

お客様各位

---

## カタログ等資料中の旧社名の扱いについて

---

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願ひ申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日

ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】 <http://japan.renesas.com/inquiry>

## ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りが無いことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。  
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット  
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）  
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注 1. 本資料において使用されている「当社」とは、ルネサスエレクトロニクス株式会社およびルネサスエレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

お客様各位

---

## 資料中の「日立製作所」、「日立XX」等名称の株式会社ルネサス テクノロジへの変更について

---

2003年4月1日を以って三菱電機株式会社及び株式会社日立製作所のマイコン、ロジック、アナログ、ディスクリート半導体、及びDRAMを除くメモリ(フラッシュメモリ・SRAM等)を含む半導体事業は株式会社ルネサス テクノロジに承継されました。従いまして、本資料中には「日立製作所」、「株式会社日立製作所」、「日立半導体」、「日立XX」といった表記が残っておりますが、これらの表記は全て「株式会社ルネサス テクノロジ」に変更されておりますのでご理解の程お願い致します。尚、会社商標・ロゴ・コーポレートステートメント以外の内容については一切変更しておりませんので資料としての内容更新ではありません。

ルネサステクノロジ ホームページ (<http://www.renesas.com>)

2003年4月1日  
株式会社ルネサス テクノロジ  
カスタマサポート部

## ご注意

### 安全設計に関するお願い

1. 弊社は品質、信頼性の向上に努めておりますが、半導体製品は故障が発生したり、誤動作する場合があります。弊社の半導体製品の故障又は誤動作によって結果として、人身事故、火災事故、社会的損害などを生じさせないような安全性を考慮した冗長設計、延焼対策設計、誤動作防止設計などの安全設計に十分ご注意ください。

### 本資料ご利用に際しての留意事項

1. 本資料は、お客様が用途に応じた適切なルネサス テクノロジ製品をご購入いただくための参考資料であり、本資料中に記載の技術情報についてルネサス テクノロジが所有する知的財産権その他の権利の実施、使用を許諾するものではありません。
2. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例の使用に起因する損害、第三者所有の権利に対する侵害に関し、ルネサス テクノロジは責任を負いません。
3. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他全ての情報は本資料発行時点のものであり、ルネサス テクノロジは、予告なしに、本資料に記載した製品または仕様を変更することがあります。ルネサス テクノロジ半導体製品のご購入に当たりましては、事前にルネサス テクノロジ、ルネサス販売または特約店へ最新の情報をご確認頂きますとともに、ルネサス テクノロジホームページ (<http://www.renesas.com>)などを通じて公開される情報に常にご注意ください。
4. 本資料に記載した情報は、正確を期すため、慎重に制作したのですが万一本資料の記述誤りに起因する損害がお客様に生じた場合には、ルネサス テクノロジはその責任を負いません。
5. 本資料に記載の製品データ、図、表に示す技術的な内容、プログラム及びアルゴリズムを流用する場合は、技術内容、プログラム、アルゴリズム単位で評価するだけでなく、システム全体で十分に評価し、お客様の責任において適用可否を判断してください。ルネサス テクノロジは、適用可否に対する責任を負いません。
6. 本資料に記載された製品は、人命にかかわるような状況の下で使用される機器あるいはシステムに用いられることを目的として設計、製造されたものではありません。本資料に記載の製品を運輸、移動体用、医療用、航空宇宙用、原子力制御用、海中継用機器あるいはシステムなど、特殊用途へのご利用をご検討の際には、ルネサス テクノロジ、ルネサス販売または特約店へご照会ください。
7. 本資料の転載、複製については、文書によるルネサス テクノロジの事前の承諾が必要です。
8. 本資料に関し詳細についてのお問い合わせ、その他お気付きの点がございましたらルネサス テクノロジ、ルネサス販売または特約店までご照会ください。

# H8/3627 シリーズ

## ハードウェアマニュアル

H8/3627	HD6473627, HD6433627
H8/3626	HD6433626
H8/3625	HD6433625
H8/3624S	HD6433624S
H8/3623S	HD6433623S
H8/3622S	HD6433622S

## ご注意

1. 本書に記載の製品及び技術のうち「外国為替及び外国貿易法」に基づき安全 保障貿易管理関連貨物・技術に該当するものを輸出する場合，または国外に持ち出す場合は日本国政府の許可が必要です。
2. 本書に記載された情報の使用に際して，弊社もしくは第三者の特許権，著作権，商標権，その他の知的所有権等の権利に対する保証または実施権の許諾を行うものではありません。また本書に記載された情報を使用した事により第三者の知的所有権等の権利に関わる問題が生じた場合，弊社はその責を負いませんので予めご了承ください。
3. 製品及び製品仕様は予告無く変更する場合がありますので，最終的な設計，ご購入，ご使用に際しましては，事前に最新の製品規格または仕様書をお求めになりご確認ください。
4. 弊社は品質・信頼性の向上に努めておりますが，宇宙，航空，原子力，燃焼制御，運輸，交通，各種安全装置，ライフサポート関連の医療機器等のように，特別な品質・信頼性が要求され，その故障や誤動作が直接人命を脅かしたり，人体に危害を及ぼす恐れのある用途にご使用をお考えのお客様は，事前に弊社営業担当迄ご相談をお願い致します。
5. 設計に際しては，特に最大定格，動作電源電圧範囲，放熱特性，実装条件及びその他諸条件につきましては，弊社保証範囲内でご使用いただきますようお願い致します。  
保証値を越えてご使用された場合の故障及び事故につきましては，弊社はその責を負いません。  
また保証値内のご使用であっても半導体製品について通常予測される故障発生率，故障モードをご考慮の上，弊社製品の動作が原因でご使用機器が人身事故，火災事故，その他の拡大損害を生じないようにフェールセーフ等のシステム上の対策を講じて頂きますようお願い致します。
6. 本製品は耐放射線設計をしておりません。
7. 本書の一部または全部を弊社の文書による承認なしに転載または複製することを堅くお断り致します。
8. 本書をはじめ弊社半導体についてのお問い合わせ，ご相談は弊社営業担当迄お願い致します。

---

# はじめに

---

H8/300L シリーズは、高速 H8/300L CPU を核にして、システム構成に必要な周辺機能を集積したシングルチップマイクロコンピュータです。

H8/300L CPU は、H8/300CPU と互換性のある命令体系を備えています。

H8/3627 シリーズは、システム構成に必要な周辺機能として、高精度 DTMF (トーンダイヤル) 発生回路、3 種類のタイマ、2 チャンネルのシリアルコミュニケーションインタフェース、A/D 変換器を内蔵しています。

本マニュアルは、H8/3627 シリーズのハードウェアについて記載しています。命令の詳細については、「H8/300L シリーズ プログラミングマニュアル」をあわせてご覧ください。

---

# 目次

---

## 第1章 概要

1.1	概要	3	
1.2	内部ブロック図	6	
1.3	端子説明	7	
	1.3.1	ピン配置	7
	1.3.2	端子機能	8

## 第2章 CPU

2.1	概要	15	
	2.1.1	特長	15
	2.1.2	アドレス空間	16
	2.1.3	レジスタ構成	16
2.2	各レジスタの説明	17	
	2.2.1	汎用レジスタ	17
	2.2.2	コントロールレジスタ	17
	2.2.3	CPU 内部レジスタの初期値	19
2.3	データ構成	20	
	2.3.1	汎用レジスタのデータ構成	21
	2.3.2	メモリ上でのデータ構成	22
2.4	アドレッシングモード	23	
	2.4.1	アドレッシングモード	23
	2.4.2	実効アドレスの計算方法	25
2.5	命令セット	29	
	2.5.1	データ転送命令	31
	2.5.2	算術演算命令	33
	2.5.3	論理演算命令	34
	2.5.4	シフト命令	34

2.5.5	ビット操作命令.....	36
2.5.6	分岐命令.....	40
2.5.7	システム制御命令.....	42
2.5.8	ブロック転送命令.....	43
2.6	基本動作タイミング.....	44
2.6.1	内蔵メモリ (RAM、ROM).....	44
2.6.2	内蔵周辺モジュール.....	45
2.7	CPUの状態.....	47
2.7.1	概要.....	47
2.7.2	プログラム実行状態.....	48
2.7.3	プログラム停止状態.....	48
2.7.4	例外処理状態.....	48
2.8	メモリマップ.....	49
2.9	使用上の注意事項.....	50
2.9.1	データアクセスに関する注意事項.....	50
2.9.2	ビット操作命令使用上の注意事項.....	52
2.9.3	EEPMOV 命令使用上の注意事項.....	58

### 第3章 例外処理

3.1	概要.....	61
3.2	リセット.....	62
3.2.1	概要.....	62
3.2.2	リセットシーケンス.....	62
3.2.3	リセット直後の割込み.....	64
3.3	割込み.....	65
3.3.1	概要.....	65
3.3.2	各レジスタの説明.....	67
3.3.3	外部割込み.....	76
3.3.4	内部割込み.....	77
3.3.5	割込み動作.....	77
3.3.6	割込み応答時間.....	82
3.4	使用上の注意事項.....	83
3.4.1	スタック領域に関する使用上の注意事項.....	83
3.4.2	ポートモードレジスタを書換える際の注意事項.....	84

## 第4章 クロック発振器

4.1	概要	89
4.1.1	ブロック図	89
4.1.2	システムクロックとサブクロック	89
4.2	システムクロック発振器	90
4.3	サブクロック発振器	93
4.4	プリスケアラ	94
4.5	発振子に関する注意事項	95

## 第5章 低消費電力モード

5.1	概要	99
5.1.1	システムコントロールレジスタ	102
5.2	スリープモード	106
5.2.1	スリープモードへの遷移	106
5.2.2	スリープモードの解除	106
5.3	スタンバイモード	107
5.3.1	スタンバイモードへの遷移	107
5.3.2	スタンバイモードの解除	107
5.3.3	スタンバイモード解除後の発振安定時間の設定	108
5.3.4	スタンバイモードへの遷移と端子状態	108
5.4	ウォッチモード	109
5.4.1	ウォッチモードへの遷移	109
5.4.2	ウォッチモードの解除	109
5.4.3	ウォッチモード解除後の発振安定時間の設定	109
5.5	サブスリープモード	110
5.5.1	サブスリープモードへの遷移	110
5.5.2	サブスリープモードの解除	110
5.6	サブアクティブモード	111
5.6.1	サブアクティブモードへの遷移	111
5.6.2	サブアクティブモードの解除	111
5.6.3	サブアクティブモードの動作周波数について	111
5.7	アクティブ(中速)モード	112
5.7.1	アクティブ(中速)モードへの遷移	112
5.7.2	アクティブ(中速)モードの解除	112
5.7.3	アクティブ(中速)モードの動作周波数について	112
5.8	直接遷移	113
5.8.1	直接遷移の概要	113

5.8.2	直接遷移の時間.....	114
-------	--------------	-----

## 第6章 ROM

6.1	概要 .....	119
6.1.1	ブロック図 .....	119
6.2	PROMモード.....	120
6.2.1	PROMモードの設定.....	120
6.2.2	ソケットアダプタの端子対応とメモリマップ .....	120
6.3	プログラミング.....	123
6.3.1	書込み/ベリファイ .....	124
6.3.2	書込み時の注意.....	126
6.4	書込み後の信頼性 .....	128

## 第7章 RAM

7.1	概要 .....	131
7.1.1	ブロック図 .....	131

## 第8章 I/Oポート

8.1	概要 .....	135
8.2	ポート1.....	137
8.2.1	概要 .....	137
8.2.2	レジスタの構成と説明.....	137
8.2.3	端子機能.....	142
8.2.4	端子状態.....	143
8.2.5	入力プルアップMOS.....	144
8.3	ポート2.....	145
8.3.1	概要 .....	145
8.3.2	レジスタの構成と説明.....	145
8.3.3	端子機能.....	150
8.3.4	端子状態.....	152
8.3.5	入力プルアップMOS.....	152
8.4	ポート5.....	153
8.4.1	概要 .....	153
8.4.2	レジスタの構成と説明.....	153
8.4.3	端子機能.....	155

	8.4.4	端子状態.....	156
	8.4.5	入力プルアップ MOS.....	156
8.5	ポート 6.....		157
	8.5.1	概要.....	157
	8.5.2	レジスタの構成と説明.....	157
	8.5.3	端子機能.....	159
	8.5.4	端子状態.....	159
	8.5.5	入力プルアップ MOS.....	159
8.6	ポート 7.....		160
	8.6.1	概要.....	160
	8.6.2	レジスタの構成と説明.....	160
	8.6.3	端子機能.....	162
	8.6.4	端子状態.....	162
8.7	ポート 8.....		163
	8.7.1	概要.....	163
	8.7.2	レジスタの構成と説明.....	163
	8.7.3	端子機能.....	165
	8.7.4	端子状態.....	165
8.8	ポート A.....		166
	8.8.1	概要.....	166
	8.8.2	レジスタの構成と説明.....	166
	8.8.3	端子機能.....	167
	8.8.4	端子状態.....	167
8.9	ポート B.....		168
	8.9.1	概要.....	168
	8.9.2	レジスタの構成と説明.....	168

## 第9章 タイマ

9.1	概要.....	171	
9.2	タイマ A.....	172	
	9.2.1	概要.....	172
	9.2.2	各レジスタの説明.....	174
	9.2.3	動作説明.....	176
	9.2.4	タイマ A の動作モード.....	177
9.3	タイマ F.....	178	
	9.3.1	概要.....	178
	9.3.2	各レジスタの説明.....	181

	9.3.3	CPUとのインタフェース.....	188
	9.3.4	動作説明.....	190
	9.3.5	使用上の注意事項.....	193
9.4		タイマG.....	195
	9.4.1	概要.....	195
	9.4.2	各レジスタの説明.....	197
	9.4.3	ノイズ除去回路.....	201
	9.4.4	動作説明.....	202
	9.4.5	使用上の注意事項.....	207
	9.4.6	タイマGの使用例.....	211

## 第10章 シリアルコミュニケーションインタフェース

10.1		概要.....	215
10.2		SCI1.....	216
	10.2.1	概要.....	216
	10.2.2	各レジスタの説明.....	218
	10.2.3	動作説明.....	222
	10.2.4	割込み要因.....	224
10.3		SCI3.....	225
	10.3.1	概要.....	225
	10.3.2	各レジスタの説明.....	228
	10.3.3	動作概要.....	245
	10.3.4	調歩同期式モード時の動作説明.....	249
	10.3.5	クロック同期式モード時の動作説明.....	258
	10.3.6	マルチプロセッサ通信機能.....	265
	10.3.7	割込み要因.....	270
	10.3.8	使用上の注意事項.....	271

## 第11章 DTMF 発生回路

11.1		概要.....	277
	11.1.1	特長.....	277
	11.1.2	ブロック図.....	278
	11.1.3	端子構成.....	278
	11.1.4	レジスタ構成.....	279
11.2		各レジスタの説明.....	280
	11.2.1	DTMF コントロールレジスタ ( DTCR ) .....	280

	11.2.2	DTMF ロードレジスタ ( DTLR ) .....	282
11.3		動作説明.....	283
	11.3.1	出力波形.....	283
	11.3.2	動作フロー .....	284
11.4		応用回路例 .....	285
11.5		使用上の注意.....	286

## 第 12 章 A/D 変換器

12.1		概要 .....	289
	12.1.1	特長 .....	289
	12.1.2	ブロック図 .....	290
	12.1.3	端子構成.....	291
	12.1.4	レジスタ構成.....	291
12.2		各レジスタの説明 .....	292
	12.2.1	A/D リザルトレジスタ ( ADRR ) .....	292
	12.2.2	A/D モードレジスタ ( AMR ) .....	292
	12.2.3	A/D スタートレジスタ ( ADSR ) .....	294
12.3		動作説明.....	295
	12.3.1	A/D 変換動作 .....	295
	12.3.2	外部トリガによる A/D 変換器の起動 .....	295
12.4		割込み要因 .....	296
12.5		使用例.....	297
12.6		使用上の注意.....	301

## 第 13 章 電源回路

13.1		概要 .....	305
13.2		内部電源降圧回路構成.....	305

## 第 14 章 電気的特性

14.1		絶対最大定格.....	309
14.2		電気的特性 .....	310
	14.2.1	電源電圧と動作範囲.....	310
	14.2.2	DC 特性.....	312
	14.2.3	AC 特性.....	317
	14.2.4	A/D 変換器特性.....	319

	14.2.5	DTMF 特性 .....	320
14.3		動作タイミング .....	321
14.4		出力負荷回路 .....	324

## 付録

A.		命令 .....	327
	A.1	命令一覧 .....	327
	A.2	オペレーションコードマップ .....	337
	A.3	命令実行ステート数 .....	338
B.		内部 I/O レジスタ一覧 .....	344
	B.1	アドレス一覧 .....	344
	B.2	機能一覧 .....	348
C.		I/O ポートブロック図 .....	385
	C.1	ポート 1 ブロック図 .....	385
	C.2	ポート 2 ブロック図 .....	392
	C.3	ポート 5 ブロック図 .....	400
	C.4	ポート 6 ブロック図 .....	401
	C.5	ポート 7 ブロック図 .....	402
	C.6	ポート 8 ブロック図 .....	403
	C.7	ポート A ブロック図 .....	404
	C.8	ポート B ブロック図 .....	405
D.		各処理状態におけるポートの状態 .....	406
E.		ROM 発注手順 .....	407
	E.1	ROM 書換え品開発の流れ (発注手順) .....	407
	E.2	ROM 発注時の注意事項 .....	408
F.		型名一覧 .....	409
G.		外形寸法図 .....	410

---

# 1. 概要

---

## 第1章 目次

1.1	概要.....	3
1.2	内部ブロック図.....	6
1.3	端子説明.....	7
	1.3.1    ピン配置.....	7
	1.3.2    端子機能.....	8



## 1.1 概要

H8/300L シリーズは、高速 H8/300L CPU を核にして、システム構成に必要な周辺機能を集積したシングルチップマイクロコンピュータ（MCU：Microcomputer Unit）です。

H8/3627 シリーズは、高精度 DTMF（トーンダイアル）発生回路を内蔵した H8/300L シリーズのシングルチップマイクロコンピュータで、その他の周辺機能として、3種類のタイマ、2チャンネルのシリアルコミュニケーションインタフェース、A/D変換器などを内蔵しています。内蔵メモリは、ROM60k / 48k / 40k バイト（H8/3627、H8/3626、H8/3625）、RAM2k バイト版、ROM32k / 24k / 16k バイト（H8/3624S、H8/3623S、H8/3622S）、RAM1k バイト版が用意されています。また、電源回路改善により、低消費電力、低放射ノイズを実現した製品です。

H8/3627 には、ユーザサイドで自由にプログラムの書込みができる PROM を内蔵した ZTAT<sup>®</sup>\*版もあります。

H8/3627 シリーズの特長を表 1.1 に示します。

【注】 \* ZTAT は（株）日立製作所の登録商標です。

表 1.1 特長（1）

項目	仕様
CPU	高速 H8/300L CPU (1) 汎用レジスタ方式 ・汎用レジスタ：8ビット×16本 (16ビット×8本としても使用可能) (2) 高速演算 ・最高動作周波数：5MHz ・加減算：0.4μs（=5MHz動作時） ・乗除算：2.8μs（=5MHz動作時） ・32.768kHz サブクロックによる動作可能 (3) H8/300CPU と互換性のある命令体系 ・命令フォーマットは2バイトまたは4バイト長 ・基本演算はレジスタ - レジスタ間で実行 ・MOV 命令によるメモリ - レジスタ間データ転送 (4) 特長ある命令 ・乗算命令（8ビット×8ビット） ・除算命令（16ビット÷8ビット） ・ビットアキュムレータ命令 ・レジスタ間接指定によりビット位置の指定が可能
割込み	29種類の割込み要因 ・外部割込み要因：13要因（IRQ <sub>4</sub> ～IRQ <sub>0</sub> 、WKP <sub>7</sub> ～WKP <sub>0</sub> ） ・内部割込み要因：16要因

表 1.1 特長 (2)

項目	仕様
クロック発振器	2種類のクロック発振器内蔵 <ul style="list-style-type: none"> <li>・システムクロック発振器：1～10MHz</li> <li>・サブクロック発振器：32.768kHz</li> </ul>
低消費電力モード	6種類の低消費電力モード <ul style="list-style-type: none"> <li>・スリープモード</li> <li>・スタンバイモード</li> <li>・ウォッチモード</li> <li>・サブスリープモード</li> <li>・サブアクティブモード</li> <li>・アクティブ(中速)モード</li> </ul>
メモリ	大容量メモリ内蔵 H8/3627   ・ROM：60k バイト・RAM：2k バイト H8/3626   ・ROM：48k バイト・RAM：2k バイト H8/3625   ・ROM：40k バイト・RAM：2k バイト H8/3624S   ・ROM：32k バイト・RAM：1k バイト H8/3623S   ・ROM：24k バイト・RAM：1k バイト H8/3622S   ・ROM：16k バイト・RAM：1k バイト
I/O ポート	I/O ポート 53 本 <ul style="list-style-type: none"> <li>・入出力端子：50 本</li> <li>・入力端子   ：3 本</li> </ul>
タイマ	3種類のタイマ内蔵 (1) タイマA：インターバル/時計用タイムベース機能を内蔵した8ビットのタイマ <ul style="list-style-type: none"> <li>・システムクロック( )*を分周した8種類の内部クロックまたは時計用クロック( w)*を分周した4種類のクロックによりカウントアップ可能</li> </ul> (2) タイマF：アウトプットコンペア機能を内蔵した16ビットのタイマ <ul style="list-style-type: none"> <li>・独立した2本の8ビットタイマとして使用可能</li> <li>・4種類の内部クロックまたは外部端子からのイベント入力によりカウントアップ可能</li> <li>・コンペアマッチ機能によりトグル出力可能</li> </ul> (3) タイマG：インプットキャプチャ/インターバル機能を内蔵した8ビットのタイマ <ul style="list-style-type: none"> <li>・4種類の内部クロックによりカウントアップ可能</li> <li>・インプットキャプチャ機能内蔵(ノイズ除去回路内蔵)</li> </ul>

【注】 \*、 w の定義は「第4章 クロック発振器」を参照してください。

表 1.1 特長 (3)

項目	仕様																																																						
シリアル コミュニケーション インタフェース	2チャンネルのシリアルコミュニケーションインタフェース内蔵 (1) SCI1 : クロック同期式 ・ 8ビット / 16ビットの転送データを選択可能 (2) SCI3 : 8ビットクロック同期式 / 調歩同期式 ・ マルチプロセッサ通信機能内蔵																																																						
A/D 変換器	抵抗ラダー方式による逐次比較方式の 8ビット A/D 変換器 ・ 2チャンネルのアナログ入力端子 ・ 変換時間 : 1チャンネル当たり 31/、62/ または 124/																																																						
DTMF 発生回路	OSC クロック (1.2MHz ~ 10MHz、400kHz 刻み) 対応のトーンダイアル内蔵																																																						
製品ラインアップ	<table border="1"> <thead> <tr> <th colspan="2">製品型名</th> <th rowspan="2">パッケージ</th> <th rowspan="2">ROM / RAMサイズ</th> </tr> <tr> <th>マスクROM版</th> <th>ZTAT<sup>®</sup>版</th> </tr> </thead> <tbody> <tr> <td>HD6433627H</td> <td>HD6473627H</td> <td>64ピンQFP (FP-64A)</td> <td>ROM60kバイト</td> </tr> <tr> <td>HD6433627FP</td> <td>HD6473627FP</td> <td>64ピンLQFP (FP-64E)</td> <td>RAM2kバイト</td> </tr> <tr> <td>HD6433626H</td> <td></td> <td>64ピンQFP (FP-64A)</td> <td>ROM48kバイト</td> </tr> <tr> <td>HD6433626FP</td> <td></td> <td>64ピンLQFP (FP-64E)</td> <td>RAM2kバイト</td> </tr> <tr> <td>HD6433625H</td> <td></td> <td>64ピンQFP (FP-64A)</td> <td>ROM40kバイト</td> </tr> <tr> <td>HD6433625FP</td> <td></td> <td>64ピンLQFP (FP-64E)</td> <td>RAM2kバイト</td> </tr> <tr> <td>HD6433624SH</td> <td></td> <td>64ピンQFP (FP-64A)</td> <td>ROM32kバイト</td> </tr> <tr> <td>HD6433624SFP</td> <td></td> <td>64ピンLQFP (FP-64E)</td> <td>RAM1kバイト</td> </tr> <tr> <td>HD6433623SH</td> <td></td> <td>64ピンQFP (FP-64A)</td> <td>ROM24kバイト</td> </tr> <tr> <td>HD6433623SFP</td> <td></td> <td>64ピンLQFP (FP-64E)</td> <td>RAM1kバイト</td> </tr> <tr> <td>HD6433622SH</td> <td></td> <td>64ピンQFP (FP-64A)</td> <td>ROM16kバイト</td> </tr> <tr> <td>HD6433622SFP</td> <td></td> <td>64ピンLQFP (FP-64E)</td> <td>RAM1kバイト</td> </tr> </tbody> </table>	製品型名		パッケージ	ROM / RAMサイズ	マスクROM版	ZTAT <sup>®</sup> 版	HD6433627H	HD6473627H	64ピンQFP (FP-64A)	ROM60kバイト	HD6433627FP	HD6473627FP	64ピンLQFP (FP-64E)	RAM2kバイト	HD6433626H		64ピンQFP (FP-64A)	ROM48kバイト	HD6433626FP		64ピンLQFP (FP-64E)	RAM2kバイト	HD6433625H		64ピンQFP (FP-64A)	ROM40kバイト	HD6433625FP		64ピンLQFP (FP-64E)	RAM2kバイト	HD6433624SH		64ピンQFP (FP-64A)	ROM32kバイト	HD6433624SFP		64ピンLQFP (FP-64E)	RAM1kバイト	HD6433623SH		64ピンQFP (FP-64A)	ROM24kバイト	HD6433623SFP		64ピンLQFP (FP-64E)	RAM1kバイト	HD6433622SH		64ピンQFP (FP-64A)	ROM16kバイト	HD6433622SFP		64ピンLQFP (FP-64E)	RAM1kバイト
製品型名		パッケージ	ROM / RAMサイズ																																																				
マスクROM版	ZTAT <sup>®</sup> 版																																																						
HD6433627H	HD6473627H	64ピンQFP (FP-64A)	ROM60kバイト																																																				
HD6433627FP	HD6473627FP	64ピンLQFP (FP-64E)	RAM2kバイト																																																				
HD6433626H		64ピンQFP (FP-64A)	ROM48kバイト																																																				
HD6433626FP		64ピンLQFP (FP-64E)	RAM2kバイト																																																				
HD6433625H		64ピンQFP (FP-64A)	ROM40kバイト																																																				
HD6433625FP		64ピンLQFP (FP-64E)	RAM2kバイト																																																				
HD6433624SH		64ピンQFP (FP-64A)	ROM32kバイト																																																				
HD6433624SFP		64ピンLQFP (FP-64E)	RAM1kバイト																																																				
HD6433623SH		64ピンQFP (FP-64A)	ROM24kバイト																																																				
HD6433623SFP		64ピンLQFP (FP-64E)	RAM1kバイト																																																				
HD6433622SH		64ピンQFP (FP-64A)	ROM16kバイト																																																				
HD6433622SFP		64ピンLQFP (FP-64E)	RAM1kバイト																																																				

## 1.2 内部ブロック図

H8/3627 シリーズの内部ブロック図を図 1.1 に示します。

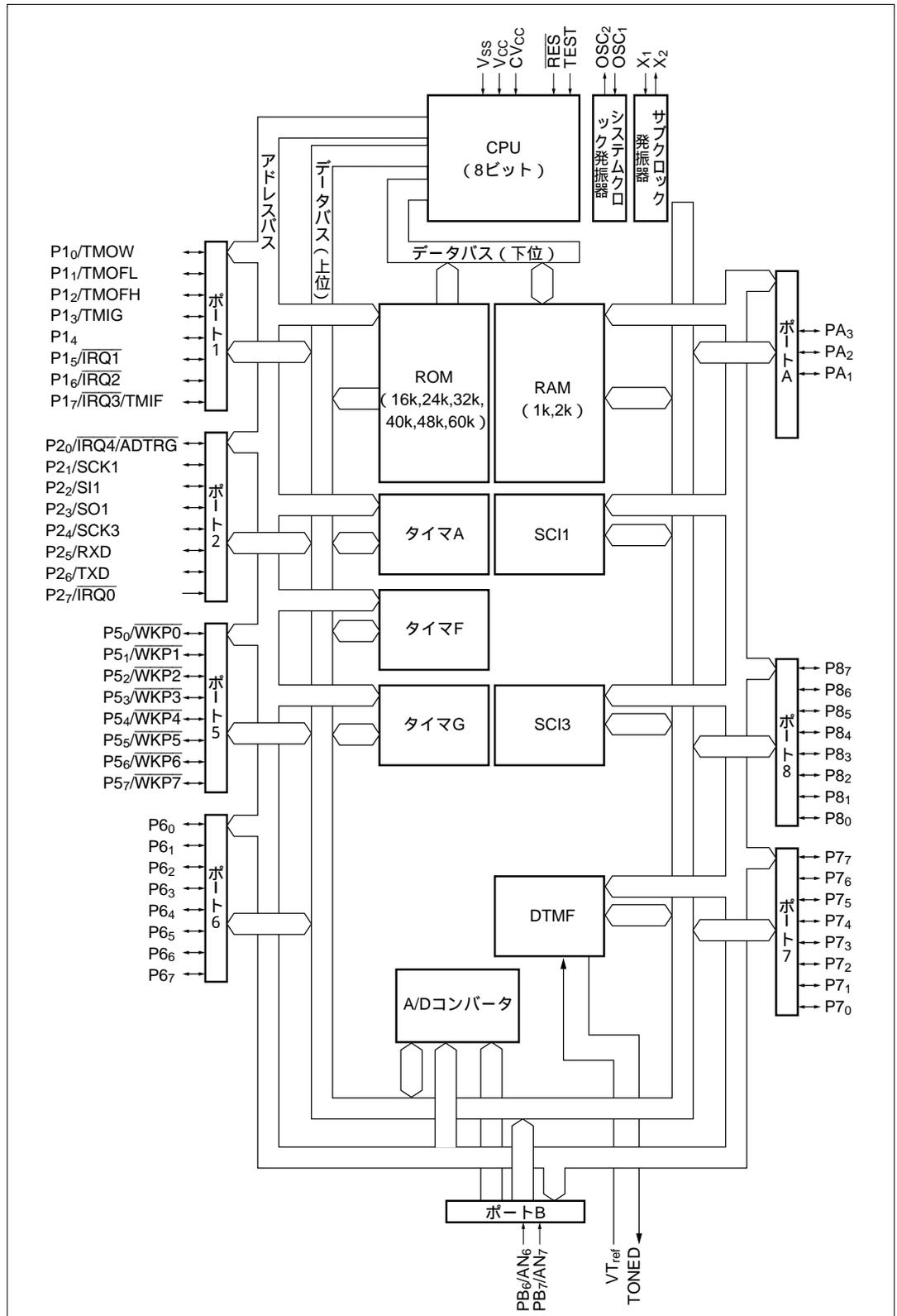


図 1.1 内部ブロック図

## 1.3 端子説明

### 1.3.1 ピン配置

H8/3627 シリーズのピン配置図を図 1.2 に示します。

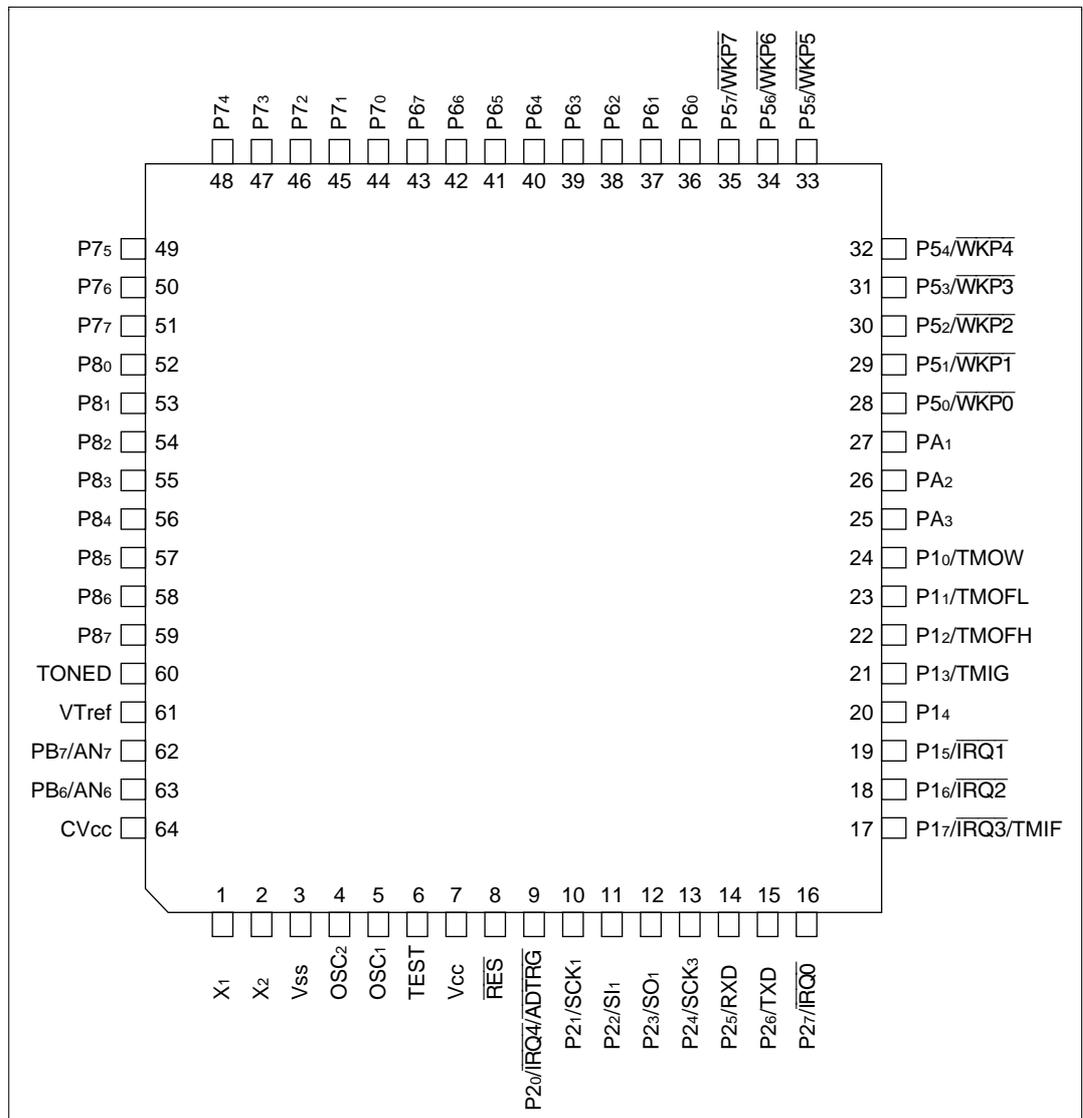


図 1.2 ピン配置図 (FP-64A、FP-64E : 上面図)

## 1.3.2 端子機能

## (1) 端子機能

各端子の機能について表 1.2 に示します。

表 1.2 端子機能 (1)

分類	記号	ピン番号		入出力	機能
		FP-64A	FP-64E		
電源	$V_{CC}$	7		入力	電源 $V_{CC}$ 端子は、システムの電源 (+5V) に接続してください。
	$V_{SS}$	3		入力	グランド $V_{SS}$ 端子は、システムの電源 (0V) に接続してください。
	$CV_{CC}$	64		入力	電源 $CV_{CC}$ 端子は、安定化容量 0.1 $\mu$ F をグランドの間に接続してください。
	$VTref$	61		入力	DTMF 発生回路基準レベル電源 DTMF 出力の基準レベル電源端子です。
クロック	$OSC_1$	5		入力	水晶発振子またはセラミック発振子を接続します。また、外部クロックを入力することもできます。接続例については「第 4 章 クロック発振器」を参照してください。
	$OSC_2$	4		出力	
	$X_1$	1		入力	
	$X_2$	2		出力	
システム制御	$\overline{RES}$	8		入力	リセット この端子を "Low" レベルにすると、リセット状態になります。
	TEST	6		入力	テスト端子 ユーザは、使用できません。 $V_{SS}$ 電位に接地してください。
割込み	$\overline{IRQ}_0$	16		入力	外部割込み要求 4~0 立上がりエッジセンス / 立下がりエッジセンスを選択可能な外部割込み入力端子です。
	$\overline{IRQ}_1$	19			
	$\overline{IRQ}_2$	18			
	$\overline{IRQ}_3$	17			
	$\overline{IRQ}_4$	9			

表 1.2 端子機能 (2)

分類	記号	ピン番号	入出力	機能
		FP-64A FP-64E		
割込み	$\overline{WKP}_7 \sim \overline{WKP}_0$	35 ~ 28	入力	<u>ウェイクアップ割込み要求 7~0</u> 立下がりエッジセンスの外部割込み入力端子です。
タイマ	TMOW	24	出力	<u>クロック出力</u> タイマ A 出力回路により生成された波形の出力端子です。
	TMIF	17	入力	<u>タイマ F イベント入力</u> タイマ F のカウンタに入力するイベント入力端子です。
	TMOFL	23	出力	<u>タイマ FL 出力</u> タイマ FL アウトプットコンペア機能により生成された波形の出力端子です。
	TMOFH	22	出力	<u>タイマ FH 出力</u> タイマ FH アウトプットコンペア機能により生成された波形の出力端子です。
	TMIG	21	入力	<u>タイマ G キャプチャ入力</u> タイマ G のインプットキャプチャの入力端子です。
I/O ポート	$PB_7, PB_6$	62, 63	入力	<u>ポート B</u> 2 ビットの入力端子です。
	$PA_3 \sim PA_1$	25 ~ 27	入出力	<u>ポート A</u> 3 ビットの入出力端子です。ポートコントロールレジスタ A (PCRA) によって、1 ビットごとに入出力を指定できます。
	$P1_7 \sim P1_0$	17 ~ 24	入出力	<u>ポート 1</u> 8 ビットの入出力端子です。ポートコントロールレジスタ 1 (PCR1) によって、1 ビットごとに入出力を指定できます。
	$P2_7$	16	入力	<u>ポート 2 (ビット 7)</u> 1 ビットの入力端子です。

表 1.2 端子機能 (3)

分類	記号	ピン番号	入出力	機能
		FP-64A FP-64E		
I/O ポート	P2 <sub>6</sub> ~ P2 <sub>0</sub>	15 ~ 9	入出力	<u>ポート 2 (ビット 6 ~ ビット 0)</u> 7 ビットの入出力端子です。ポートコントロールレジスタ 2 (PCR2) によって、1 ビットごとに入出力を指定できます。
	P5 <sub>7</sub> ~ P5 <sub>0</sub>	35 ~ 28	入出力	<u>ポート 5</u> 8 ビットの入出力端子です。ポートコントロールレジスタ 5 (PCR5) によって、1 ビットごとに入出力を指定できます。
	P6 <sub>7</sub> ~ P6 <sub>0</sub>	43 ~ 36	入出力	<u>ポート 6</u> 8 ビットの入出力端子です。ポートコントロールレジスタ 6 (PCR6) によって、1 ビットごとに入出力を指定できます。
	P7 <sub>7</sub> ~ P7 <sub>0</sub>	51 ~ 44	入出力	<u>ポート 7</u> 8 ビットの入出力端子です。ポートコントロールレジスタ 7 (PCR7) によって、1 ビットごとに入出力を指定できます。
	P8 <sub>7</sub> ~ P8 <sub>0</sub>	59 ~ 52	入出力	<u>ポート 8</u> 8 ビットの入出力端子です。ポートコントロールレジスタ 8 (PCR8) によって、1 ビットごとに入出力を指定できます。
シリアル コミュニ ケーショ ンインタ フェース (SCI)	SI <sub>1</sub>	11	入力	<u>SCI1 受信データ入力</u> SCI1 のデータ入力端子です。
	SO <sub>1</sub>	12	出力	<u>SCI1 送信データ出力</u> SCI1 のデータ出力端子です。
	SCK <sub>1</sub>	10	入出力	<u>SCI1 クロック入出力</u> SCI1 のクロック入出力端子です。

表 1.2 端子機能 (4)

分類	記号	ピン番号	入出力	機能
		FP-64A FP-64E		
シリアル コミュニ ケーショ ンインタ フェース (SCI)	RXD	14	入力	<u>SCI3 受信データ入力</u> SCI3 のデータ入力端子です。
	TXD	15	出力	<u>SCI3 送信データ出力</u> SCI3 のデータ出力端子です。
	SCK <sub>3</sub>	13	入出力	<u>SCI3 クロック入出力</u> SCI3 のクロック入出力端子です。
A/D 変換器	AN <sub>7</sub> 、 AN <sub>6</sub>	62、63	入力	<u>アナログ入力 (チャンネル 7、6)</u> A/D 変換器へのアナログデータ入力端子です。
	$\overline{\text{ADTRG}}$	9	入力	<u>A/D 変換器トリガ入力</u> A/D 変換器の外部トリガ入力端子です。
DTMF 発生回路	TONED	60	出力	<u>DTMF 信号</u> DTMF 信号の出力端子です。

---

# 2. CPU

---

## 第2章 目次

2.1	概要.....	15
	2.1.1 特長.....	15
	2.1.2 アドレス空間.....	16
	2.1.3 レジスタ構成.....	16
2.2	各レジスタの説明.....	17
	2.2.1 汎用レジスタ.....	17
	2.2.2 コントロールレジスタ.....	17
	2.2.3 CPU内部レジスタの初期値.....	19
2.3	データ構成.....	20
	2.3.1 汎用レジスタのデータ構成.....	21
	2.3.2 メモリ上でのデータ構成.....	22
2.4	アドレッシングモード.....	23
	2.4.1 アドレッシングモード.....	23
	2.4.2 実効アドレスの計算方法.....	25
2.5	命令セット.....	29
	2.5.1 データ転送命令.....	31
	2.5.2 算術演算命令.....	33
	2.5.3 論理演算命令.....	34
	2.5.4 シフト命令.....	34
	2.5.5 ビット操作命令.....	36
	2.5.6 分岐命令.....	40
	2.5.7 システム制御命令.....	42
	2.5.8 ブロック転送命令.....	43
2.6	基本動作タイミング.....	44
	2.6.1 内蔵メモリ (RAM、ROM).....	44
	2.6.2 内蔵周辺モジュール.....	45
2.7	CPUの状態.....	47

## 2. CPU

---

2.7.1	概要.....	47
2.7.2	プログラム実行状態.....	48
2.7.3	プログラム停止状態.....	48
2.7.4	例外処理状態.....	48
2.8	メモリマップ.....	49
2.9	使用上の注意事項.....	50
2.9.1	データアクセスに関する注意事項.....	50
2.9.2	ビット操作命令使用上の注意事項.....	52
2.9.3	EEPMOV 命令使用上の注意事項.....	58

## 2.1 概要

H8/300L CPU は、8 ビット×16 本（または 16 ビット×8 本）の汎用レジスタ、ならびに高速動作に適した簡潔な命令セットを備えた高速 CPU です。

### 2.1.1 特長

H8/300L CPU には、次の特長があります。

汎用レジスタ方式

- ・ 8 ビット×16 本（16 ビット×8 本としても使用可能）

55 種類の基本命令

- ・ 乗除算命令
- ・ 強力なビット操作命令

8 種類のアドレッシングモード

- ・ レジスタ直接
- ・ レジスタ間接
- ・ ディスプレースメント付レジスタ間接
- ・ ポストインクリメント/プリデクリメントレジスタ間接
- ・ 絶対アドレス
- ・ イミディエイト
- ・ プログラムカウンタ相対
- ・ メモリ間接

64k バイトのアドレス空間

高速動作

- ・ 頻出命令をすべて 2~4 ステートで実行
- ・ 高速演算

8 / 16 ビットレジスタ間加減算 0.4 $\mu$ s\*

8 × 8 ビット乗算 2.8 $\mu$ s\*

16 ÷ 8 ビット除算 2.8 $\mu$ s\*

【注】 \* 数値は、 = 5MHz 時のもの

低消費電力動作

- ・ SLEEP 命令により低消費電力動作可能

## 2.1.2 アドレス空間

H8/300L CPU がサポートするアドレス空間は、プログラムコードとデータ領域合計で最大 64k バイトです。

メモリマップの詳細は「2.8 メモリマップ」を参照してください。

## 2.1.3 レジスタ構成

H8/300L CPU の内部レジスタ構成を図 2.1 に示します。これらのレジスタは、汎用レジスタとコントロールレジスタの 2 つに分類することができます。

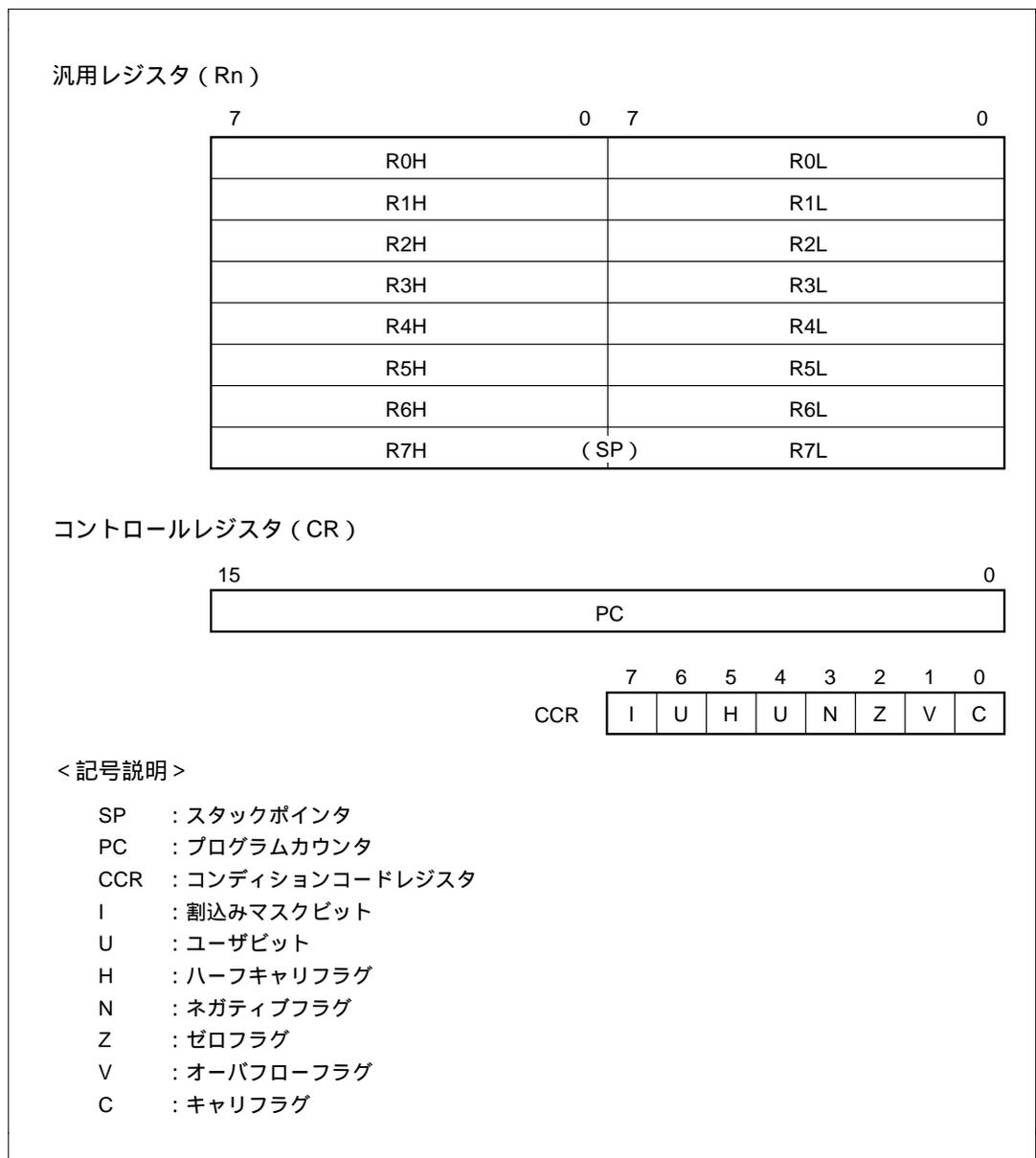


図 2.1 CPU 内部レジスタ構成

## 2.2 各レジスタの説明

### 2.2.1 汎用レジスタ

汎用レジスタは、すべて同じ機能をもっており、データレジスタ、アドレスレジスタの区別なく使用できます。

データレジスタとして使用する場合は、8ビットレジスタとして上位 (R7H~R0H) と下位 (R7L~R0L) を別々に使用することも、また 16ビットレジスタ (R7~R0) として使用することもできます。

アドレスレジスタとして使用する場合は、16ビットレジスタ (R7~R0) として使用します。

レジスタ R7 には、汎用レジスタとしての機能に加えて、スタックポインタ (SP) としての機能が割り当てられており、例外処理やサブルーチンコールなどで暗黙的に使用されます。このとき、SP は常にスタック領域の先頭を指しています。スタックの状態を図 2.2 に示します。

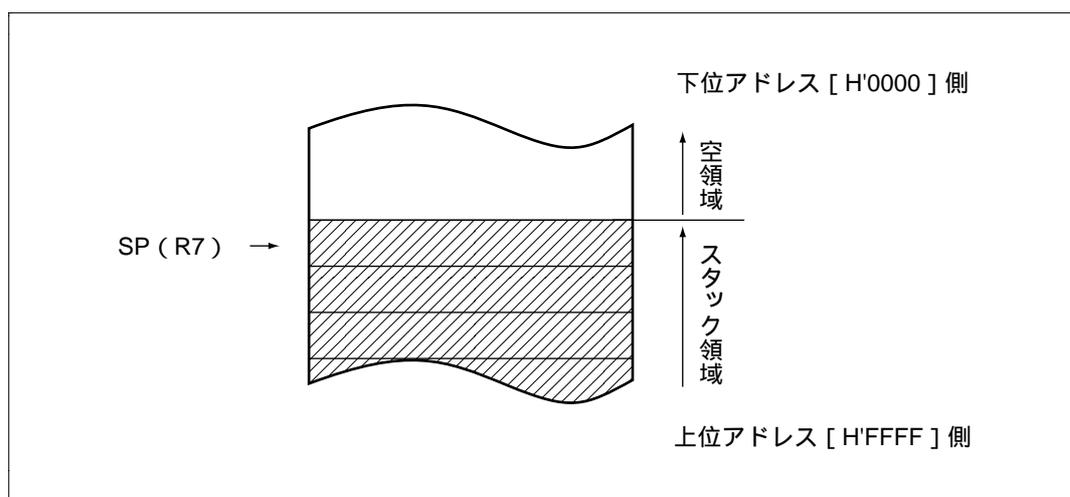


図 2.2 スタックの状態

### 2.2.2 コントロールレジスタ

コントロールレジスタには、16ビットのプログラムカウンタ (PC) と 8ビットのコンディションコードレジスタ (CCR) があります。

#### (1) プログラムカウンタ (PC)

16ビットのカウンタで、CPU が次に実行する命令のアドレスを示しています。CPU の命令は、すべて 16ビット (ワード) を単位としているため、最下位ビットは無効です (命令コードのリード時には最下位ビットは "0" とみなされます)。

## (2) コンディションコードレジスタ (CCR)

8ビットのレジスタで、CPUの内部状態を示しています。割込みマスクビット(I)とハーフキャリ(H)、ネガティブ(N)、ゼロ(Z)、オーバフロー(V)、キャリ(C)の各フラグを含む8ビットで構成されています。これらのビットは、ソフトウェア(LDC、STC、ANDC、ORC、XORC命令)でリード/ライトできます。N、Z、V、Cの各フラグは、条件分岐命令(Bcc)で使用されます。

### ビット7：割込みマスクビット(I)

本ビットが"1"にセットされると、割込みがマスクされます。例外処理の実行が開始されたときに"1"にセットされます。本ビットはソフトウェアによりリード/ライトできません。割込みマスクビットの詳細については「3.3 割込み」を参照してください。

### ビット6：ユーザビット(U)

ユーザが自由に使用できるビットです。

### ビット5：ハーフキャリフラグ(H)

ADD.B、ADDX.B、SUB.B、SUBX.B、CMP.B、NEG.B命令の実行により、ビット3にキャリまたはボローが生じたとき"1"にセットされ、生じなかったとき"0"にクリアされます。DAAおよびDAS命令実行時に、暗黙的に使用されます。ADD.W、SUB.W、CMP.W命令ではビット11にキャリまたはボローが生じたとき"1"にセットされ、生じなかったとき"0"にクリアされます。

### ビット4：ユーザビット(U)

ユーザが自由に使用できるビットです。

### ビット3：ネガティブフラグ(N)

データの最上位ビットを符号ビットとみなし、最上位ビットの値を格納します。

### ビット2：ゼロフラグ(Z)

データがゼロのとき"1"にセットされ、ゼロ以外のとき"0"にクリアされます。

### ビット1：オーバフローフラグ(V)

算術演算命令の実行により、オーバフローが生じたとき"1"にセットされます。それ以外のとき"0"にクリアされます。

#### ビット0：キャリフラグ (C)

演算の実行により、キャリが生じたとき"1"にセットされ、生じなかったとき"0"にクリアされます。キャリには次の種類があります。

- ・加算結果のキャリ
- ・減算結果のボロー
- ・シフト/ローテートのキャリ

また、キャリフラグには、ビットアキュムレータ機能があり、ビット操作命令で使用されます。

なお、命令によってはフラグが変化しない場合があります。

各命令ごとのフラグの変化については、「H8/300L シリーズ プログラミングマニュアル」を参照してください。

### 2.2.3 CPU 内部レジスタの初期値

リセット例外処理によって、CPU 内部レジスタのうち、PC はベクタアドレス (H'0000) のロードにより初期化され、CCR の I ビットは"1"にセットされますが、汎用レジスタおよび CCR の他のビットは初期化されません。レジスタ R7 (SP) の初期値も不定です。したがって、リセット直後に、R7 の初期化を行ってください。

## 2.3 データ構成

H8/300L CPU は、1ビット、4ビットBCD、8ビット(バイト)、16ビット(ワード)のデータを扱うことができます。1ビットデータはビット操作命令で扱われ、オペランドデータ(バイト)の第nビット(n=0、1、2、.....7)という形式でアクセスされます。

バイトデータは、ADDS、SUBS 以外の演算命令で扱われます。また、ワードデータは、MOV.W、ADD.W、SUB.W、CMP.W、ADDS、SUBS、MULXU(8ビット×8ビット)、DIVXU(16ビット÷8ビット)命令で扱われます。

なお、DAA および DAS の10進補正命令では、バイトデータは2桁の4ビットBCDデータとなります。

### 2.3.1 汎用レジスタのデータ構成

汎用レジスタのデータ構成を図 2.3 に示します。

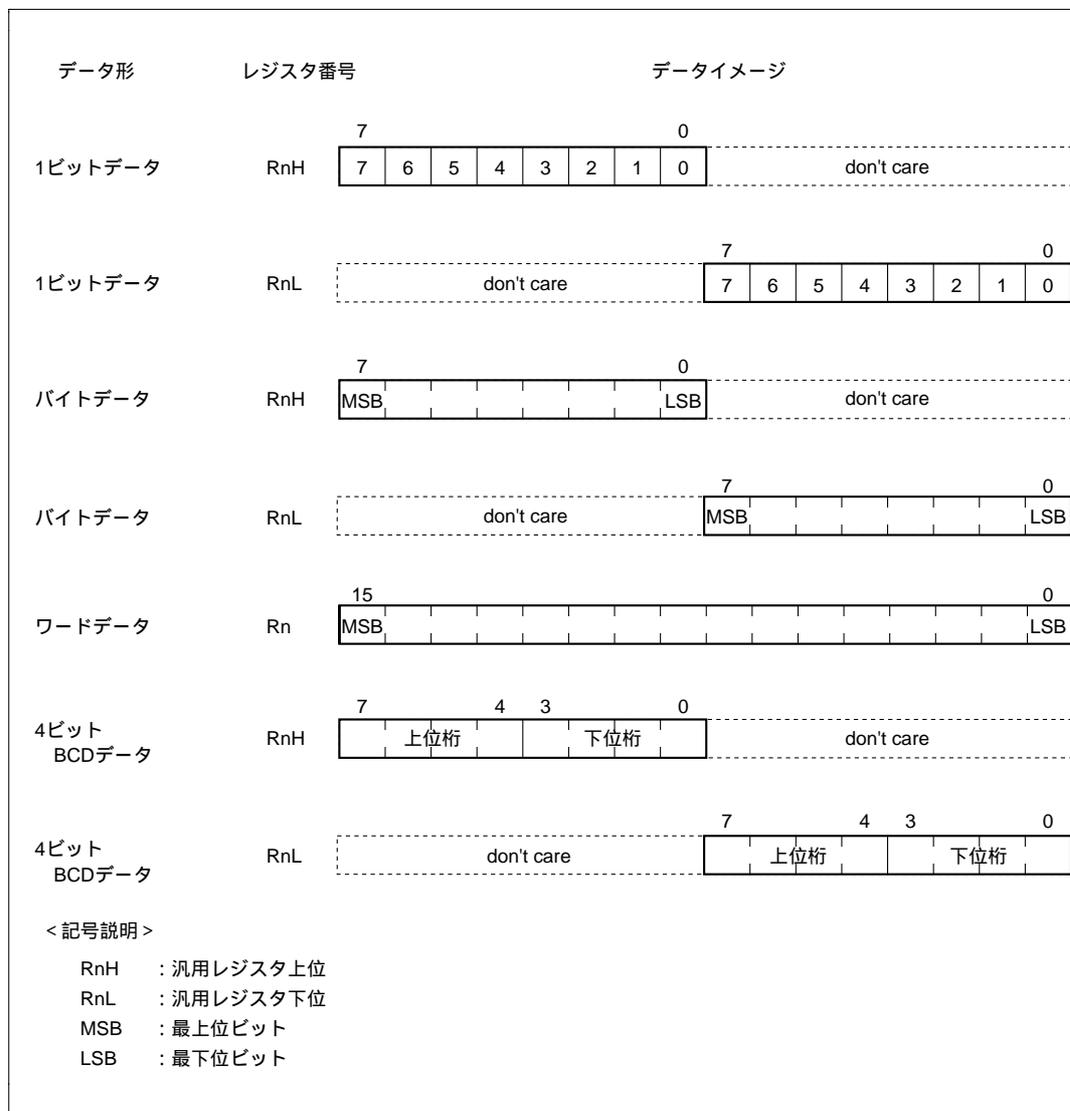


図 2.3 汎用レジスタのデータ構成

### 2.3.2 メモリ上でのデータ構成

メモリ上でのデータ構成を図2.4に示します。H8/300L CPUは、メモリ上のワードデータをアクセスすることができます（MOV.W命令）が、偶数番地から始まるワードデータに限定されます。奇数番地から始まるワードデータをアクセスした場合、アドレスの最下位ビットは"0"とみなされ、1番地前から始まるワードデータをアクセスします。命令コードについても同様です。

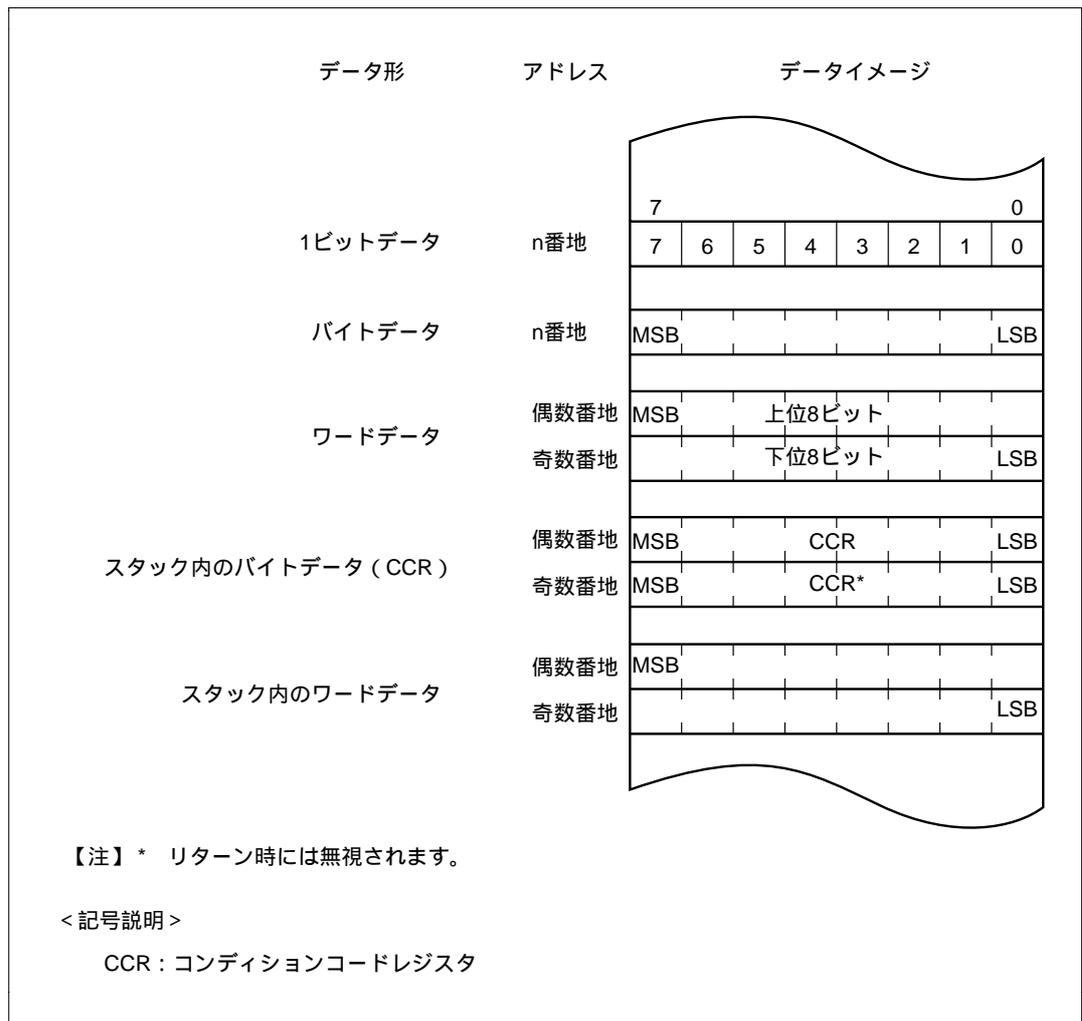


図2.4 メモリ上でのデータ構成

なお、R7 をアドレスレジスタとして使用し、スタックをアクセスするときは、必ずワードサイズでアクセスしてください。また、CCR は、ワードデータとして上位8ビット、下位8ビットに同じ値が格納され、リターン時には、下位8ビットは無視されます。

## 2.4 アドレッシングモード

### 2.4.1 アドレッシングモード

H8/300L CPU は、表 2.1 に示すように、8 種類のアドレッシングモードをサポートしています。命令ごとに、使用できるアドレッシングモードは異なります。

表 2.1 アドレッシングモード一覧表

No.	アドレッシングモード	記号
(1)	レジスタ直接	Rn
(2)	レジスタ間接	@Rn
(3)	ディスプレースメント付レジスタ間接	@(d:16, Rn)
(4)	ポストインクリメントレジスタ間接 プリデクリメントレジスタ間接	@Rn+ @-Rn
(5)	絶対アドレス	@aa:8 / @aa:16
(6)	イミディエイト	#xx:8 / #xx:16
(7)	プログラムカウンタ相対	@(d:8, PC)
(8)	メモリ間接	@@aa:8

#### (1) レジスタ直接 Rn

命令コードのレジスタフィールドで指定されるレジスタ (8 ビットまたは 16 ビット) がオペランドとなります。

16 ビットレジスタを使用する命令は、MOV.W、ADD.W、SUB.W、CMP.W、ADDS、SUBS、MULXU (8 ビット×8 ビット)、DIVXU (16 ビット÷8 ビット) の各命令です。

#### (2) レジスタ間接 @Rn

命令コードのレジスタフィールドで指定されるレジスタ (16 ビット) の内容をアドレスとしてメモリ上のオペランドを指定します。

#### (3) ディスプレースメント付レジスタ間接 @(d:16, Rn)

命令コードのレジスタフィールドで指定されるレジスタ (16 ビット) の内容に、命令コードの第 2 ワード (第 3、第 4 バイト) の 16 ビットディスプレースメントを加算した内容をアドレスとして、メモリ上のオペランドを指定します。

本アドレッシングモードは、MOV 命令のみで使用されます。特に、MOV.W 命令では、加算結果が偶数となるようにしてください。

(4) ポストインクリメントレジスタ間接 @Rn+ / プリデクリメントレジスタ間接 @-Rn

・ ポストインクリメントレジスタ間接 @Rn+

MOV (Load from) 命令で使用されます。

命令コードのレジスタフィールドで指定されるレジスタ (16 ビット) の内容をアドレスとして、メモリ上のオペランドを指定します。その後、レジスタの内容に 1 または 2 が加算され、加算結果がレジスタに格納されます。MOV.B 命令では 1、MOV.W 命令では 2 がそれぞれ加算されます。MOV.W 命令では、レジスタの内容が偶数になるようにしてください。

・ プリデクリメントレジスタ間接 @-Rn

MOV (Store to) 命令で使用されます。

命令コードのレジスタフィールドで指定されるレジスタ (16 ビット) の内容から 1 または 2 を減算した内容をアドレスとして、メモリ上のオペランドを指定します。その後、減算結果がレジスタに格納されます。MOV.B 命令では 1、MOV.W 命令では 2 がそれぞれ減算されます。MOV.W 命令では、レジスタの内容が偶数となるようにしてください。

(5) 絶対アドレス @aa:8 / @aa:16

命令コード中に含まれる絶対アドレスで、メモリ上のオペランドを指定します。

このとき、絶対アドレスは 8 ビット (@aa:8) または 16 ビット (@aa:16) で、8 ビット絶対アドレスは MOV.B、ビット操作命令で、16 ビット絶対アドレスは MOV.B、MOV.W、JMP、JSR の各命令で使用されます。

8 ビット絶対アドレスの場合、上位 8 ビットはすべて "1" (H'FF) となります。したがって、アクセス範囲は 65280 ~ 65535 (H'FF00 ~ H'FFFF) 番地です。

(6) イミディエイト #xx:8 / #xx:16

命令コードの第 2 バイト (#xx:8) または第 3、第 4 バイト (#xx:16) を直接オペランドとして使用します。#xx:16 は、MOV.W 命令のみで使用されます。

なお、ADDS および SUBS 命令では、イミディエイトデータ (1 または 2) が命令コード中に暗黙的に含まれます。ビット操作命令では、ビット番号を指定するための 3 ビットのイミディエイトデータが、命令コードの第 2 または第 4 バイトに含まれる場合があります。

#### (7) プログラムカウンタ相対 @ (d:8, PC)

Bcc、BSR の各命令で使用されます。

PC の内容に、命令コードの第 2 バイトの 8 ビットディスプレースメントを加算して、分岐アドレスを生成します。加算に際して、ディスプレースメントは 16 ビットに符号拡張され、また加算される PC の内容は次の命令の先頭アドレスとなっていますので、分岐可能範囲は分岐命令に対して -126 ~ +128 バイト (-63 ~ +64 ワード) です。このとき、加算結果が偶数となるようにしてください。

#### (8) メモリ間接 @@aa:8

JMP および JSR 命令で使用されます。

命令コードの第 2 バイトに含まれる 8 ビット絶対アドレスでメモリ上のオペランドを指定し、この内容を分岐アドレスとして分岐します。この場合、8 ビット絶対アドレスの上位 8 ビットはすべて "0" (H'00) とされますので、分岐アドレスを格納できるのは 0 ~ 255 (H'0000 ~ H'00FF) 番地です。ただし、H8/300L シリーズでは、アドレスの下位番地はベクタ領域と共通になっていますから注意してください。ベクタ領域の詳細は「3.3 割込み」を参照してください。

分岐アドレスまたは MOV.W 命令のオペランドアドレスとして奇数アドレスを指定した場合、最下位ビットは "0" とみなされ、1 番地前から始まるワードデータをアクセスします (「2.3.2 メモリ上でのデータ構成」を参照してください)。

### 2.4.2 実効アドレスの計算方法

各アドレッシングモードにおける実効アドレス (EA: Effective Address) の計算方法を表 2.2 に示します。

演算命令では、(1) レジスタ直接、および (6) イミディエイト (ADD.B、ADDX、SUBX、CMP.B、AND、OR、XOR の各命令) が使用されます。

転送命令では、(7) プログラムカウンタ相対と (8) メモリ間接を除くすべてのアドレッシングモードが使用可能です。

また、ビット操作命令では、オペランドの指定に (1) レジスタ直接、(2) レジスタ間接および (5) 絶対アドレス (8 ビット) が使用可能です。さらに、オペランド中のビット番号を指定するために (1) レジスタ直接 (BSET、BCLR、BNOT、BTST の各命令) および (6) イミディエイト (3 ビット) が独立して使用可能です。

表 2.2 実効アドレスの計算方法 (1)

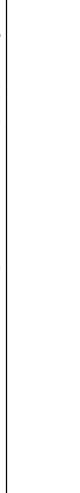
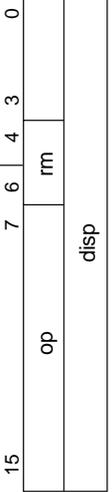
No.	アドレッシングモード・命令フォーマット	実効アドレス計算方法	実効アドレス (EA)
(1)	レジスタ直接 Rn 	実効アドレスはrm/rnが示すレジスタの内容です。	
(2)	レジスタ間接 @Rn 	rmが示すレジスタの内容 (16ビット)	
(3)	ディスプレースメント付レジスタ間接 @(d:16, Rn) 	rmが示すレジスタの内容 (16ビット) ⊕ disp	
(4)	ポストインクリメントレジスタ間接 / プリデクリメントレジスタ間接 ・ポストインクリメントレジスタ間接 @Rn+  ・プリデクリメントレジスタ間接 @-Rn 	rmが示すレジスタの内容 (16ビット) ⊕ 1 or 2	 オペランドサイズがバイトのとき1、ワードのとき2が加減算されます。

表 2.2 実効アドレスの計算方法 (2)

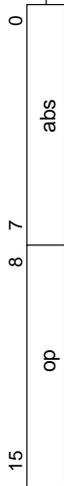
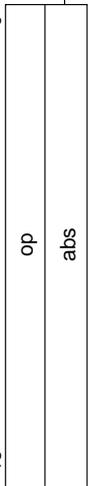
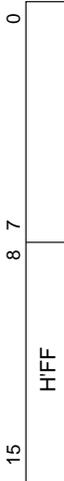
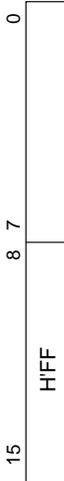
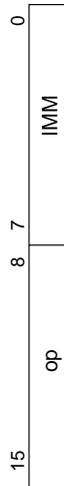
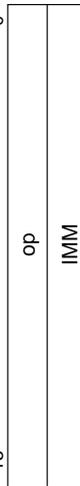
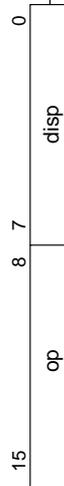
No.	アドレッシングモード・命令フォーマット	実効アドレス計算方法	実効アドレス (EA)
(5)	絶対アドレス @aa:8  @aa:16 	 	 
(6)	イミディエイト #xx:8  #xx:16 		オペランドはイミディエイトデータの1または2バイトデータです。
(7)	プログラムカウンタ相対 @ (d:8, PC) 		

表 2.2 実効アドレスの計算方法 (3)

No.	アドレッシングモード・命令フォーマット	実効アドレス計算方法	実効アドレス (EA)
(8)	メモリ間接 @aa:8		

<記号説明>

- rm, m : レジスタフィールド
- op : オペレーションフィールド
- disp : ディスプレースメント
- IMM : イミディエイトデータ
- abs : 絶対アドレス

## 2.5 命令セット

H8/300L CPUの命令は合計 55 種類あり、各命令のもつ機能によって、表 2.3 に示すように分類されます。

表 2.3 命令の分類

機能	命令	種類
データ転送命令	MOV、POP* <sup>1</sup> 、PUSH* <sup>1</sup>	1
算術演算命令	ADD、SUB、ADDX、SUBX、INC、DEC、ADDS、SUBS、DAA、DAS、MULXU、DIVXU、CMP、NEG	14
論理演算命令	AND、OR、XOR、NOT	4
シフト命令	SHAL、SHAR、SHLL、SHLR、ROTL、ROTR、ROTXL、ROTXR	8
ビット操作命令	BSET、BCLR、BNOT、BTST、BAND、BIAND、BOR、BIOR、BXOR、BIXOR、BLD、BILD、BST、BIST	14
分岐命令	Bcc* <sup>2</sup> 、JMP、BSR、JSR、RTS	5
システム制御命令	RTE、SLEEP、LDC、STC、ANDC、ORC、XORC、NOP	8
ブロック転送命令	EEPMOV	1

合計 55 種

【注】 \*1 POP Rn、PUSH Rn は、それぞれ MOV.W @SP+, Rn、MOV.W Rn, @-SP と同一です。  
機械語についても同一です。

\*2 Bcc は条件分岐命令の総称です。

各命令の機能について表 2.4 から表 2.11 に示します。各表で使用しているオペレーションの記号の意味は以下のとおりです。

## 《オペレーションの記号》

Rd	汎用レジスタ (デスティネーション側)
Rs	汎用レジスタ (ソース側)
Rn	汎用レジスタ
(EAd)、<EAd>	デスティネーションオペランド
(EAs)、<EAs>	ソースオペランド
CCR	コンディションコードレジスタ
N	CCRのN(ネガティブ)フラグ
Z	CCRのZ(ゼロ)フラグ
V	CCRのV(オーバフロー)フラグ
C	CCRのC(キャリ)フラグ
PC	プログラムカウンタ
SP	スタックポインタ
#IMM	イミディエイトデータ
disp	ディスプレースメント
+	加算
-	減算
×	乗算
÷	除算
	論理積
	論理和
⊕	排他的論理和
	転送
~	反転論理(論理的補数)
:3	3ビット長
:8	8ビット長
:16	16ビット長
( )、< >	オペランドの実効アドレスの内容

## 2.5.1 データ転送命令

データ転送命令の機能を表 2.4 に示します。

表 2.4 データ転送命令

命令	サイズ*	機能
MOV	B / W	(EAs) Rd, Rs (EAd) 汎用レジスタと汎用レジスタまたは汎用レジスタとメモリ間でデータ転送します。また、イミディエイトデータを汎用レジスタに転送します。 ワードデータは Rn、@Rn、@(d:16, Rn)、@aa:16、#xx:16、@-Rn、@Rn+ の各アドレッシングモードで扱います。@aa:8 はバイトデータのみです。 ただし、@-R7、@R7+を使用する場合は必ずワードサイズを指定してください。
POP	W	@SP+ Rn スタックから汎用レジスタへデータを復帰します。 本命令は MOV.W @SP+, Rn と同一です。
PUSH	W	Rn @-SP 汎用レジスタの内容をスタックに退避します。 本命令は MOV.W Rn, @-SP と同一です。

【注】 \* サイズはオペランドサイズを示します。

B: バイト

W: ワード

データアクセスに関して使用上の注意事項があります。詳細は「2.9.1 データアクセスに関する注意事項」を参照してください。

データ転送命令の命令フォーマットを図 2.5 に示します。

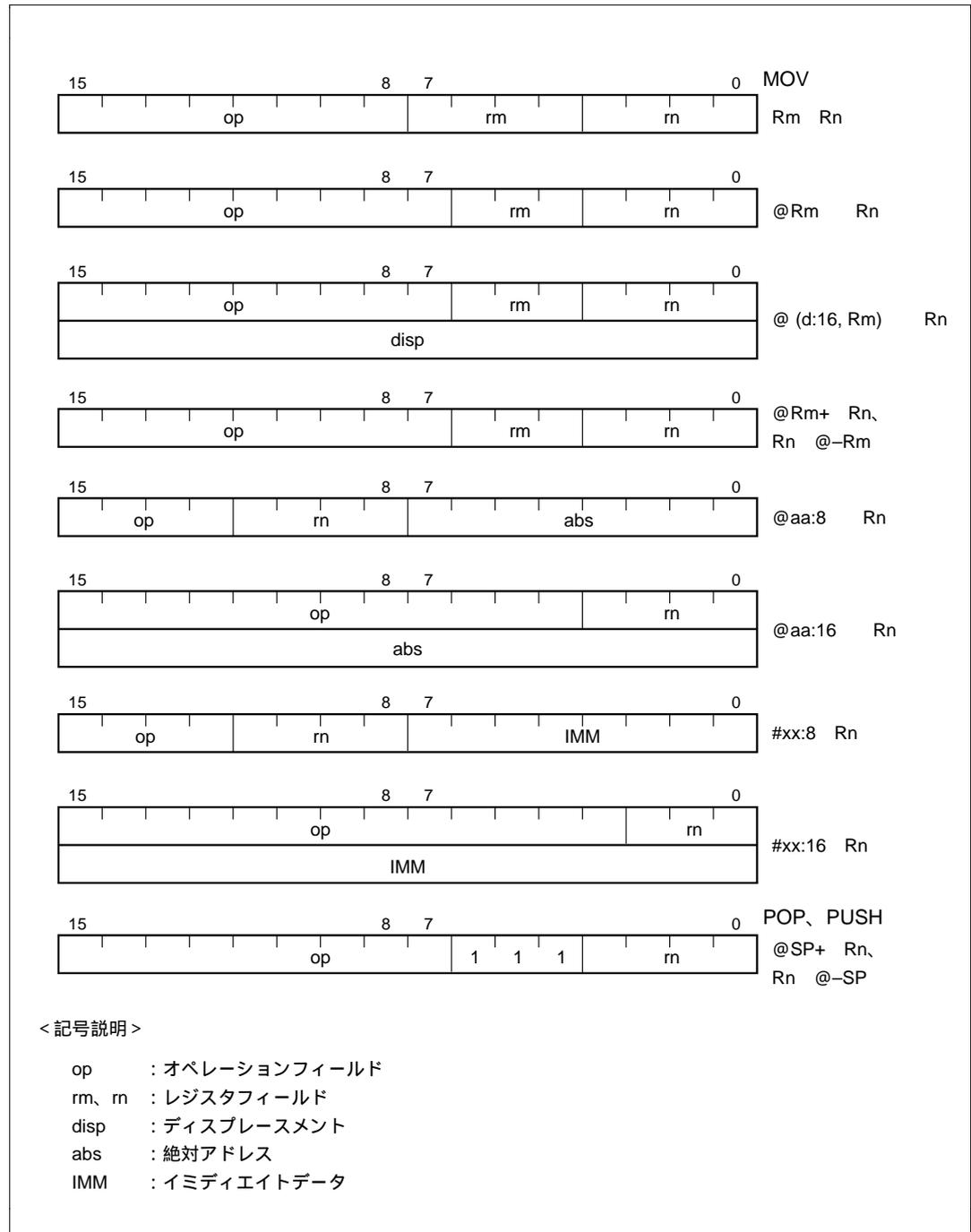


図 2.5 データ転送命令の命令フォーマット

## 2.5.2 算術演算命令

算術演算命令の機能を表 2.5 に示します。

表 2.5 算術演算命令

命令	サイズ*	機能
ADD SUB	B / W	$Rd \pm Rs$ $Rd, Rd + \#IMM$ $Rd$ 汎用レジスタ間の加減算、または汎用レジスタとイミディエイトデータの加算を行います。汎用レジスタとイミディエイトデータの減算はできません。ワードデータは、汎用レジスタ間の加減算のみで扱います。
ADDX SUBX	B	$Rd \pm Rs \pm C$ $Rd, Rd \pm \#IMM \pm C$ $Rd$ 汎用レジスタ間のキャリ付の加減算、または汎用レジスタとイミディエイトデータのキャリ付の加減算を行います。
INC DEC	B	$Rd \pm 1$ $Rd$ 汎用レジスタに 1 を加減算します。
ADDS SUBS	W	$Rd \pm 1$ $Rd, Rd \pm 2$ $Rd$ 汎用レジスタに 1 または 2 を加減算します。
DAA DAS	B	$Rd$ (10 進補正) $Rd$ 汎用レジスタ上の加減算結果を CCR を参照して 4 ビット BCD データに補正します。
MULXU	B	$Rd \times Rs$ $Rd$ 汎用レジスタ間の符号なし乗算を行います。8 ビット $\times$ 8 ビット 16 ビットの演算が可能です。
DIVXU	B	$Rd \div Rs$ $Rd$ 汎用レジスタ間の符号なし除算を行います。16 ビット $\div$ 8 ビット 商 8 ビット 余り 8 ビットの演算が可能です。
CMP	B / W	$Rd - Rs, Rd - \#IMM$ 汎用レジスタ間の比較、または汎用レジスタとイミディエイトデータの比較を行い、その結果を CCR に反映します。ワードデータは、汎用レジスタ間の比較のみで扱います。
NEG	B	$0 - Rd$ $Rd$ 汎用レジスタの内容の 2 の補数 (算術的補数) をとります。

【注】 \* サイズはオペランドサイズを示します。

B : バイト

W : ワード

### 2.5.3 論理演算命令

論理演算命令の機能を表 2.6 に示します。

表 2.6 論理演算命令

命令	サイズ*	機能
AND	B	Rd Rs Rd, Rd #IMM Rd 汎用レジスタ間の論理積、または汎用レジスタとイミディエイトデータの論理積をとります。
OR	B	Rd Rs Rd, Rd #IMM Rd 汎用レジスタ間の論理和、または汎用レジスタとイミディエイトデータの論理和をとります。
XOR	B	Rd⊕Rs Rd, Rd⊕#IMM Rd 汎用レジスタ間の排他的論理和、または汎用レジスタとイミディエイトデータの排他的論理和をとります。
NOT	B	~Rd Rd 汎用レジスタの内容の 1 の補数（論理的補数）をとります。

【注】 \* サイズはオペランドサイズを示します。

B: バイト

### 2.5.4 シフト命令

シフト命令の機能を表 2.7 に示します。

表 2.7 シフト命令

命令	サイズ*	機能
SHAL SHAR	B	Rd (シフト処理) Rd 汎用レジスタの内容を算術的にシフトします。
SHLL SHLR	B	Rd (シフト処理) Rd 汎用レジスタの内容を論理的にシフトします。
ROTL ROTR	B	Rd (ローテート処理) Rd 汎用レジスタの内容をローテートします。
ROTXL ROTXR	B	Rd (ローテート処理) Rd 汎用レジスタの内容を、キャリフラグを含めてローテートします。

【注】 \* サイズはオペランドサイズを示します。

B: バイト

算術演算命令、論理演算命令およびシフト命令の命令フォーマットを図 2.6 に示します。

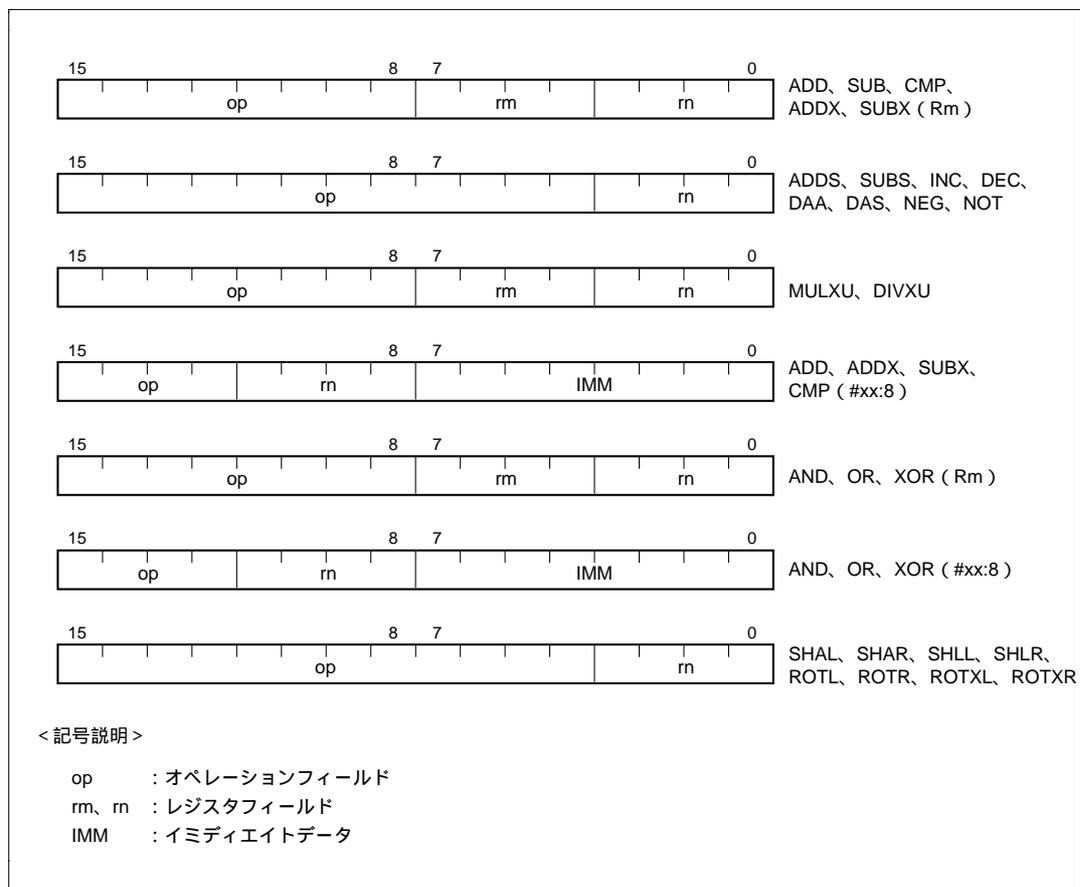


図 2.6 算術演算命令・論理演算命令・シフト命令の命令フォーマット

## 2.5.5 ビット操作命令

ビット操作命令の機能を表 2.8 に示します。

表 2.8 ビット操作命令 (1)

命令	サイズ*	機能
BSET	B	1 ( <ビット番号> of <EAd> ) 汎用レジスタまたはメモリのオペランドの指定された 1 ビットを"1"にセットします。ビット番号は、3 ビットのイミディエイトデータまたは汎用レジスタの内容下位 3 ビットで指定されます。
BCLR	B	0 ( <ビット番号> of <EAd> ) 汎用レジスタまたはメモリのオペランドの指定された 1 ビットを"0"にクリアします。ビット番号は、3 ビットのイミディエイトデータまたは汎用レジスタの内容下位 3 ビットで指定されます。
BNOT	B	~ ( <ビット番号> of <EAd> ) ( <ビット番号> of <EAd> ) 汎用レジスタまたはメモリのオペランドの指定された 1 ビットを反転します。ビット番号は、3 ビットのイミディエイトデータまたは汎用レジスタの内容下位 3 ビットで指定されます。
BTST	B	~ ( <ビット番号> of <EAd> ) Z 汎用レジスタまたはメモリのオペランドの指定された 1 ビットをテストし、ゼロフラグに反映します。ビット番号は、3 ビットのイミディエイトデータまたは汎用レジスタの内容下位 3 ビットで指定されます。
BAND	B	C ( <ビット番号> of <EAd> ) C 汎用レジスタまたはメモリのオペランドの指定された 1 ビットとキャリフラグとの論理積をとり、キャリフラグに結果を格納します。
BIAND	B	C [ ~ ( <ビット番号> of <EAd> ) ] C 汎用レジスタまたはメモリのオペランドの指定された 1 ビットを反転し、キャリフラグとの論理積をとり、キャリフラグに結果を格納します。  ビット番号は、3 ビットのイミディエイトデータで指定されます。

表 2.8 ビット操作命令 (2)

命令	サイズ*	機能
BOR	B	C (<ビット番号> of <EAd>) C 汎用レジスタまたはメモリのオペランドの指定された1ビットとキャリフラグとの論理和をとり、キャリフラグに結果を格納します。
BIOR	B	C [ ~ (<ビット番号> of <EAd> ) ] C 汎用レジスタまたはメモリのオペランドの指定された1ビットを反転し、キャリフラグとの論理和をとり、キャリフラグに結果を格納します。  ビット番号は、3ビットのイミディエイトデータで指定されます。
BXOR	B	C⊕ (<ビット番号> of <EAd>) C 汎用レジスタまたはメモリのオペランドの指定された1ビットとキャリフラグとの排他的論理和をとり、キャリフラグに結果を格納します。
BIXOR	B	C⊕ [ ~ (<ビット番号> of <EAd> ) ] C 汎用レジスタまたはメモリのオペランドの指定された1ビットを反転し、キャリフラグとの排他的論理和をとり、キャリフラグに結果を格納します。  ビット番号は、3ビットのイミディエイトデータで指定されます。
BLD	B	(<ビット番号> of <EAd>) C 汎用レジスタまたはメモリのオペランドの指定された1ビットをキャリフラグに転送します。
BILD	B	~ (<ビット番号> of <EAd>) C 汎用レジスタまたはメモリのオペランドの指定された1ビットを反転し、キャリフラグに転送します。  ビット番号は、3ビットのイミディエイトデータで指定されます。

表 2.8 ビット操作命令 (3)

命令	サイズ*	機能
BST	B	C ( <ビット番号> of <EAd> ) 汎用レジスタまたはメモリのオペランドの指定された1ビットに、キャリフラグの内容を転送します。
BIST	B	~C ( <ビット番号> of <EAd> ) 汎用レジスタまたはメモリのオペランドの指定された1ビットに、反転されたキャリフラグの内容を転送します。  ビット番号は、3ビットのイミディエイトデータで指定されます。

【注】 \* サイズはオペランドサイズを示します。

B : バイト

ビット操作命令には使用上の注意事項があります。詳細は「2.9.2 ビット操作命令使用上の注意事項」を参照してください。

ビット操作命令の命令フォーマットを図 2.7 に示します。

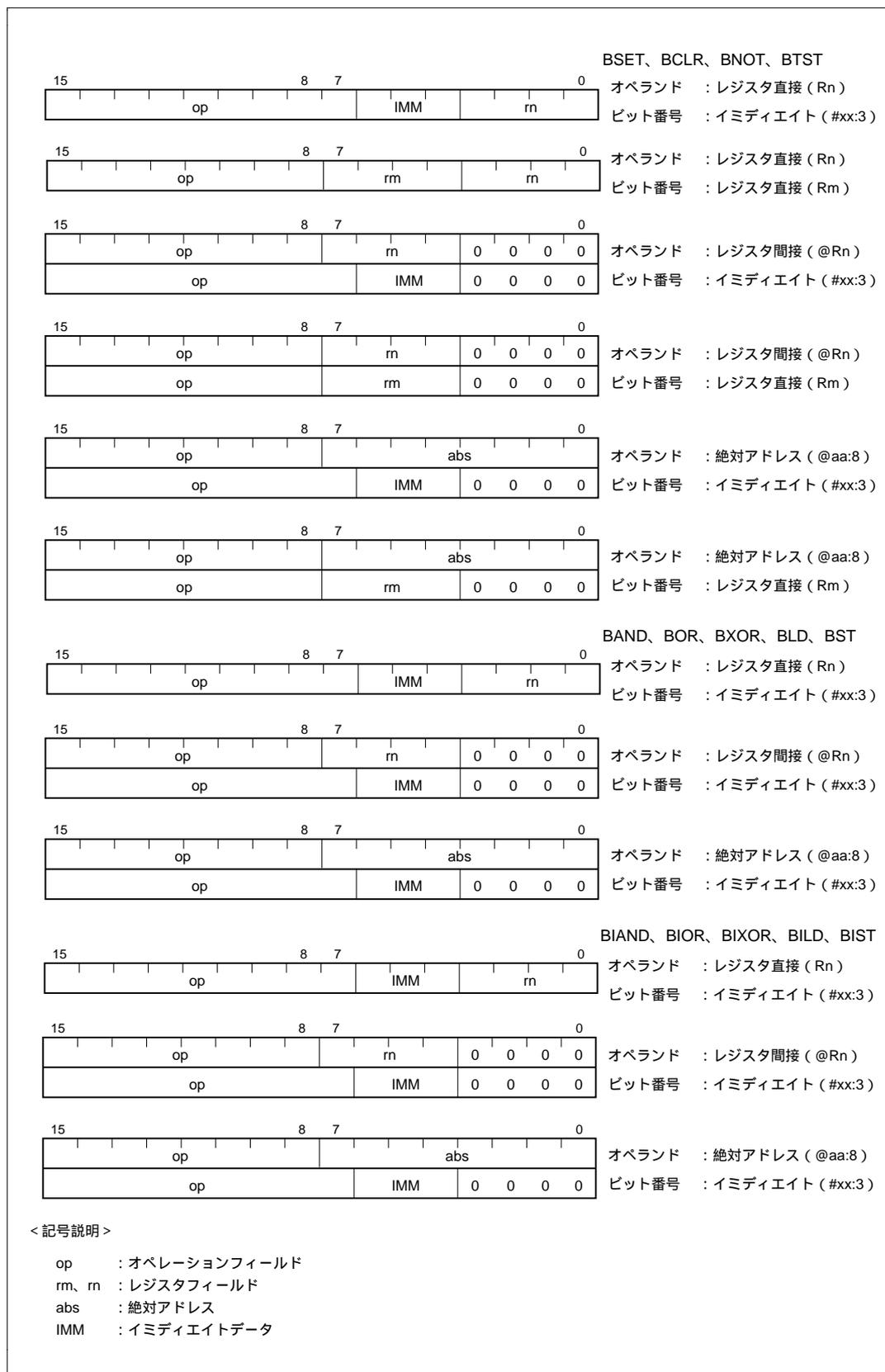


図 2.7 ビット操作命令の命令フォーマット

## 2.5.6 分岐命令

分岐命令の機能を表 2.9 に示します。

表 2.9 分岐命令

命令	サイズ	機能																																																			
Bcc		指定した条件が成立しているとき、指定されたアドレスへ分岐します。分岐条件を下表に示します。																																																			
		<table border="1"> <thead> <tr> <th>ニーモニック</th> <th>説明</th> <th>分岐条件</th> </tr> </thead> <tbody> <tr> <td>BRA ( BT )</td> <td>Always ( True )</td> <td>Always</td> </tr> <tr> <td>BRN ( BF )</td> <td>Never ( False )</td> <td>Never</td> </tr> <tr> <td>BHI</td> <td>High</td> <td>C Z = 0</td> </tr> <tr> <td>BLS</td> <td>Low or Same</td> <td>C Z = 1</td> </tr> <tr> <td>BCC ( BHS )</td> <td>Carry Clear ( High or Same )</td> <td>C = 0</td> </tr> <tr> <td>BCS ( BLO )</td> <td>Carry Set ( LOw )</td> <td>C = 1</td> </tr> <tr> <td>BNE</td> <td>Not Equal</td> <td>Z = 0</td> </tr> <tr> <td>BEQ</td> <td>Equal</td> <td>Z = 1</td> </tr> <tr> <td>BVC</td> <td>oVerflow Clear</td> <td>V = 0</td> </tr> <tr> <td>BVS</td> <td>oVerflow Set</td> <td>V = 1</td> </tr> <tr> <td>BPL</td> <td>PLus</td> <td>N = 0</td> </tr> <tr> <td>BMI</td> <td>MInus</td> <td>N = 1</td> </tr> <tr> <td>BGE</td> <td>Greater or Equal</td> <td><math>N \oplus V = 0</math></td> </tr> <tr> <td>BLT</td> <td>Less Than</td> <td><math>N \oplus V = 1</math></td> </tr> <tr> <td>BGT</td> <td>Greater Than</td> <td><math>Z ( N \oplus V ) = 0</math></td> </tr> <tr> <td>BLE</td> <td>Less or Equal</td> <td><math>Z ( N \oplus V ) = 1</math></td> </tr> </tbody> </table>	ニーモニック	説明	分岐条件	BRA ( BT )	Always ( True )	Always	BRN ( BF )	Never ( False )	Never	BHI	High	C Z = 0	BLS	Low or Same	C Z = 1	BCC ( BHS )	Carry Clear ( High or Same )	C = 0	BCS ( BLO )	Carry Set ( LOw )	C = 1	BNE	Not Equal	Z = 0	BEQ	Equal	Z = 1	BVC	oVerflow Clear	V = 0	BVS	oVerflow Set	V = 1	BPL	PLus	N = 0	BMI	MInus	N = 1	BGE	Greater or Equal	$N \oplus V = 0$	BLT	Less Than	$N \oplus V = 1$	BGT	Greater Than	$Z ( N \oplus V ) = 0$	BLE	Less or Equal	$Z ( N \oplus V ) = 1$
		ニーモニック	説明	分岐条件																																																	
		BRA ( BT )	Always ( True )	Always																																																	
		BRN ( BF )	Never ( False )	Never																																																	
		BHI	High	C Z = 0																																																	
		BLS	Low or Same	C Z = 1																																																	
		BCC ( BHS )	Carry Clear ( High or Same )	C = 0																																																	
		BCS ( BLO )	Carry Set ( LOw )	C = 1																																																	
		BNE	Not Equal	Z = 0																																																	
		BEQ	Equal	Z = 1																																																	
		BVC	oVerflow Clear	V = 0																																																	
		BVS	oVerflow Set	V = 1																																																	
		BPL	PLus	N = 0																																																	
		BMI	MInus	N = 1																																																	
		BGE	Greater or Equal	$N \oplus V = 0$																																																	
		BLT	Less Than	$N \oplus V = 1$																																																	
BGT	Greater Than	$Z ( N \oplus V ) = 0$																																																			
BLE	Less or Equal	$Z ( N \oplus V ) = 1$																																																			
JMP		指定されたアドレスへ無条件に分岐します。																																																			
BSR		指定されたアドレスへサブルーチン分岐します。																																																			
JSR		指定されたアドレスへサブルーチン分岐します。																																																			
RTS		サブルーチンから復帰します。																																																			

分岐命令の命令フォーマットを図 2.8 に示します。

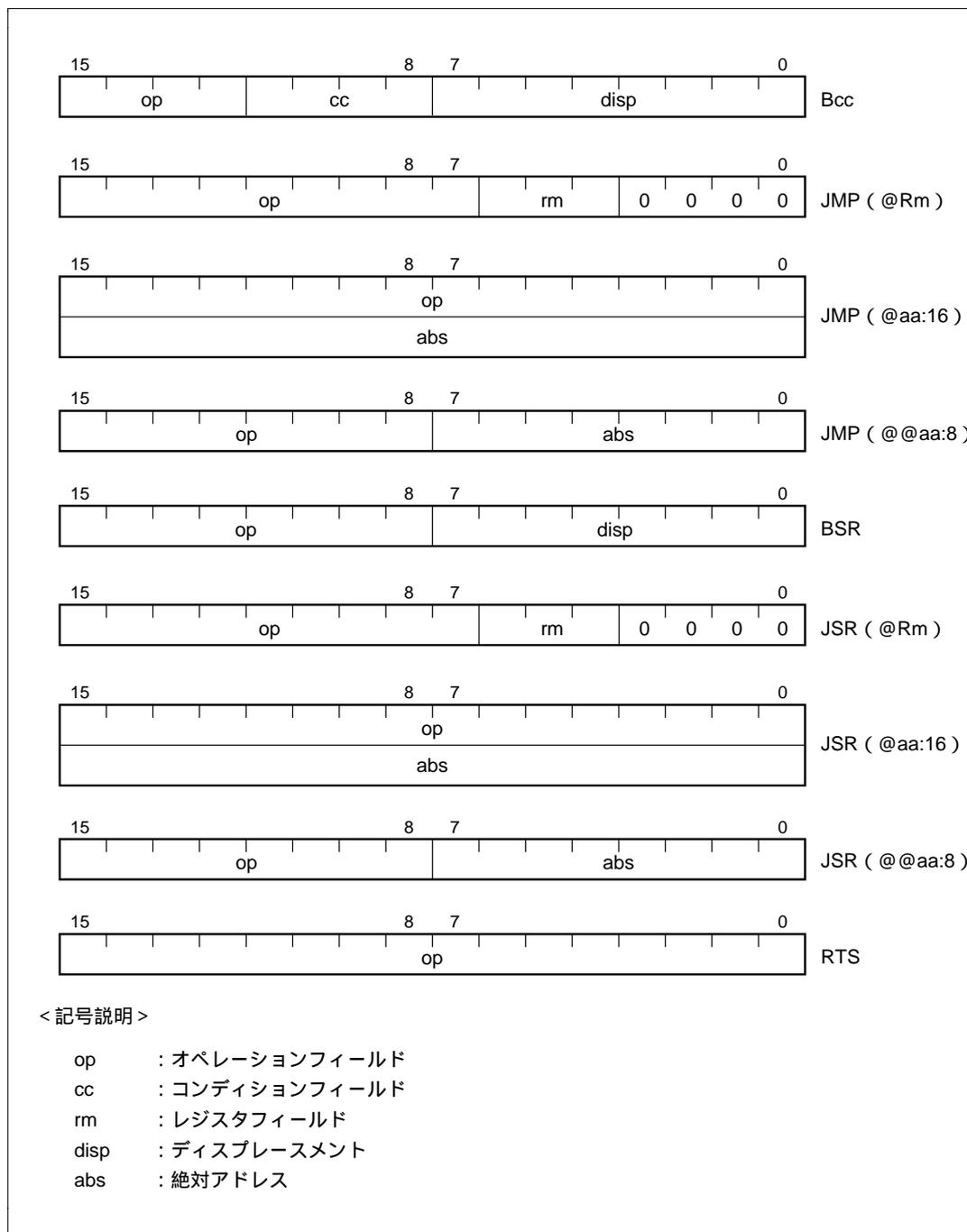


図 2.8 分岐命令の命令フォーマット

## 2.5.7 システム制御命令

システム制御命令の機能を表 2.10 に示します。

表 2.10 システム制御命令

命令	サイズ*	機能
RTE		割込み処理ルーチンから復帰します。
SLEEP		アクティブモードで本命令を実行すると、低消費電力モードに遷移します。 詳細は「第 5 章 低消費電力モード」を参照してください。
LDC	B	Rs CCR、#IMM CCR 汎用レジスタの内容、またはイミディエイトデータを CCR に転送します。
STC	B	CCR Rd CCR の内容を汎用レジスタに転送します。
ANDC	B	CCR #IMM CCR CCR とイミディエイトデータの論理積をとります。
ORC	B	CCR #IMM CCR CCR とイミディエイトデータの論理和をとります。
XORC	B	CCR⊕#IMM CCR CCR とイミディエイトデータの排他的論理和をとります。
NOP		PC+2 PC PC のインクリメントだけを行います。

【注】 \* サイズはオペランドサイズを示します。

B：バイト

システム制御命令の命令フォーマットを図 2.9 に示します。

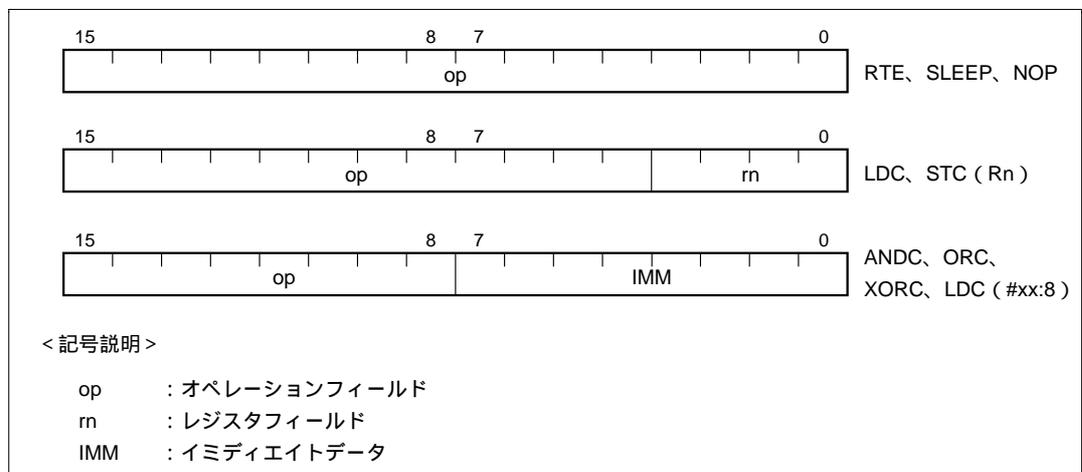


図 2.9 システム制御命令の命令フォーマット

## 2.5.8 ブロック転送命令

ブロック転送命令の機能を表 2.11 に示します。

表 2.11 ブロック転送命令

命令	サイズ	機能
EPMOV		<pre>if R4L 0 then   Repeat @R5+ @R6+, R4L-1 R4L   Until R4L=0 else next;</pre> <p>ブロック転送命令です。R5 で示されるアドレスから始まり、R4L で指定されるバイト数のデータを R6 で示されるアドレスから始まるロケーションへ転送します。転送終了後、次の命令を実行します。</p>

EPMOV 命令には、使用上の注意事項があります。詳細は「2.9.3 EPMOV 命令使用上の注意事項」を参照してください。

ブロック転送命令の命令フォーマットを図 2.10 に示します。

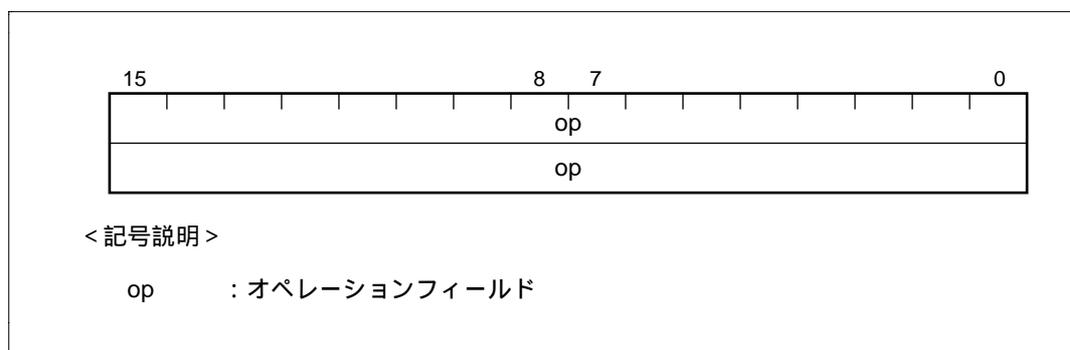


図 2.10 ブロック転送命令の命令フォーマット

## 2.6 基本動作タイミング

CPUは、システムクロック( )またはサブクロック(  $\text{SUB}$  )を基準に動作しています。システムクロック およびサブクロック  $\text{SUB}$  の定義については「第4章 クロック発振器」を参照してください。 または  $\text{SUB}$  の立上がりから次の立上がりまでの1単位をステートと呼びます。バスサイクルは、2ステートまたは3ステートで構成され、内蔵メモリ、内蔵周辺モジュールによって異なるアクセスを行います。

### 2.6.1 内蔵メモリ (RAM、ROM)

内蔵メモリのアクセスは、2ステートで行われます。このとき、データバス幅は16ビットで、バイトおよびワードサイズアクセスが可能です。

内蔵メモリアccessサイクルを図2.11に示します。

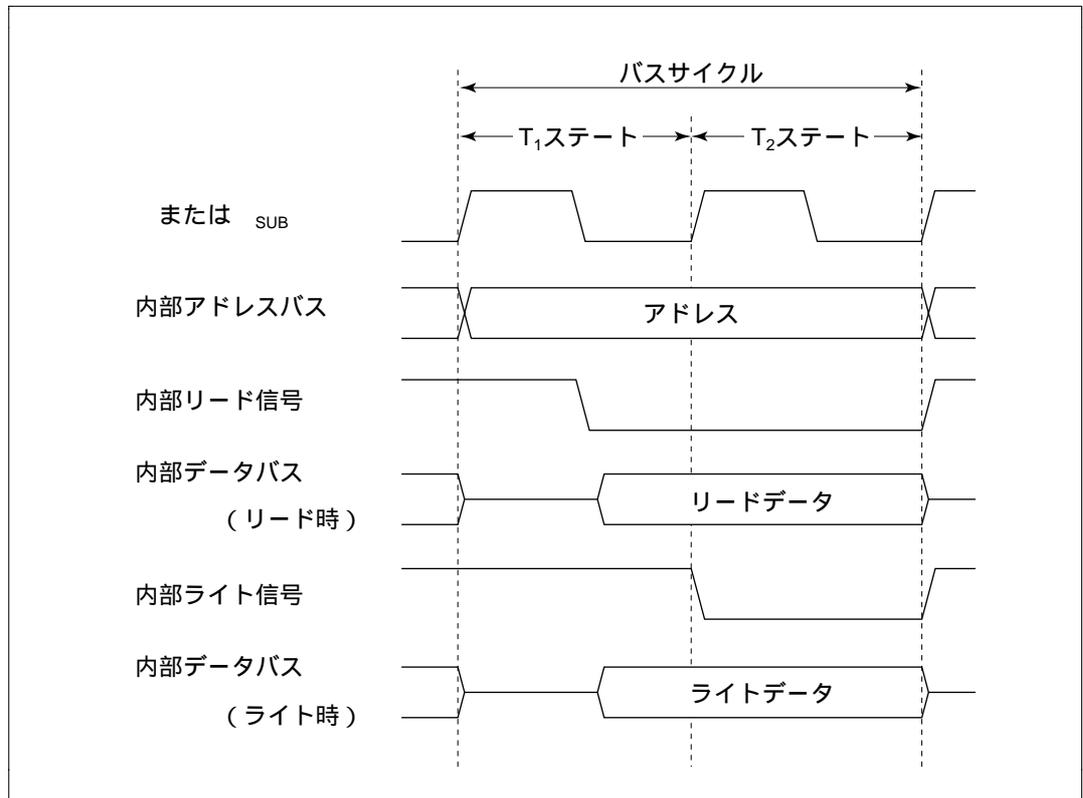


図 2.11 内蔵メモリアccessサイクル

## 2.6.2 内蔵周辺モジュール

内蔵周辺モジュールのアクセスは、2ステートまたは3ステートで行われます。このとき、データバス幅は8ビットで、バイトサイズアクセスのみ可能です。したがって、ワードデータは、2命令に分けてアクセスしてください。

### (1) 内蔵周辺モジュール2ステートアクセス

内蔵周辺モジュールを2ステートでアクセスした動作タイミングを図2.12に示します。

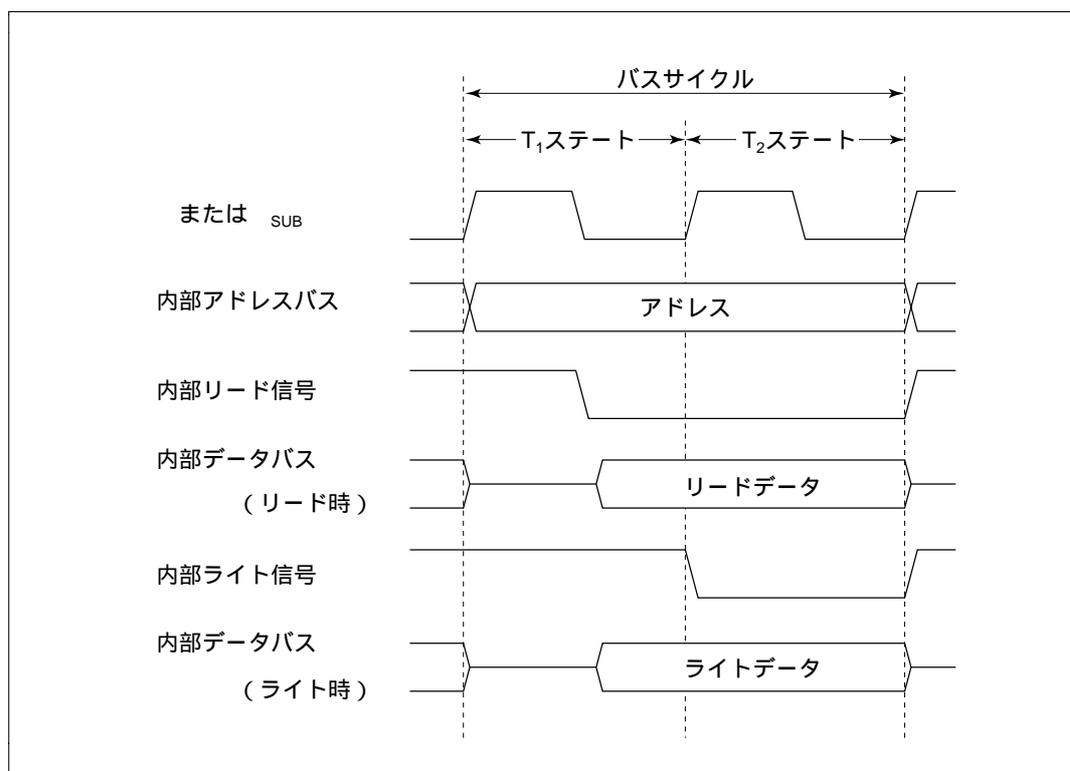


図 2.12 内蔵周辺モジュールアクセスサイクル(2ステートアクセス)

## (2) 内蔵周辺モジュール 3 ステートアクセス

内蔵周辺モジュールを 3 ステートでアクセスした動作タイミングを図 2.13 に示します。

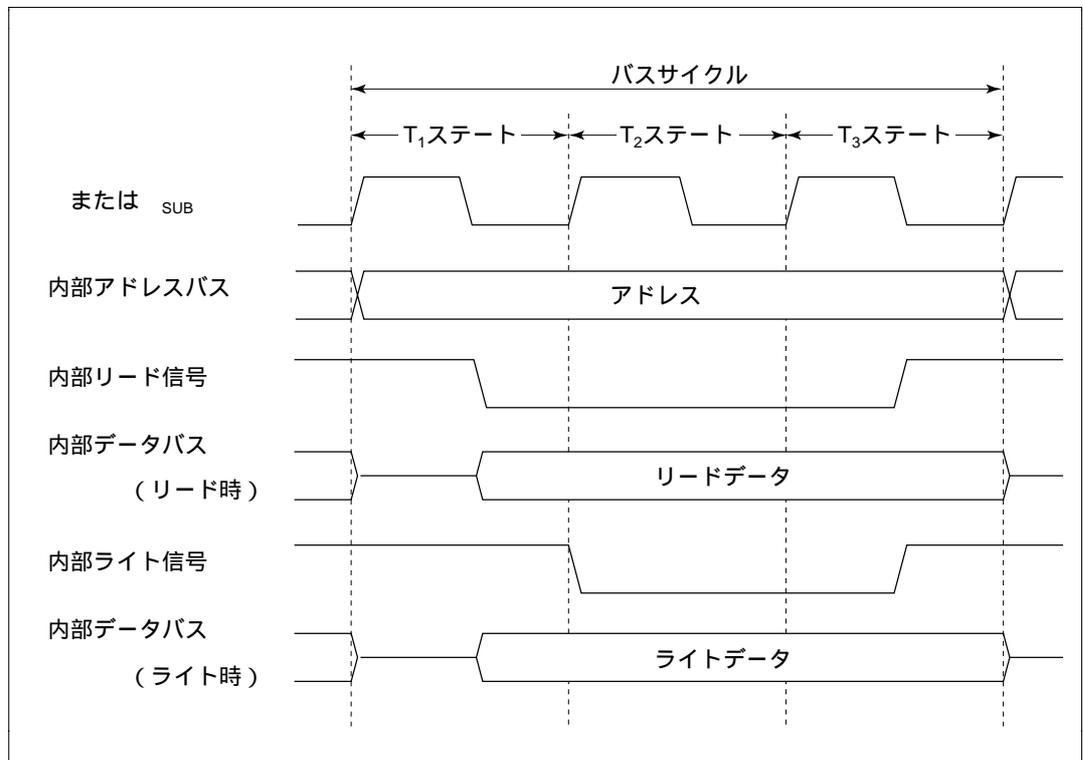


図 2.13 内蔵周辺モジュールアクセスサイクル (3 ステートアクセス)

## 2.7 CPUの状態

### 2.7.1 概要

CPUの状態には、リセット状態、プログラム実行状態、プログラム停止状態、例外処理状態の4種類があります。プログラム実行状態には、アクティブ(高速、中速)モード、サブアクティブモードがあり、プログラム停止状態には、スリープモード、スタンバイモード、ウォッチモード、およびサブスリープモードがあります。

各状態の分類を図2.14に、各状態間の遷移を図2.15に示します。

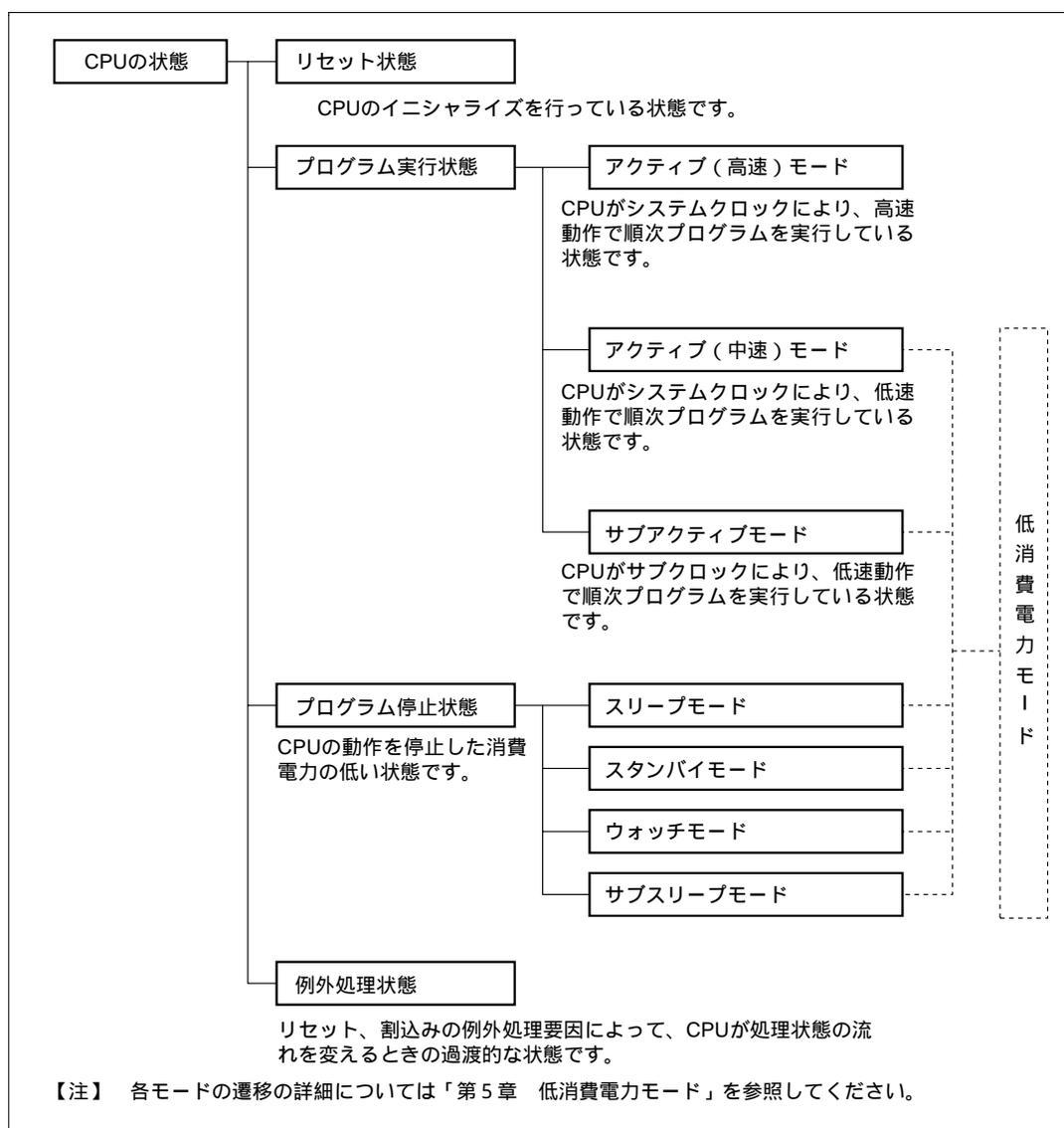


図2.14 CPUの状態の分類

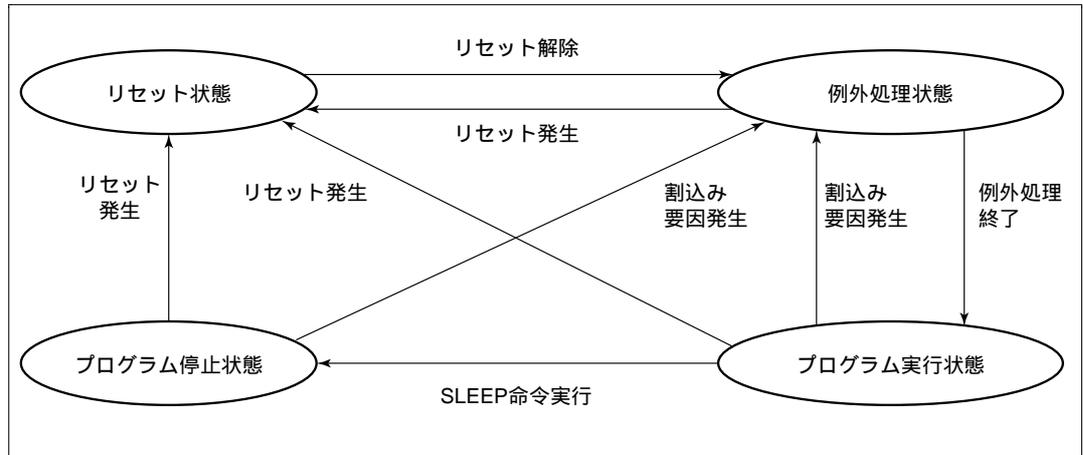


図 2.15 状態遷移図

## 2.7.2 プログラム実行状態

CPU がプログラムを順次実行している状態です。

CPU がプログラム実行状態には、アクティブ（高速、中速）モードとサブアクティブモードの3つのモードがあります。アクティブ（高速、中速）モードはシステムクロックで、サブアクティブモードはサブクロックで動作します。

これらのモードについての詳細は「第5章 低消費電力モード」を参照してください。

## 2.7.3 プログラム停止状態

プログラム停止状態には、スリープモード、スタンバイモード、ウォッチモード、およびサブスリープモードの4つのモードがあります。

これらのモードについての詳細は「第5章 低消費電力モード」を参照してください。

## 2.7.4 例外処理状態

リセット、割り込みの例外処理要因によって、CPU が通常の処理状態の流れを変えるときの過渡的な状態です。割り込み要因による例外処理では、SP (R7) を参照して、PC および CCR の退避を行います。

割り込み処理についての詳細は、「3.3 割り込み」を参照してください。

## 2.8 メモリマップ

H8/3627 シリーズのメモリマップを図 2.16 に示します。

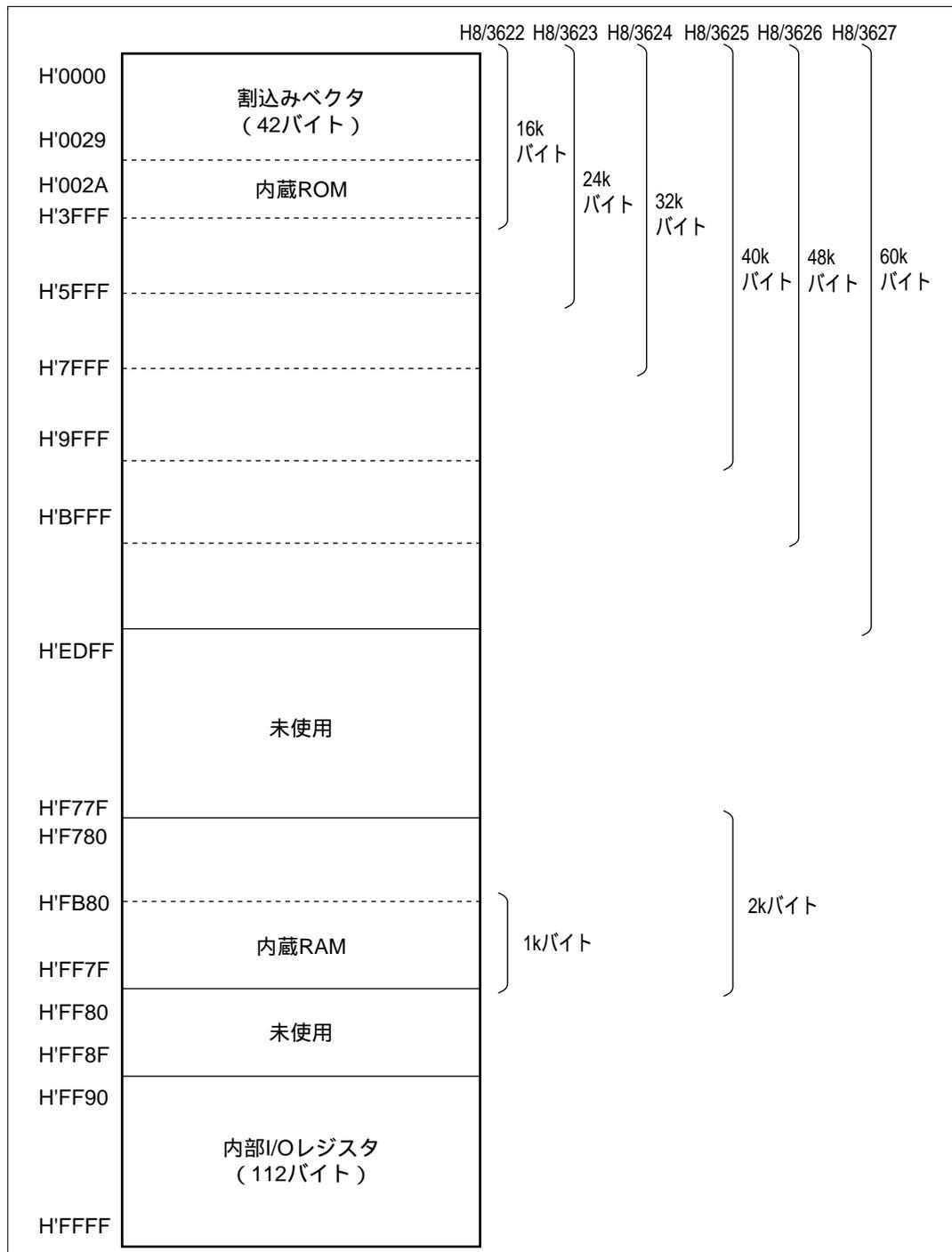


図 2.16 H8/3627 シリーズのメモリマップ

## 2.9 使用上の注意事項

### 2.9.1 データアクセスに関する注意事項

#### (1) 空きエリアへのアクセス

H8/300L CPU のアドレス空間には、ユーザに開放された ROM、RAM、レジスタ以外の空きエリアがあります。プログラムで誤ってこの空きエリアにアクセスを行うと、以下のように動作します。

<p><u>CPU から空きエリアへのデータの転送</u></p>
<p>転送データは失われます。また、CPU 誤動作の原因となる可能性があります。</p>
<p><u>空きエリアから CPU へのデータの転送</u></p>
<p>転送データは保証されません。</p>

#### (2) 内部 I/O レジスタへのアクセス

内蔵 ROM、RAM 領域以外の内蔵周辺モジュールは、内部のデータ転送が 8 ビットで行われます。この領域にワードアクセスを行うと、以下のように動作します。

<p><u>CPU から I/O レジスタ領域へのワードアクセス</u></p>
<p>上位バイト : I/O レジスタに書き込まれます。</p>
<p>下位バイト : 転送データは失われます。</p>
<p><u>内部 I/O レジスタから CPU へのワードアクセス</u></p>
<p>上位バイト : CPU 内部レジスタ上位に書き込まれます。</p>
<p>下位バイト : CPU 内部レジスタ下位に書き込まれたデータは保証されません。</p>

したがって、内蔵 ROM、RAM 領域以外の I/O レジスタ領域とのデータ転送は、バイトサイズの命令を使用してください。

図 2.17 にアクセスできるデータサイズおよびステート数と内蔵周辺モジュールの対応を示します。

アドレス	モジュール	アクセス		
		ワード	バイト	ステート数
H'0000	割込みベクタ (42バイト)			2
H'0029				
H'002A	内蔵ROM			2
H'3FFF				
H'5FFF				2
H'7FFF				
H'9FFF				2
H'BFFF				
H'EDFF	未使用	—	—	—
H'F77F	内蔵RAM			2
H'F780				
H'FF7F	未使用	—	—	—
H'FF80				2 または 3
H'FF8F				
H'FF90	内部I/Oレジスタ (112バイト)			2 または 3
H'FFFF				

図 2.17 アクセスできるデータサイズおよびステート数と内蔵周辺モジュールの対応

## 2.9.2 ビット操作命令使用上の注意事項

BSET、BCLR、BNOT、BST、BIST の各命令は、バイト単位でデータをリードし、ビット操作後に再びバイト単位でデータをライトします。

したがって、同一アドレスに2つのレジスタが割り付けられている場合や、ライト専用ビットを含むレジスタ、またはポートに対してこれらの命令を使用する場合には注意が必要です。

動作順序	動作内容	
1	リード	指定したアドレスのデータ (バイト単位) をリードします。
2	ビット操作	リードしたデータの指定された1ビットを操作します。
3	ライト	指定したアドレスに操作したデータ (バイト単位) をライトします。

### (1) 同一アドレスに割り付けられた2つのレジスタのビット操作

#### 例1：タイマロードレジスタとタイマカウンタへのビット操作

図 2.18 に同一アドレスに割り付けられた2つのレジスタを持つタイマの構成例を示します。

リロードタイマのタイマロードレジスタとタイマカウンタにビット操作命令を実行した場合、タイマロードレジスタとタイマカウンタはアドレスを共有しているため、次のように動作します。

動作順序	動作内容	
1	リード	タイマカウンタのデータ (バイト単位) をリードします。
2	ビット操作	CPU は命令で指定された1ビットを操作 (セットまたはリセット) します。
3	ライト	操作したデータ (バイト単位) をタイマロードレジスタにライトします。

タイマカウンタは、カウントを続けているのでリードした値がタイマロードレジスタとは必ずしも等しくありません。その結果操作の対象となったビット以外は、タイマカウンタの値がタイマロードレジスタへライトされます。

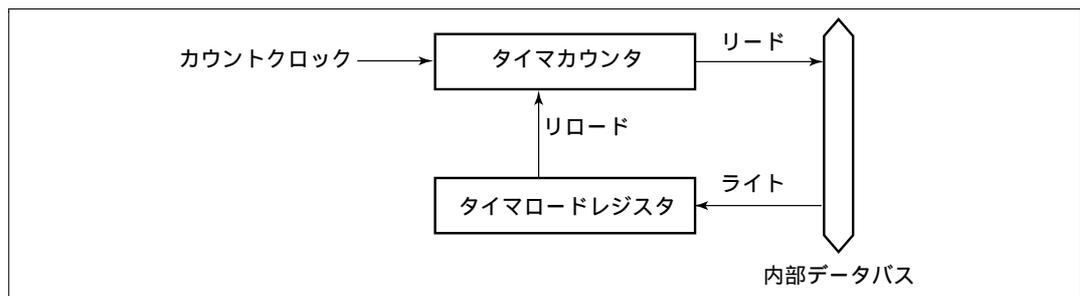


図 2.18 タイマの構成例

## 例2：ポート6にBSET命令を実行した場合

P6<sub>7</sub>、P6<sub>6</sub>は入力端子に設定され、それぞれ"Low"レベル、"High"レベルが入力されているとし、P6<sub>5</sub>~P6<sub>0</sub>は出力端子に設定され、それぞれ"Low"レベル出力状態とします。

以下に、BSET命令でP6<sub>0</sub>に"High"レベル出力を行う例を示します。

## 【A； BSET命令を実行前】

	P6 <sub>7</sub>	P6 <sub>6</sub>	P6 <sub>5</sub>	P6 <sub>4</sub>	P6 <sub>3</sub>	P6 <sub>2</sub>	P6 <sub>1</sub>	P6 <sub>0</sub>
入出力	入力	入力	出力	出力	出力	出力	出力	出力
端子状態	Low レベル	High レベル	Low レベル	Low レベル	Low レベル	Low レベル	Low レベル	Low レベル
PCR6	0	0	1	1	1	1	1	1
PDR6	1	0	0	0	0	0	0	0

## 【B； BSET命令を実行】

BSET	#0,	@PDR6
------	-----	-------

ポート6に対してBSET命令を実行します。

## 【C； BSET命令を実行後】

	P6 <sub>7</sub>	P6 <sub>6</sub>	P6 <sub>5</sub>	P6 <sub>4</sub>	P6 <sub>3</sub>	P6 <sub>2</sub>	P6 <sub>1</sub>	P6 <sub>0</sub>
入出力	入力	入力	出力	出力	出力	出力	出力	出力
端子状態	Low レベル	High レベル	Low レベル	Low レベル	Low レベル	Low レベル	Low レベル	High レベル
PCR6	0	0	1	1	1	1	1	1
PDR6	0	1	0	0	0	0	0	1

## 【D； BSET命令の動作説明】

BSET命令を実行すると、CPUは、最初にポート6をリードします。

P6<sub>7</sub>、P6<sub>6</sub>は入力端子であるので、CPUは端子の状態("Low"レベル、"High"レベル入力)をリードします。P6<sub>5</sub>~P6<sub>0</sub>は出力端子であるので、CPUはPDR6の値をリードします。したがって、この例では、PDR6はH'80ですが、CPUがリードしたデータはH'40となります。

次に、CPUは、リードしたデータのビット0を"1"にセットして、データをH'41に変更します。

最後に、この値(H'41)をPDR6に書き込んで、BSET命令を終了します。

その結果、PDR6のビット0が"1"になり、P6<sub>0</sub>は"High"レベル出力になります。しかし、PDR6のビット7、6が変化してしまいます。

そのため、PDR6と同じデータをメモリ上のワークエリアに格納し、ワークエリア上の

データに対しビット操作を行った後、このデータを PDR6 にライトしてください。

**【A ; BSET 命令を実行前】**

```
MOV.B #80, R0L
MOV.B R0L, @RAM0
MOV.B R0L, @PDR6
```

PDR6 に書き込む値 (H'80) をあらかじめメモリ上のワークエリア (RAM0) と PDR6 にライトします。

	P6 <sub>7</sub>	P6 <sub>6</sub>	P6 <sub>5</sub>	P6 <sub>4</sub>	P6 <sub>3</sub>	P6 <sub>2</sub>	P6 <sub>1</sub>	P6 <sub>0</sub>
入出力	入力	入力	出力	出力	出力	出力	出力	出力
端子状態	Low レベル	High レベル	Low レベル	Low レベル	Low レベル	Low レベル	Low レベル	Low レベル
PCR6	0	0	1	1	1	1	1	1
PDR6	1	0	0	0	0	0	0	0

RAM0	1	0	0	0	0	0	0	0
------	---	---	---	---	---	---	---	---

**【B ; BSET 命令を実行】**

```
BSET #0, @RAM0
```

PDR6 のワークエリア (RAM0) に対して BSET 命令を実行します。

**【C ; BSET 命令を実行後】**

```
MOV.B @RAM0, R0L
MOV.B R0L, @PDR6
```

ワークエリア (RAM0) の値を PDR6 にライトします。

	P6 <sub>7</sub>	P6 <sub>6</sub>	P6 <sub>5</sub>	P6 <sub>4</sub>	P6 <sub>3</sub>	P6 <sub>2</sub>	P6 <sub>1</sub>	P6 <sub>0</sub>
入出力	入力	入力	出力	出力	出力	出力	出力	出力
端子状態	Low レベル	High レベル	Low レベル	Low レベル	Low レベル	Low レベル	Low レベル	High レベル
PCR6	0	0	1	1	1	1	1	1
PDR6	1	0	0	0	0	0	0	1

RAM0	1	0	0	0	0	0	0	1
------	---	---	---	---	---	---	---	---

## (2) ライト専用ビットを含むレジスタのビット操作

例3：ポート6のPCR6にBCLR命令を実行した場合

P6<sub>7</sub>、P6<sub>6</sub>は入力端子に設定され、それぞれ"Low"レベル、"High"レベルが入力されています。P6<sub>5</sub>～P6<sub>0</sub>は出力端子に設定され、それぞれ"Low"レベル出力状態とします。

ここで、BCLR命令で、P6<sub>0</sub>を入力ポートにする例を示します。入力端子に設定されたP6<sub>0</sub>は"High"レベルが入力されるものとします。

## 【A； BCLR命令を実行前】

	P6 <sub>7</sub>	P6 <sub>6</sub>	P6 <sub>5</sub>	P6 <sub>4</sub>	P6 <sub>3</sub>	P6 <sub>2</sub>	P6 <sub>1</sub>	P6 <sub>0</sub>
入出力	入力	入力	出力	出力	出力	出力	出力	出力
端子状態	Low レベル	High レベル	Low レベル	Low レベル	Low レベル	Low レベル	Low レベル	Low レベル
PCR6	0	0	1	1	1	1	1	1
PDR6	1	0	0	0	0	0	0	0

## 【B； BCLR命令を実行】

BCLR	#0, @PCR6
------	-----------

PCR6に対してBCLR命令を実行します。

## 【C； BCLR命令を実行後】

	P6 <sub>7</sub>	P6 <sub>6</sub>	P6 <sub>5</sub>	P6 <sub>4</sub>	P6 <sub>3</sub>	P6 <sub>2</sub>	P6 <sub>1</sub>	P6 <sub>0</sub>
入出力	出力	入力						
端子状態	Low レベル	High レベル	Low レベル	Low レベル	Low レベル	Low レベル	Low レベル	High レベル
PCR6	1	1	1	1	1	1	1	0
PDR6	1	0	0	0	0	0	0	0

## 【D； BCLR命令の動作説明】

BCLR命令を実行すると、CPUは、最初にPCR6をリードします。PCR6はライト専用レジスタですので、CPUはH'FFをリードします。したがって、この例ではPCR6はH'3Fですが、CPUがリードしたデータはH'FFとなります。

次に、CPUは、リードしたデータのビット0を"0"にクリアして、データをH'FEに変更します。

最後に、このデータ（H'FE）をPCR6に書き込んで、BCLR命令を終了します。

その結果、PCR6のビット0が"0"になり、P6<sub>0</sub>は入力ポートになります。しかし、PCR6のビット7、ビット6が1になって、入力ポートであったP6<sub>7</sub>、P6<sub>6</sub>は出力ポートに変化してしまいます。

そのため、PCR6 と同じデータをメモリ上のワークエリアに格納し、ワークエリア上のデータに対しビット操作を行った後、このデータを PCR6 にライトしてください。

**【A; BCLR 命令を実行前】**

```
MOV.B #3F, R0L
MOV.B R0L, @RAM0
MOV.B R0L, @PCR6
```

PCR6 に書き込む値 (H'3F) をあらかじめメモリ上のワークエリア (RAM0) と PCR6 にライトします。

	P6 <sub>7</sub>	P6 <sub>6</sub>	P6 <sub>5</sub>	P6 <sub>4</sub>	P6 <sub>3</sub>	P6 <sub>2</sub>	P6 <sub>1</sub>	P6 <sub>0</sub>
入出力	入力	入力	出力	出力	出力	出力	出力	出力
端子状態	Low レベル	High レベル	Low レベル	Low レベル	Low レベル	Low レベル	Low レベル	Low レベル
PCR6	0	0	1	1	1	1	1	1
PDR6	1	0	0	0	0	0	0	0

RAM0	0	0	1	1	1	1	1	1
------	---	---	---	---	---	---	---	---

**【B; BCLR 命令を実行】**

```
BCLR #0, @RAM0
```

PCR6 のワークエリア (RAM0) に対して BCLR 命令を実行します。

**【C; BCLR 命令を実行後】**

```
MOV.B @RAM0, R0L
MOV.B R0L, @PCR6
```

ワークエリア (RAM0) の値を PCR6 にライトします。

	P6 <sub>7</sub>	P6 <sub>6</sub>	P6 <sub>5</sub>	P6 <sub>4</sub>	P6 <sub>3</sub>	P6 <sub>2</sub>	P6 <sub>1</sub>	P6 <sub>0</sub>
入出力	入力	入力	出力	出力	出力	出力	出力	出力
端子状態	Low レベル	High レベル	Low レベル	Low レベル	Low レベル	Low レベル	Low レベル	High レベル
PCR6	0	0	1	1	1	1	1	0
PDR6	1	0	0	0	0	0	0	0

RAM0	0	0	1	1	1	1	1	0
------	---	---	---	---	---	---	---	---

同一のアドレスに割り付けられた2つのレジスタの一覧を表 2.12 に、ライト専用ビットを含むレジスタの一覧を表 2.13 に示します。

表 2.12 同一のアドレスに割り付けられた2つのレジスタの一覧

レジスタ名	略称	アドレス
ポートデータレジスタ 1*	PDR1	H'FFD4
ポートデータレジスタ 2*	PDR2	H'FFD5
ポートデータレジスタ 5*	PDR5	H'FFD8
ポートデータレジスタ 6*	PDR6	H'FFD9
ポートデータレジスタ 7*	PDR7	H'FFDA
ポートデータレジスタ 8*	PDR8	H'FFDB
ポートデータレジスタ A*	PDRA	H'FFDD

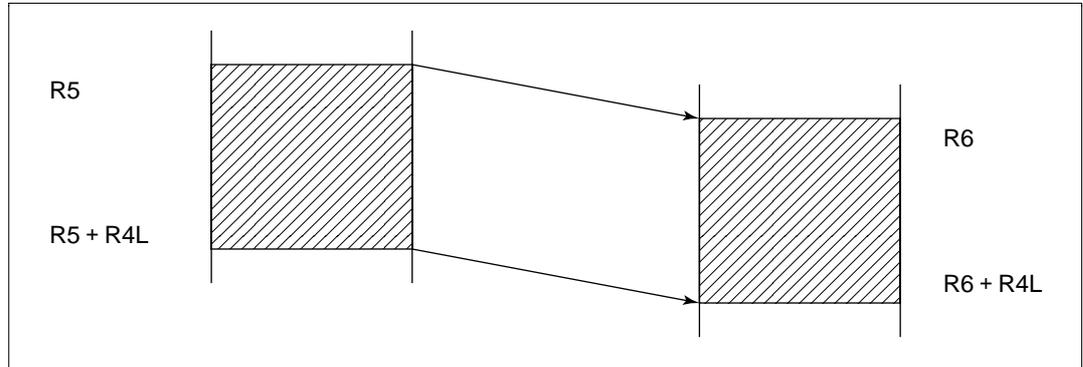
【注】 \* ポートデータレジスタと端子入力が兼用になっています。

表 2.13 ライト専用ビットを含むレジスタの一覧

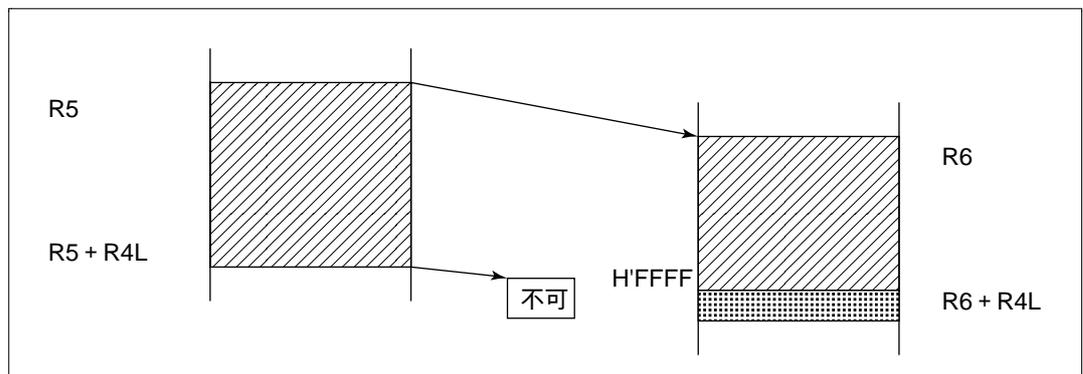
レジスタ名	略称	アドレス
ポートコントロールレジスタ 1	PCR1	H'FFE4
ポートコントロールレジスタ 2	PCR2	H'FFE5
ポートコントロールレジスタ 5	PCR5	H'FFE8
ポートコントロールレジスタ 6	PCR6	H'FFE9
ポートコントロールレジスタ 7	PCR7	H'FFEA
ポートコントロールレジスタ 8	PCR8	H'FFEB
ポートコントロールレジスタ A	PCRA	H'FFED
タイマコントロールレジスタ F	TCRF	H'FFB6

### 2.9.3 EEPMOV 命令使用上の注意事項

- (1) EEPMOV 命令はブロック転送命令で、R5 で示されるアドレスから始まる R4L で示されるバイト数のデータを、R6 で示されるアドレスへ転送します。



- (2) 転送先の最終アドレス (R6+R4L の値) が H'FFFF を超えないように (実行途中で R6 の値が H'FFFF H'0000 とならないように)、R4L、R6 を設定してください。



---

# 3. 例外処理

---

## 第3章 目次

3.1	概要	61
3.2	リセット	62
	3.2.1 概要	62
	3.2.2 リセットシーケンス	62
	3.2.3 リセット直後の割込み	64
3.3	割込み	65
	3.3.1 概要	65
	3.3.2 各レジスタの説明	67
	3.3.3 外部割込み	76
	3.3.4 内部割込み	77
	3.3.5 割込み動作	77
	3.3.6 割込み応答時間	82
3.4	使用上の注意事項	83
	3.4.1 スタック領域に関する使用上の注意事項	83
	3.4.2 ポートモードレジスタを書換える際の注意事項	84



## 3.1 概要

本 LSI の例外処理には、リセットと割込みがあります。表 3.1 に、例外処理の種類と優先度を示します。

表 3.1 例外処理の種類と優先度

優先度	例外処理要因	例外処理開始タイミング
高 ↑ 低	リセット	リセットが解除されると例外処理を開始します。
	割込み	割込み要求が発生すると、命令の実行終了時または例外処理終了時に例外処理を開始します。

## 3.2 リセット

### 3.2.1 概要

リセットは最も優先順位の高い例外処理です。

リセットによって、CPUの内部状態と、内蔵周辺モジュールの各レジスタがイニシャライズされます。

### 3.2.2 リセットシーケンス

$\overline{\text{RES}}$  端子が"Low"レベルになると、実行中の処理はすべて打ち切れ、本 LSI はリセット状態になります。

本 LSI を確実にリセットするために、電源投入時には、クロック発振器の発振安定時間の間  $\overline{\text{RES}}$  端子を"Low"レベルに保持してください。また、外部クロック、セラミック発振子使用時でも電源投入時には、「第 14 章 電気的特性 表 14.3」に示す水晶発振子の発振安定時間と同等の間、 $\overline{\text{RES}}$  端子を"Low"レベルに保持してください。また、動作中にリセットする場合は、最低 18 クロックの間、"Low"レベルに保持してください。 $\overline{\text{RES}}$  端子が一定期間"Low"レベルの後、"High"レベルになると、リセット例外処理が開始されます。リセット例外処理の動作は以下のとおりです。

- (1) CPU の内部状態と内蔵周辺モジュールの各レジスタのイニシャライズを行い、コンディションコードレジスタ (CCR) の I ビットをセットします。
- (2) リセット例外処理ベクタアドレス (H'0000 ~ H'0001) をリードして PC に転送した後、PC で示されるアドレスからプログラムの実行を開始します。

パワーオン / パワーオフ時には、 $\overline{\text{RES}}$  端子を"Low"レベルにしてください。  
リセットシーケンスを図 3.1 に示します。

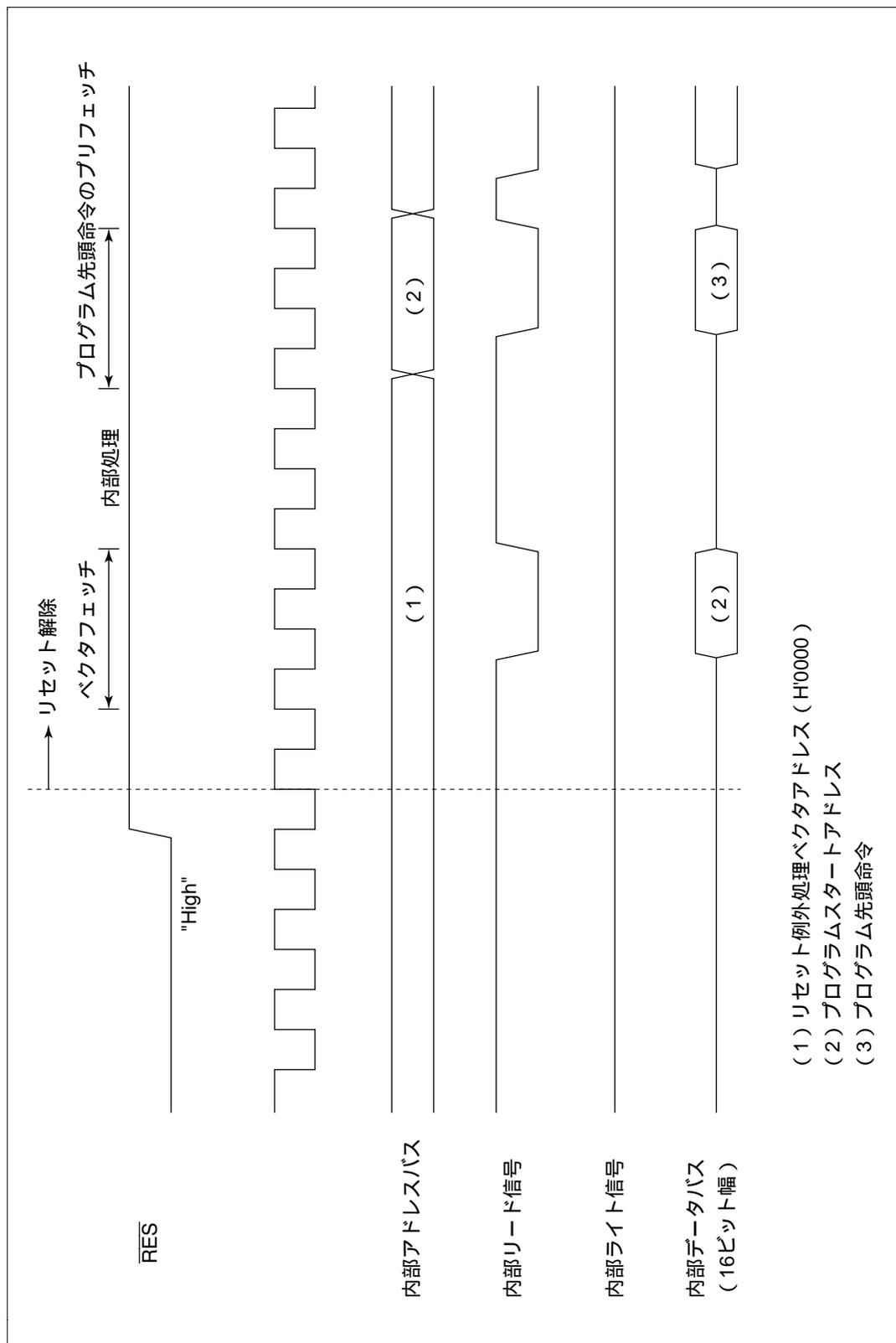


図 3.1 リセットシーケンス

### 3.2.3 リセット直後の割込み

リセット後、スタックポインタ (SP: R7) をイニシャライズする前に割込みを受け付けると、PC と CCR の退避が正常に行われなため、プログラムの暴走の原因となります。これを防ぐため、リセット例外処理が実行された直後はすべての割込み要求が禁止されています。すなわち、リセット直後はプログラムの先頭 1 命令が必ず実行されますので、プログラムの先頭命令は、SP をイニシャライズする命令としてください(例:MOV.W #xx:16, SP)。

## 3.3 割込み

### 3.3.1 概要

割込み例外処理を開始する要因には、13の外部割込み要因(WKP<sub>7</sub>~WKP<sub>0</sub>、IRQ<sub>4</sub>~IRQ<sub>0</sub>)と内蔵モジュールから16の内部割込み要因があります。割込み要因と優先度、ならびにベクタアドレスの一覧を表3.2に示します。複数の割込み要求が発生したときは、優先度の高い方から処理されます。

これらの割込みには次のような特長があります。

- (1) 内部割込みおよび外部割込みは、CCRのIビットによりマスクされます。すなわちCCRのIビットが"1"にセットされていると、割込み要求フラグはセットされますが、割込みは受け付けられません。
- (2) IRQ<sub>4</sub>~IRQ<sub>0</sub>は、立上がり/立下がりエッジセンスのいずれかに設定することができます。

表 3.2 割込み優先順位

割込み要因発生元	割込み要因	ベクタNo.	ベクタアドレス	優先順位
RES	リセット	0	H'0000 ~ H'0001	高   低
IRQ <sub>0</sub>	IRQ <sub>0</sub>	4	H'0008 ~ H'0009	
$\overline{\text{IRQ}}_1$	IRQ <sub>1</sub>	5	H'000A ~ H'000B	
$\overline{\text{IRQ}}_2$	IRQ <sub>2</sub>	6	H'000C ~ H'000D	
$\overline{\text{IRQ}}_3$	IRQ <sub>3</sub>	7	H'000E ~ H'000F	
$\overline{\text{IRQ}}_4$	IRQ <sub>4</sub>	8	H'0010 ~ H'0011	
$\overline{\text{WKP}}_0$	WKP <sub>0</sub>	9	H'0012 ~ H'0013	
$\overline{\text{WKP}}_1$	WKP <sub>1</sub>			
$\overline{\text{WKP}}_2$	WKP <sub>2</sub>			
$\overline{\text{WKP}}_3$	WKP <sub>3</sub>			
$\overline{\text{WKP}}_4$	WKP <sub>4</sub>			
$\overline{\text{WKP}}_5$	WKP <sub>5</sub>			
$\overline{\text{WKP}}_6$	WKP <sub>6</sub>			
$\overline{\text{WKP}}_7$	WKP <sub>7</sub>			
SCI1	SCI1転送完了	10	H'0014 ~ H'0015	
タイマA	タイマオーバフロー	11	H'0016 ~ H'0017	
タイマFL	タイマFLコンペアマッチ タイマFLオーバフロー	14	H'001C ~ H'001D	
タイマFH	タイマFHコンペアマッチ タイマFHオーバフロー	15	H'001E ~ H'001F	
タイマG	タイマGインプットキャプチャ タイマGオーバフロー	16	H'0020 ~ H'0021	
SCI3	SCI3受信データフル SCI3送信データエンプティ SCI3送信終了 SCI3オーバランエラー SCI3フレーミングエラー SCI3パリティエラー	18	H'0024 ~ H'0025	
A/D	A/D変換終了	19	H'0026 ~ H'0027	
(SLEEP命令の実行)	直接遷移	20	H'0028 ~ H'0029	

【注】 H'0002 ~ H'0007, H'0018 ~ H'001B, H'0022 ~ H'0023は本LSIではリザーブされており、ユーザは使用できません。

### 3.3.2 各レジスタの説明

割込みを制御するレジスタの一覧を表 3.3 に示します。

表 3.3 割込み制御レジスタ

名称	略称	R/W	初期値	アドレス
割込みエッジセレクトレジスタ	IEGR	R/W	H'60	H'FFF2
割込み許可レジスタ 1	IENR1	R/W	H'00	H'FFF3
割込み許可レジスタ 2	IENR2	R/W	H'01	H'FFF4
割込み要求レジスタ 1	IRR1	R/W*	H'20	H'FFF6
割込み要求レジスタ 2	IRR2	R/W*	H'01	H'FFF7
ウェイクアップ割込み要求レジスタ	IWPR	R/W*	H'00	H'FFF9

【注】 \* フラグクリアのための"0"ライトのみ可能です。

#### (1) 割込みエッジセレクトレジスタ (IEGR)

ビット:	7	6	5	4	3	2	1	0
	—	—	—	IEG4	IEG3	IEG2	IEG1	IEG0
初期値:	0	1	1	0	0	0	0	0
R/W :	—	—	—	R/W	R/W	R/W	R/W	R/W

IEGR は、8 ビットのリード/ライト可能なレジスタで、 $\overline{\text{IRQ}}_4 \sim \overline{\text{IRQ}}_0$  端子の立上がり / 立下がりエッジセンスを指定します。

#### ビット 7 : リザーブビット

リザーブビットです。本ビットはリードすると常に"0"が読み出されます。本ビットは"0"で使用してください。

#### ビット 6、5 : リザーブビット

リザーブビットです。各ビットはリードすると常に"1"が読み出されます。ライトは無効です。

#### ビット 4 : $\text{IRQ}_4$ エッジセレクト (IEG4)

$\overline{\text{IRQ}}_4$  端子、 $\overline{\text{ADTRG}}$  端子の入力センスを選択します。

ビット 4	説明
IEG4	
0	$\overline{\text{IRQ}}_4$ 端子、 $\overline{\text{ADTRG}}$ 端子入力の立下がりエッジを検出 (初期値)
1	$\overline{\text{IRQ}}_4$ 端子、 $\overline{\text{ADTRG}}$ 端子入力の立上がりエッジを検出

ビット3 : IRQ<sub>3</sub>エッジセレクト (IEG3)

$\overline{\text{IRQ}}_3$  端子、TMIF 端子の入力センスを選択します。

ビット3		
IEG3	説明	
0	$\overline{\text{IRQ}}_3$ 端子、TMIF 端子入力の立下がりエッジを検出	(初期値)
1	$\overline{\text{IRQ}}_3$ 端子、TMIF 端子入力の立上がりエッジを検出	

ビット2 : IRQ<sub>2</sub>エッジセレクト (IEG2)

$\overline{\text{IRQ}}_2$  端子の入力センスを選択します。

ビット2		
IEG2	説明	
0	$\overline{\text{IRQ}}_2$ 端子入力の立下がりエッジを検出	(初期値)
1	$\overline{\text{IRQ}}_2$ 端子入力の立上がりエッジを検出	

ビット1 : IRQ<sub>1</sub>エッジセレクト (IEG1)

$\overline{\text{IRQ}}_1$  端子の入力センスを選択します。

ビット1		
IEG1	説明	
0	$\overline{\text{IRQ}}_1$ 端子入力の立下がりエッジを検出	(初期値)
1	$\overline{\text{IRQ}}_1$ 端子入力の立上がりエッジを検出	

ビット0 : IRQ<sub>0</sub>エッジセレクト (IEG0)

$\overline{\text{IRQ}}_0$  端子の入力センスを選択します。

ビット0		
IEG0	説明	
0	$\overline{\text{IRQ}}_0$ 端子入力の立下がりエッジを検出	(初期値)
1	$\overline{\text{IRQ}}_0$ 端子入力の立上がりエッジを検出	

## (2) 割込み許可レジスタ 1 (IENR1)

ビット:	7	6	5	4	3	2	1	0
	IENTA	IENS1	IENWP	IEN4	IEN3	IEN2	IEN1	IEN0
初期値:	0	0	0	0	0	0	0	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IENR1 は、8 ビットのリード / ライト可能なレジスタで、割込み要求の許可 / 禁止を制御します。

## ビット 7 : タイマ A 割込みイネーブル (IENTA)

タイマ A オーバフロー割込み要求の許可 / 禁止を制御します。

ビット 7	
IENTA	説明
0	タイマ A の割込み要求を禁止 (初期値)
1	タイマ A の割込み要求を許可

## ビット 6 : SCI1 割込みイネーブル (IENS1)

SCI1 転送完了割込み要求の許可 / 禁止を制御します。

ビット 6	
IENS1	説明
0	SCI1 の割込み要求を禁止 (初期値)
1	SCI1 の割込み要求を許可

## ビット 5 : ウェイクアップ割込みイネーブル (IENWP)

WKP<sub>7</sub> ~ WKP<sub>0</sub> 割込み要求の許可 / 禁止を制御します。

ビット 5	
IENWP	説明
0	$\overline{WKP_7} \sim \overline{WKP_0}$ 端子の割込み要求を禁止 (初期値)
1	$\overline{WKP_7} \sim \overline{WKP_0}$ 端子の割込み要求を許可

ビット 4 ~ 0 : IRQ<sub>4</sub> ~ IRQ<sub>0</sub> 割込みイネーブル (IEN4 ~ IEN0)

IRQ<sub>4</sub> ~ IRQ<sub>0</sub> 割込み要求の許可 / 禁止を制御します。

ビット 4 ~ 0	
IEN4 ~ IEN0	説明
0	$\overline{IRQ_4} \sim \overline{IRQ_0}$ 端子の割込み要求を禁止 (初期値)
1	$\overline{IRQ_4} \sim \overline{IRQ_0}$ 端子の割込み要求を許可

## (3) 割込み許可レジスタ 2 (IENR2)

ビット:	7	6	5	4	3	2	1	0
	IENDT	IENAD	—	IENTG	IENTFH	IENTFL	—	—
初期値:	0	0	0	0	0	0	0	1
R/W :	R/W	R/W	—	R/W	R/W	R/W	—	—

IENR2は、8ビットのリード/ライト可能なレジスタで、割込み要求の許可/禁止を制御します。

## ビット7: 直接遷移割込みイネーブル (IENDT)

直接遷移割込み要求の許可/禁止を制御します。

ビット7		
IENDT	説明	
0	直接遷移による割込み要求を禁止	(初期値)
1	直接遷移による割込み要求を許可	

## ビット6: A/D変換器割込みイネーブル (IENAD)

A/D変換終了割込み要求の許可/禁止を制御します。

ビット6		
IENAD	説明	
0	A/D変換器の割込み要求を禁止	(初期値)
1	A/D変換器の割込み要求を許可	

## ビット5: リザーブビット

リザーブビットです。本ビットはリードすると常に"0"が読み出されます。本ビットは"0"で使用してください。

## ビット4: タイマG割込みイネーブル (IENTG)

タイマGインプットキャプチャまたはオーバフロー割込み要求の許可/禁止を制御します。

ビット4		
IENTG	説明	
0	タイマGの割込み要求を禁止	(初期値)
1	タイマGの割込み要求を許可	

## ビット3：タイマFH割込みイネーブル (IENTFH)

タイマFH コンペアマッチまたはオーバフロー割込み要求の許可 / 禁止を制御します。

ビット3		
IENTFH	説明	
0	タイマFHの割込み要求を禁止	(初期値)
1	タイマFHの割込み要求を許可	

## ビット2：タイマFL割込みイネーブル (IENTFL)

タイマFL コンペアマッチまたはオーバフロー割込み要求の許可 / 禁止を制御します。

ビット2		
IENTFL	説明	
0	タイマFLの割込み要求を禁止	(初期値)
1	タイマFLの割込み要求を許可	

## ビット1：リザーブビット

リザーブビットです。本ビットはリードすると常に"0"が読み出されます。本ビットは"0"で使用してください。

## ビット0：リザーブビット

リザーブビットです。本ビットはリードすると常に"1"が読み出されます。ライトは無効です。

なお、SCI3の割込みの制御については「10.3..2 (6) シリアルコントロールレジスタ 3 (SCR3)」を参照してください。

## (4) 割込み要求レジスタ 1 (IRR1)

ビット:	7	6	5	4	3	2	1	0
	IRR1A	IRRS1	—	IRR14	IRR13	IRR12	IRR11	IRR10
初期値:	0	0	1	0	0	0	0	0
R/W :	R/W*	R/W*	—	R/W*	R/W*	R/W*	R/W*	R/W*

【注】 \* フラグクリアのための"0"ライトのみ可能です。

IRR1は、8ビットリード/ライト可能なレジスタで、タイマA、SCI1、IRQ<sub>4</sub>~IRQ<sub>0</sub>割込み要求が発生すると対応するフラグが"1"にセットされます。各フラグは割込みが受け付けられてもオートクリアされません。各フラグをクリアする場合は"0"をライトしてクリアしてください。

## ビット7：タイマA 割込み要求フラグ (IRRTA)

ビット7	説明	
IRRTA		
0	〔クリア条件〕 IRRTA = "1"の状態 で IRRTA に "0"をライトしたとき	(初期値)
1	〔セット条件〕 タイマAのカウント値がオーバーフロー (H'FF H'00) したとき	

## ビット6：SCI1 割込み要求フラグ (IRRS1)

ビット6	説明	
IRRS1		
0	〔クリア条件〕 IRRS1 = "1"の状態 で IRRS1 に "0"をライトしたとき	(初期値)
1	〔セット条件〕 SCI1 が転送完了したとき	

## ビット5：リザーブビット

リザーブビットです。本ビットはリードすると常に"1"が読み出されます。ライトは無効です。

ビット4～0：IRQ<sub>4</sub>～IRQ<sub>0</sub> 割込み要求フラグ (IRRI4～IRRI0)

ビット n	説明	
IRRI <sub>n</sub>		
0	〔クリア条件〕 IRRI <sub>n</sub> = "1"の状態 で IRRI <sub>n</sub> に "0" をライトしたとき	(初期値)
1	〔セット条件〕 IRQ <sub>n</sub> 端子が割込み入力に設定されており、かつ当該端子に指定されたエッジが 入力されたとき	

(n = 4 ~ 0)

## (5) 割込み要求レジスタ 2 (IRR2)

ビット:	7	6	5	4	3	2	1	0
	IRRDT	IRRAD	—	IRRTG	IRRTFH	IRRTFL	—	—
初期値:	0	0	0	0	0	0	0	1
R/W :	R/W*	R/W*	—	R/W*	R/W*	R/W*	—	—

【注】 \* フラグクリアのための"0"ライトのみ可能です。

IRR2 は、直接遷移、A/D 変換器、タイマ G、タイマ FH、タイマ FL の割込み要求フラグを内蔵した 8 ビットのレジスタです。直接遷移、A/D 変換器、タイマ G、タイマ FH、タイマ FL 割込み要求が発生すると、対応するフラグが"1"にセットされます。各フラグは割込みが受け付けられてもオートクリアされません。各フラグをクリアする場合は"0"をライトしてクリアしてください。

## ビット 7 : 直接遷移割込み要求フラグ (IRRDT)

ビット 7	説明
IRRDT	
0	〔クリア条件〕 (初期値) IRRDT = "1"の状態 で IRRDT に"0"をライトしたとき
1	〔セット条件〕 DTON に"1"をセットした状態でスリープ命令を実行し直接遷移したとき

## ビット 6 : A/D 変換器割込み要求フラグ (IRRAD)

ビット 6	説明
IRRAD	
0	〔クリア条件〕 (初期値) IRRAD = "1"の状態 で IRRAD に"0"をライトしたとき
1	〔セット条件〕 A/D 変換器が変換終了し、ADSF がリセットされたとき

## ビット 5 : リザーブビット

リザーブビットです。本ビットはリードすると常に"0"が読み出されます。本ビットは"0"で使用してください。

## ビット4：タイマG 割込み要求フラグ (IRRTG)

ビット4	
IRRTG	説明
0	〔クリア条件〕 (初期値) IRRTG = "1"の状態 で IRRTG に"0"をライトしたとき
1	〔セット条件〕 TMIG 端子が TMIG 入力に設定されており、かつ当該端子に指定されたエッジが 入力されたとき。または、TMG の OVIE が"1"の状態 で TCG がオーバフロー (H'FF H'00) したとき

## ビット3：タイマFH 割込み要求フラグ (IRRTFH)

ビット3	
IRRTFH	説明
0	〔クリア条件〕 (初期値) IRRTFH = "1"の状態 で IRRTFH に"0"をライトしたとき
1	〔セット条件〕 8ビットタイマモードで TCFH と OCRFH が一致したとき、また、16ビットタイマ モードで TCF (TCFL、TCFH) と OCRF (OCRFL、OCRFH) が一致し たとき

## ビット2：タイマFL 割込み要求フラグ (IRRTFL)

ビット2	
IRRTFL	説明
0	〔クリア条件〕 (初期値) IRRTFL = "1"の状態 で IRRTFL に"0"をライトしたとき
1	〔セット条件〕 8ビットタイマモードで TCFL と OCRFL が一致したとき

## ビット1：リザーブビット

リザーブビットです。本ビットはリードすると常に"0"が読み出されます。本ビットは"0"で使用してください。

## ビット0：リザーブビット

リザーブビットです。本ビットはリードすると常に"1"が読み出されます。ライトは無効です。

## (6) ウェイクアップ割込み要求レジスタ (IWPR)

ビット:	7	6	5	4	3	2	1	0
	IWPF7	IWPF6	IWPF5	IWPF4	IWPF3	IWPF2	IWPF1	IWPF0
初期値:	0	0	0	0	0	0	0	0
R/W :	R/W*							

【注】 \* フラグクリアのための"0"ライトのみ可能です。

IWPR は、8 ビットのリード/ライト可能なレジスタで、 $\overline{WKP}_7 \sim \overline{WKP}_0$  端子がウェイクアップ入力に設定されており、当該端子に立下がりエッジが入力されたとき、対応するフラグが"1"にセットされます。各フラグは割込みが受け付けられてもオートクリアされません。各フラグをクリアする場合は"0"をライトしてクリアしてください。

## ビット7~0: ウェイクアップ割込み要求フラグ (IWPF7 ~ IWPF0)

ビット n	説明
IWPFn	
0	[クリア条件] <span style="float: right;">(初期値)</span> IWPFn = "1" の状態で IWPFn に "0" をライトしたとき
1	[セット条件] $\overline{WKP}_n$ 端子がウェイクアップ入力に設定されており、かつ当該端子に立下がりエッジが入力されたとき

(n = 7 ~ 0)

### 3.3.3 外部割込み

外部割込みには、 $WKP_7 \sim WKP_0$ 割込みと、 $IRQ_4 \sim IRQ_0$ 割込みの13要因があります。

#### (1) $WKP_7 \sim WKP_0$ 割込み

$WKP_7 \sim WKP_0$ 割込みは $\overline{WKP_7} \sim \overline{WKP_0}$ 端子の立下がりエッジ入力により要求されます。

PMR5により端子機能が $\overline{WKP_7} \sim \overline{WKP_0}$ 端子に選択された状態で立下がりエッジが入力されると、IWPRの対応するビットが"1"にセットされ、割込み要求を発生します。

ウェイクアップ割込み要求の受け付けは、IENR1のIENWPを"0"にクリアすることにより禁止できます。また、CCRのIビットを"1"にすることによりすべての割込みをマスクできます。

$WKP_7 \sim WKP_0$ 割込みの割込み例外処理が受け付けられると、CCRのIビットが"1"にセットされます。 $WKP_7 \sim WKP_0$ 割込み例外処理のベクタ番号は9です。8つの割込み要因が1つのベクタ番号に割り付けられているため、例外処理ルーチンで要因を判別してください。

#### (2) $IRQ_4 \sim IRQ_0$ 割込み

$IRQ_4 \sim IRQ_0$ 割込みは、 $\overline{IRQ_4} \sim \overline{IRQ_0}$ 端子の入力信号により要求されます。

$IRQ_4 \sim IRQ_0$ 割込みは、立上がり / 立下がりエッジセンスをIEGRのIEG4～IEG0により指定できます。

PMR2、PMR1により端子機能が $\overline{IRQ_4} \sim \overline{IRQ_0}$ 端子に選択された状態で指定されたエッジが入力されると、IRR1の対応するビットが"1"にセットされ、割込み要求を発生します。

割込み要求の受け付けは、IENR1のIEN4～IEN0を"0"にクリアすることにより、禁止できます。また、CCRのIビットを"1"にセットすることによりすべての割込みをマスクできます。

$IRQ_4 \sim IRQ_0$ 割込みの割込み例外処理が受け付けられると、CCRのIビットが"1"にセットされます。

$IRQ_4 \sim IRQ_0$ 割込み例外処理のベクタ番号は8～4です。優先順位は $IRQ_4$ (低)  $IRQ_0$ (高)の順に高くなります。詳細は表3.2を参照してください。

### 3.3.4 内部割込み

内蔵周辺モジュールからの割込みによる内部割込み要因は、16 要因あります。

内蔵周辺モジュールからの割込み要求が発生すると、IRR2、IRR1 の対応するビットが"1"にセットされます。IENR2、IENR1 の各ビットを"0"にクリアすることにより、各割込み要求の受け付けは禁止できます。また、CCR の I ビットを"1"にセットすることにより、すべての割込みをマスクできます。

これらの割込み例外処理が受け付けられると、CCR の I ビットは"1"にセットされます。ベクタ番号は 20 ~ 10 です。内蔵周辺モジュールからの割込みの優先順位については表 3.2 を参照してください。

### 3.3.5 割込み動作

割込みは、割込みコントローラによって制御されます。割込みコントローラのブロック図を図 3.2 に、割込み受け付けまでのフローを図 3.3 に示します。

割込み動作は以下のとおりです。

- (1) 割込み許可レジスタの対応するビットが"1"にセットされている状態で、割込み要因が発生したとき、割込みコントローラに対して割込み要求信号が送られます。
- (2) 割込みコントローラに割込み要求信号が送られると、割込み要求フラグがセットされます。
- (3) 割込み許可フラグが"1"にセットされている割込みの中で、優先順位にしたがって最高位の割込み要求が選択され、その他は保留となります（表 3.2 参照）。
- (4) CCR の I ビットを参照し、I ビットが"0"にクリアされている場合は、割込み要求は受け付けられますが、I ビットが"1"にセットされている場合は割込み要求は保留となります。
- (5) 割込み要求が受け付けられると、そのとき実行中の命令の処理が終了した後、PC と CCR がスタック領域に退避されます。このときのスタックの状態を図 3.4 に示します。スタックされる PC は、リターン後に実行する最初の命令のアドレスを示しています。
- (6) CCR の I ビットが"1"にセットされます。これにより、すべての割込みはマスクされます。
- (7) 受け付けた割込みに対応するベクタアドレスを生成し、そのアドレスの内容によって示されるアドレスから、割込み処理ルーチンの実行を開始します。

プログラム領域を内蔵 ROM に、スタック領域を内蔵 RAM にとった場合の割込みシーケンスを図 3.5 に示します。

- 【注】 1. 本 LSI では、割込み許可レジスタをクリアすることにより割込みをディスエーブルにする場合、または割込み要求レジスタをクリアする場合は、かならず割込みをマスクした状態 (I = "1") で行ってください。
2. I = "0" の状態で上記の操作を行うと、当該操作命令の実行と当該割込みの発生が競合した場合には、当該操作命令の実行終了時に発生した割込みに対応する例外処理を実行します。

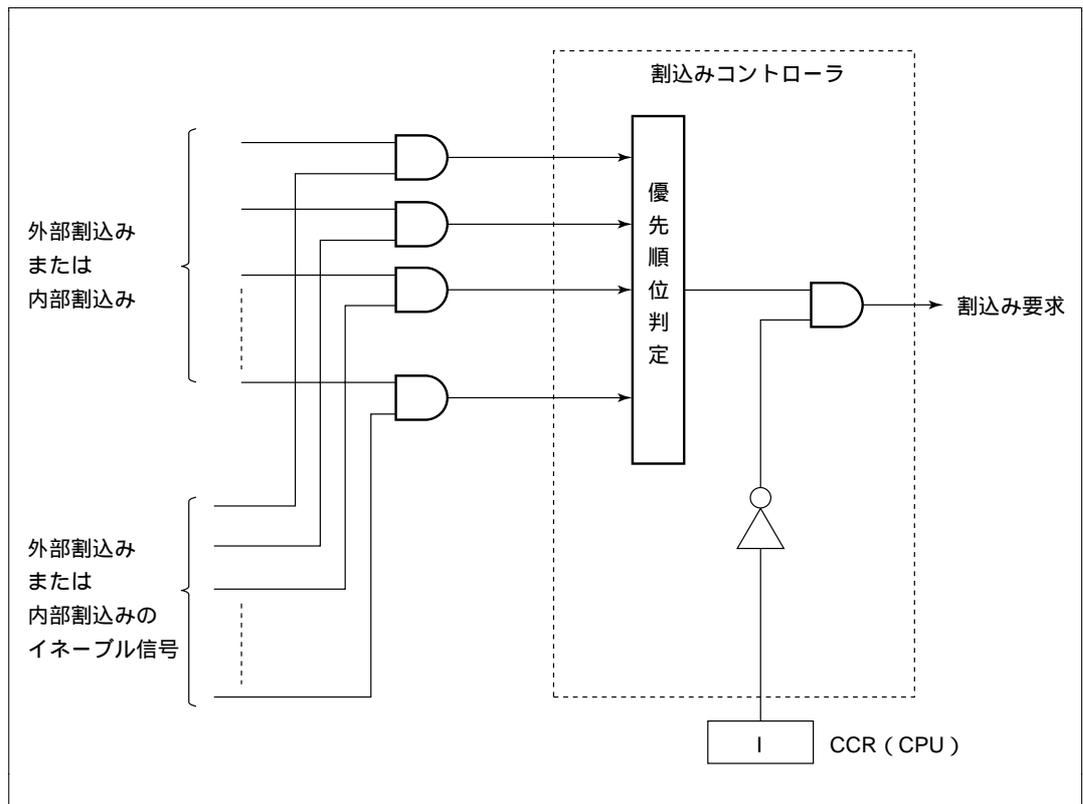


図 3.2 割込みコントローラのブロック図

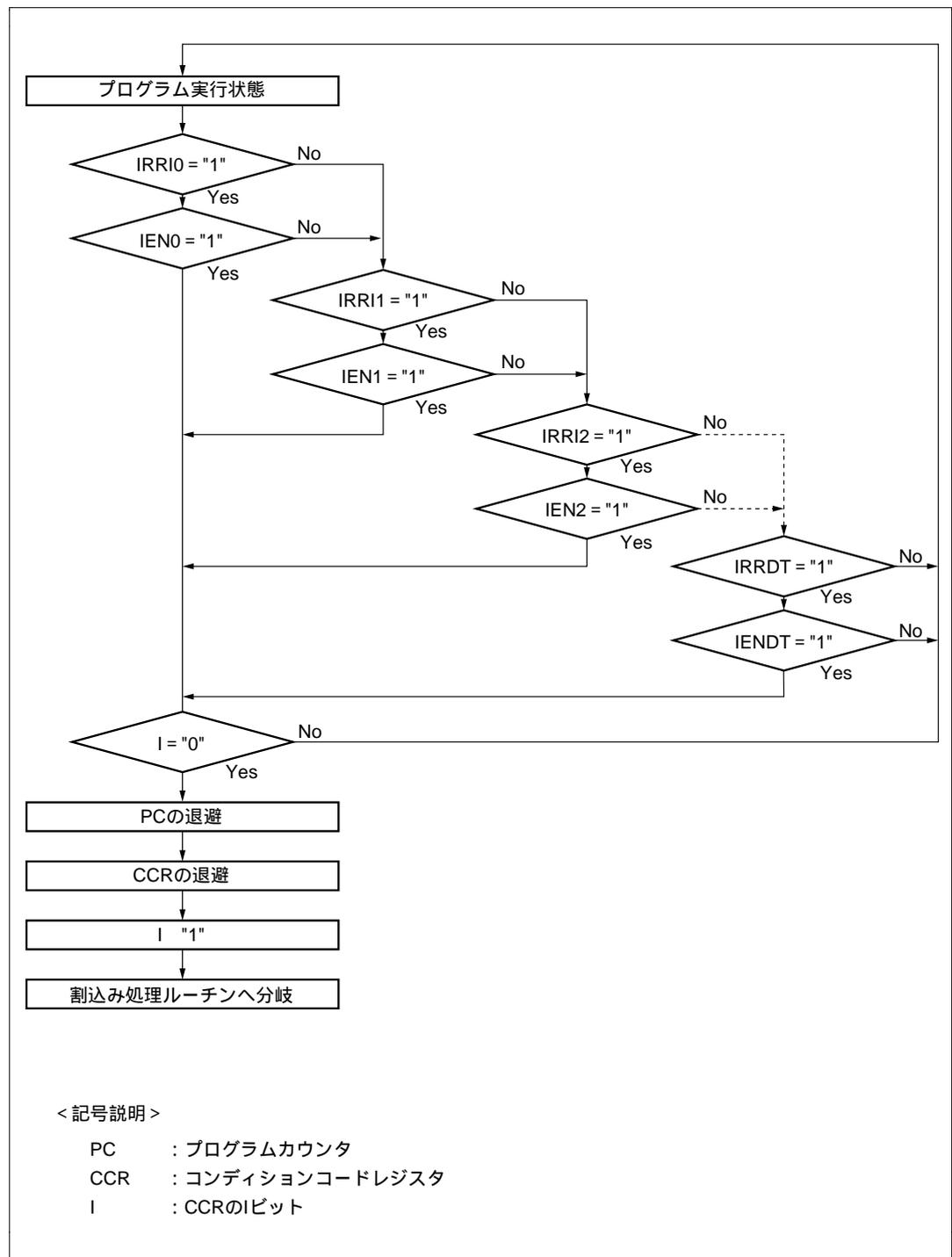


図 3.3 割り込み受け付けまでのフロー

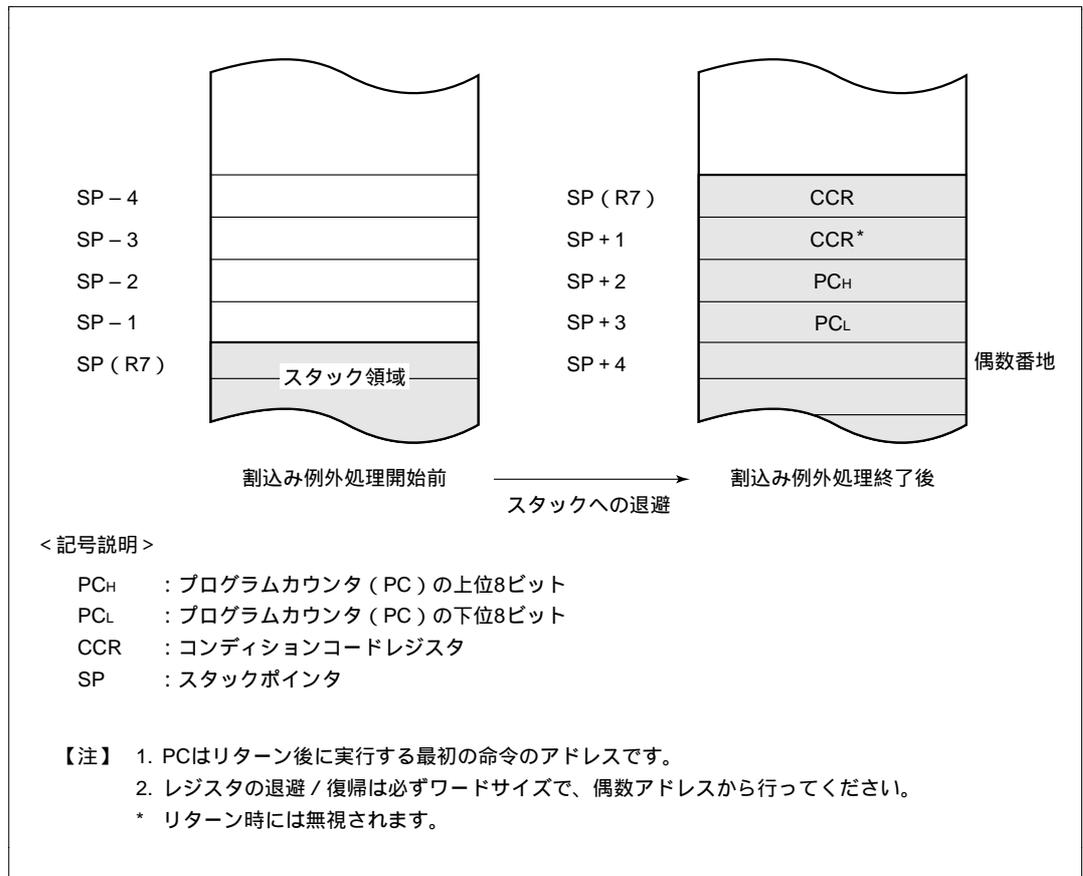


図 3.4 割り込み例外処理終了後のスタック状態

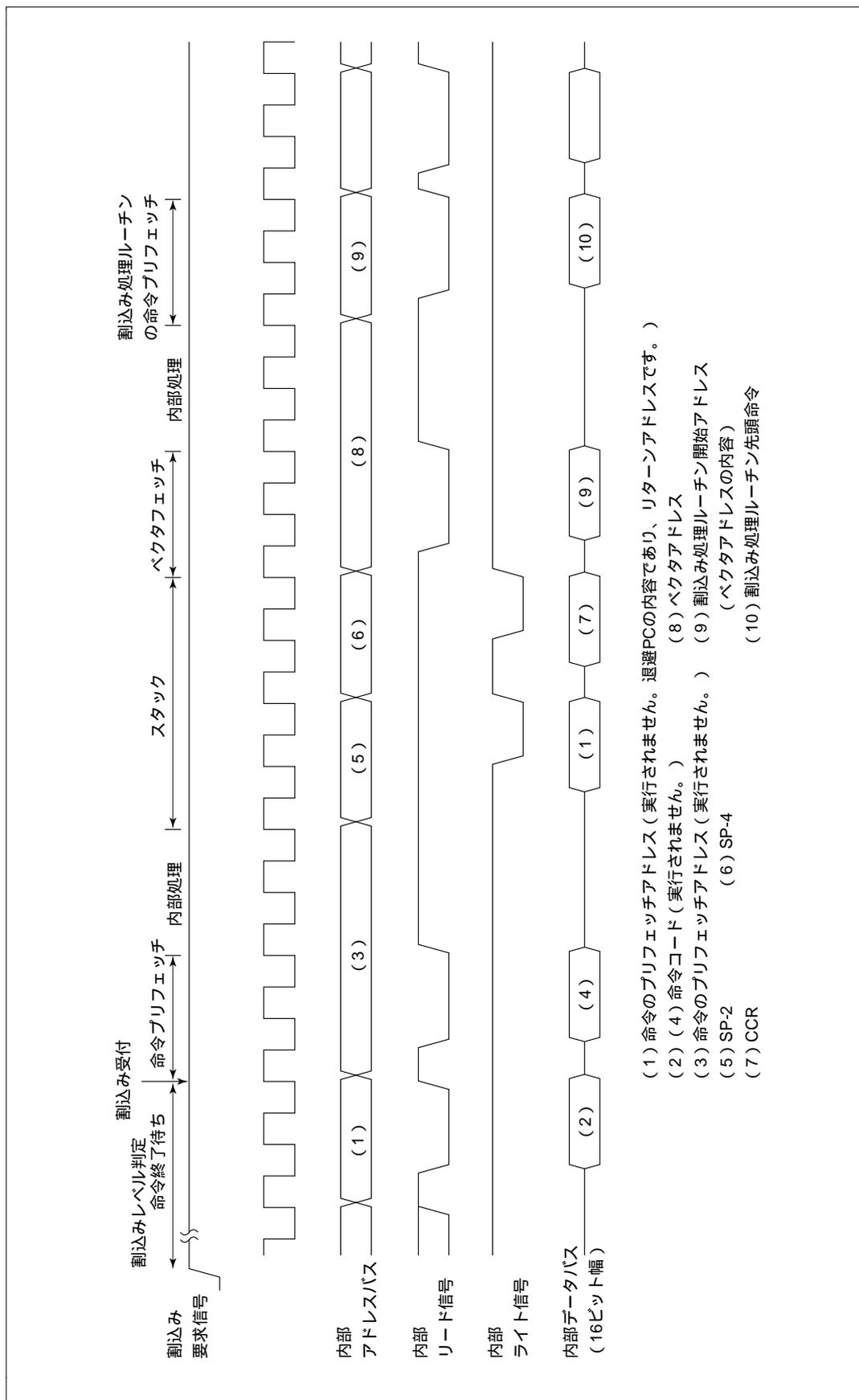


図 3.5 割り込みシーケンス

### 3.3.6 割込み応答時間

割込み要求フラグがセットされた後、割込み処理ルーチンの先頭命令を実行するまでの待ち状態数を表 3.4 に示します。

表 3.4 割込み待ち状態数

項目	状態数	合計
実行中の命令終了時の待ち時間*	1 ~ 13	15 ~ 27
PC、CCR のスタック	4	
ベクタフェッチ	2	
命令フェッチ	4	
内部処理	4	

【注】 \* EEPMOV 命令は除きます。

## 3.4 使用上の注意事項

### 3.4.1 スタック領域に関する使用上の注意事項

本 LSI では、ワードデータをアクセスする場合は、アドレスの最下位ビットは"0"とみなされます。スタック領域に対するアクセスは、常にワードサイズで行い、スタックポインタ (SP : R7) の内容は奇数にしないでください。すなわち、レジスタの退避や復帰は、「PUSH Rn (MOV.W Rn, @-SP)」または「POP Rn (MOV.W @SP+, Rn)」を使用してください。

SP に奇数を設定すると、誤動作の原因となります。SP に奇数を設定した場合の動作例を図 3.6 に示します。

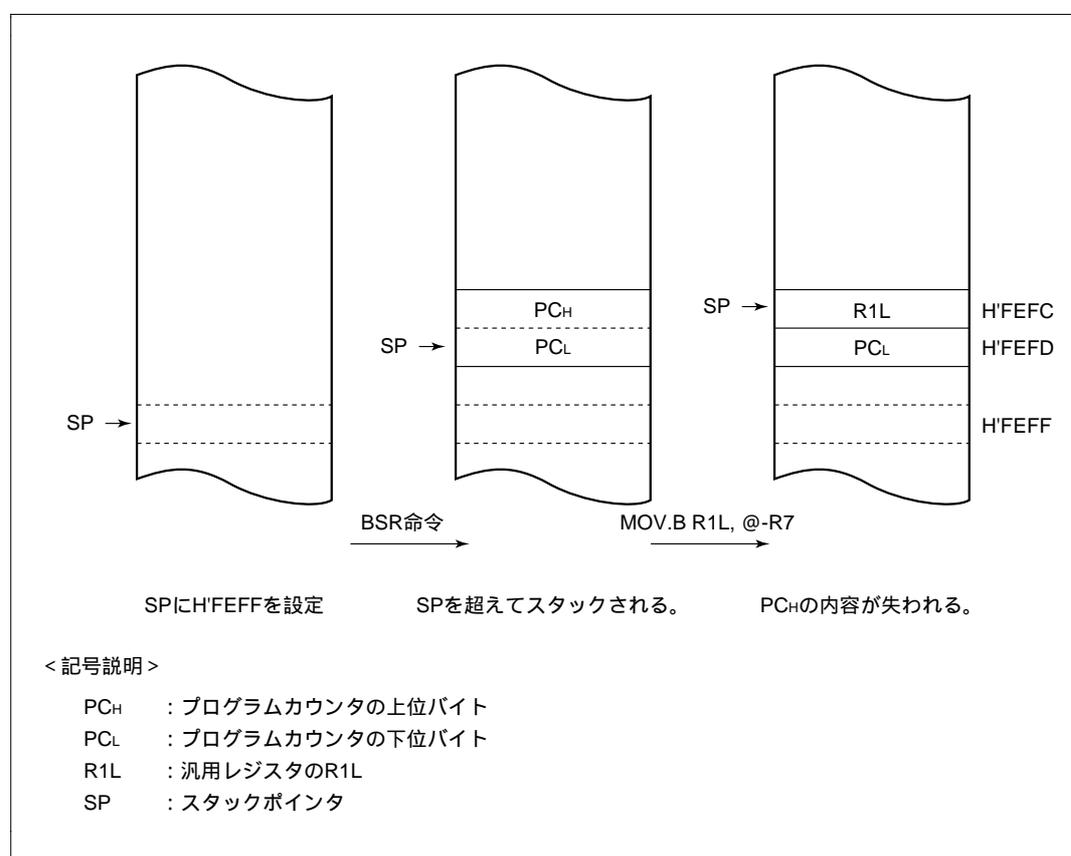


図 3.6 SP に奇数を設定したときの動作

また、割り込み例外処理および RTE 命令実行時の CCR の退避時および復帰時はワードサイズで扱われます。退避時には、ワードサイズデータの上位バイトおよび下位バイトともに CCR の値が退避されます。復帰時には、偶数アドレスの値が CCR に格納され、奇数アドレスの値は無視されます。

### 3.4.2 ポートモードレジスタを書換える際の注意事項

外部割込み端子の機能切換えのためにポートモードレジスタを書換える際には、以下の点に注意してください。

外部割込み端子 ( $\overline{\text{IRQ}}_4 \sim \overline{\text{IRQ}}_0$ ,  $\overline{\text{WKP}}_7 \sim \overline{\text{WKP}}_0$ ) を制御しているポートモードレジスタを書換えて端子機能を切換えた場合、端子に有効な割込みが入力されていなくても、端子機能を切換えた時点で割込み要求フラグが"1"にセットされますので、割込み要求フラグを"0"にクリアしてから使用してください。

"1"にセットされる割込み要求フラグとその条件を表 3.5 に示します。

表 3.5 割込み要求フラグが"1"にセットされる条件

"1"にセットされる 割込み要求フラグ		条件	
IRR1	IRRI4	<ul style="list-style-type: none"> <li>• <math>\overline{IRQ}_4</math> 端子が"Low"レベルでIEGRのIEG4が"0"の状態、PMR2のIRQ4を"0"から"1"に書換えたとき</li> <li>• <math>\overline{IRQ}_4</math> 端子が"Low"レベルでIEGRのIEG4が"1"の状態、PMR2のIRQ4を"1"から"0"に書換えたとき</li> </ul>	
	IRRI3	<ul style="list-style-type: none"> <li>• <math>\overline{IRQ}_3</math> 端子が"Low"レベルでIEGRのIEG3が"0"の状態、PMR1のIRQ3を"0"から"1"に書換えたとき</li> <li>• <math>\overline{IRQ}_3</math> 端子が"Low"レベルでIEGRのIEG3が"1"の状態、PMR1のIRQ3を"1"から"0"に書換えたとき</li> </ul>	
	IRRI2	<ul style="list-style-type: none"> <li>• <math>\overline{IRQ}_2</math> 端子が"Low"レベルでIEGRのIEG2が"0"の状態、PMR1のIRQ2を"0"から"1"に書換えたとき</li> <li>• <math>\overline{IRQ}_2</math> 端子が"Low"レベルでIEGRのIEG2が"1"の状態、PMR1のIRQ2を"1"から"0"に書換えたとき</li> </ul>	
	IRRI1	<ul style="list-style-type: none"> <li>• <math>\overline{IRQ}_1</math> 端子が"Low"レベルでIEGRのIEG1が"0"の状態、PMR1のIRQ1を"0"から"1"に書換えたとき</li> <li>• <math>\overline{IRQ}_1</math> 端子が"Low"レベルでIEGRのIEG1が"1"の状態、PMR1のIRQ1を"1"から"0"に書換えたとき</li> </ul>	
	IRRI0	IRRI0	<ul style="list-style-type: none"> <li>• <math>\overline{IRQ}_0</math> 端子が"Low"レベルでIEGRのIEG0が"0"の状態、PMR2のIRQ0を"0"から"1"に書換えたとき</li> <li>• <math>\overline{IRQ}_0</math> 端子が"Low"レベルでIEGRのIEG0が"1"の状態、PMR2のIRQ0を"1"から"0"に書換えたとき</li> </ul>
		IRRI0	<ul style="list-style-type: none"> <li>• <math>\overline{IRQ}_0</math> 端子が"Low"レベルでIEGRのIEG0が"0"の状態、PMR2のIRQ0を"0"から"1"に書換えたとき</li> <li>• <math>\overline{IRQ}_0</math> 端子が"Low"レベルでIEGRのIEG0が"1"の状態、PMR2のIRQ0を"1"から"0"に書換えたとき</li> </ul>
IWPR	IWPF7	$\overline{WKP}_7$ 端子が"Low"レベルの状態、PMR5のWKP7を"0"から"1"に書換えたとき	
	IWPF6	$\overline{WKP}_6$ 端子が"Low"レベルの状態、PMR5のWKP6を"0"から"1"に書換えたとき	
	IWPF5	$\overline{WKP}_5$ 端子が"Low"レベルの状態、PMR5のWKP5を"0"から"1"に書換えたとき	
	IWPF4	$\overline{WKP}_4$ 端子が"Low"レベルの状態、PMR5のWKP4を"0"から"1"に書換えたとき	
	IWPF3	$\overline{WKP}_3$ 端子が"Low"レベルの状態、PMR5のWKP3を"0"から"1"に書換えたとき	
	IWPF2	$\overline{WKP}_2$ 端子が"Low"レベルの状態、PMR5のWKP2を"0"から"1"に書換えたとき	
	IWPF1	$\overline{WKP}_1$ 端子が"Low"レベルの状態、PMR5のWKP1を"0"から"1"に書換えたとき	
	IWPF0	$\overline{WKP}_0$ 端子が"Low"レベルの状態、PMR5のWKP0を"0"から"1"に書換えたとき	

ポートモードレジスタの操作と割り込み要求フラグのクリアの手順を図 3.7 に示します。

端子機能を切換える場合は、ポートモードレジスタの操作前に割り込み禁止状態にして、ポートモードレジスタ操作後に、少なくとも 1 命令 (NOP 命令で可) 実行してから、"1" にセットされた割り込み要求フラグを"0"にクリアしてください。ポートモードレジスタ操作後に 1 命令実行せず割り込み要求フラグを"0"にクリアする命令を実行しても、割り込み要求フラグはクリアされませんので注意してください。

なお、端子機能切換えにともなう割り込み要求フラグのセットを回避する他の方法として、表 3.5 の条件を満たさないように端子を"High"レベルに制御して行う方法もあります。

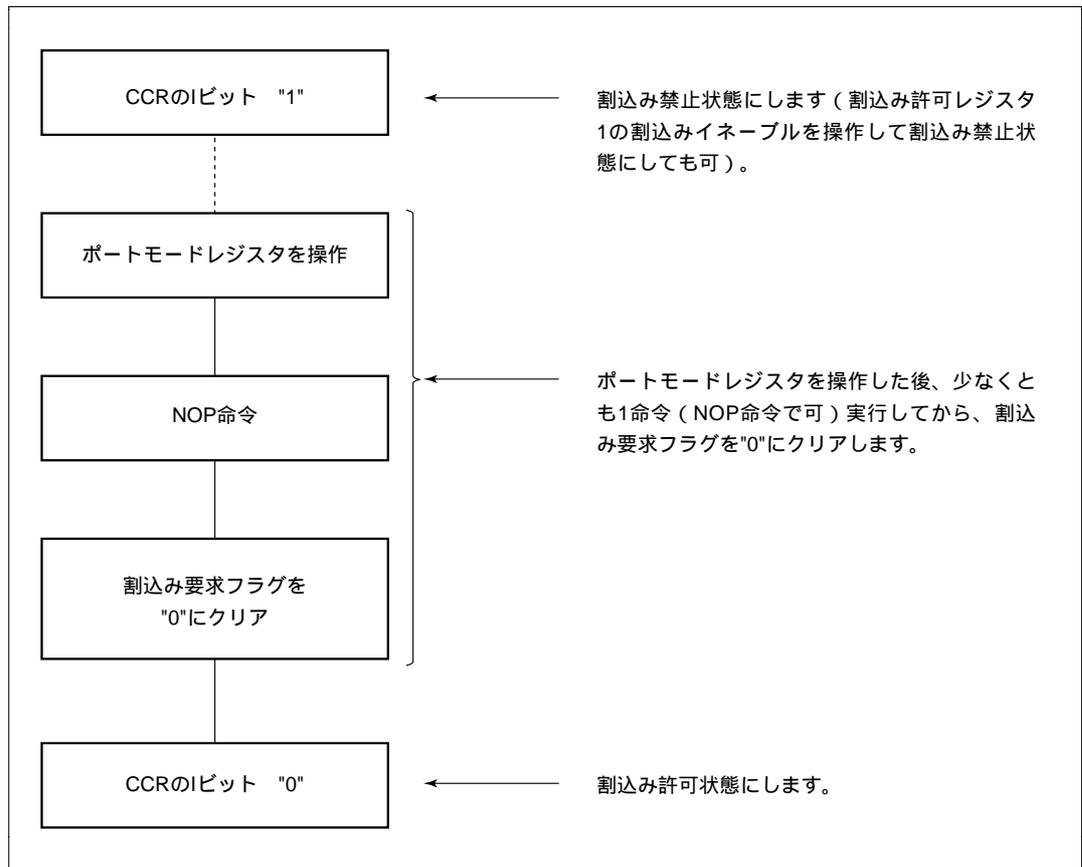


図 3.7 ポートモードレジスタ操作と割り込み要求フラグのクリア手順

---

# 4. クロック発振器

---

## 第4章 目次

4.1	概要.....	89
	4.1.1    ブロック図.....	89
	4.1.2    システムクロックとサブクロック.....	89
4.2	システムクロック発振器.....	90
4.3	サブクロック発振器.....	93
4.4	プリスケータ.....	94
4.5	発振子に関する注意事項.....	95



## 4.1 概要

本 LSI は、クロック発生回路 (CPG : Clock Pulse Generator) を内蔵しています。

クロック発生回路は、システムクロック発振器、システムクロック分周器からなるシステムクロック発生回路と、サブクロック発振器、サブクロック分周器からなるサブクロック発生回路の2つの回路から構成されます。

### 4.1.1 ブロック図

図 4.1 にクロック発生回路のブロック図を示します。

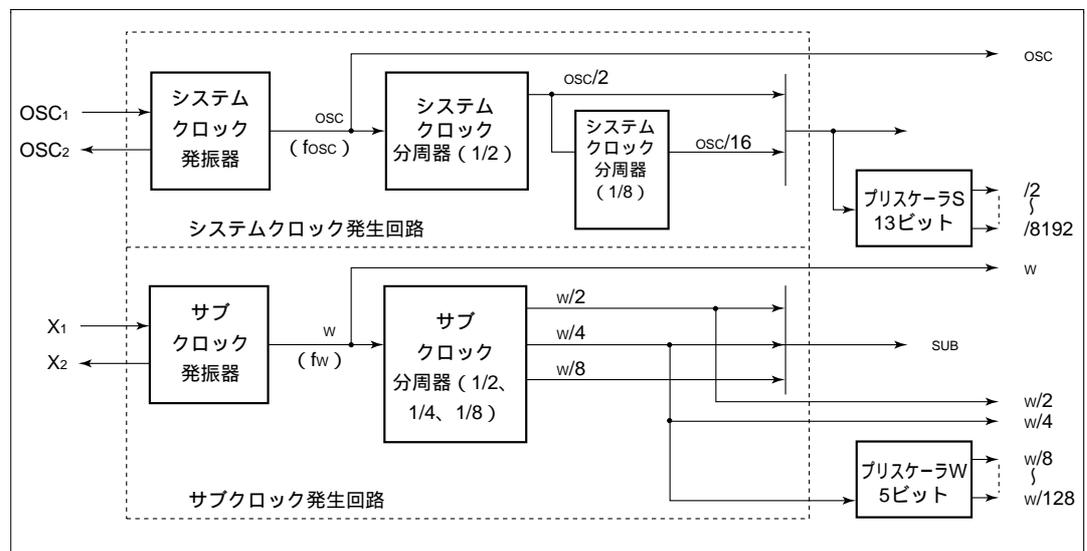


図 4.1 クロック発生回路のブロック図

### 4.1.2 システムクロックとサブクロック

クロック (  $\text{osc}$  および  $\text{SUB}$  ) は、CPU および周辺機能を動作させるための基準クロックです。

$\text{osc}$  をシステムクロック、 $\text{SUB}$  をサブクロックと呼びます。また、 $\text{osc}$  を OSC クロック、 $w$  をウォッチクロックと呼びます。

クロック  $\text{osc}$ 、 $\text{osc}/8192$ 、 $\text{osc}/4096$ 、 $\text{osc}/2048$ 、 $\text{osc}/1024$ 、 $\text{osc}/512$ 、 $\text{osc}/256$ 、 $\text{osc}/128$ 、 $\text{osc}/64$ 、 $\text{osc}/32$ 、 $\text{osc}/16$ 、 $\text{osc}/8$ 、 $\text{osc}/4$ 、 $\text{osc}/2$ 、 $w/128$ 、 $w/64$ 、 $w/32$ 、 $w/16$ 、 $w/8$ 、 $w/4$ 、 $w/2$ 、 $w$  は、周辺機能で必要とするクロックであり各周辺機能によって異なります。

## 4.2 システムクロック発振器

システムクロック分周器へクロックを供給する方法には、水晶発振子またはセラミック発振子を接続する方法と、外部クロックを入力する方法があります。

### (1) 水晶発振子を接続する方法

水晶発振子の接続例を図 4.2 に示します。

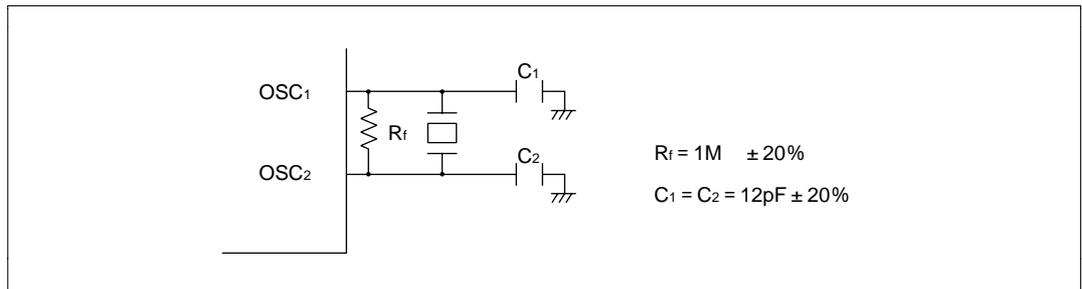


図 4.2 水晶発振子の接続例

図 4.3 に水晶発振子の等価回路を示します。発振子は表 4.1 に示す特性のものを使用してください。

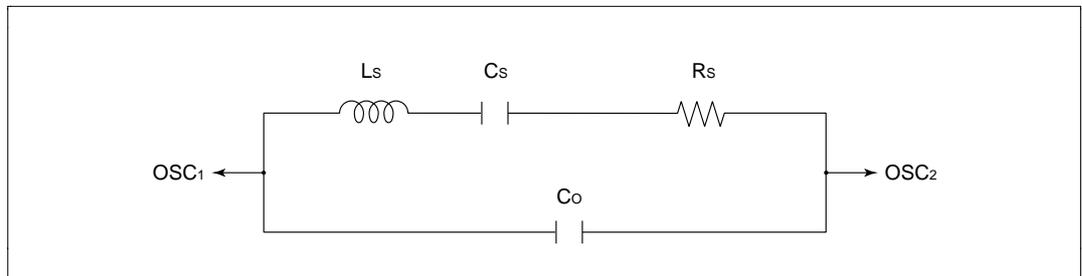


図 4.3 水晶発振子の等価回路

表 4.1 水晶発振子のパラメータ

周波数 (MHz)	2	4	8	10
$R_s$ (max)	500	100	50	30
$C_0$ (max)	7pF			

## (2) セラミック発振子を接続する方法

セラミック発振子の接続例を図 4.4 に示します。

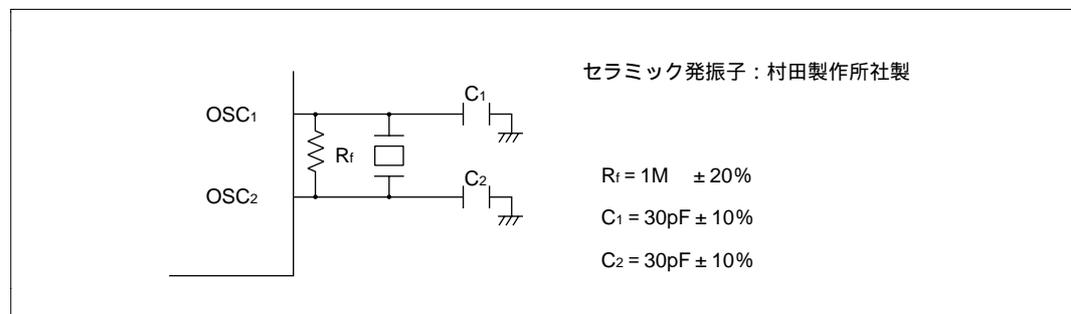


図 4.4 セラミック発振子の接続例

## (3) ボード設計上の注意

水晶発振子（セラミック発振子）を接続して発振させる場合、次の点に注意してください。

発振回路部の近くで信号線を通過させないでください。誘導により正しい発振ができなくなる場合があります（図 4.5 参照）。

また、ボード設計に際しては、発振子および負荷容量はできるだけ  $OSC_1$ 、 $OSC_2$  端子の近くに配置してください。

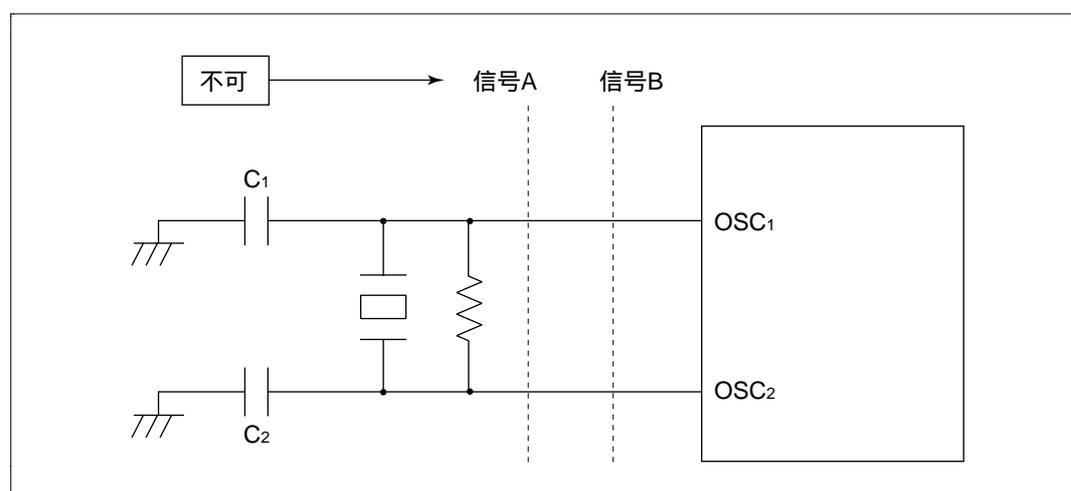


図 4.5 発振回路部のボード設計に関する注意事項

(4) 外部クロックを入力する方法

外部クロックを OSC<sub>1</sub> 端子に接続し、OSC<sub>2</sub> 端子をオープン状態にします。

この場合の接続例を図 4.6 に示します。

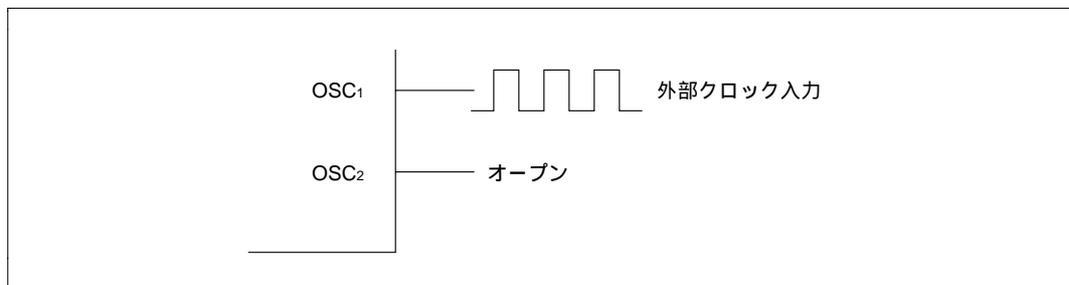


図 4.6 外部クロックを入力する場合の接続例

周波数	OSC クロック ( <sub>osc</sub> )
duty	45% ~ 55%

## 4.3 サブクロック発振器

### (1) 32.768kHz 水晶発振子を接続する方法

サブクロック分周器へクロックを供給するには、図 4.7 に示すように 32.768kHz の水晶発振子を接続します。接続する場合の注意については、「4.2(3) ボード設計上の注意」と同様です。

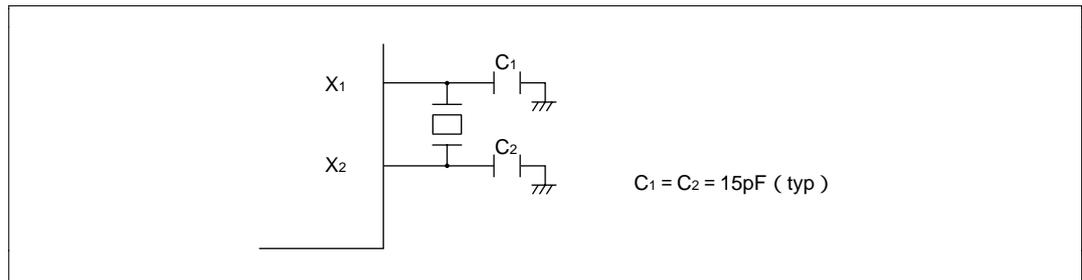


図 4.7 32.768kHz 水晶発振子の接続例

図 4.8 に 32.768kHz 水晶発振子の等価回路を示します。

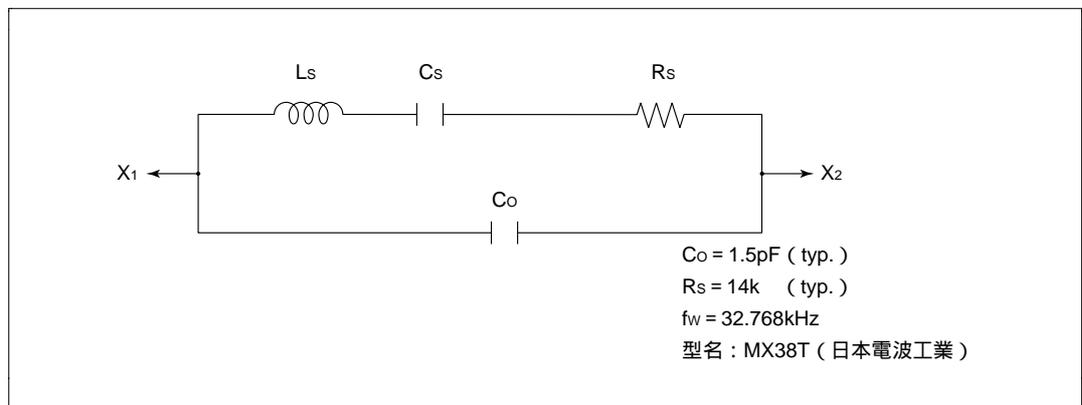


図 4.8 32.768kHz 水晶発振子の等価回路

### (2) サブクロックを必要としない場合の端子処理

サブクロックを必要としない場合には、図 4.9 に示すように X1 端子を V<sub>CC</sub> に接続し、X2 端子をオープンとしてください。

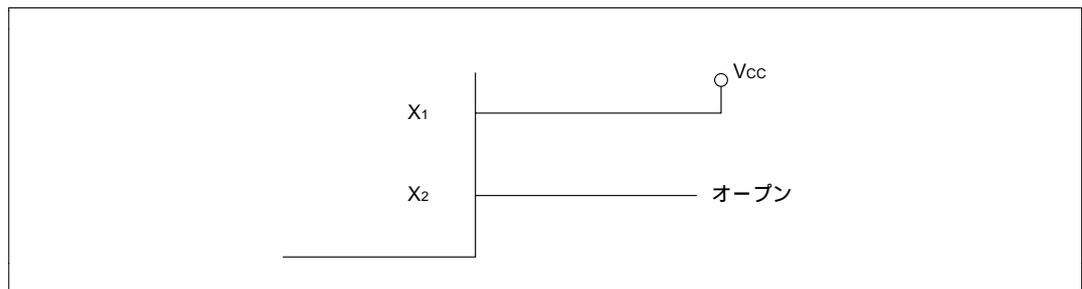


図 4.9 サブクロックを必要としない場合の端子処理

## 4.4 プリスケーラ

本 LSI は、入力クロックの異なる 2 本のプリスケーラ (プリスケーラ S、プリスケーラ W) を内蔵しています。

プリスケーラ S は、システムクロック ( ) を入力クロックとする 13 ビットのカウンタで、分周した出力を内蔵周辺モジュールの内部クロックとして使用します。プリスケーラ W は 32.768kHz を 4 分周したクロック (  $f_w/4$  ) を入力とする 5 ビットのカウンタで、分周した出力をタイマ A の時計用タイムベース動作に使用します。

### (1) プリスケーラ S (PSS)

プリスケーラ S は、システムクロック ( ) を入力クロックとする 13 ビットのカウンタで、1 サイクルごとにカウントアップします。

リセット時、プリスケーラ S は H'0000 にイニシャライズされ、リセット解除後、カウントアップを開始します。

スタンバイモード、ウォッチモード、サブアクティブモード、およびサブスリープモードでは、システムクロック発振器が停止するためプリスケーラ S の動作も停止します。このとき、プリスケーラ S は H'0000 にイニシャライズされます。

CPU によるプリスケーラ S のデータリード/ライトはできません。

プリスケーラ S の出力は、各種内蔵周辺機能で共用しており、分周比は各内蔵周辺機能で独立に設定できます。

なお、アクティブ (中速) モードではプリスケーラ S のクロック入力が  $f_{osc}/16$  となります。

### (2) プリスケーラ W (PSW)

プリスケーラ W は、32.768kHz を 4 分周したクロック (  $f_w/4$  ) を入力クロックとする 5 ビットのカウンタです。

リセット時、プリスケーラ W は H'00 にイニシャライズされ、リセット解除後、カウントアップを開始します。

スタンバイモード、ウォッチモード、サブアクティブモード、およびサブスリープモードに移行しても、 $X_1$ 、 $X_2$  端子にクロックが供給されているかぎり、プリスケーラ W は動作を継続します。

プリスケーラ W は、TMA の TMA3、TMA2 を各々 "1"、"1" に設定することでリセットできます。

また、プリスケーラ W の出力は、タイマ A のクロックとして使用できます。このとき、タイマ A は時計用タイムベースとして機能します。

## 4.5 発振子に関する注意事項

発振子に関する諸特性は、ユーザのボード設計に密接に関係しますので、本章で案内する発振子の接続例および図 4.10 水晶、セラミック発振子の配置例を参考に、マスク ROM 版、ZTAT<sup>®</sup>版共にユーザ側での十分な評価を実施してご使用願います。発振子の回路定格は発振子、実装回路の浮遊容量などにより異なるため、発振子メーカーと充分ご相談の上決定してください。発振端子に印加される電圧が最大定格を越えないような設計を行ってください。

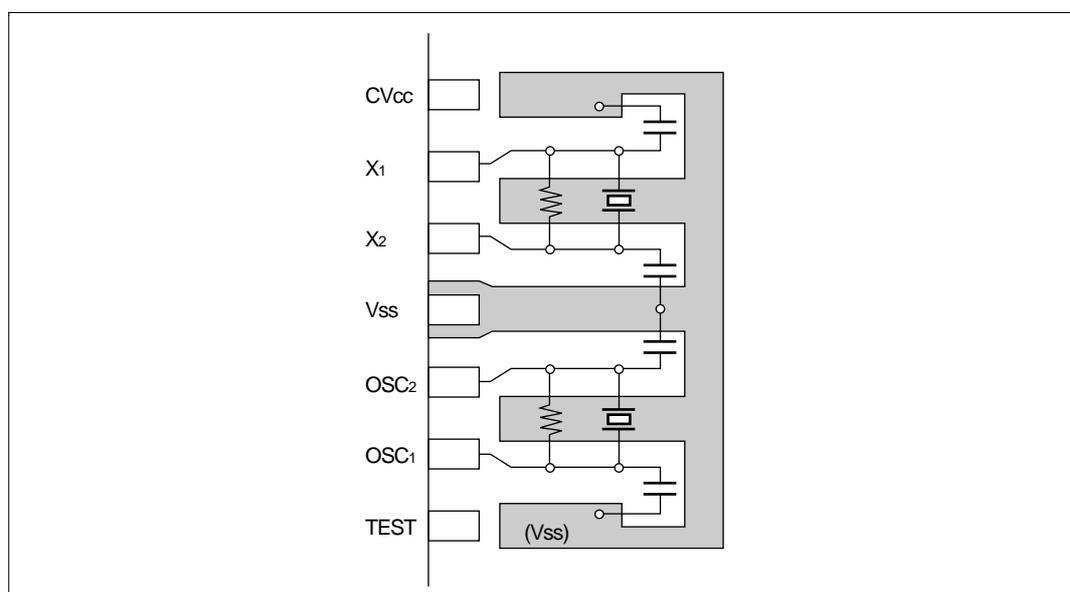


図 4.10 水晶、セラミック発振子の配置例

---

# 5. 低消費電力モード

---

## 第5章 目次

5.1	概要	99
5.1.1	システムコントロールレジスタ	102
5.2	スリープモード	106
5.2.1	スリープモードへの遷移	106
5.2.2	スリープモードの解除	106
5.3	スタンバイモード	107
5.3.1	スタンバイモードへの遷移	107
5.3.2	スタンバイモードの解除	107
5.3.3	スタンバイモード解除後の発振安定時間の設定	108
5.3.4	スタンバイモードへの遷移と端子状態	108
5.4	ウォッチモード	109
5.4.1	ウォッチモードへの遷移	109
5.4.2	ウォッチモードの解除	109
5.4.3	ウォッチモード解除後の発振安定時間の設定	109
5.5	サブスリープモード	110
5.5.1	サブスリープモードへの遷移	110
5.5.2	サブスリープモードの解除	110
5.6	サブアクティブモード	111
5.6.1	サブアクティブモードへの遷移	111
5.6.2	サブアクティブモードの解除	111
5.6.3	サブアクティブモードの動作周波数について	111
5.7	アクティブ(中速)モード	112
5.7.1	アクティブ(中速)モードへの遷移	112
5.7.2	アクティブ(中速)モードの解除	112
5.7.3	アクティブ(中速)モードの動作周波数について	112
5.8	直接遷移	113
5.8.1	直接遷移の概要	113

## 5. 低消費電力モード

---

5.8.2	直接遷移の時間 .....	114
-------	---------------	-----

## 5.1 概要

本 LSI には、リセット解除後に LSI が動作するモードとして、消費電力を著しく低下させる 6 種類の低消費電力モードを含む、7 種類の動作モードをもっています。

表 5.1 に動作モードの概要を示します。

表 5.1 動作モードの概要

動作モード	説明
アクティブ(高速)モード	CPU がシステムクロックにより、高速動作でプログラムを実行しているモードです。
アクティブ(中速)モード	CPU がシステムクロックにより、低速動作でプログラムを実行しているモードです。
サブアクティブモード	CPU がサブクロックにより、低速動作でプログラムを実行しているモードです。
スリープモード	CPU が動作を停止し、内蔵周辺機能がシステムクロックで動作しているモードです。
サブスリープモード	CPU が動作を停止し、タイマ A、およびタイマ G がサブクロックで動作しているモードです。
ウォッチモード	CPU が動作を停止し、タイマ A の時計機能がサブクロックで動作しているモードです。
スタンバイモード	CPU およびすべての内蔵の周辺機能が動作を停止しているモードです。

上記 7 種類の動作モードのうち、アクティブ(高速)モード以外の動作モードを低消費電力モードと呼びます。また、本章では、アクティブ(高速)モードとアクティブ(中速)モードを総称してアクティブモードと呼びます。

図 5.1 にモード遷移図を示します。

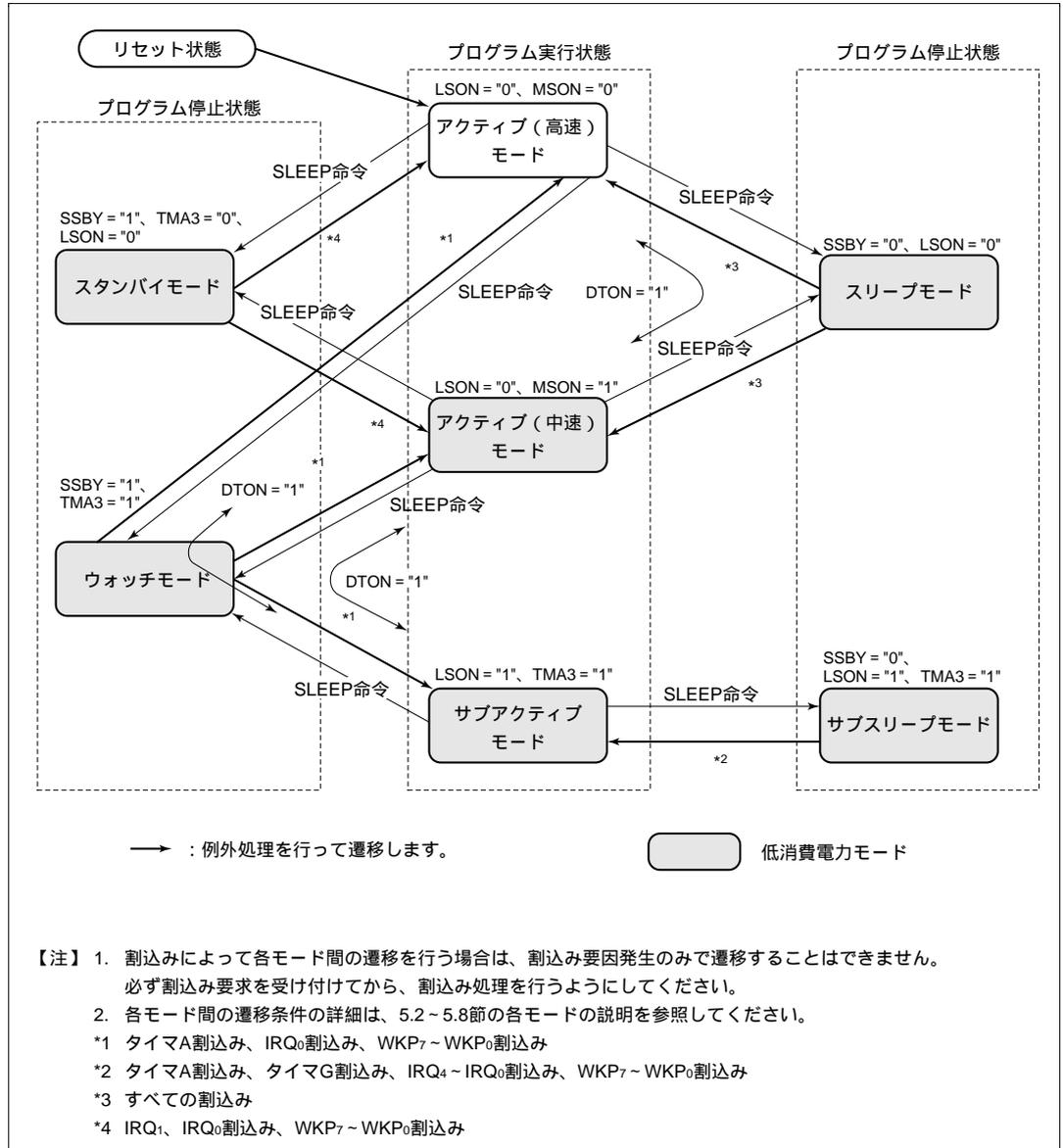


図 5.1 モード遷移図

表 5.2 に各モードでの LSI の内部状態を示します。

表 5.2 各動作モードでの LSI の状態

機能		アクティブ		スリープ	ウォッチ	サブアクティブ	サブスリープ	スタンバイ	
		高速	中速						
システムクロック発振器		動作	動作	動作	停止	停止	停止	停止	
サブクロック発振器		動作	動作	動作	動作	動作	動作	動作	
CPU 動作	命令	動作	動作	停止	停止	動作	停止	停止	
	RAM			保持	保持		保持	保持	保持
	レジスタ								保持 <sup>*1</sup>
	I/O								
外部 割込み の動作	IRQ <sub>0</sub>	動作	動作	動作	動作	動作	動作	動作	
	IRQ <sub>1</sub>				保持 <sup>*4</sup>			保持 <sup>*4</sup>	
	IRQ <sub>2</sub>								
	IRQ <sub>3</sub>								
	IRQ <sub>4</sub>								
	WKP <sub>0</sub>	動作	動作	動作	動作	動作	動作	動作	
	WKP <sub>1</sub>								
	WKP <sub>2</sub>								
	WKP <sub>3</sub>								
	WKP <sub>4</sub>								
WKP <sub>5</sub>									
WKP <sub>6</sub>									
WKP <sub>7</sub>									
周辺機能 の動作	タイマ A	動作	動作	動作	動作 <sup>*3</sup>	動作 <sup>*3</sup>	動作 <sup>*3</sup>	保持	
	タイマ F				保持	保持	保持		
	タイマ G				動作 / 保持 <sup>*2</sup>		動作 / 保持 <sup>*2</sup>		
	SCI1	動作	動作	動作	保持	保持	保持	保持	
	SCI3				リセット	リセット	リセット	リセット	
	DTMF	動作	動作	動作	リセット	リセット	リセット	リセット	
	A/D	動作	動作	動作	保持	保持	保持	保持	

【注】 \*1 レジスタは保持、出力はハイインピーダンス

\*2 外部クロックとして  $v_{DD}/2$  を選択した場合に動作、その他は停止して保持

\*3 時計用タイムベース機能を選択時に動作

\*4 外部割込み要求は無視されます。割込み要求レジスタの内容は影響を受けません。

## 5.1.1 システムコントロールレジスタ

表 5.3 に動作モードを設定するシステムコントロールレジスタを示します。

表 5.3 レジスタ構成

名称	略称	R/W	初期値	アドレス
システムコントロールレジスタ 1	SYSCR1	R/W	H'07	H'FFF0
システムコントロールレジスタ 2	SYSCR2	R/W	H'E0	H'FFF1

## (1) システムコントロールレジスタ 1 (SYSCR1)

ビット:	7	6	5	4	3	2	1	0
	SSBY	STS2	STS1	STS0	LSON	—	—	—
初期値:	0	0	0	0	0	1	1	1
R/W :	R/W	R/W	R/W	R/W	R/W	—	—	—

SYSCR1 は、8 ビットのリード/ライト可能なレジスタで、低消費電力モードの制御を行います。

SYSCR1 は、リセット時 H'07 にイニシャライズされます。

## ビット 7: ソフトウェアスタンバイ (SSBY)

スタンバイモード、ウォッチモードへの遷移を指定します。

ビット 7	説明
SSBY	
0	アクティブモードで SLEEP 命令実行後、スリープモードに遷移 サブアクティブモードで SLEEP 命令実行後、サブスリープモードに遷移 (初期値)
1	アクティブモードで SLEEP 命令実行後、スタンバイモードあるいはウォッチモードに遷移 サブアクティブモードで SLEEP 命令実行後、ウォッチモードに遷移

## ビット6～4：スタンバイタイムセレクト2～0 (STS2～STS0)

特定の割込みにより、スタンバイモード、ウォッチモードを解除し、アクティブモードに遷移する場合に、クロックが安定するまで CPU と周辺機能が待機する時間を指定します。動作周波数に応じて待機時間が 10ms 以上となるように指定してください。

ビット6	ビット5	ビット4	説明
STS2	STS1	STS0	
0	0	0	待機時間 = 8,192 ステート (初期値)
0	0	1	待機時間 = 16,384 ステート
0	1	0	待機時間 = 32,768 ステート
0	1	1	待機時間 = 65,536 ステート
1	*	*	待機時間 = 131,072 ステート

\* Don't care

## ビット3：ロースピードオンフラグ (LSON)

ウォッチモードを解除時に、CPUの動作クロックをシステムクロック( )にするか、サブクロック( <sub>SUB</sub> )にするかを選択します。他の制御ビット、割込み入力の組合せで動作モードを決定します。

ビット3	説明
LSON	
0	CPUの動作クロックはシステムクロック( ) (初期値)
1	CPUの動作クロックはサブクロック( <sub>SUB</sub> )

## ビット2～0：リザーブビット

リザーブビットです。各ビットはリードすると常に"1"が読み出されます。ライトは無効です。

## (2) システムコントロールレジスタ 2 (SYSCR2)

ビット:	7	6	5	4	3	2	1	0
	—	—	—	NESEL	DTON	MSON	SA1	SA0
初期値:	1	1	1	0	0	0	0	0
R/W :	—	—	—	R/W	R/W	R/W	R/W	R/W

SYSCR2 は、8 ビットのリード/ライト可能なレジスタで、低消費電力モードの制御を行います。

SYSCR2 は、リセット時 H'E0 にイニシャライズされます。

## ビット 7~5 : リザーブビット

リザーブビットです。各ビットはリードすると常に"1"が読み出されます。ライトは無効です。

## ビット 4 : ノイズ除去サンプリング周波数選択 (NESEL)

サブクロック発振器より生成されたウォッチクロック ( $\omega_w$ ) を、システムクロック発振器より生成された OSC クロック ( $\omega_{osc}$ ) により、サンプリングする周波数を選択します。 $\omega_{osc} = 2 \sim 10\text{MHz}$  のときは、"0"をセットしてください。

ビット 4	説明
NESEL	
0	$\omega_{osc}$ の 16 分周クロックでサンプリング (初期値)
1	$\omega_{osc}$ の 4 分周クロックでサンプリング

## ビット3：ダイレクトトランスファオンフラグ (DTON)

アクティブ(高速)モード、アクティブ(中速)モード、サブアクティブモードの各モード間を、SLEEP 命令を実行することにより直接遷移するか否かを指定します。SLEEP 命令実行後に遷移する動作モードは、本ビット以外の制御ビットの組合せで決定します。

ビット3	
DTON	説明
0	<ul style="list-style-type: none"> <li>・アクティブモードで SLEEP 命令を実行したとき、スタンバイモード、ウォッチモード、またはスリープモードに遷移</li> <li>・サブアクティブモードで SLEEP 命令を実行したとき、ウォッチモード、またはサブスリープモードに遷移 (初期値)</li> </ul>
1	<ul style="list-style-type: none"> <li>・アクティブ(高速)モードで SLEEP 命令を実行したとき、アクティブ(中速)モード (SSBY="0"、MSON="1"、LSON="0"のとき)、またはサブアクティブモード (SSBY="1"、TMA3="1"、LSON="1"のとき) に直接遷移</li> <li>・アクティブ(中速)モードで SLEEP 命令を実行したとき、アクティブ(高速)モード (SSBY="0"、MSON="0"、LSON="0"のとき)、またはサブアクティブモード (SSBY="1"、TMA3="1"、LSON="1"のとき) に直接遷移</li> <li>・サブアクティブモードで SLEEP 命令を実行したとき、アクティブ(高速)モード (SSBY="1"、TMA3="1"、LSON="0"、MSON="0"のとき)、またはアクティブ(中速)モード (SSBY="1"、TMA3="1"、LSON="0"、MSON="1"のとき) に直接遷移</li> </ul>

## ビット2：ミドルスピードオンフラグ (MSON)

スタンバイモード、ウォッチモード、スリープモード解除後、アクティブ(高速)モードで動作させるか、アクティブ(中速)モードで動作させるかを選択します。

ビット2	
MSON	説明
0	アクティブ(高速)モードで動作 (初期値)
1	アクティブ(中速)モードで動作

## ビット1、0：サブアクティブモードクロックセレクト (SA1、SA0)

サブアクティブモードのCPUの動作クロック ( $w/8$ 、 $w/4$ 、 $w/2$ )を選択します。SA1、SA0はサブアクティブモードでライトしても値は更新されません。

ビット1	ビット0	
SA1	SA0	説明
0	0	$w/8$ (初期値)
0	1	$w/4$
1	*	$w/2$

\* Don't care

## 5.2 スリープモード

### 5.2.1 スリープモードへの遷移

アクティブモードで、SYSCR1 の SSBY が"0"、LSON が"0"のとき SLEEP 命令を実行すると、スリープモードに遷移します。スリープモードでは CPU の動作は停止しますが、内蔵周辺モジュールは動作します。なお、CPU のレジスタの内容は保持されます。

### 5.2.2 スリープモードの解除

スリープモードの解除は、すべての割込み(タイマ A、タイマ F、タイマ G、IRQ<sub>4</sub>~IRQ<sub>0</sub>、WKP<sub>7</sub>~WKP<sub>0</sub>、SCI3、SCI1、A/D 変換器)、リセットによって行われます。

#### (1) 割込みによる解除

割込み要求が発生すると、スリープモードは解除され、割込み例外処理を開始します。SYSCR2 の MSON が"0"のときアクティブ(高速)モードに、MSON が"1"のときアクティブ(中速)モードに遷移します。なお、CCR の I ビットが"1"のとき、あるいは割込み許可レジスタにより当該割込みの受付が禁止されている場合はスリープ状態は解除されません。

#### (2) リセットによる解除

$\overline{\text{RES}}$  端子を"Low"レベルにすることによりリセット状態に遷移し、スリープモードは解除されます。

## 5.3 スタンバイモード

### 5.3.1 スタンバイモードへの遷移

アクティブモードで SYSCR1 の SSBY が "1"、LSON が "0"、および TMA の TMA3 が "0" のとき SLEEP 命令を実行すると、スタンバイモードに遷移します。スタンバイモードではクロック発生回路からのクロック供給を停止するため、CPU および内蔵周辺機能が停止します。規定の電圧が与えられている限り、CPU のレジスタと一部の内蔵周辺機能の内部レジスタ、および内蔵 RAM のデータは保持されています。さらに、RAM データ保持電圧で規定した電圧が与えられているかぎり、内蔵 RAM のデータは保持されています。このとき、I/O ポートはハイインピーダンス状態となります。

### 5.3.2 スタンバイモードの解除

スタンバイモードの解除は、割込み (IRQ<sub>1</sub>、IRQ<sub>0</sub>、WKP<sub>7</sub>~WKP<sub>0</sub>)、 $\overline{\text{RES}}$  端子入力によって行われます。

#### (1) 割込みによる解除

割込み要求が発生すると、システムクロックの発振が開始され、SYSCR1 の STS2~STS0 により設定された時間が経過した後、安定したシステムクロックが LSI 全体に供給されて、スタンバイモードは解除され、割込み例外処理を開始します。SYSCR2 の MSON が "0" のときはアクティブ (高速) モードに、"1" のときはアクティブ (中速) モードに遷移します。なお、CCR の I ビットが "1" のとき、あるいは、割込み許可レジスタにより当該割込みの受け付けが禁止されている場合は、スタンバイモードは解除されません。

#### (2) $\overline{\text{RES}}$ 端子による解除

$\overline{\text{RES}}$  端子を "Low" レベルにすると、システムクロックの発振が開始されます。発振安定時間経過後、 $\overline{\text{RES}}$  端子を "High" レベルにすると、CPU はリセット例外処理を開始します。なお、システムクロックの発振開始と同時に LSI 全体にシステムクロックが供給されます。 $\overline{\text{RES}}$  端子は、必ずシステムクロックの発振が安定するまで、"Low" レベルを保持してください。

### 5.3.3 スタンバイモード解除後の発振安定時間の設定

SYSCR1 の STS2 ~ STS0 の設定は、以下のようにしてください。

#### (1) 水晶発振の場合

表 5.4 に動作周波数と STS2 ~ STS0 の設定値に対する待機時間を示します。待機時間が 10ms 以上となるように STS2 ~ STS0 を設定してください。

表 5.4 動作周波数と発振安定時間

( 単位 : ms )

STS2	STS1	STS0	待機時間	5MHz	4MHz	2MHz	1MHz	0.5MHz
0	0	0	8,192 ステート	1.6	2.0	4.1	8.2	16.4
0	0	1	16,384 ステート	3.2	4.1	8.2	16.4	32.8
0	1	0	32,768 ステート	6.6	8.2	16.4	32.8	65.5
0	1	1	65,536 ステート	13.1	16.4	32.8	65.5	131.1
1	*	*	131,072 ステート	26.2	32.8	65.5	131.1	262.1

\* : Don't care

#### (2) 外部クロックの場合

任意の値を選択可能です。通常の場合は、最小時間 ( STS2 = STS1 = STS0 = "0" ) の使用を推奨します。

### 5.3.4 スタンバイモードへの遷移と端子状態

アクティブ ( 高速 ) モードまたはアクティブ ( 中速 ) モードで SYSCR1 の SSBY を "1"、LSON を "0"、TMA の TMA3 を "0" にセットした状態で SLEEP 命令を実行するとスタンバイモードに遷移します。同時に端子はハイインピーダンス状態 ( プルアップ MOS オン設定端子は除く ) になります。この時のタイミングを図 5.2 に示します。

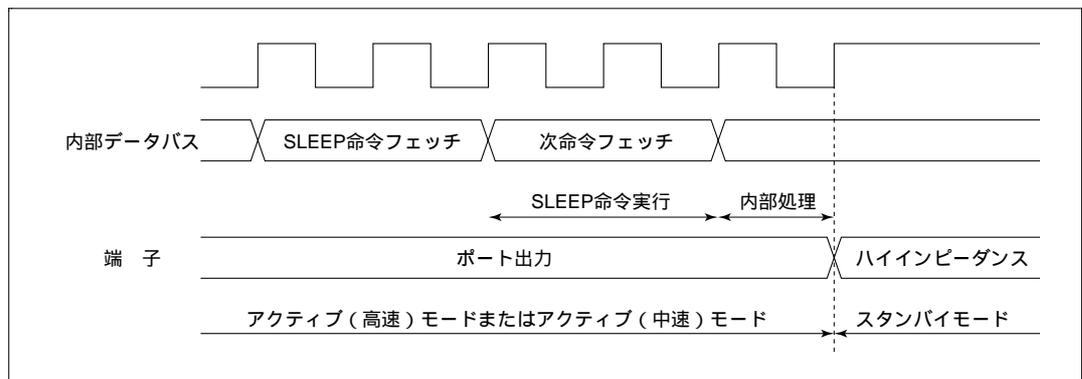


図 5.2 スタンバイモードへの遷移と端子状態

## 5.4 ウォッチモード

### 5.4.1 ウォッチモードへの遷移

アクティブモード、サブアクティブモードで SYSCR1 の SSBY が "1"、TMA の TMA3 が "1" のとき SLEEP 命令を実行すると、ウォッチモードに遷移します。ウォッチモードではタイマ A 以外の内蔵周辺機能は動作を停止します。規定の電圧が与えられている限り、CPU と一部の内蔵周辺機能の内部レジスタ\*、および内蔵 RAM の内容は保持され、I/O ポートは遷移前の状態を保持します。

【注】 \* SCI3、DTMF 発生回路のレジスタの内容はリセットされます。

### 5.4.2 ウォッチモードの解除

ウォッチモードの解除は、割込み (IRQ<sub>0</sub>、WKP<sub>7</sub>~WKP<sub>0</sub>、タイマ A)、 $\overline{\text{RES}}$  端子入力によって行われます。

#### (1) 割込みによる解除

割込み要求が発生するとウォッチモードは解除され、SYSCR1 の LSON と SYSCR2 の MSON の組合せで、LSON = "0"かつ MSON = "0"のときはアクティブ (高速) モードに、LSON = "0"かつ MSON = "1"のときはアクティブ (中速) モードに、LSON = "1"のときはサブアクティブモードに遷移します。アクティブモードに遷移するときは、SYSCR1 の STS2~STS0 により設定された時間が経過した後、安定したクロックが LSI 全体に供給され、割込み例外処理を開始します。なお、CCR の I ビットが "1" の場合、あるいは割込み許可レジスタにより当該割込みの受け付けが禁止されている場合は、ウォッチモードは解除されません。

#### (2) $\overline{\text{RES}}$ 端子による解除

$\overline{\text{RES}}$  端子による解除については、「5.3.2 スタンバイモードの解除 (2)  $\overline{\text{RES}}$  端子による解除」を参照してください。

### 5.4.3 ウォッチモード解除後の発振安定時間の設定

ウォッチモード解除後の発振安定時間の設定については、「5.3.3 スタンバイモード解除後の発振安定時間の設定」を参照してください。

## 5.5 サブスリープモード

### 5.5.1 サブスリープモードへの遷移

サブアクティブモードで SYSCR1 の SSBY が"0"、LSON が"1"、TMA の TMA3 が"1"のとき SLEEP 命令を実行すると、サブスリープモードに遷移します。サブスリープモードでは、タイマ A、タイマ G 以外の内蔵周辺機能は動作を停止します。規定の電圧が与えられている限り、CPU と一部の内蔵周辺機能の内部レジスタ\*、内蔵 RAM の内容は保持され、I/O ポートは遷移前の状態を保持します。

【注】\* SCI3、DTMF 発生回路のレジスタの内容はリセットされます。

### 5.5.2 サブスリープモードの解除

サブスリープモードの解除は、割込み（タイマ A、タイマ G、IRQ<sub>4</sub>~IRQ<sub>0</sub>、WKP<sub>7</sub>~WKP<sub>0</sub>）、 $\overline{\text{RES}}$  端子入力によって行われます。

#### (1) 割込みによる解除

割込み要求が発生するとサブスリープモードは解除され、割込み例外処理を開始します。なお、CCR の I ビットが"1"の場合、あるいは割込み許可レジスタにより当該割込みの受け付けが禁止されている場合は、サブスリープモードは解除されません。

#### (2) $\overline{\text{RES}}$ 端子による解除

$\overline{\text{RES}}$  端子による解除については、「5.3.2 スタンバイモードの解除 (2)  $\overline{\text{RES}}$  端子による解除」を参照してください。

## 5.6 サブアクティブモード

### 5.6.1 サブアクティブモードへの遷移

ウォッチモードで割込み（タイマ A、IRQ<sub>0</sub>、WKP<sub>7</sub>～WKP<sub>0</sub>）が発生したとき、SYSCR1 の LSON が"1"ならば、サブアクティブモードに遷移します。また、サブスリープモードで割込み（タイマ A、タイマ G、IRQ<sub>4</sub>～IRQ<sub>0</sub>、WKP<sub>7</sub>～WKP<sub>0</sub>）が発生したとき、サブアクティブモードに遷移します。なお、CCR の I ビットが"1"の場合、または割込み許可レジスタにより当該割込みの受け付けが禁止されている場合は、サブアクティブモードに遷移しません。また、SCI3、DTMF 発生回路のレジスタの内容はリセットされます。

### 5.6.2 サブアクティブモードの解除

サブアクティブモードの解除は、SLEEP 命令または  $\overline{\text{RES}}$  端子入力により行われます。

#### (1) SLEEP 命令による解除

SYSCR1 の SSBY が"1"、TMA の TMA3 が"1"の状態では SLEEP 命令を実行すると、サブアクティブモードは解除され、ウォッチモードに遷移します。また、SYSCR1 の SSBY が"0"、LSON が"1"、TMA の TMA3 が"1"の状態では SLEEP 命令を実行すると、サブスリープモードに遷移します。また、直接遷移によってアクティブモードへ遷移します。

直接遷移の詳細は「5.8 直接遷移」を参照してください。

#### (2) $\overline{\text{RES}}$ 端子による解除

$\overline{\text{RES}}$  端子による解除については、「5.3.2 スタンバイモードの解除 (2)  $\overline{\text{RES}}$  端子による解除」を参照してください。

### 5.6.3 サブアクティブモードの動作周波数について

サブアクティブモードの動作周波数は、SYSCR2 の SA1、SA0 により、ウォッチクロック ( $f_w$ ) の 2 分周、4 分周、8 分周から選択できます。

## 5.7 アクティブ（中速）モード

### 5.7.1 アクティブ（中速）モードへの遷移

スタンバイモードで割込み（IRQ<sub>1</sub>、IRQ<sub>0</sub>、WKP<sub>7</sub>～WKP<sub>0</sub>）が発生したとき、ウォッチモードで割込み（タイマA、IRQ<sub>0</sub>、WKP<sub>7</sub>～WKP<sub>0</sub>）が発生したとき、あるいはスリープモードですべての割込みが発生したとき、SYSCR1のLSONが"0"かつSYSCR2のMSONが"1"ならば、アクティブ（中速）モードに遷移します。なお、CCRのIビットが"1"の場合、または割込み許可レジスタにより当該割込みの受け付けが禁止されている場合は、アクティブ（中速）モードに遷移しません。

### 5.7.2 アクティブ（中速）モードの解除

アクティブ（中速）モードの解除は、SLEEP命令またはリセットにより行われます。

#### （1）SLEEP命令による解除

SYSCR1のSSBYが"1"、LSONが"0"、TMAのTMA3が"0"の状態ではSLEEP命令を実行すると、スタンバイモードに遷移します。SYSCR1のSSBYが"1"、TMAのTMA3が"1"の状態ではSLEEP命令を実行すると、ウォッチモードに遷移します。

SYSCR1のSSBYが"0"、LSONが"0"の状態ではSLEEP命令を実行すると、スリープモードに遷移します。直接遷移によってアクティブ（高速）モード、またはサブアクティブモードへ遷移します。

直接遷移の詳細は「5.8 直接遷移」を参照してください。

#### （2）リセットによる解除

$\overline{\text{RES}}$ 端子を"Low"レベルにすることによりリセット状態に遷移し、アクティブ（中速）モードは解除されます。

### 5.7.3 アクティブ（中速）モードの動作周波数について

アクティブ（中速）モードは、アクティブ（高速）モードの1/8の動作周波数のクロックによって動作します。ただし、DTMF発生回路は、OSCクロック（ $\text{osc}$ ）のまま動作します。

## 5.8 直接遷移

### 5.8.1 直接遷移の概要

CPUがプログラムを実行している動作モードにはアクティブ(高速)モード、アクティブ(中速)モード、サブアクティブモードの3つのモードがあります。この3つの動作モードの間で、プログラムを停止することなく遷移することを直接遷移と呼びます。直接遷移はSYSCR2のDTONを"1"にセットし、SLEEP命令を実行することにより可能です。遷移後は直接遷移割込み例外処理を開始します。なお、割込み許可レジスタ2(IENR2)により直接遷移割込みが禁止されている場合は、スリープモードまたはウォッチモードへ遷移します。また、CCRのIビットを"1"の状態で行うとスリープモードまたはウォッチモードに遷移し、遷移後のモードから割込みによる解除は不可能となりますので注意してください。

#### (1) アクティブ(高速)モードからアクティブ(中速)モードへの直接遷移

アクティブ(高速)モードでSYSCR1のSSBYを"0"、LSONを"0"、SYSCR2のMSONを"1"、DTONを"1"にセットした状態でSLEEP命令を実行すると、スリープモードを経由してアクティブ(中速)モードに遷移します。

#### (2) アクティブ(中速)モードからアクティブ(高速)モードへの直接遷移

アクティブ(中速)モードでSYSCR1のSSBYを"0"、LSONを"0"、SYSCR2のMSONを"0"、DTONを"1"にセットした状態でSLEEP命令を実行すると、スリープモードを経由してアクティブ(高速)モードに遷移します。

#### (3) アクティブ(高速)モードからサブアクティブモードへの直接遷移

アクティブ(高速)モードでSYSCR1のSSBYを"1"、LSONを"1"、SYSCR2のDTONを"1"、TMAのTMA3を"1"にセットした状態でSLEEP命令を実行すると、ウォッチモードを経由してサブアクティブモードに遷移します。

#### (4) サブアクティブモードからアクティブ(高速)モードへの直接遷移

サブアクティブモードでSYSCR1のSSBYを"1"、LSONを"0"、SYSCR2のMSONを"0"、DTONを"1"、TMAのTMA3を"1"にセットした状態でSLEEP命令を実行すると、ウォッチモードを経由し、SYSCR1のSTS2~STS0により設定された時間を経過した後、直接、アクティブ(高速)モードに遷移します。

#### (5) アクティブ(中速)モードからサブアクティブモードへの直接遷移

アクティブ(中速)モードでSYSCR1のSSBYを"1"、LSONを"1"、SYSCR2のDTONを"1"、TMAのTMA3を"1"にセットした状態でSLEEP命令を実行すると、ウォッチモー

ドを經由してサブアクティブモードに遷移します。

### (6) サブアクティブモードからアクティブ(中速)モードへの直接遷移

サブアクティブモードでSYSCR1のSSBYを"1"、LSONを"0"、SYSCR2のMSONを"1"、DTONを"1"、TMAのTMA3を"1"にセットした状態でSLEEP命令を実行すると、ウォッチモードを經由し、SYSCR1のSTS2~STS0により設定された時間を経過した後、直接、アクティブ(中速)モードに遷移します。

## 5.8.2 直接遷移の時間

### (1) アクティブ(高速)モードからアクティブ(中速)モードへの直接遷移時の時間について

アクティブ(高速)モードからアクティブ(中速)モードへの直接遷移はアクティブ(高速)モードでSYSCR1のSSBYを"0"、LSONを"0"、SYSCR2のMSONを"1"、DTONを"1"にセットした状態でSLEEP命令を実行することによって行われます。この場合のSLEEP命令実行から割込み例外処理が終るまでの時間(直接遷移時間)は(1)の計算式で表されます。

$$\begin{aligned} \text{直接遷移時間} = & \{ (\text{SLEEP命令実行ステート数}) + (\text{内部処理ステート数}) \} \\ & \times (\text{遷移前の } t_{\text{cyc}}) + (\text{割込み例外処理実行ステート数}) \\ & \times (\text{遷移後の } t_{\text{cyc}}) \dots\dots\dots (1) \end{aligned}$$

[例] H8/3627 シリーズの直接遷移時間 = (2 + 1) × 2t<sub>osc</sub> + 14 × 16t<sub>osc</sub> = 230t<sub>osc</sub>

<記号説明>

- t<sub>osc</sub> : OSCクロックサイクル時間
- t<sub>cyc</sub> : システムクロック( )サイクル時間

### (2) アクティブ(中速)モードからアクティブ(高速)モードへの直接遷移時の時間について

アクティブ(中速)モードからアクティブ(高速)モードへの直接遷移はアクティブ(中速)モードでSYSCR1のSSBYを"0"、LSONを"0"、SYSCR2のMSONを"0"、DTONを"1"にセットした状態でSLEEP命令を実行することによって行われます。この場合のSLEEP命令実行から割込み例外処理が終るまでの時間(直接遷移時間)は(2)の計算式で表されます。

$$\begin{aligned} \text{直接遷移時間} = & \{ (\text{SLEEP命令実行ステート数}) + (\text{内部処理ステート数}) \} \\ & \times (\text{遷移前の } t_{\text{cyc}}) + (\text{割込み例外処理実行ステート数}) \\ & \times (\text{遷移後の } t_{\text{cyc}}) \dots\dots\dots (2) \end{aligned}$$

[例] H8/3627 シリーズの直接遷移時間 = (2 + 1) × 16t<sub>osc</sub> + 14 × 2t<sub>osc</sub> = 76t<sub>osc</sub>

<記号説明>

- t<sub>osc</sub> : OSCクロックサイクル時間

$t_{cyc}$  : システムクロック ( ) サイクル時間

### (3) サブアクティブモードからアクティブ (高速) モードへの直接遷移時の時間について

サブアクティブモードからアクティブ (高速) モードへの直接遷移はサブアクティブモードで SYSCR1 の SSBY を "1"、LSON を "0"、SYSCR2 の MSON を "0"、DTON を "1"、TMA の TMA3 を "1" にセットした状態で SLEEP 命令を実行することによって行われます。この場合の SLEEP 命令実行から割込み例外処理が終るまでの時間 (直接遷移時間) は (3) の計算式で表されます。

$$\begin{aligned} \text{直接遷移時間} = & \{ (\text{SLEEP 命令実行ステート数}) + (\text{内部処理ステート数}) \} \\ & \times (\text{遷移前の } t_{\text{subcyc}}) + \{ (\text{STS2} \sim \text{STS0} \text{ で設定した待機時間}) \\ & + (\text{割込み例外処理実行ステート数}) \} \times (\text{遷移後の } t_{\text{cyc}}) \dots\dots\dots (3) \end{aligned}$$

$$\begin{aligned} \text{〔例〕 H8/3627 シリーズの直接遷移時間} &= (2+1) \times 8t_w + (8192+14) \times 2t_{\text{osc}} \\ &= 24t_w + 16412t_{\text{osc}} \end{aligned}$$

(CPU 動作クロック :  $\frac{1}{8}$ 、待機時間 : 8192 ステートを選択した場合)

<記号説明>

$t_{\text{osc}}$  : OSC クロックサイクル時間  
 $t_w$  : ウォッチクロックサイクル時間  
 $t_{\text{cyc}}$  : システムクロック ( ) サイクル時間  
 $t_{\text{subcyc}}$  : サブクロック (  $t_{\text{SUB}}$  ) サイクル時間

### (4) サブアクティブモードからアクティブ (中速) モードへの直接遷移時の時間について

サブアクティブモードからアクティブ (中速) モードへの直接遷移はサブアクティブモードで SYSCR1 の SSBY を "1"、LSON を "0"、SYSCR2 の MSON を "1"、DTON を "1"、TMA の TMA3 を "1" にセットした状態で SLEEP 命令を実行することによって行われます。SLEEP 命令実行から割込み例外処理が終るまでの時間 (直接遷移時間) は (4) の計算式で表されます。

$$\begin{aligned} \text{直接遷移時間} = & \{ (\text{SLEEP 命令実行ステート数}) + (\text{内部処理ステート数}) \} \\ & \times (\text{遷移前の } t_{\text{subcyc}}) + \{ (\text{STS2} \sim \text{STS0} \text{ で設定した待機時間}) \\ & + (\text{割込み例外処理実行ステート数}) \} \times (\text{遷移後の } t_{\text{cyc}}) \dots\dots\dots (4) \end{aligned}$$

$$\begin{aligned} \text{〔例〕 H8/3627 シリーズの直接遷移時間} &= (2+1) \times 8t_w + (8192+14) \times 16t_{\text{osc}} \\ &= 24t_w + 131296t_{\text{osc}} \end{aligned}$$

(CPU 動作クロック :  $\frac{1}{8}$ 、待機時間 : 8192 ステートを選択した場合)

<記号説明>

$t_{\text{osc}}$  : OSC クロックサイクル時間  
 $t_w$  : ウォッチクロックサイクル時間  
 $t_{\text{cyc}}$  : システムクロック ( ) サイクル時間  
 $t_{\text{subcyc}}$  : サブクロック (  $t_{\text{SUB}}$  ) サイクル時間

---

# 6. ROM

---

## 第6章 目次

6.1	概要.....	119
	6.1.1	ブロック図.....119
6.2	PROMモード.....	120
	6.2.1	PROMモードの設定.....120
	6.2.2	ソケットアダプタの端子対応とメモリマップ.....120
6.3	プログラミング.....	123
	6.3.1	書込み/ベリファイ.....124
	6.3.2	書込み時の注意.....126
6.4	書込み後の信頼性.....	128



## 6.1 概要

H8/3627 は 60k バイト、H8/3626 は 48k バイト、H8/3625 は 40k バイト、H8/3624S は 32k バイト、H8/3623S は 24k バイト、H8/3622S は 16k バイトのマスキング ROM を内蔵しています。また、H8/3627 は 60k バイトの PROM を内蔵しています。ROM は 16 ビット幅のデータバスで CPU と接続されており、バイトデータおよびワードデータにかかわらず 2 ステートの高速アクセスが可能です。

### 6.1.1 ブロック図

ROM のブロック図を図 6.1 に示します。

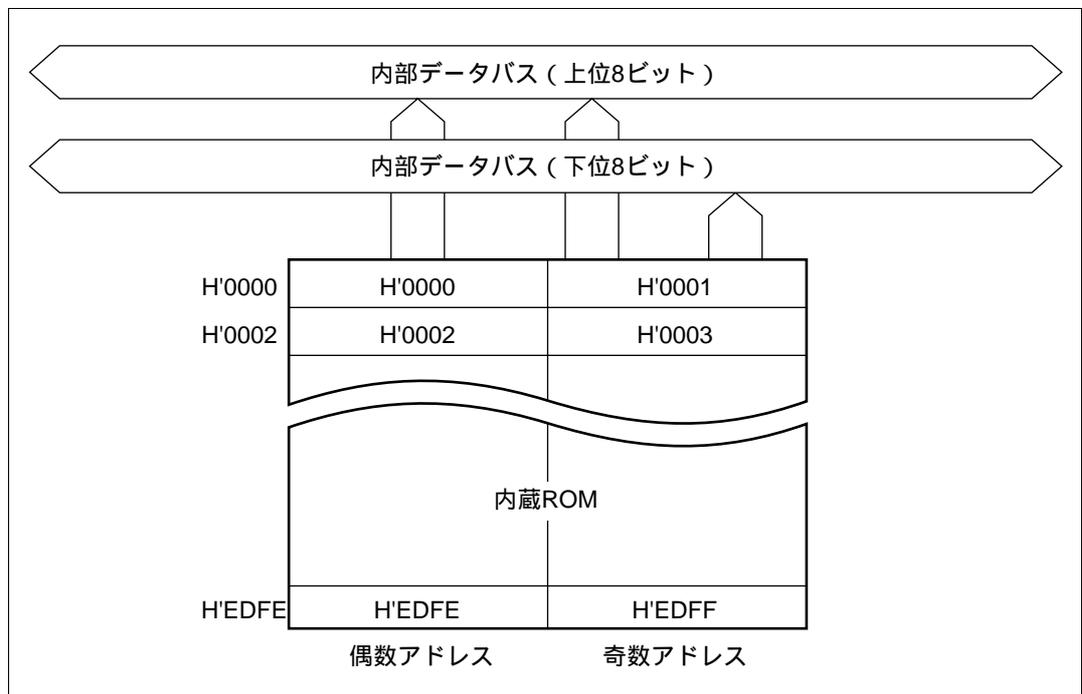


図 6.1 ROM のブロック図 (H8/3627 の場合)

## 6.2 PROM モード

### 6.2.1 PROM モードの設定

内蔵 ROM が PROM の場合、PROM モードに設定すると、マイクロコンピュータとしての機能が停止して、HN27C101 と同一の方法で内蔵 PROM のプログラムを行うことができます。

ただし、ページプログラミング方式はサポートしていません。

PROM モードの設定方法を、表 6.1 に示します。

表 6.1 PROM モードの設定

端子名	設定
TEST 端子	"High" レベル
PB <sub>7</sub> /AN <sub>7</sub> 端子	"Low" レベル
PB <sub>6</sub> /AN <sub>6</sub> 端子	

### 6.2.2 ソケットアダプタの端子対応とメモリマップ

PROM のプログラムは、パッケージに対応したソケットアダプタを付けて、32 ピンに変換し、汎用 PROM ライタでプログラミングを行います。

ソケットアダプタの端子対応図を図 6.2 に示します。また、メモリマップを図 6.3 に示します。

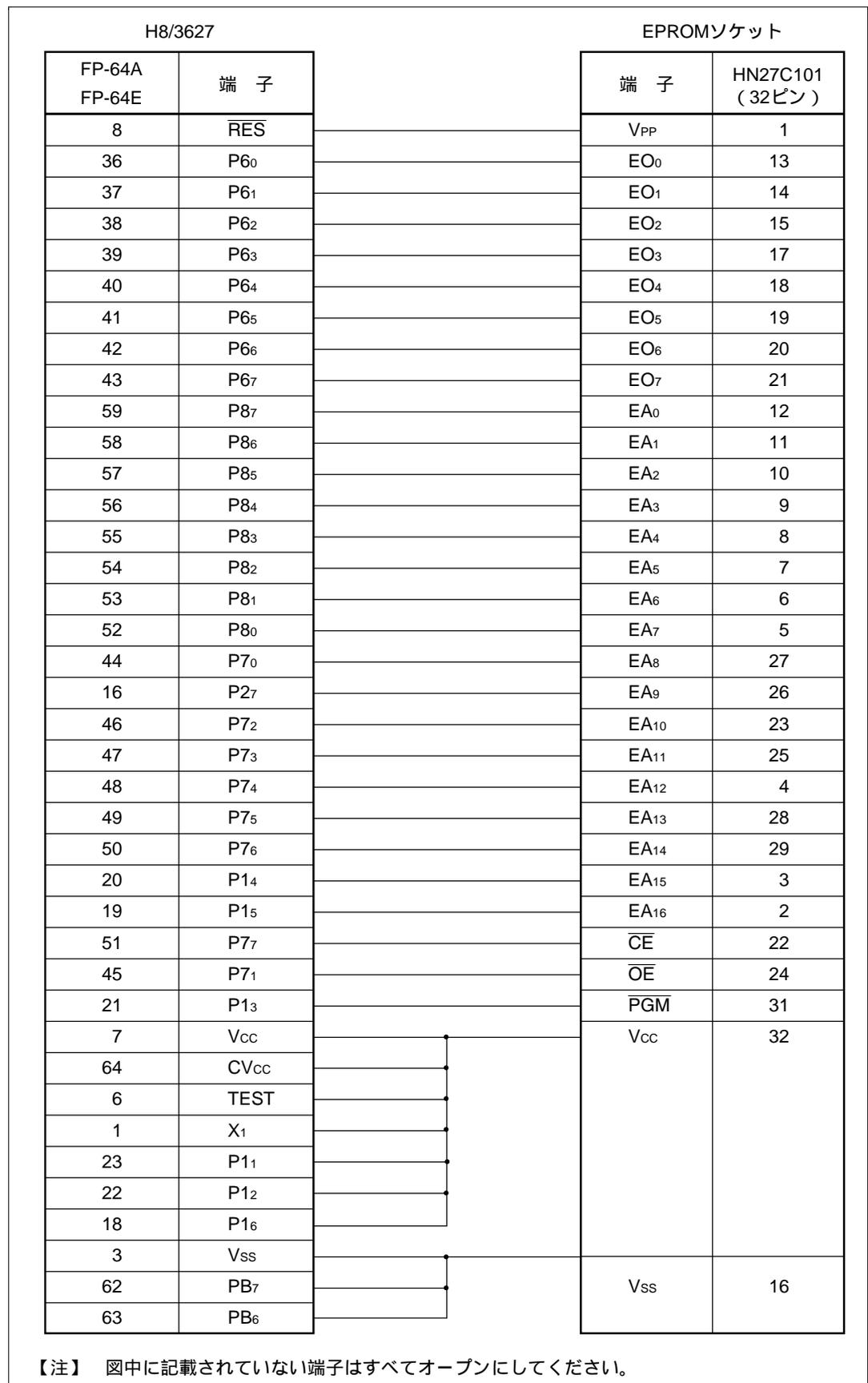


図 6.2 ソケットアダプタの端子対応図

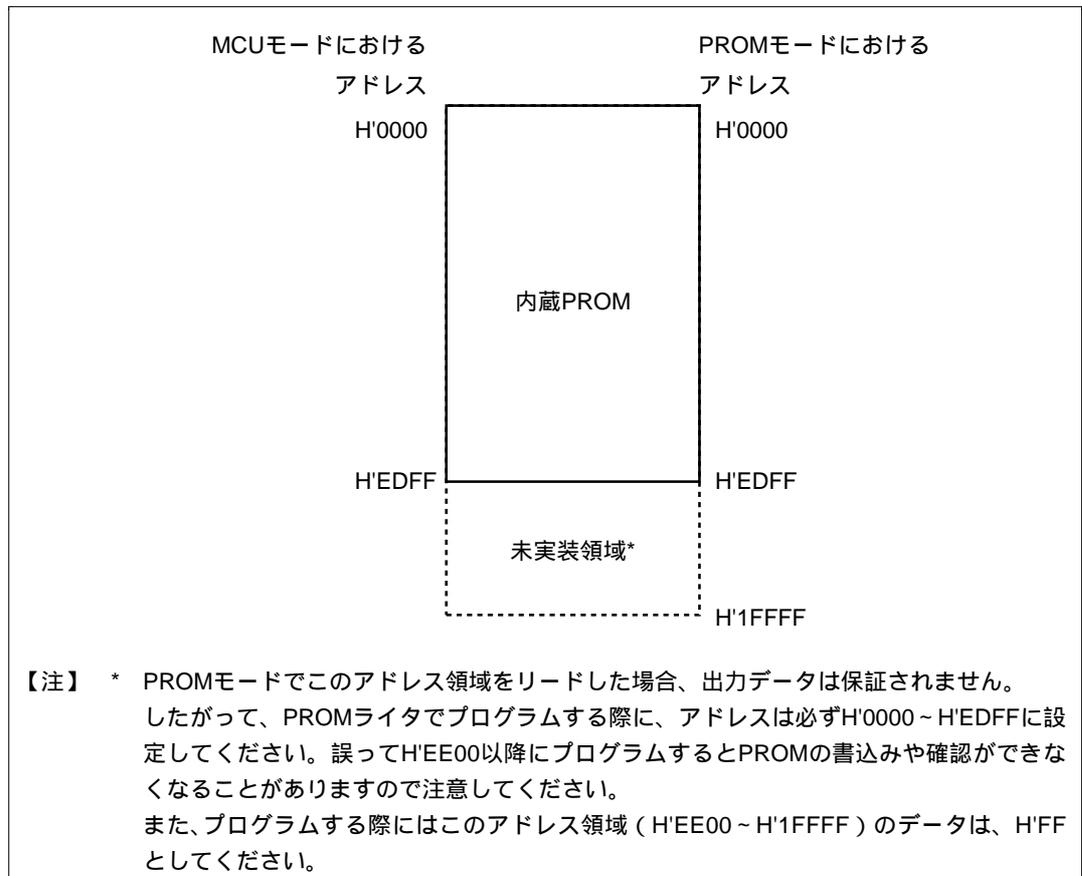


図 6.3 PROM モード時のメモリマップ

## 6.3 プログラミング

PROMモード時の書込み、ベリファイなどのモード選択は、表 6.2 に示すような設定により行います。

表 6.2 PROMモード時の書込みモードの選択

モード	ピン						
	$\overline{CE}$	$\overline{OE}$	$\overline{PGM}$	$V_{pp}$	$V_{CC}$	$EO_7 \sim EO_0$	$EA_{16} \sim EA_0$
書込み	L	H	L	$V_{pp}$	$V_{CC}$	データ入力	アドレス入力
ベリファイ	L	L	H	$V_{pp}$	$V_{CC}$	データ出力	アドレス入力
プログラミング禁止	L	L	L	$V_{pp}$	$V_{CC}$	ハイインピーダンス	アドレス入力
	L	H	H				
	H	L	L				
	H	H	H				

### 記号説明

L : "Low"レベル

H : "High"レベル

$V_{pp}$  : " $V_{pp}$ "レベル

$V_{CC}$  : " $V_{CC}$ "レベル

なお、書込み、読出しは、標準 EPROM の HN27C101 と同じ仕様になっています。ただし、ページプログラミング方式はサポートしていませんので、ページプログラミングモードに設定しないでください。ページプログラミングモードのみをサポートしている PROM ライタは使用できません。PROM ライタを選択する場合には、1 バイトごとの高速高信頼度プログラミング方式をサポートしていることを確認してください。また、アドレスは必ず H'0000 ~ H'EDFF に設定してください。

### 6.3.1 書込み / ベリファイ

書込み / ベリファイは効率のよい高速高信頼度プログラミング方式で行うことができます。この方式は、デバイスへの電圧ストレス、あるいは書込みデータの信頼性を損うことなく高速な書込みを行うことができます。未使用のアドレス領域のデータはHFFです。高速高信頼度プログラミングの基本的なフローを図 6.4 に示します。

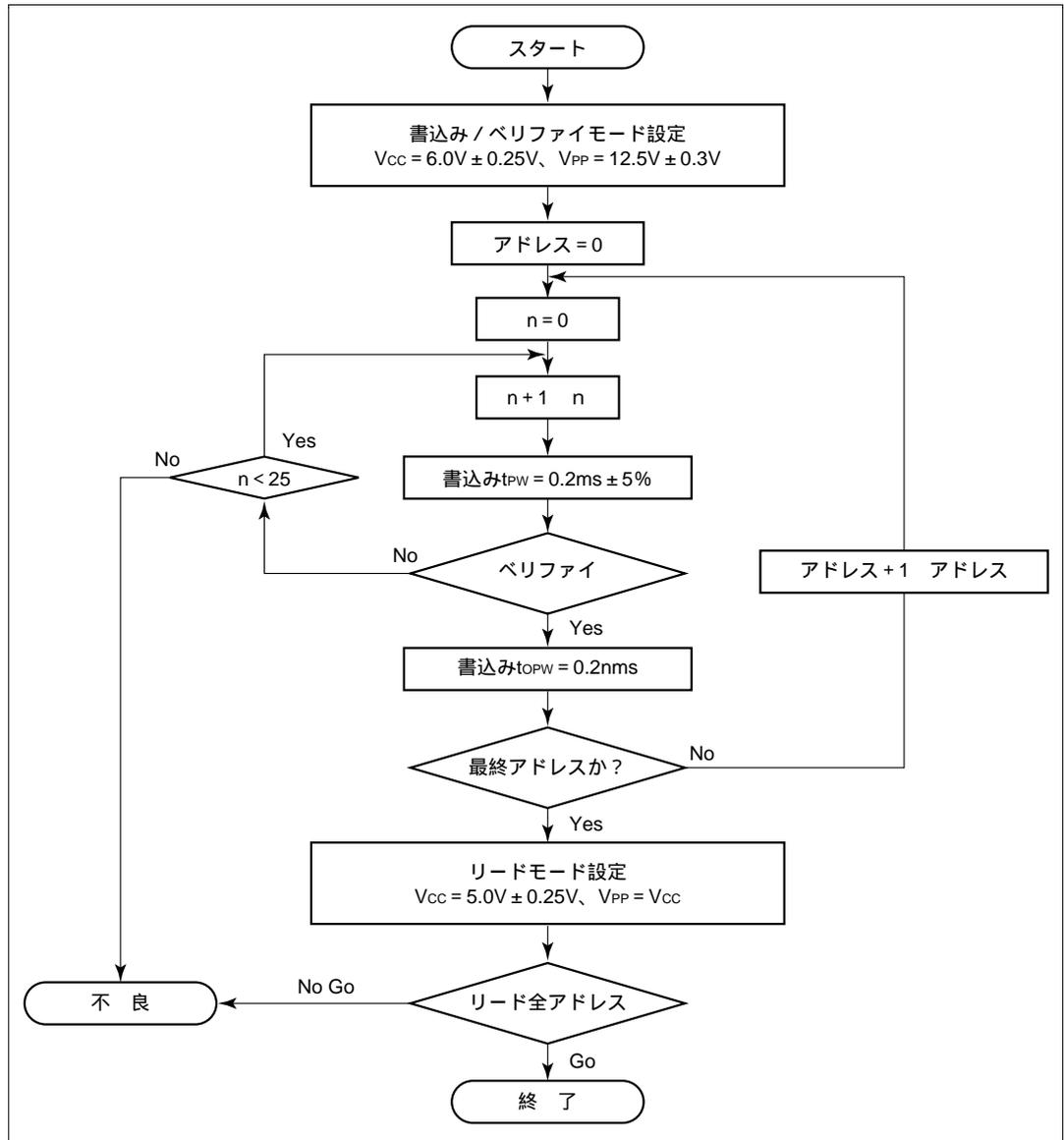


図 6.4 高速高信頼度プログラミングフローチャート

プログラミング時の電気的特性を表 6.3、表 6.4 に示します。

表 6.3 DC 特性

(条件:  $V_{CC} = 6.0V \pm 0.25V$ 、 $V_{PP} = 12.5V \pm 0.3V$ 、 $V_{SS} = 0V$ 、 $T_a = 25 \pm 5$ )

項目		記号	min	typ	max	単位	測定条件
入力"High" レベル電圧	$EO_7 \sim EO_0$ 、 $EA_{16} \sim EA_0$ $\overline{OE}$ 、 $\overline{CE}$ 、 $\overline{PGM}$	$V_{IH}$	2.4		$V_{CC} + 0.3$	V	
入力"Low" レベル電圧	$EO_7 \sim EO_0$ 、 $EA_{16} \sim EA_0$ $\overline{OE}$ 、 $\overline{CE}$ 、 $\overline{PGM}$	$V_{IL}$	-0.3		0.8	V	
出力"High" レベル電圧	$EO_7 \sim EO_0$	$V_{OH}$	2.4			V	$I_{OH} = -200\mu A$
出力"Low" レベル電圧	$EO_7 \sim EO_0$	$V_{OL}$			0.45	V	$I_{OL} = 0.8mA$
入力 リーク電流	$EO_7 \sim EO_0$ 、 $EA_{16} \sim EA_0$ $\overline{OE}$ 、 $\overline{CE}$ 、 $\overline{PGM}$	$ I_{LI} $			2	$\mu A$	$V_{in} = 5.25V/0.5V$
$V_{CC}$ 電流		$I_{CC}$			40	mA	
$V_{PP}$ 電流		$I_{PP}$			40	mA	

表 6.4 AC 特性

(条件:  $V_{CC} = 6.0V \pm 0.25V$ 、 $V_{PP} = 12.5V \pm 0.3V$ 、 $T_a = 25 \pm 5$ )

項目	記号	min	typ	max	単位	測定条件
アドレスセットアップ時間	$t_{AS}$	2			$\mu s$	図 6.5*1
$\overline{OE}$ セットアップ時間	$t_{OES}$	2			$\mu s$	
データセットアップ時間	$t_{DS}$	2			$\mu s$	
アドレスホールド時間	$t_{AH}$	0			$\mu s$	
データホールド時間	$t_{DH}$	2			$\mu s$	
データ出力ディスエーブル時間	$t_{DF}^{*2}$			130	ns	
$V_{PP}$ セットアップ時間	$t_{VPS}$	2			$\mu s$	
プログラムパルス幅	$t_{PW}$	0.19	0.20	0.21	ms	
オーバプログラム時の $\overline{PGM}$ パルス幅	$t_{OPW}^{*3}$	0.19		5.25	ms	
$V_{CC}$ セットアップ時間	$t_{VCS}$	2			$\mu s$	
$\overline{CE}$ セットアップ時間	$t_{CES}$	2			$\mu s$	
データ出力遅延時間	$t_{OE}$	0		200	ns	

【注】 \*1 入力パルスレベル: 0.45 ~ 2.4V

入力立上がり / 立下がり時間 20ns

タイミング参照レベル 入力: 0.8V、2.0V 出力: 0.8V、2.0V

\*2  $t_{DF}$  は出力が開放状態に達し、出力レベルを参照できなくなった場合で定義します。

\*3  $t_{OPW}$  は図 6.4 高速高信頼度プログラミングフローチャートに記載した値で定義されます。

PROMの書込み/ベリファイタイミングを図6.5に示します。

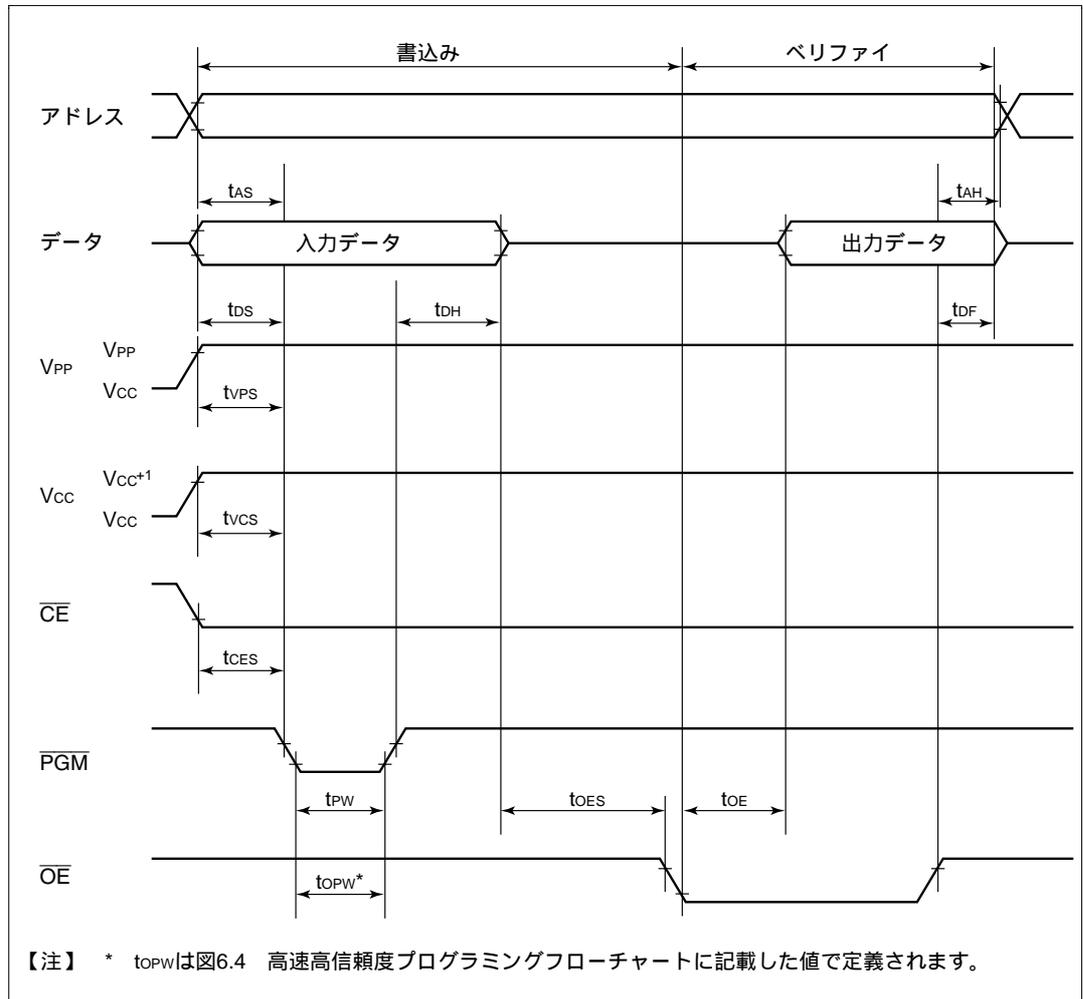


図 6.5 PROM 書込み/ベリファイタイミング

### 6.3.2 書込み時の注意

- (1) 書込みは規定された電圧、タイミングで行ってください。  
PROMモード時のプログラム電圧 ( $V_{pp}$ ) は 12.5V です。  
定格以上の電圧を加えると、製品の永久破壊に至りますので、注意してください。特に PROM ライタのオーバシュートなどには十分注意してください。  
PROM ライタの HN27C101 の日立仕様にセットすると、 $V_{pp}$  は 12.5V になります。
- (2) PROM ライタのソケット、ソケットアダプタおよび製品それぞれのインデックスが正しく一致していないと、過剰電流によって製品が破壊することがあります。書込み前に正しく PROM ライタに装着されていることを必ず確認してください。
- (3) 書込み中はソケットアダプタおよび製品には触れないようにしてください。接触不良により書込み不良となる場合があります。
- (4) プログラミングモードは、ページプログラミング方式をサポートしていませんので、プログラミングモードの設定には注意してください。

- (5) PROMライターでプログラムする際に、アドレスは必ずH0000~HEDFFに設定してください。誤ってHEE00以降にプログラムすると、PROMの書込みや確認ができなくなる場合がありますので注意してください。またプログラムする際にはHEE00~H1FFFFのアドレス領域のデータはHFFとしてください。

## 6.4 書込み後の信頼性

データ書込み後、データ保持特性を向上させるために、150 の高温放置をしてスクリーニングを行うと大変有効です。高温放置は、スクリーニングのひとつであり、PROMメモリセルの初期のデータ保持不良を短時間で除くことができます。

図 6.6 に推奨するスクリーニングフローを示します。

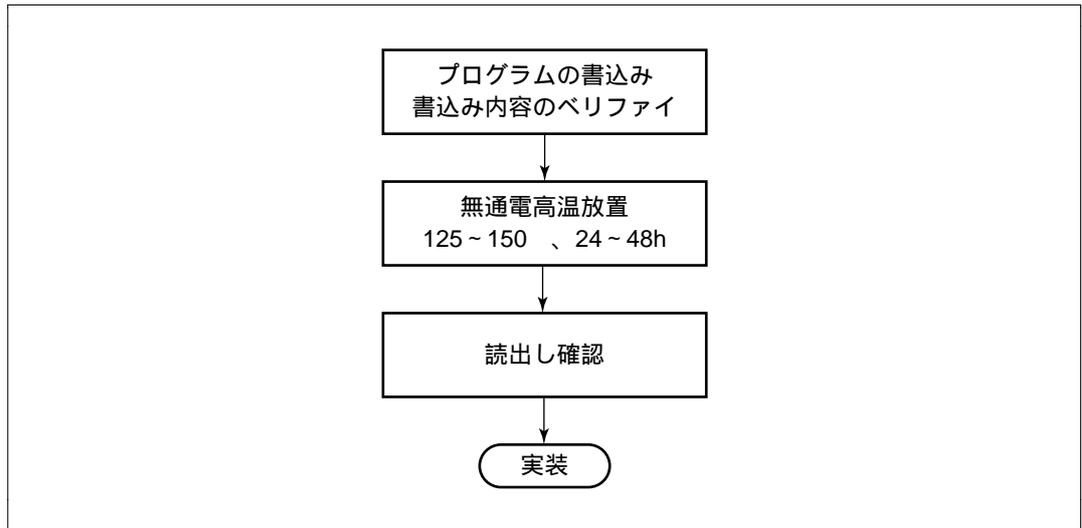


図 6.6 推奨スクリーニングフロー

同じ PROM ライタでプログラミング中、書込み不良が連続して発生した場合には書込みを中止し、PROM ライタ、ソケットアダプタなどに異常がないか確認してください。

なお、書込みあるいは高温放置後のプログラム確認において異常がありましたら、当社技術担当にご連絡ください。

---

# 7. RAM

---

## 第7章 目次

7.1	概要.....	131
7.1.1	ブロック図.....	131



## 7.1 概要

本LSIは、1kバイトまたは2kバイトの高速スタティックRAMを内蔵しています。RAMは16ビット幅のデータバスでCPUと接続されており、バイトデータおよびワードデータにかかわらず2ステートの高速アクセスが可能です。

### 7.1.1 ブロック図

RAMのブロック図を図7.1に示します。

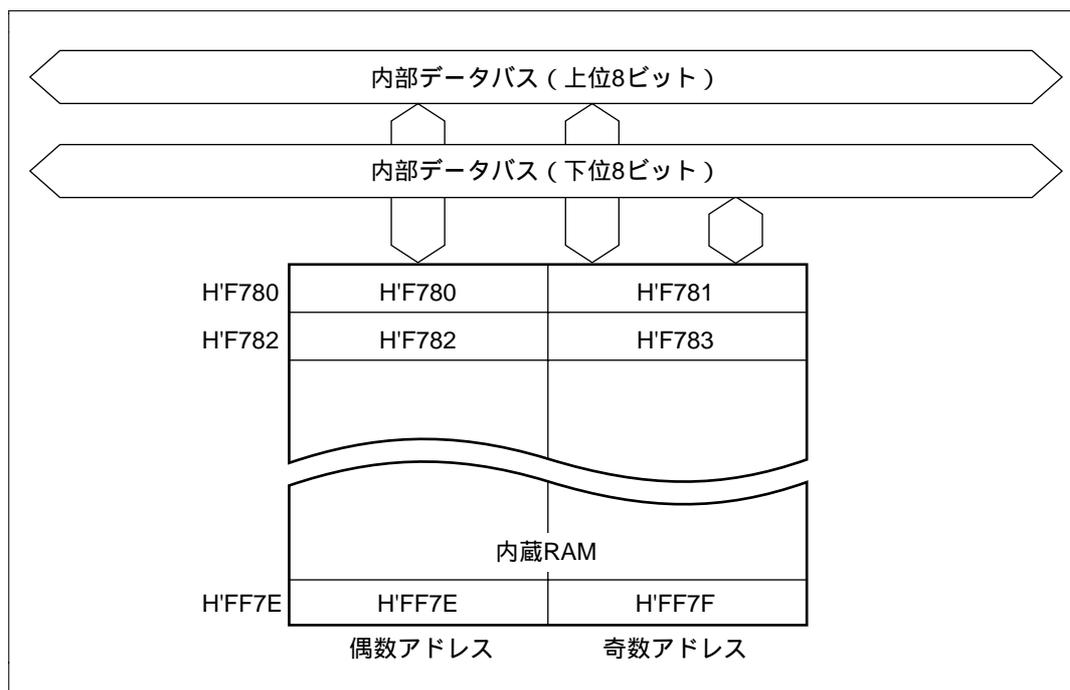


図7.1 RAMのブロック図 (2kバイトRAMの場合)

---

# 8. I/O ポート

---

## 第8章 目次

8.1	概要	135
8.2	ポート1	137
	8.2.1 概要	137
	8.2.2 レジスタの構成と説明	137
	8.2.3 端子機能	142
	8.2.4 端子状態	143
	8.2.5 入力プルアップ MOS	144
8.3	ポート2	145
	8.3.1 概要	145
	8.3.2 レジスタの構成と説明	145
	8.3.3 端子機能	150
	8.3.4 端子状態	152
	8.3.5 入力プルアップ MOS	152
8.4	ポート5	153
	8.4.1 概要	153
	8.4.2 レジスタの構成と説明	153
	8.4.3 端子機能	155
	8.4.4 端子状態	156
	8.4.5 入力プルアップ MOS	156
8.5	ポート6	157
	8.5.1 概要	157
	8.5.2 レジスタの構成と説明	157
	8.5.3 端子機能	159
	8.5.4 端子状態	159
	8.5.5 入力プルアップ MOS	159
8.6	ポート7	160
	8.6.1 概要	160

	8.6.2	レジスタの構成と説明 .....	160
	8.6.3	端子機能 .....	162
	8.6.4	端子状態 .....	162
8.7	ポート 8 .....		163
	8.7.1	概要 .....	163
	8.7.2	レジスタの構成と説明 .....	163
	8.7.3	端子機能 .....	165
	8.7.4	端子状態 .....	165
8.8	ポート A .....		166
	8.8.1	概要 .....	166
	8.8.2	レジスタの構成と説明 .....	166
	8.8.3	端子機能 .....	167
	8.8.4	端子状態 .....	167
8.9	ポート B .....		168
	8.9.1	概要 .....	168
	8.9.2	レジスタの構成と説明 .....	168

## 8.1 概要

本LSIは、8ビット入出力ポートを5本、7ビット入出力ポートを1本、3ビット入出力ポートを1本、2ビット入力専用ポートを1本、1ビット入力専用ポートを1本備えています。

各ポートの機能一覧を表8.1に示します。

各ポートは、入出力を制御するポートコントロールレジスタ(PCR)と、出力データを格納するポートデータレジスタ(PDR)から構成され、ビット単位に入出力を制御できます。PCR、PDR に対してのビット操作命令の実行については、「2.9.2 ビット操作命令 使用上の注意事項」を参照してください。

各ポートのブロック図は「付録C. I/Oポートブロック図」を参照してください。

表 8.1 ポートの機能

ポート	概要	端子	兼用機能	機能切換え 制御レジスタ
ポート 1	<ul style="list-style-type: none"> <li>・ 8 ビット入出力ポート</li> <li>・ 入力プルアップ MOS 選択可</li> </ul>	P1 <sub>7</sub> ~ P1 <sub>5</sub> /	外部割込み 3~1	PMR1
		$\overline{IRQ}_3 \sim \overline{IRQ}_1$ / TMIF	タイマイベント入力 TMIF	TCRF
		P1 <sub>4</sub>	なし	
		P1 <sub>3</sub> /TMIG	タイマ G インพุットキャプチャ入力	PMR1
		P1 <sub>2</sub> , P1 <sub>1</sub> / TMOFH, TMOFL	タイマ F アウトプットコンペア出力	PMR1
P1 <sub>0</sub> /TMOW	タイマ A クロック出力	PMR1		
ポート 2	<ul style="list-style-type: none"> <li>・ 7 ビット入出力ポート</li> <li>・ 入力プルアップ MOS 選択可</li> <li>・ 1 ビット入力専用ポート</li> </ul>	P2 <sub>7</sub> / $\overline{IRQ}_0$	外部割込み 0	PMR2
		P2 <sub>6</sub> /TXD	SCI3 のデータ出力(TXD)、データ入力	SCR3
		P2 <sub>5</sub> /RXD	(RXD)クロック入出力(SCK <sub>3</sub> )	SMR3
		P2 <sub>4</sub> /SCK <sub>3</sub>		PMR6
		P2 <sub>3</sub> /SO <sub>1</sub> P2 <sub>2</sub> /SI <sub>1</sub> P2 <sub>1</sub> /SCK <sub>1</sub>	SCI1 のデータ出力(SO <sub>1</sub> )、データ入力 (SI <sub>1</sub> )クロック入出力(SCK <sub>1</sub> )	PMR2
P2 <sub>0</sub> / $\overline{IRQ}_4$ /ADTRG	外部割込み 4 および A/D 変換器の外部トリガ	PMR2		
ポート 5	<ul style="list-style-type: none"> <li>・ 8 ビット入出力ポート</li> <li>・ 入力プルアップ MOS 選択可</li> </ul>	P5 <sub>7</sub> ~ P5 <sub>0</sub> / $\overline{WKP}_7 \sim \overline{WKP}_0$	・ ウェイクアップ入力 ( $\overline{WKP}_7 \sim \overline{WKP}_0$ )	PMR5
ポート 6	<ul style="list-style-type: none"> <li>・ 8 ビット入出力ポート</li> <li>・ 入力プルアップ MOS 選択可</li> </ul>	P6 <sub>7</sub> ~ P6 <sub>0</sub>	なし	
ポート 7	・ 8 ビット入出力ポート	P7 <sub>7</sub> ~ P7 <sub>0</sub>	なし	
ポート 8	・ 8 ビット入出力ポート	P8 <sub>7</sub> ~ P8 <sub>0</sub>	なし	
ポート A	・ 3 ビット入出力ポート	PA <sub>3</sub> ~ PA <sub>1</sub>	なし	
ポート B	・ 2 ビット入力専用ポート	PB <sub>7</sub> 、PB <sub>6</sub> / AN <sub>7</sub> 、AN <sub>6</sub>	A/D 変換器のアナログ入力	AMR

## 8.2 ポート 1

### 8.2.1 概要

ポート 1 は、8 ビットの入出力ポートです。ポート 1 の各端子は、図 8.1 に示す構成になっています。

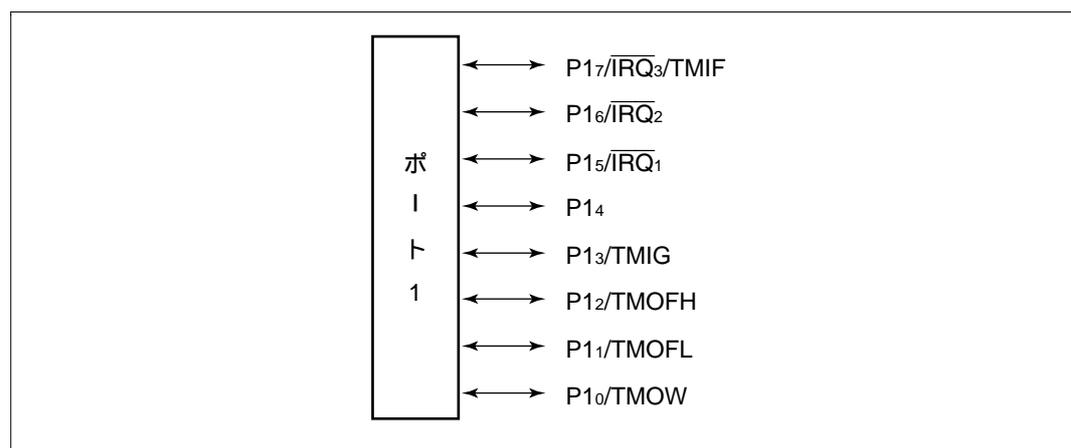


図 8.1 ポート 1 の端子構成

### 8.2.2 レジスタの構成と説明

表 8.2 にポート 1 のレジスタ構成を示します。

表 8.2 レジスタ構成

名称	略称	R/W	初期値	アドレス
ポートデータレジスタ 1	PDR1	R/W	H'00	H'FFD4
ポートコントロールレジスタ 1	PCR1	W	H'00	H'FFE4
ポートプルアップコントロールレジスタ 1	PUCR1	R/W	H'00	H'FF9C
ポートモードレジスタ 1	PMR1	R/W	H'00	H'FF98

## (1) ポートデータレジスタ1 (PDR1)

ビット:	7	6	5	4	3	2	1	0
	P17	P16	P15	P14	P13	P12	P11	P10
初期値:	0	0	0	0	0	0	0	0
R/W :	R/W							

PDR1は、ポート1の各端子P1<sub>7</sub>~P1<sub>0</sub>のデータを格納する8ビットのレジスタです。

PCR1が"1"のとき、ポート1のリードを行うと、PDR1の値を直接リードします。そのため端子状態の影響を受けません。PCR1が"0"のとき、ポート1のリードを行うと、端子状態が読み出されます。

リセット時、PDR1はH'00にイニシャライズされます。

## (2) ポートコントロールレジスタ1 (PCR1)

ビット:	7	6	5	4	3	2	1	0
	PCR17	PCR16	PCR15	PCR14	PCR13	PCR12	PCR11	PCR10
初期値:	0	0	0	0	0	0	0	0
R/W :	W	W	W	W	W	W	W	W

PCR1は、ポート1の各端子P1<sub>7</sub>~P1<sub>0</sub>の入出力をビットごとに制御します。PCR1に"1"をセットすると対応するP1<sub>7</sub>~P1<sub>0</sub>端子は出力端子となり、"0"にクリアすると入力端子となります。PMR1により当該端子が汎用入出力に設定されている場合には、PCR1およびPDR1の設定が有効となります。

リセット時、PCR1はH'00にイニシャライズされます。

本レジスタはライト専用ですが、リードした場合各ビットは常に"1"が読み出されます。

## (3) ポートプルアップコントロールレジスタ1 (PUCR1)

ビット:	7	6	5	4	3	2	1	0
	PUCR17	PUCR16	PUCR15	PUCR14	PUCR13	PUCR12	PUCR11	PUCR10
初期値:	0	0	0	0	0	0	0	0
R/W :	R/W							

PUCR1は、ポート1の各端子P1<sub>7</sub>~P1<sub>0</sub>のプルアップMOSをビットごとに制御します。

PCR1が"0"の状態でもPUCR1に"1"をセットすると対応するプルアップMOSはON状態となり、"0"にクリアするとOFF状態となります。

リセット時、PUCR1はH'00にイニシャライズされます。

## (4) ポートモードレジスタ1 (PMR1)

ビット:	7	6	5	4	3	2	1	0
	IRQ3	IRQ2	IRQ1	-	TMIG	TMOFH	TMOFL	TMOW
初期値:	0	0	0	0	0	0	0	0
R/W :	R/W	R/W	R/W	-	R/W	R/W	R/W	R/W

PMR1 は、8ビットのリード/ライト可能なレジスタで、ポート1の各端子機能の切換えを制御します。

リセット時、PMR1はH'00にイニシャライズされます。

ビット7: P1<sub>7</sub>/ $\overline{\text{IRQ}}_3$ /TMIF 端子機能切換え (IRQ3)

P1<sub>7</sub>/ $\overline{\text{IRQ}}_3$ /TMIF 端子を P1<sub>7</sub> 端子として使用するか、 $\overline{\text{IRQ}}_3$ /TMIF 端子として使用するかを設定します。

ビット7	説明
IRQ3	
0	P1 <sub>7</sub> 入出力端子として機能 (初期値)
1	$\overline{\text{IRQ}}_3$ /TMIF 入力端子として機能

【注】  $\overline{\text{IRQ}}_3$ /TMIF 端子は立上がり/立下がりエッジセンスを選択できます。TMIF 端子の設定については、「9.3.2 (3) タイマコントロールレジスタ F (TCRF)」を参照してください。

ビット6: P1<sub>6</sub>/ $\overline{\text{IRQ}}_2$  端子機能切換え (IRQ2)

P1<sub>6</sub>/ $\overline{\text{IRQ}}_2$  端子を P1<sub>6</sub> 端子として使用するか、 $\overline{\text{IRQ}}_2$  端子として使用するかを設定します。

ビット6	説明
IRQ2	
0	P1 <sub>6</sub> 入出力端子として機能 (初期値)
1	$\overline{\text{IRQ}}_2$ 入力端子として機能

【注】  $\overline{\text{IRQ}}_2$  端子は立上がり/立下がりエッジセンスを選択できます。

ビット5 : P1<sub>5</sub>/ $\overline{\text{IRQ}}_1$  端子機能切換え (IRQ1)

P1<sub>5</sub>/ $\overline{\text{IRQ}}_1$  端子を P1<sub>5</sub> 端子として使用するか、 $\overline{\text{IRQ}}_1$  端子として使用するかを設定します。

ビット5		
IRQ1	説明	
0	P1 <sub>5</sub> 入出力端子として機能	(初期値)
1	$\overline{\text{IRQ}}_1$ 入力端子として機能	

【注】  $\overline{\text{IRQ}}_1$  端子は立上がり / 立下がりエッジセンスを選択できます。

## ビット4 : リザーブビット

リザーブビットです。本ビットは、リードすると常に"0"が読み出されます。本ビットは"0"で使用して下さい。

ビット3 : P1<sub>3</sub>/TMIG 端子機能切換え (TMIG)

P1<sub>3</sub>/TMIG 端子を P1<sub>3</sub> 端子として使用するか、TMIG 端子として使用するかを設定します。

ビット3		
TMIG	説明	
0	P1 <sub>3</sub> 入出力端子として機能	(初期値)
1	TMIG 入力端子として機能	

ビット2 : P1<sub>2</sub>/TMOFH 端子機能切換え (TMOFH)

P1<sub>2</sub>/TMOFH 端子を P1<sub>2</sub> 端子として使用するか、TMOFH 端子として使用するかを設定します。

ビット2		
TMOFH	説明	
0	P1 <sub>2</sub> 入出力端子として機能	(初期値)
1	TMOFH 出力端子として機能	

ビット1：P1<sub>7</sub>/TMOFL 端子機能切換え（TMOFL）

P1<sub>7</sub>/TMOFL 端子を P1<sub>7</sub> 端子として使用するか、TMOFL 端子として使用するかを設定します。

ビット1	
TMOFL	説明
0	P1 <sub>7</sub> 入出力端子として機能 (初期値)
1	TMOFL 出力端子として機能

ビット0：P1<sub>0</sub>/TMOW 端子機能切換え（TMOW）

P1<sub>0</sub>/TMOW 端子を P1<sub>0</sub> 端子として使用するか、TMOW 端子として使用するかを設定します。

ビット0	
TMOW	説明
0	P1 <sub>0</sub> 入出力端子として機能 (初期値)
1	TMOW 出力端子として機能

## 8.2.3 端子機能

表 8.3 にポート 1 の端子機能を示します。

表 8.3 ポート 1 の端子機能 (1)

端子	選択方法と端子機能			
P1 <sub>7</sub> / $\overline{\text{IRQ}}_3$ / TMIF	PMR1 の IRQ3、TCRF の CKSL2 ~ CKSL0、PCR1 の PCR1 <sub>7</sub> の組合せで次のように切 換わります。			
	IRQ3	0		1
	PCR1 <sub>7</sub>	0	1	*
	CKSL2 ~ CKSL0	*		0**以外
	端 子 機 能	P1 <sub>7</sub> 入力端子	P1 <sub>7</sub> 出力端子	$\overline{\text{IRQ}}_3$ /TMIF 入力端子
	【注】 TMIF入力端子として使用する場合には、IENR1のIEN3を"0"にクリアして、IRQ <sub>3</sub> 割込みを禁止してください。			
P1 <sub>6</sub> / $\overline{\text{IRQ}}_2$	PMR1 の IRQ2 と PCR1 の PCR1 <sub>6</sub> の組合せで次のように切換わります。			
	IRQ2	0		1
	PCR1 <sub>6</sub>	0	1	*
	端 子 機 能	P1 <sub>6</sub> 入力端子	P1 <sub>6</sub> 出力端子	$\overline{\text{IRQ}}_2$ 入力端子
P1 <sub>5</sub> / $\overline{\text{IRQ}}_1$	PMR1 の IRQ1 と PCR1 の PCR1 <sub>5</sub> の組合せで、次のように切換わります。			
	IRQ1	0		1
	PCR1 <sub>5</sub>	0	1	*
	端 子 機 能	P1 <sub>5</sub> 入力端子	P1 <sub>5</sub> 出力端子	$\overline{\text{IRQ}}_1$ 入力端子
P1 <sub>4</sub>	PCR1 の PCR1 <sub>4</sub> により、次のように切換わります。			
	PCR1 <sub>4</sub>	0	1	
	端 子 機 能	P1 <sub>4</sub> 入力端子	P1 <sub>4</sub> 出力端子	
P1 <sub>3</sub> /TMIG	PMR1 の TMIG と PCR1 の PCR1 <sub>3</sub> の組合せで、次のように切換わります。			
	TMIG	0		1
	PCR1 <sub>3</sub>	0	1	*
	端 子 機 能	P1 <sub>3</sub> 入力端子	P1 <sub>3</sub> 出力端子	TMIG入力端子

\* : Don't care

表 8.3 ポート1の端子機能(2)

端子	選択方法と端子機能			
P1 <sub>2</sub> /TMOFH	PMR1のTMOFHとPCR1のPCR1 <sub>2</sub> の組合せで、次のように切換わります。			
	TMOFH	0		1
	PCR1 <sub>2</sub>	0	1	*
	端子機能	P1 <sub>2</sub> 入力端子	P1 <sub>2</sub> 出力端子	TMOFH出力端子
P1 <sub>1</sub> /TMOFL	PMR1のTMOFLとPCR1のPCR1 <sub>1</sub> の組合せで、次のように切換わります。			
	TMOFL	0		1
	PCR1 <sub>1</sub>	0	1	*
	端子機能	P1 <sub>1</sub> 入力端子	P1 <sub>1</sub> 出力端子	TMOFL出力端子
P1 <sub>0</sub> /TMOW	PMR1のTMOWとPCR1のPCR1 <sub>0</sub> の組合せで、次のように切換わります。			
	TMOW	0		1
	PCR1 <sub>0</sub>	0	1	*
	端子機能	P1 <sub>0</sub> 入力端子	P1 <sub>0</sub> 出力端子	TMOW出力端子

\* : Don't care

## 8.2.4 端子状態

各動作モードにおけるポート1の端子状態を表8.4に示します。

表 8.4 ポート1の端子状態

端子名	リセット	スリープ	サブスリープ	スタンバイ	ウォッチ	サブアクティブ	アクティブ
P1 <sub>7</sub> / $\overline{\text{IRQ}}_3$ /TMIF							
P1 <sub>6</sub> / $\overline{\text{IRQ}}_2$							
P1 <sub>5</sub> / $\overline{\text{IRQ}}_1$	ハイ			ハイ			
P1 <sub>4</sub>	インピーダンス	保持	保持	インピーダンス*	保持	動作	動作
P1 <sub>3</sub> /TMIG							
P1 <sub>2</sub> /TMOFH							
P1 <sub>1</sub> /TMOFL							
P1 <sub>0</sub> /TMOW							

【注】 \* プルアップMOSがON状態では"High"レベル出力となります。

### 8.2.5 入力プルアップ MOS

ポート 1 は、プログラムで制御可能な入力プルアップ MOS を内蔵しています。PCR1 が"0"にクリアされている状態で PUCR1 に"1"をセットすると入力プルアップ MOS は ON 状態となります。また、入力プルアップ MOS はリセット時、OFF 状態になります。

PCR1 <sub>n</sub>	0		1
PUCR1 <sub>n</sub>	0	1	*
入力プルアップ MOS	OFF	ON	OFF

\* : Don't care (n = 7 ~ 0)

## 8.3 ポート 2

### 8.3.1 概要

ポート 2 は、7 ビットの入出力ポートと 1 ビットの入力専用ポートです。ポート 2 の各端子は、図 8.2 に示す構成になっています。

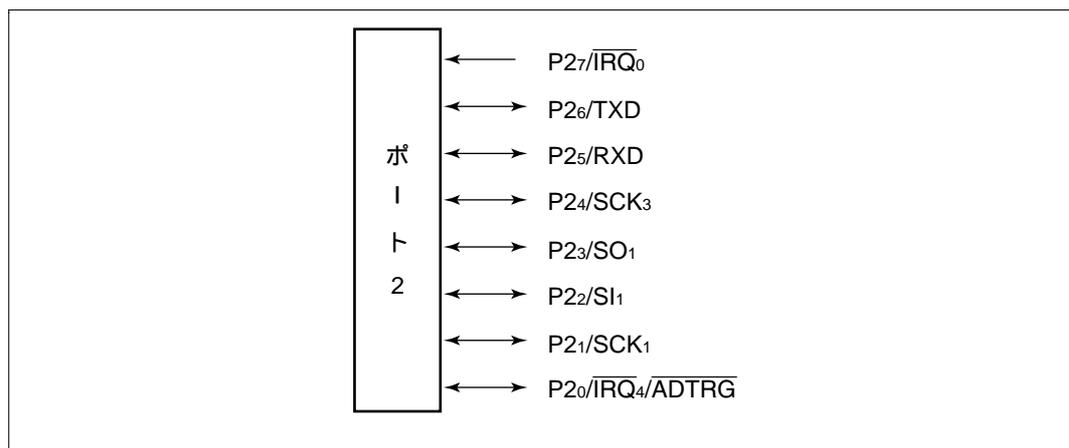


図 8.2 ポート 2 の端子構成

### 8.3.2 レジスタの構成と説明

表 8.5 にポート 2 のレジスタ構成を示します。

表 8.5 レジスタ構成

名称	略称	R/W	初期値	アドレス
ポートデータレジスタ 2	PDR2	R/W	H'00	H'FFD5
ポートコントロールレジスタ 2	PCR2	W	H'00	H'FFE5
ポートモードレジスタ 2	PMR2	R/W	H'40	H'FF99
ポートモードレジスタ 6	PMR6	R/W	H'F8	H'FF9A
ポートプルアップコントロールレジスタ 2	PUCR2	R/W	H'00	H'FF9D

## (1) ポートデータレジスタ 2 (PDR2)

ビット:	7	6	5	4	3	2	1	0
	P27	P26	P25	P24	P23	P22	P21	P20
初期値:	0	0	0	0	0	0	0	0
R/W :	R/W							

PDR2は、ポート2の各端子P2<sub>7</sub>~P2<sub>0</sub>のデータを格納する8ビットのレジスタです。

PCR2が"1"のとき、ポート2のリードを行うと、PDR2の値を直接リードします。そのため端子状態の影響を受けません。PCR2が"0"のとき、ポート2のリードを行うと、端子状態が読み出されます。

リセット時、PDR2はH'00にイニシャライズされます。

## (2) ポートコントロールレジスタ 2 (PCR2)

ビット:	7	6	5	4	3	2	1	0
	PCR27	PCR26	PCR25	PCR24	PCR23	PCR22	PCR21	PCR20
初期値:	0	0	0	0	0	0	0	0
R/W :	W	W	W	W	W	W	W	W

PCR2は、ポート2の各端子P2<sub>7</sub>~P2<sub>0</sub>の入出力をビットごとに制御します。PCR2に"1"をセットすると対応するP2<sub>7</sub>~P2<sub>0</sub>端子は出力端子となり、"0"にクリアすると入力端子となります。PMR2により当該端子が汎用入出力に設定されている場合には、PCR2およびPDR2の設定が有効となります。

リセット時、PCR2はH'00にイニシャライズされます。

本レジスタはライト専用ですが、リードした場合各ビットは常に"1"が読み出されます。

【注】 P2<sub>7</sub>端子は入力専用端子のため、PCR2<sub>7</sub>に"1"をセットした場合、ハイインピーダンス出力となります。

## (3) ポートモードレジスタ 2 (PMR2)

ビット:	7	6	5	4	3	2	1	0
	IRQ0	—	POF1	NCS	SO1	SI1	SCK1	IRQ4
初期値:	0	1	0	0	0	0	0	0
R/W :	R/W	—	R/W	R/W	R/W	R/W	R/W	R/W

PMR2は、8ビットのリード/ライト可能なレジスタで、P2<sub>0</sub>~P2<sub>3</sub>、P2<sub>7</sub>端子機能の切換え、P2<sub>3</sub>/SO<sub>1</sub>端子のPMOSのON/OFF制御、およびTMIG端子入力のノイズキャンセラを制御します。

リセット時、PMR2はH'40にイニシャライズされます。

ビット7：P2<sub>7</sub>/ $\overline{\text{IRQ}}_0$ 端子機能切替え(IRQ0)

P2<sub>7</sub>/ $\overline{\text{IRQ}}_0$ 端子をP2<sub>7</sub>端子として使用するか、 $\overline{\text{IRQ}}_0$ 端子として使用するかを設定します。

ビット7		
IRQ0	説明	
0	P2 <sub>7</sub> 入力端子として機能	(初期値)
1	$\overline{\text{IRQ}}_0$ 入力端子として機能	

## ビット6：リザーブビット

リザーブビットです。本ビットはリードすると常に"1"が読み出されます。ライトは無効です。

ビット5：P2<sub>3</sub>/SO<sub>1</sub>端子PMOSコントロール(POF1)

P2<sub>3</sub>/SO<sub>1</sub>端子出力バッファのPMOSのON/OFFを制御します。

ビット5		
POF1	説明	
0	CMOS出力	(初期値)
1	NMOSオープンドレイン出力	

## ビット4：TMIGノイズキャンセラセレクト(NCS)

インプットキャプチャ入力信号(TMIG)のノイズ除去回路の制御を行います。

ビット4		
NCS	説明	
0	ノイズ除去機能なし	(初期値)
1	ノイズ除去機能あり	

ビット3：P2<sub>3</sub>/SO<sub>1</sub>端子機能切換え(SO1)

P2<sub>3</sub>/SO<sub>1</sub>端子をP2<sub>3</sub>端子として使用するか、SO<sub>1</sub>端子として使用するかを設定します。

ビット3		
SO1	説明	
0	P2 <sub>3</sub> 入出力端子として機能	(初期値)
1	SO <sub>1</sub> 出力端子として機能	

ビット2：P2<sub>2</sub>/SI<sub>1</sub>端子機能切換え(SI1)

P2<sub>2</sub>/SI<sub>1</sub>端子をP2<sub>2</sub>端子として使用するかSI<sub>1</sub>端子として使用するかを設定します。

ビット2	説明	
SI1		
0	P2 <sub>2</sub> 入出力端子として機能	(初期値)
1	SI <sub>1</sub> 入力端子として機能	

ビット1：P2<sub>1</sub>/SCK<sub>1</sub>端子機能切換え(SCK1)

P2<sub>1</sub>/SCK<sub>1</sub>端子をP2<sub>1</sub>端子として使用するか、SCK<sub>1</sub>端子として使用するかを設定します。

ビット1	説明	
SCK1		
0	P2 <sub>1</sub> 入出力端子として機能	(初期値)
1	SCK <sub>1</sub> 入出力端子として機能	

ビット0：P2<sub>0</sub>/IRQ<sub>4</sub>/ADTRG端子機能切換え(IRQ4)

P2<sub>0</sub>/IRQ<sub>4</sub>/ADTRG端子をP2<sub>0</sub>端子として使用するか、IRQ<sub>4</sub>/ADTRG端子として使用するかを設定します。

ビット0	説明	
IRQ4		
0	P2 <sub>0</sub> 入出力端子として機能	(初期値)
1	IRQ <sub>4</sub> /ADTRG入力端子として機能	

【注】 ADTRG端子の設定については、「12.3.2 外部トリガによるA/D変換器の起動」を参照してください。

## (4) ポートモードレジスタ6(PMR6)

ビット:	7	6	5	4	3	2	1	0
	-	-	-	-	-	TXD	-	-
初期値:	1	1	1	1	1	0	0	0
R/W :	-	-	R/W	R/W	-	R/W	R/W	R/W

## ビット7, 6, 3：リザーブビット

リザーブビットです。各ビットはリードすると常に"1"が読み出されます。ライトは無効です。

## ビット5, 4：リザーブビット

リザーブビットです。本ビットは"1"で使用してください。

ビット2：P2<sub>6</sub>/TXD 端子機能切換え（TXD）

P2<sub>6</sub>/TXD 端子を P2<sub>6</sub> 端子として使用するか、TXD 端子として使用するかを設定します。

ビット2	説明	
TXD		
0	P2 <sub>6</sub> 入出力端子として機能	(初期値)
1	TXD 出力端子として機能	

## ビット1, 0：リザーブビット

リザーブビットです。本ビットは"0"で使用してください。

## (5) ポートプルアップコントロールレジスタ2 (PUCR2)

ビット:	7	6	5	4	3	2	1	0
	PUCR2 <sub>7</sub>	PUCR2 <sub>6</sub>	PUCR2 <sub>5</sub>	PUCR2 <sub>4</sub>	PUCR2 <sub>3</sub>	PUCR2 <sub>2</sub>	PUCR2 <sub>1</sub>	PUCR2 <sub>0</sub>
初期値:	0	0	0	0	0	0	0	0
R/W :	R/W							

PUCR2 は、ポート 2 の各端子 P2<sub>7</sub> ~ P2<sub>0</sub> のプルアップ MOS をビットごとに制御します。

PUCR2 が"0"の状態では PUCR2 に"1"をセットすると対応するプルアップ MOS は ON 状態となり、"0"にクリアすると OFF 状態となります。

リセット時、PUCR2 は H'00 にイニシャライズされます。

【注】 P2<sub>7</sub> 端子は入力専用端子のため、PUCR2<sub>7</sub> が"1"にセットまたは"0"にクリアのいずれの場合でも、プルアップ MOS は OFF 状態となります。

## 8.3.3 端子機能

表 8.6 にポート 2 の端子機能を示します。

表 8.6 ポート 2 の端子機能 (1)

端子	選択方法と端子機能				
P2 <sub>7</sub> /IRQ <sub>0</sub>	PMR2 の IRQ0 と PCR2 の PCR2 <sub>7</sub> の組合せで、次のように切換わります。				
	IRQ0	0		1	
	PCR2 <sub>7</sub>	0	1	*	
	端子機能	P2 <sub>7</sub> 入力端子	ハイインピーダンス出力	IRQ <sub>0</sub> 入力端子	
P2 <sub>6</sub> /TXD	PMR6 の TXD と PCR2 の PCR2 <sub>6</sub> の組合せで、次のように切換わります。				
	TXD	0		1	
	PCR2 <sub>6</sub>	0	1	*	
	端子機能	P2 <sub>6</sub> 入力端子	P2 <sub>6</sub> 出力端子	TXD出力端子	
P2 <sub>5</sub> /RXD	SCI3 の SCR の RE と PCR2 の PCR2 <sub>5</sub> の組合せで、次のように切換わります。				
	RE	0		1	
	PCR2 <sub>5</sub>	0	1	*	
	端子機能	P2 <sub>5</sub> 入力端子	P2 <sub>5</sub> 出力端子	RXD入力端子	
P2 <sub>4</sub> /SCK <sub>3</sub>	SCI3 の SCR の CKE1、CKE0、SMR の COM、および PCR2 の PCR2 <sub>4</sub> の組合せで、次のように切換わります。				
	CKE1	0		1	
	CKE0	0	0	1	*
	COM	0	1	*	*
	PCR2 <sub>4</sub>	0	1	*	*
	端子機能	P2 <sub>4</sub> 入力端子	P2 <sub>4</sub> 出力端子	SCK <sub>3</sub> 出力端子	SCK <sub>3</sub> 入力端子
P2 <sub>3</sub> /SO <sub>1</sub>	PMR2 の SO1 と PCR2 の PCR2 <sub>3</sub> の組合せで、次のように切換わります。				
	SO1	0		1	
	PCR2 <sub>3</sub>	0	1	*	
	端子機能	P2 <sub>3</sub> 入力端子	P2 <sub>3</sub> 出力端子	SO <sub>1</sub> 入力端子	

表 8.6 ポート2の端子機能(2)

端子	選択方法と端子機能				
P2 <sub>2</sub> /SI <sub>1</sub>	PMR2のSI1とPCR2のPCR2 <sub>2</sub> の組合せで、次のように切り替わります。				
	SI1	0		1	
	PCR2 <sub>2</sub>	0	1	*	
	端子機能	P2 <sub>2</sub> 入力端子	P2 <sub>2</sub> 出力端子	SI <sub>1</sub> 入力端子	
P2 <sub>1</sub> /SCK <sub>1</sub>	PMR2のSCK1、SCR1のCKS3、およびPCR2のPCR2 <sub>1</sub> の組合せで、次のように切り替わります。				
	SCK1	0		1	
	CKS <sub>3</sub>	*		0	1
	PCR2 <sub>1</sub>	0	1	*	*
	端子機能	P2 <sub>1</sub> 入力端子	P2 <sub>1</sub> 出力端子	SCK <sub>1</sub> 出力端子	SCK <sub>1</sub> 入力端子
P2 <sub>0</sub> / $\overline{\text{IRQ}}_4$ $\overline{\text{ADTRG}}$	PMR2のIRQ4、AMRのTRGE、PCR2のPCR2 <sub>0</sub> の組合せで、次のように切り替わります。				
	IRQ4	0		1	
	PCR2 <sub>0</sub>	0	1	*	
	TRGE	*		0	1
	端子機能	P2 <sub>0</sub> 入力端子	P2 <sub>0</sub> 出力端子	$\overline{\text{IRQ}}_4$ 入力端子	$\overline{\text{IRQ}}_4/\overline{\text{ADTRG}}$ 入力端子
【注】 $\overline{\text{ADTRG}}$ 入力端子として使用する場合には、IENR1のIEN4を"0"にクリアしてIRQ4割込みを禁止してください。					

\* : Don't care

### 8.3.4 端子状態

各モードにおけるポート2の端子状態を表8.7に示します。

表 8.7 ポート2の端子状態

端子名	リセット	スリープ	サブスリープ	スタンバイ	ウォッチ	サブアクティブ	アクティブ
P2 <sub>7</sub> / $\overline{\text{IRQ}}_0$	ハイ インピー ダンス	ハイ インピー ダンス	ハイ インピー ダンス	ハイ インピー ダンス	ハイ インピー ダンス	ハイ インピー ダンス	ハイ インピー ダンス
P2 <sub>6</sub> /TXD P2 <sub>5</sub> /RXD P2 <sub>4</sub> /SCK <sub>3</sub> P2 <sub>3</sub> /SO <sub>1</sub> P2 <sub>2</sub> /SI <sub>1</sub> P2 <sub>1</sub> /SCK <sub>1</sub> P2 <sub>0</sub> / $\overline{\text{IRQ}}_4$ / $\overline{\text{ADTRG}}$	ハイ インピー ダンス	保持	保持	ハイ インピー ダンス*	保持	動作	動作

【注】 \* プルアップ MOS が ON 状態では “ High ” レベル出力となります。

### 8.3.5 入力プルアップ MOS

ポート 2 は、プログラムで制御可能な入力プルアップ MOS を内蔵しています。PCR2 が “0” にクリアされている状態で PUCR2 に “1” をセットすると入力プルアップ MOS は ON 状態となります。また、入力プルアップ MOS はリセット時、OFF になります。

PCR2 <sub>n</sub>	0		1
PUCR2 <sub>n</sub>	0	1	*
入力プルアップ MOS	OFF	ON	OFF

\* Don't care ( n=6 ~ 0 )

【注】 P2<sub>7</sub> 端子は入力専用端子のため、PUCR2<sub>7</sub> が “1” にセットまたは “0” にクリアのいずれの場合でも、プルアップ MOS は OFF 状態となります。

## 8.4 ポート 5

### 8.4.1 概要

ポート 5 は、8 ビットの入出力ポートです。ポート 5 の各端子は、図 8.3 に示す構成になっています。

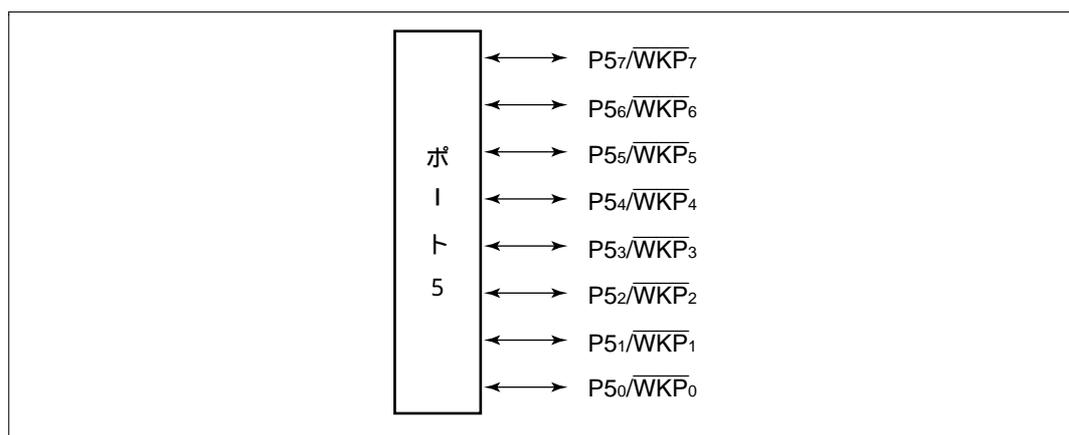


図 8.3 ポート 5 の端子構成

### 8.4.2 レジスタの構成と説明

表 8.8 にポート 5 のレジスタ構成を示します。

表 8.8 レジスタ構成

名称	略称	R/W	初期値	アドレス
ポートデータレジスタ 5	PDR5	R/W	H'00	H'FFD8
ポートコントロールレジスタ 5	PCR5	W	H'00	H'FFE8
ポートプルアップコントロールレジスタ 5	PUCR5	R/W	H'00	H'FF9E
ポートモードレジスタ 5	PMR5	R/W	H'00	H'FF9B

## (1) ポートデータレジスタ 5 (PDR5)

ビット:	7	6	5	4	3	2	1	0
	P57	P56	P55	P54	P53	P52	P51	P50
初期値:	0	0	0	0	0	0	0	0
R/W :	R/W							

PDR5 は、ポート 5 の各端子 P5<sub>7</sub> ~ P5<sub>0</sub> のデータを格納する 8 ビットのレジスタです。

PCR5 が "1" のとき、ポート 5 のリードを行うと、PDR5 の値を直接リードします。そのため端子状態の影響を受けません。PCR5 が "0" のとき、ポート 5 のリードを行うと、端子状態が読み出されます。

リセット時、PDR5 は H'00 にイニシャライズされます。

## (2) ポートコントロールレジスタ 5 (PCR5)

ビット:	7	6	5	4	3	2	1	0
	PCR57	PCR56	PCR55	PCR54	PCR53	PCR52	PCR51	PCR50
初期値:	0	0	0	0	0	0	0	0
R/W :	W	W	W	W	W	W	W	W

PCR5 は、ポート 5 の各端子 P5<sub>7</sub> ~ P5<sub>0</sub> の入出力をビットごとに制御します。PCR5 に "1" をセットすると対応する P5<sub>7</sub> ~ P5<sub>0</sub> 端子は出力端子となり、"0" にクリアすると入力端子となります。PMR5 により当該端子が汎用入出力に設定されている場合には、PCR5 および PDR5 の設定が有効となります。

リセット時、PCR5 は H'00 にイニシャライズされます。

本レジスタはライト専用ですが、リードした場合各ビットは常に "1" が読み出されます。

## (3) ポートプルアップコントロールレジスタ 5 (PUCR5)

ビット:	7	6	5	4	3	2	1	0
	PUCR57	PUCR56	PUCR55	PUCR54	PUCR53	PUCR52	PUCR51	PUCR50
初期値:	0	0	0	0	0	0	0	0
R/W :	R/W							

PUCR5 は、ポート 5 の各端子 P5<sub>7</sub> ~ P5<sub>0</sub> のプルアップ MOS をビットごとに制御します。

PCR5 が "0" の状態で PUCR5 に "1" をセットすると対応するプルアップ MOS は ON 状態となり、"0" にクリアすると OFF 状態となります。

リセット時、PUCR5 は H'00 にイニシャライズされます。

#### (4) ポートモードレジスタ 5 (PMR5)

ビット:	7	6	5	4	3	2	1	0
	WKP <sub>7</sub>	WKP <sub>6</sub>	WKP <sub>5</sub>	WKP <sub>4</sub>	WKP <sub>3</sub>	WKP <sub>2</sub>	WKP <sub>1</sub>	WKP <sub>0</sub>
初期値:	0	0	0	0	0	0	0	0
R/W :	R/W							

PMR5 は、8ビットのリード/ライト可能なレジスタで、ポート 5 の各端子機能の切換えを制御します。

リセット時、PMR5 は H'00 にイニシャライズされます。

ビット n : P5<sub>n</sub>/ $\overline{\text{WKP}}_n$  端子機能切換え (WKPn)

P5<sub>n</sub>/ $\overline{\text{WKP}}_n$  端子を P5<sub>n</sub> 端子として使用するか、 $\overline{\text{WKP}}_n$  端子として使用するかを設定します。

ビット n	説明
0	P5 <sub>n</sub> 入出力端子として機能 (初期値)
1	$\overline{\text{WKP}}_n$ 入力端子として機能

(n = 7 ~ 0)

### 8.4.3 端子機能

表 8.9 にポート 5 の端子機能を示します。

表 8.9 ポート 5 の端子機能

端子	選択方法と端子機能												
P5 <sub>7</sub> / $\overline{\text{WKP}}_7$ ~ P5 <sub>0</sub> / $\overline{\text{WKP}}_0$	PMR5 の WKP <sub>n</sub> と PCR5 の PCR5 <sub>n</sub> の組合せで、次のように切換わります。 (n = 7 ~ 0)												
	<table border="1"> <thead> <tr> <th>WKP<sub>n</sub></th> <th colspan="2">0</th> <th>1</th> </tr> </thead> <tbody> <tr> <th>PCR5<sub>n</sub></th> <td>0</td> <td>1</td> <td>*</td> </tr> <tr> <th>端子機能</th> <td>P5<sub>n</sub>入力端子</td> <td>P5<sub>n</sub>出力端子</td> <td><math>\overline{\text{WKP}}_n</math>入力端子</td> </tr> </tbody> </table>	WKP <sub>n</sub>	0		1	PCR5 <sub>n</sub>	0	1	*	端子機能	P5 <sub>n</sub> 入力端子	P5 <sub>n</sub> 出力端子	$\overline{\text{WKP}}_n$ 入力端子
WKP <sub>n</sub>	0		1										
PCR5 <sub>n</sub>	0	1	*										
端子機能	P5 <sub>n</sub> 入力端子	P5 <sub>n</sub> 出力端子	$\overline{\text{WKP}}_n$ 入力端子										

\* : Don't care

### 8.4.4 端子状態

各動作モードにおけるポート5の端子状態を表8.10に示します。

表 8.10 ポート5の端子状態

端子名	リセット	スリープ	サブスリープ	スタンバイ	ウォッチ	サブアクティブ	アクティブ
P5 <sub>7</sub> / $\overline{\text{WKP}}_7$ ~ P5 <sub>0</sub> / $\overline{\text{WKP}}_0$	ハイ インピー ダンス	保持	保持	ハイ インピー ダンス*	保持	動作	動作

【注】 \* ブルアップMOSがON状態では"High"レベル出力となります。

### 8.4.5 入力ブルアップMOS

ポート5は、プログラムで制御可能な入力ブルアップMOSを内蔵しています。PCR5が"0"にクリアされている状態でPUCR5に"1"をセットすると入力ブルアップMOSはON状態となります。また、入力ブルアップMOSはリセット時、OFF状態になります。

PCR5 <sub>n</sub>	0		1
PUCR5 <sub>n</sub>	0	1	*
入力ブルアップMOS	OFF	ON	OFF

\* : Don't care (n = 7 ~ 0)

## 8.5 ポート 6

### 8.5.1 概要

ポート 6 は、8 ビットの入出力ポートです。ポート 6 の各端子は、図 8.4 に示す構成になっています。

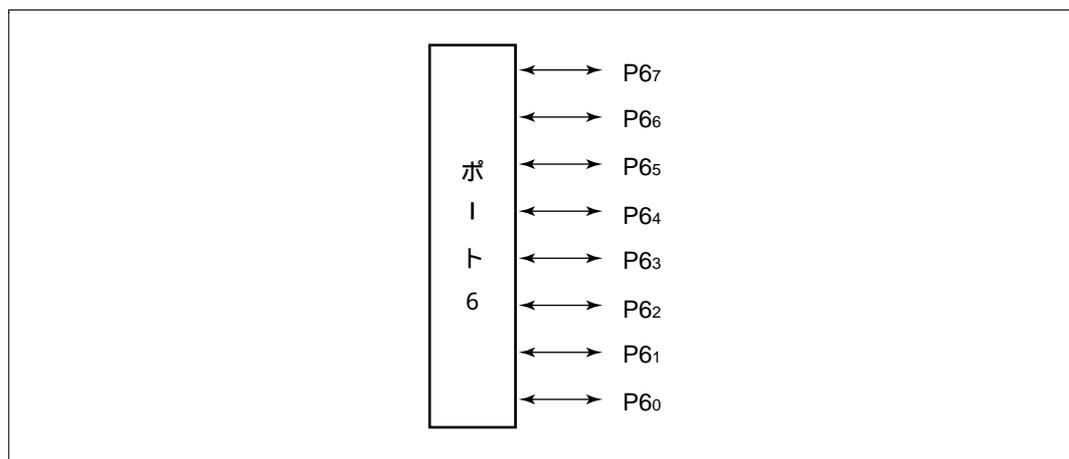


図 8.4 ポート 6 の端子構成

### 8.5.2 レジスタの構成と説明

表 8.11 にポート 6 のレジスタ構成を示します。

表 8.11 レジスタ構成

名称	略称	R/W	初期値	アドレス
ポートデータレジスタ 6	PDR6	R/W	H'00	H'FFD9
ポートコントロールレジスタ 6	PCR6	W	H'00	H'FFE9
ポートブルアップコントロールレジスタ 6	PUCR6	R/W	H'00	H'FF9F

## (1) ポートデータレジスタ 6 (PDR6)

ビット:	7	6	5	4	3	2	1	0
	P67	P66	P65	P64	P63	P62	P61	P60
初期値:	0	0	0	0	0	0	0	0
R/W :	R/W							

PDR6は、ポート6の各端子P6<sub>7</sub>~P6<sub>0</sub>のデータを格納する8ビットのレジスタです。

PCR6が"1"のとき、ポート6のリードを行うと、PDR6の値を直接リードします。そのため端子状態の影響を受けません。PCR6が"0"のとき、ポート6のリードを行うと、端子状態が読み出されます。

リセット時、PDR6はH'00にイニシャライズされます。

## (2) ポートコントロールレジスタ 6 (PCR6)

ビット:	7	6	5	4	3	2	1	0
	PCR67	PCR66	PCR65	PCR64	PCR63	PCR62	PCR61	PCR60
初期値:	0	0	0	0	0	0	0	0
R/W :	W	W	W	W	W	W	W	W

PCR6は、ポート6の各端子P6<sub>7</sub>~P6<sub>0</sub>の入出力をビットごとに制御します。PCR6に"1"をセットすると対応するP6<sub>7</sub>~P6<sub>0</sub>端子は出力端子となり、"0"にクリアすると入力端子となります。

リセット時、PCR6はH'00にイニシャライズされます。

本レジスタはライト専用ですが、リードした場合各ビットは常に"1"が読み出されます。

## (3) ポートプルアップコントロールレジスタ 6 (PUCR6)

ビット:	7	6	5	4	3	2	1	0
	PUCR67	PUCR66	PUCR65	PUCR64	PUCR63	PUCR62	PUCR61	PUCR60
初期値:	0	0	0	0	0	0	0	0
R/W :	R/W							

PUCR6は、ポート6の各端子P6<sub>7</sub>~P6<sub>0</sub>のプルアップMOSをビットごとに制御します。

PCR6が"0"の状態でもPUCR6に"1"をセットすると対応するプルアップMOSはON状態となり、"0"にクリアするとOFF状態となります。

リセット時、PUCR6はH'00にイニシャライズされます。

### 8.5.3 端子機能

表 8.12 にポート 6 の端子機能を示します。

表 8.12 ポート 6 の端子機能

端子	選択方法と端子機能	
P6 <sub>7</sub> ~ P6 <sub>0</sub>	PCR6 の PCR6 <sub>n</sub> により、次のように切り換わります。	
	(n = 7 ~ 0)	
	PCR6 <sub>n</sub>	0
端子機能	P6 <sub>n</sub> 入力端子	P6 <sub>n</sub> 出力端子

### 8.5.4 端子状態

各動作モードにおけるポート 6 の端子状態を表 8.13 に示します。

表 8.13 ポート 6 の端子状態

端子名	リセット	スリープ	サブスリープ	スタンバイ	ウォッチ	サブアクティブ	アクティブ
P6 <sub>7</sub> ~ P6 <sub>0</sub>	ハイ インピー ダンス	保持	保持	ハイ インピー ダンス*	保持	動作	動作

【注】 \* ブルアップ MOS が ON 状態では"High"レベル出力となります。

### 8.5.5 入力プルアップ MOS

ポート 6 は、プログラムで制御可能な入力プルアップ MOS を内蔵しています。PCR6 が"0"にクリアされている状態で PUCR6 に"1"をセットすると入力プルアップ MOS は ON 状態となります。また、入力プルアップ MOS はリセット時、OFF 状態になります。

PCR6 <sub>n</sub>	0		1
PUCR6 <sub>n</sub>	0	1	*
入力プルアップ MOS	OFF	ON	OFF

\* : Don't care (n = 7 ~ 0)

## 8.6 ポート7

### 8.6.1 概要

ポート7は、8ビットの入出力ポートです。ポート7の各端子は、図8.5に示す構成になっています。

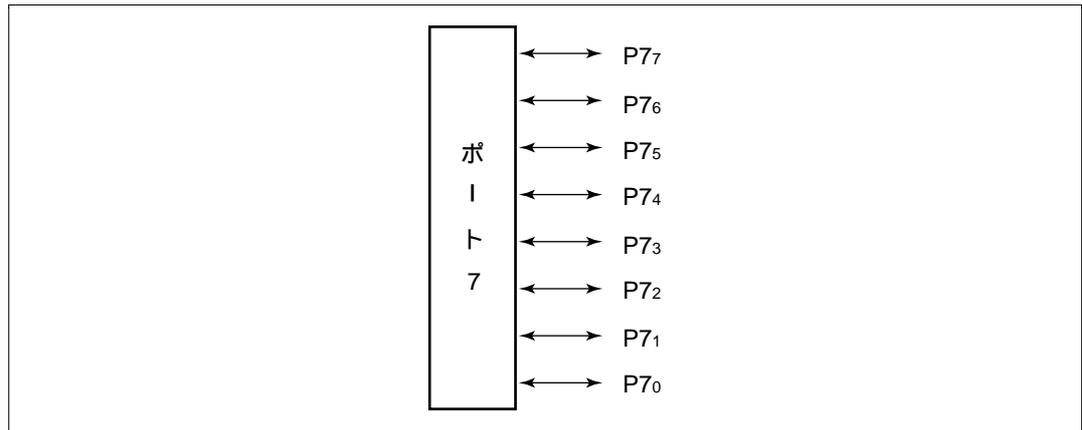


図8.5 ポート7の端子構成

### 8.6.2 レジスタの構成と説明

表8.14にポート7のレジスタ構成を示します。

表8.14 レジスタ構成

名称	略称	R/W	初期値	アドレス
ポートデータレジスタ7	PDR7	R/W	H'00	H'FFDA
ポートコントロールレジスタ7	PCR7	W	H'00	H'FFEA

## (1) ポートデータレジスタ7 (PDR7)

ビット:	7	6	5	4	3	2	1	0
	P7 <sub>7</sub>	P7 <sub>6</sub>	P7 <sub>5</sub>	P7 <sub>4</sub>	P7 <sub>3</sub>	P7 <sub>2</sub>	P7 <sub>1</sub>	P7 <sub>0</sub>
初期値:	0	0	0	0	0	0	0	0
R/W :	R/W							

PDR7は、ポート7の各端子P7<sub>7</sub>~P7<sub>0</sub>のデータを格納する8ビットのレジスタです。

PCR7が"1"のとき、ポート7のリードを行うと、PDR7の値を直接リードします。そのため端子状態の影響を受けません。PCR7が"0"のとき、ポート7のリードを行うと、端子状態が読み出されます。

リセット時、PDR7はH'00にイニシャライズされます。

## (2) ポートコントロールレジスタ7 (PCR7)

ビット:	7	6	5	4	3	2	1	0
	PCR7 <sub>7</sub>	PCR7 <sub>6</sub>	PCR7 <sub>5</sub>	PCR7 <sub>4</sub>	PCR7 <sub>3</sub>	PCR7 <sub>2</sub>	PCR7 <sub>1</sub>	PCR7 <sub>0</sub>
初期値:	0	0	0	0	0	0	0	0
R/W :	W	W	W	W	W	W	W	W

PCR7は、ポート7の各端子P7<sub>7</sub>~P7<sub>0</sub>の入出力をビットごとに制御します。PCR7に"1"をセットすると対応するP7<sub>7</sub>~P7<sub>0</sub>端子は出力端子となり、"0"にクリアすると入力端子となります。

リセット時、PCR7はH'00にイニシャライズされます。

本レジスタはライト専用ですが、リードした場合各ビットは常に"1"が読み出されます。

### 8.6.3 端子機能

表 8.15 にポート 7 の端子機能を示します。

表 8.15 ポート 7 の端子機能

端子	選択方法と端子機能					
P7 <sub>7</sub> ~ P7 <sub>0</sub>	PCR7 の PCR7 <sub>n</sub> により、次のように切り替わります。					
	(n = 7 ~ 0)					
	<table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>PCR7<sub>n</sub></th> <th>0</th> <th>1</th> </tr> </thead> <tbody> <tr> <td>端子機能</td> <td>P7<sub>n</sub>入力端子</td> <td>P7<sub>n</sub>出力端子</td> </tr> </tbody> </table>	PCR7 <sub>n</sub>	0	1	端子機能	P7 <sub>n</sub> 入力端子
PCR7 <sub>n</sub>	0	1				
端子機能	P7 <sub>n</sub> 入力端子	P7 <sub>n</sub> 出力端子				

### 8.6.4 端子状態

各動作モードにおけるポート 7 の端子状態を表 8.16 に示します。

表 8.16 ポート 7 の端子状態

端子名	リセット	スリープ	サブスリープ	スタンバイ	ウォッチ	サブアクティブ	アクティブ
P7 <sub>7</sub> ~ P7 <sub>0</sub>	ハイ インピー ダンス	保持	保持	ハイ インピー ダンス	保持	動作	動作

## 8.7 ポート 8

### 8.7.1 概要

ポート 8 は、8 ビットの入出力ポートです。ポート 8 の各端子は、図 8.6 に示す構成になっています。

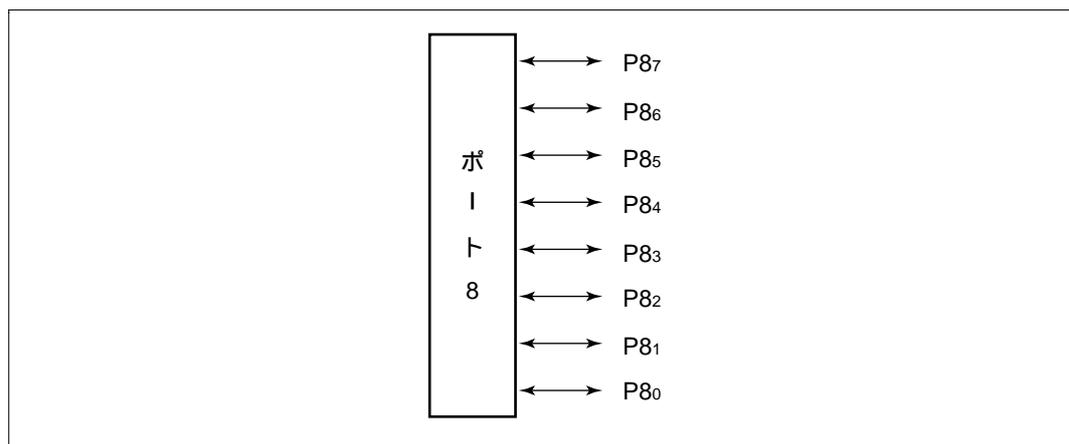


図 8.6 ポート 8 の端子構成

### 8.7.2 レジスタの構成と説明

表 8.17 にポート 8 のレジスタ構成を示します。

表 8.17 レジスタ構成

名称	略称	R/W	初期値	アドレス
ポートデータレジスタ 8	PDR8	R/W	H'00	H'FFDB
ポートコントロールレジスタ 8	PCR8	W	H'00	H'FFEB

## (1) ポートデータレジスタ 8 (PDR8)

ビット:	7	6	5	4	3	2	1	0
	P87	P86	P85	P84	P83	P82	P81	P80
初期値:	0	0	0	0	0	0	0	0
R/W :	R/W							

PDR8は、ポート8の各端子P8<sub>7</sub>~P8<sub>0</sub>のデータを格納する8ビットのレジスタです。

PCR8が"1"のとき、ポート8のリードを行うと、PDR8の値を直接リードします。そのため端子状態の影響を受けません。PCR8が"0"のとき、ポート8のリードを行うと、端子状態が読み出されます。

リセット時、PDR8はH'00にイニシャライズされます。

## (2) ポートコントロールレジスタ 8 (PCR8)

ビット:	7	6	5	4	3	2	1	0
	PCR87	PCR86	PCR85	PCR84	PCR83	PCR82	PCR81	PCR80
初期値:	0	0	0	0	0	0	0	0
R/W :	W	W	W	W	W	W	W	W

PCR8は、ポート8の各端子P8<sub>7</sub>~P8<sub>0</sub>の入出力をビットごとに制御します。PCR8に"1"をセットすると対応するP8<sub>7</sub>~P8<sub>0</sub>端子は出力端子となり、"0"にクリアすると入力端子となります。

リセット時、PCR8はH'00にイニシャライズされます。

本レジスタはライト専用ですが、リードした場合各ビットは常に"1"が読み出されます。

### 8.7.3 端子機能

表 8.18 にポート 8 の端子機能を示します。

表 8.18 ポート 8 の端子機能

端子	選択方法と端子機能						
P8 <sub>7</sub> ~ P8 <sub>0</sub>	PCR8 の PCR8 <sub>n</sub> により、次のように切り替わります。 <div style="text-align: right;">(n = 7 ~ 0)</div> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>PCR8<sub>n</sub></th> <th>0</th> <th>1</th> </tr> </thead> <tbody> <tr> <td>端子機能</td> <td>P8<sub>n</sub>入力端子</td> <td>P8<sub>n</sub>出力端子</td> </tr> </tbody> </table>	PCR8 <sub>n</sub>	0	1	端子機能	P8 <sub>n</sub> 入力端子	P8 <sub>n</sub> 出力端子
PCR8 <sub>n</sub>	0	1					
端子機能	P8 <sub>n</sub> 入力端子	P8 <sub>n</sub> 出力端子					

### 8.7.4 端子状態

各動作モードにおけるポート 8 の端子状態を表 8.19 に示します。

表 8.19 ポート 8 の端子状態

端子名	リセット	スリープ	サブスリープ	スタンバイ	ウォッチ	サブアクティブ	アクティブ
P8 <sub>7</sub> ~ P8 <sub>0</sub>	ハイ インピー ダンス	保持	保持	ハイ インピー ダンス	保持	動作	動作

## 8.8 ポート A

### 8.8.1 概要

ポート A は、3 ビットの入出力ポートです。ポート A の各端子は、図 8.7 に示す構成になっています。

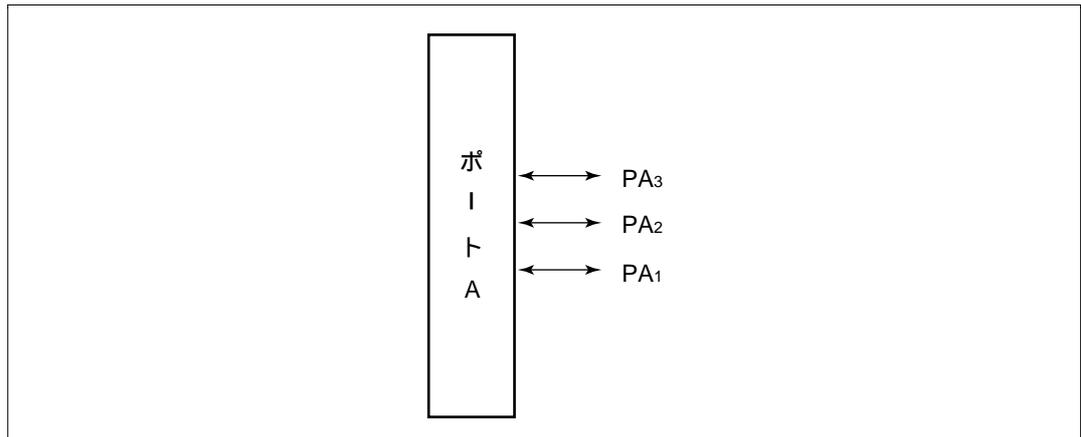


図 8.7 ポート A の端子構成

### 8.8.2 レジスタの構成と説明

表 8.20 にポート A のレジスタ構成を示します。

表 8.20 レジスタ構成

名称	略称	R/W	初期値	アドレス
ポートデータレジスタ A	PDRA	R/W	H'F0	H'FFDD
ポートコントロールレジスタ A	PCRA	W	H'F1	H'FFED

#### (1) ポートデータレジスタ A (PDRA)

ビット:	7	6	5	4	3	2	1	0
	—	—	—	—	PA <sub>3</sub>	PA <sub>2</sub>	PA <sub>1</sub>	—
初期値:	1	1	1	1	0	0	0	0
R/W :	—	—	—	—	R/W	R/W	R/W	—

PDRA は、ポート A の各端子 PA<sub>3</sub> ~ PA<sub>1</sub> のデータを格納する 8 ビットのレジスタです。

PCRA が "1" のとき、ポート A のリードを行うと、PDRA の値を直接リードします。そのため端子状態の影響を受けません。PCRA が "0" のとき、ポート A のリードを行うと、端子状態が読み出されます。

リセット時、PDRA は H'F0 にイニシャライズされます。

## (2) ポートコントロールレジスタ A (PCRA)

ビット:	7	6	5	4	3	2	1	0
	—	—	—	—	PCRA3	PCRA2	PCRA1	—
初期値:	1	1	1	1	0	0	0	1
R/W :	—	—	—	—	W	W	W	—

PCRA は、ポート A の各端子 PA<sub>3</sub> ~ PA<sub>1</sub> の入出力をビットごとに制御します。PCRA に"1" をセットすると対応する PA<sub>3</sub> ~ PA<sub>1</sub> 端子は出力端子となり、"0"にクリアすると入力端子となります。

リセット時、PCRA は HF1 にイニシャライズされます。

本レジスタはライト専用ですが、リードした場合各ビットは常に"1"が読み出されます。

### 8.8.3 端子機能

表 8.21 にポート A の端子機能を示します。

表 8.21 ポート A の端子機能

端子	選択方法と端子機能									
PA <sub>3</sub> ~ PA <sub>1</sub>	PCRA の PCRA <sub>n</sub> により、次のように切替わります。  <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td colspan="3" style="text-align: right;">(n = 3 ~ 1)</td> </tr> <tr> <td>PCRA<sub>n</sub></td> <td>0</td> <td>1</td> </tr> <tr> <td>端子機能</td> <td>PA<sub>n</sub>入力端子</td> <td>PA<sub>n</sub>出力端子</td> </tr> </table>	(n = 3 ~ 1)			PCRA <sub>n</sub>	0	1	端子機能	PA <sub>n</sub> 入力端子	PA <sub>n</sub> 出力端子
(n = 3 ~ 1)										
PCRA <sub>n</sub>	0	1								
端子機能	PA <sub>n</sub> 入力端子	PA <sub>n</sub> 出力端子								

### 8.8.4 端子状態

各動作モードにおけるポート A の端子状態を表 8.22 に示します。

表 8.22 ポート A の端子状態

端子名	リセット	スリープ	サブスリープ	スタンバイ	ウォッチ	サブアクティブ	アクティブ
PA <sub>3</sub> ~ PA <sub>1</sub>	ハイ インピー ダンス	保持	保持	ハイ インピー ダンス	保持	動作	動作

## 8.9 ポート B

### 8.9.1 概要

ポート B は、2 ビットの入力専用ポートです。ポート B の各端子は、図 8.8 に示す構成になっています。

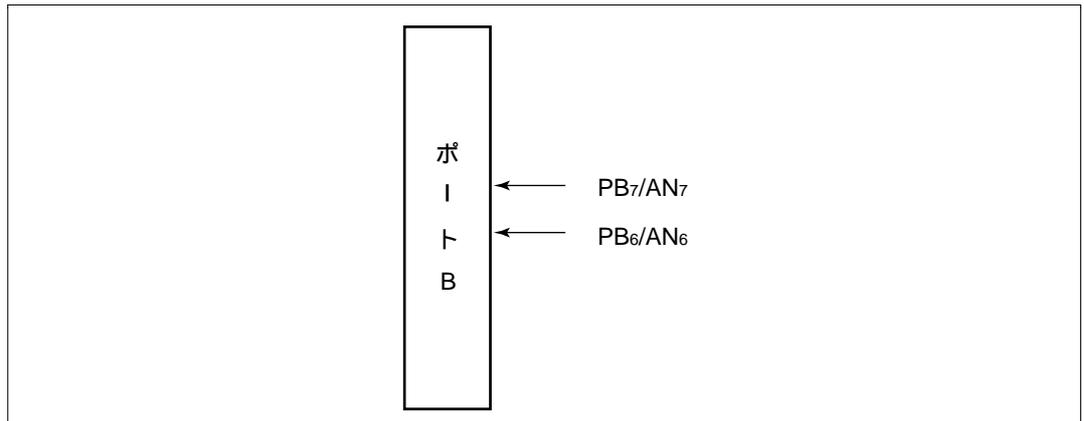


図 8.8 ポート B の端子構成

### 8.9.2 レジスタの構成と説明

表 8.23 にポート B のレジスタ構成を示します。

表 8.23 レジスタ構成

名称	略称	R/W	アドレス
ポートデータレジスタ B	PDRB	R	H'FFDE

#### (1) ポートデータレジスタ B (PDRB)

ビット:	7	6	5	4	3	2	1	0
	PB7	PB6	—	—	—	—	—	—
R/W :	R	R	—	—	—	—	—	—

PDRB をリードすると常に各端子の状態が読み出されます。ただし、A/D 変換器の AMR の CH3 ~ CH0 によりアナログ入力チャンネルが選択されている端子をリードすると入力電圧に関係なく "0" が読み出されます。

---

# 9. タイマ

---

## 第9章 目次

9.1	概要	171
9.2	タイマ A	172
	9.2.1 概要	172
	9.2.2 各レジスタの説明	174
	9.2.3 動作説明	176
	9.2.4 タイマ A の動作モード	177
9.3	タイマ F	178
	9.3.1 概要	178
	9.3.2 各レジスタの説明	181
	9.3.3 CPU とのインタフェース	188
	9.3.4 動作説明	190
	9.3.5 使用上の注意事項	193
9.4	タイマ G	195
	9.4.1 概要	195
	9.4.2 各レジスタの説明	197
	9.4.3 ノイズ除去回路	201
	9.4.4 動作説明	202
	9.4.5 使用上の注意事項	207
	9.4.6 タイマ G の使用例	211



## 9.1 概要

本LSIは3本のタイマ（タイマA、F、G）を内蔵しています。

タイマA、F、Gの機能概要を表9.1に示します。

表9.1 タイマの機能概要

タイマ 名称	機能		内部クロック	イベント 入力端子	波形 出力端子	備考
タイマA	8ビット のタイマ	・インターバル機能	$/8 \sim /8192$ (8種類)			
		・時計用タイムベース機能	$w/128$ (オーバーフロー周期は4種類選択可)			
		・クロック出力機能	$/4 \sim /32$ $w/4 \sim w/32$ (8種類)			
タイマF	<ul style="list-style-type: none"> <li>・16ビットフリーランニングタイマ</li> <li>・イベントカウント機能</li> <li>・独立した2本の8ビットタイマとして使用可能</li> <li>・アウトプットコンペア出力機能</li> </ul>		$/2 \sim /32$ (4種類)	TMIF	TMOFL TMOFH	
タイマG	<ul style="list-style-type: none"> <li>・8ビットのタイマ</li> <li>・インプットキャプチャ機能</li> <li>・インターバル機能</li> </ul>		$/2 \sim /64$ $w/2$ (4種類)	TMIG		<ul style="list-style-type: none"> <li>・カウンタのクリア指定が可能</li> <li>・キャプチャ入力信号のノイズ除去回路内蔵</li> </ul>

## 9.2 タイマ A

### 9.2.1 概要

タイマ A はインターバル / 時計用タイムベース機能を内蔵した 8 ビットのタイマです。32.768kHz の水晶発振器を接続すると時計用タイムベースとして使用できます。また、TMOW 端子より、32.768kHz を分周したクロック、およびシステムクロックを分周したクロックが出力可能です。

#### (1) 特長

タイマ A の特長を以下に示します。

8 種類の内部クロックを選択可能

8 種類の内部クロック ( /8192、 /4096、 /2048、 /512、 /256、 /128、 /32、 /8 ) からの選択が可能です。

4 種類のオーバフロー周期を選択可能

時計用タイムベースとして 4 種類のオーバフロー周期 ( 1s、0.5s、0.25s、31.25ms ) の選択が可能です ( 32.768kHz 水晶発振器を使用 ) 。

カウンタのオーバフローで割込みを発生

タイマ出力クロックを選択可能

TMOW 端子から出力するクロックとして、32.768kHz の 32、16、8、4 分周したクロック ( 1kHz、2kHz、4kHz、8kHz )、およびシステムクロックを 32、16、8、4 分周したクロックの計 8 種類の選択が可能です。

## (2) ブロック図

タイマ A のブロック図を図 9.1 に示します。

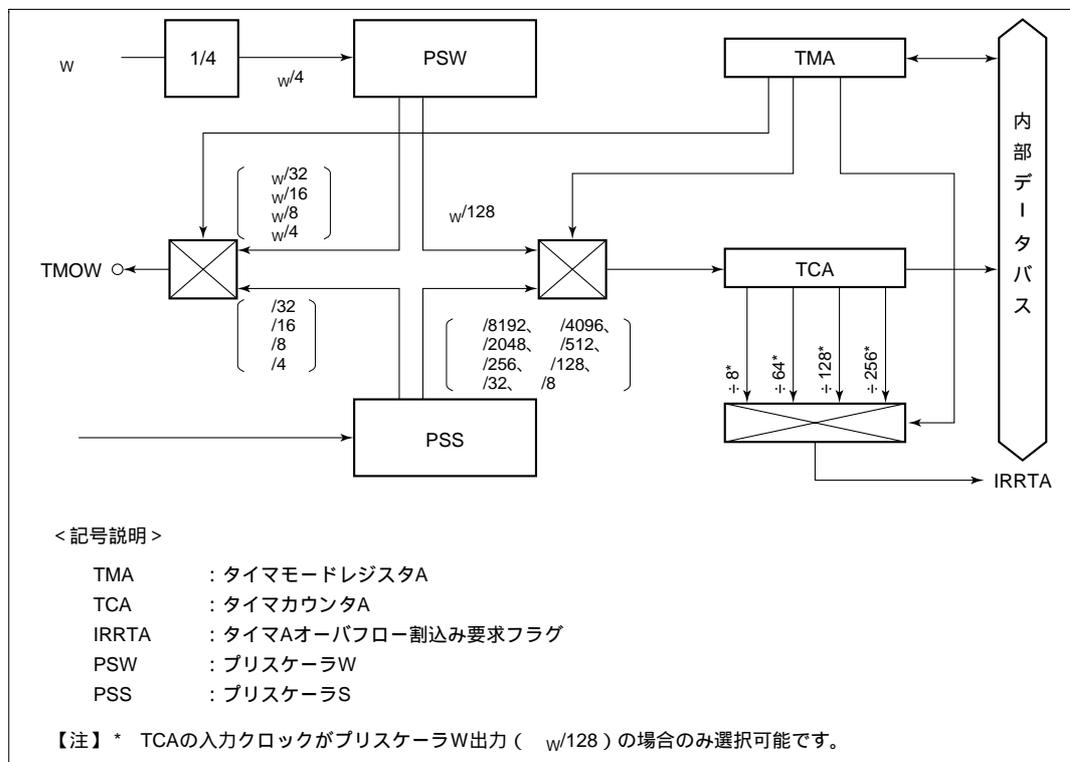


図 9.1 タイマ A ブロック図

## (3) 端子構成

タイマ A の端子構成を表 9.2 に示します。

表 9.2 端子構成

名称	略称	入出力	機能
クロック出力	TMOW	出力	タイマ A 出力回路により生成された波形の出力端子

## (4) レジスタ構成

タイマ A のレジスタ構成を表 9.3 に示します。

表 9.3 レジスタ構成

名称	略称	R/W	初期値	アドレス
タイマモードレジスタ A	TMA	R/W	H'10	H'FFB0
タイマカウンタ A	TCA	R	H'00	H'FFB1

## 9.2.2 各レジスタの説明

## (1) タイマモードレジスタ A (TMA)

ビット:	7	6	5	4	3	2	1	0
	TMA7	TMA6	TMA5	—	TMA3	TMA2	TMA1	TMA0
初期値:	0	0	0	1	0	0	0	0
R/W :	R/W	R/W	R/W	—	R/W	R/W	R/W	R/W

TMA は、8 ビットのリード/ライト可能なレジスタで、プリスケアラ、入力クロック、および出力クロックの選択を行います。

リセット時、TMA は H'10 にイニシャライズされます。

## ビット 7 ~ 5 : クロック出力セレクト (TMA7 ~ TMA5)

TMOW 端子から出力する 8 種類のクロックを選択します。システムクロックを 32、16、8、4 分周したクロックは、アクティブモード、スリープモードで出力されます。32.768kHz を 32、16、8、4 分周したクロックは、アクティブモード、スリープモード、およびサブアクティブモードで出力されます。

ビット 7	ビット 6	ビット 5	説明
TMA7	TMA6	TMA5	
0	0	0	/32 (初期値)
		1	/16
	1	0	/8
		1	/4
1	0	0	$\frac{w}{32}$
		1	$\frac{w}{16}$
	1	0	$\frac{w}{8}$
		1	$\frac{w}{4}$

## ビット4：リザーブビット

リザーブビットです。本ビットはリードすると常に"1"が読み出されます。ライトは無効です。

## ビット3～0：内部クロックセレクト（TMA3～TMA0）

TCAに入力するクロックを選択します。

ビット3	ビット2	ビット1	ビット0	説明	
TMA3	TMA2	TMA1	TMA0	プリスケラ分周比またはオーバフロー周期	機能
0	0	0	0	PSS、 /8192	インターバル       時計用 タイムベース
			1	PSS、 /4096	
		1	0	PSS、 /2048	
			1	PSS、 /512	
	1	0	0	PSS、 /256	
			1	PSS、 /128	
		1	0	PSS、 /32	
			1	PSS、 /8	
1	0	0	0	PSW、 1s	
			1	PSW、 0.5s	
		1	0	PSW、 0.25s	
			1	PSW、 0.03125s	
	1	0	0	PSW、 TCA リセット	
			1		
		1	0		
			1		

## (2) タイマカウンタ A (TCA)

ビット：	7	6	5	4	3	2	1	0
	TCA7	TCA6	TCA5	TCA4	TCA3	TCA2	TCA1	TCA0
初期値：	0	0	0	0	0	0	0	0
R/W：	R	R	R	R	R	R	R	R

TCAは、8ビットのリード可能なアップカウンタで、入力する内部クロックによりカウントアップされます。入力するクロックはTMAのTMA3～TMA0により選択します。TCAの値は、アクティブモード時はCPUからリードできますが、サブアクティブモード時ではTCAをリードすることはできません。TCAがオーバフローすると、IRR1のIRRRTAが"1"にセットされます。

TCAはTMAのTMA3～TMA2を"11"にセットすることでクリアできます。

リセット時、TCAはH'00にイニシャライズされます。

## 9.2.3 動作説明

### (1) インターバル動作

TMA の TMA3 を "0" にセットすると、タイマ A は 8 ビットインターバルタイマとして動作します。

リセット時、TCA は H'00、TMA3 は "0" にクリアされるため、リセット直後はインターバルタイマとして停止することなくカウントアップを続けます。タイマ A の動作クロックは、TMA の TMA2 ~ TMA0 により、プリスケータ S の出力する 8 種類の内部クロックを選択できます。

TCA のカウンタ値が H'FF になった後、クロックが入力されると、タイマ A はオーバーフローし、IRR1 の IRRTA が "1" にセットされます。このとき、IENR1 の IENTA が "1" ならば CPU に割込みを要求します。\*

オーバーフロー時には、TCA のカウンタ値は H'00 に戻り、再びカウントアップを開始します。したがって、256 回の入力クロックごとに、オーバーフロー出力を発生するインターバルタイマとして動作します。

【注】 \* 割込みについての詳細は、「3.3 割込み」を参照してください。

### (2) 時計用タイムベース動作

TMA の TMA3 を "1" にセットすると、タイマ A はプリスケータ W の出力するクロックをカウントして、時計用タイムベースとして動作します。タイマ A のオーバーフロー周期は、TMA の TMA1、TMA0 により 4 種類選択できます。時計用タイムベース動作時 (TMA3 = "1") に TMA2 を "1" にすると、TCA およびプリスケータ W は、ともに H'00 にクリアされます。

### (3) クロック出力の動作

PMR1 の TMOW を "1" にセットすると、TMOW 端子からクロックが出力されます。端子から出力されるクロックは、TMA の TMA7 ~ TMA5 により、8 種類のクロックが選択できます。システムクロックを 32、16、8、4 分周したクロックは、アクティブモード、スリープモードで出力され、32.768kHz を 32、16、8、4 分周したクロックは、アクティブモード、スリープモード、およびサブアクティブモードで出力されます。

## 9.2.4 タイマ A の動作モード

タイマ A の動作モードを表 9.4 に示します。

表 9.4 タイマ A の動作モード

	動作モード	リセット	アクティブ	スリープ	ウォッチ	サブアクティブ	サブスリープ	スタンバイ
TCA	インターバル	リセット	動作	動作	停止	停止	停止	停止
	時計用 タイムベース	リセット	動作	動作	動作	動作	動作	停止
	TMA	リセット	動作	保持	保持	動作	保持	保持

【注】 アクティブモード、スリープモード時に、TCA の内部クロックとして時計用タイムベース機能を選択した場合、システムクロックと内部クロックとが非同期であるため同期回路で同期をとっています。これにより、カウント周期は最大で  $1/$  (s) の誤差が生じます。

## 9.3 タイマ F

### 9.3.1 概要

タイマ F は、アウトプットコンペア機能を内蔵した 16 ビットのタイマです。外部イベントのカウントが可能のほか、コンペアマッチ信号によりカウンタのリセット、割込み要求、トグル出力などが可能です。また、2本の独立した 8 ビットタイマ（タイマ FH、タイマ FL）としても動作可能です。

#### (1) 特長

タイマ F の特長を以下に示します。

##### 5 種類のクロックを選択可能

4 種類の内部クロック（ /32、 /16、 /4、 /2 ）と、外部クロックからの選択が可能です（外部イベントのカウントが可能）。

##### トグル出力機能

1 本のコンペアマッチ信号により、TMOFH 端子にトグル出力します（トグル出力の初期値を設定可能）。

##### コンペアマッチ信号によるカウンタリセット

割込み要因：コンペアマッチ×1 要因、オーバフロー×1 要因

##### 2本の独立した 8 ビットタイマとして動作可能

2本の独立した 8 ビットタイマ（タイマ FH、タイマ FL）としても動作が可能です（8 ビットモード時）。

	タイマ FH 8 ビットタイマ*	タイマ FL 8 ビットタイマ / イベントカウンタ
内部クロック	4 種類（ /32、 /16、 /4、 /2 ）	
イベント入力	TMIF 端子	
トグル出力	コンペアマッチ信号×1 本 TMOFH 端子に出力 （初期値を設定可能）	コンペアマッチ信号×1 本 出力 TMOFL 端子に出力 （初期値を設定可能）
カウンタリセット	コンペアマッチ信号により、カウンタのリセットが可能	
割込み要因	コンペアマッチ×1 要因 オーバフロー×1 要因	

【注】 \* 16 ビットタイマとして動作する場合はタイマ FL のオーバフロー信号により動作します。

(2) ブロック図

タイマFのブロック図を図9.2に示します。

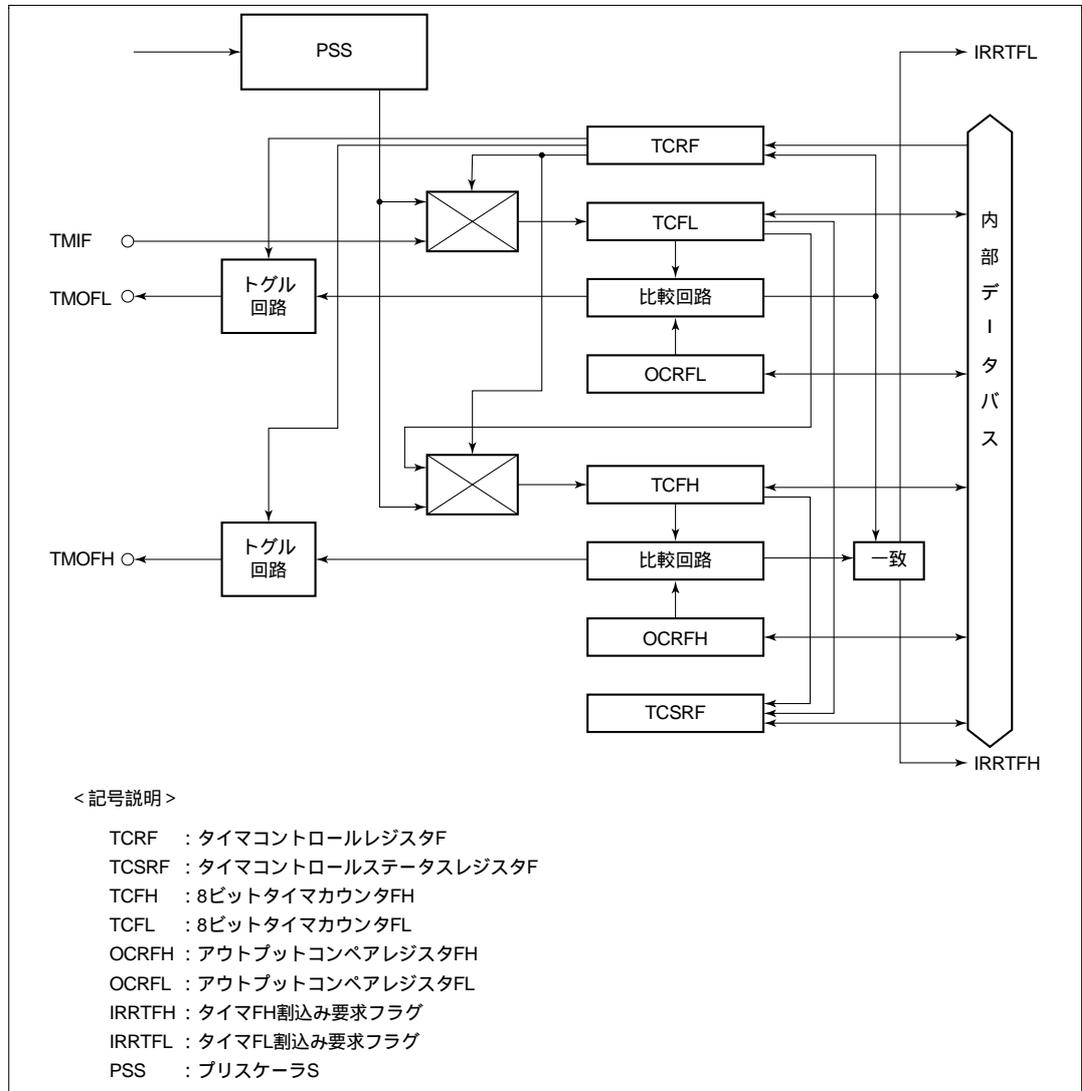


図9.2 タイマFのブロック図

## (3) 端子構成

タイマFの端子構成を表9.5に示します。

表9.5 端子構成

名称	略称	入出力	機能
タイマFイベント入力	TMIF	入力	TCFLに入力するイベント入力端子
タイマFH出力	TMOFH	出力	タイマFHトグル出力端子
タイマFL出力	TMOFL	出力	タイマFLトグル出力端子

## (4) レジスタ構成

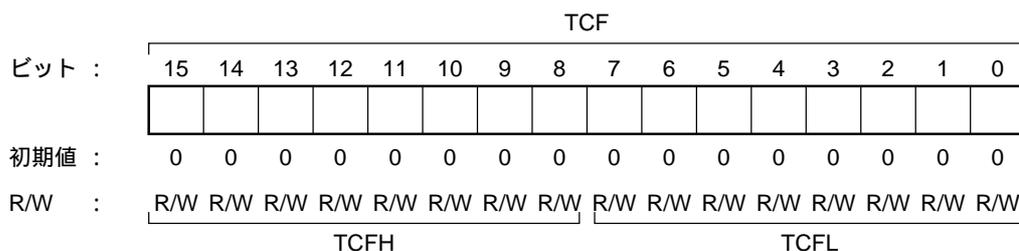
タイマFのレジスタ構成を表9.6に示します。

表9.6 レジスタ構成

名称	略称	R/W	初期値	アドレス
タイマコントロールレジスタF	TCRF	W	H'00	H'FFB6
タイマコントロールステータスレジスタF	TCSRF	R/W	H'00	H'FFB7
8ビットタイマカウンタFH	TCFH	R/W	H'00	H'FFB8
8ビットタイマカウンタFL	TCFL	R/W	H'00	H'FFB9
アウトプットコンペアレジスタFH	OCRFH	R/W	H'FF	H'FFBA
アウトプットコンペアレジスタFL	OCRFL	R/W	H'FF	H'FFBB

### 9.3.2 各レジスタの説明

- (1) 16 ビットタイマカウンタ (TCF)  
 8 ビットタイマカウンタ (TCFH)  
 8 ビットタイマカウンタ (TCFL)



TCFは16ビットのリード/ライト可能なアップカウンタで、8ビットのタイマカウンタ (TCFH、TCFL) のカスケード接続により構成されています。上位8ビットをTCFH、下位8ビットをTCFLとする16ビットカウンタとして使用できるほか、TCFH、TCFLを独立した8ビットカウンタとして使用することもできます。

TCFH、TCFLは、CPUからリード/ライト可能ですが、16ビットモードで使用する場合、CPUとのデータ転送はテンポラリレジスタ (TEMP) を介して行われます。TEMPの詳細は「9.3.3 CPUとのインタフェース」を参照してください。

リセット時、TCFH、TCFLは各々H'00にイニシャライズされます。

#### (a) 16ビットモード (TCF)

TCRFのCKSH2を"0"に設定すると、TCFは16ビットカウンタとして動作します。TCFの入力クロックは、TCRFのCKSL2~CKSL0により選択します。

TCSRfのCCLRfにより、コンペアマッチ時にTCFをクリアすることができます。

TCFがオーバーフロー (H'FFFF H'0000) すると、TCSRfのOVFfが"1"にセットされます。このときTCSRfのOVIEHが"1"の場合、IRR2のIRRTFHが"1"にセットされ、さらにIENR2のIENTFHが"1"ならばCPUに割り込みを要求します。

#### (b) 8ビットモード (TCFL/TCFH)

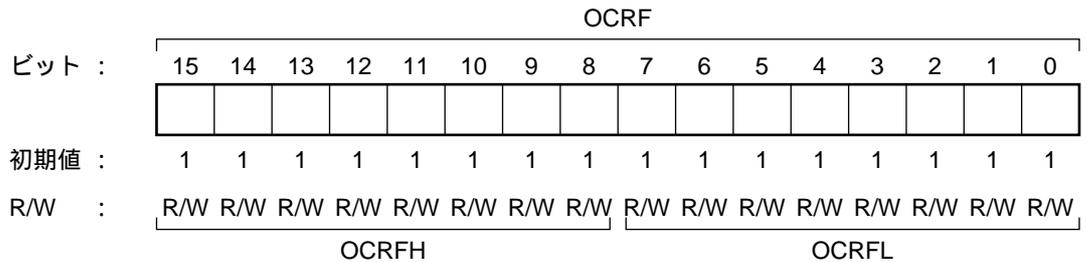
TCRFのCKSH2を"1"に設定すると、TCFH、TCFLは2本の独立した8ビットカウンタとして動作します。TCFH (TCFL) の入力クロックは、TCRFのCKSH2~CKSH0 (CKSL2~CKSL0) により選択します。

TCSRfのCCLRf (CCLRL) により、コンペアマッチ時にTCFH (TCFL) をクリアすることができます。

TCFH (TCFL) がオーバーフロー (H'FF H'00) すると、TCSRfのOVFf (OVFL) が"1"にセットされます。このときTCSRfのOVIEH (OVIEL) が"1"の場合、IRR2のIRRTFH

(IRR2) が"1"にセットされ、さらに IENR2 の IENTFH (IENTFL) が"1"ならば CPU に割込みを要求します。

- (2) 16 ビットアウトプットコンペアレジスタ (OCRF)  
 8 ビットアウトプットコンペアレジスタ (OCRFH)  
 8 ビットアウトプットコンペアレジスタ (OCRFL)



OCRFは16ビットのリード/ライト可能な2本のレジスタ(OCRFH、OCRFL)により構成されています。上位8ビットをOCRFH、下位8ビットをOCRFLとする16ビットレジスタとして使用できるほか、OCRFH、OCRFLを独立した8ビットレジスタとして使用することもできます。

OCRFH、OCRFLは、CPUからリード/ライト可能ですが、16ビットモードで使用する場合、CPUとのデータ転送はテンポラリレジスタ(TEMP)を介して行われます。TEMPの詳細は「9.3.3 CPUとのインタフェース」を参照してください。

リセット時、OCRFH、OCRFLは各々H'FFにイニシャライズされます。

#### (a) 16ビットモード(OCRF)

TCRFのCKSH2を"0"に設定すると、OCRFは16ビットレジスタとして動作します。OCRFの内容は、TCFと常に比較されており、両者の値が一致すると、TCSRのCMFHが"1"にセットされます。また、同時にIRR2のIRRTFHも"1"にセットされます。このときIENR2のIENTFHが"1"ならばCPUに割込みを要求します。

コンペアマッチによるトグル出力をTMOFH端子より出力することができます。また、TCRFのTOLHにより、出力レベル("High"/"Low")の設定が可能です。

#### (b) 8ビットモード(OCRFH/OCRFL)

TCRFのCKSH2を"1"に設定すると、OCRFは2本の8ビットレジスタとして動作します。OCRFHの内容はTCFHと、またOCRFLの内容はTCFLとそれぞれ個別に比較されます。OCRFH(OCRFL)とTCFH(TCFL)の値が一致すると、TCSRのCMFH(CMFL)が"1"にセットされます。また、同時にIRR2のIRRTFH(IRRTFL)も"1"にセットされます。このとき、IENR2のIENTFH(IENTFL)が"1"ならばCPUに割込みを要求します。

コンペアマッチによるトグル出力をTMOFH端子(TMOFL端子)より出力することができます。また、TCRFのTOLH(TOLL)により、出力レベル("High"/"Low")の設定

が可能です。

### (3) タイマコントロールレジスタ F (TCRF)

ビット:	7	6	5	4	3	2	1	0
	TOLH	CKSH2	CKSH1	CKSH0	TOLL	CKSL2	CKSL1	CKSL0
初期値:	0	0	0	0	0	0	0	0
R/W :	W	W	W	W	W	W	W	W

TCRF は、8 ビットのライト専用のレジスタです。16 ビットモード、8 ビットモードの切換え、4 種類の内部クロックおよび外部イベントの選択、TMOFH、TMOFL 端子の出力レベルの設定を行います。

リセット時、TCRF は H'00 にイニシャライズされます。

#### ビット7: トグルアウトプットレベルH (TOLH)

TMOFH 端子の出力レベルを設定します。出力レベルは、ライトした直後反映されます。

ビット7	説明	
TOLH		
0	"Low" レベル	(初期値)
1	"High" レベル	

#### ビット6~4: クロックセレクトH (CKSH2~CKSH0)

TCFH に入力するクロックを内部4種類、または TCFL のオーバフローから選択します。

ビット6	ビット5	ビット4	説明
CKSH2	CKSH1	CKSH0	
0	*	*	16 ビットモードとなり、TCFL のオーバフロー信号でカウント (初期値)
1	0	0	内部クロック: /32 でカウント
1	0	1	内部クロック: /16 でカウント
1	1	0	内部クロック: /4 でカウント
1	1	1	内部クロック: /2 でカウント

\* : Don't care

## ビット3：トグルアウトプットレベルL (TOLL)

TMOFL 端子の出力レベルを設定します。出力レベルは、ライトした直後反映されます。

ビット3		
TOLL	説明	
0	"Low" レベル	(初期値)
1	"High" レベル	

## ビット2～0：クロックセレクトL (CKSL2～CKSL0)

TCFL に入力するクロックを、内部4種類または外部イベントから選択します。

ビット2	ビット1	ビット0	説明
CKSL2	CKSL1	CKSL0	
0	*	*	外部イベント (TMIF) の立上がり / 立下がりエッジでカウント* (初期値)
1	0	0	内部クロック： /32 でカウント
1	0	1	内部クロック： /16 でカウント
1	1	0	内部クロック： /4 でカウント
1	1	1	内部クロック： /2 でカウント

\* : Don't care

【注】 \* 外部イベントのエッジ選択は、IRQ エッジセレクトレジスタ (IEGR) の IEG3 により設定します。詳細は、「3.3.2 (1) IRQ エッジセレクトレジスタ (IEGR)」を参照してください。

なお、TMIF 端子の機能切換えのために TMIF 端子が "Low" レベルの状態ポートモードレジスタ 1 (PMR1) の IRQ3 を "0" から "1" または "1" から "0" に設定変更した場合に、タイマ F のカウンタがカウントアップされることがありますので注意してください。

## (4) タイマコントロールステータスレジスタ F (TCSR F)

ビット:	7	6	5	4	3	2	1	0
	OVFH	CMFH	OVIEH	CCLR H	OVFL	CMFL	OVIEL	CCLR L
初期値:	0	0	0	0	0	0	0	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCSR F は、8 ビットのリード/ライト可能なレジスタです。カウンタクリアの選択、オーバーフローフラグのセット、コンペアマッチフラグのセット、オーバーフローによる割込み要求の許可の制御を行います。

リセット時、TCSR F は H'00 にイニシャライズされます。

## ビット7: タイマオーバーフローフラグ H (OVFH)

OVFH がオーバーフロー (H'FF H'00) したことを示すステータスフラグです。本フラグは、ソフトウェアでクリアします。セットは、ハードウェアで行われます。ソフトウェアでセットすることはできません。

ビット7	
OVFH	説明
0	〔クリア条件〕 (初期値) OVFH = "1"の状態、OVFH をリードした後、OVFH に"0"をライトしたとき
1	〔セット条件〕 OVFH の値が、H'FF H'00 になったとき

## ビット6: コンペアマッチフラグ H (CMFH)

OVFH と OCRFH がコンペアマッチしたことを示すステータスフラグです。本フラグは、ソフトウェアでクリアします。セットは、ハードウェアで行われます。ソフトウェアでセットすることはできません。

ビット6	
CMFH	説明
0	〔クリア条件〕 (初期値) CMFH = "1"の状態、CMFH をリードした後、CMFH に"0"をライトしたとき
1	〔セット条件〕 OVFH の値と OCRFH の値が、コンペアマッチしたとき

## ビット5：タイマオーバーフローインタラプトイネーブルH (OVIEH)

TCFH のオーバーフローが発生したとき、割込みの許可または禁止を選択します。

ビット5		
OVIEH	説明	
0	TCFH のオーバーフローによる割込み要求を禁止	(初期値)
1	TCFH のオーバーフローによる割込み要求を許可	

## ビット4：カウンタクリアH (CCLRH)

16 ビットモード時、TCF と OCRF がコンペアマッチしたとき、TCF をクリアするかしないかを選択します。

8 ビットモード時、TCFH と OCRFH がコンペアマッチしたとき、TCFH をクリアするかしないかを選択します。

ビット4		
CCLRH	説明	
0	16 ビットモード：コンペアマッチによる TCF のクリアを禁止 8 ビットモード：コンペアマッチによる TCFH のクリアを禁止	(初期値)
1	16 ビットモード：コンペアマッチによる TCF のクリアを許可 8 ビットモード：コンペアマッチによる TCFH のクリアを許可	

## ビット3：タイマオーバーフローフラグL (OVFL)

TCFL がオーバーフロー (H'FF H'00) したことを示すステータスフラグです。本フラグは、ソフトウェアでクリアします。セットは、ハードウェアで行われます。ソフトウェアでセットすることはできません。

ビット3		
OVFL	説明	
0	[クリア条件] OVFL = "1"の状態、OVFL をリードした後、OVFL に"0"をライトしたとき	(初期値)
1	[セット条件] TCFL の値が、H'FF H'00 になったとき	

## ビット2：コンペアマッチフラグL (CMFL)

TCFLとOCRFLがコンペアマッチしたことを示すステータスフラグです。本フラグは、ソフトウェアでクリアします。セットは、ハードウェアで行われます。ソフトウェアでセットすることはできません。

ビット2	
CMFL	説明
0	〔クリア条件〕 CMFL="1"の状態、CMFLをリードした後、CMFLに"0"をライトしたとき (初期値)
1	〔セット条件〕 TCFLの値とOCRFLの値が、コンペアマッチしたとき

## ビット1：タイマオーバーフローインタラプトイネーブルL (OVIEL)

TCFLのオーバーフローが発生したとき、割込みの許可または禁止を選択します。

ビット1	
OVIEL	説明
0	TCFLのオーバーフローによる割込み要求を禁止 (初期値)
1	TCFLのオーバーフローによる割込み要求を許可

## ビット0：カウンタクリアL (CCLRL)

TCFLとOCRFLがコンペアマッチしたとき、TCFLをクリアするかしないかを選択します。

ビット0	
CCLRL	説明
0	コンペアマッチによるTCFLのクリアを禁止 (初期値)
1	コンペアマッチによるTCFLのクリアを許可

### 9.3.3 CPU とのインタフェース

TCF、OCRFB は 16 ビットのリード/ライト可能なレジスタで構成されています。一方、CPU と内蔵周辺モジュール間のデータバスは、8 ビット幅となっています。したがって CPU が TCF、OCRFB をアクセスする場合、8 ビットのテンポラリレジスタ (TEMP) を介して行います。

16 ビットモードで TCF のリード/ライト、OCRFB のライトを行うときは、必ず 16 ビット単位 (バイトサイズの MOV 命令を 2 回連続して行う) で行い、かつ上位バイト、下位バイトの順序で行います。上位バイトのみや下位バイトのみのアクセスでは、データは正しく転送されません。

なお、8 ビットモードでは特にアクセスの順序に制限はありません。

#### (1) ライト時の動作

上位バイトのライトにより、上位バイトのデータが TEMP に転送されます。

次に下位バイトのライトで、TEMP にあるデータが上位バイトのレジスタへ、下位バイトのデータは直接下位バイトのレジスタへライトされます。

TCF に H'AA55 をライトするときの TCF のライト動作を図 9.3 に示します。

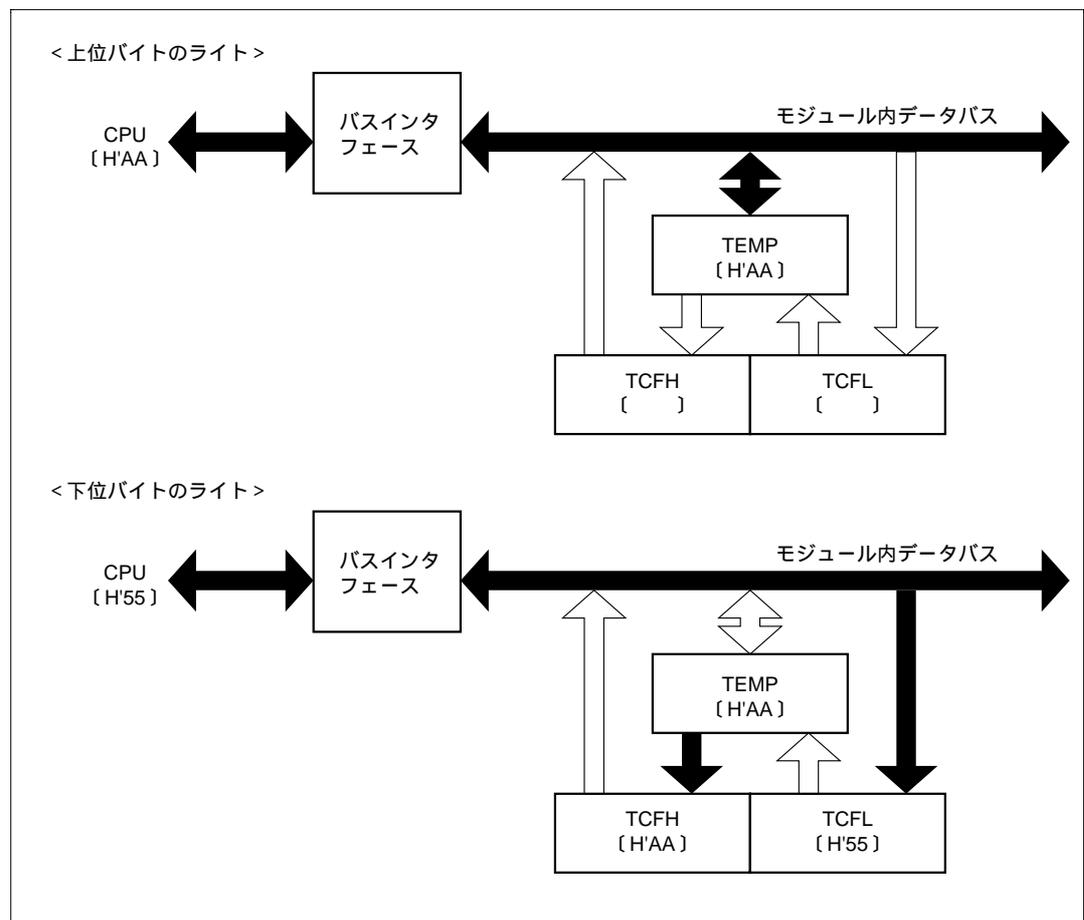


図 9.3 TCF のライト動作 (CPU TCF)

## (2) リード時の動作

TCFの場合、上位バイトのリードで、上位バイトのデータは直接 CPU に転送され、下位バイトのデータは TEMP に転送されます。

次に下位バイトのリードで、TEMP にある下位バイトのデータが CPU に転送されます。OCRf の場合、上位バイトのリードで、上位バイトのデータは直接 CPU に転送されます。下位バイトのリードで、下位バイトのデータは直接 CPU に転送されます。

H'AAFF である TCF をリードしたときの TCF のリード動作を図 9.4 に示します。

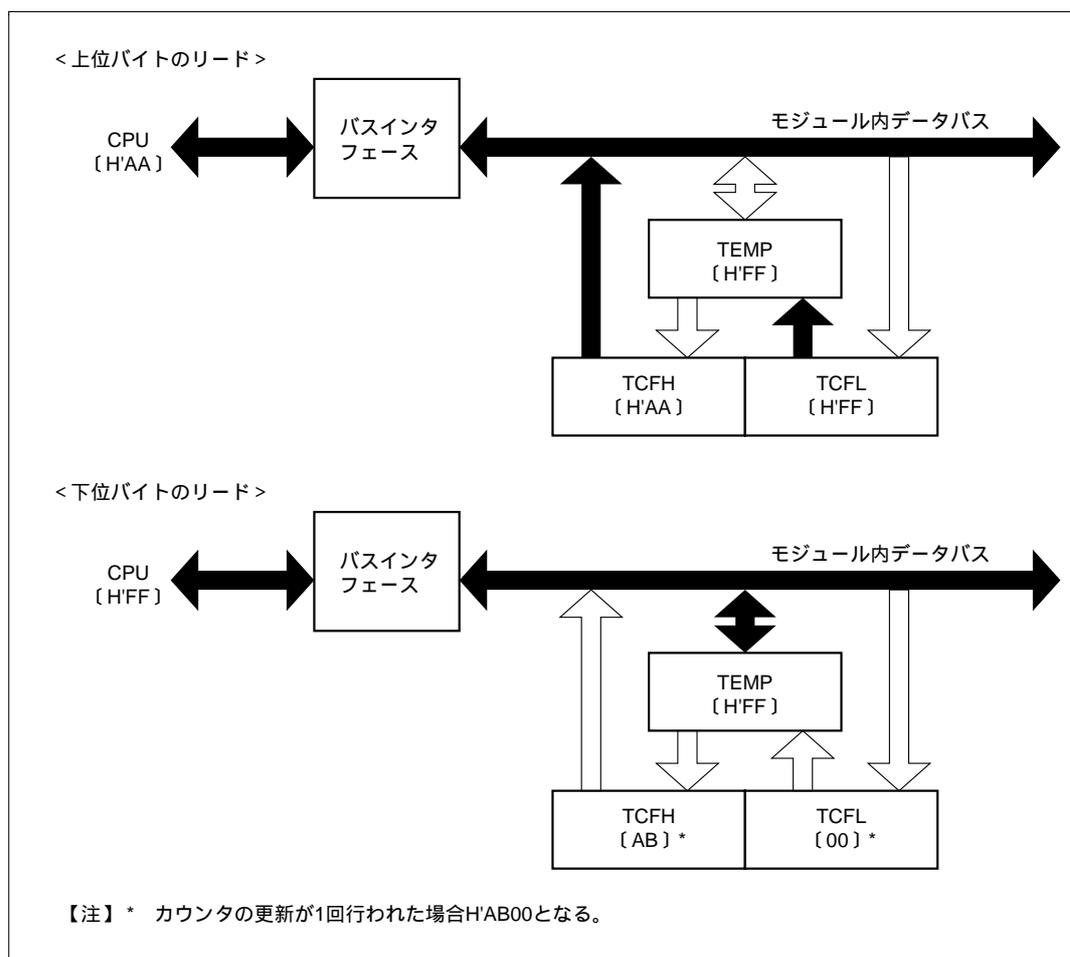


図 9.4 TCF のリード動作 (TCF CPU)

### 9.3.4 動作説明

タイマ F は、入力クロックが入るたびにカウントアップする 16 ビットのカウンタで、アウトプットコンペアレジスタ F に設定した値とタイマカウンタ F の値を常に比較しており、一致したタイミングでカウンタのクリア、割込み要求、およびポートのトグル出力が可能です。また、2 本の独立した 8 ビットタイマとしても機能できます。

#### (1) タイマ F の動作

タイマ F には、16 ビットタイマモードと 8 ビットタイマモードの 2 種類の動作モードがあります。

それぞれのモードの動作について以下に説明します。

##### (a) 16 ビットタイマモードの動作

タイマコントロールレジスタ F (TCRF) の CKSH2 ビットを "0" に設定すると、タイマ F は 16 ビットのタイマとして動作します。

リセット直後、タイマカウンタ F (TCF) は H'0000 に、アウトプットコンペアレジスタ F (OCRF) は H'FFFF に、タイマコントロールレジスタ F (TCRF)、タイマコントロールステータスレジスタ F (TCSR) は H'00 に初期化されます。カウンタは、外部イベント (TMIF) からの入力によりカウントアップを開始します。外部イベントのエッジ選択は、IRQ エッジセレクトレジスタ (IEGR) の IEG3 により設定します。

タイマ F の動作クロックは、TCRF の CKSL2 ~ CKSL0 ビットによりプリスケアラ S の出力する 4 種類の内部クロック、または外部クロックを選択できます。

TCF と OCRF の内容は常に比較されており、両者が一致すると TCSR の CMFH が "1" にセットされます。このとき IENR2 の IENTFH が "1" ならば CPU に割込みを要求し、同時に TMOFH 端子の出力をトグルします。また、TCSR の CCLR が "1" ならば TCF をクリアします。なお、TMOFH 端子の出力は TCRF の TOLH により設定できます。

TCF がオーバフロー (H'FFFF H'0000) すると、TCSR の OVFH がセットされます。このとき、TCSR の OVIEH と IENR2 の IENTFH が共に "1" ならば CPU に割込みを要求します。

##### (b) 8 ビットタイマモードの動作

TCRF の CKSH2 を "1" に設定すると、TCF は TCFH、TCFL の 2 本の独立した 8 ビットタイマとして動作します。TCFH/TCFL の入力クロックは、TCRF の CKSH2 ~ CKSH0/CKSL2 ~ CKSL0 により選定します。

OCRFH/OCRFL と TCFH/TCFL の値が一致すると、TCSR の CMFH/CMFL が "1" にセットされます。また、IENR2 の IENTFH/IENTFL が "1" ならば CPU に割込みを要求し、同時に TMOFH 端子/TMOFL 端子の出力をトグルします。また、TCRF の CCLR/CCLR が "1" ならば、TCFH/TCFL をクリアします。なお、TMOFH 端子/TMOFL 端子の出力は、TCRF

の TOLH/TOLL により設定できます。

TCFH/TCFL がオーバフロー (H'FF H'00) すると、TCSR の OVFH/OVFL が "1" にセットされます。このとき、TCSR の OVIEH/OVIEL と IENR2 の IENTFH/IENTFL が共に "1" ならば、CPU に割り込みを要求します。

## (2) TCF のカウントタイミング

TCF は、入力されたクロック (内部クロックまたは外部イベント) によりカウントアップされます。

### (a) 内部クロック動作の場合

TCRF の CKSH2 ~ CKSH0 または CKSL2 ~ CKSL0 の設定により、システムクロック ( ) を分周して作られる 4 種類の内部クロック ( /32、 /16、 /4、 /2 ) が選択されます。

### (b) 外部イベント動作の場合

TCRF の CKSL2 を "0" に設定することにより、外部イベント入力を選択されます。外部イベントは立上がり / 立下がりエッジのどちらでもカウントアップが可能です。外部イベントのエッジ選択は、割り込みコントローラの IEGR の IEG3 で設定します。なお、外部イベントのパルス幅は 2 システムクロック ( ) 以上必要です。これ以下のパルス幅では正しく動作しませんので注意してください。

## (3) TMOFH、TMOFL 出力タイミング

TMOFH、TMOFL の出力は、TCRF の TOLH、TOLL で設定した値が出力されていますが、コンペアマッチが発生したとき出力が反転します。

図 9.5 に出力タイミングを示します。

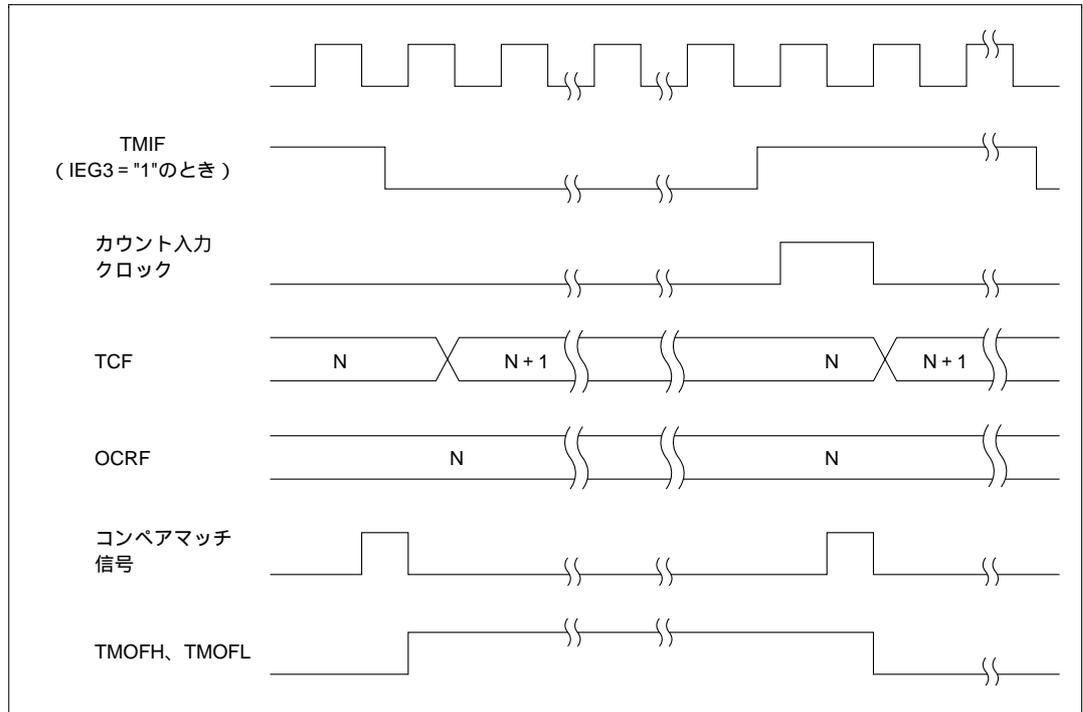


図 9.5 TMOFH、TMOFL 出力タイミング

#### (4) TCF のクリアタイミング

TCF は、OCRF とのコンペアマッチ時にクリアすることができます。

#### (5) タイマオーバフローフラグ (OVF) のセットタイミング

OVF は、TCF がオーバフロー (H'FFFF H'0000) したとき "1" にセットされます。

#### (6) コンペアマッチフラグのセットタイミング

コンペアマッチフラグ (CMFH または CMFL) は、TCF と OCRF のコンペアマッチ時に "1" にセットされます。コンペアマッチ信号は、値が一致した最後のステート (TCF が一致したカウント値を更新するタイミング) で発生します。TCF と OCRF が一致した後、カウントアップクロックが発生するまで、コンペアマッチ信号は発生しません。

## (7) タイマFの動作モード

タイマFの動作モードを表9.7に示します。

表9.7 タイマFの動作モード

動作モード	リセット	アクティブ	スリープ	ウォッチ	サブアクティブ	サブスリープ	スタンバイ
TCF	リセット	動作	動作	停止	停止	停止	停止
OCRF	リセット	動作	保持	保持	保持	保持	保持
TCRF	リセット	動作	保持	保持	保持	保持	保持
TCSRFB	リセット	動作	保持	保持	保持	保持	保持

### 9.3.5 使用上の注意事項

タイマFの動作中、次のような競合や動作が起こりますので注意してください。

#### (1) 16ビットモード

トグル出力は16ビットすべてが一致し、コンペアマッチ信号が発生したときTMOFH端子から出力されます。MOV命令によるTCRFのライトと、コンペアマッチ信号が同時に起こった場合、TCRFのライトによるTOLHのデータがTMOFH端子に出力されます。TMOFL端子は16ビットモード時には出力は不定となりますので使用しないでください。ポートとしてご使用ください。

OCRFのライトと、コンペアマッチ信号の発生が同時に起こった場合、コンペアマッチ信号は無効になります。ただし、ライトしたデータとカウンタ値がコンペアマッチする場合は、その時点でコンペアマッチ信号が発生します。コンペアマッチ信号は、TCFLのクロックに同期して出力されるので、クロックが停止している場合、コンペアマッチしていてもコンペアマッチ信号は発生しません。

コンペアマッチフラグは、16ビットすべてが一致し、コンペアマッチ信号が発生したときCMFHにセットされますが、CMFLについても下位8ビットについてのセット条件が満たされていればセットされます。

TCFがオーバーフローするとOVFHがセットされますが、OVFLについても下位8ビットがオーバーフローした時点で、セット条件が満たされていればセットされます。TCFLのライトと、オーバーフロー信号の出力が同時に起こった場合、オーバーフロー信号は出力されません。

## (2) 8ビットモード

### (a) TCFH、OCRFH

トグル出力は、コンペアマッチ時に TMOFH 端子から出力されます。MOV 命令による TCRF のライトと、コンペアマッチ信号の発生が同時に起こった場合、TCRF のライトによる TOLH のデータが TMOFH 端子に出力されます。

OCRFH のライトと、コンペアマッチ信号の発生が同時に起こった場合、コンペアマッチ信号は無効になります。ただし、ライトしたデータとカウンタ値がコンペアマッチする場合は、その時点でコンペアマッチ信号が発生します。コンペアマッチ信号は、TCFH のクロックに同期して出力されます。

TCFH のライトと、オーバフロー信号の出力が同時に起こった場合、オーバフロー信号は出力されません。

### (b) TCFL、OCRFL

トグル出力は、コンペアマッチ時に TMOFL 端子から出力されます。MOV 命令による TCRF のライトと、コンペアマッチ信号の発生が同時に起こった場合、TCRF のライトによる TOLL のデータが TMOFL 端子に出力されます。

OCRFL のライトと、コンペアマッチ信号の発生が同時に起こった場合、コンペアマッチ信号は無効になります。ただし、ライトしたデータとカウンタ値がコンペアマッチする場合は、その時点でコンペアマッチ信号が発生します。コンペアマッチ信号は、TCFL のクロックに同期して出力されるので、クロックが停止している場合、コンペアマッチしていてもコンペアマッチ信号は発生しません。

TCFL のライトと、オーバフロー信号の出力が同時に起こった場合、オーバフロー信号は出力されません。

## 9.4 タイマ G

### 9.4.1 概要

タイマ G は、インプットキャプチャ/インターバル機能を内蔵した 8 ビットのタイマです。インプットキャプチャ入力端子から入力したパルス（インプットキャプチャ入力信号）の立上がり/立下がりエッジに対し、それぞれ専用のインプットキャプチャ機能をもっています。また、ノイズ除去回路によりインプットキャプチャ入力信号の高域成分のノイズを除去することが可能です。したがって、インプットキャプチャ入力信号のデューティ比を正確に測定することができます。さらに、タイマ G はインプットキャプチャ入力を設定しない場合、8 ビットのインターバルタイマとして動作します。

#### (1) 特長

タイマ G の特長を以下に示します。

##### 4 種類のカウンタ入力クロックを選択可能

4 種類の内部クロック（ $f/64$ 、 $f/32$ 、 $f/2$ 、 $f_w/2$ ）からの選択が可能です。

##### インプットキャプチャ機能

立上がり/立下がりエッジのそれぞれ専用のインプットキャプチャ機能があります。

##### カウンタのオーバフロー時のレベルが検出可能

インプットキャプチャ入力信号の "High" レベルでオーバフローが発生したか、"Low" レベルでオーバフローが発生したかを検出することができます。

##### カウンタのクリア指定が可能

インプットキャプチャ入力信号の立上がり/立下がり/両エッジでカウンタの値をクリアするか、しないかを選択できます。

##### 2 種類の割込み要求

インプットキャプチャ × 1 要因、オーバフロー × 1 要因があります。インプットキャプチャ入力信号による割込み要求はインプットキャプチャ入力信号の立上がり/立下がりエッジを選択することができます。

##### ノイズ除去回路内蔵

ノイズ除去回路によりインプットキャプチャ入力信号の高域成分のノイズを除去することが可能です。

##### サブアクティブモード、サブスリープモードで動作

内部クロックとして  $f_w/2$  を選択した場合、サブアクティブモード、サブスリープモードで動作可能です。

(2) ブロック図

タイマGのブロック図を図9.6に示します。

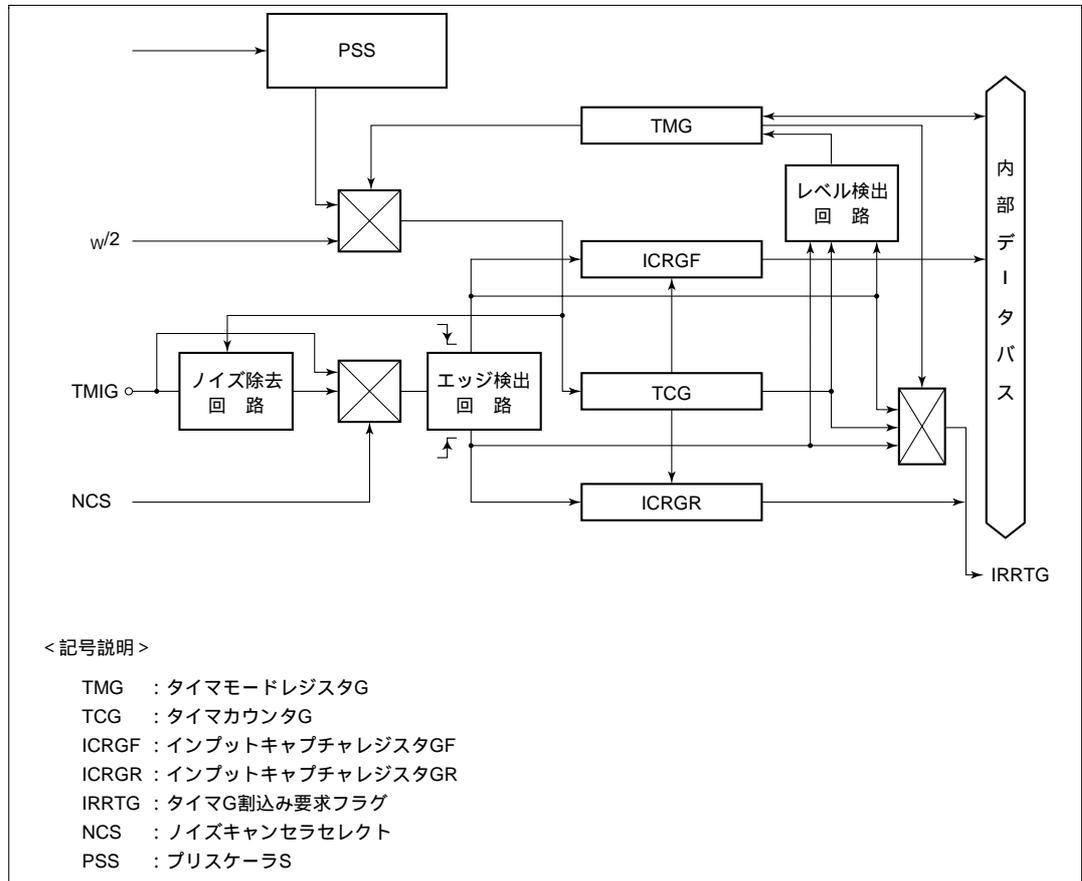


図9.6 タイマGのブロック図

### (3) 端子構成

タイマ G の端子構成を表 9.8 に示します。

表 9.8 端子構成

名称	略称	入出力	機能
インプットキャプチャ 入力	TMIG	入力	インプットキャプチャ入力端子

### (4) レジスタ構成

タイマ G のレジスタ構成を表 9.9 に示します。

表 9.9 レジスタ構成

名称	略称	R/W	初期値	アドレス
タイマモードレジスタ G	TMG	R/W	H'00	H'FFBC
タイマカウンタ G	TCG		H'00	
インプットキャプチャレジスタ GF	ICRGF	R	H'00	H'FFBD
インプットキャプチャレジスタ GR	ICRGR	R	H'00	H'FFBE

## 9.4.2 各レジスタの説明

### (1) タイマカウンタ G (TCG)

ビット:	7	6	5	4	3	2	1	0
	TCG7	TCG6	TCG5	TCG4	TCG3	TCG2	TCG1	TCG0
初期値:	0	0	0	0	0	0	0	0
R/W :	—	—	—	—	—	—	—	—

TCG は、8 ビットのアップカウンタで、入力したクロックによりカウントアップされます。入力するクロックは、TMG の CKS1、CKS0 で選択します。

TCG はインプットキャプチャタイマとして動作させる場合、PMR1 の TMIG を "1" に設定し、インターバルタイマとして動作させる場合、TMIG を "0" に設定します\*。インプットキャプチャタイマの動作時は、TMG の設定によりインプットキャプチャ入力信号の立上がり / 立下がり / 両エッジのいずれかで TCG の値をクリアすることができます。

TCG がオーバフロー (H'FF H'00) すると、TMG の OVIE が "1" の場合、IRR2 の IRRTG が "1" にセットされ、さらに IENR2 の IENTG が "1" の場合、CPU に対して割り込み要求信号が発生します。

なお、割り込みについての詳細は「3.3 割り込み」を参照してください。

TCG は CPU からリード / ライトすることはできません。

リセット時、TCGはH'00にイニシャライズされます。

【注】\* TMIGの書換えの際、インプットキャプチャ信号が発生する場合があります。

## (2) インプットキャプチャレジスタ GF (ICRGF)

ビット:	7	6	5	4	3	2	1	0
	ICRGF7	ICRGF6	ICRGF5	ICRGF4	ICRGF3	ICRGF2	ICRGF1	ICRGF0
初期値:	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R

ICRGFは、8ビットのリード専用のレジスタです。インプットキャプチャ入力信号の立下がりエッジが検出されると、そのときのTCGの値がICRGFに転送されます。このとき、TMGのIIEGSが"1"の場合、IRR2のIRRTGが"1"にセットされ、さらにIENR2のIENTGが"1"の場合、CPUに対して割込み要求信号が発生します。

なお、割込みについての詳細は「3.3 割込み」を参照してください。

インプットキャプチャ動作を確実にを行うために、インプットキャプチャ入力信号のパルス幅は、少なくとも2 または2<sub>SUB</sub>（ノイズ除去回路を使用しない場合）以上にしてください。

リセット時、ICRGFはH'00にイニシャライズされます。

## (3) インプットキャプチャレジスタ GR (ICRGR)

ビット:	7	6	5	4	3	2	1	0
	ICRGR7	ICRGR6	ICRGR5	ICRGR4	ICRGR3	ICRGR2	ICRGR1	ICRGR0
初期値:	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R

ICRGRは、8ビットのリード専用のレジスタです。インプットキャプチャ入力信号の立上がりエッジが検出されると、そのときのTCGの値がICRGRに転送されます。このとき、TMGのIIEGSが"0"の場合、IRR2のIRRTGが"1"にセットされ、さらにIENR2のIENTGが"1"の場合、CPUに対して割込み要求信号が発生します。

なお、割込みについての詳細は「3.3 割込み」を参照してください。

インプットキャプチャ動作を確実にを行うために、インプットキャプチャ入力信号のパルス幅は、少なくとも2 または2<sub>SUB</sub>（ノイズ除去回路を使用しない場合）以上にしてください。

リセット時、ICRGRはH'00にイニシャライズされます。

## (4) タイマモードレジスタ G (TMG)

ビット:	7	6	5	4	3	2	1	0
	OVFH	OVFL	OVIE	IIEGS	CCLR1	CCLR0	CKS1	CKS0
初期値:	0	0	0	0	0	0	0	0
R/W :	R/W*	R/W*	R/W	R/W	R/W	R/W	R/W	R/W

【注】\* フラグクリアのための"0"ライトのみ可能です。

TMG は、8 ビットのリード/ライト可能なレジスタです。TCG の 4 種類の内部クロックの選択、カウンタクリアの選択、インプットキャプチャ入力信号による割込み要求のエッジ選択、オーバフローによる割込み要求の許可の制御、およびオーバフローフラグの表示を行います。

リセット時、TMG は H'00 にイニシャライズされます。

## ビット7: タイマオーバフローフラグ H (OVFH)

インプットキャプチャ入力信号が"High"レベルのときに、TCG の値がオーバフロー (H'FF H'00) したことを示すステータスフラグです。本フラグは、ソフトウェアでクリアします。セットは、ハードウェアで行われます。ソフトウェアでセットすることはできません。

ビット7	
OVFH	説明
0	〔クリア条件〕 (初期値) OVFH = "1"の状態、OVFH をリードした後、OVFH に"0"をライトしたとき
1	〔セット条件〕 TCG の値が、H'FF H'00 になったとき

## ビット6: タイマオーバフローフラグ L (OVFL)

インプットキャプチャ信号が"Low"レベルのとき、またはインターバル動作時に、TCG の値がオーバフロー (H'FF H'00) したことを示すステータスフラグです。本フラグは、ソフトウェアでクリアします。セットは、ハードウェアで行われます。ソフトウェアでセットすることはできません。

ビット6	
OVFL	説明
0	〔クリア条件〕 (初期値) OVFL = "1"の状態、OVFL をリードした後、OVFL に"0"をライトしたとき
1	〔セット条件〕 TCG の値が、H'FF H'00 になったとき

## ビット5：タイマオーバーフローインタラプトイネーブル（OVIE）

TCGのオーバーフローが発生したとき、割込みの許可または禁止を選択します。

ビット5		
OVIE	説明	
0	TCGのオーバーフローによる割込み要求を禁止	（初期値）
1	TCGのオーバーフローによる割込み要求を許可	

## ビット4：インプットキャプチャインタラプトエッジセレクト（IIEGS）

インプットキャプチャ入力信号による割込み要求のエッジ選択を行います。

ビット4		
IIEGS	説明	
0	インプットキャプチャ入力信号の立上がりエッジで割込みを発生	（初期値）
1	インプットキャプチャ入力信号の立下がりエッジで割込みを発生	

## ビット3、2：カウンタクリア1、0（CCLR1、CCLR0）

インプットキャプチャ入力信号の立上がり / 立下がり / 両エッジで TCG の値をクリアするか、しないかを選択します。

ビット3	ビット2		
CCLR1	CCLR0	説明	
0	0	TCGのクリアを禁止	（初期値）
0	1	インプットキャプチャ入力信号の立下がりエッジにより TCG をクリア	
1	0	インプットキャプチャ入力信号の立上がりエッジにより TCG をクリア	
1	1	インプットキャプチャ入力信号の両エッジにより TCG をクリア	

## ビット1、0：クロックセレクト（CKS1、CKS0）

TCGに入力するクロックを、4種類の内部クロックから選択します。

ビット1	ビット0		
CKS1	CKS0	説明	
0	0	内部クロック： $f/64$ でカウント	（初期値）
0	1	内部クロック： $f/32$ でカウント	
1	0	内部クロック： $f/2$ でカウント	
1	1	内部クロック： $f_w/2$ でカウント	

### 9.4.3 ノイズ除去回路

ノイズ除去回路は、デジタルローパスフィルタとなっており、インプットキャプチャ入力端子から入力したパルスの高域成分のノイズを除去します。ノイズ除去回路は、PMR2のNCS\*により設定します。

図9.7にノイズ除去回路のブロック図を示します。

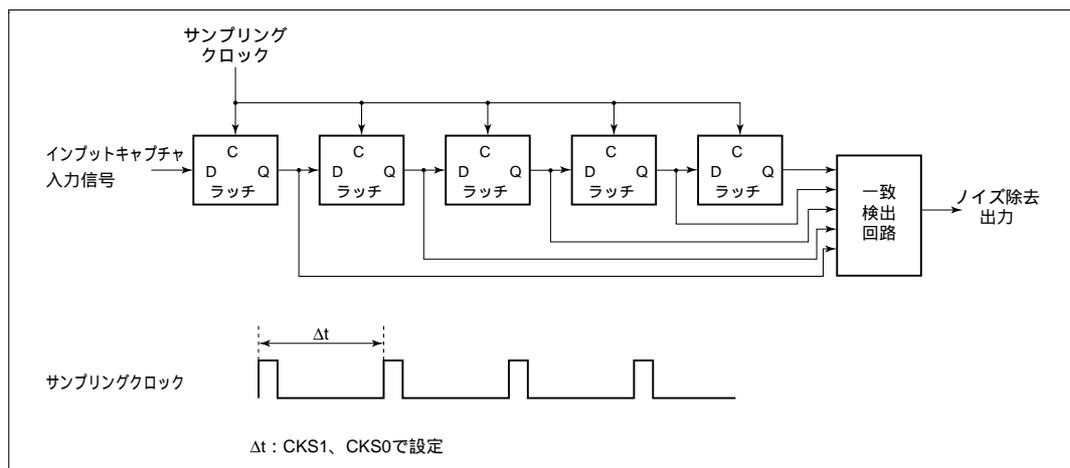


図 9.7 ノイズ除去回路のブロック図

ノイズ除去回路は5段直列に接続されたラッチ回路と一致検出回路で構成されています。ノイズ除去機能なし (NCS = "0") の場合、サンプリングクロックはシステムクロックが選択されています。ノイズ除去機能あり (NCS = "1") の場合、サンプリングクロックはTMGのCKS1、CKS0で選択された内部クロックが選択され、このクロックの立上がりエッジでインプットキャプチャ入力が入力され、すべてのラッチ出力が一致したとき正しいデータとみなします。一致しない場合は、前の値を保持します。リセット後は、ノイズ除去出力はインプットキャプチャ入力信号の立下がりエッジが5回サンプリングされた状態でイニシャライズされています。したがって、ノイズ除去機能ありを設定した後、サンプリングクロックの5倍より幅の広いパルスは確実にインプットキャプチャ信号になります。なお、ノイズ除去を行わない場合でも、インプットキャプチャ動作を確実に行うために、インプットキャプチャ入力信号のパルス幅は少なくとも2 または  $2_{SUB}$  以上必要です。

【注】 \* NCSビットの書換えの際、インプットキャプチャ信号が発生する場合があります。

図 9.8 にノイズ除去回路のタイミング例を示します。

この例は、インプットキャプチャ入力端子にサンプリングクロックの 5 倍以下の "High" レベル入力がノイズとして除去された場合です。

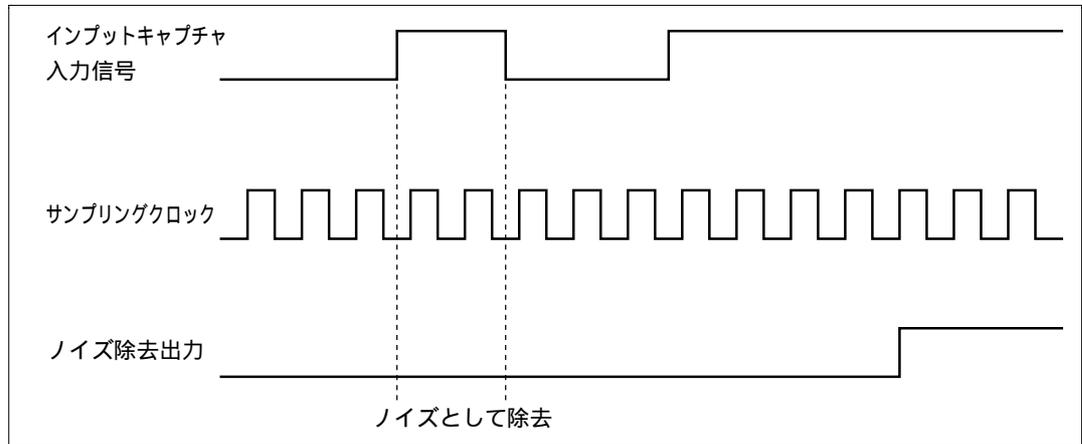


図 9.8 ノイズ除去回路タイミング (例)

#### 9.4.4 動作説明

##### (1) タイマ G の機能

タイマ G は 8 ビットのアップカウンタで、インプットキャプチャタイマとインターバルタイマの 2 種類の機能を持っています。

それぞれの動作について、以下に説明します。

##### (a) インプットキャプチャタイマの動作

ポートモードレジスタ 1 (PMR1) の TMIG ビットを "1" に設定すると、タイマ G はインプットキャプチャタイマとして機能します。\*

リセット時、タイマモードレジスタ G (TMG)、タイマカウンタ G (TCG)、インプットキャプチャレジスタ GF (ICRGF)、インプットキャプチャレジスタ GR (ICRGR) は、H'00 に初期化されます。

リセット直後、TCG は内部クロック /64 のクロックでカウントを開始します。

入力するクロックは、TMG の CKS1、CKS0 ビットにより 4 種類の内部クロックソースから選択できます。

TMIG 端子から入力されるインプットキャプチャ信号の立上がりエッジ / 立下がりエッジにより、そのときの TCG の値が ICRGR/ICRGF に転送されます。また、TMG の IIEGS ビットで選択したエッジが入力されると、IRR2 の IRRTG が "1" にセットされ、この時 IENR2 の IENTG ビットが "1" ならば CPU に割り込みを要求します。なお、割り込みについての詳細は、「3.3 割り込み」を参照してください。

TCG は TMG の CCLR1、CCLR0 ビットにより、インプットキャプチャ信号の立上がり / 立下がり / 両エッジのいずれかで TCG の値をクリアすることができます。また、イン

プットキャプチャ信号が"High"レベルのときに TCG がオーバーフローすると、TMG の OVFH ビットをセットします。一方、インプットキャプチャ信号が"Low"レベルのときに TCG がオーバーフローすると、TMG の OVFL ビットをセットします。

これらがセットされた時、TMG の OVIE ビットが"1"であれば IRR2 の IRRTG が"1"にセットされ、さらに IENR2 の IENTG ビットが"1"であれば、タイマ G は CPU に割込みを要求します。なお、割込みについての詳細は「3.3 割込み」を参照してください。

タイマ G にはノイズ除去回路が内蔵されており、TMIG 端子から入力したパルスの高域成分のノイズを除去できます。詳細は「9.4.3 ノイズ除去回路」を参照してください。

【注】\* TMIG の書換えの際、インプットキャプチャ信号が発生する場合があります。

#### (b) インターバルタイマの動作

PMR1 の TMIG ビットを"0"に設定すると、タイマ G はインターバルタイマとして機能します。リセット直後、TCG は内部クロック  $/64$  のクロックでカウントを開始します。入力クロックは TMG の CKS1、CKS0 ビットにより 4 種類の内部クロックソースから選択できます。TCG は選択されたクロックでカウントアップし、オーバーフロー (H'FF H'00) すると TMG の OVFL ビットが"1"にセットされます。この時、TMG の OVIE ビットが"1"であれば IRR2 の IRRTG が"1"にセットされ、さらに IENR2 の IENTG ビットが"1"であれば、タイマ G は CPU に割込みを要求します。なお、割込みについての詳細は「3.3 割込み」を参照してください。

## (2) カウントタイミング

TCG は入力された内部クロックによりカウントアップされます。TMG の CKS1、CKS0 の設定により、システムクロック (  $f_{sys}$  ) とウォッチクロック (  $f_w$  ) を分周して作られる 4 種類の内部クロック (  $f_{sys}/64$ 、  $f_{sys}/32$ 、  $f_{sys}/2$ 、  $f_w/2$  ) が選択されます。

## (3) インพุットキャプチャ入力のタイミング

## (a) ノイズ除去機能なしの場合

インพุットキャプチャ入力は、立上がり/立下がりエッジのそれぞれ専用のインพุットキャプチャ機能をもっています。

立上がり/立下がりエッジのインพุットキャプチャ入力のタイミングを図9.9に示します。

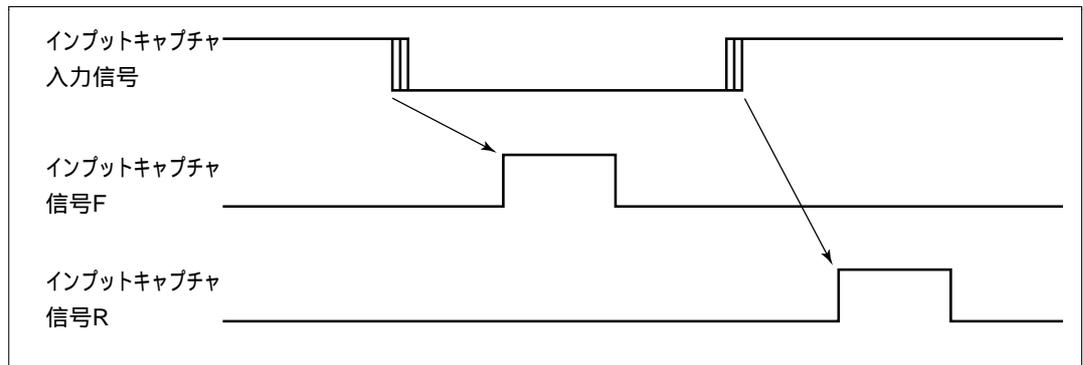


図9.9 インพุットキャプチャ入力のタイミング (ノイズ除去機能なし)

## (b) ノイズ除去機能ありの場合

インพุットキャプチャ入力でノイズ除去を行う場合、インพุットキャプチャ信号はノイズ除去回路を経由するため、インพุットキャプチャ入力信号エッジからサンプリングクロックの5周期分遅延します。

このときのタイミングを図9.10に示します。

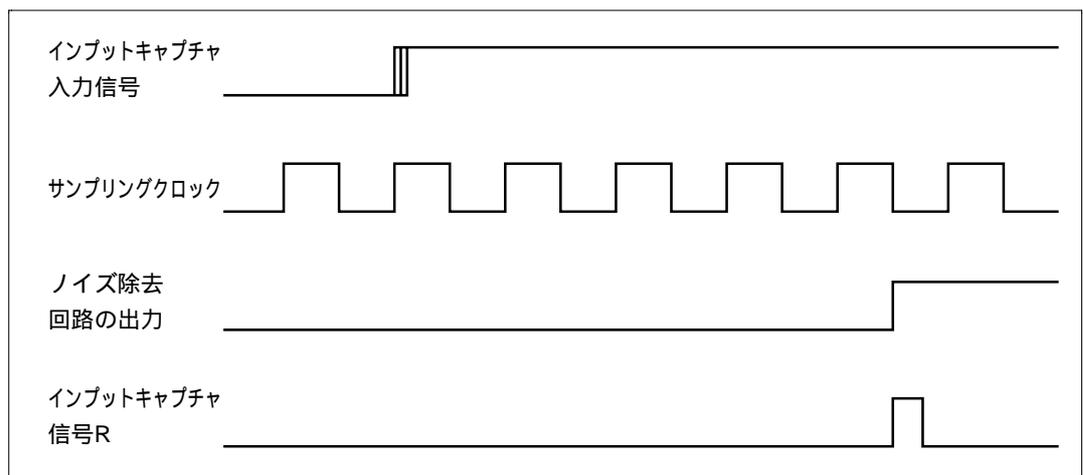


図9.10 インพุットキャプチャ入力のタイミング (ノイズ除去機能あり)

## (4) インพุットキャプチャ入力によるインพุットキャプチャタイミング

インพุットキャプチャ入力によるインพุットキャプチャタイミングを図 9.11 に示します。

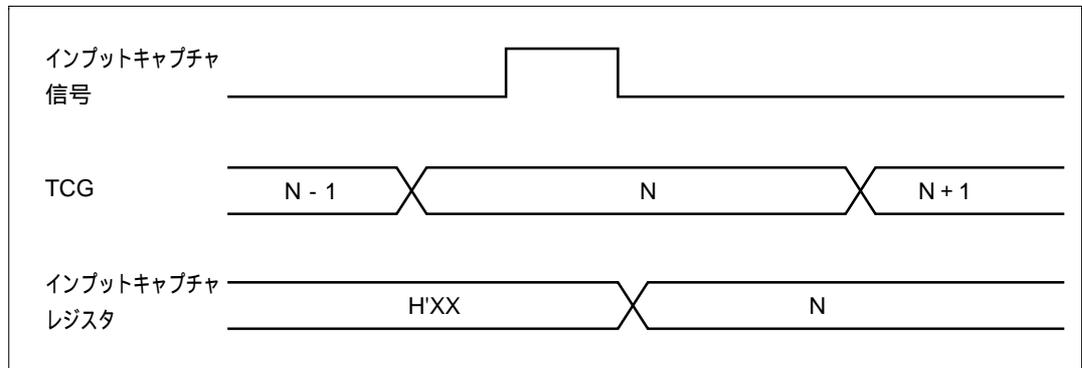


図 9.11 インพุットキャプチャ入力によるインพุットキャプチャタイミング

## (5) TCG のクリアタイミング

TCG の値は、インพุットキャプチャ入力信号の立上がりエッジ / 立下がりエッジ / 両エッジでクリアすることができます。

両エッジでクリアする場合のタイミングを図 9.12 に示します。

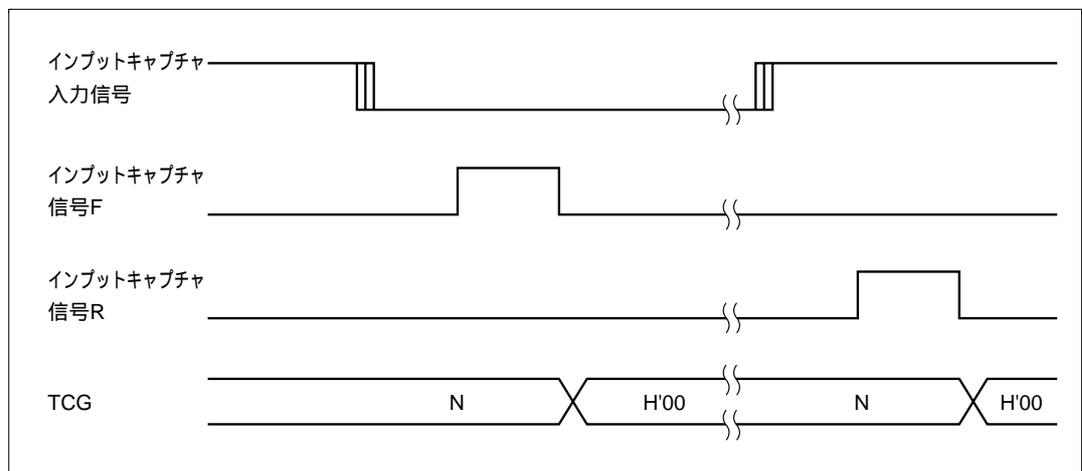


図 9.12 TCG のクリアタイミング

## (6) タイマ G の動作モード

タイマ G の動作モードを表 9.10 に示します。

表 9.10 タイマ G の動作モード

動作モード		リセット	アクティブ	スリープ	ウォッチ	サブアクティブ	サブスリープ	スタンバイ
TCG	インプット キャプチャ	リセット	動作*	動作*	停止	動作 / 停止*	動作 / 停止*	停止
	インターバル	リセット	動作*	動作*	保持	動作 / 停止*	動作 / 停止*	停止
ICRGF		リセット	動作*	動作*	保持	動作 / 停止*	動作 / 停止*	保持
ICRGR		リセット	動作*	動作*	保持	動作 / 停止*	動作 / 停止*	保持
TMG		リセット	動作	保持	保持	動作	保持	保持

【注】 \* アクティブモード、スリープモード時に、TCG の内部クロックとして  $w/2$  を選択した場合、システムクロックと内部クロックとが非同期であるため同期回路で同期をとっています。これにより、カウント周期は最大で  $1/$  (s) の誤差が生じます。また、サブアクティブモード、サブスリープモード時に、TCG の内部クロックとして  $w/2$  を選択した場合、サブクロック  $_{SUB}$  ( $w/8$ 、 $w/4$ 、 $w/2$ ) に関係なく TCG、ノイズ除去回路は内部クロック  $w/2$  で動作します。その他の内部クロックを選択した場合、TCG、ノイズ除去回路は動作せず、さらにインプットキャプチャ入力信号を入力してもインプットキャプチャされませんので注意してください。

## 9.4.5 使用上の注意事項

### (1) 内部クロックの切換えと TCG 動作

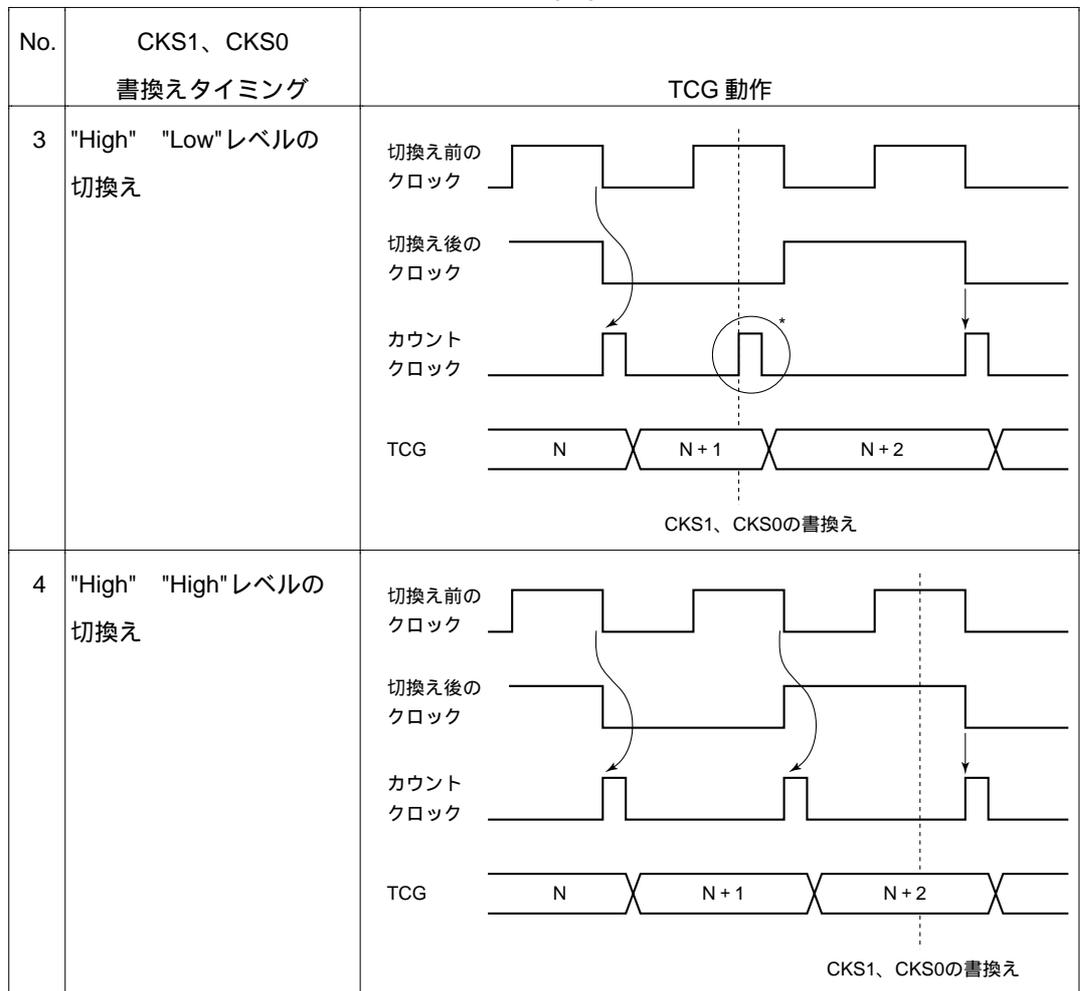
内部クロックを切換えるタイミングによっては、TCG がカウントアップされてしまう場合があります。内部クロックの切換えタイミング (CKS1、CKS0 の書換え) と TCG 動作の関係を表 9.11 に示します。

内部クロックを使用する場合、システムクロック ( ) またはサブクロック (  $w$  ) を分周した内部クロックの立下がりエッジで検出してカウントクロックを発生しています。そのため表 9.11 の No.3 のように切換え前のクロック "High" 切換え後のクロック "Low" レベルになるようなクロックの切換えを行うと、切換えタイミングを立下がりエッジとみなしてカウントクロックが発生し、TCG がカウントアップされてしまいます。

表 9.11 内部クロックの切換えと TCG 動作 (1)

No.	CKS1、CKS0 書換えタイミング	TCG 動作
1	"Low" "Low"レベルの 切換え	<p>切換え前のクロック</p> <p>切換え後のクロック</p> <p>カウントクロック</p> <p>TCG</p> <p>CKS1、CKS0の書換え</p>
2	"Low" "High"レベルの 切換え	<p>切換え前のクロック</p> <p>切換え後のクロック</p> <p>カウントクロック</p> <p>TCG</p> <p>CKS1、CKS0の書換え</p>

表 9.11 内部クロックの切換えとTCG動作(2)



【注】 \* 切換えのタイミングを下向きエッジとみなすために発生し、TCG はカウントアップされてしまいます。

## (2) ポートモードレジスタ書換えの注意事項

インプットキャプチャ入力の機能切換えのために、ポートモードレジスタを書換える際、またインプットキャプチャ入力のノイズ除去回路の機能切換えのために、ポートモードレジスタを書換える際に以下の点に注意してください。

### (a) インプットキャプチャ入力端子の機能切換え

インプットキャプチャ入力端子を制御しているポートモードレジスタ 1 (PMR1) のビット 3 (TMIG) を書換えて端子機能を切換えた場合、端子に有効なエッジが入力されていなくても、エッジが入力されたこととなりますので注意してください。インプットキャプチャ入力信号の入力エッジとその条件を表 9.12 に示します。

表 9.12 インプットキャプチャ入力端子の機能切換えによる

インプットキャプチャ入力信号の入力エッジとその条件

インプットキャプチャ 入力信号の入力エッジ	条件
立上がりエッジが発生 する場合	TMIG 端子が"High"レベルの状態、TMIG を"0"から"1"に書換えたとき
	TMIG 端子が"High"レベルの状態、NCS を"0"から"1"に書換えた後、ノイズ除去回路で 5 回サンプリングされる前に TMIG を"0"から"1"に書換えたとき
立下がりエッジが発生 する場合	TMIG 端子が"High"レベルの状態、TMIG を"1"から"0"に書換えたとき
	TMIG 端子が"Low"レベルの状態、NCS を"0"から"1"に書換えた後、ノイズ除去回路で 5 回サンプリングされる前に TMIG を"0"から"1"に書換えたとき
	TMIG 端子が"High"レベルの状態、NCS を"0"から"1"に書換えた後、ノイズ除去回路で 5 回サンプリングされた後に TMIG を"1"から"0"に書換えたとき

【注】 ポート P1<sub>3</sub> がインプットキャプチャ入力端子に設定されていない場合、タイマ G のインプットキャプチャ入力信号は"Low"レベルとなっています。

### (b) インプットキャプチャ入力のノイズ除去回路の切換え

インプットキャプチャ入力のノイズ除去回路を制御しているポートモードレジスタ 2 (PMR2) のビット 4 (NCS) を書換えて機能切換えを行う場合、TMIG を"0"にして行ってください。それ以外で書換えた場合、端子に有効なエッジが入力されていなくても、エッジが入力されたこととなりますので注意してください。インプットキャプチャ入力信号の入力エッジとその条件を表 9.13 に示します。

表 9.13 ノイズ除去回路の機能切換えによる  
インプットキャプチャ入力信号の入力エッジとその条件

インプットキャプチャ入力信号の入力エッジ	条件
立上がりエッジが発生する場合	TMIG が"1"の状態 で TMIG 端子を"Low"レベルから"High"レベルにした後、ノイズ除去回路で 5 回サンプリングする前に NCS を"1"から"0"に書換えたとき
立下がりエッジが発生する場合	TMIG が"1"の状態 で TMIG 端子を"High"レベルから"Low"レベルにした後、ノイズ除去回路で 5 回サンプリングする前に NCS を"1"から"0"に書換えたとき

端子機能を切換え、インプットキャプチャ入力信号にエッジが発生した場合、このエッジとインプットキャプチャインタラプトエッジセレクト ( IIEGS ) ビットによるエッジ選択とが一致した時、割り込み要求フラグが"1"にセットされますので、割り込み要求フラグを"0"にクリアにしてから使用してください。ポートモードレジスタの操作と割り込み要求フラグのクリア手順を図 9.13 に示します。端子機能を切換える場合、ポートモードレジスタの操作前に割り込み禁止状態にしてポートモードレジスタを切換え、ポートモードレジスタ操作後、インプットキャプチャ入力信号がインプットキャプチャ信号として確定するための期間 ( ノイズ除去回路を使用しない場合は 2 システムクロック以上、ノイズ除去回路を使用する場合はサンプリングクロックの 5 倍以上 ) 待ってから、"1"にセットされた割り込み要求フラグを"0"にクリアしてください。なお、端子機能切換えに伴う割り込み要求フラグのセットを回避する方法として表 9.12、表 9.13 の条件を満たさないように端子のレベルを制御する方法、または発生するエッジと反対のエッジに TMG の IIEGS ビットを設定して制御する方法もあります。

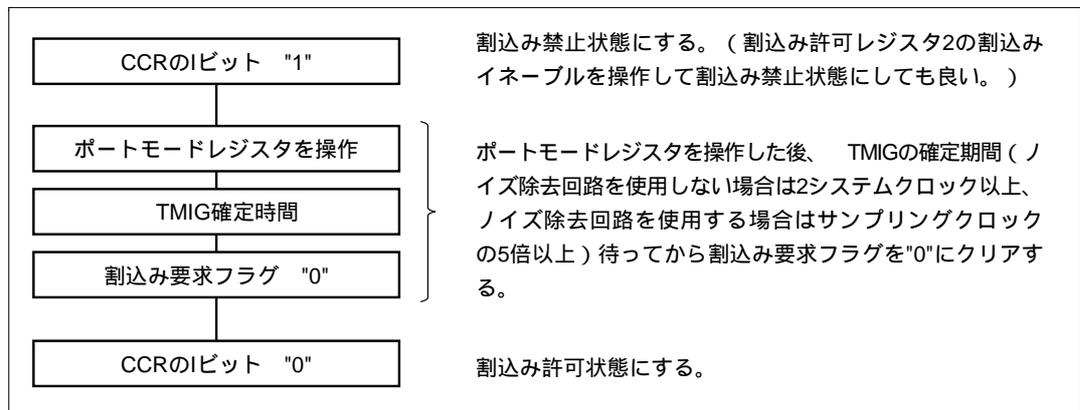


図 9.13 ポートモードレジスタの操作と割り込み要求フラグのクリア手順

### 9.4.6 タイマ G の使用例

タイマ G を使用すると、入力キャプチャ入力信号の "High" 幅、 "Low" 幅を絶対値で測定することができます。設定は、TMG の CCLR1、CCLR0 をそれぞれ "1" にセットします。

このときの動作例を図 9.14 に示します。

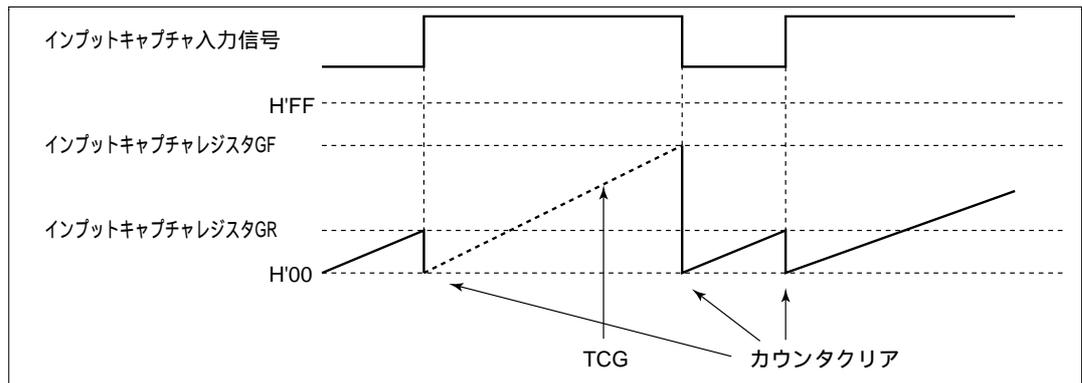


図 9.14 タイマ G の使用例

---

# 10. シリアルコミュニケーションインタフェース

---

## 第10章 目次

10.1	概要	215	
10.2	SCI1	216	
	10.2.1	概要	216
	10.2.2	各レジスタの説明	218
	10.2.3	動作説明	222
	10.2.4	割込み要因	224
10.3	SCI3	225	
	10.3.1	概要	225
	10.3.2	各レジスタの説明	228
	10.3.3	動作概要	245
	10.3.4	調歩同期式モード時の動作説明	249
	10.3.5	クロック同期式モード時の動作説明	258
	10.3.6	マルチプロセッサ通信機能	265
	10.3.7	割込み要因	270
	10.3.8	使用上の注意事項	271



## 10.1 概要

本 LSI は、2 チャンネルのシリアルコミュニケーションインタフェース (SCI : Serial Communication Interface) 1、3 を備えています。

2 チャンネルの SCI の機能概要を表 10.1 に示します。

表 10.1 SCI の機能概要

SCI の名称	機能	特長
SCI1	クロック同期式シリアル転送機能 ・転送データ長を選択可能 (8 ビット / 16 ビット) ・クロック連続出力機能	・8 種類の内部クロック ( /1024 ~ /2 ) と外部クロックを選択可能 ・オープンドレイン出力可能 ・転送完了で割込み発生
SCI3	クロック同期式シリアル転送機能 ・転送データ長 8 ビット ・送信 / 受信 / 同時送受信 調歩同期式シリアル転送機能 ・マルチプロセッサ通信機能 ・転送データ長を選択可能 (7 ビット / 8 ビット) ・ストップビット長を選択可能 (1 ビット / 2 ビット) ・パリティ付加機能	・ボーレートジェネレータ内蔵 ・受信エラーの検出 ・ブレークの検出 ・転送完了またはエラーで割込み発生

## 10.2 SCI1

### 10.2.1 概要

SCI1 (シリアルコミュニケーションインタフェース1) は、8ビット/16ビットデータのクロック同期式シリアル転送を行います。

#### (1) 特長

SCI1の特長を以下に示します。

8ビットおよび16ビットの転送データ長を選択可能

クロックソースとして8種類の内部クロック ( /1024、 /256、 /64、 /32、

/16、 /8、 /4、 /2) と外部クロックが選択可能

転送の完了で割り込み要求を発生

#### (2) ブロック図

SCI1のブロック図を図10.1に示します。

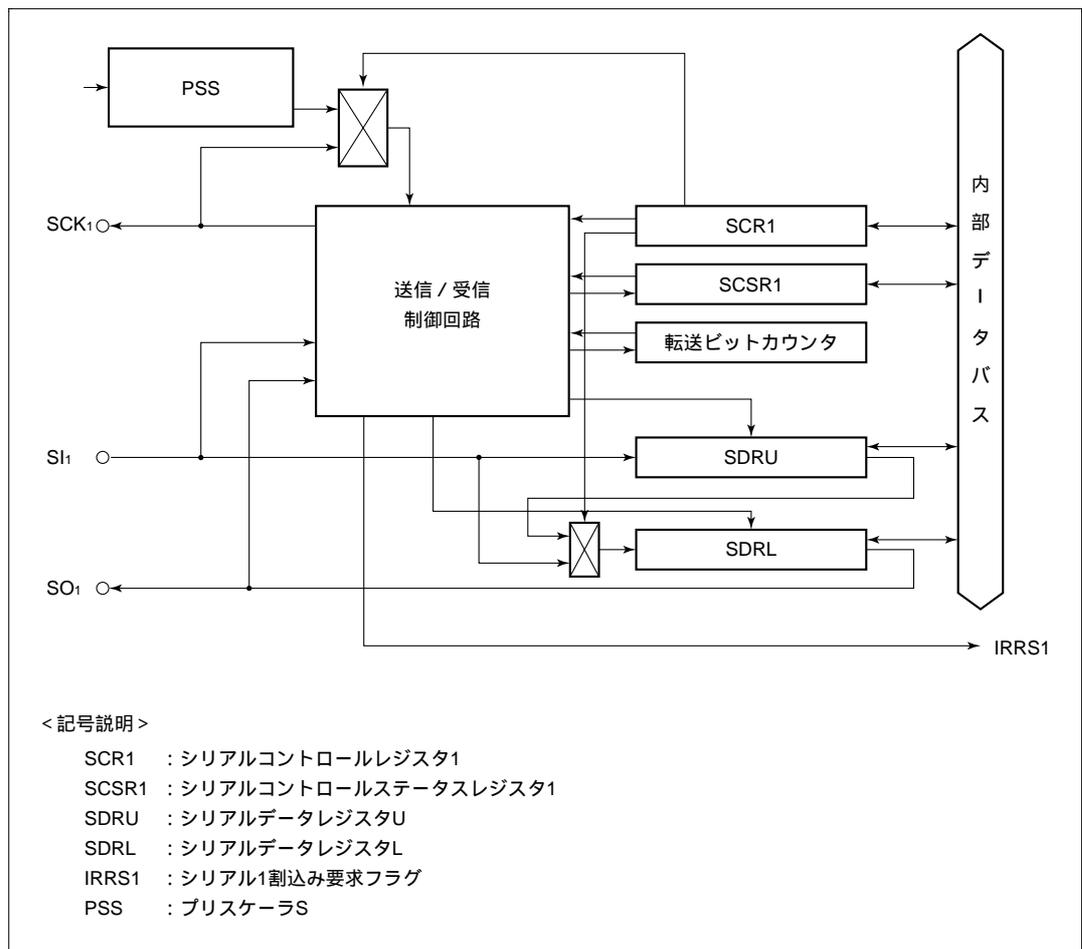


図 10.1 SCI1のブロック図

## (3) 端子構成

SCI1 の端子構成を表 10.2 に示します。

表 10.2 端子構成

名称	略称	入出力	機能
SCI1 クロック	SCK <sub>1</sub>	入出力	SCI1 のクロック入出力端子
SCI1 データ入力	SI <sub>1</sub>	入力	SCI1 の受信データ入力端子
SCI1 データ出力	SO <sub>1</sub>	出力	SCI1 の送信データ出力端子

## (4) レジスタ構成

SCI1 のレジスタ構成を表 10.3 に示します。

表 10.3 レジスタ構成

名称	略称	R/W	初期値	アドレス
シリアルコントロールレジスタ 1	SCR1	R/W	H'00	H'FFA0
シリアルコントロールステータスレジスタ 1	SCSR1	R/W	H'9C	H'FFA1
シリアルデータレジスタ U	SDRU	R/W	不定	H'FFA2
シリアルデータレジスタ L	SDRL	R/W	不定	H'FFA3

## 10.2.2 各レジスタの説明

### (1) シリアルコントロールレジスタ 1 (SCR1)

ビット:	7	6	5	4	3	2	1	0
	SNC1	SNC0	—	—	CKS3	CKS2	CKS1	CKS0
初期値:	0	0	0	0	0	0	0	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SCR1 は、8 ビットのリード/ライト可能なレジスタで、動作モード、転送クロックソースとプリスケアラ分周比を制御します。

リセット時、SCR1 は H'00 にイニシャライズされます。転送中に本レジスタに書込みを行うと転送を中止します。

ビット 7、6 : 動作モード選択 1、0 (SNC1、SNC0)

動作モードを選択します。

ビット 7	ビット 6	説明
SNC1	SNC0	
0	0	8 ビットクロック同期モード (初期値)
0	1	16 ビットクロック同期モード
1	0	クロック連続出力モード*1
1	1	リザーブ*2

【注】 \*1 SI<sub>1</sub>、SO<sub>1</sub>端子はポートとして使用してください。

\*2 SNC1、SNC0 に "11" を設定しないでください。

ビット 5、4 : リザーブビット

リザーブビットです。各ビットは "0" で使用してください。

ビット 3 : クロックソース選択 3 (CKS3)

供給するクロックソースの選択と SCK<sub>1</sub> 端子の入出力の設定を行います。

ビット 3	説明
CKS3	
0	クロックソースはプリスケアラ S、SCK <sub>1</sub> 端子は出力 (初期値)
1	クロックソースは外部クロック、SCK <sub>1</sub> 端子は入力

ビット2～0：クロック選択2～0（CKS2～CKS0）

CKS3が"0"の場合、プリスケアラ分周比と転送クロック周期を選択します。

ビット2	ビット1	ビット0	プリスケアラ分周比	転送クロック周期	
				= 5MHz	= 2.5MHz
0	0	0	/1024（初期値）	204.8μs	409.6μs
0	0	1	/256	51.2μs	102.4μs
0	1	0	/64	12.8μs	25.6μs
0	1	1	/32	6.4μs	12.8μs
1	0	0	/16	3.2μs	6.4μs
1	0	1	/8	1.6μs	3.2μs
1	1	0	/4	0.8μs	1.6μs
1	1	1	/2		0.8μs

## （2）シリアルコントロールステータスレジスタ1（SCSR1）

ビット：	7	6	5	4	3	2	1	0
	—	SOL	ORER	—	—	—	—	STF
初期値：	1	0	0	1	1	1	0	0
R/W：	—	R/W	R/(W)*	—	—	—	R	R/W

【注】\* フラグをクリアするための"0"ライトのみ可能です。

SCSR1は、動作状態、エラー状態などを示す8ビットのレジスタです。

リセット時、SCSR1はH'9Cにイニシャライズされます。

ビット7：リザーブビット

リザーブビットです。本ビットはリードすると常に"1"が読み出されます。ライトは無効です。

ビット6：拡張データビット（SOL）

SOLはSO<sub>1</sub>端子の出力レベルを変更します。また、リードするとSO<sub>1</sub>端子の出力レベルが読み出されます。送信完了後のSO<sub>1</sub>端子の出力は、送信データの最終ビットの値を保持しますが、送信前または送信後に本ビットを操作して、SO<sub>1</sub>端子の出力レベルを変更することができます。ただし、次の送信が開始するとSOLビットの設定は無効となります。したがって、送信完了後のSO<sub>1</sub>端子の出力レベルを変更する場合には、送信完了ごとにSOLビットに対するライト操作を行ってください。なお、データ転送中に本レジスタにライトすると誤動作の原因になりますので送信中は操作しないでください。

ビット6		説明
SOL		
0	リード時	SO <sub>1</sub> 端子の出力が"Low"レベル (初期値)
	ライト時	SO <sub>1</sub> 端子の出力を"Low"レベルに変更
1	リード時	SO <sub>1</sub> 端子の出力が"High"レベル
	ライト時	SO <sub>1</sub> 端子の出力を"High"レベルに変更

ビット5：オーバランエラーフラグ (ORER)

ORER は外部クロック使用時、オーバランエラーが発生したことを示します。転送中に外来雑音などにより正規の転送クロックに余分なパルスが重ね合わされた場合、転送データは保証できません。転送完了後もクロックが入力された場合、オーバラン状態であるとして、本ビットが"1"にセットされます。

ビット5		説明
ORER		
0	[クリア条件] "1"をリード後、"0"をライトしたとき	(初期値)
1	[セット条件] 外部クロック使用時、転送完了後もクロックが入力されたとき	

ビット4～2：リザーブビット

リザーブビットです。本ビットはリードすると常に"1"が読み出されます。ライトは無効です。

ビット1：リザーブビット

リザーブビットです。本ビットはリードするとリセット後は"0"が読み出されますが、リセット以外では不定です。ライトは無効です。

## ビット0：スタートフラグ（STF）

STFは、転送動作の開始を制御します。本ビットに"1"をセットするとSCI1の転送動作を開始します。

本ビットは、転送中とスタートビット待ち状態時は"1"を保持し、転送終了後に"0"にクリアされます。このため、ビジーフラグとして使用できます。

ビット0	説明	
STF		
0	リード時	転送動作は停止 (初期値)
	ライト時	無効
1	リード時	転送動作中
	ライト時	転送動作を開始

## (3) シリアルデータレジスタU（SDRU）

ビット:	7	6	5	4	3	2	1	0
	SDRU7	SDRU6	SDRU5	SDRU4	SDRU3	SDRU2	SDRU1	SDRU0
初期値:	不定							
R/W :	R/W							

SDRUは、8ビットのリード/ライト可能なレジスタで、16ビット転送時に上位8ビットのデータレジスタとして使用します（SDRLが下位8ビット）。

SDRUに書き込まれたデータは、SDRLにLSBファーストで出力されます。入れ代わりにSI<sub>1</sub>端子よりLSBファーストでデータが入力されて、MSB LSB方向にデータがシフトします。

SDRUのリード/ライトは、データの送信/受信が完了してから行う必要があります。データの送信/受信中にリード/ライトを行うとデータの内容は保証されません。

SDRUのリセット時の値は不定です。

## (4) シリアルデータレジスタL（SDRL）

ビット:	7	6	5	4	3	2	1	0
	SDRL7	SDRL6	SDRL5	SDRL4	SDRL3	SDRL2	SDRL1	SDRL0
初期値:	不定							
R/W :	R/W							

SDRLは、8ビットのリード/ライト可能なレジスタで、8ビット転送時のデータレジスタ、および16ビット転送時の下位8ビットのデータレジスタとして使用します（SDRUが上位8ビット）。

8ビット転送時、SDRL に書き込まれたデータは、SO<sub>1</sub> 端子より LSB ファーストで出力されます。入れ代わりに SI<sub>1</sub> 端子より LSB ファーストで入力されて、MSB LSB 方向にデータがシフトします。

16ビット転送時には、入力データが SDRU より取り込まれることを除けば、8ビット転送時と同様の動作となります。

SDRL のリード/ライトは、データの送信/受信が完了してから行う必要があります。データの送信/受信中にリード/ライトを行うとデータの内容は保証されません。

SDRL のリセット時の値は不定です。

### 10.2.3 動作説明

送信/受信フォーマットは8ビットおよび16ビットの転送データを選択可能です。クロックソースは、内部クロックと外部クロックを選択できます。外部クロック使用時はオーバーランエラーの検出ができます。

#### (1) クロック

転送クロックは、8種類の内部クロックと外部クロックから選択できます。内部クロックを選択した場合は、SCK<sub>1</sub> 端子はクロック出力端子となります。クロック連続出力モード(SCR1のSNC1、SNC0を"10")に設定するとCKS2~CKS0で選択したクロック( /1024 ~ /2)をSCK<sub>1</sub>端子から連続して出力します。外部クロックを選択した場合は、SCK<sub>1</sub> 端子はクロック入力端子となります。

#### (2) データ転送フォーマット

SCI1の転送フォーマットを図10.2に示します。データの最下位ビットから送受信されるLSBファースト方式による転送を行います。送信データは転送クロックの立下がりから次の立下がりまで出力されます。また、受信データは転送クロックの立上がりで取り込まれます。

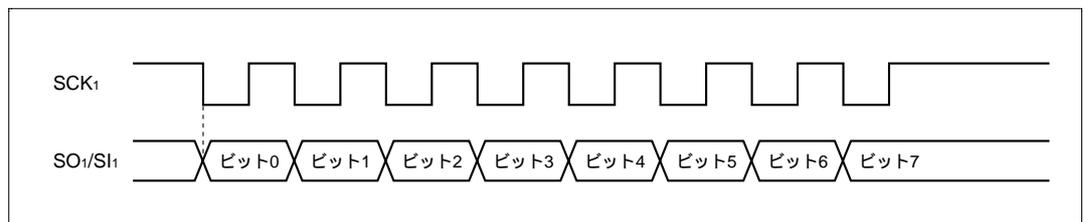


図 10.2 転送フォーマット

### (3) データの転送動作

#### (a) 送信動作

送信動作は次のように行われます。

- (1) PMR2のSO<sub>1</sub>を"1"、SCK<sub>1</sub>を"1"としてそれぞれSO<sub>1</sub>端子、SCK<sub>1</sub>端子に設定します。  
また、必要に応じて、PMR2のPOF<sub>1</sub>により、SO<sub>1</sub>端子をNMOSオープンドレイン出力とします。
- (2) SCR1のSNC<sub>1</sub>を"0"とし、SNC<sub>0</sub>を"0"または"1"として8ビットクロック同期方式か16ビットクロック同期方式に設定し、CKS<sub>3</sub>~CKS<sub>0</sub>で転送クロックを選択します。SCR1へデータの書込みを行うとSCI1の内部状態はイニシャライズされます。
- (3) SDRL、SDRUに転送データを書き込みます。  
8ビット転送モード：SDRL  
16ビット転送モード：上位バイトSDRU、下位バイトSDRL
- (4) SCSR1のSTFを"1"にすると、SCI1は動作を開始し、SO<sub>1</sub>端子から送信データが出力されます。
- (5) 送信完了後、IRR1のIRRS<sub>1</sub>が"1"にセットされます。

内部クロックを使用する場合は、送信データの出力と同時にSCK<sub>1</sub>端子から同期クロックが出力されます。送信が終了すると、次にスタートフラグを"1"にセットするまで同期クロックは出力されません。この間、SO<sub>1</sub>端子は直前のデータの最終ビットの値を出力し続けます。

外部クロックを使用する場合には、SCK<sub>1</sub>端子から入力されるクロックに同期してデータを送信します。送信終了後、引き続き同期クロックが入力されると、送信動作は行われず、オーバラン状態であるとして、SCSR1のORERが"1"にセットされます。

送信停止中のSO<sub>1</sub>端子の出力値は、SCSR1のSOLにより変更することができます。

#### (b) 受信動作

受信動作は次のように行われます。

- (1) PMR2のSI<sub>1</sub>を"1"、SCK<sub>1</sub>を"1"として、それぞれSI<sub>1</sub>端子、SCK<sub>1</sub>端子に設定します。
- (2) SCR1のSNC<sub>1</sub>を"0"とし、SNC<sub>0</sub>を"0"または"1"として8ビットクロック同期方式か16ビットクロック同期方式に設定し、CKS<sub>3</sub>~CKS<sub>0</sub>で転送クロックを選択します。SCR1へのデータの書込みを行うとSCI1の内部状態はイニシャライズされます。
- (3) SCSR1のSTFを"1"にすると、SCI1は動作を開始し、SI<sub>1</sub>端子から受信データを取り込みます。
- (4) 受信完了後、IRR1のIRRS<sub>1</sub>が"1"にセットされます。
- (5) SDRL、SDRUから受信データを読み出します。  
8ビット転送モード：SDRL

16ビット転送モード：上位バイト SDRU、下位バイト SDRL

(6) 受信終了後、引き続き同期クロックが入力されると、受信動作は行われず、オーバラン状態であるとして、SCSR1のORERが"1"にセットされます。

(c) 同時送受信動作

同時送受信動作は次のように行われます。

(1) PMR2のSO<sub>1</sub>を"1"、SI<sub>1</sub>を"1"、SCK<sub>1</sub>を"1"としてそれぞれSO<sub>1</sub>端子、SI<sub>1</sub>端子、SCK<sub>1</sub>端子に設定します。また、必要に応じてPMR2のPOF1により、SO<sub>1</sub>端子をNMOSオープンドレイン出力とします。

(2) SCR1のSNC1を"0"とし、SNC0を"0"または"1"として8ビットクロック同期方式か16ビットクロック同期方式に設定し、CKS3~CKS0で転送クロックを選択します。SCR1へのデータの書込みを行うとSCI1の内部状態はイニシャライズされます。

(3) SDRL、SDRUに転送データを書き込みます。

8ビット転送モード：SDRL

16ビット転送モード：上位バイト SDRU、下位バイト SDRL

(4) SCSR1のSTFを"1"にすると、SCI1は動作を開始し、SO<sub>1</sub>端子から送信データが出力され、またSI<sub>1</sub>端子から受信データが入力されます。

(5) 送受信完了後、IRR1のIRRS1が"1"にセットされます。

(6) SDRL、SDRUから受信データを読み出します。

8ビット転送モード：SDRL

16ビット転送モード：上位バイト SDRU、下位バイト SDRL

内部クロックを使用する場合は、送信データの出力と同時にSCK<sub>1</sub>端子から同期クロックが出力されます。送信が終了すると、次にスタートフラグを"1"にセットするまで同期クロックは出力されません。この間、SO<sub>1</sub>端子は直前のデータの最終ビットの値を出力し続けます。

外部クロックを使用する場合には、SCK<sub>1</sub>端子から入力されるクロックに同期してデータを送受信します。送受信終了後、引き続き同期クロックが入力されると、送受信動作は行われず、オーバラン状態であるとして、SCSR1のORERが"1"にセットされます。

送信停止中のSO<sub>1</sub>端子の出力値は、SCSR1のSOLにより変更することができます。

#### 10.2.4 割込み要因

SCI1の割込み要因には、転送完了があります。

SCI1が、転送完了すると、IRR1のIRRS1が"1"にセットされます。SCI1の割込み要求は、IENR1のIENS1により許可/禁止を選択できます。

詳細は、「3.3 割込み」を参照してください。

## 10.3 SCI3

### 10.3.1 概要

SCI3（シリアルコミュニケーションインタフェース3）は、調歩同期式とクロック同期式の2方式のシリアルデータ通信を行います。また、複数のプロセッサ間のシリアルデータ通信機能（マルチプロセッサ通信機能）を備えています。

#### （1）特長

SCI3の特長を以下に示します。

シリアルデータ通信モードを調歩同期式モードまたはクロック同期式モードから選択可能

- ・調歩同期式モード

キャラクタ単位で同期をとる調歩同期方式でシリアルデータ通信を行います。

Universal Asynchronous Receiver/Transmitter( UART )や、Asynchronous Communication Interface Adapter ( ACIA ) などの標準の調歩同期式通信用 LSI とのシリアルデータ通信ができます。

また、複数のプロセッサとシリアルデータ通信ができるマルチプロセッサ間通信機能を備えています。

通信フォーマットを12種類のフォーマットから選択できます。

データ長	7ビット/8ビット
ストップビット長	1ビット/2ビット
パリティ	偶数パリティ/奇数パリティ/パリティなし
マルチプロセッサビット	"1"/"0"
受信エラーの検出	パリティエラー、オーバランエラー、フレーミングエラー
ブレークの検出	フレーミングエラー発生時に RXD 端子のレベルを直接読み出すことでブレークを検出

- ・クロック同期式モード

クロックに同期してシリアルデータの通信を行います。クロック同期式通信機能をもつ他の LSI とのシリアルデータ通信が可能です。

データ長	8ビット
受信エラーの検出	オーバランエラー

全二重通信が可能

独立した送信部と受信部を備えているので、送信と受信を同時に行うことができます。また、送信部および受信部ともにダブルバッファ構造になっていますので、連続送信・連続受信ができます。

内蔵のボーレートジェネレータで任意のビットレートを選択可能

送受信クロックソースを内部クロック、または外部クロックから選択可能

6種類の割込み要因

送信終了、送信データエンpty、受信データフル、オーバランエラー、フレーミングエラー、パリティエラーの6種類の割込み要因があります。

(2) ブロック図

SCI3のブロック図を図10.3に示します。

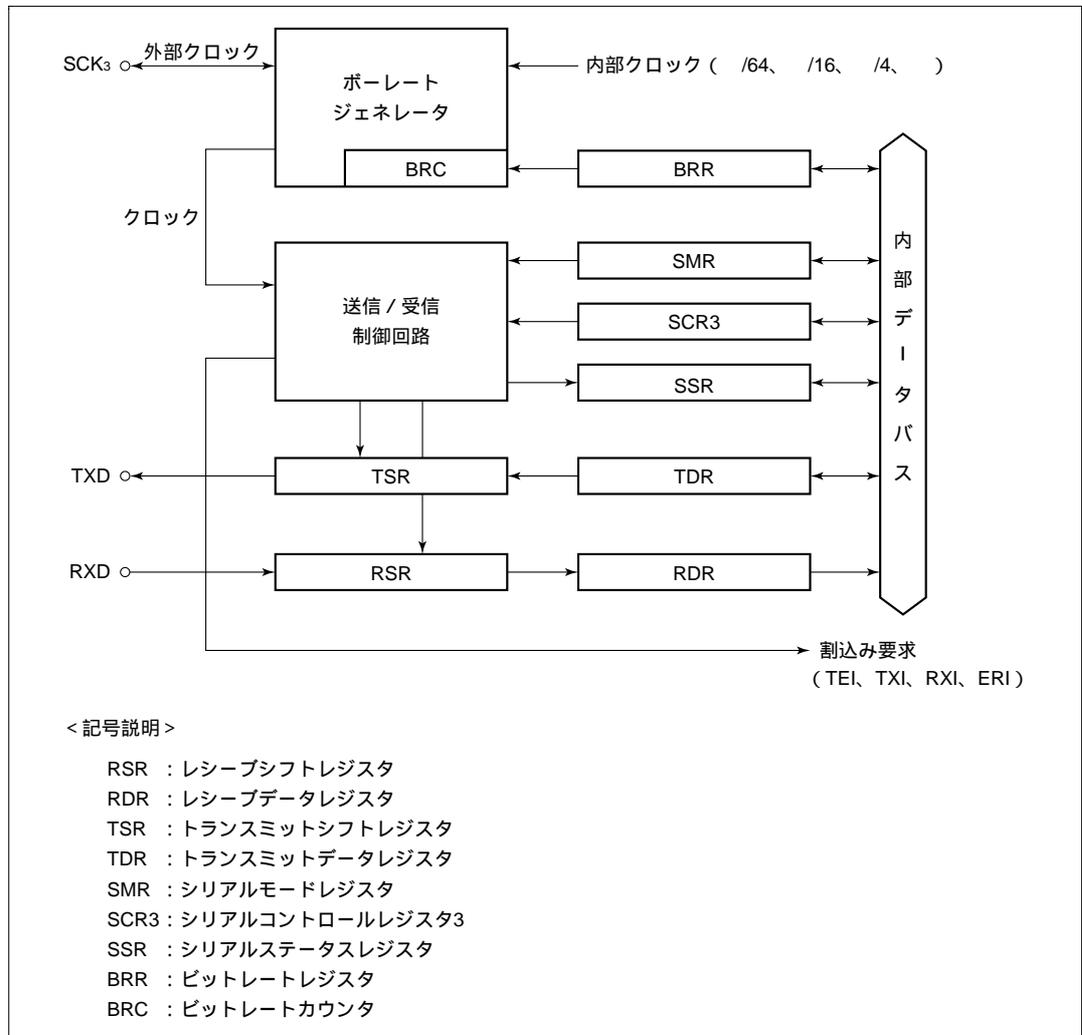


図 10.3 SCI3のブロック図

## (3) 端子構成

SCI3 の端子構成を表 10.4 に示します。

表 10.4 端子構成

名称	略称	入出力	機能
SCI3 クロック	SCK <sub>3</sub>	入出力	SCI3 のクロック入出力端子
SCI3 レシーブデータ入力	RXD	入力	SCI3 の受信データ入力端子
SCI3 トランスミットデータ出力	TXD	出力	SCI3 の送信データ出力端子

## (4) レジスタ構成

SCI3 のレジスタ構成を表 10.5 に示します。

表 10.5 レジスタ構成

名称	略称	R/W	初期値	アドレス
シリアルモードレジスタ	SMR	R/W	H'00	H'FFA8
ビットレートレジスタ	BRR	R/W	H'FF	H'FFA9
シリアルコントロールレジスタ 3	SCR3	R/W	H'00	H'FFAA
トランスミットデータレジスタ	TDR	R/W	H'FF	H'FFAB
シリアルステータスレジスタ	SSR	R/W	H'84	H'FFAC
レシーブデータレジスタ	RDR	R	H'00	H'FFAD
トランスミットシフトレジスタ	TSR	不可		
レシーブシフトレジスタ	RSR	不可		
ビットレートカウンタ	BRC	不可		

### 10.3.2 各レジスタの説明

#### (1) レシーブシフトレジスタ (RSR)

ビット:	7	6	5	4	3	2	1	0
R/W :	—	—	—	—	—	—	—	—

RSR は、シリアルデータを受信するためのレジスタです。RSR に RXD 端子から入力されたシリアルデータを、LSB (ビット 0) から受信した順にセットしパラレルデータに変換します。1 バイトのデータを受信すると、データは自動的に RDR へ転送されます。

CPU から RSR を直接リード/ライトすることはできません。

#### (2) レシーブデータレジスタ (RDR)

ビット:	7	6	5	4	3	2	1	0
	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1	RDR0
初期値:	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R

RDR は、受信したシリアルデータを格納する 8 ビットのレジスタです。

1 バイトのデータの受信が終了すると、受信したデータを RSR から RDR へ転送し、受信動作を完了します。その後、RSR は受信可能となります。

RSR と RDR はダブルバッファになっているため連続した受信動作が可能です。

RDR は、受信専用レジスタですので CPU からライトできません。

RDR は、リセット、スタンバイモード、ウォッチモード、サブアクティブモード、またはサブスリープモード時に H'00 にイニシャライズされます。

## (3) トランスミットシフトレジスタ (TSR)

ビット:	7	6	5	4	3	2	1	0
R/W :	—	—	—	—	—	—	—	—

TSR は、シリアルデータを送信するためのレジスタです。TDR から送信データをいったん TSR に転送し、LSB (ビット0) から順に TXD 端子に送出することでシリアルデータ送信を行います。1 バイトのデータを送信すると、自動的に TDR から TSR へ次の送信データを転送し、送信を開始します。ただし、TDR にデータが書き込まれていない (TDRE に"1"がセットされている) 場合には TDR から TSR へのデータ転送は行いません。

CPU から TSR を直接リード/ライトすることはできません。

## (4) トランスミットデータレジスタ (TDR)

ビット:	7	6	5	4	3	2	1	0
	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0
初期値:	1	1	1	1	1	1	1	1
R/W :	R/W							

TDR は、送信データを格納する 8 ビットのレジスタです。TSR の"空"を検出すると、TDR に書き込まれた送信データを TSR に転送し、シリアルデータ送信を開始します。TSR のシリアルデータ送信中に、TDR に次の送信データをライトしておくと、連続送信が可能です。

TDR は、常に CPU によるリード/ライトが可能です。

TDR は、リセット、スタンバイモード、ウォッチモード、サブアクティブモード、またはサブスリープモード時に H'FF にイニシャライズされます。

(5) シリアルモードレジスタ (SMR)

ビット:	7	6	5	4	3	2	1	0
	COM	CHR	PE	PM	STOP	MP	CKS1	CKS0
初期値:	0	0	0	0	0	0	0	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SMR は、シリアルデータ通信フォーマットの設定と、ポーレートジェネレータのクロックソースを選択するための8ビットのレジスタです。

SMR は、常にCPUによるリード/ライトが可能です。

SMR は、リセット、スタンバイモード、ウォッチモード、サブアクティブモード、またはサブスリープモード時にH'00にイニシャライズされます。

ビット7: コミュニケーションモード (COM)

COM は、SCI3 を調歩同期式モードとクロック同期式モードのいずれで動作させるかを選択するビットです。

ビット7	
COM	説明
0	調歩同期式モード (初期値)
1	クロック同期式モード

ビット6: キャラクタレングス (CHR)

CHR は、調歩同期式モード時におけるデータ長を選択するビットです。7ビットデータと8ビットデータのいずれかを選択できます。クロック同期式モードでは、CHR の設定にかかわらず、データ長は8ビットに固定となります。

ビット6	
CHR	説明
0	8ビットデータ (初期値)
1	7ビットデータ*

【注】 \* 7ビットデータを選択した場合、TDRのMSB(ビット7)は送信されません。

## ビット5：パリティイネーブル（PE）

PE は、調歩同期式モードで、送信時にパリティビットの付加を、受信時にパリティビットのチェックを行うかどうかを選択するビットです。クロック同期式モードではPE の設定にかかわらずパリティビットの付加およびチェックは行いません。

ビット5		
PE	説明	
0	パリティビットの付加およびチェックを禁止	（初期値）
1	パリティビットの付加およびチェックを許可*	

【注】 \* PE に"1"をセットすると送信時には、PM で指定した偶数、または奇数パリティを送信データに付加して送信します。受信時には、受信したパリティビットがPM で指定したパリティになっているかどうかをチェックします。

## ビット4：パリティモード（PM）

PM は、パリティの付加やチェックを偶数パリティ、または奇数パリティのどちらかで行うかを選択するビットです。PM の設定は、調歩同期式モードでPE に"1"を設定し、パリティビットの付加やチェックを許可したときのみ有効になります。クロック同期式モードや調歩同期式モードでパリティの付加やチェックを禁止している場合にはPMは無効です。

ビット4		
PM	説明	
0	偶数パリティ* <sup>1</sup>	（初期値）
1	奇数パリティ* <sup>2</sup>	

【注】 \*<sup>1</sup> 偶数パリティに設定すると、送信時にはパリティビットと送信データを合わせて、その中の"1"の数の合計が偶数になるようにパリティビットを付加して送信します。受信時には、パリティビットと受信データを合わせて、その中の"1"の数の合計が偶数になるかどうかをチェックします。

\*<sup>2</sup> 奇数パリティに設定すると、送信時にはパリティビットと送信データを合わせて、その中の"1"の数の合計が奇数になるようにパリティビットを付加して送信します。受信時には、パリティビットと受信データを合わせて、その中の"1"の数の合計が奇数になるかどうかをチェックします。

## ビット3：ストップビットレングス（STOP）

STOP は、調歩同期式モードでのストップビットの長さを選択するビットです。1 ビットまたは2ビットのいずれかを選択できます。STOP の設定は調歩同期式モードでのみ有効です。クロック同期式モードに設定した場合にはストップビットは付加されませんのでSTOPは無効です。

ビット3		
STOP	説明	
0	1ストップビット* <sup>1</sup>	(初期値)
1	2ストップビット* <sup>2</sup>	

【注】 \*1 送信時には、送信キャラクタの最後尾に 1 ビットの"1"（ストップビット）を付加して送信します。

\*2 送信時には、送信キャラクタの最後尾に 2 ビットの"1"（ストップビット）を付加して送信します。

なお、受信時にはSTOPの設定にかかわらず、受信したストップビットの1ビット目のみをチェックします。ストップビットの2ビット目が"1"の場合はストップビットとして扱いますが、"0"の場合は次の送信キャラクタのスタートビットとして扱います。

## ビット2：マルチプロセッサモード（MP）

MP は、マルチプロセッサ通信機能を許可するビットです。マルチプロセッサ通信機能を許可した場合、PE およびPMにおけるパリティの設定は無効になります。MPの設定は調歩同期式モードでのみ有効です。クロック同期式モードに設定した場合にはMPを"0"にしてください。

マルチプロセッサ通信機能については「10.3.6 マルチプロセッサ通信機能」を参照してください。

ビット2		
MP	説明	
0	マルチプロセッサ通信機能を禁止	(初期値)
1	マルチプロセッサ通信機能を許可	

ビット1、0：クロックセレクト1、0（CKS1、CKS0）

CKS1、CKS0は、内蔵ボーレートジェネレータのクロックソースを選択するビットです。CKS1、CKS0の組合せによって、 $/64$ 、 $/16$ 、 $/4$ 、の4種類からクロックソースを選択できます。

クロックソースと、ビットレートレジスタの設定値、およびボーレートの関係については、「(8) ビットレートレジスタ (BRR)」を参照してください。

ビット1	ビット0	説明
CKS1	CKS0	
0	0	クロック (初期値)
0	1	$/4$ クロック
1	0	$/16$ クロック
1	1	$/64$ クロック

#### (6) シリアルコントロールレジスタ3 (SCR3)

ビット:	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
初期値:	0	0	0	0	0	0	0	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SCR3は、送信/受信動作、調歩同期式モードでのクロック出力、割り込み要求の許可/禁止、および送信/受信クロックソースの選択を行う8ビットのレジスタです。

SCR3は、常にCPUによるリード/ライトが可能です。

SCR3は、リセット、スタンバイモード、ウォッチモード、サブアクティブモード、またはサブスリープモード時にH'00にイニシャライズされます。

ビット7：トランスミットインタラプトイネーブル (TIE)

TIEは、TDRからTSRへ送信データが転送され、SSRのTDREが"1"にセットされたとき、送信データエンpty割り込み要求 (TXI) の許可/禁止を選択するビットです。

TXIは、TDREを"0"にクリアするか、またはTIEを"0"にクリアすることで解除できます。

ビット7	説明
TIE	
0	送信データエンpty割り込み要求 (TXI) の禁止 (初期値)
1	送信データエンpty割り込み要求 (TXI) の許可

## ビット6：レシーブインタラプトイネーブル（RIE）

RIEは、受信データがRSRからRDRへ転送され、SSRのRDRFが"1"にセットされたとき、受信データフル割込み要求（RXI）、および受信エラー割込み要求（ERI）の許可/禁止を選択するビットです。なお、受信エラーにはオーバーランエラー、フレーミングエラー、パリティエラーの3種類があります。

RXIおよびERIは、RDRFまたは、FER、PER、OERのエラーフラグを"0"にクリアするか、RIEを"0"にクリアすることで解除できます。

ビット6	
RIE	説明
0	受信データフル割込み要求（RXI）、および受信エラー割込み要求（ERI）を禁止 (初期値)
1	受信データフル割込み要求（RXI）、および受信エラー割込み要求（ERI）を許可

## ビット5：トランスミットイネーブル（TE）

TEは、送信動作の開始の許可/禁止を選択するビットです。

ビット5	
TE	説明
0	送信動作を禁止*1（TXD端子はトランスミットデータ端子） (初期値)
1	送信動作を許可*2（TXD端子はトランスミットデータ端子）

【注】 \*1 SSRのTDREは"1"に固定されます。送信動作は禁止されますが、TXD端子はトランスミットデータ端子として機能します。入出力端子として使用する場合にはPMR6のTXDを"0"にしてください。

\*2 この状態で、TDRに送信データをライトするとSSRのTDREが"0"にクリアされシリアルデータ送信を開始します。なお、TEを"1"にセットする前に必ずSMRの設定を行い、送信フォーマットを決定してください。

## ビット4：レシーブイネーブル（RE）

REは、受信動作の開始の許可/禁止を選択するビットです。

ビット4	
RE	説明
0	受信動作を禁止*1（RXD端子は入出力ポート） (初期値)
1	受信動作を許可*2（RXD端子はレシーブデータ端子）

【注】 \*1 REを"0"にクリアしてもSSRのRDRF、FER、PER、OERの各フラグは影響を受けず、状態を保持しますので注意してください。

\*2 この状態で調歩同期式モードの場合はスタートビットを、クロック同期式モードの場合は同期クロック入力を、それぞれ検出するとシリアルデータ受信を開始します。なお、RE を"1"にセットする前に必ず SMR の設定を行い、受信フォーマットを決定してください。

### ビット3：マルチプロセッサインタラプトイネーブル（MPIE）

MPIE は、マルチプロセッサ割込み要求の許可 / 禁止を選択するビットです。MPIE の設定は、調歩同期式モードで、かつ SMR の MP が"1"に設定されている受信時にのみ有効です。COM が"1"のときや MP が"0"のときには MPIE は無効です。

ビット3	
MPIE	説明
0	マルチプロセッサ割込み要求を禁止（通常の受信動作） [ クリア条件 ] マルチプロセッサビットが"1"のデータを受信したとき (初期値)
1	マルチプロセッサ割込み要求を許可*

【注】 \* RSR から RDR への受信データの転送、および受信エラーの検出と SSR の RDRF、FER、OER の各ステータスフラグのセットは行いません。マルチプロセッサビットが"1"のデータを受け取るまで、RXI、ERI、および、SSR の RDRF、FER、OER の各フラグのセットを禁止します。マルチプロセッサビットが"1"の受信キャラクタを受け取ると、SSR の MPBR を"1"にセットし、MPIE を自動的に"0"にクリアし、RXI、ERI の発生（SCR の TIE、RIE が"1"にセットされている場合）と RDRF、FER、OER のセットが許可されます。

### ビット2：トランスミットエンドインタラプトイネーブル（TEIE）

TEIE は、MSB データ送出時に有効な送信データが TDR にないとき、送信終了割込み要求（TEI）の許可 / 禁止を選択するビットです。

ビット2	
TEIE	説明
0	送信終了割込み要求（TEI）を禁止 (初期値)
1	送信終了割込み要求（TEI）を許可*

【注】 \* TEI は、SSR の TDRE を"0"にクリアして TEND を"0"にクリアするか、TEIE を"0"にクリアすることで解除できます。

### ビット1、0：クロックイネーブル1、0（CKE1、CKE0）

CKE1、CKE0 は、クロックソースの選択、および SCK<sub>3</sub> 端子からのクロックの出力の許可 / 禁止を選択するビットです。CKE1 と CKE0 の組合せによって SCK<sub>3</sub> 端子を入出力ポートにするか、クロック出力端子にするか、またはクロック入力端子にするかが決まります。

ただし、CKE0 の設定は調歩同期式モードで内部クロック動作時 (CKE1 = "0") のみ有効です。クロック同期式モードのとき、および外部クロック動作 (CKE1 = "1") の場合は CKE0 に"0"を設定してください。

また、CKE1、CKE0 を設定した後に SMR で動作モードを決定してください。

クロックソースの選択についての詳細は「10.3.3 動作概要」の表 10.10 を参照してください。

ビット1	ビット0	説明		
CKE1	CKE0	コミュニケーションモード	クロックソース	SCK <sub>3</sub> 端子機能
0	0	調歩同期式	内部クロック	入出力ポート*1
		クロック同期式	内部クロック	同期クロック出力*1
0	1	調歩同期式	内部クロック	クロック出力*2
		クロック同期式	リザーブ	
1	0	調歩同期式	外部クロック	クロック入力*3
		クロック同期式	外部クロック	同期クロック入力
1	1	調歩同期式	リザーブ	
		クロック同期式	リザーブ	

【注】 \*1 初期値

\*2 ビットレートと同じ周波数のクロックを出力します。

\*3 ビットレートの 16 倍の周波数のクロックを入力してください。

### (7) シリアルステータスレジスタ (SSR)

ビット:	7	6	5	4	3	2	1	0
	TDRE	RDRF	OER	FER	PER	TEND	MPBR	MPBT
初期値:	1	0	0	0	0	1	0	0
R/W :	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

【注】\* フラグをクリアするための"0"ライトのみ可能です。

SSR は、SCI3 の動作状態を示すステータスフラグと、マルチプロセッサビットを内蔵した 8 ビットのレジスタです。

SSR は常に、CPU からリード/ライトできます。ただし、TDRE、RDRF、OER、PER、FER へ"1"をライトすることはできません。また、これらに"0"をライトしてクリアするためには、あらかじめ"1"をリードしておく必要があります。

また、TEND および MPBR はリード専用であり、ライトすることはできません。

SSR は、リセット、スタンバイモード、ウォッチモード、サブアクティブモード、またはサブスリープモード時に H'84 にイニシャライズされます。

## ビット7：トランスミットデータレジスタエンプティ (TDRE)

TDRE は、TDR から TSR に送信データの転送が行われたことを示すビットです。

ビット7	
TDRE	説明
0	TDR にライトされた送信データが TSR に転送されていない [クリア条件] (1) TDRE = "1"の状態をリードした後、"0"をライトしたとき (2) 命令で TDR ヘデータをライトしたとき
1	TDR に送信データがライトされていない、または TDR にライトされた送信データが TSR に転送された [セット条件] (1) SCR3 の TE が"0"のとき (2) TDR から TSR にデータ転送が行われたとき (初期値)

## ビット6：レシーブデータレジスタフル (RDRF)

RDRF は、受信したデータが RDR に格納されていることを示すビットです。

ビット6	
RDRF	説明
0	RDR に受信データが格納されていない (初期値) [クリア条件] (1) RDRF = "1"の状態をリードした後、"0"をライトしたとき (2) 命令で RDR のデータをリードしたとき
1	RDR に受信データが格納されている [セット条件] 受信が正常終了し、RSR から RDR へ受信データが転送されたとき

【注】 受信時にエラーを検出したとき、および SCR3 の RE を"0"にクリアしたときには、RDR および RDRF は影響を受けず以前の状態を保持します。RDRF が"1"にセットされたままデータの受信を完了するとオーバランエラー (OER) を発生し、受信データが失われますので注意してください。

## ビット5：オーバランエラー（OER）

OER は、受信時にオーバランエラーが発生したことを示すビットです。

ビット5		
OER	説明	
0	受信中、または受信を完了した*1 [クリア条件] OER = "1"の状態をリードした後、"0"をライトしたとき	(初期値)
1	受信時にオーバランエラーが発生した*2 [セット条件] RDRF が"1"の状態を受信を完了したとき	

【注】 \*1 SCR3 の RE を"0"にクリアしたときには、OER は影響を受けず以前の状態を保持します。

\*2 RDR では、オーバランエラーが発生する前の受信データが保持され、後から受信したデータが失われます。なお、OER が"1"にセットされた状態で、以降の受信を続けることはできません。クロック同期式モードでは送信も続けることができません。

## ビット4：フレーミングエラー（FER）

FER は、調歩同期式モードで受信時にフレーミングエラーが発生したことを示すビットです。

ビット4		
FER	説明	
0	受信中、または受信を完了した*1 [クリア条件] FER = "1"の状態をリードした後、"0"をライトしたとき	(初期値)
1	受信時にフレーミングエラーが発生した [セット条件] 受信終了時に受信データの最後尾のストップビットが"1"であるかどうかをチェックし、ストップビットが"0"であったとき*2	

【注】 \*1 SCR3 の RE を"0"にクリアしたときには、FER は影響を受けず以前の状態を保持します。

\*2 2ストップビットモード時は、1ビット目のストップビットが"1"であるかどうかのみを判定し、2ビット目のストップビットはチェックをしませんので注意してください。なお、フレーミングエラーが発生したときの受信データは RDR に転送されますが、RDRF はセットされません。さらに、FER が"1"にセットされた状態においては、以降の受信を続けることはできません。また、クロック同期式モードでは、FER が"1"にセットされていると送信および受信はできません。

## ビット3：パリティエラー（PER）

PER は、調歩同期式モードで、パリティを付加した受信時にパリティエラーが発生したことを示すビットです。

ビット3		
PER	説明	
0	受信中、または受信を完了した* <sup>1</sup> [クリア条件] PER = "1"の状態をリードした後、"0"をライトしたとき	(初期値)
1	受信時にパリティエラーが発生した* <sup>2</sup> [セット条件] 受信時に受信データとパリティビットをあわせた"1"の数が SMR の PM で設定したパリティと一致しなかったとき	

【注】 \*1 SCR3 の RE を"0"にクリアしたときには、PER は影響を受けず以前の状態を保持します。

\*2 パリティエラーが発生したときの受信データは RDR に転送されますが、RDRF はセットされません。なお、PER が"1"にセットされた状態では、以降の受信を続けることはできません。また、クロック同期式モードでは、PER が"1"にセットされていると送信および受信はできません。

## ビット2：トランスミットエンド（TEND）

TEND は、送信キャラクタの最後尾ビットを送信時に、TDRE が"1"にセットされていたことを示すビットです。

TEND は、リード専用ですのでライトすることはできません。

ビット2		
TEND	説明	
0	送信中 [クリア条件] (1) TDRE = "1"の状態をリードした後、TDRE に"0"をライトしたとき (2) 命令で TDR にデータをライトしたとき	
1	送信を終了 [セット条件] (1) SCR3 の TE が"0"のとき (2) 送信キャラクタの最後尾ビットの送信時に、TDRE が"1"であったとき	(初期値)

ビット1：マルチプロセッサビットレシーブ (MPBR)

MPBR は、調歩同期式モードで受信をマルチプロセッサフォーマットで行うときに、受信キャラクタ中のマルチプロセッサビットを格納するビットです。

リード専用ですのでライトすることはできません。

ビット1	
MPBR	説明
0	マルチプロセッサビットが"0"のデータを受信した* (初期値)
1	マルチプロセッサビットが"1"のデータを受信した

【注】 \* マルチプロセッサフォーマットで SCR3 の RE を"0"にクリアしたときには、MPBR は影響を受けず、以前の状態を保持します。

ビット0：マルチプロセッサビットトランスファ (MPBT)

MPBT は、調歩同期式モードで送信をマルチプロセッサフォーマットで行うときに、送信データに付加するマルチプロセッサビットを格納するビットです。クロック同期式モードを選択しているときやマルチプロセッサ通信機能を禁止しているとき、あるいは送信でないときには MPBT は無効です。

ビット0	
MPBT	説明
0	マルチプロセッサビット"0"を送信 (初期値)
1	マルチプロセッサビット"1"を送信

(8) ビットレートレジスタ (BRR)

ビット:	7	6	5	4	3	2	1	0
	BRR7	BRR6	BRR5	BRR4	BRR3	BRR2	BRR1	BRR0
初期値:	1	1	1	1	1	1	1	1
R/W :	R/W							

BRR は、SMR の CKS1、CKS0 で選択されるポーレートジェネレータの動作クロックとあわせて、送信 / 受信のビットレートを設定する 8 ビットのレジスタです。

BRR は、常に CPU によるリード / ライトが可能です。

BRR は、リセット、スタンバイモード、ウォッチモード、サブアクティブモード、またはサブスリープモード時に H'FF にイニシャライズされます。

調歩同期式モードのBRRの設定例を表10.6に示します。表10.6はアクティブ（高速）モードでの値を示しています。

表10.6 ビットレートに対するBRRの設定例〔調歩同期式モード〕（1）

Rビットレート (bit/s)	OSC (MHz)											
	2			2.4576			4			4.194304		
	n	N	誤差 (%)	n	N	誤差 (%)	n	N	誤差 (%)	n	N	誤差 (%)
110	1	70	+0.03	1	86	+0.31	1	141	+0.03	1	148	-0.04
150	0	207	+0.16	0	255	0	1	103	+0.16	1	108	+0.21
300	0	103	+0.16	0	127	0	0	207	+0.16	0	217	+0.21
600	0	51	+0.16	0	63	0	0	103	+0.16	0	108	+0.21
1200	0	25	+0.16	0	31	0	0	51	+0.16	0	54	-0.70
2400	0	12	+0.16	0	15	0	0	25	+0.16	0	26	+1.14
4800				0	7	0	0	12	+0.16	0	13	-2.48
9600				0	3	0				0	6	-2.48
19200				0	1	0						
31250	0	0	0				0	1	0			
38400				0	0	0						

表10.6 ビットレートに対するBRRの設定例〔調歩同期式モード〕（2）

Rビットレート (bit/s)	OSC (MHz)											
	4.9152			6			7.3728			8		
	n	N	誤差 (%)	n	N	誤差 (%)	n	N	誤差 (%)	n	N	誤差 (%)
110	1	174	-0.26	1	212	+0.03	2	64	+0.70	2	70	+0.03
150	1	127	0	1	155	+0.16	1	191	0	1	207	+0.16
300	0	255	0	1	77	+0.16	1	95	0	1	103	+0.16
600	0	127	0	0	155	+0.16	0	191	0	0	207	+0.16
1200	0	63	0	0	77	+0.16	0	95	0	0	103	+0.16
2400	0	31	0	0	38	+0.16	0	47	0	0	51	+0.16
4800	0	15	0	0	19	-2.34	0	23	0	0	25	+0.16
9600	0	7	0	0	9	-2.34	0	11	0	0	12	+0.16
19200	0	3	0	0	4	-2.34	0	5	0			
31250				0	2	0				0	3	0
38400	0	1	0				0	2	0			

表 10.6 ビットレートに対する BRR の設定例〔調歩同期式モード〕(3)

R ビットレート (bit/s)	OSC (MHz)					
	9.8304			10		
	n	N	誤差 (%)	n	N	誤差 (%)
110	2	86	+0.31	2	88	-0.25
150	1	255	0	2	64	+0.16
300	1	127	0	1	129	+0.16
600	0	255	0	1	64	+0.16
1200	0	127	0	0	129	+0.16
2400	0	63	0	0	64	+0.16
4800	0	31	0	0	32	-1.36
9600	0	15	0	0	15	+1.73
19200	0	7	0	0	7	+1.73
31250	0	4	-1.70	0	4	0
38400	0	3	0	0	3	+1.73

- 【注】 1. 誤差は、1%以内となるように設定してください。  
 2. BRR の設定値は以下の計算式で求められます。

$$N = \frac{\text{OSC}}{64 \times 2^{2n} \times B} \times 10^6 - 1$$

B : ビットレート (bit/s)

N : ポーレートジェネレータの BRR の設定値 (0 N 255)

OSC :  $\text{osc}$  の値 (MHz)

n : ポーレートジェネレータの入力クロックの No. (n=0、1、2、3)

(n とクロックの関係は表 10.7 を参照)

表 10.7 n とクロックの関係

n	クロック	SMR の設定値	
		CKS1	CKS0
0		0	0
1	/4	0	1
2	/16	1	0
3	/64	1	1

3. 表 10.6 の誤差は以下の計算式で求めた値を小数点第 3 位を四捨五入して表示してあります。

$$\text{誤差}(\%) = \frac{B(n, N, \text{OSCから求めたビットレート}) - R(\text{表10.6左欄のビットレート})}{R(\text{表10.6の左欄のビットレート})} \times 100$$

調歩同期式モードの各周波数における最大ビットレートを表 10.8 に示します。  
表 10.8 はアクティブ（高速）モードでの値を示しています。

表 10.8 各周波数における最大ビットレート〔調歩同期式モード〕

OSC (MHz)	最大ビットレート (bit/s)	設定値	
		n	N
2	31250	0	0
2.4576	38400	0	0
4	62500	0	0
4.194304	65536	0	0
4.9152	76800	0	0
6	93750	0	0
7.3728	115200	0	0
8	125000	0	0
9.8304	153600	0	0
10	156250	0	0

クロック同期式モードのBRRの設定例を表10.9に示します。表10.9はアクティブ(高速)モードでの値を示しています。

表10.9 ビットレートに対するBRRの設定例〔クロック同期式モード〕

B ビットレート (bit/s)	OSC (MHz)							
	2		4		8		10	
	n	N	n	N	n	N	n	N
110								
250	1	249	2	124	2	249		
500	1	124	1	249	2	124		
1k	0	249	1	124	1	249		
2.5k	0	99	0	199	1	99	1	124
5k	0	49	0	99	0	199	0	249
10k	0	24	0	49	0	99	0	124
25k	0	9	0	19	0	39	0	49
50k	0	4	0	9	0	19	0	24
100k			0	4	0	9		
250k	0	0*	0	1	0	3	0	4
500k			0	0*	0	1		
1M					0	0*		
2.5M								

空欄：設定不可能です。

：設定は可能ですが、誤差が出ます。

\*：連続送信 / 受信はできません。

【注】 BRR の設定値は以下の計算式で求められます。

$$N = \frac{\text{OSC}}{8 \times 2^{2n} \times B} \times 10^6 - 1$$

B：ビットレート (bit/s)

N：ボーレートジェネレータのBRRの設定値 (0 ≤ N ≤ 255)

OSC：<sub>osc</sub>の値 (MHz)

n：ボーレートジェネレータの入力クロックのNo. (n=0、1、2、3)

(nとクロックの関係は表10.10を参照)

表 10.10 n とクロックの関係

n	クロック	SMR の設定値	
		CKS1	CKS0
0		0	0
1	/4	0	1
2	/16	1	0
3	/64	1	1

### 10.3.3 動作概要

SCI3 は、キャラクタ単位で同期をとりながら通信する調歩同期式モードと、クロックパルスにより同期をとりながら通信するクロック同期式モードの2方式でシリアル通信ができます。

調歩同期式モードとクロック同期式モードの選択、および通信フォーマットの選択は、SMR で行います。これを表 10.11 に示します。

また、SCI3 のクロックソースは、SMR の COM、および SCR3 の CKE1、CKE0 の組合せできまります。これを表 10.12 に示します。

#### (1) 調歩同期式モード

- ・データ長：7ビット/8ビットから選択可能
- ・パリティの付加、マルチプロセッサビットの付加、および1ビットまたは2ビットのストップビットの付加を選択可能（これらの組合せで送信/受信フォーマット、およびキャラクタ長を決定）
- ・受信時にフレーミングエラー（FER）、パリティエラー（PER）、オーバランエラー（OER）およびブレークの検出が可能
- ・クロックソース：内部クロック/外部クロックから選択可能
  - 内部クロックを選択した場合：ポーレートジェネレータのクロックで動作し、ビットレートと同じ周波数のクロックを出力可能
  - 外部クロックを選択した場合：ビットレートの16倍の周波数のクロック入力が必要（内蔵ポーレートジェネレータを使用しない）

#### (2) クロック同期式モード

- ・送信/受信フォーマット：8ビットデータ固定
- ・受信時にオーバランエラー（OER）の検出が可能
- ・クロックソース：内部クロック/外部クロックから選択可能
  - 内部クロックを選択した場合：ポーレートジェネレータのクロックで動作し、同期クロックを出力

外部クロックを選択した場合：内蔵ポーレートジェネレータを使用せず、入力された同期クロックで動作

表 10.11 SMR の設定値と送信 / 受信フォーマット

SMR					モード	送信 / 受信フォーマット						
ビット 7	ビット 6	ビット 2	ビット 5	ビット 3		データ長	マルチ プロセッサ ビット	パリティ ビット	ストップ ビット長			
COM	CHR	MP	PE	STOP								
0	0	0	0	0	調歩同期式 モード	8ビット データ	なし	なし	1ビット			
			1	0					2ビット			
			0	1					1ビット			
			1	0					2ビット			
			0	0					1ビット			
			1	0					2ビット			
	1	0	1	*		0	調歩同期式	8ビット	あり	なし	1ビット	
				*		1	モード(マルチ プロセッサフ ォーマット)	データ			2ビット	
				*		0	7ビット	1ビット				
				*		1	データ	2ビット				
		*	0	*		*	*	クロック同期 式モード	8ビット	なし	なし	なし
						*	*	データ				

\* : Don't care

表 10.12 SMR、SCR3 の設定とクロックソースの選択

SMR	SCR3		モード	送信 / 受信クロック	
ビット 7	ビット 1	ビット 0		クロック ソース	SCK <sub>3</sub> 端子の機能
COM	CKE1	CKE0			
0	0	0	調歩同期式	内部	入出力ポート (SCK <sub>3</sub> 端子を使用しません) ビットレートと同じ周波数のクロックを出力
		1	モード		
	1	0	外部	ビットレートの 16 倍の周波数のクロックを入力	
1	0	0	クロック同期	内部	同期クロックを出力
	1	0	式モード	外部	同期クロックを入力
0	1	1	リザーブ (この組合せは指定しないでください)		
1	0	1			
1	1	1			

## (3) 割込みと連続送信 / 受信

SCI3 は、RXI を使用した連続受信、また TXI を使用した連続送信が可能です。

表 10.13 にこれらの割込みについて示します。

表 10.13 送信 / 受信割込み

割込み	フラグ	割込み発生の条件	備考
RXI	RDRF RIE	シリアル受信が正常に行われ、RSR から RDR に受信データが転送されると RDRF が"1"となり、このとき RIE が"1"ならば RXI が許可され割込みが発生します。 (図 10.4(a)参照)	RXI の割込み処理ルーチンでは、RDR に転送された受信データを読み出し、RDRF を"0"にクリアします。上記の操作を次回の RSR の受信完了までに行うことで連続受信が可能です。
TXI	TDRE TIE	TSR の空 ( 前回の送信完了 ) を検出して、TDR にセットされた送信データが TSR に転送されると TDRE は"1"にセットされます。このとき TIE が"1"ならば TXI が許可され割込みが発生します。 (図 10.4(b)参照)	TXI の割込み処理ルーチンでは、次回の送信データを TDR に書き込み、TDRE を"0"にクリアします。上記の操作を TSR に転送したデータが送信終了するまでに行うことで連続送信が可能です。
TEI	TEND TEIE	TSR の送信キャラクタの最後尾ビットを送信したとき、TDRE が"1"ならば TEND は"1"にセットされます。このとき TEIE が"1"ならば TEI が許可され割込みが発生します。 (図 10.4(c)参照)	TEI は、TSR の送信キャラクタの最後尾ビットを送信したとき、TDR に次回の送信データが書き込まれていないことを示します。

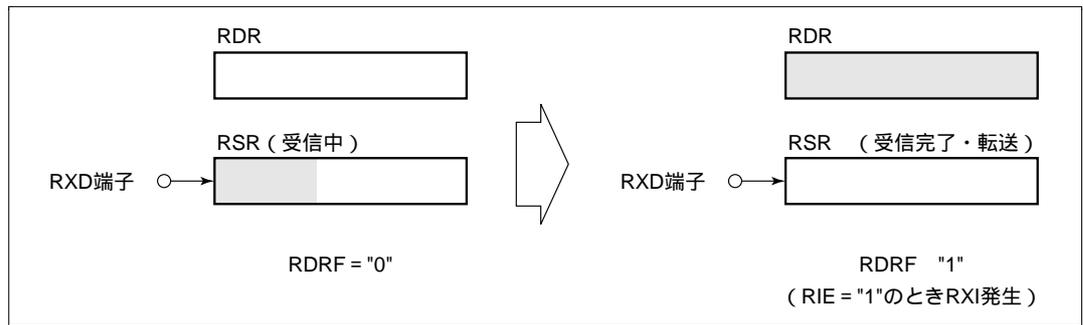


図 10. 4(a) RDRF のセットと RXI 割込み

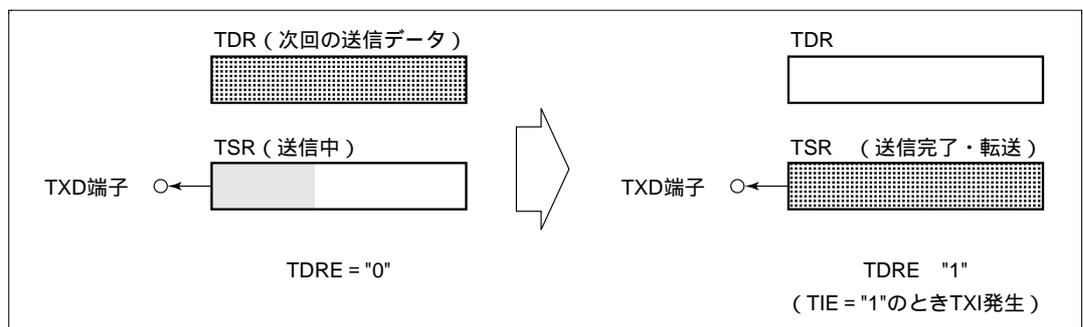


図 10. 4(b) TDRE のセットと TXI 割込み

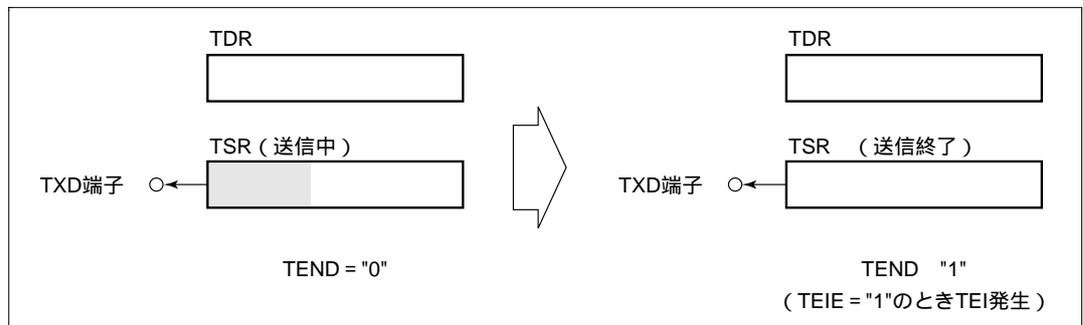


図 10. 4(c) TEND のセットと TEI 割込み

### 10.3.4 調歩同期式モード時の動作説明

調歩同期式モードは、通信開始を意味するスタートビットと通信終了を意味するストップビットとをデータに付加したキャラクタを送信 / 受信し、1キャラクタ単位で同期を取りながらシリアル通信を行うモードです。

SCI3 内部では、送信部と受信部は独立していますので、全二重通信を行うことができます。また、送信部と受信部が共にダブルバッファ構造になっていますので、送信 / 受信中にデータのリード / ライトができ、連続送信 / 受信が可能です。

#### (1) 送信 / 受信フォーマット

調歩同期式通信の通信データの一般的なフォーマットを図 10.5 に示します。

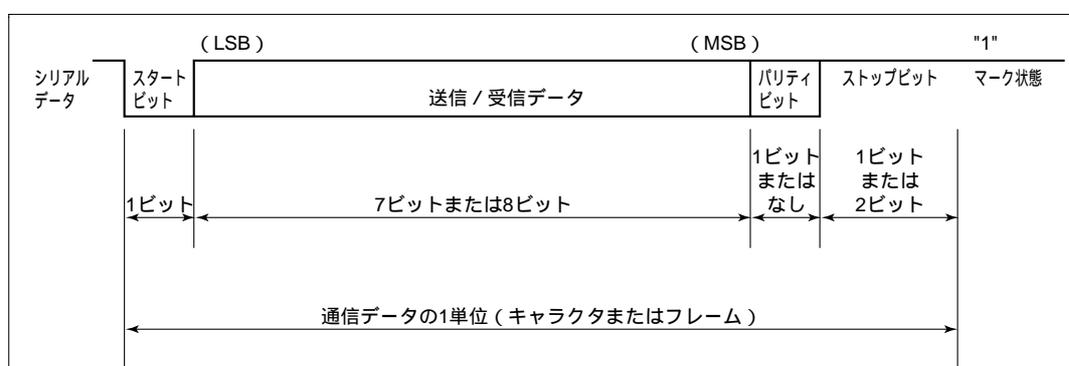


図 10.5 調歩同期式通信のデータフォーマット

調歩同期式通信では、通信回線は通常マーク状態 ("High"レベル) に保たれています。SCI3 は通信回線を監視し、スペース ("Low"レベル) になったところをスタートビットとみなしてシリアルデータ通信を開始します。

通信データの 1キャラクタは、スタートビット ("Low"レベル) から始まり、送信 / 受信データ (LSB ファースト: 最下位ビットから)、パリティビット ("High"または"Low"レベル)、最後にストップビット ("High"レベル) の順で構成されます。

調歩同期式モードでは、受信時にスタートビットの立下がりエッジで同期化を行います。また、データを 1ビット期間の 16 倍の周波数のクロックの 8 番目でサンプリングしますので、各ビットの中央で通信データを取り込みます。

調歩同期式モードで設定できる送信 / 受信フォーマットを、表 10.14 に示します。  
送信 / 受信フォーマットは 12 種類あり、SMR の設定により選択できます。

表 10.14 通信フォーマット (調歩同期式モード)

SMR				シリアル通信フォーマットとフレーム長													
CHR	PE	MP	STOP	1	2	3	4	5	6	7	8	9	10	11	12		
0	0	0	0	S	8ビットデータ								STOP				
0	0	0	1	S	8ビットデータ								STOP	STOP			
0	1	0	0	S	8ビットデータ								P	STOP			
0	1	0	1	S	8ビットデータ								P	STOP	STOP		
1	0	0	0	S	7ビットデータ							STOP					
1	0	0	1	S	7ビットデータ							STOP	STOP				
1	1	0	0	S	7ビットデータ							P	STOP				
1	1	0	1	S	7ビットデータ							P	STOP	STOP			
0	*	1	0	S	8ビットデータ								MPB	STOP			
0	*	1	1	S	8ビットデータ								MPB	STOP	STOP		
1	*	1	0	S	7ビットデータ							MPB	STOP				
1	*	1	1	S	7ビットデータ							MPB	STOP	STOP			

\* : Don't care

< 記号説明 >

- S : スタートビット
- STOP : ストップビット
- P : パリティビット
- MPB : マルチプロセスビット

## (2) クロック

SCI3の送受信クロックは、SMRのCOMとSCR3のCKE1、CKE0の設定により、内蔵ポーレートジェネレータの生成した内部クロックまたは、SCK<sub>3</sub>端子から入力された外部クロックの2種類から設定できます。クロックソースの選択については表 10.12 を参照してください。

外部クロックをSCK<sub>3</sub>端子に入力する場合には、使用するビットレートの16倍の周波数のクロックを入力してください。

内部クロックで動作させるとき、SCK<sub>3</sub>端子からクロックを出力することができます。このとき出力されるクロックの周波数はビットレートと等しく、位相は図 10.6 に示すように送信/受信データの各ビットの中央でクロックが立上がります。

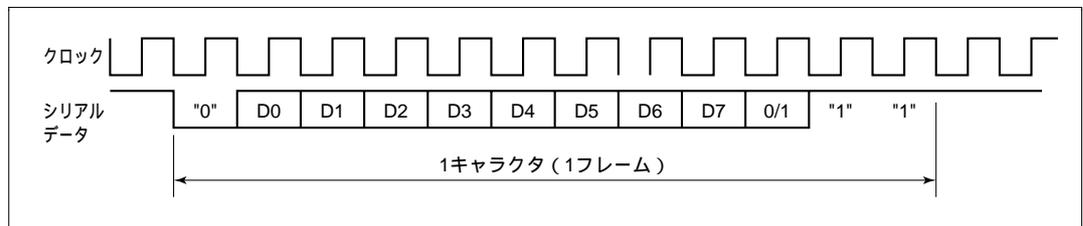


図 10.6 出力クロックと通信データの位相関係 (調歩同期式モード)  
(8ビットデータ/パリティあり/2ストップビットの例)

## (3) データの送信/受信動作

### (a) SCI3のイニシャライズ

データの送信/受信前には、まずSCR3のTEおよびREを"0"にクリアした後、以下の手順にしたがってイニシャライズしてください。

**【注】** 動作モードの変更、通信フォーマットの変更などの場合には、かならずTEおよびREを"0"にクリアしてから変更してください。TEを"0"にクリアすると、TDREは"1"にセットされます。REを"0"にクリアしても、RDRF、PER、FER、OERの各フラグ、およびRDRの内容は保持されますので注意してください。

調歩同期式モードで外部クロックを使用している場合には、イニシャライズを含めた動作中にクロックを止めないでください。クロック同期式モードで外部クロックを使用している場合には、イニシャライズ中にクロックを供給しないでください。

図 10.7 に SCI3 をイニシャライズするときのフローチャートの例を示します。

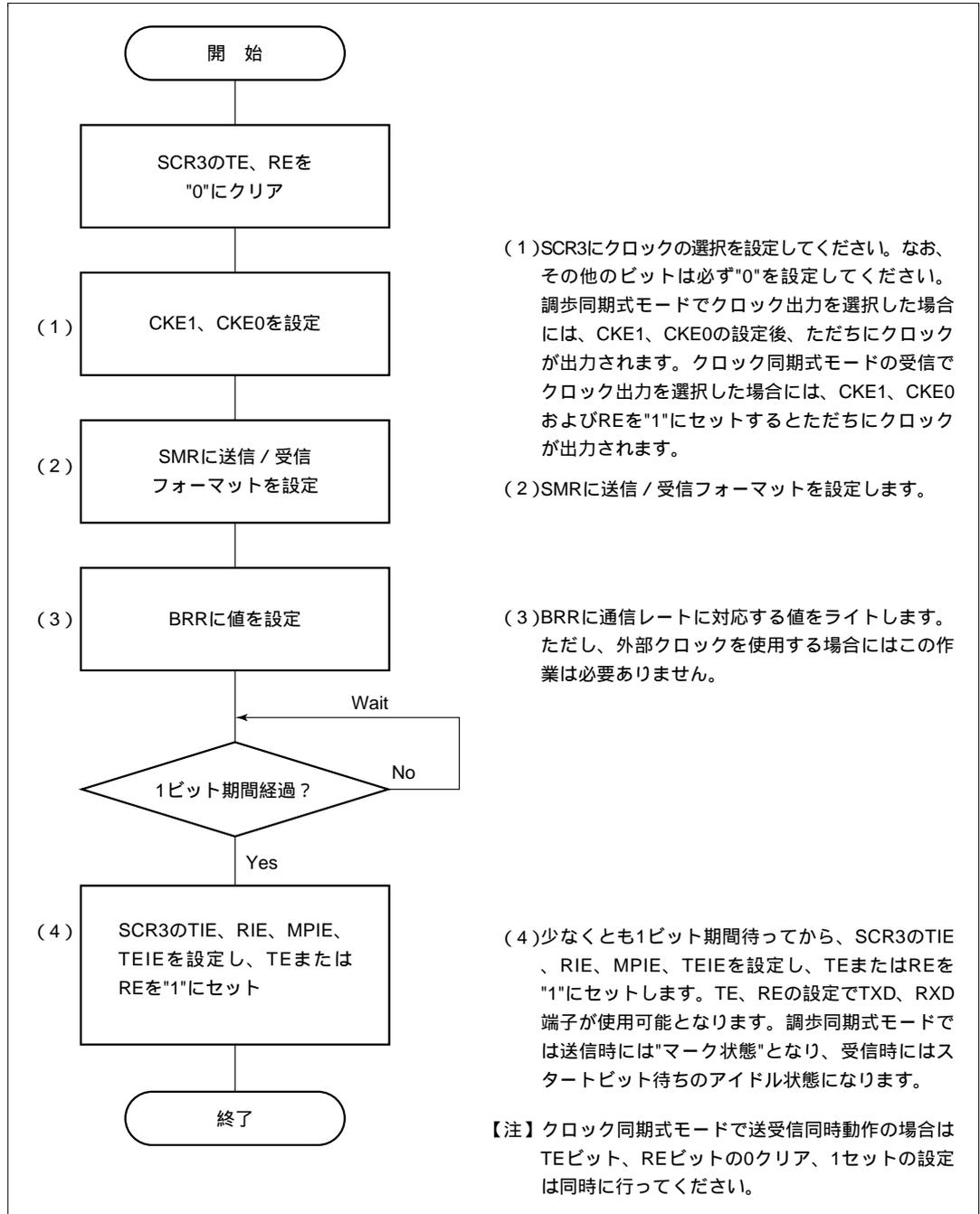


図 10.7 SCI3 をイニシャライズするときのフローチャートの例

## (b) データ送信

図 10.8 にデータ送信のフローチャートの例を示します。

データ送信は SCI3 のイニシャライズ後、以下の手順にしたがって行ってください。

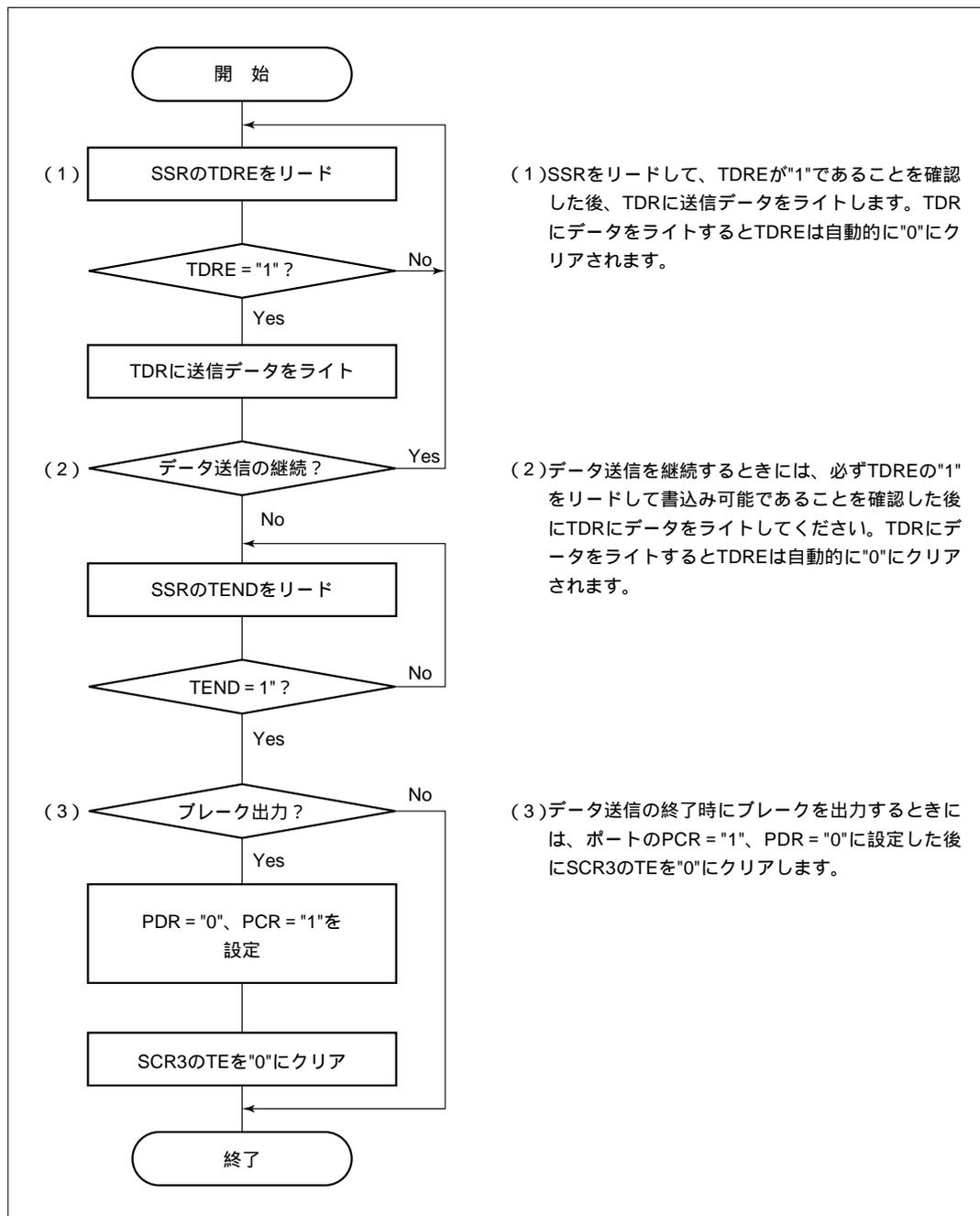


図 10.8 データ送信のフローチャートの例 (調歩同期式モード)

SCI3 はデータ送信時に以下のように動作します。

SCI3 は、SSR の TDRE を監視し、"0"であると TDR にデータが書き込まれたと認識し、TDR から TSR にデータを転送します。その後、TDRE を"1"にセットして送信を開始します。このとき、SCR3 の TIE が"1"にセットされていると TXI を発生します。

シリアルデータは、表 10.14 に示す通信フォーマットに従い TXD 端子から送信されます。

その後、ストップビットを送り出すタイミングで TDRE をチェックします。

TDRE が"0"であると TDR から TSR にデータを転送し、ストップビット送出後、次のフレームの送信を開始します。TDRE が"1"であると SSR の TEND に"1"をセットし、ストップビット送出後、"1"を出力する"マーク状態"になります。このとき SCR3 の TEIE が"1"にセットされていると TEI を発生します。

図 10.9 に調歩同期式モードの送信時の動作例を示します。

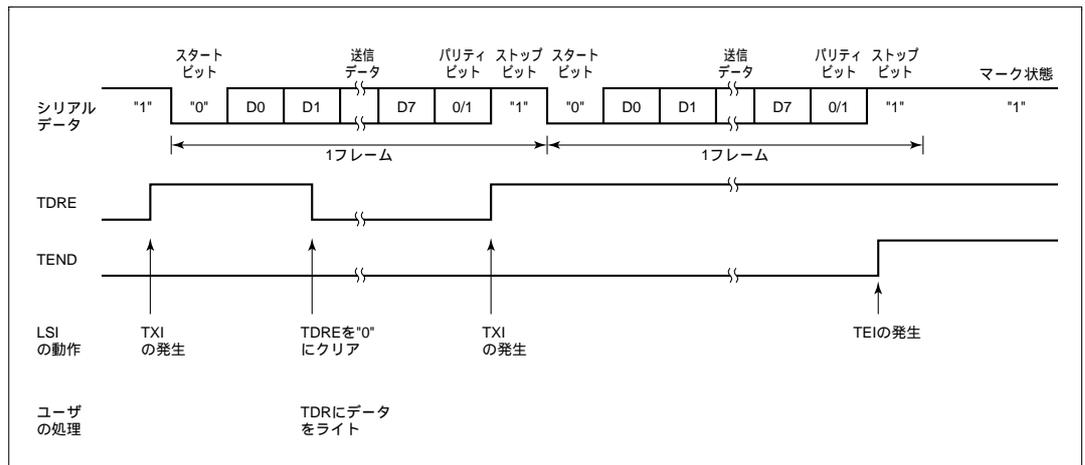


図 10.9 調歩同期式モードの送信時の動作例  
(8ビットデータ/パリティあり/1ストップビットの例)

(c) データ受信

図 10.10 にデータ受信のフローチャートの例を示します。

データ受信は SCI3 のイニシャライズ後、以下の手順にしたがって行ってください。

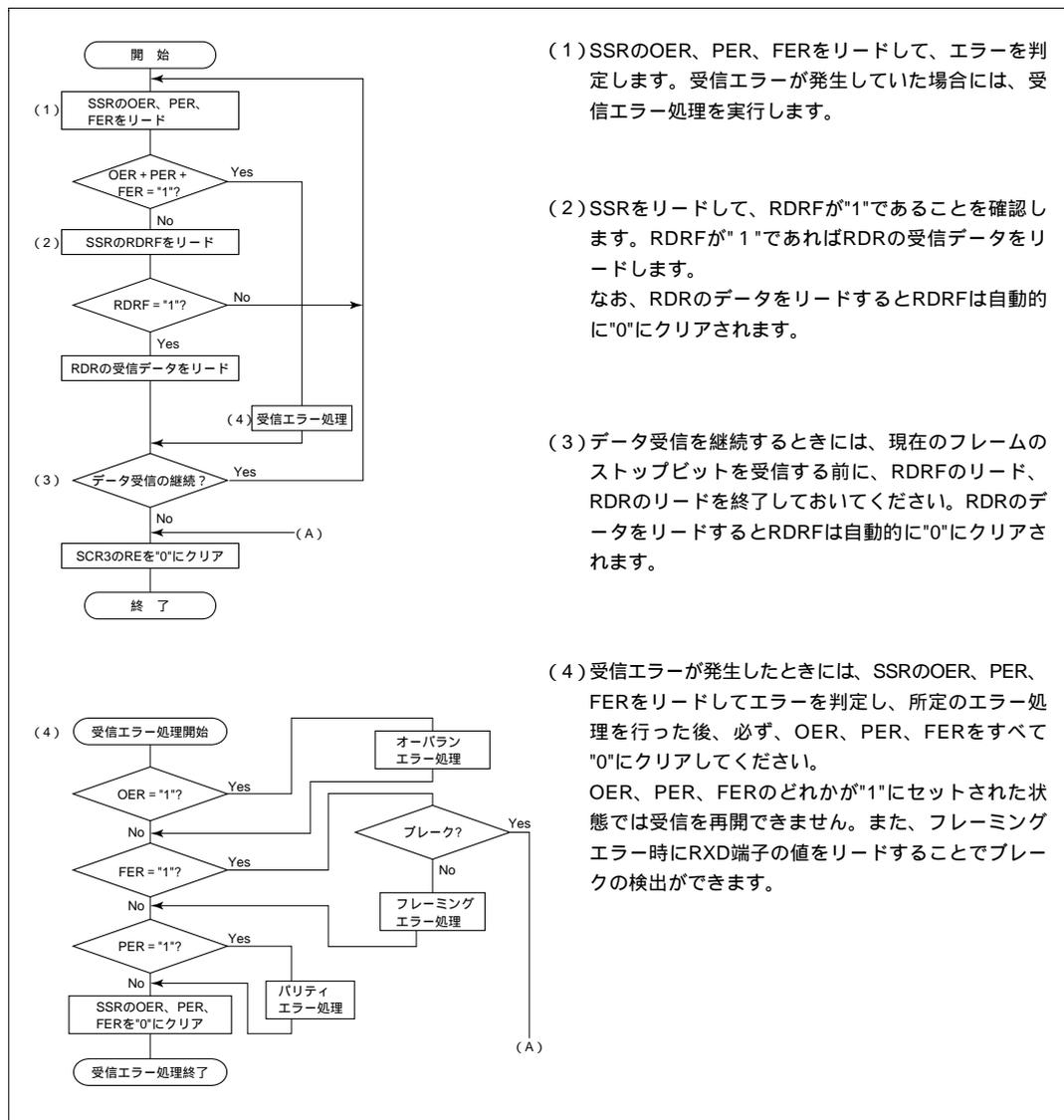


図 10.10 データ受信のフローチャートの例 (調歩同期式モード)

SCI3 は受信時に以下のように動作します。

SCI3 は通信回線を監視し、スタートビットの"0"を検出すると内部を同期化し受信を開始します。受信は表 10.14 に示す通信フォーマットに従い、まず受信したデータを RSR の LSB から MSB の順にセットし、次に、パリティビットおよびストップビットを受信します。受信後、SCI3 は以下のチェックを行います。

・パリティチェック：

受信データの"1"の数をチェックし、これが SMR の PM で設定した偶数 / 奇数パリティになっているかをチェックします。

・ストップビットチェック：

ストップビットが"1"であるかをチェックします。ただし、2ストップビットの場合、1ビット目のストップビットのみをチェックします。

・ステータスチェック：

RDRF が"0"であり、受信データを RSR から RDR に転送できる状態であることをチェックします。

以上のチェックの結果受信エラーがなかったとき RDRF が"1"にセットされ、RDR に受信データが格納されます。このとき SCR3 の RIE が"1"にセットされていると RXI を発生します。一方、エラーチェックで受信エラーを検出すると、各エラーに対応して、OER、PER、FER が"1"にセットされます。また RDRF はデータを受信する前の状態を保ちます。このとき、SCR3 の RIE が"1"にセットされていると ERI を発生します。

表 10.15 に受信エラーの検出条件と受信データの処理を示します。

【注】 受信エラーがセットされた状態では、以後の受信動作ができません。したがって、受信を継続する前に必ず OER、FER、PER および RDRF を"0"にクリアしてください。

表 10.15 受信エラーの検出条件と受信データの処理

受信エラー	略称	検出条件	受信データの処理
オーバランエラー	OER	SSR の RDRF が"1"にセットされたまま次のデータ受信を完了したとき	RSR から RDR に受信データは転送されません。
フレーミングエラー	FER	ストップビットが"0"のとき	RSR から RDR に受信データが転送されません。
パリティエラー	PER	SMR で設定した偶数 / 奇数パリティの設定と受信したデータが異なるとき	RSR から RDR に受信データが転送されません。

調歩同期モードの受信時の動作例を図 10.11 に示します。

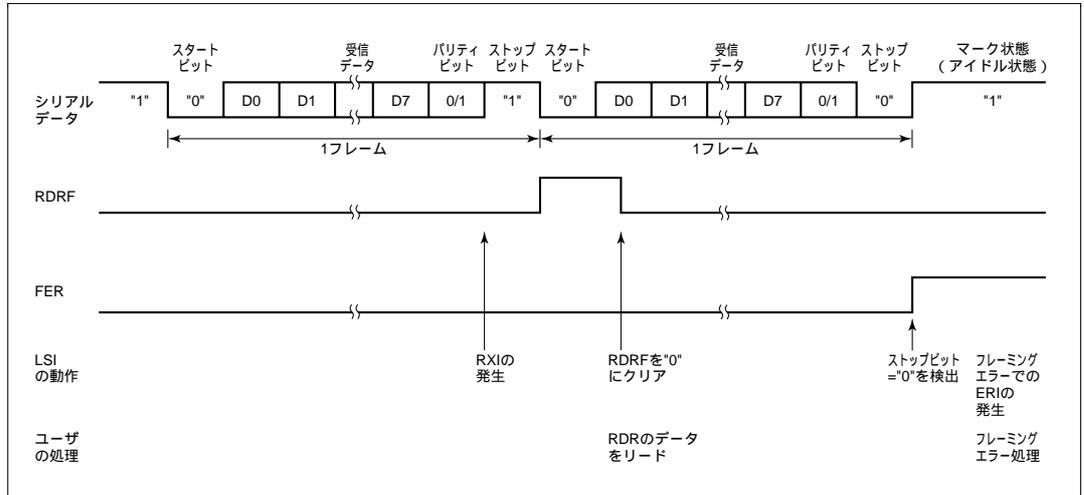


図 10.11 調歩同期モードの受信時の動作例  
(8 ビットデータ / パリティあり / 1 ストップビットの例)

### 10.3.5 クロック同期式モード時の動作説明

クロック同期式モードは、クロックパルスに同期してデータを送信、または受信するモードです。クロック同期式モードは、高速シリアル通信に適しています。

SCI3 内部では、送信部と受信部は独立していますので、クロックを共有することで全二重通信を行うことができます。

送信部 / 受信部は共にダブルバッファ構造になっていますので、送信中にデータのライトができ、連続送信が可能です。また、受信中にデータのリードができ連続受信が可能です。

#### (1) 送信 / 受信フォーマット

クロック同期式通信の通信データの一般的なフォーマットを図 10.12 に示します。

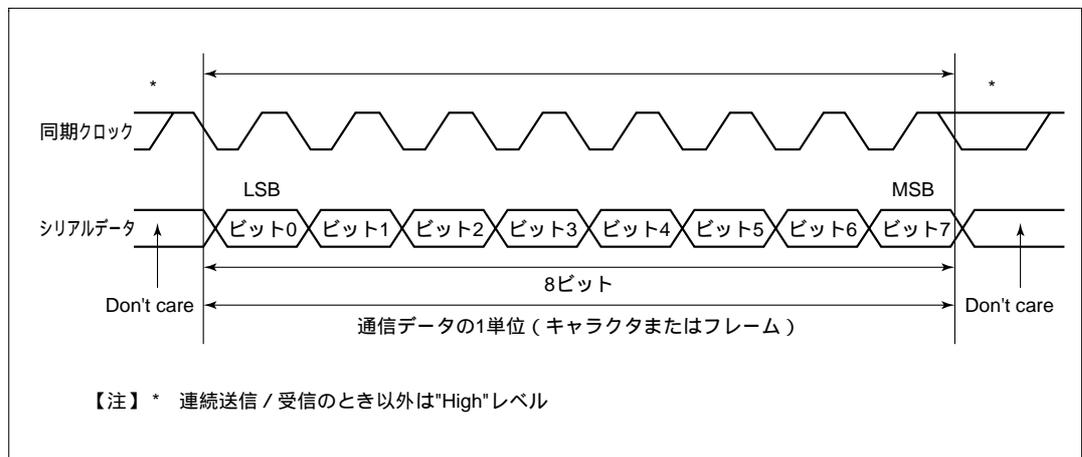


図 10.12 クロック同期式通信のデータフォーマット

クロック同期式通信では、通信回線のデータは同期クロックの立下がりから次の立下がりまで出力されます。また、同期クロックの立上がりエッジでデータの確定が保証されます。

通信データの 1 キャラクタは、LSB から始まり最後に MSB の順で構成されます。MSB 出力後の通信回線は MSB の状態を保ちます。

クロック同期式モードでは、SCI3 は、受信時に同期クロックの立上がりに同期してデータを取り込みます。

送信 / 受信フォーマットは 8 ビットデータ固定です。パリティビットやマルチプロセッサビットの付加はできません。

## (2) クロック

SMRのCOMとSCR3のCKE1、CKE0の設定により、内蔵ポーレートジェネレータの生成した内部クロック、またはSCK<sub>3</sub>端子から入力された外部同期クロックの2種類から選択できます。クロックソースの選択については表10.12を参照してください。

内部クロックで動作させるとき、SCK<sub>3</sub>端子からは同期クロックが出力されます。同期クロックは1キャラクタの送信/受信で8パルス出力され、送信および受信を行わないときには"High"レベルに固定されます。

## (3) データの送信/受信動作

### (a) SCI3のイニシャライズ

データの送信/受信前には、「10.3.4(3)(a)SCI3のイニシャライズ」の説明および、図10.7の例にしたがってSCI3をイニシャライズしてください。

### (b) データ送信

図10.13にデータ送信のフローチャートの例を示します。

データ送信はSCI3のイニシャライズ後、以下の手順にしたがって行ってください。

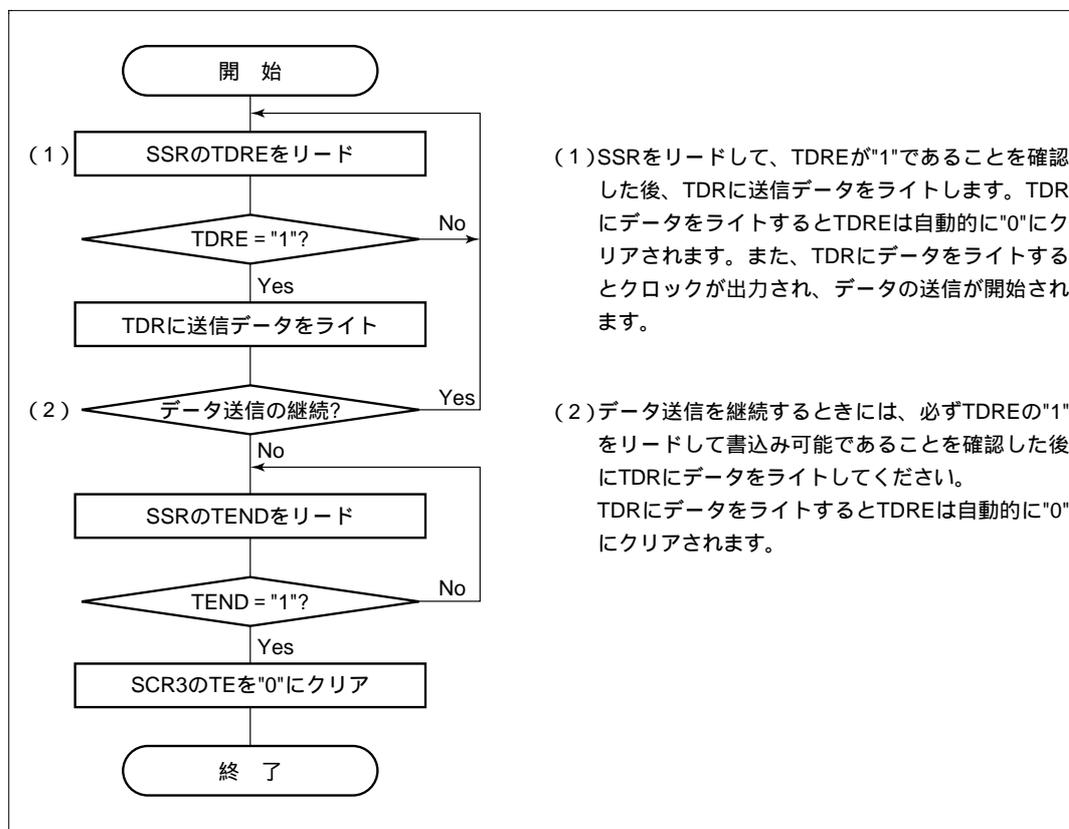


図10.13 データ送信のフローチャートの例(クロック同期式モード)

SCI3 はデータ送信時に以下のように動作します。

SCI3 は、SSR の TDRE を監視し、"0"であると TDR にデータが書き込まれたと認識し、TDR から TSR にデータを転送します。その後、TDRE を"1"にセットして送信を開始します。このとき、SCR3 の TIE が"1"にセットされていると TXI を発生します。

クロック出力モードに設定したときには、SCI3 は同期クロックを 8 パルス出力します。外部クロックに設定したときには、入力クロックに同期してデータを出力します。

シリアルデータは、LSB (ビット 0) から MSB (ビット 7) の順に TXD 端子から送信されます。

その後、MSB (ビット 7) を送り出すタイミングで TDRE をチェックします。

TDRE が"0"であると TDR から TSR にデータを転送し、次のフレームの送信を開始します。TDRE が"1"であると SSR の TEND に"1"をセットし、MSB (ビット 7) 送出後、状態を保持します。このとき SCR3 の TEIE が"1"にセットされていると TEI を発生します。

送信終了後は、SCK<sub>3</sub> 端子は"High"レベル固定になります。

**【注】** データ受信のステータスを示すエラーフラグ (OER、FER、PER) が"1"にセットされた状態では送信は行えませんので、送信の前には、エラーフラグ (OER、FER、PER) が"0"にクリアされていることを確認してください。

図 10.14 にクロック同期式モードの送信時の動作例を示します。

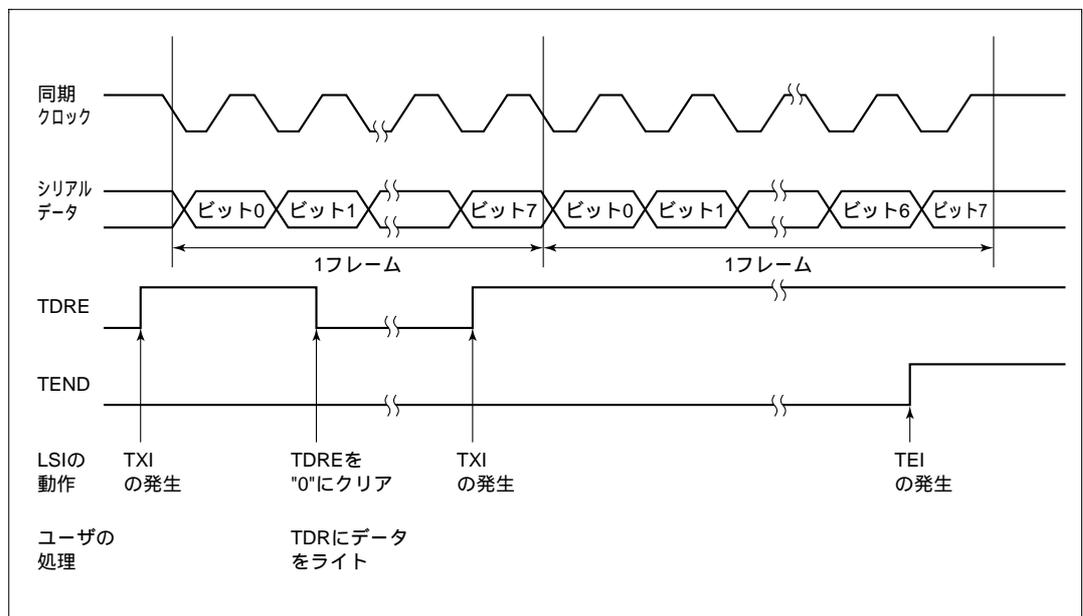


図 10.14 クロック同期式モードの送信時の動作例

## (c) データ受信

図 10.15 にデータ受信のフローチャートの例を示します。

データ受信は SCI3 のイニシャライズ後、以下の手順にしたがって行ってください。

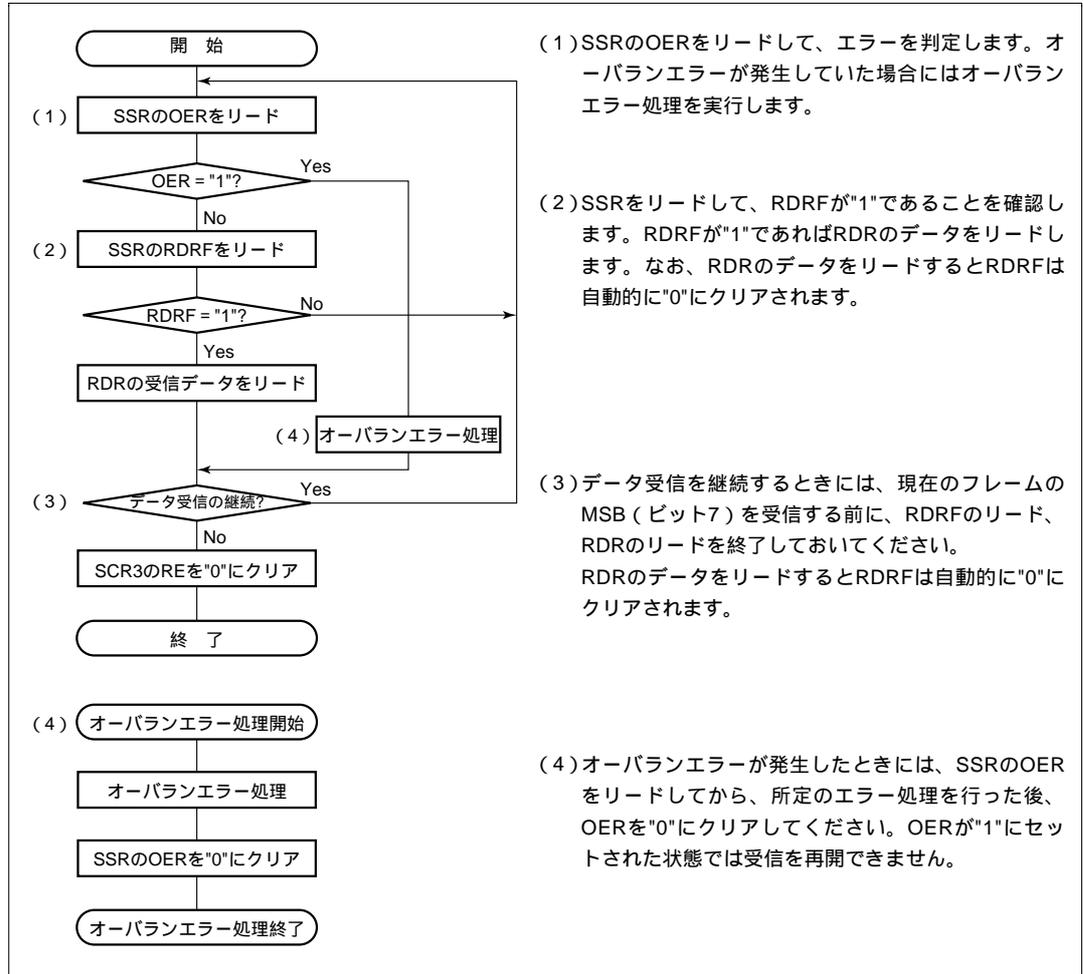


図 10.15 データ受信フローチャートの例 (クロック同期式モード)

SCI3 は受信時に以下のように動作します。

SCI3 は同期クロックの入力または、出力に同期して内部を初期化し、受信を開始します。

受信したデータを RSR の LSB から MSB の順にセットします。

受信後、SCI3 は、RDRF が"0"であり、受信データを RSR から RDR に転送できる状態であるかをチェックします。

このチェックの結果オーバーランエラーがなかったとき RDRF が"1"にセットされ、RDR に受信データが格納されます。

このとき、SCR3 の RIE が"1"にセットされていると RXI を発生します。一方、エラーチェックでオーバーランエラーを検出すると OER が"1"にセットされます。また、RDRF は"1"にセットされた状態を保ちます。このとき、SCR3 の RIE が"1"にセットされていると ERI を発生します。

オーバーランエラーの検出条件と受信データの処理については、表 10.15 を参照してください。

【注】 受信エラーがセットされた状態では、以後の受信動作ができません。したがって、受信を継続する前に必ず OER、FER、PER および RDRF を"0"にクリアしてください。

図 10.16 にクロック同期式モードの受信時の動作例を示します。

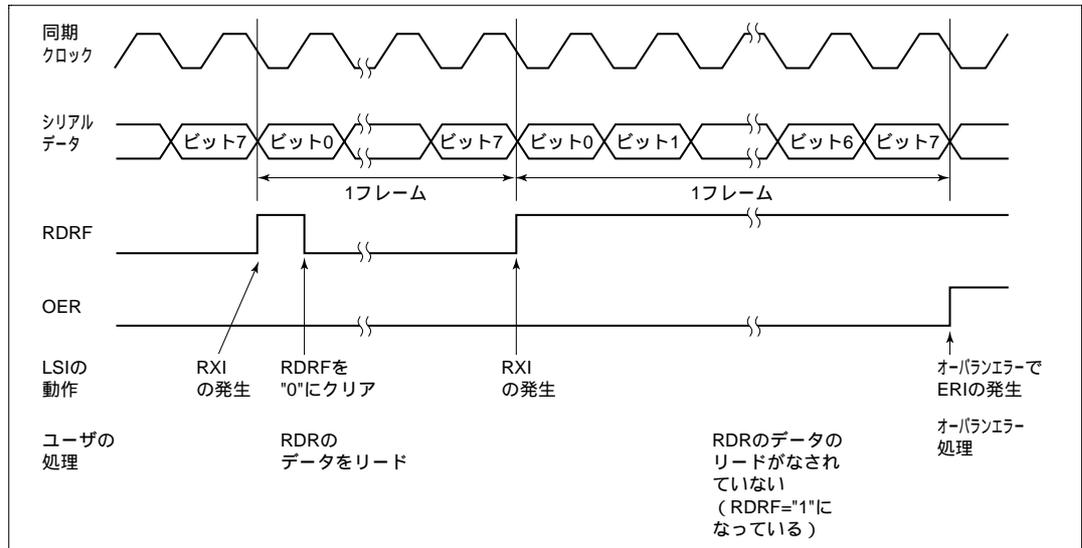


図 10.16 クロック同期式モードの受信時の動作例

## (d) データ送受信同時動作

図 10.17 にデータ送受信同時動作のフローチャートの例を示します。

データ送受信同時動作は SCI3 のイニシャライズ後、以下の手順にしたがって行ってください。

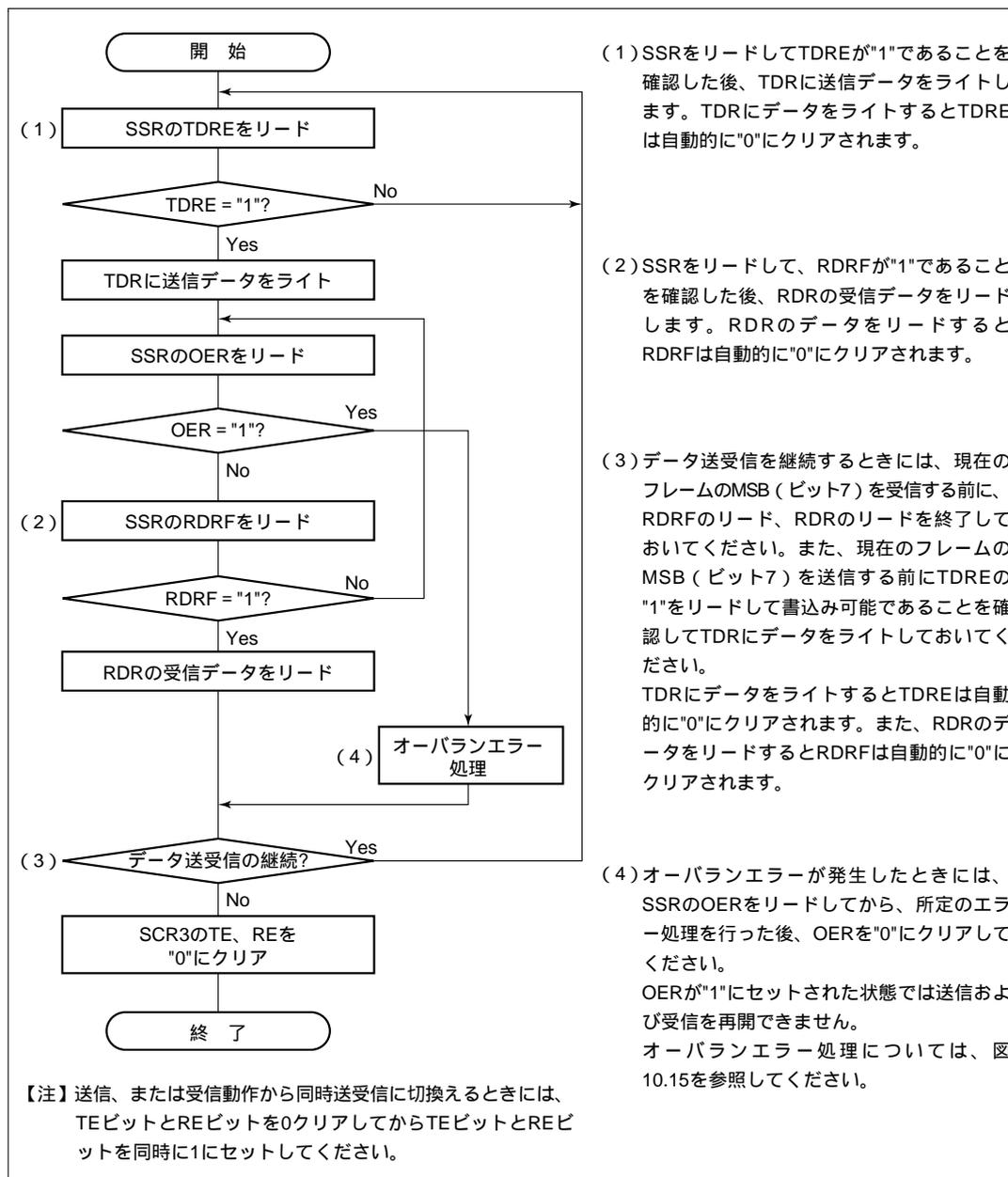


図 10.17 データ送受信同時動作のフローチャートの例 (クロック同期式モード)

- 【注】 1 送信から同時送受信へ切換えるときには、SCI3 が送信終了状態であること、TDRE および TEND が"1"にセットされていることを確認した後、TE を"0"にクリアしてから TE および RE を"1"にセットしてください。
- 2 受信から同時送受信へ切換えるときには、SCI3 が受信完了状態であることを確認し、RE を"0"にクリアしてから RDRF およびエラーフラグ ( OER、FER、PER ) が"0"にクリアされていることを確認した後、TE および RE を"1"にセットしてください。

### 10.3.6 マルチプロセッサ通信機能

マルチプロセッサ通信機能とは、調歩同期式モードでマルチプロセッサビットを付加したフォーマット(マルチプロセッサフォーマット)でシリアルデータ通信を行うことにより、複数のプロセッサ間で通信回線を共有してデータの送受信を行う機能です。

マルチプロセッサ通信を行うとき、受信局は各々固有の ID コードが割り付けられています。シリアル通信サイクルは、受信局を指定する ID 送信サイクルと指定された受信局へ通信データを送信するデータ送信サイクルの 2 つから構成されます。この ID 送信サイクルとデータ送信サイクルの区別は、マルチプロセッサビットで行います。マルチプロセッサビットが"1"のとき ID 送信サイクル、"0"のときデータ送信サイクルとなります。

送信局は、まずシリアルデータ通信を行いたい受信局の ID コードに、マルチプロセッサビット"1"を付加した通信データを送信します。続いて、送信データにマルチプロセッサビット"0"を付加した通信データを送信します。受信局は、マルチプロセッサビットが"1"の通信データを受信すると、自局の ID と比較し一致した場合は続いて送信される通信データを受信します。一致しなかった場合は再びマルチプロセッサビットが"1"の通信データが送信されるまで通信データを読みとばします。

このようにして複数のプロセッサ間のデータ送受信が行われます。

図 10.18 にマルチプロセッサフォーマットを使用したプロセッサ間通信の例を示します。

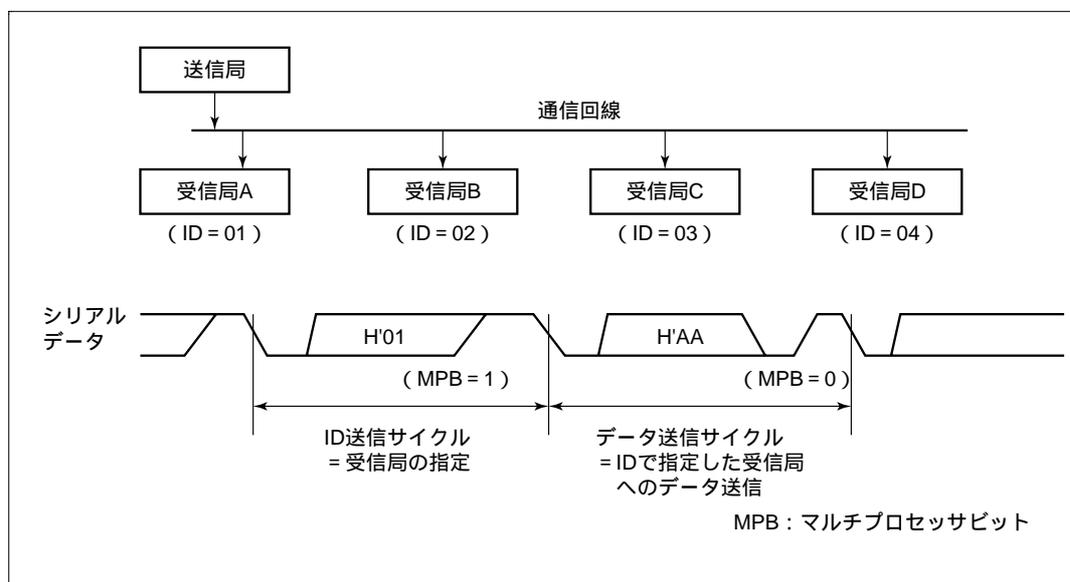


図 10.18 マルチプロセッサフォーマットを使用したプロセッサ間通信の例  
(受信局 A へのデータ H'AA の送信の例)

送信 / 受信フォーマットは 4 種類を選択できます。マルチプロセッサフォーマットを指定した場合は、パリティビットの指定は無効です。詳細は表 10.14 を参照してください。

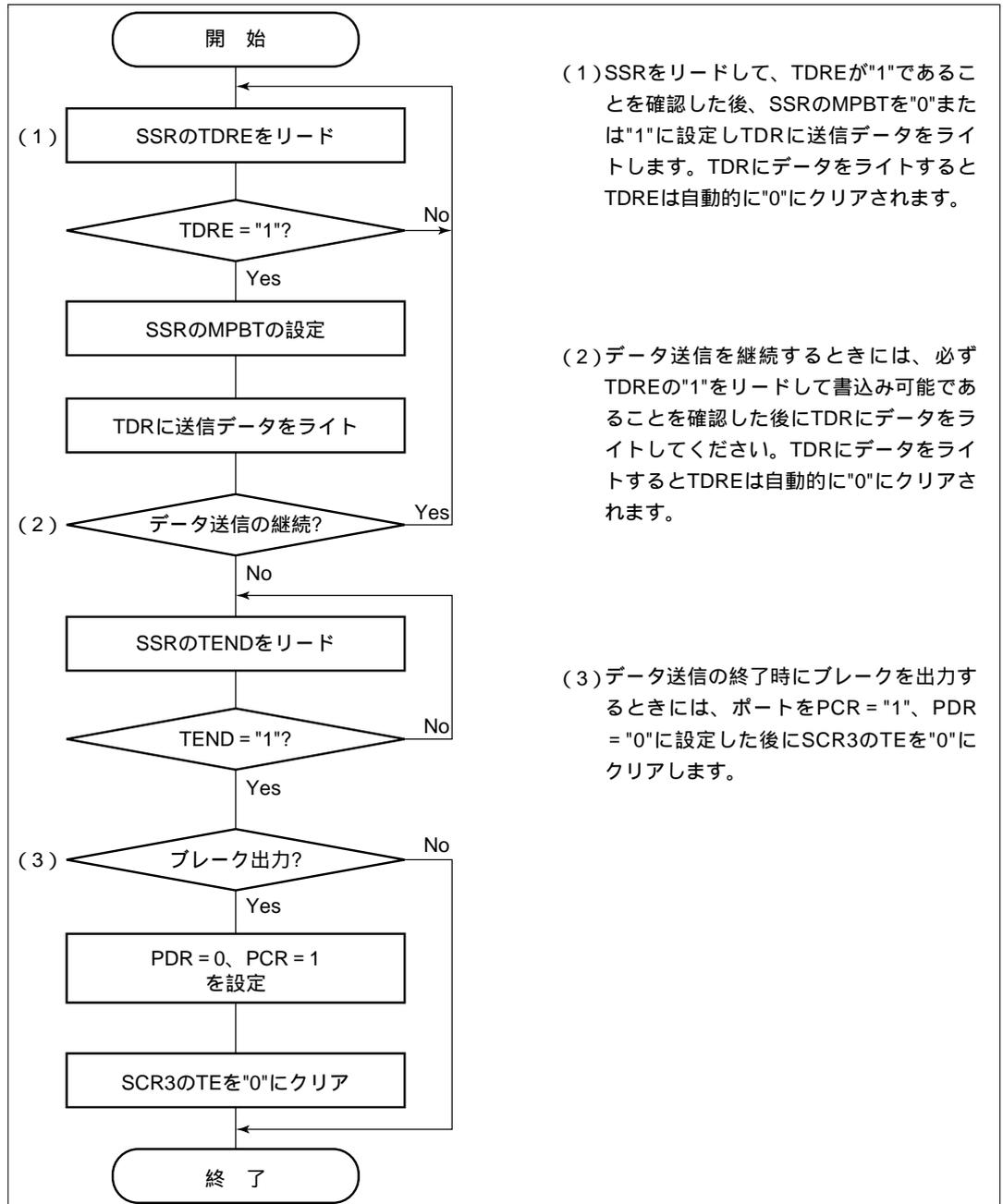
マルチプロセッサ通信を行うときのクロックについては、「10.3.4 調歩同期式モード

時の動作説明」を参照してください。

(a) マルチプロセッサデータ送信

図 10.19 にマルチプロセッサデータ送信のフローチャートの例を示します。

マルチプロセッサデータ送信は SCI3 をイニシャライズ後、以下の手順にしたがって行ってください。



(1) SSRをリードして、TDREが"1"であることを確認した後、SSRのMPBTを"0"または"1"に設定しTDRに送信データをライトします。TDRにデータをライトするとTDREは自動的に"0"にクリアされます。

(2) データ送信を継続するときには、必ずTDREの"1"をリードして書き込み可能であることを確認した後にTDRにデータをライトしてください。TDRにデータをライトするとTDREは自動的に"0"にクリアされます。

(3) データ送信の終了時にブレークを出力するときには、ポートをPCR = "1"、PDR = "0"に設定した後にSCR3のTEを"0"にクリアします。

図 10.19 マルチプロセッサデータ送信のフローチャートの例

SCI3 はデータ送信時に以下のように動作します。

SCI3 は、SSR の TDRE を監視し、"0"であると TDR にデータが書き込まれたと認識し、TDR から TSR にデータを転送します。その後、TDRE を"1"にセットして、送信を開始します。このとき、SCR3 の TIE が"1"にセットされていると TXI を発生します。

シリアルデータは、表 10.14 に示す通信フォーマットに従い、TXD 端子から送信されます。

その後、ストップビットを送り出すタイミングで TDRE をチェックします。

TDRE が"0"であると TDR から TSR にデータを転送し、ストップビット送出後、次のフレームの送信を開始します。TDRE が"1"であると SSR の TEND に"1"をセットし、ストップビット送出後、"1"を出力する"マーク状態"になります。このとき SCR3 の TEIE が"1"にセットされていると TEI を発生します。

図 10.20 にマルチプロセッサフォーマットの送信時の動作例を示します。

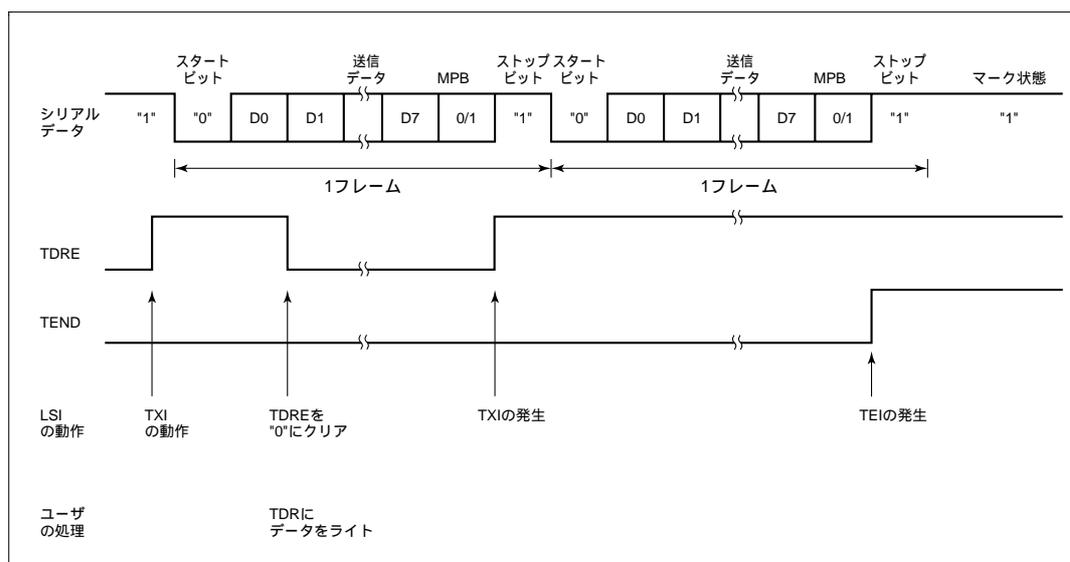


図 10.20 マルチプロセッサフォーマットの送信時の動作例

(8 ビットデータ / マルチプロセッサビットあり / 1 ストップビットの例)

(b) マルチプロセッサデータ受信

図 10.21 にマルチプロセッサデータ受信のフローチャートの例を示します。

マルチプロセッサデータ受信は SCI3 をイニシャライズ後、以下の手順にしたがっておこなってください。

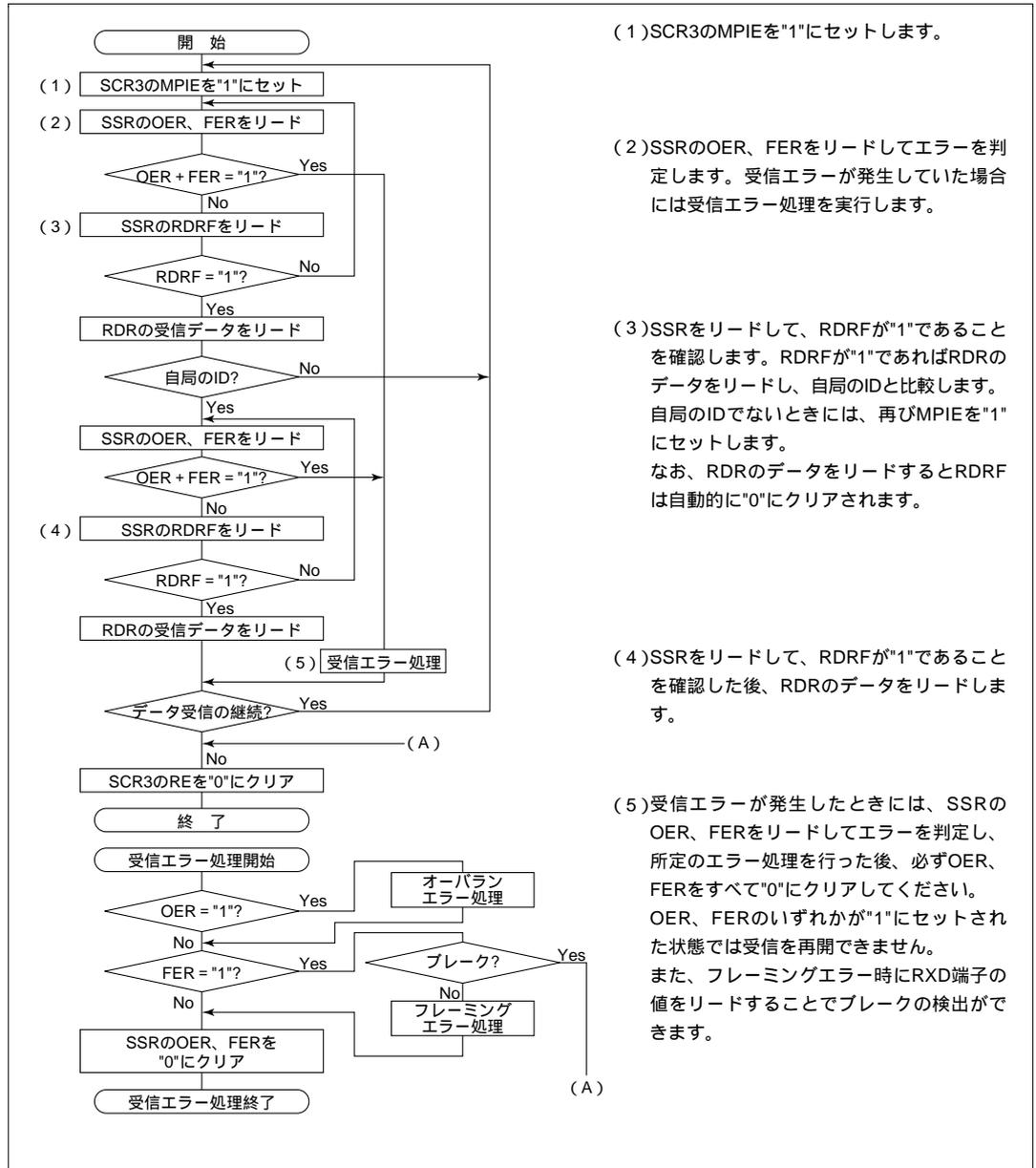


図 10.21 マルチプロセッサデータ受信のフローチャートの例

図 10.22 にマルチプロセッサフォーマットの受信時の動作例を示します。

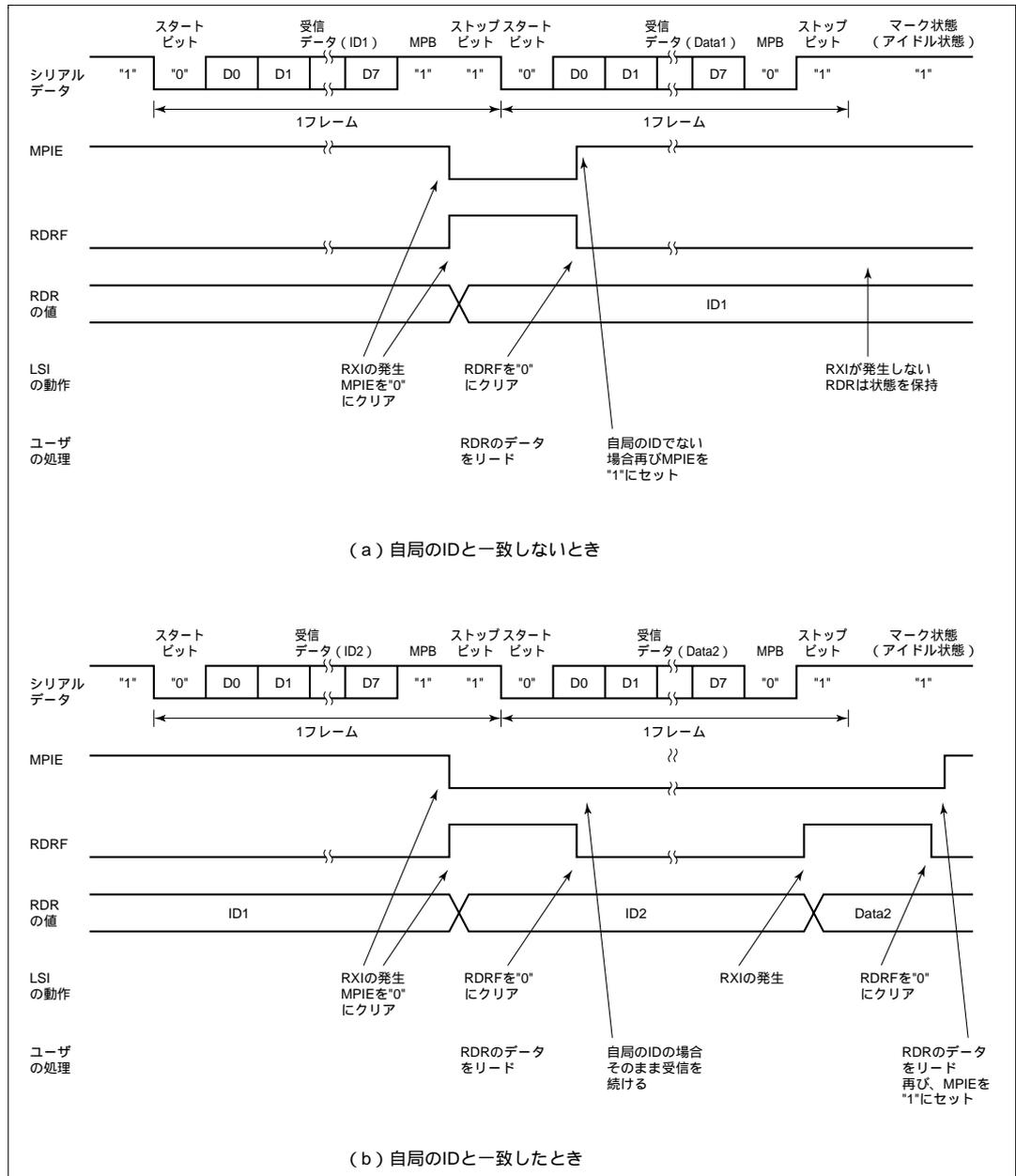


図 10.22 マルチプロセッサフォーマットの受信時の動作例  
(8ビットデータ/マルチプロセッサビットあり/1ストップビットの例)

### 10.3.7 割込み要因

SCI3 の割込み要因には、送信終了、送信データエンプティ、受信データフルおよび 3 種類の受信エラー（オーバランエラー、フレーミングエラー、パリティエラー）の計 6 種類があり、共通のベクタアドレスが割り付けられています。

表 10.16 に各割込み要求の内容を示します。

表 10.16 SCI3 割込み要求の内容

割込みの略称	割込み要求の内容	ベクタアドレス
RXI	受信データフル（RDRF）による割込み要求	H'0024
TXI	送信データエンプティ（TDRE）による割込み要求	
TEI	送信終了（TEND）による割込み要求	
ERI	受信エラー（OER、FER、PER）による割込み要求	

各割込み要求は、SCR3 の TIE、RIE で許可 / 禁止できます。

SSR の TDRE が "1" にセットされると、TXI が発生します。SSR の TEND が "1" にセットされると、TEI が発生します。この 2 つの割込みは送信時に発生します。

SSR の TDRE は初期値が "1" になっています。したがって送信データを TDR へ転送する前に SCR3 の TIE を "1" にセットして送信データエンプティ割込み要求（TXI）を許可すると、送信データが準備されていなくても TXI が発生します。

また、SSR の TEND は初期値が "1" になっています。したがって、送信データを TDR へ転送する前に SCR3 の TEIE を "1" にセットして送信終了割込み要求（TEI）を許可すると、送信データが送信されていなくても TEI が発生します。

送信データを TDR へ転送する処理を割込み処理ルーチンの中で行うようにすることで、これらの割込み要求を有効に利用できます。

一方、これらの割込み要求（TXI、TEI）の発生を防ぐためには、送信データを TDR へ転送した後に、これらの割込み要求に対応する許可ビット（TIE、TEIE）を "1" にセットしてください。

SSR の RDRF が "1" にセットされると RXI が発生します。OER、PER、FER のいずれかが "1" にセットされると ERI が発生します。この 2 つの割込み要求は受信時に発生します。

割込みに関する詳細は「3.3 割込み」を参照してください。

### 10.3.8 使用上の注意事項

SCI3 を使用する際は、以下のことに注意してください。

#### (1) TDR へのライトと TDRE の関係について

SSR の TDRE はシリアル送信するデータが TDR に準備されていないことを示すステータスフラグです。TDR へデータを書き込むと TDRE は自動的に"0"にクリアされます。また SCI3 が TDR から TSR にデータを転送すると、TDRE が"1"にセットされます。

TDR へのデータのライトは、TDRE の状態にかかわらず行うことができますが、TDRE が"0"の状態では新しいデータを TDR に書き込むと、TDR に格納されていた前のデータは、まだ TSR に転送されていない場合失われてしまいます。したがって、シリアル送信を確実に行うために TDR への送信データのライトは、必ず TDRE が"1"にセットされていることを確認してから 1 回だけ行う (2 回以上ライトしない) ようにしてください。

#### (2) 複数の受信エラーを同時に検出した場合の動作について

複数の受信エラーを同時に検出した場合、SSR の各ステータスフラグの状態は、表 10.17 に示すようにセットされます。オーバランエラーを検出した場合には RSR から RDR へのデータ転送は行われず、受信データは失われます。

表 10.17 SSR のステータスフラグの状態と受信データの転送

SSR のステータスフラグ				受信データ転送		受信エラーの状態
RDRF*	OER	FER	PER	RSR	RDR	
1	1	0	0	x		オーバランエラー
0	0	1	0			フレーミングエラー
0	0	0	1			パリティエラー
1	1	1	0	x		オーバランエラー + フレーミングエラー
1	1	0	1	x		オーバランエラー + パリティエラー
0	0	1	1			フレーミングエラー + パリティエラー
1	1	1	1	x		オーバランエラー + フレーミングエラー + パリティエラー

: RSR RDR に受信データを転送します。

x : RSR RDR に受信データを転送しません。

【注】 \* RDRF は、データ受信前の状態を保持します。

### (3) ブレークの検出と処理について

フレーミングエラー検出時に、RXD 端子の値を直接リードすることでブレークを検出できます。ブレークではRXD 端子からの入力がすべて"0"になりますので、FERがセットされ、またPERもセットされる可能性があります。

SCI3は、ブレークを受信した後も受信動作を続けます。したがってFERを"0"にクリアしてもふたたびFERが"1"にセットされますので注意してください。

### (4) マーク状態とブレークの送出

PMR6のTXD ビットが"0"のとき、TXD 端子はPDRとPCRにより入出力方向とレベルが決まるI/Oポートになります。これを利用してTXD 端子をマーク状態にしたりデータ送信時にブレークの送出をすることができます。

TEを"1"にセットするまで、通信回線をマーク状態("1"の状態)にするためには、PCR="1"、PDR="1"を設定します。このとき、TXD 端子はI/Oポートとなっており"1"が出力されます。

一方、データ送信時にブレークを送出したいときは、PCR="1"、PDR="0"に設定した後PMR6のTXD ビットを"0"にクリアします。

PMR6のTXD ビットを"0"にクリアすると現在の送信状態とは無関係にTXD 端子はI/Oポートになり、TXD 端子から"0"が出力されます。

### (5) 受信エラーフラグと送信動作について (クロック同期式モードのみ)

受信エラーフラグ(OER、PER、FER)が"1"にセットされた状態では、TDREを"0"にクリアしても送信を開始できません。必ず送信開始時には、受信エラーフラグを"0"にクリアしておいてください。

また、REを"0"にクリアしても受信エラーフラグは"0"にクリアできませんので注意してください。

## (6) 調歩同期式モードの受信データサンプリングタイミングと受信マージン

調歩同期式モードでは、SCI3 は転送レートの 16 倍の周波数の基本クロックで動作しています。

受信時には SCI3 は、スタートビットの立下がりを基本クロックでサンプリングして内部を同期化します。また、受信データを基本クロックの 8 ヶ目の立上がりエッジで内部に取り込みます。

これを図 10.23 に示します。

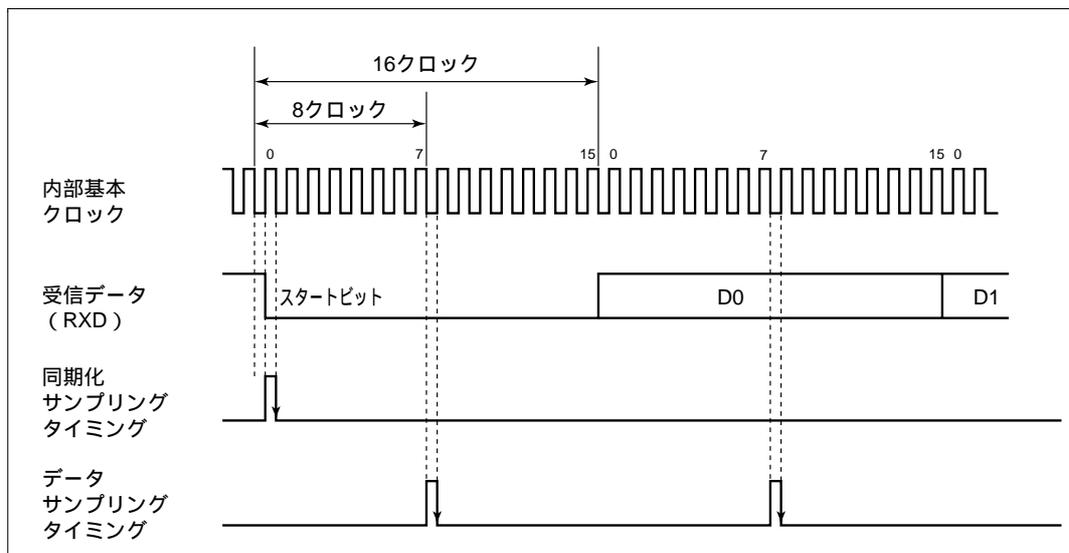


図 10.23 調歩同期式モードの受信データサンプリングタイミング

したがって、調歩同期式モードでの受信マージンは式 (1) のように表すことができます。

$$M = \left\{ \left( 0.5 - \frac{1}{2N} \right) - \frac{D - 0.5}{N} - (L - 0.5) F \right\} \times 100 \quad [\%] \quad \dots \text{式 (1)}$$

M : 受信マージン (%)

N : クロックに対するビットレートの比 (N=16)

D : クロックのデューティ (D=0.5~1.0)

L : フレーム長 (L=9~12)

F : クロック周波数の偏差の絶対値

式 (1) で、F (クロック周波数の偏差の絶対値) = 0、D (クロックのデューティ) = 0.5 とすると、受信マージンは式 (2) より 46.875% となります。

D=0.5、F=0 のとき、

$$\begin{aligned} M &= \left\{ 0.5 - 1/(2 \times 16) \right\} \times 100 \quad [\%] \\ &= 46.875\% \quad \dots \text{式 (2)} \end{aligned}$$

ただし、この値はあくまでも計算上の値ですので、システム設計の際には20～30%の余裕を持たせてください。

### (7) RDR のリードと RDRF の関係について

SCI3 は受信動作において、RDRF フラグをチェックしながら動作します。1 フレームの受信終了のタイミングで RDRF が"0"にクリアされていれば、通常 of データ受信を完了します。また RDRF が"1"にセットされていれば、オーバランエラーとなります。

RDR の内容をリードすると、RDRF は自動的に"0"にクリアされます。したがって、RDR のリードを2回以上行う場合、2回目以降のリード操作は RDRF が"0"の状態で行われます。RDRF が"0"の状態 RDR のリードを行うと、リード操作が次のフレームの受信が完了するタイミングと重なった場合、次のフレームのデータが読み出されることがありますので注意してください。これを図 10.24 に示します。

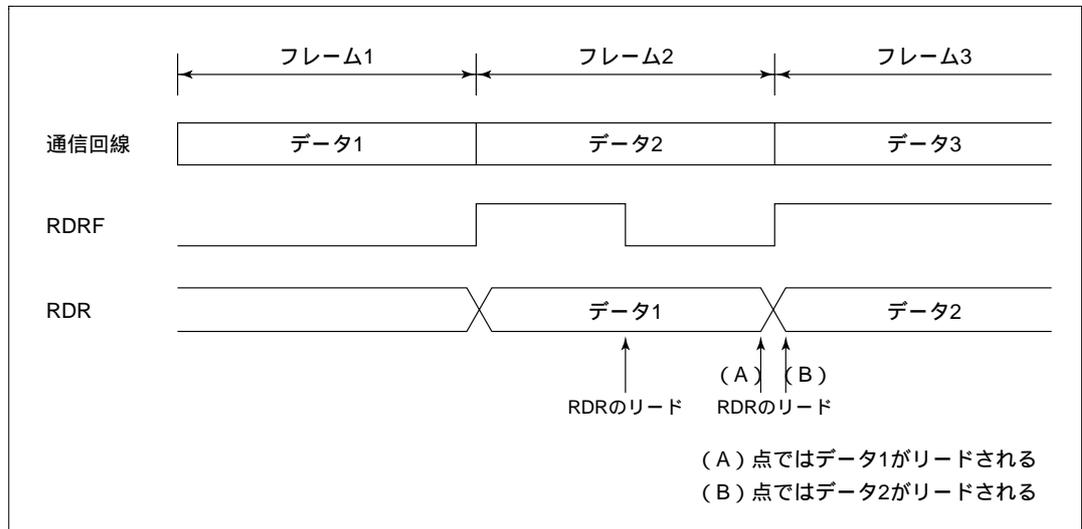


図 10.24 RDR のリードタイミングとデータの関係

この場合、RDR のリード操作は RDRF が"1"にセットされていることを確認してから、1回のみ行う(2回以上リードしない)ようにしてください。2回以上リードする場合は、1回リードしたデータを RAM などに転送し、その内容を使用するようにしてください。また、RDR のリード操作は、次のフレームの受信が完了するまでに余裕をもって行うようにしてください。具体的なタイミングとしては、クロック同期式モードではビット7の転送前まで、調歩同期式モードでは STOP ビットの転送前までに RDR のリードを完了してください。

---

# 11. DTMF 発生回路

---

## 第 11 章 目次

11.1	概要	277
11.1.1	特長	277
11.1.2	ブロック図	278
11.1.3	端子構成	278
11.1.4	レジスタ構成	279
11.2	各レジスタの説明	280
11.2.1	DTMF コントロールレジスタ (DTCR)	280
11.2.2	DTMF ロードレジスタ (DTLR)	282
11.3	動作説明	283
11.3.1	出力波形	283
11.3.2	動作フロー	284
11.4	応用回路例	285
11.5	使用上の注意	286



## 11.1 概要

本 LSI は、DTMF (Dual Tone Multi Frequency) 発生回路を内蔵しており、DTMF 信号を発生させることができます。

DTMF 信号は、交換機をアクセスする 2 種類の正弦波で構成され、図 11.1 に示す周波数のマトリックスで表せます。DTMF 発生回路は、各数字、記号に対応する周波数を発生させます。

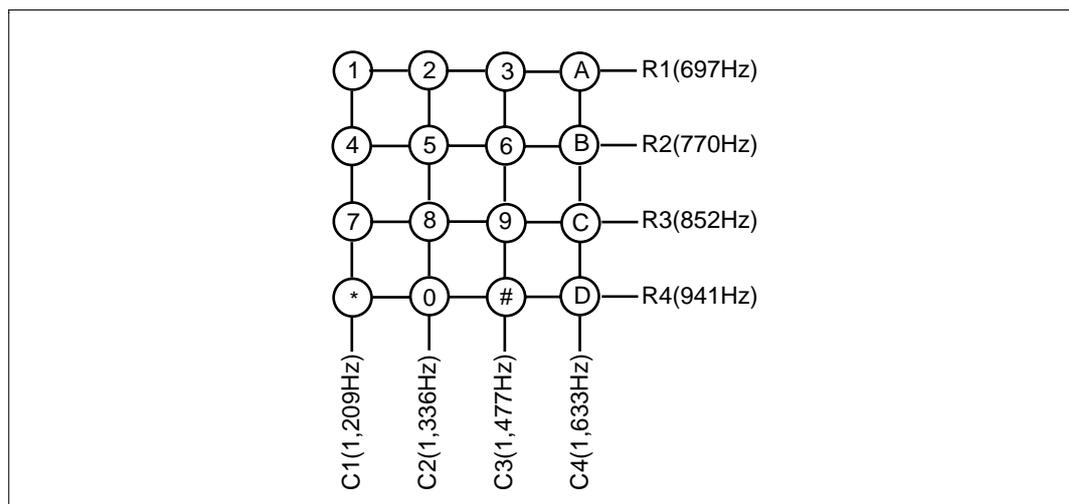


図 11.1 DTMF 周波数

### 11.1.1 特長

DTMF 発生回路の特長を以下に示します。

OSC クロック ( $f_{osc}$ ) から DTMF 周波数の正弦波を生成

OSC クロック (1.2MHz ~ 10MHz、400kHz 刻み) を分周して、400kHz のクロックを生成し、これを変形プログラムデバイダ、正弦波カウンタのフィードバックループに入力することで DTMF 周波数の正弦波を生成します。

低歪率の安定した正弦波形を出力可能

正弦波は、高精度抵抗ラダー型 D/A 変換回路で出力され、また 1 サイクルを 32 分割しているため、低歪率の安定な波形を得られます。

合成または単独の波形出力選択可能

レジスタ設定により Row グループと Column グループの合成出力または、Row グループ、Column グループ単独の出力を選択できます。

### 11.1.2 ブロック図

DTMF 発生回路のブロック図を、図 11.2 に示します。

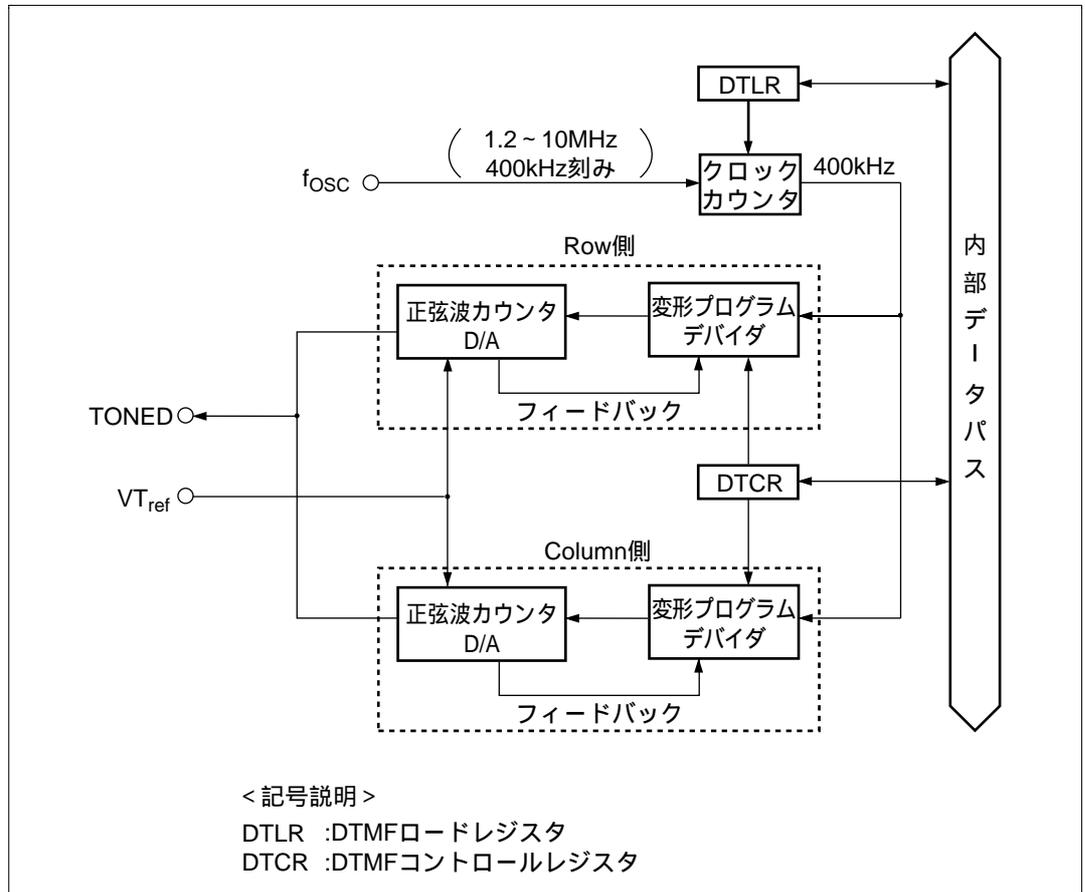


図 11.2 DTMF 発生回路のブロック図

### 11.1.3 端子構成

DTMF 発生回路の端子構成を表 11.1 に示します。

表 11.1 端子構成

名称	略称	入出力	機能
DTMF 出力基準レベル 電源	$VT_{ref}$		DTMF 出力の基準レベル電源端子
DTMF 信号出力	TONED	出力	DTMF 信号出力端子

### 11.1.4 レジスタ構成

DTMF 発生回路のレジスタ構成を表 11.2 に示します。

表 11.2 レジスタ構成

名称	略称	R/W	初期値	アドレス
DTMF コントロールレジスタ	DTCR	R/W	H'40	H'FFB2
DTMF ロードレジスタ	DTLR	R/W	H'E0	H'FFB3

## 11.2 各レジスタの説明

### 11.2.1 DTMF コントロールレジスタ (DTCR)

ビット:	7	6	5	4	3	2	1	0
	DTEN	—	CLOE	RWOE	CLF1	CLF0	RWF1	RWF0
初期値:	0	1	0	0	0	0	0	0
R/W :	R/W	—	R/W	R/W	R/W	R/W	R/W	R/W

DTCR は、8 ビットのリード/ライト可能なレジスタで、DTMF 発生回路動作、Column 側出力、Row 側出力の制御、出力周波数の選択を行います。

リセット時、DTCR は H'40 にイニシャライズされます。

#### ビット 7 : DTMF 発生回路動作制御 (DTEN)

DTMF 回路を動作または停止させます。

ビット 7	説明	
DTEN		
0	DTMF 発生回路を停止	(初期値)
1	DTMF 発生回路を動作	

#### ビット 6 : リザーブビット

リザーブビットです。リードすると常に "1" が読み出されます。ライトは無効です。

#### ビット 5 : Column 側出力制御 (CLOE)

Column 側 DTMF 信号出力を許可または禁止します。

ビット 5	説明	
CLOE		
0	Column 側 DTMF 信号出力を禁止 (ハイインピーダンス)	(初期値)
1	Column 側 DTMF 信号を許可	

#### ビット 4 : Row 側出力制御 (RWOE)

Row 側 DTMF 信号出力を許可または禁止します。

ビット 4	説明	
RWOE		
0	Row 側 DTMF 信号出力を禁止 (ハイインピーダンス)	(初期値)
1	Row 側 DTMF 信号を許可	

ビット3、2 : Column 側 DTMF 信号出力周波数 1、0 (CLF1、CLF0)

Column 側 DTMF 信号 (C1 ~ C4) の周波数を選択します。

ビット3	ビット2	説明
CLF1	CLF0	
0	0	Column 側 DTMF 信号出力周波数 : 1209Hz (C1) (初期値)
0	1	Column 側 DTMF 信号出力周波数 : 1336Hz (C2)
1	0	Column 側 DTMF 信号出力周波数 : 1447Hz (C3)
1	1	Column 側 DTMF 信号出力周波数 : 1633Hz (C4)

ビット1、0 : Row 側 DTMF 信号出力周波数 1、0 (RWF1、RWF0)

Row 側 DTMF 信号 (R1 ~ R4) の周波数を選択します。

ビット1	ビット0	説明
RWF1	RWF0	
0	0	Row 側 DTMF 信号出力周波数 : 697Hz (R1) (初期値)
0	1	Row 側 DTMF 信号出力周波数 : 770Hz (R2)
1	0	Row 側 DTMF 信号出力周波数 : 852Hz (R3)
1	1	Row 側 DTMF 信号出力周波数 : 941Hz (R4)

### 11.2.2 DTMF ロードレジスタ (DTLR)

ビット:	7	6	5	4	3	2	1	0
	—	—	—	DTL4	DTL3	DTL2	DTL1	DTL0
初期値:	1	1	1	0	0	0	0	0
R/W :	—	—	—	R/W	R/W	R/W	R/W	R/W

DTLR は、8 ビットのリード/ライト可能なレジスタで、DTMF 発生回路用に OSC クロックの分周比を設定します。

リセット時、DTLR は H'E0 にイニシャライズされます。

#### ビット 7~5 : リザーブビット

リザーブビットです。各ビットはリードすると常に "1" が読み出されます。ライトは無効です。

#### ビット 4~0 : OSC クロック分周比 4~0 (DTL4~DTL0)

DTMF 発生回路に inputs する 400kHz クロックを生成するために、OSC クロックの分周比を設定します。分周比は OSC クロック 1.2~10MHz (400kHz 刻み) に対応して、3~25 のカウント値を設定します。

ビット 4	ビット 3	ビット 2	ビット 1	ビット 0	説明	
DTL4	DTL3	DTL2	DTL1	DTL0	分周比	OSC クロック周波数
0	0	0	0	0	設定禁止	(初期値)
0	0	0	0	1	設定禁止	
0	0	0	1	0	設定禁止	
0	0	0	1	1	3	1.2 MHz
0	0	1	0	0	4	1.6 MHz
⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	1	0	0	1	25	10 MHz
1	1	0	1	*	設定禁止	
1	1	1	*	*	設定禁止	

\* Don't care

本ビットの設定が、OSC クロックと合っていない場合、正常な DTMF 信号の出力周波数が得られません。また、本ビットの設定値が、3~25 以外の場合、動作が保証されませんので、本ビットには正しい値を設定してください。

## 11.3 動作説明

### 11.3.1 出力波形

DTMF 発生回路は、TONED 端子より Row グループ、Column グループの合成波または Row グループ、Column グループ単独の正弦波 (DTMF 信号) を出力します。これらの信号は、高精度抵抗ラダー型 D/A 変換回路で生成されます。出力周波数は、DTCR により設定します。

TONED 端子出力の等価回路を図 11.3 に、また Row グループ、Column グループ単独時の出力波形を図 11.4 に示します。出力波形は 1 サイクルを 32 分割しているため、低歪率で安定な出力を得られます。

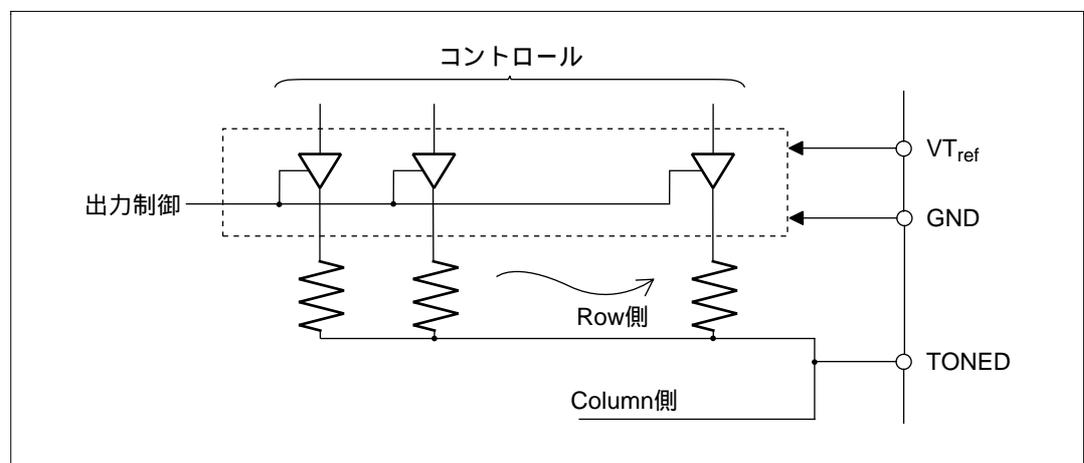


図 11.3 TONED 端子出力等価回路

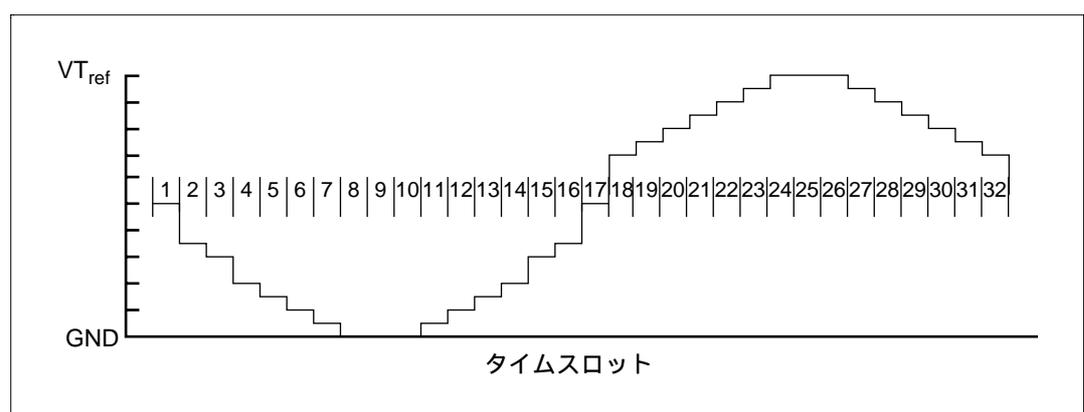


図 11.4 TONED 端子出力波形 (Row グループ、Column グループ単独時)

DTMF 発生回路の出力信号と標準信号との周波数偏差を表 11.3 に示します。

表 11.3 DTMF 出力信号と標準信号との周波数偏差

記号	標準信号 (Hz)	DTMF 信号出力 (Hz)	周波数偏差 (%)
R1	697	694.44	- 0.37
R2	770	769.23	- 0.10
R3	852	851.06	- 0.11
R4	941	938.97	- 0.22
C1	1209	1212.12	0.26
C2	1336	1333.33	- 0.20
C3	1477	1481.48	0.30
C4	1633	1639.34	0.39

### 11.3.2 動作フロー

DTMF 発生回路の操作手順を以下に示します。

- [ 1 ] 接続されている OSC クロック発振子の周波数に合わせて、OSC クロック分周比を DTLR に設定します (1.2MHz ~ 10MHz、400kHz 刻み)。
- [ 2 ] DTCR の CLF1、CLF0、RWF1、RWF0 により、Row 側 (R1 ~ R4)、Column 側 (C1 ~ C4) の周波数を設定します。
- [ 3 ] DTCR の CLOE、RWOE により、Row 側、Column 側の出力を選択し、DTEN を "1" にセットして、DTMF 発生回路を動作させます。

上記の手順で、設定された DTMF 信号が TONED 端子より出力されます。

## 11.4 応用回路例

DTMF 発生回路の応用例を図 11.5 に示します。

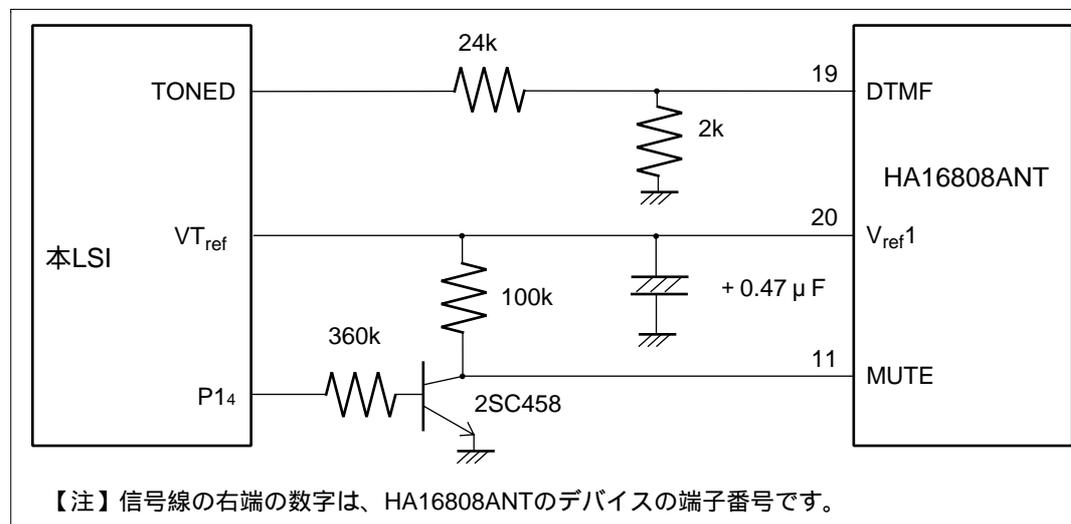


図 11.5 HA16808 ANT との接続例

## 11.5 使用上の注意

DTMF 発生回路を使用するときには、次の点に注意してください。

DTLR の設定値と、OSC クロックを必ず合わせてください。DTLR の設定値と OSC クロックが異なる場合には、正常な DTMF 信号の出力周波数が得られません。

---

# 12. A/D 変換器

---

## 第 12 章 目次

12.1	概要	289
12.1.1	特長	289
12.1.2	ブロック図	290
12.1.3	端子構成	291
12.1.4	レジスタ構成	291
12.2	各レジスタの説明	292
12.2.1	A/D リザルトレジスタ (ADRR)	292
12.2.2	A/D モードレジスタ (AMR)	292
12.2.3	A/D スタートレジスタ (ADSR)	294
12.3	動作説明	295
12.3.1	A/D 変換動作	295
12.3.2	外部トリガによる A/D 変換器の起動	295
12.4	割込み要因	296
12.5	使用例	297
12.6	使用上の注意	301



## 12.1 概要

本LSIは、抵抗ラダー方式による逐次比較型A/D変換器を内蔵しており、最大2チャンネルのアナログ入力の測定ができます。

### 12.1.1 特長

A/D変換器の特長を以下に示します。

8ビットの分解能

入力チャンネル：2チャンネル

変換時間：1チャンネル当たり 12.4 $\mu$ s (5MHz動作時)

サンプル&ホールド機能

A/D変換終了割込み要求を発生

外部トリガ入力により、A/D変換開始を指定可能

### 12.1.2 ブロック図

A/D 変換器のブロック図を図 12.1 に示します。

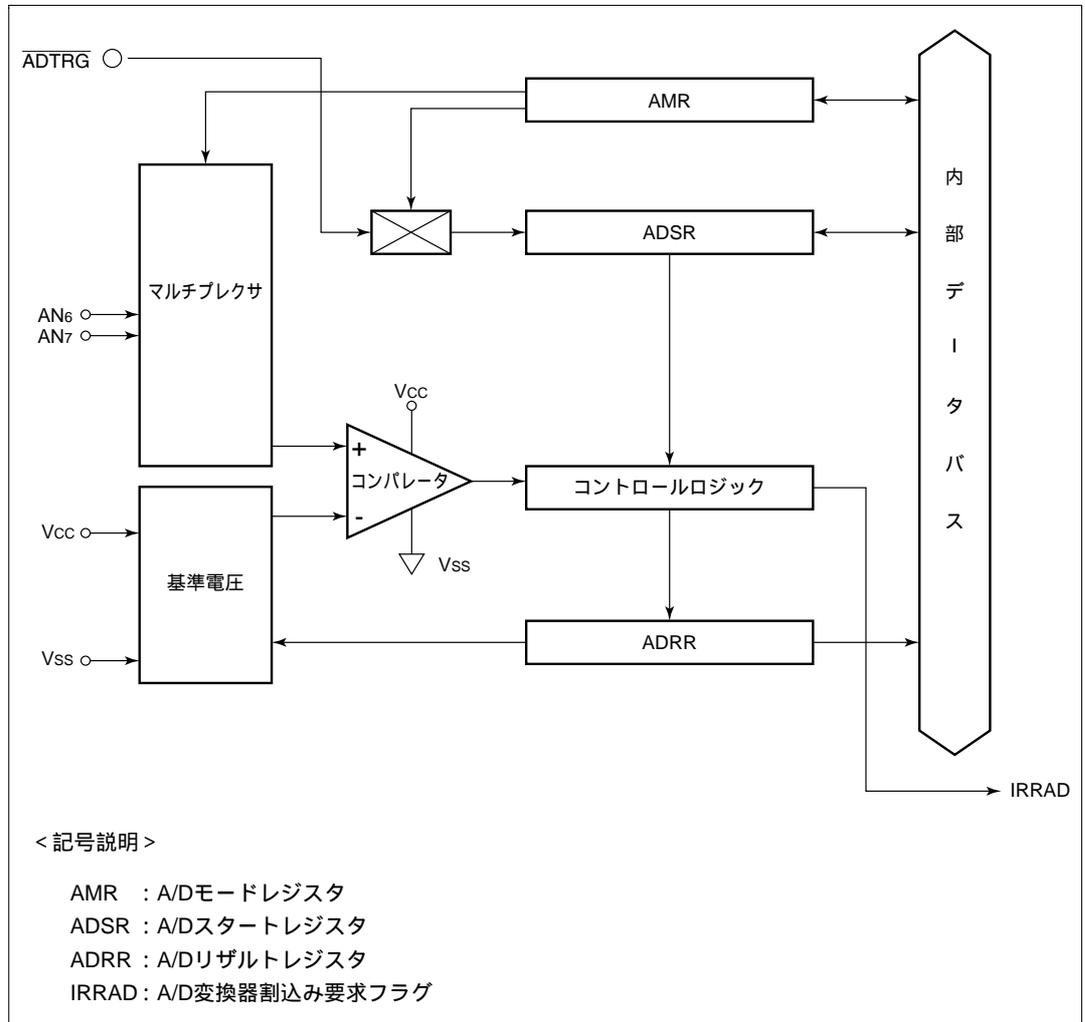


図 12.1 A/D 変換器ブロック図

### 12.1.3 端子構成

A/D変換器の端子構成を表 12.1 に示します。

表 12.1 端子構成

端子名	略称	入出力	機能
電源端子	V <sub>CC</sub>	入力	電源
グランド端子	V <sub>SS</sub>	入力	グランドおよび基準電圧
アナログ入力端子 6	AN <sub>6</sub>	入力	アナログ入力チャンネル 6
アナログ入力端子 7	AN <sub>7</sub>	入力	アナログ入力チャンネル 7
外部トリガ入力端子	$\overline{\text{ADTRG}}$	入力	A/D変換の開始を制御する外部トリガ入力

### 12.1.4 レジスタ構成

A/D変換器のレジスタ構成を表 12.2 に示します。

表 12.2 レジスタ構成

名称	略称	R/W	初期値	アドレス
A/Dモードレジスタ	AMR	R/W	H'10	H'FFC4
A/Dスタートレジスタ	ADSR	R/W	H'7F	H'FFC6
A/Dリザルトレジスタ	ADRR	R	不定	H'FFC5

## 12.2 各レジスタの説明

### 12.2.1 A/D リザルトレジスタ (ADRR)

ビット:	7	6	5	4	3	2	1	0
	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
初期値:	不定							
R/W :	R	R	R	R	R	R	R	R

ADRR は、A/D 変換された結果を格納する 8 ビットのリード専用レジスタです。

ADRR は常に CPU からリード可能です。A/D 変換中は ADRR の値は不定で、A/D 変換終了時に変換結果の 8 ビットデータが格納され、次の変換開始までこのデータが保持されます。

ADRR は、リセットでクリアされません。

### 12.2.2 A/D モードレジスタ (AMR)

ビット:	7	6	5	4	3	2	1	0
	CKS	TRGE	CKS1	—	CH3	CH2	CH1	CH0
初期値:	0	0	0	1	0	0	0	0
R/W :	R/W	R/W	R/W	—	R/W	R/W	R/W	R/W

AMR は、8 ビットのリード/ライト可能なレジスタで、A/D 変換スピードの設定、外部トリガの選択、アナログ入力端子の指定を行います。

リセット時、AMR は H'10 にイニシャライズされます。

#### ビット7: クロックセレクト (CKS)

CKS1 との組み合わせで A/D 変換スピードの設定を行います。

ビット5	ビット7	変換周期	変換時間	
			= 2MHz	= 5MHz
CKS1	CKS			
0	0	リザーブ (初期値)		
0	1	124/	62 $\mu$ s	24.8 $\mu$ s
1	0	62/	31 $\mu$ s	12.4 $\mu$ s
1	1	31/	15.5 $\mu$ s	*

【注】 \* 12.4 $\mu$ s 以下の変換時間では、動作が保証されません。12.4 $\mu$ s 以上になるように選択してください。

## ビット6：外部トリガセレクト (TRGE)

外部トリガ入力による A/D 変換の開始を許可または禁止します。

ビット6		
TRGE	説明	
0	外部トリガによる A/D 変換の開始を禁止	(初期値)
1	外部トリガ ( $\overline{\text{ADTRG}}$ ) 端子の立上がりエッジ、または立下がりエッジで A/D 変換を開始*	

【注】 \* 外部トリガ  $\overline{\text{ADTRG}}$  端子のエッジ選択は IEGR の IEG4 により設定します。詳細は「3.3.2 (1) IRQ エッジセレクトレジスタ (IEGR)」を参照してください。

## ビット5：クロックセレクト1 (CKS1)

CKS との組合わせで A/D 変換スピードの設定を行います。詳細は「ビット7：クロックセレクト (CKS)」の説明を参照してください。

## ビット4：リザーブビット

リザーブビットです。本ビットはリードすると常に"1"が読み出されます。ライトは無効です。

## ビット3～0：チャンネルセレクト3～0 (CH3～CH0)

アナログ入力チャンネルの選択を行います。

チャンネル選択の切換えは、ADSF = "0"の状態で行ってください。

ビット3	ビット2	ビット1	ビット0	
CH3	CH2	CH1	CH0	アナログ入力チャンネル
0	0	*	*	非選択 (初期値)
0	1	*	*	リザーブ
1	0	0	*	リザーブ
1	0	1	0	AN <sub>6</sub>
1	0	1	1	AN <sub>7</sub>
1	1	*	*	リザーブ

\* : Don't care

## 12.2.3 A/D スタートレジスタ (ADSR)

ビット:	7	6	5	4	3	2	1	0
	ADSF	—	—	—	—	—	—	—
初期値:	0	1	1	1	1	1	1	1
R/W :	R/W	—	—	—	—	—	—	—

ADSR は、8 ビットのリード/ライト可能なレジスタで、A/D 変換の開始または停止を指定します。

ADSF に"1"をライトまたは外部トリガのエッジ入力により、ADSF が"1"にセットされ A/D 変換が開始します。変換が終了すると変換データはADRR にセットされ、同時にADSF は"0"にクリアされます。

## ビット7: A/D スタートフラグ (ADSF)

A/D 変換の開始および終了の確認を行います。

ビット7		説明
ADSF		
0	リード時	A/D 変換の終了 (初期値)
	ライト時	A/D 変換を強制終了
1	リード時	A/D 変換中
	ライト時	A/D 変換を開始

## ビット6~0: リザーブビット

リザーブビットです。各ビットはリードすると常に"1"が読み出されます。ライトは無効です。

## 12.3 動作説明

### 12.3.1 A/D変換動作

A/D変換器は逐次比較方式で動作し、8ビットの変換結果が得られます。

ソフトウェアにより ADSF を "1" にセットすると、A/D変換を開始します。ADSF は、A/D変換中は "1" を保持しており、変換が終了すると自動的に "0" にクリアされます。

また、変換が終了すると、IRR2 の IRRAD が "1" にセットされます。このとき、IENR2 の IENAD が "1" にセットされていると、A/D変換終了割込みが発生します。

A/D変換中に、AMR により変換時間や入力チャネルの切り換えを行う場合は、誤動作を避けるために ADSF を "0" にクリアして、A/D変換を強制終了させてから行ってください。

### 12.3.2 外部トリガによる A/D変換器の起動

A/D変換器は外部トリガ入力によって A/D変換を開始させることができます。

外部トリガは I/Oポートの PMR2 の IRQ4 が "1" でかつ AMR の TRGE が "1" のとき、 $\overline{\text{ADTRG}}$  入力端子から入力されます。 $\overline{\text{ADTRG}}$  入力端子から IEGR の IEG4 で指定されたエッジが入力されると、ADSR の ADSF が "1" にセットされ、A/D変換が開始されます。

このタイミングを図 12.2 に示します。

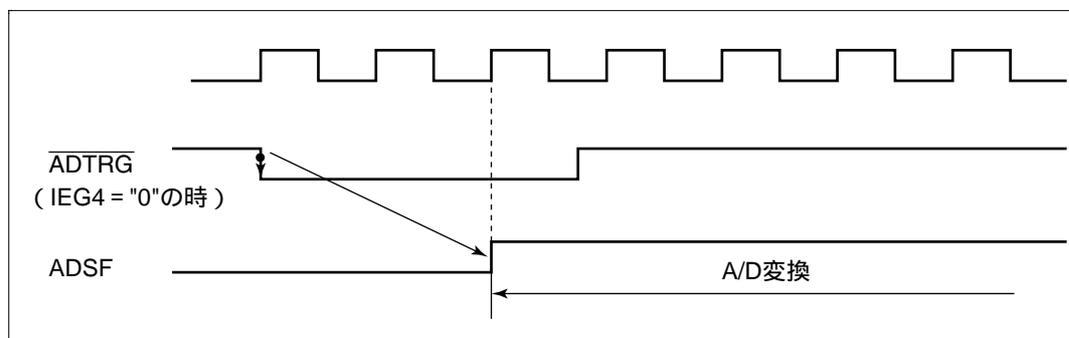


図 12.2 外部トリガ入力タイミング

## 12.4 割込み要因

A/D変換終了時 (ADSF = "1" "0")、IRR2のIRRADが"1"にセットされます。

A/D変換終了割込みは、IENR2のIENADにより、許可/禁止を指定できます。

詳細は「3.3 割込み」を参照してください。

## 12.5 使用例

チャンネル 6 ( $AN_6$ ) をアナログ入力チャンネルに選択した場合の動作例を示します。動作タイミングを図 12.3 に示します。

- (1) 入力チャンネルを  $AN_6$  (AMR の CH3 ~ CH0 を "1010")、IENAD = "1" に設定して、A/D 変換を開始 (ADSF = "1") します。
- (2) A/D 変換が終了すると、IRRAD が "1" にセットされ、A/D 変換結果が ADDR に格納されます。同時に ADSF = "0" となり、A/D 変換器は変換待機となります。
- (3) IENAD = "1" となっているため A/D 変換終了割り込み要求が発生します。
- (4) A/D 割り込み処理ルーチンが開始されます。
- (5) A/D 変換結果を読み出して、処理します。
- (6) A/D 変換処理ルーチンの実行が終了します。

この後、ADSF = "1" にセットすると A/D 変換が開始され (2) ~ (6) を行います。A/D 変換器の使用手順の概念フローを図 12.4、図 12.5 に示します。

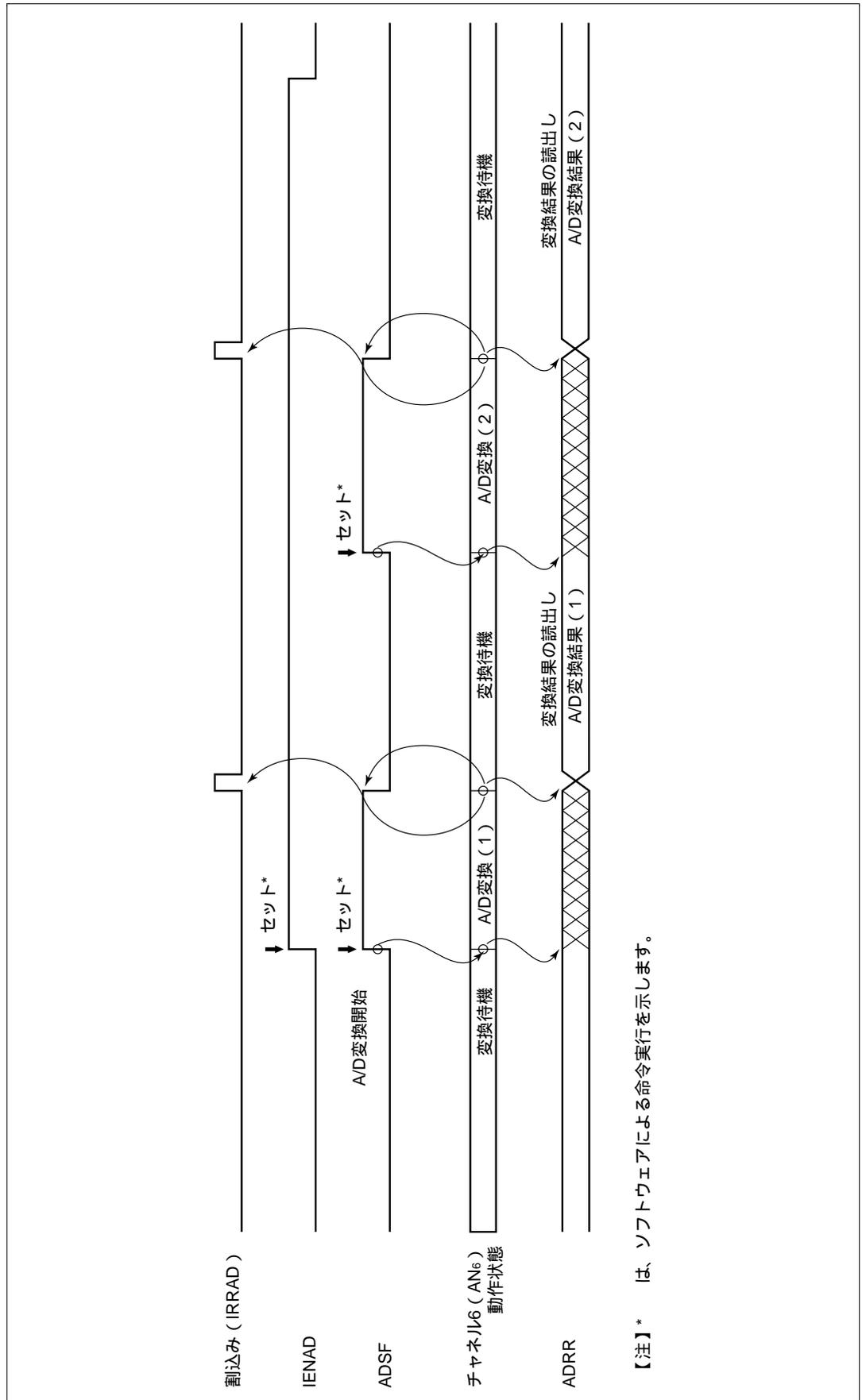


図 12.3 A/D 変換器の動作例

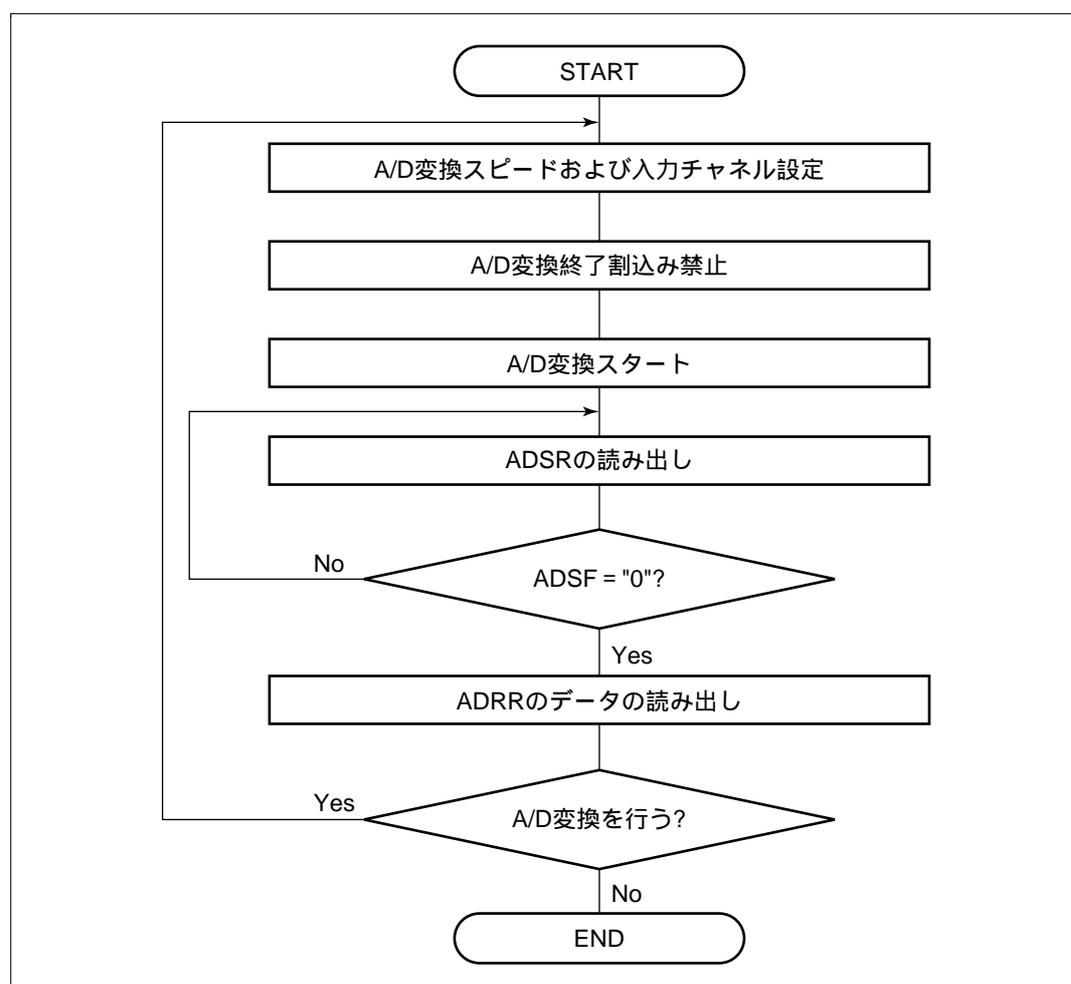


図 12.4 A/D 変換器の使用手順の概念フロー (1)  
(ソフトウェアでポーリングする場合)

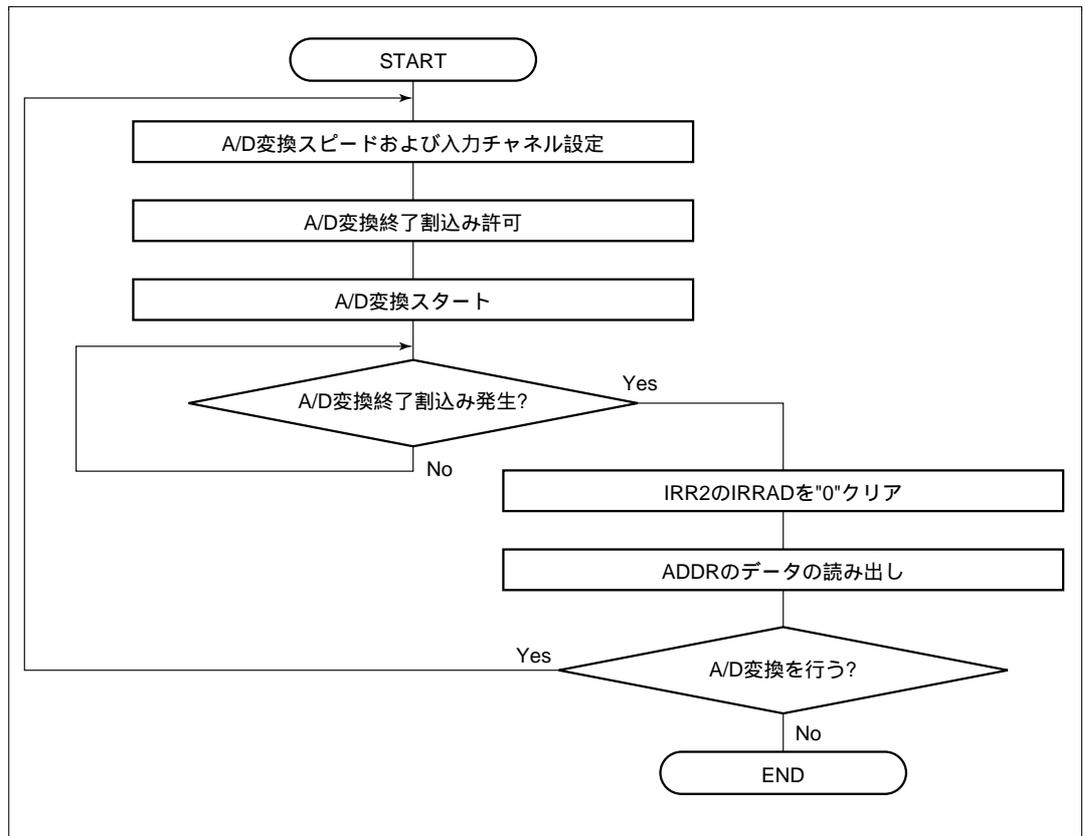


図 12.5 A/D 変換器の使用手順の概念フロー (2)  
(割り込みを使用する場合)

## 12.6 使用上の注意

- (1) ADRRの読出しは、ADSRのADSFが"0"のときに行ってください。
- (2) A/D変換中に隣接した端子のデジタル入力信号を変化させると変換精度に悪影響を及ぼします。

---

# 13. 電源回路

---

## 第 13 章 目次

13.1	概要.....	305
13.2	内部電源降圧回路構成.....	305



## 13.1 概要

本 LSI には内部電源降圧回路が内蔵されています。この内部電源降圧回路を使用することにより、外部  $V_{CC}$  端子に接続された電源電圧に依存することなく、内部電源を約 3.0V 一定に固定することができます。このため外部電源を 3.0V 以上で使用した場合に消費される電流値増加を抑えることができます。

## 13.2 内部電源降圧回路構成

図 13.1 に示すように、 $V_{CC}$  端子に外部電源を接続し、 $CV_{CC}$  と  $V_{SS}$  間に内部電源電圧安定化のため  $0.1 \mu F$  の容量を接続してください。容量を付加せず使用した場合誤動作の原因となります。

【注】 外部回路のインタフェースは  $V_{CC}$  に接続されている外部電源電圧と  $V_{SS}$  に接続されている GND 電位が基準となります。

例えば、ポートの入出力レベルは "H" が  $V_{CC}$  レベル基準、"L" が  $V_{SS}$  レベル基準となります。

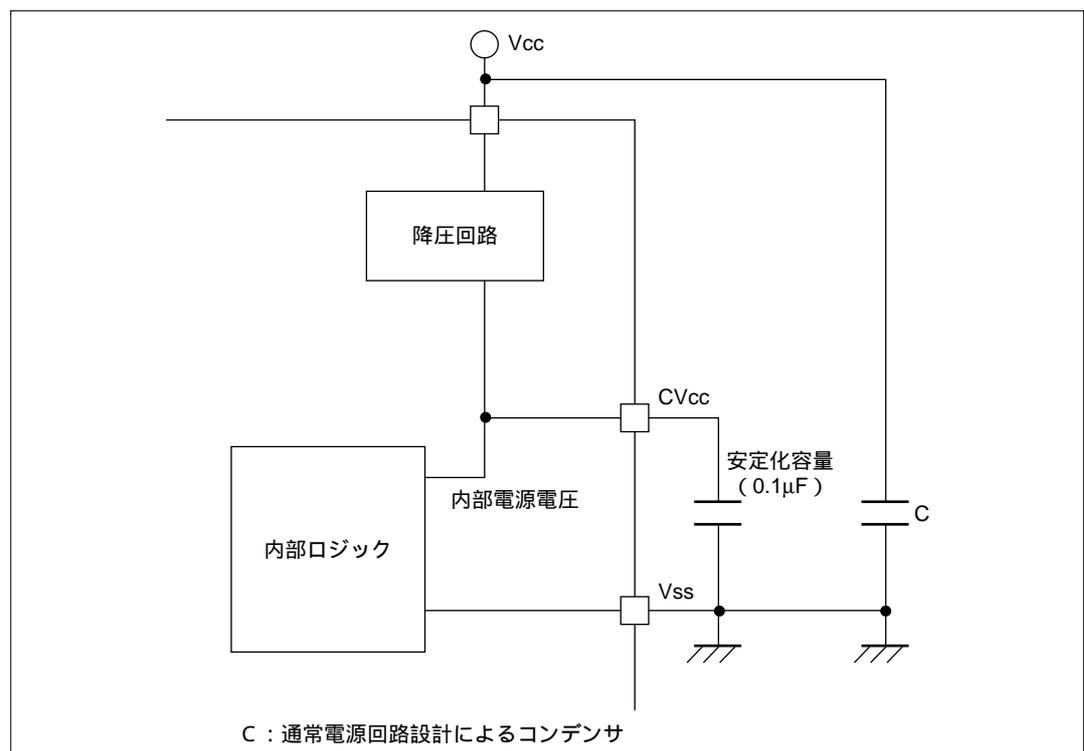


図 13.1 内部電源降圧回路の電源接続図

---

# 14. 電気的特性

---

## 第 14 章 目次

14.1	絶対最大定格.....	309
14.2	電気的特性.....	310
	14.2.1 電源電圧と動作範囲.....	310
	14.2.2 DC 特性.....	312
	14.2.3 AC 特性.....	317
	14.2.4 A/D 変換器特性.....	319
	14.2.5 DTMF 特性.....	320
14.3	動作タイミング.....	321
14.4	出力負荷回路.....	324



## 14.1 絶対最大定格

絶対最大定格を表 14.1 に示します。

表 14.1 絶対最大定格

項目	記号	規格値	単位	備考	
電源電圧	$V_{CC}$	- 0.3 ~ + 7.0	V	*	
基準レベル電源電圧	$V_{T_{ref}}$	- 0.3 ~ $V_{CC} + 0.3$	V		
プログラム電圧	$V_{PP}$	- 0.3 ~ + 13.0	V		
入力電圧	ポート B 以外	$V_{in}$	- 0.3 ~ $V_{CC} + 0.3$		V
	ポート B	$AV_{in}$	- 0.3 ~ $V_{CC} + 0.3$		V
動作温度	$T_{opr}$	- 20 ~ + 75			
保存温度	$T_{stg}$	- 55 ~ + 125			

【注】\* 絶対最大定格をこえて LSI を使用した場合、LSI の永久破壊となることがあります。また、通常動作では、「電気的特性」の条件で使用することが望ましく、この条件をこえると LSI の誤動作の原因になるとともに、LSI の信頼性に悪影響をおよぼすことがあります。

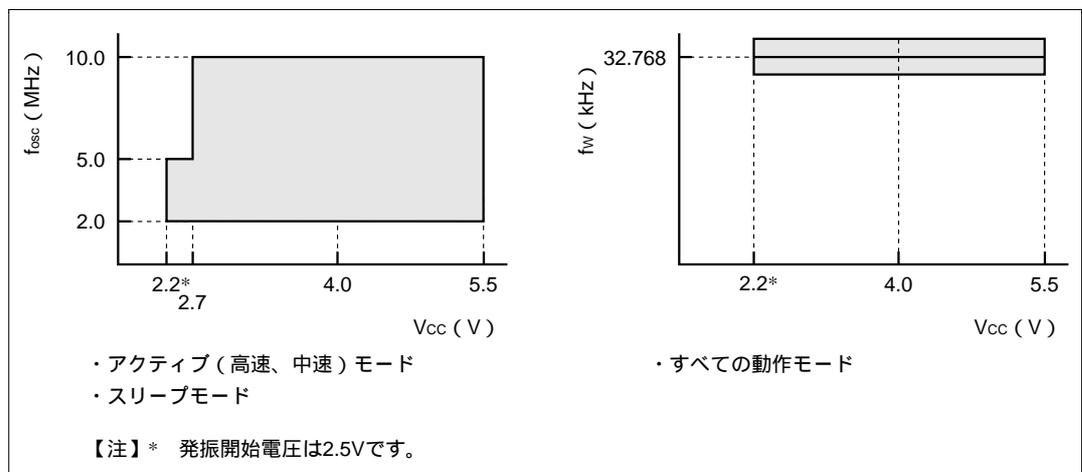
## 14.2 電気的特性

### 14.2.1 電源電圧と動作範囲

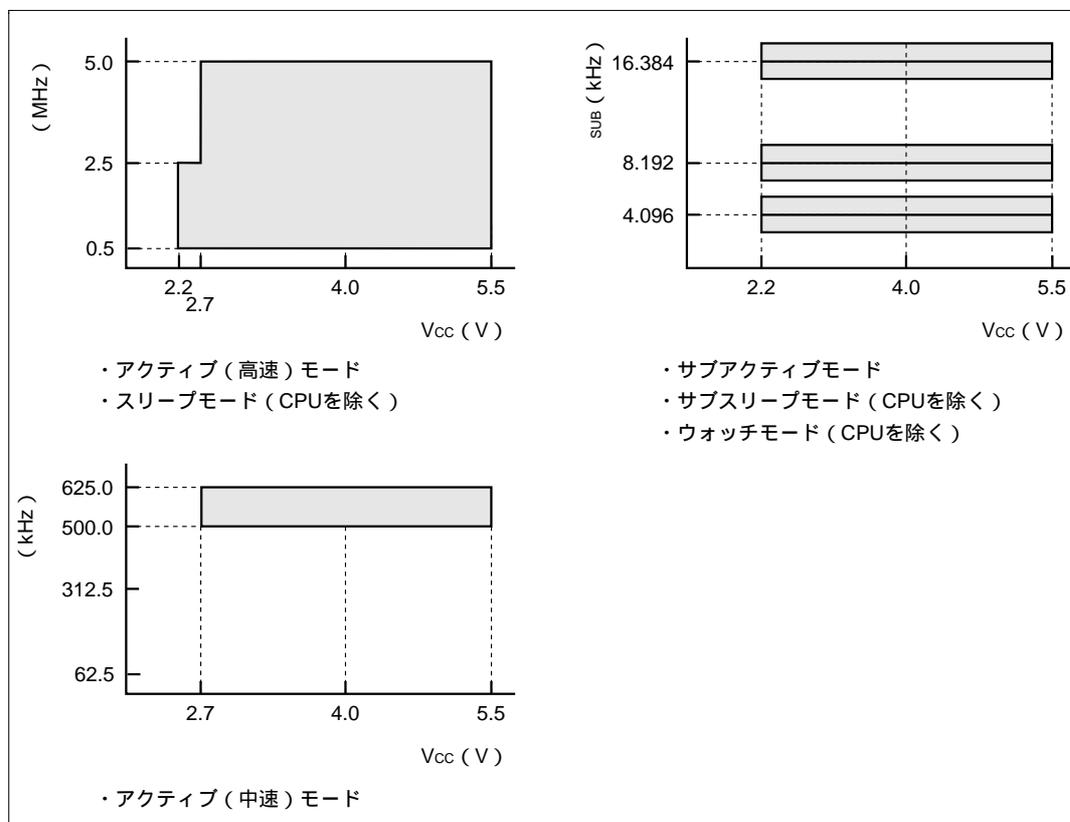
電源電圧と動作範囲（網かけ部）を以下に示します。

【注】本製品と開発ツールとでは動作保証範囲が異なりますので、開発の際にはご注意ください。

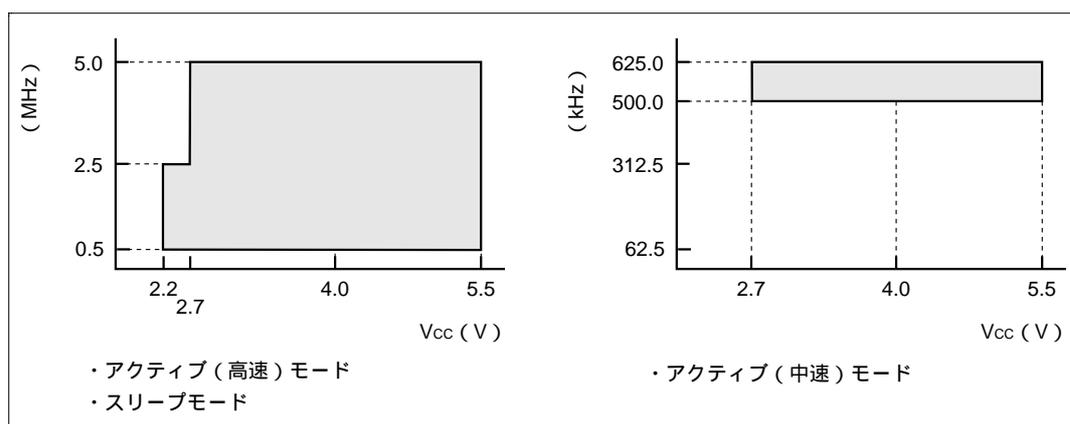
#### (1) 電源電圧と発振周波数の範囲



## (2) 電源電圧と動作周波数の範囲



## (3) アナログ電源電圧と A/D 変換器の動作範囲



### 14.2.2 DC 特性

DC 特性を表 14.2 に示します。

表 14.2 DC 特性 (1)

(特記なき場合、 $V_{CC} = 2.2 \sim 5.5V$ 、 $V_{SS} = 0.0V$ 、 $T_a = -20 \sim +75$ 、サブアクティブモードを含む)

項目	記号	適用端子	測定条件	規格値			単位	備考
				min.	typ.	max.		
入力"High" レベル電圧	$V_{IH}$	$\overline{RES}_0$ 、 $\overline{WKP}_0 \sim \overline{WKP}_7$ 、 $\overline{IRQ}_0 \sim \overline{IRQ}_4$ 、 TMIF、TMIG、 SCK <sub>1</sub> 、SCK <sub>3</sub> 、ADTRG	$V_{CC} = 2.7 \sim 5.5V$	$0.8V_{CC}$		$V_{CC} + 0.3$	V	
			$0.9V_{CC}$		$V_{CC} + 0.3$			
		SI <sub>1</sub> 、RXD	$V_{CC} = 2.7 \sim 5.5V$	$0.7V_{CC}$		$V_{CC} + 0.3$	V	
				$0.8V_{CC}$		$V_{CC} + 0.3$		
		OSC <sub>1</sub>	$V_{CC} = 2.7 \sim 5.5V$	$V_{CC} - 0.5$		$V_{CC} + 0.3$	V	
				$V_{CC} - 0.3$		$V_{CC} + 0.3$		
P1 <sub>0</sub> ~ P1 <sub>7</sub> 、 P2 <sub>0</sub> ~ P2 <sub>7</sub> 、 P5 <sub>0</sub> ~ P5 <sub>7</sub> 、 P6 <sub>0</sub> ~ P6 <sub>7</sub> 、 P7 <sub>0</sub> ~ P7 <sub>7</sub> 、 P8 <sub>0</sub> ~ P8 <sub>7</sub> 、 PA <sub>1</sub> ~ PA <sub>3</sub> 、 PB <sub>6</sub> 、PB <sub>7</sub>	$V_{CC} = 2.7 \sim 5.5V$	$0.7V_{CC}$		$V_{CC} + 0.3$	V			
		$0.8V_{CC}$		$V_{CC} + 0.3$				

【注】 TEST 端子は、 $V_{SS}$  に接続してください。

表 14.2 DC特性(2)

(特記なき場合、 $V_{CC} = 2.2 \sim 5.5V$ 、 $V_{SS} = 0.0V$ 、 $T_a = -20 \sim +75$ 、サブアクティブモードを含む)

項目	記号	適用端子	測定条件	規格値			単位	備考
				min.	typ.	max.		
入力"Low" レベル電圧	$V_{IL}$	$\overline{RES}$ 、 $\overline{WKP}_0 \sim \overline{WKP}_7$ 、 $\overline{IRQ}_0 \sim \overline{IRQ}_4$ 、 TMIF、TMIG、 SCK <sub>1</sub> 、SCK <sub>3</sub> 、ADTRG	$V_{CC} = 2.7 \sim 5.5V$	- 0.3		$0.2V_{CC}$	V	
				- 0.3		$0.1V_{CC}$		
		SI <sub>1</sub> 、RXD	$V_{CC} = 2.7 \sim 5.5V$	- 0.3		$0.3V_{CC}$	V	
				- 0.3		$0.2V_{CC}$		
		OSC <sub>1</sub>	$V_{CC} = 2.7 \sim 5.5V$	- 0.3		0.5	V	
				- 0.3		0.3		
		P1 <sub>0</sub> ~ P1 <sub>7</sub> 、 P2 <sub>0</sub> ~ P2 <sub>7</sub> 、 P5 <sub>0</sub> ~ P5 <sub>7</sub> 、 P6 <sub>0</sub> ~ P6 <sub>7</sub> 、 P7 <sub>0</sub> ~ P7 <sub>7</sub> 、 P8 <sub>0</sub> ~ P8 <sub>7</sub> 、 PA <sub>1</sub> ~ PA <sub>3</sub> 、 PB <sub>6</sub> 、PB <sub>7</sub>		$V_{CC} = 2.7 \sim 5.5V$	- 0.3		$0.3V_{CC}$	V
	- 0.3				$0.2V_{CC}$			
出力"High" レベル電圧	$V_{OH}$	P1 <sub>0</sub> ~ P1 <sub>7</sub> 、 P2 <sub>0</sub> ~ P2 <sub>6</sub> 、 P5 <sub>0</sub> ~ P5 <sub>7</sub> 、 P6 <sub>0</sub> ~ P6 <sub>7</sub> 、 P7 <sub>0</sub> ~ P7 <sub>7</sub> 、 P8 <sub>0</sub> ~ P8 <sub>7</sub> 、 PA <sub>1</sub> ~ PA <sub>3</sub>	$V_{CC} = 4.0 \sim 5.5V$ - $I_{OH} = 1.0mA$	$V_{CC} - 1.0$			V	
			$V_{CC} = 4.0 \sim 5.5V$ - $I_{OH} = 0.5mA$	$V_{CC} - 0.5$				
			- $I_{OH} = 0.1mA$	$V_{CC} - 0.5$				

14. 電気的特性

表 14.2 DC特性 (3)

(特記なき場合、 $V_{CC} = 2.2 \sim 5.5V$ 、 $V_{SS} = 0.0V$ 、 $T_a = -20 \sim +75$ 、サブアクティブモードを含む)

項目	記号	適用端子	測定条件	規格値			単位	備考	
				min.	typ.	max.			
出力"Low" レベル電圧	$V_{OL}$	P5 <sub>0</sub> ~ P5 <sub>7</sub> 、 P6 <sub>0</sub> ~ P6 <sub>7</sub> 、 P7 <sub>0</sub> ~ P7 <sub>7</sub> 、 P8 <sub>0</sub> ~ P8 <sub>7</sub> 、 PA <sub>1</sub> ~ PA <sub>3</sub>	$I_{OL} = 0.4mA$			0.5	V		
				P1 <sub>0</sub> ~ P1 <sub>7</sub> 、 P2 <sub>0</sub> ~ P2 <sub>6</sub>	$V_{CC} = 4.0 \sim 5.5V$ $I_{OL} = 10mA$				1.5
					$V_{CC} = 4.0 \sim 5.5V$ $I_{OL} = 1.6mA$				0.6
					$I_{OL} = 0.4mA$				0.5
入出力 リーク電流	$ I_{IL} $	RES、 P2 <sub>7</sub>	$V_{IN} = 0.5V \sim V_{CC} - 0.5V$			20	$\mu A$	*3	
						1		*2	
		OSC <sub>1</sub> 、 P1 <sub>0</sub> ~ P1 <sub>7</sub> 、 P2 <sub>0</sub> ~ P2 <sub>6</sub> 、 P5 <sub>0</sub> ~ P5 <sub>7</sub> 、 P6 <sub>0</sub> ~ P6 <sub>7</sub> 、 P7 <sub>0</sub> ~ P7 <sub>7</sub> 、 P8 <sub>0</sub> ~ P8 <sub>7</sub> 、 PA <sub>1</sub> ~ PA <sub>3</sub> 、 PB <sub>6</sub> 、 PB <sub>7</sub>	$V_{IN} = 0.5V \sim V_{CC} - 0.5V$			1	$\mu A$		
プルアップ MOS 電流	- $I_p$	P1 <sub>0</sub> ~ P1 <sub>7</sub> 、 P2 <sub>0</sub> ~ P2 <sub>6</sub> 、 P5 <sub>0</sub> ~ P5 <sub>7</sub> 、 P6 <sub>0</sub> ~ P6 <sub>7</sub>	$V_{CC} = 5V$ 、 $V_{IN} = 0V$	50		300	$\mu A$		
			$V_{CC} = 2.7V$ 、 $V_{IN} = 0V$		35		$\mu A$	参考値	

【注】 \*2 HD6433622、HD6433623、HD6433624、HD6433625、HD6433626、HD6433627 に適用します。

\*3 HD6473627 に適用します。

表 14.2 DC 特性 (4)

(特記なき場合、 $V_{CC} = 2.2 \sim 5.5V$ 、 $V_{SS} = 0.0V$ 、 $T_a = -20 \sim +75$ 、サブアクティブモードを含む)

項目	記号	適用端子	測定条件	規格値			単位	備考
				min.	typ.	max.		
入力容量	$C_{IN}$	電源端子を除く 全入力端子	$f = 1MHz$ 、 $V_{IN} = 0V$ 、 $T_a = 25$			15	pF	*3
		RES				80		
		P2 <sub>7</sub>				50		
アクティブ モード 消費電流	$I_{OPE1}$	$V_{CC}$	アクティブ (高速) モード $V_{CC} = 5V$ 、 $f_{OSC} = 10MHz$		6	9	mA	*4 *5
	$I_{OPE2}$	$V_{CC}$	アクティブ (中速) モード $V_{CC} = 5V$ 、 $f_{OSC} = 10MHz$		2	4	mA	*4 *5
スリープモード 消費電流	$I_{SLEEP}$	$V_{CC}$	$V_{CC} = 5V$ 、 $f_{OSC} = 10MHz$		3.5	6	mA	*4 *5
サブアクティブ モード消費電流	$I_{SUB}$	$V_{CC}$	$V_{CC} = 2.7V$ 32kHz 水晶発振器使用時 ( $I_{SUB} = I_{W}/2$ )		12	30	$\mu A$	*4 *5
			$V_{CC} = 2.7V$ 32kHz 水晶発振器使用時 ( $I_{SUB} = I_{W}/8$ )		6		$\mu A$	参考値 *4 *5
サブスリープモード 消費電流	$I_{SUBSP}$	$V_{CC}$	$V_{CC} = 2.7V$ 32kHz 水晶発振器使用時 ( $I_{SUB} = I_{W}/2$ )		5	15	$\mu A$	*4 *5
ウォッチモード 消費電流	$I_{WATCH}$	$V_{CC}$	$V_{CC} = 2.7V$ 32kHz 水晶発振器使用時			6	$\mu A$	*4 *5
スタンバイ モード消費電流	$I_{STBY}$	$V_{CC}$	32kHz 水晶発振器未使用時			5	$\mu A$	*4 *5
RAM データ 保持電圧	$V_{RAM}$	$V_{CC}$		2			V	

【注】 \*4 消費電流測定時の端子の状態

モード	内部状態	各端子	発振端子
アクティブ (高速) モード アクティブ (中速) モード	動作	$V_{CC}$	システムクロック発振器：水晶発振子 サブクロック発振器：X <sub>1</sub> 端子 = $V_{CC}$
スリープモード	タイマのみ動作	$V_{CC}$	
サブアクティブモード	動作	$V_{CC}$	システムクロック発振器：水晶発振子 サブクロック発振器：水晶発振子
サブスリープモード	タイマのみ動作 CPU は停止	$V_{CC}$	
ウォッチモード	時計用タイムベースのみ動作 CPU は停止	$V_{CC}$	
スタンバイモード	CPU、タイマともに停止	$V_{CC}$	システムクロック発振器：水晶発振子 サブクロック発振器：X <sub>1</sub> 端子 = $V_{CC}$

\*5 プルアップ MOS や出力バッファに流れる電流は除きます。

14. 電気的特性

表 14.2 DC特性 (5)

(特記なき場合、 $V_{CC} = 2.2 \sim 5.5V$ 、 $V_{SS} = 0.0V$ 、 $T_a = -20 \sim +75$ 、サブアクティブモードを含む)

項目	記号	適用端子	測定条件	規格値			単位	備考
				min.	typ.	max.		
出力"Low" レベル許容電流 (1端子あたり)	$I_{OL}$	ポート 1、2 以外の出力端子	$V_{CC} = 4.0V \sim 5.5V$			2	mA	
		ポート 1、2	$V_{CC} = 4.0V \sim 5.5V$			10		
		全出力端子				0.5		
出力"Low" レベル許容電流 (総和)	$I_{OL}$	ポート 1、2 以外の出力端子	$V_{CC} = 4.0V \sim 5.5V$			40	mA	
		ポート 1、2	$V_{CC} = 4.0V \sim 5.5V$			80		
		全出力端子				20		
出力"High" レベル許容電流 (1端子あたり)	$-I_{OH}$	全出力端子	$V_{CC} = 4.0V \sim 5.5V$			2	mA	
						0.2		
出力"High" レベル許容電流 (総和)	$-I_{OH}$	全出力端子	$V_{CC} = 4.0V \sim 5.5V$			30	mA	
						10		

## 14.2.3 AC 特性

制御信号タイミングを表 14.3 に、シリアルインタフェースタイミングを表 14.4、表 14.5 に示します。

表 14.3 制御信号タイミング (1)

(特記なき場合、 $V_{CC} = 2.2 \sim 5.5V$ 、 $V_{SS} = 0.0V$ 、 $T_a = -20 \sim +75$ 、サブアクティブモードを含む)

項目	記号	適用端子	測定条件	規格値			単位	備考
				min.	typ.	max.		
システムクロック 発振器発振周波数	$f_{OSC}$	OSC <sub>1</sub> 、OSC <sub>2</sub>	$V_{CC} = 2.7 \sim 5.5V$	2		10	MHz	
				2		5		
OSC クロック (OSC) サイクル時間	$t_{OSC}$	OSC <sub>1</sub> 、OSC <sub>2</sub>	$V_{CC} = 2.7 \sim 5.5V$	100		1000	ns	*1 図 14.1
				200		1000		
システムクロック ( ) サイクル時間	$t_{cyc}$			2		16	$t_{OSC}$	*1
						2000		
サブクロック発振器 発振周波数	$f_W$	X <sub>1</sub> 、X <sub>2</sub>			32.768		kHz	
ウォッチクロック (W) サイクル時間	$t_W$	X <sub>1</sub> 、X <sub>2</sub>			30.5		μs	
サブクロック (SUB) サイクル時間	$t_{subcyc}$			2		8	$t_W$	*2
インストラクション サイクル時間				2			$t_{cyc}$ $t_{subcyc}$	
発振安定時間 (水晶発振器)	$t_{rc}$	OSC <sub>1</sub> 、OSC <sub>2</sub>	$V_{CC} = 4.0 \sim 5.5V$			40	ms	
			$V_{CC} = 2.5 \sim 5.5V$			60		
発振安定時間	$t_{rc}$	X <sub>1</sub> 、X <sub>2</sub>	$V_{CC} = 2.5 \sim 5.5V$			2	s	
外部クロック "High"レベル幅	$t_{CPH}$	OSC <sub>1</sub>	$V_{CC} = 2.7 \sim 5.5V$	40			ns	図 14.1
				80				
外部クロック "Low"レベル幅	$t_{CPL}$	OSC <sub>1</sub>	$V_{CC} = 2.7 \sim 5.5V$	40			ns	図 14.1
				80				
外部クロック 立上り時間	$t_{CPr}$		$V_{CC} = 2.7 \sim 5.5V$			15	ns	図 14.1
						20		
外部クロック 立下り時間	$t_{CPf}$		$V_{CC} = 2.7 \sim 5.5V$			15	ns	図 14.1
						20		
RES 端子 "Low"レベル幅	$t_{REL}$	RES		18			$t_{cyc}$	図 14.2
							$t_{subcyc}$	
発振開始電圧	Vstart	OSC <sub>1</sub> 、OSC <sub>2</sub>		2.5			V	
		X <sub>1</sub> 、X <sub>2</sub>		2.5				
入力端子 "High"レベル幅	$t_{IH}$	IRQ <sub>0</sub> ~ IRQ <sub>4</sub> 、 WKP <sub>0</sub> ~ WKP <sub>7</sub> 、 ADTRG、TMIF、 TMIG		2			$t_{cyc}$ $t_{subcyc}$	図 14.3

## 14. 電気的特性

表 14.3 制御信号タイミング (2)

(特記なき場合、 $V_{CC} = 2.2 \sim 5.5V$ 、 $V_{SS} = 0.0V$ 、 $T_a = -20 \sim +75$ 、サブアクティブモードを含む)

項目	記号	適用端子	測定条件	規格値			単位	備考
				min.	typ.	max.		
入力端子 "Low"レベル幅	$t_{L}$	$\overline{IRQ}_0 \sim \overline{IRQ}_4$ 、 $\overline{WKP}_0 \sim \overline{WKP}_7$ 、 $\overline{ADTRG}$ 、 $TMIF$ 、 $TMIG$		2			$t_{cyc}$ $t_{subcyc}$	図 14.3

【注】 \*1 外部クロックを入力する場合は1~10MHzとなります。

\*2 システムコントロールレジスタ2 (SYSCR2) のSA1、SA0の設定により決定します。

表 14.4 シリアルインタフェース (SCI1) タイミング

(特記なき場合、 $V_{CC} = 2.2 \sim 5.5V$ 、 $V_{SS} = 0.0V$ 、 $T_a = -20 \sim +75$ 、サブアクティブモードを含む)

項目	記号	適用端子	測定条件	規格値			単位	備考
				min.	typ.	max.		
入力転送クロック サイクル時間	$t_{scyc}$	$SCK_1$		2			$t_{cyc}$	図 14.4
入力転送クロック "High"レベル幅	$t_{SCKH}$	$SCK_1$		0.4			$t_{scyc}$	図 14.4
入力転送クロック "Low"レベル幅	$t_{SCKL}$	$SCK_1$		0.4			$t_{scyc}$	図 14.4
入力転送クロック 立上り時間	$t_{SCKr}$	$SCK_1$	$V_{CC} = 4.0 \sim 5.5V$			60 80	ns	図 14.4
入力転送クロック 立下り時間	$t_{SCKf}$	$SCK_1$	$V_{CC} = 4.0 \sim 5.5V$			60 80	ns	図 14.4
シリアル出力データ 遅延時間	$t_{SOD}$	$SO_1$	$V_{CC} = 4.0 \sim 5.5V$			200 350	ns	図 14.4
シリアル入力データ セットアップ時間	$t_{SIS}$	$SI_1$	$V_{CC} = 4.0 \sim 5.5V$	200 400			ns	図 14.4
シリアル入力データ ホールド時間	$t_{SIH}$	$SI_1$	$V_{CC} = 4.0 \sim 5.5V$	200 400			ns	図 14.4

表 14.5 シリアルインタフェース (SCI3) タイミング

(特記なき場合、 $V_{CC} = 2.2 \sim 5.5V$ 、 $V_{SS} = 0.0V$ 、 $T_a = -20 \sim +75$ 、サブアクティブモードを含む)

項目	記号	測定条件	規格値			単位	備考
			min.	typ.	max.		
入力 クロックサイクル	調歩同期	$t_{SCYC}$	4			$t_{cyc}$	図 14.5
	クロック同期		6				
入力クロックパルス幅	$t_{SCKW}$		0.4		0.6	$t_{SCYC}$	図 14.5
送信データ遅延時間 (クロック同期)	$t_{TXD}$	$V_{CC} = 4.0 \sim 5.5V$			1	$t_{cyc}$	図 14.6
					1		
受信データセットアップ時間 (クロック同期)	$t_{RXS}$	$V_{CC} = 4.0 \sim 5.5V$	200			ns	図 14.6
			400				
受信データホールド時間 (クロック同期)	$t_{RXH}$	$V_{CC} = 4.0 \sim 5.5V$	200			ns	図 14.6
			400				

## 14.2.4 A/D 変換器特性

A/D 変換器特性を表 14.6 に示します。

表 14.6 A/D 変換器特性

(特記なき場合、 $V_{CC} = 2.2 \sim 5.5V$ 、 $V_{SS} = 0.0V$ 、 $T_a = -20 \sim +75$ )

項目	記号	適用端子	測定条件	規格値			単位	備考
				min.	typ.	max.		
アナログ入力電圧	$AV_{IN}$	$AN_6$ 、 $AN_7$		- 0.3		$V_{CC} + 0.3$	V	
アナログ入力容量	$C_{AIN}$	$AN_6$ 、 $AN_7$				30	pF	
許容信号源 インピーダンス	$R_{AIN}$					10	k	
分解能						8	ビット	
非直線性誤差						$\pm 2.0$	LSB	
量子化誤差						$\pm 0.5$		
絶対精度						$\pm 2.5$		
変換時間			$V_{CC} = 4.5 \sim 5.5V$	12.4		248	$\mu s$	
				24.8		248		

## 14.2.5 DTMF 特性

DTMF 特性を表 14.7 に示します。

表 14.7 DTMF 特性

(特記なき場合、 $V_{CC} = 2.2 \sim 5.5V$ 、 $V_{SS} = 0.0V$ 、 $T_a = -20 \sim +75$ 、サブアクティブモードを含む)

項目	記号	適用端子	測定条件	規格値			単位	備考
				min.	typ.	max.		
基準レベル 電源電圧	$V_{T_{ref}}$	$V_{T_{ref}}$		2.2		$V_{CC}+0.3$	V	
DTMF 出力電圧 (Row 側)	$V_{OR}$	TONED	$V_{T_{ref}} - GND=2.2V$ $R_L=100k$	550	723		mVrms	図 14.8 *
DTMF 出力電圧 (Column 側)	$V_{OC}$	TONED	$V_{T_{ref}} - GND=2.2V$ $R_L=100k$	567	760		mVrms	図 14.8 *
DTMF 出力歪	%DISDT	TONED	$V_{T_{ref}} - GND=2.2V$ $R_L=100k$		3	7	%	図 14.8
DTMF 出力比	$dB_{CR}$	TONED	$V_{T_{ref}} - GND=2.2V$ $R_L=100k$		2.5		dB	図 14.8

【注】 \*  $V_{OR}$ 、 $V_{OC}$ はそれぞれ単独波形出力時の出力電圧です。

## 14.3 動作タイミング

動作タイミングを図 14.1 ~ 図 14.6 に示します。

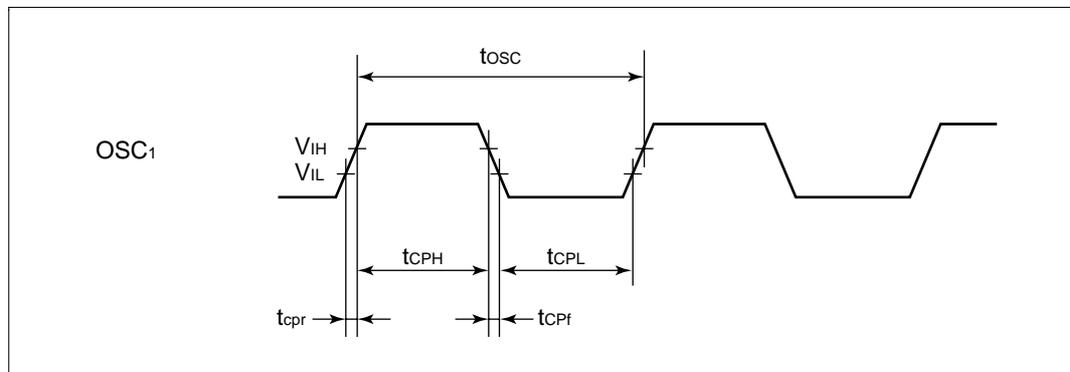


図 14.1 システムクロック入力タイミング

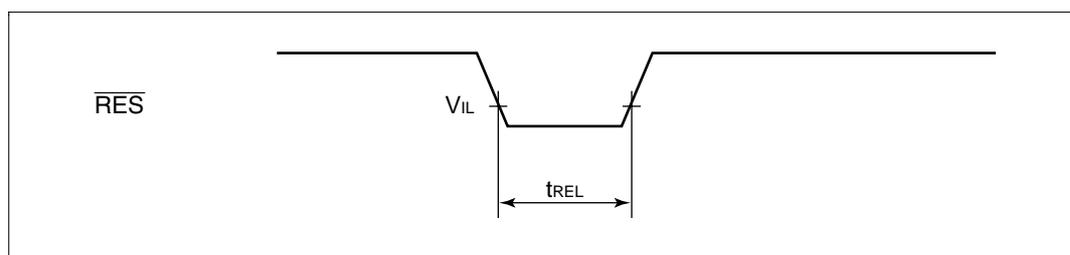


図 14.2  $\overline{RES}$  端子"Low"レベル幅タイミング

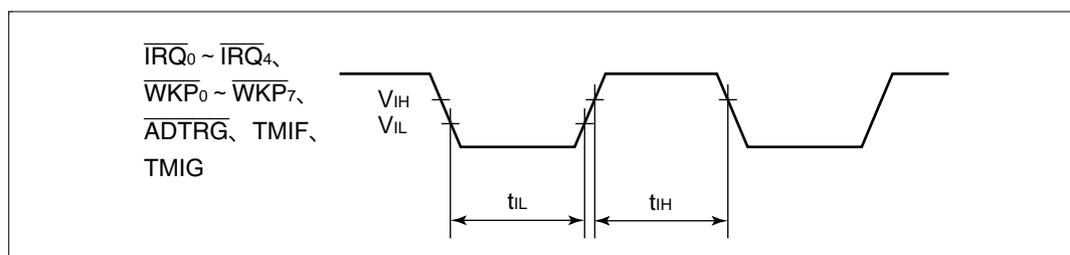


図 14.3 入力タイミング

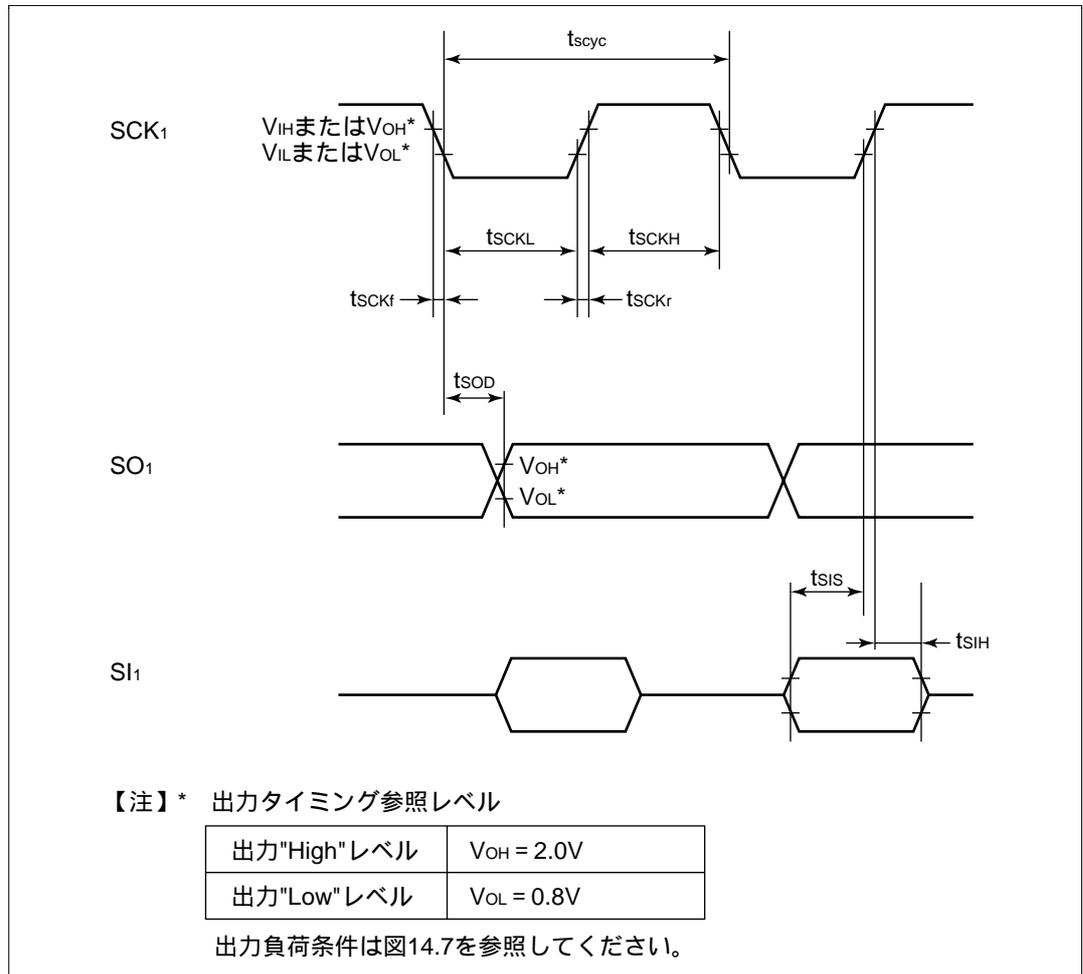


図 14.4 SCI1 入出力タイミング

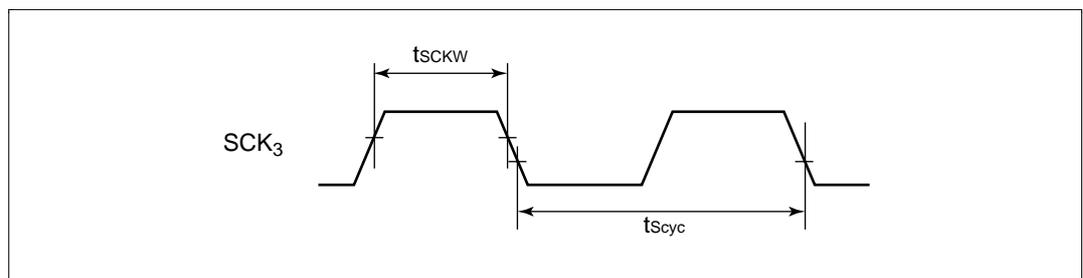


図 14.5 SCK3 入力クロックタイミング

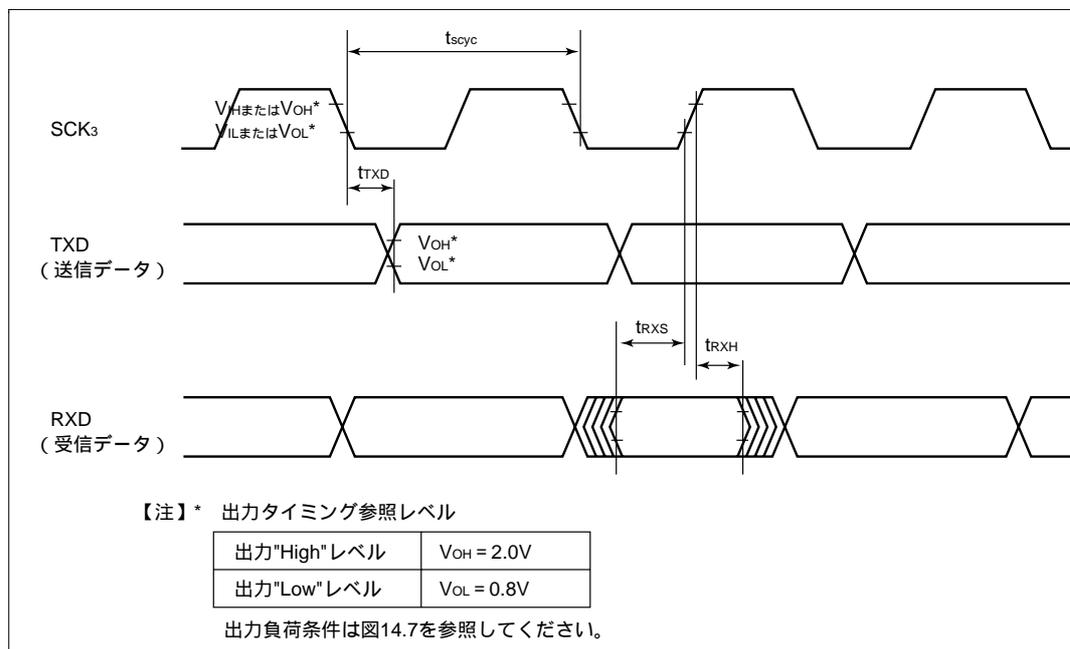


図 14.6 SCI3 クロック同期式モード入出力タイミング

## 14.4 出力負荷回路

出力負荷条件を図 14.7 に示します。

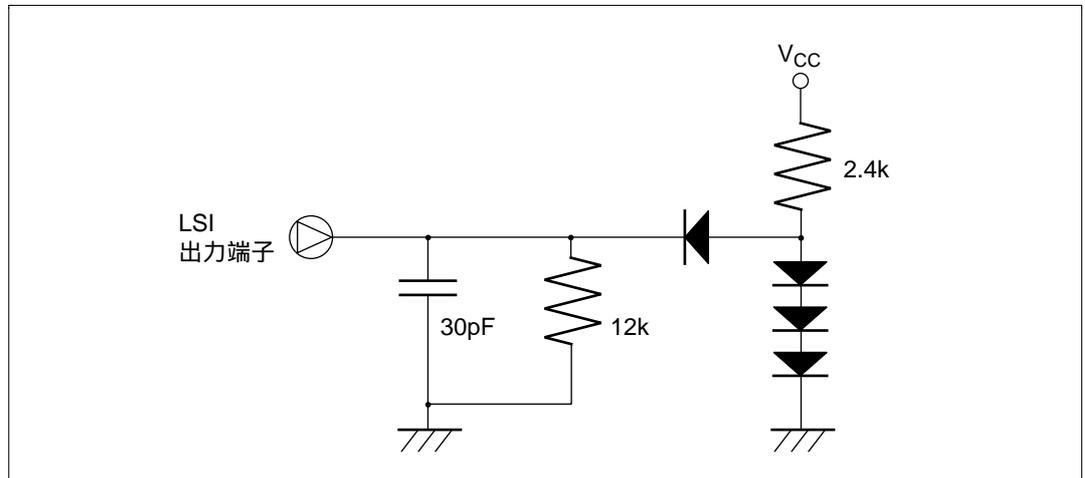


図 14.7 出力負荷条件

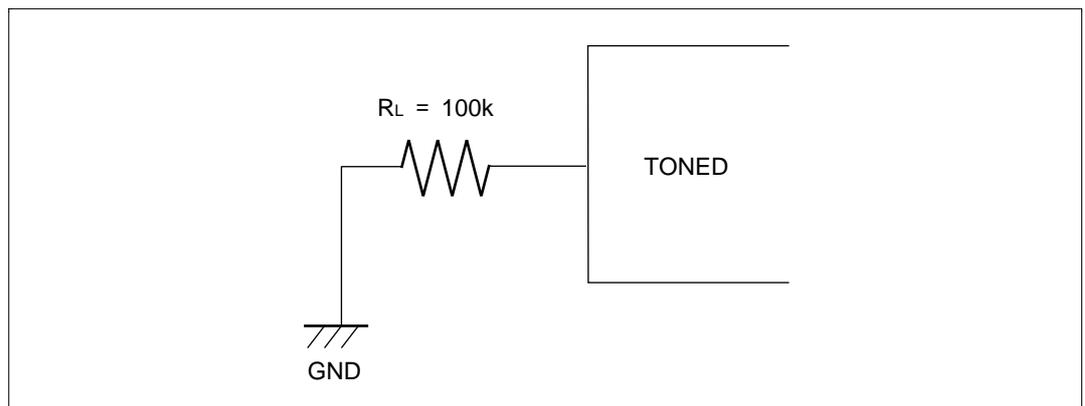


図 14.8 TONED 負荷回路

---

# 付録

---

## 付録 目次

A.	命令	327	
	A.1	命令一覧	327
	A.2	オペレーションコードマップ	337
	A.3	命令実行ステート数	338
B.	内部 I/O レジスタ一覧	344	
	B.1	アドレス一覧	344
	B.2	機能一覧	348
C.	I/O ポートブロック図	385	
	C.1	ポート 1 ブロック図	385
	C.2	ポート 2 ブロック図	392
	C.3	ポート 5 ブロック図	400
	C.4	ポート 6 ブロック図	401
	C.5	ポート 7 ブロック図	402
	C.6	ポート 8 ブロック図	403
	C.7	ポート A ブロック図	404
	C.8	ポート B ブロック図	405
D.	各処理状態におけるポートの状態	406	
E.	ROM 発注手順	407	
	E.1	ROM 書換え品開発の流れ（発注手順）	407
	E.2	ROM 発注時の注意事項	408
F.	型名一覧	409	
G.	外形寸法図	410	



## A. 命令

### A.1 命令一覧

#### 《オペレーションの記号》

Rd8/16	汎用レジスタ (デスティネーション側) 8 ビット / 16 ビット
Rs8/16	汎用レジスタ (ソース側) 8 ビット / 16 ビット
Rn8/16	汎用レジスタ 8 ビット / 16 ビット
CCR	コンディションコードレジスタ
N	CCR の N (ネガティブ) フラグ
Z	CCR の Z (ゼロ) フラグ
V	CCR の V (オーバフロー) フラグ
C	CCR の C (キャリ) フラグ
PC	プログラムカウンタ
SP	スタックポインタ
#xx:3/8/16	イミディエイトデータ 3 ビット / 8 ビット / 16 ビット
d:8/16	ディスプレイメント 8 ビット / 16 ビット
@aa:8/16	絶対アドレス 8 ビット / 16 ビット
+	加算
-	減算
×	乗算
÷	除算
	論理積
	論理和
⊕	排他的論理和
	転送
	論理的補数

#### 《コンディションコードの記号》

記号	
↓	実行結果にしたがって変化することを表します。
*	不確定であることを表します (値を保証しません)。
0	常に"0"にクリアされることを表します。
	実行結果に影響を受けないことを表します。

表 A.1 命令セット一覧 (1)

アーキテクチャ	サイズ	アドレッシングモード / 命令長 (バイト)						オペレーション	コンディションコード							実行 ステート 数
		#xx:8/16	Rn	@Rn	@(d:16, Rn)	@-Rn/@Rn+	@aa:8/16		@(d:8, PC)	@aa	I	H	N	Z	V	
MOV	MOV.B #xx:8, Rd	B	2									↑	↑	0	—	2
	MOV.B Rs, Rd	B		2								↑	↑	0	—	2
	MOV.B @Rs, Rd	B			2							↑	↑	0	—	4
	MOV.B @(d:16, Rs), Rd	B				4						↑	↑	0	—	6
	MOV.B @Rs+, Rd	B					2					↑	↑	0	—	6
	MOV.B @aa:8, Rd	B						2				↑	↑	0	—	4
	MOV.B @aa:16, Rd	B						4				↑	↑	0	—	6
	MOV.B Rs, @Rd	B		2								↑	↑	0	—	4
	MOV.B Rs, @(d:16, Rd)	B				4						↑	↑	0	—	6
	MOV.B Rs, @-Rd	B					2					↑	↑	0	—	6
	MOV.B Rs, @aa:8	B						2				↑	↑	0	—	4
	MOV.B Rs, @aa:16	B						4				↑	↑	0	—	6
	MOV.W #xx:16, Rd	W	4									↑	↑	0	—	4
	MOV.W Rs, Rd	W		2								↑	↑	0	—	2
	MOV.W @Rs, Rd	W			2							↑	↑	0	—	4
	MOV.W @(d:16, Rs), Rd	W				4						↑	↑	0	—	6
MOV.W @Rs+, Rd	W					2					↑	↑	0	—	6	
MOV.W @aa:16, Rd	W						4				↑	↑	0	—	6	
MOV.W Rs, @Rd	W		2								↑	↑	0	—	4	
MOV.W Rs, @(d:16, Rd)	W				4						↑	↑	0	—	6	

表 A.1 命令セット一覧 (2)

ニーモニック	サイズ	アドレッシングモード / 命令長 (バイト)						オペレーション	コンディションコード							実行 スタート 数
		#xx:8/16	Rn	@Rn	@(d:16, Rn)	@-Rn/@Rn+	@aa:8/16		@(d:8, PC)	@@aa	I	H	N	Z	V	
MOV	W					2					↑	↑	0		6	
		MOV.W Rs, @-Rd														
POP	W						4				↑	↑	0		6	
		MOV.W Rs, @aa:16				2										
PUSH	W														6	
		POP Rd														
ADD	B														6	
		PUSH Rs														
ADDX	B														6	
		ADD.B #xx:8, Rd	2													
ADDS	W														6	
		ADD.B Rs, Rd	2													
SUBX	B														6	
		ADD.W Rs, Rd	2													
SUBX	B														6	
		ADDX.B #xx:8, Rd	2													
SUBX	B														6	
		ADDX.B Rs, Rd	2													
SUBX	B														6	
		ADDX.B #xx:8, Rd	2													
SUBX	B														6	
		ADDX.B Rs, Rd	2													

表 A.1 命令セット一覧 (3)

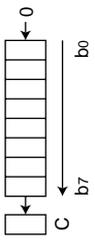
ニーモニック	サイズ	アドレッシングモード / 命令長 (バイト)				オペレーション	コンディションコード							実行 スト 数	
		Rn #xx:8/16	@Rn @ (d:16, Rn)	@-Rn/@Rn+ @aa:8/16	@(d:8, PC) @@aa		I	H	N	Z	V	C			
SUBS						Rd16-1 Rd16	—	—	—	—	—	—	—	—	2
		2				Rd16-2 Rd16	—	—	—	—	—	—	—	—	2
DEC		2				Rd8-1 Rd8	—	—	↑↑	↑↑	—	—	—	—	2
DAS		2				Rd8 10進補正 Rd8	—	*	↑↑	↑↑	*	—	—	—	2
NEG		2				0-Rd Rd	—	↑↑	↑↑	↑↑	↑↑	—	—	—	2
CMP		2				Rd8-#xx:8	—	↑↑	↑↑	↑↑	↑↑	—	—	—	2
		2				Rd8-Rs8	—	↑↑	↑↑	↑↑	↑↑	—	—	—	2
		2				Rd16-Rs16	—	(1)	↑↑	↑↑	↑↑	↑↑	—	—	2
MULXU		2				Rd8 x Rs8 Rd16	—	—	—	—	—	—	—	—	14
DIVXU		2				Rd16 ÷ Rs8 Rd16 (RdH: 余り、RdL: 商)	—	—	(5)	(6)	—	—	—	—	14
AND		2				Rd8 #xx:8 Rd8	—	—	↑↑	↑↑	0	—	—	—	2
		2				Rd8 Rs8 Rd8	—	—	↑↑	↑↑	0	—	—	—	2
OR		2				Rd8 #xx:8 Rd8	—	—	↑↑	↑↑	0	—	—	—	2
		2				Rd8 Rs8 Rd8	—	—	↑↑	↑↑	0	—	—	—	2
XOR		2				Rd8 ⊕ #xx:8 Rd8	—	—	↑↑	↑↑	0	—	—	—	2
		2				Rd8 ⊕ Rs8 Rd8	—	—	↑↑	↑↑	0	—	—	—	2
NOT		2				Rd Rd	—	—	↑↑	↑↑	0	—	—	—	2
SHAL		2					—	—	↑↑	↑↑	↑↑	—	—	—	2



表 A.1 命令セット一覧 (5)

ニーモニック	サイズ	アドレッシングモード / 命令長 (バイト)				オペレーション	コンディションコード							実行 入 数	
		#xx:8/16	Rn @Rn	@(d:16, Rn) @-Rn/@Rn+	@aa:8/16 @(d:8, PC) @aa		I	H	N	Z	V	C			
BSET	BSET #xx:3, @aa:8				4	(#xx:3 of @aa:8) 1	—	—	—	—	—	—	—	—	8
	BSET Rn, Rd	2				(Rn8 of Rd8) 1	—	—	—	—	—	—	—	—	2
	BSET Rn, @Rd		4			(Rn8 of @Rd16) 1	—	—	—	—	—	—	—	—	8
	BSET Rn, @aa:8				4	(Rn8 of @aa:8) 1	—	—	—	—	—	—	—	—	8
BCLR	BCLR #xx:3, Rd	2				(#xx:3 of Rd8) 0	—	—	—	—	—	—	—	—	2
	BCLR #xx:3, @Rd		4			(#xx:3 of @Rd16) 0	—	—	—	—	—	—	—	—	8
	BCLR #xx:3, @aa:8				4	(#xx:3 of @aa:8) 0	—	—	—	—	—	—	—	—	8
	BCLR Rn, Rd	2				(Rn8 of Rd8) 0	—	—	—	—	—	—	—	—	2
	BCLR Rn, @Rd		4			(Rn8 of @Rd16) 0	—	—	—	—	—	—	—	—	8
	BCLR Rn, @aa:8				4	(Rn8 of @aa:8) 0	—	—	—	—	—	—	—	—	8
	BNOT #xx:3, Rd	2				(#xx:3 of Rd8) (#xx:3 of Rd8)	—	—	—	—	—	—	—	—	2
	BNOT #xx:3, @Rd		4			(#xx:3 of @Rd16) (#xx:3 of @Rd16)	—	—	—	—	—	—	—	—	8
BNOT	BNOT #xx:3, @aa:8				4	(#xx:3 of @aa:8) (#xx:3 of @aa:8)	—	—	—	—	—	—	—	—	8
	BNOT Rn, Rd	2				(Rn8 of Rd8) (Rn8 of Rd8)	—	—	—	—	—	—	—	—	2
	BNOT Rn, @Rd		4			(Rn8 of @Rd16) (Rn8 of @Rd16)	—	—	—	—	—	—	—	—	8
	BNOT Rn, @aa:8				4	(Rn8 of @aa:8) (Rn8 of @aa:8)	—	—	—	—	—	—	—	—	8
	BTST #xx:3, Rd	2				(#xx:3 of Rd8) Z	—	—	—	—	↑	—	—	—	2
	BTST #xx:3, @Rd		4			(#xx:3 of @Rd16) Z	—	—	—	—	↑	—	—	—	6
BTST	BTST #xx:3, @aa:8				4	(#xx:3 of @aa:8) Z	—	—	—	—	↑	—	—	—	6
	BTST Rn, Rd	2				(Rn8 of Rd8) Z	—	—	—	—	↑	—	—	—	2

表 A.1 命令セット一覧 (6)

ニーモニック	サイズ	アドレッシングモード / 命令長 (バイト)						オペレーション	コンディションコード							実行 入数
		#xx:8/16	Rn	@Rn	@(d:16, Rn)	@-Rn/@Rn+	@aa:8/16		@(d:8, PC)	@aa	I	H	N	Z	V	
BTST	B		4					(Rn8 of @Rd16) Z	—	—	—	↑	—	—	—	6
	B						4	(Rn8 of @aa:8) Z	—	—	—	↑	—	—	—	6
BLD	B	2						(#xx:3 of Rd8) C	—	—	—	—	—	—	↑	2
	B		4					(#xx:3 of @Rd16) C	—	—	—	—	—	—	↑	6
	B						4	(#xx:3 of @aa:8) C	—	—	—	—	—	—	↑	6
BILD	B	2						(#xx:3 of Rd8) C	—	—	—	—	—	—	↑	2
	B		4					(#xx:3 of @Rd16) C	—	—	—	—	—	—	↑	6
	B						4	(#xx:3 of @aa:8) C	—	—	—	—	—	—	↑	6
BST	B	2						C (#xx:3 of Rd8)	—	—	—	—	—	—	—	2
	B		4					C (#xx:3 of @Rd16)	—	—	—	—	—	—	—	8
	B						4	C (#xx:3 of @aa:8)	—	—	—	—	—	—	—	8
BIST	B	2						C̄ (#xx:3 of Rd8)	—	—	—	—	—	—	—	2
	B		4					C̄ (#xx:3 of @Rd16)	—	—	—	—	—	—	—	8
	B						4	C̄ (#xx:3 of @aa:8)	—	—	—	—	—	—	—	8
BAND	B	2						C (#xx:3 of Rd8) C	—	—	—	—	—	—	↑	2
	B		4					C (#xx:3 of @Rd16) C	—	—	—	—	—	—	↑	6
	B						4	C (#xx:3 of @aa:8) C	—	—	—	—	—	—	↑	6
BIAND	B	2						C̄ (#xx:3 of Rd8) C	—	—	—	—	—	—	↑	2
	B		4					C̄ (#xx:3 of @Rd16) C	—	—	—	—	—	—	↑	6
	B						4	C̄ (#xx:3 of @aa:8) C	—	—	—	—	—	—	↑	6
BOR	B	2						C (#xx:3 of Rd8) C	—	—	—	—	—	—	↑	2
	B		4					C (#xx:3 of @Rd16) C	—	—	—	—	—	—	↑	6
	B						4	C (#xx:3 of @aa:8) C	—	—	—	—	—	—	↑	6

表 A.1 命令セット一覧 (7)

ニーモニック	サイズ	アドレッシングモード / 命令長 (バイト)				オペレーション		コンディションコード							実行 入ポート 数
		Rn #xx:8/16	@Rn @(d:16, Rn)	@-Rn/@Rn+ @aa:8/16	@(d:8, PC) @aa	分岐条件	I	H	N	Z	V	C			
BIOR	B	2				C (#xx:3 of Rd8) C	-	-	-	-	-	↑	2		
	B		4			C (#xx:3 of @Rd16) C	-	-	-	-	-	↑	6		
	B			4		C (#xx:3 of @aa:8) C	-	-	-	-	-	↑	6		
BXOR	B	2				C ⊕ (#xx:3 of Rd8) C	-	-	-	-	-	↑	2		
	B		4			C ⊕ (#xx:3 of @Rd16) C	-	-	-	-	-	↑	6		
	B			4		C ⊕ (#xx:3 of @aa:8) C	-	-	-	-	-	↑	6		
BIXOR	B	2				C ⊕ (#xx:3 of Rd8) C	-	-	-	-	-	↑	2		
	B		4			C ⊕ (#xx:3 of @Rd16) C	-	-	-	-	-	↑	6		
	B			4		C ⊕ (#xx:3 of @aa:8) C	-	-	-	-	-	↑	6		
Bcc	-					PC PC+d:8	-	-	-	-	-	-	4		
	-					PC PC+2	-	-	-	-	-	-	4		
BHI d:8	-					if condition C Z=0	-	-	-	-	-	-	4		
BLS d:8	-					is true then C Z=1	-	-	-	-	-	-	4		
BCC d:8 (BHS d:8)	-					PC PC+d:8 C=0	-	-	-	-	-	-	4		
BCS d:8 (BLO d:8)	-					else next; C=1	-	-	-	-	-	-	4		
BNE d:8	-					Z=0	-	-	-	-	-	-	4		
BEQ d:8	-					Z=1	-	-	-	-	-	-	4		
BVC d:8	-					V=0	-	-	-	-	-	-	4		
BVS d:8	-					V=1	-	-	-	-	-	-	4		
BPL d:8	-					N=0	-	-	-	-	-	-	4		
BMI d:8	-					N=1	-	-	-	-	-	-	4		
BGE d:8	-					N ⊕ V=0	-	-	-	-	-	-	4		
BLT d:8	-					N ⊕ V=1	-	-	-	-	-	-	4		
BGT d:8	-					Z (N ⊕ V)=0	-	-	-	-	-	-	4		
BLE d:8	-					Z (N ⊕ V)=1	-	-	-	-	-	-	4		

表 A.1 命令セット一覧 (8)

ニーモニック	サイズ	アドレッシングモード / 命令長 (バイト)						オペレーション	コンディションコード							実行 スタート 数			
		#xx:8/16	Rn	@Rn	@(d:16, Rn)	@-Rn/@Rn+	@aa:8/16		@(d:8, PC)	@@aa	I	H	N	Z	V		C		
JMP	JMP @Rn	—		2															4
	JMP @aa:16	—					4												6
	JMP @@aa:8	—							2										8
BSR	BSR d:8	—							2										6
		SP-2 SP																	
		PC @SP																	
JSR	JSR @Rn	—		2															6
		PC @SP																	
		PC Rn16																	
JSR @aa:16	—						4												8
		SP-2 SP																	
		PC @SP																	
JSR @@aa:8	—											2							8
		SP-2 SP																	
		PC @SP																	
RTS	RTS	—																	8
		PC @SP																	
RTE	RTE	—											2						8
		SP+2 SP																	
		CCR @SP																	
RTE	—																		10
		SP+2 SP																	
		PC @SP																	
RTE	—																		10
		SP+2 SP																	
		PC @SP																	

表 A.1 命令セット一覧 (9)

ニーモニック	サイズ	アドレッシングモード / 命令長 (バイト)				オペレーション	コンディションコード							実行 スタート 回数	
		Rn #xx:8/16	@Rn @Rn	@(d:16, Rn) @-Rn/@Rn+	@aa:8/16 @(d:8, PC) @aa		I	H	N	Z	V	C			
SLEEP	—					2	低消費電力状態に遷移	—	—	—	—	—	—	—	2
LDC	B	2					#xx:8 CCR	↑	↑	↑	↑	↑	↑	↑	2
	B		2				Rs8 CCR	↑	↑	↑	↑	↑	↑	↑	2
STC	B		2				CCR Rd8	—	—	—	—	—	—	—	2
ANDC	B	2					CCR #xx:8 CCR	↑	↑	↑	↑	↑	↑	↑	2
ORC	B	2					CCR #xx:8 CCR	↑	↑	↑	↑	↑	↑	↑	2
XORC	B	2					CCR ⊕ #xx:8 CCR	↑	↑	↑	↑	↑	↑	↑	2
NOP	—					2	PC PC+2	—	—	—	—	—	—	—	2
EEMOV	—					4	if R4L 0 Repeat @R5 @R6 R5+1 R5 R6+1 R6 R4L-1 R4L Until R4L=0 else next;	—	—	—	—	—	—	—	(4)

- 【注】 (1) : ビット11から桁上がりまたはビット11へ桁下がりが発生したとき"1"にセットされ、それ以外のとき"0"にクリアされます。  
 (2) : 演算結果がゼロのとき、演算前の値を保持し、それ以外のとき"0"にクリアされます。  
 (3) : 補正結果に桁上がりが発生したとき"1"にセットされ、それ以外のとき演算前の値を保持します。  
 (4) : 実行スタート数は、R4Lの設定値がnのとき4n+9となります。  
 (5) : 除数が負のとき"1"にセットされ、それ以外のとき"0"にクリアされます。  
 (6) : 除数がゼロのとき"1"にセットされ、それ以外のとき"0"にクリアされます。

## A.2 オペレーションコードマップ

表A.2にオペレーションコードマップを示します。表A.2では、命令コードの第1バイト（第1ワードのビット15~8）についてのみ示しています。

表A.2 オペレーションコードマップ

第2バイトの最上位ビット（命令コードの第1ワードのビット7）が0の場合を示します。  
 第2バイトの最上位ビット（命令コードの第1ワードのビット7）が1の場合を示します。



LO HI	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	SLEEP	STC	LDC	ORC	XORC	ANDC	LDC	ADD		INC	ADDS	MOV	ADDX		DAA
1	SHLL SHAL	SHLR SHAR	ROTL ROTR	ROTXL ROTR	OR	XOR	AND	NOT NEG	SUB		DEC	SUBS	CMP	SUBX		DAS
2	MOV															
3	MOV															
4	BRA	BRN	BHI	BLS	BCC	BCS	BNE	BEQ	BVC	BVS	BPL	BMI	BGE	BLT	BGT	BLE
5	MULXU	DIVXU			RTS	BSR	RTE				JMP				JSR	
6	BSET	BNOT	BCLR	BTST	BOR	BXOR	BAND	BST	BLD	BID	BAND	BIST	MOV*			
7					BIOR	BIXOR	BIAND	BILD	BUILD	MOV		EEPMOV				ビット操作命令
8	ADD															
9	ADDX															
A	CMP															
B	SUBX															
C	OR															
D	XOR															
E	AND															
F	MOV															

【注】 \* PUSH、POP命令の機械語はMOV命令と同一です。

### A.3 命令実行ステート数

H8/300L CPUの各命令についての実行状態と実行ステート数の計算方法を示します。

表A.4に命令の実行状態として、命令実行中に行われる命令フェッチ、データリード/ライト等のサイクル数を示し、表A.3に各々のサイクルに必要なステート数を示します。命令の実行ステート数は次の計算式で計算されます。

$$\text{実行ステート数} = I \cdot S_I + J \cdot S_J + K \cdot S_K + L \cdot S_L + M \cdot S_M + N \cdot S_N$$

#### 実行ステート数計算例

(例) 内蔵ROMより命令をフェッチし、内蔵RAMをアクセスした場合

1. BSET #0, @FF00

表A.4より

$$I = L = 2, J = K = M = N = 0$$

表A.3より

$$S_I = 2, S_L = 2$$

$$\text{実行ステート数} = 2 \times 2 + 2 \times 2 = 8$$

内蔵ROMより命令をフェッチし、内蔵ROMより分岐アドレスをリード、スタック領域は内蔵RAMとした場合

2. JSR @@30

表A.4より

$$I = 2, J = K = 1, L = M = N = 0$$

表A.3より

$$S_I = S_J = S_K = 2$$

$$\text{実行ステート数} = 2 \times 2 + 1 \times 2 + 1 \times 2 = 8$$

表A.3 実行状態(サイクル)に要するステート数

実行状態(サイクル)		アクセス対象	
		内蔵メモリ	内蔵周辺モジュール
命令フェッチ	$S_I$	2	X X X X X
分岐アドレスリード	$S_J$		
スタック操作	$S_K$		
バイトデータアクセス	$S_L$		
ワードデータアクセス	$S_M$		
内部動作	$S_N$	1	

【注】 \* 内蔵周辺モジュールによって異なります。詳細は、「2.9.1 データアクセスに関する注意事項」を参照してください。

表 A.4 命令の実行状態（サイクル数）（1）

命令	ニーモニック	命令	分岐アドレス	スタック	バイトデータ	ワードデータ	内部動作
		フェッチ	リード	操作	アクセス	アクセス	
		I	J	K	L	M	N
ADD	ADD.B #xx:8, Rd	1					
	ADD.B Rs, Rd	1					
	ADD.W Rs, Rd	1					
ADDS	ADDS.W #1, Rd	1					
	ADDS.W #2, Rd	1					
ADDX	ADDX.B #xx:8, Rd	1					
	ADDX.B Rs, Rd	1					
AND	AND.B #xx:8, Rd	1					
	AND.B Rs, Rd	1					
ANDC	ANDC #xx:8, CCR	1					
BAND	BAND #xx:3, Rd	1					
	BAND #xx:3, @Rd	2			1		
	BAND #xx:3, @aa:8	2			1		
Bcc	BRA d:8 (BT d:8)	2					
	BRN d:8 (BF d:8)	2					
	BHI d:8	2					
	BLS d:8	2					
	BCC d:8 (BHS d:8)	2					
	BCS d:8 (BLO d:8)	2					
	BNE d:8	2					
	BEQ d:8	2					
	BVC d:8	2					
	BVS d:8	2					
	BPL d:8	2					
	BMI d:8	2					
	BGE d:8	2					
	BLT d:8	2					
	BGT d:8	2					
BLE d:8	2						
BCLR	BCLR #xx:3, Rd	1					
	BCLR #xx:3, @Rd	2			2		
	BCLR #xx:3, @aa:8	2			2		
	BCLR Rn, Rd	1					
	BCLR Rn, @Rd	2			2		
	BCLR Rn, @aa:8	2			2		
BIAND	BIAND #xx:3, Rd	1					
	BIAND #xx:3, @Rd	2			1		
	BIAND #xx:3, @aa:8	2			1		

表 A.4 命令の実行状態（サイクル数）（2）

命令	二ーモニク	命令	分岐アドレス	スタック	バイトデータ	ワードデータ	内部動作
		フェッチ	リード	操作	アクセス	アクセス	
		I	J	K	L	M	N
BILD	BILD #xx:3, Rd	1					
	BILD #xx:3, @Rd	2			1		
	BILD #xx:3, @aa:8	2			1		
BIOR	BIOR #xx:3, Rd	1					
	BIOR #xx:3, @Rd	2			1		
	BIOR #xx:3, @aa:8	2			1		
BIST	BIST #xx:3, Rd	1					
	BIST #xx:3, @Rd	2			2		
	BIST #xx:3, @aa:8	2			2		
BIXOR	BIXOR #xx:3, Rd	1					
	BIXOR #xx:3, @Rd	2			1		
	BIXOR #xx:3, @aa:8	2			1		
BLD	BLD #xx:3, Rd	1					
	BLD #xx:3, @Rd	2			1		
	BLD #xx:3, @aa:8	2			1		
BNOT	BNOT #xx:3, Rd	1					
	BNOT #xx:3, @Rd	2			2		
	BNOT #xx:3, @aa:8	2			2		
	BNOT Rn, Rd	1					
	BNOT Rn, @Rd	2			2		
	BNOT Rn, @aa:8	2			2		
BOR	BOR #xx:3, Rd	1					
	BOR #xx:3, @Rd	2			1		
	BOR #xx:3, @aa:8	2			1		
BSET	BSET #xx:3, Rd	1					
	BSET #xx:3, @Rd	2			2		
	BSET #xx:3, @aa:8	2			2		
	BSET Rn, Rd	1					
	BSET Rn, @Rd	2			2		
	BSET Rn, @aa:8	2			2		
BSR	BSR d:8	2		1			
BST	BST #xx:3, Rd	1					
	BST #xx:3, @Rd	2			2		
	BST #xx:3, @aa:8	2			2		

表 A.4 命令の実行状態（サイクル数）（3）

命令	ニーモニック	命令	分岐アドレス	スタック	バイトデータ	ワードデータ	内部動作
		フェッチ	リード	操作	アクセス	アクセス	
		I	J	K	L	M	N
BTST	BTST #xx:3, Rd	1					
	BTST #xx:3, @Rd	2			1		
	BTST #xx:3, @aa:8	2			1		
	BTST Rn, Rd	1					
	BTST Rn, @Rd	2			1		
	BTST Rn, @aa:8	2			1		
BXOR	BXOR #xx:3, Rd	1					
	BXOR #xx:3, @Rd	2			1		
	BXOR #xx:3, @aa:8	2			1		
CMP	CMP.B #xx:8, Rd	1					
	CMP.B Rs, Rd	1					
	CMP.W Rs, Rd	1					
DAA	DAA.B Rd	1					
DAS	DAS.B Rd	1					
DEC	DEC.B Rd	1					
DIVXU	DIVXU.B Rs, Rd	1					12
EEPMOV	EEPMOV	2			$2n + 2^{*1}$		1
INC	INC.B Rd	1					
JMP	JMP @Rn	2					
	JMP @aa:16	2					2
	JMP @@aa:8	2	1				2
JSR	JSR @Rn	2		1			
	JSR @aa:16	2		1			2
	JSR @@aa:8	2	1	1			
LDC	LDC #xx:8, CCR	1					
	LDC Rs, CCR	1					
MOV	MOV.B #xx:8, Rd	1					
	MOV.B Rs, Rd	1					
	MOV.B @Rs, Rd	1			1		
	MOV.B @(d:16, Rs), Rd	2			1		
	MOV.B @Rs+, Rd	1			1		2
	MOV.B @aa:8, Rd	1			1		
	MOV.B @aa:16, Rd	2			1		

表 A.4 命令の実行状態（サイクル数）（4）

命令	ニーモニック	命令	分岐アドレス	スタック	バイトデータ	ワードデータ	内部動作
		フェッチ	リード	操作	アクセス	アクセス	
		I	J	K	L	M	
MOV	MOV.B Rs, @Rd	1			1		2
	MOV.B Rs, @(d:16, Rd)	2			1		
	MOV.B Rs, @-Rd	1			1		
	MOV.B Rs, @aa:8	1			1		
	MOV.B Rs, @aa:16	2			1		
	MOV.W #xx:16, Rd	2					2
	MOV.W Rs, Rd	1					
	MOV.W @Rs, Rd	1				1	
	MOV.W @(d:16, Rs), Rd	2				1	
	MOV.W @Rs+, Rd	1				1	
	MOV.W @aa:16, Rd	2				1	
	MOV.W Rs, @Rd	1				1	
	MOV.W Rs, @(d:16, Rd)	2				1	
	MOV.W Rs, @-Rd	1				1	
	MOV.W Rs, @aa:16	2				1	
MULXU	MULXU.B Rs, Rd	1					12
NEG	NEG.B Rd	1					
NOP	NOP	1					
NOT	NOT.B Rd	1					
OR	OR.B #xx:8, Rd	1					
	OR.B Rs, Rd	1					
ORC	ORC #xx:8, CCR	1					
ROTL	ROTL.B Rd	1					
ROTR	ROTR.B Rd	1					
ROTXL	ROTXL.B Rd	1					
ROTXR	ROTXR.B Rd	1					
RTE	RTE	2		2			2
RTS	RTS	2		1			2
SHAL	SHAL.B Rd	1					
SHAR	SHAR.B Rd	1					
SHLL	SHLL.B Rd	1					
SHLR	SHLR.B Rd	1					
SLEEP	SLEEP	1					
STC	STC CCR, Rd	1					
SUB	SUB.B Rs, Rd	1					
	SUB.W Rs, Rd	1					

表 A.4 命令の実行状態 (サイクル数) (5)

命令	二-モニック	命令 フェッチ	分岐アドレス リード	スタック 操作	バイトデータ アクセス	ワードデータ アクセス	内部動作
		I	J	K	L	M	
SUBS	SUBS.W #1, Rd	1					
	SUBS.W #2, Rd	1					
POP	POP Rd	1		1			2
PUSH	PUSH Rs	1		1			2
SUBX	SUBX.B #xx:8, Rd	1					
	SUBX.B Rs, Rd	1					
XOR	XOR.B #xx:8, Rd	1					
	XOR.B Rs, Rd	1					
XORC	XORC #xx:8, CCR	1					

【注】 \*1 n は R4L の設定値です。ソース側、デスティネーション側のアクセスが、それぞれ (n + 1) 回行われます。

## B. 内部 I/O レジスタ一覧

### B.1 アドレス一覧

下位 アドレス	レジスタ名	ビット名								モジュール名
		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	
H'90										
H'91										
H'92										
H'93										
H'94										
H'95										
H'96										
H'97										
H'98	PMR1	IRQ3	IRQ2	IRQ1		TMIG	TMOFH	TMOFL	TMOW	
H'99	PMR2	IRQ0		POF1	NCS	SO1	SI1	SCK1	IRQ4	
H'9A	PMR6						TXD			
H'9B	PMR5	WKP <sub>7</sub>	WKP <sub>6</sub>	WKP <sub>5</sub>	WKP <sub>4</sub>	WKP <sub>3</sub>	WKP <sub>2</sub>	WKP <sub>1</sub>	WKP <sub>0</sub>	I/Oポート
H'9C	PUCR1	PUCR1 <sub>7</sub>	PUCR1 <sub>6</sub>	PUCR1 <sub>5</sub>	PUCR1 <sub>4</sub>	PUCR1 <sub>3</sub>	PUCR1 <sub>2</sub>	PUCR1 <sub>1</sub>	PUCR1 <sub>0</sub>	
H'9D	PUCR2	PUCR2 <sub>7</sub>	PUCR2 <sub>6</sub>	PUCR2 <sub>5</sub>	PUCR2 <sub>4</sub>	PUCR2 <sub>3</sub>	PUCR2 <sub>2</sub>	PUCR2 <sub>1</sub>	PUCR2 <sub>0</sub>	
H'9E	PUCR5	PUCR5 <sub>7</sub>	PUCR5 <sub>6</sub>	PUCR5 <sub>5</sub>	PUCR5 <sub>4</sub>	PUCR5 <sub>3</sub>	PUCR5 <sub>2</sub>	PUCR5 <sub>1</sub>	PUCR5 <sub>0</sub>	
H'9F	PUCR6	PUCR6 <sub>7</sub>	PUCR6 <sub>6</sub>	PUCR6 <sub>5</sub>	PUCR6 <sub>4</sub>	PUCR6 <sub>3</sub>	PUCR6 <sub>2</sub>	PUCR6 <sub>1</sub>	PUCR6 <sub>0</sub>	
H'A0	SCR1	SNC1	SNC0			CKS3	CKS2	CKS1	CKS0	
H'A1	SCSR1		SOL	ORER					STF	
H'A2	SDRU	SDRU7	SDRU6	SDRU5	SDRU4	SDRU3	SDRU2	SDRU1	SDRU0	SCI1
H'A3	SDRL	SDRL7	SDRL6	SDRL5	SDRL4	SDRL3	SDRL2	SDRL1	SDRL0	
H'A4										
H'A5										
H'A6										
H'A7										
H'A8	SMR	COM	CHR	PE	PM	STOP	MP	CKS1	CKS0	
H'A9	BRR	BRR7	BRR6	BRR5	BRR4	BRR3	BRR2	BRR1	BRR0	
H'AA	SCR3	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	SCI3
H'AB	TDR	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0	
H'AC	SSR	TDRE	RDRF	OER	FER	PER	TEND	MPBR	MPBT	
H'AD	RDR	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1	RDR0	
H'AE										
H'AF										
H'B0	TMA	TMA7	TMA6	TMA5		TMA3	TMA2	TMA1	TMA0	タイマ A

下位 アドレス	レジスタ名	ビット名								モジュール名
		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	
H'B1	TCA	TCA7	TCA6	TCA5	TCA4	TCA3	TCA2	TCA1	TCA0	タイマ A
H'B2	DTCR	DTEN		CLOE	RWOE	CLF1	CLF0	RWF1	RWF0	DTMF
H'B3	DTLR				DTL4	DTL3	DTL2	DTL1	DTL0	発生回路
H'B4										
H'B5										
H'B6	TCRF	TOLH	CKSH2	CKSH1	CKSH0	TOLL	CKSL2	CKSL1	CKSL0	タイマ F
H'B7	TCSR	OVFH	CMFH	OVIEH	CCLRH	OVFL	CMFL	OVIEL	CCLRL	
H'B8	TCFH	TCFH7	TCFH6	TCFH5	TCFH4	TCFH3	TCFH2	TCFH1	TCFH0	
H'B9	TCFL	TCFL7	TCFL6	TCFL5	TCFL4	TCFL3	TCFL2	TCFL1	TCFL0	
H'BA	OCRFB	OCRFB7	OCRFB6	OCRFB5	OCRFB4	OCRFB3	OCRFB2	OCRFB1	OCRFB0	
H'BB	OCRFL	OCRFL7	OCRFL6	OCRFL5	OCRFL4	OCRFL3	OCRFL2	OCRFL1	OCRFL0	
H'BC	TMG	OVFH	OVFL	OVIE	IIEGS	CCLR1	CCLR0	CKS1	CKS0	タイマ G
H'BD	ICRGF	ICRGF7	ICRGF6	ICRGF5	ICRGF4	ICRGF3	ICRGF2	ICRGF1	ICRGF0	
H'BE	ICRGR	ICRGR7	ICRGR6	ICRGR5	ICRGR4	ICRGR3	ICRGR2	ICRGR1	ICRGR0	
H'BF										
H'C0										
H'C1										
H'C2										
H'C3										
H'C4	AMR	CKS	TRGE	CKS1		CH3	CH2	CH1	CH0	A/D 変換器
H'C5	ADRR	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0	
H'C6	ADSR	ADSF								
H'C7										
H'C8										
H'C9										
H'CA										
H'CB										
H'CC										
H'CD										
H'CE										
H'CF										
H'D0										
H'D1										
H'D2										
H'D3										
H'D4	PDR1	P1 <sub>7</sub>	P1 <sub>6</sub>	P1 <sub>5</sub>	P1 <sub>4</sub>	P1 <sub>3</sub>	P1 <sub>2</sub>	P1 <sub>1</sub>	P1 <sub>0</sub>	I/O ポート
H'D5	PDR2	P2 <sub>7</sub>	P2 <sub>6</sub>	P2 <sub>5</sub>	P2 <sub>4</sub>	P2 <sub>3</sub>	P2 <sub>2</sub>	P2 <sub>1</sub>	P2 <sub>0</sub>	

下位 アドレス	レジスタ名	ビット名								モジュール名
		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	
H'D6										
H'D7										
H'D8	PDR5	P5 <sub>7</sub>	P5 <sub>6</sub>	P5 <sub>5</sub>	P5 <sub>4</sub>	P5 <sub>3</sub>	P5 <sub>2</sub>	P5 <sub>1</sub>	P5 <sub>0</sub>	I/Oポート
H'D9	PDR6	P6 <sub>7</sub>	P6 <sub>6</sub>	P6 <sub>5</sub>	P6 <sub>4</sub>	P6 <sub>3</sub>	P6 <sub>2</sub>	P6 <sub>1</sub>	P6 <sub>0</sub>	
H'DA	PDR7	P7 <sub>7</sub>	P7 <sub>6</sub>	P7 <sub>5</sub>	P7 <sub>4</sub>	P7 <sub>3</sub>	P7 <sub>2</sub>	P7 <sub>1</sub>	P7 <sub>0</sub>	
H'DB	PDR8	P8 <sub>7</sub>	P8 <sub>6</sub>	P8 <sub>5</sub>	P8 <sub>4</sub>	P8 <sub>3</sub>	P8 <sub>2</sub>	P8 <sub>1</sub>	P8 <sub>0</sub>	
H'DC										
H'DD	PDRA					PA <sub>3</sub>	PA <sub>2</sub>	PA <sub>1</sub>		
H'DE	PDRB	PB <sub>7</sub>	PB <sub>6</sub>							
H'DF										
H'E0										
H'E1										
H'E2										
H'E3										
H'E4	PCR1	PCR1 <sub>7</sub>	PCR1 <sub>6</sub>	PCR1 <sub>5</sub>	PCR1 <sub>4</sub>	PCR1 <sub>3</sub>	PCR1 <sub>2</sub>	PCR1 <sub>1</sub>	PCR1 <sub>0</sub>	I/Oポート
H'E5	PCR2	PCR2 <sub>7</sub>	PCR2 <sub>6</sub>	PCR2 <sub>5</sub>	PCR2 <sub>4</sub>	PCR2 <sub>3</sub>	PCR2 <sub>2</sub>	PCR2 <sub>1</sub>	PCR2 <sub>0</sub>	
H'E6										
H'E7										
H'E8	PCR5	PCR5 <sub>7</sub>	PCR5 <sub>6</sub>	PCR5 <sub>5</sub>	PCR5 <sub>4</sub>	PCR5 <sub>3</sub>	PCR5 <sub>2</sub>	PCR5 <sub>1</sub>	PCR5 <sub>0</sub>	I/Oポート
H'E9	PCR6	PCR6 <sub>7</sub>	PCR6 <sub>6</sub>	PCR6 <sub>5</sub>	PCR6 <sub>4</sub>	PCR6 <sub>3</sub>	PCR6 <sub>2</sub>	PCR6 <sub>1</sub>	PCR6 <sub>0</sub>	
H'EA	PCR7	PCR7 <sub>7</sub>	PCR7 <sub>6</sub>	PCR7 <sub>5</sub>	PCR7 <sub>4</sub>	PCR7 <sub>3</sub>	PCR7 <sub>2</sub>	PCR7 <sub>1</sub>	PCR7 <sub>0</sub>	
H'EB	PCR8	PCR8 <sub>7</sub>	PCR8 <sub>6</sub>	PCR8 <sub>5</sub>	PCR8 <sub>4</sub>	PCR8 <sub>3</sub>	PCR8 <sub>2</sub>	PCR8 <sub>1</sub>	PCR8 <sub>0</sub>	
H'EC										
H'ED	PCRA					PCRA <sub>3</sub>	PCRA <sub>2</sub>	PCRA <sub>1</sub>		
H'EE										
H'EF										
H'F0	SYSCR1	SSBY	STS2	STS1	STS0	LSON				システム コントロール
H'F1	SYSCR2				NESEL	DTON	MSON	SA1	SA0	
H'F2	IEGR				IEG4	IEG3	IEG2	IEG1	IEG0	
H'F3	IENR1	IENTA	IENS1	IENWP	IEN4	IEN3	IEN2	IEN1	IEN0	
H'F4	IENR2	IENDT	IENAD		IENTG	IENTFH	IENTFL			
H'F5										
H'F6	IRR1	IRRRTA	IRRS1		IRRI4	IRRI3	IRRI2	IRRI1	IRRI0	システム コントロール
H'F7	IRR2	IRRDT	IRRAD		IRRTG	IRRTFH	IRRTFL			
H'F8										

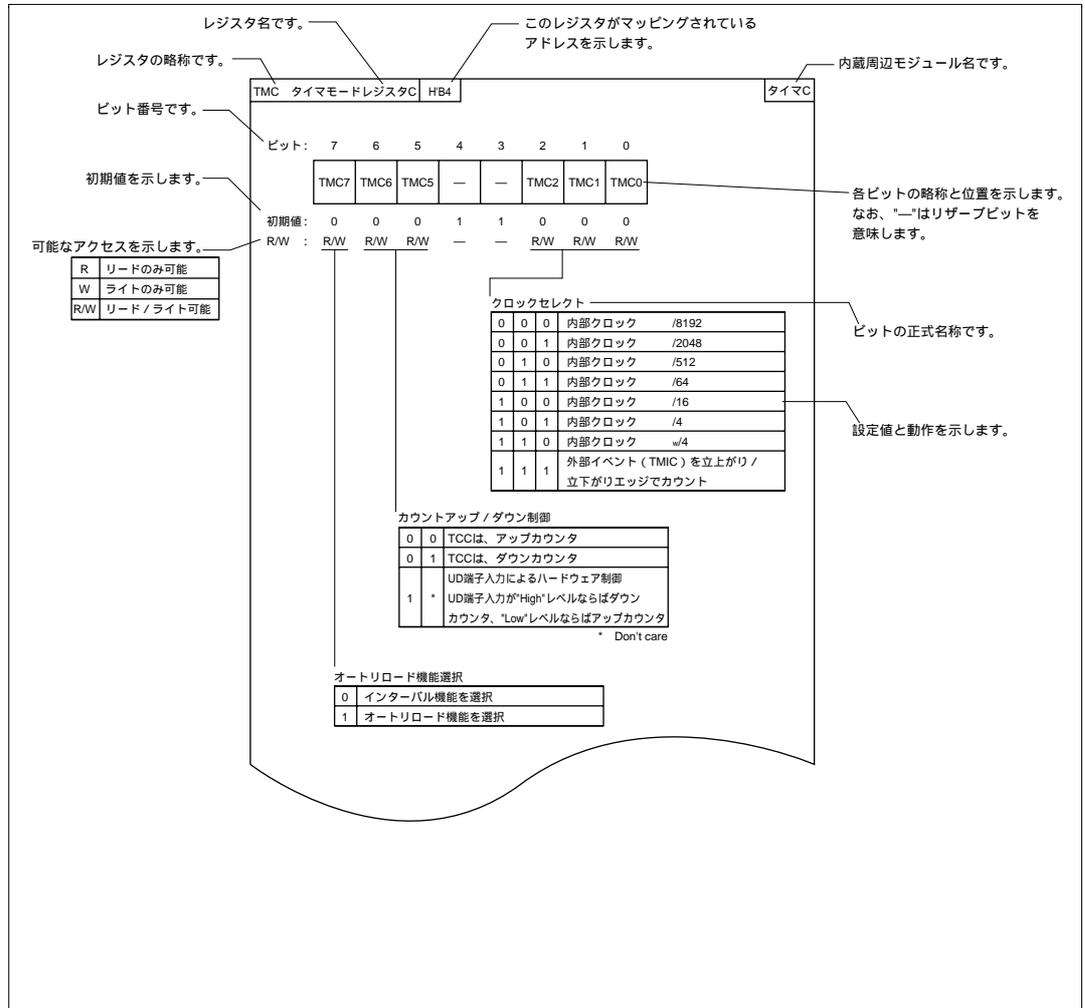
下位 アドレス	レジスタ名	ビット名								モジュール名
		ビット7	ビット6	ビット5	ビット4	ビット3	ビット2	ビット1	ビット0	
H'F9	IWPR	IWPF7	IWPF6	IWPF5	IWPF4	IWPF3	IWPF2	IWPF1	IWPF0	システム コントロール
H'FA										
H'FB										
H'FC										
H'FD										
H'FE										
H'FF										

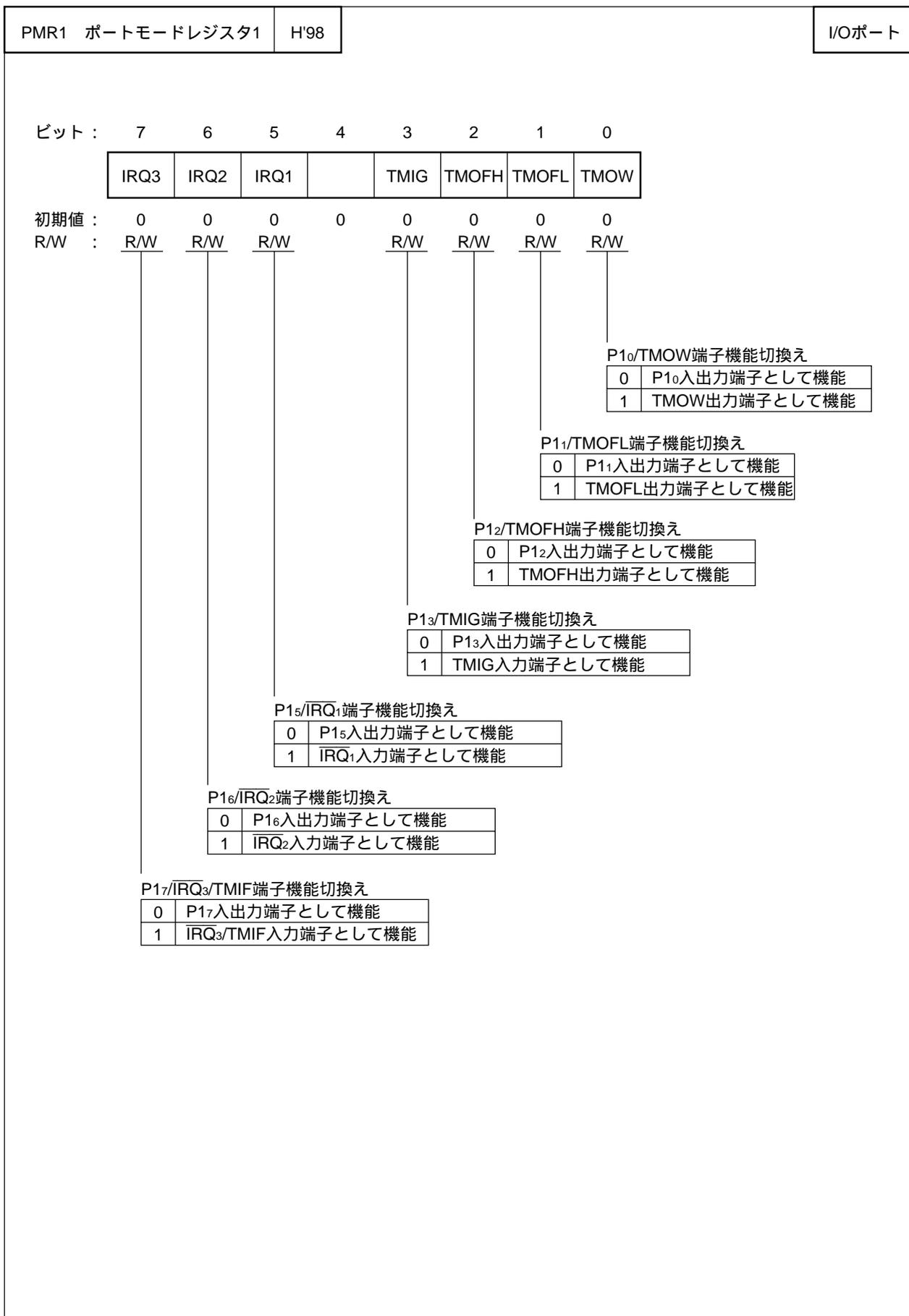
## 記号説明

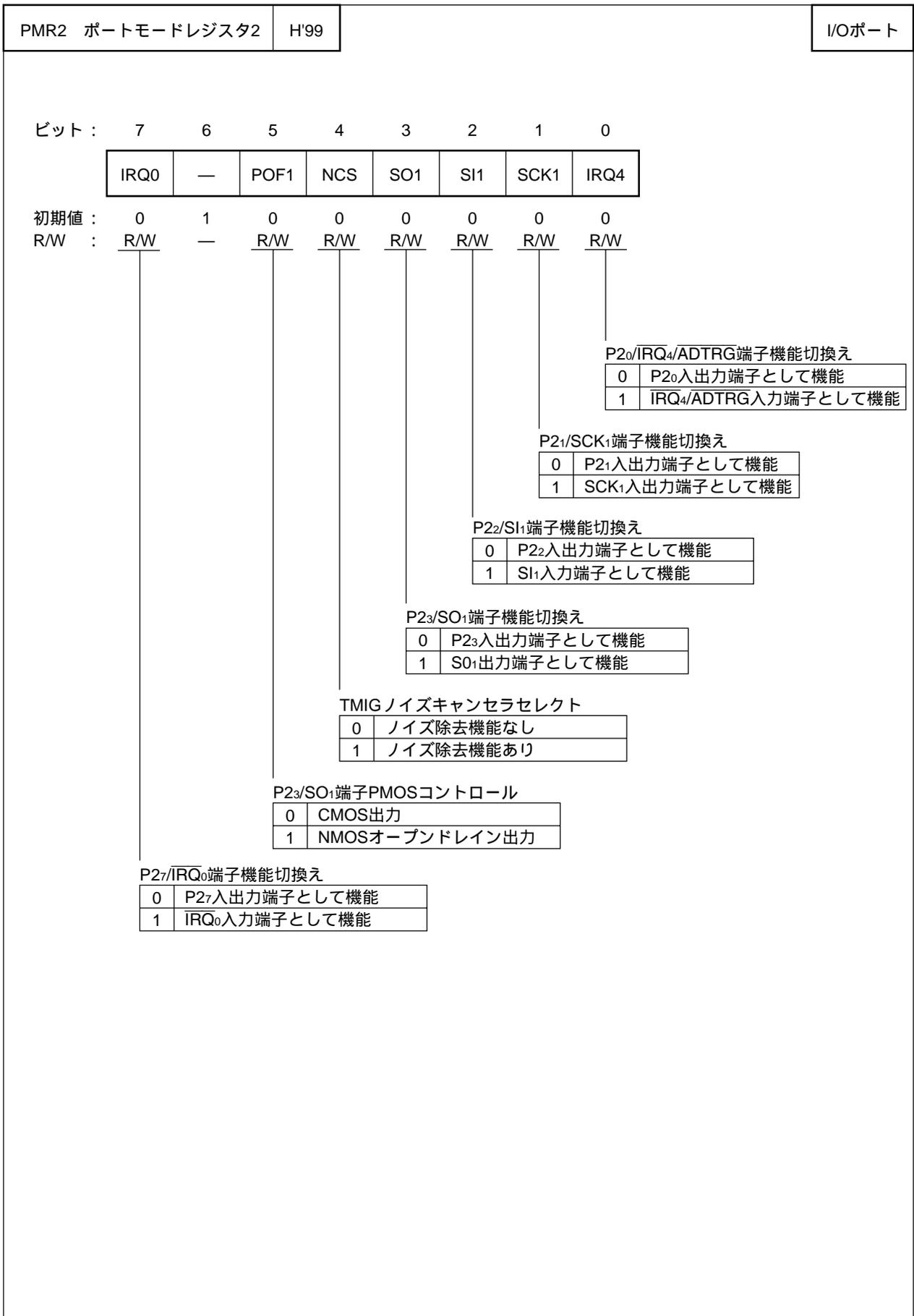
SCI1 : シリアルコミュニケーションインタフェース 1

SCI3 : シリアルコミュニケーションインタフェース 3

## B.2 機能一覧







PMR6 ポートモードレジスタ6	H'9A		I/Oポート												
<p>ビット：    7        6        5        4        3        2        1        0</p> <table border="1" style="margin-left: 40px; border-collapse: collapse; text-align: center;"> <tr> <td style="width: 20px;">—</td><td style="width: 20px;">—</td><td style="width: 20px;">—</td><td style="width: 20px;">—</td><td style="width: 20px;">—</td><td style="width: 20px;">TXD</td><td style="width: 20px;">—</td><td style="width: 20px;">—</td> </tr> </table> <p>初期値：    1        1        1        1        1        0        0        0</p> <p>R/W：        —        —        R/W    R/W    —        R/W    R/W    R/W</p> <div style="margin-left: 150px; margin-top: 20px;"> <p>P2<sub>6</sub>/TXD端子機能切換え (TXD)</p> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr> <td style="width: 20px;">0</td><td>P2<sub>6</sub>入出力端子として機能</td> </tr> <tr> <td>1</td><td>TXD出力端子として機能</td> </tr> </table> </div>				—	—	—	—	—	TXD	—	—	0	P2 <sub>6</sub> 入出力端子として機能	1	TXD出力端子として機能
—	—	—	—	—	TXD	—	—								
0	P2 <sub>6</sub> 入出力端子として機能														
1	TXD出力端子として機能														
<p>ビット：    7        6        5        4        3        2        1        0</p> <table border="1" style="margin-left: 40px; border-collapse: collapse; text-align: center;"> <tr> <td style="width: 20px;">WKP<sub>7</sub></td><td style="width: 20px;">WKP<sub>6</sub></td><td style="width: 20px;">WKP<sub>5</sub></td><td style="width: 20px;">WKP<sub>4</sub></td><td style="width: 20px;">WKP<sub>3</sub></td><td style="width: 20px;">WKP<sub>2</sub></td><td style="width: 20px;">WKP<sub>1</sub></td><td style="width: 20px;">WKP<sub>0</sub></td> </tr> </table> <p>初期値：    0        0        0        0        0        0        0        0</p> <p>R/W：        R/W    R/W    R/W    R/W    R/W    R/W    R/W    R/W</p> <div style="margin-left: 150px; margin-top: 20px;"> <p>P5<sub>n</sub>/WKP<sub>n</sub>端子機能切換え</p> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr> <td style="width: 20px;">0</td><td>P5<sub>n</sub>入出力端子として機能</td> </tr> <tr> <td>1</td><td>WKP<sub>n</sub>入力端子として機能</td> </tr> </table> <p style="text-align: right;">(n=7~0)</p> </div>				WKP <sub>7</sub>	WKP <sub>6</sub>	WKP <sub>5</sub>	WKP <sub>4</sub>	WKP <sub>3</sub>	WKP <sub>2</sub>	WKP <sub>1</sub>	WKP <sub>0</sub>	0	P5 <sub>n</sub> 入出力端子として機能	1	WKP <sub>n</sub> 入力端子として機能
WKP <sub>7</sub>	WKP <sub>6</sub>	WKP <sub>5</sub>	WKP <sub>4</sub>	WKP <sub>3</sub>	WKP <sub>2</sub>	WKP <sub>1</sub>	WKP <sub>0</sub>								
0	P5 <sub>n</sub> 入出力端子として機能														
1	WKP <sub>n</sub> 入力端子として機能														
<p>ビット：    7        6        5        4        3        2        1        0</p> <table border="1" style="margin-left: 40px; border-collapse: collapse; text-align: center;"> <tr> <td style="width: 20px;">PUCR1<sub>7</sub></td><td style="width: 20px;">PUCR1<sub>6</sub></td><td style="width: 20px;">PUCR1<sub>5</sub></td><td style="width: 20px;">PUCR1<sub>4</sub></td><td style="width: 20px;">PUCR1<sub>3</sub></td><td style="width: 20px;">PUCR1<sub>2</sub></td><td style="width: 20px;">PUCR1<sub>1</sub></td><td style="width: 20px;">PUCR1<sub>0</sub></td> </tr> </table> <p>初期値：    0        0        0        0        0        0        0        0</p> <p>R/W：        R/W    R/W    R/W    R/W    R/W    R/W    R/W    R/W</p>				PUCR1 <sub>7</sub>	PUCR1 <sub>6</sub>	PUCR1 <sub>5</sub>	PUCR1 <sub>4</sub>	PUCR1 <sub>3</sub>	PUCR1 <sub>2</sub>	PUCR1 <sub>1</sub>	PUCR1 <sub>0</sub>				
PUCR1 <sub>7</sub>	PUCR1 <sub>6</sub>	PUCR1 <sub>5</sub>	PUCR1 <sub>4</sub>	PUCR1 <sub>3</sub>	PUCR1 <sub>2</sub>	PUCR1 <sub>1</sub>	PUCR1 <sub>0</sub>								

<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%; padding: 2px;">PUCR2 ポートブルアップコントロールレジスタ2</td> <td style="width: 20%; padding: 2px;">H'9D</td> </tr> </table>	PUCR2 ポートブルアップコントロールレジスタ2	H'9D	I/Oポート						
PUCR2 ポートブルアップコントロールレジスタ2	H'9D								
<p>ビット：    7        6        5        4        3        2        1        0</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 2px;">PUCR2<sub>7</sub></td> <td style="padding: 2px;">PUCR2<sub>6</sub></td> <td style="padding: 2px;">PUCR2<sub>5</sub></td> <td style="padding: 2px;">PUCR2<sub>4</sub></td> <td style="padding: 2px;">PUCR2<sub>3</sub></td> <td style="padding: 2px;">PUCR2<sub>2</sub></td> <td style="padding: 2px;">PUCR2<sub>1</sub></td> <td style="padding: 2px;">PUCR2<sub>0</sub></td> </tr> </table> <p>初期値：    0        0        0        0        0        0        0        0</p> <p>R/W    :    R/W    R/W    R/W    R/W    R/W    R/W    R/W</p>		PUCR2 <sub>7</sub>	PUCR2 <sub>6</sub>	PUCR2 <sub>5</sub>	PUCR2 <sub>4</sub>	PUCR2 <sub>3</sub>	PUCR2 <sub>2</sub>	PUCR2 <sub>1</sub>	PUCR2 <sub>0</sub>
PUCR2 <sub>7</sub>	PUCR2 <sub>6</sub>	PUCR2 <sub>5</sub>	PUCR2 <sub>4</sub>	PUCR2 <sub>3</sub>	PUCR2 <sub>2</sub>	PUCR2 <sub>1</sub>	PUCR2 <sub>0</sub>		
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%; padding: 2px;">PUCR5 ポートブルアップコントロールレジスタ5</td> <td style="width: 20%; padding: 2px;">H'9E</td> </tr> </table>	PUCR5 ポートブルアップコントロールレジスタ5	H'9E	I/Oポート						
PUCR5 ポートブルアップコントロールレジスタ5	H'9E								
<p>ビット：    7        6        5        4        3        2        1        0</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 2px;">PUCR5<sub>7</sub></td> <td style="padding: 2px;">PUCR5<sub>6</sub></td> <td style="padding: 2px;">PUCR5<sub>5</sub></td> <td style="padding: 2px;">PUCR5<sub>4</sub></td> <td style="padding: 2px;">PUCR5<sub>3</sub></td> <td style="padding: 2px;">PUCR5<sub>2</sub></td> <td style="padding: 2px;">PUCR5<sub>1</sub></td> <td style="padding: 2px;">PUCR5<sub>0</sub></td> </tr> </table> <p>初期値：    0        0        0        0        0        0        0        0</p> <p>R/W    :    R/W    R/W    R/W    R/W    R/W    R/W    R/W</p>		PUCR5 <sub>7</sub>	PUCR5 <sub>6</sub>	PUCR5 <sub>5</sub>	PUCR5 <sub>4</sub>	PUCR5 <sub>3</sub>	PUCR5 <sub>2</sub>	PUCR5 <sub>1</sub>	PUCR5 <sub>0</sub>
PUCR5 <sub>7</sub>	PUCR5 <sub>6</sub>	PUCR5 <sub>5</sub>	PUCR5 <sub>4</sub>	PUCR5 <sub>3</sub>	PUCR5 <sub>2</sub>	PUCR5 <sub>1</sub>	PUCR5 <sub>0</sub>		
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%; padding: 2px;">PUCR6 ポートブルアップコントロールレジスタ6</td> <td style="width: 20%; padding: 2px;">H'9F</td> </tr> </table>	PUCR6 ポートブルアップコントロールレジスタ6	H'9F	I/Oポート						
PUCR6 ポートブルアップコントロールレジスタ6	H'9F								
<p>ビット：    7        6        5        4        3        2        1        0</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 2px;">PUCR6<sub>7</sub></td> <td style="padding: 2px;">PUCR6<sub>6</sub></td> <td style="padding: 2px;">PUCR6<sub>5</sub></td> <td style="padding: 2px;">PUCR6<sub>4</sub></td> <td style="padding: 2px;">PUCR6<sub>3</sub></td> <td style="padding: 2px;">PUCR6<sub>2</sub></td> <td style="padding: 2px;">PUCR6<sub>1</sub></td> <td style="padding: 2px;">PUCR6<sub>0</sub></td> </tr> </table> <p>初期値：    0        0        0        0        0        0        0        0</p> <p>R/W    :    R/W    R/W    R/W    R/W    R/W    R/W    R/W</p>		PUCR6 <sub>7</sub>	PUCR6 <sub>6</sub>	PUCR6 <sub>5</sub>	PUCR6 <sub>4</sub>	PUCR6 <sub>3</sub>	PUCR6 <sub>2</sub>	PUCR6 <sub>1</sub>	PUCR6 <sub>0</sub>
PUCR6 <sub>7</sub>	PUCR6 <sub>6</sub>	PUCR6 <sub>5</sub>	PUCR6 <sub>4</sub>	PUCR6 <sub>3</sub>	PUCR6 <sub>2</sub>	PUCR6 <sub>1</sub>	PUCR6 <sub>0</sub>		



SCSR1 シリアルコントロールステータスレジスタ1	H'A1	SCI1
----------------------------	------	------

ビット :    7        6        5        4        3        2        1        0

—	SOL	ORER	—	—	—	—	STF
---	-----	------	---	---	---	---	-----

初期値 :    1        0        0        1        1        1        0        0

R/W :        —        R/W    R/(W)\*    —        —        —        R        R/W

スタートフラグ

0	リード時	転送動作は停止
	ライト時	無効
1	リード時	転送動作中
	ライト時	転送動作を開始

オーバランエラーフラグ

0	〔クリア条件〕	"1"をリード後、"0"をライトしたとき
	〔セット条件〕	外部クロック使用時、転送完了後もクロックが入力されたとき
1		

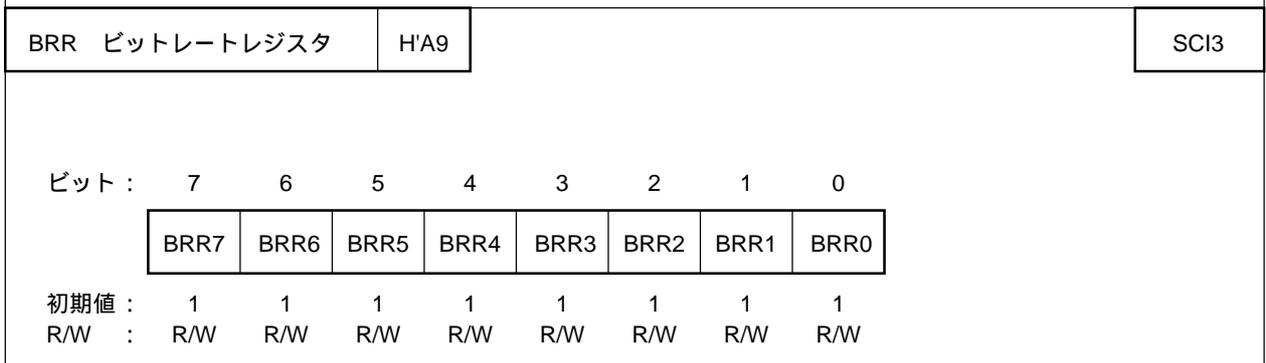
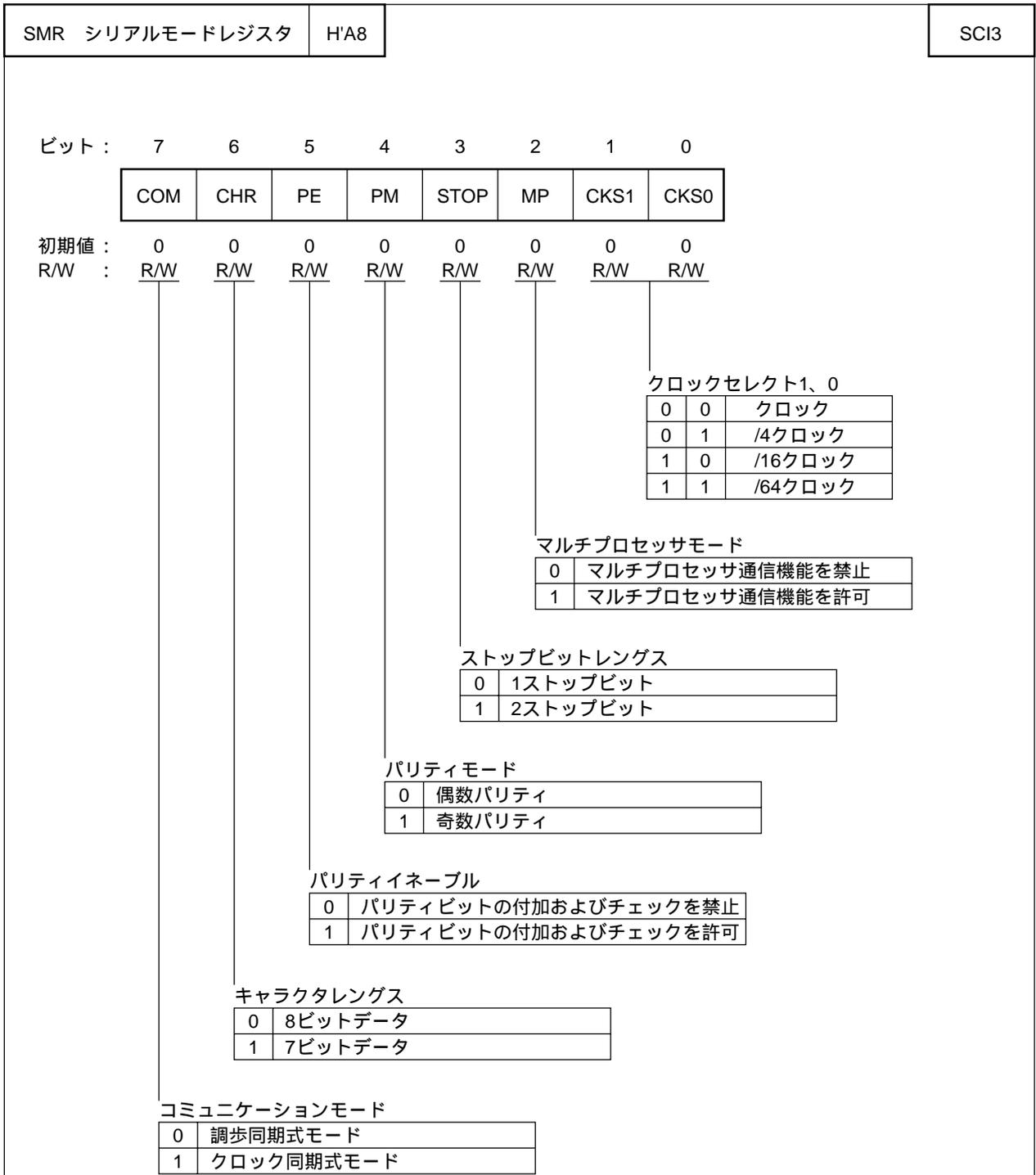
拡張データビット

0	リード時	SO <sub>1</sub> 端子の出力が"Low"レベル
	ライト時	SO <sub>1</sub> 端子の出力を"Low"レベルに変更
1	リード時	SO <sub>1</sub> 端子の出力が"High"レベル
	ライト時	SO <sub>1</sub> 端子の出力を"High"レベルに変更

【注】 \* フラグをクリアするための"0"ライトのみ可能

SDRU シリアルデータレジスタU		H'A2	SCI1								
ビット： 7      6      5      4      3      2      1      0											
<table border="1" style="width: 100%; text-align: center;"> <tr> <td>SDRU7</td><td>SDRU6</td><td>SDRU5</td><td>SDRU4</td><td>SDRU3</td><td>SDRU2</td><td>SDRU1</td><td>SDRU0</td> </tr> </table>				SDRU7	SDRU6	SDRU5	SDRU4	SDRU3	SDRU2	SDRU1	SDRU0
SDRU7	SDRU6	SDRU5	SDRU4	SDRU3	SDRU2	SDRU1	SDRU0				
初期値： 不定    不定    不定    不定    不定    不定    不定    不定											
R/W    :  R/W    R/W    R/W    R/W    R/W    R/W    R/W											
 送信データの設定、受信データの格納に使用 8ビット転送モード   : 未使用 16ビット転送モード : データレジスタ上位8ビット											
SDRL シリアルデータレジスタL		H'A3	SCI1								
ビット： 7      6      5      4      3      2      1      0											
<table border="1" style="width: 100%; text-align: center;"> <tr> <td>SDRL7</td><td>SDRL6</td><td>SDRL5</td><td>SDRL4</td><td>SDRL3</td><td>SDRL2</td><td>SDRL1</td><td>SDRL0</td> </tr> </table>				SDRL7	SDRL6	SDRL5	SDRL4	SDRL3	SDRL2	SDRL1	SDRL0
SDRL7	SDRL6	SDRL5	SDRL4	SDRL3	SDRL2	SDRL1	SDRL0				
初期値： 不定    不定    不定    不定    不定    不定    不定    不定											
R/W    :  R/W    R/W    R/W    R/W    R/W    R/W    R/W											
 送信データの設定、受信データの格納に使用 8ビット転送モード   : データレジスタ 16ビット転送モード : データレジスタ下位8ビット											



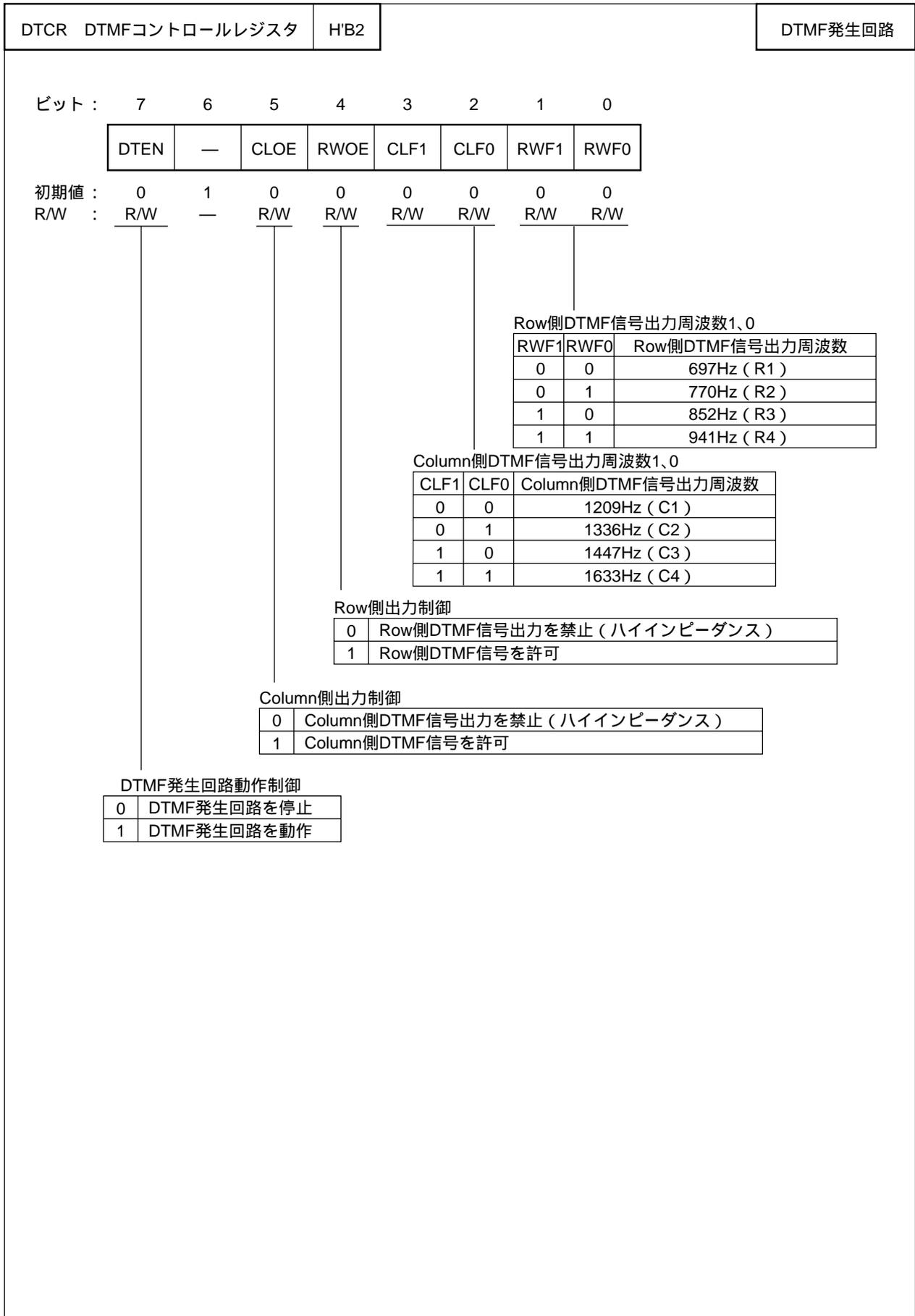
SCR3 シリアルコントロールレジスタ3	H'AA	SCI3																																																																																		
<p>ビット：    7        6        5        4        3        2        1        0</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 2px 10px;">TIE</td> <td style="padding: 2px 10px;">RIE</td> <td style="padding: 2px 10px;">TE</td> <td style="padding: 2px 10px;">RE</td> <td style="padding: 2px 10px;">MPIE</td> <td style="padding: 2px 10px;">TEIE</td> <td style="padding: 2px 10px;">CKE1</td> <td style="padding: 2px 10px;">CKE0</td> </tr> </table> <p>初期値：    0        0        0        0        0        0        0        0</p> <p>R/W    :    R/W    R/W    R/W    R/W    R/W    R/W    R/W</p> <div style="margin-top: 20px;"> <p style="text-align: center;">クロックイネーブル</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse; text-align: center;"> <thead> <tr> <th style="padding: 2px 10px;">ビット1</th> <th style="padding: 2px 10px;">ビット0</th> <th colspan="3" style="padding: 2px 10px;">説 明</th> </tr> <tr> <th style="padding: 2px 10px;">CKE1</th> <th style="padding: 2px 10px;">CKE0</th> <th style="padding: 2px 10px;">コミュニケーションモード</th> <th style="padding: 2px 10px;">クロックソース</th> <th style="padding: 2px 10px;">SCK<sub>3</sub>端子機能</th> </tr> </thead> <tbody> <tr> <td style="padding: 2px 10px;">0</td> <td style="padding: 2px 10px;">0</td> <td style="padding: 2px 10px;">調歩同期式</td> <td style="padding: 2px 10px;">内部クロック</td> <td style="padding: 2px 10px;">入出力ポート</td> </tr> <tr> <td style="padding: 2px 10px;"></td> <td style="padding: 2px 10px;"></td> <td style="padding: 2px 10px;">クロック同期式</td> <td style="padding: 2px 10px;">内部クロック</td> <td style="padding: 2px 10px;">同期クロック出力</td> </tr> <tr> <td style="padding: 2px 10px;">0</td> <td style="padding: 2px 10px;">1</td> <td style="padding: 2px 10px;">調歩同期式</td> <td style="padding: 2px 10px;">内部クロック</td> <td style="padding: 2px 10px;">クロック出力</td> </tr> <tr> <td style="padding: 2px 10px;"></td> <td style="padding: 2px 10px;"></td> <td style="padding: 2px 10px;">クロック同期式</td> <td style="padding: 2px 10px;">リザーブ</td> <td style="padding: 2px 10px;"></td> </tr> <tr> <td style="padding: 2px 10px;">1</td> <td style="padding: 2px 10px;">0</td> <td style="padding: 2px 10px;">調歩同期式</td> <td style="padding: 2px 10px;">外部クロック</td> <td style="padding: 2px 10px;">クロック出力</td> </tr> <tr> <td style="padding: 2px 10px;"></td> <td style="padding: 2px 10px;"></td> <td style="padding: 2px 10px;">クロック同期式</td> <td style="padding: 2px 10px;">外部クロック</td> <td style="padding: 2px 10px;">同期クロック入力</td> </tr> <tr> <td style="padding: 2px 10px;">1</td> <td style="padding: 2px 10px;">1</td> <td style="padding: 2px 10px;">調歩同期式</td> <td style="padding: 2px 10px;">リザーブ</td> <td style="padding: 2px 10px;"></td> </tr> <tr> <td style="padding: 2px 10px;"></td> <td style="padding: 2px 10px;"></td> <td style="padding: 2px 10px;">クロック同期式</td> <td style="padding: 2px 10px;">リザーブ</td> <td style="padding: 2px 10px;"></td> </tr> </tbody> </table> <div style="margin-top: 10px;"> <p>トランスミットエンドインタラプトイネーブル</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 2px 10px;">0</td> <td style="padding: 2px 10px;">送信終了割込み要求 (TEI) を禁止</td> </tr> <tr> <td style="padding: 2px 10px;">1</td> <td style="padding: 2px 10px;">送信終了割込み要求 (TEI) を許可</td> </tr> </table> <p>マルチプロセッサインタラプトイネーブル</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 2px 10px;">0</td> <td style="padding: 2px 10px;">マルチプロセッサ割込み要求を禁止 (通常の受信動作) 〔クリア条件〕 マルチプロセッサビットが"1"のデータを受信したとき</td> </tr> <tr> <td style="padding: 2px 10px;">1</td> <td style="padding: 2px 10px;">マルチプロセッサ割込み要求を許可 マルチプロセッサビットが"1"のデータを受け取るまで受信割込み 要求 (RXI)、受信エラー割込み要求 (ERI)、および、シリアル ステータスレジスタ (SSR) のRDRF、FER、OERの各フラグの セットを禁止</td> </tr> </table> <p>レーシブイネーブル</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 2px 10px;">0</td> <td style="padding: 2px 10px;">受信動作を禁止 (RXD端子は入出力ポート)</td> </tr> <tr> <td style="padding: 2px 10px;">1</td> <td style="padding: 2px 10px;">受信動作を許可 (RXD端子はレーシブデータ端子)</td> </tr> </table> <p>トランスミットイネーブル</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 2px 10px;">0</td> <td style="padding: 2px 10px;">送信動作を禁止 (TXD端子はトランスミットデータ端子)</td> </tr> <tr> <td style="padding: 2px 10px;">1</td> <td style="padding: 2px 10px;">送信動作を許可 (TXD端子はトランスミットデータ端子)</td> </tr> </table> <p>レーシブインタラプトイネーブル</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 2px 10px;">0</td> <td style="padding: 2px 10px;">受信データフル割込み要求 (RXI)、および受信エラー割込み要求 (ERI) を禁止</td> </tr> <tr> <td style="padding: 2px 10px;">1</td> <td style="padding: 2px 10px;">受信データフル割込み要求 (RXI)、および受信エラー割込み要求 (ERI) を許可</td> </tr> </table> <p>トランスミットインタラプトイネーブル</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 2px 10px;">0</td> <td style="padding: 2px 10px;">送信データエンpty割込み要求 (TXI) の禁止</td> </tr> <tr> <td style="padding: 2px 10px;">1</td> <td style="padding: 2px 10px;">送信データエンpty割込み要求 (TXI) の許可</td> </tr> </table> </div> </div>			TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	ビット1	ビット0	説 明			CKE1	CKE0	コミュニケーションモード	クロックソース	SCK <sub>3</sub> 端子機能	0	0	調歩同期式	内部クロック	入出力ポート			クロック同期式	内部クロック	同期クロック出力	0	1	調歩同期式	内部クロック	クロック出力			クロック同期式	リザーブ		1	0	調歩同期式	外部クロック	クロック出力			クロック同期式	外部クロック	同期クロック入力	1	1	調歩同期式	リザーブ				クロック同期式	リザーブ		0	送信終了割込み要求 (TEI) を禁止	1	送信終了割込み要求 (TEI) を許可	0	マルチプロセッサ割込み要求を禁止 (通常の受信動作) 〔クリア条件〕 マルチプロセッサビットが"1"のデータを受信したとき	1	マルチプロセッサ割込み要求を許可 マルチプロセッサビットが"1"のデータを受け取るまで受信割込み 要求 (RXI)、受信エラー割込み要求 (ERI)、および、シリアル ステータスレジスタ (SSR) のRDRF、FER、OERの各フラグの セットを禁止	0	受信動作を禁止 (RXD端子は入出力ポート)	1	受信動作を許可 (RXD端子はレーシブデータ端子)	0	送信動作を禁止 (TXD端子はトランスミットデータ端子)	1	送信動作を許可 (TXD端子はトランスミットデータ端子)	0	受信データフル割込み要求 (RXI)、および受信エラー割込み要求 (ERI) を禁止	1	受信データフル割込み要求 (RXI)、および受信エラー割込み要求 (ERI) を許可	0	送信データエンpty割込み要求 (TXI) の禁止	1	送信データエンpty割込み要求 (TXI) の許可
TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0																																																																													
ビット1	ビット0	説 明																																																																																		
CKE1	CKE0	コミュニケーションモード	クロックソース	SCK <sub>3</sub> 端子機能																																																																																
0	0	調歩同期式	内部クロック	入出力ポート																																																																																
		クロック同期式	内部クロック	同期クロック出力																																																																																
0	1	調歩同期式	内部クロック	クロック出力																																																																																
		クロック同期式	リザーブ																																																																																	
1	0	調歩同期式	外部クロック	クロック出力																																																																																
		クロック同期式	外部クロック	同期クロック入力																																																																																
1	1	調歩同期式	リザーブ																																																																																	
		クロック同期式	リザーブ																																																																																	
0	送信終了割込み要求 (TEI) を禁止																																																																																			
1	送信終了割込み要求 (TEI) を許可																																																																																			
0	マルチプロセッサ割込み要求を禁止 (通常の受信動作) 〔クリア条件〕 マルチプロセッサビットが"1"のデータを受信したとき																																																																																			
1	マルチプロセッサ割込み要求を許可 マルチプロセッサビットが"1"のデータを受け取るまで受信割込み 要求 (RXI)、受信エラー割込み要求 (ERI)、および、シリアル ステータスレジスタ (SSR) のRDRF、FER、OERの各フラグの セットを禁止																																																																																			
0	受信動作を禁止 (RXD端子は入出力ポート)																																																																																			
1	受信動作を許可 (RXD端子はレーシブデータ端子)																																																																																			
0	送信動作を禁止 (TXD端子はトランスミットデータ端子)																																																																																			
1	送信動作を許可 (TXD端子はトランスミットデータ端子)																																																																																			
0	受信データフル割込み要求 (RXI)、および受信エラー割込み要求 (ERI) を禁止																																																																																			
1	受信データフル割込み要求 (RXI)、および受信エラー割込み要求 (ERI) を許可																																																																																			
0	送信データエンpty割込み要求 (TXI) の禁止																																																																																			
1	送信データエンpty割込み要求 (TXI) の許可																																																																																			

TDR トランスミットデータレジスタ	H'AB		SCI3								
<p>ビット :    7        6        5        4        3        2        1        0</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 2px 10px;">TDR7</td> <td style="padding: 2px 10px;">TDR6</td> <td style="padding: 2px 10px;">TDR5</td> <td style="padding: 2px 10px;">TDR4</td> <td style="padding: 2px 10px;">TDR3</td> <td style="padding: 2px 10px;">TDR2</td> <td style="padding: 2px 10px;">TDR1</td> <td style="padding: 2px 10px;">TDR0</td> </tr> </table> <p>初期値 :    1        1        1        1        1        1        1        1</p> <p>R/W :    R/W    R/W    R/W    R/W    R/W    R/W    R/W</p> <div style="margin-left: 100px; margin-top: 10px;"> <p style="margin-left: 100px;">TSRへの転送用データ</p> </div>				TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0
TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0				

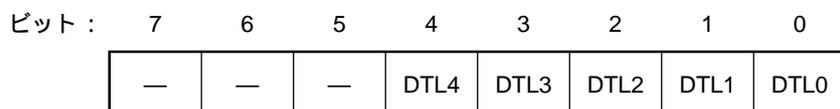
SSR シリアルステータスレジスタ		H'AC						SCI3	
ビット : 7 6 5 4 3 2 1 0									
TDRE		RDRF		OER		FER		PER	
TDRE		RDRF		OER		FER		PER	
初期値 : 1 0 0 0 0 1 0 0		R/W : R/(W)* R/(W)* R/(W)* R/(W)* R/(W)* R R R/W							
								マルチプロセッサビットトランスファ	
								0 マルチプロセッサビット"0"を送信	
								1 マルチプロセッサビット"1"を送信	
								マルチプロセッサビットレシーブ	
								0 マルチプロセッサビットが"0"のデータを受信	
								1 マルチプロセッサビットが"1"のデータを受信	
トランスミットエンド									
		0		送信中 〔クリア条件〕 (1) TDRE = "1"の状態をリードした後、TDREに"0"をライトしたとき (2) 命令でTDRにデータをライトしたとき					
		1		送信終了 〔セット条件〕 (1) シリアルコントロールレジスタ3 (SCR3) のTEが"0"のとき (2) 送信キャラクタ最後尾のビットの送信時に、TDREが"1"であったとき					
パリティエラー									
		0		受信中、または受信完了 〔クリア条件〕 PER = "1"の状態をリードした後、"0"をライトしたとき					
		1		受信時にパリティエラー発生 〔セット条件〕 受信時に受信データとパリティビットをあわせた"1"の数がシリアルレジスタ (SMR) のパリティモード (PM) で設定したパリティと一致しなかったとき					
フレーミングエラー									
		0		受信中、または受信完了 〔クリア条件〕 FER = "1"の状態をリードした後、"0"をライトしたとき					
		1		受信時にフレーミングエラー発生 〔セット条件〕 受信終了時に受信データの最後尾のストップビットが"1"であるかどうかをチェックし、ストップビットが"0"であったとき					
オーバランエラー									
		0		受信中、または受信完了 〔クリア条件〕 OER = "1"の状態をリードした後、"0"をライトしたとき					
		1		受信時にオーバランエラー発生 〔セット条件〕 RDRFが"1"の状態での受信を完了したとき					
レシーブデータレジスタフル									
		0		RDRに受信データ未格納 〔クリア条件〕 (1) RDRF = "1"の状態をリードした後、"0"をライトしたとき (2) 命令でRDRのデータをリードしたとき					
		1		RDRに受信データが格納されている 〔セット条件〕 受信が正常終了し、RSRからRDRへ受信データが転送されたとき					
トランスミットデータレジスタエンpty									
		0		TDRにライトされた送信データがTSRに転送されていない 〔クリア条件〕 (1) TDRE = "1"の状態をリードした後、"0"をライトしたとき (2) 命令でTDRへデータをライトしたとき					
		1		TDRに送信データがライトされていない、またはTDRにライトされた送信データがTSRに転送された 〔セット条件〕 (1) シリアルコントロールレジスタ3 (SCR3) のTEが"0"のとき (2) TDRからTSRにデータ転送が行われたとき					
【注】 * フラグをクリアするための"0"ライトのみ可能									

RDR レシーブデータレジスタ	H'AD		SCI3																																																			
<p>ビット：    7        6        5        4        3        2        1        0</p> <table border="1" style="margin: 10px auto; border-collapse: collapse;"> <tr> <td style="padding: 2px 5px;">RDR7</td> <td style="padding: 2px 5px;">RDR6</td> <td style="padding: 2px 5px;">RDR5</td> <td style="padding: 2px 5px;">RDR4</td> <td style="padding: 2px 5px;">RDR3</td> <td style="padding: 2px 5px;">RDR2</td> <td style="padding: 2px 5px;">RDR1</td> <td style="padding: 2px 5px;">RDR0</td> </tr> </table> <p>初期値：    0        0        0        0        0        0        0        0</p> <p>R/W    :    R        R        R        R        R        R        R        R</p>				RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1	RDR0																																											
RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1	RDR0																																															
TMA タイマモードレジスタA	H'B0		タイマA																																																			
<p>ビット：    7        6        5        4        3        2        1        0</p> <table border="1" style="margin: 10px auto; border-collapse: collapse;"> <tr> <td style="padding: 2px 5px;">TMA7</td> <td style="padding: 2px 5px;">TMA6</td> <td style="padding: 2px 5px;">TMA5</td> <td style="padding: 2px 5px;">—</td> <td style="padding: 2px 5px;">TMA3</td> <td style="padding: 2px 5px;">TMA2</td> <td style="padding: 2px 5px;">TMA1</td> <td style="padding: 2px 5px;">TMA0</td> </tr> </table> <p>初期値：    0        0        0        1        0        0        0        0</p> <p>R/W    :    R/W    R/W    R/W    —        R/W    R/W    R/W    R/W</p>				TMA7	TMA6	TMA5	—	TMA3	TMA2	TMA1	TMA0																																											
TMA7	TMA6	TMA5	—	TMA3	TMA2	TMA1	TMA0																																															
<p>内部クロックセレクト</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">TMA3</th> <th style="width: 10%;">TMA2</th> <th style="width: 10%;">TMA1</th> <th style="width: 10%;">TMA0</th> <th style="width: 40%;">プリスケラ分周比またはオーバフロー周期</th> <th style="width: 20%;">機 能</th> </tr> </thead> <tbody> <tr> <td rowspan="8" style="text-align: center; vertical-align: middle;">0</td> <td rowspan="4" style="text-align: center; vertical-align: middle;">0</td> <td rowspan="2" style="text-align: center; vertical-align: middle;">0</td> <td style="text-align: center;">0</td> <td>PSS /8192</td> <td rowspan="8" style="text-align: center; vertical-align: middle;">インターバル</td> </tr> <tr> <td style="text-align: center;">1</td> <td>PSS /4096</td> </tr> <tr> <td rowspan="2" style="text-align: center; vertical-align: middle;">1</td> <td style="text-align: center;">0</td> <td>PSS /2048</td> </tr> <tr> <td style="text-align: center;">1</td> <td>PSS /512</td> </tr> <tr> <td rowspan="4" style="text-align: center; vertical-align: middle;">1</td> <td rowspan="2" style="text-align: center; vertical-align: middle;">0</td> <td style="text-align: center;">0</td> <td>PSS /256</td> </tr> <tr> <td style="text-align: center;">1</td> <td>PSS /128</td> </tr> <tr> <td rowspan="2" style="text-align: center; vertical-align: middle;">1</td> <td style="text-align: center;">0</td> <td>PSS /32</td> </tr> <tr> <td style="text-align: center;">1</td> <td>PSS /8</td> </tr> <tr> <td rowspan="8" style="text-align: center; vertical-align: middle;">1</td> <td rowspan="4" style="text-align: center; vertical-align: middle;">0</td> <td rowspan="2" style="text-align: center; vertical-align: middle;">0</td> <td style="text-align: center;">0</td> <td>PSW 1s</td> <td rowspan="8" style="text-align: center; vertical-align: middle;">時計用タイムベース</td> </tr> <tr> <td style="text-align: center;">1</td> <td>PSW 0.5s</td> </tr> <tr> <td rowspan="2" style="text-align: center; vertical-align: middle;">1</td> <td style="text-align: center;">0</td> <td>PSW 0.25s</td> </tr> <tr> <td style="text-align: center;">1</td> <td>PSW 0.03125s</td> </tr> <tr> <td rowspan="4" style="text-align: center; vertical-align: middle;">1</td> <td rowspan="2" style="text-align: center; vertical-align: middle;">0</td> <td style="text-align: center;">0</td> <td rowspan="4" style="text-align: center; vertical-align: middle;">PSW、TCAリセット</td> </tr> <tr> <td style="text-align: center;">1</td> </tr> <tr> <td rowspan="2" style="text-align: center; vertical-align: middle;">1</td> <td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">1</td> </tr> </tbody> </table>				TMA3	TMA2	TMA1	TMA0	プリスケラ分周比またはオーバフロー周期	機 能	0	0	0	0	PSS /8192	インターバル	1	PSS /4096	1	0	PSS /2048	1	PSS /512	1	0	0	PSS /256	1	PSS /128	1	0	PSS /32	1	PSS /8	1	0	0	0	PSW 1s	時計用タイムベース	1	PSW 0.5s	1	0	PSW 0.25s	1	PSW 0.03125s	1	0	0	PSW、TCAリセット	1	1	0	1
TMA3	TMA2	TMA1	TMA0	プリスケラ分周比またはオーバフロー周期	機 能																																																	
0	0	0	0	PSS /8192	インターバル																																																	
			1	PSS /4096																																																		
		1	0	PSS /2048																																																		
			1	PSS /512																																																		
	1	0	0	PSS /256																																																		
			1	PSS /128																																																		
		1	0	PSS /32																																																		
			1	PSS /8																																																		
1	0	0	0	PSW 1s	時計用タイムベース																																																	
			1	PSW 0.5s																																																		
		1	0	PSW 0.25s																																																		
			1	PSW 0.03125s																																																		
	1	0	0	PSW、TCAリセット																																																		
			1																																																			
		1	0																																																			
			1																																																			
<p>クロック出力セレクト</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tbody> <tr> <td rowspan="4" style="text-align: center; vertical-align: middle;">0</td> <td rowspan="2" style="text-align: center; vertical-align: middle;">0</td> <td style="text-align: center;">0</td> <td>/32</td> </tr> <tr> <td style="text-align: center;">1</td> <td>/16</td> </tr> <tr> <td rowspan="2" style="text-align: center; vertical-align: middle;">1</td> <td style="text-align: center;">0</td> <td>/8</td> </tr> <tr> <td style="text-align: center;">1</td> <td>/4</td> </tr> <tr> <td rowspan="4" style="text-align: center; vertical-align: middle;">1</td> <td rowspan="2" style="text-align: center; vertical-align: middle;">0</td> <td style="text-align: center;">0</td> <td>w/32</td> </tr> <tr> <td style="text-align: center;">1</td> <td>w/16</td> </tr> <tr> <td rowspan="2" style="text-align: center; vertical-align: middle;">1</td> <td style="text-align: center;">0</td> <td>w/8</td> </tr> <tr> <td style="text-align: center;">1</td> <td>w/4</td> </tr> </tbody> </table>				0	0	0	/32	1	/16	1	0	/8	1	/4	1	0	0	w/32	1	w/16	1	0	w/8	1	w/4																													
0	0	0	/32																																																			
		1	/16																																																			
	1	0	/8																																																			
		1	/4																																																			
1	0	0	w/32																																																			
		1	w/16																																																			
	1	0	w/8																																																			
		1	w/4																																																			

TCA タイマカウンタA	H'B1		タイマA								
<p>ビット：    7        6        5        4        3        2        1        0</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr> <td style="padding: 2px 5px;">TCA7</td> <td style="padding: 2px 5px;">TCA6</td> <td style="padding: 2px 5px;">TCA5</td> <td style="padding: 2px 5px;">TCA4</td> <td style="padding: 2px 5px;">TCA3</td> <td style="padding: 2px 5px;">TCA2</td> <td style="padding: 2px 5px;">TCA1</td> <td style="padding: 2px 5px;">TCA0</td> </tr> </table> <p>初期値：    0        0        0        0        0        0        0        0</p> <p>R/W    :    R        R        R        R        R        R        R        R</p> <p style="text-align: center; margin-top: 10px;"> <span style="border-top: 1px solid black; display: inline-block; width: 100px; margin: 0 auto;"></span>              カウント値         </p>				TCA7	TCA6	TCA5	TCA4	TCA3	TCA2	TCA1	TCA0
TCA7	TCA6	TCA5	TCA4	TCA3	TCA2	TCA1	TCA0				



DTLR DTMFロードレジスタ	H'B3	DTMF発生回路
------------------	------	----------

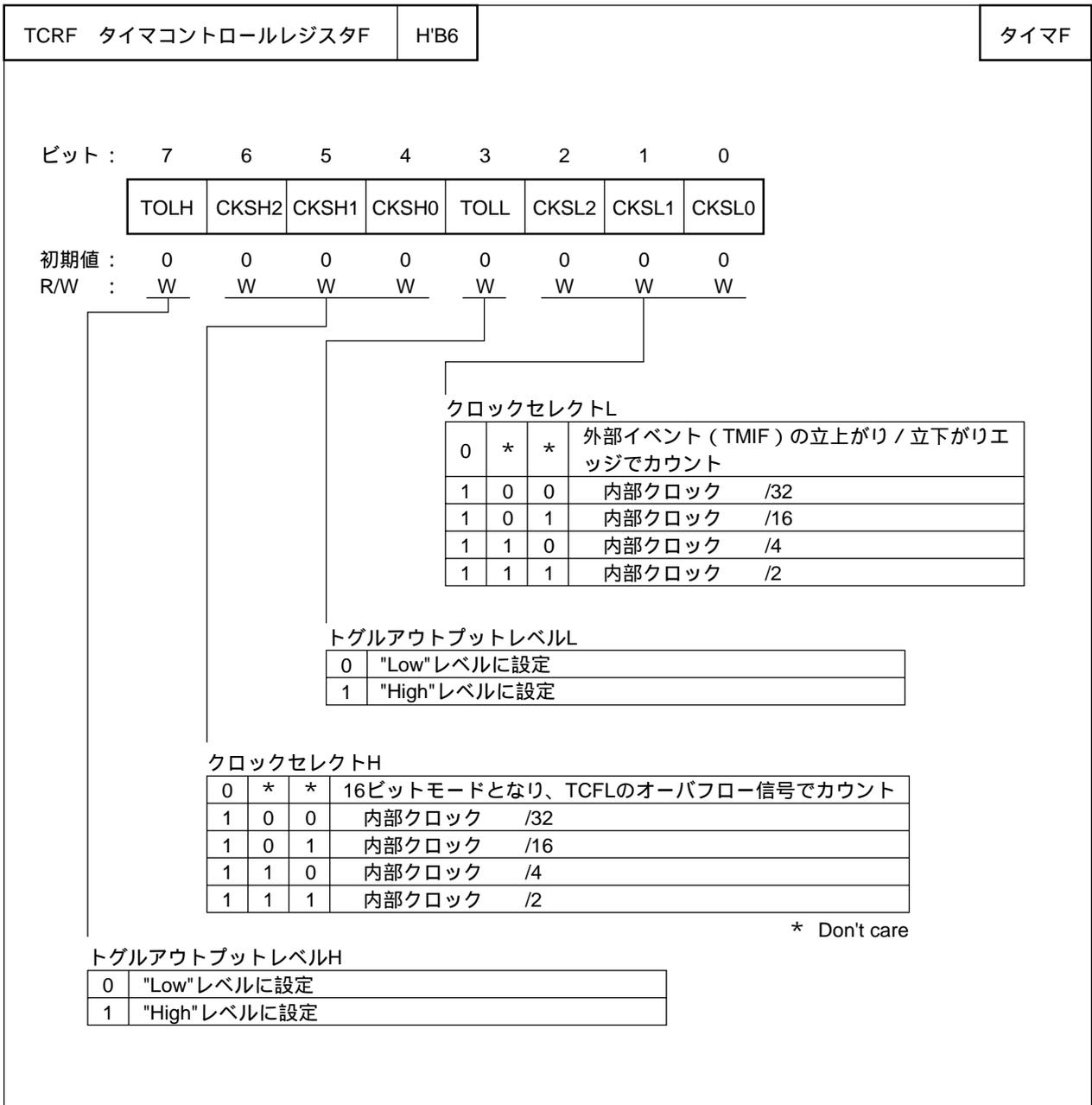


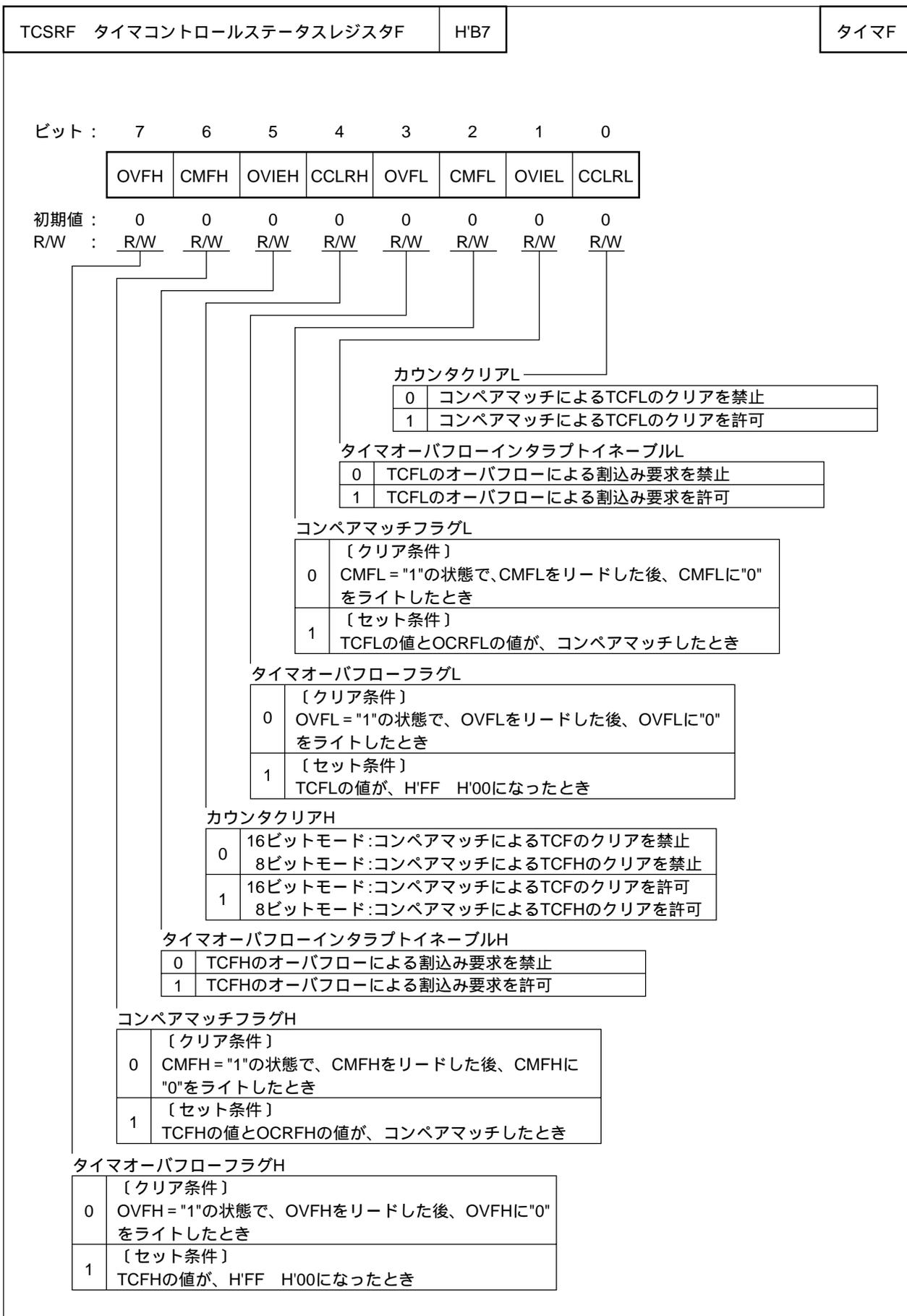
初期値 : 1      1      1      0      0      0      0      0  
R/W : —      —      —      R/W      R/W      R/W      R/W      R/W

OSCクロック分周比4~0

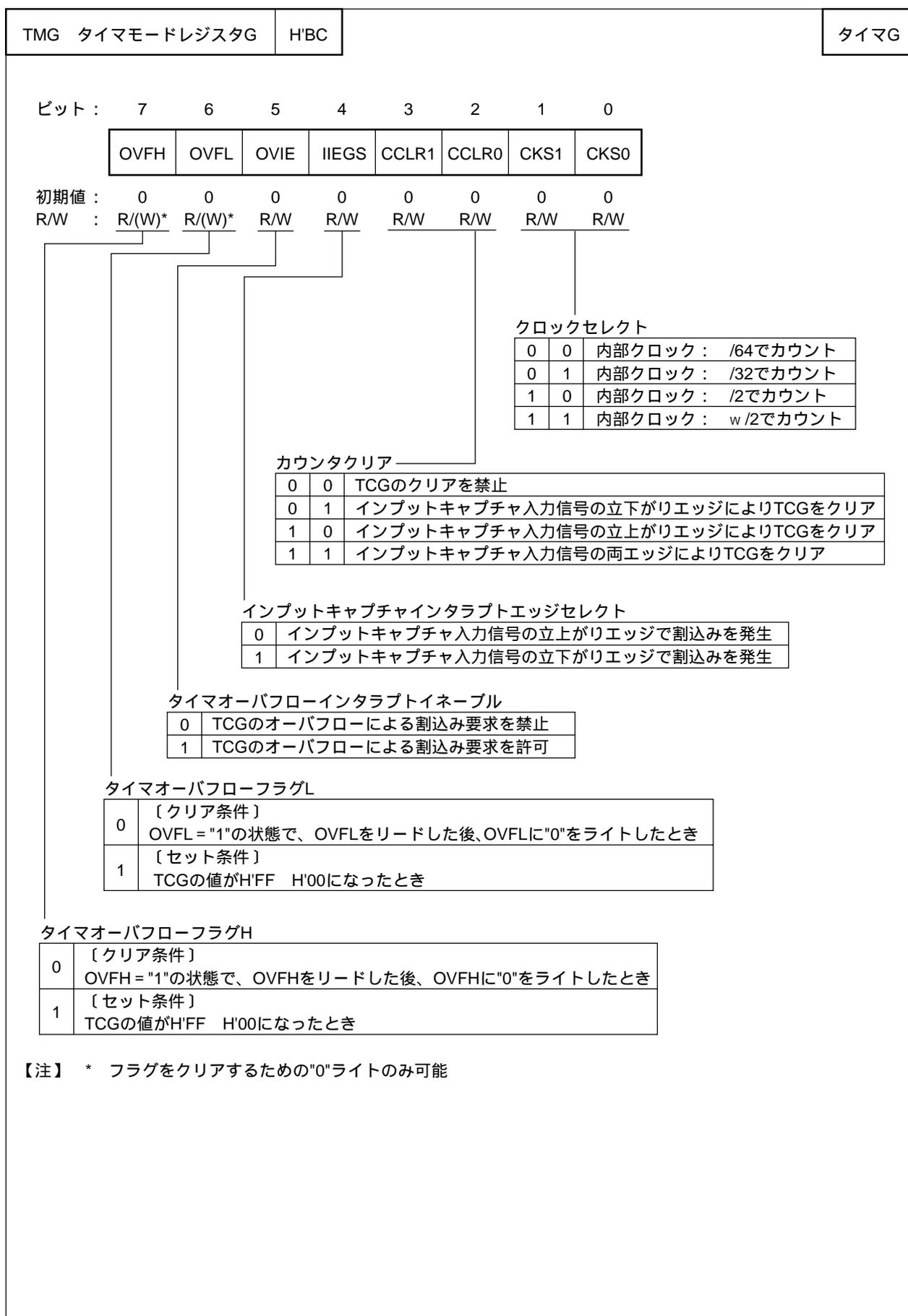
DTL4	DTL3	DTL2	DTL1	DTL0	分周比	OSCクロック周波数
0	0	0	0	0	設定禁止	(初期値)
0	0	0	0	1	設定禁止	
0	0	0	1	0	設定禁止	
0	0	0	1	1	3	1.2MHz
0	0	1	0	0	4	1.6MHz
⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	1	0	0	1	25	10MHz
1	1	0	1	*	設定禁止	
1	1	1	*	*	設定禁止	

\* : Don't care





TCFH 8ビットタイマカウンタFH	H'B8		タイムF								
<p>ビット :    7        6        5        4        3        2        1        0</p> <table style="margin-left: 40px; border-collapse: collapse;"> <tr> <td style="border: 1px solid black; padding: 2px 10px;">TCFH7</td> <td style="border: 1px solid black; padding: 2px 10px;">TCFH6</td> <td style="border: 1px solid black; padding: 2px 10px;">TCFH5</td> <td style="border: 1px solid black; padding: 2px 10px;">TCFH4</td> <td style="border: 1px solid black; padding: 2px 10px;">TCFH3</td> <td style="border: 1px solid black; padding: 2px 10px;">TCFH2</td> <td style="border: 1px solid black; padding: 2px 10px;">TCFH1</td> <td style="border: 1px solid black; padding: 2px 10px;">TCFH0</td> </tr> </table> <p>初期値 :    0        0        0        0        0        0        0        0</p> <p>R/W :    <u>R/W</u>    <u>R/W</u>    <u>R/W</u>    <u>R/W</u>    <u>R/W</u>    <u>R/W</u>    <u>R/W</u>    <u>R/W</u></p> <p style="text-align: center; margin-left: 40px;">  カウント値</p>				TCFH7	TCFH6	TCFH5	TCFH4	TCFH3	TCFH2	TCFH1	TCFH0
TCFH7	TCFH6	TCFH5	TCFH4	TCFH3	TCFH2	TCFH1	TCFH0				
TCFL 8ビットタイマカウンタFL	H'B9		タイムF								
<p>ビット :    7        6        5        4        3        2        1        0</p> <table style="margin-left: 40px; border-collapse: collapse;"> <tr> <td style="border: 1px solid black; padding: 2px 10px;">TCFL7</td> <td style="border: 1px solid black; padding: 2px 10px;">TCFL6</td> <td style="border: 1px solid black; padding: 2px 10px;">TCFL5</td> <td style="border: 1px solid black; padding: 2px 10px;">TCFL4</td> <td style="border: 1px solid black; padding: 2px 10px;">TCFL3</td> <td style="border: 1px solid black; padding: 2px 10px;">TCFL2</td> <td style="border: 1px solid black; padding: 2px 10px;">TCFL1</td> <td style="border: 1px solid black; padding: 2px 10px;">TCFL0</td> </tr> </table> <p>初期値 :    0        0        0        0        0        0        0        0</p> <p>R/W :    <u>R/W</u>    <u>R/W</u>    <u>R/W</u>    <u>R/W</u>    <u>R/W</u>    <u>R/W</u>    <u>R/W</u>    <u>R/W</u></p> <p style="text-align: center; margin-left: 40px;">  カウント値</p>				TCFL7	TCFL6	TCFL5	TCFL4	TCFL3	TCFL2	TCFL1	TCFL0
TCFL7	TCFL6	TCFL5	TCFL4	TCFL3	TCFL2	TCFL1	TCFL0				
OCRFH 8ビットアウトプットコンペアレジスタFH	H'BA		タイムF								
<p>ビット :    7        6        5        4        3        2        1        0</p> <table style="margin-left: 40px; border-collapse: collapse;"> <tr> <td style="border: 1px solid black; padding: 2px 10px;">OCRFH7</td> <td style="border: 1px solid black; padding: 2px 10px;">OCRFH6</td> <td style="border: 1px solid black; padding: 2px 10px;">OCRFH5</td> <td style="border: 1px solid black; padding: 2px 10px;">OCRFH4</td> <td style="border: 1px solid black; padding: 2px 10px;">OCRFH3</td> <td style="border: 1px solid black; padding: 2px 10px;">OCRFH2</td> <td style="border: 1px solid black; padding: 2px 10px;">OCRFH1</td> <td style="border: 1px solid black; padding: 2px 10px;">OCRFH0</td> </tr> </table> <p>初期値 :    1        1        1        1        1        1        1        1</p> <p>R/W :    <u>R/W</u>    <u>R/W</u>    <u>R/W</u>    <u>R/W</u>    <u>R/W</u>    <u>R/W</u>    <u>R/W</u>    <u>R/W</u></p>				OCRFH7	OCRFH6	OCRFH5	OCRFH4	OCRFH3	OCRFH2	OCRFH1	OCRFH0
OCRFH7	OCRFH6	OCRFH5	OCRFH4	OCRFH3	OCRFH2	OCRFH1	OCRFH0				
OCRFL 8ビットアウトプットコンペアレジスタFL	H'BB		タイムF								
<p>ビット :    7        6        5        4        3        2        1        0</p> <table style="margin-left: 40px; border-collapse: collapse;"> <tr> <td style="border: 1px solid black; padding: 2px 10px;">OCRFL7</td> <td style="border: 1px solid black; padding: 2px 10px;">OCRFL6</td> <td style="border: 1px solid black; padding: 2px 10px;">OCRFL5</td> <td style="border: 1px solid black; padding: 2px 10px;">OCRFL4</td> <td style="border: 1px solid black; padding: 2px 10px;">OCRFL3</td> <td style="border: 1px solid black; padding: 2px 10px;">OCRFL2</td> <td style="border: 1px solid black; padding: 2px 10px;">OCRFL1</td> <td style="border: 1px solid black; padding: 2px 10px;">OCRFL0</td> </tr> </table> <p>初期値 :    1        1        1        1        1        1        1        1</p> <p>R/W :    <u>R/W</u>    <u>R/W</u>    <u>R/W</u>    <u>R/W</u>    <u>R/W</u>    <u>R/W</u>    <u>R/W</u>    <u>R/W</u></p>				OCRFL7	OCRFL6	OCRFL5	OCRFL4	OCRFL3	OCRFL2	OCRFL1	OCRFL0
OCRFL7	OCRFL6	OCRFL5	OCRFL4	OCRFL3	OCRFL2	OCRFL1	OCRFL0				



ICRGF インพุットキャプチャレジスタGF	H'BD	タイムG								
<p>ビット :    7        6        5        4        3        2        1        0</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr> <td style="padding: 2px 10px;">ICRGF7</td> <td style="padding: 2px 10px;">ICRGF6</td> <td style="padding: 2px 10px;">ICRGF5</td> <td style="padding: 2px 10px;">ICRGF4</td> <td style="padding: 2px 10px;">ICRGF3</td> <td style="padding: 2px 10px;">ICRGF2</td> <td style="padding: 2px 10px;">ICRGF1</td> <td style="padding: 2px 10px;">ICRGF0</td> </tr> </table> <p>初期値 :    0        0        0        0        0        0        0        0</p> <p>R/W    :    R        R        R        R        R        R        R        R</p>			ICRGF7	ICRGF6	ICRGF5	ICRGF4	ICRGF3	ICRGF2	ICRGF1	ICRGF0
ICRGF7	ICRGF6	ICRGF5	ICRGF4	ICRGF3	ICRGF2	ICRGF1	ICRGF0			

ICRGR インพุットキャプチャレジスタGR	H'BE	タイムG								
<p>ビット :    7        6        5        4        3        2        1        0</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr> <td style="padding: 2px 10px;">ICRGR7</td> <td style="padding: 2px 10px;">ICRGR6</td> <td style="padding: 2px 10px;">ICRGR5</td> <td style="padding: 2px 10px;">ICRGR4</td> <td style="padding: 2px 10px;">ICRGR3</td> <td style="padding: 2px 10px;">ICRGR2</td> <td style="padding: 2px 10px;">ICRGR1</td> <td style="padding: 2px 10px;">ICRGR0</td> </tr> </table> <p>初期値 :    0        0        0        0        0        0        0        0</p> <p>R/W    :    R        R        R        R        R        R        R        R</p>			ICRGR7	ICRGR6	ICRGR5	ICRGR4	ICRGR3	ICRGR2	ICRGR1	ICRGR0
ICRGR7	ICRGR6	ICRGR5	ICRGR4	ICRGR3	ICRGR2	ICRGR1	ICRGR0			

AMR A/Dモードレジスタ			H'C4		A/D変換器			
ビット:	7	6	5	4	3	2	1	0
	CKS	TRGE	CKS1	—	CH3	CH2	CH1	CH0
初期値:	0	0	0	1	0	0	0	0
R/W:	R/W	R/W	R/W	—	R/W	R/W	R/W	R/W
				チャンネルセレクト				
				ビット3	ビット2	ビット1	ビット0	アナログ入力チャンネル
				CH3	CH2	CH1	CH0	
				0	0	*	*	非選択
				0	1	*	*	リザーブ
				1	0	0	*	リザーブ
				1	0	1	0	AN <sub>6</sub>
				1	0	1	1	AN <sub>7</sub>
				1	1	*	*	リザーブ
				* Don't care				
				外部トリガセレクト				
				0	外部トリガによるA/D変換の開始を禁止			
				1	外部トリガ (ADTRG) 端子の立上がりエッジ、または立下がりエッジでA/D変換を開始			
				クロックセレクト				
CKS	CKS1	変換周期	変換時間					
			= 2MHz	= 5MHz				
0	0	リザーブ	—	—				
0	1	124/	62μs	24.8μs				
1	0	62/	31μs	12.4μs				
1	1	31/	15.5μs	—	*			

【注】 \* 12.4μs以下の変換時間では、動作が保証されません。12.4μs以上になるように選択してください。

ADRR A/Dリザルトレジスタ	H'C5		A/D変換器								
<p>ビット：    7        6        5        4        3        2        1        0</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 2px 10px;">ADR7</td> <td style="padding: 2px 10px;">ADR6</td> <td style="padding: 2px 10px;">ADR5</td> <td style="padding: 2px 10px;">ADR4</td> <td style="padding: 2px 10px;">ADR3</td> <td style="padding: 2px 10px;">ADR2</td> <td style="padding: 2px 10px;">ADR1</td> <td style="padding: 2px 10px;">ADR0</td> </tr> </table> <p>初期値： 不定    不定    不定    不定    不定    不定    不定    不定</p> <p>R/W    :    R        R        R        R        R        R        R        R</p> <p style="text-align: center; margin-top: 10px;">└──────────────────┘ A/D変換結果</p>				ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0				

ADSR A/Dスタートレジスタ	H'C6		A/D変換器																		
<p>ビット：    7        6        5        4        3        2        1        0</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 2px 10px;">ADSF</td> <td style="padding: 2px 10px;">—</td> </tr> </table> <p>初期値：    0        1        1        1        1        1        1        1</p> <p>R/W    :    R/W    —        —        —        —        —        —        —</p> <p style="text-align: center; margin-top: 10px;">└──────────┘ A/Dスタートフラグ</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse; text-align: center;"> <tr> <td rowspan="2" style="padding: 5px;">0</td> <td style="padding: 2px 10px;">リード時</td> <td style="padding: 2px 10px;">A/D変換の終了</td> </tr> <tr> <td style="padding: 2px 10px;">ライト時</td> <td style="padding: 2px 10px;">A/D変換を強制終了</td> </tr> <tr> <td rowspan="2" style="padding: 5px;">1</td> <td style="padding: 2px 10px;">リード時</td> <td style="padding: 2px 10px;">A/D変換中</td> </tr> <tr> <td style="padding: 2px 10px;">ライト時</td> <td style="padding: 2px 10px;">A/D変換を開始</td> </tr> </table>				ADSF	—	—	—	—	—	—	—	0	リード時	A/D変換の終了	ライト時	A/D変換を強制終了	1	リード時	A/D変換中	ライト時	A/D変換を開始
ADSF	—	—	—	—	—	—	—														
0	リード時	A/D変換の終了																			
	ライト時	A/D変換を強制終了																			
1	リード時	A/D変換中																			
	ライト時	A/D変換を開始																			

PDR1 ポートデータレジスタ1		H'D4						I/Oポート
ビット :	7	6	5	4	3	2	1	0
	P17	P16	P15	P14	P13	P12	P11	P10
初期値 :	0	0	0	0	0	0	0	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PDR2 ポートデータレジスタ2		H'D5						I/Oポート
ビット :	7	6	5	4	3	2	1	0
	P27	P26	P25	P24	P23	P22	P21	P20
初期値 :	0	0	0	0	0	0	0	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PDR5 ポートデータレジスタ5		H'D8						I/Oポート	
ビット :		7	6	5	4	3	2	1	0
		P57	P56	P55	P54	P53	P52	P51	P50
初期値 :		0	0	0	0	0	0	0	0
R/W :		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PDR6 ポートデータレジスタ6		H'D9						I/Oポート	
ビット :		7	6	5	4	3	2	1	0
		P67	P66	P65	P64	P63	P62	P61	P60
初期値 :		0	0	0	0	0	0	0	0
R/W :		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PDR7 ポートデータレジスタ7	H'DA		I/Oポート								
<p>ビット：    7        6        5        4        3        2        1        0</p> <table border="1" style="margin-left: 40px; border-collapse: collapse; text-align: center;"> <tr> <td style="width: 20px;">P77</td><td style="width: 20px;">P76</td><td style="width: 20px;">P75</td><td style="width: 20px;">P74</td><td style="width: 20px;">P73</td><td style="width: 20px;">P72</td><td style="width: 20px;">P71</td><td style="width: 20px;">P70</td> </tr> </table> <p>初期値：    0        0        0        0        0        0        0        0</p> <p>R/W    :    R/W    R/W    R/W    R/W    R/W    R/W    R/W</p>				P77	P76	P75	P74	P73	P72	P71	P70
P77	P76	P75	P74	P73	P72	P71	P70				
PDR8 ポートデータレジスタ8	H'DB		I/Oポート								
<p>ビット：    7        6        5        4        3        2        1        0</p> <table border="1" style="margin-left: 40px; border-collapse: collapse; text-align: center;"> <tr> <td style="width: 20px;">P87</td><td style="width: 20px;">P86</td><td style="width: 20px;">P85</td><td style="width: 20px;">P84</td><td style="width: 20px;">P83</td><td style="width: 20px;">P82</td><td style="width: 20px;">P81</td><td style="width: 20px;">P80</td> </tr> </table> <p>初期値：    0        0        0        0        0        0        0        0</p> <p>R/W    :    R/W    R/W    R/W    R/W    R/W    R/W    R/W</p>				P87	P86	P85	P84	P83	P82	P81	P80
P87	P86	P85	P84	P83	P82	P81	P80				
PDRA ポートデータレジスタA	H'DD		I/Oポート								
<p>ビット：    7        6        5        4        3        2        1        0</p> <table border="1" style="margin-left: 40px; border-collapse: collapse; text-align: center;"> <tr> <td style="width: 20px;">—</td><td style="width: 20px;">—</td><td style="width: 20px;">—</td><td style="width: 20px;">—</td><td style="width: 20px;">PA3</td><td style="width: 20px;">PA2</td><td style="width: 20px;">PA1</td><td style="width: 20px;">—</td> </tr> </table> <p>初期値：    1        1        1        1        0        0        0        0</p> <p>R/W    :    —        —        —        —        R/W    R/W    R/W    —</p>				—	—	—	—	PA3	PA2	PA1	—
—	—	—	—	PA3	PA2	PA1	—				

PDRB ポートデータレジスタB	H'DE		I/Oポート												
<p>ビット：    7        6        5        4        3        2        1        0</p> <table border="1" style="margin-left: 40px; border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 2px 10px;">PB<sub>7</sub></td> <td style="padding: 2px 10px;">PB<sub>6</sub></td> <td style="padding: 2px 10px;">—</td> </tr> </table> <p>R/W    :    R        R        —        —        —        —        —        —</p>				PB <sub>7</sub>	PB <sub>6</sub>	—	—	—	—	—	—				
PB <sub>7</sub>	PB <sub>6</sub>	—	—	—	—	—	—								
PCR1 ポートコントロールレジスタ1	H'E4		I/Oポート												
<p>ビット：    7        6        5        4        3        2        1        0</p> <table border="1" style="margin-left: 40px; border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 2px 10px;">PCR<sub>17</sub></td> <td style="padding: 2px 10px;">PCR<sub>16</sub></td> <td style="padding: 2px 10px;">PCR<sub>15</sub></td> <td style="padding: 2px 10px;">PCR<sub>14</sub></td> <td style="padding: 2px 10px;">PCR<sub>13</sub></td> <td style="padding: 2px 10px;">PCR<sub>12</sub></td> <td style="padding: 2px 10px;">PCR<sub>11</sub></td> <td style="padding: 2px 10px;">PCR<sub>10</sub></td> </tr> </table> <p>初期値：    0        0        0        0        0        0        0        0</p> <p>R/W    :    W        W        W        W        W        W        W        W</p> <div style="margin-left: 100px; margin-top: 10px;"> <p>ポート1入出力選択</p> <table border="1" style="margin-left: 20px; border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 2px 10px;">0</td> <td style="padding: 2px 10px;">入力ポート</td> </tr> <tr> <td style="padding: 2px 10px;">1</td> <td style="padding: 2px 10px;">出力ポート</td> </tr> </table> </div>				PCR <sub>17</sub>	PCR <sub>16</sub>	PCR <sub>15</sub>	PCR <sub>14</sub>	PCR <sub>13</sub>	PCR <sub>12</sub>	PCR <sub>11</sub>	PCR <sub>10</sub>	0	入力ポート	1	出力ポート
PCR <sub>17</sub>	PCR <sub>16</sub>	PCR <sub>15</sub>	PCR <sub>14</sub>	PCR <sub>13</sub>	PCR <sub>12</sub>	PCR <sub>11</sub>	PCR <sub>10</sub>								
0	入力ポート														
1	出力ポート														

PCR2 ポートコントロールレジスタ2			H'E5					I/Oポート				
ビット :	7	6	5	4	3	2	1	0				
	PCR2 <sub>7</sub>	PCR2 <sub>6</sub>	PCR2 <sub>5</sub>	PCR2 <sub>4</sub>	PCR2 <sub>3</sub>	PCR2 <sub>2</sub>	PCR2 <sub>1</sub>	PCR2 <sub>0</sub>				
初期値 :	0	0	0	0	0	0	0	0				
R/W :	W	W	W	W	W	W	W	W				
ポート2入出力選択 <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>0</td> <td>入力ポート</td> </tr> <tr> <td>1</td> <td>出力ポート</td> </tr> </table>									0	入力ポート	1	出力ポート
0	入力ポート											
1	出力ポート											
【注】 P2 <sub>7</sub> 端子は入力専用端子のため、PCR2 <sub>7</sub> に"1"をセットした場合、ハイインピーダンス出力となります。												

PCR5 ポートコントロールレジスタ5			H'E8					I/Oポート				
ビット :	7	6	5	4	3	2	1	0				
	PCR5 <sub>7</sub>	PCR5 <sub>6</sub>	PCR5 <sub>5</sub>	PCR5 <sub>4</sub>	PCR5 <sub>3</sub>	PCR5 <sub>2</sub>	PCR5 <sub>1</sub>	PCR5 <sub>0</sub>				
初期値 :	0	0	0	0	0	0	0	0				
R/W :	W	W	W	W	W	W	W	W				
ポート5入出力選択 <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>0</td> <td>入力ポート</td> </tr> <tr> <td>1</td> <td>出力ポート</td> </tr> </table>									0	入力ポート	1	出力ポート
0	入力ポート											
1	出力ポート											

PCR6 ポートコントロールレジスタ6			H'E9					I/Oポート				
ビット :	7	6	5	4	3	2	1	0				
	PCR6 <sub>7</sub>	PCR6 <sub>6</sub>	PCR6 <sub>5</sub>	PCR6 <sub>4</sub>	PCR6 <sub>3</sub>	PCR6 <sub>2</sub>	PCR6 <sub>1</sub>	PCR6 <sub>0</sub>				
初期値 :	0	0	0	0	0	0	0	0				
R/W :	W	W	W	W	W	W	W	W				
ポート6入出力選択 <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>0</td> <td>入力ポート</td> </tr> <tr> <td>1</td> <td>出力ポート</td> </tr> </table>									0	入力ポート	1	出力ポート
0	入力ポート											
1	出力ポート											

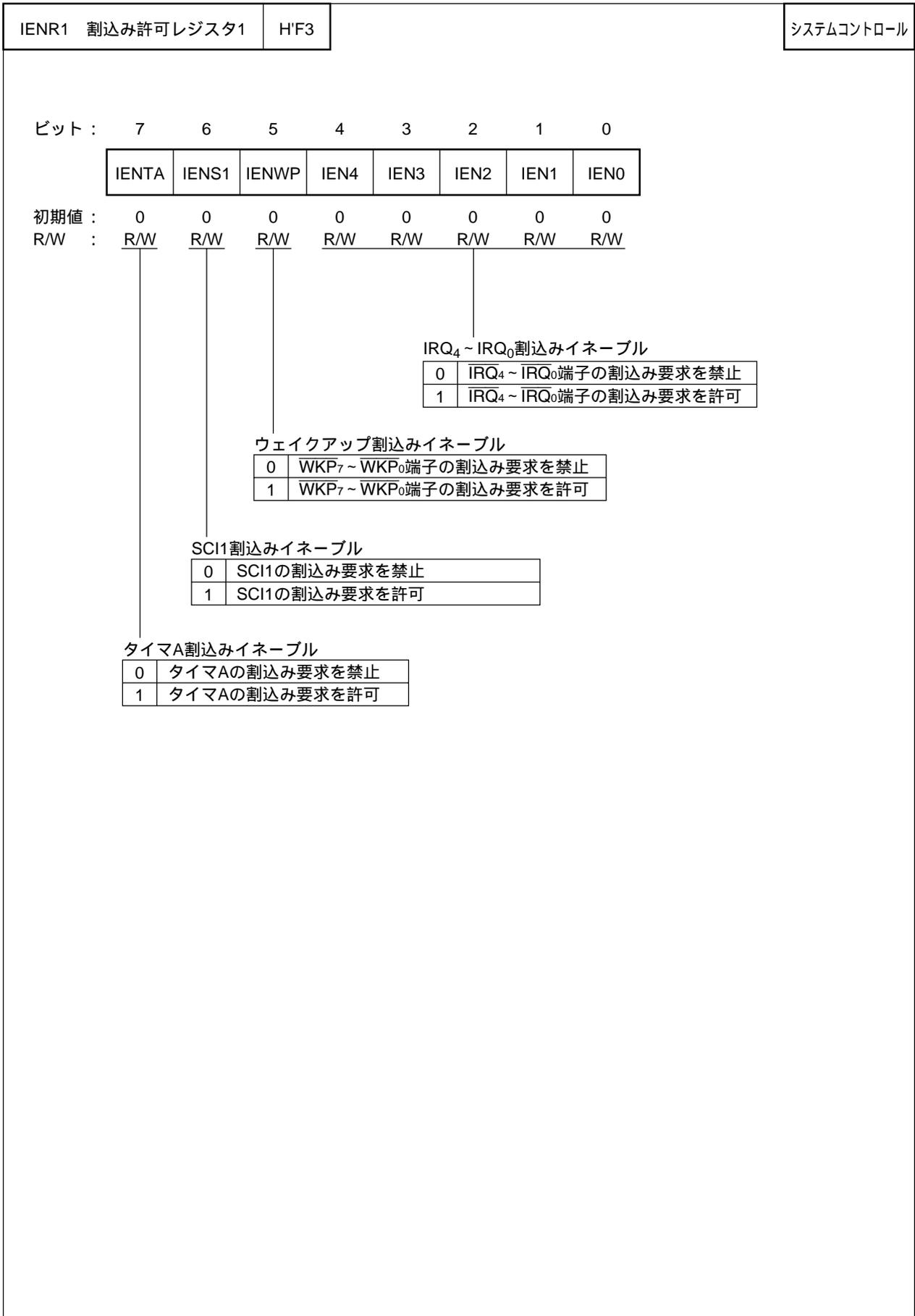
PCR7 ポートコントロールレジスタ7	H'EA		I/Oポート												
<p>ビット :    7        6        5        4        3        2        1        0</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr> <td style="padding: 2px;">PCR7<sub>7</sub></td> <td style="padding: 2px;">PCR7<sub>6</sub></td> <td style="padding: 2px;">PCR7<sub>5</sub></td> <td style="padding: 2px;">PCR7<sub>4</sub></td> <td style="padding: 2px;">PCR7<sub>3</sub></td> <td style="padding: 2px;">PCR7<sub>2</sub></td> <td style="padding: 2px;">PCR7<sub>1</sub></td> <td style="padding: 2px;">PCR7<sub>0</sub></td> </tr> </table> <p>初期値 :    0        0        0        0        0        0        0        0</p> <p>R/W :        W        W        W        W        W        W        W        W</p> <div style="text-align: center; margin-top: 10px;"> <p>ポート7入出力選択</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr> <td style="padding: 2px;">0</td> <td style="padding: 2px;">入力ポート</td> </tr> <tr> <td style="padding: 2px;">1</td> <td style="padding: 2px;">出力ポート</td> </tr> </table> </div>				PCR7 <sub>7</sub>	PCR7 <sub>6</sub>	PCR7 <sub>5</sub>	PCR7 <sub>4</sub>	PCR7 <sub>3</sub>	PCR7 <sub>2</sub>	PCR7 <sub>1</sub>	PCR7 <sub>0</sub>	0	入力ポート	1	出力ポート
PCR7 <sub>7</sub>	PCR7 <sub>6</sub>	PCR7 <sub>5</sub>	PCR7 <sub>4</sub>	PCR7 <sub>3</sub>	PCR7 <sub>2</sub>	PCR7 <sub>1</sub>	PCR7 <sub>0</sub>								
0	入力ポート														
1	出力ポート														
PCR8 ポートコントロールレジスタ8	H'EB		I/Oポート												
<p>ビット :    7        6        5        4        3        2        1        0</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr> <td style="padding: 2px;">PCR8<sub>7</sub></td> <td style="padding: 2px;">PCR8<sub>6</sub></td> <td style="padding: 2px;">PCR8<sub>5</sub></td> <td style="padding: 2px;">PCR8<sub>4</sub></td> <td style="padding: 2px;">PCR8<sub>3</sub></td> <td style="padding: 2px;">PCR8<sub>2</sub></td> <td style="padding: 2px;">PCR8<sub>1</sub></td> <td style="padding: 2px;">PCR8<sub>0</sub></td> </tr> </table> <p>初期値 :    0        0        0        0        0        0        0        0</p> <p>R/W :        W        W        W        W        W        W        W        W</p> <div style="text-align: center; margin-top: 10px;"> <p>ポート8入出力選択</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr> <td style="padding: 2px;">0</td> <td style="padding: 2px;">入力ポート</td> </tr> <tr> <td style="padding: 2px;">1</td> <td style="padding: 2px;">出力ポート</td> </tr> </table> </div>				PCR8 <sub>7</sub>	PCR8 <sub>6</sub>	PCR8 <sub>5</sub>	PCR8 <sub>4</sub>	PCR8 <sub>3</sub>	PCR8 <sub>2</sub>	PCR8 <sub>1</sub>	PCR8 <sub>0</sub>	0	入力ポート	1	出力ポート
PCR8 <sub>7</sub>	PCR8 <sub>6</sub>	PCR8 <sub>5</sub>	PCR8 <sub>4</sub>	PCR8 <sub>3</sub>	PCR8 <sub>2</sub>	PCR8 <sub>1</sub>	PCR8 <sub>0</sub>								
0	入力ポート														
1	出力ポート														

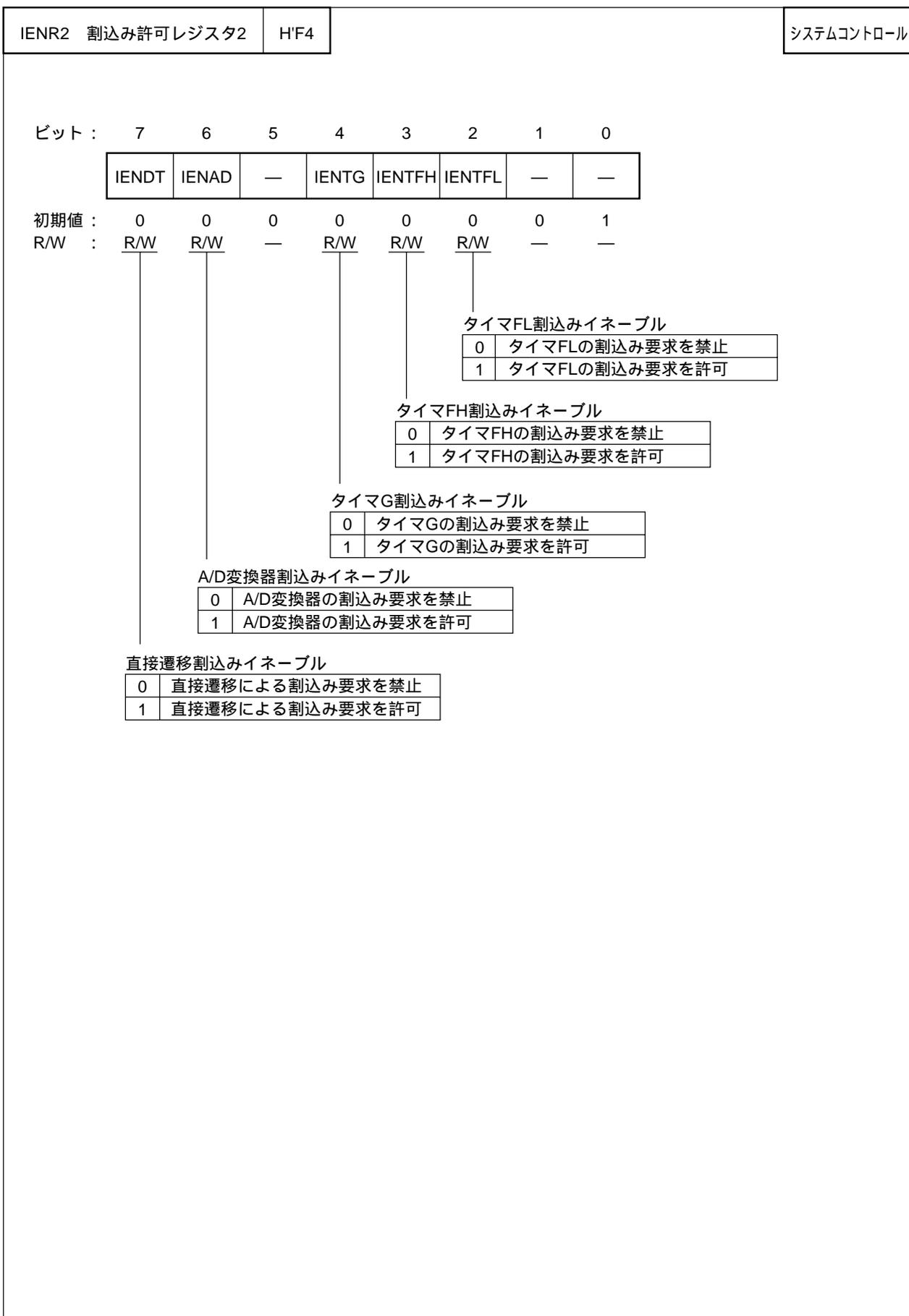
PCRA ポートコントロールレジスタA	H'ED	I/Oポート												
<p>ビット： 7      6      5      4      3      2      1      0</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr> <td style="width: 20px; height: 20px; text-align: center;">—</td> <td style="width: 20px; height: 20px; text-align: center;">PCRA<sub>3</sub></td> <td style="width: 20px; height: 20px; text-align: center;">PCRA<sub>2</sub></td> <td style="width: 20px; height: 20px; text-align: center;">PCRA<sub>1</sub></td> <td style="width: 20px; height: 20px; text-align: center;">—</td> </tr> </table> <p>初期値： 1      1      1      1      0      0      0      1</p> <p>R/W： —      —      —      —      W      W      W      —</p> <div style="margin-left: 200px; margin-top: 10px;"> <p>ポートA入出力選択</p> <table border="1" style="border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">0</td> <td>入力ポート</td> </tr> <tr> <td style="width: 20px; text-align: center;">1</td> <td>出力ポート</td> </tr> </table> </div>			—	—	—	—	PCRA <sub>3</sub>	PCRA <sub>2</sub>	PCRA <sub>1</sub>	—	0	入力ポート	1	出力ポート
—	—	—	—	PCRA <sub>3</sub>	PCRA <sub>2</sub>	PCRA <sub>1</sub>	—							
0	入力ポート													
1	出力ポート													

SYSCR1 システムコントロールレジスタ1	H'F0	システムコントロール																																				
<p>ビット： 7      6      5      4      3      2      1      0</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr> <td style="width: 20px; height: 20px; text-align: center;">SSBY</td> <td style="width: 20px; height: 20px; text-align: center;">STS2</td> <td style="width: 20px; height: 20px; text-align: center;">STS1</td> <td style="width: 20px; height: 20px; text-align: center;">STS0</td> <td style="width: 20px; height: 20px; text-align: center;">LSON</td> <td style="width: 20px; height: 20px; text-align: center;">—</td> <td style="width: 20px; height: 20px; text-align: center;">—</td> <td style="width: 20px; height: 20px; text-align: center;">—</td> </tr> </table> <p>初期値： 0      0      0      0      0      1      1      1</p> <p>R/W： R/W      R/W      R/W      R/W      R/W      —      —      —</p> <div style="margin-left: 200px; margin-top: 10px;"> <p>ロースピードオンフラグ</p> <table border="1" style="border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">0</td> <td>CPUの動作クロックはシステムクロック ( )</td> </tr> <tr> <td style="width: 20px; text-align: center;">1</td> <td>CPUの動作クロックはサブクロック ( SUB )</td> </tr> </table> </div> <div style="margin-left: 100px; margin-top: 10px;"> <p>スタンバイタイムセレクト2~0</p> <table border="1" style="border-collapse: collapse;"> <tr> <td style="width: 15px; text-align: center;">0</td> <td style="width: 15px; text-align: center;">0</td> <td style="width: 15px; text-align: center;">0</td> <td>待機時間 = 8192ステート</td> </tr> <tr> <td style="width: 15px; text-align: center;">0</td> <td style="width: 15px; text-align: center;">0</td> <td style="width: 15px; text-align: center;">1</td> <td>待機時間 = 16384ステート</td> </tr> <tr> <td style="width: 15px; text-align: center;">0</td> <td style="width: 15px; text-align: center;">1</td> <td style="width: 15px; text-align: center;">0</td> <td>待機時間 = 32768ステート</td> </tr> <tr> <td style="width: 15px; text-align: center;">0</td> <td style="width: 15px; text-align: center;">1</td> <td style="width: 15px; text-align: center;">1</td> <td>待機時間 = 65536ステート</td> </tr> <tr> <td style="width: 15px; text-align: center;">1</td> <td style="width: 15px; text-align: center;">*</td> <td style="width: 15px; text-align: center;">*</td> <td>待機時間 = 131072ステート</td> </tr> </table> <p style="text-align: right; margin-right: 50px;">* : Don't care</p> </div> <div style="margin-left: 100px; margin-top: 10px;"> <p>ソフトウェアスタンバイ</p> <table border="1" style="border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">0</td> <td> <ul style="list-style-type: none"> <li>・アクティブモードでSLEEP命令実行後、スリープモードに遷移</li> <li>・サブアクティブモードでSLEEP命令実行後、サブスリープモードに遷移</li> </ul> </td> </tr> <tr> <td style="width: 20px; text-align: center;">1</td> <td> <ul style="list-style-type: none"> <li>・アクティブモードでSLEEP命令実行後、スタンバイモードあるいはウォッチモードに遷移</li> <li>・サブアクティブモードでSLEEP命令実行後、ウォッチモードに遷移</li> </ul> </td> </tr> </table> </div>			SSBY	STS2	STS1	STS0	LSON	—	—	—	0	CPUの動作クロックはシステムクロック ( )	1	CPUの動作クロックはサブクロック ( SUB )	0	0	0	待機時間 = 8192ステート	0	0	1	待機時間 = 16384ステート	0	1	0	待機時間 = 32768ステート	0	1	1	待機時間 = 65536ステート	1	*	*	待機時間 = 131072ステート	0	<ul style="list-style-type: none"> <li>・アクティブモードでSLEEP命令実行後、スリープモードに遷移</li> <li>・サブアクティブモードでSLEEP命令実行後、サブスリープモードに遷移</li> </ul>	1	<ul style="list-style-type: none"> <li>・アクティブモードでSLEEP命令実行後、スタンバイモードあるいはウォッチモードに遷移</li> <li>・サブアクティブモードでSLEEP命令実行後、ウォッチモードに遷移</li> </ul>
SSBY	STS2	STS1	STS0	LSON	—	—	—																															
0	CPUの動作クロックはシステムクロック ( )																																					
1	CPUの動作クロックはサブクロック ( SUB )																																					
0	0	0	待機時間 = 8192ステート																																			
0	0	1	待機時間 = 16384ステート																																			
0	1	0	待機時間 = 32768ステート																																			
0	1	1	待機時間 = 65536ステート																																			
1	*	*	待機時間 = 131072ステート																																			
0	<ul style="list-style-type: none"> <li>・アクティブモードでSLEEP命令実行後、スリープモードに遷移</li> <li>・サブアクティブモードでSLEEP命令実行後、サブスリープモードに遷移</li> </ul>																																					
1	<ul style="list-style-type: none"> <li>・アクティブモードでSLEEP命令実行後、スタンバイモードあるいはウォッチモードに遷移</li> <li>・サブアクティブモードでSLEEP命令実行後、ウォッチモードに遷移</li> </ul>																																					

SYSCR2 システムコントロールレジスタ2	HF1	システムコントロール																													
<p>ビット :    7        6        5        4        3        2        1        0</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">—</td> <td style="width: 20px; text-align: center;">—</td> <td style="width: 20px; text-align: center;">—</td> <td style="width: 20px; text-align: center;">NESEL</td> <td style="width: 20px; text-align: center;">DTON</td> <td style="width: 20px; text-align: center;">MSON</td> <td style="width: 20px; text-align: center;">SA1</td> <td style="width: 20px; text-align: center;">SA0</td> </tr> </table> <p>初期値 :    1        1        1        0        0        0        0        0</p> <p>R/W :        —        —        —        R/W    R/W    R/W    R/W    R/W</p> <div style="margin-top: 20px;"> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <caption>サブアクティブモードクロックセレクト</caption> <tr><td style="width: 20px; text-align: center;">0</td><td style="width: 20px; text-align: center;">0</td><td style="width: 20px; text-align: center;">w/8</td></tr> <tr><td style="width: 20px; text-align: center;">0</td><td style="width: 20px; text-align: center;">1</td><td style="width: 20px; text-align: center;">w/4</td></tr> <tr><td style="width: 20px; text-align: center;">1</td><td style="width: 20px; text-align: center;">*</td><td style="width: 20px; text-align: center;">w/2</td></tr> </table> <p style="text-align: right; margin-right: 50px;">* : Don't care</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <caption>ミドルスピードオンフラグ</caption> <tr><td style="width: 20px; text-align: center;">0</td><td style="width: 200px;">アクティブ (高速) モードで動作</td></tr> <tr><td style="width: 20px; text-align: center;">1</td><td style="width: 200px;">アクティブ (中速) モードで動作</td></tr> </table> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <caption>ダイレクトトランスファオンフラグ</caption> <tr><td style="width: 20px; text-align: center;">0</td><td style="width: 200px;"> <ul style="list-style-type: none"> <li>・アクティブモードでSLEEP命令を実行したとき、スタンバイモード、ウォッチモード、またはスリープモードに遷移</li> <li>・サブアクティブモードでSLEEP命令を実行したとき、ウォッチモード、またはサブスリープモードに遷移</li> </ul> </td></tr> <tr><td style="width: 20px; text-align: center;">1</td><td style="width: 200px;"> <ul style="list-style-type: none"> <li>・アクティブ (高速) モードでSLEEP命令を実行したとき、アクティブ (中速) モード (SSBY = "0"、MSON = "1"、LSON = "0"のとき)、またはサブアクティブモード (SSBY = "1"、TMA3 = "1"、LSON = "1"のとき) に直接遷移</li> <li>・アクティブ (中速) モードでSLEEP命令を実行したとき、アクティブ (高速) モード (SSBY = "0"、MSON = "0"、LSON = "0"のとき)、またはサブアクティブモード (SSBY = "1"、TMA3 = "1"、LSON = "1"のとき) に直接遷移</li> <li>・サブアクティブモードでSLEEP命令を実行したとき、アクティブ (高速) モード (SSBY = "1"、TMA3 = "1"、LSON = "0"、MSON = "0"のとき) またはアクティブ (中速) モード (SSBY = "1"、TMA3 = "1"、LSON = "0"、MSON = "1"のとき) に直接遷移</li> </ul> </td></tr> </table> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <caption>ノイズ除去サンプリング周波数選択</caption> <tr><td style="width: 20px; text-align: center;">0</td><td style="width: 200px;">oscの16分周クロックでサンプリング</td></tr> <tr><td style="width: 20px; text-align: center;">1</td><td style="width: 200px;">oscの4分周クロックでサンプリング</td></tr> </table> </div>			—	—	—	NESEL	DTON	MSON	SA1	SA0	0	0	w/8	0	1	w/4	1	*	w/2	0	アクティブ (高速) モードで動作	1	アクティブ (中速) モードで動作	0	<ul style="list-style-type: none"> <li>・アクティブモードでSLEEP命令を実行したとき、スタンバイモード、ウォッチモード、またはスリープモードに遷移</li> <li>・サブアクティブモードでSLEEP命令を実行したとき、ウォッチモード、またはサブスリープモードに遷移</li> </ul>	1	<ul style="list-style-type: none"> <li>・アクティブ (高速) モードでSLEEP命令を実行したとき、アクティブ (中速) モード (SSBY = "0"、MSON = "1"、LSON = "0"のとき)、またはサブアクティブモード (SSBY = "1"、TMA3 = "1"、LSON = "1"のとき) に直接遷移</li> <li>・アクティブ (中速) モードでSLEEP命令を実行したとき、アクティブ (高速) モード (SSBY = "0"、MSON = "0"、LSON = "0"のとき)、またはサブアクティブモード (SSBY = "1"、TMA3 = "1"、LSON = "1"のとき) に直接遷移</li> <li>・サブアクティブモードでSLEEP命令を実行したとき、アクティブ (高速) モード (SSBY = "1"、TMA3 = "1"、LSON = "0"、MSON = "0"のとき) またはアクティブ (中速) モード (SSBY = "1"、TMA3 = "1"、LSON = "0"、MSON = "1"のとき) に直接遷移</li> </ul>	0	oscの16分周クロックでサンプリング	1	oscの4分周クロックでサンプリング
—	—	—	NESEL	DTON	MSON	SA1	SA0																								
0	0	w/8																													
0	1	w/4																													
1	*	w/2																													
0	アクティブ (高速) モードで動作																														
1	アクティブ (中速) モードで動作																														
0	<ul style="list-style-type: none"> <li>・アクティブモードでSLEEP命令を実行したとき、スタンバイモード、ウォッチモード、またはスリープモードに遷移</li> <li>・サブアクティブモードでSLEEP命令を実行したとき、ウォッチモード、またはサブスリープモードに遷移</li> </ul>																														
1	<ul style="list-style-type: none"> <li>・アクティブ (高速) モードでSLEEP命令を実行したとき、アクティブ (中速) モード (SSBY = "0"、MSON = "1"、LSON = "0"のとき)、またはサブアクティブモード (SSBY = "1"、TMA3 = "1"、LSON = "1"のとき) に直接遷移</li> <li>・アクティブ (中速) モードでSLEEP命令を実行したとき、アクティブ (高速) モード (SSBY = "0"、MSON = "0"、LSON = "0"のとき)、またはサブアクティブモード (SSBY = "1"、TMA3 = "1"、LSON = "1"のとき) に直接遷移</li> <li>・サブアクティブモードでSLEEP命令を実行したとき、アクティブ (高速) モード (SSBY = "1"、TMA3 = "1"、LSON = "0"、MSON = "0"のとき) またはアクティブ (中速) モード (SSBY = "1"、TMA3 = "1"、LSON = "0"、MSON = "1"のとき) に直接遷移</li> </ul>																														
0	oscの16分周クロックでサンプリング																														
1	oscの4分周クロックでサンプリング																														

IEGR 割込みエッジセレクトレジスタ			HF2		システムコントロール							
ビット :	7	6	5	4	3	2	1	0				
	—	—	—	IEG4	IEG3	IEG2	IEG1	IEG0				
初期値 :	0	1	1	0	0	0	0	0				
R/W :	—	—	—	R/W	R/W	R/W	R/W	R/W				
				<p>IRQ<sub>0</sub>エッジセレクト</p> <table border="1"> <tr> <td>0</td> <td>IRQ<sub>0</sub>端子入力の立下がりエッジを検出</td> </tr> <tr> <td>1</td> <td>IRQ<sub>0</sub>端子入力の立上がりエッジを検出</td> </tr> </table>					0	IRQ <sub>0</sub> 端子入力の立下がりエッジを検出	1	IRQ <sub>0</sub> 端子入力の立上がりエッジを検出
0	IRQ <sub>0</sub> 端子入力の立下がりエッジを検出											
1	IRQ <sub>0</sub> 端子入力の立上がりエッジを検出											
				<p>IRQ<sub>1</sub>エッジセレクト</p> <table border="1"> <tr> <td>0</td> <td>IRQ<sub>1</sub>端子入力の立下がりエッジを検出</td> </tr> <tr> <td>1</td> <td>IRQ<sub>1</sub>端子入力の立上がりエッジを検出</td> </tr> </table>					0	IRQ <sub>1</sub> 端子入力の立下がりエッジを検出	1	IRQ <sub>1</sub> 端子入力の立上がりエッジを検出
0	IRQ <sub>1</sub> 端子入力の立下がりエッジを検出											
1	IRQ <sub>1</sub> 端子入力の立上がりエッジを検出											
				<p>IRQ<sub>2</sub>エッジセレクト</p> <table border="1"> <tr> <td>0</td> <td>IRQ<sub>2</sub>端子入力の立下がりエッジを検出</td> </tr> <tr> <td>1</td> <td>IRQ<sub>2</sub>端子入力の立上がりエッジを検出</td> </tr> </table>					0	IRQ <sub>2</sub> 端子入力の立下がりエッジを検出	1	IRQ <sub>2</sub> 端子入力の立上がりエッジを検出
0	IRQ <sub>2</sub> 端子入力の立下がりエッジを検出											
1	IRQ <sub>2</sub> 端子入力の立上がりエッジを検出											
				<p>IRQ<sub>3</sub>エッジセレクト</p> <table border="1"> <tr> <td>0</td> <td>IRQ<sub>3</sub>端子、TMIF端子入力の立下がりエッジを検出</td> </tr> <tr> <td>1</td> <td>IRQ<sub>3</sub>端子、TMIF端子入力の立上がりエッジを検出</td> </tr> </table>					0	IRQ <sub>3</sub> 端子、TMIF端子入力の立下がりエッジを検出	1	IRQ <sub>3</sub> 端子、TMIF端子入力の立上がりエッジを検出
0	IRQ <sub>3</sub> 端子、TMIF端子入力の立下がりエッジを検出											
1	IRQ <sub>3</sub> 端子、TMIF端子入力の立上がりエッジを検出											
				<p>IRQ<sub>4</sub>エッジセレクト</p> <table border="1"> <tr> <td>0</td> <td>IRQ<sub>4</sub>端子、ADTRG端子入力の立下がりエッジを検出</td> </tr> <tr> <td>1</td> <td>IRQ<sub>4</sub>端子、ADTRG端子入力の立上がりエッジを検出</td> </tr> </table>					0	IRQ <sub>4</sub> 端子、ADTRG端子入力の立下がりエッジを検出	1	IRQ <sub>4</sub> 端子、ADTRG端子入力の立上がりエッジを検出
0	IRQ <sub>4</sub> 端子、ADTRG端子入力の立下がりエッジを検出											
1	IRQ <sub>4</sub> 端子、ADTRG端子入力の立上がりエッジを検出											





IRR1 割り込み要求レジスタ1	H'F6	システムコントロール																				
<p>ビット：     7       6       5       4       3       2       1       0</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr> <td style="padding: 2px 5px;">IRRTA</td> <td style="padding: 2px 5px;">IRRS1</td> <td style="padding: 2px 5px;">—</td> <td style="padding: 2px 5px;">IRRI4</td> <td style="padding: 2px 5px;">IRRI3</td> <td style="padding: 2px 5px;">IRRI2</td> <td style="padding: 2px 5px;">IRRI1</td> <td style="padding: 2px 5px;">IRRI0</td> </tr> </table> <p>初期値：     0       0       1       0       0       0       0       0</p> <p>R/W：       R/W*   R/W*   —     R/W*   R/W*   R/W*   R/W*   R/W*</p> <div style="margin-top: 20px;"> <p style="text-align: center;">IRQ<sub>4</sub> ~ IRQ<sub>0</sub>割り込み要求フラグ</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr> <td style="padding: 2px 5px;">0</td> <td style="padding: 2px 5px;">〔クリア条件〕 IRRI<sub>n</sub> = "1"の状態ではIRRI<sub>n</sub>に"0"をライトしたとき</td> </tr> <tr> <td style="padding: 2px 5px;">1</td> <td style="padding: 2px 5px;">〔セット条件〕 IRQ<sub>n</sub>端子が割り込み入力に設定されており、かつ当該端子に指定されたエッジが入力されたとき</td> </tr> </table> <p style="text-align: right;">(n = 4 ~ 0)</p> </div> <div style="margin-top: 20px;"> <p style="text-align: center;">SCI1割り込み要求フラグ</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr> <td style="padding: 2px 5px;">0</td> <td style="padding: 2px 5px;">〔クリア条件〕 IRRS1 = "1"の状態ではIRRS1に"0"をライトしたとき</td> </tr> <tr> <td style="padding: 2px 5px;">1</td> <td style="padding: 2px 5px;">〔セット条件〕 SCI1が転送完了したとき</td> </tr> </table> </div> <div style="margin-top: 20px;"> <p style="text-align: center;">タイマA割り込み要求フラグ</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr> <td style="padding: 2px 5px;">0</td> <td style="padding: 2px 5px;">〔クリア条件〕 IRRTA = "1"の状態ではIRRTAに"0"をライトしたとき</td> </tr> <tr> <td style="padding: 2px 5px;">1</td> <td style="padding: 2px 5px;">〔セット条件〕 タイマAのカウンタ値がオーバーフロー (H'FF H'00) したとき</td> </tr> </table> </div>			IRRTA	IRRS1	—	IRRI4	IRRI3	IRRI2	IRRI1	IRRI0	0	〔クリア条件〕 IRRI <sub>n</sub> = "1"の状態ではIRRI <sub>n</sub> に"0"をライトしたとき	1	〔セット条件〕 IRQ <sub>n</sub> 端子が割り込み入力に設定されており、かつ当該端子に指定されたエッジが入力されたとき	0	〔クリア条件〕 IRRS1 = "1"の状態ではIRRS1に"0"をライトしたとき	1	〔セット条件〕 SCI1が転送完了したとき	0	〔クリア条件〕 IRRTA = "1"の状態ではIRRTAに"0"をライトしたとき	1	〔セット条件〕 タイマAのカウンタ値がオーバーフロー (H'FF H'00) したとき
IRRTA	IRRS1	—	IRRI4	IRRI3	IRRI2	IRRI1	IRRI0															
0	〔クリア条件〕 IRRI <sub>n</sub> = "1"の状態ではIRRI <sub>n</sub> に"0"をライトしたとき																					
1	〔セット条件〕 IRQ <sub>n</sub> 端子が割り込み入力に設定されており、かつ当該端子に指定されたエッジが入力されたとき																					
0	〔クリア条件〕 IRRS1 = "1"の状態ではIRRS1に"0"をライトしたとき																					
1	〔セット条件〕 SCI1が転送完了したとき																					
0	〔クリア条件〕 IRRTA = "1"の状態ではIRRTAに"0"をライトしたとき																					
1	〔セット条件〕 タイマAのカウンタ値がオーバーフロー (H'FF H'00) したとき																					

IRR2 割り込み要求レジスタ2		H'F7						システムコントロール																		
ビット : 7 6 5 4 3 2 1 0																										
IRRDT		IRRAD		—		IRRTG		IRRTFH		IRRTFL		—		—												
初期値 :		0		0		0		0		0		0		1												
R/W :		R/W*		R/W*		—		R/W*		R/W*		—		—												
									タイマFL割り込み要求フラグ																	
									<table border="1"> <tr> <td colspan="2">0</td> <td colspan="4">〔クリア条件〕 IRRTFL = "1"の状態 でIRRTFLに"0"をライトしたとき</td> </tr> <tr> <td colspan="2">1</td> <td colspan="4">〔セット条件〕 8ビットタイマモードでTCFLとOCRFLが一致したとき</td> </tr> </table>						0		〔クリア条件〕 IRRTFL = "1"の状態 でIRRTFLに"0"をライトしたとき				1		〔セット条件〕 8ビットタイマモードでTCFLとOCRFLが一致したとき			
0		〔クリア条件〕 IRRTFL = "1"の状態 でIRRTFLに"0"をライトしたとき																								
1		〔セット条件〕 8ビットタイマモードでTCFLとOCRFLが一致したとき																								
									タイマFH割り込み要求フラグ																	
									<table border="1"> <tr> <td colspan="2">0</td> <td colspan="4">〔クリア条件〕 IRRTFH = "1"の状態 でIRRTFHに"0"をライトしたとき</td> </tr> <tr> <td colspan="2">1</td> <td colspan="4">〔セット条件〕 8ビットタイマモードでTCFHとOCRFHが一致したとき、 また16ビットタイマモードでTCF (TCFL、TCFH) と OCRF (OCRFL、OCRFH) が一致したとき</td> </tr> </table>						0		〔クリア条件〕 IRRTFH = "1"の状態 でIRRTFHに"0"をライトしたとき				1		〔セット条件〕 8ビットタイマモードでTCFHとOCRFHが一致したとき、 また16ビットタイマモードでTCF (TCFL、TCFH) と OCRF (OCRFL、OCRFH) が一致したとき			
0		〔クリア条件〕 IRRTFH = "1"の状態 でIRRTFHに"0"をライトしたとき																								
1		〔セット条件〕 8ビットタイマモードでTCFHとOCRFHが一致したとき、 また16ビットタイマモードでTCF (TCFL、TCFH) と OCRF (OCRFL、OCRFH) が一致したとき																								
									タイマG割り込み要求フラグ																	
									<table border="1"> <tr> <td colspan="2">0</td> <td colspan="4">〔クリア条件〕 IRRTG = "1"の状態 でIRRTGに"0"をライトしたとき</td> </tr> <tr> <td colspan="2">1</td> <td colspan="4">〔セット条件〕 TMIG端子がTMIG入力に設定されており、かつ当該端子に指定 されたエッジが入力されたとき</td> </tr> </table>						0		〔クリア条件〕 IRRTG = "1"の状態 でIRRTGに"0"をライトしたとき				1		〔セット条件〕 TMIG端子がTMIG入力に設定されており、かつ当該端子に指定 されたエッジが入力されたとき			
0		〔クリア条件〕 IRRTG = "1"の状態 でIRRTGに"0"をライトしたとき																								
1		〔セット条件〕 TMIG端子がTMIG入力に設定されており、かつ当該端子に指定 されたエッジが入力されたとき																								
									A/D変換器割り込み要求フラグ																	
									<table border="1"> <tr> <td colspan="2">0</td> <td colspan="4">〔クリア条件〕 IRRAD = "1"の状態 でIRRADに"0"をライトしたとき</td> </tr> <tr> <td colspan="2">1</td> <td colspan="4">〔セット条件〕 A/D変換器が変換終了し、ADSFがリセットされたとき</td> </tr> </table>						0		〔クリア条件〕 IRRAD = "1"の状態 でIRRADに"0"をライトしたとき				1		〔セット条件〕 A/D変換器が変換終了し、ADSFがリセットされたとき			
0		〔クリア条件〕 IRRAD = "1"の状態 でIRRADに"0"をライトしたとき																								
1		〔セット条件〕 A/D変換器が変換終了し、ADSFがリセットされたとき																								
									直接遷移割り込み要求フラグ																	
									<table border="1"> <tr> <td colspan="2">0</td> <td colspan="4">〔クリア条件〕 IRRDT = "1"の状態 でIRRDTに"0"をライトしたとき</td> </tr> <tr> <td colspan="2">1</td> <td colspan="4">〔セット条件〕 DTONに"1"をセットした状態でスリープ命令を実行し直接遷移したとき</td> </tr> </table>						0		〔クリア条件〕 IRRDT = "1"の状態 でIRRDTに"0"をライトしたとき				1		〔セット条件〕 DTONに"1"をセットした状態でスリープ命令を実行し直接遷移したとき			
0		〔クリア条件〕 IRRDT = "1"の状態 でIRRDTに"0"をライトしたとき																								
1		〔セット条件〕 DTONに"1"をセットした状態でスリープ命令を実行し直接遷移したとき																								
									【注】* フラグクリアのための"0"ライトのみ可能																	

IWPR ウェイクアップ割込み要求レジスタ	H'F9	システムコントロール												
<p>ビット：    7        6        5        4        3        2        1        0</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 2px 5px;">IWPF7</td> <td style="padding: 2px 5px;">IWPF6</td> <td style="padding: 2px 5px;">IWPF5</td> <td style="padding: 2px 5px;">IWPF4</td> <td style="padding: 2px 5px;">IWPF3</td> <td style="padding: 2px 5px;">IWPF2</td> <td style="padding: 2px 5px;">IWPF1</td> <td style="padding: 2px 5px;">IWPF0</td> </tr> </table> <p>初期値：    0        0        0        0        0        0        0        0</p> <p>R/W    :    R/W*   R/W*   R/W*   R/W*   R/W*   R/W*   R/W*   R/W*</p> <p style="text-align: center;">ウェイクアップ割込み要求フラグ</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr> <td style="width: 5%; text-align: center; padding: 5px;">0</td> <td style="padding: 5px;">〔クリア条件〕 IWPF<sub>n</sub> = "1"の状態 IWPF<sub>n</sub>に"0"をライトしたとき</td> </tr> <tr> <td style="text-align: center; padding: 5px;">1</td> <td style="padding: 5px;">〔セット条件〕 <math>\overline{WKP}_n</math>端子がウェイクアップ入力に設定されており、かつ当該端子に立下がりエッジが入力されたとき</td> </tr> </table> <p style="text-align: right; margin-right: 50px;">(n=7~0)</p> <p>【注】 * フラグクリアのための"0"ライトのみ可能</p>			IWPF7	IWPF6	IWPF5	IWPF4	IWPF3	IWPF2	IWPF1	IWPF0	0	〔クリア条件〕 IWPF <sub>n</sub> = "1"の状態 IWPF <sub>n</sub> に"0"をライトしたとき	1	〔セット条件〕 $\overline{WKP}_n$ 端子がウェイクアップ入力に設定されており、かつ当該端子に立下がりエッジが入力されたとき
IWPF7	IWPF6	IWPF5	IWPF4	IWPF3	IWPF2	IWPF1	IWPF0							
0	〔クリア条件〕 IWPF <sub>n</sub> = "1"の状態 IWPF <sub>n</sub> に"0"をライトしたとき													
1	〔セット条件〕 $\overline{WKP}_n$ 端子がウェイクアップ入力に設定されており、かつ当該端子に立下がりエッジが入力されたとき													

## C. I/O ポートブロック図

### C.1 ポート1ブロック図

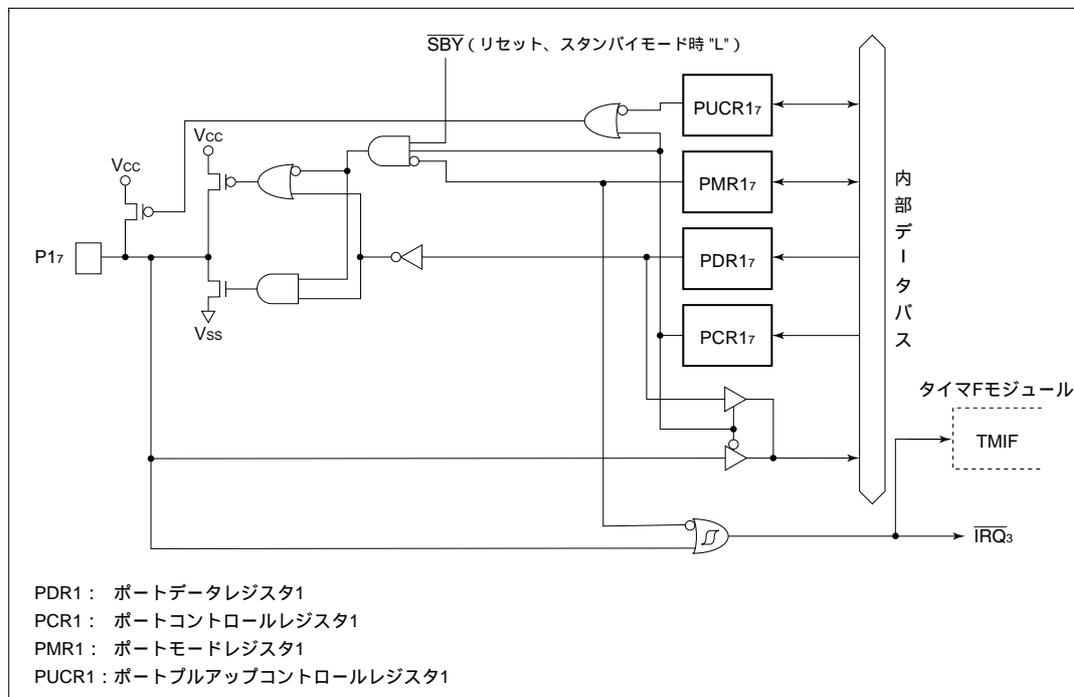


図 C.1 (a) ポート1 ブロック図 (P17端子)

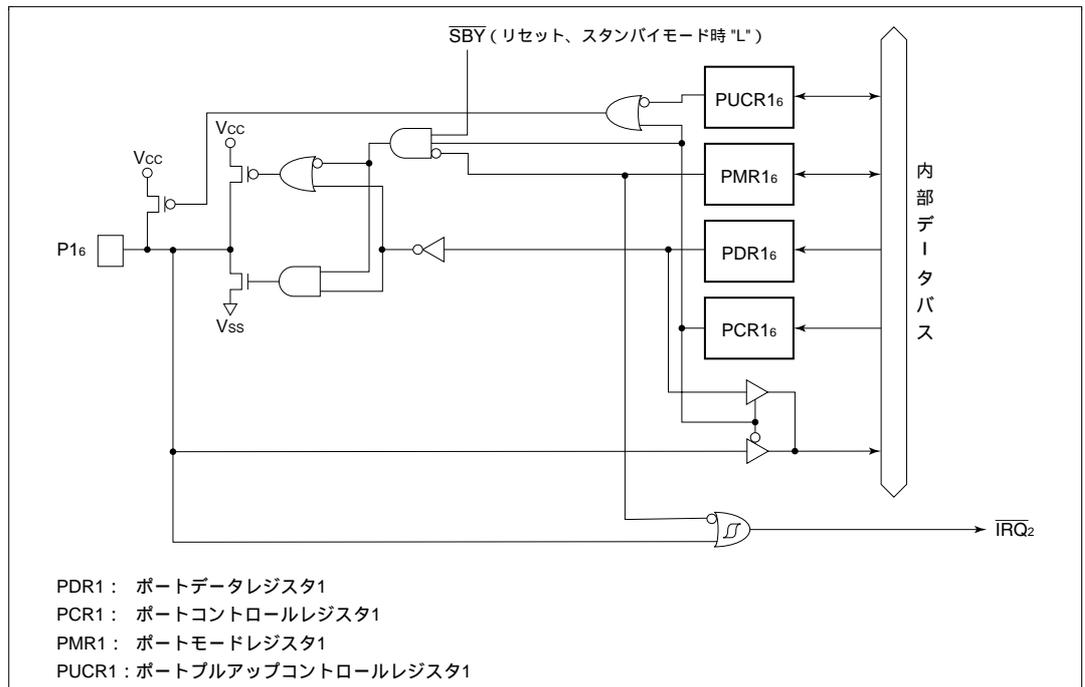


図 C.1 (b) ポート1 ブロック図 (P1<sub>6</sub>端子)

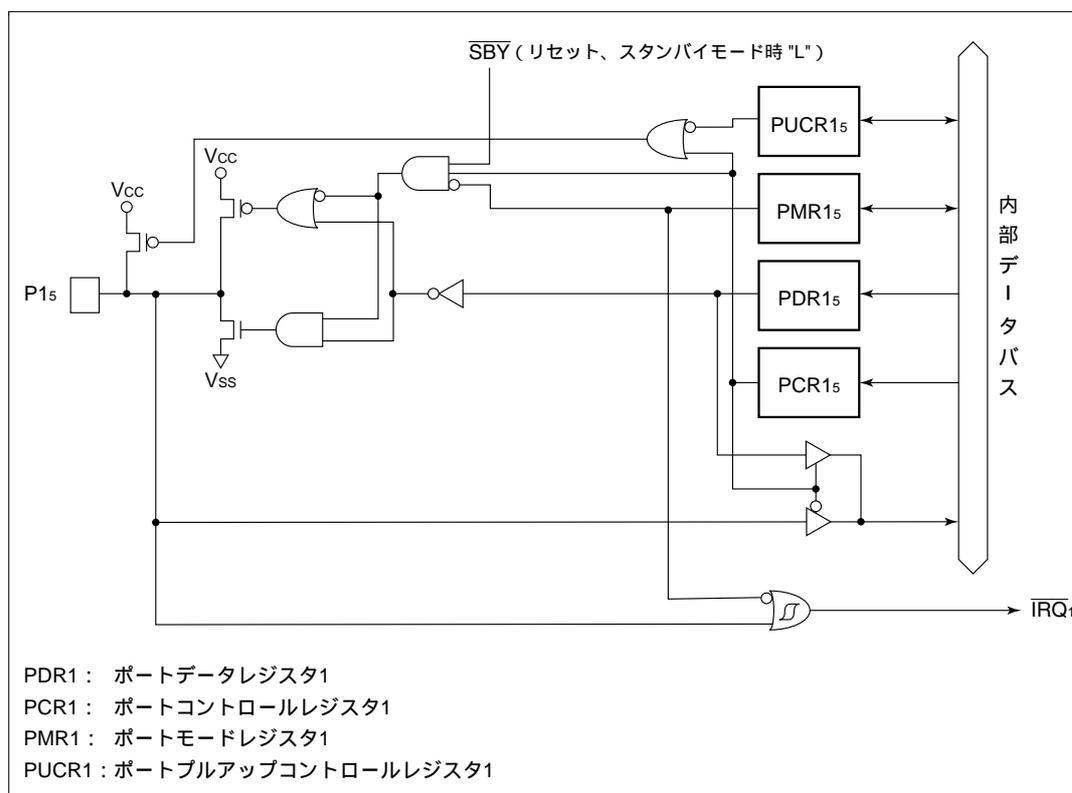


図 C.1 (c) ポート 1 ブロック図 (P1<sub>5</sub>端子)

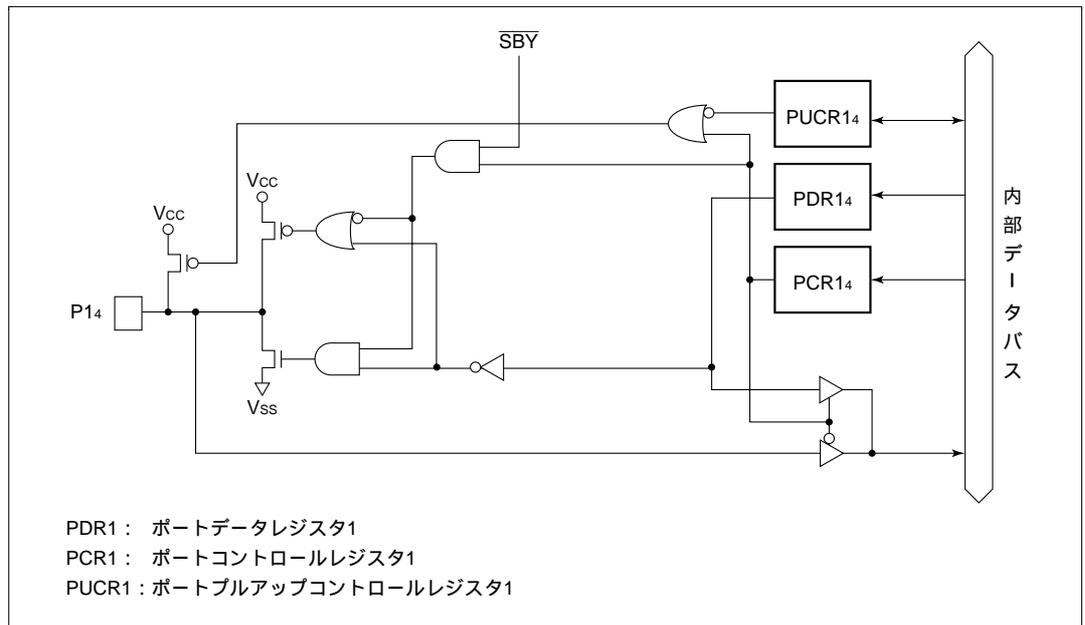


図 C.1 (d) ポート1 ブロック図 (P1<sub>4</sub>端子)

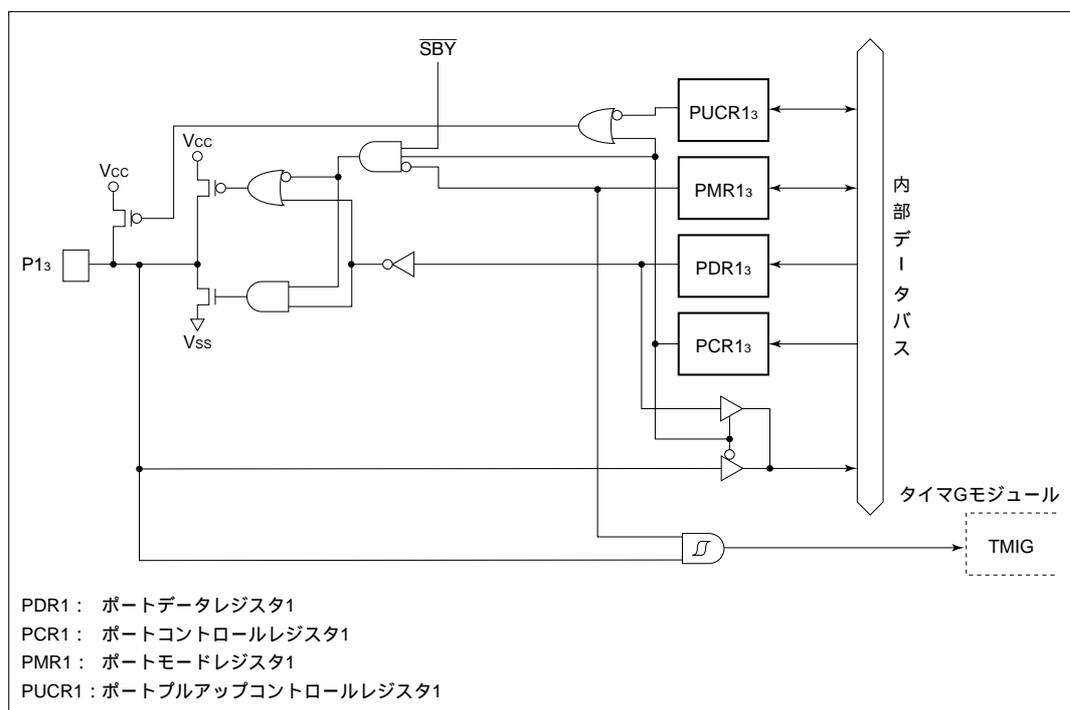


図 C.1 (e) ポート1 ブロック図 (P1<sub>3</sub>端子)

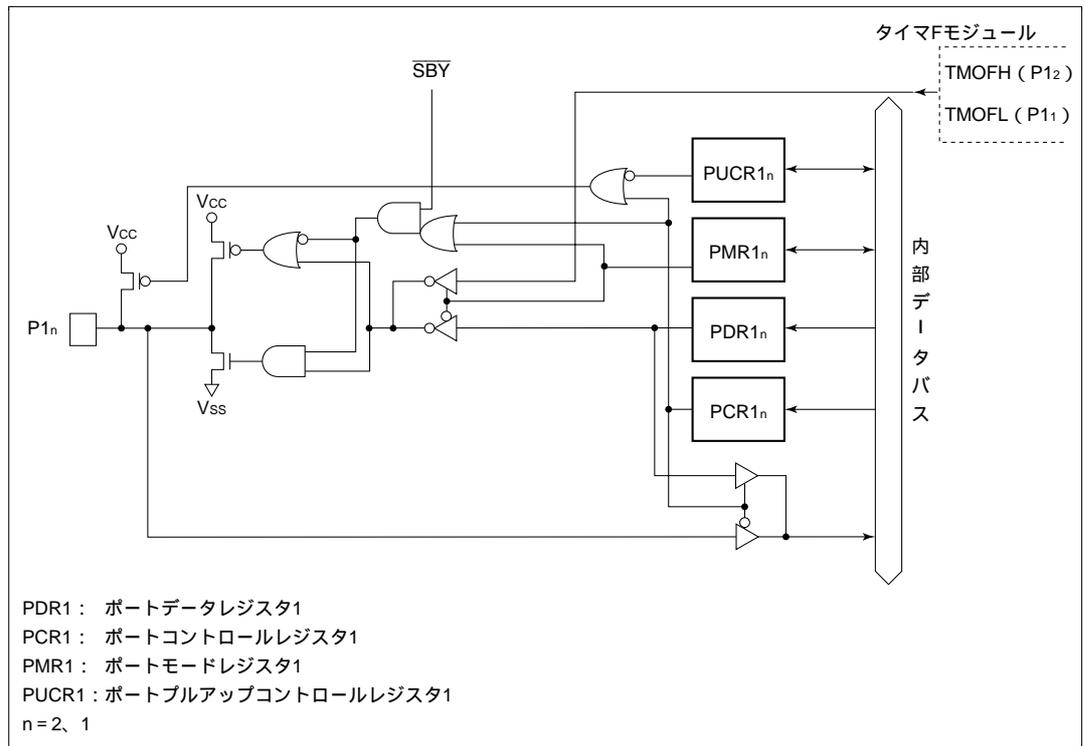


図 C.1 (f) ポート 1 ブロック図 (P1<sub>2</sub>、P1<sub>1</sub>端子)

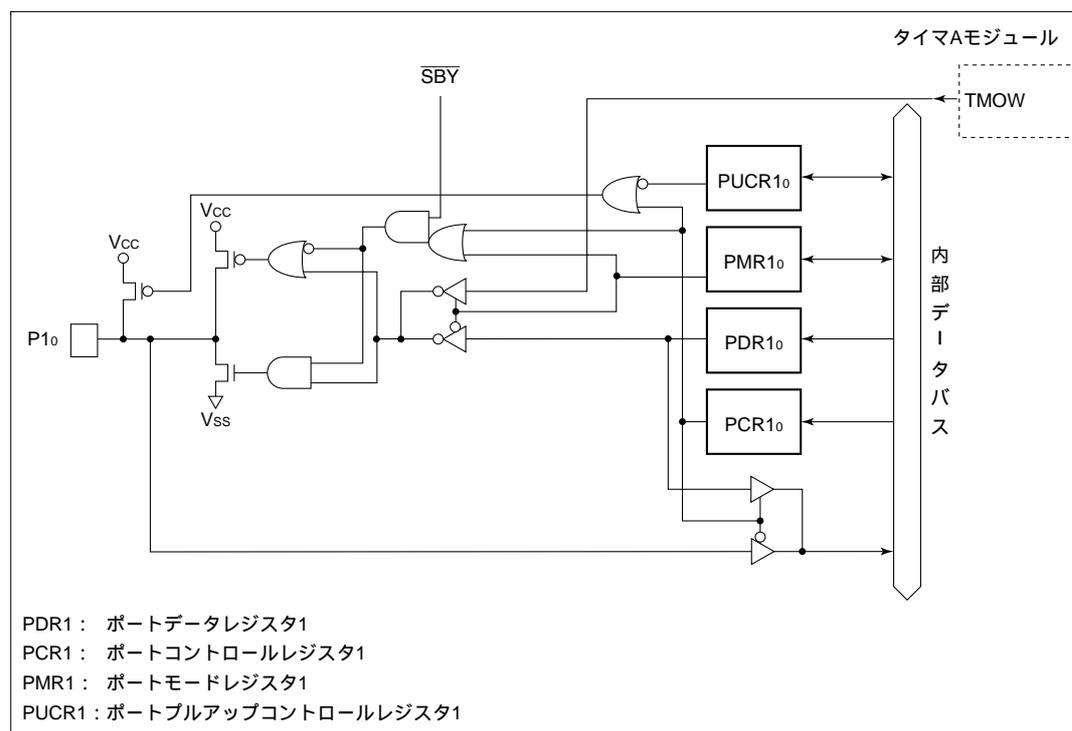


図 C.1 (g) ポート1 ブロック図 (P1<sub>0</sub>端子)

## C.2 ポート2ブロック図

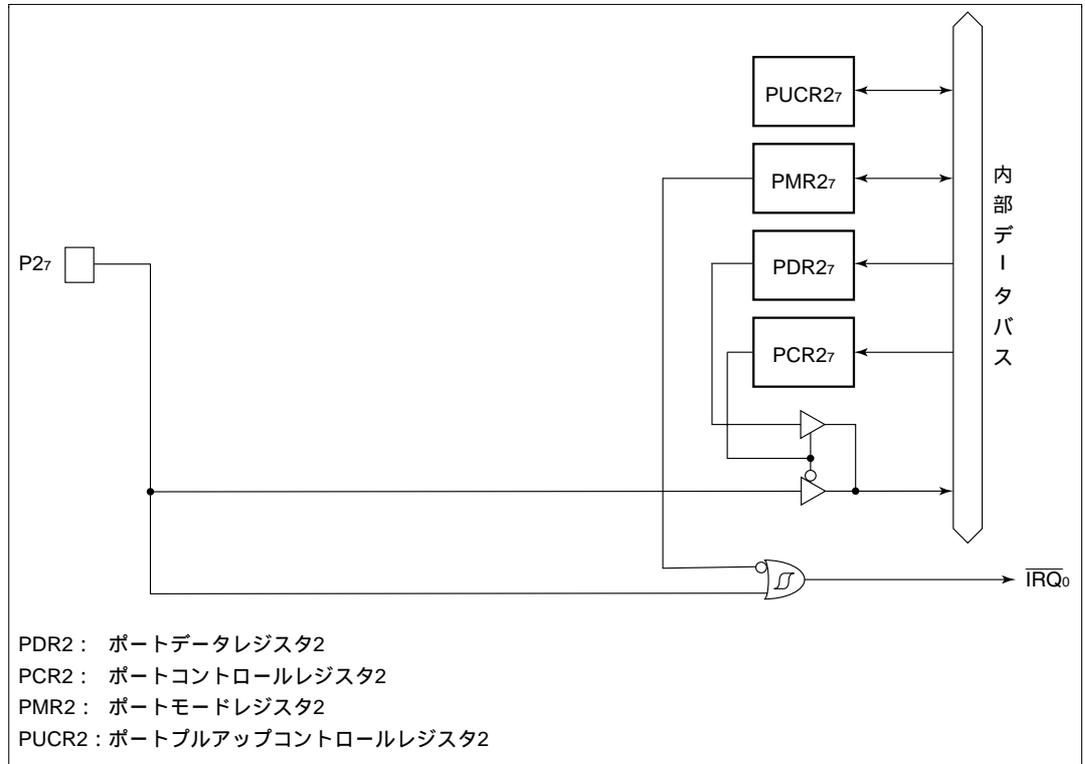


図 C.2 (a) ポート2 ブロック図 (P2<sub>7</sub>端子)

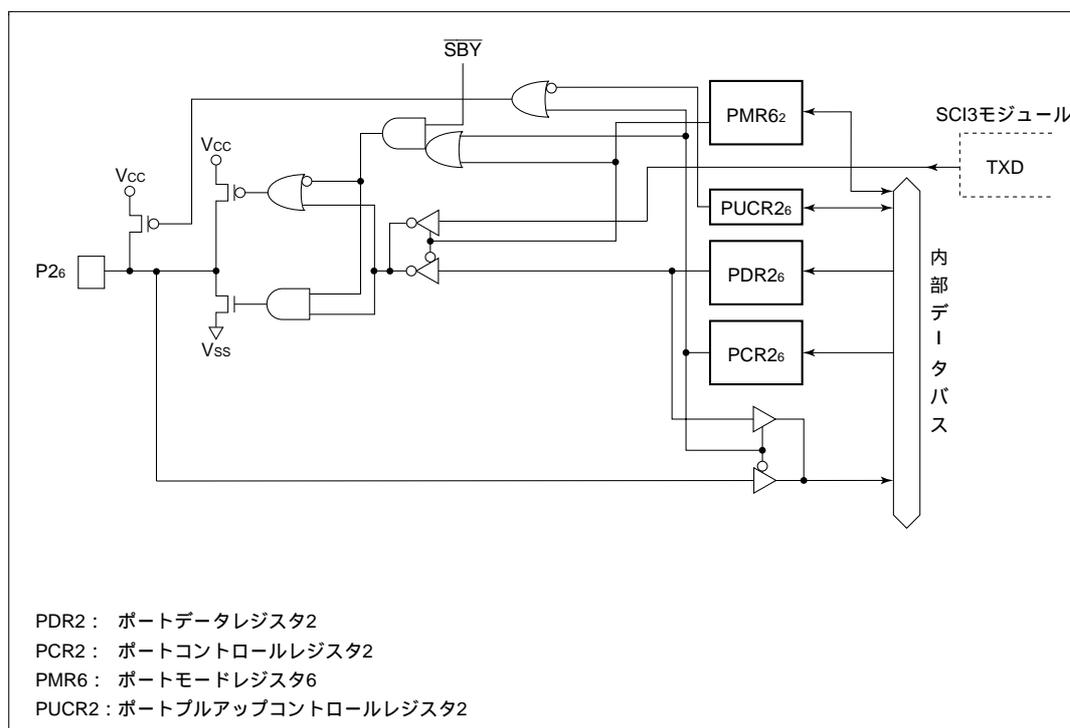


図 C.2 (b) ポート2 ブロック図 (P2<sub>6</sub>端子)

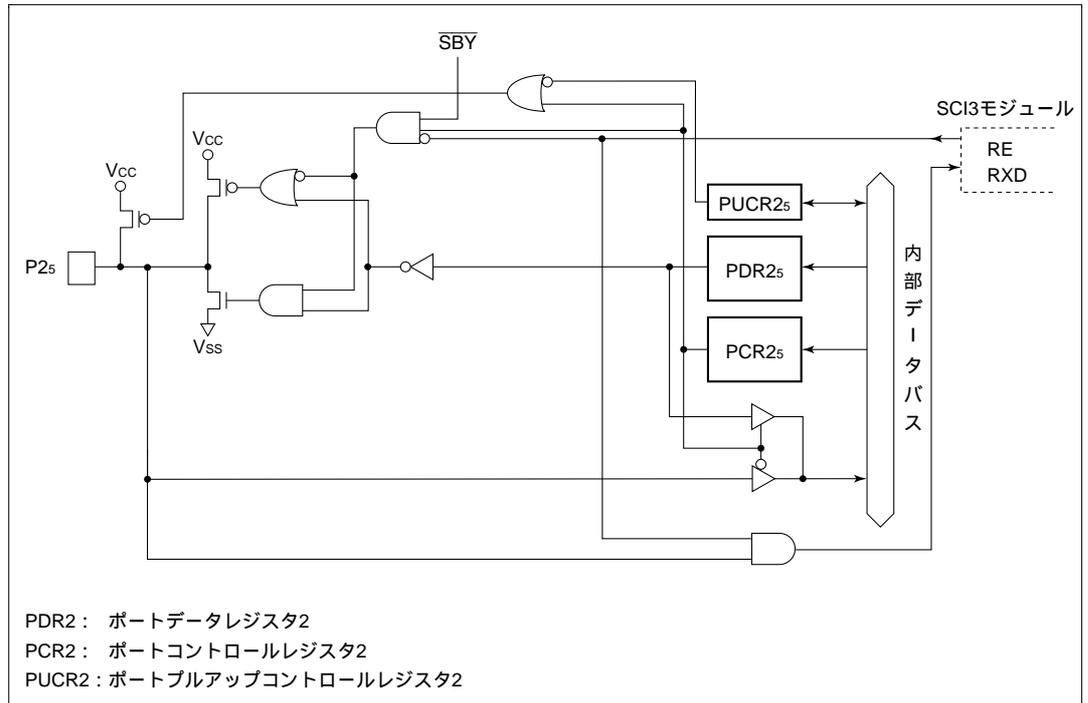


図 C.2 (c) ポート2 ブロック図 (P2<sub>5</sub>端子)

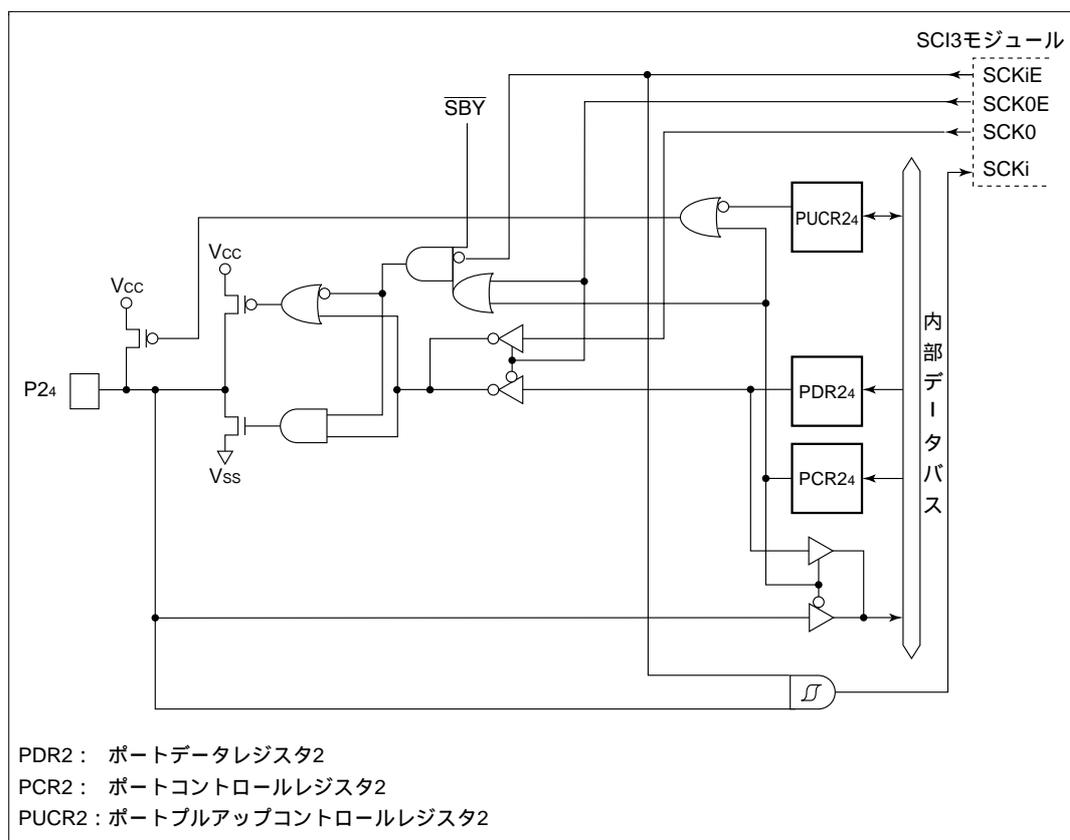


図 C.2 (d) ポート2 ブロック図 (P2<sub>4</sub>端子)

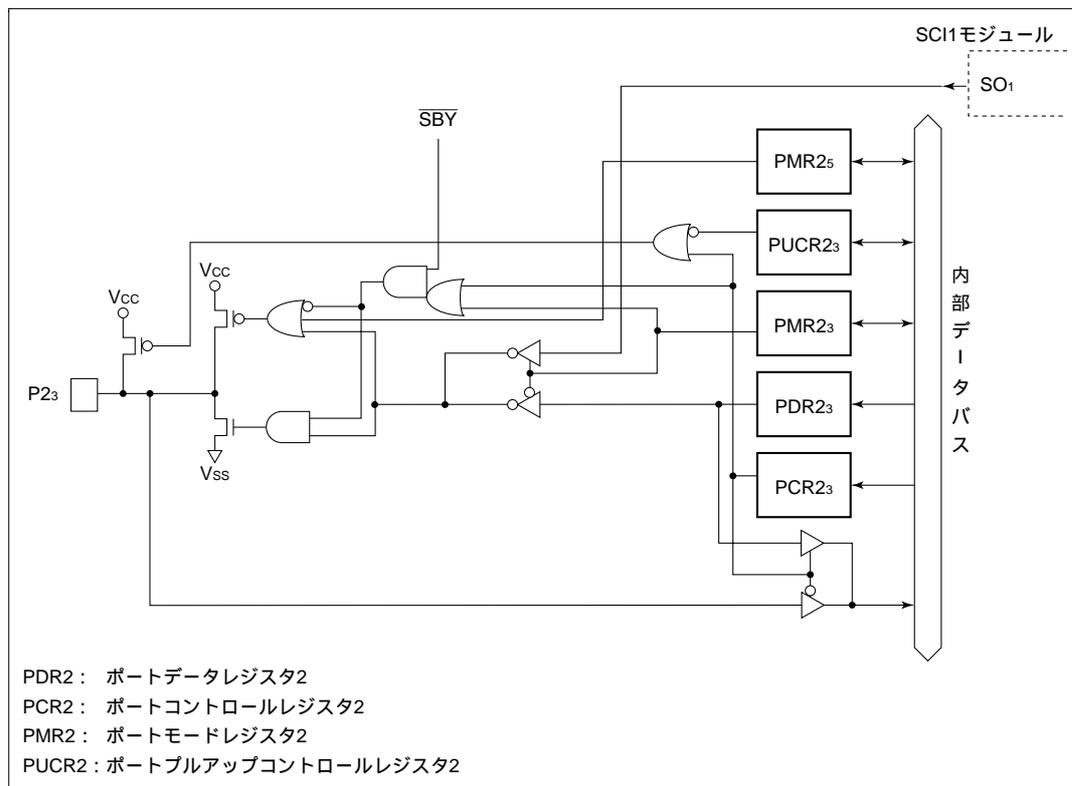


図 C.2 (e) ポート2 ブロック図 (P2<sub>3</sub>端子)

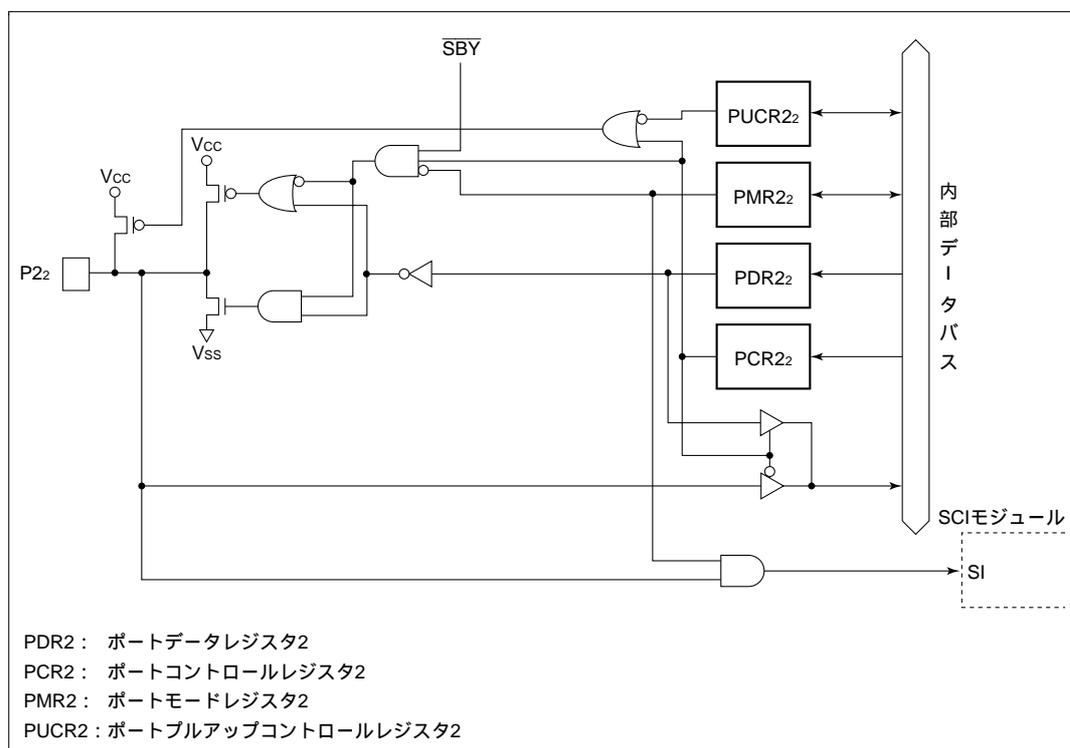


図 C.2 (f) ポート 2 ブロック図 (P2<sub>2</sub>端子)

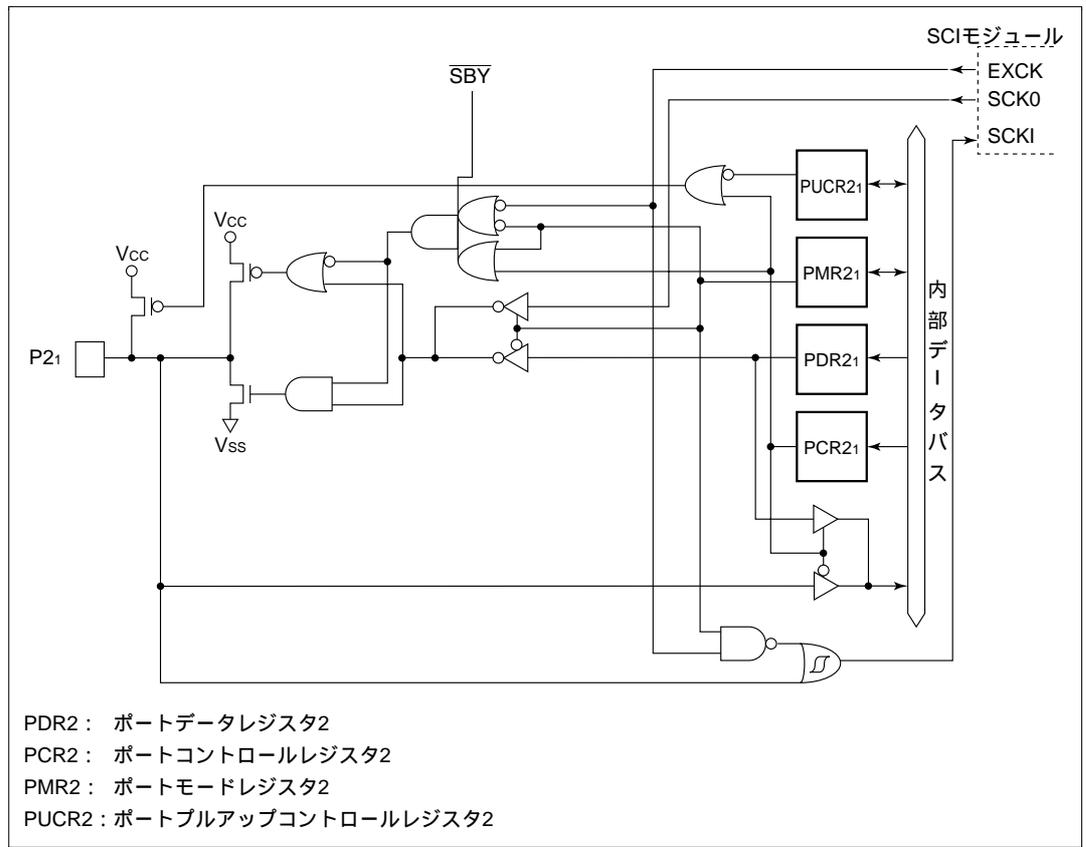


図 C.2 (g) ポート2 ブロック図 (P2<sub>1</sub>端子)

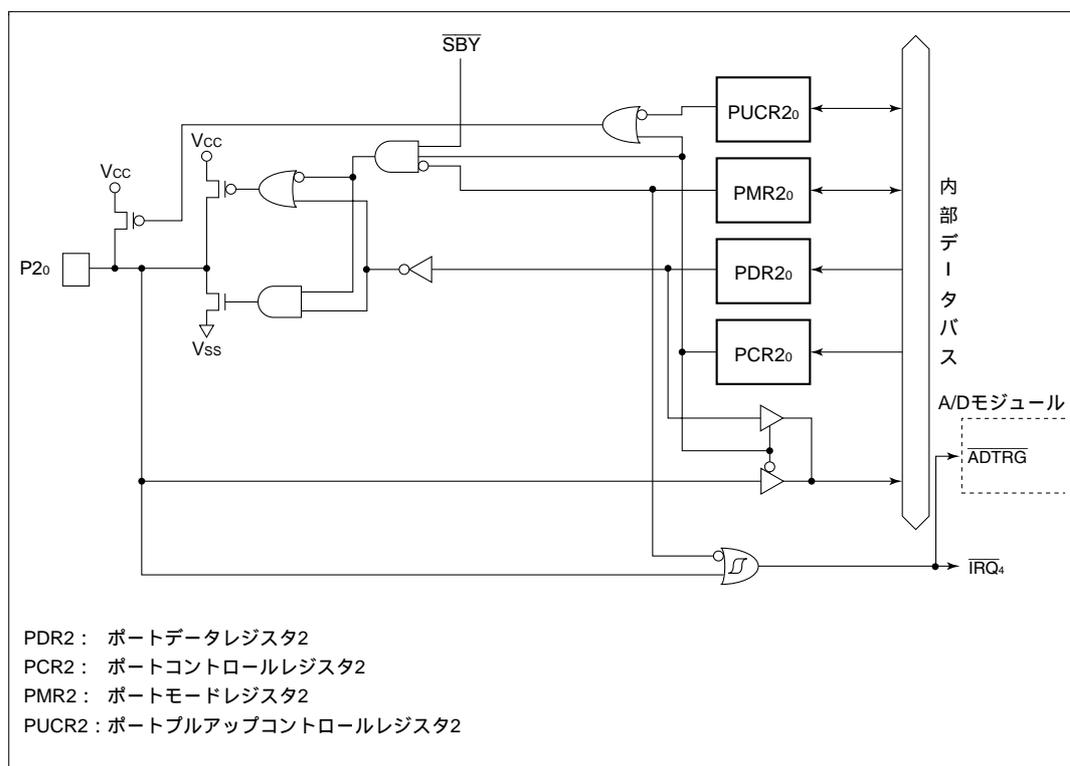


図 C.2 (h) ポート2 ブロック図 (P2<sub>0</sub>端子)

### C.3 ポート5ブロック図

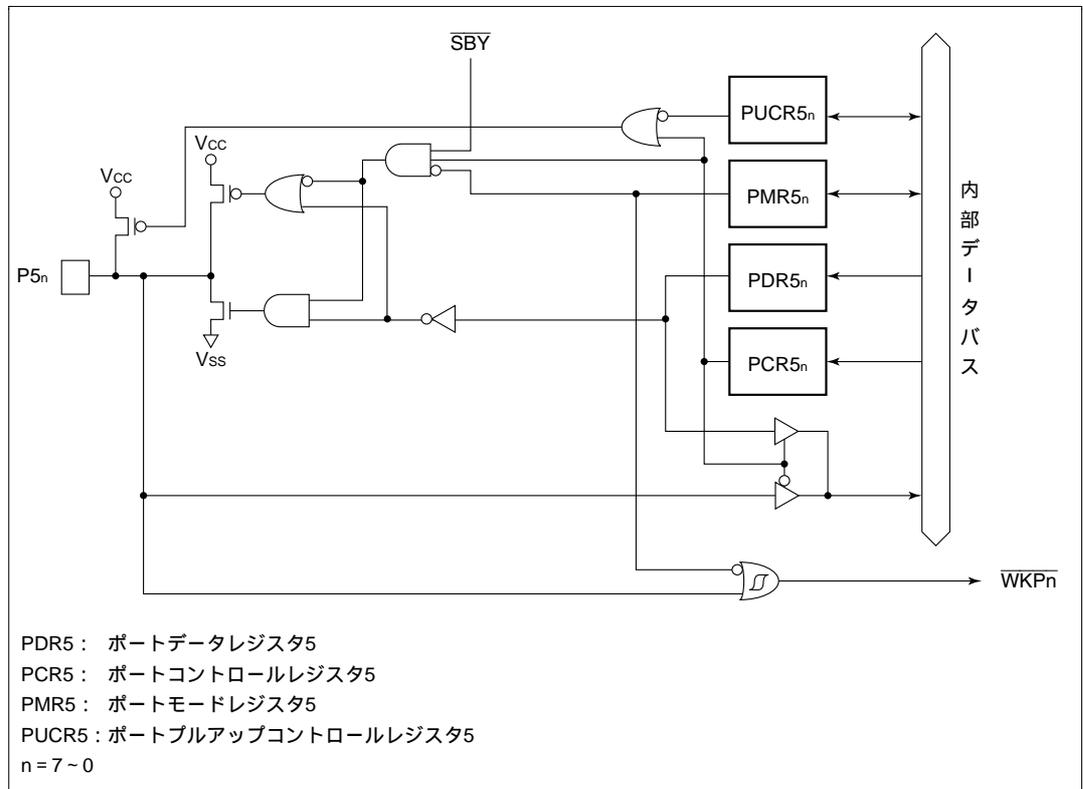


図 C.3 ポート5 ブロック図

## C.4 ポート6ブロック図

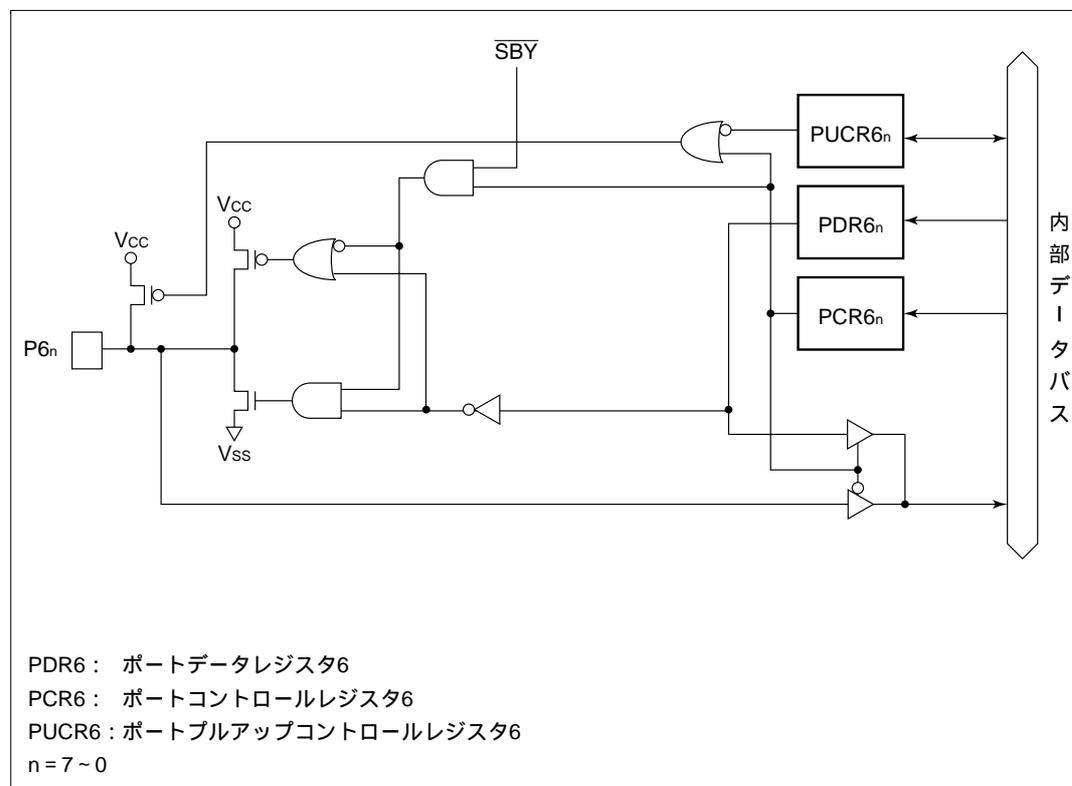


図 C.4 ポート6ブロック図

### C.5 ポート7ブロック図

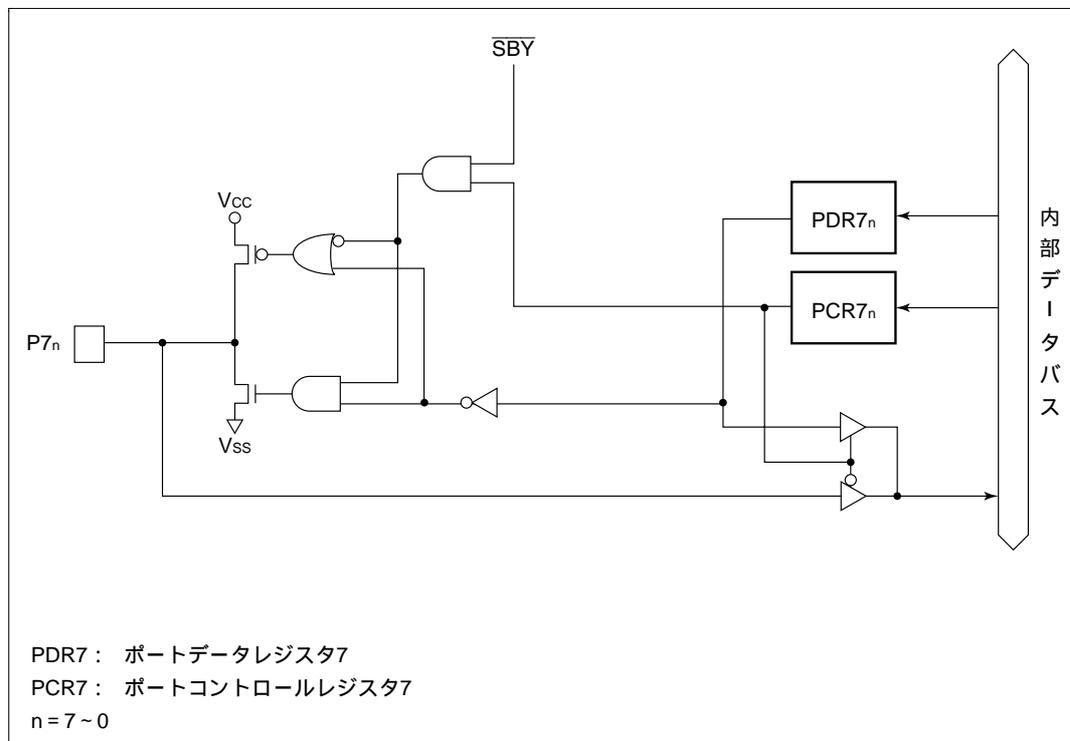


図 C.5 ポート7ブロック図

## C.6 ポート8ブロック図

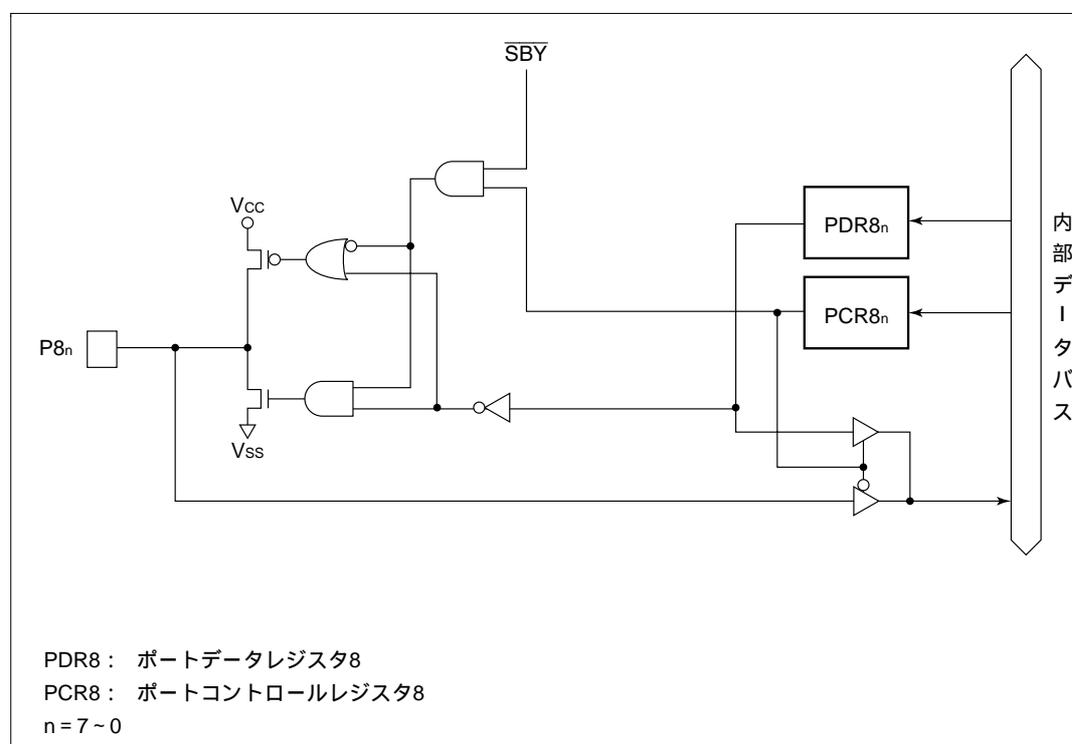


図 C.6 ポート8ブロック図

### C.7 ポートAブロック図

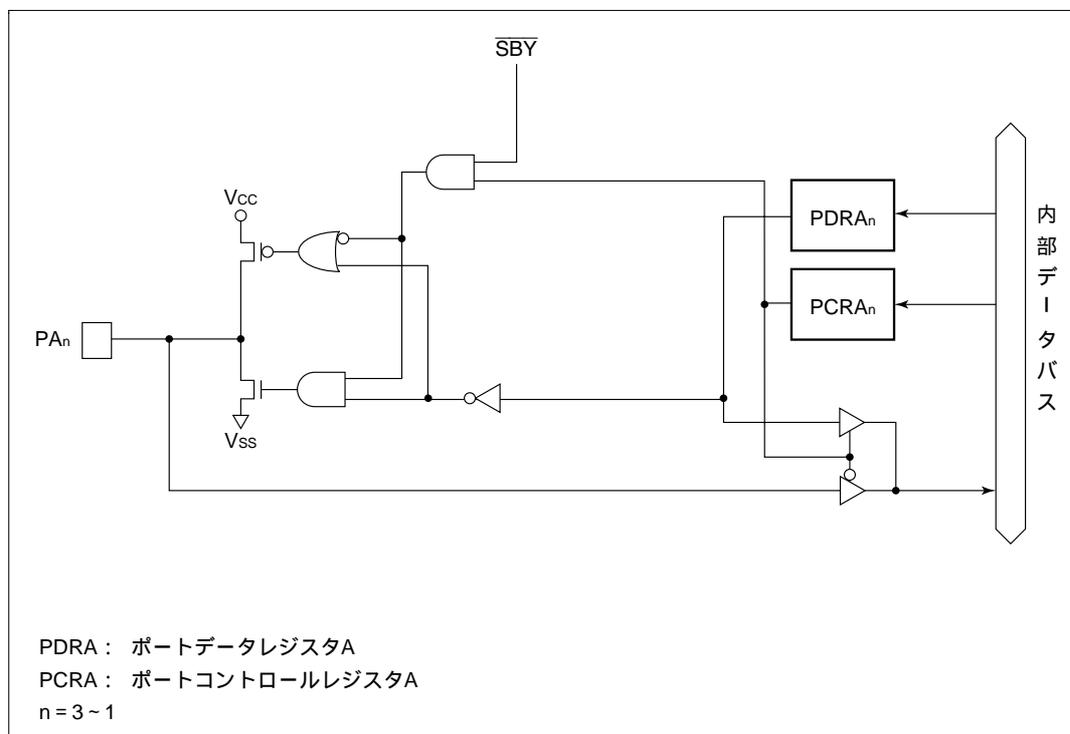


図 C.7 ポートAブロック図

### C.8 ポートBブロック図

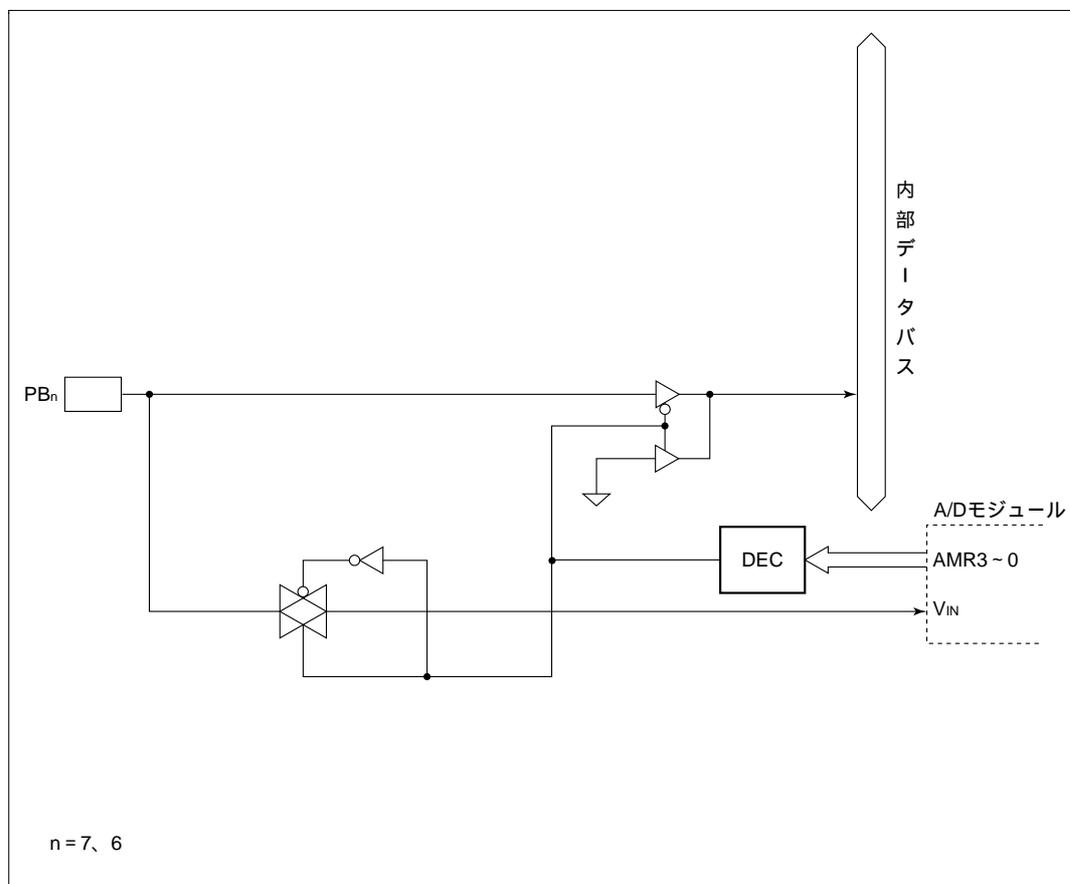


図 C.8 ポートBブロック図

## D. 各処理状態におけるポートの状態

表 D.1 各ポートの状態一覧

ポート名	リセット	スリープ	サブスリープ	スタンバイ	ウォッチ	サブアクティブ	アクティブ
P1 <sub>7</sub> ~P1 <sub>0</sub>	ハイインピ -ダンス	保持	保持	ハイインピ -ダンス*	保持	動作	動作
P2 <sub>6</sub> ~P2 <sub>0</sub>	ハイインピ -ダンス	保持	保持	ハイインピ -ダンス*	保持	動作	動作
P5 <sub>7</sub> ~P5 <sub>0</sub>	ハイインピ -ダンス	保持	保持	ハイインピ -ダンス*	保持	動作	動作
P6 <sub>7</sub> ~P6 <sub>0</sub>	ハイインピ -ダンス	保持	保持	ハイインピ -ダンス*	保持	動作	動作
P7 <sub>7</sub> ~P7 <sub>0</sub>	ハイインピ -ダンス	保持	保持	ハイインピ -ダンス	保持	動作	動作
P8 <sub>7</sub> ~P8 <sub>0</sub>	ハイインピ -ダンス	保持	保持	ハイインピ -ダンス	保持	動作	動作
PA <sub>3</sub> ~PA <sub>1</sub>	ハイインピ -ダンス	保持	保持	ハイインピ -ダンス	保持	動作	動作
P2 <sub>7</sub> 、PB <sub>7</sub> 、PB <sub>6</sub>	ハイインピ -ダンス	ハイインピ -ダンス	ハイインピ -ダンス	ハイインピ -ダンス	ハイインピ -ダンス	ハイインピ -ダンス	ハイインピ -ダンス

【注】 \* プルアップ MOS が ON 状態では"High"レベル出力となります。

## E. ROM 発注手順

### E.1 ROM 書換え品開発の流れ（発注手順）

マイコン応用システムプログラムの開発終了後、ROM データ（2 組以上）、注文仕様書、オプションリストおよびマーク仕様を一緒に提出していただきます。これにより、弊社では図 E.1 の流れ図に沿って ROM 書換え品の開発を行います。

表 E.1 に ROM 発注時に必要な提出物を示します。なお、詳細については、弊社担当営業へお問い合わせください。

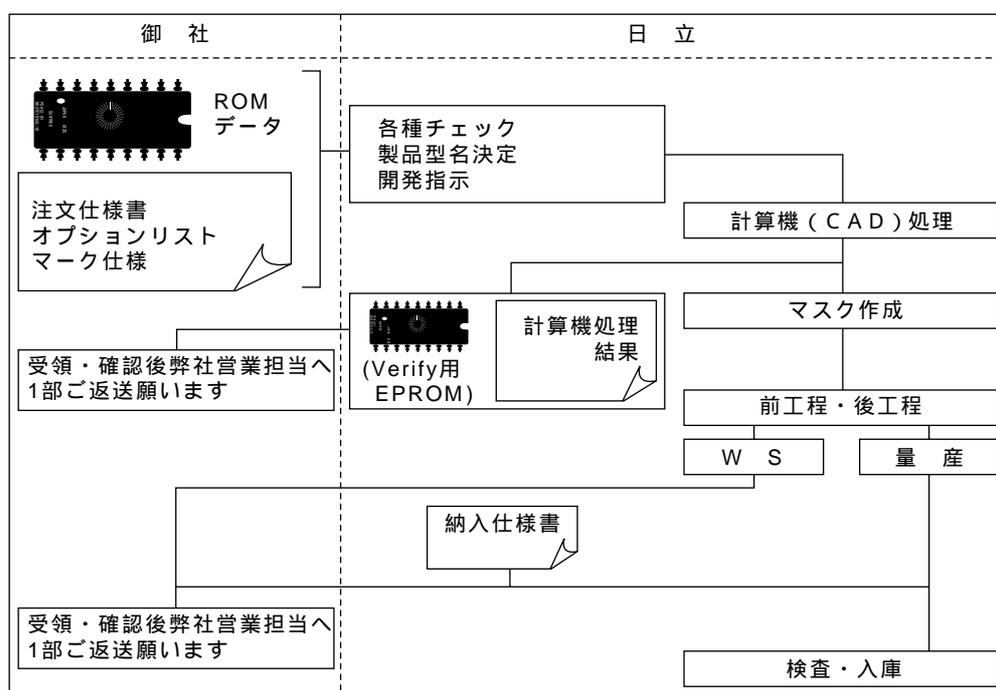


図 E.1 ROM 書き換え品開発の流れ

表 E.1 ROM 発注時に必要な提出物

発注媒体	EPROM または ZTAT <sup>®</sup> マイコン
提出物	ROM データ
	注文仕様書
	オプションリスト*1
	マーク仕様例*2

【注】 \*1 製品シリーズにより必要ないものがあります。また、内容も異なります。

\*2 特別仕様の場合には、提出してください。

## E.2 ROM 発注時の注意事項

提出していただく ROM データは、次の注意事項にしたがって、EPROMまたは ZTAT<sup>®</sup> マイコンで提出してください。なお、EPROM または ZTAT<sup>®</sup> マイコン以外の媒体（フロッピーディスク等）では対応できませんのでご注意ください。

- (1) EPROM に ROM データを書き込む際は、事前にデータを充分消去し、中途半端なレベルが出力されないことを確認してから使用してください。
- (2) 発注用 EPROM において、ROM データの未使用（NOT USED）領域またはリザーブ領域には、必ず'FF'を書き込んでください。
- (3) 提出していただく EPROM には遮光ラベルを貼り、御社の品番等を記入してください。
- (4) EPROM に書き込みを行った後は、静電気による素子の破壊、紫外線や放射線による書き込みデータの損失を招かないようにするとともに、運搬の際は導伝性のシートに梱包するなど取り扱いに充分注意してください（アルミ箔、発泡スチロール等は不可）。なお、これらによるデータの読み取りエラーに備え、同一内容の EPROM を 2 組以上提出してください。

## F. 型名一覧

表 F.1 H8/3627 シリーズ型名一覧

製品分類			製品型名	マーク型名	パッケージ (日立パッケージコード)
H8/3627	ZTAT 版	標準品	HD6473627H	HD6473627H	64 ピン QFP (FP-64A)
			HD6473627FP	HD6473627FP	64 ピン LQFP (FP-64E)
	マスク ROM 版	標準品	HD6433627H	HD6433627 (***) H	64 ピン QFP (FP-64A)
			HD6433627FP	HD6433627 (***) FP	64 ピン LQFP (FP-64E)
H8/3626	マスク	標準品	HD6433626H	HD6433626 (***) H	64 ピン QFP (FP-64A)
	ROM 版		HD6433626FP	HD6433626 (***) FP	64 ピン LQFP (FP-64E)
H8/3625	マスク	標準品	HD6s433625H	HD6433625 (***) H	64 ピン QFP (FP-64A)
	ROM 版		HD6433625FP	HD6433625 (***) FP	64 ピン LQFP (FP-64E)
H8/3624S	マスク	標準品	HD6433624SH	HD6433624S (***) H	64 ピン QFP (FP-64A)
	ROM 版		HD6433624SFP	HD6433624S (***) FP	64 ピン LQFP (FP-64E)
H8/3623S	マスク	標準品	HD6433623SH	HD6433623S (***) H	64 ピン QFP (FP-64A)
	ROM 版		HD6433623SFP	HD6433623S (***) FP	64 ピン LQFP (FP-64E)
H8/3622S	マスク	標準品	HD6433622SH	HD6433622S (***) H	64 ピン QFP (FP-64A)
	ROM 版		HD6433622SFP	HD6433622S (***) FP	64 ピン LQFP (FP-64E)

【注】 マスク ROM 版の (\*\*\*) は ROM コードです。

## G. 外形寸法図

H8/3627 シリーズの外形寸法図 FP-64A を図 G.1、FP-64E を図 G.2 に示します。

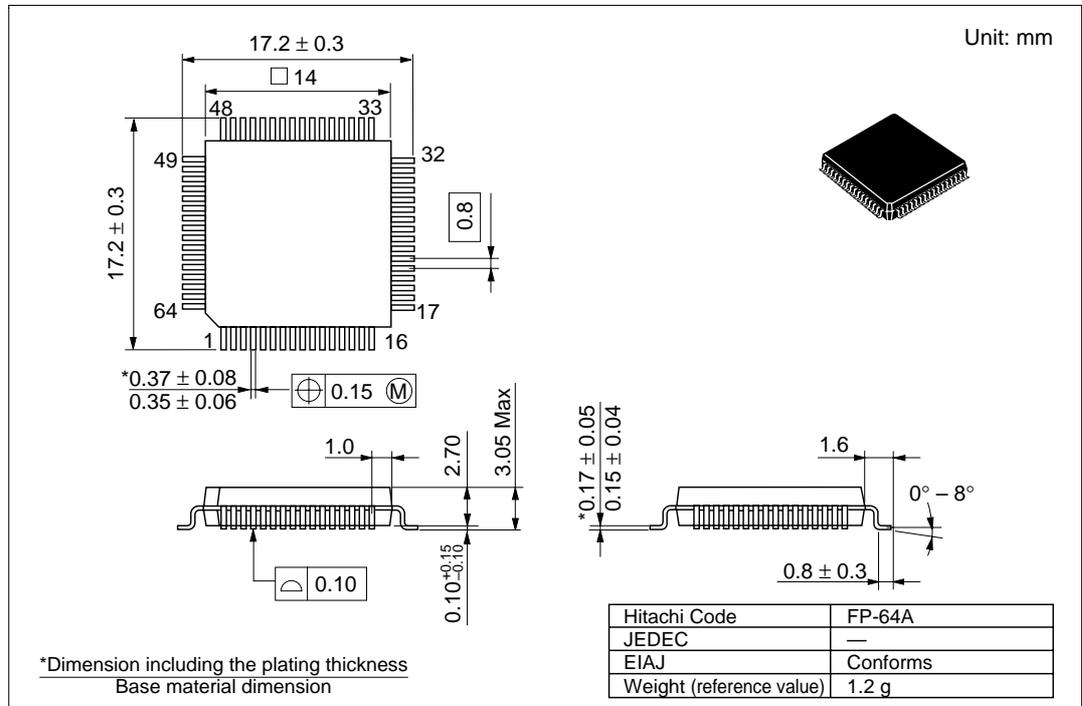


図 G.1 外形寸法図 (FP-64A)

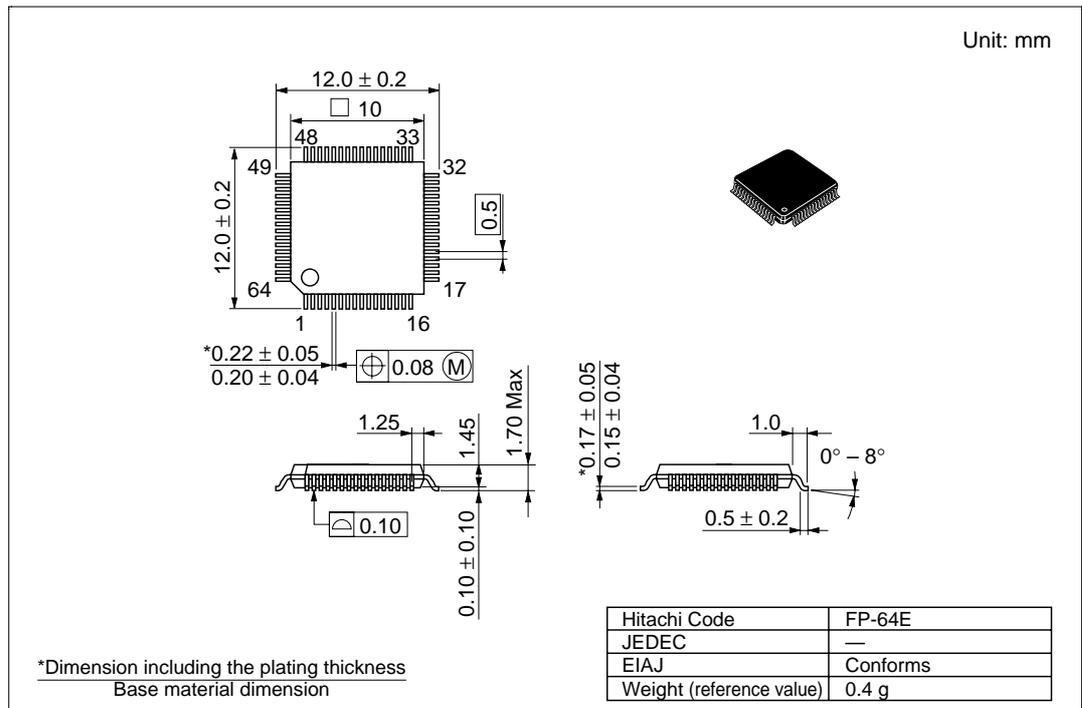


図 G.2 外形寸法図 (FP-64E)

【注】 外形寸法図については、「日立半導体パッケージ」に掲載されている寸法図を優先します。

H8/3627シリーズ ハードウェアマニュアル

発行年月 平成11年3月 第1版

発行 株式会社 日立製作所  
半導体事業本部 統括営業本部

編集 株式会社 超Lメディア  
技術ドキュメントグループ

©株式会社 日立製作所 1999

# H8/3627 シリーズ ハードウェアマニュアル



ルネサスエレクトロニクス株式会社  
神奈川県川崎市中原区下沼部1753 〒211-8668