

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い申し上げます。

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、海底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。

H8/3887 シリーズ、 H8/3847 シリーズ

ハードウェアマニュアル

ルネサスシングルチップマイクロコンピュータ

H8/3887	HD6433887
	HD6473887
H8/3886	HD6433886
H8/3885	HD6433885
H8/3884	HD6433884
H8/3883	HD6433883
H8/3882	HD6433882
H8/3847	HD6433847
	HD6473847
H8/3846	HD6433846
H8/3845	HD6433845
H8/3844	HD6433844
H8/3843	HD6433843
H8/3842	HD6433842

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。

H8/3887 シリーズと H8/3847 シリーズは、システム構成に必要な周辺機能として、LCD コントローラ/ドライバ、6 種類のタイマ、14 ビット PWM、3 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。LCD 表示を必要とするシステムの組込み用マイコンとして活用できます。

H8/3887 シリーズは、LCD の駆動電源として昇圧定電圧 (5V) 電源を内蔵しており V_{CC} に依存せず 5V 一定電圧を得ることができます。

本マニュアルは、H8/3887、H8/3847 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	7	
1.3	端子説明	8	
	1.3.1	ピン配置	8
	1.3.2	端子機能	10

第2章 CPU

2.1	概要	17	
	2.1.1	特長	17
	2.1.2	アドレス空間	18
	2.1.3	レジスタ構成	18
2.2	各レジスタの説明	19	
	2.2.1	汎用レジスタ	19
	2.2.2	コントロールレジスタ	19
	2.2.3	CPU内部レジスタの初期値	21
2.3	データ構成	22	
	2.3.1	汎用レジスタのデータ構成	22
	2.3.2	メモリ上でのデータ構成	23
2.4	アドレッシングモード	24	
	2.4.1	アドレッシングモード	24
	2.4.2	実効アドレスの計算方法	26
2.5	命令セット	30	
	2.5.1	データ転送命令	32
	2.5.2	算術演算命令	34
	2.5.3	論理演算命令	35
	2.5.4	シフト命令	35

2.5.5	ビット操作命令	37
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	44
2.6	基本動作タイミング	45
2.6.1	内蔵メモリ (RAM、ROM)	45
2.6.2	内蔵周辺モジュール	46
2.7	CPUの状態	48
2.7.1	概要	48
2.7.2	プログラム実行状態	49
2.7.3	プログラム停止状態	49
2.7.4	例外処理状態	49
2.8	メモリマップ	50
2.8.1	メモリマップ	50
2.9	使用上の注意事項	57
2.9.1	データアクセスに関する注意事項	57
2.9.2	ビット操作命令使用上の注意事項	59
2.9.3	EEPMOV 命令使用上の注意事項	65

第3章 例外処理

3.1	概要	69
3.2	リセット	70
3.2.1	概要	70
3.2.2	リセットシーケンス	70
3.2.3	リセット直後の割込み	72
3.3	割込み	73
3.3.1	概要	73
3.3.2	各レジスタの説明	75
3.3.3	外部割込み	85
3.3.4	内部割込み	86
3.3.5	割込み動作	86
3.3.6	割込み応答時間	91
3.4	使用上の注意事項	92
3.4.1	スタック領域に関する使用上の注意事項	92
3.4.2	ポートモードレジスタを書換える際の注意事項	93

第4章 クロック発振器

4.1	概要.....	99
	4.1.1 ブロック図.....	99
	4.1.2 システムクロックとサブクロック	99
4.2	システムクロック発振器.....	100
4.3	サブクロック発振器	103
4.4	プリスケータ	105
4.5	発振子に関する注意事項.....	106

第5章 低消費電力モード

5.1	概要.....	109
	5.1.1 システムコントロールレジスタ.....	112
5.2	スリープモード.....	116
	5.2.1 スリープモードへの遷移.....	116
	5.2.2 スリープモードの解除.....	116
	5.2.3 スリープ(中速)モードの動作周波数について.....	117
5.3	スタンバイモード.....	118
	5.3.1 スタンバイモードへの遷移	118
	5.3.2 スタンバイモードの解除.....	118
	5.3.3 スタンバイモード解除後の発振安定時間の設定	119
	5.3.4 スタンバイモードへの遷移と端子状態.....	120
5.4	ウォッチモード.....	121
	5.4.1 ウォッチモードへの遷移.....	121
	5.4.2 ウォッチモードの解除.....	121
	5.4.3 ウォッチモード解除後の発振安定時間の設定.....	121
5.5	サブスリープモード	122
	5.5.1 サブスリープモードへの遷移.....	122
	5.5.2 サブスリープモードの解除	122
5.6	サブアクティブモード.....	123
	5.6.1 サブアクティブモードへの遷移.....	123
	5.6.2 サブアクティブモードの解除.....	123
	5.6.3 サブアクティブモードの動作周波数について.....	123
5.7	アクティブ(中速)モード	124
	5.7.1 アクティブ(中速)モードへの遷移.....	124
	5.7.2 アクティブ(中速)モードの解除	124
	5.7.3 アクティブ(中速)モードの動作周波数について.....	124
5.8	直接遷移.....	125

5.8.1	直接遷移の概要	125
5.8.2	直接遷移の時間	126
5.9	モジュールスタンバイモード	129
5.9.1	モジュールスタンバイモードの設定	129
5.9.2	モジュールスタンバイモードの解除	129

第6章 ROM

6.1	概要	133
6.1.1	ブロック図	133
6.2	H8/3887、H8/3847 の PROM モード	134
6.2.1	PROM モードの設定	134
6.2.2	ソケットアダプタの端子対応とメモリマップ	134
6.3	H8/3887、H8/3847 のプログラミング	137
6.3.1	書込み/ベリファイ	138
6.3.2	書込み時の注意	140
6.4	書込み後の信頼性	142

第7章 RAM

7.1	概要	145
7.1.1	ブロック図	145

第8章 I/O ポート

8.1	概要	149
8.2	ポート 1	151
8.2.1	概要	151
8.2.2	レジスタの構成と説明	151
8.2.3	端子機能	156
8.2.4	端子状態	158
8.2.5	入力プルアップ MOS	158
8.3	ポート 2	159
8.3.1	概要	159
8.3.2	レジスタの構成と説明	159
8.3.3	端子機能	163
8.3.4	端子状態	163
8.4	ポート 3	164

	8.4.1	概要.....	164
	8.4.2	レジスタの構成と説明.....	164
	8.4.3	端子機能.....	168
	8.4.4	端子状態.....	169
	8.4.5	入力プルアップ MOS.....	169
8.5	ポート4.....		170
	8.5.1	概要.....	170
	8.5.2	レジスタの構成と説明.....	170
	8.5.3	端子機能.....	171
	8.5.4	端子状態.....	172
8.6	ポート5.....		173
	8.6.1	概要.....	173
	8.6.2	レジスタの構成と説明.....	173
	8.6.3	端子機能.....	175
	8.6.4	端子状態.....	176
	8.6.5	入力プルアップ MOS.....	176
8.7	ポート6.....		177
	8.7.1	概要.....	177
	8.7.2	レジスタの構成と説明.....	177
	8.7.3	端子機能.....	179
	8.7.4	端子状態.....	179
	8.7.5	入力プルアップ MOS.....	179
8.8	ポート7.....		180
	8.8.1	概要.....	180
	8.8.2	レジスタの構成と説明.....	180
	8.8.3	端子機能.....	181
	8.8.4	端子状態.....	181
8.9	ポート8.....		182
	8.9.1	概要.....	182
	8.9.2	レジスタの構成と説明.....	182
	8.9.3	端子機能.....	183
	8.9.4	端子状態.....	183
8.10	ポート9.....		184
	8.10.1	概要.....	184
	8.10.2	レジスタの構成と説明.....	184
	8.10.3	端子機能.....	185
	8.10.4	端子状態.....	186
8.11	ポートA.....		187

	8.11.1	概要.....	187
	8.11.2	レジスタの構成と説明.....	187
	8.11.3	端子機能.....	189
	8.11.4	端子状態.....	189
8.12	ポート B		190
	8.12.1	概要.....	190
	8.12.2	レジスタの構成と説明.....	190
8.13	ポート C		191
	8.13.1	概要.....	191
	8.13.2	レジスタの構成と説明.....	191
8.14	入出力データ反転機能.....		192
	8.14.1	概要.....	192
	8.14.2	レジスタの構成と説明.....	192
	8.14.3	シリアルポートコントロールレジスタを書換える際の注意事項.....	194

第9章 タイマ

9.1	概要.....		197
9.2	タイマ A		198
	9.2.1	概要.....	198
	9.2.2	各レジスタの説明.....	200
	9.2.3	動作説明.....	204
	9.2.4	タイマ A の動作モード.....	205
9.3	タイマ C		206
	9.3.1	概要.....	206
	9.3.2	各レジスタの説明.....	208
	9.3.3	動作説明.....	212
	9.3.4	タイマ C の動作モード.....	213
9.4	タイマ F.....		214
	9.4.1	概要.....	214
	9.4.2	各レジスタの説明.....	217
	9.4.3	CPU とのインタフェース	225
	9.4.4	動作説明.....	227
	9.4.5	使用上の注意事項.....	230
9.5	タイマ G		232
	9.5.1	概要.....	232
	9.5.2	各レジスタの説明.....	234
	9.5.3	ノイズ除去回路.....	239

	9.5.4	動作説明.....	240
	9.5.5	使用上の注意事項.....	245
	9.5.6	タイマ G の使用例.....	249
9.6		ウォッチドッグタイマ.....	250
	9.6.1	概要.....	250
	9.6.2	各レジスタの説明.....	251
	9.6.3	動作説明.....	255
	9.6.4	ウォッチドッグタイマの動作モード.....	256
9.7		非同期イベントカウンタ (AEC).....	257
	9.7.1	概要.....	257
	9.7.2	各レジスタの構成.....	259
	9.7.3	動作説明.....	264
	9.7.4	非同期イベントカウンタの動作モード.....	265
	9.7.5	使用上の注意事項.....	266

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.....	271
10.2		SCI1.....	272
	10.2.1	概要.....	272
	10.2.2	各レジスタの説明.....	274
	10.2.3	動作説明.....	280
	10.2.4	SSB モード時の動作説明.....	283
	10.2.5	割込み要因.....	285
	10.2.6	使用上の注意事項.....	285
10.3		SCI3.....	286
	10.3.1	概要.....	286
	10.3.2	各レジスタの説明.....	290
	10.3.3	動作説明.....	309
	10.3.4	割込み要因.....	333
	10.3.5	使用上の注意事項.....	334

第 11 章 14 ビット PWM

11.1		概要.....	341
	11.1.1	特長.....	341
	11.1.2	ブロック図.....	341
	11.1.3	端子構成.....	342

11.1.4	レジスタ構成	342
11.2	各レジスタの説明.....	343
11.2.1	PWM コントロールレジスタ (PWCR)	343
11.2.2	PWM データレジスタ U、L (PWDRU、PWDRL)	344
11.2.3	クロック停止レジスタ 2 (CKSTPR2)	344
11.3	動作説明.....	345
11.3.1	動作説明.....	345
11.3.2	PWM の動作モード.....	346

第 12 章 A/D 変換器

12.1	概要.....	349
12.1.1	特長.....	349
12.1.2	ブロック図.....	350
12.1.3	端子構成	351
12.1.4	レジスタ構成	351
12.2	各レジスタの説明.....	352
12.2.1	A/D リザルトレジスタ (ADRRH、ADRRL)	352
12.2.2	A/D モードレジスタ (AMR)	352
12.2.3	A/D スタートレジスタ (ADSR)	354
12.2.4	クロック停止レジスタ 1 (CKSTPR1)	355
12.3	動作説明.....	356
12.3.1	A/D 変換動作.....	356
12.3.2	外部トリガによる A/D 変換器の起動.....	356
12.3.3	A/D 変換器の動作モード.....	356
12.4	割込み要因.....	357
12.5	使用例	358
12.6	使用上の注意	362

第 13 章 LCD コントローラ / ドライバ

13.1	概要.....	365
13.1.1	特長.....	365
13.1.2	ブロック図.....	366
13.1.3	端子構成	367
13.1.4	レジスタ構成	367
13.2	各レジスタの説明.....	368
13.2.1	LCD ポートコントロールレジスタ (LPCR)	368

13.2.2	LCD コントロールレジスタ (LCR)	370
13.2.3	LCD コントロールレジスタ 2 (LCR2)	372
13.2.4	クロック停止レジスタ 2 (CKSTPR2)	374
13.3	動作説明	375
13.3.1	LCD 表示までのセッティング	375
13.3.2	LCD RAM と表示の関係	377
13.3.3	輝度調整機能 (V ₀ 端子)	382
13.3.4	昇圧定電圧 (5V) 電源	383
13.3.5	低消費電力 LCD 駆動方式	383
13.3.6	低消費電力モード時の動作	387
13.3.7	LCD 駆動電源の強化	388
13.3.8	HD66100 との接続	388

第 14 章 電源回路

14.1	概要	393
14.2	内部電源降圧回路を使用する場合	393
14.3	内部電源降圧回路を使用しない場合	394

第 15 章 電気的特性

15.1	H8/3887、H8/3847 シリーズ絶対最大定格	397
15.2	H8/3887、H8/3847 シリーズ電気的特性	398
15.2.1	電源電圧と動作範囲	398
15.2.2	DC 特性	401
15.2.3	AC 特性	407
15.2.4	A/D 変換器特性	410
15.2.5	LCD 特性	411
15.3	動作タイミング	412
15.4	出力負荷回路	416
15.5	発振子の等価回路	417

付録

A.	命令	421
A.1	命令一覧	421
A.2	オペレーションコードマップ	431
A.3	命令実行ステート数	433

B.	内部 I/O レジスタ一覧.....	439
	B.1 アドレス一覧.....	439
	B.2 機能一覧.....	443
C.	I/O ポートブロック図.....	497
	C.1 ポート 1 ブロック図.....	497
	C.2 ポート 2 ブロック図.....	499
	C.3 ポート 3 ブロック図.....	501
	C.4 ポート 4 ブロック図.....	505
	C.5 ポート 5 ブロック図.....	507
	C.6 ポート 6 ブロック図.....	508
	C.7 ポート 7 ブロック図.....	509
	C.8 ポート 8 ブロック図.....	510
	C.9 ポート 9 ブロック図.....	511
	C.10 ポート A ブロック図.....	512
	C.11 ポート B ブロック図.....	513
	C.12 ポート C ブロック図.....	514
D.	各処理状態におけるポートの状態	515
E.	ROM 発注手順.....	516
	E.1 ROM 書換え品開発の流れ (発注手順)	516
	E.2 ROM 発注時の注意事項	517
F.	製品型名一覧	518
G.	外形寸法図.....	520

1. 概要

第1章 目次

1.1	概要.....	3
1.2	内部ブロック図.....	7
1.3	端子説明.....	8
	1.3.1 ピン配置.....	8
	1.3.2 端子機能.....	10

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。

H8/3887、H8/3847 シリーズは、LCD（Liquid Crystal Display）コントローラ/ドライバを内蔵した 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、周辺機能として、LCD コントローラ/ドライバ、6 種類のタイマ、14 ビット PWM、3 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おり、低消費電力、LCD 表示を必要とするシステムの組込み用マイコンに最適な構成となっています。H8/3887、H8/3847 シリーズには、16k バイトの ROM、1k バイトの RAM を内蔵した H8/3882、H8/3842、24k バイトの ROM、1k バイトの RAM を内蔵した H8/3883、H8/3843、32k バイトの ROM、2k バイトの RAM を内蔵した H8/3884、H8/3844、40k バイトの ROM、2k バイトの RAM を内蔵した H8/3885、H8/3845、48k バイトの ROM、2k バイトの RAM を内蔵した H8/3886、H8/3846、および 60k バイトの ROM、2k バイトの RAM を内蔵した H8/3887、H8/3847 があります。

H8/3887、H8/3847 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[®]*版もあります。

H8/3887、H8/3847 シリーズの特長を表 1.1 に示します。

【注】 * ZTAT（Zero Turn Around Time）は（株）日立製作所の登録商標です。

表 1.1 特長

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：3MHz ・加減算：0.67 μ s (= 3MHz 動作時) ・乗除算：4.67 μ s (= 3MHz 動作時) ・32.768kHz および 38.4kHz 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

1. 概要

項目	仕様
CPU	(4) 特長ある命令 <ul style="list-style-type: none"> ・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	37 種類の割込み要因 <ul style="list-style-type: none"> ・外部割込み要因：13 要因 (IRQ₄~IRQ₀、WKP₇~WKP₀) ・内部割込み要因：24 要因
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"> ・システムクロック発振器：0.4~6MHz ・サブクロック発振器：32.768kHz、38.4kHz
低消費電力モード	7 種類の低消費電力モードおよびモジュールスタンバイモード <ul style="list-style-type: none"> ・スリープ (高速) モード ・スリープ (中速) モード ・スタンバイモード ・ウォッチモード ・サブスリープモード ・サブアクティブモード ・アクティブ (中速) モード
メモリ	大容量メモリ内蔵 H8/3882、H8/3842・ROM：16k バイト・RAM：1k バイト H8/3883、H8/3843・ROM：24k バイト・RAM：1k バイト H8/3884、H8/3844・ROM：32k バイト・RAM：2k バイト H8/3885、H8/3845・ROM：40k バイト・RAM：2k バイト H8/3886、H8/3846・ROM：48k バイト・RAM：2k バイト H8/3887、H8/3847・ROM：60k バイト・RAM：2k バイト
I/O ポート	I/O ポート 84 本 <ul style="list-style-type: none"> ・入出力端子：71 本 ・入力端子：13 本
タイマ	6 種類のタイマ内蔵 <p>(1) タイマ A：8 ビットのタイマ</p> <ul style="list-style-type: none"> ・システムクロック (f_{clk}) * 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 (f_{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

【注】 * f_{clk} 、 f_{w} の定義は「第 4 章 クロック発振器」を参照してください。

項目	仕様
タイマ	<p>(2) 非同期イベントカウンタ：16ビットのタイマ</p> <ul style="list-style-type: none"> ・マイコンの内部クロックと無関係に非同期外部イベントをカウントアップ可能 <p>(3) タイマC：8ビットのタイマ</p> <ul style="list-style-type: none"> ・7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/ダウン可能 ・オートリロード機能可能 <p>(4) タイマF：16ビットのタイマ</p> <ul style="list-style-type: none"> ・独立した2本の8ビットタイマとして使用可能 ・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 ・コンペアマッチ機能によりトグル出力可能 <p>(5) タイマG：8ビットのタイマ</p> <ul style="list-style-type: none"> ・4種類の内部クロックによりカウントアップ可能 ・インプットキャプチャ機能内蔵（ノイズ除去回路内蔵） <p>(6) ウォッチドッグタイマ</p> <ul style="list-style-type: none"> ・8ビットカウンタのオーバフローによりリセット信号を発生
シリアル コミュニケーション インタフェース	<p>3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</p> <p>(1) SCI1：クロック同期式</p> <ul style="list-style-type: none"> ・8ビット/16ビットの転送データを選択可能 <p>(2) SCI3-1：8ビットクロック同期式/調歩同期式</p> <ul style="list-style-type: none"> ・マルチプロセッサ通信機能内蔵 <p>(3) SCI3-2：8ビットクロック同期式/調歩同期式</p> <ul style="list-style-type: none"> ・マルチプロセッサ通信機能内蔵
14ビットPWM	<p>リップル低減をはかったパルス分割方式PWM</p> <ul style="list-style-type: none"> ・外部にローパスフィルタを接続することで14ビットD/A変換器として使用可能
A/D変換器	<p>抵抗ラダー方式による逐次比較方式の10ビットA/D変換器</p> <ul style="list-style-type: none"> ・12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/ または62/
LCDコントローラ/ ドライバ	<p>最大40本のセグメント端子と4本のコモン端子を備えたLCDコントローラ/ドライバ</p> <ul style="list-style-type: none"> ・4種類のデューティ比（スタティック、1/2、1/3、1/4デューティ）を選択可能 ・セグメント端子は8本ごとに汎用ポートに切換え可能
LCD昇圧電源回路	<p>H8/3887シリーズはLCD駆動電源として昇圧定電圧電源を使用可能</p>

1. 概要

項目	仕様			
	製品型名		パッケージ	ROM / RAM サイズ (バイト)
製品ラインアップ	マスク ROM 版	ZTAT 版		
	HD6433882H、 HD6433842H	-	100 ピン QFP (FP-100B)	16k / 1k
	HD6433882F、 HD6433842F	-	100 ピン QFP (FP-100A)	
	HD6433882X、 HD6433842X	-	100 ピン TQFP (TFP-100B)	
	HD6433882W、 HD6433842W	-	100 ピン TQFP (TFP-100G)	
	HD6433883H、 HD6433843H	-	100 ピン QFP (FP-100B)	24k / 1k
	HD6433883F、 HD6433843F	-	100 ピン QFP (FP-100A)	
	HD6433883X、 HD6433843X	-	100 ピン TQFP (TFP-100B)	
	HD6433883W、 HD6433843W	-	100 ピン TQFP (TFP-100G)	
	HD6433884H、 HD6433844H	-	100 ピン QFP (FP-100B)	32k / 2k
	HD6433884F、 HD6433844F	-	100 ピン QFP (FP-100A)	
	HD6433884X、 HD6433844X	-	100 ピン TQFP (TFP-100B)	
	HD6433884W、 HD6433844W	-	100 ピン TQFP (TFP-100G)	
	HD6433885H、 HD6433845H	-	100 ピン QFP (FP-100B)	40k / 2k
	HD6433885F、 HD6433845F	-	100 ピン QFP (FP-100A)	
	HD6433885X、 HD6433845X	-	100 ピン TQFP (TFP-100B)	
	HD6433885W、 HD6433845W	-	100 ピン TQFP (TFP-100G)	
	HD6433886H、 HD6433846H	-	100 ピン QFP (FP-100B)	48k / 2k
	HD6433886F、 HD6433846F	-	100 ピン QFP (FP-100A)	
	HD6433886X、 HD6433846X	-	100 ピン TQFP (TFP-100B)	
	HD6433886W、 HD6433846W	-	100 ピン TQFP (TFP-100G)	
	HD6433887H、 HD6433847H	HD6473887H、 HD6473847H	100 ピン QFP (FP-100B)	60k/2k
	HD6433887F、 HD6433847F	HD6473887F、 HD6473847F	100 ピン QFP (FP-100A)	
	HD6433887X、 HD6433847X	HD6473887X、 HD6473847X	100 ピン TQFP (TFP-100B)	
HD6433887W、 HD6433847W	HD6473887W、 HD6473847W	100 ピン TQFP (TFP-100G)		

1.2 内部ブロック図

H8/3887、H8/3847 シリーズの内部ブロック図を図 1.1 に示します。

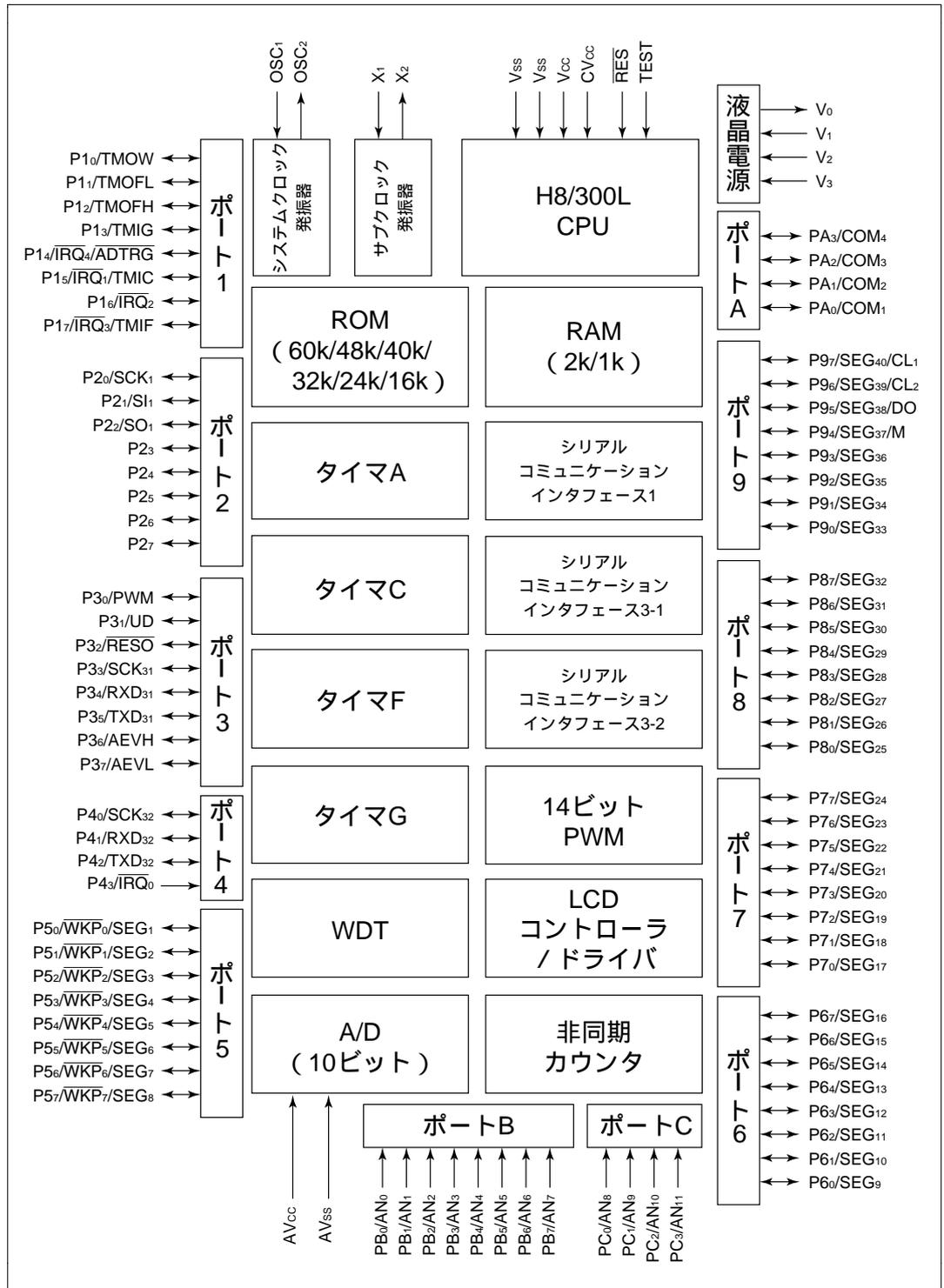


図 1.1 内部ブロック図

1.3 端子説明

1.3.1 ピン配置

H8/3887、H8/3847 シリーズのピン配置図を図 1.2、図 1.3 に示します。

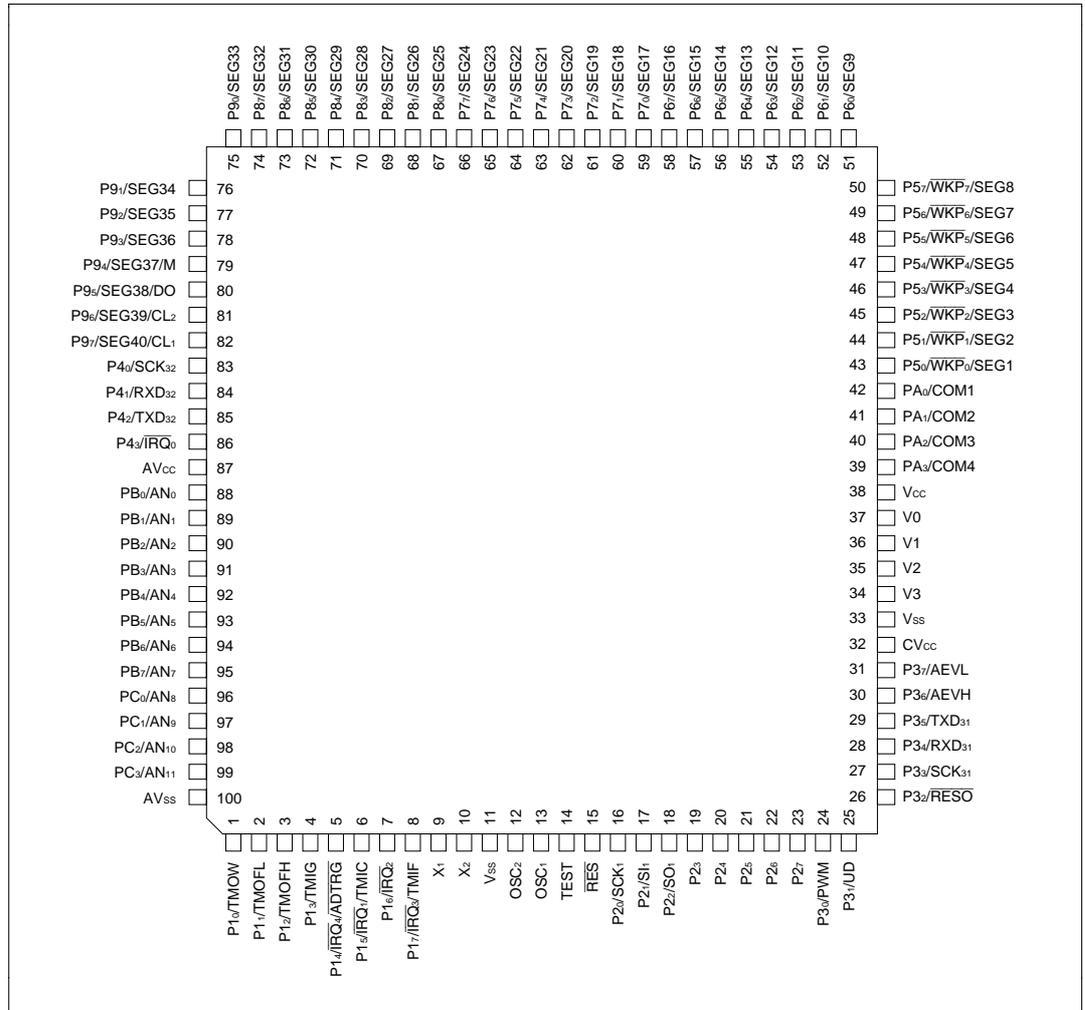


図 1.2 ピン配置図 (FP-100B、TFP-100B、TFP-100G : 上面図)

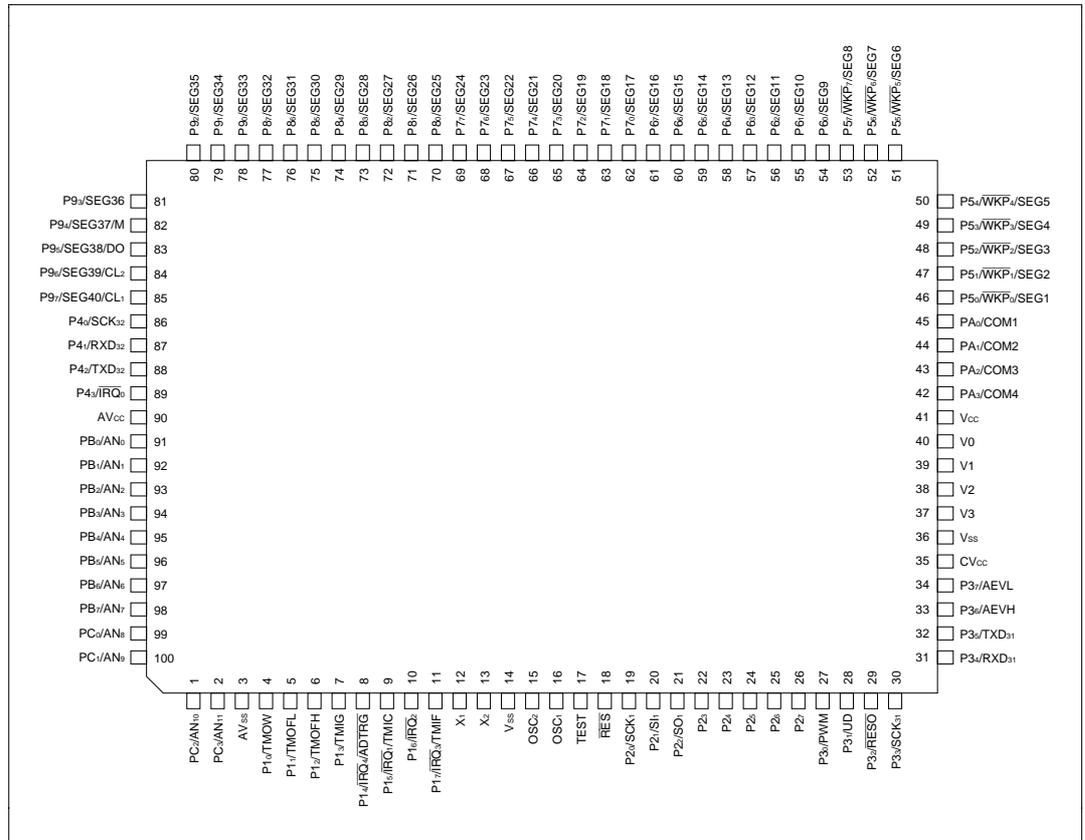


図 1.3 ピン配置図 (FP-100A : 上面図)

1.3.2 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

分類	記号	ピン番号		入出力	機能
		FP-100B TFP-100B TFP-100G	FP-100A		
電源	V _{CC}	38	41	入力	電源 V _{CC} 端子は、システムの電源に接続してください。 CV _{CC} 端子は、「第 14 章 電源回路」を参照してください。
	CV _{CC}	32	35		
	V _{SS}	11	14	入力	グランド V _{SS} 端子は、全端子、システムの電源(0V)に接続してください。
		33	36		
	AV _{CC}	87	90	入力	アナログ電源 A/D変換器用電源端子です。A/D変換器を使用しない場合、システムの電源に接続してください。
	AV _{SS}	100	3	入力	アナロググランド A/D変換器用グランド端子です。システムの電源(0V)に接続してください。
	V ₀	37	40	出力	LCD電源
V ₁ V ₂ V ₃	36 35 34	39 38 37	入力	LCDコントローラ/ドライバ用の電源端子です。電源分割抵抗が内蔵されており、通常 V ₀ と V ₁ を短絡して用います。	
クロック	OSC ₁	13	16	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	12	15	出力	
	X ₁	9	12	入力	32.768kHz または 38.4kHz の水晶発振子を接続します。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。
	X ₂	10	13	出力	

分類	記号	ピン番号		入出力	機能
		FP-100B TFP-100B TFP-100G	FP-100A		
システム 制御	$\overline{\text{RES}}$	15	18	入力	<u>リセット</u> この端子を"Low"レベルにすると、リセット状態になります。
	$\overline{\text{RESO}}$	26	29	出力	<u>リセット出力</u> CPUの内部リセット信号を出力します。
	TEST	14	17	出力	<u>テスト端子</u> 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	$\overline{\text{IRQ}}_0$	86	89	入力	<u>外部割込み要求4~0</u> 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。
	$\overline{\text{IRQ}}_1$	6	9		
$\overline{\text{IRQ}}_2$	7	10			
$\overline{\text{IRQ}}_3$	8	11			
$\overline{\text{IRQ}}_4$	5	8			
	$\overline{\text{WKP}}_7 \sim$ $\overline{\text{WKP}}_0$	50 ~ 43	53 ~ 46	入力	<u>ウェイクアップ割込み要求7~0</u> 立上がりエッジセンス / 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	1	4	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。
	AEVL AEVH	31 30	34 33	入力	<u>非同期イベントカウンタイベント入力</u> 非同期イベントカウンタに入力するイベント入力端子です。
	TMIC	6	9	入力	<u>タイマCイベント入力</u> タイマCのカウンタに入力するイベント入力端子です。
	UD	25	28	入力	<u>タイマCアップ / ダウンセレクト</u> タイマCのカウンタのアップ / ダウンカウントを選択します。"High"レベル印加でアップカウンタ、"Low"レベル印加でダウンカウンタとして動作します。
	TMIF	8	11	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。

1. 概要

分類	記号	ピン番号		入出力	機能
		FP-100B TFP-100B TFP-100G	FP-100A		
タイマ	TMOFL	2	5	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	3	6	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	4	7	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。
14 ビット PWM	PWM	24	27	出力	<u>14 ビット PWM 出力</u> 14 ビット PWM により生成された波形の出力端子です。
I/O ポート	PB ₇ ~ PB ₀	95 ~ 88	98 ~ 91	入力	<u>ポート B</u> 8 ビットの入力端子です。
	PC ₃ ~ PC ₀	99 ~ 96	2 ~ 1 100 ~ 99	入力	<u>ポート C</u> 4 ビットの入力端子です。
	P4 ₃	86	89	入力	<u>ポート 4 (ビット 3)</u> 1 ビットの入力端子です。
	P4 ₂ ~ P4 ₀	85 ~ 83	88 ~ 86	入出力	<u>ポート 4 (ビット 2 ~ ビット 0)</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 4 (PCR4) によって、1 ビットごとに入出力を指定できます。
	PA ₃ ~ PA ₀	39 ~ 42	42 ~ 45	入出力	<u>ポート A</u> 4 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 ~ P1 ₀	8 ~ 1	11 ~ 4	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 ~ P2 ₀	23 ~ 16	26 ~ 19	入出力	<u>ポート 2</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 2 (PCR2) によって、1 ビットごとに入出力を指定できます。

分類	記号	ピン番号		入出力	機能
		FP-100B TFP-100B TFP-100G	FP-100A		
I/O ポート	P3 ₇ ~ P3 ₀	31 ~ 24	34 ~ 27	入出力	<u>ポート3</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3 (PCR3) によって、1ビットごとに入出力を指定できます。
	P5 ₇ ~ P5 ₀	50 ~ 43	53 ~ 46	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 ₇ ~ P6 ₀	58 ~ 51	61 ~ 54	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 ₇ ~ P7 ₀	66 ~ 59	69 ~ 62	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 ₇ ~ P8 ₀	74 ~ 67	77 ~ 70	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。
	P9 ₇ ~ P9 ₀	82 ~ 75	85 ~ 78	入出力	<u>ポート9</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9 (PCR9) 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 ₁	17	20	入力	<u>SCI1 受信データ入力</u> SCI1 のデータ入力端子です。
	SO ₁	18	21	出力	<u>SCI1 送信データ出力</u> SCI1 のデータ出力端子です。
	SCK ₁	16	19	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。
	RXD ₃₁	28	31	入力	<u>SCI31 受信データ入力</u> SCI31 のデータ入力端子です。
	TXD ₃₁	29	32	出力	<u>SCI31 送信データ出力</u> SCI31 のデータ出力端子です。

1. 概要

分類	記号	ピン番号		入出力	機能
		FP-100B TFP-100B TFP-100G	FP-100A		
シリアル コミュニ ケーショ ンインタ フェース (SCI)	SCK ₃₁	27	30	入出力	SCI31 クロック入出力 SCI31 のクロック入出力端子です。
	RXD ₃₂	84	87	入力	SCI32 受信データ入力 SCI32 のデータ入力端子です。
	TXD ₃₂	85	88	出力	SCI32 送信データ出力 SCI32 のデータ出力端子です。
	SCK ₃₂	83	86	入出力	SCI32 クロック入出力 SCI32 のクロック入出力端子です。
A/D 変換器	AN ₁₁ ~ AN ₀	99 ~ 88	2 ~ 1 100 ~ 91	入力	アナログ入力 (チャンネル 11 ~ チャンネル 0) A/D 変換器へのアナログデータ入力端子で す。
	ADTRG	5	8	入力	A/D 変換器トリガ入力 A/D 変換器の外部トリガ入力端子です。
LCD コントロ ーラ/ド ライバ	COM ₄ ~ COM ₁	39 ~ 42	42 ~ 45	出力	LCD コモン出力 LCD のコモン出力端子です。
	SEG ₄₀ ~ SEG ₁	82 ~ 43	85 ~ 46	出力	LCD セグメント出力 LCD のセグメント出力端子です。
	CL ₁	82	85	出力	LCD ラッチクロック セグメント外部拡張用の表示データラッチク ロック出力端子です。
	CL ₂	81	84	出力	LCD シフトクロック セグメント外部拡張用の表示データシフトク ロック出力端子です。
	DO	80	83	出力	LCD シリアルデータ出力 セグメント外部拡張用のシリアル表示デー タ出力端子です。
	M	79	82	出力	LCD 交流化信号 セグメント外部拡張用の LCD 交流化信号出 力端子です。

2. CPU

第2章 目次

2.1	概要.....	17
	2.1.1 特長.....	17
	2.1.2 アドレス空間.....	18
	2.1.3 レジスタ構成.....	18
2.2	各レジスタの説明.....	19
	2.2.1 汎用レジスタ.....	19
	2.2.2 コントロールレジスタ.....	19
	2.2.3 CPU内部レジスタの初期値.....	21
2.3	データ構成.....	22
	2.3.1 汎用レジスタのデータ構成.....	22
	2.3.2 メモリ上でのデータ構成.....	23
2.4	アドレッシングモード.....	24
	2.4.1 アドレッシングモード.....	24
	2.4.2 実効アドレスの計算方法.....	26
2.5	命令セット.....	30
	2.5.1 データ転送命令.....	32
	2.5.2 算術演算命令.....	34
	2.5.3 論理演算命令.....	35
	2.5.4 シフト命令.....	35
	2.5.5 ビット操作命令.....	37
	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.....	44
2.6	基本動作タイミング.....	45
	2.6.1 内蔵メモリ (RAM、ROM).....	45
	2.6.2 内蔵周辺モジュール.....	46
2.7	CPUの状態.....	48

2.7.1	概要.....	48
2.7.2	プログラム実行状態.....	49
2.7.3	プログラム停止状態.....	49
2.7.4	例外処理状態.....	49
2.8	メモリマップ.....	50
2.8.1	メモリマップ.....	50
2.9	使用上の注意事項.....	57
2.9.1	データアクセスに関する注意事項.....	57
2.9.2	ビット操作命令使用上の注意事項.....	59
2.9.3	EEPMOV 命令使用上の注意事項.....	65

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	0.67 μ s*
8×8 ビット乗算	4.67 μ s*
16÷8 ビット除算	4.67 μ s*

【注】 * 数値は、 = 3MHz 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。

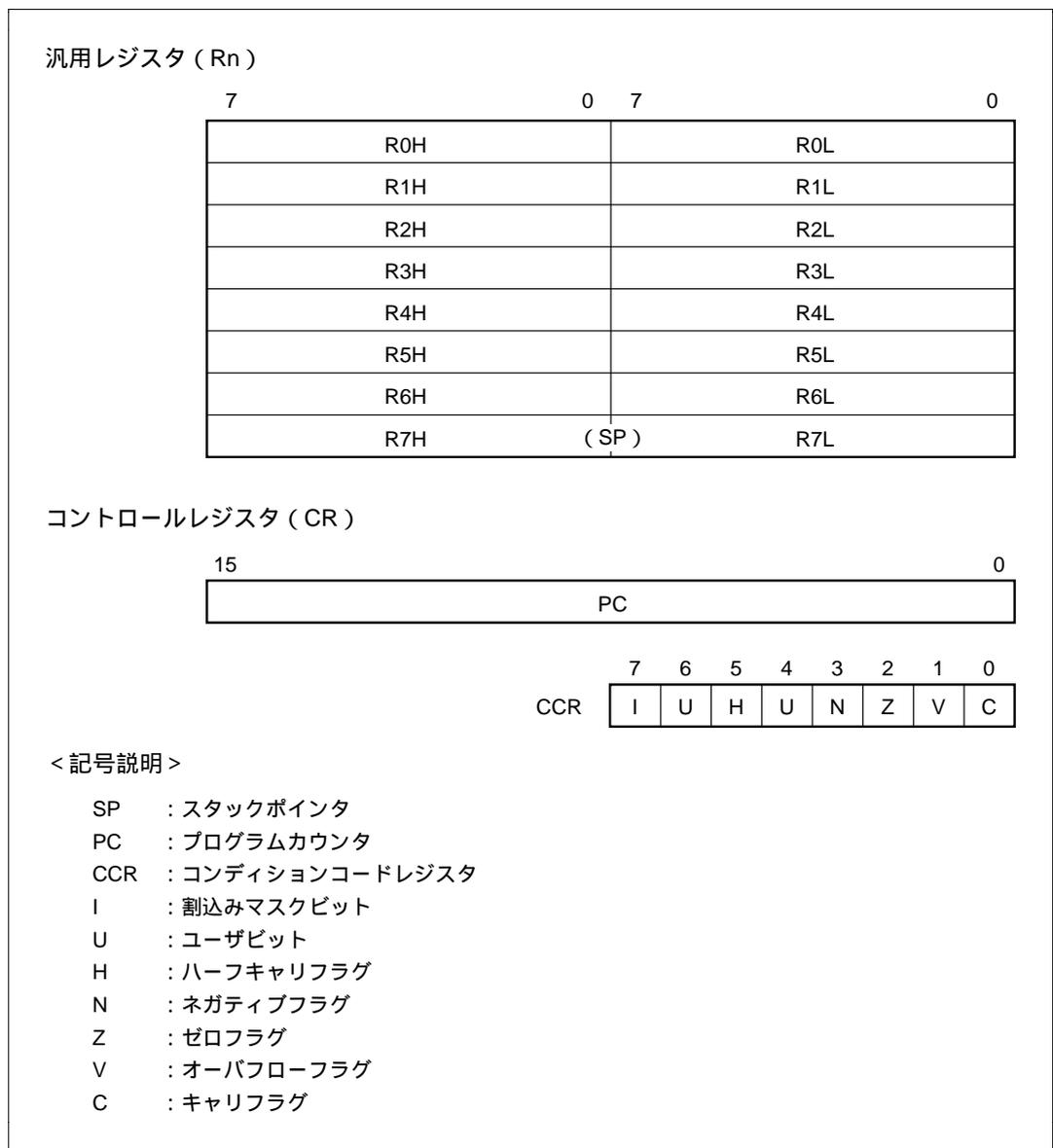


図 2.1 CPU 内部レジスタ構成

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。

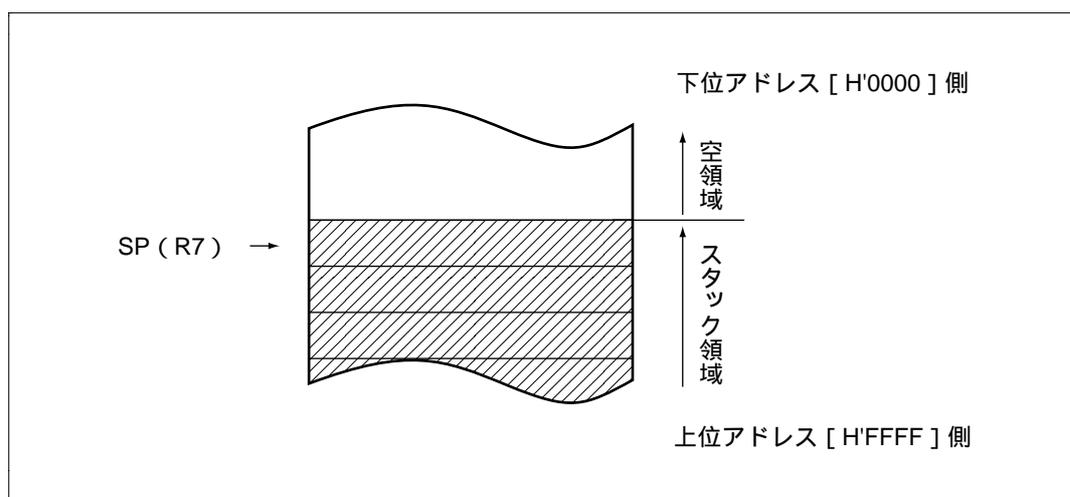


図 2.2 スタックの状態

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。

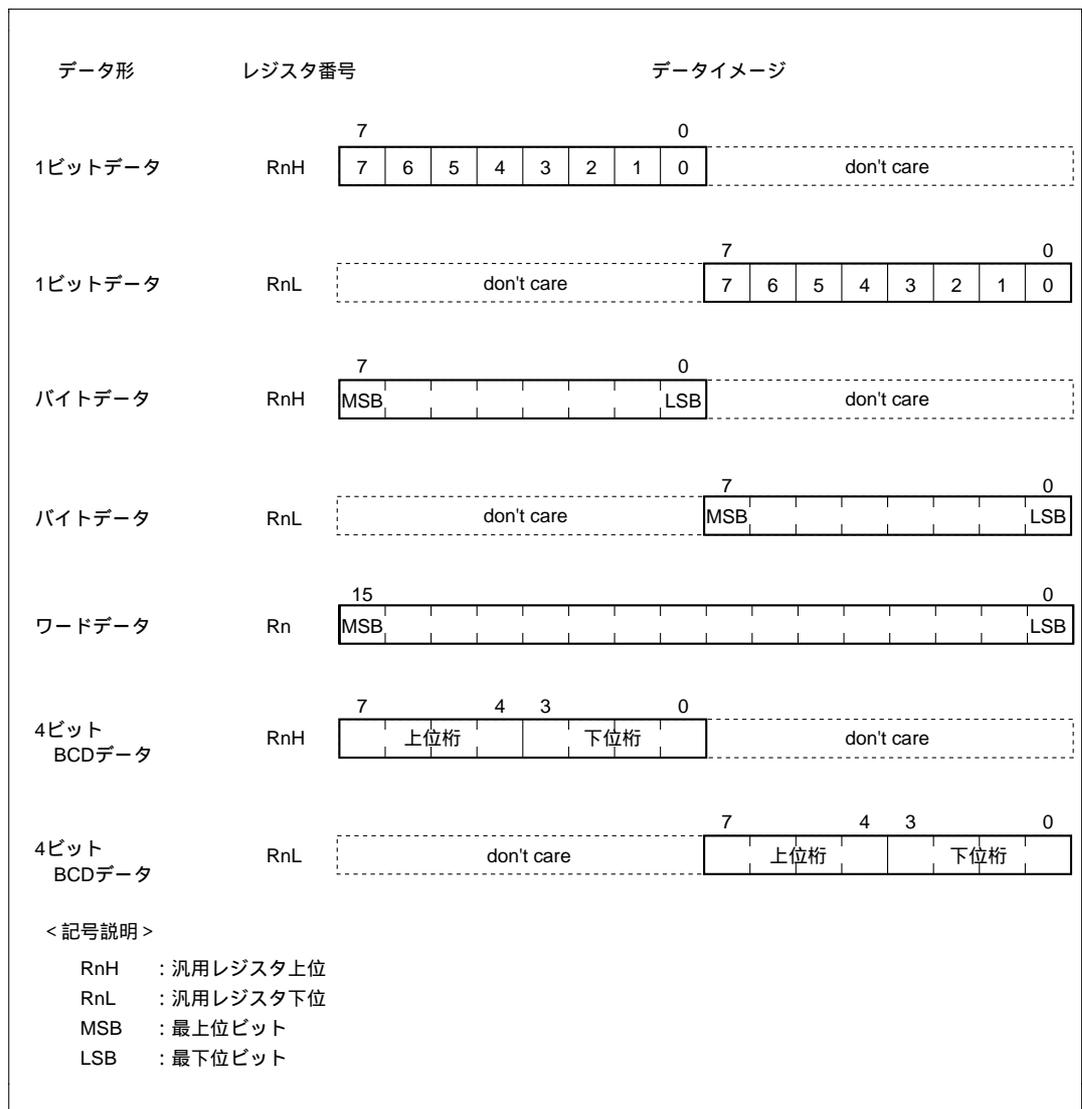


図 2.3 汎用レジスタのデータ構成

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。

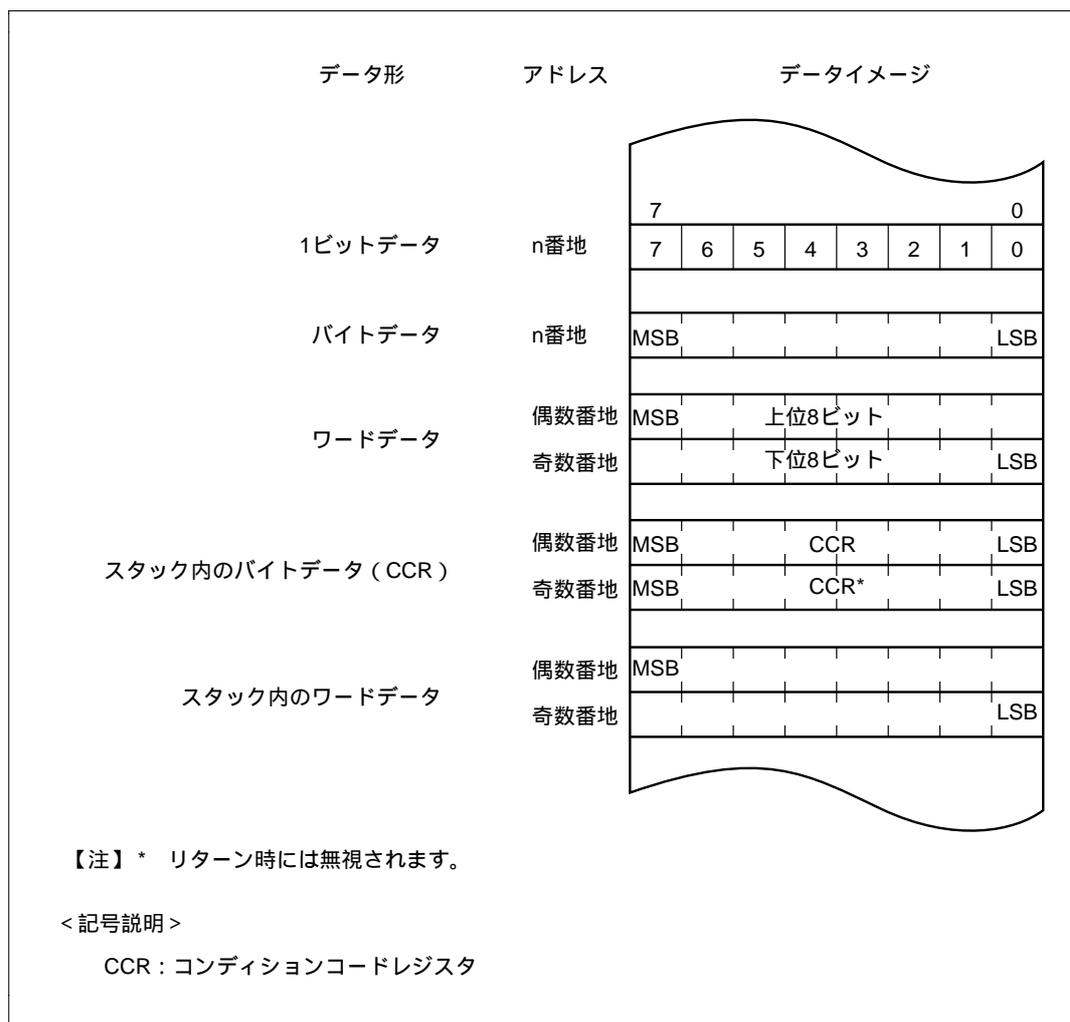


図2.4 メモリ上でのデータ構成

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

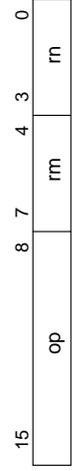
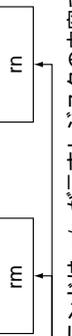
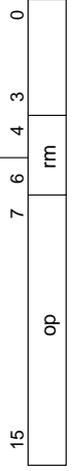
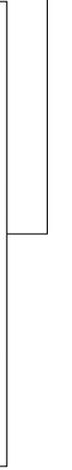
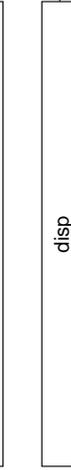
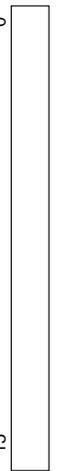
各アドレッシングモードにおける実効アドレス (EA: Effective Address) の計算法を表 2.2 に示します。

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、ADDDX、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。

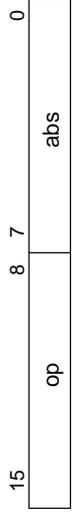
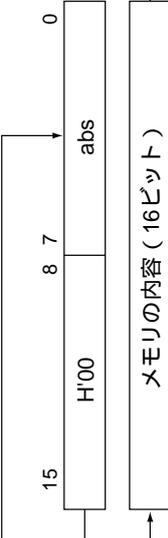
また、ビット操作命令では、オペランドの指定に (1) レジスタ直接、(2) レジスタ間接および (5) 絶対アドレス (8 ビット) が使用可能です。さらに、オペランド中のビット番号を指定するために (1) レジスタ直接 (BSET、BCLR、BNOT、BTST の各命令) および (6) イミディエイト (3 ビット) が独立して使用可能です。

表 2.2 実効アドレスの計算方法

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 		 オペランドはrm/mが示すレジスタの内容です。
2	レジスタ間接 @Rn 		
3	ディスプレイ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 ・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 	 	 

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。

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	<p>絶対アドレス</p> <p>@aa:8</p> <p>@aa:16</p>		
6	<p>イミディエイト</p> <p>#xx:8</p> <p>#xx:16</p>		<p>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。</p>
7	<p>プログラムカウンタ相対 @ (d:8, PC)</p>		

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* ¹ 、PUSH* ¹	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* ² 、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。

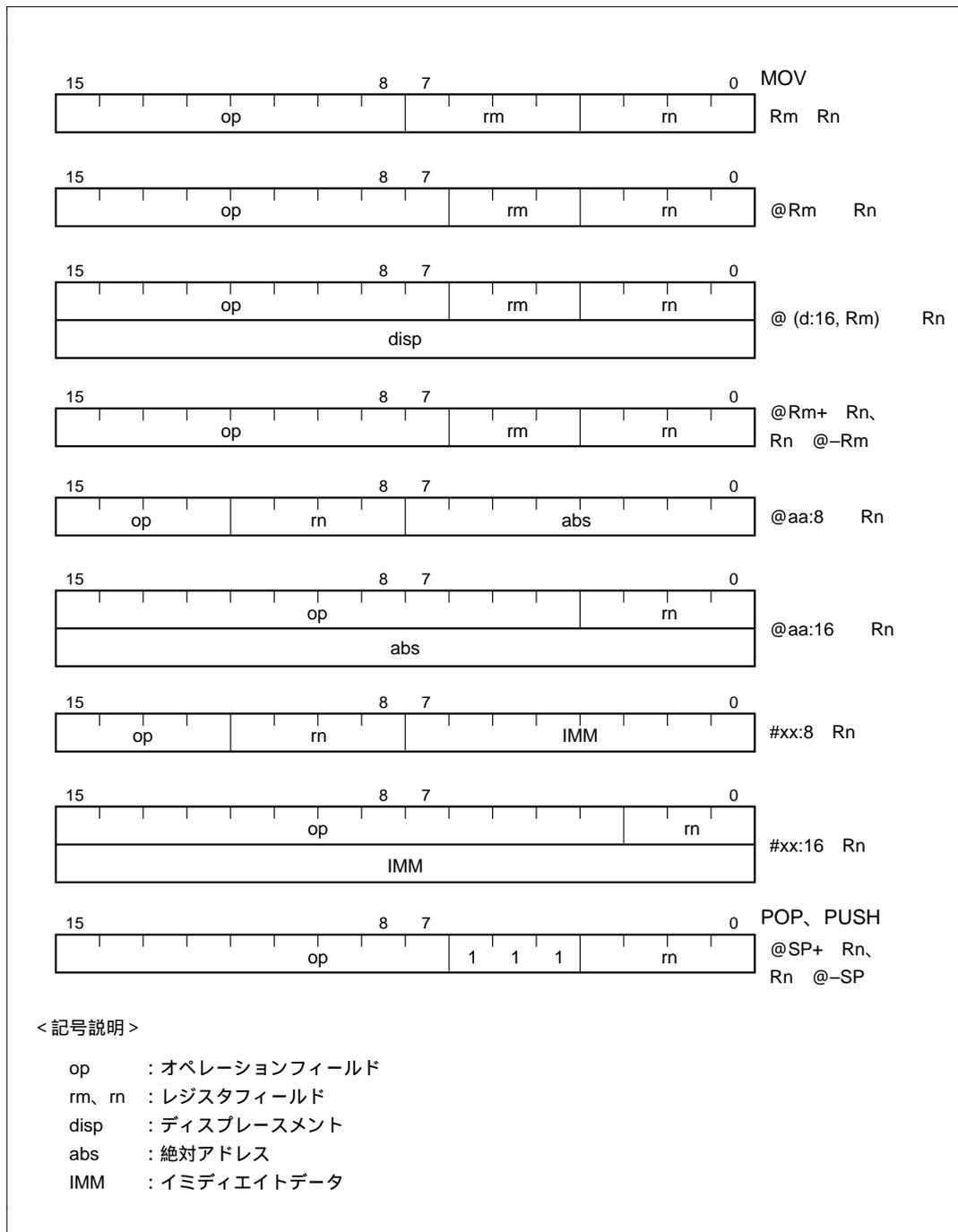


図 2.5 データ転送命令の命令フォーマット

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。

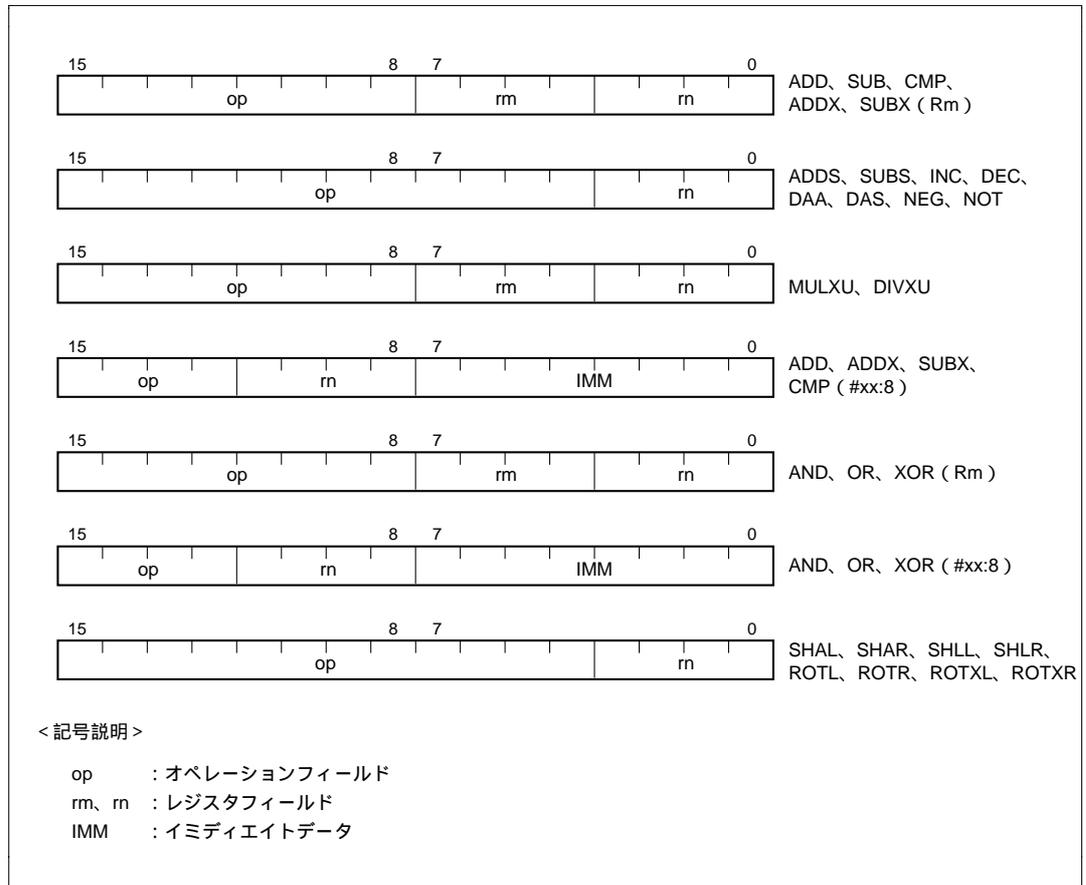


図 2.6 算術演算命令・論理演算命令・シフト命令の命令フォーマット

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

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。
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。

命令	サイズ*	機能
BXOR	B	$C \oplus (\text{ビット番号 of } \langle \text{EAd} \rangle)$ 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	$C \oplus [\sim (\text{ビット番号 of } \langle \text{EAd} \rangle)]$ 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	$(\text{ビット番号 of } \langle \text{EAd} \rangle)$ 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	$\sim (\text{ビット番号 of } \langle \text{EAd} \rangle)$ 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。
BST	B	C $(\text{ビット番号 of } \langle \text{EAd} \rangle)$ 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	$\sim C$ $(\text{ビット番号 of } \langle \text{EAd} \rangle)$ 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。

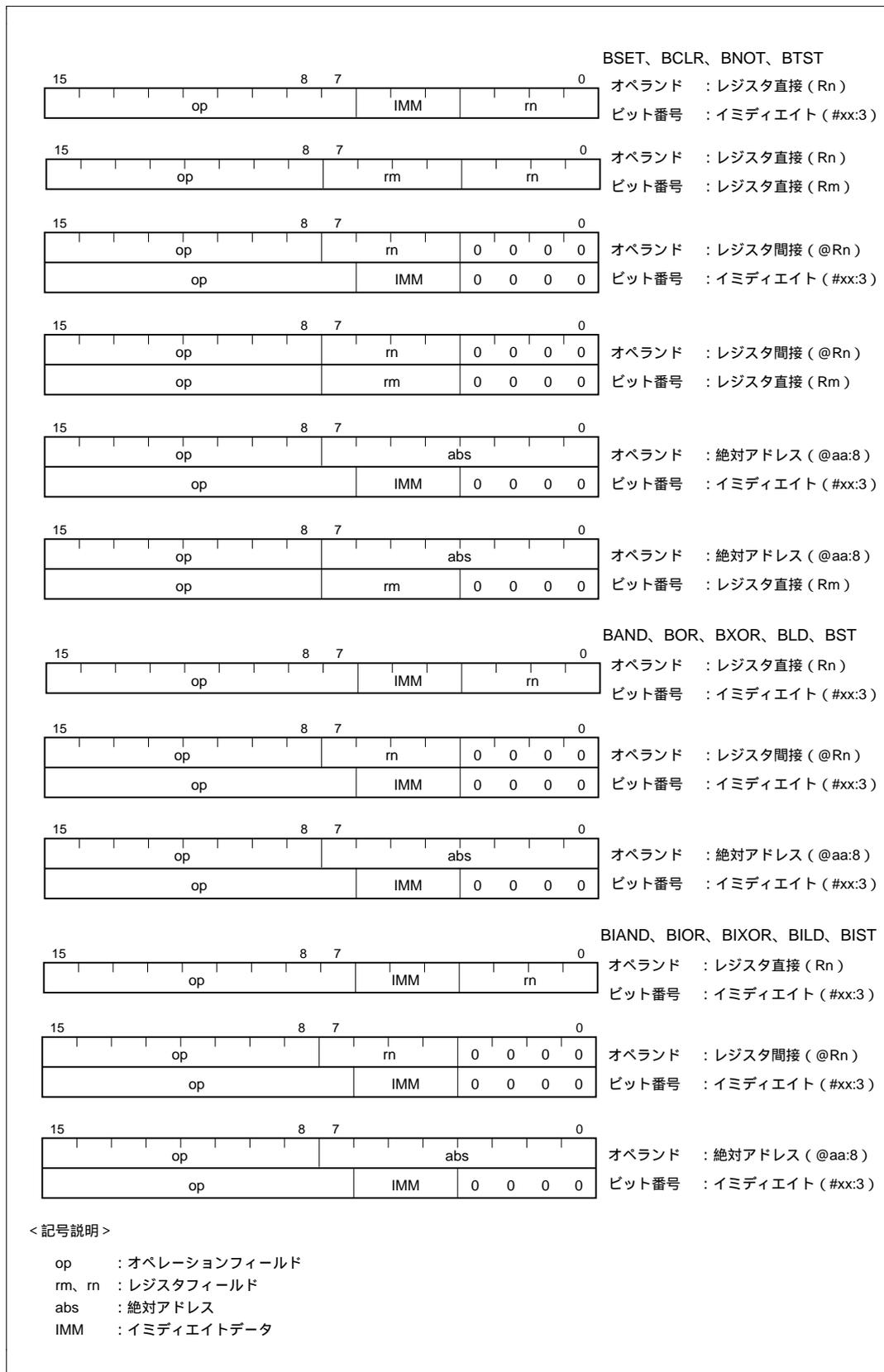


図 2.7 ビット操作命令の命令フォーマット

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>$N \oplus V = 0$</td> </tr> <tr> <td>BLT</td> <td>Less Than</td> <td>$N \oplus V = 1$</td> </tr> <tr> <td>BGT</td> <td>Greater Than</td> <td>$Z (N \oplus V) = 0$</td> </tr> <tr> <td>BLE</td> <td>Less or Equal</td> <td>$Z (N \oplus V) = 1$</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。

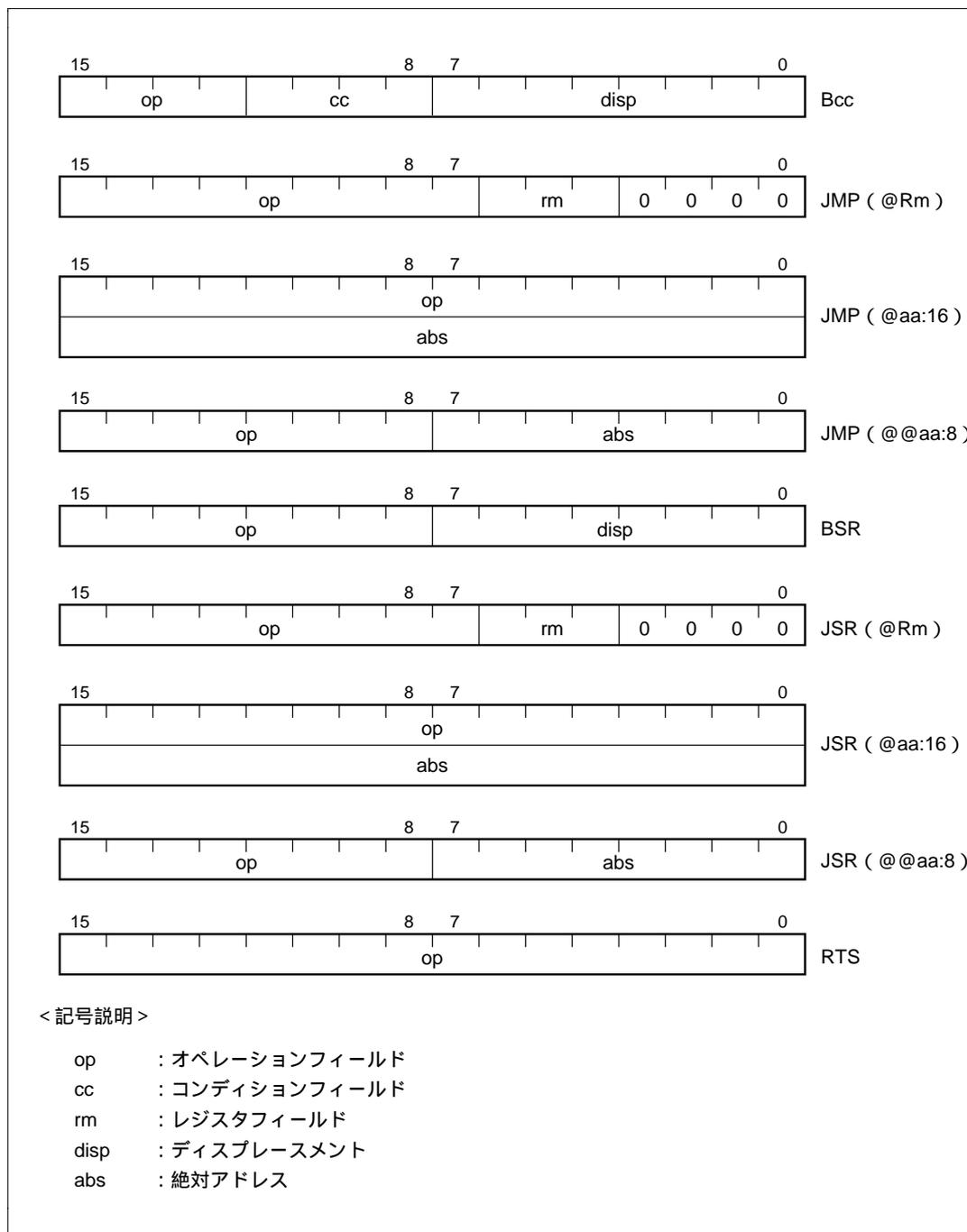


図 2.8 分岐命令の命令フォーマット

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。

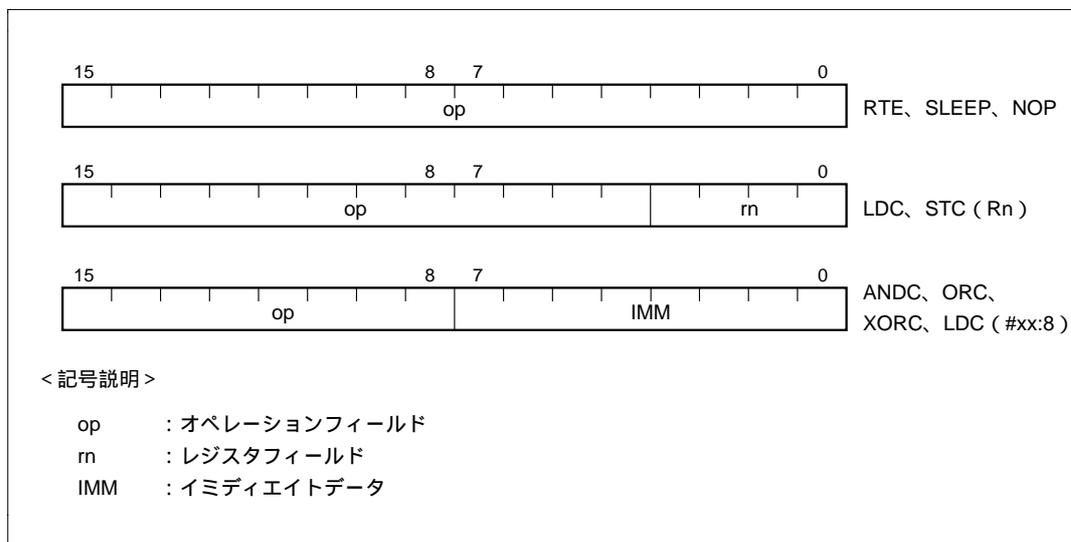


図 2.9 システム制御命令の命令フォーマット

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。

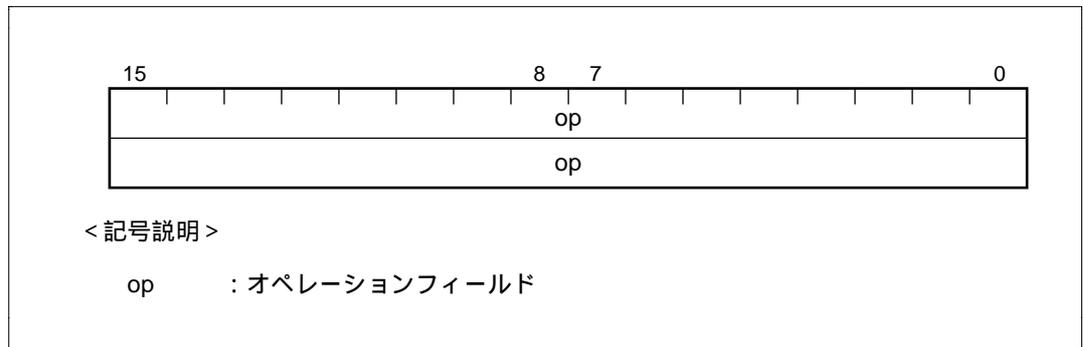


図 2.10 ブロック転送命令の命令フォーマット

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(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 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 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。

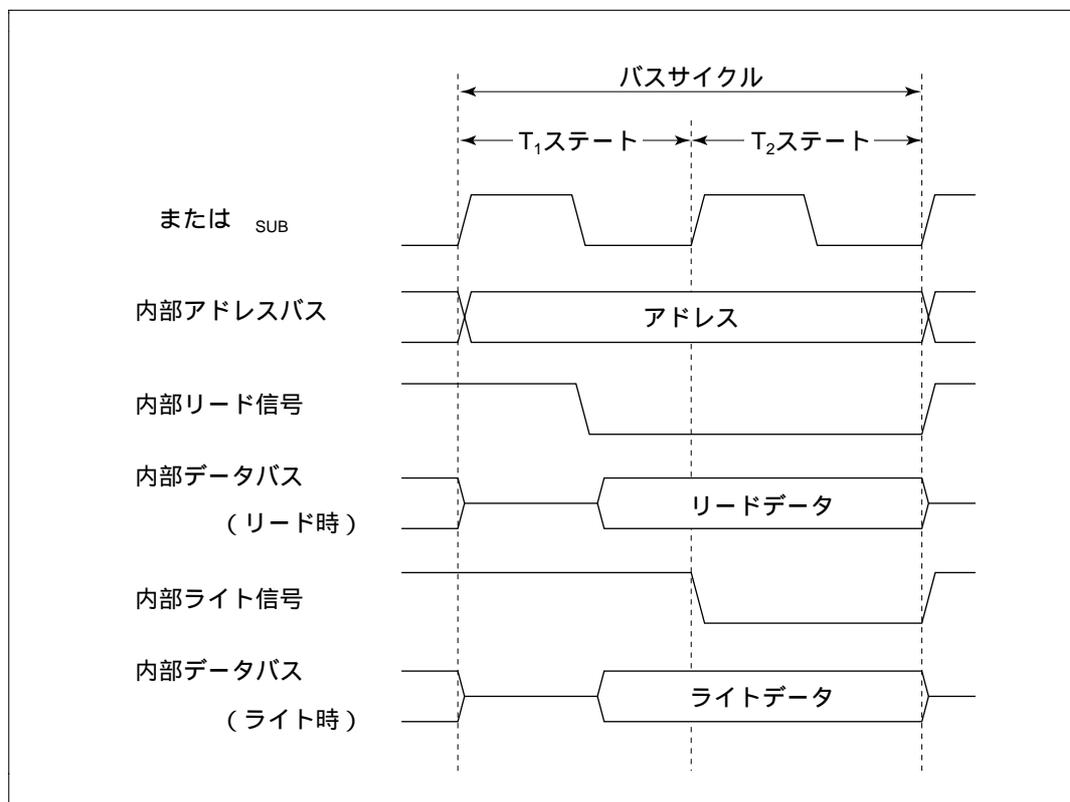


図 2.11 内蔵メモリアccessサイクル

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。

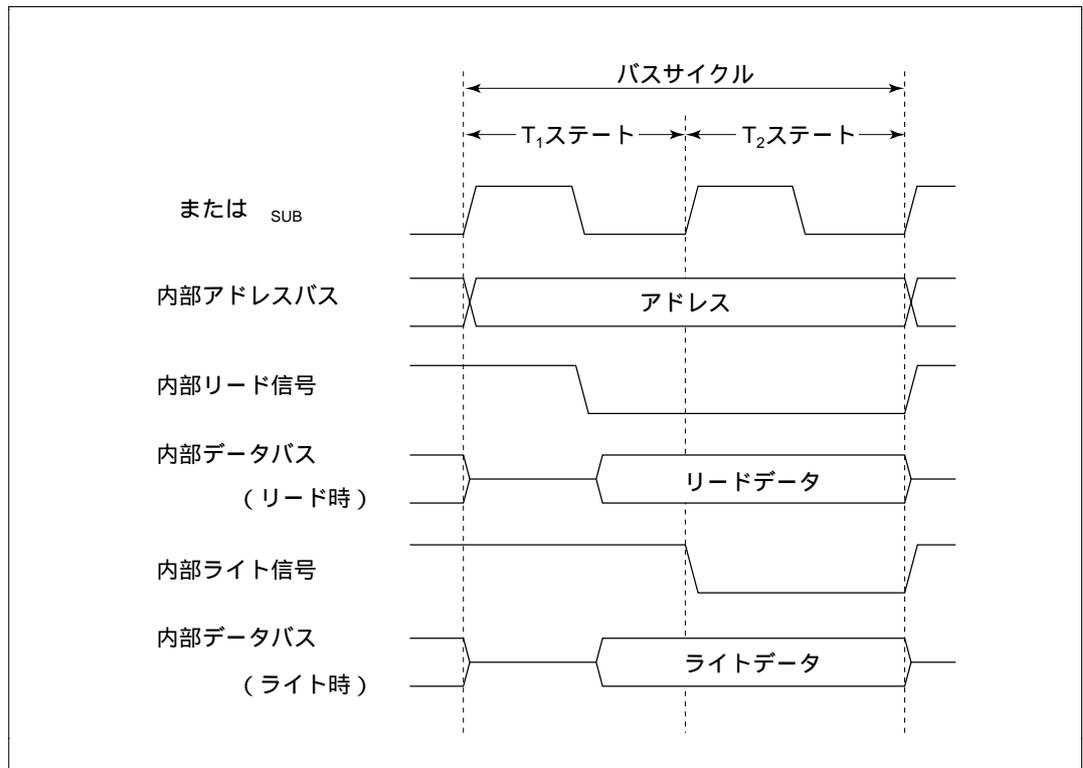


図 2.12 内蔵周辺モジュールアクセスサイクル(2ステートアクセス)

(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。

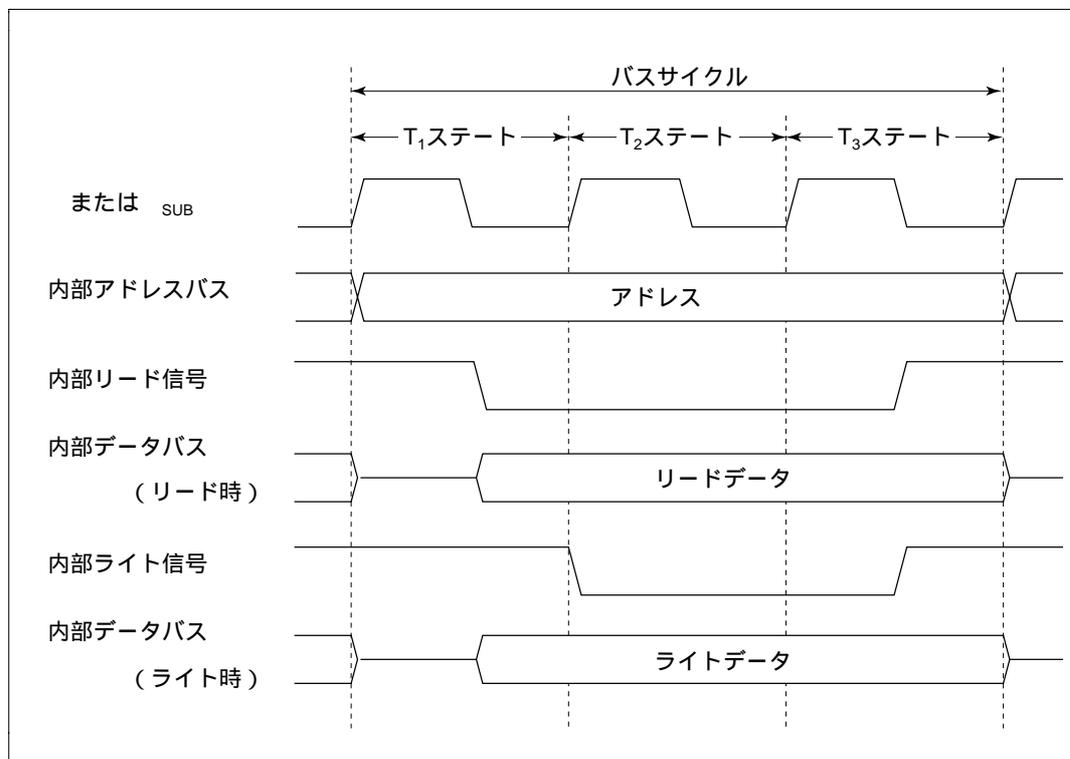


図 2.13 内蔵周辺モジュールアクセスサイクル (3 ステートアクセス)

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(高速)モード、スリープ(中速)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。

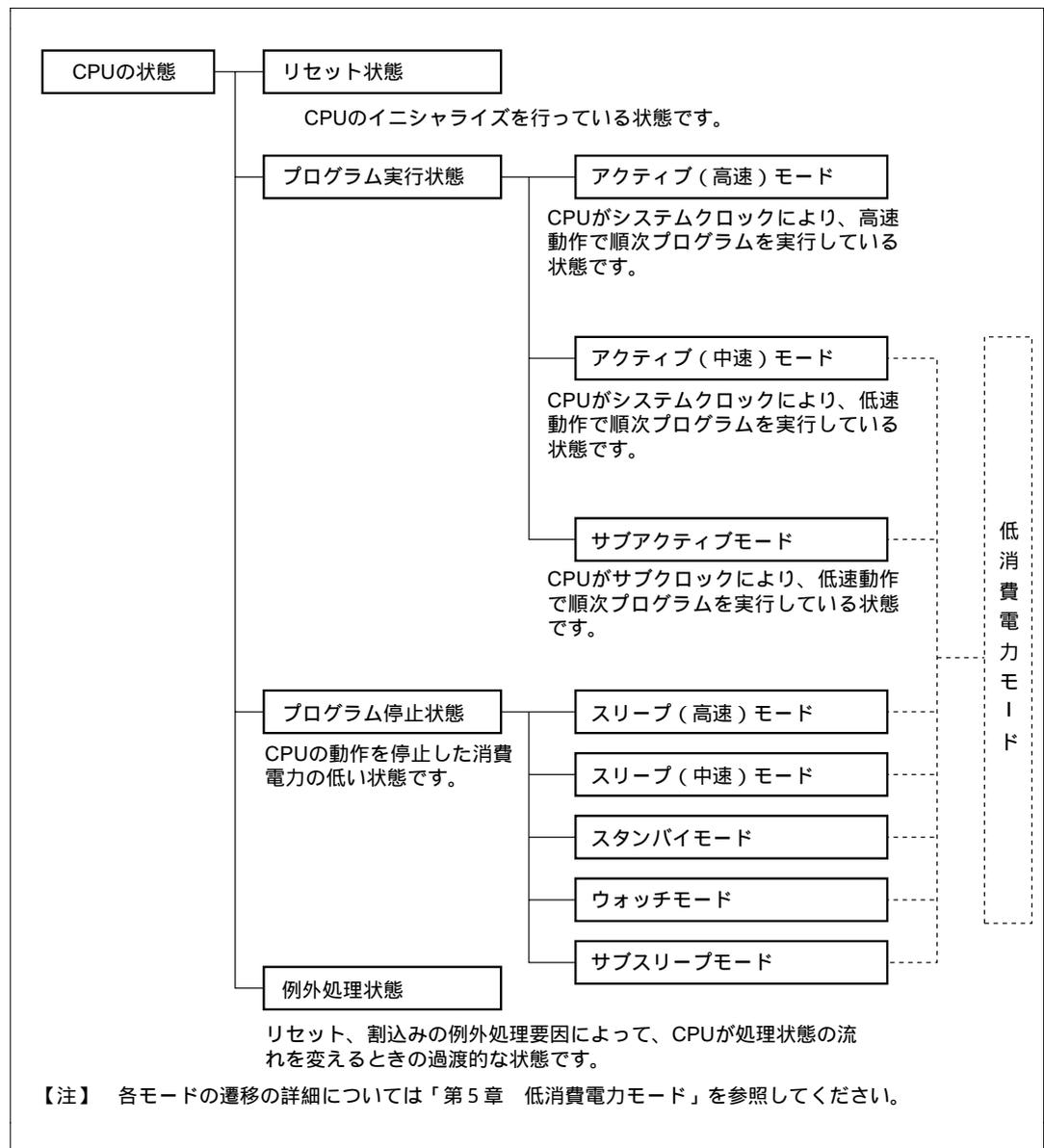


図 2.14 CPU の状態の分類

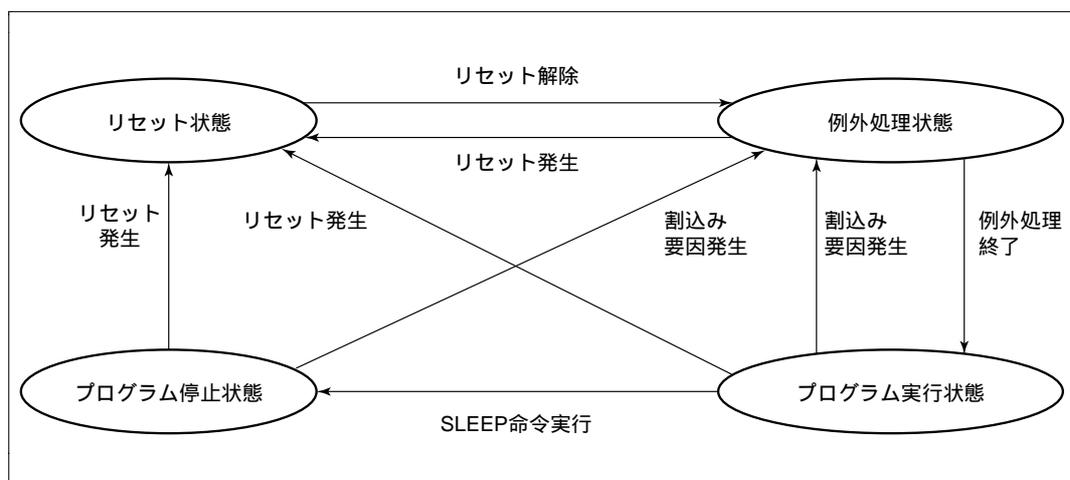


図 2.15 状態遷移図

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（高速）モード、スリープ（中速）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5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

2.8.1 メモリマップ

H8/3882、H8/3842 のメモリマップを図 2.16(1) に、H8/3883、H8/3843 のメモリマップを図 2.16(2) に、H8/3884、H8/3844 のメモリマップを図 2.16(3) に、H8/3885、H8/3845 のメモリマップを図 2.16(4) に、H8/3886、H8/3846 のメモリマップを図 2.16(5) に、H8/3887、H8/3847 のメモリマップを図 2.16(6) に示します。

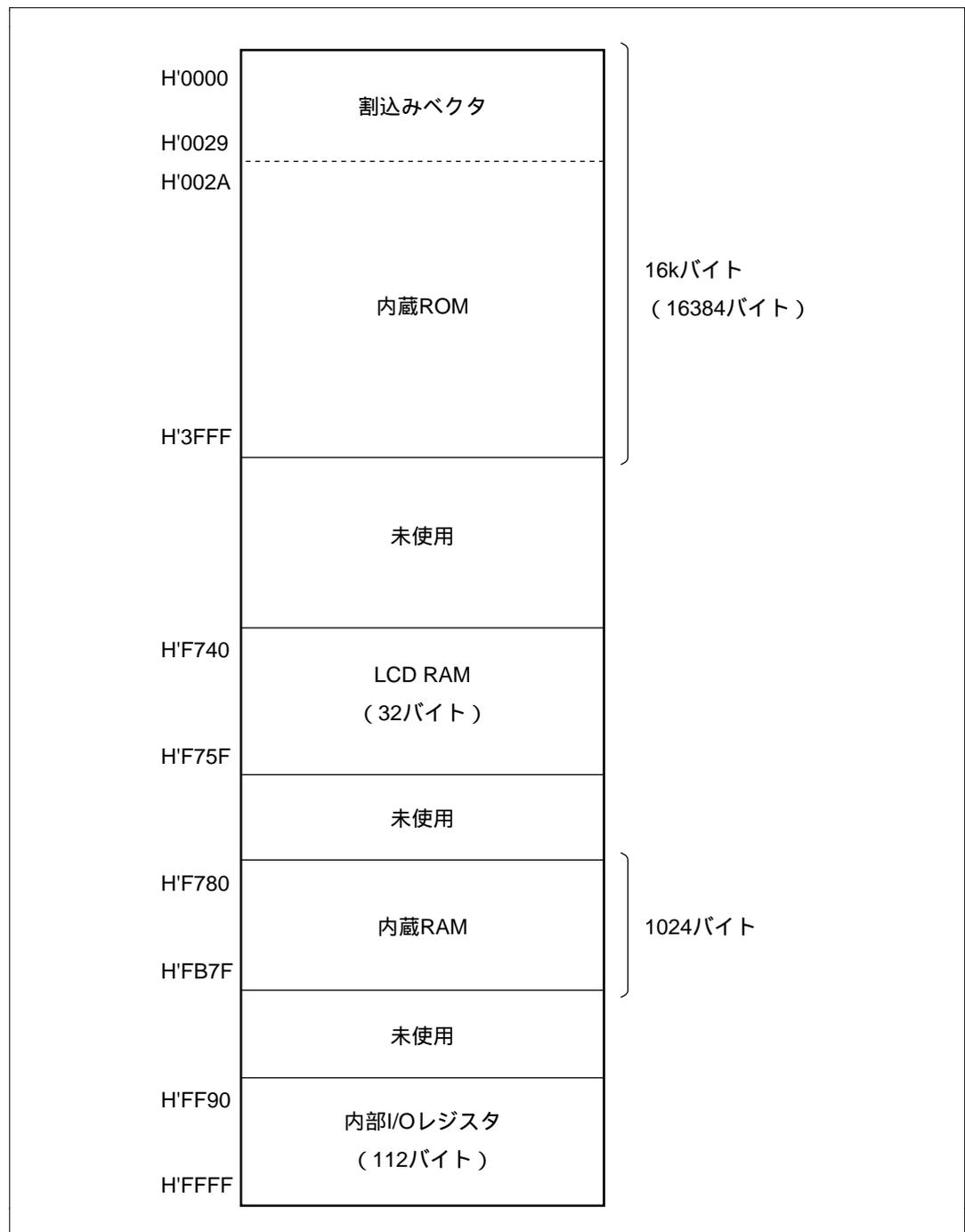


図 2.16 (1) H8/3882、H8/3842 のメモリマップ

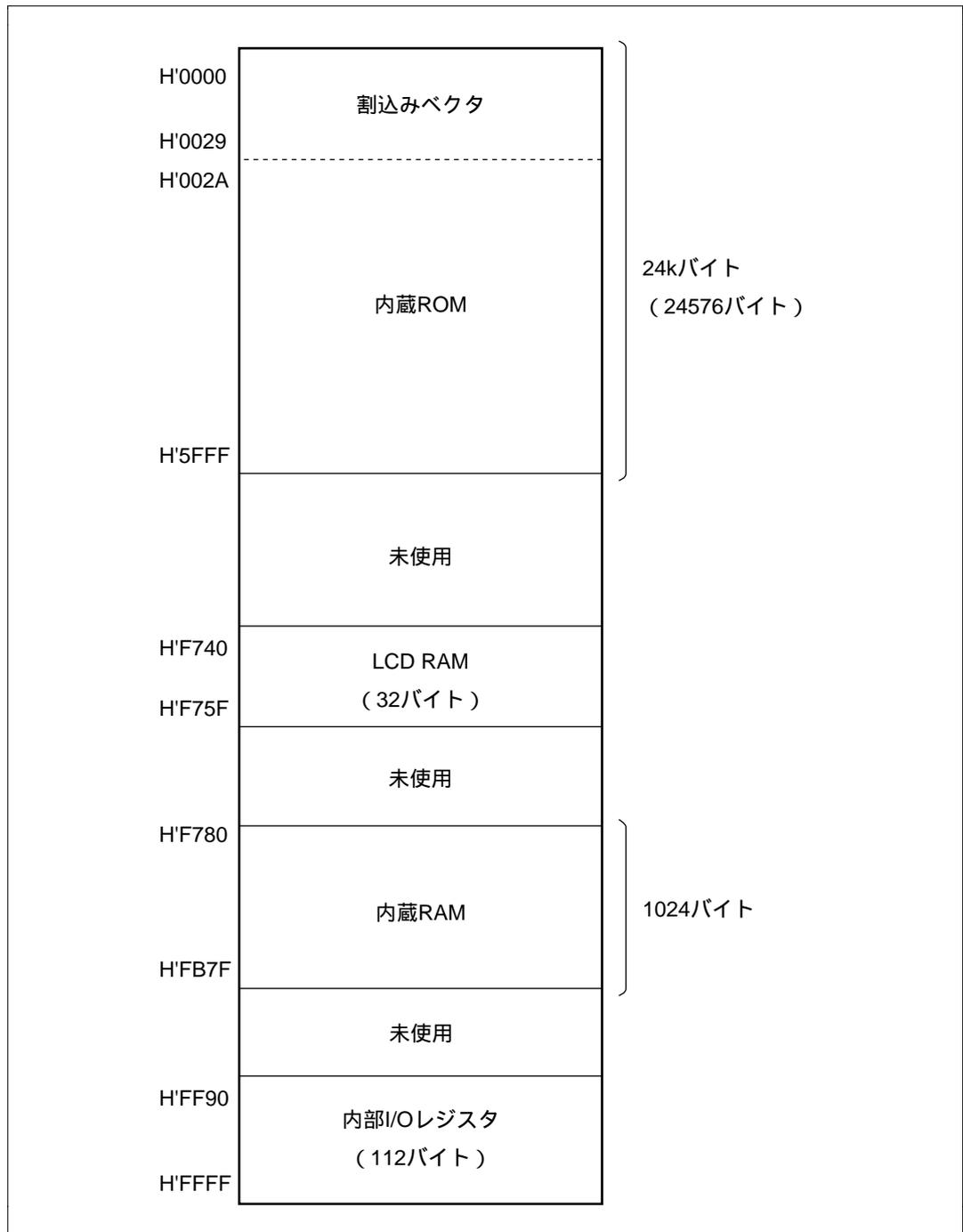


図 2.16 (2) H8/3883、H8/3843 のメモリマップ

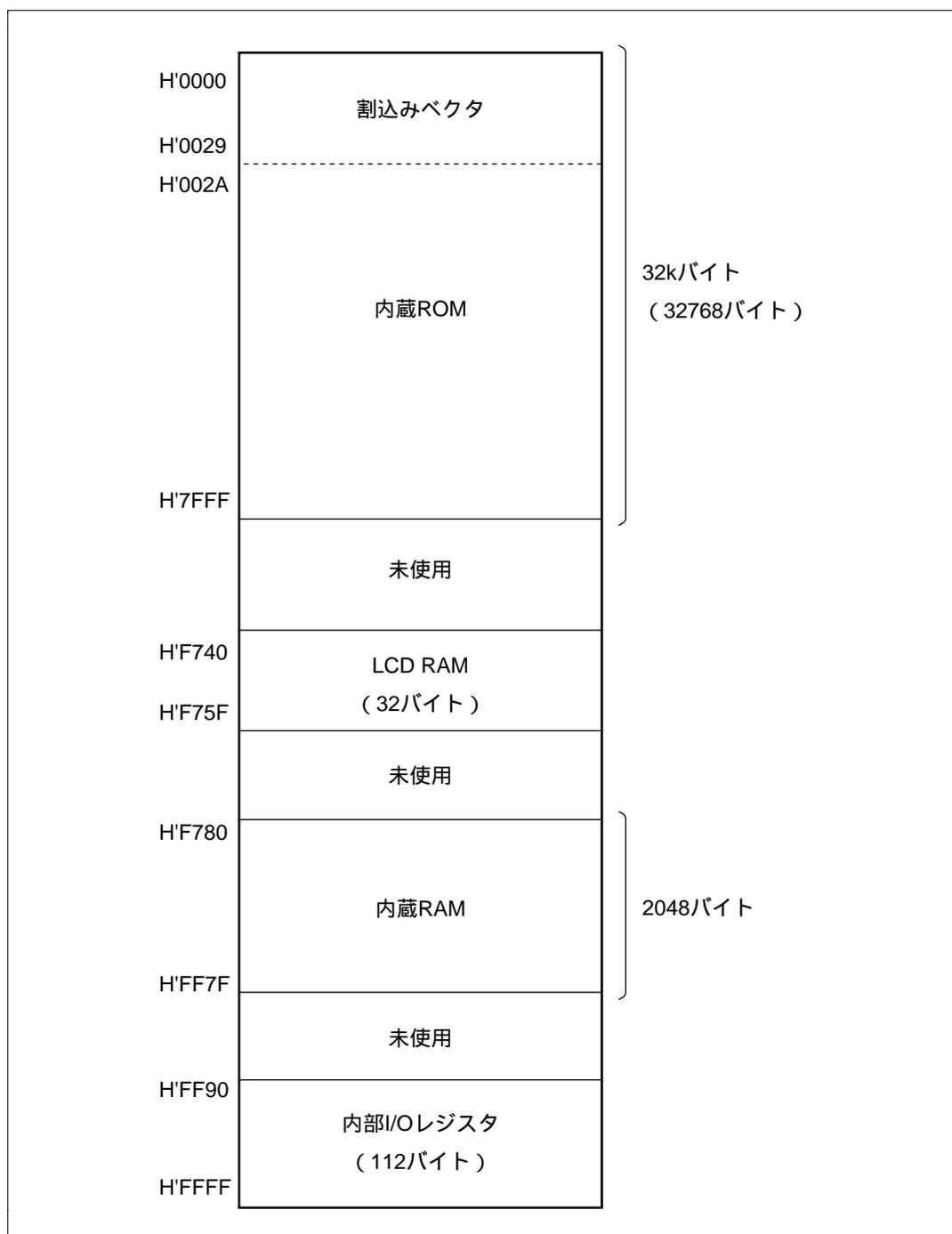


図 2.16 (3) H8/3884、H8/3844 のメモリマップ

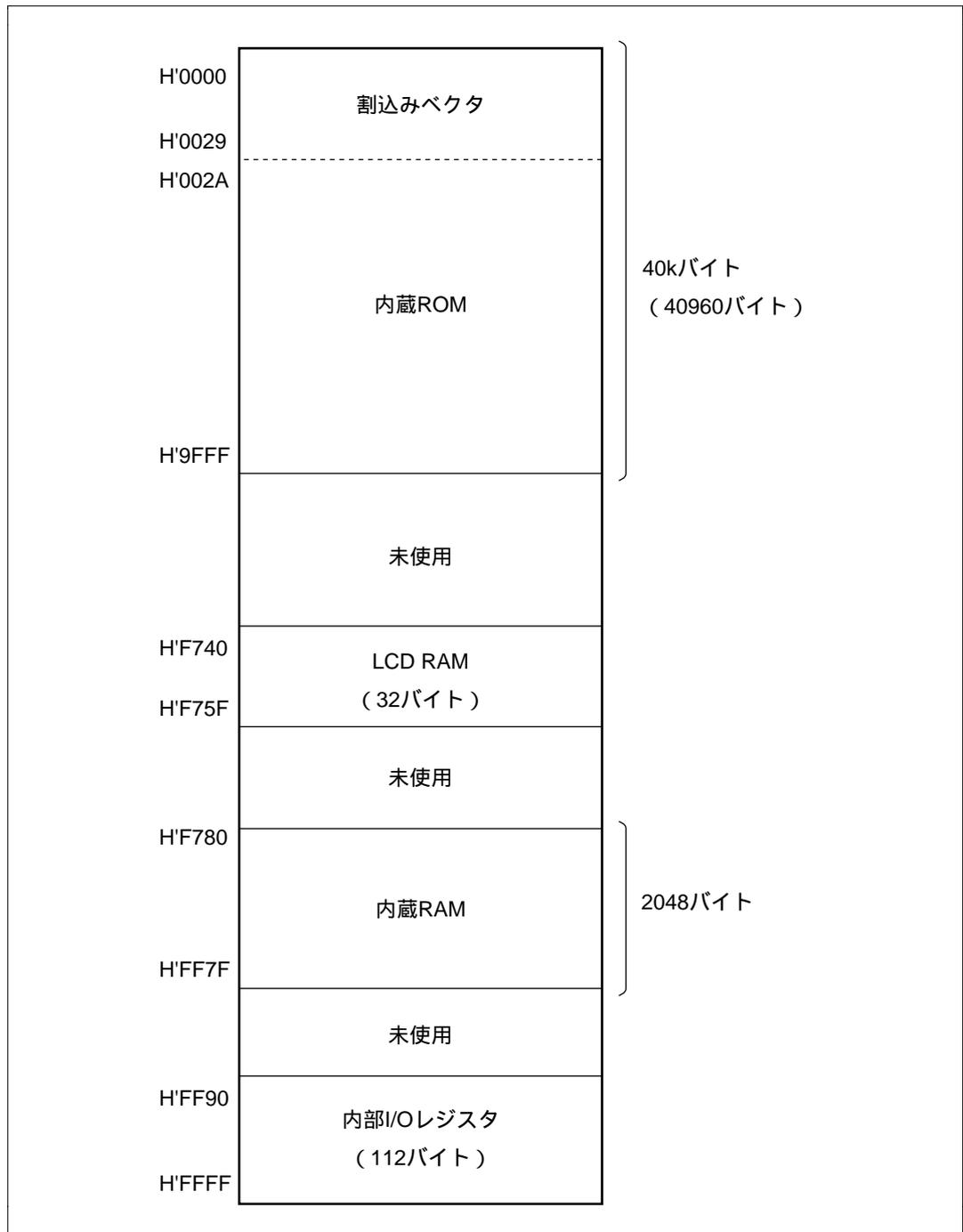


図 2.16 (4) H8/3885、H8/3845 のメモリマップ

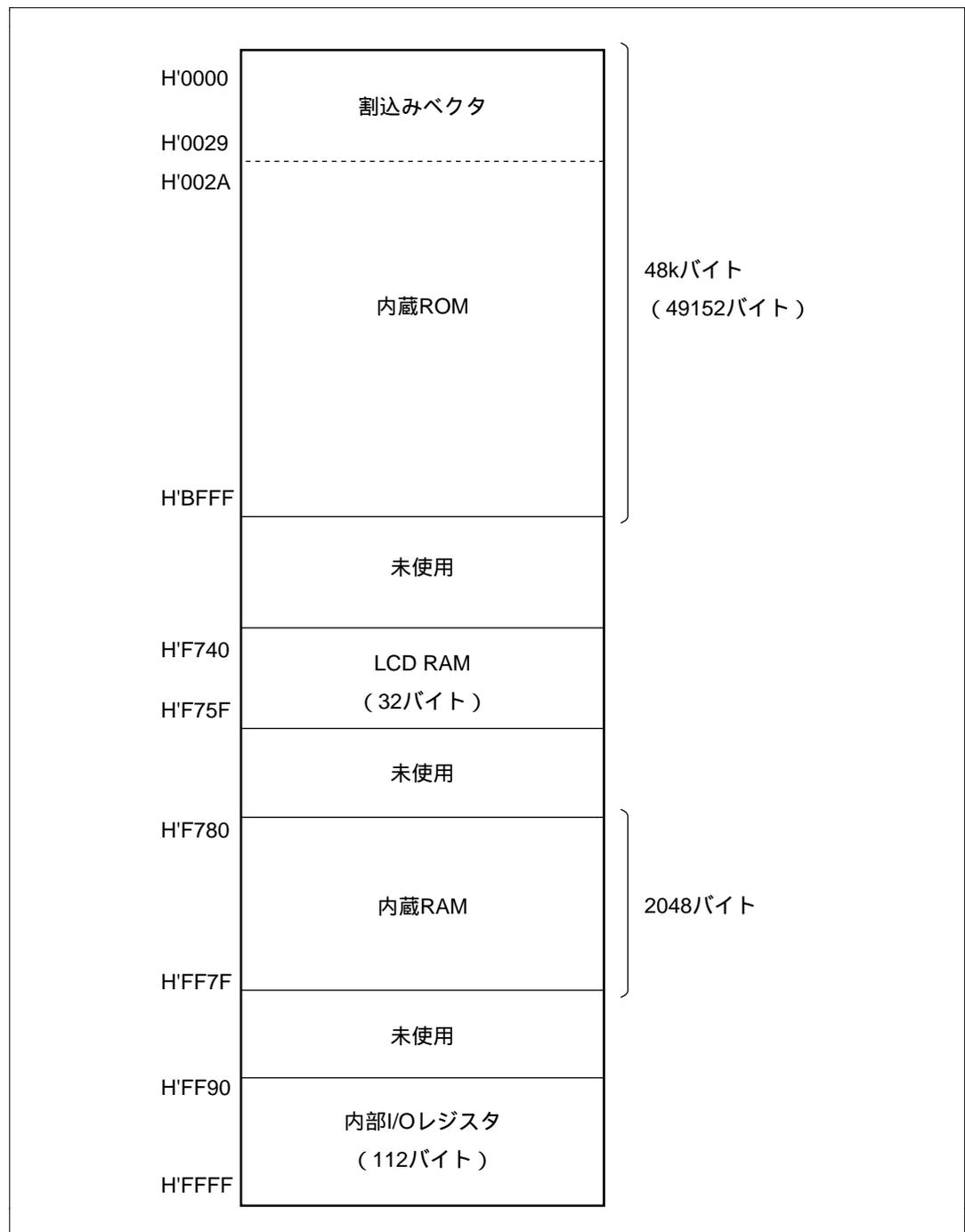


図 2.16 (5) H8/3886、H8/3846 のメモリマップ

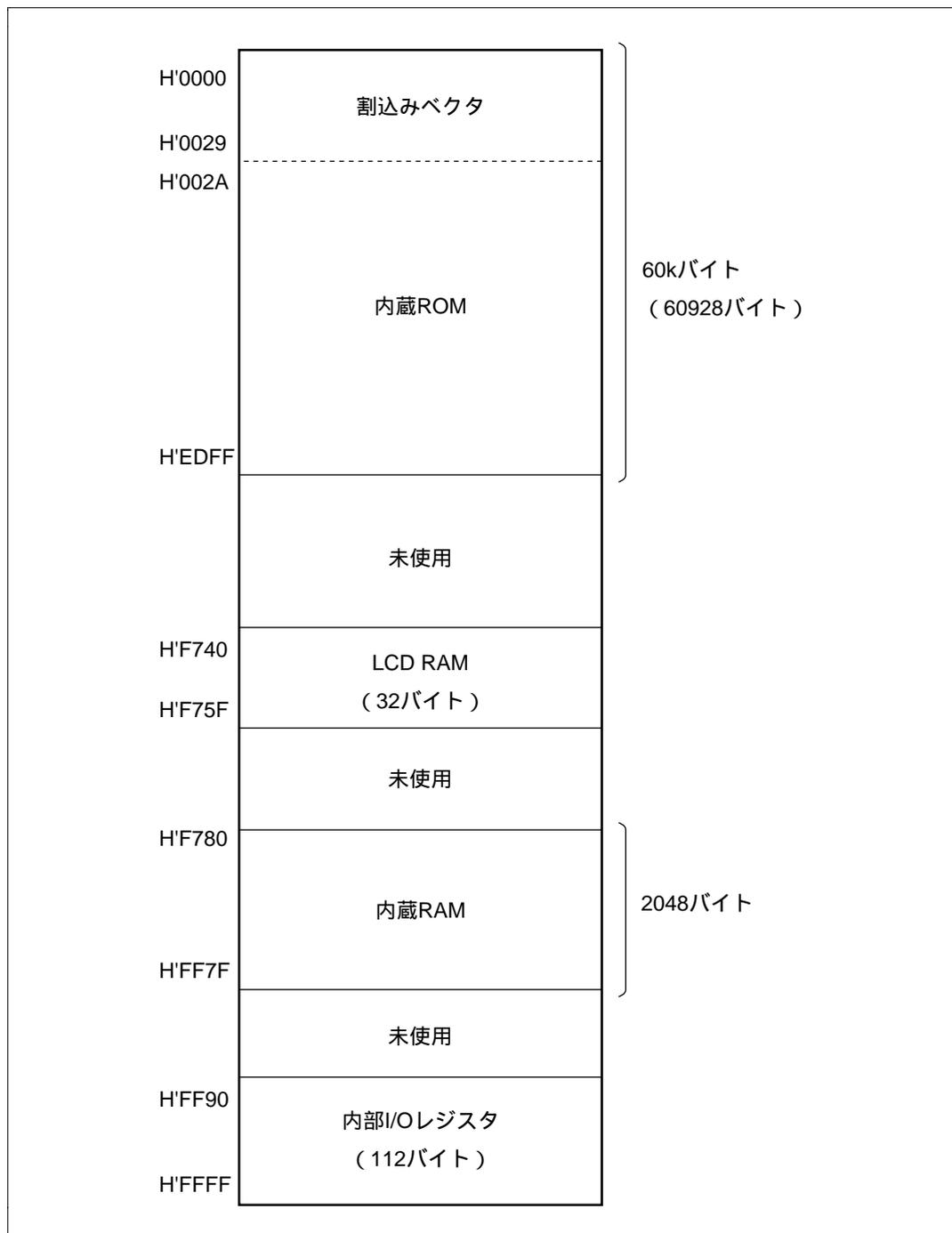


図 2.16 (6) H8/3887、H8/3847 のメモリマップ

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。

<u>CPU から空きエリアへのデータの転送</u>
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。
<u>空きエリアから CPU へのデータの転送</u>
転送データは保証されません。

(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。

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

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		ステート数
		ワード	バイト	
H'0000	割込みベクタ (42バイト)			
H'0029				
H'002A				
H'7FFF	内蔵ROM			2
H'7FFF				
H'7FFF	未使用	—	—	—
H'F740	LCD RAM (32バイト)			2
H'F75F				
H'F75F	未使用	—	—	—
H'F780	内蔵RAM			2
H'FF7F				
H'FF7F	未使用	—	—	—
H'FF90	内部I/Oレジスタ (112バイト)	x		2
H'FF98 ~ H'FF9F		x		3
H'FF9F		x		2
H'FFA8 ~ H'FFAF		x		3
H'FFFF		x		2

32kバイト*1

2048バイト*2

【注】 H8/3884、H8/3844の例です。
 *1 H8/3882、H8/3842は16kバイトでアドレスはH'3FFF、H8/3883、H8/3843は24kバイトでアドレスはH'5FFF、H8/3885、H8/3845は40kバイトでアドレスはH'9FFF、H8/3886、H8/3846は48kバイトでアドレスはH'BFFF、H8/3887、H8/3847は60kバイトでアドレスはH'EDFFとなります。
 *2 H8/3882、H8/3842、H8/3883、H8/3843は1024バイトでアドレスはH'FB7Fとなります。

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。

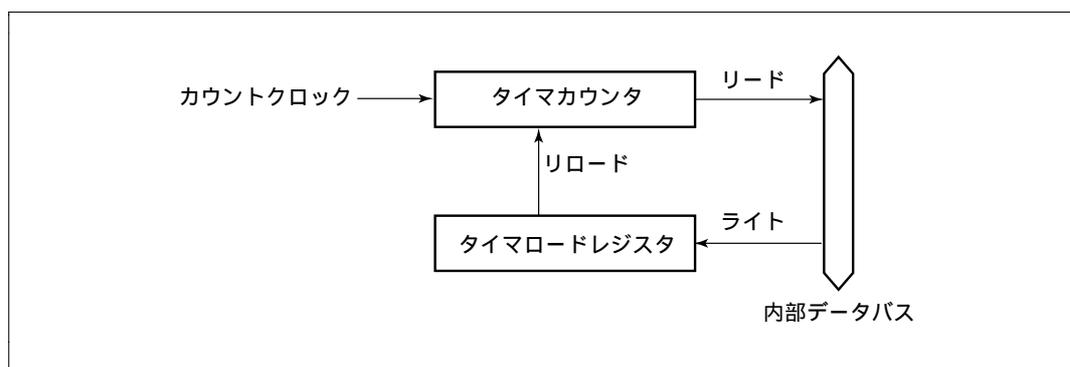


図 2.18 タイマの構成例

例2：ポート3にBSET命令を実行した場合

P3₇、P3₆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、P3₅~P3₀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P3₀に"High"レベル出力を行う例を示します。

【A； BSET命令を実行前】

	P3 ₇	P3 ₆	P3 ₅	P3 ₄	P3 ₃	P3 ₂	P3 ₁	P3 ₀
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
PCR3	0	0	1	1	1	1	1	1
PDR3	1	0	0	0	0	0	0	0

【B； BSET命令を実行】

BSET	#0,	@PDR3
------	-----	-------

ポート3に対してBSET命令を実行します。

【C； BSET命令を実行後】

	P3 ₇	P3 ₆	P3 ₅	P3 ₄	P3 ₃	P3 ₂	P3 ₁	P3 ₀
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
PCR3	0	0	1	1	1	1	1	1
PDR3	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3をリードします。

P3₇、P3₆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。P3₅~P3₀は出力端子であるので、CPUはPDR3の値をリードします。したがって、この例では、PDR3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PDR3に書き込んで、BSET命令を終了します。

その結果、PDR3のビット0が"1"になり、P3₀は"High"レベル出力になります。しかし、PDR3のビット7、6が変化してしまいます。

そのため、PDR3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PDR3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, @PDR3
```

PDR3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PDR3にライトします。

	P3 ₇	P3 ₆	P3 ₅	P3 ₄	P3 ₃	P3 ₂	P3 ₁	P3 ₀
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
PCR3	0	0	1	1	1	1	1	1
PDR3	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
```

PDR3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, @PDR3
```

ワークエリア(RAM0)の値をPDR3にライトします。

	P3 ₇	P3 ₆	P3 ₅	P3 ₄	P3 ₃	P3 ₂	P3 ₁	P3 ₀
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
PCR3	0	0	1	1	1	1	1	1
PDR3	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3のPCR3にBCLR命令を実行した場合

P3₇、P3₆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ます。P3₅～P3₀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、P3₀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P3₀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】

	P3 ₇	P3 ₆	P3 ₅	P3 ₄	P3 ₃	P3 ₂	P3 ₁	P3 ₀
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
PCR3	0	0	1	1	1	1	1	1
PDR3	1	0	0	0	0	0	0	0

【B； BCLR命令を実行】

BCLR #0 , @PCR3 PCR3 に対して BCLR 命令を実行します。

【C； BCLR命令を実行後】

	P3 ₇	P3 ₆	P3 ₅	P3 ₄	P3 ₃	P3 ₂	P3 ₁	P3 ₀
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
PCR3	1	1	1	1	1	1	1	0
PDR3	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PCR3をリードします。PCR3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PCR3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PCR3に書き込んで、BCLR命令を終了します。

その結果、PCR3のビット0が"0"になり、P3₀は入力ポートになります。しかし、PCR3のビット7、ビット6が1になって、入力ポートであったP3₇、P3₆は出力ポートに変化してしまいます。

そのため、PCR3 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 PCR3 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, @PCR3
```

PCR3 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 PCR3 にライトします。

	P3 ₇	P3 ₆	P3 ₅	P3 ₄	P3 ₃	P3 ₂	P3 ₁	P3 ₀
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
PCR3	0	0	1	1	1	1	1	1
PDR3	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
```

PCR3 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, @PCR3
```

ワークエリア (RAM0) の値を PCR3 にライトします。

	P3 ₇	P3 ₆	P3 ₅	P3 ₄	P3 ₃	P3 ₂	P3 ₁	P3 ₀
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
PCR3	0	0	1	1	1	1	1	0
PDR3	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
タイマカウンタC/タイマロードレジスタC	TCC/TLC	H'FFB5
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3*	PDR3	H'FFD6
ポートデータレジスタ4*	PDR4	H'FFD7
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9*	PDR9	H'FFDC
ポートデータレジスタA*	PDRA	H'FFDD

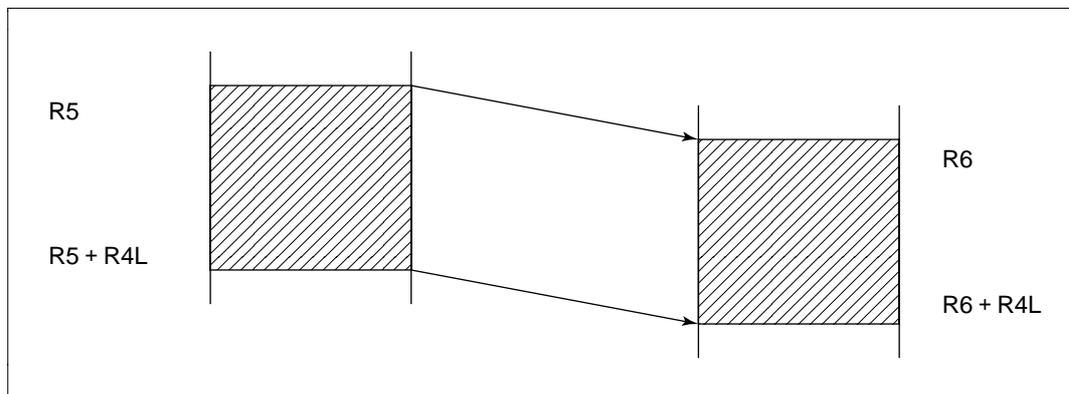
【注】 * ポートデータレジスタと端子入力が兼用になっています。

表2.13 ライト専用ビットを含むレジスタの一覧

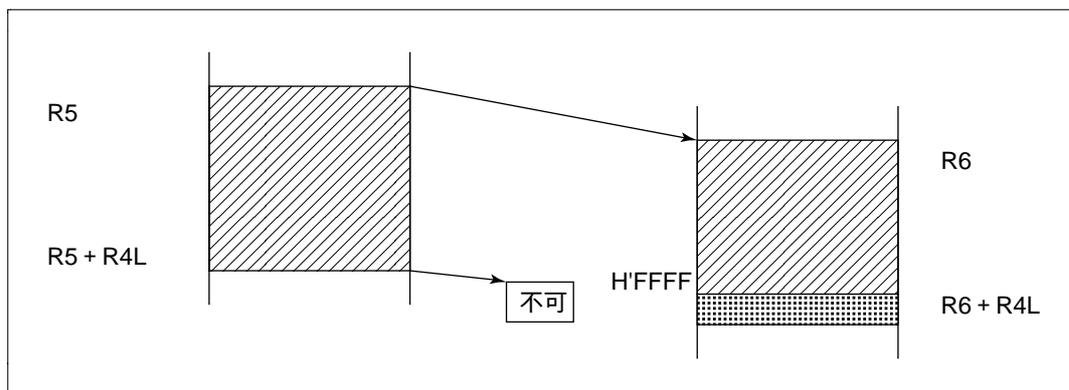
レジスタ名	略称	アドレス
ポートコントロールレジスタ1	PCR1	H'FFE4
ポートコントロールレジスタ2	PCR2	H'FFE5
ポートコントロールレジスタ3	PCR3	H'FFE6
ポートコントロールレジスタ4	PCR4	H'FFE7
ポートコントロールレジスタ5	PCR5	H'FFE8
ポートコントロールレジスタ6	PCR6	H'FFE9
ポートコントロールレジスタ7	PCR7	H'FFEA
ポートコントロールレジスタ8	PCR8	H'FFEB
ポートコントロールレジスタ9	PCR9	H'FFEC
ポートコントロールレジスタA	PCRA	H'FFED
タイマコントロールレジスタF	TCRF	H'FFB6
PWMコントロールレジスタ	PWCR	H'FFD0
PWMデータレジスタU	PWDRU	H'FFD1
PWMデータレジスタL	PWDRL	H'FFD2

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.....	69
3.2	リセット.....	70
	3.2.1 概要.....	70
	3.2.2 リセットシーケンス.....	70
	3.2.3 リセット直後の割込み.....	72
3.3	割込み.....	73
	3.3.1 概要.....	73
	3.3.2 各レジスタの説明.....	75
	3.3.3 外部割込み.....	85
	3.3.4 内部割込み.....	86
	3.3.5 割込み動作.....	86
	3.3.6 割込み応答時間.....	91
3.4	使用上の注意事項.....	92
	3.4.1 スタック領域に関する使用上の注意事項.....	92
	3.4.2 ポートモードレジスタを書換える際の注意事項.....	93

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、動作中にリセットする場合は、最低 10 システムクロックの間、"Low"レベルに保持してください。

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。

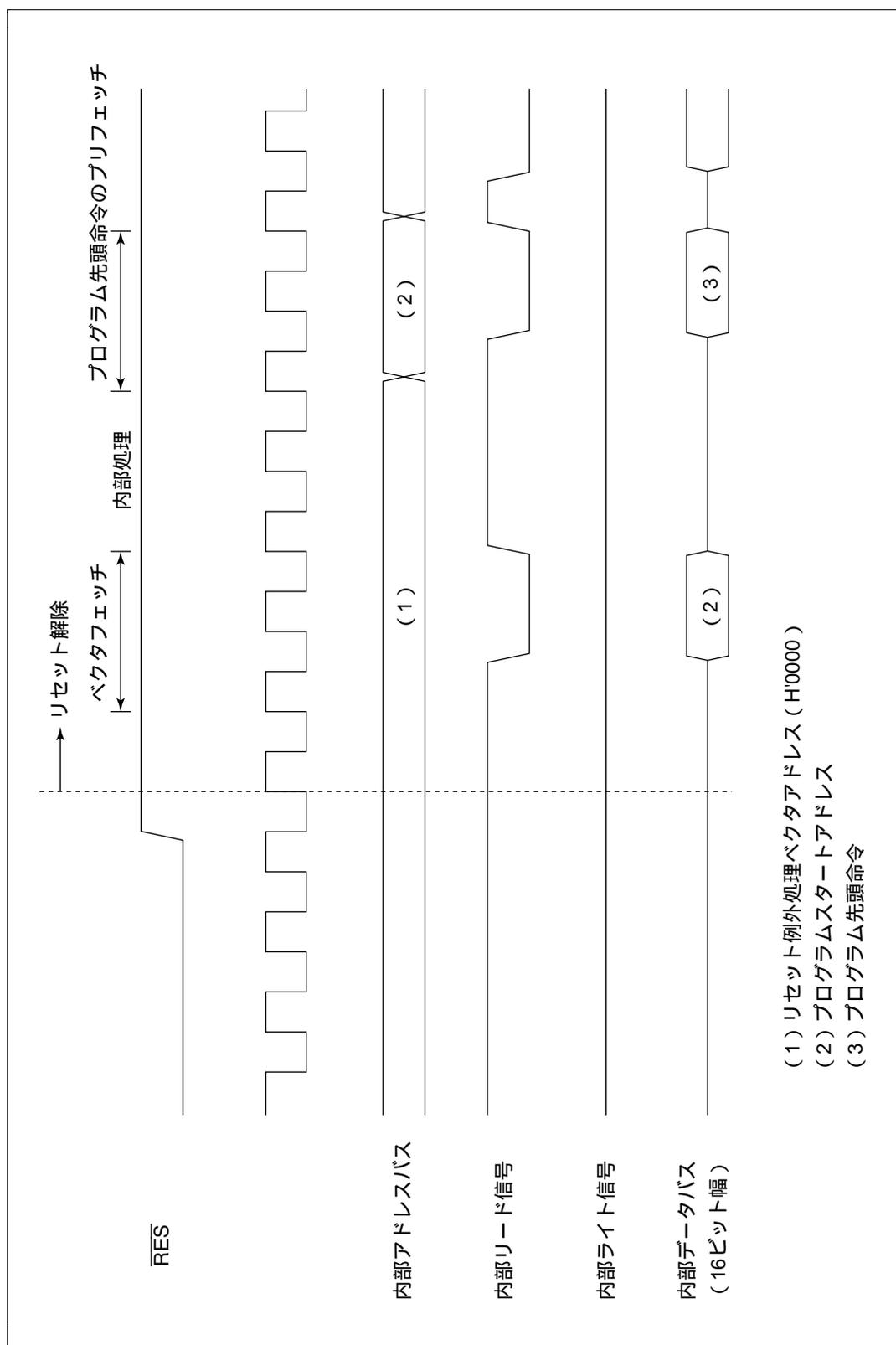


図 3.1 リセットシーケンス

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(WKP₇~WKP₀、IRQ₄~IRQ₀)と内蔵モジュールから24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。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) IRQ₄~IRQ₀、WKP₇~WKP₀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。

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
IRQ エッジセレクトレジスタ	IEGR	R/W	H'E0	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'00	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'00	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
ウェイクアップエッジセレクトレジスタ	WEGR	R/W	H'00	H'FF90

【注】 * フラグクリアのための"0"ライトのみ可能です。

(1) IRQ 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:	1	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 7~5 : 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 : IRQ4 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	$\overline{\text{IRQ}}_4$ 、 $\overline{\text{ADTRG}}$ 端子入力の下下がりエッジを検出 (初期値)
1	$\overline{\text{IRQ}}_4$ 、 $\overline{\text{ADTRG}}$ 端子入力の立上がりエッジを検出

ビット3 : IRQ₃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	$\overline{\text{IRQ}}_3$ 、TMIF 端子入力の立下がりエッジを検出	(初期値)
1	$\overline{\text{IRQ}}_3$ 、TMIF 端子入力の立上がりエッジを検出	

ビット2 : IRQ₂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 : IRQ₁エッジセレクト (IEG1)

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

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

ビット0 : IRQ₀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)

WKP₇ ~ WKP₀ 割込み要求の許可 / 禁止を制御します。

ビット 5	
IENWP	説明
0	WKP ₇ ~ WKP ₀ 端子の割込み要求を禁止 (初期値)
1	WKP ₇ ~ WKP ₀ 端子の割込み要求を許可

ビット 4 ~ 0 : IRQ₄ ~ IRQ₀ 割込みイネーブル (IEN4 ~ IEN0)

IRQ₄ ~ IRQ₀ 割込み要求の許可 / 禁止を制御します。

ビット n	
IENn	説明
0	IRQ _n 端子の割込み要求を禁止 (初期値)
1	IRQ _n 端子の割込み要求を許可

(n = 4 ~ 0)

(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	IENTC	IENEC
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

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

本ビットはリード/ライト可能なリザーブビットです。リセット時は"0"にイニシャライズされます。

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：タイマC割込みイネーブル (IENTC)

タイマC オーバフローまたはアンダフロー割込み要求の許可 / 禁止を制御します。

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

ビット0：非同期イベントカウンタ割込みイネーブル (IENEC)

非同期イベントカウンタの許可 / 禁止を制御します。

ビット0		
IENEC	説明	
0	非同期イベントカウンタの割込み要求を禁止	(初期値)
1	非同期イベントカウンタの割込み要求を許可	

なお、SCI3-1 および 3-2 の割込みの制御については「10.4.2 (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
	IRRTA	IRRS1	—	IRRI4	IRRI3	IRRI2	IRRI1	IRRI0
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。

IRRI は、8 ビットリード/ライト可能なレジスタで、タイマ A、SCI1、IRQ₄ ~ IRQ₀ 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〕 (初期値) IRRTA = "1"の状態 で 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

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〕 (初期値) IRRS1 = "1"の状態 で 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

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：IRQ₄～IRQ₀割込み要求フラグ（IRR14～IRR10）

ビット n	説明
IRRIn	
0	〔クリア条件〕 IRRIn = "1"の状態 で IRRIn に "0" をライトしたとき (初期値)
1	〔セット条件〕 IRQ _n 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	IRRTC	IRREC
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。

80

IRR2 は、8 ビットリード/ライト可能なレジスタで、直接遷移、A/D 変換器、タイマ G、タイマ FH、タイマ FL、タイマ C 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〕 IRRDT = "1"の状態 で IRRDT に "0" をライトしたとき (初期値)
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	
IRRAD	説明
0	〔クリア条件〕 IRRAD = "1"の状態 で IRRAD に"0"をライトしたとき (初期値)
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

本ビットはリード/ライト可能なリザーブビットです。リセット時は"0"にイニシャライズされます。

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〕 IRRTG = "1"の状態 で IRRTG に"0"をライトしたとき (初期値)
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 がオーバーフローし 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〕 IRRTFH = "1"の状態 で IRRTFH に"0"をライトしたとき (初期値)
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。または、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：タイマC 割込み要求フラグ (IRRTC)

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

ビット0：非同期イベントカウンタ割込み要求フラグ (IRREC)

ビット0	
IRREC	説明
0	〔クリア条件〕 (初期値) IRREC="1"の状態 で IRRECに"0"をライトしたとき
1	〔セット条件〕 16ビットカウンタモードで ECHがオーバーフローしたとき。または8ビットカウンタモードで ECHまたは ECLがオーバーフローしたとき

(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、当該端子に立上がりまたは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
0	[クリア条件] (初期値) 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、かつ当該端子に立上がり、または立下がりエッジが入力されたとき

(n = 7 ~ 0)

(7) ウェイクアップエッジセレクトレジスタ (WEGR)

ビット:	7	6	5	4	3	2	1	0
	WKEGS7	WKEGS6	WKEGS5	WKEGS4	WKEGS3	WKEGS2	WKEGS1	WKEGS0
初期値:	0	0	0	0	0	0	0	0
R/W :	R/W							

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

リセット時、WEGR は H'00 にイニシャライズされます。

ビット n : \overline{WKPn} エッジセレクト

\overline{WKPn} 端子の入力センスを選択します。

ビット n	説明
0	\overline{WKPn} 端子の立下がりエッジを検出 (初期値)
1	\overline{WKPn} 端子の立上がりエッジを検出

(n = 0 ~ 7)

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 $\overline{WKP_7} \sim \overline{WKP_0}$ 端子の立上がり / 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上がり / 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。

PMR3、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。

IRQ₄ ~ IRQ₀ 割込み例外処理のベクタ番号は 8 ~ 4 です。優先順位は IRQ₄(低) IRQ₀(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、24 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。

割込みシーケンスを図 3.5 に示します。

- 【注】 1. 本 LSI では、割り込み許可レジスタをクリアすることにより割り込みをディスエーブルにする場合、または割り込み要求レジスタをクリアする場合は、かならず割り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り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り込みに対応する例外処理を実行します。

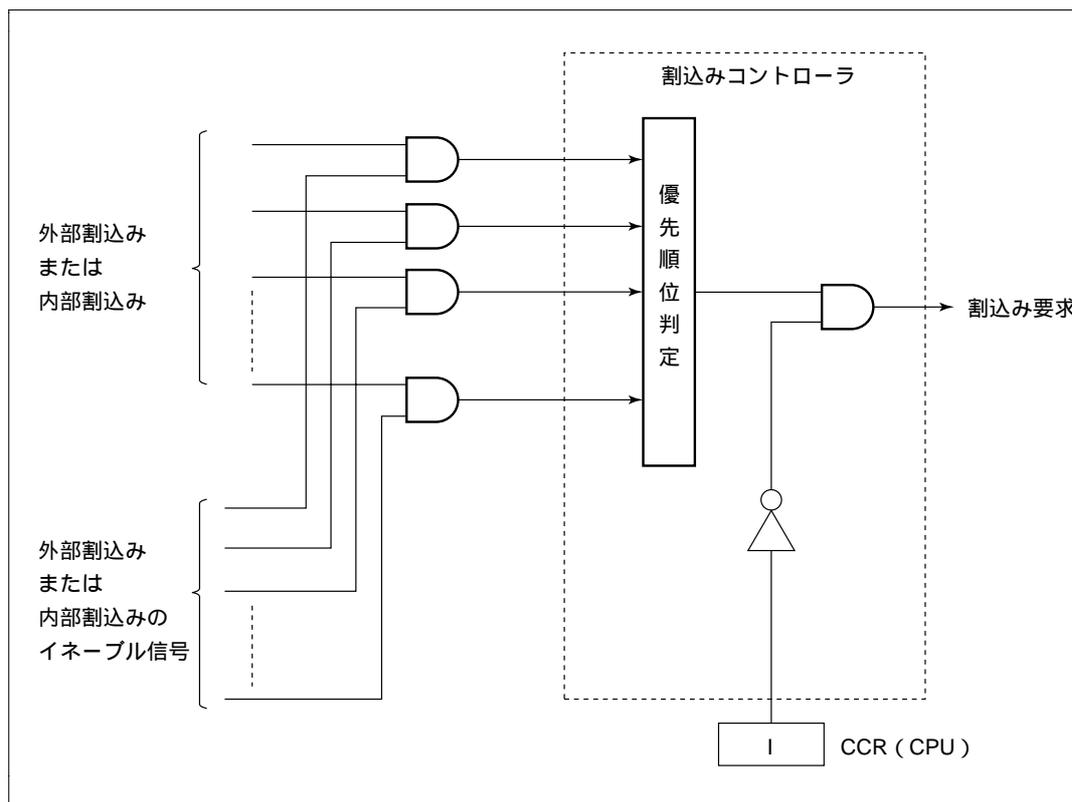


図 3.2 割り込みコントローラのブロック図

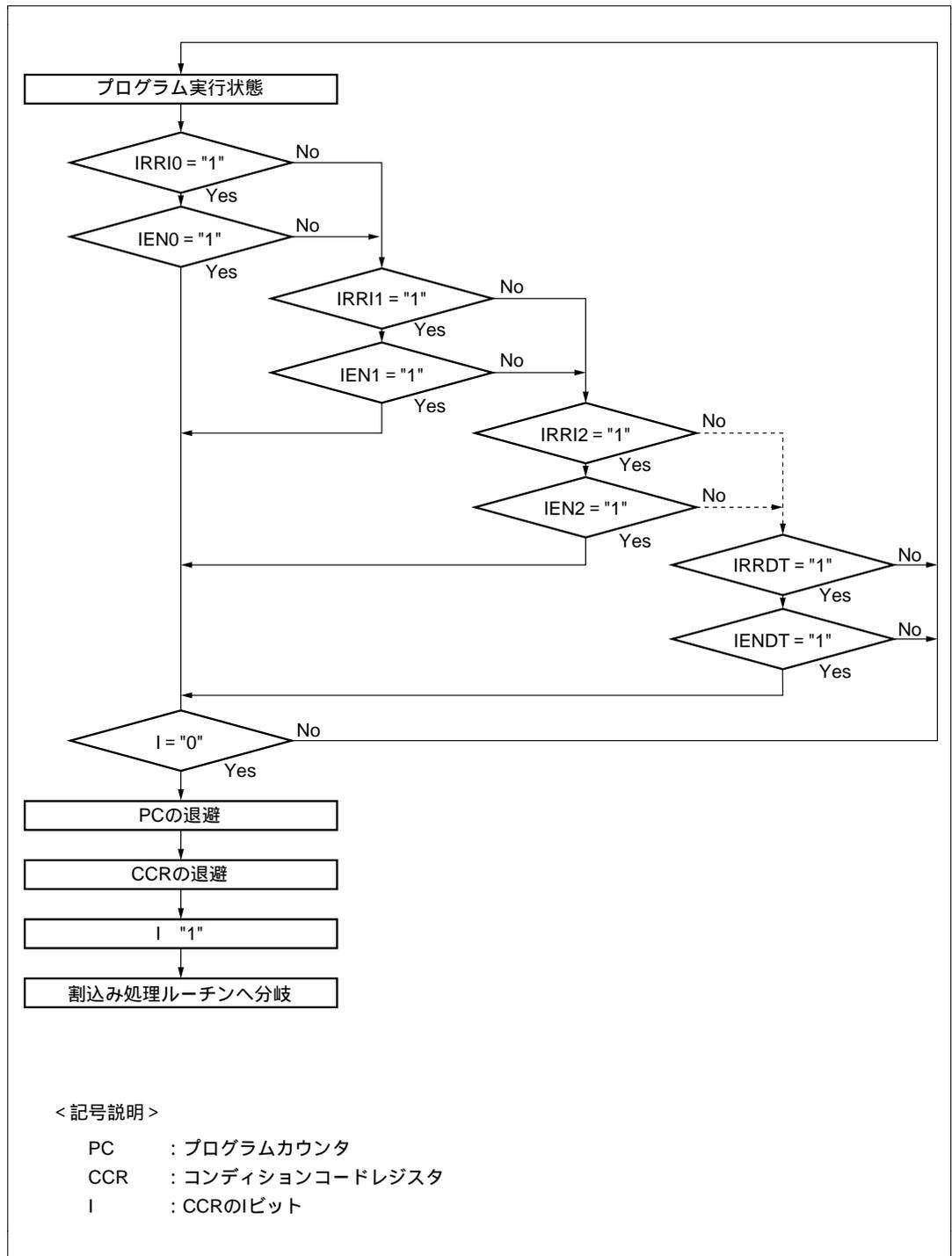


図 3.3 割り込み受けまでのフロー

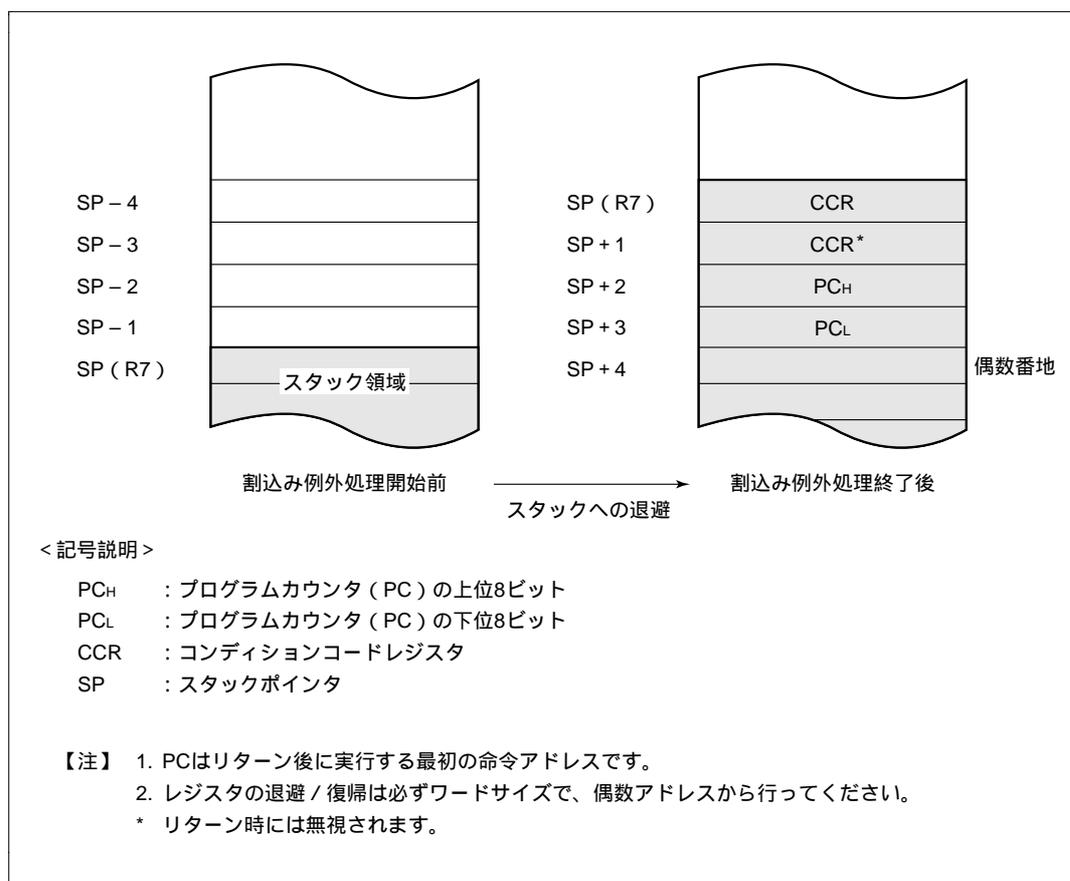


図 3.4 割込み例外処理終了後のスタック状態

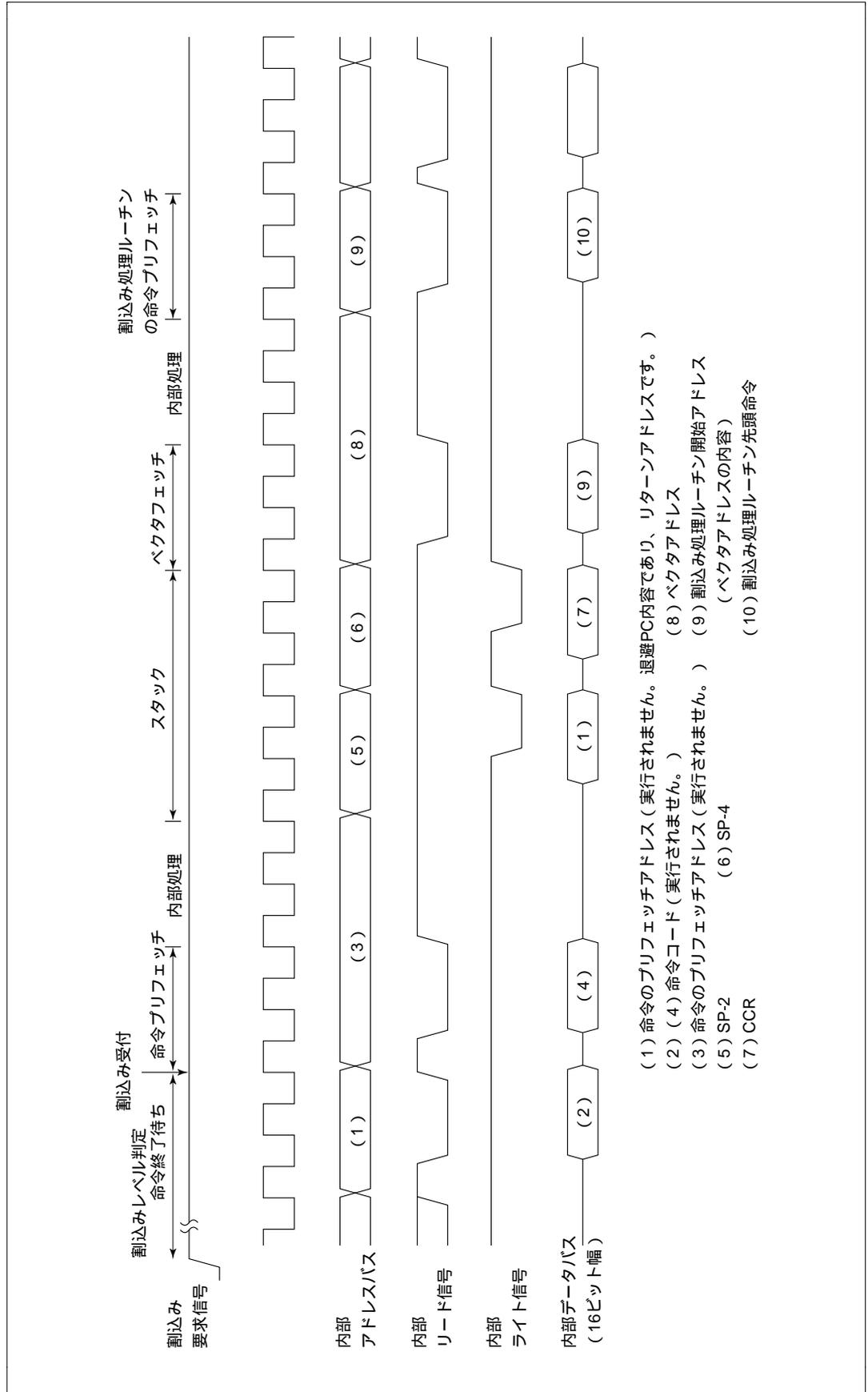


図 3.5 割り込みシーケンス

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。

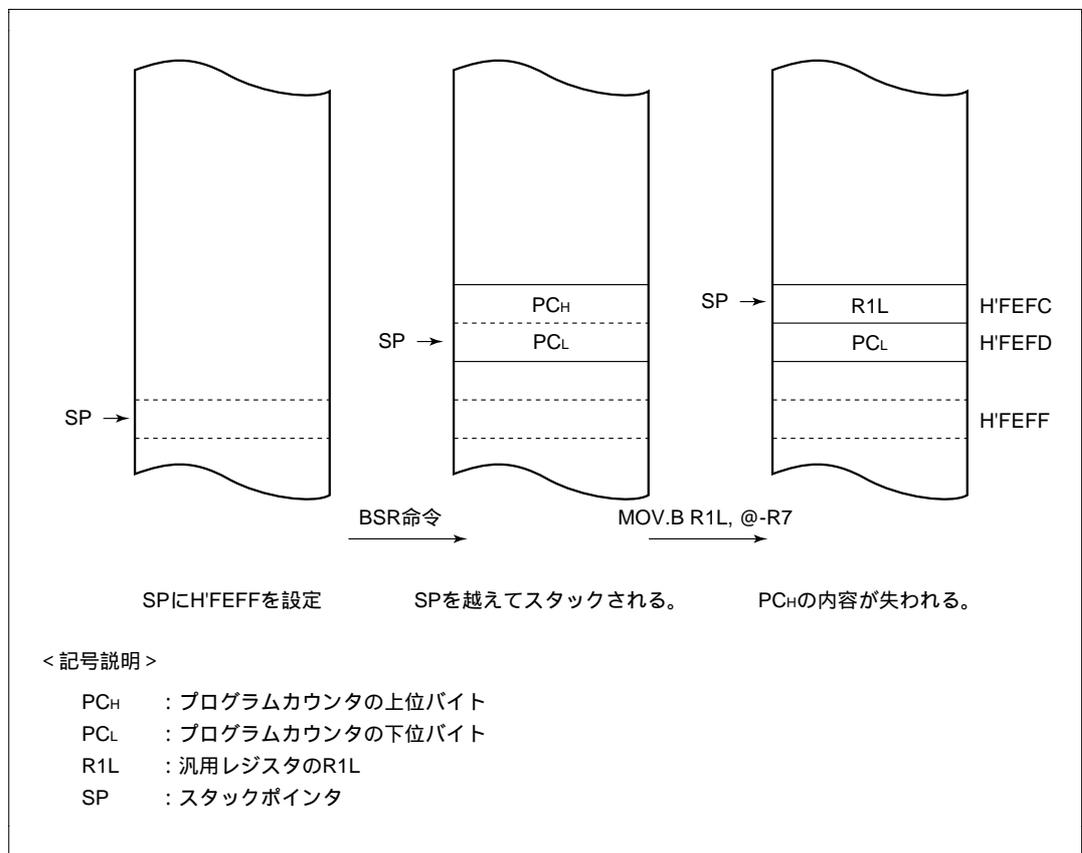


図 3.6 SP に奇数を設定したときの動作

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"> • \overline{IRQ}_4 端子が"Low"レベルでIEGRのIEG4が"0"の状態、PMR1のIRQ4を"0"から"1"に書換えたとき • \overline{IRQ}_4 端子が"Low"レベルでIEGRのIEG4が"1"の状態、PMR1のIRQ4を"1"から"0"に書換えたとき
	IRRI3	<ul style="list-style-type: none"> • \overline{IRQ}_3 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 • \overline{IRQ}_3 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
	IRRI2	<ul style="list-style-type: none"> • \overline{IRQ}_2 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 • \overline{IRQ}_2 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
	IRRI1	<ul style="list-style-type: none"> • \overline{IRQ}_1 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 • \overline{IRQ}_1 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
	IRRI0	<ul style="list-style-type: none"> • \overline{IRQ}_0 端子が"Low"レベルでIEGRのIEG0が"0"の状態、PMR3のIRQ0を"0"から"1"に書換えたとき • \overline{IRQ}_0 端子が"Low"レベルでIEGRのIEG0が"1"の状態、PMR3のIRQ0を"1"から"0"に書換えたとき
	IWPR	IWPF7
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。

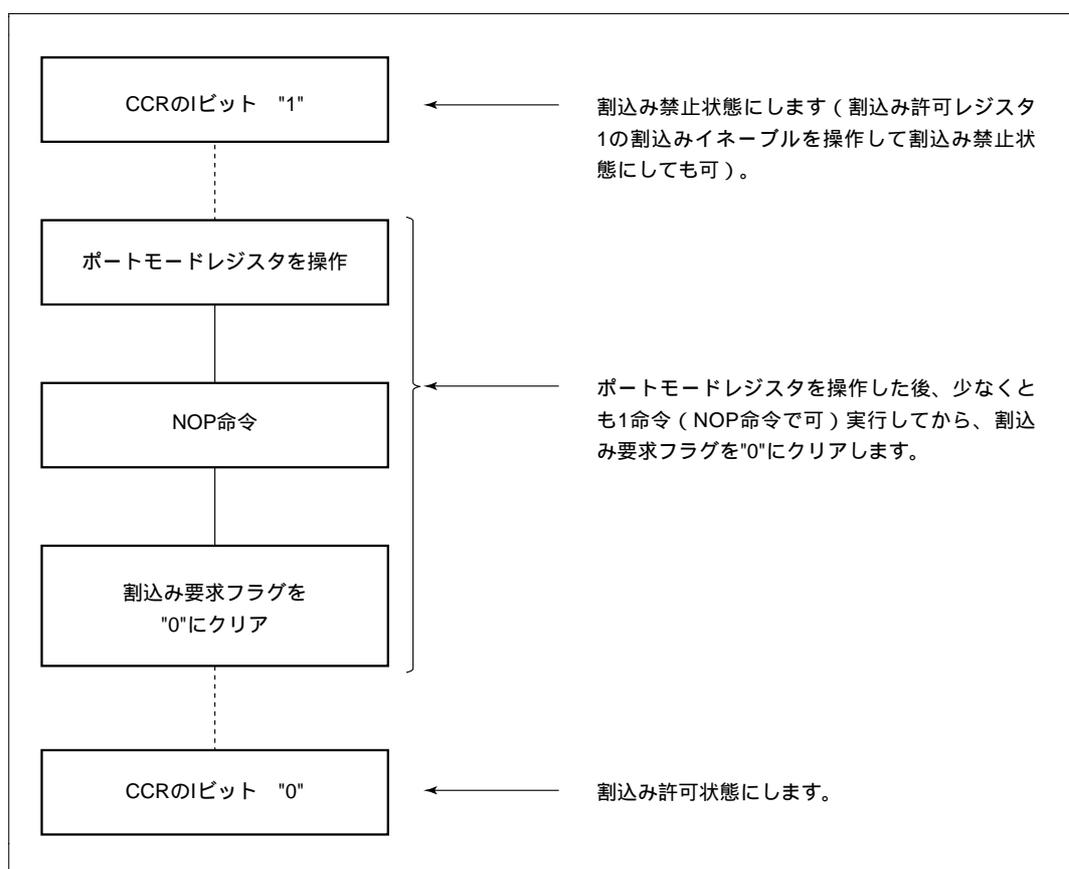


図 3.7 ポートモードレジスタ操作と割り込み要求フラグのクリア手順

4. クロック発振器

第4章 目次

4.1	概要.....	99
	4.1.1 ブロック図.....	99
	4.1.2 システムクロックとサブクロック	99
4.2	システムクロック発振器.....	100
4.3	サブクロック発振器	103
4.4	プリスケーラ	105
4.5	発振子に関する注意事項.....	106

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。

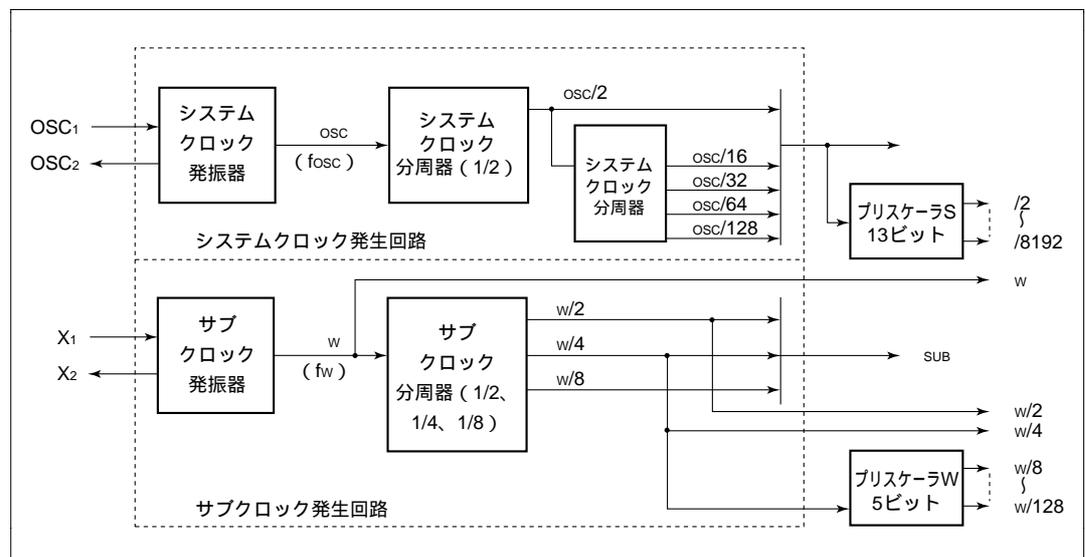


図 4.1 クロック発生回路のブロック図

4.1.2 システムクロックとサブクロック

クロック (および $_{SUB}$) 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。

$_{osc}$ をシステムクロック、 $_{SUB}$ をサブクロックと呼びます。また、 $_{osc}$ 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 $_{/8192}$ 、 $_{/4096}$ 、 $_{/2048}$ 、 $_{/1024}$ 、 $_{/512}$ 、 $_{/256}$ 、 $_{/128}$ 、 $_{/64}$ 、 $_{/32}$ 、 $_{/16}$ 、 $_{/8}$ 、 $_{/4}$ 、 $_{/2}$ 、 $_w/128$ 、 $_w/64$ 、 $_w/32$ 、 $_w/16$ 、 $_w/8$ 、 $_w/4$ 、 $_w/2$ 、 $_w$ 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。

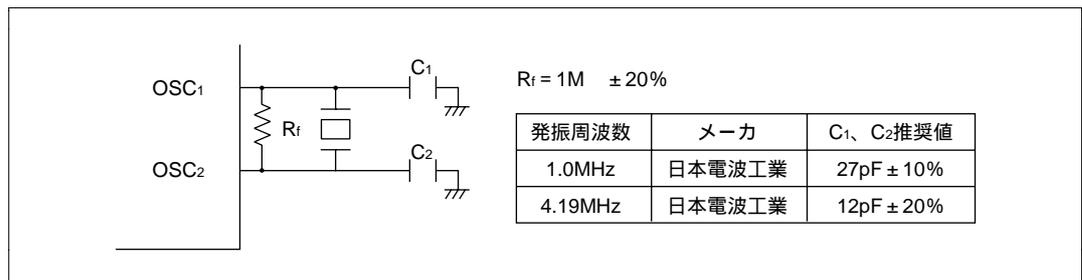


図 4.2 水晶発振子の接続例

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。

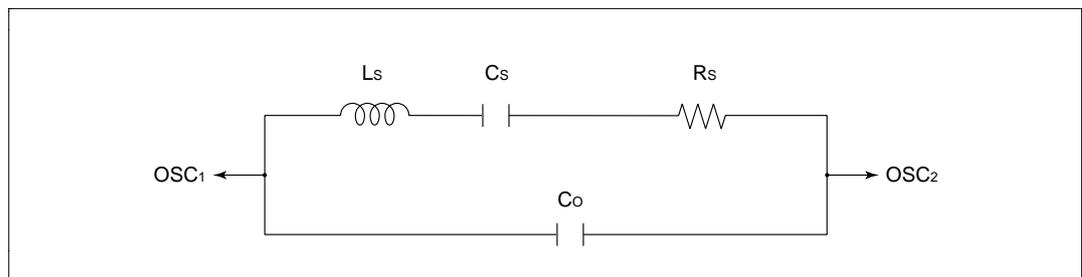


図 4.3 水晶発振子の等価回路

表 4.1 水晶発振子のパラメータ

周波数 (MHz)	1	4.193
R_s (max)	40	100
C_0 (max)	3.5pF	16pF

(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。

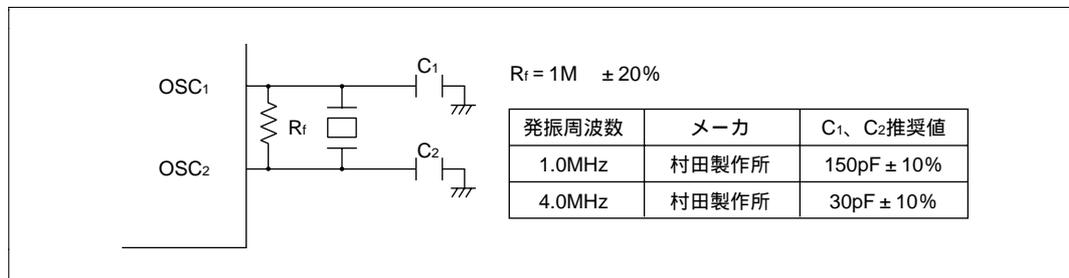


図 4.4 セラミック発振子の接続例

(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 OSC₁、OSC₂ 端子の近くに配置してください。

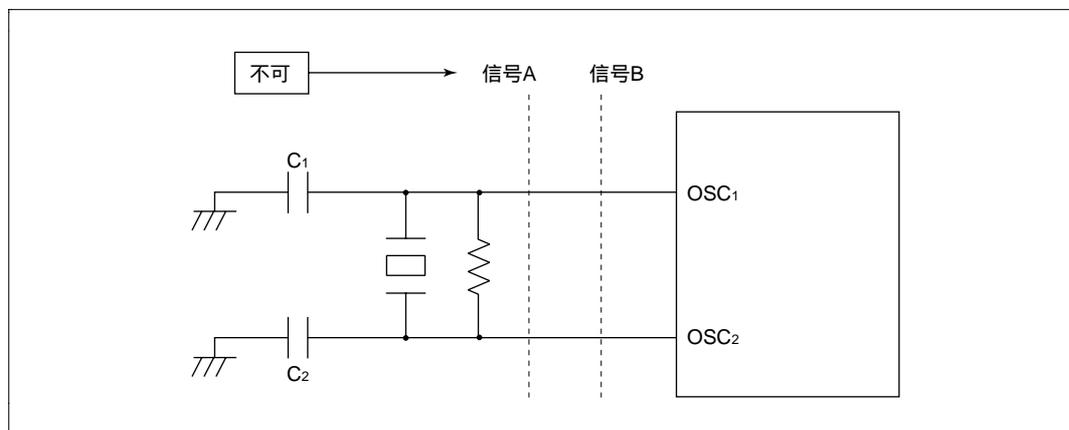


図 4.5 発振回路部のボード設計に関する注意事項

(4) 外部クロックを入力する方法

外部クロックを OSC₁ 端子に接続し、OSC₂ 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。

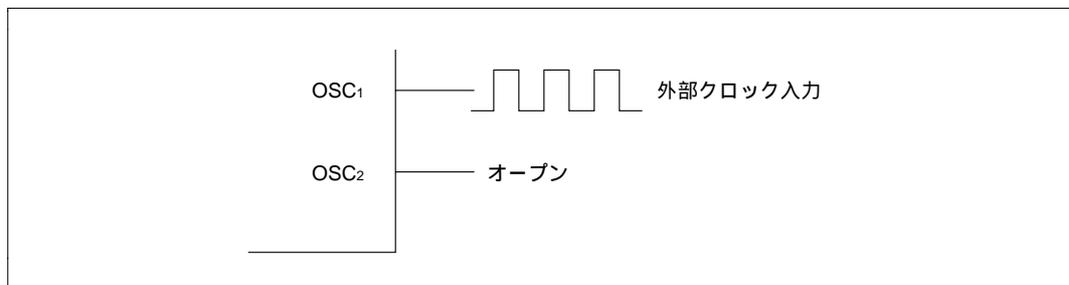


図 4.6 外部クロックを入力する場合の接続例

周波数	OSC クロック (_{osc})
duty	45% ~ 55%

4.3 サブクロック発振器

(1) 32.768kHz/38.4kHz 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 または 38.4kHz 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。

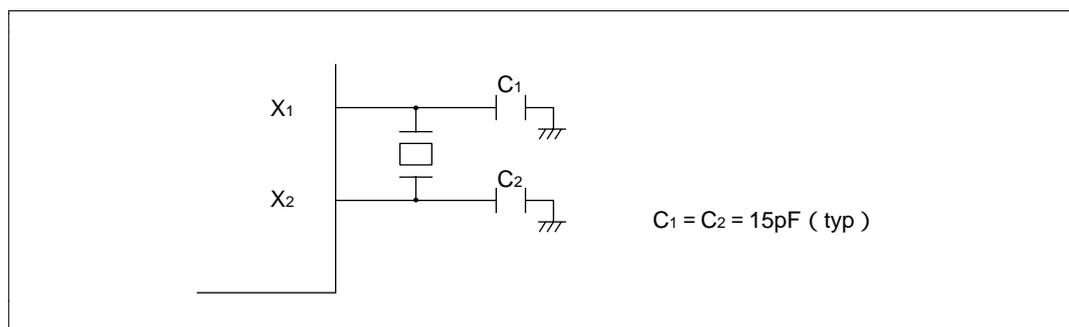


図 4.7 32.768kHz/38.4kHz 水晶発振子の接続例

図 4.8 に水晶発振子の等価回路を示します。

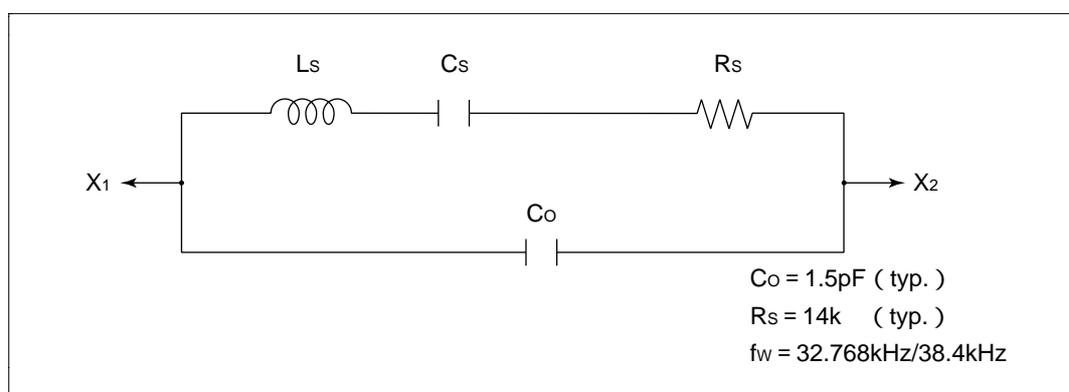


図 4.8 32.768kHz/38.4kHz 水晶発振子の等価回路

(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 GND に接続し、X₂ 端子をオープンとしてください。

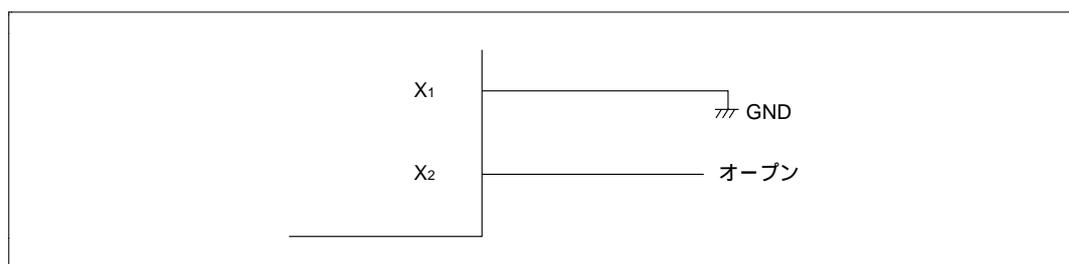


図 4.9 サブクロックを必要としない場合の端子処理

(3) 外部クロックを入力する

外部クロックを X_1 端子に接続し、 X_2 端子をオープン状態にします。

この場合の接続例を図 4.10 に示します。

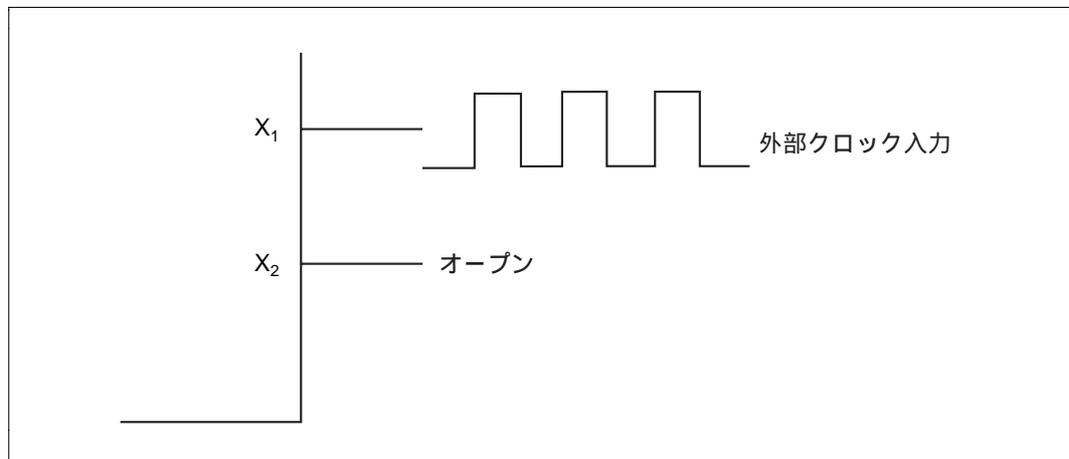


図 4.10 外部クロックを入力する場合の接続例

周波数	サブクロック (ϕ_w)
duty	45% ~ 55%

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 または 38.4kHz 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、タイマ A、タイマ C、タイマ F、タイマ G、SCI1、SCI3-1、SCI3-2、A/D 変換器、LCD コントローラ、ウォッチドッグタイマおよび 14 ビット PWM 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 $f_{osc}/16$ 、 $f_{osc}/32$ 、 $f_{osc}/64$ または $f_{osc}/128$ となります。

(2) プリスケーラ W (PSW)

プリスケーラ W は、32.768kHz または 38.4kHz 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、X1、X2 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"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を参考に、マスク ROM 版、ZTAT[®]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。

5. 低消費電力モード

第5章 目次

5.1	概要.....	109
	5.1.1	システムコントロールレジスタ..... 112
5.2	スリープモード.....	116
	5.2.1	スリープモードへの遷移..... 116
	5.2.2	スリープモードの解除..... 116
	5.2.3	スリープ（中速）モードの動作周波数について..... 117
5.3	スタンバイモード.....	118
	5.3.1	スタンバイモードへの遷移..... 118
	5.3.2	スタンバイモードの解除..... 118
	5.3.3	スタンバイモード解除後の発振安定時間の設定..... 119
	5.3.4	スタンバイモードへの遷移と端子状態..... 120
5.4	ウォッチモード.....	121
	5.4.1	ウォッチモードへの遷移..... 121
	5.4.2	ウォッチモードの解除..... 121
	5.4.3	ウォッチモード解除後の発振安定時間の設定..... 121
5.5	サブスリープモード.....	122
	5.5.1	サブスリープモードへの遷移..... 122
	5.5.2	サブスリープモードの解除..... 122
5.6	サブアクティブモード.....	123
	5.6.1	サブアクティブモードへの遷移..... 123
	5.6.2	サブアクティブモードの解除..... 123
	5.6.3	サブアクティブモードの動作周波数について..... 123
5.7	アクティブ（中速）モード.....	124
	5.7.1	アクティブ（中速）モードへの遷移..... 124
	5.7.2	アクティブ（中速）モードの解除..... 124
	5.7.3	アクティブ（中速）モードの動作周波数について..... 124
5.8	直接遷移.....	125

5. 低消費電力モード

5.8.1	直接遷移の概要	125
5.8.2	直接遷移の時間	126
5.9	モジュールスタンバイモード	129
5.9.1	モジュールスタンバイモードの設定	129
5.9.2	モジュールスタンバイモードの解除	129

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 8 種類の低消費電力モードを含む、9 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（高速）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。
スリープ（中速）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の 1/64、1/32、1/16、1/8 の周波数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、タイマ C、タイマ G、タイマ F、WDT、SCI1、SCI3-1、SCI3-2、AEC および LCD コントローラ/ドライバ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、タイマ F、タイマ G、AEC および LCD コントローラ/ドライバ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。
モジュールスタンバイモード	ソフトウェアによって指定された内蔵の周辺機能ごとにスタンバイモードの状態になり、動作を停止しているモードです。

上記 9 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。

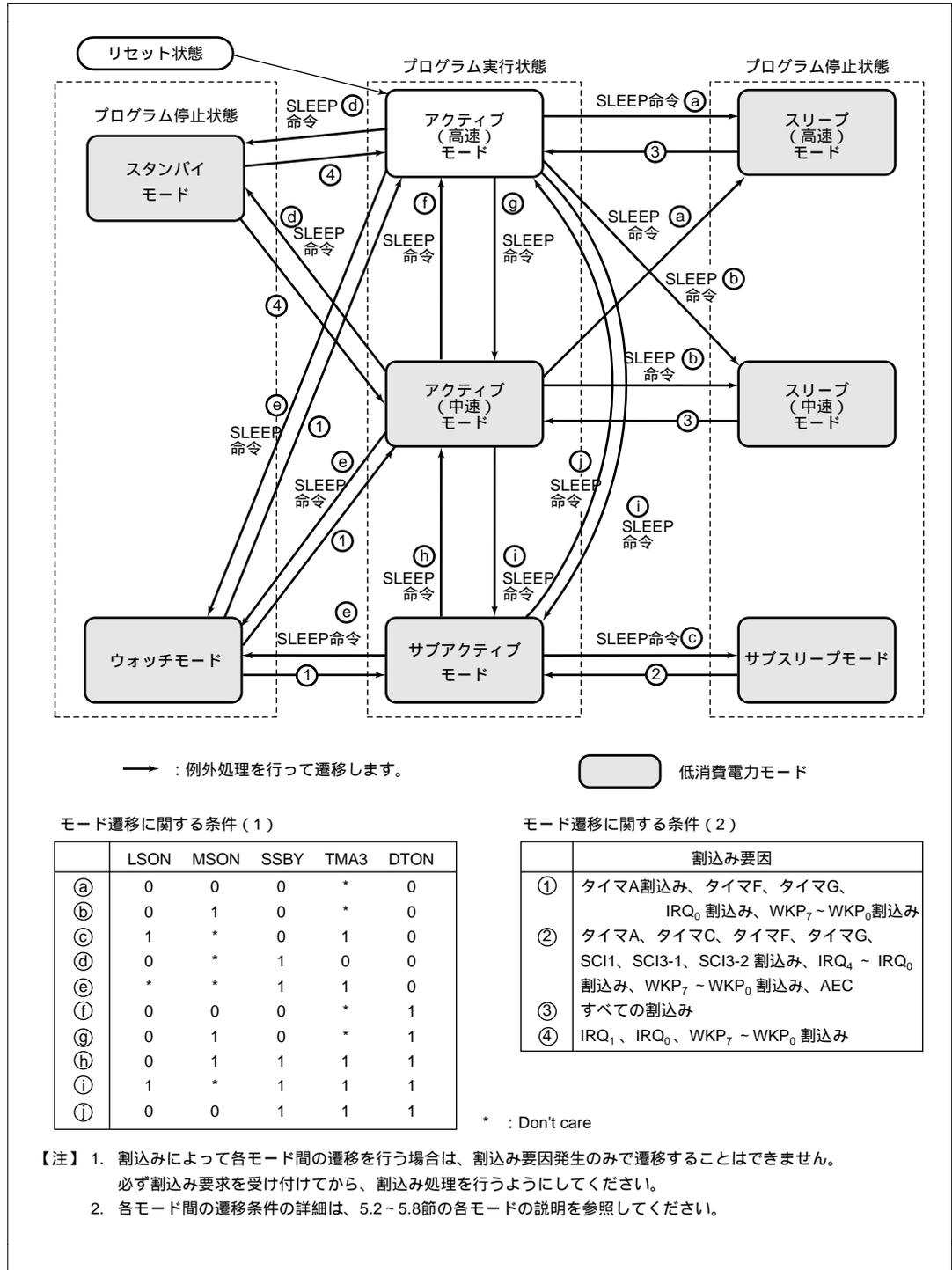


図 5.1 モード遷移図

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	高速	中速								
システムクロック発振器		動作	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	動作				
CPU 動作	命令	動作	動作	停止	停止	停止	動作	停止	停止				
	RAM			保持	保持	保持		保持					
	レジスタ			保持	保持	保持		保持					
	I/O							保持*1					
外部 割込み の動作	IRQ ₀	動作	動作	動作	動作	動作	動作	動作	動作				
	IRQ ₁					保持*6			動作	動作	保持*6		
	IRQ ₂												
	IRQ ₃												
	IRQ ₄	動作	動作	動作	動作	動作	動作	動作					
	WKP ₀												
	WKP ₁												
	WKP ₂												
	WKP ₃												
	WKP ₄												
WKP ₅													
WKP ₆													
WKP ₇													
周辺機能 の動作	タイマ A	動作	動作	動作	動作	動作*5	動作*5	動作*5	保持				
	非同期カウンタ					動作*8	動作	動作	動作*8				
	タイマ C					保持	動作 / 保持*2	動作 / 保持*2	保持				
	WDT					動作 / 保持*7	保持						
	タイマ G、タイマ F	動作	動作	動作	動作	動作 / 保持*9	動作 / 保持*9	動作 / 保持*9					
	SCI1					保持	動作 / 保持*9	動作 / 保持*9	保持				
	SCI3-1、SCI3-2					リセット	動作 / 保持*3	動作 / 保持*3	リセット				
	PWM					保持	保持	保持	保持				
	A/D 変換器					動作	動作	動作	動作	保持	保持	保持	保持
	LCD					動作	動作	動作	動作	動作 / 保持*4	動作 / 保持*4	動作 / 保持*4	保持

【注】 *1 レジスタは保持、出力はハイインピーダンス

*2 外部クロックまたは内部クロックとして $w/4$ を選択した場合に動作、その他は停止して保持

*3 内部クロックとして $w/2$ を選択した場合に動作、その他は停止して保持

*4 使用クロックとして w 、 $w/2$ または $w/4$ を選択した場合に動作、その他は停止して保持

*5 時計用タイムベース機能を選択時に動作

*6 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。

*7 内部クロックとして $w/32$ を選択した場合は動作、その他は停止して保持

*8 カウントアップ可、割込み発生不可

*9 外部クロックまたは内部クロックとして $w/4$ を選択した場合に動作、その他は停止して保持

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'F0	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	—	MA1	MA0
初期値:	0	0	0	0	0	1	1	1
R/W :	R/W	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発振安定時間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 (初期状態)
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	0	0	待機時間 = 131,072 ステート
1	0	1	待機時間 = 2 ステート (外部クロック入力モード)
1	1	0	待機時間 = 8 ステート
1	1	1	待機時間 = 16 ステート

【注】 外部クロックを入力する場合、スタンバイタイムセレクトはモード遷移を実行する前に、外部クロック入力モードに設定してください。
また、外部クロックを使用しない場合、外部クロック入力モードに設定しないでください。

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}）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} ）

ビット2：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1、0：アクティブ（中速）モードクロックセレクト（MA1、MA0）

アクティブ（中速）モードまたはスリープ（中速）モードの動作クロック（ $osc/128$ 、 $osc/64$ 、 $osc/32$ 、 $osc/16$ ）を選択します。MA1、MA0の書込みは、アクティブ（高速）モードまたはサブアクティブモードで行ってください。

ビット1	ビット0	説明
MA1	MA0	
0	0	$osc/16$
0	1	$osc/32$
1	0	$osc/64$
1	1	$osc/128$ (初期値)

(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	1	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。

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（ 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（ osc ）により、サンプリングする周波数を選択します。 $osc = 2 \sim 6\text{MHz}$ のときは、"0"をセットしてください。

ビット4	説明
NESEL	
0	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 (初期値)

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"> ・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 ・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)
1	<ul style="list-style-type: none"> ・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 ・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 ・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

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

(1) スリープ(高速)モードへの遷移

アクティブモードで、SYSCR1のSSBYが"0"、LSONが"0"、SYSCR2のMSONが"0"、DTON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。

(2) スリープ(中速)モードへの遷移

アクティブモードで、SYSCR1のSSBYが"0"、LSONが"0"、SYSCR2のMSONが"1"、DTONが"0"のときSLEEP命令を実行すると、スリープ(中速)モードに遷移します。スリープ(中速)モードではスリープ(高速)モードと同様、CPUの動作は停止し、内蔵周辺モジュールは動作します。ただし、スリープ(中速)モードでは、SYSCR1のMA1、MA0で設定した周波数のクロックによって動作します。CPUのレジスタの内容は保持されます。

なお、スリープ(中速)モードへの遷移時に1/2ステート早いタイミングで動作することがあります。

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C、タイマF、タイマG、非同期カウンタ、IRQ₄~IRQ₀、WKP₇~WKP₀、SCI1、SCI3-1、SCI3-2、A/D変換器)、 $\overline{\text{RES}}$ 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、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。

スリープ(高速)モードからはアクティブ(高速)モードに、スリープ(中速)モードからはアクティブ(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リープ状態は解除されません。

なお、割込み要求信号とシステムクロックの同期をとるため、割込み要求信号発生から割込み例外処理開始までに最大2/(s)の遅れが生じることがあります。

(2) $\overline{\text{RES}}$ 端子による解除

$\overline{\text{RES}}$ 端子を"Low"レベルにするとリセット状態に遷移し、スリープモードは解除されません。

5.2.3 スリープ（中速）モードの動作周波数について

スリープ（中速）モードは、SYSCR1 の MA1、MA0 で設定した周波数のクロックによって動作します。

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 (IRQ₁、IRQ₀、WKP₇~WKP₀)、 $\overline{\text{RES}}$ 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付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。待機時間が発振安定時間以上となるように 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	2MHz	1MHz	0.5MHz
0	0	0	8,192 ステート	4.1	8.2	16.4
0	0	1	16,384 ステート	8.2	16.4	32.8
0	1	0	32,768 ステート	16.4	32.8	65.5
0	1	1	65,536 ステート	32.8	65.5	131.1
1	0	0	131,072 ステート	65.5	131.1	262.1
1	0	1	2 ステート (使用禁止)	0.001	0.002	0.004
1	1	0	8 ステート	0.004	0.008	0.016
1	1	1	16 ステート	0.008	0.016	0.032

(2) 外部クロックの場合

STS2 = "1"、STS1 = "0"、STS0 = "1" の使用を推奨します。他の設定も使用可能ですが、STS2 = "1"、STS1 = "0"、STS0 = "1" 以外の設定では、待機時間終了前に動作を開始することがあります。

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。

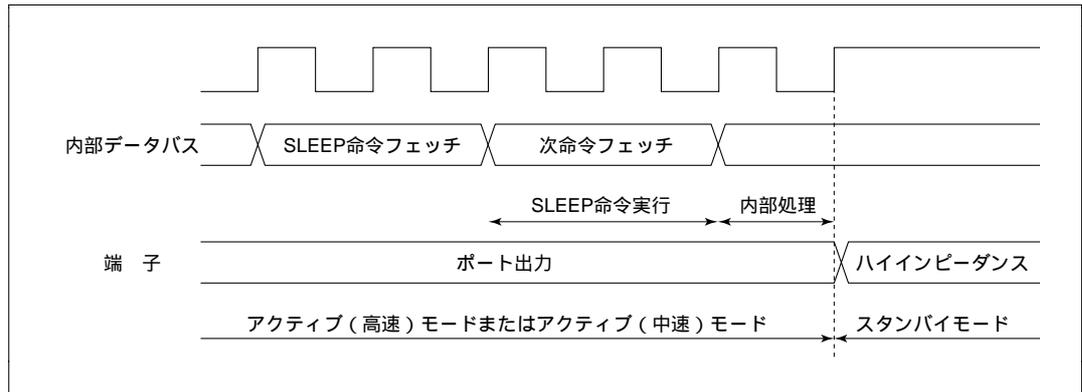


図 5.2 スタンバイモードへの遷移と端子状態

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、タイマ F、タイマ G、AEC、LCD（動作 / 停止選択可）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。

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（IRQ₀、WKP₇ ~ WKP₀、タイマ A、タイマ F、タイマ G）、 $\overline{\text{RES}}$ 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付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、A/D 変換器、PWM、ウォッチドッグタイマ以外の内蔵周辺機能は動作します。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。

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 C、タイマ F、タイマ G、非同期カウンタ、SCI1、SCI3-1、SCI3-2、IRQ₄ ~ IRQ₆、WKP₇ ~ WKP₀）、 $\overline{\text{RES}}$ 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。

なお、割込み要求信号とシステムクロックの同期をとるため、割込み要求信号発生から割込み例外処理開始までに最大 $2 / f_{\text{SUB}}$ (s) の遅れが生じることがあります。

(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、タイマ F、タイマ G、IRQ₀、WKP₇~WKP₀) 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 C、タイマ F、タイマ G、非同期カウンタ、SCI1、SCI3-1、SCI3-2、IRQ₄~IRQ₀、WKP₇~WKP₀) 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付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。

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 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 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（IRQ₁、IRQ₀、WKP₇～WKP₀）が発生したとき、ウォッチモードで割込み（タイマA、タイマF、タイマG、IRQ₀、WKP₇～WKP₀）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。

なお、アクティブ（中速）モードへの遷移時に1/2ステート早いタイミングで動作することがあります。

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）RES 端子による解除

RES端子を"Low"レベルにすると、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、SYSCR1のMA1、MA0で設定した周波数のクロックによって動作します。

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事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、ウォッチモードを経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(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] 直接遷移時間 = (2 + 1) × 2t_{osc} + 14 × 16t_{osc} = 230t_{osc}

(CPU 動作クロック: /8 を選択した場合)

<記号説明>

t_{osc}: 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

(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。

$$\begin{aligned} \text{直接遷移時間} = & \{ (\text{SLEEP 命令実行状態数}) + (\text{内部処理状態数}) \} \\ & \times (\text{遷移前の tcyc}) + (\text{割込み例外処理実行状態数}) \\ & \times (\text{遷移後の tcyc}) \dots\dots\dots (2) \end{aligned}$$

$$\text{〔例〕直接遷移時間} = (2 + 1) \times 16\text{tosc} + 14 \times 2\text{tosc} = 76\text{tosc}$$

(CPU 動作クロック: /8 を選択した場合)

<記号説明>

tosc : OSC クロックサイクル時間

tcyc : 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。

$$\begin{aligned} \text{直接遷移時間} = & \{ (\text{SLEEP 命令実行状態数}) + (\text{内部処理状態数}) \} \\ & \times (\text{遷移前の tsubcyc}) + \{ (\text{STS2} \sim \text{STS0} \text{ で設定した待機時間}) \\ & + (\text{割込み例外処理実行状態数}) \} \times (\text{遷移後の tcyc}) \dots\dots\dots (3) \end{aligned}$$

$$\text{〔例〕直接遷移時間} = (2 + 1) \times 8\text{tw} + (8192 + 14) \times 2\text{tosc} = 24\text{tw} + 16412\text{tosc}$$

(CPU 動作クロック: w/8、待機時間: 8192 ステートを選択した場合)

<記号説明>

tosc : OSC クロックサイクル時間

tw : ウォッチクロックサイクル時間

tcyc : システムクロック () サイクル時間

tsubcyc : サブクロック (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。

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

$$\text{〔例〕 直接遷移時間} = (2 + 1) \times 8t_w + (8192 + 14) \times 16t_{\text{osc}} = 24t_w + 131296t_{\text{osc}}$$

(CPU 動作クロック: $w/8$ 、 $/8$ 、待機時間: 8192 ステートを選択した場合)

<記号説明>

- t_{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_{cyc} : システムクロック () サイクル時間
- t_{subcyc} : サブクロック ($_{\text{SUB}}$) サイクル時間

5.9 モジュールスタンバイモード

5.9.1 モジュールスタンバイモードの設定

モジュールスタンバイモードは各周辺機能ごとに設定します。搭載されているすべての周辺モジュールはモジュールスタンバイモードに設定可能です。モジュールスタンバイモードに設定されると、モジュールへのシステムクロックの供給は停止され、機能は停止し、スタンバイモードと同じ状態になります。

モジュールスタンバイモードの設定はクロック停止レジスタ1 (CKSTPR1) とクロック停止レジスタ2 (CKSTPR2) の各ビットを"0"に設定することにより行います。(表 5.5 参照)

5.9.2 モジュールスタンバイモードの解除

モジュールスタンバイモードの解除はクロック停止レジスタ1 (CKSTPR1) とクロック停止レジスタ2 (CKSTPR2) の各ビットを"1"に設定することにより行います。(表 5.5 参照)

尚、リセット直後、クロック停止レジスタ1 (CKSTPR1) とクロック停止レジスタ2 (CKSTPR2) は各々"FF"、"FF"に初期化されています。

表 5.5

レジスタ名	ビット名		動作
CKSTPR1	TACKSTP	1	タイマ A のモジュールスタンバイモードは解除される
		0	タイマ A はモジュールスタンバイモードに設定される
	TCCKSTP	1	タイマ C のモジュールスタンバイモードは解除される
		0	タイマ C はモジュールスタンバイモードに設定される
	TFCKSTP	1	タイマ F のモジュールスタンバイモードは解除される
		0	タイマ F はモジュールスタンバイモードに設定される
	TGCKSTP	1	タイマ G のモジュールスタンバイモードは解除される
		0	タイマ G はモジュールスタンバイモードに設定される
	ADCKSTP	1	A/D 変換器のモジュールスタンバイモードは解除される
		0	A/D 変換器はモジュールスタンバイモードに設定される
	S1SKSTP	1	SCI1 のモジュールスタンバイモードは解除される
		0	SCI1 はモジュールスタンバイモードに設定される
S32CKSTP	1	SCI3-2 のモジュールスタンバイモードは解除される	
	0	SCI3-2 はモジュールスタンバイモードに設定される	

5. 低消費電力モード

レジスタ名	ビット名		動作
CKSTPR1	S31CKSTP	1	SCI3-1 のモジュールスタンバイモードは解除される
		0	SCI3-1 はモジュールスタンバイモードに設定される
CKSTPR2	LDCKSTP	1	LCD のモジュールスタンバイモードは解除される
		0	LCD はモジュールスタンバイモードに設定される
	PWCKSTP	1	PWM のモジュールスタンバイモードは解除される
		0	PWM はモジュールスタンバイモードに設定される
	WDCKSTP	1	ウォッチドッグタイマのモジュールスタンバイモードは解除される
		0	ウォッチドッグタイマはモジュールスタンバイモードに設定される
	AECKSTP	1	非同期イベントカウンタのモジュールスタンバイモードは解除される
		0	非同期イベントカウンタはモジュールスタンバイモードに設定される

【注】 各モジュールごとの動作の詳細は、各モジュールの章参照

6. ROM

第6章 目次

6.1	概要.....	133
	6.1.1	
	ブロック図.....	133
6.2	H8/3887、H8/3847 の PROM モード.....	134
	6.2.1	
	PROM モードの設定.....	134
	6.2.2	
	ソケットアダプタの端子対応とメモリマップ.....	134
6.3	H8/3887、H8/3847 のプログラミング.....	137
	6.3.1	
	書込み / ベリファイ.....	138
	6.3.2	
	書込み時の注意.....	140
6.4	書込み後の信頼性.....	142

6.1 概要

H8/3882、H8/3842 は 16k バイト、H8/3883、H8/3843 は 24k バイト、H8/3884、H8/3844 は 32k バイト、H8/3885、H8/3845 は 40k バイト、H8/3886、H8/3846 は 48k バイト、H8/3887、H8/3847 は 60k バイトのマスキング ROM 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。H8/3887、H8/3847 には ZTAT[®]版があり、60k バイトの PROM を備えています。

6.1.1 ブロック図

ROM のブロック図を図 6.1 に示します。

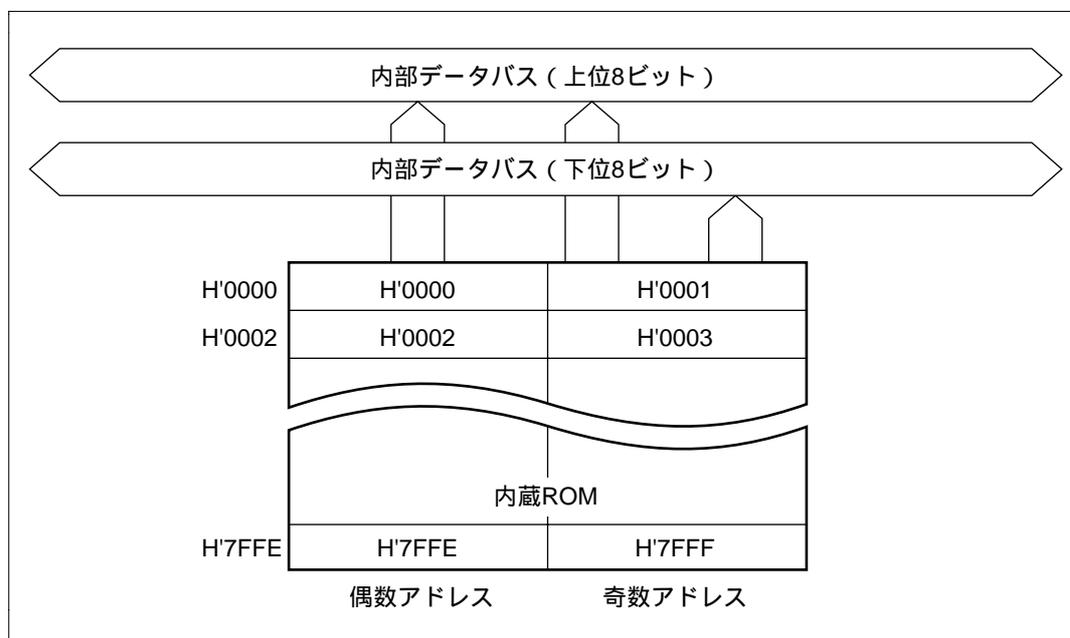


図 6.1 ROM のブロック図 (H8/3884、H8/3844 の場合)

6.2 H8/3887、H8/3847 の 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 ₄ /AN ₄ 端子	"Low" レベル
PB ₅ /AN ₅ 端子	
PB ₆ /AN ₆ 端子	"High" レベル

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、表 6.2 で示すように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。

表 6.2 ソケットアダプタ

パッケージ名	ソケットアダプタの名称 (メーカー)
100 ピン (FP-100B)	ME3887ESHS1H (MINATO) H7388BQ100D3201 (DATA-I/O)
100 ピン (FP-100A)	ME3887ESFS1H (MINATO) H7388AQ100D3201 (DATA-I/O)
100 ピン (TFP-100B)	ME3887ESNS1H (MINATO) H7388BT100D3201 (DATA-I/O)
100 ピン (TFP-100G)	開発中

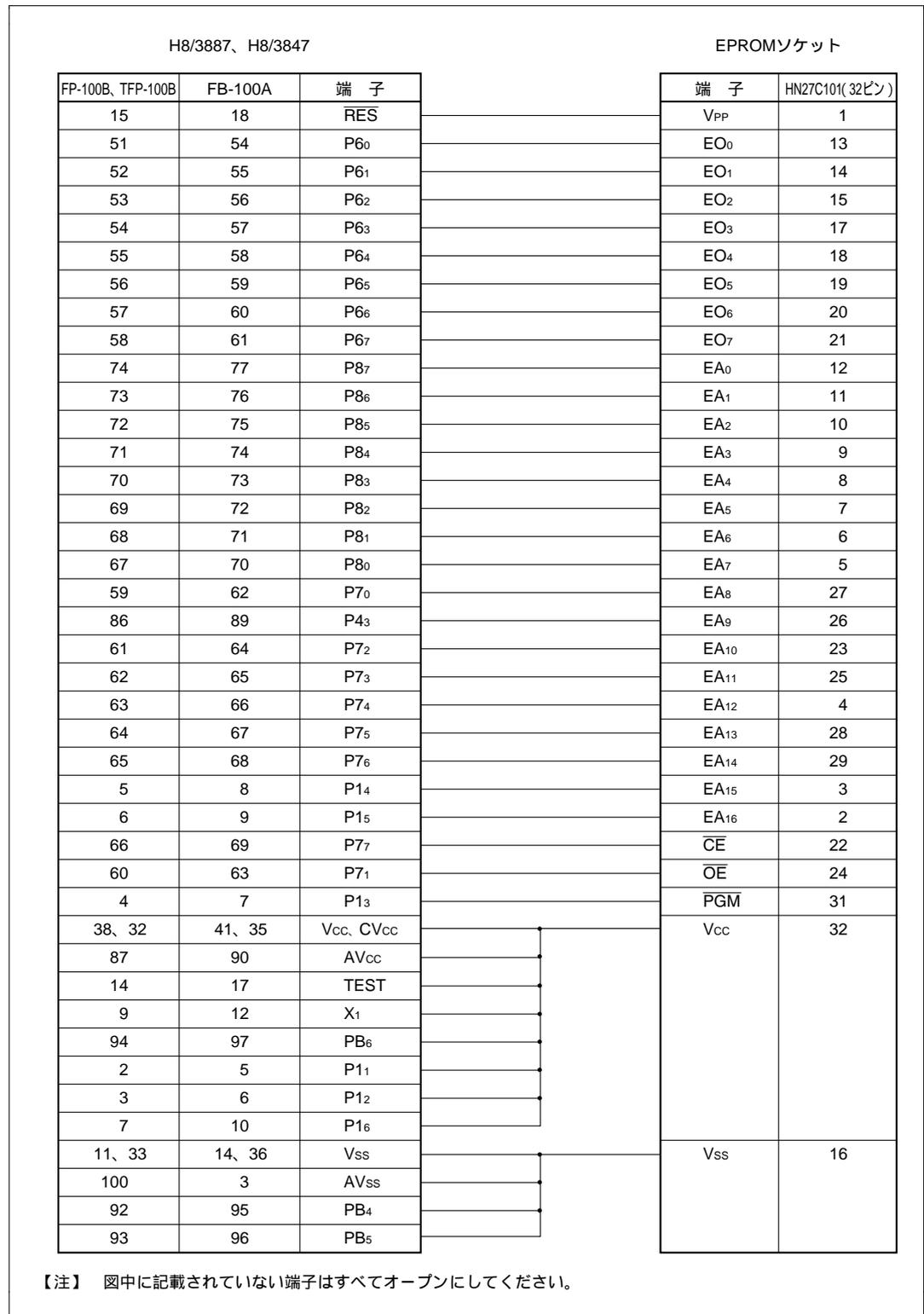


図 6.2 ソケットアダプタの端子対応図 (HN27C101)

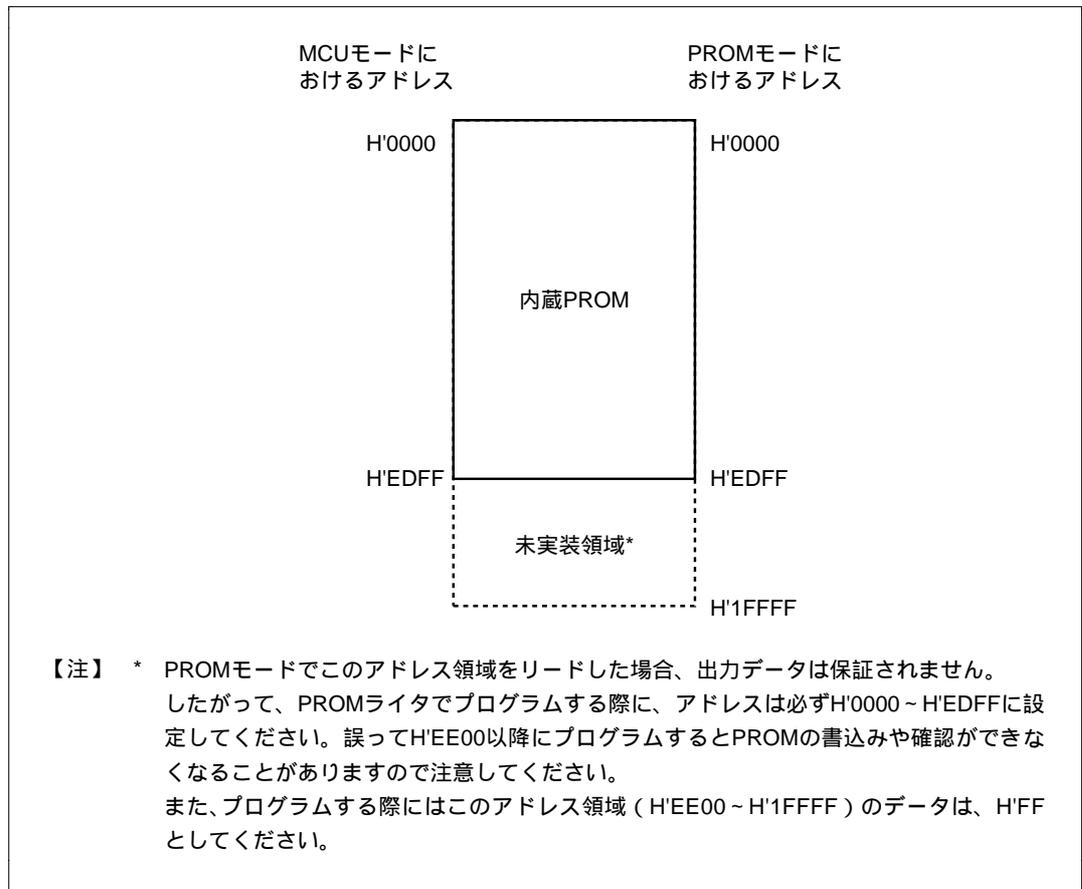


図 6.3 H8/3887、H8/3847 の PROM モード時のメモリマップ

6.3 H8/3887、H8/3847 のプログラミング

H8/3887、H8/3847 の PROM モード時の書込み、ベリファイなどのモード選択は、表 6.3 に示すような設定により行います。

表 6.3 PROM モード時の書込みモードの選択 (H8/3887、H8/3847)

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 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。

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。

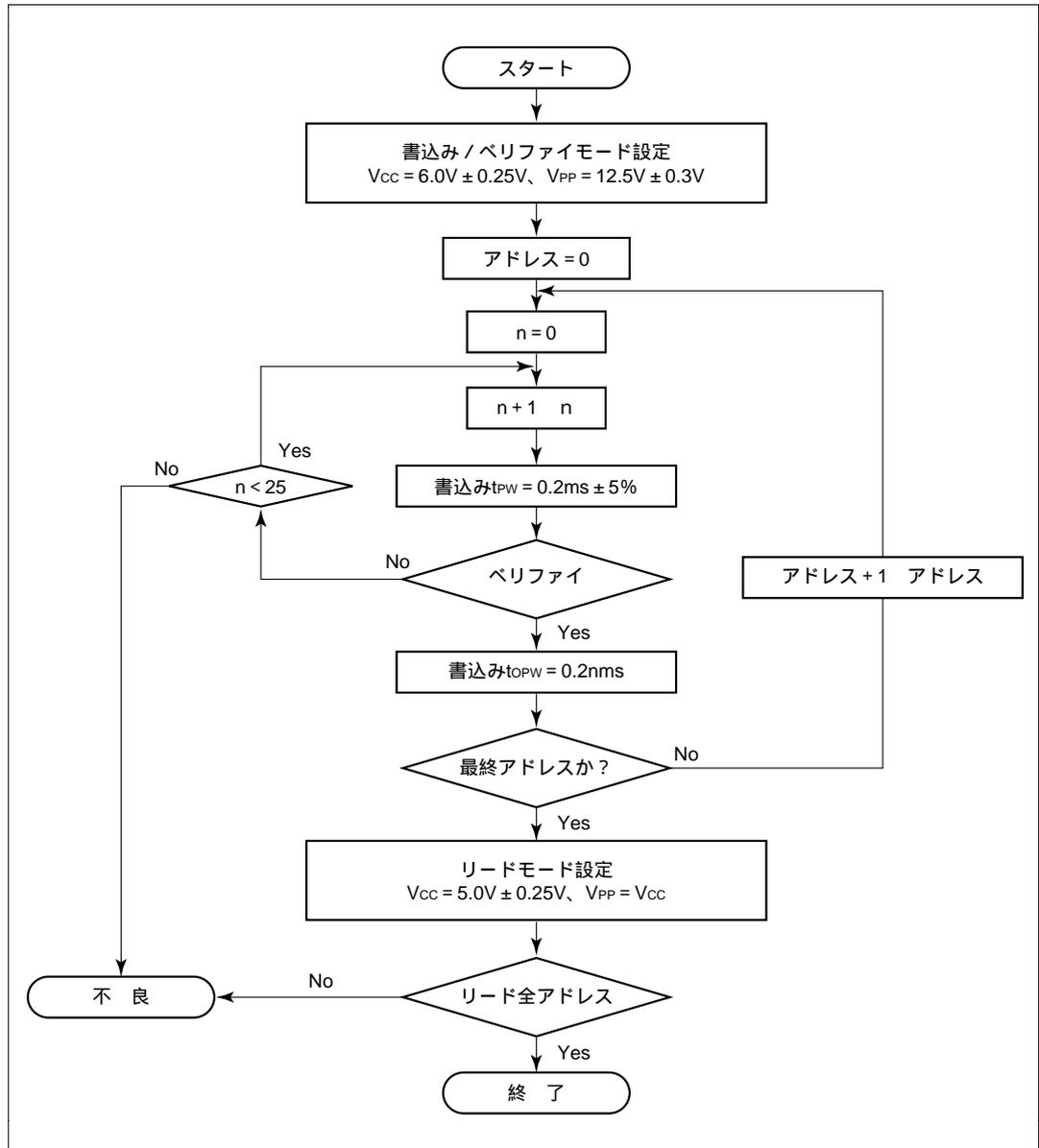


図 6.4 高速高信頼度プログラミングフローチャート

プログラミング時の電気的特性を表 6.4、表 6.5 に示します。

表 6.4 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	μA	$V_{in} = 5.25V/0.5V$
V_{CC} 電流		I_{CC}			40	mA	
V_{PP} 電流		I_{PP}			40	mA	

表 6.5 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			μs	図 6.5*1
\overline{OE} セットアップ時間	t_{OES}	2			μs	
データセットアップ時間	t_{DS}	2			μs	
アドレスホールド時間	t_{AH}	0			μs	
データホールド時間	t_{DH}	2			μs	
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			μs	
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 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			μs	
\overline{CE} セットアップ時間	t_{CES}	2			μs	
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 6.8 高性能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6.9に示します。

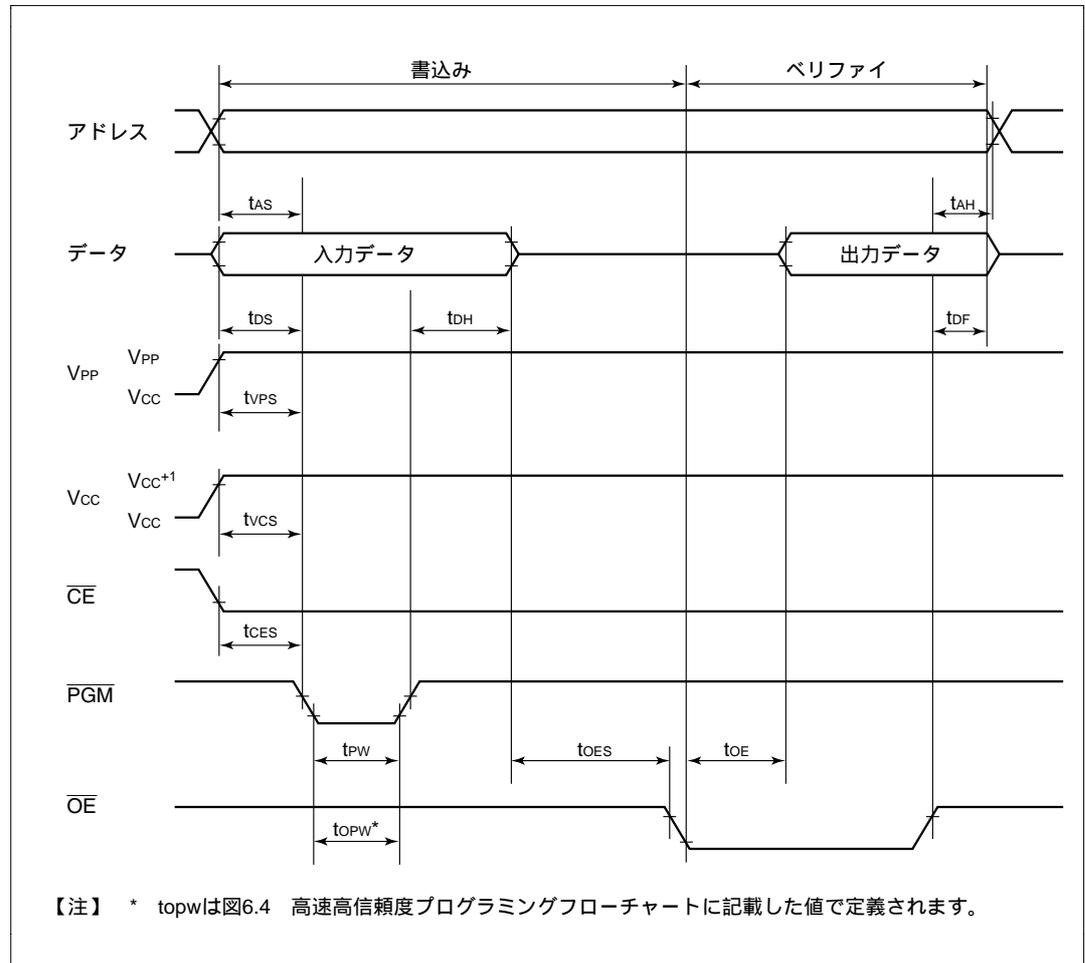


図 6.5 PROM 書込み/ベリファイタイミング

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 H'0000 ~ H'EDFF に設定してください。誤って H'EE00 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 H'EE00 ~ H'1FFFF のアドレス領域のデータは H'FF 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。

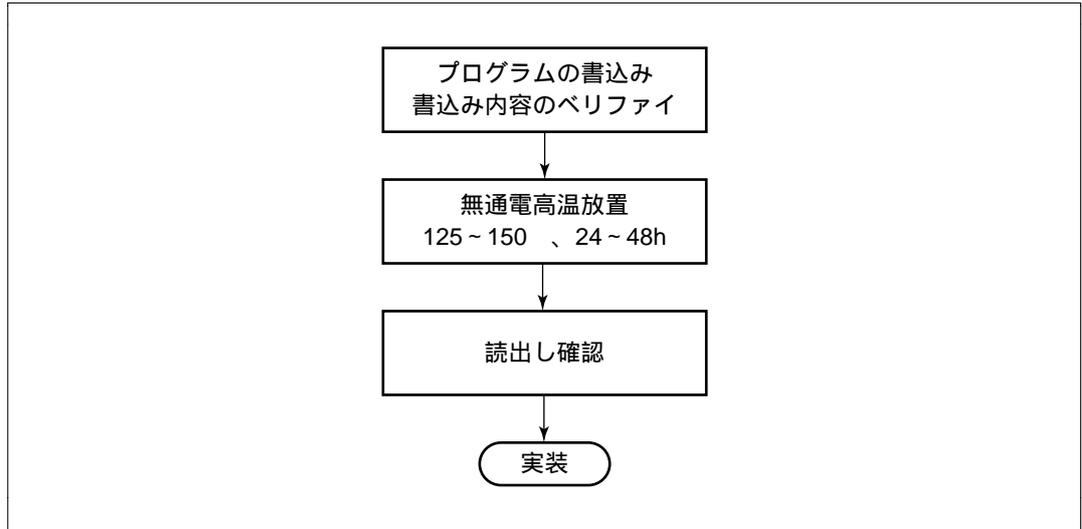


図 6.6 推奨スクリーニングフロー

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.....	145
	7.1.1	
	ブロック図.....	145

7.1 概要

H8/3882、H8/3883、H8/3842、H8/3843 は 1k バイト、H8/3884、H8/3885、H8/3886、H8/3887、H8/3844、H8/3845、H8/3846、H8/3847 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。

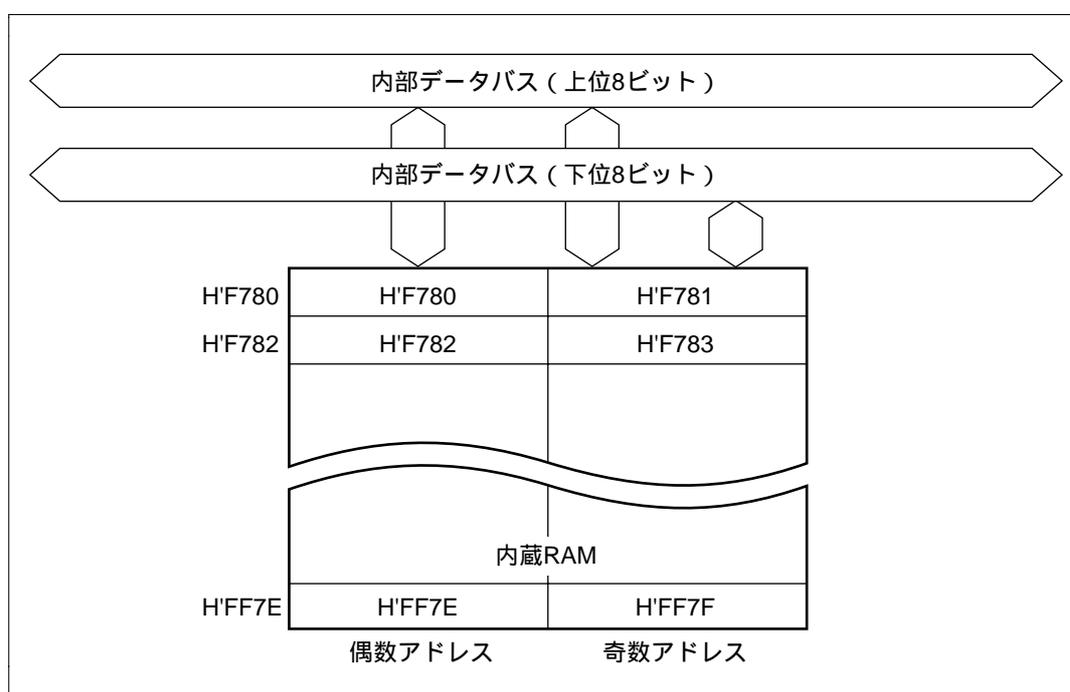


図 7.1 RAM のブロック図 (H8/3884、H8/3844 の場合)

8. I/O ポート

第8章 目次

8.1	概要.....	149
8.2	ポート1.....	151
	8.2.1 概要.....	151
	8.2.2 レジスタの構成と説明.....	151
	8.2.3 端子機能.....	156
	8.2.4 端子状態.....	158
	8.2.5 入力プルアップ MOS.....	158
8.3	ポート2.....	159
	8.3.1 概要.....	159
	8.3.2 レジスタの構成と説明.....	159
	8.3.3 端子機能.....	163
	8.3.4 端子状態.....	163
8.4	ポート3.....	164
	8.4.1 概要.....	164
	8.4.2 レジスタの構成と説明.....	164
	8.4.3 端子機能.....	168
	8.4.4 端子状態.....	169
	8.4.5 入力プルアップ MOS.....	169
8.5	ポート4.....	170
	8.5.1 概要.....	170
	8.5.2 レジスタの構成と説明.....	170
	8.5.3 端子機能.....	171
	8.5.4 端子状態.....	172
8.6	ポート5.....	173
	8.6.1 概要.....	173
	8.6.2 レジスタの構成と説明.....	173
	8.6.3 端子機能.....	175

	8.6.4	端子状態.....	176
	8.6.5	入力プルアップ MOS.....	176
8.7	ポート 6.....		177
	8.7.1	概要.....	177
	8.7.2	レジスタの構成と説明.....	177
	8.7.3	端子機能.....	179
	8.7.4	端子状態.....	179
	8.7.5	入力プルアップ MOS.....	179
8.8	ポート 7.....		180
	8.8.1	概要.....	180
	8.8.2	レジスタの構成と説明.....	180
	8.8.3	端子機能.....	181
	8.8.4	端子状態.....	181
8.9	ポート 8.....		182
	8.9.1	概要.....	182
	8.9.2	レジスタの構成と説明.....	182
	8.9.3	端子機能.....	183
	8.9.4	端子状態.....	183
8.10	ポート 9.....		184
	8.10.1	概要.....	184
	8.10.2	レジスタの構成と説明.....	184
	8.10.3	端子機能.....	185
	8.10.4	端子状態.....	186
8.11	ポート A.....		187
	8.11.1	概要.....	187
	8.11.2	レジスタの構成と説明.....	187
	8.11.3	端子機能.....	189
	8.11.4	端子状態.....	189
8.12	ポート B.....		190
	8.12.1	概要.....	190
	8.12.2	レジスタの構成と説明.....	190
8.13	ポート C.....		191
	8.13.1	概要.....	191
	8.13.2	レジスタの構成と説明.....	191
8.14	入出力データ反転機能.....		192
	8.14.1	概要.....	192
	8.14.2	レジスタの構成と説明.....	192
	8.14.3	シリアルポートコントロールレジスタを書換える際の注意事項.....	194

8.1 概要

本 LSI は、8 ビット入出力ポートを 8 本、4 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、8 ビット入力専用ポートを 1 本、4 ビット入力専用ポートを 1 本、および 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。

ポート 5、6、7、8、9、A は液晶表示用セグメント端子、コモン端子と兼用されており、8 ビット単位で選択できます。

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"> 8 ビット入出力ポート 入力プルアップ MOS 選択可 	P1 ₇ ~ P1 ₉ /	外部割込み 3~1	PMR1
		$\overline{\text{IRQ}}_3$ ~ $\overline{\text{IRQ}}_1$ /	タイマイベント入力端子 TMIF、TMIC	TCRF、TMC
		P1 ₄ / $\overline{\text{IRQ}}_4$ /	外部割り込み 4 および A/D 変換器の外部トリガ	PMR1、AMR
		P1 ₃ /TMIG	タイマ G インพุットキャプチャ入力	PMR1
		P1 ₂ 、P1 ₁ /	タイマ F アウトプットコンペア出力	PMR1
		P1 ₀ /TMOW	タイマ A クロック出力	PMR1
ポート 2	<ul style="list-style-type: none"> 8 ビット入出力ポート オープンドレイン出力選択可 大電流ポート 	P2 ₀ /SCK ₁	SCI1 のデータ出力 (SO ₁)、データ入力 (SI ₁)、クロック入出力 (SCK ₁)	PMR2
		P2 ₁ /SI ₁		
		P2 ₂ /SO ₁		
		P2 ₇ ~ P2 ₃	なし	
ポート 3	<ul style="list-style-type: none"> 8 ビット入出力ポート 入力プルアップ MOS 選択可 大電流ポート 	P3 ₇ /AEVL	SCI3-1 のデータ出力 (TXD ₃₁)、データ入力 (RXD ₃₁)、クロック入出力 (SCK ₃₁) および非同期カウンタのイベント入力端子 AEVL、AEVH	PMR3 SCR31 SMR31
		P3 ₆ /AEVH		
		P3 ₅ /TXD ₃₁		
		P3 ₄ /RXD ₃₁		
		P3 ₃ /SCK ₃₁		
		P3 ₂ /RESO	リセット出力、タイマ C カウントアップ/ダウン選択入力および 14 ビット	PMR3
		P3 ₁ /UD		
		P3 ₀ /PWM	PWM 出力	

8. I/O ポート

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 4	・ 1 ビット入力専用ポート	P4 ₃ /IRQ ₀	外部割込み 0	PMR3
	・ 3 ビット入出力ポート	P4 ₂ /TXD ₃₂ P4 ₁ /RXD ₃₂ P4 ₀ /SCK ₃₂	SCI3-2 のデータ出力 (TXD ₃₂)、データ入力 (RXD ₃₂)、クロック入出力 (SCK ₃₂)	SCR32 SMR32
ポート 5	・ 8 ビット入出力ポート ・ 入力プルアップ MOS 選択可	P5 ₇ ~ P5 ₀ / WKP ₇ ~ WKP ₀ / SEG ₈ ~ SEG ₁	ウェイクアップ入力 (WKP ₇ ~ WKP ₀)、セグメント出力 (SEG ₈ ~ SEG ₁)	PMR5 LPCR
ポート 6	・ 8 ビット入出力ポート ・ 入力プルアップ MOS 選択可	P6 ₇ ~ P6 ₀ / SEG ₁₆ ~ SEG ₉	セグメント出力 (SEG ₁₆ ~ SEG ₉)	LPCR
ポート 7	・ 8 ビット入出力ポート	P7 ₇ ~ P7 ₀ / SEG ₂₄ ~ SEG ₁₇	セグメント出力 (SEG ₂₄ ~ SEG ₁₇)	LPCR
ポート 8	・ 8 ビット入出力ポート	P8 ₇ ~ P8 ₀ / SEG ₃₂ ~ SEG ₂₅	セグメント出力 (SEG ₃₂ ~ SEG ₂₅)	LPCR
ポート 9	・ 8 ビット入出力ポート	P9 ₇ /SED ₄₀ /CL ₁ P9 ₆ /SEG ₃₉ /CL ₂ P9 ₅ /SEG ₃₈ /DO P9 ₄ /SEG ₃₇ /M P9 ₃ ~ P9 ₀ / SEG ₃₆ ~ SEG ₃₃	・ セグメント出力 (SEG ₄₀ ~ SEG ₃₇) ・ セグメント外部拡張用ラッチクロック (CL ₁)、シフトクロック (CL ₂)、表示データ (DO)、交流化信号 (M) ・ セグメント出力 (SEG ₃₆ ~ SEG ₃₃)	LPCR
ポート A	・ 4 ビット入出力ポート	PA ₃ ~ PA ₀ / COM ₄ ~ COM ₁	コモン出力 (COM ₄ ~ COM ₁)	LPCR
ポート B	・ 8 ビット入力ポート	PB ₇ ~ PB ₀ / AN ₇ ~ AN ₀	A/D 変換器のアナログ入力	AMR
ポート C	・ 4 ビット入力ポート	PC ₃ ~ PC ₀ / AN ₁₁ ~ AN ₈	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。

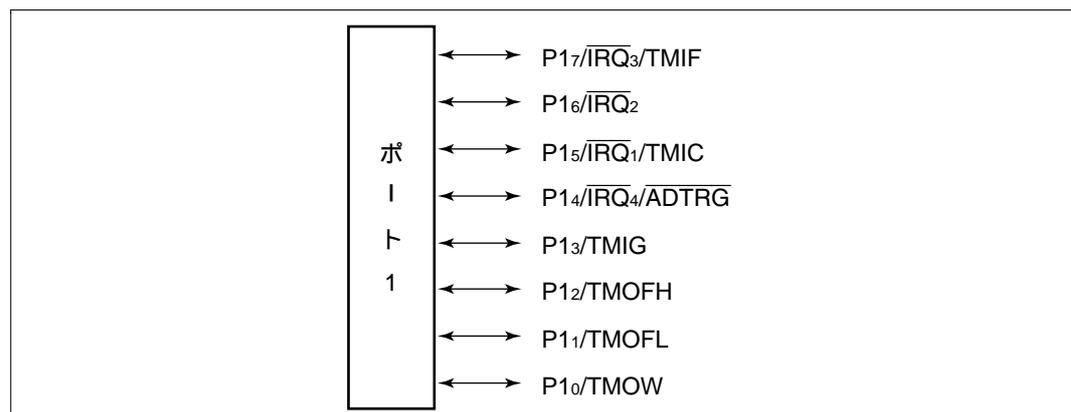


図 8.1 ポート 1 の端子構成

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'FFE0
ポートモードレジスタ 1	PMR1	R/W	H'00	H'FFC8

(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P1₇~P1₀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P1₇~P1₀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P1₇~P1₀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P1₇~P1₀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	IRQ4	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	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: P1₇/ $\overline{\text{IRQ}}_3$ /TMIF 端子機能切換え (IRQ3)

P1₇/ $\overline{\text{IRQ}}_3$ /TMIF 端子を P1₇ 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	
IRQ3	説明
0	P1 ₇ 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、「9.4.2 (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: P1₆/ $\overline{\text{IRQ}}_2$ 端子機能切換え (IRQ2)

P1₆/ $\overline{\text{IRQ}}_2$ を P1₆ 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	
IRQ2	説明
0	P1 ₆ 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：P1₅/ $\overline{\text{IRQ}}_1$ /TMIC 端子機能切換え（IRQ1）

P1₅/ $\overline{\text{IRQ}}_1$ /TMIC 端子を P1₅ 端子として使用するか、 $\overline{\text{IRQ}}_1$ /TMIC 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 ₅ 入出力端子として機能	(初期値)
1	$\overline{\text{IRQ}}_1$ /TMIC 入力端子として機能	

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

TMIC 端子の設定については、「9.3.2 (1) タイマモードレジスタ C (TMC)」を参照してください。

ビット4：P1₄/ $\overline{\text{IRQ}}_4$ / $\overline{\text{ADTRG}}$ 端子機能切換え（IRQ4）

P1₄/ $\overline{\text{IRQ}}_4$ / $\overline{\text{ADTRG}}$ 端子を P1₄ 端子として使用するか、 $\overline{\text{IRQ}}_4$ / $\overline{\text{ADTRG}}$ 端子として使用するかを設定します。

ビット4		
IRQ4	説明	
0	P1 ₄ 入出力端子として機能	(初期値)
1	$\overline{\text{IRQ}}_4$ / $\overline{\text{ADTRG}}$ 入力端子として機能	

【注】 $\overline{\text{ADTRG}}$ 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。

ビット3：P1₃/TMIG 端子機能切換え（TMIG）

P1₃/TMIG 端子を P1₃ 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 ₃ 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：P1₂/TMOFH 端子機能切換え（TMOFH）

P1₂/TMOFH 端子を P1₂ 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 ₂ 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：P1₇/TMOFL 端子機能切換え（TMOFL）

P1₇/TMOFL 端子を P1₇ 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 ₇ 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：P1₀/TMOW 端子機能切換え（TMOW）

P1₀/TMOW 端子を P1₀ 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 ₀ 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

端子	選択方法と端子機能			
P1 ₇ / $\overline{\text{IRQ}}_3$ / TMIF	PMR1 の IRQ3、TCRF の CKSL2 ~ CKSL0、PCR1 の PCR1 ₇ の組合せで次のように切り換わります。			
	IRQ3	0		1
	PCR1 ₇	0	1	*
	CKSL2 ~ CKSL0	*		0**以外 0**
端子機能	P1 ₇ 入力端子	P1 ₇ 出力端子	$\overline{\text{IRQ}}_3$ 入力端子	$\overline{\text{IRQ}}_3$ /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、IRQ3 割込みを禁止してください。			
P1 ₆ / $\overline{\text{IRQ}}_2$	PMR1 の IRQ2、PCR1 の PCR1 ₆ の組合せで次のように切り換わります。			
	IRQ2	0		1
	PCR1 ₆	0	1	*
端子機能	P1 ₆ 入力端子	P1 ₆ 出力端子	$\overline{\text{IRQ}}_2$ 入力端子	
P1 ₅ / $\overline{\text{IRQ}}_1$ / TMIC	PMR1 の IRQ1、TMC の TMC2 ~ TMC0、PCR1 の PCR1 ₅ の組合せで、次のように切り換わります。			
	IRQ1	0		1
	PCR1 ₅	0	1	*
	TMC2 ~ TMC0	*		111以外 111
端子機能	P1 ₅ 入力端子	P1 ₅ 出力端子	$\overline{\text{IRQ}}_1$ 入力端子	$\overline{\text{IRQ}}_1$ /TMIC 入力端子
	【注】 TMIC入力端子として使用する場合には、IENR1のIEN1を"0"にクリアして、IRQ1 割込みを禁止してください。			
P1 ₄ / $\overline{\text{IRQ}}_4$ / ADTRG	PMR1 の IRQ4 と AMR の TRGE、PCR1 の PCR1 ₄ の組合せで、次のように切り換わります。			
	IRQ4	0		1
	PCR1 ₄	0	1	*
	TRGE	*		0 1
端子機能	P1 ₄ 入力端子	P1 ₄ 出力端子	$\overline{\text{IRQ}}_4$ 入力端子	$\overline{\text{IRQ}}_4$ /ADTRG 入力端子
	【注】 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

端子	選択方法と端子機能			
P1 ₃ /TMIG	PMR1のTMIGとPCR1のPCR1 ₃ の組合せで、次のように切換わります。			
	TMIG	0		1
	PCR1 ₃	0	1	*
	端子機能	P1 ₃ 入力端子	P1 ₃ 出力端子	TMIG入力端子
P1 ₂ /TMOFH	PMR1のTMOFHとPCR1のPCR1 ₂ の組合せで、次のように切換わります。			
	TMOFH	0		1
	PCR1 ₂	0	1	*
	端子機能	P1 ₂ 入力端子	P1 ₂ 出力端子	TMOFH出力端子
P1 ₁ /TMOFL	PMR1のTMOFLとPCR1のPCR1 ₁ の組合せで、次のように切換わります。			
	TMOFL	0		1
	PCR1 ₁	0	1	*
	端子機能	P1 ₁ 入力端子	P1 ₁ 出力端子	TMOFL出力端子
P1 ₀ /TMOW	PMR1のTMOWとPCR1のPCR1 ₀ の組合せで、次のように切換わります。			
	TMOW	0		1
	PCR1 ₀	0	1	*
	端子機能	P1 ₀ 入力端子	P1 ₀ 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 ₇ /IRQ ₃ /TMIF	ハイ	保持	保持	ハイ	保持	動作	動作
P1 ₆ /IRQ ₂	インピー			インピー			
P1 ₅ /IRQ ₁ /TMIC	ダンス			ダンス*			
P1 ₄ /IRQ ₄ /ADTRG							
P1 ₃ /TMIG							
P1 ₂ /TMOFH							
P1 ₁ /TMOFL							
P1 ₀ /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 _n	0		1
PUCR1 _n	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、8 ビットの入出力ポートです。ポート 2 の各端子は、図 8.2 に示す構成になっています。

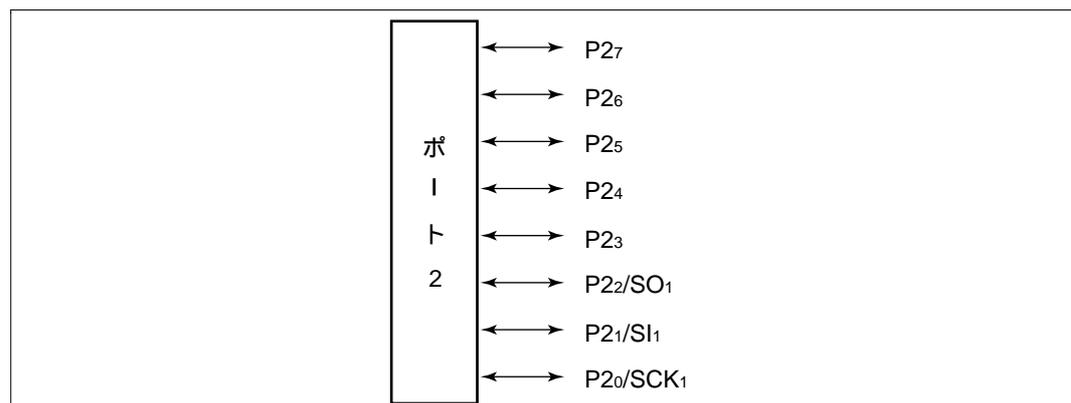


図 8.2 ポート 2 の端子構成

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'D8	H'FFC9
ポートモードレジスタ 4	PMR4	R/W	H'00	H'FFCB

(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P2₇~P2₀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P2₇~P2₀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P2₇~P2₀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。リードした場合各ビットは常に"1"が読み出されます。

(3) ポートモードレジスタ 2 (PMR2)

ビット:	7	6	5	4	3	2	1	0
	—	—	POF1	—	—	SO1	SI1	SCK1
初期値:	1	1	0	1	1	0	0	0
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、P2₀、P2₁、P2₃端子機能の切換え、P2₂/SO₁端子のPMOSのON/OFFを制御します。

リセット時、PMR2はH'D8にイニシャライズされます。

ビット7、6、4、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：P2₂/SO₁端子 PMOS コントロール (POF1)

P2₂/SO₁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2：P2₂/SO₁端子機能切換え (SO1)

P2₂/SO₁端子をP2₂端子として使用するか、SO₁端子として使用するかを設定します。

ビット2		
SO1	説明	
0	P2 ₂ 入出力端子として機能	(初期値)
1	SO ₁ 出力端子として機能	

ビット1：P2₁/SI₁端子機能切換え (SI1)

P2₁/SI₁端子をP2₁端子として使用するか、SI₁端子として使用するかを設定します。

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

ビット0：P2₀/SCK₁端子機能切換え (SCK1)

P2₀/SCK₁端子をP2₀端子として使用するか、SCK₁端子として使用するかを設定します。

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

(4) ポートモードレジスタ 4 (PMR4)

ビット:	7	6	5	4	3	2	1	0
	NMOD7	NMOD6	NMOD5	NMOD4	NMOD3	NMOD2	NMOD1	NMOD0
初期値:	0	0	0	0	0	0	0	0
R/W :	R/W							

PMR4 は、8 ビットのリード/ライト可能なレジスタで、PCR2 が"1"のとき、ポート 2 の各端子を CMOS 出力とするか、NMOS オープンドレイン出力とするかをビットごとに制御します。

リセット時、PMR4 は H'00 にイニシャライズされます。

ビット n : NMOS オープンドレイン出力セレクト (NMODn)

P2n 端子を出力端子として使用する場合に、NMOS オープンドレイン出力とします。

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

(n=7~0)

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

端子	選択方法と端子機能			
P2 ₇ ~ P2 ₃	PCR2 の各ビットにより、次のように切換わります。			
	(n = 7 ~ 3)			
	PCR2 _n	0	1	
端子機能	P2 _n 入力端子	P2 _n 出力端子		
P2 ₂ /SO ₁	PMR2 の SO1 と PCR2 の PCR2 ₂ の組合せで、次のように切換わります。			
	SO1	0	1	
	PCR2 ₂	0	1	*
	端子機能	P2 ₂ 入力端子	P2 ₂ 出力端子	SO ₁ 出力端子
P2 ₁ /SI ₁	PMR2 の SI1 と PCR2 の PCR2 ₁ の組合せで、次のように切換わります。			
	SI1	0	1	
	PCR2 ₁	0	1	*
	端子機能	P2 ₁ 入力端子	P2 ₁ 出力端子	SI ₁ 入力端子
P2 ₀ /SCK ₁	PMR2 の SCK1、PCR2 の PCR2 ₀ の組合せで、次のように切換わります。			
	SCK1	0	1	
	PCR2 ₀	0	1	*
	端子機能	P2 ₀ 入力端子	P2 ₀ 出力端子	SCK ₁ 入出力端子

* : 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 ₇ ~ P2 ₃	ハイ	保持	保持	ハイ	保持	動作	動作
P2 ₂ /SO ₁	インピー			インピー			
P2 ₁ /SI ₁	ダンス			ダンス			
P2 ₀ /SCK ₁							

8.4 ポート 3

8.4.1 概要

ポート 3 は、8 ビットの入出力ポートです。ポート 3 の各端子は、図 8.3 に示す構成になっています。

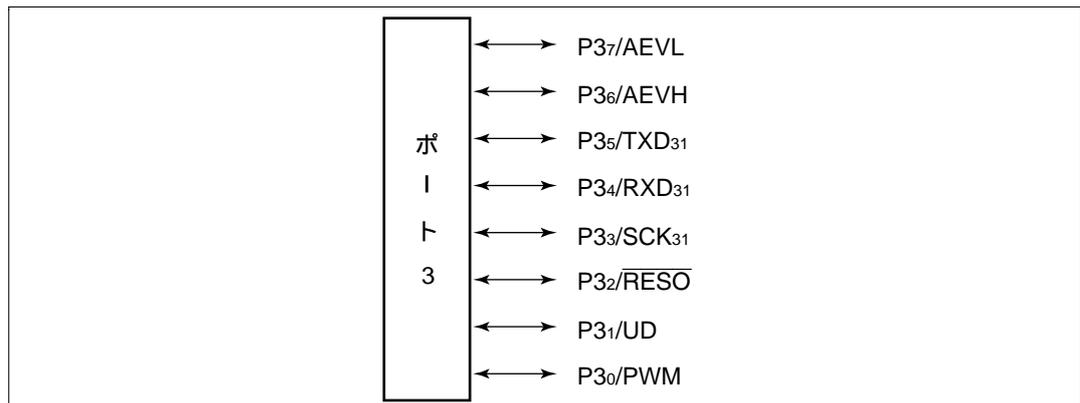


図 8.3 ポート 3 の端子構成

8.4.2 レジスタの構成と説明

表 8.8 にポート 3 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 3	PDR3	R/W	H'00	H'FFD6
ポートコントロールレジスタ 3	PCR3	W	H'00	H'FFE6
ポートプルアップコントロールレジスタ 3	PUCR3	R/W	H'00	H'FFE1
ポートモードレジスタ 3	PMR3	R/W	H'04	H'FFCA

(1) ポートデータレジスタ 3 (PDR3)

ビット:	7	6	5	4	3	2	1	0
	P37	P36	P35	P34	P33	P32	P31	P30
初期値:	0	0	0	0	0	0	0	0
R/W :	R/W							

PDR3は、ポート3の各端子P3₇~P3₀のデータを格納する8ビットのレジスタです。

PCR3が"1"のとき、ポート3のリードを行うと、PDR3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。PCR3が"0"のとき、ポート3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、PDR3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 3 (PCR3)

ビット:	7	6	5	4	3	2	1	0
	PCR37	PCR36	PCR35	PCR34	PCR33	PCR32	PCR31	PCR30
初期値:	0	0	0	0	0	0	0	0
R/W :	W	W	W	W	W	W	W	W

PCR3は、ポート3の各端子P3₇~P3₀の入出力をビットごとに制御します。PCR3に"1"をセットすると対応するP3₇~P3₀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。PMR3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、PCR3およびPDR3の設定が有効となります。

リセット時、PCR3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 3 (PUCR3)

ビット:	7	6	5	4	3	2	1	0
	PUCR37	PUCR36	PUCR35	PUCR34	PUCR33	PUCR32	PUCR31	PUCR30
初期値:	0	0	0	0	0	0	0	0
R/W :	R/W							

PUCR3は、ポート3の各端子P3₇~P3₀のプルアップMOSをビットごとに制御します。

PCR3が"0"の状態でもPUCR3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、PUCR3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3 (PMR3)

ビット:	7	6	5	4	3	2	1	0
	AEVL	AEVH	WDCKS	NCS	IRQ0	RESO	UD	PWM
初期値:	0	0	0	0	0	1	0	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PMR3 は、8ビットのリード/ライト可能なレジスタで、主にポート3の各端子機能の切換えを制御します。

リセット時、PMR3はH'04にイニシャライズされます。

ビット7: P3₇/AEVL 端子機能切換え (AEVL)

P3₇/AEVL 端子を P3₇ 端子として使用するか、AEVL 端子として使用するかを設定します。

ビット7		
AEVL	説明	
0	P3 ₇ 入出力端子として機能	(初期値)
1	AEVL 入力端子として機能	

ビット6: P3₆/AEVH 端子機能切換え (AEVH)

P3₆/AEVH 端子を P3₆ 端子として使用するか、AEVH 端子として使用するかを設定します。

ビット6		
AEVH	説明	
0	P3 ₆ 入出力端子として機能	(初期値)
1	AEVH 入力端子として機能	

ビット5: ウォッチドッグタイマソースクロック選択 (WDCKS)

ウォッチドッグタイマのソースクロックの選択を行います。

ビット5		
WDCKS	説明	
0	/8192 を選択	(初期値)
1	$v_w/32$ を選択	

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：P4₃/ $\overline{\text{IRQ}}_0$ 端子機能切換え（IRQ₀）

P4₃/ $\overline{\text{IRQ}}_0$ 端子を P4₃ 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3		
$\overline{\text{IRQ}}_0$	説明	
0	P4 ₃ 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2：P3₂/ $\overline{\text{RESO}}$ 端子機能切換え（RESO）

P3₂/ $\overline{\text{RESO}}$ 端子を P3₂ 端子として使用するか、 $\overline{\text{RESO}}$ 端子として使用するかを設定します。

ビット2		
$\overline{\text{RESO}}$	説明	
0	P3 ₂ 入出力端子として機能	
1	$\overline{\text{RESO}}$ 出力端子として機能	（初期値）

ビット1：P3₁/UD 端子機能切換え（UD）

P3₁/UD 端子を P3₁ 端子として使用するか、UD 端子として使用するかを設定します。

ビット1		
UD	説明	
0	P3 ₁ 入出力端子として機能	（初期値）
1	UD 入力端子として機能	

ビット0：P3₀/PWM 端子機能切換え（PWM）

P3₀/PWM 端子を P3₀ 端子として使用するか、PWM 端子として使用するかを設定します。

ビット0		
PWM	説明	
0	P3 ₀ 入出力端子として機能	（初期値）
1	PWM 出力端子として機能	

8.4.3 端子機能

表 8.9 にポート 3 の端子機能を示します。

表 8.9 ポート 3 の端子機能

端子	選択方法と端子機能				
P3 ₇ /AEVL	PMR3 の AEVL と PCR3 の PCR3 ₇ の組合せで、次のように切り替わります。				
	AEVL	0		1	
	PCR3 ₇	0	1	*	
	端子機能	P3 ₇ 入力端子	P3 ₇ 出力端子	AEVL入力端子	
P3 ₆ /SEVH	PMR3 の AEVH と PCR3 の PCR3 ₆ の組合せで、次のように切り替わります。				
	AEVH	0		1	
	PCR3 ₆	0	1	*	
	端子機能	P3 ₆ 入力端子	P3 ₆ 出力端子	AEVH入力端子	
P3 ₅ /TXD ₃₁	SCR3-1 の TE、SPCR の SPC31、PCR3 の PCR3 ₅ の組合せで、次のように切り替わります。				
	SPC31	0		1	
	TE	0		1	
	PCR3 ₅	0	1	*	
端子機能	P3 ₅ 入力端子	P3 ₅ 出力端子	TXD ₃₁ 出力端子		
P3 ₄ /RXD ₃₁	SCR3-1 の RE と PCR3 の PCR3 ₄ の組合せで、次のように切り替わります。				
	RE	0		1	
	PCR3 ₄	0	1	*	
	端子機能	P3 ₄ 入力端子	P3 ₄ 出力端子	RXD ₃₁ 入力端子	
P3 ₃ /SCK ₃₁	SCR3-1 の CKE1、CKE0、SMR31 および PCR3 の PCR3 ₃ の組合せで、次のように切り替わります。				
	CKE1	0		1	
	CKE0	0		1	*
	COM3 ₁	0	1	*	*
	PCR3 ₃	0	1	*	*
端子機能	P3 ₃ 入力端子	P3 ₃ 出力端子	SCK ₃₁ 出力端子	SCK ₃₁ 入力端子	
P3 ₂ /RESO	PMR3 の RESO と PCR3 の PCR3 ₂ の組合せで、次のように切り替わります。				
	RESO	0		1	
	PCR3 ₂	0	1	*	
	端子機能	P3 ₂ 入力端子	P3 ₂ 出力端子	RESO出力端子	

* : Don't care

端子	選択方法と端子機能		
P3 ₇ /UD	PMR3のUDとPCR3のPCR3 ₁ の組合せで、次のように切り替わります。		
	UD	0	1
	PCR3 ₁	0	1
	端子機能	P3 ₁ 入力端子	P3 ₁ 出力端子
P3 ₀ /PWM	PMR3のPWMおよびPCR3のPCR3 ₀ の組合せで、次のように切り替わります。		
	PWM	0	1
	PCR3 ₀	0	1
	端子機能	P3 ₀ 入力端子	P3 ₀ 出力端子

* : Don't care

8.4.4 端子状態

各動作モードにおけるポート3の端子状態を表8.10に示します。

表8.10 ポート3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
P3 ₇ /AEVL	ハイ	保持	保持	ハイ	保持	動作	動作
P3 ₆ /AEVH	インピー			インピー			
P3 ₅ /TXD ₃₁	ダンス			ダンス*			
P3 ₄ /RXD ₃₁							
P3 ₃ /SCK ₃₁							
P3 ₂ /RES ₀	リセット出力						
P3 ₁ /UD	ハイ						
P3 ₀ /PWM	インピー						
	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。

8.4.5 入力プルアップMOS

ポート3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。PCR3が"0"にクリアされている状態でPUCR3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。

PCR3 _n	0		1
PUCR3 _n	0	1	*
入力プルアップMOS	OFF	ON	OFF

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

8.5 ポート4

8.5.1 概要

ポート4は、3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4の各端子は、図8.4に示す構成になっています。

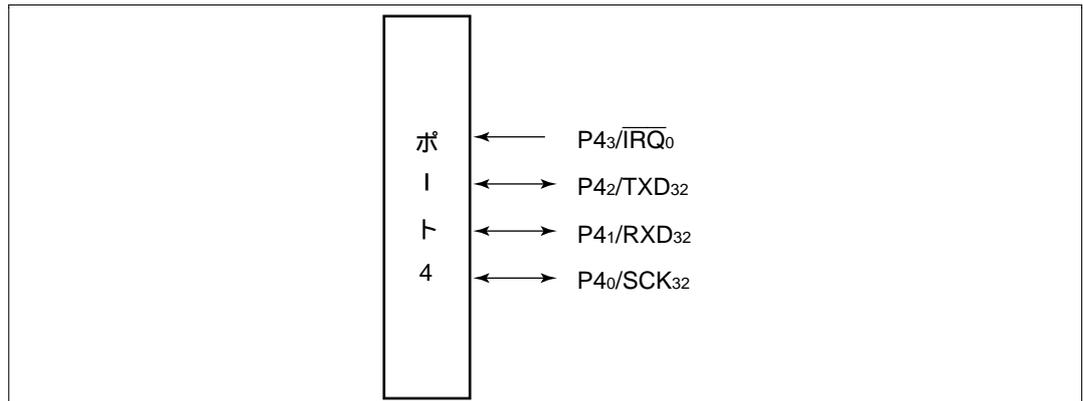


図8.4 ポート4の端子構成

8.5.2 レジスタの構成と説明

表8.11にポート4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4	PDR4	R/W	H'F8	H'FFD7
ポートコントロールレジスタ4	PCR4	W	H'F8	H'FFE7

(1) ポートデータレジスタ4 (PDR4)

ビット:	7	6	5	4	3	2	1	0
	—	—	—	—	P43	P42	P41	P40
初期値:	1	1	1	1	1	0	0	0
R/W :	—	—	—	—	R	R/W	R/W	R/W

PDR4は、ポート4の各端子P4₂~P4₀のデータを格納する8ビットのレジスタです。

PCR4が"1"のとき、ポート4のリードを行うと、PDR4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。PCR4が"0"のとき、ポート4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、PDR4はH'F8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 4 (PCR4)

ビット:	7	6	5	4	3	2	1	0
	—	—	—	—	—	PCR4 ₂	PCR4 ₁	PCR4 ₀
初期値:	1	1	1	1	1	0	0	0
R/W :	—	—	—	—	—	W	W	W

PCR4は、ポート4の各端子P4₂~P4₀の入出力をビットごとに制御します。PCR4に"1"をセットすると対応するP4₂~P4₀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。SCR3-2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、PCR4およびPDR4の設定が有効となります。

リセット時、PCR4はH'F8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。リードした場合各ビットは常に"1"が読み出されます。

8.5.3 端子機能

表8.12にポート4の端子機能を示します。

表 8.12 ポート4の端子機能

端子	選択方法と端子機能			
P4 ₃ / $\overline{\text{IRQ}}_0$	PMR3のIRQ0により次のように切り替わります。			
	IRQ0	0	1	
端子機能	P4 ₃ 入力端子		$\overline{\text{IRQ}}_0$ 入力端子	
P4 ₂ /TXD ₃₂	SCR3-2のTE、SPCRのSPC32、PCR4のPCR4 ₂ の組合せで、次のように切り替わります。			
	SPC32	0	1	
	TE	0	1	
	PCR4 ₂	0	1	*
端子機能	P4 ₂ 入力端子	P4 ₂ 出力端子	TXD ₃₂ 出力端子	
P4 ₁ /RXD ₃₂	SCR3-2のREとPCR4のPCR4 ₁ の組合せで、次のように切り替わります。			
	RE	0	1	
	PCR4 ₁	0	1	*
端子機能	P4 ₁ 入力端子	P4 ₁ 出力端子	RXD ₃₂ 入力端子	

* : Don't care

端子	選択方法と端子機能				
P4 ₀ /SCK ₃₂	SCR3-2 の CKE1、CKE0、SMR32 の COM32、および PCR4 の PCR4 ₀ の組合せで、次のように切換わります。				
	CKE1	0			1
	CKE0	0		1	*
	COM32	0		1	*
	PCR4 ₀	0	1	*	
	端子機能	P4 ₀ 入力端子	P4 ₀ 出力端子	SCK ₃₂ 出力端子	SCK ₃₂ 入力端子

* : Don't care

8.5.4 端子状態

各動作モードにおけるポート4の端子状態を表8.13に示します。

表 8.13 ポート4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
P4 ₃ /IRQ ₀	ハイ	保持	保持	ハイ	保持	動作	動作
P4 ₂ /TXD ₃₂	インピー			インピー			
P4 ₁ /RXD ₃₂	ダンス			ダンス			
P4 ₀ /SCK ₃₂							

8.6 ポート5

8.6.1 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8.5に示す構成になっています。

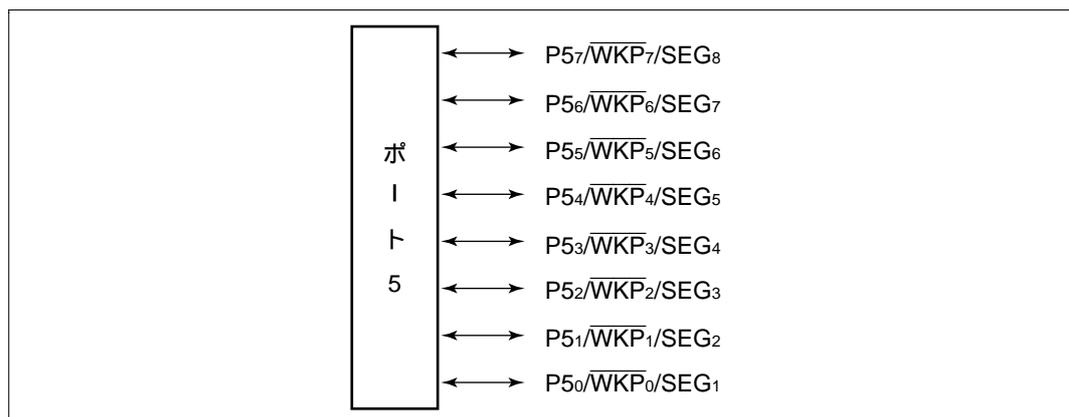


図8.5 ポート5の端子構成

8.6.2 レジスタの構成と説明

表8.14にポート5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'FFE2
ポートモードレジスタ5	PMR5	R/W	H'00	H'FFCC

(1) ポートデータレジスタ 5 (PDR5)

ビット:	7	6	5	4	3	2	1	0
	P5 ₇	P5 ₆	P5 ₅	P5 ₄	P5 ₃	P5 ₂	P5 ₁	P5 ₀
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 P5₇ ~ P5₀ 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
	PCR5 ₇	PCR5 ₆	PCR5 ₅	PCR5 ₄	PCR5 ₃	PCR5 ₂	PCR5 ₁	PCR5 ₀
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 P5₇ ~ P5₀ 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 P5₇ ~ P5₀ 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、LPCR の SGS3 ~ SGS0 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。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
	PUCR5 ₇	PUCR5 ₆	PUCR5 ₅	PUCR5 ₄	PUCR5 ₃	PUCR5 ₂	PUCR5 ₁	PUCR5 ₀
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 P5₇ ~ P5₀ 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 ₇	WKP ₆	WKP ₅	WKP ₄	WKP ₃	WKP ₂	WKP ₁	WKP ₀
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_n/WKP_n/SEG_{n+1} 端子機能切換え (WKPn)

P5_n/WKP_n/SEG_{n+1} 端子を SEG_{n+1} 端子として使用しない場合に、P5_n 端子として使用するか、WKP_n 端子として使用するかを設定します。

ビット n	説明
WKPn	
0	P5 _n 入出力端子として機能 (初期値)
1	WKP _n 入力端子として機能

(n = 7 ~ 0)

【注】 SEG_{n+1} としての使用については、「13.2.1 LCD ポートコントロールレジスタ (LPCR)」を参照してください。

8.6.3 端子機能

表 8.15 にポート 5 の端子機能を示します。

表 8.15 ポート 5 の端子機能

端子	選択方法と端子機能			
P5 ₇ /WKP ₇ /SEG ₈ ~ P5 ₀ /WKP ₀ /SEG ₁	PMR5 の WKP _n 、PCR5 の PCR5 _n 、および LPCR の SGS3 ~ SGS0 の組合せで、次のように切換わります。 (n = 7 ~ 0)			
	SGS3 ~ SGS0	0 * * *		1 * * *
	WKP _n	0	1	*
	PCR5 _n	0	1	*
	端子機能	P5 _n 入力端子	P5 _n 出力端子	WKP _n 入力端子 SEG _{n+1} 出力端子

* : Don't care

8.6.4 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8.16に示します。

表 8.16 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 ₇ / $\overline{\text{WKP}}_7$ /SEG ₈ ~ P5 ₀ / $\overline{\text{WKP}}_0$ /SEG ₁	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。

8.6.5 入力プルアップ MOS

ポート5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。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。

PCR5 _n	0		1
PUCR5 _n	0	1	*
入力プルアップ MOS	OFF	ON	OFF

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

8.7 ポート 6

8.7.1 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 8.6 に示す構成になっています。

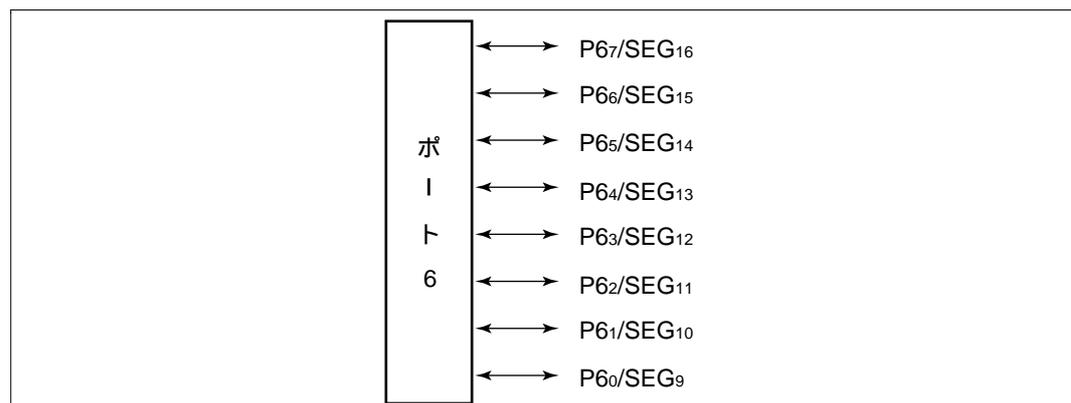


図 8.6 ポート 6 の端子構成

8.7.2 レジスタの構成と説明

表 8.17 にポート 6 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 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
ポートプルアップコントロールレジスタ 6	PUCR6	R/W	H'00	H'FFE3

(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P6₇~P6₀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P6₇~P6₀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P6₇~P6₀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。LPCRのSGS3~SGS0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、PCR6およびPDR6の設定が有効となります。

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。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P6₇~P6₀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。

8.7.3 端子機能

表 8.18 にポート 6 の端子機能を示します。

表 8.18 ポート 6 の端子機能

端子	選択方法と端子機能		
P6 ₇ /SEG ₁₆ ~ P6 ₀ /SEG ₉	PCR6 の PCR6 _n と LPCR の SGS3 ~ SGS0 の組合せで、次のように切替わります。 (n = 7 ~ 0)		
	SGS3 ~ SGS0	00**、010*	011*、1***
	PCR6 _n	0	1
	端子機能	P6 _n 入力端子	P6 _n 出力端子
			SEG _{n+9} 出力端子

* : Don't care

8.7.4 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 8.19 に示します。

表 8.19 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 ₇ /SEG ₁₆ ~ P6 ₀ /SEG ₉	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。

8.7.5 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 _n	0		1
PUCR6 _n	0	1	*
入力プルアップ MOS	OFF	ON	OFF

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

8.8 ポート7

8.8.1 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8.7に示す構成になっています。

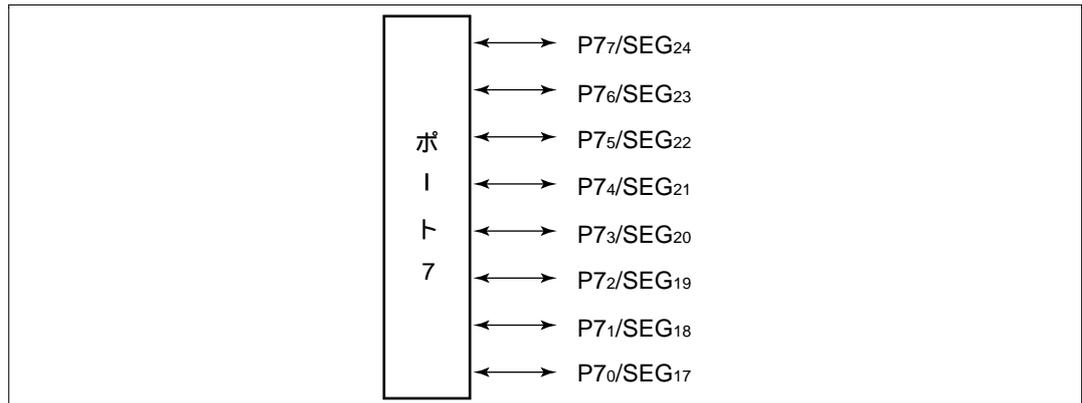


図 8.7 ポート7の端子構成

8.8.2 レジスタの構成と説明

表8.20にポート7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
	P77	P76	P75	P74	P73	P72	P71	P70
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P7₇~P7₀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 ₇	PCR7 ₆	PCR7 ₅	PCR7 ₄	PCR7 ₃	PCR7 ₂	PCR7 ₁	PCR7 ₀
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P7₇~P7₀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P7₇~P7₀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。LPCRのSGS3~SGS0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、PCR7およびPDR7の設定が有効となります。

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。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7の端子機能を示します。

表8.21 ポート7の端子機能

端子	選択方法と端子機能		
P7 ₇ /SEG ₂₄ ~ P7 ₀ /SEG ₁₇	PCR7のPCR7 _n とLPCRのSGS3~SGS0の組合せで、次のように切替わります。 (n=7~0)		
	SGS3~SGS0	00**	
		01**、1***	
	PCR7 _n	0	1
	端子機能	P7 _n 入力端子	P7 _n 出力端子
			SEG _{n+17} 出力端子

* : Don't care

8.8.4 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8.22に示します。

表8.22 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 ₇ /SEG ₂₄ ~ P7 ₀ /SEG ₁₇	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 8

8.9.1 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 8.8 に示す構成になっています。

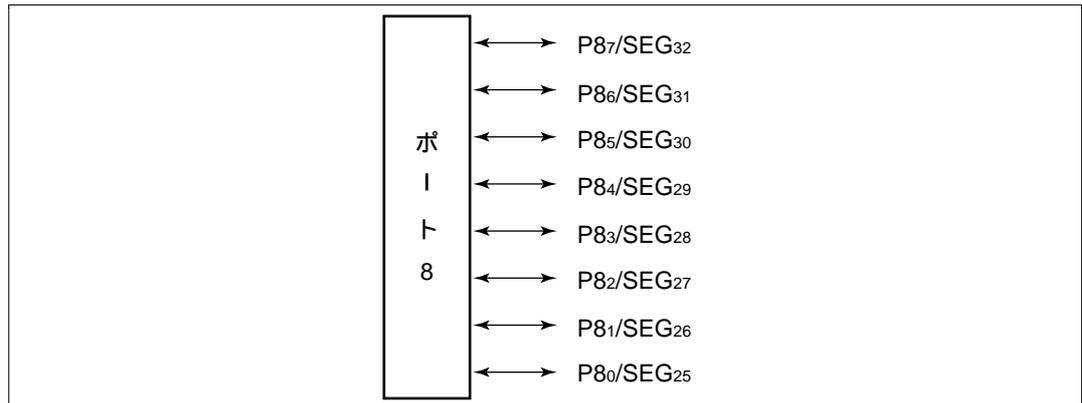


図 8.8 ポート 8 の端子構成

8.9.2 レジスタの構成と説明

表 8.23 にポート 8 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	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 P8₇ ~ P8₀ 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
	PCR8 ₇	PCR8 ₆	PCR8 ₅	PCR8 ₄	PCR8 ₃	PCR8 ₂	PCR8 ₁	PCR8 ₀
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 P8₇ ~ P8₀ 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 P8₇ ~ P8₀ 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。LPCR の SGS3 ~ SGS0 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、PCR8 および PDR8 の設定が有効となります。

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。

8.9.3 端子機能

表 8.24 にポート 8 の端子機能を示します。

表 8.24 ポート 8 の端子機能

端子	選択方法と端子機能		
P8 ₇ /SEG ₃₂ ~ P8 ₀ /SEG ₂₅	PCR8 の PCR8 _n と LPCR の SGS3 ~ SGS0 の組合せで、次のように切替わります。 (n = 7 ~ 0)		
	SGS3 ~ SGS0	000 *	
		001 *, 01 * *, 1 * * *	
	PCR8 _n	0	1
		*	
	端子機能	P8 _n 入力端子	P8 _n 出力端子
			SEG _{n+25} 出力端子

* : Don't care

8.9.4 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 8.25 に示します。

表 8.25 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 ₇ /SEG ₃₂ ~ P8 ₀ /SEG ₂₅	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

8.10 ポート 9

8.10.1 概要

ポート 9 は、8 ビットの入出力ポートです。ポート 9 の各端子は、図 8.9 に示す構成になっています。

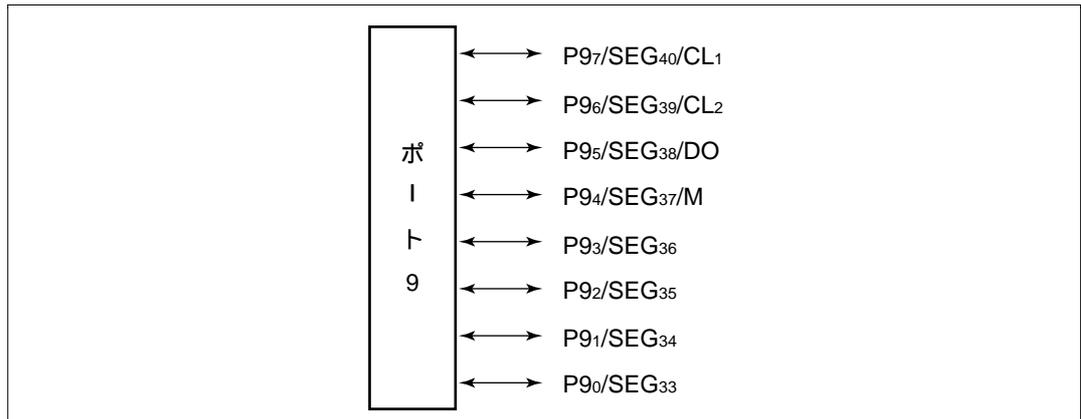


図 8.9 ポート 9 の端子構成

8.10.2 レジスタの構成と説明

表 8.26 にポート 9 のレジスタ構成を示します。

表 8.26 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 9	PDR9	R/W	H'00	H'FFDC
ポートコントロールレジスタ 9	PCR9	W	H'00	H'FFEC

(1) ポートデータレジスタ 9 (PDR9)

ビット:	7	6	5	4	3	2	1	0
	P97	P96	P95	P94	P93	P92	P91	P90
初期値:	0	0	0	0	0	0	0	0
R/W :	R/W							

PDR9 は、ポート 9 の各端子 P9₇ ~ P9₀ のデータを格納する 8 ビットのレジスタです。

PCR9 が "1" のとき、ポート 9 のリードを行うと、PDR9 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。PCR9 が "0" のとき、ポート 9 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、PDR9 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 9 (PCR9)

ビット:	7	6	5	4	3	2	1	0
	PCR9 ₇	PCR9 ₆	PCR9 ₅	PCR9 ₄	PCR9 ₃	PCR9 ₂	PCR9 ₁	PCR9 ₀
初期値:	0	0	0	0	0	0	0	0
R/W :	W	W	W	W	W	W	W	W

PCR9は、ポート9の各端子P9₇~P9₀の入出力をビットごとに制御します。PCR9に"1"をセットすると対応するP9₇~P9₀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。LPCRのSGS3~SGS0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、PCR9およびPDR9の設定が有効となります。

リセット時、PCR9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。リードした場合各ビットは常に"1"が読み出されます。

8.10.3 端子機能

表 8.27 にポート9の端子機能を示します。

表 8.27 ポート9の端子機能(1)

端子	選択方法と端子機能				
P9 ₇ /SEG ₄₀ /CL ₁	PCR9のPCR9 ₇ とLPCRのSGX、SGS3~SGS0の組合せで、次のように切替わります。				
	SGS3~SGS0	0000		0000以外	*
	SGX	0		0	1
	PCR9 ₇	0	1	*	*
	端子機能	P9 ₇ 入力端子	P9 ₇ 出力端子	SEG ₄₀ 出力端子	CL ₁ 出力端子
P9 ₆ /SEG ₃₉ /CL ₂	PCR9のPCR9 ₆ とLPCRのSGX、SGS3~SGS0の組合せで、次のように切替わります。				
	SGS3~SGS0	0000		0000以外	*
	SGX	0		0	1
	PCR9 ₆	0	1	*	*
	端子機能	P9 ₆ 入力端子	P9 ₆ 出力端子	SEG ₃₉ 出力端子	CL ₂ 出力端子
P9 ₅ /SEG ₃₈ /DO	PCR9のPCR9 ₅ とLPCRのSGX、SGS3~SGS0の組合せで、次のように切替わります。				
	SGS3~SGS0	0000		0000以外	*
	SGX	0		0	1
	PCR9 ₅	0	1	*	*
	端子機能	P9 ₅ 入力端子	P9 ₅ 出力端子	SEG ₃₈ 出力端子	DO出力端子

表 8.27 ポート 9 の端子機能 (2)

端子	選択方法と端子機能				
P9 ₄ /SEG ₃₇ /M	PCR9 の PCR9 ₄ と LPCR の SGX、SGS3 ~ SGS0 の組合せで、次のように切り替わります。				
	SGS3 ~ SGS0	0000		0000 以外	*
	SGX	0		0	1
	PCR9 ₄	0	1	*	*
	端子機能	P9 ₄ 入力端子	P9 ₄ 出力端子	SEG ₃₇ 出力端子	M 出力端子
P9 ₃ /SEG ₃₆ ~ P9 ₀ /SEG ₃₃	PCR9 の PCR9 _n と LPCR の SGS3 ~ SGS0 の組合せで、次のように切り替わります。 (n = 3 ~ 1)				
	SGS3 ~ SGS0	0000		0000 以外	*
	PCR9 _n	0	1	*	*
	端子機能	P9 _n 入力端子	P9 _n 出力端子	SEG _{n+33} 出力端子	

* : Don't care

8.10.4 端子状態

各動作モードにおけるポート 9 の端子状態を表 8.28 に示します。

表 8.28 ポート 9 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
P9 ₇ /SEG ₄₀ /CL ₁	ハイ	保持	保持	ハイ	保持	動作	動作
P9 ₆ /SEG ₃₉ /CL ₂	インピー			インピー			
P9 ₅ /SEG ₃₈ /DO	ダンス			ダンス			
P9 ₄ /SEG ₃₇ /M							
P9 ₃ /SEG ₃₆ ~							
P9 ₀ /SEG ₃₃							

8.11 ポート A

8.11.1 概要

ポート A は、4 ビットの入出力ポートです。ポート A の各端子は、図 8.10 に示す構成になっています。

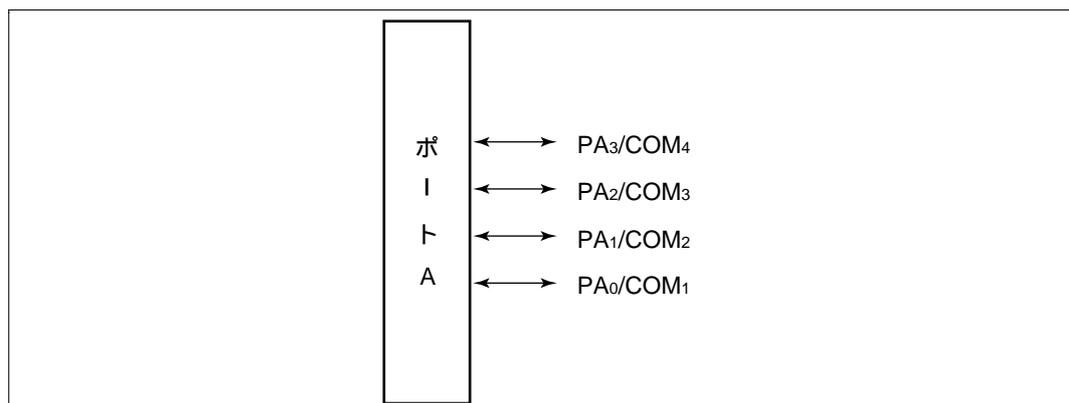


図 8.10 ポート A の端子構成

8.11.2 レジスタの構成と説明

表 8.29 にポート A のレジスタ構成を示します。

表 8.29 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'F0	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 ₃	PA ₂	PA ₁	PA ₀
初期値:	1	1	1	1	0	0	0	0
R/W :	—	—	—	—	R/W	R/W	R/W	R/W

PDRA は、ポート A の各端子 PA₃ ~ PA₀ 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 HF0 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
	—	—	—	—	PCRA ₃	PCRA ₂	PCRA ₁	PCRA ₀
初期値:	1	1	1	1	0	0	0	0
R/W :	—	—	—	—	W	W	W	W

PCRA は、ポート A の各端子 PA₃ ~ PA₀ 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 PA₃ ~ PA₀ 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。LPCR 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、PCRA および PDRA の設定が有効となります。

リセット時、PCRA は HF0 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。リードした場合各ビットは常に "1" が読み出されます。

8.11.3 端子機能

表 8.30 にポート A の端子機能を示します。

表 8.30 ポート A の端子機能

端子	選択方法と端子機能		
PA ₃ /COM ₄	PCRA の PCRA ₃ と SGS3 ~ SGS0 の組合せで、次のように切替わります。		
	SGS3 ~ SGS0	0000	
	PCRA ₃	0	1
	端子機能	PA ₃ 入力端子	PA ₃ 出力端子
PA ₂ /COM ₃	PCRA の PCRA ₂ と SGS3 ~ SGS0 の組合せで、次のように切替わります。		
	SGS3 ~ SGS0	0000	
	PCRA ₂	0	1
	端子機能	PA ₂ 入力端子	PA ₂ 出力端子
PA ₁ /COM ₂	PCRA の PCRA ₁ と SGS3 ~ SGS0 の組合せで、次のように切替わります。		
	SGS3 ~ SGS0	0000	
	PCRA ₁	0	1
	端子機能	PA ₁ 入力端子	PA ₁ 出力端子
PA ₀ /COM ₁	PCRA の PCRA ₀ と SGS3 ~ SGS0 の組合せで、次のように切替わります。		
	SGS3 ~ SGS0	0000	
	PCRA ₀	0	1
	端子機能	PA ₀ 入力端子	PA ₀ 出力端子

* : Don't care

8.11.4 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 8.31 に示します。

表 8.31 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 ₃ /COM ₄	ハイ	保持	保持	ハイ	保持	動作	動作
PA ₂ /COM ₃	インピー			インピー			
PA ₁ /COM ₂	ダンス			ダンス			
PA ₀ /COM ₁							

8.12 ポート B

8.12.1 概要

ポート B は、8 ビットの入力専用ポートです。ポート B の各端子は、図 8.11 に示す構成になっています。

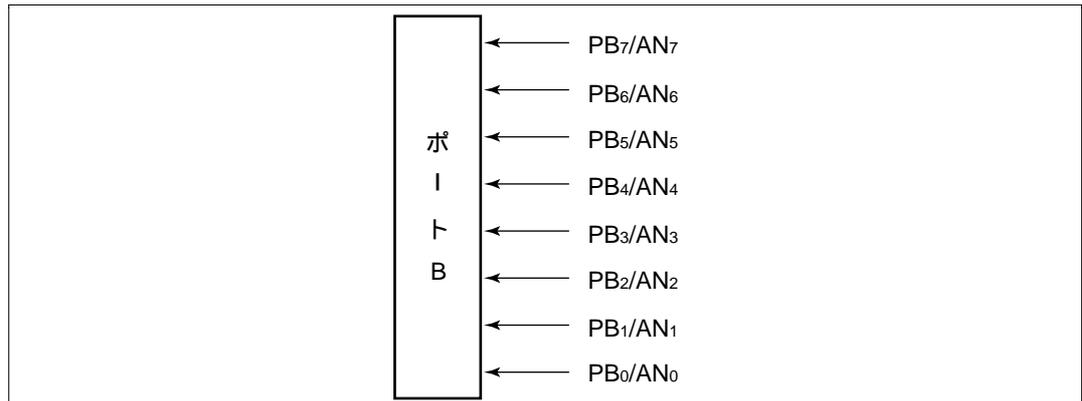


図 8.11 ポート B の端子構成

8.12.2 レジスタの構成と説明

表 8.32 にポート B のレジスタ構成を示します。

表 8.32 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	PB5	PB4	PB3	PB2	PB1	PB0
R/W :	R	R	R	R	R	R	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。

8.13 ポート C

8.13.1 概要

ポートCは、4ビットの入力専用ポートです。ポートCの各端子は、図8.12に示す構成になっています。

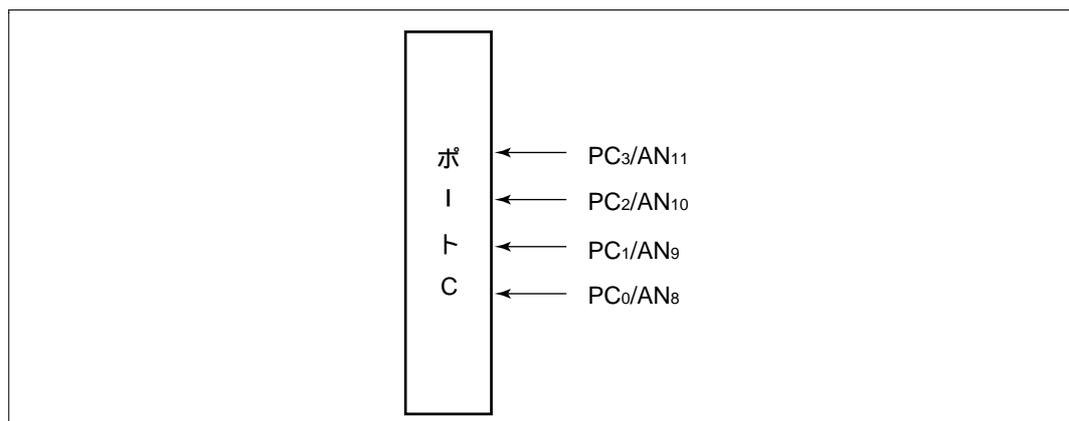


図 8.12 ポートCの端子構成

8.13.2 レジスタの構成と説明

表 8.33 にポートCのレジスタ構成を示します。

表 8.33 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C	PDRC	R	H'FFDF

(1) ポートデータレジスタC (PDRC)

ビット:	7	6	5	4	3	2	1	0
					PC3	PC2	PC1	PC0
R/W :					R	R	R	R

PDRC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。

8.14 入出力データ反転機能

8.14.1 概要

入力端子の RXD_{31} 、 RXD_{32} と出力端子の TXD_{31} 、 TXD_{32} はデータを反転させて取り扱うことが可能です。

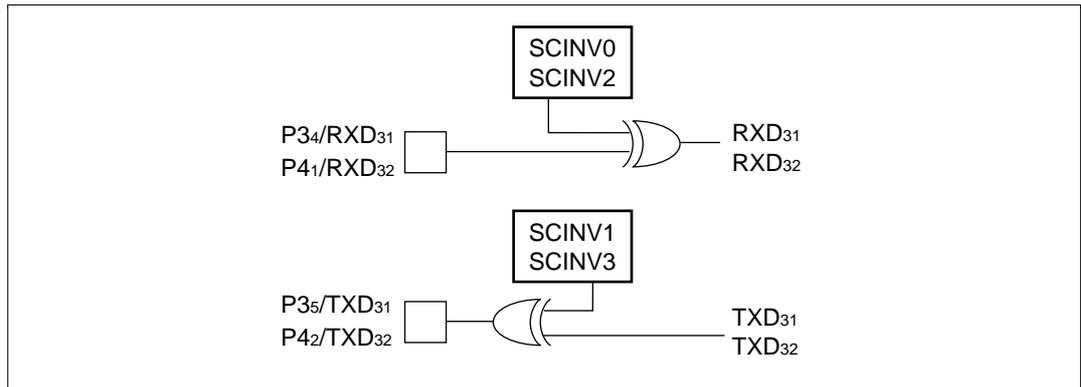


図 8.13 入出力データ反転機能

8.14.2 レジスタの構成と説明

表 8.34 に入出力データ反転機能のレジスタ構成を示します。

表 8.34 レジスタ構成

名称	略称	R/W	アドレス
シリアルポートコントロールレジスタ	SPCR	R/W	H'FF91

(1) シリアルポートコントロールレジスタ (SPCR)

ビット:	7	6	5	4	3	2	1	0
			SPC32	SPC31	SCINV3	SCINV2	SCINV1	SCINV0
初期値:	1	1	0	0	0	0	0	0
R/W :			R/W	R/W	R/W	R/W	R/W	R/W

SPCR は、8 ビットのリード/ライト可能なレジスタで、 RXD_{31} 、 RXD_{32} 、 TXD_{31} 、 TXD_{32} 端子の入出力データ反転切換えを行います。リセット時、SPCR は H'C0 にイニシャライズされます。

ビット7～6：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：P4₂/TXD₃₂端子機能切換え

P4₂/TXD₃₂端子をP4₂端子として使用するかTXD₃₂端子として使用するかを設定します。

ビット5		
SPC32	説明	
0	P4 ₂ 入出力端子として機能。	(初期値)
1	TXD ₃₂ 出力端子として機能。*	

【注】 * 本ビットを"1"に設定した後に SCR3 の TE ビットを設定してください。

ビット4：P3₅/TXD₃₁端子機能切換え

P3₅/TXD₃₁端子をP3₅端子として使用するかTXD₃₁端子として使用するかを設定します。

ビット4		
SPC31	説明	
0	P3 ₅ 入出力端子として機能。	(初期値)
1	TXD ₃₁ 出力端子として機能。*	

【注】 * 本ビットを"1"に設定した後に SCR3 の TE ビットを設定してください。

ビット3：TXD₃₂端子出力データ反転切換え

TXD₃₂端子の出力データを反転するか、しないか、の切換えを設定します。

ビット3		
SCINV3	説明	
0	TXD ₃₂ の出力データを反転しない。	(初期値)
1	TXD ₃₂ の出力データを反転する。	

ビット2：RXD₃₂端子入力データ反転切換え

RXD₃₂端子の入力データを反転するか、しないか、の切換えを設定します。

ビット2		
SCINV2	説明	
0	RXD ₃₂ の入力データを反転しない。	(初期値)
1	RXD ₃₂ の入力データを反転する。	

ビット1：TXD₃₁ 端子出力データ反転切換え

TXD₃₁ 端子の出力データを反転するか、しないか、の切換えを設定します。

ビット1		
SCINV1	説明	
0	TXD ₃₁ の出力データを反転しない。	(初期値)
1	TXD ₃₁ の出力データを反転する。	

ビット0：RXD₃₁ 端子入力データ反転切換え

RXD₃₁ 端子の入力データを反転するか、しないか、の切換えを設定します。

ビット0		
SCINV0	説明	
0	RXD ₃₁ の入力データを反転しない。	(初期値)
1	RXD ₃₁ の入力データを反転する。	

8.14.3 シリアルポートコントロールレジスタを書換える際の注意事項

シリアルポートコントロールレジスタを書換えると、それまで入力または出力されていたデータが書換えた直後に反転され、有効ではないデータの変化が入出力されます。シリアルポートコントロールレジスタを書換える際には、データ変化を無効にする状態で、書換えてください。

9. タイマ

第9章 目次

9.1	概要	197
9.2	タイマ A	198
	9.2.1 概要	198
	9.2.2 各レジスタの説明	200
	9.2.3 動作説明	204
	9.2.4 タイマ A の動作モード	205
9.3	タイマ C	206
	9.3.1 概要	206
	9.3.2 各レジスタの説明	208
	9.3.3 動作説明	212
	9.3.4 タイマ C の動作モード	213
9.4	タイマ F	214
	9.4.1 概要	214
	9.4.2 各レジスタの説明	217
	9.4.3 CPU とのインタフェース	225
	9.4.4 動作説明	227
	9.4.5 使用上の注意事項	230
9.5	タイマ G	232
	9.5.1 概要	232
	9.5.2 各レジスタの説明	234
	9.5.3 ノイズ除去回路	239
	9.5.4 動作説明	240
	9.5.5 使用上の注意事項	245
	9.5.6 タイマ G の使用例	249
9.6	ウォッチドッグタイマ	250
	9.6.1 概要	250
	9.6.2 各レジスタの説明	251

9.6.3	動作説明	255
9.6.4	ウォッチドッグタイマの動作モード	256
9.7	非同期イベントカウンタ (AEC)	257
9.7.1	概要	257
9.7.2	各レジスタの構成	259
9.7.3	動作説明	264
9.7.4	非同期イベントカウンタの動作モード	265
9.7.5	使用上の注意事項	266

9.1 概要

本LSIは6本のタイマ（タイマA、C、F、G、ウォッチドッグタイマ、非同期イベントカウンタ）を内蔵しています。

各タイマの機能概要を表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	$w/128$ (オーバフロー周期は4種類選択可)			
	・クロック出力機能	$/4 \sim /32$ w 、 $w/4 \sim w/32$ (9種類)		TMOW	
タイマC	・8ビットのタイマ ・インターバル機能 ・イベントカウント機能 ・カウントアップ/ダウン 選択可能	$/4 \sim /8192$ $w/4$ (7種類)	TMIC		カウントアップ/ダウンはソフトウェア制御、ハードウェア制御ともに可能
タイマF	・16ビットのタイマ ・イベントカウント機能 ・独立した2本の8ビット タイマとして使用可能 ・アウトプットコンペア出力機能	$/4 \sim /32$ 、 $w/4$ (4種類)	TMIF	TMOFL TMOFH	
タイマG	・8ビットのタイマ ・インプットキャプチャ機能 ・インターバル機能	$/2 \sim /64$ $w/4$ (4種類)	TMIG		・カウンタのクリア指定が可能 ・キャプチャ入力信号のノイズ除去回路内蔵
ウォッチドッグタイマ	・8ビットカウンタのオーバフローでリセット信号を発生	$/8192$ $w/32$			
非同期イベントカウンタ	・16ビットのカウンタ ・独立した2本の8ビット カウンタとして使用可能 ・ w と非同期のイベント機能		AEVL AEVH		

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、また 38.4kHz の水晶発振器を接続した場合は、38.4kHz 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 の分周無し、32、16、8、4 分周したクロック (1kHz、2kHz、4kHz、8kHz、32.768kHz) または、38.4kHz の分周無し、32、16、8、4 分周したクロック (1.2kHz、2.4kHz、4.8kHz、9.6kHz、38.4kHz)、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 9 種類の選択が可能です。

モジュールスタンバイモードにより、未使用時はモジュール単体でスタンバイモードに設定可能

(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。

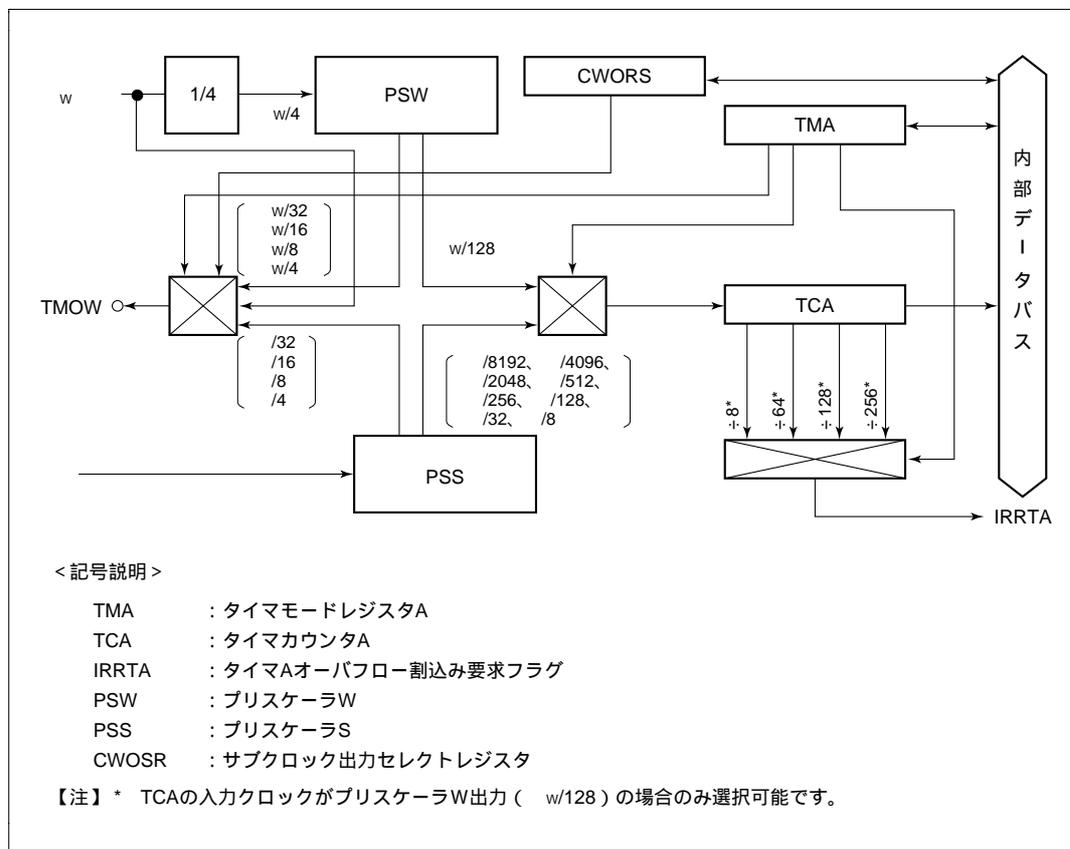


図 9.1 タイマ A ブロック図

(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
クロック停止レジスタ 1	CKSTPR1	R/W	H'FF	H'FFFA
サブクロック出力セレクトレジスタ	CWOSR	R/W	H'FE	H'FF92

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または38.4kHz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。 w はリセット以外のすべてのモードで出力されます。

CWOSR	TMA			説明
	ビット7	ビット6	ビット5	
CWOS	TMA7	TMA6	TMA5	
0	0	0	0	/32 (初期値)
			1	/16
		1	0	/8
			1	/4
	1	0	0	$w/32$
			1	$w/16$
		1	0	$w/8$
			1	$w/4$
1	*	*	*	w

* : Don't care

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	プリスケア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 (初期値)
			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 (32.768kHz 使用時)
			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には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IRRTA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。

(3) クロック停止レジスタ 1 (CKSTPR1)

ビット:	7	6	5	4	3	2	1	0
	S1CKSTP	S31CKSTP	S32CKSTP	ADCKSTP	TGCKSTP	TFCKSTP	TCCKSTP	TACKSTP
初期値:	1	1	1	1	1	1	1	1
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CKSTPR1 は、8 ビットのリード/ライト可能なレジスタで、モジュールのモジュールスタンバイモードの制御を行います。本章ではタイマ A に関するビットのみ説明します。他のビットについては各モジュールの章を参照してください。

ビット 0 : タイマ A モジュールスタンバイモード制御 (TACKSTP)

タイマ A をモジュールスタンバイモードに設定および解除を制御します。

TACKSTP	説明
0	タイマ A はモジュールスタンバイモードに設定される
1	タイマ A のモジュールスタンバイモードは解除される (初期値)

(4) サブクロック出力セレクトレジスタ (CWOSR)

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

CWOSR は、8 ビットのリード/ライト可能なレジスタで、TMOW 端子から出力するクロックを選択します。

リセット時、CWOSR は H'FE にイニシャライズされます。

ビット 7~1 : 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 0 : TMOW 端子クロックセレクト

TMOW 端子から出力するクロックを選択します。

ビット 0	説明
CWOS	
0	タイマ A が出力するクロックを出力する。(TMA 参照) (初期値)
1	w を出力する。

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、プリスケアラ 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 に割り込みを要求します。*

オーバフロー時には、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。

(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 はプリスケアラ 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 およびプリスケアラ 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 と CWOSR の CWOS により、9 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 または 38.4kHz 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。32.768kHz または 38.4kHz のクロックはリセット状態以外のすべての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	モジュール 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	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	停止
TMA、CWOSR		リセット	動作	保持	保持	動作	保持	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 C

9.3.1 概要

タイマ C は、入力クロックが入るたびにカウントアップまたはカウントダウンする 8 ビットのタイマです。タイマ C には、インターバル機能、オートリロード機能の 2 種類の機能があります。

(1) 特長

タイマ C の特長を以下に示します。

8 種類のクロックを選択可能

7 種類の内部クロック (/8192、 /2048、 /512、 /64、 /16、 /4、 $w/4$) 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) 。

カウンタのオーバフローで割込みを発生

アップ / ダウンカウンタ切換え可能

ハードウェアまたはソフトウェアにより、アップ / ダウンカウンタの切換えが可能です。

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 $w/4$ を選択した場合、もしくは外部クロックを選択した場合、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。

モジュールスタンバイモードにより、未使用時はモジュール単体でスタンバイモードに設定可能。

(2) ブロック図

タイマCのブロック図を図9.2に示します。

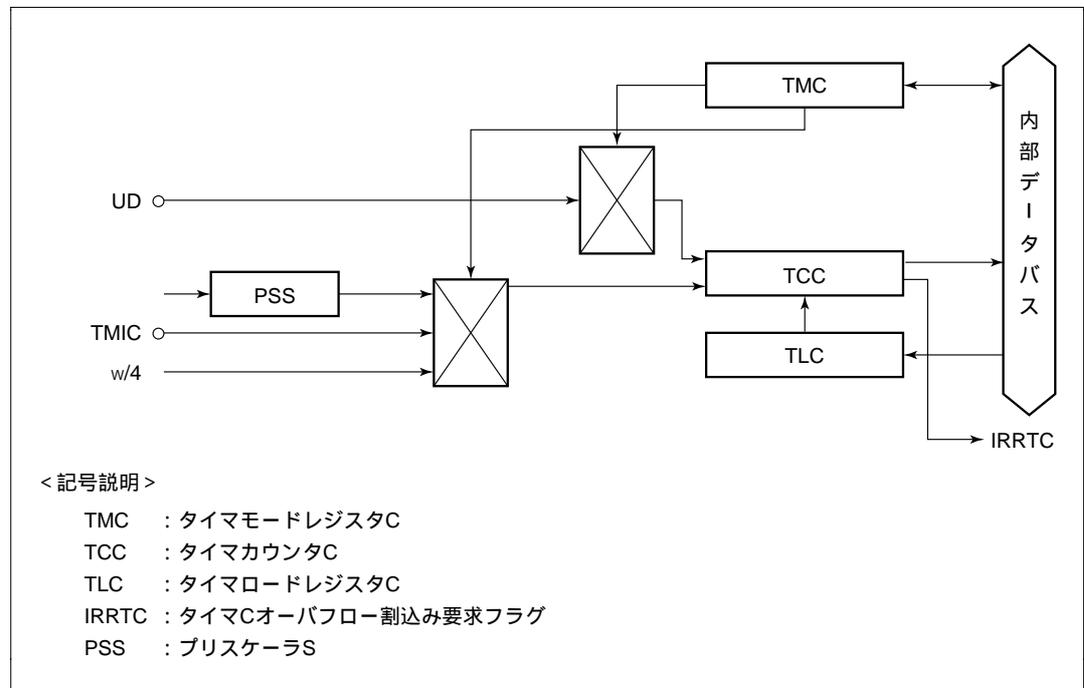


図9.2 タイマCブロック図

(3) 端子構成

タイマC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Cイベント入力	TMIC	入力	TCCに入力するイベント入力端子
タイマCアップ/ダウン セレクト	UD	入力	タイマCのアップ/ダウンカウントを選択

(4) レジスタ構成

タイマC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
タイマモードレジスタC	TMC	R/W	H'18	H'FFB4
タイマカウンタC	TCC	R	H'00	H'FFB5
タイマロードレジスタC	TLC	W	H'00	H'FFB5
クロック停止レジスタ1	CKSTPR1	R/W	H'FF	H'FFFA

9.3.2 各レジスタの説明

(1) タイマモードレジスタC (TMC)

ビット:	7	6	5	4	3	2	1	0
	TMC7	TMC6	TMC5	—	—	TMC2	TMC1	TMC0
初期値:	0	0	0	1	1	0	0	0
R/W :	R/W	R/W	R/W	—	—	R/W	R/W	R/W

TMCは、8ビットのリード/ライト可能なレジスタで、オートリロード機能の選択、カウンタのアップ/ダウン制御、および入力クロックの選択を行います。

リセット時、TMCはH'18にイニシャライズされます。

ビット7: オートリロード機能選択 (TMC7)

タイマCのオートリロード機能を選択します。

ビット7	説明
TMC7	
0	インターバル機能を選択 (初期値)
1	オートリロード機能を選択

ビット6、5：カウンタアップ/ダウン制御（TMC6、TMC5）

TCCのアップ/ダウン制御をUD端子入力によるハードウェア制御とするか、アップカウンタとするかダウンカウンタとするかを選択します。

ビット6	ビット5	説明
TMC6	TMC5	
0	0	TCCはアップカウンタ (初期値)
0	1	TCCはダウンカウンタ
1	*	UD端子入力によるハードウェア制御 UD端子入力が"High"レベル：ダウンカウンタ UD端子入力が"Low"レベル：アップカウンタ

* : Don't care

ビット4～3：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2～0：クロックセレクト（TMC2～TMC0）

TMC2～TMC0は、TCCに入力するクロックを選択します。外部からのイベント入力は、立上がり/立下がりエッジの選択が可能です。

ビット2	ビット1	ビット0	説明
TMC2	TMC1	TMC0	
0	0	0	内部クロック /8192 でカウント (初期値)
0	0	1	内部クロック /2048 でカウント
0	1	0	内部クロック /512 でカウント
0	1	1	内部クロック /64 でカウント
1	0	0	内部クロック /16 でカウント
1	0	1	内部クロック /4 でカウント
1	1	0	内部クロック $\omega/4$ でカウント
1	1	1	外部イベント (TMIC) を立上がり/立下がりエッジでカウント*

【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IEG1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、TMC2～TMC0を"111"に設定する前に必ずポートモードレジスタ1 (PMR1) のIRQ2を"1"にセットしてください。

(2) タイマカウンタ C (TCC)

ビット:	7	6	5	4	3	2	1	0
	TCC7	TCC6	TCC5	TCC4	TCC3	TCC2	TCC1	TCC0
初期値:	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R

TCCは、8ビットのリード可能なアップ/ダウンカウンタで、入力する内部クロック/外部イベントによりカウントアップ/ダウンされます。入力するクロックは、TMCのTMC2~TMC0により選択します。TCCの値は、CPUから常にリードできます。

TCCがオーバーフロー (H'FF H'00またはH'FF TLCの設定値)、またはアンダフロー (H'00 H'FFまたはH'00 TLCの設定値)すると、IRR2のIRRTCが"1"にセットされます。

TCCは、TLCと同一のアドレスに割り付けられています。

リセット時、TCCはH'00にイニシャライズされます。

(3) タイマロードレジスタ C (TLC)

ビット:	7	6	5	4	3	2	1	0
	TLC7	TLC6	TLC5	TLC4	TLC3	TLC2	TLC1	TLC0
初期値:	0	0	0	0	0	0	0	0
R/W :	W	W	W	W	W	W	W	W

TLCは、8ビットのライト専用のレジスタで、TCCのリロード値を設定します。TLCにリロード値を設定すると、同時にその値はTCCにもロードされ、TCCはその値からカウントアップ/ダウンを開始します。また、オートリロード動作時、TCCがオーバーフロー/アンダフローすると、TCCにTLCの値がロードされます。したがって、オーバーフロー/アンダフロー周期を1~256入力クロックの範囲で設定することができます。

TLCは、TCCと同一のアドレスに割り付けられています。

リセット時、TLCはH'00にイニシャライズされます。

(4) クロック停止レジスタ 1 (CKSTPR1)

ビット:	7	6	5	4	3	2	1	0
	S1CKSTP	S31CKSTP	S32CKSTP	ADCKSTP	TGCKSTP	TFCKSTP	TCCKSTP	TACKSTP
初期値:	1	1	1	1	1	1	1	1
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CKSTPR1 は、8 ビットのリード/ライト可能なレジスタで、モジュールのモジュールスタンバイモードの制御を行います。本章ではタイマ C に関するビットのみ説明します。他のビットについては各モジュールの章を参照してください。

ビット 1 : タイマ C モジュールスタンバイモード制御 (TCCKSTP)

タイマ C をモジュールスタンバイモードに設定および解除を制御します。

TCCKSTP	説明
0	タイマ C はモジュールスタンバイモードに設定される
1	タイマ C のモジュールスタンバイモードは解除される (初期値)

9.3.3 動作説明

(1) インターバルタイマの動作

TMC の TMC7 を "0" にクリアすると、タイマ C 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、TCC は H'00、TMC は H'18 に初期化されるため、リセット直後は、インターバルのアップカウンタ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 C の動作クロックは、TMC の TMC2 ~ TMC0 により、プリスケラ S、W の出力する 7 種類の内部クロック、TMIC 入力端子からの外部クロックを選択できます。

また、TCC のカウントアップ/ダウン制御は、TMC の TMC6、TMC5 により、ソフトウェア制御およびハードウェア制御のいずれかが選択可能です。

TCC のカウント値が H'FF (H'00) になった後、クロックが入力されると、タイマ C はオーバフロー(アンダフロー)し、IRR2 の IRRTC が "1" にセットされます。このとき、IENR2 の IENTC が "1" ならば CPU に割込みを要求します。

オーバフロー(アンダフロー)時には、TCC のカウント値は H'00 (H'FF) に戻り、再びカウントアップ(ダウン)を開始します。

インターバル動作時 (TMC7 = "0") に TLC を設定すると、同時に TCC にも TLC の値をロードします。

【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) オートリロードタイマの動作

TMC の TMC7 = "1" にセットすると、タイマ C は 8 ビットオートリロードタイマとして動作します。

TLC にリロード値を設定すると、同時にその値が TCC にロードされ、TCC はその値からカウントアップ/ダウンを開始します。TCC のカウント値が H'FF (H'00) になった後、クロックが入力されると、タイマ C はオーバフロー(アンダフロー)し、TLC の値が TCC にロードされ、その値からカウントアップ(ダウン)を続けます。したがって、TLC の値により、オーバフロー(アンダフロー)周期を 1 ~ 256 入力クロックの範囲で設定できます。

オートリロード動作時のクロック、アップ/ダウン制御、割込みについてはインターバル動作時と同様です。

オートリロード動作時 (TMC7 = "1") に TLC の値を再設定すると、同時に TCC にも TLC の値をロードします。

(3) イベントカウンタ

タイマCは、TMIC端子をイベント入力端子とするイベントカウンタとして動作します。TMCのTMC2～TMC0を"111"に設定すると、外部イベント動作が選択され、TCCは、TMIC端子入力の立上がり/立下がりエッジでカウントアップ/ダウンします。

外部イベント入力を使用する場合は、PMR1のIRQ2を"1"にセットし、かつIENR1のIEN2を"0"としてIRQ₂割込み要求を禁止してください。

(4) ハードウェアによるTCCアップ/ダウン制御

タイマCは、UD端子入力によるTCCのカウントアップ/ダウン制御ができます。TMCのTMC6を"1"とすると、UD端子入力が、"High"レベルならばダウンカウンタ、"Low"レベルならばアップカウンタとして動作します。

UD端子入力を使用する場合は、PMR3のUDを"1"にセットしてください。

9.3.4 タイマCの動作モード

タイマCの動作モードを表9.7に示します。

表9.7 タイマC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	モジュールスタンバイ
TCC	インターバル	リセット	動作	動作	停止	動作/停止*	動作/停止*	停止	停止
	オートリロード	リセット	動作	動作	停止	動作/停止*	動作/停止*	停止	停止
TMC		リセット	動作	保持	保持	動作	保持	保持	保持

【注】 * アクティブモード、スリープモード時に、TCCの内部クロックとして $\omega/4$ 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 $1/\omega$ (s)の誤差が生じます。

サブアクティブモード、サブスリープモード時にカウンタを動作させる場合には、必ず内部クロックとして $\omega/4$ を選択するか、または外部クロックを選択してください。その他の内部クロックを選択した場合、カウンタは動作しません。また、サブクロック ω_{SUB} として $\omega/8$ を選択時にカウンタの内部クロックとして $\omega/4$ を選択した場合、カウンタの下位2ビットは同じ周期で動作し、最下位ビットの動作はカウンタ動作とは無関係となります。

9.4 タイマ F

9.4.1 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タが可能のほか、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（ $/32$ 、 $/16$ 、 $/4$ 、 $w/4$ ）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タが可能）。

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（ $/32$ 、 $/16$ 、 $/4$ 、 $w/4$ ）	
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 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。

ウォッチモード、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 $w/4$ を選択した場合、ウォッチモード、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。

モジュールスタンバイモードにより、未使用時はモジュール単体でスタンバイモードに設定可能。

(2) ブロック図

タイマFのブロック図を図9.3に示します。

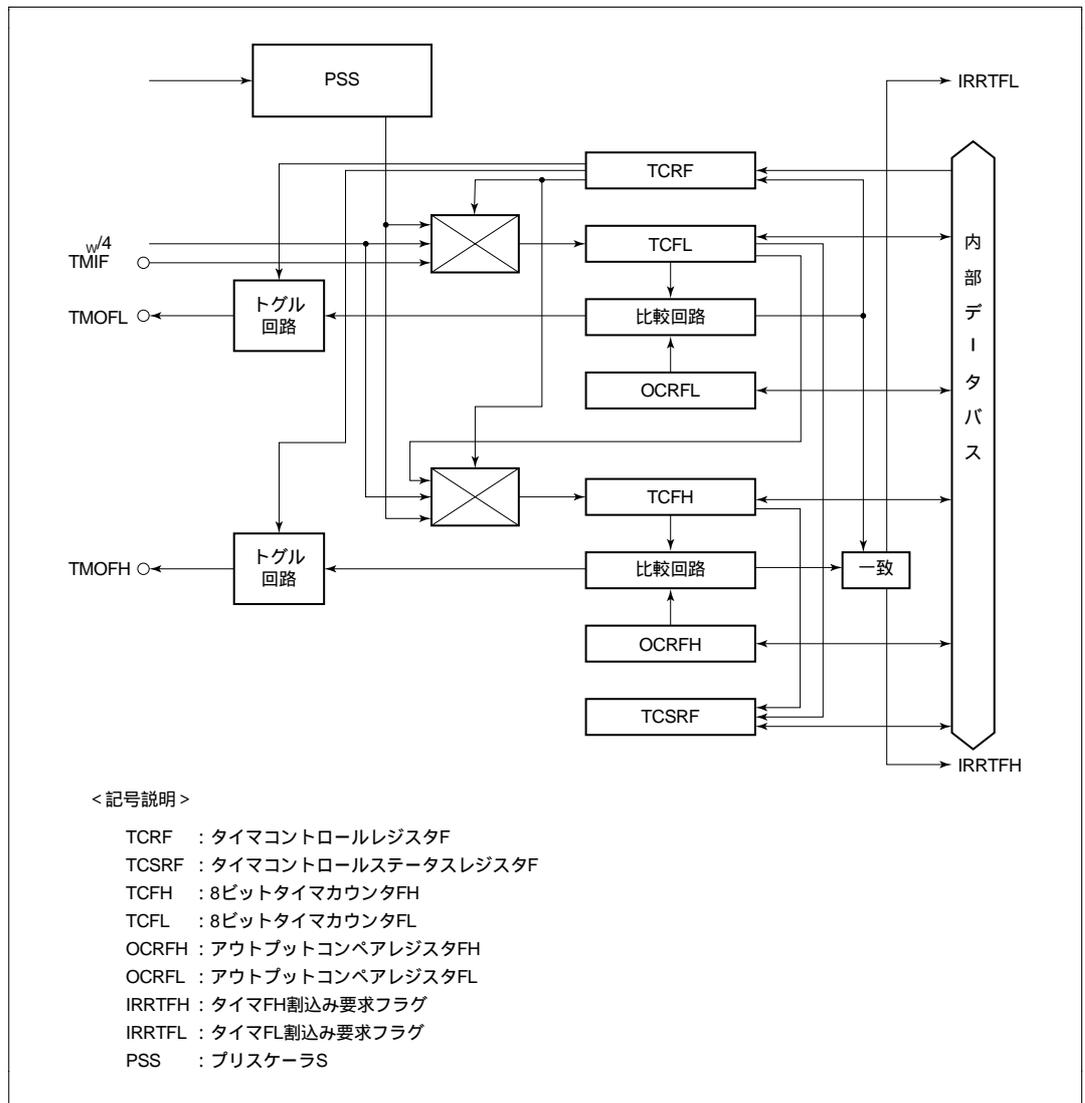


図9.3 タイマFのブロック図

(3) 端子構成

タイマFの端子構成を表9.8に示します。

表9.8 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

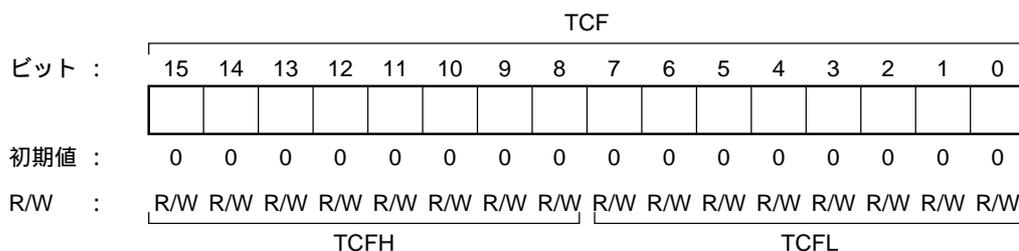
タイマFのレジスタ構成を表9.9に示します。

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
クロック停止レジスタ1	CKSTPR1	R/W	H'FF	H'FFFA

9.4.2 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「9.4.3 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 に割り込みを要求します。

- (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															
ビット :		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
初期値 :		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W :		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		OCRFH								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「9.4.3 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に割り込みを要求します。

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に割り込みを要求します。

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	0	0	(初期値)
0	0	1	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
0	1	0	
0	1	1	使用禁止
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 : $_{w}/4$ でカウント

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	0	0	(初期値)
0	0	1	外部イベント (TMIF) の立上がり / 立下がりエッジでカウント*
0	1	0	
0	1	1	使用禁止
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： $\omega/4$ でカウント

【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	CCLRL
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。

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

(5) クロック停止レジスタ 1 (CKSTPR1)

ビット:	7	6	5	4	3	2	1	0
	S1CKSTP	S31CKSTP	S32CKSTP	ADCKSTP	TGCKSTP	TFCKSTP	TCCKSTP	TACKSTP
初期値:	1	1	1	1	1	1	1	1
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CKSTPR1 は、8 ビットのリード/ライト可能なレジスタで、モジュールのモジュールスタンバイモードの制御を行います。本章ではタイマ F に関するビットのみ説明します。他のビットについては各モジュールの章を参照してください。

ビット 2 : タイマ F モジュールスタンバイモード制御 (TFCKSTP)

タイマ F をモジュールスタンバイモードに設定及び解除を制御します。

TFCKSTP	説明
0	タイマ F はモジュールスタンバイモードに設定される
1	タイマ F のモジュールスタンバイモードは解除される (初期値)

9.4.3 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 9.4 に示します。

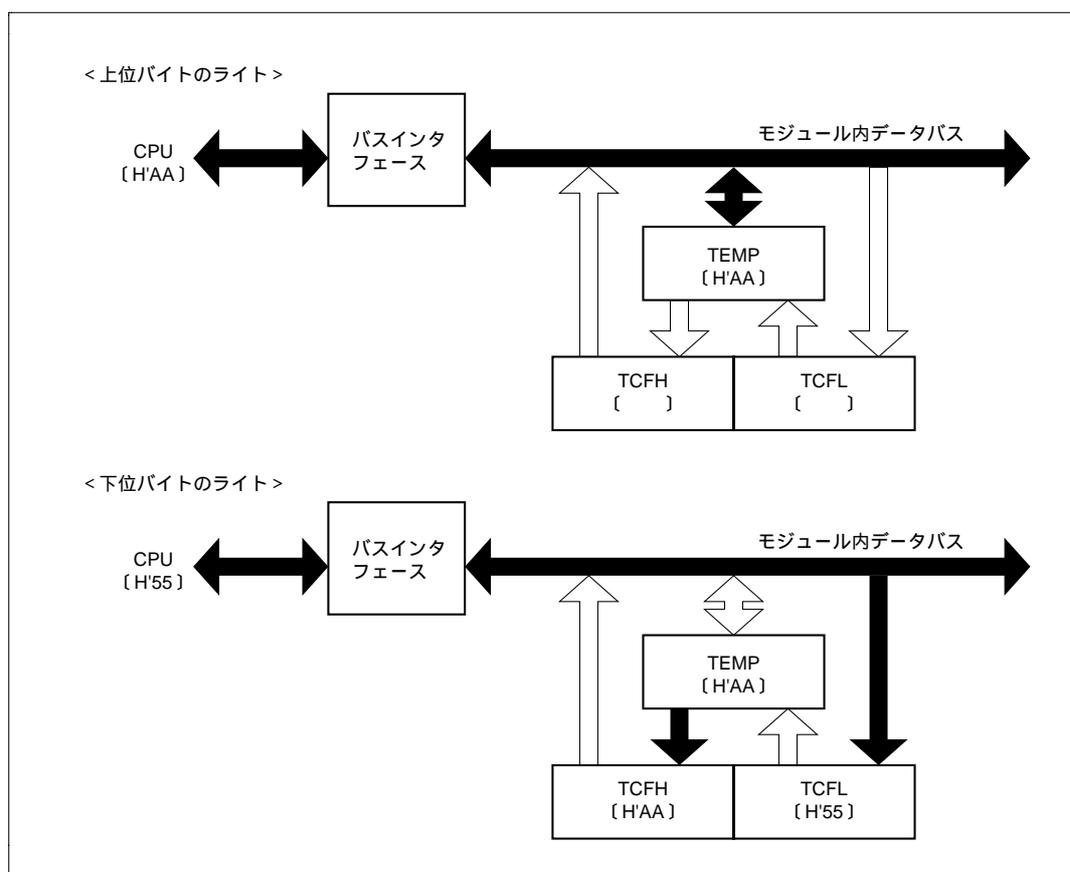


図 9.4 TCF のライト動作 (CPU TCF)

(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 9.5 に示します。

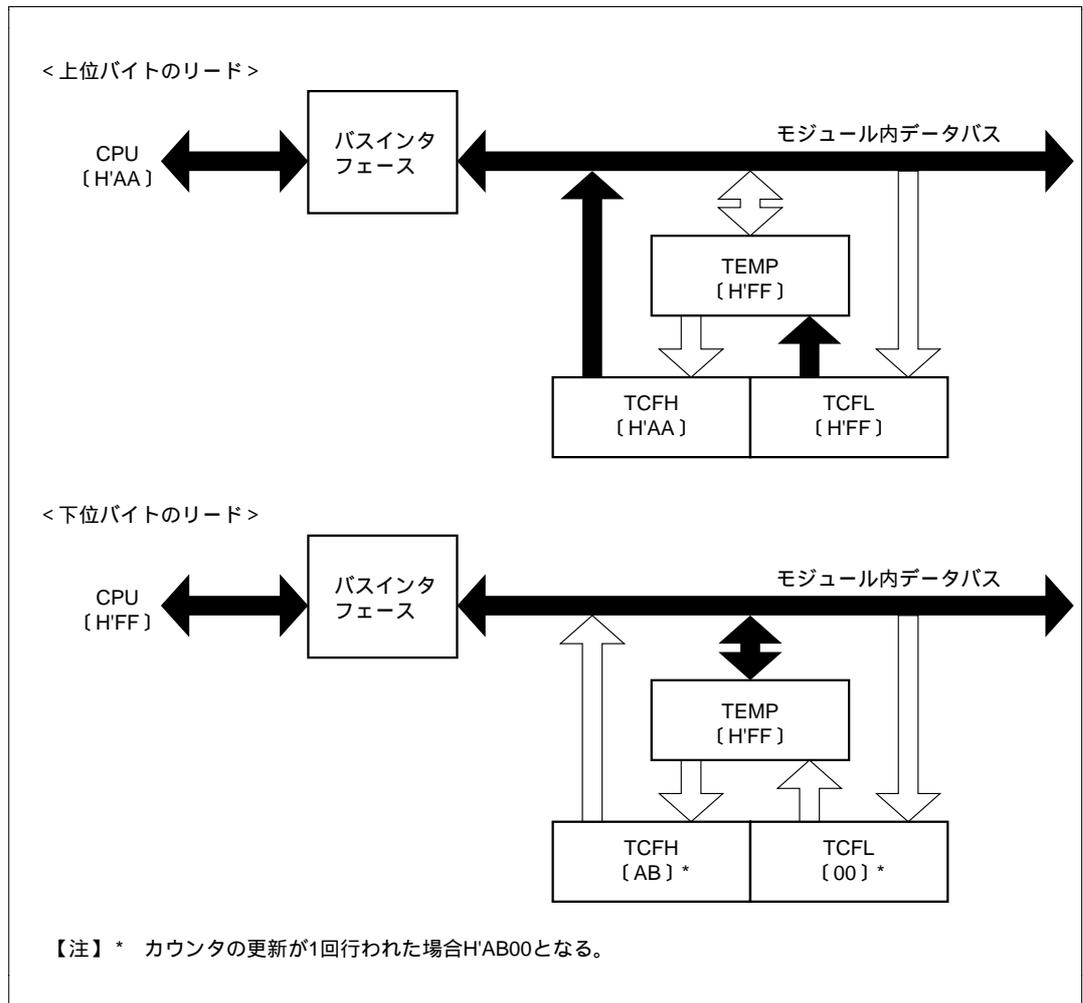


図 9.5 TCF のリード動作 (TCF CPU)

9.4.4 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/CCLRL 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 (または w) を分周して作られる 4 種類の内部クロック ($/32$ 、 $/16$ 、 $/4$ 、 $w/4$) 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 9.6 に出力タイミングを示します。

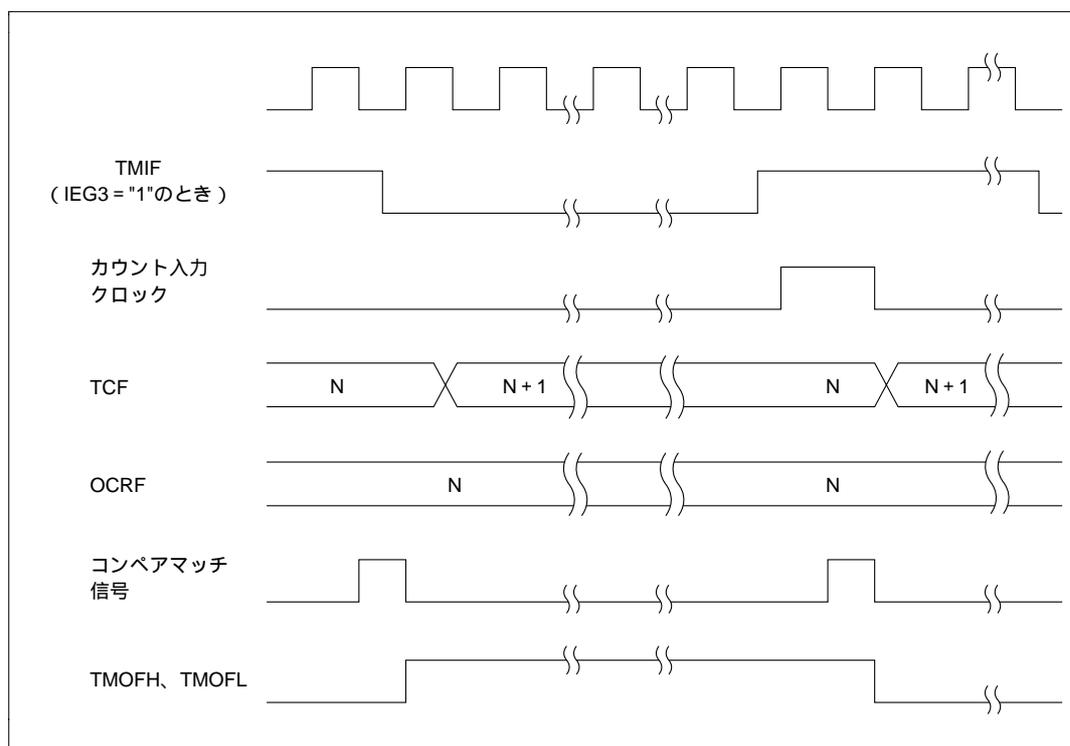


図 9.6 TMOFH、TMOFL 出力タイミング

(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 9.10 に示します。

表 9.10 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	モジュール スタンバイ
TCF	リセット	動作	動作	動作 / 停止*	動作 / 停止*	動作 / 停止*	停止	停止
OCRf	リセット	動作	保持	保持	動作	保持	保持	保持
TCRF	リセット	動作	保持	保持	動作	保持	保持	保持
TCSRf	リセット	動作	保持	保持	動作	保持	保持	保持

【注】 * アクティブモード、スリープモード時に TCF の内部クロックとして $\omega/4$ 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 $1/\omega$ (s) の誤差が生じます。
サブアクティブモード、ウォッチモード、サブスリープモード時にカウンタを動作させる場合には、必ず内部クロックとして $\omega/4$ を選択してください。その他の内部クロックを選択した場合、カウンタは動作しません。

9.4.5 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 ビットタイマモード

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。

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。

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ビットタイマモード

(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。

9.5 タイマ G

9.5.1 概要

タイマ G は、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/立下がりエッジのそれぞれ専用のインプットキャプチャ機能をもつ 8 ビットタイマです。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。また、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機能します。

(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 ($/64$ 、 $/32$ 、 $/2$ 、 $w/4$) 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 $\times 1$ 要因、オーバフロー $\times 1$ 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。

ウォッチモード、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 $w/4$ を選択した場合、ウォッチモード、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。

モジュールスタンバイモードにより、未使用時はモジュール単体でスタンバイモードに設定可能。

(2) ブロック図

タイマGのブロック図を図9.7に示します。

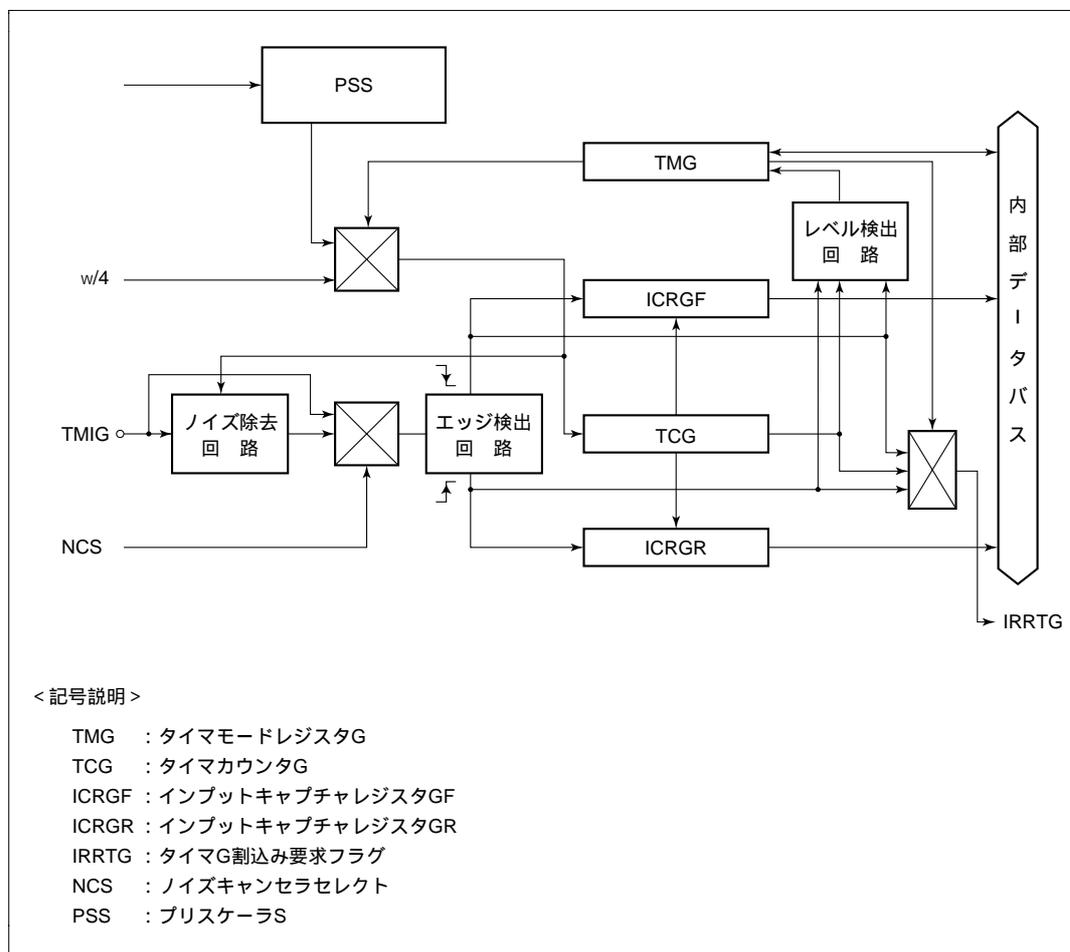


図9.7 タイマGのブロック図

(3) 端子構成

タイマGの端子構成を表9.11に示します。

表9.11 端子構成

名称	略称	入出力	機能
インพุットキャプチャ 入力	TMIG	入力	インพุ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9.12に示します。

表 9.12 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
クロック停止レジスタ1	CKSTPR1	R/W	H'FF	H'FFFA

9.5.2 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。

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พุ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พุ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พุットキャプチャ動作を確実にを行うために、インพุ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}（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พุ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พุ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พุットキャプチャ動作を確実にを行うために、インพุ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}（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	説明
OVFH	
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	説明
OVFL	
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上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下上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上上がり / 下上がり / 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下上がりエッジにより TCG をクリア	
1	0	インプットキャプチャ入力信号の上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： $f_w/4$ でカウント	

(5) クロック停止レジスタ 1 (CKSTPR1)

ビット:	7	6	5	4	3	2	1	0
	S1CKSTP	S31CKSTP	S32CKSTP	ADCKSTP	TGCKSTP	TFCKSTP	TCCKSTP	TACKSTP
初期値:	1	1	1	1	1	1	1	1
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CKSTPR1 は、8 ビットのリード/ライト可能なレジスタで、モジュールのモジュールスタンバイモードの制御を行います。本章ではタイマ G に関するビットのみ説明します。他のビットについては各モジュールの章を参照してください。

ビット 3 : タイマ G モジュールスタンバイモード制御 (TGCKSTP)

タイマ G をモジュールスタンバイモードに設定および解除を制御します。

TGCKSTP	説明
0	タイマ G はモジュールスタンバイモードに設定される
1	タイマ G のモジュールスタンバイモードは解除される (初期値)

9.5.3 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、PMR3のNCS*により設定します。

図9.8にノイズ除去回路のブロック図を示します。

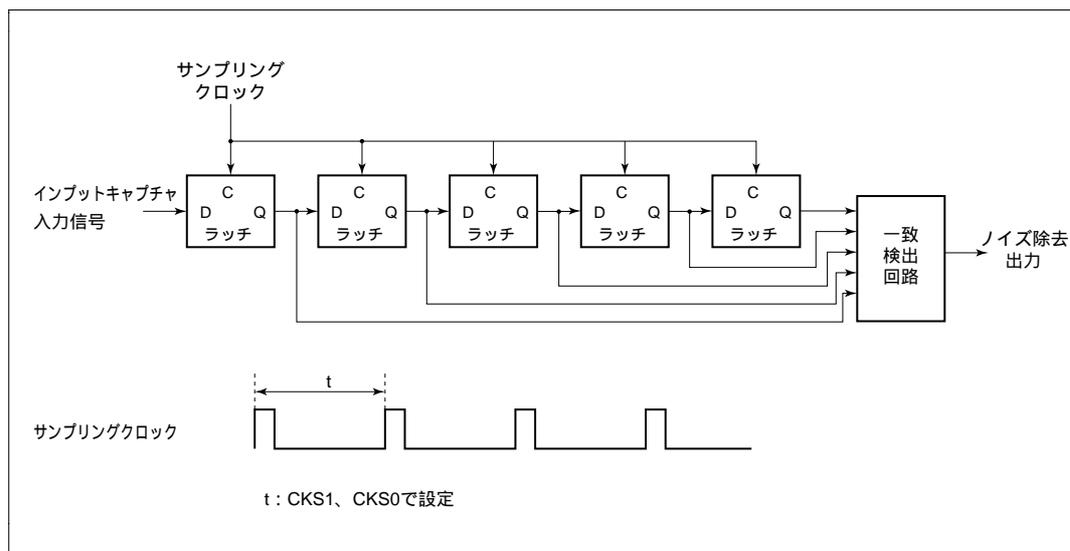


図 9.8 ノイズ除去回路のブロック図

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サンプリングされ、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 9.9 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。

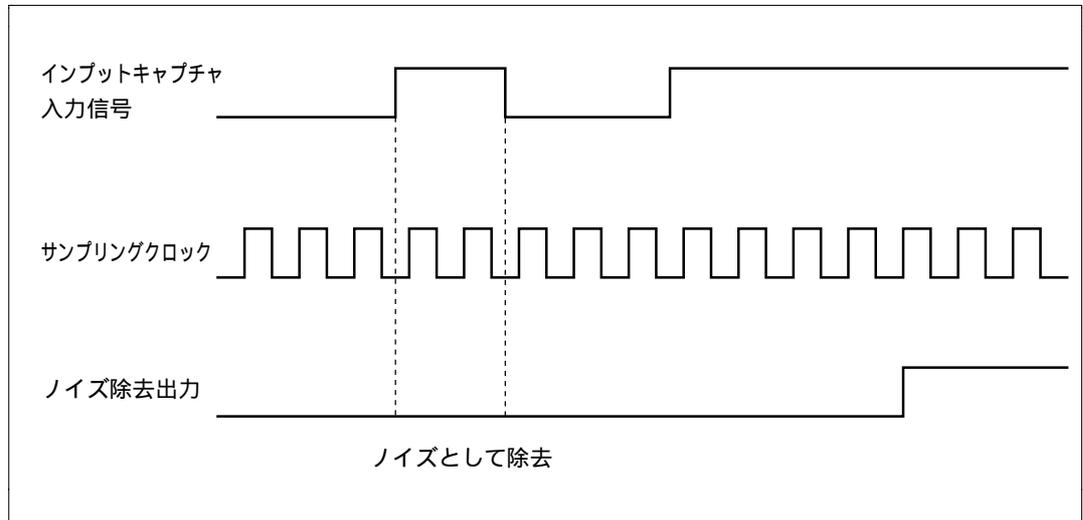


図 9.9 ノイズ除去回路タイミング (例)

9.5.4 動作説明

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。

(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。

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「9.5.3 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()とウォッチクロック(w)を分周して作られる4種類の内部クロック(/64、 /32、 /2、 $w/4$)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 9.10 に示します。

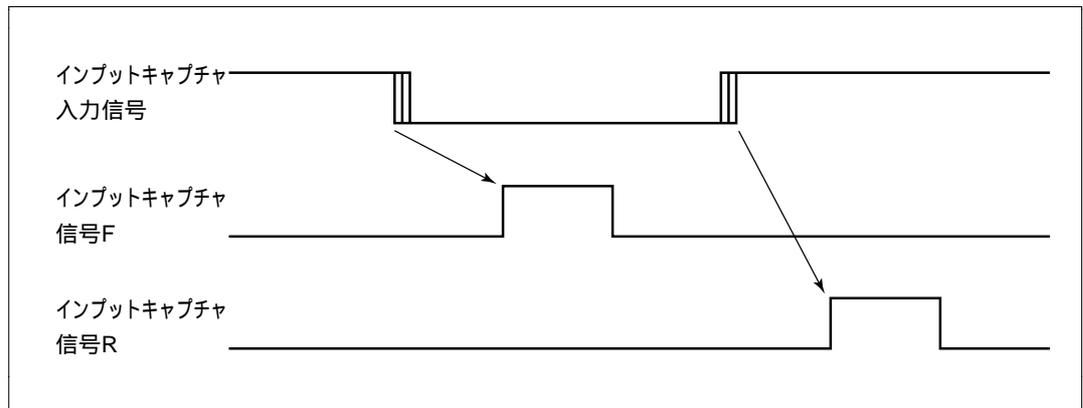


図 9.10 インพุットキャプチャ入力のタイミング (ノイズ除去機能なし)

(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 9.11 に示します。

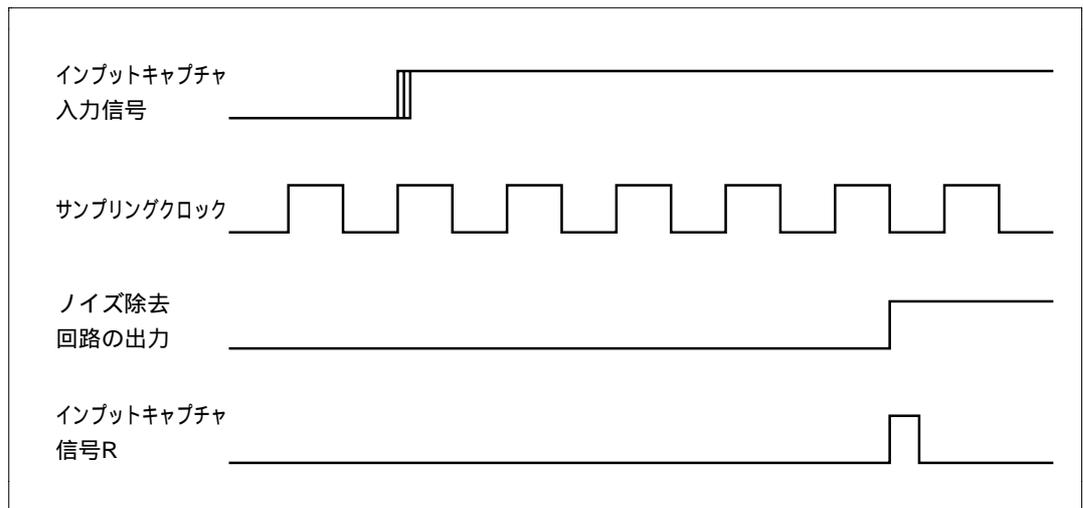


図 9.11 インพุットキャプチャ入力のタイミング (ノイズ除去機能あり)

(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 9.12 に示します。

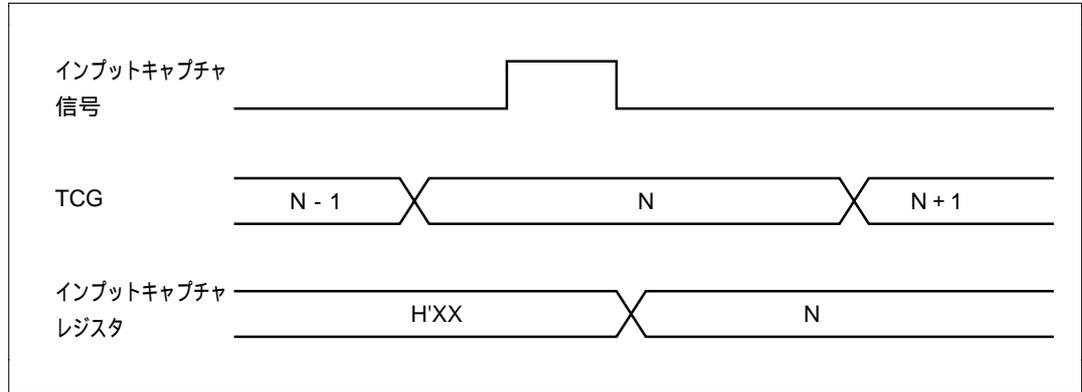


図 9.12 インพุットキャプチャ入力によるインพุットキャプチャタイミング

(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 9.13 に示します。

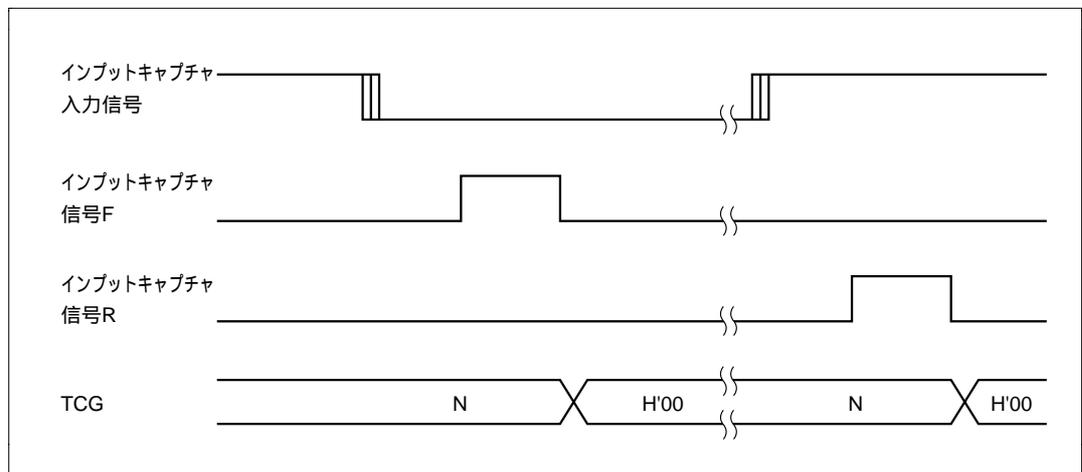


図 9.13 TCG のクリアタイミング

(6) タイマ G の動作モード

タイマ G の動作モードを表 9.13 に示します。

表 9.13 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	モジュール スタンバイ
TCG	インプット キャプチャ	リセット	動作*	動作*	動作 / 停止*	動作 / 停止*	動作 / 停止*	停止	停止
	インターバル	リセット	動作*	動作*	動作 / 停止*	動作 / 停止*	動作 / 停止*	停止	停止
ICRGF		リセット	動作*	動作*	動作 / 停止*	動作 / 停止*	動作 / 停止*	保持	保持
ICRGR		リセット	動作*	動作*	動作 / 停止*	動作 / 停止*	動作 / 停止*	保持	保持
TMG		リセット	動作	保持	保持	動作	保持	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 $w/4$ 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。ウォッチモード時に、TCG の内部クロックとして $w/4$ 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 $w/4$ 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。

サブアクティブモード、サブスリープモード時に、タイマ G を動作させるには、TCG の内部クロックとして $w/4$ を選択し、サブクロック $_{SUB}$ は $w/2$ を選択してください。その他の内部クロックを選択した場合やサブクロック $_{SUB}$ として $w/8$ 、 $w/4$ を選択した場合は、TCG、ノイズ除去回路は動作しませんので注意してください。

9.5.5 使用上の注意事項

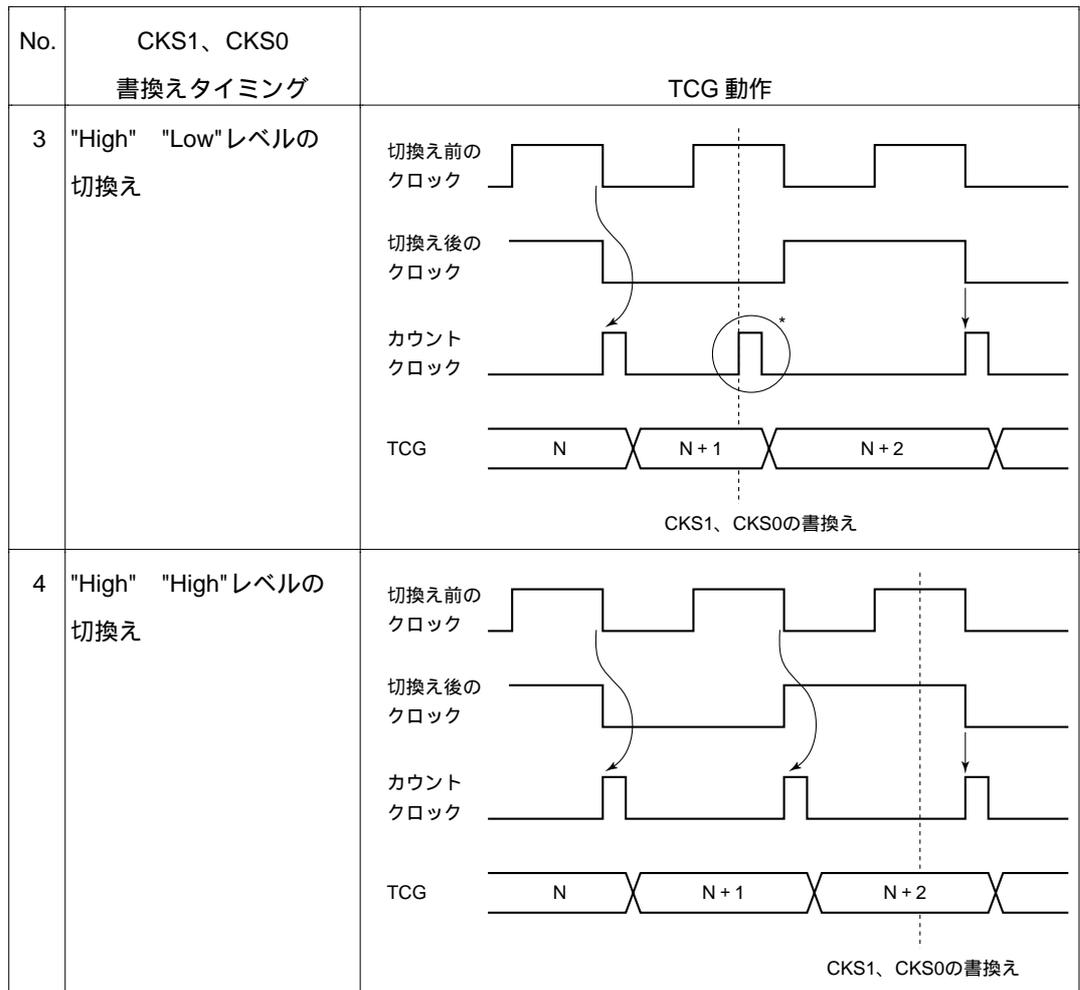
(1) 内部クロックの切換えと TCG 動作

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 9.14 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 9.14 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 9.14 内部クロックの切換えと TCG 動作

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>



【注】 * 切換のタイミングを立下がり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。

- ・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 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 9.15 に示します。

表 9.15 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】 P1₃端子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。

- ・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

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 3 (PMR3) の 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 9.16 に示します。

表 9.16 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"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 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。ポートモードレジスタの操作と割込み要求フラグのクリア手順を図 9.14 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込み要求フラグのセットを回避する方法として表 9.15、表 9.16 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。

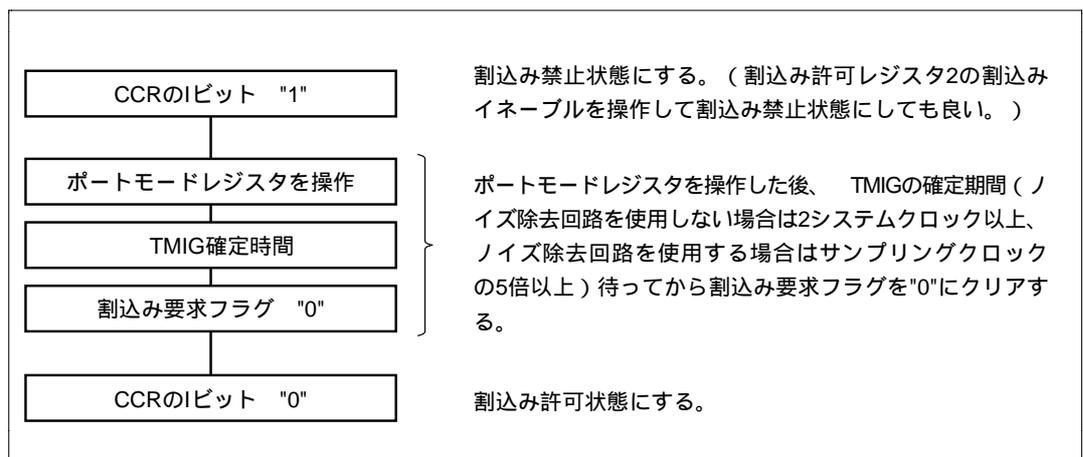


図 9.14 ポートモードレジスタの操作と割込み要求フラグのクリア手順

9.5.6 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 9.15 に示します。

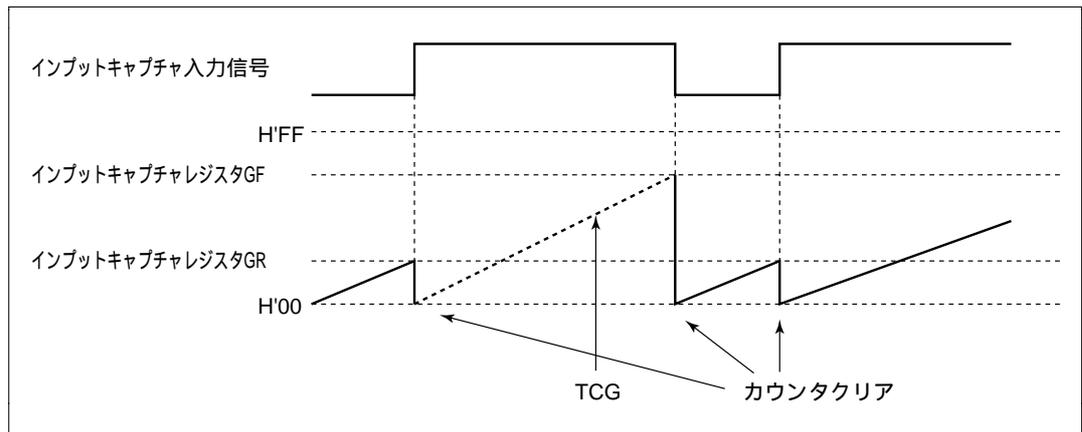


図 9.15 タイマ G の使用例

9.6 ウォッチドッグタイマ

9.6.1 概要

ウォッチドッグタイマ (Watchdog Timer) は、入力クロックが入るたびに、カウンタアップする 8 ビットのカウンタを備えており、システムの暴走などによりカウンタの値が書換えられずオーバーフローすると、LSI 内部をリセットできる機能を備えています。

(1) 特長

ウォッチドッグタイマの特長を以下に示します。

- 内部クロック $\phi_w/8192$ または $\phi_w/32$ でカウントアップ
- カウンタのオーバーフローでリセット信号を発生
- オーバーフロー周期は、 $8192/\phi_w$ または $32/\phi_w$ の 1 倍から 256 倍まで設定可能です。
($\phi_w = 2.00\text{MHz}$ のとき、約 $4\text{ms} \sim 1000\text{ms}$)
- モジュールスタンバイモードにより、未使用時はモジュール単体でスタンバイモードに設定可能

(2) ブロック図

ウォッチドッグタイマのブロック図を図 9.16 に示します。

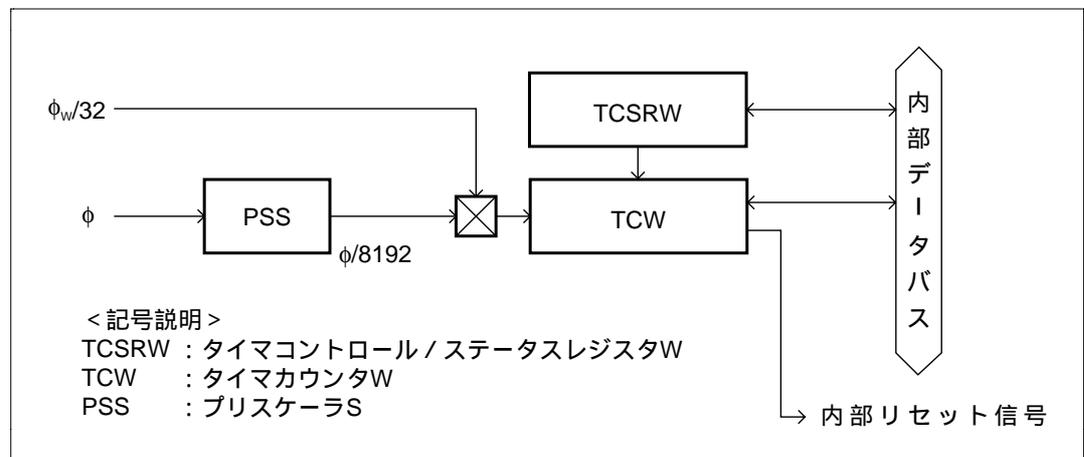


図 9.16 ウォッチドッグタイマのブロック図

(3) レジスタ構成

ウォッチドッグタイマのレジスタ構成を表 9.17 に示します。

表 9.17 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/ステータスレジスタ W	TCSRW	R/W	H'AA	H'FFB2
タイマカウンタ W	TCW	R/W	H'00	H'FFB3
クロック停止レジスタ 2	CKSTP2	R/W	H'FF	H'FFFB
ポートモードレジスタ 3	PMR3	R/W	H'00	H'FFCA

9.6.2 各レジスタの説明

(1) タイマコントロール/ステータスレジスタ W (TCSRW)

ビット:	7	6	5	4	3	2	1	0
	B6WI	TCWE	B4WI	TCSRWE	B2WI	WDON	BOWI	WRST
初期値:	1	0	1	0	1	0	1	0
R/W :	R	R(W)*	R	R(W)*	R	R(W)*	R	R(W)*

【注】 * 書込みの条件が成立している場合にのみ、書込み可能となります。書込み条件については各ビットの説明を参照してください。

TCSRW は、8 ビットのリード/ライト可能なレジスタで TCSRW、TCW の書込み制御、ウォッチドッグタイマの動作制御、動作状態を示すレジスタです。

ビット 7: ビット 6 書込み禁止 (B6WI)

TCSRW のビット 6 へのデータ書込みを制御します。

ビット 7	説明
B6WI	
0	ビット 6 への書込みを許可
1	ビット 6 への書込みを禁止 (初期値)

本ビットはリードすると常に"1"が読み出されます。ライトしても、データは格納されません。

ビット 6: タイマカウンタ W 書込み許可 (TCWE)

TCW へのビットデータの書込みを制御します。

ビット 6	説明
TCWE	
0	TCW への 8 ビットデータへの書込みを禁止 (初期値)
1	TCW への 8 ビットデータへの書込みを許可

ビット5：ビット4 書込み禁止 (B4WI)

TCSRWのビット4へのデータ書込みを制御します。

ビット5	説明
B4WI	
0	ビット4への書込みを許可
1	ビット4への書込みを禁止 (初期値)

本ビットはリードすると常に"1"が読み出されます。ライトしても、データは格納されません。

ビット4：タイマコントロール/ステータスレジスタW書込み許可 (TCSRWE)

TCSRWのビット2およびビット0へのデータの書込みを制御します。

ビット4	説明
TCSRWE	
0	ビット2およびビット0への書込みを禁止 (初期値)
1	ビット2およびビット0への書込みを許可

ビット3：ビット2 書込み禁止 (B2WI)

TCSRWのビット2へのデータの書込みを制御します。

ビット3	説明
B2WI	
0	ビット2への書込みを許可
1	ビット2への書込みを禁止 (初期値)

本ビットはリードすると常に"1"が読み出されます。ライトしてもデータは格納されません。

ビット2：ウォッチドッグタイマオン (WDON)

ウォッチドッグタイマの動作を許可します。

ビット2	説明
WDON	
0	ウォッチドッグタイマの動作を禁止 (初期値) [クリア条件] リセット、またはTCSRWE="1"の状態ではB2WIに"0"をライトしながらWDONに"0"をライトしたとき
1	ウォッチドッグタイマの動作を許可 [セット条件] TCSRWE="1"の状態ではB2WIに"0"をライトしながらWDONに"1"をライトしたとき

本ビットを"1"にセットすると、カウントアップを開始します。また、本ビットを"0"にクリアすると、カウントアップを停止します。

ビット1：ビット0 書込み禁止 (B0WI)

タイマコントロール/ステータスレジスタ W のビット 0 へのデータの書込みを制御します。

ビット1	
B0WI	説明
0	ビット0への書込みを許可
1	ビット0への書込みを禁止 (初期値)

本ビットはリードすると常に"1"が読み出されます。ライトしても、データは格納されません。

ビット0：ウォッチドッグタイマリセット (WRST)

TCW がオーバーフローし、内部リセット信号が発生したことを示します。オーバーフローしたことにより発生した内部リセット信号は、LSI全体をリセットします。

WRST は、 $\overline{\text{RES}}$ 端子によるリセット、またはソフトウェアによる"0"ライトによりクリアされます。

ビット0	
WRST	説明
0	[クリア条件] (1) $\overline{\text{RES}}$ 端子によるリセット (2) TCSRWE = "1"の状態でも B0WI に"0"をライトしながら WRST に"0"をライトしたとき
1	[セット条件] TCW がオーバーフローし、内部リセット信号が発生したとき

(2) タイマカウンタ W (TCW)

ビット:	7	6	5	4	3	2	1	0
	TCW7	TCW6	TCW5	TCW4	TCW3	TCW2	TCW1	TCW0
初期値:	0	0	0	0	0	0	0	0
R/W :	R/W							

TCW は、8 ビットのリード/ライト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、 $f_{clk}/8192$ または $f_{clk}/32$ です。TCW の値は CPU から常にリード/ライトできます。

TCW がオーバーフロー (H'FF→H'00) すると、内部リセット信号が発生し、TCSRW の WRST が"1"にセットされます。リセット時、TCW は H'00 にイニシャライズされます。

(3) クロック停止レジスタ 2 (CKSTPR2)

ビット:	7	6	5	4	3	2	1	0
					AECKSTP	WDCKSTP	PWCKSTP	LDCKSTP
初期値:	1	1	1	1	1	1	1	1
R/W :					R/W	R/W	R/W	R/W

CKSTPR2 は、8 ビットのリード/ライト可能なレジスタで、モジュールのモジュールスタンバイモードの制御を行います。本章ではウォッチドッグタイマに関するビットのみ説明します。他のビットについては各モジュールの章を参照してください。

ビット 2 : ウォッチドッグタイマモジュールスタンバイモード制御 (WDCKSTP)

ウォッチドッグタイマをモジュールスタンバイモードに設定および解除を制御します。

WDCKSTP	説明
0	ウォッチドッグタイマはモジュールスタンバイモードに設定される
1	ウォッチドッグタイマのモジュールスタンバイモードは解除される (初期値)

【注】 WDCKSTP はタイマコントロール/ステータスレジスタ W (TCSRW) の WDON が 0 のときに有効になります。WDON が 1 (ウォッチドッグタイマ動作中) のとき WDCKSTP を 0 に設定すると WDCKSTP は 0 に設定されませんが、ウォッチドッグタイマはモジュールスタンバイモードには入らずウォッチドッグ機能を継続いたします。ウォッチドッグ機能が終了し、ソフトウェアで WDON を 0 に設定すると同時に WDCKSTP が有効になり、ウォッチドッグタイマはモジュールスタンバイモードに入ります。

(4) ポートモードレジスタ 3 (PMR3)

ビット:	7	6	5	4	3	2	1	0
	AEVL	AEVH	WDCKS	NCS	IRQ0	RESO	UD	PWM
初期値:	0	0	0	0	0	1	0	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PMR3 は 8 ビットのリード/ライト可能なレジスタで、主にポート 3 の各端子機能の切換えを制御します。本章ではウォッチドッグタイマに関するビットのみ説明します。他のビットについては「第 8 章 I/O ポート」を参照してください。

ビット 5 : ウォッチドッグタイマソースクロック選択 (WDCKS)

WDCKS	説明
0	/8192 を選択 (初期値)
1	$\omega/32$ を選択

9.6.3 動作説明

ウォッチドッグタイマは、入力クロック ($f_{clk}/8192$ または $f_{clk}/32$) が入るたびにカウントアップする 8 ビットのカウンタを備えています。入力クロックの選択は、ポートモードレジスタ 3 (PMR3) の WDCKS で行います。WDCKS が 0 のとき $f_{clk}/8192$ が選択され、1 のとき $f_{clk}/32$ が選択されます。TCSRW の TCSRWE="1" の状態で B2WI に "0" をライトしながら WDON に "1" をライトすると、TCW はカウントアップを開始します。TCW のカウント値が、H'FF になった後、クロックが入力されると、ウォッチドッグタイマはオーバフローし、基準クロック (f_{osc} または f_{sub}) の 1 クロック分後に内部リセット信号を発生します。内部リセット信号は f_{osc} クロックで 512 クロック分の時間出力されます。TCW はライト可能なカウンタですので、TCW に値を設定すると、その値からカウントアップを行います。したがって、TCW の値により、オーバフロー周期を 1~256 入力クロックの範囲で設定できます。

ウォッチドッグタイマ動作例を図 9.17 に示します。

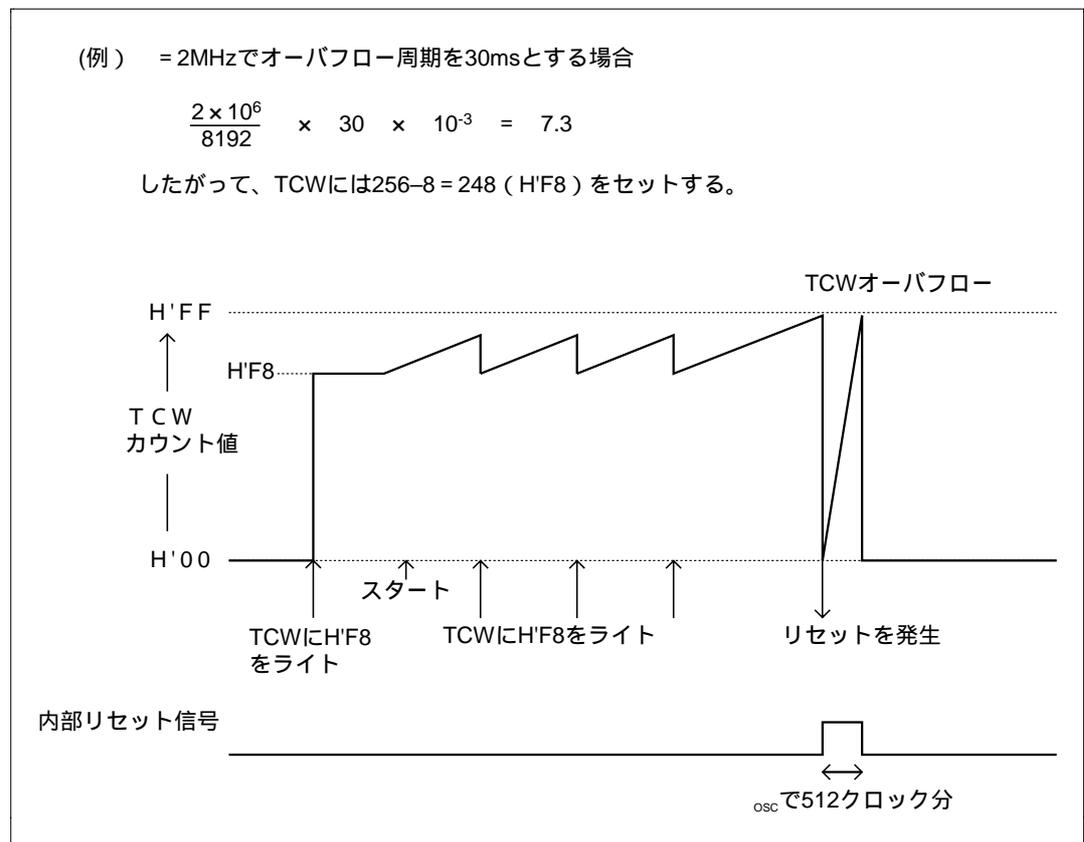


図 9.17 ウォッチドッグタイマの動作例

9.6.4 ウォッチドッグタイマの動作モード

ウォッチドッグタイマの動作モードを表9.18に示します

表9.18 ウォッチドッグタイマの動作モード

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	モジュール スタンバイ
TCW	リセット	動作	動作	停止	動作 / 停止*	停止	停止	停止
TCSRW	リセット	動作	動作	保持	動作 / 停止*	保持	保持	保持

【注】 * 入力クロックに $\omega/32$ を選択した場合に動作します。

9.7 非同期イベントカウンタ (AEC)

9.7.1 概要

非同期イベントカウンタ (Asynchronous Event Counter) は、外部イベントクロックが入るたびにカウントアップするイベントカウンタです。

(1) 特長

非同期イベントカウンタの特長を以下に示します。

非同期イベントをカウント可能

基本クロック、 SUB の動作とは無関係に非同期に入力される外部イベントをカウント可能です。

カウンタは 16 ビット構成になっており、65536 (2^{16}) 回以内のイベントのカウントが可能です。

2チャンネルの独立した 8 ビットイベントカウンタとしても使用可能。

ソフトウェアによってカウンタのリセット、カウントアップ機能の停止が制御可能。

イベントカウンタのオーバフローを検出し、自動的に割込みを発生。

モジュールスタンバイモードにより、未使用時はモジュール単体でスタンバイモードに設定可能。

(2) ブロック図

非同期イベントカウンタのブロック図を図 9.18 に示します。

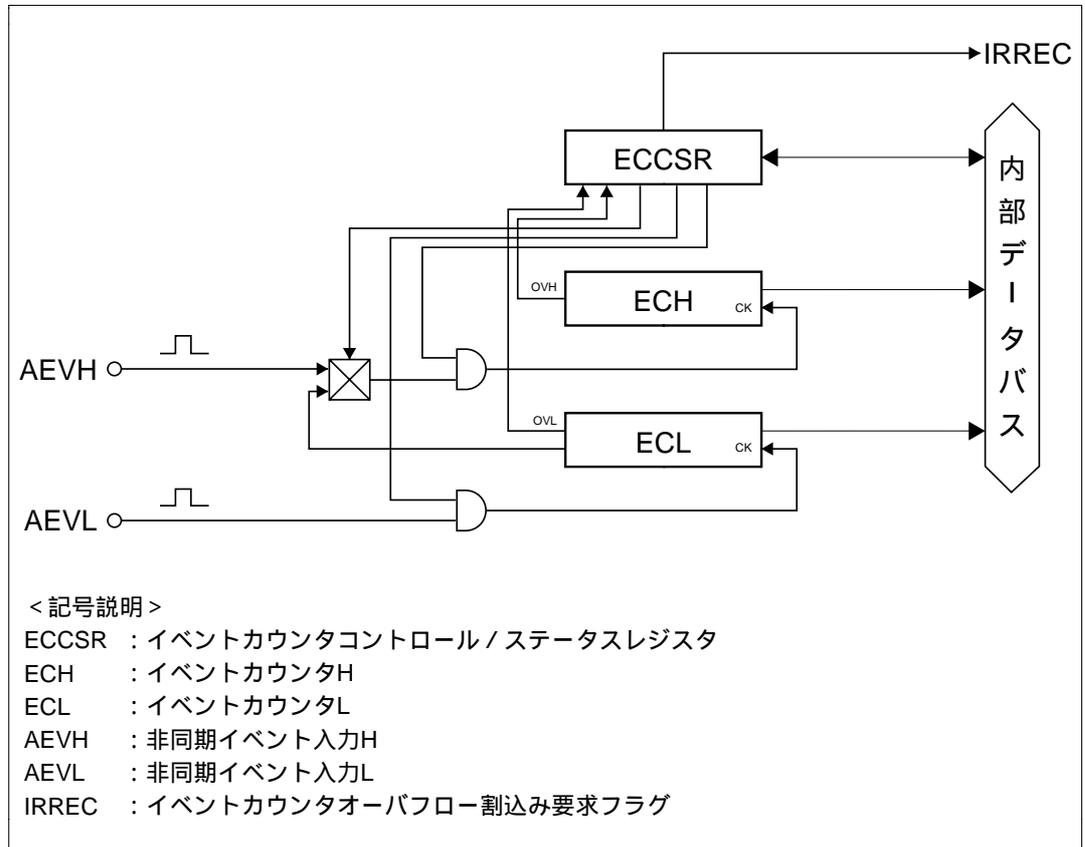


図 9.18 非同期イベントカウンタブロック図

(3) 端子構造

非同期イベントカウンタの端子構成を表 9.19 に示します。

表 9.19 端子構成

名称	略称	入出力	機能
非同期イベント入力H	AEVH	入力	イベントカウンタHに入力するイベント入力端子
非同期イベント入力L	AEVL	入力	イベントカウンタLに入力するイベント入力端子

(4) レジスタ構成

非同期イベントカウンタのレジスタ構成を表 9.20 に示します。

表 9.20 レジスタ構成

名称	略称	R/W	初期値	アドレス
イベントカウンタコントロール/ステータスレジスタ	ECCSR	R/W	H'00	H'FF95
イベントカウンタH	ECH	R	H'00	H'FF96
イベントカウンタL	ECL	R	H'00	H'FF97
クロック停止レジスタ2	CKSTP2	R/W	H'FF	H'FFFB

9.7.2 各レジスタの構成

(1) イベントカウンタコントロール/ステータスレジスタ (ECCSR)

ビット:	7	6	5	4	3	2	1	0
	OVH	OVL	—	CH2	CUEH	CUEL	CRCH	CRCL
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"0"ライトのみ可能です。

ECCSR は、8 ビットのリード/ライト可能なレジスタで、カウンタのオーバフローの検出、カウンタのリセット、カウントアップ機能の停止の制御を行います。

リセット時、ECCSR は H'00 にイニシャライズされます。

ビット7: カウンタオーバフローH (OVH)

ECH 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。ECH がオーバフローすると、本フラグがセットされます。ソフトウェアでセットすることはできません。本フラグのクリアはソフトウェアで行います。OVH が"1"の状態では OVH をリードした後、OVH に"0"をライトしたときにクリアされます。

CH2 を"0"にした状態で ECH、ECL を 16 ビットのイベントカウンタとして使用している場合には、本フラグは 16 ビットのイベントカウンタがオーバフロー (H'FFFF→H'0000) したことを示すステータスフラグです。

ビット7	説明
OVH	
0	ECH がオーバフローしていないことを示します (初期値) [クリア条件] OVH が"1"の状態では OVH をリードした後、OVH に"0"をライトしたとき
1	ECH がオーバフローしたことを示します [セット条件] ECH の値が H'FF→H'00 になったとき

ビット6：カウンタオーバーフローL (OVL)

ECL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。ECLがオーバーフローすると、本フラグがセットされます。ソフトウェアでセットすることはできません。本フラグのクリアはソフトウェアで行います。OVLが"1"の状態でもOVLをリードした後、OVLに"0"にライトしたときにクリアされます。

ビット6	
OVL	説明
0	ECLがオーバーフローしていないことを示します [クリア条件] OVLが"1"の状態でもOVLをリードした後、OVLに"0"をライトしたとき (初期値)
1	ECLがオーバーフローしたことを示します [セット条件] CH2を"1"にした状態でECLの値がH'FF→H'00になったとき

ビット5：リザーブビット

本ビットはリード/ライト可能なリザーブビットです。リセット時は"0"にイニシャライズされます。

ビット4：チャンネル選択 (CH2)

ECH、ECLを1チャンネルの16ビットのイベントカウンタとして使用するか、2チャンネルの独立した8ビットのイベントカウンタとして使用するかを選択します。CH2を"0"にクリアすると、ECH、ECLは16ビットのイベントカウンタとなり、非同期イベント入力としてAEVL端子にイベントクロックが入力されるたびにカウントアップします。このときECHの入力クロックはECLからのオーバーフロー信号が選択されます。CH2を"1"にセットすると、ECH、ECLは各々独立した8ビットのイベントカウンタとなり、非同期イベント入力として各々AEVH端子、AEVL端子にイベントクロックが入力されるたびにカウントアップします。

ビット4	
CH2	説明
0	ECH、ECLを連結した1チャンネルの16ビットイベントカウンタとして使用します (初期値)
1	ECH、ECLを独立した2チャンネルの8ビットイベントカウンタとして使用します

ビット3：カウントアップイネーブルH (CUEH)

ECH に入力されるイベントクロックの入力を許可します。本ビットに"1"をライトするとイベントクロックの入力を許可し、カウントアップをします。本ビットに"0"をライトすると、イベントクロックの入力を禁止し、ECH の値は保持されます。イベントクロックはCH2 によってAEVH 端子またはECLからのオーバフロー信号のいずれかを選択可能です。

ビット3		
CUEH	説明	
0	ECH のイベントクロックの入力を禁止 ECH の値を保持	(初期値)
1	ECH のイベントクロックの入力を許可	

ビット2：カウントアップイネーブルL (CUEL)

ECL に入力されるイベントクロックの入力を許可します。本ビットに"1"をライトするとイベントクロックの入力を許可し、カウントアップをします。本ビットに"0"をライトすると、イベントクロックの入力を禁止し、ECL の値は保持されます。

ビット2		
CUEL	説明	
0	ECL のイベントクロックの入力を禁止 ECL の値を保持	(初期値)
1	ECL のイベントクロックの入力を許可	

ビット1：カウンタリセット制御H (CRCH)

ECH のリセットを制御します。本ビットが"0"のとき ECH はリセットされます。本ビットに"1"をライトするとカウンタのリセットを解除し、ECH のカウントアップ機能を許可します。

ビット1		
CRCH	説明	
0	ECH をリセット	(初期値)
1	ECH のリセットを解除しカウントアップ機能を許可	

ビット0：カウンタリセット制御L (CRCL)

ECLのリセットを制御します。本ビットが"0"のときECLはリセットされます。本ビットに"1"をライトするとカウンタのリセットを解除し、ECLのカウントアップ機能を許可します。

ビット1	説明	
CRCL		
0	ECLをリセット	(初期値)
1	ECLのリセットを解除しカウントアップ機能を許可	

(2) イベントカウンタH (ECH)

ビット:	7	6	5	4	3	2	1	0
	ECH7	ECH6	ECH5	ECH4	ECH3	ECH2	ECH1	ECH0
初期値:	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R

ECHは8ビットのリード可能なアップカウンタで、独立した8ビットのイベントカウンタとして、またはECLと組み合わせることで16ビットのイベントカウンタの上位8ビットのアップカウンタとして動作します。入力クロックはCH2により外部非同期イベントAEVH端子、または下位の8ビットのカウンタECLからのオーバフロー信号のいずれかを選択可能です。ECHはソフトウェアでH'00にクリア可能です。リセットは、H'00にイニシャライズされます。

(3) イベントカウンタL (ECL)

ECLは8ビットのリード可能なアップカウンタで、独立した8ビットのイベントカウンタとして、またはECHと組み合わせることで16ビットのイベントカウンタの下位8ビットのアップカウンタとして動作します。入力クロックは外部非同期イベントAEVL端子からのイベントクロックを使用します。ECLはソフトウェアでH'00にクリア可能です。リセットは、H'00にイニシャライズされます。

ビット:	7	6	5	4	3	2	1	0
	ECL7	ECL6	ECL5	ECL4	ECL3	ECL2	ECL1	ECL0
初期値:	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R

(4) クロック停止レジスタ 2 (CKSTPR)

ビット:	7	6	5	4	3	2	1	0
					AECKSTP	WDCKSTP	PWCKSTP	LDCKSTP
初期値:	1	1	1	1	1	1	1	1
R/W :					R/W	R/W	R/W	R/W

CKSTPR2 は、8 ビットのリード/ライト可能なレジスタで、モジュールのモジュールスタンバイモードの制御を行います。本章では非同期イベントカウンタに関するビットのみを説明します。他のビットについては各モジュールの章を参照してください。

ビット 3 : 非同期イベントカウンタモジュールスタンバイモード制御 (AECKSTP)

非同期イベントカウンタをモジュールスタンバイモードに設定および解除を制御します。

AECKSTP	説明
0	非同期イベントカウンタはモジュールスタンバイモードに設定される
1	非同期イベントカウンタのモジュールスタンバイモードは解除される (初期値)

9.7.3 動作説明

(1) 16ビットイベントカウンタの動作

ECCSR の CH2 を "0" にクリアすると、ECH、ECL は 16 ビットのイベントカウンタとして動作します。16 ビットイベントカウンタとして使用する場合のソフトウェアの例を図 9.19 に示します。

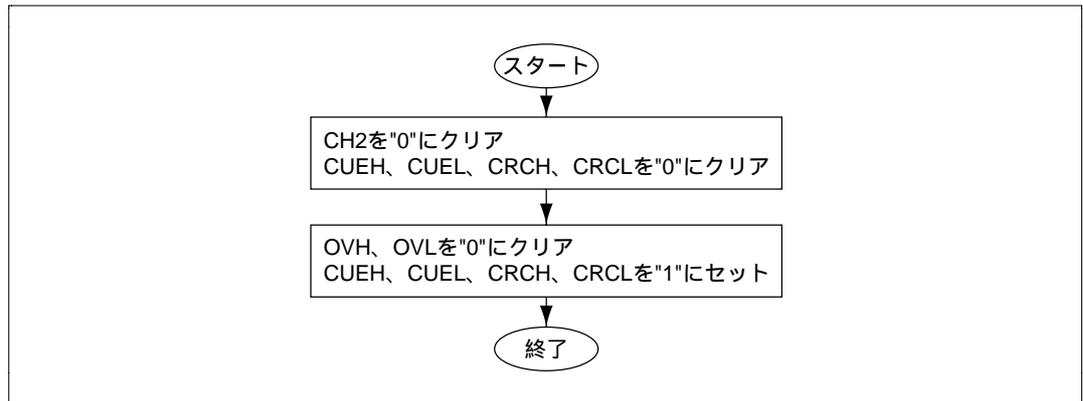


図 9.19 16 ビットイベントカウンタとして使用する場合のソフトウェアの例

リセット時、CH2 は "0" にクリアされるため、リセット後は ECH、ECL は 16 ビットイベントカウンタとして動作します。また、図 9.19 に示すソフトウェアの例で使用すると 16 ビットのイベントカウンタとして使用できます。動作クロックは AEVL 端子からの非同期イベント入力です。ECH、ECL のカウント値が共に H'FF になった後に、クロックが入力されると ECH、ECL はオーバフロー (H'FFFF ~ H'0000) し、ECCSR の OVH フラグが "1" にセットされ、ECH、ECL のカウント値は各々 H'00 に戻り、再びカウントアップを再開します。オーバフロー発生時には、IRR2 の IRREC が "1" にセットされます。このとき IENR2 の IENEC が "1" ならば CPU に割り込みを要求します。

(2) 8ビットイベントカウンタの動作

ECCSR の CH2 を "1" にセットすると、ECH、ECL は独立した 8 ビットのイベントカウンタとして動作します。8 ビットイベントカウンタとして使用する場合のソフトウェアの例を図 9.20 に示します。

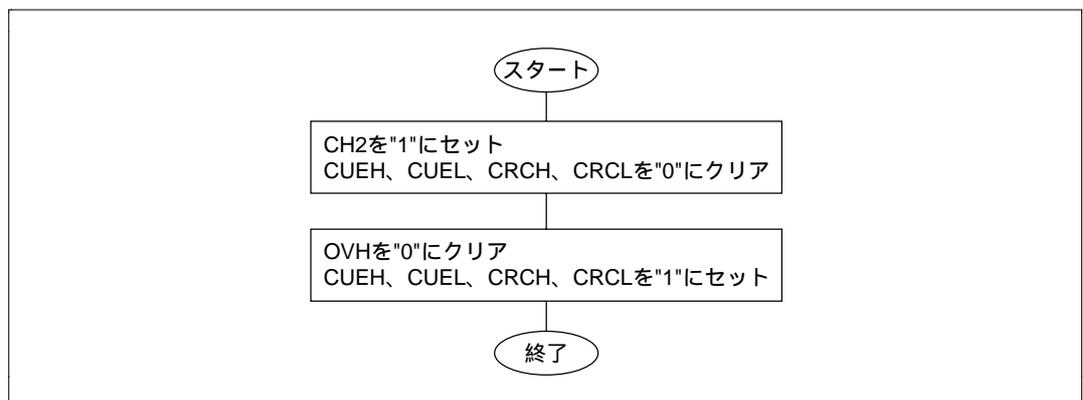


図 9.20 8 ビットイベントカウンタとして使用する場合のソフトウェアの例

図 9.20 に示すソフトウェアの例で使用すると独立した 8 ビットのイベントカウンタとして使用できます。8 ビットイベントカウンタの動作クロックは ECH が AEVH 端子からの非同期イベント入力、ECL が AEVL 端子からの非同期イベント入力です。ECH のカウント値が H'FF になった後にクロックが入力されると、ECH はオーバフローし ECCSR の OVH フラグが"1"にセットされ、ECH のカウント値は H'00 に戻り、再びカウントアップを再開します。また、ECL のカウント値が H'FF になった後にクロックが入力されると、ECL はオーバフローし ECCSR の OVL フラグが"1"にセットされ、ECL のカウント値は H'00 に戻り、再びカウントアップを再開します。オーバフロー発生時には、IRR2 の IRREC が"1"にセットされます。このとき IENR2 の IENEC が"1"ならば CPU に割込みを要求します。

9.7.4 非同期イベントカウンタの動作モード

非同期イベントカウンタの動作モードを表 9.21 に示します。

表 9.21 非同期イベントカウンタの動作モード

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	モジュール スタンバイ
ECCSR	リセット	動作	動作	保持*	動作	動作	保持*1	保持
ECH	リセット	動作	動作*	動作*	動作	動作	動作*1	停止
ECL	リセット	動作	動作*	動作*	動作	動作	動作*1	停止

【注】 * 非同期外部イベントが入力されるとカウンタはカウントアップしますが、カウンタオーバフローH/L フラグは影響を受けません。

9.7.5 使用上の注意事項

- (1) ECH、ECL の値をリードする場合には、リードする前に ECCSR の CUEH、CUEL を "0" にクリアして非同期イベント入力をカウンタに入力しないようにしてください。リードしている際にイベントカウンタがカウントアップすると正しい値がリードできません。なお、ECCSR の CUEH、CUEL の "0" クリアの際に ECH、ECL が 1 カウントカウントアップされることがあります。
- (2) AEVH、AEVL 端子に入力するクロックの周波数は最大で、内部電源降圧回路不使用時は $V_{CC} = 4.5 \sim 5.5V$ で最大 6MHz、 $V_{CC} = 3.0 \sim 5.5V$ で最大 4MHz、 $V_{CC} = 2.6 \sim 5.5V$ で最大 3.2MHz、内部電源降圧回路使用時、不使用時共 $V_{CC} = 2.2 \sim 5.5V$ で最大 2MHz それ以外では最大 1MHz までの範囲としてください。またクロックの High 幅、Low 幅は最小 83ns となるようにしてください。デューティ比はいくつでもかまいません。

モード	AEVL、AEVH 端子に入力する 最大クロック周波数
16 ビットモード	内部降圧回路不使用時
8 ビットモード アクティブ (高速)、スリープ (高速)	$V_{CC} = 4.5 \sim 5.5V/6MHz$ $V_{CC} = 3.0 \sim 5.5V/4MHz$ $V_{CC} = 2.6 \sim 5.5V/3.2MHz$ $V_{CC} = 2.2 \sim 5.5V/2MHz$ 上記以外/1MHz 内部降圧回路使用時 $V_{CC} = 2.2 \sim 5.5V/2MHz$ 上記以外/1MHz
8 ビットモード アクティブ (中速)、スリープ (中速)	$(/16)$ $2 \cdot f_{osc}$ $(/32)$ f_{osc} $(/64)$ $1/2 \cdot f_{osc}$ $f_{osc} = 400kHz \sim 4MHz$ $(/128)$ $1/4 \cdot f_{osc}$
8 ビットモード ウォッチ、サブアクティブ、サブスリープ、スタンバイ	$(w/2)$ 1000kHz $(w/4)$ 500kHz $w = 32.768kHz$ または $38.4kHz$ $(w/8)$ 250kHz

- (3) 16ビットモードで使用する際、ECCSRの設定はCUEHを"1"にセットしてからCRCHを"1"にセットするか、CUEHとCRCHを同時にセットしてからクロックを入力してください。その後16ビットモードで使用中はCUEHの値を変更しないでください。16ビットモード設定中にCUEHを変化させるとECHが誤カウントアップすることがあります。

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	271
10.2	SCI1	272
10.2.1	概要	272
10.2.2	各レジスタの説明	274
10.2.3	動作説明	280
10.2.4	SSBモード時の動作説明	283
10.2.5	割込み要因	285
10.2.6	使用上の注意事項	285
10.3	SCI3	286
10.3.1	概要	286
10.3.2	各レジスタの説明	290
10.3.3	動作説明	309
10.3.4	割込み要因	333
10.3.5	使用上の注意事項	334

10.1 概要

本 LSI は、3 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) を備えています。

3 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 ($f_{1024} \sim f_{16}$ 、 $f_w/4$) 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
SCI31、 SCI32	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。また、複数のICを制御することができるSSB (Synchronized Serial Bus) という通信方式を備えています。

(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、 $w/4$) 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

SSBモード時、HOLDモード/LATCHモードを選択可能

(2) ブロック図

SCI1のブロック図を図10.1に示します。

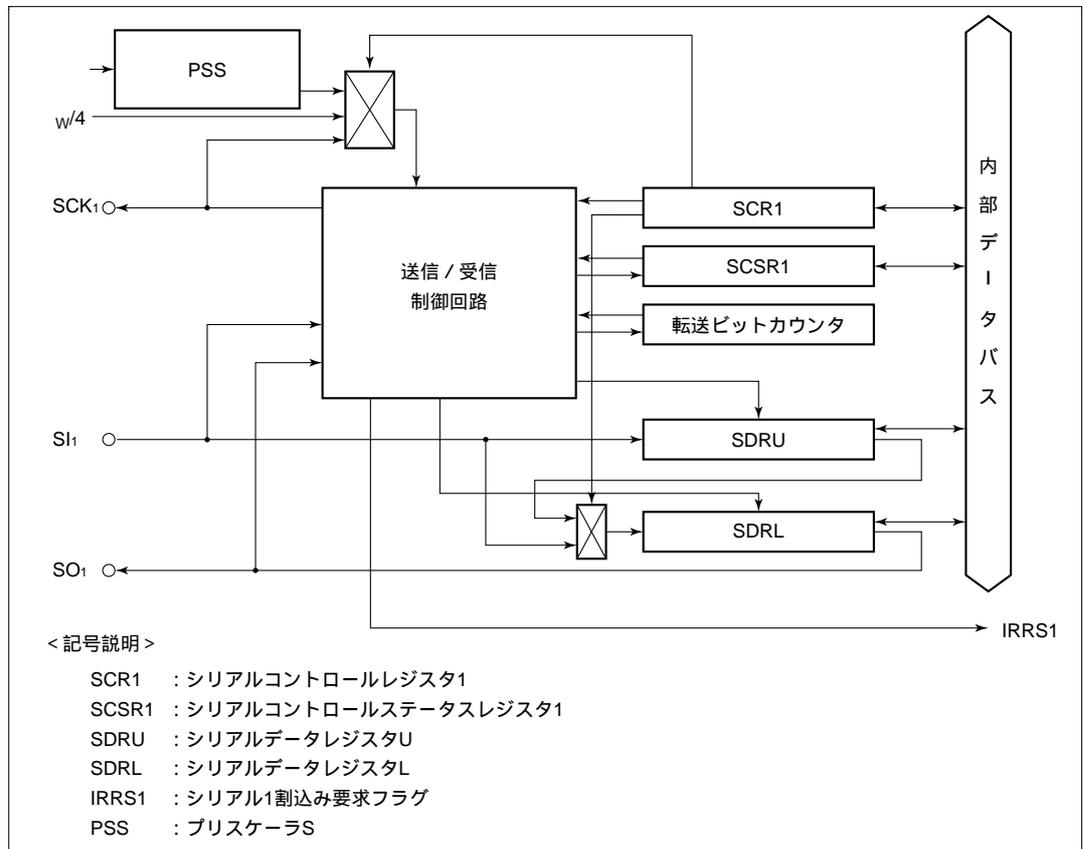


図10.1 SCI1のブロック図

(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 ₁	入出力	SCI1 のクロック入出力端子
SCI1 データ入力	SI ₁	入力	SCI1 の受信データ入力端子
SCI1 データ出力	SO ₁	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
クロック停止レジスタ 1	CKSTPR1	R/W	H'FF	H'FFFA

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	MRKON	LTCH	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 SI₁、SO₁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 : TAIL MARK 制御 (MRKON)

8 ビットまたは 16 ビットデータ転送後、TAIL MARK の出力を制御します。

ビット 5	説明
MRKON	
0	TAIL MARK を非出力 (クロック同期モード) (初期値)
1	TAIL MARK を出力 (SSB モード)

ビット4 : LATCH TAIL 制御 (LTCH)

MRKON = "1" (SSB モード) の時、TAIL MARK として、LATCH TAIL を出力するか、HOLD TAIL を出力するかを選択します。

ビット4		
LTCH	説明	
0	HOLD TAIL を出力	(初期値)
1	LATCH TAIL を出力	

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 SCK₁ 端子の入出力の設定を行います。

ビット3		
CKS3	説明	
0	クロックソースはプリスケラ S、SCK ₁ 端子は出力	(初期値)
1	クロックソースは外部クロック、SCK ₁ 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	プリスケラ分周比	転送クロック周期
CKS2	CKS1	CKS0		= 2.5MHz
0	0	0	/1024 (初期値)	409.6μs
0	0	1	/256	102.4μs
0	1	0	/64	25.6μs
0	1	1	/32	12.8μs
1	0	0	/16	6.4μs
1	0	1	/8	3.2μs
1	1	0	/4	1.6μs
1	1	1	$w/4$	122μs

(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	—	—	—	MTRF	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 SO₁ 端子の出力レベルを変更します。また、リードすると SO₁ 端子の出力レベルが読み出されます。送信完了後の SO₁ 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、SO₁ 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。

また、SSB モードでの設定も無効となります。

したがって送信完了後の SO₁ 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		
SOL	説明	
0	リード時	SO ₁ 端子出力が"Low"レベル (初期値)
	ライト時	SO ₁ 端子出力を"Low"レベルに変更
1	リード時	SO ₁ 端子出力が"High"レベル
	ライト時	SO ₁ 端子出力を"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		
ORER	説明	
0	[クリア条件]	(初期値)
	"1"をリード後、"0"をライトしたとき	
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。各ビットはリードすると"0"が読み出されます。ライトは無効です。

ビット1：TAIL MARK 送信フラグ (MTRF)

MRKON="1"の時、TAIL MARK を送信中であることを示します。本ビットはリードのみ可能です。ライトは無効です。

ビット1		
MTRF	説明	
0	送信待ち状態および8ビットまたは16ビットデータ転送中	(初期値)
1	TAIL MARK 送信中	

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。入れ替わりにSI₁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 に書込まれたデータは、SO₁ 端子より LSB ファーストで出力されます。入れ替わりに SI₁ 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。

(5) クロック停止レジスタ 1 (CKSTPR1)

ビット:	7	6	5	4	3	2	1	0
	S1CKSTP	S31CKSTP	S32CKSTP	ADCKSTP	TGCKSTP	TFCKSTP	TCCKSTP	TACKSTP
初期値:	1	1	1	1	1	1	1	1
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CKSTPR1 は、8 ビットのリード/ライト可能なレジスタで、モジュールのモジュールスタンバイモードの制御を行います。本章では SCI1 に関するビットのみ説明します。他のビットについては各モジュールの章を参照してください。

ビット 7 : SCI1 をモジュールスタンバイモード制御 (S1CKSTP)

SCI1 をモジュールスタンバイモードに設定および解除を制御します。

ビット 7	
S1CKSTP	説明
0	SCI1 はモジュールスタンバイモードに設定される。*
1	SCI1 のモジュールスタンバイモードは解除される。 (初期値)

【注】 * モジュールスタンバイモードに設定すると SCR1、SCSR1、SDRU、SDRL はリセット状態になります。

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、SCK₁ 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 ($f_w/1024 \sim f_w/4$) を SCK₁ 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、SCK₁ 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。

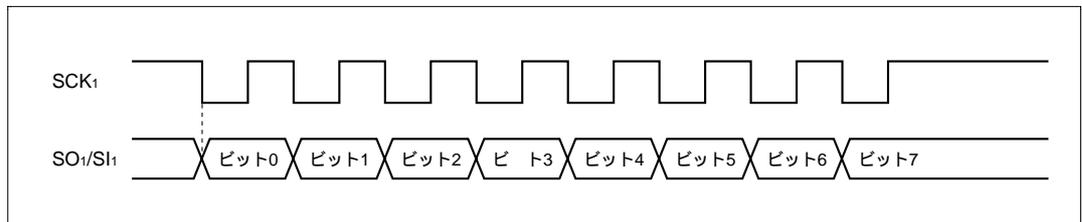


図 10.2 転送フォーマット

(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 SO1 を "1"、SCK1 を "1" としてそれぞれ SO₁ 端子、SCK₁ 端子に設定します。

また、必要に応じて、PMR2 の POF1 により、SO₁ 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"、MRKON を "0" 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 の MRKON が "0" の状態で SCR1 ヘ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、SO₁ 端子から送信データが出力されます。
- (5) 送信完了後、IRR1 の IRRS1 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 SCK₁ 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、SO₁ 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、SCK₁ 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 SO₁ 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 SI1 を "1"、SCK1 を "1" としてそれぞれ SI₁ 端子、SCK₁ 端子に設定します。
- (2) SCR1 の SNC1 を "0" とし、SNC0 を "0" または "1"、MRKON を "0" 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 の MRKON が "0" の状態で 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、SI₁ 端子から受信データを取り込みます。
- (4) 受信完了後、IRR1 の IRRS1 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SO₁を"1"、SI₁を"1"、SCK₁を"1"としてそれぞれSO₁端子、SI₁端子、SCK₁端子に設定します。また、必要に応じてPMR2のPOF₁により、SO₁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"、MRKONを"0"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のMRKONが"0"の状態では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、SO₁端子から送信データが出力され、またSI₁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SCK₁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、SO₁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、SCK₁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SO₁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 SSB モード時の動作説明

SSB 通信方式は SCL (Serial Clock) と SDA (Serial Data) との 2 線で構成され、図 10.3 で示すように接続することで複数の IC を制御することができます。

SSB モードは 8 ビットまたは 16 ビットデータ転送後、TAIL MARK を付加して送信します。TAIL MARK は HOLD TAIL と LATCH TAIL から選択できます。

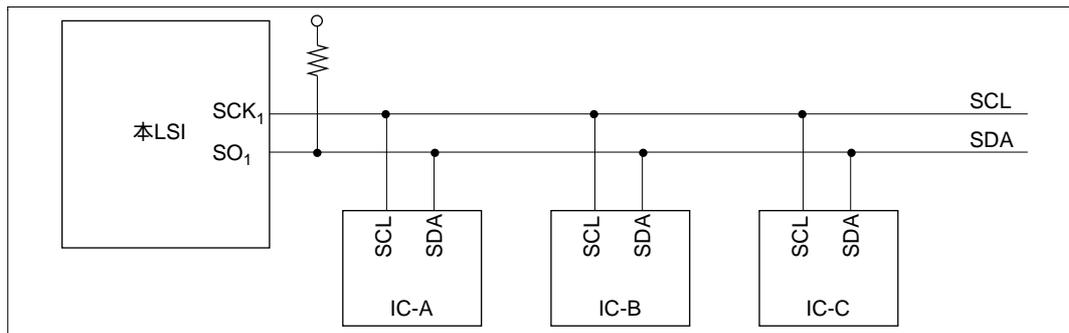


図 10.3 SSB 接続例

(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が、本 LSI がクロック出力となるため外部クロックは選択しないでください。また転送レートは、SCR1 の CKS2 ~ CKS0 で選択できますが、TAIL MARK の転送レートと兼ねているため、転送クロック周期が $2 \mu\text{s}$ 以上となるように設定してください。

(2) データ転送フォーマット

SCI1 の転送フォーマットを図 10.4 に示します。データ転送は、データの最下位ビットから送信される LSB ファースト方式による転送を行います。8 ビットまたは 16 ビットデータ転送後、TAIL MARK を付加します。

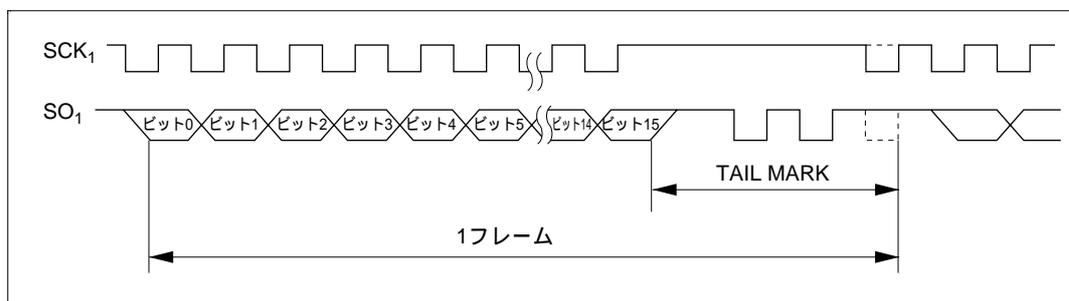


図 10.4 転送フォーマット (SNC1 = 0、SNC0 = 1、MRKON = 1 の時)

(3) TAIL MARK

TAIL MARK には HOLD TAIL と LATCH TAIL があります。HOLD TAIL と LATCH TAIL の出力波形を図 10.5 に示します。図 10.5 中の時間 t は SCR1 の CKS2 ~ CKS0 で設定された転送クロック周期から決まる時間です。

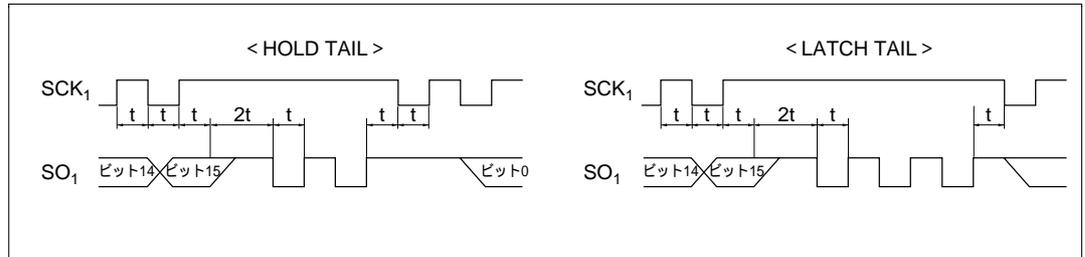


図 10.5 HOLD TAIL と LATCH TAIL の出力波形

(4) 送信動作

送信手順と動作を以下に示します。

- (1) SCSR1 の SOL を "1" にセットします。
- (2) PMR2 の SO1 を "1"、SCK1 を "1" とし、それぞれ SO₁ 端子、SCK₁ 端子に設定します。
また PMR2 の POF1 を "1" にセットし、SO₁ 端子を NMOS オープンドレイン出力とします。
- (3) SCR1 の SNC1 を "0"、SNC0 を "0" または "1" とし、8 ビットモードか 16 ビットモードを設定します。また SCR1 の MRKON を "1" にセットし、SSB モードを選択します。
- (4) SDRL、SDRU に転送データを書込みます。また SCR1 の LTCH で TAIL MARK を設定します。
8 ビットモード : SDRL
16 ビットモード : 上位バイト SDRU、下位バイト SDRL
- (5) SCSR1 の STF を "1" にすると SC11 は動作を開始し、SO₁ 端子から送信データが出力されます。
- (6) 8 ビットまたは 16 ビットデータ送信後、SCSR1 の STF は "0" にリセットされ、同時に IRR2 の IRRS1 は "1" にセットされます。またデータ送信に続いて、設定された TAIL MARK が出力されます。TAIL MARK 出力中 SCSR1 の MTRF は "1" になります。

(4) ~ (6) を繰り返し行うことにより、連続的にデータを送信することができます。なお、SCR1 の MRKON を書換える場合は、転送待ち状態であることを確認してから行ってください。

10.2.5 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。

10.2.6 使用上の注意事項

- (1) SCK₁ 端子を入力に設定し、クロックソースとして外部クロックを選択して使用する場合、SCSR1 の STF に"1"をライトして転送動作を開始する前に外部クロックを入力しないでください。
- (2) サブアクティブモード、サブスリープモード時は CPU 動作クロックが $f_w/2$ のときのみ SCI1 が使用可能となります。

10.3 SCI3

10.3.1 概要

本 LSI には SCI3-1 と SCI3-2 の 2 つのシリアルコミュニケーションインタフェースが搭載されています。この 2 つの SCI はまったく同一機能を有しています。

本マニュアルでは、これらを SCI3 と総称して説明いたします。

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 16 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 / 5 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 _{3x} 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、送信データエンプティ、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10.6に示します。

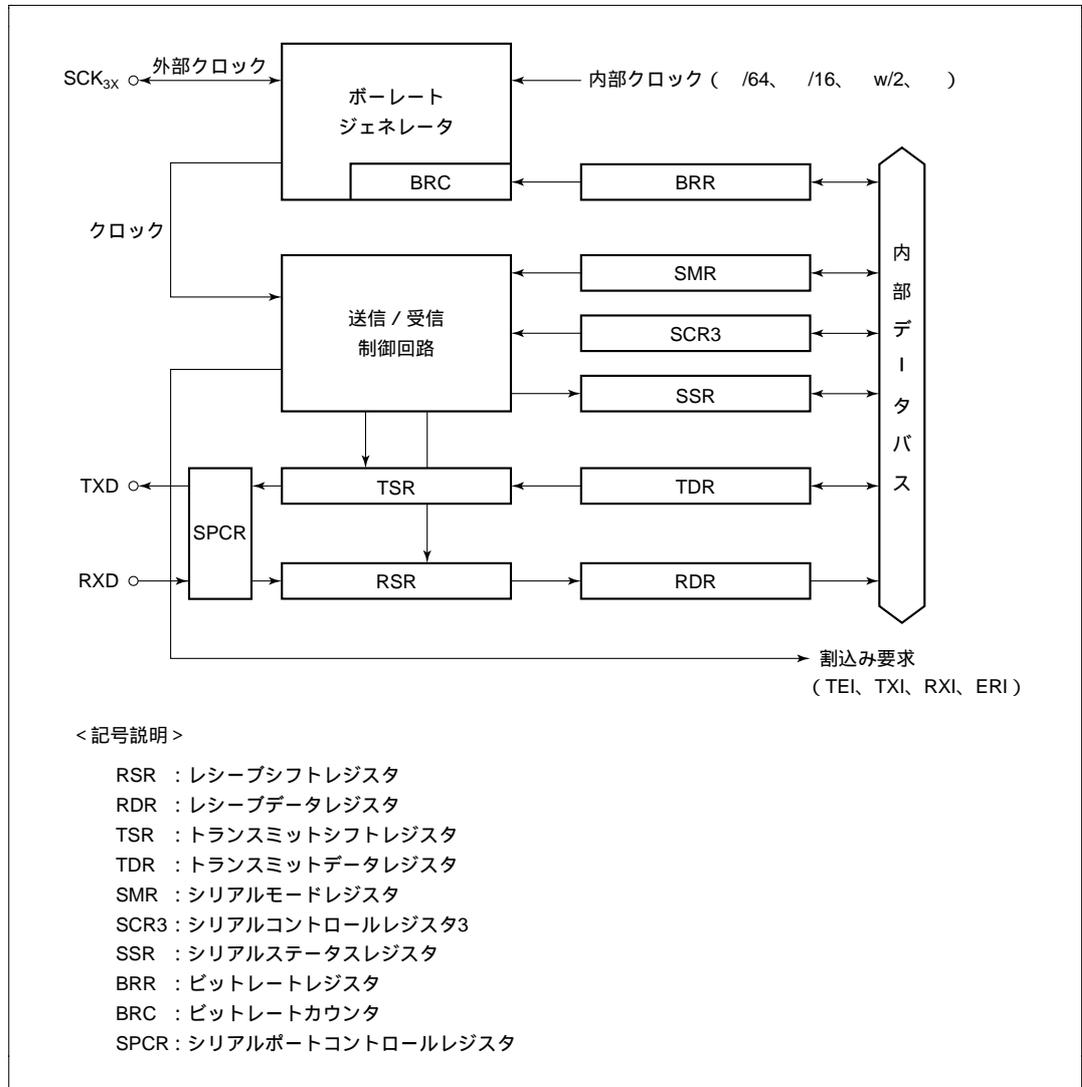


図 10.6 SCI3のブロック図

(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 _{3x}	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 _{3x}	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 _{3x}	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/FF98
ビットレートレジスタ	BRR	R/W	H'FF	H'FFA9/FF99
シリアルコントロールレジスタ 3	SCR3	R/W	H'00	H'FFAA/FF9A
トランスミットデータレジスタ	TDR	R/W	H'FF	H'FFAB/FF9B
シリアルステータスレジスタ	SSR	R/W	H'84	H'FFAC/FF9C
レシーブデータレジスタ	RDR	R	H'00	H'FFAD/FF9D
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		
クロック停止レジスタ 1	CKSTPR1	R/W	H'FF	H'FFFA
シリアルポートコントロールレジスタ	SPCR	R/W	H'C0	H'FF91

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_{3x} 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、モジュールスタンバイモード時に 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_{3X} 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、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、モジュールスタンバイモード時に 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、ボ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、モジュールスタンバイモード時に 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/5ビットデータ*2 (初期値)
1	7ビットデータ*1/5ビットデータ*2

【注】 *1 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。

*2 5ビットデータを選択する場合は、PE=1、MP=1としてください。

また、TDRのMSB(ビット7、ビット6、ビット5)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* ²	(初期値)
1	パリティビットの付加およびチェックを許可* ^{1*2}	

【注】 *1 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。

*2 5ビットデータを選択した場合は表 10.11 を参照してください。

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* ¹	(初期値)
1	奇数パリティ* ²	

【注】 *1 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。

*2 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	説明	
STOP		
0	1 ストップビット*1	(初期値)
1	2 ストップビット*2	

【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「10.1.6 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	説明	
MP		
0	マルチプロセッサ通信機能を禁止*1	(初期値)
1	マルチプロセッサ通信機能を許可*1	

【注】 *1 5 ビットデータを選択した場合は表 10.11 を参照してください。

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、 $/64$ 、 $/16$ 、 $w/2$ 、 w 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	$w/2$ クロック*1 / w クロック*2
1	0	$/16$ クロック
1	1	$/64$ クロック

【注】 *1 アクティブ（中速/高速）、スリープ（中速/高速）時では $w/2$ クロックとなります。

*2 サブアクティブ、サブスリープ時では w クロックとなります。また、サブアクティブ、サブスリープモード時は、CPU 動作クロックが $w/2$ のときのみ SCI3 が使用可能となります。

(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、モジュールスタンバイモード時に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、送信データエンプティ割込み要求（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	
TIE	説明
0	送信データエンプティ割込み要求（TXI）の禁止（初期値）
1	送信データエンプティ割込み要求（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 端子は入出力ポート）（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。

*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 の設定と SPCR の SPC31 または SPC32 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	説明	
RE		
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	説明	
MPIE		
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（SCR3 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 がないとき、送信終了割り込み要求 (TEI) の許可 / 禁止を選択するビットです。

ビット2	説明
TEIE	
0	送信終了割り込み要求 (TEI) を禁止 (初期値)
1	送信終了割り込み要求 (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_{3X} 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_{3X} 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「10.1.3 動作概要」の表 10.4 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 _{3X} 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、モジュールスタンバイモード、ウォッチモード時に 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	説明
TDRE	
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 （初期値） [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*1 [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*2 [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 (初期値) [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

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	説明
MPBT	
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で選択されるボ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、モジュールスタンバイモード、ウォッチモード時に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）

OSC	32.8KHz			38.4KHz			2MHz			2.4576MHz			4MHz		
	n	N	誤差(%)	n	N	誤差(%)	n	N	誤差(%)	n	N	誤差(%)	n	N	誤差(%)
110				—	—	—	—	—	—	2	21	-0.83	—	—	—
150				0	3	0	2	12	0.16	3	3	0	2	25	0.16
200				0	2	0	0	155	0.16	3	2	0	—	—	—
250				—	—	—	0	124	0	0	153	-0.26	0	249	0
300				0	1	0	0	103	0.16	3	1	0	2	12	0.16
600				0	0	0	0	51	0.16	3	0	0	0	103	0.16
1200				—	—	—	0	25	0.16	2	1	0	0	51	0.16
2400				—	—	—	0	12	0.16	2	0	0	0	25	0.16
4800				—	—	—	—	—	—	0	7	0	0	12	0.16
9600				—	—	—	—	—	—	0	3	0	—	—	—
19200				—	—	—	—	—	—	0	1	0	—	—	—
31250				—	—	—	—	—	—	—	—	—	0	1	0
38400				—	—	—	—	—	—	0	0	0	—	—	—

- 【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。

$$N = \frac{\text{OSC}}{64 \times 2^{2n} \times B} - 1$$

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 : ω_{osc} の値 (Hz)

n : ボーレートジェネレータの入力クロックの No. (n=0, 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
0	$\omega/2^{*1} / \omega^{*2}$	0	1
2	/16	1	0
3	/64	1	1

- 【注】 *1 アクティブ (中速 / 高速)、スリープ (中速 / 高速) 時では $\omega/2$ クロックとなります。
 *2 サブアクティブ、サブスリープ時では ω クロックとなります。また、サブアクティブモード、サブスリープモード時は、CPU 動作クロックが $\omega/2$ のときのみ SCI3 が使用可能となります。
 *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
0.0384*	600	0	0
2	31250	0	0
2.4576	38400	0	0
4	62500	0	0

- 【注】 * SMR を CKS1 = "0"、CKS0 = "1" に設定したとき

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〕

OSC	38.4KHz			2MHz			4MHz		
	n	N	誤差率	n	N	誤差率	n	N	誤差率
200	0	23	0	—	—	—	—	—	—
250	—	—	—	—	—	—	2	124	0
300	2	0	0	—	—	—	—	—	—
500				—	—	—	—	—	—
1K				0	249	0	—	—	—
2.5K				0	99	0	0	199	0
5K				0	49	0	0	99	0
10K				0	24	0	0	49	0
25K				0	9	0	0	19	0
50K				0	4	0	0	9	0
100K				—	—	—	0	4	0
250K				0	0	0	0	1	0
500K							0	0	0
1M									

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。

$$N = \frac{\text{OSC}}{8 \times 2^{2n} \times B} - 1$$

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：_{osc} の値 (Hz)

n：ボーレートジェネレータの入力クロックの No. (n=0、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
0	$w/2^{*1} / w^{*2}$	0	1
2	/16	1	0
3	/64	1	1

【注】 *1 アクティブ（中速／高速）、スリープ（中速／高速）時では $w/2$ クロックとなります。

*2 サブアクティブ、サブスリープ時は、 w クロックとなります。またサブアクティブモード、サブスリープモード時は CPU 動作クロックが $w/2$ のときのみ SCI3 が使用可能となります。

(9) クロック停止レジスタ 1 (CKSTPR1)

ビット:	7	6	5	4	3	2	1	0
	S1CKSTP	S31CKSTP	S32CKSTP	ADCKSTP	TGCKSTP	TFCKSTP	TCCKSTP	TACKSTP
初期値:	1	1	1	1	1	1	1	1
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CKSTPR1 は、8 ビットのリード／ライト可能なレジスタで、モジュールのモジュールスタンバイモードの制御を行います。本章では SCI3 に関するビットのみ説明します。他のビットについては各モジュールの章を参照してください。

ビット 6 : SCI31 をモジュールスタンバイモード制御 (S31CKSTP)

SCI31 をモジュールスタンバイモードに設定および解除を制御します。

S31CKSTP	説明
0	SCI31 はモジュールスタンバイモードに設定される。*
1	SCI31 のモジュールスタンバイモードは解除される。 (初期値)

【注】 * モジュールスタンバイモードに設定した場合 SCI31 の全レジスタはリセット状態となります。

ビット 5 : SCI32 をモジュールスタンバイモード制御 (S32CKSTP)

SCI32 をモジュールスタンバイモードに設定および解除を制御します。

S32CKSTP	説明
0	SCI32 はモジュールスタンバイモードに設定される。*
1	SCI32 のモジュールスタンバイモードは解除される。 (初期値)

【注】 * モジュールスタンバイモードに設定した場合 SCI32 の全レジスタはリセット状態となります。

(10) シリアルポートコントロールレジスタ (SPCR)

ビット:	7	6	5	4	3	2	1	0
	—	—	SPC32	SPC31	SCINV3	SCINV2	SCINV1	SCINV0
初期値:	1	1	0	0	0	0	0	0
R/W :	—	—	R/W	R/W	R/W	R/W	R/W	R/W

INVRは、8ビットのリード/ライト可能なレジスタで、RXD₃₁、RXD₃₂、TXD₃₁、TXD₃₂端子の入出力データ反転切換を行います。リセット時、INVRはH'00にイニシャライズされます。

ビット7、6：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：P4₂/TXD₃₂端子機能切換え

P4₂/TXD₃₂端子をP4₂端子として使用するかTXD₃₂端子として使用するかを設定します。

ビット5	
SPC32	説明
0	P4 ₂ 入出力端子として機能。(初期値)
1	TXD ₃₂ 出力端子として機能。*

【注】 * :本ビットを"1"に設定した後にSCR3のTEビットを設定してください。

ビット4：P3₅/TXD₃₁端子機能切換え

P3₅/TXD₃₁端子をP3₅端子として使用するかTXD₃₁端子として使用するかを設定します。

ビット4	
SPC31	説明
0	P3 ₅ 入出力端子として機能。(初期値)
1	TXD ₃₁ 出力端子として機能。*

【注】 * :本ビットを"1"に設定した後にSCR3のTEビットを設定してください。

ビット3：TXD₃₂端子出力データ反転切換え

TXD₃₂端子の出力データを反転するか、しないか、の切換えを設定します。

ビット3	
SCINV3	説明
0	TXD ₃₂ の出力データを反転しない。(初期値)
1	TXD ₃₂ の出力データを反転する。

ビット2：RXD₃₂ 端子入力データ反転切換え

RXD₃₂ 端子の入力データを反転するか、しないか、の切換えを設定します。

ビット2		
SCINV2	説明	
0	RXD ₃₂ の入力データを反転しない。	(初期値)
1	RXD ₃₂ の入力データを反転する。	

ビット1：TXD₃₁ 端子出力データ反転切換え

TXD₃₁ 端子の出力データを反転するか、しないか、の切換えを設定します。

ビット1		
SCINV1	説明	
0	TXD ₃₁ の出力データを反転しない。	(初期値)
1	TXD ₃₁ の出力データを反転する。	

ビット0：RXD₃₁ 端子入力データ反転切換え

RXD₃₁ 端子の入力データを反転するか、しないか、の切換えを設定します。

ビット0		
SCINV0	説明	
0	RXD ₃₁ の入力データを反転しない。	(初期値)
1	RXD ₃₁ の入力データを反転する。	

10.3.3 動作説明

(1) 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。

(a) 調歩同期式モード

- ・データ長：5ビット/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：ボ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(内蔵ボーレートジェネレータを使用しない)

(b) 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：ボ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：内蔵ボ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		
0	0	0	0	1				2			
0	0	0	1	0				1			
0	0	0	1	1				2			
0	1	0	0	0				7ビット データ	なし	なし	1
0	1	0	0	1						2	
0	1	0	1	0		1					
0	1	0	1	1		2					
0	0	1	0	0		8ビット データ	あり	なし	1		
0	0	1	0	1		2					
0	0	1	1	0		5ビット データ	なし		1		
0	0	1	1	1		2					
0	1	1	0	0		7ビット データ	あり		1		
0	1	1	0	1		2					
0	1	1	1	0		5ビット データ	なし	あり	1		
0	1	1	1	1		2					
1	*	0	*	*	クロック同期式 モード	8ビット データ	なし	なし	なし		

* : 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 _{3X} 端子の機能
COM	CKE1	CKE0			
0	0	0	調歩同期式 モード	内部	入出力ポート (SCK _{3X} 端子を使用しません)
0	0	1			ビットレートと同じ周波数のクロックを出力
0	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 式モード	内部	同期クロックを出力
1	1	0		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			

(c) 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 10.7 (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 10.7 (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 10.7 (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込まれていないことを示します。

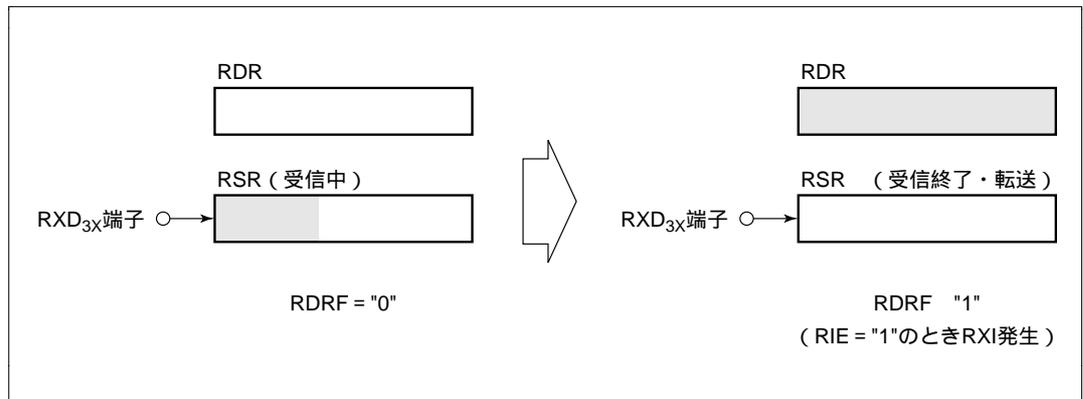


図 10.7 (a) RDRF のセットと RXI 割込み

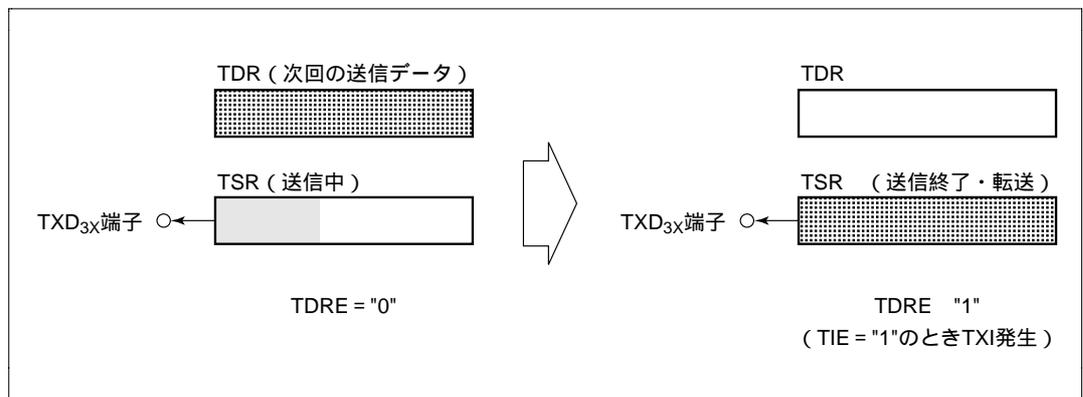


図 10.7 (b) TDRE のセットと TXI 割込み

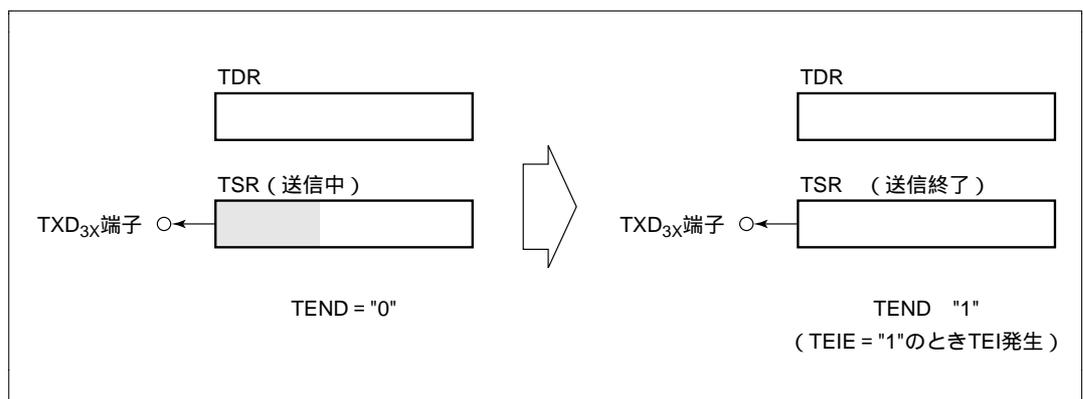


図 10.7 (c) TEND のセットと TEI 割込み

(2) 調歩同期式モード時の動作

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。

(a) 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 10.8 に示します。

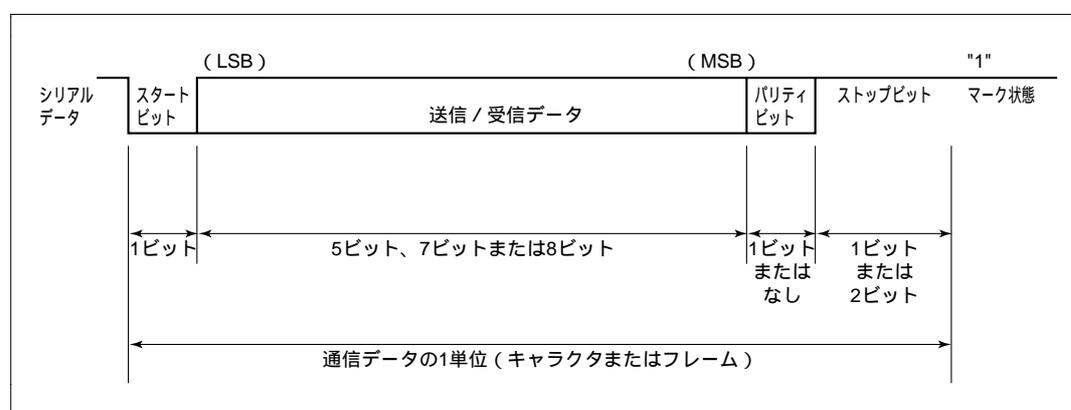


図 10.8 調歩同期式通信のデータフォーマット

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。

10. シリアルコミュニケーションインタフェース

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	START	8ビットデータ								STOP			
0	0	0	1	START	8ビットデータ								STOP	STOP		
0	0	1	0	START	8ビットデータ								MPB	STOP		
0	0	1	1	START	8ビットデータ								MPB	STOP	STOP	
0	1	0	0	START	8ビットデータ								P	STOP		
0	1	0	1	START	8ビットデータ								P	STOP	STOP	
0	1	1	0	START	5ビットデータ				STOP							
0	1	1	1	START	5ビットデータ				STOP	STOP						
1	0	0	0	START	7ビットデータ						STOP					
1	0	0	1	START	7ビットデータ						STOP	STOP				
1	0	1	0	START	7ビットデータ						MPB	STOP				
1	0	1	1	START	7ビットデータ						MPB	STOP	STOP			
1	1	0	0	START	7ビットデータ						P	STOP				
1	1	0	1	START	7ビットデータ						P	STOP	STOP			
1	1	1	0	START	5ビットデータ				P	STOP						
1	1	1	1	START	5ビットデータ				P	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 : マルチプロセッサビット

(b) 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_{3X} 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_{3X} 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_{3X} 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 10.9 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。

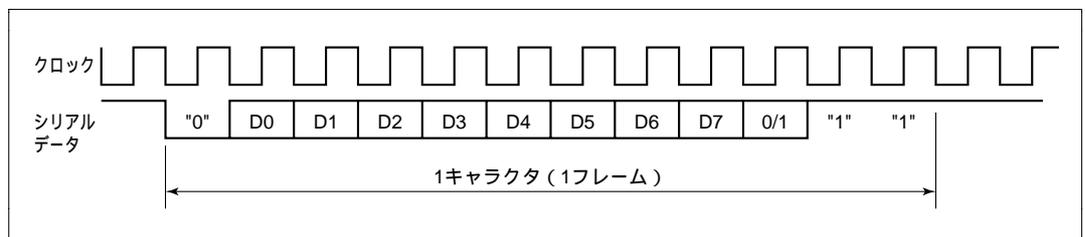


図 10.9 出力クロックと通信データの位相関係 (調歩同期式モード)
(8 ビットデータ / パリティあり / 2 ストップビットの例)

(c) データの送信 / 受信動作

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 10.10 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。

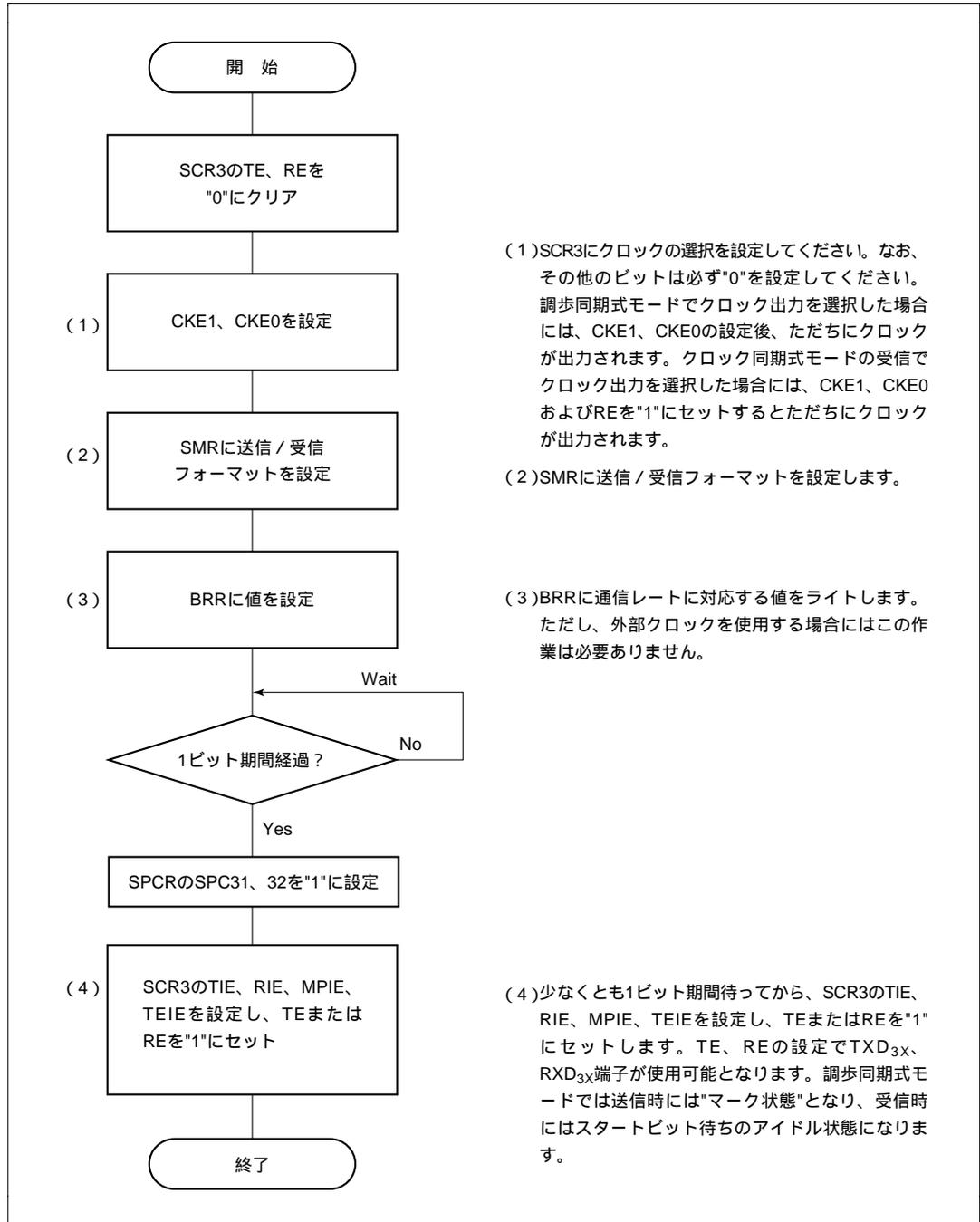


図 10.10 SCI3 をイニシャライズするときのフローチャートの例

データ送信

図 10.11 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。

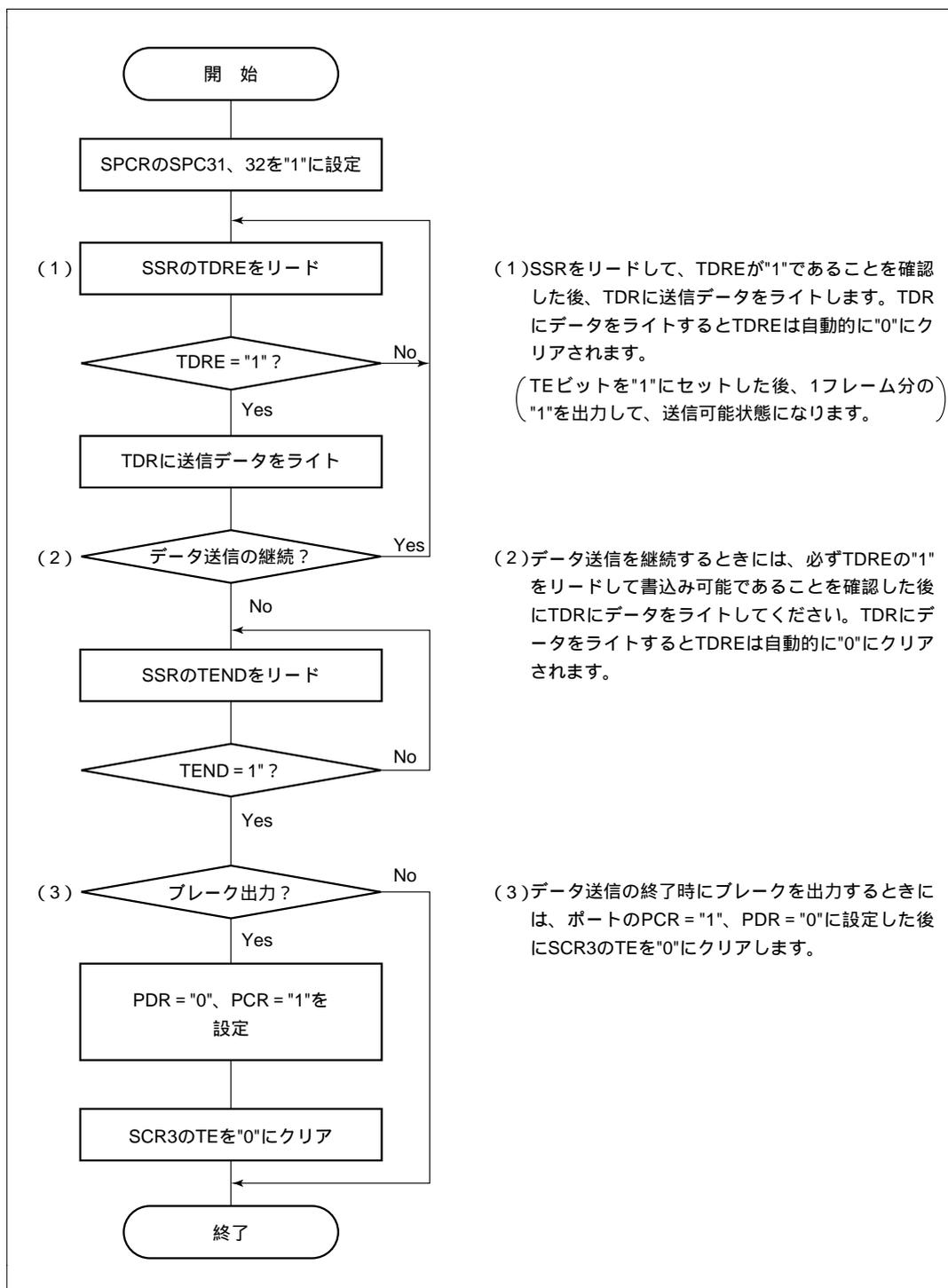


図 10.11 データ送信のフローチャートの例 (調歩同期式モード)

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_{3X} 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 10.12 に調歩同期式モードの送信時の動作例を示します。

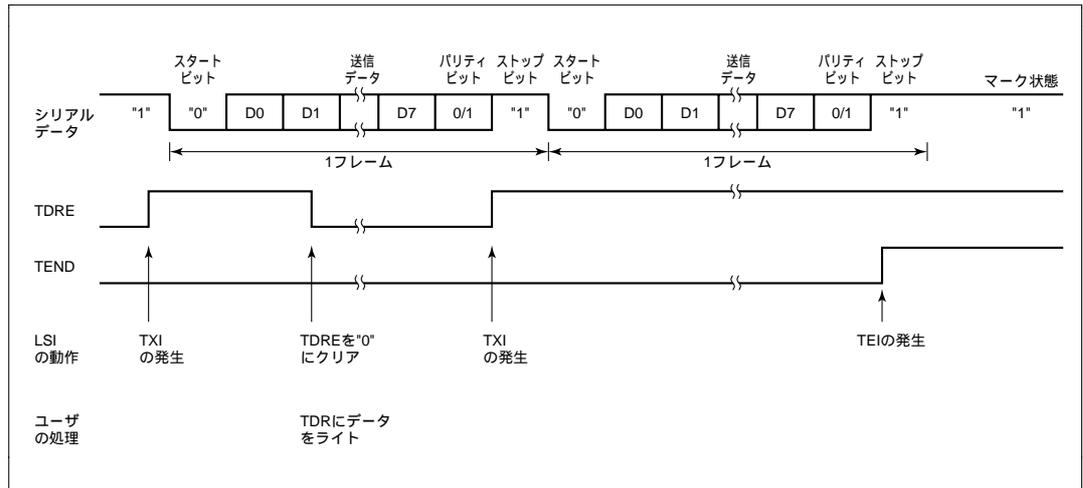


図 10.12 調歩同期式モードの送信時の動作例

(8 ビットデータ / パリティあり / 1 ストップビットの例)

データ受信

図 10.13 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。

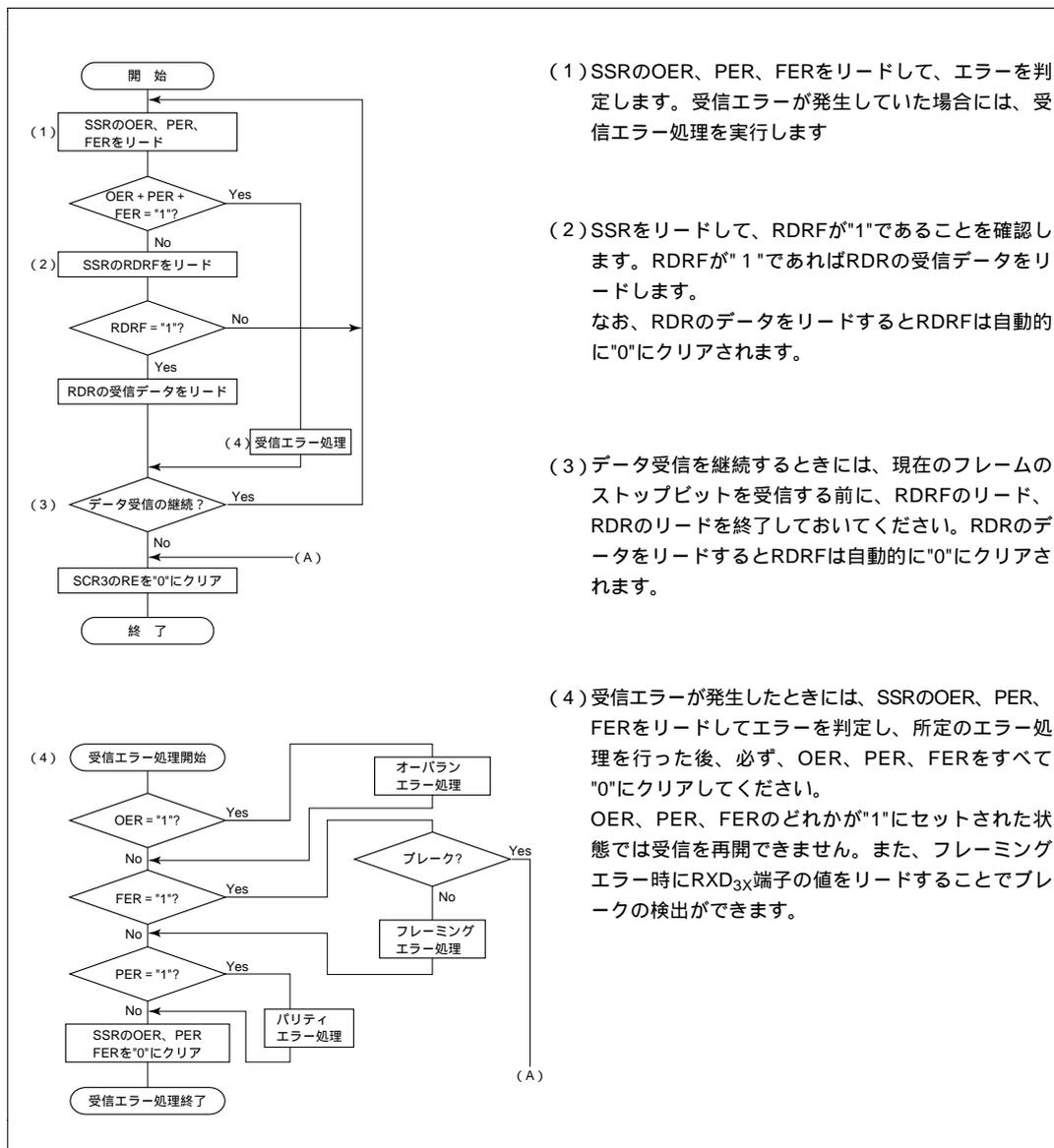


図 10.13 データ受信のフローチャートの例 (調歩同期式モード)

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 10.14 に示します。

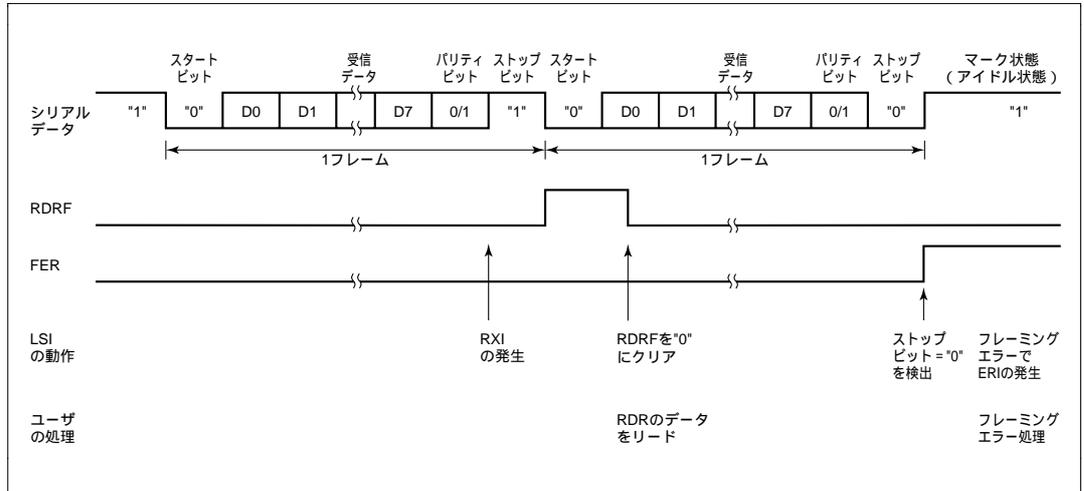


図 10.14 調歩同期モードの受信時の動作例
(8ビットデータ/パリティあり/1ストップビットの例)

(3) クロック同期式モード時の動作

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。

(a) 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 10.15 に示します。

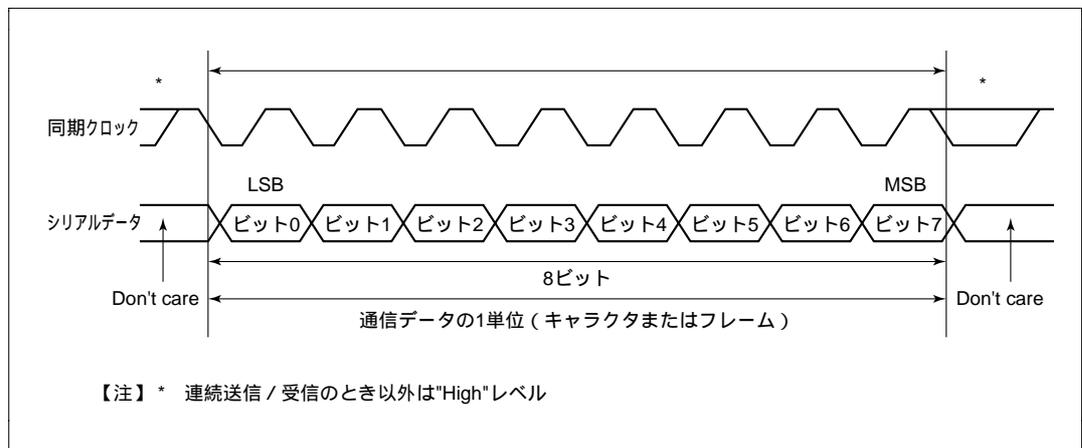


図 10.15 クロック同期式通信のデータフォーマット

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。

(b) 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_{3X}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_{3X}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。

(c) データの送信/受信動作

SCI3のイニシャライズ

データの送信/受信前には、「10.3.3(2)(c) SCI3のイニシャライズ」の説明および、図10.10の例にしたがってSCI3をイニシャライズしてください。

データ送信

図10.16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。

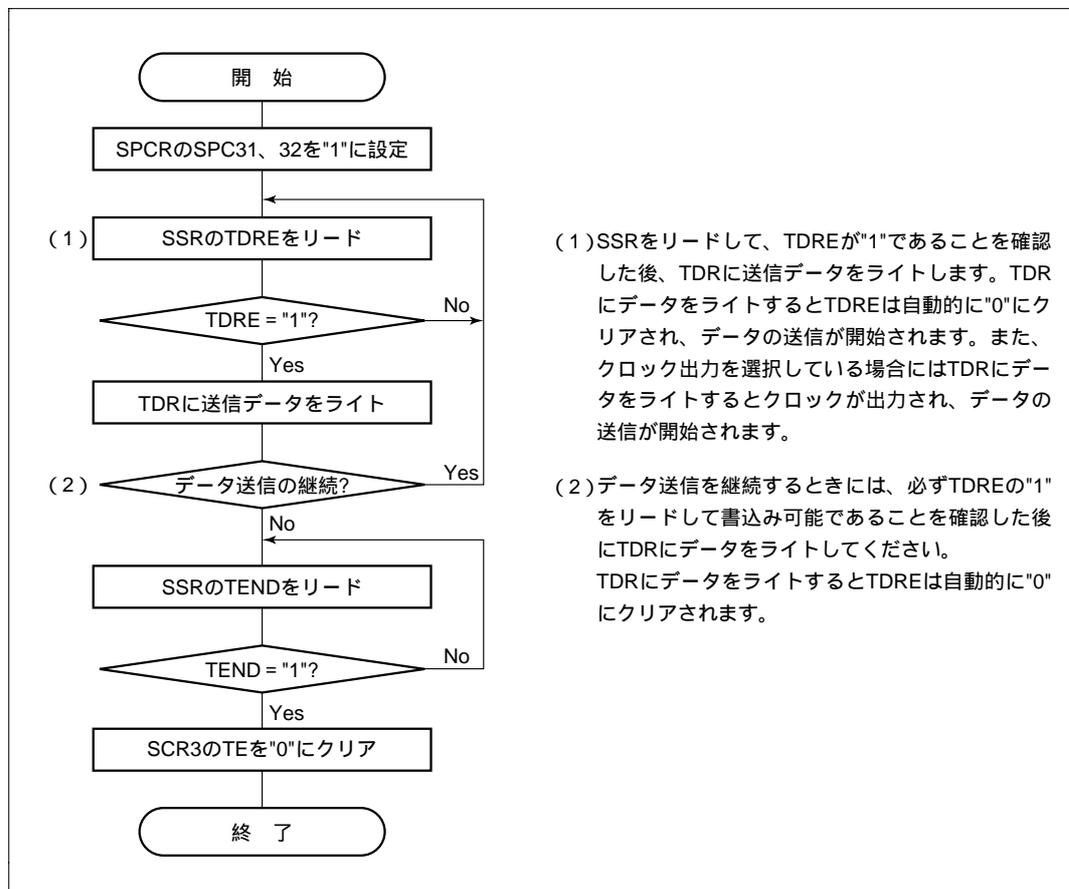


図10.16 データ送信のフローチャートの例(クロック同期式モード)

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_{3X} 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 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 10.17 にクロック同期式モードの送信時の動作例を示します。

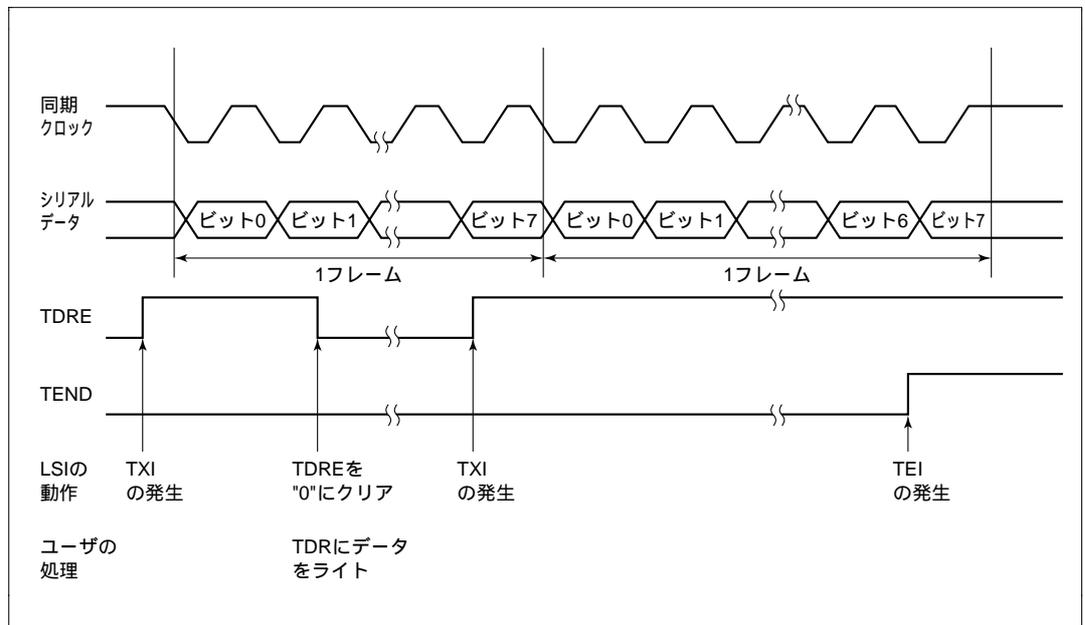


図 10.17 クロック同期式モードの送信時の動作例

データ受信

図 10.18 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。

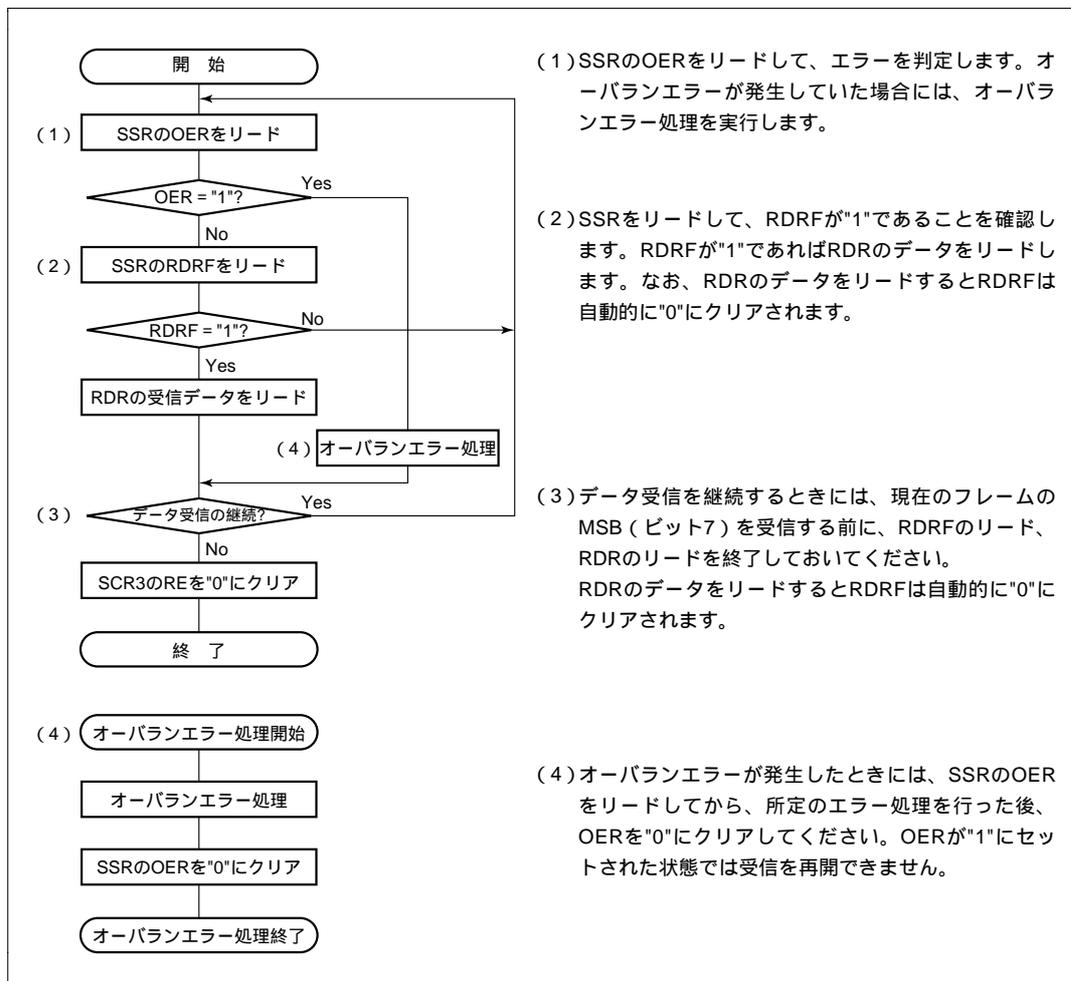


図 10.18 データ受信フローチャートの例 (クロック同期式モード)

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 10.19 にクロック同期式モードの受信時の動作例を示します。

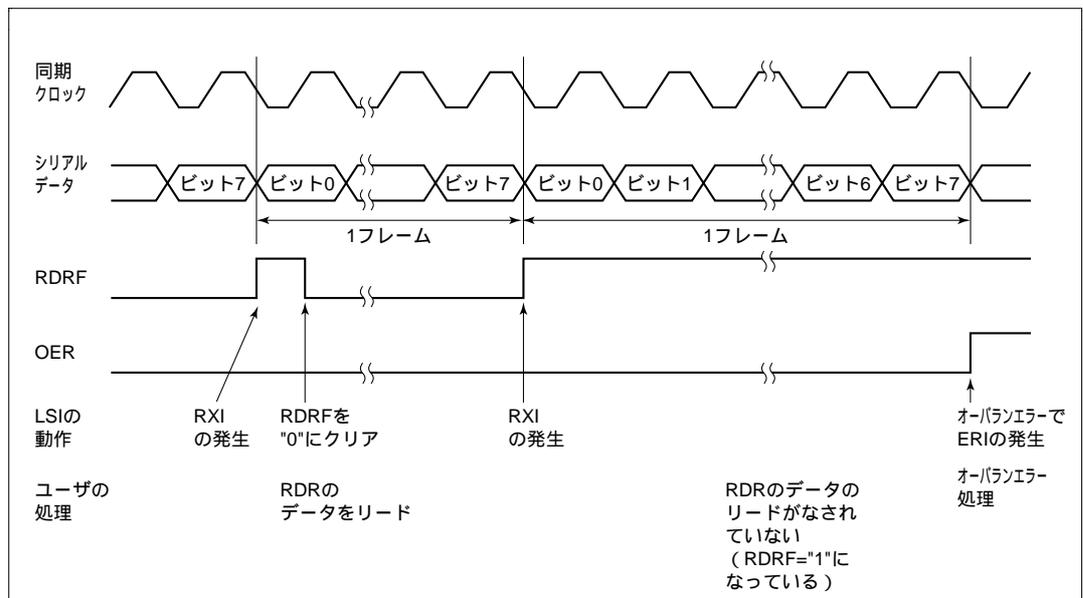


図 10.19 クロック同期式モードの受信時の動作例

データ送受信同時動作

図 10.20 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。

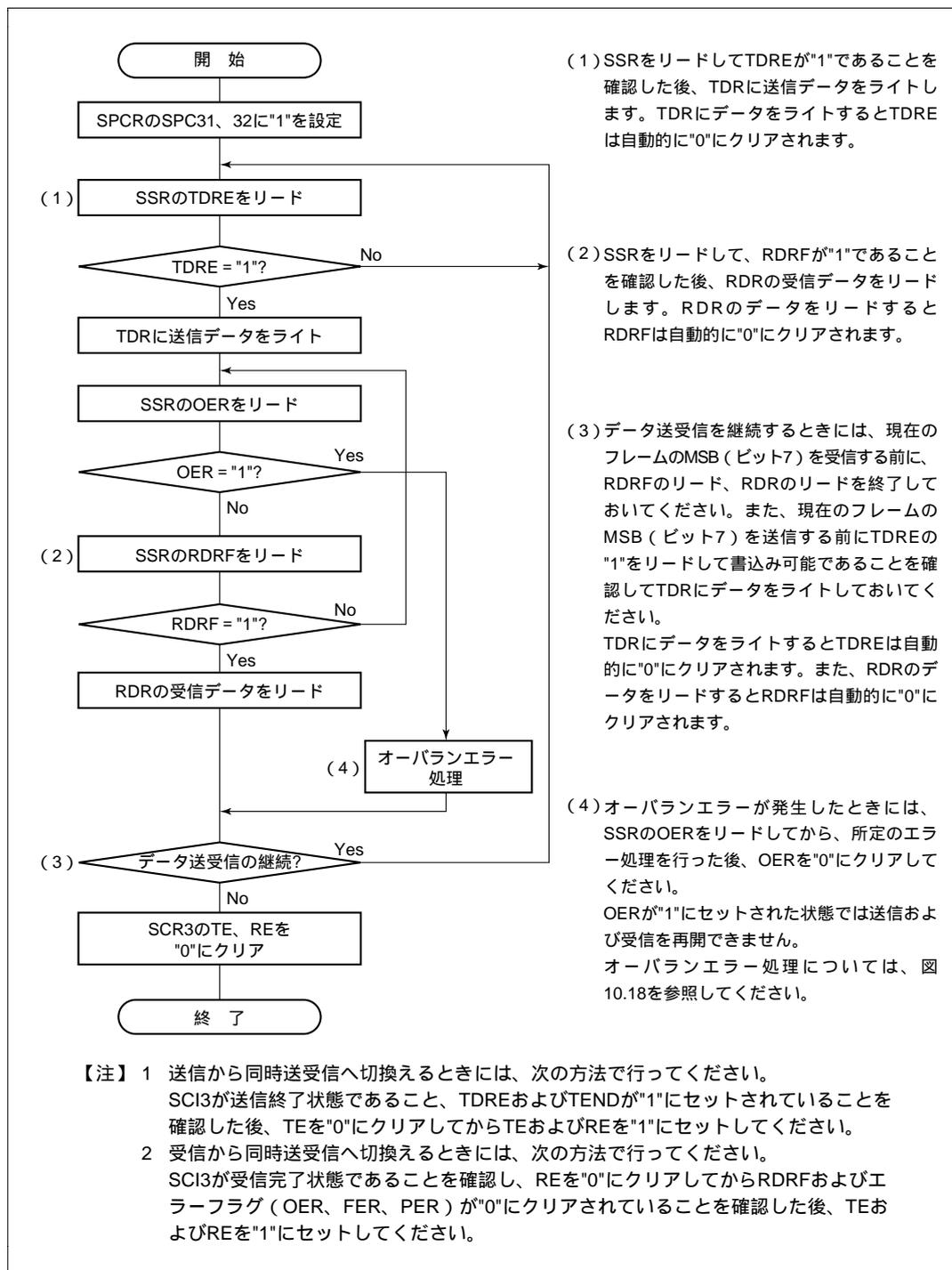


図 10.20 データ送受信同時動作のフローチャートの例 (クロック同期式モード)

(4) 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 10.21 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。

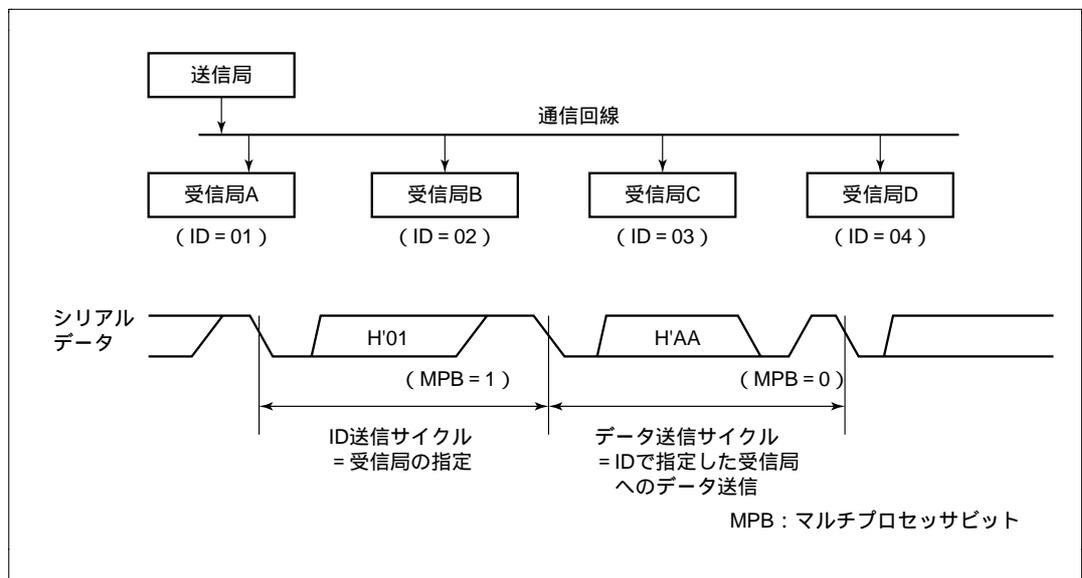


図 10.21 マルチプロセッサフォーマットを使用したプロセッサ間通信の例
(受信局 A へのデータ H'AA の送信の例)

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、「10.3.3 (2) 調歩同期式モード時の動作説明」を参照してください。

マルチプロセッサデータ送信

図 10.22 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。

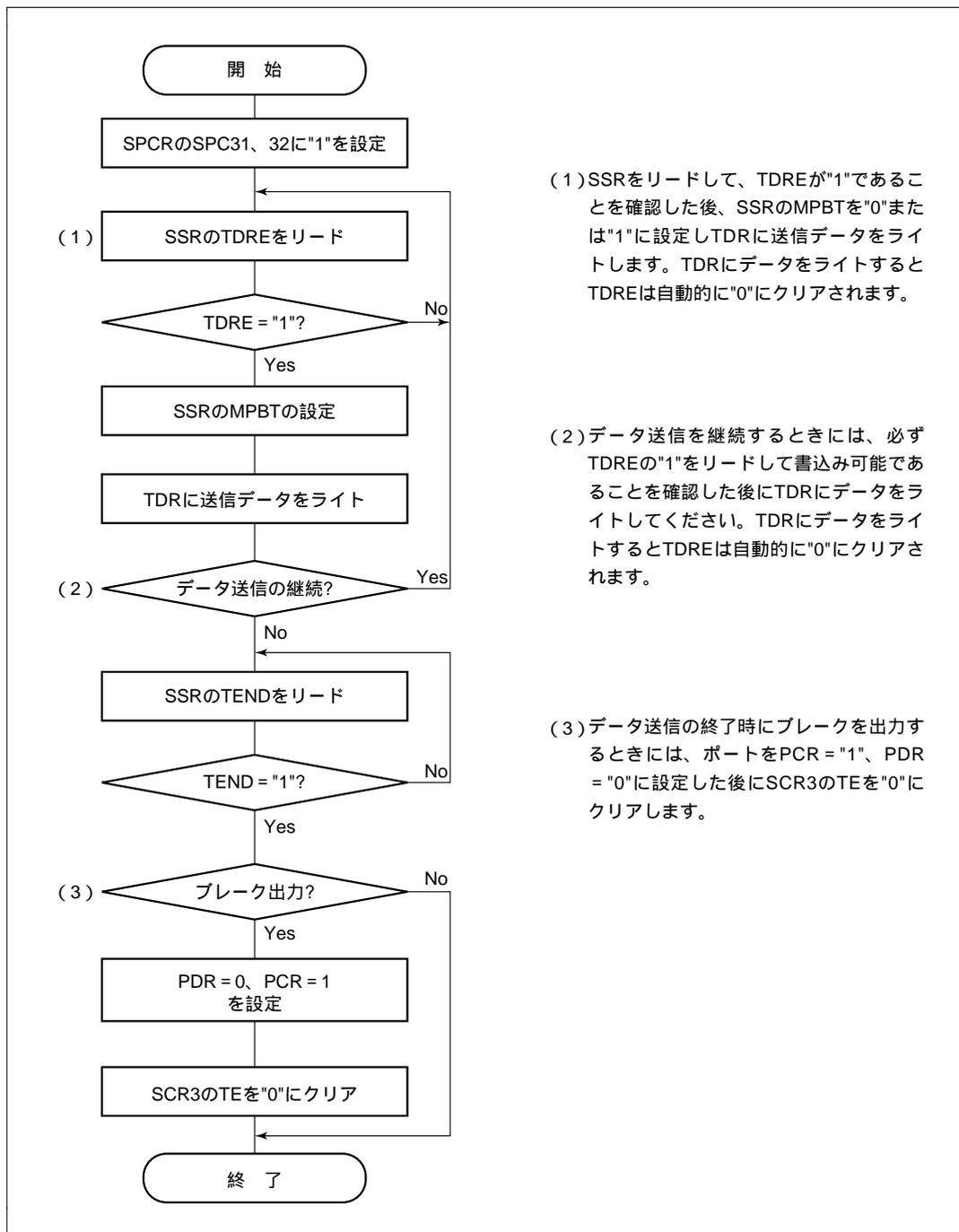


図 10.22 マルチプロセッサデータ送信のフローチャートの例

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 10.23 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。

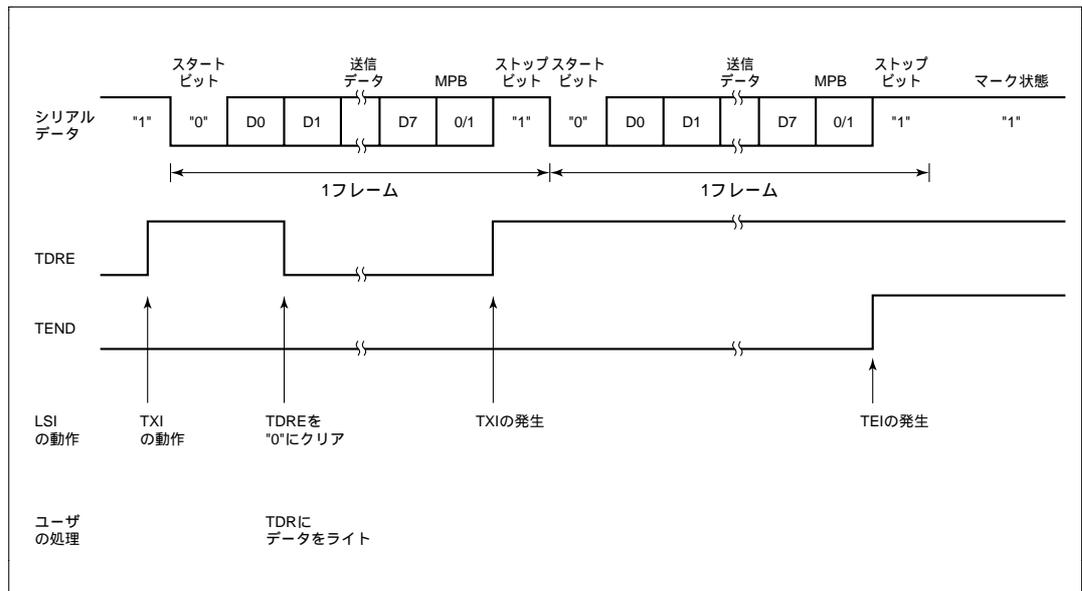


図 10.23 マルチプロセッサフォーマットの送信時の動作例
(8 ビットデータ / マルチプロセッサビットあり / 1 ストップビットの例)

マルチプロセッサデータ受信

図 10.24 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行ってください。

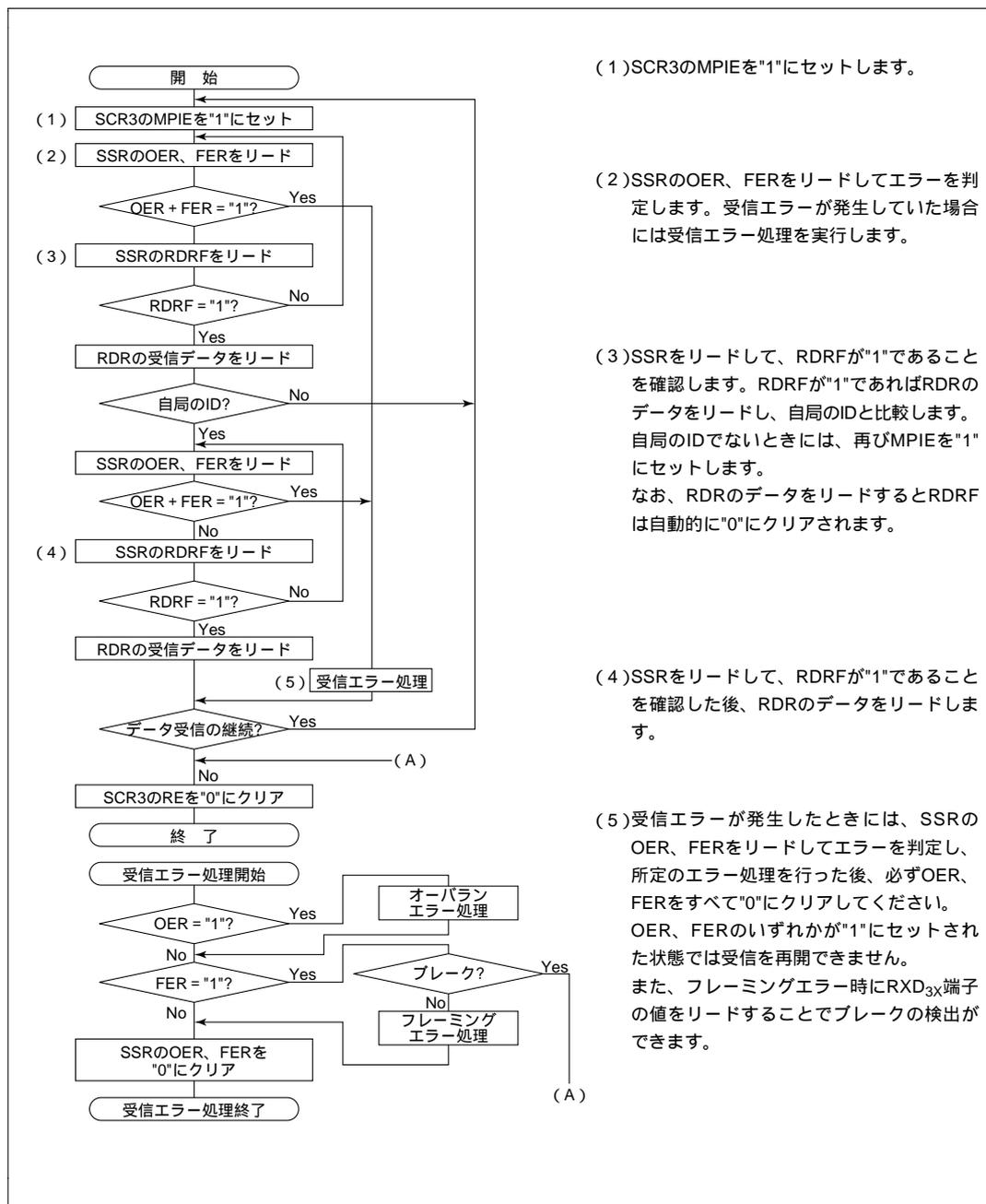


図 10.24 マルチプロセッサデータ受信のフローチャートの例

図 10.25 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。

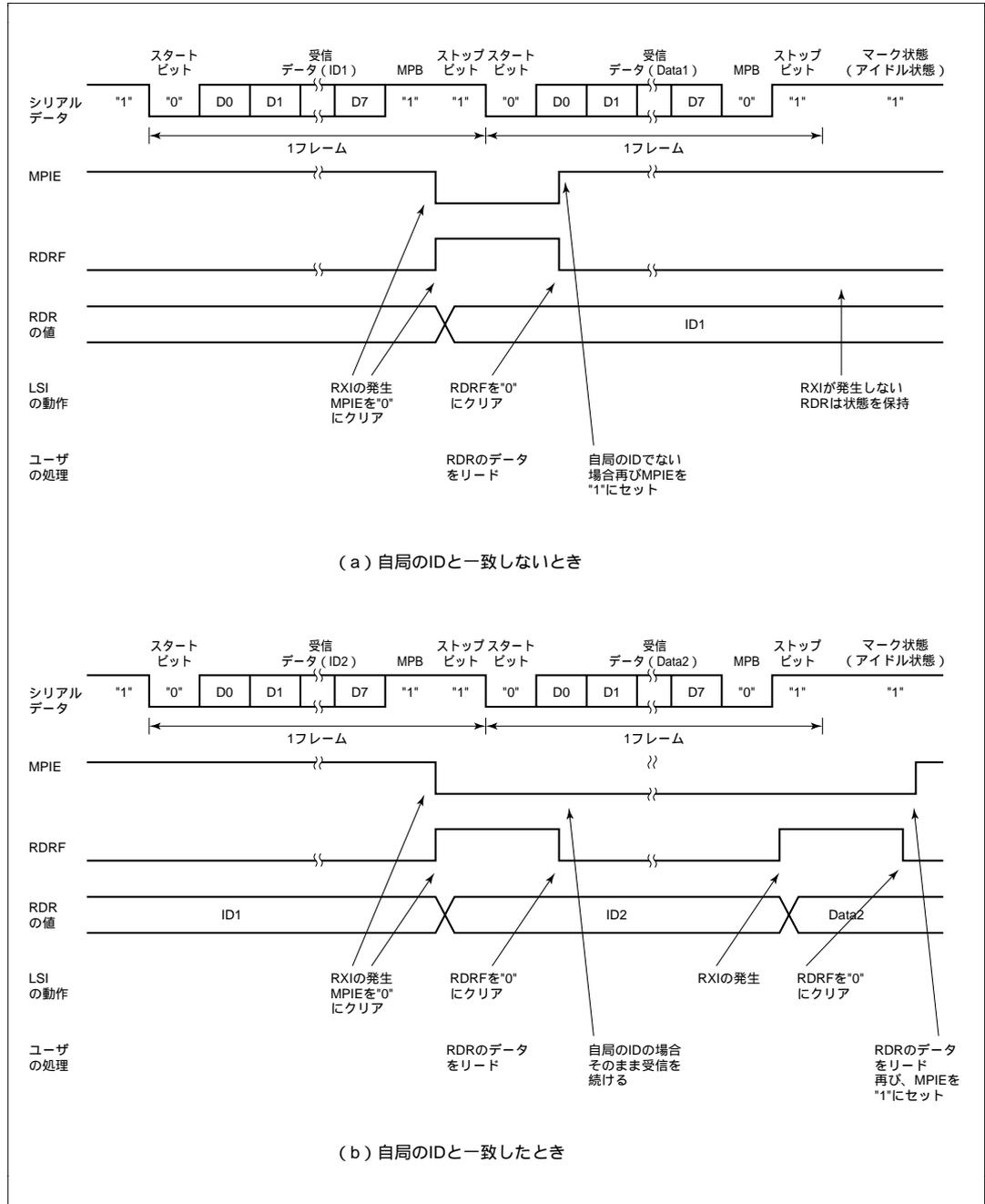


図 10.25 マルチプロセッサフォーマットの受信時の動作例
(8ビットデータ/マルチプロセッサビットあり/1ストップビットの例)

10.3.4 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'0022/ 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。

10.3.5 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を行うために 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。ただし、前のフレームの受信データのリードが遅れた結果、次のフレームのオーバランエラーが発生した後に RDR をリードした場合は、RDRF は"0"にクリアされますので注意してください。

(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_{3X}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_{3X}端子からの入力が入りすべて"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付

TEが"0"のとき、TXD_{3X}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_{3X}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付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、TEが"0"にクリアされていますので、TXD_{3X}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付したいときは、PCR="1"、PDR="0"に設定した後TEを"0"にクリアします。

TEを"0"にクリアすると現在の送信状態とは無関係に送信部は初期化され、TXD_{3X}端子はI/Oポートになり、TXD_{3X}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 10.26 に示します。

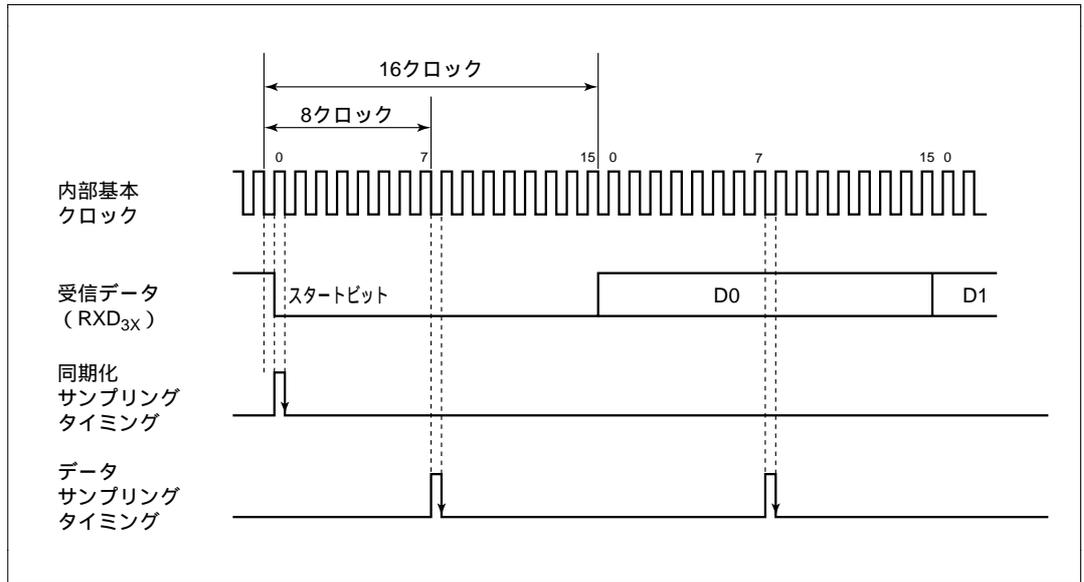


図 10.26 調歩同期式モードの受信データサンプリングタイミング

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"の状態でも 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 10.27 に示します。

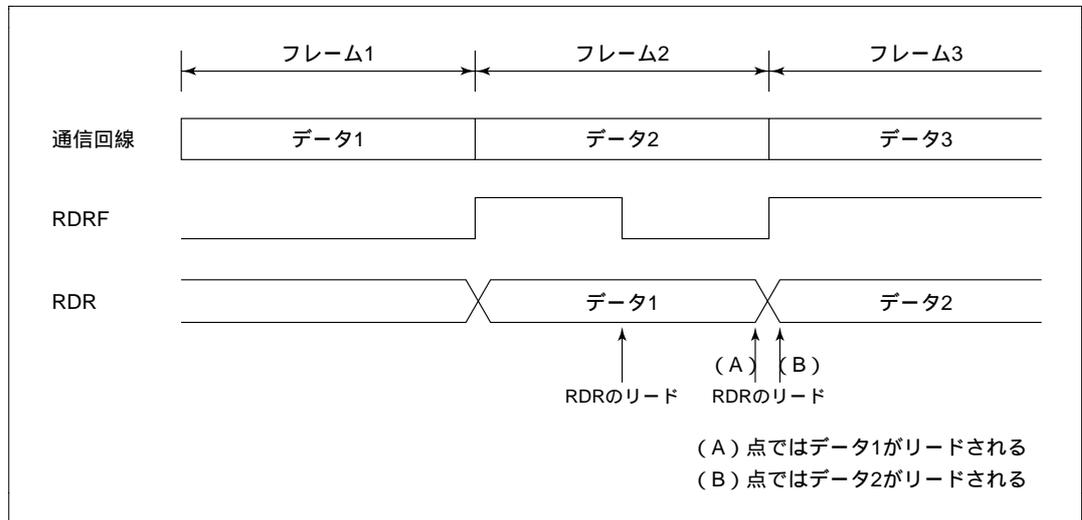


図 10.27 RDR のリードタイミングとデータの関係

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。

(8) 状態遷移時における送信および受信動作について

状態遷移処理を行う場合は、送信および受信動作が完全に終了した後、行ってください。

(9) SCK_{3X} 端子機能切換えに伴う注意事項

SCI3 をクロック同期式モードで使用した後、SCK_{3X} 端子をクロック出力から入出力ポートに端子機能を切換えると SCK_{3X} 端子に端子機能切換えのタイミング瞬時(システムクロックの1/2の期間)"Low"レベルを出力しますので注意してください。

この瞬時の"Low"レベル出力を回避するには次の方法があります。

(a) SCK_{3X} 端子をクロック出力状態から非出力状態にする場合

送受信を停止する際、1 命令で SCR3 の TE ビット、RE ビットを"0"にクリアすると同時に CKE1 ビットを"1"、CKE0 ビットを"0"に設定してください。

この場合は、SMR の COM ビットは"1"にセットされたままで使用してください。したがって、入出力ポートとしては使用できません。また、SCK_{3X} 端子に中間電位が印加しないように SCK_{3X} 端子に接続したラインは抵抗を介して V_{CC} 電位にプルアップするか、他のデバイスから出力を与えるかしてください。

(b) SCK_{3X} 端子をクロック出力状態から入出力ポートに端子機能を切換える場合

送受信を停止する際、

まず 1 命令で SCR3 の TE ビット、RE ビットを共に"0"にクリアすると同時に CKE1 ビットを"1"、CKE0 ビットを"0"に設定してください。

次に SMR の COM ビットを"0"にクリアしてください。

最後に SCR3 の CKE1、CKE0 ビットを共に"0"にクリアしてください。この場合も SCK_{3X} 端子に中間電位が印加しないように注意してください。

(10) サブアクティブモード、サブスリープモード時の設定について

サブアクティブモード、サブスリープモード時は CPU 動作クロックが $\frac{w}{2}$ のときのみ SCI3 が使用可能となります。SYSCR2 の SA1 ビットを"1"にセットしてください。

11. 14 ビット PWM

第 11 章 目次

11.1	概要.....	341
	11.1.1 特長.....	341
	11.1.2 ブロック図.....	341
	11.1.3 端子構成.....	342
	11.1.4 レジスタ構成.....	342
11.2	各レジスタの説明.....	343
	11.2.1 PWM コントロールレジスタ (PWCR)	343
	11.2.2 PWM データレジスタ U、L (PWDRU、PWDRL)	344
	11.2.3 クロック停止レジスタ 2 (CKSTPR2)	344
11.3	動作説明.....	345
	11.3.1 動作説明.....	345
	11.3.2 PWM の動作モード.....	346

11.1 概要

本LSIは、14ビットPWM (Pulse Width Modulator) を内蔵しています。ローパスフィルタを接続することでD/A変換器として使用できます。

11.1.1 特長

14ビットPWMの特長を以下に示します。

4種類の変換周期を選択可能

1変換周期 $131,072/f$ 、最小変化幅 $8/f$ (PWCR1="1"、PWCR0="1")、

1変換周期 $65,536/f$ 、最小変化幅 $4/f$ (PWCR1="1"、PWCR0="0")、

1変換周期 $32,768/f$ 、最小変化幅 $2/f$ (PWCR1="0"、PWCR0="1")、

または1変換周期 $16,384/f$ 、最小変化幅 $1/f$ (PWCR1="0"、PWCR0="0")

の選択が可能です。

リップル低減を図ったパルス分割方式

モジュールスタンバイモードにより、未使用時はモジュール単体でスタンバイモードに設定可能

11.1.2 ブロック図

14ビットPWMのブロック図を図11.1に示します。

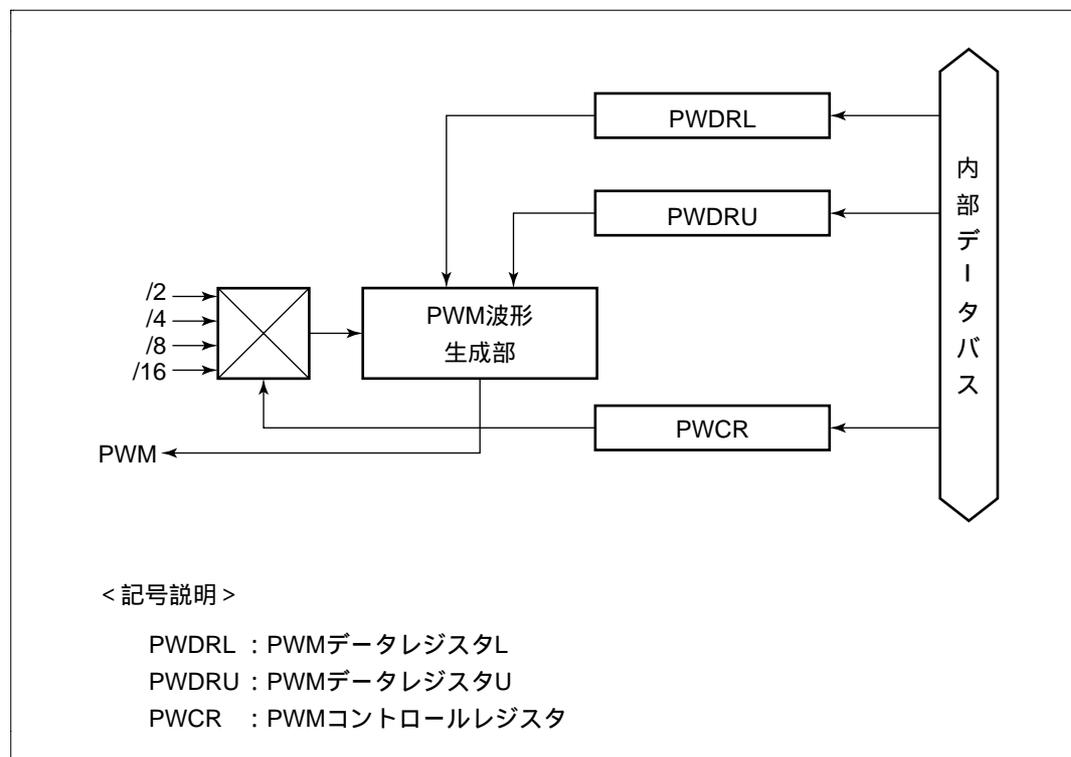


図 11.1 14ビットPWMのブロック図

11.1.3 端子構成

14ビットPWM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
PWM出力端子	PWM	出力	パルス分割方式PWM波形出力

11.1.4 レジスタ構成

14ビットPWM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
PWMコントロールレジスタ	PWCR	W	H'FC	H'FFD0
PWMデータレジスタU	PWDRU	W	H'C0	H'FFD1
PWMデータレジスタL	PWDRL	W	H'00	H'FFD2
クロック停止レジスタ	CKSTPR2	R/W	H'FF	H'FFFB

11.2 各レジスタの説明

11.2.1 PWM コントロールレジスタ (PWCR)

ビット:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PWCR1	PWCR0
初期値:	1	1	1	1	1	1	0	0
R/W :	—	—	—	—	—	—	W	W

PWCRは、8ビットのライト専用レジスタで、入力クロックの選択を行います。

リセット時、PWCRはHFCにイニシャライズされます。

ビット7~2: 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1~0: クロックセレクト1~0 (PWCR1~0)

14ビットPWMに供給されるクロックを選択します。

本ビットはライト専用です。リードすると常に"1"が読み出されます。

ビット1	ビット0	説明
PWCR1	PWCR0	
0	0	入力クロック = $t^*/2$ (初期値) 1変換周期 $16,384/t^*$ 、最小変化幅 $1/t^*$ のPWM波形を生成
0	1	入力クロック = $t^*/4$ 1変換周期 $32,768/t^*$ 、最小変化幅 $2/t^*$ のPWM波形を生成
1	0	入力クロック = $t^*/8$ 1変換周期 $65,536/t^*$ 、最小変化幅 $4/t^*$ のPWM波形を生成
1	1	入力クロック = $t^*/16$ 1変換周期 $131,072/t^*$ 、最小変化幅 $8/t^*$ のPWM波形を生成

【注】 * t : PWM入力クロックの周期

11.2.2 PWM データレジスタ U、L (PWDRU、PWDRL)

ビット:	7	6	5	4	3	2	1	0
PWDRU	—	—	PWDRU5	PWDRU4	PWDRU3	PWDRU2	PWDRU1	PWDRU0
初期値:	1	1	0	0	0	0	0	0
R/W :	—	—	W	W	W	W	W	W

ビット:	7	6	5	4	3	2	1	0
PWDRL	PWDRL7	PWDRL6	PWDRL5	PWDRL4	PWDRL3	PWDRL2	PWDRL1	PWDRL0
初期値:	0	0	0	0	0	0	0	0
R/W :	W	W	W	W	W	W	W	W

PWDRU、PWDRL は、ライト専用の 14 ビットのレジスタで、PWDRU が上位 6 ビット、PWDRL が下位 8 ビットの構成になっています。PWDRU、PWDRL に書込まれた内容は PWM 波形 1 周期の "High" レベル幅の合計に対応します。

PWDRU、PWDRL に 14 ビットのデータをライトすると、PWDRU、PWDRL の内容が PWM 波形生成部に取り込まれ、PWM 波形生成のデータの更新が行われます。14 ビットデータの設定は必ず PWDRL PWDRU の順序で行ってください。

- (1) PWDRL へ下位 8 ビットのデータをライトする。
- (2) PWDRU へ上位 6 ビットのデータをライトする。

PWDRU、PWDRL は、ライト専用レジスタです。リードした場合各ビットは常に "1" が読み出されます。

リセット時、PWDRU、PWDRL は H'C000 にイニシャライズされます。

11.2.3 クロック停止レジスタ 2 (CKSTPR2)

ビット:	7	6	5	4	3	2	1	0
	—	—	—	—	AECKSTP	WDCKSTP	PWCKSTP	LDCKSTP
初期値:	1	1	1	1	1	1	1	1
R/W :	—	—	—	—	R/W	R/W	R/W	R/W

CKSTPR2 は、8 ビットのリード/ライト可能なレジスタで、モジュールのモジュールスタンバイモードの制御を行います。本章では PWM に関するビットのみ説明します。他のビットについては各モジュールの章を参照してください。

ビット 1: PWM モジュールスタンバイモード制御 (PWCKSTP)

PWM をモジュールスタンバイモードに設定、および解除を制御します。

PWCKSTP	説明
0	PWM はモジュールスタンバイモードに設定される
1	PWM のモジュールスタンバイモードは解除される (初期値)

11.3 動作説明

11.3.1 動作説明

14ビットPWMを使用する場合、以下の順序でレジスタ設定を行ってください。

- (1) PMR3のPWM="1"としてP3₀/PWM端子をPWM出力端子に設定します。
- (2) PWCRにより、1変換周期を 131,072/ τ (PWCR1="1"、PWCR0="1")、65,536/ τ (PWCR1="1"、PWCR0="0")、32,768/ τ (PWCR1="0"、PWCR0="1")、16,384/ τ (PWCR1="0"、PWCR0="0")から選択します。
- (3) PWDRU、PWDRLに出力波形データを設定します。このとき、必ずPWDRL → PWDRUの順序で書込んでください。PWDRUへのライトと同時にPWM波形生成部にデータが取り込まれ、内部信号と同期をとってPWM波形生成の更新が行われます。

1変換周期は図11.2に示すように64個のパルスで構成され、この1変換周期中の"High"レベル幅合計(T_H)が、PWDRU、PWDRLのデータに対応しています。

この関係は次式で示されます。

$$T_H = (\text{PWDRU、PWDRLのデータ値} + 64) \times t / 2$$

ここで t は、PWM入力クロックの周期で 2/ τ (PWCR=H'0)、4/ τ (PWCR=H'1)、8/ τ (PWCR=H'2)または16/ τ (PWCR=H'3)となります。

(例) 変換周期を32,768 μ sとするためには、以下のように設定します。

PWCR1="0"、PWCR0="0"に設定すると、1変換周期は16,384/ τ なので、 $f_{\text{PWM}} = 0.5\text{MHz}$ となります。このとき、 $t_{\text{in}} = 512\mu\text{s}$ 、1/ τ (精度) = 2.0 μs です。

PWCR1="0"、PWCR0="1"に設定すると、1変換周期は32,768/ τ なので、 $f_{\text{PWM}} = 1\text{MHz}$ となります。このとき、 $t_{\text{in}} = 512\mu\text{s}$ 、2/ τ (精度) = 2.0 μs です。

PWCR1="1"、PWCR0="0"に設定すると、1変換周期は65,536/ τ なので、 $f_{\text{PWM}} = 2\text{MHz}$ となります。このとき、 $t_{\text{in}} = 512\mu\text{s}$ 、4/ τ (精度) = 2.0 μs です。

したがって、1変換周期32,768 μ sとするためには、システムクロック(f_{clk})は0.5MHz、1MHz、または2MHzで使用するようになります。

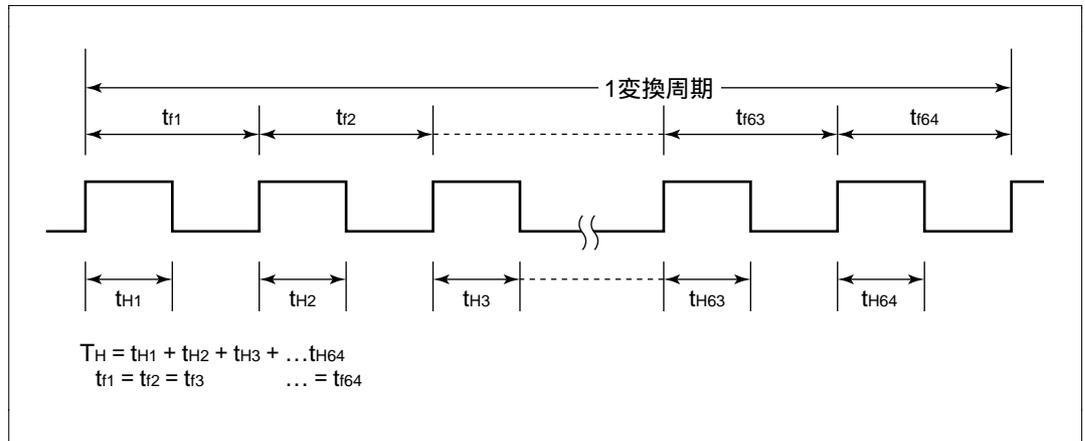


図 11.2 PWM 出力波形

11.3.2 PWM の動作モード

PWM の動作モードを表 11.3 に表示します。

表 11.3 PWM 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	モジュール スタンバイ
PWCR	リセット	動作	動作	保持	保持	保持	保持	保持
PWDRU	リセット	動作	動作	保持	保持	保持	保持	保持
PWDRL	リセット	動作	動作	保持	保持	保持	保持	保持

12. A/D 変換器

第 12 章 目次

12.1	概要.....	349
	12.1.1 特長.....	349
	12.1.2 ブロック図.....	350
	12.1.3 端子構成.....	351
	12.1.4 レジスタ構成.....	351
12.2	各レジスタの説明.....	352
	12.2.1 A/D リザルトレジスタ (ADRRH、ADRRL).....	352
	12.2.2 A/D モードレジスタ (AMR).....	352
	12.2.3 A/D スタートレジスタ (ADSR).....	354
	12.2.4 クロック停止レジスタ 1 (CKSTPR1).....	355
12.3	動作説明.....	356
	12.3.1 A/D 変換動作.....	356
	12.3.2 外部トリガによる A/D 変換器の起動.....	356
	12.3.3 A/D 変換器の動作モード.....	356
12.4	割込み要因.....	357
12.5	使用例.....	358
12.6	使用上の注意.....	362

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12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。

10ビットの分解能

入力チャンネル：12チャンネル

変換時間：1チャンネル当たり15.5 μ s（2MHz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

モジュールスタンバイモードにより、未使用時はモジュール単体でスタンバイモードに設定可能

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。

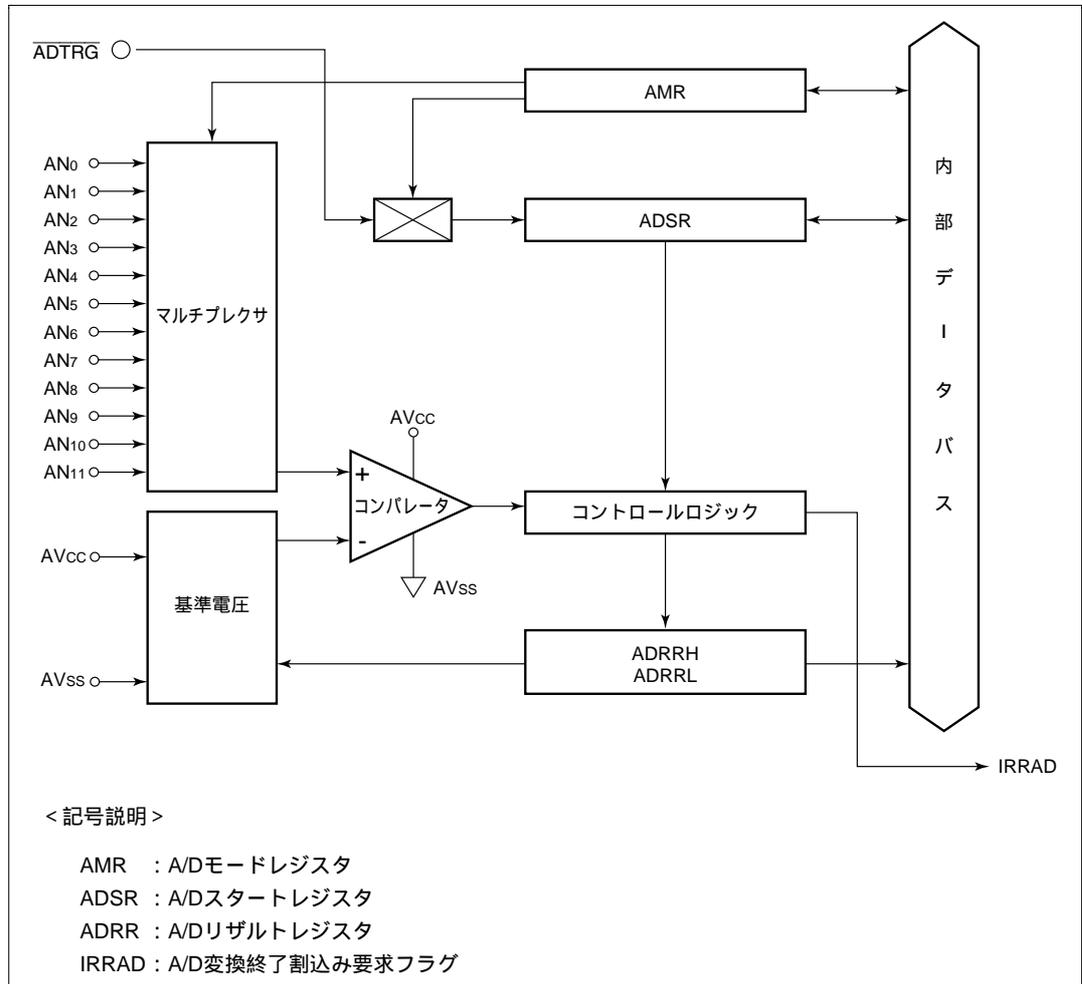


図 12.1 A/D 変換器ブロック図

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
アナログ電源端子	AV _{CC}	入力	アナログ部の電源および基準電圧
アナロググランド端子	AV _{SS}	入力	アナログ部のグランドおよび基準電圧
アナログ入力端子 0	AN ₀	入力	アナログ入力チャンネル 0
アナログ入力端子 1	AN ₁	入力	アナログ入力チャンネル 1
アナログ入力端子 2	AN ₂	入力	アナログ入力チャンネル 2
アナログ入力端子 3	AN ₃	入力	アナログ入力チャンネル 3
アナログ入力端子 4	AN ₄	入力	アナログ入力チャンネル 4
アナログ入力端子 5	AN ₅	入力	アナログ入力チャンネル 5
アナログ入力端子 6	AN ₆	入力	アナログ入力チャンネル 6
アナログ入力端子 7	AN ₇	入力	アナログ入力チャンネル 7
アナログ入力端子 8	AN ₈	入力	アナログ入力チャンネル 8
アナログ入力端子 9	AN ₉	入力	アナログ入力チャンネル 9
アナログ入力端子 10	AN ₁₀	入力	アナログ入力チャンネル 10
アナログ入力端子 11	AN ₁₁	入力	アナログ入力チャンネル 11
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'30	H'FFC6
A/Dスタートレジスタ	ADSR	R/W	H'7F	H'FFC7
A/Dリザルトレジスタ H	ADRRH	R	不定	H'FFC4
A/Dリザルトレジスタ L	ADRRL	R	不定	H'FFC5
クロック停止レジスタ 1	CKSTPRT1	R/W	H'FF	H'FFFA

12.2 各レジスタの説明

12.2.1 A/D リザルトレジスタ (ADRRH、ADRRL)

ビット:	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	ADR9	ADR8	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0						
初期値:	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定						
R/W :	R	R	R	R	R	R	R	R	R	R						
	ADRRH								ADRRL							

ADRRH と ADRRL は、A/D 変換された結果を格納する 16 ビットのリード専用レジスタです。ADRRH に上位 8 ビット、ADRRL に下位 2 ビットが格納されます。

ADRRH と ADRRL は常に CPU からリード可能です。A/D 変換中は ADRRH と ADRRL の値は不定で、A/D 変換終了時に変換結果の 10 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。

ADRRH と ADRRL 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	—	—	CH3	CH2	CH1	CH0
初期値:	0	0	1	1	0	0	0	0
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'30 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)

A/D 変換スピードの設定を行います。

ビット7	変換周期	変換時間	
		= 1MHz	= 2MHz
CKS			
0	62/ (初期値)	62μs	31μs
1	31/	31μs	15.5μs*

【注】 * 15.5μs 以下の変換時間では、動作が保証されません。15.5μs 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 INTEG4 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5～4：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	0	0	AN ₀
0	1	0	1	AN ₁
0	1	1	0	AN ₂
0	1	1	1	AN ₃
1	0	0	0	AN ₄
1	0	0	1	AN ₅
1	0	1	0	AN ₆
1	0	1	1	AN ₇
1	1	0	0	AN ₈
1	1	0	1	AN ₉
1	1	1	0	AN ₁₀
1	1	1	1	AN ₁₁

* : 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 ADDRH と ADDRRL 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。

12.2.4 クロック停止レジスタ1 (CKSTPR1)

ビット:	7	6	5	4	3	2	1	0
	S1CKSTP	S31CKSTP	S32CKSTP	ADCKSTP	TGCKSTP	TFCKSTP	TCCKSTP	TACKSTP
初期値:	1	1	1	1	1	1	1	1
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CKSTPR1 は、8ビットのリード/ライト可能なレジスタで、モジュールのモジュールスタンバイモードの制御を行います。本章ではA/D変換器に関するビットのみ説明します。他のビットについては各モジュールの章を参照してください。

ビット4： A/D変換器モジュールスタンバイモード制御 (ADCKSTP)

A/D変換器をモジュールスタンバイモードに設定、および解除を制御します。

ADCKSTP	説明
0	A/D変換器はモジュールスタンバイモードに設定される
1	A/D変換器のモジュールスタンバイモードは解除される (初期値)

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、10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変換終了割り込みが発生します。

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 PMR1 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。

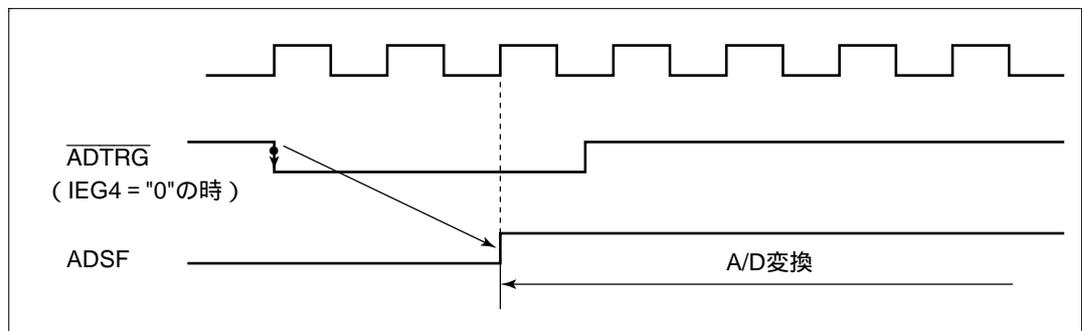


図 12.2 外部トリガ入力タイミング

12.3.3 A/D変換器の動作モード

A/D変換器の動作モードを表 12.3 に示します。

表 12.3 A/D変換器の動作モード

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	モジュール スタンバイ
AMR	リセット	動作	動作	保持	保持	保持	保持	保持
ADSR	リセット	動作	動作	保持	保持	保持	保持	保持
ADRRH	保持*	動作	動作	保持	保持	保持	保持	保持
ADRRL	保持*	動作	動作	保持	保持	保持	保持	保持

【注】 * パワーオンリセット時は不定

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1 (AN1) 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_i (AMRのCH3~CH0を"0101")、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ADRRHとADRRL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変換終了割込み要求が発生します。
- (4) A/D割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。

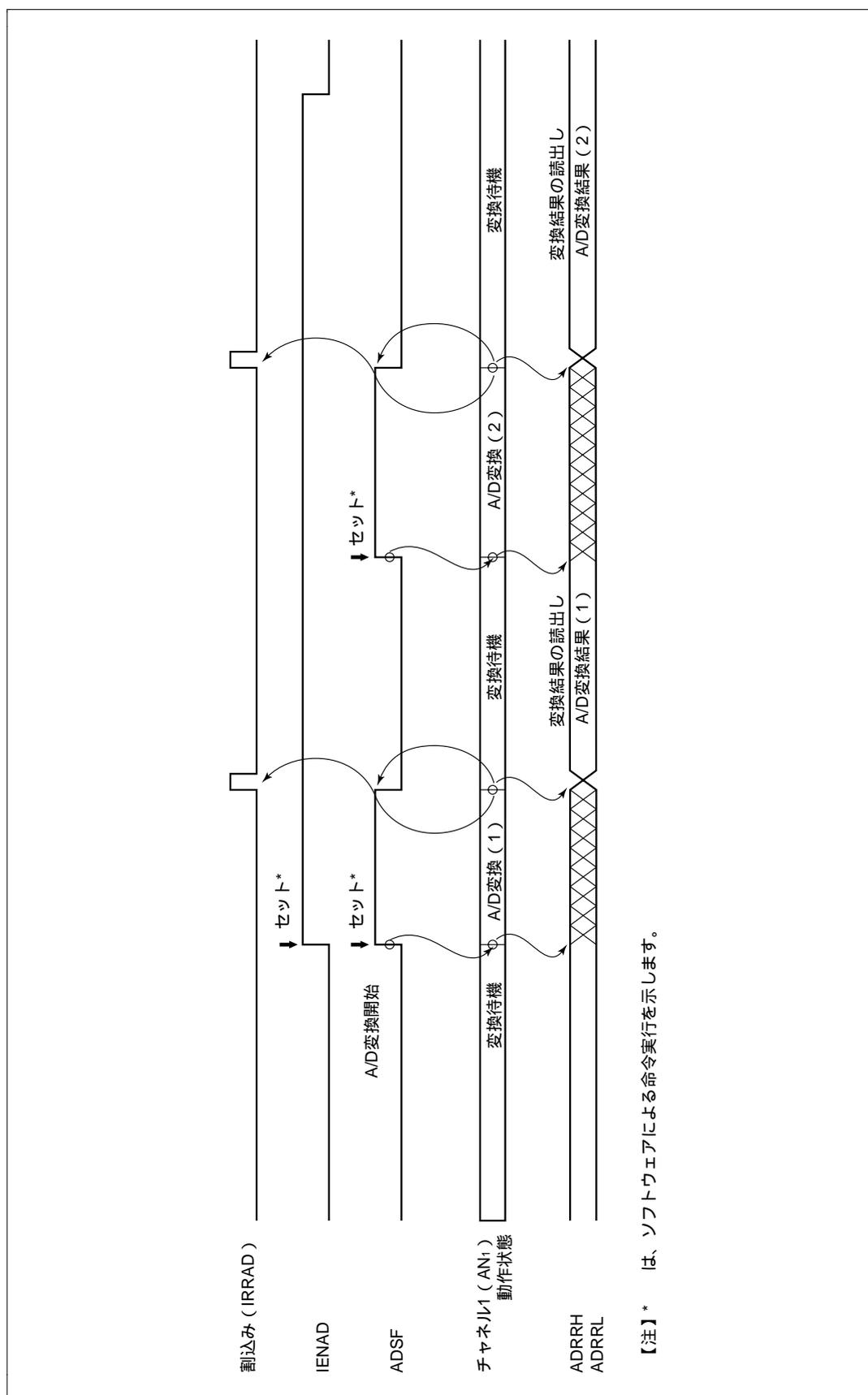


図 12.3 A/D変換器の動作例

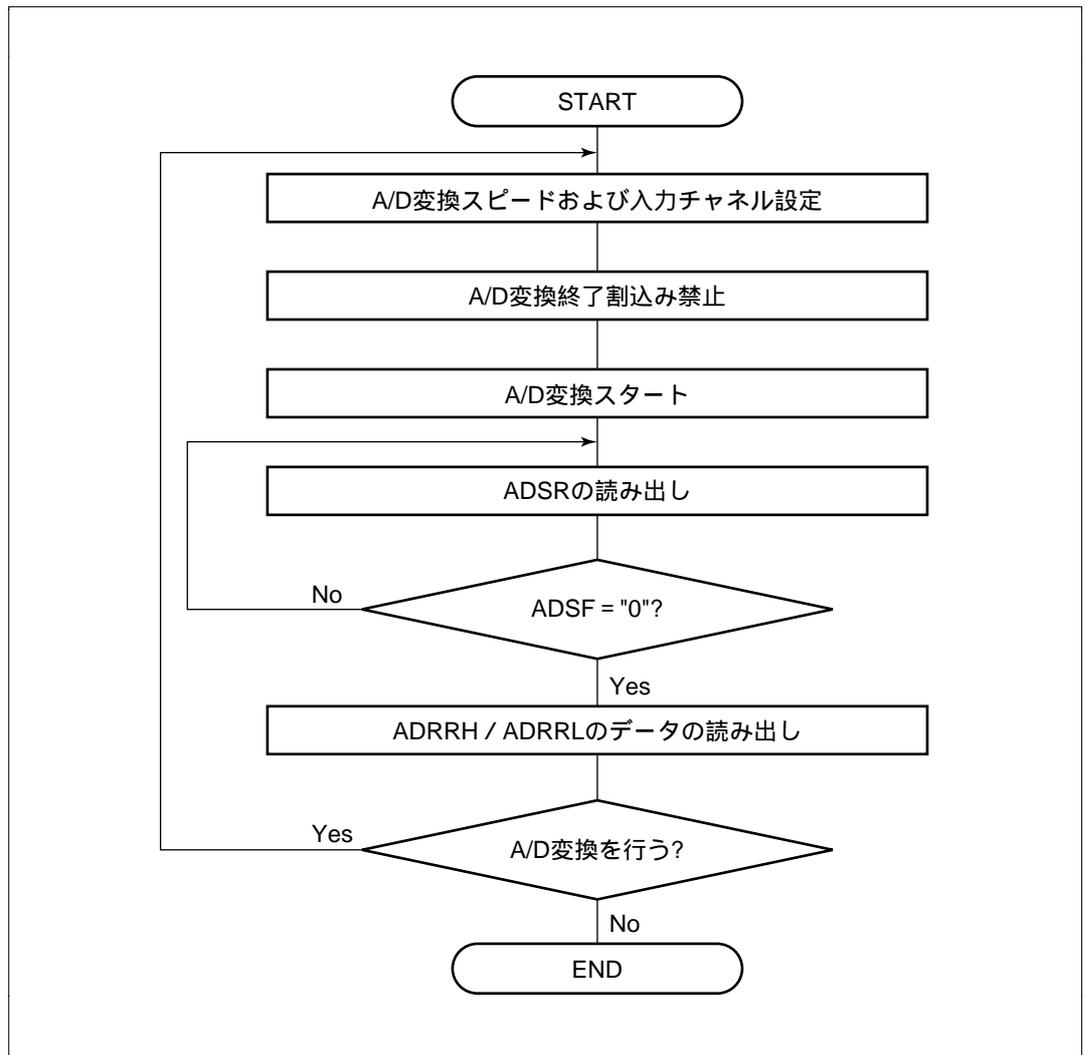


図 12.4 A/D 変換器の使用手順の概念フロー
 (ソフトウェアでポーリングする場合)

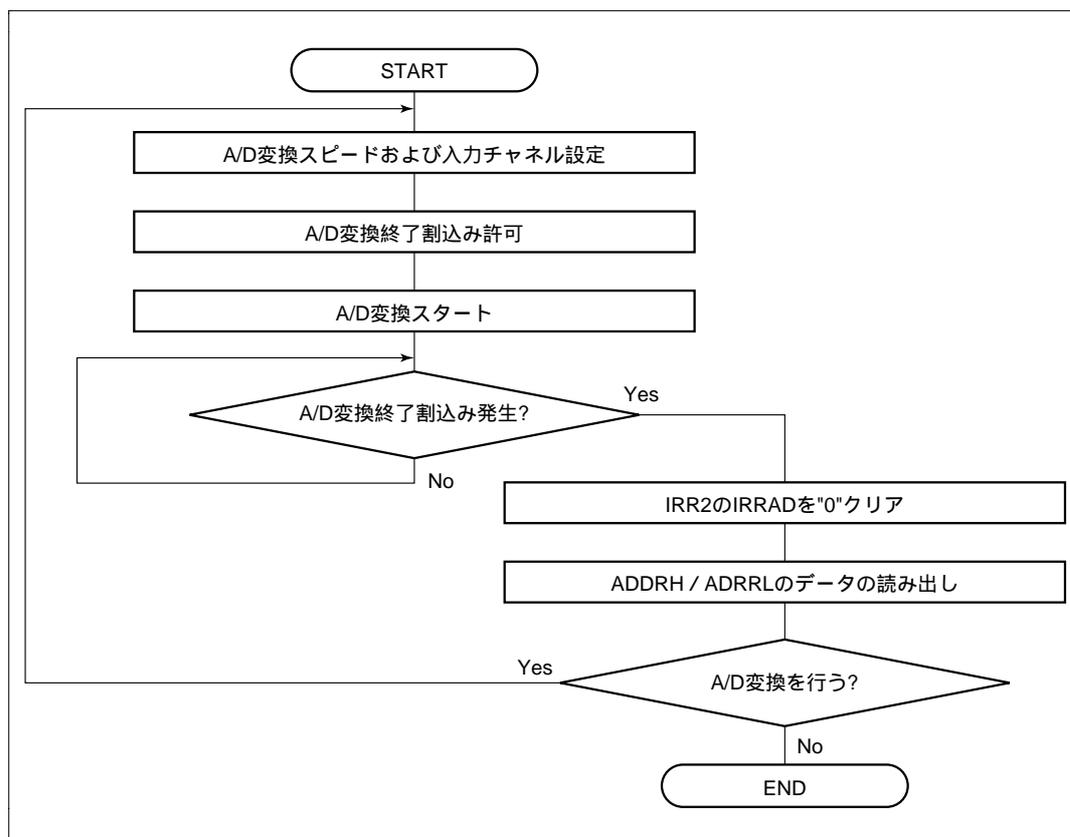


図 12.5 A/D 変換器の使用手順の概念フロー
(割り込みを使用する場合)

12.6 使用上の注意

- (1) ADRRH と ADRRL 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。
- (3) モジュールスタンバイモードを解除後、A/D変換を開始する場合は、10 クロック待ってから A/D変換をスタートしてください。

13. LCD コントローラ / ドライバ

第 13 章 目次

13.1	概要.....	365
	13.1.1 特長.....	365
	13.1.2 ブロック図.....	366
	13.1.3 端子構成.....	367
	13.1.4 レジスタ構成.....	367
13.2	各レジスタの説明.....	368
	13.2.1 LCD ポートコントロールレジスタ (LPCR).....	368
	13.2.2 LCD コントロールレジスタ (LCR).....	370
	13.2.3 LCD コントロールレジスタ 2 (LCR2).....	372
	13.2.4 クロック停止レジスタ 2 (CKSTPR2).....	374
13.3	動作説明.....	375
	13.3.1 LCD 表示までのセッティング.....	375
	13.3.2 LCD RAM と表示の関係.....	377
	13.3.3 輝度調整機能 (V ₀ 端子).....	382
	13.3.4 昇圧定電圧 (5V) 電源.....	383
	13.3.5 低消費電力 LCD 駆動方式.....	383
	13.3.6 低消費電力モード時の動作.....	387
	13.3.7 LCD 駆動電源の強化.....	388
	13.3.8 HD66100 との接続.....	388

13.1 概要

本 LSI は、セグメントタイプの LCD コントロール回路と LCD ドライバと電源回路を内蔵しているため、LCD パネルを直接駆動することができます。

13.1.1 特長

LCD コントローラ/ドライバの特長を以下に示します。

表示容量

デューティ比	内部ドライバ	セグメント外部拡張ドライバ
スタティック	40SEG	256SEG
1/2	40SEG	128SEG
1/3	40SEG	64SEG
1/4	40SEG	64SEG

LCD RAM 容量

8 ビット×32 バイト (256 ビット)

LCD RAM はワードアクセス可能

セグメント出力端子を 8 端子ごとにポートとして使用可能

デューティ比により使用しないコモン出力端子をコモンダブルバッファ用 (並列接続用) として使用可能

スタンバイモード以外の動作モードで表示可能

フレーム周波数を 11 種類より選択可能

電源分割抵抗を内蔵し、LCD 駆動電源を供給

モジュールスタンバイモードにより、未使用時はモジュール単体でスタンバイモードに設定可能

昇圧定電圧 (5V) 電源内蔵により、低電圧時でも LCD の表示可能 (H8/3887 シリーズ)

ソフトウェアにより A 波形、B 波形の選択可能

13.1.2 ブロック図

LCD コントローラ/ドライバのブロック図を図 13.1 に示します。

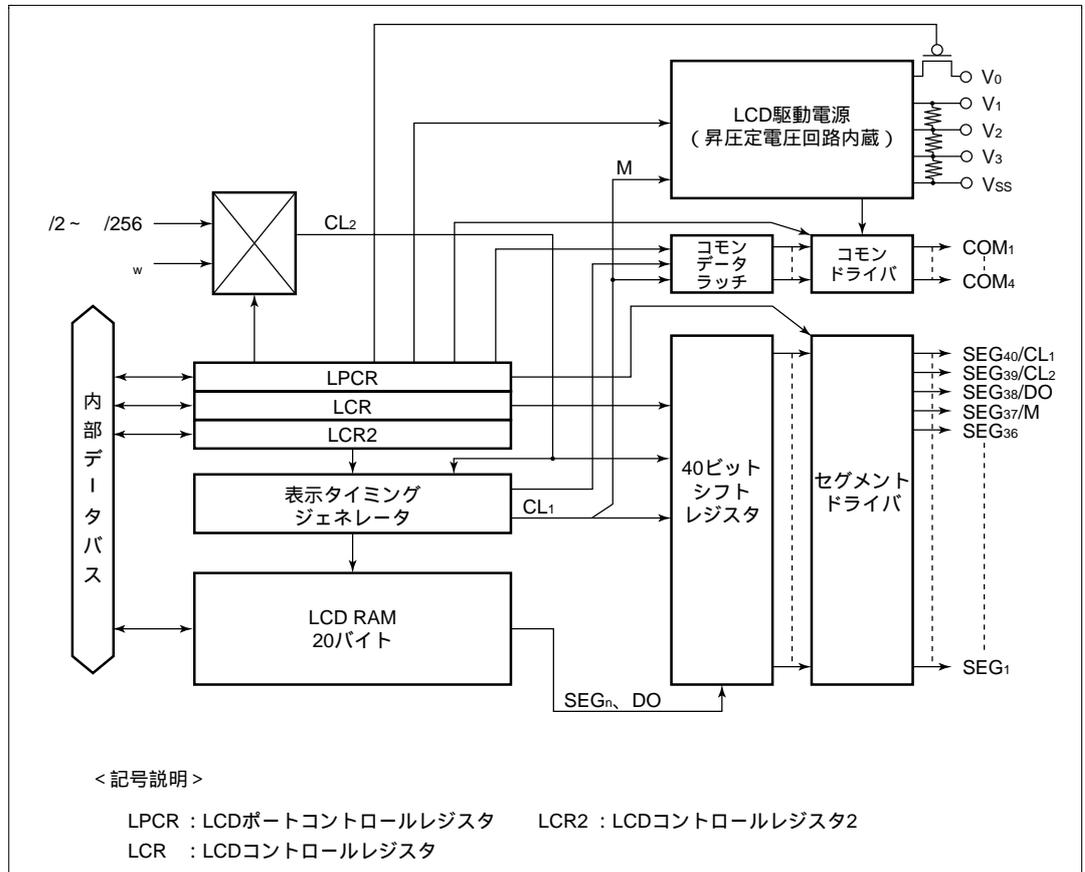


図 13.1 LCD コントローラ/ドライバのブロック図

13.1.3 端子構成

LCD コントローラ/ドライバの端子構成を表 13.1 に示します。

表 13.1 端子構成

端子名	記号	入出力	機能
セグメント出力端子	SEG ₄₀ ~ SEG ₁	出力	液晶のセグメント駆動用端子 全端子、ポートと兼用でプログラマブルに設定可能
コモン出力端子	COM ₄ ~ COM ₁	出力	液晶のコモン駆動端子 Static、1/2 デューティ時には端子の並列化が可能
セグメント 外部拡張信号端子	CL ₁	出力	表示データラッチクロック、SEG ₄₀ と兼用
	CL ₂	出力	表示データシフトクロック、SEG ₃₉ と兼用
	M	出力	LCD 交流化信号、SEG ₃₇ と兼用
	DO	出力	シリアル表示データ、SEG ₃₈ と兼用
LCD 電源端子	V ₀ 、V ₁ 、V ₂ 、 V ₃		外付けでバスコンを接続する場合、外部電源回路を使用する場合に使用

13.1.4 レジスタ構成

LCD コントローラ/ドライバのレジスタ構成を表 13.2 に示します。

表 13.2 レジスタ構成

名称	略称	R/W	初期値	アドレス
LCD ポートコントロールレジスタ	LPCR	R/W	H'00	H'FFC0
LCD コントロールレジスタ	LCR	R/W	H'80	H'FFC1
LCD コントロールレジスタ 2	LCR2	R/W	H'60	H'FFC2
LCD RAM		R/W	不定	H'F740 ~ H'F753
クロック停止レジスタ 2	CKSTPR2	R/W	H'FF	H'FFFB

13.2 各レジスタの説明

13.2.1 LCD ポートコントロールレジスタ (LPCR)

ビット:	7	6	5	4	3	2	1	0
	DTS1	DTS0	CMX	SGX	SGS3	SGS2	SGS1	SGS0
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

LPCR は、8 ビットのリード/ライト可能なレジスタで、デューティ比の選択、LCD ドライバや端子機能の選択を行います。

リセット時、LPCR は H'00 にイニシャライズされます。

ビット7~5：デューティ比選択1、0 (DTS1、DTS0)、コモン機能選択 (CMX)

DTS1、DTS0 の組合せで、スタティック、1/2 ~ 1/4 デューティのいずれかを選択します。CMX は、デューティによって使用しないコモン端子をコモンドライブ能力を大きくするために複数の端子から同じ波形を出力するか否かを選択します。

ビット7	ビット6	ビット5			補足説明
DTS1	DTS0	CMX	デューティ比	コモンドライブ	
0	0	0	スタティック	COM ₁ (初期値)	COM ₄ 、COM ₃ 、COM ₂ は使用しないでください。
0	0	1		COM ₄ ~ COM ₁	COM ₄ 、COM ₃ 、COM ₂ はCOM ₁ と同じ波形が出力。
0	1	0	1/2 デューティ	COM ₂ ~ COM ₁	COM ₄ 、COM ₃ は使用しないでください。
0	1	1		COM ₄ ~ COM ₁	COM ₄ はCOM ₃ 、COM ₂ はCOM ₁ と同じ波形が出力。
1	0	0	1/3 デューティ	COM ₃ ~ COM ₁	COM ₄ は使用しないでください。
1	0	1		COM ₄ ~ COM ₁	COM ₄ は使用しないでください。
1	1	0	1/4 デューティ	COM ₄ ~ COM ₁	
1	1	1			

ビット4：拡張信号選択 (SGX)

SGX は SEG₄₀/CL₁、SEG₃₉/CL₂、SEG₃₈/DO、SEG₃₇/M 端子をセグメント端子 (SEG₄₀ ~ SEG₃₇) として使用するか、またはセグメント外部拡張信号端子 (CL₁、CL₂、DO、M) として使用するかを選択します。

ビット4	説明
SGX	
0	SEG ₄₀ ~ SEG ₃₇ 端子* (初期値)
1	CL ₁ 、CL ₂ 、DO、M 端子

【注】 * SGS3 ~ SGS0 が "0000" のときはポートとして機能します。

ビット3 ~ 0：セグメントドライバ選択 3 ~ 0 (SGS3 ~ SGS0)

SGS3 ~ SGS0 は使用するセグメントドライバを選択します。

ビット4	ビット3	ビット2	ビット1	ビット0	SEG ₄₀ ~ SEG ₁ 端子の機能					補足説明
					SEG ₄₀ ~ SEG ₃₃	SEG ₃₂ ~ SEG ₂₅	SEG ₂₄ ~ SEG ₁₇	SEG ₁₆ ~ SEG ₉	SEG ₈ ~ SEG ₁	
0	0	0	0	0	ポート	ポート	ポート	ポート	ポート	(初期値)
0	0	0	0	1	SEG	ポート	ポート	ポート	ポート	
0	0	0	1	*	SEG	SEG	ポート	ポート	ポート	
0	0	1	0	*	SEG	SEG	SEG	ポート	ポート	
0	0	1	1	*	SEG	SEG	SEG	SEG	ポート	
0	1	*	*	*	SEG	SEG	SEG	SEG	SEG	
1	0	0	0	0	ポート*1	ポート	ポート	ポート	ポート	
1	0	0	0	1	使用禁止					
1	0	0	1	*						
1	0	1	*	*						
1	1	*	*	*						

* : Don't care

【注】 *1 SEG₄₀ ~ SEG₃₇ は外部拡張端子

13.2.2 LCD コントロールレジスタ (LCR)

ビット:	7	6	5	4	3	2	1	0
	—	PSW	ACT	DISP	CKS3	CKS2	CKS1	CKS0
初期値:	1	0	0	0	0	0	0	0
R/W :	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

LCRは、8ビットのリード/ライト可能なレジスタで、LCD 駆動電源 ON/OFF 制御、表示データの制御、フレーム周波数の選択を行います。

リセット時、LCRはH'80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: LCD 駆動電源 ON/OFF 制御 (PSW)

低消費電力モードで LCD 表示を必要としない場合、また外部電源を使用する場合に LCD 駆動電源を OFF 状態にすることができます。ACT を"0"とした場合、またスタンバイモード時には本ビットとは無関係に LCD 駆動電源が OFF 状態となります。

ビット6	
PSW	説明
0	LCD 駆動電源 OFF (初期値)
1	LCD 駆動電源 ON

ビット5: 表示機能開始 (ACT)

LCD コントローラ/ドライバを使用するかしないかを選択します。本ビットを"0"にクリアすることにより、LCD コントローラ/ドライバは動作を停止します。また、PSW の値と無関係に LCD 駆動電源が OFF 状態になります。ただし、レジスタの内容は保持されます。

ビット5	
ACT	説明
0	LCD コントローラ/ドライバ動作停止 (初期値)
1	LCD コントローラ/ドライバ動作

ビット4：表示データ制御 (DISP)

DISP はLCD RAM の内容を表示するかLCD RAM の内容に関係なくブランクデータを表示するかを選択します。

ビット4	説明	
DISP		
0	ブランクデータを表示	(初期値)
1	LCD RAM データを表示	

ビット3～0：フレーム周波数選択3～0 (CKS3～CKS0)

使用クロックの選択とフレーム周波数の選択を行います。サブアクティブモード、ウォッチモード、サブスリープモードではシステムクロック () が停止するので、 $/2 \sim /256$ を選択している場合は表示動作を行いません。これらのモードにおいて LCD 表示を行う場合は、必ず使用クロックとして w 、 $w/2$ または $w/4$ を選択するようにしてください。

ビット3	ビット2	ビット1	ビット0	使用クロック	フレーム周波数*2	
					= 2MHz	= 250kHz*1
0	*	0	0	w	128Hz*3 (初期値)	
0	*	0	1	$w/2$	64Hz*3	
0	*	1	*	$w/4$	32Hz*3	
1	0	0	0	$/2$		244Hz
1	0	0	1	$/4$	977Hz	122Hz
1	0	1	0	$/8$	488Hz	61Hz
1	0	1	1	$/16$	244Hz	30.5Hz
1	1	0	0	$/32$	122Hz	
1	1	0	1	$/64$	61Hz	
1	1	1	0	$/128$	30.5Hz	
1	1	1	1	$/256$		

* : Don't care

【注】 *1 = 2MHz 時のアクティブ (中速 $f_{osc}/16$) モードのフレーム周波数です。

*2 1/3 デューティ選択時は、フレーム周波数が表で示した値の 4/3 倍となります。

*3 $w = 32.768\text{kHz}$ 時のフレーム周波数です。

13.2.3 LCD コントロールレジスタ 2 (LCR2)

ビット:	7	6	5	4	3	2	1	0
	LCDAB	—	—	SUPS	CDS3	CDS2	CDS1	CDS0
初期値:	0	1	1	0	0	0	0	0
R/W :	R/W	—	—	R/W	R/W	R/W	R/W	R/W

LCR2は、8ビットのリード/ライト可能なレジスタで、A波形/B波形切り替えの制御、駆動電源の選択、昇圧定電圧(5V)電源の制御、電源分割抵抗を電源回路から切り離しの制御をする充放電パルスのデューティ比選択を行います。リセット時、LCR2はH'60にイニシャライズされます。

ビット7: A波形/B波形切換え制御 (LCDAB)

LCDの駆動波形をA波形にするかB波形にするかを選択します。

ビット7		
LCDAB	説明	
0	A波形で駆動	(初期値)
1	B波形で駆動	

ビット6、5: リザーブビット

本ビットはリザーブビットで、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: 駆動電源の選択、昇圧定電圧(5V)電源の制御 (SUPS) (H8/3887シリーズのみに適用)

駆動電源として V_{CC} を選択すると同時に昇圧定電圧(5V)電源は動作を停止し、駆動電源として5Vを選択すると同時に昇圧定電圧(5V)電源は動作します。

ビット4		
SUPS	説明	
0	駆動電源は V_{CC} 、昇圧定電圧(5V)電源動作停止	(初期値)
1	駆動電源は5V、昇圧定電圧(5V)電源動作	

ビット3、2、1、0：充放電パルスのデューティ比選択（CDS3～CDS0）

ビット3	ビット2	ビット1	ビット0		
CDS3	CDS2	CDS1	CDS0	デューティ比	補足説明
0	0	0	0	1	High 固定 (初期値)
0	0	0	1	1/8	
0	0	1	0	2/8	
0	0	1	1	3/8	
0	1	0	0	4/8	
0	1	0	1	5/8	
0	1	1	0	6/8	
0	1	1	1	0	Low 固定
1	0	*	*	1/16	
1	1	*	*	1/32	

* : Don't care

電源分割抵抗を電源回路に接続している期間のデューティ比選択を行います。

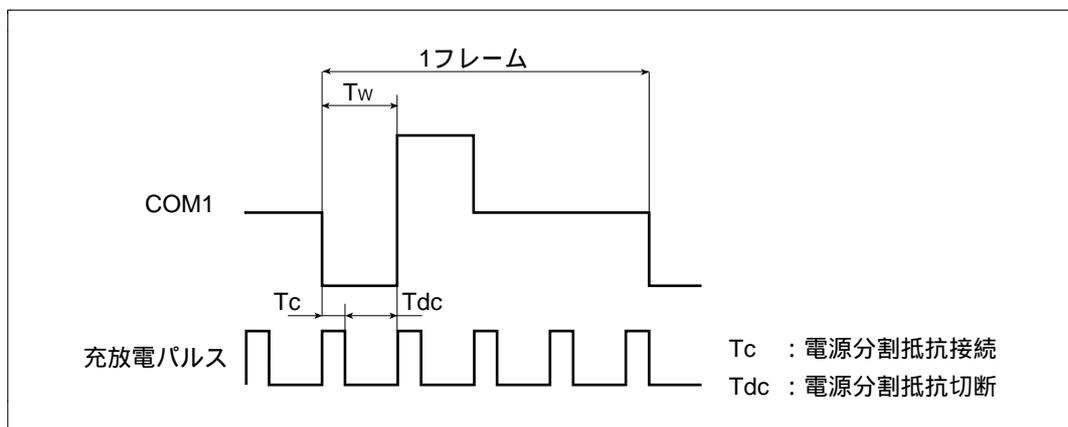
デューティ比0を選択した場合は電源分割抵抗が電源回路から切り離された状態に固定されますので、外部回路により V_1 、 V_2 、 V_3 端子に電源を供給してください。充放電パルスは図 13.2 に示す波形となり、デューティ比は T_c/T_w となります。

図 13.2 A 波形 1/2 デューティ 1/2 バイアスの例

13.2.4 クロック停止レジスタ 2 (CKSTPR2)

ビット:	7	6	5	4	3	2	1	0
	—	—	—	—	AECKSTP	WDCKSTP	PWCKSTP	LDCKSTP
初期値:	1	1	1	1	1	1	1	1
R/W :	—	—	—	—	R/W	R/W	R/W	R/W

CKSTPR2 は、8 ビットのリード/ライト可能なレジスタで、モジュールのモジュールスタンバイモードの制御を行います。本章では LCD に関するビットのみ説明します。他のビットについては各モジュールの章を参照してください。

ビット 0 : LCD モジュールスタンバイモード制御 (LDCKSTP)

LCD をモジュールスタンバイモードに設定、および解除を制御します。

ビット 0	説明
LDCKSTP	
0	LCD はモジュールスタンバイモードに設定される
1	LCD のモジュールスタンバイモードは解除される (初期値)

13.3 動作説明

13.3.1 LCD 表示までのセッティング

LCD 表示を行うにはハードウェア、ソフトウェアのそれぞれについて以下のことを決定しておく必要があります。

(1) ハードウェアのセッティング

(a) 1/2 デューティ使用

1/2 デューティで使用する際は V_2 、 V_3 端子を接続してください (図 13.3 参照)。

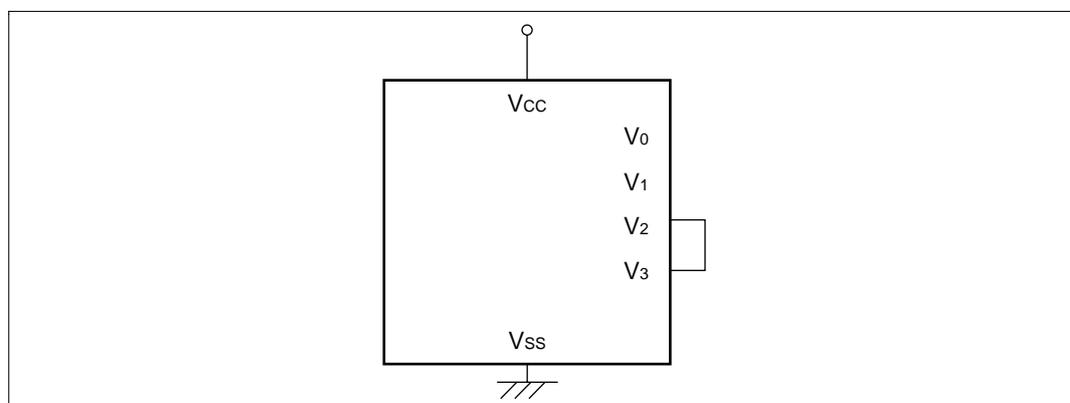


図 13.3 1/2 デューティ時の LCD 駆動電源の処理

(b) 大パネル表示

内蔵の電源分割抵抗はインピーダンスが大きいため、大きなパネルの駆動には適さないことがあります。大きなパネルを使用し表示が不鮮明になるときは「13.3.7 LCD 駆動電源の強化」を参照してください。また、スタティック、1/2 デューティを選択した場合、コモン出力の駆動能力を強化することができます。デューティ比の選択時に CMX を "1" にしてください。このモードではスタティック時に $COM_4 \sim COM_1$ 端子が同じ波形となり、1/2 デューティ時は COM_2 、 COM_1 端子から COM_1 波形が、 COM_4 、 COM_3 端子からは COM_2 波形が出力されます。

(c) 輝度調整機能 (V_0 端子)

V_0 端子と V_1 端子の間に抵抗を接続すると、輝度調整が可能となります。詳細は「13.3.3 輝度調整機能 (V_0 端子)」を参照してください。

(d) LCD 駆動電源の設定

本 LSI は LCD 駆動電源として内蔵の電源回路を使用する方法と、外部電源回路を使用する方法があります。また、H8/3887 シリーズでは、内蔵の電源回路を使用する場合には電源電圧 (V_{CC}) と昇圧定電圧 (5V) の選択が行えます。昇圧定電圧電源の詳細は「13.3.4 昇圧定電圧 (5V) 電源」を参照してください。

LCD 駆動電源として内蔵の電源回路を使用する場合は、 V_0 端子と V_1 端子を外部で接続してください。接続例を図 13.4 (a) に示します。

LCD 駆動電源として外部電源回路を使用する場合は、 V_1 端子に外部電源を接続し、 V_0 端子は外部で V_{CC} と短絡してください。接続例を図 13.4 (b) に示します。

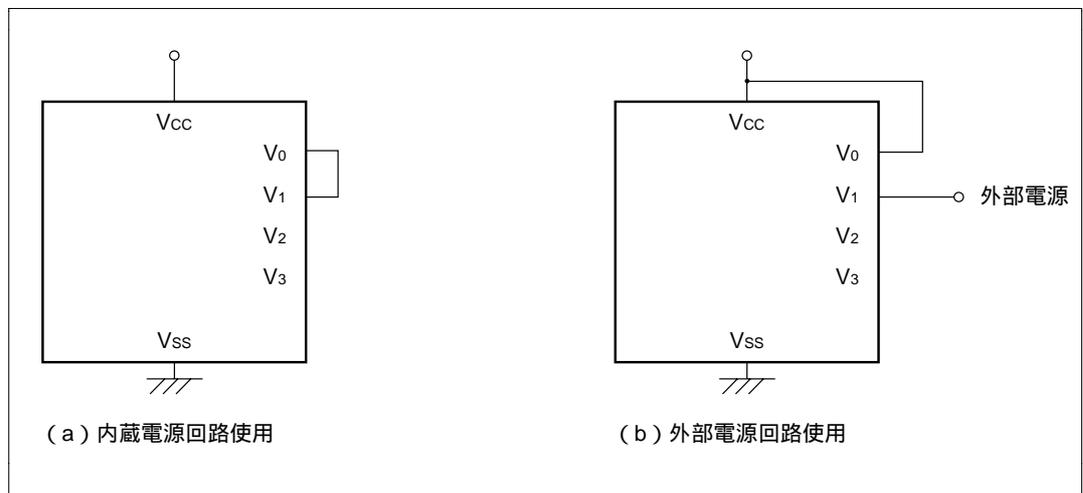


図 13.4 LCD 電源端子の接続例

(e) 低消費電力 LCD 駆動方式

低消費電力 LCD 駆動方式を使用すると、LCD 駆動に必要なとする消費電力を最適化することができます。詳細は「13.3.5 低消費電力 LCD 駆動方式」を参照してください。

(f) セグメント外部拡張

外部に HD66100 を接続してセグメント数を拡張することができます。詳細は「13.3.8 HD66100 との接続」を参照してください。

(2) ソフトウェアのセッティング

(a) デューティの選択

デューティは、DTS1、DTS0 によりスタティック、1/2 デューティ、1/3 デューティ、1/4 デューティから選択できます。

(b) セグメントドライバの選択

SGS₃ ~ SGS₀により、使用するセグメントドライバを選択できます。

(c) フレーム周波数の選択

CKS₃ ~ CKS₀を設定することでフレーム周波数を選択することができます。フレーム周波数はLCDパネルの指定にしたがって選択してください。ウォッチモード、サブアクティブモード、サブスリープモード時のクロックの選択方法は「13.3.6 低消費電力モード時の動作」を参照してください。

(d) A 波形、B 波形の選択

LCDABにより、使用するLCD波形をA波形かB波形のどちらかを選択できます。

(e) LCD 駆動電源の選択

内蔵の電源回路を使用する場合にはSUPSにより使用する電源を選択することができます。外部電源回路を使用する場合にはSUPSでV_{CC}を選択し、PSWでLCD駆動電源をOFF状態にしてください。

13.3.2 LCD RAM と表示の関係

LCD RAM と表示セグメントの関係は、デューティ比によって異なります。セグメント外部拡張しない場合のそれぞれのデューティ比に対応したLCD RAM のマップを図 13.5 ~ 図 13.8 に、セグメント外部拡張した場合のLCD RAM のマップを図 13.9 ~ 図 13.12 に示します。

表示に必要なレジスタ群を設定した後、デューティに対応する部分に通常のRAMと同様な命令によってデータを書込み、表示をONすれば自動的に表示を開始します。RAM設定にはワード/バイトアクセス命令が使用できます。

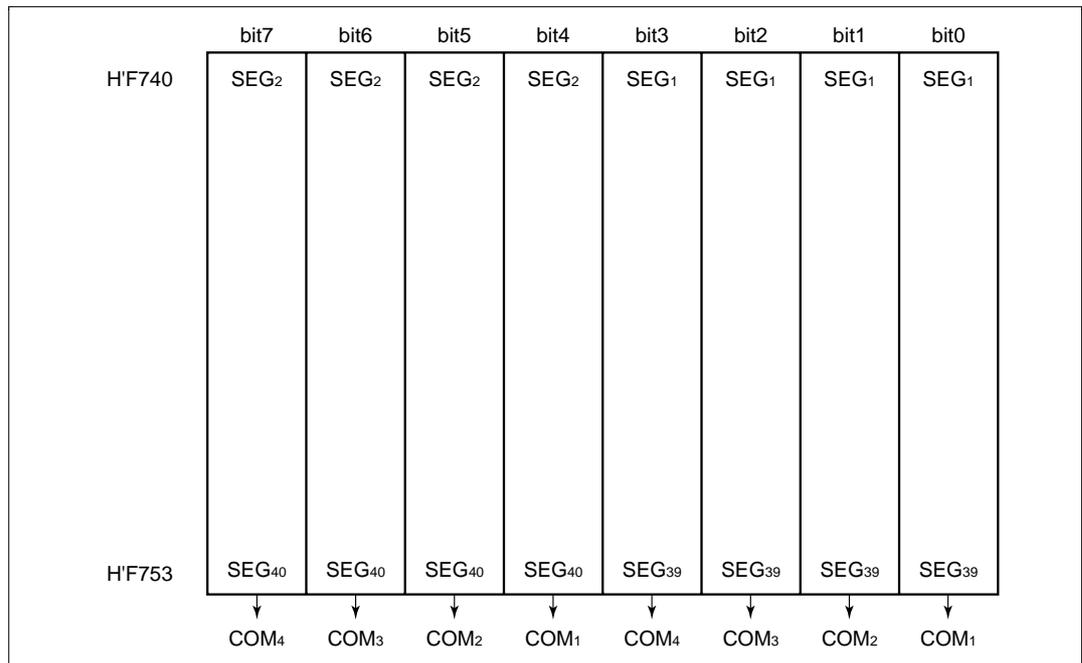


図 13.5 セグメント外部拡張しない場合の LCD RAM マップ (1/4 デューティ)

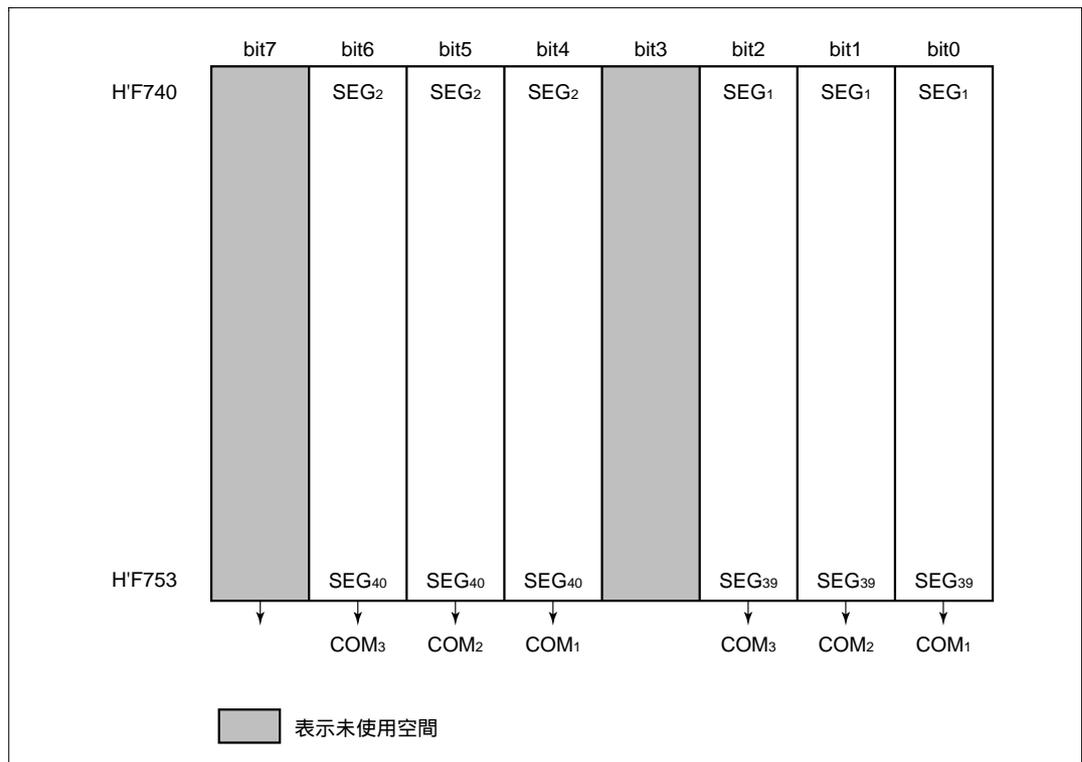


図 13.6 セグメント外部拡張しない場合の LCD RAM マップ (1/3 デューティ)

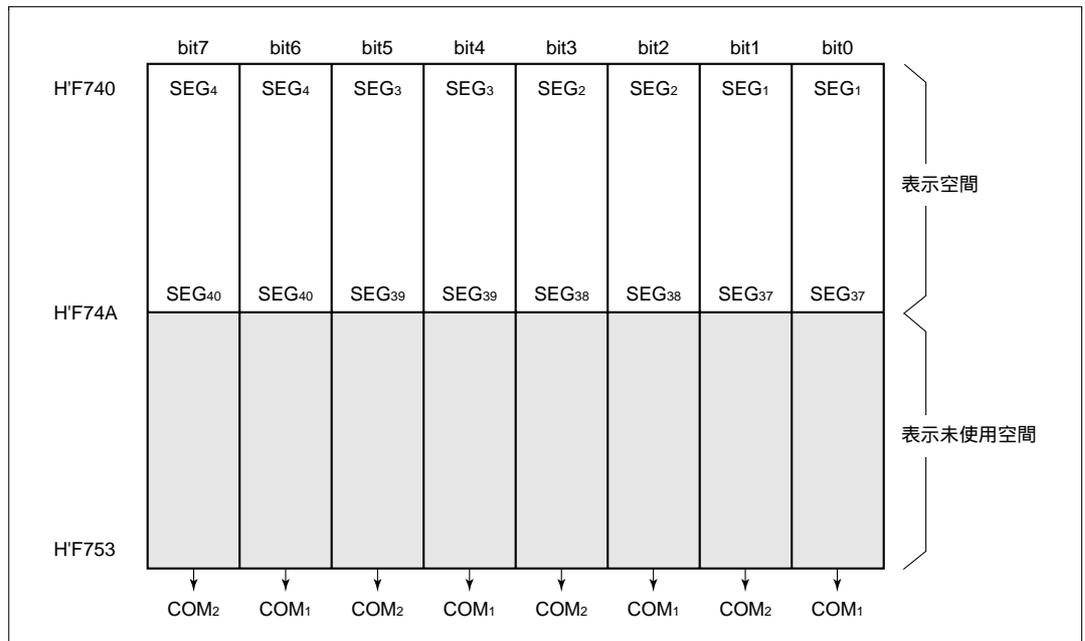


図 13.7 セグメント外部拡張しない場合の LCD RAM マップ (1/2 デューティ)

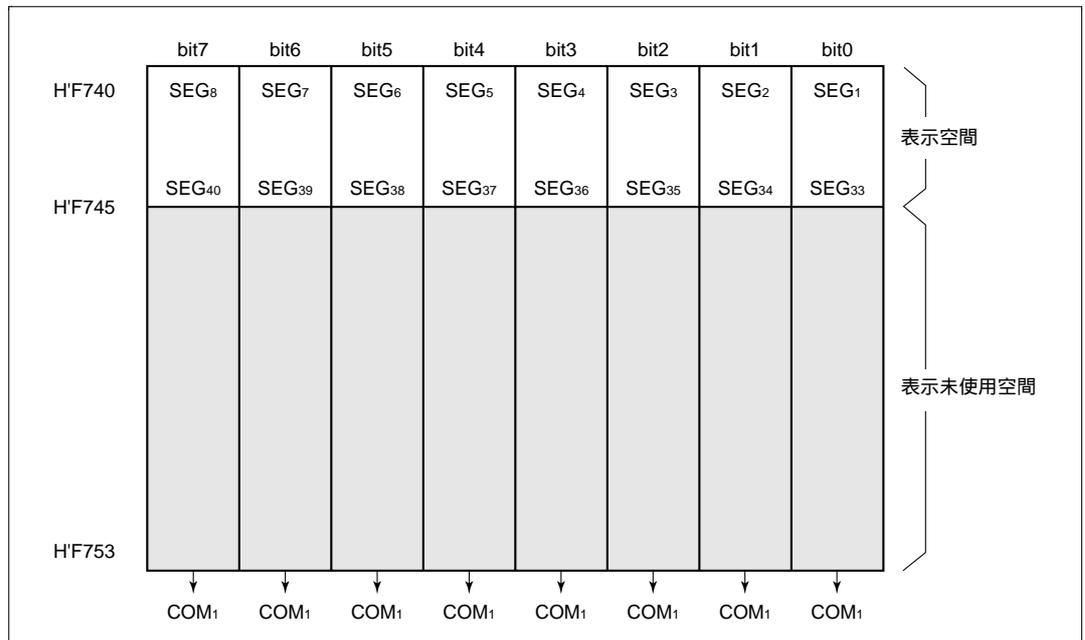


図 13.8 セグメント外部拡張しない場合の LCD RAM マップ (スタティック)

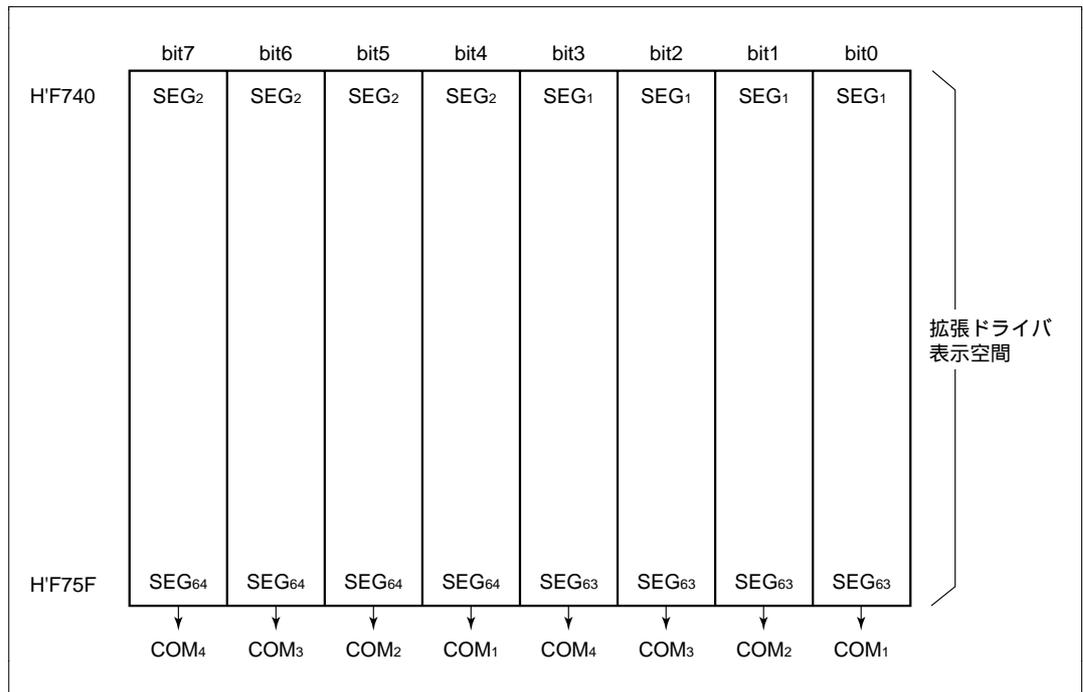


図 13.9 セグメント外部拡張した場合の LCD RAM マップ
(SGX = "1", SGS3 ~ SGS0 = "0000" 1/4 デューティ)

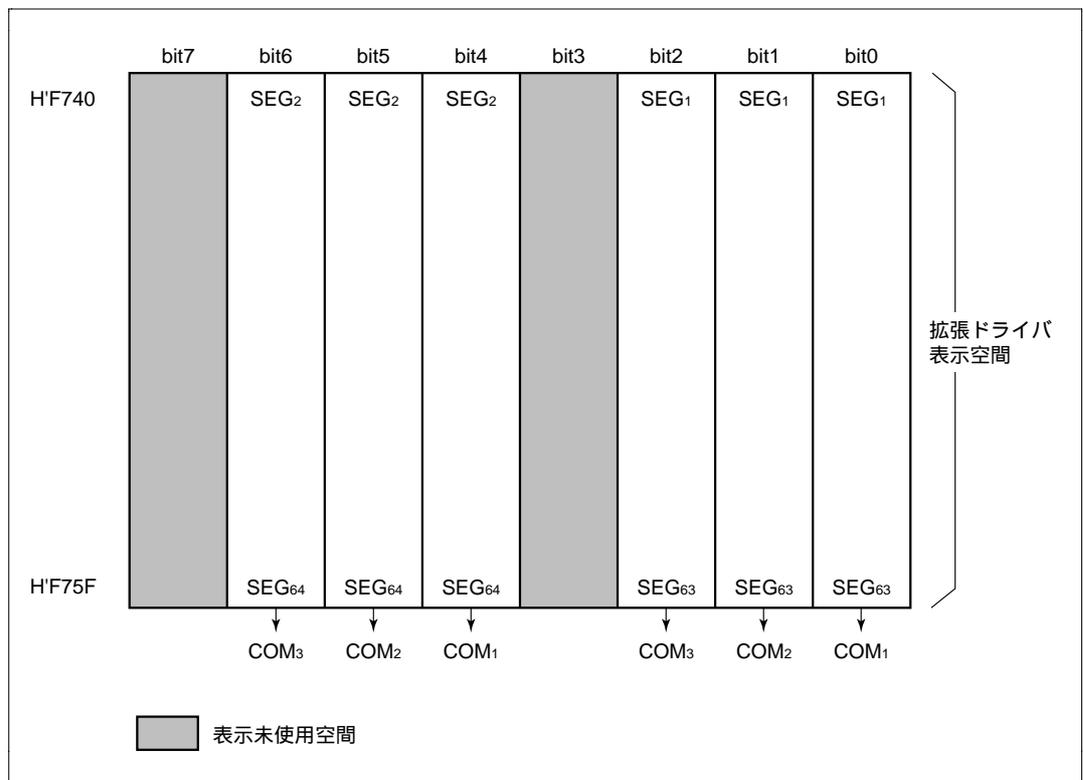


図 13.10 セグメント外部拡張した場合の LCD RAM マップ
(SGX = "1", SGS3 ~ SGS0 = "0000" 1/3 デューティ)

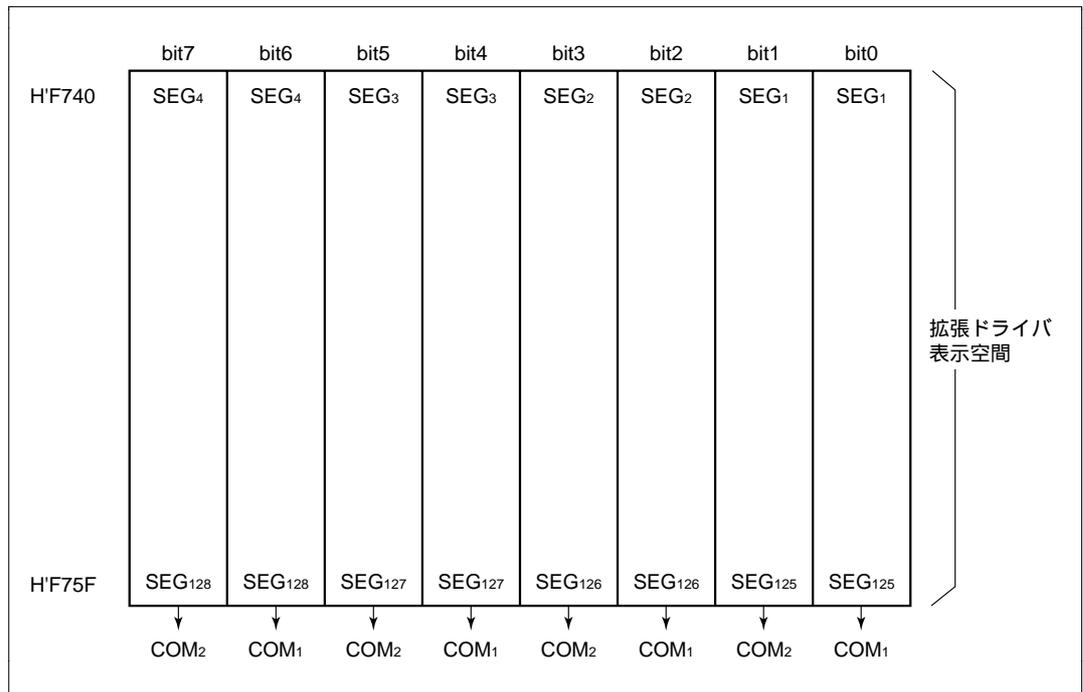


図 13.11 セグメント外部拡張した場合の LCD RAM マップ
(SGX = "1", SGS3 ~ SGS0 = "0000" 1/2 デューティ)

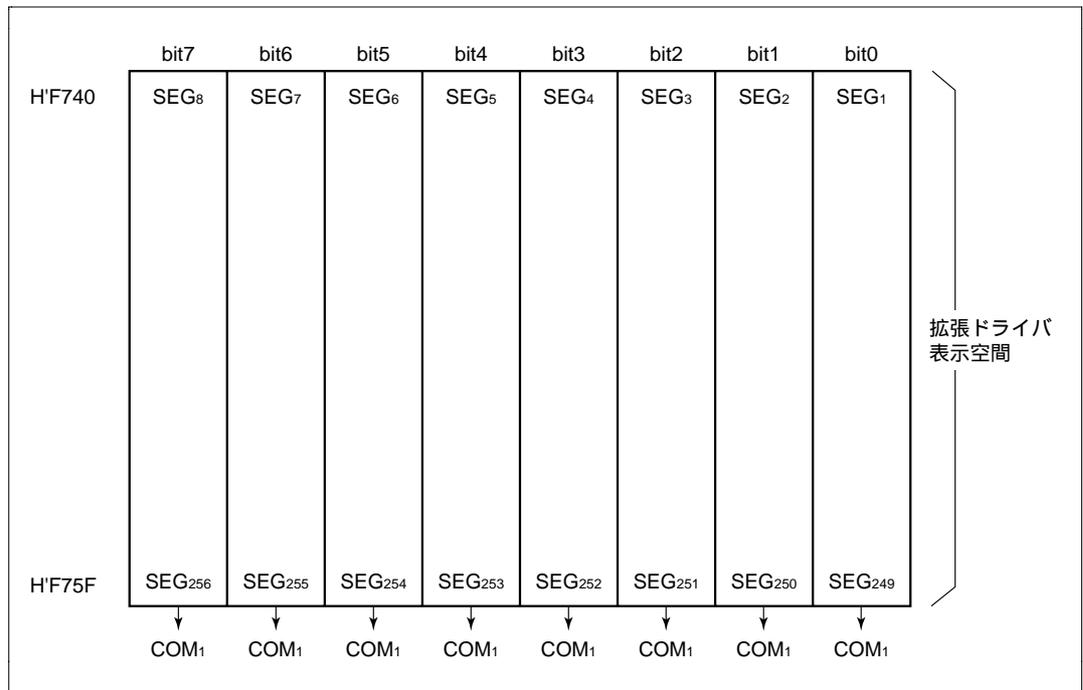


図 13.12 セグメント外部拡張しない場合の LCD RAM マップ
(SGX = "1", SGS3 ~ SGS0 = "0000" スタティック)

13.3.3 輝度調整機能 (V₀ 端子)

LCD 駆動電源部の詳細ブロック図を図 13.13 に示します。

V₀ 端子には、V_{CC} が昇圧定電圧電源回路からの出力 5V のどちらかが出力されます。これらの電圧を直接 LCD 駆動電圧として使用する場合は V₀ 端子と V₁ 端子を短絡して使用します。また、V₀ 端子と V₁ 端子の間に可変抵抗 R を接続することにより、V₁ 端子に印加される電圧を調整することができ、LCD パネルの輝度調整が可能となります。

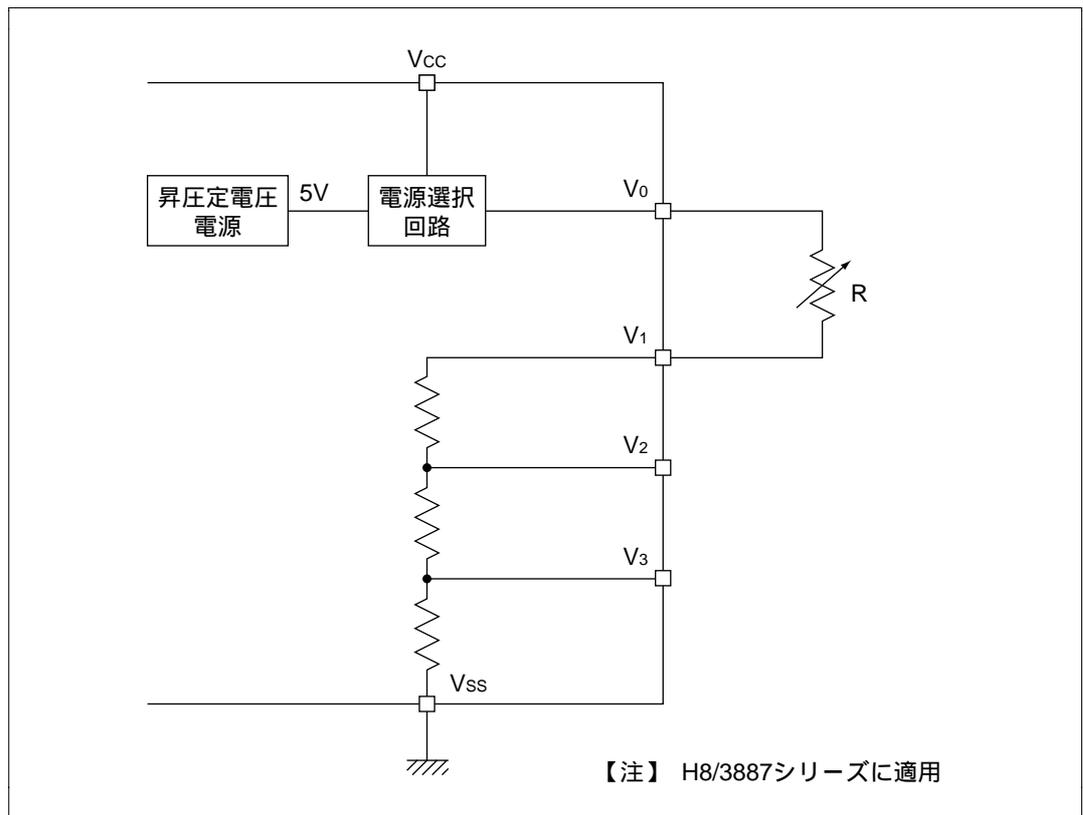


図 13.13 LCD 駆動電源部

13.3.4 昇圧定電圧 (5V) 電源

H8/3887 シリーズは昇圧定電圧 (5V) 電源を内蔵しており、 V_{CC} に依存せず 5V 一定電圧を得ることができます。H8/3847 シリーズには内蔵しておりません。

LCD コントロールレジスタ 2 (LCR2) の SUPS を 1 にすることにより、昇圧定電圧 (5V) 電源が動作し、 V_0 端子から 5V 定電圧が出力されます。 V_0 端子と V_1 端子を短絡するか、または抵抗を接続し、分圧して使用してください。

【注】 昇圧定電圧 (5V) 電源は、本 LSI の LCD 駆動電源以外に使用しないでください。
また、大きなパネルを駆動する場合、電源容量が不足する場合があります。この場合には、 V_{CC} を電源として使用するか、外部電源回路を使用してください。

13.3.5 低消費電力 LCD 駆動方式

LCD 電源回路には、通常内蔵分割抵抗を用いるのが最も簡単な方法ですが、内蔵抵抗が固定のため、常に内蔵抵抗の V_{CC} から V_{SS} へ一定の直流電流が流れていることとなります。この電流は LCD パネルの消費電流に依存しないため、消費電流の小さな LCD パネルを使用する場合、無駄な電力を消費していることとなります。本 LSI にはこの無駄な電力を改善する機能が内蔵されています。この機能を活用することにより、LCD パネルの消費電流に最適な電源回路を得ることができます。

(1) 原理

1. 図 13.10 に示すように、LCD 電源端子の V_1 、 V_2 、 V_3 にコンデンサを外付回路として接続します。
2. V_1 、 V_2 、 V_3 に接続されたコンデンサは図 13.14 に示すような周期で充電と放電を繰り返し、電位を保持します。
3. このとき、充電される電位は V_1 、 V_2 、 V_3 各々の端子に相当する電位です。
(例えば、 $1/3$ バイアス駆動の場合、 V_2 は V_1 の $3/2$ 、 V_3 は V_1 の $3/1$ の電位を充電します。)
4. これらのコンデンサに充電された電荷により LCD パネルに電源を供給します。
5. したがって、LCD パネルの消費電流の値によって、これらのコンデンサの容量と充放電期間が決まります。
6. 充放電期間はソフトウェアにより選択することができます。

(2) 動作例 ($1/3$ バイアス駆動の場合)

1. 図中の充電期間 T_c の間、 V_1 、 V_2 、 V_3 端子には図 13.14 に示すような電位 (V_2 は V_1 の $3/2$ 、 V_3 は V_1 の $3/1$ の電位) が内蔵分割抵抗により分圧され、外付コンデンサの C_1 、 C_2 、 C_3 を充電します。この期間も LCD パネルは駆動されています。
2. 次の放電期間 T_{dc} には、充電は停止し、各コンデンサに充電された電荷を放電することにより、LCD パネルを駆動します。

3. このとき、放電により若干の電圧降下が生じますが、充電期間とコンデンサの容量に適切な値を選択し、LCD パネルの駆動に影響がないようにしなければなりません。
4. このようにして V1、V2、V3 に接続されたコンデンサは図 13.14 に示すような周期で充電と放電を繰り返し、電位を保持しながら LCD パネルを駆動続けます。
5. 以上から分かるように、コンデンサの容量の値と充放電期間は、使用する LCD パネルの消費電流の値によって決まります。また充放電期間は CDS3~0 により選択することができます。
6. 実際のコンデンサの容量と充放電期間は、LCD パネルの消費電流の要求に合わせて実験的に決める必要がありますが、内部分割抵抗に常時直流電流が流れる場合に比べ、最適な電流値を選択することができます。

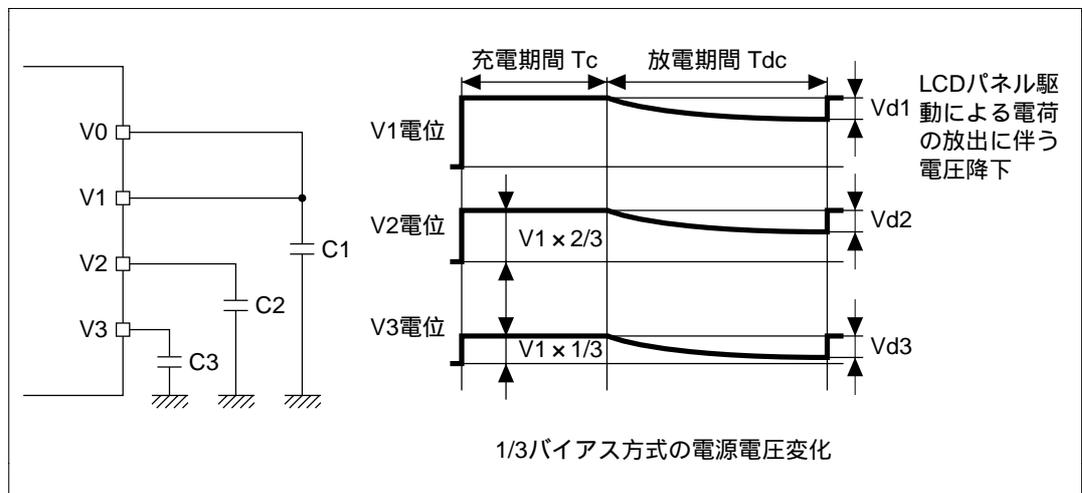


図 13.14 低消費電力 LCD 駆動方式動作例

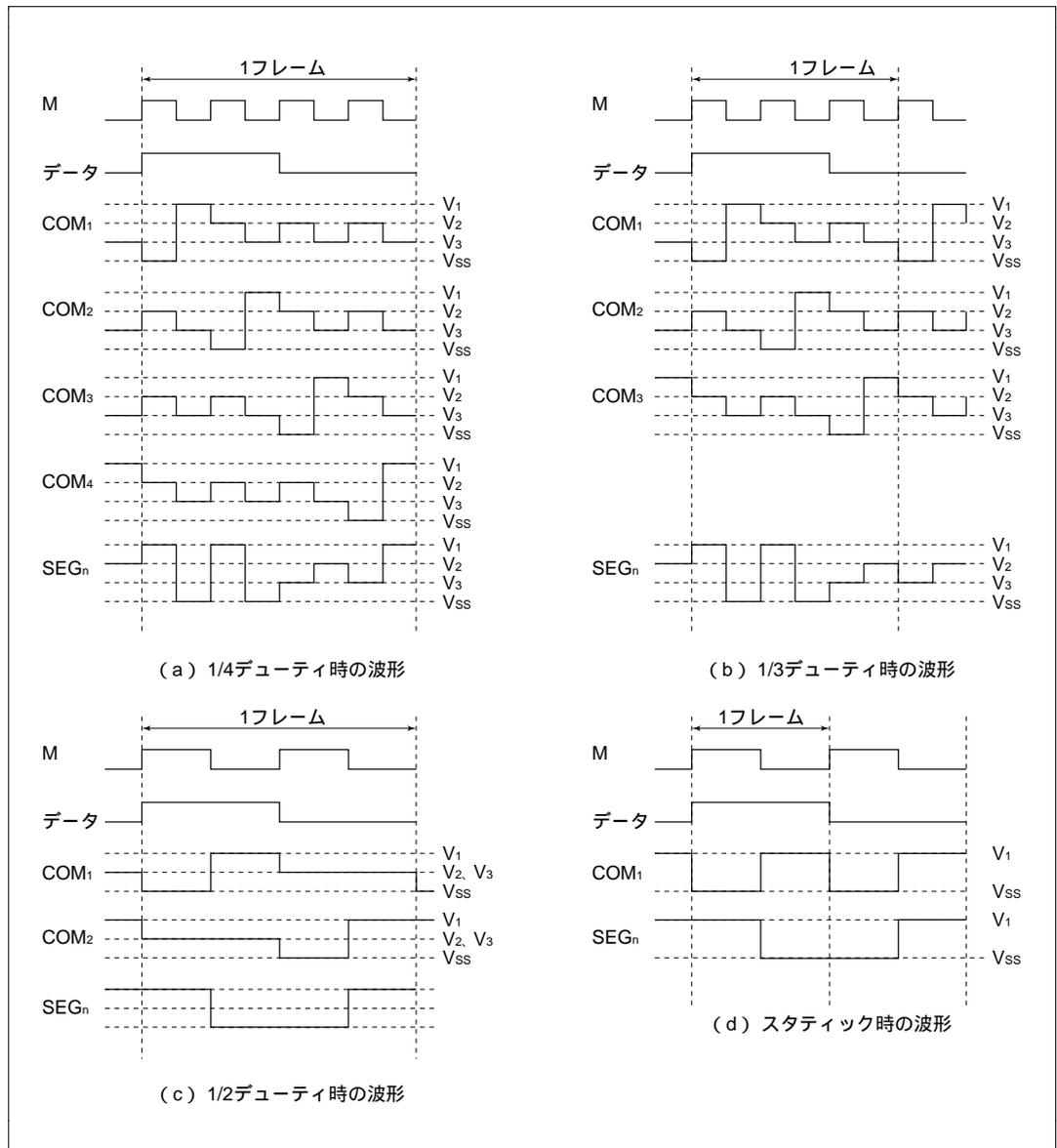


図 13.15 各デューティでの出力波形 (A 波形)

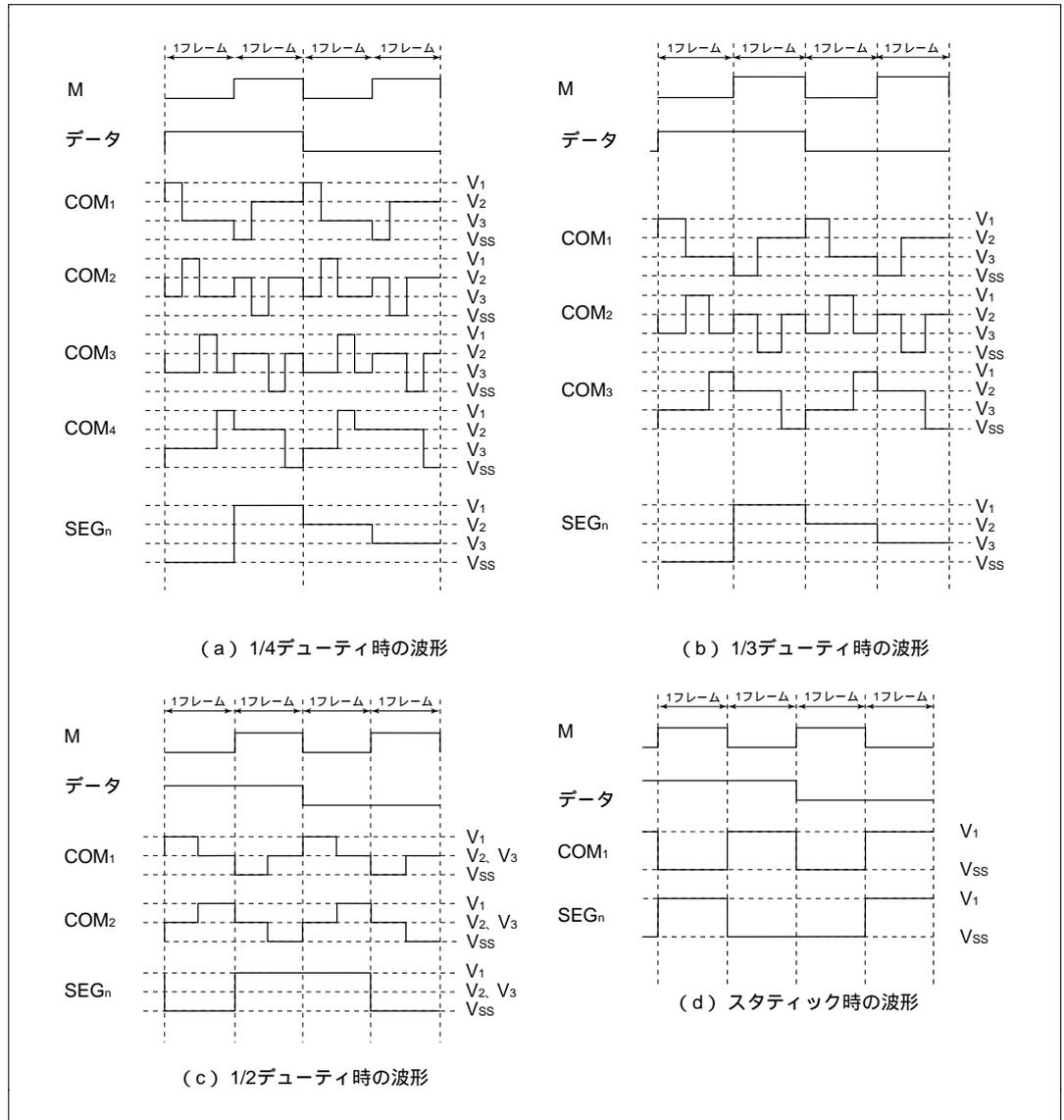


図 13.16 各デューティでの出力波形 (B 波形)

表 13.3 出力レベルの関係

データ		0	0	1	1
M		0	1	0	1
スタティック	コモン出力	V_1	V_{SS}	V_1	V_{SS}
	セグメント出力	V_1	V_{SS}	V_{SS}	V_1
1/2 デューティ	コモン出力	V_2, V_3	V_2, V_3	V_1	V_{SS}
	セグメント出力	V_1	V_{SS}	V_{SS}	V_1
1/3 デューティ	コモン出力	V_3	V_2	V_1	V_{SS}
	セグメント出力	V_2	V_3	V_{SS}	V_1
1/4 デューティ	コモン出力	V_3	V_2	V_1	V_{SS}
	セグメント出力	V_2	V_3	V_{SS}	V_1

13.3.6 低消費電力モード時の動作

本 LSI は低消費電力モード時でも、LCD コントローラ/ドライバを動作させることができます。低消費電力モード時の LCD コントローラ/ドライバの動作状態を表 13.4 に示します。

サブアクティブモード/ウォッチモード/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ので、CKS3~CKS0 で w 、 $w/2$ または $w/4$ を選択していないとクロックが供給されず、表示が停止します。この場合 LCD パネルには直流電圧がかかる可能性がありますので必ず w 、 $w/2$ または $w/4$ を選択するようにしてください。また、アクティブ(中速)モードではシステムクロックが切換わるので、フレーム周波数が変化しないように CKS3~CKS0 を変更する必要があります。

表 13.4 低消費電力モードと表示動作の関係

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	モジュール スタンバイ
		クロック	動作	動作	動作	停止	停止	停止	停止
	w	動作	動作	動作	動作	動作	動作	停止*1	停止*4
表示動作	ACT = "0"	停止	停止	停止	停止	停止	停止	停止*2	停止
	ACT = "1"	停止	表示	表示	表示*3	表示*3	表示*3	停止*2	停止

【注】 *1 サブクロック発振器は停止しませんがクロックの供給は停止します。

*2 PSW に関係なく LCD 駆動電源を OFF します。

*3 使用クロックに w 、 $w/2$ または $w/4$ を選択していないと表示動作を行いません。

*4 LCD に供給されるクロックは停止します。

13.3.7 LCD 駆動電源の強化

大きなパネルを駆動する場合、内蔵の電源容量では足りないことがあります。昇圧定電圧（5V）電源を使用して電源容量が不足する場合は、 V_{CC} を電源として使用するか、外部電源回路を使用してください。 V_{CC} を電源として使用した場合で電源容量が不足する場合は、電源のインピーダンスを下げる必要があります。この対策として、図 13.17 に示すように $V_1 \sim V_3$ 端子に $0.1 \sim 0.3\mu\text{F}$ 程度のバイパスコンデンサを接続する方法や、新たに分割抵抗を外部に設ける方法があります。

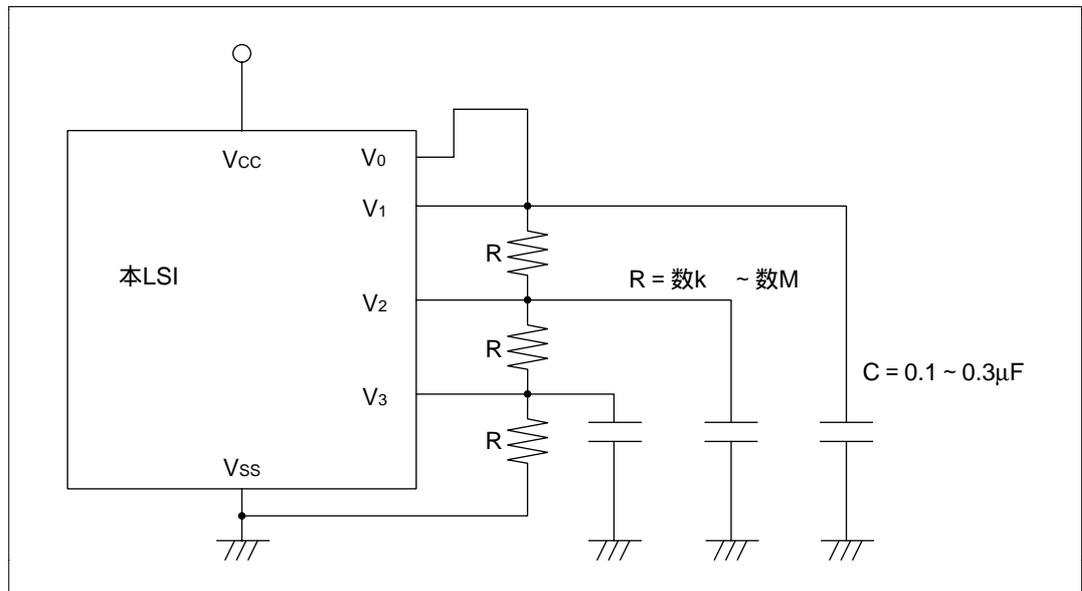


図 13.17 外部分割抵抗の接続方法

13.3.8 HD66100 との接続

セグメントを外部に拡張したい場合はHD66100を接続します。HD66100を1個接続することで80セグメント拡張することができます。外部拡張を行う際はLPCRのSGXによりSEG₄₀～SEG₃₇端子の機能を拡張用信号とし、SGS3～SGS0に"0000"を設定します。外部にはLCD RAMのSEG₁からのデータを出力します。SEG₃₆～SEG₁はポートとして機能します。

図 13.18 にHD66100との接続例を示します。出力レベルはデータとM端子出力の組み合わせにより決定しますが、その組み合わせがHD66100と異なります。表 13.3 にLCD 駆動電源の出力レベルを示します。また、デューティごとのコモン/セグメント波形を図 13.11、図 13.12 に示します。

ACT="0"とするとCL₂="0"、CL₁="0"、M="0"、DOはその瞬間に出力していたデータ("1"か"0")で停止します。また、スタンバイモード時には拡張端子はハイインピーダンス状態（フローティング）となります。

外部拡張を行うとLCDパネルにおける負荷が増し、内蔵電源では電流容量が足りない場合があります。その場合は「13.3.7 LCD 駆動電源の強化」を参照してください。

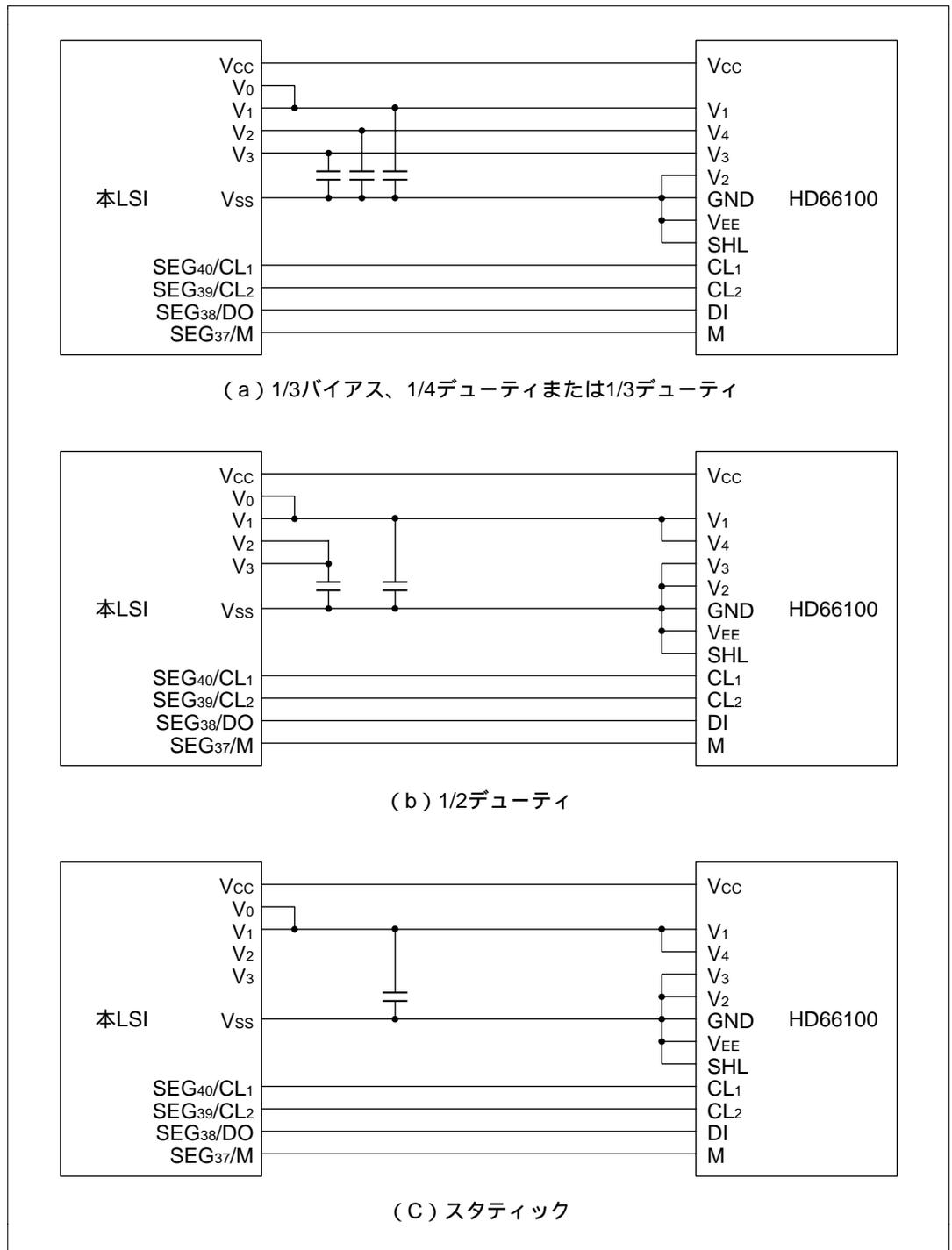


図 13.18 HD66100 との接続

14. 電源回路

第 14 章 目次

14.1	概要.....	393
14.2	内部電源降圧回路を使用する場合	393
14.3	内部電源降圧回路を使用しない場合	394

14.1 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 1.5V 一定に固定することができます。このため外部電源を 1.8V 以上で使用した場合に消費される電流値は約 1.5V で使用した場合とほぼ同一の低電流に抑えることができます。もちろん内部電源降圧回路を使用せずに外部電源電圧と内部電源電圧を同一にして使用することも可能です。

14.2 内部電源降圧回路を使用する場合

図 14.1 に示すように、 V_{CC} 端子に外部電源を接続し、 CV_{CC} と V_{SS} 間に約 $0.1 \mu F$ の容量を接続してください。この外部回路を付加するだけで内部降圧回路が有効になります。

- 【注】
1. 外部回路のインターフェースは 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。
 2. 内部電源降圧回路を使用している場合の動作周波数は $V_{CC}=2.2 \sim 5.5V$ の場合 $f_{osc}=0.4MHz \sim 2MHz$ となり、それ以外の場合は $f_{osc}=0.4MHz \sim 1MHz$ となります。
 3. LCD 電源、A/D 変換器のアナログ電源は内部降圧の影響は受けません。

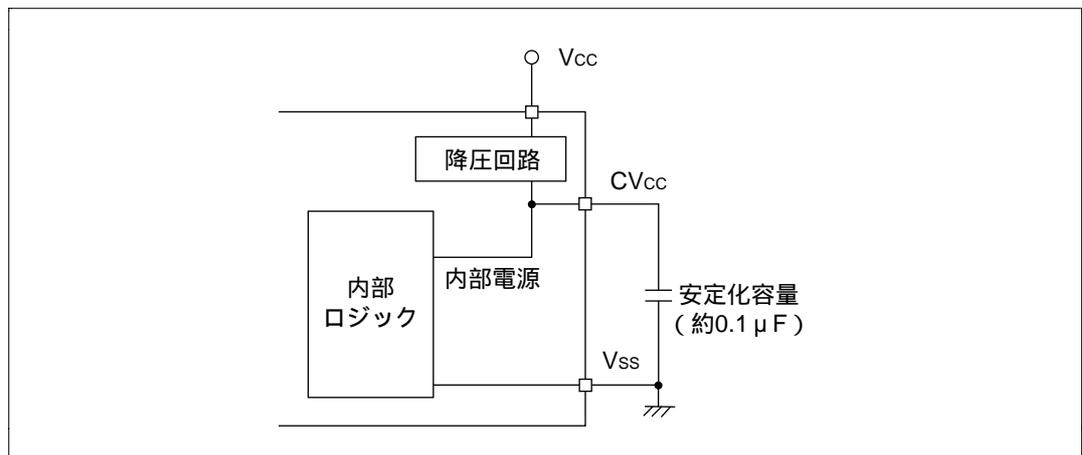


図 14.1 内部電源降圧回路を使用する場合の電源接続図

14.3 内部電源降圧回路を使用しない場合

内部電源降圧回路を使用しない場合は図 14.2 に示すように、Vcc 端子と CVcc 端子に外部電源を接続してください。直接外部電源が内部電源に入力されます。

【注】 使用可能な電源電圧は 1.8V ~ 5.5V です。この範囲以外（1.8V 未満、5.5V を超える）の電圧を入力した場合の動作は保証されません。

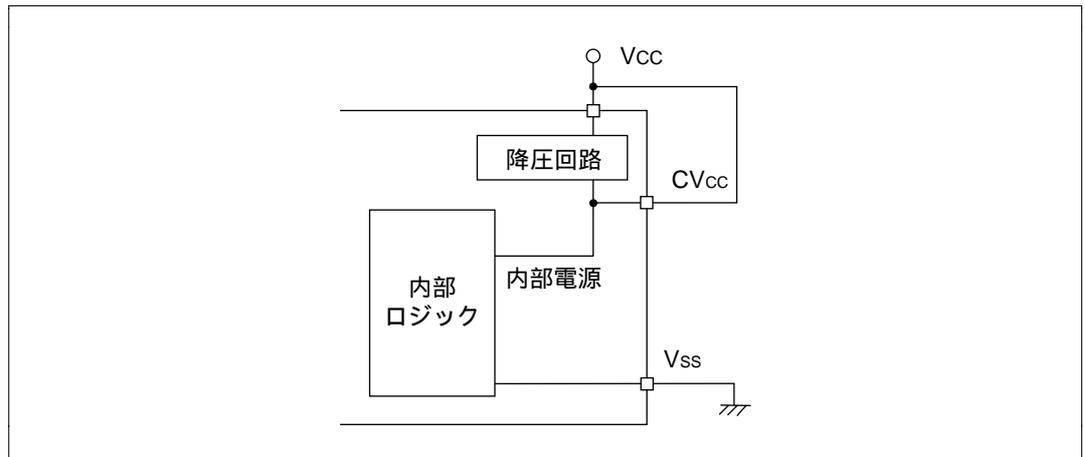


図 14.2 内部電源降圧回路を使用しない場合の電源接続図

15. 電気的特性

第 15 章 目次

15.1	H8/3887、H8/3847 シリーズ絶対最大定格.....	397
15.2	H8/3887、H8/3847 シリーズ電気的特性.....	398
	15.2.1 電源電圧と動作範囲.....	398
	15.2.2 DC 特性.....	401
	15.2.3 AC 特性.....	407
	15.2.4 A/D 変換器特性.....	410
	15.2.5 LCD 特性.....	411
15.3	動作タイミング.....	412
15.4	出力負荷回路.....	416
15.5	発振子の等価回路.....	417

15.1 H8/3887、H8/3847 シリーズ絶対最大定格

絶対最大定格を表 15.1 に示します。

表 15.1 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	*	
アナログ電源電圧	AV_{CC}	- 0.3 ~ + 7.0	V		
プログラム電圧	V_{PP}	- 0.3 ~ + 13.0	V		
入力電圧	ポート B、C 以外	V_{in}	- 0.3 ~ $V_{CC} + 0.3$		V
	ポート B、C	AV_{in}	- 0.3 ~ $AV_{CC} + 0.3$		V
動作温度	T_{opr}	- 20 ~ + 75			
保存温度	T_{stg}	- 55 ~ + 125			

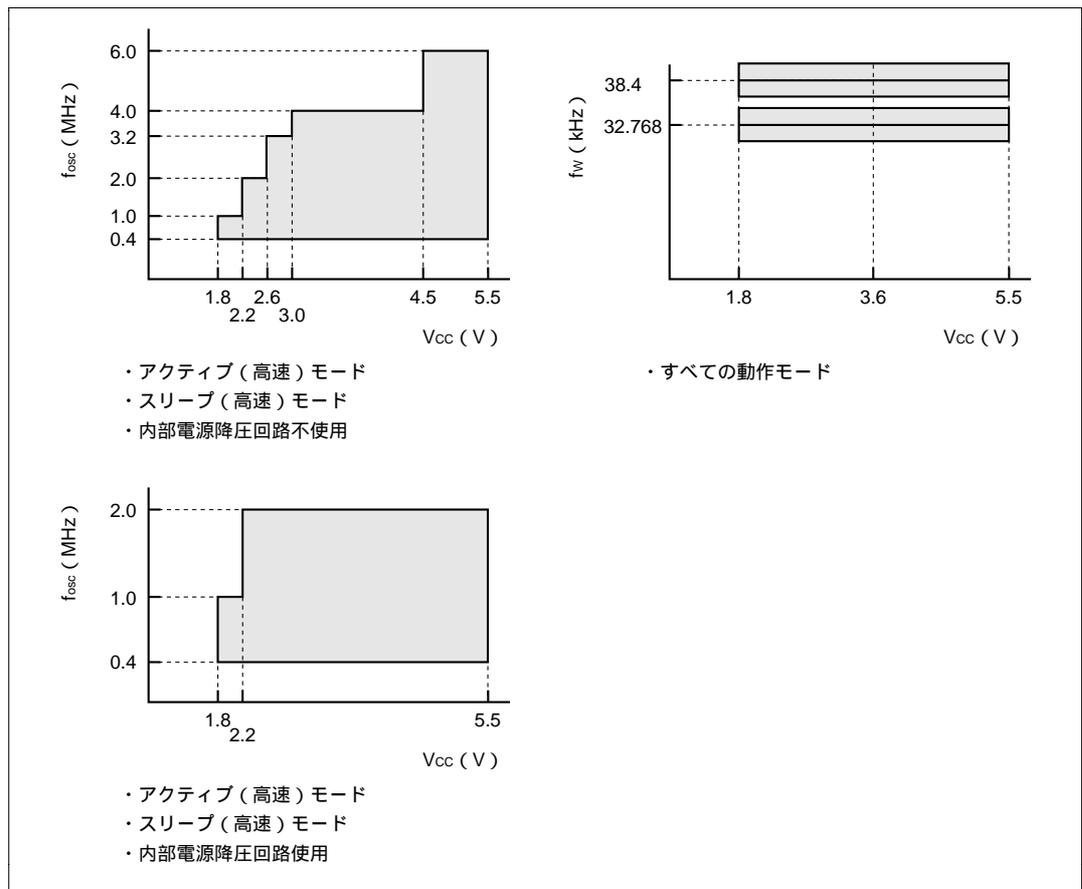
【注】 * 絶対最大定格をこえて LSI を使用した場合、LSI の永久破壊となることがあります。また、通常動作では、「電気的特性」の条件で使用することが望ましく、この条件をこえると LSI の誤動作の原因になるとともに、LSI の信頼性に悪影響をおよぼすことがあります。

15.2 H8/3887、H8/3847 シリーズ電気的特性

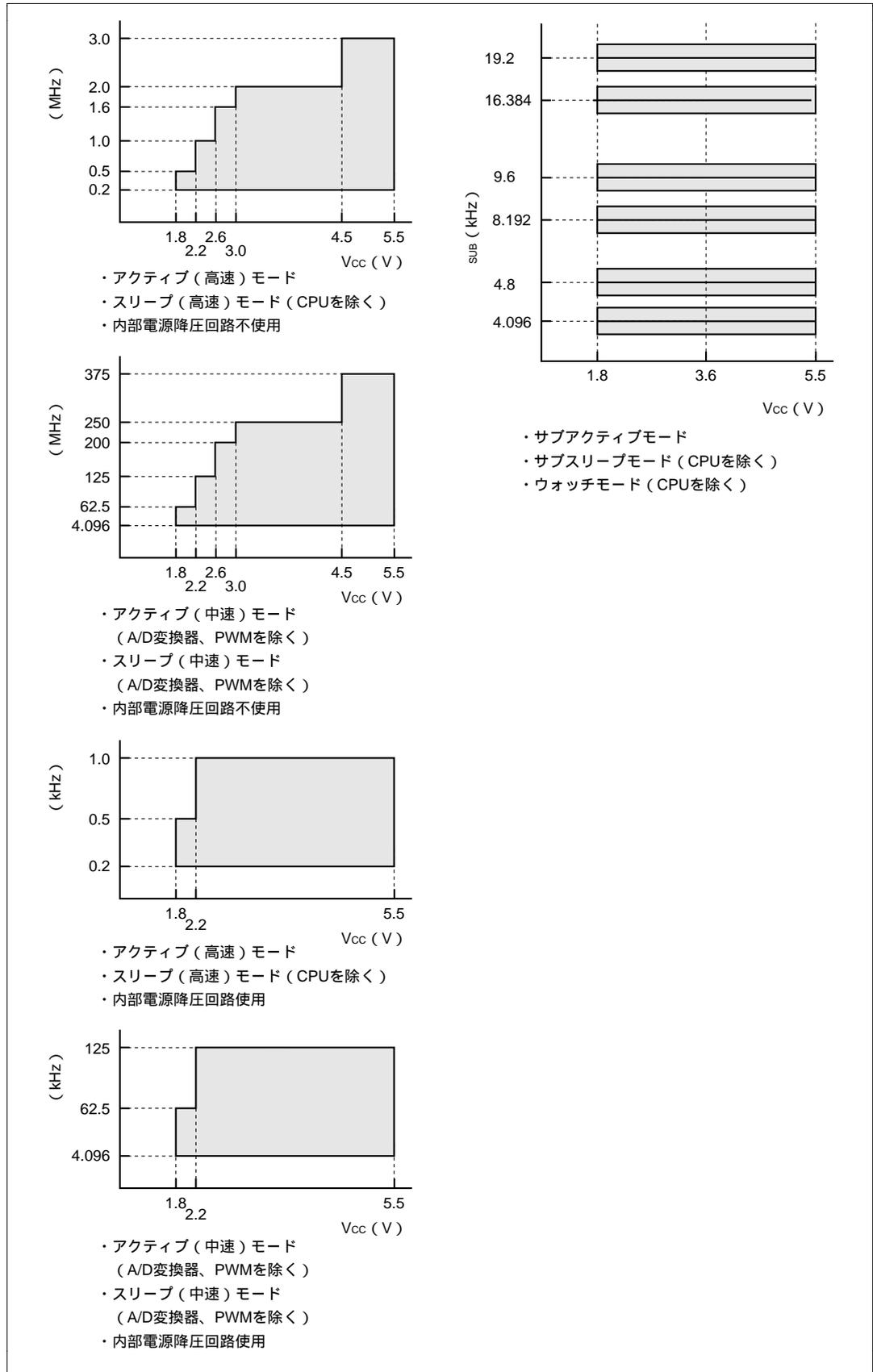
15.2.1 電源電圧と動作範囲

電源電圧と動作範囲（網かけ部）を以下に示します。

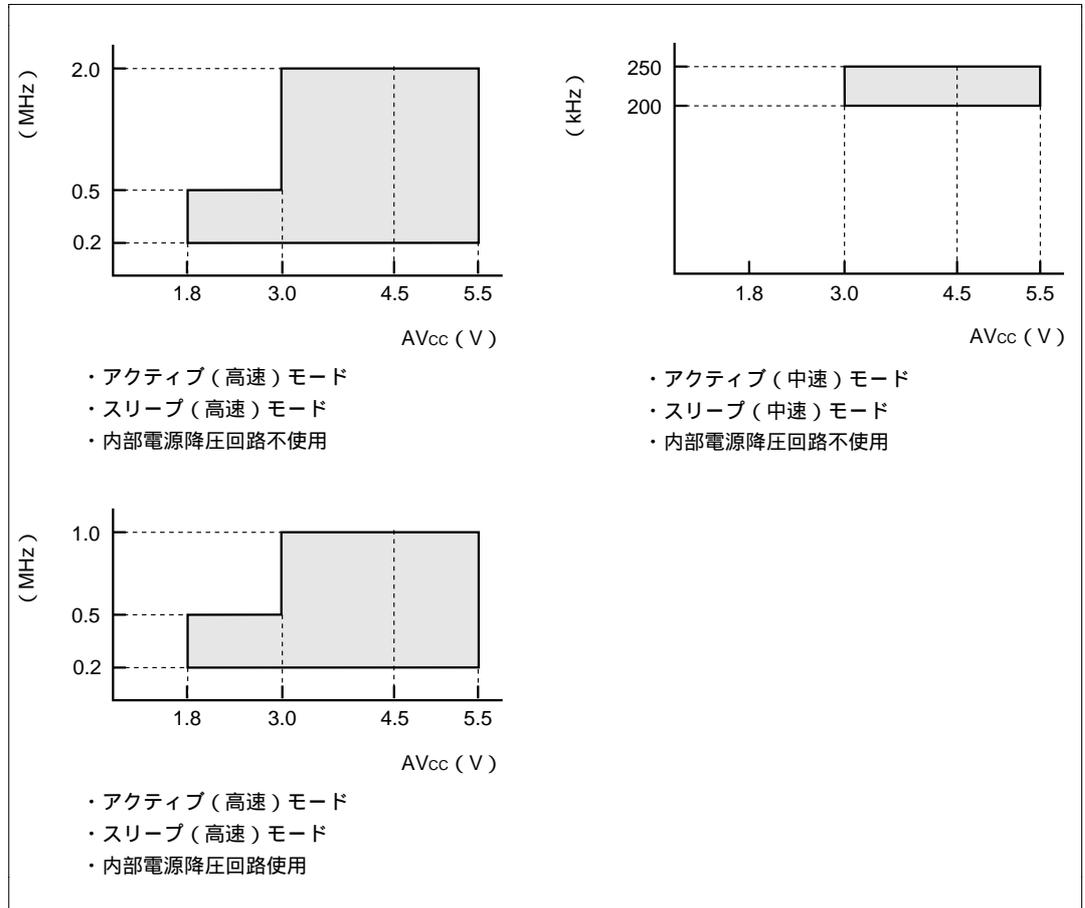
(1) 電源電圧と発振周波数の範囲



(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



15.2.2 DC 特性

DC 特性を表 15.2 に示します。

表 15.2 DC 特性 (1)

(特記なき場合、 $V_{CC} = 1.8 \sim 5.5V$ 、 $AV_{CC} = 1.8 \sim 5.5V$ 、 $V_{SS} = AV_{SS} = 0.0V$ 、 $T_a = -20 \sim +75$ 、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} 、 $\overline{WKP}_0 \sim \overline{WKP}_7$ 、 $\overline{IRQ}_0 \sim \overline{IRQ}_4$ 、 AEVL、AEVH、 TMIC、TMIF、 TMIG、SCK ₁ 、 SCK ₃₁ 、SCK ₃₂ 、 \overline{ADTRG}	$V_{CC} = 4.0 \sim 5.5V$	$0.8V_{CC}$		$V_{CC} + 0.3$	V	
			上記以外	$0.9V_{CC}$		$V_{CC} + 0.3$		
		SI ₁ 、RXD ₃₁ 、RXD ₃₂	$V_{CC} = 4.0 \sim 5.5V$	$0.7V_{CC}$		$V_{CC} + 0.3$	V	
		UD	上記以外	$0.8V_{CC}$		$V_{CC} + 0.3$		
		OSC ₁	$V_{CC} = 4.0 \sim 5.5V$	$0.8V_{CC}$		$V_{CC} + 0.3$	V	
			上記以外	$0.9V_{CC}$		$V_{CC} + 0.3$		
		X ₁	$V_{CC} = 1.8 \sim 5.5V$	$0.9V_{CC}$		$V_{CC} + 0.3$	V	
		P1 ₀ ~ P1 ₇ 、 P2 ₀ ~ P2 ₇ 、 P3 ₀ ~ P3 ₇ 、 P4 ₀ ~ P4 ₃ 、 P5 ₀ ~ P5 ₇ 、 P6 ₀ ~ P6 ₇ 、 P7 ₀ ~ P7 ₇ 、 P8 ₀ ~ P8 ₇ 、 P9 ₀ ~ P9 ₇ 、 PA ₀ ~ PA ₃	$V_{CC} = 4.0 \sim 5.5V$	$0.7V_{CC}$		$V_{CC} + 0.3$	V	
			上記以外	$0.8V_{CC}$		$V_{CC} + 0.3$		
		PB ₀ ~ PB ₇	$V_{CC} = 4.0 \sim 5.5V$	$0.7V_{CC}$		$AV_{CC} + 0.3$		
			上記以外	$0.8V_{CC}$		$AV_{CC} + 0.3$		
		PC ₀ ~ PC ₃						

【注】 TEST 端子は、 V_{SS} に接続してください。

15. 電気的特性

表 15.2 DC 特性 (2)

(特記なき場合、 $V_{CC} = 1.8 \sim 5.5V$ 、 $AV_{CC} = 1.8 \sim 5.5V$ 、 $V_{SS} = AV_{SS} = 0.0V$ 、 $T_a = -20 \sim +75$ 、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} 、 $\overline{WKP}_0 \sim \overline{WKP}_7$ 、 $\overline{IRQ}_0 \sim \overline{IRQ}_4$ 、 AEVL、AEVH、 TMIC、TMIF、 TMIG、SCK ₁ 、 SCK ₃₁ 、SCK ₃₂ 、 \overline{ADTRG}	$V_{CC} = 4.0 \sim 5.5V$	- 0.3		$0.2V_{CC}$	V		
		上記以外	- 0.3		$0.1V_{CC}$				
		SI ₁ 、RXD ₃₁ 、RXD ₃₂	$V_{CC} = 4.0 \sim 5.5V$	- 0.3		$0.3V_{CC}$	V		
		UD	上記以外	- 0.3		$0.2V_{CC}$			
		OSC ₁	内部降圧使用時		- 0.3		0.2	V	
			$V_{CC} = 4.0 \sim 5.5V$		- 0.3		$0.2V_{CC}$		
			上記以外		- 0.3		$0.1V_{CC}$		
		X ₁		$V_{CC} = 1.8 \sim 5.5V$	- 0.3		$0.1V_{CC}$	V	
		P1 ₀ ~ P1 ₇ 、 P2 ₀ ~ P2 ₇ 、 P3 ₀ ~ P3 ₇ 、 P4 ₀ ~ P4 ₃ 、 P5 ₀ ~ P5 ₇ 、 P6 ₀ ~ P6 ₇ 、 P7 ₀ ~ P7 ₇ 、 P8 ₀ ~ P8 ₇ 、 P9 ₀ ~ P9 ₇ 、 PA ₀ ~ PA ₃ 、 PB ₀ ~ PB ₇ 、 PC ₀ ~ PC ₃		$V_{CC} = 4.0 \sim 5.5V$	- 0.3		$0.3V_{CC}$	V	
			上記以外		- 0.3		$0.2V_{CC}$		
出力"High" レベル電圧	V_{OH}	P1 ₀ ~ P1 ₇ 、 P2 ₀ ~ P2 ₇ 、 P3 ₀ ~ P3 ₇ 、 P4 ₀ ~ P4 ₂ 、 P5 ₀ ~ P5 ₇ 、 P6 ₀ ~ P6 ₇ 、	$V_{CC} = 4.0 \sim 5.5V$ - $I_{OH} = 1.0mA$	$V_{CC} - 1.0$			V		
			$V_{CC} = 4.0 \sim 5.5V$ - $I_{OH} = 0.5mA$	$V_{CC} - 0.5$					

表 15.2 DC特性 (3)

(特記なき場合、 $V_{CC} = 1.8 \sim 5.5V$ 、 $AV_{CC} = 1.8 \sim 5.5V$ 、 $V_{SS} = AV_{SS} = 0.0V$ 、 $T_a = -20 \sim +75$ 、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"High" レベル電圧	V_{OH}	P7 ₀ ~ P7 ₇ 、 P8 ₀ ~ P8 ₇ 、 P9 ₀ ~ P9 ₇ 、 PA ₀ ~ PA ₃	- $I_{OH} = 0.1mA$	$V_{CC} - 0.3$			V			
出力"Low" レベル電圧	V_{OL}	P1 ₀ ~ P1 ₇ 、 P4 ₀ ~ P4 ₂	$V_{CC} = 4.0 \sim 5.5V$			0.6	V			
			$I_{OL} = 1.6mA$			0.5				
		P5 ₀ ~ P5 ₇ 、 P6 ₀ ~ P6 ₇ 、 P7 ₀ ~ P7 ₇ 、 P8 ₀ ~ P8 ₇ 、 P9 ₀ ~ P9 ₇ 、 PA ₀ ~ PA ₃	$I_{OL} = 0.4mA$			0.5				
			P2 ₀ ~ P2 ₇ 、 P3 ₀ ~ P3 ₇	$V_{CC} = 4.0 \sim 5.5V$					1.5	
				$I_{OL} = 10mA$					0.6	
				$V_{CC} = 4.0 \sim 5.5V$					0.5	
P2 ₀ ~ P2 ₇ 、 P3 ₀ ~ P3 ₇	$I_{OL} = 1.6mA$			0.5						
	$I_{OL} = 0.4mA$			0.5						
	入出力 リーク電流	I_{IL}	\overline{RES} 、P4 ₃	$V_{IN} = 0.5V \sim V_{CC} - 0.5V$			20.0	μA	*2	
							1.0		*1	
入出力 リーク電流	I_{IL}	OSC ₁ 、X ₁ 、 P1 ₀ ~ P1 ₇ 、 P2 ₀ ~ P2 ₇ 、 P3 ₀ ~ P3 ₇ 、 P4 ₀ ~ P4 ₂ 、 P5 ₀ ~ P5 ₇ 、 P6 ₀ ~ P6 ₇ 、 P7 ₀ ~ P7 ₇ 、 P8 ₀ ~ P8 ₇ 、 P9 ₀ ~ P9 ₇ 、 PA ₀ ~ PA ₃	$V_{IN} = 0.5V \sim V_{CC} - 0.5V$			1.0	μA			
				PB ₀ ~ PB ₇ 、 PC ₀ ~ PC ₃	$V_{IN} = 0.5V \sim AV_{CC} - 0.5V$			1.0		

15. 電気的特性

表 15.2 DC 特性 (4)

(特記なき場合、 $V_{CC} = 1.8 \sim 5.5V$ 、 $AV_{CC} = 1.8 \sim 5.5V$ 、 $V_{SS} = AV_{SS} = 0.0V$ 、 $T_a = -20 \sim +75$ 、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.		
プルアップ MOS 電流	- I_p	P1 ₀ ~ P1 ₇ 、 P3 ₀ ~ P3 ₇ 、 P5 ₀ ~ P5 ₇ 、 P6 ₀ ~ P6 ₇	$V_{CC} = 5V$ 、 $V_{IN} = 0V$	50.0		300.0	μA	
			$V_{CC} = 2.7V$ 、 $V_{IN} = 0V$		35.0		μA	参考値
入力容量	C_{IN}	電源、 \overline{RES} 、P4 ₃ 、 PB ₀ ~ PB ₇ 端子を除く 全入力端子	$f = 1MHz$ 、 $V_{IN} = 0V$ 、 $T_a = 25$			15.0	pF	
		\overline{RES}				80.0		*2
						15.0		*1
		P4 ₃				50.0		*2
		PB ₀ ~ PB ₇				15.0		*1
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 $V_{CC} = 5V$ 、 $f_{OSC} = 2MHz$		0.7	1.0	mA	*3 *4 *5
	I_{OPE2}	V_{CC}	アクティブ (中速) モード $V_{CC} = 5V$ 、 $f_{OSC} = 2MHz$ $_{OSC}/128$ 時		0.3	0.5		*3 *4 *5
スリープモード 消費電流	I_{SLEEP}	V_{CC}	$V_{CC} = 5V$ 、 $f_{OSC} = 2MHz$		0.4	0.6	mA	*3 *4 *5
サブアクティブ モード消費電流	I_{SUB}	V_{CC}	$V_{CC} = 2.7V$ 、LCD 点灯 32kHz 水晶発振子使用時 ($_{SUB} = w/2$)		15	30	μA	*3 *4 *5
			$V_{CC} = 2.7V$ 、LCD 点灯 32kHz 水晶発振子使用時 ($_{SUB} = w/8$)		8			μA
サブスリープモード 消費電流	I_{SUBSP}	V_{CC}	$V_{CC} = 2.7V$ 、LCD 点灯 32kHz 水晶発振子使用時 ($_{SUB} = w/2$)		7.5	16	μA	*3 *4 *5

表 15.2 DC 特性 (5)

(特記なき場合、 $V_{CC} = 1.8 \sim 5.5V$ 、 $AV_{CC} = 1.8 \sim 5.5V$ 、 $V_{SS} = AV_{SS} = 0.0V$ 、 $T_a = -20 \sim +75$ 、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.		
ウォッチモード 消費電流	I_{WATCH}	V_{CC}	$V_{CC} = 2.7V$ 、32kHz 水晶発振子使用時 LCD 未使用		2.8	6.0	μA	*3 *4 *5
スタンバイ モード消費電流	I_{STBY}	V_{CC}	32kHz 水晶発振子未使用時		1.0	5.0	μA	*3 *4
RAM データ 保持電圧	V_{RAM}	V_{CC}		1.5			V	*3 *4

- 【注】 *1 マスク ROM 製品に適用します。
*2 HD6473887、HD6473847 に適用します。
*3 消費電流測定時の端子の状態

消費電流測定時の端子の状態

モード	\overline{RES} 端子	内部状態	各端子	定電圧 LCD 電源	発振端子
アクティブ (高速) モード アクティブ (中速) モード	V_{CC}	CPU のみ動作	V_{CC}	停止	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_1 端子 = GND
スリープモード	V_{CC}	タイマのみ動作	V_{CC}		
サブアクティブ モード	V_{CC}	CPU のみ動作	V_{CC}	停止	システムクロック発振器： 水晶発振子
サブスリープ モード	V_{CC}	タイマのみ動作 CPU は停止	V_{CC}	停止	サブクロック発振器： 水晶発振子
ウォッチモード	V_{CC}	時計用タイム ベースのみ動作 CPU は停止	V_{CC}	停止	
スタンバイモード	V_{CC}	CPU、タイマ ともに停止	V_{CC}	停止	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_1 端子 = GND

*4 プルアップ MOS や出力バッファに流れる電流は除きます。

*5 内部降圧使用時

15. 電気的特性

表 15.2 DC特性 (6)

(特記なき場合、 $V_{CC} = 1.8 \sim 5.5V$ 、 $AV_{CC} = 1.8 \sim 5.5V$ 、 $V_{SS} = AV_{SS} = 0.0V$ 、 $T_a = -20 \sim +75$ 、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 2、3 以外の出力端子	$V_{CC} = 4.0V \sim 5.5V$			2.0	mA	
		ポート 2、3	$V_{CC} = 4.0V \sim 5.5V$			10.0		
		全出力端子				0.5		
出力"Low" レベル許容電流 (総和)	I_{OL}	ポート 2、3 以外の出力端子	$V_{CC} = 4.0V \sim 5.5V$			40.0	mA	
		ポート 2、3	$V_{CC} = 4.0V \sim 5.5V$			80.0		
		全出力端子				20.0		
出力"High" レベル許容電流 (1端子あたり)	$-I_{OH}$	全出力端子	$V_{CC} = 4.0V \sim 5.5V$			2.0	mA	
			上記以外					
出力"High" レベル許容電流 (総和)	$-I_{OH}$	全出力端子	$V_{CC} = 4.0V \sim 5.5V$			15.0	mA	
			上記以外					

15.2.3 AC 特性

制御信号タイミングを表 15.3 に、シリアルインタフェースタイミングを表 15.4、表 15.5 に示します。

表 15.3 制御信号タイミング

(特記なき場合、 $V_{CC} = 1.8 \sim 5.5V$ 、 $AV_{CC} = 1.8 \sim 5.5V$ 、 $V_{SS} = AV_{SS} = 0.0V$ 、 $T_a = -20 \sim +75$ 、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	参照図
				min.	typ.	max.		
システムクロック 発振器発振周波数	f_{OSC}	OSC ₁ 、OSC ₂	$V_{CC} = 4.5 \sim 5.5V$	0.4		6	MHz	*2
			$V_{CC} = 3.0 \sim 5.5V$	0.4		4		
			$V_{CC} = 2.6 \sim 5.5V$	0.4		3.2		
			$V_{CC} = 2.2 \sim 5.5V$	0.4		2		
			上記以外	0.4		1		
OSC クロック (OSC) サイクル時間	t_{OSC}	OSC ₁ 、OSC ₂	$V_{CC} = 4.5 \sim 5.5V$	167		2500	ns	図 15.1 *2
			$V_{CC} = 3.0 \sim 5.5V$	250		2500		
			$V_{CC} = 2.6 \sim 5.5V$	313		2500		
			$V_{CC} = 2.2 \sim 5.5V$	500		2500		
			上記以外	1000		2500		
システムクロック () サイクル時間	t_{cyc}			2		128	t_{OSC}	
						244.1	μs	
サブクロック発振器 発振周波数	f_W	X ₁ 、X ₂			32.768 または 38.4		kHz	
ウォッチクロック (W) サイクル時間	t_W	X ₁ 、X ₂			30.5 または 26.0		μs	図 15.1
サブクロック (SUB) サイクル時間	t_{subcyc}			2		8	t_W	*1
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	t_{rc}	OSC ₁ 、OSC ₂	図 15.10 の場合 $V_{CC} = 2.2 \sim 5.5V$		20	45	μs	図 15.10 *2
			図 15.10 の場合 $V_{CC} = 2.2 \sim 5.5V$		0.1	8	ms	図 15.10
			上記以外			50	ms	
発振安定時間	t_{rc}	X ₁ 、X ₂				2.0	s	
外部クロック "High" レベル幅	t_{CPH}	OSC ₁	$V_{CC} = 4.5 \sim 5.5V$	70			ns	図 15.1 *2
			$V_{CC} = 3.0 \sim 5.5V$	100				
			$V_{CC} = 2.6 \sim 5.5V$	140				
			$V_{CC} = 2.2 \sim 5.5V$	200				
			上記以外	400				

15. 電気的特性

項目	記号	適用端子	測定条件	規格値			単位	参照図
				min.	typ.	max.		
外部クロック "High"レベル幅	t_{CPH}	X_1			15.26 または 13.02		μs	図 15.1
外部クロック "Low"レベル幅	t_{CPL}	OSC ₁	$V_{CC} = 4.5 \sim 5.5V$	70			ns	図 15.1
			$V_{CC} = 3.0 \sim 5.5V$	100				
			$V_{CC} = 2.6 \sim 5.5V$	140				
			$V_{CC} = 2.2 \sim 5.5V$	200				
		上記以外	400			μs	図 15.1	
X_1			15.26 または 13.02					
外部クロック 立上り時間	t_{CPr}	OSC ₁	$V_{CC} = 4.5 \sim 5.5V$			20	ns	図 15.1 *2
			$V_{CC} = 2.6 \sim 5.5V$			30		
			上記以外			55		
		X_1				55.0	ns	図 15.1
外部クロック 立下り時間	t_{CPf}	OSC ₁	$V_{CC} = 4.5 \sim 5.5V$			20	ns	図 15.1 *2
			$V_{CC} = 2.6 \sim 5.5V$			30		
			上記以外			55		
		X_1				55.0	ns	図 15.1
RES 端子 "Low"レベル幅	t_{REL}	\overline{RES}		10			t_{cyc}	図 15.2
入力端子 "High"レベル幅	t_{IH}	$\overline{IRQ}_0 \sim \overline{IRQ}_4$ 、 $\overline{WKP}_0 \sim \overline{WKP}_7$ 、 \overline{ADTRG} 、 TMIC、 TMIF、TMIG、 AEVL、AEVH		2			t_{cyc} t_{subcyc}	図 15.3
入力端子 "Low"レベル幅	t_{IL}	$\overline{IRQ}_0 \sim \overline{IRQ}_4$ 、 $\overline{WKP}_0 \sim \overline{WKP}_7$ 、 \overline{ADTRG} 、 TMIC、 TMIF、TMIG、 AEVL、AEVH		2			t_{cyc} t_{subcyc}	図 15.3
UD 端子 最小変化幅	t_{UDH} t_{UDL}	UD		4			t_{cyc} t_{subcyc}	図 15.4

【注】 *1 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。

*2 内部電源降圧回路不使用

表 15.4 シリアルインタフェース (SCI1) タイミング

(特記なき場合、 $V_{CC} = 1.8 \sim 5.5V$ 、 $AV_{CC} = 1.8 \sim 5.5V$ 、 $V_{SS} = AV_{SS} = 0.0V$ 、 $T_a = -20 \sim +75$)

項目	記号	適用端子	測定条件	規格値			単位	参照図
				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 15.5
入力転送クロック "High"レベル幅	t_{SCKH}	SCK ₁		0.4			t_{Scyc}	図 15.5
入力転送クロック "Low"レベル幅	t_{SCKL}	SCK ₁		0.4			t_{Scyc}	図 15.5
入力転送クロック 立上り時間	t_{SCKr}	SCK ₁	$V_{CC} = 4.0 \sim 5.5V$			60.0	ns	図 15.5*
						80.0		図 15.5
入力転送クロック 立下り時間	t_{SCKf}	SCK ₁	$V_{CC} = 4.0 \sim 5.5V$			60.0	ns	図 15.5*
						80.0		図 15.5
シリアル出力データ 遅延時間	t_{SOD}	SO ₁	$V_{CC} = 4.0 \sim 5.5V$			200.0	ns	図 15.5*
						350.0		図 15.5
シリアル入力データ セットアップ時間	t_{SIS}	SI ₁	$V_{CC} = 4.0 \sim 5.5V$	200.0			ns	図 15.5*
				400.0				図 15.5
シリアル入力データ ホールド時間	t_{SIH}	SI ₁	$V_{CC} = 4.0 \sim 5.5V$	200.0			ns	図 15.5*
				400.0				図 15.5

【注】 * 内部電源降圧回路不使用

表 15.5 シリアルインタフェース (SCI31、SCI32) タイミング

(特記なき場合、 $V_{CC} = 1.8 \sim 5.5V$ 、 $AV_{CC} = 1.8 \sim 5.5V$ 、 $V_{SS} = AV_{SS} = 0.0V$ 、 $T_a = -20 \sim +75$)

項目	記号	測定条件	規格値			単位	参照図
			min.	typ.	max.		
入力 クロックサイクル	調歩同期	t_{Scyc}	4			t_{cyc} または t_{Subcyc}	図 15.6
	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 15.6
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} または t_{Subcyc}	図 15.7
		上記以外			1		
受信データセットアップ時間 (クロック同期)	t_{RXS}	$V_{CC} = 4.0 \sim 5.5V$	200.0			ns	図 15.7*
		上記以外	400.0				図 15.7
受信データホールド時間 (クロック同期)	t_{RXH}	$V_{CC} = 4.0 \sim 5.5V$	200.0			ns	図 15.7*
		上記以外	400.0				図 15.7

【注】 * 内部電源降圧回路不使用

15.2.4 A/D 変換器特性

A/D 変換器特性を表 15.6 に示します。

表 15.6 A/D 変換器特性

(特記なき場合、 $V_{CC} = 1.8 \sim 5.5V$ 、 $V_{SS} = AV_{SS} = 0.0V$ 、 $T_a = -20 \sim 75$)

項目	記号	適用端子	測定条件	規格値			単位	備考
				min.	typ.	max.		
アナログ電源電圧	AV_{CC}	AV_{CC}		1.8		5.5	V	*1
アナログ入力電圧	AV_{IN}	$AN_0 \sim AN_{11}$		-0.3		$AV_{CC} + 0.3$	V	
アナログ電源電流	AI_{OPE}	AV_{CC}	$AV_{CC} = 5.0V$			1.5	mA	
	AI_{STOP1}	AV_{CC}			600		μA	*2 参考値
	AI_{STOP2}	AV_{CC}				5	μA	*3
アナログ入力容量	C_{AIN}	$AN_0 \sim AN_{11}$				15.0	pF	
許容信号源 インピーダンス	R_{AIN}					10.0	k	
分解能 (データ長)						10	ビット	
非直線性誤差			$AV_{CC} = 3.0 \sim 5.5V$ $V_{CC} = 3.0 \sim 5.5V$			± 2.5	LSB	*4
			$AV_{CC} = 2.0 \sim 5.5V$ $V_{CC} = 2.0 \sim 5.5V$			± 5.5		
			上記以外			± 7.5		*5
量子化誤差						± 0.5	LSB	
絶対精度			$AV_{CC} = 3.0 \sim 5.5V$ $V_{CC} = 3.0 \sim 5.5V$			± 3.0	LSB	*4
			$AV_{CC} = 2.0 \sim 5.5V$ $V_{CC} = 2.0 \sim 5.5V$			± 6.0		
			上記以外			± 8.0		*5
変換時間			$AV_{CC} = 3.0 \sim 5.5V$ $V_{CC} = 3.0 \sim 5.5V$	15.5		15.5	μs	*4
			上記以外	62		155		

【注】 *1 A/D 変換器を使用しない場合は $AV_{CC} = V_{CC}$ としてください。

*2 AI_{STOP1} はアクティブモード、スリープモードでの A/D 変換待機時の電流値です。

*3 AI_{STOP2} 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、およびサブスリープモードでの A/D 変換待機時の電流値です。

*4 内部電源降圧回路不使用

*5 変換時間 124 μs

15.2.5 LCD 特性

LCD 特性を表 15.7 に示します。

表 15.7 LCD 特性

(特記なき場合、 $V_{CC} = 1.8 \sim 5.5V$ 、 $AV_{CC} = 1.8 \sim 5.5V$ 、 $V_{SS} = AV_{SS} = 0.0V$ 、 $T_a = -20 \sim +75$ 、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.		
セグメントドライバ 降下電圧	V_{DS}	SEG ₁ ~ SEG ₄₀	$I_D = 2\mu A$ $V_1 = 2.7 \sim 5.5V$			0.6	V	*1
コモンドライバ 降下電圧	V_{DC}	COM ₁ ~ COM ₄	$I_D = 2\mu A$ $V_1 = 2.7 \sim 5.5V$			0.3	V	*1
LCD 電源 分割抵抗	R_{LCD}		$V_1 - V_{SS}$ 間	0.5	3.0	9.0	M	
内蔵液晶表示電源電圧	V_{LP}	V_0			5.0		V	*2 参考値
液晶表示電圧	V_{LCD}	V_1		2.2		5.5	V	*3

【注】 *1 電源端子 V_1 、 V_2 、 V_3 、 V_{SS} から各セグメント端子またはコモン端子までの電圧降下です。

*2 昇圧定電圧電源動作時の出力電圧です。

*3 液晶表示電圧を外部電源より供給する場合は、 V_1 、 V_2 、 V_3 、 V_{SS} の関係を維持してください。

表 15.8 セグメント外部拡張 AC 特性

(特記なき場合、 $V_{CC} = 1.8 \sim 5.5V$ 、 $V_{SS} = 0.0V$ 、 $T_a = -20 \sim +75$ 、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.		
クロック"High"レベル幅	t_{CWH}	CL ₁ 、CL ₂	*	800.0			ns	図 15.8
クロック"Low"レベル幅	t_{CWL}	CL ₂	*	800.0			ns	図 15.8
クロックセットアップ時間	t_{CSU}	CL ₁ 、CL ₂	*	500.0			ns	図 15.8
データセットアップ時間	t_{SU}	DO	*	300.0			ns	図 15.8
データ保持時間	t_{DH}	DO	*	300.0			ns	図 15.8
M 遅延時間	t_{DM}	M		-1000.0		1000.0	ns	図 15.8
クロック立上がり 立下がり時間	t_{CT}	CL ₁ 、CL ₂				170.0	ns	図 15.8

【注】 * フレーム周波数を 488Hz ~ 30.5Hz に設定したときの値です。

15.3 動作タイミング

動作タイミングを図 15.1 ~ 図 15.7 に示します。

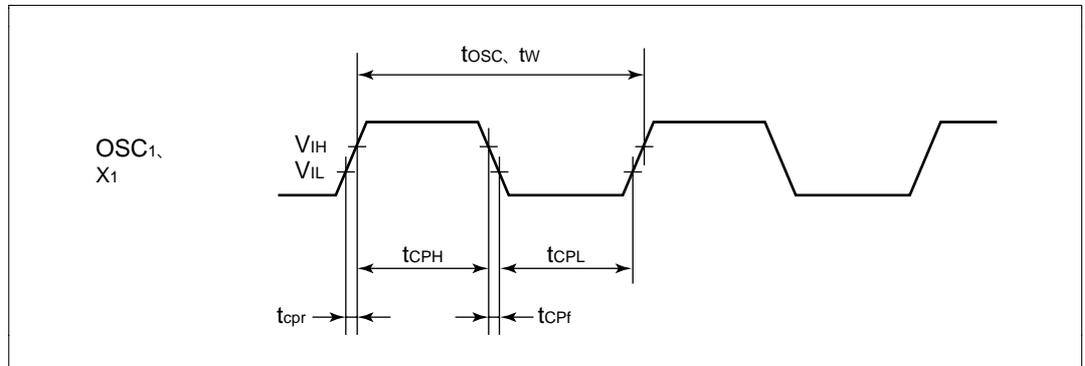


図 15.1 クロック入力タイミング

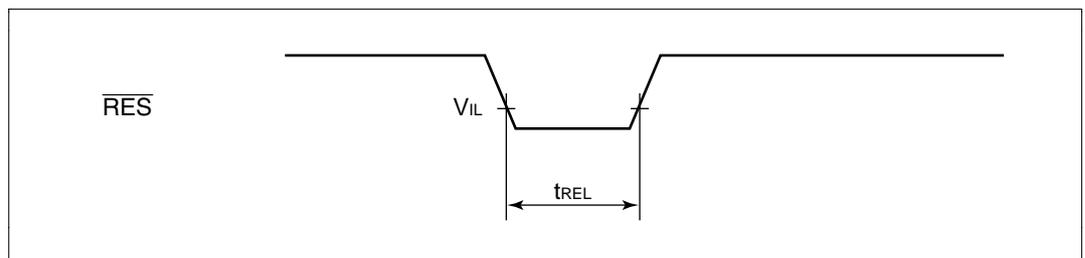


図 15.2 RES 端子"Low"レベル幅タイミング

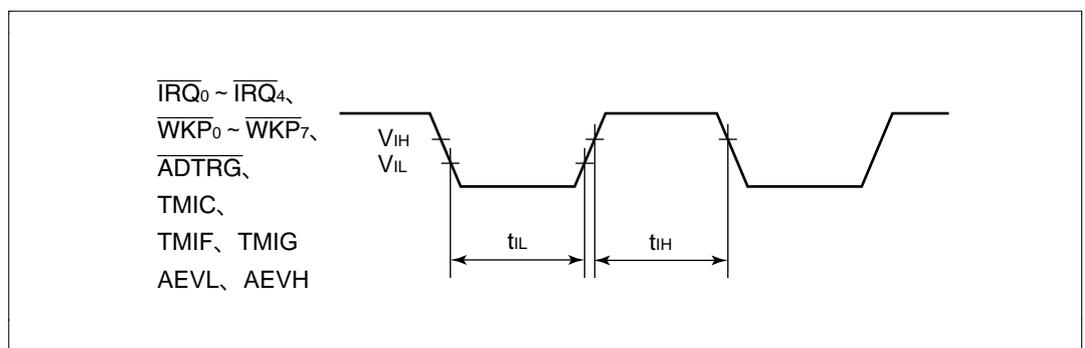


図 15.3 入力タイミング

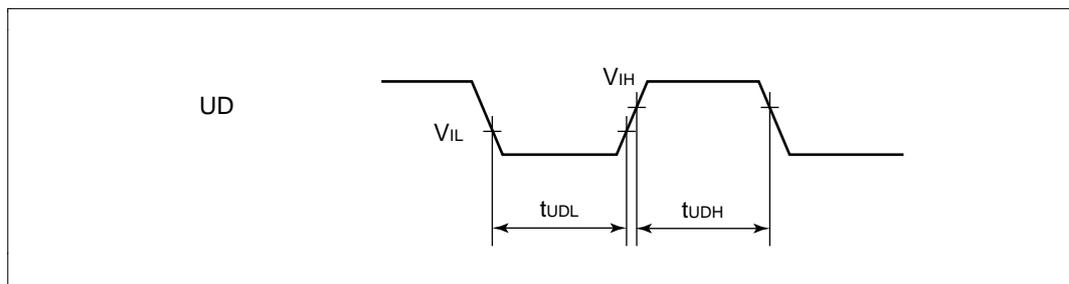


図 15.4 UD 端子最小変化幅タイミング

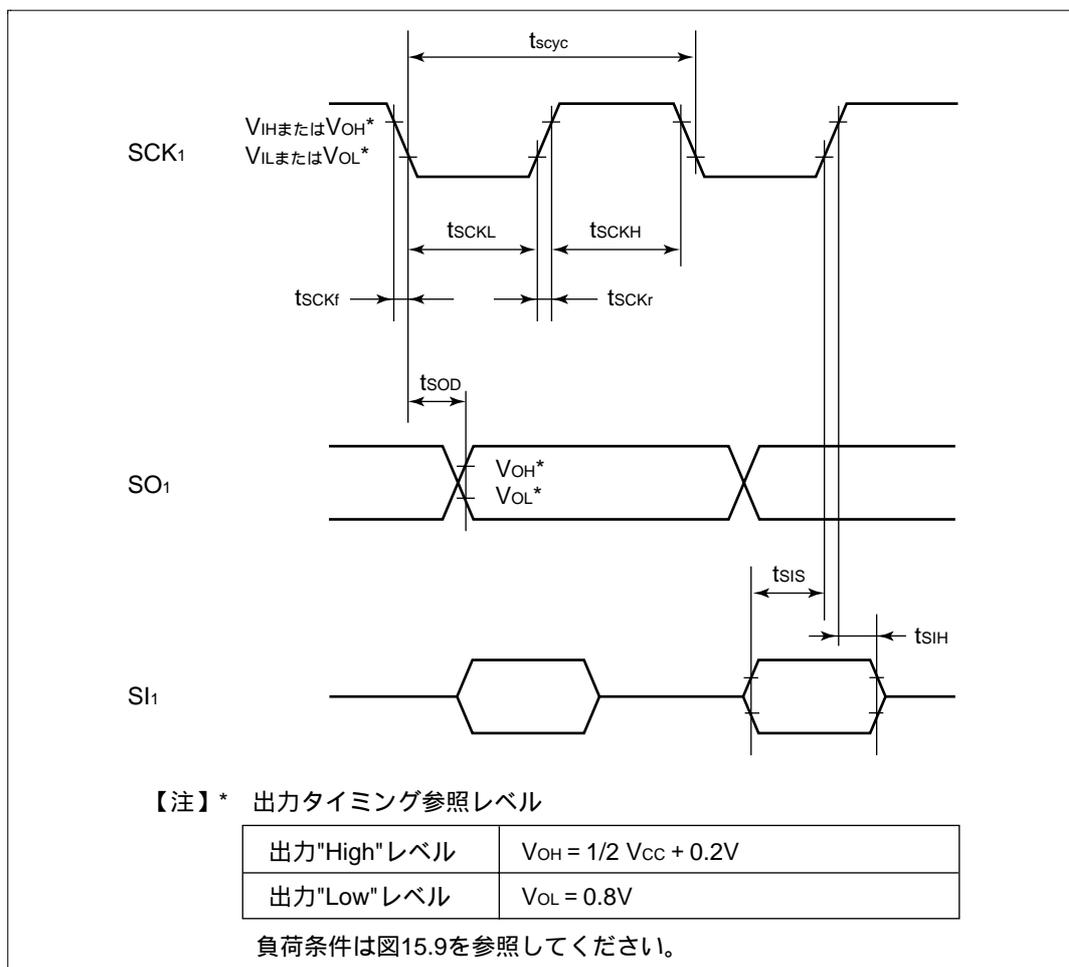


図 15.5 SCI1 入出力タイミング

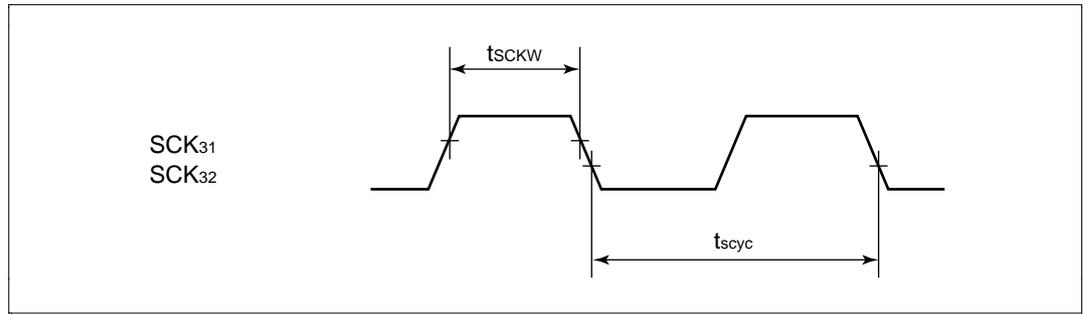


図 15.6 SCK3 入力クロックタイミング

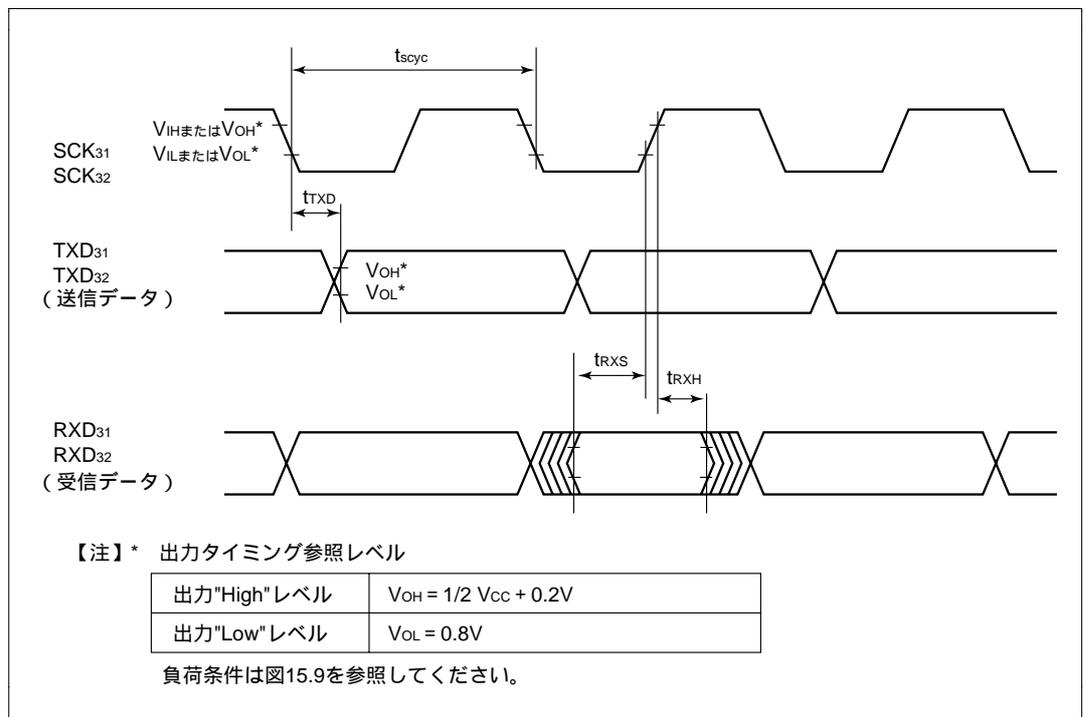


図 15.7 SCI3 クロック同期式モード入出力タイミング

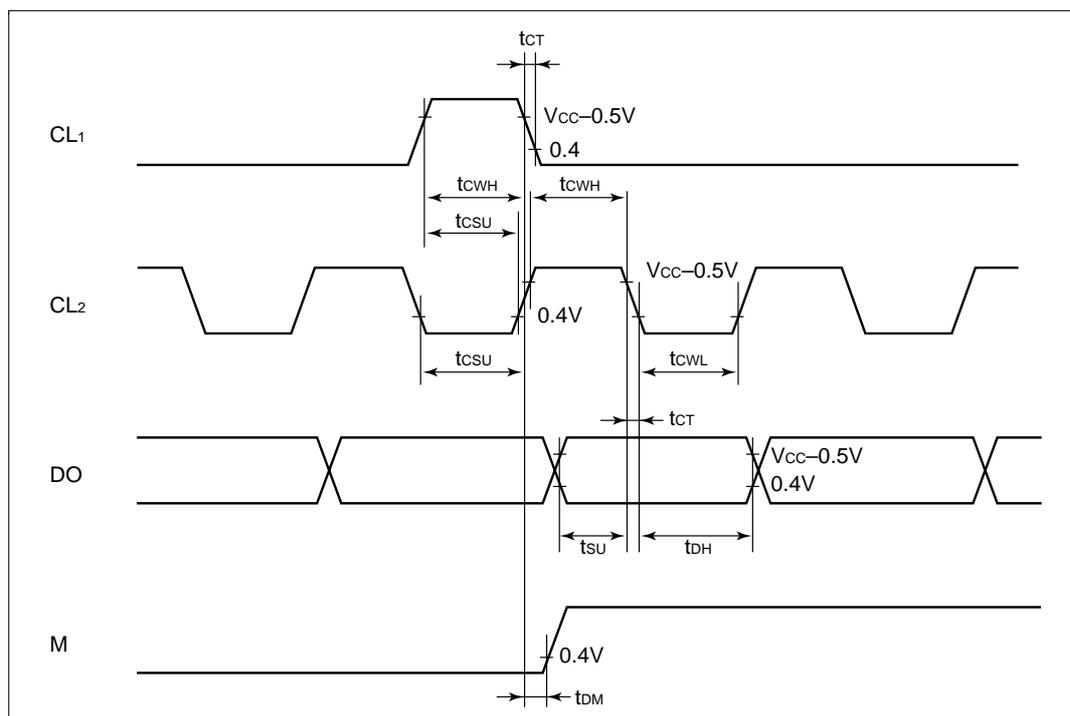


図 15.8 セグメント拡張信号タイミング

15.4 出力負荷回路

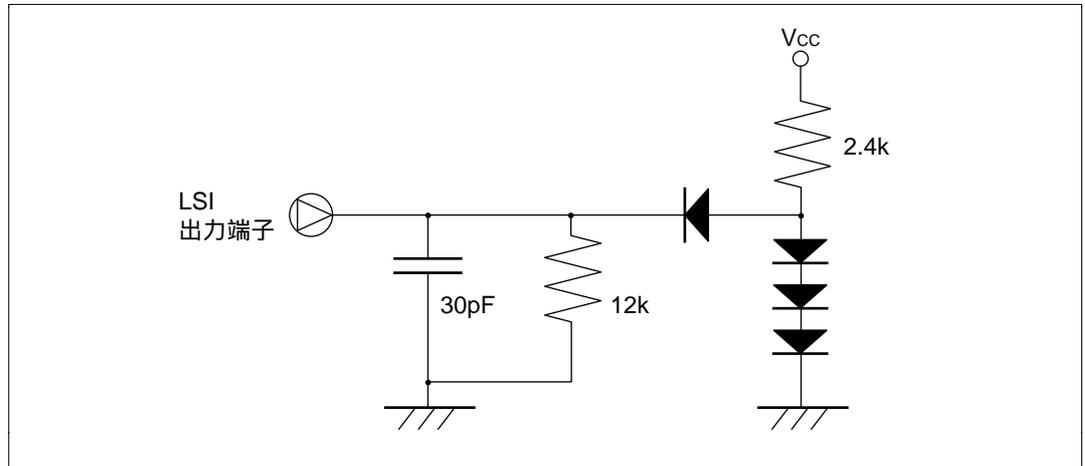


図 15.9 出力負荷条件

15.5 発振子の等価回路

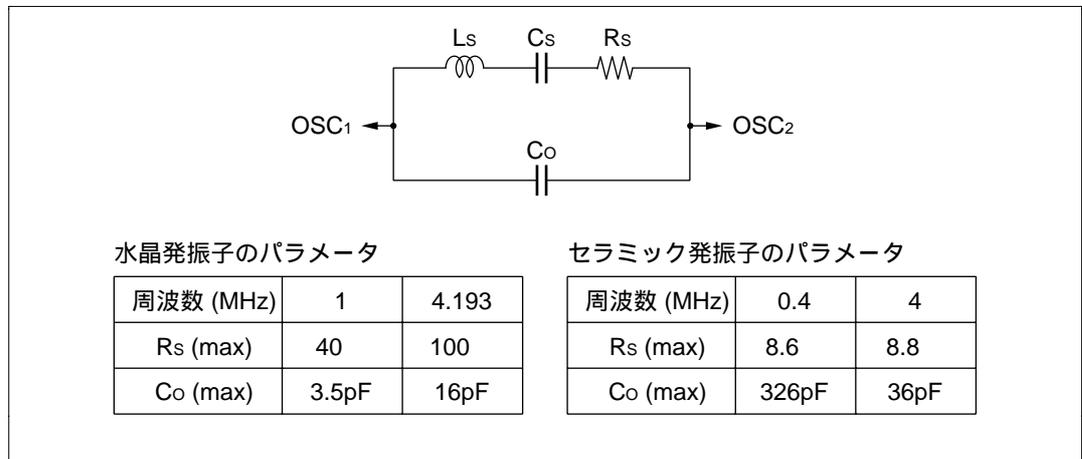


図 15.10 発振子の等価回路

付録

付録 目次

A.	命令	421	
	A.1	命令一覧	421
	A.2	オペレーションコードマップ	431
	A.3	命令実行ステート数	433
B.	内部 I/O レジスタ一覧	439	
	B.1	アドレス一覧	439
	B.2	機能一覧	443
C.	I/O ポートブロック図	497	
	C.1	ポート 1 ブロック図	497
	C.2	ポート 2 ブロック図	499
	C.3	ポート 3 ブロック図	501
	C.4	ポート 4 ブロック図	505
	C.5	ポート 5 ブロック図	507
	C.6	ポート 6 ブロック図	508
	C.7	ポート 7 ブロック図	509
	C.8	ポート 8 ブロック図	510
	C.9	ポート 9 ブロック図	511
	C.10	ポート A ブロック図	512
	C.11	ポート B ブロック図	513
	C.12	ポート C ブロック図	514
D.	各処理状態におけるポートの状態	515	
E.	ROM 発注手順	516	
	E.1	ROM 書き換え品開発の流れ（発注手順）	516
	E.2	ROM 発注時の注意事項	517
F.	製品型名一覧	518	
G.	外形寸法図	520	

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》

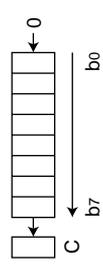
記号	
↓	実行結果にしたがって変化することを表します。
*	不確定であることを表します (値を保証しません)。
0	常に"0"にクリアされることを表します。
	実行結果に影響を受けないことを表します。

表 A.1 命令セット一覧

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 ステップ 数
		#xx:8/16	Rn	@Rn	@(d:16, Rn)	@-Rn/@Rn+ @aa:8/16	@(d:8, PC)		@@aa	I	H	N	Z	V	C	
MOV	B	2						#xx:8 Rd8	—	—	↑	↑	0	—	2	
	B		2					Rs8 Rd8	—	—	↑	↑	0	—	2	
	B			2				@Rs16 Rd8	—	—	↑	↑	0	—	4	
	B				4			@(d:16, Rs16) Rd8	—	—	↑	↑	0	—	6	
	B					2		@Rs16 Rd8 Rs16+1 Rs16	—	—	↑	↑	0	—	6	
	B						2	@aa:8 Rd8	—	—	↑	↑	0	—	4	
	B						4	@aa:16 Rd8	—	—	↑	↑	0	—	6	
	B		2					Rs8 @Rd16	—	—	↑	↑	0	—	4	
	B				4			Rs8 @(d:16, Rd16)	—	—	↑	↑	0	—	6	
	B					2		Rd16-1 Rd16	—	—	↑	↑	0	—	6	
	B							Rs8 @Rd16	—	—	↑	↑	0	—	4	
	B						2	Rs8 @aa:8	—	—	↑	↑	0	—	4	
	B						4	Rs8 @aa:16	—	—	↑	↑	0	—	6	
	W	4						#xx:16 Rd	—	—	↑	↑	0	—	4	
	W		2					Rs16 Rd16	—	—	↑	↑	0	—	2	
	W			2				@Rs16 Rd16	—	—	↑	↑	0	—	4	
	W				4			@(d:16, Rs16) Rd16	—	—	↑	↑	0	—	6	
	W					2		@Rs16 Rd16	—	—	↑	↑	0	—	6	
	W						4	@aa:16 Rd16	—	—	↑	↑	0	—	6	
	W		2					Rs16 @Rd16	—	—	↑	↑	0	—	4	
	W				4			Rs16 @(d:16, Rd16)	—	—	↑	↑	0	—	6	

オペレーション	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		
	サイズ	Rn	@Rn	@(d:16, Rn)	@-Rn/@Rn+	@aa:8/16		@(d:8, PC)	@aa	I	H	N	Z	V		C	
MOV	MOV.W Rs, @-Rd				2							↑	↑	0	—	6	
	MOV.W Rs, @aa:16					4						↑	↑	0	—	6	
POP	POP Rd				2							↑	↑	0	—	6	
PUSH	PUSH Rs				2							↑	↑	0	—	6	
ADD	ADD.B #xx:8, Rd	B	2									↑	↑	↑	↑	2	
	ADD.B Rs, Rd	B	2									↑	↑	↑	↑	2	
ADDX	ADD.W Rs, Rd	W	2									(1)	↑	↑	↑	2	
	ADDX.B #xx:8, Rd	B	2									↑	↑	(2)	↑	2	
	ADDX.B Rs, Rd	B	2									↑	↑	(2)	↑	2	
ADDS	ADDS.W #1, Rd	W	2									—	—	—	—	2	
	ADDS.W #2, Rd	W	2									—	—	—	—	2	
INC	INC.B Rd	B	2									↑	↑	↑	—	2	
DAA	DAA.B Rd	B	2									*	↑	↑	*	(3)	2
SUB	SUB.B Rs, Rd	B	2									↑	↑	↑	↑	2	
	SUB.W Rs, Rd	W	2									(1)	↑	↑	↑	2	
SUBX	SUBX.B #xx:8, Rd	B	2									↑	↑	(2)	↑	2	
	SUBX.B Rs, Rd	B	2									↑	↑	(2)	↑	2	

オペレーション	アドレッシングモード / 命令長 (バイト)						コンディションコード							実行 ステート 数	
	サイズ	Rn	@Rn	@(d:16, Rn)	@-Rn/@Rn+	@aa:8/16	@(d:8, PC)	@aa	I	H	N	Z	V		C
SUBS	W	2													2
DEC	B	2								*	↑	↑	*		2
DAS	B	2								↑	↑	↑	↑	↑	2
NEG	B	2								↑	↑	↑	↑	↑	2
CMP	B	2								↑	↑	↑	↑	↑	2
MULXU	B	2								↑	↑	↑	↑	↑	2
DIVXU	B	2								(1)	↑	↑	↑	↑	2
AND	B	2													2
OR	B	2													2
XOR	B	2													2
NOT	B	2													2
SHAL	B	2													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	C	
SHAR	B		2										↑	↑	0	↑	2	
SHLL	B		2										↑	↑	0	↑	2	
SHLR	B		2										0	↑	0	↑	2	
ROTXL	B		2										↑	↑	0	↑	2	
ROTXR	B		2										↑	↑	0	↑	2	
ROTL	B		2										↑	↑	0	↑	2	
ROTR	B		2										↑	↑	0	↑	2	
BSET	B		2														2	
								(#xx:3 of Rd8) 1										
	B		4					(#xx:3 of @Rd16) 1									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							実行ステップ数
		#xx:8/16	Rn	@Rn	@(d:16, Rn)	@-Rn/@Rn+	@aa:8/16		@(d:8, PC)	@aa	I	H	N	Z	V	
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	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 #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
BTST	BTST #xx:3, Rd								(Rn8 of @aa:8) (Rn8 of @aa:8)	-	-	-	-	-	-	8
	BTST #xx:3, @Rd		4						(#xx:3 of @Rd16) (#xx:3 of @Rd16)	-	-	-	-	-	-	6
	BTST #xx:3, @aa:8						4		(#xx:3 of @aa:8) (#xx:3 of @aa:8)	-	-	-	-	-	-	6
	BTST Rn, Rd	2							(Rn8 of Rd8) (Rn8 of Rd8)	-	-	-	-	-	-	2

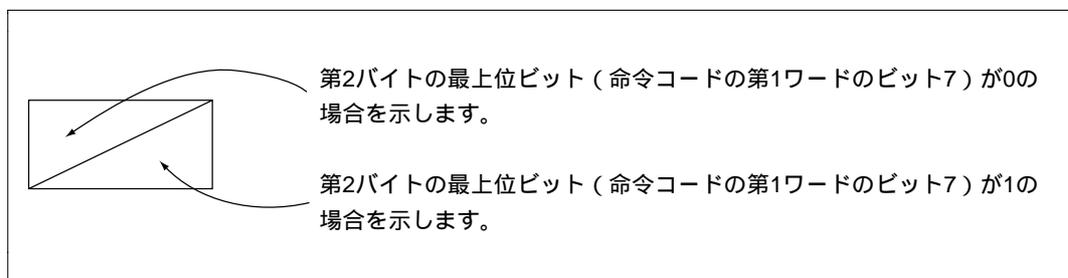
二モードック	サイズ	アドレッシングモード/命令長(バイト)						オペレーション	コンディションコード						実行ステート数	
		#xx:8/16	Rn	@Rn	@(d:16, Rn)	@-Rn/@Rn+	@aa:8/16		@(d:8, PC)	@aa	I	H	N	Z		V
BTST	BTST Rn, @Rd			4									↑	—	—	6
	BTST Rn, @aa:8						4						↑	—	—	6
BLD	BLD #xx:3, Rd		2										—	—	↑	2
	BLD #xx:3, @Rd			4									—	—	↑	6
	BLD #xx:3, @aa:8						4						—	—	↑	6
	BLD #xx:3, Rd		2										—	—	↑	2
BILD	BILD #xx:3, @Rd			4									—	—	↑	6
	BILD #xx:3, @aa:8						4						—	—	↑	6
	BILD #xx:3, Rd		2										—	—	↑	6
BST	BST #xx:3, Rd		2										—	—	—	2
	BST #xx:3, @Rd			4									—	—	—	8
	BST #xx:3, @aa:8						4						—	—	—	8
BIST	BIST #xx:3, Rd		2										—	—	—	2
	BIST #xx:3, @Rd			4									—	—	—	8
	BIST #xx:3, @aa:8						4						—	—	—	8
BAND	BAND #xx:3, Rd		2										—	—	↑	2
	BAND #xx:3, @Rd			4									—	—	↑	6
	BAND #xx:3, @aa:8						4						—	—	↑	6
	BAND #xx:3, Rd		2										—	—	↑	2
BIAND	BIAND #xx:3, @Rd			4									—	—	↑	6
	BIAND #xx:3, @aa:8						4						—	—	↑	6
	BIAND #xx:3, Rd		2										—	—	↑	2
BOR	BOR #xx:3, @Rd			4									—	—	↑	6
	BOR #xx:3, @aa:8						4						—	—	↑	6
	BOR #xx:3, Rd		2										—	—	↑	2

二一モニック	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 ステップ 数	
		#xx:8/16	Rn	@Rn	@(d:16, Rn)	@-Rn/@Rn+	@aa:8/16	@(d:8, PC)	@aa	—	I	H	N	Z	V		C
BIOR	BIOR #xx:3, Rd		2						C (#xx:3 of Rd8) C	—	—	—	—	—	—	↑	2
	BIOR #xx:3, @Rd			4					C (#xx:3 of @Rd16) C	—	—	—	—	—	—	↑	6
	BIOR #xx:3, @aa:8				4				C (#xx:3 of @aa:8) C	—	—	—	—	—	—	↑	6
BXOR	BXOR #xx:3, Rd		2						C ⊕ (#xx:3 of Rd8) C	—	—	—	—	—	—	↑	2
	BXOR #xx:3, @Rd			4					C ⊕ (#xx:3 of @Rd16) C	—	—	—	—	—	—	↑	6
	BXOR #xx:3, @aa:8				4				C ⊕ (#xx:3 of @aa:8) C	—	—	—	—	—	—	↑	6
BIXOR	BIXOR #xx:3, Rd		2						C ⊕ (#xx:3 of Rd8) C	—	—	—	—	—	—	↑	2
	BIXOR #xx:3, @Rd			4					C ⊕ (#xx:3 of @Rd16) C	—	—	—	—	—	—	↑	6
	BIXOR #xx:3, @aa:8				4				C ⊕ (#xx:3 of @aa:8) C	—	—	—	—	—	—	↑	6
Bcc	BRA d:8 (BT d:8)	—						2	PC PC+d:8	—	—	—	—	—	—	—	4
	BRN d:8 (BF d:8)	—						2	PC PC+2	—	—	—	—	—	—	—	4
	BHI d:8	—						2	if condition	—	—	—	—	—	—	—	4
	BLS d:8	—						2	is true then	—	—	—	—	—	—	—	4
	BCC d:8 (BHS d:8)	—						2	PC PC+d:8	—	—	—	—	—	—	—	4
	BCS d:8 (BLO d:8)	—						2	else next;	—	—	—	—	—	—	—	4
	BNE d:8	—						2		—	—	—	—	—	—	—	4
	BEQ d:8	—						2		—	—	—	—	—	—	—	4
	BVC d:8	—						2		—	—	—	—	—	—	—	4
	BVS d:8	—						2		—	—	—	—	—	—	—	4
	BPL d:8	—						2		—	—	—	—	—	—	—	4
	BMI d:8	—						2		—	—	—	—	—	—	—	4
	BGE d:8	—						2		N ⊕ V=0	—	—	—	—	—	—	4
	BLT d:8	—						2		N ⊕ V=1	—	—	—	—	—	—	4
	BGT d:8	—						2		Z (N ⊕ V)=0	—	—	—	—	—	—	4
BLE d:8	—						2		Z (N ⊕ V)=1	—	—	—	—	—	—	4	

オペレーション	アドレッシングモード / 命令長 (バイト)							コンディションコード							実行 ステップ 数			
	サイズ	ニーモニック	#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
	—	JSR @Rn		2								—	—	—	—	—	—	6
JSR	—	JSR @aa:16						4				—	—	—	—	—	—	8
	—	JSR @aa:8								2		—	—	—	—	—	—	8
	—	RTS									2	—	—	—	—	—	—	8
RTE	—	RTE									2	↕	↕	↕	↕	↕	↕	10

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

HI	LO	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	SLEEP	STC	LDC	LDC	ORC	XORC	ANDC	LDC	ADD	INC	ADD	ADD	MOV	ADDX		
1	SHLL SHAR	SHLR SHAR	ROTL ROTR	ROTXR ROTR	OR	OR	XOR	AND	NOT NEG	SUB	DEC	SUBS	SUBS	CMP	SUBX		
2																	
3																	
4	BRA	BRN	BHI	BLS	BCC	BCC	BCS	BNE	BEQ	BVC	BVS	BPL	BMI	BGE	BLT	BGT	BLE
5	MULXU	DIVXU			RTS	RTS	BSR	RTE				JMP				JSR	
6	BSET	BNOT	BCLR	BTST					BST								
7					BOR BIOR	BAND BIAND	BXOR BIXOR	BAND BIAND	BLT BILD					MOV*			
8																	
9																	
A																	
B																	
C																	
D																	
E																	
F																	

【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	2または3*
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)

命令	ニーモニック	命令	分岐アドレ	スタック	バイトデー	ワードデー	内部動作
		フェッチ	スリード	操作	タアクセス	タアクセス	
		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		

命令	ニーモニック	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	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		

命令	ニーモニック	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	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
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			

命令	ニーモニック	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	N	
MOV	MOV.B Rs, @Rd	1			1			
	MOV.B Rs, @(d:16, Rd)	2			1			
	MOV.B Rs, @-Rd	1			1		2	
	MOV.B Rs, @aa:8	1			1			
	MOV.B Rs, @aa:16	2			1			
	MOV.W #xx:16, Rd	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	2
	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	2
	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						
SUBS	SUBS.W #1, Rd	1						
	SUBS.W #2, Rd	1						

命令	ニーモニック	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	N
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】 * 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	WEGR	WKEGS7	WKEGS6	WKEGS5	WKEGS4	WKEGS3	WKEGS2	WKEGS1	WKEGS0	システム コントロール
H'91	SPCR			SPC32	SPC31	SCINV3	SCINV2	SCINV1	SCINV0	SCI
H'92	CWOSR								CWOS	タイマA
H'93										
H'94										
H'95	ECCSR	OVH	OVL		CH2	CUEH	CUEL	CRCH	CRCL	非同期 イベント カウンタ
H'96	ECH	ECH7	ECH6	ECH5	ECH4	ECH3	ECH2	ECH1	ECH0	
H'97	ECL	ECL7	ECL6	ECL5	ECL4	ECL3	ECL2	ECL1	ECL0	
H'98	SMR31	COM31	CHR31	PE31	PM31	STOP31	MP31	CKS311	CKS310	SCI31
H'99	BRR31	BRR317	BRR316	BRR315	BRR314	BRR313	BRR312	BRR311	BRR310	
H'9A	SCR31	TIE31	RIE31	TE31	RE31	MPIE31	TEIE31	CKE311	CKE310	
H'9B	TDR31	TDR317	TDR316	TDR315	TDR314	TDR313	TDR312	TDR311	TDR310	
H'9C	SSR31	TDRE31	RDRF31	OER31	FER31	PER31	TEND31	MPBR31	MPBT31	
H'9D	RDR31	RDR317	RDR316	RDR315	RDR314	RDR313	RDR312	RDR311	RDR310	
H'9E										
H'9F										
H'A0	SCR1	SNC1	SNC0	MRKON	LTCH	CKS3	CKS2	CKS1	CKS0	SCI1
H'A1	SCSR1		SOL	ORER				MTRF	STF	
H'A2	SDRU	SDRU7	SDRU6	SDRU5	SDRU4	SDRU3	SDRU2	SDRU1	SDRU0	
H'A3	SDRL	SDRL7	SDRL6	SDRL5	SDRL4	SDRL3	SDRL2	SDRL1	SDRL0	
H'A4										
H'A5										
H'A6										
H'A7										
H'A8	SMR32	COM32	CHR32	PE32	PM32	STOP32	MP32	CKS321	CKS320	SCI32
H'A9	BRR32	BRR327	BRR326	BRR325	BRR324	BRR323	BRR322	BRR321	BRR320	
H'AA	SCR32	TIE32	RIE32	TE32	RE32	MPIE32	TEIE32	CKE321	CKE320	
H'AB	TDR32	TDR327	TDR326	TDR325	TDR324	TDR323	TDR322	TDR321	TDR320	
H'AC	SSR32	TDRE32	RDRF32	OER32	FER32	PER32	TEND32	MPBR32	MPBT32	
H'AD	RDR32	RDR327	RDR326	RDR325	RDR324	RDR323	RDR322	RDR321	RDR320	
H'AE										
H'AF										

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'B0	TMA	TMA7	TMA6	TMA5		TMA3	TMA2	TMA1	TMA0	タイマ A
H'B1	TCA	TCA7	TCA6	TCA5	TCA4	TCA3	TCA2	TCA1	TCA0	
H'B2	TCSRW	B6WI	TCWE	B4WI	TCSRWE	B2WI	WDON	BOWI	WRST	ウォッチ ドッグタイマ
H'B3	TCW	TCW7	TCW6	TCW5	TCW4	TCW3	TCW2	TCW1	TCW0	
H'B4	TMC	TMC7	TMC6	TMC5			TMC2	TMC1	TMC0	タイマ C
H'B5	TCC/TLC	TCC7/ TLC7	TCC6/ TLC6	TCC5/ TLC5	TCC4/ TLC4	TCC3/ TLC3	TCC2/ TLC2	TCC1/ TLC1	TCC0/ TLC0	
H'B6	TCRF	TOLH	CKSH2	CKSH1	CKSH0	TOLL	CKSL2	CKSL1	CKSL0	タイマ F
H'B7	TCSRFB	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	OCRFH	OCRFH7	OCRFH6	OCRFH5	OCRFH4	OCRFH3	OCRFH2	OCRFH1	OCRFH0	
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	LPCR	DTS1	DTS0	CMX	SGX	SGS3	SGS2	SGS1	SGS0	LCD
H'C1	LCR		PSW	ACT	DISP	CKS3	CKS2	CKS1	CKS0	コントローラ / ドライバ
H'C2	LCR2	LCDAB			SUPS	CDS3	CDS2	CDS1	CDS0	
H'C3										
H'C4	ADRRH	ADR9	ADR8	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	A/D 変換器
H'C5	ADRRL	ADR1	ADR0							
H'C6	AMR	CKS	TRGE			CH3	CH2	CH1	CH0	
H'C7	ADSR	ADSF								
H'C8	PMR1	IRQ3	IRQ2	IRQ1	IRQ4	TMIG	TMOFH	TMOFL	TMOW	I/O ポート
H'C9	PMR2			POF1			SO1	SI1	SCK1	
H'CA	PMR3	AEVL	AEVH	WDCKS	NCS	IRQ0	RESO	UD	PWM	
H'CB	PMR4	NMOD7	NMOD6	NMOD5	NMOD4	NMOD3	NMOD2	NMOD1	NMOD0	
H'CC	PMR5	WKP ₇	WKP ₆	WKP ₅	WKP ₄	WKP ₃	WKP ₂	WKP ₁	WKP ₀	
H'CD										
H'CE										
H'CF										
H'D0	PWCR							PWCR1	PWCR0	14 ビット PWM
H'D1	PWDRU			PWDRU5	PWDRU4	PWDRU3	PWDRU2	PWDRU1	PWDRU0	
H'D2	PWDRL	PWDRL7	PWDRL6	PWDRL5	PWDRL4	PWDRL3	PWDRL2	PWDRL1	PWDRL0	
H'D3										
H'D4	PDR1	P1 ₇	P1 ₆	P1 ₅	P1 ₄	P1 ₃	P1 ₂	P1 ₁	P1 ₀	I/O ポート

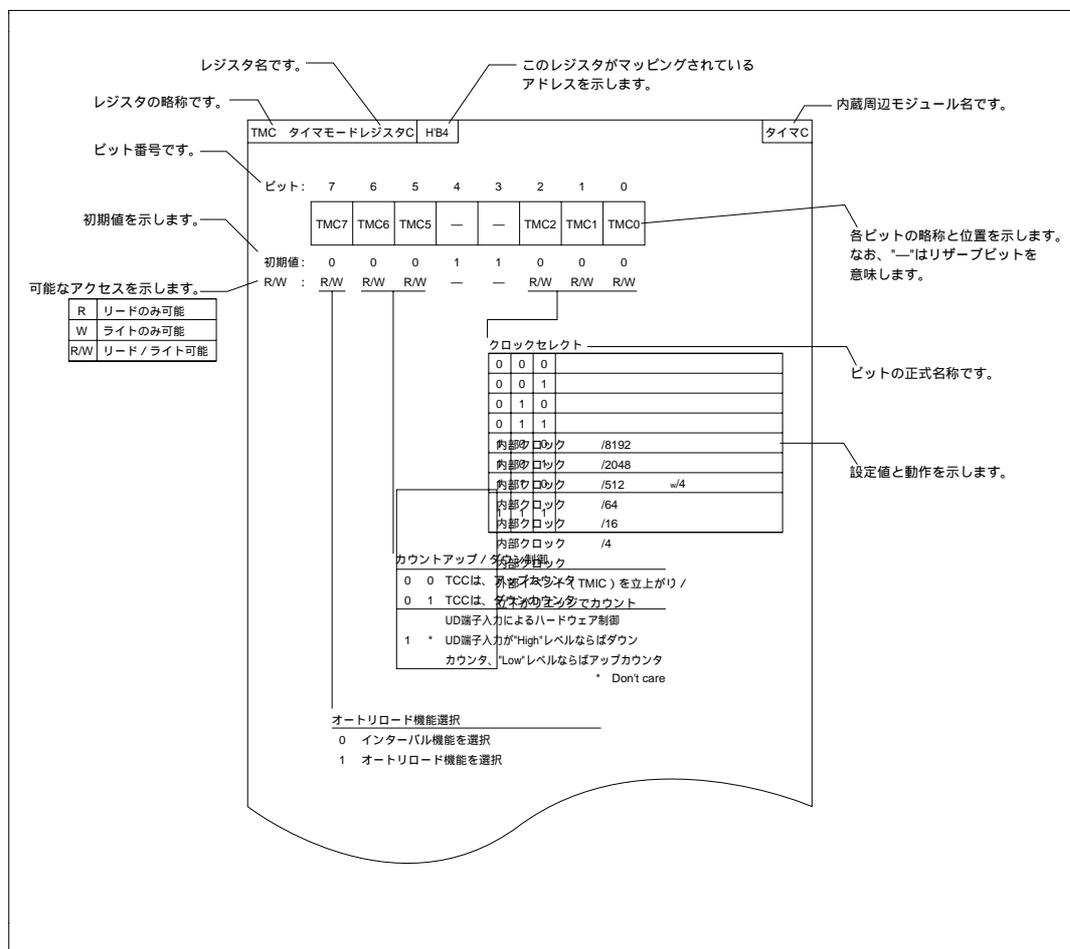
下位 アドレス	レジスタ名	ビット名								モジュール名
		ビット7	ビット6	ビット5	ビット4	ビット3	ビット2	ビット1	ビット0	
H'D5	PDR2	P2 ₇	P2 ₆	P2 ₅	P2 ₄	P2 ₃	P2 ₂	P2 ₁	P2 ₀	I/Oポート
H'D6	PDR3	P3 ₇	P3 ₆	P3 ₅	P3 ₄	P3 ₃	P3 ₂	P3 ₁	P3 ₀	
H'D7	PDR4					P4 ₃	P4 ₂	P4 ₁	P4 ₀	
H'D8	PDR5	P5 ₇	P5 ₆	P5 ₅	P5 ₄	P5 ₃	P5 ₂	P5 ₁	P5 ₀	
H'D9	PDR6	P6 ₇	P6 ₆	P6 ₅	P6 ₄	P6 ₃	P6 ₂	P6 ₁	P6 ₀	
H'DA	PDR7	P7 ₇	P7 ₆	P7 ₅	P7 ₄	P7 ₃	P7 ₂	P7 ₁	P7 ₀	
H'DB	PDR8	P8 ₇	P8 ₆	P8 ₅	P8 ₄	P8 ₃	P8 ₂	P8 ₁	P8 ₀	
H'DC	PDR9	P9 ₇	P9 ₆	P9 ₅	P9 ₄	P9 ₃	P9 ₂	P9 ₁	P9 ₀	
H'DD	PDRA					PA ₃	PA ₂	PA ₁	PA ₀	
H'DE	PDRB	PB ₇	PB ₆	PB ₅	PB ₄	PB ₃	PB ₂	PB ₁	PB ₀	
H'DF	PDRC					PC ₃	PC ₂	PC ₁	PC ₀	
H'E0	PUCR1	PUCR1 ₇	PUCR1 ₆	PUCR1 ₅	PUCR1 ₄	PUCR1 ₃	PUCR1 ₂	PUCR1 ₁	PUCR1 ₀	I/Oポート
H'E1	PUCR3	PUCR3 ₇	PUCR3 ₆	PUCR3 ₅	PUCR3 ₄	PUCR3 ₃	PUCR3 ₂	PUCR3 ₁	PUCR3 ₀	
H'E2	PUCR5	PUCR5 ₇	PUCR5 ₆	PUCR5 ₅	PUCR5 ₄	PUCR5 ₃	PUCR5 ₂	PUCR5 ₁	PUCR5 ₀	
H'E3	PUCR6	PUCR6 ₇	PUCR6 ₆	PUCR6 ₅	PUCR6 ₄	PUCR6 ₃	PUCR6 ₂	PUCR6 ₁	PUCR6 ₀	
H'E4	PCR1	PCR1 ₇	PCR1 ₆	PCR1 ₅	PCR1 ₄	PCR1 ₃	PCR1 ₂	PCR1 ₁	PCR1 ₀	
H'E5	PCR2	PCR2 ₇	PCR2 ₆	PCR2 ₅	PCR2 ₄	PCR2 ₃	PCR2 ₂	PCR2 ₁	PCR2 ₀	
H'E6	PCR3	PCR3 ₇	PCR3 ₆	PCR3 ₅	PCR3 ₄	PCR3 ₃	PCR3 ₂	PCR3 ₁	PCR3 ₀	
H'E7	PCR4						PCR4 ₂	PCR4 ₁	PCR4 ₀	
H'E8	PCR5	PCR5 ₇	PCR5 ₆	PCR5 ₅	PCR5 ₄	PCR5 ₃	PCR5 ₂	PCR5 ₁	PCR5 ₀	
H'E9	PCR6	PCR6 ₇	PCR6 ₆	PCR6 ₅	PCR6 ₄	PCR6 ₃	PCR6 ₂	PCR6 ₁	PCR6 ₀	
H'EA	PCR7	PCR7 ₇	PCR7 ₆	PCR7 ₅	PCR7 ₄	PCR7 ₃	PCR7 ₂	PCR7 ₁	PCR7 ₀	
H'EB	PCR8	PCR8 ₇	PCR8 ₆	PCR8 ₅	PCR8 ₄	PCR8 ₃	PCR8 ₂	PCR8 ₁	PCR8 ₀	
H'EC	PCR9	PCR9 ₇	PCR9 ₆	PCR9 ₅	PCR9 ₄	PCR9 ₃	PCR9 ₂	PCR9 ₁	PCR9 ₀	
H'ED	PCRA					PCRA ₃	PCRA ₂	PCRA ₁	PCRA ₀	
H'EE										
H'EF										
H'F0	SYSCR1	SSBY	STS2	STS1	STS0	LSON		MA1	MA0	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	IENFLL	IENFCL	IENFEC	
H'F5										
H'F6	IRR1	IRRRTA	IRRS1		IRRI4	IRRI3	IRRI2	IRRI1	IRRI0	
H'F7	IRR2	IRRRTD	IRRAD		IRRTG	IRRTFH	IRRTFL	IRRTC	IRREC	
H'F8										

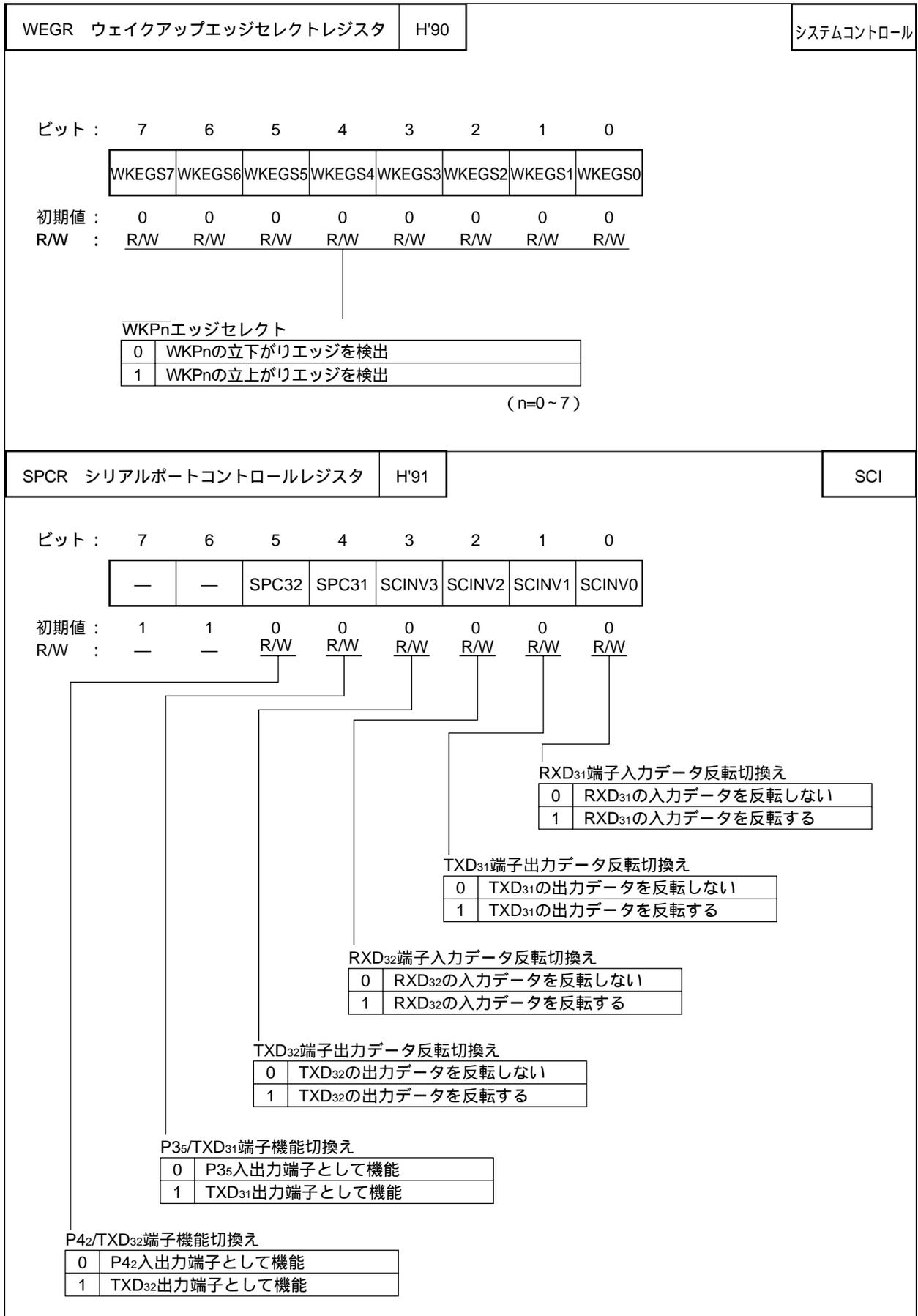
下位 アドレス	レジスタ名	ビット名								モジュール名
		ビット7	ビット6	ビット5	ビット4	ビット3	ビット2	ビット1	ビット0	
H'F9	IWPR	IWPF7	IWPF6	IWPF5	IWPF4	IWPF3	IWPF2	IWPF1	IWPF0	システム
H'FA	CKSTPR1	S1CKSTP	S31CKSTP	S32CKSTP	ADCKSTP	TGCKSTP	TFCKSTP	TCCKSTP	TACKSTP	コントロール
H'FB	CKSTPR2					AECKSTP	WDCKSTP	PWCKSTP	LDCKSTP	
H'FC										
H'FD										
H'FE										
H'FF										

記号説明

SCI : シリアルコミュニケーションインタフェース

B.2 機能一覧

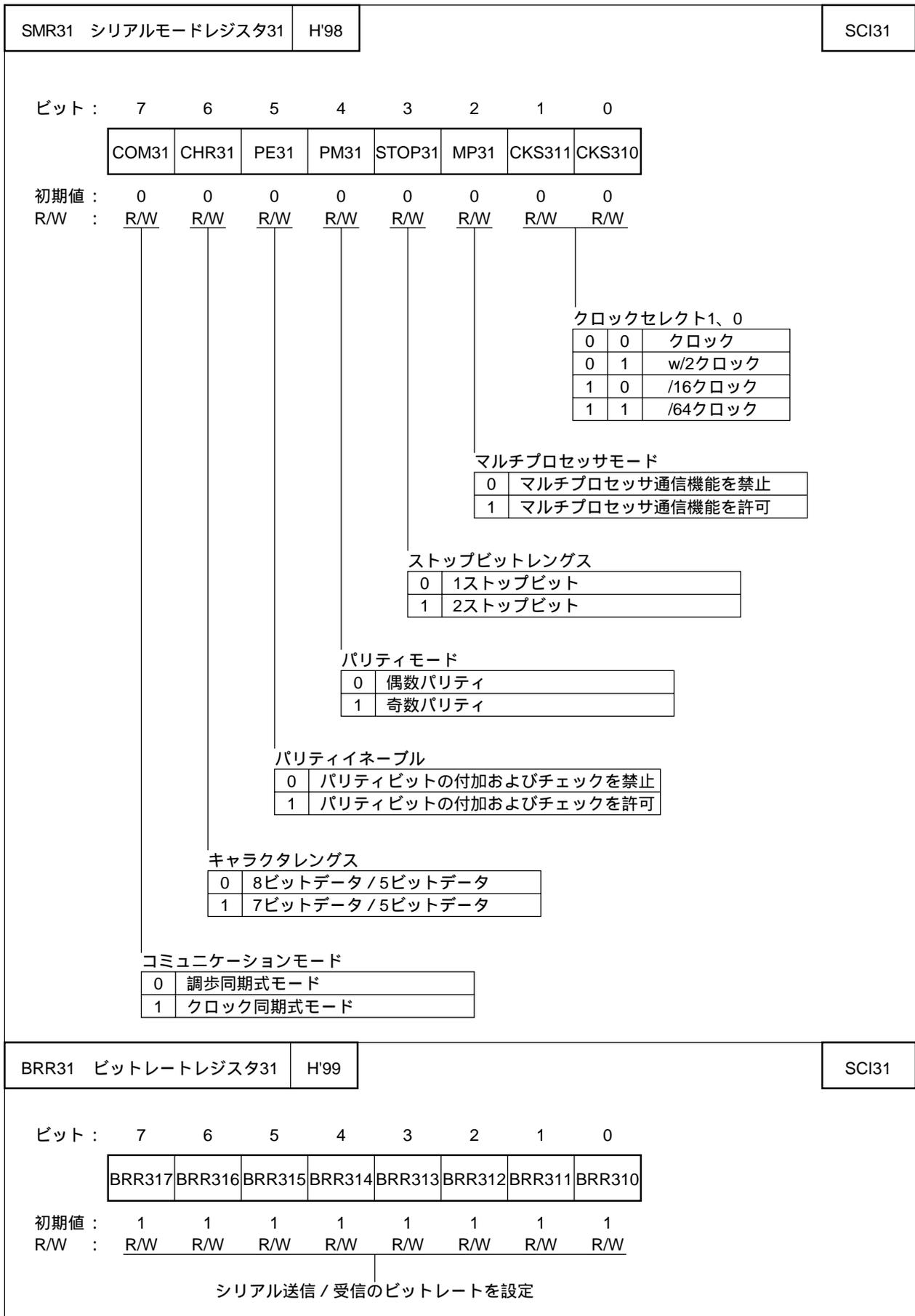




CWOSR サブクロック出力セレクトレジスタ		H'92						タイマA				
ビット:	7	6	5	4	3	2	1	0				
	—	—	—	—	—	—	—	CWOS				
初期値:	1	1	1	1	1	1	1	0				
R/W:	R	R	R	R	R	R	R	R/W				
								TMOW端子クロックセレクト <table border="1"> <tr> <td>0</td> <td>TMAで指定したクロックを出力する</td> </tr> <tr> <td>1</td> <td>wを出力する</td> </tr> </table>	0	TMAで指定したクロックを出力する	1	wを出力する
0	TMAで指定したクロックを出力する											
1	wを出力する											

ECCSR イベントカウンタコントロール/ステータスレジスタ	H'95	AEC																																				
<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;">OVH</td> <td style="padding: 2px 10px;">OVL</td> <td style="padding: 2px 10px;">—</td> <td style="padding: 2px 10px;">CH2</td> <td style="padding: 2px 10px;">CUEH</td> <td style="padding: 2px 10px;">CUEL</td> <td style="padding: 2px 10px;">CRCH</td> <td style="padding: 2px 10px;">CRCL</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;"> <div style="margin-bottom: 10px;"> <p style="text-align: right; margin-right: 20px;">カウンタリセット制御</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 10px;">ECLをリセット</td></tr> <tr><td style="padding: 2px 5px;">1</td><td style="padding: 2px 10px;">ECLのリセットを解除し、カウントアップ機能を許可</td></tr> </table> </div> <div style="margin-bottom: 10px;"> <p style="text-align: right; margin-right: 20px;">カウンタリセット制御</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 10px;">ECHをリセット</td></tr> <tr><td style="padding: 2px 5px;">1</td><td style="padding: 2px 10px;">ECHのリセットを解除し、カウントアップ機能を許可</td></tr> </table> </div> <div style="margin-bottom: 10px;"> <p style="text-align: right; margin-right: 20px;">カウントアップイネーブルL</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 10px;">ECLのイベントクロックの入力を禁止。ECLの値を保持</td></tr> <tr><td style="padding: 2px 5px;">1</td><td style="padding: 2px 10px;">ECLのイベントクロックの入力を許可</td></tr> </table> </div> <div style="margin-bottom: 10px;"> <p style="text-align: right; margin-right: 20px;">カウントアップイネーブルH</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 10px;">ECHのイベントクロックの入力を禁止。ECHの値を保持</td></tr> <tr><td style="padding: 2px 5px;">1</td><td style="padding: 2px 10px;">ECHのイベントクロックの入力を許可</td></tr> </table> </div> <div style="margin-bottom: 10px;"> <p style="text-align: right; margin-right: 20px;">チャンネル選択</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 10px;">ECH、ECLを連結した1チャンネルの16ビットイベントカウンタとして使用します</td></tr> <tr><td style="padding: 2px 5px;">1</td><td style="padding: 2px 10px;">ECH、ECLを独立した2チャンネルの8ビットイベントカウンタとして使用します</td></tr> </table> </div> <div style="margin-bottom: 10px;"> <p style="text-align: right; margin-right: 20px;">カウンタオーバーフローL</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 10px;">ECLがオーバーフローしていない 〔クリア条件〕 OVLが"1"の状態でもOVLをリードした後、OVLに"0"をライトしたとき</td></tr> <tr><td style="padding: 2px 5px;">1</td><td style="padding: 2px 10px;">ECLがオーバーフローした 〔セット条件〕 CH2を"1"にした状態でECLの値がH'FF H'00になったとき</td></tr> </table> </div> <div> <p style="text-align: right; margin-right: 20px;">カウンタオーバーフローH</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 10px;">ECHがオーバーフローしていない 〔クリア条件〕 OVHが"1"の状態でもOVHをリードした後、OVHに"0"を書き込んだとき</td></tr> <tr><td style="padding: 2px 5px;">1</td><td style="padding: 2px 10px;">ECHがオーバーフローした 〔セット条件〕 ECHの値がH'FF H'00になったとき</td></tr> </table> </div> </div> <p style="margin-top: 10px;">【注】 * フラグクリアのための"0"ライトのみ可能</p>			OVH	OVL	—	CH2	CUEH	CUEL	CRCH	CRCL	0	ECLをリセット	1	ECLのリセットを解除し、カウントアップ機能を許可	0	ECHをリセット	1	ECHのリセットを解除し、カウントアップ機能を許可	0	ECLのイベントクロックの入力を禁止。ECLの値を保持	1	ECLのイベントクロックの入力を許可	0	ECHのイベントクロックの入力を禁止。ECHの値を保持	1	ECHのイベントクロックの入力を許可	0	ECH、ECLを連結した1チャンネルの16ビットイベントカウンタとして使用します	1	ECH、ECLを独立した2チャンネルの8ビットイベントカウンタとして使用します	0	ECLがオーバーフローしていない 〔クリア条件〕 OVLが"1"の状態でもOVLをリードした後、OVLに"0"をライトしたとき	1	ECLがオーバーフローした 〔セット条件〕 CH2を"1"にした状態でECLの値がH'FF H'00になったとき	0	ECHがオーバーフローしていない 〔クリア条件〕 OVHが"1"の状態でもOVHをリードした後、OVHに"0"を書き込んだとき	1	ECHがオーバーフローした 〔セット条件〕 ECHの値がH'FF H'00になったとき
OVH	OVL	—	CH2	CUEH	CUEL	CRCH	CRCL																															
0	ECLをリセット																																					
1	ECLのリセットを解除し、カウントアップ機能を許可																																					
0	ECHをリセット																																					
1	ECHのリセットを解除し、カウントアップ機能を許可																																					
0	ECLのイベントクロックの入力を禁止。ECLの値を保持																																					
1	ECLのイベントクロックの入力を許可																																					
0	ECHのイベントクロックの入力を禁止。ECHの値を保持																																					
1	ECHのイベントクロックの入力を許可																																					
0	ECH、ECLを連結した1チャンネルの16ビットイベントカウンタとして使用します																																					
1	ECH、ECLを独立した2チャンネルの8ビットイベントカウンタとして使用します																																					
0	ECLがオーバーフローしていない 〔クリア条件〕 OVLが"1"の状態でもOVLをリードした後、OVLに"0"をライトしたとき																																					
1	ECLがオーバーフローした 〔セット条件〕 CH2を"1"にした状態でECLの値がH'FF H'00になったとき																																					
0	ECHがオーバーフローしていない 〔クリア条件〕 OVHが"1"の状態でもOVHをリードした後、OVHに"0"を書き込んだとき																																					
1	ECHがオーバーフローした 〔セット条件〕 ECHの値がH'FF H'00になったとき																																					

ECH イベントカウンタH H'96								AEC
ビット :	7	6	5	4	3	2	1	0
	ECH7	ECH6	ECH5	ECH4	ECH3	ECH2	ECH1	ECH0
初期値 :	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R
↓ カウント値								
【注】 ECHを上位、ECLを下位とする16ビットイベントカウンタ（EC）としても使用できます。								
ECL イベントカウンタL H'97								AEC
ビット :	7	6	5	4	3	2	1	0
	ECL7	ECL6	ECL5	ECL4	ECL3	ECL2	ECL1	ECL0
初期値 :	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R
↓ カウント値								
【注】 ECHを上位、ECLを下位とする16ビットイベントカウンタ（EC）としても使用できます。								



SCR31 シリアルコントロールレジスタ31				H'9A				SCI31							
ビット :								7	6	5	4	3	2	1	0
TIE31		RIE31		TE31		RE31		MPIE31		TEIE31		CKE311		CKE310	
初期値 :		0		0		0		0		0		0		0	
R/W :		R/W		R/W		R/W		R/W		R/W		R/W		R/W	
クロックイネーブル															
ビット1		ビット0		説明											
CKE311		CKE310		コミュニケーションモード		クロックソース		SCK ₃₁ 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(使用禁止)									
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(使用禁止)									
				クロック同期式		リザーブ(使用禁止)									
トランスミットエンドインタラプトイネーブル															
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													
マルチプロセッサインタラプトイネーブル															
0		マルチプロセッサ割込み要求を禁止 (通常受信動作) 〔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													
レーシブイネーブル															
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)													
トランスミットイネーブル															
0		送信動作を禁止 (TXD端子は入出力ポート)													
1		送信動作を許可 (TXD端子はトランスミットデータ端子)													
レーシブインタラプトイネーブル															
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													
トランスミットインタラプトイネーブル															
0		送信データエンプティ割込み要求 (TXI) の禁止													
1		送信データエンプティ割込み要求 (TXI) の許可													

TDR31 トランスミットデータレジスタ31	H'9B	SCI31								
<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;">TDR317</td> <td style="padding: 2px 5px;">TDR316</td> <td style="padding: 2px 5px;">TDR315</td> <td style="padding: 2px 5px;">TDR314</td> <td style="padding: 2px 5px;">TDR313</td> <td style="padding: 2px 5px;">TDR312</td> <td style="padding: 2px 5px;">TDR311</td> <td style="padding: 2px 5px;">TDR310</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>			TDR317	TDR316	TDR315	TDR314	TDR313	TDR312	TDR311	TDR310
TDR317	TDR316	TDR315	TDR314	TDR313	TDR312	TDR311	TDR310			

SSR31 シリアルステータスレジスタ31				H'9C				SCI31							
ビット :								7	6	5	4	3	2	1	0
TDRE31		RDRF31		OER31		FER31		PER31		TEND31		MPBR31		MPBT31	
初期値 :		1		0		0		0		0		1		0	
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) TDRE31 = "1"の状態をリードした後、TDRE31に"0"をライトしたとき (2) 命令でTDR31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31 (SCR31) のTE31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、TDRE31が"1"であったとき
														パリティエラー	
														0	受信中、または正常に受信完了 〔クリア条件〕 PER31 = "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"の数がシリアルモードレジスタ (SMR31) のパリティモード (PM31) 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、または正常に受信完了 〔クリア条件〕 FER31 = "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〕 OER31 = "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〕 RDRF31が"1"の状態での次のシリアル受信を完了したとき
														レシーブデータレジスタフル	
														0	RDR31に受信データ未格納 〔クリア条件〕 (1) RDRF31 = "1"の状態をリードした後、"0"をライトしたとき (2) 命令でRDR31のデータをリードしたとき
														1	RDR31に受信データ格納 〔セット条件〕 受信が正常終了し、RSR31からRDR31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	TDR31にライトされた送信データがTSR31に転送されていない 〔クリア条件〕 (1) TDRE31 = "1"の状態をリードした後、"0"をライトしたとき (2) 命令でTDR31へデータをライトしたとき
														1	TDR31に送信データがライトされていない、またはTDR31にライトされた送信データがTSR31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31 (SCR31) のTE31が"0"のとき (2) TDR31からTSR31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															

RDR31 レシーブデータレジスタ31 H'9D		SCI31
--------------------------	--	-------

ビット： 7 6 5 4 3 2 1 0

RDR317	RDR316	RDR315	RDR314	RDR313	RDR312	RDR311	RDR310
--------	--------	--------	--------	--------	--------	--------	--------

初期値： 0 0 0 0 0 0 0 0
R/W： R R R R R R R R

シリアル受信データを格納

SCR1 シリアルコントロールレジスタ1 H'A0		SCI1
---------------------------	--	------

ビット： 7 6 5 4 3 2 1 0

SNC1	SNC0	MRKON	LTCH	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

クロック選択2~0

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 クロック同期
				= 2.5MHz
0	0	0	/1024	409.6 μs
0	0	1	/256	102.4 μs
0	1	0	/64	25.6 μs
0	1	1	/32	12.8 μs
1	0	0	/16	6.4 μs
1	0	1	/8	3.2 μs
1	1	0	/4	1.6 μs
1	1	1	w/4	122 μs

クロックソース選択

0	クロックソースはプリスケアラS、SCK _i 端子は出力
1	クロックソースは外部クロック、SCK _i 端子は入力

LATCH TAIL選択

0	HOLD TAILを出力
1	LATCH TAILを出力

TAIL MARK制御

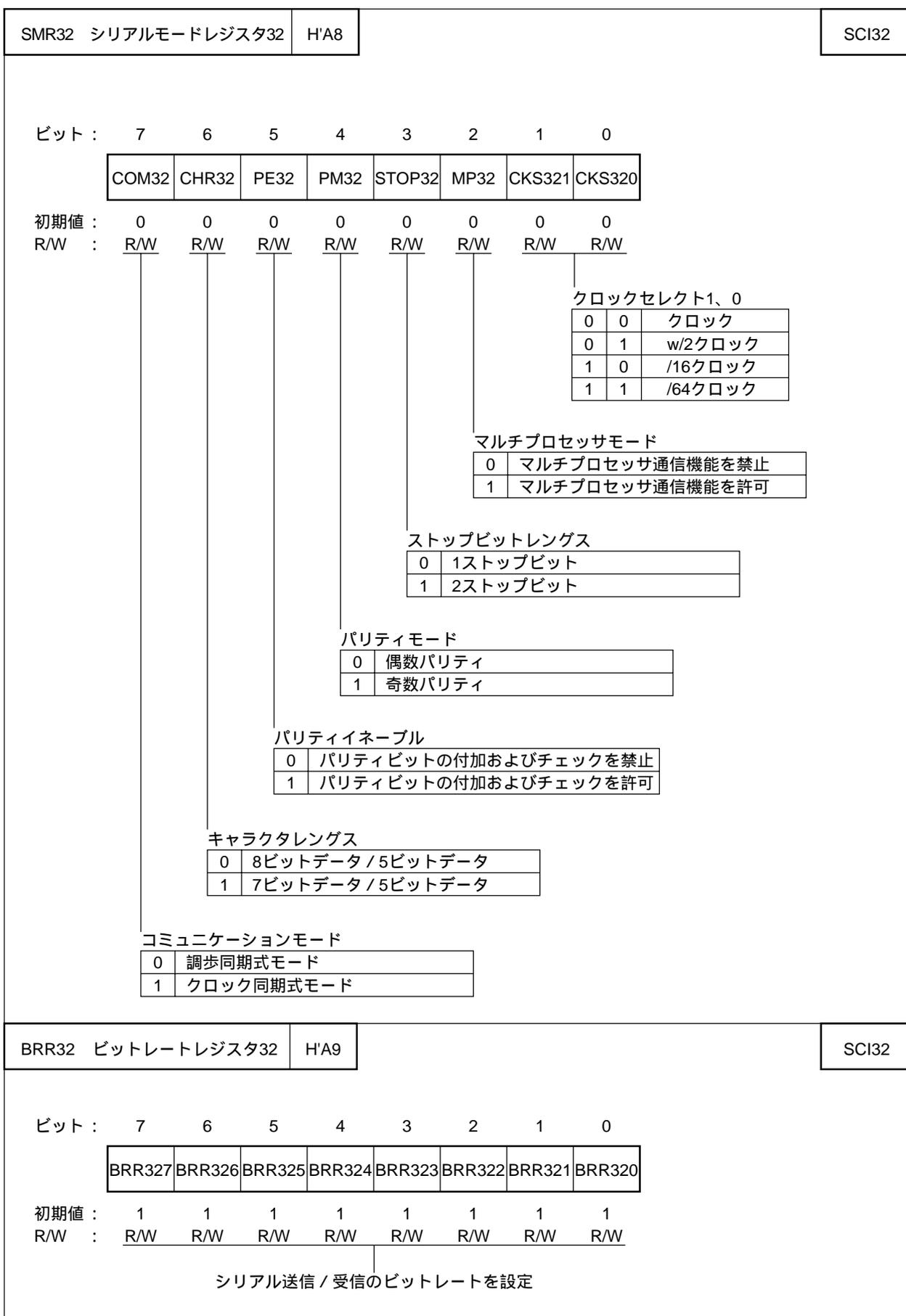
0	TAIL MARKを非出力（クロック同期モード）
1	TAIL MARKを出力（SSBモード）

動作モード選択

0	0	8ビットクロック同期モード
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	リザーブ

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	—	—	—	MTRF	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 <table border="1"> <tr> <td rowspan="2">0</td> <td>リード時</td> <td>転送動作は停止</td> </tr> <tr> <td>ライト時</td> <td>無効</td> </tr> <tr> <td rowspan="2">1</td> <td>リード時</td> <td>転送動作中</td> </tr> <tr> <td>ライト時</td> <td>転送動作を開始</td> </tr> </table>		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
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																
							TAIL MARK送信フラグ <table border="1"> <tr> <td>0</td> <td>転送待ち状態および8ビットまたは16ビットデータ転送中</td> </tr> <tr> <td>1</td> <td>TAIL MARK送信中</td> </tr> </table>		0	転送待ち状態および8ビットまたは16ビットデータ転送中	1	TAIL MARK送信中						
0	転送待ち状態および8ビットまたは16ビットデータ転送中																	
1	TAIL MARK送信中																	
							オーバランエラーフラグ <table border="1"> <tr> <td rowspan="2">0</td> <td colspan="2">〔クリア条件〕</td> </tr> <tr> <td colspan="2">"1"をリード後、"0"をライトしたとき</td> </tr> <tr> <td rowspan="2">1</td> <td colspan="2">〔セット条件〕</td> </tr> <tr> <td colspan="2">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</td> </tr> </table>		0	〔クリア条件〕		"1"をリード後、"0"をライトしたとき		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	
0	〔クリア条件〕																	
	"1"をリード後、"0"をライトしたとき																	
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																	
							拡張データビット <table border="1"> <tr> <td rowspan="2">0</td> <td>リード時</td> <td>SO_i端子の出力が"Low"レベル</td> </tr> <tr> <td>ライト時</td> <td>SO_i端子の出力を"Low"レベルに変更</td> </tr> <tr> <td rowspan="2">1</td> <td>リード時</td> <td>SO_i端子の出力が"High"レベル</td> </tr> <tr> <td>ライト時</td> <td>SO_i端子の出力を"High"レベルに変更</td> </tr> </table>		0	リード時	SO _i 端子の出力が"Low"レベル	ライト時	SO _i 端子の出力を"Low"レベルに変更	1	リード時	SO _i 端子の出力が"High"レベル	ライト時	SO _i 端子の出力を"High"レベルに変更
0	リード時	SO _i 端子の出力が"Low"レベル																
	ライト時	SO _i 端子の出力を"Low"レベルに変更																
1	リード時	SO _i 端子の出力が"High"レベル																
	ライト時	SO _i 端子の出力を"High"レベルに変更																
【注】 * フラグをクリアにするための"0"ライトのみ可能																		

SDRU シリアルデータレジスタU	H'A2		SCI1								
<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;">SDRU7</td> <td style="padding: 2px 5px;">SDRU6</td> <td style="padding: 2px 5px;">SDRU5</td> <td style="padding: 2px 5px;">SDRU4</td> <td style="padding: 2px 5px;">SDRU3</td> <td style="padding: 2px 5px;">SDRU2</td> <td style="padding: 2px 5px;">SDRU1</td> <td style="padding: 2px 5px;">SDRU0</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/W R/W R/W R/W R/W R/W R/W</p> <p style="margin-left: 100px;"> </p> <p style="margin-left: 100px;">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</p>				SDRU7	SDRU6	SDRU5	SDRU4	SDRU3	SDRU2	SDRU1	SDRU0
SDRU7	SDRU6	SDRU5	SDRU4	SDRU3	SDRU2	SDRU1	SDRU0				
SDRL シリアルデータレジスタL	H'A3		SCI1								
<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;">SDRL7</td> <td style="padding: 2px 5px;">SDRL6</td> <td style="padding: 2px 5px;">SDRL5</td> <td style="padding: 2px 5px;">SDRL4</td> <td style="padding: 2px 5px;">SDRL3</td> <td style="padding: 2px 5px;">SDRL2</td> <td style="padding: 2px 5px;">SDRL1</td> <td style="padding: 2px 5px;">SDRL0</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/W R/W R/W R/W R/W R/W R/W</p> <p style="margin-left: 100px;"> </p> <p style="margin-left: 100px;">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</p>				SDRL7	SDRL6	SDRL5	SDRL4	SDRL3	SDRL2	SDRL1	SDRL0
SDRL7	SDRL6	SDRL5	SDRL4	SDRL3	SDRL2	SDRL1	SDRL0				



SCR32 シリアルコントロールレジスタ32	H'AA		SCI32
------------------------	------	--	-------

ビット： 7 6 5 4 3 2 1 0

TIE32	RIE32	TE32	RE32	MPIE32	TEIE32	CKE321	CKE320
-------	-------	------	------	--------	--------	--------	--------

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

クロックイネーブル

ビット1	ビット0	説明		
CKE321	CKE320	コミュニケーションモード	クロックソース	SCK ₃₂ 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（使用禁止）	
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（使用禁止）	
		クロック同期式	リザーブ（使用禁止）	

トランスミットエンドインタラプトイネーブル

0	送信終了割り込み要求（TEI）を禁止
1	送信終了割り込み要求（TEI）を許可

マルチプロセッサインタラプトイネーブル

0	マルチプロセッサ割り込み要求を禁止（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り込み要求を許可 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り込み要求（RXI）、受信エラー割り込み要求（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

レーシブイネーブル

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）

トランスミットイネーブル

0	送信動作を禁止（TXD端子は入出力ポート）
1	送信動作を許可（TXD端子はトランスミットデータ端子）

レーシブインタラプトイネーブル

0	受信データフル割り込み要求（RXI）、および受信エラー割り込み要求（ERI）を禁止
1	受信データフル割り込み要求（RXI）、および受信エラー割り込み要求（ERI）を許可

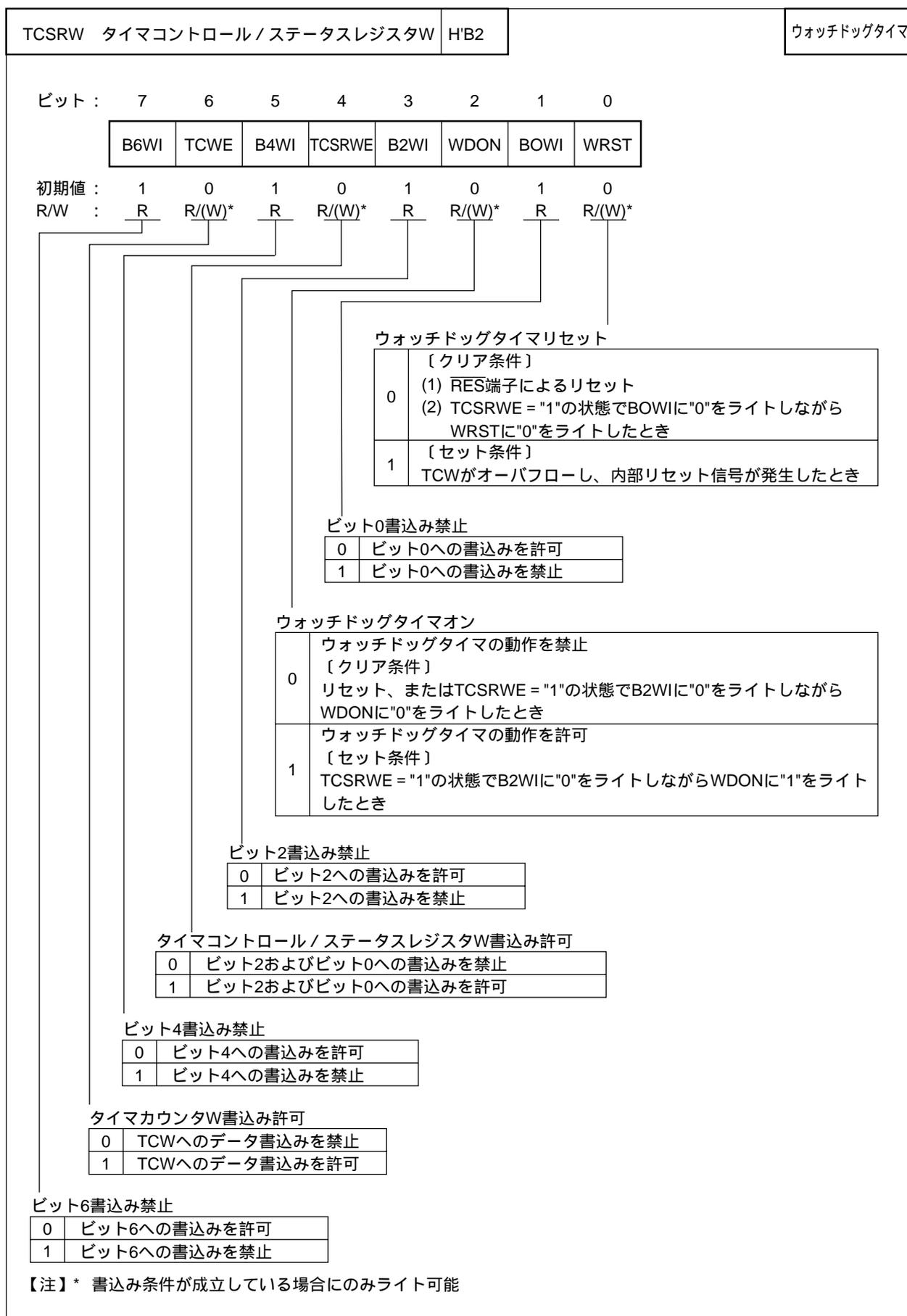
トランスミットインタラプトイネーブル

0	送信データエンプティ割り込み要求（TXI）の禁止
1	送信データエンプティ割り込み要求（TXI）の許可

RDR32 レシーブデータレジスタ32 H'AD	SCI32								
<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;">RDR327</td> <td style="padding: 2px;">RDR326</td> <td style="padding: 2px;">RDR325</td> <td style="padding: 2px;">RDR324</td> <td style="padding: 2px;">RDR323</td> <td style="padding: 2px;">RDR322</td> <td style="padding: 2px;">RDR321</td> <td style="padding: 2px;">RDR320</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;">シリアル受信データを格納</p>		RDR327	RDR326	RDR325	RDR324	RDR323	RDR322	RDR321	RDR320
RDR327	RDR326	RDR325	RDR324	RDR323	RDR322	RDR321	RDR320		

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-left: auto; margin-right: auto; border-collapse: collapse;"> <tr> <td style="padding: 2px;">TMA7</td> <td style="padding: 2px;">TMA6</td> <td style="padding: 2px;">TMA5</td> <td style="padding: 2px;">—</td> <td style="padding: 2px;">TMA3</td> <td style="padding: 2px;">TMA2</td> <td style="padding: 2px;">TMA1</td> <td style="padding: 2px;">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> <div style="margin-top: 10px;"> <p>内部クロックセレクト</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>TMA3</th> <th>TMA2</th> <th>TMA1</th> <th>TMA0</th> <th>プリスケラ分周比又はオーバフロー周期</th> <th>機 能</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>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>0</td><td>0</td><td>0</td><td>1</td><td>PSS /4096</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td><td>PSS /2048</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>PSS /512</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td><td>PSS /256</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td><td>PSS /128</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td><td>PSS /32</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td><td>PSS /8</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>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 (32,768kHz使用時)</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td><td>PSW 0.5s</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td><td>PSW 0.25s</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td><td>PSW 0.03125s</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td><td rowspan="4" style="text-align: center;">PSW、TCAリセット</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td></tr> </tbody> </table> </div> <div style="margin-top: 10px;"> <p>クロック出力セレクト*</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>0</td><td>0</td><td>0</td><td>/32</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>/16</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>/8</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>/4</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>w/32</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>w/16</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>w/8</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>w/4</td></tr> </table> </div> <p>【注】* CWOSRのCWOSビット=0の場合の値です。CWOSビット=1のときは、TMA7~5ビットの値に関係なく wが出力されます。</p>		TMA7	TMA6	TMA5	—	TMA3	TMA2	TMA1	TMA0	TMA3	TMA2	TMA1	TMA0	プリスケラ分周比又はオーバフロー周期	機 能	0	0	0	0	PSS /8192	インターバル	0	0	0	1	PSS /4096	0	0	1	0	PSS /2048	0	0	1	1	PSS /512	0	1	0	0	PSS /256	0	1	0	1	PSS /128	0	1	1	0	PSS /32	0	1	1	1	PSS /8	1	0	0	0	PSW 1s	時計用タイムベース (32,768kHz使用時)	1	0	0	1	PSW 0.5s	1	0	1	0	PSW 0.25s	1	0	1	1	PSW 0.03125s	1	1	0	0	PSW、TCAリセット	1	1	0	1	1	1	1	0	1	1	1	1	0	0	0	/32	0	0	1	/16	0	1	0	/8	0	1	1	/4	1	0	0	w/32	1	0	1	w/16	1	1	0	w/8	1	1	1	w/4
TMA7	TMA6	TMA5	—	TMA3	TMA2	TMA1	TMA0																																																																																																																							
TMA3	TMA2	TMA1	TMA0	プリスケラ分周比又はオーバフロー周期	機 能																																																																																																																									
0	0	0	0	PSS /8192	インターバル																																																																																																																									
0	0	0	1	PSS /4096																																																																																																																										
0	0	1	0	PSS /2048																																																																																																																										
0	0	1	1	PSS /512																																																																																																																										
0	1	0	0	PSS /256																																																																																																																										
0	1	0	1	PSS /128																																																																																																																										
0	1	1	0	PSS /32																																																																																																																										
0	1	1	1	PSS /8																																																																																																																										
1	0	0	0	PSW 1s	時計用タイムベース (32,768kHz使用時)																																																																																																																									
1	0	0	1	PSW 0.5s																																																																																																																										
1	0	1	0	PSW 0.25s																																																																																																																										
1	0	1	1	PSW 0.03125s																																																																																																																										
1	1	0	0	PSW、TCAリセット																																																																																																																										
1	1	0	1																																																																																																																											
1	1	1	0																																																																																																																											
1	1	1	1																																																																																																																											
0	0	0	/32																																																																																																																											
0	0	1	/16																																																																																																																											
0	1	0	/8																																																																																																																											
0	1	1	/4																																																																																																																											
1	0	0	w/32																																																																																																																											
1	0	1	w/16																																																																																																																											
1	1	0	w/8																																																																																																																											
1	1	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; text-align: center;"> <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> <div style="text-align: center; margin-top: 10px;"> <p>↓</p> <p>カウント値</p> </div>				TCA7	TCA6	TCA5	TCA4	TCA3	TCA2	TCA1	TCA0
TCA7	TCA6	TCA5	TCA4	TCA3	TCA2	TCA1	TCA0				



TCW タイマカウンタW	H'B3		ウォッチドッグ タイマ								
<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;">TCW7</td> <td style="padding: 2px 5px;">TCW6</td> <td style="padding: 2px 5px;">TCW5</td> <td style="padding: 2px 5px;">TCW4</td> <td style="padding: 2px 5px;">TCW3</td> <td style="padding: 2px 5px;">TCW2</td> <td style="padding: 2px 5px;">TCW1</td> <td style="padding: 2px 5px;">TCW0</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> <p style="text-align: center; margin-left: 100px;"> カウント値</p>				TCW7	TCW6	TCW5	TCW4	TCW3	TCW2	TCW1	TCW0
TCW7	TCW6	TCW5	TCW4	TCW3	TCW2	TCW1	TCW0				

TMC タイマモードレジスタC		H'B4						タイマC																																																	
ビット : 7 6 5 4 3 2 1 0																																																									
TMC7		TMC6		TMC5		—		—		TMC2		TMC1		TMC0																																											
初期値 :		0		0		0		1		1		0		0		0																																									
R/W :		R/W		R/W		R/W		—		—		R/W		R/W		R/W																																									
									クロックセレクト <table border="1"> <tr><td>0</td><td>0</td><td>0</td><td>内部クロック</td><td>/8192</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>内部クロック</td><td>/2048</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>内部クロック</td><td>/512</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>内部クロック</td><td>/64</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>内部クロック</td><td>/16</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>内部クロック</td><td>/4</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>内部クロック</td><td>w/4</td></tr> <tr><td>1</td><td>1</td><td>1</td><td colspan="2">外部イベント (TMIC) : 立上がり / 立下がりエッジでカウント</td></tr> </table>									0	0	0	内部クロック	/8192	0	0	1	内部クロック	/2048	0	1	0	内部クロック	/512	0	1	1	内部クロック	/64	1	0	0	内部クロック	/16	1	0	1	内部クロック	/4	1	1	0	内部クロック	w/4	1	1	1	外部イベント (TMIC) : 立上がり / 立下がりエッジでカウント	
0	0	0	内部クロック	/8192																																																					
0	0	1	内部クロック	/2048																																																					
0	1	0	内部クロック	/512																																																					
0	1	1	内部クロック	/64																																																					
1	0	0	内部クロック	/16																																																					
1	0	1	内部クロック	/4																																																					
1	1	0	内部クロック	w/4																																																					
1	1	1	外部イベント (TMIC) : 立上がり / 立下がりエッジでカウント																																																						
									カウンタアップ/ダウン制御 <table border="1"> <tr><td>0</td><td>0</td><td>TCCは、アップカウンタ</td></tr> <tr><td>0</td><td>1</td><td>TCCは、ダウンカウンタ</td></tr> <tr><td>1</td><td>*</td><td>TCCのアップ/ダウンは、UD端子入力によるハードウェア制御 UD端子入力が"High"レベルならばダウンカウンタ、"Low"レベルならばアップカウンタ</td></tr> </table> <p style="text-align: right;">* Don't care</p>									0	0	TCCは、アップカウンタ	0	1	TCCは、ダウンカウンタ	1	*	TCCのアップ/ダウンは、UD端子入力によるハードウェア制御 UD端子入力が"High"レベルならばダウンカウンタ、"Low"レベルならばアップカウンタ																															
0	0	TCCは、アップカウンタ																																																							
0	1	TCCは、ダウンカウンタ																																																							
1	*	TCCのアップ/ダウンは、UD端子入力によるハードウェア制御 UD端子入力が"High"レベルならばダウンカウンタ、"Low"レベルならばアップカウンタ																																																							
									オートリロード機能選択 <table border="1"> <tr><td>0</td><td>インターバル機能を選択</td></tr> <tr><td>1</td><td>オートリロード機能を選択</td></tr> </table>									0	インターバル機能を選択	1	オートリロード機能を選択																																				
0	インターバル機能を選択																																																								
1	オートリロード機能を選択																																																								

TCC タイマカウンタC		H'B5						タイマC									
ビット : 7 6 5 4 3 2 1 0																	
TCC7		TCC6		TCC5		TCC4		TCC3		TCC2		TCC1		TCC0			
初期値 :		0		0		0		0		0		0		0			
R/W :		R		R		R		R		R		R		R			
									カウント値								
									【注】 TCCは、TLCと同一アドレスに割り付けられています。リード時には、TCCの値が読み出されます。								

TLC タイマロードレジスタC	H'B5		タイマC
-----------------	------	--	------

ビット： 7 6 5 4 3 2 1 0

TLC7	TLC6	TLC5	TLC4	TLC3	TLC2	TLC1	TLC0
------	------	------	------	------	------	------	------

初期値： 0 0 0 0 0 0 0 0

R/W : W W W W W W W W

リロード値設定

【注】 TLCは、TCCと同一アドレスに割り付けられています。ライト時には、TLCに値が書込まれます。

TCRF タイマコントロールレジスタF	H'B6		タイマF
---------------------	------	--	------

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

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
1	0	0	内部クロック /32
1	0	1	内部クロック /16
1	1	0	内部クロック /4
1	1	1	内部クロック $\sqrt{w}/4$

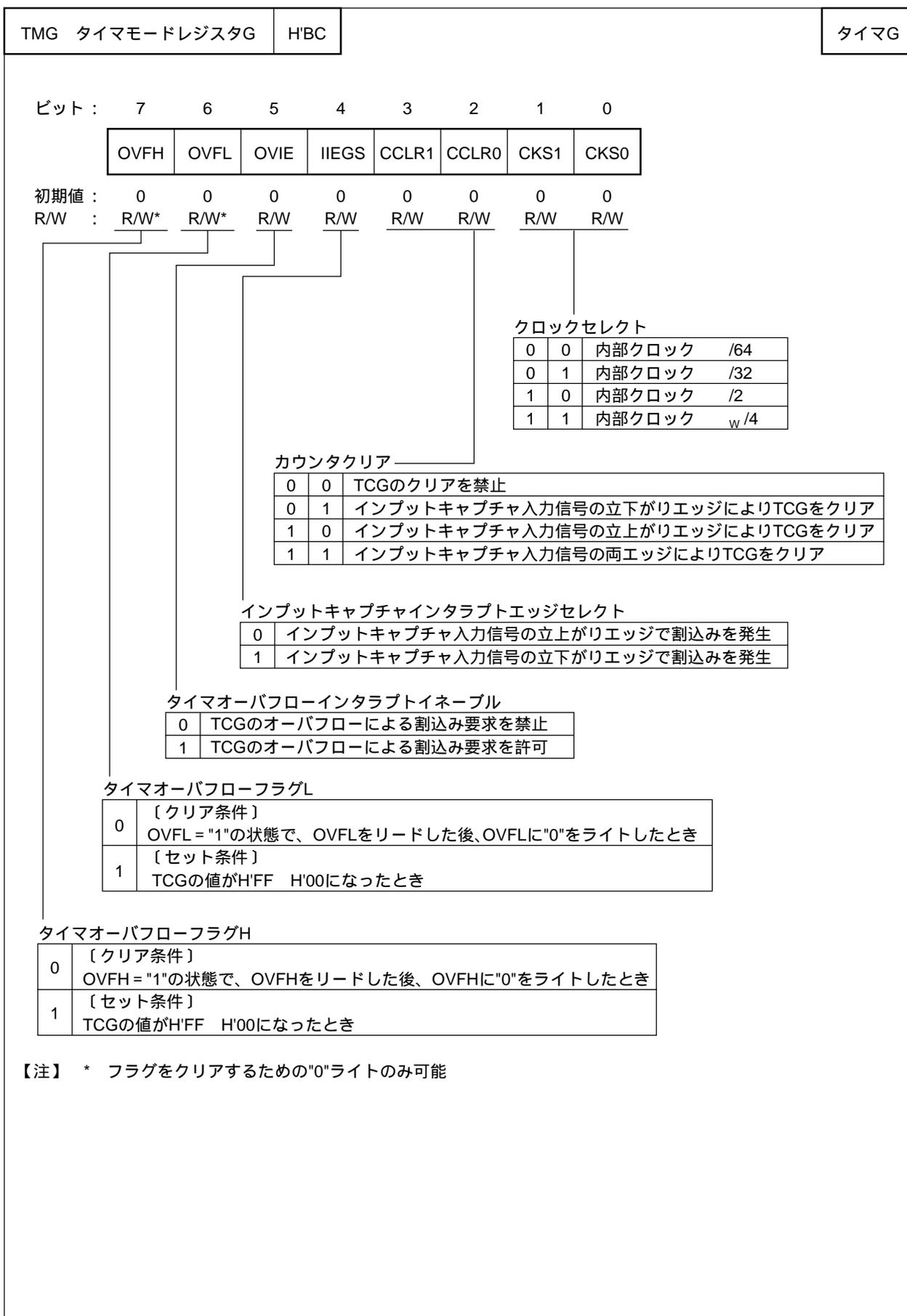
0	"Low"レベルに設定
1	"High"レベルに設定

0	*	*	16ビットモードとなり、TCFLのオーバーフロー信号でカウント
1	0	0	内部クロック /32
1	0	1	内部クロック /16
1	1	0	内部クロック /4
1	1	1	内部クロック $\sqrt{w}/4$

* Don't care

0	"Low"レベルに設定
1	"High"レベルに設定

TCSRFB タイマコントロールステータスレジスタF	H'B7	タイムF																																																								
<p>ビット： 7 6 5 4 3 2 1 0</p> <table border="1" style="margin: auto; border-collapse: collapse;"> <tr> <td style="width: 12.5%;">OVFH</td> <td style="width: 12.5%;">CMFH</td> <td style="width: 12.5%;">OVIEH</td> <td style="width: 12.5%;">CCLRH</td> <td style="width: 12.5%;">OVFL</td> <td style="width: 12.5%;">CMFL</td> <td style="width: 12.5%;">OVIEL</td> <td style="width: 12.5%;">CCLRL</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-top: 20px;"> <p style="text-align: center;">カウンタクリアL</p> <table border="1" style="margin: auto; border-collapse: collapse;"> <tr> <td style="width: 5%;">0</td> <td>コンペアマッチによるTCFLのクリアを禁止</td> </tr> <tr> <td>1</td> <td>コンペアマッチによるTCFLのクリアを許可</td> </tr> </table> <p style="text-align: center;">タイマオーバーフローインタラプトイネーブルL</p> <table border="1" style="margin: auto; border-collapse: collapse;"> <tr> <td style="width: 5%;">0</td> <td>TCFLのオーバーフローによる割込み要求を禁止</td> </tr> <tr> <td>1</td> <td>TCFLのオーバーフローによる割込み要求を許可</td> </tr> </table> <p style="text-align: center;">コンペアマッチフラグL</p> <table border="1" style="margin: auto; border-collapse: collapse;"> <tr> <td style="width: 5%;"></td> <td style="text-align: center;">〔クリア条件〕</td> </tr> <tr> <td>0</td> <td>CMFL = "1"の状態、CMFLをリードした後、CMFLに"0"をライトしたとき</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">〔セット条件〕</td> </tr> <tr> <td></td> <td>TCFLの値とOCRFLの値が、コンペアマッチしたとき</td> </tr> </table> <p style="text-align: center;">タイマオーバーフローL</p> <table border="1" style="margin: auto; border-collapse: collapse;"> <tr> <td style="width: 5%;"></td> <td style="text-align: center;">〔クリア条件〕</td> </tr> <tr> <td>0</td> <td>OVFL = "1"の状態、OVFLをリードした後、OVFLに"0"をライトしたとき</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">〔セット条件〕</td> </tr> <tr> <td></td> <td>TCFLの値が、H'FF H'00になったとき</td> </tr> </table> <p style="text-align: center;">カウンタクリアH</p> <table border="1" style="margin: auto; border-collapse: collapse;"> <tr> <td style="width: 5%;">0</td> <td>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</td> </tr> <tr> <td>1</td> <td>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</td> </tr> </table> <p style="text-align: center;">タイマオーバーフローインタラプトイネーブルH</p> <table border="1" style="margin: auto; border-collapse: collapse;"> <tr> <td style="width: 5%;">0</td> <td>TCFHのオーバーフローによる割込み要求を禁止</td> </tr> <tr> <td>1</td> <td>TCFHのオーバーフローによる割込み要求を許可</td> </tr> </table> <p style="text-align: center;">コンペアマッチフラグH</p> <table border="1" style="margin: auto; border-collapse: collapse;"> <tr> <td style="width: 5%;"></td> <td style="text-align: center;">〔クリア条件〕</td> </tr> <tr> <td>0</td> <td>CMFH = "1"の状態、CMFHをリードした後、CMFHに"0"をライトしたとき</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">〔セット条件〕</td> </tr> <tr> <td></td> <td>TCFHの値とOCRFHの値が、コンペアマッチしたとき</td> </tr> </table> <p style="text-align: center;">タイマオーバーフローフラグH</p> <table border="1" style="margin: auto; border-collapse: collapse;"> <tr> <td style="width: 5%;"></td> <td style="text-align: center;">〔クリア条件〕</td> </tr> <tr> <td>0</td> <td>OVFH = "1"の状態、OVFHをリードした後、OVFHに"0"をライトしたとき</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">〔セット条件〕</td> </tr> <tr> <td></td> <td>TCFHの値が、H'FF H'00になったとき</td> </tr> </table> </div>			OVFH	CMFH	OVIEH	CCLRH	OVFL	CMFL	OVIEL	CCLRL	0	コンペアマッチによるTCFLのクリアを禁止	1	コンペアマッチによるTCFLのクリアを許可	0	TCFLのオーバーフローによる割込み要求を禁止	1	TCFLのオーバーフローによる割込み要求を許可		〔クリア条件〕	0	CMFL = "1"の状態、CMFLをリードした後、CMFLに"0"をライトしたとき	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		〔クリア条件〕	0	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〕		TCFLの値が、H'FF H'00になったとき	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	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	0	TCFHのオーバーフローによる割込み要求を禁止	1	TCFHのオーバーフローによる割込み要求を許可		〔クリア条件〕	0	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〕		TCFHの値とOCRFHの値が、コンペアマッチしたとき		〔クリア条件〕	0	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〕		TCFHの値が、H'FF H'00になったとき
OVFH	CMFH	OVIEH	CCLRH	OVFL	CMFL	OVIEL	CCLRL																																																			
0	コンペアマッチによるTCFLのクリアを禁止																																																									
1	コンペアマッチによるTCFLのクリアを許可																																																									
0	TCFLのオーバーフローによる割込み要求を禁止																																																									
1	TCFLのオーバーフローによる割込み要求を許可																																																									
	〔クリア条件〕																																																									
0	CMFL = "1"の状態、CMFLをリードした後、CMFLに"0"をライトしたとき																																																									
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																																																									
	〔クリア条件〕																																																									
0	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〕																																																									
	TCFLの値が、H'FF H'00になったとき																																																									
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																																																									
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																																																									
0	TCFHのオーバーフローによる割込み要求を禁止																																																									
1	TCFHのオーバーフローによる割込み要求を許可																																																									
	〔クリア条件〕																																																									
0	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〕																																																									
	TCFHの値とOCRFHの値が、コンペアマッチしたとき																																																									
	〔クリア条件〕																																																									
0	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〕																																																									
	TCFHの値が、H'FF H'00になったとき																																																									
<p>【注】 * フラグをクリアするための"0"ライトのみ可能</p>																																																										



ICRGF イン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; text-align: center;"> <tr> <td style="padding: 2px;">ICRGF7</td> <td style="padding: 2px;">ICRGF6</td> <td style="padding: 2px;">ICRGF5</td> <td style="padding: 2px;">ICRGF4</td> <td style="padding: 2px;">ICRGF3</td> <td style="padding: 2px;">ICRGF2</td> <td style="padding: 2px;">ICRGF1</td> <td style="padding: 2px;">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> <p style="text-align: center; margin-top: 10px;">↑</p> <p style="text-align: center;">インットキャプチャ信号の立下がりエッジで、TCGの値を格納</p>			ICRGF7	ICRGF6	ICRGF5	ICRGF4	ICRGF3	ICRGF2	ICRGF1	ICRGF0
ICRGF7	ICRGF6	ICRGF5	ICRGF4	ICRGF3	ICRGF2	ICRGF1	ICRGF0			

ICRGR イン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; text-align: center;"> <tr> <td style="padding: 2px;">ICRGR7</td> <td style="padding: 2px;">ICRGR6</td> <td style="padding: 2px;">ICRGR5</td> <td style="padding: 2px;">ICRGR4</td> <td style="padding: 2px;">ICRGR3</td> <td style="padding: 2px;">ICRGR2</td> <td style="padding: 2px;">ICRGR1</td> <td style="padding: 2px;">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> <p style="text-align: center; margin-top: 10px;">↑</p> <p style="text-align: center;">インットキャプチャ信号の上上がりエッジで、TCGの値を格納</p>			ICRGR7	ICRGR6	ICRGR5	ICRGR4	ICRGR3	ICRGR2	ICRGR1	ICRGR0
ICRGR7	ICRGR6	ICRGR5	ICRGR4	ICRGR3	ICRGR2	ICRGR1	ICRGR0			

LPCR LCDポートコントロールレジスタ	H'CO	LCD コントローラ/ドライバ
-----------------------	------	-----------------

ビット： 7 6 5 4 3 2 1 0

DTS1	DTS0	CMX	SGX	SGS3	SGS2	SGS1	SGS0
------	------	-----	-----	------	------	------	------

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

セグメントドライバ選択

ビット4	ビット3	ビット2	ビット1	ビット0	SEG ₄₀ ~ SEG ₁ 端子の機能					説明の補足
SGX	SGS3	SGS2	SGS1	SGS0	SEG ₄₀ ~ SEG ₃₃	SEG ₃₂ ~ SEG ₂₅	SEG ₂₄ ~ SEG ₁₇	SEG ₁₆ ~ SEG ₉	SEG ₈ ~ SEG ₁	
0	0	0	0	0	ポート	ポート	ポート	ポート	ポート	(初期値)
	0	0	0	1	SEG	ポート	ポート	ポート	ポート	
	0	0	1	*	SEG	SEG	ポート	ポート	ポート	
	0	1	0	*	SEG	SEG	SEG	ポート	ポート	
	0	1	1	*	SEG	SEG	SEG	SEG	ポート	
	1	*	*	*	SEG	SEG	SEG	SEG	SEG	
1	0	0	0	0	ポート*1	ポート	ポート	ポート	ポート	使用禁止
	0	0	0	1						
	0	0	1	*						
	0	1	*	*						
	1	*	*	*						

* : Don't care

【注】 *1 SEG₄₀ ~ SEG₃₇は外部拡張端子

ビット4	説明
SGX	
0	SEG ₄₀ ~ SEG ₃₇ 端子* (初期値)
1	CL ₁ 、CL ₂ 、DO、M端子

【注】 * SGS3 ~ SGS0が"0000"のときはポートとして機能します。

DUTY選択、コモン機能選択

ビット7	ビット6	ビット5	デューティ比	コモンドライバ	補足説明
DTS1	DTS0	CMX			
0	0	0	スタティック	COM ₁	COM ₄ ~ COM ₂ は使用禁止
		1		COM ₄ ~ COM ₁	COM ₄ ~ COM ₂ はCOM ₁ と同じ波形が出力
0	1	0	1/2デューティ	COM ₂ 、COM ₁	COM ₄ 、COM ₃ は使用禁止
		1		COM ₄ ~ COM ₁	COM ₄ はCOM ₃ 、COM ₂ はCOM ₁ と同じ波形が出力
1	0	0	1/3デューティ	COM ₃ ~ COM ₁	COM ₄ は使用禁止
		1		COM ₄ ~ COM ₁	COM ₄ は使用禁止
1	1	0	1/4デューティ	COM ₄ ~ COM ₁	—
		1			

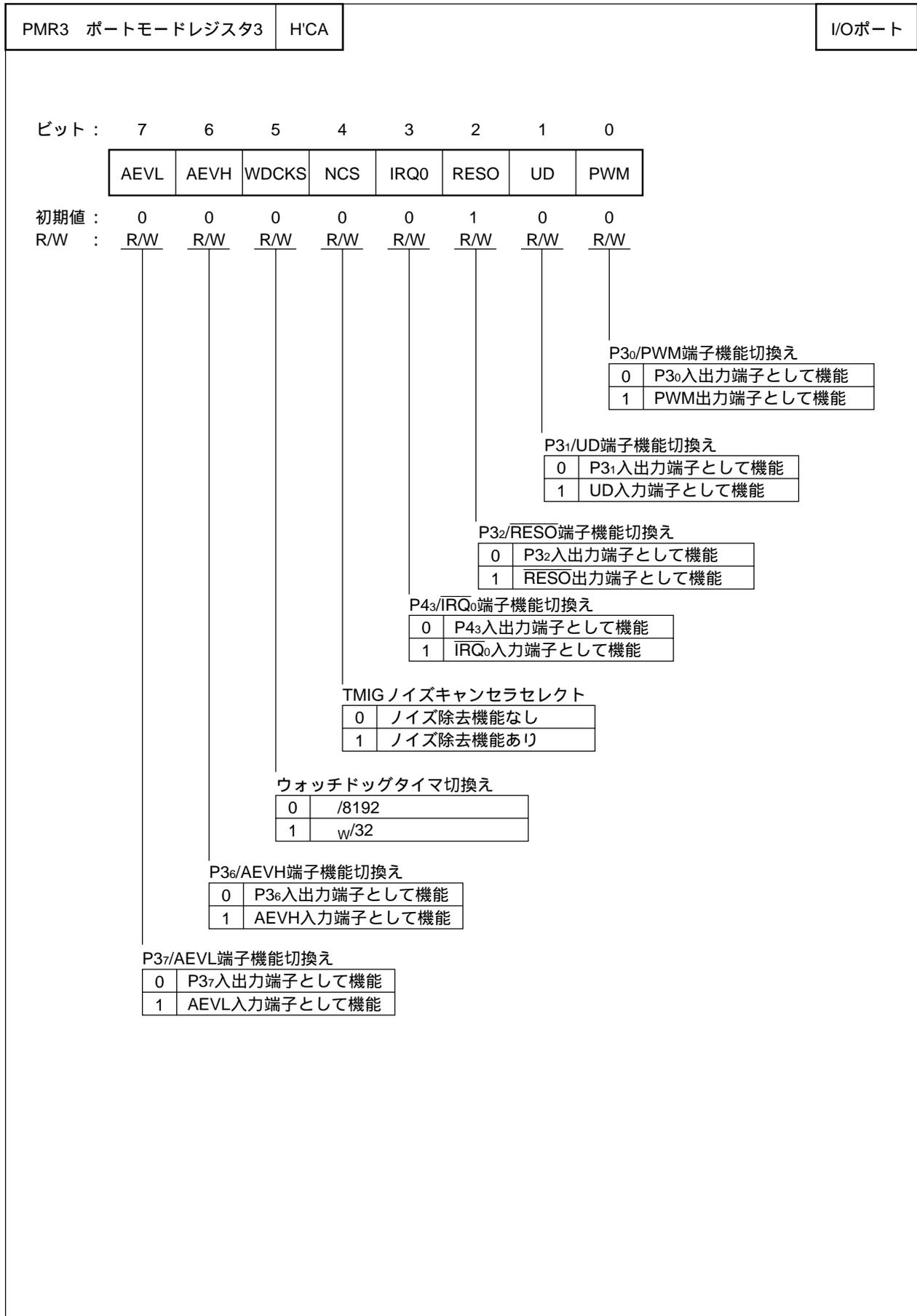
LCR LCDコントロールレジスタ	H'C1	LCD コントローラ/ドライバ																																																																																					
<p>ビット： 7 6 5 4 3 2 1 0</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">—</td> <td style="width: 20px; text-align: center;">PSW</td> <td style="width: 20px; text-align: center;">ACT</td> <td style="width: 20px; text-align: center;">DISP</td> <td style="width: 20px; text-align: center;">CKS3</td> <td style="width: 20px; text-align: center;">CKS2</td> <td style="width: 20px; text-align: center;">CKS1</td> <td style="width: 20px; text-align: center;">CKS0</td> </tr> </table> <p>初期値： 1 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-left: 100px; margin-top: 20px;"> <p>フレーム周波数選択</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">ビット3</th> <th style="width: 15%;">ビット2</th> <th style="width: 15%;">ビット1</th> <th style="width: 15%;">ビット0</th> <th style="width: 20%;">使用クロック</th> </tr> <tr> <th>CKS3</th> <th>CKS2</th> <th>CKS1</th> <th>CKS0</th> <th></th> </tr> </thead> <tbody> <tr><td>0</td><td>*</td><td>0</td><td>0</td><td>w</td></tr> <tr><td>0</td><td>*</td><td>0</td><td>1</td><td>w/2</td></tr> <tr><td>0</td><td>*</td><td>1</td><td>*</td><td>w/4</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td><td>/2</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td><td>/4</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td><td>/8</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td><td>/16</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td><td>/32</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td><td>/64</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>0</td><td>/128</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>/256</td></tr> </tbody> </table> </div> <div style="margin-left: 100px; margin-top: 20px;"> <p>表示データ制御</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr><td style="width: 20px; text-align: center;">0</td><td>ブランクデータを表示</td></tr> <tr><td style="text-align: center;">1</td><td>LCD RAMデータを表示</td></tr> </table> </div> <div style="margin-left: 100px; margin-top: 20px;"> <p>表示機能開始</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr><td style="width: 20px; text-align: center;">0</td><td>LCDコントローラ/ドライバ動作停止</td></tr> <tr><td style="text-align: center;">1</td><td>LCDコントローラ/ドライバ動作</td></tr> </table> </div> <div style="margin-left: 100px; margin-top: 20px;"> <p>LCD駆動電源ON/OFF制御</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr><td style="width: 20px; text-align: center;">0</td><td>LCD駆動電源OFF</td></tr> <tr><td style="text-align: center;">1</td><td>LCD駆動電源ON</td></tr> </table> </div>			—	PSW	ACT	DISP	CKS3	CKS2	CKS1	CKS0	ビット3	ビット2	ビット1	ビット0	使用クロック	CKS3	CKS2	CKS1	CKS0		0	*	0	0	w	0	*	0	1	w/2	0	*	1	*	w/4	1	0	0	0	/2	1	0	0	1	/4	1	0	1	0	/8	1	0	1	1	/16	1	1	0	0	/32	1	1	0	1	/64	1	1	1	0	/128	1	1	1	1	/256	0	ブランクデータを表示	1	LCD RAMデータを表示	0	LCDコントローラ/ドライバ動作停止	1	LCDコントローラ/ドライバ動作	0	LCD駆動電源OFF	1	LCD駆動電源ON
—	PSW	ACT	DISP	CKS3	CKS2	CKS1	CKS0																																																																																
ビット3	ビット2	ビット1	ビット0	使用クロック																																																																																			
CKS3	CKS2	CKS1	CKS0																																																																																				
0	*	0	0	w																																																																																			
0	*	0	1	w/2																																																																																			
0	*	1	*	w/4																																																																																			
1	0	0	0	/2																																																																																			
1	0	0	1	/4																																																																																			
1	0	1	0	/8																																																																																			
1	0	1	1	/16																																																																																			
1	1	0	0	/32																																																																																			
1	1	0	1	/64																																																																																			
1	1	1	0	/128																																																																																			
1	1	1	1	/256																																																																																			
0	ブランクデータを表示																																																																																						
1	LCD RAMデータを表示																																																																																						
0	LCDコントローラ/ドライバ動作停止																																																																																						
1	LCDコントローラ/ドライバ動作																																																																																						
0	LCD駆動電源OFF																																																																																						
1	LCD駆動電源ON																																																																																						

LCR2 LCDコントロールレジスタ2	H'C2		LCD																																																																												
<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;">LCDAB</td> <td style="padding: 2px;">—</td> <td style="padding: 2px;">—</td> <td style="padding: 2px;">SUPS</td> <td style="padding: 2px;">CDS3</td> <td style="padding: 2px;">CDS2</td> <td style="padding: 2px;">CDS1</td> <td style="padding: 2px;">CDS0</td> </tr> </table> <p>初期値： 0 1 1 0 0 0 0 0</p> <p>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;">充放電パルスのデューティ比選択</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <thead> <tr> <th style="padding: 2px;">ビット3</th> <th style="padding: 2px;">ビット2</th> <th style="padding: 2px;">ビット1</th> <th style="padding: 2px;">ビット0</th> <th style="padding: 2px;">デューティ比</th> </tr> <tr> <th style="padding: 2px;">CDS3</th> <th style="padding: 2px;">CDS2</th> <th style="padding: 2px;">CDS1</th> <th style="padding: 2px;">CDS0</th> <th style="padding: 2px;"></th> </tr> </thead> <tbody> <tr><td style="padding: 2px;">0</td><td style="padding: 2px;">0</td><td style="padding: 2px;">0</td><td style="padding: 2px;">0</td><td style="padding: 2px;">1</td></tr> <tr><td style="padding: 2px;">0</td><td style="padding: 2px;">0</td><td style="padding: 2px;">0</td><td style="padding: 2px;">1</td><td style="padding: 2px;">1/8</td></tr> <tr><td style="padding: 2px;">0</td><td style="padding: 2px;">0</td><td style="padding: 2px;">1</td><td style="padding: 2px;">0</td><td style="padding: 2px;">2/8</td></tr> <tr><td style="padding: 2px;">0</td><td style="padding: 2px;">0</td><td style="padding: 2px;">1</td><td style="padding: 2px;">1</td><td style="padding: 2px;">3/8</td></tr> <tr><td style="padding: 2px;">0</td><td style="padding: 2px;">1</td><td style="padding: 2px;">0</td><td style="padding: 2px;">0</td><td style="padding: 2px;">4/8</td></tr> <tr><td style="padding: 2px;">0</td><td style="padding: 2px;">1</td><td style="padding: 2px;">0</td><td style="padding: 2px;">1</td><td style="padding: 2px;">5/8</td></tr> <tr><td style="padding: 2px;">0</td><td style="padding: 2px;">1</td><td style="padding: 2px;">1</td><td style="padding: 2px;">0</td><td style="padding: 2px;">6/8</td></tr> <tr><td style="padding: 2px;">0</td><td style="padding: 2px;">1</td><td style="padding: 2px;">1</td><td style="padding: 2px;">1</td><td style="padding: 2px;">0</td></tr> <tr><td style="padding: 2px;">1</td><td style="padding: 2px;">0</td><td style="padding: 2px;">*</td><td style="padding: 2px;">*</td><td style="padding: 2px;">1/16</td></tr> <tr><td style="padding: 2px;">1</td><td style="padding: 2px;">1</td><td style="padding: 2px;">*</td><td style="padding: 2px;">*</td><td style="padding: 2px;">1/32</td></tr> </tbody> </table> <p style="text-align: right; margin-top: 5px;">* Don't care</p> </div> <div style="margin-top: 20px;"> <p style="text-align: center;">駆動電源の選択、昇圧定電圧（5V）電源の制御</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;">駆動電源はV_{CC}、昇圧定電圧（5V）電源動作停止</td> </tr> <tr> <td style="padding: 2px;">1</td> <td style="padding: 2px;">駆動電源は5V、昇圧定電圧（5V）電源動作</td> </tr> </table> </div> <div style="margin-top: 20px;"> <p style="text-align: center;">A波形/B波形切換え制御</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;">A波形で駆動</td> </tr> <tr> <td style="padding: 2px;">1</td> <td style="padding: 2px;">B波形で駆動</td> </tr> </table> </div>				LCDAB	—	—	SUPS	CDS3	CDS2	CDS1	CDS0	ビット3	ビット2	ビット1	ビット0	デューティ比	CDS3	CDS2	CDS1	CDS0		0	0	0	0	1	0	0	0	1	1/8	0	0	1	0	2/8	0	0	1	1	3/8	0	1	0	0	4/8	0	1	0	1	5/8	0	1	1	0	6/8	0	1	1	1	0	1	0	*	*	1/16	1	1	*	*	1/32	0	駆動電源はV _{CC} 、昇圧定電圧（5V）電源動作停止	1	駆動電源は5V、昇圧定電圧（5V）電源動作	0	A波形で駆動	1	B波形で駆動
LCDAB	—	—	SUPS	CDS3	CDS2	CDS1	CDS0																																																																								
ビット3	ビット2	ビット1	ビット0	デューティ比																																																																											
CDS3	CDS2	CDS1	CDS0																																																																												
0	0	0	0	1																																																																											
0	0	0	1	1/8																																																																											
0	0	1	0	2/8																																																																											
0	0	1	1	3/8																																																																											
0	1	0	0	4/8																																																																											
0	1	0	1	5/8																																																																											
0	1	1	0	6/8																																																																											
0	1	1	1	0																																																																											
1	0	*	*	1/16																																																																											
1	1	*	*	1/32																																																																											
0	駆動電源はV _{CC} 、昇圧定電圧（5V）電源動作停止																																																																														
1	駆動電源は5V、昇圧定電圧（5V）電源動作																																																																														
0	A波形で駆動																																																																														
1	B波形で駆動																																																																														

AMR A/Dモードレジスタ	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;"> <tr> <td style="width: 20px; text-align: center;">CKS</td> <td style="width: 20px; text-align: center;">TRGE</td> <td style="width: 20px; text-align: center;">—</td> <td style="width: 20px; text-align: center;">—</td> <td style="width: 20px; text-align: center;">CH3</td> <td style="width: 20px; text-align: center;">CH2</td> <td style="width: 20px; text-align: center;">CH1</td> <td style="width: 20px; text-align: center;">CH0</td> </tr> </table> <p>初期値： 0 0 1 1 0 0 0 0</p> <p>R/W： R/W R/W — — R/W R/W R/W R/W</p> <div style="margin-left: 100px; margin-top: 10px;"> <p>チャンネルセレクト</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">ビット3</th> <th style="width: 15%;">ビット2</th> <th style="width: 15%;">ビット1</th> <th style="width: 15%;">ビット0</th> <th style="width: 40%;">アナログ入力チャンネル</th> </tr> <tr> <th>CH3</th> <th>CH2</th> <th>CH1</th> <th>CH0</th> <th></th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>*</td><td>*</td><td>非選択</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td><td>AN₀</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td><td>AN₁</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td><td>AN₂</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td><td>AN₃</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td><td>AN₄</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td><td>AN₅</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td><td>AN₆</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td><td>AN₇</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td><td>AN₈</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td><td>AN₉</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>0</td><td>AN₁₀</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>AN₁₁</td></tr> </tbody> </table> <p style="text-align: right; margin-right: 20px;">* Don't care</p> </div> <div style="margin-left: 100px; margin-top: 10px;"> <p>外部トリガセレクト</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">0</td> <td>外部トリガによるA/D変換の開始を禁止</td> </tr> <tr> <td style="text-align: center;">1</td> <td>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</td> </tr> </table> </div> <div style="margin-left: 100px; margin-top: 10px;"> <p>クロックセレクト</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <thead> <tr> <th rowspan="2" style="width: 15%;">ビット7</th> <th rowspan="2" style="width: 15%;">変換周期</th> <th colspan="2" style="width: 70%;">変換時間</th> </tr> <tr> <th style="width: 35%;">= 1MHz</th> <th style="width: 35%;">= 2MHz</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">62/</td> <td style="text-align: center;">62μs</td> <td style="text-align: center;">31μs</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">31/</td> <td style="text-align: center;">31μs</td> <td style="text-align: center;">15.5μs *</td> </tr> </tbody> </table> </div>			CKS	TRGE	—	—	CH3	CH2	CH1	CH0	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	0	0	AN ₀	0	1	0	1	AN ₁	0	1	1	0	AN ₂	0	1	1	1	AN ₃	1	0	0	0	AN ₄	1	0	0	1	AN ₅	1	0	1	0	AN ₆	1	0	1	1	AN ₇	1	1	0	0	AN ₈	1	1	0	1	AN ₉	1	1	1	0	AN ₁₀	1	1	1	1	AN ₁₁	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	ビット7	変換周期	変換時間		= 1MHz	= 2MHz	0	62/	62μs	31μs	1	31/	31μs	15.5μs *
CKS	TRGE	—	—	CH3	CH2	CH1	CH0																																																																																																
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	0	0	AN ₀																																																																																																			
0	1	0	1	AN ₁																																																																																																			
0	1	1	0	AN ₂																																																																																																			
0	1	1	1	AN ₃																																																																																																			
1	0	0	0	AN ₄																																																																																																			
1	0	0	1	AN ₅																																																																																																			
1	0	1	0	AN ₆																																																																																																			
1	0	1	1	AN ₇																																																																																																			
1	1	0	0	AN ₈																																																																																																			
1	1	0	1	AN ₉																																																																																																			
1	1	1	0	AN ₁₀																																																																																																			
1	1	1	1	AN ₁₁																																																																																																			
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																																																																																																						
ビット7	変換周期	変換時間																																																																																																					
		= 1MHz	= 2MHz																																																																																																				
0	62/	62μs	31μs																																																																																																				
1	31/	31μs	15.5μs *																																																																																																				

ADDRH A/DリザルトレジスタH H'C4 ADDRLL A/DリザルトレジスタL H'C5	A/D変換器																
<p>ADDRH</p> <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;">ADR9</td> <td style="width: 20px;">ADR8</td> <td style="width: 20px;">ADR7</td> <td style="width: 20px;">ADR6</td> <td style="width: 20px;">ADR5</td> <td style="width: 20px;">ADR4</td> <td style="width: 20px;">ADR3</td> <td style="width: 20px;">ADR2</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;">A/D変換結果</p> <p>ADDRLL</p> <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;">ADR1</td> <td style="width: 20px;">ADR0</td> <td style="width: 20px;">—</td> </tr> </table> <p>初期値： 不定 不定 — — — — — —</p> <p>R/W： R R — — — — — —</p> <p style="text-align: center;">A/D変換結果</p>		ADR9	ADR8	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0	—	—	—	—	—	—
ADR9	ADR8	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2										
ADR1	ADR0	—	—	—	—	—	—										

ADSR A/Dスタートレジスタ H'C7	A/D変換器												
<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;">ADSF</td> <td style="width: 20px;">—</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;">A/Dスタートフラグ</p> <table border="1" style="margin-left: 40px; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">0</td> <td>〔リード時〕 A/D変換の終了 〔ライト時〕 A/D変換を強制終了</td> </tr> <tr> <td style="width: 20px; text-align: center;">1</td> <td>〔リード時〕 A/D変換中 〔ライト時〕 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												



PMR4 ポートモードレジスタ4	H'CB	I/Oポート								
ビット : 7 6 5 4 3 2 1 0										
<table border="1" style="width: 100%; text-align: center;"> <tr> <td>NMOD7</td><td>NMOD6</td><td>NMOD5</td><td>NMOD4</td><td>NMOD3</td><td>NMOD2</td><td>NMOD1</td><td>NMOD0</td> </tr> </table>			NMOD7	NMOD6	NMOD5	NMOD4	NMOD3	NMOD2	NMOD1	NMOD0
NMOD7	NMOD6	NMOD5	NMOD4	NMOD3	NMOD2	NMOD1	NMOD0			
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										
NMOSオープンドレイン出力セレクト										
<table border="1" style="width: 100%; text-align: center;"> <tr> <td>0</td><td>P2_nはCMOS出力</td> </tr> <tr> <td>1</td><td>P2_nはNMOSオープンドレイン出力</td> </tr> </table>			0	P2 _n はCMOS出力	1	P2 _n はNMOSオープンドレイン出力				
0	P2 _n はCMOS出力									
1	P2 _n はNMOSオープンドレイン出力									
(n = 7 ~ 0)										
【注】 PCR2を"1"に指定したとき（出力ポートに指定）										

PMR5 ポートモードレジスタ5	H'CC	I/Oポート								
ビット : 7 6 5 4 3 2 1 0										
<table border="1" style="width: 100%; text-align: center;"> <tr> <td>WKP7</td><td>WKP6</td><td>WKP5</td><td>WKP4</td><td>WKP3</td><td>WKP2</td><td>WKP1</td><td>WKP0</td> </tr> </table>			WKP7	WKP6	WKP5	WKP4	WKP3	WKP2	WKP1	WKP0
WKP7	WKP6	WKP5	WKP4	WKP3	WKP2	WKP1	WKP0			
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										
P5 _n /WKP _n /SEG _{n+1} 端子機能切換え										
<table border="1" style="width: 100%; text-align: center;"> <tr> <td>0</td><td>P5_n入出力端子として機能</td> </tr> <tr> <td>1</td><td>WKP_n入力端子として機能</td> </tr> </table>			0	P5 _n 入出力端子として機能	1	WKP _n 入力端子として機能				
0	P5 _n 入出力端子として機能									
1	WKP _n 入力端子として機能									
(n = 7 ~ 0)										

PWCR PWMコントロールレジスタ	H'D0		14ビットPWM																		
<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;">PWCR1</td> <td style="width: 20px; text-align: center;">PWCR0</td> </tr> </table> <p>初期値： 1 1 1 1 1 1 0 0</p> <p>R/W ： — — — — — — W W</p> <div style="margin-left: 100px; margin-top: 10px;"> <p>クロックセレクト</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr> <td rowspan="2" style="width: 20px; text-align: center;">0</td> <td style="width: 20px; text-align: center;">0</td> <td style="padding: 2px;">入力クロック /2 ($t^* = 2/$)。1変換周期16384/、最小変化幅1/</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="padding: 2px;">入力クロック /4 ($t^* = 4/$)。1変換周期32768/、最小変化幅2/</td> </tr> <tr> <td rowspan="2" style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="padding: 2px;">入力クロック /8 ($t^* = 8/$)。1変換周期65536/、最小変化幅4/</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="padding: 2px;">入力クロック /16 ($t^* = 16/$)。1変換周期131072/、最小変化幅8/</td> </tr> </table> <p>【注】*t : PWM入力クロックの周期</p> </div>				—	—	—	—	—	—	PWCR1	PWCR0	0	0	入力クロック /2 ($t^* = 2/$)。1変換周期16384/、最小変化幅1/	1	入力クロック /4 ($t^* = 4/$)。1変換周期32768/、最小変化幅2/	1	0	入力クロック /8 ($t^* = 8/$)。1変換周期65536/、最小変化幅4/	1	入力クロック /16 ($t^* = 16/$)。1変換周期131072/、最小変化幅8/
—	—	—	—	—	—	PWCR1	PWCR0														
0	0	入力クロック /2 ($t^* = 2/$)。1変換周期16384/、最小変化幅1/																			
	1	入力クロック /4 ($t^* = 4/$)。1変換周期32768/、最小変化幅2/																			
1	0	入力クロック /8 ($t^* = 8/$)。1変換周期65536/、最小変化幅4/																			
	1	入力クロック /16 ($t^* = 16/$)。1変換周期131072/、最小変化幅8/																			
PWDRU PWMデータレジスタU	H'D1		14ビットPWM																		
<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;">PWDRU5</td> <td style="width: 20px; text-align: center;">PWDRU4</td> <td style="width: 20px; text-align: center;">PWDRU3</td> <td style="width: 20px; text-align: center;">PWDRU2</td> <td style="width: 20px; text-align: center;">PWDRU1</td> <td style="width: 20px; text-align: center;">PWDRU0</td> </tr> </table> <p>初期値： 1 1 0 0 0 0 0 0</p> <p>R/W ： — — W W W W W W</p> <div style="margin-left: 100px; margin-top: 10px;"> <p>PWM波形生成用データ上位6ビット</p> </div>				—	—	PWDRU5	PWDRU4	PWDRU3	PWDRU2	PWDRU1	PWDRU0										
—	—	PWDRU5	PWDRU4	PWDRU3	PWDRU2	PWDRU1	PWDRU0														

PWDRL PWMデータレジスタ	H'D2	14ビットPWM								
ビット： 7 6 5 4 3 2 1 0 <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>PWDRL7</td> <td>PWDRL6</td> <td>PWDRL5</td> <td>PWDRL4</td> <td>PWDRL3</td> <td>PWDRL2</td> <td>PWDRL1</td> <td>PWDRL0</td> </tr> </table> 初期値： 0 0 0 0 0 0 0 0 R/W： W W W W W W W W PWM波形生成用データ下位8ビット			PWDRL7	PWDRL6	PWDRL5	PWDRL4	PWDRL3	PWDRL2	PWDRL1	PWDRL0
PWDRL7	PWDRL6	PWDRL5	PWDRL4	PWDRL3	PWDRL2	PWDRL1	PWDRL0			
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 <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>P17</td> <td>P16</td> <td>P15</td> <td>P14</td> <td>P13</td> <td>P12</td> <td>P11</td> <td>P10</td> </tr> </table> 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 ポート1の各端子のデータを格納			P17	P16	P15	P14	P13	P12	P11	P10
P17	P16	P15	P14	P13	P12	P11	P10			
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 <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>P27</td> <td>P26</td> <td>P25</td> <td>P24</td> <td>P23</td> <td>P22</td> <td>P21</td> <td>P20</td> </tr> </table> 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 ポート2の各端子のデータを格納			P27	P26	P25	P24	P23	P22	P21	P20
P27	P26	P25	P24	P23	P22	P21	P20			

PDR3 ポートデータレジスタ3	H'D6		I/Oポート								
<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 10px;">P37</td> <td style="padding: 2px 10px;">P36</td> <td style="padding: 2px 10px;">P35</td> <td style="padding: 2px 10px;">P34</td> <td style="padding: 2px 10px;">P33</td> <td style="padding: 2px 10px;">P32</td> <td style="padding: 2px 10px;">P31</td> <td style="padding: 2px 10px;">P30</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> <p style="text-align: center; margin-top: 10px;">ポート3の各端子のデータを格納</p>				P37	P36	P35	P34	P33	P32	P31	P30
P37	P36	P35	P34	P33	P32	P31	P30				
PDR4 ポートデータレジスタ4	H'D7		I/Oポート								
<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 10px;">—</td> <td style="padding: 2px 10px;">—</td> <td style="padding: 2px 10px;">—</td> <td style="padding: 2px 10px;">—</td> <td style="padding: 2px 10px;">P43</td> <td style="padding: 2px 10px;">P42</td> <td style="padding: 2px 10px;">P41</td> <td style="padding: 2px 10px;">P40</td> </tr> </table> <p>初期値： 1 1 1 1 1 0 0 0</p> <p>R/W : — — — — R R/W R/W R/W</p> <p style="text-align: center; margin-top: 10px;">P43端子状態を読み出す</p> <p style="text-align: right; margin-right: 100px;">ポートP42～P40端子のデータを格納</p>				—	—	—	—	P43	P42	P41	P40
—	—	—	—	P43	P42	P41	P40				
PDR5 ポートデータレジスタ5	H'D8		I/Oポート								
<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 10px;">P57</td> <td style="padding: 2px 10px;">P56</td> <td style="padding: 2px 10px;">P55</td> <td style="padding: 2px 10px;">P54</td> <td style="padding: 2px 10px;">P53</td> <td style="padding: 2px 10px;">P52</td> <td style="padding: 2px 10px;">P51</td> <td style="padding: 2px 10px;">P50</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> <p style="text-align: center; margin-top: 10px;">ポート5の各端子のデータを格納</p>				P57	P56	P55	P54	P53	P52	P51	P50
P57	P56	P55	P54	P53	P52	P51	P50				
PDR6 ポートデータレジスタ6	H'D9		I/Oポート								
<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 10px;">P67</td> <td style="padding: 2px 10px;">P66</td> <td style="padding: 2px 10px;">P65</td> <td style="padding: 2px 10px;">P64</td> <td style="padding: 2px 10px;">P63</td> <td style="padding: 2px 10px;">P62</td> <td style="padding: 2px 10px;">P61</td> <td style="padding: 2px 10px;">P60</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> <p style="text-align: center; margin-top: 10px;">ポート6の各端子のデータを格納</p>				P67	P66	P65	P64	P63	P62	P61	P60
P67	P66	P65	P64	P63	P62	P61	P60				

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: 10px auto; border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 2px 5px;">P7₇</td> <td style="padding: 2px 5px;">P7₆</td> <td style="padding: 2px 5px;">P7₅</td> <td style="padding: 2px 5px;">P7₄</td> <td style="padding: 2px 5px;">P7₃</td> <td style="padding: 2px 5px;">P7₂</td> <td style="padding: 2px 5px;">P7₁</td> <td style="padding: 2px 5px;">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> <p style="text-align: center; margin-top: 10px;">└──────────────────┘ ポート7の各端子のデータを格納</p>				P7 ₇	P7 ₆	P7 ₅	P7 ₄	P7 ₃	P7 ₂	P7 ₁	P7 ₀
P7 ₇	P7 ₆	P7 ₅	P7 ₄	P7 ₃	P7 ₂	P7 ₁	P7 ₀				
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: 10px auto; border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 2px 5px;">P8₇</td> <td style="padding: 2px 5px;">P8₆</td> <td style="padding: 2px 5px;">P8₅</td> <td style="padding: 2px 5px;">P8₄</td> <td style="padding: 2px 5px;">P8₃</td> <td style="padding: 2px 5px;">P8₂</td> <td style="padding: 2px 5px;">P8₁</td> <td style="padding: 2px 5px;">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> <p style="text-align: center; margin-top: 10px;">└──────────────────┘ ポート8の各端子のデータを格納</p>				P8 ₇	P8 ₆	P8 ₅	P8 ₄	P8 ₃	P8 ₂	P8 ₁	P8 ₀
P8 ₇	P8 ₆	P8 ₅	P8 ₄	P8 ₃	P8 ₂	P8 ₁	P8 ₀				
PDR9 ポートデータレジスタ9	H'DC		I/Oポート								
<p>ビット： 7 6 5 4 3 2 1 0</p> <table border="1" style="margin: 10px auto; border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 2px 5px;">P9₇</td> <td style="padding: 2px 5px;">P9₆</td> <td style="padding: 2px 5px;">P9₅</td> <td style="padding: 2px 5px;">P9₄</td> <td style="padding: 2px 5px;">P9₃</td> <td style="padding: 2px 5px;">P9₂</td> <td style="padding: 2px 5px;">P9₁</td> <td style="padding: 2px 5px;">P9₀</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> <p style="text-align: center; margin-top: 10px;">└──────────────────┘ ポート9の各端子のデータを格納</p>				P9 ₇	P9 ₆	P9 ₅	P9 ₄	P9 ₃	P9 ₂	P9 ₁	P9 ₀
P9 ₇	P9 ₆	P9 ₅	P9 ₄	P9 ₃	P9 ₂	P9 ₁	P9 ₀				
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: 10px auto; border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 2px 5px;">—</td> <td style="padding: 2px 5px;">—</td> <td style="padding: 2px 5px;">—</td> <td style="padding: 2px 5px;">—</td> <td style="padding: 2px 5px;">PA₃</td> <td style="padding: 2px 5px;">PA₂</td> <td style="padding: 2px 5px;">PA₁</td> <td style="padding: 2px 5px;">PA₀</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 R/W</p> <p style="text-align: center; margin-top: 10px;">└──────────────────┘ ポートAの各端子のデータを格納</p>				—	—	—	—	PA ₃	PA ₂	PA ₁	PA ₀
—	—	—	—	PA ₃	PA ₂	PA ₁	PA ₀				

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: 0 auto; border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 2px 5px;">PB7</td> <td style="padding: 2px 5px;">PB6</td> <td style="padding: 2px 5px;">PB5</td> <td style="padding: 2px 5px;">PB4</td> <td style="padding: 2px 5px;">PB3</td> <td style="padding: 2px 5px;">PB2</td> <td style="padding: 2px 5px;">PB1</td> <td style="padding: 2px 5px;">PB0</td> </tr> </table> <p>R/W : R R R R R R R R</p> <p style="text-align: center;">ポートBの各端子状態を読み出す</p>				PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0				
PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0								
PDCR ポートデータレジスタC	H'DF		I/Oポート												
<p>ビット： 7 6 5 4 3 2 1 0</p> <table border="1" style="margin: 0 auto; border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 2px 5px;">—</td> <td style="padding: 2px 5px;">—</td> <td style="padding: 2px 5px;">—</td> <td style="padding: 2px 5px;">—</td> <td style="padding: 2px 5px;">PC3</td> <td style="padding: 2px 5px;">PC2</td> <td style="padding: 2px 5px;">PC1</td> <td style="padding: 2px 5px;">PC0</td> </tr> </table> <p>R/W : — — — — R R R R</p> <p style="text-align: center;">ポートCの各端子状態を読み出す</p>				—	—	—	—	PC3	PC2	PC1	PC0				
—	—	—	—	PC3	PC2	PC1	PC0								
PUCR1 ポートプルアップコントロールレジスタ1	H'E0		I/Oポート												
<p>ビット： 7 6 5 4 3 2 1 0</p> <table border="1" style="margin: 0 auto; border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 2px 5px;">PUCR17</td> <td style="padding: 2px 5px;">PUCR16</td> <td style="padding: 2px 5px;">PUCR15</td> <td style="padding: 2px 5px;">PUCR14</td> <td style="padding: 2px 5px;">PUCR13</td> <td style="padding: 2px 5px;">PUCR12</td> <td style="padding: 2px 5px;">PUCR11</td> <td style="padding: 2px 5px;">PUCR10</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;">ポート1入力プルアップMOSコントロール</p> <table border="1" style="margin: 0 auto; border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 2px 5px;">0</td> <td style="padding: 2px 5px;">入力プルアップMOSはOFF</td> </tr> <tr> <td style="padding: 2px 5px;">1</td> <td style="padding: 2px 5px;">入力プルアップMOSはON</td> </tr> </table> <p style="text-align: center;">【注】 PCR1を"0"に指定したとき（入力ポートに指定）</p>				PUCR17	PUCR16	PUCR15	PUCR14	PUCR13	PUCR12	PUCR11	PUCR10	0	入力プルアップMOSはOFF	1	入力プルアップMOSはON
PUCR17	PUCR16	PUCR15	PUCR14	PUCR13	PUCR12	PUCR11	PUCR10								
0	入力プルアップMOSはOFF														
1	入力プルアップMOSはON														
PUCR3 ポートプルアップコントロールレジスタ3	H'E1		I/Oポート												
<p>ビット： 7 6 5 4 3 2 1 0</p> <table border="1" style="margin: 0 auto; border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 2px 5px;">PUCR37</td> <td style="padding: 2px 5px;">PUCR36</td> <td style="padding: 2px 5px;">PUCR35</td> <td style="padding: 2px 5px;">PUCR34</td> <td style="padding: 2px 5px;">PUCR33</td> <td style="padding: 2px 5px;">PUCR32</td> <td style="padding: 2px 5px;">PUCR31</td> <td style="padding: 2px 5px;">PUCR30</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;">ポート3入力プルアップMOSコントロール</p> <table border="1" style="margin: 0 auto; border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 2px 5px;">0</td> <td style="padding: 2px 5px;">入力プルアップMOSはOFF</td> </tr> <tr> <td style="padding: 2px 5px;">1</td> <td style="padding: 2px 5px;">入力プルアップMOSはON</td> </tr> </table> <p style="text-align: center;">【注】 PCR3を"0"に指定したとき（入力ポートに指定）</p>				PUCR37	PUCR36	PUCR35	PUCR34	PUCR33	PUCR32	PUCR31	PUCR30	0	入力プルアップMOSはOFF	1	入力プルアップMOSはON
PUCR37	PUCR36	PUCR35	PUCR34	PUCR33	PUCR32	PUCR31	PUCR30								
0	入力プルアップMOSはOFF														
1	入力プルアップMOSはON														

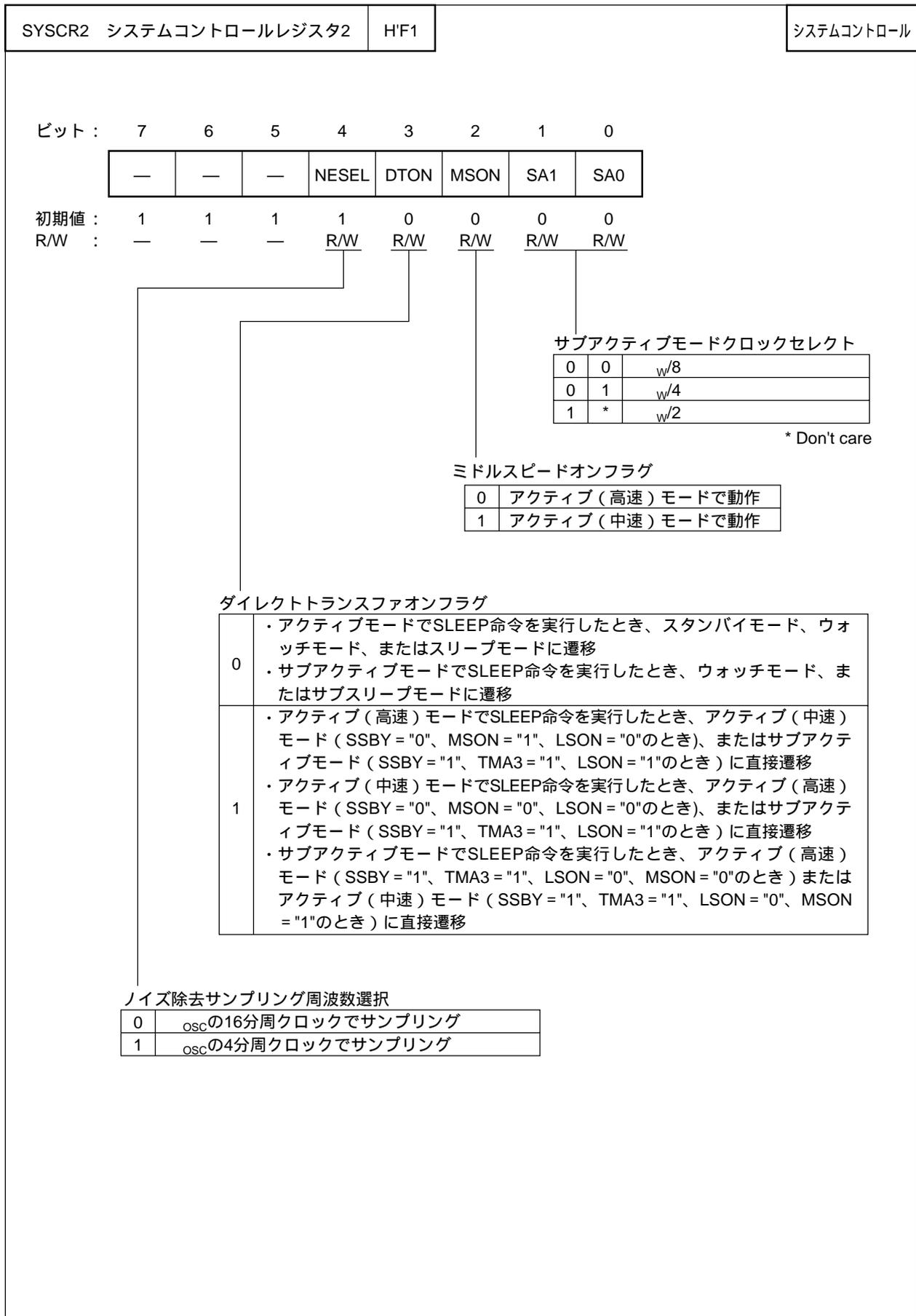
PUCR5 ポートプルアップコントロールレジスタ5		H'E2	I/Oポート								
ビット： 7 6 5 4 3 2 1 0											
<table border="1" style="width:100%; text-align:center;"> <tr> <td>PUCR5₇</td><td>PUCR5₆</td><td>PUCR5₅</td><td>PUCR5₄</td><td>PUCR5₃</td><td>PUCR5₂</td><td>PUCR5₁</td><td>PUCR5₀</td> </tr> </table>				PUCR5 ₇	PUCR5 ₆	PUCR5 ₅	PUCR5 ₄	PUCR5 ₃	PUCR5 ₂	PUCR5 ₁	PUCR5 ₀
PUCR5 ₇	PUCR5 ₆	PUCR5 ₅	PUCR5 ₄	PUCR5 ₃	PUCR5 ₂	PUCR5 ₁	PUCR5 ₀				
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											
ポート5入力プルアップMOSコントロール											
<table border="1" style="width:100%; text-align:center;"> <tr> <td>0</td><td>入力プルアップMOSはOFF</td> </tr> <tr> <td>1</td><td>入力プルアップMOSはON</td> </tr> </table>				0	入力プルアップMOSはOFF	1	入力プルアップMOSはON				
0	入力プルアップMOSはOFF										
1	入力プルアップMOSはON										
【注】 PCR5を"0"に指定したとき（入力ポートに指定）											
PUCR6 ポートプルアップコントロールレジスタ6		H'E3	I/Oポート								
ビット： 7 6 5 4 3 2 1 0											
<table border="1" style="width:100%; text-align:center;"> <tr> <td>PUCR6₇</td><td>PUCR6₆</td><td>PUCR6₅</td><td>PUCR6₄</td><td>PUCR6₃</td><td>PUCR6₂</td><td>PUCR6₁</td><td>PUCR6₀</td> </tr> </table>				PUCR6 ₇	PUCR6 ₆	PUCR6 ₅	PUCR6 ₄	PUCR6 ₃	PUCR6 ₂	PUCR6 ₁	PUCR6 ₀
PUCR6 ₇	PUCR6 ₆	PUCR6 ₅	PUCR6 ₄	PUCR6 ₃	PUCR6 ₂	PUCR6 ₁	PUCR6 ₀				
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											
ポート6入力プルアップMOSコントロール											
<table border="1" style="width:100%; text-align:center;"> <tr> <td>0</td><td>入力プルアップMOSはOFF</td> </tr> <tr> <td>1</td><td>入力プルアップMOSはON</td> </tr> </table>				0	入力プルアップMOSはOFF	1	入力プルアップMOSはON				
0	入力プルアップMOSはOFF										
1	入力プルアップMOSはON										
【注】 PCR6を"0"に指定したとき（入力ポートに指定）											
PCR1 ポートコントロールレジスタ1		H'E4	I/Oポート								
ビット： 7 6 5 4 3 2 1 0											
<table border="1" style="width:100%; text-align:center;"> <tr> <td>PCR1₇</td><td>PCR1₆</td><td>PCR1₅</td><td>PCR1₄</td><td>PCR1₃</td><td>PCR1₂</td><td>PCR1₁</td><td>PCR1₀</td> </tr> </table>				PCR1 ₇	PCR1 ₆	PCR1 ₅	PCR1 ₄	PCR1 ₃	PCR1 ₂	PCR1 ₁	PCR1 ₀
PCR1 ₇	PCR1 ₆	PCR1 ₅	PCR1 ₄	PCR1 ₃	PCR1 ₂	PCR1 ₁	PCR1 ₀				
初期値： 0 0 0 0 0 0 0 0											
R/W： W W W W W W W W											
ポート1入出力選択											
<table border="1" style="width:100%; text-align:center;"> <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										

PCR2 ポートコントロールレジスタ2	H'E5		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 5px;">PCR2₇</td> <td style="padding: 2px 5px;">PCR2₆</td> <td style="padding: 2px 5px;">PCR2₅</td> <td style="padding: 2px 5px;">PCR2₄</td> <td style="padding: 2px 5px;">PCR2₃</td> <td style="padding: 2px 5px;">PCR2₂</td> <td style="padding: 2px 5px;">PCR2₁</td> <td style="padding: 2px 5px;">PCR2₀</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2入出力選択</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;">入力ポート</td> </tr> <tr> <td style="padding: 2px 5px;">1</td> <td style="padding: 2px 5px;">出力ポート</td> </tr> </table> </div>				PCR2 ₇	PCR2 ₆	PCR2 ₅	PCR2 ₄	PCR2 ₃	PCR2 ₂	PCR2 ₁	PCR2 ₀	0	入力ポート	1	出力ポート
PCR2 ₇	PCR2 ₆	PCR2 ₅	PCR2 ₄	PCR2 ₃	PCR2 ₂	PCR2 ₁	PCR2 ₀								
0	入力ポート														
1	出力ポート														
PCR3 ポートコントロールレジスタ3	H'E6		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 5px;">PCR3₇</td> <td style="padding: 2px 5px;">PCR3₆</td> <td style="padding: 2px 5px;">PCR3₅</td> <td style="padding: 2px 5px;">PCR3₄</td> <td style="padding: 2px 5px;">PCR3₃</td> <td style="padding: 2px 5px;">PCR3₂</td> <td style="padding: 2px 5px;">PCR3₁</td> <td style="padding: 2px 5px;">PCR3₀</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3入出力選択</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;">入力ポート</td> </tr> <tr> <td style="padding: 2px 5px;">1</td> <td style="padding: 2px 5px;">出力ポート</td> </tr> </table> </div>				PCR3 ₇	PCR3 ₆	PCR3 ₅	PCR3 ₄	PCR3 ₃	PCR3 ₂	PCR3 ₁	PCR3 ₀	0	入力ポート	1	出力ポート
PCR3 ₇	PCR3 ₆	PCR3 ₅	PCR3 ₄	PCR3 ₃	PCR3 ₂	PCR3 ₁	PCR3 ₀								
0	入力ポート														
1	出力ポート														
PCR4 ポートコントロールレジスタ4	H'E7		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 5px;">—</td> <td style="padding: 2px 5px;">PCR4₂</td> <td style="padding: 2px 5px;">PCR4₁</td> <td style="padding: 2px 5px;">PCR4₀</td> </tr> </table> <p>初期値： 1 1 1 1 1 0 0 0</p> <p>R/W : — — — — — W W W</p> <div style="text-align: center; margin-top: 10px;"> <p>ポート4入出力選択</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;">入力ポート</td> </tr> <tr> <td style="padding: 2px 5px;">1</td> <td style="padding: 2px 5px;">出力ポート</td> </tr> </table> </div>				—	—	—	—	—	PCR4 ₂	PCR4 ₁	PCR4 ₀	0	入力ポート	1	出力ポート
—	—	—	—	—	PCR4 ₂	PCR4 ₁	PCR4 ₀								
0	入力ポート														
1	出力ポート														

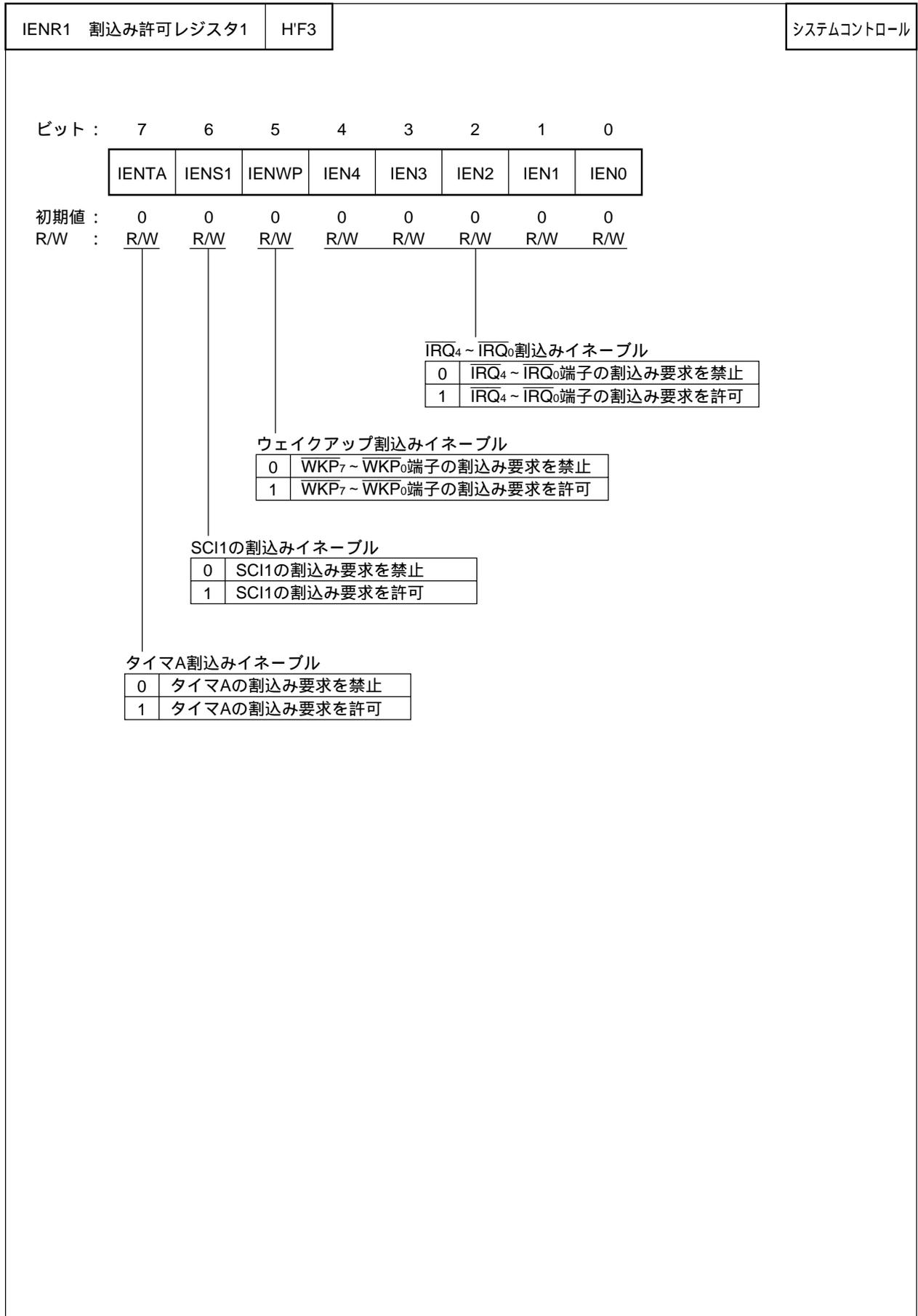
PCR5	ポートコントロールレジスタ5	H'E8	I/Oポート												
ビット： 7 6 5 4 3 2 1 0 <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>PCR5₇</td> <td>PCR5₆</td> <td>PCR5₅</td> <td>PCR5₄</td> <td>PCR5₃</td> <td>PCR5₂</td> <td>PCR5₁</td> <td>PCR5₀</td> </tr> </table> 初期値： 0 0 0 0 0 0 0 0 R/W： W W W W W W W W <div style="text-align: center; margin-top: 10px;"> 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> </div>				PCR5 ₇	PCR5 ₆	PCR5 ₅	PCR5 ₄	PCR5 ₃	PCR5 ₂	PCR5 ₁	PCR5 ₀	0	入力ポート	1	出力ポート
PCR5 ₇	PCR5 ₆	PCR5 ₅	PCR5 ₄	PCR5 ₃	PCR5 ₂	PCR5 ₁	PCR5 ₀								
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 <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>PCR6₇</td> <td>PCR6₆</td> <td>PCR6₅</td> <td>PCR6₄</td> <td>PCR6₃</td> <td>PCR6₂</td> <td>PCR6₁</td> <td>PCR6₀</td> </tr> </table> 初期値： 0 0 0 0 0 0 0 0 R/W： W W W W W W W W <div style="text-align: center; margin-top: 10px;"> 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> </div>				PCR6 ₇	PCR6 ₆	PCR6 ₅	PCR6 ₄	PCR6 ₃	PCR6 ₂	PCR6 ₁	PCR6 ₀	0	入力ポート	1	出力ポート
PCR6 ₇	PCR6 ₆	PCR6 ₅	PCR6 ₄	PCR6 ₃	PCR6 ₂	PCR6 ₁	PCR6 ₀								
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												
ビット： 7 6 5 4 3 2 1 0 <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>PCR7₇</td> <td>PCR7₆</td> <td>PCR7₅</td> <td>PCR7₄</td> <td>PCR7₃</td> <td>PCR7₂</td> <td>PCR7₁</td> <td>PCR7₀</td> </tr> </table> 初期値： 0 0 0 0 0 0 0 0 R/W： W W W W W W W W <div style="text-align: center; margin-top: 10px;"> ポート7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> </div>				PCR7 ₇	PCR7 ₆	PCR7 ₅	PCR7 ₄	PCR7 ₃	PCR7 ₂	PCR7 ₁	PCR7 ₀	0	入力ポート	1	出力ポート
PCR7 ₇	PCR7 ₆	PCR7 ₅	PCR7 ₄	PCR7 ₃	PCR7 ₂	PCR7 ₁	PCR7 ₀								
0	入力ポート														
1	出力ポート														

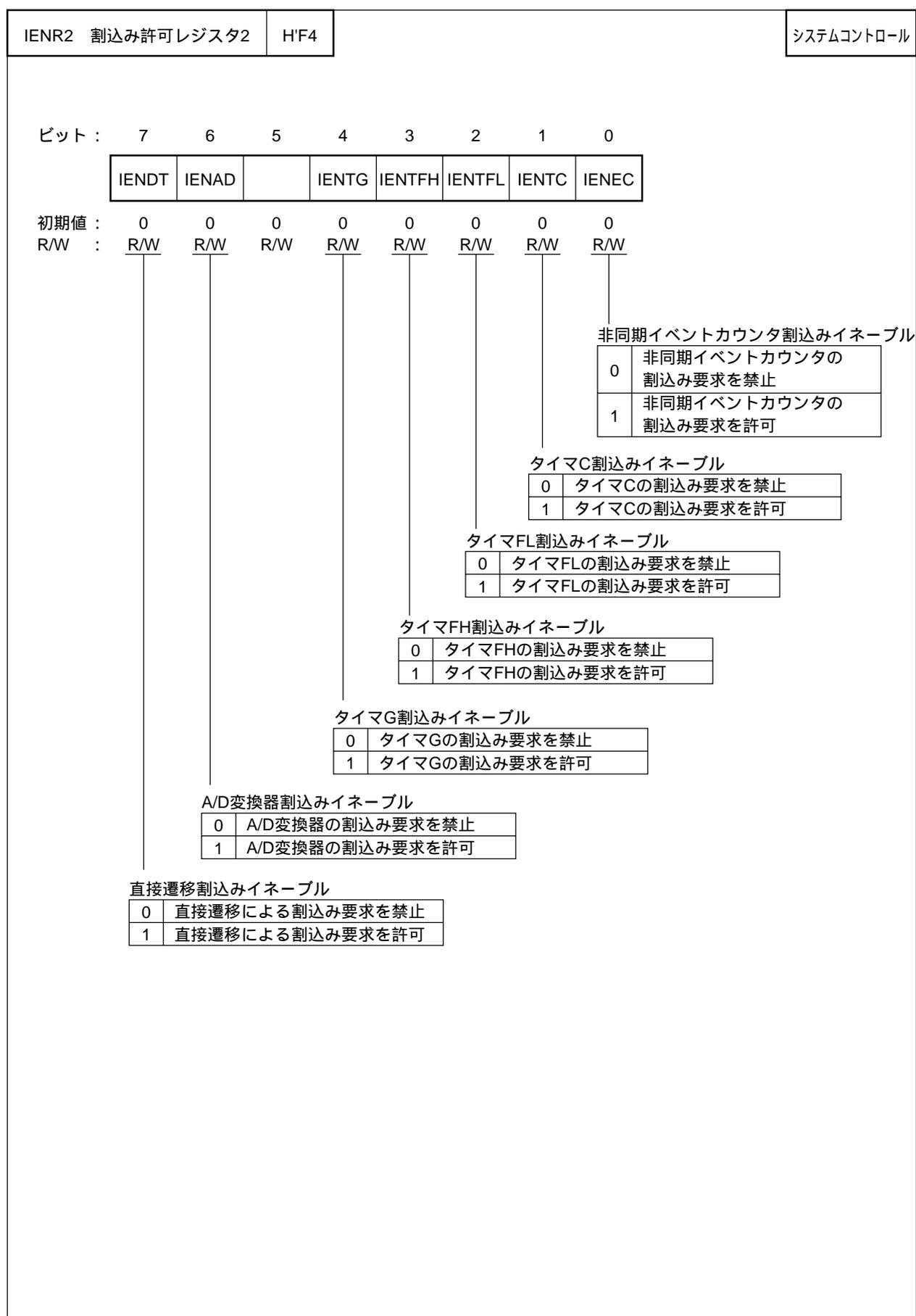
PCR8 ポートコントロールレジスタ8	H'EB	I/Oポート												
<p>ビット： 7 6 5 4 3 2 1 0</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 2px 5px;">PCR8₇</td> <td style="padding: 2px 5px;">PCR8₆</td> <td style="padding: 2px 5px;">PCR8₅</td> <td style="padding: 2px 5px;">PCR8₄</td> <td style="padding: 2px 5px;">PCR8₃</td> <td style="padding: 2px 5px;">PCR8₂</td> <td style="padding: 2px 5px;">PCR8₁</td> <td style="padding: 2px 5px;">PCR8₀</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 5px;">0</td> <td style="padding: 2px 5px;">入力ポート</td> </tr> <tr> <td style="padding: 2px 5px;">1</td> <td style="padding: 2px 5px;">出力ポート</td> </tr> </table> </div>			PCR8 ₇	PCR8 ₆	PCR8 ₅	PCR8 ₄	PCR8 ₃	PCR8 ₂	PCR8 ₁	PCR8 ₀	0	入力ポート	1	出力ポート
PCR8 ₇	PCR8 ₆	PCR8 ₅	PCR8 ₄	PCR8 ₃	PCR8 ₂	PCR8 ₁	PCR8 ₀							
0	入力ポート													
1	出力ポート													
PCR9 ポートコントロールレジスタ9	H'EC	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; text-align: center;"> <tr> <td style="padding: 2px 5px;">PCR9₇</td> <td style="padding: 2px 5px;">PCR9₆</td> <td style="padding: 2px 5px;">PCR9₅</td> <td style="padding: 2px 5px;">PCR9₄</td> <td style="padding: 2px 5px;">PCR9₃</td> <td style="padding: 2px 5px;">PCR9₂</td> <td style="padding: 2px 5px;">PCR9₁</td> <td style="padding: 2px 5px;">PCR9₀</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9入出力選択</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;">入力ポート</td> </tr> <tr> <td style="padding: 2px 5px;">1</td> <td style="padding: 2px 5px;">出力ポート</td> </tr> </table> </div>			PCR9 ₇	PCR9 ₆	PCR9 ₅	PCR9 ₄	PCR9 ₃	PCR9 ₂	PCR9 ₁	PCR9 ₀	0	入力ポート	1	出力ポート
PCR9 ₇	PCR9 ₆	PCR9 ₅	PCR9 ₄	PCR9 ₃	PCR9 ₂	PCR9 ₁	PCR9 ₀							
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; text-align: center;"> <tr> <td style="padding: 2px 5px;">—</td> <td style="padding: 2px 5px;">—</td> <td style="padding: 2px 5px;">—</td> <td style="padding: 2px 5px;">—</td> <td style="padding: 2px 5px;">PCRA₃</td> <td style="padding: 2px 5px;">PCRA₂</td> <td style="padding: 2px 5px;">PCRA₁</td> <td style="padding: 2px 5px;">PCRA₀</td> </tr> </table> <p>初期値： 1 1 1 1 0 0 0 0</p> <p>R/W : — — — — W W W W</p> <div style="text-align: center; margin-top: 10px;"> <p>ポートA入出力選択</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;">入力ポート</td> </tr> <tr> <td style="padding: 2px 5px;">1</td> <td style="padding: 2px 5px;">出力ポート</td> </tr> </table> </div>			—	—	—	—	PCRA ₃	PCRA ₂	PCRA ₁	PCRA ₀	0	入力ポート	1	出力ポート
—	—	—	—	PCRA ₃	PCRA ₂	PCRA ₁	PCRA ₀							
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									
ビット : 7 6 5 4 3 2 1 0																	
SSBY		STS2		STS1		STS0		LSON		—		MA1		MA0			
初期値 : 0		0		0		0		0		1		1		1			
R/W : R/W		R/W		R/W		R/W		R/W		—		R/W		R/W			
												アクティブ (中速) モードクロックセレクト					
												0 0		osc/16			
												0 1		osc/32			
												1 0		osc/64			
												1 1		osc/128			
												ロースピードオンフラグ					
												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)			
												スタンバイタイムセレクト2~0					
												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 0 0		待機時間 = 131072ステート			
												1 0 1		待機時間 = 2ステート			
												1 1 0		待機時間 = 8ステート			
												1 1 1		待機時間 = 16ステート			
												ソフトウェアスタンバイ					
												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			
												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			



IEGR IRQエッジセレクトレジスタ			H'F2					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:	1	1	1	0	0	0	0	0				
R/W:	—	—	—	R/W	R/W	R/W	R/W	R/W				
				<p>IRQ₀エッジセレクト</p> <table border="1"> <tr> <td>0</td> <td>IRQ₀端子入力の立下がりエッジを検出</td> </tr> <tr> <td>1</td> <td>IRQ₀端子入力の立上がりエッジを検出</td> </tr> </table>					0	IRQ ₀ 端子入力の立下がりエッジを検出	1	IRQ ₀ 端子入力の立上がりエッジを検出
0	IRQ ₀ 端子入力の立下がりエッジを検出											
1	IRQ ₀ 端子入力の立上がりエッジを検出											
				<p>IRQ₁エッジセレクト</p> <table border="1"> <tr> <td>0</td> <td>IRQ₁端子、TMIC端子入力の立下がりエッジを検出</td> </tr> <tr> <td>1</td> <td>IRQ₁端子、TMIC端子入力の立上がりエッジを検出</td> </tr> </table>					0	IRQ ₁ 端子、TMIC端子入力の立下がりエッジを検出	1	IRQ ₁ 端子、TMIC端子入力の立上がりエッジを検出
0	IRQ ₁ 端子、TMIC端子入力の立下がりエッジを検出											
1	IRQ ₁ 端子、TMIC端子入力の立上がりエッジを検出											
				<p>IRQ₂エッジセレクト</p> <table border="1"> <tr> <td>0</td> <td>IRQ₂端子入力の立下がりエッジを検出</td> </tr> <tr> <td>1</td> <td>IRQ₂端子入力の立上がりエッジを検出</td> </tr> </table>					0	IRQ ₂ 端子入力の立下がりエッジを検出	1	IRQ ₂ 端子入力の立上がりエッジを検出
0	IRQ ₂ 端子入力の立下がりエッジを検出											
1	IRQ ₂ 端子入力の立上がりエッジを検出											
				<p>IRQ₃エッジセレクト</p> <table border="1"> <tr> <td>0</td> <td>IRQ₃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>IRQ₃端子、TMIF端子入力の立上がりエッジを検出</td> </tr> </table>					0	IRQ ₃ 端子、TMIF端子入力の立下がりエッジを検出	1	IRQ ₃ 端子、TMIF端子入力の立上がりエッジを検出
0	IRQ ₃ 端子、TMIF端子入力の立下がりエッジを検出											
1	IRQ ₃ 端子、TMIF端子入力の立上がりエッジを検出											
				<p>IRQ₄エッジセレクト</p> <table border="1"> <tr> <td>0</td> <td>IRQ₄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>IRQ₄端子、ADTRG端子入力の立上がりエッジを検出</td> </tr> </table>					0	IRQ ₄ 端子、ADTRG端子入力の立下がりエッジを検出	1	IRQ ₄ 端子、ADTRG端子入力の立上がりエッジを検出
0	IRQ ₄ 端子、ADTRG端子入力の立下がりエッジを検出											
1	IRQ ₄ 端子、ADTRG端子入力の立上がりエッジを検出											

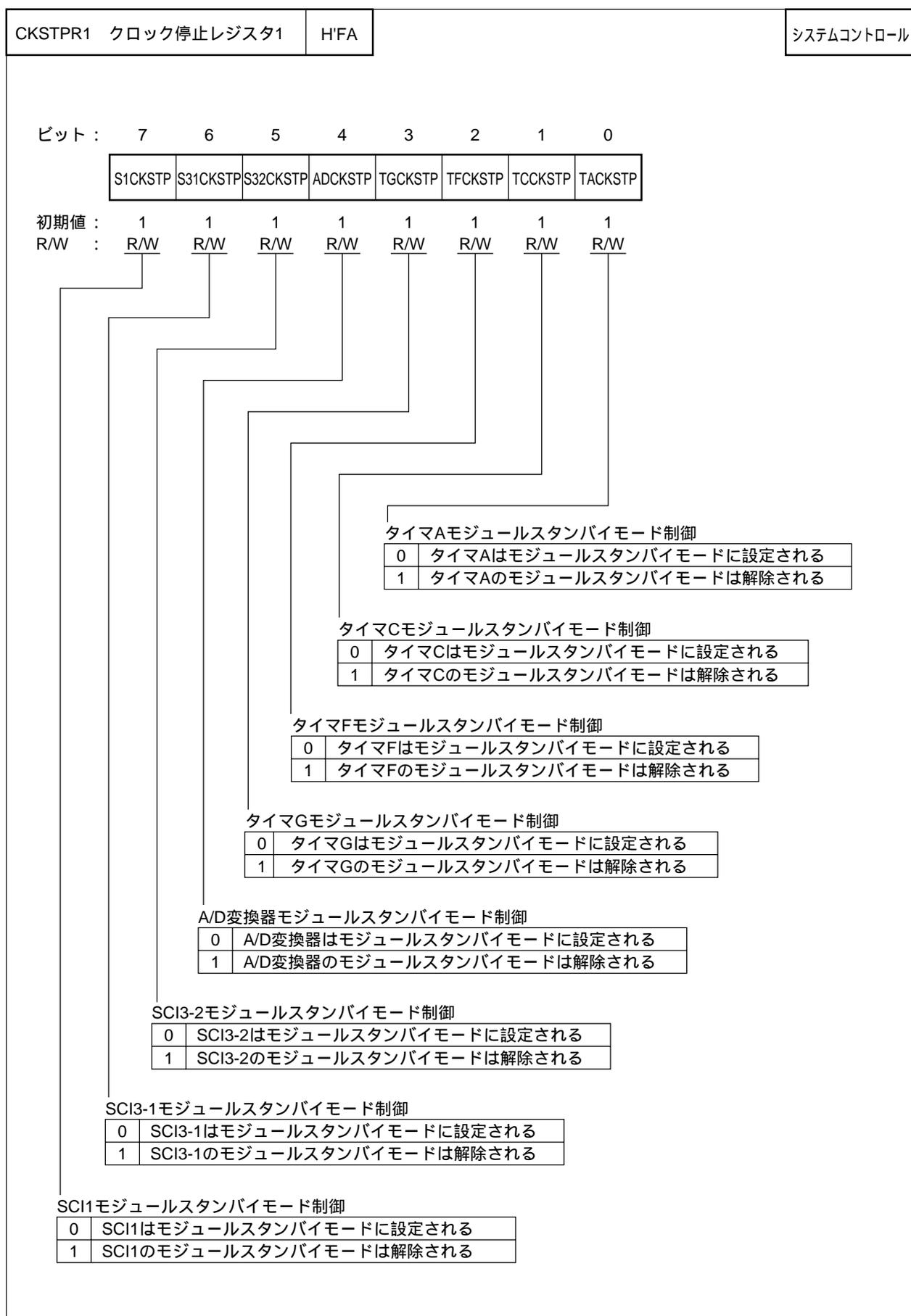


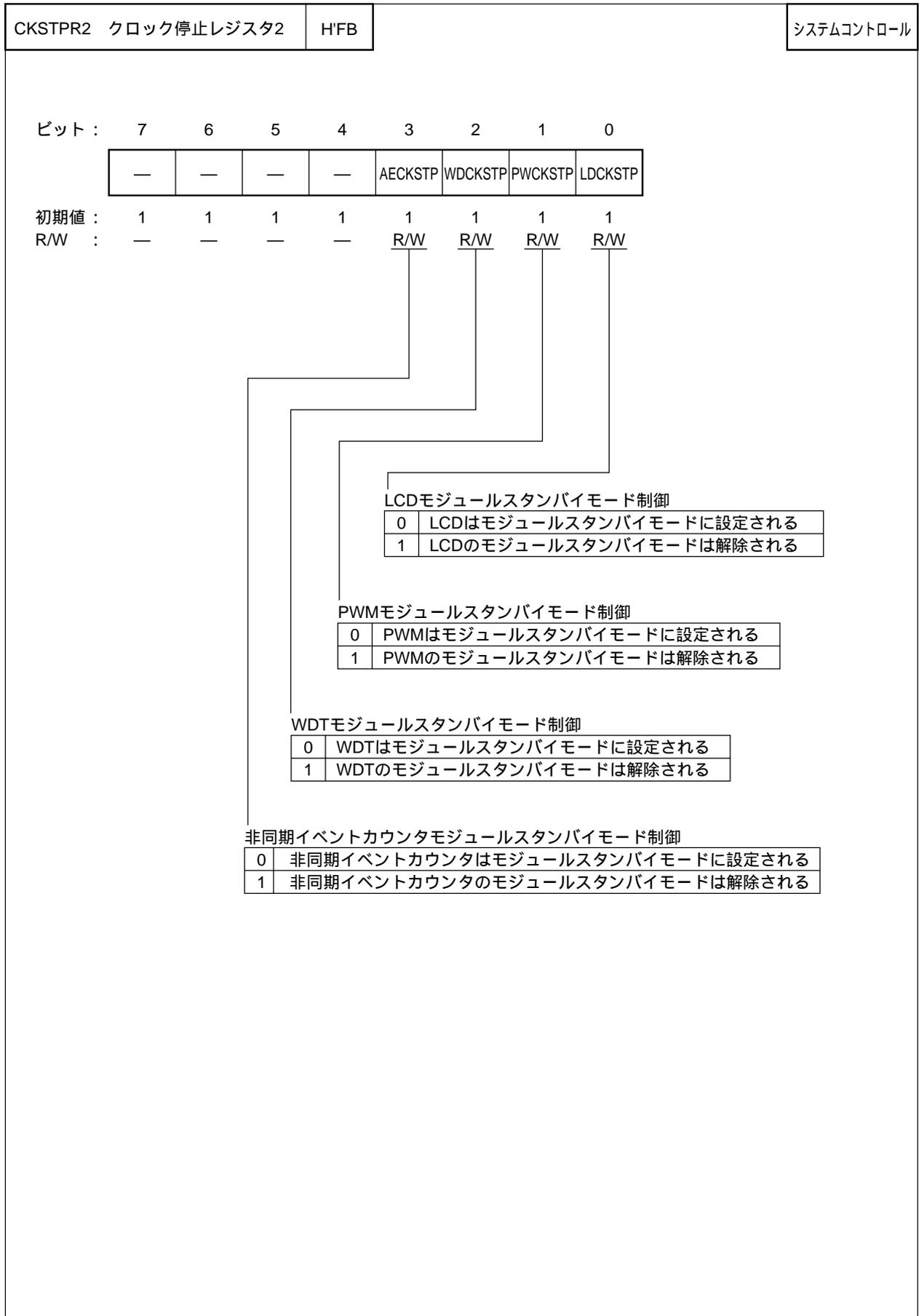


IRR1 割込み要求レジスタ1	H'F6	システムコントロール																				
<p>ビット： 7 6 5 4 3 2 1 0</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr> <td style="width: 5%;">IRRRTA</td> <td style="width: 5%;">IRRS1</td> <td style="width: 5%;">—</td> <td style="width: 5%;">IRRI4</td> <td style="width: 5%;">IRRI3</td> <td style="width: 5%;">IRRI2</td> <td style="width: 5%;">IRRI1</td> <td style="width: 5%;">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;">IRQ₄ ~ IRQ₀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;">0</td> <td>〔クリア条件〕 IRRI4 = "1"の状態(IRRI4に"0"をライトしたとき) IRRI3 ~ IRRI0についても同様</td> </tr> <tr> <td style="text-align: center;">1</td> <td>〔セット条件〕 IRQ₄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 IRRI3 ~ IRRI0についても同様</td> </tr> </table> <p style="margin-top: 10px; text-align: center;">SCI1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;">0</td> <td>〔クリア条件〕 IRRS1 = "1"の状態(IRRS1に"0"をライトしたとき)</td> </tr> <tr> <td style="text-align: center;">1</td> <td>〔セット条件〕 SCI1が転送完了したとき</td> </tr> </table> <p style="margin-top: 10px; text-align: center;">タイマA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;">0</td> <td>〔クリア条件〕 IRRRTA = "1"の状態(IRRRTAに"0"をライトしたとき)</td> </tr> <tr> <td style="text-align: center;">1</td> <td>〔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</td> </tr> </table> </div>			IRRRTA	IRRS1	—	IRRI4	IRRI3	IRRI2	IRRI1	IRRI0	0	〔クリア条件〕 IRRI4 = "1"の状態(IRRI4に"0"をライトしたとき) IRRI3 ~ IRRI0についても同様	1	〔セット条件〕 IRQ ₄ 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 IRRI3 ~ IRRI0についても同様	0	〔クリア条件〕 IRRS1 = "1"の状態(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〕 IRRRTA = "1"の状態(IRRRTAに"0"をライトしたとき)	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
IRRRTA	IRRS1	—	IRRI4	IRRI3	IRRI2	IRRI1	IRRI0															
0	〔クリア条件〕 IRRI4 = "1"の状態(IRRI4に"0"をライトしたとき) IRRI3 ~ IRRI0についても同様																					
1	〔セット条件〕 IRQ ₄ 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 IRRI3 ~ IRRI0についても同様																					
0	〔クリア条件〕 IRRS1 = "1"の状態(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〕 IRRRTA = "1"の状態(IRRRTAに"0"をライトしたとき)																					
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																					

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	IRRTC	IRREC																												
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)*																												
<p>非同期イベントカウンタ 割り込み要求フラグ</p> <table border="1"> <tr> <td>0</td> <td>〔クリア条件〕 IRREC = "1"の状態(IRRECに"0"をライトしたとき)</td> </tr> <tr> <td>1</td> <td>〔セット条件〕 非同期イベントカウンタのカウンタ値がオーバーフローしたとき</td> </tr> </table> <p>タイマC割り込み要求フラグ</p> <table border="1"> <tr> <td>0</td> <td>〔クリア条件〕 IRRTC = "1"の状態(IRRTCに"0"をライトしたとき)</td> </tr> <tr> <td>1</td> <td>〔セット条件〕 タイマCのカウンタ値がオーバーフロー (H'FF H'00) またはアンダ フロー (H'00 H'FF) したとき</td> </tr> </table> <p>タイマFL割り込み要求フラグ</p> <table border="1"> <tr> <td>0</td> <td>〔クリア条件〕 IRRTFL = "1"の状態(IRRTFLに"0"をライトしたとき)</td> </tr> <tr> <td>1</td> <td>〔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> <p>タイマFH割り込み要求フラグ</p> <table border="1"> <tr> <td>0</td> <td>〔クリア条件〕 IRRTFH = "1"の状態(IRRTFHに"0"をライトしたとき)</td> </tr> <tr> <td>1</td> <td>〔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、または、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> <p>タイマG割り込み要求フラグ</p> <table border="1"> <tr> <td>0</td> <td>〔クリア条件〕 IRRTG = "1"の状態(IRRTGに"0"をライトしたとき)</td> </tr> <tr> <td>1</td> <td>〔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がオーバーフローしたとき</td> </tr> </table> <p>A/D変換器割り込み要求フラグ</p> <table border="1"> <tr> <td>0</td> <td>〔クリア条件〕 IRRAD = "1"の状態(IRRADに"0"をライトしたとき)</td> </tr> <tr> <td>1</td> <td>〔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> <p>直接遷移割り込み要求フラグ</p> <table border="1"> <tr> <td>0</td> <td>〔クリア条件〕 IRRDT = "1"の状態(IRRDTに"0"をライトしたとき)</td> </tr> <tr> <td>1</td> <td>〔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〕 IRREC = "1"の状態(IRRECに"0"をライトしたとき)	1	〔セット条件〕 非同期イベントカウンタのカウンタ値がオーバーフローしたとき	0	〔クリア条件〕 IRRTC = "1"の状態(IRRTCに"0"をライトしたとき)	1	〔セット条件〕 タイマCのカウンタ値がオーバーフロー (H'FF H'00) またはアンダ フロー (H'00 H'FF) したとき	0	〔クリア条件〕 IRRTFL = "1"の状態(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〕 IRRTFH = "1"の状態(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、または、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〕 IRRTG = "1"の状態(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がオーバーフローしたとき	0	〔クリア条件〕 IRRAD = "1"の状態(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〕 IRRDT = "1"の状態(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〕 IRREC = "1"の状態(IRRECに"0"をライトしたとき)																																			
1	〔セット条件〕 非同期イベントカウンタのカウンタ値がオーバーフローしたとき																																			
0	〔クリア条件〕 IRRTC = "1"の状態(IRRTCに"0"をライトしたとき)																																			
1	〔セット条件〕 タイマCのカウンタ値がオーバーフロー (H'FF H'00) またはアンダ フロー (H'00 H'FF) したとき																																			
0	〔クリア条件〕 IRRTFL = "1"の状態(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〕 IRRTFH = "1"の状態(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、または、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〕 IRRTG = "1"の状態(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がオーバーフローしたとき																																			
0	〔クリア条件〕 IRRAD = "1"の状態(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〕 IRRDT = "1"の状態(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																																			
<p>【注】* フラグクリアのための "0"ライトのみ可能</p>																																				

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 10px;">IWPF7</td> <td style="padding: 2px 10px;">IWPF6</td> <td style="padding: 2px 10px;">IWPF5</td> <td style="padding: 2px 10px;">IWPF4</td> <td style="padding: 2px 10px;">IWPF3</td> <td style="padding: 2px 10px;">IWPF2</td> <td style="padding: 2px 10px;">IWPF1</td> <td style="padding: 2px 10px;">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> <div style="margin-left: 100px; margin-top: 10px;"> <p>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: 30px; text-align: center; vertical-align: middle;">0</td> <td style="padding: 2px;">〔クリア条件〕 IWPF7 = "1"の状態 IWPF7に"0"をライトしたとき IWPF6 ~ IWPF0についても同様</td> </tr> <tr> <td style="width: 30px; text-align: center; vertical-align: middle;">1</td> <td style="padding: 2px;">〔セット条件〕 WKP₇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 IWPF6 ~ IWPF0についても同様</td> </tr> </table> </div>			IWPF7	IWPF6	IWPF5	IWPF4	IWPF3	IWPF2	IWPF1	IWPF0	0	〔クリア条件〕 IWPF7 = "1"の状態 IWPF7に"0"をライトしたとき IWPF6 ~ IWPF0についても同様	1	〔セット条件〕 WKP ₇ 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 IWPF6 ~ IWPF0についても同様
IWPF7	IWPF6	IWPF5	IWPF4	IWPF3	IWPF2	IWPF1	IWPF0							
0	〔クリア条件〕 IWPF7 = "1"の状態 IWPF7に"0"をライトしたとき IWPF6 ~ IWPF0についても同様													
1	〔セット条件〕 WKP ₇ 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 IWPF6 ~ IWPF0についても同様													





C.2 ポート2ブロック図

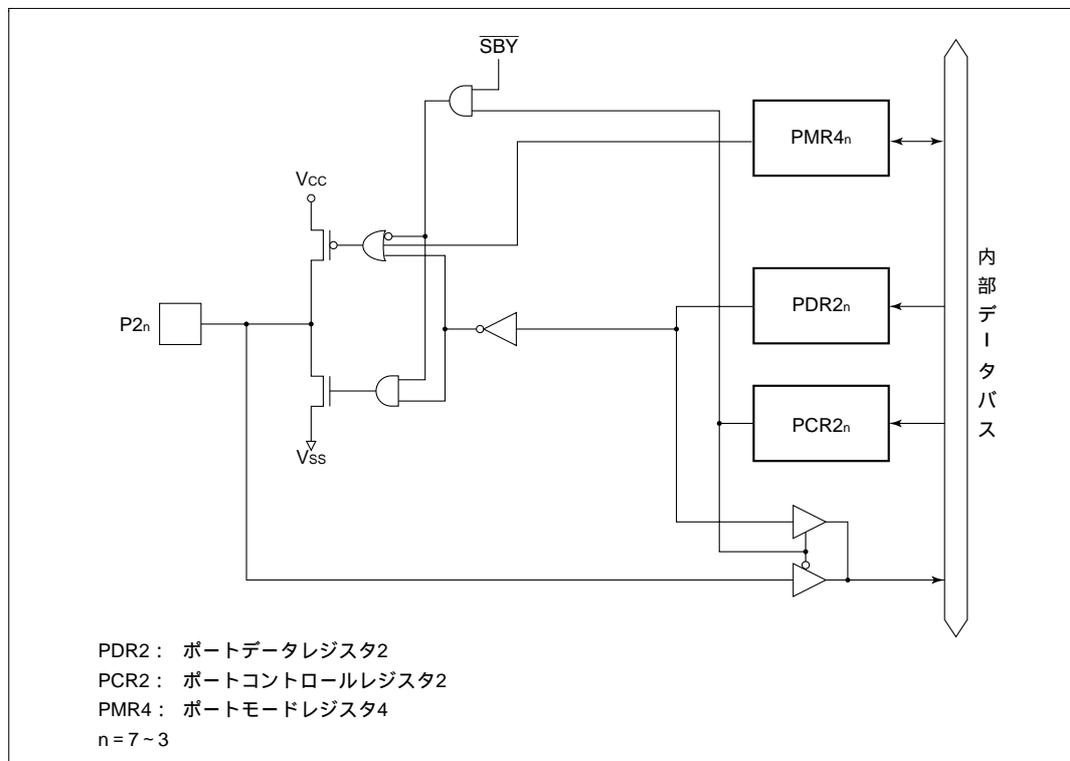


図 C.2 (a) ポート2 ブロック図 (P2₇ ~ P2₃端子)

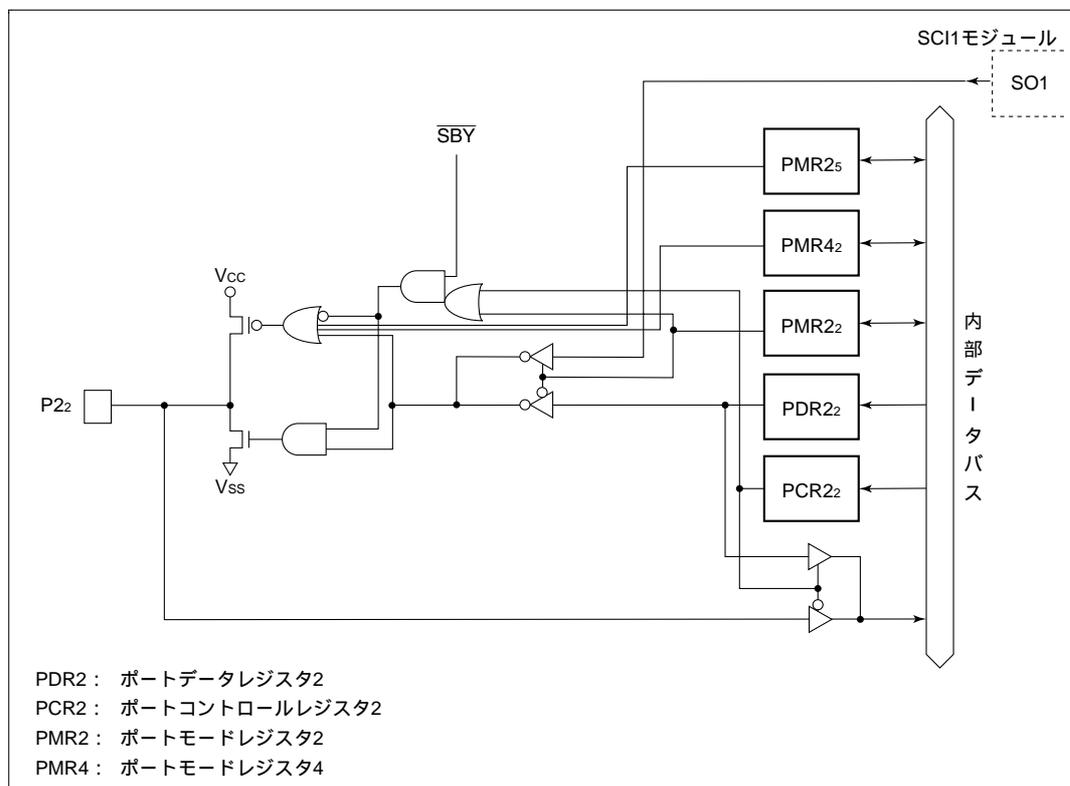


図 C.2 (b) ポート2 ブロック図 (P2₂端子)

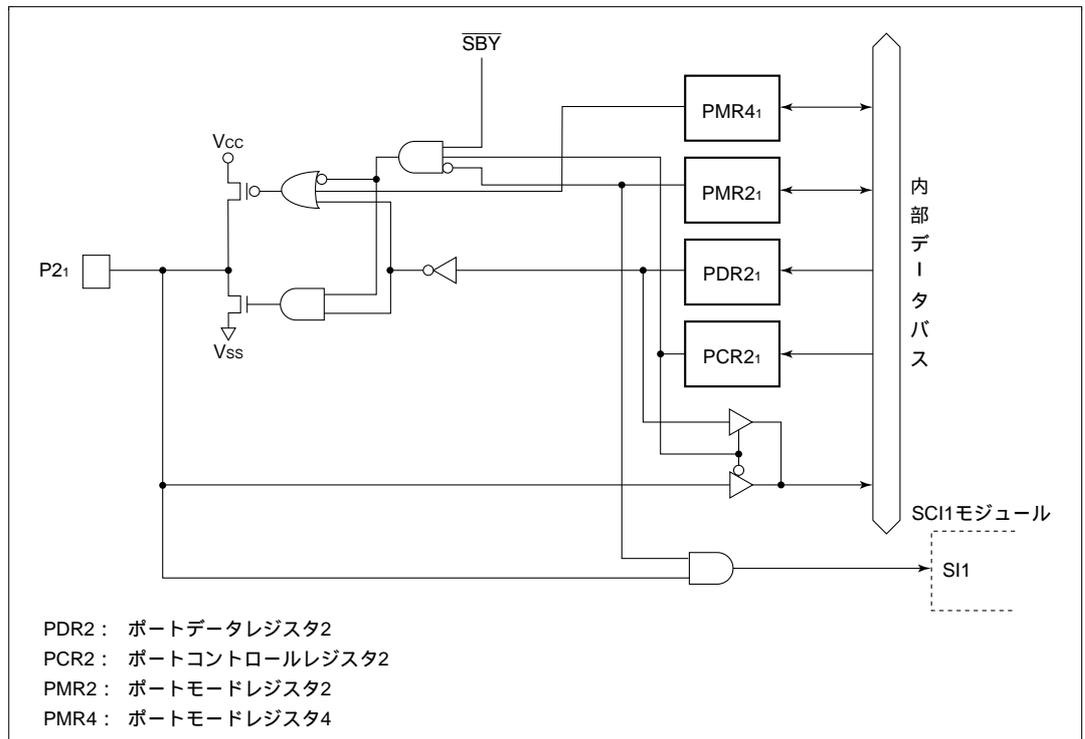


図 C.2 (c) ポート2 ブロック図 (P2₁端子)

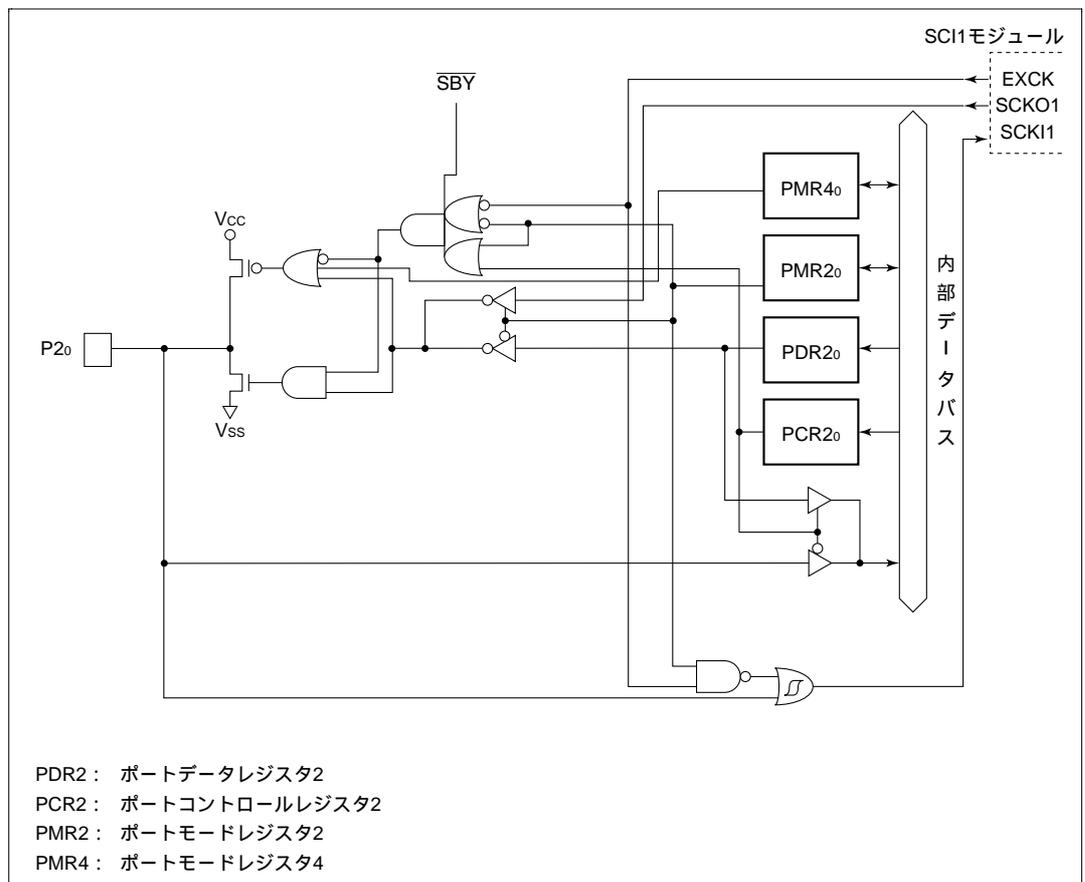


図 C.2 (d) ポート2 ブロック図 (P2₀端子)

C.3 ポート3ブロック図

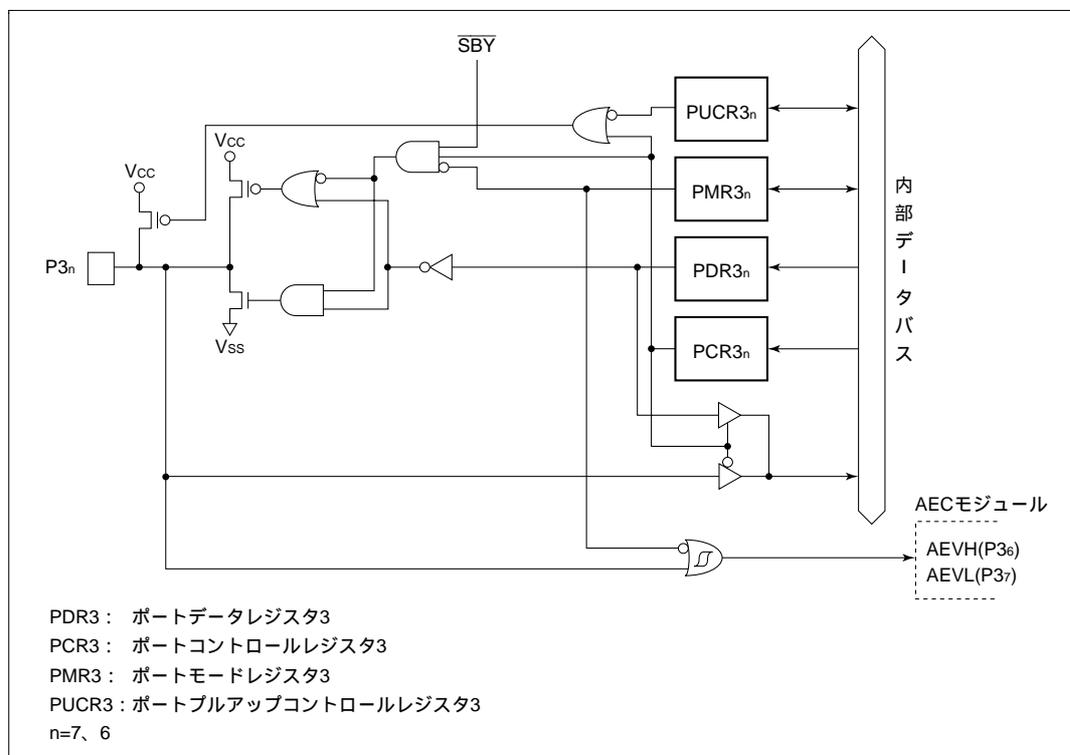


図 C.3 (a) ポート3 ブロック図 (P3₇~P3₆端子)

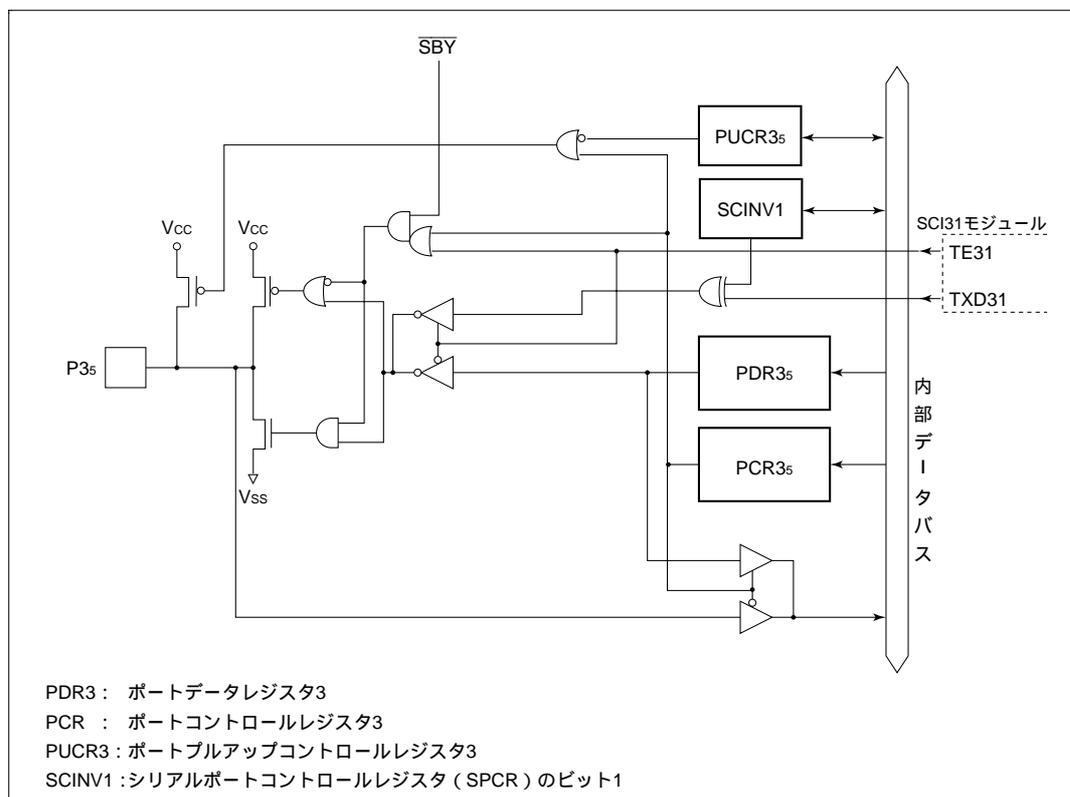


図 C.3 (b) ポート3 ブロック図 (P3_s端子)

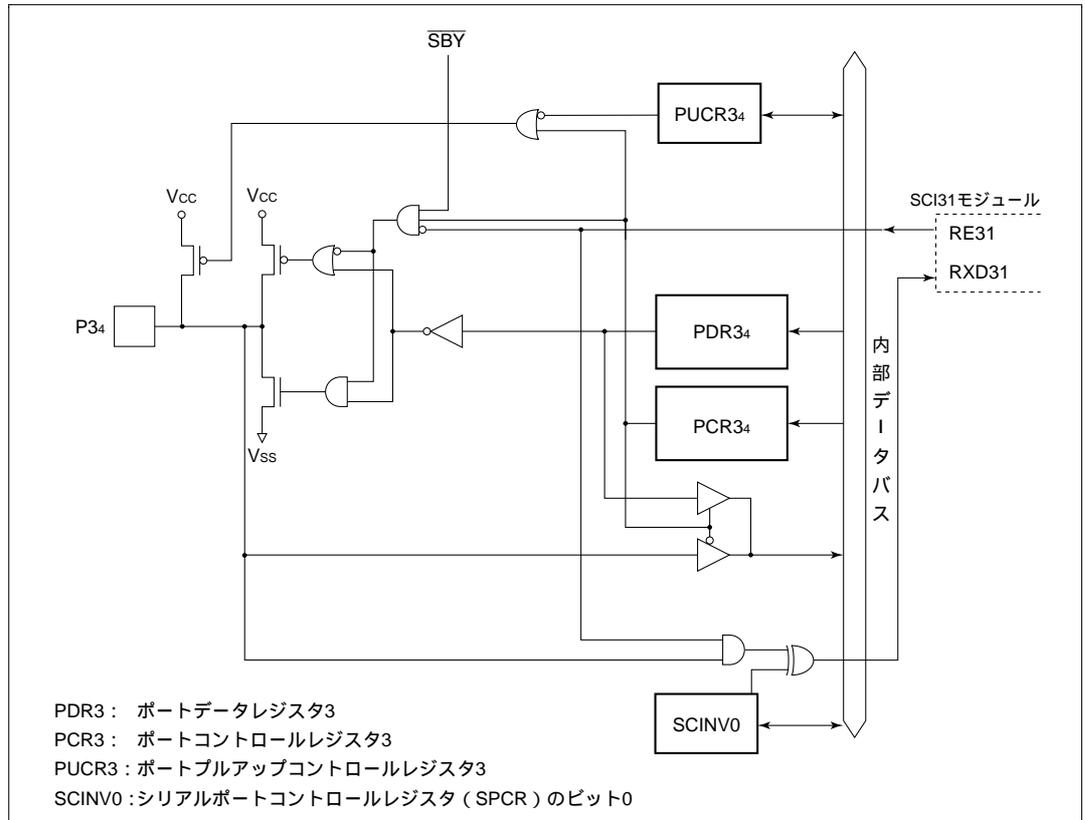


図 C.3 (c) ポート 3 ブロック図 (P₃₄端子)

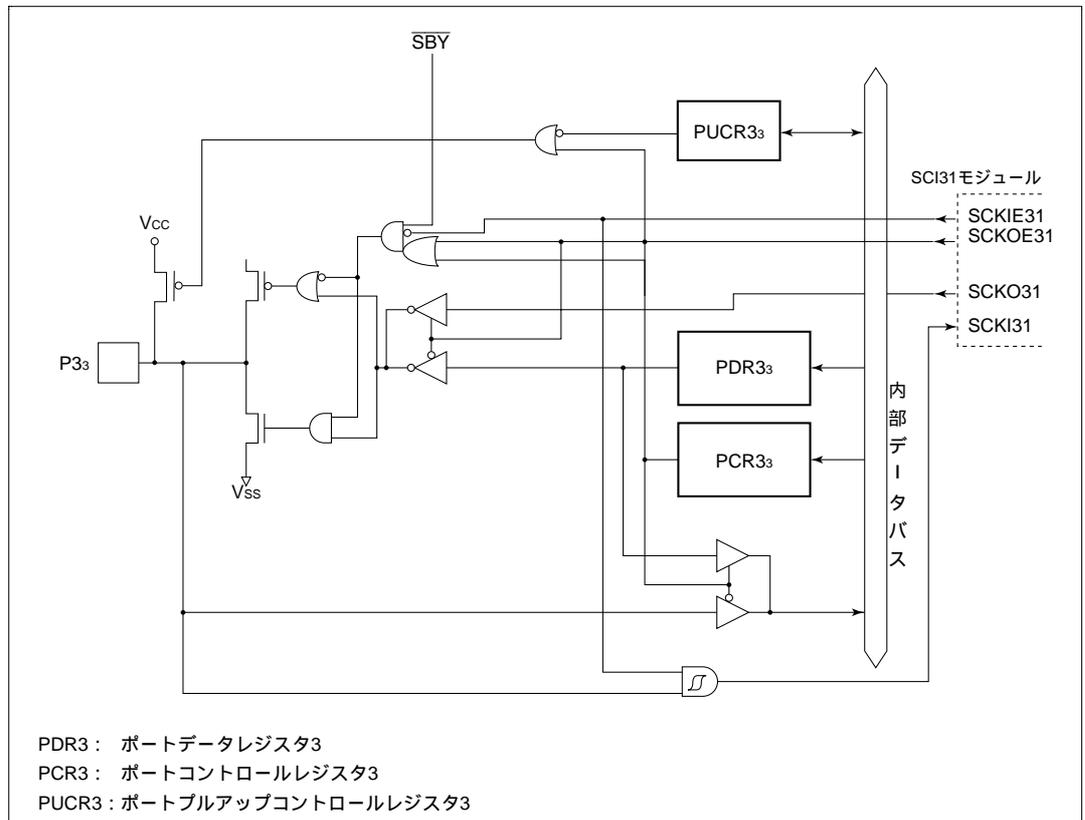


図 C.3 (d) ポート 3 ブロック図 (P₃₃端子)

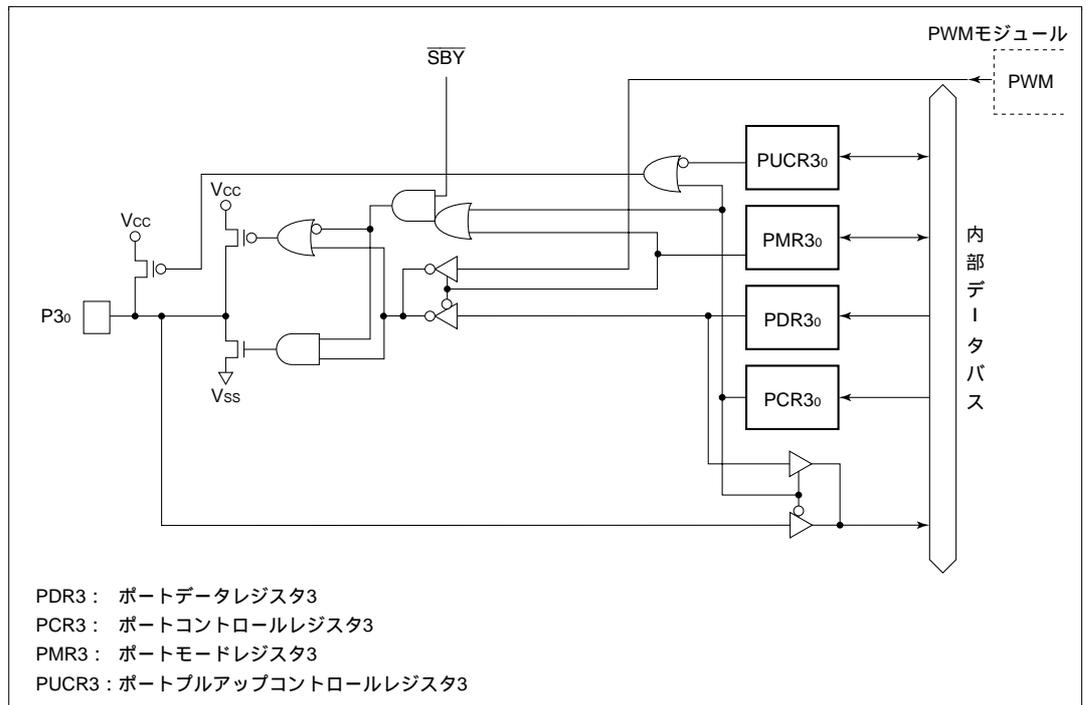


図 C.3 (g) ポート3 ブロック図 (P3₀端子)

C.4 ポート4ブロック図

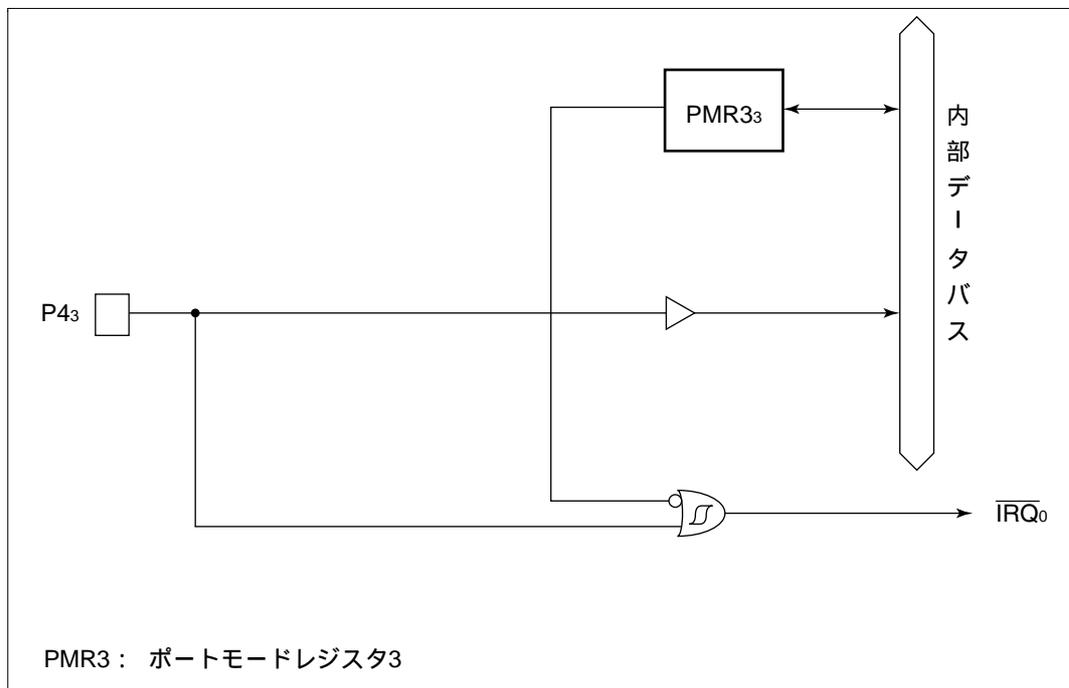


図 C.4 (a) ポート4 ブロック図 (P4₃端子)

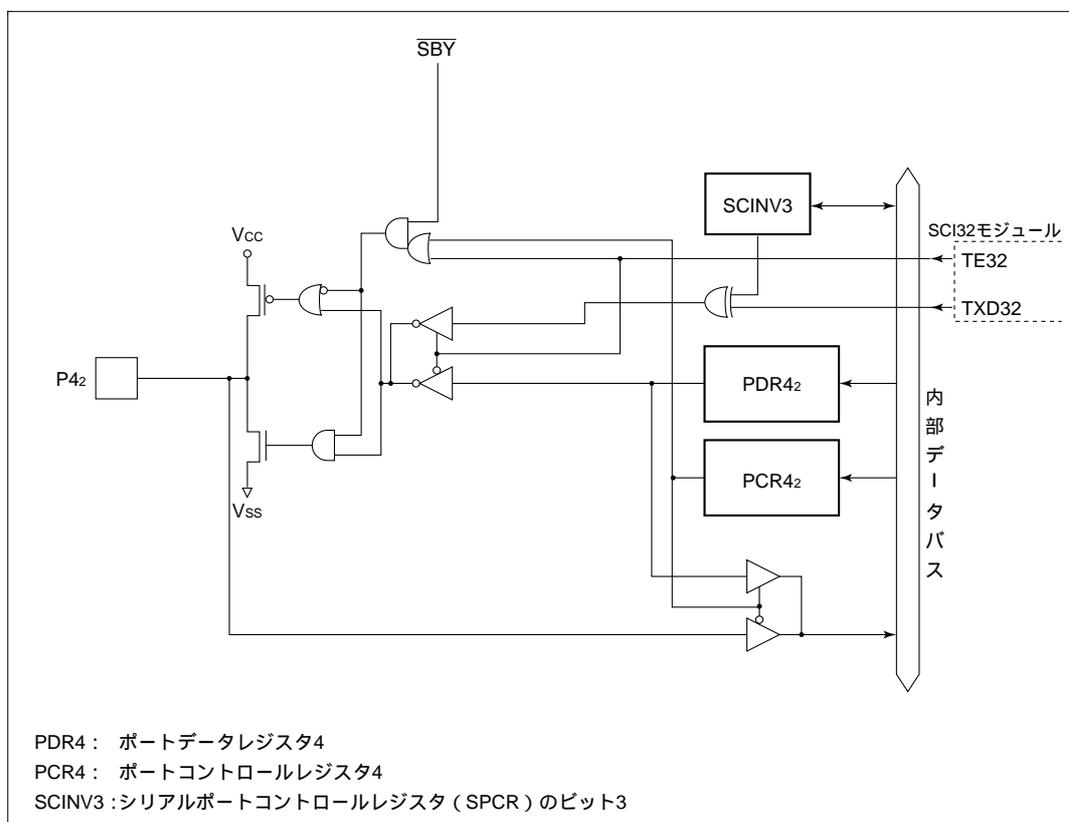


図 C.4 (b) ポート4 ブロック図 (P4₂端子)

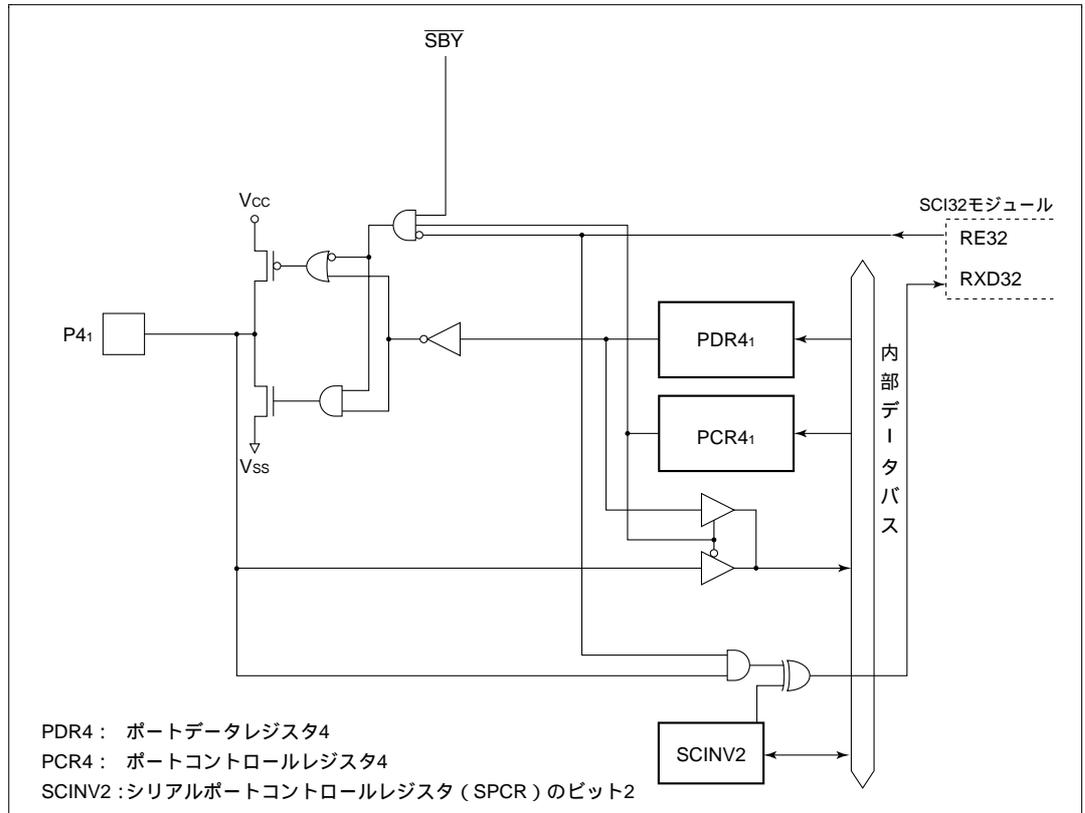


図 C.4 (c) ポート4 ブロック図 (P4₁端子)

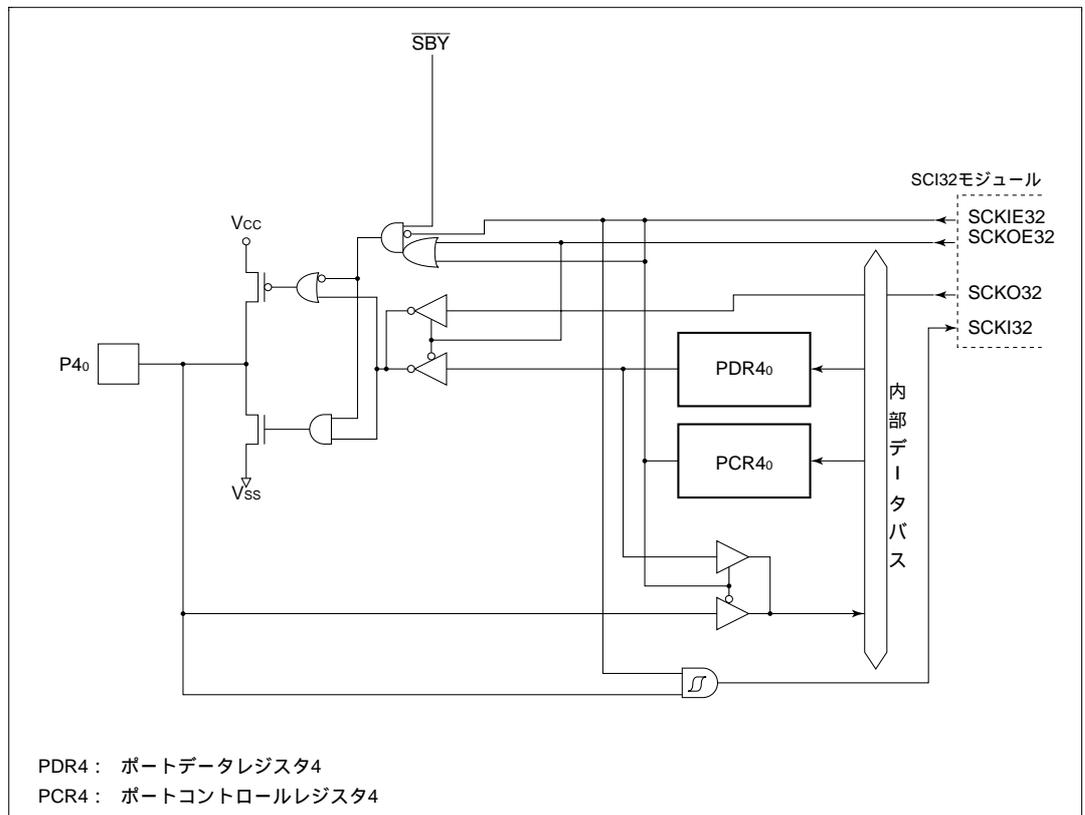


図 C.4 (d) ポート4 ブロック図 (P4₀端子)

C.5 ポート5ブロック図

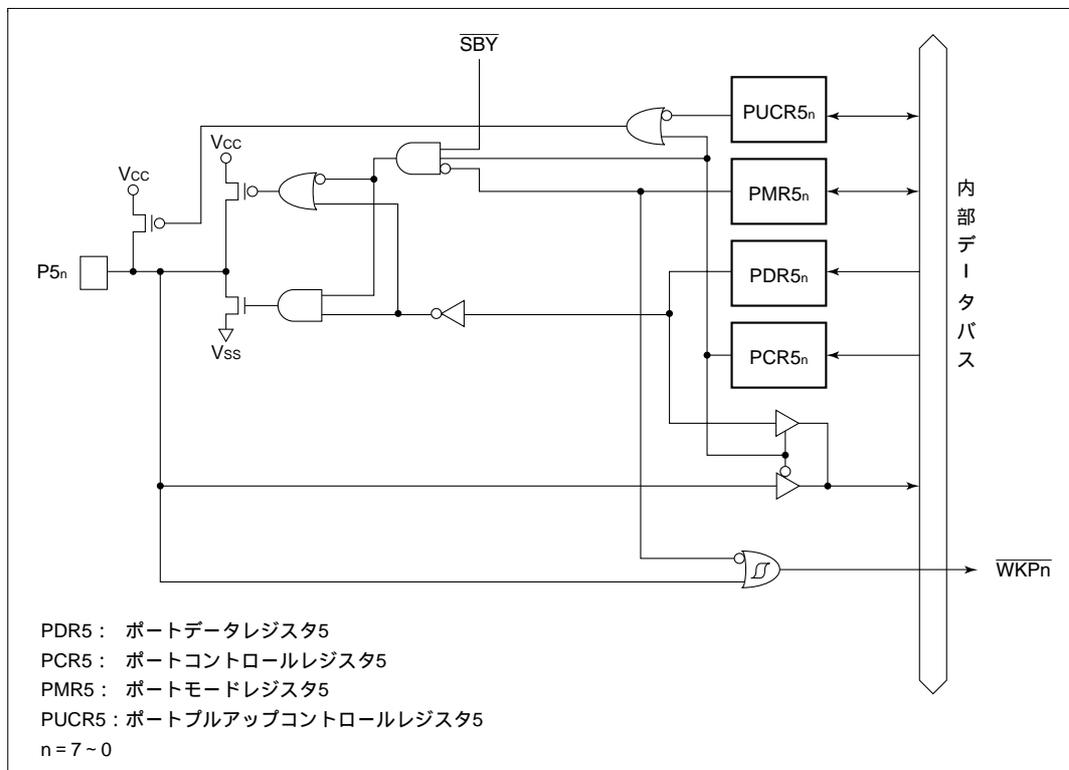


図 C.5 ポート5ブロック図

C.6 ポート6ブロック図

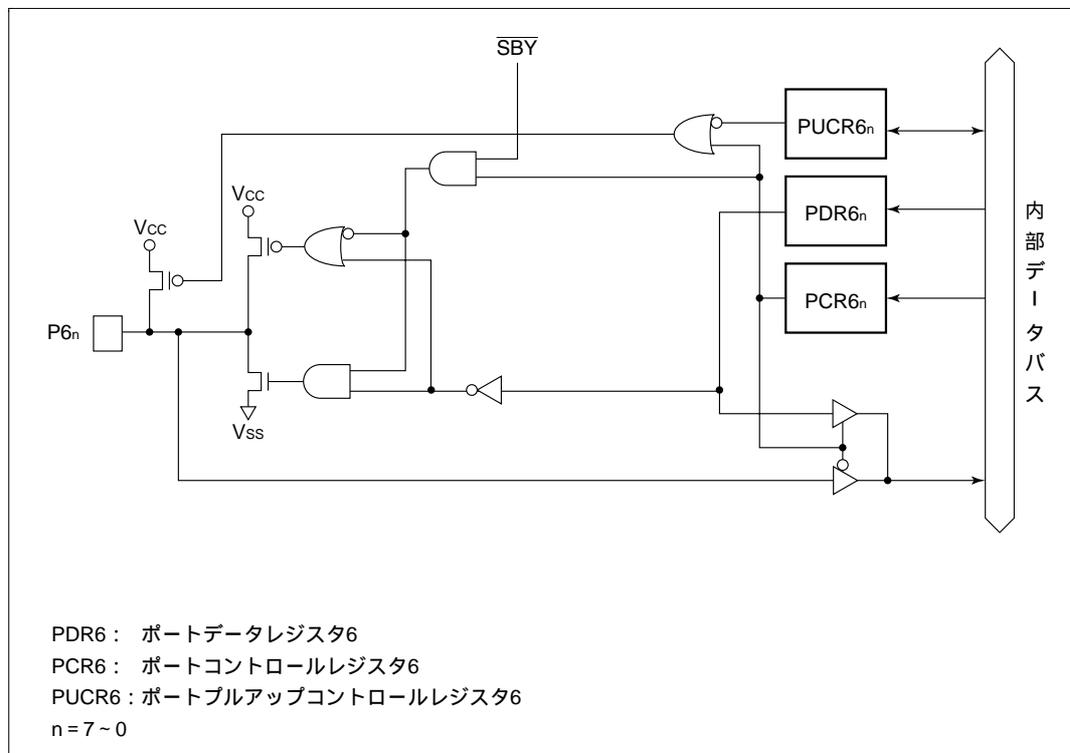


図 C.6 ポート6ブロック図

C.8 ポート8ブロック図

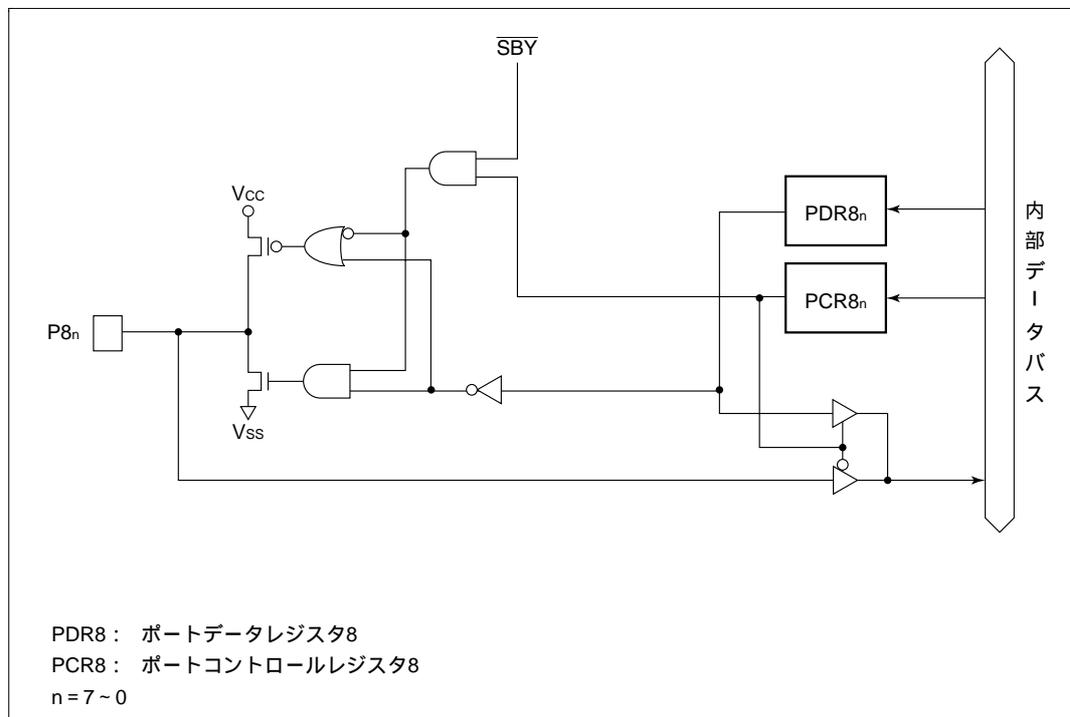


図 C.8 ポート8ブロック図

C.9 ポート9ブロック図

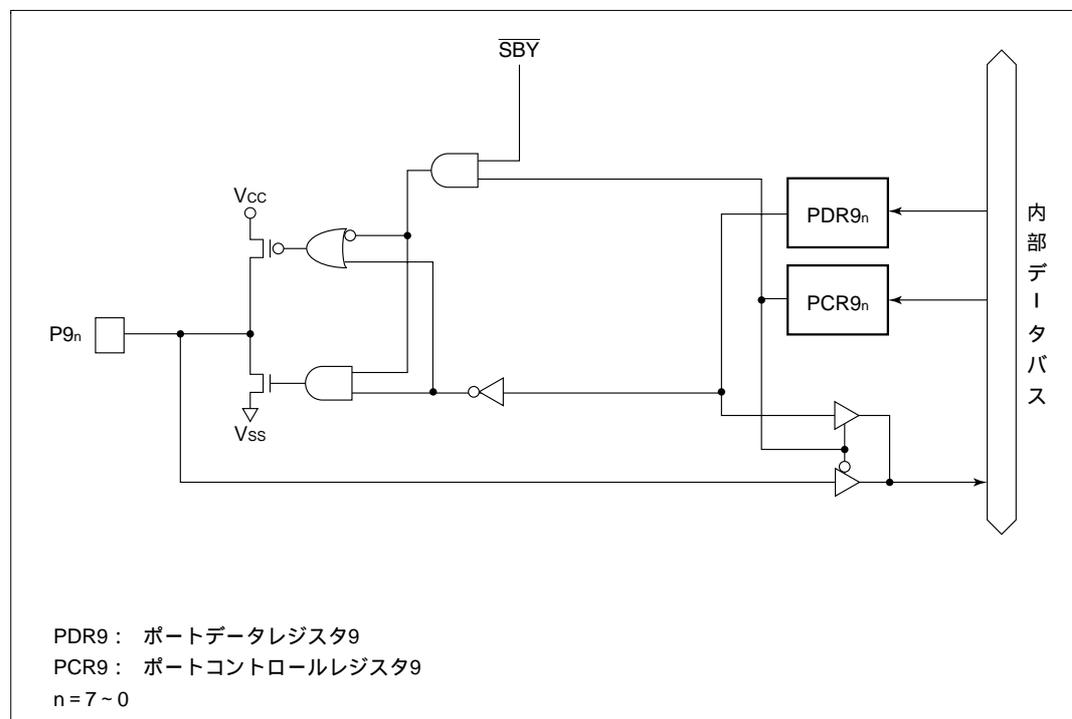


図 C.9 ポート9ブロック図

C.10 ポートAブロック図

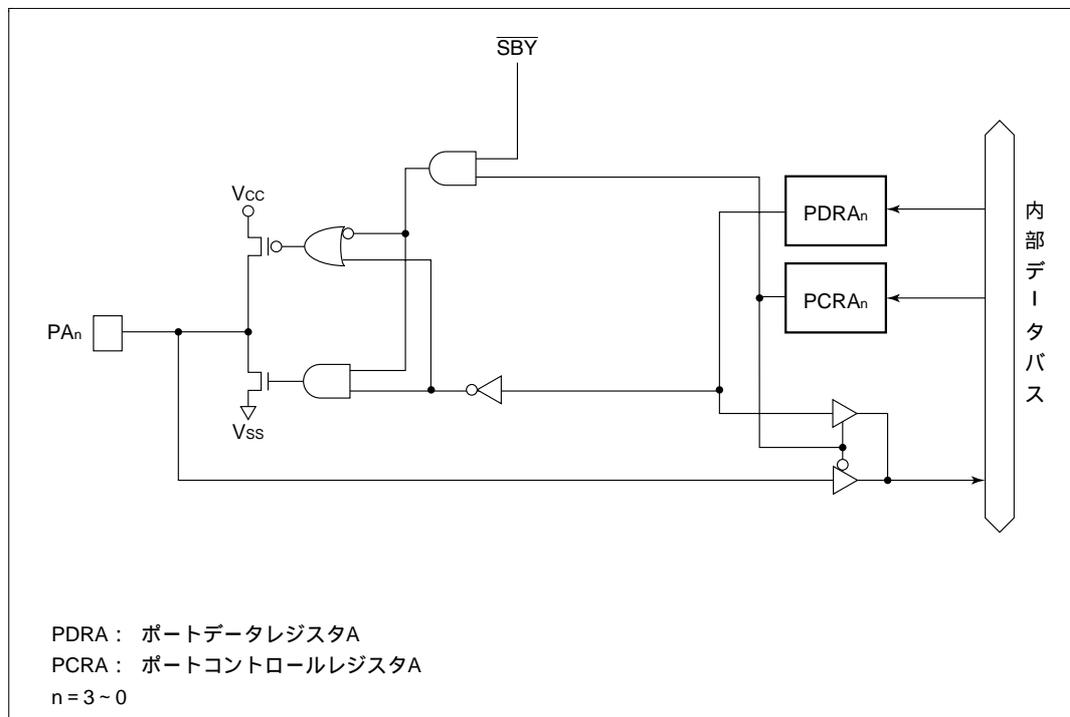


図 C.10 ポートAブロック図

C.11 ポートBブロック図

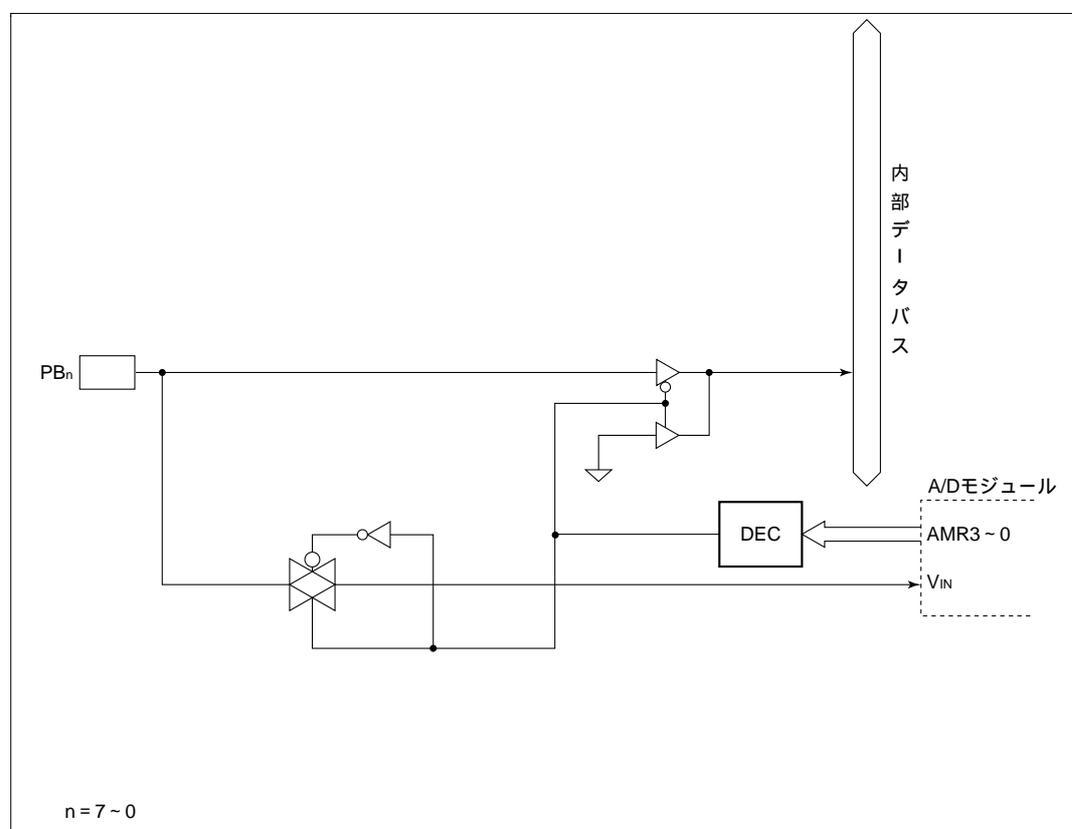


図 C.11 ポートBブロック図

C.12 ポートCブロック図

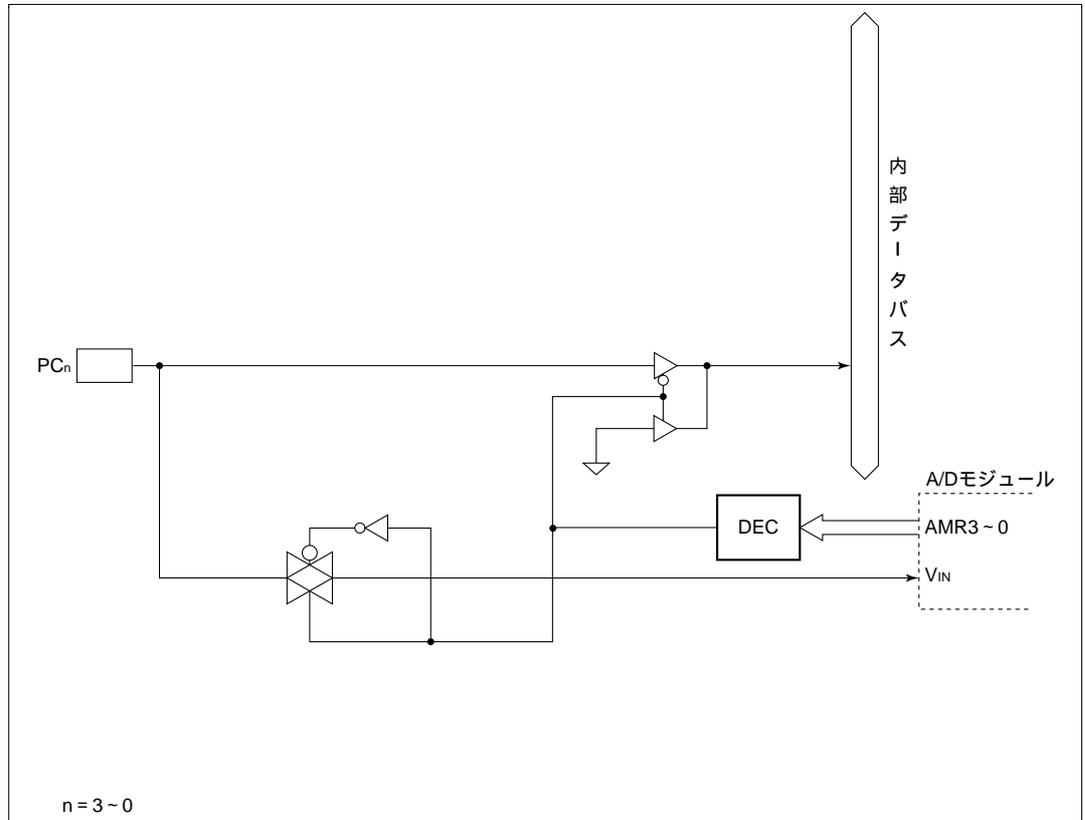


図 C.12 ポートCブロック図

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 ₇ ~P1 ₀	ハイインピ -ダンス	保持	保持	ハイインピ -ダンス*1	保持	動作	動作
P2 ₇ ~P2 ₀	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
P3 ₇ ~P3 ₀	ハイインピ -ダンス*2	保持	保持	ハイインピ -ダンス*1	保持	動作	動作
P4 ₃ ~P4 ₀	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 ₇ ~P5 ₀	ハイインピ -ダンス	保持	保持	ハイインピ -ダンス*1	保持	動作	動作
P6 ₇ ~P6 ₀	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 ₇ ~P7 ₀	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 ₇ ~P8 ₀	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
P9 ₇ ~P9 ₀	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 ₃ ~PA ₀	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
PB ₇ ~PB ₀	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
PC ₃ ~PC ₀	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】 *1 ブルアップ MOS が ON 状態では"High"レベル出力となります。

*2 P3₂端子のみリセット出力となります。

E. ROM 発注手順

E.1 ROM 書き換え品開発の流れ（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 書き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。

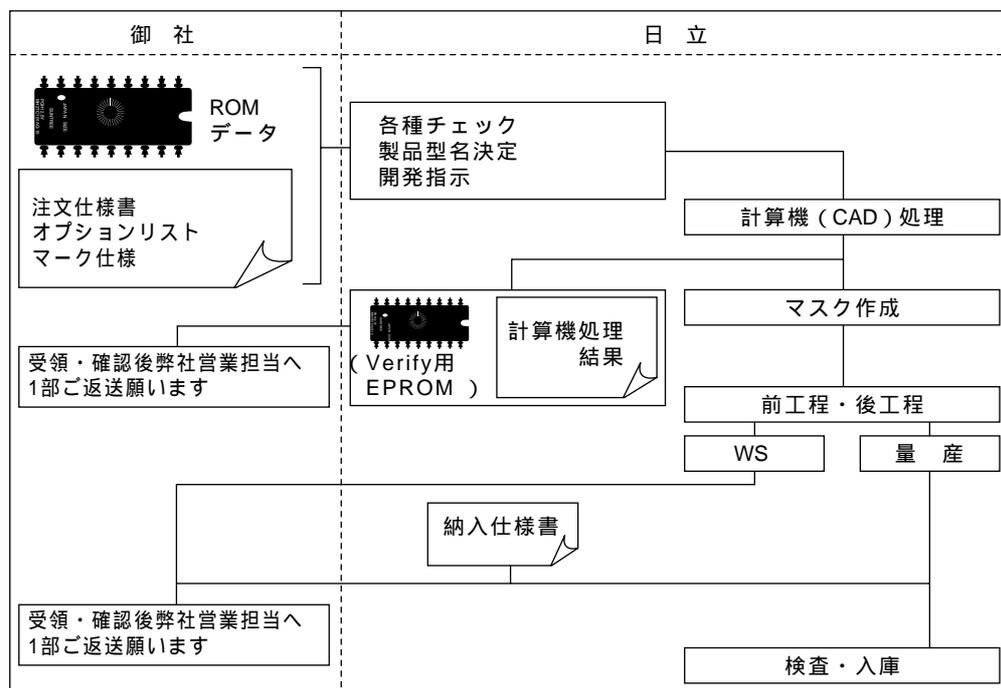


図 E.1 ROM 書き換え品開発の流れ

表 E.1 ROM 発注時に必要な提出物

発注媒体	EPROM または ZTAT [®] 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®マイコンで提出してください。なお、EPROMまたはZTAT®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. 製品型名一覧

表 F.1 型名一覧 (1)

	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)
H8/3887 シリーズ	H8/3882	マスク	HD6433882H	HD6433882 (***) H	100 ピン QFP (FP-100B)
		ROM 版	HD6433882F	HD6433882 (***) F	100 ピン QFP (FP-100A)
			HD6433882X	HD6433882 (***) X	100 ピン TQFP (TFP-100B)
			HD6433882W	HD6433882 (***) W	100 ピン TQFP (TFP-100G)
	H8/3883	マスク	HD6433883H	HD6433883 (***) H	100 ピン QFP (FP-100B)
		ROM 版	HD6433883F	HD6433883 (***) F	100 ピン QFP (FP-100A)
			HD6433883X	HD6433883 (***) X	100 ピン TQFP (TFP-100B)
			HD6433883W	HD6433883 (***) W	100 ピン TQFP (TFP-100G)
	H8/3884	マスク	HD6433884H	HD6433884 (***) H	100 ピン QFP (FP-100B)
		ROM 版	HD6433884F	HD6433884 (***) F	100 ピン QFP (FP-100A)
			HD6433884X	HD6433884 (***) X	100 ピン TQFP (TFP-100B)
			HD6433884W	HD6433884 (***) W	100 ピン TQFP (TFP-100G)
	H8/3885	マスク	HD6433885H	HD6433885 (***) H	100 ピン QFP (FP-100B)
		ROM 版	HD6433885F	HD6433885 (***) F	100 ピン QFP (FP-100A)
			HD6433885X	HD6433885 (***) X	100 ピン TQFP (TFP-100B)
			HD6433885W	HD6433885 (***) W	100 ピン TQFP (TFP-100G)
	H8/3886	マスク	HD6433886H	HD6433886 (***) H	100 ピン QFP (FP-100B)
		ROM 版	HD6433886F	HD6433886 (***) F	100 ピン QFP (FP-100A)
			HD6433886X	HD6433886 (***) X	100 ピン TQFP (TFP-100B)
			HD6433886W	HD6433886 (***) W	100 ピン TQFP (TFP-100G)
H8/3887	マスク		HD6433887H	HD6433887 (***) H	100 ピン QFP (FP-100B)
			HD6433887F	HD6433887 (***) F	100 ピン QFP (FP-100A)
			HD6433887X	HD6433887 (***) X	100 ピン TQFP (TFP-100B)
			HD6433887W	HD6433887 (***) W	100 ピン TQFP (TFP-100G)
	ZTAT 版		HD6473887H	HD6473887H	100 ピン QFP (FP-100B)
			HD6473887F	HD6473887F	100 ピン QFP (FP-100A)
			HD6473887X	HD6473887X	100 ピン TQFP (TFP-100B)
			HD6473887W	HD6473887W	100 ピン TQFP (TFP-100G)

表 F.1 型名一覧 (2)

	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)
H8/3847 シリーズ	H8/3842	マスク ROM 版	HD6433842H	HD6433842 (***) H	100 ピン QFP (FP-100B)
			HD6433842F	HD6433842 (***) F	100 ピン QFP (FP-100A)
			HD6433842X	HD6433842 (***) X	100 ピン TQFP (TFP-100B)
			HD6433842W	HD6433842 (***) W	100 ピン TQFP (TFP-100G)
	H8/3843	マスク ROM 版	HD6433843H	HD6433843 (***) H	100 ピン QFP (FP-100B)
			HD6433843F	HD6433843 (***) F	100 ピン QFP (FP-100A)
			HD6433843X	HD6433843 (***) X	100 ピン TQFP (TFP-100B)
			HD6433843W	HD6433843 (***) W	100 ピン TQFP (TFP-100G)
	H8/3844	マスク ROM 版	HD6433844H	HD6433844 (***) H	100 ピン QFP (FP-100B)
			HD6433844F	HD6433844 (***) F	100 ピン QFP (FP-100A)
			HD6433844X	HD6433844 (***) X	100 ピン TQFP (TFP-100B)
			HD6433844W	HD6433844 (***) W	100 ピン TQFP (TFP-100G)
	H8/3845	マスク ROM 版	HD6433845H	HD6433845 (***) H	100 ピン QFP (FP-100B)
			HD6433845F	HD6433845 (***) F	100 ピン QFP (FP-100A)
			HD6433845X	HD6433845 (***) X	100 ピン TQFP (TFP-100B)
			HD6433845W	HD6433845 (***) W	100 ピン TQFP (TFP-100G)
	H8/3846	マスク ROM 版	HD6433846H	HD6433846 (***) H	100 ピン QFP (FP-100B)
			HD6433846F	HD6433846 (***) F	100 ピン QFP (FP-100A)
			HD6433846X	HD6433846 (***) X	100 ピン TQFP (TFP-100B)
			HD6433846W	HD6433846 (***) W	100 ピン TQFP (TFP-100G)
	H8/3847	マスク ROM 版	HD6433847H	HD6433847 (***) H	100 ピン QFP (FP-100B)
			HD6433847F	HD6433847 (***) F	100 ピン QFP (FP-100A)
			HD6433847X	HD6433847 (***) X	100 ピン TQFP (TFP-100B)
			HD6433847W	HD6433847 (***) W	100 ピン TQFP (TFP-100G)
ZTAT 版		HD6473847H	HD6473847H	100 ピン QFP (FP-100B)	
		HD6473847F	HD6473847F	100 ピン QFP (FP-100A)	
		HD6473847X	HD6473847X	100 ピン TQFP (TFP-100B)	
		HD6473847W	HD6473847W	100 ピン TQFP (TFP-100G)	

【注】 マスク ROM 版の (***) は ROM コードです。

G. 外形寸法図

H8/3887、H8/3847 シリーズの外形寸法図 FP-100A を図 G.1、FP-100B を図 G.2、TFP-100B を図 G.3、TFP-100G を図 G.4 に示します。

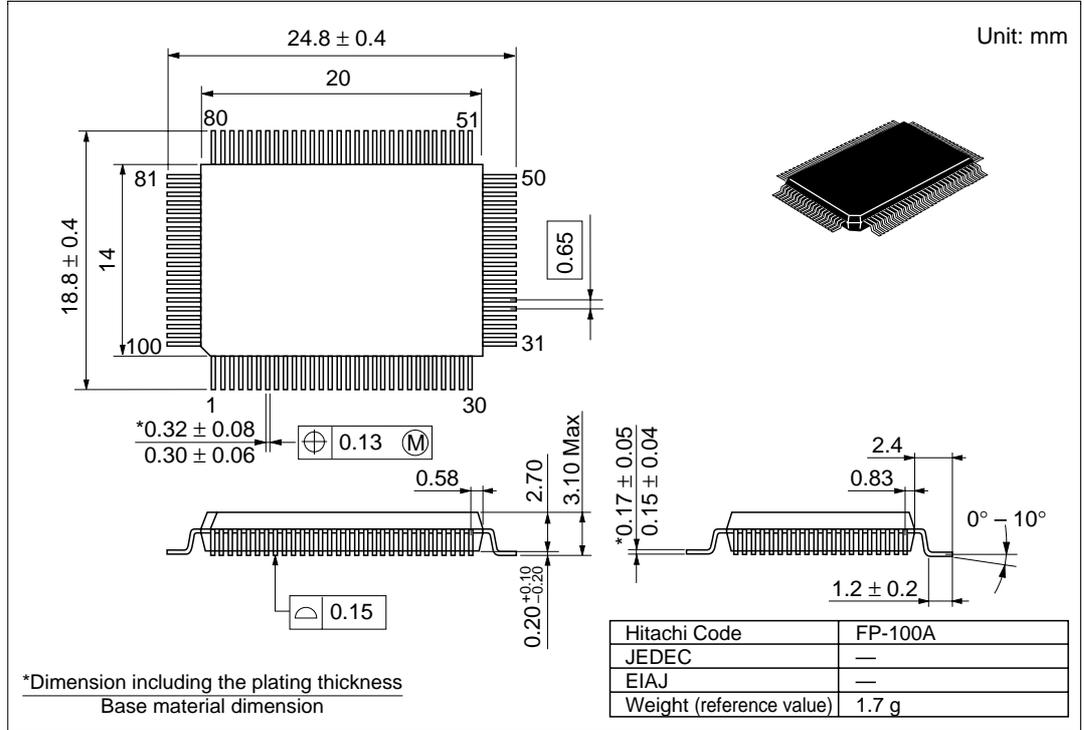


図 G.1 外形寸法図〔FP-100A〕

H8/3887 シリーズ, H8/3847 シリーズ
ハードウェアマニュアル

発行年月 平成 10 年 3 月 第 1 版

平成 11 年 2 月 第 2 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 1998

H8/3887 シリーズ、H8/3847 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

ADJ-602-172A (H)

— 日立マイクロコンピュータ技術情報 —

〒100-0004

東京都千代田区大手町2丁目6番2号

(日本ビル)

TEL (03)5201-5028 (ダイヤルイン)

株式会社 日立製作所 半導体グループ

題 目	タイマFの使用上の注意事項について		発行番号	TN-H8*-171A		
			分類	1. 仕様変更 ② ドキュメント訂正追加等 ③ 使用上の注意事項		
適 用 製 品	H8/3887シリーズ, H8/3847シリーズ, H8/3867シリーズ, H8/3827シリーズ, H8/3847Rシリーズ, H8/3827Rシリーズ, H8/3802シリーズ	対象ロット等	関 連 資 料	H8/3887シリーズ, H8/3847シリーズ ハードウェアマニュアル第2版 (ADJ-602-172A) H8/3867シリーズ, H8/3827シリーズ ハードウェアマニュアル第2版 (ADJ-602-159B) H8/3847Rシリーズハードウェアマニュアル第1版 (ADJ-602-227) H8/3827Rシリーズハードウェアマニュアル第1版 (ADJ-602-226) H8/3802シリーズハードウェアマニュアル第1版 (ADJ-602-230)	Rev.	有効期限
		全ロット			左記	永年
<p>日立シングルチップマイクロコンピュータ H8/3887/47/67/27/47R/27Rシリーズ、 H8/3802シリーズハードウェアマニュアルにおきまして、タイマFの使用上の注意事項 を追加致します。ご理解、ご了承頂きたく、お願い申し上げます。</p> <p style="text-align: center;">- 記 -</p> <p>< 追加内容 ></p> <p>9.4 タイマF --- (H8/3887/47シリーズは、9.4の214ページ。H8/3867/27/27Rシリーズ は、9.4の212ページ。H8/3847Rシリーズは、9.4の216ページ。 H8/3802シリーズは、9.3の189ページ。)</p> <p>9.4.5 使用上の注意事項 --- (H8/3887/47シリーズは、9.4.5の231ページ。 H8/3867/27/27Rシリーズは、9.4.5の229ページ。 H8/3847Rシリーズは、9.4.5の233ページ。H8/3802 シリーズは、9.3.5の205ページ。)</p> <p>(3) タイマFH、タイマFL割込み要求フラグ(IRRTFH、IRRTFL)、タイマオーバフロー フラグH、L(OVFH、OVFL)及びコンペアマッチフラグH、L(CMFH、CMFL)のクリア</p> <p>内部クロックとして $w/4$ を選択した場合、「割込み要因の発生を示す信号」 の制御は w で行われるため、この信号は w の幅出力されます。また、「オーバ フローしたことを示す信号」及び「コンペアマッチしたことを示す信号」の制御は、 w の2周期分の信号で行われるため、これらの信号は w の2周期の幅出力され ます。(図1)</p> <p>アクティブ(高速、中速)モードにおいて、「割込み要因の発生を示す信号」が有効 な間は、割込み要求フラグをクリアしても、すぐにまた割込み要求フラグがセット されます。(図1 -)</p> <p>また、「オーバフローしたことを示す信号」及び「コンペアマッチしたことを示す 信号」が有効な間は、タイマオーバフローフラグ、コンペアマッチフラグをクリア 出来ません。</p>						

割り込み要求フラグをクリアしても、すぐにまた割り込み要求フラグがセットされるため、1回のタイマFH、タイマFL割り込みに対し割り込み処理が複数回行なわれることがあります。(図1 -)

そのため、アクティブ(高速、中速)モードにおいて、割り込み要求フラグを確実にクリアするためには、以下の(1)の計算式で計算される時間の後で、クリアを行う必要があります。また、タイマオーバフローフラグ、コンペアマッチフラグを確実にクリアするためには、以下の(1)の計算式で計算される時間の後で、タイマコントロールステータスレジスタF(TCSR F)をリードした後、クリアを行う必要があります。

(1)の計算式のS Tには使用されている命令の中で最も実行ステート数が長い命令の実行ステート数を代入して下さい。(MULXU、DIVXU命令を使用しない場合はRTE命令の10ステート、ULXU、DIVXU命令を使用する場合は14ステート)

サブアクティブモードでは、割り込み要求フラグ、タイマオーバフローフラグ、コンペアマッチフラグのクリアに対する制限はありません。

「割り込み要因の発生を示す信号」が有効な期間

$$= w \text{ の } 1 \text{ 周期} + \text{実行中の命令の実行完了待ち時間} + (\text{割り込みを } \text{ で同期化する時間})$$

$$= 1 / w + S T \times (1 / \text{)} + (2 / \text{)} \quad (\text{秒}) \quad \dots\dots\dots (1)$$

S T : 実行中の命令の実行ステート数

具体的には下記の方法がありますが、時間の有効活用の面から(方法1)を推奨致します。

(方法1) 割り込み処理ルーチン内で割り込みを禁止(IENFH、IENFLを"0"に設定)し、通常処理に復帰した後で(1)の計算式の時間以上を待って、割り込み要求フラグ(IRRTFH、IRRTFL)をクリアし、タイマコントロールステータスレジスタF(TCSR F)をリードした後、タイマオーバフローフラグ(OVFH、OVFL)及びコンペアマッチフラグ(CMFH、CMFL)をクリアし、割り込みの許可(IENFH、IENFLを"1"に設定)を行います。

(方法2) 割り込み処理ルーチンの処理時間を(1)の計算式の時間以上にし、割り込み処理ルーチンの最後で割り込み要求フラグ(IRRTFH、IRRTFL)をクリアし、タイマコントロールステータスレジスタF(TCSR F)をリードした後、タイマオーバフローフラグ(OVFH、OVFL)及びコンペアマッチフラグ(CMFH、CMFL)のクリアを行います。

以上の注意事項は、16ビットモード、8ビットモードとも同じです。

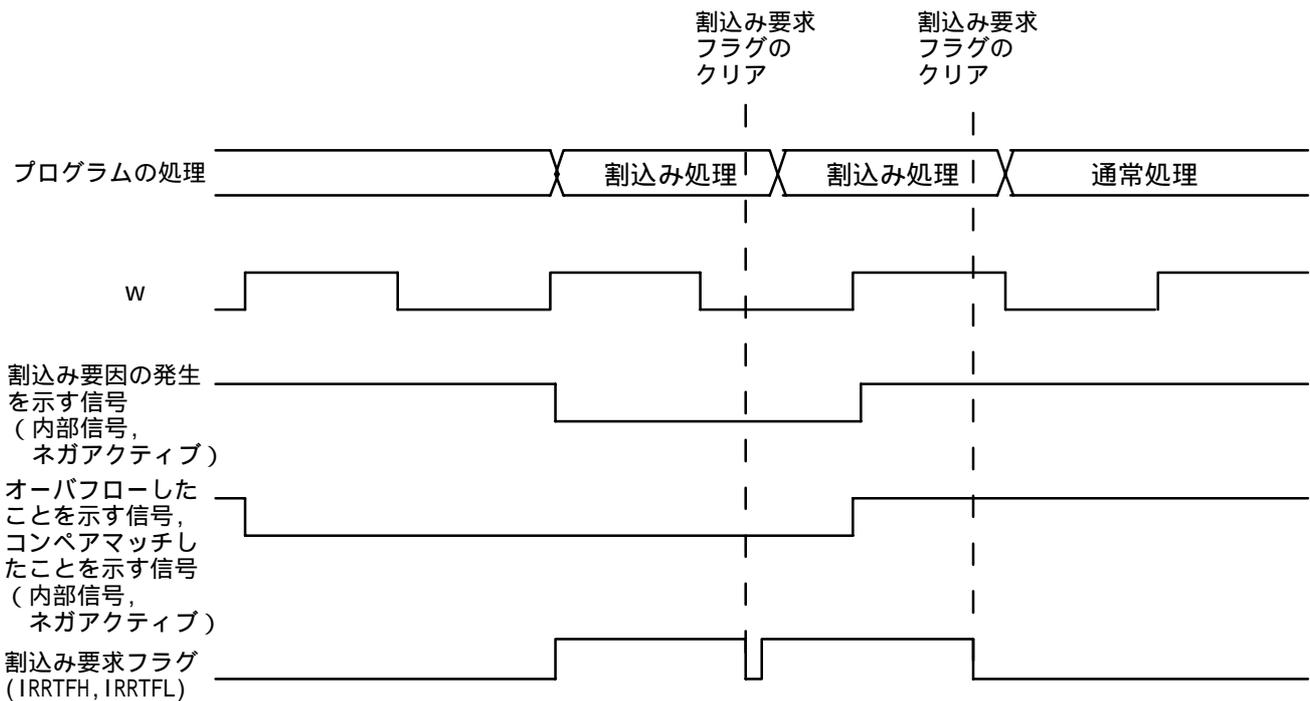


図1 割り込み要因の発生を示す信号が有効な時に割り込み要求フラグのクリアを行った場合

(4) タイマカウンタ(TCF)のリード/ライト

アクティブ(高速、中速)モードにおいて、内部クロックとして $w/4$ を選択した場合TCFに対するライトは出来ません。また、リードする場合、システムクロックと内部クロック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、TCFのリード値は最大 ± 1 の誤差が生じます。

アクティブ(高速、中速)モードでTCFをリード/ライトする必要がある場合は、内部クロックとして $w/4$ 以外を選択してからリード/ライトを行って下さい。

サブアクティブモードでは、内部クロックとして $w/4$ を選択した場合でもTCFは正常にリード/ライト出来ます。

- 以上 -