



ATtiny87/167

tinyAVR® データシート

序説

ATtiny87/167はAVR®強化RISC構造に基づく低電力CMOS 8ビット マイクロ コントローラです。ATtiny87/167は8/16Kバイトのフラッシュ メモリ、512バイトのSRAM、512バイトのEEPROMを持つ20/32ピン デバイスです。単一クロック周期での強力な命令の実行により、デバイスはMHz当たり1MIPS(100万命令/秒)近くの単位処理量を達成し、消費電力対処理速度に対するデバイスの最適化をシステム設計者に許します。

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

特徴

- 高性能、低消費AVR® 8ビット マイクロ コントローラ
- 進化したRISC構造
 - 強力な123命令(多くは1周期実行)
 - 32個の1バイト長汎用レジスタ
 - 完全なスタティック動作
- データ メモリと不揮発性プログラム メモリ
 - 実装書き換え可能な8K/16Kバイト(4K/8K語)フラッシュ メモリ内蔵
 - 10,000回の書き換えが可能
 - 実装書き換え可能な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
 - データ保持力: 20年/85°C, 100年/25°C
 - SPIポート経由の実装書き込み(ISP)
 - 低容量LIN/UARTソフトウェア実装書き込み可能
 - ソフトウェア保護用の設定可能な施錠機能
- 内蔵周辺機能
 - LIN2.1と1.3の制御器または8ビットUART
 - 1つの前置分周器付き8ビット非同期タイマ/カウンタ
 - 1つの比較または8ビットPWMチャネル
 - 1つの前置分周器付き16ビット同期タイマ/カウンタ
 - 外部事象計数器
 - 2つの比較または各々が最大4つの出力ピンを駆動するPWMチャネル
 - 主装置/従装置動作SPI直列インターフェース
 - 開始条件検出器付き多用途直列インターフェース(USI)
 - 10ビット A/D変換器
 - 11本のシングル エント' チャネル
 - 設定可能な利得(×8, ×20)付きの8つの差動ADCチャネル対
 - 選択可能な基準電圧付きチップ内蔵アナログ比較器
 - LIN節(ノード)認証用100µA±10%電流源
 - チップ上の温度感知器
 - 独立した発振器付きの設定可能なウォッチドッグ タイマ
- 特殊マイクロ コントローラ機能
 - 電力制御やEMC減少のためのソフトウェア制御クロック切り替え
 - デバッグWIRE内蔵デバッグ システム
 - 外部及び内部の割り込み
 - アイドル、A/D変換雑音低減、パワーダウン、パワーセーブの4つの低消費動作
 - 電源ONリセット回路と設定可能な低電圧検出器(BOD)
 - 内蔵8MHz校正付き発振器
 - 4~16MHzと32kHzのクリスタル発振子/セラミック振動子用の発振器
- I/Oと外圍器
 - 16ビットの設定可能なI/O
 - 20ピンSOIC、20ピンTSSOP、20ピンWQFN、32ピンVQFN
- 動作電圧
 - 1.8~5.5V
- 動作温度
 - -40~85°C
- 動作速度
 - 0~4MHz/1.8~5.5V
 - 0~8MHz/2.7~5.5V
 - 0~16MHz/4.5~5.5V
- 工業用温度範囲

目次

序説	1	9.3. 交換ポート機能	54
特徴	1	9.4. I/Oポート用レジスタ	62
1. 説明	5	10. 8ビット タイマ/カウンタ0と非同期動作	63
1.1. ATtiny87とATtiny167間の違い	5	10.1. 特徴	63
1.2. デバイス説明	5	10.2. 概要	63
1.3. 構成図	6	10.3. タイマ/カウンタのクロック	64
1.4. ピン配置	7	10.4. 計数器部	64
1.5. ピン説明	9	10.5. 比較出力部	65
1.6. 資料	9	10.6. 比較一致出力部	66
1.7. コード例について	9	10.7. 動作種別	67
1.8. データ保持力	9	10.8. タイマ/カウンタのタイミング	70
1.9. お断り	9	10.9. タイマ/カウンタ0の非同期動作	71
2. AVR CPU コア	10	10.10. タイマ/カウンタ0の前置分周器	72
2.1. 概要	10	10.11. 8ビット タイマ/カウンタ0用レジスタ	73
2.2. ALU (Arithmetic Logic Unit)	10	11. タイマ/カウンタ1の前置分周器	77
2.3. ステータス レジスタ	11	11.1. 概要	77
2.4. 汎用レジスタ ファイル	12	11.2. タイマ/カウンタ前置分周器制御用レジスタ	78
2.5. スタック ポインタ	12	12. 16ビット タイマ/カウンタ1	79
2.6. 命令実行タイミング	13	12.1. 特徴	79
2.7. リセットと割り込みの扱い	13	12.2. 概要	79
3. AVRのメモリ	15	12.3. 16ビット レジスタのアクセス	81
3.1. 実装書き換え可能なプログラム用フラッシュ メモリ	15	12.4. タイマ/カウンタのクロック	83
3.2. データ用SRAMメモリ	16	12.5. 計数器部	83
3.3. データ用EEPROMメモリ	17	12.6. 捕獲入力部	84
3.4. I/Oメモリ (レジスタ)	19	12.7. 比較出力部	85
3.5. メモリ関係レジスタ	19	12.8. 比較一致出力部	87
4. システム クロックとクロック任意選択	21	12.9. 動作種別	89
4.1. クロック系統とその配給	21	12.10. タイマ/カウンタのタイミング	93
4.2. クロック元	22	12.11. 16ビット タイマ/カウンタ1用レジスタ	94
4.3. 動的クロック切り替え	26	13. 直列周辺インターフェース (SPI)	99
4.4. システム クロック前置分周器	30	13.1. 特徴	99
4.5. クロック関係レジスタ	31	13.2. SSピンの機能	101
5. 電力管理と休止形態	34	13.3. データ転送形式	103
5.1. 休止形態種別	34	14. 多用途直列インターフェース (USI)	104
5.2. 低電圧検出器(BOD)禁止	34	14.1. 特徴	104
5.3. アイドル動作	34	14.2. 概要	104
5.4. A/D変換雑音低減動作	35	14.3. 機能説明	105
5.5. パワーダウンス動作	35	14.4. USIでの代替使用	108
5.6. パワーセーブ動作	35	14.5. USI用レジスタ	109
5.7. 電力削減 (レジスタ)	35	15. LIN/UART	112
5.8. 消費電力の最小化	36	15.1. LIN特徴	112
5.9. 電力管理用レジスタ	37	15.2. UART特徴	112
6. システム制御とリセット	39	15.3. LIN規約	112
6.1. リセット	39	15.4. LIN/UART制御器	114
6.2. 内部基準電圧	41	15.5. LIN/UART説明	117
6.3. ウォッチドッグ タイマ	42	15.6. LIN/UART用レジスタ	123
7. 割り込み	45	16. 電流源 (ISRC)	129
7.1. ATtiny87/167の割り込みベクタ	45	16.1. 特徴	129
7.2. ATtiny87でのプログラム構成設定	46	16.2. 代表的な応用	129
7.3. ATtiny167でのプログラム構成設定	46	16.3. 電流源用レジスタ	130
8. 外部割り込み	47	17. アナログ比較器 (AC)	131
8.1. 概要	47	17.1. アナログ比較器入力	131
8.2. ピン変化割り込みタイミング	47	17.2. アナログ比較器用レジスタ	132
8.3. 外部割り込み用レジスタ	47	18. A/D変換器 (ADC)	134
9. 入出力ポート	50	18.1. 特徴	134
9.1. 序説	50	18.2. 概要	134
9.2. 標準デジタル入出力としてのポート	50	18.3. 操作	136

18.4.	変換の開始	136
18.5.	前置分周と変換タイミング	137
18.6.	チャネル変更と基準電圧選択	138
18.7.	雑音低減機能	139
18.8.	A/D変換の結果	140
18.9.	温度測定	141
18.10.	内部基準電圧出力	141
18.11.	A/D変換用レジスタ	142
19.	デバッグWIRE内蔵デバッグ機能	146
19.1.	特徴	146
19.2.	概要	146
19.3.	物理インターフェース	146
19.4.	ソフトウェア中断点	146
19.5.	デバッグWIREの制限	146
19.6.	デバッグWIRE用レジスタ	146
20.	フラッシュメモリのプログラミング	147
20.1.	フラッシュメモリの自己プログラミング	147
20.2.	自己プログラミングでのフラッシュアドレス指定	147
21.	メモリプログラミング	152
21.1.	プログラムメモリとデータメモリ用施錠ビット	152
21.2.	ヒューズビット	153
21.3.	識票バイト	154
21.4.	校正バイト	154
21.5.	ページ容量	154
21.6.	並列プログラミング	155
21.7.	並列プログラミング手順	156
21.8.	直列プログラミング	161
22.	電気的特性	164
22.1.	絶対最大定格	164
22.2.	DC特性	164
22.3.	速度	165
22.4.	クロック特性	165
22.5.	リセット特性	166
22.6.	内部基準電圧特性	166
22.7.	電流源特性	166
22.8.	A/D変換器特性	167
22.9.	並列プログラミング特性	168
22.10.	SPI タイミング特性	170
23.	代表特性	171
23.1.	活動動作消費電流	171
23.2.	アイドル動作消費電流	172
23.3.	入出力単位部の消費(供給)電流	173
23.4.	パワーダウン動作消費電流	174
23.5.	リセット消費電流	174
23.6.	プルアップ抵抗	175
23.7.	出力駆動部能力	176
23.8.	入力閾値とヒステリシス(対入出力ポート)	178
23.9.	低電圧検出器(BOD)、バンドキャップ、リセット	178
23.10.	内部発振器	180
24.	レジスタ要約	182
25.	命令要約	186
26.	注文情報	188
27.	外圍器情報	189
28.	障害情報	193
29.	データシート改訂履歴	194

1. 説明

1.1. ATtiny87とATtiny167間の違い

ATtiny87とATtiny167はハードウェアとソフトウェア互換です。それらは表1-1.で示されるようにメモリ容量だけが異なります。

表1-1. メモリ容量要約

デバイス名	フラッシュ メモリ	EEPROM	SRAM	割り込みベクタの大きさ
ATtiny87	8Kバイト	512バイト	512バイト	1命令語/ベクタ
ATtiny167	16Kバイト	512バイト	512バイト	2命令語/ベクタ

1.2. デバイス説明

ATtiny87/167はAVR強化RISC構造に基づく低消費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によって、ATtiny87/167は電力消費対処理速度の最適化をシステム設計者に許す、MHzあたり1MIPSに達する単位処理量を成し遂げます。

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

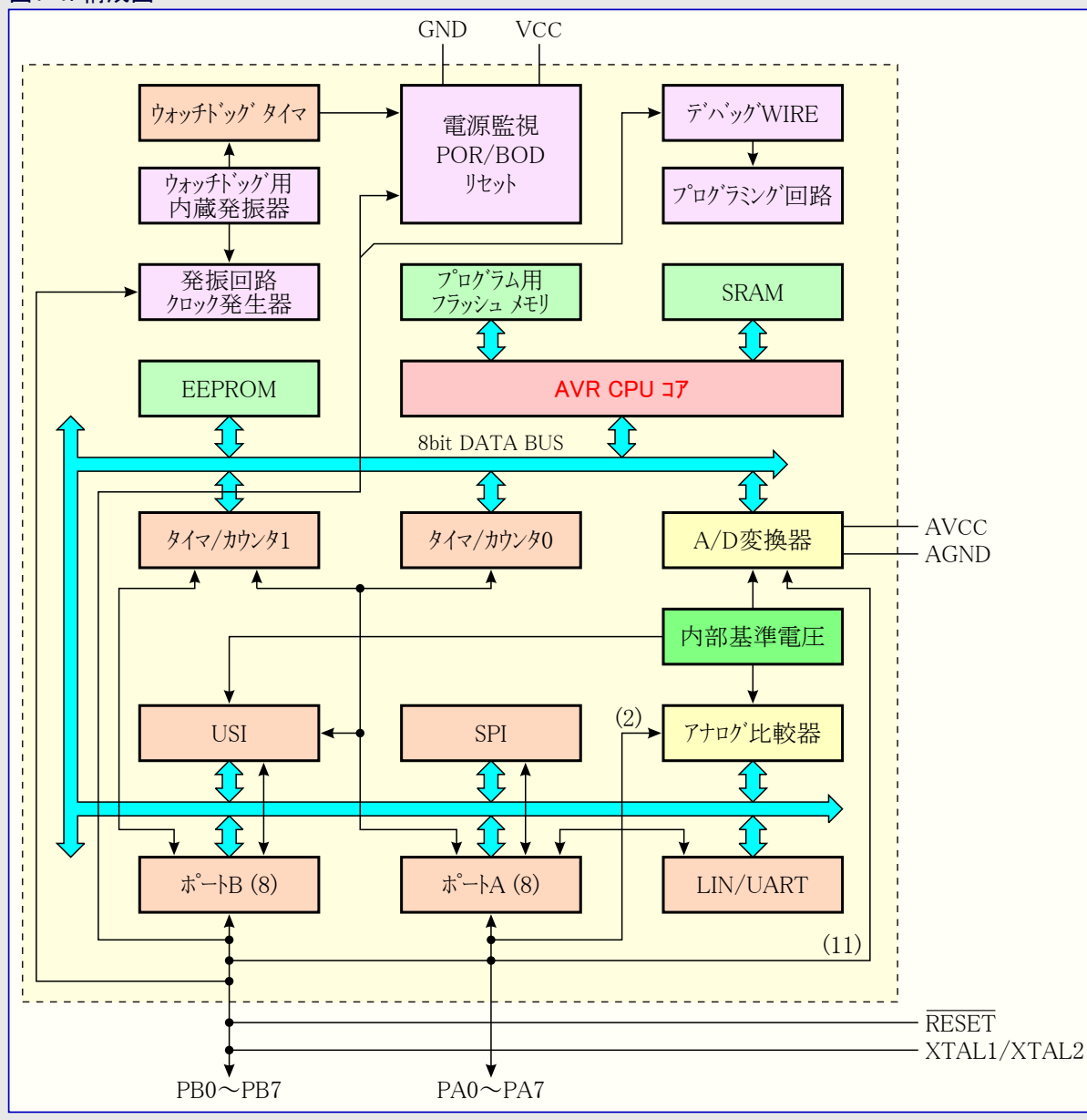
ATtiny87/167は次の特徴、8/16Kバイトの実装書き換え可能なフラッシュ メモリと512バイトのEEPROM、512バイトのSRAM、16本の汎用入出力線、32個の汎用レジスタ、比較動作やPWMを含む柔軟性のある1つ8ビット タイマ/カウンタと1つの16ビット タイマ/カウンタ、多用途直列インターフェース(USI)、LIN制御器、内部及び外部の割り込み、11チャネルの10ビットA/D変換器、設定変更可能な内部発振器付きウォッチドッグ タイマ、ソフトウェアで選択できる4つの低消費動作機能を提供します。アイドル動作では動作を停止しますが、SRAM、タイマ/カウンタ、A/D変換器、アナログ比較器、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、以降のハードウェアリセットか外部割り込みまで他の全機能を禁止(無効)にします。A/D変換雑音低減動作はA/D変換中の切り替え雑音を最小とために、A/D変換器を除く周辺機能とCPUを停止します。

本デバイスはMicrochip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コア上ブートプログラムの実行によって再書き込みができます。ブートプログラムはフラッシュ メモリ内の応用プログラムの読み込みでどのインターフェースでも使えます。モノリシック チップ上の自己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Microchip ATtiny87/167は多くの組み込み制御の応用に対して高度な柔軟性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。

ATtiny87/167 AVRはCコンパイラ、マクロ アセンブラ、プログラム デバッガ/シミュレータ、インサーキット エミュレータ、評価キットを含む専用のプログラム及びシステム開発ツールで支援されます。

1.3. 構成図

図1-1. 構成図



1.4. ピン配置

図1-2. ATtiny87/167ピン配置 – SIOC20-TSSOP20

(PCINT0/RXLIN/RXD/ADC0) PA0	1	20	PB0 (OC1AU/DI/SDA/PCINT8)
(PCINT1/TXLIN/TXD/ADC1) PA1	2	19	PB1 (OC1BU/DO/PCINT9)
(PCINT2/MISO/DO/OC0A/ADC2) PA2	3	18	PB2 (OC1AV/USCK/SCL/PCINT10)
(PCINT3/INT1/ISRC/ADC3) PA3	4	17	PB3 (OC1BV/PCINT11)
AVCC	5	16	GND
AGND	6	15	VCC
(PCINT4/MOSI/SDA/DI/ICP1/ADC4) PA4	7	14	PB4 (OC1AW/XTAL1/PCINT12)
(PCINT5/SCK/SCL/USCK/T1/ADC5) PA5	8	13	PB5 (ADC8/OC1BW/XTAL2/CLKO/PCINT13)
(PCINT6/SS/AIN0/ADC6) PA6	9	12	PB6 (ADC9/OC1AX/INT0/PCINT14)
(PCINT7/AREF/XREF/AIN1/ADC7) PA7	10	11	PB7 (ADC10/OC1BX/RESET/dW/PCINT15)

図1-3. ATtiny87/167ピン配置 – VQFN32

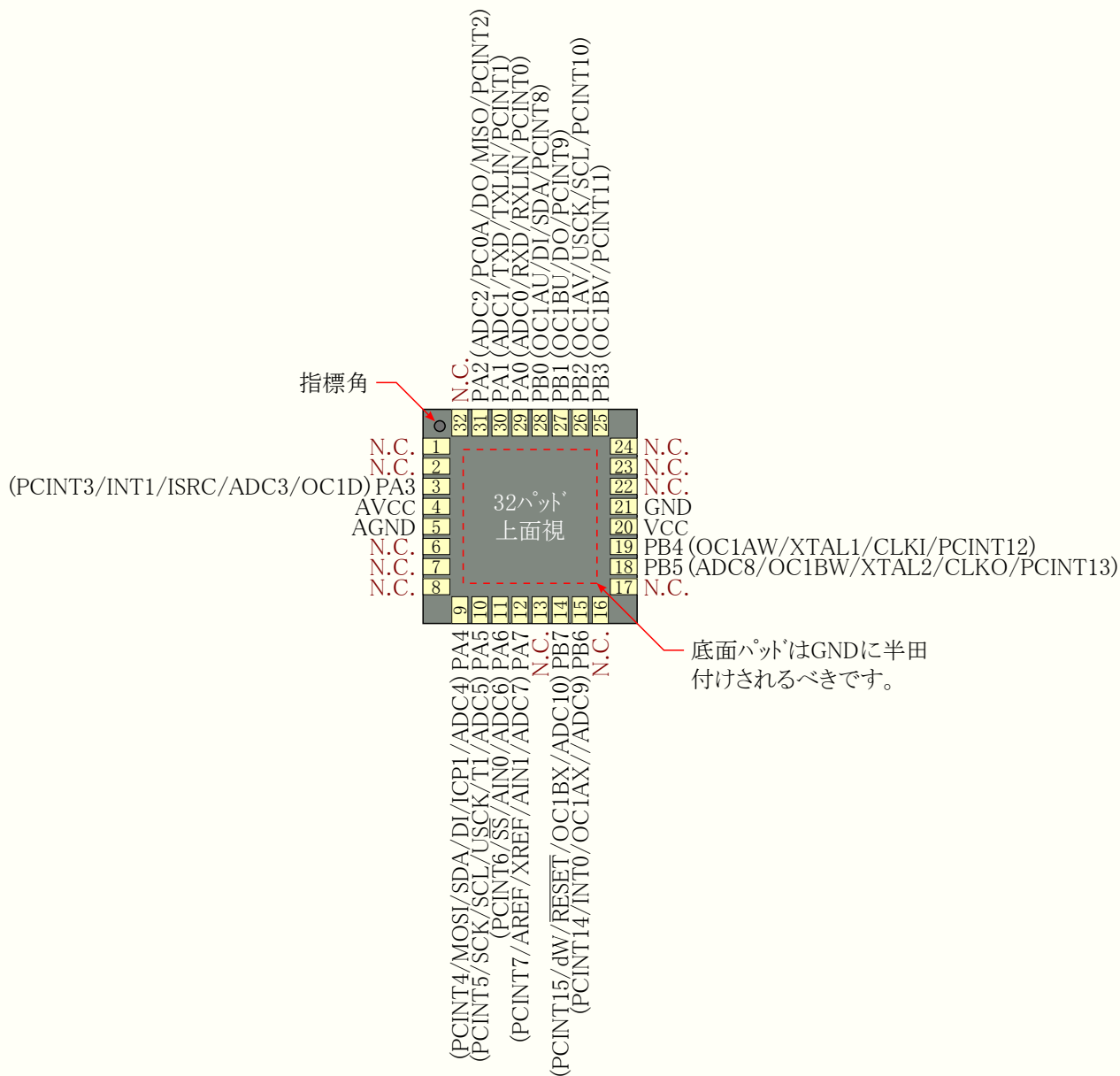
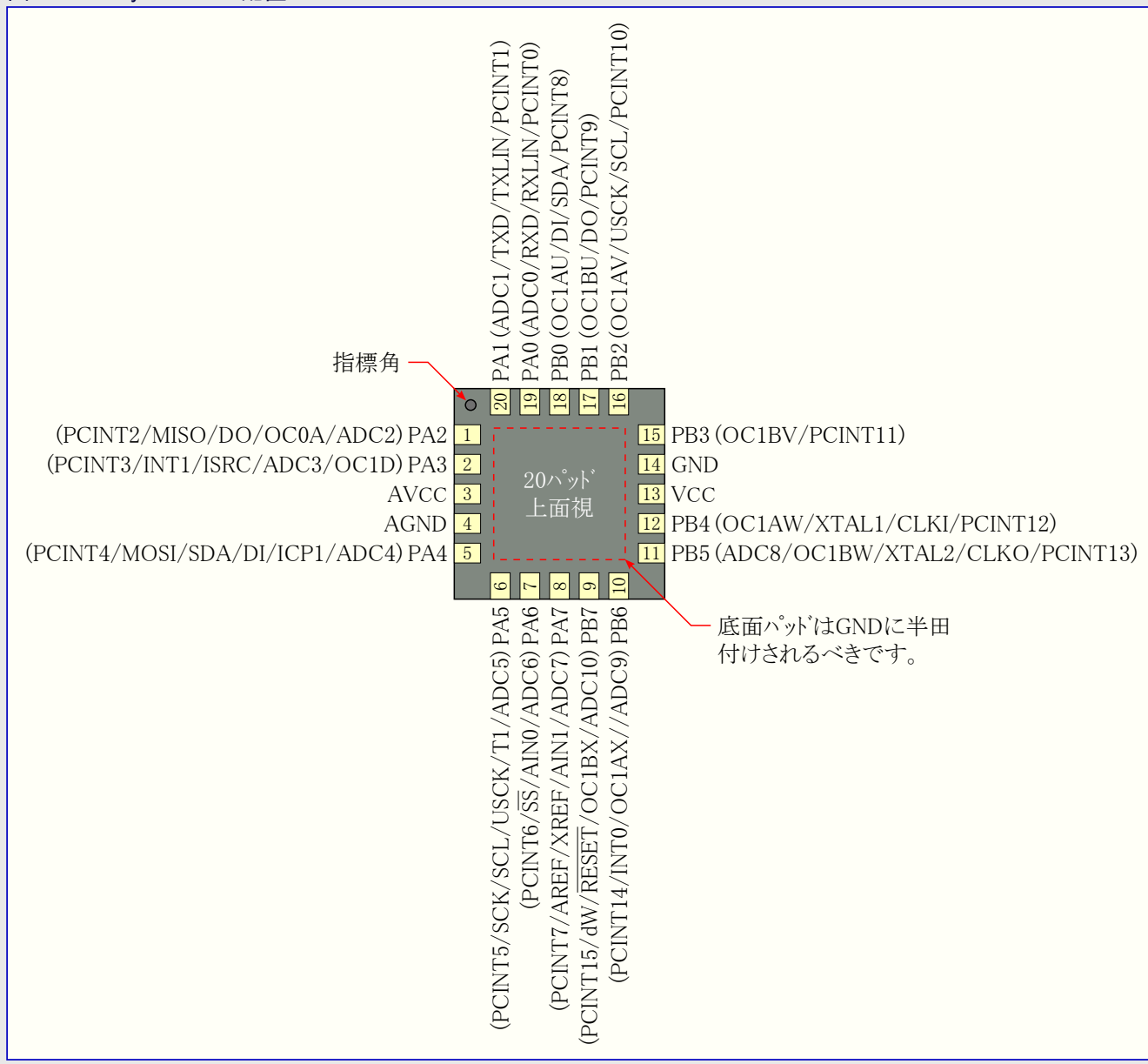


図1-4. ATtiny87/167ピン配置 - WQFN20



1.5. ピン説明

1.5.1. VCC

電源ピン。

1.5.2. GND

接地ピン。

1.5.3. AVCC

アナログ電源ピン。

1.5.4. AGND

アナログ接地ピン。

1.5.5. PA7～PA0 (ポートA)

ポートAは(ビット単位で選択される)内蔵プルアップ抵抗付きの8ビット双方向入出力ポートです。ポートA出力緩衝部は共に高い吐き出し/吸い込み能力の対称駆動特性です。入力の際にプルアップ抵抗が有効の場合、外部的にLowへ引き込まれたポートAピンには吐き出し電流が流れます。リセット条件が有効になると、クロックが走行していなくても、ポートAピンはHi-Zにされます。

ポートAは55頁の「9.3.3. ポートAの交換機能」で一覧されるようにATtiny87/167の様々な特殊機能も扱います。

1.5.6. PB7～PB0 (ポートB)

ポートBは(ビット単位で選択される)内蔵プルアップ抵抗付きの8ビット双方向入出力ポートです。ポートB出力緩衝部は共に高い吐き出し/吸い込み能力の対称駆動特性です。入力の際にプルアップ抵抗が有効の場合、外部的にLowへ引き込まれたポートBピンには吐き出し電流が流れます。リセット条件が有効になると、クロックが走行していなくても、ポートBピンはHi-Zにされます。

ポートBは59頁の「9.3.4. ポートBの交換機能」で一覧されるようにATtiny87/167の様々な特殊機能も扱います。

1.6. 資料

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

1.7. コード例について

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

1.8. データ保持力

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

1.9. お断り

本データシート内で示された代表値はシミュレーションと同じ製法技術で製造された他のAVRマイクロコントローラの特性に基いています。最小と最大の値はデバイスの特性が記載された後に利用可能になります。

2. AVR CPU コア

2.1. 概要

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。

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、本項内で後述されます。

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

プログラムの流れは条件/無条件分岐や呼び出し命令によって提供され、全アドレス空間を直接アドレス指定できます。殆どのAVR命令は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。

割り込みやサブルーチン呼び出し中、戻りアドレスを示すプログラムカウンタ(PC)はスタックに保存されます。スタックは一般的なデータ用SRAM上に実際には割り当てられ、従ってスタック容量は全SRAM容量とSRAM使用量でのみ制限されます。全てのユーザープログラムはリセット処理ルーチンで(サブルーチン呼び出しや割り込みが実行される前に)、スタックポインタ(SP)を初期化しなければなりません。SPはI/O空間で読み書きアクセスが可能です。データ用SRAMはAVR構造で支援される5つの異なるアドレス指定種別を通して容易にアクセスできます。

AVR構造に於けるメモリ空間は全て直線的な普通のメモリ配置です。

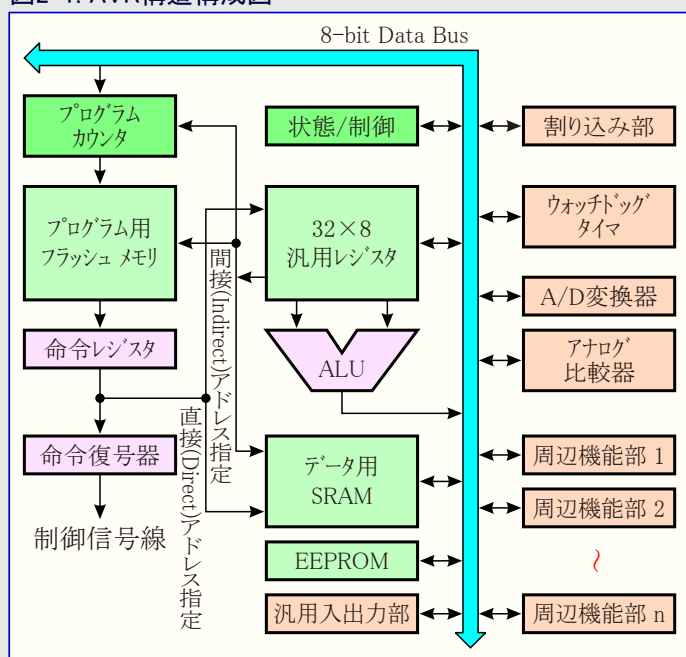
柔軟な割り込み部にはI/O空間の各制御レジスタとステータスレジスタ(SREG)の特別な全割り込み許可(I)ビットがあります。全ての割り込みは割り込みベクタ表に個別の割り込みベクタを持ちます。割り込みには割り込みベクタ表の位置に従う優先順があります。下位側割り込みベクタアドレスが高い優先順位です。

I/Oメモリ空間は制御レジスタ、SPI、他のI/O機能としてCPU周辺機能用の64アドレスを含みます。I/Oメモリは直接またはレジスタファイルの次のデータ空間位置\$20～\$5Fとしてアクセスできます。加えてATtiny87/167にはST/STS/STDとLD/LDS/LDD命令だけ使えるSRAM内の\$60～\$FFに拡張I/O空間があります(訳注:共通性から本行追加)。

2.2. 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2-1. AVR構造構成図



2.3. ステータス レジスタ

ステータスレジスタは最も直前に実行した演算命令の結果についての情報を含みます。この情報は条件処理を行うためのプログラムの流れ変更に使えます。ステータスレジスタは「[命令一式参考書](#)」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。

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

AVRのステータスレジスタ(SREG)は次のように定義されます。

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

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

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

● ビット6 – T: ビット変数 (Bit Copy Storage)

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

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

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

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

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

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

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

● ビット2 – N: 負フラグ (Negative Flag)

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

● ビット1 – Z: ゼロフラグ (Zero Flag)

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

● ビット0 – C: キャリー フラグ (Carry Flag)

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

2.4. 汎用レジスタ ファイル

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

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

図2-2.はCPU内の32個の汎用作業レジスタの構造を示します。

レジスタ ファイルを操作する殆どの命令は全てのレジスタに直接アクセスし、それらの殆どは単一周期命令です。

図2-2.で示されるように各レジスタは使用者データ空間の最初の32位置へ直接配置することで、それらはデータ メモリ アドレスも割り当てられます。例えば物理的にSRAM位置として実装されていなくても、X,Y,Zレジスタ(ポインタ)がレジスタ ファイル内のどのレジスタの指示にも設定できるように、このメモリ構成は非常に柔軟なレジスタのアクセスを提供します。

図2-2. AVR CPU 汎用レジスタ構成図

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

2.4.1. Xレジスタ, Yレジスタ, Zレジスタ

R26～R31レジスタには通常用途の使用にいくつかの追加機能があります。これらのレジスタはデータ空間の間接アドレス指定用の16ビットアドレス ポインタです。3つのX,Y,Z間接アドレス レジスタは図2-3.で記載したように定義されます。

種々のアドレス指定種別で、これらのアドレスレジスタは固定変位、自動増加、自動減少としての機能を持ちます(詳細については「命令一式参考書」をご覧ください)。

図2-3. X,Y,Zレジスタ構成図

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

2.5. スタック ポインタ

スタックは主に一時データの保存、局所変数の保存、割り込みとサブルーチン呼び出し後の戻りアドレスの保存に使われます。スタック ポインタレジスタは常にこのスタックの先頭(訳注:次に使われるべき位置)を指し示します。スタックが高位メモリから低位メモリへ伸長するように実行されることに注意してください。これはスタックへのPUSH命令はスタック ポインタを減少するという意味です。

スタック ポインタはサブルーチンや割り込みのスタックが配置されるデータSRAMのスタック領域を指し示します。データSRAM内のスタック空間はサブルーチン呼び出しの実行や割り込みの許可の何れにも先立って、プログラムによって定義されなければなりません。スタック ポインタは\$60以上を指示するように設定しなければなりません。スタック ポインタはPUSH命令でデータがスタックに格納されると-1され、サブルーチン呼び出しや割り込みで戻りアドレスがスタックに格納されると-2されます。スタック ポインタはPOP命令でデータがスタックから引き出されると+1され、サブルーチンからの復帰(RET)命令や割り込みからの復帰(RETI)命令でアドレスがスタックから引き出されると+2されます。

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

2.5.1. SPH,SPL (SP) – スタック ポインタ (Stack Pointer)

ビット	15	14	13	12	11	10	9	8	
\$3E (\$5E)	–	–	–	–	–	–	SP9	SP8	SPH
Read/Write	R	R	R	R	R	R	R/W	R/W	
初期値	ISRAM end (15頁の表3-1.をご覧ください。)								
ビット	7	6	5	4	3	2	1	0	
\$3D (\$5D)	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	SPL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	ISRAM end (15頁の表3-1.をご覧ください。)								

(訳補) 内蔵SRAMは512バイト(\$0100～\$02FF)ですので、ISRAM endは\$02FF(0000 0010 1111 1111)です。

2.6. 命令実行タイミング

本項は命令実行の一般的なアクセスタイミングの概念を記述します。AVR CPUはチップ(デバイス)用に選択したクロック元から直接的に生成したCPUクロック(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2-4.はハーバート構造と高速アクセスレジスタファイルの概念によって可能とされる並列の命令取得と命令実行を示します。これは機能対費用、機能対クロック、機能対電源部に関する好結果と対応するMHzあたり1MIPSを達成するための基本的なパイプラインの概念です。

図2-5.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2-4. 命令の取得と実行の並列動作

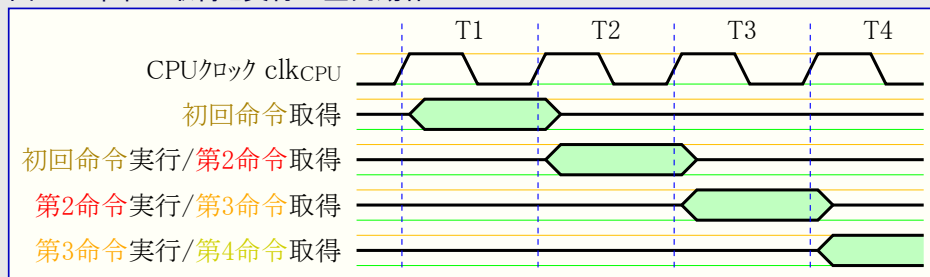
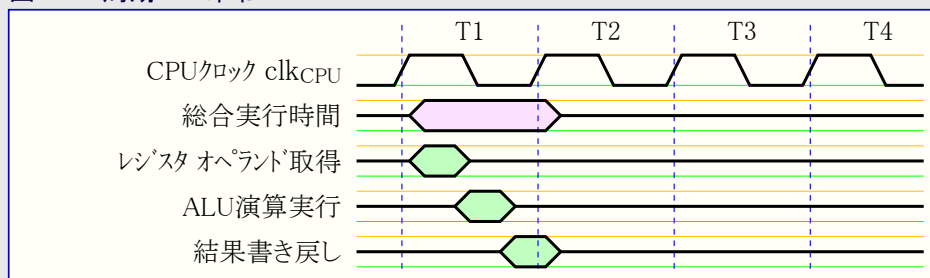


図2-5. 1周期ALU命令



2.7. リセットと割り込みの扱い

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

既定でのプログラムメモリ空間の最下位アドレスはリセットと割り込みベクタとして定義されます。ベクタの完全な一覧は45頁の「7. 割り込み」章で示されます。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。

2.7.1. 割り込みの動き

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

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

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

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

ステータスレジスタ(SREG)は割り込みルーチンへ移行時の保存も、復帰時の再設定も自動的に行われない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。

2.7.2. 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2/3クロック周期(**訳注:**原文は3(JMP命令=3)を想定、**RJMP**命令の場合は2)要します。複数周期命令実行中に割り込みが起こると、その割り込みが扱われる前に、この命令が完了されます。MCUが休止形態の時に割り込みが起こると、割り込み実行応答時間は4クロック周期増やされます。この増加は選択した休止形態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。

3. AVRのメモリ

この項はATtiny87/167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。加えてこれらのデバイスはデータ保存用EEPROMメモリが特徴です。3つのメモリ空間全ては一般的な直線的アドレスです。

表3-1. メモリ割付

メモリ種別		ニモニック	ATtiny87	ATtiny167
フラッシュメモリ	容量	Flash size	8Kバイト	16Kバイト
	開始アドレス	–	\$00000	
	終了アドレス	Flash end	\$1FFF(注1), \$0FFF(注2)	\$3FFF(注1), \$1FFF(注2)
汎用レジスタ	容量	–	32バイト	
	開始アドレス	–	\$0000	
	終了アドレス	–	\$001F	
(標準)I/Oレジスタ	容量	–	64バイト	
	開始アドレス	–	\$0020	
	終了アドレス	–	\$005F	
拡張I/Oレジスタ	容量	–	160バイト	
	開始アドレス	–	\$0060	
	終了アドレス	–	\$00FF	
内蔵SRAM	容量	ISRAM size	512バイト	
	開始アドレス	ISRAM start	\$0100	
	終了アドレス	ISRAM end	\$02FF	
内蔵EEPROM	容量	E2 size	512バイト	
	開始アドレス	–	\$0000	
	終了アドレス	E2 end	\$01FF	

注1: バイトアドレスです。

注2: 語(ワード=16ビット)アドレスです。

3.1. 実装書き換え可能なプログラム用フラッシュメモリ

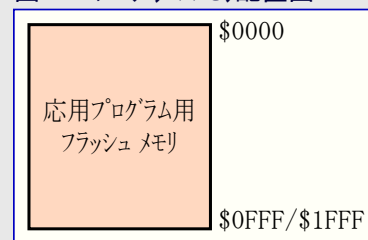
ATtiny87/167はプログラム保存用に実装書き換え可能な8/16Kバイトのフラッシュメモリをチップ上に含みます。全てのAVR命令が16または32ビット幅のため、このフラッシュメモリは4/8K×16ビットとして構成されます。ATtiny87/167は独立したブートプログラム領域と応用プログラム領域を持たず、SPM命令はフラッシュメモリ全体で実行できます。より多くの詳細については148頁の「20.2.1. SPM命令制御/状態レジスタ」項のSPM操作許可(SPMEN)の記述をご覧ください。

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。ATtiny87/167のプログラムカウンタ(PC)は12/13ビット幅で、従って4/8Kプログラムメモリ位置のアドレス指定です。152頁の「21. メモリプログラミング」章はSPIまたは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とSPM命令記述参照)

命令の取得と実行のタイミング図は13頁の「2.6. 命令実行タイミング」項で示されます。

図3-1. プログラムメモリ配置図



3.2. データ用SRAM メモリ

図3-2.はATtiny87/167のSRAMメモリ構成方法を示します。

ATtiny87/167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(15頁の表3-1.で”ISRAM size”をご覧ください)。

直接、間接、変位付き間接、事前減少付き間接、事後増加付き間接の5つの異なるアドレス指定種別でデータメモリ(空間)を網羅します。レジスタファイル内のレジスタR26～R31は間接アドレス指定ポインタ用レジスタが特徴です。

直接アドレス指定はデータ空間全体に届きます。

変位付き間接動作はYまたはZレジスタで与えられる基準アドレスからの63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。

ATtiny87/167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12頁の「汎用レジスタファイル」で記述されます。

3.2.1. データメモリアクセスタイミング

本節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3-3.で記載されるように2clkCPU周期で実行されます。

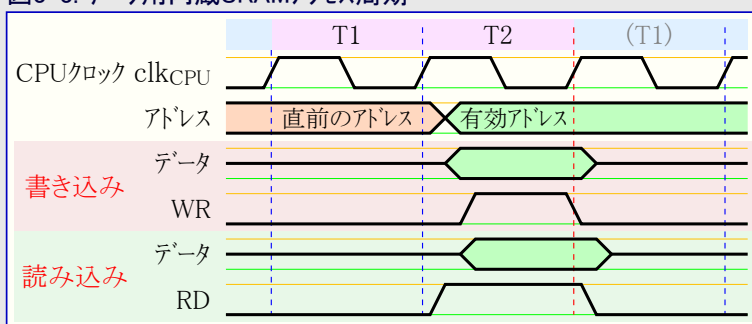
(訳注) 内蔵SRAMのアクセスを含む代表的な命令はT1,T2の2周期で実行され、T1で対象アドレスを取得/(算出)/確定し、T2で実際のアクセスが行われます。後続する(T1)は次の命令のT1です。

図3-2. データメモリ配置図

		アドレス
レジスタファイル (32×8)	R0	\$0000
	～ R31	～ \$001F
I/Oレジスタ (64×8)	\$00	\$0020
	～ \$3F	～ \$005F
拡張I/Oレジスタ (160×8)	\$0060	\$0060
	～ \$00FF	～ \$00FF
内蔵SRAM (512×8)	\$0100	\$0100
	～ \$0xFF	～ \$02FF

注: 赤字はI/Oアドレス

図3-3. データ用内蔵SRAMアクセス周期



3.3. データ用EEPROMメモリ

ATtiny87/167は64バイトのデータ用EEPROMを含みます(15頁の表3-1.で"E2 size"をご覧ください)。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。

152頁の「21. メモリプログラミング」章はSPIまたは並列プログラミングの詳細な記述を含みます。

3.3.1. 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の消去/書き込み(訳注:原文はアクセス)時間は表3-2.で与えられます。(書き込みは)自己タイミング機能ですが、使用者ソフトウェアは**次バイトが書ける時を検知**してください。使用者コードがEEPROMに書く命令を含む場合、いくつかの予防処置が取られねばなりません。嚴重に濾波した電源では電源投入/切断でVCCが緩やかに上昇または下降しそうです。これはデバイスが何周期かの時間、使われるクロック周波数に於いて最小として示されるより低い電圧で走行する原因になります。これらの状態で問題を避ける方法の詳細については以下の「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次の「**非分離バイト書き込み**」と「**分離バイト書き込み**」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。

3.3.2. 非分離バイトプログラミング

非分離バイトプログラミングの使用は最も簡単な形態です。EEPROMにバイトを書くとき、使用者はEEARにアドレス、EEDRにデータを書かなければなりません。EEPmnビットが'00'ならば、(EEMPEが1を書かれる後の4周期内の)EEPEの1書き込みは消去/書き込み動作を起動します。消去と書き込みの両周期は1操作で行われ、総プログラミング時間は表3-2.で与えられます。EEPEビットは消去と書き込み動作が完了されるまで設定(1)に留まります。デバイスがプログラミング動作中、他のどのEEPROM操作の実行も不可能です。

3.3.3. 分離バイトプログラミング

2つの異なる操作として消去と書き込みの周期を分離することが可能です。これは或る時間制限(代表的には電源電圧不足)に対してシステムが短いアクセス時間を必要とする場合に有用かもしれません。この方法の優位性を得るため、書かれるべき位置が書き込み操作前に消去されてしまっていることが必要とされます。しかし、消去と書き込みが分離されるため、時間が重大な操作の実行をシステムが許す時(代表的には電源投入後)に消去操作を行うことが可能です。

3.3.4. 消去

バイトを消去するにはアドレスがEEARに書かれなければなりません。EEPmnビットが'01'なら、(EEMPEが1を書かれた後の4周期内の)EEPEの1書き込みは消去動作だけを起動します(プログラミング時間は表3-2.で与えられます)。EEPEビットは消去動作が完了されるまで設定(1)に留まります。デバイスがプログラミング動作中、他のどのEEPROM操作の実行も不可能です。

3.3.5. 書き込み

(特定)位置を書くため、使用者はEEARにアドレス、EEDRにデータを書かなければなりません。EEPmnビットが'10'なら、(EEMPEが1を書かれる後の4周期内の)EEPEの1書き込みは書き込み動作だけを起動します(プログラミング時間は表3-2.で与えられます)。EEPEビットは書き込み動作が完了されるまで設定(1)に留まります。書かれるべき位置が書き込み前に消去されていなければ、元の格納データは失ったとみなされなければなりません。デバイスがプログラミング動作中、他のどのEEPROM操作の実行も不可能です。

EEPROMアクセスの時間に校正済み内蔵発振器が使われます。発振器周波数が31頁の「OSCCAL – 発振校正レジスタ」で記述した必要条件内であることを確かめてください。

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。(訳注:共通性から以下の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 */
}
```

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 */
}
```

3.3.6. EEPROMデータ化けの防止

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

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

EEPROMデータ化けは次の推奨設計によって容易に避けられます。

不十分な供給電源電圧の期間中、AVRのRESETを活性(Low)に保ってください。これは内蔵低電圧検出器(BOD)を許可することによって行えます。内蔵BODの検出電圧が必要とした検出電圧と一致しない場合、外部低VCCリセット保護回路が使えます。書き込み動作実行中にリセットが起ると、この書き込み操作は供給電源電圧が充分ならば(継続)完了されます。

3.4. I/O メモリ (レジスタ)

ATtiny87/167のI/O空間定義は182頁の「24. レジスタ要約」章で示されます。

ATtiny87/167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。ATtiny87/167はINやOUT命令で予約した64位置で支援されるよりも多くの周辺機能部を持つ複合マイクロコントローラです。SRAM(データ空間)内\$60～\$FFの拡張I/O領域に対してはLD/LDS/LDDとST/STS/STD命令だけが使えます。

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

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

I/Oと周辺制御レジスタは以降の項で説明されます。

3.4.1. 汎用I/Oレジスタ

ATtiny87/167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。

3.5. メモリ関係レジスタ

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

ビット	15	14	13	12	11	10	9	8	
\$22 (\$42)	–	–	–	–	–	–	–	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として読めます。

● 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256バイトEEPROM空間(15頁の表3-1.の”E2 size”参照)のEEPROMアドレスを指定します。EEPROMデータバイトは0～255間で直線的に配されます。EEARの初期値は不定です。EEPROMがアクセスされるであろう前に適切な値が書かれねばなりません。

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

ビット	7	6	5	4	3	2	1	0	
\$20 (\$40)	(MSB)							(LSB)	EEDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

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

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

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

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

● ビット7,6 – Res : 予約 (Reserved)

これらのビットは将来の使用に予約されており、常に0として読めます。読み込み後、これらのビットを遮蔽してください。将来のAVRデバイスとの互換性のため、常にこれらのビットに0を書いてください。

●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3-2.で示されます。EEPEが設定(1)されている間はEEPMMnへのどの書き込みも無視されます。リセット中、EEPMMnビットはEEPROMがプログラミング作業中を除いて'00'にリセットされます。

表3-2. EEPROMプログラミング種別

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

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

EERIEの1書き込みはステータスレジスタ(SREG)の全割り込み許可(I)ビットが設定(1)されているなら、EEPROM操作可割り込みを許可します。EERIEの0書き込みは、この割り込みを禁止します。EEPROM操作可割り込みは不揮発性メモリ(フラッシュメモリとEEPROM)がプログラミングの準備可ならば継続する割り込みを発生します。EEPROM書き込みとSPM命令の間、本割り込みは生成されません。

●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)は選択したアドレスの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。

●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。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。EEPEが設定(1)されてしまうと、次の命令が実行される前にCPUは2周期停止されます。

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

EEPROM読み込み許可信号(EERE)はEEPROMへの読み込みストローブです。EERE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。

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

ビット	7	6	5	4	3	2	1	0	
\$2B (\$4B)	(MSB)							(LSB)	GPIOR2
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

このレジスタはどんな種類のデータの格納にも自由に使えるかもしれません。

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

ビット	7	6	5	4	3	2	1	0	
\$2A (\$4A)	(MSB)							(LSB)	GPIOR1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

このレジスタはどんな種類のデータの格納にも自由に使えるかもしれません。

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

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

4. システム クロックとクロック選択

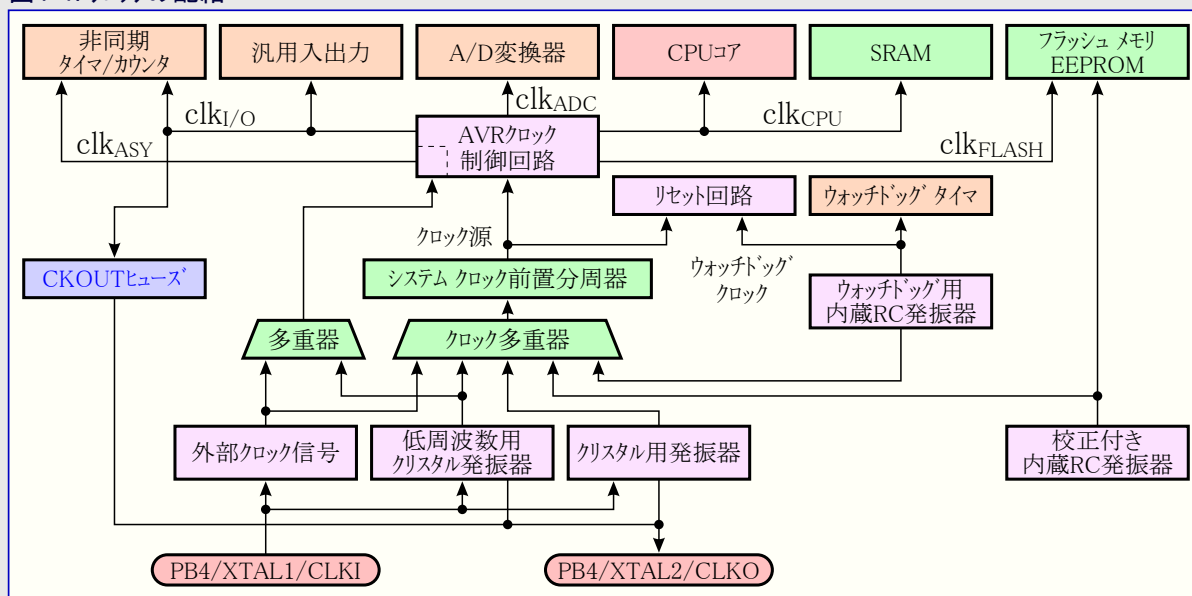
ATtiny87/167は多数のクロック元を提供します。それらは内部と外部の2つの種類に分けることができます。いくつかの外部クロック元は非同期タイマ/カウンタと共用され得ます。リセット後、クロック元は**CKSELヒューズ**によって決められます。一旦デバイスが走行すると、ソフトウェアクロック切り替えは他のどのクロック元にもできます。

クロック切り替え管理のためにハードウェア制御が提供されますが、いくつかの特別な手続きを遵守しなければなりません。クロック切り替えはデバイスが不正な構成設定を持つ結果となり得るいくつかの設定に注意して実行されるべきです。

4.1. クロックシステムとその配給

図4-1はAVR内の主要なクロックシステム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、各種休止形態の使用または動的クロック切り替え回路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(34頁の「電力管理と休止形態」と26頁の「動的クロック切り替え」をご覧ください)。クロックシステムは以下で詳細に示されます。

図4-1. クロックの配給



4.1.1. 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。

4.1.2. I/O クロック – clk_{I/O}

I/Oクロックは同期タイマ/カウンタのようなI/O部の大部分で使われます。I/Oクロックは外部割り込み部でも使われますが、いくつかの外部割り込みは例え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。

4.1.3. 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。

4.1.4. 非同期タイマ クロック – clk_{ASY}

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

4.1.5. A/D変換クロック – clk_{ADC}

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

4.2. クロック元

このデバイスは下で示されるように、フラッシュヒューズビット(既定)、または(動的クロック切り替え回路)クロック選択レジスタ(CLKSELR)によって選択可能な以下のクロック元選択を持ちます。選択した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4-1. デバイス クロック任意選択(注1)、PB4とPB5の機能

クロック種別	CKSEL3~0 (注2) CSEL3~0 (注3)	PB4	PB5
外部クロック信号	0 0 0 0	CLKI	CLKO・I/O
(予約)	0 0 0 1	–	–
校正付き8MHz内部RC発振器	0 0 1 0	I/O	CLKO・I/O
128kHz内部RC発振器(ウォッチドッグ発振器)	0 0 1 1	I/O	CLKO・I/O
外部低周波数水晶/セラミック用発振器	0 1 x x	XTAL1	XTAL2
外部水晶/セラミック用発振器(0.4~0.9MHz)	1 0 0 x	XTAL1	XTAL2
外部水晶/セラミック用発振器(0.9~3.0MHz)	1 0 1 x	XTAL1	XTAL2
外部水晶/セラミック用発振器(3.0~8.0MHz)	1 1 0 x	XTAL1	XTAL2
外部水晶/セラミック用発振器(8.0~16.0MHz)	1 1 1 x	XTAL1	XTAL2

注1: 全てのヒューズは1=非プログラム、0=プログラムです。

注2: フラッシュのヒューズビットです。

注3: クロック選択レジスタ(CLKSELR)のビットです。

各クロック選択に対する各種選択は以降の項で得られます。

CPUがパワーダウんまたはパワーセーブから起きる時か、または動的クロック切り替え回路によって新しいクロック元が許可される時に、選択したクロック元が始動時間に使われ、命令実行開始前に安定な発振器動作を保証します。

CPUがリセットから始まる時は通常動作開始前に安定電圧へ達することを電源に許す付加遅延があります。ウォッチドッグ発振器はこの起動時間の実時間部のタイミングに使われます。各計時完了に使われるウォッチドッグ(WDT)発振器の周期数は、表4-2.で示されます。

表4-2. ウォッチドッグ発振器周期数

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

4.2.1. 既定のクロック元

リセットに於いてCKSELヒューズとSUTヒューズがクロック選択レジスタ(CLKSELR)に複写されます。そしてデバイスはCLKSELRレジスタ(クロック元選択(CSEL3~0))とクロック始動時間(CSUT1,0))によって定義されたクロック元と始動タイミングを使います。

本デバイスはCKSELヒューズ=0010, SUTヒューズ=10, CKDIV8ヒューズ=プログラム(0)で出荷されます。従って既定のクロック元設定は最長起動時間、8.0MHzで走行する内蔵RC発振器で、8分周の初期システムクロックです。この既定設定は全ての使用者が実装または高電圧書き込み器を使い、それらを希望したクロック元設定にできることを保証します。この構成設定はISPツール使用時に注意しなければなりません。

4.2.2. 校正付き内蔵RC発振器

既定による内蔵RC発振器は概ね8.0MHzまたは1MHzのクロックを提供します。電圧と温度に依存しますが、このクロックは使用者によって高精度な校正ができます。より多くの詳細については165頁の表22-1と180頁の「23.10. 内部発振器」項をご覧ください。

選択したなら、外部部品なしで動作します。リセット中、ハードウェアが発振校正(OSCCAL)レジスタに校正値バイトを設定し、これによってRC発振器を自動的に構成設定(校正)します。この校正の精度は161頁の表22-1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22-1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とき、ウォッチドッグ発振器は未だウォッチドッグ タイムとリセット付加遅延タイムに使われます。予め設定された校正値のより多くの情報については154頁の「校正バイト」項をご覧ください。

この発振器が選択されると、起動時間は表4-4で示されるようにSUTヒューズまたはクロック始動時間(CSUT)領域によって決定されます。

表4-3. 校正付き内蔵RC発振器動作種別(注1)

CKSEL3~0 (注3,4)	周波数範囲 (注2)
CSEL3~0 (注5)	(MHz)
0 0 1 0	7.6~8.4

注1: 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。

注2: この周波数範囲は指針値です。

注3: デバイスはこのCKSEL3~0='0010'で出荷されます。

注4: フラッシュのヒューズ ビットです。

注5: クロック選択レジスタ(CLKSELR)のビットです。

表4-4. 校正付き内蔵RC発振器用起動遅延時間選択表

SUT1,0 (注1) CSUT1,0 (注2)	パワーダウンからの 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 (注3)	6×CK	14×CK	低電圧検出リセット(BOD)許可
0 1	6×CK	14×CK+4ms	高速上昇電源
1 0 (注4)	6×CK	14×CK+64ms	低速上昇電源
1 1	(予約)		

注1: フラッシュのヒューズ ビットです。

注2: クロック選択レジスタ(CLKSELR)のビットです。

注3: この設定はRSTDISBLEヒューズが設定されていない場合にだけ利用可能です。

注4: デバイスはこの選択で出荷されます。

4.2.3. 128kHz内部発振器

128kHz内部発振器は128kHzのクロックを供給する低電力発振器です。この周波数は3V,25℃での公称値です。このクロックは表4-1で示されるようにCKSELヒューズまたはクロック元選択(CSEL)領域のプログラミング(設定)によってシステム クロックとして選択できます。

このクロック元が選択されると、起動時間は表4-5で示されるようにSUTヒューズまたはクロック始動時間(CSUT)領域によって決定されます。

表4-5. 128kHz内部発振器用起動遅延時間選択表

SUT1,0 (注1) CSUT1,0 (注2)	パワーダウンからの 起動遅延時間	リセットからの 付加遅延時間	推奨使用法
0 0 (注3)	6×CK	14×CK	低電圧検出(BOD)リセット許可
0 1	6×CK	14×CK+4ms	高速上昇電源
1 0	6×CK	14×CK+64ms	低速上昇電源
1 1	(予約)		

注1: フラッシュのヒューズ ビットです。

注2: クロック選択レジスタ(CLKSELR)のビットです。

注3: この設定はRSTDISBLEヒューズが設定されていない場合にだけ利用可能です。

4.2.4. クリスタル用発振器

XTAL1とXTAL2は図4-2.で示されるように、チップ上の発振器としての使用に設定できる反転増幅器の各々、入力と出力です。クリスタル発振子またはセラミック振動子のどちらでも使えます。

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4-6.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4-6.で示されるようにCKSEL3~1ヒューズまたはクロック元選択(CSEL3~1)領域によって選択されます。

CKSEL0ヒューズはSUT1,0ヒューズと、またはクロック元選択(CSEL0)領域はクロック始動時間(CSUT)領域と共に表4-7.で示されるように起動時間を選択します。

図4-2. クリスタル発振子接続図

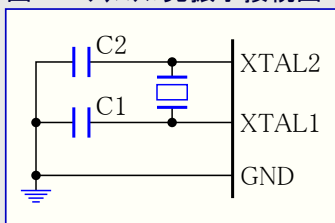


表4-6. クリスタル用発振器動作種別

CKSEL3~1 (注1) CSEL3~1 (注2)	周波数範囲	推奨C1,2容量
100 (注)	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1: フラッシュのヒューズビットです。

注2: クロック選択レジスタ(CLKSELR)のビットです。

注3: この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4-7. クリスタル発振子/セラミック振動子用発振器起動遅延時間選択表

CKSEL0 (注1) CSEL0 (注2)	SUT1,0 (注1) CSUT1,0 (注2)	パワーダウンからの 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3)	14 × CK+4ms	外部セラミック振動子、高速上昇電源
	0 1	258 × CK (注3)	14 × CK+64ms	外部セラミック振動子、低速上昇電源
	1 0 (注5)	1K × CK (注4)	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4)	14 × CK+4ms	外部セラミック振動子、高速上昇電源
1	0 0	1K × CK (注4)	14 × CK+64ms	外部セラミック振動子、低速上昇電源
	0 1 (注5)	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+4ms	外部クリスタル発振子、高速上昇電源
	1 1	16K × CK	14 × CK+64ms	外部クリスタル発振子、低速上昇電源

注1: フラッシュのヒューズビットです。

注2: クロック選択レジスタ(CLKSELR)のビットです。

注3: 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4: 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。

注5: この設定はRSTDISBLEヒューズが設定されていない場合にだけ利用可能です。

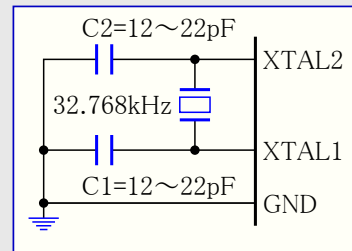
4.2.5. 低周波数クリスタル用発振器

デバイスに対するクロック元として時計用32.768kHzクリスタルを使うには、表4-1.で示されるようにCKSELヒューズまたはクロック元選択(CSEL)領域の設定によって低周波数クリスタル発振器が選択されなければなりません。クリスタルは図4-3.で示されるように接続されるべきです。32.768kHzクリスタル用の適当な負荷容量を探すには製造業者のデータシートを調べてください(訳注:整合性から本業修正)。

時計用32.768kHzクリスタル用発振器は(高周波数)クリスタル用発振器が走行していない、または外部クロックが許可されていない場合に非同期タイマ/カウンタによって使うことができます(27頁の「クロック元許可/禁止」をご覧ください)。非同期タイマ/カウンタはそれ自身でこの低周波数クリスタル用発振器を開始できます。

この発振器が選択されると、起動遅延時間は表4-8.で示されるようにSUTヒューズまたはクロック始動時間(CSUT)領域によって決定されます。

図4-3. クリスタル発振子接続図



寄生インピーダンス(パッド、線、基板)が非常に低い場合に12~22pFの容量が必要かもしれません。

表4-8. 低周波数クリスタル発振器起動遅延時間選択表

SUT1,0 (注1) CSUT1,0 (注2)	パワーダウンからの 起動遅延時間	リセットからの付加遅延 時間 (VCC=5.0V)	推奨使用法
0 0	1K × CK (注3)	4ms	高速上昇電源または低電圧検出(BOD)リセット許可
0 1	1K × CK (注3)	64ms	低速上昇電源
1 0	32K × CK	64ms	起動時の周波数安定
1 1	(予約)		

注1: フラッシュのヒューズビットです。

注2: クロック選択レジスタ(CLKSELR)のビットです。

注3: これらの選択は起動での周波数安定性が重要でない場合に使われるべきです。

4.2.6. 外部クロック信号

この外部クロック元からデバイスを駆動するには、CLKIが図4-4.で示されるように駆動されるべきです。外部クロックでデバイスを走行するには、表4-1.で示されるようにCKSELヒューズまたはクロック元選択(CSEL)領域がプログラム(設定)されなければなりません。

このクロック元が選択されると、起動時間は表4-9.で示されるようにSUTヒューズまたはクロック始動時間(CSUT)領域によって決定されます。

この外部クロックは高周波数または低周波数クリスタル用発振器が走行していない場合に非同期タイマ/カウンタによって使うことができます(27頁の「クロック元許可/禁止」をご覧ください)。そして非同期タイマ/カウンタはこの入力を許可することができます。

図4-4. 外部クロック信号駆動接続図

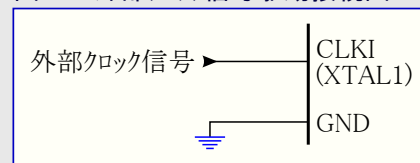


表4-9. 外部クロック信号駆動用起動遅延時間選択表

SUT1,0 (注1) CSUT1,0 (注2)	パワーダウンからの 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(+4ms) (注3)	低電圧検出(BOD)リセット許可
0 1	6 × CK	14 × CK+4ms	高速上昇電源
1 0	6 × CK	14 × CK+64ms	低速上昇電源
1 1	(予約)		

注1: フラッシュのヒューズビットです。

注2: クロック選択レジスタ(CLKSELR)のビットです。

注3: RSTDISBLEヒューズが設定されている場合は付加遅延(+4ms)が有効になります。

今までどおりの安定な動作を保証すると同時に内部クロック周波数の走行時変更を実装するのにシステムクロック前置分周器を使うことができることを覚えておいてください。詳細については「システムクロック前置分周器」を参照してください。

4.2.7. クロック出力緩衝部 (外部クロック出力)

クリスタル用発振器を使わない場合、デバイスはCLKOピンにシステムクロックを出力することができます。出力を許可するには、CKOUTヒューズまたはクロック選択レジスタ(CLKSELR)のクロック出力(COUT)ビットがプログラム(設定)されなければなりません。この形態はデバイスのクロックがシステム上の他の回路を駆動するのに必要とされる時に有用です。リセット中にクロックが出力されないのと、ヒューズがプログラム(0)されている時に入出力ピンの標準動作が無効にされることに注意してください。システムクロック前置分周器が使われると、分周されたシステムクロックが出力されます。

4.3. 動的クロック切り替え

4.3.1. 特徴

ATtiny87/167は進行中にデバイスのクロックのON/OFF切り替えを使用者に許す強力な動的クロック切り替えを提供します。組み込み整合回路がクロックに非同期での許可と禁止を許します。これは簡単且つ素早い実装のための効率的な電力管理の仕組みを許します。安全な応用では、動的クロック切り替え回路が外部クロックの継続監視を許し、クロック異常の場合に代替切り替えを実行します。

動的クロック切り替え回路の制御はソフトウェアによって管理されなければなりません。この操作は以下の特徴によって容易にされます。

- **安全な命令**：予期せぬ命令を避けるため、CLKCSRレジスタのビットを変更するには特別な手続きに従わなければなりません(32頁の「CLKCSR – クロック制御/状態レジスタ」をご覧ください)
- **排他的な活動**：活動はCLKCSRレジスタに書かれる復号(命令)表によって制御されます。これは1つの命令操作だけが同時に発行され得ることを保証します。復号表の主な活動は次のとおりです。
 - ‘クロック元禁止’
 - ‘クロック元許可’
 - ‘クロック有効性要求’
 - ‘クロック元切り替え’
 - ‘システム クロック元回復’
 - ‘自動再設定形態でのウォッチドッグ許可’
- **命令状況返し**：‘クロック有効性要求’命令はCLKCSRレジスタのクロック準備可(CLKRDY)ビット経由で状態を返します。‘システム クロック元回復’命令はクロック選択レジスタ(CLKSELR)で現在のクロック元の符号を返します。この情報は27頁の「クロック切り替え」で示されるように管理ソフトウェア ルーチンで使われます。

4.3.2. クロック選択レジスタ (CLKSELR)

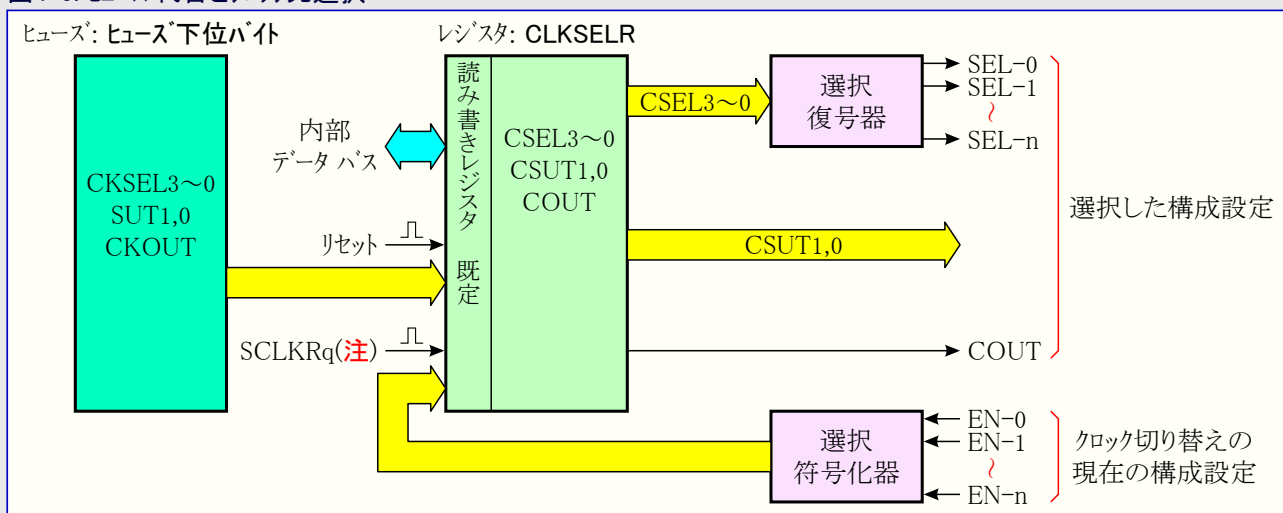
4.3.2.1. ヒューズ代替

リセット中、ヒューズ下位バイトのビットがクロック選択レジスタ(CLKSELR)に複写されます。その後、このレジスタの内容はヒューズ下位バイトからの既定値を上書きするように使用者変更することができます。図4-5.で示されるようにCKSEL3~0、SUT1,0、CKOUTのヒューズは各々CLKSELRレジスタのクロック元選択(CSEL3~0)、クロック始動時間(CSUT1,0)、クロック出力(COUT)に対応します。

4.3.2.2. クロック元選択

利用可能なクロック元の符号は22頁の表4-1.で与えられます。

図4-5. ヒューズ代替とクロック元選択



注: SCLKRqはクロック制御/状態レジスタの命令

CLKSELRレジスタは‘クロック元許可/禁止’、‘クロック有効性要求’または‘クロック元切り替え’の命令によって使われるクロック元選択(CSEL3~0)、クロック始動時間(CSUT1,0)、クロック出力(COUT)の値を含みます。

4.3.2.3. クロック元回復

‘システム クロック元回復’命令はCLKSELRレジスタのCKSEL領域を更新します(次頁の「システム クロック元回復」をご覧ください)。

4.3.3. クロック元許可/禁止

‘**クロック元許可**’命令は**クロック選択(CLKSELR)レジスタ**の設定によって構成設定されたクロック元を選択して許可します。(丁度CKSELとSUTのヒューズビットが行うように)**クロック元選択(CSEL3~0)**がクロック元を選び、**クロック始動時間(CSUT1,0)**が始動時間を選びます。クロック元が動いていることを保証するために、‘**クロック元許可**’命令の後で‘**クロック有効性要求**’命令が実行されなければなりません。これは有効なクロックが操作可能で利用可能なことを**クロック制御/状態レジスタ(CLKCSR)**の**クロック準備可(CLKRDY)ビット**経由で示します。

‘**クロック元禁止**’命令はCLKSELRレジスタ(CSEL3~0のみ)の設定によって示されるクロック元を禁止します。そのクロック元がシステムクロックを駆動するのに使われていたものなら、その命令は実行されません。

選択した構成設定がクロック供給元レベルでラッチされるため、与えられた時間で多くのクロック元を許可することが可能です(例:システムクロック用内蔵RC発振器+外部クリスタル用発振器)。使用者(コード)にはこの管理の責任があります。

4.3.4. CKOUT命令

‘**CKOUT**’命令はCLKOピンの駆動を許します。使用については25頁の「**4.2.7. クロック出力緩衝部 (外部クロック出力)**」を参照してください。

4.3.5. クロック有効性

‘**クロック有効性要求**’命令は選択したクロック元(CSEL3~0)によって駆動されるハードウェア発振周期計数器を許可します。計数限界値はCSUT1,0の設定によって決められます。クロックは計数限界値に達した時に準備可(**クロック準備可(CLKRDY)=1**)を宣言されます。CLKRDYフラグは計数開始時にリセットされます。一旦設定されると、このフラグは新しい計数が指示されるまで無変化に留まります。この検査を実行するために、CSELとCSUTの領域は操作が動作中の間、変更すべきではありません。

以下のように2つの使い方が可能です。

一旦新しいクロック元が選択される(‘**クロック元許可**’命令)と、計数手順が自動的に開始されることに注意してください。使用者(コード)は新しく選択するクロックを使う前に**クロック制御/状態レジスタ(CLKCSR)**のCLKRDYフラグの設定(**1**)を待つべきです。

何時でも、使用者(コード)はクロック元の有効性を問うことができます。使用者(コード)はCLKCSRレジスタに‘**クロック有効性要求**’命令を書くことによってそれを要求することができます。従ってクロック元の状態の完全なポーリングを行うことができます。

4.3.6. システムクロック元回復

‘**システムクロック元回復**’命令は**表4-1**に対するようにシステムクロックを駆動するのに用いる現在のクロック元を返します。その後に**クロック選択(CLKSELR)レジスタ**の**クロック元選択(CSEL3~0)**領域がこの返された値で更新されます。使われるSUTの情報やCKOUTの状態はありません。

4.3.7. クロック切り替え

システムクロックを駆動するために、使用者は現在のクロック元から以下の他のどの1つ(それらの1つが現在のクロック元です)にも切り替えることができます。

1. 8.0MHz校正付き内蔵RC発振器
2. ウォッチドッグ用128kHz内部発振器
3. 外部クロック信号
4. 外部低周波数クリスタル発振子
5. 外部クリスタル発振子/セラミック振動子

クロック切り替えは命令の流れで実行されます。最初に、使用者(コード)は新しいクロック元が動作中であることを保証しなければなりません。その後に‘**クロック元切り替え**’命令を発行することができます。‘**システムクロック元回復**’命令を用いて一旦この命令が成功裏に完了されると、使用者(コード)は直前のクロック元を停止することができます。

一旦割り込みが禁止されている場合にだけ、この手順を動かすことが強く推奨されます。使用者(コード)にはこのクロック切り替え手順の正しい実行の責任があります。

これはこのような命令の手順を記述する“軽い”C言語コードです。

C言語プログラム例

```
void ClockSwitching (unsigned char clk_number, unsigned char sut) {

#define CLOCK_RECOVER      0x05
#define CLOCK_ENABLE      0x02
#define CLOCK_SWITCH      0x04
#define CLOCK_DISABLE     0x01

    unsigned char previous_clk, temp;

    // 全割り込み禁止
    asm ("cli"); temp = SREG;
    // 現在のシステム クロック元保存
    CLKCSR = 1 << CLKCCE;
    CLKCSR = CLOCK_RECOVER;
    previous_clk = CLKSELR & 0x0F;
    // “クロック元許可”命令
    CLKSELR = ((sut << 4) & 0x30) | (clk_number & 0x0F);
    CLKCSR = 1 << CLKCCE;
    CLKCSR = CLOCK_ENABLE;
    // クロック有効性待ち
    while ((CLKCSR & (1 << CLKRDY)) == 0);
    // “クロック元切り替え”命令
    CLKCSR = 1 << CLKCCE;
    CLKCSR = CLOCK_SWITCH;
    // 切り替え効果待ち
    while (1){
        CLKCSR = 1 << CLKCCE;
        CLKCSR = CLOCK_RECOVER;
        if ((CLKSELR & 0x0F) == (clk_number & 0x0F)) break;
    }
    // “クロック元禁止”命令
    CLKSELR = previous_clk;
    CLKCSR = 1 << CLKCCE;
    CLKCSR = CLOCK_DISABLE;
    // 全割り込み再許可
    SREG = temp; asm ("sei");
}
```

警告:

ATtiny87/167では、3つの外部クロック元の内の1つだけが与えられた時に許可することができます。また、外部クロックと外部低周波数発振器の許可は非同期タイマ/カウンタと共用されます。

4.3.8. クロック監視

安全なシステムはそのクロック元を監視する必要があります。以下の2つの領域が監視を必要とします。

- 周辺機能用クロック元
- システム クロック生成用クロック元

最初の領域で、使用者(コード)はクロックの有効性を容易に検査することができます(27頁の「COUT命令」をご覧ください)。

2つ目の領域で、クロックなしはコード走行なしに帰着します。故に、システム クロックの存在はハードウェアによって監視されることが必要です。

チップ上のウォッチドッグの使用がこの監視を許します。通常、ウォッチドッグの再設定はコードが或る特定のラベルに達する場合にだけ実行され、これらのラベルへの到達はシステム クロックが走行していることを証明します。さもなければ、ウォッチドッグ リセットが許可されます。この動きはクロック監視と見做すことができます。

標準的なウォッチドッグ機能が求められないなら、ATtiny87/167のウォッチドッグは完全なソフトウェア ウォッチドッグ処理部の複雑さに頼らずに、監視されることをシステム クロックに許します。ATtiny87/167で提案される解決策は作業の先頭に於いて1つの命令だけでウォッチドッグ再設定を自動化することです。

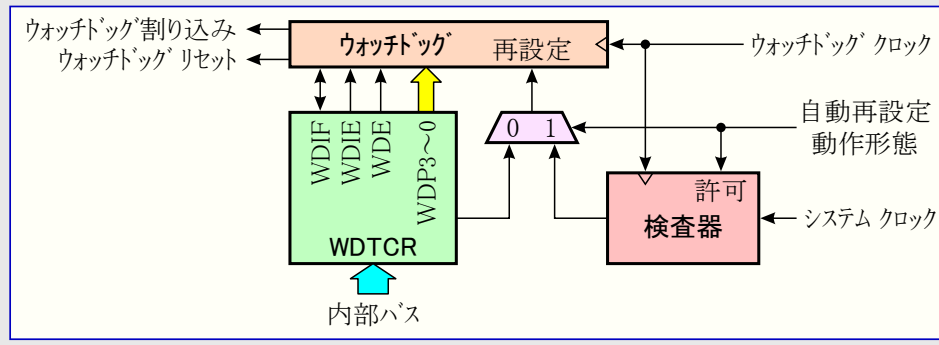
故に、システム クロックを監視するため、使用者は以下の2つの任意選択を持ちます。

1. 標準的なウォッチドッグ機能の使用(ソフトウェア再設定)
2. または、自動的な再設定(ハードウェア再設定)

この2つの任意選択は排他的です。

注: 警告: これら2つの任意選択はリセットでのクロック元が内部供給元の場合にだけ意味を成します。
ヒューズ設定がこの操作を決めます。

図4-6. 自動再設定付きウォッチドッグ タイマ



‘自動再設定形態でウォッチドッグ許可’命令は標準のウォッチドッグ許可を超える優先権を持ちます。この動作形態ではウォッチドッグのリセット機能だけが許可されます(もはやウォッチドッグ割り込みはありません)。ウォッチドッグ制御レジスタ(WDTCR)のウォッチドッグ タイマ前置分周選択(WDP3~0)ビットが常にウォッチドッグ タイマの前置分周を決めます。

‘自動再設定形態でウォッチドッグ許可’命令を実行する前にウォッチドッグが活性でないとき、外部クロック元へ切り替える前にこの命令を活性にすることが推奨されます(上の警告をご覧ください)。

注: ・ (ウォッチドッグ リセットを含む)リセットだけがこの機能を禁止します。ウォッチドッグ システム リセット フラグ(MCU状態レジスタ(MCUSR)のWDRF)はリセット発生を監視することができます。

- ・ (ウォッチドッグ クロック周波数×4)と等しい、またはより大きなクロック周波数だけを監視することができます。

これは自動クロック監視を用いるクロック切り替え関数の“軽い”C言語コードです。

C言語プログラム例

```
void ClockSwitching (unsigned char clk_number, unsigned char sut) {

#define CLOCK_RECOVER                        0x05
#define CLOCK_ENABLE                        0x02
#define CLOCK_SWITCH                        0x04
#define CLOCK_DISABLE                       0x01
#define WD_ARL_ENABLE                       0x06

#define WD_2048CYCLES                        0x07

unsigned char previous_clk, temp;

// 割り込み禁止
    temp = SREG; asm ("cli");
// 現在のシステムクロック元保存
    CLKCSR = 1 << CLKCCE;
    CLKCSR = CLOCK_RECOVER;
    previous_clk = CLKSELR & 0x0F;
// 新しいクロック元許可
    CLKSELR = ((sut << 4) & 0x30) | (clk_number & 0x0F);
    CLKCSR = 1 << CLKCCE;
    CLKCSR = CLOCK_ENABLE;
// クロック有効待機
    while ((CLKCSR & (1 << CLKRDY)) == 0);
// 自動再設定動作形態でウォッチドッグ許可
    WDTCSR = (1 << WDCE) | (1 << WDE);
    WDTCSR = (1 << WDE) | WD_2048CYCLES;
    CLKCSR = 1 << CLKCCE;
    CLKCSR = WD_ARL_ENABLE;
// クロック元切り替え
    CLKCSR = 1 << CLKCCE;
    CLKCSR = CLOCK_SWITCH;
// 切り替え有効待機
    while (1) {
        CLKCSR = 1 << CLKCCE;
        CLKCSR = CLOCK_RECOVER;
        if ((CLKSELR & 0x0F) == (clk_number & 0x0F)) break;
    }
// 不要クロック元停止
    if (previous_clk != (clk_number & 0x0F)) {
        CLKSELR = previous_clk;
        CLKCSR = 1 << CLKCCE;
        CLKCSR = CLOCK_DISABLE;
    }
// 割り込み再許可
    SREG = temp;
}
```

4.4. システムクロック前置分周器

4.4.1. 特徴

ATtiny87/167の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することができます。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}、clk_{ADC}は表4-10.で示された係数によって分周されます。

4.4.2. 切り替え時間

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、クロック系で不整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プル計数器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も速いかもしれません。従って例え(計数器値が)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。

4.5. クロック関係レジスタ

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

ビット (\$66)	7	6	5	4	3	2	1	0	
	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)

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、165頁の表22-1.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OSCCALレジスタへ書くことができます。この発振器は表22-1.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。

この発振器が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よっ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動作の範囲を決めます。このビットの0設定は最低周波数範囲を与え、このビットの1設定は最高周波数範囲を与えます。この2つの周波数範囲は重なっていて、換言するとOSCCAL=\$7FはOSCCAL=\$80よりも高い周波数を与えます。

CAL6～0ビットは選択した範囲内の周波数を調整するのに使われます。\$00の設定はその範囲での最低周波数を与え、\$7Fがその範囲での最高周波数を与えます。CAL6～0の1増加は7.3～8.1MHzの周波数範囲で2%未満の周波数増加を与えます。

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

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

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

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

●ビット6～4 – Res : 予約 (Reserved)

これらのビットは予約されており、常に0として読まれます。

●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択したクロック元と内部システムクロック間の分周値を定義します。これらのビットは応用の必要条件に合わせた各種クロック周波数を実行時に書くことができます。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4-10.で与えられます。

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

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

前置分周器設定変更時、手順が邪魔されないために、割り込みは禁止されなければなりません。

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も高い周波数のクロック元を選択した場合に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も高い周波数のクロック元を選択した場合、充分な分周値が選択されることを保証しなければなりません。このデバイスはCKDIV8ヒューズがプログラム(0)で出荷されます。

表4-10. 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)						

4.5.3. CLKCSR – クロック制御/状態レジスタ (Clock Control & Status Register)

ビット (\$62)	7	6	5	4	3	2	1	0	
	CLKCCE	–	–	CLKRDY	CLKC3	CLKC2	CLKC1	CLKC0	CLKCSR
Read/Write	R/W	R	R	R	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

● ビット7 – CLKCCE : クロック制御変更許可 (Clock Control Change Enable)

CLKCSRのビットの変更を許可するにはCLKCCEビットが論理1を書かれなければなりません。CLKCCEビットはCLKCSRの他のビットが同時に0を書かれる時にだけ更新されます。CLKCCEはそれが書かれた4周期後、またはCLKCSRのビットが書かれた時にハードウェアによって解除(0)されます。この制限時間内のCLKCCEビットの再書き込みは制限時間の延長もCLKCCEビットの解除(0)もどちらも行いません。

● ビット6,5 – Res : 予約 (Reserved)

これらのビットは予約されており、常に0として読まれます。

● ビット4 – CLKRDY : クロック準備可フラグ (Clock Ready Flag)

このフラグは'クロック有効性'論理回路の出力です。このフラグは'クロック有効性要求'命令または'クロック元許可'命令に入ることによって解除(0)されます。

'クロック有効性'論理回路が(選択した)クロックが動いて安定であることを通知する時にこれが設定(1)されます。要求からの遅延とフラグ設定は一定ではなく、クロック始動時間、クロック周波数、そして勿論クロックが生きているかに依存します。「クロック信号なし」と「クロック信号が未だ有効ではない」との違いは使用者が行わなければなりません。

● ビット3~0 – CLKC3~0 : クロック制御 (Clock Control Bits 3~0)

これらのビットは'クロック切り替え'部へ提供する命令を定義します。CLKC3~0ビットを変更するには特別な書き込み手順に従わなければなりません(「**ビット7 – CLKCCE : クロック制御変更許可**」をご覧ください)。

1. クロック制御変更許可(CLKCCE)ビットに1を、そしてCLKCSRの他の全てのビットに0を書いてください。
2. 4周期内に、CLKCSRレジスタに望む値を書くと同時にCLKCCEビットを解除(0)してください。

この手順を邪魔しないために、CLKCSRレジスタ設定時に割り込みは禁止されるべきです。

表4-11. クロック命令一覧

CLKC3~0	クロック命令
0 0 0 0	(命令なし)
0 0 0 1	クロック元禁止
0 0 1 0	クロック元許可
0 0 1 1	クロック有効性要求
0 1 0 0	クロック元切り替え
0 1 0 1	システム クロック元回復
0 1 1 0	自動再設定動作形態でウォッチドッグ許可
0 1 1 1	CKOUT命令
1 x x x	(命令なし)

4.5.4. CLKSELR – クロック選択レジスタ (Clock Selection Register)

ビット (\$63)	7	6	5	4	3	2	1	0	
	–	COUT	CSUT1	CSUT0	CSEL3	CSEL2	CSEL1	CSEL0	CLKSELR
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	CKOUTヒューズ	SUT1,0ヒューズ		CKSEL3~0ヒューズ				

● ビット7 – Res : 予約 (Reserved)

このビットは予約されており、常に0として読まれます。

● ビット6 – COUT : クロック出力 (Clock Out)

COUTビットはCKOUTヒューズビットで初期化されます。COUTビットは'CKOUT'命令の場合にだけ使われます。使用については25頁の「**4.2.7. クロック出力緩衝部 (外部クロック出力)**」項を参照してください。'システム クロック元回復'命令の場合、COUTは影響を及ぼされません(この設定の回復なし)。

- ビット5,4 – CSUT1,0 : クロック始動時間 (Clock Start-up Time)

CSUTビットはSUTヒューズビットの値で初期化されます。'クロック元許可/禁止'命令の場合、CSUT領域はクロック始動時間の符号を提供します。クロック始動時間の符号については18頁の「4.2. クロック元」項の一部分を参照してください。'システムクロック元回復'命令の場合、CSUT領域は影響を及ぼされません(SUT符号の回復なし)。

- ビット3~0 – CSEL3~0 : クロック元選択 (Clock Source Select)

CSELビットはCKSELヒューズビットの値で初期化されます。'クロック元許可/禁止'、'クロック有効性要求'、または'クロック元切り替え'の命令の場合、CSEL領域はそのクロック元の符号を提供します。クロック元符号については22頁の表4-1.と22頁の「4.2. クロック元」項の一部分を参照してください。'システムクロック元回復'命令の場合、CSEL領域は21頁の図4-1.で記述されるようにクロック制御部を駆動するのに使われたクロック元の符号を含みます。

5. 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。

許可されると、低電圧検出器(BOD)は休止期間の間に電源電圧を活発に監視します。更なる節電のため、いくつかの休止形態でBODを禁止することが可能です。より多くの詳細については「[低電圧検出器\(BOD\)禁止](#)」をご覧ください。

5.1. 休止形態種別

21頁の図4-1はATtiny87/167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選択を助けます。表5-1は異なる休止形態とそれらの起動元とBOD禁止の可能性を示します。

表5-1. 各休止形態に於ける動作クロック範囲と復帰起動要因

休止種別	動作クロック範囲					動作発振器		復帰起動要因 (割り込み)							ソフトウェア BOD 禁止
	clkCPU	clkFLASH	clkIO	clkADC	clkASY	主クロック 供給元	タイマ0 発振器	INT1 INT0 ピン変化	USI 開始 条件	タイマ/ カウンタ0	SPM EEPROM 操作可	A/D 変換 完了	ウォッチ ドッグ	その他 I/O	
アイドル			○	○	○	○	○	○	○	○	○	○	○	○	
A/D変換雑音低減				○	○	○	○	①	○	○	○	○	○		
パワーダウン								①	○				○		○
パワーセーブ					○		○	①	○	○			○		○

①: INT1とINT0についてはレベル割り込みだけです。

4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(SM1,0)ビット**は**SLEEP**命令によって活性(有効)にされる休止形態(アイドル、A/D変換雑音低減、パワーダウン、パワーセーブ)のどれかを選びます。一覧については表5-2をご覧ください。

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

5.2. 低電圧検出器(BOD)禁止

低電圧検出器(BOD)が153頁の表21-4のBODLEVELヒューズによって許可されていると、BODは休止期間中に電源電圧を活発に監視します。節電のため、休止形態のいくつかに対してソフトウェアによってBODを禁止することが可能です。表5-1をご覧ください。この休止形態電力消費はBODがヒューズによって全面的に禁止される時と同じ水準になるでしょう。BODがソフトウェアで禁止される場合、BOD機能は休止形態移行後、直ちにOFFされます。休止からの起動復帰でBODは再び自動的に許可されます。これは休止期間中にVCCレベルが落ちた場合の安全な動作を保証します。

BODが禁止されてしまうと、MCUがコードの実行を継続する前にBODが正しく動作することを保証するために、休止形態からの起動時間は概ね60μsになります。

BOD禁止はMCU制御レジスタ(MCUCR)のビット6、**BOD休止(BODS)ビット**によって制御されます。37頁の「**MCUCR – MCU制御レジスタ**」をご覧ください。このビットへの1書き込みは関連する休止形態でのBODをOFFにし、一方このビットの0はBOD活動(有効)を保ちます。既定設定はBOD活動維持、換言するとBODSは0に解除されます。

BODSビットへの書き込みは許可ビットと時間制限手順によって制御されます。37頁の「**MCUCR – MCU制御レジスタ**」をご覧ください。

5.3. アイドル動作

休止種別選択(SM1,0)ビットが'00'を書かれるとき、**SLEEP**命令はMCUをアイドル動作へ移行させ、CPUを停止しますが、SPI、アナログ比較器、A/D変換器、USI開始条件、非同期タイマ/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SPIなどの内部割り込みだけでなく、外部で起動された割り込みからの起動も可能にします。アナログ比較器割り込みからの起動が必要とされないなら、**アナログ比較器制御/状態レジスタ(ACSR)のアナログ比較器禁止(ACD)ビット**を設定(1)することによってアナログ比較器を電源断にできます。これはアイドル動作での消費電力を削減します。A/D変換が許可されるなら、この動作に移行すると変換が自動的に始まります。

5.4. A/D変換雑音低減動作

休止種別選択(SM1,0)ビットが'01'を書かれるとき、**SLEEP**命令はMCUをA/D変換雑音低減動作へ移行させ、CPUを停止しますが、A/D変換器、外部割り込み、USI開始条件、非同期タイマ/カウンタ、ウォッチドッグの(許可されていれば)継続動作を許します。この休止形態は基本的にclkI/O、clkCPU、clkFLASHを停止する一方、他のクロックに走行を許します。

これはA/D変換に対する雑音環境を改善し、より高い分解能の測定を可能にします。A/D変換器が許可されている場合、この動作に移行すると、変換が自動的に始まります。A/D変換完了割り込みからの他、外部リセット、ウォッチドッグシステムリセット、ウォッチドッグ割り込み、低電圧検出(BOD)リセット、USI開始条件割り込み、非同期タイマ/カウンタの割り込み、SPM/EEPROM操作可割り込み、INT0またはINT1の外部レベル割り込み、ピン変化割り込みだけが、A/D変換雑音低減動作からMCUを起動できます。

5.5. パワーダウン動作

SM1,0ビットが'10'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、USI開始条件、ウォッチドッグ機能は(許可ならば)継続して動作します。外部リセット、ウォッチドッグシステムリセット、ウォッチドッグ割り込み、低電圧検出(BOD)リセット、USI開始条件割り込み、INT0またはINT1の外部レベル割り込み、ピン変化割り込みだけがMCUを起動できます。この休止形態は基本的に生成した全てのクロックを停止し、非同期部の動作だけを許します。

レベルで起動した割り込みがパワーダウン動作からの復帰に使われる場合、MCUを起動するためには変更したレベルが一定時間保持されなければならないことに注意してください。詳細については47頁の「8. 外部割り込み」章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22頁の「4.2. クロック元」項で記述されるように、リセット付加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。

5.6. パワーセーフ動作

SM1,0ビットが'11'を書かれると、**SLEEP**命令はMCUをパワーセーフ動作へ移行させます。この動作は(次の)1つの例外を除いてパワーダウン動作と同じです。

タイマ/カウンタ0が非同期でクロック駆動される、換言すると、**タイマ/カウンタ0非同期状態レジスタ(ASR)**の**タイマ/カウンタ0非同期動作許可(AS0)ビット**が設定(0)された場合、タイマ/カウンタ0は休止中に動きます。デバイスは許可される場合、それらは休止中も走行(動作)を維持します。**ステータスレジスタ(SREG)**の**全割り込み許可(I)**ビットが設定(1)され、**タイマ/カウンタ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**または**比較A割り込み許可(OCIE0A)ビット**が設定(1)されるなら、デバイスは対応するどちらの割り込みからでも起動できます。

非同期タイマ/カウンタが非同期にクロック駆動されないなら、AS0が0の場合、非同期タイマ/カウンタのレジスタ内容はパワーセーフ形態で起き上がった後に不定と見做されるべきなので、パワーセーフ動作の代わりにパワーダウン動作が推奨されます。

休止動作形態は基本的にclkASYを除く全てのクロックを停止し、非同期にクロック駆動されている場合のタイマ/カウンタ0を含み、非同期の単位部だけを許します。

5.7. 電力削減 (レジスタ)

電力削減レジスタ(38頁の**電力削減レジスタ(PRR)**参照)は消費電力を削減するために個別周辺機能へのクロックを停止する方法を提供します。周辺機能は現状で固定化され、I/Oレジスタは読み込みも書き込みもできません。クロックを停止している時に周辺機能によって使われていた資源は占有されたままなので、その周辺機能は殆どの場合、クロックを停止する前に禁止されるべきです。周辺機能部の起動は電力削減レジスタ(PRR)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。

5.8. 消費電力の最小化

これらはAVRが制御するシステムで消費電力の最小化を試みる時に考慮するには多数の可能性があります。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ために選択されるべきです。必要とされない全ての機能は禁止されるべきです。特に次の機能部は最低可能消費電力の達成を試みるとき、特別な考慮を必要とするでしょう。

5.8.1. A/D変換器 (ADC)

許可したなら、A/D変換器は全休止形態で許可されます。節電するため、休止形態の何れかへ移行する前にA/D変換器は禁止されるべきです。A/D変換器がOFFそして再びONに切り替えられると、次の(最初の)変換は延長された(初回)変換になります。A/D変換器操作の詳細については134頁の「18. A/D変換器」章を参照してください。

5.8.2. アナログ比較器

アイドル動作へ移行するとき、アナログ比較器は使われないなら、禁止されるべきです。A/D変換雑音削減動作へ移行するとき、アナログ比較器は禁止されるべきです。他の休止形態で、アナログ比較器は自動的に禁止されます。とは言え、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131頁の「17. アナログ比較器」章を参照してください。

5.8.3. 低電圧検出器 (BOD)

低電圧検出器(BOD)が応用で必要とされないなら、この単位部はOFFにされるべきです。低電圧検出器がBODLEVELヒューズによって許可されていると全休止形態で許可され、故に常時電力を消費します。これはより深い休止形態での総消費電流にとって重要な一因になります。低電圧検出器(BOD)設定法の詳細については40頁の「6.1.5. 低電圧検出(BOD)」項を参照してください。

5.8.4. 内部基準電圧

内部基準電圧は低電圧検出器(BOD)、アナログ比較器、A/D変換器によって必要とされる時に許可されます。これら単位部が上の項目で記述されたように禁止されると、内部基準電圧は禁止され、電力を消費しません。再び許可する場合、この出力が使われる前に使用者は基準電圧へ起動(安定時間)を与えなければなりません。基準電圧が休止形態でON保持される場合、この出力は直ちに使えます。起動時間の詳細については41頁の「6.2. 内部基準電圧」項を参照してください。

内部基準電圧の出力はより深い休止形態で必要とされません。この単位部は総消費電流に重要な減少のためにOFFされるべきです。内部基準電圧出力を禁止する方法の詳細については141頁の「AMISCR – アナログ雑多制御レジスタ」を参照してください。

5.8.5. 内部電流源

内部電流源はより深い休止形態で必要とされません。この単位部は総消費電流に重要な減少のためにOFFされるべきです。内部電流源を禁止する方法の詳細については145頁の「AMISCR – アナログ雑多制御レジスタ」を参照してください。

5.8.6. 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42頁の「6.3. ウォッチドッグ タイマ」項を参照してください。

5.8.7. 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53頁の「9.2.6. デジタル入力許可と休止形態」項を参照してください。入力緩衝部が許可され、入力信号が浮いている状態のままか、またはアナログ信号電圧がVCC/2付近の場合、入力緩衝部は過大な電力を消費するでしょう。

アナログ入力ピンに対するデジタル入力緩衝部は常に禁止されるべきです。入力ピンでのVCC/2付近のアナログ信号入力は通常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(DIDR0とDIDR1)の書き込みによって禁止することができます。詳細については133頁と144頁の「18.11.5. DIDR0 – デジタル入力禁止レジスタ0」項、145頁の「18.11.6. DIDR1 – デジタル入力禁止レジスタ1」項を参照してください。

5.8.8. 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され、従って常に電力を消費します。より深い休止形態に於いて、これは総消費電流にとって重要な一因になります。

5.9. 電力管理用レジスタ

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

この休止形態制御レジスタは電力管理用の制御ビットを含みます。

ビット	7	6	5	4	3	2	1	0	
\$33 (\$53)	–	–	–	–	–	SM1	SM0	SE	SMCR
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として読まれます。

● ビット2,1 – SM1,0 : 休止種別選択 (Sleep Mode Select Bit 1 and 0)

これらのビットは表5-2. で示される利用可能な4つの休止形態の1つを選択します。

表5-2. 休止動作種別選択

SM1	SM0	休止動作種別
0	0	アイドル動作
0	1	A/D変換雑音低減動作
1	0	パワーダウン動作
1	1	パワーセーブ動作

● 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。

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

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

● ビット6 – BODS : BOD休止 (BOD Sleep)

休止間にBODをOFFにするために、BODSビットは論理1を書かれなければなりません。34頁の表5-1.をご覧ください。BODSビットの書き込みはMCUCRのBOD休止許可(BODSE)ビットと時間制限手順によって制御されます。関連する休止形態でBODを禁止するには最初にBODSとBODSEの両方が1に設定されなければなりません。その後BODSビットを設定するためにBODSが1に設定され、そして4クロック周期内にBODSEが0に設定されなければなりません。

BODSビットはそれが設定された後の3クロック周期間活性(有効)です。**SLEEP**命令は実際の休止形態に対してBODをOFFにするために、BODSが活性(有効)の間に実行されなければなりません。BODSビットは3クロック周期後、自動的に解除(0)されます。

● ビット5 – BODSE : BOD休止許可 (BOD Sleep Enable)

BODSEはBOD休止(BODS)ビット記述で説明されるように、BODS制御ビットの設定を許可します。このBOD禁止は時間制限手順によって制御されます。

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

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

● ビット7,6 – Res : 予約 (Reserved)

これらのビットは予約されています。

● ビット5 – PRLIN : LIN/UART制御器電力削減 (Power Reduction LIN/UART controller)

このビットへの1書き込みはこの単位部へのクロック停止によってLINを切断します。LINの再起動時、LIN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4 – PRSPI : 直列周辺インターフェース(SPI)電力削減 (Power Reduction Serial Peripheral Interface)

デバッグWIRE内蔵デバッグ システムを使う場合、このビットは1を書かれるべきです。

このビットへの1書き込みはこの単位部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)を切断します。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部を切断します。タイマ/カウンタ1が許可されると、動作は切断前のように継続します。

● ビット2 – 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書き込みは同期動作形態(AS0=0)に於けるタイマ/カウンタ0部を切断します。タイマ/カウンタ0が許可されると、動作は切断前のように継続します。

● ビット1 – PRUSI : 多用途直列インターフェース(USI)電力削減 (Power Reduction USI)

このビットへの1書き込みはこの単位部へのクロック停止によって多用途直列インターフェース(USI)を切断します。USIの再起動時、USI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0 – PRADC : A/D変換器(ADC)電力削減 (Power Reduction ADC)

このビットへの1書き込みはこの単位部へのクロック停止によってA/D変換器(ADC)を切断します。A/D変換器は切断前に禁止されなければなりません。A/D変換器切断時、アナログ比較器はADC入力多重器を使うことができません。

6. システム制御とリセット

6.1. リセット

6.1.1. AVRのリセット

リセット中、全てのI/Oレジスタはそれらの初期値に設定され、プログラムはリセットベクタから実行を開始します。リセットベクタに配置される命令は、きっとリセット処理ルーチンへのJMP(絶対分岐)命令でしょう。プログラムが決して割り込み元を許可しないなら、割り込みベクタは使われず、これらの位置に通常のプログラムコードが配置できます。図6-1.の回路構成図はリセット論理回路を示します。166ページの「22.5. リセット特性」項の表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。

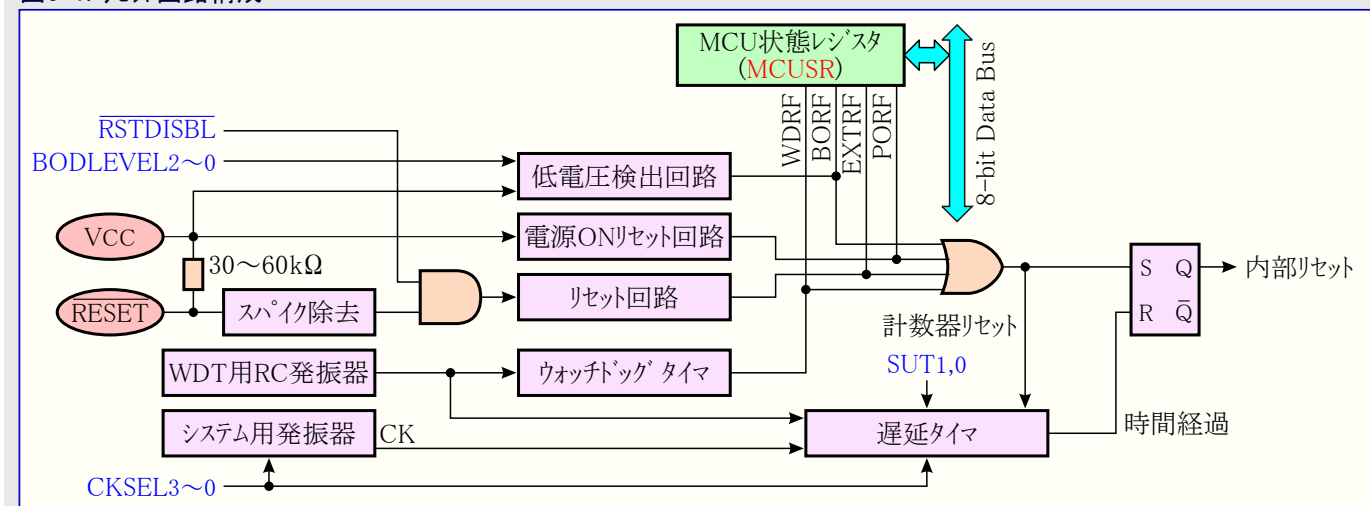
全てのリセット元が無効にされてしまった後、遅延計数器(タイマ)が始動され、内部リセットを引き伸ばします。これは通常動作開始前に安定電圧へ達することを電源に許します。遅延タイマの遅延時間はSUTヒューズとCKSELヒューズを通して使用者によって定義されます。この遅延時間についての各種選択は22ページの「4.2. クロック元」項で示されます。

6.1.2. リセット元

ATtiny87/167には次の4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。
- ・ **ウォッチドッグシステムリセット** ... ウォッチドッグシステムリセット動作が許可され、ウォッチドッグタイマが終了すると、MCUがリセットされます。
- ・ **低電圧リセット** 低電圧検出が許可され、供給電圧(VCC)が低電圧検出電圧(V_{BOT})以下でMCUがリセットされます。

図6-1. リセット回路構成



6.1.3. 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166頁の表22-4.で定義されます。POR信号はVCCが検出電圧以下の時に必ず活性(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うことができます。

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のを保証します。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6-2. 内蔵電源ONリセット (RESETはVCCに接続)

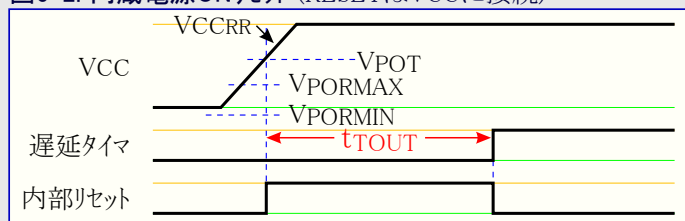
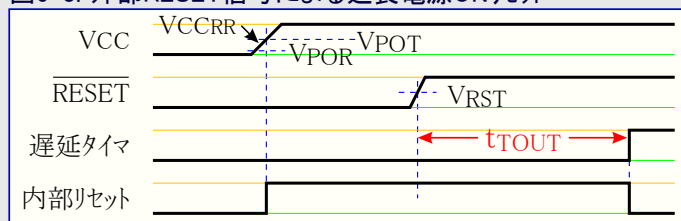


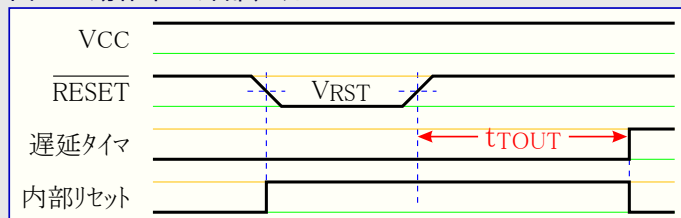
図6-3. 外部RESET信号による延長電源ONリセット



6.1.4. 外部リセット

外部リセットはRESETピンのLowレベルによって生成されます。例えばクロックが動いていなくても、最小パルス幅(166頁の表22-3.参照)以上のリセットパルスは、リセットを生成します。短すぎるパルスはリセットの生成が保証されません。印加された信号の上昇が、リセット閾値電圧(V_{RST})に達すると(遅延タイマが起動され)、遅延タイマは遅延時間(t_{TOUT})経過後にMCUを始動します。外部リセットはRSTDISBLヒューズによって禁止できます。153頁の表21-4.をご覧ください。

図6-4. 動作中の外部リセット

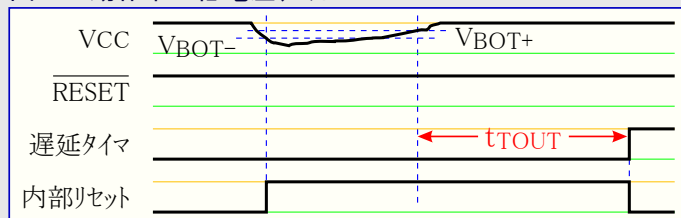


6.1.5. 低電圧(ブラウンアウト)検出リセット

ATtiny87/167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択できます(166頁のBODLEVELヒューズ(VBAT)設定)をご覧ください。起動電圧はスパイク対策BODを保证するためにヒステリシスを持ちます。検出電圧のヒステリシスは $V_{BOT+} = V_{BOT} + V_{HYST}/2$ 、 $V_{BOT-} = V_{BOT} - V_{HYST}/2$ と解釈すべきです。

BODが許可され、VCCが起動電圧以下の値に下降すると(図6-5.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6-5.の V_{BOT+})、(遅延タイマが起動されて)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166頁の表22-6.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。

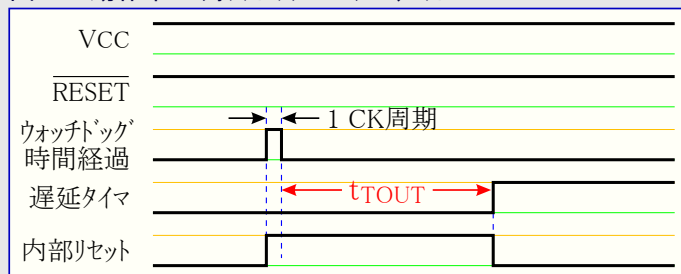
図6-5. 動作中の低電圧リセット



6.1.6. ウォッチドッグ システム リセット

ウォッチドッグ時間経過時、(内部的に)1CK周期幅の短いリセットパルスを生成します。この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42頁を参照してください。

図6-6. 動作中のウォッチドッグ システム リセット



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

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

ビット	7	6	5	4	3	2	1	0	
\$34 (\$54)	–	–	–	–	WDRF	BORF	EXTRF	PORF	MCUSR
Read/Write	R	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7～4 – Res : 予約 (Reserved)

これらのビットは将来の使用に予約されています。

● ビット3 – WDRF : ウォッチドッグ システム リセット フラグ (Watchdog System Reset Flag)

このビットはウォッチドッグ システム リセットが起こると設定(1)されます。このビットは電源ONリセットまたは、このフラグへの論理0書き込みによってリセット(0)されます。

● ビット2 – BORF : 低電圧リセット フラグ (Brown-Out Reset Flag)

このビットは低電圧リセットが起こると設定(1)されます。このビットは電源ONリセットまたは、このフラグへの論理0書き込みによってリセット(0)されます。

● ビット1 – EXTRF : 外部リセット フラグ (External Reset Flag)

このビットは外部リセットが起こると設定(1)されます。このビットは電源ONリセットまたは、このフラグへの論理0書き込みによってリセット(0)されます。

● ビット0 – PORF : 電源ONリセット フラグ (Power-on Reset Flag)

このビットは電源ONリセットが起こると設定(1)されます。このビットはこのフラグへの論理0書き込みによってのみリセット(0)されます。

リセット条件の確認にリセット フラグを使うため、使用者はプログラム内で可能な限り早くMCUSRを読み、そして解除(0)すべきです。別のリセットが起こる前にこのレジスタが解除(0)されると、そのリセット元はリセット フラグを調べることによって得られます。

6.2. 内部基準電圧

ATtiny87/167は内部基準電圧が特徴です。この基準電圧は低電圧検出(BOD)に使われ、A/D変換やアナログ比較器の入力としても使えます。

6.2.1. 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7-4.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)のアナログ比較器内部基準選択(ACIRS)=1)
3. A/D変換部動作許可時 (A/D変換制御/状態レジスタA(ADCSRA)のA/D動作許可(ADEN)=1)

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

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: 9頁の「コード例について」をご覧ください。

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

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

アセンブリ言語プログラム例

```
WDT_PRS:    CLI                ;全割り込み禁止
            WDR                ;ウォッチドッグ タイマ リセット
            LDS     R16, WDTCSR ;現WDTCSR値を取得(他ビット保護用)
            ORI     R16, (1<<WDCE) | (1<<WDE) ;WDCEとWDE論理1値を設定
            STS     WDTCSR, R16 ;WDCEとWDEに論理1書き込み
            LDI     R16, (1<<WDE) | (1<<WDP2) | (1<<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: 9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。

6.3.2. クロック監視

ウォッチドッグ タイマはシステム クロックの消失を検知するのに使うことができます。この構成設定は動的クロック切り替え回路によって駆動されます。より多くの情報については28頁の「4.3.8. クロック監視」項を参照してください。

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

ビット (\$60)	7	6	5	4	3	2	1	0	
	WDIF	WDIE	WDP3	WDCE	WDE	WDP2	WDP1	WDP0	WDTCSR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	不定	0	0	0	

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

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

● ビット6 – WDIE : ウォッチドッグ割り込み許可 (Watchdog Interrupt Enable)

このビットが1を書かれ、ステータスレジスタ(SREG)の全割り込み許可(I)ビットが設定(1)されると、ウォッチドッグ割り込みが許可されます。この設定(=1)との組み合わせでウォッチドッグリセット許可(WDE)ビットが解除(0)されると、割り込み動作種別になり、ウォッチドッグ タイマで計時完了が起こると、対応する割り込みが実行されます。

WDEが設定(1)されると、ウォッチドッグ タイマは割り込み及びシステムリセット動作種別になります。ウォッチドッグ タイマでの最初の計時完了がウォッチドッグ割り込み要求(WDIF)フラグを設定(1)します。対応する割り込みベクタの実行はハードウェアによってWDIEとWDIFを自動的に解除(0)します。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。

ウォッチドッグ タイマがクロック監視(32頁の「ビット3～0 – CLKC3～0: クロック制御ビット3～0」を参照)として使われる場合、システムリセット動作形態と割り込み動作形態は自動的に禁止されます。

表6-1. ウォッチドッグ タイマ設定

クロック監視	WDTON	WDE	WDIE	動作種別	計時完了での動作
x	1	0	0	停止	なし
ON	y (注2)	y (注2)	y (注2)	システムリセット	リセット
OFF	1	0	1	割り込み	割り込み
	1	1	0	システムリセット	リセット
	1	1	1	割り込み及びシステムリセット	割り込み、その後システムリセット動作種別
	0	x	x	システムリセット	リセット

(訳注)

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

注1: WDTONヒューズは0でプログラム、1で非プログラムに設定です。

注2: WDTON, WDE, WDIEが各々'1', '0', '0'以外の場合です。

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

このビットはウォッチドッグリセット許可(WDE)と前置分周器ビットの変更用の時間制限手順で使われます。WDEビットの解除(0)や前置分周器ビット変更のため、WDCEは設定(1)されなければなりません。

一旦1を書かれると、4クロック周期後にハードウェアがWDCEを解除(0)します。

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

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

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

このWDP3～0ビットはウォッチドッグ タイマが走行する時のウォッチドッグ タイマの前置分周を決めます。各種前置分周値と対応する計時完了周期は表6-2.で示されます。

表6-2. ウォッチドッグ前置分周器選択

WDP3	0								1							
WDP2	0				1				0				1			
WDP1	0		1		0		1		0		1		0		1	
WDP0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
WDT発振周期数	2k	4k	8k	16k	32k	64k	128k	256k	512k	1024k	(予約) (注)					
代表的計時完了周期 (VCC=5V)	16ms	32ms	64ms	0.125s	0.25s	0.5s	1.0s	2.0s	4.0s	8.0s						

注: 選択した場合、1010未満の有効な設定の(どれか)1つが使われます(訳注: 共通性から本注追加)。

7. 割り込み

本項はATtiny87/167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13頁の「リセットと割り込みの扱い」を参照してください。

7.1. ATtiny87/167の割り込みベクタ

表7-1. ATtiny87/167の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		発生元	割り込み定義
	ATtiny87	ATtiny167		
1	\$0000	\$0000	RESET	外部ピン, 電源ON, WDT, BOD等の各種リセット
2	\$0001	\$0002	INT0	外部割り込み要求0
3	\$0002	\$0004	INT1	外部割り込み要求1
4	\$0003	\$0006	PCINT0	ピン変化0群割り込み要求 (PCI0)
5	\$0004	\$0008	PCINT1	ピン変化1群割り込み要求 (PCI1)
6	\$0005	\$000A	WDT	ウォッチドッグ計時終了
7	\$0006	\$000C	TIMER1_CAPT	タイマ/カウンタ1捕獲発生
8	\$0007	\$000E	TIMER1_COMPA	タイマ/カウンタ1比較A一致
9	\$0008	\$0010	TIMER1_COMPB	タイマ/カウンタ1比較B一致
10	\$0009	\$0012	TIMER1_OVF	タイマ/カウンタ1溢れ
11	\$000A	\$0014	TIMER0_COMPA	タイマ/カウンタ0比較A一致
12	\$000B	\$0016	TIMER0_OVF	タイマ/カウンタ0溢れ
13	\$000C	\$0018	LIN_TC	LIN 転送完了
14	\$000D	\$001A	LIN_ERR	LIN 異常発生
15	\$000E	\$001C	SPI_STC	SPI 転送完了
16	\$000F	\$001E	ADC	A/D変換完了
17	\$0010	\$0020	EE_RDY	EEPROM 操作可
18	\$0011	\$0022	ANA_COMP	アナログ比較器出力遷移
19	\$0012	\$0024	USI_START	USI 開始条件検出
20	\$0013	\$0026	USI_OVF	USI 計数器溢れ

7.2. ATtiny87でのプログラム構成設定

ATtiny87での最も代表的且つ一般的なリセットと割り込みのベクタ アドレス用プログラム構成設定は次のとおり(RJMP命令を用いた2バイト毎)です。

アドレス	ラベル	命令	注釈
\$0000		RJMP RESET	;各種リセット
\$0001		RJMP INT0addr	;外部割り込み要求0
\$0002		RJMP INT1addr	;外部割り込み要求1
\$0003		RJMP PCINT0addr	;ピン変化0群割り込み要求
\$0004		RJMP PCINT1addr	;ピン変化1群割り込み要求
\$0005		RJMP WDTaddr	;ウォッチドッグ計時完了
\$0006		RJMP ICP1addr	;タイマ/カウンタ1捕獲発生
\$0007		RJMP OC1Aaddr	;タイマ/カウンタ1比較A一致
\$0008		RJMP OC1Baddr	;タイマ/カウンタ1比較B一致
\$0009		RJMP OVFladdr	;タイマ/カウンタ1溢れ
\$000A		RJMP OCOAaddr	;タイマ/カウンタ0比較A一致
\$000B		RJMP OVFOaddr	;タイマ/カウンタ0溢れ
\$000C		RJMP LINTCaddr	;LIN 転送完了
\$000D		RJMP LINERRaddr	;LIN 異常発生
\$000E		RJMP SPIaddr	;SPI転送完了
\$000F		RJMP ADCaddr	;ADC変換完了
\$0010		RJMP ERDYaddr	;EEPROM操作可
\$0011		RJMP ACIaddr	;アナログ比較器出力遷移
\$0012		RJMP USISTARTaddr	;USI 開始条件検出
\$0013		RJMP USIOVFaddr	;USI 計数器溢れ
;			
\$0014	RESET:	LDI R16, HIGH (RAMEND)	;RAM最終アドレス上位を取得
\$0015		OUT SPH, R16	;スタック ポインタ上位を初期化
\$0016		LDI R16, LOW (RAMEND)	;RAM最終アドレス下位を取得
\$0017		OUT SPL, R16	;スタック ポインタ下位を初期化
		}	;以下、I/O初期化など

7.3. ATtiny167でのプログラム構成設定

ATtiny167での最も代表的且つ一般的なリセットと割り込みのベクタ アドレス用プログラム構成設定は次のとおり(JMP命令を用いた4バイト毎)です。

アドレス	ラベル	命令	注釈
\$0000		JMP RESET	;各種リセット
\$0002		JMP INT0addr	;外部割り込み要求0
\$0004		JMP INT1addr	;外部割り込み要求1
\$0006		JMP PCINT0addr	;ピン変化0群割り込み要求
\$0008		JMP PCINT1addr	;ピン変化1群割り込み要求
\$000A		JMP WDTaddr	;ウォッチドッグ計時完了
\$000C		JMP ICP1addr	;タイマ/カウンタ1捕獲発生
\$000E		JMP OC1Aaddr	;タイマ/カウンタ1比較A一致
\$0010		JMP OC1Baddr	;タイマ/カウンタ1比較B一致
\$0012		JMP OVFladdr	;タイマ/カウンタ1溢れ
\$0014		JMP OCOAaddr	;タイマ/カウンタ0比較A一致
\$0016		JMP OVFOaddr	;タイマ/カウンタ0溢れ
\$0018		JMP LINTCaddr	;LIN 転送完了
\$001A		JMP LINERRaddr	;LIN 異常発生
\$001C		JMP SPIaddr	;SPI転送完了
\$001E		JMP ADCaddr	;ADC変換完了
\$0020		JMP ERDYaddr	;EEPROM操作可
\$0022		JMP ACIaddr	;アナログ比較器出力遷移
\$0024		JMP USISTARTaddr	;USI 開始条件検出
\$0026		JMP USIOVFaddr	;USI 計数器溢れ
;			
\$0028	RESET:	LDI R16, HIGH (RAMEND)	;RAM最終アドレス上位を取得
\$0029		OUT SPH, R16	;スタック ポインタ上位を初期化
\$002A		LDI R16, LOW (RAMEND)	;RAM最終アドレス下位を取得
\$002B		OUT SPL, R16	;スタック ポインタ下位を初期化
		}	;以下、I/O初期化など

8. 外部割り込み

8.1. 概要

外部割り込みはINT0,INT1ピンまたはPCINT0～15ピンの何れかによって起動されます。許可したなら、例えINT0,1またはPCINT0～15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PCII**は許可したPCINT8～15の何れかが切り替わると起動します。**ピン変化割り込みPCIO**は許可したPCINT0～7の何れかが切り替わると起動します。ピン変化割り込み許可レジスタn(**PCMSK0,PCMSK1**)は、どのピンがピン変化割り込み要因となるかを制御します。PCINT0～15でのピン変化割り込みは非同期に検知されます。これはそれらの割り込みがアイドル動作以外の休止形態からもデバイスを起動するのに使えることを意味します。

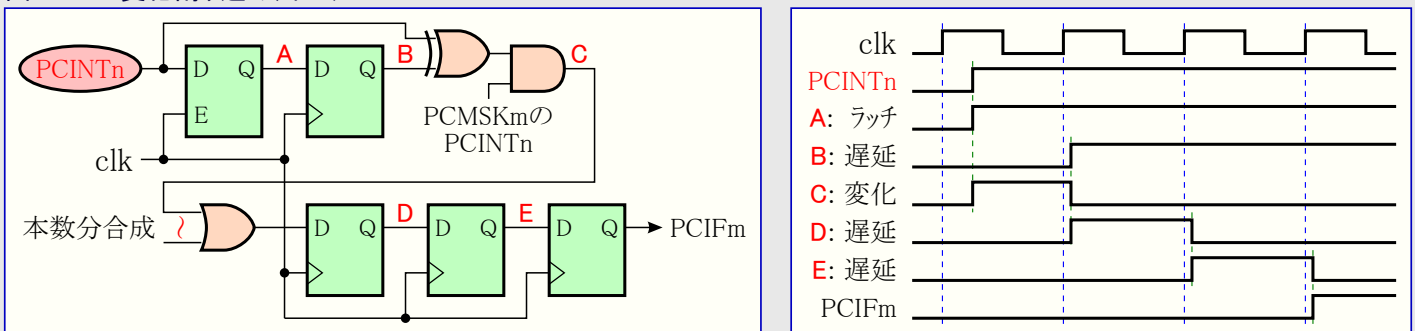
INT0とINT1割り込みは上昇端または下降端(含む両端)、またはLowレベルによって起動できます。これは**外部割り込み制御レジスタA(EICRA)**の詳述で示される設定です。INT0またはINT1割り込みがレベル起動として設定、且つ許可されると、そのピンがLowに保持される限り、割り込みは(継続的に)起動します。INT0またはINT1の上昇端や下降端割り込みの認知は、**21頁の「クロックシステムとその配給」**で記述されるI/Oクロックの存在を必要とします。INT0とINT1のLowレベル割り込みとエッジ割り込みは非同期に検知されます。これはそれらの割り込みが**アイドル動作**以外の**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**と**パワーセーブ動作**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**21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。

8.2. ピン変化割り込みタイミング

ピン変化割り込みの例は**図8-1**で示されます。

図8-1. ピン変化割り込みタイミング



8.3. 外部割り込み用レジスタ

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

外部割り込み制御レジスタAは割り込み条件制御ビットを含みます。

ビット (\$69)	7	6	5	4	3	2	1	0	
	–	–	–	–	ISC11	ISC10	ISC01	ISC00	EICRA
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として読まれます。

● ビット3,2 – ISC11,0 : 外部割り込み1条件制御 (Interrupt Sense Control 1 bit1 and 0)

外部割り込み1は**ステータスレジスタ(SREG)の全割り込み許可(I)ビット**と**外部割り込み許可レジスタ(EIMSK)の外部割り込み1許可(INT1)ビット**が設定(1)される場合のINT1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INT1ピンのエッジとレベルは**表8-1**で定義されます。INT1ピンの値はエッジ検出に先立って採取されています。エッジまたは論理変化割り込みが選択される場合、1クロック周期よりも長く留まるパルスは割り込みを生成します。より短いパルスは割り込みの発生が保証されません。Lowレベル割り込みが選択される場合、そのLowレベルは割り込みを生成する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。

表8-1. 外部割り込み1(INT1)割り込み条件

ISC11	ISC10	割り込み発生条件
0	0	INT1ピンがLowレベルで発生。
0	1	INT1ピンの論理変化(両端)。
1	0	INT1ピンの下降端で発生。
1	1	INT1ピンの上昇端で発生。

●ビット1,0 – ISC01,0 : 外部割り込み0条件制御 (Interrupt Sense Control 0 bit1 and 0)

外部割り込み0はステータスレジスタ(SREG)の全割り込み許可(I)ビットと外部割り込み許可レジスタ(EIMSK)の外部割り込み0許可(INT0)ビットが設定(1)される場合のINT0外部ピンによって活性(有効)にされます。割り込みを活性にする外部INT0ピンのエッジとレベルは表8-2で定義されます。INT0ピンの値はエッジ検出に先立って採取されています。エッジまたは論理変化割り込みが選択される場合、1クロック周期よりも長く留まるパルスは割り込みを生成します。より短いパルスは割り込みの発生が保証されません。Lowレベル割り込みが選択される場合、そのLowレベルは割り込みを生成する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。

表8-2. 外部割り込み0(INT0)割り込み条件

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

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

ビット	7	6	5	4	3	2	1	0	
\$1D (\$3D)	–	–	–	–	–	–	INT1	INT0	EIMSK
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)

これらのビットは予約されており、常に0として読めます。

●ビット1 – INT1 : 外部割り込み1許可 (External Interrupt Request 1 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、INT1ビットが設定(1)されると、INT1外部ピン割り込みが許可されます。外部割り込み制御レジスタA(EICRA)の割り込み条件制御1のビット1と0(ISC11,ISC10)は、この外部割り込みがINT1ピンの上昇端、下降端、両端、またはLowレベルのどれで活性(有効)にされるかを定義します。例えばINT1ピンが出力として設定されても、このピンの動きは割り込み要求を引き起こします。外部割り込み要求1に対応する割り込みはINT1割り込みベクタから実行されます。

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

ステータスレジスタ(SREG)の全割り込み許可(I)ビットが設定(1)され、INT0ビットが設定(1)されると、INT0外部ピン割り込みが許可されます。外部割り込み制御レジスタA(EICRA)の割り込み条件制御0のビット1と0(ISC01,ISC00)は、この外部割り込みがINT0ピンの上昇端、下降端、両端、またはLowレベルのどれで活性(有効)にされるかを定義します。例えばINT0ピンが出力として設定されても、このピンの動きは割り込み要求を引き起こします。外部割り込み要求0に対応する割り込みはINT0割り込みベクタから実行されます。

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

ビット	7	6	5	4	3	2	1	0	
\$1C (\$3C)	–	–	–	–	–	–	INTF1	INTF0	EIFR
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)

これらのビットは予約されており、常に0として読めます。

●ビット1 – INTF1 : 外部割り込み1要求フラグ (External Interrupt Flag 1)

INT1ピン上のエッジまたは論理変化が割り込み要求を起動すると、INTF1が設定(1)になります。ステータスレジスタ(SREG)の全割り込み許可(I)ビットと外部割り込み許可レジスタ(EIMSK)の外部割り込み1許可(INT1)ビットが設定(1)なら、MCUは対応する割り込みベクタへ飛びます。このフラグは割り込みルーチンが実行されると解除(0)されます。代わりにこのフラグは論理1を書くことによって解除(0)できます。INT1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。

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

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

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

ビット (\$68)	7	6	5	4	3	2	1	0	
	–	–	–	–	–	–	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)

これらのビットは予約されており、常に0として読めます。

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

ステータスレジスタ(SREG)の全割り込み許可(I)ビットが設定(1)され、PCIE1ビットが設定(1)されると、ピン変化割り込み1が許可されます。許可したPCINT8～15ピンの何れかの変化が割り込みを起こします。このピン変化割り込み要求に対応する割り込みはPCIE1割り込みベクタから実行されます。PCINT8～15ピンはピン変化割り込み許可レジスタ1(PCMSK1)によって個別に許可されます。

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

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

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

ビット \$1B (\$3B)	7	6	5	4	3	2	1	0	
	–	–	–	–	–	–	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)

これらのビットは予約されており、常に0として読めます。

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

PCINT8～15ピンの何れかの論理変化が割り込み要求を起動すると、PCIF1が設定(1)になります。ステータスレジスタ(SREG)の全割り込み許可(I)ビットとピン変化割り込み制御レジスタ(PCICR)のピン変化1群割り込み許可(PCIE1)ビットが設定(1)なら、MCUは対応する割り込みベクタへ飛びます。このフラグは割り込み処理ルーチンが開始されると解除(0)されます。代わりにこのフラグは論理1を書くことによっても解除(0)できます。

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

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

8.3.6. PCMSK1 – ピン変化割り込み許可レジスタ1 (Pin Change Enable Mask 8～15)

ビット (\$6C)	7	6	5	4	3	2	1	0	
	PCINT15	PCINT14	PCINT13	PCINT12	PCINT11	PCINT10	PCINT9	PCINT8	PCMSK1
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 – PCINT15～PCINT8 : ピン変化割り込み15～8許可 (Pin Change Enable Mask 15～8)

各PCINT8～15ビットは対応するI/Oピンでピン変化割り込みが許可されるかどうかを選びます。PCINT8～15とピン変化割り込み制御レジスタ(PCICR)のPCIE1が設定(1)なら、対応するI/Oピンのピン変化割り込みが許可されます。PCINT8～15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。

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

ビット (\$6B)	7	6	5	4	3	2	1	0	
	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0	PCMSK0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

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

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

9. 入出力ポート

9.1. 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角をも無意識に変更することなく、1つのポートピンの方角を変更できることを意味します。(出力として設定されていれば)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9-1.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164頁の「電気的特性」を参照してください。

本項内の全てのレジスタとビットの参照は一般形で記されます。小文字の'x'はポート番号文字、小文字の'n'はビット番号を表します。けれどもプログラム内でレジスタやビット定義に使うとき、正確な形式(例えば、ここで一般に記された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62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)ビットまたはポート制御レジスタ(PORTCR)のポート単位プルアップ禁止(PUDx)ビットが設定(1)されると、全ポートまたは指定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54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。

9.2. 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9-2.はここで属にPxnと呼ばれるI/Oポートピンの1つの機能説明を示します。

図9-1. 入出力ピン等価回路

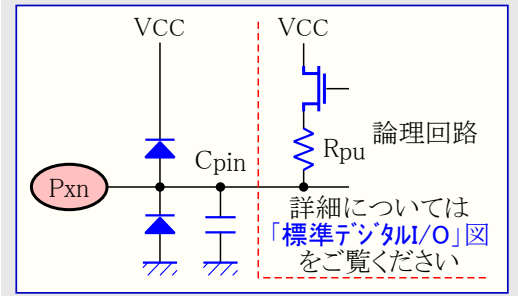
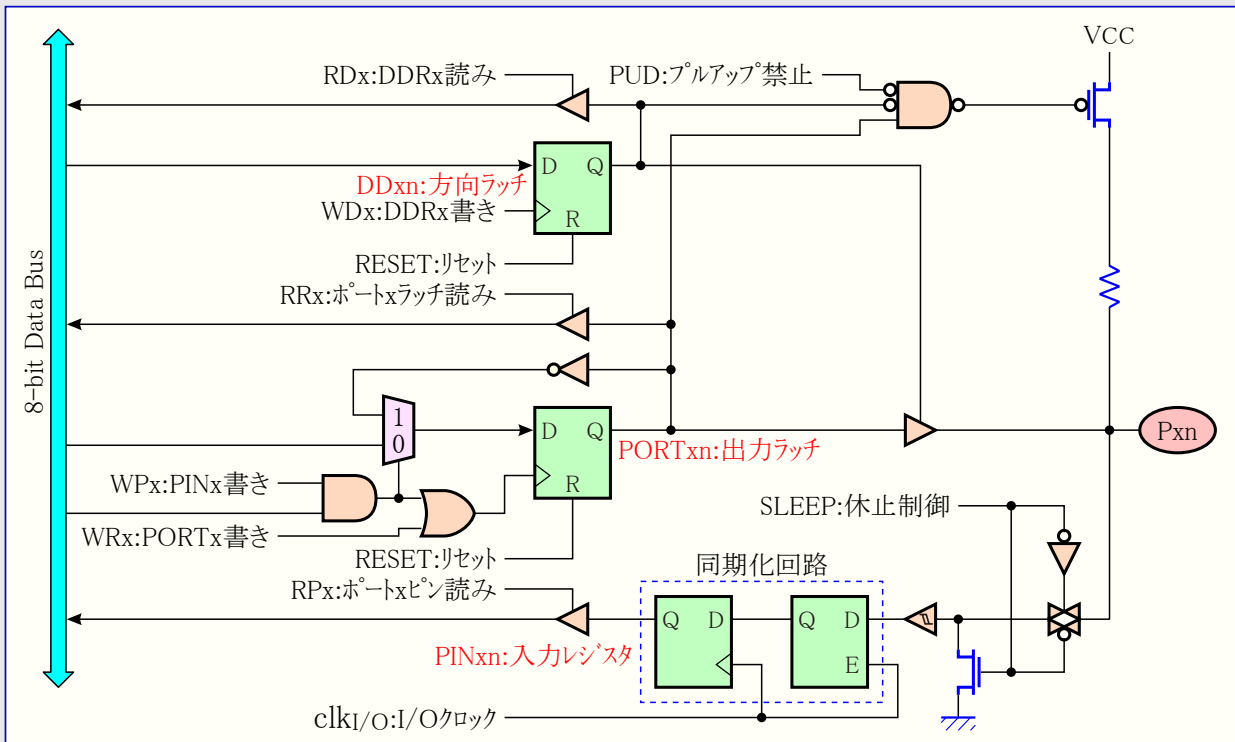


図9-2. 標準デジタル入出力回路構成



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。

9.2.1. 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。62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択します。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。

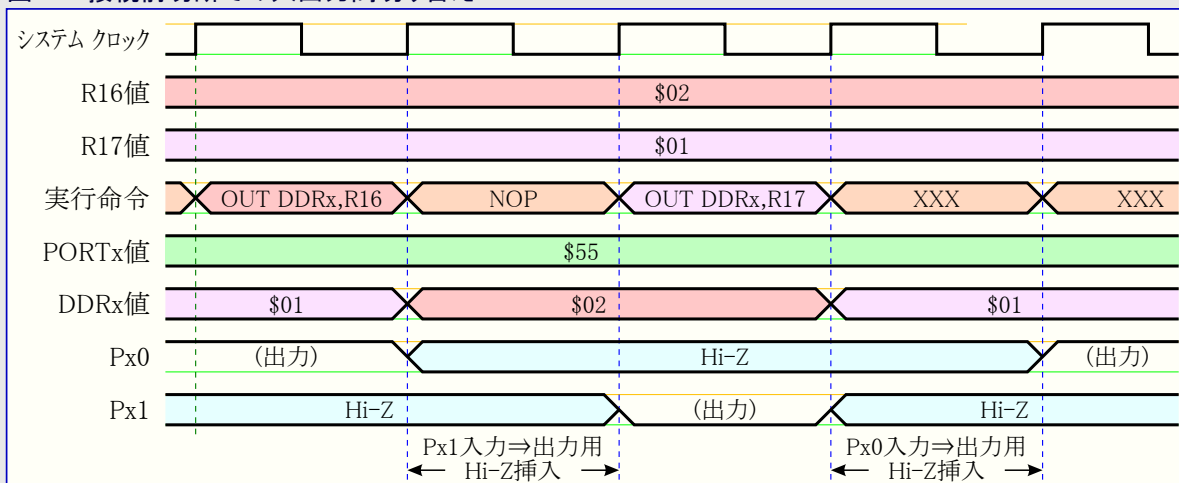
9.2.2. ピンの出力交互切り替え

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。

9.2.3. 接続前切断(Break-Before-Make)切り替え

接続前切断動作ではDDRxnを入力から出力へ切り換える時に1システムクロック持続する隣接Hi-Z区間が図9-3.で示されるように導入されます。例えば、システムクロックが4MHzでDDRxnが出力にするように書かれた場合、PORTxnの値がポートピンで見える前に250nsの隣接Hi-Z区間が導入されます。異常を避けるため、DDRxn最大切り替え周波数は2システムクロックが推奨されます。この接続前切断はポート単位動作で、ポート単位の接続前切断許可(BBMx)ビットによって活性にされます。BBMxビットについての更なる情報に関しては55頁の「**PORTCR – ポート制御レジスタ**」をご覧ください。DDRxnビットを出力から入力に切り替える時に隣接Hi-Z区間は導入されません。

図9-3. 接続前切断での入出力間切り替え



9.2.4. 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、全ポートまたは特定ポートの全プルアップを禁止するのに、**MCU制御レジスタ(MCUCR)**の**プルアップ禁止(PUD)**ビットまたは**ポート制御レジスタ(PORTCR)**の**ポート単位プルアップ禁止(PUDx)**ビットを設定(**1**)することができます。

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9-1. はピン値に対する制御信号の一覧を示します。

表9-1. ポートピンの設定

DDxn	PORTxn	PUD (MCUCR) (注)	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

注: またはポート制御レジスタ(PORTCR)のポート単位プルアップ禁止(PUDx)ビット

9.2.5. 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9-2.で示されるようにPINxnレジスタビットと先行するラッチは同期化回路を構成します。これは物理ピンが内部クロックのエッジ付近で値を変える場合の未定義状態(メタステーブル)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9-4.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9-4.で)システムクロックの最初の上昇端直後から始まるクロック周期を考察してください。このラッチはクロックがLowの時に閉じ、クロックがHighの時に同期ラッチ信号の斜線部分で示されるように通過(トランスペアレント)となります。この信号値はシステムクロックがLowになる時に保持(ラッチ)されます。それが続くクロックの上昇端でPINxnレジスタに取り込まれます。2つの矢印 $t_{pd,min}$ と $t_{pd,max}$ によって示されるように、ピン上の単一信号遷移は出現時点に依存して0.5~1.5システムクロック周期遅らされます。

ソフトウェアが指定したピン値を読み戻す時は、図9-5.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9-4. 外部供給ピン値読み込み時の同期化

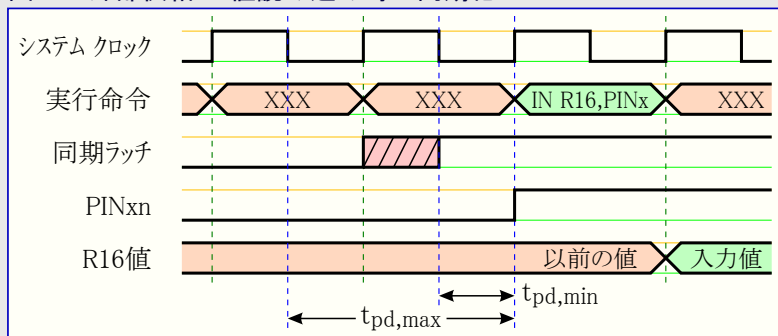
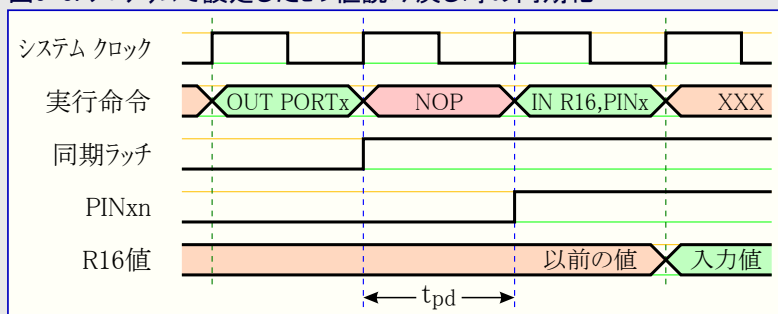


図9-5. プログラムで設定したピン値読み戻し時の同期化



次のコード例はポートピンの0と1をHigh出力、2と3をLow出力、6と7をプルアップ指定として4~7を入力に設定する方法を示します。結果のピン値が再び読み戻されますが、前記で検討されたように、いくつかのピンへ直前に指定された値を読み戻すことができるようにNOP命令が挿入されます。

アセンブリ言語プログラム例

```

~
LDI    R16, (1<<PB7) | (1<<PB6) | (1<<PB1) | (1<<PB0)    ;プルアップとHigh値を取得
LDI    R17, (1<<DDB3) | (1<<DDB2) | (1<<DDB1) | (1<<DDB0) ;出力ビット値を取得
OUT    PORTB, R16    ;プルアップとHigh値を設定
OUT    DDRB, R17     ;入出力方向を設定
NOP    ;同期化遅延対処
IN     R16, PINB     ;ピン値読み戻し
~
;
```

C言語プログラム例

```

unsigned char i;

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

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

9.2.6. デジタル入力許可と休止形態

図9-2. で示されるようにデジタル入力信号はシュミットトリガの入力をGNDにクランプできます。この図でSLEEPと印された信号は入力信号のいくつかが開放のまま、またはVCC/2付近のアナログ信号電圧を持つ場合の高消費電力を避けるため、パワーダウン動作またはパワーセーブ動作で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54ページの「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。

9.2.7. 未接続ピン

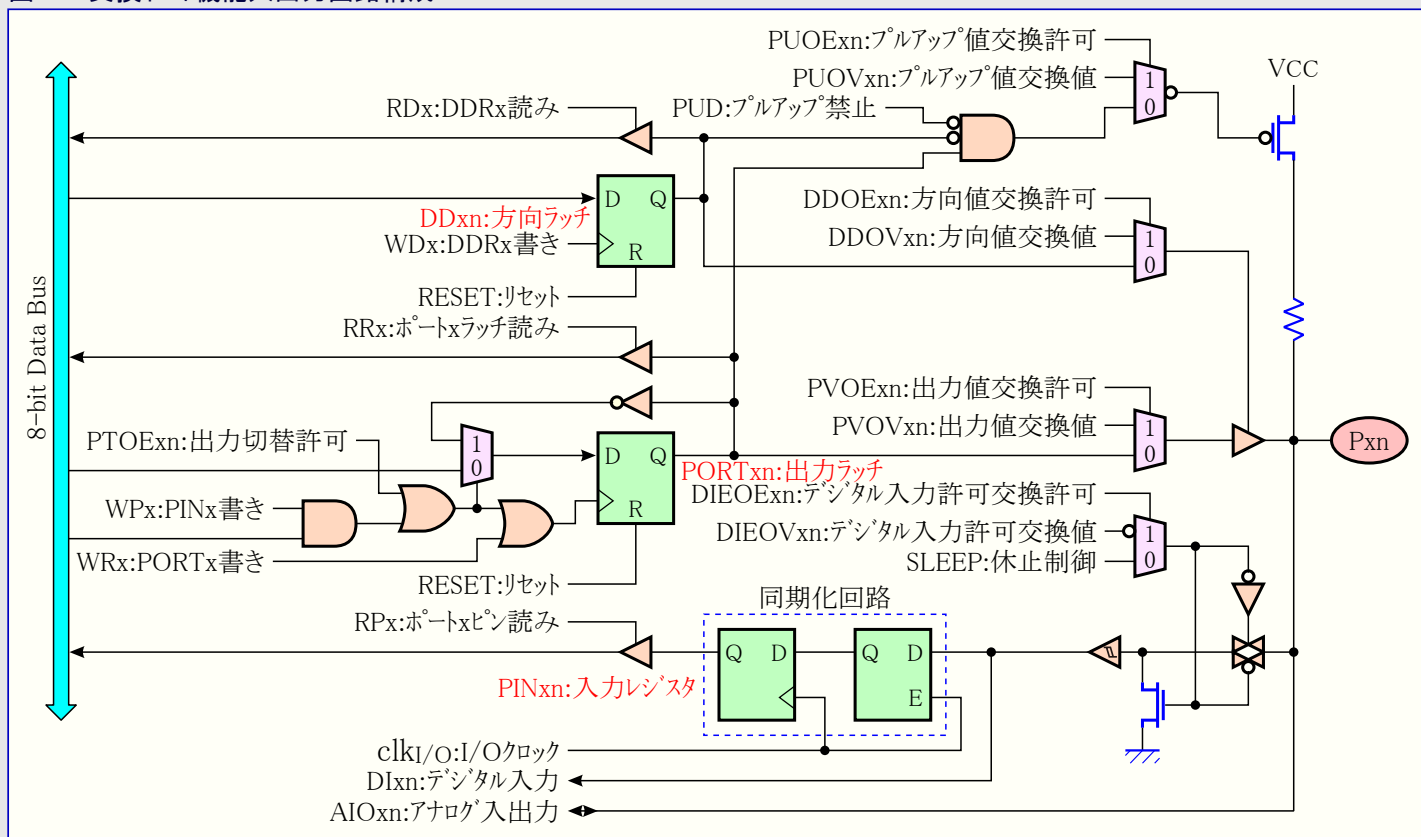
いくつかのピンが未使用にされる場合、それらのピンが定義されたレベルを持つのを保証することが推奨されます。例え上記のような深い休止形態で多くのデジタル入力が禁止されるとしても、デジタル入力が許可される他の全ての動作(リセット、活動動作、アイドル動作)で消費電流削減のため、浮き状態入力は避けられるべきです。

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。

9.3. 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9-6は単純化された図9-2.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。

図9-6. 交換ポート機能入出力回路構成



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。他の信号は各ピン固有です。

表9-2は重複(交換)信号の機能一覧を示します。図9-6.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9-2. 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トリが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。

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

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

ビット	7	6	5	4	3	2	1	0	
\$35 (\$55)	–	BODS	BODSE	PUD	–	–	–	–	MCUCR
Read/Write	R	R/W	R/W	R/W	R	R	R	R	
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51頁の「[ピンの設定](#)」をご覧ください。

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

ビット	7	6	5	4	3	2	1	0	
\$12 (\$32)	–	–	BBMB	BBMA	–	–	PUDB	PUDA	PORTCR
Read/Write	R	R	R/W	R/W	R	R	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

● ビット5,4 – BBMx : 接続前切断動作許可 (Break-Before-Make Mode Enable)

これらのビットが1を書かれると、ポート単位の接続前切断動作が活性(有効)にされます。そして出力にするためのDDRxn書き込み時に仲介Hi-Z周期が挿入されます。更なる情報については51頁の「[接続前切断切り替え](#)」をご覧ください。

● ビット1,0 – PUDx : ポート単位プルアップ禁止 (Port-Wise Pull-up Disable)

これらのビットが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、定義された入出力ポートでポート単位のプルアップが禁止されます。これらのポート単位プルアップ禁止ビットはMCU制御レジスタ(MCUCR)の全プルアップ禁止(PUD)ビットと論理和(OR)されます。この特徴についてより多くの詳細に関しては51頁の「[ピンの設定](#)」をご覧ください。

9.3.3. ポートAの交換機能

ポートAピンの交換機能は表9-3.で示されます。

表9-3. ポートA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
PA7	ADC7 (A/D変換チャンネル7入力) AIN1 (アナログ比較器反転入力) XREF (内部基準電圧出力) AREF (外部基準電圧入力) PCINT7 (ピン変化割り込み7入力)	PA3	INT1 (外部割り込み1 入力) ISRC (電流源ピン) ADC3 (A/D変換チャンネル3入力) PCINT3 (ピン変化割り込み3入力)
PA6	SS (SPI 従装置選択入力) ADC6 (A/D変換チャンネル6入力) AIN0 (アナログ比較器非反転入力) PCINT6 (ピン変化割り込み6入力)	PA2	OC0A (タイマ/カウンタ0 比較A/PWM出力) MISO (SPIデータ 主側入力/従側出力) DO (3線動作USIデータ出力) ADC2 (A/D変換チャンネル2入力) PCINT2 (ピン変化割り込み2入力)
PA5	T1 (タイマ/カウンタ1 外部クロック入力) SCK (SPIクロック 主側出力/従側入力) USCK (3線動作USIクロック入出力) SCL (2線動作USI(TWI)クロック入出力) ADC5 (A/D変換チャンネル5入力) PCINT5 (ピン変化割り込み5入力)	PA1	TXD (UART送信ピン) TXLIN (LIN送信ピン) ADC1 (A/D変換チャンネル1入力) PCINT1 (ピン変化割り込み1入力)
PA4	ICP1 (タイマ/カウンタ1 捕獲起動入力) MOSI (SPIデータ 主側出力/従側入力) DI (3線動作USIデータ入力) SDA (2線動作USI(TWI)データ入出力) ADC4 (A/D変換チャンネル4入力) PCINT4 (ピン変化割り込み4入力)	PA0	RXD (UART受信ピン) RXLIN (LIN受信ピン) ADC0 (A/D変換チャンネル0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。

• **ポートA ピン7 : PA7 – ADC7/AIN1/XREF/AREF/PCINT7**

- **ADC7** : A/D変換チャネル7入力。
- **AIN1** : アナログ比較器非反転入力。このピンはアナログ比較器の非反転入力へ直接的に接続されます。
- **XREF** : 内部基準電圧出力。**内部基準電圧出力許可(XREFEN)**が設定(1)されている時で、2.56Vか1.1VのどちらかがA/D変換器の基準電圧として使われている場合に内部基準電圧の2.56Vまたは1.1Vが出力されます。
- **AREF** : A/D変換器用外部基準電圧入力。ピンがA/D変換器の外部基準電圧として使われる時、またはピンがA/D変換器の電圧基準に対する迂回コンデンサを接続するのに使われる時だけ、ポートピンのプルアップとデジタル出力駆動部が禁止されます。
- **PCINT7** : ピン変化割り込み7入力。

• **ポートA ピン6 : PA6 – \overline{SS} /ADC6/AIN0/PCINT6**

- **\overline{SS}** : SPI従装置選択入力。SPIが従装置として許可されると、本ピンは**ポートA方向レジスタ(DDRA)のDDA6**の設定に拘らず、入力として設定されます。従装置としてこのピンが**Low**に駆動される時に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DDA6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A出力レジスタ(PORTA)のPORTA6**によって制御できます。
- **ADC6** : A/D変換チャネル6入力。
- **AIN0** : **アナログ比較器**反転入力。このピンはアナログ比較器の反転入力へ直接的に接続されます。
- **PCINT6** : ピン変化割り込み6入力。

• **ポートA ピン5 : PA5 – T1/SCK/USCK/SCL/ADC5/PCINT5**

- **T1** : タイマ/カウンタ1の外部クロック入力。
- **SCK** : SPIチャネル用の主装置クロック出力、従装置クロック入力。SPIが従装置として許可されると、本ピンは**ポートA方向レジスタ(DDRA)のDDA5**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DDA5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A出力レジスタ(PORTA)のPORTA5**によって制御できます。
- **USCK** : 3線動作USIのクロック入出力。
- **SCL** : 2線動作USI(TWI)のクロック入出力。
- **ADC5** : A/D変換チャネル5入力。
- **PCINT5** : ピン変化割り込み5入力。

• **ポートA ピン4 : PA4 – ICP1/MOSI/DI/SDA/ADC4/PCINT4**

- **ICP1** : **タイマ/カウンタ1**の捕獲起動入力。PA3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。
- **MOSI** : **SPI**チャネル用の主装置データ出力、従装置データ入力。SPIが従装置として許可されると、このピンは**ポートA方向レジスタ(DDRA)のDDA4**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DDA4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A出力レジスタ(PORTA)のPORTA4**によって制御できます。
- **DI** : **3線動作USI**のデータ入力。3線動作USIは標準ポート機能を無効にしませんので、ピンはDI機能用入力として構成設定しなければなりません。
- **SDA** : **2線動作USI**のデータ入出力(オープンドレイン出力)。
- **ADC4** : A/D変換チャネル4入力。
- **PCINT4** : ピン変化割り込み4入力。

• **ポートA ピン3 : PA3 – INT1/ISRC/ADC3/PCINT3**

- **INT1** : 外部割り込み1入力。PA3ピンは外部割り込み元として扱えます。
- **ISRC** : **電流源**出力ピン。電流源部によって電流が供給されている間に使用者はピン電圧を測定するためにA/D変換器チャネル3(ADC3)を使うことができます。
- **ADC3** : A/D変換チャネル3入力。
- **PCINT3** : ピン変化割り込み3入力。

- **ポートA ピット2 : PA2 – OC0A/MISO/DO/ADC2/PCINT2**

- **OC0A** : **タイマ/カウンタ0**の比較A一致出力。PA2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、PA2ピンは出力として設定(**DDA2=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。
- **MISO** : **SPI**チャネル用の主装置データ入力、従装置データ出力。SPIが主装置として許可されると、このピンは**ポートA方向レジスタ(DDRA)のDDA2**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DDA2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A出力レジスタ(PORTA)のPORTA2**によって制御できます。
- **DO** : **3線動作USI**のデータ出力。データ出力(DO)は**PORTA2**値を無効にし、**データ方向(DDA2)ビット**が設定(**1**)されると、ポートが駆動されます。方向が入力でPORTA2が設定(**1**)されるなら、PORTA2は未だプルアップを許可します。
- **ADC2** : A/D変換チャネル2入力。
- **PCINT2** : ピン変化割り込み2入力。

- **ポートA ピット1 : PA1 – TXD/TXLIN/ADC1/PCINT1**

- **TXD** : UART送信ピン。UART送信部が許可されると、このピンは**ポートA方向レジスタ(DDRA)のDDA1**の値に拘らず、出力として構成設定されます。方向が入力で**PORTA1**が設定(**1**)されるなら、PORTA1は未だプルアップを許可します。
- **TXLIN** : LIN送信ピン。LINが許可される時に、このピンはポートA方向レジスタ(DDRA)のDDA1の値に拘らず、出力として構成設定されます。方向が入力でPORTA1が設定(**1**)されるなら、PORTA1は未だプルアップを許可します。
- **ADC1** : A/D変換チャネル1入力。
- **PCINT1** : ピン変化割り込み1入力。

- **ポートA ピット0 : PA0 – RXD/RXLIN/ADC0/PCINT0**

- **RXD** : **UART**受信ピン。UART受信部が許可されると、このピンは**ポートA方向レジスタ(DDRA)のDDA0**の値に拘らず、入力として構成設定されます。このピンが入力を強制されるとき、**PORTA0**の論理**1**は内部プルアップをONにします。
- **RXLIN** : **LIN**受信ピン。LINが許可される時に、このピンはポートA方向レジスタ(DDRA)のDDA0の値に拘らず、入力として構成設定されます。このピンが入力を強制されるとき、PORTA0の論理**1**は内部プルアップをONにします。
- **ADC0** : A/D変換チャネル0入力。
- **PCINT0** : ピン変化割り込み0入力。

表9-4.と表9-5.はポートAの交換機能を54頁の図9-6.で示される交換信号に関連付けます。

表9-4. ポートA7~4の交換機能用交換信号

信号名	PA7/ADC7/AIN1/XREF/ AREF/PCINT7	PA6/ \overline{SS} /ADC6/AIN0/ PCINT6	PA5/T1/SCK/USCK/ SCL/ADC5/PCINT5	PA4/ICP1/MOSI/DI/ SDA/ADC4/PCINT4
PUOE	0	SPE· \overline{MSTR}	SPE· \overline{MSTR}	SPE· \overline{MSTR}
PUOV	0	PORTA6· \overline{PUD}	PORTA5· \overline{PUD}	PORTA4· \overline{PUD}
DDOE	0	SPE· \overline{MSTR}	(SPE· \overline{MSTR}) + (2線USI·USIPOS)	SPE· \overline{MSTR} + (2線USI·USIPOS)
DDOV	0	0	(USI SCL保持+ $\overline{PORTA5}$) + DDA5	{{(SPE· \overline{MSTR})?(0): (USI移動出力+ $\overline{PORTA4}$)· DDA4}}
PVOE	0	0	(SPE· \overline{MSTR}) + (2線USI·USIPOS·DDA5)	(SPE· \overline{MSTR}) + (2線USI·USIPOS·DDA4)
PVOV	0	0	{{(SPE· \overline{MSTR})?(SCK出力): ~(2線USI·USIPOS·DDA5)}	{{(SPE· \overline{MSTR})?(MOSI出力): ~(2線USI·USIPOS·DDA4)}
PTOE	0	0	USI_PTOE·USIPOS	0
DIEOE	ADC7D+(PCINT7·PCIE0)	ADC6D+(PCINT6·PCIE0)	ADC5D+(USISIE·USIPOS) + (PCINT5·PCIE0)	ADC4D+(USISIE·USIPOS) + (PCINT4·PCIE0)
DIEOV	PCINT7·PCIE0	PCINT6·PCIE0	(USISIE·USIPOS) + (PCINT5·PCIE0)	(USISIE·USIPOS) + (PCINT4·PCIE0)
DI	PCINT7入力	\overline{SS} /PCINT6入力	T1/SCK/USCK/SCL/ PCINT5入力	ICP1/MOSI/DI/SDA/ PCINT4入力
AIO	ADC7/AIN1/XREF/AREF	ADC6/AIN0	ADC5	ADC4

表9-5. ポートA3~0の交換機能用交換信号

信号名	PA3/INT1/ISRC/ADC3/ PCINT3	PA2/OC0A/MISO/DO/ ADC2/PCINT2	PA1/TXD/TXLIN/ADC1/ PCINT1	PA0/RXD/RXLIN/ADC0/ PCINT0
PUOE	0	SPE· \overline{MSTR}	LIN送信許可	LIN受信許可
PUOV	PORTA3· \overline{PUD}	PORTA2· \overline{PUD}	{{(LIN送信許可)? (0):(PORTA1· \overline{PUD})}}	PORTA0· \overline{PUD}
DDOE	0	SPE· \overline{MSTR}	LIN送信許可	LIN受信許可
DDOV	0	0	LIN送信許可	0
PVOE	0	(SPE· \overline{MSTR}) + (2線USI· 3線USI·USIPOS) + OC0A	LIN送信許可	0
PVOV	0	{{(SPE· \overline{MSTR})?(MISO出力): ((2線USI·3線USI·USIPOS)? (USI移動出力):(OC0A))}}	{{(LIN送信許可)? (LIN送信):(0)}}	0
PTOE	0	0	0	0
DIEOE	ADC3D+INT1許可 + (PCINT3·PCIE0)	ADC2D+(PCINT2·PCIE0)	ADC1D+(PCINT1·PCIE0)	ADC0D+(PCINT0·PCIE0)
DIEOV	ADC3D+INT1許可 + (PCINT3·PCIE0)	PCINT2·PCIE0	PCINT1·PCIE0	PCINT0·PCIE0
DI	INT1/PCINT3入力	MISO/PCINT2入力	PCINT1入力	PCINT0入力
AIO	ISRC/ADC3	ADC2	ADC1	ADC0

9.3.4. ポートBの交換機能

ポートBピンの交換機能は表9-6.で示されます。

表9-6. 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	OC1BX (タイマ/カウンタ1 比較B-X/PWM出力)	PB3	OC1BV (タイマ/カウンタ1 比較B-V/PWM出力)
	ADC10 (A/D変換チャネル10入力)		PCINT11 (ピン変化割り込み11入力)
PB6	PCINT15 (ピン変化割り込み15入力)	PB2	OC1AV (タイマ/カウンタ1 比較A-V/PWM出力)
	RESET (リセット入力ピン)		USCK (3線動作USI既定クロック入出力)
PB5	dW (デバグWIRE入出力)		SCL (2線動作USI(TWI)既定クロック入出力)
	INT0 (外部割り込み0 入力)	PB1	PCINT10 (ピン変化割り込み10入力)
PB4	OC1AX (タイマ/カウンタ1 比較A-X/PWM出力)		OC1BU (タイマ/カウンタ1 比較B-U/PWM出力)
	ADC9 (A/D変換チャネル9入力)	PB0	DO (3線動作USI既定データ出力)
	PCINT14 (ピン変化割り込み14入力)		PCINT9 (ピン変化割り込み9入力)
PB3	XTAL2 (チップ クロック発振器ピン2)	PB2	OC1AU (タイマ/カウンタ1 比較A-U/PWM出力)
	CLKO (システム クロック出力)		DI (3線動作USI既定データ入力)
PB2	OC1BW (タイマ/カウンタ1 比較B-W/PWM出力)	PB1	SDA (2線動作USI(TWI)既定データ入出力)
	ADC8 (A/D変換チャネル8入力)		PCINT8 (ピン変化割り込み8入力)
PB1	PCINT13 (ピン変化割り込み13入力)	PB0	
	XTAL1 (チップ クロック発振器ピン1)		
PB0	CLKI (外部クロック入力)		

交換ピンの設定は次のとおりです。

• ポートB ビット7 : PB7 – OC1BX/ADC10/PCINT15/RESET/dW

- OC1BX : タイマ/カウンタ1用の比較出力とPWM出力B-X。この機能を扱うにはPB7ピンが出力(DDB7=1)として構成設定されなければなりません。このOC1BXピンはPWM動作形態タイマ機能に対する出力ピンでもあります(タイマ/カウンタ1制御レジスタD(TCCR1D)のOC1BX許可(OC1BX)ビット参照)。
- ADC10 : A/D変換チャネル10入力。
- PCINT15 : ピン変化割り込み15入力。
- RESET : リセット入力ピン。RSTDISBLヒューズがプログラム(0)されると、このピンは標準のI/Oピンとして機能し、デバイスはリセット元として電源ONリセットと低電圧検出(BOD)リセットに頼らなければなりません。RSTDISBLヒューズが非プログラム(1)にされると、このピンにリセット回路が接続され、このピンはI/Oピンとして使えません。

PB7がリセットピンとして使われると、PORTB7, DDB7, PINB7は全て0を読みます。

- dW : 施錠ビットが非プログラム(1)でデバグWIRE許可(DWEN)ヒューズがプログラム(0)されると、RESETポートピンはプルアップが許可されたワイヤードAND(オープントレイン)の双方向入出力ピンとして構成設定され、目的対象とエミュレータ間の通信交換器になります。

• ポートB ビット6 : PB6 – INT0/OC1AX/ADC9/PCINT14

- INT0 : 外部割り込み0入力。PB6ピンは外部割り込み元として扱えます。
- OC1AX : タイマ/カウンタ1用の比較出力とPWM出力A-X。この機能を扱うにはPB6ピンが出力(DDB6=1)として構成設定されなければなりません。このOC1AXピンはPWM動作形態タイマ機能に対する出力ピンでもあります(TCCR1DのOC1AX許可(OC1AX)ビット参照)。
- ADC9 : A/D変換チャネル9入力。
- PCINT14 : ピン変化割り込み14入力。

• ポートB ビット5 : PB5 – OC1BW/ADC8/PCINT13/XTAL2/CLKO

- OC1BW : タイマ/カウンタ1用の比較出力とPWM出力B-W。この機能を扱うにはPB5ピンが出力(DDB5=1)として構成設定されなければなりません。このOC1BWピンはPWM動作形態タイマ機能に対する出力ピンでもあります(TCCR1DのOC1BW許可(OC1BW)ビット参照)。
- XTAL2 : チップ クロック発振器ピン2。クリスタル用発振器または低周波数クリスタル用発振器用クロックピンとして使います。クロックピンとして使われると、このピンはI/Oピンとして使えません。
- CLKO : システムクロック出力。分周したシステムクロックはPB5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)された場合にPORTB5とDDB5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。
- ADC8 : A/D変換チャネル8入力。
- PCINT13 : ピン変化割り込み13入力。

• **ポートB ビット4 : PB4 – OC1AW/PCINT12/XTAL1/CLKI**

- **OC1AW** : タイマ/カウンタ1用の比較出力とPWM出力A-W。この機能を扱うにはPB4ピンが出力(**DDB4=1**)として構成設定されなければなりません。このOC1AWピンはPWM動作形態タイマ機能に対する出力ピンでもあります(TCCR1Dの**OC1AW許可(OC1AW)ビット**参照)。
- **PCINT12** : ピン変化割り込み12入力。
- **XTAL1** : チップ クロック発振器ピン1。校正付き内蔵RC発振器を除く全てのチップ クロック元に使われます。クロック ピンとして使われる時に、このピンはI/Oピンとして使えません。
- **CLKI** : 外部クロック入力。クロック ピンとして使われると、このピンはI/Oピンとして使えません。

注: PB4がクロック ピン(XTAL1またはCLKI)として使われる場合、**DDB4,PORTB4,PINB4**は全て**0**を読みます。

• **ポートB ビット3 : PB3 – OC1BV/PCINT11**

- **OC1BV** : タイマ/カウンタ1用の比較出力とPWM出力B-V。この機能を扱うにはPB3ピンが出力(**DDB3=1**)として構成設定されなければなりません。このOC1BVピンはPWM動作形態タイマ機能に対する出力ピンでもあります(TCCR1Dの**OC1BV許可(OC1BV)ビット**参照)。
- **PCINT11** : ピン変化割り込み11入力。

• **ポートB ビット2 : PB2 – OC1AV/USCK/SCL/PCINT10**

- **OC1AV** : タイマ/カウンタ1用の比較出力とPWM出力A-V。この機能を扱うにはPB2ピンが出力(**DDB2=1**)として構成設定されなければなりません。このOC1AVピンはPWM動作形態タイマ機能に対する出力ピンでもあります(TCCR1Dの**OC1AV許可(OC1AV)ビット**参照)。
- **USCK** : 3線動作USIのクロック入出力。
- **SCL** : 2線動作USI(TWI)のクロック入出力。
- **PCINT10** : ピン変化割り込み10入力。

• **ポートB ビット1 : PB1 – OC1BU/DO/PCINT9**

- **OC1BU** : タイマ/カウンタ1用の比較出力とPWM出力B-U。この機能を扱うにはPB1ピンが出力(**DDB1=1**)として構成設定されなければなりません。このOC1BUピンはPWM動作形態タイマ機能に対する出力ピンでもあります(TCCR1Dの**OC1BU許可(OC1BU)ビット**参照)。
- **DO** : 3線動作USIのデータ出力。データ出力(DO)は**PORTB1**値を無効にし、**データ方向(DDB1)ビット**が設定(**1**)されると、ポートが駆動されます。方向が入力で**PORTB1**が設定(**1**)されるなら、**PORTB1**は未だプルアップを許可します。
- **PCINT9** : ピン変化割り込み9入力。

• **ポートB ビット0 : PB0 – OC1AU/DI/SDA/PCINT8**

- **OC1AU** : **タイマ/カウンタ1**用の比較出力とPWM出力A-U。この機能を扱うにはPB0ピンが出力(**DDB0=1**)として構成設定されなければなりません。このOC1AUピンはPWM動作形態タイマ機能に対する出力ピンでもあります(TCCR1Dの**OC1AU許可(OC1AU)ビット**参照)。
- **DI** : **3線動作USI**のデータ入力。3線動作USIは標準ポート機能を無効にしませんので、ピンはDI機能用入力として構成設定しなければなりません。
- **SDA** : **2線動作USI**のデータ入出力(オープンドレイン出力)。
- **PCINT8** : ピン変化割り込み8入力。

表9-7.と表9-8.はポートBの交換機能を54頁の図9-6.で示される交換信号に関連付けます。

表9-7. ポートB7～4の交換機能用交換信号

信号名	PB7/OC1BX/ADC10/ PCINT15/RESET/dW	PB6/INT0/OC1AX/ADC9/ PCINT14	PB5/OC1BW/ADC8/ PCINT13/XTAL2/CLKO	PB4/OC1AW/PCINT12/ XTAL1/CLKI
PUOE	0	0	0	0
PUOV	0	0	0	0
DDOE	0	0	0	0
DDOV	0	0	0	0
PVOE	OC1B許可・OC1BX	OC1A許可・OC1AX	OC1B許可・OC1BW	OC1A許可・OC1AW
PVOV	OC1B	OC1A	OC1B	OC1A
PTOE	0	0	0	0
DIEOE	ADC10D+ (PCINT15・PCIE1)	ADC9D+INT0許可+ (PCINT14・PCIE1)	ADC8D+(PCINT13・PCIE1)	PCINT12・PCIE1
DIEOV	PCINT15・PCIE1	INT0許可+ (PCINT14・PCIE1)	PCINT13・PCIE1	1
DI	PCINT15入力	INT0/PCINT14入力	PCINT13入力	PCINT12入力
AIO	ADC10/RESET	ADC9	ADC8/XTAL2	XTAL1/CLKI

表9-8. ポートB3～0の交換機能用交換信号

信号名	PB3/OC1BV/PCINT11	PB2/OC1AV/USCK/SCL/ PCINT10	PB1/OC1BU/DO/PCINT9	PB0/OC1AU/DI/SDA/ PCINT8
PUOE	0	0	0	0
PUOV	0	0	0	0
DDOE	0	2線USI・USIPOS	0	2線USI・USIPOS
DDOV	0	(USI SCL保持+PORTB2)+ DDB2	0	(USI移動出力+PORTB0)・ DDB0
PVOE	OC1B許可・OC1BV	(2線USI・USIPOS・DDB2)+ (OC1A許可・OC1AV)	(2線USI・3線USI・USIPOS)+ (OC1B許可・OC1BU)	(2線USI・USIPOS・DDB0)+ (OC1A許可・OC1AU)
PVOV	OC1B	{{(2線USI・USIPOS・DDB2)? (0):(OC1A)}}	{{(2線USI・3線USI・USIPOS)? (USI移動出力):(OC1B)}}	{{(2線USI・USIPOS・DDB0)? (0):(OC1A)}}
PTOE	0	USI_PTOE・USIPOS	0	0
DIEOE	PCINT11・PCIE1	(USISIE・USIPOS)+ (PCINT10・PCIE1)	PCINT9・PCIE1	(USISIE・USIPOS)+ (PCINT8・PCIE1)
DIEOV	1	(USISIE・USIPOS)+ (PCINT10・PCIE1)	1	(USISIE・USIPOS)+ (PCINT8・PCIE1)
DI	PCINT11入力	USCK/SCL/PCINT10入力	PCINT9入力	DI/SDA/PCINT8入力
AIO	—	—	—	—

9.4. I/Oポート用レジスタ

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

ビット	7	6	5	4	3	2	1	0	
\$02 (\$22)	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	PORTA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

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

ビット	7	6	5	4	3	2	1	0	
\$01 (\$21)	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	DDRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

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

ビット	7	6	5	4	3	2	1	0	
\$00 (\$20)	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	PINA
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	

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

ビット	7	6	5	4	3	2	1	0	
\$05 (\$25)	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	PORTB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

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

ビット	7	6	5	4	3	2	1	0	
\$04 (\$24)	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	DDRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

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

ビット	7	6	5	4	3	2	1	0	
\$03 (\$23)	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	PINB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	不定	不定	不定	不定	不定	不定	不定	不定	

10.2.2. 定義

表10-1.の定義は本資料を通して広範囲に渡って使われます。

表10-1. 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。

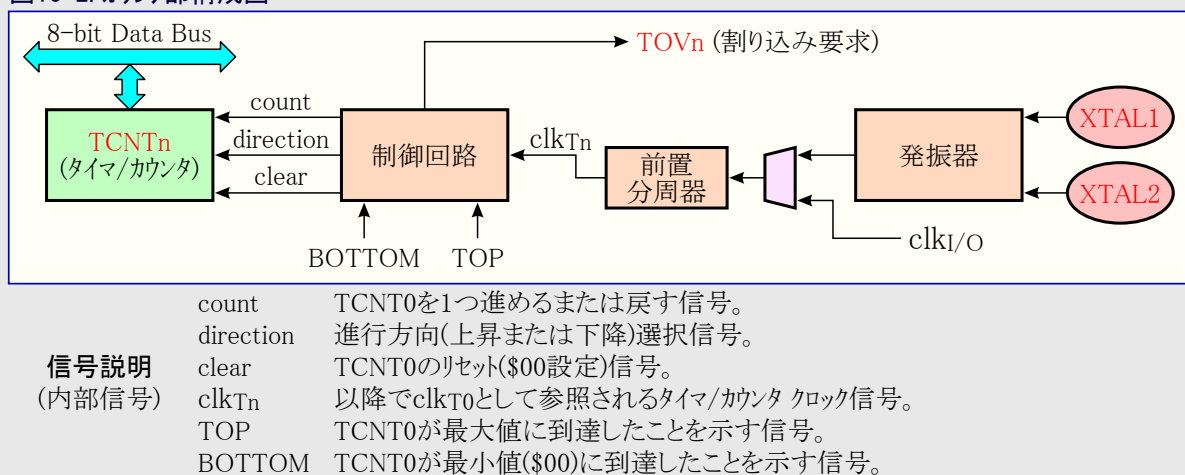
10.3. タイマ/カウンタのクロック

このタイマ/カウンタは内部同期または外部非同期クロック元によってクロック駆動できます。クロック元は**タイマ/カウンタ0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択されます。既定のクロック元(clkT0)はMCUクロック(clkI/O)と同じです。非同期状態レジスタ(ASSR)の**非同期動作許可(AS0)ビット**が論理1を書かれると、クロック元はXTAL1とXTAL2に繋がったタイマ/カウンタ用発振器からまたはXTAL1から直接的に取得されます。非同期動作の詳細については75頁の「**ASSR – 非同期状態レジスタ**」をご覧ください。クロック元と前置分周器の詳細については72頁の「**タイマ/カウンタ0の前置分周器**」をご覧ください。

10.4. 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10-2.は、このカウンタとその周辺環境の構成図を示します。

図10-2. カウンタ部構成図



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)ビットによって選択された内部または外部のクロック元から生成できます。クロック元が選択されない(CS02~0=000)時にタイマ/カウンタは停止されます。けれどもTCNT0値はタイマ/カウンタ クロック(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(TCCR0A)**に配置された**波形生成種別(WGM01,0)ビット**の設定によって決定されます。これらはカウンタ動作(計数)方法とOCR0A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67頁の「**動作種別**」をご覧ください。

タイマ/カウンタ溢れ(TOV0)フラグはWGM01,0ビットによって選択された動作種別に従って設定(=1)されます。TOV0はCPU割り込み発生に使えます。

10.5. 比較出力部

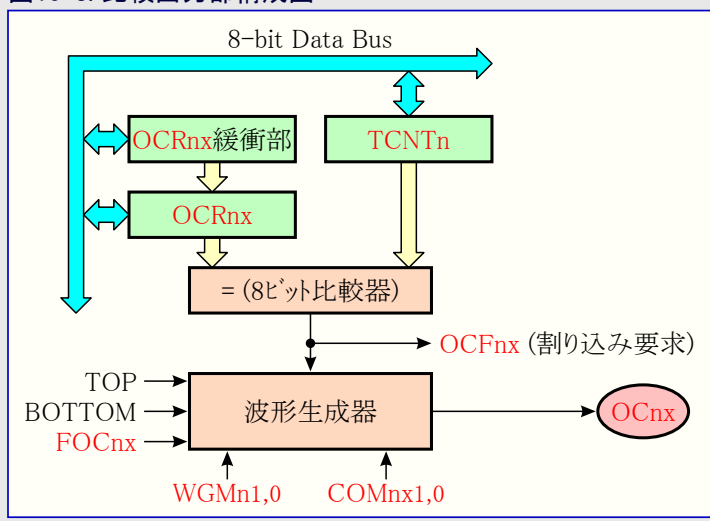
この8ビット比較器はTCNT0と比較レジスタ(OCR0A)を継続的に比較します。TCNT0とOCR0Aが等しければ、比較器は一致を指示します。この一致は次のタイマ/カウンタクロック周期で比較割り込み要求フラグ(OCF0A)を設定(1)します。許可(I=1, OCIE0A=1)なら、この比較割り込み要求フラグは比較割り込みを発生します。OCF0Aは割り込みが実行されると自動的に解除(0)されます。代わりにOCF0Aはこのビット位置に論理1を書くことによってソフトウェアでも解除(0)できます。波形生成器は波形生成種別(WGM01,0)ビットと比較出力選択(COM0A1,0)ビットで設定された動作種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(67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10-3.は比較出力部の構成図を示します。

OCR0Aはパルス幅変調(PWM)のどれかを使う時に2重緩衝されます。標準動作と比較一致タイマ/カウンタ解除(CTC)動作については2重緩衝動作が禁止されます。2重緩衝動作は計数の流れのTOPまたはBOTTOMのどちらかに対してOCR0Aレジスタの更新を同期化します。この同期化は奇数長や非対称PWMパルスの発生を防ぎ、それによって不具合なしの出力を作成します。

OCR0A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OCR0A緩衝部をアクセスし、禁止されるとOCR0Aレジスタを直接アクセスします。

図10-3. 比較出力部構成図



10.5.1. 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(FOC0A)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(OCF0A)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、OC0A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(COM0A1,0ビット設定がOC0Aピンの設定(1)、解除(0)、1/0交互のどれかを定義)。

10.5.2. 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OCR0Aに許します。

10.5.3. 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にかかわらず、比較出力部を使う場合にTCNT0を変更する時は危険を伴います。TCNT0に書かれた値がOCR0A値と同じ場合、比較一致は失われ(一致が発生せず)、不正な波形生成に終わります。同様にタイマ/カウンタが下降計数のとき、BOTTOMに等しいTCNT0値を書いてはいけません。

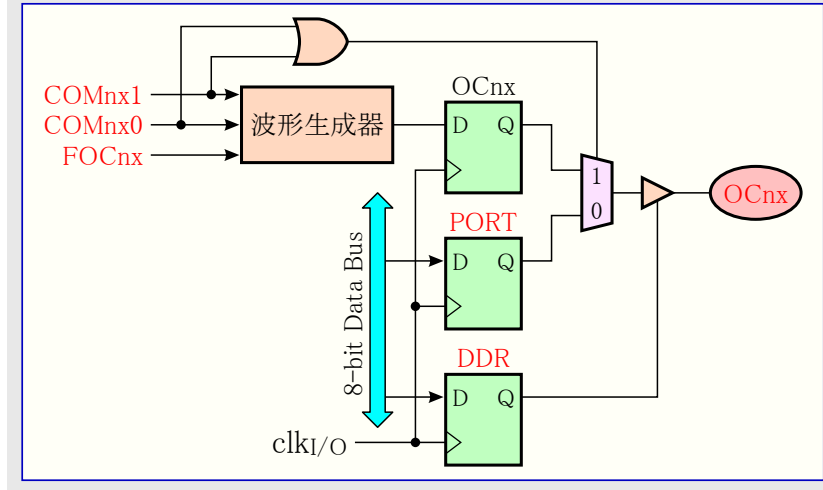
OC0Aの初期設定はポートピンに対するポート方向レジスタを出力に設定する前に行われるべきです。OC0A値を設定する一番簡単な方法は標準動作で強制変更(FOC0A)スローブビットを使うことです。波形生成動作種別間を変更する時であっても、OC0A(内部)レジスタはその値を保ちます。

比較出力選択(COM0A1,0)ビットが比較値(OCR0A)と共に2重緩衝されないことに気付いてください。COM0A1,0ビットの変更は直ちに有効となります。

10.6. 比較一致出力部

比較出力選択(COM0A1,0)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(OC0A)状態の定義にCOM0A1,0ビットを使います。またCOM0A1,0ビットはOC0Aピン出力元を制御します。図10-4はCOM0A1,0ビット設定によって影響を及ぼされる論理回路の簡略化した図を示します。図のI/Oレジスタ、I/Oビット、I/Oピンは赤文字(訳注:原文は太字)で示されます。COM0A1,0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。OC0Aの状態を参照するとき、その参照はOC0Aピンでなく内部OC0Aレジスタに対してです。

図10-4. 比較一致出力回路図



10.6.1. 比較出力機能

COM0A1,0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(OC0A)によって無効にされます。けれどもOC0A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。OC0Aピンに対するポート方向レジスタのビット(DDR_OC0A)はOC0A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OC0A状態の初期化を許します。いくつかのCOM0A1,0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。73頁の「8ビット タイマ/カウンタ用レジスタ」をご覧ください。

10.6.2. 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COM0A1,0ビットを違うふうに使います。全ての動作種別に対してCOM0A1,0=00設定は次の比較一致で実行すべきOC0A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73頁の表10-2を参照してください。高速PWM動作については73頁の表10-3、位相基準PWMについては73頁の表10-4を参照してください。

COM0A1,0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(FOC0A)スロー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。

10.7. 動作種別

動作種別(換言するとタイマ/カウンタと比較出力ピンの動作)は**波形生成種別(WGM01,0)**ビットと**比較出力選択(COM0A1,0)**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。COM0A1,0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COM0A1,0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(66ページの「**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70ページの「**タイマ/カウンタのタイミング**」を参照してください。

10.7.1. 標準動作

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

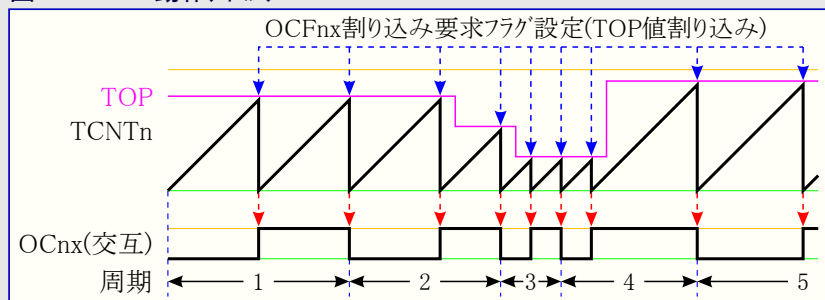
比較出力部は与えられた或る時間に割り込みを生成するのに使えます。標準動作で波形を生成するのに比較出力を使うのは、それが大変多くのCPU時間を占有するため推奨されません。

10.7.2. 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(WGM01,0=10)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10-5.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、そしてその後カウンタ(TCNT0)は解除(\$00)されます。

図10-5. 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)に設定されるとき、 $f_{OC0A} = f_{clk_I/O} / 2$ 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(1,8,32,64,128,256,1024)を表します。

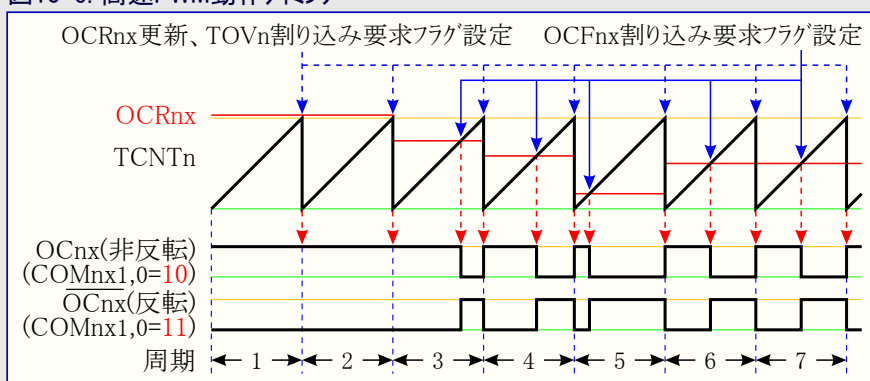
標準動作と同じように、**タイマ/カウンタ溢れ(TOV0)フラグ**はカウンタがMAXから\$00へ計数するのと同じタイマ/カウンタクロック周期で設定(1)されます。

10.7.3. 高速PWM動作

高速パルス幅変調(PWM)動作(WGM01,0=11)は高周波数PWM波形生成選択を提供します。高速PWMはそれが単一傾斜(鋸波)動作であることによって他のPWM動作と異なります。カウンタはBOTTOMからMAXまで計数し、その後にBOTTOMから再び始めます。非反転比較出力動作(COM0A1,0=10)での比較出力(OC0A)は、TCNT0とOCR0A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(COM0A1,0=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MAX値と一致するまで増加されます。そしてタイマ/カウンタは(一致の)次のタイマ/カウンタクロック周期で解除(\$00)されます。高速PWM動作のタイミング図は図10-6.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OCR0A値を示し、TCNT0値との交点(接点)がTCNT0とOCR0A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(OCF0A)はOCR0A=MAXを除いて比較一致が起こると設定(1)されます(訳注:共通性のため本行追加)。

図10-6. 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MAXに到達する時毎に設定(1)されます。割り込みが許可されるなら、その割り込み処理ルーチンは比較値を更新するのに使えます。

高速PWM動作での比較部はOC0AピンでのPWM波形の生成を許します。COM0A1,0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COM0A1,0を'11'に設定することで生成できます(73頁の表10-3.をご覧ください)。実際のOC0A値はポートピンに対するデータ方向(DDR_OC0A)が出力として設定される場合にだけ見えるでしょう。PWM波形はTCNT0とOCR0A間の比較一致でOC0A(内部)レジスタを設定(1)(または解除(0))と、カウンタが解除(\$00、MAXからBOTTOMへ変更)されるタイマ/カウンタクロック周期でOC0A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。

$$f_{OCnxPWM} = \frac{f_{clk_I/O}}{N \times 256}$$

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

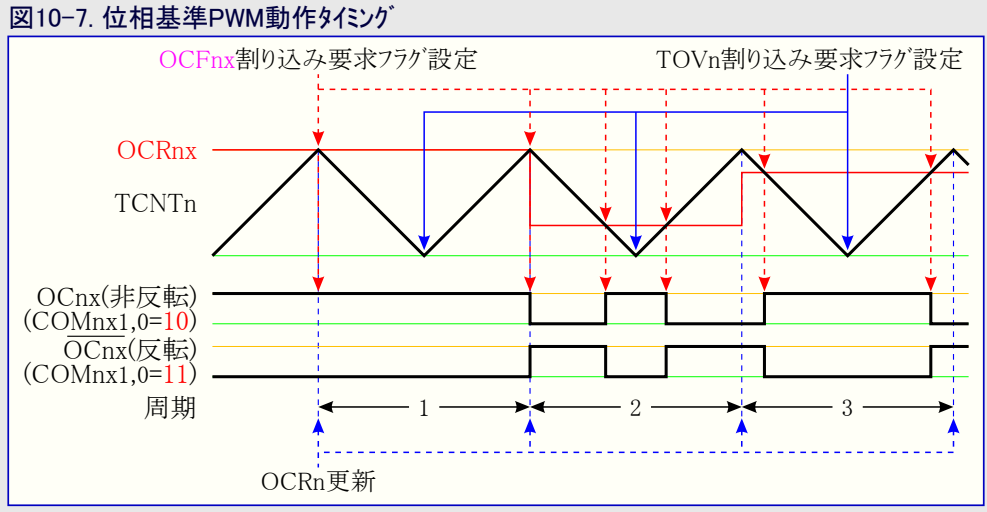
OCR0Aの両端値は高速PWM動作でPWM波形出力を生成する時の特別な場合に当たります。OCR0AがBOTTOM(\$00)と等しく設定されると、出力はMAX+1 タイマ/カウンタクロック周期毎の狭いスパイク(パルス)になるでしょう。OCR0AがMAXに等しく設定されると、(COM0A1,0ビットによって設定される出力極性に依存して)定常的なLowまたはHigh出力に終わるでしょう。

(訳注:以下は本動作種別がTOP=MAX(\$FF)で固定のため基本的に不適切です。) 高速PWM動作で(デューティ比50%)周波数の波形出力は比較一致毎に論理反転する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 $f_{OC0A} = f_{clk_I/O} / 2$ 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。

10.7.4. 位相基準PWM動作

位相基準パルス幅変調(PWM)動作(WGM01,0=01)は高分解能で正しい位相のPWM波形生成選択を提供します。位相基準PWM動作は両傾斜(三角波)動作に基づきます。カウンタはBOTTOMからMAXへそして次にMAXからBOTTOMへを繰り返し計数します。非反転比較出力動作(COM0A1,0=10)での比較出力(OC0A)は上昇計数中のTCNT0とOCR0A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(COM0A1,0=11)での動作は逆にされます。両傾斜(三角波)動作は単一傾斜(鋸波)動作よりも低い最大動作周波数になります。けれども両傾斜(三角波)動作の対称特性のため、これらの動作種別は電動機制御の応用に好まれます。

位相基準PWM動作のPWM分解能は8ビットに固定されます。位相基準PWM動作でのカウンタはカウンタ値がMAXと一致するまで増加されます。カウンタはMAXに到達すると計数方向を変えます。このTCNT0値は1タイマ/カウンタクロック周期間MAXと等しくなります。位相基準PWM動作のタイミング図は図10-7.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OCR0A値を示し、TCNT0値との交点(接点)がTCNT0とOCR0A間の比較一致を示します(訳注:図補正に伴い本行若干変更)。



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

位相基準PWM動作での比較部はOC0AピンでのPWM波形の生成を許します。COM0A1,0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COM0A1,0ビットを'11'に設定することで生成できます(73頁の表10-4.をご覧ください)。実際のOC0A値はそのポートピンに対するデータ方向(DDR_OC0A)が出力として設定される場合にだけ見えるでしょう。PWM波形はカウンタが増加する時のTCNT0とOCR0A間の比較一致でOC2A(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OCR0A間の比較一致でOC0A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。

$$f_{OCnxPCPWM} = \frac{f_{clk_I/O}}{N \times 510}$$

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

OCR0A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OCR0AがBOTTOM(\$00)に等しく設定されると出力は定常的なLow、MAXに等しく設定されると定常的なHighになるでしょう。反転PWMに対する出力は逆の論理値になります。(訳注:共通性から以下追加)

図10-7.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10-7.のようにOCR0AはTOPからその値を変更します。OCR0A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OCR0A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。)

10.8. タイマ/カウンタのタイミング

以下の図は同期動作でのタイマ/カウンタを示し、従ってタイマ/カウンタクロック(clk_{T0})が計数許可信号として示されます。非同期動作では $\text{clk}_{I/O}$ がタイマ/カウンタ発振器クロックによって置換されるべきです。この図は割り込みフラグが設定(1)される時の情報を含みます。図10-8. は基本的なタイマ/カウンタ動作についてのタイミング データを示します。この図は位相基準PWM動作以外の全ての動作種別でのMAX値 近辺の計数の流れを示します。

図10-8. 前置分周なし(1/1)のタイマ/カウンタ タイミング

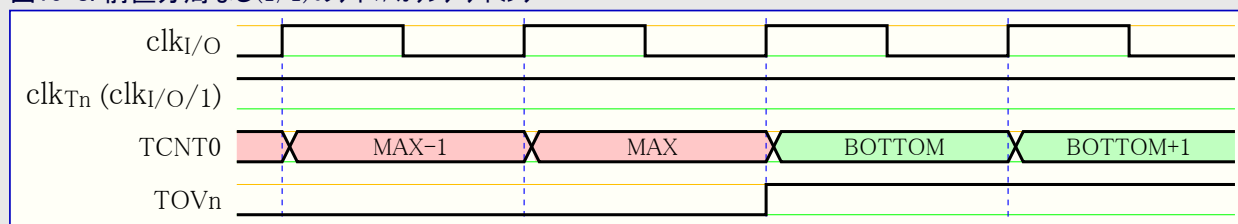


図10-9. は同じタイミング データを示しますが、前置分周器が許可されています。

図10-9. 前置分周器($f_{\text{clk}_{I/O}}/8$)のタイマ/カウンタ タイミング

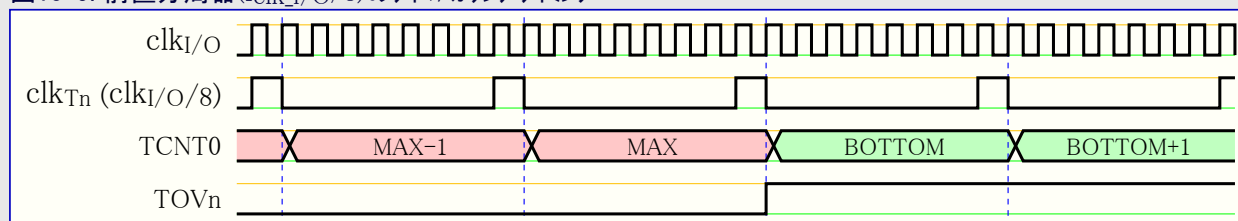


図10-10. はCTC動作を除く全ての動作種別でのOCF0Aの設定を示します。

図10-10. 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

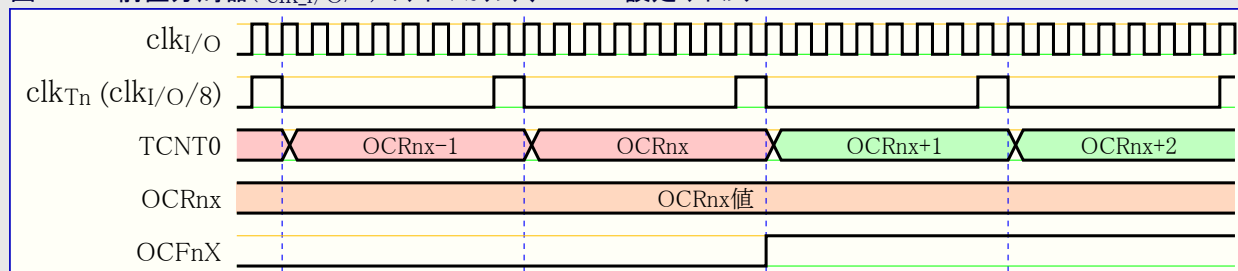
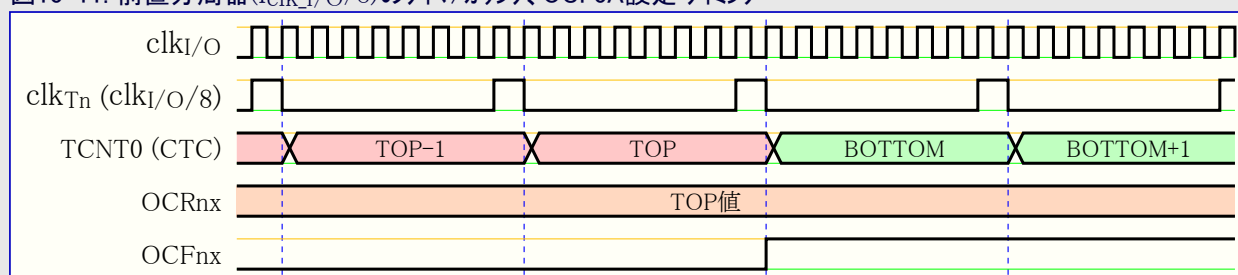


図10-11. はCTC動作でのTCNT0の解除とOCF0Aの設定を示します。

図10-11. 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



10.9. タイマ/カウンタ0の非同期動作

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

- ・ **警告:** タイマ/カウンタ0の同期/非同期クロック駆動間を切り替えると、**タイマ/カウンタ0(TCNT0)**、**タイマ/カウンタ0比較レジスタ(OCR0A)**、**タイマ/カウンタ0制御レジスタA/B(TCCR0A,TCCR0B)**が不正となるかもしれません。クロック元を切り替える安全な手順を次に示します。
 1. **タイマ/カウンタ0割り込み許可レジスタ(TIMSK0)**の**OCIE0A**と**TOIE0**の解除(0)により、タイマ/カウンタ0割り込みを禁止します。
 2. **非同期状態レジスタ(ASSR)**の**非同期動作許可(AS0)**設定によってクロック元を適切に選択します。
 3. **TCNT0,OCR0A,TCCR0A,TCCR0B**に新しい値を書きます。
 4. 非同期動作へ切り替えるには、**TCN0UB, OCR0AUB, TCR0AUB, TCR0BUB**について(=0まで)待機します。
 5. **タイマ/カウンタ0割り込み要求フラグ レジスタ(TIFR0)**の**OCF0A**と**TOV0**フラグを解除(0)します。
 6. 必要とされるなら、割り込みを許可します。
- ・ CPU主クロック周波数はタイマ発振器周波数の4倍よりも高くなければなりません。
- ・ **TCNT0,OCR0A,TCCR0A,TCCR0B**レジスタの1つに書くとき、その値は一時レジスタへ転送され、XTAL1で2つの上昇端後、(実レジスタ)に設定されます。一時レジスタの内容がそれらの転送先へ転送されてしまう前に新しい値を書くべきではありません。記載された4つのレジスタの各々がそれら個別の一時レジスタを持ちます。それは、例えば**TCNT0**書き込みが**OCR0A**書き込みの実行を妨げないことを意味します。転送先レジスタへの転送が起きたことを検知するために非同期状態レジスタ(ASSR)は実装されました。
- ・ **TCNT0,OCR0A,TCCR0A,TCCR0B**に書いた後で**パワーセーブ動作**または**A/D変換雑音低減動作**へ移行するとき、デバイスを起動するのにタイマ/カウンタ0が使われる場合、使用者は書かれたレジスタが更新されてしまうまで待たなければなりません。さもなければMCUは変更が有効になる前に**休止形態**へ移行するでしょう。タイマ/カウンタ0比較一致割り込みがデバイスを起動するのに使われる場合、**TCNT0**または**OCR0A**書き込み中の比較機能が禁止されるため、これは非常に重要です。書き込み周期が完了されず、**OCR0AUB**が0に戻る前にMCUが**休止形態**へ移行すると、デバイスは決して比較一致割り込みを受け取らず、そしてMCUは起動しないでしょう。
- ・ **パワーセーブ**または**A/D変換雑音低減動作**からデバイスを起動するのにタイマ/カウンタ0が使われる場合、使用者がこれら動作へ(復帰後)再移行することを欲するなら、予防処置を講じなければなりません。割り込み論理回路はリセットするために1 XTAL1周期必要です。起動と休止形態再移行間の時間が1 XTAL1周期未満の場合に割り込みが起きず、デバイスは起動に失敗するでしょう。**パワーセーブ**または**A/D変換雑音低減動作**再移行前の時間が充分であるかどうか迷うなら、1 XTAL1周期が経過されるのを保証するために次の手順が使えます。
 1. **TCNT0,OCR0A,TCCR0A,TCCR0B**に値を書きます。
 2. 非同期状態レジスタ(ASSR)の対応する更新中フラグが0に戻るまで待ちます。
 3. **パワーセーブ**または**A/D変換雑音低減動作**へ移行します。
- ・ **非同期動作が選択**されるとき、タイマ/カウンタ0用32.768kHz発振器は**パワーダウン動作**を除いて常に動作します。電源投入リセット、パワーダウン動作から起動後、この発振器が安定するのに1秒程度かかるかもしれないことを使用者は承知すべきです。電源投入、パワーダウン動作から起動後、タイマ/カウンタ0を使う前に少なくとも1秒待機することが推奨されます。この発振器の使用またはクロック信号がXTALピンに印加されるかのどちらかに拘らず、起動時の不安定的なクロック信号のため、パワーダウン動作からの起動復帰後、タイマ/カウンタ0の全レジスタの内容が失われたとみなされなければなりません。
- ・ タイマ/カウンタ0が非同期でクロック駆動される時の**パワーセーブ**または**A/D変換雑音低減動作**から起動の説明。割り込み条件が合致すると、タイマ/カウンタ0クロックの次の周期で起動処理が開始され、プロセッサがカウンタ値を読めるのに先立ってタイマ/カウンタは常に最低1進行されます。起動後MCUは4周期停止され、割り込み処理ルーチンを実行し、そして**SLEEP**命令の次の命令から実行を再開します。
- ・ **パワーセーブ動作**から起動直後の**TCNT0**の読み込みは不正な結果を得るかもしれません。**TCNT0**が非同期**TOSC**クロックでクロック駆動されるため、**TCNT0**読み込みは内部I/Oクロック領域に同期したレジスタを通して行われなければなりません。同期化はXTALクロックの全上昇端で行われます。**パワーセーブ動作**から起動し、I/Oクロック(**clk_{I/O}**)が再び活性(有効)になると、**TCNT0**はXTALクロックの次の上昇端まで以前(休止形態移行前)の値を読むでしょう。**パワーセーブ動作**から起動後のXTALクロックの位相は起動時間に依存するため本質的に特定できません。従って**TCNT0**読み込みに対する推奨手順は次のとおりです。
 1. **OCR0A, TCCR0A**または**TCCR0B**のどれかに何か値を書きます。
 2. 非同期状態レジスタ(ASSR)の対応する更新中フラグが解除(0)されるまで待ちます。
 3. **TCNT0**を読みます。
- ・ 非同期動作の間中、非同期タイマ用割り込み要求フラグの同期化は3プロセッサ周期+1タイマ周期かかります。従ってプロセッサが割り込み要求フラグ設定の原因となったタイマ値を読めるのに先立って、このタイマは最低1、進行されます。比較出力ピンはタイマクロックで変更され、プロセッサクロックに同期されません。

10.10. タイマ/カウンタ0の前置分周器

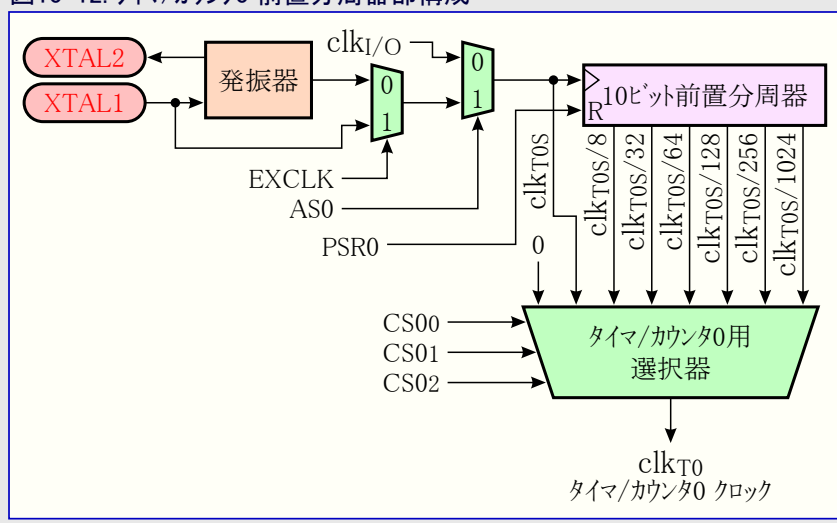
タイマ/カウンタ0用クロック元の名前は clkT0S です。既定での clkT0S は主システム I/Oクロック(clkI/O)に接続されます。**タイマ/カウンタ0非同期状態レジスタ(ASSR)の非同期クロック(AS0)ビット**の設定(1)により、タイマ/カウンタ0はXTAL発振器またはXTAL1ピンから非同期にクロック駆動されます。これは実時間計数器(RTC)としてタイマ/カウンタ0の使用を可能にします。

そして水晶発振子はタイマ/カウンタ0用の独立したクロック元として扱うため、XTAL1とXTAL2ピン間に接続できます。

入力としてXTAL1を用いて外部クロックを使うこともできます。AS0と**外部クロック許可(EXCLK)**の設定(1)がこの設定を許可します。

タイマ/カウンタ0に対して可能な前置分周済み選択は $\text{clkT0S}/8, \text{clkT0S}/32, \text{clkT0S}/64, \text{clkT0S}/128, \text{clkT0S}/256, \text{clkT0S}/1024$ で、加えて、0(停止)は勿論 clkT0S も選択可能です。**一般タイマ/カウンタ制御レジスタ(GTCCR)のタイマ/カウンタ0前置分周器リセット(PSR0)ビット**の設定(1)は前置分周器をリセットします。これは予測可能な前置分周器での操作を使用者に許します。

図10-12. タイマ/カウンタ0 前置分周器部構成



10.11. 8ビット タイマ/カウンタ0用レジスタ

10.11.1. TCCR0A – タイマ/カウンタ0制御レジスタA (Timer/Counter0 Control Register A)

ビット	7	6	5	4	3	2	1	0	
\$25 (\$45)	COM0A1	COM0A0	–	–	–	–	WGM01	WGM00	TCCR0A
Read/Write	R/W	R/W	R	R	R	R	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

●ビット5,4 – COM0A1,0 : 比較0A出力選択 (Compare Match 0A 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WGM01,0ビット設定に依存します。

表10-2.はWGM01,0ビットが標準動作またはCTC動作(即ちPWM以外)に設定される時のCOM0A1,0ビット機能を示します。

表10-3.はWGM01,0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10-4.はWGM01,0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10-2. 非PWM動作での比較出力選択

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10-3. 高速PWM動作での比較出力選択

COM0A1	COM0A0	意味
0	0	標準ポート動作 (OC0A切断)
0	1	(予約)
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)

注: COM0A1が設定(1)され、OCR0A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)は行われます。より多くの詳細については68頁の「高速PWM動作」をご覧ください。

表10-4. 位相基準PWM動作での比較出力選択

COM2A1	COM2A0	意味
0	0	標準ポート動作 (OC0A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

注: COM0A1が設定(1)され、OCR0A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、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69頁の「位相基準PWM動作」をご覧ください。

●ビット5~2 – Res : 予約 (Reserved)

これらのビットは予約されており、常に0として読まれます。

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

これらのビットはカウンタの計数順序(方向)、最大(TOP)カウンタ値の供給元、使われるべき波形生成のどの形式かを制御します。タイマ/カウンタ部によって支援される動作種別は標準動作、比較一致タイマ/カウンタ解除(CTC)動作と、2形式のパルス幅変調(PWM)動作です。表10-5.と67頁の「動作種別」をご覧ください。

表10-5. 波形生成種別選択

番号	WGM01 (CTC0)	WGM00 (PWM0)	タイマ/カウンタ動作種別	TOP値	OCR0A 更新時	TOV2 設定時
0	0	0	標準動作	\$FF	即時	MAX
1	0	1	位相基準PWM動作	\$FF	TOP	BOTTOM
2	1	0	比較一致タイマ/カウンタ解除(CTC)動作	OCR0A	即時	MAX
3	1	1	高速PWM動作	\$FF	BOTTOM	MAX

注: MAX=\$FF, BOTTOM=\$00

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

ビット	7	6	5	4	3	2	1	0	
\$26 (\$46)	FOC0A	–	–	–	–	CS02	CS01	CS00	TCCR0B
Read/Write	W	R	R	R	R	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

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

FOC0AビットはWGM01,0ビットが非PWM動作を指示する時だけ有効です。けれども将来のデバイスとの共通性を保証するため、PWM動作で扱う時に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ビットがスト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スト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として読まれます。

● ビット6~3 – Res : 予約 (Reserved)

これらのビットは予約されており、常に0として読まれます。

● 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択します。表10-6をご覧ください。

表10-6. タイマ/カウンタ2入力クロック選択

CS02	CS01	CS00	意味
0	0	0	停止 (タイマ/カウンタ動作停止)
0	0	1	clk _{T2S} (前置分周なし)
0	1	0	clk _{T2S} /8 (8分周)
0	1	1	clk _{T2S} /32 (32分周)
1	0	0	clk _{T2S} /64 (64分周)
1	0	1	clk _{T2S} /128 (128分周)
1	1	0	clk _{T2S} /256 (256分周)
1	1	1	clk _{T2S} /1024 (1024分周)

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

ビット	7	6	5	4	3	2	1	0	
(\$B2)	(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OCR0A間の比較一致消失の危険を誘発します。

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

ビット	7	6	5	4	3	2	1	0	
(\$B3)	(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。

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

ビット (\$B6)	7	6	5	4	3	2	1	0	
	–	EXCLK	AS0	TCN0UB	OCR0AUB	–	TCR0AUB	TCR0BUB	ASSR
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 – Res : 予約 (Reserved)

このビットは将来の使用に予約されており、常に0を読みます。

● ビット6 – EXCLK : 外部クロック信号許可 (Enable External Clock Input)

EXCLKが1を書かれ、非同期クロックが選択されると、外部クロック入力緩衝部が許可され、32kHzクリスタルの代わりに外部クロックをXTAL1ピンへ入力できます。EXCLKへの書き込みは非同期動作が選択される前に行うべきです。クリスタル発振器はこのビットが0の時にだけ走行(動作)することに注意してください。

● ビット5 – AS0 : タイマ/カウンタ0非同期動作許可 (Asynchronous Timer/Counter0)

AS0が0を書かれると、タイマ/カウンタ0はI/Oクロック(clk_{I/O})からクロック駆動され、タイマ/カウンタ0は同期周辺機能として働きます。

AS0が1を書かれると、外部クロック信号許可(EXCLK)ビットの設定に依存して、タイマ/カウンタ0は低周波数クリスタル用発振器(25頁の「低周波数クリスタル用発振器」をご覧ください)、またはXTAL1ピン上の外部クロック(25頁の「外部クロック信号」をご覧ください)からクロック駆動されます。AS0の値が変更されると、タイマ/カウンタ0(TCNT0)、比較0レジスタ(OCR0A)、タイマ/カウンタ0制御レジスタA(TCCR0A)の内容は不正にされるかもしれません。

AS0はフラグとしても働きます。タイマ/カウンタ0はシステムクロックを駆動するのに校正付き内蔵RC発振器または内部ウォッチドッグ発振器が使用されている場合にだけ、低周波数クリスタルまたは外部クロックからクロック駆動されます。AS0設定(1)後、切り替えが可能ならば、AS0は1を返し、さもなければ0に強制されます。

● ビット4 – TCN0UB : タイマ/カウンタ0更新中フラグ (Timer/Counter0 Update Busy)

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

● ビット3 – OCR0AUB : 比較0レジスタ更新中フラグ (Output Compare 0 Register A Update Busy)

タイマ/カウンタ0が非同期に動き、比較0レジスタ(OCR0A)が書かれると、このビットが設定(1)になります。OCR0Aが一時保存レジスタから更新されてしまうと、このビットはハードウェアによって解除(0)されます。このビットの論理0はOCR0Aが新しい値で更新される用意ができたことを示します。

● ビット2 – Res : 予約 (Reserved)

このビットは将来の使用に予約されており、常に0を読みます。

● ビット1 – TCR0AUB : タイマ/カウンタ0制御レジスタA更新中フラグ (Timer/Counter0 Control Register A Update Busy)

タイマ/カウンタ0が非同期に動き、タイマ/カウンタ0制御レジスタA(TCCR0A)が書かれると、このビットが設定(1)になります。TCCR0Aが一時保存レジスタから更新されてしまうと、このビットはハードウェアによって解除(0)されます。このビットの論理0はTCCR0Aが新しい値で更新される用意ができたことを示します。

● ビット0 – TCR0BUB : タイマ/カウンタ0制御レジスタB更新中フラグ (Timer/Counter0 Control Register B Update Busy)

タイマ/カウンタ0が非同期に動き、タイマ/カウンタ0制御レジスタB(TCCR0B)が書かれると、このビットが設定(1)になります。TCCR0Bが一時保存レジスタから更新されてしまうと、このビットはハードウェアによって解除(0)されます。このビットの論理0はTCCR0Bが新しい値で更新される用意ができたことを示します。

更新中フラグが設定(1)中に4つのタイマ/カウンタ0レジスタのどれかに書き込みが実行されると、更新された値は不正にされ、予期せぬ割り込みを起こす原因になるかもしれません。

TCNT0、OCR0A、TCCR0A、TCCR0B読み込みについての機構は異なります。TCNT0を読む時は実際のタイマ/カウンタ値が読まれ、OCR0A、TCCR0AまたはTCCR0Bを読む時は一時保存レジスタの値が読まれます。

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

ビット (\$6E)	7	6	5	4	3	2	1	0	
	–	–	–	–	–	–	OCIE0A	TOIE0	TIMSK0
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 – OCIE0A : タイマ/カウンタ0比較A割り込み許可 (Timer/Counter0 Output Compare A Match Interrupt Enable)

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比較一致割り込みが許可されます。タイマ/カウンタ0で比較一致が起こる、換言するとタイマ/カウンタ0割り込み要求フラグレジスタ(TIFR0)でタイマ/カウンタ0比較一致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。

10.11.7. TIFR0 – タイマ/カウンタ0割り込み要求フラグレジスタ (Timer/Counter0 Interrupt Flag Register)

ビット \$15 (\$35)	7	6	5	4	3	2	1	0	
	–	–	–	–	–	–	OCF0A	TOV0	TIFR0
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 – 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き、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き、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。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。

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

ビット \$23 (\$43)	7	6	5	4	3	2	1	0	
	TSM	–	–	–	–	–	PSR0	PSR1	GTCCR
Read/Write	R/W	R	R	R	R	R	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

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

このビットが1の時にタイマ/カウンタ0の前置分周器はリセットします。通常、このビットはハードウェアによって直ちに解除(0)されます。タイマ/カウンタ0が非同期動作の時にこのビットが(1)を書かれると、このビットは前置分周器がリセットされてしまうまで1に留まります。TSMビットが設定(1)される場合、このビットはハードウェアによって解除(0)されません。タイマ/カウンタ同期(同時)動作の記載については78頁の「●ビット7 – TSM : タイマ/カウンタ同時動作」の記述を参照してください。

11. タイマ/カウンタ1の前置分周器

11.1. 概要

本項内の多くのビット参照は一般形で書かれています。小文字の'n'はタイマ/カウンタ番号を表します。

11.1.1. 内部クロック元

タイマ/カウンタはシステムクロック(CSn2~0=001設定)によって直接的にクロック駆動できます。これはシステムクロック周波数($f_{clk_I/O}$)と等しいタイマ/カウンタ最大クロック周波数での最高速動作を提供します。選択で前置分周器からの4つのタップの1つがクロック元として使えます。この前置分周したクロックは $f_{clk_I/O}/8$, $f_{clk_I/O}/64$, $f_{clk_I/O}/256$, $f_{clk_I/O}/1024$ の何れかの周波数です。

11.1.2. 前置分周器リセット

この前置分周器は自由走行、換言すると、タイマ/カウンタのクロック選択論理回路と無関係に動作します。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、前置分周器により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。

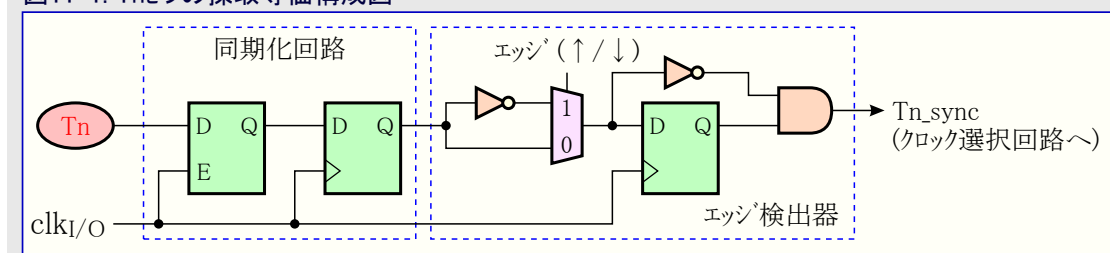
プログラム実行にタイマ/カウンタを同期することに対して前置分周器リセットを使うことが可能です。しかし、同じ前置分周器を共用する他のタイマ/カウンタも前置分周を使う場合、注意が必要とされなければなりません。前置分周器リセットはそれが接続される全タイマ/カウンタに関する前置分周器周期に影響を及ぼします。

11.1.3. 外部クロック元

T1ピンに印加された外部クロック元はタイマ/カウンタクロック(f_{clk_T1})として使えます。このT1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エッジ検出器を通して通過されます。図11-1はT1同期化とエッジ検出器論理回路の機能等価構成図を示します。レジスタは内部システムクロック($f_{clk_I/O}$)の上昇端でクロック駆動されます。ラッチは内部システムクロックのHigh区間で通過(Low区間で保持)です。

エッジ検出器は上昇端(CSn2~0=111)または下降端(CSn2~0=110)の検出毎に1つの clk_{T1} パルスを生成します。

図11-1. Tnピンの採取等価構成図



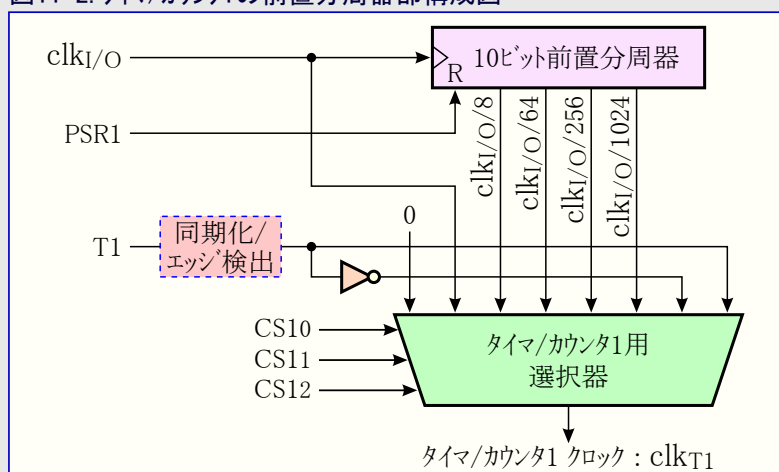
同期化とエッジ検出器論理回路はT1ピンへ印加されたエッジから計数器が更新されるまでに2.5~3.5システムクロック周期の遅延をもたらします。

クロック入力の許可と禁止はT1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ず、さもなければ不正なタイマ/カウンタクロックパルスが生成される危険があります。

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

外部クロック元は前置分周できません。

図11-2. タイマ/カウンタ1の前置分周器部構成図



注: 入力(T1)ピンの同期化/エッジ検出論理回路は図11-1.で示されます。

11.2. 同期タイマ/カウンタ前置分周器用レジスタ

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

ビット	7	6	5	4	3	2	1	0	
\$23 (\$43)	TSM	–	–	–	–	–	PSR0	PSR1	GTCCR
Read/Write	R/W	R	R	R	R	R	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

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

TSMビットに1を書くことはタイマ/カウンタ同期化動作を活性(有効)にします。この動作でPSR0とPSR1へ書かれる値は保持され、従って対応する前置分周器リセット信号の有効を保持します。これは対応するタイマ/カウンタを停止し、設定中にそれらの1つが進行する危険なしに同じ値に設定できることを保証します。TSMビットが0を書かれると、PSR0とPSR1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 – PSR1 : タイマ/カウンタ1 前置分周器リセット (Prescaler Reset Timer/Counter 1)

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

12. 16ビット 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。主な特徴は次のとおりです。

12.1. 特徴

- ・ 真の16ビット設計 (換言すれば16ビットPWMの許容)
- ・ 2つの独立した比較出力部
- ・ 比較部毎4つの制御された出力ピン
- ・ 2重緩衝の比較レジスタ
- ・ 1つの捕獲入力部
- ・ 捕獲入力雑音消去器
- ・ 比較一致でのタイマ/カウンタ解除 (自動再設定)
- ・ 不具合なしで正しい位相のパルス幅変調器 (PWM)
- ・ 可変PWM周期
- ・ 周波数発生器
- ・ 外部事象計数器
- ・ 4つの独立した割り込み (TOV1, OCF1A, OCF1B, ICF1)

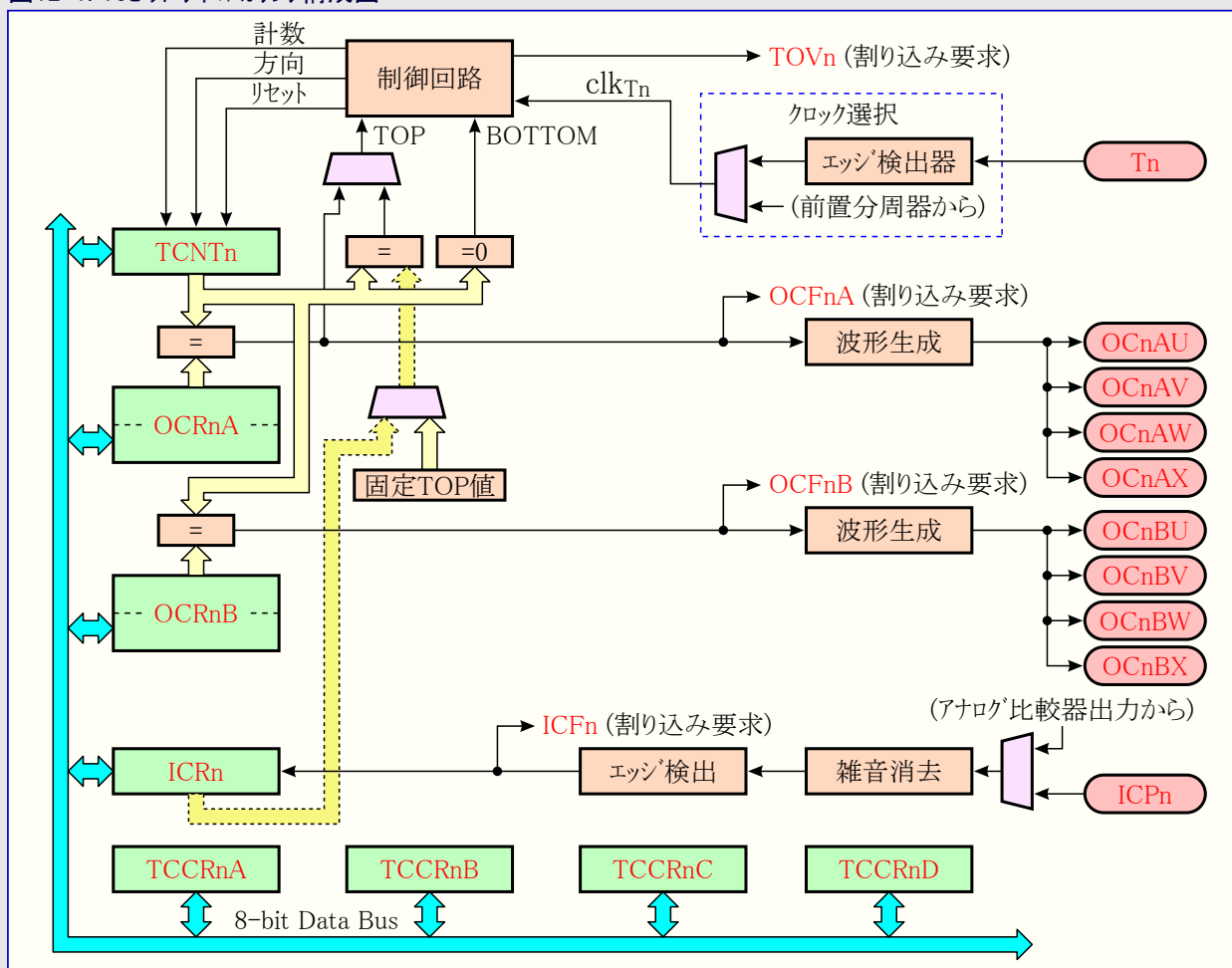
12.2. 概要

本項内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/カウンタ番号の置き換えで、この場合は1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)。
- ・ 小文字の'x'は比較出力部の置き換えで、この場合はAまたはB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の比較A値に対するアクセスのOCR1Aのように)。
- ・ 小文字の'i'は比較出力ピンの指標置き換えで、この場合はU,V,WまたはX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。

この16ビット タイマ/カウンタの簡便化した構成図は図12-1.で示されます。I/Oピンの実際の配置については7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94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。

図12-1. 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7頁の「ピン配置」、55頁の表9-3、59頁の表9-6を参照してください。

12.2.1. 関係レジスタ

タイマ/カウンタ(TCNT1)、比較レジスタ(OCR1A,OCR1B)、捕獲レジスタ(ICR1)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81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(TCCR1A,TCCR1B,TCCR1C,TCCR1D)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(TIFR1)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(TIMSK1)で個別に遮蔽(禁止)されます。TIFR1とTIMSK1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T1ピンの外部クロック元によってクロック駆動されます。クロック選択論理部はタイマ/カウンタが値を増加(または減少)するのに使うクロック元とエッジを制御します。クロック元が選択されないとき、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(clk_{T1})として参照されます。

2重緩衝した比較レジスタ(OCR1A,OCR1B)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(OC1Ai,OC1Bi)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。85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(OCF1A,OCF1B)も設定(1)します。

捕獲レジスタ(ICR1)は捕獲起動(ICP1)ピンまたはアナログ比較器出力(131頁の「アナログ比較器」参照)のどちらかの外部(エッジで起動された)事象でタイマ/カウンタ値を捕獲(複写)できます。捕獲入力部は尖頭雑音を捕らえる機会を軽減するためにデジタル濾波部(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、OCR1A、ICR1、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OCR1Aを使うと、OCR1AはPWM出力生成用に使えません。けれどもこの場合、TOP値は動作中に変更されるのをTOP値に許す2重緩衝化します。固定的なTOP値が必要とされる場合、ICR1が代わりに使え、PWM出力として使われるべきOCR1Aを開放します。

12.2.2. 定義

次の定義は本資料を通して広範囲に使われます。

表12-1. 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)、OCR1A値、ICR1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。

12.3. 16ビットレジスタのアクセス

TCNT1, OCR1A, OCR1B, ICRI1は8ビットバス経由でAVR CPUによってアクセスできる16ビットレジスタです。この16ビットレジスタは2回の読みまたは書き操作を使ってバイトアクセスされなければなりません。各々の16ビットタイマ/カウンタは16ビットアクセスの上位バイトの一時保存用に1つの8ビットレジスタを持ちます。16ビットタイマ/カウンタ内の全ての16ビットレジスタ間で、この同じ一時レジスタが共用されます。下位バイトアクセスが16ビット読み書き動作を起動します。16ビットレジスタの下位バイトがCPUによって書かれると、一時レジスタに保存した上位バイトと書かれた下位バイトは同じクロック周期で両方が16ビットレジスタに複写されます。16ビットレジスタの下位バイトがCPUによって読まれると、16ビットレジスタの上位バイトは下位バイトが読まれるのと同じクロック周期で一時レジスタに複写されます。

全ての16ビットアクセスが上位バイトに対して一時レジスタを使う訳ではありません。OCR1A, OCR1B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。

12.3.1. コード例

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。OCR1A, OCR1B, ICRI1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
STS    TCNT1H, R17
STS    TCNT1L, R16

LDS    R16, TCNT1L
LDS    R17, TCNT1H
~
;
;[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;
~
TCNT1 = 0x1FF;
i = TCNT1;
~
/* */
/* 16ビット($01FF)書き込み */
/* 16ビット読み込み */
/* */
```

注: このコード例は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。

アセンブリ言語コード例はR17:R16レジスタ対にTCNT1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TCNT1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、OCR1A,OCR1B,ICR1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_TCNT1:   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
            LDS      R16, TCNT1L       ; TCNT1下位バイト取得(上位バイト⇒一時レジスタ)
            LDS      R17, TCNT1H       ; TCNT1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_TCNT1(void)
{
    unsigned char sreg;                /* ステータスレジスタ一時保存変数定義 */
    unsigned int i;                   /* TCNT1読み出し変数定義 */
    sreg = SREG;                      /* 現全割り込み許可フラグ(I)を保存 */
    _CLI();                          /* 全割り込み禁止 */
    i = TCNT1;                        /* TCNT1値を取得 */
    SREG = sreg;                      /* 全割り込み許可フラグ(I)を復帰 */
    return i;                         /* TCNT1値で呼び出し元へ復帰 */
}
```

注: このコード例は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。

アセンブリ言語コード例はR17:R16レジスタ対にTCNT1値を戻します。

次のコード例はTCNT1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、OCR1A,OCR1B,ICR1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_TCNT1:   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
            STS      TCNT1H, R17       ; TCNT1上位バイト設定(一時レジスタ)
            STS      TCNT1L, R16       ; TCNT1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_TCNT1(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;                   /* TCNT1書き込み変数定義 */
    sreg = SREG;                      /* 現全割り込み許可フラグ(I)を保存 */
    _CLI();                          /* 全割り込み禁止 */
    TCNT1 = i;                        /* TCNT1値を設定 */
    SREG = sreg;                      /* 全割り込み許可フラグ(I)を復帰 */
}
```

注: このコード例は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。

アセンブリ言語コード例はR17:R16レジスタ対がTCNT1へ書かれるべき値を含むことが必要です。

12.3.2. 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。

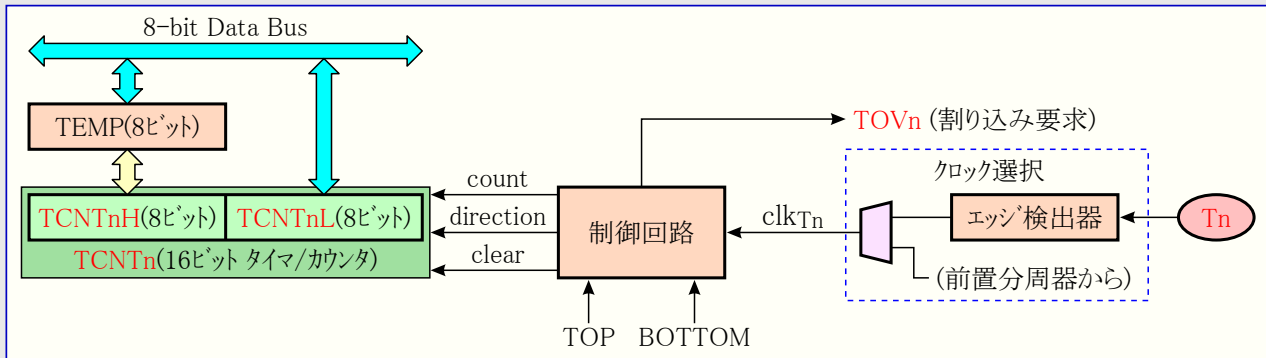
12.4. タイマ/カウンタのクロック

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(TCCR1B)**に配置された**クロック選択(CS12~0)ビット**によって制御されるクロック選択論理回路で選択されます。クロック元と前置分周器の詳細については77頁の「**タイマ/カウンタ1の前置分周器**」をご覧ください。

12.5. 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12-2はこのカウンタとその周辺の構成図を示します。

図12-2. カウンタ部構成図



信号説明 (内部信号)	count	TCNT1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	TCNT1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	TCNT1が最大値に到達したことを示す信号。
	BOTTOM	TCNT1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(TCNT1H)と下位8ビットを含むカウンタ下位(TCNT1L)の2つの8ビット I/Oメモリ位置に配置されます。TCNT1H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TCNT1H 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TCNT1Lが読まれる時にTCNT1H値で更新され、TCNT1Lが書かれる時にTCNT1H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TCNT1書き込みの特別な場合に注意することが重要です。この特別な場合はそれらが重要となる項目で記述されます。

使った動作種別に依存して、カウンタは各タイマ/カウンタ クロック(clkT1)でリセット(\$0000)、増加(+1)、または減少(-1)されます。clkT1はクロック選択(CS12~0)ビットによって選択された内部または外部のクロック元から生成できます。クロック元が選択されない(CS12~0=000)時にカウンタは停止されます。けれどもTCNT1値はタイマ/カウンタ クロック(clkT1)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)。

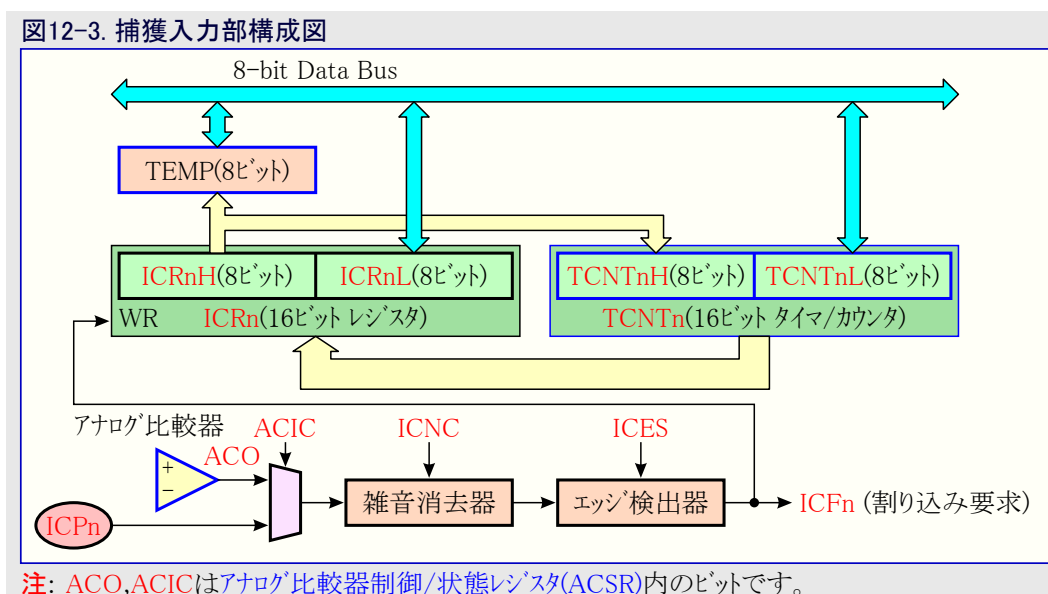
計数順序(方法)は**タイマ/カウンタ制御レジスタA(TCCR1A)**と**タイマ/カウンタ制御レジスタB(TCCR1B)**に配置された**波形生成種別(WGM13~0)ビット**の設定によって決定されます。これらはカウンタ動作(計数)方法と波形がOC1A/B比較出力に生成される方法間の接続に近いものです。進化した計数順序と波形生成についてより多くの詳細に対しては89頁の「**動作種別**」をご覧ください。

タイマ/カウンタ溢れ(TOV1)フラグはWGM13~0ビットによって選択された動作種別に従って設定(1)されます。TOV1はCPU割り込み発生に使えます。

12.6. 捕獲入力部

タイマ/カウンタは外部の出来事を捕獲でき、発生時間を示す時間印(タイマ/カウンタ値)を与える捕獲入力部と合体します。出来事または複数の出来事を示す外部信号はICP1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12-3.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。



捕獲起動入力(ICP1)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(TCNT1)の16ビット値が捕獲レジスタ(ICR1)に書かれます。捕獲割り込み要求フラグ(ICF1)はTCNT1値がICR1に複写されるのと同じシステムクロックで設定(1)されます。許可(I=1, ICIE1=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。ICF1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(ICR1)の16ビット値読み込みは、初めに下位バイト(ICR1L)、その後に上位バイト(ICR1H)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ICR1H I/O位置を読むと、この一時レジスタをアクセスします。

ICR1はカウンタのTOP値定義にICR1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ICR1に書かれ得る前に波形生成種別(WGM13~0)ビットが設定されなければなりません。ICR1に書く時は下位バイトがICR1Lに書かれる前に、上位バイトがICR1H 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81頁の「16ビットレジスタのアクセス」を参照してください。

12.6.1. 捕獲起動元

捕獲入力部用の主な起動元は捕獲起動入力(ICP1)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択されます。起動元を変更することが捕獲を起動し得ることに気付いてください。従って捕獲割り込み要求フラグ(ICF1)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(ICP1)ピンとアナログ比較器出力(ACO)の両入力、T1ピン(77頁の図11-1.参照)についてと同じ技法を使って採取されます。エッジ検出器も全く同じです。けれども雑音消去が許可されると、付加論理回路がエッジ検出器の前に挿入され、そして遅延を4システムクロック周期増やします。タイマ/カウンタがTOP値定義にICR1を使う波形生成種別に設定されないなら、雑音消去器とエッジ検出器の入力が常に許可されることに注意してください。

捕獲入力にはICP1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。

12.6.2. 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、エッジ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(TCCR1B)の捕獲入力雑音消去許可(ICNC1)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ICR1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。

12.6.3. 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(ICR1)**の値をプロセッサが読めなかった場合、ICR1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ICR1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ICR1が読まれてしまった後に可能な限り早く行われなければなりません。エッジの変更後、**捕獲割り込み要求フラグ(ICF1)**はソフトウェア(I/Oビット位置への論理1書き込み)によって解除(0)されなければなりません(訳補:エッジ変更によってICF1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)ICF1の解除(0)は必要とされません。

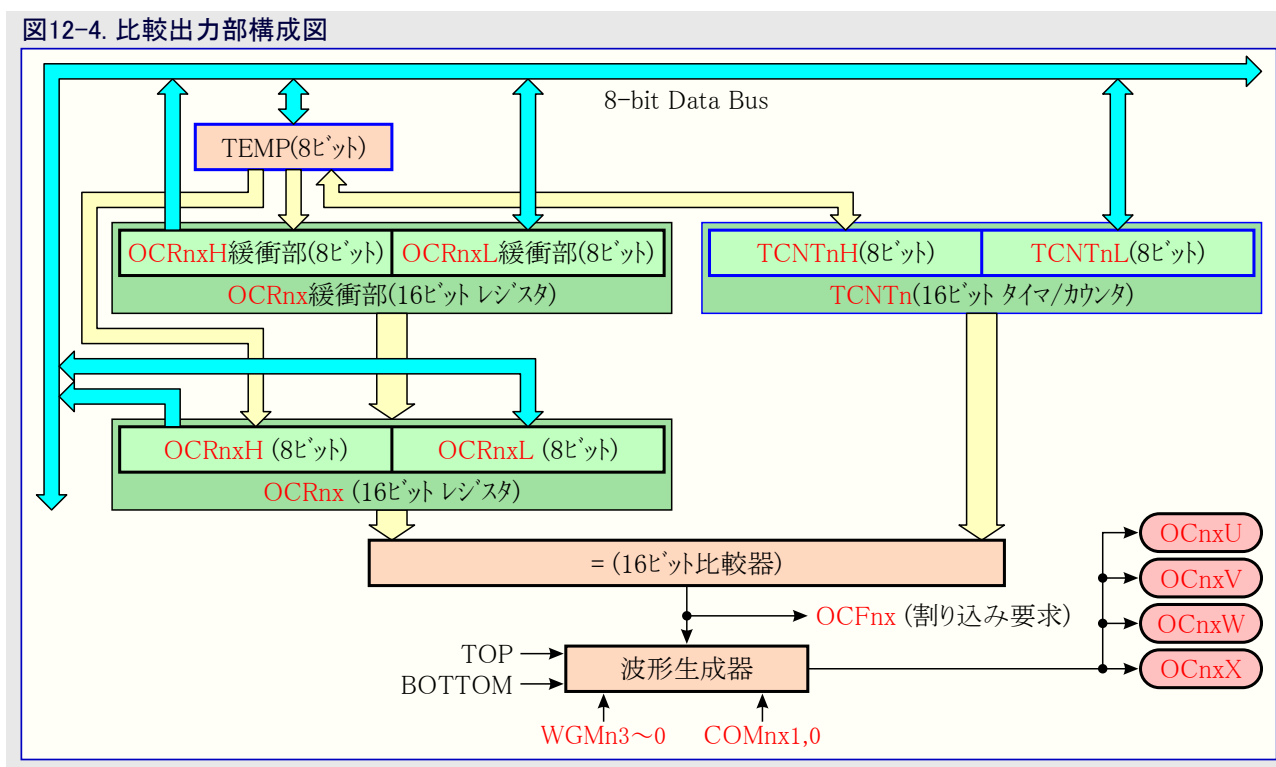
12.7. 比較出力部

この16ビット比較器は**TCNT1**と**比較レジスタ(OCR1x)**を継続的に比較します。TCNT1とOCR1x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(OCF1x)**を設定(1)します。許可(I=1, **OCIE1x=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。OCF1x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OCF1x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(WGM13~0)ビット**と**比較出力選択(COM1x1,0)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(89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12-4.は比較出力部の構成図を示します。直接的な比較出力部の部分でない構成図の要素は青枠(訳注:原文は灰色背景)で示されます。

図12-4. 比較出力部構成図



OCR1x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OCR1x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。

OCR1x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OCR1x緩衝部をアクセスし、禁止されるとOCR1xレジスタを直接アクセスします。OCR1x(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TCNT1やICR1のようにOCR1xを自動的に更新しません)。従ってOCR1x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。OCR1x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(OCR1xH)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(OCR1xL)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OCR1x緩衝部またはOCR1x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81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1xレジスタ全体をOCR1x、OCR1xを構成する緩衝部分をOCR1x緩衝部、実際の比較に使われるレジスタ本体部分をOCR1x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。

12.7.1. 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(FOC1x)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(OCF1x)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、OCnxipinは実際の比較一致が起きた場合と同様に更新されます(COM1x1,0ビット設定がOCnxipinの設定(1)、解除(0)、1/0交互のどれかを定義)。

12.7.2. TCNT1書き込みによる比較一致妨害

TCNT1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、TCNT1と同じ値に初期化されることをOCR1xに許します。

12.7.3. 比較一致部の使用

どの動作種別でのTCNT1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なにかに拘らず、何れかの比較出力部を使う場合、TCNT1を変更する時に危険を伴います。TCNT1に書かれた値がOCR1x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TCNT1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TCNT1値を書いてはいけません。

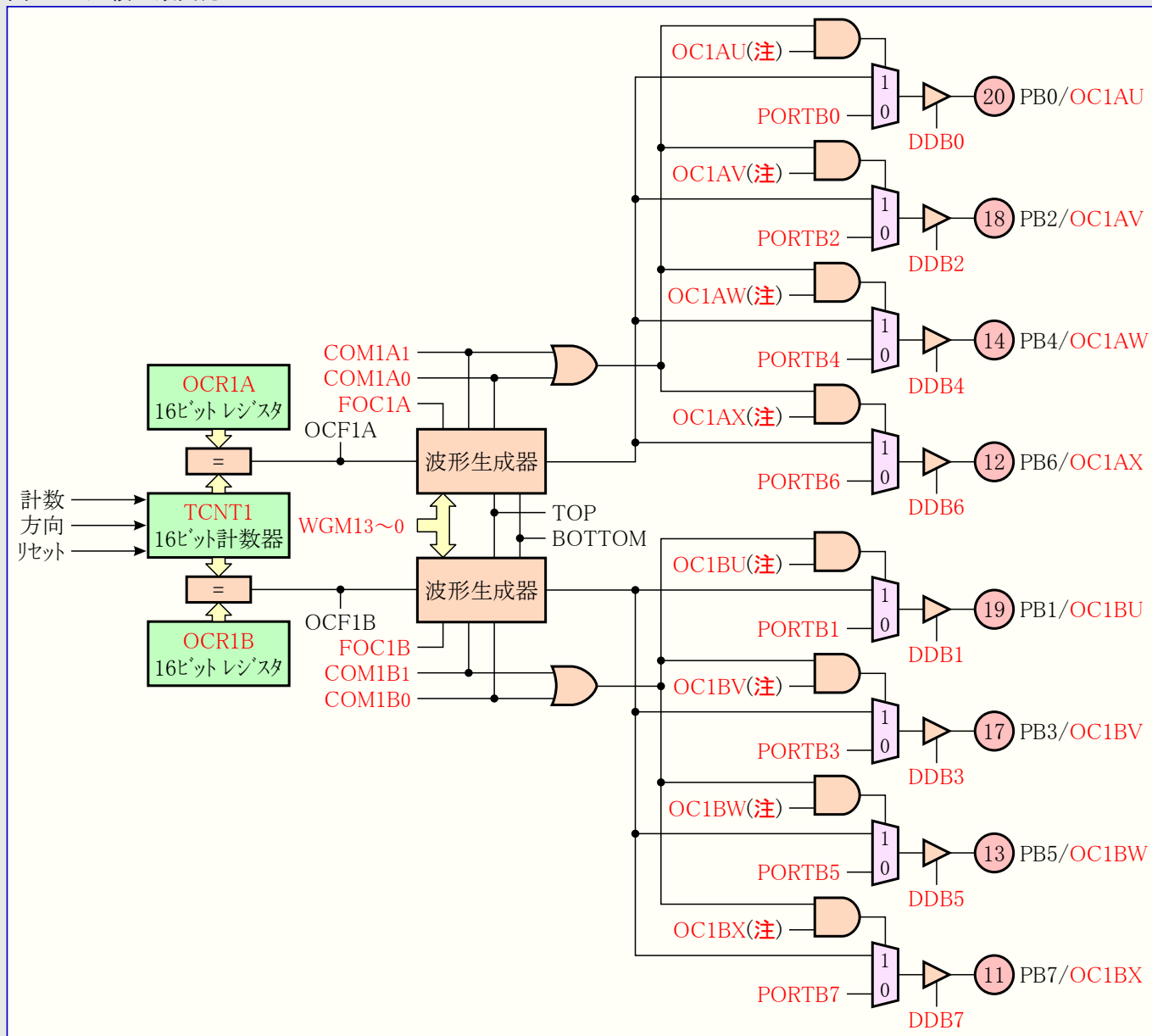
OCR1xの初期設定はポートピンに対するポート方向レジスタを出力に設定する前に行われるべきです。OCR1x値を設定する一番簡単な方法は標準動作で強制変更(FOC1x)スローブビットを使うことです。波形生成種別間を変更する時であっても、OCR1x(内部)レジスタはその値を保ちます。

比較出力選択(COM1x1,0)ビットが比較値(OCR1x)と共に2重緩衝されないことに気付いてください。COM1x1,0ビットの変更は直ちに有効となります。

12.8. 比較一致出力部

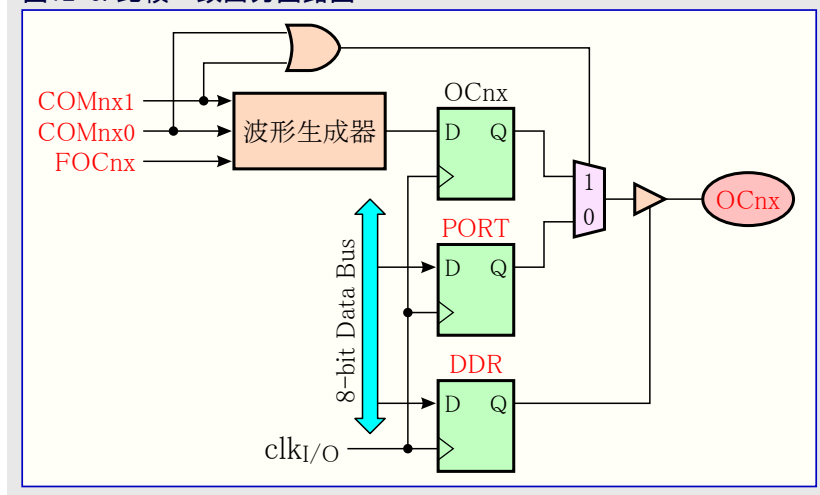
比較出力選択($COM1x1,0$)ビットは2つの機能を持ちます。波形生成器は次の比較一致での比較出力($OC1x$)状態の定義に $COM1x1,0$ ビットを使います。次に $COM1x1,0$ ビットは $OC1x$ ピン出力元を制御します。図12-6は $COM1x1,0$ ビット設定によって影響される論理回路の簡略化した図を示します。図のI/Oレジスタ、I/Oビット、I/Oピンは赤文字(訳注:原文は太字)で示されます。 $COM1x1,0$ ビットによって影響を及ぼされる標準I/Oポート制御レジスタ(PORTとDDR)の部分だけが示されます。 $OC1x$ の状態を参照するとき、その参照は $OC1xi$ ピンでなく内部 $OC1x$ レジスタに対してです。システムリセットが起こると、 $OC1x$ レジスタは0にリセットされます。

図12-5. 比較一致出力



注: $OC1xi$ - タイマ/カウンタ1制御レジスタD(TCCR1D)のレジスタビット

図12-6. 比較一致出力回路図



12.8.1. 比較出力機能

COM1x1,0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(OC1x)によって無効にされます。けれどもOC1xi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。OC1xiピンに対するポート方向レジスタのビット(DDR_OC1xi)はOC1x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12-2、表12-3、表12-4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OC1x状態の初期化を許します。いくつかのCOM1x1,0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。94頁の「16ビットタイマ/カウンタ用レジスタ」をご覧ください。

COM1x1,0ビットは捕獲入力部での何の効果もありません。

12.8.2. 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COM1x1,0ビットを違うふうに使います。全動作種別に対してCOM1x1,0=00設定は次の比較一致で実行すべきOC1x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94頁の表12-2を参照してください。高速PWM動作については94頁の表12-3、位相基準PWMと位相/周波数基準PWMについては94頁の表12-4を参照してください。

COM1x1,0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(FOC1x)スローブビットを使うことによって直ちに効果を得ることを強制できます。

12.9. 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(WGM13~0)ビット**と**比較出力選択(COM1x1,0)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。COM1x1,0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COM1x1,0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。87ページの「**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93ページの「**タイマ/カウンタのタイミング**」を参照してください。

12.9.1. 標準動作

最も単純な動作種別が標準動作(WGM13~0=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(TOV1)フラグ**はTCNT1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TOV1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TOV1フラグを自動的に解除(0)するタイマ/カウンタ溢れ割り込みと組み合わせたタイマ/カウンタの分解能はソフトウェアによって増やせます。標準動作での考慮に特別な場合はなく、新しいカウンタ値は何時でも書けます。

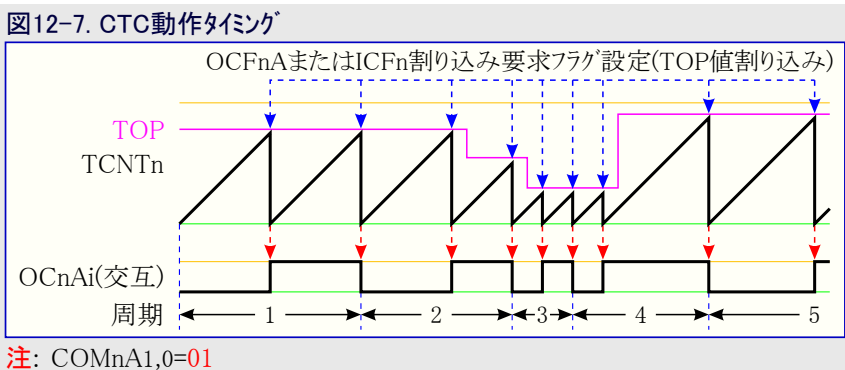
捕獲入力部は標準動作での使用が容易です。けれども外部の事象間の最大間隔がタイマ/カウンタの分解能(16ビット長)を越えてはならないことに気付いてください。事象間の間隔が長すぎる場合、捕獲部に対して分解能を拡張するために、タイマ/カウンタ溢れ割り込みまたは前置分周器が使われなければなりません。

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

12.9.2. 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(WGM13~0=0100または1100)ではOCR1AまたはICR1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(TCNT1)値がOCR1A(WGM13~0=4)またはICR1(WGM13~0=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。OCR1AまたはICR1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12-7.で示されます。カウンタ(TCNT1)値はOCR1AまたはICR1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(TCNT1)は解除(\$0000)されます。



TOP値を定義するのに使われるレジスタに対してOCF1AまたはICF1のどちらかを使うことにより、カウンタ値がTOP値に到達する時毎に割り込みが生成できます。割り込みが許可されるなら、割り込み処理ルーチンはTOP値を更新するのに使えます。けれども前置分周なしまたは低い前置分周値でカウンタが走行している時にBOTTOMと近い値にTOPを変更するのは、CTC動作が2重緩衝機能を持たないために注意して行わなければなりません。OCR1AまたはICR1に書かれた新しい値がTCNT1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。OCR1Aが2重緩衝されるので、代替はTOPを定義するのにOCR1Aを用いる**高速PWM動作(WGM13~0=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、OC1A出力は**比較出力選択(COM1A1,0)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。OC1A値はそのピンに対するデータ方向が出力(DDR_OC1Ai=1)に設定され、**タイマ/カウンタ制御レジスタD(TCCR1D)の比較A出力ピン許可(OC1Ai)**が設定(1)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OCR1Aが0(\$0000)に設定される時に $f_{OC1A} = f_{clk_I/O} / 2$ 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(TOV1)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。

12.9.3. 高速PWM動作

高速パルス幅変調(PWM)動作(WGM13~0=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(COM1x1,0=10)での比較出力(OC1x)はTCNT1とOCR1x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(COM1x1,0=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OCR1AかICR1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(OCR1AまたはICR1が\$0003設定)、最大分解能は16ビット(OCR1AまたはICR1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(WGM13~0=0101,0110,0111)、ICR1値(WGM13~0=1110)またはOCR1A値(WGM13~0=1111)のどれかと一致するまで増加されます。そしてカウンタは(一致の)次のタイマ/カウンタクロック周期で解除(\$0000)されます。高速PWM動作のタイミング図は図12-8で示されます。本図はOCR1AかICR1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。TCNT1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OCR1x値を示し、TCNT1値との交点(接点)がTCNT1とOCR1x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(OCF1x)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(TOV1)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、OCR1AかICR1のどちらかがTOP値を定義するのに使われると、OCF1AまたはICF1割り込み要求フラグはTOV1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、TCNT1とそのOCR1x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OCR1x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。

ICR1がTOP値を定義するのに使われるとき、ICR1を更新する手順はOCR1Aの更新と異なります。ICR1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ICR1が小さな値に変更される場合、書かれた新しいICR1値がTCNT1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、OCR1A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OCR1AのI/O位置に許します。OCR1A I/O位置が書かれると、書かれた値はOCR1A緩衝部に置かれます。OCR1A(比較)レジスタはその後TCNT1がTOPと一致した次のタイマ/カウンタクロック周期にOCR1A緩衝部の値で更新されます。この更新はTCNT1の解除(\$0000)やTOV1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ICR1を使うことは決まったTOP値を使う時に上手いことです。ICR1を使うことにより、OC1AでのPWM出力を生成するためにOCR1A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、OCR1Aが2重緩衝機能のため、TOPとしてOCR1A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OC1xiピンでのPWM波形の生成を許します。COM1x1,0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COM1x1,0を'11'に設定することで生成できます。94頁の表12-3をご覧ください。実際のOC1x値はそのポートピンに対するデータ方向が出力(DDR_OC1xi=1)として設定され、タイマ/カウンタ制御レジスタD(TCCR1D)の比較x出力ピン許可(OC1xi)が設定(1)される場合にだけ見えるでしょう。PWM波形はTCNT1とOCR1x間の比較一致でOC1x(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OC1xレジスタを解除(0)(または設定(1))することによって生成されます。

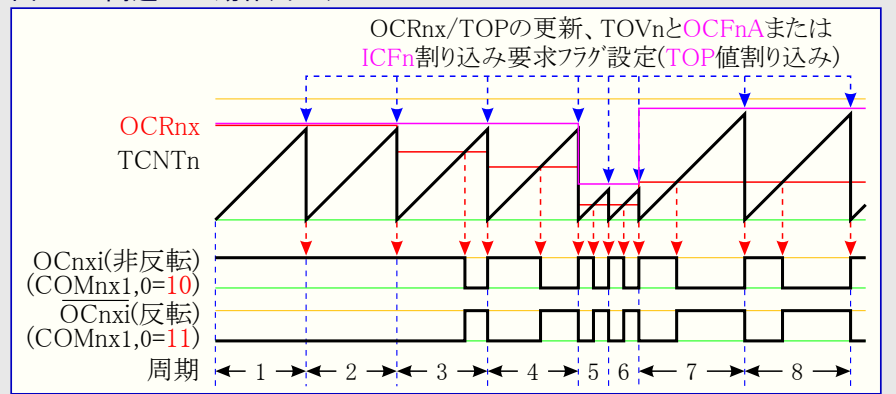
PWM出力周波数は次式によって計算できます。変数Nは前置分周数(1,8,64,256,1024)を表します。

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

OCR1xの両端値は高速PWM動作でPWM波形出力を生成する時の特別な場合にあたります。OCR1x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OCR1x設定は(COM1x1,0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、OC1A設定(COM1A1,0=01)によって達成できます。これはTOP値を定義するのにOCR1Aが使われる(WGM13~0=1111)の場合にだけ適用されます。生成された波形はOCR1Aが0(\$0000)に設定される時に $f_{\text{OC1A}} = f_{\text{clk_I/O}} / 2$ の最大周波数でしょう。この特性は高速PWM動作で比較出力部の2重緩衝機能が許可されることを除いて、CTC動作でのOC1A交互出力(COM1A1,0=01)と同じです。

図12-8. 高速PWM動作タイミング



12.9.4. 位相基準PWM動作

位相基準パルス幅変調(PWM)動作(WGM13~0=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(COM1x1,0=10)での比較出力(OC1x)は上昇計数中のTCNT1とOCR1x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(COM1x1,0=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。

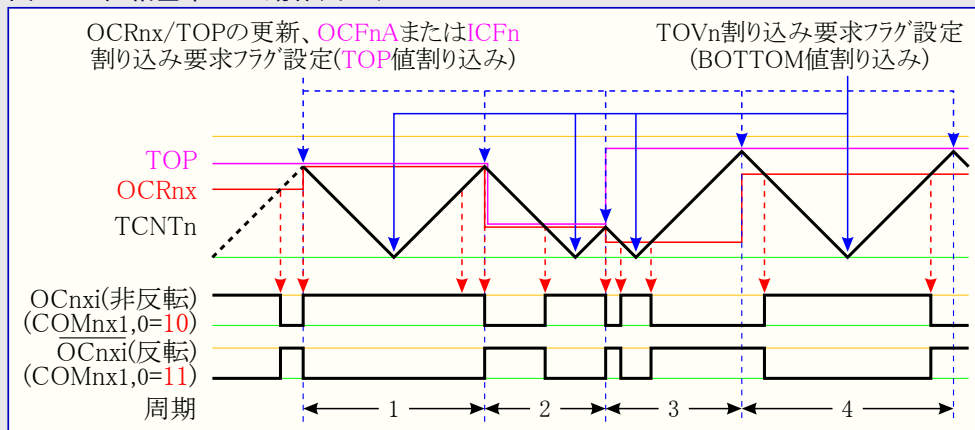
位相基準PWM動作のPWM分解能は8,9,10ビットに固定、若しくはOCR1AかICR1のどちらかによって定義できます。許された最小分解能は2ビット(OCR1AまたはICR1が\$0003設定)、最大分解能は16ビット(OCR1AまたはICR1がMAX設定)です。ビットでのPWM分解能は次式を使うことによって計算できます。

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

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(WGM13~0=0001,0010,0011)、ICR1値(WGM13~0=1010)またはOCR1A値(WGM13~0=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TCNT1値は1タイマ/カウンタクロック

周期、TOPと等しくなります。位相基準PWM動作のタイミング図は図12-9.で示されます。この図はOCR1AかICR1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。TCNT1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OCR1x値を示し、TCNT1値との交点(接点)がTCNT1とOCR1x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(OCF1x)は比較一致が起こると設定(1)されます。

図12-9. 位相基準PWM動作タイミング



タイマ/カウンタ溢れ(TOV1)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。OCR1AかICR1のどちらかがTOP値を定義するのに使われるとき、OCF1AまたはICF1割り込み要求フラグはOCR1xレジスタが(TOPに於いて)2重緩衝部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、TCNT1とそのOCR1x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OCR1x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12-9.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OCR1xレジスタの更新時に見出せます。OCR1x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値を使うとき、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OC1xiピンでのPWM波形の生成を許します。COM1x1,0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COM1x1,0を'11'に設定することで生成できます(94頁の表12-4.をご覧ください)。実際のOC1x値はそのポートピンに対するデータ方向が出力(DDR_OC1xi=1)として設定され、タイマ/カウンタ制御レジスタD(TCCR1D)の比較x出力ピン許可(OC1xi)が設定(1)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TCNT1とOCR1x間の比較一致でOC1x(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TCNT1とOCR1x間の比較一致でOC1xレジスタを解除(0)(または設定(1))することによって生成されます。

位相基準PWMを使う時の出力に対するPWM周波数は次式によって計算できます。変数Nは前置分周数(1,8,64,256,1024)を表します。

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

OCR1x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OCR1x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。

12.9.5. 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(WGM13~0=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(COM1x1,0=10)での比較出力(OC1x)は上昇計数中のTCNT1とOCR1x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(COM1x1,0=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。

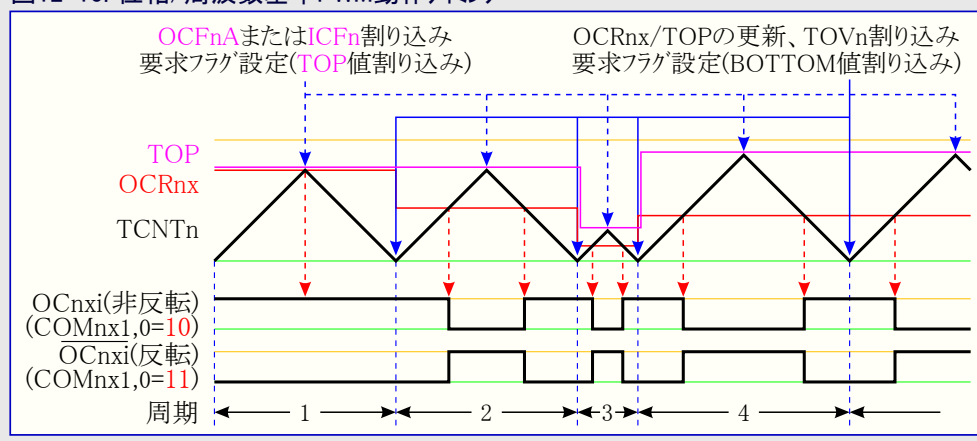
位相基準と位相/周波数基準PWM動作間の主な違いはOCR1xレジスタがOCR1x緩衝部によって更新される時(訳補:TOPとBOTTOM)です(図12-9と図12-10参照)。

位相/周波数基準PWM動作のPWM分解能はOCR1AかICR1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(OCR1AまたはICR1が\$0003設定)、最大分解能は16ビット(OCR1AまたはICR1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ICR1値(WGM13~0=1000)かOCR1A値(WGM13~0=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TCNT1値は1タイマ/カウンタクロック周期、TOPと等しくなります。位相/周波数基準PWM動作のタイミング図は図12-10で示されます。この図はOCR1AかICR1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OCR1x値を示し、TCNT1値との交点(接点)がTCNT1とOCR1x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(OCF1x)は比較一致が起こると設定(1)されます。

図12-10. 位相/周波数基準PWM動作タイミング



タイマ/カウンタ溢れ(TOV1)フラグはOCR1xレジスタが(BOTTOMに於いて)2重緩衝部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。OCR1AかICR1のどちらかがTOP値を定義するのに使われるとき、OCF1AまたはICF1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、TCNT1とそのOCR1x間で比較一致は決して起きません。

図12-10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。OCR1x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ICR1を使うことは決まったTOP値を使う時に上手くいきます。ICR1を使うことにより、OC1AでのPWM出力を生成するためにOCR1A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、OCR1Aが2重緩衝機能のため、TOPとしてOCR1A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OC1xiピンでのPWM波形の生成を許します。COM1x1,0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COM1x1,0を'11'に設定することで生成できます(94頁の表12-4をご覧ください)。実際のOC1x値はそのポートピンに対するデータ方向が出力(DDR_OC1xi=1)として設定され、タイマ/カウンタ制御レジスタD(TCCR1D)の比較x出力ピン許可(OC1xi)が設定(1)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TCNT1とOCR1x間の比較一致でOC1x(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TCNT1とOCR1x間の比較一致でOC1x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。

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

OCR1x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あたります。非反転PWM動作ではOCR1x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。

12.10. 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_{T1})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 OCR1x レジスタが OCR1x 緩衝部値で更新される時(2重緩衝を使う動作種別のみ)の情報を含みます。図12-11は OCF1x の設定についてのタイミング図を示します。

図12-11. 前置分周なし(1/1)のタイマ/カウンタ、 OCF1x 設定 タイミング

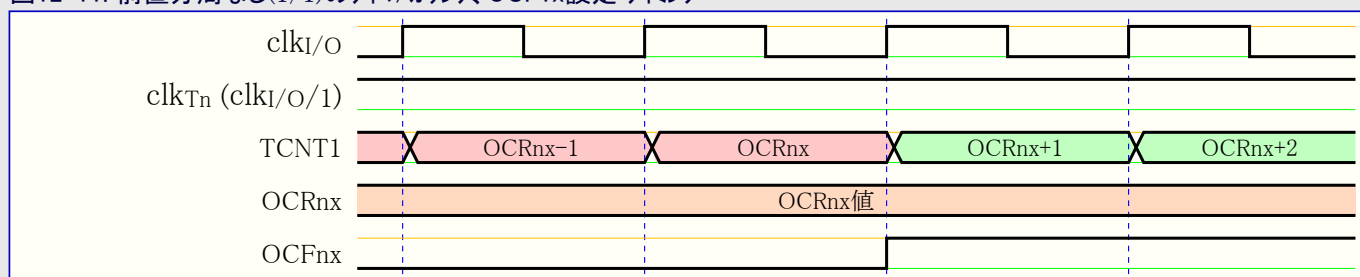


図12-12は同じタイミング データを示しますが、前置分周器が許可されています。

図12-12. 前置分周器($f_{\text{clk}_{I/O}}/8$)のタイマ/カウンタ、 OCF1x 設定 タイミング

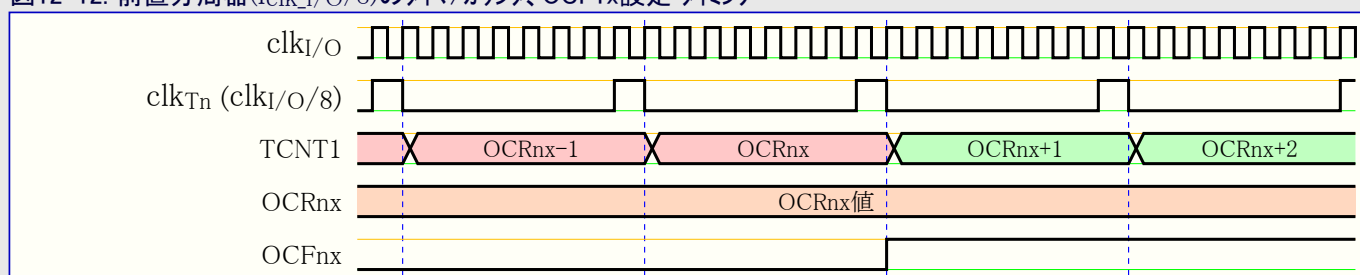


図12-13は各動作種別でのTOP近辺の計数手順を示します。位相/周波数基準PWM動作使用時の OCR1x レジスタはBOTTOMで更新されます。タイミング図は同じになりますが、当然TOPはBOTTOMで、TOP-1はBOTTOM+1でなどのように置き換えられます。BOTTOMで TOV1 を設定(1)する動作種別についても、同様な名称変更が適用されます。

図12-13. 前置分周なし(1/1)のタイマ/カウンタ、TOP近辺 タイミング

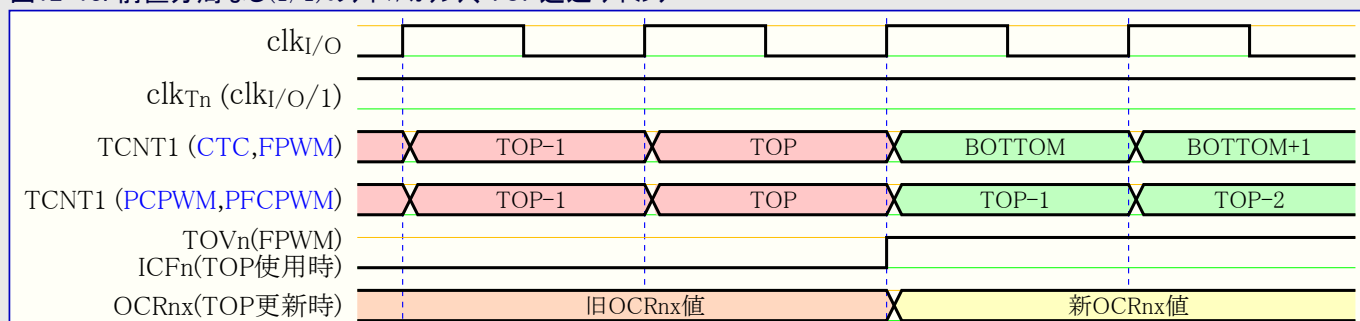
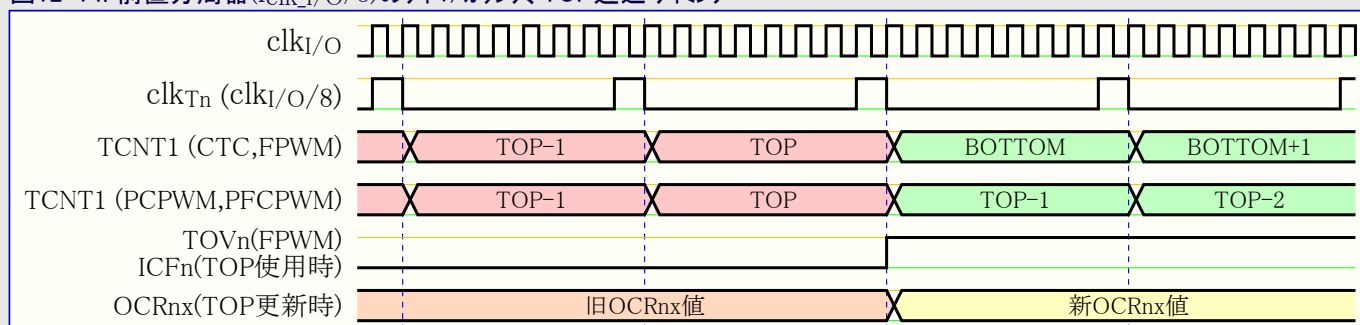


図12-14は同じタイミング データを示しますが、前置分周器が許可されています。

図12-14. 前置分周器($f_{\text{clk}_{I/O}}/8$)のタイマ/カウンタ、TOP近辺 タイミング



12.11. 16ビット タイマ/カウンタ1用レジスタ

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

ビット (\$80)	7	6	5	4	3	2	1	0	
	COM1A1	COM1A0	COM1B1	COM1B0	–	–	WGM11	WGM10	TCCR1A
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 – COM1A1,0 : 比較1A出力選択 (Compare Output Mode A bit 1 and 0)

●ビット5,4 – COM1B1,0 : 比較1B出力選択 (Compare Output Mode B bit 1 and 0)

COM1A1,0とCOM1B1,0は(各々OC1AiとOC1Biの)比較出力ピンの動作を制御します。COM1A1,0ビットの1つまたは両方が1を書かれると、OC1Ai出力はそのI/Oピンの標準ポート機能を無効にし、そのI/Oピンに接続されます。COM1B1,0ビットの1つまたは両方が1を書かれると、OC1Bi出力はそのI/Oピンの標準ポート機能を無効にし、そのI/Oピンに接続されます。けれども出力駆動部を許可するため、OC1AiまたはOC1Biの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。

OC1AiまたはOC1Biがピンに接続されるとき、COM1x1,0ビットの機能はWGM13~0ビット設定に依存します。表12-2はWGM13~0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COM1x1,0ビット機能を示します。

表12-2. 非PWM動作での比較出力選択 (注: xはAまたはB)

COM1x1	COM1x0	意味
0	0	標準ポート動作 (OC1x切断)
0	1	比較一致でOC1xiピントグル(交互)出力
1	0	比較一致でOC1xiピン Lowレベル出力
1	1	比較一致でOC1xiピン Highレベル出力

表12-3はWGM13~0ビットが高速PWM動作に設定される時のCOM1x1,0ビット機能を示します。

表12-3. 高速PWM動作での比較出力選択 (注: xはAまたはB, Xは0または1)

COM1x1	COM1x0	意味
0	0	標準ポート動作 (OC1x切断)
0	1	WGM13~0=111X : 比較一致でOC1Aiピントグル(交互)出力、OC1B切断(標準ポート動作) WGM13~0上記以外 : 標準ポート動作 (OC1x切断)
1	0	比較一致でLow、BOTTOMでHighをOC1xiピンへ出力 (非反転動作)
1	1	比較一致でHigh、BOTTOMでLowをOC1xiピンへ出力 (反転動作)

注: COM1x1が設定(1)され、OCR1x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90頁の「高速PWM動作」をご覧ください。

表12-4はWGM13~0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COM1x1,0ビット機能を示します。

表12-4. 位相基準または位相/周波数基準PWM動作での比較出力選択 (注: xはAまたはB, Xは0または1)

COM1x1	COM1x0	意味
0	0	標準ポート動作 (OC1x切断)
0	1	WGM13~0=10X1 : 比較一致でOC1Aiピントグル(交互)出力、OC1B切断(標準ポート動作) WGM13~0上記以外 : 標準ポート動作 (OC1x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OC1xi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OC1xiピンへ出力

注: COM1x1が設定(1)され、OCR1x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91頁の「位相基準PWM動作」をご覧ください。

●ビット3,2 – 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。

●ビット1,0 – WGM11,0 : 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(TCCR1B)で得られるWGM13,2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12-5参照)。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。89頁の「動作種別」をご覧ください。

表12-5. 波形生成種別選択

番号	WGM13	WGM12 (CTC1)	WGM11 (PWM11)	WGM10 (PWM10)	タイマ/カウンタ動作種別	TOP値	OCR1x 更新時	TOV1 設定時
0	0	0	0	0	標準動作	\$FFFF	即時	MAX
1	0	0	0	1	8ビット位相基準PWM動作	\$00FF	TOP	BOTTOM
2	0	0	1	0	9ビット位相基準PWM動作	\$01FF	TOP	BOTTOM
3	0	0	1	1	10ビット位相基準PWM動作	\$03FF	TOP	BOTTOM
4	0	1	0	0	比較一致タイマ/カウンタ解除(CTC)動作	OCR1A	即時	MAX
5	0	1	0	1	8ビット高速PWM動作	\$00FF	BOTTOM	TOP
6	0	1	1	0	9ビット高速PWM動作	\$01FF	BOTTOM	TOP
7	0	1	1	1	10ビット高速PWM動作	\$03FF	BOTTOM	TOP
8	1	0	0	0	位相/周波数基準PWM動作	ICR1	BOTTOM	BOTTOM
9	1	0	0	1	位相/周波数基準PWM動作	OCR1A	BOTTOM	BOTTOM
10	1	0	1	0	位相基準PWM動作	ICR1	TOP	BOTTOM
11	1	0	1	1	位相基準PWM動作	OCR1A	TOP	BOTTOM
12	1	1	0	0	比較一致タイマ/カウンタ解除(CTC)動作	ICR1	即時	MAX
13	1	1	0	1	(予約)	-	-	-
14	1	1	1	0	高速PWM動作	ICR1	BOTTOM	TOP
15	1	1	1	1	高速PWM動作	OCR1A	BOTTOM	TOP

注: CTC1とPWM11,0ビット定義名は旧名です。WGM12~0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。

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

ビット (\$81)	7	6	5	4	3	2	1	0	
	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10	TCCR1B
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

●ビット7 – ICNC1 : 捕獲起動入力1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(ICP1)ピンからの入力が増幅されます。この増幅機能はそれが出力を更新するのに、連続4回等しく評価されたICP1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 – ICES1 : 捕獲起動入力1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(ICP1)ピンのどちらかのエッジを選択します。ICES1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、ICES1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ICES1設定に従って起動されると、カウンタ値が捕獲レジスタ(ICR1)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(ICF1)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。

ICR1がTOP値として使われると(TCCR1AとTCCR1Bに配置されたWGM13~0ビットの記述をご覧ください)、ICP1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、TCCR1Bが書かれるとき、このビットは0を書かれなければならない。

●ビット4,3 – WGM13,2 : 波形生成種別 (Waveform Generation Mode bit 3 and 2)

TCCRNのWGM11,0ビット記述をご覧ください。

● ビット2~0 – CS12~0 : クロック選択1 (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(TCNT1)によって使われるべきクロック元を選択します。図12-11.と図12-12.をご覧ください。

タイマ/カウンタ1に対して外部ピン(クロック)動作が使われる場合、例えばT1ピンが出力として設定されても、T1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12-6. タイマ/カウンタ1入力クロック選択

CS12	CS11	CS10	意味
0	0	0	停止 (タイマ/カウンタ1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	T1ピンの下降端 (外部クロック)
1	1	1	T1ピンの上昇端 (外部クロック)

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

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

● ビット7 – FOC1A : OC1A強制変更 (Force Output Compare A)

● ビット6 – FOC1B : OC1B強制変更 (Force Output Compare B)

FOC1A/FOC1BビットはWGM13~0ビットが非PWM動作を指示する時だけ有効です。FOC1A/FOC1B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。OC1x出力はCOM1x1,0ビット設定に従って変更されます。FOC1A/ FOC1B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COM1x1,0ビットに存在する値です。

FOC1A/FOC1Bストロープは何れの割り込みの生成もTOPとしてOCR1A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。

FOC1A/FOC1Bビットは常に0として読まれます。

● ビット5~0 – Res : 予約 (Reserved) (訳注: 共通性から本項追加)

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、TCCR1C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。

12.11.4. TCCR1D – タイマ/カウンタ1制御レジスタD (Timer/Counter1 Control Register D)

ビット (\$83)	7	6	5	4	3	2	1	0	
	OC1BX	OC1BW	OC1BV	OC1BU	OC1AX	OC1AW	OC1AV	OC1AU	TCCR1D
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7~4 – OC1Bi : 比較B出力ピン許可 (Output Compare Pin Enable for Channel B)

OC1Biビットは87頁の図12-5.で示されるように比較B出力ピンを許可します。

● ビット3~0 – OC1Ai : 比較A出力ピン許可 (Output Compare Pin Enable for Channel A)

OC1Aiビットは87頁の図12-5.で示されるように比較A出力ピンを許可します。

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

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

この2つのタイマ/カウンタ I/O位置(TCNT1HとTCNT1Lを合わせたTCNT1)は、読み書き両方に対してタイマ/カウンタ部の16ビット カウンタに直接アクセスします。CPUがこれらのレジスタをアクセスする時に上位と下位の両バイトが同時に読み書きされるのを保証するため、このアクセスは8ビット上位バイト一時レジスタ(TEMP)を使って実行されます。この一時レジスタは他の全ての16ビット レジスタによって共用されます。[81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(TCNT1)を変更することはOCR1xの1つとTCNT1間の比較一致消失の危険を誘発します。

TCNT1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。

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

ビット (\$89)	15	14	13	12	11	10	9	8	
	(MSB)								OCR1AH
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	
ビット (\$88)	7	6	5	4	3	2	1	0	
								(LSB)	OCR1AL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

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

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

この比較レジスタは継続的にカウンタ(TCNT1)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OC1xi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。[81頁の「16ビット レジスタのアクセス」](#)をご覧ください。

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

ビット (\$87)	15	14	13	12	11	10	9	8	
	(MSB)								ICR1H
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	
ビット (\$86)	7	6	5	4	3	2	1	0	
								(LSB)	ICR1L
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

この捕獲レジスタはICP1ピン(または任意の[アナログ比較器出力](#))で出来事が起こる毎にカウンタ(TCNT1)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。[81頁の「16ビット レジスタのアクセス」](#)をご覧ください。

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

ビット (\$6F)	7	6	5	4	3	2	1	0	
	–	–	ICIE1	–	–	OCIE1B	OCIE1A	TOIE1	TIMSK1
Read/Write	R	R	R/W	R	R	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/Counter1 Input Capture Interrupt Enable)

このビットが1を書かれ、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)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(45頁の「割り込みベクタ」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/Counter1 Output Compare B Match Interrupt Enable)

このビットが1を書かれ、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)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1B割り込み要求フラグ(OCF1B)が設定(1)されると、対応する割り込みベクタ(45頁の「割り込みベクタ」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/Counter1 Output Compare A Match Interrupt Enable)

このビットが1を書かれ、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)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1A割り込み要求フラグ(OCF1A)が設定(1)されると、対応する割り込みベクタ(45頁の「割り込みベクタ」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/Counter1 Overflow Interrupt Enable)

このビットが1を書かれ、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)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(45頁の「割り込みベクタ」参照)が実行されます。

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

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

● ビット5 – ICF1 : タイマ/カウンタ1捕獲割り込み要求フラグ (Timer/Counter1, Input Capture Flag)

ICP1ピンに捕獲の事象が起こると、このフラグが設定(1)されます。捕獲レジスタ(ICR1)がWGM13~0によってTOP値として設定されると、ICF1フラグはカウンタがTOP値に到達する時に設定(1)されます。

捕獲割り込みベクタが実行されると、ICF1は自動的に解除(0)されます。代わりにこのビット位置へ論理1を書くことによってもICF1は解除(0)できます。

● ビット2 – OCF1B : タイマ/カウンタ1比較B割り込み要求フラグ (Timer/Counter1, Output Compare B Match Flag)

このフラグはカウンタ(TCNT1)値が比較Bレジスタ(OCR1B)と一致した後(次)のタイマ/カウンタクロック周期で設定(1)されます。

強制的な比較出力(FOC1B)ストロープがOCF1Bフラグを設定(1)しないことに注意してください。

比較B一致割り込みベクタが実行されると、OCF1Bは自動的に解除(0)されます。代わりにこのビット位置へ論理1を書くことによってもOCF1Bは解除(0)できます。

● ビット1 – OCF1A : タイマ/カウンタ1比較A割り込み要求フラグ (Timer/Counter1, Output Compare A Match Flag)

このフラグはカウンタ(TCNT1)値が比較Aレジスタ(OCR1A)と一致した後(次)のタイマ/カウンタクロック周期で設定(1)されます。

強制的な比較出力(FOC1A)ストロープがOCF1Aフラグを設定(1)しないことに注意してください。

比較A一致割り込みベクタが実行されると、OCF1Aは自動的に解除(0)されます。代わりにこのビット位置へ論理1を書くことによってもOCF1Aは解除(0)できます。

● ビット0 – TOV1 : タイマ/カウンタ1溢れ割り込み要求フラグ (Timer/Counter1 Overflow Flag)

このフラグの設定(1)はWGM13~0ビット設定に依存します。標準またはCTC動作でのTOV1フラグはタイマ/カウンタ1溢れ時に設定(1)されます。他のWGM13~0ビット設定を使う時のTOV1フラグ動作については95頁の表12-5を参照してください。

タイマ/カウンタ1溢れ割り込みベクタが実行されると、TOV1は自動的に解除(0)されます。代わりにこのビット位置へ論理1を書くことによってもTOV1は解除(0)できます。

(訳注) 原書のTIMSK1とTIFR1に対する未定義ビット記述を省略しました。

13. 直列周辺インターフェース (SPI: Serial Peripheral Interface)

直列周辺インターフェースはATtiny87/167と様々なAVRデバイスや周辺デバイス間の高速同期データ転送を許します。ATtiny87/167のSPIは次の特徴を含みます。

13.1. 特徴

- ・ 全二重3線同期データ転送
- ・ 主装置/従装置動作
- ・ LSB/MSB先行データ転送
- ・ 設定変更可能な7つのビット速度
- ・ 送信完了割り込み要求フラグ
- ・ 送信上書きフラグ保護
- ・ アイドル動作からの起動
- ・ 倍速(CK/2)主装置SPI動作

SPIでの主装置と従装置のCPU間相互連結は図13-2で示されます。このシステムは2つの移動レジスタと主装置クロック発生器から成ります。SPI主装置は希望した従装置のSS(従装置選択)ピンをLowへ引き込む時に一群の通信を始めます。主装置と従装置は各々の移動レジスタで送出すべきデータを用意し、主装置はデータを交換するのに必要なクロックパルスをSCK信号線に生成します。データは常にMOSI(Master Out Slave In)信号線を主装置から従装置へ、MISO(Master In Slave Out)信号線を従装置から主装置へ移動されます。各データパケット後、主装置はSS(従装置選択)ピンをHighへ引き上げることによって従装置と同期を取ります。

主装置として設定されると、SPIインターフェースにはSS信号線の自動制御がありません。これは通信を開始できるのに先立って使用者ソフトウェアによって操作されなければなりません。これが行われると、SPIデータレジスタ(SPDR)へのバイト書き込みがSPIクロック発生器を開始し、ハードウェアが従装置内へ8ビットを移動します。1バイトの移動後、SPIクロック発生器は停止し、SPI状態レジスタ(SPSR)の転送完了フラグ(SPIF)を設定(1)します。SPI制御レジスタ(SPCR)でSPI割り込み許可(SPIE)ビットが設定(1)されていれば割り込みが要求されます。主装置はSPDR内へ次バイトを書くことによって次バイトの移動を継続、またはSS(従装置選択)信号線をHighへ引き上げることによってパケットの終了を指示することができます。最後の到着バイトはその後の使用のため、緩衝レジスタ内に保持されます。

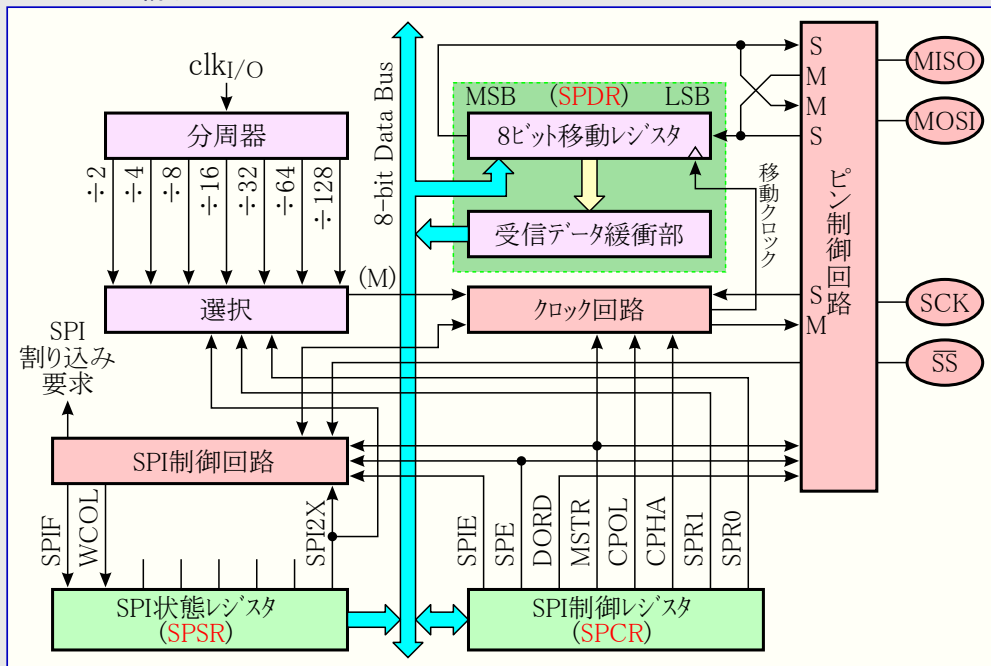
従装置として設定されると、SPIインターフェースはSSピンがHighに駆動される限り、MISOをHi-Zにした休止状態に留まります。この状態でプログラムはSPIデータレジスタ(SPDR)の内容を更新できますが、そのデータはSSピンがLowに駆動されるまでSCKピンでの到着クロックパルスによって移動出力されません。1バイトが完全に移動されてしまうと転送完了フラグ(SPIF)が設定(1)されます。SPCRでSPI割り込み許可(SPIE)ビットが設定(1)されていれば割り込みが要求されます。従装置は受信データを読む前にSPDR内へ送られるべき次のデータの配置を続けられます。最後の到着バイトはその後の使用のため、緩衝レジスタ内に保持されます。

このシステムは送信側で単一緩衝、受信側で2重緩衝です。これは一連の移動全体が完了される前に送信されるべきバイトがSPIデータレジスタ(SPDR)へ書けないことを意味します。しかし、データを受信するとき、次のデータが完全に移動入力される前に受信したデータがSPIデータレジスタ(SPDR)から読まなければなりません。さもなければ始めのバイトは失われます。

SPI従装置動作では制御論理回路がSCKピンの到着信号を採取します。このクロック信号の正しい採取を保証するため、High/Lowの最小時間は各々2CPUクロック周期よりも長くあるべきです。

SPIが許可されると、MOSI, MISO, SCK, SSピンのデータ方向は表13-1に従って無視されます。自動的なポート無視のより多くの詳細については54頁の「交換ポート機能」を参照してください。

図13-1. SPI構成図



注: SPIピン配置については7頁の「ピン配置」と55頁の表9-3を参照してください。

図13-2. SPI 主装置/従装置の連結

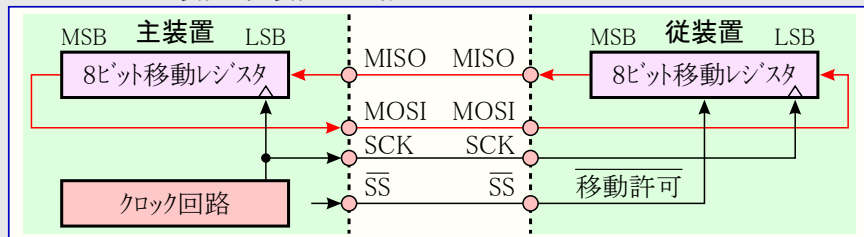


表13-1. SPIピン方向規定

ピン名	主装置時の方向規定	従装置時の方向規定
SCK	ポートB方向レジスタ(DDRB)の指定	入力
MISO	入力	ポートB方向レジスタ(DDRB)の指定
MOSI	ポートB方向レジスタ(DDRB)の指定	入力
SS	ポートB方向レジスタ(DDRB)の指定	入力

注: 使用者定義SPIピンの方向定義方法の詳細記述については59頁の「ポートBの交換機能」を参照してください。

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

例でのDDR_SPIはSPIピンを制御する実際のポート方向レジスタに置き換えられなければなりません。DD_MOSI, DD_MISO, DD_SCKはこれらのピンに対する実際のポート方向ビットに置き換えられなければなりません。例えばMOSIがPB5ピンに配置されるなら、DD_MOSIはDDB5、DDR_SPIはDDRBに置き換えます。

アセンブリ言語プログラム例

```
SPI_M_Init:  LDI    R17, (1<<DD_MOSI) | (1<<DD_SCK)      ;MOSI, SCK=出力、他は入力値を取得
              OUT    DDR_SPI, R17                        ;MOSI, SCK=出力、他は入力に設定
              LDI    R17, (1<<SPE) | (1<<MSTR) | (1<<SPR0)  ;SPI許可、主装置、16分周値を取得
              OUT    SPCR, R17                          ;SPI許可、主装置、16分周に設定
              RET                                         ;呼び出し元へ復帰

SPI_M_Tx:    OUT    SPDR, R16                            ;データ(R16)送信開始
SPI_M_Tx_W:  SBIS   SPSR, SPIF                          ;転送完了ならばスキップ
              RJMP   SPI_M_Tx_W                        ;転送完了まで待機
;
              RET                                         ;呼び出し元へ復帰
```

C言語プログラム例

```
void SPI_MasterInit(void)
{
    DDR_SPI = (1<<DD_MOSI) | (1<<DD_SCK);                /* MOSI, SCK=出力、他は入力に設定 */
    SPCR = (1<<SPE) | (1<<MSTR) | (1<<SPR0);              /* SPI許可、主装置、16分周に設定 */
}

void SPI_MasterTransmit(char cData)
{
    SPDR = cData;                                         /* データ送信開始 */
    while(!(SPSR & (1<<SPIF)));                          /* 転送完了まで待機 */
}
```

注: このコード例は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。

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: このコード例は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。

13.2. SSピンの機能

13.2.1. 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。

13.2.2. 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択して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。

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

ビット	7	6	5	4	3	2	1	0	
\$2C (\$4C)	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

● ビット7 – SPIE : SPI割り込み許可 (SPI Interrupt Enable)

ステータスレジスタ(SREG)の全割り込み許可(I)ビットが設定(1)されて、SPI状態レジスタ(SPSR)でSPI割り込み要求フラグ(SPIF)が設定(1)されると、このビットがSPI割り込みを実行させます。

● ビット6 – SPE : SPI許可 (SPI Enable)

SPEビットが1を書かれるとSPIが許可されます。どのSPI操作を許可するにも、このビットが設定(1)されなければなりません。

● ビット5 – DORD : データ順選択 (Data Order)

DORDビットが1を書かれるとデータ語のLSBが最初に転送されます。DORDビットが0を書かれるとMSBが最初に転送されます。

● ビット4 – MSTR : 主装置/従装置選択 (Master/Slave Select)

このビットは1を書かれると主装置動作、論理0を書かれると従装置動作を選択します。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13-3.と図13-4.を参照してください。CPOL機能は右で要約されます。

表13-2. 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13-3.と図13-4.を参照してください。CPHA機能は右で要約されます。

表13-3. 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13-4. 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	$f_{OSC}/2$		$f_{OSC}/4$		$f_{OSC}/8$		$f_{OSC}/16$	

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

ビット	7	6	5	4	3	2	1	0	
\$2D (\$4D)	SPIF	WCOL	–	–	–	–	–	SPI2X	SPSR
Read/Write	R	R	R	R	R	R	R	R/W	
初期値	0	0	0	0	0	0	0	0	

●ビット7 – SPIF : SPI割り込み要求フラグ (SPI Interrupt Flag)

直列転送が完了すると、このSPIFフラグが設定(1)されます。全割り込みが許可(ステータスレジスタ(SREG)の全割り込み許可(I)ビット=1)されて、SPI制御レジスタ(SPCR)でSPI割り込み許可(SPIE)ビットが設定(1)されるなら、割り込みが生成されます。SPIが主装置動作の時に \overline{SS} ピンが入力でLowに駆動されるなら、これもこのSPIFフラグを同様に設定(1)します。対応する割り込み処理ベクタを実行する時にSPIFはハードウェアによって解除(0)されます。代わりにSPIFが設定(1)されたSPI状態レジスタ(SPSR)を始めに読み、その後にSPIデータレジスタ(SPDR)をアクセスすることによってもSPIFフラグは解除(0)されます。

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

データ転送中にSPIデータレジスタ(SPDR)が書かれると、このWCOLビットが設定(1)されます。WCOLビット(とSPIFビット)はWCOLが設定(1)されたSPI状態レジスタ(SPSR)を始めに読み、その後にSPIデータレジスタ(SPDR)をアクセスすることによって解除(0)されます。

●ビット5~1 – Res : 予約 (Reserved Bit)

これらのビットは予約されており、常に0として読まれます。

●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13-4.参照)。これは最小SCK周期が2CPUクロック周期であることを意味します。SPIが従装置として設定される時に、SPIは 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。

ATtiny87/167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161頁をご覧ください。

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

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

SPIデータレジスタはSPI移動レジスタとレジスタファイル(汎用レジスタ)間のデータ転送に使われる読み書き可能なレジスタです。このレジスタへの書き込みがデータ送信を開始します。このレジスタの読み込みは移動レジスタの受信緩衝部読み出しを引き起こします。

13.3. データ転送形式

直列データに関してはSPI制御レジスタ(SPCR)のSCK位相(CPHA)とSCK極性(CPOL)制御ビットによって決定されるSCK位相と極性で4つの組み合わせがあります。このSPIデータ転送形式は図13-3と図13-4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13-2と表13-3を要約することによって明解にされます。

表13-5. 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13-3. SPIデータ転送形式 (CPHA=0)

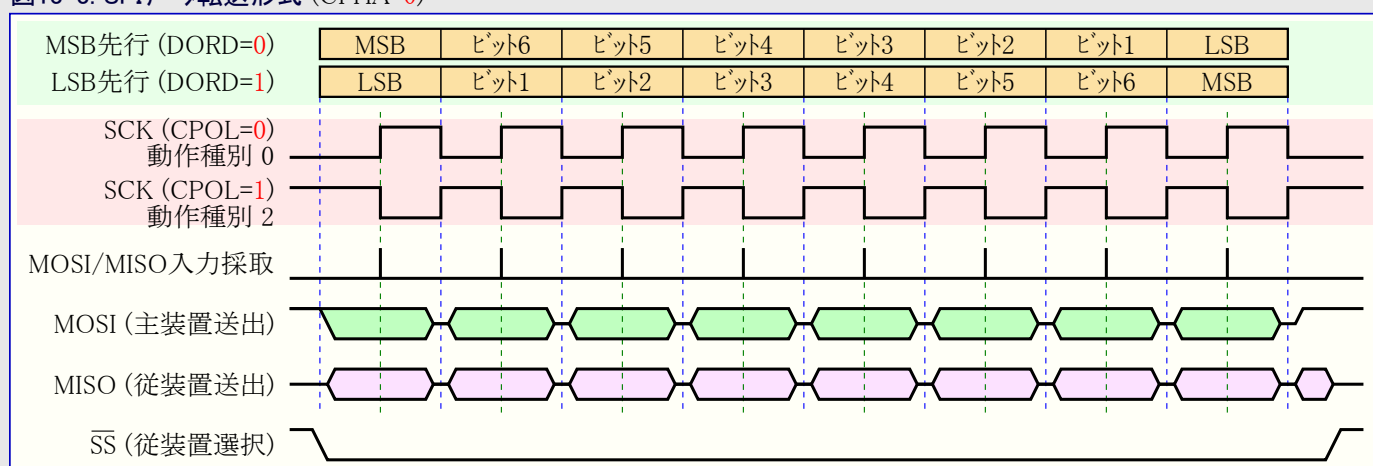
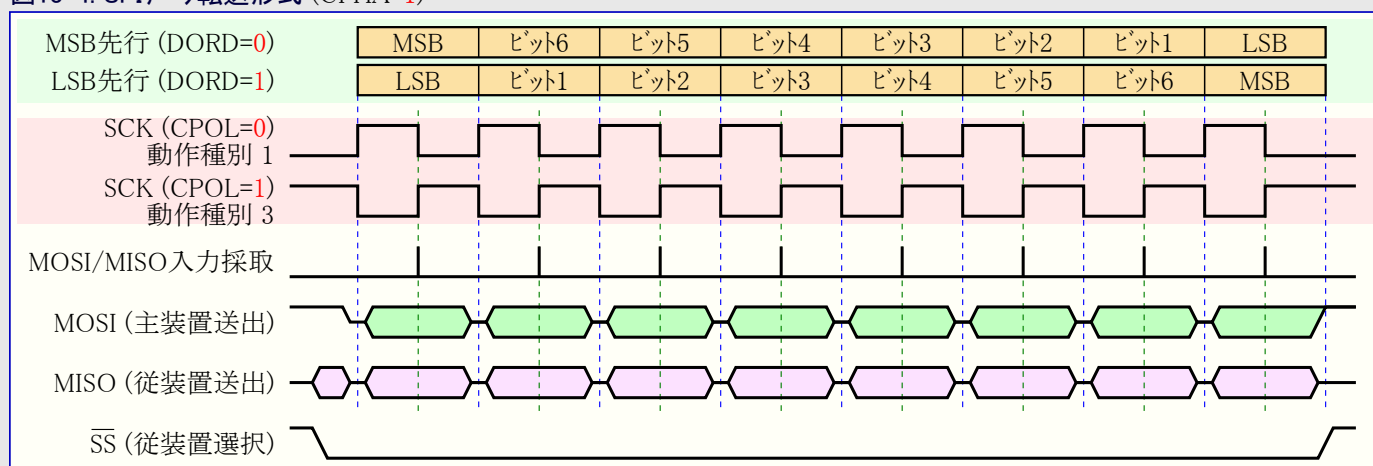


図13-4. SPIデータ転送形式 (CPHA=1)



14.3. 機能説明

14.3.1. 3線動作

USIの3線動作は直列周辺インターフェース(SPI)の動作種別0と1に準拠していますが、従装置選択(SS)ピン機能を持ちません。けれども、この特性(機能)は必要ならばソフトウェアで実現できます。この動作で使われるピン名はDI,DO,USCKです。

図14-2. 3線動作簡略構成図

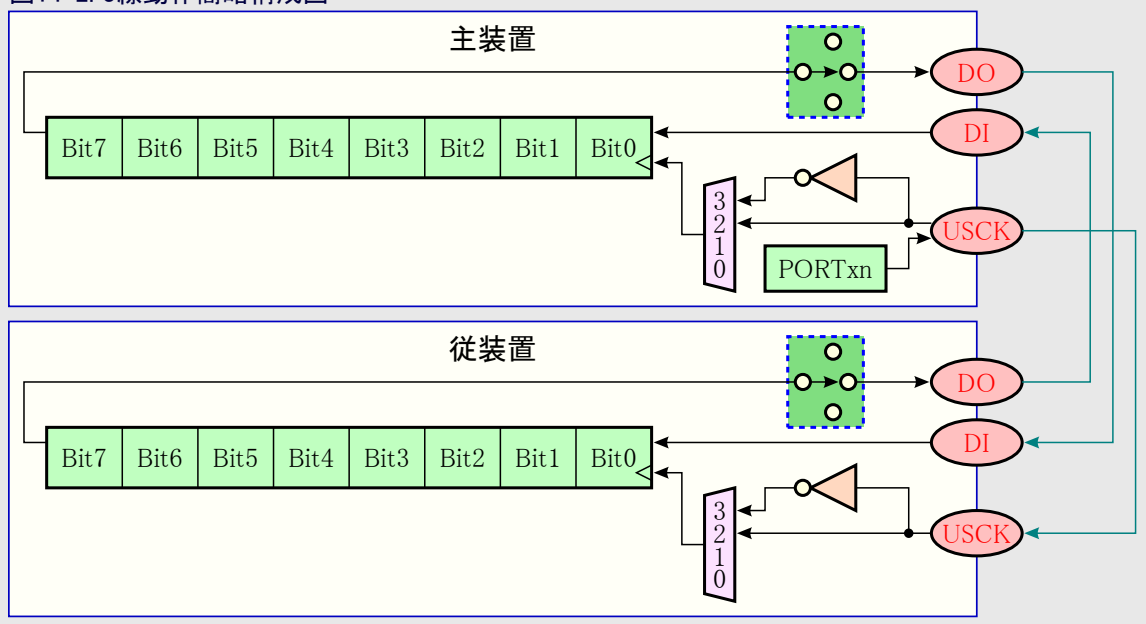
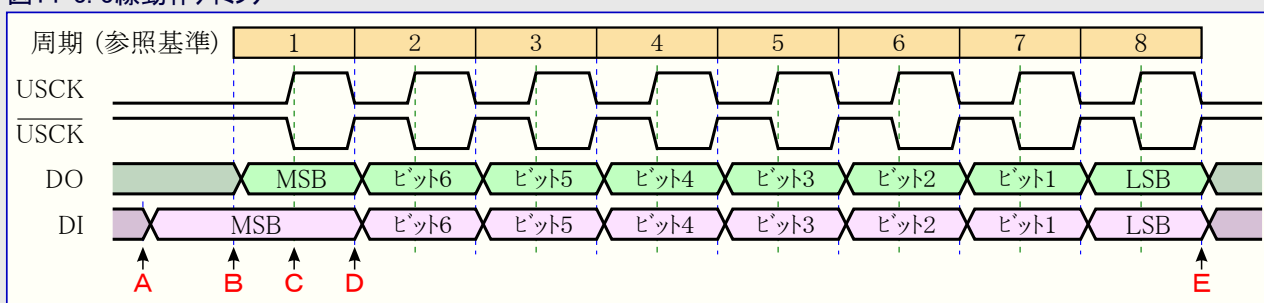


図14-2は3線動作での2つのUSI部(一方は主装置、他方は従装置)動作を示します。このような方法で連結された2つのUSIデータレジスタは8USCKクロック後、各々のレジスタのデータが交換されます。この同じクロック(USCK)がUSIの4ビット計数器も増加(+1)します。従って**計数器溢れ割り込み要求フラグ(USIOIF)**は転送が完了される時を決めるのに使えます。

このクロックはPORTレジスタ経由USCKピン1/0交互切り替え、またはUSICRのUSITCビットへの1書き込みによる主装置装置ソフトウェアによって生成されます。

図14-3. 3線動作タイミング



3線動作タイミングは図14-3で示されます。図の最上部はUSCK周期参照基準です。これらの各周期に対して1ビットが**USIデータレジスタ(USIDR)**に移動されます。USCKタイミングは両方の外部クロック動作について示されます。外部クロック動作0(**USICS0=0**)でのDIは上昇端で採取され、DOは下降端で変更(USIDRが1つ移動)されます。外部クロック動作1(**USICS0=1**)は外部クロック動作0に対して逆端を使用、換言すると下降端でデータを採取、上昇端で出力を変更します。USIクロック動作種別はSPIデータ動作種別0と1に対応します。

このタイミング図(図14-3)を参照すると、バス転送は次の手順を含みます。

1. 主装置と従装置はそれらのデータ出力を設定し、使用規約に依存してそれらの出力駆動部を許可します(AとB)。この出力は送信されるべきデータをUSIデータレジスタへ書くことによって設定します。出力の許可はポートBのデータ方向レジスタ内の対応するビットを設定(1)することによって行います。A点とB点の優先順はありませんが、両方共にデータが採取されるC点の最低1/2USCK周期前で行わなければならないことに注意してください。これはデータ設定の必要条件が満足されることを保証するために行われなければなりません。4ビット計数器は0にリセットします。
2. 主装置はUSCK線をソフトウェアで2度切り替えることによってクロックパルスを生成します(CとD)。従装置と主装置のデータ入力(DI)ピンのビット値は最初のエッジ(C)でUSIによって採取され、データ出力は逆端(D)で変更されます。4ビット計数器は両端で計数します。
3. レジスタ(バイト)転送完了のために手順2.が8回繰り返されます。
4. 8クロックパルス(換言すると16クロック端)後、計数器が溢れて転送完了を示します。転送したバイトデータは新規転送が開始され得るのに先立って直ぐに処置されなければなりません。溢れ割り込みはプロセッサがアイドル動作に設定されているなら、プロセッサを起動復帰します。使用規約によって、従装置は直ぐに出力をHi-Z設定にできます。

14.3.2. SPI主装置操作例

次のコードはSPI主装置としてのUSI使用法を実際に示します。

アセンブリ言語プログラム例

```

SPIM:      STS      USIDR, R16                ;送信データを設定
           LDI      R16, (1<<USIOIF)         ;USIOIFビットのみ1値を取得
           STS      USISR, R16               ;フラグ解除/計数器初期化
           LDI      R16, (1<<USIWM0) | (1<<USICS1) | (1<<USICLK) | (1<<USITC) ;3線動作クロック生成値を取得
;
SPIM_LP:   STS      USICR, R16                ;USCKクロック端発生
           LDS      R16, USISR               ;USI状態フラグを取得
           SBRS     R16, USIOIF              ;計数器溢れでスキップ
           RJMP     SPIM_LP                 ;計数器溢れまで継続
;
           LDS      R16, USIDR               ;受信データを取得
           RET                                ;呼び出し元へ復帰

```

このコードは8命令(+RET)のみ使用の容量最適化です。このコード例はDOとUSCKピンがDDRAまたはDDRB出力として許可される仮定です。この関数が呼び出される前にR16レジスタへ格納した値は従装置に転送され、転送が完了された時に従装置から受信したデータがR16レジスタに格納されて戻ります。

2行目と3行目の命令は計数器溢れ割り込み要求フラグ(USIOIF)を解除(0)し、USI 4ビット計数器値を解除(=0)します。4行目と6行目の命令は3線動作、上昇端クロック、USITCスロープ計数、SCK出力交互切り替え(PA5またはPB2)を設定します。この繰り返しは16回繰り返されます。

次のコードは最高速($f_{SCK}=f_{CK}/4$)でのSPI主装置としてのUSI使用法を実際に示します。

アセンブリ言語プログラム例

```

SPIM_F:    STS      USIDR, R16                ;送信データを設定
           LDI      R16, (1<<USIWM0) | (0<<USICS0) | (1<<USITC)         ;3線動作初期値を取得
           LDI      R17, (1<<USIWM0) | (0<<USICS0) | (1<<USITC) | (1<<USICLK) ;3線動作クロック生成値を取得
;
           STS      USICR, R16                ;MSB転送
           STS      USICR, R17                ;
           STS      USICR, R16                ;ビット6転送
           STS      USICR, R17                ;
           STS      USICR, R16                ;ビット5転送
           STS      USICR, R17                ;
           STS      USICR, R16                ;ビット4転送
           STS      USICR, R17                ;
           STS      USICR, R16                ;ビット3転送
           STS      USICR, R17                ;
           STS      USICR, R16                ;ビット2転送
           STS      USICR, R17                ;
           STS      USICR, R16                ;ビット1転送
           STS      USICR, R17                ;
           STS      USICR, R16                ;LSB転送
           STS      USICR, R17                ;
;
           LDS      R16, USIDR               ;受信データを取得
           RET                                ;呼び出し元へ復帰

```

14.3.3. SPI従装置操作例

次のコードはSPI従装置としてのUSI使用法を実際に示します。

アセンブリ言語プログラム例

```

init:      LDI      R16, (1<<USIWM0) | (1<<USICS1)      ;3線動作,外部クロック値を取得
          STS      USICR, R16                          ;3線動作,外部クロック設定

SPIS:      STS      USIDR, R16                          ;送信データを設定
          LDI      R16, (1<<USIOIF)                    ;USIOIFビットのみ1値を取得
          STS      USISR, R16                          ;フラグ解除/計数器初期化

;
SPIS_LP:   LDS      R16, USISR                          ;USI状態フラグを取得
          SBRS     R16, USIOIF                          ;計数器溢れでスキップ
          RJMP     SPIS_LP                             ;計数器溢れまで継続

;
          LDS      R16, USIDR                          ;受信データを取得
          RET                                           ;呼び出し元へ復帰

```

このコードは8命令(+RET)のみ使用の容量最適化です。このコード例はDDRAまたはDDRBLレジスタでDOピンが出力、USCKピンが入力として設定されると仮定します。この関数が呼び出される前にR16レジスタへ格納した値は主装置に転送され、転送が完了された時に主装置から受信したデータがR16レジスタに格納されて戻ります。

最初の2命令は初期化用だけで、一度だけ実行されるのを必要とすることに注意してください。これらの命令は3線動作と上昇端クロックを設定します。この繰り返しはUSI計数器溢れフラグが設定(1)されるまで繰り返されます。

14.3.4. 2線動作

USIの2線動作はI²C(TWI)バス規約に準拠していますが、出力のスレーブ制限と入力雑音濾波器がありません。この動作で使われるピン名はSCLとSDAです。

図14-4. 2線動作簡略構成図

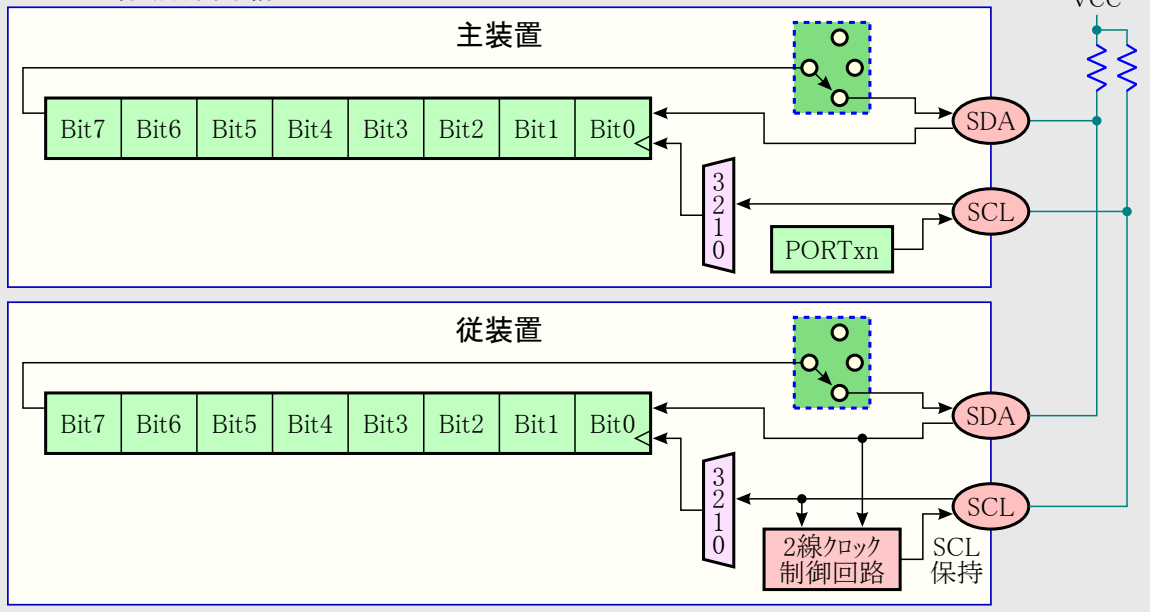


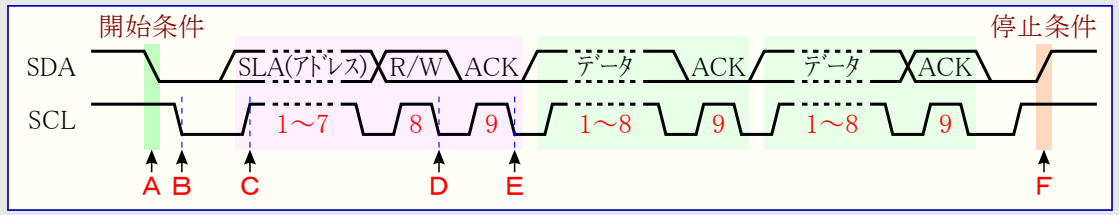
図14-4.は2線動作での2つのUSI部（一方は主装置、他方は従装置）動作を示します。システム動作が使用通信構造に大きく依存するために示されるのは物理層のみです。この段階での主装置動作と従装置動作間の主な違いは常に主装置によって行われる直列クロック生成と、従装置だけがクロック制御部(回路)を使うことです。クロック生成はソフトウェアで実行されなければなりません、移動動作は両装置で自動的に行われます。データの移動に関してこの動作種別で下降端でのクロックだけを実際に使うことに注意してください。従装置はSCLクロックをLowに強制することにより、転送の開始と終了で待ち状態(ウェイト ステート)を挿入できます。これは(主装置が)上昇端を生成してしまっただ後に主装置はSCL線が実際に開放されたかを常に検査しなければならないことを意味します。

このクロックが(4ビット)計数器も増加(+1)するため、計数器溢れ(USIOIF)は転送が完了されたことを示すのに使えます。このクロックは主装置によるPORTAまたはPORTBレジスタ経由のSCLピン交互切り替えによって生成されます。

データ方向は物理層によって与えられません。データの流れを制御するためにTWIバスで使ったような或る種の規約が実装されなければなりません。

このタイミング図(図14-5.)を参照すると、バス転送は次の手順を含みます。

図14-5. 代表的な2線動作タイミング



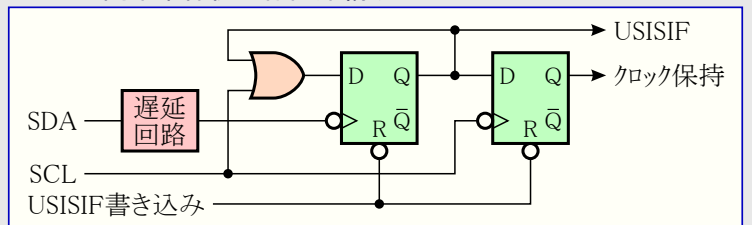
1. 開始条件は主装置によるSCL線High保持中のSDA線Low強制によって生成されます(A)。SDAはUSIデータレジスタのビット7への0書き込みか、またはPORTレジスタでの対応ビットの0設定のどちらかによってLowを強制できます。許可されるべき出力に対してデータ方向(DDR)レジスタのビットが1に設定されなければならないことに注意してください。従装置の開始条件検出器論理回路(図14-6. 参照)が開始条件を検出してUSISIFフラグを設定(1)します。このフラグは必要ならば割り込みを発生できます。
2. 加えて開始条件検出器は主装置がSCL線で上昇端を強制してしまった後にSCL線をLowに保持します(B)。これはアドレスを受信するために移動レジスタを設定する前に休止形態から起動復帰または他の処理を完了することを従装置に許します。これ(保持解除)は(4ビット)計数器をリセット(=0)して開始条件検出フラグ(USISIF)を解除(0)することによって行われます。
3. 主装置は転送されるべき最初のビットを設定してSCL線を開放します(C)。従装置はSCLクロックの上昇端でデータを採取して移動レジスタへ移動します。
4. 従装置アドレスとデータ方向(R/W)を含む8ビットが転送された後、従装置の(4ビット)計数器が溢れてSCL線がLowを強制されます(D)。従装置は主装置の1つがアドレス指定してしまわない場合、SCL線を開放して新規開始条件を待ちます。
5. 従装置がアドレス指定されると、再びSCL線をLowに保持する前の応答(ACK)周期の間中、SDA線をLowに保持します(換言するとD点でSCLを開放する前にUSI(4ビット)計数器レジスタは14に設定されなければなりません)。R/Wビットに依存して主装置が従装置は出力を許可します。このビットが設定(1)の場合、主装置読み込み動作が進行中です(換言すると従装置がSDA線を駆動します)。従装置は応答後(E)にSCL線をLowに保持することができます。
6. 主装置によって停止条件(F)か新規開始条件(再送開始条件)が与えられるまで全て同方向で複数バイトが直ぐに転送できます。従装置はより多くのデータを受信できない場合、最後に受信したバイトデータに 응답しません。主装置が読み込み動作を行うとき、最後のバイトが転送された後の応答ビットをLowに強制することによってこの動作を終了しなければなりません。

14.3.5. 開始条件検出器

開始条件検出器は図14-6.で示されます。SDA線はSCL線の有効な採取を保証するために(50~300nsの範囲で)遅延されます。開始条件検出器は2線動作でだけ許可されます。

開始条件検出器は非同期に動作し、従ってパワーダウン休止動作からプロセッサを起動できます。けれども使う規約がSCL保持時間で制限を持つかもしれません。従ってこの場合にこの特徴(機能)を使う時にCKSELヒューズで設定する発振器起動時間(21頁の「クロックシステムとその配給」参照)も考慮内に取り入れられなければなりません。より多くの詳細については109頁のUSISIFフラグ記述を参照してください。

図14-6. 開始条件検出器回路構成



14.4. USIでの代替使用

14.4.1. 半二重非同期データ転送

14.4.2. 4ビット計数器

14.4.3. 12ビット計数器

14.4.4. エッジ起動外部割り込み

14.4.5. ソフトウェア割り込み

USIの柔軟な設計は直列通信が不要の時に他の作業への使用を許します。

3線動作でのUSIデータレジスタの使用はソフトウェアだけよりも簡潔で高性能のUART実装が可能です。

4ビット計数器は溢れ割り込みとの自立型計数器として使えます。この計数器が外部的にクロック駆動される場合、両クロック端が計数値を増加(+1)することに注意してください。

4ビットUSI計数器と8ビットタイマ/カウンタの1つとの組み合わせは12ビット計数器を作成します。

4ビット計数器を最大値(\$F)に設定することによって追加外部割り込みとして機能できます。溢れフラグと割り込み許可ビットはこの外部割り込みに対して使われます。この特徴(機能)はUSICS1ビットによって選択(=1)されます。

4ビット計数器の溢れ割り込みはクロックストロブによって起動するソフトウェア割り込みとして使えます。

14.5. USI用レジスタ

14.5.1. USIDR – USIデータレジスタ (USI Data Register)

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

USIデータレジスタ(USIDR)をアクセスする時に直列レジスタは直接的にアクセスされます。レジスタが書かれるのと同じ周期で直列クロックが起こると、レジスタは書いた値を含み、移動は実行されません。(左)移動動作はUSICS1,0ビット設定に従って実行されます。この移動動作は外部クロック端、**タイマ/カウンタ0比較一致**により、または**USICLKスロープ**ビットを使うソフトウェアによって直接的に制御できます。2または3線動作種別が選択されない(USIWM1,0=00)時でも、外部データ入力(DI/SDA)と外部クロック入力(USCK/SCL)の両方が未だ移動レジスタによって使われ得ることに注意してください。

動作種別に従って使う出力ピン(DOまたはSDA)は出力ラッチを経由してデータレジスタの最上位ビット(ビット7)に接続されます。この出力ラッチは外部クロック元が選択される(USICS1=1)時に直列クロック周期の前半中が透過(トランスパレント)で、内部クロック元が使われる(USICS1=0)時に定常的な透過です。ラッチが透過である限り、新規MSBが書かれると、出力は直ちに更新されます。このラッチは違う(逆の)クロック端でデータ入力が採取され、データ出力が更新されるのを保証します。

移動レジスタからの出力データを許可するために、ピンに対応するデータ方向レジスタ(DDxn)が1に設定されなければならないことに注意してください。

14.5.2. USIBR – USI緩衝レジスタ (USI Buffer Register)

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

移動レジスタの内容は転送が完了される時にUSI緩衝レジスタ(USIBR)へ格納され、CPUが受信したデータを読むとき、**USIデータレジスタ(USIDR:直列移動レジスタ)**をアクセスする代わりにUSIデータ緩衝部(USIBR)をアクセスできます。これはUSI制御が微妙なタイミングでなくなるため、他のプログラム処理にも取り扱うためのCPU時間を与えます。USIのフラグはUSIDRを読む時と同様に設定されます。

14.5.3. USISR – USI状態レジスタ (USI Status Register) USISR

ビット (\$B9)	7 USISIF	6 USIOIF	5 USIPF	4 USIDC	3 USICNT3	2 USICNT2	1 USICNT1	0 USICNT0	USISR
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

この状態レジスタは割り込み要求フラグ、状態フラグ、計数値を含みます。

● ビット7 – USISIF : 開始条件検出割り込み要求フラグ (Start Condition Interrupt Flag)

2線動作選択時に**開始条件**が検出されると、USISIFフラグが設定(1)されます。出力禁止動作または3線動作の選択時、USCKピンのどのエッジもがこのフラグを設定(1)します。

ステータスレジスタ(SREG)の全割り込み許可(I)ビットと**USI制御レジスタ(USICR)**で**開始条件検出許可(USISIE)ビット**が設定(1)の場合にこのフラグが設定(1)されると、割り込みが生成されます。このフラグはUSISIFビットへの論理1書き込みによってのみ解除(0)されます。このビットの解除(0)は2線動作でのSCLの**開始条件保持**を解除(開放)します。

開始条件検出割り込みは全ての**休止形態**からプロセッサを起動します。

● ビット6 – USIOIF : 計数器溢れ割り込み要求フラグ (Counter Overflow Interrupt Flag)

このフラグは4ビット計数器が溢れると(換言すると15から0への遷移で)設定(1)されます。SREGの全割り込み許可(I)ビットとUSICRで**計数器溢れ割り込み許可(USIOIE)ビット**が設定(1)の場合にこのフラグが設定(1)されると、割り込みが生成されます。このフラグはUSIOIFビットへの論理1書き込みによってのみ解除(0)されます。このビットの解除(0)は2線動作でのSCLの**計数器溢れ保持**を解除(開放)します。

計数器溢れ割り込みは**アイドル休止動作**からプロセッサを起動します。

● ビット5 – USIPF : 停止条件検出フラグ (Stop Condition Flag)

2線動作が選択された時に**停止条件**が検出されると、USIPFフラグが設定(1)されます。このフラグはこのビットへの1書き込みによって解除(0)されます。これが割り込み要求フラグでないことに注意してください。この合図は2線バス主装置調停の実装時に有用です。

● ビット4 – USIDC : 出力データ衝突フラグ (Data Output Collision Flag)

USIデータレジスタ(USIDR)のビット7(MSB)が物理ピン値と異なる時にこのビットは論理1です。このフラグは2線動作が使われる時にだけ有効です。この合図は2線バス主装置調停の実装時に有用です。

●ビット3~0 – USICNT3~0 : 計数値 (Counter Value)

これらのビットは現在の4ビット計数値を反映します。この4ビット計数値はCPUによって直接的に読み書きできます。

この4ビット計数器は外部クロック端検出器、タイマ/カウンタ0比較A一致、USICLKまたはUSITCストローブビットを使うソフトウェアのどれかによって生成した各クロックに対して1つ増加します。このクロック元はUSICS1,0ビット設定に依存します。外部クロック動作に対してUSITCストローブビットへの書き込みによって生成されるのをクロックに許す特別な特徴が付加されます。この特徴は外部クロック元(USICS1=1)設定とUSICLKへの1書き込みによって許可されます。

出力禁止動作(USIWM1,0=00)が選択されている時でも、外部クロック入力(USCK/SCL)が未だ計数器によって使われることに注意してください。

14.5.4. USICR – USI制御レジスタ (USI Control Register)

ビット (\$B8)	7	6	5	4	3	2	1	0	
	USISIE	USIOIE	USIWM1	USIWM0	USICS1	USICS0	USICLK	USITC	USICR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	W	W	
初期値	0	0	0	0	0	0	0	0	

この制御レジスタは割り込み許可制御、動作種別設定、クロック選択設定、クロックストローブを含みます。

●ビット7 – USISIE : 開始条件検出割り込み許可 (Start Condition Interrupt Enable)

このビットを1に設定することが開始条件検出割り込みを許可します。ステータスレジスタ(SREG)の全割り込み許可(I)ビットとUSISIEビットが1に設定され、保留割り込みありなら、割り込みが直ちに実行されます。より多くの詳細については109頁の開始条件検出割り込み要求フラグ(USISIF)記述を参照してください。

●ビット6 – USIOIE : 計数器溢れ割り込み許可 (Counter Overflow Interrupt Enable)

このビットを1に設定することが計数器溢れ割り込みを許可します。ステータスレジスタ(SREG)の全割り込み許可(I)ビットとUSIOIEビットが1に設定され、保留割り込みありなら、割り込みが直ちに実行されます。より多くの詳細については109頁の計数器溢れ割り込み要求フラグ(USIOIF)記述を参照してください。

●ビット5,4 – USIWM1,0 : 動作種別選択 (Wire Mode)

これらのビットは使われるべき動作種別(2または3線動作)を設定します。基本的に出力の機能だけがこれらのビットによって影響を及ぼされます。データとクロック入力は選択した動作種別によって影響を及ぼされず、常に同じ機能を持ちます。従って出力が禁止される時でも移動レジスタと計数器は外部的にクロック駆動でき、入力データが採取されます。USIWM1,0とUSI動作の関連は表14-1で要約されます。

表14-1. USI操作とUSIWM1,0の関係

USIWM1	USIWM0	説明
0	0	出力、クロック保持、開始条件検出器は禁止。ポートピンは標準として動作。
0	1	3線動作。DO,DI,USCKピンを使用。 データ出力(DO)ピンはPORTBレジスタの対応ビットを無視します。けれども対応するDDRBビットは未だデータ方向を制御します。ポートピンが入力として設定(DDBn=0)されると、そのピンのプルアップはPORTBビットによって制御されます。 データ入力(DI)と直列クロック(USCK)ピンは標準ポート操作に影響を及ぼしません。主装置として動作する時のクロックパルスはデータ方向が出力に設定されている(DDBn=1)間のPORTBレジスタの交互切り替えによってソフトウェアで生成されます。USICRのUSITCビットがこの目的に使えます。
1	0	2線動作。SDA(DI)とSCL(USCK)ピンを使用。(注1) 直列データ(SDA)と直列クロック(SCL)ピンは双方向でオープンドレイン出力駆動を使います。この出力駆動部はDDRBレジスタでSDAとSCLに対応するビットの設定(=1)によって許可されます。 SDAピンに対して出力駆動部が許可されると、出力駆動部はUSIデータレジスタ(USIDR)の出力またはPORTBレジスタの対応ビットが0ならばSDA線をLowに強制します。さもなければ、SDA線は駆動されません(換言すると開放されます)。SCLピン出力駆動部が許可されると、SCL線はPORTBレジスタの対応ビットが0なら、または開始条件検出器によってLowを強制されます。さもなければSCL線は駆動されません。 SCL線は出力が許可され、開始条件検出器が開始条件を検出すると、Lowに保持されます。開始条件検出フラグ(USISIF)の解除(0)がその線を開放します。SDAとSCLピン入力はこの動作の許可によって影響を及ぼされません。SDAとSCLピンのプルアップは2線動作で禁止されます。
1	1	2線動作。SDAとSCLピンを使用。 計数器溢れが起きた時から計数器溢れフラグ(USIOIF)が解除(0)されるまでもSCL線がLowに保持されることを除いて、上の2線動作と同じ動作です。

注1: DIとUSCKピンは動作種別間での混乱を避けるため、各々直列データ(SDA)と直列クロック(SCL)に改称されます。

●ビット3,2 – USICS1,0 : クロック選択 (Clock Source Select)

これらのビットは移動レジスタと計数器に対するクロック元を設定します。外部クロック元(USCK/SCL)を使う時にデータ出力ラッチはデータ入力(DI/SDA)の採取の逆端で出力が変更されるのを保証します。タイマ/カウンタ0比較A一致またはソフトウェア ストローブが選択されると、出力ラッチは透過で、従って出力は直ちに變更されます。USICS1,0ビットの解除(=00)がソフトウェア ストローブ選択を許可します。この選択を使う時のUSICKLKビットへの1書き込みは移動レジスタと計数器の両方をクロック駆動します。外部クロック元(USICS1=1)に対してUSICKLKビットはもはやストローブとして使われませんが、外部クロックとUSITCストローブ ビットによるソフトウェア クロック間を選択します。

表14-2. は移動レジスタと4ビット計数器に対して使われるクロック元とUSICS1,0、USICKLK設定間の関連を示します。

表14-2. USICS1,0とUSICKLK設定との関係

USICS1	USICS0	USICKLK	移動レジスタ クロック元	4ビット計数器クロック元
0	0	0	クロックなし	クロックなし
		1	ソフトウェア クロック ストローブ (USICKLK)	ソフトウェア クロック ストローブ (USICKLK)
	1	X	タイマ/カウンタ0比較A一致	タイマ/カウンタ0比較A一致
1	0	0	外部クロック上昇端	外部クロック両端
	1		外部クロック下降端	
	0	1	外部クロック上昇端	ソフトウェア クロック ストローブ (USITC)
	1		外部クロック下降端	

●ビット1 – USICKLK : クロック ストローブ (Clock Strobe)

ソフトウェア ストローブ選択が選ばれるように行うことによってUSICS1,0ビットがゼロ(=00)に設定されるなら、このビット位置への1書き込みは移動レジスタを1段移動して計数器を1つ増加します。このクロック ストローブが実行される時、直ちに、換言すると同じ命令周期内で出力が替わります。移動レジスタ内に移動した値は直前の命令周期で採取されます。このビットは0として読まれます。

外部クロック元が選択される(USICS1=1)と、USICKLK機能はクロック ストローブからクロック選択レジスタに変更されます。この場合のUSICKLKビットの設定(1)は4ビット計数器に対するクロック元としてのUSITCストローブを選びます(表14-2.参照)。

●ビット0 – USITC : クロック値切り替え (Toggle Clock Port Pin)

このビット位置に1を書くことはUSCK/SCL値を0から1、または1から0のどちらかへ切り替えます。この切り替えはデータ方向レジスタでの設定と無関係ですが、PORT値がピンで見られるべきなら、DDA5またはDDB2が出力として設定(=1)されなければなりません。この特徴は主装置を実現する時の容易なクロック生成を許します。このビットは0として読まれます。

外部クロック元が選択され(USICS1=1)、USICKLKビットが1に設定されると、USITCストローブ ビットへの書き込みは直接的に4ビット計数器をクロック駆動します。これは主装置として動作する時に転送が行われる時の早い検出を許します。

14.5.5. USIPP – USIピン位置レジスタ (USI Pin Position Register)

ビット (\$BC)	7	6	5	4	3	2	1	0	
	–	–	–	–	–	–	–	USIPOS	USIPP
Read/Write	R	R	R	R	R	R	R	R/W	
初期値	0	0	0	0	0	0	0	0	

●ビット7~1 – Res : 予約 (Reserved)

これらのビットは予約されており、常に0として読まれます。

●ビット1 – USIPOS : USIピン位置 (USI Pin Position)

このビットの設定(1)と解除(0)がUSIピン位置を変更します。

表14-3. USIピン位置

USIPOS		USIピン位置	
0	ポートB (既定)	DI,SDA	PB0 – (OC1AU/PCINT8)
		DO	PB1 – (OC1BU/PCINT9)
		USCK,SCL	PB2 – (OC1AV/PCINT10)
1	ポートA (代替)	DI,SDA	PA4 – (ICP1/MOSI/ADC4/PCINT4)
		DO	PA2 – (OC0A/MISO/ADC2/PCINT2)
		USCK,SCL	PA5 – (T1/SCK/ADC5/PCINT5)

15. LIN/UART – (局所相互連結網制御器またはUART)

局所相互連結網(LIN:Local Interconnect Network)は分配された車載応用に於ける電子機械節点(ノード)の制御を効果的に支援する直列通信規約ですが、工業応用に等しく適合します。LINバスの主な特性は以下です。

- ・ 複数従装置を持つ単一主装置の概念
- ・ 一般的なUART/SCIインターフェースに基づく安価なシリコン実装
- ・ 従節点(ノード)でのチップ上発振器付き自己同期
- ・ 先行して計算可能な信号伝播時間を持つ確定的信号転送
- ・ 最大20kビット/sの速度

LINはCANの帯域や多能性が必要とされない費用効率的なバス通信を提供します。信号線の駆動部/受信部の仕様はISO9141 NRZ規格に一致する必要があります。

LINが必要とされない場合、この制御器は代わりにUART(Universal Asynchronous serial Receiver and Transmitter)としてプログラムすることができます。

15.1. LIN特徴

- ・ LIN2.1(LIN1.3互換)のハードウェア実装
- ・ LIN2.1仕様の「LIN作業の流れの概念」に基づく、小さく、CPU効率的で独立した主/従ルーチン
- ・ 自動的なLIN先頭部処理と無関係なLINフレームの濾過
- ・ 自動LIN応答処理
- ・ 拡張されたLIN異常検出と合図
- ・ ハードウェアによるフレーム時間超過検出
- ・ 「データ内中断」支援能力
- ・ 正しいフレームの完全性を保証するための自動再同期
- ・ 十分に柔軟な拡張フレーム支援能力

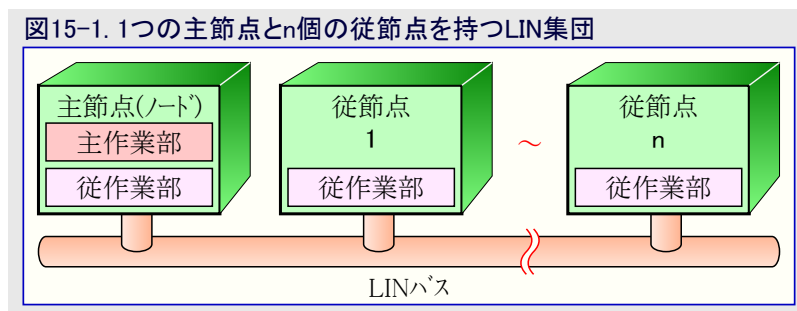
15.2. UART特徴

- ・ 全二重動作(独立した直列受信と送信の処理)
- ・ 非同期動作
- ・ 高分解能ボーレート生成器
- ・ 8ビットデータ、奇/偶/なしのパリティビット、1停止ビットのフレームのハードウェア支援
- ・ データオーバーランとフレーミング異常の検出

15.3. LIN規約

15.3.1. 主作業と従作業

LIN集団は1つの主作業と多数の従作業から成ります。主節点(ノード)は従作業だけでなく主作業も含みます。他の全ての節点は従作業だけを含みます。



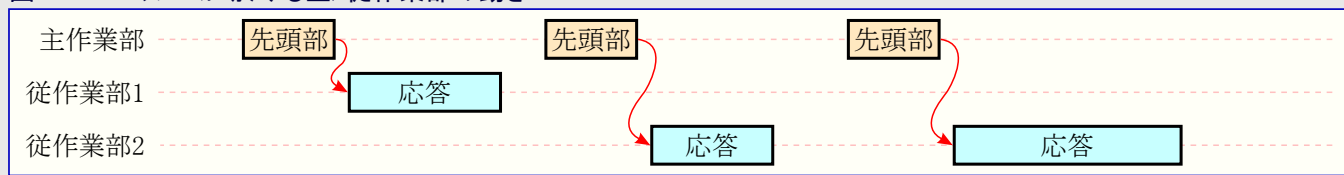
主作業はバス上にどのフレームが転送されるべきかとその時を決めます。従作業は各フレームによって転送されたデータを提供します。主と従の両作業はフレーム処理部の一部です。

15.3.2. フレーム

フレームは(主作業によって供給される)先頭部と(従作業によって供給される)応答から成ります。

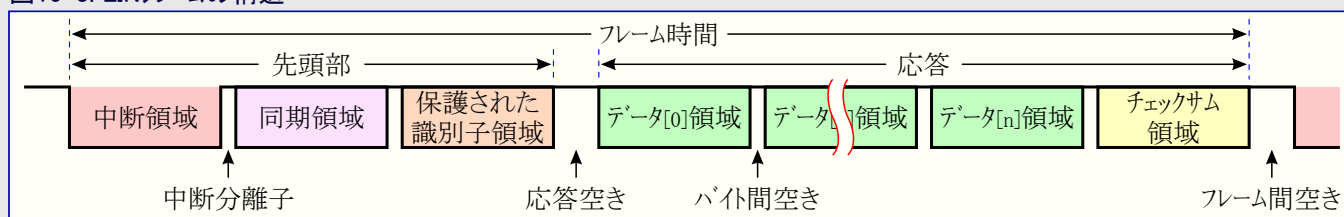
先頭部は保護された識別子(PROTECTED IDENTIFIER)領域が後続する、中断(BREAK)と同期(SYNC)の領域から成ります。識別子はそのフレームの目的を一意に定義します。識別子に関連する応答を提供するように指定された従作業部はそれを送出します。応答はデータ(DATA)領域とチェックサム(CHECKSUM)領域から成ります。

図15-2. LINフレームに於ける主/従作業部の動き



その識別子に関連するデータを待っている従作業部は、その応答を受信してチェックサム照合後に転送されたデータを使います。

図15-3. LINフレームの構造



注: 各バイト領域はLSB先行の連続バイトとして送信されます。

15.3.3. データ転送

フレームに於いて信号または診断メッセージの2つのデータ形式が転送されるかもしれません。

- ・ 信号

信号はフレームのデータ領域内に詰められた数量値またはバイト配列です。信号は同じ識別子を持つ全てのフレームのデータ領域の常に同じ位置に存在します。

- ・ 診断メッセージ

診断メッセージは2つの予約された識別子を持つフレームで転送されます。データ領域の解釈はデータ領域自体だけでなく通信節点(ノード)の状態にも依存します。

15.3.4. 計画表

(主節点(ノード)に於ける)主作業は計画表に基づいてフレーム先頭部を送信します。計画表は各先頭部に対する識別子と、フレームの開始と後続するフレーム間の間隔を指定します。主応用は異なる計画表を用い、それらの中から選択するかもしれません。

15.3.5. LIN1.3との互換性

LIN2.1はLIN1.3の上位仕様です。

LIN2.1の主節点(ノード)はLIN1.3とLIN2.1両方の従節点から成る集合を処理することができます。そして主装置はLIN1.3従装置から新しいLIN2.1機能を要求することを避けます。

- ・ 強化されたチェックサム
- ・ 形態再設定と診断
- ・ 自動ポーレート検出
- ・ “応答異常”状態監視

LIN2.1従節点はLIN1.3主節点で操作することができません(例えば、LIN1.3主装置は強化されたチェックサムを支援しません)。

LIN2.1物理層はLIN1.3物理層と過去互換です。しかし、別物ではありません。LIN2.1物理層は大きな必要条件を整え、換言すると、LIN2.1物理層を使う主節点はLIN1.3集合で動作することができます。

15.4. LIN/UART制御器

LIN/UART制御器は3つの主機能に分けられます。

- Tx LIN先頭部機能
- Rx LIN先頭部機能
- LIN応答機能

これらの機能は主に2つの役務を使います。

- Rx役務
- Tx役務

これら2つの役務が基本的にUART役務なので、制御器はUART機能に切り替えることもできます。

15.4.1. LIN概要

LIN/UART制御器はLINソフトウェア応用構造に可能な限り近く合致するように設計されています。LINソフトウェア応用は多数の従作業と1つの主作業の独立した作業として開発されます(113頁の「15.3.4. 計画表」項参照)。ATtiny87/167はこの見通しに従います。主作業と従作業間の繋がりは一且新しい識別子が利用可能になると割り込みルーチンが呼ばれる交差点だけです。従って、主節点では主と従の両作業を収容し、Tx LIN先頭部機能は識別子の存在する従作業に警報を出します。同様に、従節点ではRx LIN先頭部機能が識別子の存在する従作業に警報を出します。

従作業が識別子の存在を警告される時に、応答で何を行うのかを知るため、最初にそれを分析しなければなりません。ハードウェアのフラグが60(\$3C)~63(\$3F)の特定識別子の1つの存在を認識します。

LIN通信に関しては、管理のために以下の4つの割り込みだけが必要です。

- LIDOK : 新しいLIN識別子利用可
- LRXOK : LIN応答受信
- LTXOK : LIN応答送信
- LERR : LIN異常

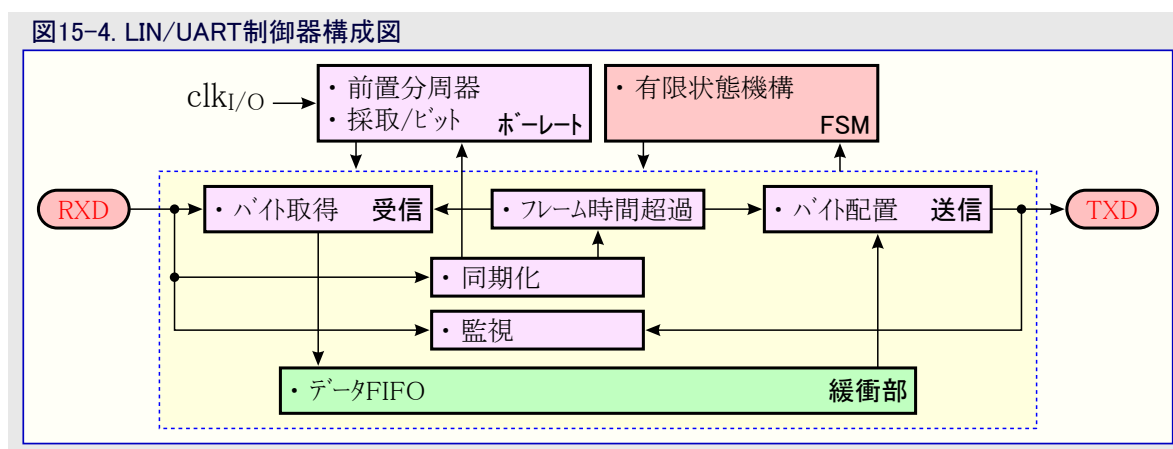
起動管理はLIN2.1に関して最小5ビットのLow(\$F0)、LIN1.3に関して8ビットのLow(\$80)を送出する節点(ノード)と、UART起動能力を使って自動化することができます。LIN起動信号でのピン変化割り込みは休止形態の1つのデバイスを抜けるのに使うこともできます。

62(\$3E)と63(\$3F)の拡張フレーム識別子は使用者定義メッセージ形式と将来のLIN形式の組み込みを許すために予約されています。UARTによって提供されるバイト転送形態はLIN規約に適応するLIN従装置の上位互換を保証します。

15.4.2. UART概要

LIN/UART制御器は伝統的なUARTとしても機能することができます。既定により、UARTは全二重制御器として動作します。これは試験目的に局所戻し閉路を持ちます。UARTは送信用に1文字と受信用に2文字分の緩衝する能力を持ちます。受信緩衝部は1つの8ビット直列レジスタと後続する1つの独立した緩衝部レジスタから作られます。応用が文字を置くまたは得る時の自動的なフラグ管理が実装され、故にソフトウェアの付随作業を減らします。送信と受信の役務が独立しているため、使用者は2つの役務の1つが使われない時に1つのデバイスピンを節約することができます。UARTはどんなクロック周波数と目的ボーレートでも2%の最大誤差を提供する強化されたボーレート生成器を持ちます。

15.4.3. LIN/UART制御器構造



15.4.4. LIN/UART指令概要

図15-5. LIN/UART指令依存性

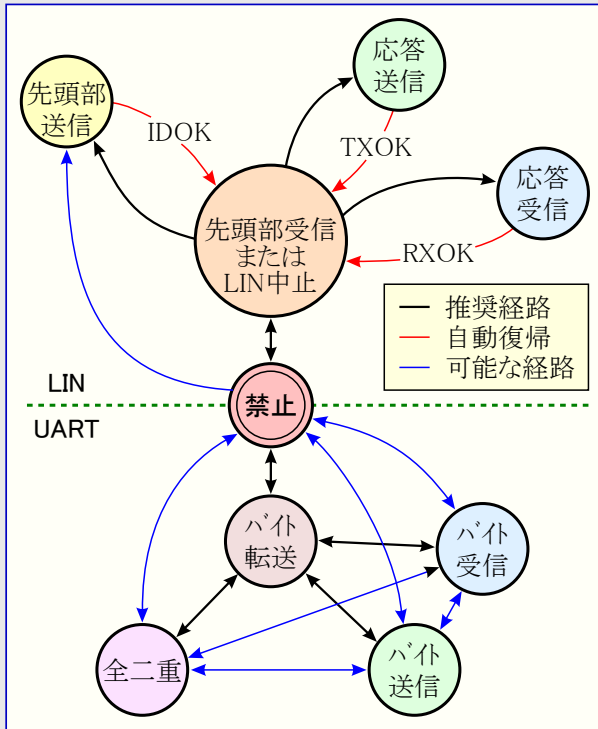


表15-1. LIN/UART指令一覧

LENA	LCMD2~0	指令	注釈
0	x x x	周辺機能禁止	
	0 0 0	先頭部受信 - LIN中止	LIN撤回
	0 0 1	先頭部送信	送信後、LCMD2~0=000
	0 1 0	応答受信	受信後、LCMD2~0=000
1	0 1 1	応答送信	送信後、LCMD2~0=000
	1 0 0	バイト転送	CRCなし、時間超過監視なし、LTXDL=LRXDL=0
	1 0 1	バイト送信	(LINDLRは読み込み専用レジスタ)
	1 1 0	バイト受信	
	1 1 1	全二重	

15.4.5. 許可/禁止

LINCRLレジスタの許可(LENA)ビットの設定(1)がLIN/UART制御器を許可します。LIN/UART制御器を禁止するにはLENAビットが0を書かれなければならない。待ち状態が全く実装されないため、禁止指令は直ちに行われます。

15.4.6. LIN指令

LIN制御レジスタ(LINCRL)の動作形態(LCMD2)ビットの解除(0)がLIN指令を許可します。

表15-1.で示されるように、LINCRLレジスタの指令(LCMD1,0)ビットによって制御される4つの機能が利用可能です(図15-5.参照)。

15.4.6.1. 先頭部受信/LIN中止機能

この機能(または状態)は主に制御器の撤回形態です。

制御器が主作業を実行しなければならない時に、この状態は先頭部送信指令を許可する前の開始位置です。

制御器が従作業だけを実行しなければならない時に、LIN先頭部検出/獲得が背面機能として許可されます。このような獲得(先頭部受信機能)の最後に於いて、自動的に適切なフラグが設定(1)され、LIN1.3ではLINデータ長レジスタ(LINDLR)が符号化されていない長さ値で設定されます。

この状態は応答送信または応答受信の指令を許可する前の開始位置でもあります。

機能(換言すると、先頭部送信、応答送信、または応答受信)の走行はLIN制御レジスタ(LINCRL)の指令(LCMD1,0)ビットを解除(00)することによって中止することができます(122頁の「データ内での中断」をご覧ください)。この場合、他のソフトウェア作業に通知するために、LIN異常レジスタ(LINERR)の中断(LABORT)フラグが設定(1)されます。待ち状態が全く実装されないため、中止指令は直ちに行われます。

先頭部受信機能は以下に対して責任があります。

- ・ 中断(BREAK)領域検出
- ・ 同期(SYNC)領域を分析するハードウェア再同期化
- ・ 保護された識別子(PROTECTED IDENTIFIER)領域の受信、パリティ制御とLIN1.3の場合に於けるLINDLRレジスタの更新
- ・ フレーム時間超過監視の開始
- ・ LIN通信の完全性検査

15.4.6.2. 先頭部送信機能

LIN規約に従って、主作業だけがこの機能を許可しなければなりません。先頭部は設定されたボーレート(LINボーレートレジスタ(LINBRR)とLINビットタイミングレジスタ(LINBTR)参照)と適切な間隔で送出されます。

制御器は以下に対して責任があります。

- ・ 中断(BREAK)領域の送信 - 13個の優先ビット
- ・ 同期(SYNC)領域の送信 - 文字(\$55)
- ・ 保護された識別子(PROTECTED IDENTIFIER)領域の送信。これはLIN識別子レジスタ(LINIDR)の完全な内容です(検査ビット自動内包)。

この送信の最後に於いて、制御器は適切なフラグを設定した後で自動的に先頭部受信/LIN中止状態(換言すると、指令(LCMD1,0)ビット=00)に復帰します。この機能は制御器を先頭部受信機能後と同じ設定にします。これはLIN1.3に於いて先頭部送信機能の最後でLINデータ長レジスタ(LINDLR)が符号化されていない長さ値で設定されることを意味します。

この機能中、制御器は以下に対しても責任があります。

- ・ フレーム時間超過監視の開始
- ・ LIN通信の完全性検査

15.4.6.3. 応答送信と応答受信の機能

これらの機能はLIN節点(ノード)の従作業によって開始されます。これらは先頭部送信(主作業)後、または先頭部受信(従作業に属すると見做される)後に使われなければなりません。応答送信指令が送られると、送信が始まります。応答受信指令は先頭ビットの最終直列ビットの受信(停止ビットの前)までに送ることができます。

LIN1.3では先頭部間隔がLINデータ長レジスタ(LINDLR)を構成設定します。LIN2.1では使用者がLINDLRレジスタを、応答受信に対してLIN受信データ長(LRXDL3~0)または応答送信に対してLIN送信データ長(LTXDL3~0)のどちらかを構成設定しなければなりません。

指令開始時、制御器はチェックサム計算に対して正しい規則を適用するためにLIN制御レジスタ(LINCR)のLIN1.3形態(LIN13)ビットを調べます。データ(DATA)ビットと保護された識別子(PROTECTED IDENTIFIER)に渡るチェックサム計算は拡張チェックサムと呼ばれ、LIN2.1従装置との通信に用いられます。データ(DATA)ビットだけに渡るチェックサム計算は古典的チェックサムと呼ばれ、LIN1.3従装置との通信に用いられます。60~63(\$3C~\$3F)の識別子が常に古典的チェックサムを使うことに注意してください。

この送受信の最後に於いて、適切なフラグを設定した後で制御器は自動的に先頭部受信/LIN中止状態(換言すると、指令(LCMD1,0)ビット=00)に復帰します。

LIN異常が起きた場合、送信または受信は中止され、適切なフラグが設定され、そしてLINバスは劣性状態のままにされます。

これらの関数中、制御器は以下に対して責任があります。

- ・ チェックサム操作部の初期化
- ・ チェックサム計算の更新と共に'n'データの送信または受信
- ・ チェックサム領域の送信または検査
- ・ フレーム時間超過監視の開始
- ・ LIN通信の完全性検査

制御器が応答を送信または受信の間に、中断(BREAK)と同期(SYNC)の領域を検知し、この新先頭部の識別子を記録することができます。勿論、直前の応答での特定の異常はこの識別子受信と共に保持されます。

15.4.6.4. LIN応答のデータの扱い

LIN応答のデータ用にFIFOデータ緩衝器が用いられます。LINデータ緩衝部選択(LINSEL)レジスタの全項目設定後、LINデータ(LINDAT)レジスタへの繰り返しアクセスがデータ読み込みまたはデータ書き込みを実行します(123頁の「データ管理」参照)。

LINDLRのLIN受信データ長(LRXDL3~0)とLIN送信データ長(LTXDL3~0)がデータアクセスに関連しないことに注意してください。

15.4.7. UART指令

LIN制御レジスタ(LINCR)の動作形態(LCMD2)ビットの設定(1)がUART指令を許可します。

- ・ バイト送信とバイト受信の役務は115頁の表15-1.で示されるように独立です。
- ・ バイト転送：UARTが選択されますが、送受信の両役務は禁止されます。
- ・ バイト受信：受信役務が許可されるだけで、送信役務は禁止されます。
- ・ バイト送信：受信役務が許可されるだけで、受信役務は禁止されます。
- ・ 全二重：UARTが許可され、送受信の両役務が許可されます。

この役務の組み合わせはLINCRレジスタの指令(LCMD1,0)ビットによって制御されます(図15-5.参照)。

15.4.7.1. データの扱い

LIN通信に使われるFIFOはUARTアクセス中に禁止されます。それでLINデータ長レジスタ(LINDLR)のLIN受信データ長(LRXDL3~0)とLIN送信データ長(LTXDL3~0)は無関係です。そしてデータレジスタとしてLINデータ(LINDAT)レジスタが使われ、LINデータ緩衝部選択(LINSEL)レジスタは関係ありません。

15.4.7.2. 受信役務

一旦この役務が許可されると、使用者はLINSIRの**受信実行割り込み要求(LRXOK)フラグ**によって到着文字を警告されます。**LINデータ(LINDAT)レジスタ**読み込みはこのフラグを解除(0)して緩衝部の第2段目を開放します。使用者がそれを読むことなく到着文字が無関係と見做すなら、このフラグを直接解除(0)することができます(124頁の「**LINSIR – LIN状態/割り込み要求レジスタ**」で記述される特定フラグの管理をご覧ください)。

受信役務の本質的な構造が2バイトの緩衝器を提供します。1つ目は直列⇒並列変換用に使われ、2つ目は変換結果を受け取ります。LINDATレジスタ読み込みにはこの第2緩衝バイトに至ります。2バイトの緩衝器が満杯の場合、新規到着文字は既に記録された2つ目を上書きします。そして**LIN異常レジスタ(LINERR)のオーバーラン異常(LOVERR)フラグ**が読み込み時にこの文字と連動します。

LINERRレジスタの**フレーミング異常(LFERR)フラグ**はフレーミング異常の場合に設定(1)されます。

15.4.7.3. 送信役務

この役務が許可されると、使用者は**LINデータ(LINDAT)レジスタ**を書くことによって文字を送ります。自動的にLINSIRの**送信実行割り込み要求(LTXOK)フラグ**が解除(0)されます。これは直列送信の最後で立ちます。送るべき新規文字がなければ、LTXOKフラグは独立して(別に)解除(0)することができます(124頁の「**LINSIR – LIN状態/割り込み要求レジスタ**」で記述される特定フラグの管理をご覧ください)。

送信の緩衝はありません。

この役務によって検知される異常はありません。

15.5. LIN/UART説明

15.5.1. リセット

AVRコアのリセット論理回路信号はLIN/UART制御器もリセットします。存在するリセットの別の形式、ソフトウェアリセットは**LIN制御レジスタ(LINCR)のソフトウェアリセット(LSWRES)ビット**によって制御されます。この自己リセットビットは表15-2.で示されるように部分的なリセットを実行します。

表15-2. LIN/UARTレジスタのリセット

レジスタ名称	レジスタ略称	システム リセット値	LSWRES ソフトウェア リセット値	注釈
LIN制御レジスタ	LINCR	0000 0000	0000 0000	
LIN状態/割り込み要求レジスタ	LINSIR	0000 0000	0000 0000	
LIN割り込み許可レジスタ	LINENIR	0000 0000	xxxx 0000	
LIN異常レジスタ	LINERR	0000 0000	0000 0000	
LINビットタイミングレジスタ	LINBTR	0010 0000	0010 0000	x=不定 u=不変
LINボーレート下位レジスタ	LINBRRL	0000 0000	uuuu uuuu	
LINボーレート上位レジスタ	LINBRRH	0000 0000	xxxx uuuu	
LINデータ長レジスタ	LINDLR	0000 0000	0000 0000	
LIN識別子レジスタ	LINIDR	1000 0000	1000 0000	
LINデータ緩衝部選択レジスタ	LINSEL	0000 0000	xxxx 0000	
LINデータレジスタ	LINDAT	0000 0000	0000 0000	

15.5.2. クロック

I/Oクロック信号(clkI/O)はLIN/UART制御器をもクロック駆動します。これはその唯一のクロックです。

15.5.3. LIN規約選択

LIN規約を選択するのに**LIN制御レジスタ(LINCR)のLIN1.3形態(LIN13)ビット**が使われます。

- LIN13=0(既定) : LIN2.1規約
- LIN13=1 : LIN1.3規約

制御器はチェックサム(LIN2.1での拡張チェックサム、LIN1.3での古典的チェックサム)計算でLIN13ビットを調べます。

このビットはUART指令に対して無関係です。

15.5.4. 構成設定

(LINまたはUART)形態に依存して、LIN制御レジスタ(LINCR)の構成設定(LCONF1,0)ビットは制御器を以下の形態(表15-3.)に設定します。

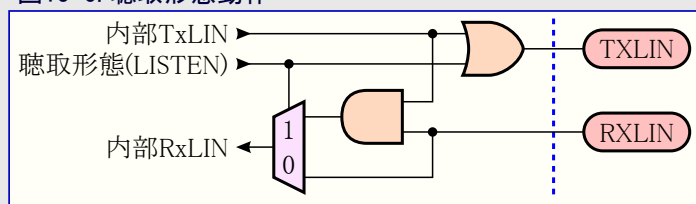
表15-3. 動作形態に対する構成設定表

LCONF1,0	設定形態	
	LIN動作形態	UART動作形態
0 0	LIN標準構成設定 (既定)	8ビット データ, パリティなし, 1停止ビット
0 1	CRC領域の検出または送信なし	8ビット データ, 偶数パリティ, 1停止ビット
1 0	フレーム時間超過監視禁止	8ビット データ, 奇数パリティ, 1停止ビット
1 1	聴取形態動作	聴取形態動作(8ビット データ, パリティなし, 1停止ビット)

LIN構成設定は設定したLIN規約と無関係です。

聴取形態動作は内部TxLINと内部RxLINを共に接続します。この動作形態ではTXLIN出力ピンが禁止され、RXLIN入力は常に許可されます。同じ仕組みはUART形態でも利用可能です。

図15-6. 聴取形態動作

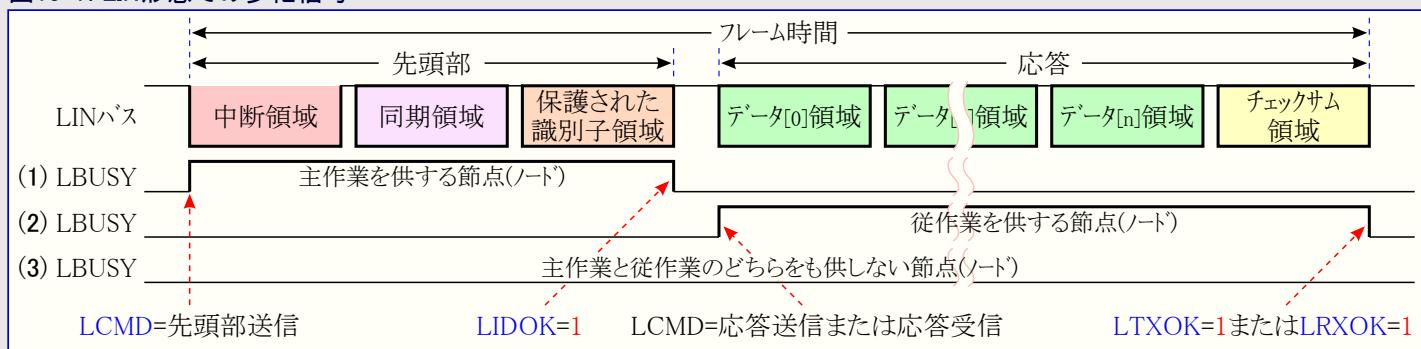


15.5.5. 多忙信号

LIN状態/割り込み要求レジスタ(LINSIR)の多忙信号(LBUSY)フラグは多忙(BUSY)信号の投影です。これはハードウェアによって設定(1)と解除(0)が成されます。これは制御器がLINまたはUART通信で多忙なことを合図します。

15.5.5.1. LIN形態での多忙信号

図15-7. LIN形態での多忙信号



多忙信号が設定(1)されると、いくつかのレジスタが施錠され、使用者書き込みが許されません。

- LIN制御レジスタ(LINCR) - 形態/指令(LCMD2~0)、許可(LENA)、ソフトウェアリセット(LSWRES)を除く
- LINボーレートレジスタ(LINBRRH, LINBRRL)
- LINデータ長レジスタ(LINDLR)
- LIN識別子レジスタ(LINIDR)
- LINデータレジスタ(LINDAT)

多忙信号が設定(1)されている場合に利用可能な指令は以下だけです。

- LCMD1,0=00、中止指令はバイトの最後で処理されます。
- LENA=0やLCMD2=0、禁止指示は直ちに処理されます。
- LSWRES=0、リセット指示は直ちに処理されます。

多忙信号(=1)中に別の指令が許可された場合に新しい指令は有効にされず、LIN異常レジスタ(LINERR)のオーバーラン異常(LOVERR)フラグが設定(1)されることに注意してください。進行中の転送は割り込まれません。

15.5.5.2. UART形態での多忙信号

バイト送信中の間に多忙信号が設定(1)されます。これは書き込みからいくつかのレジスタを施錠します。

- LIN制御レジスタ(LINCR) - 形態/指令(LCMD2~0)、許可(LENA)、ソフトウェアリセット(LSWRES)を除く
- LINデータレジスタ(LINDAT)

バイト受信中に多忙信号は生成されません。

15.5.6. ビット タイミング

15.5.6.1. ボーレート生成器

ボーレートは秒当たりのビット数(bps)での転送速度を定義します。

- BAUD : ボーレート(bps)
- $\text{clk}_{I/O}$: システムI/Oクロック周波数
- LDIV11~0 : LINボーレートレジスタ(LINBRRH, LINBRRL)の内容(0~4095)。入力クロックとして $\text{clk}_{I/O}$ を受け取る前置分周器。
- LBT5~0 : LINビット タイミング レジスタ(LINBTR)の下位側(0~63)はLINまたはUARTでの採取番号です(既定値=32)。

ボーレートを計算するための式は次です。

$$\text{BAUD} = \text{clk}_{I/O} \div (\text{LBT5} \sim 0 \times (\text{LDIV11} \sim 0 + 1))$$

LIN分周値(LDIV11~0)設定用の式は次です。

$$\text{LDIV11} \sim 0 = (\text{clk}_{I/O} \div (\text{LBT5} \sim 0 \times \text{BAUD})) - 1$$

受信に於いて3採取の多数決が行われることに留意してください。

15.5.6.2. LIN形態での再同期

先頭部受信待機時、LINBTRのLINビット タイミング(LBT5~0)は32です。再同期は中断(BREAK)検出時に始まります。中断(BREAK)量が範囲(最小=11、標準=13、最大28ビット)内でなければ、中断(BREAK)は拒絶されます。再同期は受信した先頭部の同期(SYNC)領域(\$55)に対してLBT5~0を補正することによって行われます。その後、新規LBT5~0値を用いて保護された識別子(PROTECTED IDENTIFIER)が採取されます。この制御器に実装された再同期は±20%のクロック偏差を許容し、ボーレートを±2%範囲に補正します。

新しいLBT5~0値は応答の最後まで使われます。そしてLBT5~0は次の先頭部のために32にリセットされます。

LINBTRレジスタはクロック発振器の(ソフトウェア)再校正に用いることができます。

LIN節点(ノード)が主装置として許可されている場合に再同期は実行されません。

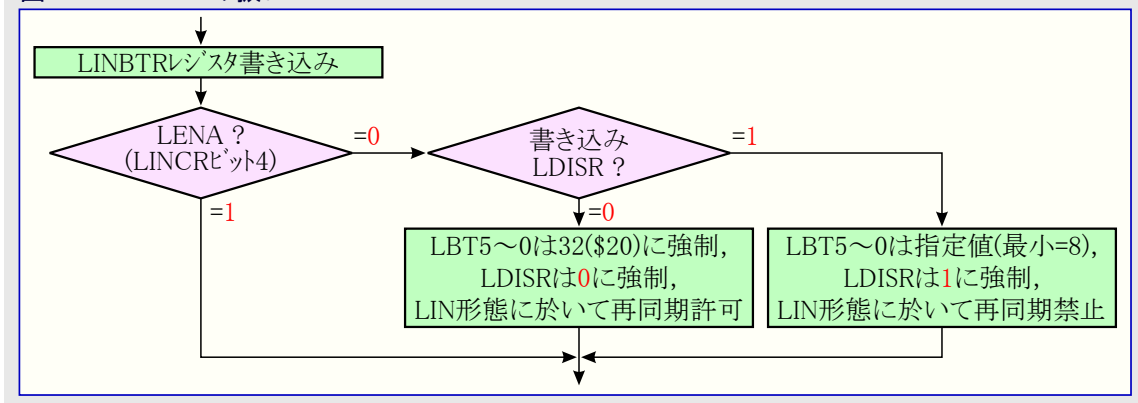
15.5.6.3. LBT5~0の扱い

LINBTRレジスタのビット タイミング再同期禁止(LDISR)ビットは以下のために使われます。

- 試験目的のためにLIN従形態で再同期を禁止。
- LBT5~0の設定を許可するため(特にUART形態の場合でボーレートを手動補正するため)。

この処理のためにLIN制御レジスタ(LINCR)の許可(LENA)ビットが重要であることに注意してください(図15-8をご覧ください)。

図15-8. LBT5~0の扱い



15.5.7. データ長

115頁の「15.4.6. LIN指令」項は応答送受信に先立つ、LINデータ長レジスタ(LINDLR)のLIN受信データ長(LRXDL3~0)またはLIN送信データ長(LTXDL3~0)領域を自動的にどう設定するか、または設定方法を記述します。

応答送信の場合のLRXDL3~0は既に成功裏に送られたバイト数を数えるため、ハードウェアによって使われます。

応答受信の場合のLTXDL3~0は既に成功裏に受け取ったバイト数を数えるため、ハードウェアによって使われます。

異常が起きた場合、この情報はLINメッセージを回復するためのプログラム作成者に有用です。

15.5.7.1. LIN2.1でのデータ長

- LTXDL3~0=0なら、チェックサム(CHECKSUM)だけが送られました。
- LRXDL3~0=0なら、最初に受信したバイトがチェックサム(CHECKSUM)として解釈されました。
- LTXDL3~0またはLRXDL3~0が>8なら、その値は指令設定後で最初のバイトの送信または受信に先立って8に強制されます。

15.5.7.2. LIN1.3でのデータ長

- LRXDLとLTXDLの両領域は受信した保護された識別子(PROTECTED IDENTIFIER)に含まれるデータ長符号の復号により、識別子割り込み要求(LIDOK)フラグを設定(1)する前にハードウェアで更新されます(LRXDL=LTXDL)。
- 上の構造經由では、0または>8の長さは有り得ません。

15.5.8. xxOKフラグ

LINSIRレジスタには3つのxxOKフラグがあります。

- LIN識別子OK(LIDOK)

先頭部送信または先頭部受信のどちらかにより、先頭部の終りで設定(1)されます。LIN1.3ではLIDOK生成前に制御器がLINデータ長レジスタ(LINDLR)のLIN受信データ長(LRXDL)とLIN送信データ長(LTXDL)を更新します。

- LIN応答受信完了(LRXOK)

LIN形態で応答受信機能による応答の終りで、UART形態で一旦文字が受信されると設定(1)されます。

- LIN応答送信完了(LTXOK)

LIN形態で応答送信機能による応答の終りで、UART形態で一旦文字が送信されてしまうと設定(1)されます。

これらのフラグは対応する割り込み許可ビットがLIN割り込み許可レジスタ(LINENIR)で設定(1)されているなら、割り込みを生成することができます(118頁の「15.5.13. 割り込み」項をご覧ください)。

15.5.9. xxERRフラグ

LIN状態/割り込み要求レジスタ(LINSIR)の異常割り込み要求(LERR)フラグはLIN異常レジスタ(LINERR)の全ビットの論理和(OR)です(122頁の「15.5.13. 割り込み」項をご覧ください)。

- LINビット異常(LBERR:LIN Bit ERRor)

バス上にビットを送出する装置はバスの監視も行います。LINビット異常は監視されたビット値が送出したビット値と異なる時にフラグを立てられます。LINビット異常検出後の送信は中止されます。

- LINチェックサム異常(LCERR:LIN Checksum ERRor)

LINチェックサム異常は全ての受信データバイト(LIN2.1ではそれと保護された識別子)の総和の256の剰余の反転とチェックサムの加算が\$FFの結果でない場合にフラグを立てられます。

- LINパリティ異常(LPERR:LIN Parity ERRor) (識別子)

識別子(IDENTIFIER)領域内のLINパリティ異常はパリティビットの値が識別子値と一致しない場合にフラグを立てられます(「15.6.8. LIN IDR – LIN識別子レジスタ」項をご覧ください)。LIN従装置応用は不正なパリティビットと不正な識別子を区別しません。ハードウェアはどんな修正の義務も負いません。けれども、LIN従装置応用は以下としてこれを解決しなければなりません。

- (パリティビットが不正な)既知の識別子、または
- 無視されるべき不正な識別子、または
- 新しい識別子

- LIN同期異常(LSERR:LIN Synchronization ERRor)

LIN同期異常は従装置が与えられた許容外で同期(SYNC)領域のエッジを検出した場合にフラグを立てられます。

- LINフレーミング異常(LFERR:LIN Framing ERRor)

フレーミング異常は優性の停止(STOP)ビットが採取された場合にフラグを立てられます。UART形態と同じ機能です。

- LINフレーム時間超過異常(LTOERR:LIN Time Out ERRor)

時間超過異常は同期(SYNC)と識別子(IDENTIFIER)の領域の送信によって、どれかの従作業によるメッセージ フレームが最大長(T_{Frame_Maximum})内で完全に完了されなかった場合にフラグを立てられます(122頁の「15.5.10. フレーム時間超過」項をご覧ください)。

- LINオーバーラン異常(LOVERR:LIN OVerrun ERRor)

オーバーラン異常は'多忙信号'が存在中に(LIN中断以外の)新規指令が入力された場合にフラグを立てられます。UART形態でのオーバーラン異常は受信したバイトが直列入力緩衝器に格納されたバイトを上書きした場合にフラグを立てられます。

- LIN中止(LABORT)

LIN転送中止は'多忙信号'が存在中の以前のLIN中止指令(LCMD2~0=000)を反映します。

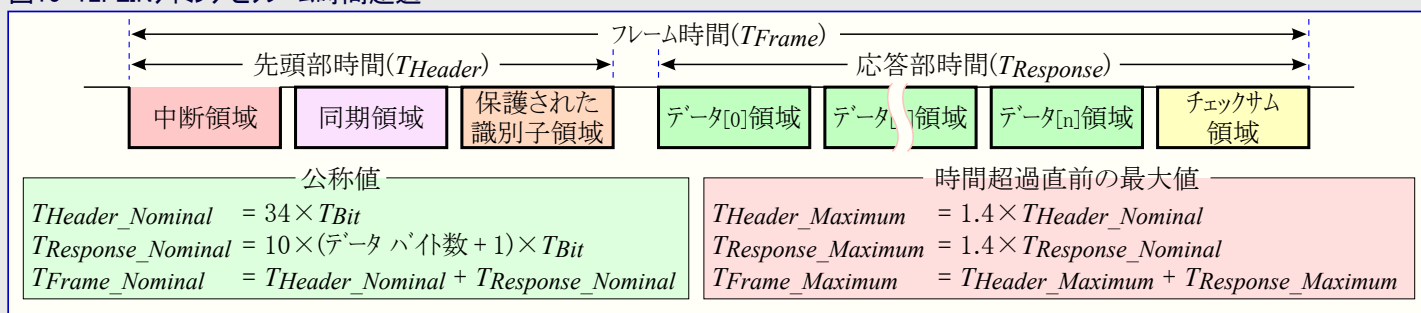
各LIN異常後、LIN制御器はその直前の活動を停止し、120頁の図15-11.で図解されるように撤回形態(LCMD2~0=000)に戻ります。

LINSIRレジスタのLERRへの書き込みはLERRビットとLINERRレジスタの全ビットをリセットします。

15.5.10. フレーム時間超過

LIN規約に従い、フレーム時間超過は $T_{Frame} > T_{Frame_Maximum}$ の場合にフラグを立てられます。この機能はLIN/UART制御器で実装されます。

図15-12. LINタイミングとフレーム時間超過



15.5.11. データ内での中断

LIN規約に従い、LIN/UART制御器は例えば応答のバイトで中断(BREAK)が部分的に重なられても、中断/同期(BREAK/SYNC)領域手順を検出することができます。中断/同期(BREAK/SYNC)領域手順が起ると、進行中の転送は中止され、新規フレームの処理が始まります。

- ・ 従節点(ノード)では、異常が生成されます(換言すると、**応答送信**の場合に**ビット異常(LBERR)**、**応答受信**の場合に**フレーミング異常(LFERR)**)。データ異常の情報も利用可能で、120頁の「**異常後のデータ長**」を参照してください。
- ・ 主節点(ノード)では、このフレームの中止に関して使用者(コード)が責任を持ちます。これを行うため、主作業は最初に進行中の通信を中止(LCMD2〜0ビットを解除(=000:LIN中止指令))し、そして**先頭部送信**指令を与えます。この場合、**中止異常(LABORT)**フラグが設定(1)されます。

従節点では、LIN/UART制御器が(中止した)応答を処理していた時に利用可能だった同期設定で中断(BREAK)検出が処理されます。けれども例の如く再同期が再始動し、時間超過値が僅かに不正確となり得ます。

15.5.12. チェックサム

フレームの最後の領域がチェックサムです。

LIN2.1に於けるチェックサムは保護された識別子と全データバイトに渡る桁溢れを伴う総和の論理反転8ビットを含みます。この計算は拡張チェックサムと呼ばれます。

$$\text{チェックサム} = 255 - \left(\text{unsigned char} \left(\sum_{n=0}^n \text{データ}_n \right) + \text{保護された識別子} \right) + \text{unsigned char} \left(\left(\sum_{n=0}^n \text{データ}_n \right) + \text{保護された識別子} \right) \gg 8$$

LIN1.3に於けるチェックサムは全データバイトに渡る桁溢れを伴う総和の論理反転8ビットを含みます。この計算は古典的チェックサムと呼ばれます。

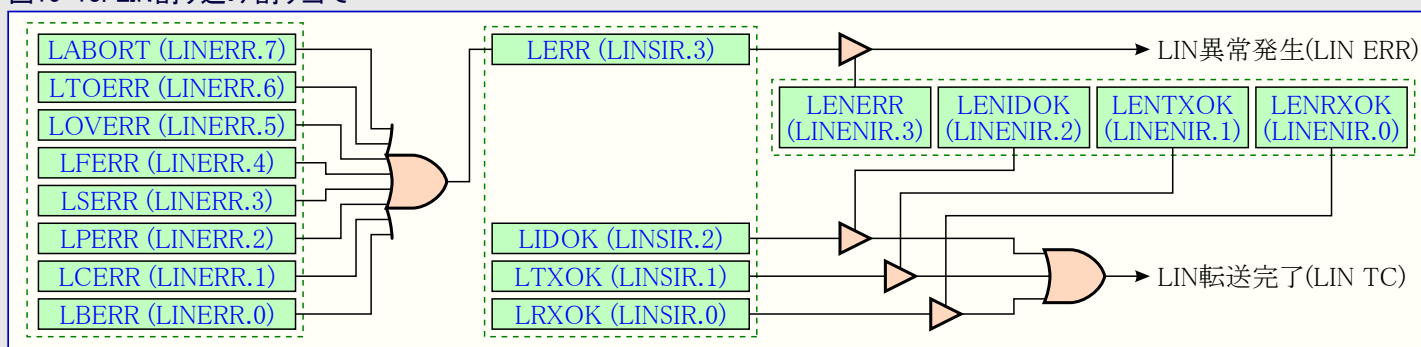
$$\text{チェックサム} = 255 - \left(\text{unsigned char} \left(\sum_{n=0}^n \text{データ}_n \right) + \text{unsigned char} \left(\sum_{n=0}^n \text{データ}_n \right) \gg 8 \right)$$

60(\$3C)〜63(\$3F)のフレーム識別子は常に古典的チェックサムを使います。

15.5.13. 割り込み

図15-13.で示されるように、LIN状態/割り込み要求レジスタ(LINSIR)の4つの通信フラグが2つ割り込みを駆動するように組み合わせられます。これらのフラグの各々はLIN割り込み許可レジスタ(LINENIR)に各々の割り込み許可ビットを持ちます(121頁の「15.5.8. xxOKフラグ」と「15.5.9. xxERRフラグ」をご覧ください)。

図15-13. LIN割り込み割り当て



15.5.14. メッセージ濾過

全ての識別子に基づくメッセージ濾過は実装されていません。識別子として\$3C,\$3D,\$3E,\$3Fを持つフレーム先頭部だけがLIN状態/割り込み要求レジスタ(LINSIR)で利用可能です。

60(\$3C)~63(\$3F)の識別子を持つメッセージは古典的チェックサム(データバイトだけに渡る総和)を用いることと、LIN規約は言っています。ソフトウェアには、この期待したチェックサムを供給/検査するため、LIN1.3形態(LIN13)ビットを正確に切り替える(応答送信または応答受信の指令直後に、CRCの計算に識別子領域の挿入を含めるか否かの)責任があります。

表15-4. フレーム情報

LIDST2~0	フレーム情報
0 x x	特定識別子なし
1 0 0	60(\$3C)識別子
1 0 1	61(\$3D)識別子
1 1 0	62(\$3E)識別子
1 1 1	63(\$3F)識別子

15.5.15. データ管理

15.5.15.1. LIN FIFOデータ緩衝器

レジスタ配置位置を保つため、LINデータ緩衝器は(アドレス指示器でアクセス可能な)FIFOのように見えます。このFIFOはLINデータ緩衝器選択(LINSEL)レジスタのFIFO LINデータ緩衝器指示子(LINDX2~0)領域経由でLINデータレジスタ(LINDAT)を通してアクセスされます。

データ指示子(LINDX2~0)は必要とするデータバイトへの位置指示子です。データバイトは読み書きできます。データ緩衝器指示子自動進行(LAINC,Low活性)が解除(0)されている場合、データ指示子は各LINDATアクセス後、自動的に増加されます。超過巻き戻りが実装され、データ指示子=7後、それはデータ指示子=0になります。さもなければ、またはLAINCビットが設定(1)されている場合、各LINDATアクセス前にデータ指示子は書き込み(更新)が必要です。

LINフレームの先頭バイトはデータ指示子=0に、2つ目はデータ指示子=1に、以下同様に格納されます。にも拘らず、LINSELは使用前に使用者によって初期化されなければなりません。

15.5.15.2. UARTデータレジスタ

LINDATレジスタがデータレジスタです(緩衝なし、FIFOなし)。書き込みアクセスではLINDATがデータ出力用で、読み込みアクセスではLINDATがデータ入力用です。

UART形態に於けるLINSELレジスタは未使用です。

15.5.16. OCD支援

デバッグの中断(ブレーク)が起こると、LIN/UART制御器の状態機構が(フレーム時間超過を含めて)停止され、更なる通信が不正になるかもしれません。

15.6. LIN/UART用レジスタ

表15-5. LIN/UARTレジスタビット要約

名称	ビット7		ビット6		ビット5		ビット4		ビット3		ビット2		ビット1		ビット0	
LINCR	LSWRES		LIN13		LCONF1		LCONF0		LENA		LCMD2		LCMD1		LCMD0	
	0	R/W	0	R/W	0	R/W	0	R/W	0	R/W	0	R/W	0	R/W	0	R/W
LINSIR	LIDST2		LIDST1		LIDST0		LBUSY		LERR		LIDOK		LTXOK		LRXOK	
	0	R	0	R	0	R	0	R	0	R/W1	0	R/W1	0	R/W1	0	R/W1
LINENIR	-		-		-		-		LENERR		LENIDOK		LENTXOK		LENRXOK	
	0	R	0	R	0	R	0	R	0	R/W	0	R/W	0	R/W	0	R/W
LINERR	LABORT		LTOERR		LOVERR		LFERR		LSERR		LPERR		LCERR		LBERR	
	0	R	0	R	0	R	0	R	0	R	0	R	0	R	0	R
LINBTR	LDIRS		-		LBT5		LBT4		LBT3		LBT2		LBT1		LBT0	
	0	R/W	0	R	1	R/(W)	0	R/(W)	0	R/(W)	0	R/(W)	0	R/(W)	0	R/(W)
LINBRL	LDIV7		LDIV6		LDIV5		LDIV4		LDIV3		LDIV2		LDIV1		LDIV0	
	0	R/W	0	R/W	0	R/W	0	R/W	0	R/W	0	R/W	0	R/W	0	R/W
LINBRRH	-		-		-		-		LDIV11		LDIV10		LDIV9		LDIV8	
	0	R	0	R	0	R	0	R	0	R/W	0	R/W	0	R/W	0	R/W
LINDLR	LTXDL3		LTXDL2		LTXDL1		LTXDL0		LRXDL3		LRXDL2		LRXDL1		LRXDL0	
	0	R/W	0	R/W	0	R/W	0	R/W	0	R/W	0	R/W	0	R/W	0	R/W
LINIDR	LP1		LP0		LID5/LDL1		LID4/LDL0		LID3		LID2		LID1		LID0	
	1	R	0	R	0	R/W	0	R/W	0	R/W	0	R/W	0	R/W	0	R/W
LINSEL	-		-		-		-		LAINC		LINDX2		LINDX1		LINDX0	
	0	R	0	R	0	R	0	R	0	R/W	0	R/W	0	R/W	0	R/W
LINDAT	LDATA7		LDATA6		LDATA5		LDATA4		LDATA3		LDATA2		LDATA1		LDATA0	
	0	R/W	0	R/W	0	R/W	0	R/W	0	R/W	0	R/W	0	R/W	0	R/W

15.6.1. LINCR – LIN制御レジスタ (LIN Control Register)

ビット (\$C8)	7	6	5	4	3	2	1	0	
	LSWRES	LIN13	LCONF1	LCONF0	LENA	LCMD2	LCMD1	LCMD0	LINCR
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 – LSWRES : ソフトウェア リセット (Software Reset)

- 0=動きなし
- 1=ソフトウェア リセット(このビットはリセット手順の最後で自己リセット(0)されます。)

● ビット6 – LIN13 : LIN1.3形態 (LIN1.3 mode)

- 0=LIN2.1 (既定)
- 1=LIN1.3

● ビット5,4 – LCONF1,0 : 構成設定 (Configuration)

a. LIN動作形態 (既定=00)

- 00=LIN標準形態 (聴取動作形態-OFF、CRC-ON、フレーム時間監視-ON)
- 01=CRCなし、時間監視なし (聴取動作形態-OFF)
- 10=フレーム時間監視なし (聴取動作形態-OFF、CRC-ON)
- 11=聴取動作形態 (CRC-ON、フレーム時間監視-ON)

b. UART動作形態 (既定=00)

- 00=8ビット データ、パリティなし (聴取動作形態-OFF)
- 01=8ビット データ、偶数パリティ (聴取動作形態-OFF)
- 10=8ビット データ、奇数パリティ (聴取動作形態-OFF)
- 11=聴取動作形態、8ビット データ、パリティなし

● ビット3 – LENA : 許可 (Enable)

- 0=(LINとUARTの両形態)禁止
- 1=(LINとUARTの両形態)許可

● ビット2~0 – LCMD2~0 : 形態と指令 (Command and mode)

指令は許可(LENA)が設定(1)されている場合にだけ利用可能です。

- 000=LIN先頭部受信 – LIN中止
- 001=LIN先頭部送信
- 010=LIN応答受信
- 011=LIN応答送信
- 100=UARTバイト転送
- 1x1=UARTバイト送信
- 11x=UARTバイト受信

15.6.2. LINSIR – LIN状態/割り込み要求レジスタ (LIN Status and Interrupt Register)

ビット (\$C9)	7	6	5	4	3	2	1	0	
	LIDST2	LIDST1	LIDST0	LBUSY	LERR	LIDOK	LTXOK	LRXOK	LINSIR
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7~5 – LIDST2~0 : 識別子状態 (Identifier Status)

- 0xx=特定識別子以外
- 100=60(\$3C)識別子
- 101=61(\$3D)識別子
- 110=62(\$3E)識別子
- 111=63(\$3F)識別子

● ビット4 – LBUSY : 多忙信号 (Busy Signal)

- 0=多忙でない
- 1=(送信または受信で)多忙

● ビット3 – LERR : 異常割り込み要求フラグ (Error Interrupt)

- ・ 0=異常なし
- ・ 1=異常発生

この割り込みをリセットするため、使用者は1を書くことによってこのビットを解除(0)します。LERRのリセットはLIN異常レジスタ(LINERR)の全ビットもリセットします。

UART形態では、LINデータ(LINDAT)レジスタを読むことによってこのビットが解除(0)されます。

● ビット2 – LIDOK : 識別子割り込み要求フラグ (Identifier Interrupt)

このビットは対応する許可(LENIDOK)ビットがLINENIRレジスタで設定(1)されている場合に割り込みを生成します。

- ・ 0=識別子なし
- ・ 1=従作業 : 識別子存在、主作業 : 先頭部送信完了

この割り込みをリセットするため、使用者は1を書くことによってこのビットを解除(0)します。

● ビット1 – LTXOK : 送信実行割り込み要求フラグ (Transmit Performed Interrupt)

このビットは対応する許可(LENTXOK)ビットがLINENIRレジスタで設定(1)されている場合に割り込みを生成します。

- ・ 0=送信なし
- ・ 1=応答送信完了

この割り込みをリセットするため、使用者は1を書くことによってこのビットを解除(0)します。

UART形態では、LINDATレジスタに書くことによってこのビットが解除(0)されます。

● ビット0 – LRXOK : 受信実行割り込み要求フラグ (Receive Performed Interrupt)

このビットは対応する許可(LENRXOK)ビットがLIN割り込み許可レジスタ(LINENIR)で設定(1)されている場合に割り込みを生成します。

- ・ 0=受信なし
- ・ 1=応答受信完了

この割り込みをリセットするため、使用者は1を書くことによってこのビットを解除(0)します。

UART形態では、LINデータ(LINDAT)レジスタを読むことによってこのビットが解除(0)されます。

15.6.3. LINENIR – LIN割り込み許可レジスタ (LIN Enable Interrupt Register)

ビット (\$CA)	7	6	5	4	3	2	1	0	
	–	–	–	–	LENERR	LENIDOK	LENTXOK	LENRXOK	LINENIR
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7～5 – Res : 予約 (Reserved)

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、LINENIRが書かれる時にこれらは0を書かれなければなりません。

● ビット3 – LENERR : 異常割り込み許可 (Enable Error Interrupt)

- ・ 0=異常割り込み遮蔽
- ・ 1=異常割り込み許可

● ビット2 – LENIDOK : 識別子割り込み要求フラグ (Enable Identifier Interrupt)

- ・ 0=識別子割り込み遮蔽
- ・ 1=識別子割り込み許可

● ビット1 – LENTXOK : 送信実行割り込み要求フラグ (Enable Transmit Performed Interrupt)

- ・ 0=送信実行割り込み遮蔽
- ・ 1=送信実行割り込み許可

● ビット0 – LENRXOK : 受信実行割り込み要求フラグ (Enable Receive Performed Interrupt)

- ・ 0=受信実行割り込み遮蔽
- ・ 1=受信実行割り込み許可

15.6.4. LINERR – LIN異常レジスタ (LIN Error Register)

ビット (\$CB)	7	6	5	4	3	2	1	0	
	LABORT	LTOERR	LOVERR	LFERR	LSERR	LPERR	LCERR	LBERR	LINERR
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 – LABORT : 中止フラグ (Abort Flag)

- ・ 0=警告なし
- ・ 1=LIN中止指令発生

このビットはLINSIRレジスタのLERRフラグが解除(0)される時に解除(0)されます。

● ビット6 – LTOERR : フレーム時間超過異常フラグ (Frame_Time_Out Error Flag)

- ・ 0=異常なし
- ・ 1=フレーム時間超過異常

このビットはLINSIRレジスタのLERRフラグが解除(0)される時に解除(0)されます。

● ビット5 – LOVERR : オーバーラン異常フラグ (Overrun Error Flag)

- ・ 0=異常なし
- ・ 1=オーバーラン異常

このビットはLINSIRレジスタのLERRフラグが解除(0)される時に解除(0)されます。

● ビット4 – LFERR : フレーミング異常フラグ (Framing Error Flag)

- ・ 0=異常なし
- ・ 1=フレーミング異常

このビットはLIN状態/割り込み要求レジスタ(LINSIR)の異常割り込み要求(LERR)フラグが解除(0)される時に解除(0)されます。

● ビット3 – LSERR : 同期異常フラグ (Synchronization Error Flag)

- ・ 0=異常なし
- ・ 1=同期異常

このビットはLINSIRレジスタのLERRフラグが解除(0)される時に解除(0)されます。

● ビット2 – LPERR : パリティ異常フラグ (Parity Error Flag)

- ・ 0=異常なし
- ・ 1=パリティ異常

このビットはLINSIRレジスタのLERRフラグが解除(0)される時に解除(0)されます。

● ビット1 – LCERR : チェックサム異常フラグ (Checksum Error Flag)

- ・ 0=異常なし
- ・ 1=チェックサム異常

このビットはLINSIRレジスタのLERRフラグが解除(0)される時に解除(0)されます。

● ビット0 – LBERR : ビット異常フラグ (Bit Error Flag)

- ・ 0=異常なし
- ・ 1=ビット異常

このビットはLIN状態/割り込み要求レジスタ(LINSIR)の異常割り込み要求(LERR)フラグが解除(0)される時に解除(0)されます。

15.6.5. LINBTR – LINビット タイミング レジスタ (LIN Bit Timing Register)

ビット (\$CC)	7	6	5	4	3	2	1	0	
	LDISR	–	LBT5	LBT4	LBT3	LBT2	LBT1	LBT0	LINBTR
Read/Write	R/W	R	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 – LDISR : ビット タイミング再同期禁止 (Disable Bit Timing Re-synchronization)

- ・ 0=ビット タイミング再同期許可 (既定)
- ・ 1=ビット タイミング再同期禁止

● ビット5~0 – LBT5~0 : LINビット タイミング (LIN Bit Timing)

ビットの採取数(位置)を与えます。

採取時間=(1÷fclkI/O)×(LDIV11~0 + 1)

既定値:LBT5~0=32、最小値:LBT5~0=8、最大値:LBT5~0=63

15.6.6. LINBRRH, LINBRL (LINBRR) – LINボーレートレジスタ (LIN Baud Rate Register)

ビット (\$CE)	15	14	13	12	11	10	9	8	
	–	–	–	–	LDIV11	LDIV10	LDIV9	LDIV8	LINBRRH
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 (\$CD)	7	6	5	4	3	2	1	0	
	LDIV7	LDIV6	LDIV5	LDIV4	LDIV3	LDIV2	LDIV1	LDIV0	LINBRL
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)

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、LINBRRHが書かれる時にこれらは0を書かれなければならないかもしれません。

● ビット11～0 – LDIV11～0 : 前置分周係数 (Scaling of clkI/O Frequency)

LDIV値は適切なLINまたはUARTのボーレートを達成するために入力するclkI/O周波数を分周するのに使われます。

15.6.7. LINDLR – LINデータ長レジスタ (LIN Data Length Register)

ビット (\$CF)	7	6	5	4	3	2	1	0	
	LTXDL3	LTXDL2	LTXDL1	LTXDL0	LRXDL3	LRXDL2	LRXDL1	LRXDL0	LINDLR
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7～4 – LTXDL3～0 : 送信データ長 (LIN Transmit Data Length)

LIN形態でのこの領域は送信されるべきバイト数を与えます(最大8で留め置かれます)。

UART形態でのこの領域は未使用です。

● ビット3～0 – LRXDL3～0 : 受信データ長 (LIN Receive Data Length)

LIN形態でのこの領域は受信されるべきバイト数を与えます(最大8で留め置かれます)。

UART形態でのこの領域は未使用です。

15.6.8. LINIDR – LIN識別子レジスタ (LIN Identifier Register)

ビット (\$D0)	7	6	5	4	3	2	1	0	
	LP1	LP0	LID5/LDL1	LID4/LDL0	LID3	LID2	LID1	LID0	LINIDR
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

● ビット7,6 – LP1,0 : パリティ (Parity)

LIN形態では、 $LP1 = \neg(LID5 \wedge LID4 \wedge LID3 \wedge LID1)$ 、 $LP0 = LID4 \wedge LID2 \wedge LID1 \wedge LID0$

UART形態でのこの領域は未使用です。

● ビット5,4 – LDL1,0 : LIN1.3データ長 (LIN1.3 Data Length)

LIN形態では次のとおりです。

- ・ 00=2バイト応答
- ・ 01=2バイト応答
- ・ 10=4バイト応答
- ・ 11=8バイト応答

UART形態でのこの領域は未使用です。

● ビット3～0 – LID3～0 : LIN1.3識別子 (LIN1.3 Identifier)

LIN1.3形態：4ビット識別子

UART形態でのこの領域は未使用です。

● ビット5～0 – LID5～0 : LIN2.1識別子 (LIN2.1 Identifier)

LIN2.1形態：6ビット識別子(長さは転送されません。)

UART形態でのこの領域は未使用です。

15.6.9. LINSEL – LINデータ緩衝器選択レジスタ (LIN Data Buffer Selection Register)

ビット (\$D1)	7	6	5	4	3	2	1	0	
	–	–	–	–	LAINC	LINDX2	LINDX1	LINDX0	LINSEL
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
初期値	–	–	–	–	0	0	0	0	

● ビット7～4 – Res : 予約 (Reserved)

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、LINSELが書かれる時にこれらは0を書かれなければならない。

● ビット3 – LAINC : データ緩衝器指示子の自動進行 (Auto Increment of Data Buffer Index)

LIN形態では次のとおりです。

- ・ 0=FIFOデータ緩衝器位置指示子の自動進行 (既定)
- ・ 1=自動進行なし

UART形態でのこのビットは未使用です。

● ビット2～0 – LINDX2～0 : FIFO LINデータ緩衝器位置指示子 (FIFO LIN Data Buffer Index)

LIN形態では、FIFOデータ緩衝器ないのLIN応答データ バイトの位置(指示子)です。FIFOデータ緩衝器はLINデータ(LINDAT)レジスタを通してアクセスされます。

UART形態でのこの領域は未使用です。

15.6.10. LINDAT – LINデータ レジスタ (LIN Data Register)

ビット (\$D2)	7	6	5	4	3	2	1	0	
	LDATA7	LDATA6	LDATA5	LDATA4	LDATA3	LDATA2	LDATA1	LDATA0	LINDAT
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 – LDATA7～0 : LINデータ入出力 (LIN Data in / Data out)

LIN形態では、FIFOデータ緩衝器のポートです。

UART形態では、データ レジスタ(データ緩衝器なし、FIFOなし)です。

- ・ 書き込みアクセスはデータ出力
- ・ 読み込みアクセスはデータ入力

16. 電流源 (ISRC)

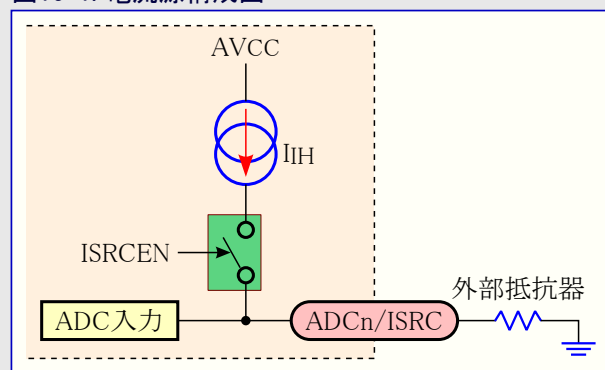
16.1. 特徴

- 100 μ A定電流源
- $\pm 10\%$ 絶対精度

ATtiny87/167は100 μ A $\pm 10\%$ の電流源が特徴です。要求で、電流は外部抵抗器を通して流れます。その電圧はA/D変換器と共用する専用ピンで測定することができます。誤差0.5%以下の直列抵抗器の使用が推奨されます。大きな値に対してデバイスを保護するため、A/D変換器(ADC)は最初の測定を実行するのに内部基準電圧としてAVCCで構成設定されなければなりません。その後、結果を改善するために直前の測定値に従って他の内部基準電圧を選ぶことができます。

アナログ雑多制御レジスタ(AMISCR)の電流源許可(ISRCEN)ビットが設定(1)されると、ISRCピンは100 μ Aを吐き出します。さもなければ、このピンは初期の機能を維持します。

図16-1. 電流源構成図



16.2. 代表的な応用

16.2.1. LIN電流源

集合内のLIN節点(ノード)の構成設定中に、集合の全ての節点に対して動的に固有の物理アドレスにする必要があるかもしれません。それを行う方法はLIN規約に記述されていません。

電流源はLIN節点によって支援される応用に対して物理アドレスを関連付けする素晴らしい解決策を提供します。集合内のLIN節点構成設定に完全な動的節点構成設定を使うことができます。

ATtiny87/167は電流減と共に使われる外部抵抗器を持つことを提唱します。デバイスはA/D変換器経由で抵抗の境界に対する電圧を測定します。結果の電圧は節点がLIN通信に参加する時に使う通信処理部の物理アドレスを定義します。

ATtiny87/167の内部電流源解決策はどんな種類の電圧変動に対してもアドレス検出を免疫性にします。

表16-1. 8アドレス系用抵抗値($\pm 5\%$)例 (AVCC=5V(注))

物理アドレス	抵抗値 (Rload)	代表測定電圧 (V)	2.56V基準での読み込み値		
			最小	代表	最大
0	1k Ω	0.1		40	
1	2.2k Ω	0.22		88	
2	3.3k Ω	0.33		132	
3	4.7k Ω	0.47		188	
4	6.8k Ω	0.68		272	
5	10k Ω	1		400	
6	15k Ω	1.5		600	
7	22k Ω	2.2		880	

注: 5V範囲での最大抵抗値=30k Ω
3V範囲での最大抵抗値=15k Ω

表16-2. 16アドレス系用抵抗値($\pm 5\%$)例 (AVCC=5V(注))

物理アドレス	抵抗値 (Rload)	代表測定電圧 (V)	2.56V基準での読み込み値		
			最小	代表	最大
0	1k Ω	0.1	38	40	45
1	1.2k Ω	0.12	46	48	54
2	1.5k Ω	0.15	57	60	68
3	1.8k Ω	0.18	69	72	81
4	2.2k Ω	0.22	84	88	99
5	2.7k Ω	0.27	104	108	122
6	3.3k Ω	0.33	127	132	149
7	4.7k Ω	0.47	181	188	212
8	6.8k Ω	0.68	262	272	306
9	8.2k Ω	0.82	316	328	369
10	10k Ω	1.0	386	400	450
11	12k Ω	1.2	463	480	540
12	15k Ω	1.5	579	600	675
13	18k Ω	1.8	694	720	810
14	22k Ω	2.2	849	880	989
15	27k Ω	2.7	1023	1023	1023

注: 5V範囲での最大抵抗値=30k Ω
3V範囲での最大抵抗値=15k Ω

16.2.2. 低費用変換器

可変抵抗器に基づく外部変換器を電流源に接続することができます。これは例えば以下のようなことができます。

- ・ 温度感知器として使われるサーミスタまたは温度感知性抵抗器
- ・ 光感知器として使われるCdS光導電セルまたは光感知性抵抗器
- ・ ～

この形式の変換器とでの電流源使用は追加部品を無くし、さもなければ抵抗網またはホイートストーンブリッジが必要とされます。

16.2.3. 外部デバイス用電圧基準

電流源と共に使われる外部抵抗器は外部デバイス用の電圧基準として使うことができます。電流源精度(2%)よりも低い誤差の直列抵抗器の使用が推奨されます。表16-2は標準抵抗値を用いた電圧基準の例を与えます。

16.2.4. 内部アナログ比較器用閾値基準

電流源と共に使われる外部抵抗器は内部アナログ比較器(131頁の「[アナログ比較器 \(AC\)](#)」をご覧ください。)用の電圧基準として使うことができます。これはAIN1(アナログ比較器非反転入力ピン)だけでなく、AIN0(アナログ比較器反転入力ピン)にも接続することができます。電流源精度(2%)よりも低い誤差の直列抵抗器の使用が推奨されます。表16-2は標準抵抗値を用いた電圧基準の例を与えます。

16.3. 電流源用レジスタ

16.3.1. AMISCR – アナログ雑多制御レジスタ (Analog Miscellaneous Control Register)

ビット (\$77)	7	6	5	4	3	2	1	0	
	–	–	–	–	–	AREFEN	XREFEN	ISRCEN	AMISCR
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0 – ISRCEN : 電流源許可 (Current Source Enable)

このビットの1書き込みは図16-1.で示されるように電流源を許可します。ISRCENが設定(1)される時にデジタル入力禁止レジスタ(DIDR)ビットの機能を用いることが推奨されます。可能な限り早く電流源をOFFすることも推奨されます(例:一旦A/D変換測定が行われる場合)。

17.1.2. アナログ比較器反転入力

アナログ比較器への反転入力を取り替えるのに内部基準電圧を選択することができます。アナログ比較器制御/状態レジスタ(ACSR)の内部基準電圧選択(ACIRS)ビットが設定(1)される時にD/A変換器の内部基準電圧を用いて2ビットD/A変換器の出力が利用可能です。基準電圧分圧係数はADCSRBレジスタのアナログ比較器内部基準電圧選択(ACIR1,0)ビットによって行われます。

ACIRSが解除(0)されるなら、アナログ比較器への反転入力にAIN0ピンが印加されます。

表17-2. アナログ比較器反転入力

ACIRS	ACIR1,0	REFS1,0	アナログ比較器反転入力	注釈
0	x	x x	AIN0	
1	x	0 0	(予約)	2.56V内部基準電圧使用
		0 1		
		1 0		
	0 0	1 1	2.56V	
	0 1		1.28V (2.56V/2)	
	1 0		0.64V (2.56V/4)	
	1 1		0.32V (2.56V/8)	

17.2. アナログ比較器用レジスタ

17.2.1. ACSR – アナログ比較器 制御/状態レジスタ (Analog Comparator Control and Status Register)

ビット	7	6	5	4	3	2	1	0	
\$30 (\$50)	ACD	ACIRS	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	ACSR
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	不定	0	0	0	0	0	

● ビット7 – ACD : アナログ比較器禁止 (Analog Comparator Disable)

このビットが論理1を書かれると、アナログ比較器への電力がOFFにされます。このビットはアナログ比較器をOFFにするために何時でも設定(1)できます。これは活動動作やアイドル動作で電力消費を削減します。ACDビットを変更する時にACSRのアナログ比較器割り込み許可(ACIE)ビットを解除(0)することによってアナログ比較器割り込みが禁止されなければなりません。さもなければ、このビットが変更される時に割り込みが起こり得ます。

● ビット6 – ACIRS : 内部基準電圧選択 (Analog Comparator Internal Reference Select)

このビットが設定(1)されると、内部基準電圧(公称1.1V)がアナログ比較器への反転入力に置き換わります(表17-2参照)。本ビットが解除(0)されると、AIN0がアナログ比較器の反転入力に印加されます。41頁の「内部基準電圧」をご覧ください。

● ビット5 – ACO : アナログ比較器出力 (Analog Comparator Output)

アナログ比較器の出力は同期化され、その後直接ACOへ接続されます。この同期化は1~2クロック周期の遅延をもたらします。

● ビット4 – ACI : アナログ比較器割り込み要求フラグ (Analog Comparator Interrupt Flag)

このビットは比較器出力での出来事がACSRのアナログ比較器割り込み条件(ACIS1,0)ビットによって定義した割り込み方法で起動する時に設定(1)されます。ACSRのアナログ比較器割り込み許可(ACIE)ビットが設定(1)され、ステータスレジスタ(SREG)の全割り込み許可(I)ビットが設定(1)されていると、アナログ比較器割り込みルーチンが実行されます。対応する割り込み処理ベクタを実行すると、ACIはハードウェアによって解除(0)されます。代わりにこのフラグへ論理1を書くことによってもACIは解除(0)されます。

● ビット3 – ACIE : アナログ比較器割り込み許可 (Analog Comparator Interrupt Enable)

ACIEビットが論理1を書かれ、ステータスレジスタ(SREG)の全割り込み許可(I)ビットが設定(1)されていると、アナログ比較器割り込みが活性(有効)にされます。論理0を書かれると、この割り込みは禁止されます。

● ビット2 – ACIC : アナログ比較器捕獲起動許可 (Analog Comparator Input Capture Enable)

論理1を書かれると、このビットはアナログ比較器によって起動されるタイマ/カウンタ1の捕獲機能を許可します。この場合、比較器出力は比較器にタイマ/カウンタ1捕獲割り込みの雑音除去機能とエッジ選択機能を利用させる捕獲入力前置論理回路へ直接的に接続されます。論理0を書かれると、アナログ比較器と捕獲機能間の接続は存在しません。比較器がタイマ/カウンタ1捕獲割り込みを起動するには、タイマ/カウンタ1割り込み許可レジスタ(TIMSK1)の捕獲割り込み許可(ICIE1)ビットが設定(1)されなければなりません。

●ビット1,0 – ACIS1,0 : アナログ比較器割り込み条件 (Analog Comparator Interrupt Mode Select)

これらのビットは比較器のどの事象がアナログ比較器割り込みを起動するかを決めます。各種設定は表17-3.で示されます。

注: ACIS1,ACIS0ビットを変更する時にACSRのアナログ比較器割り込み許可(ACIE)ビットを解除(0)することによってアナログ比較器割り込みが禁止されなければなりません。さもないと、これらのビットが変更される時に割り込みが起き得ます。

表17-3. アナログ比較器割り込み条件選択

ACIS1	ACIS0	割り込み発生条件
0	0	比較器出力の変移 (トグル)
0	1	(予約)
1	0	比較器出力の下降端
1	1	比較器出力の上昇端

17.2.2. ADCSRB – A/D変換制御/状態レジスタB (ADC Control and Status Register B)

ビット (\$7B)	7	6	5	4	3	2	1	0	
	BIN	ACME	ACIR1	ACIR0	–	ADTS2	ADTS1	ADTS0	ADCSRB
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	

●ビット6 – ACME : アナログ比較器多重器許可 (Analog Comparator Multiplexer Enable)

このビットが論理1を書かれ、A/D変換部がOFF(ADCSRAのADENビットが0)にされると、A/D変換の多重器がアナログ比較器への非反転入力を選択します。このビットが論理0を書かれると、AIN1がアナログ比較器の非反転入力に印加されます。

A/D変換器がシングル エント 入力チャネルとして構成設定されると、アナログ比較器への非反転入力を取り替えるのにADC10~0のどれでも選択することができます。A/D変換多重器(MUX4~0)はこの入力を選ぶのに使われ、結果として、A/D変換器はこの機能を利用するためにOFFされなければなりません。

●ビット5,4 – ACIR1,0 : アナログ比較器内部基準電圧選択 (Analog Comparator Internal Voltage Reference Select)

アナログ比較器制御/状態レジスタ(ACSR)で内部基準電圧選択(ACIRS)ビットが設定(1)されると、これらのビットはアナログ比較器への反転入力用の基準電圧を選びます。132頁の表17-2.をご覧ください。

17.2.3. DIDR0 – デジタル入力禁止レジスタ0 (Digital Input Disable Register 0)

ビット (\$7E)	7	6	5	4	3	2	1	0	
	ADC7D AIN1D	ADC6D AIN0D	ADC5D	ADC4D	ADC3D	ADC2D	ADC1D	ADC0D	DIDR0
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 – 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、対応するアナログ比較器入力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。

18. A/D変換器 (ADC)

18.1. 特徴

- 10ビット分解能
- 積分非直線性誤差1.0LSB
- 絶対精度 ± 2 LSB
- 変換時間13~260 μ s(低~高分解能)
- 最大分解能で最大15kSPS(採取/s)
- 11の多重化されたシングル エント 入力チャネル
- 8つの選択可能な利得付き差動入力対
- 温度感知器入力チャネル
- 内部電流源駆動(ISRC)からの電圧
- A/D変換結果読み出しに対する任意の左揃え
- 0~AVCC A/D変換入力電圧範囲
- 選択可能な1.1V/2.56V A/D変換基準電圧
- 連続と単独の変換動作
- 割り込み元の自動起動によるA/D変換開始
- A/D変換完了割り込み
- 休止形態雑音低減機能
- 単極性/両極性入力動作
- 入力極性反転動作

18.2. 概要

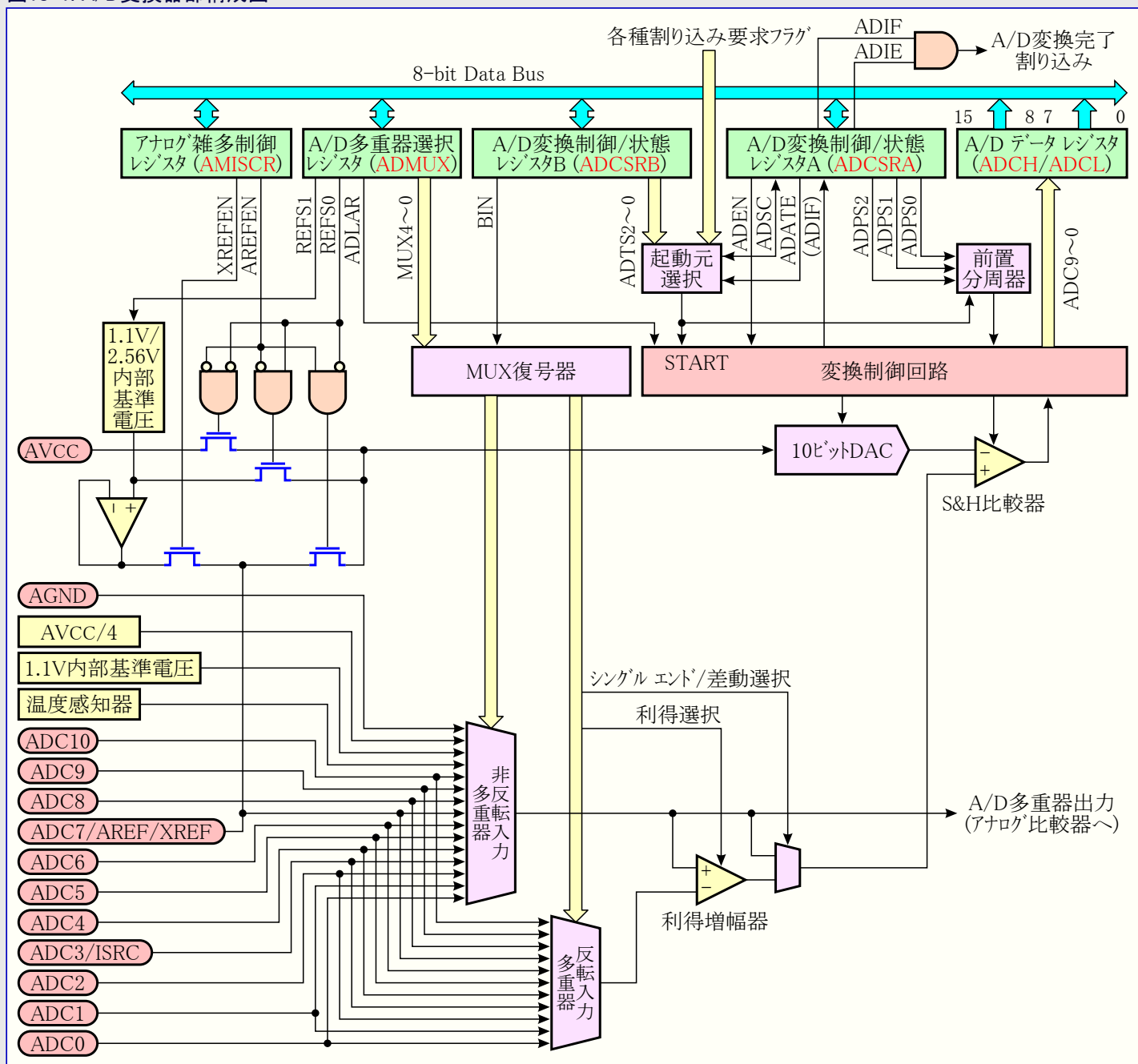
ATtiny87/167は10ビット逐次比較A/D変換器が特徴です。このA/D変換器はPA7~PA0とPB7~PB4で構成した16の差動電圧入力の組み合わせと、11のシングル エント 電圧入力を許す11チャネルのアナログ多重器に接続されます。差動入力A/D変換前の差動入力電圧で $\times 8$ 、 $\times 20$ の増幅段を提供する、設定可能な利得段が装備されます。シングル エント 電圧入力は0V(AGND)が基準です。

このA/D変換器はA/D変換器への入力電圧が変換中に一定の値で保持されることを保証する採取&保持(S/H)回路を含みます。A/D変換部の構成図は図18-1.で示されます。

公称1.1Vまたは2.56Vの内蔵基準電圧がチップ上で提供されます。代わりにAVCCがシングル エント チャネル用の基準電圧として使えます。1.1Vまたは2.56Vの内部基準電圧の出力、または内部基準電圧をOFFにして外部基準電圧を使う任意選択もあります。これらの任意選択はA/D変換多重器選択(ADMUX)レジスタの基準電圧選択(REFS1,0)ビットとアナログ雑多制御レジスタ(AMISCR)の外部基準電圧入力許可(AREFEN)と内部基準電圧出力許可(XREFEN)のビットを使って選択されます。

38頁での「PRR – 電力削減レジスタ」のPRADCBITはA/D変換部を許可するために0を書かれなければなりません。

図18-1. A/D変換器部構成図



18.3. 操作

A/D変換部は逐次比較を通してアナログ入力電圧を10ビットのデジタル値に変換します。最小値はAGNDを表し、最大値はAVCC電圧、AREFピンの基準電圧、または1.1V/2.56V内部基準電圧を表します。

A/D変換器用の基準電圧はADCSRBとA/D多重器選択レジスタ(ADMUX)の基準電圧選択(REFS1,0)ビットとアナログ雑多制御レジスタ(AMISCR)の外部基準電圧入力許可(AREFEN)ビットへの書き込みによって選択できます。AVCC電源、AREFピン、または1.1V/2.56V内部基準電圧がA/D変換器基準電圧として選択できます。

アナログ入力チャネルと差動利得はADMUXレジスタのチャネル選択(MUX4~0)ビットへの書き込みによって選択されます。11のADC入力ピン(ADC10~0)のどれもがA/D変換器のシングル エント 入力として選択できます。差動利得増幅器への非反転及び反転入力は表18-5で記述されます。

差動チャネルが選択されると、差動利得段は選択した入力チャネル対間の差電圧を、ADMUXレジスタのMUX4~0ビットの設定に従って、8または20倍の選択した増幅率で増幅します。それからこの増幅した値はA/D変換器の入力になります。シングル エント チャネルが使われると、利得増幅器全体が迂回(無視)されます。

チップ上の温度感知器は専用のADCチャネルがA/D変換器入力として使われるとき、(即ち)ADMUXレジスタのMUX4~0ビットへ表18-5で定義された符号を書くことによって選択されます。

(表18-5で定義される)特定のADCチャネルは内部電流源(ISRC)によって駆動される電流が流れる外部抵抗の境界電圧を測定するのに使われます。

A/D変換部はA/D変換制御/状態レジスタA(ADCSRA)のA/D許可(ADEN)ビットの設定(1)によって(動作が)許可されます。基準電圧と入力チャネルの選択はADENが設定(1)されるまで実施しません。ADENが解除(0)されているとA/D変換部は電力を消費しないので、節電をする休止形態へ移行する前にA/D変換部をOFFに切り替えることが推奨されます。

A/D変換部はA/Dデータ レジスタ(ADCH,ADCL)で示される10ビットの結果を生成します。既定では、この結果は右揃え(16ビットのビット0側10ビット)で表されますが、ADMUXレジスタで左揃え選択(ADLAR)ビットを設定(1)することにより、任意で左揃え(16ビットのビット15側10ビット)で表せます。

この結果が左揃え補正され、8ビットを越える精度が必要とされない場合はADCHを読むことで足りります。さもないとデータ レジスタの内容が同じ変換に属すること(からの結果)を保証するため、ADCLが初めに、次にADCHが読まれなければなりません。一度ADCLが読まれると、A/D変換器からのA/Dデータ レジスタ(ADCH,ADCL)アクセスが阻止されます。これはADCLが読まれてしまい、ADCHが読まれる前に変換が完了すると、どちらのレジスタ(ADCH,ADCL)も更新されず、その変換からの結果が失われることを意味します。ADCHが読まれると、ADCH,ADCLへのA/D変換器アクセスが再び許可されます。

A/D変換部には変換完了時に起動できる自身の割り込みがあります。A/Dデータ レジスタへのA/D変換器アクセスがADCLとADCHの読み込み間で禁止されている場合、例えその変換結果が失われても割り込みは起動します。

18.4. 変換の開始

単独変換はADCSRAレジスタで変換開始(ADSC)ビットに論理1を書くことによって開始されます。このビットは変換が進行中である限り、1に留まり、変換が完了されるとハードウェアによって解除(0)されます。変換が進行中に違う入力チャネルが選択されると、A/D変換部はそのチャネル変更を実行する前に現在の変換を済ませます。

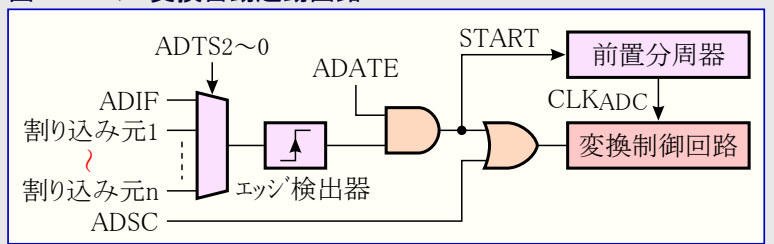
代わりに、変換は様々な起動元によって自動的に起動できます。自動起動はA/D変換制御/状態レジスタA(ADCSRA)のA/D変換自動起動許可(ADATE)ビットの設定(1)によって許可されます。起動元はA/D変換制御/状態レジスタB(ADCSRB)のA/D変換起動元選択(ADTS2~0)ビットの設定によって選択されます(起動元の一覧についてはADTSビットの記述をご覧ください)。選択した起動信号上に上昇端が起きると、A/D変換用前置分周器がリセットし、変換が開始されます。これは一定間隔での変換開始の方法を提供します。変換完了時、起動信号が未だ設定(1)されている場合は新規の変換が開始されません。変換中にこの起動信号上で別の上昇端が起きると、そのエッジは無視されます。指定した割り込みが禁止またはステータス レジスタ(SREG)の全割り込み許可(I)ビットが解除(0)でも、割り込み要求フラグが設定(1)されることに注意してください。従って割り込みを起こさずに変換が起動できます。けれども次の割り込み要因で新規変換を起動するために、割り込み要求フラグは解除(0)されなければなりません。

起動元としてA/D変換完了割り込み要求フラグ(ADIF)を使うことは、A/D変換器に実行中の変換が完了されると直ぐに新規変換を開始させます。そのためA/D変換器は連続動作で動き、継続的な採取(変換)とA/Dデータ レジスタを更新します。最初の変換はADCSRAレジスタでADSCビットに論理1を書くことにより始めなければなりません。この動作でのA/D変換器はA/D変換完了割り込み要求フラグ(ADIF)が解除(0)されるかどうかにかかわらず、連続的な変換を実行します。

自動起動が許可されている場合、ADCSRAレジスタのADSCビットに論理1を書くことによって単独変換を開始できます。ADSCは変換が進行中かを決めるためにも使えます。ADSCビットは変換がどう開始されたかに拘らず、変換中は1として読めます。

変換はA/D変換雑音低減機能の使用によっても開始され得ます。この機能はアイドル休止動作とA/D変換雑音低減休止動作中に変換を許可します。詳細については139頁の「雑音低減機能」をご覧ください。(訳注:共通性から2行追加)

図18-2. A/D変換自動起動回路



18.5. 前置分周と変換タイミング

既定での逐次比較回路は最大分解能を得るのに50～200kHzの入力クロック周波数を必要とします。10ビットよりも低い分解能が必要とされるなら、A/D変換器への入力クロック周波数はより高い採取速度を得るために200kHzよりも高くできます。

A/D変換部は100kHz以上のどんなCPUクロックからも受け入れ可能なA/D変換クロック周波数を生成する前置分周器を含みます。この前置分周はA/D変換制御/状態レジスタA(ADCSRA)のA/Dクロック選択(ADPS2～0)ビットによって設定されます。前置分周器はADCSRAレジスタでA/D許可(ADEN)ビットの設定(1)によってA/D変換部がONにされた瞬間から計数を始めます。前置分周器はADENビットが設定(1)される限り走行を保ち、ADENが0の時は継続的にリセットします。

ADCSRAレジスタのA/D変換開始(ADSC)ビットの設定(1)によってシングルエンド入力の変換を起動すると、その変換は直後の変換クロックの上昇端で始まります。

通常の変換は13変換クロック周期で行われます。A/D変換部がONされた(ADCSRAレジスタのADEN=1)後の最初の変換はアナログ回路を初期化するために25変換クロック周期で行われます。

実際の採取&保持(保持開始点)は通常変換の開始後1.5変換クロック周期、初回変換の開始後13.5変換クロック周期で行われます。変換が完了すると、結果がA/Dデータレジスタ(ADCH,ADCL)に書かれ、ADCSRAのA/D変換完了割り込み要求フラグ(ADIF)が設定(1)されます。単独変換動作(ADATE=0)では同時にADCSRAのA/D変換開始(ADSC)ビットが解除(0)されます。その後にソフトウェアは再びADSCを設定(1)でき、新規変換は変換クロックの最初の上昇端で開始されます。

自動起動が使われると、前置分周器は起動要因発生時にリセットされます。これは起動要因から変換開始までの一定の遅延を保証します。この動作での採取&保持は起動要因となる信号の上昇後、2変換クロック周期で採取が行われます。同期化論理回路(エッジ検出器)に対して、追加の3CPUクロック周期が費やされます。

連続変換動作(ADATE=1)では変換完了後直ちに新規変換が開始され、一方ADSCは1に留まります。変換時間の概要については表18-1をご覧ください。

図18-3. A/D変換前置分周器部構成

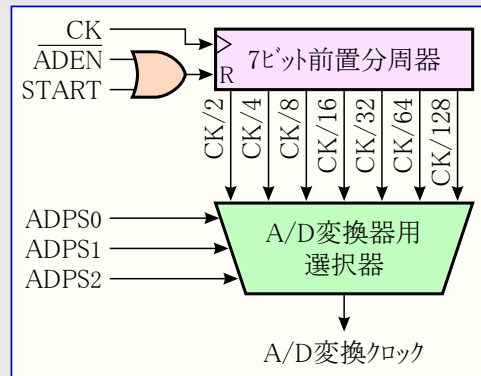


図18-4. 初回変換タイミング (単独変換動作)

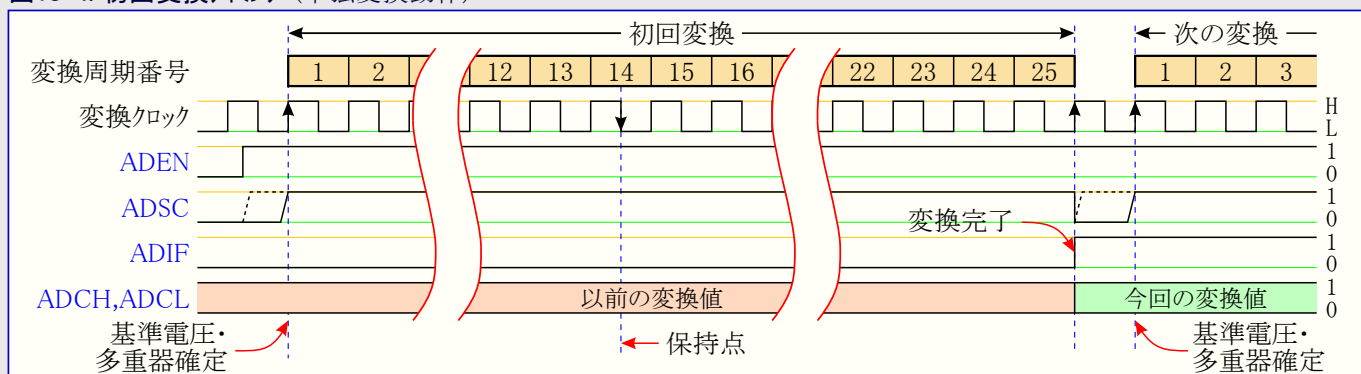


図18-5. 通常変換タイミング (単独変換動作)

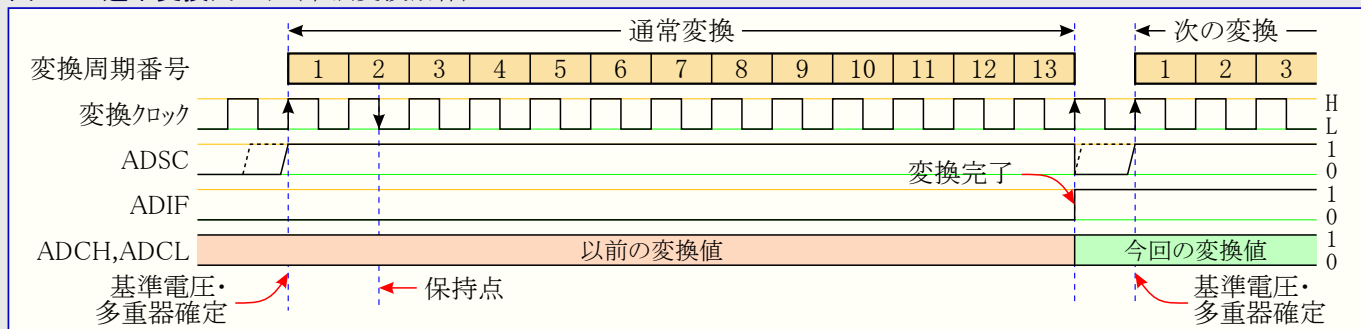


図18-6. 通常変換タイミング (自動起動変換動作)

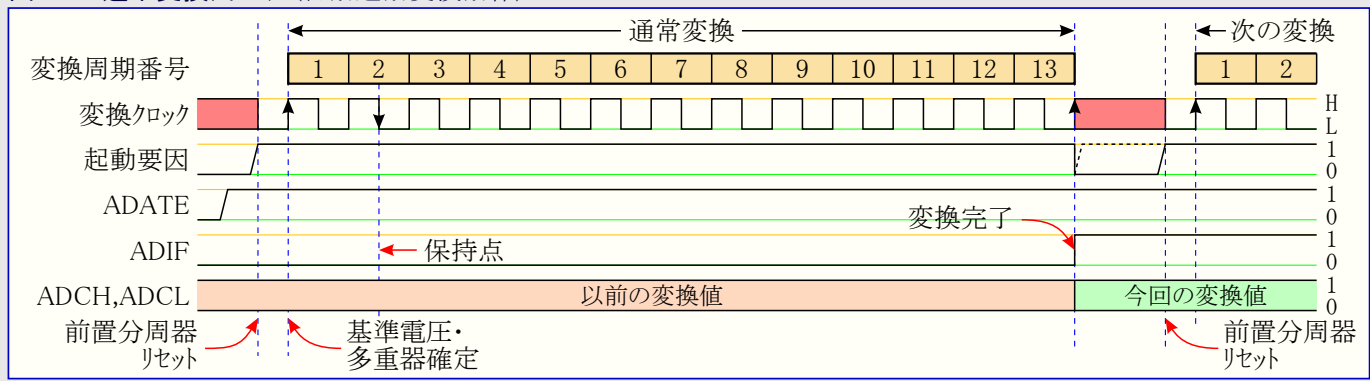


図18-7. 連続変換動作タイミング

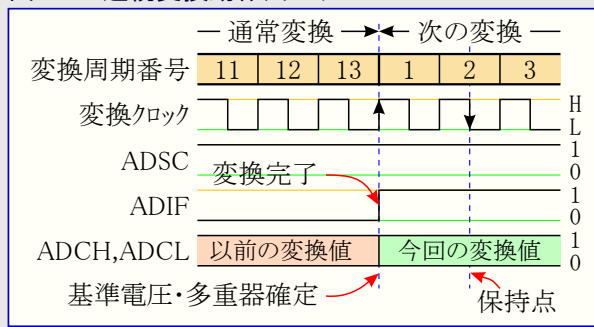


表18-1. A/D変換時間

変換種別	保持点	変換時間
初回変換	13.5	25
シングルエンド入力通常変換	1.5	13
自動起動変換	1.5 (2)	13.5

注: 変換時間を除く各値は変換開始からの変換クロック数です。

18.6. チャネル変更と基準電圧選択

A/D多重器選択レジスタ(ADMUX)のチャネル選択(MUX4~0)ビットと基準電圧選択(REFS1,0)ビットはCPUが乱順にアクセスするための一時レジスタを通し単独緩衝されます。これはチャネルと基準電圧の選択が変換中の安全な所でだけ行いのを保証します。チャネルと基準電圧の選択は変換が開始されるまで継続的に更新されます。一旦変換が始まると、A/D変換器に対して充分な採取/変換時間を保証するためにチャネルと基準電圧の選択は固定されます。継続的な更新は変換完了(ADCSRAのADIF=1)前の最後の変換クロック周期で再開します。ADCSRAレジスタの変換開始(ADSC)ビットが書かれた後の次の変換クロックの上昇端で変換が始まることに注意してください。従って使用者はADSC書き込み後、1変換クロック周期(経過)まで新しいチャネルまたは基準電圧選択値をADCSRBとADMUXに書かないことが推奨されます。

自動起動が使われる場合、起動要因の正確な時間は確定できません。変換が新規設定によって影響されるように制御するにはADMUXレジスタの更新時に特別な注意が払われなければなりません。

ADCSRAレジスタのA/D許可(ADEN)とA/D変換自動起動許可(ADATE)の両方が1を書かれると、何時でも割り込みが起き得ます。この期間でADMUXレジスタが変更されると、使用者は次の変換が旧新のどちらの設定が基準にされるかを知ることができません。ADMUXレジスタは次の方法で安全に更新できます。

- ADENまたはADATEが解除(0)されているとき。
- 変換開始後、最低1変換クロック周期経過後の変換中。
- 変換後から、変換起動元として使った割り込みフラグが解除(0)される直前まで。

これら条件の1つでADCSRBとADMUXを更新すると、新設定は次のA/D変換に影響を及ぼします。

18.6.1. A/D入力チャネル

チャネル選択を変更する時に使用者は正しいチャネルが選択されることを保証するために次の指針を守るべきです。

単独変換動作では常に変換を始める前にチャネルを選択してください。チャネル選択はADSCへの1書き込み後、1変換クロック周期で変更されるかもしれません。とは言え、最も簡単な方法はチャネル選択を変更する前に変換が完了するまで待つことです。

連続変換動作では常に最初の変換を始める前にチャネルを選択してください。チャネル選択はADSCへの1書き込み後、1変換クロック周期で変更されるかもしれません。とは言え、最も簡単な方法は最初の変換が完了するまで待ち、その後にチャネル選択を変更することです。既に次の変換が自動的に開始されているので、次の結果は直前のチャネル選択を反映します。それに続く変換は新しいチャネル選択を反映します。

18.6.2. A/D変換基準電圧

このA/D変換用の基準電圧(VREF)はA/D変換に対する変換範囲を示します。VREFを越えるシングルエンド入力チャネルは\$3FFで打ち切るコードに帰着します。VREFはAVCC、1.1V/2.56V内部基準電圧、外部AREFピンのどれかとして選択できます。基準電圧源切り替え後の最初のA/D変換結果は不正確かもしれず、使用者はこの結果を破棄することが推奨されます。

18.7. 雑音低減機能

A/D変換部はCPUコアと他の周辺I/Oが誘導した雑音を低減するために**休止形態**中の変換を可能にする雑音低減機能が特徴です。雑音低減機能は**A/D変換雑音低減動作**と**アイドル動作**で使うことができます。この機能を使うには次の手順が使われるべきです。

1. A/D変換部が許可(**ADEN=1**)され、変換中でない(**ADSC=0**)ことを確認してください。単独変換動作が選択(**ADATE=0**)され、且つA/D変換完了割り込みが許可(**ADIE=1**)されていなければなりません。
2. A/D変換雑音低減(またはアイドル)動作に移行してください。一旦CPUが停止されてしまうと、A/D変換部は変換を始めます。
3. A/D変換完了前に他の割り込みが起こらなければ、A/D変換完了割り込みはCPUを起動してA/D変換完了割り込みルーチンを実行します。A/D変換完了前に他の割り込みがCPUを起動すると、その割り込みが実行され、A/D変換完了割り込み要求はA/D変換完了時に生成されます。CPUは新規**SLEEP**命令が実行されるまで活動動作に留まります。

アイドル動作とA/D変換雑音低減動作を除く他の休止形態へ移行する時にA/D変換部が自動的にOFFへ切り替えられないことに注意してください。使用者は余分な消費電力を避けるため、このような休止形態へ移行する前にADENへ**0**を書くことが推奨されます。

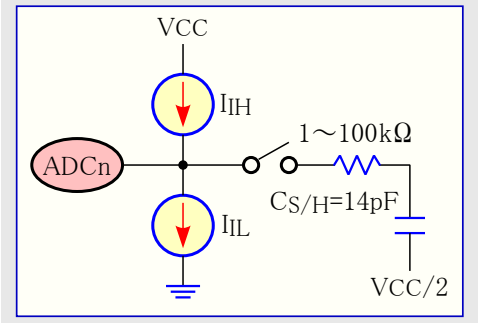
18.7.1. アナログ入力回路

シングル エント 入力チャネルのアナログ回路は**図18-8**で図示されます。ADCnに印加したアナログ(信号)源はそのチャネルがADC入力として選択されているかどうかにかかわらず、ピン容量とそのピンの漏れ電流に左右されます。そのチャネルが選択されると、(アナログ信号)源は直列抵抗(入力経路の合成抵抗)を通してS/Hコンデンサを駆動しなければなりません。

A/D変換部は概ね10kΩ若しくはそれ以下の出力インピーダンスのアナログ信号用に最適化されています。このような(アナログ信号)源が使われるなら、採取時間は無視してもよいでしょう。より高いインピーダンスの(アナログ信号)源が使われる場合、採取時間は広範囲に変化し得るS/Hコンデンサを充電するために(アナログ信号)源がどれくらいの時間を必要とするかに依存します。必要とされるS/Hコンデンサへの充放電を最小とするため、使用者は緩やかに変化する低インピーダンス(アナログ信号)源だけを使うことが推奨されます。

特定できない信号の渦からの歪を避けるために、どのチャネルに対してもナイキスト周波数($f_{ADC}/2$)よりも高い信号成分が存在すべきではありません。使用者はADC入力として信号を印加する前に低域通過濾波器(ローパス フィルタ)で高い周波数成分を取り除くことが推奨されます。

図18-8. アナログ入力回路



18.7.2. アナログ雑音低減技術

デバイス内外のデジタル回路がアナログ測定の精度に影響を及ぼすかもしれないEMIを発生します。変換精度が重要なら、次の技法を適用することによって雑音レベルを低減できます。

- アナログ信号経路を可能な限り最短にしてください。アナログ信号経路がアナログGND面上を走ることを確認し、それらを高速切り替えデジタル信号線から充分分離することを守ってください。
- CPUからの誘導雑音を減らすために**A/D変換雑音低減機能**を使ってください。
- 何れかのADCポート ピンがデジタル出力として使われる場合、変換進行中にこれらを切り替えないことが重要です。

18.7.3. A/D変換の精度定義

シングル エント 入力電圧のnビットA/D変換はGNDとVREF間を 2^n で直線的に変換します。最低値コードは0として読み、最高値コードは 2^n-1 として読みます。以下の各種パラメータは理想状態からの偏差を表します。

• オフセット誤差 – 図18-9.

最初の遷移点(\$000から\$001)に於いて理想遷移点(差0.5LSB)と比較した偏差です。理想値は0LSBです。

• 利得誤差 – 図18-10.

オフセット誤差補正後の最後の遷移点(\$3FEから\$3FF)に於いて理想遷移点(最大差1.5LSB以下)と比較した偏差です。理想値は0LSBです。

• 積分非直線性誤差 (INL) – 図18-11.

オフセット誤差と利得誤差補正後の全ての遷移点に於いて理想遷移点と比較した最大偏差です。理想値は0LSBです。

• 微分非直線性誤差 (DNL) – 図18-12.

実際のコードの幅(隣接する2つの遷移点間)に於いて理想コード幅(1LSB)と比較した最大偏差です。理想値は0LSBです。

• 量子化誤差

有限数のコードで入力電圧を量子化するため、1LSB幅となる入力電圧範囲は同じ値のコードになります。この値は常に ± 0.5 LSBです。

• 絶対精度

補正しない全ての遷移点に於いて理想遷移点と比較した最大偏差です。これは、オフセット誤差、利得誤差、差動誤差、非直線誤差の影響の合成です。理想値は ± 0.5 LSBです。

図18-9. オフセット誤差

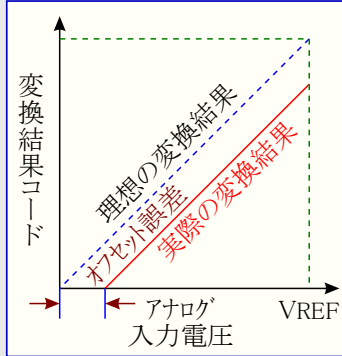


図18-10. 利得誤差

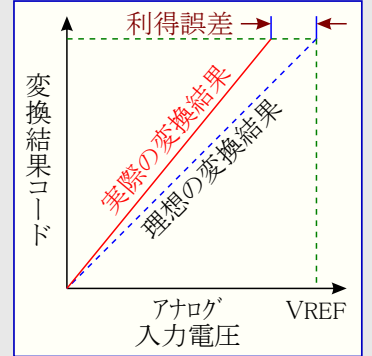


図18-11. 積分非直線性誤差

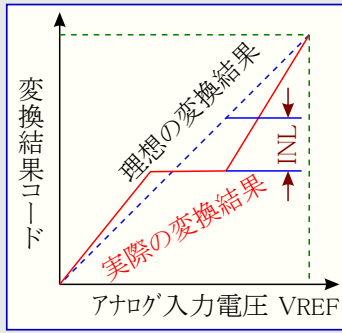
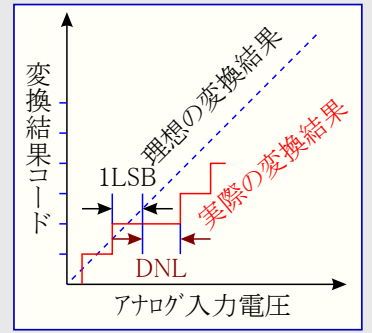


図18-12. 微分非直線性誤差



18.8. A/D変換の結果

変換完了(ADIF=1)後、変換結果はA/Dデータレジスタ(ADCH,ADCL)で得られます。変換結果の形式はシングル エント 変換、単極差動変換、両極差動変換の3つの変換形式に依存します。

18.8.1. シングル エント 変換

シングル エント 入力変換での結果は右式で示されます。

VINは選択した入力ピンの電圧で、VREFは選択した基準電圧です(142頁の表18-4.と142頁の表18-5.をご覧ください)。\$000はアナログGNDを表し、\$3FFは選択した基準電圧-1LSBを表します。結果は0(\$000)~1023(\$3FF)の符号なし2進数で表されます。

$$ADC = \frac{V_{IN} \times 1024}{V_{REF}}$$

18.8.2. 単極差動変換

差動チャネルで単極入力動作が使われると、その結果は右式で示されます。

VPOSは非反転入力ピンの電圧、VNEGは反転入力ピンの電圧、GAINは選択した倍率、VREFは選択した基準電圧です(142頁の表18-4.と142頁の表18-5.をご覧ください)。非反転ピンの電圧は常に反転ピンの電圧より高くなければならず、さもなければ差電圧は0に飽和します。結果は0(\$000)~1023(\$3FF)の符号なし2進数で表されます。GAINは8,20倍のどちらかです。

$$ADC = \frac{(V_{POS} - V_{NEG}) \times GAIN \times 1024}{V_{REF}}$$

18.8.3. 両極差動変換

既定としてA/D変換器は単極入力動作で働きますが、A/D変換制御/状態レジスタB(ADCSRB)の両極入力動作(BIN)ビットの1書き込みにより両極入力動作が選べます。両極入力動作では \pm 両側の差電圧が許され、従って反転入力ピンの電圧は非反転入力ピンの電圧よりも高くもできます。差動チャネルで両極入力動作が使われると、その結果は右式で示されます。

$$ADC = \frac{(V_{POS} - V_{NEG}) \times GAIN \times 512}{V_{REF}}$$

VPOSは非反転入力ピンの電圧、VNEGは反転入力ピンの電圧、GAINは選択した倍率、VREFは選択した基準電圧です。結果は-512(\$200)から0(\$000)を通して+511(\$1FF)までの符号付き2進数で表されます。GAINは8,20倍のどちらかです。

けれども信号が本質動作(10ビットとして符号+9ビット)で両極でない場合、この仕組みは変換器の機能範囲(ダイナミックレンジ)の1ビットを失います。そこで使用者が最大機能範囲での変換実行を欲するなら、使用者は(変換)結果を素早く調べ、選択可能な差動入力対で単極差動変換を使えます。極性検査実行時、それは結果のMSB読み込み(ADCHレジスタのADC9ビット)で充分です。このビットが1ならば結果が負、0ならば正です。

18.9. 温度測定

温度測定はシングル エント ADCチャネルに連結されるチップ上の温度感知器に基づきます。A/D多重器選択(ADMUX)レジスタのチャンネル選択(MUX4~0)ビットがこの温度感知器を許可します。温度感知器測定に於いて1.1V内部基準電圧もA/D変換器基準電圧源に選択されなければなりません。温度感知器が許可されると、A/D変換器は温度感知器上の電圧測定を単独変換動作(ADATE=0)で使うことができます。

測定した電圧は表18-2.で記述されたように温度に対して直線的関係を持ちます。感度は概ね1LSB/°Cで、精度は使用者校正に依存します。代表的には、室温校正との仮定で、単独温度校正後の測定精度は±10°Cです。より良好な精度は校正に対して2つの温度点を使うことで達せられます。

表18-2. 温度対感知器出力電圧 (代表値)

温度(°C)	-40°C	+25°C	+85°C
A/D変換値(LSB)	246	324	440

表18-2.に記載した値は代表値です。然しながら製法変化のため、温度感知器出力電圧は或るチップと別のチップで変化します。より正確な結果達成を可能とするために温度測定は応用ソフトウェアで校正できます。校正ソフトウェアは次式を使って行なえます。

$$T(\text{温度}) = k \times (\text{ADCH} \ll 8 | \text{ADCL}) - T_{OS}$$

ここでADCH:ADCLはA/D変換器データレジスタ、 T はケルビンでの温度、 k は固定傾斜係数、 T_{OS} は温度感知器オフセット値です。代表的に係数 k は1.0に大変近く、単点校正ではこの係数を省略できるかもしれません。高精度が要求される場合、傾斜係数は2つ温度での測定を基に評価されるべきです。

18.10. 内部基準電圧出力

A/D変換器がONの場合に表18-3.で記述されるように内部基準電圧はXREFピンに出力されます(41頁の「基準電圧許可信号と起動時間」をご覧ください)。XREFピンに外部濾波容量(5~10pF)の追加が必要かもしれません。XREF電流負荷はXREF=1.1Vに対して2.7~5.5V、XREF=2.56Vに対して4.5~5.5VのVCCで1~100µAでなければなりません。

XREFピンはA/D変換器のアナログ入力に繋ぐことができます(7頁の「ピン配置」をご覧ください)。

表18-3. 内部基準電圧出力

XREFEN (注1)	REFS1 (注2)	REFS0 (注2)	基準電圧出力 ($I_{load} \leq 100\mu A$)
0	x	x	Hi-Z、ピンはAREF入力または他の交換機能として使用することができます。
1 (注1)	0	1	XREF=1.1V (注3)
	1	1	XREF=2.56V (注3,4)

注1: 145頁の「ビット1 - XREFEN : 内部基準電圧出力許可」をご覧ください。

注2: 142頁の「ビット7,6 - REFS1,0 : 基準電圧選択」をご覧ください。

注3: これらの構成設定ではピンのプルアップがOFFにされ、ピンのデジタル出力がHi-Zに設定されなければなりません。

注4: VCCは4.5~5.5Vの範囲内です。

18.11. A/D変換用レジスタ

18.11.1. ADMUX – A/D多重器選択レジスタ (ADC Multiplexer Select Register)

ビット (\$7C)	7	6	5	4	3	2	1	0	
	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	ADMUX
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

●ビット7,6 – REFS1,0 : 基準電圧選択 (Voltage Reference Selection Bits)

これらのビットとアナログ雑多制御レジスタ(AMISCR)からの外部基準電圧入力許可(AREFEN)ビットは、表18-4.で示されるようにA/D変換器の基準電圧を選びます。これらのビットが変換中に変更されると、その変更は変換が完了する(ADCSRAのADIF=1)まで実施しません。これらのビットが変更されると必ず、次の変換は25 ADC変換クロックかかります。活動的なチャンネルが使われる場合、AVCCまたは(AVCC-1V)よりも高い外部AREFの使用は、これがA/D変換精度に影響を及ぼすために推奨されません。AREFピンに外部電圧が印加されている場合、内部基準電圧選択は使うことができません。

表18-4. A/D変換部の基準電圧選択

REFS1	REFS0	AREFEN	基準電圧 (VREF)
x	0	0	基準電圧としてAVCC使用、AREFピンから切断
x	0	1	AREFピンの外部基準電圧 (AREF ≥ 2.0V)
0	1	0	1.1V内部基準電圧
1	1	0	2.56V内部基準電圧

●ビット5 – ADLAR : 左揃え選択 (ADC Left Adjust Result)

ADLARビットはA/Dデータレジスタ内の変換結果の配置に影響を及ぼします。結果を左揃えにするにはADLARに1を書いてください。さもなければ結果は右揃えです。ADLARビットの変更はどんな進行中の変換にも拘らず、直ちにA/Dデータレジスタの内容に影響を及ぼします。このビットの完全な記述については144頁の「A/Dデータレジスタ」をご覧ください。

●ビット4~0 – MUX4~0 : A/Dチャネル選択 (Analog Channel and Gain Selection Bits 4~0)

これらのビットはA/D変換器にどのアナログ入力の組み合わせが接続されるかを選びます。差動入力の場合、利得選択もこれらのビットで行なわれます。差動利得段への両入力として同じピンの選択はオフセット測定を可能にします。シングルエンドチャネルADC11選択は温度感知器を許可します。詳細については表18-5.をご覧ください。これらのビットが変換中に変更される場合、その変更は変換が完了する(ADCSRAレジスタのADIF=1)まで実施しません。

表18-5. アナログ入力チャネル選択

MUX4~0	シングルエンド 入力	差動入力			MUX4~0	シングルエンド 入力	差動入力		
		非反転入力	反転入力	利得			非反転入力	反転入力	利得
00000	ADC0 (PA1)				10000		ADC0	ADC1	×8
00001	ADC1 (PA1)				10001		(PA0)	(PA1)	×20
00010	ADC2 (PA2)				10010		ADC1	ADC2	×8
00011	ADC3 (PA3)				10011		(PA1)	(PA2)	×20
00100	ADC4 (PA4)				10100		ADC2	ADC3	×8
00101	ADC5 (PA5)				10101		(PA2)	(PA3)	×20
00110	ADC6 (PA6)				10110		ADC4	ADC5	×8
00111	ADC7 (PA7)				10111		(PA4)	(PA5)	×20
01000	ADC8 (PB5)				11000		ADC5	ADC6	×8
01001	ADC9 (PB6)				11001		(PA5)	(PA6)	×20
01010	ADC10 (PB7)				11010		ADC6	ADC7	×8
01011	温度感知器				11011		(PA6)	(PA7)	×20
01100	1.1V(VBG)				11100		ADC8	ADC9	×8
01101	AVCC/4				11101		(PB5)	(PB6)	×20
01110	0V(AGND)				11110		ADC9	ADC10	×8
01111	(予約)				11111		(PB6)	(PB7)	×20

18.11.2. ADCSRA – A/D制御/状態レジスタ (ADC Control and Status Register A)

ビット (\$7A)	7	6	5	4	3	2	1	0	
	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	0	0	0	0	0	0	0	0	

● ビット7 – ADEN : A/D許可 (ADC Enable)

このビットに**1**を書くことがA/D変換部(動作)を許可します。**0**を書くことによってA/D変換部は(電源が)OFFされます。変換が進行中にA/D変換部をOFFにすることはその変換を(途中)終了します。

● ビット6 – ADSC : A/D変換開始 (ADC Start Conversion)

単独変換動作で各変換を始めるにはこのビットへ**1**を書いてください。連続変換動作で最初の変換を始めるにはこのビットへ**1**を書いてください。A/D変換部が許可される(ADEN=**1**)と同時にADSCが書かれるか、またはA/D変換部が許可されてしまった後にADSCが書かれた後の**初回変換**は、通常の13に代わって25変換クロック周期で行います。この初回変換はA/D変換部の初期化を実行します。

ADSCは変換が進行中である限り**1**として読めます。変換が完了すると**0**に戻ります。このビットへの**0**書き込みは無効です。

● ビット5 – ADATE : A/D変換自動起動許可 (ADC Auto Trigger Enable)

このビットが**1**が書かれると、A/D変換の自動起動が許可されます。A/D変換器は選択した起動信号の上昇端で変換を開始します。この起動元はA/D変換制御/状態レジスタB(ADCSRB)のA/D変換起動要因選択(ADTS2~0)ビット設定によって選択されます。

● ビット4 – ADIF : A/D変換完了割り込み要求フラグ (ADC Interrupt Flag)

A/D変換が完了し、A/Dデータレジスタが更新されると、このフラグが設定(**1**)されます。ステータスレジスタ(SREG)の全割り込み許可(I)ビットとA/D変換完了割り込み許可(ADIF)ビットが設定(**1**)されていれば、A/D変換完了割り込みが実行されます。対応する割り込み処理ベクタを実行する時にADIFはハードウェアによって解除(**0**)されます。代わりにこのフラグに論理**1**を書くことによってもADIFは解除(**0**)されます。ADCSRAで読み-変更-書き(リード モディファイ ライト)を行うと、保留中の割り込みが禁止され得ることに注意してください。これはSBI,CBI命令が使われる場合にも適用されます。

● ビット3 – ADIE : A/D変換完了割り込み許可 (ADC Interrupt Enable)

このビットが**1**が書かれ、SREGの全割り込み許可(I)ビットが設定(**1**)されていると、A/D変換完了割り込みが活性に(許可)されます。

● ビット2~0 – ADPS2~0 : A/D変換クロック選択 (ADC Prescaler Select Bits)

これらのビットはシステム クロック周波数とA/D変換部への入力クロック間の分周値を決めます。

表18-6. A/D変換クロック選択 (CK=システム クロック)

ADPS2	0	0	0	0	1	1	1	1
ADPS1	0	0	1	1	0	0	1	1
ADPS0	0	1	0	1	0	1	0	1
A/D変換クロック	CK/2	CK/2	CK/4	CK/8	CK/16	CK/32	CK/64	CK/128

18.11.3. ADCSRB – A/D変換 制御/状態レジスタB (ADC Control and Status Register B)

ビット (\$7B)	7	6	5	4	3	2	1	0	
	BIN	ACME	ACIR1	ACIR0	–	ADTS2	ADTS1	ADTS0	ADCSRB
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	

● ビット7 – BIN : 両極入力動作 (Bipolar Input Mode)

利得段は既定として単極動作で動きますが、ADCSRBのBINビット(**1**)書き込みによって両極動作が選択できます。単極動作では(±)の片側変換だけが支援され、非反転入力電圧は反転入力電圧よりも常に高くなければなりません。さもなければ、その結果は基準電圧に飽和されます。両極動作では(±)の両側変換が支援され、その結果は2の補数形式で表されます。単極動作での分解能は10ビットで、両極動作での分解能は9ビット+1符号ビットです。

● ビット5 – Res : 予約 (Reserved Bits)

このビットは予約されています。将来のデバイスとの互換性のため、ADCSRBレジスタが書かれる時に**0**が書かれなければなりません。

●ビット2～0 – ADTS2～0 : A/D変換自動起動要因選択 (ADC Auto Trigger Source)

A/D変換制御/状態レジスタ(ADCSRA)のA/D変換自動起動許可(ADATE)ビットが1を書かれると、これらのビットの値はどの起動元がA/D変換を起動するかを選択します。ADATEが解除(0)されると、ADTS2～0設定は無効です。変換は選択した割り込みフラグの上昇端によって起動されます。解除(0)されている起動元から設定(1)されている起動元への切り替えが、起動信号上に上昇端を生成することに注意してください。ADC SRAレジスタのA/D許可(ADEN)ビットが設定(1)されているなら、これが変換を開始させます。連続変換動作(ADTS2～0=0)への切り替えは、例えばA/D変換完了割り込み要求フラグが設定(1)されていても、起動事象を引き起こしません。

表18-7. A/D変換自動起動元選択

ADTS2	ADTS1	ADTS0	起動元
0	0	0	連続変換動作
0	0	1	アナログ比較器
0	1	0	外部割り込み要求0
0	1	1	タイマ/カウンタ1比較A一致
1	0	0	タイマ/カウンタ1溢れ
1	0	1	タイマ/カウンタ1比較B一致
1	1	0	タイマ/カウンタ1捕獲事象
1	1	1	ウォッチドッグ割り込み要求

18.11.4. ADCH,ADCL – A/Dデータレジスタ (ADC Data Register)

ADLAR=0時								
ビット (\$78)	15	14	13	12	11	10	9	8
	–	–	–	–	–	–	ADC9	ADC8
Read/Write	R	R	R	R	R	R	R	R
初期値	0	0	0	0	0	0	0	0
ビット (\$79)	7	6	5	4	3	2	1	0
	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0
Read/Write	R	R	R	R	R	R	R	R
初期値	0	0	0	0	0	0	0	0
ADLAR=1時								
	15	14	13	12	11	10	9	8
	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2
	7	6	5	4	3	2	1	0
	ADC1	ADC0	–	–	–	–	–	–

A/D変換が完了すると、その結果がこれら2つのレジスタで得られます。

ADCLが読まれると、A/DデータレジスタはADCHが読まれるまで更新されません。従ってこの結果が左揃え、且つ8ビットを越える精度が必要とされないなら、ADCHを読むことで用が足ります。さもなくばADCLが初めに、その後にADCHが読まれなければなりません。

A/D多重器選択レジスタ(ADMUX)の左揃え選択(ADLAR)ビットとA/Dチャネル選択(MUX4～0)ビットは本レジスタから結果を読む方法に影響を及ぼします。ADLARが設定(1)されると結果は左揃えされます。ADLARが解除(0)されていると(既定)、結果は右揃えされます。

●ADC9～0 : A/D変換結果 (ADC Conversion result)

これらのビットは140頁の「A/D変換の結果」で詳述されるように変換での結果を表します。

18.11.5. DIDR0 – デジタル入力禁止レジスタ0 (Digital Input Disable Register 0)

ビット (\$7E)	7	6	5	4	3	2	1	0
	ADC7D AIN1D	ADC6D AIN0D	ADC5D	ADC4D	ADC3D	ADC2D	ADC1D	ADC0D
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 – ADC7D～ADC0D : ADC7～0 デジタル入力禁止 (ADC7～0 Digital Input Disable)

このビットが論理1を書かれると、対応するADCnピンのデジタル入力緩衝部が禁止されます。このビットが設定(1)されると、対応するポート入力レジスタのビット(PINx)は常に0として読みます。アナログ信号がADC7～0ピンに印加され、そのピンからのデジタル入力が必要とされない時にデジタル入力緩衝部での消費電力を削減するため、そのビットは論理1を書かれるべきです。

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

ビット (\$7F)	7	6	5	4	3	2	1	0	
	–	ADC10D	ADC9D	ADC8D	–	–	–	–	DIDR1
Read/Write	R	R/W	R/W	R/W	R	R	R	R	
初期値	0	0	0	0	0	0	0	0	

● ビット6～4 – ADC10D～ADC8D : ADC10～8 デジタル入力禁止 (ADC10～8 Digital Input Disable)

このビットが論理1を書かれると、対応するADCnピンのデジタル入力緩衝部が禁止されます。このビットが設定(1)されると、対応するポート入力レジスタのビット(PINx)は常に0として読みます。アナログ信号がADC10～8ピンに印加され、そのピンからのデジタル入力が必要とされない時にデジタル入力緩衝部での消費電力を削減するため、そのビットは論理1を書かれるべきです。

● ビット7,3～0 – Res : 予約 (Reserved Bits) (訳注:共通性から本項追加)

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のため、DIDR1が書かれる時にこれらのビットは0が書かれなければなりません。

18.11.7. AMISCR – アナログ雑多制御レジスタ (Analog Miscellaneous Control Register)

ビット (\$77)	7	6	5	4	3	2	1	0	
	–	–	–	–	–	AREFEN	XREFEN	ISRCEN	AMISCR
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 Bits)

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のため、AMISCRが書かれる時にこれらのビットは0が書かれなければなりません。

● ビット2 – AREFEN : 外部基準電圧入力許可 (External Voltage Reference Input Enable)

このビットが論理1を書かれると、A/D変換器用の基準電圧は142頁の表18-4.で記述されるようにAREFピンから入力されます。活動的なチャネルが使われる場合、AVCCまたは(AVCC-1V)よりも高い外部AREFの使用は、これがA/D変換精度に影響を及ぼすために推奨されません。AREFピンに外部電圧が印加されている場合、内部基準電圧選択は使うことができません。AREFENが設定(1)される時はDIDRレジスタビットの機能(デジタル入力禁止)を使うことが推奨されます。

● ビット1 – XREFEN : 内部基準電圧出力許可 (Internal Voltage Reference Output Enable)

このビットが論理1を書かれると、141頁の「内部基準電圧出力」で記述されるように1.1Vまたは2.56Vの内部基準電圧がXREFピンで出力されます。XREFENが設定(1)される時はDIDRレジスタビットの機能(デジタル入力禁止)を使うことが推奨されます。

19. デバッグWIRE 内蔵デバッグ システム

19.1. 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

19.2. 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。

19.3. 物理インターフェース

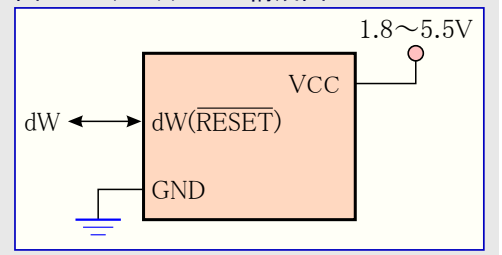
デバッグWIRE許可(DWEN)ヒューズがプログラム(0)され、**施錠ビット**が非プログラム(1)にされると、対象デバイス内のデバッグWIREシステムが活性(有効)にされます。RESETポートピンはプルアップ許可のANDタイ(オープンドレイン)双方向I/Oピンとして設定され、対象デバイスとエミュレータ間の通信路になります。

図19-1.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**で選択したクロック元です。

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以下が厳守されなければなりません。

- dW/(RESET)線のプルアップ抵抗は10kΩよりも小さくしてはなりません。プルアップ抵抗はデバッグWIRE機能に関して必要条件ではありません。
- RESETピンのVCCへの直接的な接続では動作しません。
- RESETピンに接続され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。

図19-1. デバッグWIRE構成図



19.4. 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。Atmel 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き、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Atmel 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。

19.5. 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レジスタにアクセスする間、注意が払われなければなりません。

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。

19.6. デバッグWIRE用レジスタ

次項はデバッグWIREで使うレジスタを記述します。

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

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。

20. フラッシュ メモリのプログラミング

本デバイスはMCU自身によるプログラム コードのダウンロードとアップロード用の自己プログラミング機構を提供します。自己プログラミング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ために、利用可能なデータ インターフェース(例えば、LIN, UART, へ)と関連する規約のどれもが使えます。

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は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、その後に変更して書かれます。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)機能をデバイスが提供します。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。

20.1. フラッシュ メモリの自己プログラミング

20.1.1. 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'00000011'を書き、SPMCSR書き込み後4クロック周期内にSPM命令を実行してください。R1とR0のデータは無視されます。ページ アドレスはZポインタのPCPAGEに書かれなければなりません。この操作中、Zポインタの他のビットは無視されます。

- ・ページ消去中、CPUは停止されます。

20.1.2. 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用されます。一時緩衝部はページ書き込み操作後、またはSPMCSRのCTPBビット(=1)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。

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。

20.1.3. 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'00000101'を書き、SPMCSR書き込み後4クロック周期内にSPM命令を実行してください。R1とR0のデータは無視されます。ページ アドレスは(Zポインタの)PCPAGEに書かれなければなりません。この操作の間、Zポインタの他のビットは0を書かれなければなりません。

- ・ページ書き込み中、CPUは停止されます。

20.2. 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

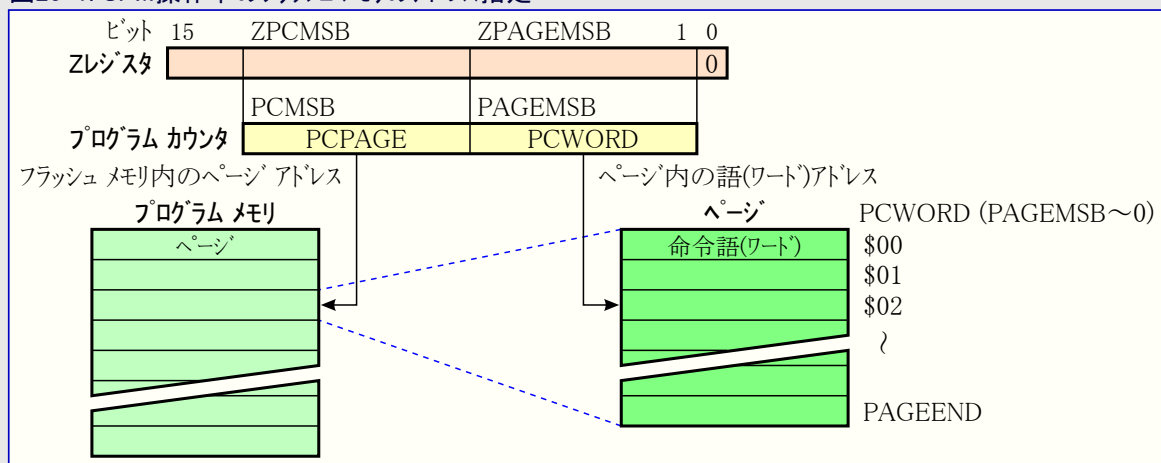
Zポインタ(レジスタ)はSPM命令でのアドレス指定に使われます。Zポインタはレジスタ ファイル内のZHとZLのZレジスタから成ります。実際に使われるビット数は実装依存です。

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(154頁の表21-7.参照)、プログラム カウンタ(アドレス ポインタ)は2つの違う領域を持つように取り扱われます。1つの領域は下位側ビットから成り、ページ内の語(ワード)をアドレス指定し、一方上位側ビットはそのページをアドレス指定します。これは次頁の図20-1.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。

LPM命令は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20-1. 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154頁の表21-7.で一覧されます。

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

このレジスタはプログラムメモリ操作を制御するために必要とする制御ビットを含みます。

ビット	7	6	5	4	3	2	1	0	
\$37 (\$57)	–	RWWSB	SIGRD	CTPB	RFLB	PGWRT	PGERS	SPMEN	SPMCSR
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 – Res : 予約 (Reserved Bits)

このビットは予約されており、常に0として読まれます。

● ビット6 – RWWSB : RWW領域多忙 (Read-While-Write Section Busy)

このビットは書き中読み可(Read-While-Write)を支援するデバイスとの互換用です。ATtiny87/167では常に0として読みます。

● 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149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。

● ビット4 – CTPB : ページ一時緩衝部消去 (Clear Temporary Page Buffer)

ページ一時緩衝部を満たしている間にCTPBビットが1を書かれると、ページ一時緩衝部は消去され、データが失われます。

● ビット3 – RFLB : ヒューズ/施錠ビット読み込み (Read Fuse and Lock Bits)

SPMCSRでRFLBとSPMENが設定(1)された後の3クロック周期内のLPM命令は(ZポイントのZ0に依存して)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149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ポイント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、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ポイント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。ページ全体の消去中、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、SPM命令を許可します。このビットがSIGRD,CTPB,RFLB,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ポイント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: 何時も1つのSPM命令だけが活動すべきです。

20.2.2. 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書き込み動作中に妨げられます。使用者はEEPROM制御レジスタ(EECR)のEEPROMプログラム許可(EEPE)ビットを検査し、SPM命令制御/状態レジスタ(SPMCSR)へ書く前にこのビットが解除(0)されているのを確認することが推奨されます。

20.2.3. ソフトウェアからのヒューズビットと施錠ビットの読み出し

ソフトウェアからヒューズと施錠ビットの両方を読むことが可能です。施錠ビットを読むにはZポインタに\$0001を設定し、SPMCSRのSPMENとヒューズ/施錠ビット読み込み(RFLB)ビットを設定(1)してください。SPMENとRFLBビットがSPMCSRに設定された後3CPU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RFLBビットは施錠ビット読み出しの完了で、または3CPU周期内にLPM命令が実行されないか、または4CPU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RFLB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 (Z=\$0001)	-	-	-	-	-	-	LB2	LB1

ヒューズ下位バイトを読む手順は上記の施錠ビット読み出しと同様です。ヒューズ下位バイトを読み出すにはZポインタに\$0000を設定してSPMCSRのSPMENとRFLBビットを設定(1)してください。SPMENとRFLBビットがSPMCSRに設定された後3CPU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バイト(FLB)の値が転送先レジスタに格納されます。ヒューズ下位バイトの配置と詳細な記述については153頁の表21-5を参照してください。

ビット	7	6	5	4	3	2	1	0
Rd (Z=\$0000)	FLB7	FLB6	FLB5	FLB4	FLB3	FLB2	FLB1	FLB0

同様に、ヒューズ上位バイトを読むにはZポインタに\$0003を設定してください。SPMENとRFLB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バイト(FHB)の値が転送先レジスタに格納されます。ヒューズ上位バイトの配置と詳細な記述については153頁の表21-4を参照してください。

ビット	7	6	5	4	3	2	1	0
Rd (Z=\$0003)	FHB7	FHB6	FHB5	FHB4	FHB3	FHB2	FHB1	FHB0

ヒューズ拡張バイトはZポインタに\$0002を設定することによって読むことができます。SPMENとRFLB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拡張バイト(FEB)の値が転送先レジスタに格納されます。ヒューズ拡張バイトの配置と詳細な記述については153頁の表21-3を参照してください。

ビット	7	6	5	4	3	2	1	0
Rd (Z=\$0002)	-	-	-	-	-	-	-	EFB0

プログラム(0)されたヒューズと施錠ビットは0として読めます。非プログラム(1)にされたヒューズと施錠ビットは1として読めます。

20.2.4. 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20-1.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、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3CPU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、または3CPU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は命令一式手引書で記述されるように動作します。

表20-1. 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
8MHz RC発振器校正值	\$0001
温度感知器変位(TSOFFSET)	\$0003
温度感知器利得(TSGAIN)	\$0005

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。

20.2.5. 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)。

- 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。

20.2.6. 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20-2.**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20-2. 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

注: 最小と最大の時間は(項目の)個別操作毎に対してです。

20.2.7. アセンブリ言語による簡単なブート ロータ例

ATtiny87/167ではRWWSBビットが常に0として読まれることに注意してください。それでも書き中読み可(Read-While-Write)を支援するデバイスとの互換性を保証するため、コード例で示されるようにこのビットを検査することが推奨されます。

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。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、レジスタ使用はコード量を犠牲にすれば最適化できます。割り込みが禁止される前提です。

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

21. メモリ フログラミング

21.1. フログラム メモリとデータ メモリ用施錠ビット

ATtiny87/167は非プログラム(1)のままか、表21-2.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2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。ATtiny87/167は独立したブート ロード領域を持ちません。

表21-1. 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)
—	5		1 (非プログラム)
—	4		1 (非プログラム)
—	3		1 (非プログラム)
—	2		1 (非プログラム)
LB2	1	フラッシュとEEPROMメモリに対する一般保護用施錠ビット	1 (非プログラム)
LB1	0		1 (非プログラム)

注: 0はプログラム、1は非プログラムを意味します。

表21-2. 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 ビットが固定されます。(注1)

注1: 施錠ビットを書く前にヒューズ 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: 0はプログラム、1は非プログラムを意味します。

21.2. ヒューズ ビット

ATtiny87/167には3つのヒューズ バイトがあります。表21-3.~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。

SELFPRGENヒューズがプログラム(0)されると、SPM命令はフラッシュ メモリ全体に対して許可され、さもなければ禁止されます。

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21-3. ヒューズ 拡張バイト一覧

名称	ビット	意味	既定値
—	7~1		1 (非プログラム)
SELFPRGEN	0	自己プログラミング機能許可。	1 (非プログラム) 自己プログラミング不許可

表21-4. ヒューズ 上位バイト一覧

名称	ビット	意味	既定値
RSTDISBL	7	PB7がI/OピンかまたはRESETピンかを選択します。 (注1)	1 (非プログラム) PB7はRESETピン
DWEN	6	デバッグWIRE機能許可。	1 (非プログラム) デバッグWIRE不許可
SPIEN	5	直列プログラミング許可。 (注2)	0 (プログラム) 直列プログラミング許可
WDTON	4	ウォッチドッグ タイマ常時有効。 (注3)	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
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4)	1 (非プログラム)
BODLEVEL0	0		1 (非プログラム)

注1: RSTDISBLヒューズの記述については59頁の「ポートBの交換機能」をご覧ください。

注2: 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3: 詳細については44頁の「WDTCSR - ウォッチドッグ タイマ制御レジスタ」をご覧ください。

注4: BODLEVELヒューズの符号化については166頁の表22-5.をご覧ください。

表21-5. ヒューズ 下位バイト一覧

名称	ビット	意味	既定値
CKDIV8	7	システム クロック 8分周選択。 (注4)	0 (プログラム) 8分周
CKOUT	6	システム クロック出力許可。 (注3)	1 (非プログラム) 不許可
SUT1	5	起動時間選択。 (注1)	1 (非プログラム)
SUT0	4		0 (プログラム)
CKSEL3	3	クロック種別選択。 (注2)	0 (プログラム)
CKSEL2	2		0 (プログラム)
CKSEL1	1		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23頁の表4-4.をご覧ください。

注2: CKSEL3~0の既定設定は8MHz校正付き内蔵発振器になります。詳細については22頁の表4-1.をご覧ください。

注3: CKOUTはポートB0に出力するのをシステム クロックに許します。詳細については25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30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。

21.2.1. 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ます。

21.3. 識票バイト

全てのMicrochipマイクロコントローラはデバイスを識別する3バイトの識票符号を持ちます。この符号は直列と並列の両プログラミング動作で、またデバイスが施錠されても読めます。この3バイトは分離された空間に存在します。

表21-6. 識票バイト

デバイス	アドレス	値	識票バイト説明
ATtiny87	0	\$1E	Microchipによる製造を示します。
	1	\$93	8Kバイトフラッシュメモリを示します。
	2	\$87	アドレス1の内容が\$93の時にATtiny87デバイスを示します。
ATtiny167	0	\$1E	Microchipによる製造を示します。
	1	\$94	16Kバイトフラッシュメモリを示します。
	2	\$87	アドレス1の内容が\$94の時にATtiny167デバイスを示します。

21.4. 校正バイト

ATtiny87/167は内蔵発振器用に1バイトの校正値を持ちます。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。

21.5. ページ容量

表21-7. 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
ATtiny87	4K語(8Kバイト)	64語	PC5~0	64	PC11~6	11
ATtiny167	8K語(16Kバイト)	64語	PC5~0	128	PC12~6	12

表21-8. 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
ATtiny87	512バイト	4バイト	EEA1~0	128	EEA8~2	8
ATtiny167						

21.6. 並列プログラミング

この項はATtiny87/167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。

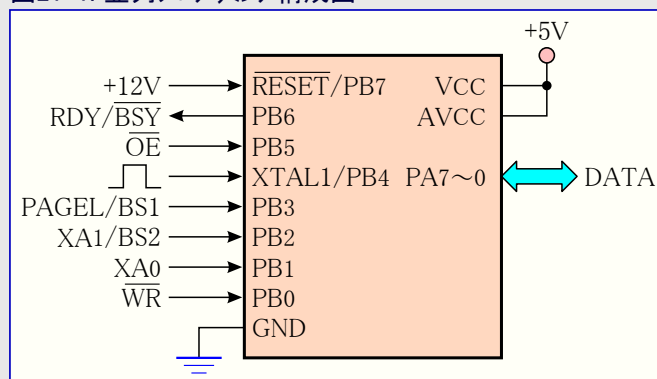
21.6.1. 信号名

本項でATtiny87/167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21-1**と**表21-9**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21-11**で示されます。

\overline{WR} または \overline{OE} パルスを送るとき、設定した指令が実行される動作を決めます。各種指令は**表21-12**で示されます。

図21-1. 並列プログラミング構成図



注: $VCC - 0.3V < AVCC < VCC + 0.3V$ ですが、AVCCは常に4.5~5.5V内にすべきです。

表21-9. 信号名とピン名の関係

信号名	ピン名	入出力	機能
WR	PB0	出力	書き込みパルス(負論理)
XA0	PB1	入力	XTAL1動作ビット0
XA1/BS2	PB2	入力	<ul style="list-style-type: none"> XTAL1動作ビット1 上位/下位バイト選択2 (0:下位, 1:上位) (ヒューズビット用)
PAGEL/BS1	PB3	入力	<ul style="list-style-type: none"> 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 上位/下位バイト選択1 (0:下位, 1:上位) (一般用)
\overline{OE}	PB4	入力	XTAL1クロック入力
RDY/BSY	PB6	入力	出力許可(負論理)
+12V	PB7	入力	0(Low): 多忙(プログラミング中) 1(High): 準備可(指令受付可)
DATA	PA7~0	入出力	<ul style="list-style-type: none"> リセット(Low活性) 並列プログラミング動作(+12V)
			双方向データバス (\overline{OE} =Low時出力)

表21-10. プログラミング動作移行時のピン値

ピン名	シンボル	値
PAGEL/BS1	Prog_enable[3]	0
XA1/BS2	Prog_enable[2]	0
XA0	Prog_enable[1]	0
WR	Prog_enable[0]	0

表21-11. XA0とXA1の符号化(機能)

XA1	XA0	XTAL1パルス時の動作
0	0	フラッシュまたはEEPROMのアドレス設定 (上位/下位はBS1で指定)
0	1	データ設定 (フラッシュの上位/下位はBS1で指定)
1	0	指令設定
1	1	アイドル (動作なし)

表21-12. 指令バイトのビット符号化

指令バイト	指令の機能
\$80 (1000 0000)	チップ消去
\$40 (0100 0000)	ヒューズビット書き込み
\$20 (0010 0000)	施錠ビット書き込み
\$10 (0001 0000)	フラッシュメモリ書き込み
\$11 (0001 0001)	EEPROM書き込み
\$08 (0000 1000)	識票バイト、校正バイト読み出し
\$04 (0000 0100)	ヒューズビット、施錠ビット読み出し
\$02 (0000 0010)	フラッシュメモリ読み出し
\$03 (0000 0011)	EEPROM読み出し

21.7. 並列プログラミング手順

21.7.1. 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(高電圧)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。
- ③ 155頁の表21-10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。

21.7.2. 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。

21.7.3. チップ消去

チップ消去はフラッシュメモリ、EEPROM、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。

1. 「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。

注: 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。

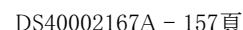
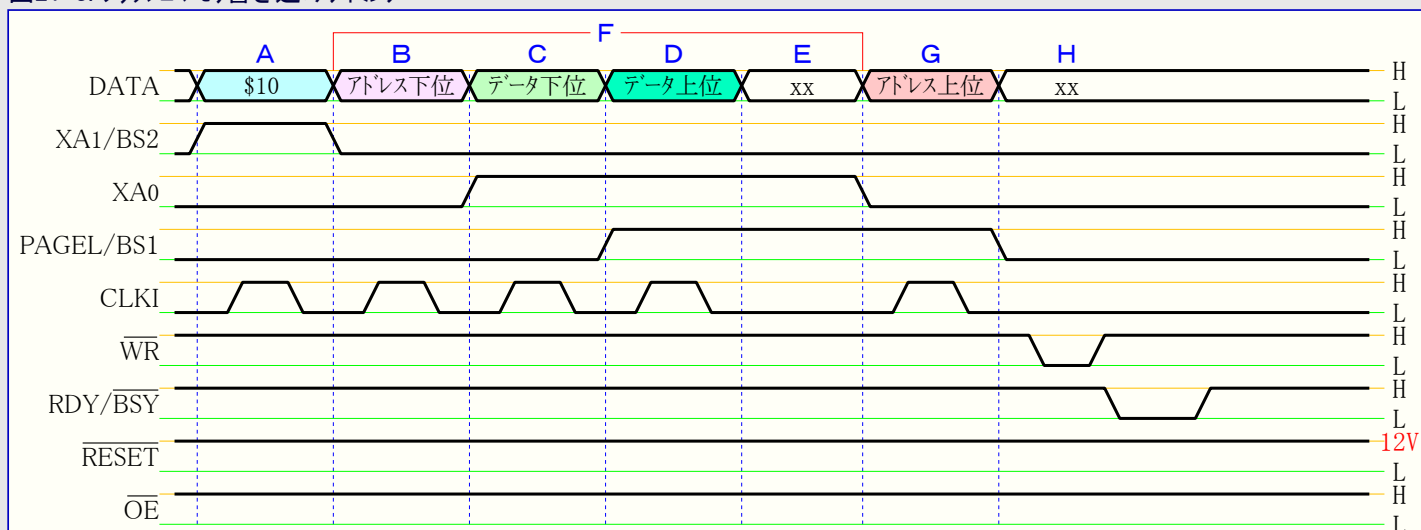


図21-3. フラッシュ メリ書き込みタイミング



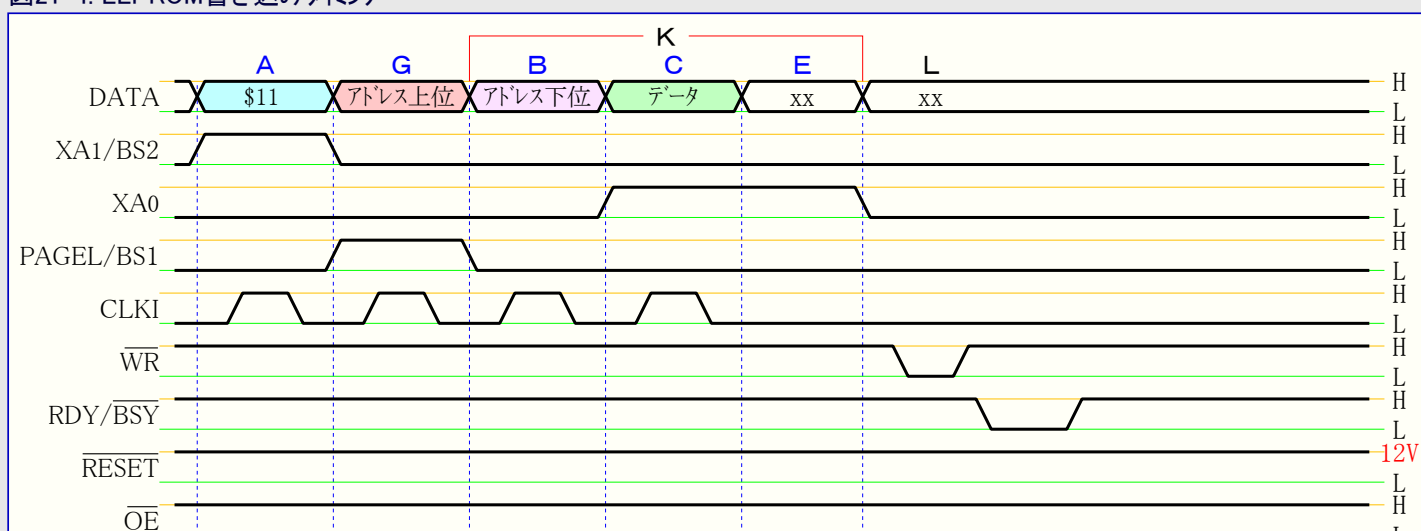
注: xx値は無関係です。A～Hは前記プログラミングを参照してください。

21.7.5. EEPROM書き込み

EEPROMはページで構成されます(154頁の表21-8参照)。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157頁の「フラッシュ メリの書き込み」を参照。図21-4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 メリ書き込み」のAを参照)
2. アドレス上位バイト(\$00～\$01)を設定します。(「フラッシュ メリ書き込み」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 メリ書き込み」のEを参照)
- K. 緩衝部全体が満たされるまで3～5を繰り返します。
- L. EEPROMページ書き込み
 - ① BS1をLow(0)に設定します。
 - ② WRに負パルスを与えます。これはEEPROMページ書き込みを開始します。RDY/BSYがLow(0)になります。
 - ③ 次のページを書く前に、RDY/BSYがHigh(1)になるまで待ちます。

図21-4. EEPROM書き込みタイミング



21.7.6. フラッシュ メリ読み出し

フラッシュ メ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157頁の「フラッシュ メリの書き込み」を参照)

1. フラッシュ メリ読み出し指令\$02(0000 0010)を設定します。(「フラッシュ メリ書き込み」のAを参照)
2. アドレス上位バイト(\$00～\$0F/\$1F)を設定します。(「フラッシュ メリ書き込み」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 メリ書き込み」のBを参照)
4. BS1をLow(0)、 \overline{OE} をLow(0)に設定します。フラッシュ メリ語(ワード)の下位バイトが直ぐにDATAで読めます。
5. BS1をHigh(1)に設定します。フラッシュ メリ語(ワード)の上位バイトが直ぐにDATAで読めます。
6. \overline{OE} をHigh(1)に設定します。DATAはHi-Zになります。

21.7.7. EEPROM読み出し

データ用EEPROMの読み出し方法は次のとおりです。(指令とアドレス設定の詳細については157頁の「フラッシュ メリの書き込み」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 メリ書き込み」のAを参照)
2. アドレス上位バイト(\$00～\$01)を設定します。(「フラッシュ メリ書き込み」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 メリ書き込み」のBを参照)
4. BS1をLow(0)、 \overline{OE} をLow(0)に設定します。EEPROMのバイト データが直ぐにDATAで読めます。
5. \overline{OE} をHigh(1)に設定します。DATAはHi-Zになります。

21.7.8. ヒューズ ビット書き込み (訳注:原書での拡張/上位/下位ヒューズ'項を1つに纏めました。)

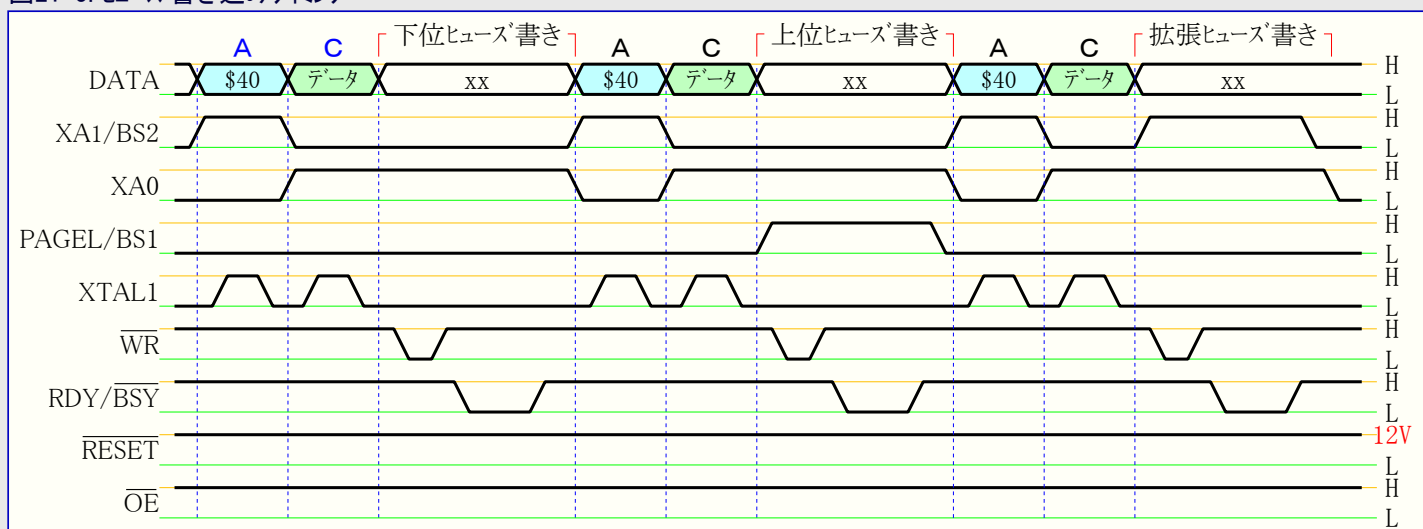
各ヒューズ バイトの書き込み方法は次のとおりです。(指令とデータ設定の詳細については157頁の「フラッシュ メリの書き込み」を参照)

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 メリ書き込み」のAを参照)
2. データ下位バイトを設定します。0=プログラム,1=非プログラム(消去)です。(「フラッシュ メリ書き込み」のCを参照)
3. BS1とBS2を右表の目的バイトに対応する設定にします。
4. \overline{WR} に負パルスを与え、RDY/BSYがHighになるまで待ちます。
5. 3.でHighに設定したBS1、BS2をLow(0)に戻します。これはデータ下位バイトを選択します。

表A. ヒューズ バイト対応BS1,BS2設定

ヒューズ バイト	BS1	BS2
拡張バイト	Low(0)	High(1)
上位バイト	High(1)	Low(0)
下位バイト	Low(0)	Low(0)

図21-5. ヒューズ書き込みタイミング



21.7.9. 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157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施錠ビットはプログラミングできません。(「フラッシュメモリの書き込み」のCを参照)
3. \overline{WR} に負パルスを与え、RDY/BSYがHighになるまで待ちます。

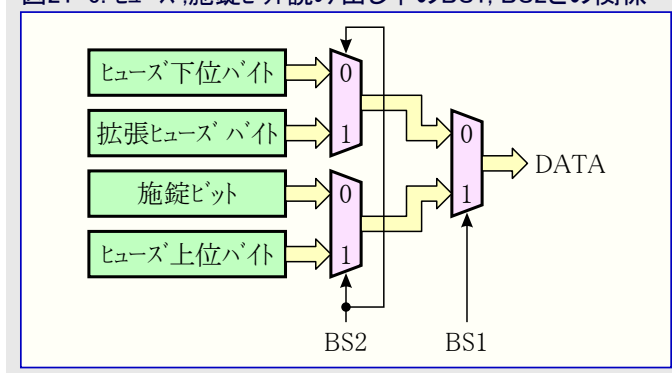
施錠ビットはチップ消去の実行によってのみ解除(1)できます。

21.7.10. 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157頁の「フラッシュメモリの書き込み」を参照)

1. ヒューズビットと施錠ビットの読み出し指令\$04(0000 0100)を設定します。(「フラッシュメモリの書き込み」のAを参照)
2. BS1とBS2をLow(0)、 \overline{OE} をLow(0)に設定します。ヒューズ下位ビットの状態が直ぐにDATAで読めます。(0=プログラム)
3. BS1とBS2をHigh(1)、 \overline{OE} をLow(0)に設定します。ヒューズ上位ビットの状態が直ぐにDATAで読めます。(0=プログラム)
4. BS1をLow(0)、BS2をHigh(1)、 \overline{OE} をLow(0)に設定します。拡張ヒューズビットの状態が直ぐにDATAで読めます。(0=プログラム)
5. BS1をhigh(1)、BS2をLow(0)、 \overline{OE} をLow(0)に設定します。施錠ビットの状態が直ぐにDATAで読めます。(0=プログラム)
6. \overline{OE} をHigh(1)に設定します。DATAはHi-Zになります。

図21-6. ヒューズ、施錠ビット読み出し中のBS1, BS2との関係



21.7.11. 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157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。選択した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。

21.7.12. 8MHz RC発振器校正バイト読み出し

8MHz RC発振器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157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。8MHz RC発振器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。

21.7.13. 温度感知器パラメータバイト読み出し

温度感知器パラメータ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157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\$03または\$05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。温度感知器パラメータバイトが直ぐに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。

21.8. 直列プログラミング

フラッシュメモリとEEPROMの両方はRESETがGNDに引かれている間に直列SPIバスを使ってプログラミングできます。この直列インターフェースはSCK入力、MOSI入力、MISO出力から成ります。RESETがLowに設定された後、書き込み/消去操作が実行され得るのに先立って**プログラミング許可命令**が初めに実行されることを必要とします。

注: 表21-13.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。

表21-13. 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
MISO	PA2	出力	直列データ出力
MOSI	PA4	入力	直列データ入力
SCK	PA5	入力	直列クロック

EEPROMをプログラミングする時に自動消去期間が自動書き込み動作内に組み入れられるので(直列プログラミングのみ)、**チップ消去命令**を初めに実行する必要がありません。チップ消去操作はプログラム(フラッシュメモリ)とEEPROM両方の全てのメモリ位置の内容を\$FFにします。

CKSELヒューズによる有効なクロックが与えられなければなりません。直列クロック(SCK)入力のLowとHighの区間の最小値は次のように定義されます。

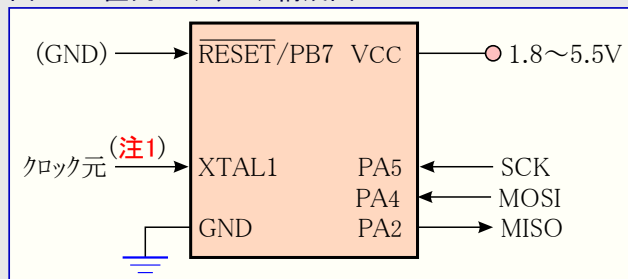
$f_{CK} < 12\text{MHz}$: Low区間 > 2CPUクロック周期

$f_{CK} < 12\text{MHz}$: High区間 > 2CPUクロック周期

$f_{CK} \geq 12\text{MHz}$: Low区間 > 3CPUクロック周期

$f_{CK} \geq 12\text{MHz}$: High区間 > 3CPUクロック周期

図21-7. 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。

21.8.1. 直列プログラミング手順

ATtiny87/167への直列データ書き込み時、データはSCKの上昇端でクロック駆動されます。ATtiny87/167からのデータ読み込み時、データはSCKの下降端でクロック駆動されます。タイミングの詳細については163頁の図21-9をご覧ください。

直列プログラミング動作でのATtiny87/167のプログラミングと照合は次手順が推奨されます(表21-15.の命令一式参照)。

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2CPU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、MOSIピンに**プログラミング許可命令**を送ることによって直列プログラミングを許可してください。

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

4. フラッシュメモリは1ページ単位で書かれます。ページ容量は154頁の表21-7.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ビットとデータを提供することによって1バイトずつ設定されます。ページが正しく設定されるのを保証するため、与えられたアドレスにデータ上位バイトが適用される前にデータ下位バイトが設定されなければなりません。プログラムメモリのページはアドレスの上位6/7ビットを含む**ページ書き込み命令**の設定によって(フラッシュメモリに)格納されます。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21-14.参照)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提供することによって1バイト単位で書かれます。EEPROMのメモリ位置は新規データが書かれるのに先立って始めて自動的に消去されます。ポーリング(BSY/RDY)が使われない場合、次のバイトを行う前に最低tWD_EEPROM(表21-14.参照)待たなければなりません。

ページ単位: EEPROMの1ページは**EEPROMページ設定命令**と共にアドレスの下位2ビットとデータを提供することによって1バイトずつ設定されます。EEPROMページはアドレスの上位7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21-8.参照)を行う前に最低tWD_EEPROM(表21-14.参照)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択したアドレスの内容を直列出力MISOに読み戻す**読み出し命令**を使うことによって照合できます。

7. プログラミング作業終了時、RESETは通常動作を開始するため、High(1)に設定できます。

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21-14. ヒューズ、フラッシュ、EEPROM次位置書き込み前の待機時間

シンボル	最低待機時間	備考
tWD_FUSE	4.5ms	ヒューズ書き込み
tWD_FLASH	4.5ms	フラッシュメモリ書き込み
tWD_EEPROM	4.0ms	EEPROM書き込み
tWD_ERASE	9.0ms	チップ消去

21.8.2. 直列プログラミング命令一式

表21-15と次頁の図21-8.は命令一式を記述します。

表21-15. 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	
識別バイト読み出し	\$30	\$00	アドレス	識別バイト	
ヒューズ下位ビット読み出し	\$50	\$00	\$00	ヒューズ下位	
ヒューズ上位ビット読み出し	\$58	\$08	\$00	ヒューズ上位	
ヒューズ拡張ビット読み出し	\$50	\$08	\$00	ヒューズ拡張	
校正バイト読み出し	\$38	\$00	\$00	校正バイト	
書き込み命令 (注2)					
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	
ヒューズ下位ビット書き込み	\$AC	\$A0	\$00	ヒューズ下位	
ヒューズ上位ビット書き込み	\$AC	\$A8	\$00	ヒューズ上位	
ヒューズ拡張ビット書き込み	\$AC	\$A4	\$00	ヒューズ拡張	

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2: プログラム用メモリにアクセスする命令は語(ワード)アドレスを使います。このアドレスはページ範囲内で乱順にできます。

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。

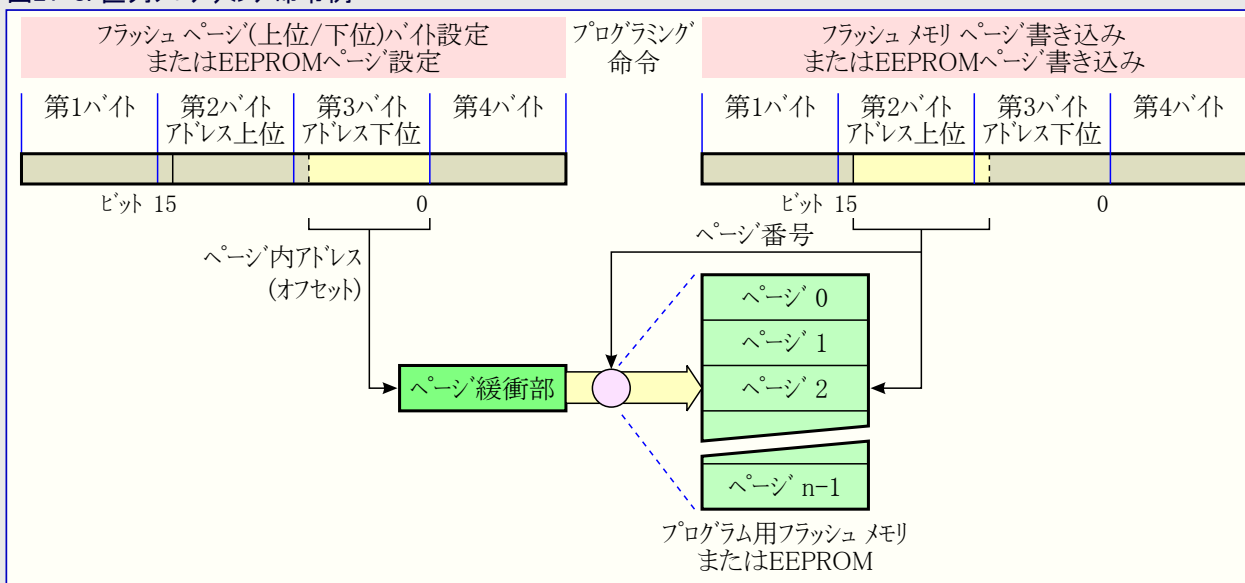
- ヒューズ、施錠ビット、識別バイト、校正バイト、ページ容量については対応項目を参照してください。
- プログラミングと書き込み器に関する応用記述については <https://www.microchip.com> をご覧ください。
- 第4バイトの赤背景はホスト読み込み(デバイス出力)を示します。

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。次頁の図21-8.をご覧ください。

図21-8. 直列プログラミング命令例



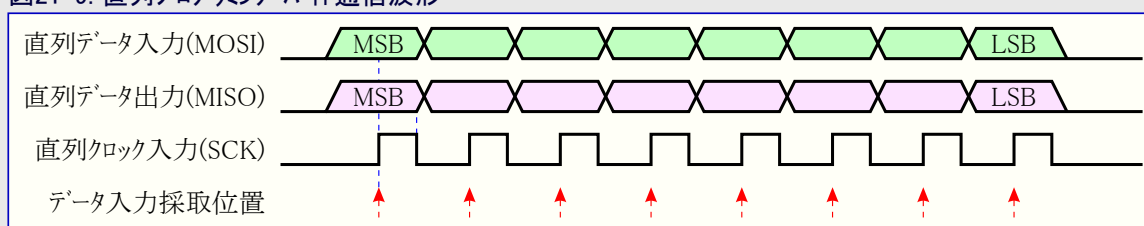
(訳補) フラッシュメモリ、EEPROM、ページ緩衝部内のアドレス(位置)指定に使われるビットはメモリ容量とページ構成に依存します。ATtiny87/167でのこれらの指定方法は次表で要約されます。

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	ATtiny87/167 : L=PC5~0
EEPROMページ内バイト設定	0000 0000	0000 00LL	ATtiny87/167 : L=EEA1~0
フラッシュメモリ読み出し	0000 HHHH 000H HHHH	LLLL LLLL	ATtiny87 : H=PC11~8, L=PC7~0 ATtiny167 : H=PC12~8, L=PC7~0
EEPROM読み出し	0000 000H	LLLL LLLL	ATtiny87/167 : H=EEA8, L=EEA7~0
フラッシュページ書き込み	0000 HHHH 000H HHHH	LLL0 0000	ATtiny87 : H=PC11~8, L=PC7~5 ATtiny167 : H=PC12~8, L=PC7~5
EEPROMバイト書き込み	0000 000H	LLLL LLLL	ATtiny87/167 : H=EEA8, L=EEA7~0
EEPROMページ書き込み	0000 000H	LLLL LL00	ATtiny87/167 : H=EEA8, L=EEA7~2

21.8.3. 直列プログラミング特性

図21-9. 直列プログラミングバイト通信波形



SPI部の特性については、170ページの「SPIタイミング特性」を参照してください。

22. 電気的特性

22.1. 絶対最大定格 (警告)

動作温度	-40°C ~ +125°C
保存温度	-65°C ~ +150°C
RESETを除くピン許容電圧	-0.5V ~ VCC+0.5V
RESETピン許容電圧	-0.5V ~ +13.0V
VCC許容電圧	-0.5V ~ +6.0V
入出力ピン出力電流	40.0mA
消費電流	200.0mA
VCC=0~5Vでの強制注入電流 (注2)	±5.0mA (注1)

(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。

注1: ポート当たりの最大電流=±30mA

注2: 機能不全が起こるかもしれません。

22.2. DC特性

TA=-40°C~85°C, VCC=1.8V~5.5V (特記事項を除く)

シンボル	項目	条件	最小	代表 (注1)	最大	単位
VIL	Lowレベル入力電圧	XTAL1, RESETを除く	-0.5		0.2VCC (注2)	V
VIL1		XTAL1外部クロック駆動	-0.5		0.1VCC (注2)	
VIL2		RESET	-0.5		0.2VCC (注2)	
VIL3		I/OとしてのRESET	-0.5		0.2VCC (注2)	
VIH	Highレベル入力電圧	XTAL1, RESETを除く	0.7VCC (注3)		VCC+0.5	V
VIH1		XTAL1外部クロック駆動	0.8VCC (注3)		VCC+0.5	
VIH2		RESET	0.9VCC (注3)		VCC+0.5	
VIH3		I/OとしてのRESET	0.7VCC (注3)		VCC+0.5	
VOL	Lレベル出力電圧 (ポートA,B) (注4)	IOL=10mA, VCC=5V			0.6	
		IOL=5mA, VCC=3V			0.5	
VOH	Hレベル出力電圧 (ポートA,B) (注5)	IOH=-10mA, VCC=5V	4.3			
		IOH=-5mA, VCC=3V	2.5			
IIL	I/OピンLowレベル入力漏れ電流	VCC=5.5V		<0.05	1	μA
IIH	I/OピンHighレベル入力漏れ電流	確実なH/L範囲		<0.05	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	
ICC	活動動作消費電流 (注6)	VCC=2V, 1MHz		0.3	0.6	mA
		VCC=3V, 4MHz		1.8	2.5	
		VCC=5V, 8MHz		5.5	7.0	
	アイドル動作消費電流 (注6)	VCC=2V, 1MHz		0.05	0.2	mA
		VCC=3V, 4MHz		0.5	0.8	
		VCC=5V, 8MHz		1.8	2.5	
	パワーダウン動作消費電流 (注7)	VCC=3V, WDT有効		5	10	μA
		VCC=3V, WDT禁止		0.15	4	
VACIO	アナログ比較器入力オフセット電圧	VCC=5V, Vin=VCC/2	-10	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 (同相入力VCC/2)	VCC=2.7V		170		ns
		VCC=5.0V		180		

注1: “代表”は25°Cでの代表値です。最大値は特性付けされた値で製造中に限度を検査しません。

注2: “最大”はLowレベルの認識が保証される最高電圧を意味します。

注3: “最小”はHighレベルの認識が保証される最低電圧を意味します。

注4~注7は次頁へ

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($V_{CC}=5V$ で10mA、 $V_{CC}=3V$ で5mA)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。

- ・全IOLの合計が120mA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も関連する仕様を超えます。ピンは検査条件で一覧される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5: 各I/Oポートは安定状態(非過渡時)に於いて検査条件($V_{CC}=5V$ で10mA、 $V_{CC}=3V$ で5mA)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。

- ・全IOHの合計が120mAを超えるべきではありません。

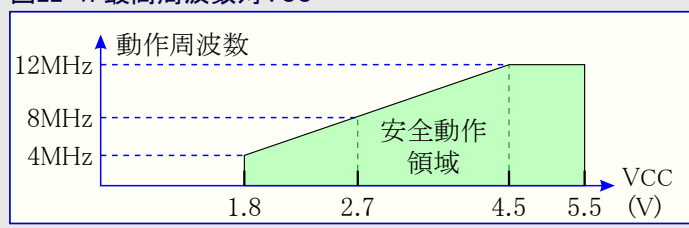
IOHが検査条件を超える場合、VOHも関連する仕様を超えます。ピンは検査条件で一覧されるよりも大きな吐き出し電流を流すことは保証されません。

注6: 値は36頁の「消費電力の最小化」で記述した方法を用いる外部クロックです。電力削減許可(PRR=\$FF)でI/O駆動なしです。

注7: 低電圧検出(BOD)禁止です。

22.3. 速度

図22-1. 最高周波数対 V_{CC}



22.4. クロック特性

22.4.1. 校正付き内蔵RC発振器の精度

工場既定校正よりも高い精度に内蔵発振器を手動校正することが可能です。この発振器周波数が温度と電圧に依存することに注意してください。電圧と温度の特性は181頁の図23-29で得られます。

表22-1. 校正付き内蔵RC発振器の校正精度

校正種別	周波数	V_{CC}	温度	校正精度 (注)
工場校正	8.0MHz	3V	25°C	±10%
使用者校正	7.3~8.1MHz内の一定周波数	1.8~5.5V内の一定電圧	-40~85°C内の一定温度	±1%

注: 校正点での発振器周波数精度(一定温度と一定電圧)

22.4.2. 外部クロック信号駆動

図22-2. 外部クロック駆動波形

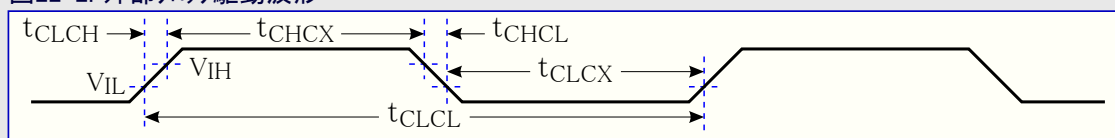


表22-2. 外部クロック駆動特性

シンボル	項目	$V_{CC}=1.8\sim5.5V$		$V_{CC}=2.7\sim5.5V$		$V_{CC}=4.5\sim5.5V$		単位
		最小	最大	最小	最大	最小	最大	
$1/t_{CLCL}$	クロック周波数	0	4	0	8	0	16	MHz
t_{CLCL}	クロック周期	250		125		62.5		
t_{CHCX}	Highレベル時間	100		50		25		ns
t_{CLCX}	Lowレベル時間	100		50		25		
t_{CLCH}	上昇時間		2.0		1.6		0.5	μs
t_{CHCL}	下降時間		2.0		1.6		0.5	
Δt_{CLCL}	隣接クロック周期間の変化率		2		2		2	%

注: 詳細については25頁の「外部クロック信号」を参照してください。

22.5. リセット特性

表22-3. 外部リセット特性

シンボル	項目	条件	最小	代表	最大	単位
V _{RST}	RESETピン閾値電圧	VCC=5V	0.2VCC		0.9VCC	V
t _{RST}	リセットパルス幅		2.5			μs
V _{BG}	基準電圧	VCC=2.7V, TA=25°C	1.0	1.1	1.2	V
t _{BG}	起動時間			40	70	μs
I _{BG}	消費電流			15		μA

表22-4. 電源ONリセット特性

シンボル	項目	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		V
	下降時電源ONリセット閾値電圧 (注)	1.0	1.3	1.6	
V _{PORMAX}	内部電源ONリセット信号を保証するためのVCC最大開始電圧。			0.4	
V _{PORMIN}	内部電源ONリセット信号を保証するためのVCC最小開始電圧。	-0.1			
V _{CCR}	内部電源ONリセットを保証するためのVCC上昇率	0.01			V/ms

注: 上昇に先立って、リセットを保証するために供給はV_{PORMIN}とV_{PORMAX}間でなければなりません。

表22-5. BODLEVELヒューズ (V_{BOT}) 設定 (注1)

BODLEVEL2~0	最小	代表	最大	単位
1 1 1	低電圧検出(BOD)リセット禁止			
1 1 0	1.7	1.8	2.0	V
1 0 1	2.5	2.7	2.9	
1 0 0	4.1	4.3	4.5	
0 0 0 ~ 0 1 1	(予約)			

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。この検査は低動作電圧についてBODLEVEL=101、高動作電圧についてBODLEVEL=100を用いて実行されます。

表22-6. 低電圧検出(BOD)特性

シンボル	項目	最小	代表	最大	単位
V _{HYST}	低電圧検出ヒステシス電圧		80		mV
t _{BOD}	最小低電圧検出時間		2		μs

22.6. 内部基準電圧特性

表22-7. 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=4.5V, TA=25°C	1.0	1.1	1.2	V
t _{BG}	起動時間			40	70	μs
I _{BG}	消費電流			10		μA

22.7. 電流源特性

表22-8. 電流源特性

シンボル	項目	条件	最小	代表	最大	単位
I _{ISRC}	電流	VCC=2.7V/5.5V, TA=140°C/+85°C	94		106	μA
t _{ISRC}	電流源始動時間	VCC=4.5V, TA=25°C		60		μs

22.8. A/D変換器特性

表22-9. A/D変換特性 (TA=-40~+85°C)

	シンボル	項目	条件			最小	代表	最大	単位
シングル エンド 入力 変換		分解能					10		ビット
	TUE	絶対精度					2.0	3.5	
	INL	積分非直線性誤差	VCC=4.5V, VREF=2.56V, 変換クロック=200kHz				0.6	2.0	LSB
	DNL	微分非直線性誤差					0.3	0.8	
		利得誤差				-6.0	-2.5	2.0	
		オフセット(ゼロ)誤差				-3.5	1.5	3.5	
	AREF	外部基準電圧				2.0		AVCC	V
		入力周波数帯域					38.5		kHz
差動 入力 変換		分解能					8		ビット
	TUE	絶対精度		両極	×8利得		1.0	3.0	
					×20利得		1.5	3.5	
				単極	×8利得		2.0	4.5	
					×20利得		2.0	6.0	
	INL	積分非直線性誤差		両極	×8利得		0.2	1.0	
					×20利得		0.4	1.5	
				単極	×8利得		0.5	2.0	
					×20利得		1.6	5.0	
	DNL	微分非直線性誤差	VCC=5V, VREF=4V, 変換クロック=200kHz	両極	×8利得		0.3	0.8	LSB
					×20利得		0.3	0.8	
				単極	×8利得		0.4	0.8	
					×20利得		0.6	1.6	
		利得誤差		両極	×8利得	-3.0	1.0	3.0	
					×20利得	-4.0	1.5	4.0	
				単極	×8利得	-5.0	-2.5	0.0	
					×20利得	-4.0	-0.5	4.0	
		オフセット(ゼロ)誤差		両極, ×20利得		-2.0	0.5	2.0	
				単極, ×20利得		-2.0	0.5	2.0	
	AREF	外部基準電圧				2.0		AVCC-1.0	V
	V _{DIFF}	入力差電圧				-VREF/利得		+VREF/利得	
	V _{IN}	入力電圧				0		AVCC	
		A/D変換出力				-511		+511	LSB
		入力周波数帯域					4		kHz
共通	AVCC	アナログ供給電圧				VCC-0.3		VCC+0.3	V
	V _{INT}	内部1.1V基準電圧				1.0	1.1	1.2	
		内部2.56V基準電圧	VCC>3.0V			2.3	2.56	2.8	
	RREF	基準電圧入力インピーダンス					32		kΩ
	RAIN	アナログ入力インピーダンス					100		MΩ

(訳注) 原書の図22-9と図22-10は図22-9として纏めました。

22.9. 並列プログラミング特性

図22-3. 並列プログラミング タイミング (一般的な必要条件)

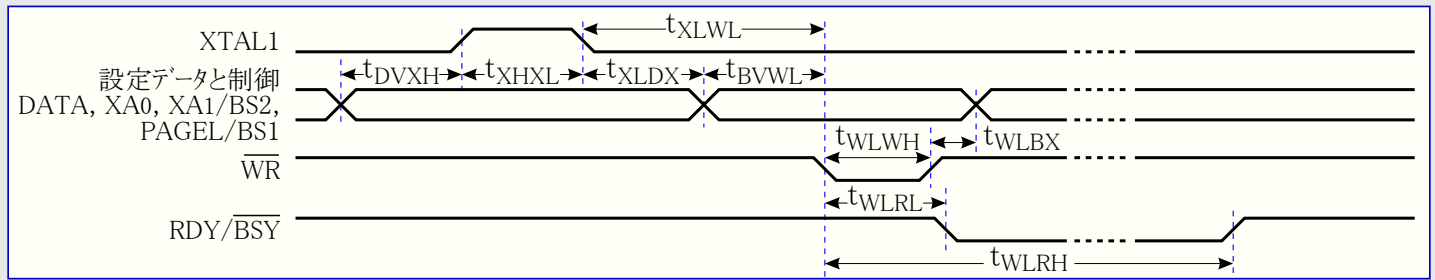
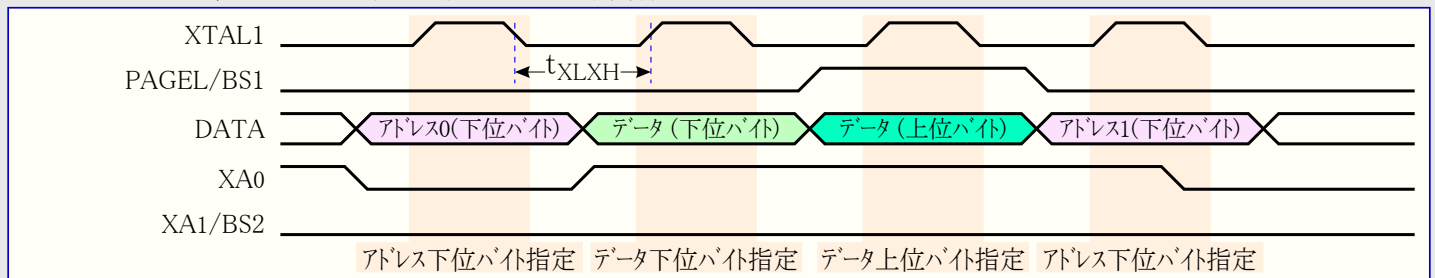
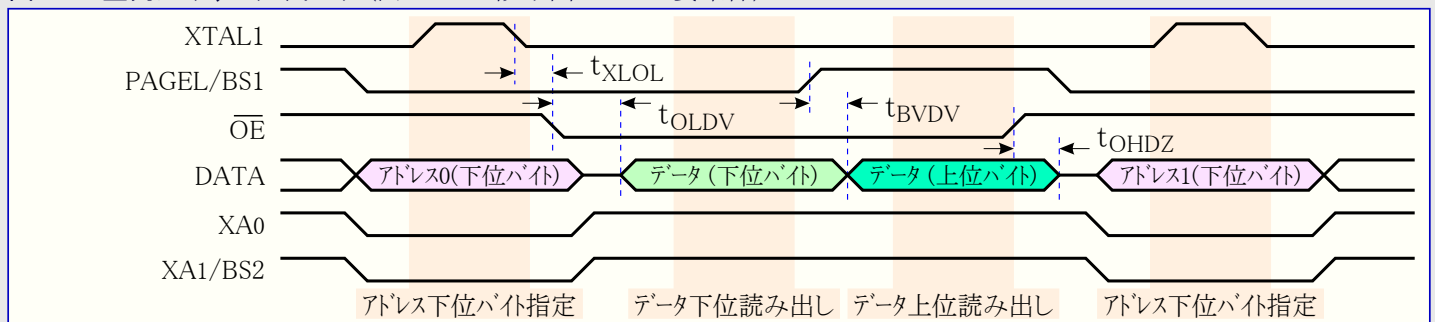


図22-4. 並列プログラミング タイミング (ページ設定での必要条件)



注: 図22-3.で示されるタイミング必要条件(即ち、 t_{DVXH} 、 t_{XHLX} 、 t_{XLDX})は設定操作にも適用されます。

図22-5. 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22-3.で示されるタイミング必要条件(即ち、 t_{DVXH} 、 t_{XHLX} 、 t_{XLDX})は読み出し操作にも適用されます。

表22-11. 並列プログラミング特性 (VCC=5V±10%)

シンボル	項目	最小	代表	最大	単位
V _{PP}	プログラミング許可電圧	11.5		12.5	V
I _{PP}	プログラミング許可電流			250	μA
t _{DVXH}	XTAL1 ↑ に対するデータと制御の準備時間	67			
t _{XLXH}	XTAL1 ↓ から次XTAL1 ↑ までの待機時間	200			
t _{XHXL}	XTAL1 Highパルス幅	150			
t _{XLDX}	XTAL1パルス ↓ 後のデータと制御の保持時間	67			
t _{XLWL}	XTAL1パルス ↓ 後の \overline{WR} ↓ 待機時間	0			
t _{BVPH}	PAGELパルス ↑ に対するBS1準備時間	67			ns
t _{PLBX}	PAGELパルス ↓ 後のBS1保持時間	67			
t _{WLBX}	\overline{WR} パルス ↓ 後のBS1,BS2保持時間	67			
t _{PLWL}	PAGELパルス ↓ 後の \overline{WR} パルス ↓ 待機時間	67			
t _{BVWL}	\overline{WR} パルス ↓ に対するBS1準備時間	67			
t _{WLWH}	\overline{WR} Lowパルス幅	150			
t _{WLRL}	\overline{WR} パルス ↓ 後のRDY/ \overline{BSY} ↓ 遅延時間	0		1	μs
t _{WLRH}	書き込み時間 (\overline{WR} ↓ からRDY/ \overline{BSY} ↑) (注1)	3.7		4.5	ms
t _{WLRH_CE}	チップ消去時間 (\overline{WR} ↓ からRDY/ \overline{BSY} ↑) (注2)	7.5		9	
t _{XLCL}	XTAL1パルス ↓ 後の \overline{OE} ↓ 待機時間	0			
t _{BVDV}	BS1有効からのDATA遅延時間	0		250	ns
t _{OLDV}	\overline{OE} ↓ 後のDATA出力遅延時間			250	
t _{OHdZ}	\overline{OE} ↑ 後のDATA Hi-Z遅延時間			250	

注1: フラッシュメモリ、EEPROM、ヒューズビット、施錠ビット書き込み指令に対して有効です。

注2: チップ消去指令に対して有効です。

22.10. SPIタイミング特性

図22-6. SPI タイミング必要条件 (主装置動作)

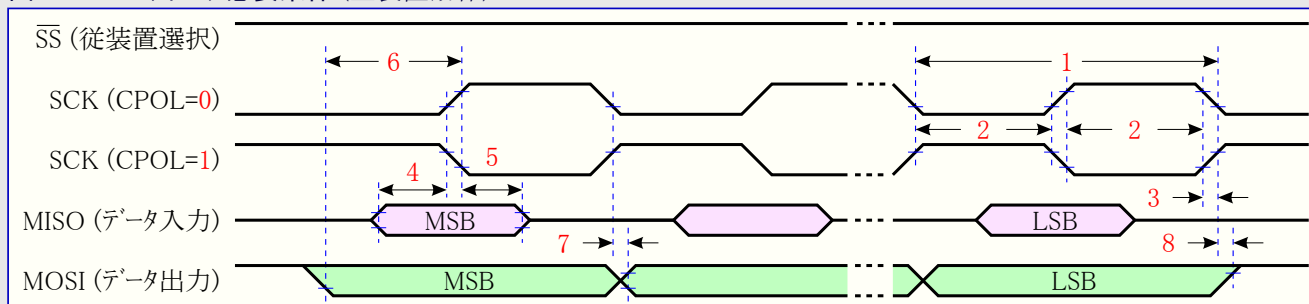


図22-7. SPI タイミング必要条件 (従装置動作)

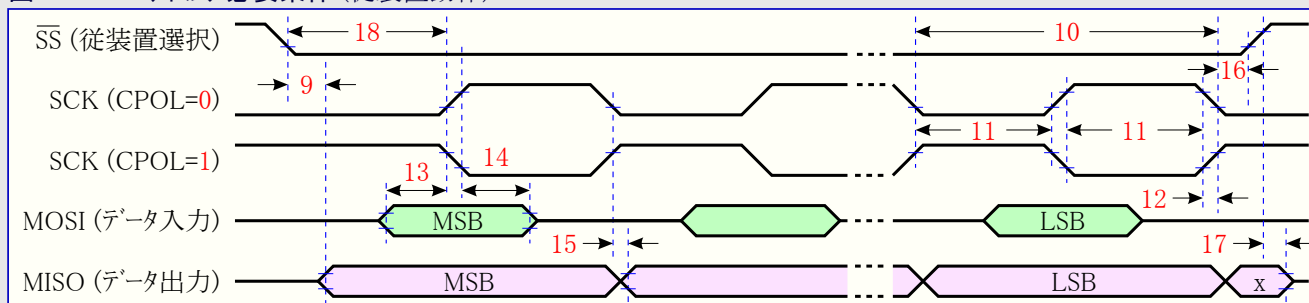


表22-12. SPI タイミング特性

番号	項目	動作種別	最小	代表	最大	単位
1	SCK周期	主装置		表13-4.参照		
2	SCK High/Low期間	主装置		50%デューティ比		
3	SCK上昇/下降時間	主装置		3.6		
4	入力データ 準備時間	主装置		10		
5	入力データ 保持時間	主装置		10		
6	出力からSCK変移時間	主装置		0.5tSCK		ns
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	4tCK			
11	SCK High/Low期間 (注)	従装置	2tCK			
12	SCK上昇/下降時間	従装置			1.6	μs
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	tCK			
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			ns
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	2tCK			

注: SPIプログラミングでの最小SCK High/Low期間は、 $2t_{CLCL}(t_{CK} < 12\text{MHz})$ 、 $3t_{CLCL}(t_{CK} \geq 12\text{MHz})$ です。

23. 代表特性

本項内に含まれたデータは主に同じ製法と設計法の類似デバイスの特徴付けとシミュレーションに基づいています。従って、このデータはデバイスがどう反応するかについての指標として扱われるべきです。

以下の図は代表的な特性を示します。これらの図は製造中に検査されていません。全ての消費電流測定は全I/Oピンを入力として設定した内部プルアップ許可で行われています。電源幅振幅の方形波発振器がクロック源として使われています。

パワーダウン動作での消費電力はクロック選択と無関係です。

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

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

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。

23.1. 活動動作消費電流

図23-1. 活動動作消費電流 対 周波数 (100kHz~1MHz, PRR=\$FF)

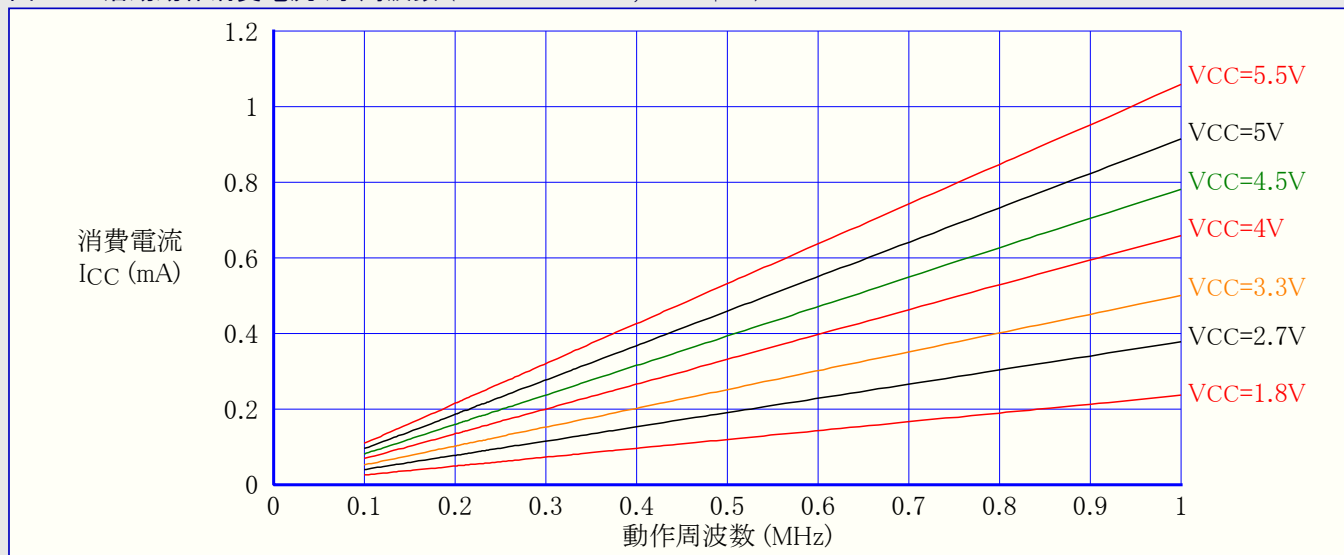


図23-2. 活動動作消費電流 対 周波数 (1MHz~16MHz, PRR=\$FF)

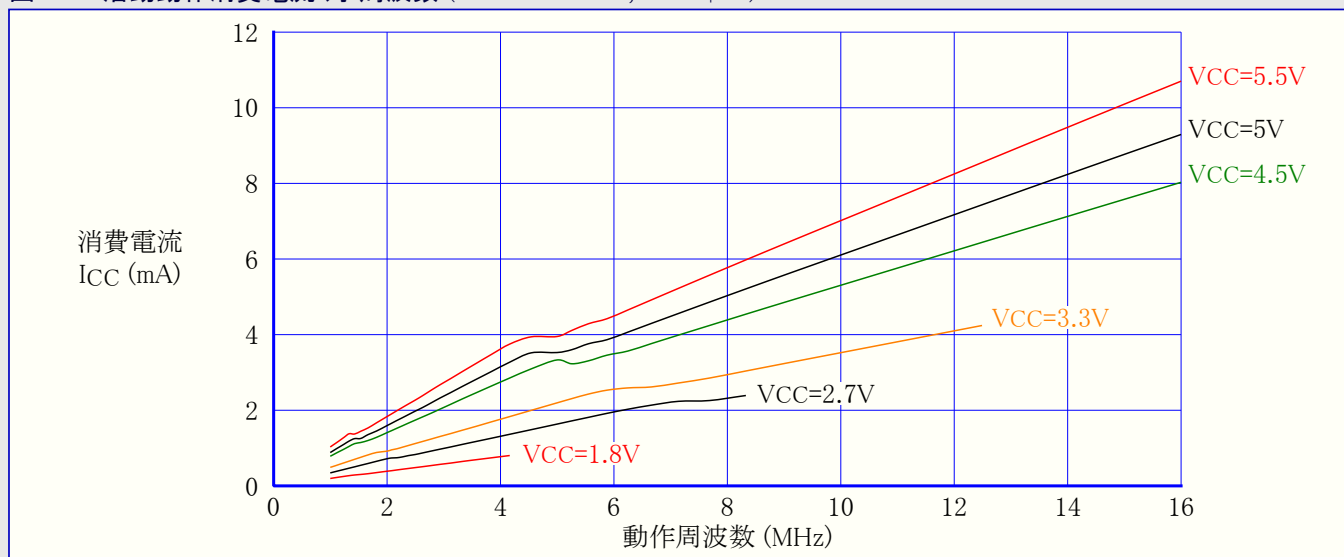


図23-3. 活動動作消費電流 対 動作電圧 (内蔵RC発振器,8MHz)

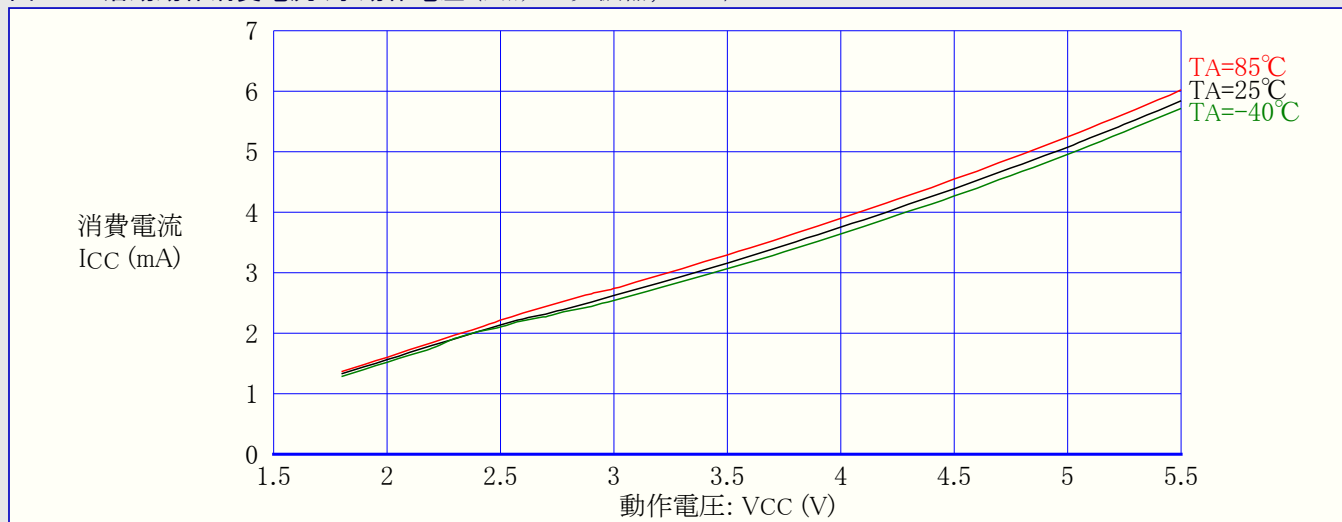


図23-4. 活動動作消費電流 対 動作電圧 (内蔵WDT発振器,128kHz)



23.2. アイドル動作消費電流

図23-5. アイドル動作消費電流 対 周波数 (1MHz~16MHz, PRR=\$FF)

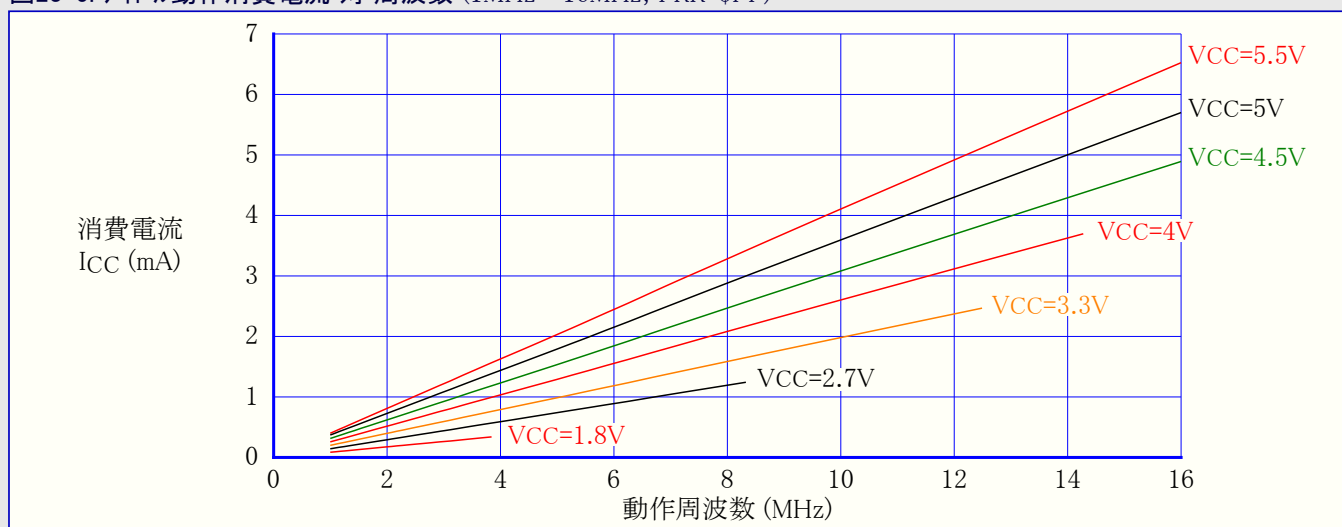


図23-6. アイドル動作消費電流 対 動作電圧 (内蔵RC発振器,8MHz)

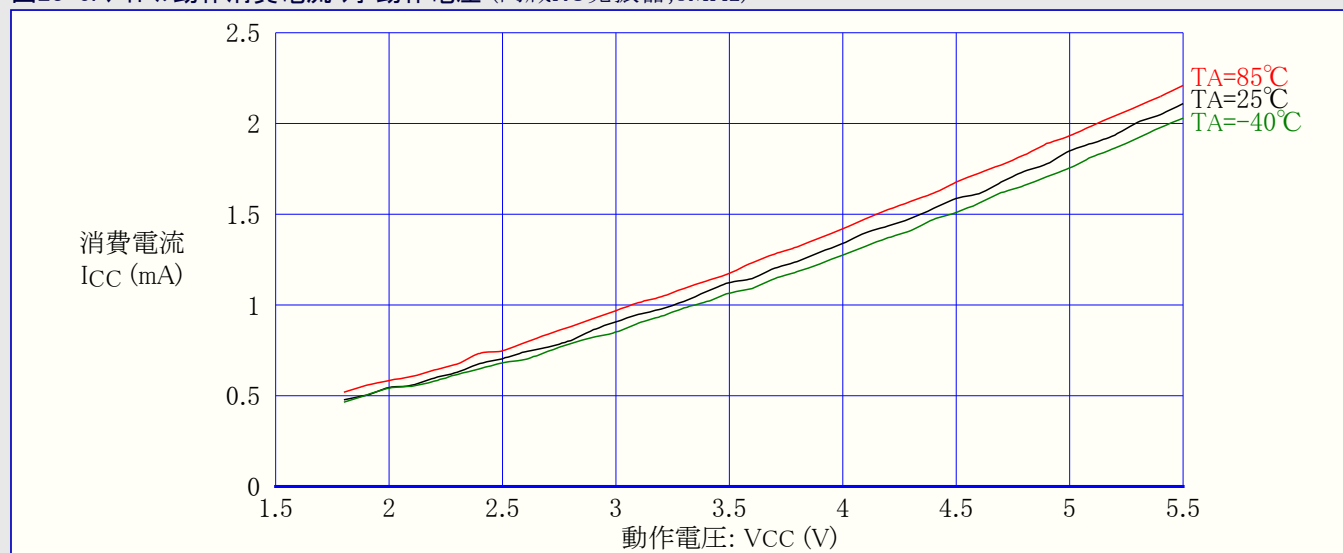
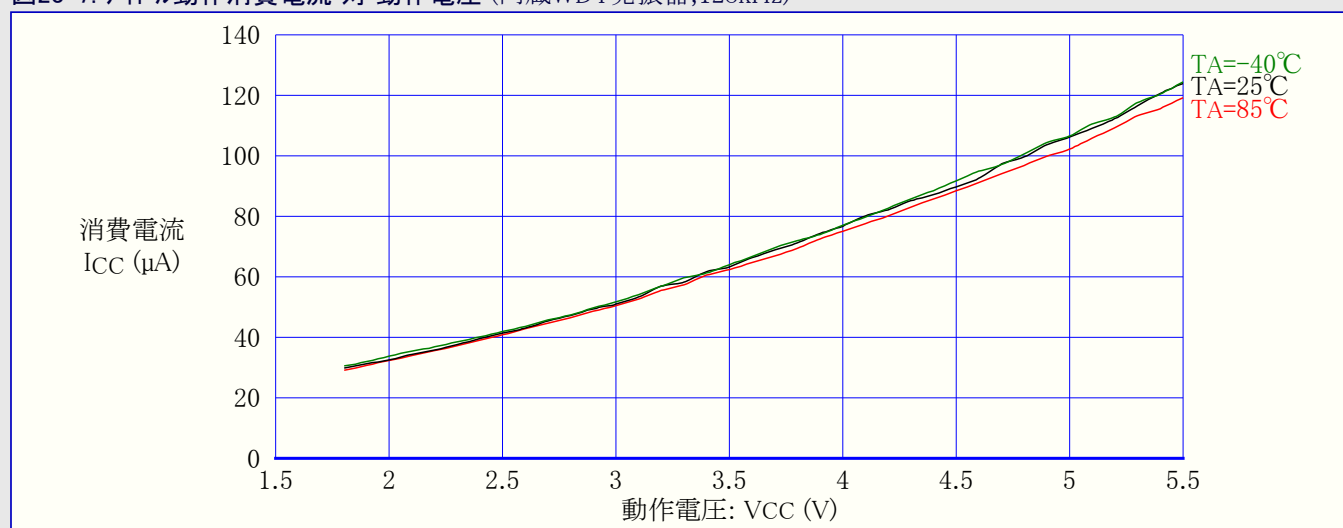


図23-7. アイドル動作消費電流 対 動作電圧 (内蔵WDT発振器,128kHz)



23.3. 入出力単位部の消費(供給)電流

以下の表は各種入出力単位部アイドル動作に関する追加消費電流の計算に使えます。入出力単位部の許可や禁止は電力削減レジスタによって制御されます。詳細については38頁の「5.9.3. PRR – 電力削減レジスタ」をご覧ください。

表23-1. 各種入出力単位部に関する追加消費電流 (絶対値)

単位部	VCC=5V, f=16MHz	VCC=5V, f=8MHz	VCC=3V, f=8MHz	VCC=3V, f=4MHz	単位
LIN/UART	770	370	200	100	μA
SPI	310	140	80	40	
タイマ/カウンタ1	280	130	80	40	
タイマ/カウンタ0	410	200	100	50	
USI	140	50	40	20	
A/D変換器	480	220	100	50	

23.4. パワーダウン動作消費電流

図23-8. パワーダウン動作消費電流 対 動作電圧 (ウォッチドッグ タイマ禁止)

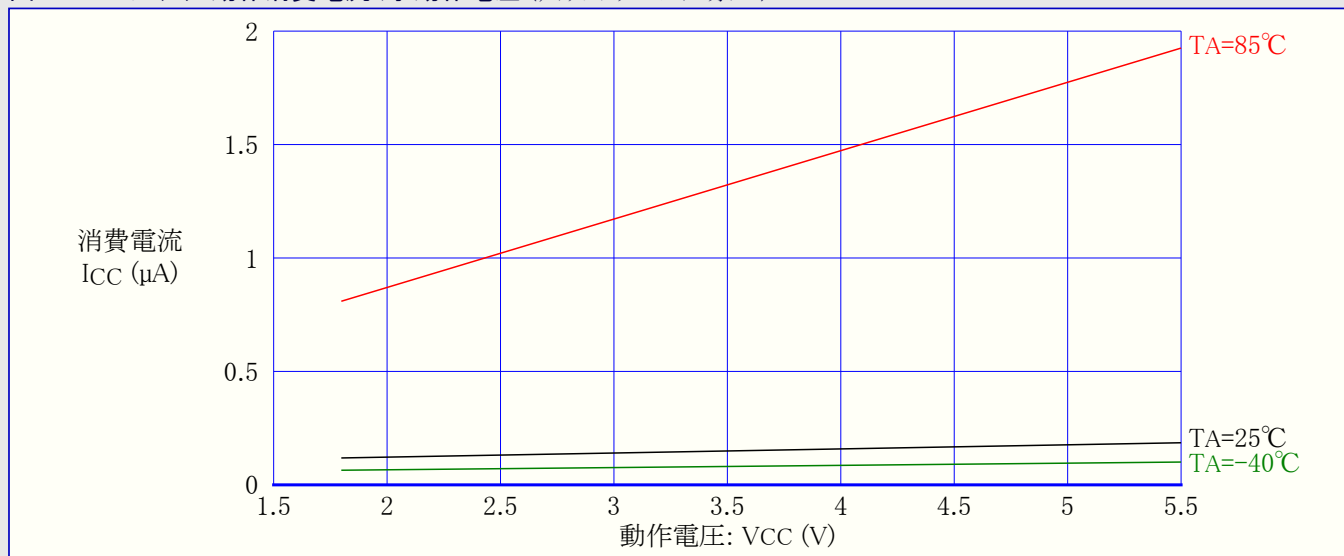
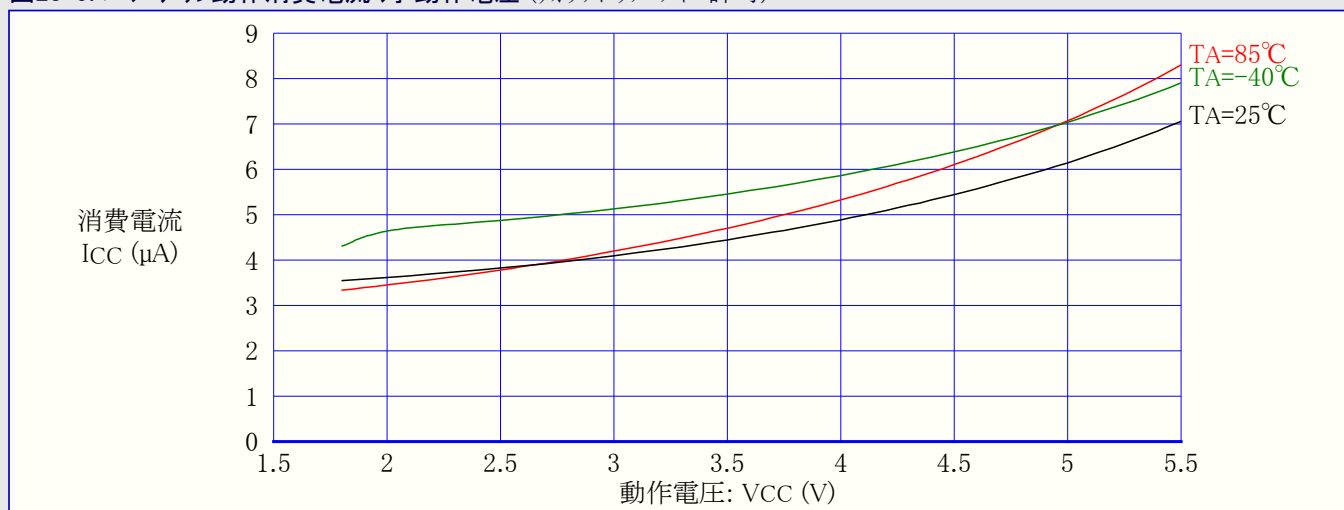


図23-9. パワーダウン動作消費電流 対 動作電圧 (ウォッチドッグ タイマ許可)



23.5. リセット消費電流

図23-10. リセット消費(供給)電流 対 周波数 (100kHz~1MHz、RESETプルアップ電流を除く)

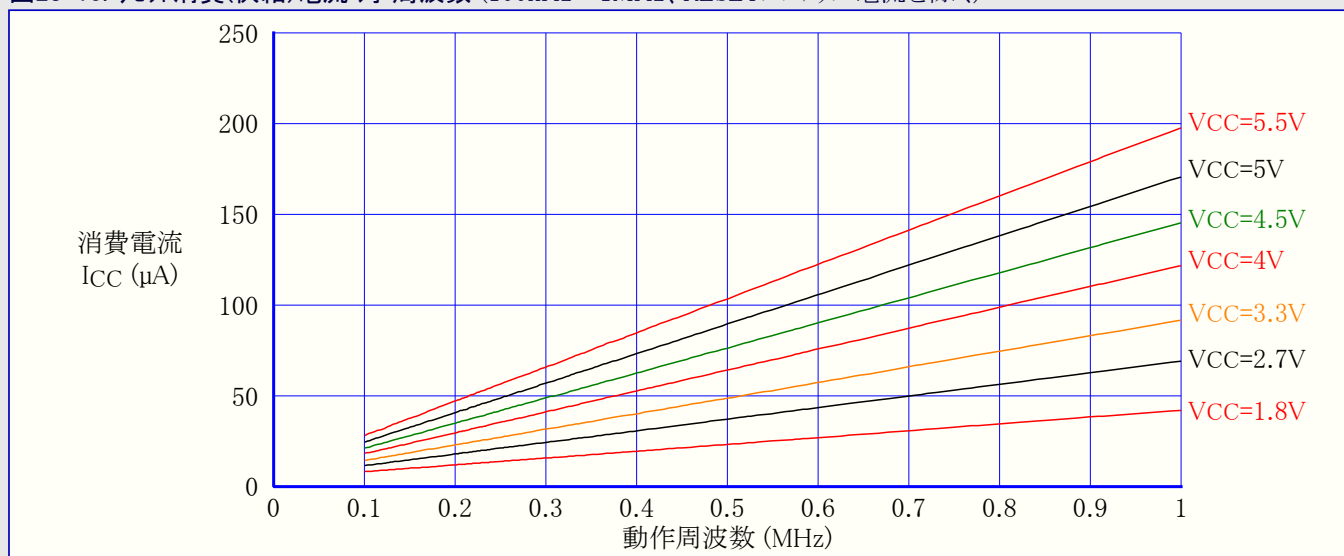
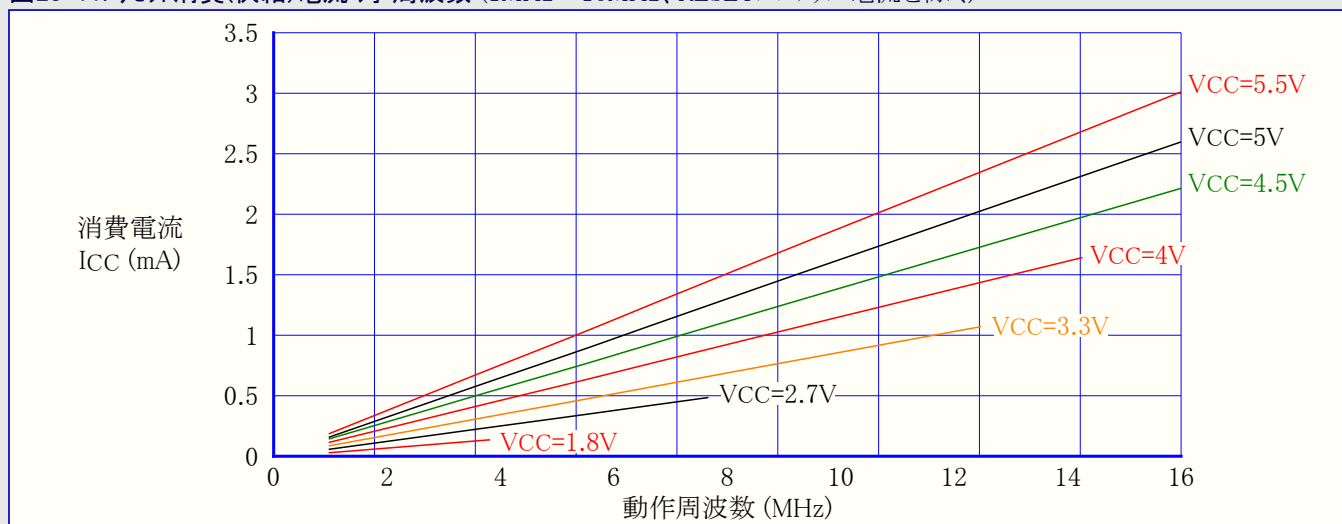


図23-11. リセット消費(供給)電流 対 周波数 (1MHz~16MHz、RESETプルアップ電流を除く)



23.6. プルアップ抵抗

図23-12. I/Oピンプルアップ抵抗電流 対 入力電圧 (VCC=2.7V)

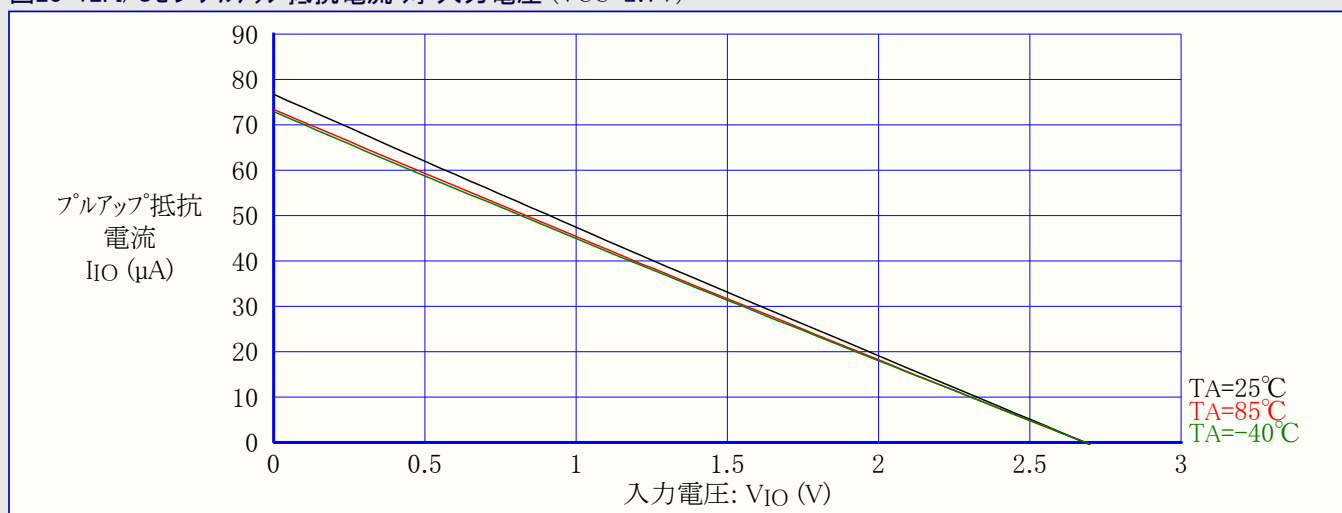


図23-13. I/Oピンプルアップ抵抗電流 対 入力電圧 (VCC=5V)

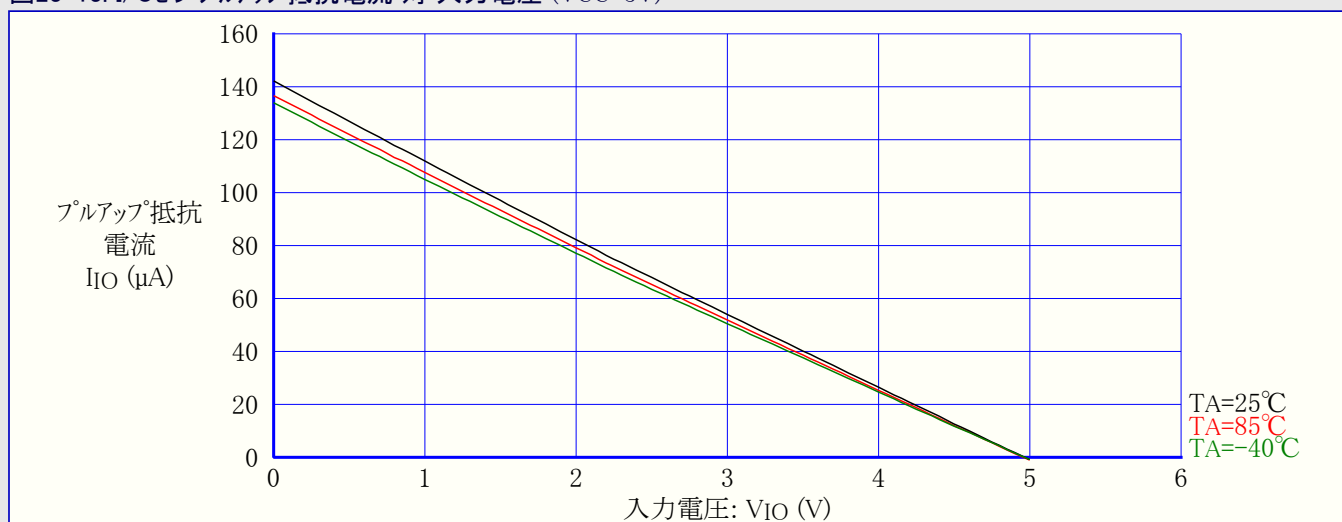


図23-14. RESETプルアップ抵抗電流 対 入力電圧 (VCC=2.7V)

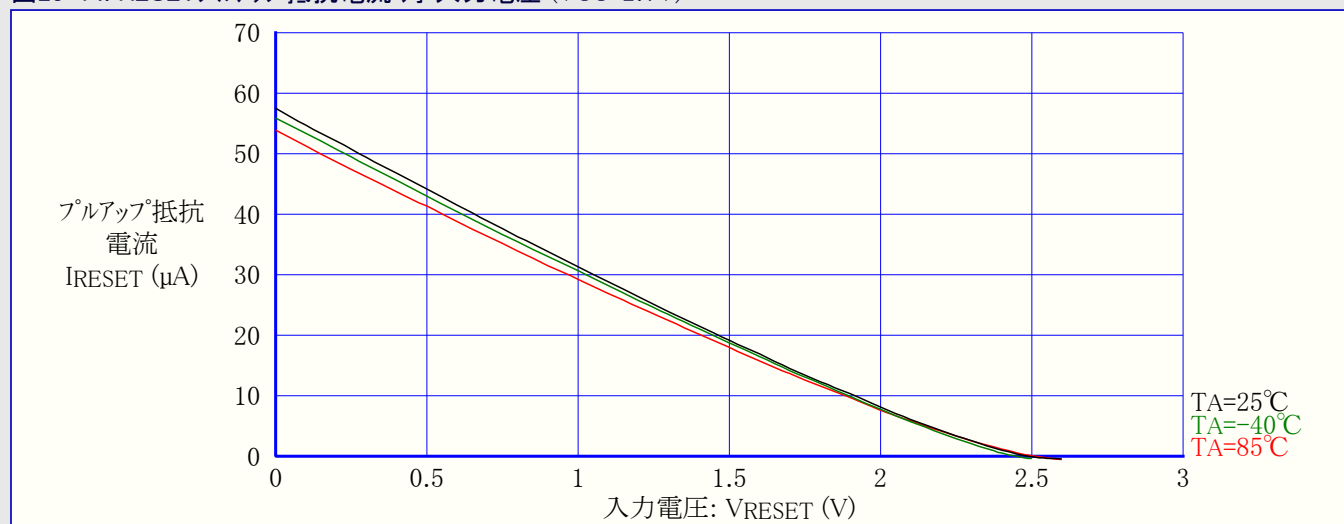
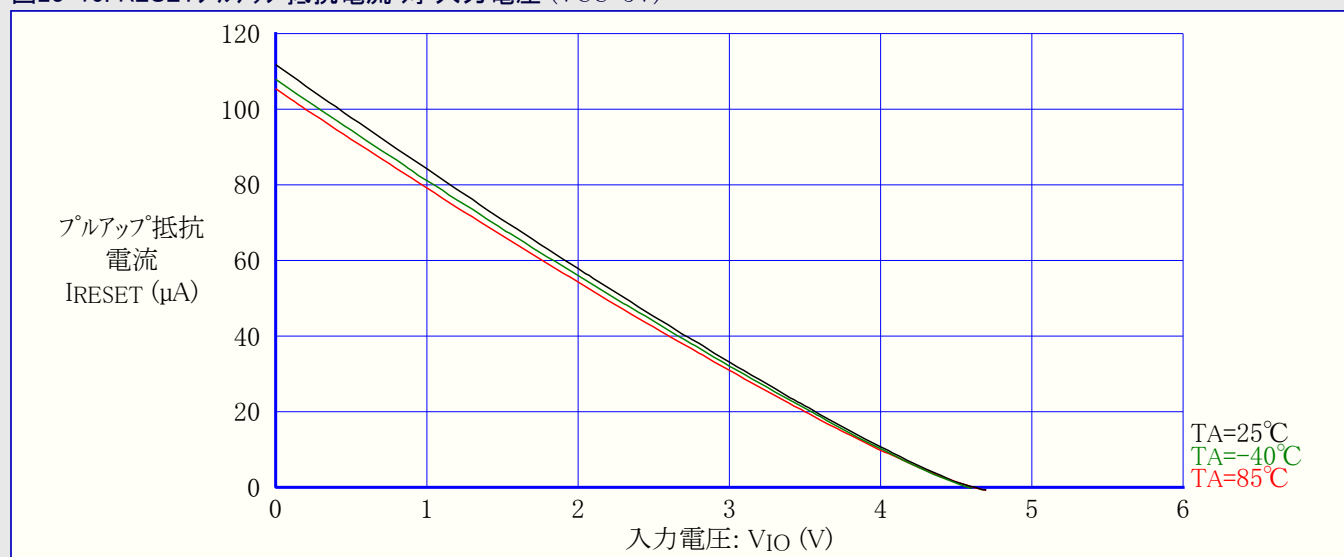


図23-15. RESETプルアップ抵抗電流 対 入力電圧 (VCC=5V)



23.7. 出力駆動部能力

図23-16. I/Oピン出力電圧 対 吸い込み電流 (VCC=3V)

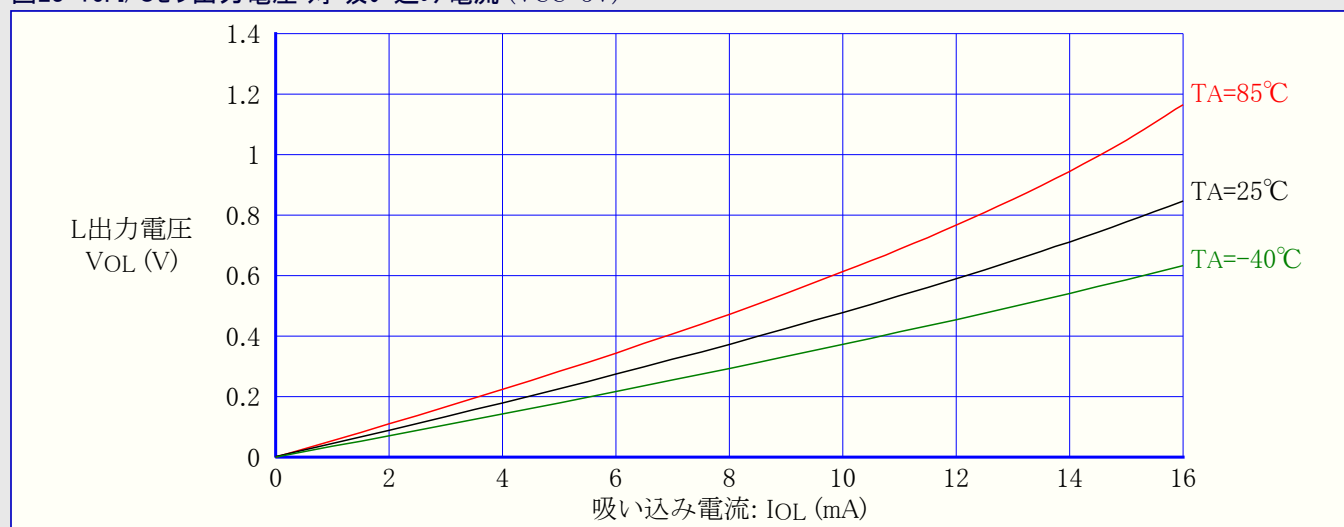


図23-17. I/Oピン出力電圧 対 吸い込み電流 (VCC=5V)

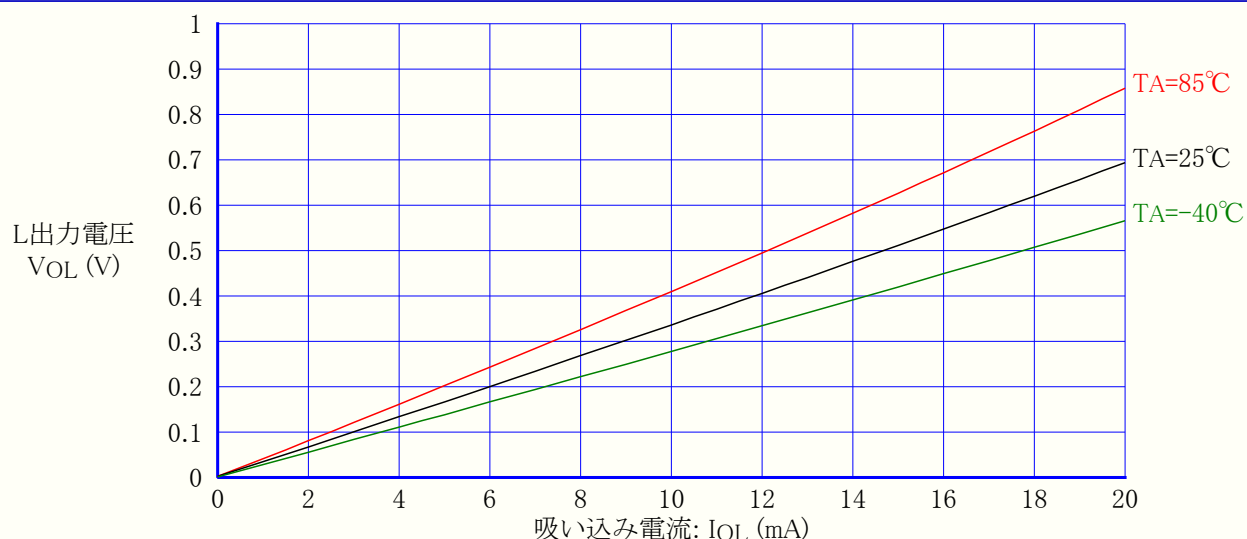


図23-18. I/Oピン出力電圧 対 吐き出し電流 (VCC=3V)

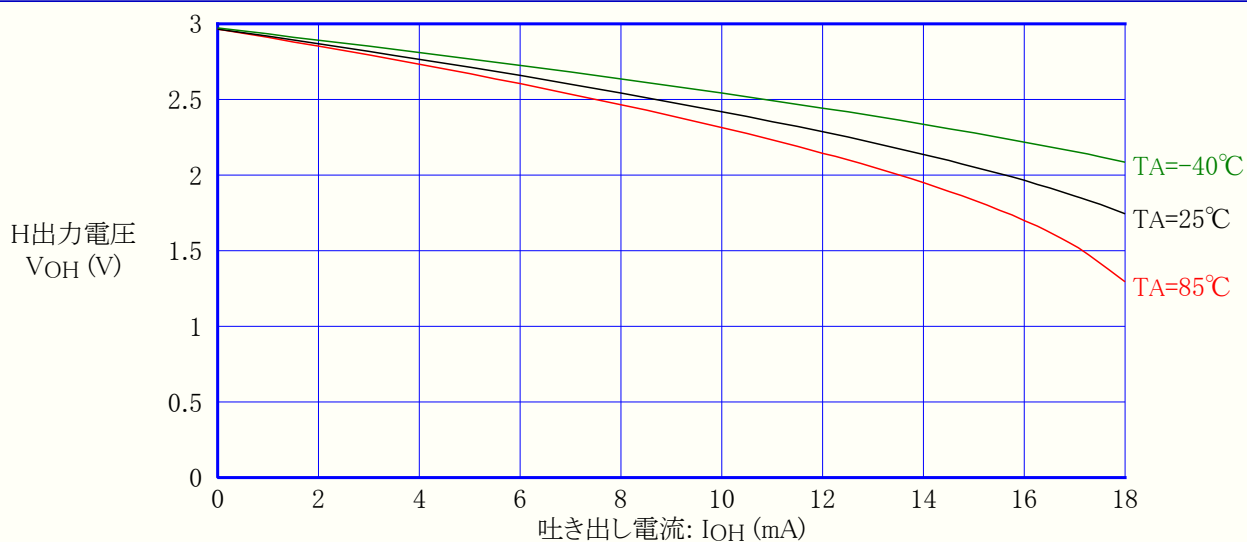
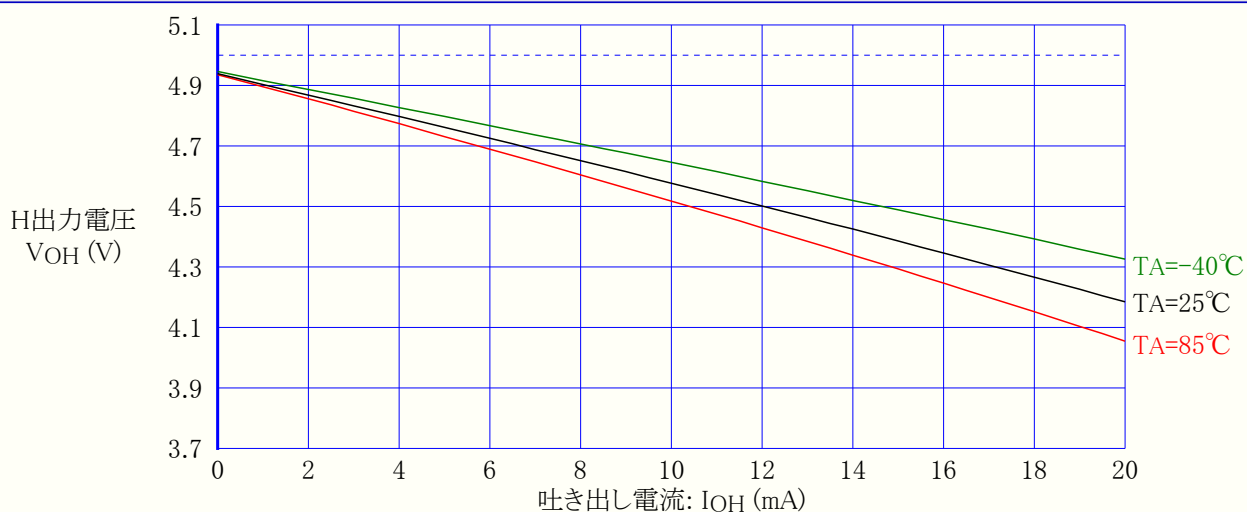


図23-19. I/Oピン出力電圧 対 吐き出し電流 (VCC=5V)



23.8. 入力閾値とヒステリシス(対入出力ポート)

図23-20. I/Oピン入力閾値(スレッショルド)電圧 対 動作電圧 (V_{IH} , 1読み値)

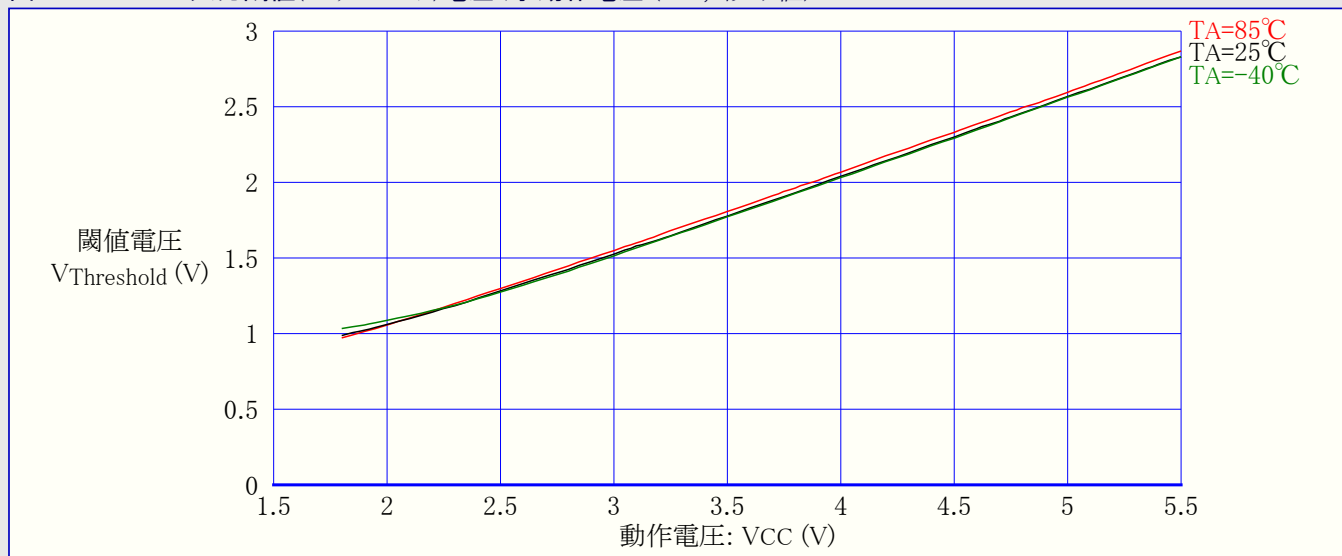
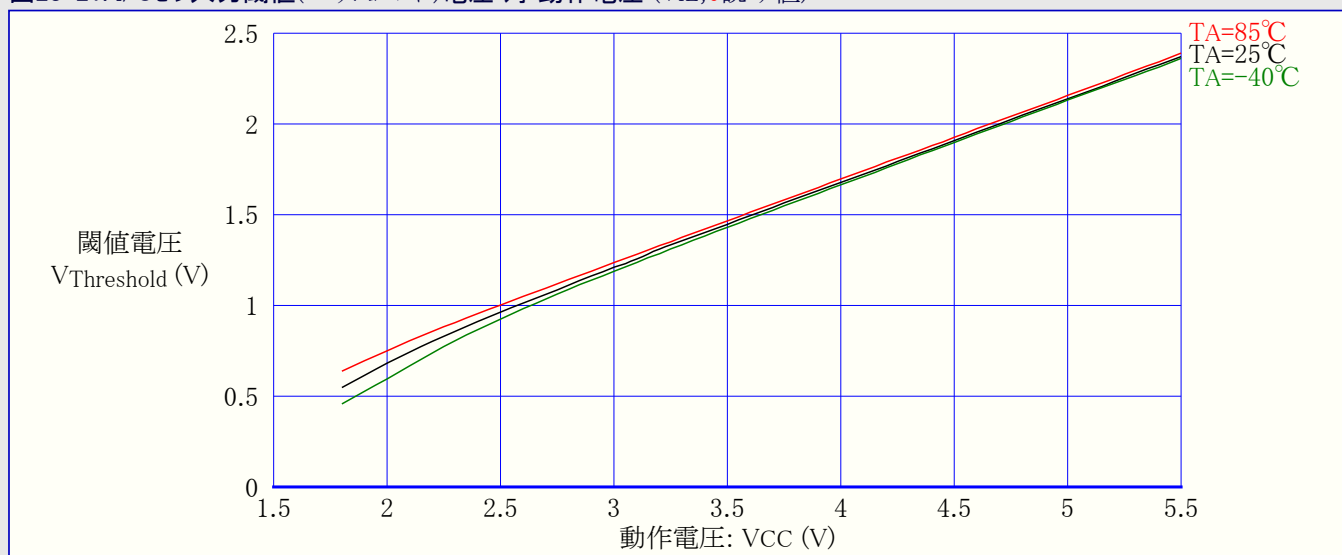


図23-21. I/Oピン入力閾値(スレッショルド)電圧 対 動作電圧 (V_{IL} , 0読み値)



23.9. 低電圧検出器(BOD)、バンドギャップ、リセット

図23-22. 低電圧検出器(BOD)閾値(スレッショルド)電圧 対 動作温度 (検出電圧4.3V)

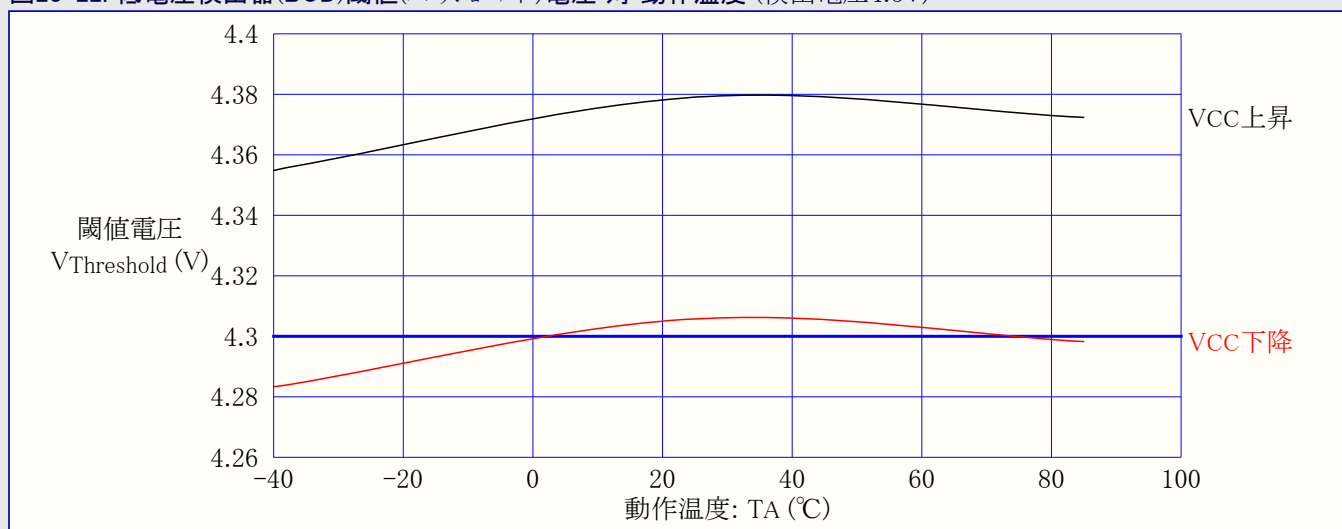


図23-23. 低電圧検出器(BOD)閾値(スレッシュホールド)電圧 対 動作温度 (検出電圧2.7V)

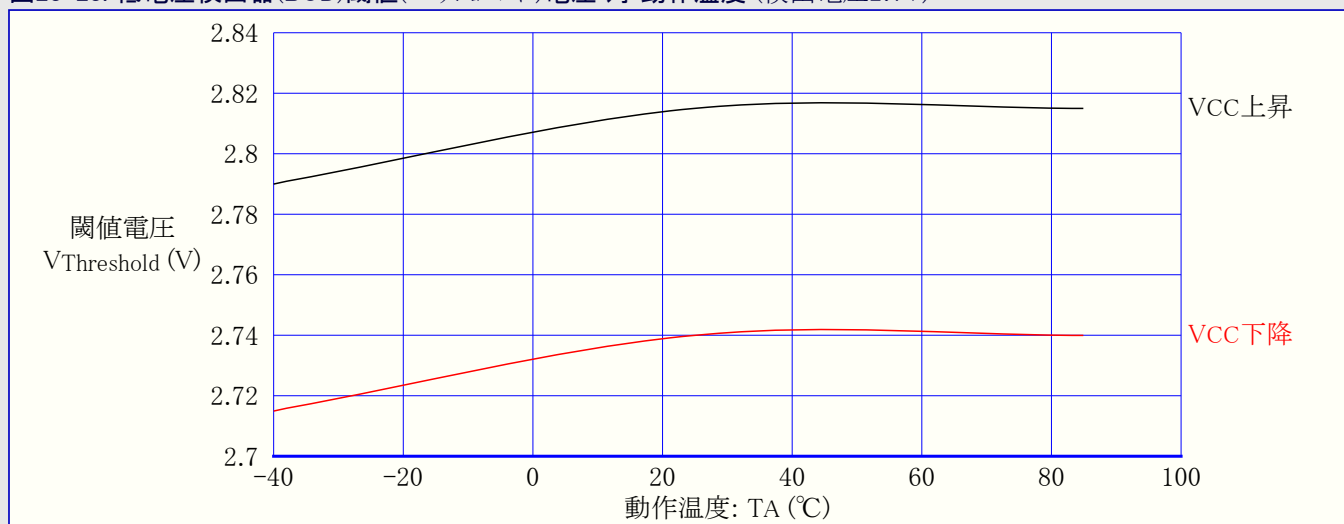


図23-24. 低電圧検出器(BOD)閾値(スレッシュホールド)電圧 対 動作温度 (検出電圧1.8V)

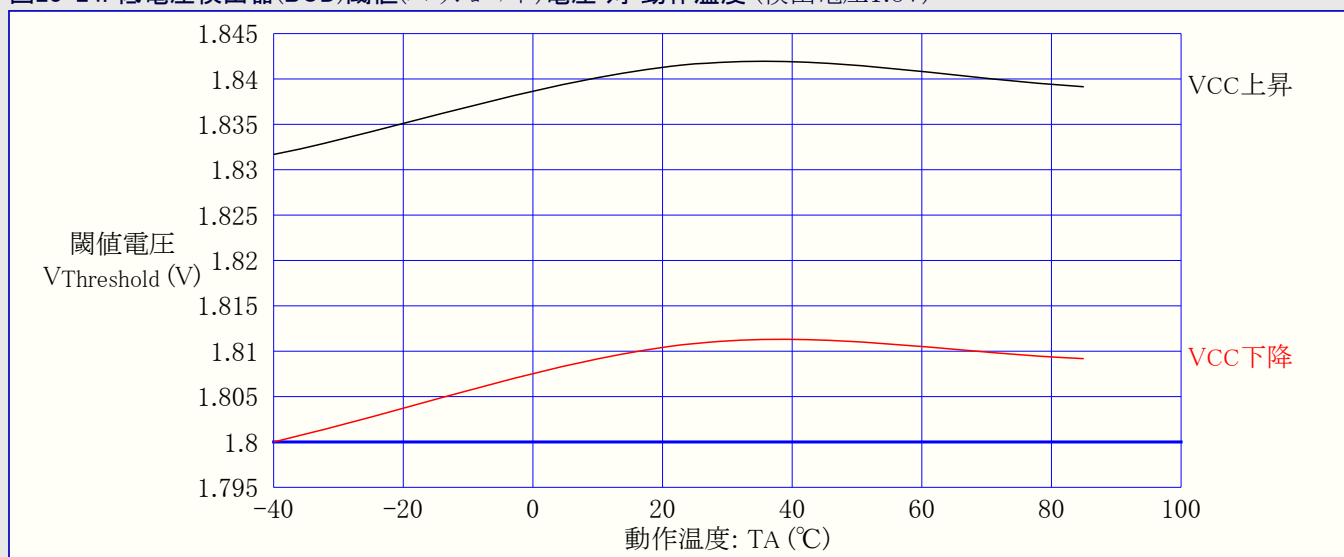


図23-25. 内部バンドギャップ電圧 対 動作電圧

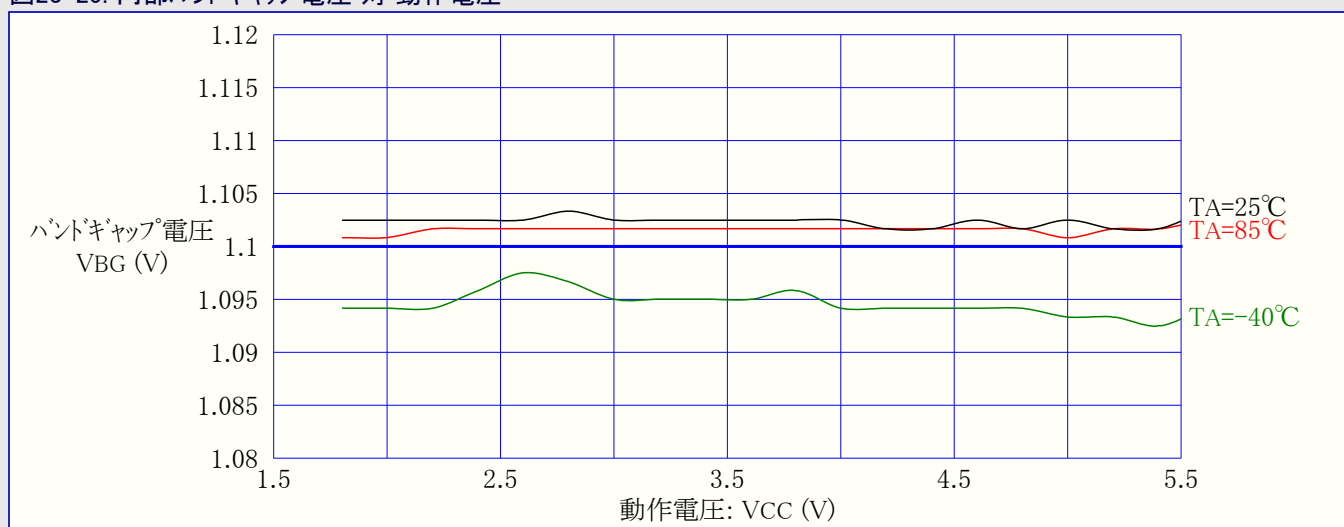


図23-26. RESET入力閾値(スレッショルド)電圧 対 動作電圧 (V_{IH} , 1読み値)

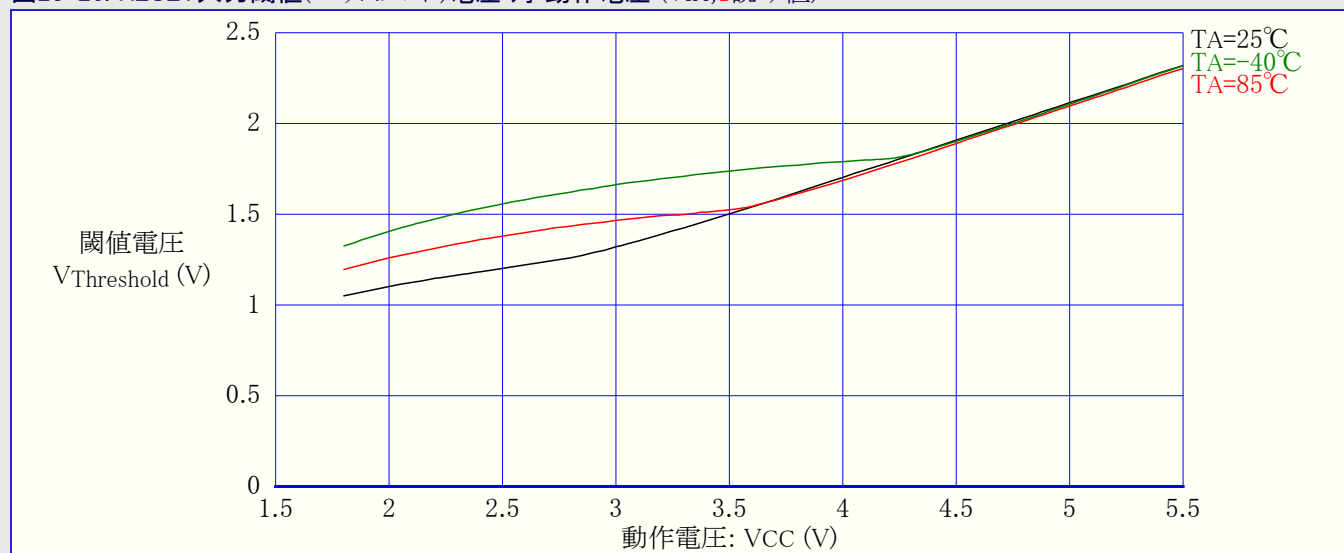
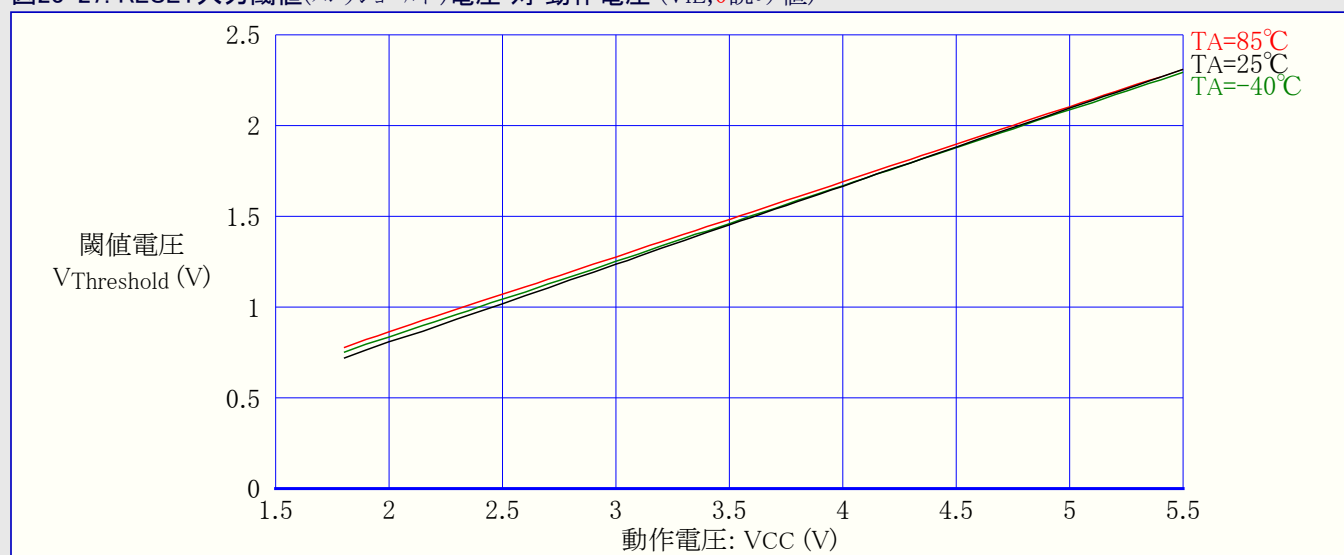


図23-27. RESET入力閾値(スレッショルド)電圧 対 動作電圧 (V_{IL} , 0読み値)



23.10. 内部発振器

図23-28. ウォッチドッグ発振器周波数 対 動作温度

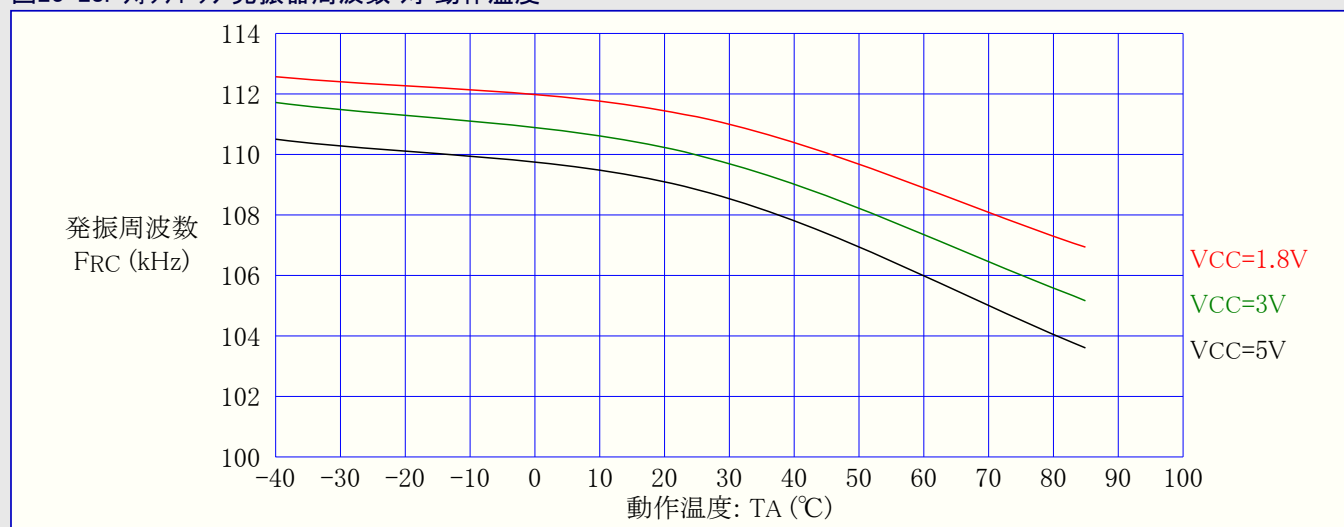
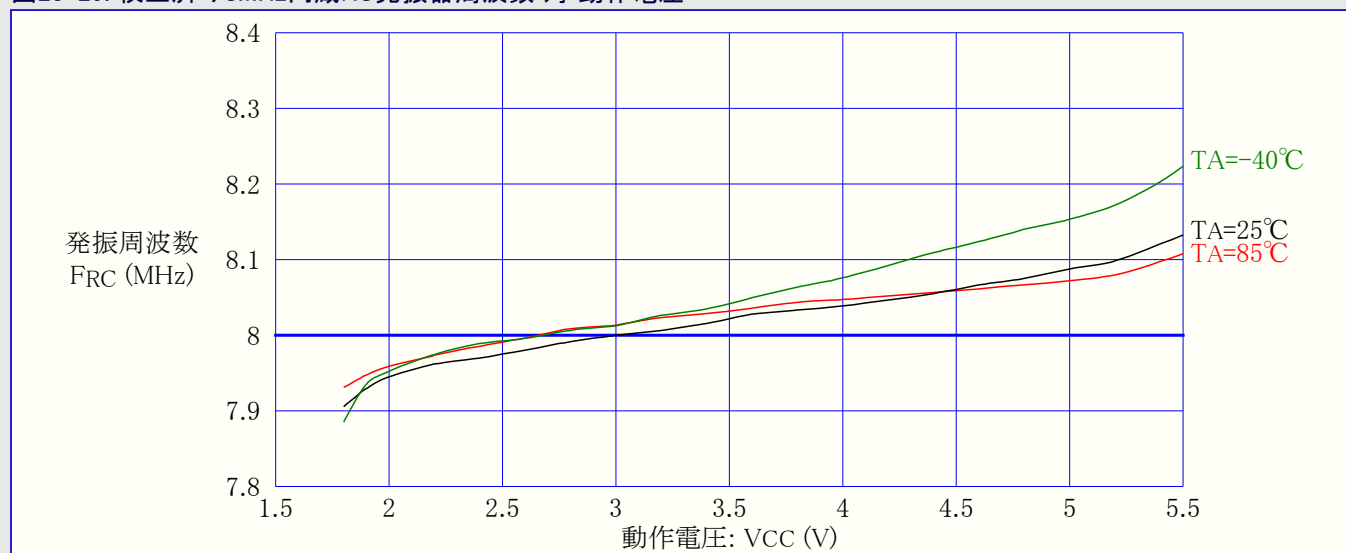


図23-29. 校正済み8MHz内蔵RC発振器周波数 対 動作電圧



24. レジスタ要約

拡張I/Oレジスタ領域 (1/3)

アドレス	レジスタ略称	ビット7	ビット6	ビット5	ビット4	ビット3	ビット2	ビット1	ビット0	頁
(\$FF)	予約									
(\$FE)	予約									
(\$FD)	予約									
(\$FC)	予約									
(\$FB)	予約									
(\$FA)	予約									
(\$F9)	予約									
(\$F8)	予約									
(\$F7)	予約									
(\$F6)	予約									
(\$F5)	予約									
(\$F4)	予約									
(\$F3)	予約									
(\$F2)	予約									
(\$F1)	予約									
(\$F0)	予約									
(\$EF)	予約									
(\$EE)	予約									
(\$ED)	予約									
(\$EC)	予約									
(\$EB)	予約									
(\$EA)	予約									
(\$E9)	予約									
(\$E8)	予約									
(\$E7)	予約									
(\$E6)	予約									
(\$E5)	予約									
(\$E4)	予約									
(\$E3)	予約									
(\$E2)	予約									
(\$E1)	予約									
(\$E0)	予約									
(\$DF)	予約									
(\$DE)	予約									
(\$DD)	予約									
(\$DC)	予約									
(\$DB)	予約									
(\$DA)	予約									
(\$D9)	予約									
(\$D8)	予約									
(\$D7)	予約									
(\$D6)	予約									
(\$D5)	予約									
(\$D4)	予約									
(\$D3)	予約									
(\$D2)	LINDAT	LDAT7	LDAT6	LDAT5	LDAT4	LDAT3	LDAT2	LDAT1	LDAT0	128
(\$D1)	LINSEL	-	-	-	-	LAINC	LINDX2	LINDX1	LINDX0	128
(\$D0)	LINIDR	LP1	LP0	LID5/LIL1	LID4/LIL0	LID3	LID2	LID1	LID0	127
(\$CF)	LINDLR	LTXDL3	LTXDL2	LTXDL1	LTXDL0	LRXDL3	LRXDL2	LRXDL1	LRXDL0	127
(\$CE)	LINBRRH	-	-	-	-	LIN/UART ボーレートレジスタ上位 (LDIV11~8)				127
(\$CD)	LINBRRL	LIN/UART ボーレートレジスタ下位 (LDIV7~0)								
(\$CC)	LINBTR	LDISR	-	LBT5	LBT4	LBT3	LBT2	LBT1	LBT0	126
(\$CB)	LINERR	LABORT	LTOERR	LOVERR	LFERR	LSERR	LPERR	LCERR	LBERR	126
(\$CA)	LINENIR	-	-	-	-	LENERR	LENIDOK	LENTXOK	LENRXOK	125
(\$C9)	LINSIR	LIDST2	LIDST1	LIDST0	LBUSY	LERR	LIDOK	LTXOK	LRXOK	124
(\$C8)	LINCR	LSWRES	LIN13	LCONF1	LCONF0	LENA	LCMD2	LCMD1	LCMD0	124
(\$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)	USIPP	-	-	-	-	-	-	-	USIPOS	111
(\$BB)	USIBR	USI 緩衝レジスタ								109
(\$BA)	USIDR	USI データレジスタ								109
(\$B9)	USISR	USISIF	USIOIF	USIPF	USIDC	USICNT3	USICNT2	USICNT1	USICNT0	109
(\$B8)	USICR	USISIE	USIOIE	USIWM1	USIWM0	USICS1	USICS0	USICLK	USITC	110
(\$B7)	予約									
(\$B6)	ASSR	-	EXCLK	AS0	TCN0UB	OCR0AUB	-	TCR0AUB	TCR0BUB	75
(\$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)	予約									
(\$8C)	予約									
(\$8B)	OCR1BH	タイマ/カウンタ1 比較Bレジスタ上位バイト								97
(\$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								97
(\$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								97
(\$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								97
(\$84)	TCNT1L	タイマ/カウンタ1 下位バイト								
(\$83)	TCCR1D	OC1BX	OC1BW	OC1BV	OC1BU	OC1AX	OC1AW	OC1AV	OC1AU	96
(\$82)	TCCR1C	FOC1A	FOC1B	-	-	-	-	-	-	96
(\$81)	TCCR1B	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10	95
(\$80)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	-	-	WGM11	WGM10	94

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	–	ADC10D	ADC9D	ADC8D	–	–	–	–	145
(\$7E)	DIDR0	ADC7D AIN1D	ADC6D AIN0D	ADC5D	ADC4D	ADC3D	ADC2D	ADC1D	ADC0D	133,144
(\$7D)	予約									
(\$7C)	ADMUX	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	142
(\$7B)	ADCSRB	BIN	ACME	ACIR1	ACIR0	–	ADTS2	ADTS1	ADTS0	143,132
(\$7A)	ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	143
(\$79)	ADCH	A/Dデータレジスタ上位バイト (ADC9~8またはADC9~2)								144
(\$78)	ADCL	A/Dデータレジスタ下位バイト (ADC7~0またはADC1~0)								
(\$77)	AMISCR	–	–	–	–	–	AREFEN	XREFEN	ISRCEN	145,130
(\$76)	予約									
(\$75)	予約									
(\$74)	予約									
(\$73)	予約									
(\$72)	予約									
(\$71)	予約									
(\$70)	予約									
(\$6F)	TIMSK1	–	–	ICIE1	–	–	OCIE1B	OCIE1A	TOIE1	87
(\$6E)	TIMSK0	–	–	–	–	–	OCIE0B	OCIE0A	TOIE0	76
(\$6D)	予約									
(\$6C)	PCMSK1	PCINT15	PCINT14	PCINT13	PCINT12	PCINT11	PCINT10	PCINT9	PCINT8	49
(\$6B)	PCMSK0	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0	49
(\$6A)	予約									
(\$69)	EICRA	–	–	–	–	ISC11	ISC10	ISC01	ISC00	47
(\$68)	PCICR	–	–	–	–	–	–	PCIE1	PCIE0	49
(\$67)	予約									
(\$66)	OSCCAL	内蔵RC発振器 発振校正値レジスタ								31
(\$65)	予約									
(\$64)	PRR	–	–	PRLIN	PRSPI	PRTIM1	PRTIM0	PRUSI	PRADC	29
(\$63)	CLKSELR	–	COUT	CSUT1	CSUT0	CSEL3	CSEL2	CSEL1	CSEL0	32
(\$62)	CLKCSR	CLKCCE	–	–	CLKRDY	CLKC3	CLKC2	CLKC1	CLKC0	32
(\$61)	CLKPR	CLKPCE	–	–	–	CLKPS3	CLKPS2	CLKPS1	CLKPS0	31
(\$60)	WDTCSR	WDIF	WDIE	WDP3	WDCE	WDE	WDP2	WDP1	WDP0	44

注意: 将来のデバイスとの共通性のため、アクセスされる場合の予約ビットは0を書かれるべきです。予約したI/Oメモリアドレスは決して書かれるべきではありません。

- アドレス範囲\$00~\$1F内のI/Oレジスタは**SBI**と**CBI**命令を使う直接ビットアクセスが可能です。これらのレジスタでは**SBIS**と**SBIC**命令を使うことによって単一ビット値が検査できます。
- いくつかの状態ビットはそれらへ論理1を書くことによって解除(0)されます。他の多くのAVRと異なり、**CBI**と**SBI**命令は指定ビットだけを操作し、故にこのような状態フラグを含むレジスタで使えることに注意してください。**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。ATtiny87/167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	11
\$3E (\$5E)	SPH	—	—	—	—	—	—	SP9	SP8	12
\$3D (\$5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	
\$3C (\$5C)	予約									
\$3B (\$5B)	予約									
\$3A (\$5A)	予約									
\$39 (\$59)	予約									
\$38 (\$58)	予約									
\$37 (\$57)	SPMCSR	—	RWWSB	SIGRD	CTPB	RFLB	PGWRT	PGERS	SPMEN	148
\$36 (\$56)	予約									
\$35 (\$55)	MCUCR	—	BODS	BODSE	PUD	—	—	—	—	37,55
\$34 (\$54)	MCUSR	—	—	—	—	WDRF	BORF	EXTRF	PORF	41
\$33 (\$53)	SMCR	—	—	—	—	—	SM1	SM0	SE	37
\$32 (\$52)	予約									
\$31 (\$51)	DWDR	モニタ(デバッグWIRE) データレジスタ								146
\$30 (\$50)	ACSR	ACD	ACIRS	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	131
\$2F (\$4F)	予約									
\$2E (\$4E)	SPDR	SPI データレジスタ								102
\$2D (\$4D)	SPSR	SPIF	WCOL	—	—	—	—	—	SPI2X	102
\$2C (\$4C)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	101
\$2B (\$4B)	GPIOR2	汎用I/Oレジスタ2								20
\$2A (\$4A)	GPIOR1	汎用I/Oレジスタ1								20
\$29 (\$49)	予約									
\$28 (\$48)	OCR0A	タイマ/カウンタ0 比較Aレジスタ								74
\$27 (\$47)	TCNT0	タイマ/カウンタ0								74
\$26 (\$46)	TCCR0B	FOC0A	—	—	—	—	CS02	CS01	CS00	74
\$25 (\$45)	TCCR0A	COM0A1	COM0A0	—	—	—	—	WGM01	WGM00	73
\$24 (\$44)	予約									
\$23 (\$43)	GTCCR	TSM	—	—	—	—	—	PSR0	PSR1	78,76
\$22 (\$42)	EEARH ①	—	—	—	—	—	—	—	EEAR8	19
\$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								19
\$1F (\$3F)	EECR	—	—	EEPM1	EEPM0	EERIE	EEMPE	EEPE	EERE	19
\$1E (\$3E)	GPIOR0	汎用I/Oレジスタ0								20
\$1D (\$3D)	EIMSK	—	—	—	—	—	—	INT1	INT0	48
\$1C (\$3C)	EIFR	—	—	—	—	—	—	INTF1	INTF0	48
\$1B (\$3B)	PCIFR	—	—	—	—	—	—	PCIF1	PCIF0	49
\$1A (\$3A)	予約									
\$19 (\$39)	予約									
\$18 (\$38)	予約									
\$17 (\$37)	予約									
\$16 (\$36)	TIFR1	—	—	ICF1	—	—	OCF1B	OCF1A	TOV1	87
\$15 (\$35)	TIFR0	—	—	—	—	—	—	OCF0A	TOV0	76
\$14 (\$34)	予約									
\$13 (\$33)	予約									
\$12 (\$32)	PORTCR	—	—	BBMB	BBMA	—	—	PUDB	PUDA	55
\$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)	予約									
\$0A (\$2A)	予約									
\$09 (\$29)	予約									
\$08 (\$28)	予約									
\$07 (\$27)	予約									
\$06 (\$26)	予約									
\$05 (\$25)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	62
\$04 (\$24)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	62
\$03 (\$23)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	62
\$02 (\$22)	PORTA	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	62
\$01 (\$21)	DDRA	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	62
\$00 (\$20)	PINA	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	62

注: ① EEAMSB(154頁の表21-8.)を越えるアドレスビットは無効です。

(訳注) 原書本位置の一般注意は前頁に移動しました。

25. 命令要約

ニーモニック	オペランド	意味	動作	フラグ	クロック
算術、論理演算命令					
ADD	Rd,Rr	汎用レジスタ間の加算	$Rd \leftarrow Rd + Rr$	I,T,H,S,V,N,Z,C	1
ADC	Rd,Rr	キャリーを含めた汎用レジスタ間の加算	$Rd \leftarrow Rd + Rr + C$	I,T,H,S,V,N,Z,C	1
ADIW	Rd,K6	即値の語(ワード)長加算	$RdH:RdL \leftarrow RdH:RdL + K6$	I,T,H,S,V,N,Z,C	2
SUB	Rd,Rr	汎用レジスタ間の減算	$Rd \leftarrow Rd - Rr$	I,T,H,S,V,N,Z,C	1
SUBI	Rd,K	汎用レジスタから即値の減算	$Rd \leftarrow Rd - K$	I,T,H,S,V,N,Z,C	1
SBIW	Rd,K6	即値の語(ワード)長減算	$RdH:RdL \leftarrow RdH:RdL - K6$	I,T,H,S,V,N,Z,C	2
SBC	Rd,Rr	キャリーを含めた汎用レジスタ間の減算	$Rd \leftarrow Rd - Rr - C$	I,T,H,S,V,N,Z,C	1
SBCI	Rd,K	汎用レジスタからキャリーと即値の減算	$Rd \leftarrow Rd - K - C$	I,T,H,S,V,N,Z,C	1
AND	Rd,Rr	汎用レジスタ間の論理積(AND)	$Rd \leftarrow Rd \text{ AND } Rr$	I,T,H,S,0,N,Z,C	1
ANDI	Rd,K	汎用レジスタと即値の論理積(AND)	$Rd \leftarrow Rd \text{ AND } K$	I,T,H,S,0,N,Z,C	1
OR	Rd,Rr	汎用レジスタ間の論理和(OR)	$Rd \leftarrow Rd \text{ OR } Rr$	I,T,H,S,0,N,Z,C	1
ORI	Rd,K	汎用レジスタと即値の論理和(OR)	$Rd \leftarrow Rd \text{ OR } K$	I,T,H,S,0,N,Z,C	1
EOR	Rd,Rr	汎用レジスタ間の排他的論理和(Ex-OR)	$Rd \leftarrow Rd \text{ EOR } Rr$	I,T,H,S,0,N,Z,C	1
COM	Rd	1の補数(論理反転)	$Rd \leftarrow \$FF - Rd$	I,T,H,S,0,N,Z,1	1
NEG	Rd	2の補数	$Rd \leftarrow \$00 - Rd$	I,T,H,S,V,N,Z,C	1
SBR	Rd,K	汎用レジスタの(複数)ビット設定(1)	$Rd \leftarrow Rd \text{ OR } K$	I,T,H,S,0,N,Z,C	1
CBR	Rd,K	汎用レジスタの(複数)ビット解除(0)	$Rd \leftarrow Rd \text{ AND } (\$FF - K)$	I,T,H,S,0,N,Z,C	1
INC	Rd	汎用レジスタの増加(+1)	$Rd \leftarrow Rd + 1$	I,T,H,S,V,N,Z,C	1
DEC	Rd	汎用レジスタの減少(-1)	$Rd \leftarrow Rd - 1$	I,T,H,S,V,N,Z,C	1
TST	Rd	汎用レジスタのゼロとマイナス検査	$Rd \leftarrow Rd \text{ AND } Rd$	I,T,H,S,0,N,Z,C	1
CLR	Rd	汎用レジスタの全0設定(=\$00)	$Rd \leftarrow Rd \text{ EOR } Rd$	I,T,H,0,0,0,1,C	1
SER	Rd	汎用レジスタの全1設定(=\$FF)	$Rd \leftarrow \$FF$	I,T,H,S,V,N,Z,C	1
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		Zレジスタ間接無条件分岐	$PC \leftarrow Z$	I,T,H,S,V,N,Z,C	2
JMP (注)	k	絶対無条件分岐	$PC \leftarrow k$	I,T,H,S,V,N,Z,C	3
RCALL	k	相対サブルーチン呼び出し	$STACK \leftarrow PC, PC \leftarrow PC + k + 1$	I,T,H,S,V,N,Z,C	3
ICALL		Zレジスタ間接サブルーチン呼び出し	$STACK \leftarrow PC, PC \leftarrow Z$	I,T,H,S,V,N,Z,C	3
CALL (注)	k	絶対サブルーチン呼び出し	$STACK \leftarrow PC, PC \leftarrow k$	I,T,H,S,V,N,Z,C	4
RET		サブルーチンからの復帰	$PC \leftarrow STACK$	I,T,H,S,V,N,Z,C	4
RETI		割り込みからの復帰	$PC \leftarrow STACK$	I,T,H,S,V,N,Z,C	4
CPSE	Rd,Rr	汎用レジスタ間比較、一致でスキップ	$Rd=Rr$ なら, $PC \leftarrow PC + 2or3$	I,T,H,S,V,N,Z,C	1/2,3
CP	Rd,Rr	汎用レジスタ間の比較	$Rd - Rr$	I,T,H,S,V,N,Z,C	1
CPC	Rd,Rr	キャリーを含めた汎用レジスタ間の比較	$Rd - Rr - C$	I,T,H,S,V,N,Z,C	1
CPI	Rd,K	汎用レジスタと即値の比較	$Rd - K$	I,T,H,S,V,N,Z,C	1
SBRC	Rr,b	汎用レジスタのビットが解除(0)でスキップ	$Rr(b)=0$ なら, $PC \leftarrow PC + 2or3$	I,T,H,S,V,N,Z,C	1/2,3
SBRs	Rr,b	汎用レジスタのビットが設定(1)でスキップ	$Rr(b)=1$ なら, $PC \leftarrow PC + 2or3$	I,T,H,S,V,N,Z,C	1/2,3
SBIC	P,b	I/Oレジスタのビットが解除(0)でスキップ	$P(b)=0$ なら, $PC \leftarrow PC + 2or3$	I,T,H,S,V,N,Z,C	1/2,3
SBS	P,b	I/Oレジスタのビットが設定(1)でスキップ	$P(b)=1$ なら, $PC \leftarrow PC + 2or3$	I,T,H,S,V,N,Z,C	1/2,3
BRBS	s,k	ステータスフラグが設定(1)で分岐	$SREG(s)=1$ なら, $PC \leftarrow PC + K + 1$	I,T,H,S,V,N,Z,C	1/2
BRBC	s,k	ステータスフラグが解除(0)で分岐	$SREG(s)=0$ なら, $PC \leftarrow PC + K + 1$	I,T,H,S,V,N,Z,C	1/2
BREQ	k	一致で分岐	$Z=1$ なら, $PC \leftarrow PC + K + 1$	I,T,H,S,V,N,Z,C	1/2
BRNE	k	不一致で分岐	$Z=0$ なら, $PC \leftarrow PC + K + 1$	I,T,H,S,V,N,Z,C	1/2
BRCS	k	キャリーフラグが設定(1)で分岐	$C=1$ なら, $PC \leftarrow PC + K + 1$	I,T,H,S,V,N,Z,C	1/2
BRCC	k	キャリーフラグが解除(0)で分岐	$C=0$ なら, $PC \leftarrow PC + K + 1$	I,T,H,S,V,N,Z,C	1/2
BRSH	k	符号なしの \geq で分岐	$C=0$ なら, $PC \leftarrow PC + K + 1$	I,T,H,S,V,N,Z,C	1/2
BRLO	k	符号なしの $<$ で分岐	$C=1$ なら, $PC \leftarrow PC + K + 1$	I,T,H,S,V,N,Z,C	1/2
BRMI	k	-(マイナス)で分岐	$N=1$ なら, $PC \leftarrow PC + K + 1$	I,T,H,S,V,N,Z,C	1/2
BRPL	k	+(プラス)で分岐	$N=0$ なら, $PC \leftarrow PC + K + 1$	I,T,H,S,V,N,Z,C	1/2
BRGE	k	符号付きの \geq で分岐	$(N \text{ EOR } V)=0$ なら, $PC \leftarrow PC + K + 1$	I,T,H,S,V,N,Z,C	1/2
BRLT	k	符号付きの $<$ で分岐	$(N \text{ EOR } V)=1$ なら, $PC \leftarrow PC + K + 1$	I,T,H,S,V,N,Z,C	1/2
BRHS	k	ハーフキャリーフラグが設定(1)で分岐	$H=1$ なら, $PC \leftarrow PC + K + 1$	I,T,H,S,V,N,Z,C	1/2
BRHC	k	ハーフキャリーフラグが解除(0)で分岐	$H=0$ なら, $PC \leftarrow PC + K + 1$	I,T,H,S,V,N,Z,C	1/2
BRTS	k	一時フラグが設定(1)で分岐	$T=1$ なら, $PC \leftarrow PC + K + 1$	I,T,H,S,V,N,Z,C	1/2
BRTC	k	一時フラグが解除(0)で分岐	$T=0$ なら, $PC \leftarrow PC + K + 1$	I,T,H,S,V,N,Z,C	1/2
BRVS	k	2の補数溢れフラグが設定(1)で分岐	$V=1$ なら, $PC \leftarrow PC + K + 1$	I,T,H,S,V,N,Z,C	1/2
BRVC	k	2の補数溢れフラグが解除(0)で分岐	$V=0$ なら, $PC \leftarrow PC + K + 1$	I,T,H,S,V,N,Z,C	1/2
BRIE	k	割り込み許可で分岐	$I=1$ なら, $PC \leftarrow PC + K + 1$	I,T,H,S,V,N,Z,C	1/2
BRID	k	割り込み禁止で分岐	$I=0$ なら, $PC \leftarrow PC + K + 1$	I,T,H,S,V,N,Z,C	1/2

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)
注: これらの命令はATtiny167でだけ利用できます。

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,1	1
CLC		キャリーフラグを解除(0)	$C \leftarrow 0$	I,T,H,S,V,N,Z,0	1
SEN		負フラグを設定(1)	$N \leftarrow 1$	I,T,H,S,V,1,Z,C	1
CLN		負フラグを解除(0)	$N \leftarrow 0$	I,T,H,S,V,0,Z,C	1
SEZ		ゼロフラグを設定(1)	$Z \leftarrow 1$	I,T,H,S,V,N,1,C	1
CLZ		ゼロフラグを解除(0)	$Z \leftarrow 0$	I,T,H,S,V,N,0,C	1
SEI		全割り込み許可	$I \leftarrow 1$	I,T,H,S,V,N,Z,C	1
CLI		全割り込み禁止	$I \leftarrow 0$	0,T,H,S,V,N,Z,C	1
SES		符号フラグを設定(1)	$S \leftarrow 1$	I,T,H,1,V,N,Z,C	1
CLS		符号フラグを解除(0)	$S \leftarrow 0$	I,T,H,0,V,N,Z,C	1
SEV		2の補数溢れフラグを設定(1)	$V \leftarrow 1$	I,T,H,S,1,N,Z,C	1
CLV		2の補数溢れフラグを解除(0)	$V \leftarrow 0$	I,T,H,S,0,N,Z,C	1
SET		一時フラグを設定(1)	$T \leftarrow 1$	I,1,H,S,V,N,Z,C	1
CLT		一時フラグを解除(0)	$T \leftarrow 0$	I,0,H,S,V,N,Z,C	1
SEH		ハーフキャリーフラグを設定(1)	$H \leftarrow 1$	I,T,H,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

26. 注文情報

デバイス	速度(MHz)	電源電圧	注文符号 (注3)	外囲器 (注1)	動作範囲
ATtiny87	16	1.8～5.5V	ATtiny87-SU	20S2	
			ATtiny87-SUR (注2)		
			ATtiny87-XU	20X	
			ATtiny87-XUR (注2)		
			ATtiny87-MU	32PN	
			ATtiny87-MUR (注2)		
ATtiny167	16	1.8～5.5V	ATtiny167-SU	20S2	工業用 (-40℃～85℃)
			ATtiny167-SUR (注2)		
			ATtiny167-XU	20X	
			ATtiny167-XUR (注2)		
			ATtiny167-MU	32PN	
			ATtiny167-MUR (注2)		
			ATtiny167-MMU	TSB	
			ATtiny167-MMUR (注2)		

注1: 全ての外囲器は鉛フリー、ハロゲン化合物フリーで完全に安全で、これらは有害物質使用制限に関する欧州指令(RoHS指令)に適合します。

注2: テープとリール。

注3: このデバイスはウェハー(チップ単体)形状でも供給できます。最低数量と詳細な注文情報については最寄のMicrochip営業所へお問い合わせください。

外囲器形式

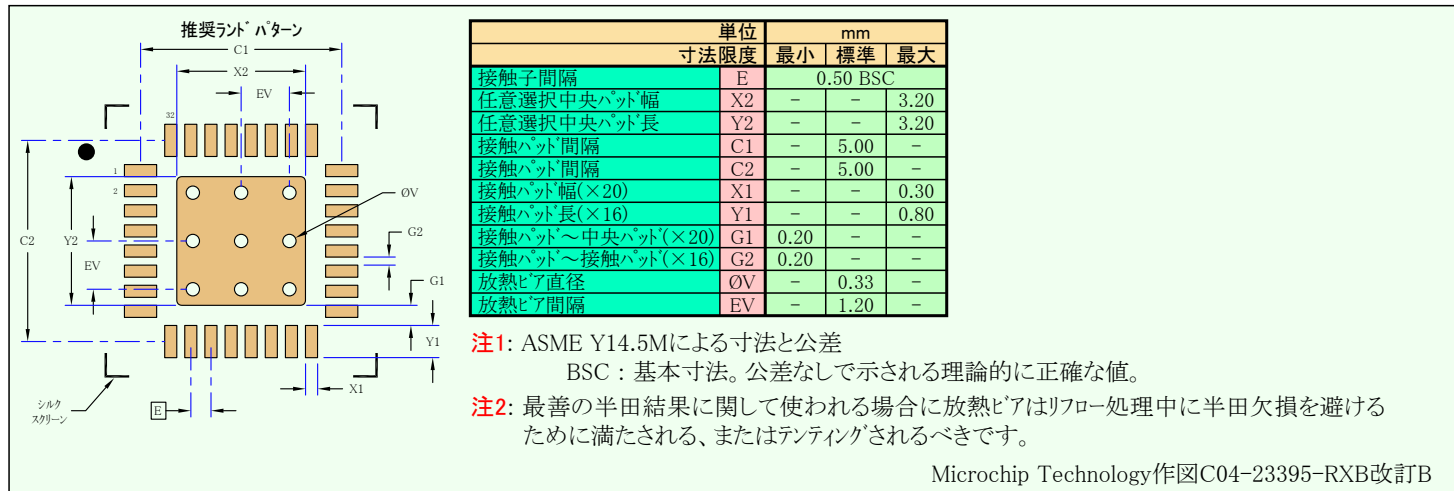
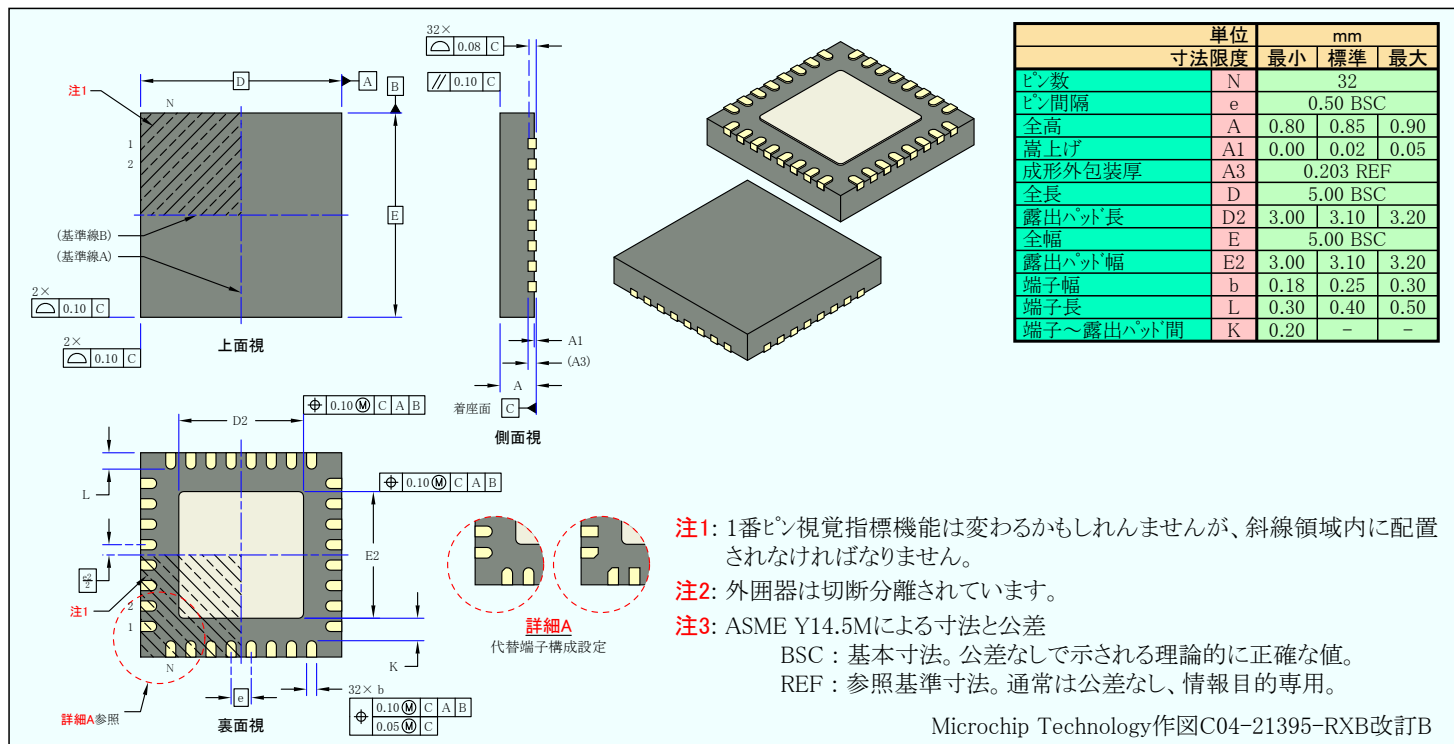
20S2	20リード 300mil幅 プラスティック翼状小外形外囲器 (SOIC)
20X	20リード 4.4mm幅 プラスティック縮小薄型小外形外囲器 (TSSOP)
32PN	32パッド 0.5mmピッチ 極薄4方向平板リードなし外囲器 (VQFN)
TSB	20パッド 極薄4方向平板リードなし外囲器 (WQFN)

27. 外圍器情報

注: 最新の外圍器図については<http://www.microchip.com/packaging>に置かれたMicrochip外圍器仕様をご覧ください。

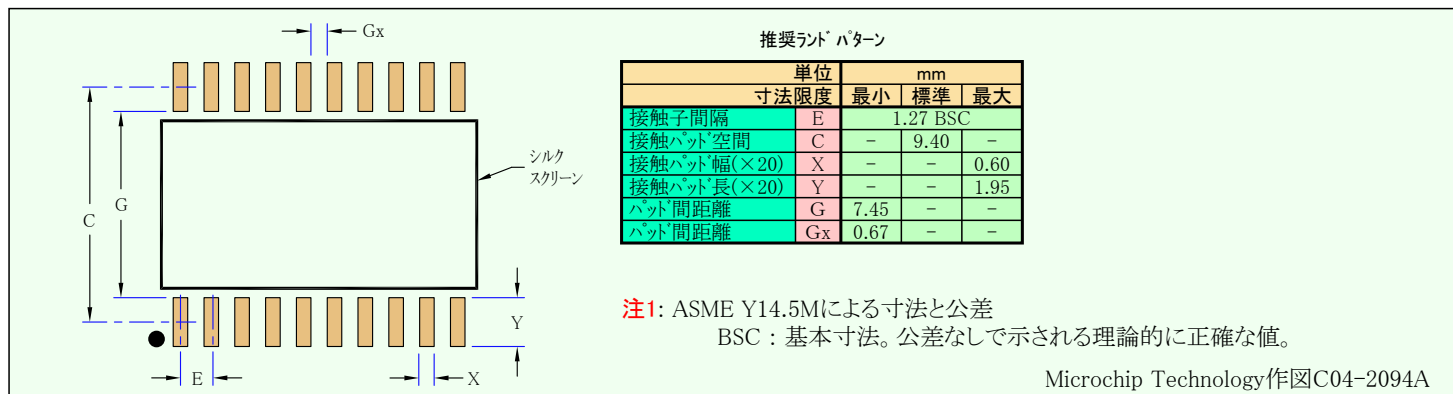
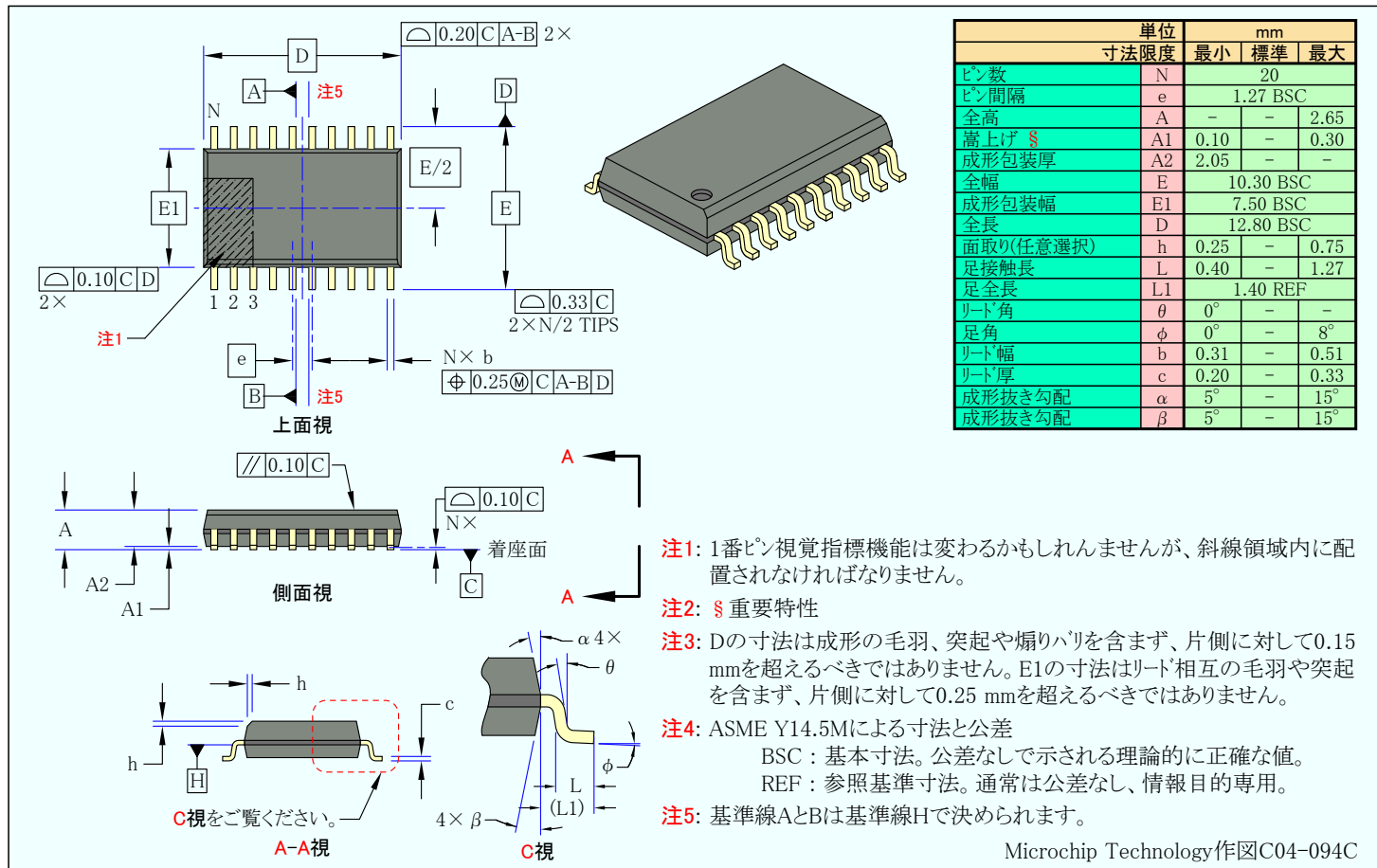
27.1. 32PN

32パッド極薄プラスチック四角平板リードなし外圍器(RXB) – 5×5×0.9mm本体 [VQFN] 3.1×3.1mm露出パッド付き(Atmel伝承全般外圍器符号ZMF)



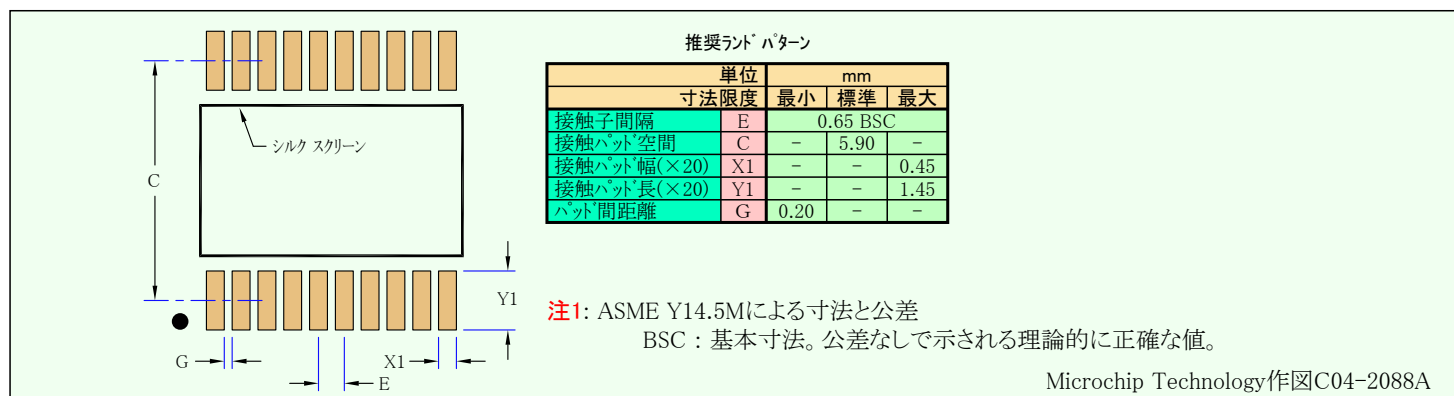
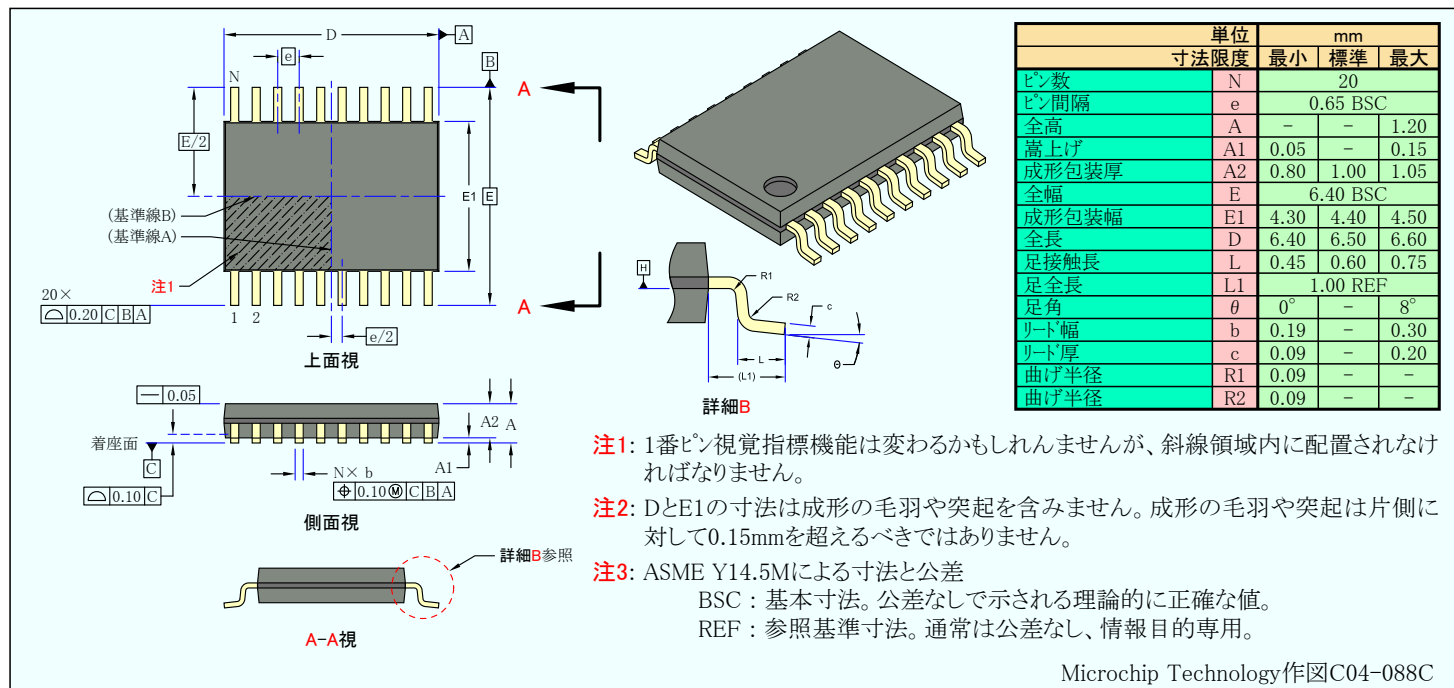
27.2. 20S2

20リード プラスティック小型外形(SO) - 7.50mm幅本体 [SOIC]



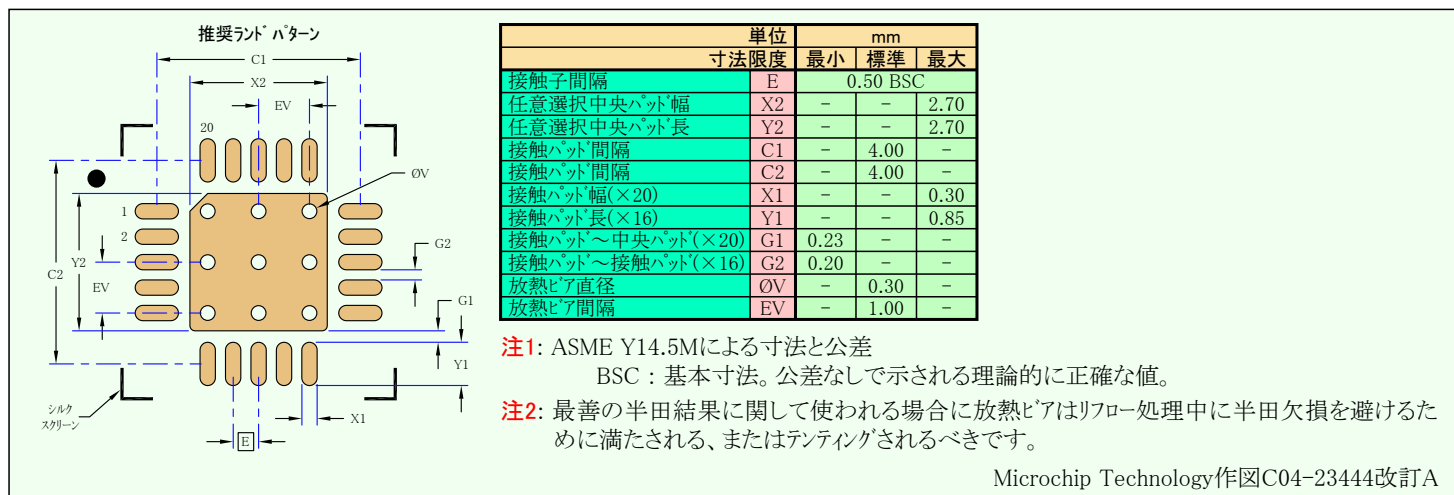
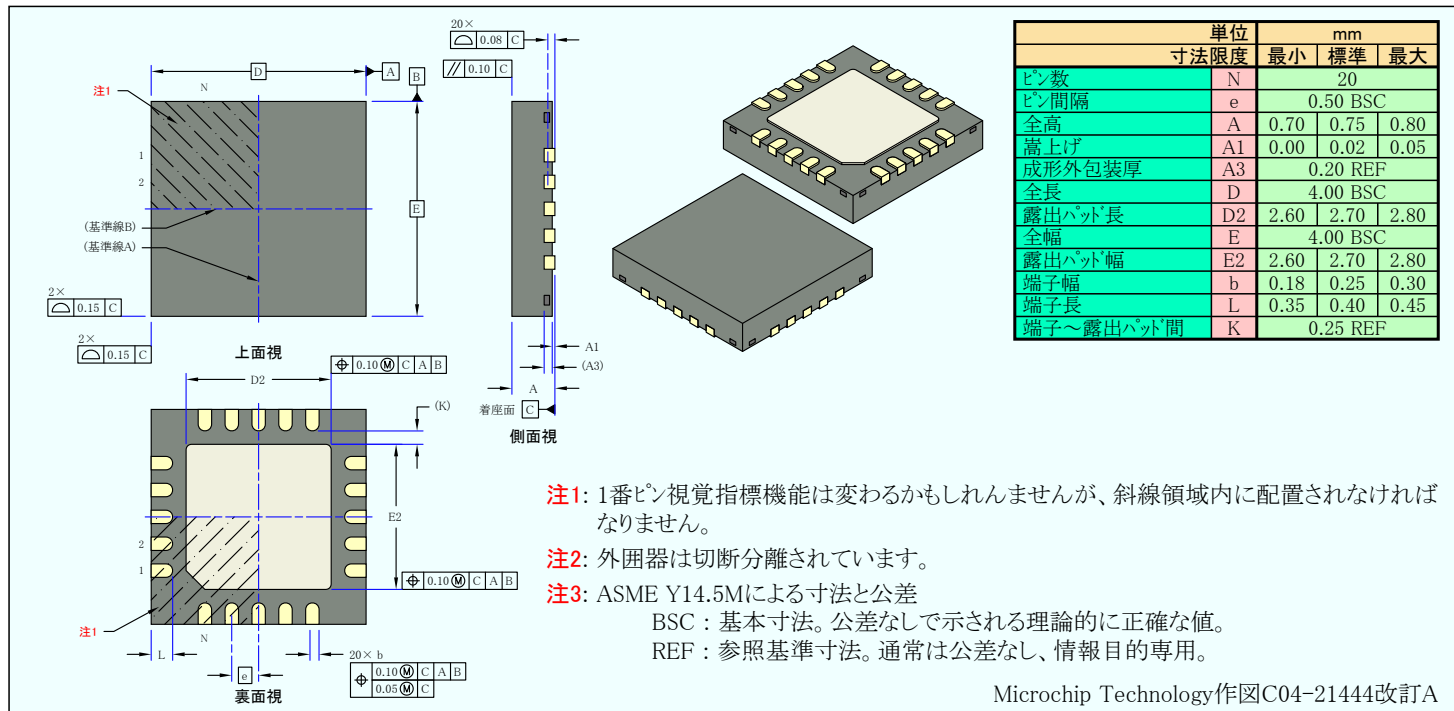
27.3. 20X

20リード プラスティック薄型縮小外形(ST) - 4.4mm本体 [TSSOP]



27.4. TSB

20リード 超薄四角平板リードなし外囲器(TSB) - 4×4×0.8mm本体 [WQFN] 2.7mm露出パッド付き(Atmel伝承全般外囲器符号ZAJ)



28. 障害情報

障害内容は<http://www.microchip.com/DS80000850>にある独立した文書「ATtiny87/167シリコン障害情報とデータシート説明」に移動されています。(訳注:本書では便宜のためこの文書の障害情報も含みます。)

この章の改訂番号はATtiny87/167デバイスの改訂版を参照してください。

ATtiny87 改訂A,B : 試供されていません。

ATtiny167 改訂A,B : 試供されていません。

- ・ **クリスタル用発振器の利得制御** 87-C,167-C
- ・ **'クロック元禁止'命令が許可のままになる** 87-C,167-C

1. クリスタル用発振器の利得制御 (87-C,167-C)

クリスタル用発振器(0.4~16MHz)はその利得制御(CKSEL/CSEL2~0ビット)をラッチしません。

- a. 'システムクロック元回復'命令はCSEL2~0ビットを返しませんが、
- b. 利得制御は**クロック選択レジスタ(CLKSELR)**を変更するならば進行中に変更することができます。

対策/対処

- a. 対策はありません。
- b. どのCLKSELR変更後も可能な限り早く、CLKSELRに適切なクリスタル用発振器設定(CSEL3=1とCSEL2~0/CSUT1,0)を再書き込みしてください。

コード例:

```

; クリスタル用発振器(16MHzクリスタル、高速立ち上がり電源)選択
    LDI    temp1, (($0F<<CSEL0) | ($02<<CSUT0))
    STS    CLKSELR, temp1
; クロック元許可(クリスタル用発振器)
    LDI    temp2, (1<<CLKCCE)
    LDI    temp3, ($02<<CLKC0)          ; CSEL="0010"
    STS    CLKCSR, temp2                ; CLKCSRレジスタ アクセス許可
    STS    CLKCSR, temp3                ; クリスタル用発振器クロック許可
; クロック元切り替え
    LDI    temp3, ($04<<CLKC0)          ; CSEL="0100"
    STS    CLKCSR, temp2                ; CLKCSRレジスタ アクセス許可
    STS    CLKCSR, temp3                ; クロック元切り替え
; ウォッチドッグクロック(128kHz、高速立ち上がり電源)選択
    LDI    temp3, (($03<<CSEL0) | ($02<<CSUT0))
    STS    CLKSELR, temp3                ; (*)
; (*) !!! ここでクリスタル用発振器の利得制御を失います。!!!
; => 対策は、
    STS    CLKSELR, temp1

```

2. 'クロック元禁止'命令が許可のままになる (87-C,167-C)

動的クロック切り替え単位部に於いて、'クロック元禁止'命令は目的対象のクロック元を禁止した(クロック元が**クロック選択レジスタ(CLKSELR)**に設定された)後も作動のままになります。

対策/対処

'クロック元禁止'命令後、\$80を書いて**クロック制御/状態レジスタ(CLKCSR)**をリセットしてください。

コード例:

```

; クリスタル用発振器選択
    LDI    temp1, ($0F<<CSEL0)
    STS    CLKSELR, temp1
; クロック元(クリスタル用発振器)禁止
    LDI    temp2, (1<<CLKCCE)
    LDI    temp3, ($01<<CLKC0)          ; CSEL="0001"
    STS    CLKCSR, temp2                ; CLKCSRレジスタ アクセス許可
    STS    CLKCSR, temp3                ; (*) クリスタル用発振器クロック禁止
; (*) !!! この瞬間に於いて、CLKSELRによって他の何れかのクロック元が選択された場合、このクロック元も停止します。!!!
; => 対策は、
    STS    CLKCSR, temp2

```

29. データシート改訂履歴

29.1. 改訂8265A – 2010年8月

1. 初版

29.2. 改訂8265B – 2010年9月

更新:

- ・ 55頁の「ポートAの交換機能」、ビット3とビット4
- ・ 73頁の「タイマ/カウンタ0制御レジスタ(TCCR0A)」、ビット1とビット0の説明
- ・ 188頁の「注文情報」、ATtiny167の注文コード
- ・ データシート全体を通してビット構文法、例えばCS02:0からCS0[2:0] (**訳補**:表記法が異なるので無視)

29.3. 改訂8265C – 2012年3月

更新:

- ・ 5頁の「メモリ容量要約」の表
- ・ データシートの状況を“暫定”から“完成”に変更
- ・ 188頁の「注文情報」、32NPによって32M1-Aを置き換え
- ・ 189頁の「外圍器情報」、32NPによって32M1-Aを置き換え
- ・ 青いロゴと最終頁での新アドレスを含む新しいAtmelの雛形を持つ全体資料

29.4. 改訂8265D – 2014年1月

更新:

- ・ Atmelの”青い新ロゴ”と住所でデータシートを更新 (**訳注**:本書では完全な雛形に沿って全てを更新)
- ・ 171頁の図23-2、172頁の図23-5、175頁の図23-11を周波数16MHzに従って更新
- ・ 188頁の「ATtiny167」の表で指定速度(周波数)を16MHzに修正

29.5. 改訂A – 2020年1月

1. 新しいMicrochip文書番号。以前版はAtmelデータシート改訂8265Dでした。
2. WQFN20用ピン配置図を追加
3. 「ATtiny87/167の割り込みベクタ」表を更新
4. 操作の正しい順番を強調するために「ポート生成器」の式に括弧を追加
5. 「注文情報」を更新
6. 「外圍器情報」をMicrochip形式に更新
7. 障害情報は独立した文書に移動されました。

開発支援

Microchipの受賞歴のある開発ツールで設計を構想から製造へ記録的な速さで行動してください。Microchipのツールはどの計画に対しても最新のデバッグを提供するために無料のMPLAB®XおよびAtmel Studio統合開発環境 (IDE) での使い易い画像使用者インターフェース (GUI) とコード生成ツールと共に動きます。究極に使い易い体験を提供するMicrochipの書き込み器、デバッグ、エミュレータは当社のソフトウェア ツールと共に継ぎ目なく動きます。Microchipの開発基板は応用に最適なシリコン デバイスの評価に役立ち、同時に当社の第三者のツール群は包括的な開発ツール解決策を完成させます。

MicrochipのMPLAB XとAtmel Studio協業系はPIC® MCU、AVR® MCU、SAM MCU、dsPIC® DSCのような多数のデバイスを支援する、考慮すべき様々な組み込み設計ツールを提供します。MPLAB XツールはWindows®、Linux®、Mac®オペレーティング システムに適合し、一方でAtmel StudioツールはWindowsに適合します。

より多くの情報と詳細については次のウェブサイトへ行ってください。

<https://www.microchip.com/development-tools/>

Microchipウェブサイト

Microchipはwww.microchip.comで当社のウェブサイト経由でのオンライン支援を提供します。このウェブサイトはお客様がファイルや情報を容易に利用可能にする手段として使われます。お気に入りのインターネット ブラウザを用いてアクセスすることができ、ウェブサイトは以下の情報を含みます。

- **製品支援** – データシートと障害情報、応用記述と試供プログラム、設計資源、使用者の手引きとハードウェア支援資料、最新ソフトウェア配布と保管されたソフトウェア
- **全般的な技術支援** – 良くある質問 (FAQ)、技術支援要求、オンライン検討グループ、Microchip相談役プログラム員一覧
- **Microchipの事業** – 製品選択器と注文の手引き、最新Microchip報道発表、セミナーとイベントの一覧、Microchip営業所の一覧、代理店と代表する工場

お客様支援

Microchip製品の使用者は以下のいくつかのチャネルを通して支援を受け取ることができます。

- 代理店または販売会社
- 最寄りの営業所
- 現場応用技術者 (FAE: Field Application Engineer)
- 技術支援

お客様は支援に関してこれらの代理店、販売会社、または現場応用技術者 (FAE) に連絡を取るべきです。最寄りの営業所もお客様の手助けに利用できます。営業所と位置の一覧はこの資料の後ろに含まれます。

技術支援は<http://microchip.com/support>でのウェブサイトを通して利用できます。

お客様への変更通知サービス

Microchipのお客様通知サービスはMicrochip製品を最新に保つのに役立ちます。加入者は指定した製品系統や興味のある開発ツールに関連する変更、更新、改訂、障害情報がある場合に必ず電子メール通知を受け取ります。

登録するにはwww.microchip.comでMicrochipのウェブサイトアクセスしてください。"Support" 下で "Customer Change Notification" をクリックして登録指示に従ってください。

Microchipデバイスでの以下のコード保護機能の以下の詳細に注意してください。

- Microchip製品はそれら特定のMicrochipデータシートに含まれる仕様に合致します。
- Microchipは意図した方法と通常条件下で使われる時に、その製品系統が今日の市場でその種類の最も安全な系統の1つであると考えます。
- コード保護機能を破るのに使われる不正でおそらく違法な方法があります。当社の知る限りこれらの方法の全てはMicrochipのデータシートに含まれた動作仕様外の方法でMicrochip製品を使うことが必要です。おそらく、それを行う人は知的財産の窃盗に関与しています。
- Microchipはそれらのコードの完全性について心配されているお客様と共に働きたいと思います。
- Microchipや他のどの半導体製造業者もそれらのコードの安全を保証することはできません。コード保護は当社が製品を"破ることができない"として保証すると言うことを意味しません。

コード保護は常に進化しています。Microchipは当社製品のコード保護機能を継続的に改善することを約束します。Microchipのコード保護機能を破る試みはデジタル ミレニアム著作権法に違反するかもしれません。そのような行為があなたのソフトウェアや他の著作物に不正なアクセスを許す場合、その法律下の救済のために訴権を持つかもしれません。

デバイス応用などに関してこの刊行物に含まれる情報は皆さまの便宜のためにだけ提供され、更新によって取り換えられるかもしれません。皆さまの応用が皆さまの仕様に合致するのを保証するのは皆さまの責任です。**Microchipはその条件、品質、性能、商品性、目的適合性を含め、明示的にも黙示的にもその情報に関連して書面または表記された書面または黙示の如何なる表明や保証もしません。**Microchipはこの情報とそれの使用から生じる全責任を否認します。生命維持や安全応用でのMicrochipデバイスの使用は完全に購入者の危険性で、購入者はそのような使用に起因する全ての損害、請求、訴訟、費用からMicrochipを擁護し、補償し、免責にすることに同意します。他に言及されない限り、Microchipのどの知的財産権下でも暗黙的または違う方法で許認可は譲渡されません。

商標

Microchipの名前とロゴ、Microchipロゴ、Adaptec、AnyRate、AVR、AVRロゴ、AVR Freaks、BesTime、BitCloud、chipKIT、chipKITロゴ、CryptoMemory、CryptoRF、dsPIC、FlashFlex、flexPWR、HELDO、IGLOO、JukeBlox、KeeLoq、Kleer、LANCheck、LinkMD、maXStylus、maXTouch、MediaLB、megaAVR、Microsemi、Microsemiロゴ、MOST、MOSTロゴ、MPLAB、OptoLyzer、PacTime、PIC、picoPower、PICSTART、PIC32ロゴ、PolarFire、Prochip Designer、QTouch、SAM-BA、SenGenuity、SpyNIC、SST、SSTロゴ、SuperFlash、Symmetricom、SyncServer、Tachyon、TempTrackr、TimeSource、tinyAVR、UNI/O、Vectron、XMEGAは米国と他の国に於けるMicrochip Technology Incorporatedの登録商標です。

APT、ClockWorks、The Embedded Control Solutions Company、EtherSynch、FlashTec、Hyper Speed Control、HyperLight Load、IntelliMOS、Libero、motorBench、mTouch、Powermite 3、Precision Edge、ProASIC、ProASIC Plus、ProASIC Plusロゴ、Quiet-Wire、SmartFusion、SyncWorld、Temux、TimeCesium、TimeHub、TimePictra、TimeProvider、Vite、WinPath、ZLは米国に於けるMicrochip Technology Incorporatedの登録商標です。

Adjacent Key Suppression、AKS、Analog-for-the-Digital Age、Any Capacitor、AnyIn、AnyOut、BlueSky、BodyCom、CodeGuard、CryptoAuthentication、CryptoCompanion、CryptoController、dsPICDEM、dsPICDEM.net、Dynamic Average Matching、DAM、ECAN、EtherGREEN、In-Circuit Serial Programming、ICSP、INICnet、Inter-Chip Connectivity、JitterBlocker、KleerNet、KleerNetロゴ、memBrain、Mindi、MiWi、MPASM、MPF、MPLAB Certifiedロゴ、MPLAB、MPLINK、MultiTRAK、NetDetach、Omniscient Code Generation、PICDEM、PICDEM.net、PICkit、Pictail、PowerSmart、PureSilicon、QMatrix、REAL ICE、Ripple Blocker、SAM-ICE、Serial Quad I/O、SMART-I.S.、SQI、SuperSwitcher、SuperSwitcher II、Total Endurance、TSHARC、USBCheck、VariSense、ViewSense、WiperLock、Wireless DNA、ZENAは米国と他の国に於けるMicrochip Technology Incorporatedの商標です。

SQTPは米国に於けるMicrochip Technology Incorporatedの役務標章です。

Adaptecロゴ、Frequency on Demand、Silicon Storage Technology、Symcomは他の国に於けるMicrochip Technology Inc.の登録商標です。

GestICは他の国に於けるMicrochip Technology Inc.の子会社であるMicrochip Technology Germany II GmbH & Co. KGの登録商標です。

ここで言及した以外の全ての商標はそれら各々の会社の所有物です。

© 2020年、Microchip Technology Incorporated、米国印刷、不許複製

Microchipの品質管理システムに関する情報についてはwww.microchip.com/qualityを訪ねてください。

日本語© HERO 2022.

本データシートはMicrochipのATtiny87/167英語版データシート(DS40002167A-2020年1月)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。

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。

世界的な販売とサービス

米国

本社

2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200

Fax: 480-792-7277

技術支援:

[http://www.microchip.com/
support](http://www.microchip.com/support)

ウェブアドレス:

www.microchip.com

アトランタ

Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

オースチン TX

Tel: 512-257-3370

ボストン

Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

シカゴ

Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

ダラス

Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

デトロイト

Novi, MI
Tel: 248-848-4000

ヒューストン TX

Tel: 281-894-5983

インディアナポリス

Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453
Tel: 317-536-2380

ロサンゼルス

Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608
Tel: 951-273-7800

ローリー NC

Tel: 919-844-7510

ニューヨーク NY

Tel: 631-435-6000

サンホセ CA

Tel: 408-735-9110
Tel: 408-436-4270

カナダ - トロント

Tel: 905-695-1980
Fax: 905-695-2078

亜細亜/太平洋

オーストラリア - シドニー

Tel: 61-2-9868-6733

中国 - 北京

Tel: 86-10-8569-7000

中国 - 成都

Tel: 86-28-8665-5511

中国 - 重慶

Tel: 86-23-8980-9588

中国 - 東莞

Tel: 86-769-8702-9880

中国 - 広州

Tel: 86-20-8755-8029

中国 - 杭州

Tel: 86-571-8792-8115

中国 - 香港特别行政区

Tel: 852-2943-5100

中国 - 南京

Tel: 86-25-8473-2460

中国 - 青島

Tel: 86-532-8502-7355

中国 - 上海

Tel: 86-21-3326-8000

中国 - 瀋陽

Tel: 86-24-2334-2829

中国 - 深圳

Tel: 86-755-8864-2200

中国 - 蘇州

Tel: 86-186-6233-1526

中国 - 武漢

Tel: 86-27-5980-5300

中国 - 西安

Tel: 86-29-8833-7252

中国 - 廈門

Tel: 86-592-2388138

中国 - 珠海

Tel: 86-756-3210040

亜細亜/太平洋

インド - ハンガロール

Tel: 91-80-3090-4444

インド - ニューデリー

Tel: 91-11-4160-8631

インド - プネー

Tel: 91-20-4121-0141

日本 - 大阪

Tel: 81-6-6152-7160

日本 - 東京

Tel: 81-3-6880-3770

韓国 - 大邱

Tel: 82-53-744-4301

韓国 - ソウル

Tel: 82-2-554-7200

マレーシア - クアラルンプール

Tel: 60-3-7651-7906

マレーシア - ペナン

Tel: 60-4-227-8870

フィリピン - マニラ

Tel: 63-2-634-9065

シンガポール

Tel: 65-6334-8870

台湾 - 新竹

Tel: 886-3-577-8366

台湾 - 高雄

Tel: 886-7-213-7830

台湾 - 台北

Tel: 886-2-2508-8600

タイ - バンコク

Tel: 66-2-694-1351

ベトナム - ホーチミン

Tel: 84-28-5448-2100

欧州

オーストラリア - ウェルズ

Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

デンマーク - コペンハーゲン

Tel: 45-4450-2828
Fax: 45-4485-2829

フィンランド - エスポー

Tel: 358-9-4520-820

フランス - パリ

Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

ドイツ - ガルピング

Tel: 49-8931-9700

ドイツ - ハーネ

Tel: 49-2129-3766400

ドイツ - ハイムブロン

Tel: 49-7131-72400

ドイツ - カールスルーエ

Tel: 49-721-625370

ドイツ - ミュンヘン

Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

ドイツ - ローゼンハイム

Tel: 49-8031-354-560

イスラエル - ラーナナ

Tel: 972-9-744-7705

イタリア - ミラノ

Tel: 39-0331-742611
Fax: 39-0331-466781

イタリア - バドパ

Tel: 39-049-7625286

オランダ - デルフト

Tel: 31-416-690399
Fax: 31-416-690340

ノルウェー - トロンハイム

Tel: 47-72884388

ポーランド - ワルシャワ

Tel: 48-22-3325737

ルーマニア - ブカレスト

Tel: 40-21-407-87-50

スペイン - マドリード

Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

スウェーデン - イェテボリ

Tel: 46-31-704-60-40

スウェーデン - ストックホルム

Tel: 46-8-5090-4654

イギリス - ウォーキングム

Tel: 44-118-921-5800
Fax: 44-118-921-5820