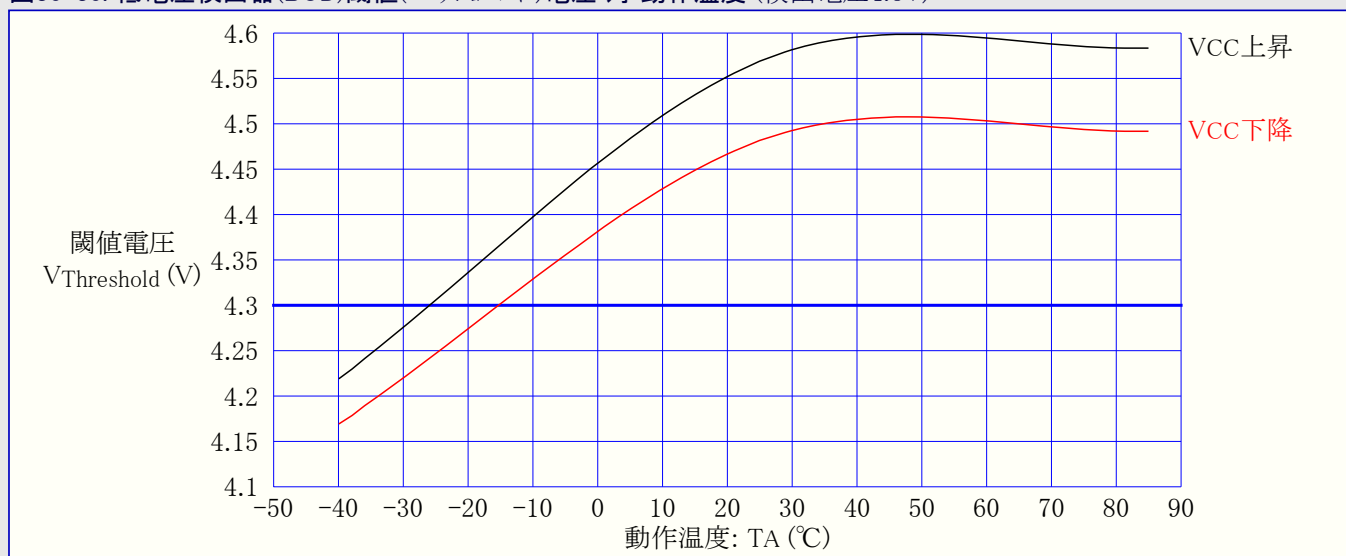


図30-31. バントキャップ電圧 対 動作電圧

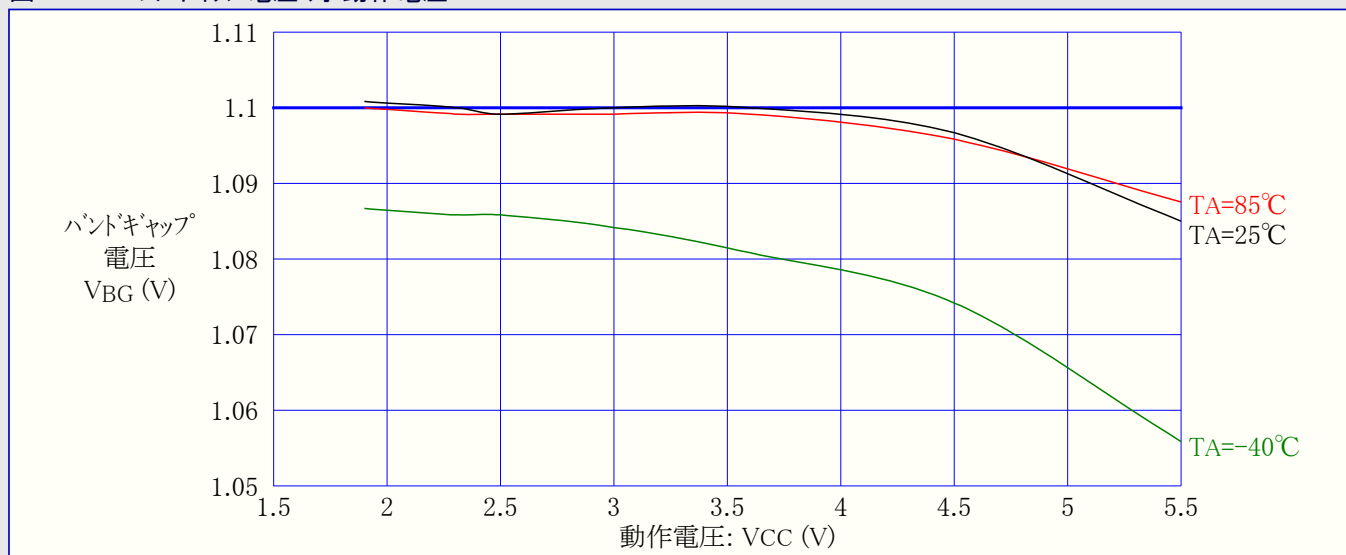
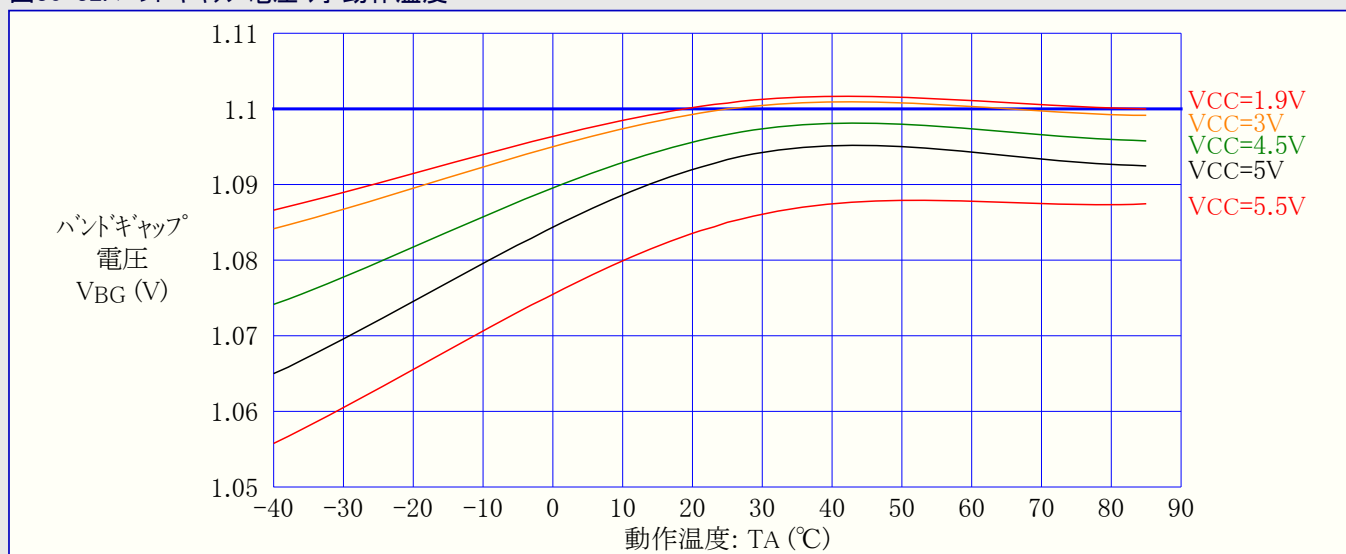


図30-32. バントキャップ電圧 対 動作温度



30.9. 内部発振器周波数

図30-33. ウォッチドッグ発振器周波数 対 動作温度

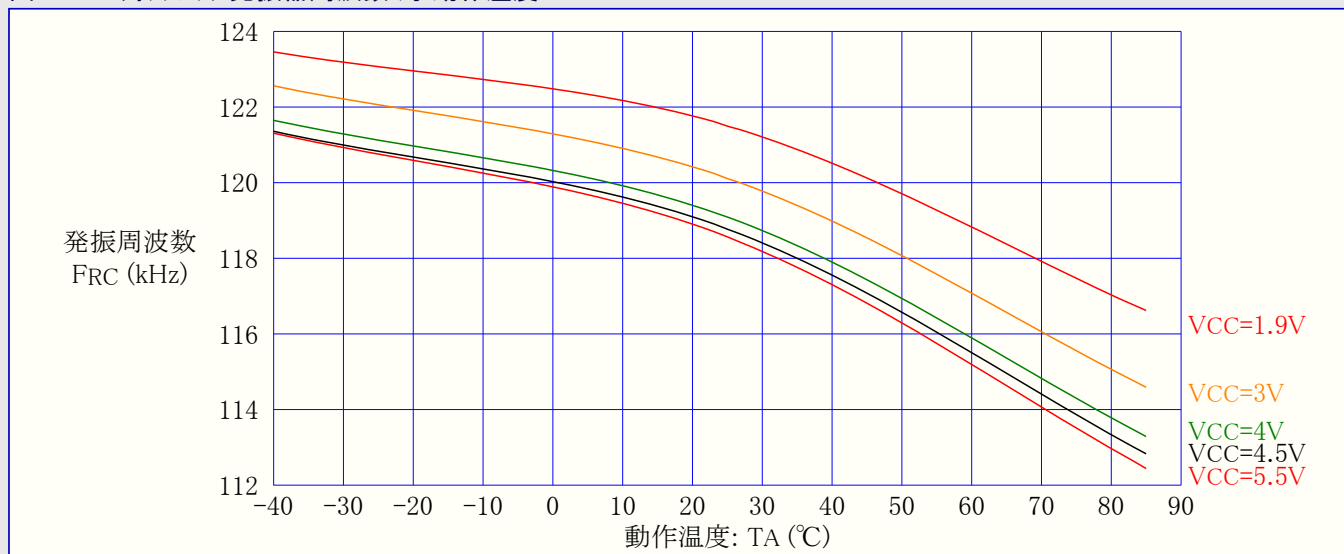


図30-34. ウォッチドッグ発振器周波数 対 動作電圧

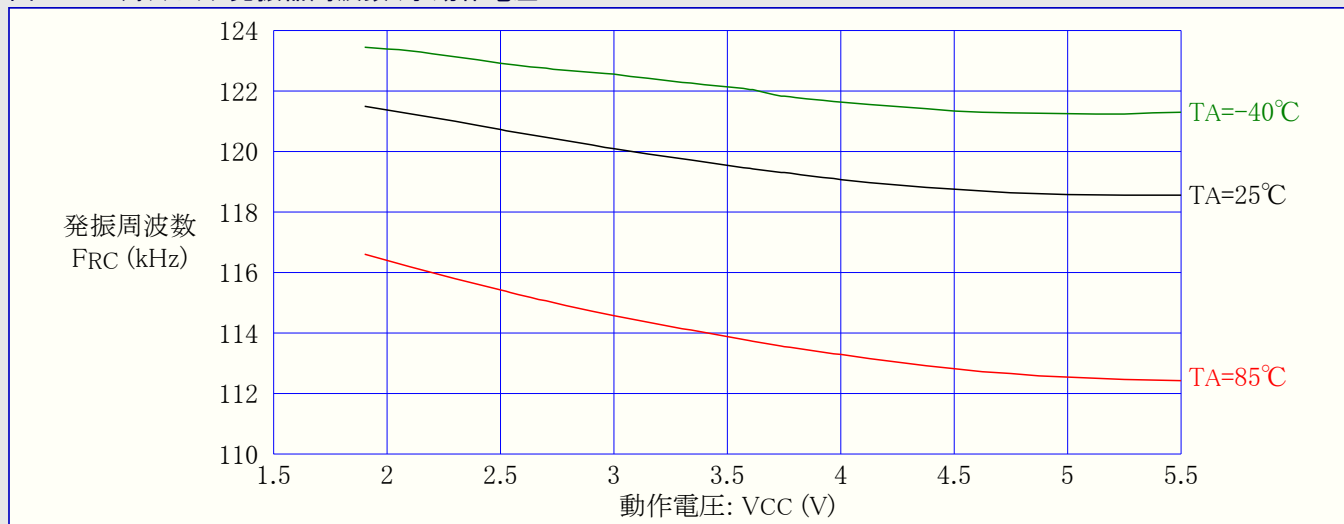


図30-35. 校正付き8MHz内蔵RC発振器周波数 対 発振校正(OSCCAL)値

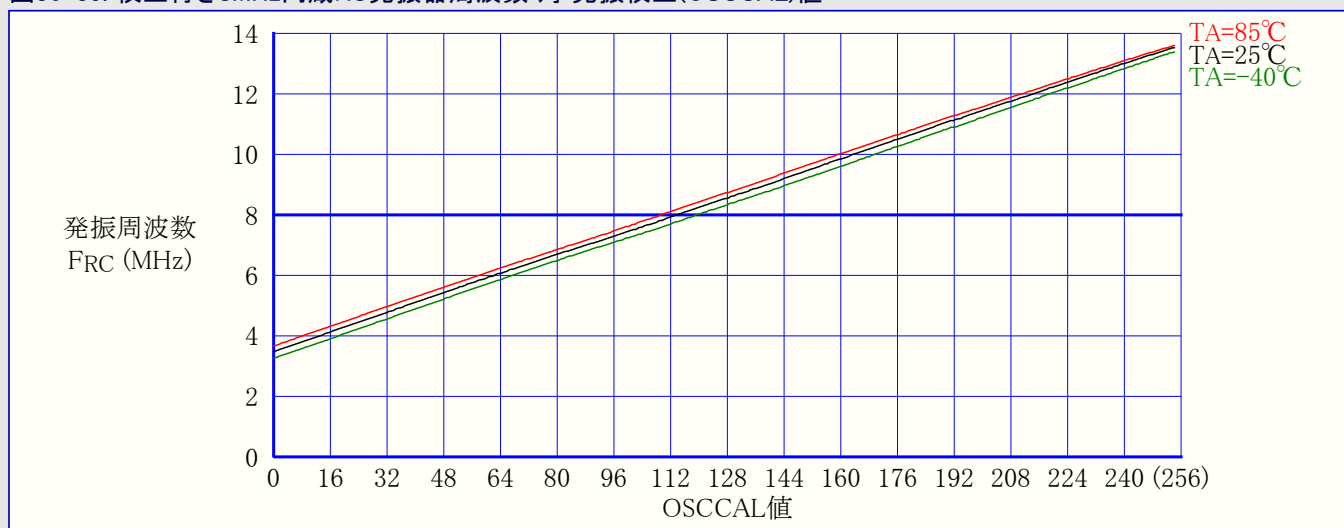


図30-36. 校正済み8MHz内蔵RC発振器周波数 対 動作温度

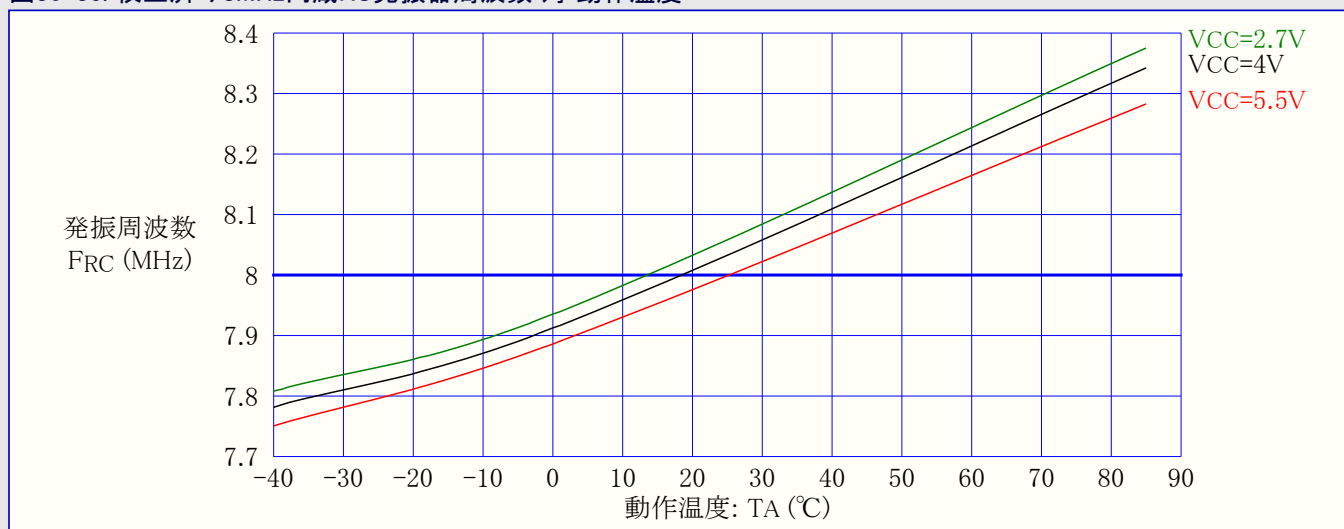


図30-37. 校正済み8MHz内蔵RC発振器周波数 対 動作電圧

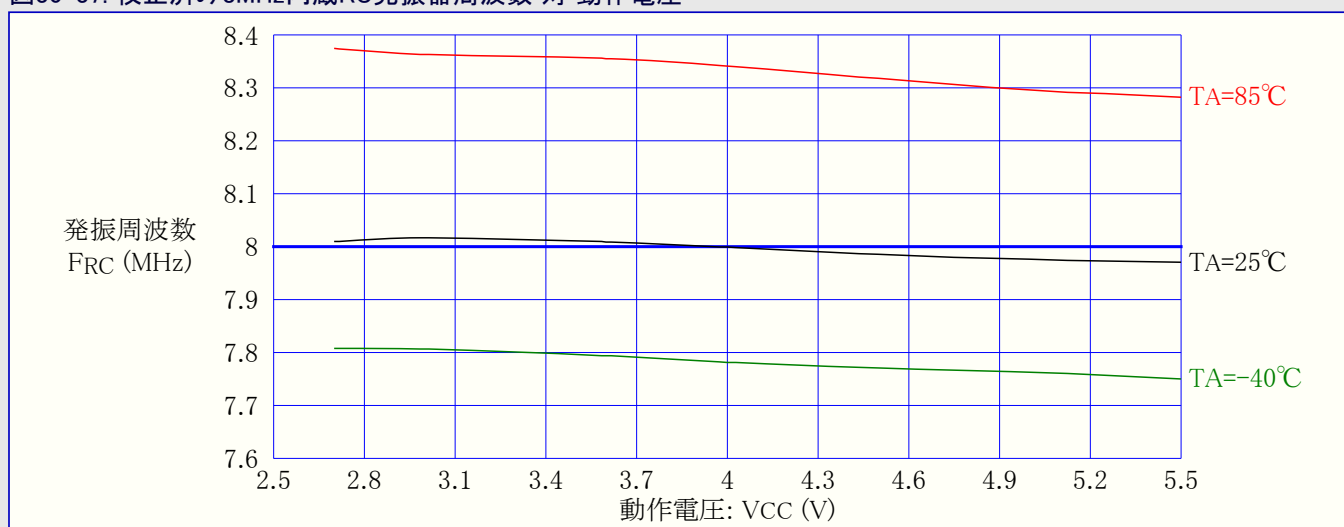
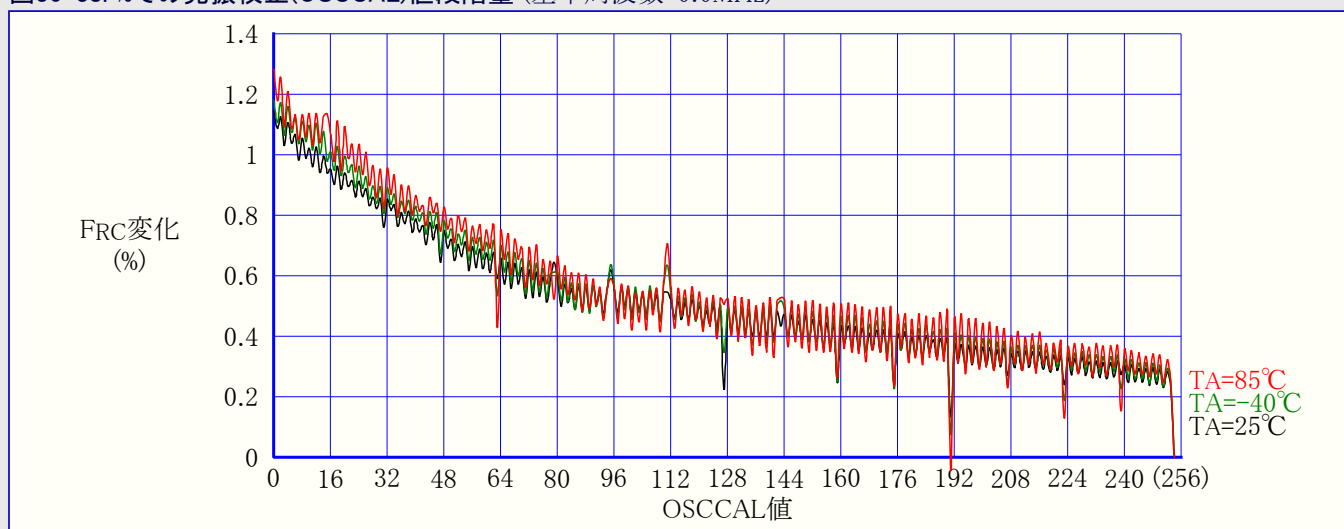


図30-38. %での発振校正(OSCCAL)値段階量 (基準周波数=0.0MHz)



30.10. 周辺機能部消費電流

図30-39. USB電圧調整器出力電圧 対 入力電圧 (無負荷)

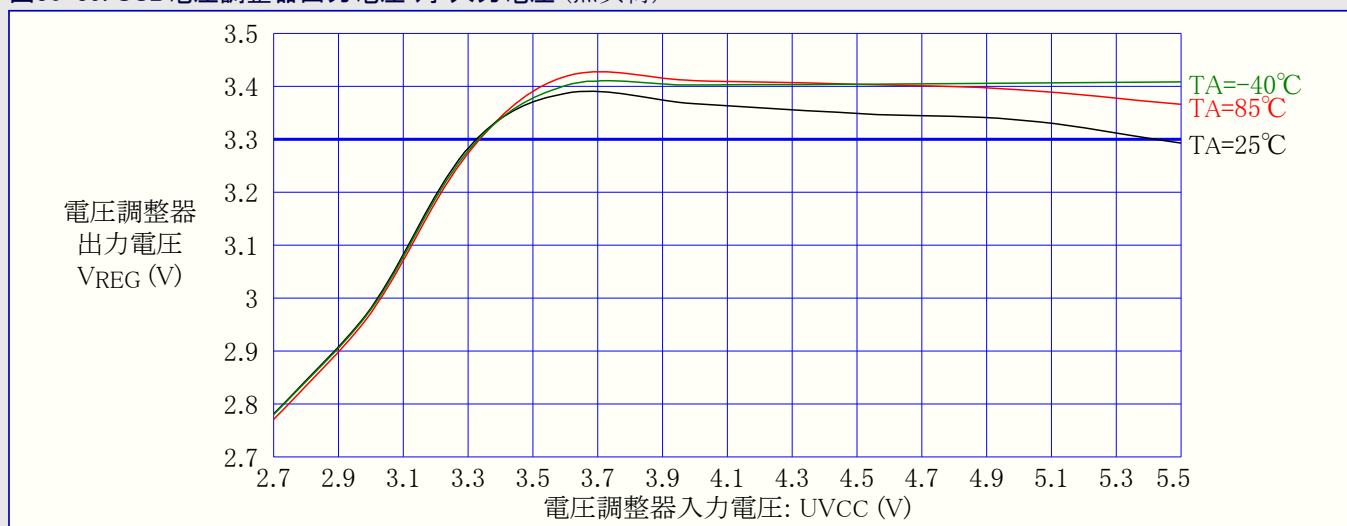


図30-40. USB電圧調整器出力電圧 対 入力電圧 (75Ω負荷)

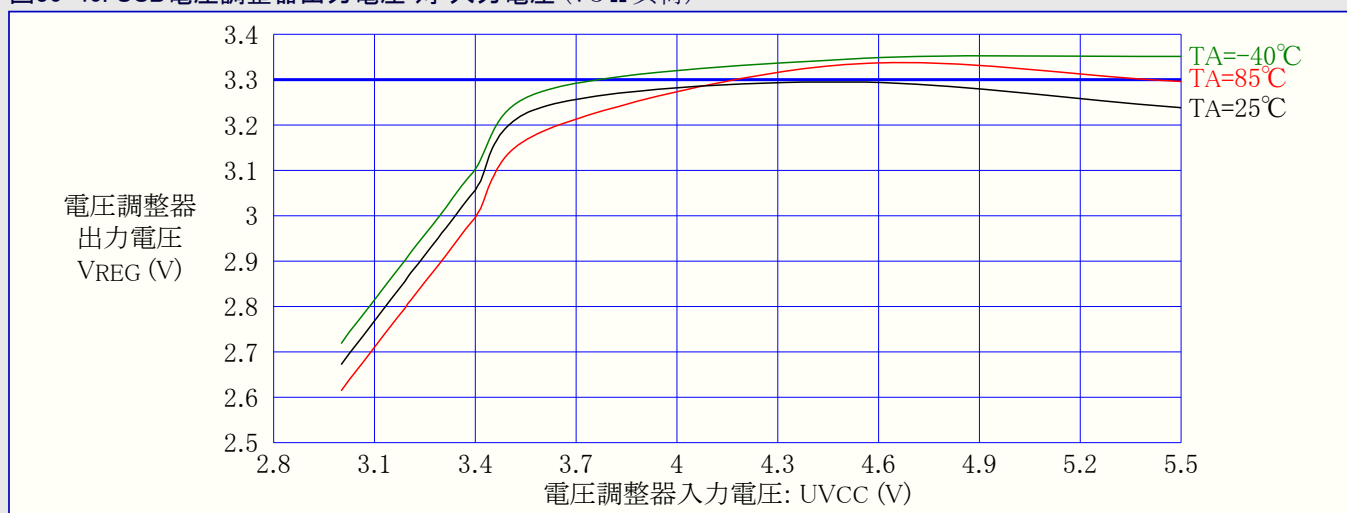


図30-41. A/D変換器内部基準電圧(VREF) 対 動作電圧

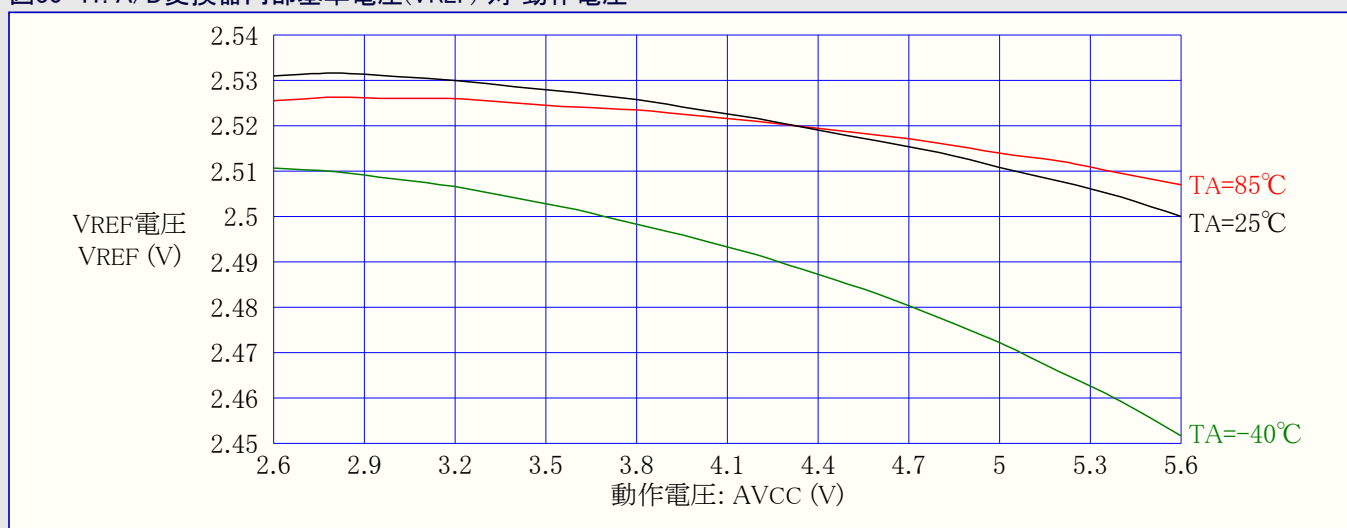
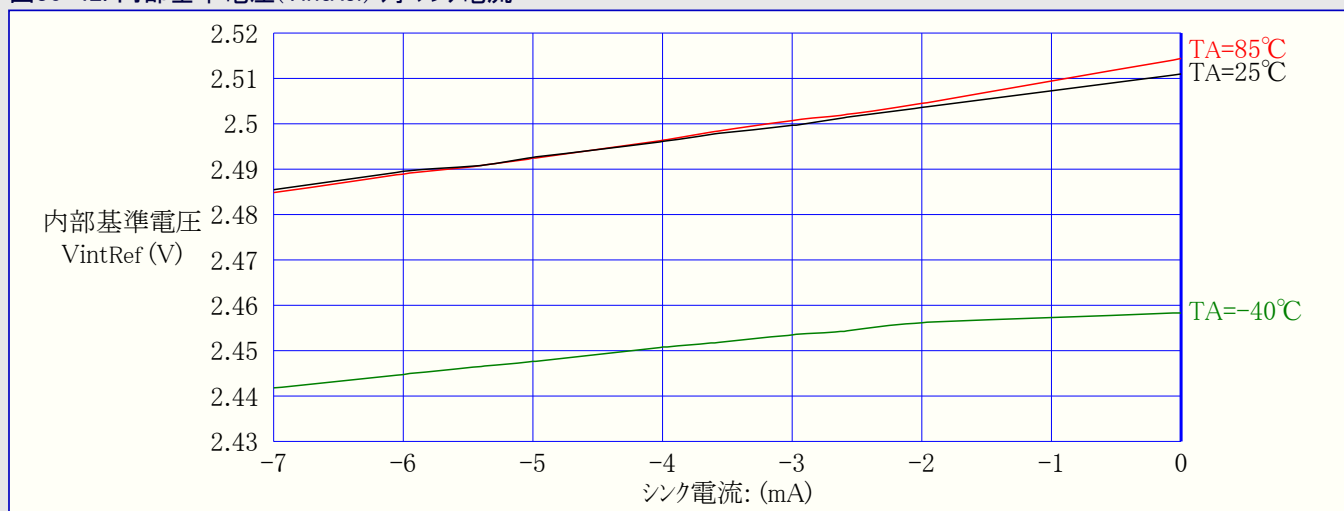
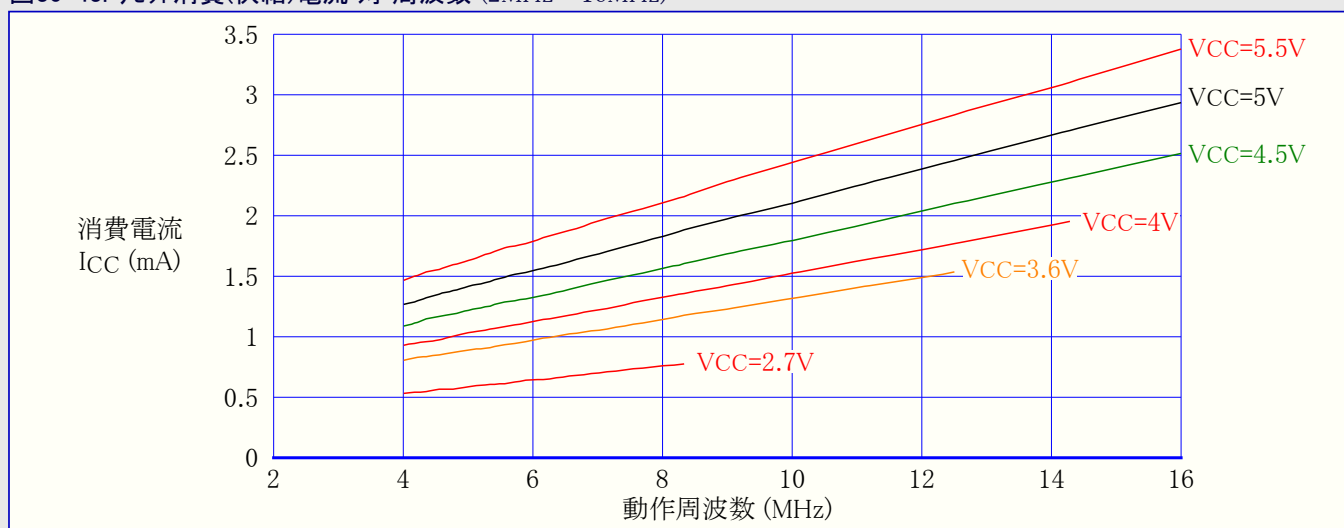


図30-42. 内部基準電圧(V_{intRef}) 対 シンク電流



30.11. リセット消費電流とリセット パルス幅

図30-43. リセット消費(供給)電流 対 周波数 (2MHz~16MHz)



31. レジスタ要約

拡張I/Oレジスタ領域 (1/3)

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									
(\$FE)	予約									
(\$FD)	予約									
(\$FC)	予約									
(\$FB)	予約									
(\$FA)	予約									
(\$F9)	予約									
(\$F8)	予約									
(\$F7)	予約									
(\$F6)	予約									
(\$F5)	予約									
(\$F4)	UEINT	-	EPINT6	EPINT5	EPINT4	EPINT3	EPINT2	EPINT1	EPINT0	191
(\$F3)	UEBCHX	-	-	-	-	-	EP-X バイト計数上位 (BYCT10~8)			191
(\$F2)	UEBCLX	エンドポイントX バイト計数下位 (BYCT7~0)								
(\$F1)	UEDATX	エンドポイントX データ (DAT7~0)								191
(\$F0)	UEIENX	FLERRE	NAKINE	-	NAKOUTE	RXSTPE	RXOUTE	STALLEDE	TXINE	190
(\$EF)	UESTA1X	-	-	-	-	-	CTRLDIR	CURRBK1	CURRBK0	189
(\$EE)	UESTA0X	CFGOK	OVERFI	UNDERFI	-	DTSEQ1	DTSEQ0	NBUSYBK1	NBUSYBK0	188
(\$ED)	UECFG1X	-	EPSIZE2	EPSIZE1	EPSIZE0	EPBK1	EPBK0	ALLOC	-	188
(\$EC)	UECFG0X	EPTYPE1	EPTYPE0	-	-	-	-	-	EPDIR	187
(\$EB)	UECONX	-	-	STALLRQ	STALLRQC	RSTDT	-	-	EPEN	187
(\$EA)	UERST	-	EPRST6	EPRST5	EPRST4	EPRST3	EPRST2	EPRST1	EPRST0	186
(\$E9)	UENUM	-	-	-	-	-	EPNUM2	EPNUM1	EPNUM0	186
(\$E8)	UEINTX	FIFOCON	NAKINI	RWAL	NAKOUTI	RXSTPI	RXOUTI	STALLEDI	TXINI	189
(\$E7)	予約									
(\$E6)	UDMFN	-	-	-	FNCERR	-	-	-	-	186
(\$E5)	UDFNUMH	-	-	-	-	-	装置フレーム番号上位 (FNUM10~8)			186
(\$E4)	UDFNUML	装置フレーム番号下位 (FNUM7~0)								
(\$E3)	UDADDR	ADDEN	装置アドレス (UADD6~0)							185
(\$E2)	UDIEN	-	UPRSME	EORSME	WAKEUPE	EORSTE	SOFE	-	SUSPE	185
(\$E1)	UDINT	-	UPRSMI	EORSMI	WAKEUPI	EORSTI	SOFI	-	SUSPI	184
(\$E0)	UDCON	-	-	-	-	RSTCPU	LSM	RMWKUP	DETACH	184
(\$DF)	予約									
(\$DE)	予約									
(\$DD)	予約									
(\$DC)	予約									
(\$DB)	予約									
(\$DA)	USBINT	-	-	-	-	-	-	-	VBUSTI	175
(\$D9)	USBSTA	-	-	-	-	-	-	ID	VBUS	174
(\$D8)	USBCON	USBE	-	FRZCLK	OTGPADE	-	-	-	VBUSTE	174
(\$D7)	UHWCON	-	-	-	-	-	-	-	UVREGE	174
(\$D6)	予約									
(\$D5)	予約									
(\$D4)	DT4	DT4H3	DT4H2	DT4H1	DT4H0	DT4L3	DT4L2	DT4L1	DT4L0	114
(\$D3)	予約									
(\$D2)	OCR4D	タイマ/カウンタ4 比較Dレジスタ								112
(\$D1)	OCR4C	タイマ/カウンタ4 比較Cレジスタ								112
(\$D0)	OCR4B	タイマ/カウンタ4 比較Bレジスタ								112
(\$CF)	OCR4A	タイマ/カウンタ4 比較Aレジスタ								112
(\$CE)	UDR1	USART1 データレジスタ								136
(\$CD)	UBRR1H	-	-	-	-	USART1 ホーレートレジスタ上位 (UBRR111~8)				139
(\$CC)	UBRR1L	USART1 ホーレートレジスタ下位 (UBRR17~0)								
(\$CB)	UCSR1D	-	-	-	-	-	-	CTSEN1	RTSEN1	138
(\$CA)	UCSR1C	UMSEL11	UMSEL10	UPM11	UPM10	USBS1	UCSZ11	UCSZ10	UCPOL1	137
(\$C9)	UCSR1B	RXCIE1	TXCIE1	UDRIE1	RXEN1	TXEN1	UCSZ12	RXB81	TXB81	137
(\$C8)	UCSR1A	RXC1	TXC1	UDRE1	FE1	DOR1	UPE1	U2X1	MPCM1	136
(\$C7)	CLKSTA	-	-	-	-	-	-	RCON	EXTON	24
(\$C6)	CLKSEL1	RCCKSEL3	RCCKSEL2	RCCKSEL1	RCCKSEL0	EXCKSEL3	EXCKSEL2	EXCKSEL1	EXCKSEL0	23
(\$C5)	CLKSEL0	RCSUT1	RCSUT0	EXSUT1	EXSUT0	RCE	EXTE	-	CLKS	23
(\$C4)	TCCR4E	TLOCK4	ENHC4	OC4OE5	OC4OE4	OC4OE3	OC4OE2	OC4OE1	OC4OE0	111
(\$C3)	TCCR4D	FPIE4	FPEN4	FPNC4	FPES4	FPAC4	FPF4	WGM41	WGM40	110
(\$C2)	TCCR4C	COM4A1S	COM4A0S	COM4B1S	COM4B0S	COM4D1	COM4D0	FOC4D	PWM4D	109
(\$C1)	TCCR4B	PWM4X	PSR4	DTPS41	DTPS40	CS43	CS42	CS41	CS40	108
(\$C0)	TCCR4A	COM4A1	COM4A0	COM4B1	COM4B0	FOC4A	FOC4B	PWM4A	PWM4B	106

拡張I/Oレジスタ領域 (2/3)

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
(\$BF)	TC4H	—	—	—	—	—	TC410	TC49	TC48	111
(\$BE)	TCNT4	タイマ/カウンタ4 (下位バイト)								111
(\$BD)	TWAMR	TWAM6	TWAM5	TWAM4	TWAM3	TWAM2	TWAM1	TWAM0	—	165
(\$BC)	TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	—	TWIE	163
(\$BB)	TWDR	2線直列インターフェースデータレジスタ								164
(\$BA)	TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE	164
(\$B9)	TWSR	TWS7	TWS6	TWS5	TWS4	TWS3	—	TWPS1	TWPS0	164
(\$B8)	TWBR	2線直列インターフェースビット速度レジスタ								163
(\$B7)	予約									
(\$B6)	予約									
(\$B5)	予約									
(\$B4)	予約									
(\$B3)	予約									
(\$B2)	予約									
(\$B1)	予約									
(\$B0)	予約									
(\$AF)	予約									
(\$AE)	予約									
(\$AD)	予約									
(\$AC)	予約									
(\$AB)	予約									
(\$AA)	予約									
(\$A9)	予約									
(\$A8)	予約									
(\$A7)	予約									
(\$A6)	予約									
(\$A5)	予約									
(\$A4)	予約									
(\$A3)	予約									
(\$A2)	予約									
(\$A1)	予約									
(\$A0)	予約									
(\$9F)	予約									
(\$9E)	予約									
(\$9D)	OCR3CH	タイマ/カウンタ3 比較レジスタ上位バイト								87
(\$9C)	OCR3CL	タイマ/カウンタ3 比較レジスタ下位バイト								
(\$9B)	OCR3BH	タイマ/カウンタ3 比較レジスタ上位バイト								87
(\$9A)	OCR3BL	タイマ/カウンタ3 比較レジスタ下位バイト								
(\$99)	OCR3AH	タイマ/カウンタ3 比較レジスタ上位バイト								87
(\$98)	OCR3AL	タイマ/カウンタ3 比較レジスタ下位バイト								
(\$97)	ICR3H	タイマ/カウンタ3 捕獲レジスタ上位バイト								88
(\$96)	ICR3L	タイマ/カウンタ3 捕獲レジスタ下位バイト								
(\$95)	TCNT3H	タイマ/カウンタ3 上位バイト								86
(\$94)	TCNT3L	タイマ/カウンタ3 下位バイト								
(\$93)	予約									
(\$92)	TCCR3C	FOC3A	—	—	—	—	—	—	—	85
(\$91)	TCCR3B	ICNC3	ICES3	—	WGM33	WGM32	CS32	CS31	CS30	84
(\$90)	TCCR3A	COM3A1	COM3A0	COM3B1	COM3B0	COM3C1	COM3C0	WGM31	WGM30	83
(\$8F)	予約									
(\$8E)	予約									
(\$8D)	OCR1CH	タイマ/カウンタ1 比較レジスタ上位バイト								86
(\$8C)	OCR1CL	タイマ/カウンタ1 比較レジスタ下位バイト								
(\$8B)	OCR1BH	タイマ/カウンタ1 比較レジスタ上位バイト								86
(\$8A)	OCR1BL	タイマ/カウンタ1 比較レジスタ下位バイト								
(\$89)	OCR1AH	タイマ/カウンタ1 比較レジスタ上位バイト								86
(\$88)	OCR1AL	タイマ/カウンタ1 比較レジスタ下位バイト								
(\$87)	ICR1H	タイマ/カウンタ1 捕獲レジスタ上位バイト								88
(\$86)	ICR1L	タイマ/カウンタ1 捕獲レジスタ下位バイト								
(\$85)	TCNT1H	タイマ/カウンタ1 上位バイト								86
(\$84)	TCNT1L	タイマ/カウンタ1 下位バイト								
(\$83)	予約									
(\$82)	TCCR1C	FOC1A	FOC1B	FOC1C	—	—	—	—	—	85
(\$81)	TCCR1B	ICNC1	ICES1	—	WGM13	WGM12	CS12	CS11	CS10	84
(\$80)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	COM1C1	COM1C0	WGM11	WGM10	83

拡張I/Oレジスタ領域 (3/3)

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
(\$7F)	DIDR1	–	–	–	–	–	–	–	AIN0D	193
(\$7E)	DIDR0	ADC7D	ADC6D	ADC5D	ADC4D	–	–	ADC1D	ADC0D	205
(\$7D)	DIDR2	–	–	ADC13D	ADC12D	ADC11D	ADC10D	ADC9D	ADC8D	205
(\$7C)	ADMUX	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	202
(\$7B)	ADCSRB	ADHSM	ACME	MUX5	–	ADTS3	ADTS2	ADTS1	ADTS0	192,203
(\$7A)	ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	203
(\$79)	ADCH	A/Dデータレジスタ上位バイト (ADC9~8またはADC9~2)								204
(\$78)	ADCL	A/Dデータレジスタ下位バイト (ADC7~0またはADC1,0)								
(\$77)	予約									
(\$76)	予約									
(\$75)	予約									
(\$74)	予約									
(\$73)	予約									
(\$72)	TIMSK4	OCIE4D	OCIE4A	OCIE4B	–	–	TOIE4	–	–	113
(\$71)	TIMSK3	–	–	ICIE3	–	OCIE3C	OCIE3B	OCIE3A	TOIE3	88
(\$70)	予約									
(\$6F)	TIMSK1	–	–	ICIE1	–	OCIE1C	OCIE1B	OCIE1A	TOIE1	88
(\$6E)	TIMSK0	–	–	–	–	–	OCIE0B	OCIE0A	TOIE0	68
(\$6D)	予約									
(\$6C)	予約									
(\$6B)	PCMSK0	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0	40
(\$6A)	EICRB	–	–	ISC61	ISC60	–	–	–	–	39
(\$69)	EICRA	ISC31	ISC30	ISC21	ISC20	ISC11	ISC10	ISC01	ISC00	38
(\$68)	PCICR	–	–	–	–	–	–	–	PCIE0	40
(\$67)	RCCTRL	–	–	–	–	–	–	–	RCFREQ	20
(\$66)	OSCCAL	内蔵RC発振器 発振校正値レジスタ								20
(\$65)	PRR1	PRUSB	–	–	PRTIM4	PRTIM3	–	–	PRUSART	29
(\$64)	PRR0	PRTWI	–	PRTIM0	–	PRTIM1	PRSPI	–	PRADC	28
(\$63)	予約									
(\$62)	予約									
(\$61)	CLKPR	CLKPCE	–	–	–	CLKPS3	CLKPS2	CLKPS1	CLKPS0	24
(\$60)	WDTCR	WDIF	WDIE	WDP3	WDCE	WDE	WDP2	WDP1	WDP0	35

注意: ● 将来のデバイスとの共通性のため、アクセスされる場合の予約ビットは0を書かれるべきです。予約した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。

- アドレス範囲\$00~\$1F内のI/OレジスタはSBIとCBI命令を使う直接ビットアクセスが可能です。これらのレジスタではSBISとSBIC命令を使うことによって単一ビット値が検査できます。
- いくつかの状態ビットはそれらへ論理1を書くことによって解除(0)されます。他の多くのAVRと異なり、CBIとSBI命令は指定ビットだけ操作し、故にこのような状態フラグを含むレジスタで使えます。CBIとSBI命令は\$00~\$1Fのレジスタだけで動作します。
- 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とST命令を使ってデータ空間としてI/Oレジスタをアドレス指定する時はこれらのアドレスに\$20が加算されなければなりません。ATmega16U4/32U4はINとOUT命令で予約した64位置で支援できるよりも多くの周辺部(機能)の複合マイクロコントローラです。SRAM(データ空間)内の拡張I/O空間はST/STS/STDとLD/LDS/LDD命令だけが使えます。

標準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	7
\$3E (\$5E)	SPH	–	–	–	–	(SP11)	SP10	SP9	SP8	8
\$3D (\$5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	
\$3C (\$5C)	予約									
\$3B (\$5B)	予約									
\$3A (\$5A)	予約									
\$39 (\$59)	予約									
\$38 (\$58)	予約									
\$37 (\$57)	SPMCSR	SPMIE	RWWSB	SIGRD	RWWSRE	BLBSET	PGWRT	PGERS	SPMEN	219
\$36 (\$56)	予約									
\$35 (\$55)	MCUCR	JTD	–	–	PUD	–	–	IVSEL	IVCE	212,53,37
\$34 (\$54)	MCUSR	–	–	USBRF	JTRF	WDRF	BORF	EXTRF	PORF	212,34
\$33 (\$53)	SMCR	–	–	–	–	SM2	SM1	SM0	SE	28
\$32 (\$52)	PLLFRQ	PINMUX	PLLUSB	PLLTm1	PLLTm0	PDIV3	PDIV2	PDIV1	PDIV0	25
\$31 (\$51)	OCDR/ MONDR	OCDR7/ MOND7	OCDR6/ MOND6	OCDR5/ MOND5	OCDR4/ MOND4	OCDR3/ MOND3	OCDR2/ MOND2	OCDR1/ MOND1	OCDR0/ MOND0	208
\$30 (\$50)	ACSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	193
\$2F (\$4F)	予約									
\$2E (\$4E)	SPDR	SPI データレジスタ								121
\$2D (\$4D)	SPSR	SPIF	WCOL	–	–	–	–	–	SPI2X	120
\$2C (\$4C)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	120
\$2B (\$4B)	GPOR2	汎用I/Oレジスタ2								16
\$2A (\$4A)	GPOR1	汎用I/Oレジスタ1								16
\$29 (\$49)	PLLCSR	–	–	–	PINDIV	–	–	PLLE	PLOCK	24
\$28 (\$48)	OCR0B	タイマ/カウンタ0 比較Aレジスタ								67
\$27 (\$47)	OCR0A	タイマ/カウンタ0 比較Aレジスタ								67
\$26 (\$46)	TCNT0	タイマ/カウンタ0								67
\$25 (\$45)	TCCR0B	FOC0A	FOC0B	–	–	WGM02	CS02	CS01	CS00	66
\$24 (\$44)	TCCR0A	COM0A1	COM0A0	COM0B1	COM0B0	–	–	WGM01	WGM00	65
\$23 (\$43)	GTCCR	TSM	–	–	–	–	–	–	PSRSYNC	56
\$22 (\$42)	EEARH	–	–	–	–	–	–	(EEAR9)	EEAR8	13
\$21 (\$41)	EEARL	EEPROMアドレスレジスタ下位バイト (EEAR7~0)								
\$20 (\$40)	EEDR	EEPROMデータレジスタ								13
\$1F (\$3F)	EECR	–	–	EEPM1	EEPM0	EERIE	EEMPE	EEPE	EERE	13
\$1E (\$3E)	GPOR0	汎用I/Oレジスタ0								16
\$1D (\$3D)	EMSK	–	INT6	–	–	INT3	INT2	INT1	INT0	39
\$1C (\$3C)	EIFR	–	INTF6	–	–	INTF3	INTF2	INTF1	INTF0	39
\$1B (\$3B)	PCIFR	–	–	–	–	–	–	–	PCIF0	40
\$1A (\$3A)	予約									
\$19 (\$39)	TIFR4	OCF4D	OCF4A	OCF4B	–	–	TOV4	–	–	113
\$18 (\$38)	TIFR3	–	–	ICF3	–	OCF3C	OCF3B	OCF3A	TOV3	89
\$17 (\$37)	予約									
\$16 (\$36)	TIFR1	–	–	ICF1	–	OCF1C	OCF1B	OCF1A	TOV1	89
\$15 (\$35)	TIFR0	–	–	–	–	–	OCF0B	OCF0A	TOV0	68
\$14 (\$34)	予約									
\$13 (\$33)	予約									
\$12 (\$32)	予約									
\$11 (\$31)	PORTF	PORTF7	PORTF6	PORTF5	PORTF4	–	–	PORTF1	PORTF0	54
\$10 (\$30)	DDRF	DDF7	DDF6	DDF5	DDF4	–	–	DDF1	DDF0	54
\$0F (\$2F)	PINF	PINF7	PINF6	PINF5	PINF4	–	–	PINF1	PINF0	54
\$0E (\$2E)	PORTE	–	PORTE6	–	–	–	PORTE2	–	–	54
\$0D (\$2D)	DDRE	–	DDE6	–	–	–	DDE2	–	–	54
\$0C (\$2C)	PINE	–	PINE6	–	–	–	PINE2	–	–	54
\$0B (\$2B)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	53
\$0A (\$2A)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	53
\$09 (\$29)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	53
\$08 (\$28)	PORTC	PORTC7	PORTC6	–	–	–	–	–	–	53
\$07 (\$27)	DDRC	DDC7	DDC6	–	–	–	–	–	–	53
\$06 (\$26)	PINC	PINC7	PINC6	–	–	–	–	–	–	53
\$05 (\$25)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	53
\$04 (\$24)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	53
\$03 (\$23)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	53
\$02 (\$22)	予約									
\$01 (\$21)	予約									
\$00 (\$20)	予約									

注: () 付きのビットはATmega16U4で無効です。

(訳注) 原書本位置の注意は前頁に移動しました。

32. 命令要約

ニーモニック	オペランド	意味	動作	フラグ	クロック
算術、論理演算命令					
ADD	Rd,Rr	汎用レジスタ間の加算	$Rd \leftarrow Rd + Rr$	I,T,H,S,V,N,Z,C	1
ADC	Rd,Rr	キャリーを含めた汎用レジスタ間の加算	$Rd \leftarrow Rd + Rr + C$	I,T,H,S,V,N,Z,C	1
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,1	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,0,0,0,1,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
分岐命令					
RJMP	k	相対無条件分岐	$PC \leftarrow PC + k + 1$	I,T,H,S,V,N,Z,C	2
IJMP		レジスタ間無条件分岐	$PC \leftarrow Z$	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		レジスタ間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, $PC \leftarrow PC + 2 \text{ or } 3$	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, $PC \leftarrow PC + 2 \text{ or } 3$	I,T,H,S,V,N,Z,C	1/2,3
SBRS	Rr,b	汎用レジスタのビットが設定(1)でスキップ*	$Rr(b)=1$ なら, $PC \leftarrow PC + 2 \text{ or } 3$	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, $PC \leftarrow PC + 2 \text{ or } 3$	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, $PC \leftarrow PC + 2 \text{ or } 3$	I,T,H,S,V,N,Z,C	1/2,3
BRBS	s,k	ステータスフラグが設定(1)で分岐	$SREG(s)=1$ なら, $PC \leftarrow PC + K + 1$	I,T,H,S,V,N,Z,C	1/2
BRBC	s,k	ステータスフラグが解除(0)で分岐	$SREG(s)=0$ なら, $PC \leftarrow PC + K + 1$	I,T,H,S,V,N,Z,C	1/2
BREQ	k	一致で分岐	$Z=1$ なら, $PC \leftarrow PC + K + 1$	I,T,H,S,V,N,Z,C	1/2
BRNE	k	不一致で分岐	$Z=0$ なら, $PC \leftarrow PC + K + 1$	I,T,H,S,V,N,Z,C	1/2
BRCS	k	キャリーフラグが設定(1)で分岐	$C=1$ なら, $PC \leftarrow PC + K + 1$	I,T,H,S,V,N,Z,C	1/2
BRCC	k	キャリーフラグが解除(0)で分岐	$C=0$ なら, $PC \leftarrow PC + K + 1$	I,T,H,S,V,N,Z,C	1/2
BRSH	k	符号なしの \geq で分岐	$C=0$ なら, $PC \leftarrow PC + K + 1$	I,T,H,S,V,N,Z,C	1/2
BRLO	k	符号なしの $<$ で分岐	$C=1$ なら, $PC \leftarrow PC + K + 1$	I,T,H,S,V,N,Z,C	1/2
BRMI	k	-(マイナス)で分岐	$N=1$ なら, $PC \leftarrow PC + K + 1$	I,T,H,S,V,N,Z,C	1/2
BRPL	k	+(プラス)で分岐	$N=0$ なら, $PC \leftarrow PC + K + 1$	I,T,H,S,V,N,Z,C	1/2
BRGE	k	符号付きの \geq で分岐	$(N \text{ EOR } V)=0$ なら, $PC \leftarrow PC + K + 1$	I,T,H,S,V,N,Z,C	1/2
BRLT	k	符号付きの $<$ で分岐	$(N \text{ EOR } V)=1$ なら, $PC \leftarrow PC + K + 1$	I,T,H,S,V,N,Z,C	1/2
BRHS	k	ハーフキャリーフラグが設定(1)で分岐	$H=1$ なら, $PC \leftarrow PC + K + 1$	I,T,H,S,V,N,Z,C	1/2
BRHC	k	ハーフキャリーフラグが解除(0)で分岐	$H=0$ なら, $PC \leftarrow PC + K + 1$	I,T,H,S,V,N,Z,C	1/2
BRTS	k	一時フラグが設定(1)で分岐	$T=1$ なら, $PC \leftarrow PC + K + 1$	I,T,H,S,V,N,Z,C	1/2
BRTC	k	一時フラグが解除(0)で分岐	$T=0$ なら, $PC \leftarrow PC + K + 1$	I,T,H,S,V,N,Z,C	1/2
BRVS	k	2の補数溢れフラグが設定(1)で分岐	$V=1$ なら, $PC \leftarrow PC + K + 1$	I,T,H,S,V,N,Z,C	1/2
BRVC	k	2の補数溢れフラグが解除(0)で分岐	$V=0$ なら, $PC \leftarrow PC + K + 1$	I,T,H,S,V,N,Z,C	1/2
BRIE	k	割り込み許可で分岐	$I=1$ なら, $PC \leftarrow PC + K + 1$	I,T,H,S,V,N,Z,C	1/2
BRID	k	割り込み禁止で分岐	$I=0$ なら, $PC \leftarrow PC + K + 1$	I,T,H,S,V,N,Z,C	1/2

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
データ移動命令					
MOV	Rd,Rr	汎用レジスタ間の複写	$Rd \leftarrow Rr$	I,T,H,S,V,N,Z,C	1
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X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
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
ビット関係命令					
SBI	P,b	I/Oレジスタのビット設定(1)	$I/O(P,b) \leftarrow 1$	I,T,H,S,V,N,Z,C	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	ニフル(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)	$SREG(s) \leftarrow 1$	1,1,1,1,1,1,1,1	1
BCLR	s	ステータスレジスタのビット解除(0)	$SREG(s) \leftarrow 0$	0,0,0,0,0,0,0,0	1
BST	Rr,b	汎用レジスタのビットを一時フラグへ移動	$T \leftarrow Rr(b)$	I,T,H,S,V,N,Z,C	1
BLD	Rd,b	一時フラグを汎用レジスタのビットへ移動	$Rd(b) \leftarrow T$	I,T,H,S,V,N,Z,C	1
SEC		キャリーフラグを設定(1)	$C \leftarrow 1$	I,T,H,S,V,N,Z,1	1
CLC		キャリーフラグを解除(0)	$C \leftarrow 0$	I,T,H,S,V,N,Z,0	1
SEN		負フラグを設定(1)	$N \leftarrow 1$	I,T,H,S,V,1,Z,C	1
CLN		負フラグを解除(0)	$N \leftarrow 0$	I,T,H,S,V,0,Z,C	1
SEZ		ゼロフラグを設定(1)	$Z \leftarrow 1$	I,T,H,S,V,N,1,C	1
CLZ		ゼロフラグを解除(0)	$Z \leftarrow 0$	I,T,H,S,V,N,0,C	1
SEI		全割り込み許可	$I \leftarrow 1$	I,T,H,S,V,N,Z,C	1
CLI		全割り込み禁止	$I \leftarrow 0$	0,T,H,S,V,N,Z,C	1
SES		符号フラグを設定(1)	$S \leftarrow 1$	I,T,H,1,V,N,Z,C	1
CLS		符号フラグを解除(0)	$S \leftarrow 0$	I,T,H,0,V,N,Z,C	1
SEV		2の補数溢れフラグを設定(1)	$V \leftarrow 1$	I,T,H,S,1,N,Z,C	1
CLV		2の補数溢れフラグを解除(0)	$V \leftarrow 0$	I,T,H,S,0,N,Z,C	1
SET		一時フラグを設定(1)	$T \leftarrow 1$	I,1,H,S,V,N,Z,C	1
CLT		一時フラグを解除(0)	$T \leftarrow 0$	I,0,H,S,V,N,Z,C	1
SEH		ハーフキャリーフラグを設定(1)	$H \leftarrow 1$	I,T,H,1,S,V,N,Z,C	1
CLH		ハーフキャリーフラグを解除(0)	$H \leftarrow 0$	I,T,0,S,V,N,Z,C	1
MCU制御命令					
NOP		無操作		I,T,H,S,V,N,Z,C	1
SLEEP		休止形態開始	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

33. 注文情報

速度(MHz)	電源電圧	注文符号	既定発振器	外周器	動作範囲
16	2.7～5.5V	ATmega16U4-16AU	外部クリスタル用	44ML	工業用 (-40℃～85℃)
		ATmega16U4RC-16AU (注1,2,3)	校正付き内蔵RC		
		ATmega16U4-16MU	外部クリスタル用	44PW	
		ATmega16U4RC-16MU (注1,2,3)	校正付き内蔵RC		
16	2.7～5.5V	ATmega32U4-16AU	外部クリスタル用	44ML	工業用 (-40℃～85℃)
		ATmega32U4RC-16AU (注1,2,3)	校正付き内蔵RC		
		ATmega32U4-16MU	外部クリスタル用	44PW	
		ATmega32U4RC-16MU (注1,2,3)	校正付き内蔵RC		

注1: 内蔵RC発振器からUSBを走行するより多くの情報については、「AVR291: Atmel ATmega32U4RCでのUSB低速(Low-speed)用8MHz内部発振器校正」応用記述を調べてください。

注2: 内蔵RC発振器からのUSB動作は0~40°C間でだけ保証されます。

注3: これらのデバイスは予め書かれたUSBブートローダなしで出荷されます。

外周器形式

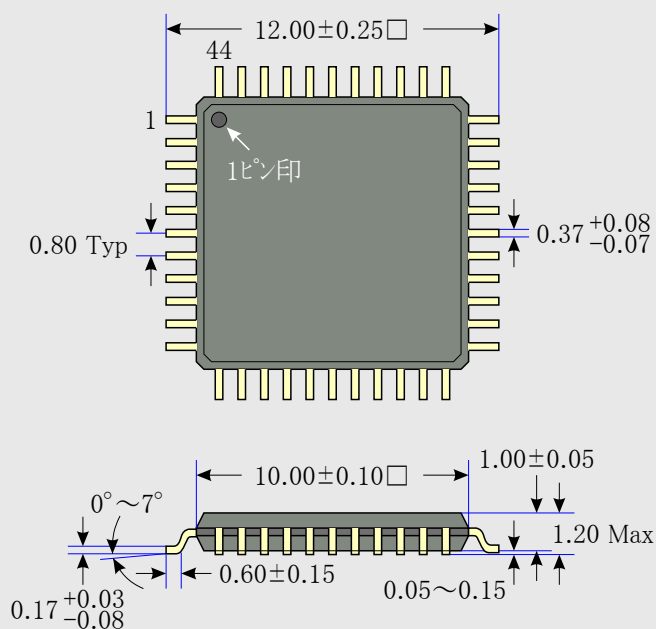
TQFP44 (44ML)	44リード 10×10mm, 1.0mm厚, 0.8mmピッチ 薄型プラスチック4方向平板外周器 (TQFP)
QFN44 (44PW)	44パッド 7×7mm, 0.5mmピッチ 4方向平板リードなし外周器 (QFN)

34. 外周器情報

34.1. TQFP44

44リード 0.8mmピッチ プラスチック4方向平板外周器 (TQFP)

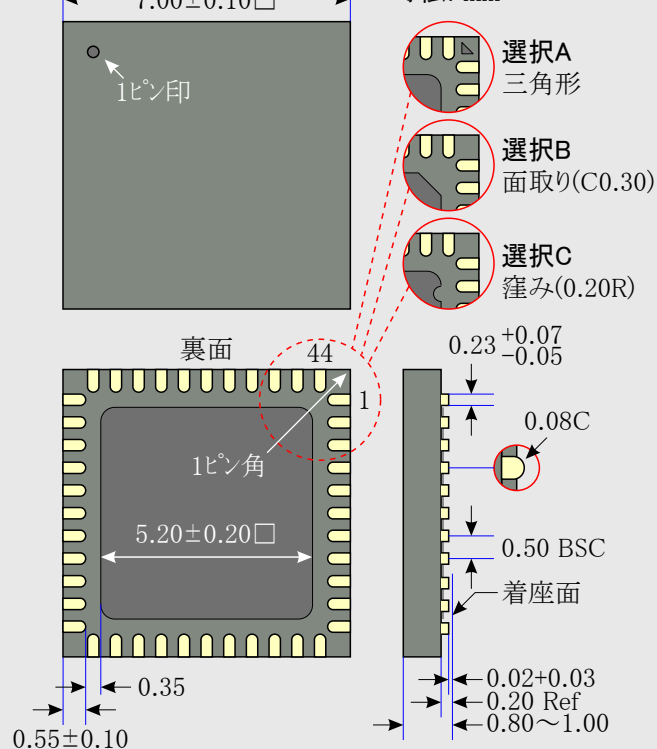
寸法: mm
JEDEC規格 MS-026 ACB



34.2. QFN44

44パッド 0.5mmピッチ 4方向平板リードなし外周器 (QFN)

JEDEC規格 MO-220変形VKKD-1
寸法: mm



35. 障害情報

この章の改訂番号はATmega16U4/32U4デバイスの改訂版を参照してください。

ATmega16U4/ATmega32U4改訂C 試供されていません。

● TWI許可時のTWIピンでのスパイク	16U4-A/B/D/E, 32U4-A/B/D/E
● 休止形態での高消費電流	16U4-A/B/D/E, 32U4-A/B/D/E
● タイマ/カウンタ4の11ビット増強PWM動作の問題	16U4-A/B/D, 32U4-A/B/D
● VBUSTI割り込みルーチンに対する不正なCPUの動き	16U4-A/B, 32U4-A/B
● OCR4A/B/DのMSBが11ビット増強PWMで書き込み専用	16U4-E, 32U4-E
● 余分な電力消費	16U4-A, 32U4-A
● 校正付き内蔵RC発振器起動の問題	16U4-A, 32U4-A
● 校正付き内蔵RC発振器校正の問題	16U4-A, 32U4-A

1. TWI許可時のTWIピンでのスパイク (16U4-A/B/D/E, 32U4-A/B/D/E)

TWIが許可される時にSDAとSCLピンで100nsの負のスパイクが発生します。

対策/対処

TWI網の他の接続点に対してATmega16U4/32U4を最初に許可してください。

2. 休止形態での高消費電流 (16U4-A/B/D/E, 32U4-A/B/D/E)

保留割り込みが選んだ休止形態からデバイスを起動復帰し得ない場合、SEI命令直後のSLEEP命令実行時の休止中、消費電流が増加するでしょう。

対策/対処

休止移行に先立って休止形態からデバイスの起動復帰に使われない割り込みは禁止されるべきです。

3. タイマ/カウンタ4の11ビット増強PWM動作 (16U4-A/B/D, 32U4-A/B/D)

タイマ/カウンタ4の11ビット増強動作は機能しません。

対策/対処

ありません。

4. VBUSTI割り込みルーチンに対する不正なCPUの動き (16U4-A/B, 32U4-A/B)

CPUコアはVBUSTI割り込み要求フラグに関連する割り込みベクタを不正に実行するかもしれません。

対策/対処

この割り込みを許可しないでください。ファームウェアはVBUSTIのポーリングによってこのUSB事象を処理しなければなりません。

5. OCR4A/B/DのMSBが11ビット増強PWMで書き込み専用 (16U4-E, 32U4-E)

11ビット増強PWM動作に於いてOCR4A/B/DのMSBは書き込み専用です。OCR4A/B/Dの読み込みは常にMSB位置に0を返します。

対策/対処

ありません。

6. 余分な電力消費 (16U4-A, 32U4-A)

代表的な消費電力はパワーダウン動作で約30μA増加されます。

対策/対処

ありません。

7. 校正付き内蔵RC発振器起動の問題 (16U4-A, 32U4-A)

デバイスが内蔵RC発振器での開始に構成設定されている時にこの発振器が電源ON後に正しく始まらないかもしれません。

対策/対処

デバイスを内蔵RC発振器での開始に構成設定してはなりません(デバイス既定構成設定は外部クリスタル用発振器での開始です(非RC製品))。

8. 校正付き内蔵RC発振器校正の問題 (16U4-A, 32U4-A)

校正付き内蔵RC発振器既定周波数は8MHzよりも低くなるかもしれません。

対策/対処

デバイスは内蔵RC発振器周波数が既定の8MHz目的周波数に可能な限り近くなるように設定されています。

36. 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。

36.1. 7766A – 2008年3月

1. 初版

36.2. 7766B – 2008年7月

1. ATmega16U4デバイス追加
2. 「障害情報」項目作成とATmega16U4追加
3. 90頁の「10ビット高速タイマ/カウンタ」で非同期記述を更新

36.3. 7766C – 2008年11月

1. 「障害情報」項を更新

36.4. 7766D – 2009年1月

1. 1頁の「特徴」でメモリ項目を更新
2. 5頁に「資料」項を追加
3. 5頁に「データ保持力」項を追加
4. 275頁の「注文情報」を更新

36.5. 7766E – 2010年4月

1. 1頁の「特徴」を更新
2. 166頁の「特徴」を更新
3. 169頁の図21-9を更新
4. 171頁の21.8.項を更新
5. 194頁の「特徴」を更新
6. 214頁の「ATmega16U4/32U4の境界走査順」を更新
7. 227頁の表28-5を更新
8. 248頁の「電気的特性」を更新
9. 250頁の図29-2を更新
10. 254頁から「代表特性」を追加
11. 275頁の「注文情報」を更新
12. 276頁の「障害情報」を更新

36.6. 7766F – 2010年11月

1. 192頁の「ADCSRB – A/D変換制御/状態レジスタB」を更新、203頁のように定義
2. 図更新によって275頁の「QFN44」を更新
3. 新しいAtmelの品質様式指針に従って最終頁を更新

36.7. 7766G – 2014年2月

1. 115頁で「比較出力変調器(COM1C0A)」の「説明」を更新。論理積(AND)と論理和(OR)がPORTB7に基づいて実行されることを詳述。
2. 138頁の「UCSRnD – USART制御/状態レジスタD」を更新。「ビット7~2 – 予約」は読み出し専用。
3. 168頁の「クリスタルなし動作」を更新。温度範囲を「0~40°C内」に変更。
4. 192頁の「ADCSRB – A/D変換制御/状態レジスタB」のMUXビットをR/Wに変更
5. 204頁の表24-6を更新。起動元:タイマ/カウンタ0比較一致をタイマ/カウンタ0比較A一致に更新。
6. 248頁の「DC特性」を更新。ICC/消費電流に16MHz, VCC=5Vでの活動動作が最大27mAを追加。
7. 269頁の「レジスタ要約」を更新。アドレス \$CBにUCSR1Dを追加。
8. 外圍器図更新によって275頁の「TQFP44」と「QFN44」を置換
9. 新しいAtmelの品質様式指針(新ロゴ)に従って最終頁を更新 (訳注:本書は全頁に渡って新雛形を適用)

36.8. 7766H – 2014年9月

1. 表紙で暫定を削除
2. 1頁の「特徴」で既定USBブートローダで予め書かれたデバイスの記述を更新
3. 19頁の表6-3.と20頁の表6-6.で周波数範囲の脚注を削除
4. 35頁の表8-4.で値更新と脚注削除
5. 99頁の「位相/周波数基準PWM動作」の第1節を修正
6. 249頁の表29-2.でVCC=1.5~5.5Vの列を削除
7. 249頁の表29-2.に対する脚注を変更
8. 252頁の表29-5.で上昇/下降時間に対して最大値を追加
9. 275頁の「33. 注文情報」でRC部品番号に対して3つの脚注を追加
10. 2014年5月からの新データシート雛形で更新
11. 新雛形に従って様々な修正が行われました。
12. 最終頁に商標が追加されました。

36.9. 7766I – 2015年7月

1. 内容全体を通してAtmel品質を適用して内容进行评估
2. 25頁のPLLFRQ記述を更新。説明の一部が欠落していました。

36.10. 7766J – 2016年4月

1. 226頁の「メモリプログラミング」: ATmega16U4とATmega32U4に対するフラッシュメモリとEEPROMのページ数とページ内語数を更新。
230頁の表28-11.と表28-12.を参照してください。

目次

特徴	1	9. 割り込み	36
1. ピン配置	2	9.1. ATmega16U4/32U4の割り込みベクタ	36
2. 概要	3	9.2. ベクタ移動用レジスタ	37
2.1. 構成図	3	10. 外部割り込み	38
2.2. ピン説明	4	10.1. ピン変化割り込みタイミング	38
3. 諸注意	5	10.2. 外部割込み用レジスタ	38
3.1. お断り	5	11. 入出力ポート	41
3.2. 資料	5	11.1. 序説	41
3.3. コード例	5	11.2. 標準デジタル入出力としてのポート	41
3.4. データ保持力	5	11.3. 交換ポート機能	44
4. AVR CPU コア	6	11.4. I/Oポート用レジスタ	53
4.1. 序説	6	12. タイマ/カウンタ0,1,3の前置分周器	55
4.2. 構造概要	6	12.1. 内部クロック	55
4.3. ALU (Arithmetic Logic Unit)	6	12.2. 前置分周器リセット	55
4.4. ステータス レジスタ	7	12.3. 外部クロック	55
4.5. 汎用レジスタ ファイル	8	12.4. 同期系タイマ/カウンタ前置分周器用レジスタ	56
4.6. スタック ポインタ	8	13. 8ビット タイマ/カウンタ0 (PWM)	57
4.7. 命令実行タイミング	9	13.1. 概要	57
4.8. リセットと割り込みの扱い	9	13.2. タイマ/カウンタのクロック	58
5. AVRのメモリ	11	13.3. 計数器部	58
5.1. 実装書き換え可能なプログラム用フラッシュ メモリ	12	13.4. 比較出力部	59
5.2. データ用SRAMメモリ	12	13.5. 比較一致出力部	60
5.3. データ用EEPROMメモリ	13	13.6. 動作種別	61
5.4. I/Oメモリ (レジスタ)	16	13.7. タイマ/カウンタのタイミング	64
6. システム クロックとクロック選択	17	13.8. 8ビット タイマ/カウンタ0用レジスタ	65
6.1. クロック系統とその配給	17	14. 16ビット タイマ/カウンタ1と3	69
6.2. クロック元	18	14.1. 概要	69
6.3. クリスタル用低電力発振器	19	14.2. 16ビット レジスタのアクセス	71
6.4. 低周波数クリスタル用発振器	19	14.3. タイマ/カウンタのクロック	73
6.5. 校正付き内蔵RC発振器	20	14.4. 計数器部	73
6.6. 外部クロック信号	21	14.5. 捕獲入力部	74
6.7. クロック切り替え	21	14.6. 比較出力部	75
6.8. システム クロック出力緩衝部	22	14.7. 比較一致出力部	77
6.9. システム クロック前置分周器	22	14.8. 動作種別	78
6.10. PLL	22	14.9. タイマ/カウンタのタイミング	82
6.11. クロック切り替え方法	22	14.10. 16ビット タイマ/カウンタ1,3用レジスタ	83
6.12. クロック関係用レジスタ	23	15. 10ビット高速タイマ/カウンタ4	90
7. 電力管理と休止形態	26	15.1. 特徴	90
7.1. アイドル動作	26	15.2. 概要	90
7.2. A/D変換雑音低減動作	26	15.3. 計数器部	92
7.3. パワーダウン動作	26	15.4. 比較出力部	93
7.4. パワーセーブ動作	26	15.5. 沈黙時間生成器	94
7.5. スタンバイ動作	26	15.6. 比較一致出力部	95
7.6. 拡張スタンバイ動作	26	15.7. 同期更新	96
7.7. 電力削減(電力削減レジスタ)	27	15.8. 動作種別	97
7.8. 消費電力の最小化	27	15.9. タイマ/カウンタのタイミング	102
7.9. 電力管理用レジスタ	28	15.10. 障害保護部	103
8. システム制御とリセット	30	15.11. 10ビット レジスタのアクセス	104
8.1. AVRのリセット	30	15.12. 10ビット タイマ/カウンタ4用レジスタ	106
8.2. リセット元	30	16. 比較出力変調器 (COM1C0A)	115
8.3. 電源ONリセット	30	16.1. 概要	115
8.4. 外部リセット	31	16.2. 説明	115
8.5. 低電圧(ブラウンアウト)検出リセット	31	17. SPI (直列周辺インターフェース)	116
8.6. ウォッチドッグ リセット	31	17.1. SSピンの機能	118
8.7. USBリセット	31	17.2. データ転送形式	119
8.8. 内部基準電圧	32	17.3. SPI用レジスタ	120
8.9. ウォッチドッグ タイマ	32	18. USART	122
8.10. リセット関係用レジスタ	34		

18.1.	概要	122	22.17.	装置割り込み	183
18.2.	クロック生成	123	22.18.	USB装置一般用レジスタ	184
18.3.	フレーム形式	124	22.19.	USB装置エンドポイント用レジスタ	186
18.4.	USARTの初期化	125	23.	アナログ比較器	192
18.5.	USARTのデータ送信	126	23.1.	アナログ比較器入力選択	192
18.6.	USARTのデータ受信	127	23.2.	アナログ比較器用レジスタ	192
18.7.	非同期受信	130	24.	A/D変換器	194
18.8.	複数プロセッサ通信動作	132	24.1.	特徴	194
18.9.	ハードウェア流れ(フロー)制御	133	24.2.	操作	195
18.10.	ボーレート設定例	134	24.3.	変換の開始	195
18.11.	USART用レジスタ	136	24.4.	前置分周と変換タイミング	196
19.	USARTでのSPI動作	140	24.5.	チャネル変更と基準電圧選択	198
19.1.	概要	140	24.6.	温度感知器	199
19.2.	クロック生成	140	24.7.	雑音低減機能	200
19.3.	データ転送形式	140	24.8.	A/D変換の結果	201
19.4.	フレーム形式	141	24.9.	A/D変換用レジスタ	202
19.5.	データ転送	142	25.	JTAGインターフェースと内蔵デバッグ機能	206
19.6.	USARTでのMSPIMとSPIの比較	143	25.1.	特徴	206
19.7.	MSPIMでのUSART用レジスタ	144	25.2.	概要	206
20.	2線直列インターフェース (TWI)	146	25.3.	検査入出力ポート (TAP:Test Access Port)	207
20.1.	特徴	146	25.4.	TAP制御器	207
20.2.	2線直列インターフェース バスの定義	146	25.5.	境界走査チェーン(Boundary-Scan Chain)の使用	208
20.3.	データ転送とフレーム形式	146	25.6.	内蔵デバッグ機能の使用	208
20.4.	複数主装置バスシステムの調停と同期	148	25.7.	内蔵デバッグ特殊JTAG命令	208
20.5.	TWI部の概要	149	25.8.	内蔵デバッグ関連I/Oメモリ内のレジスタ	208
20.6.	TWIの使用法	151	25.9.	JTAGプログラミング能力の使用	209
20.7.	転送種別	153	25.10.	参考文献	209
20.8.	複数主装置システムでのバス競合と調停	162	26.	IEEE1149.1(JTAG)境界走査	210
20.9.	TWI用レジスタ	163	26.1.	特徴	210
21.	USB制御器	166	26.2.	概要	210
21.1.	特徴	166	26.3.	データレジスタ	210
21.2.	構成図	166	26.4.	境界走査(Boundary-Scan)用JTAG命令	211
21.3.	代表的な応用実装	166	26.5.	境界走査関連I/Oメモリ内のレジスタ	212
21.4.	クリスタルなし動作	168	26.6.	境界走査チェーン(Boundary-Scan Chain)	213
21.5.	設計の指針	168	26.7.	ATmega16U4/32U4の境界走査順	214
21.6.	一般的な動作	168	26.8.	境界走査記述言語(BSDL)ファイル	214
21.7.	電力動作種別	170	27.	ブートローダ支援 - RWW 自己プログラミング	215
21.8.	速度制御	171	27.1.	特徴	215
21.9.	メモリ管理	171	27.2.	フラッシュメモリの応用領域とブートローダ領域	215
21.10.	バット休止	172	27.3.	書き中に読みが可能な領域と不能な領域	215
21.11.	プラグ接続検出	172	27.4.	ブートローダ施錠ビット	217
21.12.	USBソフトウェア操作種別	173	27.5.	ブートローダプログラムへの移行	218
21.13.	USB一般用レジスタ	174	27.6.	自己プログラミングでのフラッシュアドレス指定	220
22.	USB装置動作	176	27.7.	フラッシュメモリの自己プログラミング	220
22.1.	序説	176	28.	メモリプログラミング	226
22.2.	電源ONとリセット	176	28.1.	プログラムメモリとデータメモリ用施錠ビット	226
22.3.	エンドポイントリセット	176	28.2.	ヒューズビット	227
22.4.	USBリセット	176	28.3.	識票バイト	228
22.5.	エンドポイント選択	176	28.4.	校正バイト	228
22.6.	エンドポイントの活性化(有効設定)	177	28.5.	並列プログラミング	229
22.7.	アドレス設定	177	28.6.	並列プログラミング手順	230
22.8.	休止、起動復帰と再開	178	28.7.	直列プログラミング	238
22.9.	分離(切り離し)	178	28.8.	直列プログラミング手順	238
22.10.	遠隔起動復帰	178	28.9.	JTAGインターフェース経由プログラミング	241
22.11.	不能応答(STALL)要求	178	29.	電気的特性	248
22.12.	制御エンドポイント管理	179	29.1.	絶対最大定格	248
22.13.	OUTエンドポイント管理	180	29.2.	DC特性	248
22.14.	INエンドポイント管理	181	29.3.	外部クロック特性	249
22.15.	等時(Isochronous)動作	182	29.4.	システムとリセットの特性	249
22.16.	オーバーフロー	182			

29.5.	最高速と動作電圧	250
29.6.	2線直列インターフェース特性	251
29.7.	SPI タイミング特性	252
29.8.	ハードウェア ブート受付タイミング特性	252
29.9.	A/D変換器特性	253
29.10.	校正付き内蔵RC発振器精度	253
30.	代表特性	254
30.1.	活動動作消費電流	254
30.2.	アイドル動作消費電流	256
30.3.	パワーダウン動作消費電流	257
30.4.	パワーセーブ動作消費電流	258
30.5.	ピン プルアップ	258
30.6.	ピン駆動能力	259
30.7.	ピン 閾値とヒステリシス	261
30.8.	低電圧検出器(BOD)閾値	263
30.9.	内部発振器周波数	265
30.10.	周辺機能部消費電流	267
30.11.	リセット消費電流とリセット パルス幅	268
31.	レジスタ要約	269
32.	命令要約	273
33.	注文情報	275
34.	外圍器情報	275
35.	障害情報	276
36.	データシート改訂履歴	277

Atmel®, Atmelロゴとそれらの組み合わせ、AVR®, AVR Studio®, Enabling Unlimited Possibilities®とその他はAtmel Corporationとその子会社の登録商標または商標またはその付属物です。Windows®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のウェブサイトに表示する販売の条件とAtmelの定義での詳しい説明を除いて、商品性、特定目的に関する適合性、または適法性の暗黙保証に制限せず、Atmelはそれらを含むその製品に関連する暗示的、明示的または法令による如何なる保証も否認し、何ら責任がないと認識します。たとえAtmelがそのような損害賠償の可能性を進言されたとしても、本資料を使用できない、または使用以外で発生する(情報の損失、事業中断、または利益と損失に関する制限なしの損害賠償を含み)直接、間接、必然、偶然、特別、または付随して起こる如何なる損害賠償に対しても決してAtmelに責任がないでしょう。Atmelは本資料の内容の正確さまたは完全性に関して断言または保証を行わず、予告なしでいつでも製品内容と仕様の変更を行う権利を保留します。Atmelはここに含まれた情報を更新することに対してどんな公約も行いません。特に別の方法で提供されなければ、Atmel製品は車載応用に対して適当ではなく、使用されるべきではありません。Atmel製品は延命または生命維持を意図した応用での部品としての使用に対して意図、認定、または保証されません。

安全重視、軍用、車載応用のお断り: Atmel製品はAtmelが提供する特別に書かれた承諾を除き、そのような製品の機能不全が著しく人に危害を加えたり死に至らしめることがかなり予期されるどんな応用(“安全重視応用”)に対しても設計されず、またそれらとの接続にも使用されません。安全重視応用は限定なしで、生命維持装置とシステム、核施設と武器システムの操作の装置やシステムを含みます。Atmelによって軍用等級として特に明確に示される以外、Atmel製品は軍用や航空宇宙の応用や環境のために設計も意図もされていません。Atmelによって車載等級として特に明確に示される以外、Atmel製品は車載応用での使用のために設計も意図もされていません。

© HERO 2025.

本データシートはAtmelのATmega16U4/32U4英語版データシート(Rev.7766J-04/2016)の翻訳日本語版です。日本語では不自然となる重複する形容表現は省略されている場合があります。日本語では難解となる表現は大幅に意識されている部分もあります。必要に応じて一部加筆されています。頁割の変更により、原本より頁数が少なくなっています。

汎用入出力ポートの出力データレジスタとピン入力は、対応関係からの理解の容易さから出力レジスタと入力レジスタで統一表現されています。一部の用語がより適切と思われる名称に変更されています。必要と思われる部分には()内に英語表記や略称などを残す形で表記しています。

青字の部分はリンクとなっています。一般的に赤字の0,1は論理0,1を表します。その他の赤字は重要な部分を表します。

原書に対して若干構成が異なるため、一部の節/項番号が異なります。更に一部の図表番号も異なります。