

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125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
 - 完全なステディック動作
 - 16MHz時、16MIPSに達する高速動作
- 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8/16Kバイト(4/8K語)フラッシュメモリ内蔵
 - 10,000回の書き換えが可能
 - 個別施錠ビットを持つ任意のブートコード領域
 - 工場で既定ブートルoader書き込み済み
 - リセット後ハードウェアで活動されるチップ内ブ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
 - 512バイトのEEPROM (100,000回の書き換えが可能)
 - 512バイトの内蔵SRAM
 - ソフトウェア保護用の設定可能な施錠機能
- 転送完了割り込み付きUSB2.0全速(Full-speed)装置部
 - USB仕様改訂2.0(Universal Serial Bus Specification REV 2.0)完全適合
 - 全速バス動作(12Mbps転送速度)用48MHz PLL
 - 完全に独立したエンドポイントメモリ割付用176バイトUSBデュアルポートRAM(DPRAM)
 - 8~64バイトまでの制御転送用エンドポイント0
 - 設定可能な4つのエンドポイント
 - 入出力(IN/OUT)方向
 - 大量(Bulk)、割り込み(Interrupt)、等時(Isochronous)転送
 - 8から64バイトまでに設定可能なパケット容量
 - 単一または二重に設定可能な緩衝部
 - 休止(Suspend:サスペンド)/再開(Resume:リジューム)割り込み
 - 分離(Detach)なしUSBバスリセットでのマイクロコントローラリセット
 - マイクロコントローラ要求でのUSBバス切断
 - 単一ケーブル機能用にPS/2とで多重化されたUSBハット
- 内蔵周辺機能
 - PS/2適合ハット
 - 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 (2つの8ビットPWMチャネル)
 - 独立した前置分周器、比較、捕獲機能付き、1つの16ビットタイマ/カウンタ (3つの16ビットPWMチャネル)
 - SPI主装置専用動作とハードウェア流れ制御(RTS/CTS)付きUSART
 - 主装置/従装置動作SPI直列インターフェース
 - 設定可能な専用発振器付きウォッチドッグタイマ
 - アナログ比較器
 - ピン変化での割り込みと起動復帰
- 内蔵デバッグインターフェース(デバッグWIRE)
- 特殊マイクロコントローラ機能
 - 電源ONリセット回路と設定可能な低電圧検出器(BOD)
 - 校正可能な内蔵RC発振器
 - 外部及び内部の割り込み
 - アイドル、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5つの低消費動作
- I/Oと外圍器
 - 22ビットの設定可能なI/O
 - 32ピンTQFP、32ピンQFN
- 動作電圧
 - 2.7~5.5V
- 温度範囲
 - -40~85°C (工業用)
- 最大速度
 - 8MHz/2.7V (全温度範囲)
 - 16MHz/4.5V (全温度範囲)



8ビット **AVR[®]**
 マイクロ コントローラ
 実装書き換え可能な
 8/16Kバイト
 フラッシュメモリ、
 USB制御器内蔵

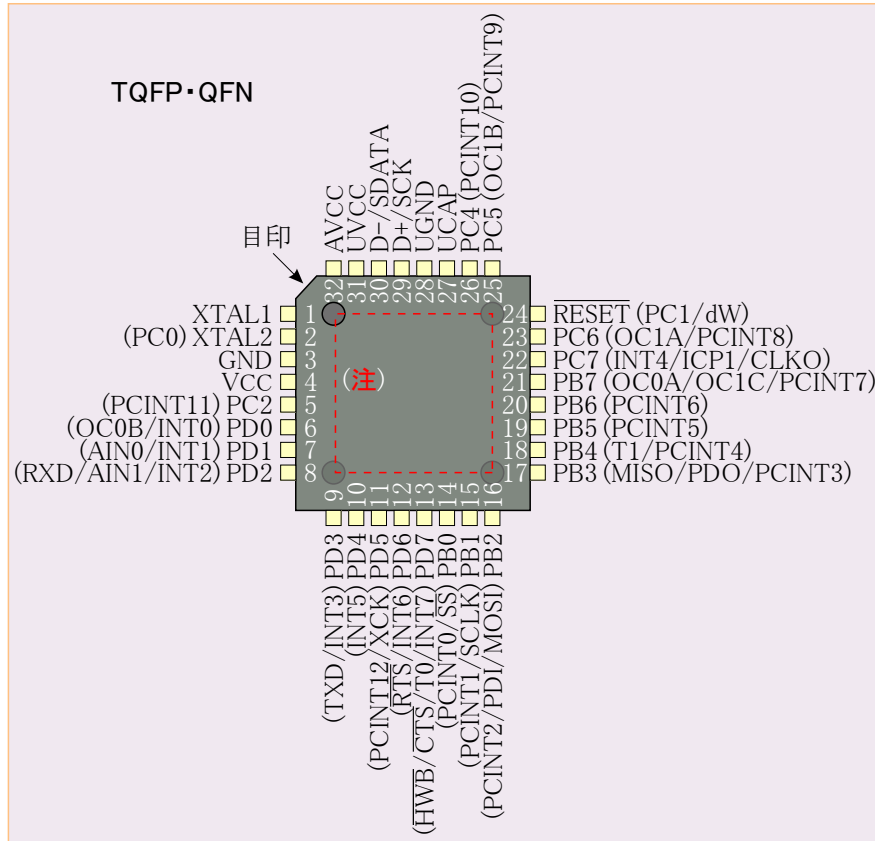
AT90USB82
 AT90USB162



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

Rev. 7707F-11/10, 7707FJ7-08/22

1. ピン配置



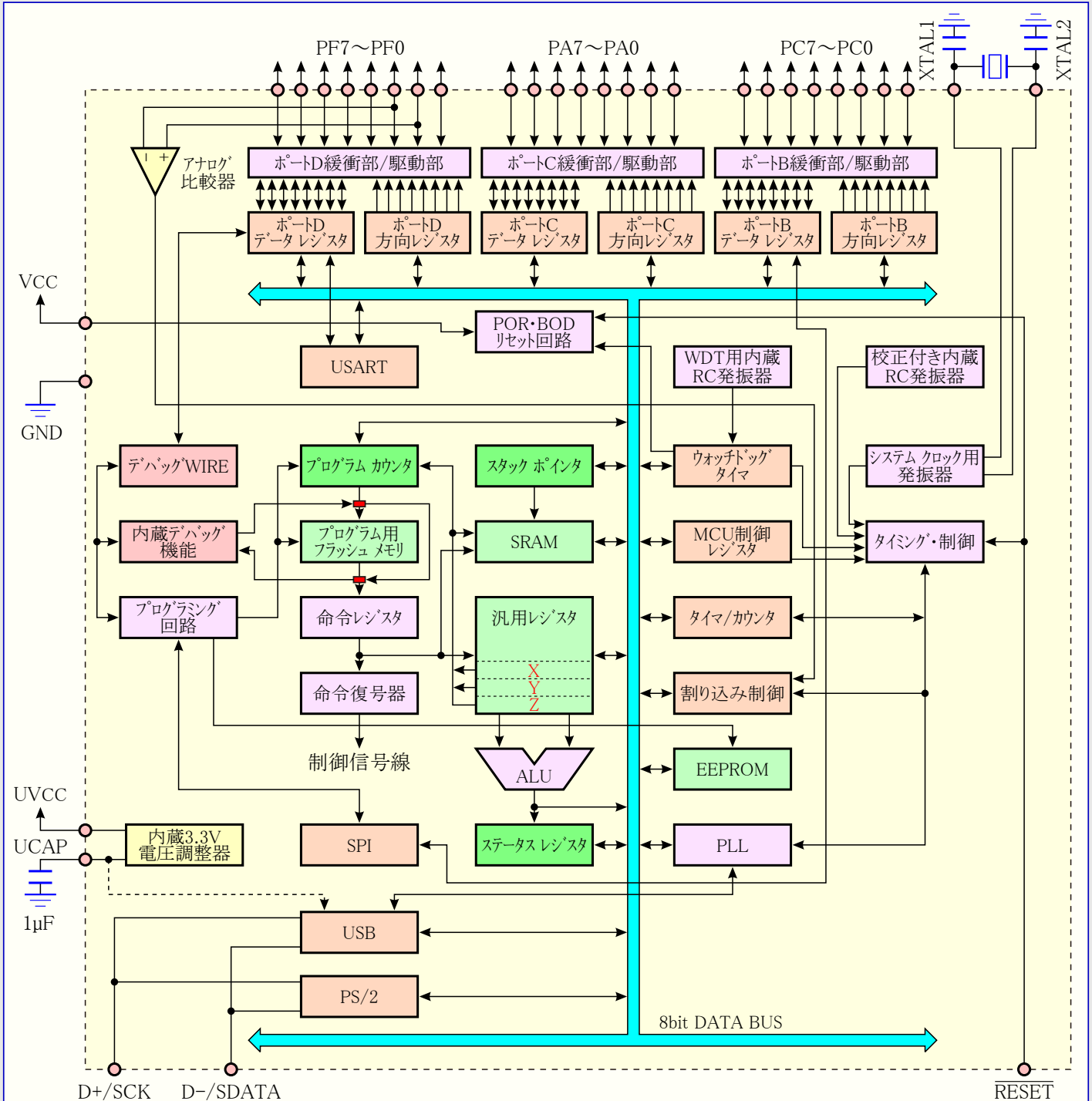
注: QFN外囲器底面中央の大パッドは金属で作られ、内部でGNDに接続されています。これは良好な機構的安定性を保証するために、基板へ貼り付けるか、半田付けされるべきです。この中央パッドが未接続の場合、基板から脱落するかもしれません。

2. 概要

AT90USB82/162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最大10倍の単位処理量向上効果があります。

AT90USB82/162は次の特徴、書き込み中読める能力を持つ8/16Kバイトの実装書き換え可能なフラッシュメモリ、512バイトのEEPROM、512バイトのSRAM、22本の汎用入出力線、32個の汎用作業レジスタ、比較動作やPWMを含む柔軟性のある2つのタイマ/カウンタ、1つのUSART、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デバッグWIREインターフェース、ソフトウェアで選べる5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。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。拡張スタンバイ動作では主発振器が動作を継続します。

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。モノ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Atmel AT90USB82/162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。

AT90USB82/162 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48頁で一覧されるAT90USB82/162の様々な特殊機能も扱います。

2.2.4. PC7~PC0 - (ポートC)

ポートCは(ビット毎に選ばれる)内蔵プルアップ抵抗付き7ビット双方向入出力ポートです。ポートC出力緩衝部は高い吐き出しと吸い込み両方の能力の対称駆動特性です。入力としてプルアップ抵抗が活性(有効)なら、外部的にLowへ引かれたポートCピンは電流を吐き出します。リセット条件が有効になるとクロックが動いていなくてもポートCピンはHi-Zにされます。

ポートCは50頁で一覧されるAT90USB82/162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はアナログ比較器へのアナログ入力として扱います。

(PD2/PD1ピンに関して)アナログ比較器が使われないなら、ポートD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52頁で一覧されるAT90USB82/162の様々な特殊機能も扱います。

2.2.6. D-/SDATA

USB全速(Full-speed)負極性データ上側ポート / PS/2データポート。

2.2.7. D+/SCK

USB全速(Full-speed)正極性データ上側ポート / PS/2クロックポート。

2.2.8. UGND

USB接地。

2.2.9. UVCC

USBハット用内蔵電圧調整器の供給電圧入力。

2.2.10. 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(1 μ F)に接続されるべきです。

2.2.11. RESET/PC1/dW

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30頁の表9-1.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。本ピンは選択的に標準入出力またはデバッグWIREチャネルとしても扱います。この状態(設定)はヒューズのRSTDISBLとDWENに依存します。

2.2.12. 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。

2.2.13. XTAL2/PC0

ヒューズによって許可なら、発振器反転増幅器からの出力。標準入出力としても扱います。

3. コード例について

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

これらのコード例はアセンブルまたはコンパイルに先立ってデバイス定義ファイルがインクルードされることが前提です。拡張I/O領域に配置したI/Oレジスタに対し、IN, OUT, SBIS, SBIC, CBI, SBI命令は拡張I/O領域へのアクセスを許す命令に置き換えられなければなりません。代表的にはSBRS, SBRC, SBR, CBR命令と組み合わせたLDS, STS命令です。

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ド構造を使います。プログラムメモリ内の命令は、単一段のパイプラインで実行されます。1命令の実行中に次の命令がプログラムメモリから事前取得されます。この概念は全部のクロック周期で命令実行を可能にします。プログラムメモリは実装書き換え可能なフラッシュメモリです。

高速レジスタファイルは1クロック周期アクセスの32個の8ビット長汎用レジスタを含みます。これは1クロック周期ALU(Arithmetic Logic Unit)操作を許します。代表的なALU操作では2つのオペランドがレジスタファイルからの出力で、1クロック周期内でその操作が実行され、その結果がレジスタファイルに書き戻されます。

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。

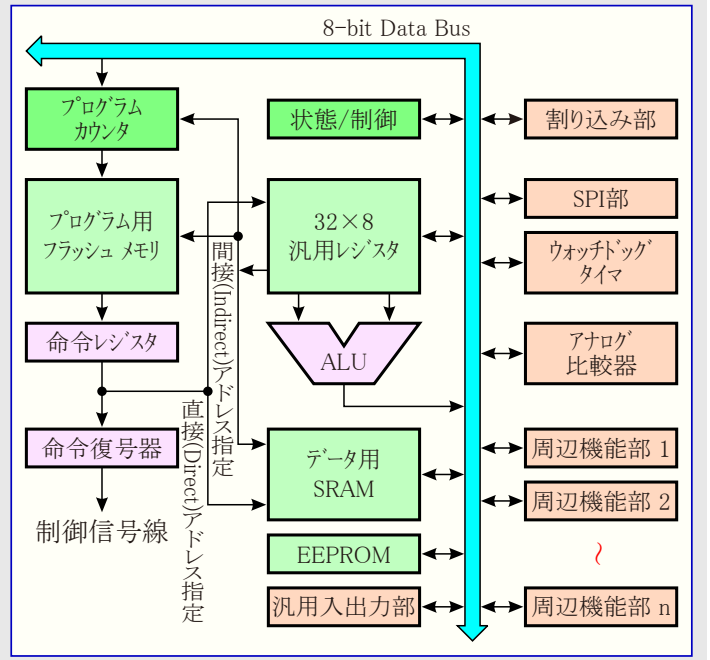
柔軟な割り込み部にはI/O空間の各制御レジスタとステータスレジスタ(SREG)の特別な全割り込み許可(1)ビットがあります。全ての割り込みは割り込みベクタ表に個別の割り込みベクタを持ちます。割り込みには割り込みベクタ表の位置に従う優先順があります。下位側割り込みベクタアドレスが高い優先順位です。

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AT90USB82/162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。

4.7. 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(clk_{CPU})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4-4.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4-5.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4-4. 命令の取得と実行の並列動作

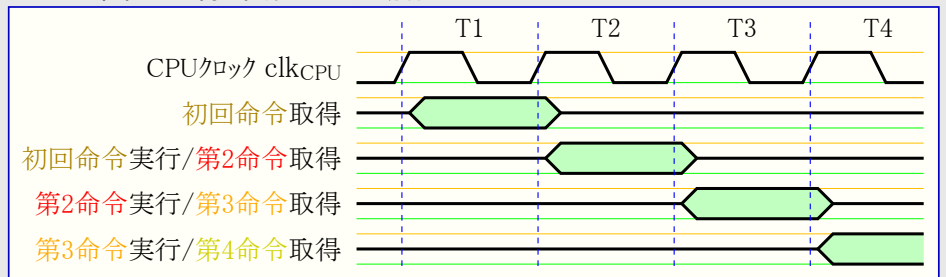
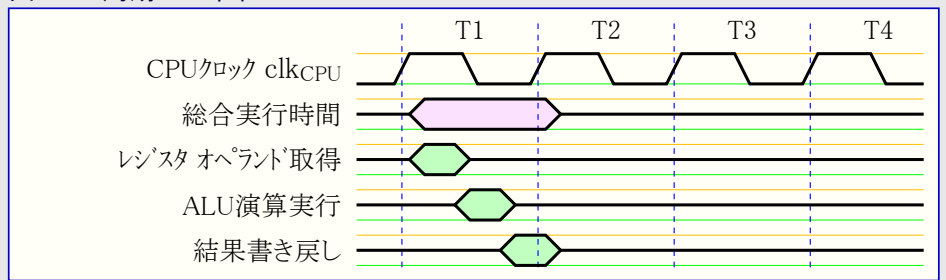


図4-5. 1周期ALU命令



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値によっては割り込みが自動的に禁止されるかもしれません。この特質はソフトウェア保護を改善します。詳細については157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39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39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。146頁の「ブートローダ支援 - 書き込み中読み出し可能な自己プログラミング」をご覧ください。

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

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

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

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

ステータスレジスタ(SREG)は割り込みルーチンへ移行時の保存も、復帰時の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。

(訳注) 原書の本項記述は(修正誤りを含む)3バイト長PC値での記述と思われます。AT90USB82/162については2バイト長PC値です。で、本書記述になるものと予測されます。けれども、これらのデバイスが上位256Kバイトフラッシュ搭載製品と同一構造なら、本書記述は誤りです。

同様にベクタでの無条件分岐命令クロック数はAT90USB162が対象で、AT90USB82の場合は3ではなく2クロック周期が予測されますが、これも上記同様の特別な構造の場合は同一となるかもしれません。

5. AVR AT90USB82/162のメモリ

この項はAT90USB82/162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AT90USB82/162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。

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

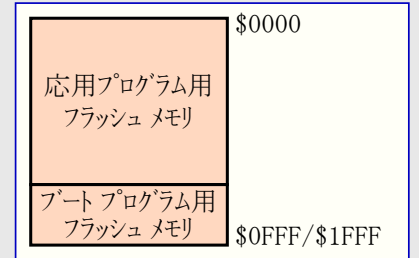
AT90USB82/162はプログラム保存用に実装書き換え可能な8/16Kバイトのフラッシュメモリをチップ上に含みます。全てのAVR命令が16または32ビット幅のため、このフラッシュメモリは4/8K×16ビットとして構成されます。ソフトウェア保護のため、フラッシュプログラムメモリ空間はブートプログラム領域と応用プログラム領域の2つに分けられます。

フラッシュメモリは最低10,000回の消去/書き込み回数の耐久性があります。AT90USB82/162のプログラムカウンタ(PC)は12/13ビット幅で、故に4/8K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146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。157頁の「[メモリプログラミング](#)」はSPI、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AT90USB82/162のSRAMメモリ構成方法を示します。

AT90USB82/162はINやOUT命令で予約した64位置で支援されるより多くの周辺機能部を持つ複合マイクロコントローラです。SRAM(データ空間)内\$60～\$FFの拡張I/O空間に対してLD/LDS/LDDとST/STS/STD命令だけが使えます。

下位768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512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。

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

自動の事前減少付きと事後増加付きのレジスタ間接アドレス指定動作を使う時に(使われる)X,Y,Zアドレスレジスタは減少(-1)または増加(+1)されます。

AT90USB82/162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、512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. データメモリ配置図

		アドレス
レジスタ ファイル (32×8)	R0 ～ R31	\$0000 ～ \$001F
I/O レジスタ (64×8)	\$00 ～ \$3F	\$0020 ～ \$005F
拡張I/O レジスタ (160×8)	\$0060 ～ \$00FF	\$0060 ～ \$00FF
内蔵 SRAM (512×8)	\$0100 ～ \$02FF	\$0100 ～ \$02FF

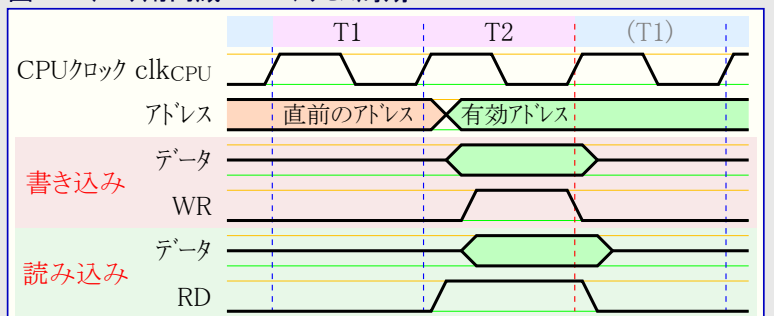
注: 赤字はI/Oアドレス

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 clkCPU周期で実行されます。

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

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

AT90USB82/162は512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へのSPI、デバッグWIRE、並列プログラミングの詳細記述については各々[169頁](#)、[145頁](#)、[160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[13頁の表5-1](#)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。厳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[14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. EEPROMアドレスレジスタ (EEPROM Address Register) EEARH,EEARL (EEAR)

ビット	15	14	13	12	11	10	9	8	
\$22 (\$42)	—	—	—	—	—	—	—	EEAR8	EEARH
Read/Write	R	R	R	R	R	R	R	R/W	
初期値	0	0	0	0	0	0	0	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15~9 - 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8~0 - EEAR8~0 : EEPROMアドレス (EEPROM Address)

EEPROMアドレスレジスタ(EEARHとEEARL)は512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. EEPROMデータレジスタ (EEPROM Data Register) EEDR

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

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5-1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5-1. 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プログラム許可(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プログラム許可(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検査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(書き込みを)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146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プです。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。表5-2.はCPUからのEEPROMアクセスに対する代表的な書き込み時間を示します。

表5-2. 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<<EEPM1) | (0<<EEPM0) ;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<<EEPM1) | (0<<EEPM0);          /* 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「コード例について」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「コード例について」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自身が命令を間違って実行し得ます。

EEPROMデータ化けは次の推奨設計によって容易に避けられます。

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

5.4. I/O メモリ (レジスタ)

AT90USB82/162のI/O空間定義は188頁の「レジスタ要約」で示されます。

AT90USB82/162の全てのI/Oと周辺機能はI/O空間に配置されます。全てのI/O位置はI/O空間と32個の汎用作業レジスタ間のデータ転送を行うLD/LDS/LDD命令とST/STS/STD命令によってアクセスされます。アドレス範囲\$00～\$1F内のI/OレジスタはSBI命令とCBI命令の使用で直接的にビットアクセス可能です。これらのレジスタではSBISとSBIC命令の使用によって単一ビット値が検査できます。より多くの詳細については「命令要約」章を参照してください。I/O指定命令INとOUTを使う時はI/Oアドレス\$00～\$3Fが使われなければなりません。LD命令とST命令を使い、データ空間としてI/Oレジスタをアクセスする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。

AT90USB82/162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

AT90USB82/162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. 汎用I/Oレジスタ2 (General Purpose I/O Register 2) GPIOR2

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

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

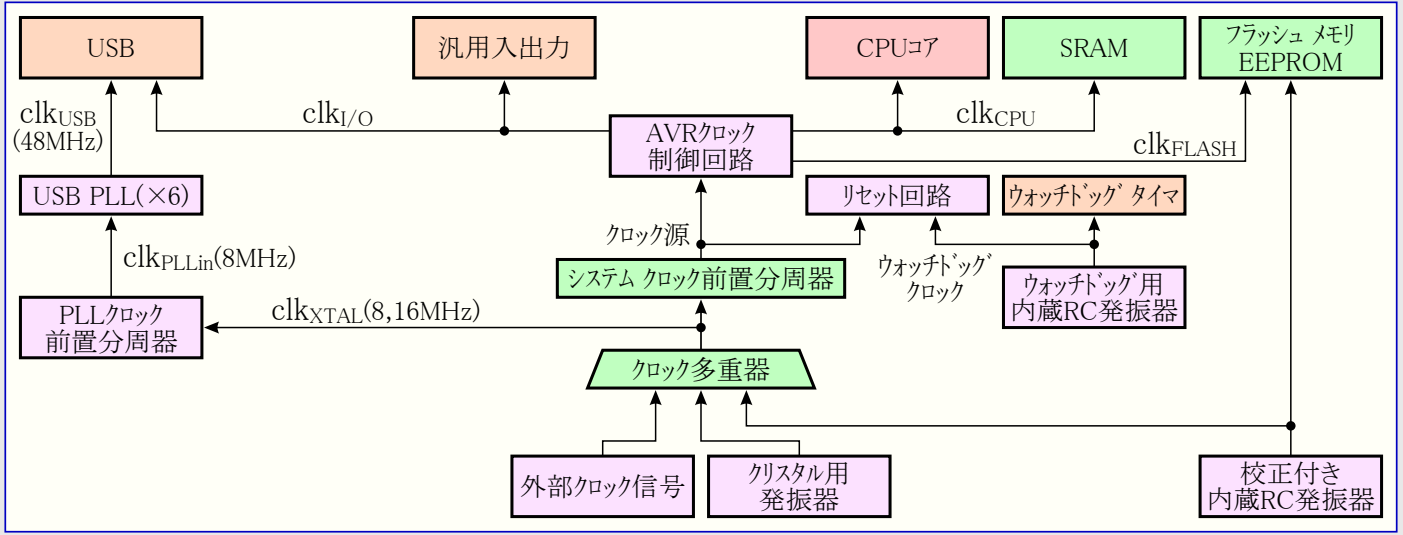
ビット	7	6	5	4	3	2	1	0	
\$1E (\$3E)	(MSB)							(LSB)	GPIOR0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

6. システム クロックとクロック選択

6.1. クロック系統とその配給

図6-1.はAVR内の主要なクロック系統とその配給を示します。全てのクロックが与えられた時間有効である必要はありません。消費電力低減のため、26頁の「電力管理と休止形態」で記述される各種休止形態の使用によって、使われない部分のクロックを停止することができます。クロック系統は以下で詳細に示されます。

図6-1. クロックの配給



6.1.1. CPU クロック – clk_{CPU}

CPUクロックはAVRコアの動作と関係する系統の部分に配給されます。このような部分の例は汎用レジスタファイル、ステータスレジスタ、スタックポインタを保持するデータメモリです。CPUクロックの停止はコアが一般的な操作や計算を実行することを禁止します。

6.1.2. I/O クロック – $clk_{I/O}$

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

6.1.3. フラッシュ クロック – clk_{FLASH}

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

6.1.4. USBクロック – clk_{USB}

USBには専用のクロック範囲が提供されます。このクロックは48MHzで走行するチップ上のPLLで生成されます。このPLLは常に入力周波数を6倍します。従ってPLLクロックレジスタはPLL入力で8MHzを生成するために、ソフトウェアによってプログラム(設定)されるべきです。

6.2. クロック切り替え

AT90USB82/162製品ではクロック多重器とシステムクロック前置分周器がソフトウェアによって変更できます。

6.2.1. 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起こり得ます。そして消費削減の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。このような設定では外部クロックが禁止されます。

ファームウェアは応用での事象があるかを検出するために、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、ファームウェアは校正付き内蔵RC発振器から外部クロックへクロック多重器を切り替えます。

図6-2. USBホストからの起動復帰でのクロック切り替え例

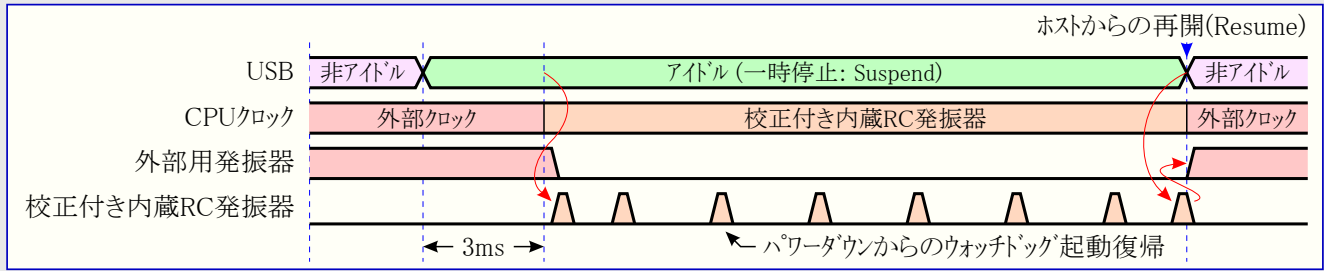
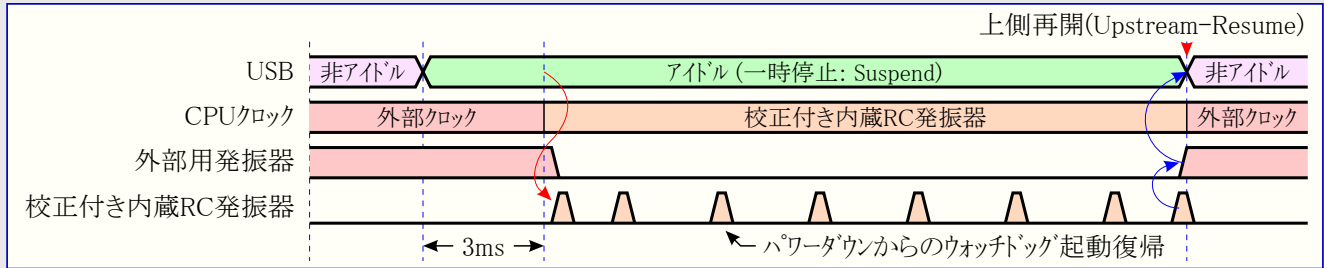


図6-3. USB装置からの起動復帰でのクロック切り替え例



6.2.2. クロック切り替え方法

6.2.2.1. 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;
)

```

6.2.2.2. 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();            // CLKSELO.EXT=1;
    while (!External_clock_ready());    // while (CLKSTA.EXTON!=1);
    Select_External_clock();            // CLKSELO.CLKS=1;
    Enable_pll();                       // PLLCSR.PLLE=1;
    Disable_RC_clock();                 // CLKSELO.RCE=0;
    while (!Pll_ready());               // while (PLLCSR.PLOCK!=1);
    Usb_unfreeze_clock();               // USBCON.FRZCLK=0;
)

```

6.2.3. クロック選択レジスタ0 (Clock selection Register 0) CLKSELO

ビット (\$D0)	7	6	5	4	3	2	1	0	
	RCSUT1	RCSUT0	EXSUT1	EXSUT0	RCE	EXTE	–	CLKS	CLKSELO
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。このRC発振器がヒューズビットによって設定されている場合、そのSUTヒューズがこれらのビット内に複写されます。この付加起動時間が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/低電力発振器用SUT値 (SUT for External Oscillator/Low Power Oscillator)

これら2ビットは外部発振器/低電力発振器用SUT値です。外部発振器/低電力発振器がヒューズビットによって設定されている場合、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。この値は外部発振器/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/低電力発振器許可 (Enable External Oscillator/Low Power Oscillator)

外部発振器/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。外部発振器/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発振器/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発振器/低電力発振器が選ばれるなら、CLKSビットは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。

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

ビット (\$D1)	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/低電力発振器用CKSEL値 (CKSEL for External Oscillator/Low Power Oscillator)

外部発振器/低電力発振器用クロック設定。リセット後、外部発振器/低電力発振器がヒューズビットによって選ばれているなら、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。

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

ビット (\$D2)	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、この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/低電力発振器ON (External Oscillator/Low Power Oscillator On)

外部発振器/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。外部発振器/低電力発振器が停止なら、このビットはハードウェアによって解除(0)されます。

6.3. 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
外部クリスタル低電力発振器	1111~1000
(予約)	0111~0011
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

6.3.1. 既定のクロック元

このデバイスはクリスタル用低電力発振器でプログラム(0)されたCKDIV8ヒューズで出荷されます。起動時間は計時完了周期が許可され、最大に設定されます。(CKSEL=1110, SUT=01, CKDIV8=プログラム(0))

6.3.2. 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_{TOUT})の内部リセットを発生します。30ページの「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_{TOUT})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。選択可能な遅延は表6-2で示されます。176ページの「代表特性(暫定)」で示されるようにウォッチドッグ発振器の周波数は電圧に依存します。

表6-2. 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)

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。パワーダウン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。

6.4. クリスタル用低電力発振器

XTAL1とXTAL2は図6-4.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なく、雑音が多い環境で、より雑音の影響を受け易くなります。

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表6-3.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。この動作種別は表6-3.で示されるようにCKSEL3~1ヒューズによって選ばれます。

CKSEL0ヒューズはSUT1,0ヒューズと共に表6-4.で示されるように起動時間を選びます。

図6-4. クリスタル発振子接続図

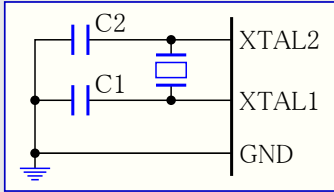


表6-3. クリスタル用低電力発振器動作種別

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: 周波数範囲は暫定値で実際値はTBDです。

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。

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。

(訳注) 原書本位置の表6-5.は表6-7.(本書の表6-6.)が誤って挿入されたもののため削除しました。

6.5. 校正付き内蔵RC発振器 (訳注: 内容を新版対応に変更)

既定による校正された内蔵RC発振器は概ね8.0MHzのクロックを供給します。電圧と温度に依存しますが、このクロックは使用者によって高精度校正ができます。より多くの詳細については175頁の表26-4と186頁の「校正付き内蔵RC発振器周波数」をご覧ください。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。より多くの詳細については23頁の「システムクロック前置分周器」をご覧ください。

このクロックは表6-5で示されるようにCKSELヒューズのプログラミングによってシステムクロックとして選べます。選んだなら、外部部品なしで動作します。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。この校正の精度は表26-4で工場校正として示されます。

ソフトウェアからOSCCALレジスタを変更することによって(発振校正レジスタ(OSCCAL)参照)、工場校正を使うよりも高い精度を得ることができます。この校正の精度は表26-4で使用者校正として示されます。

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159頁の「校正バイト」項をご覧ください。

この発振器が選ばれると、起動時間は表6-6で示されるようにSUTヒューズによって決定されます。

表6-5. 校正付き内蔵RC発振器動作種別 注: デバイスはこの選択で出荷されます。

CKSEL3~0	周波数範囲 (MHz)
0 0 1 0	7.3~8.1

注: 周波数範囲は暫定値で実際の値はTBDです。

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。

表6-6. 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 (注1)	6×CK	14×CK+65ms	低速上昇電源
1 1			(予約)

注1: デバイスはこの選択で出荷されます。

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

ビット (\$66)	7	6	5	4	3	2	1	0	
	CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	OSCCAL
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) (訳注: 内容を新版対応に変更)

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、175頁の表26-4で指定されるように工場で校正された周波数を与える、予めプログラムされた値が本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。この発振器は表26-4で指定される周波数に校正できます。この範囲外への校正は推奨されません。

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書き込みは失敗するかもしれません。

CAL7ビットは発振器に対する操作範囲を決めます。このビットの(0)設定は低周波数範囲になり、(1)設定は高周波数範囲になります。この2つの周波数範囲は重複し、別の言葉では、OSCCAL=\$7F設定はOSCCAL=\$80設定より高い周波数になります。

CAL6~0ビットは選んだ範囲内の周波数調整に使われます。\$00設定はその範囲の最低周波数になり、\$7F設定はその範囲の最高周波数になります。

6.6. 外部クロック信号

本デバイスは図6-5.で示されるように外部クロック信号を利用できます。外部クロックでデバイスを走行するためには、CKSELヒューズが表6-1.で示されるようにプログラム(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、起動時間は表6-7.で示されるようにSUTヒューズによって決定されます。

図6-5. 外部クロック信号駆動接続図

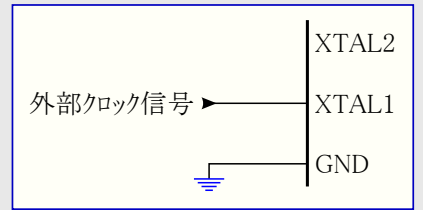


表6-7. 外部クロック信号駆動用起動遅延時間選択表

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	$6 \times CK$	$14 \times CK$	低電圧検出(BOD)リセット許可
0 1	$6 \times CK$	$14 \times CK + 4.1ms$	高速上昇電源
1 0	$6 \times CK$	$14 \times CK + 65ms$	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23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ばず。システムクロック前置分周器が使われると、CKOUTヒューズがプログラム(0)された時の出力は分周したシステムクロックです。

6.8. システム クロック前置分周器

AT90USB82/162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。clk_{CPU}、clk_{FLASH}、clk_{I/O}は**表6-8**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(CLKPCCE)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CLKPCCE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。

6.8.1. クロック前置分周レジスタ (Clock Prescaler Register) CLKPR

ビット (\$61)	7	6	5	4	3	2	1	0	
	CLKPCCE	—	—	—	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 – CLKPCCE : 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CLKPCCEビットは論理**1**を書かれなければなりません。CLKPCCE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。CLKPCCE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CLKPCCEビット再書き込みは制限時間の延長もCLKPCCEビットの解除(**0**)も行いません。

• ビット6~4 – 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3~0 – 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んだ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**表6-8**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んだ場合、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んだ場合、十分な分周値が選ば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6-8. 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		1	
CLKPS0	0	1	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)						

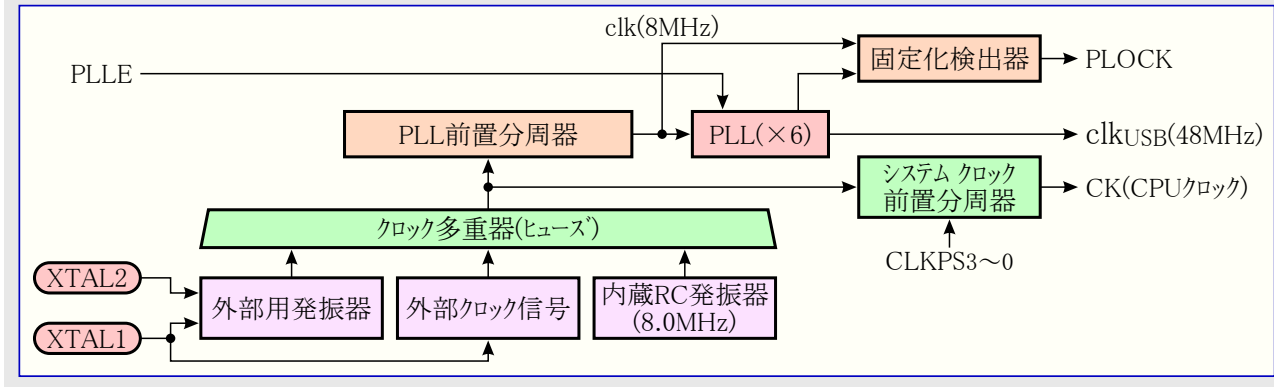
6.9. PLL

このPLLはUSBインターフェース用の内部高周波数(48MHz)の生成に使われ、PLL入力には外部低周波数(XTAL1ピンからの外部クロック入力または水晶発振器)から生成されます。

6.9.1. USBインターフェース用内部PLL

AT90USB82/162の内部PLLは公称8MHz入力から6倍したクロック周波数を生成します。8MHzのPLL入力クロック元は、8MHzを生成する内部PLLクロック前置分周器の出力です。

図6-6. PLLクロック構成図



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

ビット	7	6	5	4	3	2	1	0	
\$29 (\$49)	—	—	—	PLLP2	PLLP1	PLLP0	PLLE	PLOCK	PLLCSR
Read/Write	R	R	R	R/W	R/W	R/W	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4~2 – PLLP2~0 : PLL前置分周選択 (PLL prescaler)

これらのビットはPLL用に8MHz入力クロックを生成するためのPLL入力前置分周器設定を許します。

表6-9. PLL入力前置分周器選択

PLLP2	PLLP1	PLLP0	分周数	USB動作に必要な外部XTAL (MHz)
0	0	0	1	8
0	0	1	2	16
0	1	0	(予約)	—
0	1	1	(予約)	—
1	0	0	(予約)	—
1	0	1	(予約)	—
1	1	0	(予約)	—
1	1	1	(予約)	—

• ビット1 – PLLE : PLL許可 (PLL Enable)

PLLEが設定(1)されると、PLLが起動され、必要ならばPLL基準クロックとして校正付き内蔵RC発振器が始動されます。PLLがシステムクロック元として選ばれる場合、本ビットの値は常に1です。

• 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が基準クロックに固定化され、USBに対するclkUSB使用は安全です。PLL許可後、PLLが固定化するのに約1~100msかかります。

7. 電力配給

AT90USB82/162製品はUSBハット[®](図7-1.参照)、応用に依存して外部部品またはマイクロコントローラ自身(図7-2.参照)にも供給を許す内蔵5V⇒3.3V電圧調整器を含みます。

図7-1. 5V単独形態

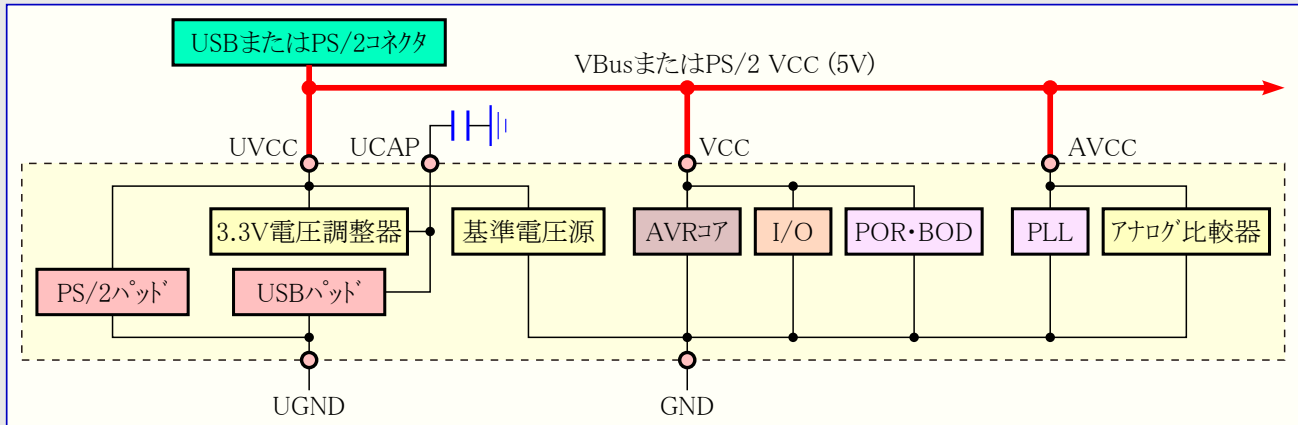


図7-2. 5V/3.3V併用形態

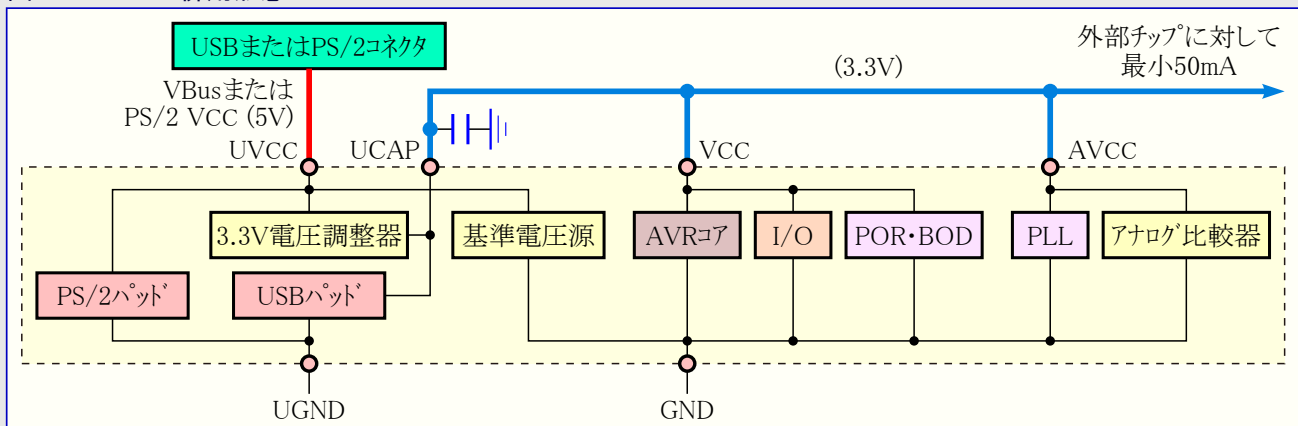
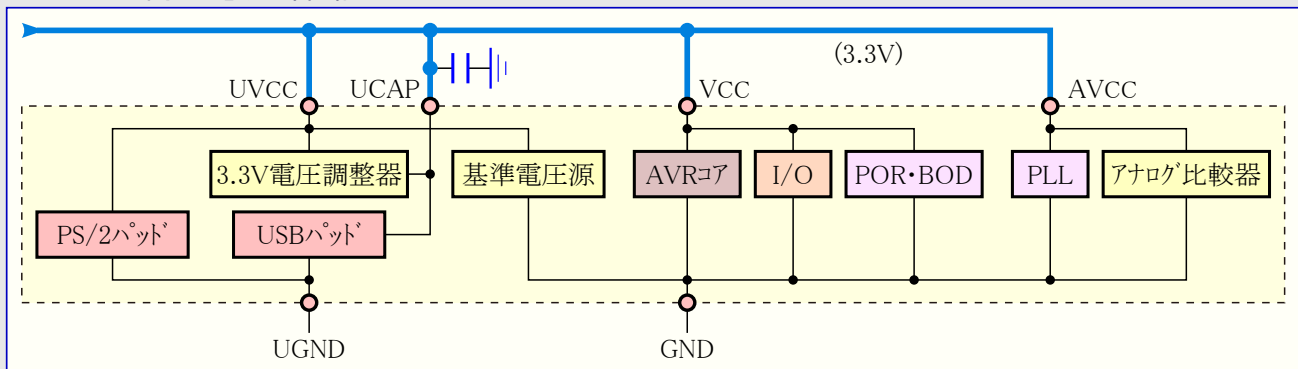


図7-3. 3.3V単独形態(外部供給)



重要注意: 3.3V単独形態での内部3.3V電圧調整器は迂回されます。余分な電力消費を避けるため、電圧調整器は禁止されなければなりません。

7.0.1. 電圧調整器制御レジスタ (Regulator Control Register) REGCR

ビット (\$63)	7	6	5	4	3	2	1	0	
	—	—	—	—	—	—	—	REGDIS	REGCR
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0 - REGDIS : 電圧調整器禁止 (Regulator Disable)

内部3.3V電圧調整器を禁止するには、このビットを設定(1)してください。このビットは完全な3.3V応用でだけ使われるべきです。このビットはリセット毎に0へリセットされます。ファームウェアは必要ならばリセット毎にこれを設定(1)する責任があります。

8. 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。

5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、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8-1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れば、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れば、MCUは起動し、リセットベクタから実行します。

16頁の図6-1はAT90USB82/162の各種クロック系統とその配給を示します。この図は適切な休止形態を選ぶ助けになります。

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

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7~4 – Res : 予約 (Reserved)

これらのビットは予約されており、常に0として読みます。

• ビット3~1 – 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8-1で示される利用可能な5つの休止形態の1つを選びます。

表8-1. 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	(予約)
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。

8.1. 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、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の送信完了や(SOFI, WAKEUPI...のような)いくつかのUSB割り込み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)することにより、アナログ比較器を電源断にできます。これはアイドル動作での消費電力を削減します。

8.2. 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、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、INT7~4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(WAKEUPIのみ)だけがMCUを起動できます。この休止形態は基本的に生成した全てのクロックを停止し、非同期部の動作だけを許します。

レベルで起動した割り込みがパワーダウン動作からの復帰に使われる場合、MCUを起動するためには変更したレベルが一定時間保持されなければならないことに注意してください。詳細については41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19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。

8.3. 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。本動作はパワーダウン動作と同じです。この動作種別は上位製品との互換目的用に維持されています。

8.4. 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。

8.5. 拡張スタンバイ動作

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

表8-2. 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 主クロック 供給元	復帰起動要因 (割り込み)					
	clk _{CPU}	clk _{FLASH}	clk _{IO}		INT _n ピン変化 割り込み	SPM EEPROM 操作可	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	○
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 INT7~4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WAKEUPIだけです。

8.6. 電力削減

電力削減レジスタ(PRRn)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(PRRn)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。

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

ビット (\$64)	7	6	5	4	3	2	1	0	
	—	—	PRTIM0	—	PRTIM1	PRSPI	—	—	PRR0
Read/Write	R	R	R/W	R	R/W	R/W	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 – 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1,0 – 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。

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

ビット (\$65)	7	6	5	4	3	2	1	0	
	PRUSB	—	—	—	—	—	—	PRUSART	PRR1
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 – 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6~1 – 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。

8.7. 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。

8.7.1. 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。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143頁の「[アナログ比較器](#)」を参照してください。

8.7.2. 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。

8.7.3. 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33頁の「[内部基準電圧](#)」を参照してください。

8.7.4. 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34頁の「[ウォッチドッグ タイマ](#)」を参照してください。

8.7.5. ポートピン

休止形態へ移行するとき、全てのポートピンは最小電力使用に設定されるべきです。最も重要なことはその時にピンが抵抗性負荷を駆動しないのを保証することです。I/Oクロック(clk_{I/O})とA/D変換クロック(clk_{ADC})の両方が停止される休止形態ではデバイスの入力緩衝部が禁止されます。これは必要とされない時に入力論理回路によって電力が消費されないことを保証します。いくつかの場合で入力論理回路は起動条件を検出するために必要とされ、その時は許可されます。どのピンが許可されるかの詳細については46頁の「[デジタル入力許可と休止形態](#)」を参照してください。入力緩衝部が許可され、入力信号が浮いている状態のままか、またはアナログ信号電圧がVCC/2付近の場合、入力緩衝部は過大な電力を消費するでしょう。

アナログ入力ピンに対するデジタル入力緩衝部は常に禁止されるべきです。入力ピンでのVCC/2付近のアナログ信号入力は活動動作でも重要な電流を引き起こし得ます。デジタル入力緩衝部はデジタル入力禁止レジスタ(DIDR1)の書き込みによって禁止できます。詳細については144頁の「[デジタル入力禁止レジスタ\(DIDR1\)](#)」を参照してください。

8.7.6. 内蔵デバッグ機能 (dW)

内蔵デバッグ機能がDWENヒューズによって許可され、チップが休止形態へ移行すると、主クロック元は許可に留まり、従って常に電力を消費します。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。

9. システム制御とリセット

9.1. 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を配置できます。これはリセットベクタが応用領域の一方、割り込みベクタがポート領域の場合やその逆も同様です。図9-1.の回路構成図はリセット論理回路を示します。表9-1.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19頁の「クロック元」で示されます。

9.2. リセット元

AT90USB82/162には次の5つのリセット元があります。

- **電源ONリセット** 供給電圧が電源ONリセット閾値電圧(V_{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。34頁の「ウォッチドッグタイマ」をご覧ください。
- **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。
- **USBリセット** USB部が許可され、USBリセットを検出すると、MCUはリセットされます。このリセットではUSB部が許可に留まり、故にデバイスがバスに接続された状態に留まることに注意してください。

図9-1. リセット回路構成

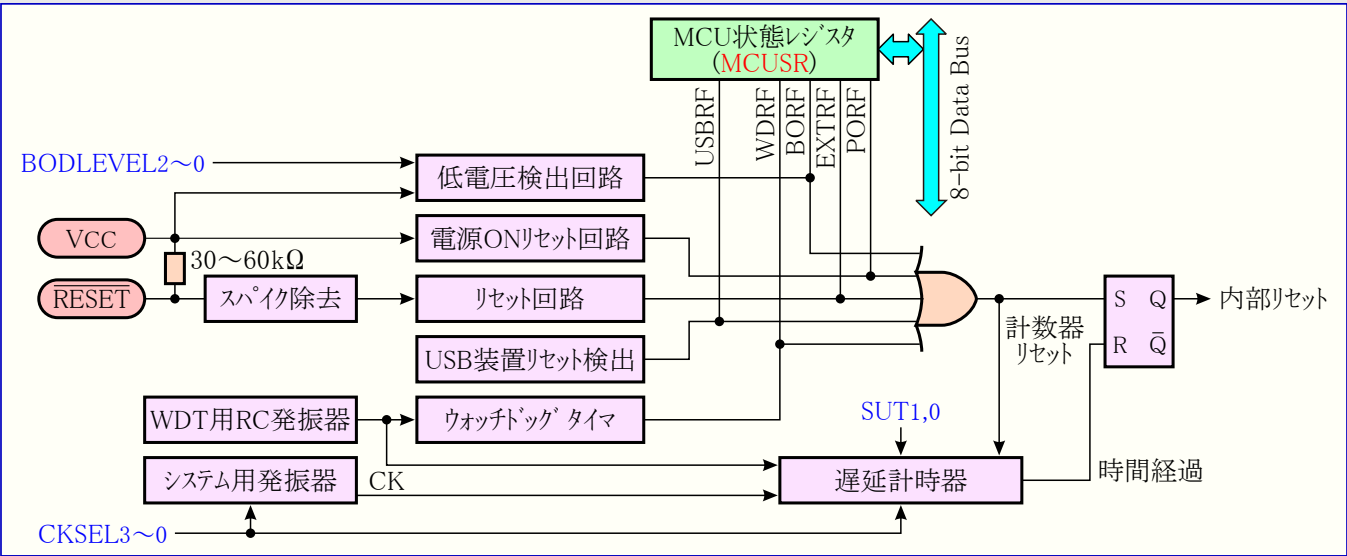


表9-1. リセット電気的特性

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 _{CCRR}	内部電源ONリセット信号保証のVCC上昇率		0.8			V/ms
V _{RST}	RESETピン閾値電圧		0.2VCC		0.85VCC	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。

9.3. 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。検出電圧は表9-1で定義されます。POR信号はVCCが検出電圧以下の時は必ず活性(有効)にされます。POR回路は供給電圧異常検出は勿論、始動リセットの起動にも使えます。

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

図9-2. 内蔵電源ONリセット ($\overline{RESET}$ はVCCに接続)

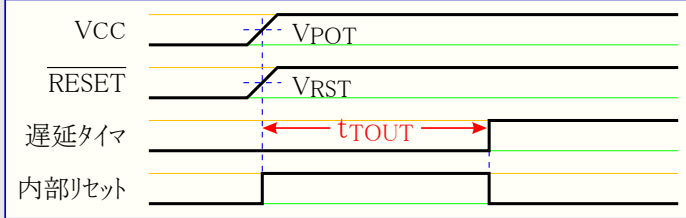
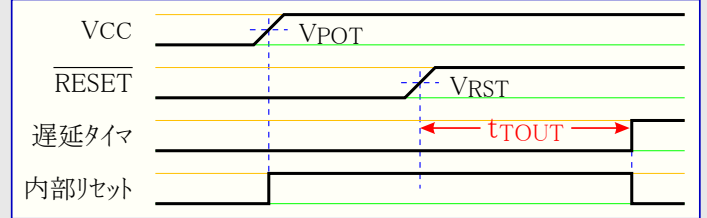


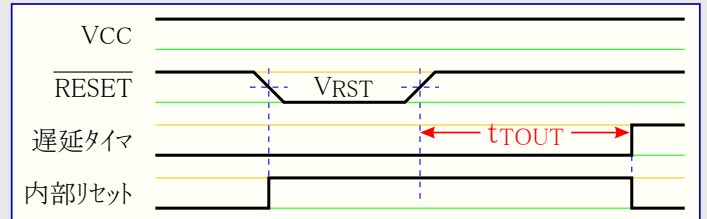
図9-3. 外部RESET信号による延長電源ONリセット



9.4. 外部リセット

外部リセットは $\overline{RESET}$ ピンのLowレベルによって生成されます。クロックが動いていなくても、最小パルス幅(表9-1参照)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生成が保証されません。印加された信号の上昇がリセット閾値電圧(V_{RST})に達すると(遅延タイマを起動し)、遅延タイマは遅延時間(t_{TOUT})経過後にMCUを始動します。

図9-4. 動作中の外部リセット



9.5. 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。この起動電圧はスパイク対策BODを保証するためにヒステリシスを持ちます。検出電圧のヒステリシスは $V_{BOT+}=V_{BOT}+V_{HYST}/2$ 、 $V_{BOT-}=V_{BOT}-V_{HYST}/2$ と解釈されるべきです。

BODが許可され、VCCが起動電圧以下に下降すると(図9-5の V_{BOT-})、低電圧リセットが直ちに有効とされます。VCCが起動電圧以上に上昇すると(図9-5の V_{BOT+})、(遅延タイマが起動され)遅延タイマは遅延時間(t_{TOUT})経過後にMCUを始動します。

BOD回路は電圧が表9-3で与えられる t_{BOD} 時間より長く起動電圧以下に留まる場合のみ、VCCでの低下を検出します。

図9-5. 動作中の低電圧検出リセット

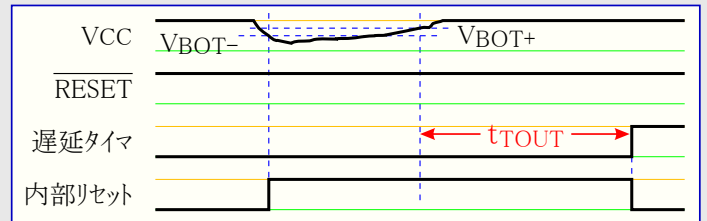


表9-3. 低電圧検出(BOD)特性

シンボル	項目	最小	代表	最大	単位
V_{HYST}	低電圧検出ヒステリシス電圧		50		mV
t_{BOD}	最小低電圧検出時間		2		μs

表9-2. BODLEVELヒューズ (V_{BOT}) 設定 (注1)

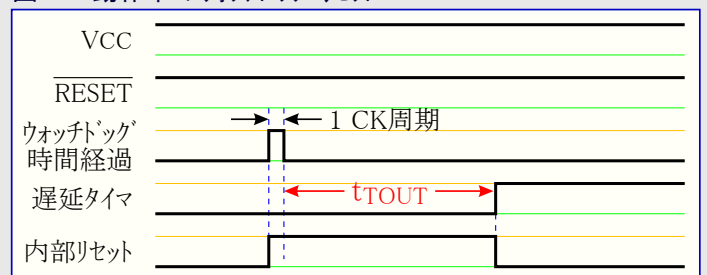
BODLEVEL2~0	000	001	010	011	100	101	110	111
最大	4.5	4.2	3.8	3.7	3.2	3.1	2.9	低電圧検出 (BOD) リセット禁止
代表	4.3	4.0	3.6	3.5	3.0	2.9	2.7	
最小	4.1	3.8	3.4	3.3	2.8	2.7	2.5	
単位	V							

注1: いくつかのデバイスで V_{BOT} が公称最低動作電圧以下の可能性があります。この状態のデバイスについては製造検査中、 $VCC=V_{BOT}$ に落として検査されています。これはマイクロコントローラの正しい動作がもはや保証されない電圧になる前に低電圧検出(BOD)リセットが起きることを保証します。

9.6. 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)1 CK周期幅の短いリセットパルスを生成します。本パルスの下降端で遅延タイマは遅延時間(t_{TOUT})の計時を始めます。ウォッチドッグタイマ操作の詳細については34頁を参照してください。

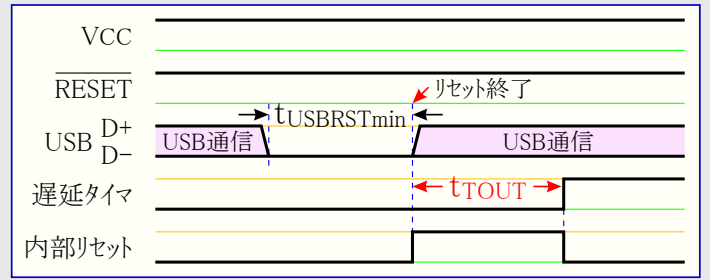
図9-6. 動作中のウォッチドッグリセット



9.7. USBリセット

USB部が許可され、**USBリセットMCU機能が許可で設定**され、そして有効なUSBリセット信号が検出されると、USB部が許可に留まることを除いてマイクロコントローラはリセットされます。これはリセット中とリセット後にバスへの接続を保つ一方で、ファームウェアの信頼性強化をデバイスに許します。

図9-7. 動作中のUSBリセット



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

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	—	WDRF	BORF	EXTRF	PORF	MCUSR
Read/Write	R	R	R/W	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7,6 – 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 – USBRF : USBリセット フラグ (USB Reset Flag)

このビットは**USBリセット**が起これと設定(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 – 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 – 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**が起これと設定(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**が起これと設定(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**が起これと設定(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**が起これと設定(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れ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。

9.8. 内部基準電圧

AT90USB82/162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、アナログ比較器の入力としても使えます。

9.8.1. 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9-4.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)

従って低電圧検出(BOD)が許可されていないと、ACBGの設定(=1)後、使用者はアナログ比較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2つの状態を避けられます。

表9-4. 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=2.7~5.5V	1.0	1.1	1.2	V
t _{BG}	起動時間			40	70	μs
I _{BG}	消費電流			10		μA

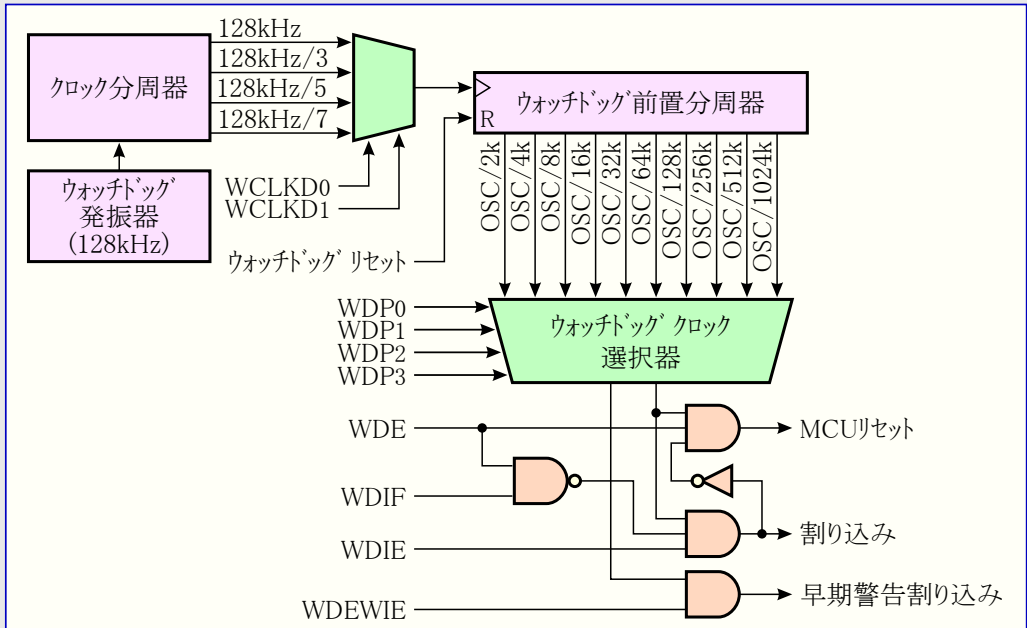
注: これらの値は目安で、実際はTBDです。

9.9. ウォッチドッグ タイマ

AT90USB82/162は強化されたウォッチドッグ タイマ (WDT)を持ちます。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～112sの計時完了時間
- 安全動作のウォッチドッグ常時ONハードウェア ヒューズが可能
- 1経過時間到達後の早期警告、2経過時間到達後の設定可能なリセット (操作動作参照)

図9-8. ウォッチドッグ タイマ構成図



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早期警告割り込みを生じます。WDTは計数器が与えられた計時完了値に2度達した時に割り込みまたはシステム リセットを生じます。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、システムはウォッチドッグ 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2度の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、または一般的なシステム 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2度の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による、他の2つの動作種別の組み合わせであ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。

これらの動作種別に加えて、WDT計数器の初回経過時に割り込みを生成するために、早期警告割り込みが許可できます。

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。

WDT前置分周器が2のべき乗分周係数(2,4,8,...)だけを許す一方で、このWDT周辺機能はクロック元で直接的に働くクロック分周器も含まれます。この分周器は奇数分周係数(3,5,7)を扱います。前置分周器との組み合わせで大きな経過時間数値が得られます。

この分周係数変更は最初に**システムリセット動作(WDE)ビット**と**ウォッチドッグ変更許可(WDCE)ビット**が設定(1)されなければならない、そして**ウォッチドッグタイマクロック分周(WDTCKD)レジスタ**内に新規分周値を格納するのに4周期が利用可能な時間制限手順によっても規則化されます。本操作後、WDEが設定(1)に留まることに注意してください。従って操作順の重要性に留意してください。前置分周器と分周器の指定値とで割り込み動作でWDTを設定するとき、分周レジスタ(WDTCKD)は前置分周レジスタ(WDTCSR)に先立って設定されなければならない。

1. WDEとWDCEを設定(1)します。
2. WDTCKD内に分周係数を設定します。
3. WDCEが自動的に解除(0)されるのを待ちます(2周期以上待機)。
4. 再びWDEとWDCEを設定(1)します。
5. 同一操作内でWDEを解除(0)、**割り込み動作(WDIE)**を設定(1)し、WDTCSR内に前置分周係数を設定します。
6. 今やシステムは割り込み専用動作に正しく設定されています。この2つの操作の反転操作は「割り込み及びシステムリセット動作」に帰着し、WDEを解除(0)するために3つ目の操作を必要とするでしょう。

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「コード例について」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)、デバイスがリセットし、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_PRG:    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<<WDP0) ; 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<<WDP0); /* 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「コード例について」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。

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

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2度起こると、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9-5. 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 (非プログラム)	0	0	停止	なし
1 (非プログラム)	0	1	割り込み	割り込み
1 (非プログラム)	1	0	システムリセット	リセット
1 (非プログラム)	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 (プログラム)	x	x	システムリセット	リセット

• **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表9-6~9で示されます。

9.9.2. ウォッチドッグタイマクロック分周レジスタ (Watchdog Timer Clock Divider Register) WDTCKD

ビット (\$62)	7	6	5	4	3	2	1	0	
	–	–	–	–	WDEWIF	WDEWIE	WCLKD1	WCLKD0	WDTCKD
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

• **ビット7~4 – 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 – WDEWIF : ウォッチドッグ早期警告割り込み要求フラグ (Watchdog Early Warning Interrupt Flag)**

ウォッチドッグ早期警告割り込み許可(WDEWIE)ビットが許可され、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、WDEWIFはハードウェアによって解除(0)されます。代わりにWDEWIF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WDEWIE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2 – WDEWIE : ウォッチドッグ早期警告割り込み許可 (Watchdog Early Warning Interrupt Enable)**

このビットがソフトウェアによって設定(1)されているとき、ウォッチドッグ早期警告割り込み要求フラグ(WDEWIF)がハードウェアによって設定(1)されると、ウォッチドッグ割り込みベクタで割り込みが生成されます。

• **ビット1,0 – WCLKD1,0 : ウォッチドッグタイマクロック分周選択 (Watchdog Timer Divider)**

- WCLKD=00 : $\text{ClkWDT} = \text{Clk128k}$
- WCLKD=01 : $\text{ClkWDT} = \text{Clk128k}/3$
- WCLKD=10 : $\text{ClkWDT} = \text{Clk128k}/5$
- WCLKD=11 : $\text{ClkWDT} = \text{Clk128k}/7$

表9-6. ウォッチドッグ前置分周選択, WCLKD=00 (ClkWDT=Clk128k)

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(1期間)		2k	4k	8k	16k	32k	64k	128k	256k	512k	1024k	(予約)					
代表的 計時完了 周期 (VCC=5V)	早期警告 割り込み	16ms	32ms	64ms	125ms	0.25s	0.5s	1.0s	2.0s	4.0s	8.0s						
	リセット/ 割り込み	32ms	64ms	128ms	0.25s	0.5s	1.0s	2.0s	4.0s	8.0s	16.0s						

表9-7. ウォッチドッグ前置分周選択, WCLKD=01 (ClkWDT=Clk128k/3)

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(1期間)		2k	4k	8k	16k	32k	64k	128k	256k	512k	1024k	(予約)					
代表的 計時完了 周期 (VCC=5V)	早期警告 割り込み	48ms	96ms	192ms	375ms	0.75s	1.5s	3s	6s	12s	24s						
	リセット/ 割り込み	96ms	192ms	384ms	0.75s	1.5s	3s	6s	12s	24s	48s						

表9-8. ウォッチドッグ前置分周選択, WCLKD=10 (ClkWDT=Clk128k/5)

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(1期間)		2k	4k	8k	16k	32k	64k	128k	256k	512k	1024k	(予約)					
代表的 計時完了 周期 (VCC=5V)	早期警告 割り込み	80ms	160ms	320ms	625ms	1.25s	2.5s	5s	10s	20s	40s						
	リセット/ 割り込み	160ms	320ms	640ms	1.25s	2.5s	5s	10s	20s	40s	80s						

表9-9. ウォッチドッグ前置分周選択, WCLKD=11 (ClkWDT=Clk128k/7)

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(1期間)		2k	4k	8k	16k	32k	64k	128k	256k	512k	1024k	(予約)					
代表的 計時完了 周期 (VCC=5V)	早期警告 割り込み	112ms	224ms	448ms	875ms	1.75s	3.5s	7s	14s	28s	56s						
	リセット/ 割り込み	224ms	448ms	896ms	1.75s	3.5s	7s	14s	28s	56s	112s						

10. 割り込み

本章はAT90USB82/162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。

10.1. AT90USB82/162の割り込みベクタ

表10-1. 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	INT4	外部割り込み要求4
7	\$000C	INT5	外部割り込み要求5
8	\$000E	INT6	外部割り込み要求6
9	\$0010	INT7	外部割り込み要求7
10	\$0012	PCINT0 (PCI0)	ピン変化0群割り込み要求
11	\$0014	PCINT1 (PCI1)	ピン変化1群割り込み要求
12	\$0016	USB一般	USB一般割り込み要求
13	\$0018	USBエントポイント/パイプ	USBエントポイント/パイプ割り込み要求
14	\$001A	ウォッチドッグ WDT	ウォッチドッグ計時完了
15	\$001C	タイマ/カウンタ1 CAPT	タイマ/カウンタ1捕獲発生
16	\$001E	タイマ/カウンタ1 COMPA	タイマ/カウンタ1比較A一致
17	\$0020	タイマ/カウンタ1 COMPB	タイマ/カウンタ1比較B一致
18	\$0022	タイマ/カウンタ1 COMPC	タイマ/カウンタ1比較C一致
19	\$0024	タイマ/カウンタ1 OVF	タイマ/カウンタ1溢れ
20	\$0026	タイマ/カウンタ0 COMPA	タイマ/カウンタ0比較A一致
21	\$0028	タイマ/カウンタ0 COMPB	タイマ/カウンタ0比較B一致
22	\$002A	タイマ/カウンタ0 OVF	タイマ/カウンタ0溢れ
23	\$002C	SPI STC	SPI 転送完了
24	\$002E	USART RX	USART 受信完了
25	\$0030	USART UDRE	USART 送信緩衝部空き
26	\$0032	USART TX	USART 送信完了
27	\$0034	アナログ比較器 ANA_COMP	アナログ比較器出力遷移
28	\$0036	EEPROM EE_RDY	EEPROM 操作可
29	\$0038	SPM命令 SPM_RDY	SPM命令操作可

注1: BOOTRSTヒューズがプログラム(0)されると、デバイスはリセットでブート ローダ アドレスへ飛びます。146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。

表10-2.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を置けます。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10-2. 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156頁の表24-8.で示されます。

10.1.1. 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。

(訳注) 本頁の記述は原書に従っていますが、AT90USB82について例外的に2語/1ベクタとなるのかは不明です。他のAVRと同様の場合は1語/1ベクタになります。

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

ビット	7	6	5	4	3	2	1	0	
\$35 (\$55)	—	—	—	PUD	—	—	IVSEL	IVCE	MCUCR
Read/Write	R	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146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146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 */
}

```

11. 外部割り込み

外部割り込みはINT7～0ピンまたはPCINT0～12ピンの何れかによって起動されます。許可したなら、例えばINT7～0またはPCINT0～12ピンが出力として設定されても、割り込みが起動することに注目してください。この特徴はソフトウェア割り込みを生成する方法を提供します。

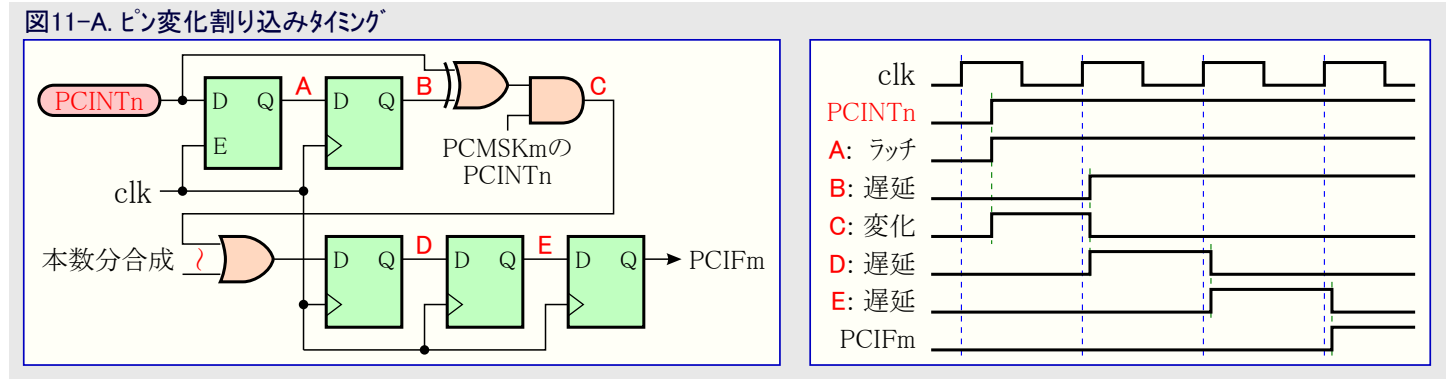
ピン変化割り込みPCIF0は許可したPCINT0～7の何れかが切り替わると起動します。ピン変化割り込み許可レジスタ0(PCMSK0)は、どのピンがピン変化割り込み要因となるかを制御します。ピン変化割り込みPCIF1は許可したPCINT8～12の何れかが切り替わると起動します。ピン変化割り込み許可レジスタ1(PCMSK1)は、どのピンがピン変化割り込み要因となるかを制御します。PCINT0～12のピン変化割り込みは非同期に検知されます。これはそれらの割り込みがアイドル動作以外の休止形態からもデバイスを起動するのに使うことを意味します。

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(INT7～4))の詳述で示される設定です。外部割り込みがレベル起動として設定、且つ許可されると、そのピンがLowに保持される限り、割り込みは(継続的に)起動します。INT7～4の上昇端または下降端割り込みの認知は23頁の「クロックシステムとその配給」で記述されるI/Oクロックの存在を必要とすることに注意してください。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16頁の「システムクロックとクロック選択」で示されるようにSUTヒューズとCKSELヒューズで定義されます。

11.1. 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11-Aで示されます。



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

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)とレベルは表11-1で定義されます。INT3～0の端は非同期に記録されます。表11-2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を書くことによって解除(0)されるべきです。

表11-1. 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11-2. 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

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

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

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

外部割り込み7~4はステータスレジスタ(SREG)の全割り込み許可(I)ビットと外部割り込み許可レジスタ(EIMSK)の対応する割り込み許可ビットが設定(1)される場合のINT7~4外部ピンによって活性(有効)にされます。割り込みを活性にする外部INT7~4ピンの端(エッジ)とレベルは表11-3.で定義されます。INT7~4ピンの値は端検知前に採取されています。端または論理変化割り込みが選ばれる場合、1クロック周期より長く留まるパルスは割り込みを生成します。より短いパルスは割り込みの発生が保証されません。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11-3. 外部割り込み7~4 割り込み条件

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

注: n=7~4

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

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

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

• ビット7~0 – INT7~INT0 : 外部割り込み7~0許可 (External Interrupt Request 7~0 Enable)

ステータスレジスタ(SREG)の全割り込み許可(I)ビットが設定(1)され、INT7~0ビットが1を書かれると、対応する外部ピン割り込みが許可されます。外部割り込み制御レジスタ(EICRAとEICRB)の割り込み条件制御n(ISCn1,ISCn0)ビットは、その外部割り込みが上昇端、下降端、両端、またはLowレベルのどれで活性(有効)にされるかを定義します。例え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。

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

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

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

INT7~0ピン上の端(エッジ)または論理変化が割り込み要求を起動すると、INTF7~0が設定(1)になります。ステータスレジスタ(SREG)の全割り込み許可(I)ビットと外部割り込み許可レジスタ(EIMSK)の対応する割り込み許可(INT7~0)ビットが設定(1)なら、MCUは対応する割り込みベクタへ飛びます。このフラグは割り込みルーチンが実行されると解除(0)されます。代わりにこのフラグは論理1を書くことによっても解除(0)できます。INT7~0がレベル割り込みとして設定されると、このフラグは常に解除(0)されます。INT3~0割り込み禁止で休止形態へ移行すると、これらのピンの入力緩衝部が禁止されることに注意してください。これはINTF3~0フラグを設定(1)する内部信号の論理変化を起こすかもしれません。より多くの情報については46頁の「デジタル入力許可と休止形態」をご覧ください。

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

ビット (\$68)	7	6	5	4	3	2	1	0	
	—	—	—	—	—	—	PCIE1	PCIE0	PCICR
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) (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1,0 – PCIE1,PCIE0 : ピン変化0/1群割り込み許可 (Pin Change Interrupt Enable 1,0)

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、このPCIE1/0ビットが設定(1)されると、ピン変化割り込み1/0群が許可されます。許可したPCINT12～8/7～0ピンの何れかの変化が割り込みを起こします。このピン変化割り込み要求に対応する割り込みはPCIE1/0割り込みベクタから実行されます。PCINT12～8/7～0ピンはピン変化割り込み許可レジスタ1/0(PCMSK1/0)によって個別に許可されます。

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

ビット \$1B (\$3B)	7	6	5	4	3	2	1	0	
	—	—	—	—	—	—	PCIF1	PCIF0	PCIFR
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) (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1,0 – PCIF1,PCIF0 : ピン変化0/1群割り込み要求フラグ (Pin Change Interrupt Flag 1,0)

PCINT12～8/7～0ピンの何れかの論理変化が割り込み要求を起動すると、PCIF1/0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)ビットとピン変化割り込み制御レジスタ(PCICR)のピン変化割り込み1/0群許可(PCIE1/0)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。

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

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PCINT7～0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。PCINT7～0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。PCINT7～0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。

11.1.8. ピン変化割り込み許可レジスタ1 (Pin Change Enable Mask 12～8) PCMSK1

ビット (\$6C)	7	6	5	4	3	2	1	0	
	—	—	—	PCINT12	PCINT11	PCINT10	PCINT9	PCINT8	PCMSK1
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

- ビット4～0 – PCINT12～PCINT8 : ピン変化割り込み12～8許可 (Pin Change Enable Mask 12～8)

各PCINT12～8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。PCINT12～8とピン変化割り込み制御レジスタ(PCICR)のPCIE1が設定(1)なら、対応するI/Oピンのピン変化割り込みが許可されます。PCINT12～8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。

12. 入出力ポート

12.1. 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12-1.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172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54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47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。

12.2. 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12-2.はここで属にPx_n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12-1. 入出力ピン等価回路

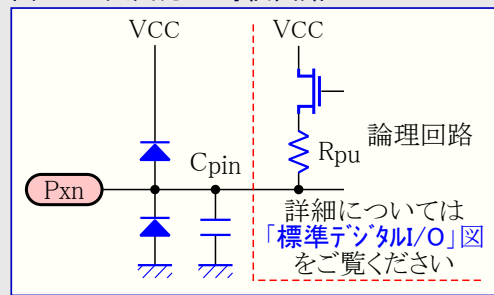
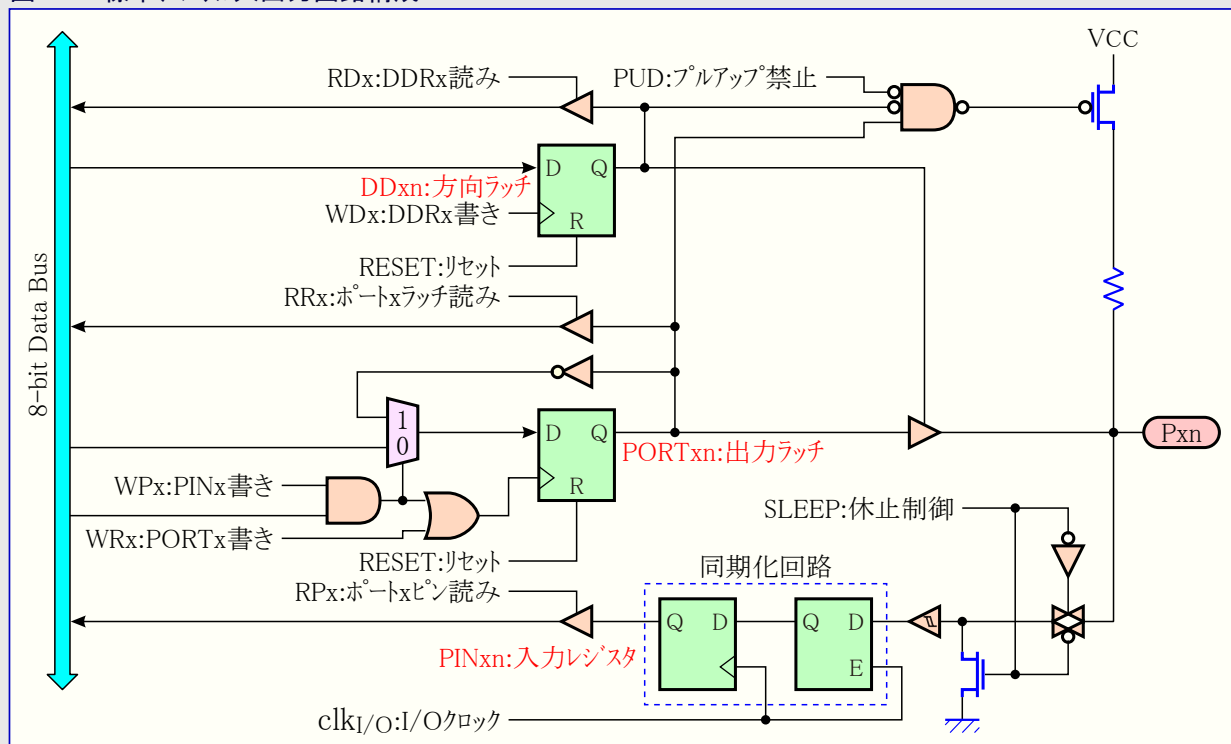


図12-2. 標準デジタル入出力回路構成



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。

12.2.1. 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。54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。

12.2.2. 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。

12.2.3. 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12-1. はピン値に対する制御信号の一覧を示します。

表12-1. 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

12.2.4. 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12-2.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12-3.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 $t_{pd,min}$ と $t_{pd,max}$ で示されます。

(図12-3.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(トランスパレント)となります。この信号値はシステムクロックがLowになる時に保持(ラッチ)されます。それが続くクロックの上昇端でPINxnレジスタに取り込まれます。2つの矢印 $t_{pd,min}$ と $t_{pd,max}$ によって示されるように、ピン上の単一信号遷移は出現時点に依存して0.5~1.5システムクロック周期遅らされます。

ソフトウェアが指定したピン値を読み戻す時は、図12-4.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(t_{pd})は1システムクロック周期です。

図12-3. 外部供給ピン値読み込み時の同期化

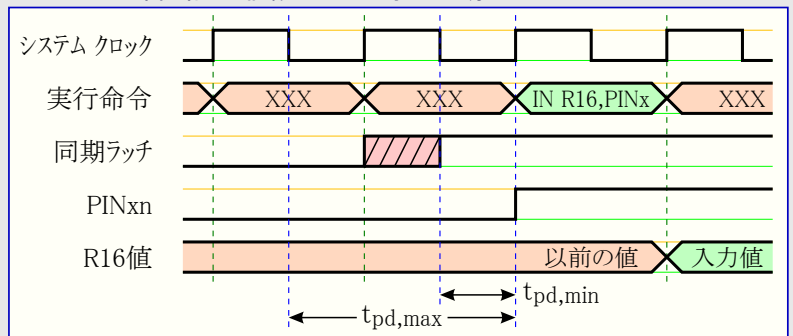
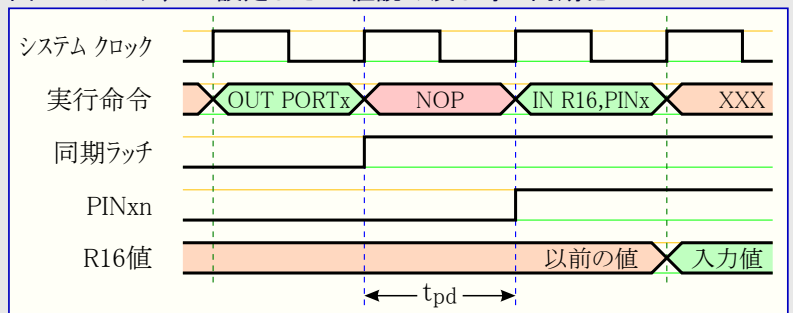


図12-4. プログラムで設定したピン値読み戻し時の同期化



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。

12.2.5. デジタル入力許可と休止形態

図12-2.で示されるようにデジタル入力信号はシュmitt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47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。

12.2.6. 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るとしても、デジタル入力が許可される他の全ての動作(リセット、活動動作、アイドル動作)で消費電流削減のため、浮き状態入力は避けられるべきです。

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

12.3. 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12-5は単純化された図12-2.でのポートピン制御信号が交換機能によってどう重複できるかを示します。この重複信号は全てのポートピンに存在する訳ではありませんが、この図はAVR マイクロ コントローラシステムの全ポートピンに適用できる一般的な記述として取り扱います。

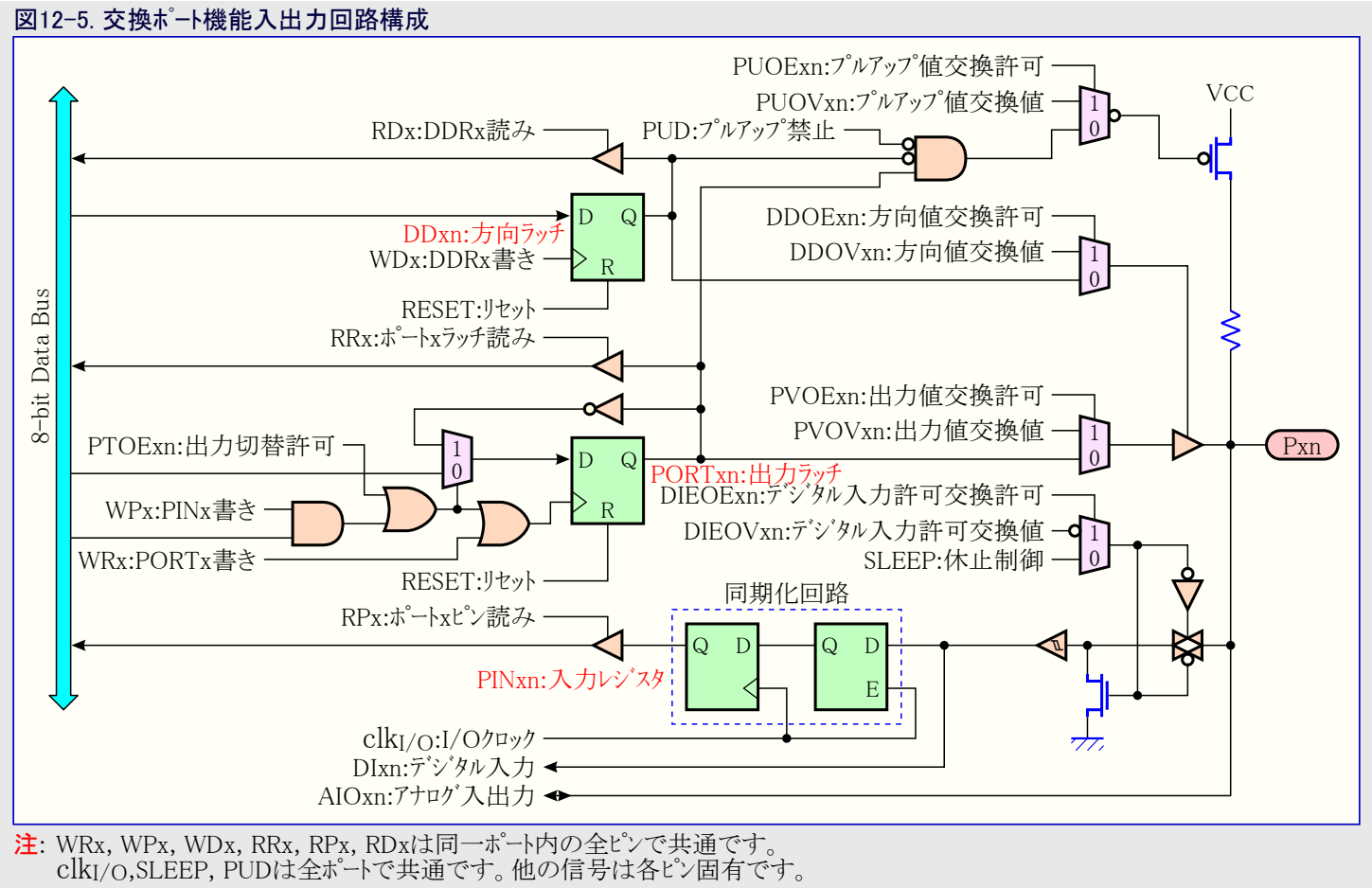


表12-2は重複(交換)信号の機能一覧を示します。図12-5.で示すピンとポートは次表で示されません。重複(交換)信号は交換機能を持つ機能部で内部的に生成されます。

表12-2. 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。

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

ビット	7	6	5	4	3	2	1	0	
\$35 (\$55)	—	—	—	PUD	—	—	IVSEL	IVCE	MCUCR
Read/Write	R	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45頁の「[ピン](#)の設定」をご覧ください。

12.3.2. ポートBの交換機能

ポートBピンの交換機能は表12-3.で示されます。

表12-3. 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) 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	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	PCINT5 (ピン変化割り込み5入力)	PB1	SCLK (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	T1 (タイマ/カウンタ1 外部クロック入力) PCINT4 (ピン変化割り込み4入力)	PB0	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/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動作出力ピンでもあります。

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。

• PCINT6 – ポートB ビット6 : PB6

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。

• PCINT5 – ポートB ビット5 : PB5

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。

• T1/PCINT4 – ポートB ビット4 : PB4

T1 : タイマ/カウンタ1の外部クロック入力ピンです。

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AT90USB82/162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(DDRB)のDDB3の設定に拘らず、入力として設定されます。SPIが従装置として許可されると、このピンのデータ方向はDDB3によって制御されます。このピンがSPIによって入力を強制されるとき、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せん。

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AT90USB82/162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の設定に拘らず、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。

• SCLK/PCINT1 – ポートB ビット1 : PB1

SCLK : 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時、プルアップは未だポートB出力レジスタ(PORTB)のPORTB1によって制御できません。このピンは直列プログラミング インターフェース用のクロックとしても扱います。

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12-4.と表12-5.は、ポートBの交換機能を47頁の図12-5.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12-4. ポートB7~4の交換機能用交換信号

信号名	PB7/OC0A/OC1C/PCINT7	PB6/PCINT6	PB5/PCINT5	PB4/T1/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	0	0	0
PVOV	OC0A/OC1C	0	0	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	T1/PCINT4入力
AIO	–	–	–	–

表12-5. 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/SCLK/PCINT1	PB0/ $\overline{SS}$ /PCINT0
PUOE	SPE・MSTR	SPE・MSTR	SPE・MSTR	SPE・MSTR
PUOV	PORTB3・PUD	PORTB2・PUD	PORTB1・PUD	PORTB0・PUD
DDOE	SPE・MSTR	SPE・MSTR	SPE・MSTR	SPE・MSTR
DDOV	0	0	0	0
PVOE	SPE・MSTR	SPE・MSTR	SPE・MSTR	0
PVOV	SPI従装置出力	SPI主装置出力	SCK出力	0
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	SCLK/PCINT1入力	SPI $\overline{SS}$ /PCINT0入力
AIO	–	–	–	–

12.3.3. ポートCの交換機能

ポートCの交換機能は次のとおりです。

表12-6. 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	INT4 (外部割り込み4 入力) ICP1 (タイマ/カウンタ1 捕獲起動入力) CLKO (システム クロック出力)	–	–
PC6	OC1A (タイマ/カウンタ1 比較A一致/PWM出力) PCINT8 (ピン変化割り込み8入力)	PC2	PCINT11 (ピン変化割り込み11入力)
PC5	OC1B (タイマ/カウンタ1 比較B一致/PWM出力) PCINT9 (ピン変化割り込み9入力)	PC1	RESET (外部リセット入力) dW (デバッグWIRE機能入出力)
PC4	PCINT10 (ピン変化割り込み10入力)	PC0	XTAL2 (外部用発振増幅器出力)

交換ピンの設定は次のとおりです。

- INT4/ICP1/CLKO – ポートC ビット7 : PC7

INT4 : 外部割り込み4入力。PC7ピンはMCUへの外部割り込み元として扱えます。

ICP1 : タイマ/カウンタ1の捕獲起動入力。PC7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。

CLKO : システム クロック出力。分周したシステム クロックがPC7ピンに出力できます。分周したシステム クロックはCKOUTヒューズがプログラム(0)されると、PORTC7とDDC7設定に拘らず、出力されます。これはリセット中にも出力されます。

- OC1A/PCINT8 – ポートC ビット6 : PC6

OC1A : タイマ/カウンタ1の比較A一致出力。PC6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(DDC6=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。

PCINT8 : ピン変化割り込み8入力。PC6ピンは外部割り込み元としても扱えます。

- OC1B/PCINT9 – ポートC ビット5 : PC5

OC1B : タイマ/カウンタ1の比較B一致出力。PC5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(DDC5=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。

PCINT9 : ピン変化割り込み9入力。PC5ピンは外部割り込み元としても扱えます。

- PCINT10 – ポートC ビット4 : PC4

PCINT10 : ピン変化割り込み10入力。PC4ピンは外部割り込み元としても扱えます。

- PCINT11 – ポートC ビット2 : PC2

PCINT11 : ピン変化割り込み11入力。PC2ピンは外部割り込み元としても扱えます。

- RESET/dW – ポートC ビット1 : PC1

RESET : リセット入力。外部リセット入力はRSTDISBLヒューズの非プログラム(1)によって許可され、Low活性(有効)です。本ピンがRESETピンとして使われると、プルアップが活性(許可)にされ、出力駆動部とデジタル入力は非活性(禁止)にされます。

dW : デバッグWIREチャネル。デバッグWIRE許可(DWEN)ヒューズがプログラム(0)され、施錠ビットが非プログラム(1)にされると、目的デバイス内のデバッグWIREシステムが活性にされます。dWピンはプルアップ許可のワイヤードAND(オープンドレイン)双方向I/Oピンとして設定され、エミュレータと目的デバイス間の通信路になります。

- XTAL2 – ポートC ビット0 : PC0

XTAL2 : 発振部出力。PC0ピンは内蔵発振増幅器の反転出力として扱えます。

表12-7.と表12-8.はポートCの交換機能を47頁の図12-5.で示される交換信号に関連付けます。

表12-7. ポートC7～4の交換機能用交換信号

信号名	PC7/INT4/ICP1/CLKO	PC6/OC1A/PCINT8	PC5/OC1B/PCINT9	PC4/PCINT10
PUOE	0	0	0	0
PUOV	0	0	0	0
DDOE	0	0	0	0
DDOV	0	0	0	0
PVOE	0	OC1A許可	OC1B許可	0
PVOV	0	OC1A	OC1B	0
PTOE	–	–	–	–
DIOE	INT4許可	PCINT8許可	PCINT9許可	PCINT10許可
DIOV	1	1	1	1
DI	INT4入力	PCINT8入力	PCINT9入力	PCINT10入力
AIO	–	–	–	–

表12-8. ポートC2～0の交換機能用交換信号

信号名		PC2/PCINT11	PC1/RESET/dW	PC0/XTAL2
PUOE		0	0	0
PUOV		0	0	0
DDOE		0	0	0
DDOV		0	0	0
PVOE		0	0	0
PVOV		0	0	0
PTOE		–	–	–
DIOE		PCINT11許可	0	0
DIOV		1	0	0
DI		PCINT11入力	–	–
AIO		–	–	–

12.3.4. ポートDの交換機能

ポートDピンの交換機能は表12-9.で示されます。

表12-9. 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	INT7 (外部割り込み7 入力)	PD3	INT3 (外部割り込み3 入力)
	T0 (タイマ/カウンタ0 外部クロック入力)		TXD (USART 送信データ出力)
	CTS (USART送信許可入力)	PD2	INT2 (外部割り込み2 入力)
	HWB (ブートローダ強制実行入力)		RXD (USART 受信データ入力)
PD6	INT6 (外部割り込み6 入力)	PD1	AIN1 (アナログ比較器反転入力)
	RTS (USART送信要求出力)		INT1 (外部割り込み1 入力)
PD5	XCK (USART 外部クロック入出力)	PD0	AIN0 (アナログ比較器非反転入力)
	PCINT12 (ピン変化割り込み12入力)		INT0 (外部割り込み0 入力)
PD4	INT5 (外部割り込み5 入力)		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。

• INT7/T0/CTS/HWB – ポートD ビット7 : PD7

INT7 : 外部割り込み7入力。PD7ピンはMCUへの外部割り込み元として扱えます。

T0 : タイマ/カウンタ0の外部クロック入力ピンです。

CTS : USARTの送信部流れ制御入力。このピンはその状態の機能によって送信部を制御できます。

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、リセット後のブートローダ部実行を許します。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。

• INT6/RTS – ポートD ビット6 : PD6

INT6 : 外部割り込み6入力。PD6ピンはMCUへの外部割り込み元として扱えます。

RTS : USARTの受信部流れ制御出力。このピンはその状態の機能によって受信部を制御できます。

• XCK/PCINT12 – 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)。

PCINT12 : ピン変化割り込み12入力。PD5ピンは外部割り込み元としても扱えます。

• INT5 – ポートD ビット4 : PD4

INT5 : 外部割り込み5入力。PD4ピンはMCUへの外部割り込み元として扱えます。

• 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/AIN1 – 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。

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

• INT1/AIN0 – 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。

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。

• INT0/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。

OC0B : タイマ/カウンタ0の比較B一致出力。PD0ピンはタイマ/カウンタ0の比較B一致用外部出力として扱えます。この機能を扱うため、このピンは出力として設定(DDD0=1)されなければなりません。このOC0Bピンはタイマ機能のPWM動作出力ピンでもあります。

表12-10.と表12-11.はポートDの交換機能を47頁の図12-5.で示される交換信号に関連付けます。

表12-10. ポートD7～4の交換機能用交換信号

信号名	PD7/INT7/T0/CTS/HWB	PD6/INT6/RTS	PD5/XCK/PCINT12	PD4/INT5
PUOE	CTS	RTS	0	0
PUOV	PORTD7・PUD	0	0	0
DDOE	CTS	RTS	0	0
DDOV	0	1	0	0
PVOE	0	RTS出力許可	XCK出力許可	0
PVOV	0	RTS出力	XCK出力	0
PTOE	–	–	–	–
DIOE	INT7/CTS許可	INT6許可	PCINT12許可	INT5許可
DIOV	1	1	1	1
DI	INT7/T0/CTS入力	INT6入力	XCK/PCINT12入力	INT5入力
AIO	–	–	–	–

表12-11. ポートD3～0の交換機能用交換信号

信号名	PD3/INT3/TXD	PD2/INT2/RXD/AIN1	PD1/INT1/AIN0	PD0/INT0/OC0B
PUOE	TXEN	RXEN	0	0
PUOV	0	PORTD2・PUD	0	0
DDOE	TXEN	RXEN	0	0
DDOV	1	0	0	0
PVOE	TXEN	0	0	OC0B許可
PVOV	TXD	0	0	OC0B
PTOE	–	–	–	–
DIOE	INT3許可	INT2/AIN1許可	INT1/AIN0許可	INT0許可
DIOV	1	AIN1許可	AIN0許可	1
DI	INT3入力	INT2入力/RXD	INT1入力	INT0入力
AIO	–	AIN1入力	AIN0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。

12.4. I/Oポート用レジスタ

12.4.1. ポートB出カレジスタ (Port B Data Register) PORTB

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	

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

ビット	7	6	5	4	3	2	1	0	
\$04 (\$24)	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0	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	

12.4.3. ポートB入カレジスタ (Port B Input Address) PINB

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	

12.4.4. ポートC出カレジスタ (Port C Data Register) PORTC

ビット	7	6	5	4	3	2	1	0	
\$08 (\$28)	PORTC7	PORTC6	PORTC5	PORTC4	—	PORTC2	PORTC1	PORTC0	PORTC
Read/Write	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

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

ビット	7	6	5	4	3	2	1	0	
\$07 (\$27)	DDC7	DDC6	DDC5	DDC4	—	DDC2	DDC1	DDC0	DDRC
Read/Write	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

12.4.6. ポートC入カレジスタ (Port C Input Address) PINC

ビット	7	6	5	4	3	2	1	0	
\$06 (\$26)	PINC7	PINC6	PINC5	PINC4	—	PINC2	PINC1	PINC0	PINC
Read/Write	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	
初期値	不定	不定	不定	不定	0	不定	不定	不定	

12.4.7. ポートD出カレジスタ (Port D Data Register) PORTD

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	

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

ビット	7	6	5	4	3	2	1	0	
\$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	

12.4.9. ポートD入カレジスタ (Port D Input Address) PIND

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	

13. タイマ/カウンタ0とタイマ/カウンタ1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または1)は一般名として使われます。

13.0.1. 内部クロック

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。

13.0.2. 前置分周器リセット

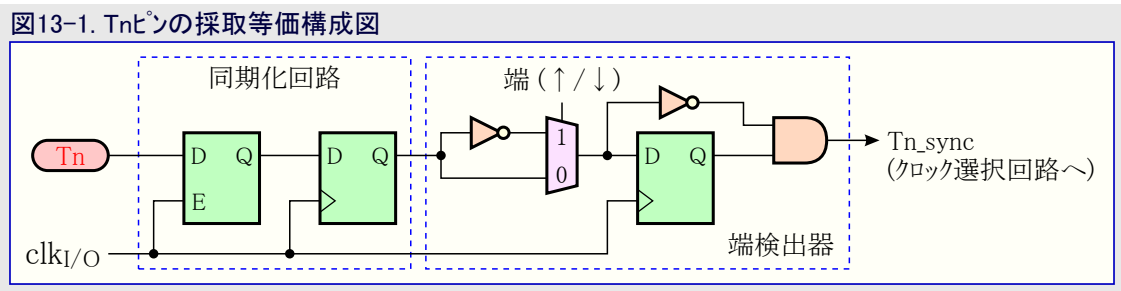
この前置分周器は自由走行で(換言するとタイマ/カウンタのクロック選択論理回路と無関係に動作する)、Tnによって共用されます。前置分周器はタイマ/カウンタのクロック選択によって影響を及ぼされないため、前置分周器の状態は前置分周したクロックが使われる状況に対して密接に関係します。タイマ/カウンタが許可され、前置分周器によってクロック駆動される(CSn2=0=5~2)とき、前置分周加工の一例が生じます。タイマ/カウンタが許可される時から最初の計数が起きるまでのシステムクロック周期数はNが前置分周値(8,64,256,1024)とすると、1~N+1システムクロック周期になり得ます。

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。

13.0.3. 外部クロック

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13-1.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。

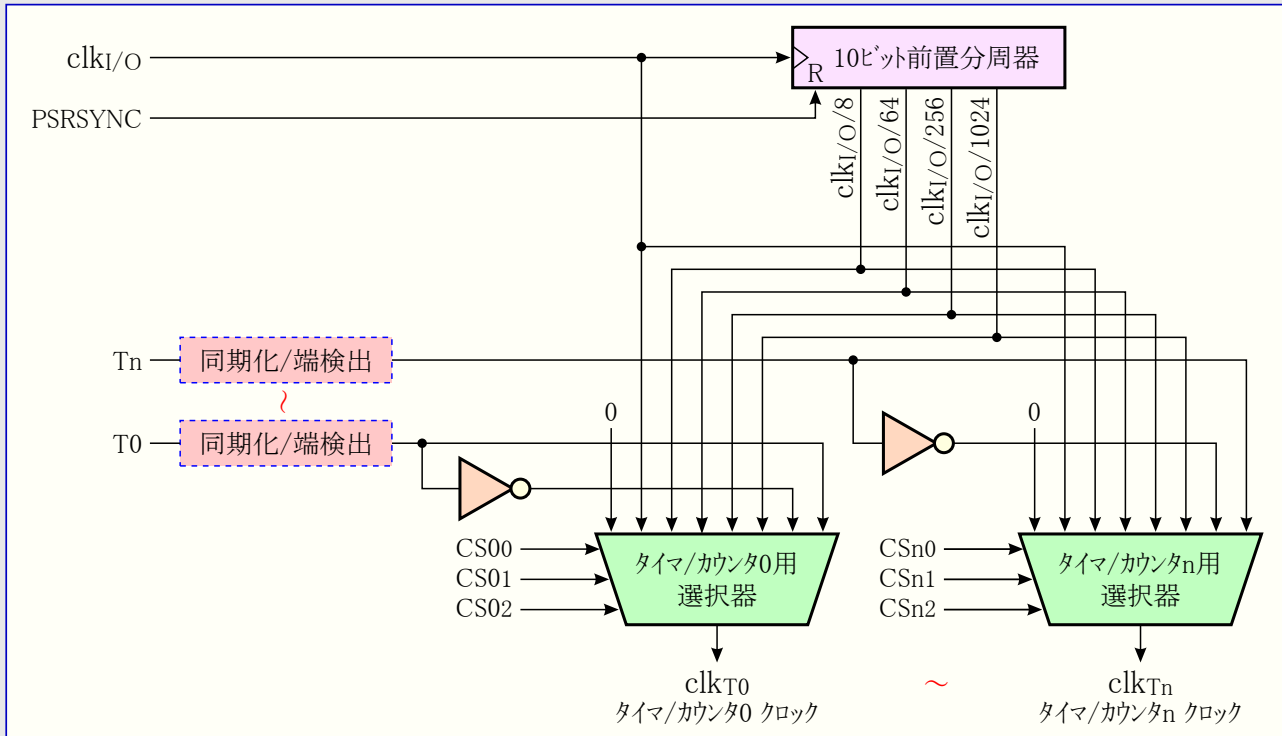


同期化と端検出器論理回路はTnピンへ印加された端からカウンタが更新されるまでに2.5~3.5システムクロック周期の遅延をもたらします。クロック入力の許可と禁止はTnが最低1システムクロック周期に対して安定してしまっている時に行われなければならない、さもなければ不正なタイマ/カウンタクロックパルスが生成される危険があります。

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らない。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らない。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(水晶発振子、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13-2. タイマ/カウンタ0とタイマ/カウンタ1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13-1.で示されます。

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

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ことを保証します。TSMビットが0を書かれると、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とタイマ/カウンタ1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いて、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とタイマ/カウンタ1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。

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)

この8ビット タイマ/カウンタの簡単化した構成図は図14-1.で示されます。I/Oピンの実際の配置については2頁の「ピン配置」を参照してください。CPUがアクセス可能な(I/OビットとI/Oピンを含む)I/Oレジスタは赤文字(訳注:原文は太字)で示されます。デバイス仕様のI/Oレジスタとビット位置は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。

図14-1. 8ビット タイマ/カウンタ構成図

[illegible]

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。

14.1.2. 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14-1.の定義は本文書を通して広範囲に渡って使われます。

表14-1. 用語定義

用語	意味
BOTTOM	タイマ/カウンタが\$00に到達した時。
MAX	タイマ/カウンタが\$FF(255)に到達した時。
TOP	タイマ/カウンタが指定された固定値(\$FF)または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。

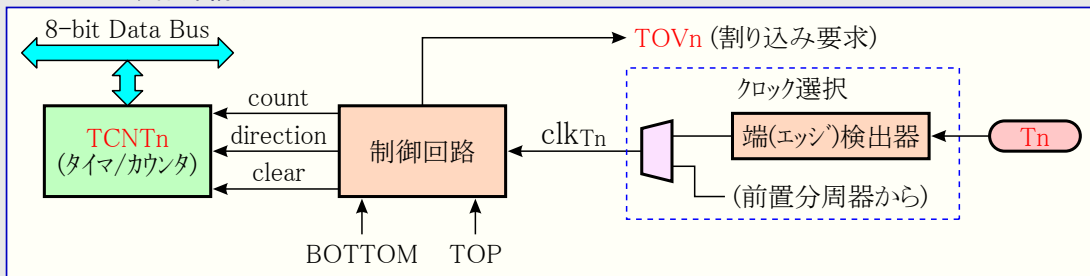
14.2. タイマ/カウンタのクロック

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とタイマ/カウンタ1の前置分周器」をご覧ください。

14.3. 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14-2.は、このカウンタとその周辺環境の構成図を示します。

図14-2. 計数器部構成図



信号説明 (内部信号)	説明
count	TCNT0を1つ進めるまたは戻す信号。
direction	進行方向(上昇または下降)選択信号。
clear	TCNT0のリセット(\$00設定)信号。
clkTn	以降でclkT0として参照されるタイマ/カウンタ クロック信号。
TOP	TCNT0が最大値に到達したことを示す信号。
BOTTOM	TCNT0が最小値(\$00)に到達したことを示す信号。

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)とき、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。

14.4. 比較出力部

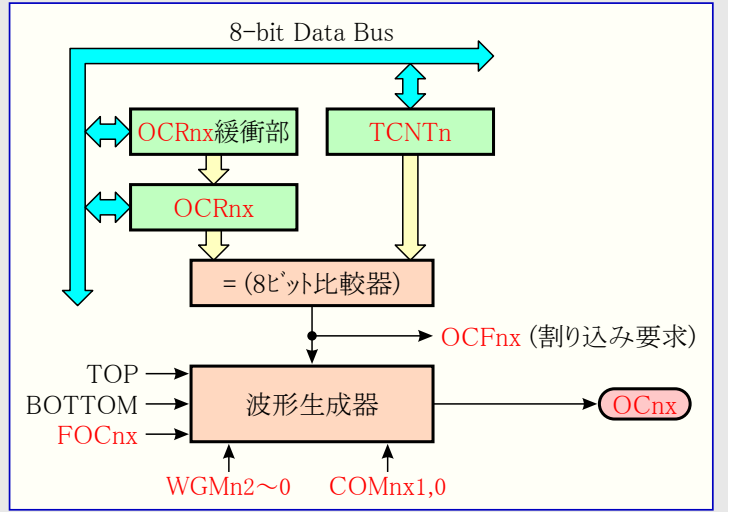
この8ビット比較器はTCNT0と比較レジスタ(OCR0AとOCR0B)を継続的に比較します。TCNT0がOCR0AまたはOCR0Bと等しければ比較器は一致を指示します。この一致は次のタイマ/カウンタクロック周期で比較割り込み要求フラグ(OCF0AまたはOCF0B)を設定(1)します。対応する割り込みが許可(I=1, OCIE0AまたはOCIE0B=1)されているならば、その比較割り込み要求フラグは比較割り込みを発生します。比較割り込み要求フラグは割り込みが実行されると自動的に解除(0)されます。代わりにこのフラグはこのI/Oビット位置に論理1を書くことによってソフトウェアでも解除(0)できます。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14-3. 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14-3. 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。

14.4.1. 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)。

14.4.2. 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。

14.4.3. 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。

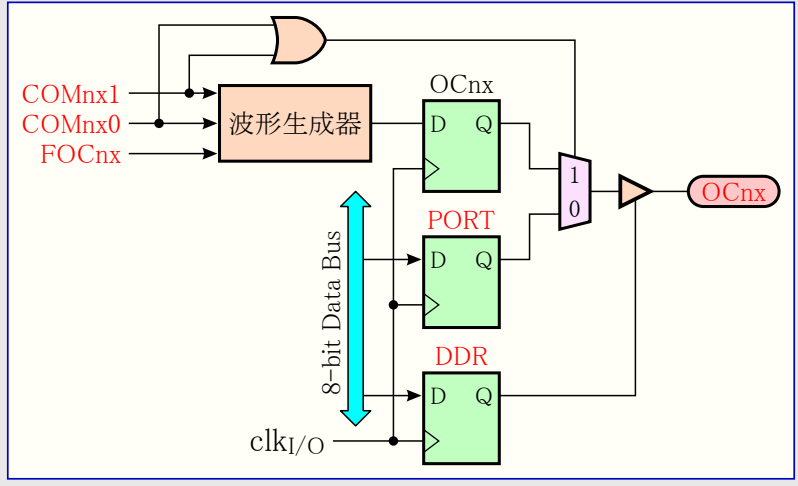
OC0xの初期設定はポートピンに対するポート方向レジスタを出力に設定する前に行われるべきです。OC0x値を設定する一番簡単な方法は標準動作で強制変更(FOC0x)スローブビットを使うことです。波形生成動作種別間を変更する時でも、OC0x(内部)レジスタはその値を保ちます。

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。

14.5. 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14-4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14-4. 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0用レジスタ」をご覧ください。

14.5.1. 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14-2と表14-5を参照してください。高速PWM動作については65頁の表14-3と表14-6、位相基準PWMについては65頁の表14-4と表14-7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($FOC0x$)ストップビットを使うことによって直ちに効果を得ることを強制できます。

14.6. 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/カウンタのタイミング」を参照してください。

14.6.1. 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。

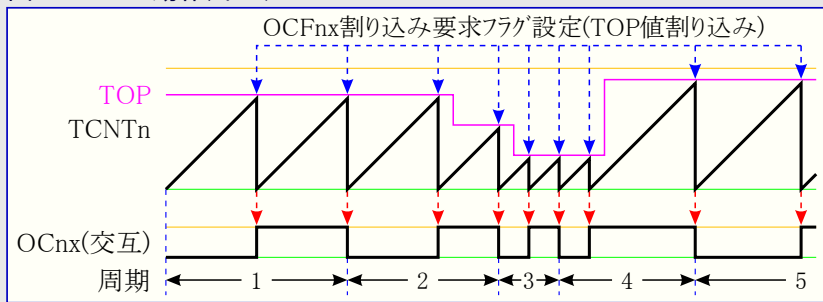
比較出力部は与えられた或る時間に割り込みを生成するのに使えます。標準動作で波形を生成するのに比較出力を使うのは、それが大変多くのCPU時間を占有するため推奨されません。

14.6.2. 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と一致すると、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14-5.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14-5. 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。その後のカウンタは比較一致が起こるのに先立って最大値(\$FF)へ、そして次に\$00から始める計数をしなければならないでしょう。

CTC動作で波形出力を生成するため、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。

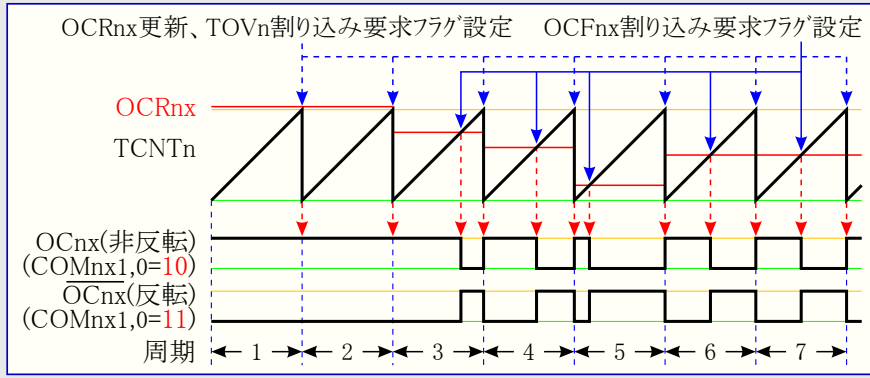
標準動作と同じように、タイマ/カウンタ溢れ(TOV0)フラグはカウンタがMAXから\$00へ計数するのと同じタイマ/カウンタクロック周期で設定(1)されます。

14.6.3. 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14-6.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14-6. 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14-3と表14-6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。

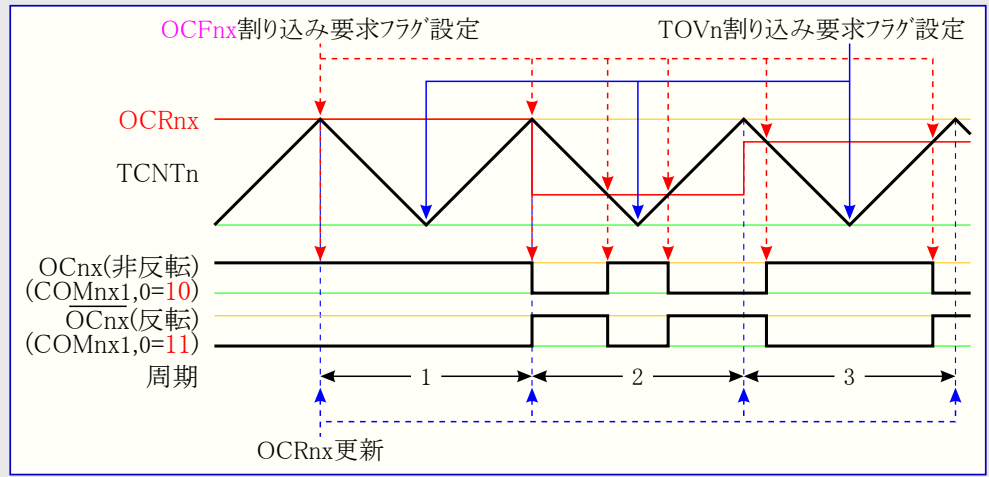
(訳補:WGM02~0=111の場合については、)高速PWM動作での(デューティ比50%)周波数の波形出力は比較一致毎に論理反転するOC0A設定(COM0A1,0=01)によって達成できます。生成された波形はOCR0Aが0(\$00)に設定される時に $f_{OC0x} = f_{clk_I/O} / 2$ の最大周波数でしょう。この特性は高速PWM動作で比較出力部の2重緩衝機能が許可されることを除いて、CTC動作でのOC0A交互出力(COM0A1,0=01)と同じです。

14.6.4. 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14-7.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)。

図14-7. 位相基準PWM動作タイミング



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

位相基準PWM動作での比較部は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14-4と表14-7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るでしょう。PWM波形はカウンタが増加する時のTCNT0とOCR0x間の比較一致でOC0x(内部)レジスタを設定(1)(または解除(0))と、カウンタが減少する時のTCNT0とOCR0x間の比較一致でOC0xレジスタを解除(0)(または設定(1))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当たります。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14-7.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14-7.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。)

14.7. 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14-8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14-8. 前置分周なし(1/1)のタイマ/カウンタ タイミング

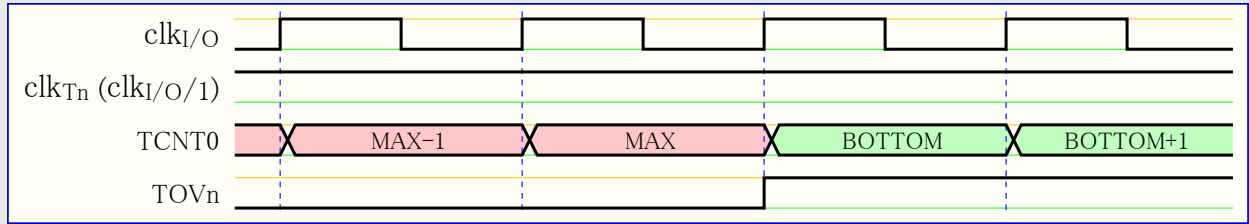


図14-9. は同じタイミング データを示しますが、前置分周器が許可されています。

図14-9. 前置分周器($f_{\text{clk}_I/O/8}$)のタイマ/カウンタ タイミング

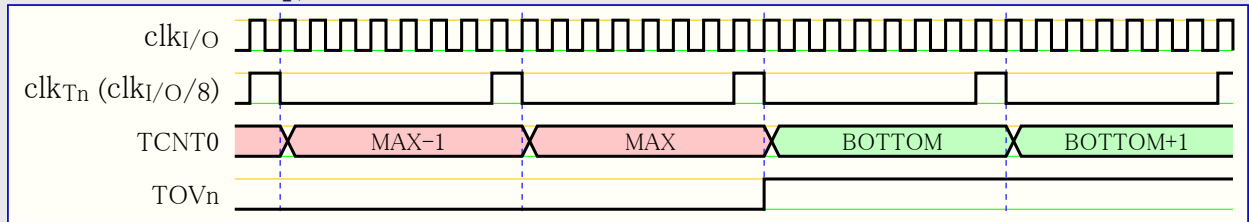


図14-10. はCTC動作とOCR0AがTOPのPWM動作を除く全動作種別でのOCF0Aと全動作種別でのOCF0Bの設定を示します。

図14-10. 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

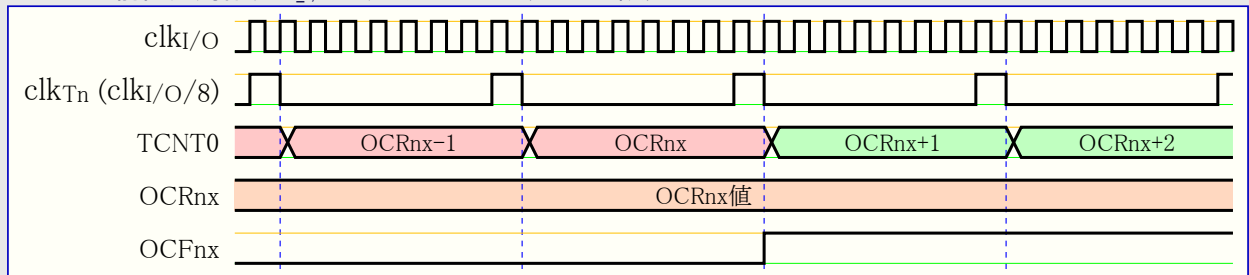
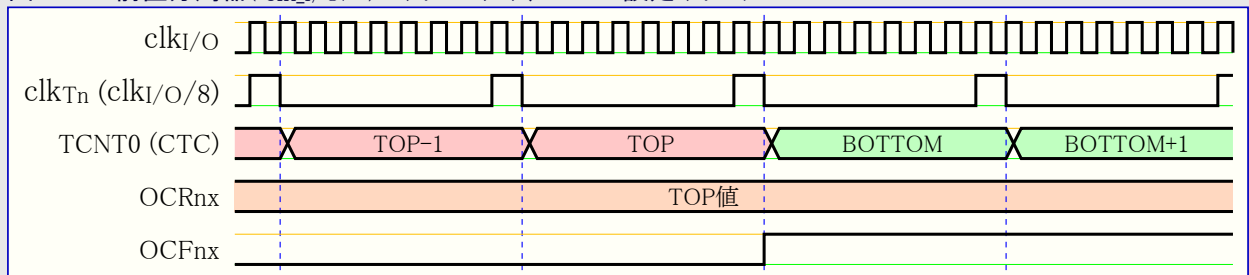


図14-11. はOCR0AがTOPの高速PWM動作と、CTC動作でのTCNT0の解除とOCF0Aの設定を示します。

図14-11. 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



14.8. 8ビット タイマ/カウンタ0用レジスタ

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

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14-2. 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14-3. 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14-4. 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14-2. 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14-3. 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)

表14-4. 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

• 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14-5. 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14-6. 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14-7. 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14-5. 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14-6. 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)

表14-7. 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

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14-3,4,6,7.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(表14-8参照)。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14-8. 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。

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

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「タイマ/カウンタ制御レジスタA(TCCR0A)」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14-9. 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、例えば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。

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

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。

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

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。

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

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。

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

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。

14.8.7. 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) TIFR0

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頁の波形生成種別ビット記述の表14-8を参照してください。

15. 16ビット タイマ/カウンタ (タイマ/カウンタ1)

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
- 捕獲入力雑音除去器
- 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
- 5つの独立した割り込み (TOV1, OCF1A, OCF1B, OCF1C, ICF1)

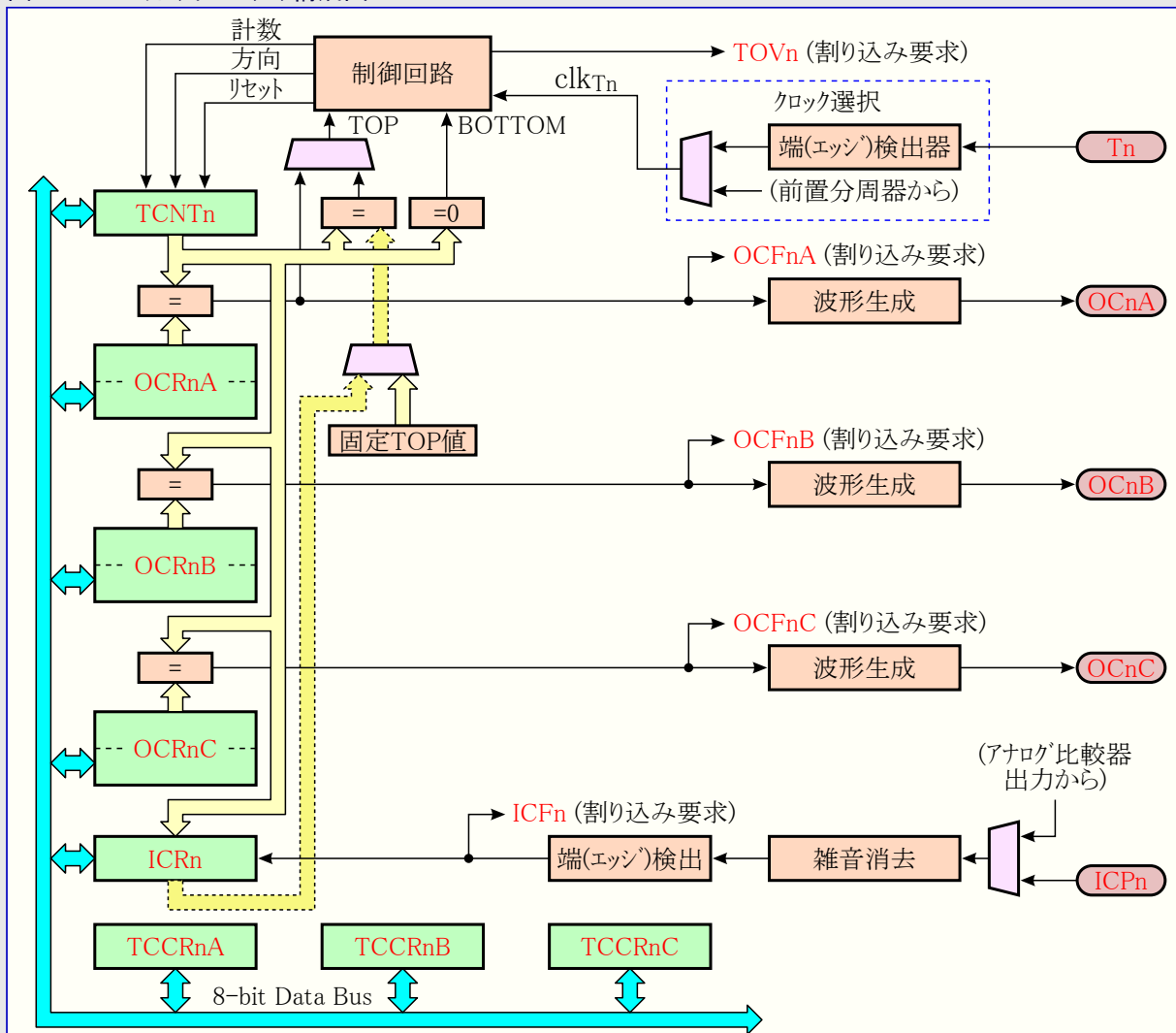
15.1. 概要

この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(本製品は1のみ利用可)、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/カウンタの簡略化した構成図は図15-1.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用レジスタ」で示されます。

28頁の「電力削減レジスタ(PRR0)」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。

図15-1. 16ビット タイマ/カウンタ構成図



注: タイマ/カウンタ1のピン配置と説明については2頁の「ピン配置」、48頁の表12-3、50頁の表12-6を参照してください。

15.1.1. 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(143頁の「アナログ比較器」参照)のどちらかの外部(端(エッジ))で起動された事象でタイマ/カウンタ値を捕獲(複写)できます。捕獲入力部は尖頭雑音を捕らえる機会を軽減するためにデジタル濾波部(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が代わりに使え、PWM出力として使われるべきOCRnAを開放します。

15.1.2. 定義

次の定義は本文書を通じて広範囲に使われます。

表15-1. 用語定義

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

15.2. 16ビットレジスタのアクセス

TCNTn, OCRnA, OCRnB, OCRnC, ICRnは8ビットバス経由でAVR CPUによってアクセスできる16ビットレジスタです。この16ビットレジスタは2回の読みまたは書き操作を使ってバイトアクセスされなければなりません。各々の16ビットタイマ/カウンタは16ビットアクセスの上位バイトの一時保存用に1つの8ビットレジスタを持ちます。16ビットタイマ/カウンタ内の全ての16ビットレジスタ間で、この同一一時レジスタが共用されます。下位バイトアクセスが16ビット読み書き動作を起動します。16ビットレジスタの下位バイトがCPUによって書かれると、一時レジスタに保存した上位バイトと書かれた下位バイトは同じクロック周期で両方が16ビットレジスタに複写されます。16ビットレジスタの下位バイトがCPUによって読まれると、16ビットレジスタの上位バイトは下位バイトが読まれるのと同じクロック周期で一時レジスタに複写されます。

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

16ビット書き込みを行うために上位バイトは下位バイトに先立って書かれなければなりません。16ビット読み込みについては下位バイトが上位バイトの前に読まれなければなりません。

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

アセンブリ言語プログラム例

```

~
LDI    R17, $01
LDI    R16, $FF
OUT    TCNTnH, R17
OUT    TCNTnL, R16

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

```

C言語プログラム例

```

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

```

注: 5頁の「コード例について」をご覧ください。

アセンブリ言語コード例はR17:R16レジスタ対にTCNTn値を戻します。

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

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

アセンブリ言語プログラム例

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

C言語プログラム例

```
unsigned int TIM16_Read_TCNTn(void)
{
    unsigned char sreg;                /* ステータスレジスタ一時保存変数定義 */
    unsigned int i;                   /* TCNTn読み出し変数定義 */
    sreg = SREG;                      /* 現全割り込み許可フラグ(I)を保存 */
    __disable_interrupt();            /* 全割り込み禁止 */
    i = TCNTn;                        /* TCNTn値を取得 */
    SREG = sreg;                      /* 全割り込み許可フラグ(I)を復帰 */
    return i;                         /* TCNTn値で呼び出し元へ復帰 */
}
```

注: 5頁の「コード例について」をご覧ください。

アセンブリ言語コード例はR17:R16レジスタ対にTCNTn値を戻します。

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

アセンブリ言語プログラム例

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

C言語プログラム例

```
void TIM16_Write_TCNTn(unsigned int i)
{
    unsigned char sreg;                /* ステータスレジスタ一時保存変数定義 */
    unsigned int i;                   /* TCNTn書き込み変数定義 */
    sreg = SREG;                      /* 現全割り込み許可フラグ(I)を保存 */
    __disable_interrupt();            /* 全割り込み禁止 */
    TCNTn = i;                       /* TCNTn値を設定 */
    SREG = sreg;                      /* 全割り込み許可フラグ(I)を復帰 */
}
```

注: 5頁の「コード例について」をご覧ください。

アセンブリ言語コード例はR17:R16レジスタ対がTCNTnへ書かれるべき値を含む必要があります。

15.2.1. 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。

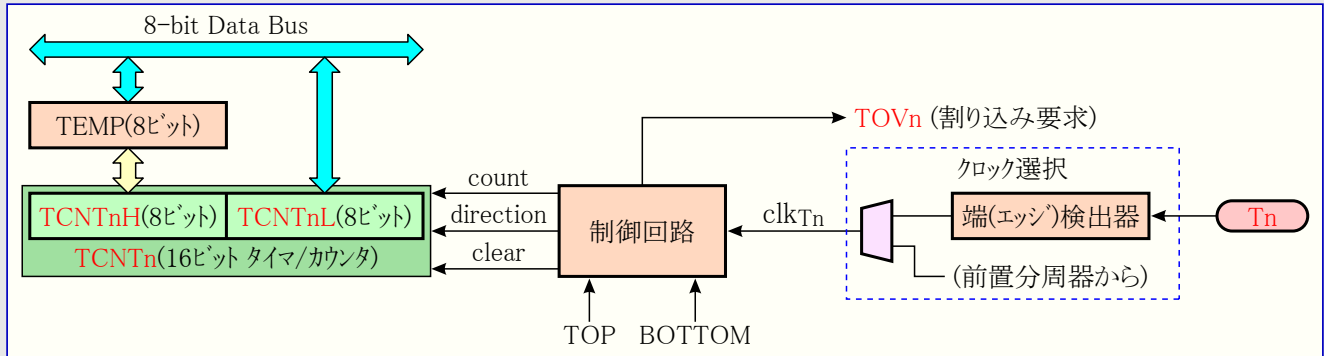
15.3. タイマ/カウンタのクロック

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とタイマ/カウンタ1の前置分周器**」をご覧ください。

15.4. 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15-2はこのカウンタとその周辺の構成図を示します。

図15-2. 計数器部構成図



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

この16ビット カウンタはカウンタの上位8ビットを含むカウンタ上位(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って、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。

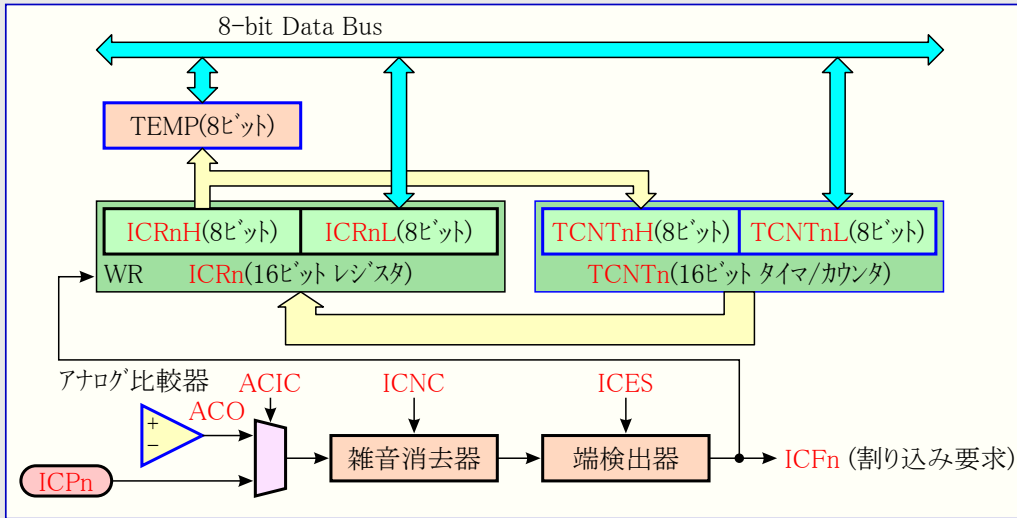
タイマ/カウンタ溢れ(TOVn)フラグはWGMn3~0ビットによって選ばれた動作種別に従って設定(1)されます。TOVnはCPU割り込み発生に使えます。

15.5. 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示す外部信号はICPnピンまたは代わりにアナログ比較器経由で印加できます。時間印はその後、周波数、デューティ比、印加された信号の他の特性の計算に使えます。代わりに時間印は出来事の記録作成にも使えます。

捕獲入力部は図15-3.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。

図15-3. 捕獲入力部構成図



注: ACO, ACICはアナログ比較器制御/状態レジスタ(ACSR)内のビットです。

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

捕獲レジスタ(ICRn)の16ビット値読み込みは、初めに下位バイト(ICRnL)、その後に上位バイト(ICRnH)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ICRnH I/O位置を読むと、この一時レジスタにアクセスします。

ICRnはカウンタのTOP値定義にICRnを利用する波形生成種別を使う時にだけ書けます。これらの場合、TOP値がICRnに書かれ得る前に波形生成種別(WGMn3~0)ビットが設定されなければなりません。ICRnに書く時は下位バイトがICRnLに書かれる前に、上位バイトがICRnH I/O位置に書かれなければなりません。

16ビットレジスタアクセス法のより多くの情報については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。

15.5.1. 捕獲起動元

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

捕獲起動入力(ICPn)ピンとアナログ比較器出力(ACO)の両入力、Tnピン(55頁の図13-1.参照)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ICRn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ICPnピンのポートを制御することによってソフトウェアで起動できます。

15.5.2. 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(TCCRB)の捕獲入力雑音消去許可(ICNC)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ICRnの更新までに4システムクロック周期の追加遅延をもたらします。雑音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。

15.5.3. 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します。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。

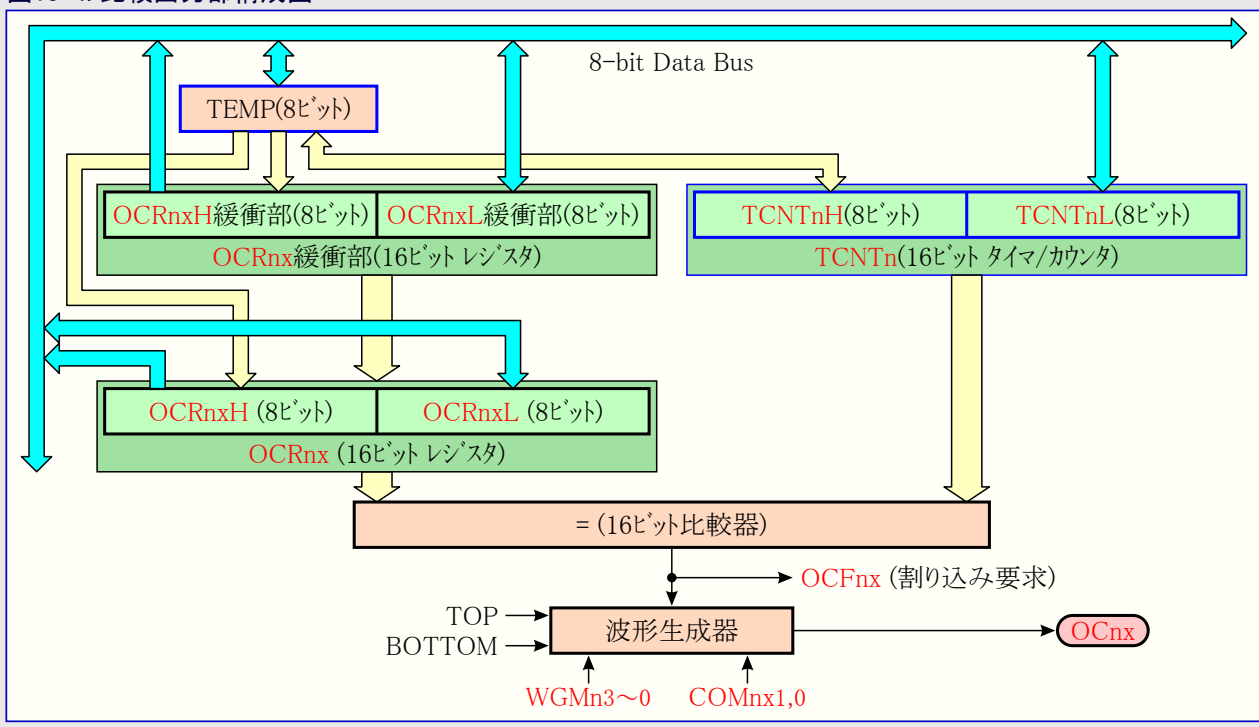
15.6. 比較出力部

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15-4.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15-4. 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。

15.6.1. 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)。

15.6.2. 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。

15.6.3. 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否かに拘らず、何れかの比較出力部を使う場合、TCNTnを変更する時に危険を伴います。TCNTnに書かれた値がOCRnx値と同じ場合、比較一致は失われ(一致が発生せず)、不正な波形生成に終わります。可変TOP値のPWM動作でTOPに等しいTCNTnを書いてはいけません。(行った場合)TOPに対する比較一致は無視され、カウンタは\$FFFFへ(計数を)続けます。同様にカウンタが下降計数のとき、BOTTOMに等しいTCNTn値を書いてはいけません。

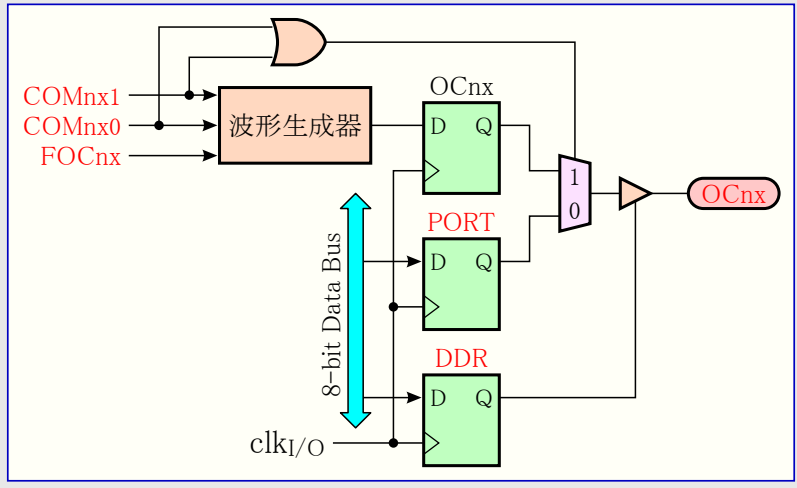
OCnxの初期設定はポートピンに対するポート方向レジスタを出力に設定する前に行われるべきです。OCnx値を設定する一番簡単な方法は標準動作で強制変更(FOCnx)ストローブビットを使うことです。波形生成種別間を変更する時であっても、OCnx(内部)レジスタはその値を保ちます。

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

15.7. 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15-5はCOMnx1,0ビット設定によって影響される論理回路の簡略化図を示します。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15-5. 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15-2、表15-3、表15-4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。

15.7.1. 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15-2を参照してください。高速PWM動作については83頁の表15-3、位相基準PWMと位相/周波数基準PWMについては83頁の表15-4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)ストップビットを使うことによって直ちに効果を得ることを強制できます。

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(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/カウンタのタイミング**」を参照してください。

15.8.1. 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させたタイマ/カウンタの分解能はソフトウェアにより増やせます。標準動作での考慮に特別な場合はなく、新しいカウンタ値は何時でも書けます。

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

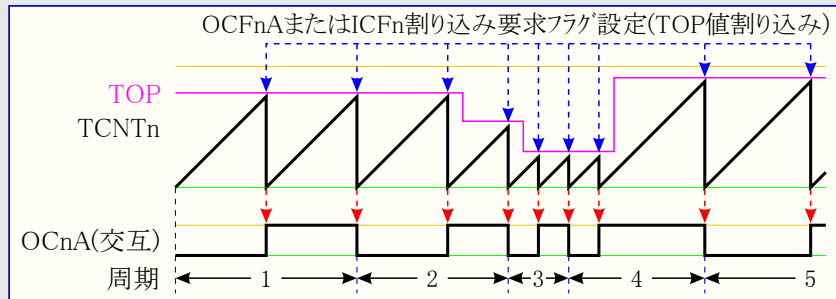
比較出力部は与えられた或る時間に割り込みを生成するのに使えます。標準動作で波形を生成するのに比較出力を使うのは、それが大変多くのCPU時間を占有するため推奨されません。

15.8.2. 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15-6.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。

図15-6. CTC動作タイミング



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。

15.8.3. 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)の何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15-7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、OCFxA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15-3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。

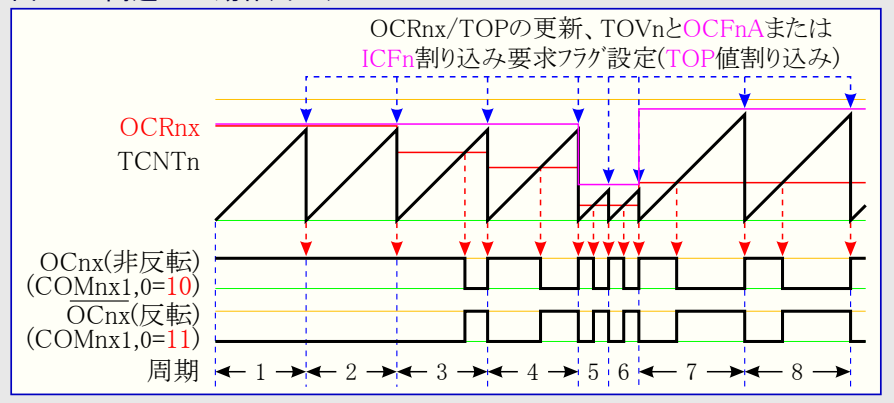
PWM出力周波数は次式によって計算できます。変数Nは前置分周数(1,8,64,256,1024)を表します。

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

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 $f_{\text{OCnA}} = f_{\text{clkI/O}}/2$ 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15-7. 高速PWM動作タイミング



15.8.4. 位相基準PWM動作

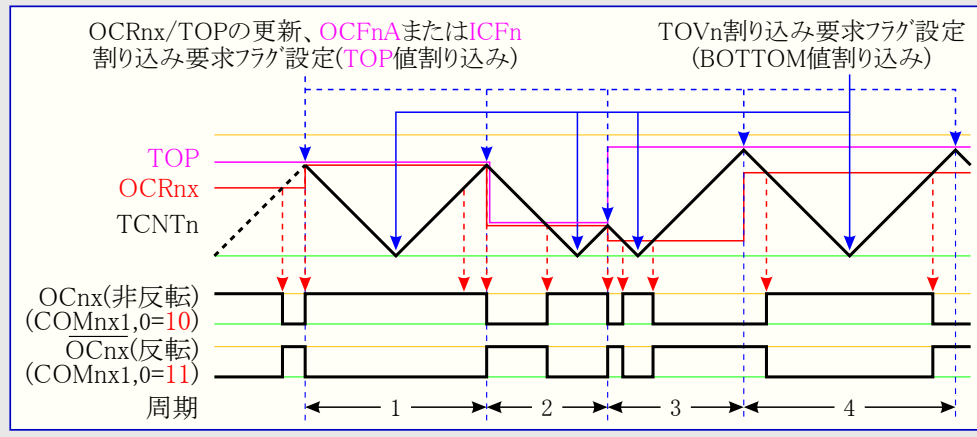
位相基準パルス幅変調(PWM)動作(WGMn3~0=0001,0010,0011,1010,1011)は高分解能で正しい位相のPWM波形生成選択を提供します。位相基準PWM動作は両傾斜(三角波)動作を基準とした位相/周波数基準PWMと似ています。カウンタはBOTTOM(\$0000)からTOPへ、そしてその後にTOPからBOTTOMへを繰り返し計数します。非反転比較出力動作(COMnx1,0=10)での比較出力(OCnx)は上昇計数中のTCNTnとOCRnx間の比較一致で解除(0)され、下降計数中の比較一致で設定(1)されます。反転出力動作(COMnx1,0=11)での動作は逆にされます。両傾斜(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15-8.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15-8. 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、OCFnxまたはICFnx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15-8.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15-4.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ます。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。

$$f_{OCnxPCPWM} = \frac{f_{clk1/O}}{2 \times N \times TOP}$$

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。

15.8.5. 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。

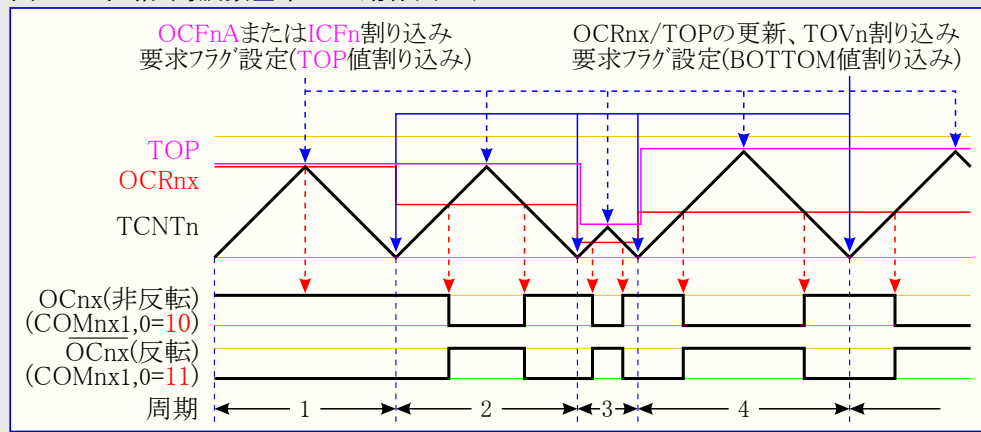
位相基準と位相/周波数基準PWM動作間の主な違いはOCRnxレジスタがOCRnx緩衝部によって更新される時(訳補:TOPとBOTTOM)です(図15-8と図15-9参照)。

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15-9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15-9. 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、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15-9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。

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15-4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。

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

OCRnxの両端値は位相/周波数基準PWM動作でPWM波形出力を生成する時の特別な場合に当たります。非反転PWM動作ではOCRnxがBOTTOMに等しく設定されると出力は定常的なLow、TOPに等しく設定されると定常的なHighになるでしょう。反転PWMに対する出力は逆の論理値になります。

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

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)の情報を含まず。図15-10は OCF_{nx} の設定についてのタイミング図を示します。

図15-10. 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

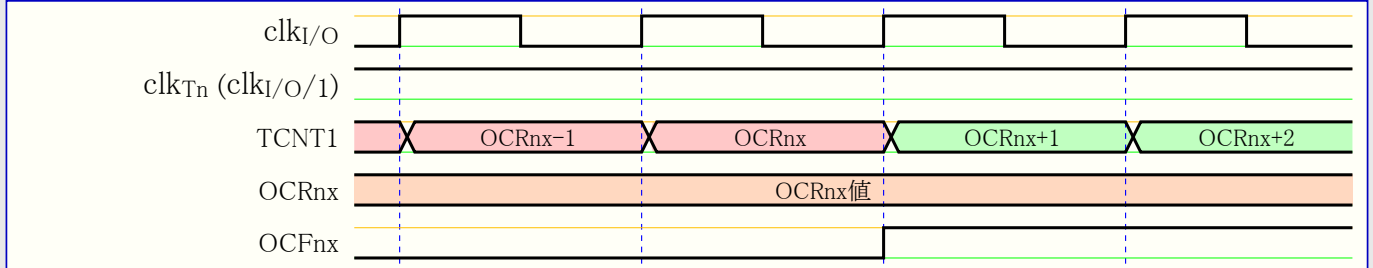


図15-11. は同じタイミング データを示しますが、前置分周器が許可されています。

図15-11. 前置分周器($f_{clk_{I/O}/8}$)のタイマ/カウンタ、 OCF_{nx} 設定 タイミング

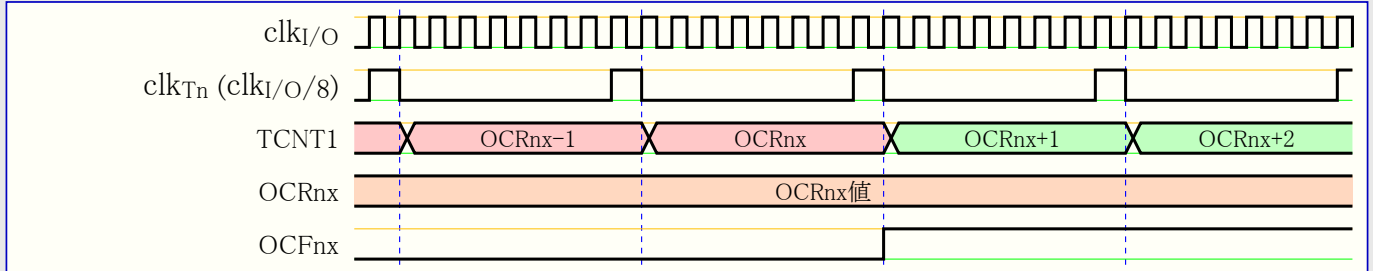


図15-12. は各動作種別でのTOP近辺の計数手順を示します。位相/周波数基準PWM動作使用時の OCR_{nx} レジスタはBOTTOMで更新されます。タイミング図は同じになりますが、当然TOPはBOTTOMで、TOP-1はBOTTOM+1でどのように置き換えられます。BOTTOMで TOV_n を設定(1)する動作種別についても、同様な名称変更が適用されます。

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

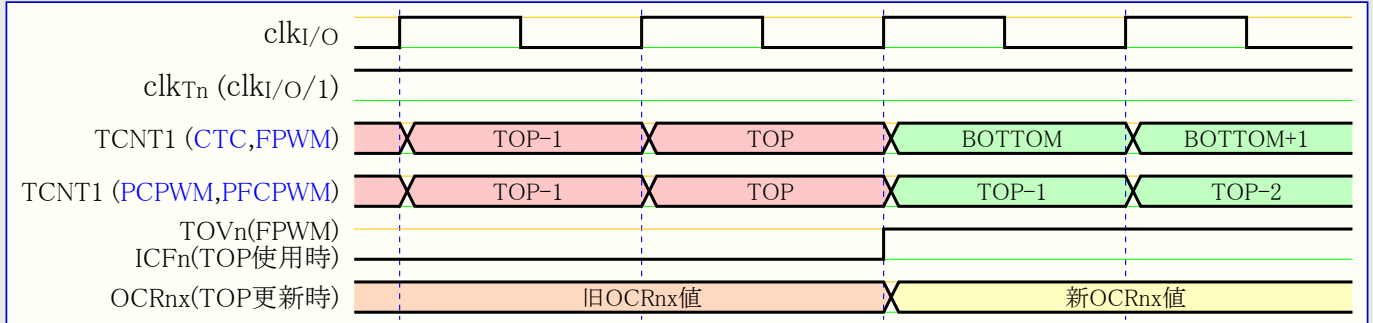
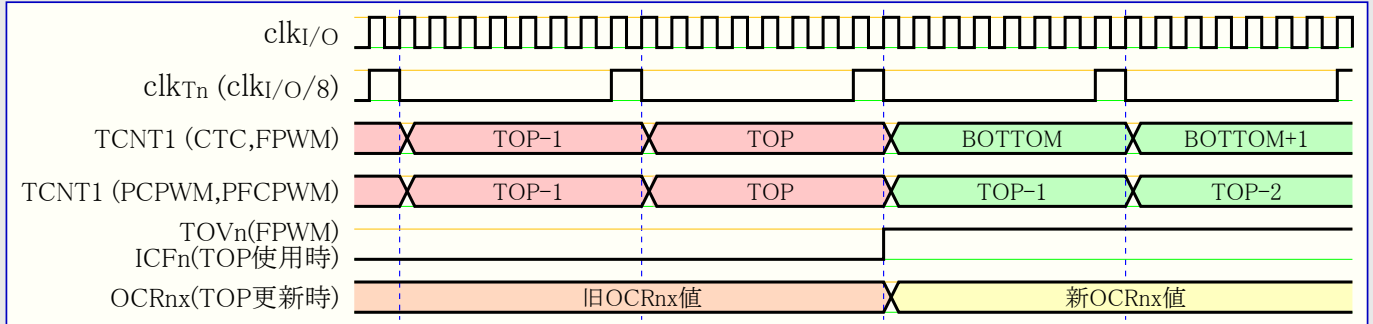


図15-13. は同じタイミング データを示しますが、前置分周器が許可されています。

図15-13. 前置分周器($f_{clk_{I/O}/8}$)のタイマ/カウンタ、TOP近辺 タイミング



15.10. 16ビット タイマ/カウンタ1用レジスタ

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

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	

- 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15-2.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15-2. 非PWM動作での比較出力選択 (注: nは1、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15-3.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15-3. 高速PWM動作での比較出力選択 (注: nは1、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15-4.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15-4. 位相基準または位相/周波数基準PWM動作での比較出力選択 (注: nは1、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(表15-5参照)。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15-5. 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。

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

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	

• ビット7 – ICNCn : 捕獲起動入力雑音消去許可 (Input Capture Noise Canceler)

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

TCCRNb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15-10と図15-11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15-6. 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)

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

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	

•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。

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

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	

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。

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

ビット (\$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	

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

ビット (\$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	

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

ビット (\$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	

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。

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

ビット (\$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	

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。

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

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	

• ビット5 – ICIE1 : タイマ/カウンタ1捕獲割り込み許可 (Timer/Counter 1 Input Capture Interrupt Enable)

このビットが1を書かれて、ステータスレジスタ(SREG)の全割り込み許可(I)ビットが設定(1)されると、タイマ/カウンタ1捕獲割り込みが許可されます。タイマ/カウンタ1割り込み要求フラグレジスタ(TIFR1)に配置された捕獲割り込み要求フラグ(ICF1)が設定(1)されると、対応する割り込みベクタ(39頁の「割り込み」参照)が実行されます。

• ビット3 – OCIE1C : タイマ/カウンタ1比較C割り込み許可 (Timer/Counter 1 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1比較C一致割り込みが許可されます。タイマ/カウンタ1割り込み要求フラグレジスタ(TIFR1)に配置された比較C割り込み要求フラグ(OCF1C)が設定(1)されると、対応する割り込みベクタ(39頁の「割り込み」参照)が実行されます。

• ビット2 – OCIE1B : タイマ/カウンタ1比較B割り込み許可 (Timer/Counter 1 Output Compare B Match Interrupt Enable)

このビットが1を書かれて、ステータスレジスタ(SREG)の全割り込み許可(I)ビットが設定(1)されると、タイマ/カウンタ1比較B一致割り込みが許可されます。タイマ/カウンタ1割り込み要求フラグレジスタ(TIFR1)に配置された比較B割り込み要求フラグ(OCF1B)が設定(1)されると、対応する割り込みベクタ(39頁の「割り込み」参照)が実行されます。

• ビット1 – OCIE1A : タイマ/カウンタ1比較A割り込み許可 (Timer/Counter 1 Output Compare A Match Interrupt Enable)

このビットが1を書かれて、ステータスレジスタ(SREG)の全割り込み許可(I)ビットが設定(1)されると、タイマ/カウンタ1比較A一致割り込みが許可されます。タイマ/カウンタ1割り込み要求フラグレジスタ(TIFR1)に配置された比較A割り込み要求フラグ(OCF1A)が設定(1)されると、対応する割り込みベクタ(39頁の「割り込み」参照)が実行されます。

• ビット0 – TOIE1 : タイマ/カウンタ1溢れ割り込み許可 (Timer/Counter 1 Overflow Interrupt Enable)

このビットが1を書かれて、ステータスレジスタ(SREG)の全割り込み許可(I)ビットが設定(1)されると、タイマ/カウンタ1溢れ割り込みが許可されます。タイマ/カウンタ1割り込み要求フラグレジスタ(TIFR1)に配置されたタイマ/カウンタ1溢れ割り込み要求フラグ(TOV1)が設定(1)されると、対応する割り込みベクタ(39頁の「割り込み」参照)が実行されます。

15.10.10. 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) TIFR1

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	

• ビット5 – ICFn : 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)

ICPn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(ICRn)がWGMn3~0によってTOP値として設定されると、ICFn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、ICFn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ICFnは解除(0)できます。

• ビット3 – OCFnC : 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(TCNTn)値が比較Cレジスタ(OCRnC)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(FOCnC)ストローブがOCFnC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、OCFnC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OCFnCは解除(0)できます。

• ビット2 – OCFnB : 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(TCNTn)値が比較Bレジスタ(OCRnB)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(FOCnB)ストローブがOCFnB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、OCFnB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OCFnBは解除(0)できます。

• ビット1 – OCFnA : 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(TCNTn)値が比較Aレジスタ(OCRnA)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(FOCnA)ストローブがOCFnA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、OCFnA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OCFnAは解除(0)できます。

• ビット0 – TOVn : 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(1)設定はWGMn3~0ビット設定に依存します。標準またはCTC動作でのTOVnフラグはタイマ/カウンタ1溢れ時に設定(1)されます。他のWGMn3~0ビット設定を使う時のTOVnフラグ動作については84頁の表15-5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、TOVn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TOVnは解除(0)できます。

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「コード例について」をご覧ください。

次のコード例は従装置としてSPIを初期化する方法と簡単な受信を実行する方法を示します。

アセンブリ言語プログラム例

```
SPI_S_Init: LDI    R17, (1<<DD_MISO)                  ;MISO出力、他は入力値を取得
            OUT    DDR_SPI, R17                      ;MISO出力、他は入力に設定
            LDI    R17, (1<<SPE)                    ;SPI許可値を取得
            OUT    SPCR, R17                        ;SPI許可設定
            RET                                         ;呼び出し元へ復帰

SPI_S_Rx:   SBIS    SPSR, SPIF                      ;受信(転送)完了ならばスキップ
            RJMP   SPI_S_Rx                        ;受信(転送)完了まで待機
;
            IN     R16, SPDR                        ;受信データを取得
            RET                                         ;呼び出し元へ復帰
```

C言語プログラム例

```
void SPI_SlaveInit(void)
{
    DDR_SPI = (1<<DD_MISO);                          /* MISO出力、他は入力に設定 */
    SPCR = (1<<SPE);                                  /* SPI許可設定 */
}

char SPI_SlaveReceive(void)
{
    while(!(SPSR & (1<<SPIF)));                        /* 受信(転送)完了まで待機 */
    return SPDR;                                       /* 受信データと共に復帰 */
}
```

注: 5頁の「コード例について」をご覧ください。

16.1. SSピンの機能

16.1.1. 従装置動作

SPIが従装置として設定されると、従装置選択(SS)ピンは常に入力です。SSがLowに保たれるとSPIは活性に(作動)され、使用者によってそのように設定されていればMISOは出力になります。他の全てのピンは入力です。SSがHighに駆動されると、出力として使用者設定され得るMISOを除く全てのピンは入力、SPIは非活動で、それは到着データを受信しないことを意味します。一旦SSピンがHighに駆動されると、SPI論理回路がリセットすることに注意してください。

このSSピンはパケット/バイト同期に対して、従装置ビット計数器が主装置クロック発生器との同期を保つのに有用です。SSピンがHighに駆動されると、SPI従装置は直ちに送受信論理回路をリセットし、それは移動レジスタ内で部分的に受信したデータのデータも取り落とします。

16.1.2. 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、SSピンの方向は使用者が決められます。

SSが出力として設定されると、このピンはSPIシステムに影響を及ぼされない標準出力ピンです。代表的にはこのピンがSPI従装置のSSピンを駆動するでしょう。

SSが入力として設定されると、SPI主装置動作を保証するため、それはHighに保持されなければなりません。SSピンが入力として定義されたSPI主装置として設定される時に周辺回路によってSSピンがLowに駆動されると、SPIシステムは他の主装置が従装置として選んでデータ送信を始めると解釈します。バスの衝突を避けるためにSPIシステムは次の動作を行います。

1. SPCRで主/従選択(MSTR)ビットが解除(0)され、SPIシステムは従装置になります。SPIシステムが従装置になる結果としてMOSIとSCKピンが入力になります。
2. SPI状態レジスタ(SPSR)でSPI割り込み要求フラグ(SPIF)が設定(1)され、そしてSPI割り込みが許可(SPCRのSPIE=1)され、且つステータスレジスタ(SREG)の全割り込み許可(I)ビットが設定(1)なら、割り込みルーチンが実行されます。

従って割り込み駆動SPI送信が主装置動作で使われ、SSがLowに駆動される可能性があるとき、その割り込み(処理)はMSTRビットが未だ設定(1)されていることを常に検査すべきです。MSTRビットが従装置選択によって解除(0)されてしまっていると、それはSPI主装置動作を再び許可するため、使用者によって設定(1)されなければなりません。

16.1.3. SPI制御レジスタ (SPI Control Register) SPCR

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。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16-3.と図16-4.を参照してください。CPOL機能は右で要約されます。

表16-2. 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16-3.と図16-4.を参照してください。CPHA機能は右で要約されます。

表16-3. 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 f_{OSC} 間の関連は次表で示されます。

表16-4. SCK速度選択 (f_{OSC} =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	$f_{OSC}/2$	$f_{OSC}/4$	$f_{OSC}/8$	$f_{OSC}/16$	$f_{OSC}/32$	$f_{OSC}/64$	$f_{OSC}/128$	

16.1.4. SPI状態レジスタ (SPI Status Register) SPSR

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(1)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、その後にSPIデータレジスタ(SPCR)をアクセスすることによってもSPIFフラグは解除(0)されます。

• **ビット6 – WCOL : 上書き発生フラグ** (Write Collision Flag)

データ転送中にSPIデータレジスタ(SPCR)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(SPCR)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16-4.参照)。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 f_{OSC} (CPUクロック周波数)/4またはそれ以下の動作のみ保証されます。

AT90USB82/162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169頁をご覧ください。

16.1.5. SPIデータレジスタ (SPI Data Register) SPDR

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。

16.2. 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16-3と図16-4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16-2と表16-3を要約することによって明解にされます。

表16-5. 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16-3. SPIデータ転送形式 (CPHA=0)

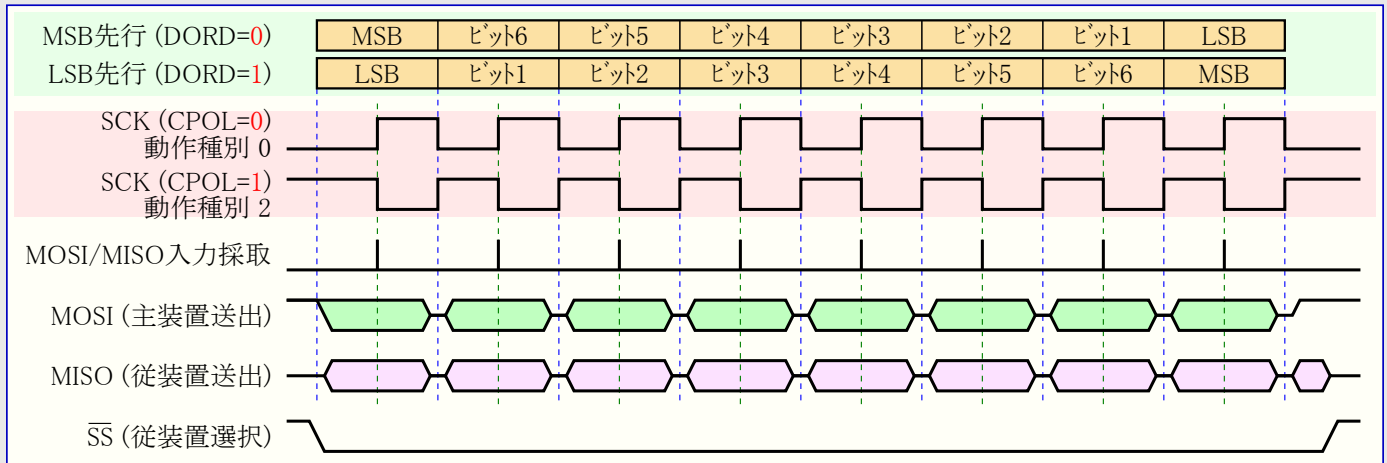
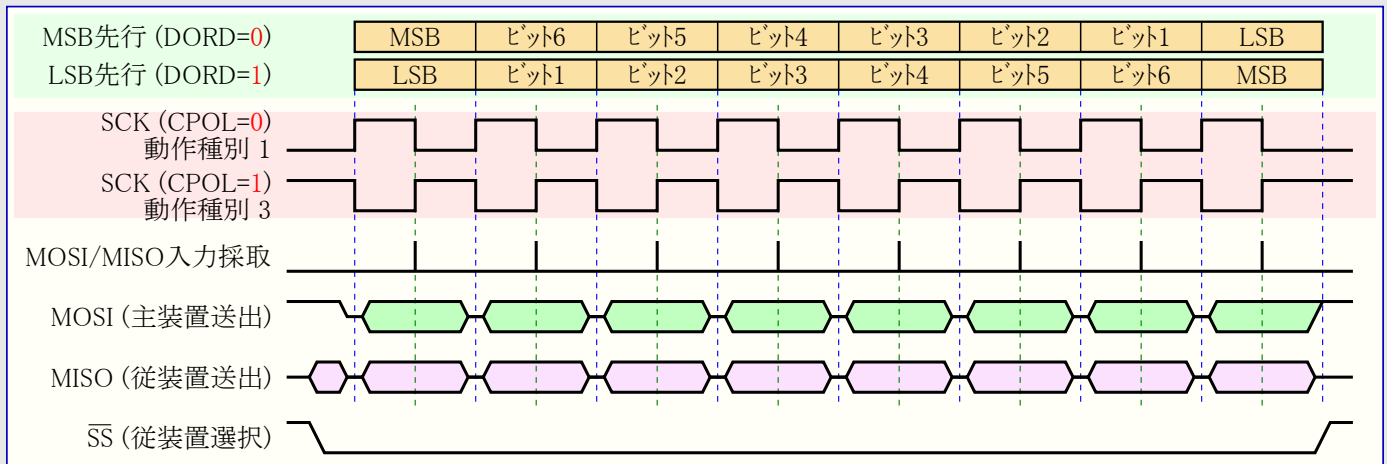


図16-4. SPIデータ転送形式 (CPHA=1)



17. 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。

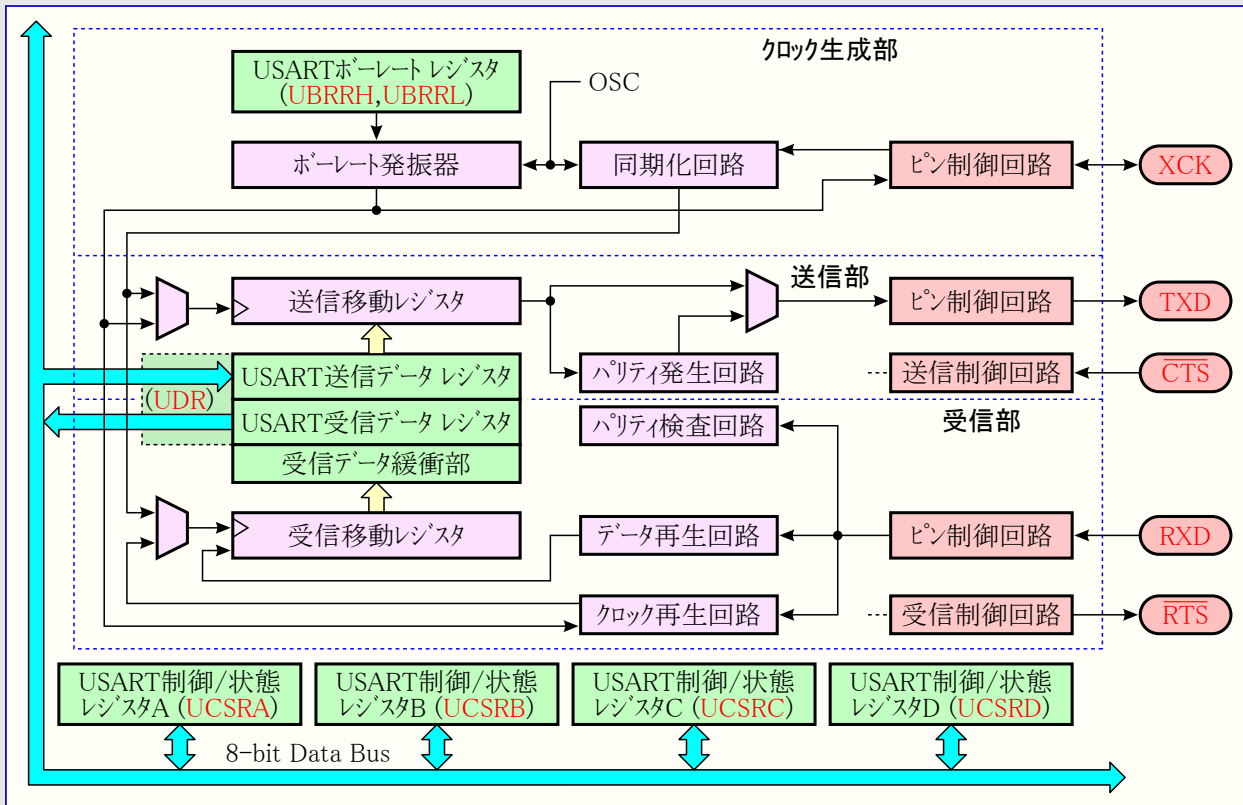
- 全二重動作 (独立した送受信レジスタ)
- 同期または非同期動作
- CTS/RTS信号ハードウェア管理流れ制御
- 同期クロック駆動された主装置/従装置動作
- 高分解能ボーレート発振器
- 5, 6, 7, 8または9ビットデータと1または2停止ビットの直列フレームの支援
- ハードウェアによって支援された奇数または偶数パリティの生成と検査
- データオーバーラン検出
- フレーミング異常検出
- 不正開始ビット検出とデジタル低域通過濾波器を含む雑音濾波器
- 受信完了、送信完了、送信データレジスタ空きの3つの分離した割り込み
- 複数プロセッサ通信機能
- 倍速非同期通信動作

17.1. 概要

USARTの簡便化した構成図は図17-1.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。112頁の「USARTでのSPI動作」をご覧ください。28頁の「電力削減レジスタ(PRR1)」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17-1. USART構成図



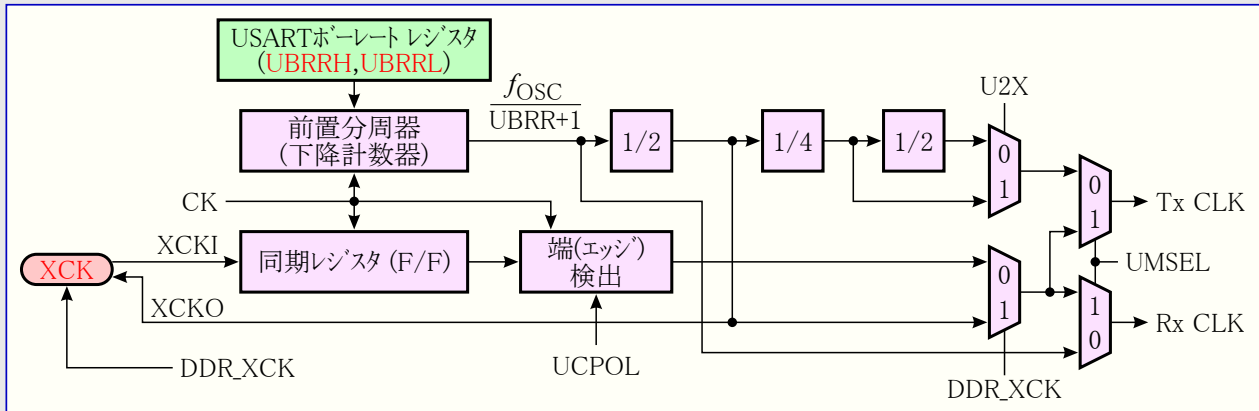
注: USARTピン配置については2頁の「ピン配置」、52頁の表12-9をご覧ください。

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。

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17-2.はクロック生成論理回路の構成図を示します。

図17-2. クロック生成部構成図



信号説明	Tx CLK	: 送信クロック (内部信号)
	Rx CLK	: 受信クロック (内部信号)
	XCKI	: XCKピンからの内部入力信号 (同期従装置時)
	XCKO	: XCKピンへの内部出力信号 (同期主装置時)
	DDR_XCK	: 対応するポート方向レジスタの方向ビット
	f _{OSC}	: システム クロック周波数

17.2.1. ホ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17-2を参照してください。

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

表17-1.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17-1. ボーレートレジスタ(UBRRn)値計算式

動作種別	波特率計算式	UBRRn値計算式
標準速非同期動作 (U2Xn=0)	$\text{BAUD} = \frac{f_{\text{OSC}}}{16 \times (\text{UBRRn} + 1)}$	$\text{UBRRn} = \frac{f_{\text{OSC}}}{16 \times \text{BAUD}} - 1$
倍速非同期動作 (U2Xn=1)	$\text{BAUD} = \frac{f_{\text{OSC}}}{8 \times (\text{UBRRn} + 1)}$	$\text{UBRRn} = \frac{f_{\text{OSC}}}{8 \times \text{BAUD}} - 1$
同期主装置動作	$\text{BAUD} = \frac{f_{\text{OSC}}}{2 \times (\text{UBRRn} + 1)}$	$\text{UBRRn} = \frac{f_{\text{OSC}}}{2 \times \text{BAUD}} - 1$

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110～111頁の表17-9.で得られます。

17.2.2. 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に減らします。けれども受信部がデータ採取とクロック再生に対して半分の(16から8に減じた)採取数をこの場合だけ使い、従ってこの動作種別が使われる時はシステムクロックとボーレート設定の精度がより必要とされることに注意してください。送信部についての低下要因はありません。

17.2.3. 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17-2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。

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

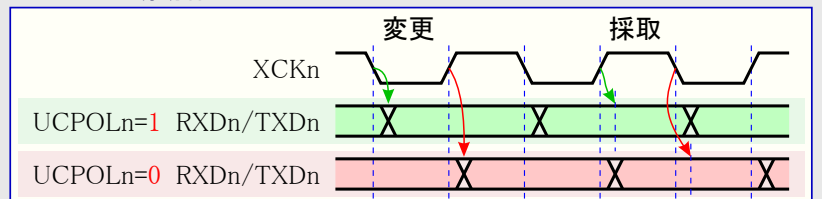
f_{OSC} がシステムクロック元の安定度に依存することに注意してください。従って周波数変動によるデータ消失の可能性を避けるため、いくらかの余裕分を追加することが推奨されます。

17.2.4. 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17-3.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17-3. 同期動作XCKnタイミング



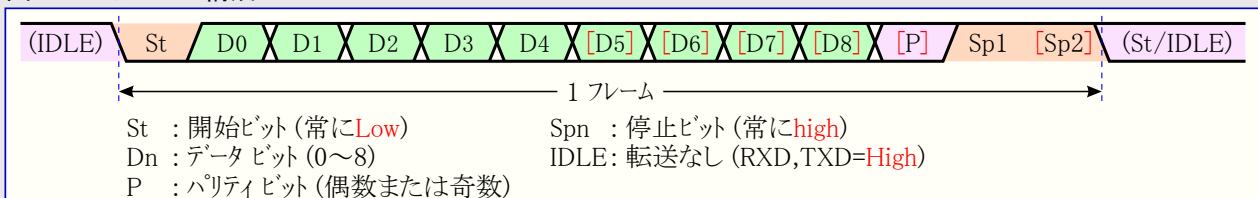
17.3. フレーム形式

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17-4.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17-4. 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。

17.3.1. 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 = $D_0 \text{ Ex-OR } D_1 \text{ Ex-OR } D_2 \text{ Ex-OR } \sim \text{Ex-OR } D_{n-1}$
 奇数パリティビット = $D_0 \text{ Ex-OR } D_1 \text{ Ex-OR } D_2 \text{ Ex-OR } \sim \text{Ex-OR } D_{n-1} \text{ Ex-OR } 1$ 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。

17.4. 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\(0\)ビット](#)は解除(0)される(そして全割り込みが禁止される)べきです。

ボーレートまたはフレーム形式の変更を伴う再初期化を行う前には、レジスタが変更される期間中に進行中の送信がないことを確実にしてください。[USART制御/状態レジスタ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「コード例について」をご覧ください。

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

17.5. 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。

17.5.1. 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時はUDRnに書かれた上位ビットが無視されます。

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「コード例について」をご覧ください。

この関数は送信されるべき新規データを設定する前に、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込みます。

17.5.2. 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「コード例について」をご覧ください。

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

17.5.3. 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る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効です。

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。

17.5.4. 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。

17.5.5. 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。

17.6. 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。

17.6.1. 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「コード例について」をご覧ください。

この関数は緩衝部を読んで値を戻す前に、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。

17.6.2. 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)ビットから読まれなければなりません。この規則はフレーミング異常(FEn)、オーバーラン発生(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;          /* 一時変数定義 */
    while ( !(UCSRnA & (1<<RxCn)) );          /* 受信完了待機 */
    status = UCSRnA;                          /* 状態フラグ取得 */
    resh = UCSRnB;                            /* 受信第9ビット取得 */
    resl = UDRn;                              /* 受信データ取得 */
    if ( status & ((1<<FEn) | (1<<DORn) | (1<<UPEn)) ) return -1; /* 受信異常で-1値設定/復帰 */
    resh = (resh>>1) & 0x01;                  /* RXB8nビットのみ有効最下位へ */
    return ((resh<<8) | resl);                 /* 結果9ビット データ取得/復帰 */
}
```

注: 5頁の「コード例について」をご覧ください。

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

17.6.3. 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。

17.6.4. 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97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。

17.6.5. 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。

17.6.6. 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。

17.6.7. 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「コード例について」をご覧ください。

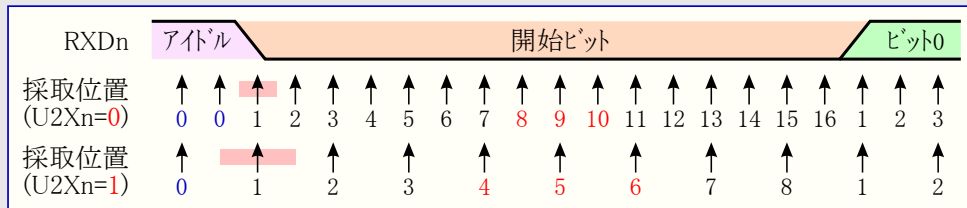
17.7. 非同期受信

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ために使われます。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。

17.7.1. 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17-5は到着フレームの開始ビットの採取手順を図解します。採取速度は標準速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17-5. 開始ビットの採取

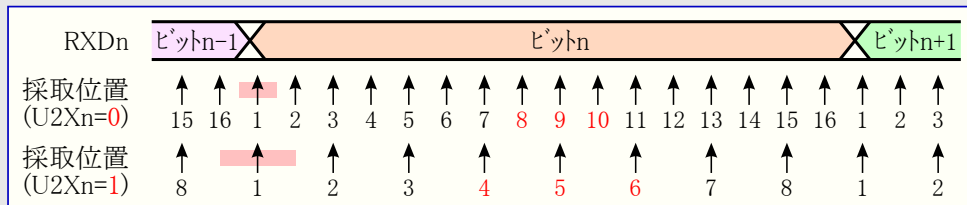


クロック再生論理回路がRXDn信号線でHigh(アイドル)からLow(開始)の遷移を検出すると、開始ビット検出手順が開始されます。図で示されるように採取1は最初のLow採取を意味します。その後にクロック再生回路は有効な開始ビットが受信されるかを決定するために、標準速動作に対して採取8,9,10、倍速動作に対して採取4,5,6(図の赤字(訳注:原文は箱枠内))で示された採取番号)を使います。これら3回で2回以上の採取が論理Highレベルの場合(多数決)、この開始ビットは尖頭雑音として捨てられ、受信部は次のHighからLowの遷移を探し始めます。けれども有効な開始ビットが検出されると、クロック再生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

17.7.2. 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速動作で16段、倍速動作で8段の順列回路を使います。図17-6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。

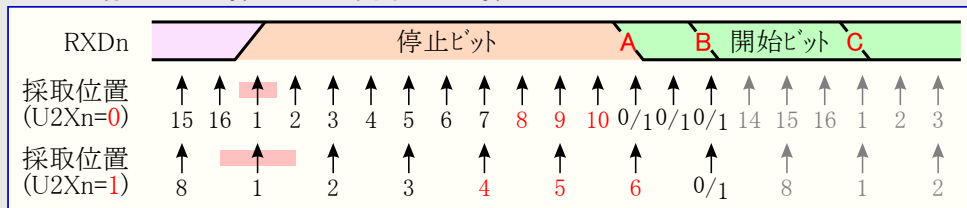
図17-6. データビットとパリティビットの採取



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17-7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17-7. 停止ビットの採取と次の開始ビットの採取



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

新規フレームの開始ビットを示すHighからLowへの遷移は多数決に使った最後のビット後に実現できます。標準速動作での最初のLowレベル採取は図17-7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。

17.7.3. 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17-2.参照)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。

表17-2.は許容できる最大受信部ボーレート誤差一覧です。標準速動作には、より高いボーレート変動許容力があることに注目してください。

表17-2. 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17-2.に標準速、表17-3.に倍速を記載していますが、比較が容易なように表17-2.として纏めました。

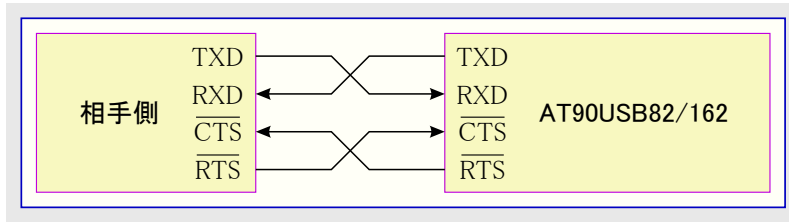
受信部ボーレートの推奨最大許容誤差は最大総合許容誤差を送信部と受信部で等分割するという仮定の元で作られました。

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。

17.8. 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。

CTS : Clear to Send (相手側からの送信要求/許可信号入力)
RTS : Request to Send (相手側への送信要求/許可信号出力)



17.8.1. 受信部流れ制御

受信の流れは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。RTS使用と関連する流れ制御はUSART制御/状態レジスタD(UCSRD)のRTS許可(RTSEN)ビットを使うことで許可されます。

図17-8.は受信例を示します。

図17-8. 受信流れ制御波形例

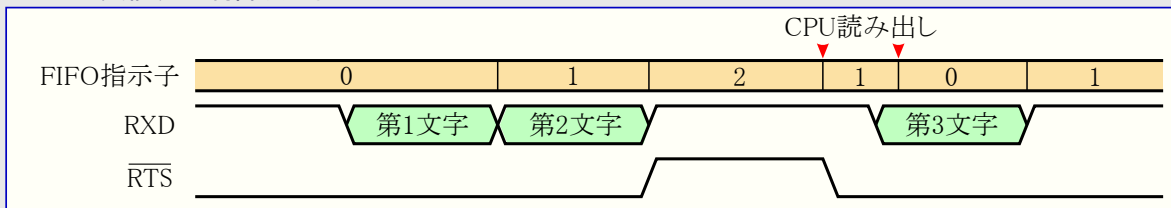
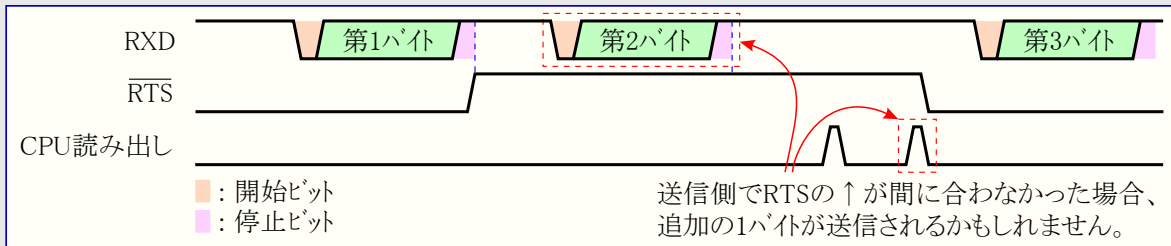


図17-9. RTSの動き



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。

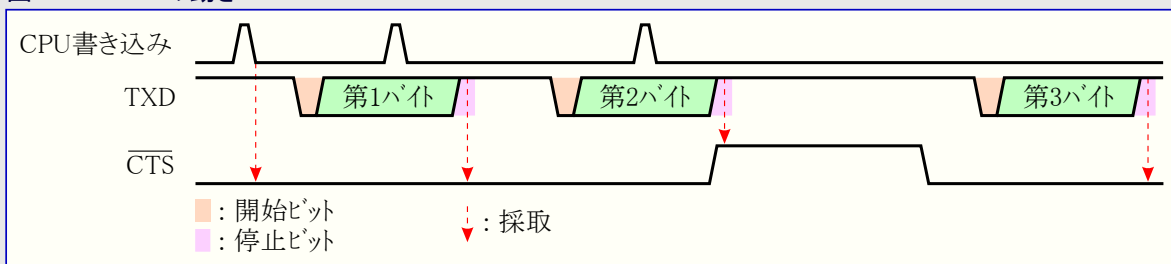
確かな送信を保証するため、RTS上昇後も追加データを未だ受信でき、受信移動レジスタ内に格納されます。

17.8.2. 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(CTS=1)の時に(こちらからの)送信を停止することです。CTS使用と関連する流れ制御はUSART制御/状態レジスタD(UCSRD)のCTS許可(CTSEN)ビットを使うことで許可されます。

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17-10. CTSの動き



17.9. 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。

17.9.1. 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(リード-モディファイ-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) AT90USB82/162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。

17.10. USART用レジスタ

17.10.1. USARTデータレジスタ (USART I/O Data Register) UDR1

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	UDR1
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。)

17.10.2. USART制御/状態レジスタA (USART Control and Status Register A) UCSR1A

ビット (\$C8)	7	6	5	4	3	2	1	0	
	RXCn	TXCn	UDREN	FEn	DORn	UPEn	U2Xn	MPCMn	UCSR1A
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るかどうかを示します。UDRENが1ならば緩衝部は空で、従って書かれる準備ができています。UDRENフラグは送信緩衝部空き割り込みを発生できます(UCSRnBの**送信データレジスタ空き割り込み許可(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。同期動作を使う時はこのビットに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105頁の「**複数プロセッサ通信動作**」をご覧ください。

17.10.3. USART制御/状態レジスタB (USART Control and Status Register B) UCSR1B

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	UCSR1B
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(UCSR_{nA})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、UCSR_{nA}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(UCSR_{nA})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、UCSR_{nA}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(UCSR_{nA})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、UCSR_{nA}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(UPE_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(UCSR_{nC})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。

17.10.4. USART制御/状態レジスタC (USART Control and Status Register C) UCSR1C

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	UCSR1C
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)

このビットは表17-4.で示されるようにUSART動作種別を選びます。

表17-4. 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112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17-5. 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(誤補:常に第1停止ビットだけが有効)。

表17-6. 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17-7. 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17-8. 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

17.10.5. USART制御/状態レジスタD (USART Control and Status Register D) UCSR1D

ビット (\$CB)	7	6	5	4	3	2	1	0	
	–	–	–	–	–	–	CTSENn	RTSENn	UCSR1D
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1 – CTSENn : CTS許可 (USART CTS Enable)

送信流れ(フロー)制御(CTS)を許可するにはファームウェアによってこのビットを1に設定してください。送信はCTS=0の場合に許されます。送信流れ(フロー)制御(CTS)を禁止するにはファームウェアによってこのビットを0に設定してください。送信は常に許されます。

• ビット0 – RTSENn : RTS許可 (USART RTS Enable)

受信流れ(フロー)制御(RTS)を許可するにはファームウェアによってこのビットを1に設定してください。受信流れ(フロー)制御(RTS)を禁止するにはファームウェアによってこのビットを0に設定してください。

17.10.6. USARTボーレートレジスタ (USART Baud Rate Register) UBRR1H, UBRR1L (UBRR1)

ビット (\$CD)	15	14	13	12	11	10	9	8	
	–	–	–	–	UBRR11	UBRR10	UBRR9	UBRR8	UBRR1H
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	
	UBRR7	UBRR6	UBRR5	UBRR4	UBRR3	UBRR2	UBRR1	UBRR0	UBRR1L
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。

17.11. ホーレート設定例

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ホーレートは表17-9のUBRRn設定を使うことによって生成できます。目的のホ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(103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。

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

表17-9. Xtal、ホーレート対UBRRnH,UBRRnL設定 (UBRR=UBRRnH:UBRRnL)

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

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

注: 最高速はUBRR=0、誤差=0.0%です。以降の周波数は次頁へ続く。

(訳注) 原書では頁割された表毎に表番号9~12となっていますが、共通性から纏めて表17-9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17-9 (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			
	U2X=0		U2X=1		U2X=0		U2X=1		U2X=0		U2X=1		U2X=0		U2X=1	
	UBRR	誤差(%)	UBRR	誤差(%)	UBRR	誤差(%)	UBRR	誤差(%)	UBRR	誤差(%)	UBRR	誤差(%)	UBRR	誤差(%)	UBRR	誤差(%)
1200	255	0.0	511	0.0	319	0.0	639	0.0	383	0.0	767	0.0	416	-0.1	832	0.0
2400	127	0.0	255	0.0	159	0.0	319	0.0	191	0.0	383	0.0	207	0.2	416	-0.1
4800	63	0.0	127	0.0	79	0.0	159	0.0	95	0.0	191	0.0	103	0.2	207	0.2
9600	31	0.0	63	0.0	39	0.0	79	0.0	47	0.0	95	0.0	51	0.2	103	0.2
14400	20	1.6	42	-0.8	26	-1.2	52	0.6	31	0.0	63	0.0	34	-0.8	68	0.6
19200	15	0.0	31	0.0	19	0.0	39	0.0	23	0.0	47	0.0	25	0.2	51	0.2
28800	10	-3.0	20	1.6	12	2.6	26	-1.2	15	0.0	31	0.0	16	2.1	34	-0.8
38400	7	0.0	15	0.0	9	0.0	19	0.0	11	0.0	23	0.0	12	0.2	25	0.2
57600	4	6.7	10	-3.0	6	-4.8	12	2.6	7	0.0	15	0.0	8	-3.5	16	2.1
76800	3	0.0	7	0.0	4	0.0	9	0.0	5	0.0	11	0.0	6	-7.0	12	0.2
115.2k	2	-11.1	4	6.7	2	11.1	6	-4.8	3	0.0	7	0.0	3	8.5	8	-3.5
230.4k	1	33.3	2	-11.1	1	-16.7	2	11.1	1	0.0	3	0.0	1	8.5	3	8.5
250k	1	22.9	2	-18.1	1	-23.2	2	2.4	1	-7.8	3	0.0	1	0.0	3	0.0
500k	0	-38.6	0	22.9	0	-23.2	1	-23.2	0	-7.8	1	-7.8	0	0.0	1	0.0
1M	-	-	0	-38.6	-	-	0	-23.2	-	-	0	-7.8	-	-	0	0.0
最高速	307.2k		614.4k		384k		768k		460.8k		921.6k		500k		1M	

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

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

注: 最高速はUBRR=0、誤差=0.0%です。

18. 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(MS PIM)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($f_{XCKmax}=f_{CK}/2$)
- 柔軟な割り込み生成

18.1. 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。

18.2. 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設定は同じ式を使って計算できます。表18-1をご覧ください。

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
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

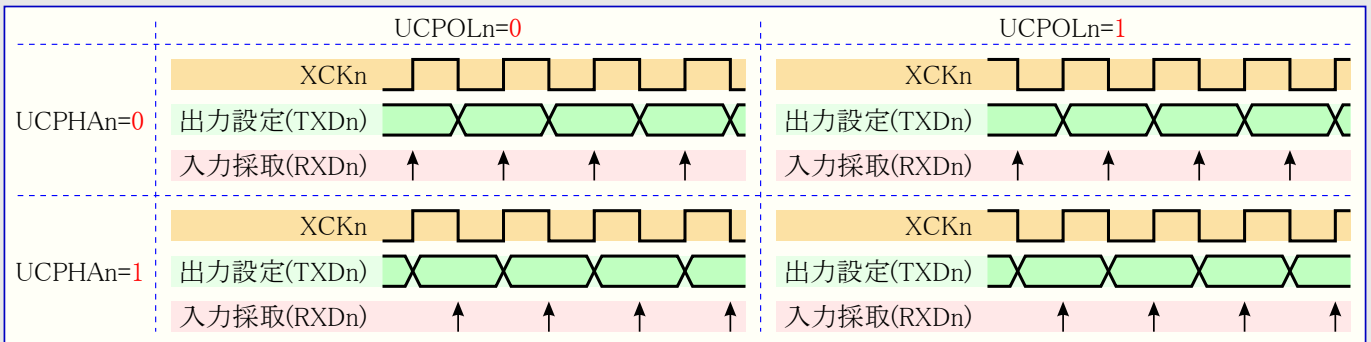
18.3. データ転送形式

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18-1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の機能は表18-2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18-2. 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18-1. UCPHAnとUCPOLnによるデータ転送タイミング



18.4. 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。

18.4.1. 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(DDRXCKnの1)設定、フレーム形式設定、送受信部許可から成ります。送信部だけが独立して操作できます。割り込み駆動USART操作に関して初期化を行う時にステータスレジスタの全割り込み許可(I)ビットが解除(0)される(そして全割り込みが禁止される)べきです。

注: XCKn出力の初期化を直ちに保証するため、ボーレートレジスタ(UBRRn)は送信部が許可される時に0でなければなりません。標準動作のUSART操作と逆に、UBRRnは送信部が許可されて最初の送信が開始される前に、希望する値を書かれなければなりません。この初期化がリセット後直ちに行なわれるなら、UBRRnが0にリセットされているので、送信部許可前にUBRRnを0に設定することは必要ありません。

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

次の簡単なUSART初期化コード例が示すアセンブリ言語とC言語の関数は機能的に同じです。この例はボーリングを使う(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);
    ;動作種別値を取得
    UCSRnB = (1<<RXENn) | (1<<TXENn);
    ;送受信許可
    UBRRn = baud;             /* ボーレート設定 */
}

```

注: 5頁の「コード例について」をご覧ください。

18.5. 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    UCSRA, 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_SPI:  SBIS    UCSRA, RXCn      ;受信完了でスキップ
             RJMP    USART_SPI      ;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 ( !(UCSRA & (1<<UDREN)) ); /* 送信緩衝部空き待機 */
    UDRn = data;                      /* データ送信(送信開始) */
    while ( !(UCSRA & (1<<RXCn)) ); /* 受信完了待機 */
    return UDRn;                      /* 受信データ取得 */
}
```

注: 5頁の「コード例について」をご覧ください。

18.5.1. 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。

18.5.2. 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。

18.6. 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。

18.6.1. USART MSPIMデータレジスタ (USART I/O Data Register) UDR1

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。[106頁の「USARTデータレジスタ\(UDR1\)」](#)をご覧ください。

18.6.2. USART MSPIM制御/状態レジスタA (USART Control and Status Register A) UCSR1A

ビット (\$C8)	7	6	5	4	3	2	1	0	
	RXCn	TXCn	UDREn	–	–	–	–	–	UCSR1A
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時、自動的に解除(0)されるか、またはこのビット位置に1を書くことによって解除(0)できます。TXCnフラグは送信完了割り込みを発生できます(UCSRnBの送信完了割り込み許可(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。

18.6.3. USART MSPIM制御/状態レジスタB (USART Control nad Status Register B) UCSR1B

ビット (\$C9)	7	6	5	4	3	2	1	0	
	RXCIEn	TXCIEn	UDRIEn	RXENn	TXENn	–	–	–	UCSR1B
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。

18.6.4. USART MSPIM制御/状態レジスタC (USART Control nad Status Register C) UCSR1C

ビット (\$CA)	7	6	5	4	3	2	1	0	
	UMSELn1	UMSELn0	–	–	–	UDORDn	UCPHAn	UCPOLn	UCSR1C
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18-3.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、107頁の「USART制御/状態レジスタC(UCSR1C)」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18-3. 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113頁の「フレーム形式」節を参照してください。

• **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112頁の「データ転送形式」節を参照してください。

• **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112頁の「データ転送形式」節を参照してください。

18.6.5. USART MSPIMボーレートレジスタ (USART Baud Rate Register) UBRR1H, UBRR1L (UBRR1)

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。109頁の「USARTボーレートレジスタ(UBRR1H,UBRR1L)」をご覧ください。

18.7. USARTでのMSPIMとSPIの比較

USARTでのMSPI動作は次に関してSPIと完全な互換性があります。

- 主装置動作タイミング図
- クロック極性選択(UCPOLn)ビットはSPIのSCK極性選択(CPOL)ビットと機能的に同じです。
- クロック位相選択(UCPHAn)ビットはSPIのSCK位相選択(CPHA)ビットと機能的に同じです。
- データ順選択(UDORDn)ビットはSPIのデータ順選択(DORD)ビットと機能的に同じです。

けれどもUSARTでのMSPI動作がUSART資源を再使用するため、USARTでのMSPI動作はSPIと比較して多少異なります。加えて制御レジスタビットの差異、主装置動作だけがUSARTでのMSPI動作によって支援されること、2つの部間で異なる次の特質があります。

- USARTでのMSPI動作は送信部の(2重)緩衝部を含みます。SPIは緩衝部を持ちません。
- USARTのMSPI動作での受信部は追加の緩衝段を含みます。
- SPIの上書き(WCOL)ビットはUSARTでのMSPI動作に含まれません。
- SPIの倍速許可(SPI2X)ビットは含まれません。しかし、対応するボーレートレジスタ(UBRRn)設定によって同じ効果が達せられます。
- 割り込みタイミングに互換性はありません。
- USARTでのMSPI動作が主装置動作だけのため、ピン制御が異なります。

USARTのMSPI動作とSPIでのピンは表18-4.で示されます。

表18-4. 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	$\overline{SS}$	USARTでのMSPIMで未支援

19. USB制御器

19.1. 特徴

- 全速(Full-speed)を支援
- 明解な切替での交互动作(2バンク)支援
- 176バイトの2ポートRAM(DPRAM)
 - 最大64バイトの1つのエンドポイント(既定制御エンドポイント)
 - 最大64バイトの2つのエンドポイント(1バンク)
 - 最大64バイトの2つのエンドポイント(1または2バンク)

19.2. 構成図

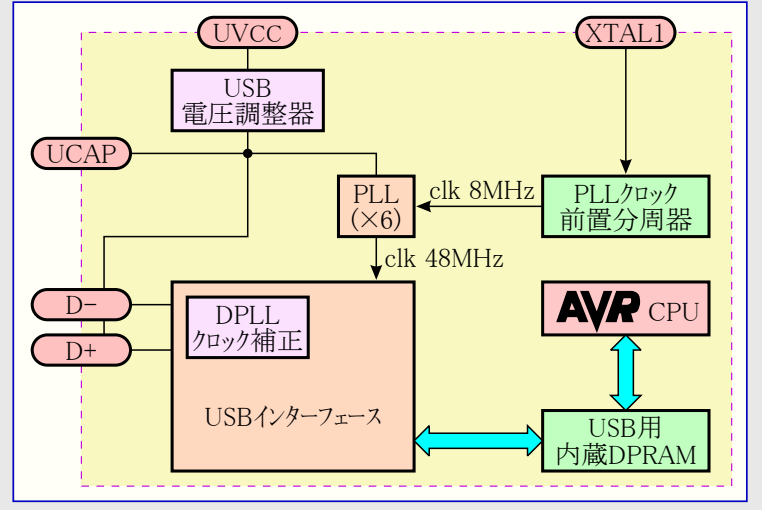
USB制御器は2ポートメモリ(DPRAM)に格納されたデータの流れへUSB接続をインターフェースするハードウェアを提供します。

このUSB制御器は内部PLL出力の48MHz $\pm$ 0.25%基準クロックを必要とします。このPLLはUSBインターフェース用に内部高周波数(48MHz)クロックを生成し、PLL入力には外部低周波数から生成されます(USB周波数精度とジッタを満足するためのクリスタル発振器またはXTAL1ピンからの外部クロック入力、これらのクロック元だけがUSB制御器の正しい機能を許します)。

48MHzクロックは受信したUSB差動データからの12MHz全速(Full-speed)のビットクロック生成と、全速USB装置許容誤差に従ったデータ送信に使われます。クロック補正はUSBバスの細動(ジッタ)仕様に適合したデジタル位相固定化閉路(DPLL)部によって行われます。

USBの電気的特性に従うため、USBパッド(D+とD-)は3.0～3.6V範囲内で給電されるべきです。AT90USB82/162は5.5Vまで給電できるので、内蔵電圧調整器がUSBパッド給電を保証できます。

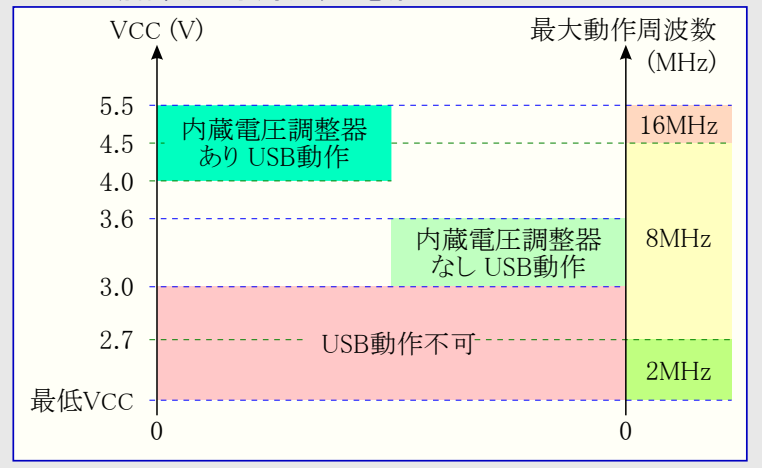
図19-1. USB制御器概要構成図



19.3. 代表的な応用実装

USB動作種別や対象応用電源に応じて、AT90USB82/162は異なる代表的ハードウェア実装を必要とします。

図19-2. 動作種別 対 周波数と電源



19.3.1. バス給電装置

図19-3. 5V入出力での代表的バス給電応用

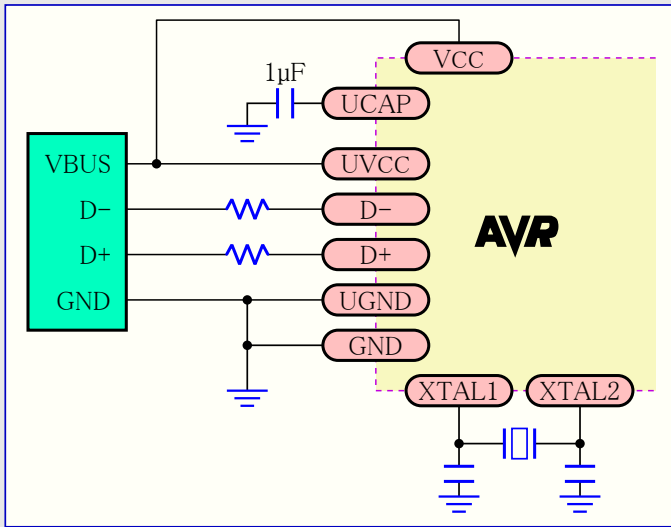
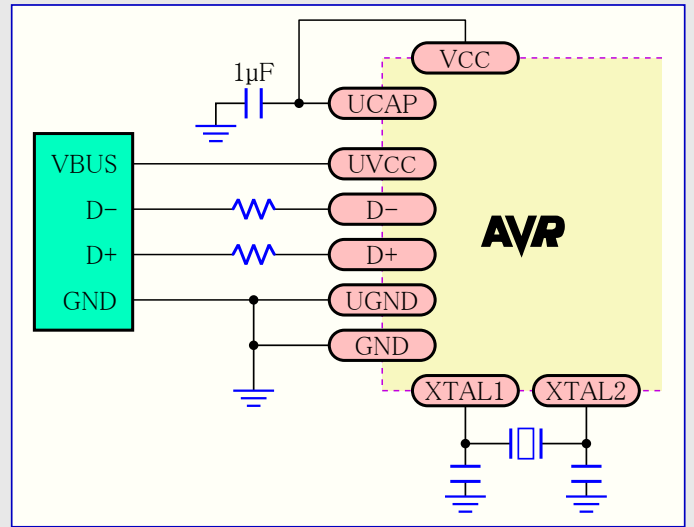


図19-4. 3V入出力での代表的バス給電応用



19.3.2. 自己給電装置

図19-5. 3.4～5.5V入出力での代表的自己給電応用

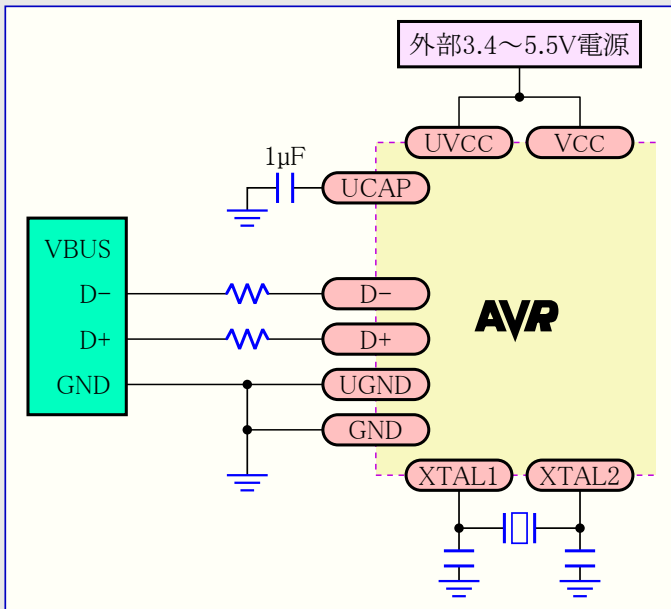
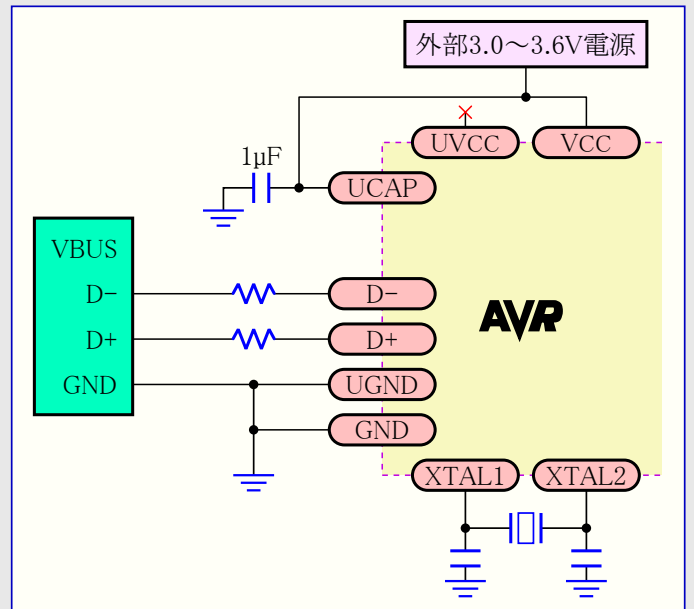


図19-6. 3.0～3.6V入出力での代表的自己給電応用



19.3.3. 設計の指針

- USBデータ線の直列抵抗器は値が 22Ω ($\pm 5\%$)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パッド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 $1\mu\text{F}$ ($\pm 10\%$)であるべきです。
- VBUS線上には $10\mu\text{F}$ コンデンサが強く推奨されます。

19.4. 一般的な動作

19.4.1. 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
- デバッグWIRE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)。

19.4.2. 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。

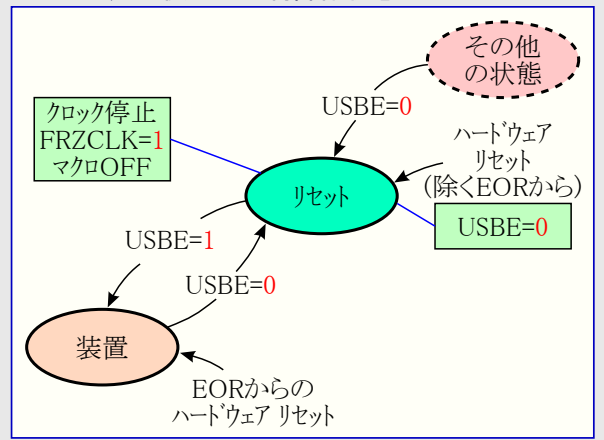
ハードウェア リセット後のUSB制御器状態は「リセット」です。この状態は以下です。

- USB制御(USBCON)レジスタのUSB全体許可(USBEN)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。

USBEN設定(=1)後、USB制御器は装置状態へ移行します。

USB制御器はUSBENの解除(0)によって何時でも「停止」にできます。実際にはUSBEN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19-7. リセット後のUSB制御器状態



19.4.3. 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19-8. USB割り込み概要

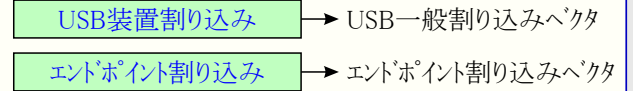
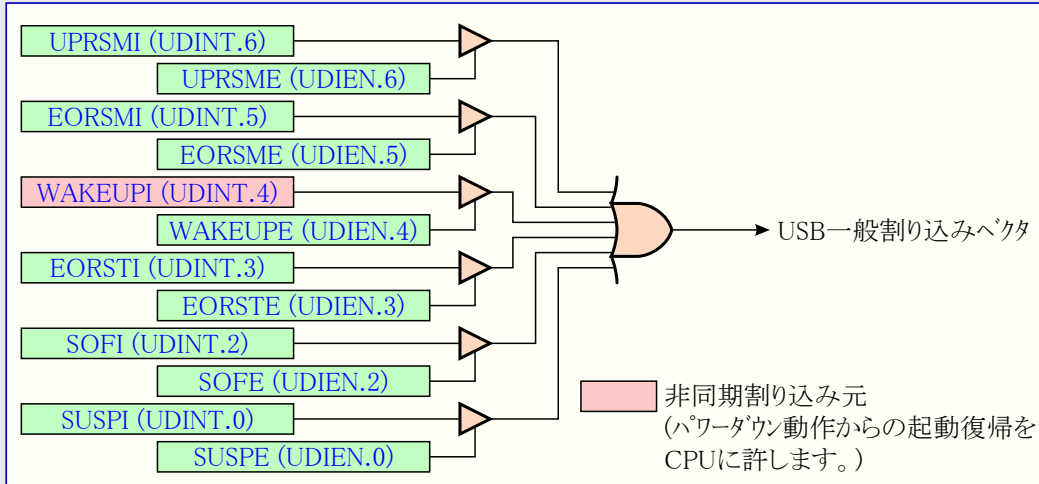


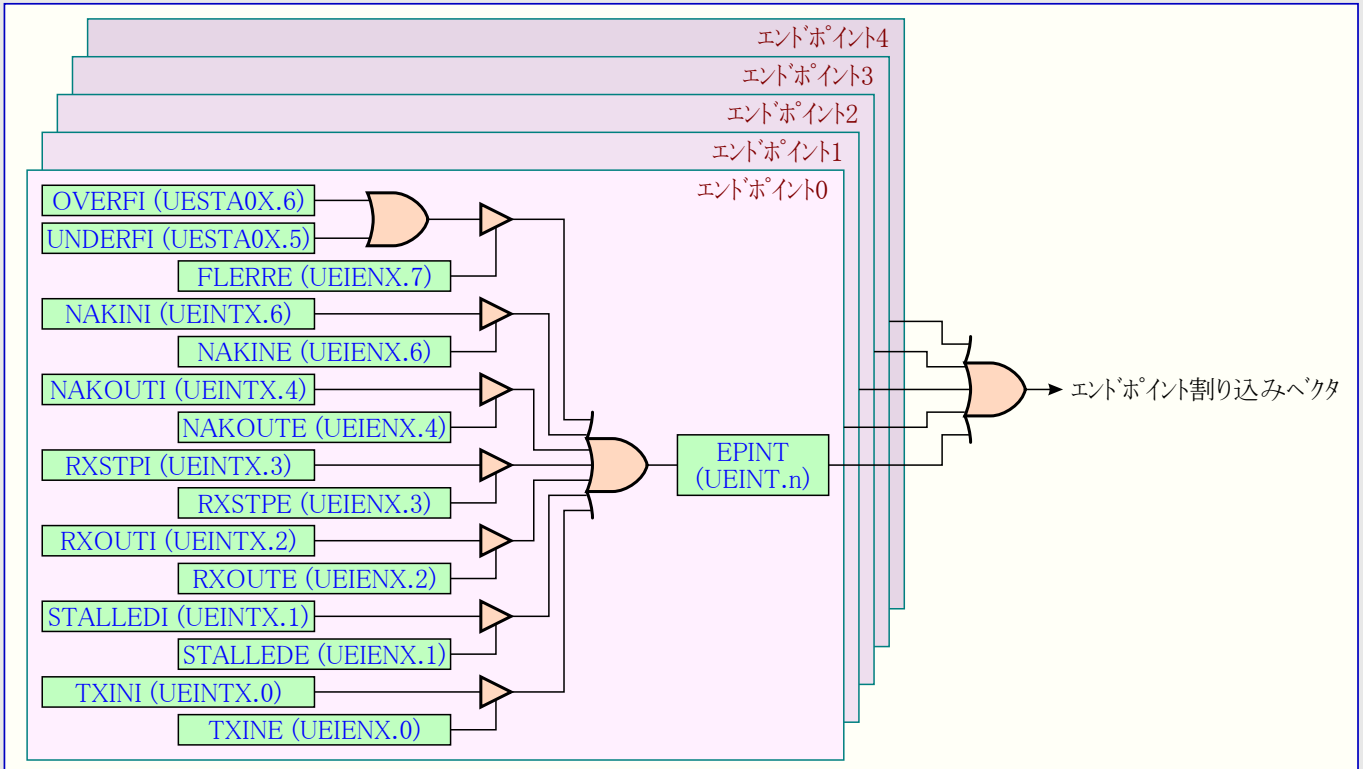
図19-9. USB一般割り込みベクタ割り込み元



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割り込みを除いて、これら割り込みの各々は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。

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19-10. 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。

19.5. 電力動作種別

19.5.1. 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制御器が走行中またはそうでなくても使われます。CPUは何れかのUSB割り込みで起動復帰します。

19.5.2. パワーダウン動作

この動作では発振器が停止され、全ての部分(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 (単一非同期割り込み)

19.5.3. 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
- DPRAM直接アクセス (DPADD7~0, UEDATX) (訳注: DPADDレジスタ削除により直接アクセス不能)
- UDCON (分離(DETACH), ...)
- UDINT
- UDIEN

更にFRZCLKが設定(1)されている時は次の非同期割り込みだけが起動され得ます。

- CPU起動復帰割り込み(WAKEUPI)

19.6. メリ管理

本制御器は以下のメリ割り当て管理だけを支援します。

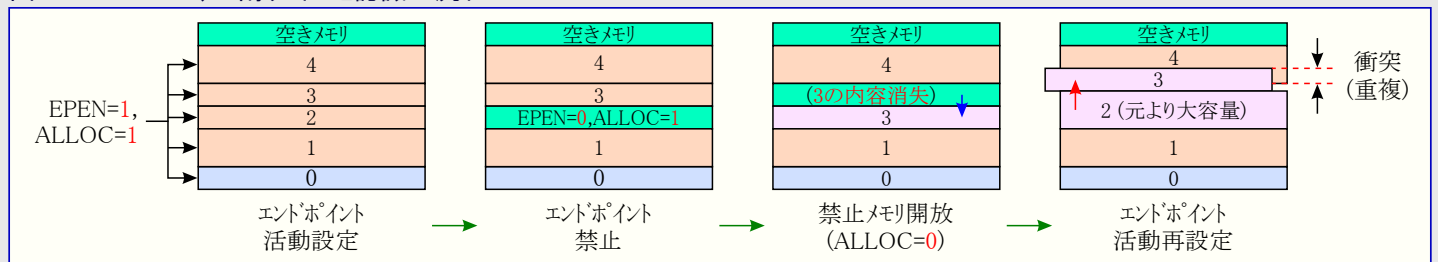
エンドポイントの予約は(エンドポイント0から最終エンドポイントへ)順に増やす場合にだけ行えます。従ってファームウェアはこれと同じ順で設定すべきです。

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 " $k+1$ " のメリは上へ移動し、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(EPESIZE, EPBK)やALLOCビットのどちらも解除(0)しません。メ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メリの割り当てと認識を図示します。

図19-11. USBメリの割り当てと認識の流れ



- 最初にエンドポイント0~4は昇順で設定されます。各メリはDPRAM内に予約されます。
- そしてエンドポイント2が禁止(EPEN=0)されますが、メ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3が下へ移動しますが、エンドポイント4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のエンドポイント2の再設定を選ぶ場合です。制御器はエンドポイント1の後にメリを予約し、エンドポイント3を自動的に移動します。エンドポイント4は動かず、エンドポイント3と4の共通領域使用で衝突(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。

19.7. 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。

図19-13. ハットの状態遷移

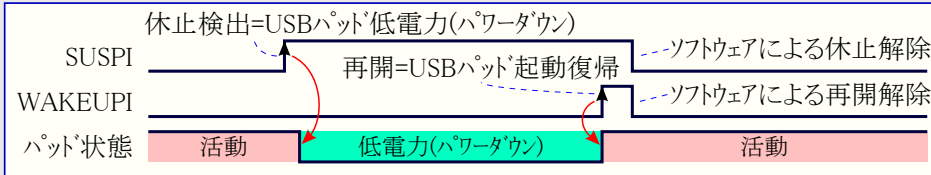
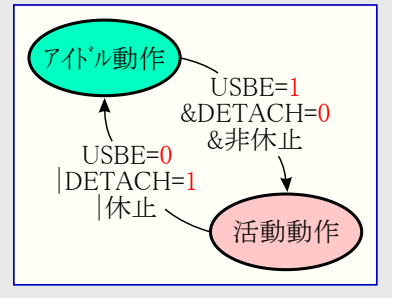


図19-12. ハットの動き



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。

19.8. D+/D-読み書き

D+とD-のレベルは**USB/PS2ソフトウェア出力許可(UPOE)レジスタ**使用で読み書きできます。USB制御器は書き込み値に対して許可しなければなりません。読み込み操作について、USB制御器は許可または禁止にできます。

19.9. USB一般用レジスタ

19.9.1. USB制御レジスタ (USB Control register) USBCON

ビット (\$D8)	7	6	5	4	3	2	1	0	
	USBE	—	FRZCLK	—	—	—	—	—	USBCON
Read/Write	R/W	R	R/W	R	R	R	R	R	
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4~0 – 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。

19.9.2. USB/PS2 ソフトウェア出力許可レジスタ (USB/PS2 Software Output Enable register) UPOE

ビット (\$FB)	7	6	5	4	3	2	1	0	
	UPWE1	UPWE0	UPDRV1	UPDRV0	SCKI	DATAI	DPI	DMI	UPOE
Read/Write	R/W	R/W	R/W	R/W	R	R	R	R	
初期値	0	0	0	0	0	0	0	0	

- ・ビット7,6 – UPWE1,0 : USB/PS2出力許可 (USB/OS2 Output enable)

USB/PS2ソフトウェア駆動を許可または禁止するには、次の設定でこれらのビットを設定してください。

00=直接駆動禁止、**01**=(予約)、**10**=USBレベルでD+/D-直接駆動(UPDRV1,0値)、**11**=PS/2レベルでD+/D-直接駆動(UPDRV1,0値)

- ・ビット5,4 – UPDRV1,0 : USB/PS2直接駆動値 (USB/PS2 direct drive values)

UPWE1,0設定に従ってD+/SCKに書く(出力する)ための値をUPDRV1に書いてください。UPWE1,0設定に従ってD-/SDATAに書く(出力する)ための値をUPDRV0に書いてください。

• **ビット3 – SCKI : SCK入力値** (SCK Input value)

SCK(PS/2パット)で1が読まれる場合、このビットはハードウェアによって1に設定されます。SCK(PS/2パット)で0が読まれる場合、このビットはハードウェアによって0に設定されます。

• **ビット2 – DATAI : DATA入力値** (DATA Input value)

SDATA(PS/2パット)で1が読まれる場合、このビットはハードウェアによって1に設定されます。SDATA(PS/2パット)で0が読まれる場合、このビットはハードウェアによって0に設定されます。

• **ビット1 – DPI : D+入力値** (D+ Input value)

D+(USBパット)で1が読まれる場合、このビットはハードウェアによって1に設定されます。D+(USBパット)で0が読まれる場合、このビットはハードウェアによって0に設定されます。

• **ビット0 – DMI : D-入力値** (D- Input value)

D-(USBパット)で1が読まれる場合、このビットはハードウェアによって1に設定されます。D-(USBパット)で0が読まれる場合、このビットはハードウェアによって0に設定されます。

19.10. 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

- 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エンドポイント0設定)
- 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

20. USB装置動作

20.1. 序説

USB装置制御器は全速(Full-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4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1,2 : 64バイトまでのFIFO容量設定可能
- エンドポイント3,4 : 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。

20.2. 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。

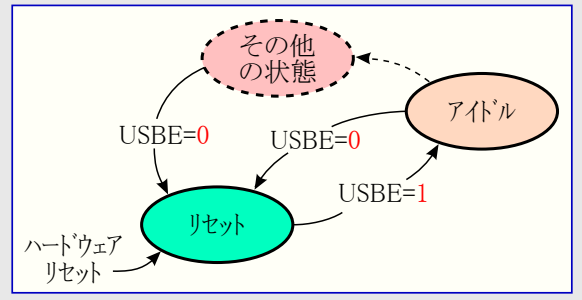
- 電力消費を最小とするため、全体クロックが停止されます(USB制御(USBCON)レジスタのUSBクロック停止(FRZCLK)=1)。
- USB装置制御器の内部状態はリセットです(全てのレジスタがそれらの既定値でリセットされます。USB装置制御(UDCON)レジスタの分離(DETACH)ビットが設定(1)されることに注意してください)。
- エンドポイントのバンクはリセットされます。
- D+のプルアップは非活性(分離(Detach)動作)にされます。

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(USB0)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20-1. リセット後のUSB装置制御器状態



20.3. 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トが復元(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。

20.4. USB リセット

USB線上でUSBリセット(最小100μs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状態に留まります。
- 既定制御エンドポイントの交互データ番号が解除(初期化)されます。

ハードウェアリセット機能が選ばれる場合、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(USBリセット後と同じに留まる)。

20.5. エンドポイント選択

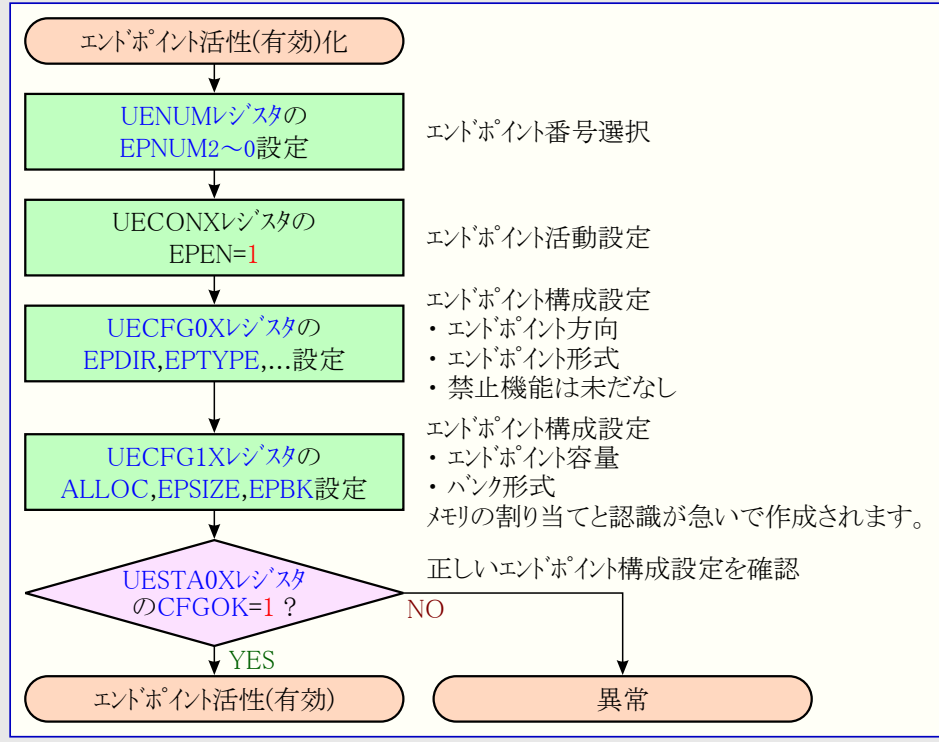
CPUによって何らかの操作が行われる前にエンドポイントが先に選ばれなければなりません。これはCPUによって管理されるエンドポイント番号でエンドポイント番号(UNUM)レジスタのエンドポイント番号(EPNUM2~0)ビットを設定することによって行われます。

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。

20.6. 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。

図20-2. エンドポイント活性(有効)化流れ図



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125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122頁の「メモリ管理」をご覧ください。

20.7. 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(USBFE)=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。

20.8. 休止、起動復帰と再開

USB線が非活性(J状態)の3ms間後、制御器は**休止(Suspend)割り込み(SUSPI)**を設定(1)し、(許可されていれば)対応する割り込みを起動します。そしてファームウェアは**USB制御(UDCON)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は更に低い電力消費にさせるのにOLLの禁止やアイドル動作へ移行することもできます(バス給電応用で重要)。

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ったと直ぐに、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。

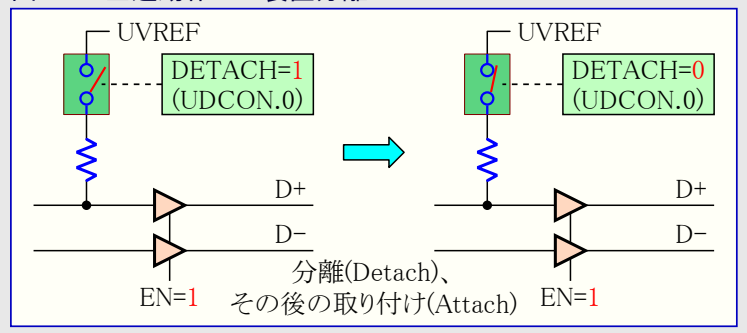
20.9. 分離 (切り離し)

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(信号線放電時間を考慮しなければなりません)。

- USB装置制御器が全速(Full-speed)動作の場合、DETACHビットの設定(1)はD+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)はD+上のプルアップを接続します。

図20-3. 全速動作での装置分離



20.10. 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))要求は、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)されている場合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。これはUSB信号線の5ms非活性後、制御器によって自動的に行われます。
- 制御器が上方向再開(Upstream resume)送信を始めると、**上方向再開割り込み(UPRSMI)フラグ**が設定(1)され、(許可されていれば)割り込みが起動されます。SUSPIが設定(1)されていた場合、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。

20.11. 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)ハント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)でハントシェイクされます。

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ょう。

不能応答(STALL)ハント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。

20.11.1. 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し得る第8節の検査に適合します。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。

20.11.2. 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。

20.12. 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。

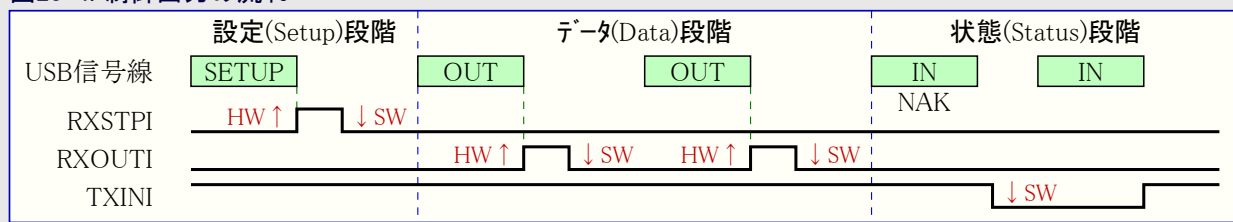
制御エンドポイントは割り込みで管理されるのではなく、単に状態ビットのポーリングによって管理されるべきです。

20.12.1. 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(NAKIND)**のポーリングができ、処理単位(Transaction)は既に状態段階です。

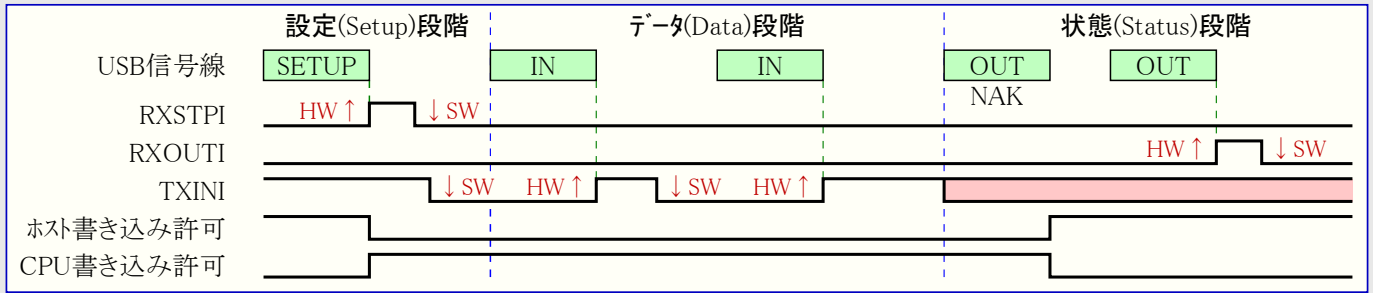
図20-4. 制御出力の流れ



20.12.2. 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20-5. 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、他の何れのフラグも解除(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。

20.13. 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。

20.13.1. 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20-6. 1つのOUTデータバンク例

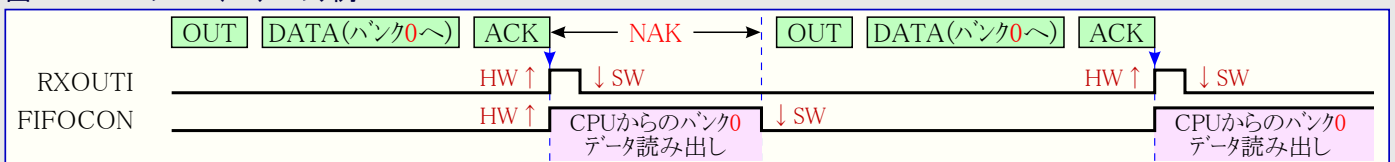
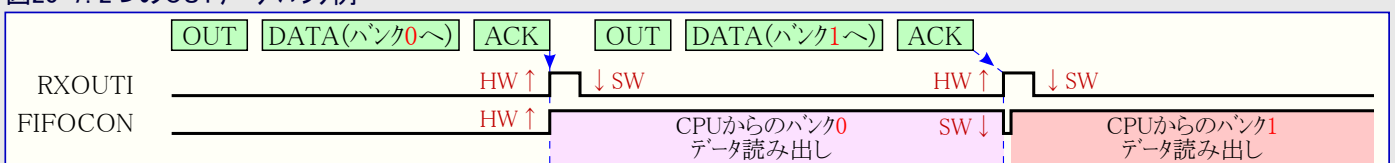


図20-7. 2つのOUTデータバンク例



20.13.2. 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(以下条件)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**がハードウェアによって解除(**0**)された直後

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、)RXOUTIが直ちに設定(**1**)されます。

20.14. 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。

20.14.1. 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20-8. 1つのINデータバンク例

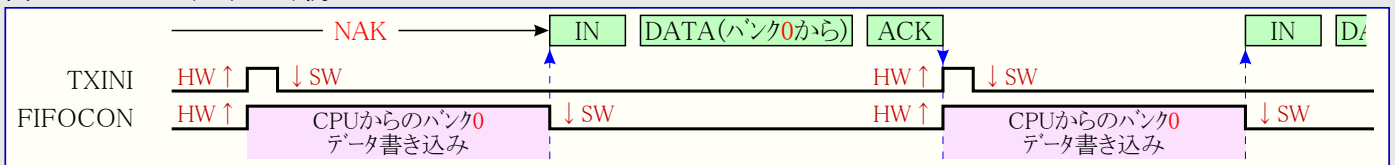
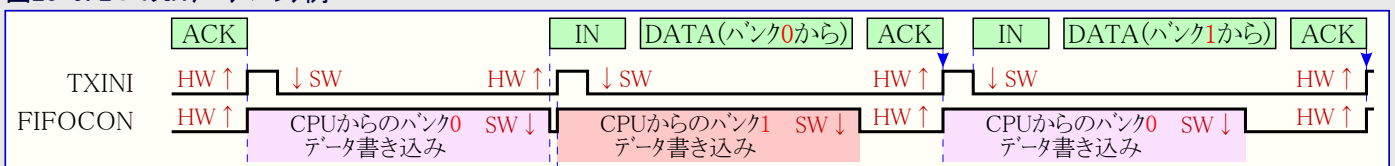


図20-9. 2つのINデータバンク例



20.14.2. 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。

- バンクが空のとき、**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(以下条件)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**がハードウェアによって解除(**0**)された直後

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。

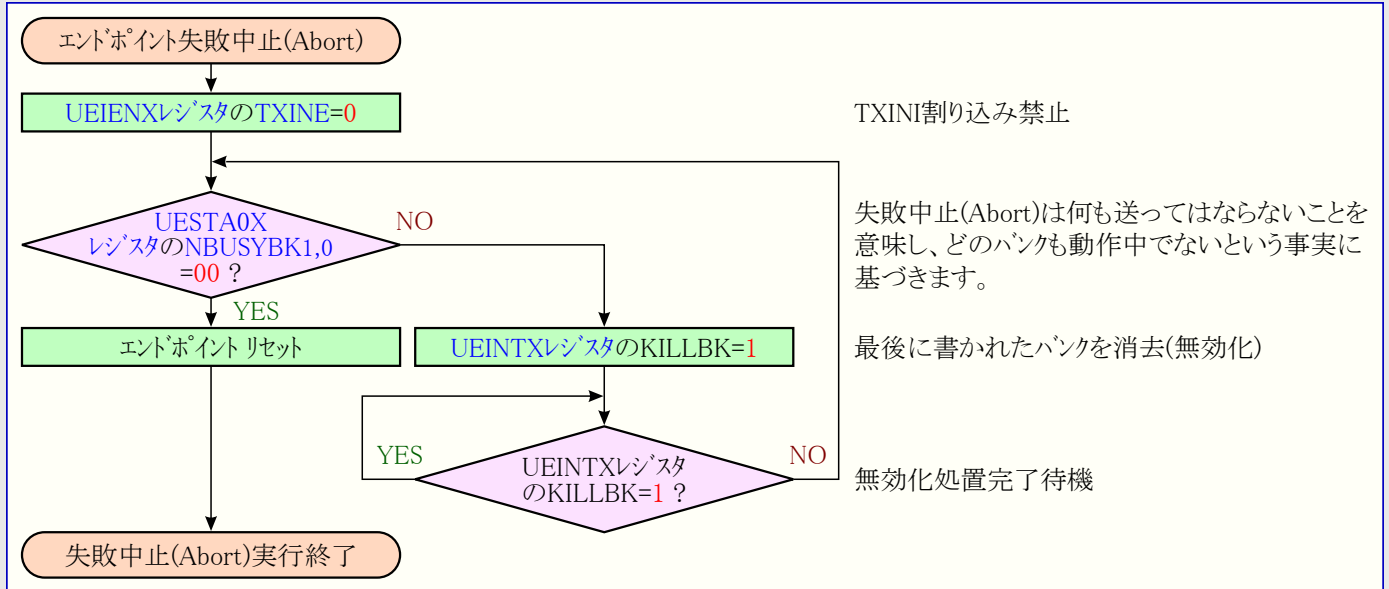
20.14.3. 失敗中止(Abort)

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20-10. 失敗中止の流れ図



20.15. 等時(Isochronous)動作

20.15.1. 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る場合(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。

20.15.2. 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。

20.16. 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る場合(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。

20.17. 装置割り込み

次図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。

- 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20-11. USB装置制御器割り込み元

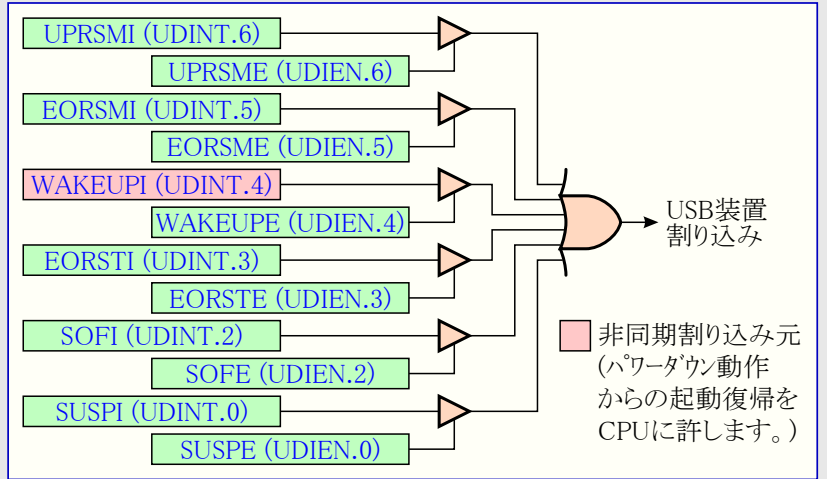
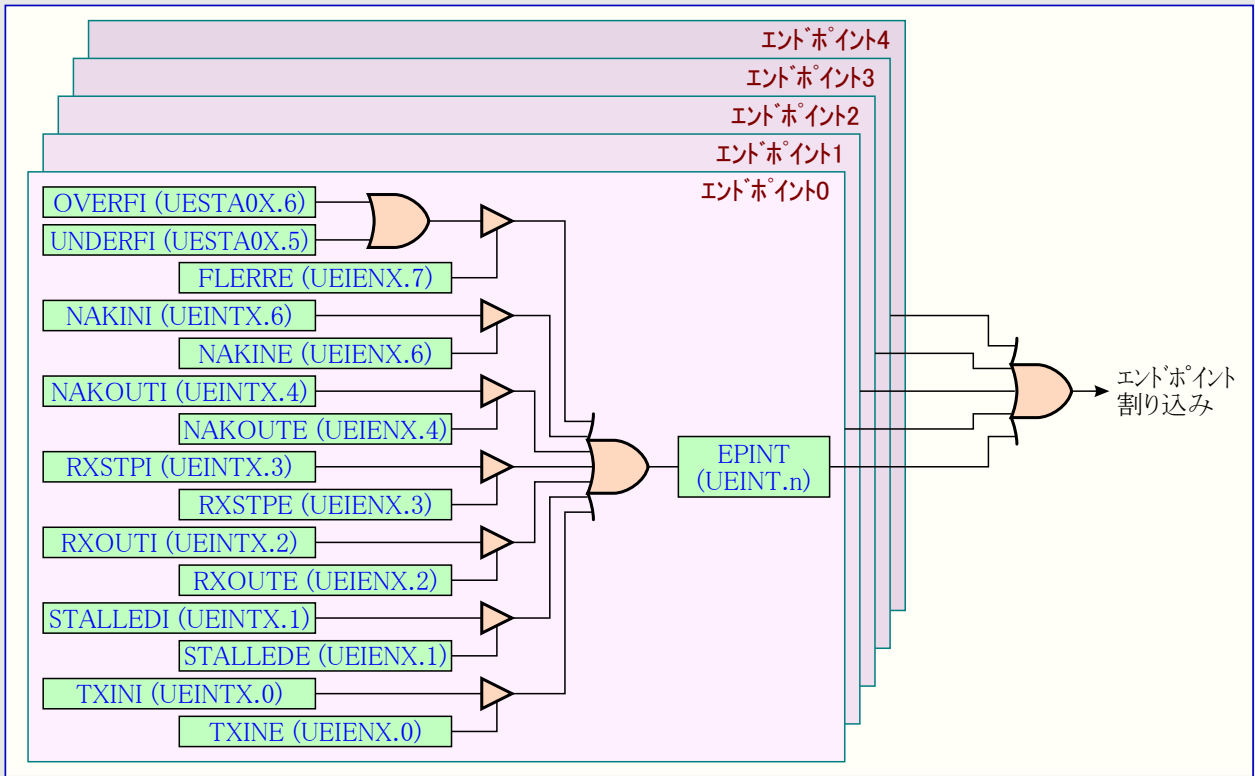


図20-12. USBエンドポイント割り込みベクタ割り込み元



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)

20.18. USB装置一般用レジスタ

20.18.1. USB装置制御レジスタ (USB Device Control Register) UDCON

ビット (\$E0)	7	6	5	4	3	2	1	0	
	–	–	–	–	–	RSTCPU	RMWKUP	DETACH	UDCON
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	1	

- 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 – RSTCPU : CPUリセット (USB Reset CPU Bit)

次のUSBリセットで(USB制御器の禁止を除いて)CPUをリセットするには、このビットをファームウェアによって設定(1)してください。このビット値はリセット終了(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(設定(1)に留まります)。このビットはUSB制御器が禁止されると、リセット(0)されます。

- 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。遠隔起動復帰送信を許すには**休止割り込み要求(SUSPI)フラグ**が設定(1)されなければなりません。

このビットはハードウェアによって解除(0)されます。ソフトウェアでの解除(0)は無効です。

より多くの詳細については127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+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127頁の「**分離 (切り離し)**」をご覧ください。

20.18.2. USB装置割り込み要求レジスタ (USB Device Interrupt Register) UDINT

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(UPRSMI)**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(EORSMI)**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)非アイドル信号瀟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(WAKEUPI)**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127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(EORSTI)**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127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。

20.18.3. USB装置割り込み許可レジスタ (USB Device Interrupt Enable Register) UDIEEN

ビット (\$E2)	7	6	5	4	3	2	1	0	
	–	UPRSME	EORSME	WAKEUPE	EORSTE	SOFE	–	SUSPE	UDIEEN
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。

20.18.4. USB装置アドレスレジスタ (USB Device Address Register) UDADDR

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(USBアドレス)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126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。

これは解除(0)されるべきではありません。

20.18.5. USB装置フレーム番号レジスタ (USB Device Frame Number Register) UDFNUMH, UDFNUML (UDFNUM)

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	UDFNUML
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UDFNUMLは原書での個別表記を統一表記に纏めました。

20.18.6. USB装置フレーム番号状態レジスタ (USB Device Maintained Frame Number Register) UDMFN

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。

20.19. USB装置エンドポイント用レジスタ

20.19.1. エンドポイント番号レジスタ (USB Endpoint Number Register) UENUM

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125頁の「エンドポイント選択」項をご覧ください。

EPNUM2～0=100よりも大きな値は許されません。

20.19.2. エンドポイントリセットレジスタ (USB Endpoint Reset Register) UERST

ビット (\$EA)	7	6	5	4	3	2	1	0	
	–	–	–	EPRST4	EPRST3	EPRST2	EPRST1	EPRST0	UERST
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

• ビット7～5 – Res : 予約 (Reserved)

これらのビット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4～0 – EPRST4～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125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。

20.19.3. エンドポイントX制御レジスタ (USB Endpoint X Control Register) UECONX

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128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128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)

装置構成設定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126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。

20.19.4. エンドポイントX構成0レジスタ (USB Endpoint X Configuration 0 Register) UECFG0X

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。

20.19.5. エンドポイントX構成1レジスタ (USB Endpoint X Configuration 1 Register) UECFG1X

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、1xx=予約(本設定使用禁止)

- 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126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。

20.19.6. エンドポイントX状態0レジスタ (USB Endpoint X Status 0 Register) UESTA0X

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ると、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131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ると、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131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

20.19.7. エンドポイントX状態1レジスタ (USB Endpoint X Status 1 Register) UESTA1X

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) (デバッグ目的用)

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。

20.19.8. エンドポイントX割り込み要求レジスタ (USB Endpoint X Interrupt Register) UEINTX

ビット (\$E8)	7	6	5	4	3	2	1	0	UEINTX
	FIFOCON	NAKINI	RWAL	NAKOUTI	RXSTPI	RXOUTI	STALLEDI	TXINI	
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131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。

20.19.9. エンドポイントX割り込み許可レジスタ (USB Endpoint X Interrupt Enable Register) UEIENX

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。

20.19.10. エンドポイントX データレジスタ (USB Endpoint X Data Register) UEDATX

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。

20.19.11. エンドポイントX バイト数レジスタ (USB Endpoint X Byte Count Register) UEBCLX

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7~0 – BYCT7~0 : 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。

20.19.12. エンドポイント割り込み要求レジスタ (USB Endpoint Interrupt Register) UEINT

ビット (\$F4)	7	6	5	4	3	2	1	0	
	–	–	–	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7~5 – 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4~0 – EPINT4~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。

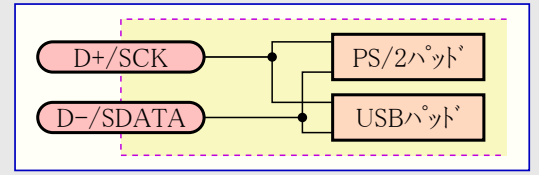
21. PS/2

21.1. 特性

PS/2パッドの入出力特性は次のとおりです。

- 5.5V支援 (最小4.5V, 最大5.5V)
- オープンドレイン出力
- 1k Ω プルアップでのMCU Lowレベル出力駆動: $V_{OLmax}=0.7V$
- 1k Ω プルアップと500pF負荷での下降時間: $250ns < t_{fall} < 1\mu s$
- ファームウェアによって制御される(5Vで)3.2k $\Omega \sim 6.5k\Omega$ の内蔵プルアップ
- 禁止時のHi-Z

図21-1. USBパッドとPS/2パッドの多重化



21.1.1. PS/2制御レジスタ (PS/2 Control Register) PS2CON

ビット (\$FA)	7	6	5	4	3	2	1	0	
	—	—	—	—	—	—	—	PS2EN	PS2CON
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 – PS2EN : PS/2パッド許可 (PS/2パッド許可)

PS/2パッドを許可するにはこのビットを1に設定してください。

PS/2パッドを禁止するにはこのビットを0に設定してください。

パッド駆動についてのより多くの詳細に関しては123頁の「USB/PS2ソフトウェア出力許可レジスタ (UPOE)」をご覧ください。

その項で記述されたUPOEレジスタはパッドの値の読み書きを許します。PS/2規約は全てソフトウェアで扱われなければなりません。

図21-2. PS/2パッド構成図

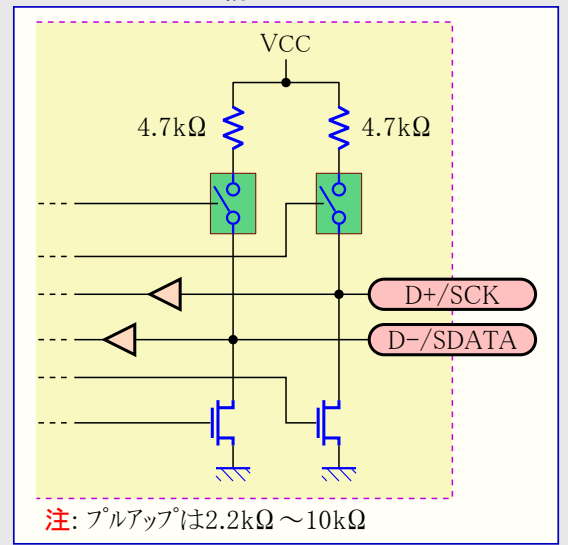


表21-1. PS/2 DC特性

シンボル	項目	条件	最小	代表	最大	単位
UVCC	PS/2電源電圧	UVCC=4.5～5.5V	4.0	4.5	5.5	V
VIL	Lowレベル入力電圧				0.7	
VIH	Highレベル入力電圧		2.4			
RPU	プルアップ抵抗		4.4	5.1	6.5	k Ω

表21-2. PS/2 AC特性

シンボル	項目	条件	最小	代表	最大	単位
t_{fall_output}	下降時間	UVCC=4.5～5.5V, 負荷抵抗=1k Ω 負荷容量=500pF, 10%～90%	330	375	450	ns
t_{spike}	尖頭濾波幅				60	

22.0.2. デジタル入力禁止レジスタ1 (Digital Input Disable Register 1) DIDR1

ビット (\$7F)	7	6	5	4	3	2	1	0	
	–	–	–	–	–	–	AIN1D	AIN0D	DIDR1
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) (誤注: 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1,0 – AIN1D, AIN0D : AIN1, AIN0 デジタル入力禁止 (AIN1, AIN0 Digital Input Disable)

このビットが論理1を書かれると、AIN1/0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。AIN1/0ピンにアナログ信号が印加され、そのピンからのデジタル入力が必要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。

23. デバッグWIRE 内蔵デバッグ システム

23.1. 特徴

- 完全なプログラムの流れ制御
- RESETピンを除くデジタルとアナログ両方でのチップ全機能のエミュレート
- 実時間(リアルタイム)動作
- シンボリックデバッグ支援 (アセンブリ及びC言語または他の高位言語)
- 無制限数のプログラム中断点(ブレークポイント: ソフトウェア中断点使用)
- 邪魔しない動作
- 実デバイスと同じ電気的特性
- 自動設定システム
- 高速動作
- 不揮発性メモリのプログラミング

23.2. 概要

デバッグWIRE内蔵デバッグシステムはCPUでのAVR命令実行、プログラムの流れ制御、各種不揮発性メモリのプログラミングのための1本線の双方向インターフェースを使います。

23.3. 物理インターフェース

デバッグWIRE許可(DWEN)ヒューズがプログラム(0)され、施錠ビットが非プログラム(1)にされると、対象デバイス内のデバッグWIREシステムが活性(有効)にされます。RESETポートピンはプルアップ許可のANDタイ(オープンドレイン)双方向I/Oピンとして設定され、対象デバイスとエミュレータ間の通信路になります。

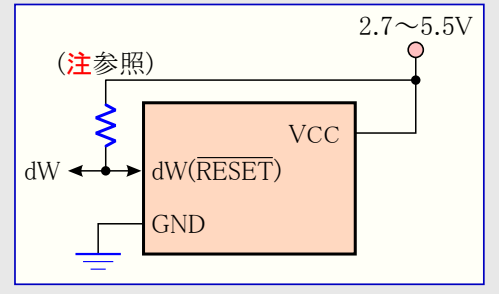
図23-1.はエミュレータと許可したデバッグWIREでの対象MCUとの接続の図を示します。システムクロックはデバッグWIREによって影響を及ぼされず、常にCKSELヒューズで選んだクロック元です。

デバッグWIREが使われるシステムの設計時、正しい動作のために次の注意点が厳守されなければなりません。

- RESETピンのVCCへの直接的な接続では動作しません。
- RESETピンに挿入したコンデンサはデバッグWIRE使用時、切断されなければなりません。
- 全ての外部リセット元は切断されなければなりません。

注: JTAGICEmk II のいくつかの公開版は5V動作時で8~14k Ω のプルアップ抵抗を必要とするかもしれません。

図23-1. デバッグWIRE構成図



23.4. ソフトウェア中断点(ブレークポイント)

デバッグWIREはAVRのBREAK命令によってプログラムメモリの中断点を支援します。AVR Studio®での中断点設定はプログラムメモリにBREAK命令を挿入します。BREAK命令で置換した(元の)命令は保存されます。プログラム実行が継続されると、プログラムメモリから継続される前に保存した命令が実行されます。一時停止(ブレーク)はプログラムにBREAK命令を置くことによって手動で挿入できます。

フラッシュメモリは中断点の変更される度毎に再書き換えされなければなりません。これはデバッグWIREインターフェースを通してAVR Studioによって自動的に操作されます。従って中断点の使用はフラッシュメモリのデータ保持力を低下させます。デバッグ目的に使ったデバイスは最終顧客へ出荷すべきではありません。

23.5. デバッグWIREの制限

デバッグWIRE通信(dW)ピンは物理的に外部リセット(RESET)と同じピンに配置されます。従ってデバッグWIREが許可されると、外部リセット元が支援されません。

デバッグWIREシステムは全速度、換言するとCPUのプログラムが走行する時に全I/O機能を正確エミュレートします。CPUが停止される時にデバッグ(AVR Studio)経由でいくつかのI/Oレジスタをアクセスする間、注意が払われなければなりません。この制限の詳細説明についてはデバッグWIRE資料をご覧ください(訳注: 共通性から本行追加)。

プログラム(0)にしたDWENヒューズは全休止形態でクロック系のいくつかの部分の走行を許可します。これは休止間中の消費電力を増加します。従ってDWENヒューズはデバッグWIREが使われない場合、禁止されるべきです。

23.6. デバッグWIREに関連するI/Oメモリ内のレジスタ

次項はデバッグWIREで使うレジスタを記述します。

23.6.1. デバッグWIRE データレジスタ (debugWIRE Data Register) DWDR

ビット	7	6	5	4	3	2	1	0	
\$31 (\$51)	(MSB)							(LSB)	DWDR
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	

DWDRはMCU内で走行する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。このレジスタはデバッグWIREでだけアクセス可能で、従って通常動作で一般目的レジスタとして使えません。

24. 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。

24.1. 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(161頁の表25-11.参照)。このページ構成は通常動作に影響を及ぼしません。

24.2. 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24-2.参照)。各領域の容量は156頁の表24-8.と図24-2.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。

24.2.1. 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(148頁の表24-2.参照)。応用領域から実行される時にSPM命令が禁止されるので、応用領域はどんなブートローダコードも決して格納し得ません。

24.2.2. 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をアクセスできます。ブートローダ領域用保護基準はブートローダ施錠ビット(ブート施錠ビット1)によって選べます(148頁の表24-3.参照)。

24.3. 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147頁の図24-2.と156頁の表24-9.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。

24.3.1. 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150頁の「SPM命令制御/状態レジスタ(SPMCSR)」をご覧ください。

24.3.2. 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24-1. 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24-1. RWW領域とNRWW領域の関係

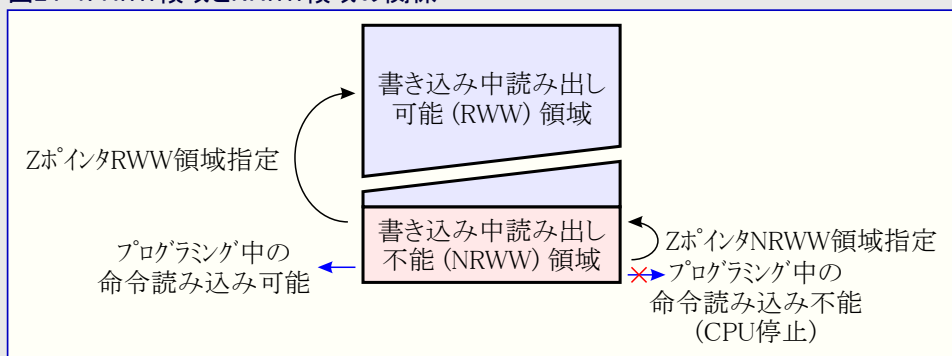
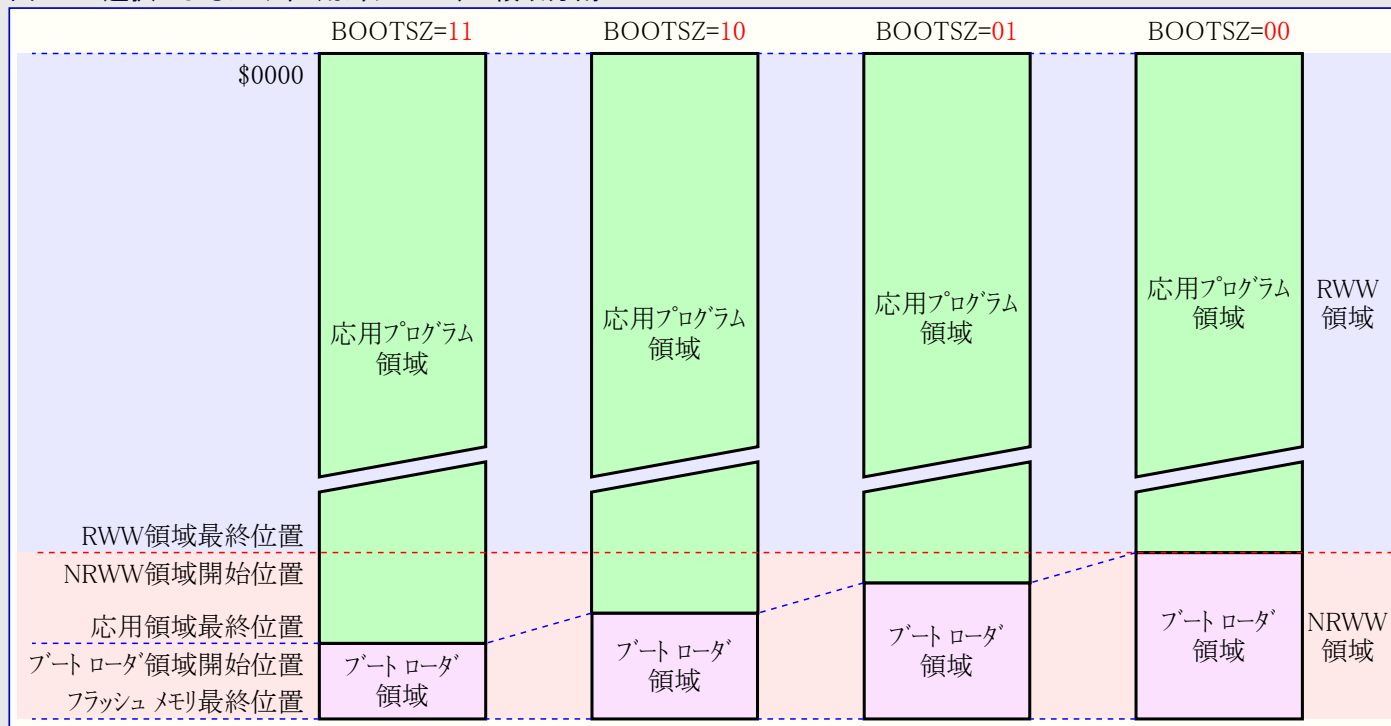


図24-2. 選択によるプログラム用フラッシュメモリの領域分割



注: 図内の各パラメータについては156頁の表24-8.で与えられます。

24.4. ブート ロード 施錠ビット

ブート ロード 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 ロード 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 ロード 領域だけを保護
- MCUによって更新するソフトウェアからフラッシュ メモリの応用領域だけを保護
- フラッシュ メモリ全体で更新するソフトウェアを許可

より多くの詳細については表24-2と表24-3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24-2. 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 ロード 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 ロード 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 ロード 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24-3. ブート ロード 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 ロード 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 ロード 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 ロード 領域への書き込みと、応用領域でのLPM命令によるブート ロード 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 ロード 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 ロード 領域での実行時に割り込みが禁止されます。

24.5. 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。

24.5.1. 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。

24.5.2. 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 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24-4. 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 (156頁の表24-8.参照)
1	応用リセット	\$0000

24.5.3. 外部ハードウェア状況

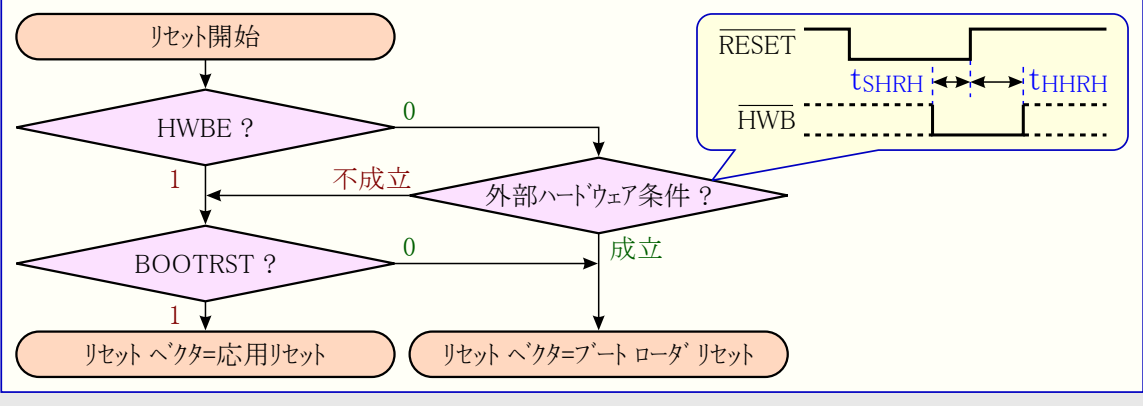
リセット下の特別なハードウェア条件でブート ロータの実行がリセット後に強制されるようにハードウェア ブート許可(HWBE)ヒューズをプログラム(0)できます(表24-5.をご覧ください)。

HWBEヒューズ許可時、ALE/ $\overline{\text{HWB}}$ 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ALE/ $\overline{\text{HWB}}$ 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24-3.をご覧ください)。

表24-5. 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	PD7/ $\overline{\text{HWB}}$ 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	PD7/ $\overline{\text{HWB}}$ 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。

図24-3. ブート処理内容



24.5.4. SPM命令制御/状態レジスタ (Store Program Memory Control and Status Register) SPMCSR

このレジスタはブートローダ動作を制御するために必要とされる制御ビットを含みます。

ビット	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(1)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153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(E)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153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。

24.6. 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

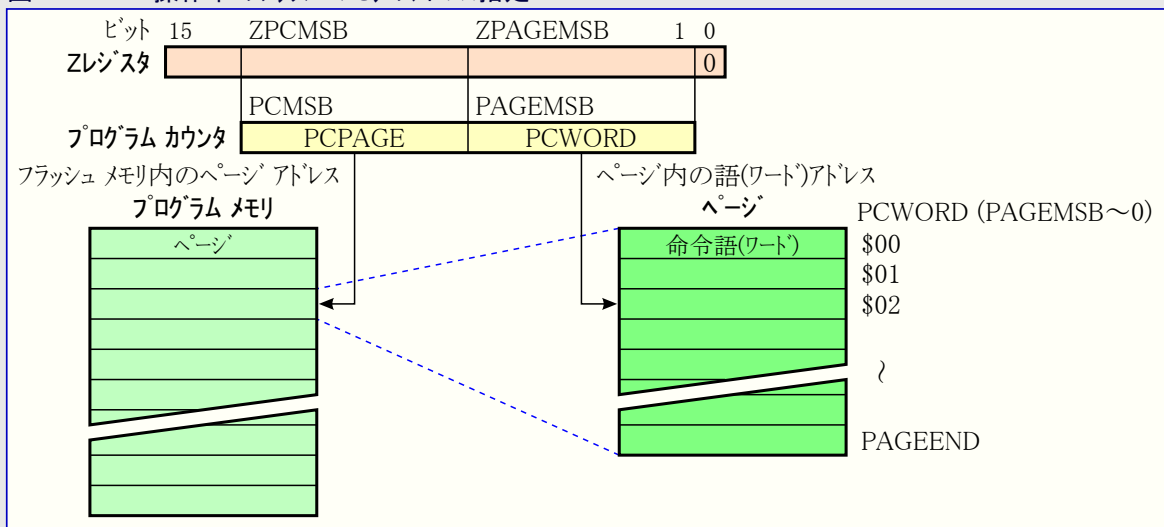
Zポインタ(レジスタ)はSPM命令でのアドレス指定に使われます。

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(161頁の表25-11.参照)、プログラムカウンタ(アドレスポインタ)は2つの違う領域を持つように扱われます。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24-4.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24-4. 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156頁の表24-10.で一覧されます。
PCPAGEとPCWORDは161頁の表25-11.で一覧されます。(訳注: 共通性から追加)

24.7. 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155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。

24.7.1. 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らない。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。

24.7.2. 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。

24.7.3. 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らない。Zポインタの他のビットは0を書かれなければならない。

- 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。

24.7.4. 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39頁の「割り込み」で記述されます。

24.7.5. 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らない。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。

24.7.6. 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らない。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39頁の「割り込み」で記述されるようにブートローダ領域(BLS)へ移動されるべきか、または割り込みが禁止されなければならない。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)しなければならない。例については155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。

24.7.7. 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24-2と表24-3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)00001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。

24.7.8. 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(EPE)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。

24.7.9. 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158頁の表25-5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158頁の表25-4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158頁の表25-3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。

24.7.10. 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表24-6.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24-6. 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発振器校正值	\$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。

24.7.11. フラッシュ メモリ データ化けの防止

低VCCの期間中、CPUとフラッシュ メモリの正しい動作に対して供給電圧が低すぎるためにフラッシュ メモリのプログラムが不正にされ得ます。これらの問題はフラッシュ メモリを使う基板段階の装置と同じで、同じ設計上の解決策が適用されるべきです。

フラッシュ メモリのプログラム化けは電圧が低すぎる時の2つの状態によって起こされます。1つ目としてフラッシュ メモリへの通常の書き込み手順は正しく動作するための最低電圧が必要です。2つ目として供給電圧が低すぎると、CPU自身が命令を間違えて実行し得ます。

フラッシュ メモリ化けは次の推奨設計によって容易に避けられます(1つは必須)。

- そのシステムでブート ロード 更新が必要ない場合、どんなブート ロード ソフトウェア更新をも防ぐためにブート ロード 施錠ビットをプログラム(0)してください。
- 不十分な供給電源電圧の期間中、AVR RESETを活性(Low)に保ってください。これは動作電圧が検出電圧と一致するなら、内部低電圧検出器(BOD)を許可することによって行えます。そうでなければ外部低VCCリセット保護回路が使えます。書き込み操作進行中にリセットが起こると、その書き込み動作は供給電源電圧が充分であれば完了されます。
- 低VCCの期間中、AVRコアを「パワーダウン休止動作」に保ってください。これはCPUが命令の復号と実行を試みるのを防ぎ、SPMCSR 従ってフラッシュ メモリを予期せぬ書き込みから効果的に保護します。

24.7.12. SPM命令使用時のフラッシュ メモリ用プログラミング(書き込み)時間

校正された内蔵RC発振器がフラッシュ メモリ アクセス時間に使われます。表24-7.はCPUからのフラッシュ メモリ アクセスに対する代表的なプログラミング時間を示します。

表24-7. SPM命令によるフラッシュ メモリのプログラミング時間

項目	最小	最大
SPM命令によるフラッシュ書き込み (ページ消去、ページ書き込み、施錠ビット書き込み)	3.7ms	4.5ms

24.7.13. 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)。

ラベル	命令	注釈
	.EQU PGSZB = PAGESIZE*2 .ORG SMALLBOOTSTART	; PGSZBはページ内のバイト数です。(PAGESIZEは語(ワード)数) ;
WRPG:	LDI SPMC, (1<<PGERS)+(1<<SELFPRGEN) CALL SPMJ	; [ページ消去] ; ページ消去SPMCSR値を取得 ; ページ消去 ; [RWW領域読み出し再許可]
	LDI SPMC, (1<<RWWSRE)+(1<<SELFPRGEN) 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 ; [RAMからフラッシュページ一時緩衝部へ転送]
WLP:	LDI CNTL, LOW(PGSZB) LDI CNTH, HIGH(PGSZB) LD R0, Y+ LD R1, Y+ LDI SPMC, (1<<SELFPRGEN) CALL SPMJ ADIW ZH:ZL, 2 SBIW CNTH:CNTH, 2 BRNE WLP	; バイト計数器を初期化 ; (削除) ; RAM上の下位データを取得(ポインタ進行) ; RAM上の上位データを取得(ポインタ進行) ; ページ一時緩衝部書き込みSPMCSR値を取得 ; 対応語(ワード)データをページ一時緩衝部に設定 ; ページ一時緩衝部ポインタ進行 ; 計数器を減数 (SUBI) ; 指定バイト数分継続 ; [ページ書き込み] ; ページ一時緩衝部先頭にポインタを復帰 ; (削除)
	SUBI ZL, LOW(PGSZB) SBCI ZH, HIGH(PGSZB) LDI SPMC, (1<<PGWRT)+(1<<SELFPRGEN) CALL SPMJ	; フラッシュ書き込みSPMCSR値を取得 ; 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<<SELFPRGEN) 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 ; [読み戻し照合(任意)]
	LDI CNTL, LOW(PGSZB) LDI CNTH, HIGH(PGSZB) SUBI YL, LOW(PGSZB) SBCI YH, HIGH(PGSZB)	; バイト計数器を初期化 ; (削除) ; RAMデータ先頭にポインタを復帰 ;
RLP:	LPM R0, Z+ LD R1, Y+ CPSE R0, R1 JMP ERROR	; フラッシュメモリから1バイト取得(ポインタ進行) ; RAMから1バイトデータを取得(ポインタ進行) ; 値一致でスキップ ; 不一致で異常処理へ
;		
	SBIW CNTH:CNTH, 1 BRNE RLP	; 計数器を減数 (SUBI) ; 指定バイト数分継続 ; [RWW領域へ復帰]
RTN:	IN TMP, SPMCSR SBRS TMP, RWWSB RET	; SPM命令制御/状態レジスタ値を取得 ; RWW領域多忙でスキップ ; 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<<SELFPRGEN) CALL SPMJ RJMP RTN	; RWW領域読み出し許可SPMCSR値を取得 ; RWW領域読み出し許可 ; 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 SBRC TMP, SELFPRGEN RJMP SPMJ	; SPM命令制御/状態レジスタ値を取得 ; 操作可能(直前のSPM完了)でスキップ ; 操作可まで待機
;		
	IN TMP, SREG CLI SBIC EECR, EEPE RJMP WAIT	; ステータスレジスタ値を保存 ; 全割り込み禁止 ; EEPROMプログラミング中以外でスキップ ; EEPROMプログラミング完了まで待機
WAIT:		
;		
	OUT SPMCSR, SPMC SPM OUT SREG, TMP RET	; SPM動作指定 ; 対応SPM動作実行 ; ステータスレジスタ値を復帰 ; 呼び出し元へ復帰

(訳注) AT90USB82では上記のCALL,JMP命令をRCALL,RJMP命令に置換してください。

24.7.14. AT90USB82/162用ブートローダのパラメータ

自己プログラミングの記述で使ったパラメータは表24-8.~10.で与えられます。

表24-8. 応用領域とブートローダ領域の分割設定 (AT90USB82/AT90USB162)

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	4/4	\$0000~\$0EFF/\$1EFF	\$0F00/\$1F00~\$0FFF/\$1FFF	\$0F00/\$1F00
1	0	512	8/8	\$0000~\$0DFF/\$1DFF	\$0E00/\$1E00~\$0FFF/\$1FFF	\$0E00/\$1E00
0	1	1024	16/16	\$0000~\$0BFF/\$1BFF	\$0C00/\$1C00~\$0FFF/\$1FFF	\$0C00/\$1C00
0	0	2048	32/32	\$0000~\$07FF/\$17FF	\$0800/\$1800~\$0FFF/\$1FFF	\$0800/\$1800

(ページ容量=64語=128バイト)

注: 各種BOOTSZヒューズ設定は図24-2.で示されます。

表24-9. RWW領域とNRWW領域の範囲 (AT90USB82/AT90USB162)

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	32/96	\$0000/\$0000~\$07FF/\$17FF
書き込み中読み出し不能(NRWW)領域	32/32	\$0800/\$1800~\$0FFF/\$1FFF

注: これら2つの領域についての詳細に関しては147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24-10. 図24-4.で使った各変数説明とZポイントの配置 (AT90USB82/AT90USB162)

変数名	対応値		意味
	PC	Zポイント (注)	
PCMSB	PC11/12		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12/13ビット、PC11/12~0)
PAGEMSB	PC5/5		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64/64語(ワード)には6/6ビット PC5/5~0が必要)
ZPCMSB		Z12/13	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)
ZPAGEMSB		Z6/6	PAGEMSB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、ZPAGEMSB=PAGEMSB+1)
PCPAGE	PC11/12~6/6	Z12/13~7/7	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	PC5/5~0	Z6/6~1	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: Z15~13/14: 常に無視されます。

Z0: 全てのSPM命令に対して0であるべきで、LPM命令に対するビット選択です。

自己プログラミング中のZポイントの使用についての詳細に関しては151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。

25. メモリプログラミング

25.1. プログラム メモリとデータ メモリ用施錠ビット

AT90USB82/162は非プログラム(1)のままか、表25-2.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25-1. 施錠ビット バイトの内容

名称	ビット番号	意味	既定値 (注)
—	7		1 (非プログラム)
—	6		1 (非プログラム)
BLB12	5	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)
BLB11	4		0 (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)
LB1	0		0 (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25-2. 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、またはデバッグWIRE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。デバッグWIREは全面的に禁止されます。

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。

25.2. ヒューズ ビット

AT90USB82/162には3つのヒューズ バイトがあります。表25-3.~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25-3. 拡張ヒューズ バイト一覧

名称	ビット	意味	既定値 (\$F4)
—	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。	0 (プログラム) HWBEピン有効
BODLEVEL2	2		1 (非プログラム)
BODLEVEL1	1	低電圧検出(BOD)リセットの制御と検出電圧選択。 (注1)	0 (プログラム)
BODLEVEL0	0		0 (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9-2をご覧ください。

表25-4. ヒューズ 上位バイト一覧

名称	ビット	意味	既定値 (\$D9)
DWEN (注4)	7	デバッグWIRE機能許可。	1 (非プログラム) デバッグWIRE不許可
RSTDISBL	6	PC1がI/OピンかまたはRESETピンかを選びます。	1 (非プログラム) PC1はRESETピン
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 タイマ常時有効。	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は未保護
BOOTSZ1	2	ブート ロータ容量選択。(表24-8.参照) (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 ロータ領域)選択。	1 (非プログラム) 応用領域

注1: SPIENヒューズは直列プログラミングでアクセスできません。

注2: BOOTSZ1,0既定値は最大ブート ロータ容量になります。156頁の表24-8.をご覧ください。

注3: 詳細については36頁の「ウォッチドッグ タイマ制御レジスタ(WDTCSR)」をご覧ください。

注4: 施錠ビットやRSTDISBLヒューズの設定に拘らず、DWEN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DWEN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。

表25-5. ヒューズ 下位バイト一覧

名称	ビット	意味	既定値 (\$5E)
CKDIV8 (注4)	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3)	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。 (注1)	0 (プログラム)
SUT0	4		1 (非プログラム)
CKSEL3	3	クロック種別選択。 (注2)	1 (非プログラム)
CKSEL2	2		1 (非プログラム)
CKSEL1	1		1 (非プログラム)
CKSEL0	0		0 (プログラム)

注1: SUT1,0の既定値は最大起動時間になります。詳細については20頁の表6-4.をご覧ください。

注2: CKSEL3~0の既定設定は外部クリスタル用発振器になります。詳細については19頁の表6-1.をご覧ください。

注3: 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4: 詳細については23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。

25.2.1. 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。

25.3. 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。

AT90USB82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 : \$93 フラッシュメモリ容量8Kバイトを示します。
- ③ \$002 : \$82 ②値\$93と合せ、AT90USB82を示します。

AT90USB162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 : \$82 ②値\$94と合せ、AT90USB162を示します。

25.4. 校正バイト

AT90USB82/162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。

25.5. 並列プログラミング

本章はAT90USB82/162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。

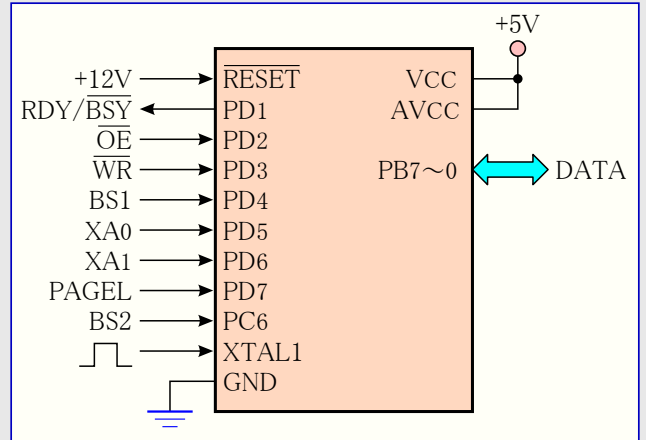
25.5.1. 信号名

本章でAT90USB82/162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25-1.**と**表25-6.**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25-9.**で示されます。

$\overline{\text{WR}}$ または $\overline{\text{OE}}$ 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25-10.**で示されます。

図25-1. 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25-6. 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)
$\overline{\text{OE}}$	PD2	入力	出力許可(負論理)
$\overline{\text{WR}}$	PD3	入力	書き込みパルス(負論理)
BS1	PD4	入力	バイト選択1 (表25-7.参照)
XA0	PD5	入力	XTAL動作ビット0
XA1	PD6	入力	XTAL動作ビット1
PAGED	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	PC6	入力	バイト選択2 (表25-7.参照)
DATA	PB7~0	入出力	双方向データバス ($\overline{\text{OE}}$ =Low時出力)

表25-7. 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25-8. プログラミング動作移行時のピン値

ピン名	シンボル	値
PAGED	Prog_enable[3]	0
XA1	Prog_enable[2]	0
XA0	Prog_enable[1]	0
BS1	Prog_enable[0]	0

表25-9. 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25-10. 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)	識別バイト、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25-11. 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
AT90USB82	4K語 (8Kバイト)	64語	PC5~0	64	PC11~6	11
AT90USB162	8K語 (16Kバイト)			128	PC12~6	12

表25-12. 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
AT90USB82	512バイト	4バイト	EEA1~0	128	EEA8~2	8
AT90USB162	512バイト			128	EEA8~2	8

25.6. 並列プログラミング手順

25.6.1. 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。
- ③ 160頁の表25-8.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。

25.6.2. 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。

25.6.3. 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。

25.6.4. フラッシュ メモリ書き込み (図25-3.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(161頁の表25-11.参照)。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25-2.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~\$0F/\$1F)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

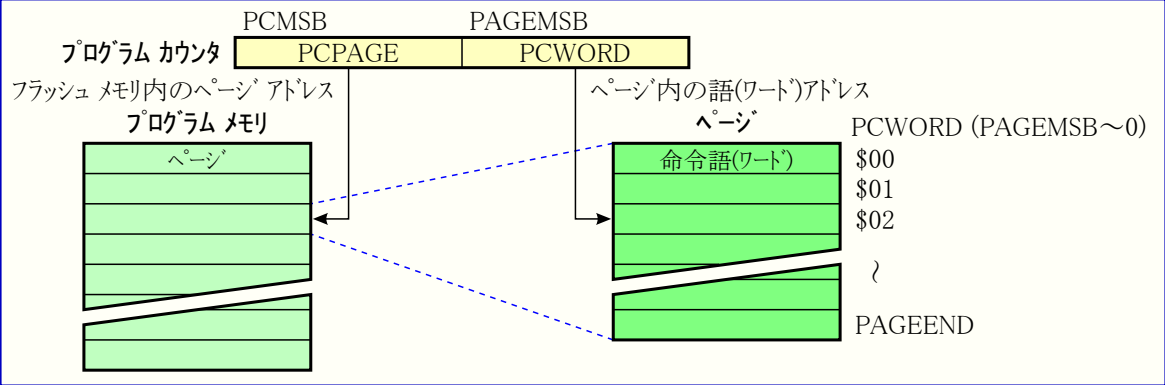
- ① BS1をLow(0)に設定します。
- ② WRに負パルスを与えます。これはデータのページ全体の書き込みを開始します。RDY/BSYがLow(0)になります。
- ③ RDY/BSYがHigh(1)になるまで待ちます。

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

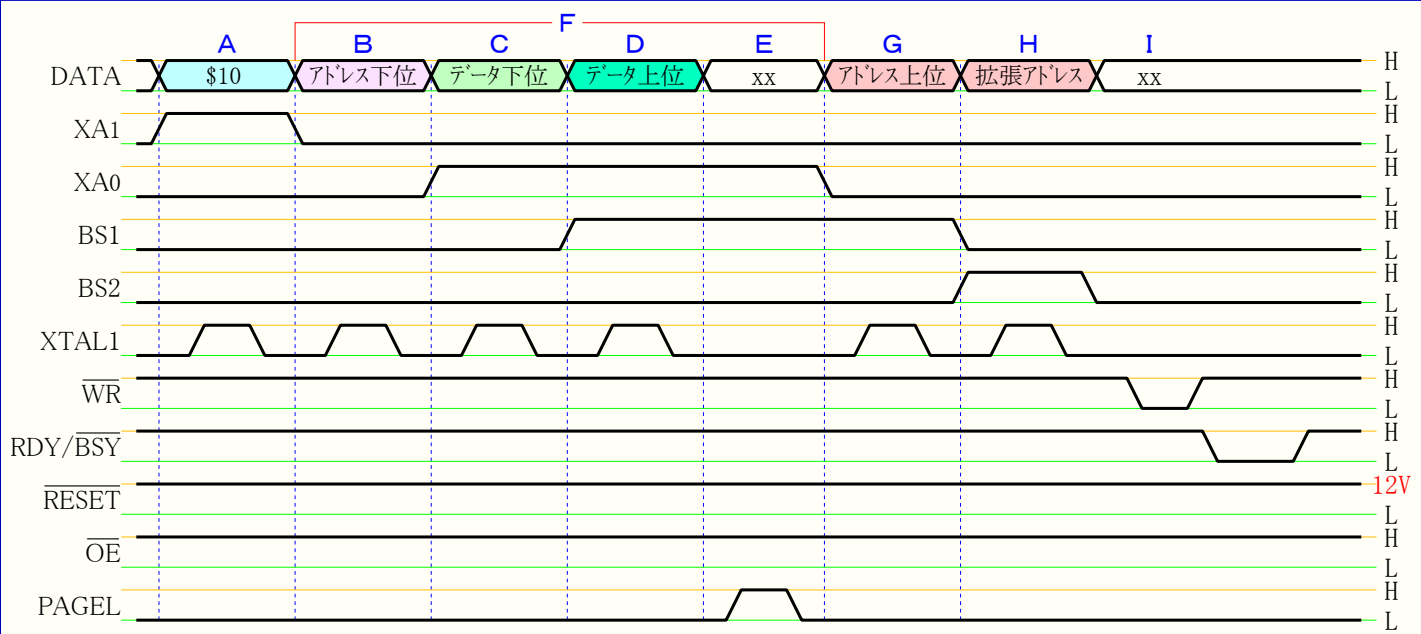
- ① XA1をHigh(1)、XA0をLow(0)に設定します。これは指令設定を許可します。
- ② DATAを\$00(0000 0000)にします。これは無操作指令です。
- ③ XTAL1に正パルスを与えます。これは無操作指令を設定し、内部書き込み信号がリセットされます。

図25-2. 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161頁の表25-11.で一覧されます。

図25-3. フラッシュメモリ書き込みタイミング



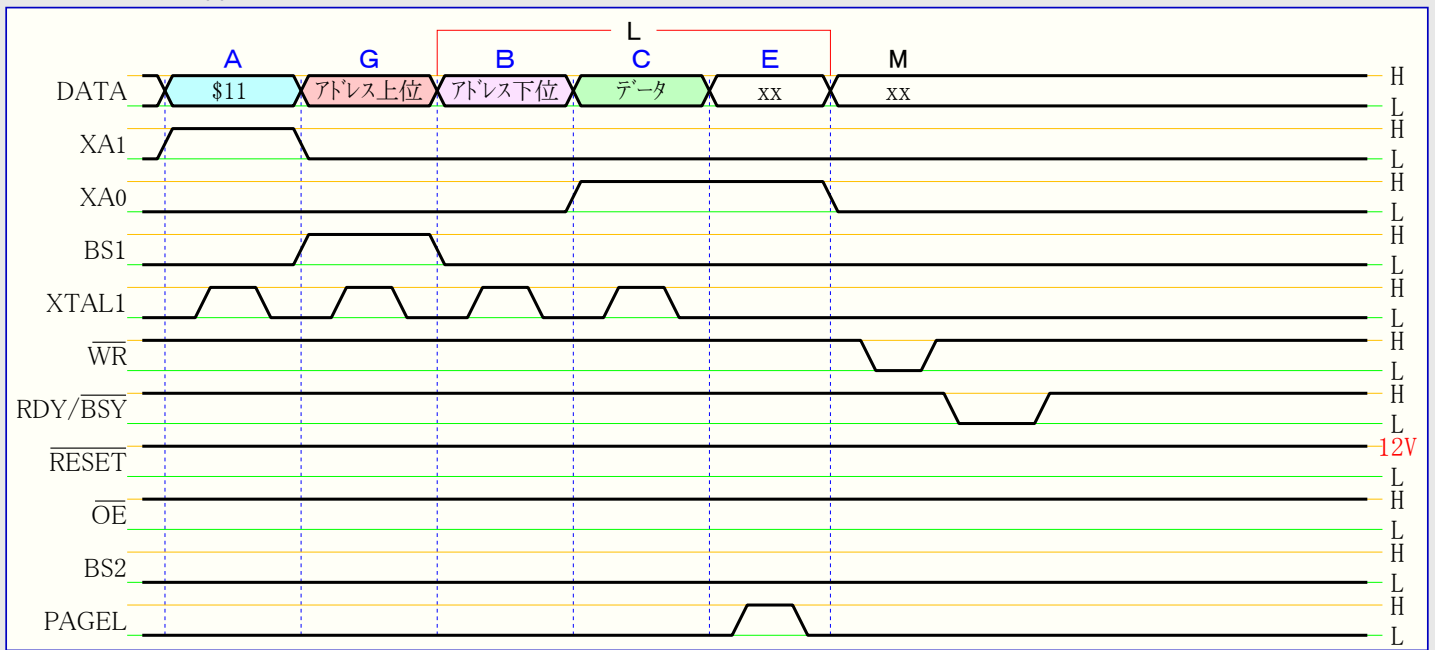
注: xx値は無関係です。A~Iは前記プログラミングを参照してください。

25.6.5. EEPROM書き込み

EEPROMはページで構成されます(161頁の表25-12参照)。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162頁の「フラッシュメモリの書き込み」を参照。図25-4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/\$01)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 WRに負パルスを与えます。これはEEPROMページ書き込みを開始します。RDY/BSYがLow(0)になります。
 - ③ 次のページを書く前に、RDY/BSYがHigh(1)になるまで待ちます。

図25-4. EEPROM書き込みタイミング



25.6.6. 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162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~\$0F/\$1F)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)、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. OEをHigh(1)に設定します。DATAはHi-Zになります。

25.6.7. 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162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/\$01)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)、OEをLow(0)に設定します。EEPROMのバイトデータが直ぐにDATAで読めます。
5. OEをHigh(1)に設定します。DATAはHi-Zになります。

25.6.8. 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。)

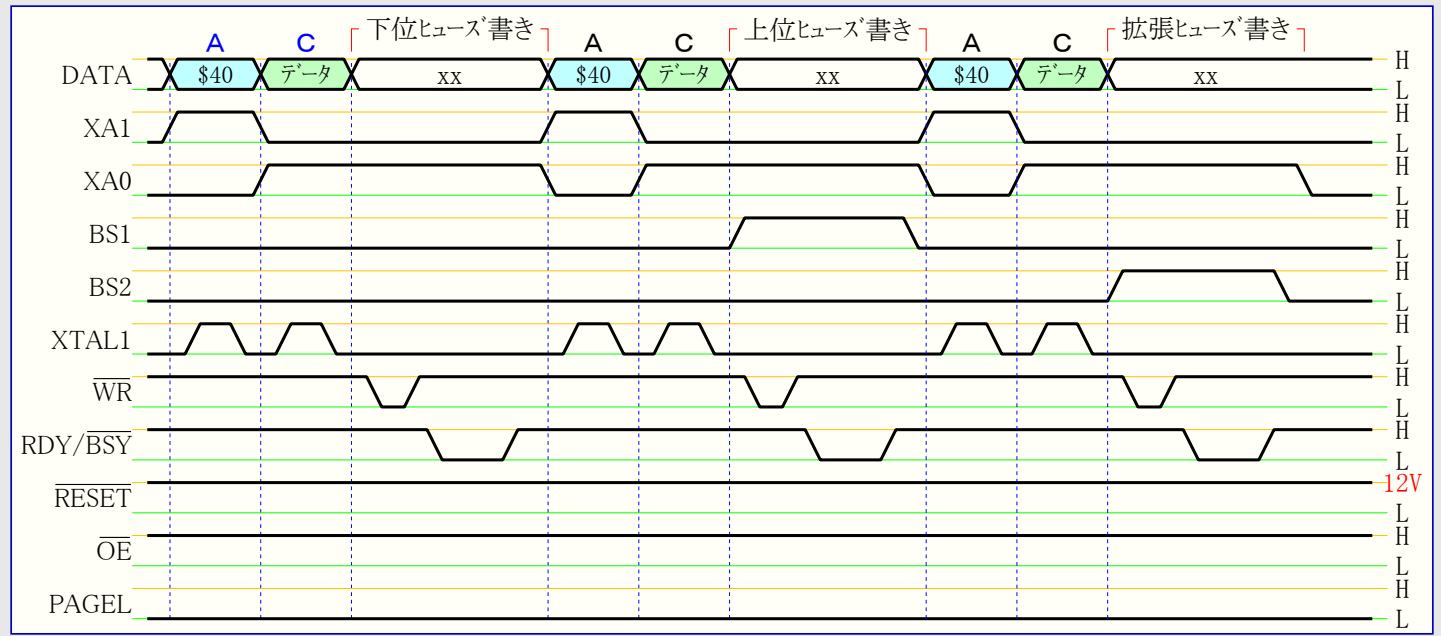
各ヒューズビットの書き込み方法は次のとおりです。(指令とデータ設定の詳細については162頁の「フラッシュメモリの書き込み」を参照)

1. ヒューズビット書き込み指令\$40(0100 0000)を設定します。(「フラッシュメモリの書き込み」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。(「フラッシュメモリの書き込み」のCを参照)
3. BS1とBS2を右表の目的バイトに対応する設定にします。
4. WRに負パルスを与え、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. ヒューズビット対応BS1,BS2設定

ヒューズビット	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25-5. ヒューズ書き込みタイミング



25.6.9. 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162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. WRに負パルスを与え、RDY/BSYがHighになるまで待ちます。

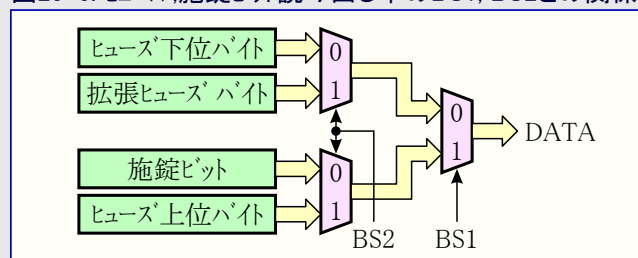
施錠ビットはチップ消去の実行によってのみ解除(1)できます。

25.6.10. 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162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)、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)、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)、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)、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. OEをHigh(1)に設定します。DATAはHi-Zになります。

図25-6. ヒューズ、施錠ビット読み出し中のBS1、BS2との関係



25.6.11. 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162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)、 $\overline{OE}$ をLow(0)に設定します。選んだ識票バイトが直ぐにDATAで読めます。
4. $\overline{OE}$ をHigh(1)に設定します。DATAはHi-Zになります。

25.6.12. 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162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)、 $\overline{OE}$ をLow(0)に設定します。校正バイトが直ぐにDATAで読めます。
4. $\overline{OE}$ をHigh(1)に設定します。DATAはHi-Zになります。

25.6.13. 並列プログラミング特性

図25-7. 並列プログラミング タイミング (一般的な必要条件)

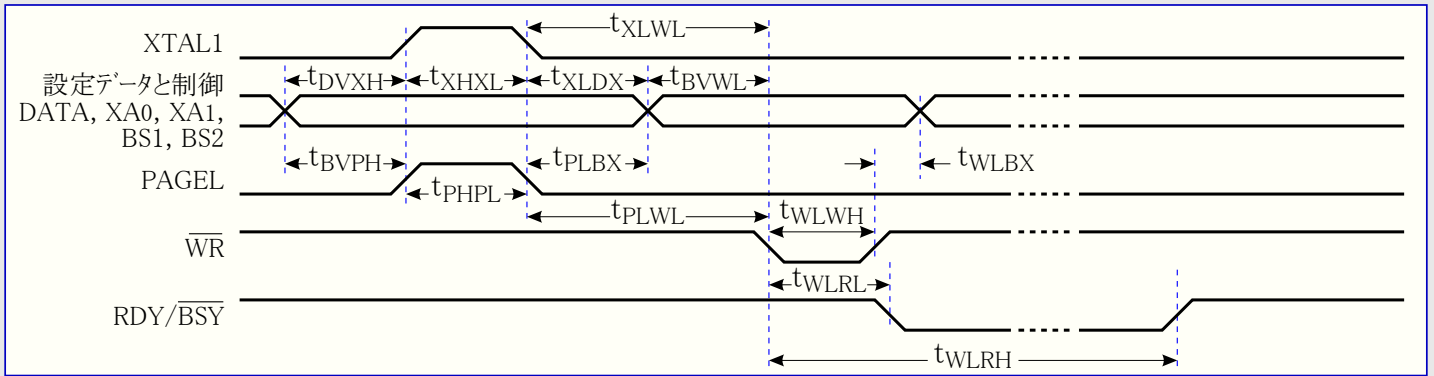
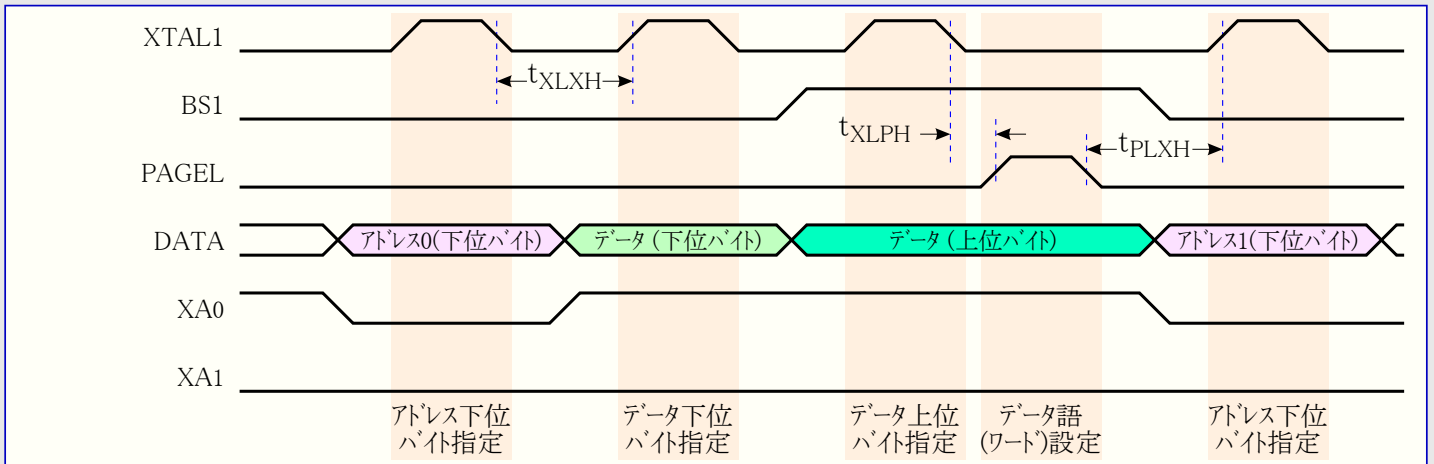
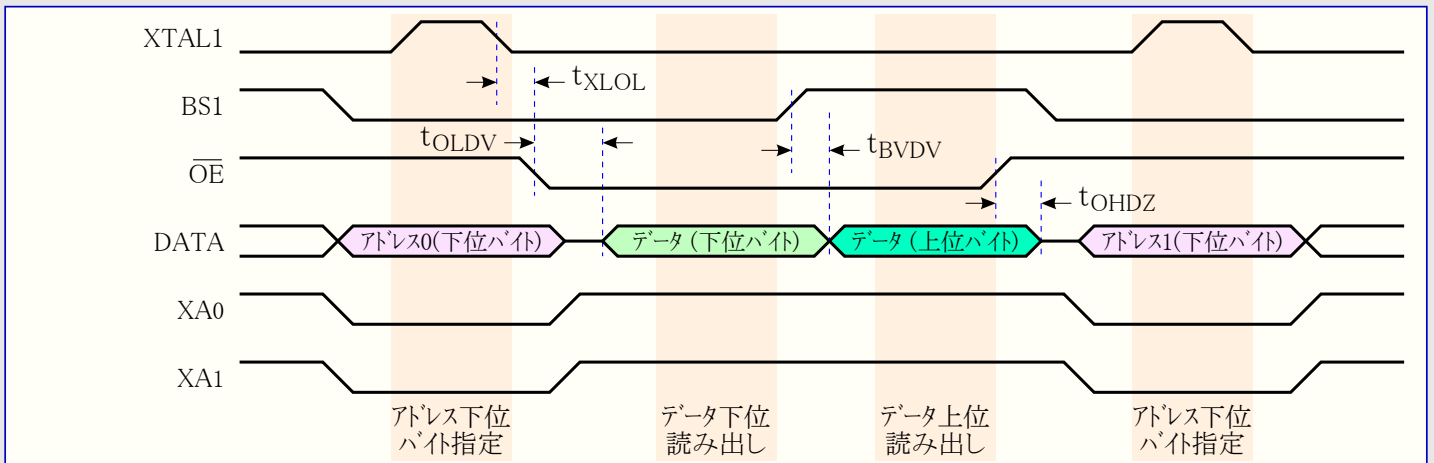


図25-8. 並列プログラミング タイミング (ページ設定での必要条件)



注: 図25-7.で示されるタイミング必要条件(即ち、 t_{DVXH} 、 t_{XHXL} 、 t_{XLDX})は設定操作にも適用されます。

図25-9. 並列プログラミング タイミング (同一ページ読み出しでの必要条件)



注: 図25-7.で示されるタイミング必要条件(即ち、 t_{DVXH} 、 t_{XHXL} 、 t_{XLDX})は読み出し操作にも適用されます。

表25-13. 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
VPP	プログラミング許可電圧	11.5		12.5	V
IPP	プログラミング許可電流			250	μA
tDVXH	XTAL1 ↑ に対するデータと制御の準備時間	67			ns
tXLXH	XTAL1 ↓ から次XTAL1 ↑ までの待機時間	200			
tXHXL	XTAL1 Highパルス幅	150			
tXLDX	XTAL1パルス ↓ 後のデータと制御の保持時間	67			
tXLWL	XTAL1パルス ↓ 後のWR ↓ 待機時間	0			
tXLPH	XTAL1パルス ↓ 後のPAGELパルス ↑ 待機時間	0			
tPLXH	PAGELパルス ↓ 後のXTAL1パルス ↑ 待機時間	150			
tBVPH	PAGELパルス ↑ に対するBS1準備時間	67			
tPHPL	PAGEL Highパルス幅	150			
tPLBX	PAGELパルス ↓ 後のBS1保持時間	67			
tWLBX	WRパルス ↓ 後のBS1,BS2保持時間	67			
tPLWL	PAGELパルス ↓ 後のWRパルス ↓ 待機時間	67			
tBVWL	WRパルス ↓ に対するBS1準備時間	67			
tWLWH	WR Lowパルス幅	150			
tWLRL	WRパルス ↓ 後のRDY/BSY ↓ 遅延時間	0		1	μs
tWLRH	書き込み時間 (WR ↓ からRDY/BSY ↑) (注1)	3.7		4.5	ms
tWLRH_CE	チップ消去時間 (WR ↓ からRDY/BSY ↑) (注2)	7.5		9	
tXLCL	XTAL1パルス ↓ 後のOE ↓ 待機時間	0			ns
tBVDV	BS1有効からのDATA遅延時間	0		250	
tOLDV	OE ↓ 後のDATA出力遅延時間			250	
tOHDZ	OE ↑ 後のDATA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。

25.7. 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25-14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。

25.7.1. 直列プログラミング用ピン配置

表25-14. 直列プログラミング用ピン配置

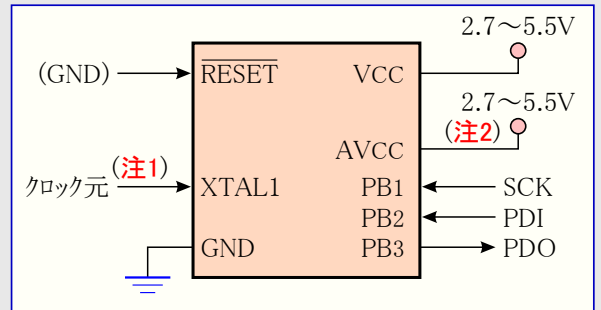
信号名	ピン名	入出力	機能
SCK	PB1	入力	直列クロック
PDI/MOSI	PB2	入力	直列データ入力
PDO/MISO	PB3	出力	直列データ出力

EEPROMをプログラミングする時に自動消去周期が自動書き込み動作内に組み入れられ(直列プログラミングのみ)、**チップ消去命令**を初めに実行する必要があります。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25-10. 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。

25.8. 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。)

AT90USB82/162に直列データを書く時にデータはSCKの上昇端で行われ、AT90USB82/162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25-11をご覧ください。

直列プログラミング動作で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25-16を参照)。

1. 電源投入手順:

RESETとSCKがLow(0)に設定されている間にVCCとGND間へ電源を印加します。いくつかのシステムで電源投入中、SCKがLowに保持されるのを書き込み器が保証できません。この場合、SCKがLow(0)に設定されてしまった後、RESETは最低2 CPUクロック周期幅の正パルスを与えられなければなりません。

2. 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。

3. 通信が同期を外していると、直列プログラミング命令は動作しません。同期していると、プログラミング許可命令の第3バイト送出時に第2バイト(\$53)が戻ります。この戻りが正しいかどうかによらず、命令の4バイト全てが送信されなければなりません。\$53が戻らない場合、RESETに正パルスを与え、新規プログラミング許可命令を行ってください。

4. フラッシュメモリは1ページ単位で書かれます。ページ容量は161頁の表25-11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ビットとデータを供給することによって1バイトずつ設定されます。ページが正しく設定されるのを保証するため、与えられたアドレス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6/7ビット(PC11/12~6)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25-15参照)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。

5. 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供給することによって1バイト単位で書かれます。EEPROMのメモリ位置は新規データが書かれるのに先立って始めて自動的に消去されます。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25-15参照)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ビットとデータを供給することによって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25-12参照)を行う前に最低tWD_EEPROM(表25-15参照)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。

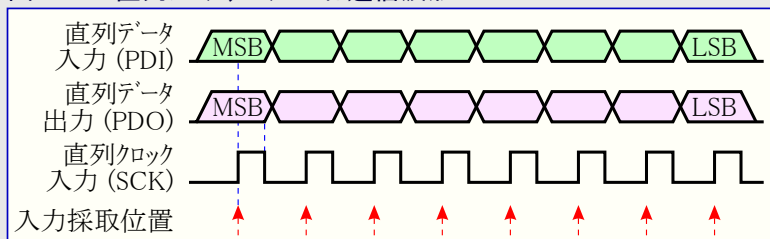
6. どのメモリ位置も選んだアドレスの内容を直列出力PDOに読み戻す**読み出し命令**を使うことで照合できます。フラッシュメモリ読み出し時、フラッシュメモリ読み出し命令内に含まれない上位アドレスバイト定義に拡張アドレスバイト設定命令を使用してください。拡張アドレスバイトはこの命令が再実行されるまで保存されます。換言すると、本命令は初回ページと64K語(ワード)境界を過ぎる時にだけ実行される必要があります。

7. プログラミング作業終了時、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25-15. 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25-11. 直列プログラミングバイト通信波形



25.8.1. 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25-16.と次頁の図25-12.は命令一式を記述します。

表25-16. 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。

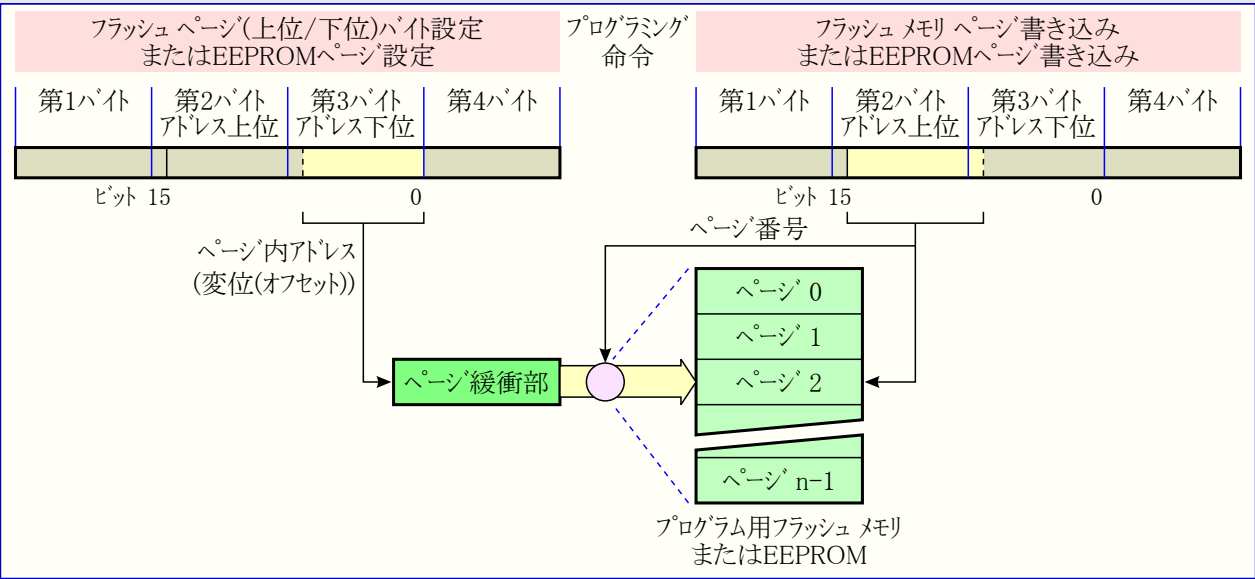
- ・ ヒューズ、施錠ビット、識票バイト、校正バイト、ページ容量については対応項目を参照してください。
- ・ プログラミングと書き込み器に関する応用記述については <http://www.atmel.com/avr> をご覧ください。
- ・ 第4バイトの赤背景はホスト読み込み(デバイス出力)を示します。

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25-12をご覧ください。

図25-12. 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。AT90USB82/162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	AT90USB82/162 : L=PC5~0
EEPROMページ内バイト設定	0000 0000	0000 00LL	AT90USB82/162 : L=EEA1~0
フラッシュメモリ読み出し	0000 HHHH 000H HHHH	LLLL LLLL	AT90USB82 : H=PC11~8, L=PC7~0 AT90USB162 : H=PC12~8, L=PC7~0
EEPROM読み出し	0000 000H	LLLL LLLL	AT90USB82/162 : H=EEA8, L=EEA7~0
フラッシュページ書き込み	0000 HHHH 000H HHHH	LL00 0000	AT90USB82 : H=PC11~8, L=PC7 AT90USB162 : H=PC12~8, L=PC7
EEPROMバイト書き込み	0000 000H	LLLL LLLL	AT90USB82/162 : H=EEA8, L=EEA7~0
EEPROMページ書き込み	0000 000H	LLLL LL00	AT90USB82/162 : H=EEA8, L=EEA7~2

25.8.2. 直列プログラミング特性

SPI部の特性については174頁の「SPIタイミング特性」を参照してください。

26. 電気的特性

(重要注意) 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。これらの値は設計目標を示す暫定値で、実デバイスの特性付け後に更新されます。

26.1. 絶対最大定格 (警告)

動作温度	-55°C ~ +125°C
保存温度	-65°C ~ +150°C
RESET, UVCC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
UVCC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。

26.2. 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
VIL	標準I/Oピン(注5)Lowレベル入力電圧	VCC=2.7~5.5V	-0.5		0.2VCC (注1)	V
VIL1	XTAL1ピン Lowレベル入力電圧	VCC=2.7~5.5V	-0.5		0.1VCC (注1)	
VIL2	RESETピン Lowレベル入力電圧	VCC=2.7~5.5V	-0.5		0.1VCC (注1)	
VIL3	MOSI/MISOピン Lowレベル入力電圧	VCC=2.7~5.5V	-0.5		0.8	
VIL4	PS2ピン Lowレベル入力電圧	VCC=4.5~5.5V	-0.5		0.7	
VIH	標準I/Oピン(注5)Highレベル入力電圧	VCC=2.7~5.5V	0.6VCC (注2)		VCC+0.5	
VIH1	XTAL1ピン Highレベル入力電圧	VCC=2.7~5.5V	0.7VCC (注2)		VCC+0.5	
VIH2	RESETピン Highレベル入力電圧	VCC=2.7~5.5V	0.9VCC (注2)		VCC+0.5	
VIH3	MOSI/MISOピン Highレベル入力電圧	VCC=2.7~5.5V	2		VCC+0.5	
VIH4	PS2ピン Highレベル入力電圧	VCC=4.5~5.5V	2.4		VCC+0.5	
VOL	標準I/O(注5), MOSI/MISOピン Lレベル出力電圧 (注3)	IOL=10mA, VCC=5V IOL=5mA, VCC=3V			0.7 0.5	
VOL1	PS2ピン Lレベル出力電圧	IOL=5mA, VCC=4.5~5.5V			0.7	
VOH	標準I/O(注5), MOSI/MISOピン Hレベル出力電圧 (注4)	IOH=-20mA, VCC=5V IOH=-10mA, VCC=3V	4.2 2.3			
IIL	I/OピンLowレベル入力漏れ電流	VCC=5.5V			1	μA
IIH	I/OピンHighレベル入力漏れ電流	確実なH/L範囲			1	
RRST	RESETピン プルアップ抵抗		30		60	kΩ
RPU	I/Oピン プルアップ抵抗		20		50	
RPUDP	USB D+内部プルアップ抵抗		0.9		1.5	
RPUPS2	PS2 プルアップ抵抗	VCC=5V	3.2	4.7	6.5	
ICC	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, 8MHz		7	8.5	mA
		VCC=5V, 16MHz		13.5	25	
	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, 8MHz		3.5		
		VCC=5V, 16MHz		4	4.5	
	パワーダウン動作 消費電流	VCC=3V, WDT/電圧調整器禁止		5	10	μA
		VCC=3V, WDT許可, 電圧調整器禁止		10	15	
		VCC=5V, WDT/BOD/電圧調整器許可		41	65	
	スタンバイ動作 消費電流	VCC=3V, WDT/BOD許可, 電圧調整器禁止		70	100	
		VCC=5V, WDT/BOD/電圧調整器許可		120	150	
VACIO	アナログ比較器入力変位(オフセット)電圧	VCC=5V, Vin=VCC/2		<10	40	mV
IACLK	アナログ比較器入力漏れ電流		-50		50	nA
tACPD	アナログ比較器伝播遅延時間	VCC=2.7V		750		ns
		VCC=4.0V		500		
RUSB	外部USB直列抵抗			22±5%		Ω

次頁へ続く

(続き) TA=-40℃～85℃, 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
VREG	電圧調整器出力電圧	VCC ≥ 4.0V, I ≤ 100mA, C _{UCAP} = 1μF ± 20%	3.0	3.3	3.6	V
UVCC	UVCC入力電圧	VCC = 2.7～5.5V	4		5.5	

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:** 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。

1. ポートB7～0、C7～0、D7～0のIOLの合計が150mA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。

1. ポートB7～0、C7～0、D7～0のIOHの合計が150mA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: 標準I/OピンはXTAL1, MOSI, MISO, PS2, RESETピンを除きます。**注6:** USB電氣的特性節で詳述されるようにD+/D-パッドは22Ω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。

26.3. 外部クロック特性

図26-1. 外部クロック駆動波形

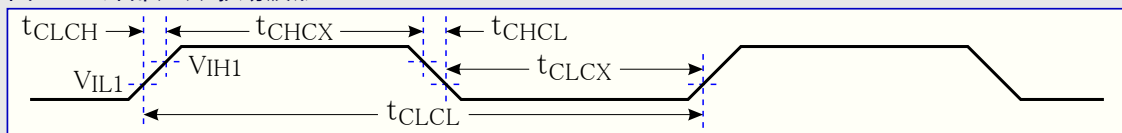


表26-1. 外部クロック特性

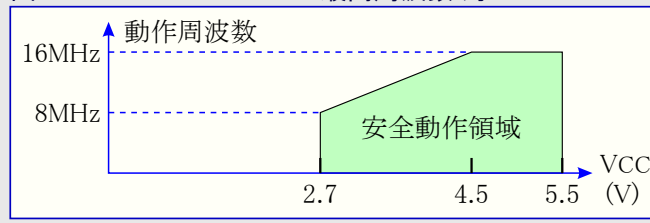
シンボル	項目	VCC=2.7～5.5V		VCC=4.5～5.5V		単位
		最小	最大	最小	最大	
1/tCLCL	クロック周波数	0	8	0	16	MHz
tCLCL	クロック周期	125		62.5		ns
tCHCX	Highレベル時間	50		25		
tCLCX	Lowレベル時間	50		25		
tCLCH	上昇時間		1.6		0.5	μs
tCHCL	下降時間		1.6		0.5	
ΔtCLCL	隣接クロック周期間の変化率		2		2	%

注: 詳細については22ページの「外部クロック信号」を参照してください。

26.4. 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26-2.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。

図26-2. AT90USB82/162の最高周波数 対 VCC



26.5. SPIタイミング特性

図26-3. SPI タイミング必要条件 (主装置動作)

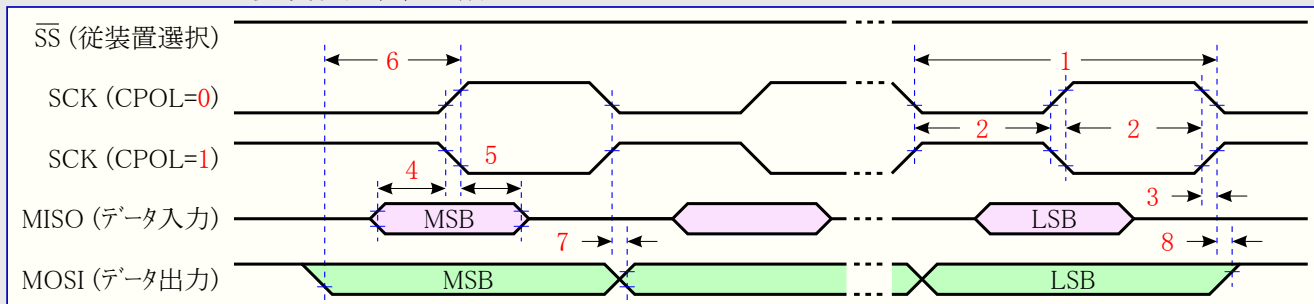


図26-4. SPI タイミング必要条件 (従装置動作)

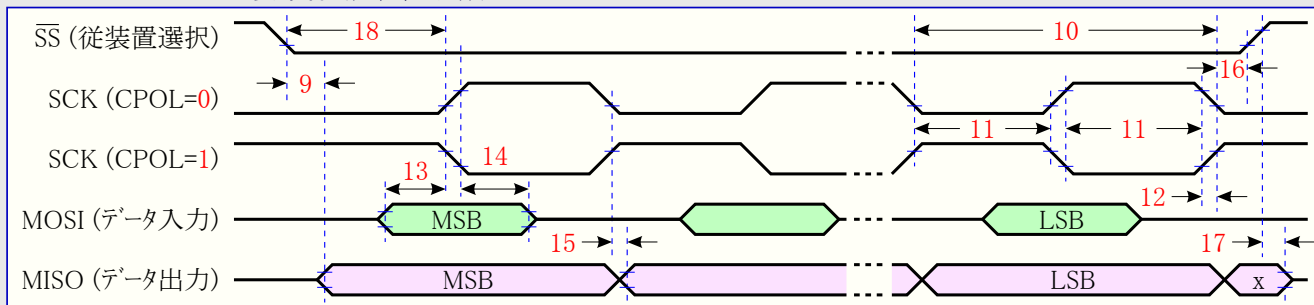


表26-2. SPI タイミング特性

番号	項目	動作種別	最小	代表	最大	単位
1	SCK周期	主装置		表16-4.参照		ns
2	SCK High/Low期間	主装置		50%デューティ比		
3	SCK上昇/下降時間	主装置		TBD		
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		
10	SCK周期	従装置	4t _{CK}			μs
11	SCK High/Low期間 (注)	従装置	2t _{CK}			
12	SCK上昇/下降時間	従装置		TBD		
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。

26.6. ハードウェア ブート受付タイミング特性

図26-5. ハードウェア ブート タイミング必要条件

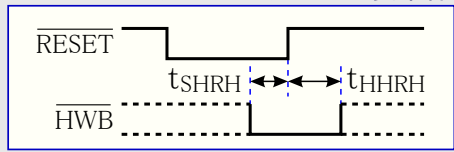


表26-3. ハードウェア ブート タイミング

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)		

26.7. 校正付き内蔵RC発振器の精度

表26-4. 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℃	±10%
使用者校正	7.3～8.1MHz	2.7～5.5V	-40～85℃	±1%

27. 代表特性 (暫定)

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。

パワーダウン動作での消費電力はクロック選択と無関係です。

消費電流は動作電圧、動作周波数、I/Oピンの負荷、I/Oピンの切り替え速度、命令実行、周囲温度のような様々な要素の関数です。支配的な要素は動作電圧と動作周波数です。

容量性負荷のピンの引き込み電流は(1つのピンに対して) $CL(\text{負荷容量}) \times V_{CC}(\text{動作電圧}) \times f(\text{I/Oピンの平均切り替え周波数})$ として推測できます。

デバイスは検査範囲よりも高い周波数特性を示します。デバイスは注文番号が示す周波数よりも高い周波数での機能特性を保証されません。

ウォッチドッグ タイマ許可のパワーダウン動作での消費電流とウォッチドッグ タイマ禁止のパワーダウン動作での消費電流間の違いは、ウォッチドッグ タイマによって引き込んだ(消費した)差電流を表します。

27.1. 入力電圧水準

図27-1. D+/D-,MOSI/MISOを除く全I/O入力L電圧 対 動作電圧

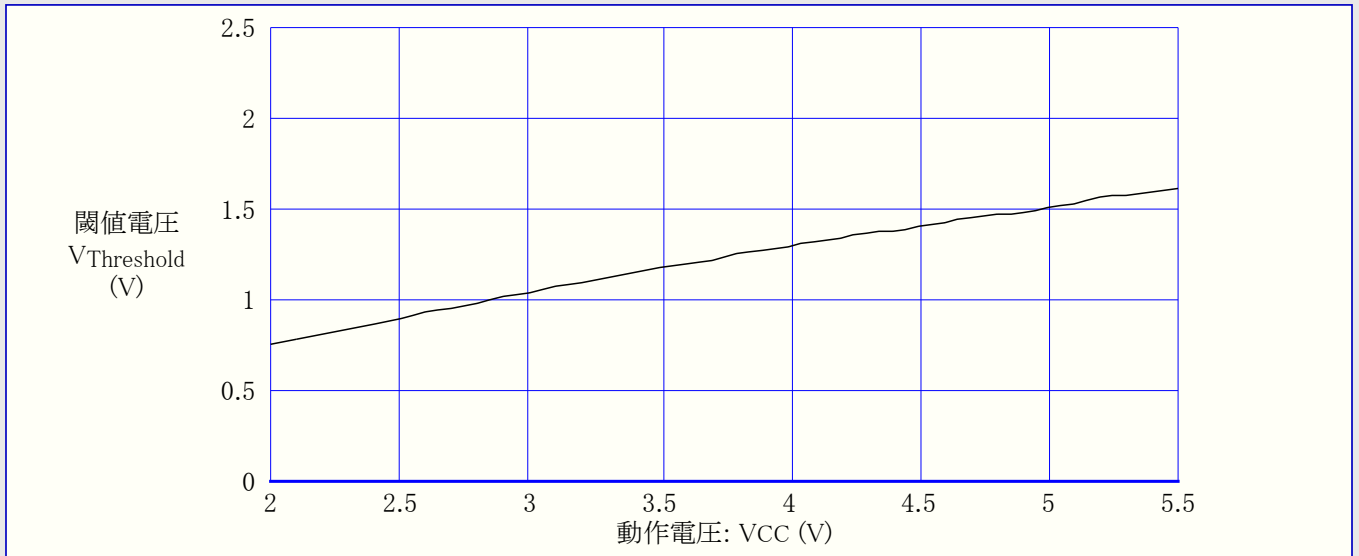


図27-2. D+/D-,MOSI/MISOを除く全I/O入力H電圧 対 動作電圧

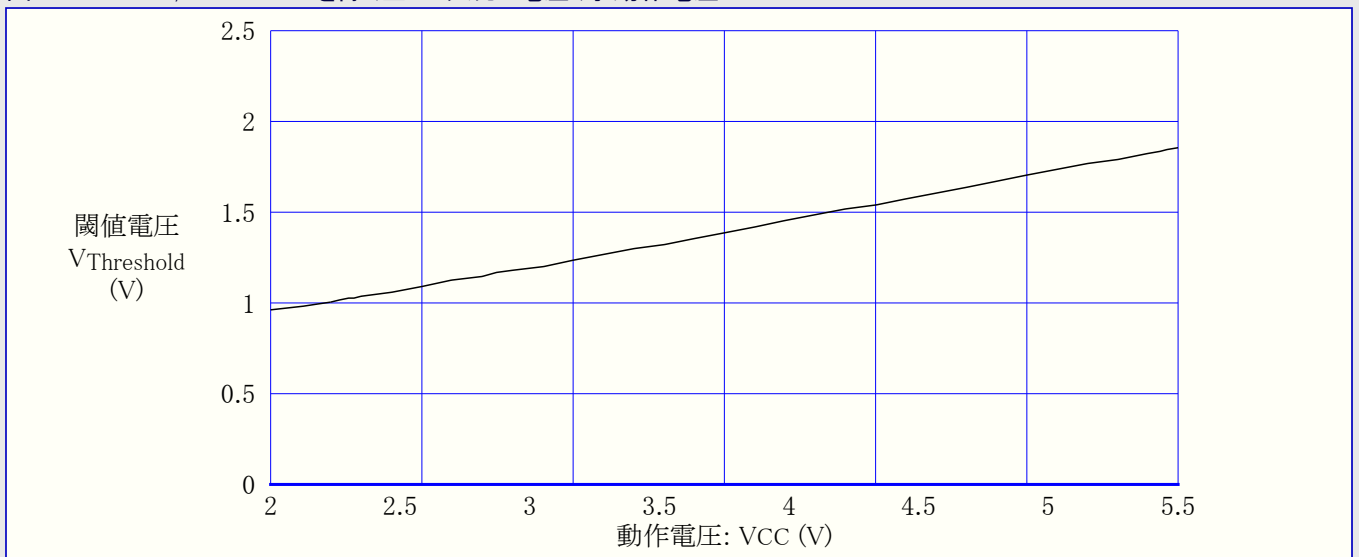


図27-3. MOSI/MISO入力L電圧 対 動作電圧

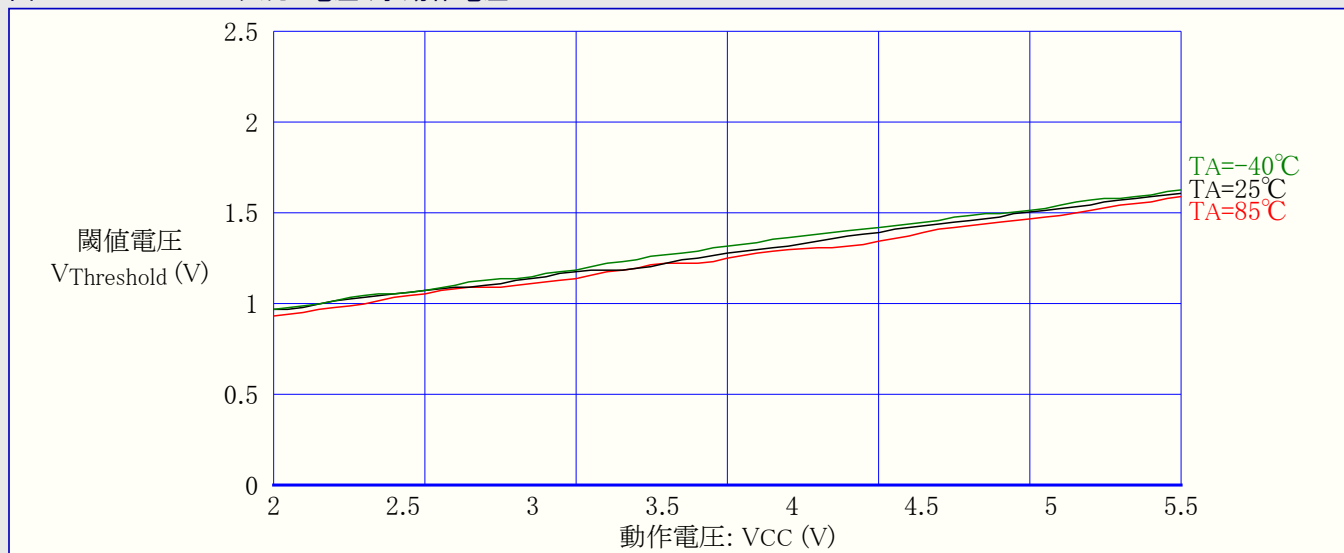
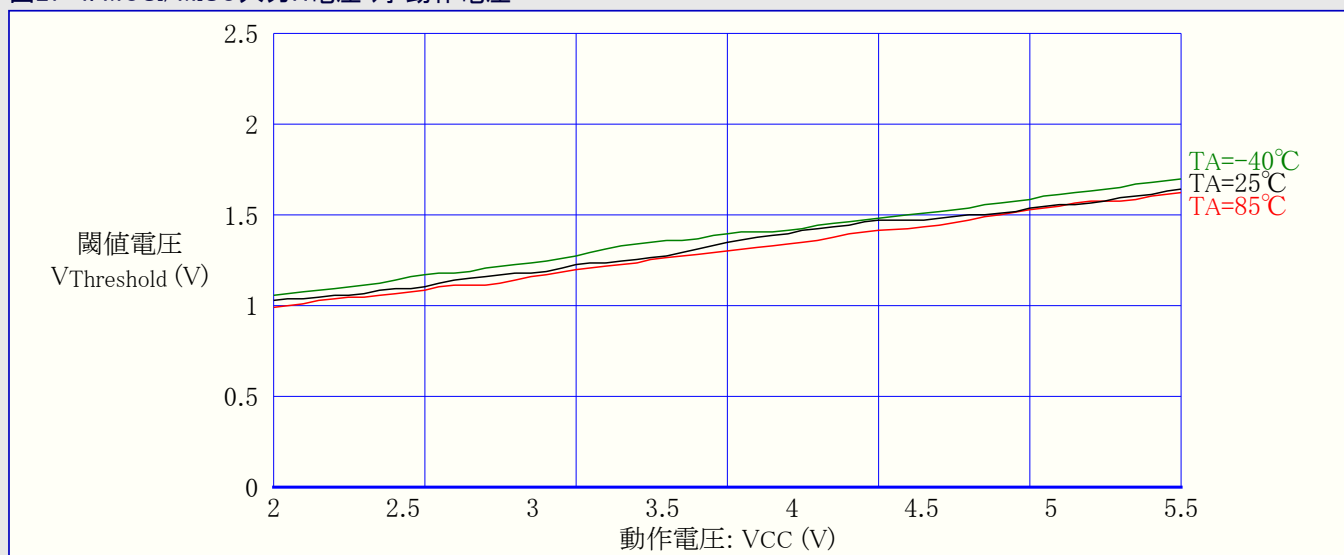


図27-4. MOSI/MISO入力H電圧 対 動作電圧



27.2. 出力電圧水準

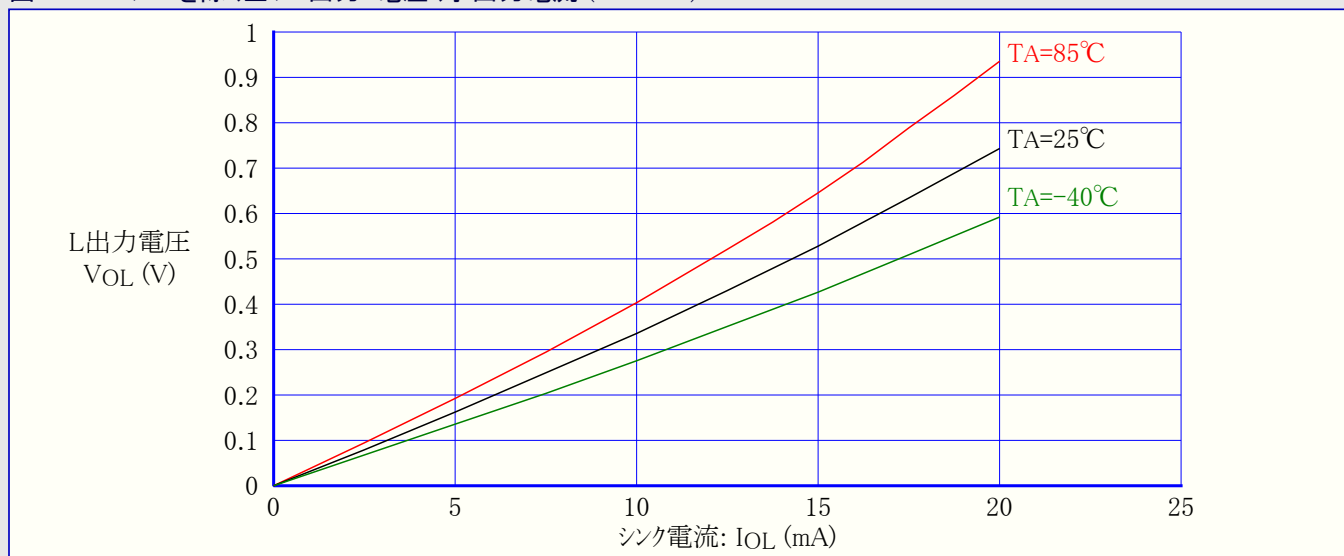
図27-5. D+/D-を除く全I/O出力L電圧 対 出力電流 ($V_{CC}=3\text{V}$)

図27-6. D+/D-を除く全I/O出力L電圧 対 出力電流 (VCC=5V)

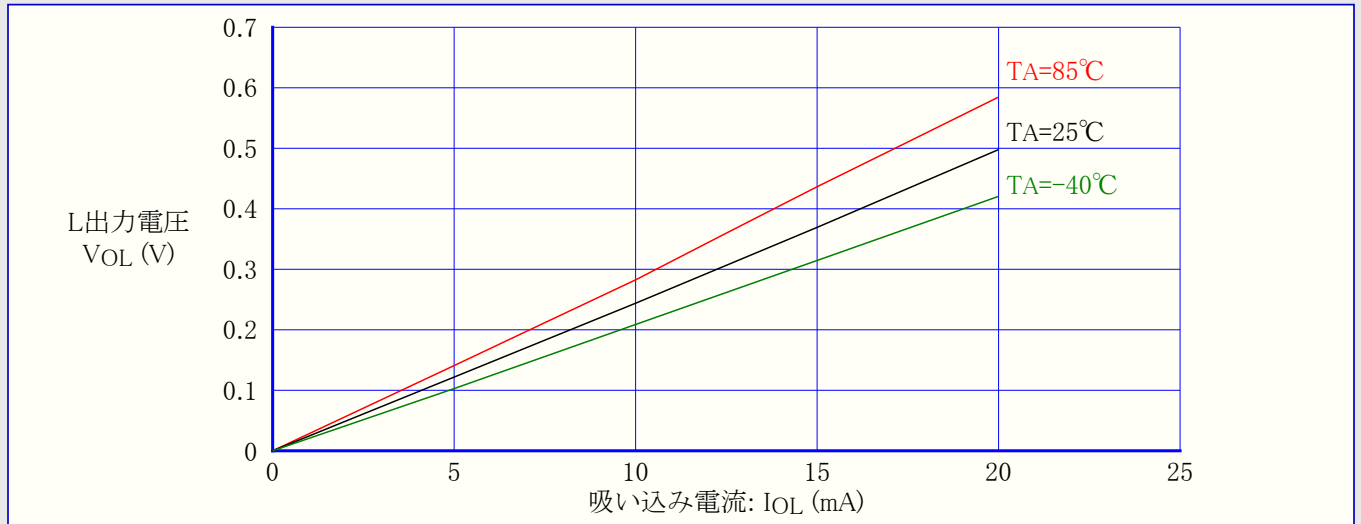


図27-7. PS2出力L電圧 対 出力電流 (VCC=5V)

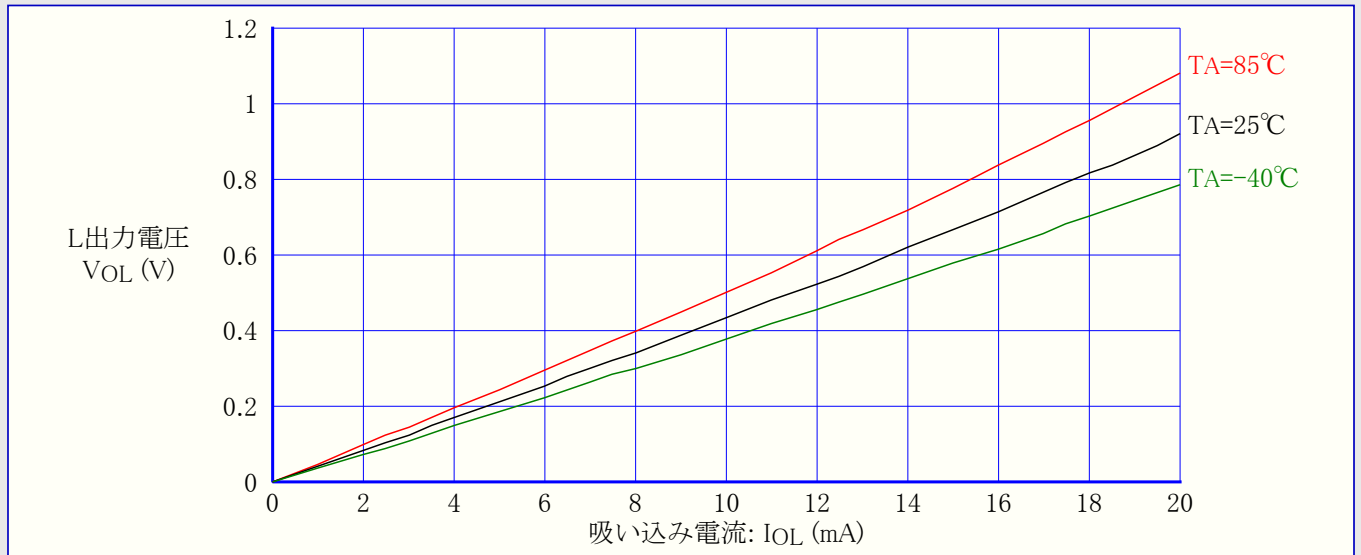


図27-8. D+/D-を除く全I/O出力H電圧 対 出力電流 (VCC=3V)

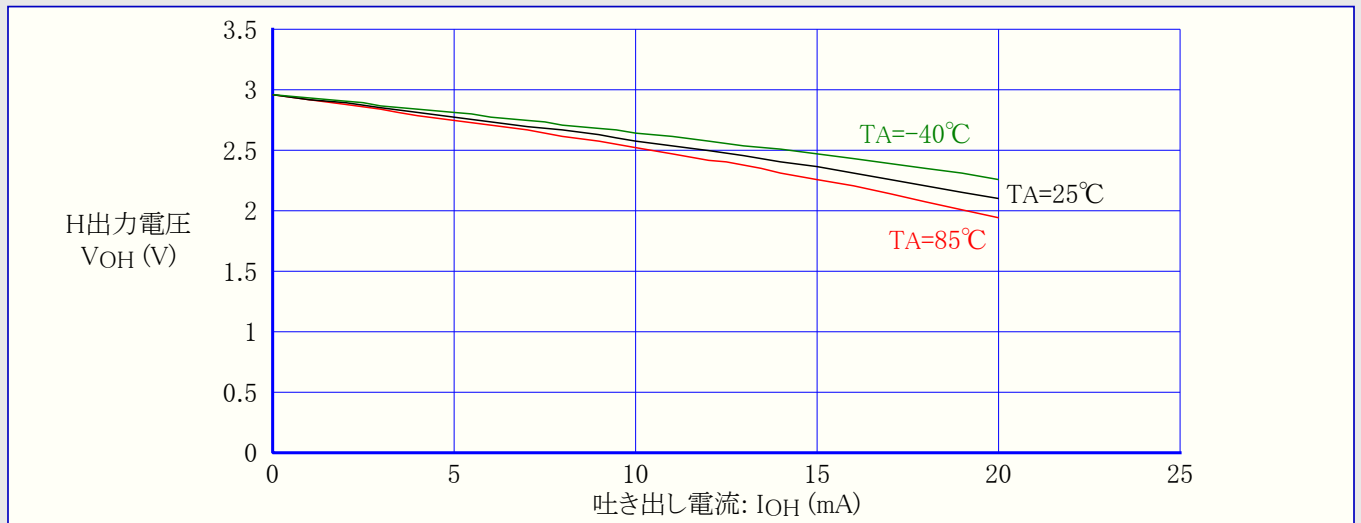
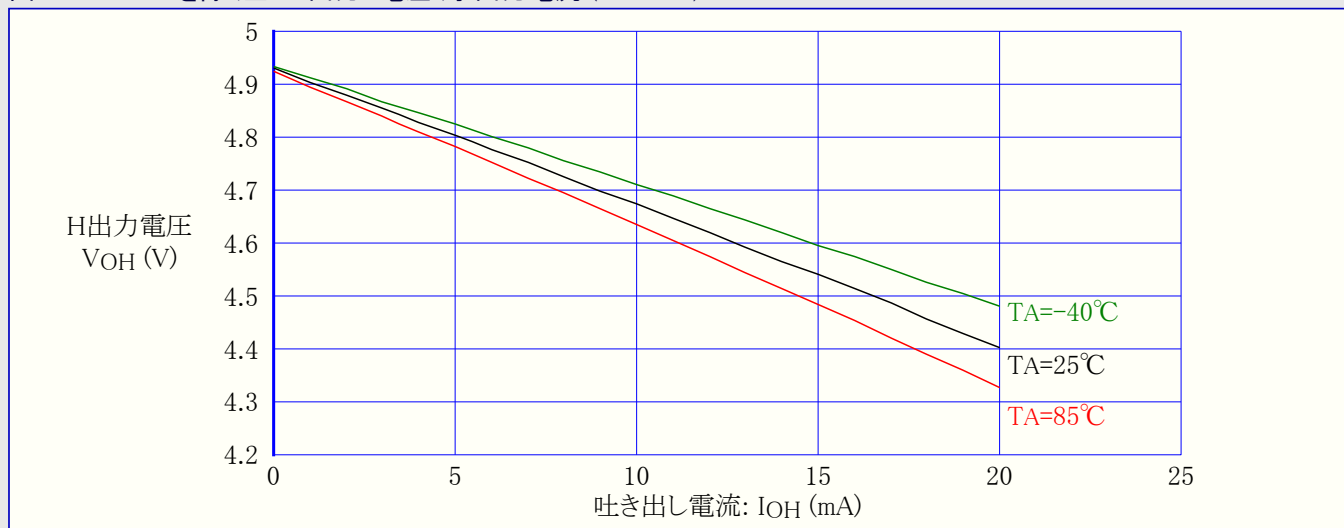
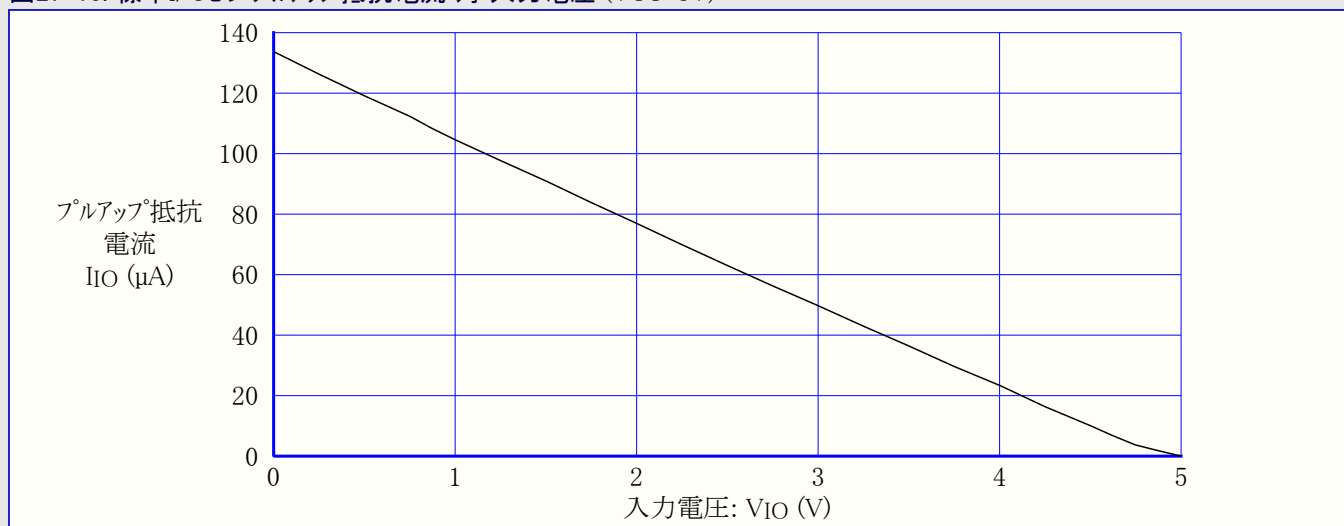


図27-9. D+/D-を除く全I/O出力H電圧 対 出力電流 (VCC=5V)



27.3. I/O プルアップ電流

図27-10. 標準I/Oピンプルアップ抵抗電流 対 入力電圧 (VCC=5V)



27.4. パワーダウン動作消費電流

図27-11. パワーダウン動作消費電流 対 動作電圧 (WDT禁止, BOD禁止, 電圧調整器禁止)

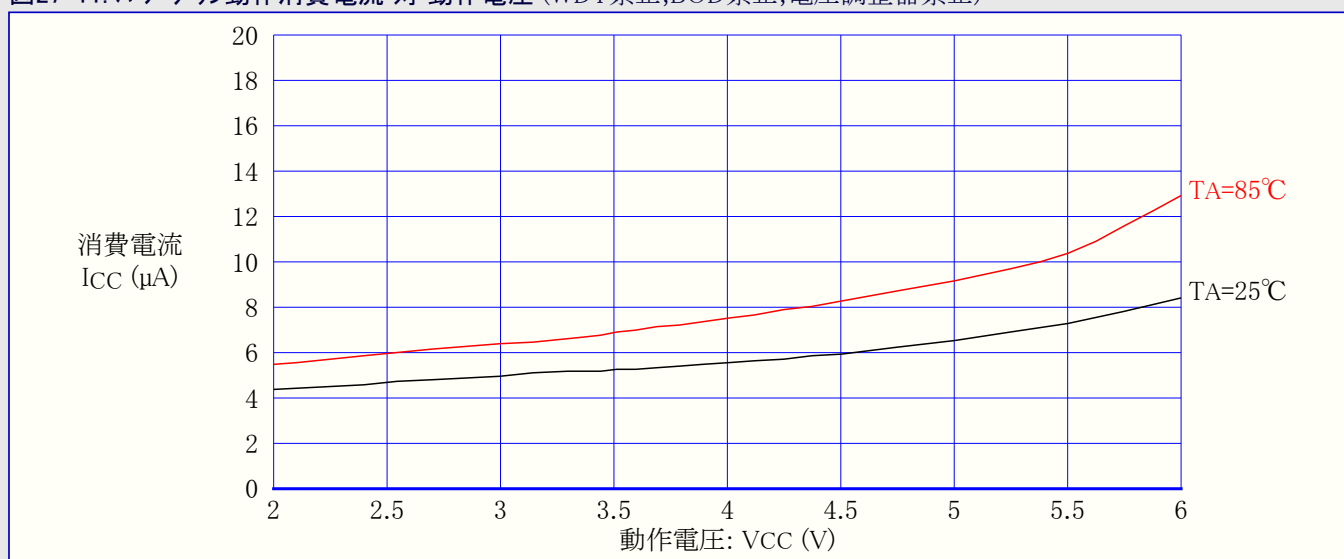


図27-12. パワーダウン動作消費電流 対 動作電圧 (WDT許可,BOD禁止,電圧調整器禁止)

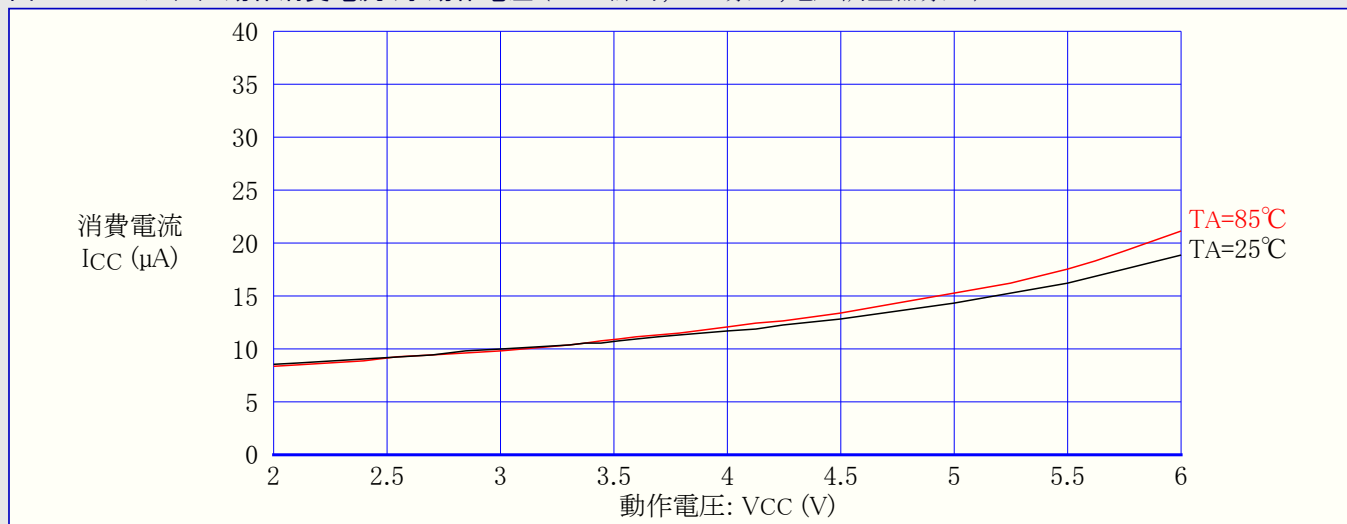


図27-13. パワーダウン動作消費電流 対 動作電圧 (WDT禁止,BOD許可,電圧調整器禁止)

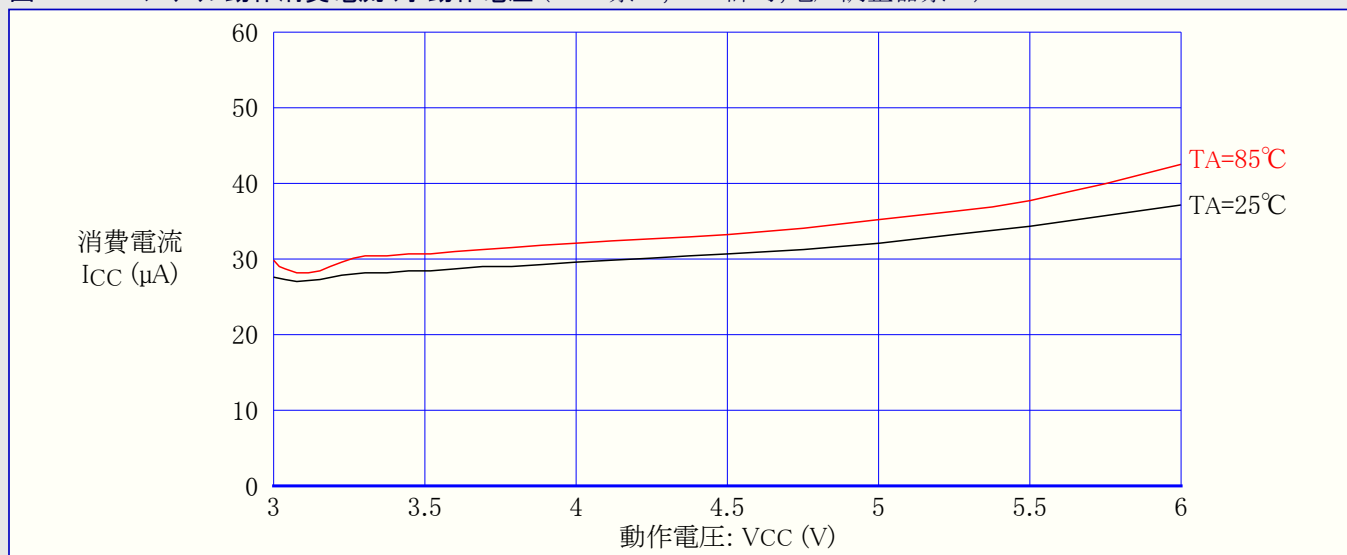


図27-14. パワーダウン動作消費電流 対 動作電圧 (WDT許可,BOD許可,電圧調整器禁止)

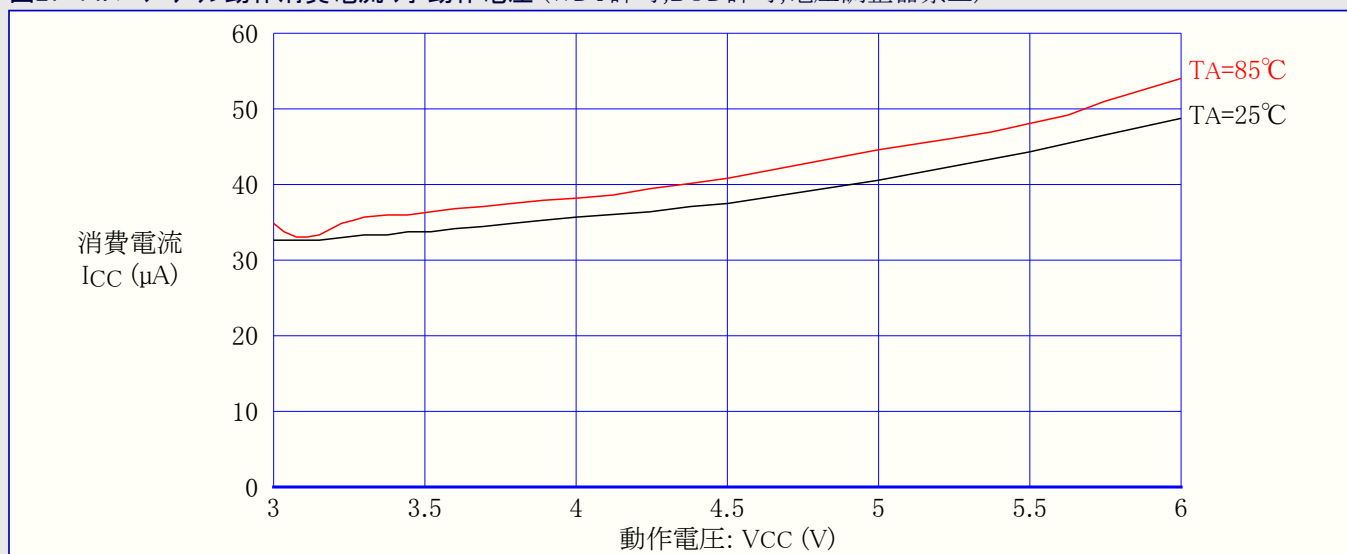
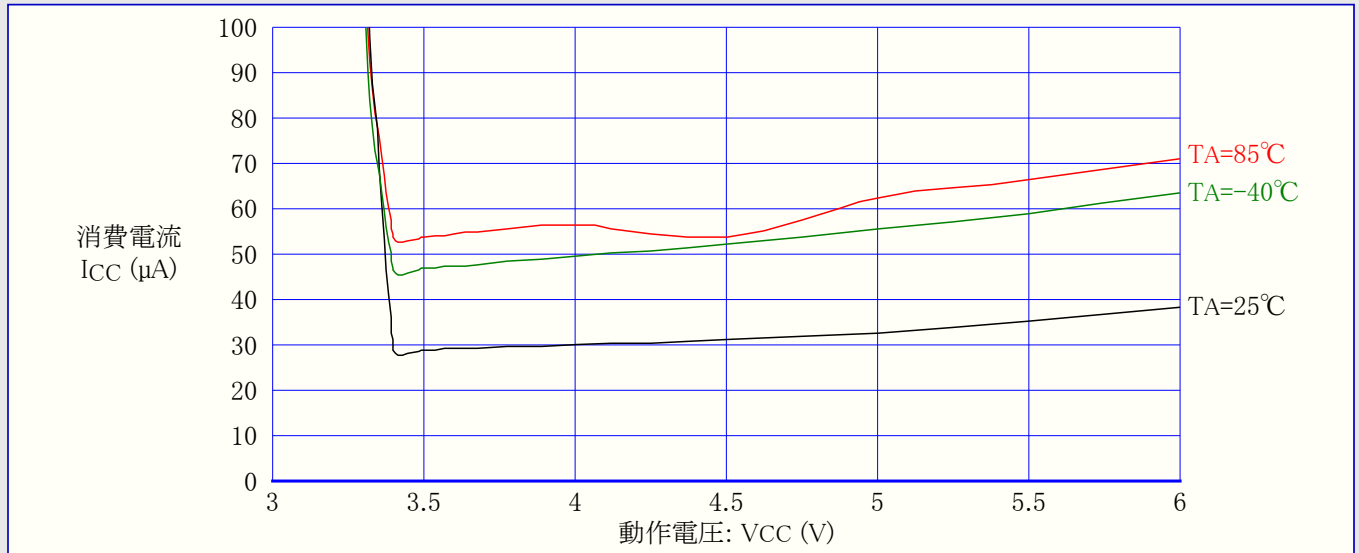
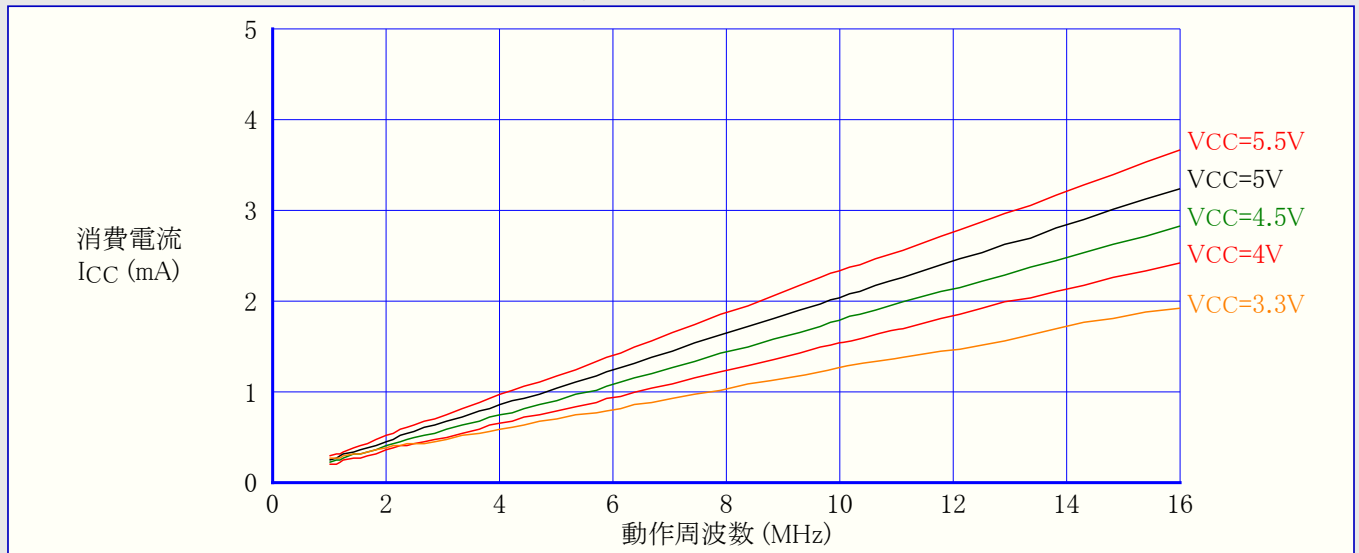


図27-15. パワーダウン動作消費電流 対 動作電圧 (WDT禁止,BOD許可,電圧調整器許可)



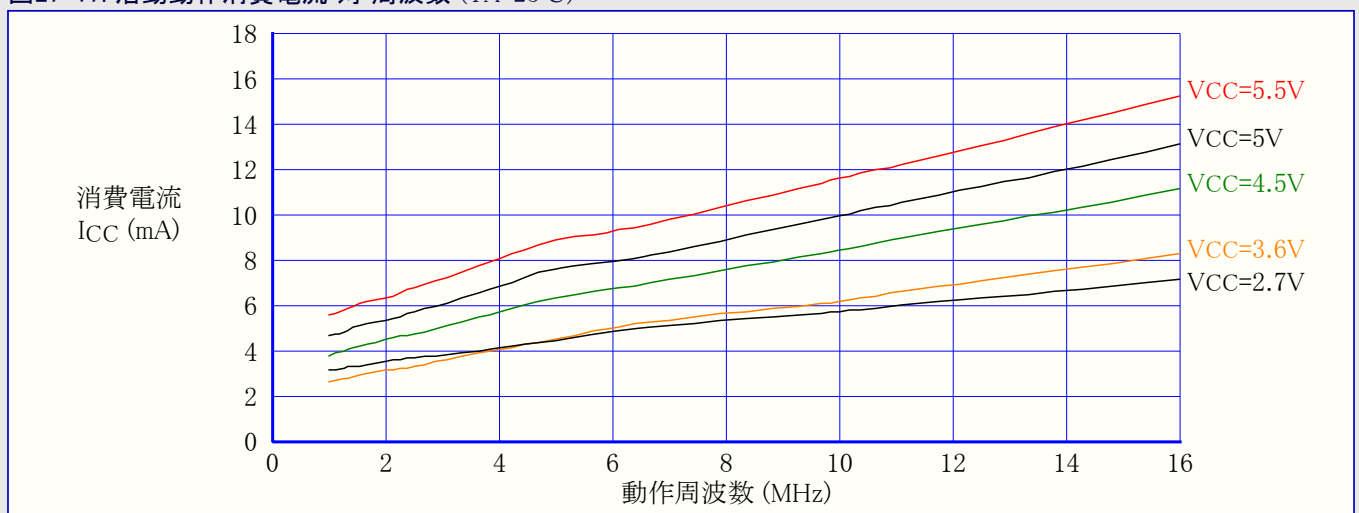
27.5. アイドル動作消費電流

図27-16. アイドル動作消費電流 対 周波数 (TA=25°C,4V以下電圧調整器禁止)



27.6. 活動動作消費電流

図27-17. 活動動作消費電流 対 周波数 (TA=25°C)



27.7. リセット消費電流

図27-18. リセット消費電流 対 周波数 (TA=25°C)

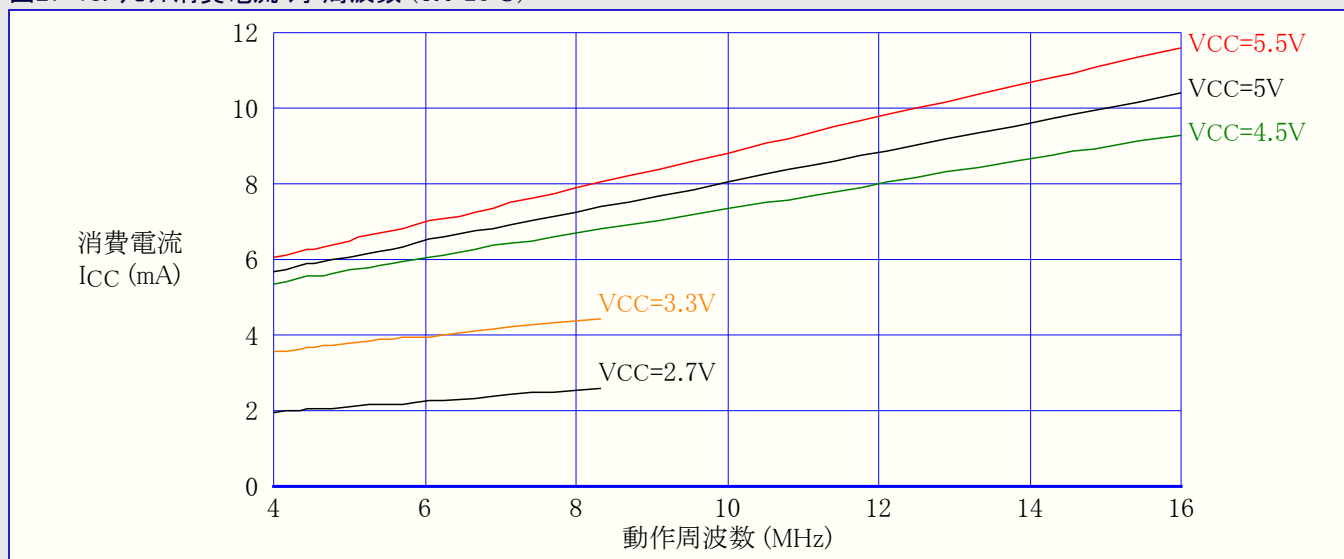
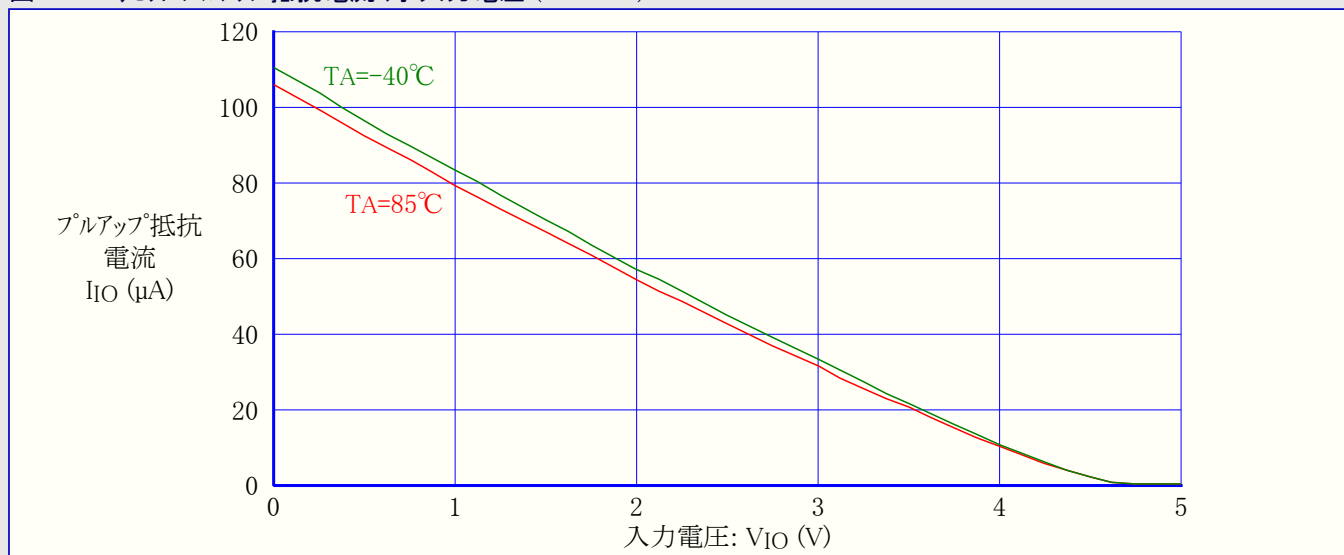
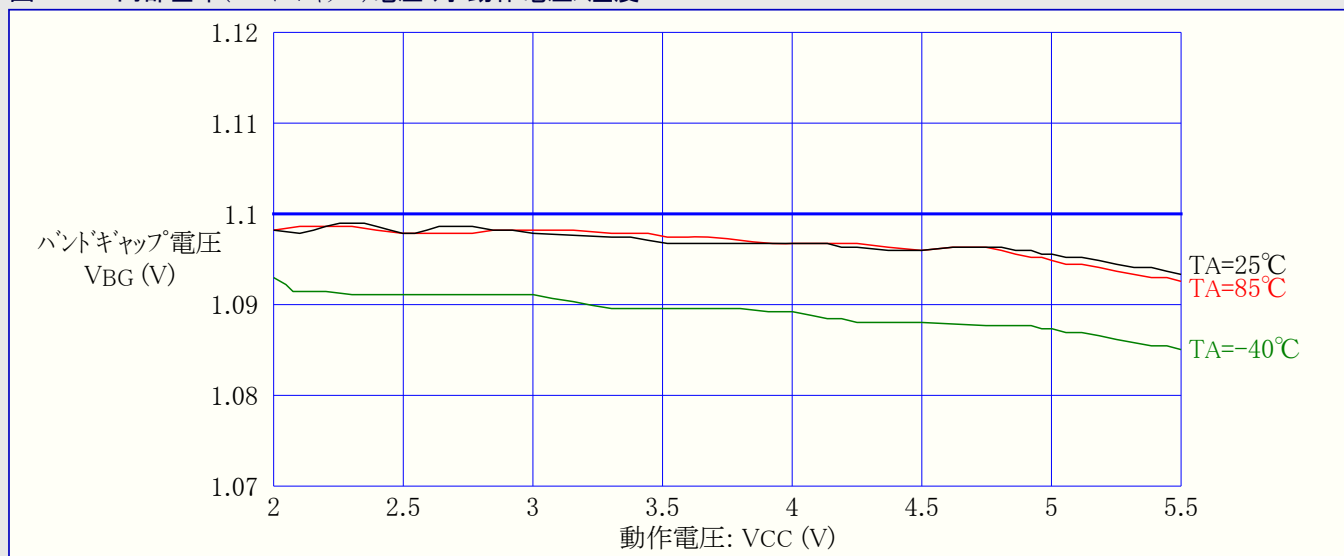


図27-19. リセットプルアップ抵抗電流 対 入力電圧 ($V_{CC}=5V$)



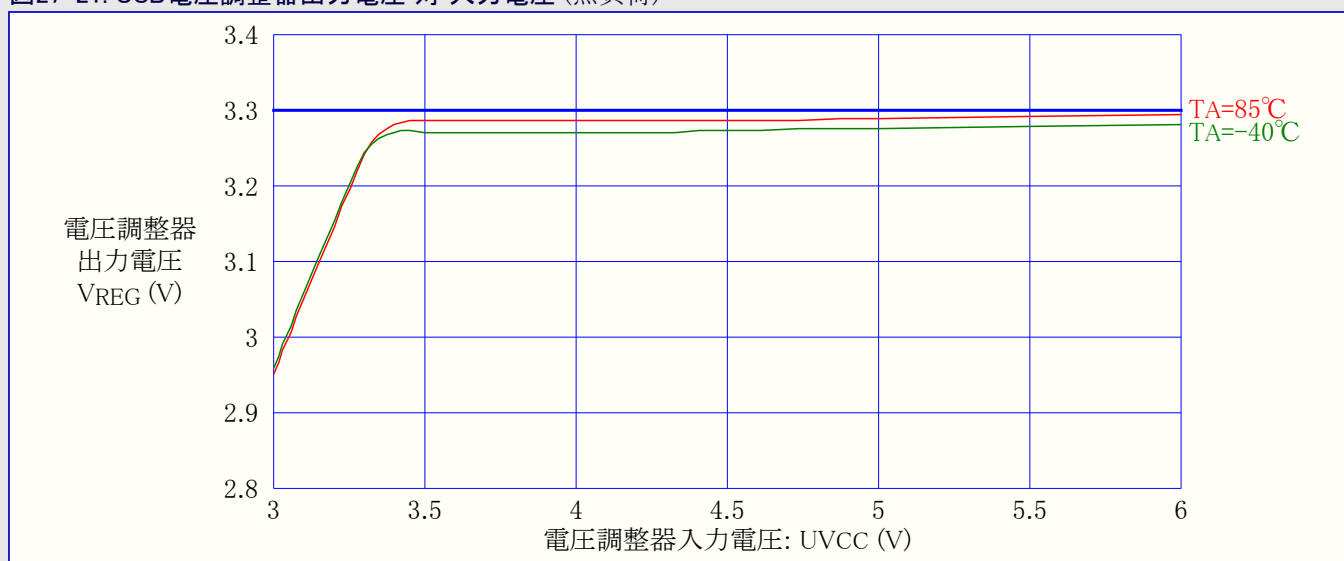
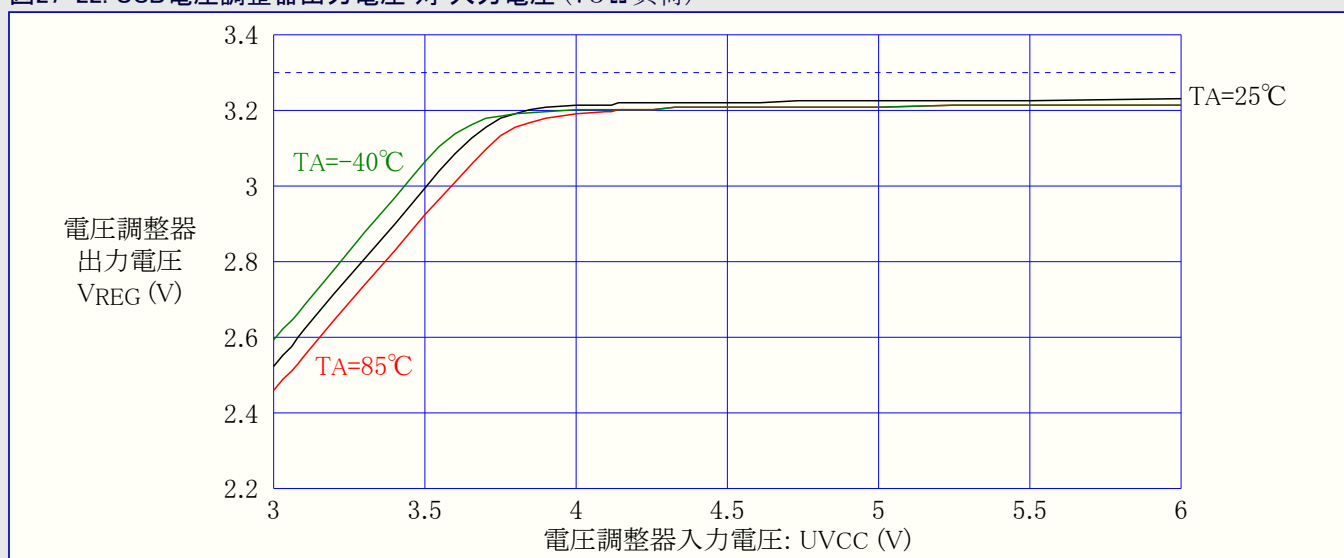
27.8. 内部基準(バンドギャップ)電圧

図27-20. 内部基準(バンドギャップ)電圧 対 動作電圧/温度



27.9. USB電圧調整器

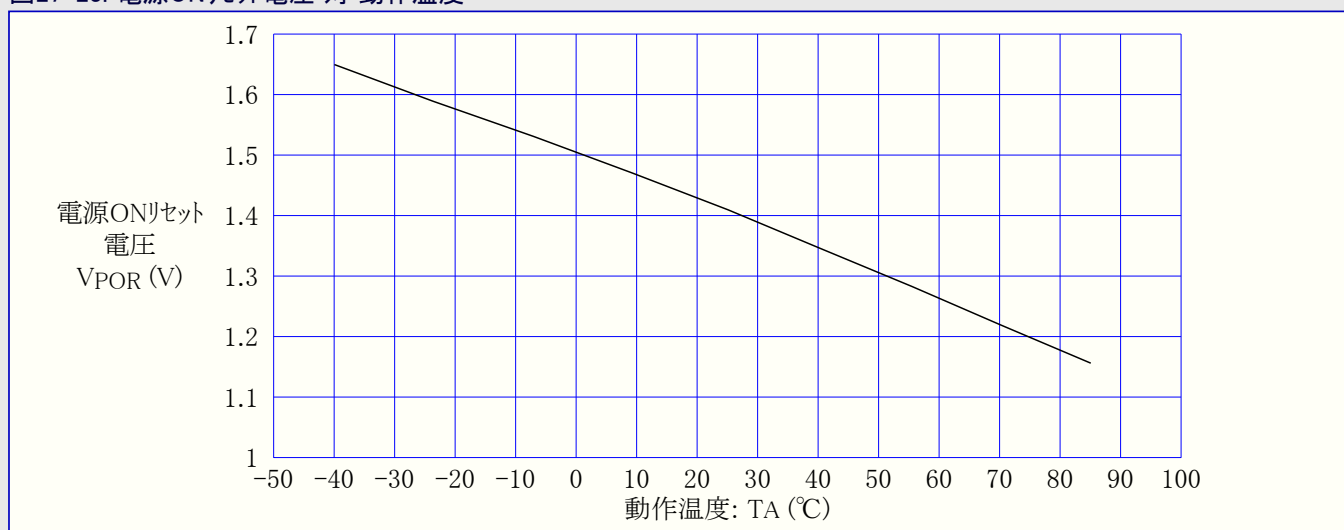
図27-21. USB電圧調整器出力電圧 対 入力電圧 (無負荷)

図27-22. USB電圧調整器出力電圧 対 入力電圧 (75 Ω 負荷)

注: 75 Ω は(全バス負荷)動作中のUSB周辺機能の最大平均消費と等価です。

27.10. 電源ONリセット

図27-23. 電源ONリセット電圧 対 動作温度



27.11. 低電圧検出器(BOD)閾値

図27-24. 低電圧検出器(BOD)閾値(スレッシュホールド)電圧 対 動作温度 (検出電圧2.7V)

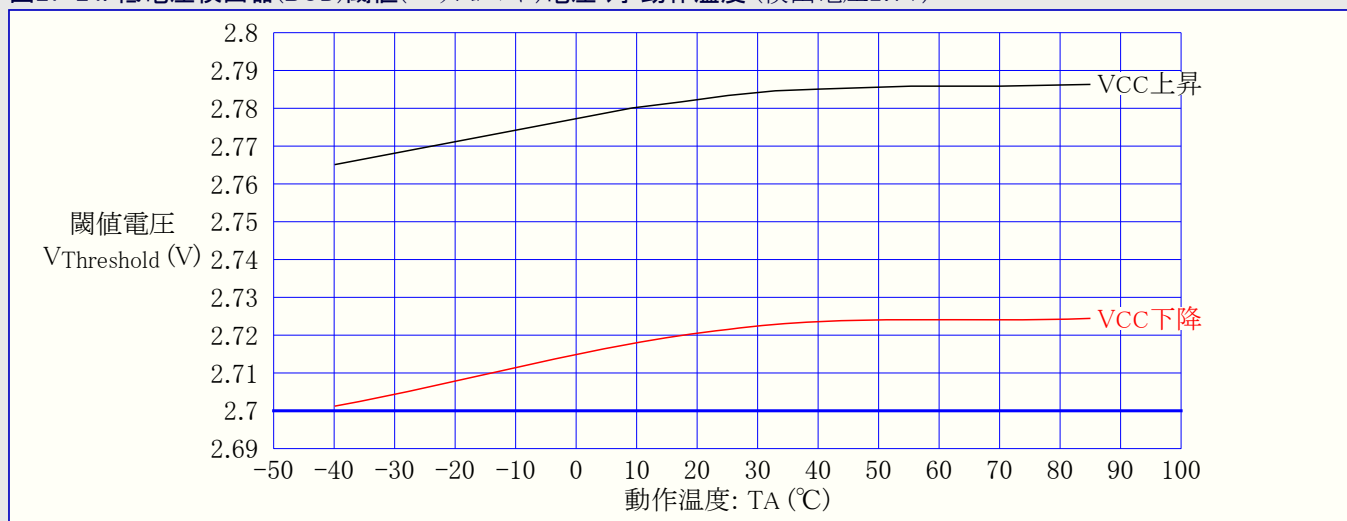


図27-25. 低電圧検出器(BOD)閾値(スレッシュホールド)電圧 対 動作温度 (検出電圧2.9V)

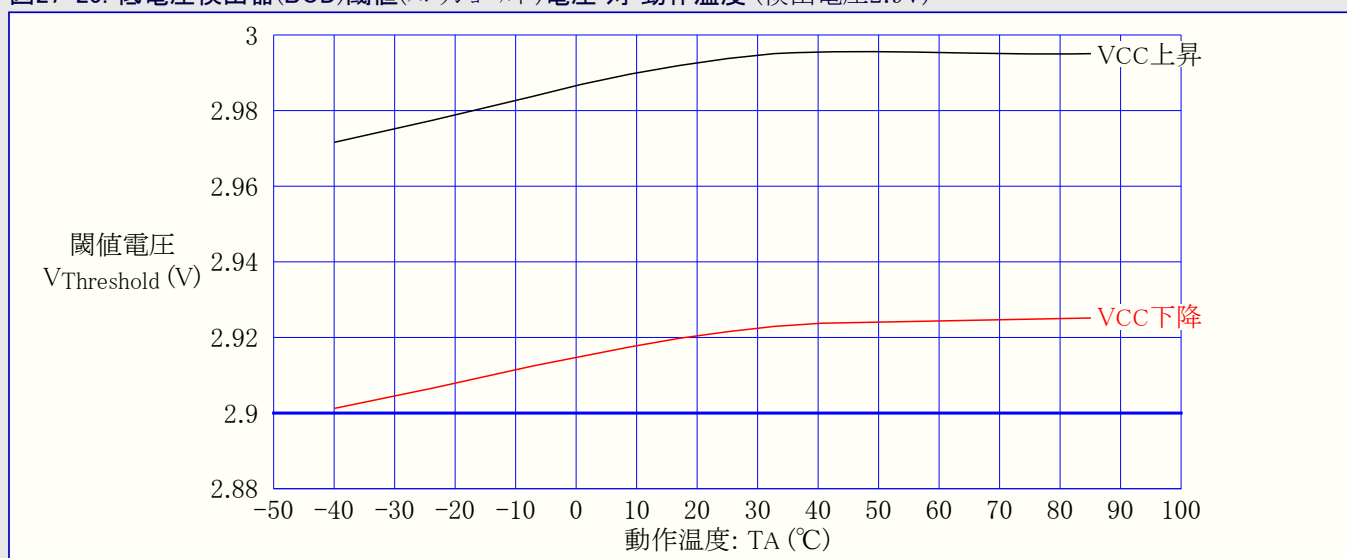


図27-26. 低電圧検出器(BOD)閾値(スレッシュホールド)電圧 対 動作温度 (検出電圧3.0V)

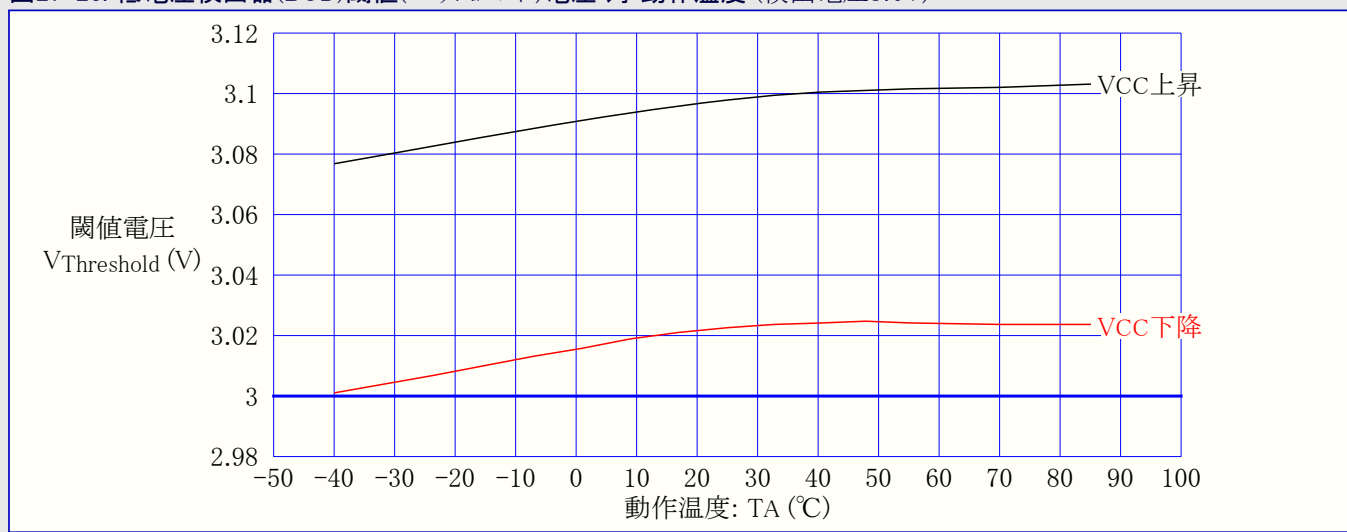


図27-27. 低電圧検出器(BOD)閾値(スレッシュホールド)電圧 対 動作温度 (検出電圧3.5V)

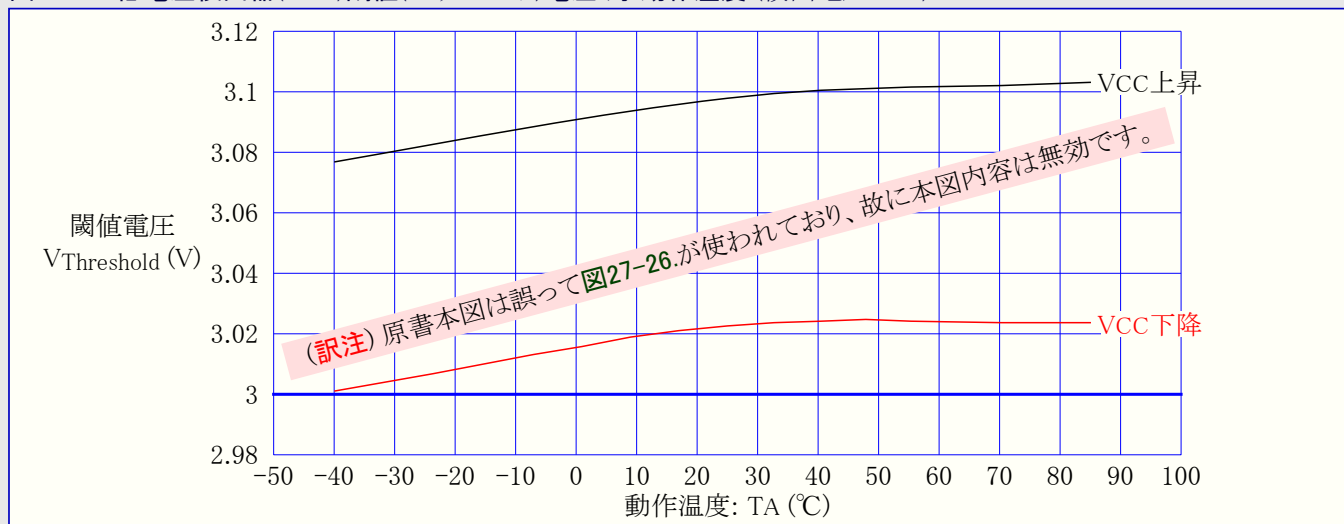


図27-28. 低電圧検出器(BOD)閾値(スレッシュホールド)電圧 対 動作温度 (検出電圧3.6V)

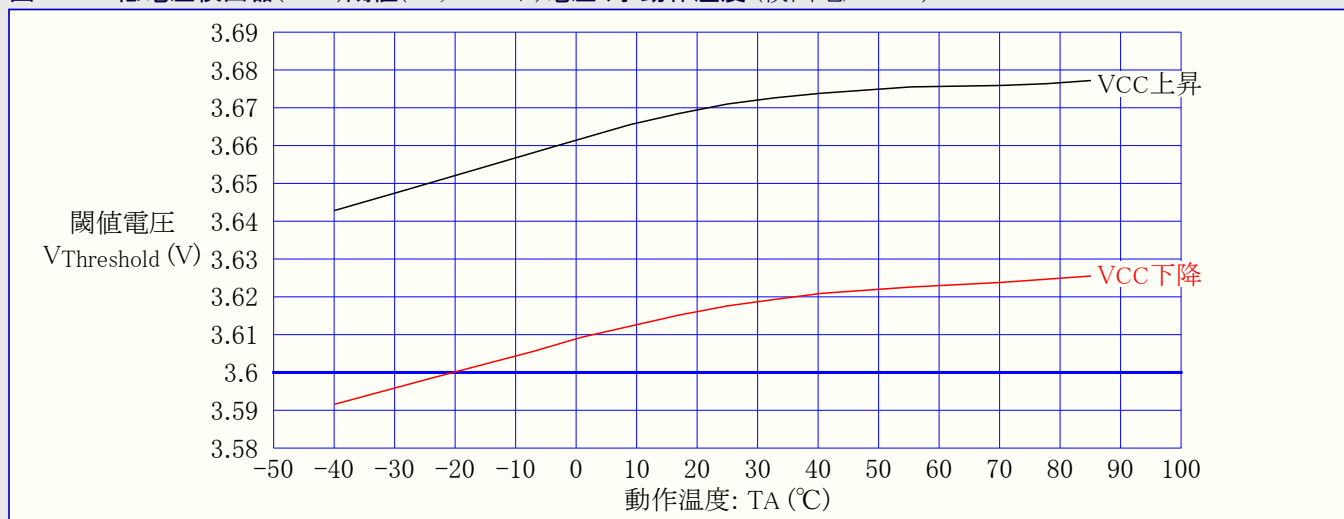


図27-29. 低電圧検出器(BOD)閾値(スレッシュホールド)電圧 対 動作温度 (検出電圧4.0V)

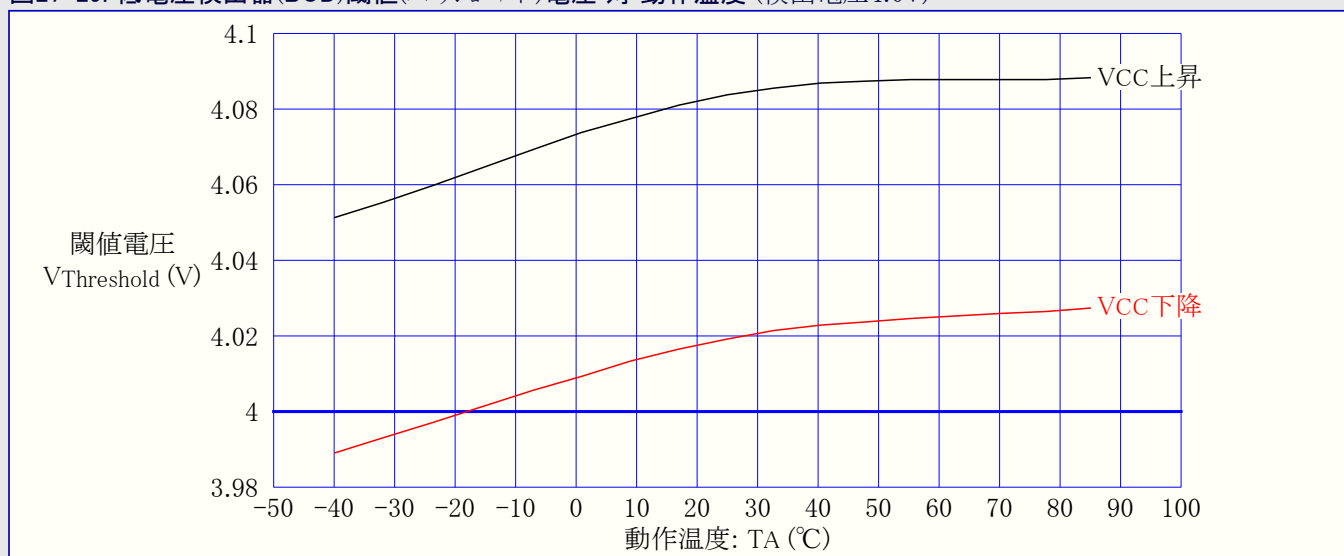
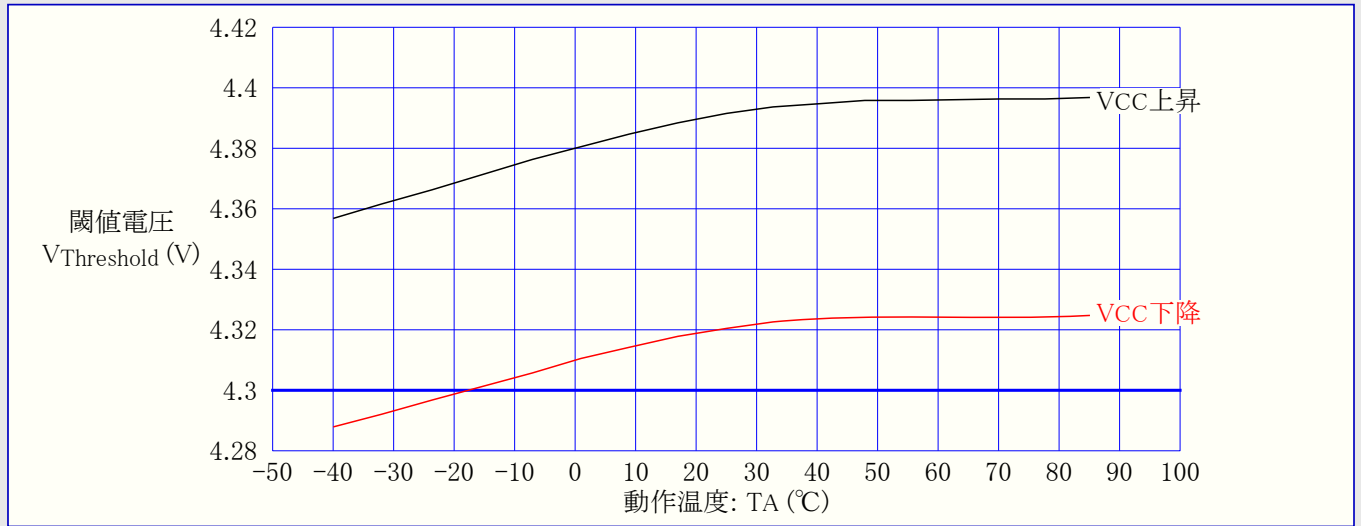
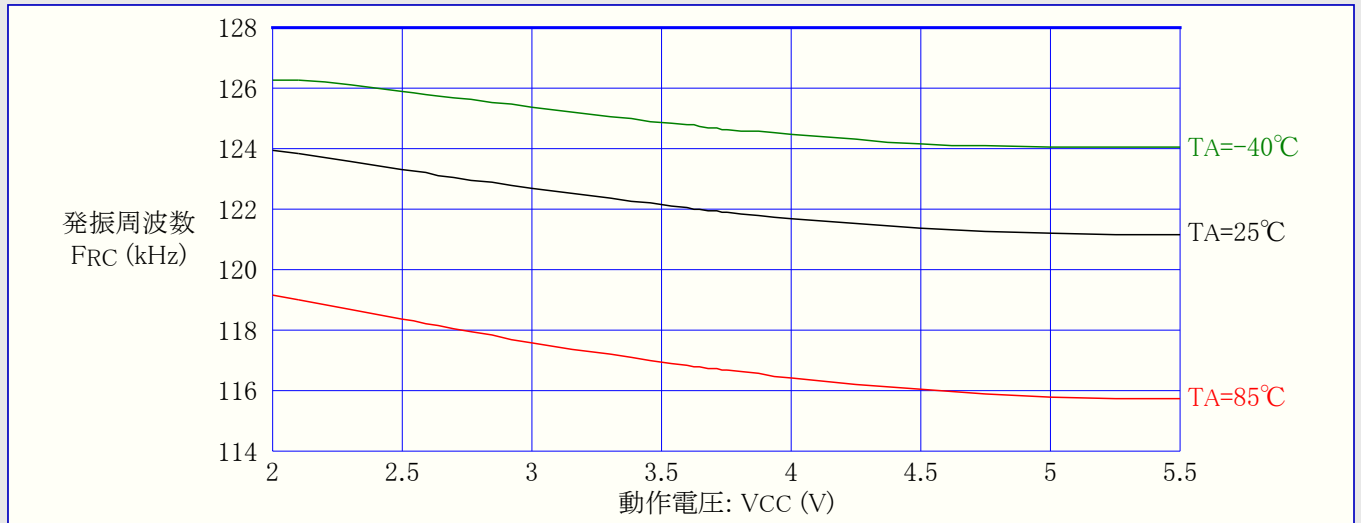


図27-30. 低電圧検出器(BOD)閾値(スレッシュホールド)電圧 対 動作温度 (検出電圧4.3V)



27.12. ウォッチドッグ タイマ周波数

図27-31. ウォッチドッグ発振器周波数 対 動作電圧



27.13. 校正付き内蔵RC発振器周波数

図27-32. 校正付き内蔵RC発振器周波数 対 発振校正(OSCCAL)値 (TA=25°C)

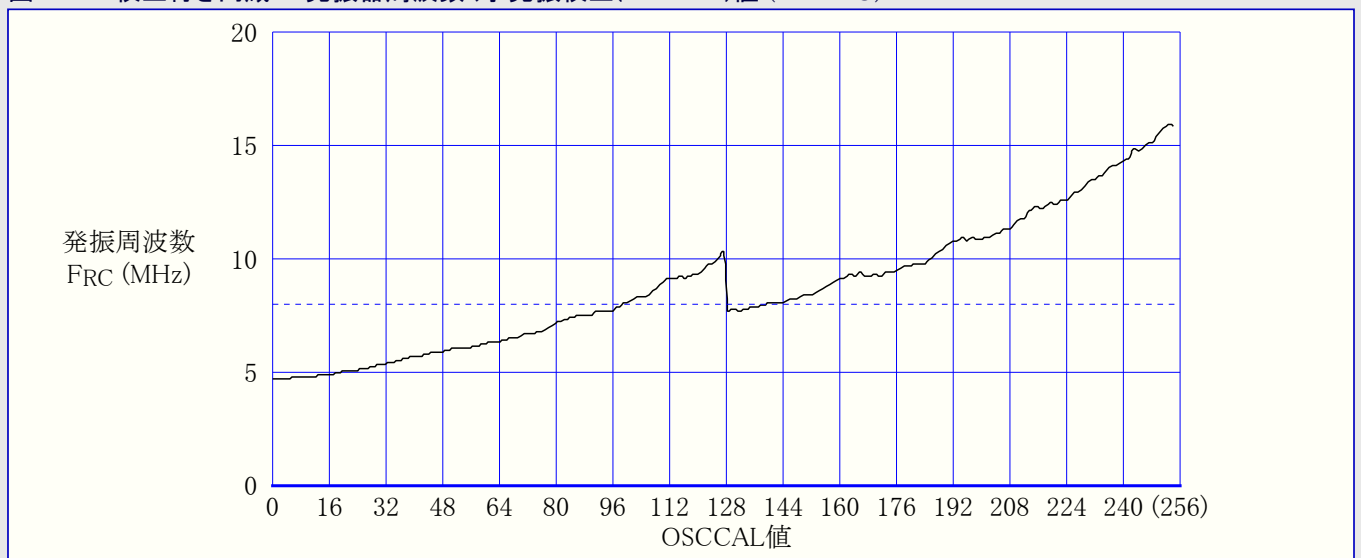
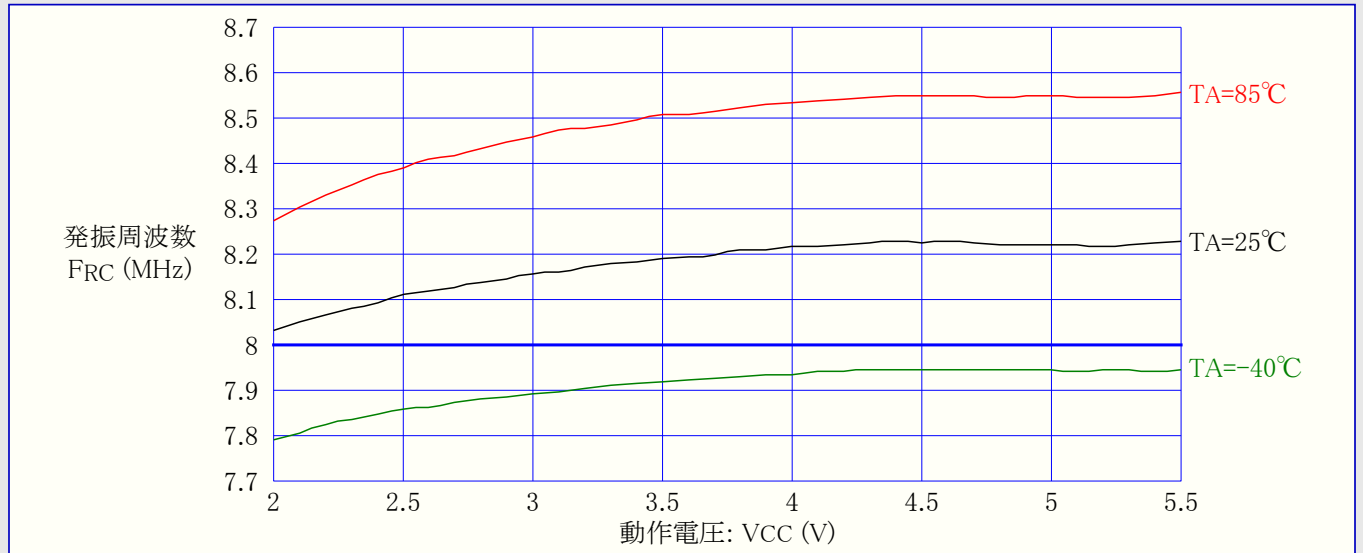


図27-33. 校正付き内蔵RC発振器周波数 対 動作電圧



28. レジスタ要約

拡張1/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)	UPOE	UPWE1	UPWE0	UPDRV1	UPDRV0	SCKI	DATAI	DPI	DMI	123	
(\$FA)	PS2CON	-	-	-	-	-	-	-	PS2EN	142	
(\$F9)	予約										
(\$F8)	予約										
(\$F7)	予約										
(\$F6)	予約										
(\$F5)	予約										
(\$F4)	UEINT	-	-	-	EPINT4	EPINT3	EPINT2	EPINT1	EPINT0	141	
(\$F3)	予約										
(\$F2)	UEBCLX	エントポイントX バイト計数器 (BYCT7~0)								141	
(\$F1)	UEDATX	エントポイントX データ (DAT7~0)								140	
(\$F0)	UEIENX	FLERRE	NAKINE	-	NAKOUTE	RXSTPE	RXOUTE	STALLEDE	TXINE	140	
(\$EF)	UESTA1X	-	-	-	-	-	CTRLDIR	CURRBK1	CURRBK0	138	
(\$EE)	UESTA0X	CFGOK	OVERFI	UNDERFI	-	DTSEQ1	DTSEQ0	NBUSYBK1	NBUSYBK0	138	
(\$ED)	UECFG1X	-	EPSIZE2	EPSIZE1	EPSIZE0	EPBK1	EPBK0	ALLOC	-	137	
(\$EC)	UECFG0X	EPTYPE1	EPTYPE0	-	-	-	-	-	EPDIR	137	
(\$EB)	UECONX	-	-	STALLRQ	STALLRQC	RSTDT	-	-	EPEN	136	
(\$EA)	UERST	-	-	-	EPRST4	EPRST3	EPRST2	EPRST1	EPRST0	136	
(\$E9)	UENUM	-	-	-	-	-	EPNUM2	EPNUM1	EPNUM0	135	
(\$E8)	UEINTX	FIFOCON	NAKINI	RWAL	NAKOUTI	RXSTPI	RXOUTI	STALLEDI	TXINI	139	
(\$E7)	予約										
(\$E6)	UDMFN	-	-	-	FNCERR	-	-	-	-	135	
(\$E5)	UDFNUMH	-	-	-	-	-	装置フレーム番号上位 (FNUM10~8)			135	
(\$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)							134	
(\$E2)	UDIEN	-	UPRSME	EORSME	WAKEUPE	EORSTE	SOFE	-	SUSPE	134	
(\$E1)	UDINT	-	UPRSMI	EORSMI	WAKEUPI	EORSTI	SOFI	-	SUSPI	133	
(\$E0)	UDCON	-	-	-	-	-	RSTCPU	RMWKUP	DETACH	133	
(\$DF)	予約										
(\$DE)	予約										
(\$DD)	予約										
(\$DC)	予約										
(\$DB)	予約										
(\$DA)	予約										
(\$D9)	予約										
(\$D8)	USBCON	USBE	-	FRZCLK	-	-	-	-	-	123	
(\$D7)	予約										
(\$D6)	予約										
(\$D5)	予約										
(\$D4)	予約										
(\$D3)	予約										
(\$D2)	CLKSTA	-	-	-	-	-	-	RCON	EXTON	18	
(\$D1)	CLKSEL1	RCCKSEL3	RCCKSEL2	RCCKSEL1	RCCKSEL0	EXCKSEL3	EXCKSEL2	EXCKSEL1	EXCKSEL0	18	
(\$D0)	CLKSEL0	RCSUT1	RCSUT0	EXSUT1	EXSUT0	RCE	EXTE	-	CLKS	18	
(\$CF)	予約										
(\$CE)	UDR	USART データレジスタ								106	
(\$CD)	UBRRH	-	-	-	-	USART ボーレートレジスタ上位 (UBRR11~8)					109
(\$CC)	UBRRL	USART ボーレートレジスタ下位 (UBRR7~0)									
(\$CB)	UCSRD	-	-	-	-	-	-	CTSEN	RTSEN	108	
(\$CA)	UCSRC	UMSEL1	UMSEL0	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL	107	
(\$C9)	UCSRB	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	107	
(\$C8)	UCSRA	RXC	TXC	UDRE	FE	DOR	UPE	U2X	MPCM	106	
(\$C7)	予約										
(\$C6)	予約										
(\$C5)	予約										
(\$C4)	予約										
(\$C3)	予約										
(\$C2)	予約										
(\$C1)	予約										
(\$C0)	予約										

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)	予約									
(\$BE)	予約									
(\$BD)	予約									
(\$BC)	予約									
(\$BB)	予約									
(\$BA)	予約									
(\$B9)	予約									
(\$B8)	予約									
(\$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)	予約									
(\$9C)	予約									
(\$9B)	予約									
(\$9A)	予約									
(\$99)	予約									
(\$98)	予約									
(\$97)	予約									
(\$96)	予約									
(\$95)	予約									
(\$94)	予約									
(\$93)	予約									
(\$92)	予約									
(\$91)	予約									
(\$90)	予約									
(\$8F)	予約									
(\$8E)	予約									
(\$8D)	OCR1CH			タイマ/カウンタ1 比較Cレジスタ上位バイト						86
(\$8C)	OCR1CL			タイマ/カウンタ1 比較Cレジスタ下位バイト						
(\$8B)	OCR1BH			タイマ/カウンタ1 比較Bレジスタ上位バイト						86
(\$8A)	OCR1BL			タイマ/カウンタ1 比較Bレジスタ下位バイト						
(\$89)	OCR1AH			タイマ/カウンタ1 比較Aレジスタ上位バイト						86
(\$88)	OCR1AL			タイマ/カウンタ1 比較A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						86
(\$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						85
(\$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	–	–	–	–	–	–	AIN1D	AIN0D	144
(\$7E)	予約									
(\$7D)	予約									
(\$7C)	予約									
(\$7B)	予約									
(\$7A)	予約									
(\$79)	予約									
(\$78)	予約									
(\$77)	予約									
(\$76)	予約									
(\$75)	予約									
(\$74)	予約									
(\$73)	予約									
(\$72)	予約									
(\$71)	予約									
(\$70)	予約									
(\$6F)	TIMSK1	–	–	ICIE1	–	OCIE1C	OCIE1B	OCIE1A	TOIE1	87
(\$6E)	TIMSK0	–	–	–	–	–	OCIE0B	OCIE0A	TOIE0	68
(\$6D)	予約									
(\$6C)	PCMSK1	–	–	–	PCINT12	PCINT11	PCINT10	PCINT9	PCINT8	43
(\$6B)	PCMSK0	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0	43
(\$6A)	EICRB	ISC71	ISC70	ISC61	ISC60	ISC51	ISC50	ISC41	ISC40	42
(\$69)	EICRA	ISC31	ISC30	ISC21	ISC20	ISC11	ISC10	ISC01	ISC00	41
(\$68)	PCICR	–	–	–	–	–	–	PCIE1	PCIE0	43
(\$67)	予約									
(\$66)	OSCCAL	内蔵RC発振器 発振校正値レジスタ								21
(\$65)	PRR1	PRUSB	–	–	–	–	–	–	PRUSART	28
(\$64)	PRR0	–	–	PRTIM0	–	PRTIM1	PRSPI	–	–	28
(\$63)	REGCR	–	–	–	–	–	–	–	REGDIS	25
(\$62)	WDTCKD	–	–	–	–	WDEWIF	WDEWIE	WCLKD1	WCLKD1	37
(\$61)	CLKPR	CLKPCE	–	–	–	CLKPS3	CLKPS2	CLKPS1	CLKPS1	23
(\$60)	WDTCSR	WDIF	WDIE	WDP3	WDCE	WDE	WDP2	WDP1	WDP0	36

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を検査できます。
- 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。AT90USB82/162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	—	—	—	—	—	—	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									150
\$38 (\$58)	予約									
\$37 (\$57)	SPMCSR	SPMIE	RWWSB	SIGRD	RWWSRE	BLBSET	PGWRT	PGERS	SPMEN	
\$36 (\$56)	予約									
\$35 (\$55)	MCUCR	—	—	—	PUD	—	—	IVSEL	IVCE	
\$34 (\$54)	MCUSR	—	—	USBRF	—	WDRF	BORF	EXTRF	PORF	32
\$33 (\$53)	SMCR	—	—	—	—	SM2	SM1	SM0	SE	26
\$32 (\$52)	予約									145
\$31 (\$51)	DWDR	デバッグ WIRE データ レジスタ								
\$30 (\$50)	ACSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	
\$2F (\$4F)	予約									
\$2E (\$4E)	SPDR	SPI データ レジスタ								
\$2D (\$4D)	SPSR	SPIF	WCOL	—	—	—	—	—	SPI2X	92
\$2C (\$4C)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	91
\$2B (\$4B)	GPIOR2	汎用I/Oレジスタ2								15
\$2A (\$4A)	GPIOR1	汎用I/Oレジスタ1								15
\$29 (\$49)	PLLCSR	—	—	—	PLL2	PLL1	PLL0	PLLE	PLOCK	24
\$28 (\$48)	OCR0B	タイマ/カウンタ0 比較B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	—	—	—	—	—	—	—	EEAR8	12
\$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								
\$1F (\$3F)	EECR	—	—	EEPM1	EEPM0	EERIE	EEMPE	EEPE	EERE	
\$1E (\$3E)	GPIOR0	汎用I/Oレジスタ0								15
\$1D (\$3D)	EIMSK	INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0	42
\$1C (\$3C)	EIFR	INTF7	INTF6	INTF5	INTF4	INTF3	INTF2	INTF1	INTF0	42
\$1B (\$3B)	PCIFR	—	—	—	—	—	—	PCIF1	PCIF0	43
\$1A (\$3A)	予約									
\$19 (\$39)	予約									
\$18 (\$38)	予約									
\$17 (\$37)	予約									
\$16 (\$36)	TIFR1	—	—	ICF1	—	OCF1C	OCF1B	OCF1A	TOV1	88
\$15 (\$35)	TIFR0	—	—	—	—	—	OCF0B	OCF0A	TOV0	68
\$14 (\$34)	予約									
\$13 (\$33)	予約									
\$12 (\$32)	予約									
\$11 (\$31)	予約									
\$10 (\$30)	予約									
\$0F (\$2F)	予約									
\$0E (\$2E)	予約									
\$0D (\$2D)	予約									
\$0C (\$2C)	予約									
\$0B (\$2B)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	54
\$0A (\$2A)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	54
\$09 (\$29)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	54
\$08 (\$28)	PORTC	PORTC7	PORTC6	PORTC5	PORTC4	—	PORTC2	PORTC1	PORTC0	54
\$07 (\$27)	DDRC	DDC7	DDC6	DDC5	DDC4	—	DDC2	DDC1	DDC0	54
\$06 (\$26)	PINC	PINC7	PINC6	PINC5	PINC4	—	PINC2	PINC1	PINC0	54
\$05 (\$25)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	54
\$04 (\$24)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	54
\$03 (\$23)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	54
\$02 (\$22)	予約									
\$01 (\$21)	予約									
\$00 (\$20)	予約									

(訳注) 原書本位置の注意は前頁に移動しました。

29. 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,C	1
NEG	Rd	2の補数	$Rd \leftarrow \$00 - Rd$	I,T,H,S,V,N,Z,C	1
SBR	Rd,K	汎用レジスタの(複数)ビット設定(1)	$Rd \leftarrow Rd \text{ OR } K$	I,T,H,S,0,N,Z,C	1
CBR	Rd,K	汎用レジスタの(複数)ビット解除(0)	$Rd \leftarrow Rd \text{ AND } (\$FF - K)$	I,T,H,S,0,N,Z,C	1
INC	Rd	汎用レジスタの増加(+1)	$Rd \leftarrow Rd + 1$	I,T,H,S,V,N,Z,C	1
DEC	Rd	汎用レジスタの減少(-1)	$Rd \leftarrow Rd - 1$	I,T,H,S,V,N,Z,C	1
TST	Rd	汎用レジスタのゼロとマイナス検査	$Rd \leftarrow Rd \text{ AND } Rd$	I,T,H,S,0,N,Z,C	1
CLR	Rd	汎用レジスタの全0設定(=\$00)	$Rd \leftarrow Rd \text{ EOR } Rd$	I,T,H,0,0,0,I,C	1
SER	Rd	汎用レジスタの全1設定(=\$FF)	$Rd \leftarrow \$FF$	I,T,H,S,V,N,Z,C	1
分岐命令					
RJMP	k	相対無条件分岐	$PC \leftarrow PC + k + 1$	I,T,H,S,V,N,Z,C	2
IJMP		レジスタ間接無条件分岐	$PC \leftarrow Z$	I,T,H,S,V,N,Z,C	2
JMP	k	絶対無条件分岐	$PC \leftarrow k$	I,T,H,S,V,N,Z,C	3
RCALL	k	相対サブルーチン呼び出し	$STACK \leftarrow PC, PC \leftarrow PC + k + 1$	I,T,H,S,V,N,Z,C	3
ICALL		レジスタ間接サブルーチン呼び出し	$STACK \leftarrow PC, PC \leftarrow Z$	I,T,H,S,V,N,Z,C	3
CALL	k	絶対サブルーチン呼び出し	$STACK \leftarrow PC, PC \leftarrow k$	I,T,H,S,V,N,Z,C	4
RET		サブルーチンからの復帰	$PC \leftarrow STACK$	I,T,H,S,V,N,Z,C	4
RETI		割り込みからの復帰	$PC \leftarrow STACK$	I,T,H,S,V,N,Z,C	4
CPSE	Rd,Rr	汎用レジスタ間比較、一致でスキップ	$Rd=Rr$ なら, $PC \leftarrow PC + 2 \text{ or } 3$	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Zレジスタ間接での取得	$R0 \leftarrow (Z)$	I,T,H,S,V,N,Z,C	3
LPM	Rd,Z	同上 (任意のレジスタへ)	$Rd \leftarrow (Z)$	I,T,H,S,V,N,Z,C	3
LPM	Rd,Z+	同上 (事後増加付き)	$Rd \leftarrow (Z), Z \leftarrow Z + 1$	I,T,H,S,V,N,Z,C	3
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	ニブル(4ビット)上位/下位交換	$Rd(7\sim4) \leftrightarrow Rd(3\sim0)$	I,T,H,S,V,N,Z,C	1
BSET	s	ステータスレジスタのビット設定(1)	$SREG(s) \leftarrow 1$	1,1,1,1,1,1,1,1	1
BCLR	s	ステータスレジスタのビット解除(0)	$SREG(s) \leftarrow 0$	0,0,0,0,0,0,0,0	1
BST	Rr,b	汎用レジスタのビットを一時フラグへ移動	$T \leftarrow Rr(b)$	I,T,H,S,V,N,Z,C	1
BLD	Rd,b	一時フラグを汎用レジスタのビットへ移動	$Rd(b) \leftarrow T$	I,T,H,S,V,N,Z,C	1
SEC		キャリーフラグを設定(1)	$C \leftarrow 1$	I,T,H,S,V,N,Z,C	1
CLC		キャリーフラグを解除(0)	$C \leftarrow 0$	I,T,H,S,V,N,Z,0	1
SEN		負フラグを設定(1)	$N \leftarrow 1$	I,T,H,S,V,1,Z,C	1
CLN		負フラグを解除(0)	$N \leftarrow 0$	I,T,H,S,V,0,Z,C	1
SEZ		ゼロフラグを設定(1)	$Z \leftarrow 1$	I,T,H,S,V,N,1,C	1
CLZ		ゼロフラグを解除(0)	$Z \leftarrow 0$	I,T,H,S,V,N,0,C	1
SEI		全割り込み許可	$I \leftarrow 1$	I,T,H,S,V,N,Z,C	1
CLI		全割り込み禁止	$I \leftarrow 0$	0,T,H,S,V,N,Z,C	1
SES		符号フラグを設定(1)	$S \leftarrow 1$	I,T,H,1,V,N,Z,C	1
CLS		符号フラグを解除(0)	$S \leftarrow 0$	I,T,H,0,V,N,Z,C	1
SEV		2の補数溢れフラグを設定(1)	$V \leftarrow 1$	I,T,H,S,1,N,Z,C	1
CLV		2の補数溢れフラグを解除(0)	$V \leftarrow 0$	I,T,H,S,0,N,Z,C	1
SET		一時フラグを設定(1)	$T \leftarrow 1$	I,1,H,S,V,N,Z,C	1
CLT		一時フラグを解除(0)	$T \leftarrow 0$	0,0,H,S,V,N,Z,C	1
SEH		ハーフキャリーフラグを設定(1)	$H \leftarrow 1$	I,T,H,S,V,N,Z,C	1
CLH		ハーフキャリーフラグを解除(0)	$H \leftarrow 0$	I,T,0,S,V,N,Z,C	1
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WIRE機能専用	I,T,H,S,V,N,Z,C	N/A

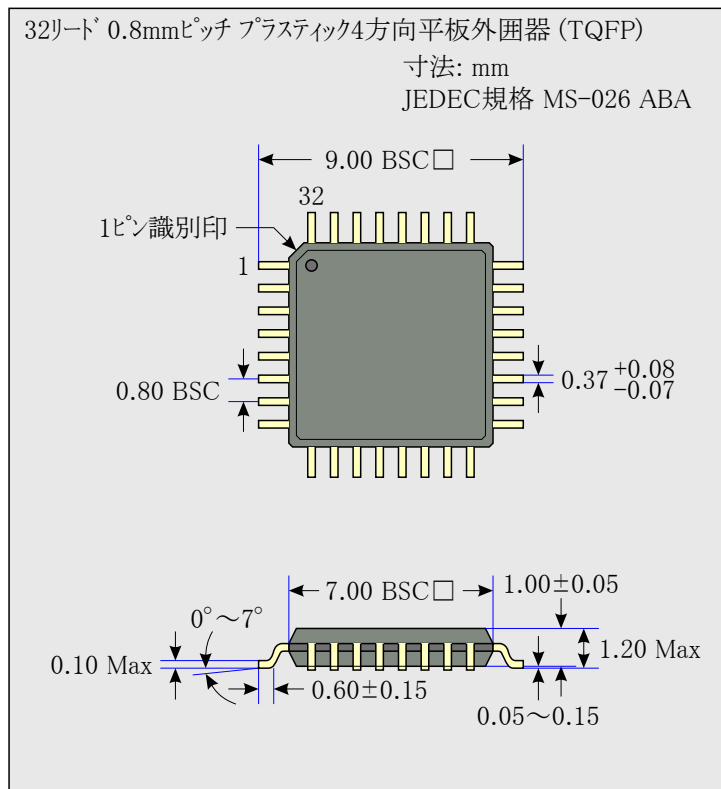
30. 注文情報

注文符号	速度 (MHz)	電源電圧	外囲器	動作範囲	製品刻印
AT90USB82-16MU	8~16	2.7~5.5V	QFN32	工業用 (-40°C~85°C), Green	90USB82-16MU
AT90USB162-16AU	8~16	2.7~5.5V	TQFP32	工業用 (-40°C~85°C)	90USB162-16AU
AT90USB162-16MU			QFN32	工業用 (-40°C~85°C)	90USB162-16MU

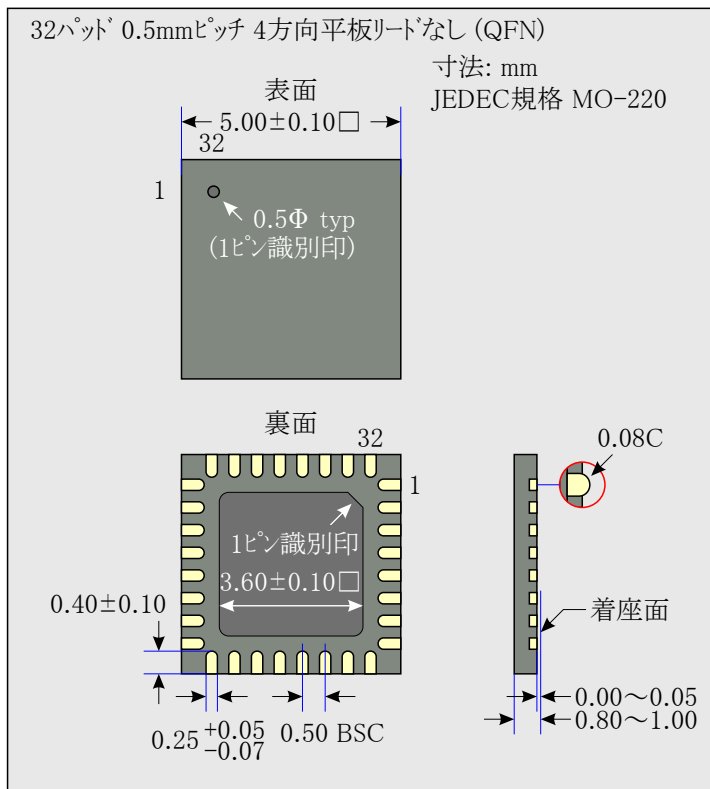
31. 外囲器情報

外囲器形式	
TQFP32	32リード 7×7×1mm 0.8mmピッチ プラスティック4方向平板外囲器 (TQFP) 注: 組み立てに超音波工程が使われる場合、印加されるべき周波数は12~26kHz範囲よりも低い、または高いかのどちらかであるべきことが推奨されます。
QFN32	32パッド 5×5×1mm 0.5mmピッチ 4方向平板リードなし/極小リード枠外囲器 (QFN)

31.1. TQFP32



31.2. QFN32



32. 障害情報

この章の改訂番号はAT90USB82/162デバイスの改訂版を参照してください。

シリコン版	日付, ロット番号刻印		
	AT90USB82	AT90USB162	
	QFN32	TQFP32	QFN32
初版	全ロット	0705 6J4972 0709 J4973-2 0709 J5597-1	90USB162-16MES
第2版		0709 F3150-1	0714 50-2 0722 50-3 0735 3151
第3版		0709以降の日付符号全て	上記以外の全ロット

- 休止形態での高消費電流
- USB休止(Suspend)動作での一時的挙動で過消費
- PS2のHLレベルがUCAPに強制
- 追加電力消費

82-①, 162-①~③
82-①, 162-①~③
162-①
162-②

1. 休止形態での高消費電流 (82-①, 162-①~③)

選んだ動作種別から保留割り込みがデバイスを起動復帰できない場合、SEI命令直後のSLEEP命令実行での休止中、消費電流が増します。

対策/対処

休止移行に先立って、休止形態からデバイスを起動復帰するのに使われない割り込みは禁止されるべきです。

2. USB休止(Suspend)での一時的挙動で過消費 (82-①, 162-①~③)

装置動作でUSBが休止(Suspend)の時にUSB信号線での一時的挙動が復帰状態を生じます。けれどもその挙動以降のアイドル状態はそれ以上休止(Suspend)割り込み(SUSPI)ビットを設定(1)しません。内部USB部は休止動作に留まりますが、USB差動受信部は未だ許可され、代表的に300μAの追加電力消費を生じます。一時的挙動後の休止状態検知は(SUIPIビット読みに代わり)ソフトウェアによって実行されるべきでしょう。

対策/対処

USB規格は休止(Suspend)状態で2.5mAまでの消費を給電装置に許します。

3. PS2のHLレベルがUCAPに強制 (162-①)

PS2動作種別設定時、その出力HLレベルはUCAP電圧水準に強制されます。

対策/対処

ありません。

4. 追加電力消費 (162-②)

代表的な電力消費は5Vで90μA、最悪の場合は160μAに増します。

対策/対処

ありません。

33. 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。

33.1. 7707A – 2007年3月

1. 初版

33.2. 7707B – 2007年11月

1. タイマ/カウンタ2とA/D変換器への全参照を削除
2. 電力削減動作とタイマ/カウンタ1項内の情報を明白化
3. USB設計指針と回路図を追加
4. AC/DC特性を更新
5. 障害情報項を更新

33.3. 7707C – 2007年12月

1. 障害情報項を更新

33.4. 7707D – 2008年7月

1. 158頁での発振器記述修正

33.5. 7707E – 2008年11月

1. 外圍器記述更新
2. 超音波組み立てに関する推奨を追加
3. 代表的な自己給電応用を更新

33.6. 7707F – 2010年11月

1. 121頁の「**割り込み**」を更新、FRZCLK=**1**をFRZCLK=**0**で置換
2. 172頁の「**電気的特性**」を更新、UVCCの最小と最大の値を追加
3. 図更新によって194頁の「**QFN32**」を置換
4. Atmelの新しい品質様式指針に従って最終頁を更新

目次

特徴	1	12.3. 交換ポート機能	47
1. ピン配置	2	12.4. I/Oポート用レジスタ	54
2. 概要	3	13. タイマ/カウンタ0,1の前置分周器	55
2.1. 構成図	3	14. 8ビット タイマ/カウンタ0 (PWM)	57
2.2. ピン説明	4	14.1. 概要	57
3. コード例について	5	14.2. タイマ/カウンタのクロック	58
4. AVR CPU コア	6	14.3. 計数器部	58
4.1. 序説	6	14.4. 比較出力部	59
4.2. 構造概要	6	14.5. 比較一致出力部	60
4.3. ALU (Arithmetic Logic Unit)	6	14.6. 動作種別	61
4.4. ステータス レジスタ	7	14.7. タイマ/カウンタのタイミング	64
4.5. 汎用レジスタ ファイル	8	14.8. 8ビット タイマ/カウンタ0用レジスタ	65
4.6. スタック ポインタ	8	15. 16ビット タイマ/カウンタ1	69
4.7. 命令実行タイミング	9	15.1. 概要	69
4.8. リセットと割り込みの扱い	9	15.2. 16ビット レジスタのアクセス	71
5. AVR AT90USB82/162のメモリ	11	15.3. タイマ/カウンタのクロック	73
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	11	15.4. 計数器部	73
5.2. データ用SRAMメモリ	11	15.5. 捕獲入力部	74
5.3. データ用EEPROMメモリ	12	15.6. 比較出力部	75
5.4. I/Oメモリ (レジスタ)	15	15.7. 比較一致出力部	77
6. システム クロックとクロック選択	16	15.8. 動作種別	78
6.1. クロックシステムとその配給	16	15.9. タイマ/カウンタのタイミング	82
6.2. クロック切り替え	16	15.10. 16ビット タイマ/カウンタ1用レジスタ	83
6.3. クロック元	19	16. SPI (直列周辺インターフェース)	89
6.4. クリスタル用低電力発振器	20	16.1. SSピンの機能	91
6.5. 校正付き内蔵RC発振器	21	16.2. データ転送形式	93
6.6. 外部クロック信号	22	17. USART	94
6.7. システム クロック出力緩衝部	22	17.1. 概要	94
6.8. システム クロック前置分周器	23	17.2. クロック生成	95
6.9. PLL	24	17.3. フレーム形式	96
7. 電力配給	25	17.4. USARTの初期化	97
8. 電力管理と休止形態	26	17.5. USARTのデータ送信	98
8.1. アイドル動作	26	17.6. USARTのデータ受信	99
8.2. パワーダウン動作	27	17.7. 非同期受信	102
8.3. パワーセーブ動作	27	17.8. ハードウェア流れ(フロー)制御	104
8.4. スタンバイ動作	27	17.9. 複数プロセッサ通信動作	105
8.5. 拡張スタンバイ動作	27	17.10. USART用レジスタ	106
8.6. 電力削減(電力削減レジスタ)	28	17.11. ホールレート設定例	110
8.7. 消費電力の最小化	29	18. USARTでのSPI動作	112
9. システム制御とリセット	30	18.1. 概要	112
9.1. AVRのリセット	30	18.2. クロック生成	112
9.2. リセット元	30	18.3. データ転送形式	112
9.3. 電源ONリセット	31	18.4. フレーム形式	113
9.4. 外部リセット	31	18.5. データ転送	114
9.5. 低電圧検出(BOD)リセット	31	18.6. MSPIMでのUSART用レジスタ	115
9.6. ウォッチドッグ リセット	31	18.7. USARTでのMSPIMとSPIの比較	117
9.7. USBリセット	32	19. USB制御器	118
9.8. 内部基準電圧	33	19.1. 特徴	118
9.9. ウォッチドッグ タイマ	34	19.2. 構成図	118
10. 割り込み	39	19.3. 代表的な応用実装	118
10.1. AT90USB82/162の割り込みベクタ	39	19.4. 一般的な動作	120
11. 外部割り込み	41	19.5. 電力動作種別	122
11.1. ピン変化割り込みタイミング	41	19.6. メモリ管理	122
12. 入出力ポート	44	19.7. パット休止	123
12.1. 序説	44	19.8. D+/D-読み書き	123
12.2. 標準デジタル入出力としてのポート	44	19.9. USB一般用レジスタ	123
		19.10. USBソフトウェア操作種別	124

20. USB装置動作	125	27.4. パワーダウン動作消費電流	179
20.1. 序説	125	27.5. アイドル動作消費電流	181
20.2. 電源ONとリセット	125	27.6. 活動動作消費電流	181
20.3. エントポイント リセット	125	27.7. リセット消費電流	182
20.4. USBリセット	125	27.8. 内部基準(バンドギャップ)電圧	182
20.5. エントポイント選択	125	27.9. USB電圧調整器	183
20.6. エントポイントの活性化(有効設定)	126	27.10. 電源ONリセット	183
20.7. アドレス設定	126	27.11. 低電圧検出器(BOD)閾値	184
20.8. 休止、起動復帰と再開	127	27.12. ウォッチドッグ タイマ周波数	186
20.9. 分離(切り離し)	127	27.13. 校正付き内蔵RC発振器周波数	186
20.10. 遠隔起動復帰	127	28. レジスタ要約	188
20.11. 不能応答(STALL)要求	127	29. 命令要約	192
20.12. 制御エントポイント管理	128	30. 注文情報	194
20.13. OUTエントポイント管理	129	31. 外圍器情報	194
20.14. INエントポイント管理	130	32. 障害情報	195
20.15. 等時(Isochronous)動作	131	33. データシート改訂履歴	196
20.16. オーバーフロー	131		
20.17. 装置割り込み	132		
20.18. USB装置一般用レジスタ	133		
20.19. USB装置エントポイント用レジスタ	135		
21. PS/2	142		
21.1. 特性	142		
22. アナログ比較器	143		
23. デバッグWIRE内蔵デバッグ機能	145		
23.1. 特徴	145		
23.2. 概要	145		
23.3. 物理インターフェース	145		
23.4. ソフトウェア中断点	145		
23.5. デバッグWIREの制限	145		
23.6. デバッグWIRE関連のI/Oメモリ内レジスタ	145		
24. ブートローダ支援 - RWW 自己プログラミング	146		
24.1. 特徴	146		
24.2. フラッシュメモリの応用領域とブートローダ領域	146		
24.3. 書き中に読みが可能な領域と不能な領域	146		
24.4. ブートローダ施錠ビット	148		
24.5. ブートローダプログラムへの移行	149		
24.6. 自己プログラミングでのフラッシュアドレス指定	151		
24.7. フラッシュメモリの自己プログラミング	151		
25. メモリプログラミング	157		
25.1. プログラムメモリとデータメモリ用施錠ビット	157		
25.2. ヒューズビット	158		
25.3. 識票バイト	159		
25.4. 校正值バイト	159		
25.5. 並列プログラミング	160		
25.6. 並列プログラミング手順	161		
25.7. 直列プログラミング	169		
25.8. 直列プログラミング手順	169		
26. 電気的特性	172		
26.1. 絶対最大定格	172		
26.2. DC特性	172		
26.3. 外部クロック特性	173		
26.4. 最高速と動作電圧	173		
26.5. SPI タイミング特性	174		
26.6. ハードウェアブート受付タイミング特性	174		
26.7. 校正付き内蔵RC発振器精度	175		
27. 代表特性(暫定)	176		
27.1. 入力電圧水準	176		
27.2. 出力電圧水準	177		
27.3. I/Oプルアップ電流	179		



Atmel Corporation

2325 Orchard Parkway
San Jose, CA 95131
USA
TEL (+1)(408) 441-0311
FAX (+1)(408) 487-2600
www.atmel.com

Atmel Asia Limited

Unit 01-5 & 16, 19F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
HONG KONG
TEL (+852) 2245-6100
FAX (+852) 2722-1369

Atmel Munich GmbH

Business Campus
Parking 4
D-85748 Garching b. Munich
GERMANY
TEL (+49) 89-31970-0
FAX (+49) 89-3194621

Atmel Japan

141-0032 東京都品川区
大崎1-6-4
新大崎勸業ビル 16F
アトメル ジャパン合同会社
TEL (+81)(3)-6417-0300
FAX (+81)(3)-6417-0370

© 2010 Atmel Corporation. 不許複製

Atmel®、ロゴとそれらの組み合わせ、それとその他はAtmel Corporationの登録商標または商標またはその付属物です。他の用語と製品名は一般的に他の商標です。

お断り: 本資料内の情報はAtmel製品と関連して提供されています。本資料またはAtmel製品の販売と関連して承諾される何れの知的所有権も禁反言あるいはその逆によって明示的または暗示的に承諾されるものではありません。Atmelのウェブサイトにある販売の条件とAtmelの定義での詳しい説明を除いて、商品性、特定目的に関する適合性、または適法性の暗黙保証に制限せず、Atmelはそれらを含むその製品に関連する暗示的、明示的または法令による如何なる保証も否認し、何ら責任がないと認識します。たとえAtmelがそのような損害賠償の可能性を進言されたとしても、本資料を使用できない、または使用以外で発生する(情報の損失、事業中断、または利益と損失に関する制限なしの損害賠償を含み)直接、間接、必然、偶然、特別、または付随して起こる如何なる損害賠償に対しても決してAtmelに責任がないでしょう。Atmelは本資料の内容の正確さまたは完全性に関して断言または保証を行わず、予告なしでいつでも製品内容と仕様の変更を行う権利を保留します。Atmelはここに含まれた情報を更新することに対してどんな公約も行いません。特に別の方法で提供されなければ、Atmel製品は車載応用に対して適当ではなく、使用されるべきではありません。Atmel製品は延命または生命維持を意図した応用での部品としての使用に対して意図、認定、または保証されません。

© HERO 2022.

本データシートはAtmelのAT90USB82/162英語版データシート(Rev.7707F-11/10)の翻訳日本語版です。日本語では不自然となる重複する形容表現は省略されている場合があります。日本語では難解となる表現は大幅に意識されている部分もあります。必要に応じて一部加筆されています。頁割の変更により、原本より頁数が少なくなっています。

汎用入出力ポートの出力データレジスタとピン入力、対応関係からの理解の容易さから出力レジスタと入力レジスタで統一表現されています。一部の用語がより適切と思われる名称に変更されています。必要と思われる部分には()内に英語表記や略称などを残す形で表記しています。

青字の部分はリンクとなっています。一般的に赤字の0,1は論理0,1を表します。その他の赤字は重要な部分を表します。

USB関係記述に不具合がありますのでご注意ください。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。