

改訂一覧は改訂箇所をまとめたものであり、
詳細については必ず本文の内容をご確認ください。

H8SX/1720S グループ

ユーザーズマニュアル ハードウェア編

ルネサス 32 ビット CISC マイクロコンピュータ
H8SX ファミリ／H8SX/1700 シリーズ

H8SX/1727S R5F61727S

H8SX/1725S R5F61725S

本資料に記載の全ての情報は本資料発行時点のものであり、ルネサス エレクトロニクスは、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 エレクトロニクスのホームページなどにより公開される最新情報をご確認ください。

ご注意書き

1. 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・システムの設計において、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て、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。
2. 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、誤りがな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。
3. 本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害に関し、当社は、何らの責任を負うものではありません。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。
4. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製等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。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。
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、産業用ロボット等
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、各種安全装置等
当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図し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損害が生じても、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の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。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、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、社会的損害等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、お客様の機器・システムとしての出荷保証を行ってください。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、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の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て、当社は、一切その責任を負いません。
9. 本資料に記載されている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・システムに使用することはできません。また、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に使用しないでください。当社製品または技術を輸出する場合は、「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。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にてご負担して頂きますのでご了承ください。
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禁じます。

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が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。

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本文を参照してください。なお、本マニュアルの本文と異なる記載がある場合は、本文の記載が優先するものとします。

1. 未使用端子の処理

【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

CMOS製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI周辺のノイズが印加され、LSI内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

2. 電源投入時の処置

【注意】電源投入時は、製品の状態は不定です。

電源投入時には、LSIの内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。

同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

3. リザーブアドレスのアクセス禁止

【注意】リザーブアドレスのアクセスを禁止します。

アドレス領域には、将来の機能拡張用に割り付けられているリザーブアドレスがあります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。

プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

5. 製品間の相違について

【注意】型名の異なる製品に変更する場合は、事前に問題ないことをご確認下さい。

同じグループのマイコンでも型名が違くと、内部メモリ、レイアウトパターンの相違などにより、特性が異なる場合があります。型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。

このマニュアルの使い方

1. 目的と対象者

このマニュアルは、本マイコンのハードウェア機能と電気的特性をユーザに理解していただくためのマニュアルです。本マイコンを用いた応用システムを設計するユーザを対象にしています。このマニュアルを使用するには、電気回路、論理回路、マイクロコンピュータに関する基本的な知識が必要です。

このマニュアルは、大きく分類すると、製品の概要、CPU、システム制御機能、周辺機能、電気的特性、使用上の注意で構成されています。

本マイコンは、注意事項を十分確認の上、使用してください。注意事項は、各章の本文中、各章の最後、注意事項の章に記載しています。

改訂記録は旧版の記載内容に対して訂正または追加した主な箇所をまとめたものです。改訂内容すべてを記録したものではありません。詳細は、このマニュアルの本文でご確認ください。

H8SX/1720S グループでは次のドキュメントを用意しています。ドキュメントは最新版を使用してください。最新版はルネサス エレクトロニクスのホームページに掲載されています。

ドキュメントの種類	記載内容	資料名	資料番号
データシート	ハードウェアの概要と電気的特性	—	—
ユーザーズマニュアル ハードウェア編	ハードウェアの仕様（ピン配置、メモリマップ、周辺機能の仕様、電気的特性、タイミング）と動作説明 ※周辺機能の使用方法はアプリケーションノートを参照してください。	H8SX/1720S グループ ユーザーズマニュアル ハードウェア編	本ユーザーズマニュアル
ユーザーズマニュアル ソフトウェア編	CPU 命令セットの説明	H8SX/ファミリ ソフトウェアマニュアル	RJJ09B0048
アプリケーションノート	周辺機能の使用手法、応用例 参考プログラム アセンブリ言語、C 言語によるプログラムの作成方法	ルネサス エレクトロニクスホームページに掲載されています。	
RENESAS TECHNICAL UPDATE	製品の仕様、ドキュメント等に関する速報		

2. 数や記号の表記

このマニュアルで使用するレジスタ名やビット名、数字や記号の表記の凡例を以下に説明します。

(1) 全体的な表記

本文中ではビットの説明をする場合、モジュールやレジスタとの関連を明確にするため、ビット名を「モジュール名、レジスタ名、ビット名」または「レジスタ名、ビット名」と表記している場合があります。

(2) レジスタの表記

同一または類似した機能が複数チャンネルに存在する場合に「レジスタ名_チャンネル番号」の表記を使用します。

(例) CMCSR_0 : コンペアマッチタイマのチャンネル0 (_0) のCMCSRレジスタを示します。

(3) 数字の表記

2進数はB'nnnn (明らかに2進数と判断できる場合はB'を省略)、16進数はH'nnnnまたは0xnnnn、

10進数はnnnnで表します。

(例) 2進数 : B'11または11

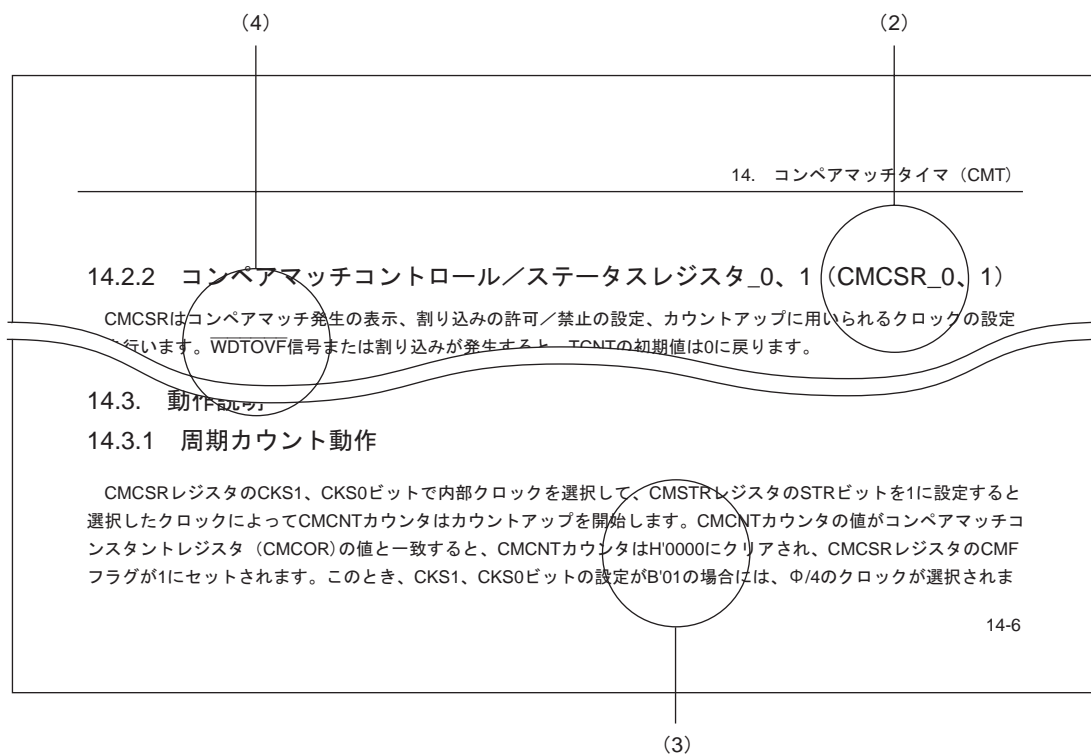
16進数 : H'EFA0または0xEFA0

10進数 : 1234

(4) ローアクティブの表記

ローアクティブの信号および端子には上線を付けて表記しています。

(例) WDTOVF

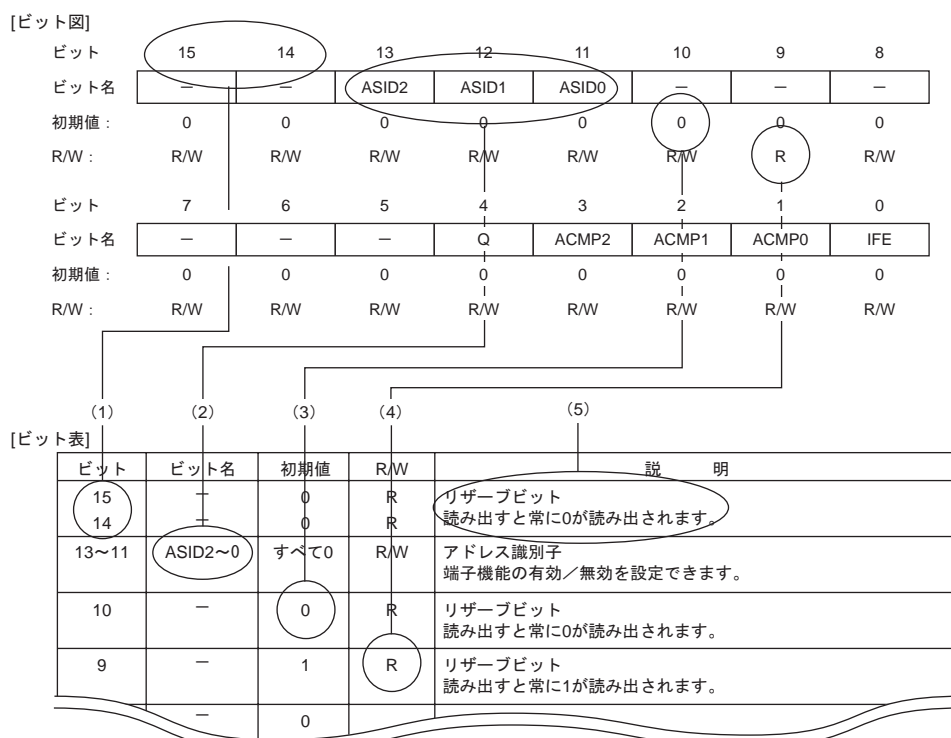


【注】 上記した図中のビット名や文章は例であり、マニュアルの内容とは関係がありません。

3. レジスタの表記

各レジスタの説明には、ビットの並びを示すビット図とビットに設定する内容を説明するビット表があります。

使用する記号、用語を以下に説明します。



【注】 上記した図中のビット名や文章は例であり、マニュアルの内容とは関係がありません。

(1) ビット

ビット番号を示します。

32ビットレジスタの場合は31から0の順に、16ビットレジスタの場合は15から0の順に示します。

(2) ビット名

ビット名またはフィールド名を示します。

フィールドでビット桁数を明示する必要があるときは、ASID[3:0]というように桁数の表記を追加します。

また、リザーブビットの場合は「-」と表記します。

ただし、タイムカウンタなどをはじめとして、ビット名の記載をせずに空白のままとしているものもあります。

(3) 初期値

各ビットのパワーオンリセット後の値を初期値として示します。

0 : 初期値は0であることを示します。

1 : 初期値は1であることを示します。

- : 初期値は不定であることを示します。

(4) R/W

各ビットが読み出し可能か、書き込み可能か、または書き込みも読み出しも不可であることを示します。

使用する表記を以下に説明します。

R/W : 読み出しおよび書き込みが可能なビットまたはフィールドです。

R(W) : 読み出しおよび書き込みが可能なビットまたはフィールドです。

ただし書き込みは、フラグをクリアするための書き込みのみ可能です。

R : 読み出しが可能なビットまたはフィールドです。

リザーブビットはすべて「R」と表記します。書き込む必要がある場合は、ビット表で指定された値を書き込んでください。

W : 書き込みが可能なビットまたはフィールドです。

(5) 説明

ビットの機能について説明しています。

4. 略語および略称の説明

以下に本書内で使用されている略語または略称を示します。

- 本製品固有の略語または略称

略称	英語名	日本語名
BSC	Bus Controller	バスコントローラ
CPG	Clock Pulse Generator	クロック発振器
DTC	Data Transfer Controller	データトランスファコントローラ
INTC	Interrupt Controller	割り込みコントローラ
PPG	Programmable Pulse Generator	プログラマブルパルスジェネレータ
SCI	Serial Communication Interface	シリアルコミュニケーションインタフェース
TMR	8-Bit Timer	8ビットタイマ
TPU	16-Bit Timer Pulse Unit	16ビットタイマパルスユニット
RCAN	Controller Area Network	コントローラエリアネットワーク
RSPI	Renesas Serial Peripheral Interface	ルネサスシリアルペリフェラルインタフェース
WDT	Watchdog Timer	ウォッチドッグタイマ

- その他の略語または略称

略語／略称	英語名	日本語名
ACIA	Asynchronous Communication Interface Adapter	調歩同期式通信アダプタ
bps	bits per second	転送速度を表す単位
CRC	Cyclic Redundancy Check	周期的冗長検査
DMA	Direct Memory Access	ダイレクトメモリアクセス
DMAC	Direct Memory Access Controller	ダイレクトメモリアクセスコントローラ
GSM	Global System for Mobile Communications	ジーエスエム
Hi-Z	High Impedance	ハイインピーダンス
IEBus	Inter Equipment bus	—
I/O	Input/Output	入出力
IrDA	Infrared Data Association	赤外線データアソシエーション
LSB	Least Significant Bit	最下位ビット
MSB	Most Significant Bit	最上位ビット
NC	Non-Connection	未接続端子
PLL	Phase Locked Loop	位相ロックループ
PWM	Pulse Width Modulation	パルス幅変調
SFR	Special Function Registers	周辺回路制御用レジスタ群
SIM	Subscriber Identity Module	ISO-7816 規定の通信方式
UART	Universal Asynchronous Receiver/Transmitter	非同期シリアルインタフェース
VCO	Voltage Controlled Oscillator	電圧制御発振器

すべての商標および登録商標は、それぞれの所有者に帰属します。

目次

1. 概要.....	1-1
1.1 特長	1-1
1.2 内部ブロック図	1-2
1.3 端子説明	1-3
1.3.1 ピン配置図	1-3
1.3.2 動作モード別ピン配置一覧.....	1-4
1.3.3 端子機能	1-7
2. CPU	2-1
2.1 特長	2-1
2.2 CPU動作モード.....	2-3
2.2.1 ノーマルモード	2-3
2.2.2 ミドルモード	2-5
2.2.3 アドバンストモード	2-6
2.2.4 マキシマムモード	2-7
2.3 命令フェッチ	2-8
2.4 アドレス空間	2-9
2.5 レジスタ構成	2-9
2.5.1 汎用レジスタ	2-11
2.5.2 プログラムカウンタ (PC)	2-12
2.5.3 コンディションコードレジスタ (CCR)	2-12
2.5.4 エクステンドレジスタ (EXR)	2-13
2.5.5 ベクタベースレジスタ (VBR)	2-14
2.5.6 ショートアドレスベースレジスタ (SBR)	2-14
2.5.7 積和レジスタ (MAC)	2-14
2.5.8 CPU 内部レジスタの初期値.....	2-14
2.6 データ形式	2-14
2.6.1 汎用レジスタのデータ形式.....	2-15
2.6.2 メモリ上でのデータ形式	2-16
2.7 命令セット	2-17
2.7.1 命令とアドレッシングモードの組み合わせ.....	2-19
2.7.2 命令の機能別一覧	2-23
2.7.3 命令の基本フォーマット	2-32
2.8 アドレッシングモードと実効アドレスの計算方法.....	2-33

2.8.1	レジスタ直接 Rn.....	2-33
2.8.2	レジスタ間接 @ERn.....	2-34
2.8.3	ディスプレイメント付レジスタ間接 @ (d:2,ERn) /@ (d:16,ERn) /@ (d:32,ERn)	2-34
2.8.4	ディスプレイメント付インデックスレジスタ間接 @(d:16,RnL.B)/ @(d:32,RnL.B)/@(d:16,Rn.W)/@(d:32,Rn.W)/@(d:16,ERn.L)/@(d:32,ERn.L).....	2-34
2.8.5	プリ/ポストインクリメント/デクリメントレジスタ間接 @ERn+/@-ERn/@+ERn/@ERn-.....	2-34
2.8.6	絶対アドレス @aa:8/@aa:16/@aa:24/@aa:32.....	2-35
2.8.7	イミディエイト #xx	2-36
2.8.8	プログラムカウンタ相対 @ (d:8, PC) /@ (d:16, PC)	2-36
2.8.9	プログラムカウンタインデックス相対 @ (RnL.B, PC) /@ (Rn.W, PC) /@ (ERn.L, PC)	2-37
2.8.10	メモリ間接 @@aa:8	2-37
2.8.11	拡張メモリ間接 @@vec:7	2-38
2.8.12	実効アドレスの計算方法	2-38
2.8.13	MOVA 命令	2-40
2.9	処理状態	2-41
3.	MCU 動作モード.....	3-1
3.1	動作モードの選択	3-1
3.2	レジスタの説明	3-2
3.2.1	モードコントロールレジスタ (MDCR)	3-2
3.2.2	システムコントロールレジスタ (SYSCR0)	3-3
3.2.3	システムコントロールレジスタ (SYSCR1)	3-4
3.3	動作モードの説明	3-5
3.3.1	モード 1	3-5
3.3.2	モード 2	3-5
3.3.3	モード 3	3-5
3.3.4	アドレスマップ	3-6
4.	例外処理	4-1
4.1	例外処理の種類と優先度	4-1
4.2	例外処理要因とベクタテーブル.....	4-2
4.3	リセット	4-4
4.3.1	リセット例外処理	4-4
4.3.2	リセット直後の割り込み	4-4
4.3.3	リセット解除後の内蔵周辺機能.....	4-4
4.4	トレース例外処理	4-6
4.5	アドレスエラー	4-7
4.5.1	アドレスエラー発生要因	4-7

4.5.2	アドレスエラー例外処理	4-8
4.6	割り込み	4-9
4.6.1	割り込み要因	4-9
4.6.2	割り込み例外処理	4-10
4.7	命令による例外処理	4-10
4.7.1	トラップ命令例外処理	4-10
4.7.2	不当命令例外処理	4-11
4.8	例外処理後のスタックの状態.....	4-12
4.9	使用上の注意事項	4-13
5.	割り込みコントローラ.....	5-1
5.1	特長	5-1
5.2	入出力端子	5-3
5.3	レジスタの説明	5-3
5.3.1	割り込みコントロールレジスタ (INTCR)	5-4
5.3.2	CPU プライオリティコントロールレジスタ (CPUPCR)	5-5
5.3.3	インタラプトプライオリティレジスタ A~P (IPRA~IPRP)	5-7
5.3.4	IRQ イネーブルレジスタ (IER)	5-9
5.3.5	IRQ センسコントロールレジスタ H、L (ISCRH、ISCRL)	5-10
5.3.6	IRQ ステータスレジスタ (ISR)	5-14
5.3.7	ソフトウェアスタンバイ解除 IRQ イネーブルレジスタ (SSIER)	5-15
5.4	割り込み要因	5-16
5.4.1	外部割り込み要因	5-16
5.4.2	内部割り込み	5-17
5.5	割り込み例外処理ベクタテーブル.....	5-18
5.6	割り込み制御モードと割り込み動作.....	5-24
5.6.1	割り込み制御モード 0	5-24
5.6.2	割り込み制御モード 2	5-26
5.6.3	割り込み例外処理シーケンス.....	5-28
5.6.4	割り込み応答時間	5-29
5.6.5	割り込みによる DTC、DMAC の起動.....	5-30
5.7	CPUに対するDTC、DMACの優先レベル制御機能.....	5-33
5.8	使用上の注意事項	5-35
5.8.1	割り込みの発生とディスエーブルとの競合.....	5-35
5.8.2	割り込みを禁止している命令.....	5-37
5.8.3	割り込み禁止期間	5-37
5.8.4	EEPMOV 命令実行中の割り込み.....	5-37
5.8.5	MOVMD、MOVSD 命令実行中の割り込み.....	5-37

5.8.6	周辺モジュール割り込み要因フラグ	5-37
6.	バスコントローラ (BSC)	6-1
6.1	特長	6-1
6.2	レジスタの説明	6-2
6.2.1	バスコントロールレジスタ 2 (BCR2)	6-2
6.3	バス構成	6-3
6.4	マルチクロック機能とアクセスステート数	6-4
6.5	内部バス	6-5
6.5.1	内部アドレス空間へのアクセス	6-5
6.6	ライトデータバッファ機能	6-6
6.7	バスアービトレーション	6-7
6.7.1	動作説明	6-7
6.7.2	バス権移行タイミング	6-7
6.8	リセットとバスコントローラ	6-8
6.9	使用上の注意事項	6-8
7.	DMA コントローラ (DMAC)	7-1
7.1	特長	7-1
7.2	レジスタの説明	7-3
7.2.1	DMA ソースアドレスレジスタ (DSAR)	7-4
7.2.2	DMA デスティネーションアドレスレジスタ (DDAR)	7-5
7.2.3	DMA オフセットレジスタ (DOFR)	7-5
7.2.4	DMA 転送カウントレジスタ (DTCR)	7-6
7.2.5	DMA ブロックサイズレジスタ (DBSR)	7-7
7.2.6	DMA モードコントロールレジスタ (DMDR)	7-8
7.2.7	DMA アドレスコントロールレジスタ (DACR)	7-14
7.2.8	DMA モジュールリクエストセレクトレジスタ (DMRSR)	7-19
7.3	転送モード	7-19
7.4	動作説明	7-20
7.4.1	アドレスモード	7-20
7.4.2	転送モード	7-24
7.4.3	起動要因	7-28
7.4.4	バスモード	7-30
7.4.5	拡張リピートエリア機能	7-32
7.4.6	オフセットを使ったアドレス更新機能	7-34
7.4.7	DMA 転送中のレジスタ	7-38
7.4.8	チャンネルの優先順位	7-43

7.4.9	基本バスサイクル	7-44
7.4.10	デュアルアドレスモードのバスサイクル.....	7-45
7.4.11	シングルアドレスモードのバスサイクル.....	7-54
7.5	DMA転送終了	7-59
7.6	DMACと他のバスマスタの関係.....	7-61
7.6.1	CPU に対する DMAC の優先レベル制御機能.....	7-61
7.6.2	他のバスマスタとのバス権の調停.....	7-62
7.7	割り込み要因	7-63
7.8	使用上の注意事項	7-66
8.	データトランスファコントローラ (DTC)	8-1
8.1	特長	8-1
8.2	レジスタの説明	8-3
8.2.1	DTC モードレジスタ A (MRA)	8-4
8.2.2	DTC モードレジスタ B (MRB)	8-5
8.2.3	DTC ソースアドレスレジスタ (SAR)	8-6
8.2.4	DTC デスティネーションアドレスレジスタ (DAR)	8-6
8.2.5	DTC 転送カウントレジスタ A (CRA)	8-6
8.2.6	DTC 転送カウントレジスタ B (CRB)	8-7
8.2.7	DTC イネーブルレジスタ A~G (DTCERA~DTCERG)	8-8
8.2.8	DTC コントロールレジスタ (DTCCR)	8-9
8.2.9	DTC ベクタベースレジスタ (DTCVBR)	8-10
8.3	起動要因	8-10
8.4	転送情報の配置とDTCベクタテーブル.....	8-10
8.5	動作説明	8-15
8.5.1	バスサイクルの分割	8-17
8.5.2	転送情報リードスキップ機能.....	8-19
8.5.3	転送情報ライトバックスキップ機能.....	8-19
8.5.4	ノーマル転送モード	8-20
8.5.5	リピート転送モード	8-21
8.5.6	ブロック転送モード	8-22
8.5.7	チェイン転送	8-23
8.5.8	動作タイミング	8-24
8.5.9	DTC の実行ステート	8-25
8.5.10	DTC のバス権解放タイミング.....	8-26
8.5.11	CPU に対する DTC の優先レベル制御.....	8-26
8.6	割り込みによるDTCの起動	8-27
8.7	DTC使用例	8-28

8.7.1	ノーマル転送	8-28
8.7.2	チェイン転送	8-28
8.7.3	カウンタ=0のときのチェイン転送	8-29
8.8	割り込み要因	8-30
8.9	使用上の注意	8-31
8.9.1	モジュールストップモードの設定	8-31
8.9.2	内蔵 RAM	8-31
8.9.3	DMAC 転送終了割り込み	8-31
8.9.4	DTCE ビットの設定	8-31
8.9.5	チェイン転送	8-31
8.9.6	転送情報先頭アドレス/ソースアドレス/デスティネーションアドレス	8-31
8.9.7	転送情報の書き換え	8-32
8.9.8	エンディアン	8-32
9.	I/O ポート	9-1
9.1	レジスタの説明	9-5
9.1.1	データディレクションレジスタ (PnDDR) (n=1、3、6、A、D、H、J、K)	9-7
9.1.2	データレジスタ (PnDR) (n=1、3、6、A、D、H、J、K)	9-7
9.1.3	ポートレジスタ (PORTn) (n=1、3、4、5、6、A、D、H、J、K)	9-7
9.1.4	入力バッファコントロールレジスタ (PnICR) (n=1、3、4、5、6、A、D、H、J、K)	9-8
9.1.5	プルアップ MOS コントロールレジスタ (PnPCR) (n=D、H、J、K)	9-8
9.1.6	ドライバビリティ設定レジスタ (DSR) (n=1、6、A、D、H)	9-9
9.1.7	端子状態設定レジスタ (PSR) (n=1、6、A、D、H)	9-9
9.2	出力バッファ制御	9-10
9.2.1	ポート 1	9-10
9.2.2	ポート 3	9-12
9.2.3	ポート 6	9-15
9.2.4	ポート A	9-17
9.2.5	ポート D	9-19
9.2.6	ポート H	9-21
9.2.7	ポート J	9-21
9.2.8	ポート K	9-24
9.3	ポートファンクションコントローラ	9-33
9.3.1	ポートファンクションコントロールレジスタ 5 (PFCR5)	9-33
9.3.2	ポートファンクションコントロールレジスタ 6 (PFCR6)	9-34
9.3.3	ポートファンクションコントロールレジスタ 8 (PFCR8)	9-35
9.3.4	ポートファンクションコントロールレジスタ 9 (PFCR9)	9-36
9.3.5	ポートファンクションコントロールレジスタ A (PFCRA)	9-37

9.3.6	ポートファンクションコントロールレジスタ B (PFCRB)	9-38
9.3.7	ポートファンクションコントロールレジスタ C (PFCRC)	9-39
9.3.8	ポートファンクションコントロールレジスタ D (PFCRD)	9-40
9.4	使用上の注意事項	9-41
9.4.1	入力バッファコントロールレジスタ (ICR) の設定	9-41
9.4.2	ポートファンクションコントロールレジスタ (PFCR) の設定	9-41
10.	16 ビットタイマパルスユニット (TPU)	10-1
10.1	特長	10-1
10.2	入出力端子	10-8
10.3	レジスタの説明	10-10
10.3.1	タイマコントロールレジスタ (TCR)	10-14
10.3.2	タイマモードレジスタ (TMDR)	10-18
10.3.3	タイマ I/O コントロールレジスタ (TIOR)	10-19
10.3.4	タイマインタラプトイネーブルレジスタ (TIER)	10-37
10.3.5	タイマステータスレジスタ (TSR)	10-38
10.3.6	タイマカウンタ (TCNT)	10-41
10.3.7	タイマジェネラルレジスタ (TGR)	10-41
10.3.8	タイマスタートレジスタ (TSTR)	10-42
10.3.9	タイマシンクロレジスタ (TSYR)	10-43
10.4	動作説明	10-44
10.4.1	基本動作	10-44
10.4.2	同期動作	10-50
10.4.3	バッファ動作	10-52
10.4.4	カスケード接続動作	10-56
10.4.5	PWM モード	10-58
10.4.6	位相計数モード	10-63
10.5	割り込み要因	10-69
10.6	DTCの起動	10-70
10.7	A/D変換器の起動	10-70
10.8	動作タイミング	10-71
10.8.1	入出力タイミング	10-71
10.8.2	割り込み信号タイミング	10-74
10.9	使用上の注意事項	10-77
10.9.1	モジュールストップモードの設定	10-77
10.9.2	入力クロックの制限事項	10-77
10.9.3	周期設定上の注意事項	10-78
10.9.4	TCNT のライトとクリアの競合	10-78

10.9.5	TCNT のライトとカウントアップの競合	10-79
10.9.6	TGR のライトとコンペアマッチの競合	10-80
10.9.7	バッファレジスタのライトとコンペアマッチの競合	10-81
10.9.8	TGR のリードとインプットキャプチャの競合	10-81
10.9.9	TGR のライトとインプットキャプチャの競合	10-82
10.9.10	バッファレジスタのライトとインプットキャプチャの競合	10-82
10.9.11	オーバフロー／アンダフローとカウンタクリアの競合	10-83
10.9.12	TCNT のライトとオーバフロー／アンダフローの競合	10-83
10.9.13	モジュールストップ時の割り込み	10-84
11.	プログラマブルパルスジェネレータ (PPG)	11-1
11.1	特長	11-1
11.2	入出力端子	11-2
11.3	レジスタの説明	11-2
11.3.1	ネクストデータインネブルレジスタ H、L (NDERH、NDERL)	11-3
11.3.2	アウトプットデータレジスタ H、L (PODRH、PODRL)	11-4
11.3.3	ネクストデータレジスタ H、L (NDRH、NDRL)	11-5
11.3.4	PPG 出力コントロールレジスタ (PCR)	11-7
11.3.5	PPG 出力モードレジスタ (PMR)	11-8
11.4	動作説明	11-9
11.4.1	出力タイミング	11-9
11.4.2	通常動作のパルス出力設定手順例	11-11
11.4.3	パルス出力通常動作例 (5 相パルス出力例)	11-12
11.4.4	パルス出力ノンオーバラップ動作	11-13
11.4.5	ノンオーバラップ動作のパルス出力設定手順例	11-14
11.4.6	パルス出力ノンオーバラップ動作例 (4 相の相補ノンオーバラップ出力例)	11-15
11.4.7	パルス反転出力	11-17
11.4.8	インプットキャプチャによるパルス出力	11-18
11.5	使用上の注意事項	11-18
11.5.1	モジュールストップモードの設定	11-18
11.5.2	パルス出力端子の動作	11-18
12.	ウォッチドッグタイマ (WDT)	12-1
12.1	特長	12-1
12.2	レジスタの説明	12-2
12.2.1	タイマカウンタ (TCNT)	12-2
12.2.2	タイマコントロール／ステータスレジスタ (TCSR)	12-3
12.2.3	リセットコントロール／ステータスレジスタ (RSTCSR)	12-5

12.3	動作説明	12-6
12.3.1	ウォッチドッグタイマモード.....	12-6
12.3.2	インターバルタイマモード.....	12-7
12.4	割り込み要因	12-7
12.5	使用上の注意事項	12-8
12.5.1	レジスタアクセス時の注意.....	12-8
12.5.2	タイマカウンタ (TCNT) のライトとカウントアップの競合.....	12-9
12.5.3	CKS2~CKS0 ビットの書き換え	12-9
12.5.4	ウォッチドッグタイマモードとインターバルタイマモードの切り替え	12-9
12.5.5	ウォッチドッグタイマモードとソフトウェアスタンバイモードへの遷移.....	12-9
13.	シリアルコミュニケーションインタフェース (SCI)	13-1
13.1	特長	13-1
13.2	入出力端子	13-3
13.3	レジスタの説明	13-4
13.3.1	レシーブシフトレジスタ (RSR)	13-5
13.3.2	レシーブデータレジスタ (RDR)	13-5
13.3.3	トランスミットデータレジスタ (TDR)	13-5
13.3.4	トランスミットシフトレジスタ (TSR)	13-5
13.3.5	シリアルモードレジスタ (SMR)	13-6
13.3.6	シリアルコントロールレジスタ (SCR)	13-8
13.3.7	シリアルステータスレジスタ (SSR)	13-10
13.3.8	ビットレートレジスタ (BRR)	13-13
13.4	調歩同期式モードの動作	13-18
13.4.1	送受信フォーマット	13-19
13.4.2	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	13-20
13.4.3	クロック	13-21
13.4.4	SCI の初期化 (調歩同期式)	13-22
13.4.5	シリアルデータ送信 (調歩同期式)	13-23
13.4.6	シリアルデータ受信 (調歩同期式)	13-25
13.5	マルチプロセッサ通信機能	13-29
13.5.1	マルチプロセッサシリアルデータ送信.....	13-30
13.5.2	マルチプロセッサシリアルデータ受信.....	13-31
13.6	クロック同期式モードの動作.....	13-34
13.6.1	クロック	13-34
13.6.2	SCI の初期化 (クロック同期式)	13-35
13.6.3	シリアルデータ送信 (クロック同期式)	13-36
13.6.4	シリアルデータ受信 (クロック同期式)	13-38

13.6.5	シリアルデータ送受信同時動作（クロック同期式）	13-40
13.7	割り込み要因	13-42
13.7.1	通常のシリアルコミュニケーションインタフェースモードにおける割り込み	13-42
13.8	使用上の注意事項	13-43
13.8.1	モジュールストップモードの設定	13-43
13.8.2	ブレークの検出と処理について	13-43
13.8.3	マーク状態とブレークの送付	13-43
13.8.4	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）	13-43
13.8.5	TDR へのライトと TDRE フラグの関係について	13-43
13.8.6	DMAC または DTC 使用上の制約事項	13-44
13.8.7	低消費電力状態遷移時の動作について	13-44
13.8.8	クロック同期式モード外部クロック入力	13-47
14.	コントローラエリアネットワーク（RCAN-TL1）	14-1
14.1	特長	14-1
14.1.1	RCAN-TL1 の特長	14-1
14.1.2	本マイコンにおける特長	14-2
14.2	構成	14-2
14.2.1	ブロック図	14-2
14.2.2	各ブロックの機能	14-4
14.2.3	端子構成	14-5
14.2.4	メモリマップ	14-5
14.3	メールボックス	14-7
14.3.1	メールボックスの構成	14-7
14.3.2	メッセージコントロールフィールド	14-12
14.3.3	ローカルアクセプタンスフィルタマスク（LAFM）	14-18
14.3.4	メッセージデータフィールド	14-19
14.3.5	タイムスタンプ	14-19
14.3.6	送信トリガタイム（TTT）とタイムトリガコントロール	14-21
14.4	RCAN-TL1のコントロールレジスタ	14-25
14.4.1	マスタコントロールレジスタ（MCR）	14-25
14.4.2	ジェネラルステータスレジスタ（GSR）	14-31
14.4.3	ビットコンフィギュレーションレジスタ 0、1（BCR0、BCR1）	14-33
14.4.4	インタラプトリクエストレジスタ（IRR）	14-38
14.4.5	インタラプトマスクレジスタ（IMR）	14-45
14.4.6	送信エラーカウンタ（TEC）／受信エラーカウンタ（REC）	14-46
14.5	RCAN-TL1のメールボックスレジスタ	14-47
14.5.1	送信待ちレジスタ 1、0（TXPR1、TXPR0）	14-48

14.5.2	送信キャンセルレジスタ 1、0 (TXCR1、TXCR0)	14-51
14.5.3	送信アックノリッジレジスタ 1、0 (TXACK1、TXACK0)	14-53
14.5.4	アボートアックノリッジレジスタ 1、0 (ABACK1、ABACK0)	14-55
14.5.5	データフレーム受信完了レジスタ 1、0 (RXPR1、RXPR0)	14-57
14.5.6	リモートフレーム受信完了レジスタ 1、0 (RFPR1、RFPR0)	14-58
14.5.7	メールボックスインタラプトマスクレジスタ 1、0 (MBIMR1、MBIMR0)	14-60
14.5.8	未読メッセージステータスレジスタ 1、0 (UMSR1、UMSR0)	14-61
14.5.9	タイマレジスタ	14-63
14.5.10	タイムトリガコントロールレジスタ 0 (TTCR0)	14-64
14.5.11	サイクルマキシマム/Tx_Enable_Window レジスタ (CMAX_TEW)	14-66
14.5.12	リファレンストリガオフセットレジスタ (RFTROFF)	14-68
14.5.13	タイマステータスレジスタ (TSR)	14-69
14.5.14	サイクルカウンタレジスタ (CCR)	14-72
14.5.15	タイマカウンタレジスタ (TCNTR)	14-73
14.5.16	サイクルタイムレジスタ (CYCTR)	14-74
14.5.17	リファレンスマークレジスタ (RFMK)	14-74
14.5.18	タイマコンペアマッチレジスタ 0~2 (TCMR0~2)	14-75
14.5.19	送信トリガタイムセレクトレジスタ (TTTSEL)	14-77
14.6	動作説明	14-78
14.6.1	RCAN-TL1 の設定	14-78
14.6.2	テストモードの設定	14-83
14.6.3	メッセージ送信シーケンス	14-85
14.6.4	メッセージ受信シーケンス	14-96
14.6.5	メールボックスの再設定	14-99
14.7	割り込み要因	14-101
14.8	DMACインタフェース	14-102
14.9	PORTインタフェース	14-103
14.9.1	RCAN モニタレジスタ (RCANMON)	14-103
14.10	CANバスインタフェース	14-104
14.11	RCAN-TL1端子ポート設定	14-105
14.12	使用上の注意事項	14-106
14.12.1	1チャンネル64メールボックスでのポート設定についての注意事項	14-106
14.12.2	モジュールストップモード	14-106
14.12.3	リセット	14-107
14.12.4	CAN スリープモード	14-107
14.12.5	レジスタアクセス	14-107
14.12.6	割り込み	14-107

15. ルネサスシリアルペリフェラルインタフェース (RSPI)	15-1
15.1 特長	15-1
15.2 入出力端子	15-4
15.3 レジスタの説明	15-5
15.3.1 RSPI 制御レジスタ (SPCR)	15-9
15.3.2 RSPI スレーブセレクト極性レジスタ (SSLP)	15-11
15.3.3 RSPI 端子制御レジスタ (SPPCR)	15-12
15.3.4 RSPI ステータスレジスタ (SPSR)	15-13
15.3.5 RSPI データレジスタ (SPDR)	15-17
15.3.6 RSPI シーケンス制御レジスタ (SPSCR)	15-18
15.3.7 RSPI シーケンスステータスレジスタ (SPSSR)	15-19
15.3.8 RSPI ビットレートレジスタ (SPBR)	15-21
15.3.9 RSPI データコントロールレジスタ (SPDCR)	15-22
15.3.10 RSPI クロック遅延レジスタ (SPCKD)	15-26
15.3.11 RSPI スレーブセレクトネゲート遅延レジスタ (SSLND)	15-27
15.3.12 RSPI 次アクセス遅延レジスタ (SPND)	15-28
15.3.13 RSPI 制御レジスタ 2 (SPCR2)	15-29
15.3.14 RSPI コマンドレジスタ (SPCMD)	15-31
15.4 動作説明	15-35
15.4.1 RSPI 動作の概要	15-35
15.4.2 RSPI 端子の制御	15-37
15.4.3 RSPI システム構成例	15-38
15.4.4 転送フォーマット	15-44
15.4.5 データフォーマット	15-46
15.4.6 通信動作モード	15-55
15.4.7 送信バッファエンプティ/受信バッファフルフラグ	15-57
15.4.8 エラー検出	15-58
15.4.9 RSPI の初期化	15-63
15.4.10 SPI 動作	15-64
15.4.11 クロック同期式動作	15-75
15.4.12 エラー処理	15-82
15.4.13 ループバックモード	15-84
15.4.14 パリティ機能の自己診断	15-85
15.4.15 割り込み要因	15-86
16. ハードウェア LIN (HWLIN)	16-1
16.1 特長	16-1
16.2 入出力端子	16-2

16.3	レジスタの説明	16-3
16.3.1	LIN コントロールレジスタ (LINCR)	16-3
16.3.2	LIN ステータスレジスタ (LINSTR)	16-4
16.3.3	LIN タイマコントロールレジスタ (LINTCR)	16-6
16.3.4	LIN タイマカウンタ (LINTCNT)	16-7
16.3.5	LIN タイムアウトカウンタ (LINTOCNT)	16-7
16.3.6	LIN タイマカウンタと LIN タイムアウトカウンタの設定.....	16-8
16.4	動作説明	16-13
16.4.1	マスタモード	16-13
16.4.2	スレーブモード	16-17
16.5	バス衝突検出機能	16-23
16.6	タイムアウト検出機能	16-24
16.6.1	マスタモードでのタイムアウト検出.....	16-24
16.6.2	スレーブモードでのタイムアウト検出.....	16-25
16.7	ウェイクアップ検出方法	16-25
16.8	割り込み要求	16-26
16.9	使用上の注意事項	16-27
16.9.1	モジュールストップモードの設定.....	16-27
16.9.2	LIN タイマカウンタ (LINTCNT) のライトとカウントダウンの競合	16-27
16.9.3	LIN タイムアウトカウンタ (LINTOCNT) のライトとカウントダウンの競合.....	16-28
16.9.4	LIN タイマカウンタ (LINTCNT) のライトとアンダフローの競合	16-29
16.9.5	LIN タイムアウトカウンタ (LINTOCNT) のライトとアンダフローの競合.....	16-30
17.	CRC 演算器	17-1
17.1	特長	17-1
17.2	レジスタの説明	17-1
17.2.1	CRC コントロールレジスタ (CRCCR)	17-2
17.2.2	CRC データ入力レジスタ (CRCDIR)	17-3
17.2.3	CRC データ出力レジスタ (CRCDOR)	17-3
17.3	CRC演算器の動作説明	17-4
17.4	CRC演算器使用上の注意事項.....	17-7
18.	A/D 変換器	18-1
18.1	特長	18-1
18.2	入出力端子	18-4
18.3	レジスタの説明	18-5
18.3.1	A/D データレジスタ A~H (ADDRA~ADDRH)	18-6
18.3.2	A/D コントロール/ステータスレジスタ (ADCSR)	18-7

18.3.3	A/D コントロールレジスタ (ADCR)	18-8
18.3.4	アナログポートプルダウンコントロールレジスタ (APPDCR)	18-10
18.3.5	A/D 自己診断レジスタ (ADDIAGR)	18-10
18.4	動作説明	18-11
18.4.1	シングルモード	18-11
18.4.2	スキャンモード	18-12
18.4.3	入力サンプリングと A/D 変換時間.....	18-13
18.4.4	外部トリガ入力タイミング.....	18-14
18.5	割り込み要因	18-15
18.6	A/D変換精度の定義	18-15
18.7	アナログポートプルダウン機能.....	18-17
18.8	A/D変換器の自己診断.....	18-18
18.9	使用上の注意事項	18-19
18.9.1	モジュールストップモードの設定.....	18-19
18.9.2	ソフトウェアスタンバイ時の A/D 変換保持機能.....	18-19
18.9.3	許容信号源インピーダンスについて.....	18-19
18.9.4	絶対精度への影響	18-20
18.9.5	アナログ電源端子他の設定範囲.....	18-20
18.9.6	ボード設計上の注意	18-20
18.9.7	ノイズ対策上の注意	18-20
18.9.8	A/D 変換器停止方法に関する注意事項.....	18-22
19.	RAM.....	19-1
19.1	特長	19-1
19.2	レジスタの説明	19-2
19.2.1	RAM イネーブルコントロールレジスタ (RAMEN)	19-3
19.2.2	RAM 書き込みイネーブルコントロールレジスタ (RAMWEN)	19-6
19.2.3	RAM ECC イネーブルコントロールレジスタ (RAMECC)	19-9
19.2.4	RAM エラーステータスレジスタ (RAMERR)	19-11
19.2.5	RAM エラー割り込みコントロールレジスタ (RAMINT)	19-13
19.2.6	RAM アクセスサイクル設定レジスタ (RAMACYC)	19-14
19.2.7	レジスタアクセス時の注意.....	19-16
19.3	動作説明	19-17
19.4	RAMのデータ保持	19-18
19.4.1	リセット時のデータ保持	19-18
19.5	使用上の注意事項	19-18
19.5.1	電源立ち上げ後の RAM 初期化	19-18

20. フラッシュメモリ	20-1
20.1 特長	20-1
20.2 入出力端子	20-4
20.3 レジスタの説明	20-5
20.3.1 フラッシュモードレジスタ (FMODR)	20-6
20.3.2 フラッシュアクセスステータスレジスタ (FASTAT)	20-7
20.3.3 フラッシュアクセスエラー割り込み許可レジスタ (FAEINT)	20-9
20.3.4 フラッシュメモリマツト選択レジスタ (ROMMAT)	20-10
20.3.5 FCU RAM イネーブルレジスタ (FCURAME)	20-11
20.3.6 フラッシュステータスレジスタ 0 (FSTATR0)	20-12
20.3.7 フラッシュステータスレジスタ 1 (FSTATR1)	20-15
20.3.8 FCU RAM ECC エラーコントロールレジスタ (FRAMECCR)	20-17
20.3.9 フラッシュ P/E モードエントリレジスタ (FENTRYR)	20-18
20.3.10 フラッシュプロテクトレジスタ (FPROTR)	20-20
20.3.11 フラッシュリセットレジスタ (FRESETR)	20-21
20.3.12 FCU コマンドレジスタ (FCMDR)	20-22
20.3.13 FCU 処理切り替えレジスタ (FCPSR)	20-23
20.3.14 フラッシュ P/E ステータスレジスタ (FPESTAT)	20-24
20.3.15 フラッシュクロック通知レジスタ (FCKAR)	20-25
20.4 フラッシュメモリ関連モード概要	20-26
20.5 ブートモード	20-28
20.5.1 システム構成	20-28
20.5.2 ブートモードの状態遷移	20-29
20.5.3 ビットレートの自動調整	20-31
20.5.4 問い合わせ設定ホストコマンド待ち状態	20-32
20.5.5 書き込み/消去ホストコマンド待ち状態	20-47
20.6 ユーザプログラムモード	20-57
20.6.1 FCU コマンド一覧	20-57
20.6.2 FCU コマンド受け付け条件	20-59
20.6.3 FCU コマンド使用方法	20-62
20.6.4 サスペンド動作	20-80
20.7 ライターモード	20-83
20.8 プロテクト	20-83
20.8.1 ハードウェアプロテクト	20-83
20.8.2 ソフトウェアプロテクト	20-83
20.8.3 エラープロテクト	20-84
20.9 使用上の注意事項	20-86
20.9.1 その他のご注意	20-86

21. データフラッシュ (EEPROM)	21-1
21.1 特長	21-1
21.2 入出力端子	21-5
21.3 レジスタの説明	21-6
21.3.1 フラッシュモードレジスタ (FMODR)	21-7
21.3.2 フラッシュアクセスステータスレジスタ (FASTAT)	21-8
21.3.3 フラッシュアクセスエラー割り込み許可レジスタ (FAEINT)	21-10
21.3.4 EEPROM 読み出し許可レジスタ 0 (EEPRE0)	21-12
21.3.5 EEPROM 読み出し許可レジスタ 1 (EEPRE1)	21-13
21.3.6 EEPROM 書き込み/消去許可レジスタ 0 (EEPWE0)	21-14
21.3.7 EEPROM 書き込み/消去許可レジスタ 1 (EEPWE1)	21-15
21.3.8 フラッシュ P/E モードエントリレジスタ (FENTRYR)	21-16
21.3.9 EEPROM ブランクチェック制御レジスタ (EEPBCCNT)	21-17
21.3.10 EEPROM ブランクチェックステータスレジスタ (EEPBCSTAT)	21-18
21.3.11 EEPROM マット選択レジスタ (EEPMAT)	21-19
21.4 EEPROM関連モード概要	21-20
21.5 ブートモード	21-22
21.5.1 問い合わせ設定ホストコマンド	21-22
21.5.2 書き込み/消去ホストコマンド	21-23
21.6 ユーザモード/ユーザプログラムモード	21-25
21.6.1 FCU コマンド一覧	21-25
21.6.2 FCU コマンド受け付け条件	21-27
21.6.3 FCU コマンド使用方法	21-30
21.7 プロテクト	21-34
21.7.1 ハードウェアプロテクト	21-34
21.7.2 ソフトウェアプロテクト	21-34
21.7.3 エラープロテクト	21-35
21.8 製品情報マット	21-36
21.9 使用上の注意事項	21-37
22. クロック発振器	22-1
22.1 レジスタの説明	22-3
22.1.1 システムクロックコントロールレジスタ 0 (SCKCR0)	22-4
22.1.2 システムクロックコントロールレジスタ 1 (SCKCR1)	22-6
22.1.3 リカバリーオシレータコントロールレジスタ (ROSCCR)	22-8
22.2 発振器	22-10
22.2.1 水晶発振子を接続する方法	22-10
22.2.2 外部クロックを入力する方法	22-11

22.3	PLL回路	22-11
22.4	メインクロック分周器	22-11
22.5	外部発振停止検出機能	22-12
22.5.1	概要	22-12
22.5.2	外部発振停止検出機能の設定方法.....	22-12
22.5.3	外部発振停止検出機能動作時の注意事項.....	22-13
22.6	使用上の注意事項	22-13
22.6.1	クロック発振器に関する使用上の注意事項.....	22-13
22.6.2	発振子に関する注意事項	22-14
22.6.3	ボード設計上の注意	22-14
22.6.4	入力クロック周波数に関する注意.....	22-15
23.	低消費電力.....	23-1
23.1	特長	23-1
23.2	レジスタの説明	23-4
23.2.1	スタンバイコントロールレジスタ (SBYCR)	23-4
23.2.2	モジュールストップコントロールレジスタ A、B、C、D、E (MSTPCRA、MSTPCRB、MSTPCRC、MSTPCRD、E)	23-7
23.3	マルチクロック機能	23-14
23.4	モジュールストップ機能	23-15
23.5	スリープモード	23-15
23.5.1	スリープモードへの遷移	23-15
23.5.2	スリープモードの解除	23-15
23.6	全モジュールクロックストップモード.....	23-16
23.7	ソフトウェアスタンバイモード.....	23-17
23.7.1	ソフトウェアスタンバイモードへの遷移.....	23-17
23.7.2	ソフトウェアスタンバイモードの解除.....	23-17
23.7.3	ソフトウェアスタンバイモード解除後の発振安定時間の設定.....	23-18
23.7.4	ソフトウェアスタンバイモードの応用例.....	23-19
23.8	φクロック出力制御	23-20
23.9	使用上の注意事項	23-21
23.9.1	I/O ポートの状態	23-21
23.9.2	発振安定待機中の消費電流.....	23-21
23.9.3	DMAC、DTC のモジュールストップ.....	23-21
23.9.4	内蔵周辺モジュールの割り込み.....	23-21
23.9.5	MSTPCRA、MSTPCRB、MSTPCRC、MSTPCRD、MSTPCRE のライト.....	23-21

24. レジスタ一覧	24-1
24.1 レジスタアドレス一覧 (アドレス順)	24-2
24.2 レジスタビット一覧	24-44
24.3 各動作モードにおけるレジスタの状態	24-72
25. 電气的特性	25-1
25.1 絶対最大定格	25-1
25.2 DC特性	25-2
25.3 AC特性	25-5
25.3.1 クロックタイミング	25-6
25.3.2 制御信号タイミング	25-9
25.3.3 内蔵周辺モジュールタイミング	25-11
25.3.4 A/D 変換特性	25-19
25.4 フラッシュメモリ特性	25-20
25.5 EEPROM特性	25-22
25.6 その他の特性	25-23
25.6.1 外部発振停止検出条件	25-23
25.6.2 外部発振停止検出タイミング	25-23
25.6.3 内部発振周波数	25-24
付録	付録-1
A. 各処理状態におけるポートの状態	付録-1
B. 型名一覧	付録-1
C. 外形寸法図	付録-2
本版で修正または追加された箇所	改訂-1
索引	索引-1

1. 概要

1.1 特長

- 32ビット高速H8SX V2 CPU
H8/300 CPU、H8/300H CPU、H8S および H8SX V1 CPU とオブジェクトレベルで上位互換
汎用レジスタ：16 ビット×16 本
基本命令：87 種類
- 豊富な周辺機能
DMA コントローラ (DMAC)
データトランスファコントローラ (DTC)
16 ビットタイマパルスユニット (TPU)
プログラマブルパルスジェネレータ (PPG)
ウォッチドッグタイマ (WDT)
調歩同期式またはクロック同期式シリアルコミュニケーションインタフェース (SCI)
コントローラエリアネットワーク (RCAN-TL1)
ルネサスシリアルペリフェラルインタフェース (RSPI)
ハードウェア LIN (HWLIN)
CRC 演算器 (CRC)
10 ビット A/D 変換器
クロック発振器
- 内蔵メモリ

製品分類		製品型名	ROM (フラッシュ メモリ)	データ フラッシュ	RAM	動作周波数
フラッシュ メモリ版	H8SX/1727S	R5F61727S	512K バイト	32K バイト	40K バイト	8MHz~80MHz
	H8SX/1725S	R5F61725S	256K バイト	16K バイト	24K バイト	

- 汎用入出力ポート
入出力ポート：61 本
入力ポート：17 本
- 各種低消費電力モードをサポート
- 小型パッケージ

パッケージ	コード	ボディサイズ	ピンピッチ
LQFP-100	PLQP0100KB-A	14.0×14.0mm	0.50mm

1.2 内部ブロック図

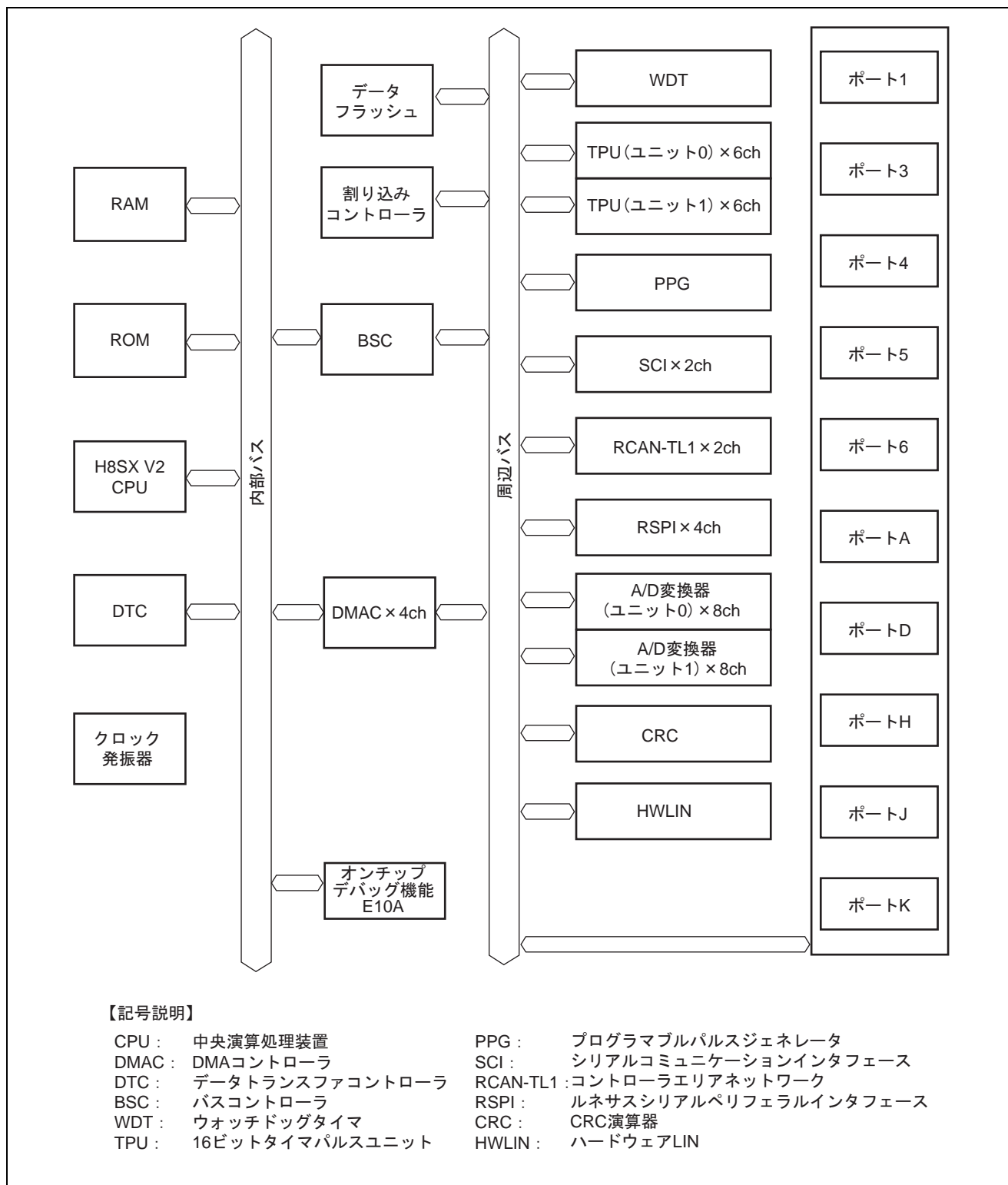


図 1.1 H8SX/1725S、H8SX/1727S 内部ブロック図

1.3 端子説明

1.3.1 ピン配置図

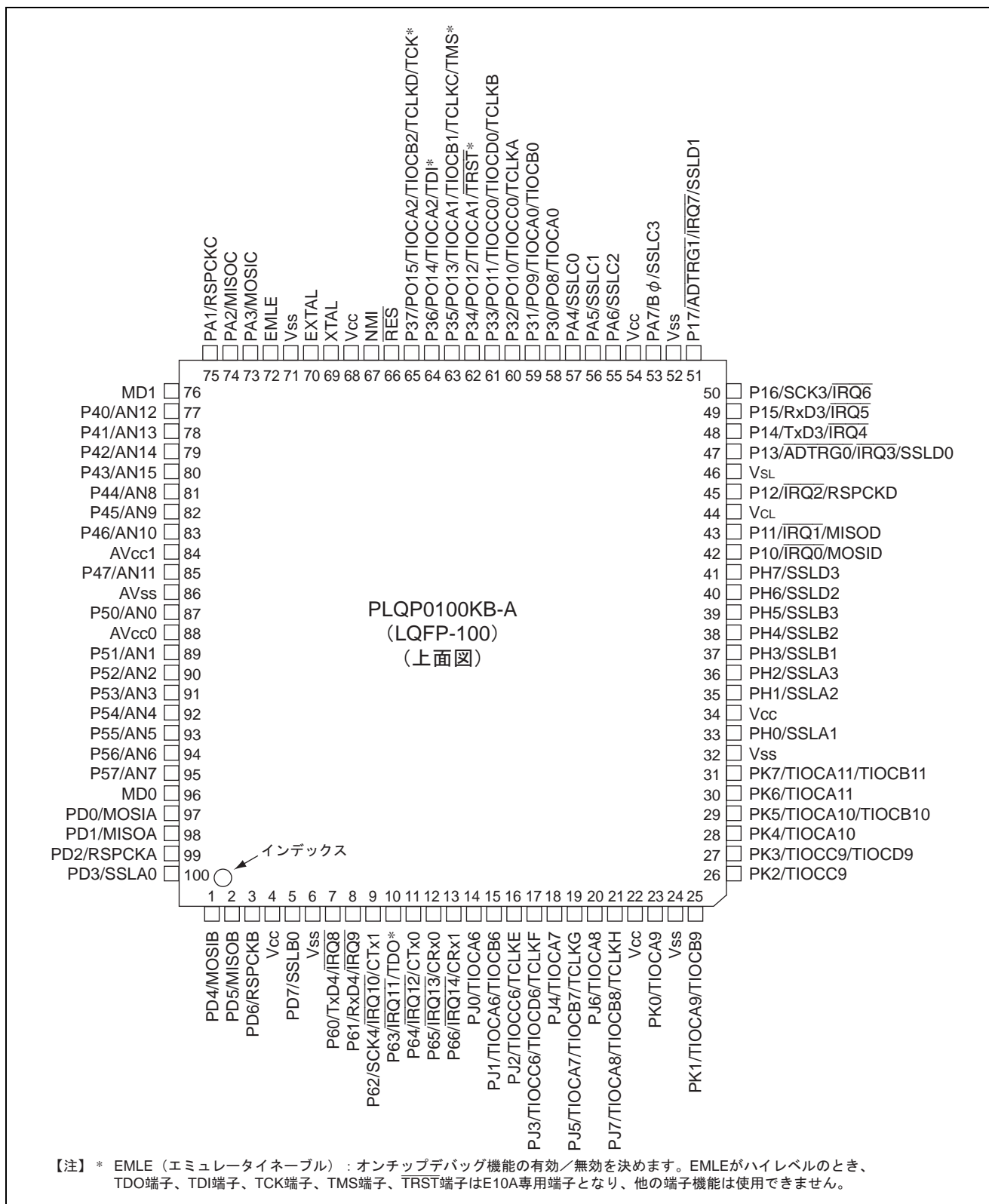


図 1.3 H8SX/1725S、H8SX/1727S ピン配置図

1.3.2 動作モード別ピン配置一覧

表 1.1 動作モード別ピン配置一覧

ピン番号	端子名
	モード 1、モード 2、モード 3
1	PD4/MOSIB
2	PD5/MISOB
3	PD6/RSPCKB
4	Vcc
5	PD7/SSLB0
6	Vss
7	P60/TxD4/ $\overline{\text{IRQ8}}$
8	P61/RxD4/ $\overline{\text{IRQ9}}$
9	P62/SCK4/ $\overline{\text{IRQ10}}$ /CTx1
10	P63/ $\overline{\text{IRQ11}}$ /TDO*
11	P64/ $\overline{\text{IRQ12}}$ /CTx0
12	P65/ $\overline{\text{IRQ13}}$ /CRx0
13	P66/ $\overline{\text{IRQ14}}$ /CRx1
14	PJ0/TIOCA6
15	PJ1/TIOCA6/TIOCB6
16	PJ2/TIOCC6/TCLKE
17	PJ3/TIOCC6/TIOCD6/TCLKF
18	PJ4/TIOCA7
19	PJ5/TIOCA7/TIOCB7/TCLKG
20	PJ6/TIOCA8
21	PJ7/TIOCA8/TIOCB8/TCLKH
22	Vcc
23	PK0/TIOCA9
24	Vss
25	PK1/TIOCA9/TIOCB9
26	PK2/TIOCC9
27	PK3/TIOCC9/TIOCD9
28	PK4/TIOCA10
29	PK5/TIOCA10/TIOCB10
30	PK6/TIOCA11
31	PK7/TIOCA11/TIOCB11
32	Vss
33	PH0/SSLA1

ピン番号	端子名
	モード 1、モード 2、モード 3
34	Vcc
35	PH1/SSLA2
36	PH2/SSLA3
37	PH3/SSLB1
38	PH4/SSLB2
39	PH5/SSLB3
40	PH6/SSLD2
41	PH7/SSLD3
42	P10/ $\overline{\text{IRQ0}}$ /MOSID
43	P11/ $\overline{\text{IRQ1}}$ /MISOD
44	V _{CL}
45	P12/ $\overline{\text{IRQ2}}$ /RSPCKD
46	V _{SL}
47	P13/ $\overline{\text{ADTRG0}}$ / $\overline{\text{IRQ3}}$ /SSLD0
48	P14/TxD3/ $\overline{\text{IRQ4}}$
49	P15/RxD3/ $\overline{\text{IRQ5}}$
50	P16/SCK3/ $\overline{\text{IRQ6}}$
51	P17/ $\overline{\text{ADTRG1}}$ / $\overline{\text{IRQ7}}$ /SSLD1
52	V _{SS}
53	PA7/B ϕ /SSLC3
54	Vcc
55	PA6/SSLC2
56	PA5/SSLC1
57	PA4/SSLC0
58	P30/PO8/TIOCA0
59	P31/PO9/TIOCA0/TIOCB0
60	P32/PO10/TIOCC0/TCLKA
61	P33/PO11/TIOCC0/TIOCD0/TCLKB
62	P34/PO12/TIOCA1/ $\overline{\text{TRST}}$ *
63	P35/PO13/TIOCA1/TIOCB1/TCLKC/TMS*
64	P36/PO14/TIOCA2/TDI*
65	P37/PO15/TIOCA2/TIOCB2/TCLKD/TCK*
66	RES
67	NMI
68	Vcc
69	XTAL

ピン番号	端子名
	モード 1、モード 2、モード 3
70	EXTAL
71	Vss
72	EMLE
73	PA3/MOSIC
74	PA2/MISOC
75	PA1/RSPCKC
76	MD1
77	P40/AN12
78	P41/AN13
79	P42/AN14
80	P43/AN15
81	P44/AN8
82	P45/AN9
83	P46/AN10
84	AVcc1
85	P47/AN11
86	AVss
87	P50/AN0
88	AVcc0
89	P51/AN1
90	P52/AN2
91	P53/AN3
92	P54/AN4
93	P55/AN5
94	P56/AN6
95	P57/AN7
96	MD0
97	PD0/MOSIA
98	PD1/MISOA
99	PD2/RSPCKA
100	PD3/SSLA0

【注】 * EMLE (エミュレータイネーブル) : オンチップデバッグ機能の有効/無効を決めます。EMLE がハイレベルのとき、TDO 端子、TDI 端子、TCK 端子、TMS 端子、TRST 端子は E10A 専用端子となり、他の端子機能は使用できません。

1.3.3 端子機能

表 1.2 端子機能

分類	記号	ピン番号	入出力	名称および機能
電源	V _{CC}	4、22、34、54、68	入力	電源端子です。システムの電源に接続してください。
	V _{CL}	44	入力	0.47 μ F のコンデンサを介して V _{SL} に接続してください (端子近くに配置してください)。
	V _{SL}	46	入力	V _{CL} 専用のグランド端子です。
	V _{SS}	6、24、32、52、71	入力	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。
クロック	XTAL	69	入力	水晶発振子接続端子です。また、EXTAL 端子は外部クロックを入力することもできます。接続例は、「第 22 章 クロック発振器」を参照してください。
	EXTAL	70	入力	
	B ϕ	53	出力	外部デバイスにシステムクロックを供給します。
動作モード コントロール	MD1	76	入力	動作モードを設定します。これらの端子は、動作中に変化させないでください。
	MD0	96		
システム制御	$\overline{\text{RES}}$	66	入力	リセット端子です。この端子が Low レベルになると、リセット状態となります。
	EMLE	72	入力	オンチップエミュレータイネーブル端子です。通常は Low レベルに設定してください。
割り込み	NMI	67	入力	ノンマスクブル割り込み要求端子です。未使用の場合は High レベルに固定してください。
	$\overline{\text{IRQ14}}$	13	入力	マスク可能な割り込みを要求します。
	$\overline{\text{IRQ13}}$	12		
	$\overline{\text{IRQ12}}$	11		
	$\overline{\text{IRQ11}}$	10		
	$\overline{\text{IRQ10}}$	9		
	$\overline{\text{IRQ9}}$	8		
	$\overline{\text{IRQ8}}$	7		
	$\overline{\text{IRQ7}}$	51		
	$\overline{\text{IRQ6}}$	50		
	$\overline{\text{IRQ5}}$	49		
	$\overline{\text{IRQ4}}$	48		
	$\overline{\text{IRQ3}}$	47		
	$\overline{\text{IRQ2}}$	45		
	$\overline{\text{IRQ1}}$	43		
	$\overline{\text{IRQ0}}$	42		

分類	記号	ピン番号	入出力	名称および機能
オンチップ エミュレータ	$\overline{\text{TRST}}$	62	入力	オンチップエミュレータ用デバッグインタフェース端子です。
	TMS	63	入力	
	TDO	10	出力	
	TDI	64	入力	
	TCK	65	入力	
16 ビットタイマ パルスユニット (TPU) (ユニット0)	TCLKA	60	入力	外部クロックを入力します。
	TCLKB	61		
	TCLKC	63		
	TCLKD	65		
	TIOCA0	58、59	入出力	TGRA_0~TGRD_0 のインプットキャプチャ入力/ アウトプットコンペア出力/PWM 出力端子です。
	TIOCB0	59		
	TIOCC0	60、61		
	TIOCD0	61		
	TIOCA1	62、63	入出力	TGRA_1、TGRB_1 のインプットキャプチャ入力/ アウトプットコンペア出力/PWM 出力端子です。
	TIOCB1	63		
TIOCA2	64、65	入出力	TGRA_2、TGRB_2 のインプットキャプチャ入力/ アウトプットコンペア出力/PWM 出力端子です。	
TIOCB2	65			
16 ビットタイマ パルスユニット (TPU) (ユニット1)	TCLKE	16	入力	外部クロックを入力します。
	TCLKF	17		
	TCLKG	19		
	TCLKH	21		
	TIOCA6	14、15	入出力	TGRA_6~TGRD_6 のインプットキャプチャ入力/ アウトプットコンペア出力/PWM 出力端子です。
	TIOCB6	15		
	TIOCC6	16、17		
	TIOCD6	17		
	TIOCA7	18、19	入出力	TGRA_7、TGRB_7 のインプットキャプチャ入力/ アウトプットコンペア出力/PWM 出力端子です。
	TIOCB7	19		
TIOCA8	20、21	入出力	TGRA_8、TGRB_8 のインプットキャプチャ入力/ アウトプットコンペア出力/PWM 出力端子です。	
TIOCB8	21			

分類	記号	ピン番号	入出力	名称および機能
16ビット タイマパルス ユニット (TPU) (ユニット1)	TIOCA9	23、25	入出力	TGRA_9~TGRD_9のインプットキャプチャ入力/ アウトプットコンペア出力/PWM出力端子です。
	TIOCB9	25		
	TIOCC9	26、27		
	TIOCD9	27	入出力	TGRA_10、TGRB_10のインプットキャプチャ入力/ アウトプットコンペア出力/PWM出力端子です。
	TIOCA10	28、29		
	TIOCB10	29		
プログラマブル パルス ジェネレータ (PPG)	PO15	65	出力	パルス出力端子です。
	PO14	64		
	PO13	63		
	PO12	62		
	PO11	61		
	PO10	60		
	PO9	59		
PO8	58			
シリアル コミュニケーション インタフェース (SCI)	TxD3	48	出力	送信データ出力端子です。
	TxD4	7		
	RxD3	49	入力	受信データ入力端子です。
	RxD4	8		
	SCK3	50	入出力	クロック入出力端子です。
	SCK4	9		
コントローラ エリア ネットワーク (RCAN-TL1)	CTx0	11	出力	CANバス送信用端子です。
	CTx1	9		
	CRx0	12	入力	CANバス受信用端子です。
	CRx1	13		

分類	記号	ピン番号	入出力	名称および機能
ルネサス シリアル ペリフェラル インタフェース (RSPI)	MOSID	42	入出力	データ入出力端子です。
	MOSIC	73		
	MOSIB	1		
	MOSIA	97		
	MISOD	43	入出力	データ入出力端子です。
	MISOC	74		
	MISOB	2		
	MISOA	98		
	RSPCKD	45	入出力	クロック入出力端子です。
	RSPCKC	75		
	RSPCKB	3		
	RSPCKA	99		
	SSLA3	36	入出力	チップセレクト入出力端子です。
	SSLA2	35		
	SSLA1	33		
	SSLA0	100		
	SSLB3	39	入出力	チップセレクト入出力端子です。
	SSLB2	38		
	SSLB1	37		
	SSLB0	5		
SSLC3	53	入出力	チップセレクト入出力端子です。	
SSLC2	55			
SSLC1	56			
SSLC0	57			
SSLD3	41	入出力	チップセレクト入出力端子です。	
SSLD2	40			
SSLD1	51			
SSLD0	47			

分類	記号	ピン番号	入出力	名称および機能
A/D 変換器	AN15	80	入力	A/D 変換器のアナログ入力端子です。
	AN14	79		
	AN13	78		
	AN12	77		
	AN11	85		
	AN10	83		
	AN9	82		
	AN8	81		
	AN7	95		
	AN6	94		
	AN5	93		
	AN4	92		
	AN3	91		
	AN2	90		
	AN1	89		
	AN0	87		
	ADTRG0	47	入力	A/D 変換開始のための外部トリガ入力端子です。
	ADTRG1	51		
	AV _{cc0}	88	入力	A/D 変換器のアナログ電源端子および基準電圧端子です。A/D 変換器を使用しない場合は、システムの電源に接続してください。
	AV _{cc1}	84		
	AV _{ss}	86	入力	A/D 変換器のグランド端子です。システムの電源 (0V) に接続してください。
I/O ポート	P17	51	入出力	8 ビットの入出力端子です。
	P16	50		
	P15	49		
	P14	48		
	P13	47		
	P12	45		
	P11	43		
	P10	42		

分類	記号	ピン番号	入出力	名称および機能
I/O ポート	P37	65	入出力	8 ビットの入出力端子です。
	P36	64		
	P35	63		
	P34	62		
	P33	61		
	P32	60		
	P31	59		
	P30	58		
	P47	85	入力	8 ビットの入力端子です。
	P46	83		
	P45	82		
	P44	81		
	P43	80		
	P42	79		
	P41	78		
	P40	77		
	P57	95	入力	8 ビットの入力端子です。
	P56	94		
	P55	93		
	P54	92		
	P53	91		
	P52	90		
	P51	89		
	P50	87		
	P66	13	入出力	7 ビットの入出力端子です。
	P65	12		
	P64	11		
	P63	10		
P62	9			
P61	8			
P60	7			

分類	記号	ピン番号	入出力	名称および機能
I/O ポート	PA7	53	入力	1 ビットの入力端子です。
	PA6	55	入出力	6 ビットの入出力端子です。
	PA5	56		
	PA4	57		
	PA3	73		
	PA2	74		
	PA1	75		
	PD7	5	入出力	8 ビットの入出力端子です。
	PD6	3		
	PD5	2		
	PD4	1		
	PD3	100		
	PD2	99		
	PD1	98		
	PD0	97		
	PH7	41	入出力	8 ビットの入出力端子です。
	PH6	40		
	PH5	39		
	PH4	38		
	PH3	37		
	PH2	36		
	PH1	35		
	PH0	33		
	PJ7	21	入出力	8 ビットの入出力端子です。
	PJ6	20		
	PJ5	19		
	PJ4	18		
	PJ3	17		
PJ2	16			
PJ1	15			
PJ0	14			

分類	記号	ピン番号	入出力	名称および機能
I/O ポート	PK7	31	入出力	8ビットの入出力端子です。
	PK6	30		
	PK5	29		
	PK4	28		
	PK3	27		
	PK2	26		
	PK1	25		
	PK0	23		

2. CPU

H8SX CPU は、H8/300 CPU、H8/300H CPU および H8S CPU に対し上位互換のアーキテクチャを持つ内部 32 ビット構成の高速 CPU です。

H8SX CPU は、16 ビット×16 本の汎用レジスタを持ち、4G バイトのリニアなアドレス空間を扱うことができ、リアルタイム制御に最適です。

2.1 特長

- H8/300 CPU、H8/300H CPU および H8S CPU に対し上位互換の CPU
これらの CPU のオブジェクトプログラムを実行可能
- 汎用レジスタ：16 ビット×16 本
8 ビット×16 本、32 ビット×8 本としても使用可能
- 基本命令：87 種類
8/16/32 ビット演算命令
乗除算命令
ビットフィールド転送命令
強力なビット操作命令
ビット条件分岐命令
積和演算命令
- アドレッシングモード：11 種類
レジスタ直接 Rn
レジスタ間接 @ERn
ディスプレイメント付レジスタ間接 @(d:2,ERn)/@(d:16,ERn)/@(d:32,ERn)
ディスプレイメント付インデックスレジスタ間接 @(d:16,RnL.B)/@(d:32,RnL.B)/@(d:16,Rn.W)/
@(d:32,Rn.W)/@(d:16,ERn.L)/@(d:32,ERn.L)
プリ/ポストインクリメント/デクリメントレジスタ間接 @+ERn/@-ERn/@ERn+/@ERn-
絶対アドレス @aa:8/@aa:16/@aa:24/@aa:32
イミディエイト #xx:3/#xx:4/#xx:8/#xx:16/#xx:32
プログラムカウンタ相対 @(d:8,PC)/@(d:16,PC)
プログラムカウンタインデックス相対 @(RnL.B,PC)/@(Rn.W,PC)/@(ERn.L,PC)
メモリ間接 @@aa:8
拡張メモリ間接 @@vec:7

- ベースレジスタ：2本
ベクタベースレジスタ
ショートアドレスベースレジスタ
- アドレス空間：4Gバイト
プログラム：4G バイト
データ：4G バイト
- 高速動作
頻出命令をすべて1~2 ステートで実行
8/16/32 ビットレジスタ間加減算：1 ステート
8×8 ビットレジスタ間乗算：1 ステート
16÷8 ビットレジスタ間除算：10 ステート
16×16 ビットレジスタ間乗算：1 ステート
32÷16 ビットレジスタ間除算：18 ステート
32×32 ビットレジスタ間乗算：5 ステート
32÷32 ビットレジスタ間除算：18 ステート
- CPU動作モード：4種類
ノーマルモード
ミドルモード
アドバンストモード
マキシマムモード
- 低消費電力状態
SLEEP 命令により低消費電力状態に遷移
CPU 動作クロックを選択可能

-
- 【注】 1. H8SX/1720S グループの CPU 動作モードは、アドバンストモードのみです。
ノーマルモード、ミドルモード、マキシマムモードは使用できません。
2. H8SX/1720S グループは、乗算器、除算器をサポートしています。
-

2.2 CPU 動作モード

H8SX CPU は、アドレス空間について、ノーマルモード、ミドルモード、アドバンスモードおよびマキシマムモードの4種類のCPU動作モードを持っています。各モードの選択方法は、「3.1 動作モードの選択」を参照してください。

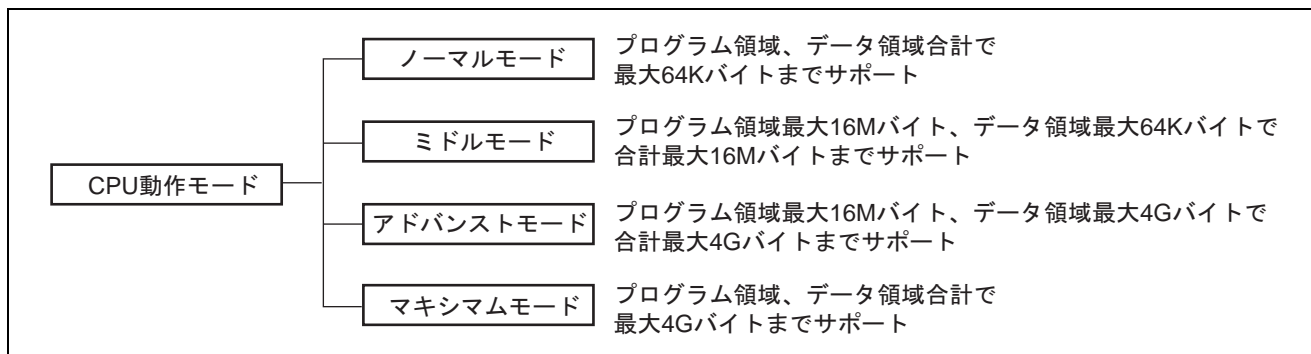


図 2.1 CPU 動作モード

2.2.1 ノーマルモード

ノーマルモードでは例外処理ベクタ、スタックの構造が H8/300 CPU と同一です。

【注】 本製品ではサポートしていません。

- アドレス空間
最大 64K バイトをアクセス可能です。
- 拡張レジスタ (En)
拡張レジスタ (E0~E7) は、16 ビットレジスタとして、または 32 ビットレジスタの上位 16 ビットとして使用できます。
拡張レジスタ En は、対応する汎用レジスタ Rn をアドレスレジスタとして使用している場合でも、16 ビットレジスタとして任意の値を設定することができます (プリ/ポストインクリメント/デクリメントレジスタ間接により汎用レジスタ Rn が参照された場合、キャリ/ボローが発生すると、対応する拡張レジスタ En の内容に伝播しますので注意してください)。
- 命令セット
命令およびアドレッシングモードはすべて使用できます。実効アドレス (EA) の下位 16 ビットのみが有効となります。
- 例外処理ベクタテーブルおよびメモリ間接の分岐アドレス
ノーマルモードでは、H'0000 から始まる先頭領域に例外処理ベクタテーブル領域が割り当てられており、各 16 ビットの分岐先アドレスを格納します。例外処理ベクタテーブルの構造を図 2.2 に示します。

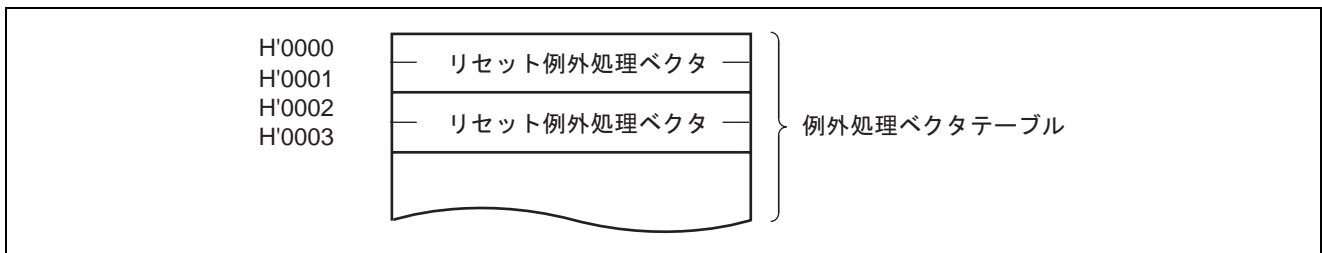


図 2.2 例外処理ベクタテーブル（ノーマルモード）

メモリ間接 (@@aa:8) および拡張メモリ間接 (@@ vec:7) は、JMP および JSR 命令で使用されます。命令コードに含まれる 8 ビット絶対アドレスによりメモリ上のオペランドを指定し、この内容が分岐先アドレスとなります。

- スタック構造

サブルーチン分岐時の PC のスタック構造と、例外処理時の PC と CCR のスタック構造を図 2.3 に示します。PC は 16 ビットで退避／復帰されます。

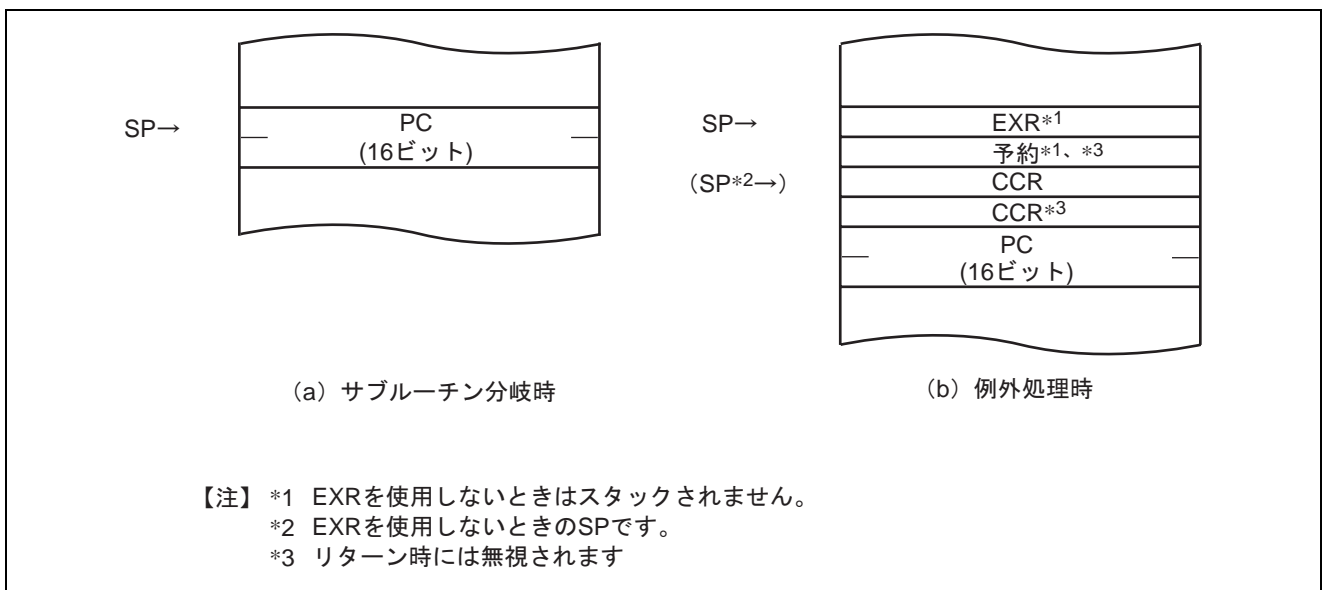


図 2.3 スタック構造（ノーマルモード）

【注】 H8SX/1720S グループでは本モードはサポートしません。

2.2.2 ミドルモード

ノーマルモードに対して、プログラム領域を 16M バイトに拡張しています。

- アドレス空間
プログラム領域最大 16M バイト、データ領域最大 64K バイトで合計最大 16M バイトをアクセス可能です。
- 拡張レジスタ (En)
拡張レジスタ (E0~E7) は、16 ビットレジスタとして、または 32 ビットレジスタの上位 16 ビットとして使用できます。
拡張レジスタ En は、対応する汎用レジスタ Rn をデータ用のアドレスレジスタとして使用している場合でも、16 ビットレジスタとして任意の値を設定することができます (JMP および JSR 命令は除きます。また、プリ/ポストインクリメント/デクリメントレジスタ間接により汎用レジスタ Rn が参照された場合、キャリ/ボローが発生すると、対応する拡張レジスタ En の内容に伝播しますので注意してください)。
- 命令セット
命令およびアドレッシングモードはすべて使用できます。データ用の実効アドレス (EA) は下位 16 ビットのみが有効となり、上位 8 ビットは符号拡張されます。
- 例外処理ベクタテーブルおよびメモリ間接の分岐アドレス
ミドルモードでは、H'000000 から始まる先頭領域に 32 ビット単位で例外処理ベクタテーブル領域が割り当てられており、上位 8 ビットは無視され、24 ビットの分岐先アドレスを格納します。例外処理ベクタテーブルの構造を図 2.4 に示します。
メモリ間接 (@@aa:8) および拡張メモリ間接 (@@vec:7) は、JMP および JSR 命令で使用されます。命令コードに含まれる 8 ビット絶対アドレスによりメモリ上のオペランドを指定し、この内容が分岐先アドレスとなります。
ミドルモードでは、オペランドは 32 ビット (ロングワード) となり、この 32 ビットが分岐先アドレスとなります。このうち上位 8 ビットは予約領域となっており、H'00 と見なされます。
- スタック構造
サブルーチン分岐時の PC のスタック構造と、例外処理時の PC と CCR のスタック構造を図 2.5 に示します。PC は 24 ビットで退避/復帰されます。

【注】 H8SX/1720S グループでは本モードはサポートしません。

2.2.3 アドバンストモード

ミドルモードに対して、データ領域を 4G バイトに拡張しています。

- アドレス空間
プログラム領域最大 16M バイト、データ領域最大 4G バイト、合計最大 4G バイトをリニアにアクセス可能です。
- 拡張レジスタ (En)
拡張レジスタ (E0~E7) は、16 ビットレジスタとして、または 32 ビットレジスタあるいはアドレスレジスタの上位 16 ビットとして使用できます。
- 命令セット
命令およびアドレッシングモードはすべて使用できます。
- 例外処理ベクタテーブル、メモリ間接の分岐アドレス
アドバンストモードでは、H'00000000 から始まる先頭領域に 32 ビット単位で例外処理ベクタテーブル領域が割り当てられており、上位 8 ビットは無視され、24 ビットの分岐先アドレスを格納します。例外処理ベクタテーブルの構造を図 2.4 に示します。

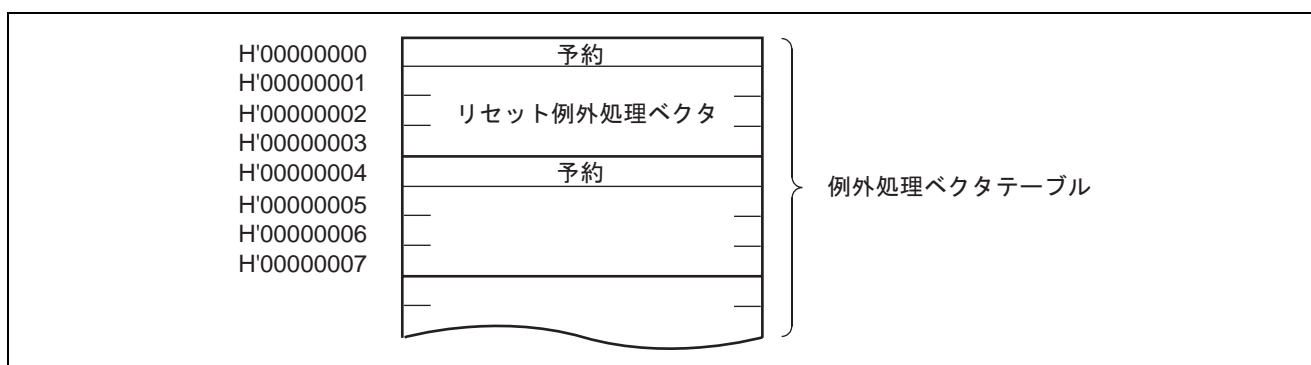


図 2.4 例外処理ベクタテーブル (ミドルモード、アドバンストモード)

メモリ間接 (@@aa:8) および拡張メモリ間接 (@@vec:7) は、JMP および JSR 命令で使用されます。命令コードに含まれる 8 ビット絶対アドレスによりメモリ上のオペランドを指定し、この内容が分岐先アドレスとなります。

アドバンストモードでは、オペランドは 32 ビット (ロングワード) となり、この 32 ビットが分岐先アドレスとなります。このうち上位 8 ビットは予約領域となっており、H'00 と見なされます。

- スタック構造

アドバンストモード時のサブルーチン分岐時の PC のスタック構造と、例外処理時の PC と CCR のスタック構造を図 2.5 に示します。PC は 24 ビットで退避／復帰されます。

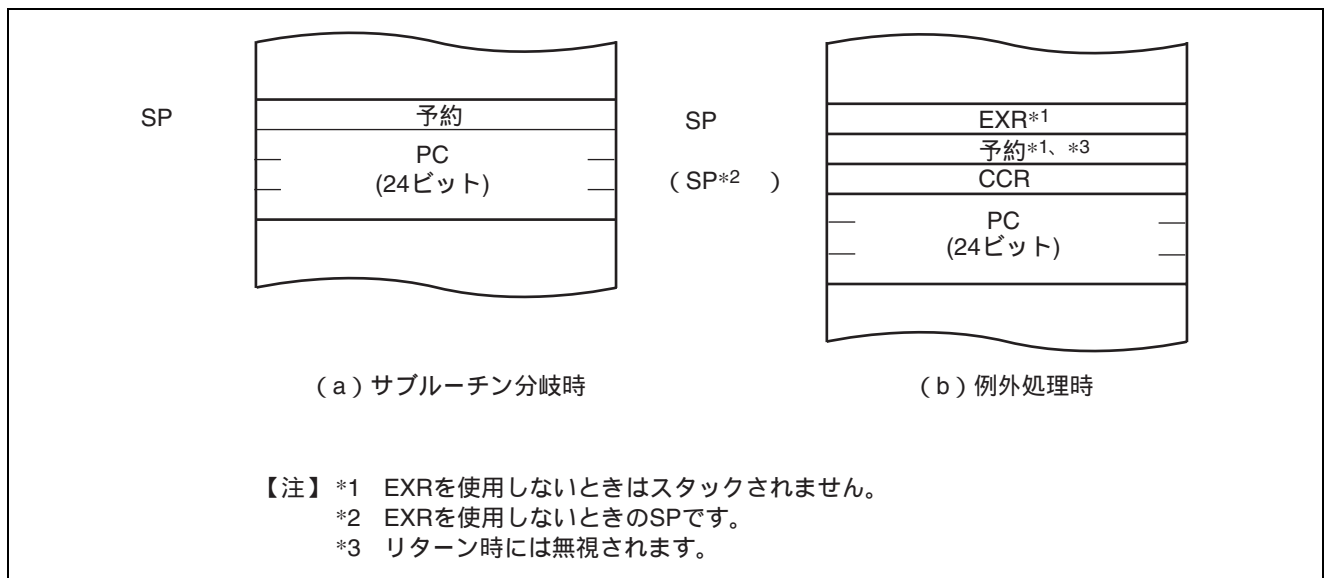


図 2.5 スタック構造（ミドルモード、アドバンストモード）

2.2.4 マキシマムモード

アドバンストモードに対して、プログラム領域を 4G バイトに拡張しています。

- アドレス空間
最大 4G バイトをリニアにアクセス可能です。
- 拡張レジスタ (En)
拡張レジスタ (E0~E7) は、16 ビットレジスタとして、または 32 ビットレジスタあるいはアドレスレジスタの上位 16 ビットとして使用できます。
- 命令セット
命令およびアドレッシングモードはすべて使用できます。
- 例外処理ベクタテーブル、メモリ間接の分岐アドレス
マキシマムモードでは、H'00000000 から始まる先頭領域に 32 ビット単位で例外処理ベクタテーブル領域が割り当てられており、32 ビットの分岐先アドレスを格納します。例外処理ベクタテーブルの構造を図 2.6 に示します。

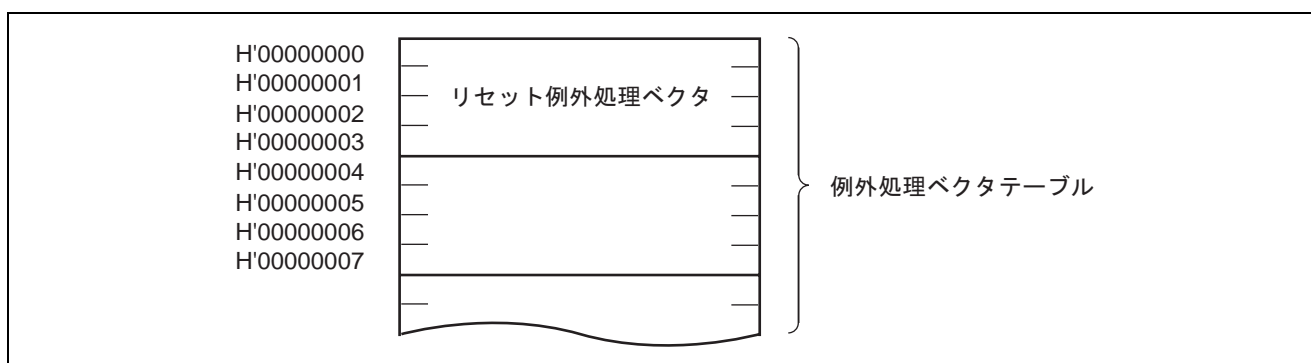


図 2.6 例外処理ベクタテーブル（マキシマムモード）

メモリ間接 (@@aa:8) および拡張メモリ間接 (@@vec:7) は、JMP および JSR 命令で使用されます。命令コードに含まれる 8 ビット絶対アドレスによりメモリ上のオペランドを指定し、この内容が分岐先アドレスとなります。

マキシマムモードでは、オペランドは 32 ビット（ロングワード）となり、この 32 ビットが分岐先アドレスとなります。

- スタック構造

マキシマムモード時のサブルーチン分岐時の PC のスタック構造と、例外処理時の PC と CCR のスタック構造を図 2.7 に示します。PC は 32 ビットで退避／復帰されます。EXR の使用／不使用によらず、EXR は退避／復帰されます。

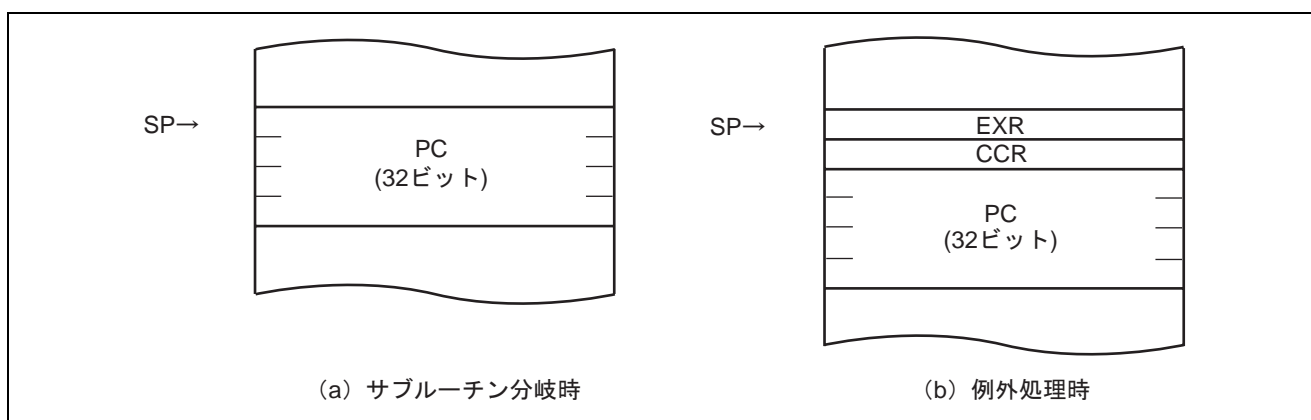


図 2.7 スタック構造（マキシマムモード）

【注】 H8SX/1720S グループでは本モードはサポートしません。

2.3 命令フェッチ

H8SX CPU は、命令フェッチについて、16 ビットモードと 32 ビットモードの 2 つのモードを持っています。プログラムを格納するメモリのバス幅に合わせて設定することを推奨します。

命令フェッチの 16 ビットモード／32 ビットモードの選択は、命令フェッチ以外のデータアクセスなどには影響しません。

【注】 H8SX/1720S グループの命令フェッチは 32 ビットモードです。

2.4 アドレス空間

H8SX CPU のメモリマップを図 2.8 に示します。H8SX CPU のアドレス空間は、CPU 動作モードによって異なります。

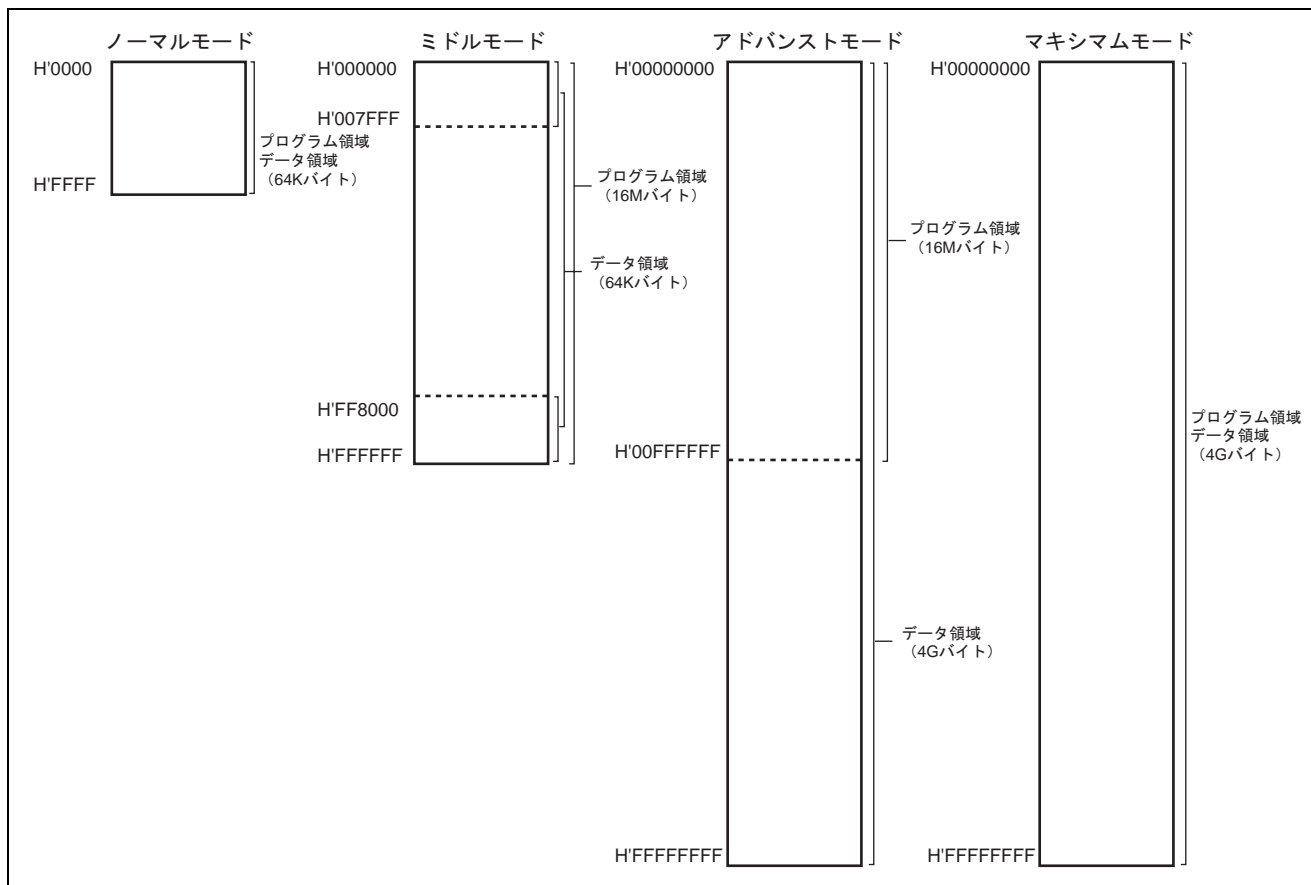


図 2.8 メモリマップ

2.5 レジスタ構成

H8SX CPU の内部レジスタ構成を図 2.9 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。コントロールレジスタには、32 ビットのプログラムカウンタ (PC)、8 ビットのエクステンドレジスタ (EXR)、8 ビットのコンディションコードレジスタ (CCR)、32 ビットのベクタベースレジスタ (VBR)、32 ビットのショートアドレスベースレジスタ (SBR)、および 64 ビットの積和レジスタ (MAC) があります。

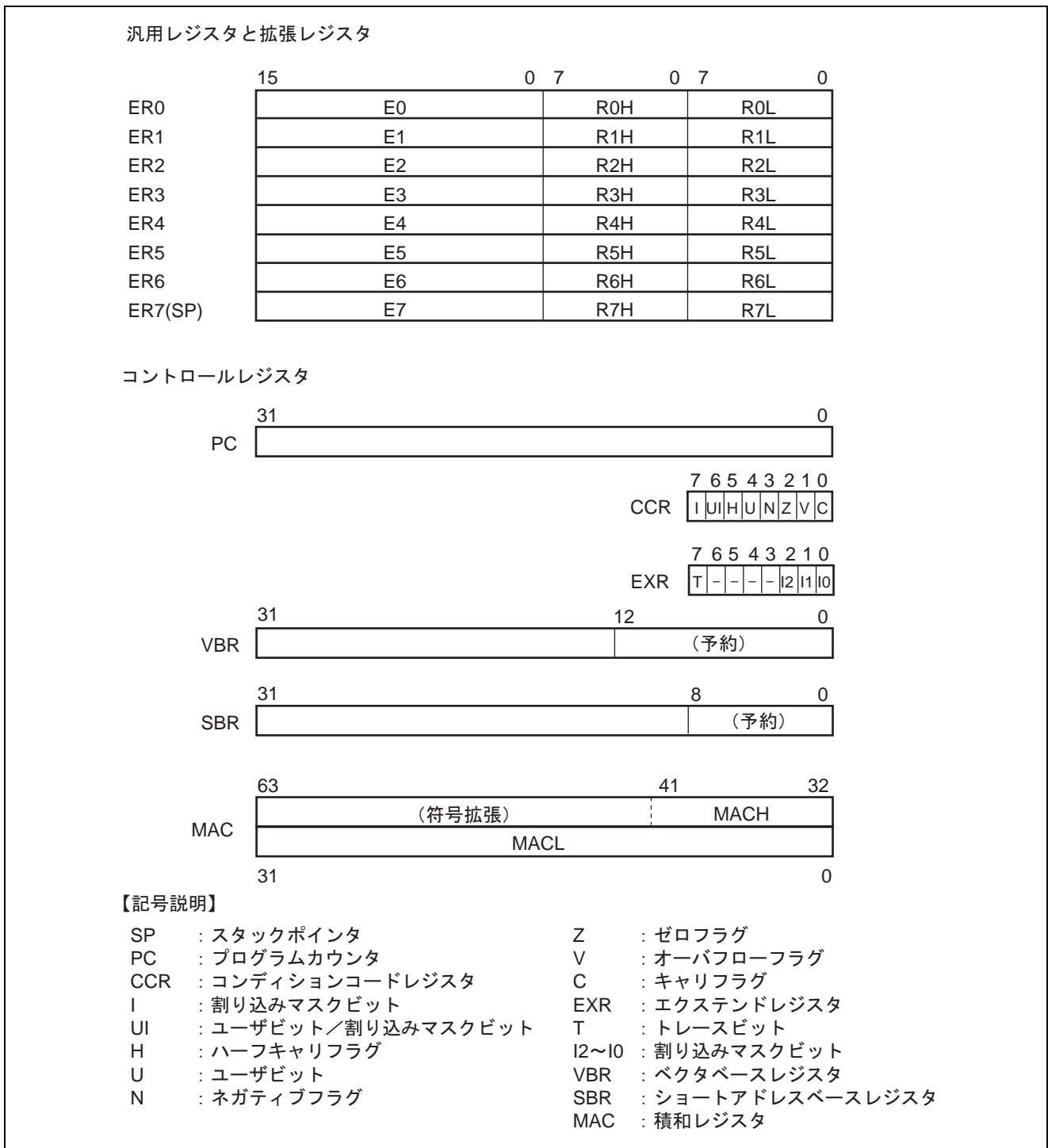


図 2.9 CPU 内部レジスタ構成

2.5.1 汎用レジスタ

H8SX CPU は、32 ビット長の汎用レジスタを 8 本持っています。汎用レジスタは、すべて同じ機能を持っており、アドレスレジスタまたはデータレジスタとして使用することができます。データレジスタとしては 32 ビット、16 ビット、または 8 ビットレジスタとして使用できます。汎用レジスタの使用方法を図 2.10 に示します。

アドレスレジスタまたは 32 ビットレジスタの場合は、一括して汎用レジスタ ER (ER0~ER7) として使用します。

16 ビットレジスタの場合は、汎用レジスタ ER を分割して汎用レジスタ E (E0~E7)、汎用レジスタ R (R0~R7) として使用します。これらは同等の機能を持っており、16 ビットレジスタを最大 16 本まで使用することができます。なお、汎用レジスタ E (E0~E7) を特に拡張レジスタと呼ぶ場合があります。

8 ビットレジスタの場合は、汎用レジスタ R を分割して汎用レジスタ RH (R0H~R7H)、汎用レジスタ RL (R0L~R7L) として使用します。これらは同等の機能を持っており、8 ビットレジスタを最大 16 本まで使用することができます。

インデックスレジスタとしては、汎用レジスタ ER (ER0~ER7)、汎用レジスタ R (R0~R7)、汎用レジスタ RL (R0L~R7L) を使用します。これらはアドレッシングモード中のインデックスレジスタのサイズで指定します。

各レジスタ独立に使用方法を選択することができます。

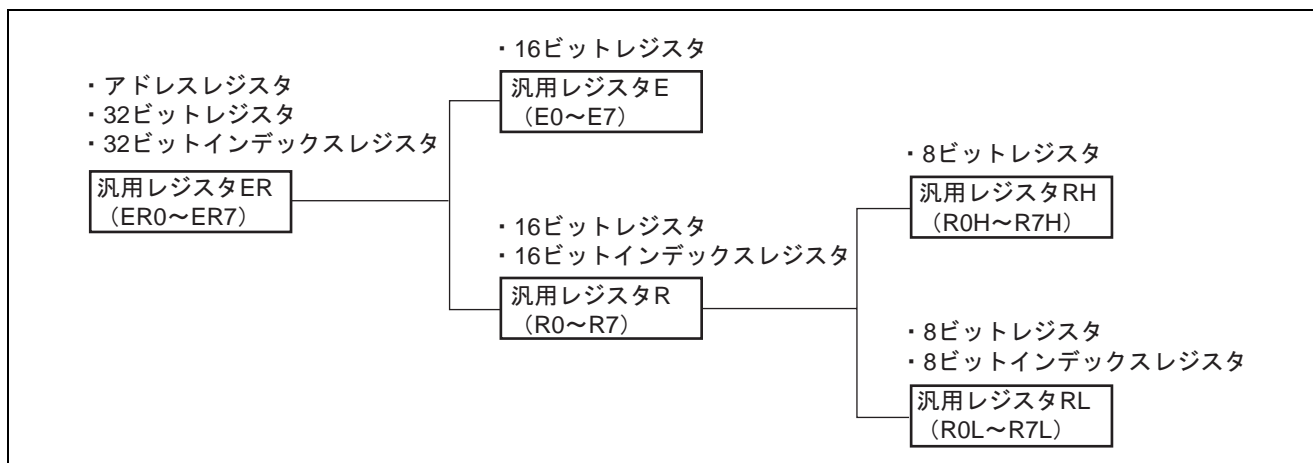


図 2.10 汎用レジスタの使用方法

汎用レジスタ ER7 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分岐などで暗黙的に使用されます。スタックの状態を図 2.11 に示します。

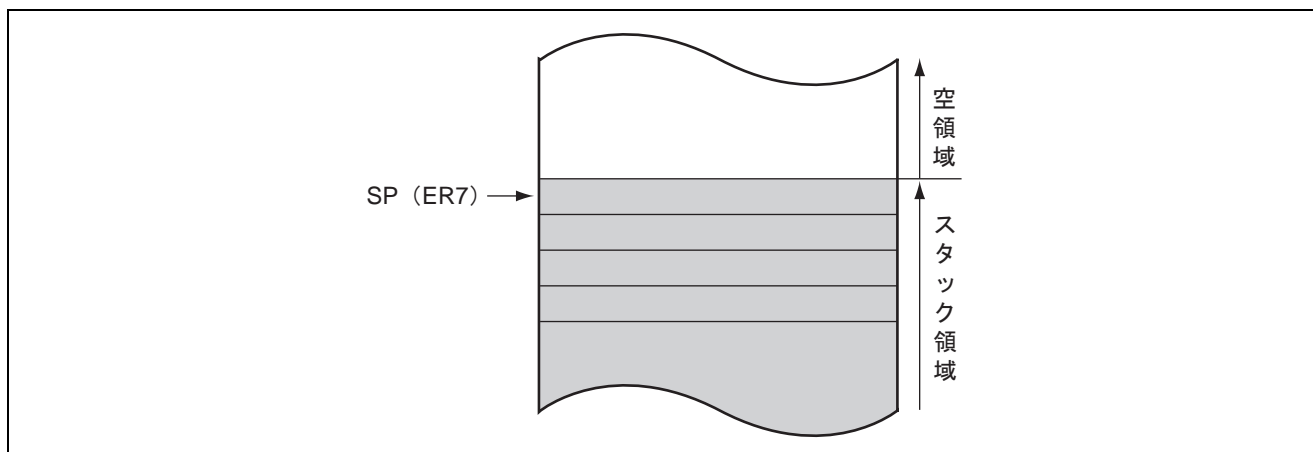


図 2.11 スタックの状態

2.5.2 プログラムカウンタ (PC)

PC は 32 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 2 バイト (ワード) を単位としているため、最下位ビットは無効です。

2.5.3 コンディションコードレジスタ (CCR)

CCR は、8 ビットのレジスタで、CPU の内部状態を示しています。割り込みマスクビット (I)、ユーザビット (UI、U) 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。

CCR は、LDC、STC、ANDC、ORC、XORC 命令で操作することができます。また、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	6	5	4	3	2	1	0
ビット名	I	UI	H	U	N	Z	V	C
初期値:	1	不定	不定	不定	不定	不定	不定	不定
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ビット	ビット名	初期値	R/W	説明
7	I	1	R/W	割り込みマスクビット このビットが 1 にセットされると、割り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。
6	UI	不定	R/W	ユーザビット/割り込みマスクビット ソフトウェア (LDC、STC、ANDC、ORC、XORC 命令) でリード/ライトできます。割り込みマスクビットとしても使用可能です。

ビット	ビット名	初期値	R/W	説明
5	H	不定	R/W	ハーフキャリフラグ 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。また、ADD.W、SUB.W、CMP.W、NEG.W 命令の実行により、ビット 11 にキャリまたはボローが生じたとき、もしくは ADD.L、SUB.L、CMP.L、NEG.L 命令の実行により、ビット 27 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。
4	U	不定	R/W	ユーザビット ソフトウェア (LDC、STC、ANDC、ORC、XORC 命令) でリード/ライトできます。
3	N	不定	R/W	ネガティブフラグ 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。
2	Z	不定	R/W	ゼロフラグ 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。
1	V	不定	R/W	オーバフローフラグ 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。
0	C	不定	R/W	キャリフラグ 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。 <ul style="list-style-type: none"> ・ 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。

2.5.4 エクステンדרレジスタ (EXR)

EXR は、8 ビットのレジスタで、トレースビット (T)、割り込みマスクビット (I2~I0) を含んでいます。

EXR は、LDC、STC、ANDC、ORC、XORC 命令で操作することができます。詳細は「第 4 章 例外処理」を参照してください。

ビット	7	6	5	4	3	2	1	0
ビット名	T	—	—	—	—	I2	I1	I0
初期値:	0	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

ビット	ビット名	初期値	R/W	説明
7	T	0	R/W	トレースビット このビットが1にセットされているときは、1命令実行するごとにトレース例外処理を開始します。0にクリアされているときは、命令を順次実行します。
6~3	—	すべて1	R/W	リザーブビットです。リードすると常に1がリードされます。
2~0	I2	1	R/W	割り込み要求マスクレベル (0~7) を指定します。
	I1	1	R/W	
	I0	1	R/W	

2.5.5 ベクタベースレジスタ (VBR)

VBR は 32 ビットのレジスタで、上位 20 ビットが有効です。リードすると下位 12 ビットは 0 が読み出されます。リセットと CPU アドレスエラー以外の例外処理のベクタ領域のベースアドレスになります (拡張メモリ間接は対象外です)。VBR の初期値は、H'00000000 です。

VBR は、LDC、STC 命令で操作することができます。

2.5.6 ショートアドレスベースレジスタ (SBR)

SBR は 32 ビットのレジスタで、上位 24 ビットが有効です。リードすると下位 8 ビットは 0 が読み出されます。絶対アドレス 8 ビット (@aa:8) 使用時の上位アドレスになります。SBR の初期値は、H'FFFFFF00 です。

SBR は、LDC、STC 命令で操作することができます。

2.5.7 積和レジスタ (MAC)

MAC は 64 ビットのレジスタで、積和演算結果を格納します。32 ビットの MACH、MACL から構成されます。MACH は下位 10 ビットが有効で、上位は符号拡張されています。

MAC は、MAC、CLRMAC、LDMAC、STMAC 命令で操作することができます。

2.5.8 CPU 内部レジスタの初期値

CPU 内部レジスタのうち PC は、リセット例外処理によってベクタアドレスからスタートアドレスをロードすることにより初期化されます。また、EXR の T ビットは 0 にクリアされ、EXR、CCR の I ビットは 1 にセットされますが、汎用レジスタ、MAC、CCR の他のビットは初期化されません。SP (ER7) の初期値も不定です。したがって、リセット直後に MOV.L 命令を使用して、SP の初期化を行ってください。

2.6 データ形式

H8SX 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)、および 32 ビット (ロングワード) 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。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。

2.6.1 汎用レジスタのデータ形式

汎用レジスタのデータ形式を図 2.12 に示します。

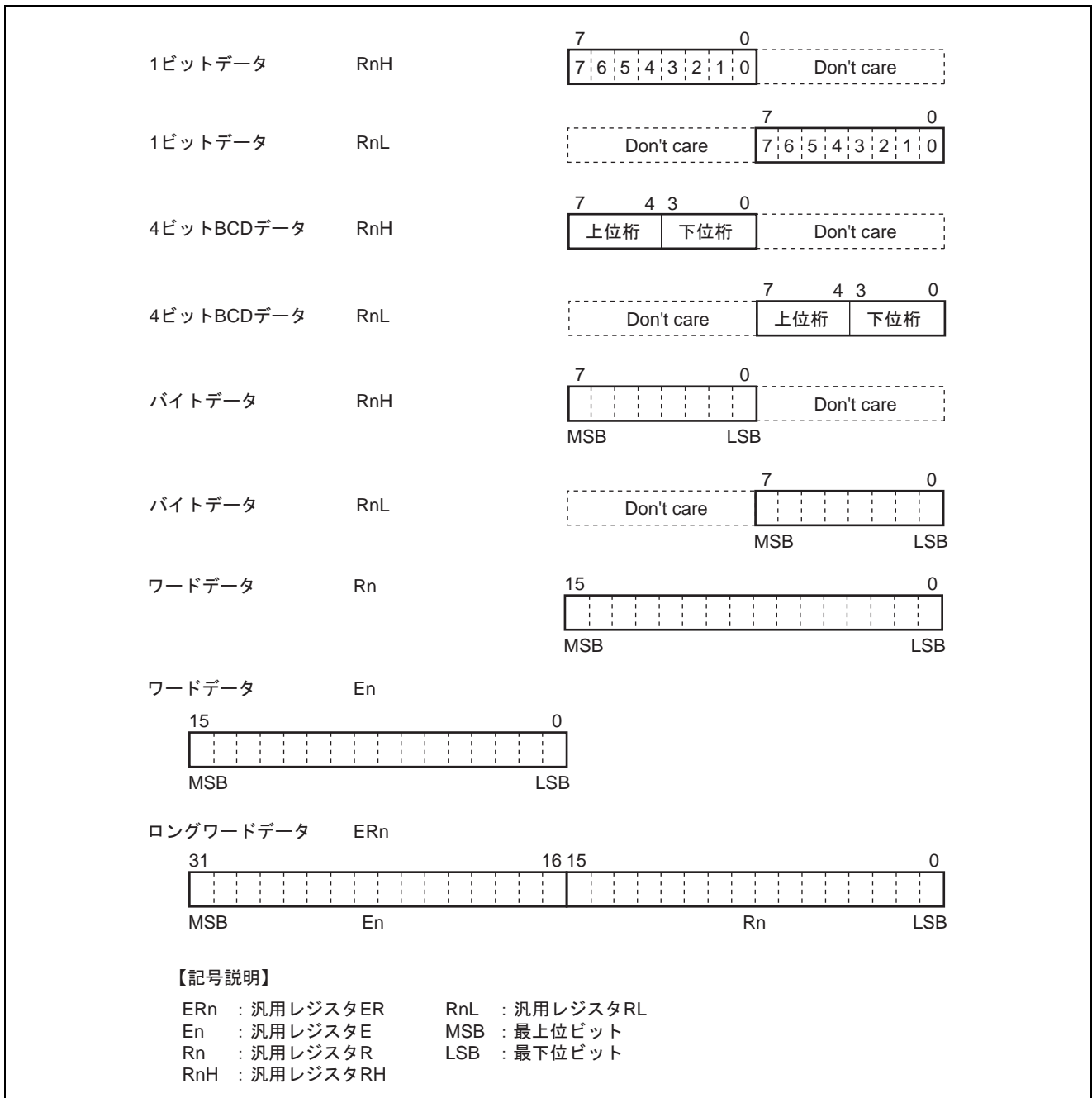


図 2.12 汎用レジスタのデータ形式

2.6.2 メモリ上でのデータ形式

メモリ上でのデータ形式を図 2.13 に示します。

H8SX CPU は、メモリ上のワードデータ/ロングワードデータをアクセスすることができます。これらは、任意のアドレスに配置することができます。ワードデータが偶数番地、ロングワードデータが 4 の倍数番地から始まらない場合は、複数回に分割してアクセスします。例えば、奇数番地から始まる場合、ロングワードデータは、バイト→ワード→バイトに分割されてバスサイクルが生成されます。この場合、バス制御においては別々のバスサイクルとして認識されます。

また、命令リード、スタック操作、分岐テーブル、およびブロック転送命令、MAC 命令におけるワードデータ/ロングワードデータは、偶数番地から配置してください。

なお、SP (ER7) をアドレスレジスタとしてスタックをアクセスするときは、必ずワードサイズまたはロングワードサイズでアクセスしてください。

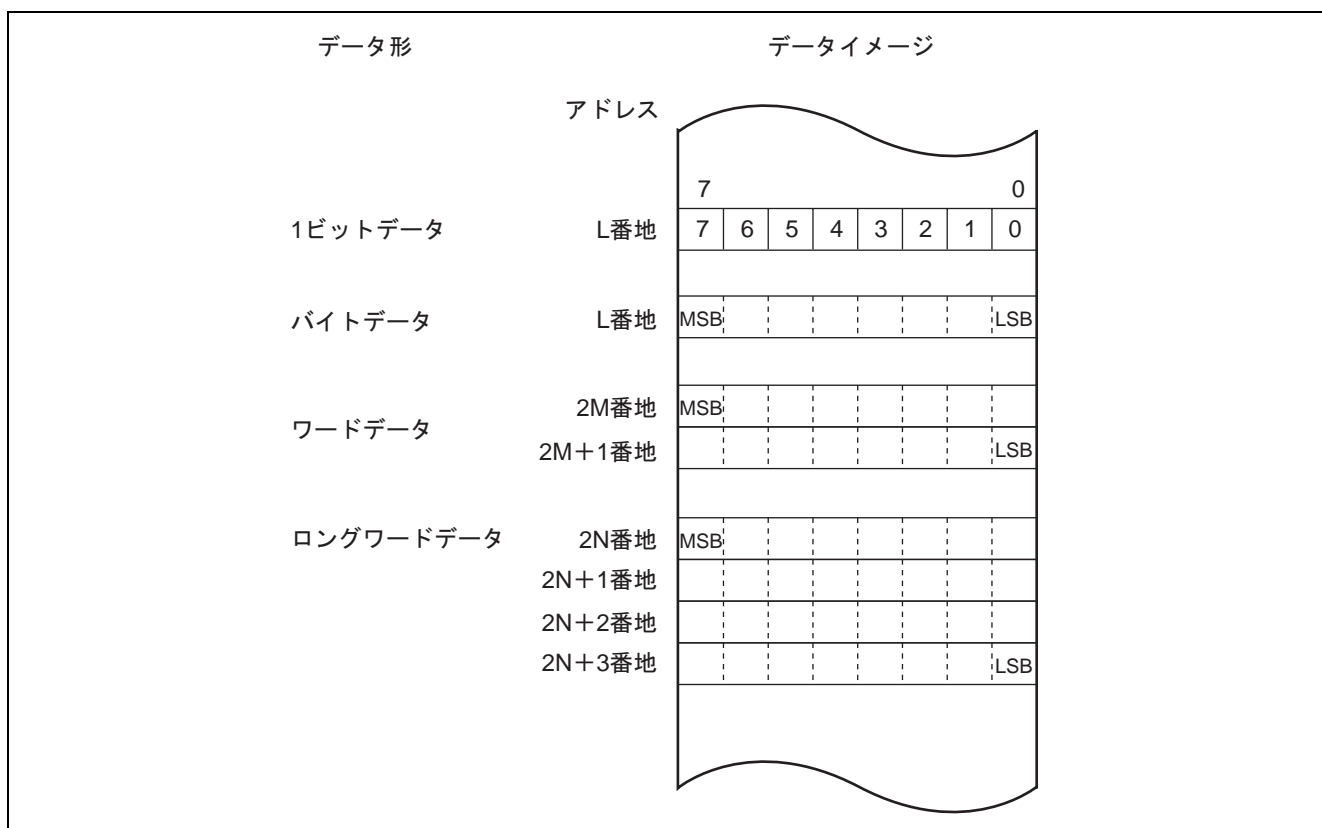


図 2.13 メモリ上でのデータ形式

2.7 命令セット

H8SX CPU の命令は合計 87 種類あり、各命令のもつ機能によって表 2.1 に示すように分類されます。

【注】 算術演算命令、論理演算命令、シフト命令、ビット操作命令を総称して演算命令と呼ぶ場合があります。

表 2.1 命令の分類

分類	命令	サイズ	種類
データ転送命令	MOV	B/W/L	6
	MOVFP* ⁶ 、MOVTP* ⁶	B	
	POP、PUSH* ¹	W/L	
	LDM、STM	L	
	MOVA	B/W* ²	
ブロック転送命令	EEPMOV	B	3
	MOVMD	B/W/L	
	MOVSD	B	
算術演算命令	ADD、ADDX、SUB、SUBX、CMP、NEG、INC、DEC	B/W/L	27
	DAA、DAS	B	
	ADDS、SUBS	L	
	MULXU、DIVXU、MULXS、DIVXS	B/W	
	MULU、DIVU、MULS、DIVS	W/L	
	MULU/U、MULS/U	L	
	EXTU、EXTS	W/L	
	TAS	B	
	MAC	—	
	LDMAC、STMAC	—	
	CLRMAC	—	
論理演算命令	AND、OR、XOR、NOT	B/W/L	4
シフト命令	SHLL、SHLR、SHAL、SHAR、ROTL、ROTR、ROTXL、ROTXR	B/W/L	8
ビット操作命令	BSET、BCLR、BNOT、BTST、BAND、BIAND、BOR、BIOR、BXOR、BIXOR、BLD、BILD、BST、BIST	B	20
	BSET/EQ、BSET/NE、BCLR/EQ、BCLR/NE、BSTZ、BISTZ	B	
	BFLD、BFST	B	
分岐命令	BRA/BS、BRA/BC、BSR/BS、BSR/BC	B* ³	9
	Bcc* ⁴ 、JMP、BSR、JSR、RTS	—	
	RTS/L	L* ⁵	
	BRA/S	—	

分類	命令	サイズ	種類
システム制御命令	TRAPA、RTE、SLEEP、NOP	—	10
	RTE/L	L* ⁵	
	LDC、STC、ANDC、ORC、XORC	B/W/L	
		合計	87

【記号説明】

B : バイトサイズ

W : ワードサイズ

L : ロングワードサイズ

【注】 *1 POP.W Rn、PUSH.W Rn は MOV.W @SP+,Rn、MOV.W Rn, @-SP と同一です。また、POP.L ERn、PUSH.L ERn は MOV.L @SP+,ERn、MOV.L ERn, @-SP と同一です。

*2 ディスプレースメントに加算するデータの指定方法

*3 条件として指定するデータのサイズ

*4 Bcc は条件分岐命令の総称です。

*5 復帰する汎用レジスタのサイズ

*6 本 LSI では使用できません。

2.7.1 命令とアドレッシングモードの組み合わせ

H8SX CPU で使用できる命令とアドレッシングモードの組み合わせを表 2.2 に示します。

表 2.2 命令とアドレッシングモードの組み合わせ (1)

分類	命令	サイズ	アドレッシングモード								
			#xx	Rn	@ERn	@(d, ERn)	@(d, RnL.B/ Rn.W/ ERn.L)	@-ERn/ @ERn+/ @ERn-/ @+ERn	@aa:8	@aa:16/ @aa:32	—
データ 転送命令	MOV	B/W/L	S	SD	SD	SD	SD	SD		SD	
		B		S/D					S/D		
	MOVFPPE,MOVTPPE *12	B		S/D						S/D*1	
	POP,PUSH	W/L		S/D				S/D*2			
	LDM,STM	L		S/D				S/D*2			
	MOVA*4	B/W		S	S	S	S	S		S	
ブロック 転送命令	EPMOV	B									SD*3
	MOVMD	B/W/L									SD*3
	MOVSD	B									SD*3
算術演算命令	ADD,CMP	B	S	D	D	D	D	D	D	D	
		B		S	D	D	D	D	D	D	
		B		D	S	S	S	S	S	S	
		B			SD	SD	SD	SD		SD	
		W/L	S	SD	SD	SD	SD	SD		SD	
	SUB	B	S		D	D	D	D	D	D	
		B		S	D	D	D	D	D	D	
		B		D	S	S	S	S	S	S	
		B			SD	SD	SD	SD		SD	
		W/L	S	SD	SD	SD	SD	SD		SD	
	ADDX,SUBX	B/W/L	S	SD							
		B/W/L	S		SD						
		B/W/L	S					SD*5			
	INC,DEC	B/W/L		D							
	ADDS,SUBS	L		D							
	DAA,DAS	B		D							
MULXU,DIVXU	B/W	S : 4	SD								
MULU,DIVU	W/L	S : 4	SD								
MULXS,DIVXS	B/W	S : 4	SD								
MULS,DIVS	W/L	S : 4	SD								

分類	命令	サイズ	アドレッシングモード									
			#xx	Rn	@ERn	@(d, ERn)	@(d, Rn.L.B/ Rn.W/ ERn.L)	@-ERn/ @ERn+/ @ERn-/ @+ERn	@aa:8	@aa:16/ @aa:32	—	
算術演算命令	NEG	B		D	D	D	D	D	D	D	D	
		W/L		D	D	D	D	D		D		
	EXTU,EXTS	W/L		D	D	D	D	D		D		
	TAS	B			D							
	MAC	—										
	CLRMAC	—										○
	LDMAC	—		S								
	STMAC	—		D								
論理演算命令	AND,OR,XOR	B		S	D	D	D	D	D	D	D	
		B		D	S	S	S	S	S	S	S	
		B			SD	SD	SD	SD		SD		
		W/L	S	SD	SD	SD	SD	SD		SD		
	NOT	B		D	D	D	D	D	D	D	D	
		W/L		D	D	D	D	D		D		
シフト命令	SHLL,SHLR	B		D	D	D	D	D	D	D		
		W/L ^{*6}		D	D	D	D	D		D		
		B/W/L ^{*7}		D								
	SHAL,SHAR,ROTL, ROTR,ROTXL, ROTXR	B		D	D	D	D	D	D	D		
		W/L		D	D	D	D	D		D		
ビット 操作命令	BSET,BCLR,BNOT, BTST,BSET/cc, BCLR/cc	B		D	D				D	D		
		B		D	D				D	D		
	BAND,BIAND,BOR, BIOR,BXOR, BIXOR, BLD,BILD, BST,BIST,BSTZ, BISTZ	BFLD	B		D	S				S	S	
		BFST	B		S	D				D	D	
		分岐命令	BRA/BS,BRA/BC ^{*8}	B			S				S	S
BSR/BS,BSR/BC ^{*8}	B				S				S	S		

分類	命令	サイズ	アドレッシングモード								
			#xx	Rn	@ERn	@(d, ERn)	@(d, Rn.L.B/ Rn.W/ ERn.L)	@-ERn/ @ERn+/ @ERn-/ @+ERn	@aa:8	@aa:16/ @aa:32	—
システム制御 命令	LDC (CCR,EXR)	BW* ⁹	S	S	S	S		S* ¹⁰		S	
	LDC (VBR,SBR)	L		S							
	STC (CCR,EXR)	BW* ⁹		D	D	D		D* ¹¹		D	
	STC (VBR,SBR)	L		D							
	ANDC,ORC,XORC	B	S								
	SLEEP	—									○
	NOP	—									○

【記号説明】

- d: d: 16 または d: 32
- S: ソースとして指定可能
- D: デスティネーションとして指定可能
- SD: ソースまたはデスティネーションの一方または両方として指定可能
- S/D: ソースまたはデスティネーションの一方として指定可能
- S:4: ソースとして4ビットイミディエイトデータを指定可能

【注】

- *1 @aa:16 のみ
- *2 ソースのとき@ERn+、デスティネーションのとき@-ERn
- *3 データ転送のソースのアドレスはER5、デスティネーションのアドレスはER6
- *4 ディスプレースメントに加算するデータの指定方法
- *5 @ERn-のみ
- *6 シフトビット数が1、2、4、8または16ビットのとき
- *7 シフトビット数を5ビットイミディエイトデータまたは汎用レジスタで指定するとき
- *8 条件として指定するデータの指定方法
- *9 イミディエイトまたはレジスタ直接のときバイト、そのほかのときワード
- *10 @ERn+のみ
- *11 @-ERnのみ
- *12 本LSIでは使用できません。

表 2.2 命令とアドレッシングモードの組み合わせ (2)

分類	命令	サイズ	アドレッシングモード							
			@ERn	@(d,PC)	@(RnL.B/ Rn.W/ ERn.L,PC)	@aa:24	@aa:32	@@aa:8	@@vec:7	—
分岐命令	BRA/BS、BRA/BC	—		○						
	BSR/BS、BSR/BC	—		○						
	Bcc	—		○						
	BRA	—		○	○					
	BRA/S	—		○*						
	JMP	—	○			○	○	○	○	
	BSR	—		○						
	JSR	—	○			○	○	○	○	
	RTS、RTS/L	—								○
システム制御 命令	TRAPA	—								○
	RTE、RTE/L	—								○

【記号説明】

d: d:8 または d:16

【注】 * @(d:8,PC)のみ

2.7.2 命令の機能別一覧

各命令の機能を表 2.4～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を表 2.3 に示します。

表 2.3 オペレーションの記号

記号	説明
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*
ERn	汎用レジスタ（32 ビットレジスタ）
(EAd)	デスティネーションオペランド
(EAs)	ソースオペランド
EXR	エクステンドレジスタ
CCR	コンディションコードレジスタ
VBR	ベクタベースレジスタ
SBR	ショートアドレスベースレジスタ
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）
:8/:16/:24/:32	8/16/24/32 ビット長

【注】 * 汎用レジスタは、8 ビット（R0H～R7H、R0L～R7L）、16 ビット（R0～R7、E0～E7）、または 32 ビットレジスタ（ER0～ER7）です。

表 2.4 データ転送命令

命令	サイズ	機能
MOV	B/W/L	#IMM→(EAd)、(EAs)→(EAd) イミディエイトデータ、汎用レジスタ、またはメモリの間でデータ転送を行います。
MOVFPPE*	B	(EAs)→Rd
MOVTPPE*	B	Rs→(EAs)
POP	W/L	@SP+→Rn スタックから汎用レジスタへその内容を復帰します。
PUSH	W/L	Rn→@-SP 汎用レジスタの内容をスタックに退避します。
LDM	L	@SP+→Rn (レジスタ群) スタックから複数の汎用レジスタへその内容を復帰します。連続した番号の2、3または4本の汎用レジスタを指定できます。
STM	L	Rn (レジスタ群)→@-SP 複数の汎用レジスタの内容をスタックに退避します。連続した番号の2、3または4本の汎用レジスタを指定できます。
MOVA	B/W	EA→Rd 指定した汎用レジスタまたはメモリのデータをゼロ拡張およびシフトして、ディスプレイメントに加算し、結果を汎用レジスタに格納します。

【注】 * 本 LSI では使用できません。

表 2.5 ブロック転送命令

命令	サイズ	機能
EEPMOV.B EEPMOV.W	B	ブロック転送命令です。ER5で示されるアドレスから始まり、R4またはR4Lで指定される数のバイトデータを、ER6で示されるアドレスへ転送します。
MOVMD.B	B	ブロック転送命令です。ER5で示されるアドレスから始まり、R4で指定される数のバイトデータを、ER6で示されるアドレスへ転送します。
MOVMD.W	W	ブロック転送命令です。ER5で示されるアドレスから始まり、R4で指定される数のワードデータを、ER6で示されるアドレスへ転送します。
MOVMD.L	L	ブロック転送命令です。ER5で示されるアドレスから始まり、R4で指定される数のロングワードデータを、ER6で示されるアドレスへ転送します。
MOVSD.B	B	ゼロデータ検出付きのブロック転送命令です。ER5で示されるアドレスから始まり、R4で指定される数のバイトデータを、ER6で示されるアドレスへ転送します。途中でゼロデータが検出されると、データ転送を打ち切り、指定したアドレスに分岐します。

表 2.6 算術演算命令

命令	サイズ	機能
ADD SUB	B/W/L	(EAd) ±#IMM → (EAd) 、 (EAd) ± (EAs) → (EAd) イミディエイトデータ、汎用レジスタ、またはメモリの間でデータの加減算を行います。バイトサイズでの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。
ADDX SUBX	B/W/L	(EAd) ±#IMM±C → (EAd) 、 (EAd) ± (EAs) ±C → (EAd) イミディエイトデータ、汎用レジスタ、またはメモリの間でデータのキャリ付きの加減算を行います。メモリを指定するアドレッシングモードは、レジスタ間接、ポストデクリメントレジスタ間接です。
INC DEC	B/W/L	Rd±1→Rd、Rd±2→Rd 汎用レジスタに 1 または 2 を加減算します (バイトサイズでは 1 の加減算のみ可能です)。
ADDS SUBS	L	Rd±1→Rd、Rd±2→Rd、Rd±4→Rd 汎用レジスタに 1、2 または 4 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 2 桁 BCD データに補正します。
MULXU	B/W	Rd×Rs→Rd 汎用レジスタと汎用レジスタの間の符号なし乗算を行います。 8 ビット×8 ビット→16 ビット、16 ビット×16 ビット→32 ビットの乗算が可能です。
MULU	W/L	Rd×Rs→Rd 汎用レジスタと汎用レジスタの間の符号なし乗算を行います。 16 ビット×16 ビット→16 ビット、32 ビット×32 ビット→32 ビットの乗算が可能です。
MULU/U	L	Rd×Rs→Rd 汎用レジスタと汎用レジスタの間の 32 ビット×32 ビットの符号なし乗算を行い、結果の上位 32 ビットを得ます。
MULXS	B/W	Rd×Rs→Rd 汎用レジスタと汎用レジスタの間の符号付き乗算を行います。 8 ビット×8 ビット→16 ビット、16 ビット×16 ビット→32 ビットの乗算が可能です。
MULS	W/L	Rd×Rs→Rd 汎用レジスタと汎用レジスタの間の符号付き乗算を行います。 16 ビット×16 ビット→16 ビット、32 ビット×32 ビット→32 ビットの乗算が可能です。
MULS/U	L	Rd×Rs→Rd 汎用レジスタと汎用レジスタの間の 32 ビット×32 ビットの符号付き乗算を行い、結果の上位 32 ビットを得ます。
DIVXU	B/W	Rd÷Rs→Rd 汎用レジスタと汎用レジスタの間の符号なし除算を行います。 16 ビット÷8 ビット→商 8 ビット余り 8 ビット、32 ビット÷16 ビット→商 16 ビット余り 16 ビットの除算が可能です。

命令	サイズ	機能
DIVU	W/L	$Rd \div Rs \rightarrow Rd$ 汎用レジスタと汎用レジスタの間の符号なし除算を行います。 16ビット÷16ビット→商16ビット、32ビット÷32ビット→商32ビットの除算が可能です。
DIVXS	B/W	$Rd \div Rs \rightarrow Rd$ 汎用レジスタと汎用レジスタの間の符号付き除算を行います。 16ビット÷8ビット→商8ビット余り8ビット、32ビット÷16ビット→商16ビット余り16ビットの除算が可能です。
DIVS	W/L	$Rd \div Rs \rightarrow Rd$ 汎用レジスタと汎用レジスタの間の符号付き除算を行います。 16ビット÷16ビット→商16ビット、32ビット÷32ビット→商32ビットの除算が可能です。
CMP	B/W/L	(EAd) - #IMM、(EAd) - (EAs) イミディエイトデータ、汎用レジスタ、またはメモリの間でデータの比較を行い、その結果をCCRに格納します。
NEG	B/W/L	$0 - (EAd) \rightarrow (EAd)$ 汎用レジスタまたはメモリの内容の2の補数（算術的補数）を取ります。
EXTU	W/L	(EAd) (ゼロ拡張) → (EAd) 汎用レジスタまたはメモリの下位8ビットまたは16ビットを、ワードサイズまたはロングワードサイズにゼロ拡張します。 下位8ビット→ワードサイズ、下位8ビット→ロングワードサイズ、下位16ビット→ロングワードサイズのゼロ拡張が可能です。
EXTS	W/L	(EAd) (符号拡張) → (EAd) 汎用レジスタまたはメモリの下位8ビットまたは16ビットを、ワードサイズまたはロングワードサイズに符号拡張します。 下位8ビット→ワードサイズ、下位8ビット→ロングワードサイズ、下位16ビット→ロングワードサイズの符号拡張が可能です。
TAS	B	@ERd-0, 1 → (<ビット7>of@EAd) メモリの内容をテストした後、最上位ビット（ビット7）を1にセットします。
MAC	-	(EAd) × (EAs) + MAC → MAC メモリとメモリの間の符号付乗算を行い、結果をMACに加算します。
CLRMAC	-	0 → MAC MACをゼロクリアします。
LDMAC	-	Rs → MAC 汎用レジスタの内容をMACに転送します。
STMAC	-	MAC → Rd MACの内容を汎用レジスタに転送します。

表 2.7 論理演算命令

命令	サイズ	機能
AND	B/W/L	$(EAd) \wedge \#IMM \rightarrow (EAd)$ 、 $(EAd) \wedge (EAs) \rightarrow (EAd)$ イミディエイトデータ、汎用レジスタ、またはメモリの間でデータの論理積を取ります。
OR	B/W/L	$(EAd) \vee \#IMM \rightarrow (EAd)$ 、 $(EAd) \vee (EAs) \rightarrow (EAd)$ イミディエイトデータ、汎用レジスタ、またはメモリの間でデータの論理和を取ります。
XOR	B/W/L	$(EAd) \oplus \#IMM \rightarrow (EAd)$ 、 $(EAd) \oplus (EAs) \rightarrow (EAd)$ イミディエイトデータ、汎用レジスタ、またはメモリの間でデータの排他的論理和を取ります。
NOT	B/W/L	$\sim (EAd) \rightarrow (EAd)$ 汎用レジスタまたはメモリの内容の1の補数（論理的補数）を取ります。

表 2.8 シフト命令

命令	サイズ	機能
SHLL SHLR	B/W/L	(EAd) （シフト処理） $\rightarrow (EAd)$ 汎用レジスタまたはメモリの内容を論理的にシフトします。汎用レジスタまたはメモリに対して1、2、4、8または16ビットのシフトが可能です。汎用レジスタに対しては、任意ビットのシフトが可能です。この場合、シフトビット数は、5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5ビットで指定します。
SHAL SHAR	B/W/L	(EAd) （シフト処理） $\rightarrow (EAd)$ 汎用レジスタまたはメモリの内容を算術的にシフトします。1または2ビットのシフトが可能です。
ROTL ROTR	B/W/L	(EAd) （ローテート処理） $\rightarrow (EAd)$ 汎用レジスタまたはメモリの内容をローテートします。1または2ビットのシフトが可能です。
ROTXL ROTXR	B/W/L	(EAd) （ローテート処理） $\rightarrow (EAd)$ 汎用レジスタまたはメモリの内容をキャリフラグを含めてローテートします。1または2ビットのシフトが可能です。

表 2.9 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ビットで指定します。
BSET/cc	B	if cc, 1→ (<ビット番号>of<EAd>) 指定した条件が成立しているとき、メモリの指定された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ビットで指定します。条件は、Z=1、Z=0が指定可能です。
BCLR	B	0→ (<ビット番号>of<EAd>) 汎用レジスタまたはメモリの指定された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ビットで指定します。
BCLR/cc	B	if cc, 0→ (<ビット番号>of<EAd>) 指定した条件が成立しているとき、メモリの指定された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ビットで指定します。条件は、Z=1、Z=0が指定可能です。
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ビットで指定します。
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ビットで指定します。
BAND	B	C ∧ (<ビット番号>of<EAd>) → C 汎用レジスタまたはメモリの指定された1ビットとキャリフラグとの論理積をとり、結果をキャリフラグに格納します。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、結果をキャリフラグに格納します。ビット番号は、3ビットのイミディエイトデータで指定します。
BOR	B	C ∨ (<ビット番号>of<EAd>) → C 汎用レジスタまたはメモリの指定された1ビットとキャリフラグとの論理和をとり、結果をキャリフラグに格納します。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、結果をキャリフラグに格納します。ビット番号は、3ビットのイミディエイトデータで指定します。
BXOR	B	C ⊕ (<ビット番号>of<EAd>) → C 汎用レジスタまたはメモリの指定された1ビットとキャリフラグとの排他的論理和をとり、結果をキャリフラグに格納します。ビット番号は、3ビットのイミディエイトデータで指定します。

命令	サイズ	機能
BIXOR	B	$C \oplus [\sim (\text{ビット番号} \text{ of } \text{EAd})] \rightarrow C$ 汎用レジスタまたはメモリの指定された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、結果をキャリフラグに格納します。ビット番号は、3ビットのイミディエイトデータで指定します。
BLD	B	$(\text{ビット番号} \text{ of } \text{EAd}) \rightarrow C$ 汎用レジスタまたはメモリの指定された1ビットを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します。
BILD	B	$\sim (\text{ビット番号} \text{ of } \text{EAd}) \rightarrow C$ 汎用レジスタまたはメモリの指定された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します。
BST	B	$C \rightarrow (\text{ビット番号} \text{ of } \text{EAd})$ 汎用レジスタまたはメモリの指定された1ビットに、キャリフラグの内容を転送します。ビット番号は、3ビットのイミディエイトデータで指定します。
BSTZ	B	$Z \rightarrow (\text{ビット番号} \text{ of } \text{EAd})$ メモリの指定された1ビットに、ゼロ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します。
BIST	B	$\sim C \rightarrow (\text{ビット番号} \text{ of } \text{EAd})$ 汎用レジスタまたはメモリの指定された1ビットに、キャリフラグを反転して転送します。ビット番号は、3ビットのイミディエイトデータで指定します。
BISTZ	B	$\sim Z \rightarrow (\text{ビット番号} \text{ of } \text{EAd})$ メモリの指定された1ビットに、ゼロフラグを反転して転送します。ビット番号は、3ビットのイミディエイトデータで指定します。
BFLD	B	$(\text{EAs}) (\text{ビットフィールド}) \rightarrow \text{Rd}$ メモリのオペランドの指定したビットフィールドを、指定した汎用レジスタの下位側に転送します。
BFST	B	$\text{Rs} \rightarrow (\text{EAd}) (\text{ビットフィールド})$ メモリのオペランドの指定したビットフィールドへ、指定した汎用レジスタの下位側を転送します。

表 2.10 分岐命令

命令	サイズ	機能
BRA/BS BRA/BC	B	メモリのオペランドの指定したビットをテストし、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。
BSR/BS BSR/BC	B	メモリのオペランドの指定したビットをテストし、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サブルーチン分岐します。
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。
BRA/S	—	次の命令を実行した後、指定されたアドレスに無条件に分岐します。次の命令は1ワード命令で、かつブロック転送命令、分岐命令以外に限定されます。
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。
RTS/L	—	スタックから複数の汎用レジスタへその内容を復帰し、サブルーチンから復帰します。

表 2.11 システム制御命令

命令	サイズ	機能
TRAPA	—	命令トラップ例外処理を行います。
RTE	—	例外処理ルーチンから復帰します。
RTE/L	—	スタックから複数の汎用レジスタへその内容を復帰し、例外処理ルーチンから復帰します。
SLEEP	—	低消費電力状態に遷移します。
LDC	B/W	#IMM→CCR、(EAs)→CCR、#IMM→EXR、(EAs)→EXR イミディエイトデータ、汎用レジスタまたはメモリの内容を CCR、EXR へ転送します。CCR、EXR は 8 ビットですが、メモリとの転送はワードサイズで行われ、上位 8 ビットが有効です。
	L	Rs→VBR、Rs→SBR 汎用レジスタの内容を VBR、SBR へ転送します。
STC	B/W	CCR→(EAd)、EXR→(EAd) CCR、EXR の内容を汎用レジスタまたはメモリへ転送します。 CCR、EXR は 8 ビットですが、メモリとの転送はワードサイズで行われ、上位 8 ビットが有効です。
	L	VBR→Rd、SBR→Rd VBR、SBR の内容を汎用レジスタへ転送します。
ANDC	B	CCR∧#IMM→CCR、EXR∧#IMM→EXR CCR、EXR 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、EXR∨#IMM→EXR CCR、EXR 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、EXR⊕#IMM→EXR CCR、EXR 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。

2.7.3 命令の基本フォーマット

H8SX CPU の命令は、2 バイト (ワード) を単位にしています。各命令はオペレーションフィールド (op)、レジスタフィールド (r)、EA 拡張部 (EA)、およびコンディションフィールド (cc) から構成されています。

図 2.14 に命令フォーマットの例を示します。

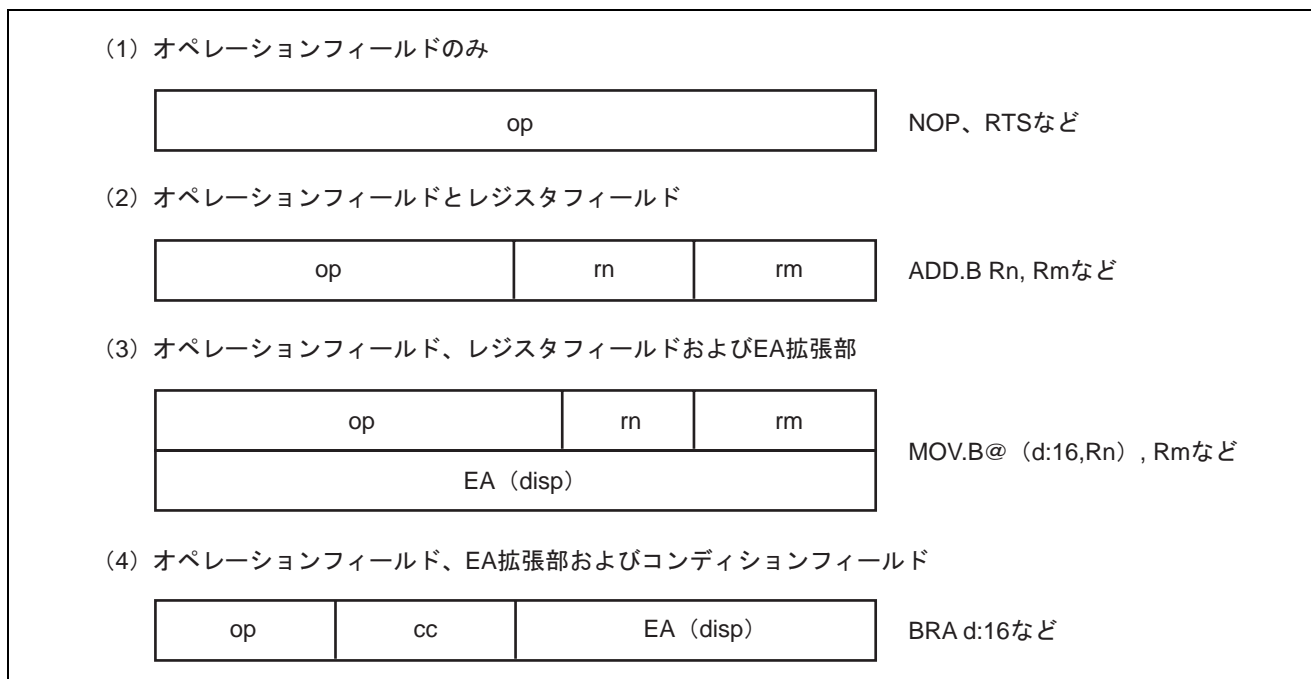


図 2.14 命令フォーマットの例

- オペレーションフィールド
 命令の機能を表し、アドレッシングモードの指定、オペランドの処理内容を指定します。命令の先頭 4 ビットを必ず含みます。2つのオペレーションフィールドを持つ場合もあります。
- レジスタフィールド
 汎用レジスタを指定します。アドレスレジスタのとき 3 ビット、データレジスタのとき 3 ビットまたは 4 ビットです。2つのレジスタフィールドを持つ場合、またはレジスタフィールドを持たない場合もあります。
- EA拡張部
 イミディエイトデータ、絶対アドレスまたはディスプレースメントを指定します。8 ビット、16 ビット、または 32 ビットです。
- コンディションフィールド
 Bcc 命令の分岐条件を指定します。

2.8 アドレッシングモードと実効アドレスの計算方法

H8SX CPU は表 2.12 に示すように、11 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。

ビット操作命令では、オペランドの指定にレジスタ直接、レジスタ間接、および絶対アドレスが使用できます。さらに、オペランド中のビット番号を指定するためにレジスタ直接（BSET、BCLR、BNOT、BTST の各命令）、およびイミディエイト（3 ビット）が独立して使用できます。

表 2.12 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	@ERn
3	ディスプレースメント付レジスタ間接	@(d:2,ERn)/@(d:16,ERn)/@(d:32,ERn)
4	ディスプレースメント付インデックスレジスタ間接	@(d:16,RnL.B)/@(d:16,Rn.W)/@(d:16,ERn.L) @(d:32,RnL.B)/@(d:32,Rn.W)/@(d:32,ERn.L)
5	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 プリインクリメントレジスタ間接 ポストデクリメントレジスタ間接	@ERn+ @-ERn @+ERn @ERn-
6	絶対アドレス	@aa:8/@aa:16/@aa:24/@aa:32
7	イミディエイト	#xx:3/#xx:4/#xx:8/#xx:16/#xx:32
8	プログラムカウンタ相対	@(d:8,PC)/@(d:16,PC)
9	プログラムカウンタインデックスレジスタ相対	@(RnL.B,PC)/@(Rn.W,PC)/@(ERn.L,PC)
10	メモリ間接	@@aa:8
11	拡張メモリ間接	@@ vec:7

2.8.1 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、16 ビットまたは 32 ビット）が、オペランドとなります。

8 ビットレジスタとしては、R0H～R7H、R0L～R7L を指定可能です。

16 ビットレジスタとしては、R0～R7、E0～E7 を指定可能です。

32 ビットレジスタとしては、ER0～ER7 を指定可能です。

2.8.2 レジスタ間接 @ERn

命令コードのレジスタフィールドで指定されるアドレスレジスタ (ERn) 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。

アドバンスモードのとき、分岐命令では下位 24 ビットが有効になり、上位 8 ビットはすべて 0 (H'00) と見なされます。

2.8.3 ディスプレースメント付レジスタ間接 @ (d:2,ERn) / @ (d:16,ERn) / @ (d:32,ERn)

命令コードのレジスタフィールドで指定されるアドレスレジスタ (ERn) の内容に、命令コード中に含まれる 16 ビットディスプレースメント、または 32 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。加算に際して、16 ビットディスプレースメントは符号拡張されます。

また、データのサイズがバイトの場合は、ディスプレースメントが 1、2、3 のとき短縮形 @ (d:2,ERn) が用意されています。同様に、ワードの場合は 2、4、6 のとき、ロングワードの場合は 4、8、12 のとき、それぞれ短縮形 @ (d:2,ERn) が用意されています。

2.8.4 ディスプレースメント付インデックスレジスタ間接 @(d:16,RnL.B)/ @(d:32,RnL.B)/@(d:16,Rn.W)/@(d:32,Rn.W)/@(d:16,ERn.L)/@(d:32,ERn.L)

命令コードのレジスタフィールドで指定されるアドレスレジスタの指定されたビット数 (RnL、Rn、ERn) の内容を 32 ビットにゼロ拡張し、1、2 または 4 を乗算し、乗算結果と命令コード中に含まれる 16 ビットディスプレースメント、または 32 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。指定するメモリ上のオペランドサイズに応じて、バイトサイズでは 1、ワードサイズでは 2、ロングワードサイズでは 4 が、それぞれ乗算されます。加算に際して、16 ビットディスプレースメントは符号拡張されます。

2.8.5 プリ/ポストインクリメント/デクリメントレジスタ間接 @ERn+ / @-ERn / @+ERn / @ERn-

(1) ポストインクリメントレジスタ間接 @ERn+

命令コードのレジスタフィールドで指定されるアドレスレジスタ (ERn) 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、アドレスレジスタの内容に 1、2 または 4 が加算され、加算結果がアドレスレジスタに格納されます。バイトサイズでは 1、ワードサイズでは 2、ロングワードサイズでは 4 が、それぞれ加算されます。

(2) プリデクリメントレジスタ間接 @-ERn

命令コードのレジスタフィールドで指定されるアドレスレジスタ (ERn) の内容から、1、2 または 4 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、減算結果がアドレスレジスタに格納されます。バイトサイズでは 1、ワードサイズでは 2、ロングワードサイズでは 4 が、それぞれ減算されます。

(3) プリインクリメントレジスタ間接 @+ERn

命令コードのレジスタフィールドで指定されるアドレスレジスタ (ERn) の内容から、1、2 または 4 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。その後、加算結果がアドレスレジスタに格納されます。バイトサイズでは 1、ワードサイズでは 2、ロングワードサイズでは 4 が、それぞれ加算されます。

(4) ポストデクリメントレジスタ間接 @ERn-

命令コードのレジスタフィールドで指定されるアドレスレジスタ (ERn) 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、アドレスレジスタの内容に 1、2 または 4 が減算され、減算結果がアドレスレジスタに格納されます。バイトサイズでは 1、ワードサイズでは 2、ロングワードサイズでは 4 が、それぞれ減算されます。

以上の(1)~(4)の場合、アドレスレジスタで指定した汎用レジスタをデータレジスタとして、メモリへのライトを行う場合、実効アドレス計算後の汎用レジスタの内容がライトされます。また、同一の汎用レジスタを指定して 2 つの実効アドレス計算を行う場合、1 回目の実効アドレス計算後の汎用レジスタの内容が、2 回目の実効アドレス計算に用いられます。

例 1 MOV.W R0, @ER0+

実行前の ER0 が H'12345678 のとき H'12345678 番地に H'567A がライトされます。

例 2 MOV.B @ER0+, @ER0+

実行前の ER0 が H'00001000 のとき H'00001000 番地をリードし、H'00001001 番地へライトします。

実行後の ER0 は H'00001002 です。

2.8.6 絶対アドレス @aa:8/@aa:16/@aa:24/@aa:32

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。

絶対アドレスは 8 ビット (@aa:8)、16 ビット (@aa:16)、24 ビット (@aa:24)、または 32 ビット (@aa:32) です。

データ領域としては、8 ビット (@aa:8)、16 ビット (@aa:16)、または 32 ビット (@aa:32) を使用します。8 ビット絶対アドレスの場合、上位 24 ビットは SBR で指定されます。16 ビット絶対アドレスの場合、上位 16 ビットは符号拡張されます。32 ビット絶対アドレスの場合、全アドレス空間をアクセスできます。

プログラム領域としては 24 ビット (@aa:24)、または 32 ビット (@aa:32) を使用します。24 ビット (@aa:24) のとき上位 8 ビットはすべて 0 (H'00) となります。

絶対アドレスのアクセス範囲を表 2.13 に示します。

表 2.13 絶対アドレスのアクセス範囲

絶対アドレス		ノーマルモード	ミドルモード	アドバンスモード	マキシマムモード
データ領域	8 ビット (@aa:8)	SBR を上位アドレスとした任意の連続した 256 バイト			
	16 ビット (@aa:16)	H'0000~H'FFFF	H'000000~ H'007FFF、	H'00000000~H'00007FFF、 H'FFFF8000~H'FFFFFFFF	
	32 ビット (@aa:32)		H'FF8000~ H'FFFFFF	H'00000000~H'FFFFFFFF	
プログラム領域	24 ビット (@aa:24)		H'000000~ H'FFFFFF	H'00000000~H'00FFFFFF	
	32 ビット (@aa:32)	H'00000000~ H'00FFFFFF		H'00000000~ H'FFFFFFFF	

2.8.7 イミディエイト #xx

命令コード中に含まれる 8 ビット (#xx:8)、16 ビット (#xx:16)、または 32 ビット (#xx:32) のデータを直接オペランドとして使用します。

短縮形として、3 ビットまたは 4 ビットのイミディエイトデータを使用可能な場合があります。

データのサイズ (バイト/ワード/ロングワード) より、イミディエイトデータのビット数が小さい場合にはゼロ拡張されます。

なお、ADDS、SUBS、INC、DEC 命令では、イミディエイトデータ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中に含まれる場合があります。BFLD、BFST 命令では、ビットフィールドを指定するための 8 ビットのイミディエイトデータが命令コードの中に含まれます。また、TRAPA 命令では、ベクタアドレスを指定するための 2 ビットのイミディエイトデータが命令コードの中に含まれます。

2.8.8 プログラムカウンタ相対 @ (d:8, PC) /@ (d:16, PC)

Bcc、BSR 命令で使用されます。PC の内容で指定される 32 ビットのアドレスに、命令コード中に含まれる 8 ビット、または 16 ビットディスプレイメントを加算して、32 ビットの分岐アドレスを生成します。加算に際して、ディスプレイメントは 32 ビットに符号拡張されます。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)、または -32766~+32768 バイト (-16383~+16384 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。アドバンスモードのとき、加算結果は下位 24 ビットが有効になり、上位 8 ビットはすべて 0 (H'00) と見なされます。

2.8.9 プログラムカウンタインデックス相対 @ (RnL,B, PC) /@ (Rn.W, PC) / @ (ERn.L, PC)

Bcc、BSR 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アドレスレジスタの指定されたビット数 (RnL、Rn、ERn) の内容を 32 ビットにゼロ拡張し、2 を乗算し、乗算結果と PC の内容で指定される 32 ビットのアドレスを加算して、32 ビットの分岐アドレスを生成します。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。アドバンスモードのとき、加算結果は下位 24 ビットが有効になり、上位 8 ビットはすべて 0 (H'00) と見なされます。

2.8.10 メモリ間接 @@aa:8

JMP、JSR 命令で使用されます。命令コード中に含まれる 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。

8 ビット絶対アドレスの上位ビットはすべて 0 となりますので、分岐アドレスを格納できるのは、0~255 (ノーマルモードのとき H'0000~H'00FF、ノーマルモード以外のとき H'000000~H'0000FF) 番地です。

ノーマルモードの場合は、メモリ上のオペランドはワードサイズで指定し、16 ビットの分岐アドレスを生成します。

そのほかの場合は、メモリ上のオペランドはロングワードサイズで指定します。ミドルモードとアドバンスモードの場合は、このうち先頭の 1 バイトはすべて 0 (H'00) とみなされます。

ただし、分岐アドレスを格納可能なアドレスの先頭領域は、例外処理ベクタ領域と共通になっていますので注意してください。なお、リセットおよび CPU アドレスエラー以外の例外処理ベクタのアドレスは VBR で変更できます。

メモリ間接による分岐アドレスの指定を図 2.15 に示します。

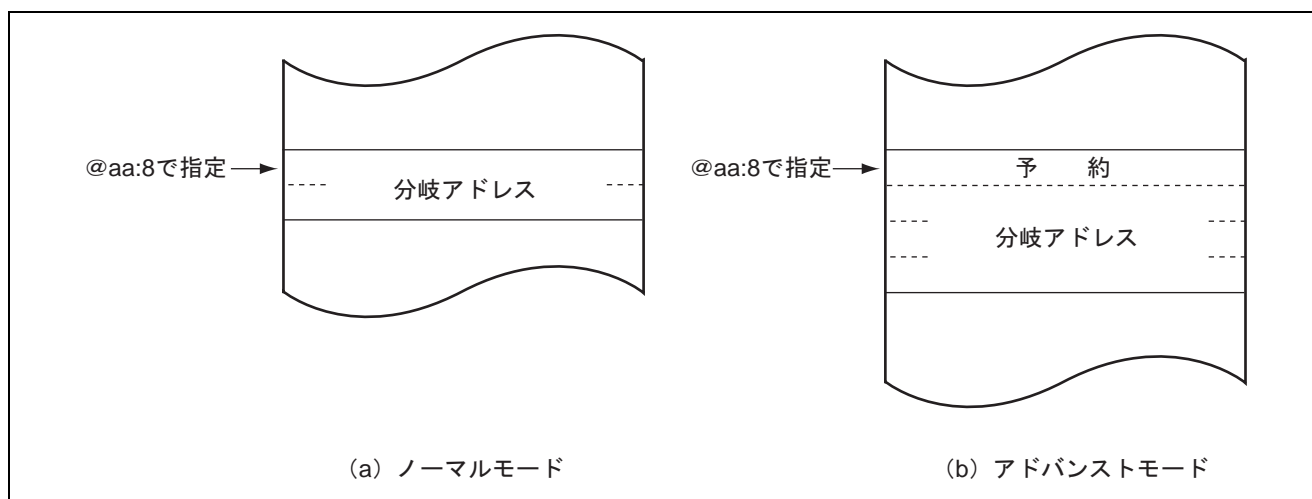


図 2.15 メモリ間接による分岐アドレスの指定

2.8.11 拡張メモリ間接 @@vec:7

JMP、JSR 命令で使用されます。命令コード中に含まれる 7 ビットのデータに H'80 を加算し、2 または 4 を乗算し、乗算結果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。

分岐アドレスを格納できるのは、ノーマルモードのとき H'0100～H'01FF、ノーマルモード以外るとき H'000200～H'0003FF 番地です。なお、アセンブラ表記上は、分岐アドレスを格納するアドレスを指定します。

ノーマルモードの場合は、メモリ上のオペランドはワードサイズで指定し、16 ビットの分岐アドレスを生成します。

そのほかの場合は、メモリ上のオペランドはロングワードサイズで指定します。ミドルモードとアドバンストモードの場合は、このうち先頭の 1 バイトはすべて 0 (H'00) と見なされます。

2.8.12 実効アドレスの計算方法

各アドレッシングモードにおける実効アドレス (EA : Effective Address) の計算方法を表 2.14、表 2.15 に示します。

CPU 動作モードに応じて、実効アドレス計算結果の下位ビットが有効になり、上位ビットは無視 (ゼロ拡張、または符号拡張) されます。

たとえばミドルモードのとき、

転送命令、演算命令の実効アドレスの下位 16 ビットが有効になり、上位 16 ビットは符号拡張されます。

分岐命令では、実効アドレスの下位 24 ビットが有効になり、上位 8 ビットはゼロ拡張されます。

表 2.14 転送／演算命令の実効アドレスの計算方法


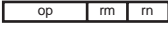

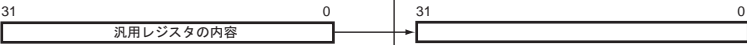
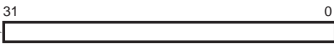
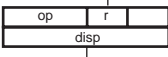
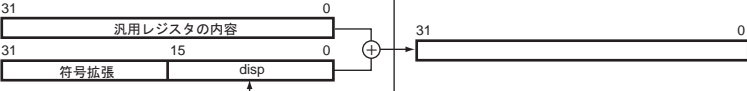
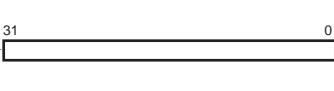
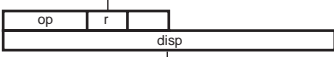
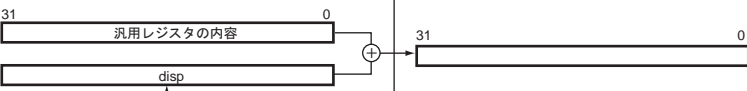
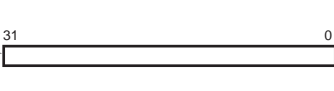
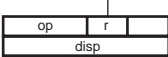
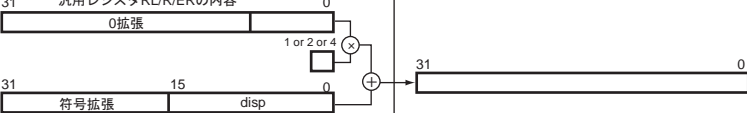
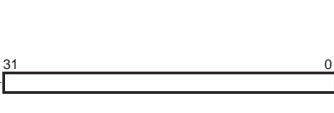

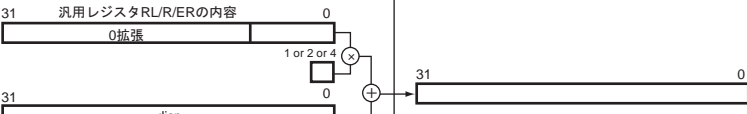
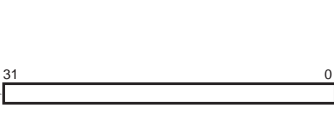

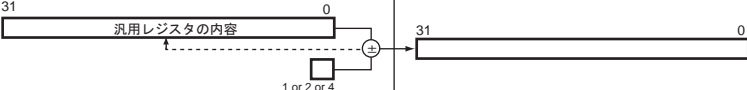
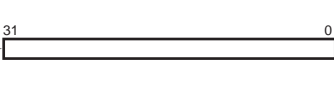
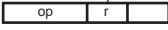
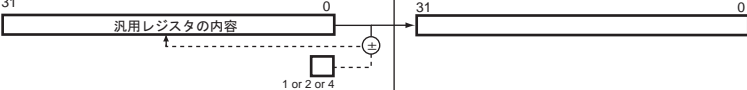
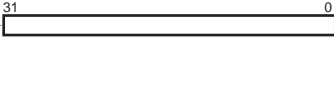
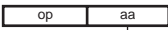
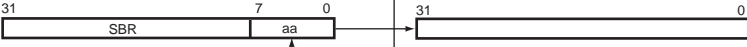
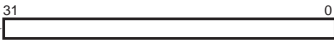
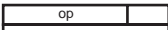





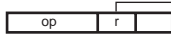

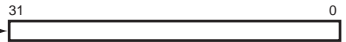
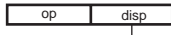
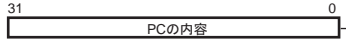
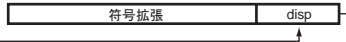

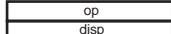

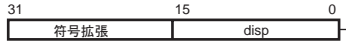





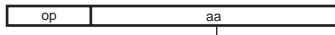
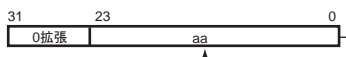
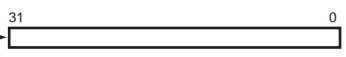

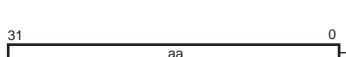
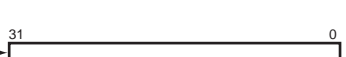
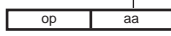
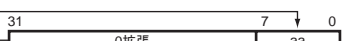
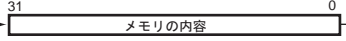

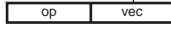

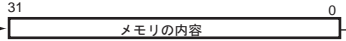

No.	アドレッシングモード・命令フォーマット	実効アドレスの計算方法	実効アドレス (EA)
1	イミディエイト 		
2	レジスタ直接 		
3	レジスタ間接 		
4	・16ビットディスプレイメント付きレジスタ間接 		
	・32ビットディスプレイメント付きレジスタ間接 		
5	・16ビットディスプレイメントインデックスレジスタ間接 		
	・32ビットディスプレイメントインデックスレジスタ間接 		
6	・ポストインクリメント／デクリメントレジスタ間接 		
	・プリインクリメント／デクリメントレジスタ間接 		
7	・8ビット絶対アドレス 		
	・16ビット絶対アドレス 		
	・32ビット絶対アドレス 		

表 2.15 分岐命令の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	実効アドレスの計算方法	実効アドレス (EA)
1	レジスタ間接 	汎用レジスタの内容 	31 0 
2	・プログラムカウンタ相対ディスプレースメント8ビット 	PCの内容  符号拡張  +	31 0 
	・プログラムカウンタ相対ディスプレースメント16ビット 	PCの内容  符号拡張  +	31 0 
3	プログラムカウンタインデックスレジスタ相対 	汎用レジスタRL/R/ERの内容 0拡張  × 2 PCの内容  +	31 0 
4	・24ビット絶対アドレス 	0拡張  aa	31 0 
	・32ビット絶対アドレス 	aa 	31 0 
5	メモリ間接 	0拡張  aa ↓ メモリの内容 	31 0 
6	拡張メモリ間接 	0拡張  1 vec ↓ 2 or 4 ↓ メモリの内容 	31 0 

2.8.13 MOVA 命令

MOVA 命令は、実効アドレスを汎用レジスタに格納します。

1. 表2.14のNo.2のアドレッシングモードでデータを取得します。
 2. 次に、このデータを表2.14のNo.5の汎用レジスタの代わりにインデックスとした実効アドレス計算を行い、その結果を汎用レジスタに格納します。
- 詳細は、「H8SX ファミリー ソフトウェアマニュアル」を参照してください。

2.9 処理状態

H8SX CPU の処理状態には、リセット状態、例外処理状態、プログラム実行状態、バス権解放状態、およびプログラム停止状態の 5 種類があります。処理状態間の状態遷移図を図 2.16 に示します。

- リセット状態

CPU および内蔵周辺モジュールがすべて初期化され、停止している状態です。リセット端子が Low レベルになると、実行中の処理はすべて中止され、CPU はリセット状態になります。リセット状態ではすべての割り込みが禁止されます。リセット端子を Low レベルから High レベルにすると、リセット例外処理を開始します。リセットの詳細は「第 4 章 例外処理」を参照してください。ウォッチドッグタイマを内蔵する製品では、ウォッチドッグタイマのオーバフローによってもリセットすることもできます。

- 例外処理状態

例外処理状態は、リセット、トレース、割り込み、またはトラップ命令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、例外処理ベクタテーブルからスタートアドレス（ベクタ）を取り出してそのスタートアドレスに分岐する過渡的な状態です。詳細は「第 4 章 例外処理」を参照してください。

- 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。

- バス権解放状態

CPU 以外のバスマスタからのバス権要求に対してバス権を解放した状態です。バス権解放状態では CPU は動作を停止します。

- プログラム停止状態

CPU が動作を停止し、消費電力を低下させた状態です。SLEEP 命令の実行、またはハードウェアスタンバイモード*への遷移で CPU はプログラム停止状態になります。詳細は「第 23 章 低消費電力」を参照してください。

【注】 * 本 LSI には、ハードウェアスタンバイモードはありません。

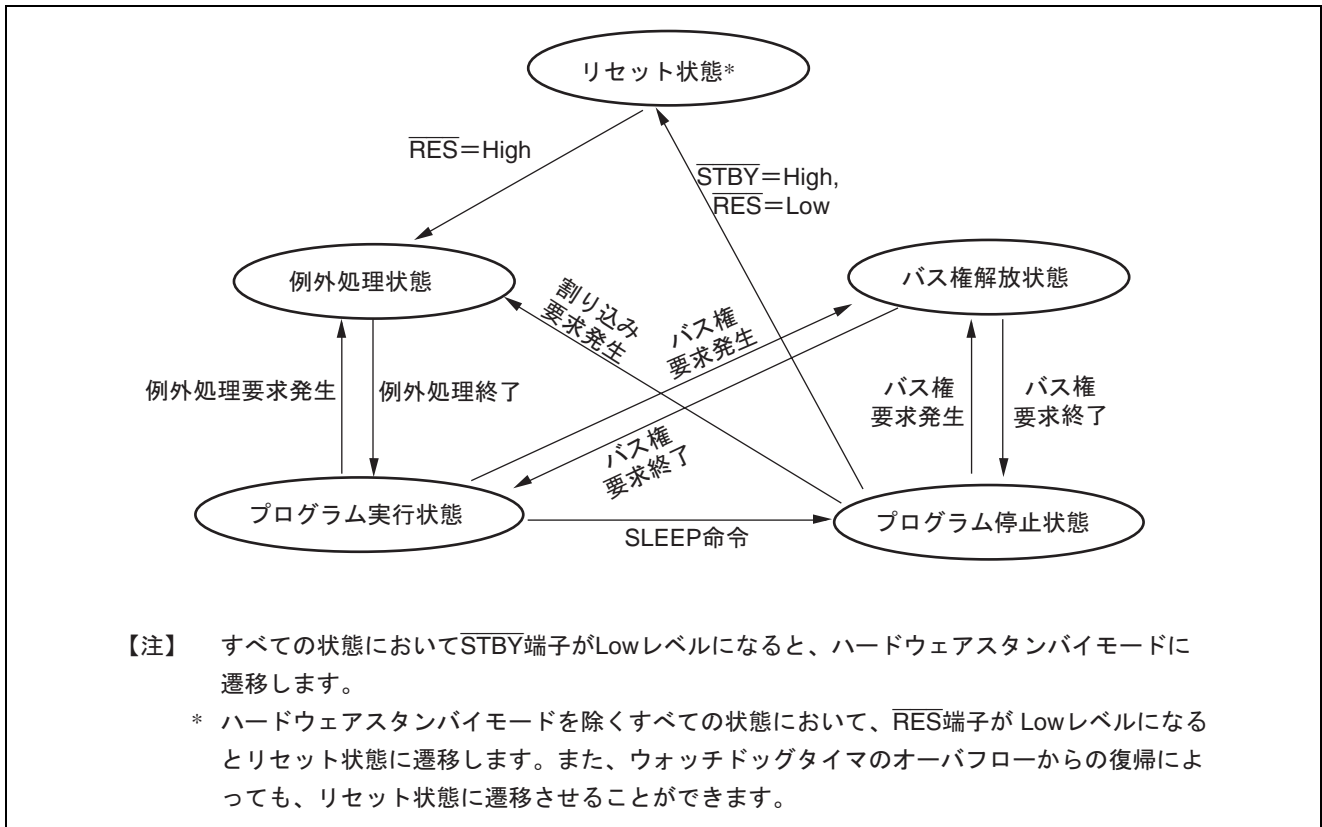


図 2.16 状態遷移図

3. MCU 動作モード

3.1 動作モードの選択

本 LSI には、3 種類の動作モード（モード 1～3）があります。動作モードは、モード端子（MD1、MD0）の設定で決まります。表 3.1 に MCU 動作モードの選択を示します。この表以外の組み合わせは設定しないでください。

CPU 動作モードはアドバンストモード、アドレス空間は 16M バイトです。起動モードは、フラッシュメモリに書き込み／消去を行えるブートモード起動、シングルチップモード起動から選択できます。

表 3.1 MCU 動作モードの選択

MCU 動作モード	MD1	MD0	CPU 動作モード	アドレス空間	内容	内蔵 ROM
1	0	1	—	—	リザーブ	—
2	1	0	アドバンスト	16M バイト	ブートモード	有効
3	1	1			シングルチップモード	有効

モード 1 は、リザーブです。本製品にはありません。設定しないでください。

モード 2 は、フラッシュメモリに書き込み／消去を行えるブートモードです。ブートモードについては、「第 20 章 フラッシュメモリ」を参照してください。

モード 3 は、シングルチップモードです。

3.2 レジスタの説明

動作モードに関連するレジスタには以下のものがあります。

- モードコントロールレジスタ (MDCR)
- システムコントロールレジスタ (SYSCR0)
- システムコントロールレジスタ (SYSCR1)

3.2.1 モードコントロールレジスタ (MDCR)

MDCR は、現在の動作モードをモニタします。MDCR をリードすると、MD1、MD0 端子の入力レベルがこれらのビットにラッチされます。このラッチはリセットで解除されます。

ビット	15	14	13	12	11	10	9	8
ビット名	—	—	—	—	MDS3	MDS2	MDS1	MDS0
初期値:	0	1	0	1	不定*	不定*	不定*	不定*
R/W:	R	R	R	R	R	R	R	R
ビット	7	6	5	4	3	2	1	0
ビット名	—	—	—	—	—	—	—	—
初期値:	0	1	0	1	不定*	不定*	不定*	不定*
R/W:	R	R	R	R	R	R	R	R

【注】 * MD1、MD0端子の設定により決定されます。

ビット	ビット名	初期値	R/W	説明
15	—	0	R	リザーブビット これらのビットはリードのみ有効で、ライトは無効です。
14	—	1	R	
13	—	0	R	
12	—	1	R	
11	MDS3	不定*	R	モードセレクト 3~0 モード端子 (MD1、MD0) により設定された動作モードに対応した値を示します (表 3.2 参照)。
10	MDS2	不定*	R	
9	MDS1	不定*	R	
8	MDS0	不定*	R	
7	—	0	R	リザーブビット これらのビットはリードのみ有効で、ライトは無効です。
6	—	1	R	
5	—	0	R	
4	—	1	R	
3	—	不定*	R	
2	—	不定*	R	
1	—	不定*	R	
0	—	不定*	R	

【注】 * MD1、MD0 端子の設定により決定されます。

表 3.2 MDS2~MDS0 ビットの値

MCU 動作モード	MD1	MD0	MDCR			
			MDS3	MDS2	MDS1	MDS0
1	0	1	リザーブ			
2	1	0	0	0	1	0
3	1	1	0	0	1	1

3.2.2 システムコントロールレジスタ (SYSCR0)

SYSCR0 は、MAC 飽和演算制御、内蔵 RAM の有効/無効の選択を行います。

ビット	15	14	13	12	11	10	9	8
ビット名	—	—	MACS	—	—	—	—	RAME
初期値 :	1	1	0	1	0	1	0	1
R/W :	R	R	R/W	R	R/W	R/W	R/W	R/W
ビット	7	6	5	4	3	2	1	0
ビット名	—	—	—	—	—	—	—	—
初期値 :	0	0	0	0	0	0	1	1
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ビット	ビット名	初期値	R/W	説明
15	—	1	R	リザーブビット
14	—	1	R	これらのビットはリードのみ有効で、ライトは無効です。
13	MACS	0	R/W	MAC 飽和演算制御 MAC 命令の飽和演算、非飽和演算を選択します。 0 : MAC 命令は非飽和演算 1 : MAC 命令は飽和演算
12	—	1	R	リザーブビット このビットはリードのみ有効で、ライトは無効です。
11	—	0	R/W	リザーブビット リードすると常に 0 が読み出されます。ライトする値は常に 0 にしてください。
10	—	1	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください。
9	—	0	R/W	リザーブビット リードすると常に 0 が読み出されます。ライトする値は常に 0 にしてください。

ビット	ビット名	初期値	R/W	説明
8	RAME	1	R/W	RAM イネーブル 内蔵 RAM の有効または無効を選択します。このビットは、リセットを解除すると初期化されず。内蔵 RAM アクセス中に 0 ライトを行わないでください。 0 : 内蔵 RAM 無効 1 : 内蔵 RAM 有効
7~2	—	すべて 0	R/W	リザーブビット リードすると常に 0 が読み出されます。ライトする値は常に 0 にしてください。
1	—	1	R/W	リザーブビット
0	—	1	R/W	リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください。

3.2.3 システムコントロールレジスタ (SYSCR1)

SYSCR1 はデータフラッシュへのアクセスステート制御を行います。

ビット	15	14	13	12	11	10	9	8
ビット名	—	—	—	—	—	—	—	—
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
ビット	7	6	5	4	3	2	1	0
ビット名	—	—	—	—	—	—	—	EEPWT
初期値 :	0	0	0	0	0	0	0	1
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ビット	ビット名	初期値	R/W	説明
15	—	0	R/W	リザーブビット
14	—	0	R/W	リードすると常に 0 が読み出されます。ライトする値は常に 0 にしてください。
13	—	0	R/W	
11	—	0	R/W	
10	—	0	R/W	
9	—	0	R/W	
8	—	0	R/W	
7	—	0	R/W	
6	—	0	R/W	
5	—	0	R/W	
4	—	0	R/W	
3	—	0	R/W	
2	—	0	R/W	
1	—	0	R/W	

ビット	ビット名	初期値	R/W	説明
0	EEPWT	1	R/W	データフラッシュウェイトコントロール データフラッシュをアクセスするとき、ウェイトを1ステート挿入します。 0: ウェイトを挿入しない (2ステートアクセス) 1: ウェイトを1ステート挿入 (3ステートアクセス) 詳細は「第21章 データフラッシュ (EEPROM)」を参照してください。

3.3 動作モードの説明

3.3.1 モード1

モード1は、リザーブです。本LSIにはありません。設定しないでください。

3.3.2 モード2

モード2は、フラッシュメモリのブートモードです。フラッシュメモリの書き込み、消去以外は、モード3と同様の動作となります。

3.3.3 モード3

モード3は、CPUがアドバンスモード、アドレス空間16Mバイト、内蔵ROM有効のシングルチップモードです。

3.3.4 アドレスマップ

各動作モードのアドレスマップを図 3.1、図 3.2 に示します。

H'000000	内蔵ROM空間 (リード)
H'080000	アクセス禁止空間
H'C00000	FCU用 ファーム空間
H'C02000	アクセス禁止空間
H'D00000	内蔵ROM空間 (ライト)
H'D80000	アクセス禁止空間
H'E00000	内蔵EEPROM空間*
H'E08000	アクセス禁止空間
H'EF8000	FCU用 内蔵RAM空間
H'EFA000	アクセス禁止空間
H'FF2000	内蔵RAM空間
H'FFC000	周辺 モジュール空間
H'FFFFFF	

【注】 * 内蔵EEPROM空間には、ロットトレース情報が含まれます。詳細は、「第21章 データフラッシュ (EEPROM)」を参照してください。

図 3.1 アドレスマップ (H8SX/1727S)

H'000000	内蔵ROM空間 (リード)
H'040000	アクセス禁止空間
H'C00000	FCU用 ファーム空間
H'C02000	アクセス禁止空間
H'D00000	内蔵ROM空間 (ライト)
H'D40000	アクセス禁止空間
H'E00000	内蔵EEPROM空間*
H'E04000	アクセス禁止空間
H'EF8000	FCU用 内蔵RAM空間
H'EFA000	アクセス禁止空間
H'FF6000	内蔵RAM空間
H'FFC000	周辺 モジュール空間
H'FFFFFF	

【注】 * 内蔵EEPROM空間には、ロットトレース情報が含まれます。詳細は、「第21章 データフラッシュ (EEPROM)」を参照してください。

図 3.2 アドレスマップ (H8SX/1725S)

4. 例外処理

4.1 例外処理の種類と優先度

例外処理要因には、表 4.1 に示すように、リセット、トレース、アドレスエラー、割り込み、トラップ命令、および不当命令（一般不当命令、スロット不当命令）があります。これらの例外処理要因には、表 4.1 のように優先順位が設けられており、複数の例外処理要因が同時に発生した場合は、この優先度に従って受け付けられます。例外処理は割り込み制御モードによって、例外処理要因やスタックの構造、CPU の動作が異なります。割り込み制御モードの詳細については、「第 5 章 割り込みコントローラ」を参照してください。

表 4.1 例外処理の種類と優先度

優先度	例外処理の種類	例外処理開始タイミング
高 ↑ 低	リセット	RES 端子の Low レベルから High レベルへの変化、または、ウォッチドッグタイマのオーバーフローにより開始します。 $\overline{\text{RES}}$ 端子が Low レベルのときリセット状態になります。
	不当命令	未定義コードが実行されると開始します。
	トレース* ¹	EXR のトレース (T) ビット=1 の状態で、実行中の命令または例外処理の完了後開始します。
	アドレスエラー	アドレスエラーのバスサイクルが発生後、命令実行終了時に例外処理を開始します。
	割り込み	割り込み要求が発生すると、実行中の命令または例外処理の完了後開始します。* ²
	トラップ命令* ³	トラップ (TRAPA) 命令の実行により開始します。

【注】 *1 トレースは割り込み制御モード 2 でのみ有効です。トレース例外処理は、RTE 命令の実行終了後には実行しません。

*2 ANDC、ORC、XORC、LDC 命令の実行終了時点、またはリセット例外処理の終了時点では割り込みの検出を行いません。

*3 トラップ命令例外処理は、プログラム実行状態で常に受け付けられます。

4.2 例外処理要因とベクタテーブル

例外処理要因には、それぞれ異なるベクタテーブルアドレスオフセットが割り当てられています。ベクタテーブルアドレスは、ベクタベースレジスタ（VBR）の値と対応するベクタ番号のベクタテーブルアドレスオフセットから算出されます。例外処理では、このベクタテーブルアドレスが示す例外処理ベクタテーブルから、例外サービスルーチンのスタートアドレスが取り出されます。

例外処理要因と、ベクタテーブルアドレスオフセットとの対応を表 4.2 に示します。例外処理ベクタテーブルアドレスの算出法を表 4.3 に示します。

製品によって使用できるモードが異なりますので、製品ごとの詳細は「第 3 章 MCU 動作モード」を参照してください。

表 4.2 例外処理ベクタテーブル

例外処理要因	ベクタ番号	ベクタテーブルアドレスオフセット* ¹
		アドバンストモード
リセット	0	H'0000~H'0003
システム予約	1	H'0004~H'0007
	2	H'0008~H'000B
	3	H'000C~H'000F
不当命令	4	H'0010~H'0013
トレース	5	H'0014~H'0017
システム予約	6	H'0018~H'001B
割り込み（NMI）	7	H'001C~H'001F
トラップ命令（#0）	8	H'0020~H'0023
トラップ命令（#1）	9	H'0024~H'0027
トラップ命令（#2）	10	H'0028~H'002B
トラップ命令（#3）	11	H'002C~H'002F
CPU アドレスエラー	12	H'0030~H'0033
DMA アドレスエラー* ²	13	H'0034~H'0037
システム予約	14	H'0038~H'003B
	63	H'00FC~H'00FF

例外処理要因	ベクタ番号	ベクタテーブルアドレスオフセット* ¹
		アドバンスモード
外部割り込み IRQ0	64	H'0100~H'0103
外部割り込み IRQ1	65	H'0104~H'0107
外部割り込み IRQ2	66	H'0108~H'010B
外部割り込み IRQ3	67	H'010C~H'010F
外部割り込み IRQ4	68	H'0110~H'0113
外部割り込み IRQ5	69	H'0114~H'0117
外部割り込み IRQ6	70	H'0118~H'011B
外部割り込み IRQ7	71	H'011C~H'011F
外部割り込み IRQ8	72	H'0120~H'0123
外部割り込み IRQ9	73	H'0124~H'0127
外部割り込み IRQ10	74	H'0128~H'012B
外部割り込み IRQ11	75	H'012C~H'012F
外部割り込み IRQ12	76	H'0130~H'0133
外部割り込み IRQ13	77	H'0134~H'0137
外部割り込み IRQ14	78	H'0138~H'013B
システム予約	79	H'013C~H'013F
内部割り込み* ³	80	H'0140~H'0143
	255	H'03FC~H'03FF

【注】 *1 先頭アドレスの下位 16 ビットを示しています。

*2 DMA アドレスエラーの要因発生元は、DTC、DMAC です。

*3 割り込みのベクタテーブルは、「5.5 割り込み例外処理ベクタテーブル」を参照してください。

表 4.3 例外処理ベクタテーブルアドレスの算出法

例外処理要因	ベクタテーブルアドレス算出法
リセット、CPU アドレスエラー	ベクタテーブルアドレス = (ベクタテーブルアドレスオフセット)
上記以外	ベクタテーブルアドレス = VBR + (ベクタテーブルアドレスオフセット)

【注】 VBR : ベクタベースレジスタ

ベクタテーブルアドレスオフセット : 表 4.2 を参照

4.3 リセット

リセットは、最も優先順位の高い例外処理です。

$\overline{\text{RES}}$ 端子が Low レベルになると、実行中の処理はすべて打ち切られ、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、動作中は $\overline{\text{RES}}$ 端子を最低 100 ステートの間、Low レベルに保持してください。

また、ウォッチドッグタイマのオーバフローによって、リセット状態とすることもできます。詳細は「第 12 章 ウォッチドッグタイマ (WDT)」を参照してください。

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初期化されます。リセット直後は、割り込み制御モードは 0 になっています。

4.3.1 リセット例外処理

$\overline{\text{RES}}$ 端子が一定期間 Low レベルの後 High レベルになると、リセット例外処理を開始し、本 LSI は次のように動作します。

1. CPUの内部状態と内蔵周辺モジュールの各レジスタが初期化され、VBRがH'00000000にクリアされ、EXRのTビットが0にクリアされます。EXR、CCRのIビットは1にセットされます。
2. リセット例外処理ベクタアドレスをリードして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。

リセットシーケンスの例を図 4.1 に示します。

4.3.2 リセット直後の割り込み

リセット直後、スタックポインタ (SP) を初期化する前に割り込みを受け付けると、PC と CCR の退避が正常に行われないため、プログラムの暴走につながります。これを防ぐため、リセット例外処理が実行された直後は、NMI を含めたすべての割り込み要求が禁止されます。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、プログラム先頭命令は SP を初期化する命令としてください (例: MOV.L #xx : 32, SP)。

4.3.3 リセット解除後の内蔵周辺機能

リセット解除後、MSTPCRA は H'1FFF、MSTPCRB は H'FFFF、MSTPCRC は H'FF00、MSTPCRD は H'FF00、MSTPCRE は H'FF00 に初期化され、DMAC と DTC を除くすべてのモジュールがモジュールストップモードになっています。そのため、各内蔵周辺モジュールのレジスタは、リード/ライトできません。モジュールストップモードを解除することにより、レジスタのリード/ライトが可能となります。

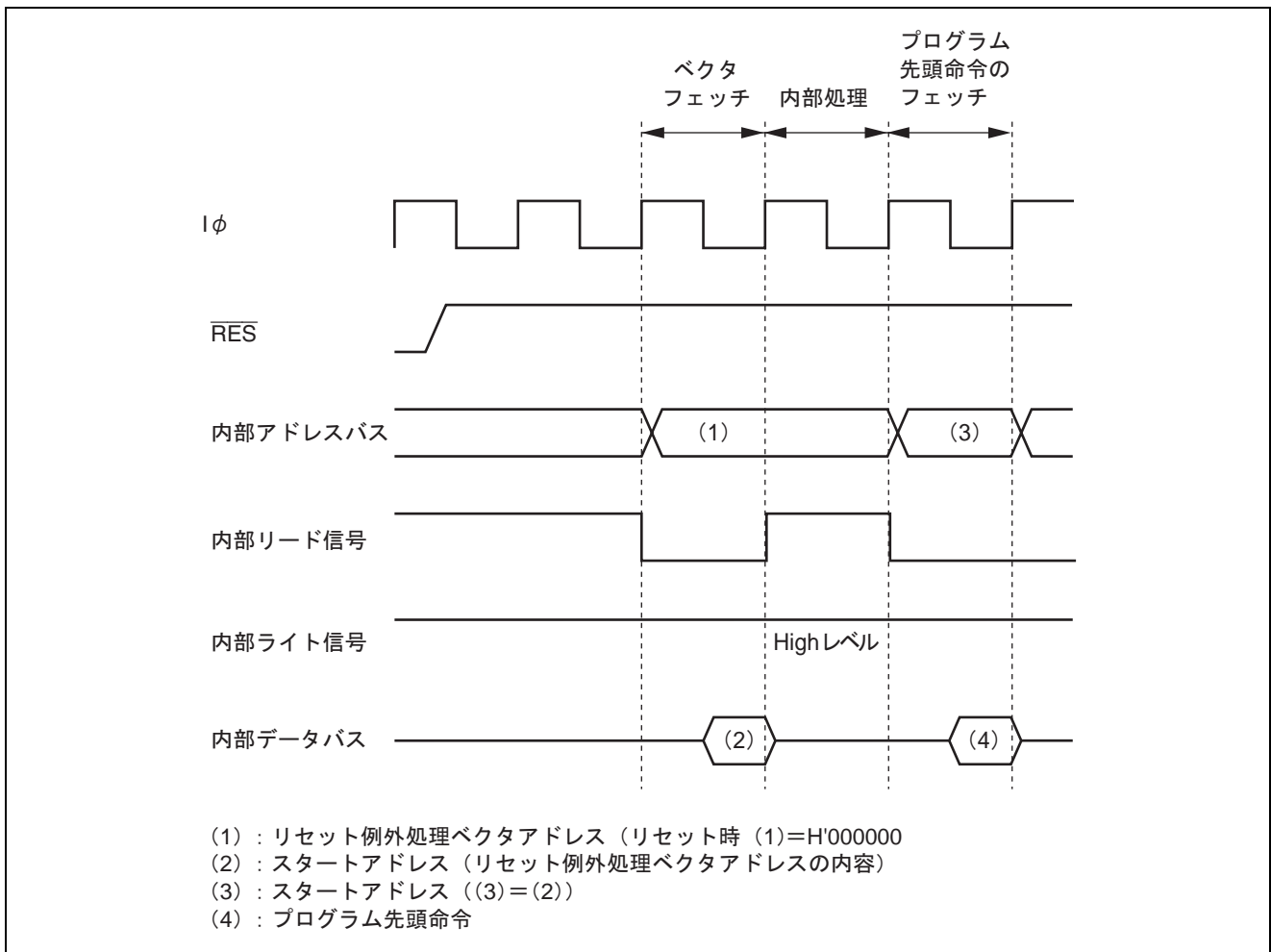


図 4.1 リセットシーケンス (アドバンスモード/内蔵 ROM 有効)

4.4 トレース例外処理

トレースは、割り込み制御モード2で有効です。割り込み制御モード0では、Tビットの状態にかかわらず、トレースモードにはなりません。割り込み制御モードについては、「第5章 割り込みコントローラ」を参照してください。割り込み制御モードを変更する場合は、Tビットを0の状態に変更してください。

EXRのTビットを1にセットすると、トレースモードになります。トレースモードではCPUが1命令の実行を完了するたびにトレース例外処理を開始します。トレース例外処理はCCRの割り込みマスクビットの影響を受けません。表4.4にトレース例外処理後のCCR、EXRの状態を示します。トレース例外処理によってEXRのTビットが0にクリアされてトレースモードが解除されますが、スタックに退避されたTビットは1を保持しており、RTE命令によってトレース例外処理ルーチンから復帰した後は再びトレースモードになります。RTE命令の実行ではトレース例外処理を行いません。

トレース例外処理ルーチンでも割り込みを受け付けます。

表 4.4 トレース例外処理後のCCR、EXRの状態

割り込み制御モード	CCR		EXR	
	I	UI	I2~I0	T
0	トレース例外処理は使用できません。			
2	1	—	—	0

【記号説明】

- 1 : 1にセットされます。
- 0 : 0にクリアされます。
- : 実行前の値が保持されます。

4.5 アドレスエラー

4.5.1 アドレスエラー発生要因

アドレスエラーは、表 4.5 に示すように命令フェッチ、スタック操作、データ読み出し／書き込み時に発生します。

表 4.5 バスサイクルとアドレスエラー

バスサイクル		バスサイクルの内容	アドレスエラーの発生
種類	バスマスタ		
命令フェッチ	CPU	偶数アドレスから命令をフェッチ	なし（正常）
		奇数アドレスから命令をフェッチ	アドレスエラー発生
		内蔵 ROM 空間（ライト）、内蔵 EEPROM 空間（リード／ライト）、FCU 用内蔵 RAM 空間、内蔵周辺モジュール空間*1 から命令をフェッチ	アドレスエラー発生
		内蔵 ROM 空間（ライト）、内蔵 EEPROM 空間（リード／ライト）、FCU 用内蔵 RAM 空間、内蔵周辺モジュール空間*1 以外から命令をフェッチ	なし（正常）
		アクセス禁止空間*2 から命令をフェッチ	アドレスエラー発生
スタック操作	CPU	スタックポインタの値が偶数アドレスでスタックをアクセス	なし（正常）
		スタックポインタの値が奇数アドレスでスタックをアクセス	アドレスエラー発生
データ読み出し／書き込み	CPU	ワードデータを偶数アドレスからアクセス	なし（正常）
		ワードデータを奇数アドレスからアクセス	なし（正常）
		シングルチップモード時に外部メモリ空間をアクセス	アドレスエラー発生
		アクセス禁止空間*3 をアクセス	アドレスエラー発生
データ読み出し／書き込み	DTC／DMAC	ワードデータを偶数アドレスからアクセス	なし（正常）
		ワードデータを奇数アドレスからアクセス	なし（正常）
		シングルチップモード時に外部メモリ空間をアクセス	アドレスエラー発生
		アクセス禁止空間*3 をアクセス	アドレスエラー発生

【注】 *1 内蔵周辺モジュール空間については、「第 6 章 バスコントローラ（BSC）」を参照してください。

*2 アクセス禁止空間については、「3.3.4 アドレスマップ」を参照してください。

ただし、以下のアクセス禁止空間の領域から命令フェッチをした場合、アドレスエラーは発生しません。

H'800000～H'BFFFFFF、H'C01800～H'FFFFFF

*3 アクセス禁止空間については、「3.3.4 アドレスマップ」を参照してください。

ただし、以下のアクセス禁止空間の領域をアクセスした場合、アドレスエラーは発生しません。

H'800000～H'BFFFFFF、H'C01800～H'FFFFFF、H'D40000～H'DFFFFFF、H'E040000～H'EF7FFF、H'EF9800～H'FFFFFF

4.5.2 アドレスエラー例外処理

アドレスエラーが発生すると、アドレスエラーを起こしたバスサイクルが終了し、実行中の命令が完了してからアドレスエラー例外処理が開始されます。このとき、CPU は次のように動作します。

1. PCとCCR、EXRの内容をスタックに退避します。
2. 割り込みマスクビットを更新し、Tビットを0にクリアします。
3. 発生したアドレスエラーに対応する例外処理ベクタテーブルアドレスを生成し、ベクタテーブルから例外サービスルーチンスタートアドレスをPCにロードして、その番地からプログラムの実行を開始します。

アドレスエラー例外処理に遷移するときにアドレスエラーが発生した場合、そのときのアドレスエラーは受け付けません。これは例外処理のスタッキングで発生するアドレスエラーを回避して、無限にアドレスエラー例外処理によるスタッキングが続かないようにするためです。

アドレスエラー例外処理が発生したときにSPが2の倍数になっていない場合、スタックされた値(PC, CCR, EXR)は不定です。

また、アドレスエラーが発生すると、DTC、DMACを停止させるために次の制御を行います。

- DTCのDTCCRのERRビットを1にセット
- DMACのDMDR_0のERRFビットを1にセット
- DMACの全チャンネルのDMDRのDTEビットを0にクリア、および強制転送終了

表 4.6 にアドレスエラー例外処理後のCRC、EXRの状態を示します。

表 4.6 アドレスエラー例外処理後のCCR、EXR

割り込み制御モード	CCR		EXR	
	I	UI	I2~I0	T
0	1	—	—	—
2	1	—	7	0

【記号説明】

- 1 : 1にセットされます。
- 0 : 0にクリアされます。
- : 実行前の値が保持されます。

4.6 割り込み

4.6.1 割り込み要因

割り込み例外処理を起動させる要因には、表 4.7 に示すように NMI、IRQ0～IRQ14、内蔵周辺モジュールがあります。

表 4.7 割り込み要因

種類	要求元	要因数
NMI	NMI 端子 (外部からの入力)	1
IRQ0～IRQ14	$\overline{\text{IRQ0}}\sim\overline{\text{IRQ14}}$ 端子 (外部からの入力)	15
内蔵周辺モジュール	ウォッチドッグタイマ (WDT)	1
	メモリエラー (ROM/RAM)	2
	A/D 変換器	2
	クロック発振器 (外部発振停止検出)	1
	16 ビットタイマパルスユニット (TPU)	52
	DMA コントローラ (DMAC)	8
	コントローラエリアネットワーク (RCAN-TL1)	4
	ハードウェア LIN	1
	シリアルコミュニケーションインタフェース (SCI)	8
	ルネサスシリアルペリフェラルインタフェース (RSPI)	16

各割り込み要因には、それぞれ異なるベクタ番号とベクタテーブルオフセットが割り当てられています。ベクタ番号とベクタテーブルアドレスオフセットについては「第 5 章 割り込みコントローラ」の表 5.2 を参照してください。

4.6.2 割り込み例外処理

割り込みは、割り込みコントローラによって制御されます。割り込み制御には、2つの割り込み制御モードがあり、NMI以外の割り込みに8レベルの優先順位/マスクレベルを設定して、多重割り込みの制御を行うことができます。割り込み例外処理を開始させる要因とベクタアドレスは製品によって異なります。詳細は「**第5章 割り込みコントローラ**」を参照してください。

割り込み例外処理は、次のように動作します。

1. PCとCCR、EXRの内容をスタックに退避します。
2. 割り込みマスクビットを更新し、Tビットを0にクリアします。
3. 割り込み要因に対応する例外処理ベクタテーブルアドレスを生成し、ベクタテーブルから例外サービスルーチンスタートアドレスをPCにロードして、その番地からプログラムの実行を開始します。

4.7 命令による例外処理

例外処理を起動する命令には、トラップ命令、不当命令があります。

4.7.1 トラップ命令例外処理

トラップ命令例外処理は、TRAPA命令を実行すると例外処理を開始します。トラップ命令例外処理はプログラム実行状態で常に実行可能です。

トラップ命令例外処理は、次のように動作します。

1. PCとCCR、EXRの内容をスタックに退避します。
2. 割り込みマスクビットを更新し、Tビットを0にクリアします。
3. TRAPA命令で指定したベクタ番号に対応する例外処理ベクタテーブルアドレスを生成し、ベクタテーブルから例外サービスルーチンスタートアドレスをPCにロードして、その番地からプログラムの実行を開始します。

TRAPA命令は、命令コード中で指定した0~3のベクタ番号に対応するベクタテーブルからスタートアドレスを取り出します。

トラップ命令例外処理実行後のCCR、EXRの状態を表4.8に示します。

表 4.8 トラップ命令例外処理後のCCR、EXRの状態

割り込み制御モード	CCR		EXR	
	I	UI	I2~I0	T
0	1	—	—	—
2	1	—	7	0

【記号説明】

- 1 : 1にセットされます。
- 0 : 0にクリアされます。
- : 実行前の値が保持されます。

4.7.2 不当命令例外処理

不当命令には、一般不当命令とスロット不当命令があります。

一般不当命令例外処理は、未定義コードを実行すると例外処理を開始します。

スロット不当命令例外処理は、遅延スロットの命令（遅延分岐命令の直後の命令）として、2ワード以上の命令、またはPCを書き換える命令を実行すると、例外処理を開始します。

一般不当命令例外処理、およびスロット不当命令例外処理はプログラム実行状態で常に実行可能です。

一般不当命令例外処理、およびスロット不当命令例外処理は、次のように動作します。

1. PCとCCR、EXRの内容をスタックに退避します。
2. 割り込みマスクビットを更新し、Tビットを0にクリアします。
3. 発生した例外に対応する例外処理ベクタテーブルアドレスを生成し、ベクタテーブルから例外サービスルーチンスタートアドレスをPCにロードして、その番地からプログラムの実行を開始します。

表 4.9 に、不当命令例外処理後の CCR、EXR の状態を示します。

表 4.9 不当命令例外処理後の CCR、EXR の状態

割り込み制御モード	CCR		EXR	
	I	UI	I2~I0	T
0	1	—	—	—
2	1	—	—	0

【記号説明】

1 : 1にセットされます。

0 : 0にクリアされます。

— : 実行前の値が保持されます。

4.8 例外処理後のスタックの状態

例外処理後のスタックの状態を図 4.2 に示します。

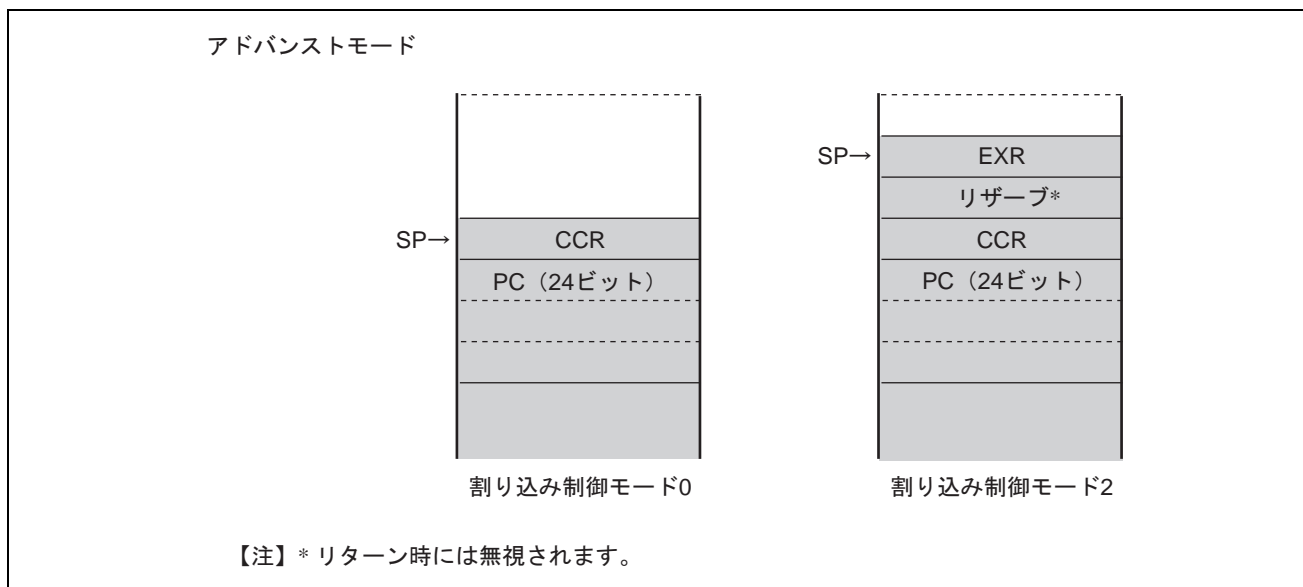


図 4.2 例外処理終了後のスタックの状態

4.9 使用上の注意事項

スタックを操作するアクセスを行う場合は、アドレスの最下位ビットは0と見なされます。スタック領域に対するアクセスは、常にワードサイズまたはロングワードサイズで行い、スタックポインタ (SP : ER7) の内容は奇数にしないでください。

すなわち、レジスタの退避は、

PUSH.W Rn (MOV.W Rn, @-SP)

PUSH.L ERn (MOV.L ERn, @-SP)

また、レジスタの復帰は、

POP.W Rn (MOV.W @SP+, Rn)

POP.L ERn (MOV.L @SP+, ERn)

を使用してください。

SP を奇数に設定した状態でスタック操作を行うとアドレスエラーが発生します。SP を奇数に設定した場合の動作例を図 4.3 に示します。

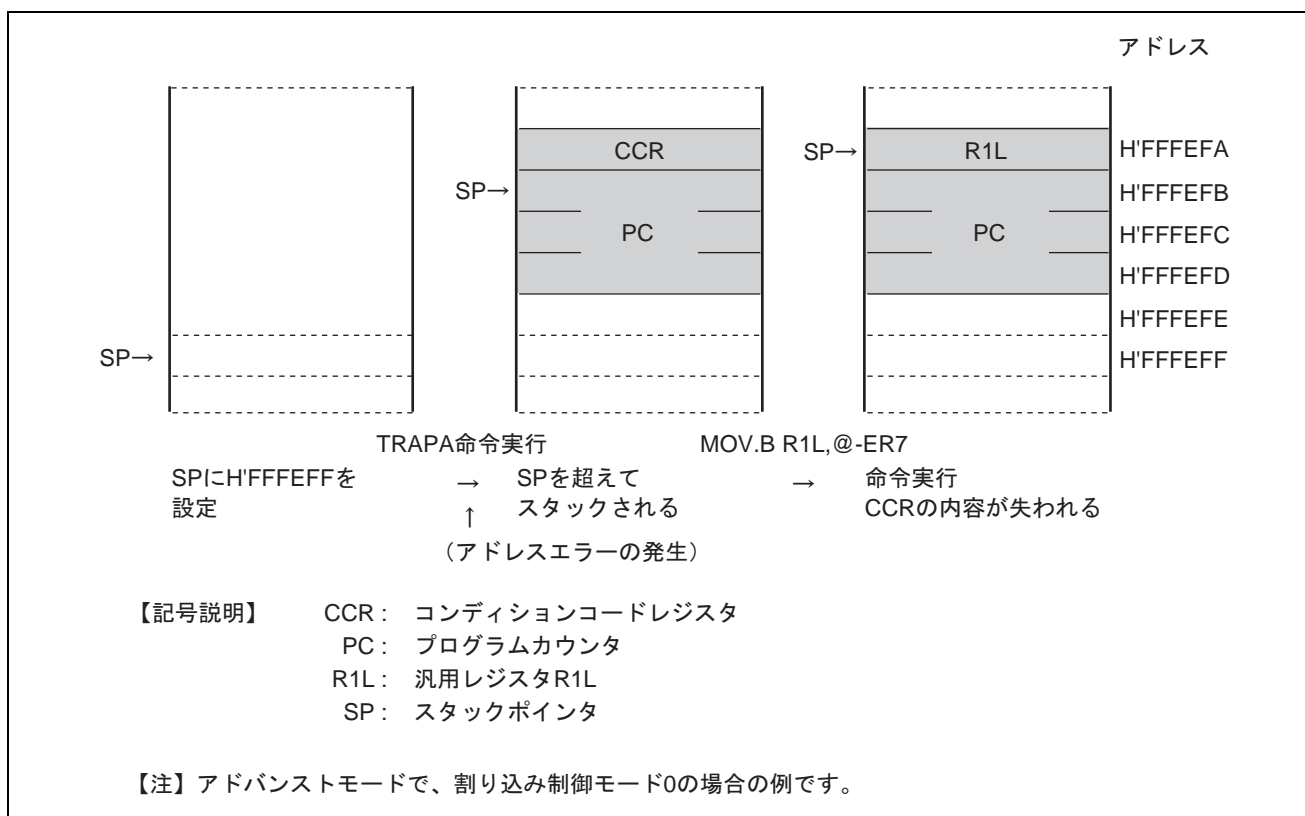


図 4.3 SP を奇数に設定したときの動作

5. 割り込みコントローラ

5.1 特長

- 2種類の割り込み制御モード
割り込みコントロールレジスタ (INTCR) の INTM1、INTM0 ビットにより、2種類の割り込み制御モードを選択できます。
- インタラプトプライオリティレジスタ (IPR) により、優先順位を設定可能
IPR により、以下に示した割り込み要求以外は、モジュールごとに 8 レベルの優先順位を設定できます。
(1) ~ (6) の要求は、最優先のレベル 8 の割り込みとして常に受け付けられます。
(1) NMI
(2) 不当命令
(3) トレース
(4) トラップ命令
(5) CPU アドレスエラー
(6) DMA アドレスエラー*
- 独立したベクタアドレス
すべての割り込み要因には独立したベクタアドレスが割り当てられており、割り込み処理ルーチンで要因を判別する必要がありません。
- 16本の外部割り込み端子
NMI は最優先の割り込みで常に受け付けられます。NMI は立ち上がりエッジ、または立ち下がりエッジを選択できます。 $\overline{\text{IRQ14}} \sim \overline{\text{IRQ0}}$ は立ち下がりエッジ、立ち上がりエッジ、両エッジ、レベルセンスのいずれかをそれぞれ独立に選択できます。
- DTC、DMACの制御
割り込み要求により、DTC、DMAC を起動することができます。
- CPUの優先レベル制御機能
CPU と、DTC、DMAC との間の優先レベルを設定できます。CPU の優先レベルは例外処理により自動設定することが可能で、CPU の割り込み例外処理などを DTC、DMAC の転送処理よりも優先させることができます。

【注】 * DMA アドレスエラーの要因発生元は、DTC、DMAC です。

割り込みコントローラのブロック図を図 5.1 に示します。

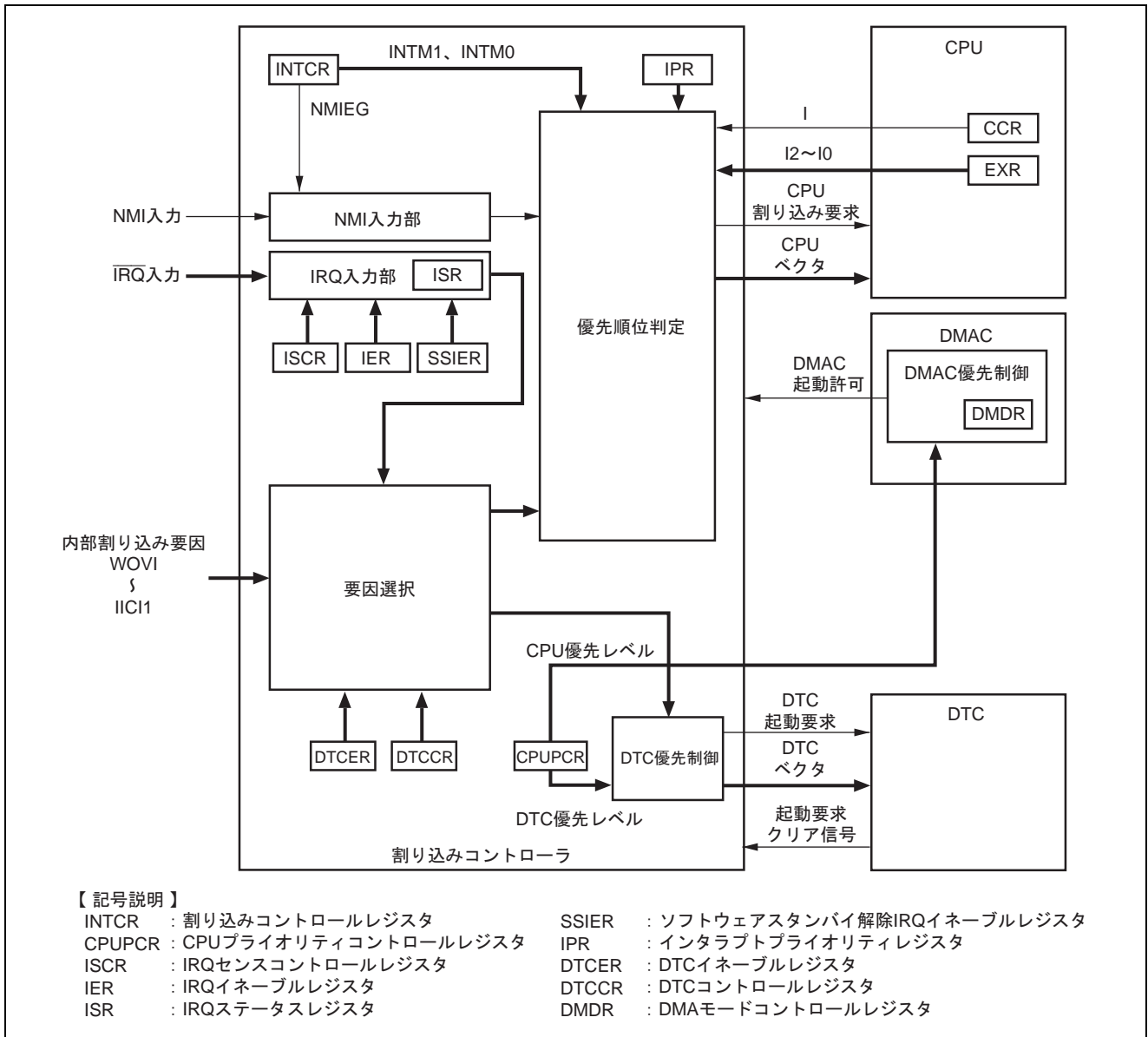


図 5.1 割り込みコントローラのブロック図

5.2 入出力端子

割り込みコントローラの端子構成を表 5.1 に示します。

表 5.1 端子構成

名称	入出力	機能
NMI	入力	ノンマスクابل外部割り込み端子 立ち上がりエッジ、または立ち下がりエッジを選択できます。
$\overline{\text{IRQ14}}\sim\overline{\text{IRQ0}}$	入力	マスク可能な外部割り込み端子 立ち下がりエッジ、立ち上がりエッジ、両エッジ、レベルセンスのいずれかを独立に選択できます。

5.3 レジスタの説明

割り込みコントローラには以下のレジスタがあります。

- 割り込みコントロールレジスタ (INTCR)
- CPUプライオリティコントロールレジスタ (CPUPCR)
- インタラプトプライオリティレジスタA~P (IPRA~IPRP)
- IRQイネーブルレジスタ (IER)
- IRQセンスコントロールレジスタH、L (ISCRH、ISCRL)
- IRQステータスレジスタ (ISR)
- ソフトウェアスタンバイ解除IRQイネーブルレジスタ (SSIER)

5.3.1 割り込みコントロールレジスタ (INTCR)

INTCR は、割り込み制御モードの選択、NMI の検出エッジの選択を行います。

ビット	7	6	5	4	3	2	1	0
ビット名	—	—	INTM1	INTM0	NMIEG	—	—	—
初期値:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R	R	R

ビット	ビット名	初期値	R/W	説明
7	—	0	R	リザーブビット
6	—	0	R	リードすると常に0が読み出されます。ライトは無効です。
5	INTM1	0	R/W	割り込み制御選択モード 1、0
4	INTM0	0	R/W	割り込みコントローラの割り込み制御モードを選択します。 00: 割り込み制御モード 0 CCR の 1 ビットで割り込みを制御します。 01: 設定禁止 10: 割り込み制御モード 2 EXR の I2~I0 ビットと IPR で割り込みを制御します。 11: 設定禁止
3	NMIEG	0	R/W	NMI エッジセレクト NMI 端子の入力エッジ選択を行います。 0: NMI 入力の立ち下がりで割り込み要求を発生 1: NMI 入力の立ち上がりエッジで割り込み要求を発生
2~0	—	すべて 0	R	リザーブビット リードすると常に0が読み出されます。ライトは無効です。

5.3.2 CPU プライオリティコントロールレジスタ (CPUPCR)

CPUPCR は、DTC、DMAC に対して CPU の優先レベルを設定します。CPU の割り込み例外処理などを DTC、DMAC の転送処理よりも優先させることができます。DTC の優先レベルは、CPUPCR の DTCP2~DTCP0 ビットで設定します。DMAC の優先レベルは、各チャンネル毎に DMAC のコントロールレジスタで設定します。

ビット	7	6	5	4	3	2	1	0
ビット名	CPUPCE	DTCP2	DTCP1	DTCP0	IPSETE	CPUP2	CPUP1	CPUP0
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】 * IPSETEビットが1にセットされると、自動更新になるためライトは無効です。

ビット	ビット名	初期値	R/W	説明
7	CPUPCE	0	R/W	CPU プライオリティコントロールイネーブル CPU の優先レベル制御機能を制御します。このビットを 1 にセットすると、DTC、DMAC に対して CPU の優先レベル制御が有効になります。 0 : CPU は常に優先レベル最低 1 : CPU の優先レベル制御は有効
6	DTCP2	0	R/W	DTC プライオリティレベル 2~0
5	DTCP1	0	R/W	DTC の優先レベルを設定します。
4	DTCP0	0	R/W	000 : 優先レベル 0 (最低) 001 : 優先レベル 1 010 : 優先レベル 2 011 : 優先レベル 3 100 : 優先レベル 4 101 : 優先レベル 5 110 : 優先レベル 6 111 : 優先レベル 7 (最高)
3	IPSETE	0	R/W	インタラプトプライオリティセットイネーブル 割り込み優先順位を、自動的に CPU の優先レベルに設定する機能を制御します。このビットを 1 にセットすると、CPU の割り込みマスクビット (CCR の I、EXR の I2~I0) の値により、CPUP2~CPUP0 ビットを自動的に設定します。 0 : CPUP2~CPUP0 ビットは自動更新しない 1 : 割り込みマスクビットの値を CPUP2~CPUP0 ビットに反映する

ビット	ビット名	初期値	R/W	説明
2	CPUP2	0	R/(W)*	CPU プライオリティレベル 2~0 CPU の優先レベルを設定します。CPUPCE が 1 のとき、DTC、DMAC に対して CPU の優先レベル制御が有効になり、CPUP2~CPUP0 ビットの設定に従って CPU の処理の優先レベルが設定されます。 000 : 優先レベル 0 (最低) 001 : 優先レベル 1 010 : 優先レベル 2 011 : 優先レベル 3 100 : 優先レベル 4 101 : 優先レベル 5 110 : 優先レベル 6 111 : 優先レベル 7 (最高)
1	CPUP1	0	R/(W)*	
0	CPUP0	0	R/(W)*	

【注】 * IPSETE ビットが 1 にセットされると、自動更新になるためライトは無効です。

5.3.3 インタラプトプライオリティレジスタ A~P (IPRA~IPRP)

IPR は、NMI を除く割り込み要因の優先順位（レベル 0~7）を設定します。ビット 14~12、ビット 10~8、ビット 6~4、ビット 2~0 の各 3 ビットに B'000 から B'111 の範囲の値を設定することによって、対応する割り込み要求の優先順位が決まります。各割り込み要因と IPR の対応については、表 5.2 参照してください。

ビット	15	14	13	12	11	10	9	8
ビット名	—	IPR14	IPR13	IPR12	—	IPR10	IPR9	IPR8
初期値：	0	1	1	1	0	1	1	1
R/W：	R	R/W	R/W	R/W	R	R/W	R/W	R/W
ビット	7	6	5	4	3	2	1	0
ビット名	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0
初期値：	0	1	1	1	0	1	1	1
R/W：	R	R/W	R/W	R/W	R	R/W	R/W	R/W

ビット	ビット名	初期値	R/W	説明
15	-	0	R	リザーブビット リードすると常に 0 が読み出されます。ライトは無効です。
14	IPR14	1	R/W	対応する割り込み要因の優先順位を設定します。 000：優先レベル 0（最低） 001：優先レベル 1 010：優先レベル 2 011：優先レベル 3 100：優先レベル 4 101：優先レベル 5 110：優先レベル 6 111：優先レベル 7（最高）
13	IPR13	1	R/W	
12	IPR12	1	R/W	
11	-	0	R	リザーブビット リードすると常に 0 が読み出されます。ライトは無効です。
10	IPR10	1	R/W	対応する割り込み要因の優先順位を設定します。 000：優先レベル 0（最低） 001：優先レベル 1 010：優先レベル 2 011：優先レベル 3 100：優先レベル 4 101：優先レベル 5 110：優先レベル 6 111：優先レベル 7（最高）
9	IPR9	1	R/W	
8	IPR8	1	R/W	

ビット	ビット名	初期値	R/W	説明
7	-	0	R	リザーブビット リードすると常に0が読み出されます。ライトは無効です。
6	IPR6	1	R/W	対応する割り込み要因の優先順位を設定します。 000 : 優先レベル0 (最低) 001 : 優先レベル1 010 : 優先レベル2 011 : 優先レベル3 100 : 優先レベル4 101 : 優先レベル5 110 : 優先レベル6 111 : 優先レベル7 (最高)
5	IPR5	1	R/W	
4	IPR4	1	R/W	
3	-	0	R	リザーブビット リードすると常に0が読み出されます。ライトは無効です。
2	IPR2	1	R/W	対応する割り込み要因の優先順位を設定します。 000 : 優先レベル0 (最低) 001 : 優先レベル1 010 : 優先レベル2 011 : 優先レベル3 100 : 優先レベル4 101 : 優先レベル5 110 : 優先レベル6 111 : 優先レベル7 (最高)
1	IPR1	1	R/W	
0	IPR0	1	R/W	

5.3.4 IRQ イネーブルレジスタ (IER)

IER は、IRQ14~IRQ0 割り込み要求をイネーブルにします。

ビット	15	14	13	12	11	10	9	8
ビット名	—	IRQ14E	IRQ13E	IRQ12E	IRQ11E	IRQ10E	IRQ9E	IRQ8E
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
ビット	7	6	5	4	3	2	1	0
ビット名	IRQ7E	IRQ6E	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E
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

ビット	ビット名	初期値	R/W	説明
15	-	0	R/W	リザーブビット リードすると常に0が読み出されます。ライトする値は常に0にしてください。
14	IRQ14E	0	R/W	IRQ14 イネーブル このビットが1のとき IRQ14 割り込み要求がイネーブルになります。
13	IRQ13E	0	R/W	IRQ13 イネーブル このビットが1のとき IRQ13 割り込み要求がイネーブルになります。
12	IRQ12E	0	R/W	IRQ12 イネーブル このビットが1のとき IRQ12 割り込み要求がイネーブルになります。
11	IRQ11E	0	R/W	IRQ11 イネーブル このビットが1のとき IRQ11 割り込み要求がイネーブルになります。
10	IRQ10E	0	R/W	IRQ10 イネーブル このビットが1のとき IRQ10 割り込み要求がイネーブルになります。
9	IRQ9E	0	R/W	IRQ9 イネーブル このビットが1のとき IRQ9 割り込み要求がイネーブルになります。
8	IRQ8E	0	R/W	IRQ8 イネーブル このビットが1のとき IRQ8 割り込み要求がイネーブルになります。
7	IRQ7E	0	R/W	IRQ7 イネーブル このビットが1のとき IRQ7 割り込み要求がイネーブルになります。
6	IRQ6E	0	R/W	IRQ6 イネーブル このビットが1のとき IRQ6 割り込み要求がイネーブルになります。
5	IRQ5E	0	R/W	IRQ5 イネーブル このビットが1のとき IRQ5 割り込み要求がイネーブルになります。
4	IRQ4E	0	R/W	IRQ4 イネーブル このビットが1のとき IRQ4 割り込み要求がイネーブルになります。

ビット	ビット名	初期値	R/W	説明
3	IRQ3E	0	R/W	IRQ3 イネーブル このビットが1のとき IRQ3 割り込み要求がイネーブルになります。
2	IRQ2E	0	R/W	IRQ2 イネーブル このビットが1のとき IRQ2 割り込み要求がイネーブルになります。
1	IRQ1E	0	R/W	IRQ1 イネーブル このビットが1のとき IRQ1 割り込み要求がイネーブルになります。
0	IRQ0E	0	R/W	IRQ0 イネーブル このビットが1のとき IRQ0 割り込み要求がイネーブルになります。

5.3.5 IRQ センسコントロールレジスタ H、L (ISCRH、ISCRL)

ISCR は、 $\overline{\text{IRQ14}} \sim \overline{\text{IRQ0}}$ 入力から割り込み要求を発生させる要因を選択します。

ISCR の設定変更時に、内部動作により ISR の IRQnF ($n=0 \sim 14$) が意図しないで1にセットされる場合があります。このとき、 IRQn 割り込み要求が許可されていると割り込み例外処理を実行します。この意図しない割り込みを防ぐには、ISCR の設定変更を IRQn 割り込みを禁止した状態で行い、その後 ISR の IRQnF を0にクリアしてください。

・ ISCRH

ビット	15	14	13	12	11	10	9	8
ビット名	—	—	IRQ14SR	IRQ14SF	IRQ13SR	IRQ13SF	IRQ12SR	IRQ12SF
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
ビット	7	6	5	4	3	2	1	0
ビット名	IRQ11SR	IRQ11SF	IRQ10SR	IRQ10SF	IRQ9SR	IRQ9SF	IRQ8SR	IRQ8SF
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

・ ISCRL

ビット	15	14	13	12	11	10	9	8
ビット名	IRQ7SR	IRQ7SF	IRQ6SR	IRQ6SF	IRQ5SR	IRQ5SF	IRQ4SR	IRQ4SF
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
ビット	7	6	5	4	3	2	1	0
ビット名	IRQ3SR	IRQ3SF	IRQ2SR	IRQ2SF	IRQ1SR	IRQ1SF	IRQ0SR	IRQ0SF
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

• ISCRH

ビット	ビット名	初期値	R/W	説明
15	—	0	R/W	リザーブビット
14	—	0	R/W	リードすると常に0が読み出されます。ライトする値は常に0にしてください。
13	IRQ14SR	0	R/W	IRQ14 センスコントロールライズ
12	IRQ14SF	0	R/W	IRQ14 センスコントロールフォール 00: $\overline{\text{IRQ14}}$ 入力の Low レベルで割り込み要求を発生 01: $\overline{\text{IRQ14}}$ 入力の立ち下がリエッジで割り込み要求を発生 10: $\overline{\text{IRQ14}}$ 入力の立ち上がりエッジで割り込み要求を発生 11: $\overline{\text{IRQ14}}$ 入力の立ち下がり、立ち上がりの両エッジで割り込み要求を発生
11	IRQ13SR	0	R/W	IRQ13 センスコントロールライズ
10	IRQ13SF	0	R/W	IRQ13 センスコントロールフォール 00: $\overline{\text{IRQ13}}$ 入力の Low レベルで割り込み要求を発生 01: $\overline{\text{IRQ13}}$ 入力の立ち下がリエッジで割り込み要求を発生 10: $\overline{\text{IRQ13}}$ 入力の立ち上がりエッジで割り込み要求を発生 11: $\overline{\text{IRQ13}}$ 入力の立ち下がり、立ち上がりの両エッジで割り込み要求を発生
9	IRQ12SR	0	R/W	IRQ12 センスコントロールライズ
8	IRQ12SF	0	R/W	IRQ12 センスコントロールフォール 00: $\overline{\text{IRQ12}}$ 入力の Low レベルで割り込み要求を発生 01: $\overline{\text{IRQ12}}$ 入力の立ち下がリエッジで割り込み要求を発生 10: $\overline{\text{IRQ12}}$ 入力の立ち上がりエッジで割り込み要求を発生 11: $\overline{\text{IRQ12}}$ 入力の立ち下がり、立ち上がりの両エッジで割り込み要求を発生
7	IRQ11SR	0	R/W	IRQ11 センスコントロールライズ
6	IRQ11SF	0	R/W	IRQ11 センスコントロールフォール 00: $\overline{\text{IRQ11}}$ 入力の Low レベルで割り込み要求を発生 01: $\overline{\text{IRQ11}}$ 入力の立ち下がリエッジで割り込み要求を発生 10: $\overline{\text{IRQ11}}$ 入力の立ち上がりエッジで割り込み要求を発生 11: $\overline{\text{IRQ11}}$ 入力の立ち下がり、立ち上がりの両エッジで割り込み要求を発生
5	IRQ10SR	0	R/W	IRQ10 センスコントロールライズ
4	IRQ10SF	0	R/W	IRQ10 センスコントロールフォール 00: $\overline{\text{IRQ10}}$ 入力の Low レベルで割り込み要求を発生 01: $\overline{\text{IRQ10}}$ 入力の立ち下がリエッジで割り込み要求を発生 10: $\overline{\text{IRQ10}}$ 入力の立ち上がりエッジで割り込み要求を発生 11: $\overline{\text{IRQ10}}$ 入力の立ち下がり、立ち上がりの両エッジで割り込み要求を発生

ビット	ビット名	初期値	R/W	説明
3	IRQ9SR	0	R/W	IRQ9 センスコントロールライズ
2	IRQ9SF	0	R/W	IRQ9 センスコントロールフォール 00: $\overline{\text{IRQ9}}$ 入力の Low レベルで割り込み要求を発生 01: $\overline{\text{IRQ9}}$ 入力の立ち下がリエッジで割り込み要求を発生 10: $\overline{\text{IRQ9}}$ 入力の立ち上がリエッジで割り込み要求を発生 11: $\overline{\text{IRQ9}}$ 入力の立ち下がり、立ち上がりの両エッジで割り込み要求を発生
1	IRQ8SR	0	R/W	IRQ8 センスコントロールライズ
0	IRQ8SF	0	R/W	IRQ8 センスコントロールフォール 00: $\overline{\text{IRQ8}}$ 入力の Low レベルで割り込み要求を発生 01: $\overline{\text{IRQ8}}$ 入力の立ち下がリエッジで割り込み要求を発生 10: $\overline{\text{IRQ8}}$ 入力の立ち上がリエッジで割り込み要求を発生 11: $\overline{\text{IRQ8}}$ 入力の立ち下がり、立ち上がりの両エッジで割り込み要求を発生

- ISCR1

ビット	ビット名	初期値	R/W	説明
15	IRQ7SR	0	R/W	IRQ7 センスコントロールライズ
14	IRQ7SF	0	R/W	IRQ7 センスコントロールフォール 00: $\overline{\text{IRQ7}}$ 入力の Low レベルで割り込み要求を発生 01: $\overline{\text{IRQ7}}$ 入力の立ち下がリエッジで割り込み要求を発生 10: $\overline{\text{IRQ7}}$ 入力の立ち上がリエッジで割り込み要求を発生 11: $\overline{\text{IRQ7}}$ 入力の立ち下がり、立ち上がりの両エッジで割り込み要求を発生
13	IRQ6SR	0	R/W	IRQ6 センスコントロールライズ
12	IRQ6SF	0	R/W	IRQ6 センスコントロールフォール 00: $\overline{\text{IRQ6}}$ 入力の Low レベルで割り込み要求を発生 01: $\overline{\text{IRQ6}}$ 入力の立ち下がリエッジで割り込み要求を発生 10: $\overline{\text{IRQ6}}$ 入力の立ち上がリエッジで割り込み要求を発生 11: $\overline{\text{IRQ6}}$ 入力の立ち下がり、立ち上がりの両エッジで割り込み要求を発生
11	IRQ5SR	0	R/W	IRQ5 センスコントロールライズ
10	IRQ5SF	0	R/W	IRQ5 センスコントロールフォール 00: $\overline{\text{IRQ5}}$ 入力の Low レベルで割り込み要求を発生 01: $\overline{\text{IRQ5}}$ 入力の立ち下がリエッジで割り込み要求を発生 10: $\overline{\text{IRQ5}}$ 入力の立ち上がリエッジで割り込み要求を発生 11: $\overline{\text{IRQ5}}$ 入力の立ち下がり、立ち上がりの両エッジで割り込み要求を発生

ビット	ビット名	初期値	R/W	説明
9	IRQ4SR	0	R/W	IRQ4 センスコントロールライズ
8	IRQ4SF	0	R/W	IRQ4 センスコントロールフォール 00: $\overline{\text{IRQ4}}$ 入力の Low レベルで割り込み要求を発生 01: $\overline{\text{IRQ4}}$ 入力の立ち下がリエッジで割り込み要求を発生 10: $\overline{\text{IRQ4}}$ 入力の立ち上がリエッジで割り込み要求を発生 11: $\overline{\text{IRQ4}}$ 入力の立ち下がり、立ち上がりの両エッジで割り込み要求を発生
7	IRQ3SR	0	R/W	IRQ3 センスコントロールライズ
6	IRQ3SF	0	R/W	IRQ3 センスコントロールフォール 00: $\overline{\text{IRQ3}}$ 入力の Low レベルで割り込み要求を発生 01: $\overline{\text{IRQ3}}$ 入力の立ち下がリエッジで割り込み要求を発生 10: $\overline{\text{IRQ3}}$ 入力の立ち上がリエッジで割り込み要求を発生 11: $\overline{\text{IRQ3}}$ 入力の立ち下がり、立ち上がりの両エッジで割り込み要求を発生
5	IRQ2SR	0	R/W	IRQ2 センスコントロールライズ
4	IRQ2SF	0	R/W	IRQ2 センスコントロールフォール 00: $\overline{\text{IRQ2}}$ 入力の Low レベルで割り込み要求を発生 01: $\overline{\text{IRQ2}}$ 入力の立ち下がリエッジで割り込み要求を発生 10: $\overline{\text{IRQ2}}$ 入力の立ち上がリエッジで割り込み要求を発生 11: $\overline{\text{IRQ2}}$ 入力の立ち下がり、立ち上がりの両エッジで割り込み要求を発生
3	IRQ1SR	0	R/W	IRQ1 センスコントロールライズ
2	IRQ1SF	0	R/W	IRQ1 センスコントロールフォール 00: $\overline{\text{IRQ1}}$ 入力の Low レベルで割り込み要求を発生 01: $\overline{\text{IRQ1}}$ 入力の立ち下がリエッジで割り込み要求を発生 10: $\overline{\text{IRQ1}}$ 入力の立ち上がリエッジで割り込み要求を発生 11: $\overline{\text{IRQ1}}$ 入力の立ち下がり、立ち上がりの両エッジで割り込み要求を発生
1	IRQ0SR	0	R/W	IRQ0 センスコントロールライズ
0	IRQ0SF	0	R/W	IRQ0 センスコントロールフォール 00: $\overline{\text{IRQ0}}$ 入力の Low レベルで割り込み要求を発生 01: $\overline{\text{IRQ0}}$ 入力の立ち下がリエッジで割り込み要求を発生 10: $\overline{\text{IRQ0}}$ 入力の立ち上がリエッジで割り込み要求を発生 11: $\overline{\text{IRQ0}}$ 入力の立ち下がり、立ち上がりの両エッジで割り込み要求を発生

5.3.6 IRQ ステータスレジスタ (ISR)

ISR は、IRQ14~IRQ0 割り込み要求レジスタです。

ビット	15	14	13	12	11	10	9	8
ビット名	—	IRQ14F	IRQ13F	IRQ12F	IRQ11F	IRQ10F	IRQ9F	IRQ8F
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)*
ビット	7	6	5	4	3	2	1	0
ビット名	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
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】 * フラグをクリアするための0ライトのみ可能です。

ビット	ビット名	初期値	R/W	説明
15	—	0	R/W	リザーブビット リードすると常に0が読み出されます。ライトする値は常に0にしてください。
14	IRQ14F	0	R/(W)*	[セット条件]
13	IRQ13F	0	R/(W)*	• ISCR で選択した割り込み要因が発生したとき
12	IRQ12F	0	R/(W)*	[クリア条件]
11	IRQ11F	0	R/(W)*	• 1 の状態をリードした後、0 をライトしたとき
10	IRQ10F	0	R/(W)*	• Low レベル検出設定の状態、かつ $\overline{\text{IRQn}}$ 入力が High レベルの状態での割り込み例外処理を実行したとき
9	IRQ9F	0	R/(W)*	
8	IRQ8F	0	R/(W)*	• 立ち下がりエッジ、立ち上がりエッジ、両エッジ検出設定時の状態で IRQn 割り込み例外処理を実行したとき
7	IRQ7F	0	R/(W)*	
6	IRQ6F	0	R/(W)*	• IRQn 割り込みにより DTC が起動され、DTC の MRB の DISEL が 0 のとき
5	IRQ5F	0	R/(W)*	
4	IRQ4F	0	R/(W)*	
3	IRQ3F	0	R/(W)*	
2	IRQ2F	0	R/(W)*	
1	IRQ1F	0	R/(W)*	
0	IRQ0F	0	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。

5.3.7 ソフトウェアスタンバイ解除 IRQ イネーブルレジスタ (SSIER)

SSIER は、ソフトウェアスタンバイ状態から復帰するために使用する $\overline{\text{IRQ14}} \sim \overline{\text{IRQ0}}$ 端子を設定します。

ソフトウェアスタンバイ状態から復帰するために使用する IRQ 割り込みは、DTC 起動要因として設定しないでください。

ビット	15	14	13	12	11	10	9	8
ビット名	—	SSI14	SSI13	SSI12	SSI11	SSI10	SSI9	SSI8
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
ビット	7	6	5	4	3	2	1	0
ビット名	SSI7	SSI6	SSI5	SSI4	SSI3	SSI2	SSI1	SSI0
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

ビット	ビット名	初期値	R/W	説明
15	—	0	R/W	リザーブビット リードすると常に0が読み出されます。ライトする値は常に0にしてください。
14	SSI14	0	R/W	ソフトウェアスタンバイ解除 IRQ 設定 SSIn ビットは、ソフトウェアスタンバイ状態から復帰するために使用する $\overline{\text{IRQn}}$ 端子を設定します (n=14~0)。 0 : IRQn 割り込み要求は、ソフトウェアスタンバイ状態ではサンプリングされません。 1 : ソフトウェアスタンバイ状態で IRQn 割り込み要求があると、発振安定時間を経てソフトウェアスタンバイ状態から復帰します。
13	SSI13	0	R/W	
12	SSI12	0	R/W	
11	SSI11	0	R/W	
10	SSI10	0	R/W	
9	SSI9	0	R/W	
8	SSI8	0	R/W	
7	SSI7	0	R/W	
6	SSI6	0	R/W	
5	SSI5	0	R/W	
4	SSI4	0	R/W	
3	SSI3	0	R/W	
2	SSI2	0	R/W	
1	SSI1	0	R/W	
0	SSI0	0	R/W	

5.4 割り込み要因

5.4.1 外部割り込み要因

外部割り込みには、NMI、IRQ14～IRQ0 の 16 要因があります。このうち、外部割り込みはソフトウェアスタンバイモードからの復帰に使用できます。

(1) NMI 割り込み

ノンマスクابل割り込み要求 NMI は最優先の外部割り込み要求で、割り込み制御モードや CPU の割り込みマスクビットの状態にかかわらず常に受け付けられます。NMI 割り込み要求を NMI 入力の立ち上がりエッジと、立ち下がりエッジのどちらで発生させるかは、INTCR の NMIEG ビットで選択できます。

NMI 割り込みが発生すると、割り込みコントローラはエラー発生と見なして次の制御を行います。

- DTCのDTCCRのERRビットを1にセット
- DMACのDMDR_0のERRFビットを1にセット
- DMACの全チャンネルのDMDRのDTEビットを0にクリア、および強制転送終了

(2) IRQn 割り込み

IRQn 割り込みは、 $\overline{\text{IRQn}}$ 入力により割り込み要求を発生します (n=14～0)。

IRQn 割り込みには以下の特長があります。

- $\overline{\text{IRQn}}$ 入力のLowレベル、立ち下がりエッジ、立ち上がりエッジおよび両エッジのいずれで割り込み要求を発生させるか、ISCRで選択できます。
- IRQn割り込み要求は、IERにより選択できます。
- IPRにより、割り込み要因の優先順位を設定できます。
- IRQn割り込み要求のステータスは、ISRに表示されます。ISRのフラグは、ソフトウェアで0にクリアすることができます。ISRのフラグのクリアは、ビット操作命令またはメモリ演算命令を使用してください。

IRQn 割り込みの検出は P1ICR、P5ICR、P6ICR の設定により有効となり、当該端子の出力の設定に依存しません。したがって、外部割り込み入力端子として使用する場合には、対応する DDR を 0 にクリアして、そのほかの機能の入出力端子として使用しないでください。

IRQn 割り込みのブロック図を図 5.2 に示します。

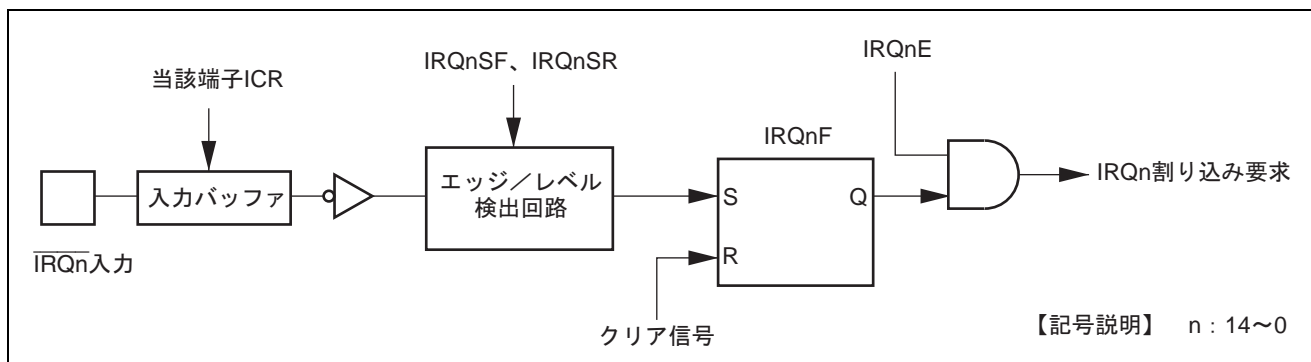


図 5.2 IRQn 割り込みのブロック図

IRQn 割り込み要求を ISCR の設定により $\overline{\text{IRQn}}$ 入力の Low レベルで発生するようにした場合、割り込み要求時には当該 $\overline{\text{IRQn}}$ 入力を割り込み処理が開始されるまで Low レベルに保持してください。その後、割り込み処理ルーチン内で、当該 $\overline{\text{IRQn}}$ 入力を High レベルに戻し、かつ IRQnF を 0 にクリアしてください。割り込み処理が開始される前に、当該 $\overline{\text{IRQn}}$ 入力を High レベルに戻すと、当該割り込みが実行されない場合があります。

5.4.2 内部割り込み

内蔵周辺モジュールからの内部割り込み要因には以下の特長があります。

- 各内蔵周辺モジュールには、割り込み要求のステータスを表示するフラグとこれらの割り込みイネーブルビットがあり、独立に制御することができます。イネーブルビットが1のとき、割り込み要求が割り込みコントローラに送られます。
- IPRによって、割り込み要因の優先順位を設定できます。
- TPU、SCIなどの割り込み要求により、DTC、DMACを起動することができます。
- DTC、DMACの起動は、CPUに対するDTC、DMACの優先レベル制御機能により優先レベルを設定できます。

5.5 割り込み例外処理ベクタテーブル

割り込み例外処理要因とベクタテーブルアドレスオフセットおよび割り込み優先順位の一覧を表 5.2 に示します。デフォルトの優先順位はベクタ番号の小さいものほど高くなっています。割り込み制御モード 2 に設定されている場合、優先順位は IPR 設定単位ごとに IPR により変更することができます。同一優先順位の IPR に設定された割り込み要因は、デフォルトの優先順位に従います。同一優先順位の IPR 設定単位内の優先順位は固定されています。

表 5.2 割り込み要因とベクタアドレスオフセットおよび割り込み優先順位

割り込み要因 発生元	名称	ベクタ 番号	ベクタテーブル アドレス オフセット*	IPR	優先 順位	DTC 起動	DMAC 起動
			アドバンスモード				
外部端子	NMI	7	H'001C	—	↑ 高	—	—
	IRQ0	64	H'0100	IPRA14~IPRA12		○	—
	IRQ1	65	H'0104	IPRA10~IPRA8		○	—
	IRQ2	66	H'0108	IPRA6~IPRA4		○	—
	IRQ3	67	H'010C	IPRA2~IPRA0		○	—
	IRQ4	68	H'0110	IPRB14~IPRB12		○	—
	IRQ5	69	H'0114	IPRB10~IPRB8		○	—
	IRQ6	70	H'0118	IPRB6~IPRB4		○	—
	IRQ7	71	H'011C	IPRB2~IPRB0		○	—
	IRQ8	72	H'0120	IPRC14~IPRC12		○	—
	IRQ9	73	H'0124	IPRC10~IPRC8		○	—
	IRQ10	74	H'0128	IPRC6~IPRC4		○	—
	IRQ11	75	H'012C	IPRC2~IPRC0		○	—
	IRQ12	76	H'0130	IPRD14~IPRD12		○	—
	IRQ13	77	H'0134	IPRD10~IPRD8		○	—
IRQ14	78	H'0138	IPRD6~IPRD4	○	—		
—	システム予約	79	H'013C	IPRD2~IPRD0	—	—	
—	システム予約	80	H'0140	IPRE14~IPRE12	—	—	
WDT	WOVI	81	H'0144	IPRE10~IPRE8	—	—	
メモリエラー	RAME	82	H'0148	IPRE6~IPRE4	—	—	
	システム予約	83	H'014C		—	—	
	システム予約	84	H'0150		—	—	
	FIFE	85	H'0154		低	—	—

割り込み要因 発生元	名称	ベクタ 番号	ベクタテーブル アドレス オフセット*	IPR	優先 順位	DTC 起動	DMAC 起動
			アドバンスモード				
A/D_0	ADI0	86	H'0158	IPRF10~IPRF8	高 ↑	○	○
A/D_1	ADI1	87	H'015C			○	○
TPU_0	TGI0A	88	H'0160	IPRF6~IPRF4		○	○
	TGI0B	89	H'0164			○	—
	TGI0C	90	H'0168			○	—
	TGI0D	91	H'016C			○	—
	TCI0V	92	H'0170			—	—
TPU_1	TGI1A	93	H'0174	IPRF2~IPRF0		○	○
	TGI1B	94	H'0178			○	—
	TCI1V	95	H'017C			—	—
	TCI1U	96	H'0180		—	—	
TPU_2	TGI2A	97	H'0184	IPRG14~IPRG12	○	○	
	TGI2B	98	H'0188		○	—	
	TCI2V	99	H'018C		—	—	
	TCI2U	100	H'0190		—	—	
TPU_3	TGI3A	101	H'0194	IPRG10~IPRG8	○	○	
	TGI3B	102	H'0198		○	—	
	TGI3C	103	H'019C		○	—	
	TGI3D	104	H'01A0		○	—	
	TCI3V	105	H'01A4		—	—	
TPU_4	TGI4A	106	H'01A8	IPRG6~IPRG4	○	○	
	TGI4B	107	H'01AC		○	—	
	TCI4V	108	H'01B0		—	—	
	TCI4U	109	H'01B4		—	—	
TPU_5	TGI5A	110	H'01B8	IPRG2~IPRG0	○	○	
	TGI5B	111	H'01BC		○	—	
	TCI5V	112	H'01C0		—	—	
	TCI5U	113	H'01C4		—	—	
クロック発振器	OSCERI	114	H'01C8	IPRE2~IPRE0	—	—	
—	システム予約	115	H'01CC	—	—	—	
—	システム予約	116	H'01D0	—	—	—	
		117	H'01D4		—	—	
		118	H'01D8		—	—	
					低		

割り込み要因 発生元	名称	ベクタ 番号	ベクタテーブル アドレス オフセット*	IPR	優先 順位	DTC 起動	DMAC 起動
			アドバンスモード				
-	システム予約	119	H'01DC	-	↑ 高	-	-
		120	H'01E0			-	-
		121	H'01E4			-	-
-	システム予約	122	H'01E8	-		-	-
	システム予約	123	H'01EC			-	-
	システム予約	124	H'01F0			-	-
-	システム予約	125	H'01F4	-		-	-
	システム予約	126	H'01F8			-	-
	システム予約	127	H'01FC			-	-
DMAC	DMTEND0	128	H'0200	IPRI14~IPRI12		○	-
	DMTEND1	129	H'0204	IPRI10~IPRI8		○	-
	DMTEND2	130	H'0208	IPRI6~IPRI4		○	-
	DMTEND3	131	H'020C	IPRI2~IPRI0		○	-
RCAN-TL1_0	RM0_0	132	H'0210	IPRJ14~IPRJ12		○	○
	ERS0_0/OVR0_0/ RM1_0/SLE0_0	133	H'0214			○	-
RCAN-TL1_1	RM0_1	134	H'0218	IPRJ10~IPRJ8		○	○
	ERS0_1/OVR0_1/ RM1_1/SLE0_1	135	H'021C			○	-
DMAC	DMEEND0	136	H'0220	IPRJ6~IPRJ4	○	-	
	DMEEND1	137	H'0224		○	-	
	DMEEND2	138	H'0228		○	-	
	DMEEND3	139	H'022C		○	-	
HW_LIN	HW_LIN	140	H'0230	IPRJ2~IPRJ0	-	-	
-	システム予約	141	H'0234	-	-	-	
		142	H'0238		-	-	
-	システム予約	143	H'023C	-	-	-	
-	システム予約	144	H'0240	-	-	-	
		145	H'0244		-	-	
		146	H'0248		-	-	
		147	H'024C		-	-	
					↓ 低		

割り込み要因 発生元	名称	ベクタ 番号	ベクタテーブル アドレス オフセット*	IPR	優先 順位	DTC 起動	DMAC 起動
			アドバンスモード				
TPU_10	TGI10A	182	H'02D8	IPRN2~IPRN0	↑ 高	○	○
	TGI10B	183	H'02DC			○	—
	システム予約	184	H'02E0			○	—
	システム予約	185	H'02E4			○	—
	TCI10V	186	H'02E8	IPRO14~IPRO12		○	—
	TCI10U or システム予約	187	H'02EC			—	—
TPU_11	TGI11A	188	H'02F0	IPRO10~IPRO8		○	○
	TGI11B	189	H'02F4	IPRO6~IPRO4		○	—
	TCI11V	190	H'02F8			—	—
	TCI11U	191	H'02FC	—		—	
—	システム予約	192	H'0300	—	—	—	
		193	H'0304		—	—	
		194	H'0308		—	—	
		195	H'030C		—	—	
RSPI_0	SPEI_0	196	H'0310	IPRP14~IPRP12	—	—	
	SPRI_0	197	H'0314		○	○	
	SPTI_0	198	H'0318		○	○	
RSPI_1	SPEI_1	199	H'031C	IPRP10~IPRP8	—	—	
	SPRI_1	200	H'0320		○	○	
	SPTI_1	201	H'0324		○	○	
RSPI_2	SPEI_2	202	H'0328	IPRP6~IPRP4	—	—	
	SPRI_2	203	H'032C		○	○	
	SPTI_2	204	H'0330		○	○	
RSPI_3	SPEI_3	205	H'0334	IPRP2~IPRP0	—	—	
	SPRI_3	206	H'0338		○	○	
	SPTI_3	207	H'033C		○	○	
—	システム予約	208	H'0340	—	—	—	
		209	H'0344		—	—	
		210	H'0348		—	—	
		211	H'034C		—	—	—
		212	H'0350	—		—	
		213	H'0354	—		—	
		214	H'0358	—		—	
		215	H'035C	—	—		
					低		

割り込み要因発生元	名称	ベクタ番号	ベクタテーブルアドレス オフセット*	IPR	優先順位	DTC起動	DMAC起動
			アドバンストモード				
—	システム予約	251	H'03EC	—	↑ 高 低	—	—
		252	H'03F0			—	—
		253	H'03F4			—	—
		254	H'03F8			—	—
		255	H'03FC			—	—

【注】 * 先頭アドレスの下位 16 ビットを示しています。

5.6 割り込み制御モードと割り込み動作

割り込みコントローラには、割り込み制御モード 0 と割り込み制御モード 2 の 2 種類の割り込み制御モードがあります。割り込み制御モードの選択は、INTCR で行います。割り込み制御モード 0 と割り込み制御モード 2 の相違点を表 5.3 に示します。

表 5.3 割り込み制御モード

割り込み制御モード	割り込み優先順位	割り込みマスクビット	説明
0	デフォルト	I	各割り込み要因の優先順位は、デフォルトで固定されています。 NMI を除く割り込み要因は、I ビットによりマスクされます。
2	IPR	I2~I0	IPR により、NMI を除く各割り込み要因に 8 レベルの優先順位を設定できます。 I2~I0 ビットにより、8 レベルの割り込みマスク制御を行います。

5.6.1 割り込み制御モード 0

割り込み制御モード 0 では、NMI を除く割り込み要求は、CPU の CCR の I ビットによってマスクされます。割り込み受け付け動作のフローチャートを図 5.3 に示します。

1. 割り込みイネーブルビットが I にセットされている割り込み要因が発生すると、割り込み要求が割り込みコントローラに送られます。
2. CCR の I ビットが I にセットされているときは、割り込みコントローラは NMI 以外の割り込み要求を保留します。I ビットが 0 にクリアされているときは、割り込み要求を受け付けます。
3. 複数の割り込み要求があるときは、割り込みコントローラは優先順位に従って最も優先度の高い割り込み要求を選択して、CPU に対して割り込み処理を要求し、その他は保留します。
4. CPU は割り込み要求を受け付けると、実行中の命令の処理が終了した後、割り込み例外処理を開始します。
5. 割り込み例外処理によって、PC と CCR がスタック領域に退避されます。PC にはリターン後に実行する最初の命令のアドレスが退避されます。
6. CCR の I ビットを I にセットします。これにより、NMI を除く割り込みがマスクされます。
7. CPU は受け付けた割り込み要求に対応するベクタアドレスを生成し、ベクタテーブルから割り込みルーチン開始アドレスを読み取って割り込み処理を開始します。

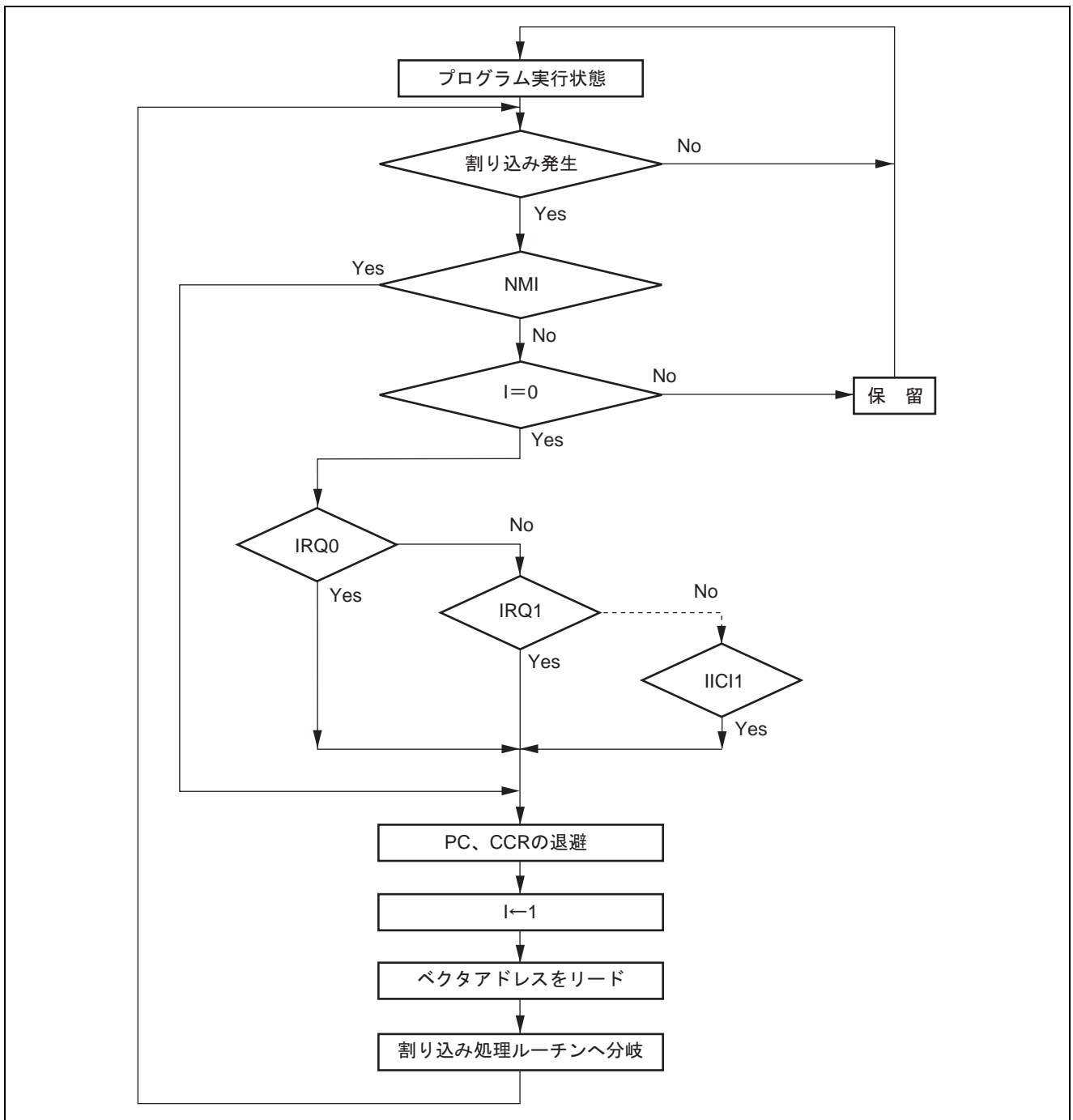


図 5.3 割り込み制御モード 0 の割り込み受け付けまでのフロー

5.6.2 割り込み制御モード2

割り込み制御モード2では、NMIを除く割り込み要求は、CPUのEXRの割り込みマスクレベル（I2～I0ビット）とIPRとの比較によって8レベルのマスク制御を行います。割り込み受け付け動作のフローチャートを図5.4に示します。

1. 割り込みイネーブルビットが1にセットされている割り込み要因が発生すると、割り込み要求が割り込みコントローラに送られます。
2. 複数の割り込み要求があるときは、割り込みコントローラはIPRに設定された割り込み要因の優先順位に従って最も優先レベルの高い割り込みを選択し、それよりも優先レベルの低い割り込み要求は保留します。優先レベルが同一の場合は、表5.2に示すデフォルトの優先順位に従って割り込み要求を選択します。
3. 選択した割り込み要求の優先順位とEXRの割り込みマスクレベルとを比較します。設定されていたマスクレベル以下であれば保留し、割り込みマスクレベルより優先順位が高ければCPUに対して割り込み処理を要求します。
4. CPUは割り込み要求を受け付けると、実行中の命令の処理が終了した後、割り込み例外処理を開始します。
5. 割り込み例外処理によって、PC、CCR、およびEXRがスタック領域に退避されます。PCにはリターン後に実行する最初の命令のアドレスが退避されます。
6. EXRのTビットが0にクリアされます。割り込みマスクレベルは受け付けた割り込みの優先レベルに書き換えられます。受け付けた割り込みがNMIのときは割り込みマスクレベルはH'7に設定されます。
7. CPUは受け付けた割り込み要求に対応するベクタアドレスを生成し、ベクタテーブルから割り込みルーチン開始アドレスを読み取って割り込み処理を開始します。

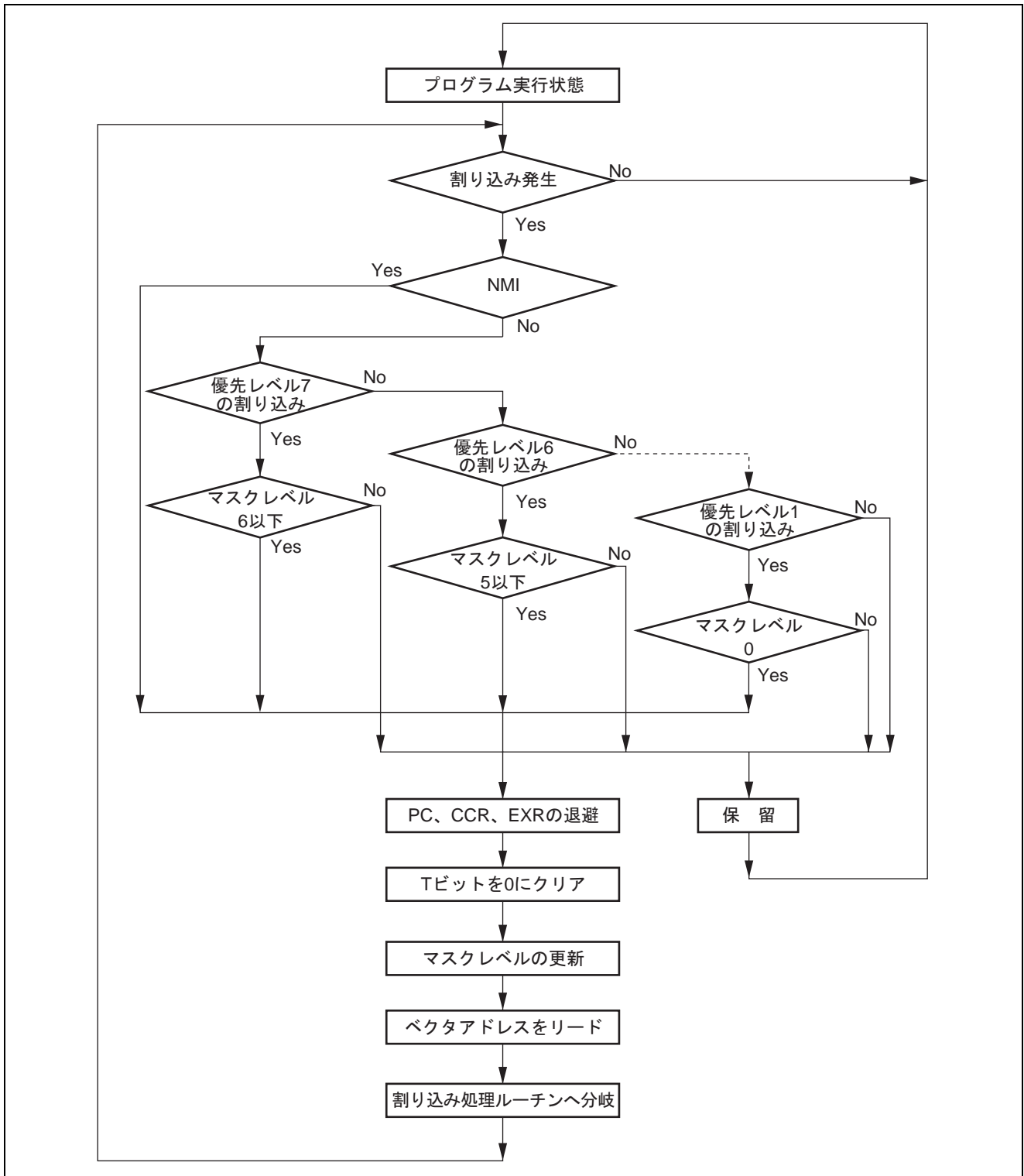


図 5.4 割り込み制御モード2の割り込み受け付けまでのフロー

5.6.3 割り込み例外処理シーケンス

割り込み例外処理シーケンスを図 5.5 に示します。アドバンスモードで割り込み制御モード 0、プログラム領域およびスタック領域が内蔵メモリの場合の例です。

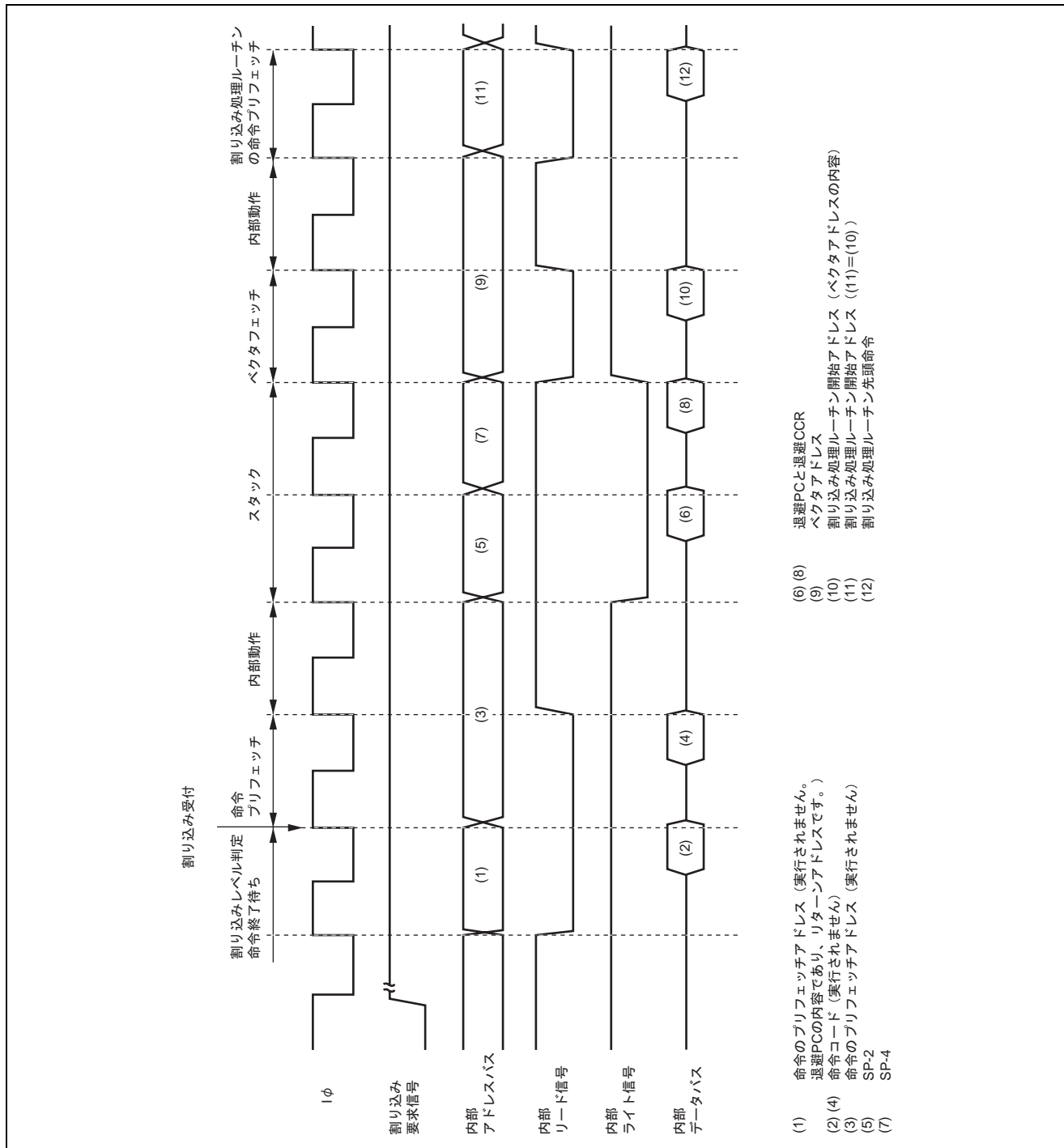


図 5.5 割り込み例外処理

5.6.4 割り込み応答時間

割り込み要求が発生してから、割り込み例外処理ルーチンの先頭命令が実行されるまでの割り込み応答時間を表 5.4 に示します。表 5.4 の実行状態の記号については表 5.5 を参照してください。本 LSI は内蔵メモリに対して高速ワードアクセスが可能のため、プログラム領域を内蔵 ROM、スタック領域を内蔵 RAM に設けることで処理速度の向上が図れます。

表 5.4 割り込み応答時間

No.	実行状態	アドバンストモード	
		割り込み制御 モード 0	割り込み制御 モード 2
1	割り込み優先順位判定*1	3	
2	実行中の命令が終了するまでの 待ちステート数*2	$1 \sim 19 + 2 \cdot S_i$	
3	PC、CCR および EXR のスタック	$S_k \sim 2 \cdot S_k^{*5}$	$2 \cdot S_k$
4	ベクタフェッチ	Sh	
5	命令フェッチ*3	$2 \cdot S_i$	
6	内部処理*4	2	
合計（内蔵メモリ使用時）		10 ~ 31	11 ~ 31

- 【注】 *1 内部割り込みの場合 2 ステートとなります。
 *2 MULXS、DIVXS 命令について示しています。
 *3 割り込み受け付け後のプリフェッチおよび割り込み処理ルーチンのプリフェッチです。
 *4 割り込み受け付け後の内部処理およびベクタフェッチ後の内部処理です。
 *5 SP の値を $4n$ に設定すると S_k 、 $4n+2$ に設定すると $2 \cdot S_k$ となります。

表 5.5 割り込み例外処理の実行状態のステート数

記号	アクセス対象				
	内部メモリ	外部デバイス			
		8ビットバス		16ビットバス	
		2ステートアクセス	3ステートアクセス	2ステートアクセス	3ステートアクセス
ベクタフェッチ Sh	1	8	$12 + 4m$	4	$6 + 2m$
命令フェッチ S_i	1	4	$6 + 2m$	2	$3 + m$
スタック操作 S_k	1	8	$12 + 4m$	4	$6 + 2m$

【記号説明】

m : 外部デバイスアクセス時のウェイトステート数

5.6.5 割り込みによる DTC、DMAC の起動

割り込み要求により、DTC、DMAC を起動することができます。この場合、以下の選択を行うことができます。

1. CPUに対する割り込み要求
2. DTCに対する起動要求
3. DMACに対する起動要求
4. 1～3.の複数の選択

なお、DTC、DMAC を起動できる割り込み要求については、表 5.2 および「第 7 章 DMA コントローラ (DMAC)」、「第 8 章 データトランスファコントローラ (DTC)」を参照してください。

図 5.6 に、DTC、DMAC と割り込みコントローラのブロック図を示します。

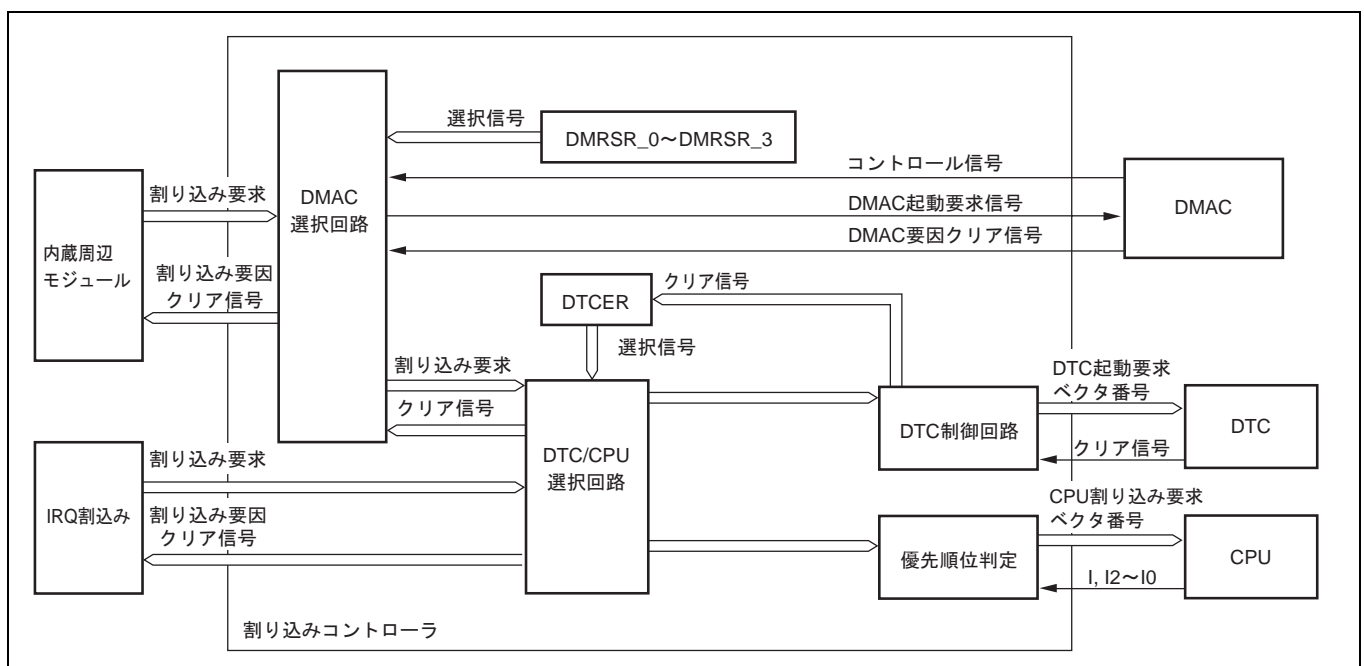


図 5.6 DTC、DMAC と割り込みコントローラ

(1) 割り込み要因の選択

DMAC の各チャンネルの起動要因は、DMRSR により選択します。選択した起動要因は、選択回路を經由して DMAC に入力されます。内蔵モジュール割り込みによる転送が有効になるように設定 (DMDR の DTF1=1、DTF0=0、DTA=1) して、DMDR の DTE ビットが 1 にセットされていると、その DMAC の起動要因になった割り込み要因は DMAC が管理することになり、DTC の起動要因および CPU の割り込み要因になりません。

DMAC に管理されている割り込み以外の割り込み要因は、DTC の DTCERA~DTCERG の DTCE ビットにより、DTC 起動要求とするか、CPU 割り込み要求とするかを選択します。

DTC の MRB の DISEL ビットの指定により、DTC のデータ転送後、DTCE ビットを 0 にクリアして、CPU に割り込みを要求することができます。

なお、DTC が所定回数のデータ転送を行い、転送カウンタが 0 になった場合には、DTC のデータ転送後、DTCE もビットを 0 にクリアして、CPU に割り込みを要求します。

同じ割り込み要因を、DTC の起動要因と CPU の割り込み要因に同時に設定する場合、DTC 優先レベルより CPU の優先レベルを高く設定しないでください。CPUPCR の IPSETE を 1 にセットしている場合は、割り込み要因に該当する IPR のレベルの設定が対象になります。必ず、(DMAP、DTCP) \geq (CPUP または割り込み要因に該当する IPR) となるように設定してください。CPU の優先レベルが高くなると、DTC、DMAC が起動されずに転送が行われなくなる場合があります。

(2) 優先順位判定

DTC の起動要因は、デフォルトの優先順位に従って選択されます。マスクレベルや優先レベルなどの影響を受けません。それぞれの優先順位は表 8.1 を参照してください。

(3) 動作順序

同一の割り込みを DTC の起動要因と CPU の割り込み要因に選択した場合、DTC のデータ転送終了後、CPU の割り込み例外処理を行います。同一の割り込みを DTC、DMAC の起動要因、または CPU の割り込み要因に選択した場合、各々独立に動作を行います。

表 5.6 に、DMAC の DMDR の DTA ビット、DTC の DTCERA～DTCERG の DTCE ビット、および DTC の MRB の DIESEL ビットの設定による割り込み要因の選択と割り込み要因クリア制御を示します。

表 5.6 割り込み要因の選択とクリア制御

設定内容			割り込み要因選択／クリア制御		
DMAC	DTC				
DTA	DTCE	DIESEL	DMAC	DTC	CPU
0	0	*		×	
	1	0			×
		1			
1	*	*		×	×
0	*	*	×	×	×

【記号説明】

- ◎ : 当該割り込みを使用します。割り込み要因のクリアを行います。
(CPU は割り込み処理ルーチンで、要因フラグをクリアしてください。)
- : 当該割り込みを使用します。割り込み要因をクリアしません。
- × : 当該割り込みは使用できません。
- * : Don't care

(4) 使用上の注意

SCI、RCAN、RSPI、TPU、および A/D 変換器の割り込み要因は、DTC または DMAC が所定のレジスタをリード／ライトすると表 5.6 の設定に従ってクリアされます。

同一の割り込みで、DTC、DMAC の複数のチャンネルを起動する場合は、同じプライオリティに設定してください。

5.7 CPU に対する DTC、DMAC の優先レベル制御機能

割り込みコントローラには、CPU の優先レベルに対して、DTC、DMAC 各々に優先レベルを設定することにより、DTC、DMAC と CPU との間の優先レベルを制御する機能があります。CPU の優先レベルは例外処理により自動設定することが可能で、CPU の割り込み例外処理時などに、DTC、DMAC の転送処理に対して優先実行が可能になります。

CPU の優先レベルは、CPUPCR の CPUP2～CPUP0 ビットで設定します。DTC の優先レベルは、CPUPCR の DTCP2～DTCP0 ビットで設定します。DMAC の優先レベルは、チャンネル毎に DMA モードコントロールレジスタ 0～3 (DMDR_0～DMDR_3) の DMAP2～DMAPO ビットで設定します。

CPU に対する DTC、DMAC の優先レベル制御機能は、CPUPCR の CPUPCE ビットを 1 にセットすると有効になります。CPUPCE ビットが 1 のとき、DTC、DMAC の起動要因は各々の優先レベルによって制御します。

DTC の起動要因は、CPUP2～CPUP0 ビットで示される CPU の優先レベルと、DTCP2～DTCP0 ビットで示される DTC の優先レベルにより制御します。CPU の優先レベルが高い場合は、DTC の起動要因は保留されます。保留された起動要因は、保留されている条件 (CPUPCE=1 かつ CPUP2～0>DTCP2～0) が解除されると DTC を起動します。DTC の優先レベルは、起動要因の区別がなく、CPUPCR の DTCP2～DTCP0 ビットで設定します。

DMAC の優先レベルは、チャンネルごとに設定できます。DMAC の起動要因は、対応するチャンネルの DMAP2～DMAPO ビットで設定される DMAC の優先レベルと、CPU の優先レベルにより制御します。CPU の優先レベルが高い場合は、当該チャンネルの起動要因は保留されます。保留された起動要因は、保留されている条件 (CPUPCE=1 かつ CPUP2～0>DMAP2～0) が解除されると有効になります。チャンネル毎に優先レベルを異なる値に設定した場合は、優先レベルの高いチャンネルは転送処理を継続して実行し、CPU よりも優先レベルが低いチャンネルのみが起動要因を保留されます。

CPU の優先レベルを設定する方法は、CPUPCR の IPSETE ビットにより 2 種類の方法を選択できます。IPSETE ビットを 1 にセットすると、CPU の割り込みマスクビットを自動的に優先レベルにセットする機能が有効になります。IPSETE ビットを 0 にクリアすると、優先レベルは自動更新されなくなり、CPUP2～CPUP0 ビットを直接ソフトウェアで書き換えて設定します。IPSETE ビットが 1 のときも CPU の割り込みマスクビット (CCR の I ビットまたは EXR の I2～I0 ビット) をソフトウェアで書き換えることで CPU の優先レベルを設定できます。

IPSETE ビットが 1 のときに自動設定する優先レベルは、割り込み制御モードにより値が異なります。

割り込み制御モード 0 の場合、CPU の CCR の I ビットの値を CPUP2 ビットに反映します。CPUP1～CPUP0 ビットは 0 固定です。割り込み制御モード 2 の場合、CPU の EXR の I2～I0 ビットの値を CPUP2～CPUP0 ビットに反映します。

CPU の優先レベルの制御を表 5.7 に示します。

表 5.7 CPU の優先レベルの制御

割り込み制御 モード	割り込み優先 順位	割り込み マスクビット	CPUPCR の IPSETE	制御状態	
				CPUP2~CPUP0 の値	CPUP2~CPUP0 の書き換え
0	デフォルト	l = 任意	0	B'111 ~ B'000	書き換え可
		l = 0	1	B'000	無効
		l = 1		B'100	
2	IPR の設定	l2~l0	0	B'111 ~ B'000	書き換え可
			1	l2 ~ l0	無効

CPU に対する DTC、DMAC の優先レベル制御機能の設定例と、そのときの転送要求制御状態を表 5.8 に示します。DMAC はチャンネルごとに独立して優先レベルを設定できますが、表中では 1 チャンネル分を示しています。DMAC では各チャンネルに異なる優先レベルを設定して独立に転送の制御を行うことが可能です。

表 5.8 CPU に対する DTC、DMAC の優先レベル制御機能の設定例とそのときの制御状態

割り込み制御 モード	CPUPCR の CPUPCE	CPUP2~ CPUP0 の値	DTC2~ DTC0 の値	DMAP2~ DMAP0 の値	転送要求制御状態	
					DTC	DMAC
0	0	任意	任意	任意	許可	許可
		1	B'000	B'000	B'000	許可
	B'100		B'000	B'000	マスク	マスク
	B'100		B'000	B'011	マスク	マスク
	B'100		B'111	B'101	許可	許可
	B'000	B'111	B'101	許可	許可	
2	0	任意	任意	任意	許可	許可
	1	B'000	B'000	B'000	許可	許可
		B'000	B'011	B'101	許可	許可
		B'011	B'011	B'101	許可	許可
		B'100	B'011	B'101	マスク	許可
		B'101	B'011	B'101	マスク	許可
		B'110	B'011	B'101	マスク	マスク
		B'111	B'011	B'101	マスク	マスク
		B'101	B'011	B'101	マスク	許可
	B'101	B'110	B'101	許可	許可	

5.8 使用上の注意事項

5.8.1 割り込みの発生とディスエーブルとの競合

割り込みイネーブルビットをクリアして割り込み要求をマスクする場合、割り込みのマスクはその命令実行終了後に有効になります。BCLR 命令、MOV 命令などで割り込みイネーブルビットをクリアする場合、命令実行中にその割り込みが発生すると、命令実行終了時点では当該割り込みはイネーブル状態にあるため、命令実行終了後にその割り込み例外処理を開始します。割り込みモード 2 の場合、命令実行終了後、EXR の I2~I0 ビットで設定した CPU の割り込みマスクレベル以下の割り込み例外処理を実行する場合があります。ただし、その割り込みより優先順位の高い割り込み要求がある場合には優先順位の高い割り込み例外処理を実行し、その割り込みは無視されます。割り込み要因フラグを 0 にクリアする場合も同様です。割り込みモード 2 の場合、IPR のレジスタ書き換えで、割り込みをディスエーブルにする場合も同様です。TPU の TIER の TCIEV ビットを 0 にクリアする場合の例を図 5.7 に示します。なお、割り込みをマスクした状態でイネーブルビット、または割り込み要因フラグを 0 にクリアまたは、IPR レジスタの書き換えをすれば、上記の競合は発生しません。本競合を回避するための割り込みマスク方法を図 5.8 に示します。

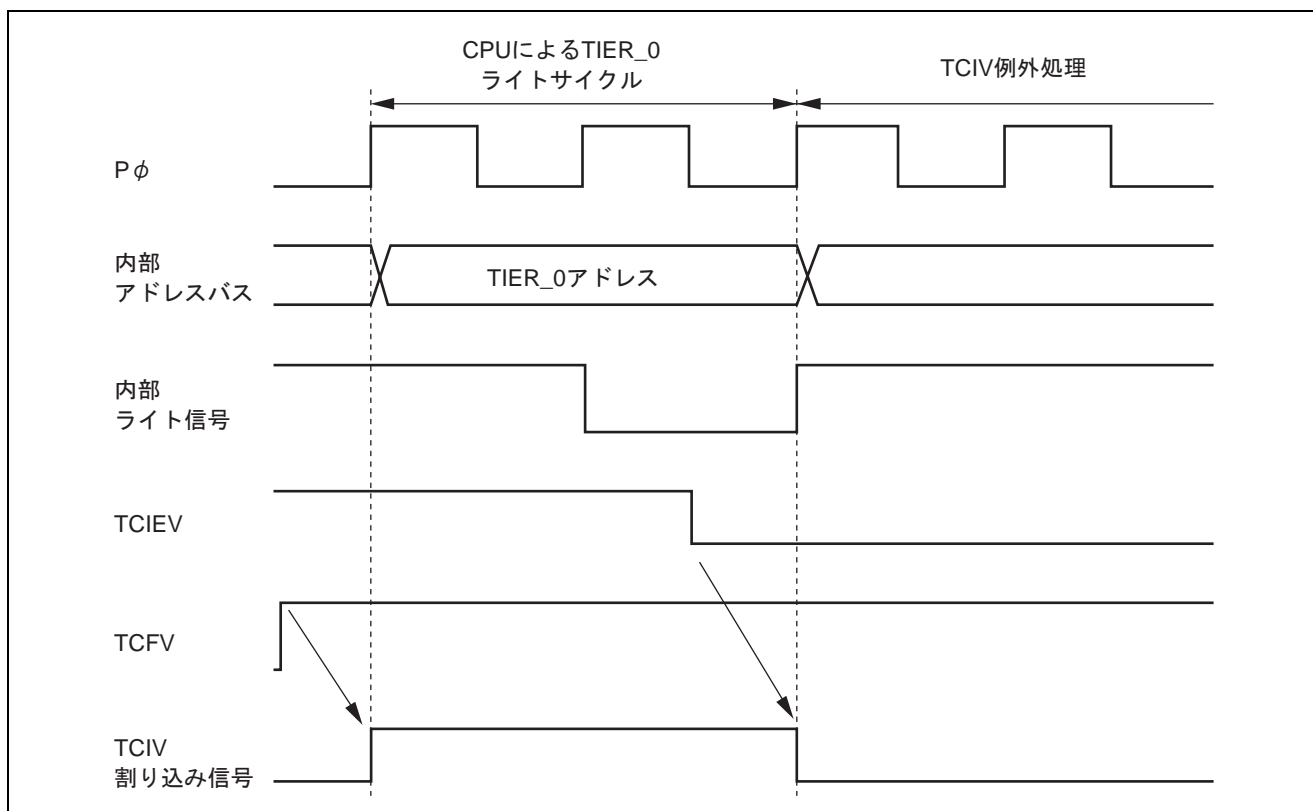


図 5.7 割り込みの発生とディスエーブルの競合

同様に DTC を起動する場合、DTC イネーブルビットの書き換え直前に割り込みが発生すると DTC の起動と CPU の割り込み例外処理を両方実行します。DTC イネーブルビットを変更する場合は、対応する割り込み要求を発生しない状態で行ってください。

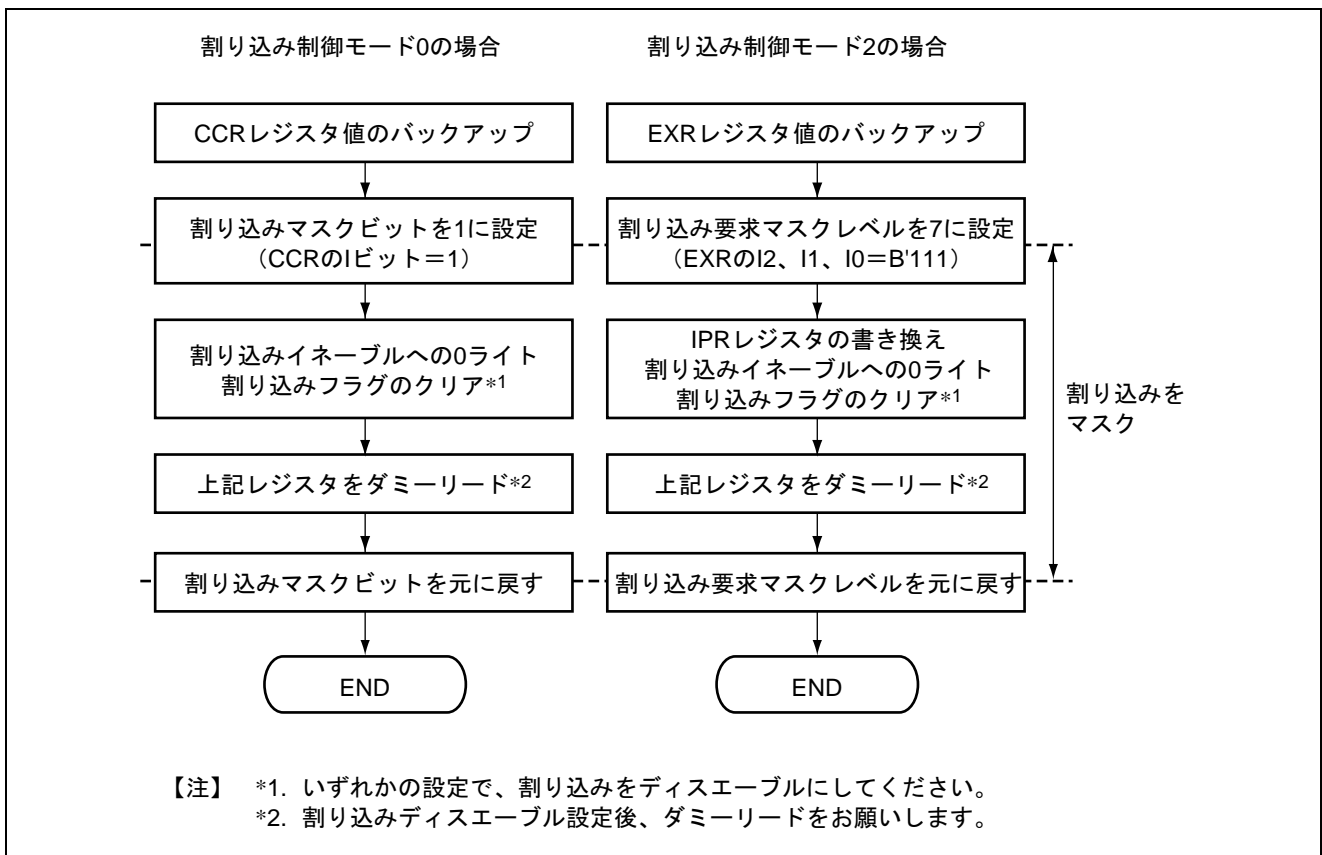


図 5.8 割り込み競合の回避方法

5.8.2 割り込みを禁止している命令

実行直後に割り込み要求を受け付けない命令として、LDC、ANDC、ORC、XORC 命令があります。これらの命令実行終了後は NMI 割り込みを含めて割り込みが禁止され、必ず次の命令を実行します。これらの命令により I ビットを設定した場合、命令実行終了の 2 ステート後に新しい値が有効になります。

5.8.3 割り込み禁止期間

割り込みコントローラには割り込み要求の受け付けを禁止している期間があります。割り込みコントローラは、CPU が LDC、ANDC、ORC、XORC 命令によってマスクレベルを更新した後の 3 ステート期間、および割り込みコントローラのレジスタにライトしている期間は、割り込み要求を受け付けません。

5.8.4 EEPMOV 命令実行中の割り込み

EEPMOV.B 命令と EEPMOV.W 命令では、割り込み動作が異なります。

EEPMOV.B 命令によるデータ転送中に NMI を含めた割り込み要求があっても、転送終了まで割り込みを受け付けません。

EEPMOV.W 命令によるデータ転送中に割り込み要求があった場合、転送サイクルの切れ目で割り込み例外処理が開始されます。このときスタックされる PC の値は、次の命令のアドレスとなります。このため、EEPMOV.W 命令実行中に割り込みが発生する場合には、以下のプログラムとしてください。

```
L1: EEPMOV.W
      MOV.W  R4,R4
      BNE   L1
```

5.8.5 MOVMD、MOVSD 命令実行中の割り込み

MOVMD 命令または MOVSD 命令によるデータ転送中に割り込み要求があった場合、転送サイクルの切れ目で割り込み例外処理が開始されます。このときスタックされる PC の値は、MOVMD 命令または MOVSD 命令のアドレスとなります。割り込み処理ルーチンからの復帰後に、残りのデータ転送を継続します。

5.8.6 周辺モジュール割り込み要因フラグ

システムクロックに対して周辺モジュールのクロックの分周によらず、CPU によって周辺モジュールの割り込み要因フラグ、割り込みイネーブルビットをクリアするときは、周辺モジュールと同期を取るために、割り込み処理ルーチン内でフラグおよび、割り込みイネーブルレジスタをクリアした後にはかならず当該レジスタをリードしてください。

6. バスコントローラ (BSC)

本 LSI はバスコントローラ (BSC) を内蔵しています。バスコントローラはバス権調停機能をもっており、内部バスマスタである CPU、DMAC、および DTC の動作を制御します。

6.1 特長

- ライトデータバッファ機能
内蔵周辺モジュールへのライトアクセスと内蔵メモリアccessを並列に実行可能
- バス権調停機能 (バスアービトレーション)
バスアービタを内蔵し、CPU、DMAC、DTC のバス権要求を調停
- マルチクロック機能
内部周辺機能は周辺モジュールクロックに同期して動作可能

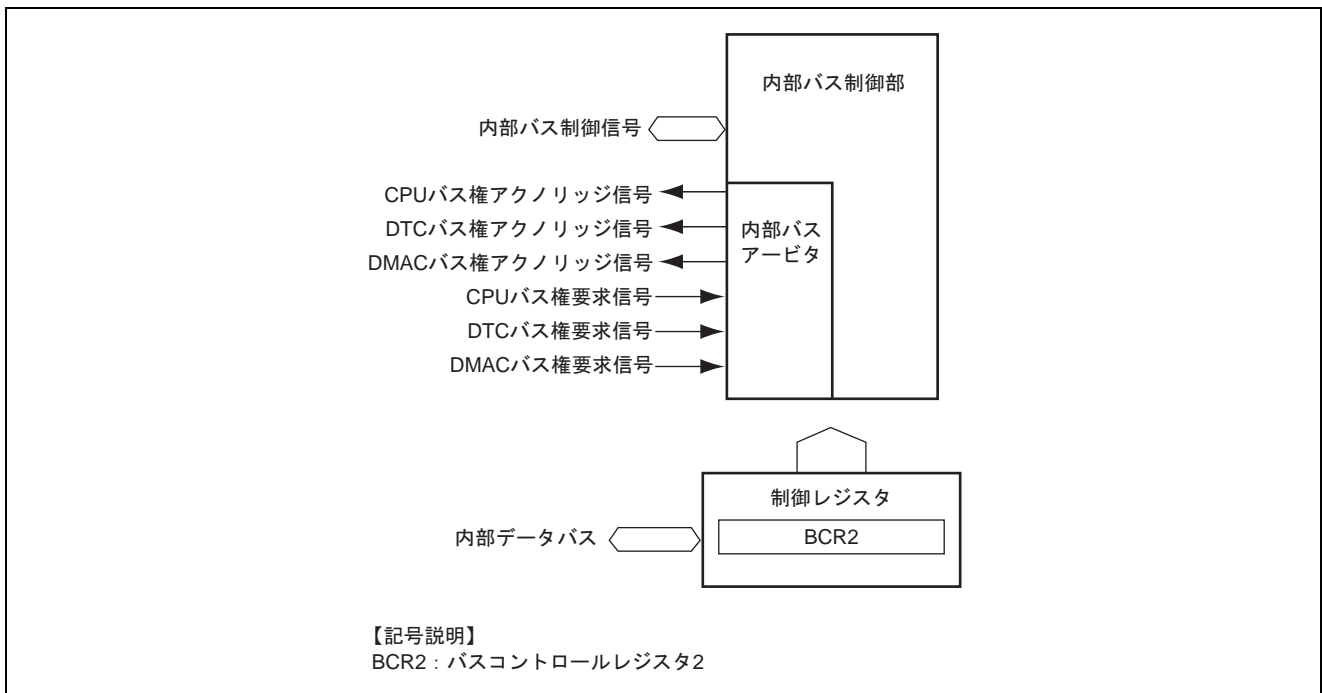


図 6.1 バスコントローラのブロック図

6.2 レジスタの説明

バスコントローラには以下のレジスタがあります。

- バスコントロールレジスタ2 (BCR2)

6.2.1 バスコントロールレジスタ 2 (BCR2)

BCR2 は、CPU、DMAC、および DTC のバスアービトレーション制御、周辺モジュールへのライトデータバッファ機能の許可/禁止の設定を行います。

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

ビット	ビット名	初期値	R/W	説明
7	—	0	R	リザーブビット
6	—	0	R	初期値を変更しないでください。
5	—	0	R/W	リザーブビット リードすると常に0が読み出されます。ライトする値は常に0にしてください。
4	IBCCS	0	R/W	内部バスサイクルコントロールセレクト 内部バスアービタの機能を選択します。 0 : 優先順位に従ってバス権を解放 1 : DMAC、または DTC のバス権要求と CPU のバス権要求が競合したとき、交互にバスサイクルを実行します。
3	—	0	R	リザーブビット
2	—	0	R	初期値を変更しないでください。
1	—	1	R	リザーブビット 初期値を変更しないでください。
0	PWDBE	0	R/W	周辺モジュールライトデータバッファイネーブル 周辺モジュールへのライトサイクルのとき、ライトデータバッファ機能を使用できます。 0 : ライトデータバッファ機能を使用しない 1 : ライトデータバッファ機能を使用する

6.3 バス構成

本 LSI の内部バス構成を図 6.2 に示します。バス構成は、以下に示す 2 種類で構成されています。

1. 内部バス：CPU、DTC、DMACと、内蔵ROM、内蔵RAM、内部周辺バスを接続するバス
2. 周辺バス：DMAC、バスコントローラ、割り込みコントローラなどのレジスタと、SCIやタイマなどの周辺モジュールのレジスタをアクセスするバス

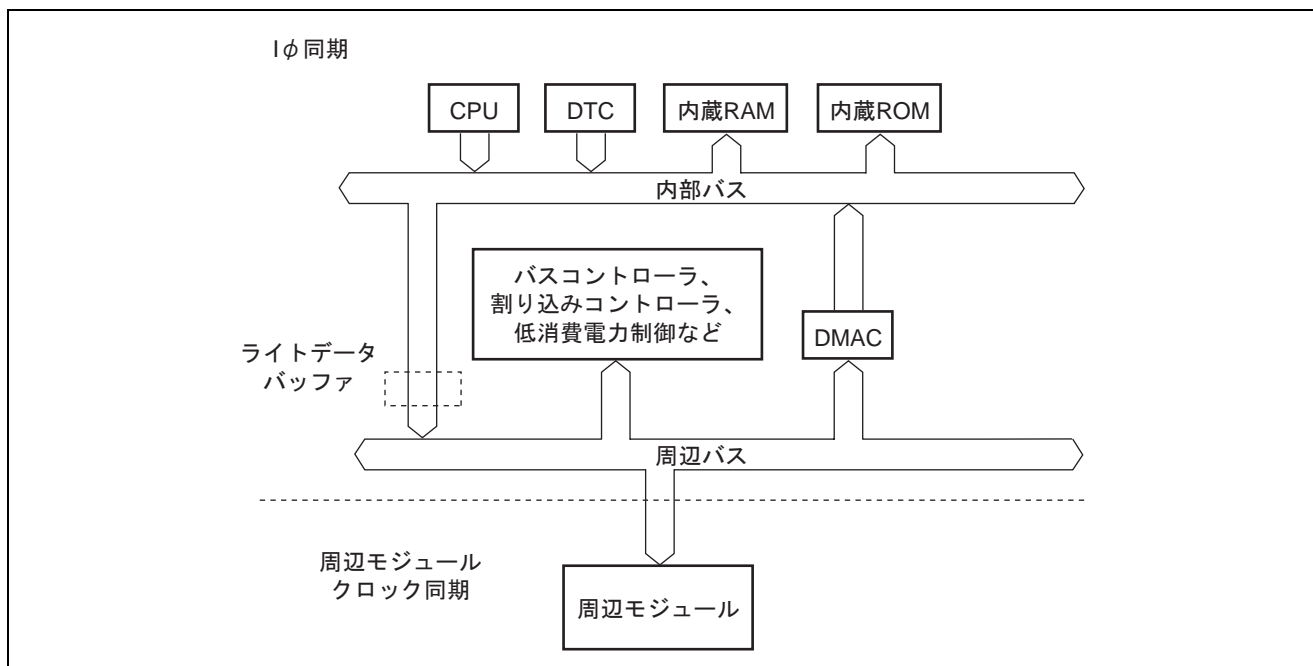


図 6.2 内部バス構成

6.4 マルチクロック機能とアクセスステート数

本 LSI の内蔵機能は、システムクロック (I ϕ)、周辺モジュールクロック (P ϕ 、F ϕ 、A ϕ 、R ϕ) のいずれかのクロックに同期して動作します。表 6.1 にそれぞれの同期クロックと対応する機能を示します。

表 6.1 同期クロックと対応する機能

同期クロック	機能名
I ϕ	MCU 動作モード 割り込みコントローラ バスコントローラ CPU DMAC DTC 内蔵メモリ クロック発振器 低消費電力制御
P ϕ	I/O ポート TPU PPG WDT SCI RCAN-TL1 LIN CRC RCU
F ϕ	FLASH/EEPROM
A ϕ	A/D
R ϕ	RSPI

各同期クロック (I ϕ 、P ϕ 、F ϕ 、A ϕ 、R ϕ) の周波数は、システムクロックコントロールレジスタ 0、1 (SCKCR0、SCKCR1) の設定により独立に制御することが可能です。詳細は「第 22 章 クロック発振器」を参照してください。

6.5 内部バス

6.5.1 内部アドレス空間へのアクセス

本 LSI の内部アドレス空間には、内蔵 ROM 空間、内蔵 RAM 空間、内蔵周辺モジュールレジスタ空間があり、それぞれアクセスに要するサイクル数が異なります。

表 6.2 に内蔵メモリ空間のアクセスサイクル数を示します。

表 6.2 内蔵メモリ空間のアクセスサイクル数

アクセス対象	アクセス	サイクル数
内蔵 ROM 空間	リード	11φサイクル
	ライト	21φサイクル
内蔵 RAM 空間	リード	11φサイクル
	ライト	21φ (11φサイクル)

内蔵周辺モジュールレジスタへのアクセスは、レジスタによってアクセスサイクル数が異なります。バスマスタの動作クロックと周辺モジュールの動作クロックが 1 : n に分周されている場合、外部バスクロックの分周と同様にレジスタアクセスに 0~n-1 の分周クロック同期化サイクルが挿入されます。

表 6.3 に内蔵周辺モジュールレジスタ空間のアクセスサイクル数を示します。

表 6.3 内蔵周辺モジュールレジスタ空間のアクセスサイクル数

アクセス対象	サイクル数		ライトデータバッファ機能
	リード	ライト	
DMAC のレジスタ	21φ		無効
MCU 動作モード、クロック発振器、消費電力制御のレジスタ、割り込みコントローラ、バスコントローラ、DTC のレジスタ	21φ	31φ	無効
I/O ポートの PFCR、WDT のレジスタ	2Pφ	3Pφ	無効
RCAN-TL1	2Pφ	3Pφ	有効
I/O ポートの PFCR 以外のレジスタ、TPU、PPG、SCI、LIN、CRC、RCU のレジスタ	2Pφ		有効
FLASH/EEPROM	2Fφ	3Fφ	有効
A/D	2Aφ		有効
RSPI	2Rφ		有効

6.6 ライトデータバッファ機能

本 LSI は、周辺モジュールアクセスのためにライトデータバッファ機能を備えています。ライトデータバッファ機能を使用すると、周辺モジュールライトサイクルと、内蔵メモリアクセスを並行して実行することができます。BCR2 の PWDBE ビットを 1 にセットすると、ライトデータバッファ機能を使用することができます。

図 6.3 にライトデータバッファ機能を使用したときのタイミング例を示します。この機能を使用したとき、内部 I/O レジスタライトが 2 ステート以上続き、次に内蔵 RAM、内蔵 ROM アクセスがある場合は最初の 2 ステートは内部 I/O レジスタライトのみが実行されますが、次のステートから内部 I/O レジスタライトの終了を待たずに内蔵メモリアクセスが並行して実行されます。

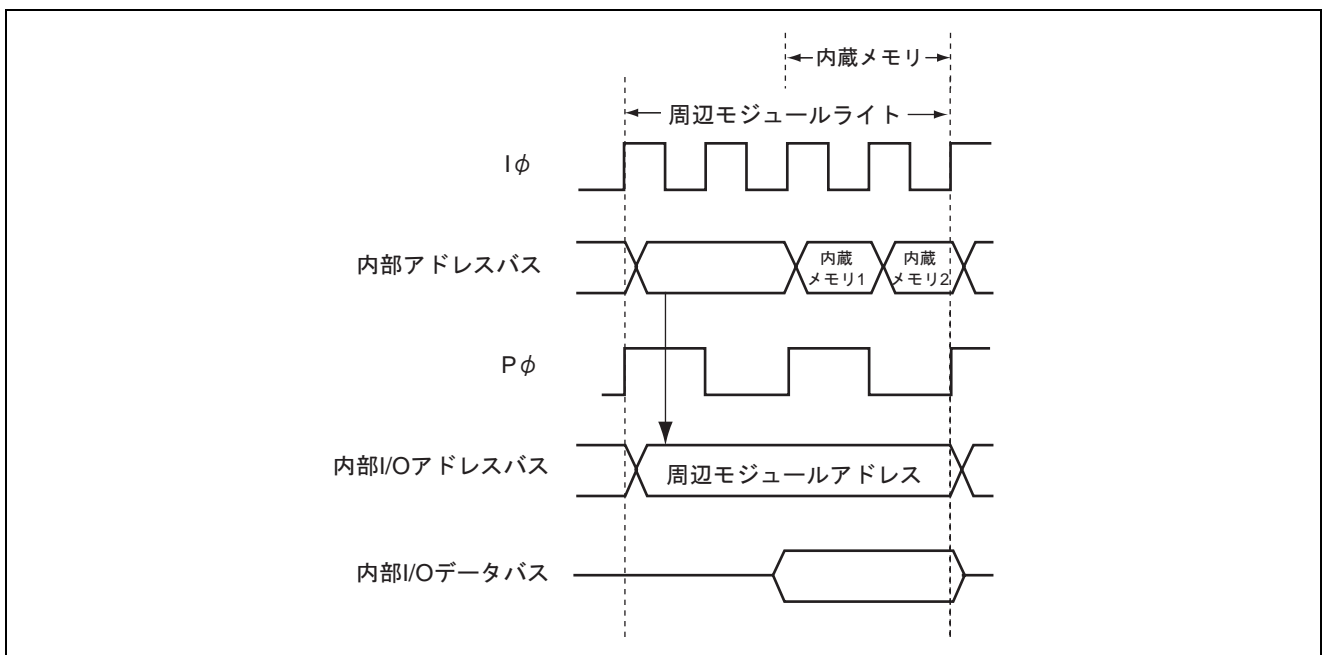


図 6.3 周辺モジュールライトデータバッファ機能使用時のタイミング例

6.7 バスアービトレーション

本 LSI は、バスマスタの動作を調停 (バスアービトレーション) するバスアービタを内蔵しています。内部バスアービタの対象は、CPU、DTC、DMAC のそれぞれのアクセスです。

バスアービタは所定のタイミングで優先順位を判定し、バス権要求アクノリッジ信号によりバスの使用を許可します。

6.7.1 動作説明

バスアービタは、バスマスタのバス権要求信号を検出して、バス権が要求されていれば、そのバスマスタにバス権要求アクノリッジ信号を与えます。複数のバスマスタからバス権要求があれば、最も優先順位の高いものにバス権要求アクノリッジ信号を与えます。バス権要求アクノリッジ信号を受け取ったバスマスタは、以後この信号が取り消されるまでバスを占有します。

バスマスタの優先順位は以下のとおりです。

内部バスアービトレーション:

(高) DMAC > DTC > CPU (低)

ただし、BCR2 の IBCCS ビットを 1 にセットすることにより、DMAC または DTC のアクセスが連続する場合に CPU の優先順位を高くして、DMAC または DTC と交互にバス権を取ることができます。このとき、DMAC と DTC 間の優先順位に変化はありません。

6.7.2 バス権移行タイミング

バス権を獲得して動作しているバスマスタよりも優先順位の高いバスマスタからのバス権要求があったときでも、すぐにバス権が移行するとは限りません。各バスマスタには、バス権を譲ることができるタイミングがあります。

(1) CPU

CPU は最も優先順位が低いバスマスタであり、DTC、DMAC からのバス権要求があると、内部バスアービタはバス権を CPU からバス権要求のあったバスマスタに移行します。

バス権が移行するタイミングはバスサイクルの切れ目です。スリープモード中はクロックに同期してバス権を移行します。

ただし、以下の場合には、バス権の移行を禁止しています。

- ワード、ロングワードサイズのアクセスを分割して実行しているとき
- スタック操作を複数バスサイクルで実行しているとき
- メモリ間転送命令、ブロック転送命令、およびTAS命令で、転送データのリードとライトの期間
(ブロック転送命令は、ライトサイクルと次の転送データのリードサイクルの間は、バス権を移行することができます。)
- ビット操作命令やメモリ演算命令で、対象のデータをリードしてからライトするまでの期間
(命令の条件によりライトを行わない場合も、ライトに相当するステートまでの期間が該当します。)

(2) DTC

DTC は起動要求が発生すると内部バスアービタに対してバス権を要求します。

DTC は、いったんバス権を取得すると一連の転送処理サイクルを連続して実行します。この期間に DTC より優先順位の高いバスマスタからのバス権要求があるとバス権を移行します。BCR2 の IBCCS ビットが 1 にセットされているときは、バス権を CPU に移行します。

ただし、以下の場合は、バス権の移行を禁止しています。

- 転送情報のリード中
- 1回のデータを転送中
- 転送情報のライトバック中

起動要求に対応する一連の転送処理サイクルが終了すると次のサイクルはバス権を解放します。

(3) DMAC

DMAC は起動要求が発生すると内部バスアービタに対してバス権を要求します。

DMAC の転送サイクルには、いったんバス権を取得すると連続して転送を行う場合と、転送サイクルごとにバス権を解放する場合があります。

バス権を解放せずに連続して実行する転送サイクルには、以下の場合があります。

- デュアルアドレスモードのリードサイクルから、リードサイクルに対応するライトサイクルの期間
DMAC より優先順位の高いバスマスタからのバス権要求がなく、BCR2 の IBCCS ビットが 0 にクリアされているとき、バス権を解放せずに連続して実行する転送サイクルには、以下の場合があります。

- ブロック転送モードの1ブロック転送中
- バーストモードの転送中

上記以外の転送サイクルの場合、バスサイクルが終了した時点でバス権を移行します。

6.8 リセットとバスコントローラ

リセットで、バスコントローラを含めて本 LSI はその時点でリセット状態になります。実行中のバスサイクルは途中で打ち切られます。

6.9 使用上の注意事項

(1) 全モジュールクロックストップモード

本 LSI は、MSTPCRA レジスタの ACSE ビットを 1 にセットし、すべての周辺モジュールのクロックを停止 (MSTPCRA、MSTPCRB、MSTPCRC、MSTPCRD、MSTPCRE=H'FFFFFFF) すると、バスコントローラもクロックを停止する全モジュールクロックストップモードへ遷移します。詳細は「第 23 章 低消費電力」を参照してください。

7. DMA コントローラ (DMAC)

本 LSI は、4 チャンネルの DMA コントローラ (DMAC) を内蔵しています。

7.1 特長

- 最大4Gバイトのアドレス空間をアクセス可能
 - データ転送単位はバイト、ワード、またはロングワードを選択可能
 - 総転送サイズは最大4Gバイト (4,294,967,295バイト) 設定可能
総転送サイズを設定しないフリーランニングモードも選択可能
 - DMACの起動方法は、オートリクエスト、内蔵モジュール割り込み、外部リクエストを選択可能
オートリクエスト：CPUより起動 (サイクルスチールモードとバーストモードの選択が可能)
内蔵モジュール割り込み：各種内蔵周辺モジュールからの割り込み要求を起動要因として選択可能
外部リクエスト： $\overline{\text{DREQ}}$ 信号のLowレベル検出、または立ち下がりエッジ検出を選択可能
(4チャンネルすべて外部リクエストを受け付け可能)
 - アドレスモードはデュアルアドレスモード、シングルアドレスモードを選択可能
デュアルアドレスモード：転送元、転送先双方をアドレス指定しデータを転送
シングルアドレスモード：転送元、または転送先の周辺デバイスを $\overline{\text{DACK}}$ 信号でアクセスし、
もう一方をアドレス指定してデータを転送
 - 転送モードはノーマル転送モード、リピート転送モード、ブロック転送モードを選択可能
ノーマル転送モード：1バイト/1ワード/1ロングワードを1回の転送要求で転送
リピート転送モード：1バイト/1ワード/1ロングワードを1回の転送要求で転送
リピートサイズ分、データを転送すると転送開始時のアドレスに復帰。
リピートサイズは最大64k回 (65,536バイト/ワード/ロングワード) 設定可能
ブロック転送モード：1ブロックのデータを1回の転送要求で転送
ブロックサイズは最大64k回 (65,536バイト/ワード/ロングワード) 設定可能
 - 拡張リピートエリア機能 (転送アドレスレジスタの上位ビットの値を固定して特定範囲のアドレス値を繰り返させ、リングバッファなどのデータ転送を行う) を選択可能
拡張リピートエリアには、1ビット (2バイト) ~27ビット (128Mバイト) を転送元、転送先別に設定可能
 - アドレス更新は、1、2または4の増減、固定、オフセット加算を選択可能
オフセット加算でアドレスを更新させると、途中のアドレスを飛ばしたデータ転送可能
- 【注】 H8SX/1720S グループでは、外部リクエスト、シングルアドレスモードは使用できません。

- ワードデータや、ロングワードデータを各データ境界から外れたアドレスに転送可能
データ転送時にアドレスに応じた最適なデータ (バイトやワード) に分割して転送可能
- CPUに2種類の割り込み要求を発生
転送終了割り込み：転送カウンタで設定したデータ数を転送終了後に発生
転送エスケープ終了割り込み：残る総転送サイズが1回の転送要求で転送する設定サイズを下回ったとき、リピートサイズ分のデータ転送を終了したとき、または拡張リピートエリアオーバーフローしたときに発生。

DMAC のブロックを図 7.1 に示します。

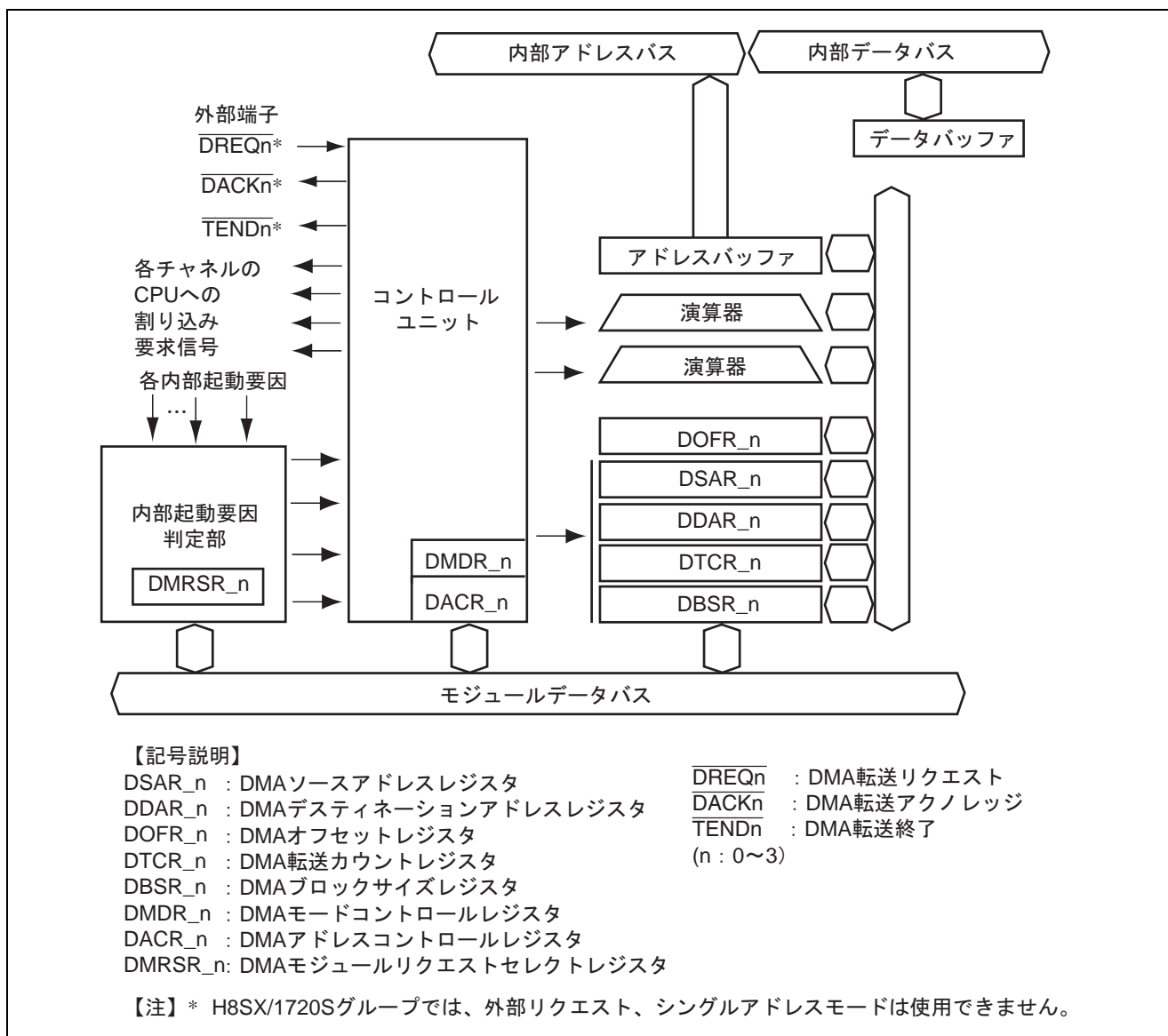


図 7.1 DMAC のブロック図

7.2 レジスタの説明

DMAC には以下のレジスタがあります。

【チャンネル0】

- DMAソースアドレスレジスタ_0 (DSAR_0)
- DMAデスティネーションアドレスレジスタ_0 (DDAR_0)
- DMAオフセットレジスタ_0 (DOFR_0)
- DMA転送カウントレジスタ_0 (DTCR_0)
- DMAブロックサイズレジスタ_0 (DBSR_0)
- DMAモードコントロールレジスタ_0 (DMDR_0)
- DMAアドレスコントロールレジスタ_0 (DACR_0)
- DMAモジュールリクエストセレクトレジスタ_0 (DMRSR_0)

【チャンネル1】

- DMAソースアドレスレジスタ_1 (DSAR_1)
- DMAデスティネーションアドレスレジスタ_1 (DDAR_1)
- DMAオフセットレジスタ_1 (DOFR_1)
- DMA転送カウントレジスタ_1 (DTCR_1)
- DMAブロックサイズレジスタ_1 (DBSR_1)
- DMAモードコントロールレジスタ_1 (DMDR_1)
- DMAアドレスコントロールレジスタ_1 (DACR_1)
- DMAモジュールリクエストセレクトレジスタ_1 (DMRSR_1)

【チャンネル2】

- DMAソースアドレスレジスタ_2 (DSAR_2)
- DMAデスティネーションアドレスレジスタ_2 (DDAR_2)
- DMAオフセットレジスタ_2 (DOFR_2)
- DMA転送カウントレジスタ_2 (DTCR_2)
- DMAブロックサイズレジスタ_2 (DBSR_2)
- DMAモードコントロールレジスタ_2 (DMDR_2)
- DMAアドレスコントロールレジスタ_2 (DACR_2)
- DMAモジュールリクエストセレクトレジスタ_2 (DMRSR_2)

【チャンネル 3】

- DMAソースアドレスレジスタ_3 (DSAR_3)
- DMAデスティネーションアドレスレジスタ_3 (DDAR_3)
- DMAオフセットレジスタ_3 (DOFR_3)
- DMA転送カウントレジスタ_3 (DTCR_3)
- DMAブロックサイズレジスタ_3 (DBSR_3)
- DMAモードコントロールレジスタ_3 (DMDR_3)
- DMAアドレスコントロールレジスタ_3 (DACR_3)
- DMAモジュールリクエストセレクトレジスタ_3 (DMRSR_3)

7.2.1 DMA ソースアドレスレジスタ (DSAR)

DSAR は、転送元のアドレスを指定するための 32 ビットのリード/ライト可能なレジスタです。アドレス更新機能を持ち、転送処理が行われるたびに次の転送元アドレスに更新されます。シングルアドレスモードにおいて、DDAR をデスティネーションアドレスとして転送すると (DACR の DIRS=1)、DSAR の値は無視されます。

DSAR は CPU から常にリード可能ですが、転送処理中のチャンネルの DSAR をリードするときは、ロングワードサイズでリードしてください。DMA 動作中のチャンネルの DSAR にはライトしないでください。

ビット	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ビット名																
初期値:	0	0	0	0	0	0	0	0	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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ビット	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ビット名																
初期値:	0	0	0	0	0	0	0	0	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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7.2.2 DMA デスティネーションアドレスレジスタ (DDAR)

DDAR は、転送先のアドレスを指定するための 32 ビットのリード/ライト可能なレジスタです。アドレス更新機能を持ち、転送処理が行われるたびに次の転送先アドレスに更新されます。シングルアドレスモードにおいて、DSAR をソースアドレスとして転送すると (DACR の DIRS=0)、DDAR の値は無視されます。

DDAR は CPU から常にリード可能ですが、転送処理中のチャンネルの DDAR をリードするときは、ロングワードサイズでリードしてください。DMA 動作中のチャンネルの DDAR にはライトしないでください。

ビット	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ビット名																
初期値:	0	0	0	0	0	0	0	0	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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ビット	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ビット名																
初期値:	0	0	0	0	0	0	0	0	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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7.2.3 DMA オフセットレジスタ (DOFR)

DOFR は、ソースアドレス、デスティネーションアドレスの更新に、オフセット加算を選択したときのオフセット値を設定する 32 ビットのリード/ライト可能なレジスタです。各チャンネル毎に独立して設定できますが、同一チャンネル内ではソース側、デスティネーション側ともに同じオフセット値を使用します。

ビット	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ビット名																
初期値:	0	0	0	0	0	0	0	0	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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ビット	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ビット名																
初期値:	0	0	0	0	0	0	0	0	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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7.2.4 DMA 転送カウントレジスタ (DTCR)

DTCR は、転送するデータのサイズ (総転送サイズ) を設定するための 32 ビットのリード/ライト可能なレジスタです。

DTCR を H'00000001 に設定すると、総転送サイズは 1 バイトになります。H'00000000 に設定すると、「総転送サイズ=指定なし」となり、転送カウンタは停止して転送を行います (フリーランニングモード)。このとき、転送カウンタによる転送終了割り込みは発生しません。H'FFFFFFFF に設定すると、総転送サイズは最大値 4G バイト (4,294,967,295 バイト) になります。DMA 動作中は残りの転送サイズを示します。1 データ転送毎に、転送したデータアクセスサイズに応じた値がデクリメントされます (バイト: -1、ワード: -2、ロングワード: -4)。

DTCR は CPU から常にリード可能ですが、転送処理中のチャンネルの DTCR をリードするときは、ロングワードサイズでリードしてください。DMA 動作中のチャンネルの DTCR にはライトしないでください。

ビット	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ビット名																
初期値:	0	0	0	0	0	0	0	0	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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ビット	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ビット名																
初期値:	0	0	0	0	0	0	0	0	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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7.2.5 DMA ブロックサイズレジスタ (DBSR)

DBSR は、リピートサイズ、ブロックサイズを設定します。DBSR は、リピート転送モード、ブロック転送モードのときに有効となり、ノーマル転送モードのときには無効です。

ビット	31	...	16
ビット名	BKSZH31	...	BKSZH16
初期値 :	0	...	0
R/W :	R/W	...	R/W
ビット	15	...	0
ビット名	BKSZ15	...	BKSZ0
初期値 :	0	...	0
R/W :	R/W	...	R/W

ビット	ビット名	初期値	R/W	説明
31~16	BKSZH31 ~ BKSZH16	不定	R/W	リピートサイズ、ブロックサイズを設定します。 これらのビットを H'0001 に設定すると 1 バイト、1 ワード、または 1 ロングワードになります。H'0000 に設定すると最大値になります (表 7.1 参照)。DMA 動作中は常に固定です。
15~0	BKSZ15 ~BKSZ0	不定	R/W	DMA 動作中は、リピートサイズ、ブロックサイズの残りサイズを示します。1 データ転送毎に-1 されます。残りサイズが 0 になると BKSZH の値がロードされず。ライトするときは、BKSZH と同じ値を設定してください。

表 7.1 データアクセスサイズと有効ビット、設定可能サイズの対応

モード	データアクセスサイズ	BKSZH 有効ビット	BKSZ 有効ビット	設定可能サイズ (バイト)
リピート転送モード	バイト	31~16	15~0	1~65,536
ブロック転送モード	ワード			2~131,072
	ロングワード			4~262,144

7.2.6 DMA モードコントロールレジスタ (DMDR)

DMDR は、DMAC の動作を制御します。

• DMDR0

ビット	31	30	29	28	27	26	25	24
ビット名	DTE	DACKE	TENDE	—	DREQS	NRD	—	—
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	R
ビット	23	22	21	20	19	18	17	16
ビット名	ACT	—	—	—	ERRF	—	ESIF	DTIF
初期値:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/(W)*	R	R/(W)*	R/(W)*
ビット	15	14	13	12	11	10	9	8
ビット名	DTSZ1	DTSZ0	MDS1	MDS0	TSEIE	—	ESIE	DTIE
初期値:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W
ビット	7	6	5	4	3	2	1	0
ビット名	DTF1	DTF0	DTA	—	—	DMAP2	DMAP1	DMAP0
初期値:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R	R/W	R/W	R/W

【注】 * フラグをクリアするための1リード後の0ライトのみ可能です。

• DMDR_1~DMDR_3

ビット	31	30	29	28	27	26	25	24
ビット名	DTE	DACKE	TENDE	—	DREQS	NRD	—	—
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	R
ビット	23	22	21	20	19	18	17	16
ビット名	ACT	—	—	—	—	—	ESIF	DTIF
初期値:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/(W)*	R/(W)*
ビット	15	14	13	12	11	10	9	8
ビット名	DTSZ1	DTSZ0	MDS1	MDS0	TSEIE	—	ESIE	DTIE
初期値:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W
ビット	7	6	5	4	3	2	1	0
ビット名	DTF1	DTF0	DTA	—	—	DMAP2	DMAP1	DMAP0
初期値:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R	R/W	R/W	R/W

【注】 * フラグをクリアするための1リード後の0ライトのみ可能です。

ビット	ビット名	初期値	R/W	説明
31	DTE	0	R/W	<p>データトランスファイナブル</p> <p>対応するチャンネルのデータ転送を許可または禁止します。このビットが1にセットされていると、DMA 動作中であることを示します。</p> <p>オートリクエストのときに、このビットを1にセットすると転送処理に入ります。内蔵モジュール割り込み、外部リクエストでは、このビットに1をセットした後に転送要求が発生すると転送処理に入ります。DMA 動作中にこのビットを0にクリアすると転送を停止します。</p> <p>ブロック転送モードでは、DMA 動作中にこのビットを0にクリアすると、処理中の1ブロックの転送終了後にこのビットが0にクリアされます。</p> <p>転送を終了(中断)させる外的要因が発生した場合は、自動的にこのビットが0にクリアされ転送を終了させます。</p> <p>このビットを1にセットした状態で、動作モード、転送方法などを変更しないでください。</p> <p>0: データ転送を禁止 1: データ転送を許可 (DMA 動作中)</p> <p>[クリア条件]</p> <ul style="list-style-type: none"> 設定の総転送サイズ数の転送を終了したとき リピートサイズ終了割り込みにより停止したとき 拡張リピートエリアオーバフロー割り込みにより停止したとき 転送サイズエラー割り込みにより停止したとき 0をライトして転送を終了したとき <p>ただし、ブロック転送モードでは1ブロック転送終了後に反映</p> <ul style="list-style-type: none"> アドレスエラー、NMI 割り込みが発生したとき リセット、ハードウェアスタンバイモード*時 <p>【注】* 本 LSI には、ハードウェアスタンバイモードはありません。</p>
30	DACK	0	R/W	<p>DACK 端子出カインープル</p> <p>シングルアドレスモードのとき、$\overline{\text{DACK}}$ 端子の出力を許可または禁止します。デュアルアドレスモードのときは、このビットは無視されます。</p> <p>0: $\overline{\text{DACK}}$ 端子の出力を禁止 1: $\overline{\text{DACK}}$ 端子の出力を許可</p>
29	TENDE	0	R/W	<p>$\overline{\text{TEND}}$ 端子出カインープル</p> <p>$\overline{\text{TEND}}$ 端子の出力を許可または禁止します。</p> <p>0: $\overline{\text{TEND}}$ 端子の出力を禁止 1: $\overline{\text{TEND}}$ 端子の出力を許可</p>
28	—	0	R/W	<p>リザーブビット</p> <p>初期値を変更しないでください。</p>

ビット	ビット名	初期値	R/W	説明
27	DREQS	0	R/W	<p>$\overline{\text{DREQ}}$ セレクト</p> <p>外部リクエストモードで使用する $\overline{\text{DREQ}}$ 端子のサンプリング方法を、Low レベル検出にするか、立ち下がリエッジ検出にするかを選択します。</p> <p>外部リクエストモードでブロック転送を行う場合は、このビットを 0 に設定してください。</p> <p>0 : Low レベル検出 1 : 立ち下がリエッジ検出</p> <p>(転送許可後の最初の転送は、Low レベルで検出します。)</p>
26	NRD	0	R/W	<p>ネクストリクエストディレイ</p> <p>次のリクエストの受付タイミングを選択します。</p> <p>0 : 転送中のバスサイクル終了後に、次の転送要求受付を開始 1 : 転送中のバスサイクル終了時点から Bφ で 1 サイクル後に、次の転送要求の受付を開始</p>
25	—	0	R	リザーブビット
24	—	0	R	リードすると常に 0 が読み出されます。ライトは無効です。
23	ACT	0	R	<p>アクティブステート</p> <p>当該チャネルの動作状態を示すビットです。</p> <p>0 : 転送要求待ち状態、または DTE=0 による転送禁止状態 1 : アクティブ状態</p>
22~20	—	すべて 0	R	<p>リザーブビット</p> <p>リードすると常に 0 が読み出されます。ライトは無効です。</p>
19	ERRF	0	R/(W)*	<p>システムエラーフラグ</p> <p>アドレスエラー、または NMI 割り込みが発生したことを示すフラグです。このビットは DMDR_0 でのみ有効なビットです。このビットが 1 にセットされていると、全チャネルの DTE ビットへのライトが禁止されます。DMDR_1~DMDR_3 では、リザーブビットとなっています。リードすると常に 0 が読み出されます。ライトは無効です。</p> <p>0 : アドレスエラー、または NMI 割り込みなし 1 : アドレスエラー、または NMI 割り込み発生</p> <p>[クリア条件]</p> <ul style="list-style-type: none"> ERRF=1 の状態をリードした後、0 をライトしたとき <p>[セット条件]</p> <ul style="list-style-type: none"> アドレスエラー、または NMI 割り込みが発生したとき <p>ただし、DMAC がモジュールストップ状態にある場合は、アドレスエラー、または NMI 割り込みが発生しても 1 にセットされません。</p>
18	—	0	R	<p>リザーブビット</p> <p>リードすると常に 0 が読み出されます。ライトは無効です。</p>

ビット	ビット名	初期値	R/W	説明
17	ESIF	0	R/(W)*	<p>転送エスケープインタラプトフラグ</p> <p>転送カウンタが0になる前に転送エスケープ終了割り込み要求が発生し、転送エスケープ終了したことを示すフラグです。</p> <p>0：転送エスケープ終了割り込み要求なし</p> <p>1：転送エスケープ終了割り込み要求発生</p> <p>[クリア条件]</p> <ul style="list-style-type: none"> • DTE ビットに1をライトしたとき • ESIF=1の状態をリードした後、0をライトしたとき <p>[セット条件]</p> <ul style="list-style-type: none"> • 転送サイズエラー割り込み要求が発生したとき • リピートサイズ終了割り込み要求が発生したとき • 拡張リピートエリアオーバフロー終了割り込み要求が発生したとき
16	DTIF	0	R/(W)*	<p>データトランスファインタラプトフラグ</p> <p>転送カウンタによる転送終了割り込み要求が発生したことを示すフラグです。</p> <p>0：転送カウンタによる転送終了割り込み要求なし</p> <p>1：転送カウンタによる転送終了割り込み要求発生</p> <p>[クリア条件]</p> <ul style="list-style-type: none"> • DTE ビットに1をライトしたとき • DTIF=1の状態をリードした後、0をライトしたとき <p>[セット条件]</p> <ul style="list-style-type: none"> • DTCR が0になり転送が終了したとき
15 14	DTSZ1 DTSZ0	0 0	R/W R/W	<p>データアクセスサイズ1、0</p> <p>転送するデータアクセスサイズを選択します。</p> <p>00：バイトサイズ (8ビット)</p> <p>01：ワードサイズ (16ビット)</p> <p>10：ロングワードサイズ (32ビット)</p> <p>11：設定禁止</p>
13 12	MDS1 MDS0	0 0	R/W R/W	<p>転送モードセレクト1、0</p> <p>転送モードを選択します。</p> <p>00：ノーマル転送モード</p> <p>01：ブロック転送モード</p> <p>10：リピート転送モード</p> <p>11：(設定禁止)</p>

ビット	ビット名	初期値	R/W	説明
11	TSEIE	0	R/W	<p>転送サイズエラーインタラプトイネーブル</p> <p>転送サイズエラー割り込み要求を許可または禁止します。</p> <p>このビットが1にセットされているときに、DMACの転送により転送カウンタの値が1要求あたりに転送するデータサイズよりも小さい場合、次の転送要求が発生するとDTEビットを0にクリアします。同時にESIFビットが1にセットされ、転送サイズエラー割り込み要求が発生したことを示します。</p> <p>転送サイズエラー割り込み要求の発生要因は次の条件です。</p> <ul style="list-style-type: none"> ・ ノーマル転送モード、リピータ転送モードにおいて、DTCRで設定した総転送サイズがデータアクセスサイズよりも小さいとき ・ ブロック転送モードにおいて、DTCRで設定した総転送サイズがブロックサイズよりも小さいとき <p>0：転送サイズエラー割り込み要求を禁止 1：転送サイズエラー割り込み要求を許可</p>
10	—	0	R	<p>リザーブビット</p> <p>リードすると常に0が読み出されます。ライトは無効です。</p>
9	ESIE	0	R/W	<p>転送エスケープインタラプトイネーブル</p> <p>DMA転送中に発生した転送エスケープ終了割り込み要求を許可または禁止します。このビットを1にセットすると、ESIFビットが1にセットされたとき、CPUまたはDTCに転送エスケープ終了割り込み要求が発生します。転送エスケープ終了割り込み要求は、このビットを0にクリアするか、ESIFビットを0にクリアすると解除されます。</p> <p>0：転送エスケープ割り込み要求を禁止 1：転送エスケープ割り込み要求を許可</p>
8	DTIE	0	R/W	<p>データトランスファインタラプトイネーブル</p> <p>転送カウンタによる転送終了割り込み要求を許可または禁止します。このビットを1にセットすると、DTIFビットが1にセットされたとき、CPUまたはDTCに転送終了割り込み要求が発生します。転送終了割り込み要求は、このビットを0にクリアするか、DTIFビットを0にクリアすると解除されます。</p> <p>0：転送終了割り込み要求を禁止 1：転送終了割り込み要求を許可</p>
7	DTF1	0	R/W	データトランスファファクタ 1、0
6	DTF0	0	R/W	<p>DMACを起動する要因を選択します。内蔵モジュール割り込みのときは、DMRSRで割り込み要因を選択します。外部リクエストのときは、DREQSビットでサンプリング方法を選択できます。</p> <p>00：オートリクエスト（サイクルスチール） 01：オートリクエスト（バースト） 10：内蔵モジュール割り込み 11：外部リクエスト</p>

ビット	ビット名	初期値	R/W	説明
5	DTA	0	R/W	<p>データトランスファアクノリッジ</p> <p>内蔵モジュール割り込みによる DMA 転送時に有効となります。</p> <p>DMRSR によって選択されている要因フラグのクリアを許可または禁止します。</p> <p>0 : 内蔵モジュール割り込み要因による DMA 転送時のクリアを禁止 内蔵モジュール割り込み要因は、DMA 転送によりクリアされませんので、CPU または DTC 転送でクリアしてください。</p> <p>1 : 内蔵モジュール割り込み要因による DMA 転送時のクリアを許可 内蔵モジュール割り込み要因は、DMA 転送によりクリアされ、CPU または DTC に割り込みを要求しません。</p>
4	—	0	R	リザーブビット
3	—	0	R	リードすると常に 0 が読み出されます。ライトは無効です。
2	DMA P2	0	R/W	<p>DMA プライオリティレベル 2~0</p> <p>CPU に対する DTC、DMAC の優先レベルを選択します。DMAC の優先レベルより CPU の優先レベルが高いときは、DMAC は転送要因の受け付けをマスクし、CPU の優先レベルが低くなるまで待機します。チャンネル毎に独立に優先レベルを設定可能です。このビットは、CPUPCR の CPUPCE ビットが 1 のとき有効になります。</p> <p>000 : 優先レベル 0 (最低)</p> <p>001 : 優先レベル 1</p> <p>010 : 優先レベル 2</p> <p>011 : 優先レベル 3</p> <p>100 : 優先レベル 4</p> <p>101 : 優先レベル 5</p> <p>110 : 優先レベル 6</p> <p>111 : 優先レベル 7 (最高)</p>
1	DMA P1	0	R/W	
0	DMA P0	0	R/W	

【注】 * フラグをクリアするための 1 リード後の 0 ライトのみ可能です。

7.2.7 DMA アドレスコントロールレジスタ (DACR)

DACR は、動作モード、転送方法などを設定します。

ビット	31	30	29	28	27	26	25	24
ビット名	AMS	DIRS	—	—	—	RPTIE	ARS1	ARS0
初期値:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R/W	R/W	R/W
ビット	23	22	21	20	19	18	17	16
ビット名	—	—	SAT1	SAT0	—	—	DAT1	DAT0
初期値:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R/W	R/W
ビット	15	14	13	12	11	10	9	8
ビット名	SARIE	—	—	SARA4	SARA3	SARA2	SARA1	SARA0
初期値:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R/W	R/W	R/W	R/W	R/W
ビット	7	6	5	4	3	2	1	0
ビット名	DARIE	—	—	DARA4	DARA3	DARA2	DARA1	DARA0
初期値:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R/W	R/W	R/W	R/W	R/W

ビット	ビット名	初期値	R/W	説明
31	AMS	0	R/W	アドレスモードセレクト アドレスモードをデュアルアドレスモードとシングルアドレスモードから選択します。シングルアドレスモードにすると、DMDR の DACKE ビットの設定により \overline{DACK} 端子が有効になります。 0: デュアルアドレスモード 1: シングルアドレスモード
30	DIRS	0	R/W	シングルアドレスディレクションセレクト シングルアドレスモードのときのデータ転送方向を指定します。デュアルアドレスモードのときは、このビットは無視されます。 0: DSAR をソースアドレスとして転送 1: DDAR をデスティネーションアドレスとして転送
29~27	—	すべて 0	R	リザーブビット リードすると常に 0 が読み出されます。ライトは無効です。

ビット	ビット名	初期値	R/W	説明
26	RPTIE	0	R/W	<p>リピートサイズ終了インタラプトイネーブル</p> <p>リピートサイズ終了割り込み要求を許可または禁止します。</p> <p>リピート転送モードにおいて、このビットが1にセットされているときに、1リピートサイズ分の転送終了後に次の転送要因が発生すると、DMDRのDTEビットを0にクリアします。同時にDMDRのESIFビットが1にセットされ、リピートサイズ終了割り込み要求が発生したことを示します。リピートエリアを指定しない(ARS1、ARS0=B'10)ときでも、1リピートサイズ分の転送終了後にリピートサイズ終了割り込み要求を発生させることができます。</p> <p>また、ブロック転送モードで、このビットが1にセットされているときに、1ブロックサイズ分の転送終了後に次の転送要因が発生すると、DMDRのDTEビットを0にクリアします。同時にDMDRのESIFビットは1にセットされ、リピートサイズ終了割り込み要求が発生されたことを示します。</p> <p>0: リピートサイズ終了割り込み要求を禁止 1: リピートサイズ終了割り込み要求を許可</p>
25	ARS1	0	R/W	<p>エリアセレクト1、0</p> <p>ブロック転送モード、リピート転送モードのときのブロックエリア/リピートエリアを選択します。</p> <p>00: ブロックエリア/リピートエリアはソースアドレス側 01: ブロックエリア/リピートエリアはデスティネーションアドレス側 10: ブロックエリア/リピートエリアは指定しない 11: 設定禁止</p>
24	ARS0	0	R/W	
23	—	0	R	リザーブビット
22	—	0	R	リードすると常に0が読み出されます。ライトは無効です。
21	SAT1	0	R/W	<p>ソースアドレス更新モード1、0</p> <p>ソースアドレス(DSAR)の増減を選択します。シングルアドレスモードにおいて、転送元をDSARに指定していない場合は、このビットは無視されます。</p> <p>00: ソースアドレスは固定 01: ソースアドレスはオフセットアドレスを加算 10: ソースアドレスは増加 (データアクセスサイズにより、+1/+2/+4) 11: ソースアドレスは減少 (データアクセスサイズにより、-1/-2/-4)</p>
20	SAT0	0	R/W	
19	—	0	R	リザーブビット
18	—	0	R	リードすると常に0が読み出されます。ライトは無効です。

ビット	ビット名	初期値	R/W	説明
17	DAT1	0	R/W	デスティネーションアドレス更新モード 1、0 デスティネーションアドレス (DDAR) の増減を指定します。シングルアドレスモードにおいて、転送元を DDAR に指定していない場合は、このビットは無視されます。 00: デスティネーションアドレスは固定 01: デスティネーションアドレスはオフセットを加算 10: デスティネーションアドレスは増加 (データアクセスサイズにより、+1/+2/+4) 11: デスティネーションアドレスは減少 (データアクセスサイズにより、-1/-2/-4)
16	DAT0	0	R/W	
15	SARIE	0	R/W	ソースアドレス拡張リピートエリアオーバーフローインタラプトイネーブル ソースアドレス拡張リピートエリアオーバーフロー割り込み要求を許可または禁止します。 このビットが 1 にセットされているときに、ソースアドレスの拡張リピートエリアのオーバーフローが発生すると、DMDR の DTE ビットを 0 にクリアします。同時に DMDR の ESIF ビットが 1 にセットされ、ソースアドレス拡張リピートエリアオーバーフロー割り込み要求が発生したことを示します。 ブロック転送モードと併用する場合、割り込み要求は 1 ブロックサイズの転送終了後に発生します。割り込みにより転送終了したチャンネルの DMDR の DTE ビットを 1 にセットすると、転送終了した状態から再び転送を開始します。 ソースアドレスに対して拡張リピートエリアを設定していないときは、このビットは無視されます。 0: ソースアドレス拡張リピートエリアオーバーフロー割り込み要求を禁止 1: ソースアドレス拡張リピートエリアオーバーフロー割り込み要求を許可
14	—	0	R	リザーブビット
13	—	0	R	リードすると常に 0 が読み出されます。ライトは無効です。
12	SARA4	0	R/W	ソースアドレス拡張リピートエリア ソースアドレス (DSAR) に拡張リピートエリアを設定するビットです。拡張リピートエリア機能は、指定の下位アドレスをアドレス更新の対象として、残りの上位アドレスは常に固定値をとるようにして実現されます。拡張リピートエリアのサイズは、4 バイトから 128M バイトまで設定可能です。設定間隔は 2 のべき乗バイト単位です。 アドレスの増減により拡張リピートエリアからオーバーフローした下位アドレスは、アドレスが増加すると拡張リピートエリアの先頭アドレスになり、アドレスが減少すると拡張リピートエリアの最後のアドレスになります。 SARIE ビットが 1 にセットされているとき、拡張リピートエリアのオーバーフローが発生したときに割り込み要求が発生することができます。 表 7.3 に拡張リピートエリアの設定と範囲を示します。
11	SARA3	0	R/W	
10	SARA2	0	R/W	
9	SARA1	0	R/W	
8	SARA0	0	R/W	

ビット	ビット名	初期値	R/W	説明
7	DARIE	0	R/W	<p>デスティネーションアドレス拡張リピートエリアオーバフローインタラプトイネーブル</p> <p>デスティネーションアドレス拡張リピートエリアオーバフロー割り込み要求を許可または禁止します。</p> <p>このビットが1にセットされているときに、デスティネーションアドレスの拡張リピートエリアのオーバフローが発生すると、DMDR の DTE ビットを0にクリアします。同時に DMDR の ESIF ビットが1にセットされ、デスティネーションアドレス拡張リピートエリアオーバフロー割り込み要求が発生したことを示します。</p> <p>ブロック転送モードと併用する場合、割り込み要求は1ブロックサイズの転送終了後に発生します。割り込みにより転送終了したチャネルの DMDR の DTE ビットを1にセットすると、転送終了した状態から再び転送を開始します。</p> <p>デスティネーションアドレスに拡張リピートエリアを設定していないときは、このビットは無視されます。</p> <p>0: デスティネーションアドレス拡張リピートエリアオーバフロー割り込み要求を禁止</p> <p>1: デスティネーションアドレス拡張リピートエリアオーバフロー割り込み要求を許可</p>
6	—	0	R	リザーブビット
5	—	0	R	リードすると常に0が読み出されます。ライトは無効です。
4	DARA4	0	R/W	<p>デスティネーションアドレス拡張リピートエリア</p> <p>デスティネーションアドレス (DDAR) に拡張リピートエリアを設定するビットです。</p> <p>拡張リピートエリア機能は、指定の下位アドレスをアドレス更新の対象として、残りの上位アドレスは常に固定値をとるようにして実現されます。拡張リピートエリアのサイズは、4 バイトから 128M バイトまで設定可能です。設定間隔は2のべき乗バイト単位です。</p> <p>アドレスの増減により拡張リピートエリアからオーバフローした下位アドレスは、アドレスが増加すると拡張リピートエリアの先頭アドレスになり、アドレスが減少すると拡張リピートエリアの最後のアドレスになります。</p> <p>DARIE ビットが1にセットされているとき、拡張リピートエリアのオーバフローが発生したときに割り込み要求が発生することができます。</p> <p>表 7.2 に拡張リピートエリアの設定と範囲を示します。</p>
3	DARA3	0	R/W	
2	DARA2	0	R/W	
1	DARA1	0	R/W	
0	DARA0	0	R/W	

表 7.2 拡張リポートエリアの設定と範囲

SARA4~SARA0/ DARA4~DARA0 の値	拡張リポートエリアの範囲
00000	拡張リポートエリアを設定しない
00001	当該アドレスの下位 1 ビット (2 バイト) を拡張リポートエリアに設定する
00010	当該アドレスの下位 2 ビット (4 バイト) を拡張リポートエリアに設定する
00011	当該アドレスの下位 3 ビット (8 バイト) を拡張リポートエリアに設定する
00100	当該アドレスの下位 4 ビット (16 バイト) を拡張リポートエリアに設定する
00101	当該アドレスの下位 5 ビット (32 バイト) を拡張リポートエリアに設定する
00110	当該アドレスの下位 6 ビット (64 バイト) を拡張リポートエリアに設定する
00111	当該アドレスの下位 7 ビット (128 バイト) を拡張リポートエリアに設定する
01000	当該アドレスの下位 8 ビット (256 バイト) を拡張リポートエリアに設定する
01001	当該アドレスの下位 9 ビット (512 バイト) を拡張リポートエリアに設定する
01010	当該アドレスの下位 10 ビット (1k バイト) を拡張リポートエリアに設定する
01011	当該アドレスの下位 11 ビット (2k バイト) を拡張リポートエリアに設定する
01100	当該アドレスの下位 12 ビット (4k バイト) を拡張リポートエリアに設定する
01101	当該アドレスの下位 13 ビット (8k バイト) を拡張リポートエリアに設定する
01110	当該アドレスの下位 14 ビット (16k バイト) を拡張リポートエリアに設定する
01111	当該アドレスの下位 15 ビット (32k バイト) を拡張リポートエリアに設定する
10000	当該アドレスの下位 16 ビット (64k バイト) を拡張リポートエリアに設定する
10001	当該アドレスの下位 17 ビット (128k バイト) を拡張リポートエリアに設定する
10010	当該アドレスの下位 18 ビット (256k バイト) を拡張リポートエリアに設定する
10011	当該アドレスの下位 19 ビット (512k バイト) を拡張リポートエリアに設定する
10100	当該アドレスの下位 20 ビット (1M バイト) を拡張リポートエリアに設定する
10101	当該アドレスの下位 21 ビット (2M バイト) を拡張リポートエリアに設定する
10110	当該アドレスの下位 22 ビット (4M バイト) を拡張リポートエリアに設定する
10111	当該アドレスの下位 23 ビット (8M バイト) を拡張リポートエリアに設定する
11000	当該アドレスの下位 24 ビット (16M バイト) を拡張リポートエリアに設定する
11001	当該アドレスの下位 25 ビット (32M バイト) を拡張リポートエリアに設定する
11010	当該アドレスの下位 26 ビット (64M バイト) を拡張リポートエリアに設定する
11011	当該アドレスの下位 27 ビット (128M バイト) を拡張リポートエリアに設定する
111XX	設定禁止

【記号説明】 X : Don't care

7.2.8 DMA モジュールリクエストセレクトレジスタ (DMRSR)

DMRSR は、内蔵モジュール割り込み要因を指定する 8 ビットのリード/ライト可能な 8 ビットのレジスタです。

割り込み要因のベクタ番号を 8 ビットで指定します。ただし、0 は割り込み要因なしと見なします。内蔵モジュール割り込み要因のベクタ番号は、表 7.4 を参照してください。

ビット	7	6	5	4	3	2	1	0
ビット名								
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

7.3 転送モード

DMAC の転送モードを表 7.3 に示します。転送モードは、各チャンネル毎に設定可能です。

表 7.3 転送モード

アドレスモード	転送モード	起動要因	共通機能	アドレスレジスタ	
				ソース	デスティネーション
デュアルアドレスモード	<ul style="list-style-type: none"> ノーマル転送モード リピート転送モード ブロック転送モード (リピートサイズ/ブロックサイズ = 1~65,536 バイト/ワード/ロングワード) 	<ul style="list-style-type: none"> オートリクエスト (CPU より起動) 内蔵モジュール割り込み 外部リクエスト 	<ul style="list-style-type: none"> 総転送サイズ: 1~4G バイト、または指定なし オフセット加算 拡張リピートエリア機能 	DSAR	DDAR
シングルアドレスモード	<ul style="list-style-type: none"> ソース、またはデスティネーションアドレスレジスタの代わりに、$\overline{\text{DACK}}$ 端子を用いて直接外部デバイスとのデータ転送が可能 アドレスレジスタの設定以外は、上記の各転送モードを指定可能 1 バスサイクルで 1 回の転送が可能 (転送モードのバリエーションは、上記デュアルアドレスモードと同じです。) 			DSAR/ $\overline{\text{DACK}}$	$\overline{\text{DACK}}$ / DDAR

起動要因がオートリクエストのときは、サイクルスチールモード/バーストモードから選択できます。

総転送サイズを指定しない場合 (DTCR=H'00000000)、転送カウンタは停止して転送回数に制限がなくなり、永続して転送が可能になります。

7.4 動作説明

7.4.1 アドレスモード

(1) デュアルアドレスモード

デュアルアドレスモードは、転送元アドレスを DSAR、転送先アドレスを DDAR で指定して、1 回の転送を 2 バスサイクルで実行するモードです（データバス幅がデータアクセスサイズよりも小さいときや、アクセスするアドレスがデータアクセスサイズのデータ境界を外れているときは、バスサイクルが分割されるため 2 バスサイクルを超えます）。

転送動作は、最初のバスサイクルで転送元アドレスにあるデータをリードして、そのデータを次のバスサイクルで転送先アドレスへライトします。

リードサイクルとライトサイクルの間は不可分割となっています。そのため、2 つのバスサイクルの間に他のバスサイクル（他のバスマスタのサイクル、リフレッシュサイクル、外部バス解放サイクル）は発生しません。

DMDR の TEND \bar ビットにより、 $\overline{\text{TEND}}$ 出力の許可/禁止の設定ができます。 $\overline{\text{TEND}}$ は連続する 2 バスサイクルの期間出力されます。バスサイクルの直前にアイドルサイクルが挿入される場合は、アイドルサイクル期間に対しても $\overline{\text{TEND}}$ は出力されます。 $\overline{\text{DACK}}$ は出力されません。

図 7.2 にデュアルアドレスモードのタイミング例を、図 7.3 にデュアルアドレスモードの動作を示します。

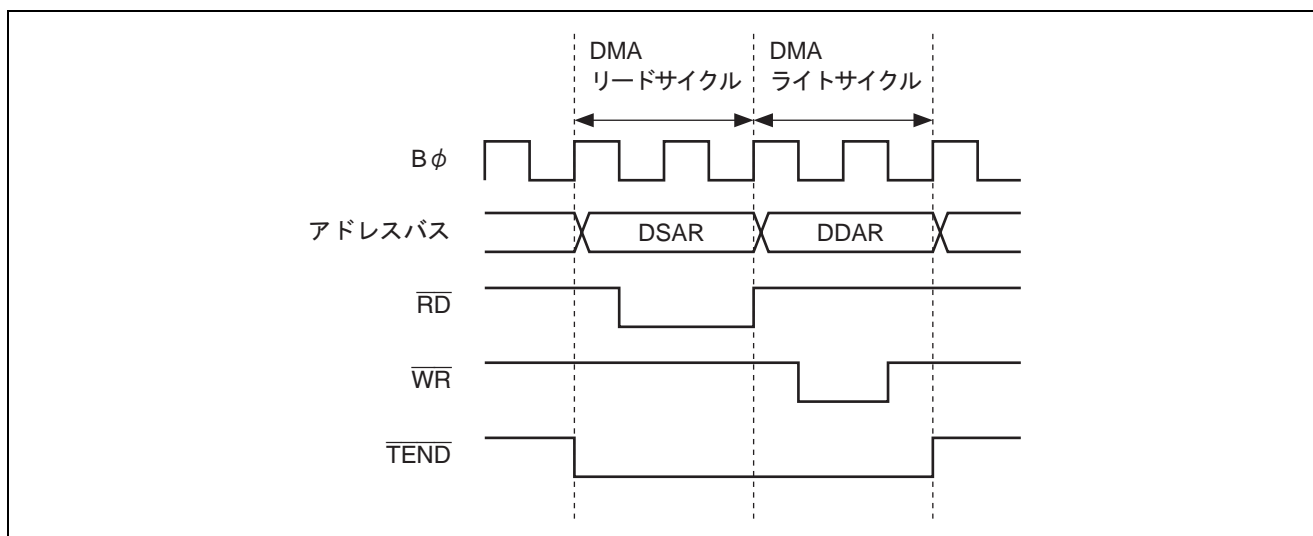


図 7.2 デュアルアドレスモードのタイミング例

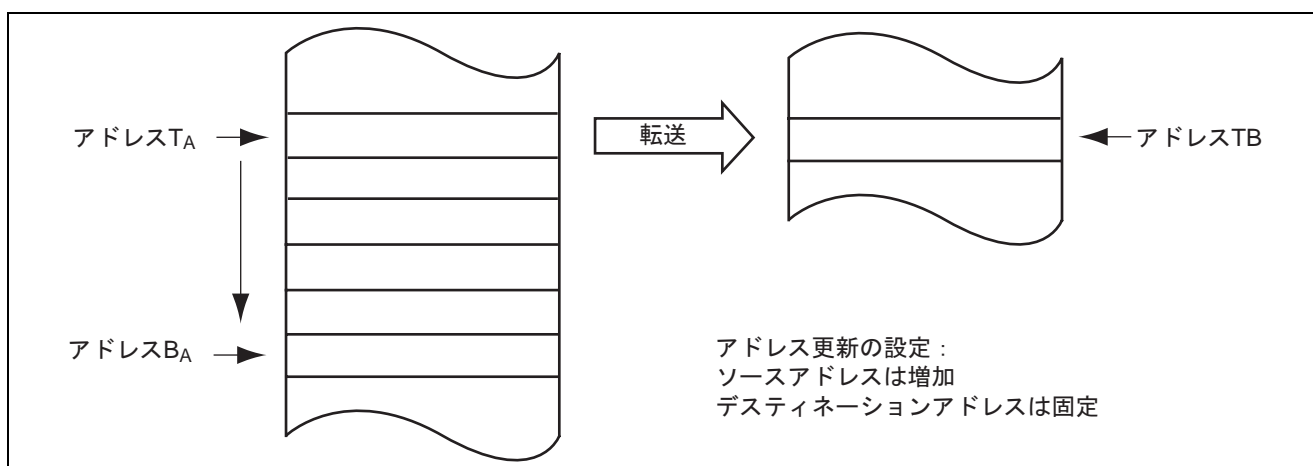


図 7.3 デュアルアドレスモードの動作

(2) シングルアドレスモード

シングルアドレスモードは、DSAR または DDAR の代わりに $\overline{\text{DACK}}$ 端子を用いて、外部デバイスと外部メモリ間で直接データを転送するモードです。1 回の転送を 1 バスサイクルで実行します。

シングルアドレスモードを使用する場合は、データバス幅とデータアクセスサイズを一致させてください。データバス幅に関しては、「第 6 章 バスコントローラ (BSC)」を参照してください。

DMAC は $\overline{\text{DACK}}$ 付き外部デバイスへのストロブ信号 ($\overline{\text{DACK}}$) を、転送元または転送先のいずれか一方の外部デバイスに出力してアクセスすると同時に、もう一方の転送相手にアドレスを出力してアクセスします。これにより、1 つのバスサイクルで DMA 転送を行うことができます。図 7.4 に示す外部メモリと $\overline{\text{DACK}}$ 付き外部デバイスとの転送例では、外部デバイスがデータバスにデータを出力するのと同じバスサイクルで、そのデータが外部メモリに書き込まれます。

転送方向は DACR の DIRS ビットにより、 $\overline{\text{DACK}}$ 付き外部デバイスを転送元にするか転送先にするかを設定できます。DIRS=0 では外部メモリ (DSAR) → $\overline{\text{DACK}}$ 付き外部デバイスへの転送、DIRS=1 では $\overline{\text{DACK}}$ 付き外部デバイス → 外部メモリ (DDAR) への転送になります。転送の対象として使用しないソースまたはデスティネーションアドレスのレジスタの設定は無視されます。

$\overline{\text{DACK}}$ 出力は、シングルアドレスモードのときに DMDR の DACKE ビットの設定により有効になります。 $\overline{\text{DACK}}$ 出力は、ローアクティブです。

DMDR の TENDE ビットにより、 $\overline{\text{TEND}}$ 出力の許可/禁止の設定ができます。 $\overline{\text{TEND}}$ は 1 バスサイクルの期間出力されます。バスサイクルの直前にアイドルサイクルが挿入される場合は、アイドルサイクル期間に対しても $\overline{\text{TEND}}$ は出力されます。

図 7.5 にシングルアドレスモードのタイミング例を、図 7.6 にシングルアドレスモードの動作を示します。

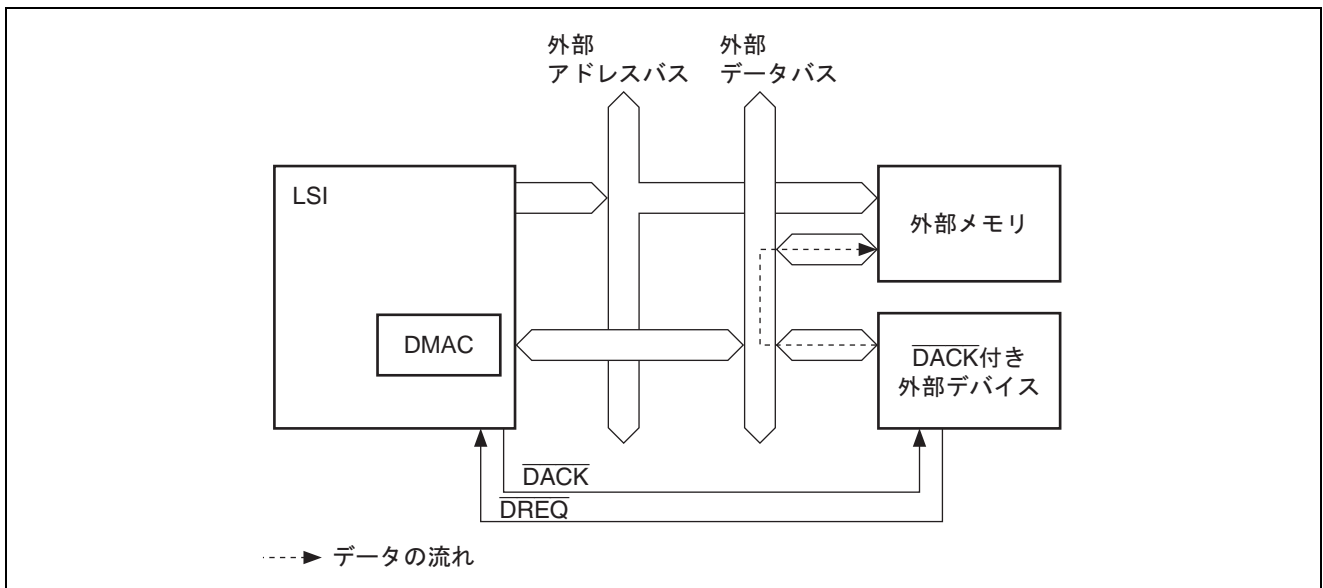


図 7.4 シングルアドレスモードでのデータの流れ

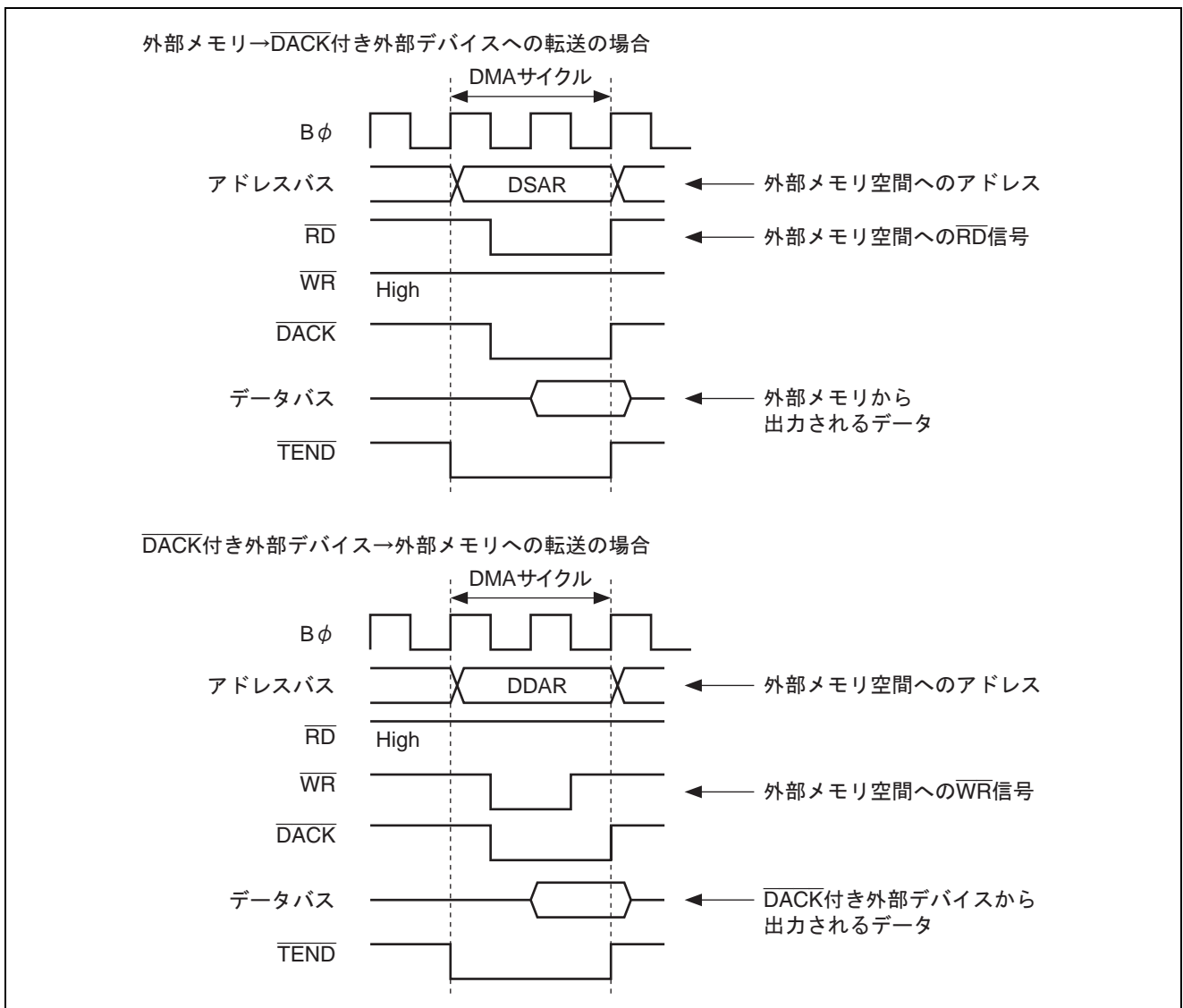


図 7.5 シングルアドレスモードのタイミング例

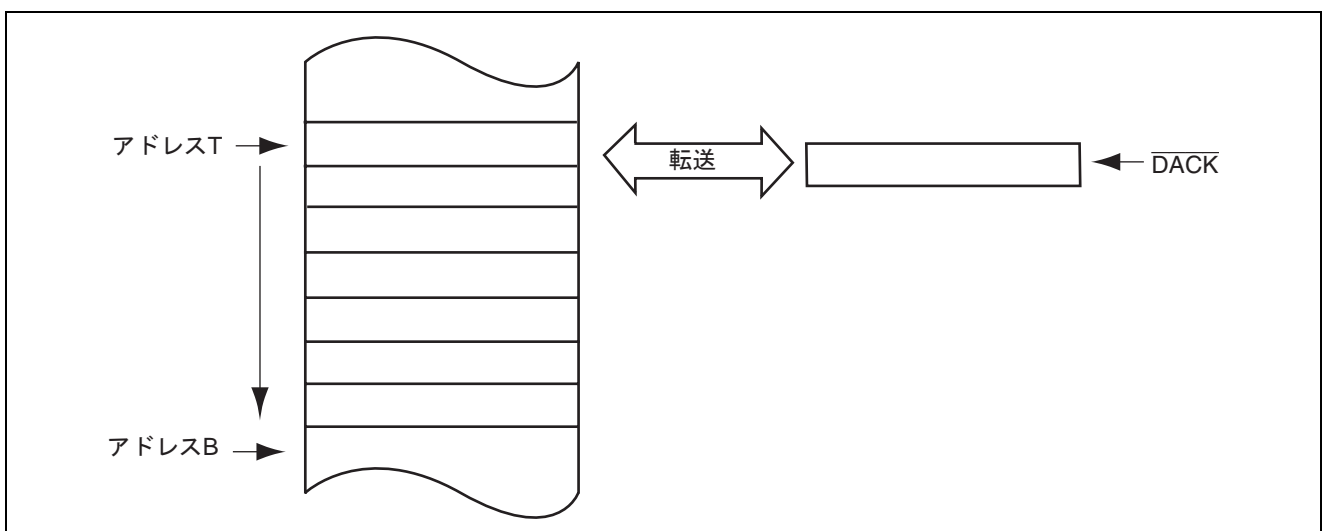


図 7.6 シングルアドレスモードの動作

7.4.2 転送モード

(1) ノーマル転送モード

ノーマル転送モードは、1回の転送要求について1データアクセスサイズ単位の転送を行います。DTCRで最大4Gバイトの総転送サイズを設定できます。DBSRはノーマル転送モードのときには無効です。

$\overline{\text{TEND}}$ 出力は、最後のDMA転送でのみ出力されます。

図7.7にノーマル転送モードのタイミング例を、図7.8にデュアルアドレスモード時のノーマル転送モードの動作を示します。

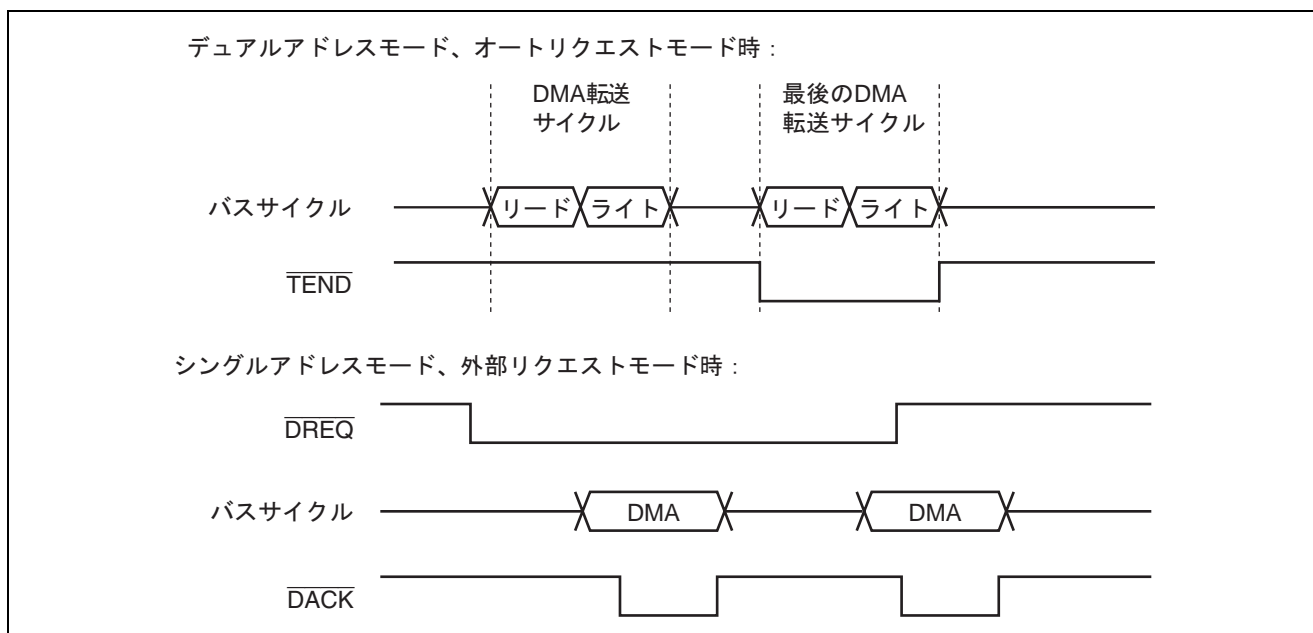


図 7.7 ノーマル転送モードのタイミング例

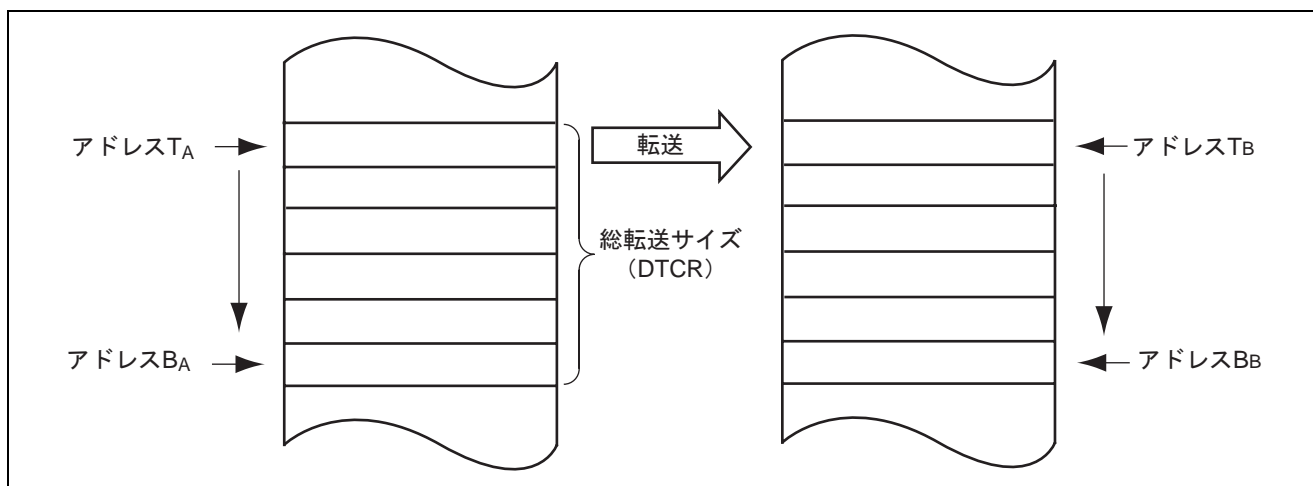


図 7.8 ノーマル転送モードの動作

(2) リpeat転送モード

リpeat転送モードは、1回の転送要求について1データアクセスサイズ単位の転送を行います。DTCRで最大4Gバイトの総転送サイズを設定できます。DBSRで最大64k×データアクセスサイズのリpeatサイズを設定できます。

DACRのARS1、ARS0ビットによって、リpeatエリアをソースアドレス側またはデスティネーションアドレス側に指定することができます。リpeatエリアに指定されたアドレス側は、リpeatサイズ分の転送を終了すると、転送開始アドレスに戻ります。この動作は、DTCRに設定した総転送サイズ分の転送が終了するまで行われます。ただし、DTCRにH'00000000を設定した場合はフリーランニングモードと見なし、DMDRのDTEビットが0にクリアされるまでリpeat転送を続けます。

また、リpeatサイズ分の転送を終了すると、DMA転送を一時停止させ、CPUまたはDTCに対してリpeatサイズ終了割り込み要求を発生させることができます。DACRのRPTIEビットが1にセットされているときに、リpeatサイズ分の転送終了後に次の転送要求が発生すると、DMDRのESIFビットを1にセットし、DMDRのDTEビットを0にクリアして転送を終了します。このとき、DMDRのESIEビットが1にセットされている場合、CPUまたはDTCに対して割り込み要求を発生させます。

\overline{TEND} 出力のDMA転送タイミングは、ノーマル転送モードと同じです。

図7.9にデュアルアドレスモード時のリpeat転送モードの動作を示します。

リpeatエリアをソースアドレス側またはデスティネーションアドレス側に指定しないときの動作は、図7.8のノーマル転送モードの動作と同じです。この場合も、リpeatサイズ分の転送を終了すると、リpeatサイズ終了割り込みを発生させることができます。

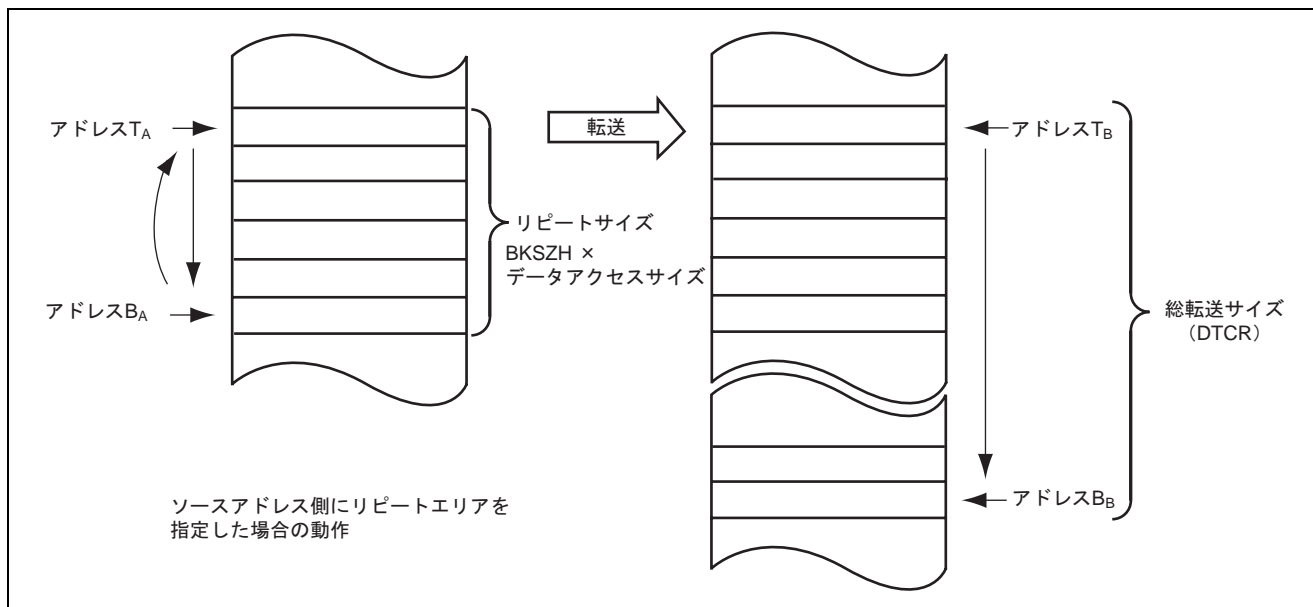


図 7.9 リpeat転送モードの動作

(3) ブロック転送モード

ブロック転送モードは、1回の転送要求について1ブロックサイズ単位の転送を行います。DTCRで最大4Gバイトの総転送サイズを設定できます。DBSRで最大64k×データアクセスサイズのブロックサイズを設定できます。

1ブロック分の転送中は、他のチャンネルの転送要求は待たされます。1ブロック分の転送が終了すると、バス権を他のバスマスタに解放します。

DACRのARS1、ARS0ビットによって、ブロックエリアをソースアドレス側もしくはデスティネーションアドレス側に指定することができます。ブロックエリアに指定されたアドレス側は、1ブロック分の転送終了ごとに、転送開始アドレスに戻ります。ソースアドレス側にもデスティネーションアドレス側にもリピートエリアを指定しない場合は、アドレスは転送開始アドレスに戻らずシーケンシャルに進みます。リピートサイズ終了割り込みを発生させることができます。

TENDは、1ブロック分の転送毎に各ブロックの終わりのDMA転送サイクルで出力されます。起動要因として外部リクエストを使用する場合は、DREQ端子のサンプリング方法をLowレベル検出(DREQS=0)に設定してください。

ブロック転送モードで拡張リピートエリアオーバフロー割り込みを設定する場合には注意が必要です。詳細は「7.4.5 拡張リピートエリア機能」を参照してください。

図7.10にブロック転送モードのDMA転送タイミング例を示します。転送条件は次の通りです。

- アドレスモード：シングルアドレスモード
- データアクセスサイズ：バイト
- 1ブロックサイズ：3バイト

図7.11にシングルアドレスモードのブロック転送モードの動作を、図7.12にデュアルアドレスモードのブロック転送モードの動作を示します。

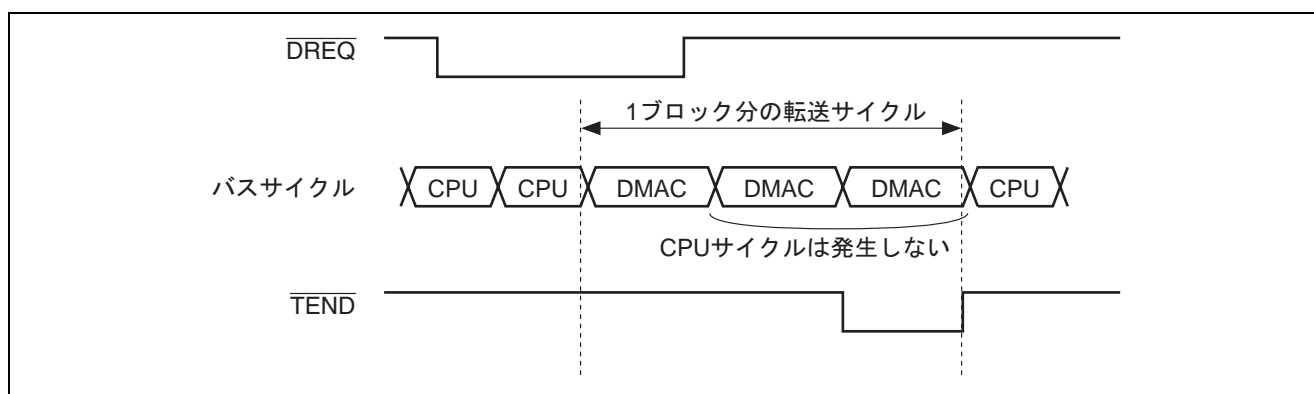


図 7.10 ブロック転送モードの例

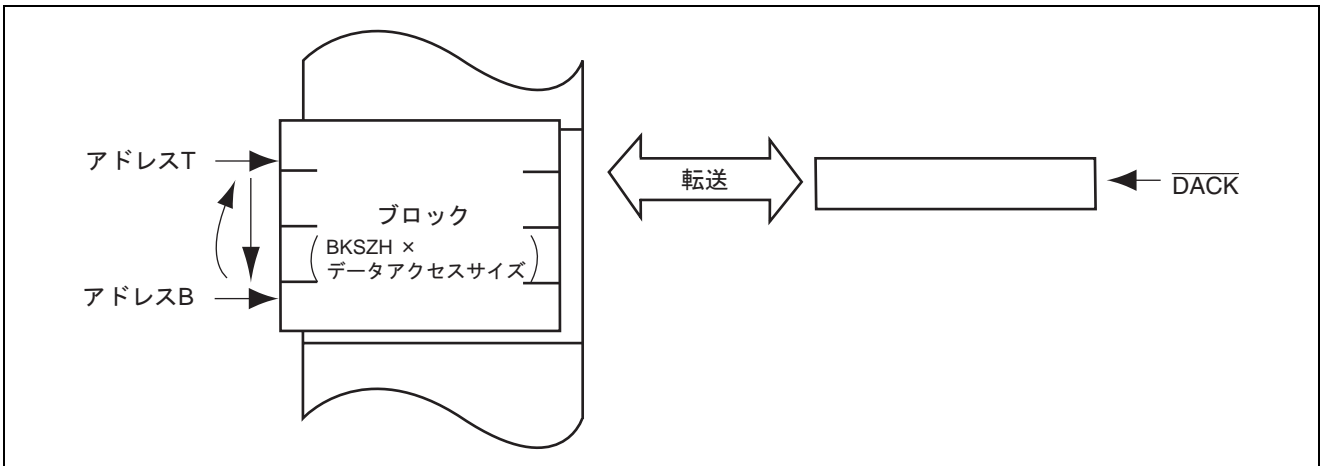


図 7.11 シングルアドレスモードのブロック転送モード（ブロックエリアの指定あり）の動作

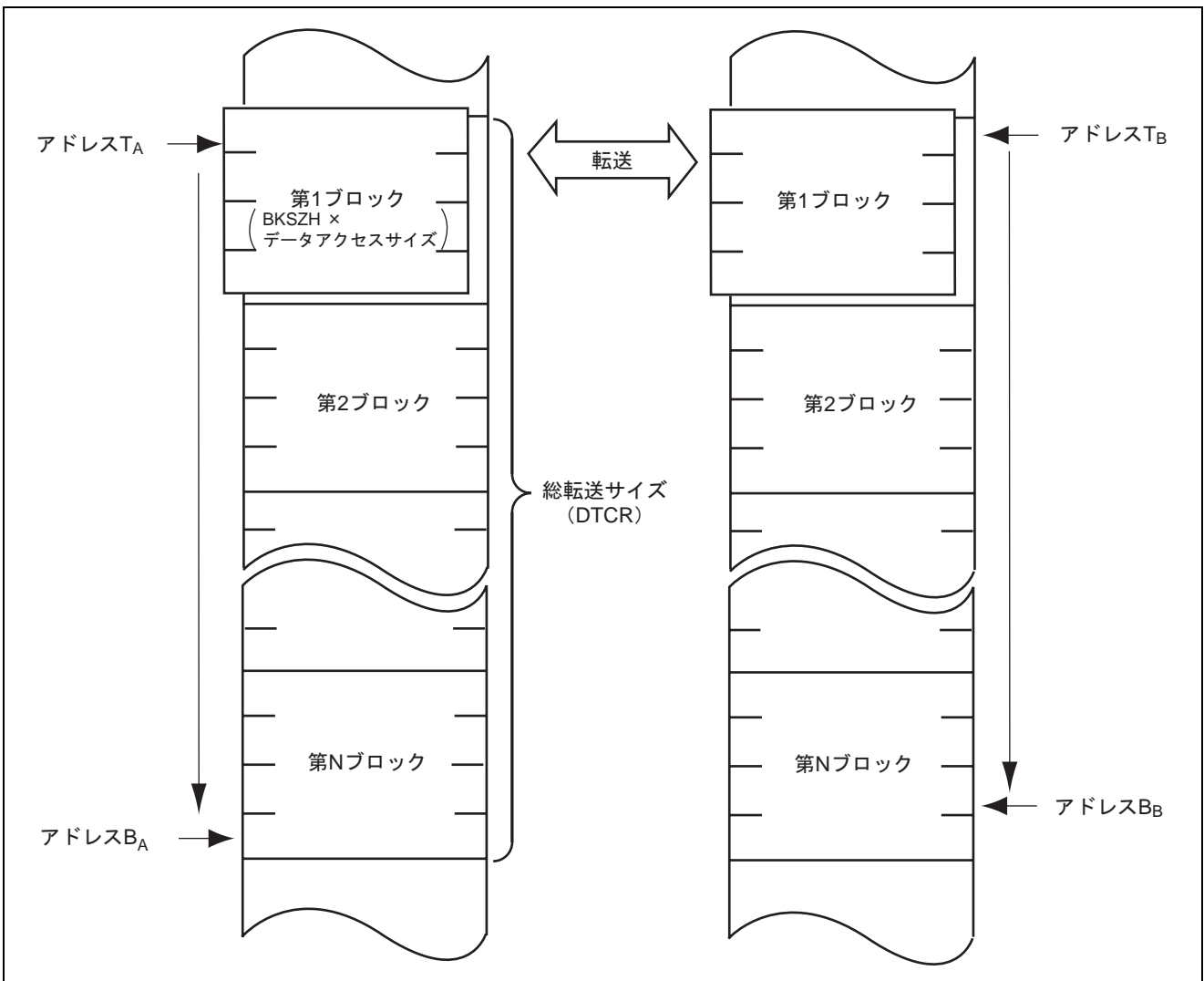


図 7.12 デュアルアドレスモードのブロック転送モード（ブロックエリアの指定なし）の動作

7.4.3 起動要因

DMAC の起動要因には、オートリクエスト、内蔵モジュール割り込み、および外部リクエストがあります。これらの起動要因は DMDR の DTF1、DTF0 ビットで選択します。

(1) オートリクエストによる起動

オートリクエストは、メモリとメモリ間の転送や、転送要求信号を発生できない周辺モジュールとメモリ間の転送のように、外部や周辺モジュールから転送要求信号の発生がない場合に、DMAC 内部で自動的に転送要求信号を発生させます。オートリクエストによる起動では、DMDR の DTE ビットを 1 にセットすると転送が開始されます。オートリクエストでは、バスモードをサイクルスチールモードとバーストモードから選択することができます。

(2) 内蔵モジュール割り込みによる起動

内蔵モジュール割り込みは、各周辺モジュールからの割り込み要求（内蔵モジュール割り込み）を転送要求信号として使用します。DMA 転送が許可されているとき (DTE=1)、内蔵モジュール割り込み要求により DMA 転送を開始します。

起動要因の内蔵モジュール割り込みは、DMA モジュールリクエストセレクトレジスタ (DMRSR) で選択します。チャンネル毎に独立して設定することができます。表 7.4 に DMAC 内蔵モジュール割り込み一覧を示します。

起動要因として選択された割り込み要求は、CPU、DTC に対しても同時に割り込み要求を発生させることができます。詳細は「第 5 章 割り込みコントローラ」を参照してください。

DMAC は、内蔵モジュール割り込み要求による起動要因を、割り込みコントローラとは独立して受け付けます。このため、割り込みコントローラによる優先順位設定の影響を受けません。

DTA=1 で DMAC を起動する場合、割り込み要求フラグは DMA 転送により自動的にクリアされます。複数のチャンネルで同一の割り込み要求を起動要因とした場合、最も優先順位の高いチャンネルが起動された時点で割り込み要求フラグがクリアされます。この場合、その他のチャンネルの転送要求は DMAC 内部で保持されず、起動されない場合があります。

DTA=0 で DMAC を起動する場合、割り込み要求フラグは DMAC によりクリアされません。CPU または DTC 転送でクリアしてください。

DTE=0 の状態では、選択された起動要因は DMAC に転送を要求しません。この場合、当該割り込みは CPU または DTC に割り込みを要求します。

また、DTE ビットへの 1 ライトは、内蔵モジュール割り込みの発生元となる割り込み要求フラグビットが 0 にクリアされていることを確認した後に行ってください。

表 7.4 DMAC 内蔵モジュール割り込み一覧

内蔵モジュール割り込み要因 (割り込み要因)	発生元	DMRSR 値 (ベクタ番号)
ADI0 (A/D 変換終了割り込み)	A/D_0	86
ADI1 (A/D 変換終了割り込み)	A/D_1	87
TGI0A (TGR0A インพุットキャプチャ/コンペアマッチ)	TPU_0	88
TGI1A (TGR1A インพุットキャプチャ/コンペアマッチ)	TPU_1	93
TGI2A (TGR2A インพุットキャプチャ/コンペアマッチ)	TPU_2	97
TGI3A (TGR3A インพุットキャプチャ/コンペアマッチ)	TPU_3	101
TGI4A (TGR4A インพุットキャプチャ/コンペアマッチ)	TPU_4	106
TGI5A (TGR5A インพุットキャプチャ/コンペアマッチ)	TPU_5	110
RM0_0 (RCAN-TL1 チャンネル 0 のメールボックス 0 のメッセージ受信割り込み)	RCAN_0	132
RM0_1 (RCAN-TL1 チャンネル 1 のメールボックス 1 のメッセージ受信割り込み)	RCAN_1	134
RX13 (SCI チャンネル 3 の受信データフル割り込み)	SCI_3	157
TX13 (SCI チャンネル 3 の送信データエンブティ割り込み)	SCI_3	158
RX14 (SCI チャンネル 4 の受信データフル割り込み)	SCI_4	161
TX14 (SCI チャンネル 4 の送信データエンブティ割り込み)	SCI_4	162
TGI6A (TGR6A インพุットキャプチャ/コンペアマッチ)	TPU_6	164
TGI7A (TGR7A インพุットキャプチャ/コンペアマッチ)	TPU_7	169
TGI8A (TGR8A インพุットキャプチャ/コンペアマッチ)	TPU_8	173
TGI9A (TGR9A インพุットキャプチャ/コンペアマッチ)	TPU_9	177
TGI10A (TGR10A インพุットキャプチャ/コンペアマッチ)	TPU_10	182
TGI11A (TGR11A インพุットキャプチャ/コンペアマッチ)	TPU_11	188
SPRI_0 (RSPI チャンネル 0 の受信データフル割り込み)	RSPI_0	197
SPTI_0 (RSPI チャンネル 0 の送信データエンブティ割り込み)	RSPI_0	198
SPRI_1 (RSPI チャンネル 1 の受信データフル割り込み)	RSPI_1	200
SPTI_1 (RSPI チャンネル 1 の送信データエンブティ割り込み)	RSPI_1	201
SPRI_2 (RSPI チャンネル 2 の受信データフル割り込み)	RSPI_2	203
SPTI_2 (RSPI チャンネル 2 の送信データエンブティ割り込み)	RSPI_2	204
SPRI_3 (RSPI チャンネル 3 の受信データフル割り込み)	RSPI_3	206
SPTI_3 (RSPI チャンネル 3 の送信データエンブティ割り込み)	RSPI_3	207

(3) 外部リクエストによる起動

外部リクエストによる起動は、外部デバイスからの転送要求信号 ($\overline{\text{DREQ}}$) によって転送を開始します。DMA 転送が許可されているとき ($\text{DTE}=1$)、 $\overline{\text{DREQ}}$ が入力されると DMA 転送を開始します。内部空間から内部空間への DMA 転送を行う場合は、起動要因を外部リクエスト以外 (オートリクエスト、内蔵モジュール割り込み) に設定してください。

転送要求信号は $\overline{\text{DREQ}}$ 端子で受け付けます。 $\overline{\text{DREQ}}$ を立ち下がりエッジで検出するか、Low レベルで検出するかは、DMDR の DREQS ビットで選択します。

起動要因として外部リクエストを指定する場合は、あらかじめ該当する端子の DDR ビットを 0、ICR ビットを 1 に設定しておいてください。詳細は「第 9 章 I/O ポート」を参照してください。

7.4.4 バスモード

バスモードには、サイクルスチールモードとバーストモードがあります。

起動要因がオートリクエストの場合は、サイクルスチールモードとバーストモードを DMDR の DTF0 ビットで選択することができます。起動要因が内蔵モジュール割り込み、または外部リクエストの場合は、サイクルスチールモードになります。

(1) サイクルスチールモード

サイクルスチールモードでは、DMAC は 1 回の転送単位 (バイト、ワード、ロングワード、または 1 ブロックサイズ) の転送を終了するたびにバス権を解放します。その後、転送要求があればバス権を取り戻し、再び 1 転送単位の転送を行い、その転送を終了するとまたバス権を解放します。この動作を転送終了条件が満たされるまで繰り返します。

DMA 転送中に他のチャンネルに転送要求がある場合は、いったん他のバスマスタに対してバス権を解放した後に、転送要求のあったチャンネルの転送を行います。複数のチャンネルに転送要求がある場合の動作の詳細については、「7.4.8 チャンネルの優先順位」を参照してください。

図 7.13 にサイクルスチールモードのタイミング例を示します。転送条件は以下のとおりです。

- アドレスモード：シングルアドレスモード
- $\overline{\text{DREQ}}$ 端子のサンプリング方法：Low レベル検出

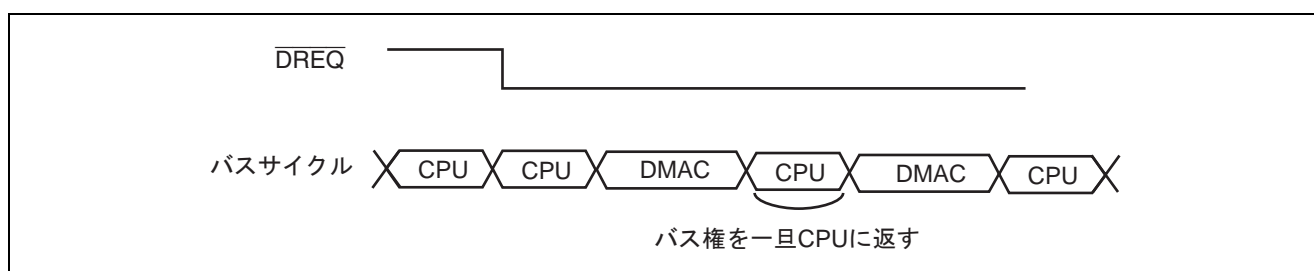


図 7.13 サイクルスチールモードのタイミング例

(2) バーストモード

バーストモードでは、DMAC は一度バス権を取ると転送終了条件が満たされるまでバス権を解放せずに転送を続けます。バーストモードでは、転送を開始すると優先順位の高い他のチャンネルに転送要求がある場合でも転送を中断することはありません。バーストモードのチャンネルが転送を終了すると、次のサイクルでサイクルスチールモードと同様にバス権を解放します。ただし、バスコントローラの BCR2 の IBCCS ビットを 1 にセットすることにより、他のバスマスタを優先して DMAC のバス権をいったん解放することができます。

ブロック転送モードでは、バーストモードの設定は無効です (1 ブロック転送中はバーストモードと同様の動作)。常にサイクルスチールモードとして動作します。

DMDR の DTE ビットを 0 にクリアすると、DMA 転送は停止します。ただし、DTE ビットを 0 にクリアするまでに DMAC 内部で発生した転送要求分の DMA 転送は実行されます。転送サイズエラー割り込みや、リピートサイズ終了割り込み、拡張リピートエリアオーバフロー割り込みが発生した場合は、DTE ビットを 0 にクリアして転送終了します。

図 7.14 にバーストモードのタイミング例を示します。

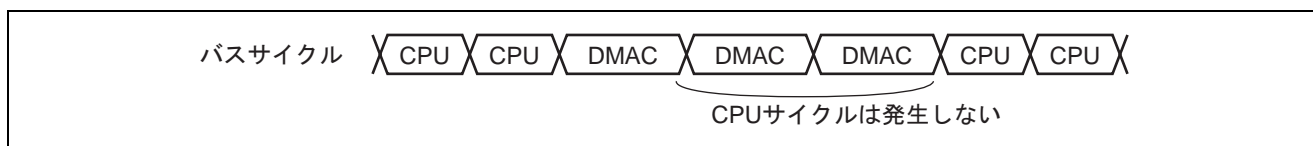


図 7.14 バーストモードのタイミング例

7.4.5 拡張リピートエリア機能

DMAC にはソースアドレス、デスティネーションアドレスに拡張リピートエリアを設定する機能があります。拡張リピートエリアを設定すると、アドレスレジスタは拡張リピートエリアに指定された範囲のアドレス値を繰り返します。リングバッファを転送の対象にした場合は、アドレスレジスタの値がバッファの最終アドレスになるたびに（リングバッファに対するアドレスのオーバフロー）、アドレスレジスタの値をバッファの先頭アドレスに戻す操作が必要となり、拡張リピートエリア機能を使うと自動的にアドレスレジスタの値をバッファの先頭アドレスに戻す操作を DMAC 内で行うことができます。

拡張リピートエリア機能は、ソースアドレスレジスタ (DSAR) とデスティネーションアドレスレジスタ (DDAR) に独立して設定できます。

ソースアドレスの拡張リピートエリアは、DACR の SARA4~SARA0 ビットで設定します。デスティネーションアドレスの拡張リピートエリアは、DACR の DARA4~DARA0 ビットで設定します。各々の拡張リピートエリアのサイズは独立に設定できます。

アドレスレジスタの値が拡張リピートエリアの終端になり拡張リピートエリアがオーバフローすると、DMA 転送を一時停止させて、CPU に対して拡張リピートエリアオーバフロー割り込み要求を発生させることができます。DACR の SARIE ビットを 1 にセットすると、DSAR の拡張リピートエリアがオーバフローしたときに DMDR の ESIF ビットを 1 セットし、DMDR の DTE ビットを 0 にクリアして転送を終了します。このとき、DMDR の ESIE ビットが 1 にセットされていると、CPU に対して拡張リピートエリアオーバフロー割り込み要求を発生します。DACR の DARIE ビットを 1 にセットするとデスティネーションアドレスレジスタが対象になります。割り込み発生中に DMDR の DTE ビットに 1 をセットすると、引き続き転送を再開します。

図 7.15 に拡張リピートエリア機能の例を示します。

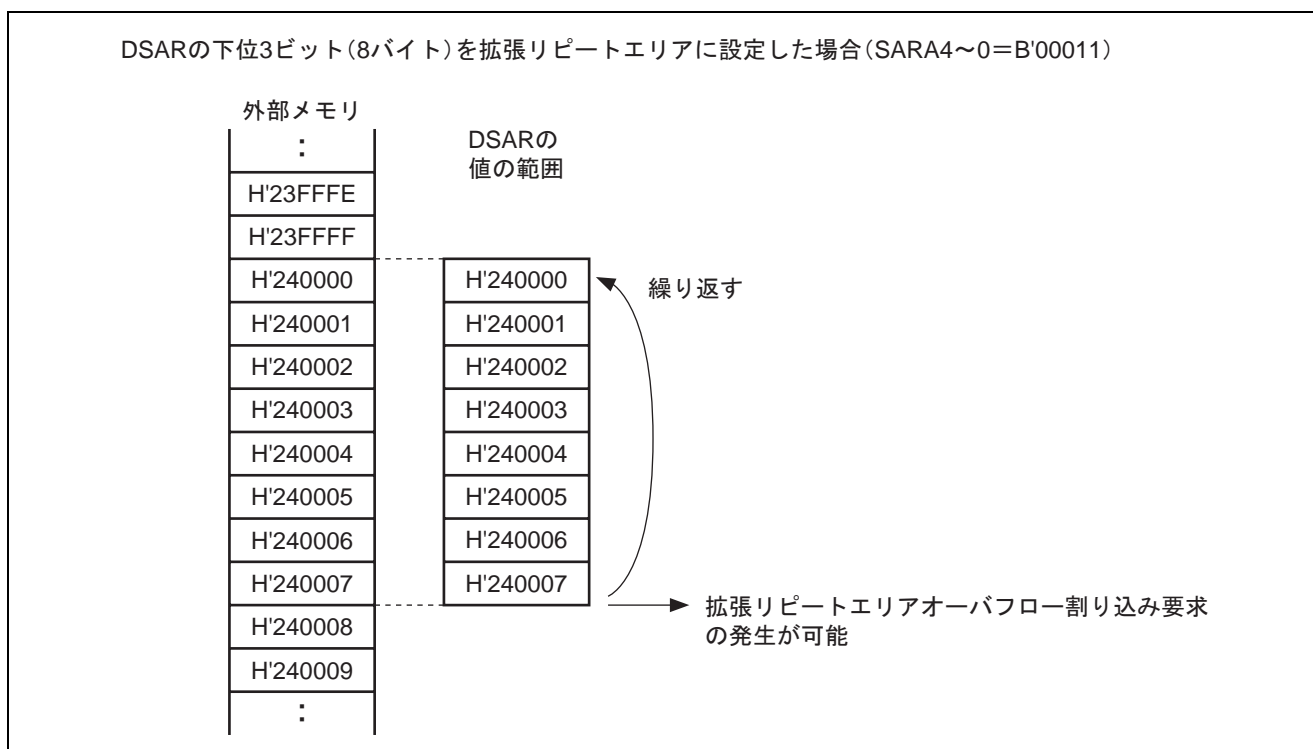


図 7.15 拡張リピートエリア機能の例

拡張リピートエリアオーバフロー割り込みをブロック転送モードと併用する場合は、次の注意が必要です。
 拡張リピートエリアのオーバフローの発生で転送を終了させる場合は、ブロックサイズを2のべき乗になるように設定するか、またはブロックサイズの切れ目と拡張リピートエリアの範囲の切れ目が一致するようにアドレスレジスタの値を設定する必要があります。また、1ブロックサイズを転送中に拡張リピートエリアにオーバフローが発生した場合は、1ブロックサイズの転送が終了するまで拡張リピートエリアオーバフロー割り込み要求は保留され、転送はオーバーランします。

拡張リピートエリアオーバフロー割り込みをブロック転送モードと併用するときと同様の注意が必要です。
 図 7.16 にブロック転送モードと拡張リピートエリア機能を併用したときの例を示します。

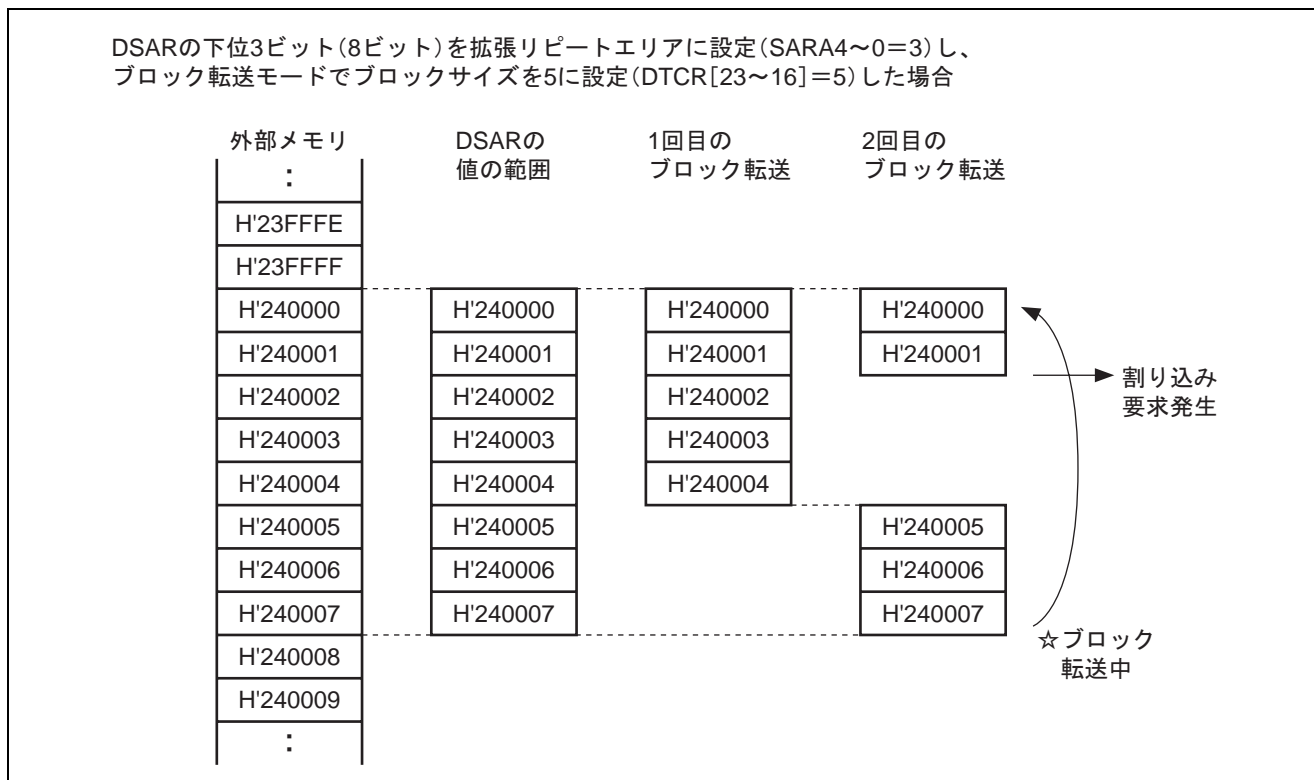


図 7.16 ブロック転送モードと拡張リピートエリア機能を併用したときの例

7.4.6 オフセットを使ったアドレス更新機能

転送先、転送元のアドレスの更新方法には、「固定」、「1、2または4の増減」の他に「オフセット加算」があります。「オフセット加算」では、DMACがデータアクセスサイズ分の転送を行うたびにオフセットレジスタ (DOFR) で設定したオフセットを加算します。この機能により、途中のアドレスを飛ばしてデータ転送ができます。

図 7.17 にアドレス更新方法を示します。

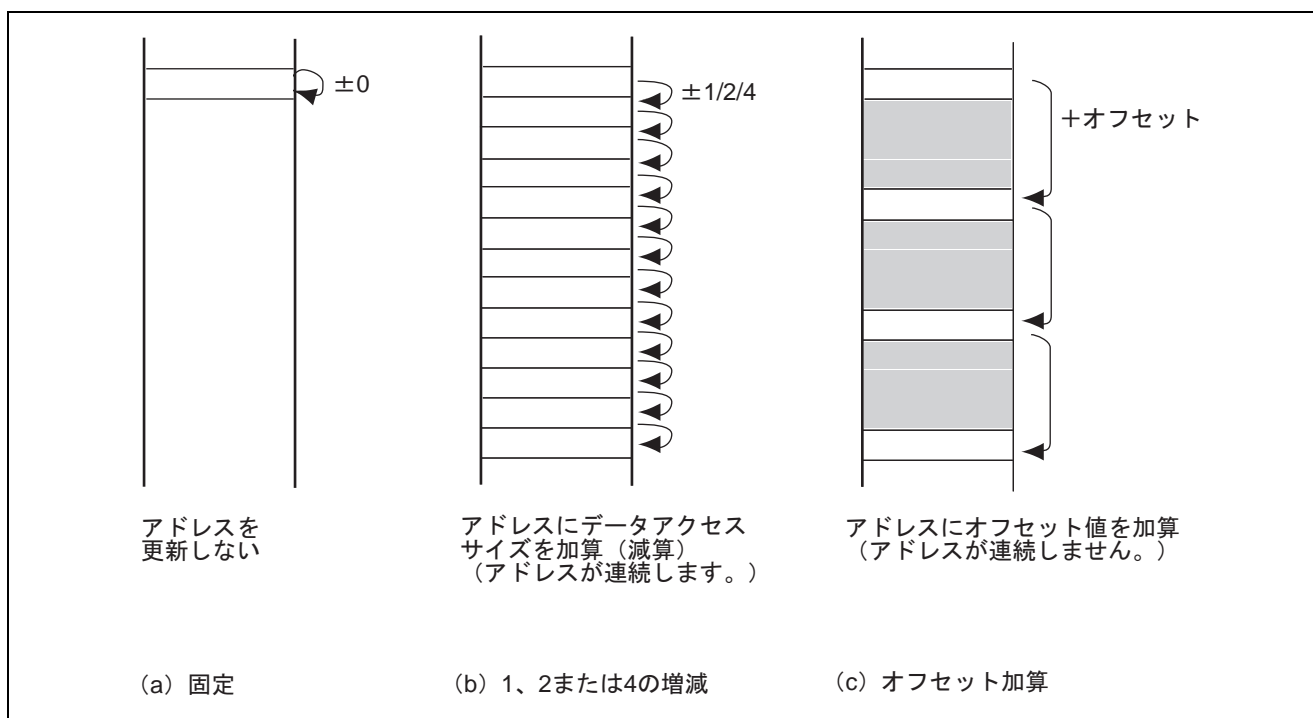


図 7.17 アドレスの更新方法

(a) の「固定」の場合は、転送先あるいは転送元のアドレスの更新が行われず常に同じアドレスを示します。

(b) の「1、2または4の増減」の場合は、データの転送が行われるたびにデータアクセスサイズに応じた値を転送先あるいは転送元のアドレスに加減算します。データアクセスサイズにはバイト、ワード、ロングワードを指定できます。バイト指定では1、ワード指定では2、ロングワード指定では4の値をアドレスの加減算に使用します。この機能により、DMACは連続するアドレスの転送を実現しています。

(c) の「オフセット加算」の場合は、データアクセスサイズに依存するアドレスの演算を実施しません。DMACはデータアクセスサイズ分の転送が行われるたびにDOFRで設定した値を転送先あるいは転送元のアドレスに加算します。

DMACは、DOFRにオフセット値を設定し、DSARやDDARと演算します。このとき、DMACはオフセット値の加算しか実行できませんが、DOFRに負値を設定することにより、オフセット値による減算も実現できます。負値を設定する場合は、負値にするオフセット値を2の補数で指定してください。

(1) オフセットを使った基本的な転送

図 7.18 に基本的な動作を示します。

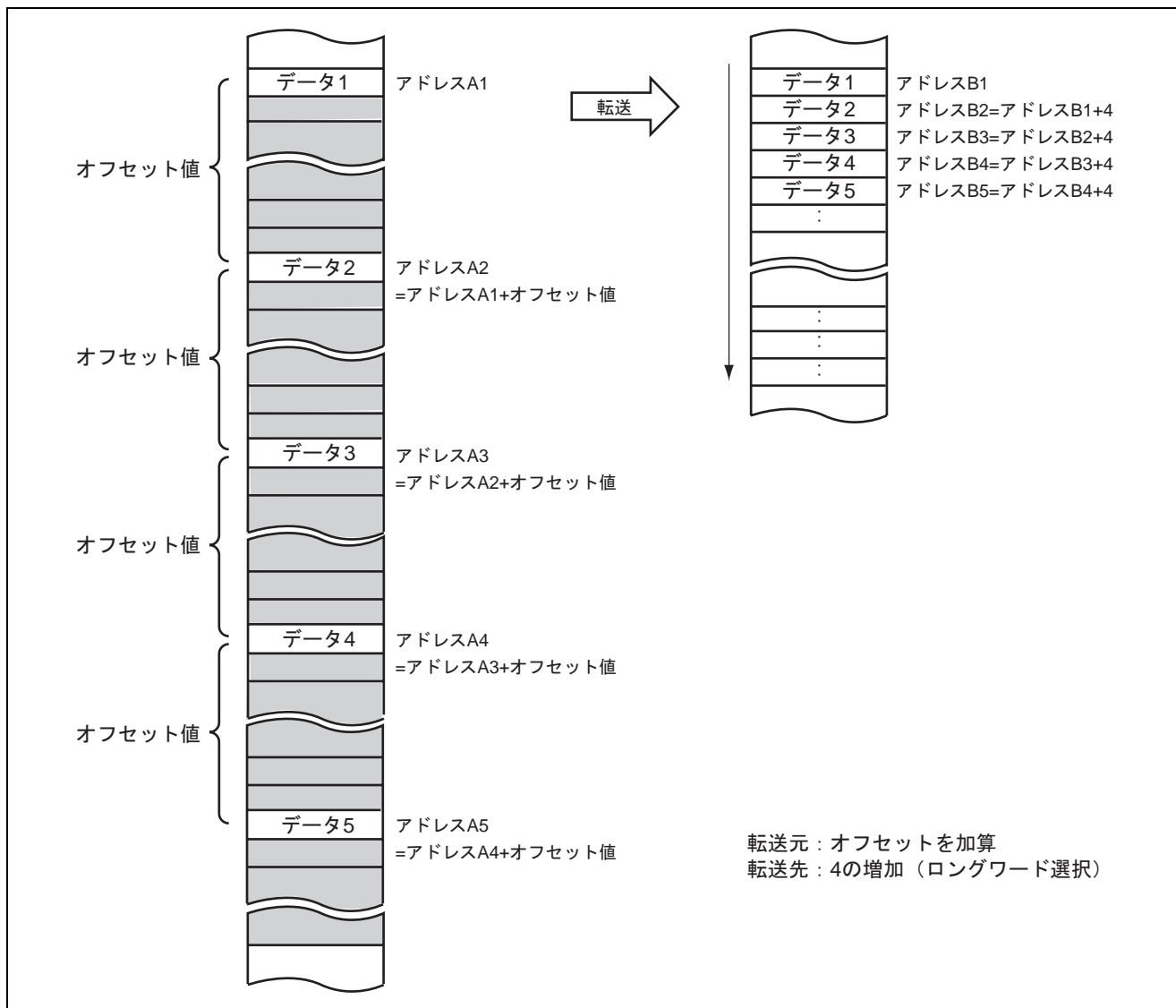


図 7.18 オフセット更新機能の動作

図 7.18 では、転送元アドレスの更新には「オフセット加算」を設定し、転送先アドレスの更新に「1、2、または4の増減」を設定しています。転送元アドレスの2回目以降の更新は、前の転送時のアドレスからオフセット値分ジャンプしたアドレスのデータのリードとなります。この一定間隔を空けてリードしてきたデータは、転送先では連続した領域にライトされます。

(2) オフセットを使ったXY変換例

図 7.19 にリピート転送モードと「オフセット加算」を組み合わせるときの動作を示します。

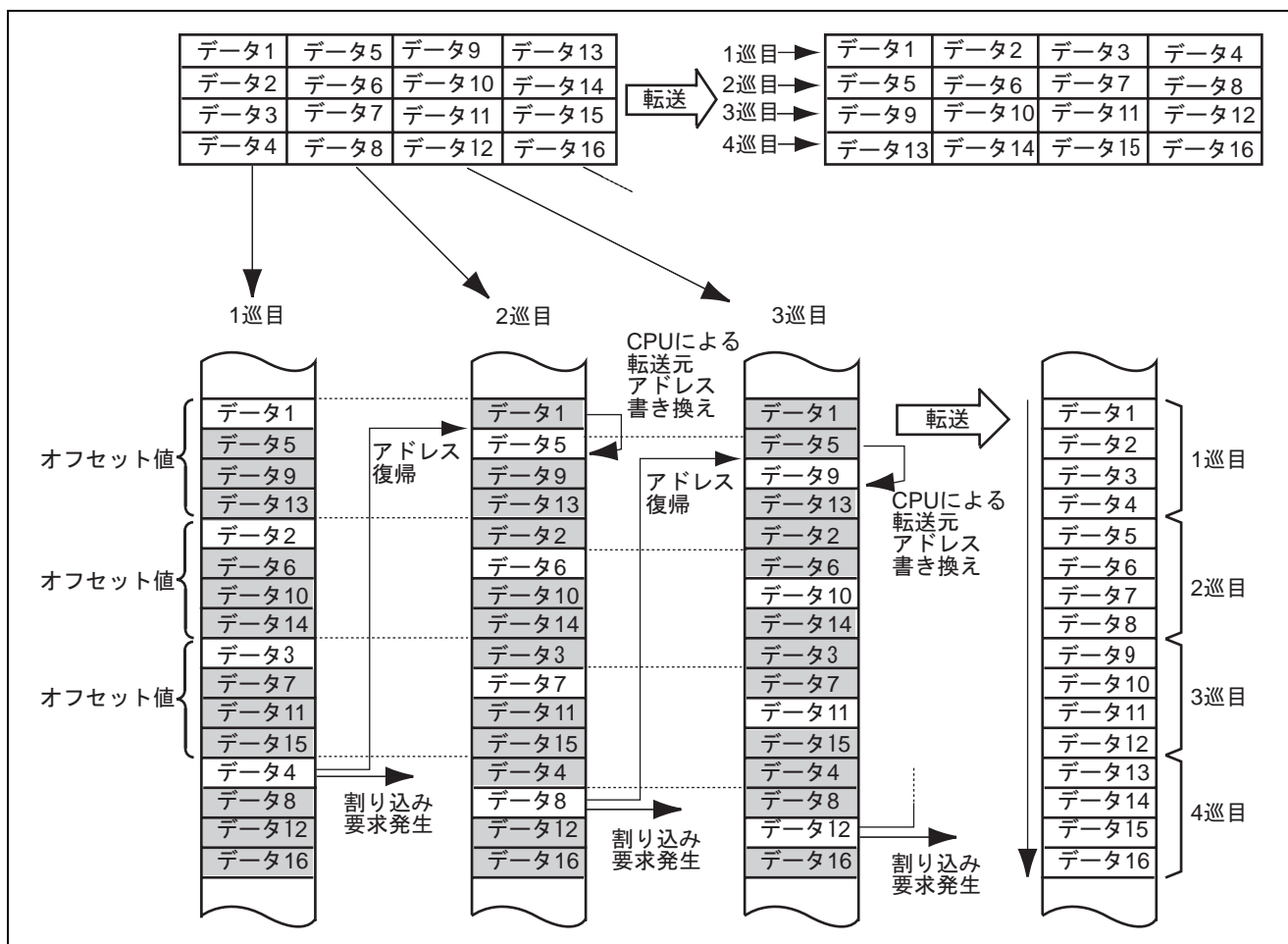


図 7.19 リピート転送モード+オフセット加算による XY 変換のときの動作

図 7.19 では、DACR でソースアドレス側をリピートエリアに設定し、かつ DACR で「オフセット加算」を設定しています。オフセット値は、4×データアクセスサイズに相当するアドレスです（たとえば、データアクセスサイズがロングワードであるならば、DOFR に H'00000010 を指定したことになります）。リピートサイズは 4×データアクセスサイズです（たとえばデータアクセスサイズがロングワードであると、4×4=16 バイトをリピートサイズに指定したことになります）。転送先は「1、2、および 4 の加算」を設定しています。また、DACR の RPTIE ビットを 1 にセットし、リピートサイズ分の転送が終了するとリピートサイズ終了割り込み要求が発生するように設定しています。

転送が開始されると、転送元はアドレスにオフセット値を加算しデータを転送します。転送データは、転送先で転送順に連続して並べられます。データ 4 までのデータが転送されると、リピートサイズ分のデータを転送したことになり、DMAC は転送元のアドレスを転送開始時のアドレス（転送元データ 1 のアドレス）に復帰させます。また、同時にリピートサイズ終了割り込み要求を発生させます。この割り込み要求によりいったん転送が中断するので、CPU による I/O レジスタアクセスで DSAR の値をデータ 5 のアドレスに書き換えてください（ロングワード転送ならデータ 1 のアドレス+4 にアドレスを書き換えます）。DMDR の DTE ビットを 1 にセットすると、転送中断した状態から引き続き転送を開始します。以降同様な処理を繰り返すと、転送元のデータが転送先に XY 変換されて転送されます。

図 7.20 に XY 変換の処理フローを示します。

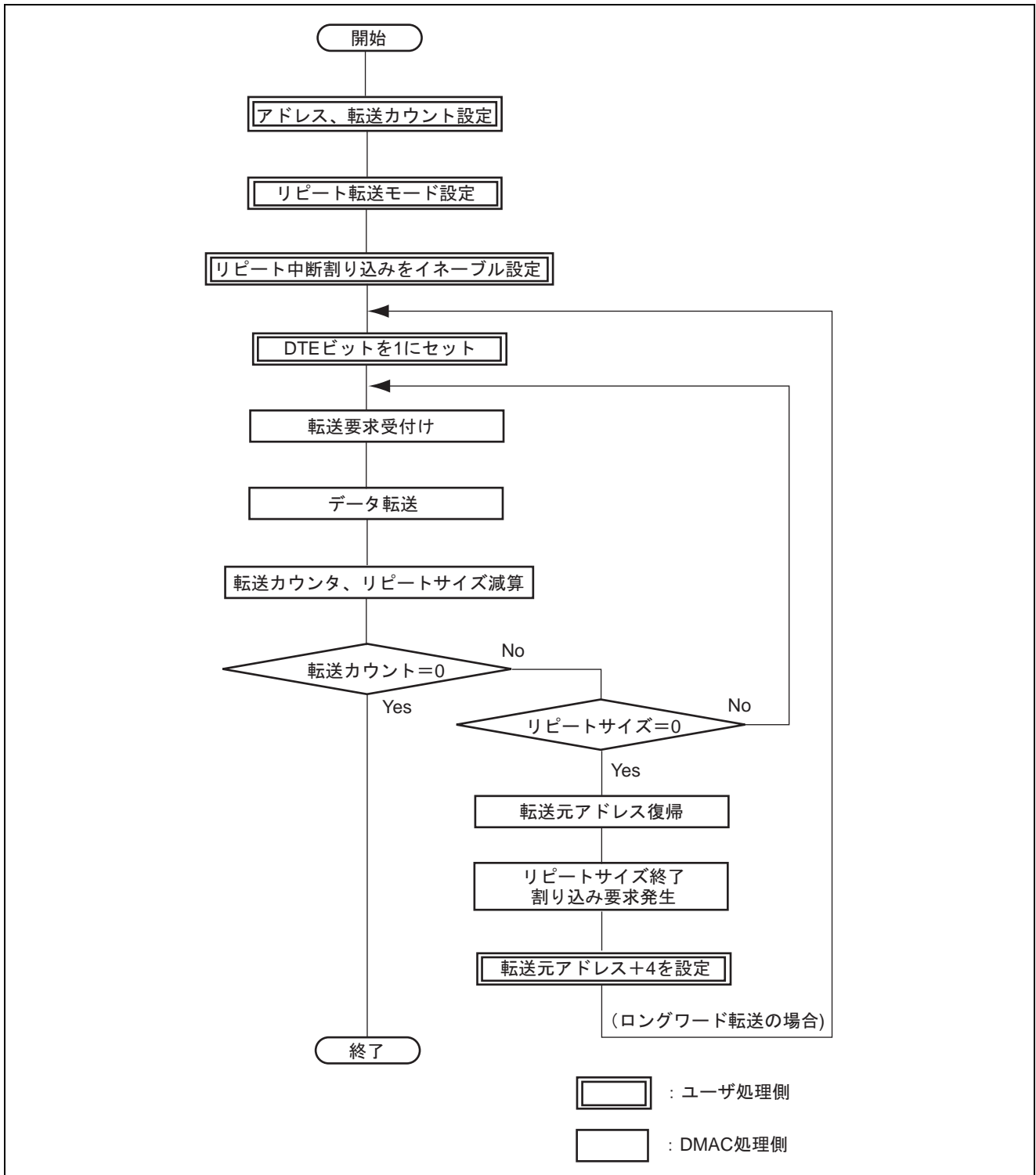


図 7.20 リピート転送モード+オフセット加算による XY 変換のフロー

(3) オフセット減算の指定方法

DOFR に負値を設定する場合はオフセット値を 2 の補数で指定してください。2 の補数は次式で求められます。

$$[\text{負オフセット値の 2 の補数表現}] = \sim[\text{オフセット値}] + 1 \quad (\sim: \text{ビット反転})$$

例: H'0001FFFF の 2 の補数表現
 = H'FFFE0000 + H'00000001
 = H'FFFE0001

2 の補数は、CPU の NEG.L 命令でも求められます。

7.4.7 DMA 転送中のレジスタ

DMAC のレジスタは、DMA 転送処理により値を更新します。更新される値は、各種設定や転送の状態により異なります。更新されるレジスタは、DSAR、DDAR、DTCR、DBSR の BKSZH、BKSZ ビット、DMDR の DTE、ACT、ERRF、ESIF、DTIF ビットです。

(1) DMA ソースアドレスレジスタ (DSAR)

転送元の DSAR のアドレスをアクセスすると、DSAR の値を出力し、次にアクセスするアドレスに更新されます。

DACR の SAT1、SAT0 ビットでアドレスの増減を設定します。SAT1、0=B'00 のときアドレスは固定されます。SAT1、0=B'01 のときアドレスはオフセットレジスタ値が加算されます。SAT1、0=B'10 のときアドレスは増加し、SAT1、0=B'11 のときアドレスは減少します（増減サイズは、転送するデータアクセスサイズにより決まります）。

DMDR の DTSZ1、DTSZ0 ビットでデータアクセスサイズを設定します。DTSZ1、0=B'00 のときバイトサイズになり、アドレスは±1 増減されます。DTSZ1、0=B'01 のときワードサイズになり、アドレスは±2 増減されます。DTSZ1、0=B'10 のときロングワードサイズになり、アドレスは±4 増減されます。また、ワードサイズ、ロングワードサイズの設定であっても、ソースアドレスがワード境界、ロングワード境界を外れている場合、データはバイトやワードに分割してリードされます。1 ワードサイズ分、1 ロングワードサイズ分のデータを分割してリードしている間、アドレス増減サイズは実際にリードされるデータのサイズであるバイトやワードに合わせて、+1、+2 になります。1 ワードサイズ分、1 ロングワードサイズ分のデータのリード後は、リード開始時のアドレスに対して、SAT1、0 の設定に従ったアドレスの増減を行います。

ブロック転送モード（またはリピート転送モード）においてソースアドレス側にブロックエリア（リピートエリア）を設定している場合、ブロックサイズ（リピートサイズ）分の転送を終了すると、ソースアドレスは転送開始アドレスに戻りアドレス更新の影響を受けません。

ソースアドレス側に拡張リピートエリアを設定している場合は、その設定に従います。拡張リピートエリアに設定された上位側のアドレスは固定され、アドレス更新の影響を受けません。

転送中に DSAR をリードするときは、ロングワードサイズでアクセスしてください。転送中の DSAR は、CPU からのアクセスに関係なく更新される可能性があり、上位ワードと下位ワードを別々にリードすると正しい値がリードできない可能性があります。また、転送中のチャンネルの DSAR にライトしないでください。

(2) DMA デスティネーションアドレスレジスタ (DDAR)

転送先の DDAR のアドレスをアクセスすると、DDAR の値を出力し、次にアクセスするアドレスに更新されます。

DACR の DAT1、DAT0 ビットでアドレスの増減を設定します。DAT1、0=B'00 のときアドレスは固定されます。DAT1、0=B'01 のときアドレスはオフセットレジスタ値が加算されます。DAT1、0=B'10 のときアドレスは増加し、DAT1、0=B'11 のときアドレスは減少します（増減サイズは、転送するデータアクセスサイズにより決まります）。

DMDR の DTSZ1、DTSZ0 ビットでデータアクセスサイズを設定します。DTSZ1、0=B'00 のときバイトサイズになり、アドレスは±1 増減されます。DTSZ1、0=B'01 のときワードサイズになり、アドレスは±2 増減されます。DTSZ1、0=B'10 のときロングワードサイズになり、アドレスは±4 増減されます。また、ワードサイズ、ロングワードサイズの設定であっても、デスティネーションアドレスがワード境界、ロングワード境界を外れている場合、データはバイトやワードに分割してライトされます。1 ワードサイズ分、1 ロングワードサイズ分のデータを分割してライトしている間、アドレス増減サイズは実際にライトされるデータのサイズであるバイトやワードに合わせて、+1、+2 になります。1 ワードサイズ分、1 ロングワードサイズ分のデータのライト後は、ライト開始時のアドレスに対して、SAT1、0 の設定に従ったアドレスの増減を行います。

ブロック転送モード（またはリピート転送モード）においてデスティネーションアドレス側にブロックエリア（リピートエリア）を設定している場合、ブロックサイズ（リピートサイズ）分の転送を終了すると、デスティネーションアドレスは転送開始アドレスに戻りアドレス更新の影響を受けません。

デスティネーションアドレス側に拡張リピートエリアを設定している場合は、その設定に従います。拡張リピートエリアに設定された上位側のアドレスは固定され、アドレス更新の影響を受けません。

転送中に DDAR をリードするときは、ロングワードサイズでアクセスしてください。転送中の DDAR は、CPU からのアクセスに関係なく更新される可能性があり、上位ワードと下位ワードを別々にリードすると正しい値がリードできない可能性があります。また、転送中のチャンネルの DDAR にライトしないでください。

(3) DMA 転送カウントレジスタ (DTCR)

DMA 転送を行うと、DTCR の値は転送したバイト数分減少します。バイト転送した場合は-1、ワード転送した場合は-2、ロングワード転送した場合は-4 されます。ただし、DTCR の値が 0 のとき、転送回数はカウントされないため DTCR の値は変化しません。

DTCR の全ビットが変化する可能性があるため、DMA 転送中に CPU から DTCR をリードするときは、ロングワードサイズでアクセスしてください。転送中の DTCR は CPU からのアクセスに関係なく更新される可能性があり、上位ワードと下位ワードを別々にリードすると正しい値がリードできない可能性があります。また、動作中のチャンネルの DTCR にライトしないでください。

DMA 転送に伴うアドレスの更新と CPU によるライトが競合した場合は、CPU によるライトが優先されます。DTCR=1、2、4→0 への更新と CPU によるライト（値は 0 以外）が競合した場合は、DTCR の値は CPU によるライトが優先されますが、転送は終了します。

(4) DMA ブロックサイズレジスタ (DBSR)

DBSRはブロック転送モード、またはリピート転送モードのときに有効となります。DBSR[31:16]はBKSZH、DBSR[15:0]はBKSZとして機能します。BKSZHの16ビットはブロックサイズやリピートサイズを保持し、その値は変化しません。BKSZの16ビットはブロックサイズやリピートサイズのカウンタとして機能し、1データ転送毎に-1します。DMA転送によってBKSZの値が0になると判断されると、DMACはBKSZに0をストアせず、BKSZHの値をストアします。

DBSRは上位16ビットが更新されることがないので、ワードサイズでアクセスすることができます。動作中のチャンネルのDBSRにライトしないでください。

(5) DMDR の DTE ビット

DMDRのDTEビットは、CPUによりライトしてデータ転送の許可/禁止を制御しますが、DMA転送状態によってDMACにより自動的にDTEビットを0にクリアすることがあります。

DMACによりDTEビットが0にクリアされる条件は以下です。

- 設定の総転送サイズの転送が終了したとき
- 転送サイズエラー割り込み要求が発生し、転送が終了したとき
- リピートサイズ終了割り込み要求が発生し、転送が終了したとき
- 拡張リピートエリアオーバフロー割り込み要求が発生し、転送が終了したとき
- NMI割り込みが発生し、転送が停止したとき
- アドレスエラーが発生し、転送が停止したとき
- リセット時
- ハードウェアスタンバイモード*時
- DTEビットに0をライトして、転送が停止したとき

DTEビットが1にセットされているチャンネルのレジスタのライトは禁止されています(DTEビットを除く)。DTEビットに0をライトしてから各レジスタの設定を変更するときは、DTEビットが0にクリアされていることを確認してください。

【注】 * 本LSIには、ハードウェアスタンバイモードはありません。

図 7.21 に、動作中のチャンネルのレジスタ設定を変更する手順を示します。

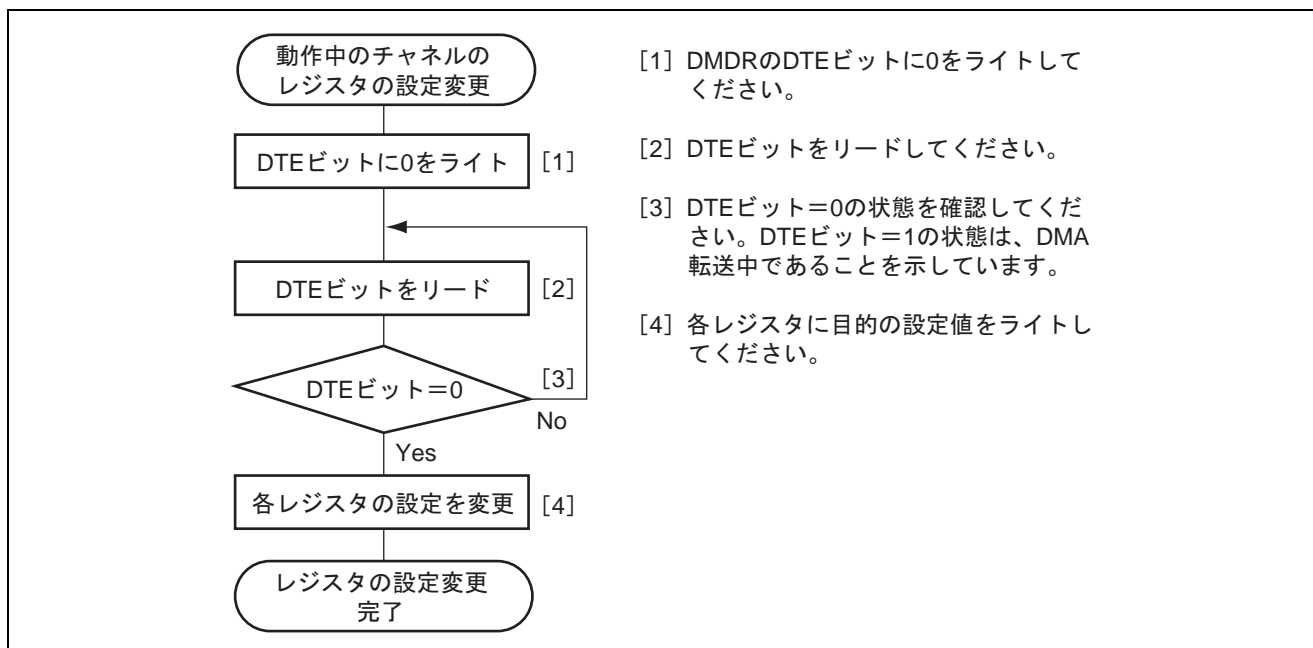


図 7.21 動作中のチャンネルのレジスタ設定を変更するときの手順

(6) DMDR の ACT ビット

DMDR の ACT ビットは、DMAC が待機状態かアクティブ状態かを示します。DTE=0 の場合と、DTE=1 で転送要求待ち状態の場合に ACT=0 となり、それ以外の場合（DMAC がアクティブ状態）には ACT=1 になります。DTE ビットに 0 をライトして転送を停止させても DMA 転送中であれば、ACT ビットは 1 を保持します。

ブロック転送モードの場合、DTE ビットに 0 をライトして転送を停止させても 1 ブロックサイズの転送は中断されません。DTE ビットに 0 をライトしてから 1 ブロックサイズの転送が終了するまでの期間、ACT ビットは 1 を保持します。

バーストモードの場合、DTE ビットに 0 をライトしたバスサイクルから最大 3 回の DMA 転送が行われてから転送を停止します。DTE ビットに 0 をライトしてから最後の DMA サイクルが終了するまでの期間、ACT ビットは 1 を保持します。

(7) DMDR の ERRF ビット

アドレスエラー、または NMI 割り込みが発生した場合、DMAC は全チャンネルの DTE ビットを 0 にクリアして転送を終了します。また、DMAC は転送中か否かに関わらず、DMDR_0 の ERRF ビットを 1 にセットし、アドレスエラー、または NMI 割り込みが発生したことを示します。ただし、DMAC がモジュールストップ状態にある場合、アドレスエラーや NMI 割り込みに対しては、ERRF ビットを 1 にセットしません。

(8) DMDR の ESIF ビット

転送サイズ割り込み、リピートサイズ終了割り込み、拡張リピートエリアオーバーフロー割り込み要求が発生したとき、DMDR の ESIF ビットは 1 にセットされます。ESIF ビットが 1 にセットされ、DMDR の ESIE ビットが 1 にセットされていると転送エスケープ割り込み要求を CPU や DTC に対して発生します。

ESIF ビットに 1 がセットされるタイミングは、割り込み要求を発生させる要因になった DMA 転送のバスサイクルが終了して、DMDR の ACT ビットが 0 になって転送を終了したときです。

割り込み処理中に DTE ビットを 1 にセットして転送を再開した場合、自動的に ESIF ビットが 0 にクリアされ、割り込み要求は解除されます。

割り込みについては「7.7 割り込み要因」を参照してください。

(9) DMDR の DTIF ビット

DMA 転送によって総転送サイズ分の転送を終了すると、DMDR の DTIF ビットは 1 にセットされます。DTIF ビットが 1 にセットされ、DMDR の DTIE ビットが 1 にセットされていると転送カウンタによる転送終了割り込み要求を CPU や DTC に対して発生します。

DTIF ビットが 1 にセットされるタイミングは、DMA 転送のバスサイクルが終了して、DMDR の ACT ビットが 0 になって転送を終了したときです。

割り込み処理中に DTE ビットに 1 をセットして転送を再開した場合、自動的に DTIF ビットが 0 クリアされ、割り込み要求は解除されます。

割り込みについては「7.7 割り込み要因」を参照してください。

7.4.8 チャンネルの優先順位

DMACのチャンネル間の優先順位は、チャンネル0>チャンネル1>チャンネル2>チャンネル3の順になっています。表7.5にDMACのチャンネル間の優先順位を示します。

表 7.5 DMAC のチャンネル間の優先順位

チャンネル	優先順位
チャンネル0	高 ↑ 低
チャンネル1	
チャンネル2	
チャンネル3	

転送中に他のチャンネルからの転送要求があると、転送中のチャンネルを除いて最も優先度の高いチャンネルが選択されます。選択されたチャンネルは、転送中のチャンネルがバスを解放後に転送を開始します。このとき、DMAC以外の他のバスマスタからバス権要求があると、他のバスマスタのサイクルが入ります。

バースト転送中、および1ブロック分のブロック転送中は、チャンネルを切り換えて転送することはありません。

図 7.22 に、チャンネル0~2からの転送要求が同時に発生したときの転送例を示します。

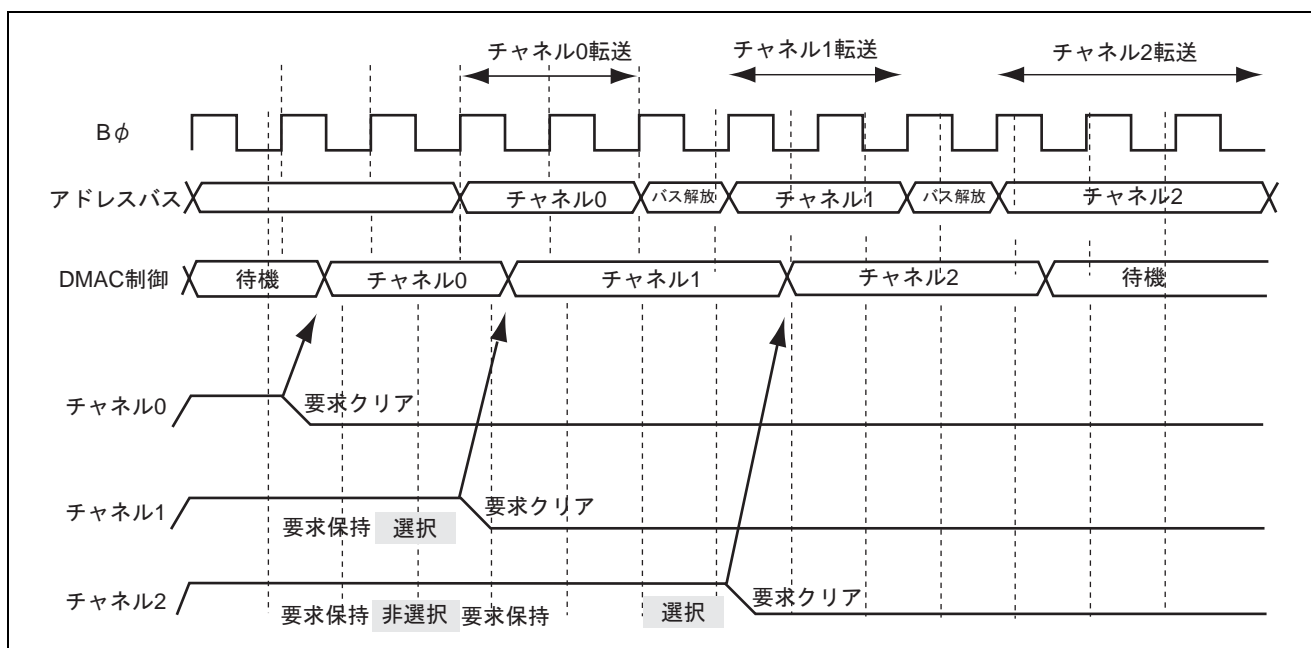


図 7.22 チャンネルの優先順位のタイミング例

7.4.9 基本バスサイクル

基本的なバスサイクルのタイミング例を図 7.23 に示します。図 7.23 は、ワードサイズで 16 ビット 2 ステートアクセス空間から 8 ビット 3 ステートアクセス空間へ転送する例です。CPU から DMAC にバス権が移ると、ソースアドレスのリード、デスティネーションアドレスのライトを行います。このリード/ライト動作の間に、他のバス権要求などによってバスを解放することはありません。DMAC サイクルは CPU サイクルと同様にバスコントローラの設定に従います。

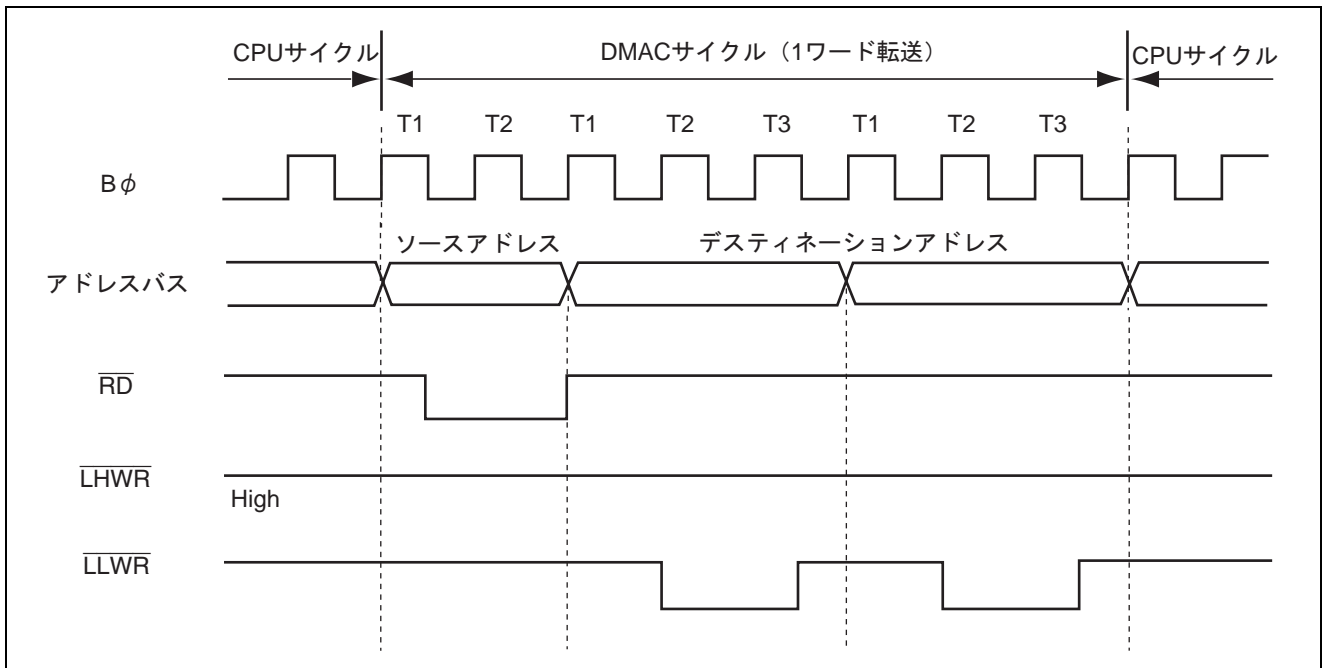


図 7.23 DMA 転送バスタイミング例

7.4.10 デュアルアドレスモードのバスサイクル

(1) ノーマル転送モード (サイクルスチールモード)

サイクルスチールモードでは、1回の転送単位 (1 バイト、1 ワードまたは 1 ロングワード) の転送を終了するたびにバスを解放します。バス解放期間中は CPU、または DTC によるバスサイクルが 1 回以上入ります。

図 7.24 に、 $\overline{\text{TEND}}$ 出力を許可して、外部 16 ビット 2 ステートアクセス空間から外部 16 ビット 2 ステートアクセス空間へワードサイズでノーマル転送モードかつサイクルスチールモードで転送を行った場合の例を示します。

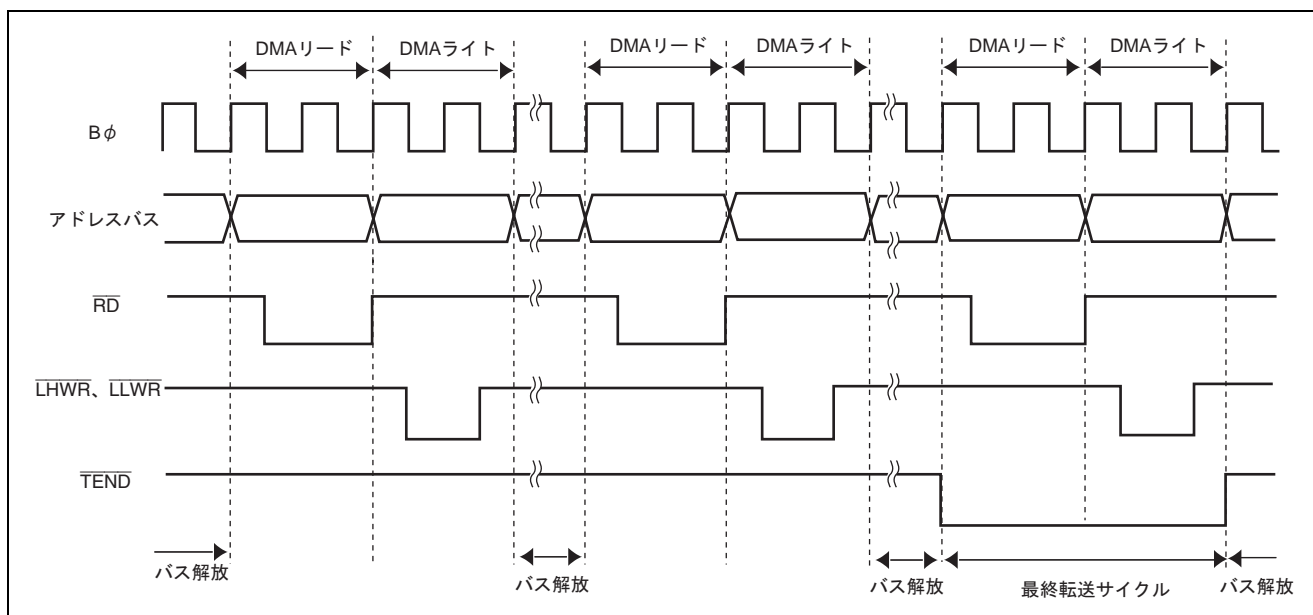


図 7.24 ノーマル転送モードかつサイクルスチールモードの転送例

図 7.25、図 7.26 に、 $\overline{\text{TEND}}$ 出力を許可して、外部 16 ビット 2 ステートアクセス空間から 16 ビット 2 ステートアクセス空間へロングワードサイズでノーマル転送モードかつサイクルスチールモードで転送を行った場合の例を示します。

図 7.25 では、転送元は (DSAR=ロングワード境界を外れたアドレス)、転送先は (DDAR=ロングワード境界に沿ったアドレス) です。

図 7.26 では、転送元は (DSAR=ロングワード境界に沿ったアドレス)、転送先は (DDAR=ロングワード境界を外れたアドレス) です。

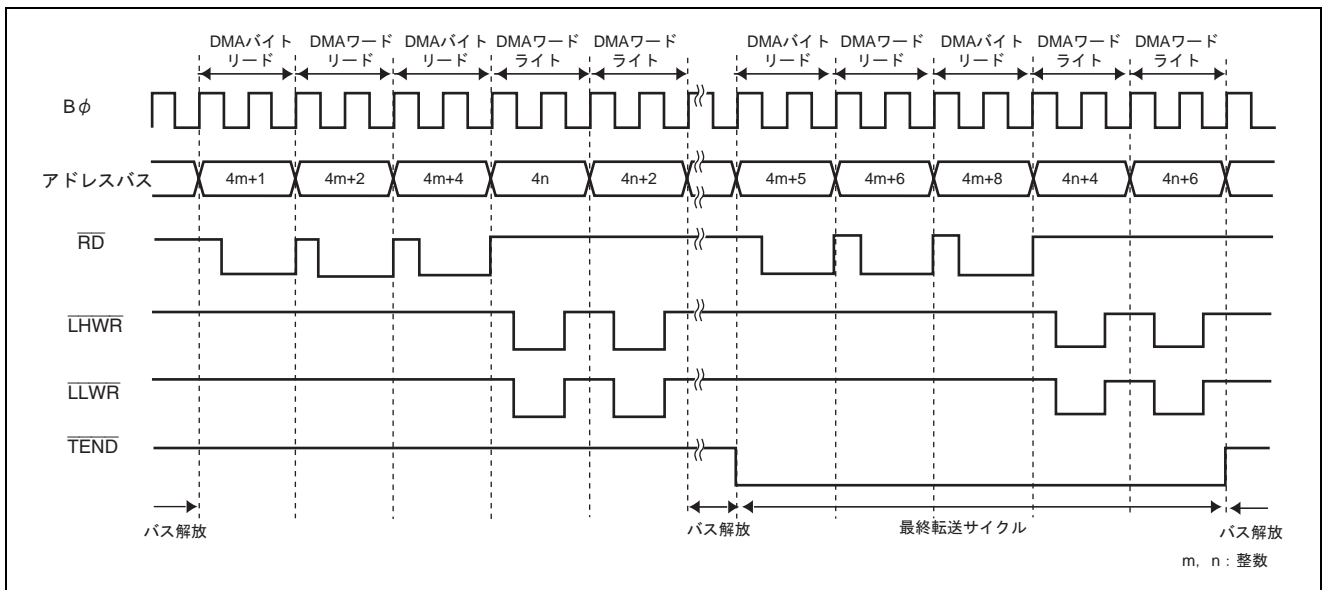


図 7.25 ノーマル転送モードかつサイクルスチールモードの転送例
(転送元 DSAR=奇数アドレス、ソースアドレス増加)

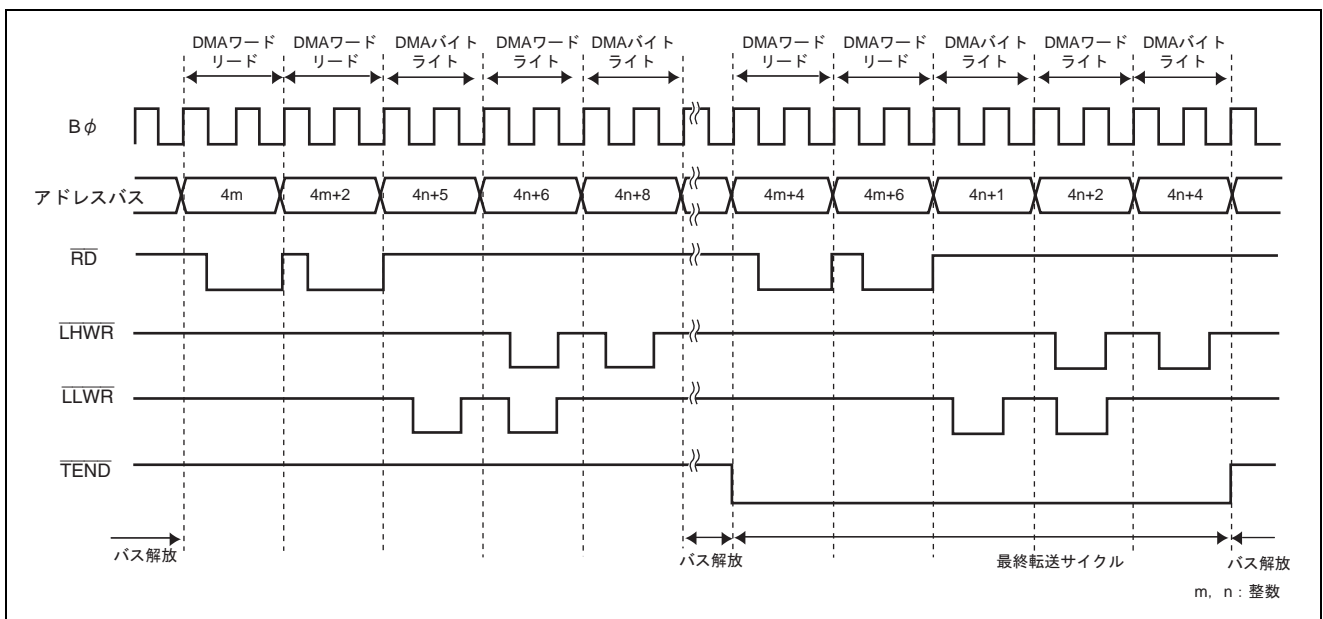


図 7.26 ノーマル転送モードかつサイクルスチールモードの転送例
(転送先 DDAR=奇数アドレス、デスティネーションアドレス減少)

(2) ノーマル転送モード (バーストモード)

バーストモードでは、1バイト、1ワードまたは1ロングワードの転送を転送終了条件が満たされるまで続けます。

バースト転送を開始すると、優先順位の高い他のチャネルの転送要求が発生してもバースト転送が終了するまで待たされます。

図 7.27 に、 $\overline{\text{TEND}}$ 出力を許可して、外部 16 ビット 2 ステートアクセス空間から外部 16 ビット 2 ステートアクセス空間へワードサイズでノーマル転送モードかつバーストモードで転送を行った場合の例を示します。

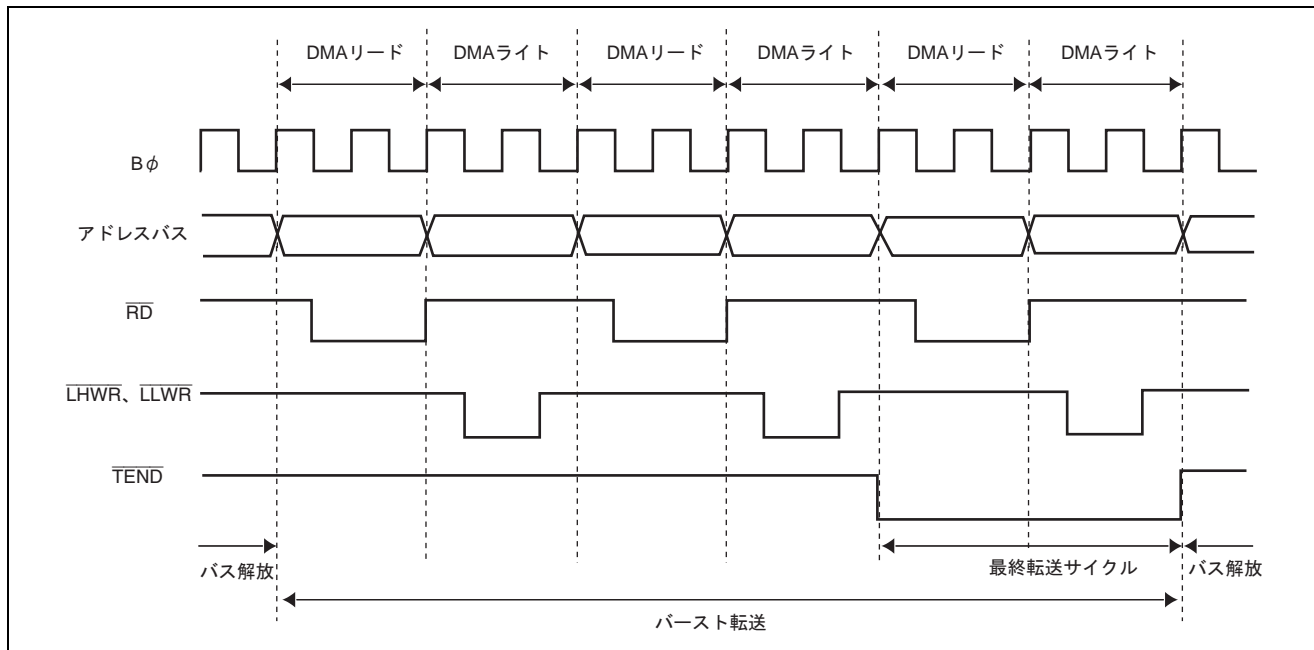


図 7.27 ノーマル転送モードかつバーストモードの転送例

(3) ブロック転送モード

ブロック転送モードでは、1回の転送要求について1ブロック分の転送を終了するたびにバスを解放します。

図 7.28 に、 $\overline{\text{TEND}}$ 出力を許可して、外部 16 ビット 2 ステートアクセス空間から外部 16 ビット 2 ステートアクセス空間へワードサイズでブロック転送モードで転送を行った場合の例を示します。

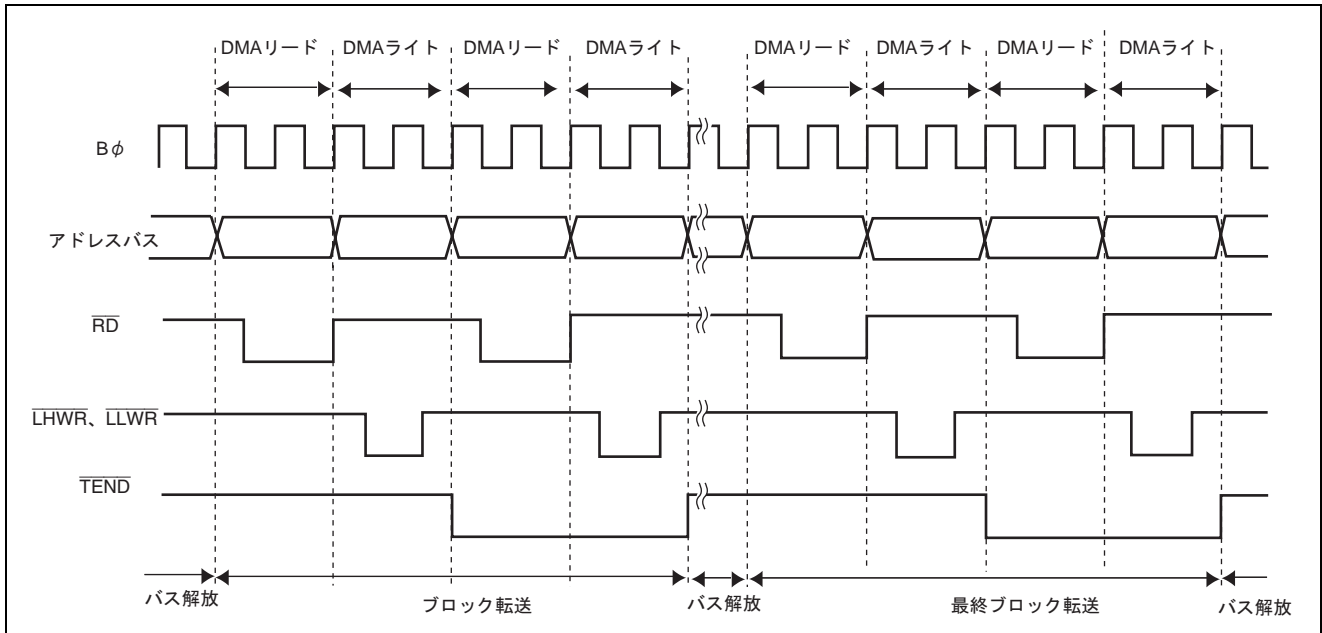


図 7.28 ブロック転送モードの転送例

(4) $\overline{\text{DREQ}}$ 立ち下がリエッジ起動タイミング

図 7.29 に $\overline{\text{DREQ}}$ 立ち下がリエッジ起動時のノーマル転送モードの転送例を示します。

$\overline{\text{DREQ}}$ のサンプリングは、DTE ビットのライトサイクル終了直後の次の B ϕ の立ち上がりを起点に毎サイクル行われます。

$\overline{\text{DREQ}}$ による転送要求の受け付けが可能な状態で $\overline{\text{DREQ}}$ の Low レベルがサンプリングされると、DMAC 内部で転送要求が保持されます。次に DMAC 内部で起動がかかると転送要求はクリアされ、エッジ検出のため $\overline{\text{DREQ}}$ の High レベルのサンプリングが開始されます。DMA ライトサイクル終了までに $\overline{\text{DREQ}}$ の High レベルのサンプリングが済んでいればライトサイクル終了後に転送要求の受け付けが再開され、再び $\overline{\text{DREQ}}$ の Low レベルをサンプリングして転送終了までこの動作を繰り返します。

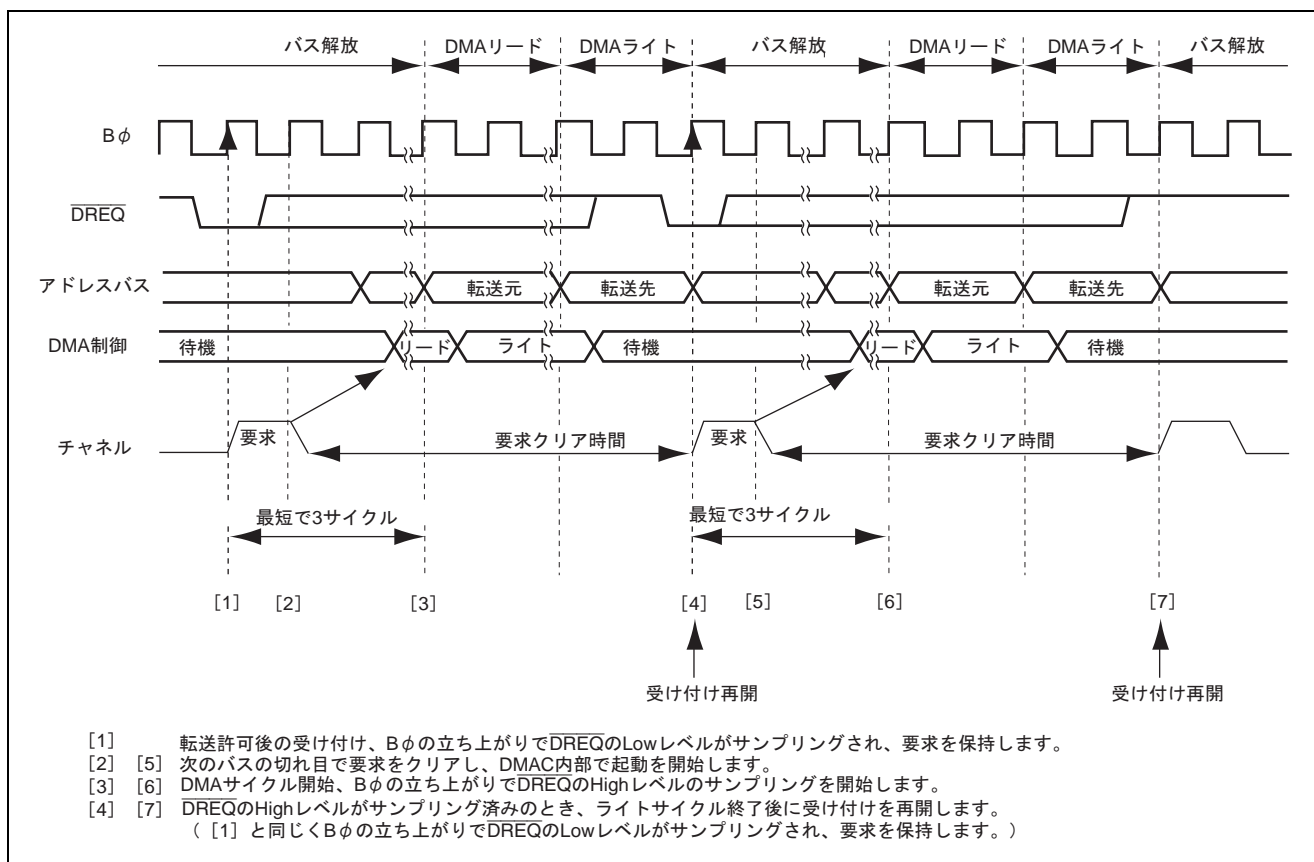


図 7.29 $\overline{\text{DREQ}}$ 立ち下がリエッジ起動時のノーマル転送モードの転送例

図 7.30 に $\overline{\text{DREQ}}$ 立ち下がりエッジ起動時のブロック転送モードの転送例を示します。

$\overline{\text{DREQ}}$ のサンプリングは、DTE ビットのライトサイクル終了直後の次の $\text{B}\phi$ の立ち上がりを起点に毎サイクル行われます。

$\overline{\text{DREQ}}$ による転送要求の受け付けが可能な状態で $\overline{\text{DREQ}}$ の Low レベルがサンプリングされると、DMAC 内部で転送要求が保持されます。次に DMAC 内部で起動がかかると転送要求はクリアされ、エッジ検出のため $\overline{\text{DREQ}}$ の High レベルのサンプリングが開始されます。DMA ライトサイクル終了までに $\overline{\text{DREQ}}$ の High レベルのサンプリングが済んでいればライトサイクル終了後に転送要求の受け付けが再開され、再び $\overline{\text{DREQ}}$ の Low レベルをサンプリングして転送終了までこの動作を繰り返します。

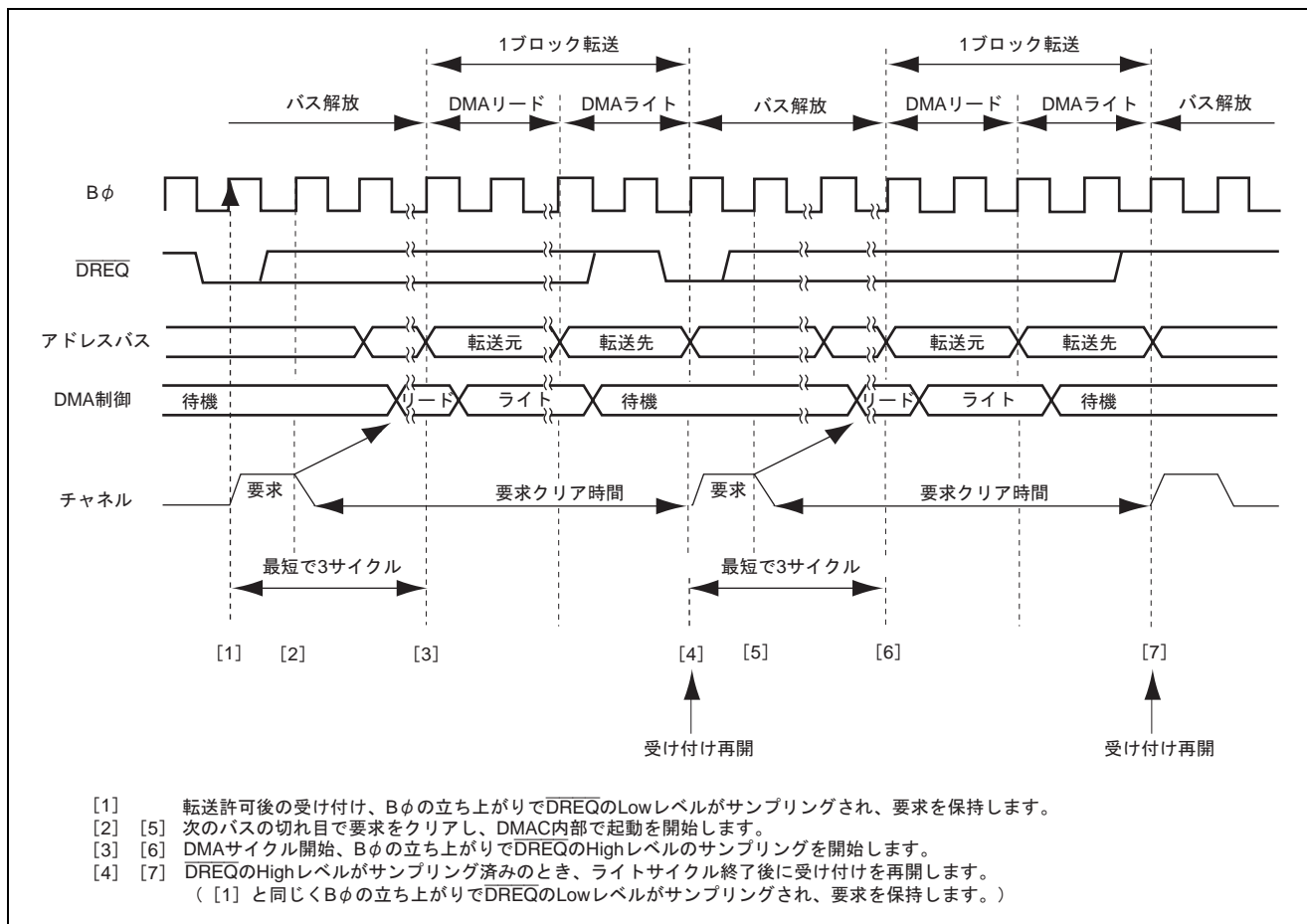


図 7.30 $\overline{\text{DREQ}}$ 立ち下がりエッジ起動時のブロック転送モードの転送例

(5) $\overline{\text{DREQ}}$ レベル起動タイミング

図 7.31 に $\overline{\text{DREQ}}$ レベル起動時のノーマル転送モードの転送例を示します。

$\overline{\text{DREQ}}$ のサンプリングは、DTE ビットのライトサイクル終了直後の次の $\text{B}\phi$ の立ち上がりを起点に毎サイクル行われます。

$\overline{\text{DREQ}}$ による転送要求の受け付けが可能な状態で $\overline{\text{DREQ}}$ の Low レベルがサンプリングされると、DMAC 内部で転送要求が保持されます。次に DMAC 内部で起動がかかると転送要求はクリアされます。ライトサイクル終了後に転送要求の受け付けが再開され、再び $\overline{\text{DREQ}}$ 端子の Low レベルをサンプリングして転送終了までこの動作を繰り返します。

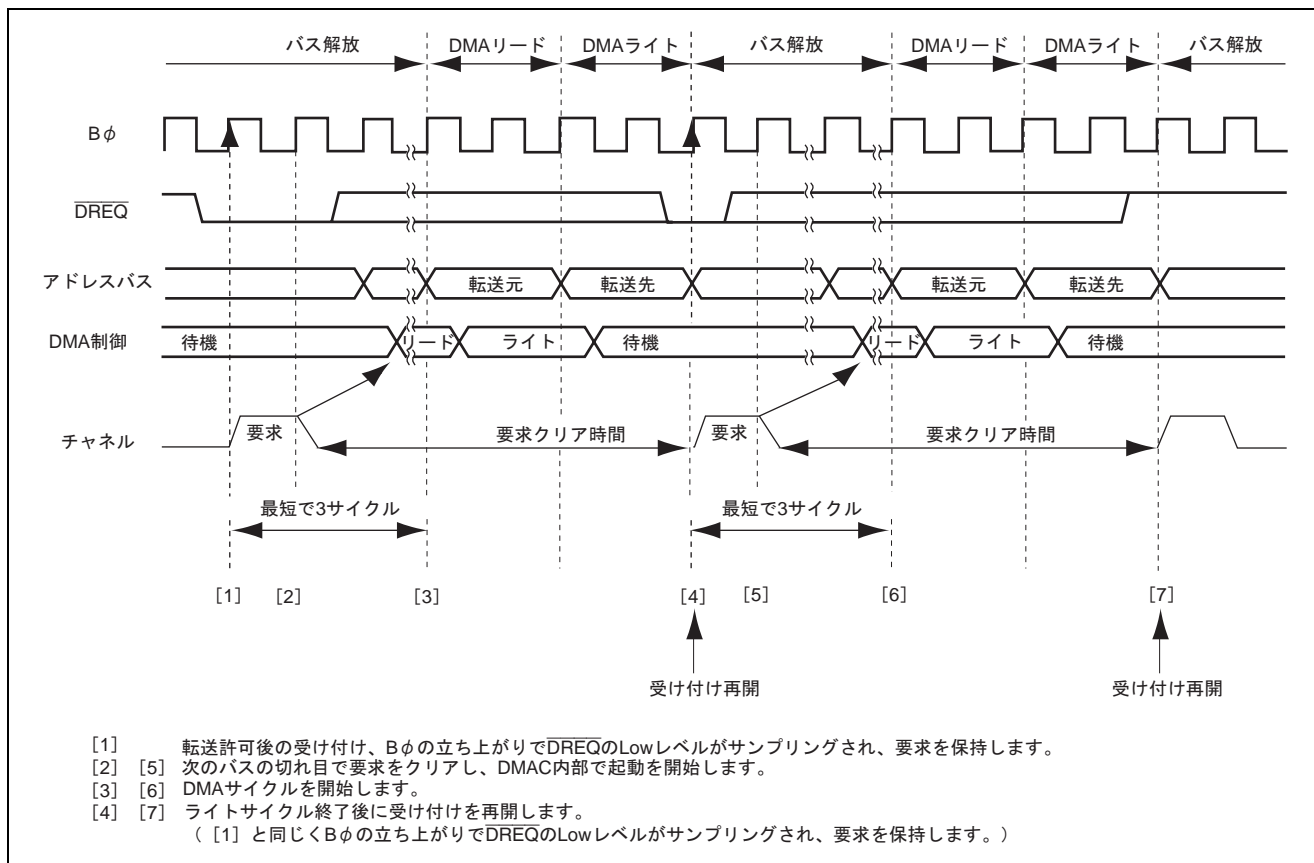


図 7.31 $\overline{\text{DREQ}}$ レベル起動時のノーマル転送モードの転送例

図 7.32 に \overline{DREQ} レベル起動時のブロック転送モードの転送例を示します。

\overline{DREQ} のサンプリングは、DTE ビットのライトサイクル終了直後の次の $B\phi$ の立ち上がりを起点に毎サイクル行われます。

\overline{DREQ} による転送要求の受け付けが可能な状態で \overline{DREQ} の Low レベルがサンプリングされると、DMAC 内部で転送要求が保持されます。次に DMAC 内部で起動がかかると転送要求はクリアされます。ライトサイクル終了後に転送要求の受け付けが再開され、再び \overline{DREQ} の Low レベルをサンプリングして転送終了までこの動作を繰り返します。

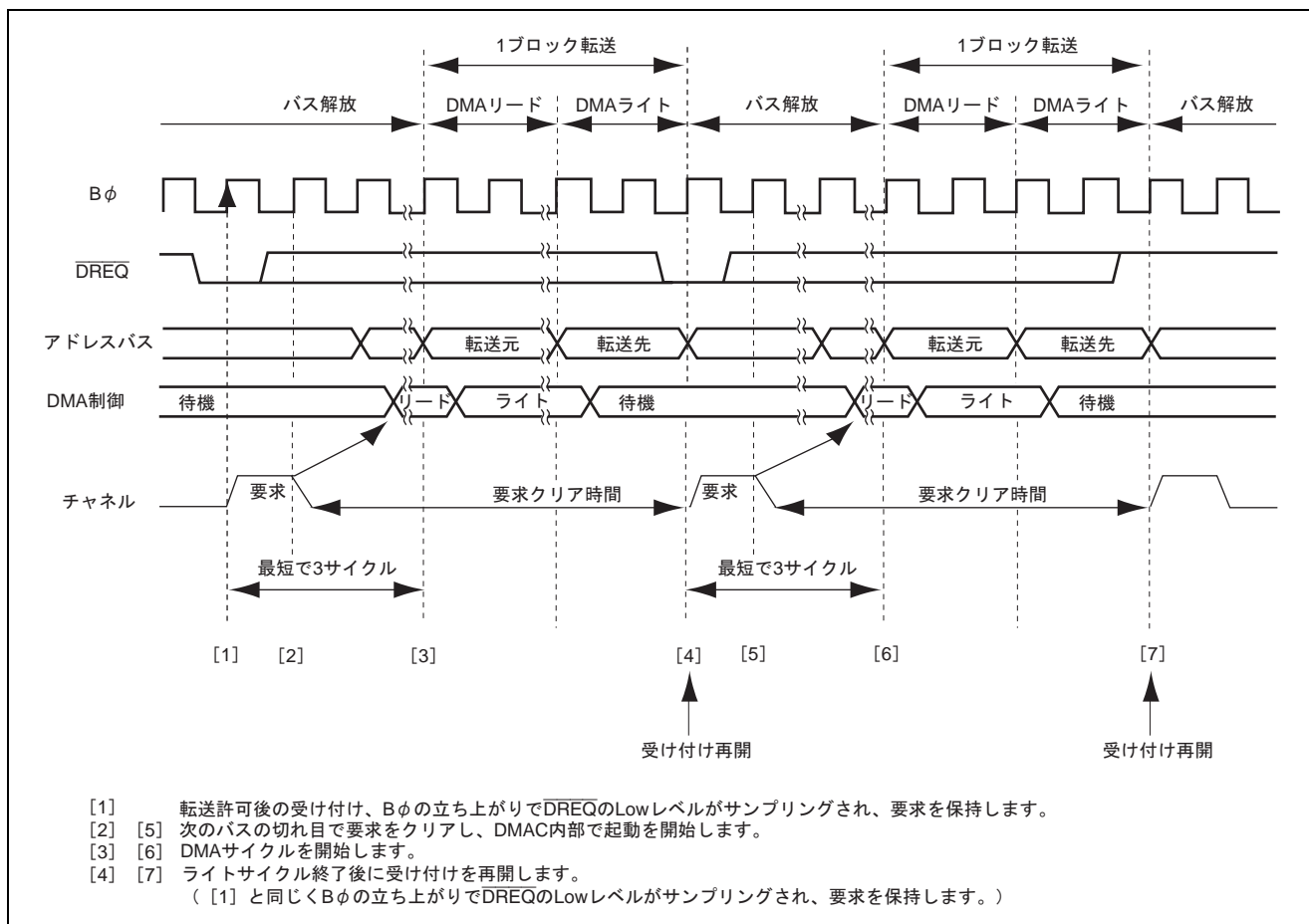


図 7.32 \overline{DREQ} レベル起動時のブロック転送モードの転送例

(6) NRD=1 のときの $\overline{\text{DREQ}}$ Low レベル起動タイミング

DMDR の NRD ビットを 1 にセットすると、次の転送要求の受け付けタイミングを 1 サイクル遅延させることができます。

図 7.33 に NRD=1 のときの $\overline{\text{DREQ}}$ Low レベル起動時のノーマル転送モードの転送例を示します。

$\overline{\text{DREQ}}$ のサンプリングは、DTE ビットのライトサイクル終了直後の次の B ϕ の立ち上がりを起点に毎サイクル行われます。

$\overline{\text{DREQ}}$ による転送要求の受け付けが可能な状態で $\overline{\text{DREQ}}$ の Low レベルがサンプリングされると、DMAC 内部で要求が保持されます。次に DMAC 内部で起動がかかると転送要求はクリアされます。ライトサイクル終了後に NRD=1 による 1 サイクルの要求クリア期間後に受け付けが再開され、再び $\overline{\text{DREQ}}$ の Low レベルをサンプリングして転送終了までこの動作を繰り返します。

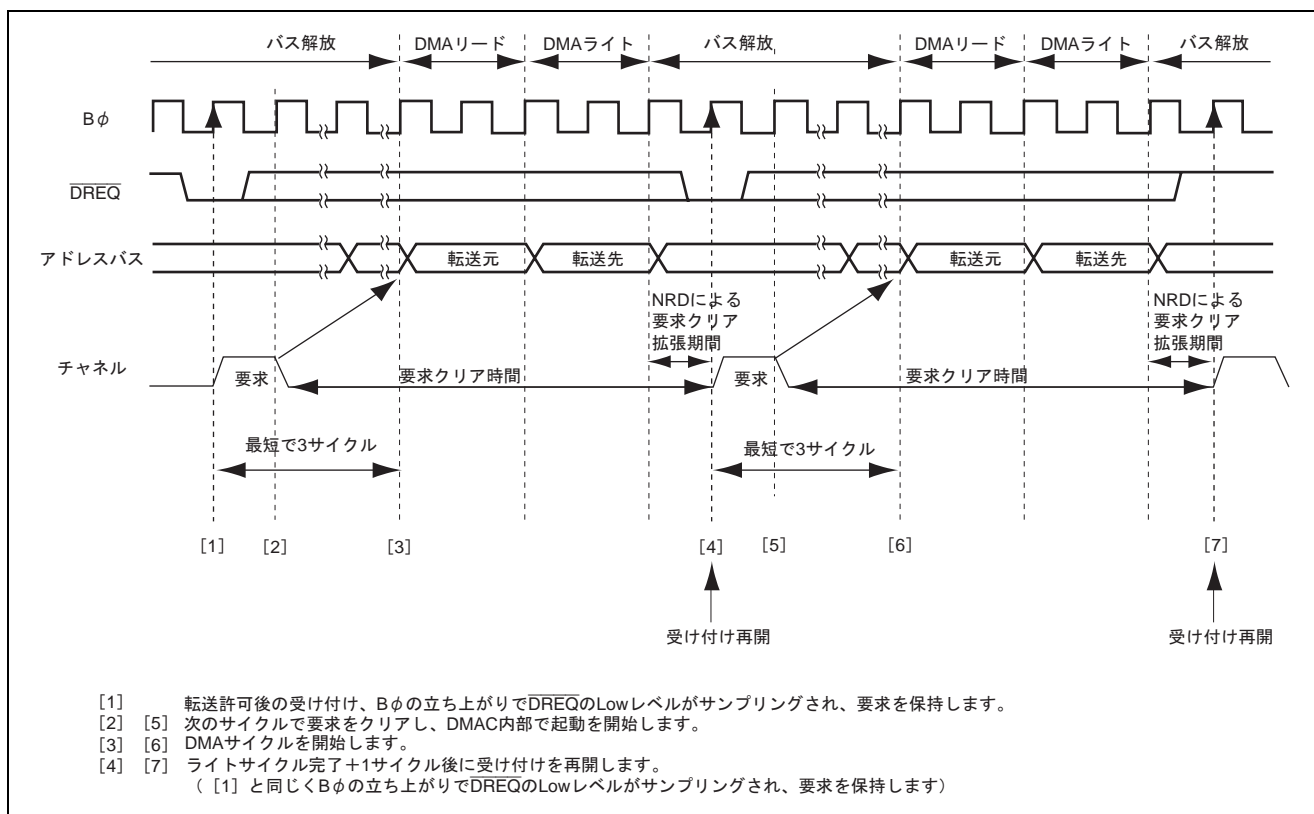


図 7.33 NRD=1 のときの $\overline{\text{DREQ}}$ Low レベル起動時のノーマル転送モードの転送例

7.4.11 シングルアドレスモードのバスサイクル

(1) シングルアドレスモード (リード、サイクルスチールモード)

シングルアドレスモードでは、1回の転送要求について1バイト、1ワード、または1ロングワードの転送を行い、転送終了後にいったんバスを解放します。バス解放期間中はCPUまたはDTCによるバスサイクルが1回以上入ります。

図 7.34 に、 \overline{TEND} 出力を許可して、外部 8 ビット 2 ステートアクセス空間から外部デバイスへバイトサイズでシングルアドレスモードで転送 (リード) を行った場合の例を示します。

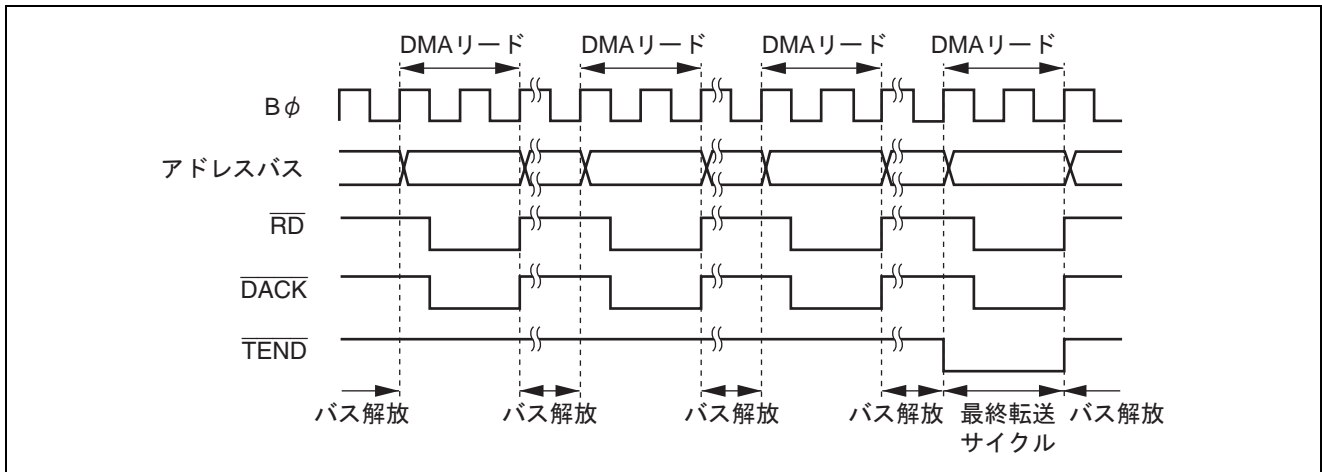


図 7.34 シングルアドレスモード (バイトリード) の転送例

(2) シングルアドレスモード (ライト、サイクルスチールモード)

シングルアドレスモードでは、1回の転送要求について1バイト、1ワード、または1ロングワードの転送を行い、転送終了後にいったんバスを解放します。バス解放期間中はCPU、またはDTCによるバスサイクルが1回以上入ります。

図 7.35 に、 $\overline{\text{TEND}}$ 出力を許可して、外部デバイスから外部 8 ビット 2 ステートアクセス空間へバイトサイズでシングルアドレスモードで転送 (ライト) を行った場合の例を示します。

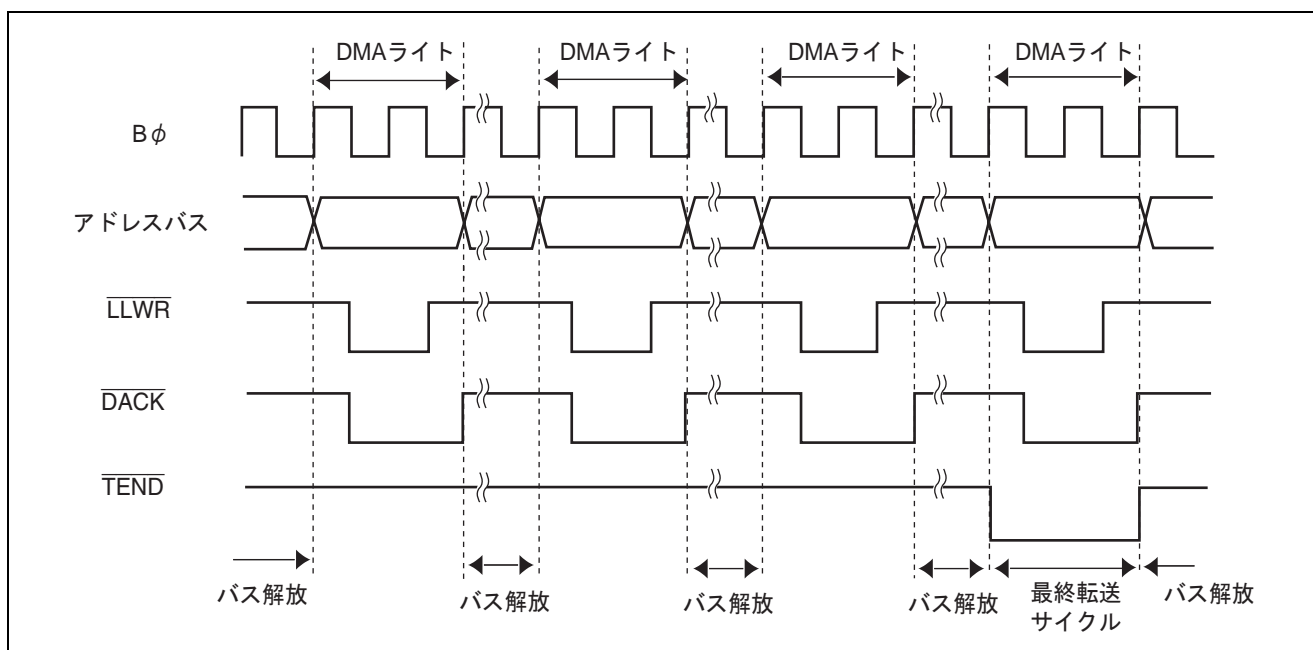


図 7.35 シングルアドレスモード (バイトライト) の転送例

(3) \overline{DREQ} 立ち下がリエッジ起動タイミング

図 7.36 に \overline{DREQ} 立ち下がリエッジ起動時のシングルアドレスモードの転送例を示します。

\overline{DREQ} のサンプリングは、DTE ビットのライトサイクル終了直後の次の Bφ の立ち上がりを起点に毎サイクル行われます。

\overline{DREQ} による転送要求の受け付けが可能な状態で \overline{DREQ} の Low レベルがサンプリングされると、DMAC 内部で要求が保持されます。次に DMAC 内部で起動がかかると転送要求はクリアされ、エッジ検出のために \overline{DREQ} の High レベルのサンプリングが開始されます。DMA シングルサイクル終了までに \overline{DREQ} の High レベルのサンプリングが済んでいれば、シングルサイクル終了後に転送要求の受け付けが再開され、再び \overline{DREQ} の Low レベルをサンプリングして転送終了までこの動作を繰り返します。

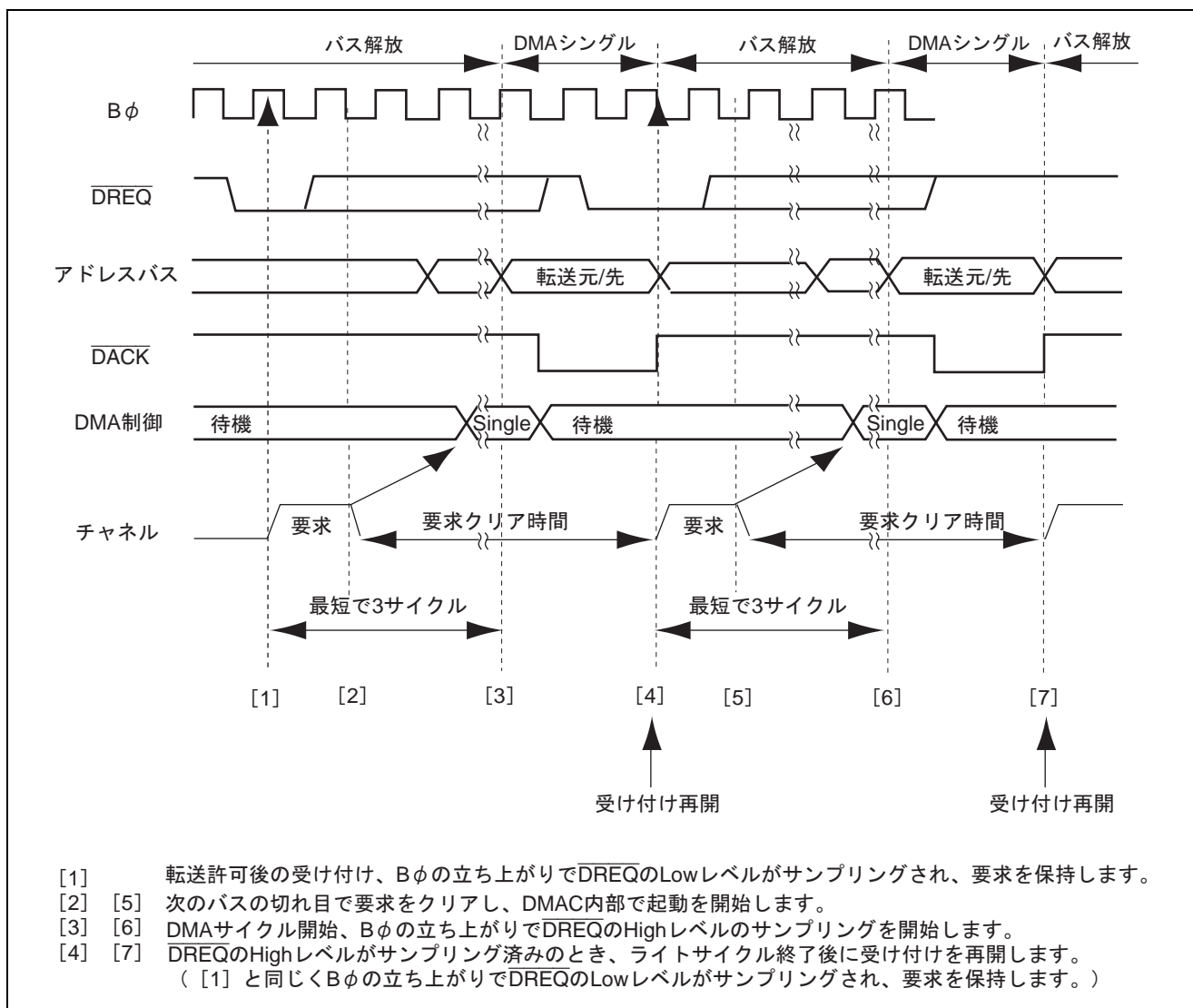


図 7.36 \overline{DREQ} 立ち下がリエッジ起動時のシングルアドレスモードの転送例

(4) $\overline{\text{DREQ}}$ Low レベル起動タイミング

図 7.37 に $\overline{\text{DREQ}}$ Low レベル起動時のシングルアドレスモードの転送例を示します。

$\overline{\text{DREQ}}$ のサンプリングは、DTE ビットのライトサイクル終了直後の次の B ϕ の立ち上がりを起点に毎サイクル行われます。

$\overline{\text{DREQ}}$ による転送要求の受け付けが可能な状態で $\overline{\text{DREQ}}$ の Low レベルがサンプリングされると、DMAC 内部で転送要求が保持されます。次に DMAC 内部で起動がかかると転送要求はクリアされます。シングルサイクル終了後に転送要求の受け付けが再開され、再び $\overline{\text{DREQ}}$ 端子の Low レベルをサンプリングして転送終了までこの動作を繰り返します。

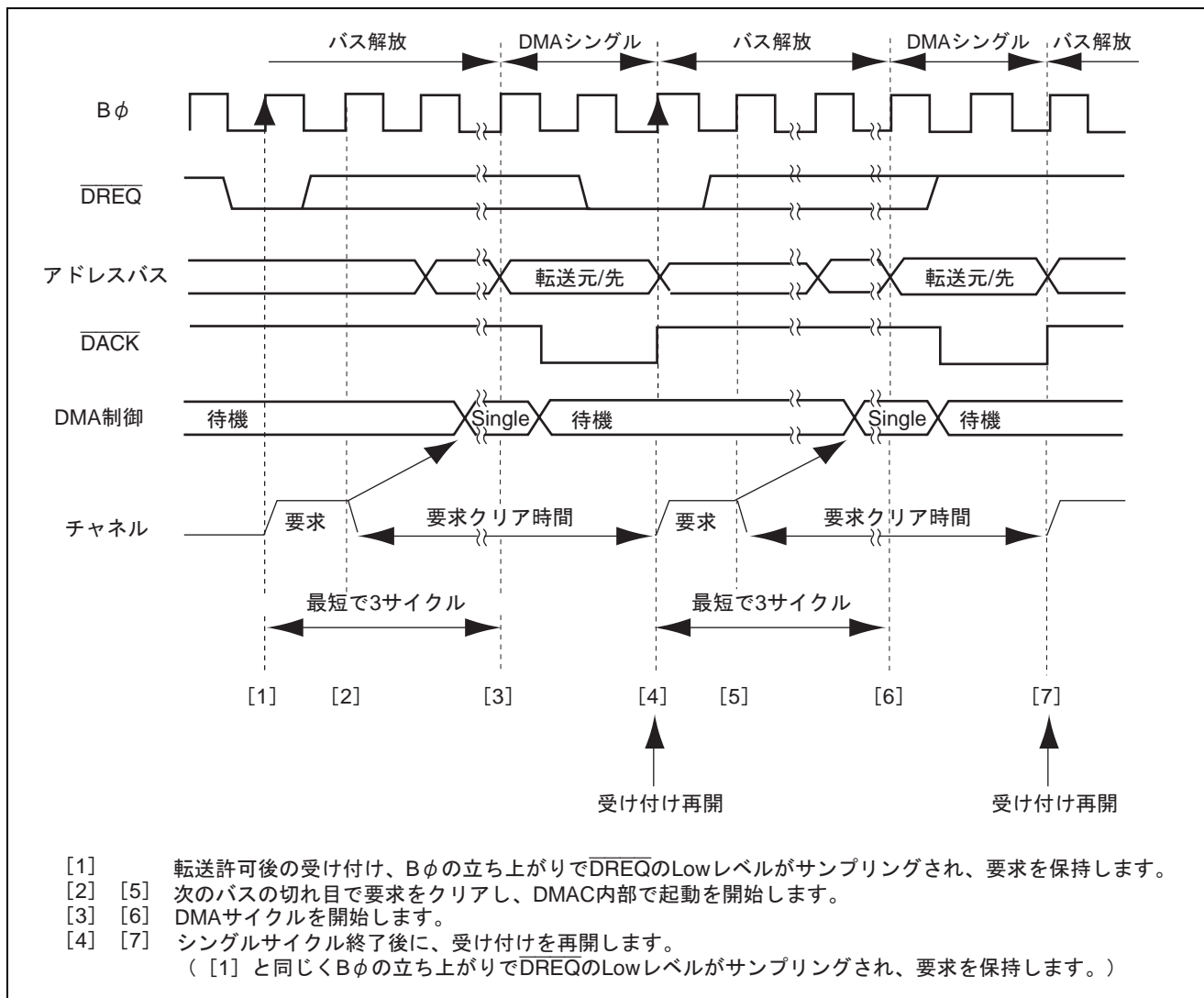


図 7.37 $\overline{\text{DREQ}}$ Low レベル起動時のシングルアドレスモードの転送例

7.5 DMA 転送終了

DMA 転送終了は、転送終了条件によって動作が異なります。DMA 転送が終了すると、DMDR の DTE ビットと ACT ビットが 1 から 0 になり、DMA 転送が終了したことを示します。

(1) DTCR=1、2、4→0 による転送終了

DTCR の値が 1、2、または 4 から 0 になると対応するチャンネルの DMA 転送が終了し、DMDR の DTE ビットが 0 にクリアされ、同時に DMDR の DTIF ビットが 1 にセットされます。このとき DMDR の DTIE ビットが 1 にセットされていると、転送カウンタによる転送終了割り込み要求が発生します。DTCR の値が転送前から 0 の場合は、DMA 転送は終了しません。

(2) 転送サイズエラー割り込みによる転送終了

DMDR の TSEIE ビットが 1 にセットされているときに次の条件を満たすと、転送サイズエラー割り込み要求が発生し、DMA 転送を終了します。このとき DMDR の DTE ビットは 0 にクリアされ、同時に DMDR の ESIF ビットが 1 にセットされます。

- ノーマル転送モード、リピート転送モードにおいて、データアクセスサイズよりも DTCR の値が小さくなり、1 要求あたりのデータアクセスサイズ分の転送ができなくなった状態で、次の転送要求が発生したとき
- ブロック転送モードにおいて、ブロックサイズよりも DTCR の値が小さくなり、1 要求あたりのブロックサイズ分の転送ができなくなった状態で、次の転送要求が発生したとき

DMDR の TSEIE ビットが 0 にクリアされているときは、DTCR が 0 になるまで転送を行い、転送サイズエラー割り込み要求は発生しません。各転送モードごとの動作を以下に示します。

- ノーマル転送モードやリピート転送モードにおいて、データアクセスサイズよりも DTCR の値が小さいと、データアクセスサイズをバイトに固定して転送を行います。
- ブロック転送モードにおいて、ブロックサイズよりも DTCR の値が小さいと、ブロックサイズ分の転送を行わず DTCR 分の転送を行います。データアクセスサイズよりも DTCR の値が小さいときは、データアクセスサイズをバイトに固定して転送を行います。

(3) リピートサイズ終了割り込みによる転送終了

リピート転送モードにおいて、DACR の RPTIE ビットが 1 にセットされているときに、1 リピートサイズ分の転送終了後に次の転送要求が発生すると、リピートサイズ終了割り込み要求が発生します。割り込み要求の発生により DMA 転送を終了し、DMDR の DTE ビットが 0 にクリアされ、同時に DMDR の ESIF ビットが 1 にセットされます。この状態から DTE ビットを 1 にセットすると転送を再開させることができます。

ブロック転送モードにおいても、リピートサイズ終了割り込み要求を発生させることができます。ブロック転送モードでは、1 ブロック分の転送終了後に次の転送要求が発生すると、リピートサイズ終了割り込み要求が発生します。

(4) 拡張リピートエリアオーバフロー割り込みによる転送終了

拡張リピートエリアを指定し、DACR の SARIE ビットまたは DARIE ビットが 1 にセットされているときに、アドレスの拡張リピートエリアがオーバフローすると、拡張リピートエリアオーバフロー割り込み要求が発生します。割り込み要求の発生により DMA 転送は終了し、DMDR の DTE ビットが 0 にクリアされ、同時に DMDR の ESIF ビットが 1 にセットされます。

デュアルアドレスモードでは、リードサイクル中に拡張リピートエリアオーバフロー割り込み要求が発生しても続くライトサイクル処理は実行されます。

ブロック転送モードでは、1 ブロック分の転送中に拡張リピートエリアオーバフロー割り込み要求が発生しても 1 ブロック分の転送は実行されます。拡張リピートエリアオーバフロー割り込みによる転送終了は、ブロックサイズの区切りで発生します。

(5) DMDR の DTE ビットに 0 をライトして転送終了

CPU 等によって DMDR の DTE ビットに 0 をライトすると、転送中の DMA サイクルおよび転送要求を受け付けた DMA サイクルの実行後に転送終了されます。

ブロック転送モードでは、転送中の 1 ブロックサイズの転送終了後に DMA 転送を終了します。

(6) NMI 割り込みによる転送終了

NMI 割り込みが発生した場合、DMAC は全チャンネルの DTE ビットを 0 にクリアし、かつ DMDR_0 の ERRF ビットを 1 にセットします。DMA 転送中に NMI 割り込みが発生した場合、転送は強制的に終了します。NMI 割り込み発生後に DMA 転送を行う場合は、ERRF ビットを 0 にクリアした後に各チャンネルの DTE ビットを 1 にセットしてください。

以下に、NMI 割り込み検出後の転送終了タイミングを各モード毎に示します。

(a) ノーマル転送モード、リピート転送モード

デュアルアドレスモードでは、1 転送単位の DMA 転送のライトサイクル終了後に DMA 転送を終了します。

シングルアドレスモードでは、1 転送単位の DMA 転送のバスサイクル終了後に DMA 転送を終了します。

(b) ブロック転送モード

ブロックサイズ分の DMA 転送を中断して強制終了します。ブロックサイズ分の転送が正しく実行されなくなるため、転送要求に対する整合性は保証されません。

デュアルアドレスモードでは、ノーマル転送モードと同様にリードサイクルに対応するライトサイクルは実行されます。

(7) アドレスエラーによる転送終了

アドレスエラーが発生すると DMAC は全チャンネルの DTE ビットを 0 にクリアし、かつ DMDR_0 の ERRF ビットを 1 にセットします。DMA 転送中にアドレスエラーが発生した場合、転送は強制的に終了します。アドレスエラー発生後に DMA 転送を行う場合は、ERRF ビットを 0 にクリアした後に各チャンネルの DTE ビットを 1 にセットしてください。

アドレスエラー検出後の転送終了タイミングは、NMI 割り込みによる強制終了タイミングと同様です。

(8) ハードウェアスタンバイモード*、リセット入力による転送終了

ハードウェアスタンバイモード*またはリセット入力により、DMAC は初期化されます。DMA 転送は保証されません。

【注】 * 本 LSI には、ハードウェアスタンバイモードはありません。

7.6 DMAC と他のバスマスタの関係

7.6.1 CPU に対する DMAC の優先レベル制御機能

CPU プライオリティコントロールレジスタ (CPUPCR) の設定により、CPU に対する DMAC の優先レベル制御機能を使用することができます。詳細は「5.7 CPU に対する DTC、DMAC の優先レベル制御機能」を参照してください。

DMAC の優先レベルは、DMDR の DMAP2~0 ビットで設定します。チャンネル毎に優先レベルを独立に設定することができます。

CPU の優先レベルは、CPUPCR の CPUP2~0 ビットで設定します。CPU の優先レベルは、CPUP2~0 ビットの値を割り込みマスクビットの値で更新することにより、例外処理のプライオリティレベルに対応した優先レベルに設定されます。

CPUPCR の CPUPCE ビットを 1 にセットして CPU の優先レベル制御を有効にすると、DMAC の優先レベルが CPU の優先レベルよりも低くなると対応するチャンネルの転送要求はマスクされ、当該チャンネルは起動されなくなります。他のチャンネルの優先レベルが CPU の優先レベルと同じか、高い場合は、チャンネル間の優先順位に関係なく、他のチャンネルの転送要求は受け付けられ転送が可能です。

CPU の優先レベル制御機能により転送要求をマスクされた転送要因は保留され、CPU の優先レベルまたは当該チャンネルの優先レベルを更新して CPU の優先レベルが当該チャンネルの優先レベルよりも低くなると、転送要求を受け付けて転送を開始します。保留されている転送要求は、DTE ビットに 0 をライトするとクリアされます。

CPUPCE ビットを 0 にクリアすると CPU の優先レベルは最低と見なされます。

7.6.2 他のバスマスタとのバス権の調停

DMA 転送サイクルが連続する場合、各バスサイクルの間に他のバスマスタのサイクルが挿入される場合と、挿入されない場合があります。設定により DMAC のバス権をいったん解放して他のバスマスタにバス権を譲ることができます。

DMA 転送サイクルが連続するバスサイクルには、転送モードの仕様により不可分割となる場合、転送モードの仕様により高速アクセスのためバスサイクルが連続する場合があります。

DMA 転送サイクルのリードとライトの間は不可分割となっています。このため、DMA 転送サイクルのリードとライトの間にリフレッシュサイクル、外部バス解放サイクル、内部バスマスタ (CPU、DTC) のサイクルは発生しません。

ブロック転送モード、およびオートリクエストのバーストモードの転送では、DMA 転送のバスサイクルは連続します。この期間、CPU、DTC はバス権の優先順位が DMAC よりも低いいため外部空間アクセスは待たされます (バスコントロールレジスタ 2 (BCR2) の IBCCS=0 のとき)。

他のチャンネルに切り替わるとき、およびオートリクエストのサイクルスチールモードの転送では、DMA 転送サイクルは内部バスマスタと交互にバスサイクルを実行します。

BCR2 の IBCCS ビットを 1 にセットして DMAC と内部バスマスタとの間の調停機能を有効にすると、不可分割のバスサイクルを除いて、バス権をいったん解放して内部バスマスタとの間で交互にバス権を取得します。詳細は「第 6 章 バスコントローラ (BSC)」を参照してください。

DMAC が外部空間をアクセスする場合、外部バス解放サイクルとの競合が発生することがあります。このときバスコントローラは、外部バスの優先順位に従って、DMAC がバースト転送またはブロック転送を行っているときでも転送をいったん中断し、外部バス解放サイクルを挿入します (DMAC よりも優先順位の低い DTC、CPU の外部アクセスは、DMAC が外部バスを解放するまで動作しません)。

DMAC の転送モードがデュアルアドレスモードの場合、DMAC が外部バスを解放するタイミングは外部ライトサイクルの後となります。外部リードサイクルと外部ライトサイクルの間は不可分割のため、この間にバスを解放することはありません。

DMAC が内部空間 (内蔵メモリ、内部 I/O レジスタ) をアクセスする場合、DMAC サイクルが外部バス解放サイクルと同時に実行される場合があります。

7.7 割り込み要因

DMAC が発生する割り込み要因は、転送カウンタによる転送終了、転送カウンタが 0 にならずに転送終了するエスケープ終了割り込みです。表 7.6 に割り込み要因と優先度を示します。

表 7.6 割り込み要因と優先度

名称	割り込み要因	割り込み優先順位
DMTEND0	チャンネル 0 の転送カウンタによる転送終了割り込み	高 ↑ 低
DMTEND1	チャンネル 1 の転送カウンタによる転送終了割り込み	
DMTEND2	チャンネル 2 の転送カウンタによる転送終了割り込み	
DMTEND3	チャンネル 3 の転送カウンタによる転送終了割り込み	
DMEEND0	チャンネル 0 の転送サイズエラーによる割り込み チャンネル 0 のリピートサイズ終了割り込み チャンネル 0 のソースアドレス拡張リピートエリアオーバーフローによる割り込み チャンネル 0 のデスティネーションアドレス拡張リピートエリアオーバーフローによる割り込み	
DMEEND1	チャンネル 1 の転送サイズエラーによる割り込み チャンネル 1 のリピートサイズ終了割り込み チャンネル 1 のソースアドレス拡張リピートエリアオーバーフローによる割り込み チャンネル 1 のデスティネーションアドレス拡張リピートエリアオーバーフローによる割り込み	
DMEEND2	チャンネル 2 の転送サイズエラーによる割り込み チャンネル 2 のリピートサイズ終了割り込み チャンネル 2 のソースアドレス拡張リピートエリアオーバーフローによる割り込み チャンネル 2 のデスティネーションアドレス拡張リピートエリアオーバーフローによる割り込み	
DMEEND3	チャンネル 3 の転送サイズエラーによる割り込み チャンネル 3 のリピートサイズ終了割り込み チャンネル 3 のソースアドレス拡張リピートエリアオーバーフローによる割り込み チャンネル 3 のデスティネーションアドレス拡張リピートエリアオーバーフローによる割り込み	

各割り込み要因は、対応するチャンネルの DMDR の DTIE ビットと ESIE ビットにより許可または禁止を設定できます。DTIE ビットは、DMDR の DTIF ビットとの組み合わせで DMTEND 割り込みを発生させます。ESIE ビットは、DMDR の ESIF ビットとの組み合わせで DMEEND 割り込みを発生させます。DMEEND 内の割り込み要因は共通の割り込みとして区別されません。チャンネル間の割り込みの優先順位は、割り込みコントローラによって決められており、表 7.7 に示すようになっています。詳細は「第 5 章 割り込みコントローラ」を参照してください。

各割り込み要因は、対応するチャンネルのレジスタの割り込みイネーブルビットで設定します。転送カウンタによる転送終了割り込みは DMDR の DTIE ビット、転送サイズエラーによる割り込みは DMDR の TSEIE ビット、リピートサイズ終了による割り込みは DACR の RPTIE ビット、ソースアドレス拡張リピートエリアオーバーフロー割り込みは DACR の SARIE ビット、デスティネーションアドレス拡張リピートエリアオーバーフロー割り込みは DACR の DARIE ビットにより許可または禁止を設定します。

転送カウンタによる転送終了割り込みは、DMDR の DTIE ビットが 1 にセットされていると、転送により DTCR が 0 になり DMDR の DTIF ビットが 1 にセットされたときに発生します。

転送カウンタによる転送終了割り込み以外の割り込みは、対応する割り込みイネーブルビットが 1 にセットされていると、転送により対応する条件が発生し DMDR の ESIF ビットが 1 にセットされたときに発生します。

転送サイズエラーによる割り込みは、データアクセスサイズよりも DTCR の値が小さくなり 1 要求あたりのデータアクセスサイズ分の転送ができないときに転送要求があると発生します。ブロック転送モードでは、ブロックサイズと DTCR の値を比較して転送サイズエラー判定を行います。

リピートサイズ終了割り込みは、リピート転送モードにおいてリピートサイズ分の転送終了後に次の転送要求があると発生します。リピートエリアをアドレスレジスタに設定しない場合でもリピートサイズに設定した値により定期的に転送を中断させることができます。同時に転送カウンタによる転送終了割り込みが発生した場合も、ESIF ビットは 1 にセットされます。

ソースアドレス拡張リピートエリアオーバフロー割り込み、およびデスティネーションアドレス拡張リピートエリアオーバフロー割り込みは、拡張リピートエリアを設定して対応するアドレスがリピートエリアオーバフローになったときに発生します。同時に転送カウンタによる転送終了割り込みが発生した場合も、ESIF ビットは 1 にセットされます。

各種割り込みと割り込みフラグのブロック図を図 7.39 に示します。割り込みを解除するには、割り込み処理ルーチンにて DMDR の DTIF ビットまたは ESIF ビットを 0 にクリアする方法と、レジスタの再設定後に DMDR の DTE ビットを 1 にセットして転送継続の処理を行う方法があります。割り込みを解除して、転送を再開する手順例を図 7.40 に示します。

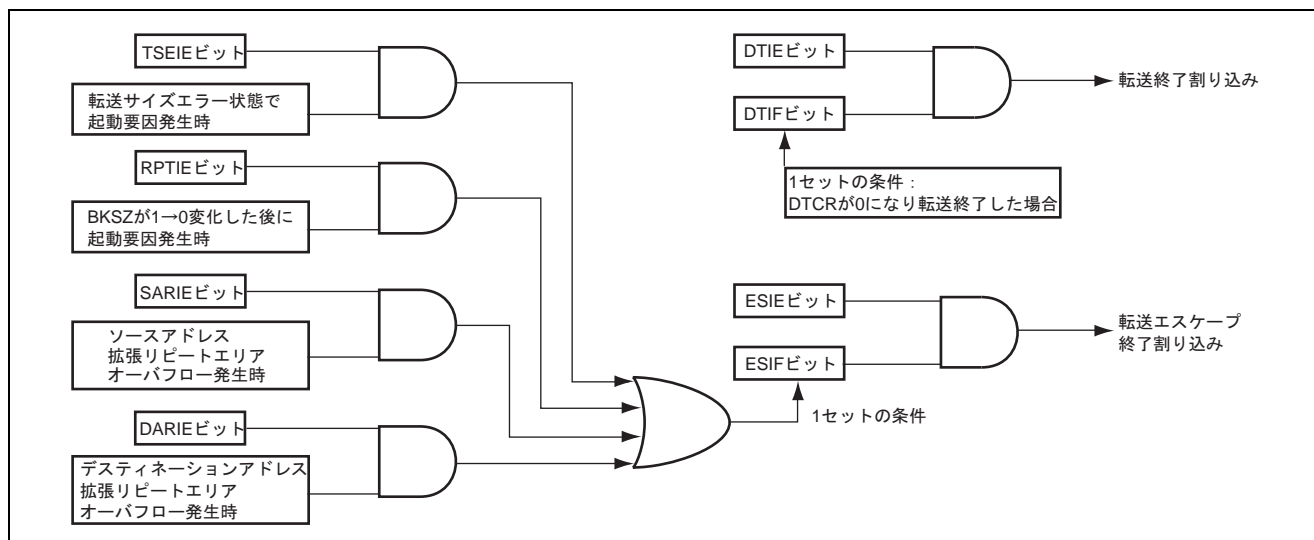


図 7.39 割り込みと割り込み要因

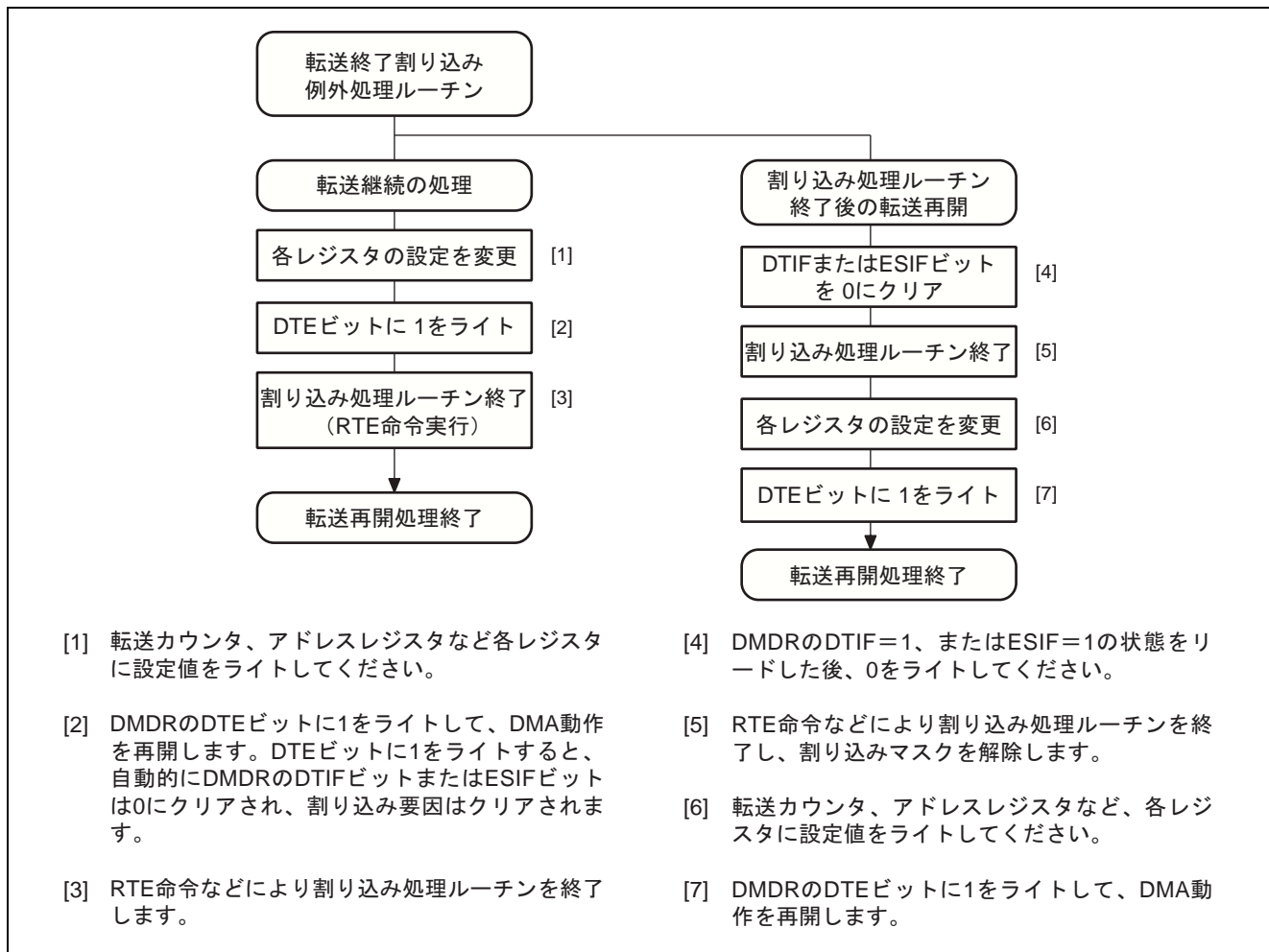


図 7.40 割り込みを解除して、転送を再開する手順例

7.8 使用上の注意事項

(1) 動作中の DMAC レジスタのアクセス

DMDR の DTE ビットを 0 にクリアする場合を除き、動作中（転送待ち状態を含む）にチャンネルの設定を変更しないでください。動作中にチャンネルの設定を変更する場合は、必ず転送禁止状態で行ってください。

(2) モジュールストップ機能の設定

モジュールストップコントロールレジスタにより、DMAC の動作停止／許可を設定することができます。初期値では、DMAC は動作許可状態です。

MSTPCRA の MSTPA13 ビットを 1 にセットすると、DMAC に供給されるクロックが停止し、DMAC はモジュールストップ状態になります。ただし、DMAC のいずれかのチャンネルが転送許可状態になっている場合、および割り込み要求が発生中の場合は、MSTPA13 ビットを 1 にセットできません。DMDR の DTE ビットを 0 にクリアして、DMDR の DTIF ビットまたは DTIE ビットを 0 にクリアしてから MSTPA13 ビットを設定してください。

DMAC のクロックが停止すると、DMAC のレジスタに対するアクセスができなくなります。次の DMAC のレジスタ設定は、モジュールストップ状態でも有効となりますので、必要に応じてモジュールストップ状態に先立って無効にしてください。

- DMDR の TEND \bar{E} =1 ($\overline{\text{TEND}}$ 端子イネーブル)
- DMDR の DACK \bar{E} =1 ($\overline{\text{DACK}}$ 端子イネーブル)

(3) $\overline{\text{DREQ}}$ 立ち下がりエッジ起動

$\overline{\text{DREQ}}$ 立ち下がりエッジ検出は、DMAC の内部動作に同期して行います。

1. 起動要求待ち状態： $\overline{\text{DREQ}}$ の Low レベル検出を待ち。[2.] に遷移します。
2. 転送待ち状態：DMAC のデータ転送許可待ち。[3.] に遷移します。
3. 起動要求禁止状態： $\overline{\text{DREQ}}$ の High レベル検出を待ち。[1.] に遷移します。

DMAC の転送許可後は [1.] に遷移します。このため、転送許可後の最初の起動は Low レベル検出で行われます。

(4) 起動要因の受け付け

起動要因の受け付け開始時は、 $\overline{\text{DREQ}}$ の立ち下がりエッジセンス／Low レベルセンスとともに Low レベルを検出しています。したがって、転送許可状態にするための DMDR ライト実行以前から発生している $\overline{\text{DREQ}}$ の Low レベル検出による転送要求を受け付けます。

DMAC の起動時には、前回の転送終了時に $\overline{\text{DREQ}}$ の Low レベルが残らないようにしてください。

8. データトランスファコントローラ (DTC)

本 LSI は、データトランスファコントローラ (DTC) を内蔵しています。DTC は、割り込み要求によって起動され、データ転送を行うことができます。

8.1 特長

- 任意チャネル数の転送が可能
一つの起動要因に対して複数のデータ転送が可能 (チェイン転送)
データ転送後にチェイン転送の実行を設定可能 (カウンタ=0 のとき)
- 転送モード : 3種類
ノーマル転送モード、リピート転送モード、ブロック転送モードの選択が可能
転送元、転送先アドレスのインクリメント、デクリメント、固定の選択が可能
- ショートアドレスモード/フルアドレスモードを選択可能
転送情報は、ショートアドレスモードのとき 3 ロングワード、フルアドレスモードのとき 4 ロングワードに配置
ショートアドレスモードでは転送元、転送先アドレスを 24 ビットで指定でき、16M バイトのアドレス空間を直接指定可能
フルアドレスモードでは転送元、転送先アドレスを 32 ビットで指定でき、4G バイトのアドレス空間を直接指定可能
- データ転送のデータサイズをバイト、ワード、ロングワードに設定可能
奇数アドレスを指定し、ワード、ロングワード転送を行った場合にバスサイクルを分割して転送
4n+2 アドレスを指定し、ロングワード転送を行った場合にバスサイクルを分割して転送
- DTCを起動した割り込みをCPUに要求可能
1 回のデータ転送終了後に CPU に対する割り込み要求を発生可能
指定したデータ転送終了後に CPU に対する割り込み要求を発生可能
- 転送情報のリードスキップを指定可能
- 固定を選択した転送元アドレス、転送先アドレスはライトバックスキップを実行
- モジュールストップモードの設定可能

【注】 本 LSI はアドレス空間が 16M バイトであるため、ショートアドレスモードのみサポートします。フルアドレスモードは選択できません。

DTCブロック図を図 8.1 に示します。DTC の転送情報は、データ領域に配置可能です*。転送情報を内蔵 RAM に配置した場合、DTC と内蔵 RAM は 32 ビットバスで接続されていますので、DTC の転送情報のリード/ライトを 32 ビット 1 ステート (ライトは 1 ステートまたは 2 ステート) で実行できます。

【注】 * RAM へのリード/ライト可能な設定にしてください。詳細は「第 19 章 RAM」を参照してください。

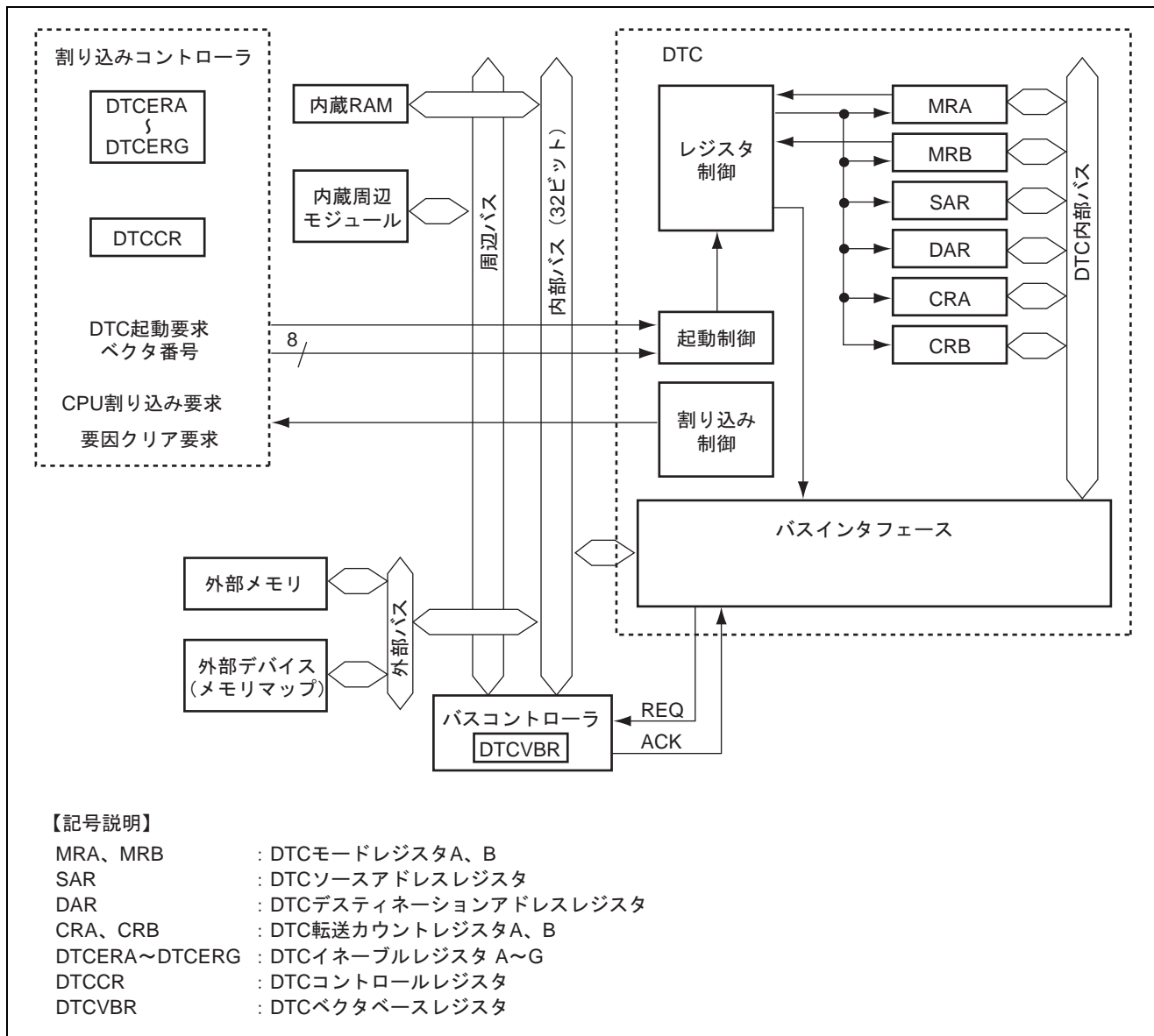


図 8.1 DTC のブロック図

8.2 レジスタの説明

DTC には以下のレジスタがあります。

- DTCモードレジスタA (MRA)
- DTCモードレジスタB (MRB)
- DTCソースアドレスレジスタ (SAR)
- DTCデスティネーションアドレスレジスタ (DAR)
- DTC転送カウントレジスタA (CRA)
- DTC転送カウントレジスタB (CRB)

MRA、MRB、SAR、DAR、CRA、CRB の 6 本のレジスタは、CPU から直接アクセスすることはできません。データ領域に転送情報として配置します。DTC 起動要因が発生すると、起動要因ごとに決められたベクタアドレスに従って転送情報の先頭アドレスを読み出し、任意の転送情報を DTC 内に転送してデータ転送を行います。転送が終了すると、これらのレジスタの内容がライトバックされます。

- DTCイネーブルレジスタA~G (DTCERA~DTCERG)
- DTCコントロールレジスタ (DTCCR)
- DTCベクタベースレジスタ (DTCVBR)

8.2.1 DTC モードレジスタ A (MRA)

MRA は、DTC の動作モードの選択を行います。MRA は、CPU から直接アクセスすることができません。

ビット	7	6	5	4	3	2	1	0
ビット名	MD1	MD0	Sz1	Sz0	SM1	SM0	—	—
初期値 :	不定	不定	不定	不定	不定	不定	不定	不定
R/W :	—	—	—	—	—	—	—	—

ビット	ビット名	初期値	R/W	説明
7	MD1	不定	—	DTC モード 1、0
6	MD0	不定	—	DTC の転送モードを指定します。 00 : ノーマル転送モード 01 : リピート転送モード 10 : ブロック転送モード 11 : 設定禁止
5	Sz1	不定	—	DTC データトランスファサイズ 1、0
4	Sz0	不定	—	転送データのサイズを指定します。 00 : バイトサイズ転送 01 : ワードサイズ転送 10 : ロングワードサイズ転送 11 : 設定禁止
3	SM1	不定	—	ソースアドレスモード 1、0
2	SM0	不定	—	データ転送後の SAR の動作を指定します。 0X : SAR は固定 (SAR のライトバックはスキップされます。) 10 : 転送後 SAR をインクリメント (Sz1、Sz0 が B'00 のとき+1、B'01 のとき+2、B'10 のとき+4) 11 : 転送後 SAR をデクリメント (Sz1、Sz0 が B'00 のとき-1、B'01 のとき-2、B'10 のとき-4)
1	—	不定	—	リザーブビット
0	—	不定	—	ライトする値は常に 0 にしてください。

【注】 X : Don't care

8.2.2 DTC モードレジスタ B (MRB)

MRB は、DTC の動作モードの選択を行います。MRB は、CPU から直接アクセスできません。

ビット	7	6	5	4	3	2	1	0
ビット名	CHNE	CHNS	DISEL	DTS	DM1	DM0	—	—
初期値 :	不定	不定	不定	不定	不定	不定	不定	不定
R/W :	—	—	—	—	—	—	—	—

ビット	ビット名	初期値	R/W	説明
7	CHNE	不定	—	DTC チェイン転送イネーブル チェイン転送を指定します。チェイン転送の詳細は「8.5.7 チェイン転送」を参照してください。チェイン転送の条件の選択は CHNS ビットで行います。 0 : チェイン転送禁止 1 : チェイン転送許可
6	CHNS	不定	—	DTC チェイン転送セレクト チェイン転送の条件を選択します。次の転送がチェイン転送の場合、指定した転送回数の終了判定、起動要因フラグまたは DTCER のクリアは行いません。 0 : 連続してチェイン転送を行う 1 : 転送カウンタ=0 のときのみチェイン転送を行う
5	DISEL	不定	—	DTC インタラプトセレクト このビットが 1 のとき、DTC データ転送のたびに CPU に対して割り込み要求を発生します。このビットが 0 のときは指定されたデータ転送を終了したときだけ CPU に対して割り込み要求を発生します。
4	DTS	不定	—	DTC 転送モードセレクト リピート転送モードまたはブロック転送モードのとき、ソース側とデスティネーション側のいずれをリピート領域またはブロック領域とするかを指定します。 0 : デスティネーション側がリピート領域またはブロック領域 1 : ソース側がリピート領域またはブロック領域
3 2	DM1 DM0	不定 不定	— —	デスティネーションアドレスモード 1、0 データ転送後の DAR の動作を指定します。 0X : DAR は固定 (DAR のライトバックはスキップされます。) 10 : 転送後 DAR をインクリメント (Sz1、Sz0 が B'00 のとき+1、B'01 のとき+2、B'10 のとき+4) 11 : 転送後 DAR をデクリメント (Sz1、Sz0 が B'00 のとき-1、B'01 のとき-2、B'10 のとき-4)
1	—	不定	—	リザーブビット
0	—	不定	—	ライトする値は常に 0 にしてください。

【注】 x : Don't care

8.2.3 DTC ソースアドレスレジスタ (SAR)

SAR は 32 ビットのレジスタで、DTC の転送するデータの転送元アドレスを指定します。

フルアドレスモードでは 32 ビットが有効です。ショートアドレスモードでは下位 24 ビットが有効で、ビット 31~24 の設定は無視されます。このときアドレスの上位 8 ビットは、ビット 23 で指定した値で補完されず。

SAR に奇数アドレスを設定してワードアクセス、ロングワードアクセスする場合と、アドレスを $4n+2$ 番地に設定してロングワードアクセスする場合は、バスサイクルを分割してデータ転送を行います。詳細は、「8.5.1 バスサイクルの分割」を参照してください。

SAR は、CPU から直接アクセスすることはできません。

8.2.4 DTC デスティネーションアドレスレジスタ (DAR)

DAR は 32 ビットのレジスタで、DTC の転送するデータの転送先アドレスを指定します。

フルアドレスモードでは 32 ビットが有効です。ショートアドレスモードでは下位 24 ビットが有効で、ビット 31~24 の設定は無視されます。このときアドレスの上位 8 ビットは、ビット 23 で指定した値で補完されず。

DAR に奇数アドレスを設定してワードアクセス、ロングワードアクセスする場合と、アドレスを $4n+2$ 番地に設定してロングワードアクセスする場合は、バスサイクルを分割してデータ転送を行います。詳細は、「8.5.1 バスサイクルの分割」を参照してください。

DAR は、CPU から直接アクセスすることはできません。

8.2.5 DTC 転送カウンタレジスタ A (CRA)

CRA は 16 ビットのレジスタで、DTC のデータ転送の転送回数を指定します。

ノーマル転送モードでは、一括して 16 ビットの転送カウンタ (1~65536) として機能します。1 回のデータ転送を行うたびにデクリメント (-1) されます。カウンタ値が H'0000 になると、起動要因に対応する DTCE_n ビット (n=15~0) をクリアした後に CPU に割り込み要求を発生します。転送回数は、設定値が H'0001 のときは 1 回、H'FFFF のときは 65535 回で、H'0000 のときは 65536 回になります。

リピート転送モードでは、上位 8 ビットの CRAH と下位 8 ビットの CRAL に分割されます。CRAH は転送回数を保持し、CRAL は 8 ビットの転送カウンタ (1~256) として機能します。CRAL は 1 回のデータ転送を行うたびにデクリメント (-1) され、カウンタ値が H'00 になると CRAH の内容が転送されます。転送回数は、設定値が CRAH=CRAL=H'01 のときは 1 回、H'FF のときは 255 回で、H'00 のときは 256 回になります。

ブロック転送モードでは、上位 8 ビットの CRAH と下位 8 ビットの CRAL に分割されます。CRAH はブロックサイズを保持し、CRAL は 8 ビットのブロックサイズカウンタ (1~256 バイト、1~256 ワード、または 1~256 ロングワード) として機能します。CRAL は 1 回のデータ転送を行うたびに 1 バイト (または 1 ワード、1 ロングワード) 毎にデクリメント (-1) され、カウンタ値が H'00 になると、CRAH の内容が転送されます。ブロックサイズは設定値が CRAH=CRAL=H'01 のときは 1 バイト (または 1 ワード、1 ロングワード)、H'FF のときは 255 バイト (または 255 ワード、255 ロングワード) で、H'00 のときは 256 バイト (または 256 ワード、256 ロングワード) になります。

CRA は、CPU から直接アクセスすることはできません。

8.2.6 DTC 転送カウントレジスタ B (CRB)

CRB は 16 ビットのレジスタで、ブロック転送モードのとき DTC のブロックデータ転送の転送回数を指定します。16 ビットの転送回数カウンタ (1~65536) として機能し、1 回のデータ転送を行うたびにデクリメント (-1) され、カウンタ値が H'0000 になると、起動要因に対応する DTCE_n ビット (n=15~0) をクリアした後 CPU に割り込み要求を発生します。転送回数は、設定値が H'0001 のときは 1 回、H'FFFF のときは 65535 回で、H'0000 のときは 65536 回になります。

ノーマル転送モードおよびリピート転送モードでは、CRB は使用しません。CRB は、CPU から直接アクセスすることはできません。

8.2.7 DTC イネーブルレジスタ A~G (DTCERA~DTCERG)

DTCE は、DTC を起動する割り込み要因を選択するためのレジスタで、DTCERA~DTCERG があります。各割り込み要因と DTCE ビットの対応については表 8.1 を参照してください。DTCE ビットの設定は、BSET 命令、BCLR 命令などのビット操作命令を使用してください。ただし初期設定に限り、複数の起動要因を一度に設定するために、割り込みをマスクして対象となるレジスタをダミーリードした後ライトすることができます。

ビット	15	14	13	12	11	10	9	8
ビット名	DTCE15	DTCE14	DTCE13	DTCE12	DTCE11	DTCE10	DTCE9	DTCE8
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

ビット	7	6	5	4	3	2	1	0
ビット名	DTCE7	DTCE6	DTCE5	DTCE4	DTCE3	DTCE2	DTCE1	DTCE0
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

ビット	ビット名	初期値	R/W	説明
15	DTCE15	0	R/W	DTC 起動イネーブル 15~0
14	DTCE14	0	R/W	1 をセットすると、対応する割り込み要因が DTC 起動要因として選択されます。 [クリア条件] クリアするビットの 1 の状態をリードした後、0 をライトしたとき MRB の DISEL ビットが 1 で、1 回のデータ転送を終了したとき 指定した回数の転送が終了したとき DISEL ビットが 0 で、指定した回数の転送が終了していないときはクリアされません。
13	DTCE13	0	R/W	
12	DTCE12	0	R/W	
11	DTCE11	0	R/W	
10	DTCE10	0	R/W	
9	DTCE9	0	R/W	
8	DTCE8	0	R/W	
7	DTCE7	0	R/W	
6	DTCE6	0	R/W	
5	DTCE5	0	R/W	
4	DTCE4	0	R/W	
3	DTCE3	0	R/W	
2	DTCE2	0	R/W	
1	DTCE1	0	R/W	
0	DTCE0	0	R/W	

8.2.8 DTC コントロールレジスタ (DTCCR)

DTCCR は、転送情報リードスキップを設定します。

ビット	7	6	5	4	3	2	1	0
ビット名	—	—	—	RRS	RCHNE	—	—	ERR
初期値 :	0	0	0	0	0	0	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。

ビット	ビット名	初期値	R/W	説明
7~5	—	0	R/W	リザーブビット リードすると常に0が読み出されます。ライトする値は常に0にしてください。
4	RRS	0	R/W	DTC 転送情報リードスキップイネーブル ベクタアドレスのリードと転送情報のリードを制御します。DTC ベクタ番号は、常に前回起動のベクタ番号と比較されます。ベクタ番号の値が一致し、このビットが1のとき、ベクタアドレスのリードと転送情報のリードを行わず、DTC のデータ転送を実施します。前回の起動がチェーン転送のときは、必ずベクタアドレスのリードと転送情報のリードが行われます。 0 : 転送情報リードスキップを行わない 1 : ベクタ番号の値が一致したとき転送情報リードスキップを行う
3	RCHNE	0	R/W	DTC リピート転送後チェーン転送イネーブル リピート転送において、転送カウンタ=0でのチェーン転送を許可/禁止します。リピート転送では、転送カウンタ (CRAL) =0 となった場合、CRAL は CRAH で指定した値に書き戻されるため、転送カウンタ=0でのチェーン転送は発生しません。このビットを1にセットすることで、転送カウンタの書き戻し時のチェーン転送が許可されます。 0 : リピート転送後のチェーン転送を禁止 1 : リピート転送後のチェーン転送を許可
2	—	0	R	リザーブビット
1	—	0	R	これらのビットはリードのみ有効で、ライトは無効です。
0	ERR	0	R/(W)*	転送停止フラグ アドレスエラー、または NMI 割り込み要求が発生したことを示すフラグです。アドレスエラー、または NMI 割り込み要求が発生すると DTC は停止します。 0 : 割り込み要求なし 1 : 割り込み要求発生 [クリア条件] • 1の状態をリードした後、0をライトしたとき

【注】 * フラグをクリアするための0ライトのみ可能です。

8.2.9 DTC ベクタベースレジスタ (DTCVBR)

DTCVBR は 32 ビットのレジスタで、ベクタテーブルアドレス算出時のベースアドレスを設定します。ビット 31~28、ビット 11~0 は 0 に固定されおり、ライトは無効です。

DTCVBR の初期値は、H'00000000 です。

ビット	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ビット名																
初期値:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ビット	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ビット名																
初期値:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

8.3 起動要因

DTC は、割り込み要求により起動します。起動する割り込み要因は、DTCER で選択します。対応するビットを 1 にセットすると DTC の起動要因となり、0 にクリアすると CPU の割り込み要因となります。1 回のデータ転送（チェーン転送の場合、連続した最後の転送）終了時に、起動要因となった割り込みフラグまたは DTCER の対応するビットをクリアします。

8.4 転送情報の配置と DTC ベクタテーブル

転送情報は、データ領域上に配置します。転送情報の先頭アドレスは、4n 番地としてください。4n 番地以外を指定した場合、下位 2 ビットを無視してアクセスします ([1:0]=B'00)。転送情報は、ショートアドレスモード（3 ロングワード）、フルアドレスモード（4 ロングワード）のいずれかで配置できます。データ領域上での転送情報の配置を図 8.2 に示します。DTC は起動要因別にベクタテーブルから転送情報の先頭アドレスをリードし、この先頭アドレスから転送情報をリードします。DTC ベクタテーブルと転送情報の対応を図 8.3 に示します。

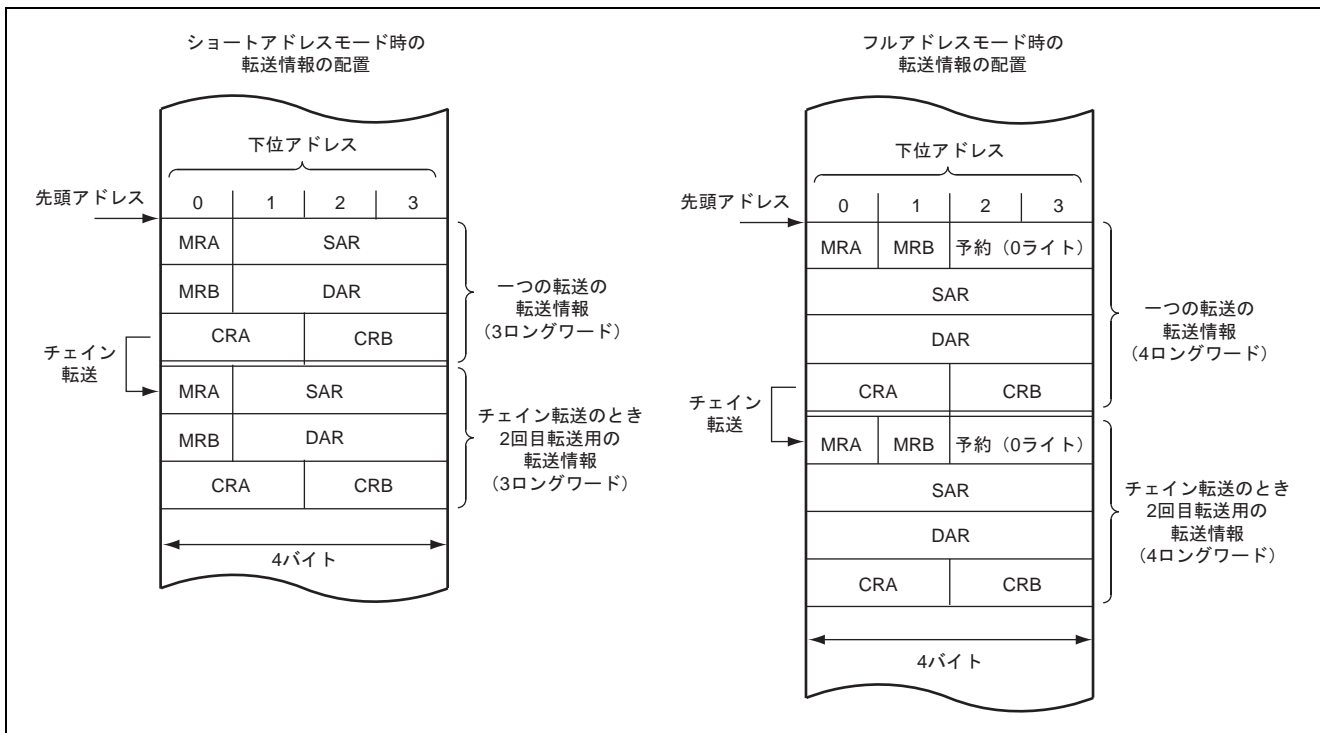


図 8.2 データ領域上での転送情報の配置

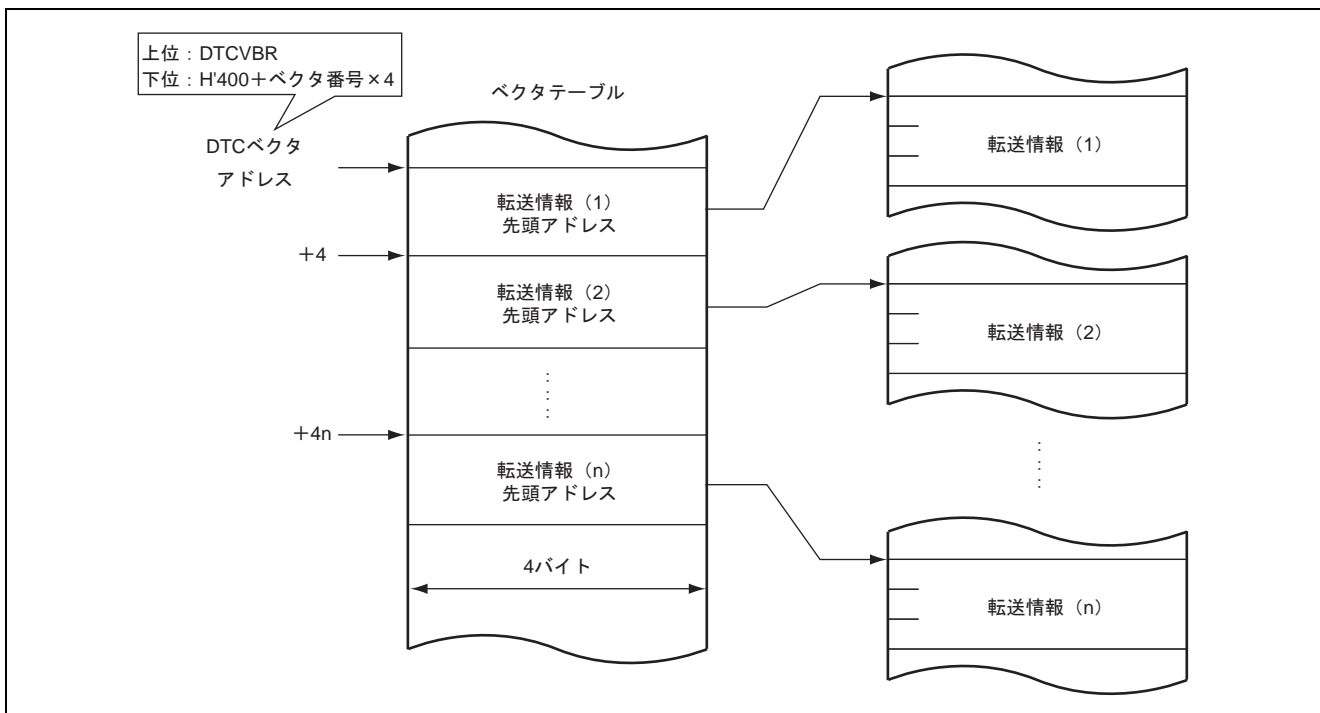


図 8.3 DTC ベクタテーブルと転送情報の対応

8.5 動作説明

DTC は、転送情報をデータ領域に格納します。DTC が起動すると、データ領域から転送情報をリードしてデータ転送を行い、データ転送後の転送情報をライトバックします。転送情報をデータ領域に格納することで、任意のチャンネル数のデータ転送を行うことができます。転送モードには、ノーマル転送モード、リピート転送モード、ブロック転送モードがあります。

DTC は、転送元アドレスを SAR、転送先アドレスを DAR で指定します。SAR、DAR は転送後、それぞれ独立にインクリメントまたはデクリメント、あるいは固定されます。

DTC の転送モードを表 8.2 に示します。

表 8.2 DTC の転送モード

転送モード	1 回の転送要求で 転送可能なデータサイズ	メモリアドレスの増減	転送回数
ノーマル転送モード	1 バイト/ワード/ロングワード	1、2 または 4 増減・固定	1~65536 回
リピート転送モード*1	1 バイト/ワード/ロングワード	1、2 または 4 増減・固定	1~256 回*3
ブロック転送モード*2	CRAH で指定したブロックサイズ (1~256 バイト/ワード/ロング ワード)	1、2 または 4 増減・固定	1~65536 回

【注】 *1 ソースまたはデスティネーションのいずれかをリピートエリアに設定

*2 ソースまたはデスティネーションのいずれかをブロックエリアに設定

*3 指定回数転送後、初期状態を回復して動作を継続

また、MRB の CHNE ビットを 1 にセットしておくことにより、一つの起動要因で複数の転送を行うことができます (チェーン転送)。MRB の CHNS ビットの設定で、転送カウンタ=0 のときにチェーン転送を行う設定も可能です。

DTC の動作フローチャートを図 8.4 に示します。チェーン転送の条件を表 8.3 に示します (第 2 の転送から第 3 の転送を行う組み合わせは省略してあります)。

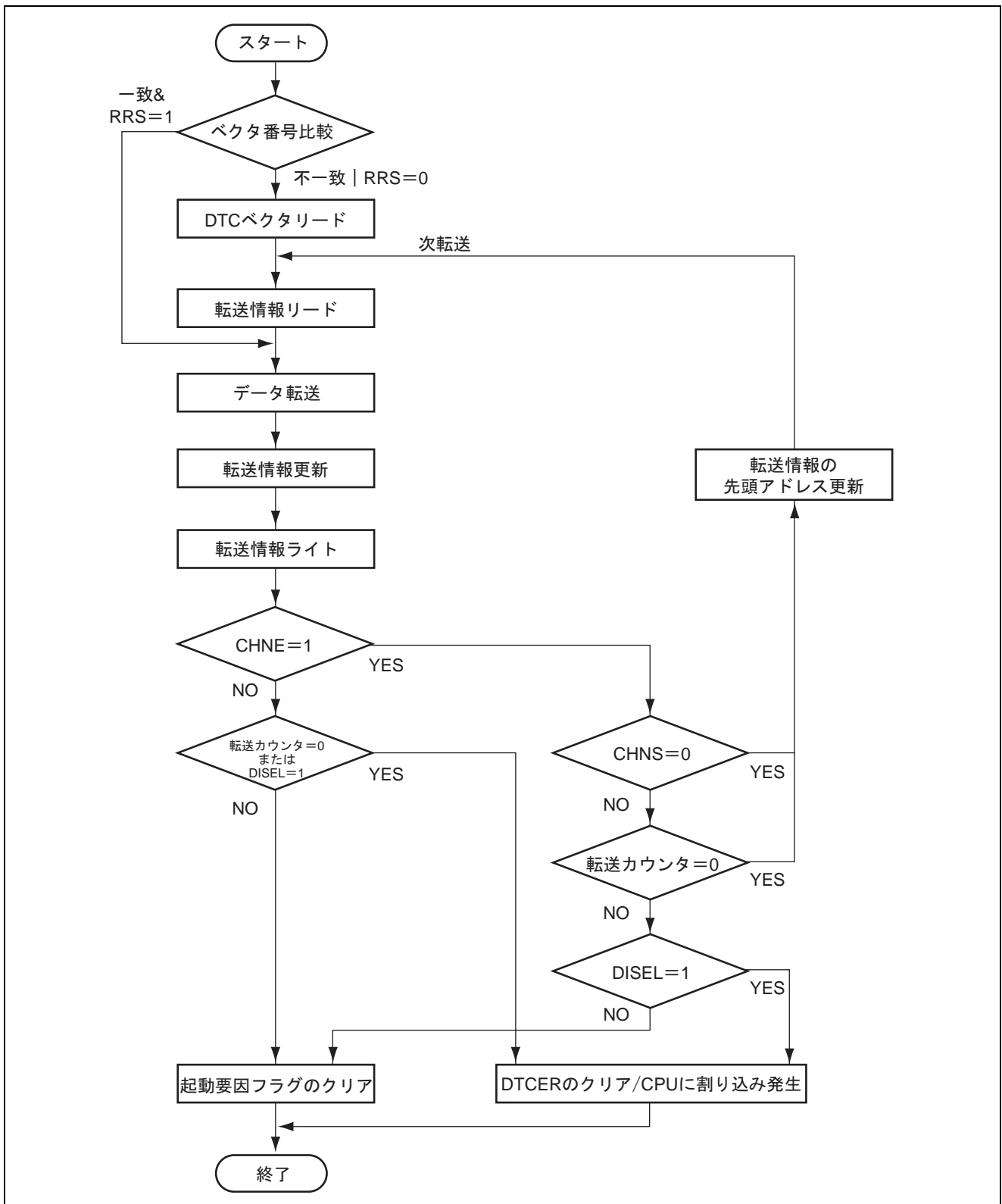


図 8.4 DTC 動作フローチャート

表 8.3 チェイン転送の条件

第 1 の転送				第 2 の転送				DTC 転送
CHNE	CHNS	DISEL	転送 カウンタ* ¹	CHNE	CHNS	DISEL	転送 カウンタ* ¹	
0	—	0	0 以外	—	—	—	—	第 1 転送で終了
0	—	0	0* ²	—	—	—	—	第 1 の転送で終了
0	—	1		—	—	—	—	CPU へ割り込み要求
1	0	—	—	0	—	0	0 以外	第 2 の転送で終了
				0	—	0	0* ²	第 2 の転送で終了
				0	—	1	—	CPU へ割り込み要求
1	1	0	0 以外	—	—	—	—	第 1 転送で終了
1	1	—	0* ²	0	—	0	0 以外	第 2 転送で終了
				0	—	0	0* ²	第 2 の転送で終了
				0	—	1	—	CPU へ割り込み要求
1	1	1	0 以外	—	—	—	—	第 1 の転送で終了 CPU へ割り込み要求

【注】 *1 ノーマル転送モード : CRA、リポート転送モード : CRAL、ブロック転送モード : CRB

*2 リポート転送モードで CRAL の内容が CRAH の内容に書き換わるとき

8.5.1 バスサイクルの分割

転送データサイズにワード、またはロングワードを指定した場合、ワード境界、ロングワード境界を外れて SAR、DAR を設定すると、バスサイクルを分割し、データはバイトサイズ、ワードサイズでリード、またはライトされます。

表 8.4 に SAR、DAR のアドレス設定値、転送データサイズ設定値とバスサイクル分割数、アクセスデータサイズの関係性を、図 8.5 にバスサイクル分割時の動作例を示します。

表 8.4 バスサイクル分割数とアクセスサイズ

SAR、DAR 設定値	設定データサイズ		
	バイト (B)	ワード (W)	ロングワード (LW)
4n アドレス	1 (B)	1 (W)	1 (LW)
2n+1 アドレス	1 (B)	2 (B-B)	3 (B-W-B)
4n+2 アドレス	1 (B)	1 (W)	2 (W-W)

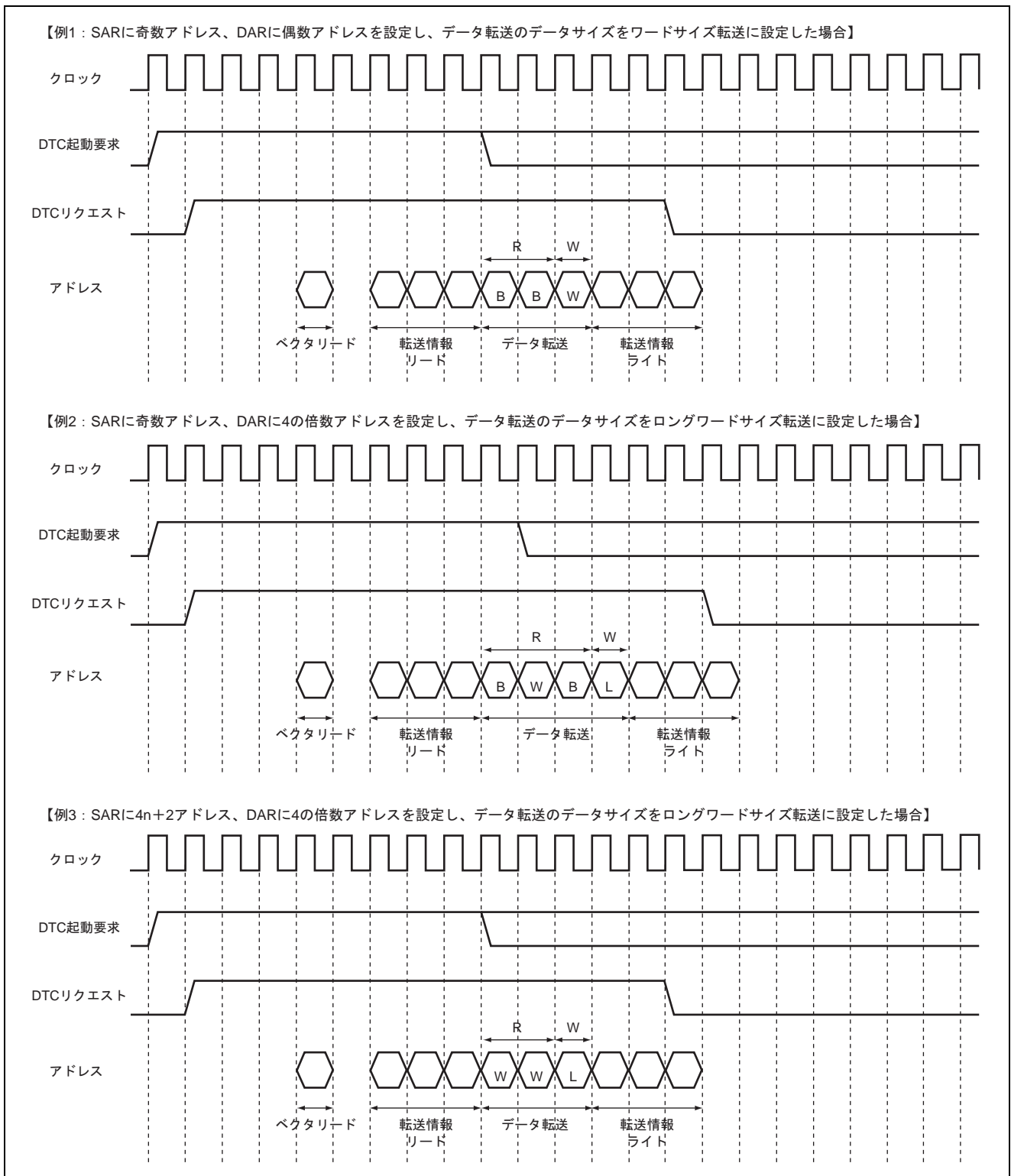


図 8.5 バスサイクルが分割される例

8.5.2 転送情報リードスキップ機能

DTCの RRS ビットの設定で、ベクタアドレスのリードと転送情報のリードをスキップすることができます。DTC ベクタ番号は、常に前回起動のベクタ番号と比較します。比較結果が一致し、RRS=1 のとき、ベクタアドレスのリードと転送情報のリードを行わず、DTC のデータ転送を行います。前回の起動がチェーン転送のときは、必ずベクタアドレスのリードと転送情報のリードが行われます。転送情報リードスキップのタイミングチャートを図 8.6 に示します。

ベクタテーブルと転送情報を更新する場合には、一度 RRS=0 に設定し、ベクタテーブルと転送情報を更新した後、RRS ビットを設定してください。RRS=0 にすると、保持されていたベクタ番号は破棄され、次の起動時に更新されたベクタテーブルおよび転送情報がリードされます。

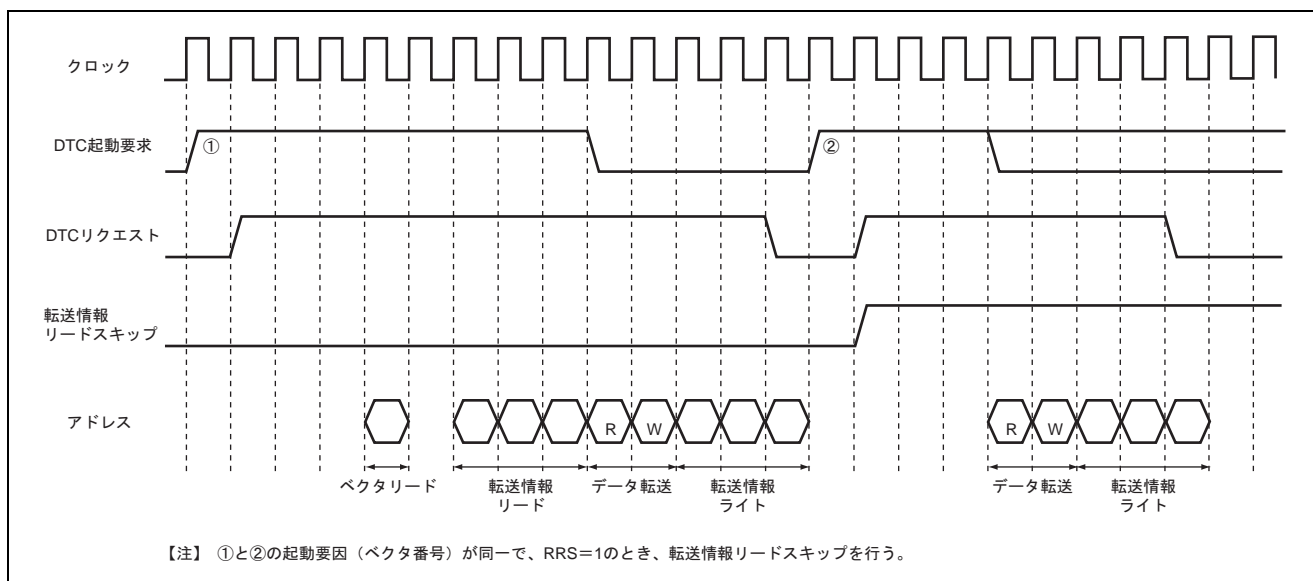


図 8.6 転送情報リードスキップのタイミングチャート

8.5.3 転送情報ライトバックスキップ機能

MRA の SM1 ビットと MRB の DM1 ビットをアドレス固定に設定すると、転送情報の一部はライトバックされません。この機能は、ショートアドレスモード、フルアドレスモードの設定にかかわらず行われます。転送情報ライトバックスキップの条件とライトバックスキップされるレジスタを表 8.5 に示します。CRA、CRB は、ショートアドレスモード、フルアドレスモードの設定にかかわらず、必ずライトバックされます。また、フルアドレスモードでは、MRA、MRB は必ずライトバックスキップされます。

表 8.5 転送情報ライトバックスキップの条件とライトバックスキップされるレジスタ

SM1	DM1	SAR	DAR
0	0	スキップ	スキップ
0	1	スキップ	ライトバック
1	0	ライトバック	スキップ
1	1	ライトバック	ライトバック

8.5.4 ノーマル転送モード

一つの起動要因で、1バイト、1ワードまたは1ロングワードの転送を行います。転送回数は1~65536です。転送元アドレスと転送先アドレスは、増加、減少または固定にそれぞれ設定できます。指定回数の転送が終了すると、CPUへ割り込み要求を発生することができます。

ノーマル転送モードのレジスタ機能を表 8.6 に、ノーマル転送モードのメモリマップを図 8.7 に示します。

表 8.6 ノーマル転送モードのレジスタ機能

レジスタ	機能	転送情報書き込みで書き戻される値
SAR	転送元アドレス	増加/減少/固定*
DAR	転送先アドレス	増加/減少/固定*
CRA	転送カウントA	CRA-1
CRB	転送カウントB	更新されません

【注】 * 転送情報のライトバックはスキップされます。

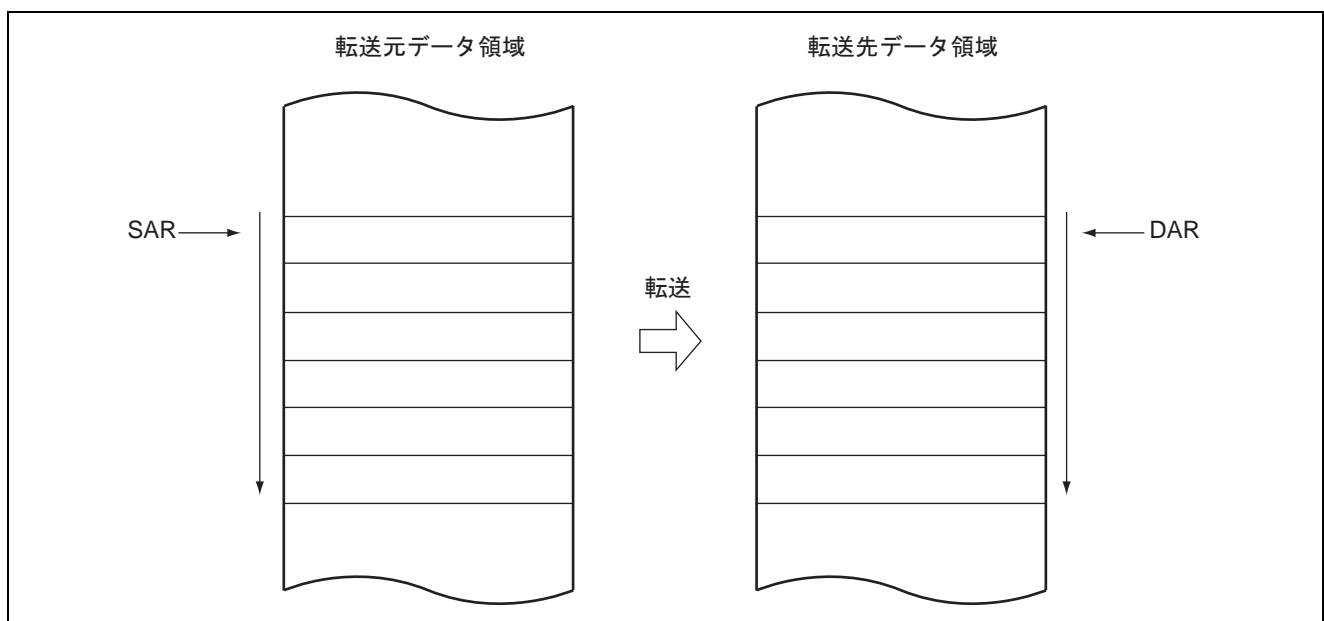


図 8.7 ノーマル転送モードのメモリマップ

8.5.5 リピート転送モード

一つの起動要因で、1バイト、1ワードまたは1ロングワードの転送を行います。MRBのDTSビットにより、転送元、転送先のいずれか一方をリピートエリアに指定します。転送回数は1~256で、指定回数の転送が終了すると、転送カウンタおよびリピートエリアに指定された方のアドレスレジスタの初期状態が回復し、転送を繰り返します。他方のアドレスレジスタは、連続してインクリメントまたはデクリメント、あるいは固定されます。リピート転送モードでは、転送カウンタ (CRAL) がH'00になるとCRALはCRAHで設定した値に更新されます。このため、転送カウンタはH'00にならないので、DISEL=0のときに、CPUへの割り込み要求は発生しません。

リピート転送モードのレジスタ機能を表8.7に、リピート転送モードのメモリマップを図8.8に示します。

表 8.7 リピート転送モードのレジスタ機能

レジスタ	機能	転送情報書き込みで書き戻される値	
		CRALが1以外するとき	CRALが1のとき
SAR	転送元アドレス	増加/減少/固定*	(DTS=0)増加/減少/固定* (DTS=1)SARの初期値
DAR	転送先アドレス	増加/減少/固定*	(DTS=0)DARの初期値 (DTS=1)増加/減少/固定*
CRAH	転送カウンタ保持	CRAH	CRAH
CRAL	転送カウンタA	CRAL-1	CRAH
CRB	転送カウンタB	更新されません	更新されません

【注】 * 転送情報のライトバックはスキップされます。

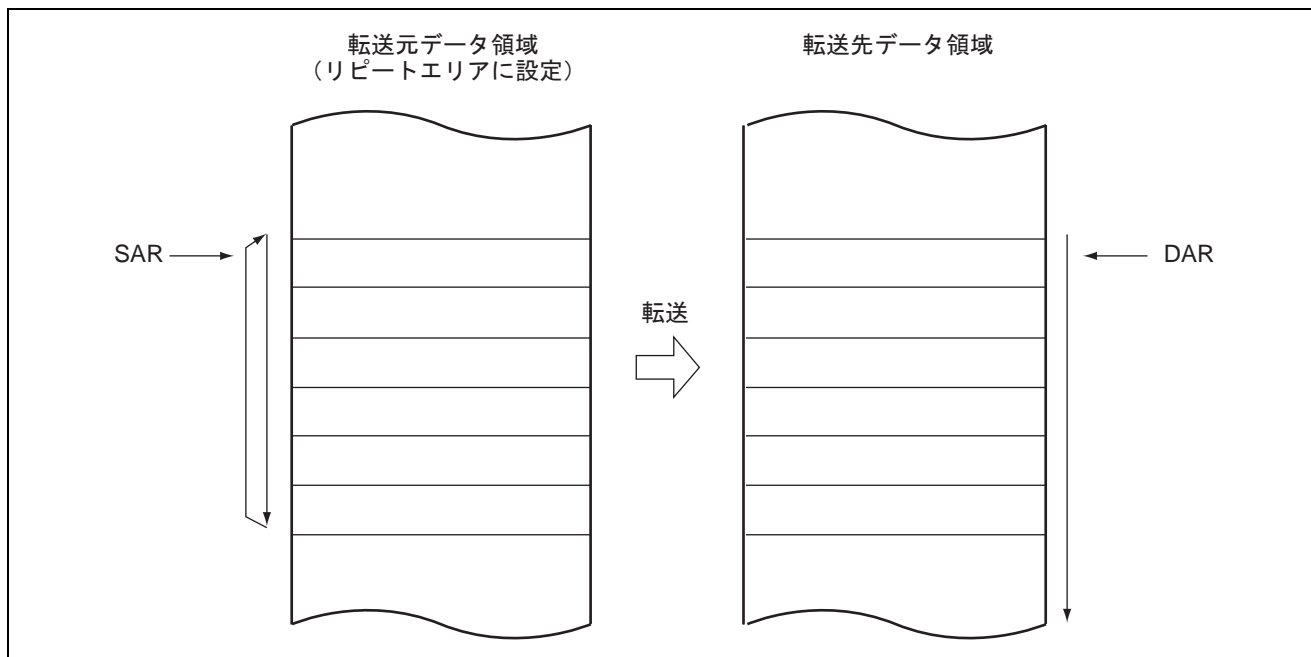


図 8.8 リピート転送モードのメモリマップ (転送元をリピートエリアに指定した場合)

8.5.6 ブロック転送モード

一つの起動要因で、1ブロックの転送を行います。MRBのDTSビットにより、転送元、転送先のいずれか一方をブロックエリアに指定します。ブロックサイズは1~256バイト(または1~256ワード、1~256ロングワード)です。1ブロックの転送が終了すると、ブロックサイズカウンタ(CRAL)とブロックエリアに指定したアドレスレジスタ(DTS=1のときSAR、DTS=0のときDAR)の初期状態が回復します。他方のアドレスレジスタは、連続してインクリメントまたはデクリメント、あるいは固定されます。転送回数は1~65536です。指定回数ブロック転送が終了すると、CPUへ割り込み要求を発生することができます。

ブロック転送モードのレジスタ機能を表8.8に、ブロック転送モードのメモリマップを図8.9に示します。

表 8.8 ブロック転送モードのレジスタ機能

レジスタ	機能	転送情報書き込みで書き戻される値
SAR	転送元アドレス	(DTS=0)増加/減少/固定* (DTS=1)SARの初期値
DAR	転送先アドレス	(DTS=0)DARの初期値 (DTS=1)増加/減少/固定*
CRAH	ブロックサイズ保持	CRAH
CRAL	ブロックサイズカウンタ	CRAH
CRB	ブロック転送回数カウンタ	CRB-1

【注】 * 転送情報のライトバックはスキップされます。

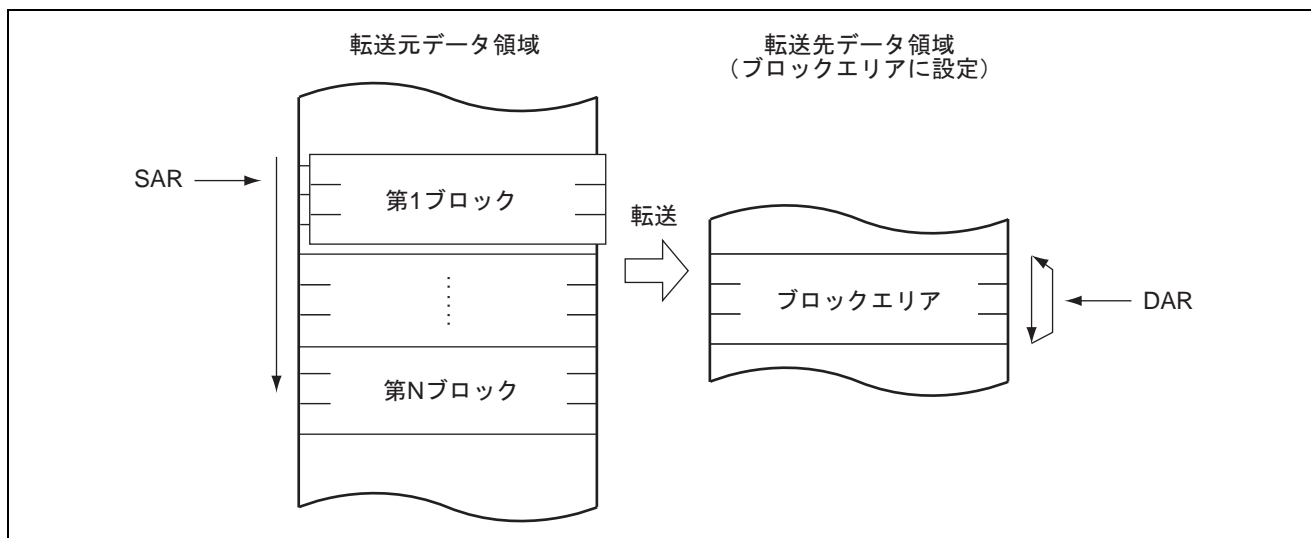


図 8.9 ブロック転送モードのメモリマップ (転送先をブロックエリアに指定した場合)

8.5.7 チェイン転送

MRB の CHNE ビットを 1 にセットすると、一つの起動要因で複数のデータ転送を連続して行うことができます。また、MRB の CHNE ビット、CHNS ビットをそれぞれ 1 にセットすると、転送カウンタ=0 のときのみチェーン転送を行います。データ転送を定義する SAR、DAR、CRA、CRB および MRA、MRB はそれぞれ独立に設定できます。チェーン転送の動作を図 8.10 に示します。

CHNE=1 に設定したデータ転送では、指定した転送回数の終了による CPU への割り込み要求や、DISEL=1 による CPU への割り込み要求は発生しません。また、CHNE=1 の転送は、起動要因となった割り込み要因フラグおよび DTCER に影響を与えません。

リピート転送モードでは、DTCCR の RCHNE ビット、MRB の CHNE、CHNS ビットをそれぞれ 1 にセットすると、転送カウンタ=1 の転送後にチェーン転送を行うことができます。

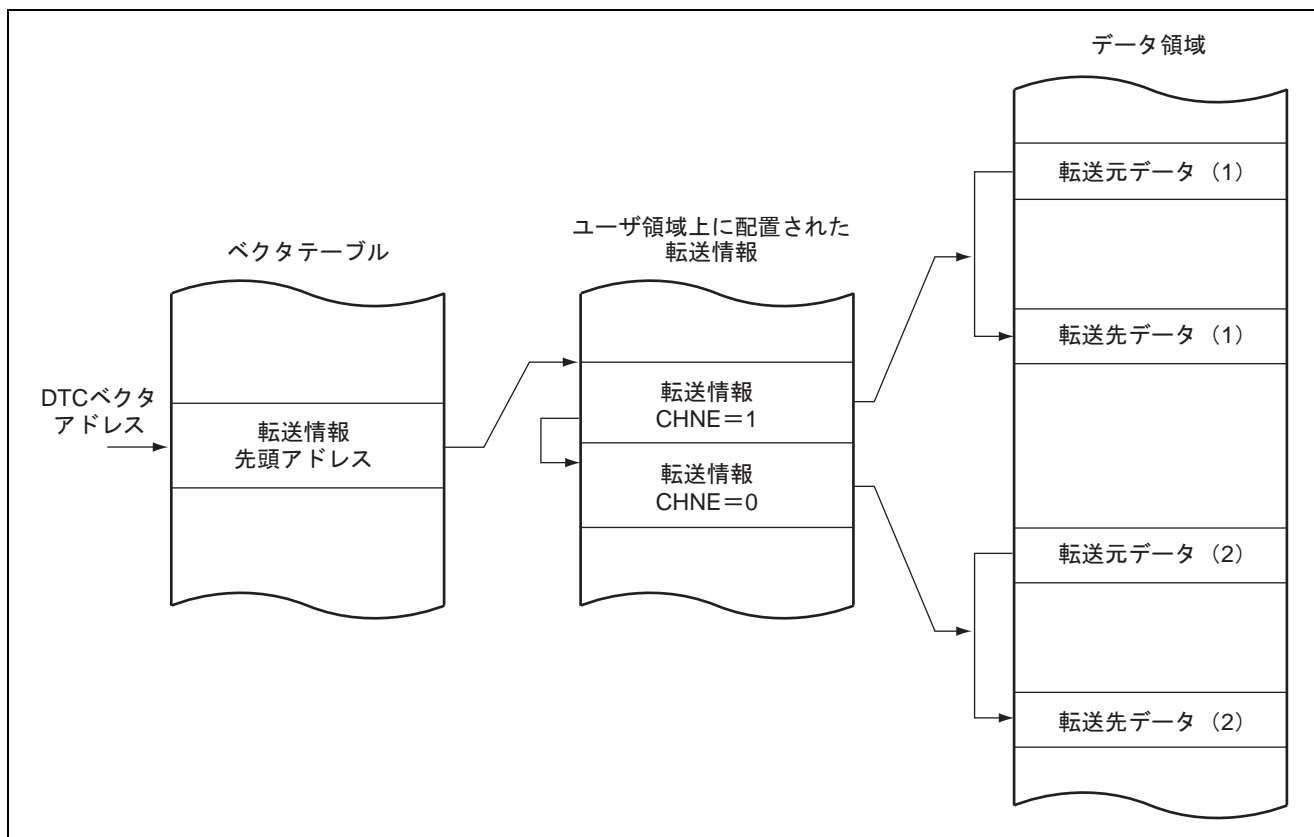


図 8.10 チェイン転送の動作

8.5.8 動作タイミング

DTC の動作タイミングを図 8.11～図 8.14 に示します。

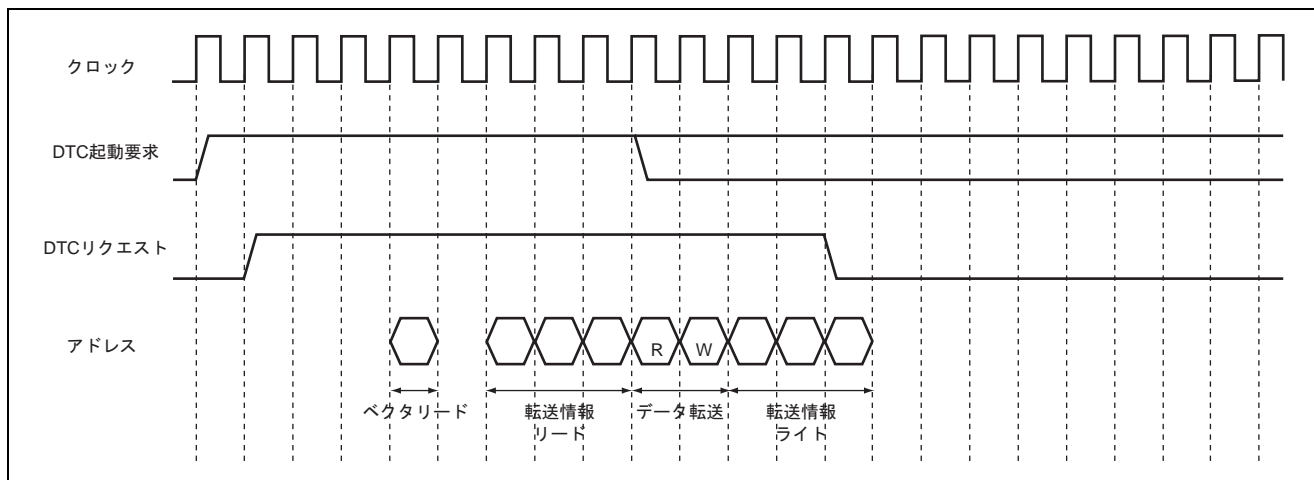


図 8.11 DTC の動作タイミング例 (ショートアドレスモード、ノーマル転送モード、リピート転送モード)

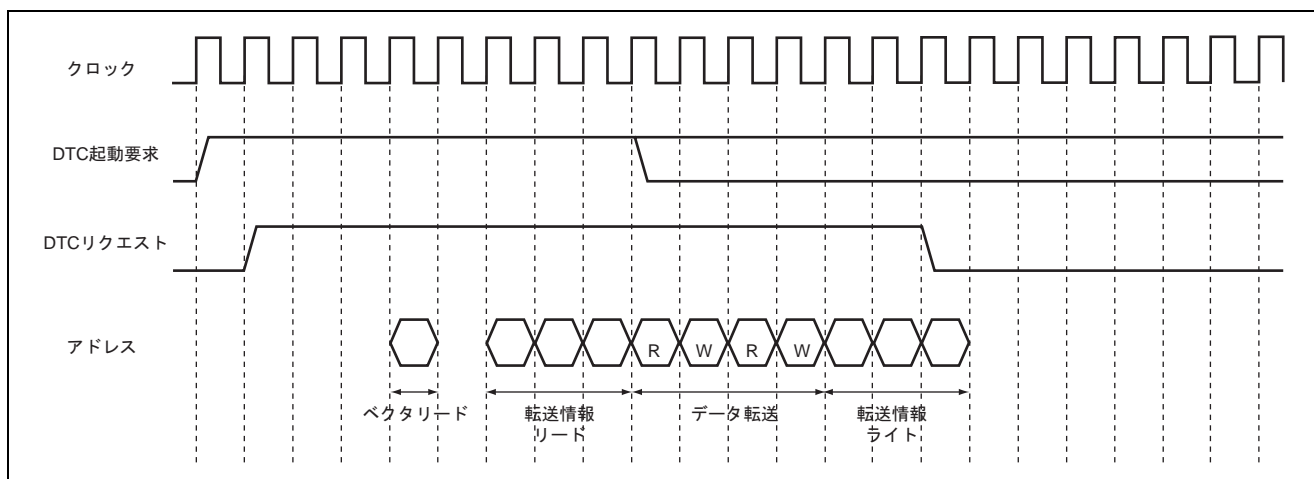


図 8.12 DTC の動作タイミング例 (ショートアドレスモード、ブロック転送モード、ブロックサイズ=2)

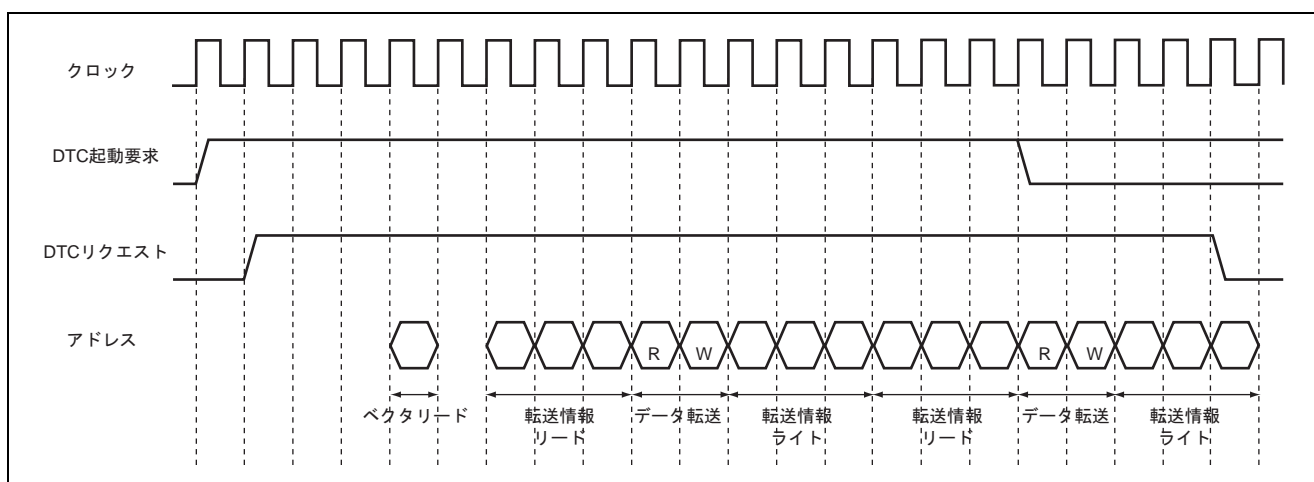


図 8.13 DTC の動作タイミング例 (ショートアドレスモード、チェーン転送)

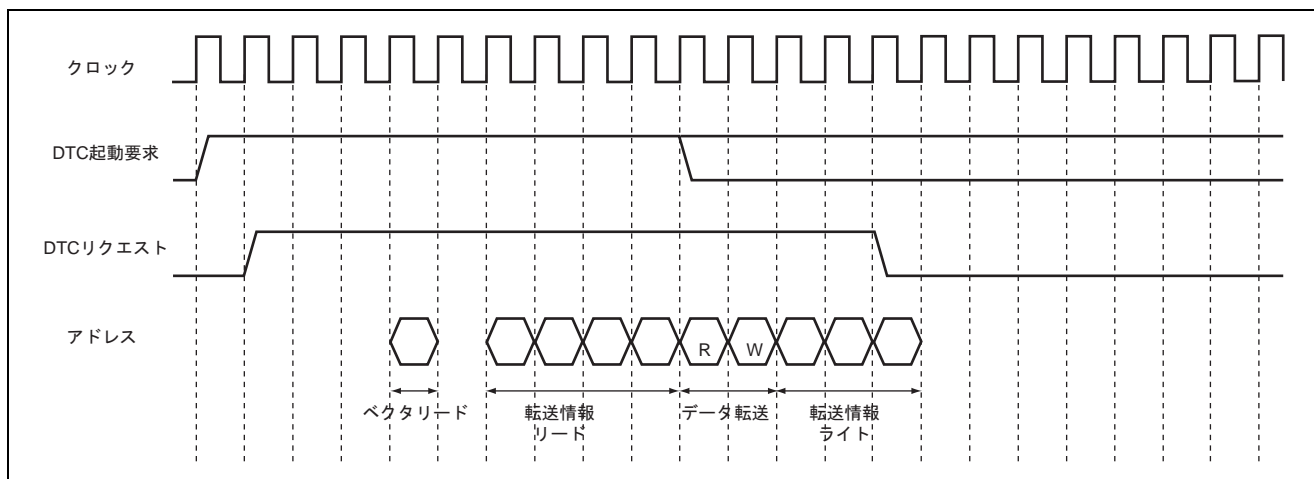


図 8.14 DTC の動作タイミング例 (フルアドレスモード、ノーマル転送モード、リピート転送モード)

8.5.9 DTC の実行ステート

DTC の 1 回のデータ転送の実行状態を表 8.9 に示します。また、実行状態に必要なステート数を表 8.10 に示します。

表 8.9 DTC の実行状態

モード	ベクタリード I		転送情報 リード J			転送情報 ライト K			データリード L			データライト M			内部動作 N	
ノーマル	1	0* ¹	4* ²	3* ³	0* ¹	3* ^{2, 3}	2* ⁴	1* ⁵	3* ⁶	2* ⁷	1	3* ⁶	2* ⁷	1	1	0* ¹
リピート	1	0* ¹	4* ²	3* ³	0* ¹	3* ^{2, 3}	2* ⁴	1* ⁵	3* ⁶	2* ⁷	1	3* ⁶	2* ⁷	1	1	0* ¹
ブロック	1	0* ¹	4* ²	3* ³	0* ¹	3* ^{2, 3}	2* ⁴	1* ⁵	3・P* ⁶	2・P* ⁷	1・P	3・P* ⁶	2・P* ⁷	1・P	1	0* ¹

- 【注】 *1 転送情報リードスキップのとき
 *2 フルアドレスモード動作のとき
 *3 ショートアドレスモード動作のとき
 *4 SAR もしくは DAR が固定モードのとき
 *5 SAR と DAR が固定モードのとき
 *6 アドレスレジスタに奇数アドレスを設定しロングワード転送をしたとき
 *7 アドレスレジスタに奇数アドレスを設定しワード転送をしたとき、もしくは、アドレスレジスタに4n+2のアドレスを設定しロングワード転送をしたとき
 P : ブロックサイズ (CRAH、CRAL の初期設定)

表 8.10 実行状態に必要なステート数

アクセス対象		内蔵 RAM	内蔵 ROM	内蔵 I/O レジスタ			外部デバイス			
				8	16	32	8		16	
バス幅		32	32	8	16	32	8		16	
アクセスステート		1	1	2	2	2	2	3	2	3
実行 状態	ベクタリード S_I	1	1	—	—	—	8	12+4m	4	6+2m
	転送情報リード S_J	1	1	—	—	—	8	12+4m	4	6+2m
	転送情報ライト S_K	1	1	—	—	—	8	12+4m	4	6+2m
	バイトデータリード S_L	1	1	2	2	2	2	3+m	2	3+m
	ワードデータリード S_L	1	1	4	2	2	4	4+2m	2	3+m
	ロングワードデータリード S_L	1	1	8	4	2	8	12+4m	4	6+2m
	バイトデータライト S_M	1	1	2	2	2	2	3+m	2	3+m
	ワードデータライト S_M	1	1	4	2	2	4	4+2m	2	3+m
	ロングワードデータライト S_M	1	1	8	4	2	8	12+4m	4	6+2m
	内部動作 S_N	1								

【記号説明】 m : ウェイト数 0~7

実行ステート数は、次の計算式で計算されます。なお、 Σ は一つの起動要因で転送する回数分 (CHNE ビットを 1 にセットした数+1) の和を示します。

$$\text{実行ステート数} = I \cdot S_I + \Sigma (J \cdot S_J + K \cdot S_K + L \cdot S_L + M \cdot S_M) + N \cdot S_N$$

8.5.10 DTC のバス権解放タイミング

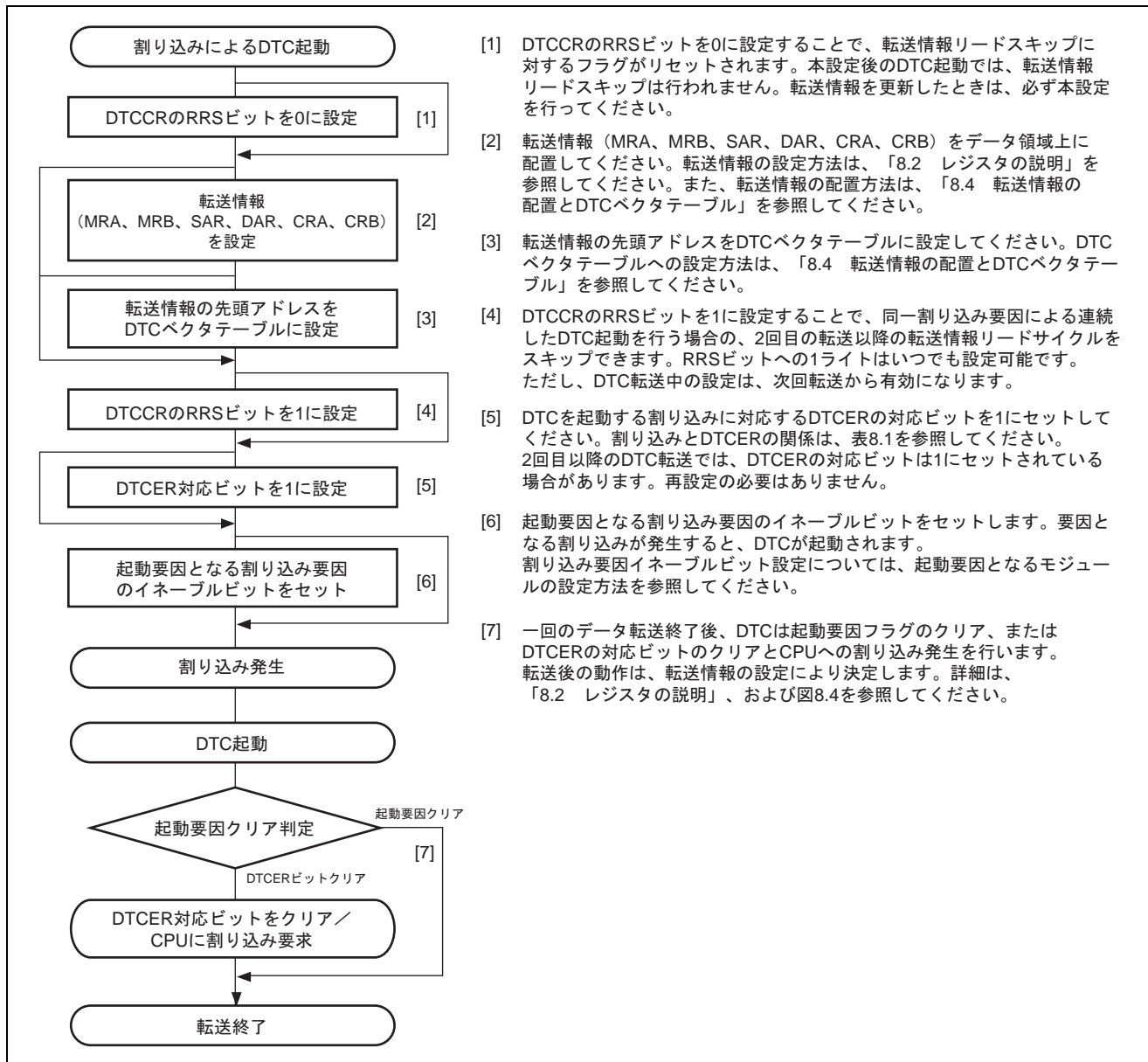
DTC は起動要求が発生するとバスアービタに対してバス権を要求します。DTC がバス権を解放するのは、ベクタリードの後、転送情報のリード後、1 回のデータ転送後、転送情報ライトバック後です。転送情報リード中、1 回のデータ転送中、転送情報ライトバック中にはバスを解放しません。

8.5.11 CPU に対する DTC の優先レベル制御

DTC の起動要因は、CPUPCR の CPUP2~CPUP0 ビットで示される CPU の優先レベルと、DTCP2~DTCP0 ビットで示される DTC の優先レベルにより、DTC と CPU との間の優先レベルを制御することができます。詳細は、「第 5 章 割り込みコントローラ」を参照してください。

8.6 割り込みによる DTC の起動

DTC の割り込み起動による使用手順を図 8.15 に示します。



- [1] DTCRRSのRRSビットを0に設定することで、転送情報リードスキップに対するフラグがリセットされます。本設定後のDTC起動では、転送情報リードスキップは行われません。転送情報を更新したときは、必ず本設定を行ってください。
- [2] 転送情報 (MRA、MRB、SAR、DAR、CRA、CRB) をデータ領域上に配置してください。転送情報の設定方法は、「8.2 レジスタの説明」を参照してください。また、転送情報の配置方法は、「8.4 転送情報の配置とDTCベクタテーブル」を参照してください。
- [3] 転送情報の先頭アドレスをDTCベクタテーブルに設定してください。DTCベクタテーブルへの設定方法は、「8.4 転送情報の配置とDTCベクタテーブル」を参照してください。
- [4] DTCRRSのRRSビットを1に設定することで、同一割り込み要因による連続したDTC起動を行う場合の、2回目の転送以降の転送情報リードサイクルをスキップできます。RRSビットへの1ライトはいつでも設定可能です。ただし、DTC転送中の設定は、次回転送から有効になります。
- [5] DTCを起動する割り込みに対応するDTCERの対応ビットを1にセットしてください。割り込みとDTCERの関係は、表8.1を参照してください。2回目以降のDTC転送では、DTCERの対応ビットは1にセットされている場合があります。再設定の必要はありません。
- [6] 起動要因となる割り込み要因のイネーブルビットをセットします。要因となる割り込みが発生すると、DTCが起動されます。割り込み要因イネーブルビット設定については、起動要因となるモジュールの設定方法を参照してください。
- [7] 一回のデータ転送終了後、DTCは起動要因フラグのクリア、またはDTCERの対応ビットのクリアとCPUへの割り込み発生を行います。転送後の動作は、転送情報の設定により決定します。詳細は、「8.2 レジスタの説明」、および図8.4を参照してください。

図 8.15 割り込みによる DTC 起動方法

8.7 DTC 使用例

8.7.1 ノーマル転送

DTC の使用例として、SCI による 128 バイトのデータ受信を行う例を示します。

1. MRAはソースアドレス固定 (SM1=SM0=0)、デスティネーションアドレスインクリメント (DM1=1、DM0=0)、ノーマル転送モード (MD1=MD0=0)、バイトサイズ (Sz1=Sz0=0) を設定します。DTSビットは任意の値とすることができます。MRBは1回の割り込みで1回のデータ転送 (CHNE=0、DISEL=0) を行います。SARはSCIのRDRのアドレス、DARはデータを格納するRAMの先頭アドレス、CRAは128 (H'0080) を設定します。CRBは任意の値とすることができます。
2. RXI割り込み用の転送情報の先頭アドレスを、DTCベクタテーブルに設定します。
3. DTCERの対応するビットを1にセットします。
4. SCIを所定の受信モードに設定します。SCRのRIEビットを1にセットし、受信完了 (RXI) 割り込みを許可します。また、SCIの受信動作中に受信エラーが発生すると、以後の受信が行われませんので、CPUが受信エラー割り込みを受け付けられるようにしてください。
5. SCIの1バイトのデータ受信が完了するごとに、SSRのRDRFフラグが1にセットされ、RXI割り込みが発生し、DTCが起動されます。DTCによって、受信データがRDRからRAMへ転送され、DARのインクリメント、CRAのデクリメントを行います。RDRFフラグは自動的に0にクリアされます。
6. 128回のデータ転送終了後、CRAが0になると、RDRFフラグは1のまま保持され、DTCEビットが0にクリアされ、CPUにRXI割り込みが要求されます。割り込み処理ルーチンで終了処理を行ってください。

8.7.2 チェイン転送

DTC のチェイン転送の例として、PPG によるパルス出力を行う例を示します。チェイン転送を使ってパルス出力データの転送と PPG 出力トリガの周期の変更を行うことができます。チェイン転送の前半で PPG の NDR へのリポート転送モード、後半で TPU の TGR へのノーマル転送モードを行います。起動要因のクリアや指定した回数の転送終了時の割り込み発生は、チェイン転送の後半 (CHNE=0 のときの転送) に限られるためです。

1. PPGのNDRへの転送の設定を行います。MRAはソースアドレスインクリメント (SM1=1、SM0=0)、デスティネーションアドレス固定 (DM1=DM0=0)、リポート転送モード (MD1=0、MD0=1)、ワードサイズ (Sz1=0、Sz0=1) を設定します。ソース側をリポート領域 (DTS=1) に設定します。MRBはチェイン転送 (CHNE=1、CHNS=0、DISEL=0) に設定します。SARはデータテーブルの先頭アドレス、DARはNDRHのアドレス、CRAH、CRALはデータテーブルサイズを設定します。CRBは任意の値とすることができます。
2. TPUのTGRへの転送の設定を行います。MRAはソースアドレスインクリメント (SM1=1、SM0=0)、デスティネーションアドレス固定 (DM1=DM0=0)、ノーマル転送モード (MD1=MD0=0)、ワードサイズ (Sz1=0、Sz0=1) を設定します。SARはデータテーブルの先頭アドレス、DARはTGRAのアドレス、CRAはデータテーブルサイズを設定します。CRBは任意の値とすることができます。
3. NDR転送用転送情報の後に連続してTPU転送用転送情報を配置します。
4. NDR転送用転送情報の先頭アドレスをDTCベクタテーブルに設定します。
5. DTCERのTGIA割り込みに対応するビットを1にセットします。
6. TIORでTGRAをアウトプットコンペアレジスタ (出力禁止) に設定し、TIERでTGIA割り込み要求を許可します。
7. PODRに出力初期値を設定し、NDRに次の出力値を設定します。DDR、NDERの出力を行うビットを1にセットします。また、PCRで出力トリガとなるTPUのコンペアマッチを選択します。

8. TSTRのCSTビットを1にセットし、TCNTのカウンタ動作を開始します。
9. TGRAのコンペアマッチが発生するたびに次の出力値がNDRへ、次の出力トリガ周期の設定値がTGRAへそれぞれ転送されます。起動要因のTGFAフラグはクリアされます。
10. 指定した回数の転送終了後 (TPU転送用CRAが0になると)、TGFAフラグは1のまま保持され、DTCEビットが0にクリアされ、CPUにTGIA割り込みが要求されます。割り込み処理ルーチンで終了処理を行ってください。

8.7.3 カウンタ=0のときのチェーン転送

カウンタが0になったときのみ第2のデータ転送を行い、第1のデータ転送の再設定を行うことによって、転送回数が256回以上のリピート転送を行うことができます。

128k バイトの入力バッファを構成する例を示します。ただし、入力バッファは下位アドレス H'0000 から始まるように設定するものとします。カウンタ=0のときのチェーン転送を図 8.16 に示します。

1. 第1のデータ転送として、入力データ用のノーマル転送モードを設定します。転送元アドレスは固定、CRA = H'0000 (65,536回)、CHNE=1、CHNS=1、DISEL=0としてください。
2. 第1のデータ転送の転送先アドレスの65,536回ごとの先頭アドレスの上位8ビットアドレスを別の領域 (ROMなど) に用意してください。たとえば、入力バッファをH'200000~H'21FFFFとするときには、H'21、H'20を用意します。
3. 第2のデータ転送として、第1のデータ転送の転送先アドレス再設定用のリピート転送モード (ソース側をリピート領域) とします。転送先は第1の転送情報領域のDARの上位8ビットとします。このときCHNE=DISEL=0としてください。上記入力バッファをH'200000~H'21FFFFとする場合には、転送カウンタ=2としてください。
4. 割り込みによって第1のデータ転送を65,536回実行します。第1のデータ転送の転送カウンタが0になると、第2のデータ転送が起動されます。第1のデータ転送の転送元アドレス上位8ビットをH'21に設定します。第1のデータ転送の転送先アドレス下位16ビットの転送カウンタは、H'0000になっています。
5. 引き続き割り込みによって第1のデータ転送を、第1のデータ転送で指定した65,536回実行します。第1のデータ転送の転送カウンタが0になると、第2のデータ転送が起動されます。第1のデータ転送の転送元アドレス上位8ビットをH'20に設定します。第1のデータ転送の転送先アドレス下位16ビットの転送カウンタはH'0000になっています。
6. 上記4.、5.を無限に繰り返します。第2のデータ転送がリピート転送モードのため、CPUには割り込みを要求しません。

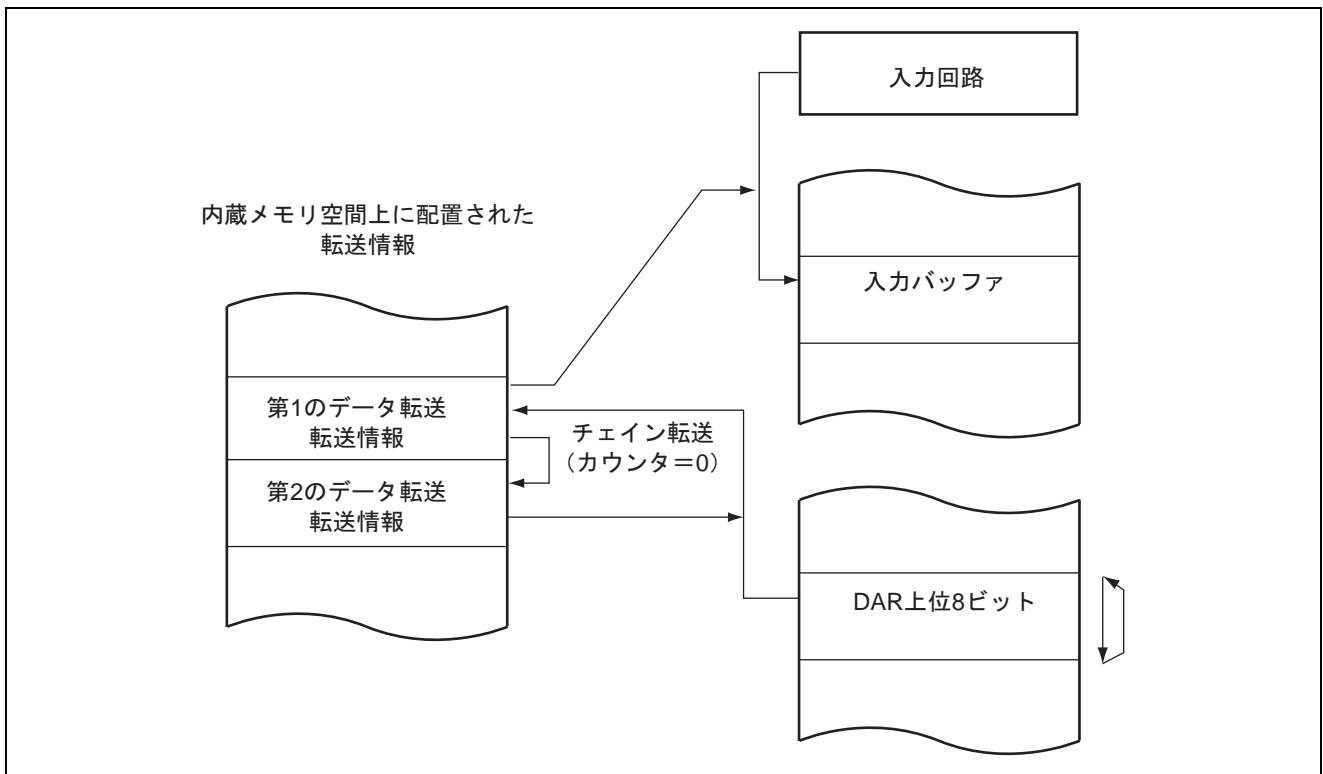


図 8.16 カウンタ=0 時のチェーン転送

8.8 割り込み要因

DTC が指定された回数 of データ転送を終了したとき、および **DISEL** ビットが 1 にセットされたデータ転送を終了したとき、CPU に対して割り込みを要求します。割り込み起動の場合、起動要因に設定した割り込みが発生します。これらの CPU に対する割り込みは CPU のマスクレベルや割り込みコントローラのプライオリティレベルの制御を受けます。

8.9 使用上の注意

8.9.1 モジュールストップモードの設定

モジュールストップコントロールレジスタにより、DTC の動作禁止/許可を設定することができます。初期値では DTC の動作許可状態です。モジュールストップモードに設定することにより、レジスタのアクセスが禁止されます。ただし、DTC が起動中はモジュールストップモードに設定できません。詳細は「第 23 章 低消費電力」を参照してください。

8.9.2 内蔵 RAM

転送情報は、内蔵 RAM に配置可能です。この場合は、SYSCR0 の RAME ビットを 0 にクリアしないでください。

8.9.3 DMAC 転送終了割り込み

DMAC 転送終了割り込みで DTC を起動したとき、転送カウンタ、DISEL ビットにかかわらず、DMDR の DTE ビットは DTC の制御を受けずにライトデータが優先されます。このため DTC の転送カウンタが 0 になった場合でも、CPU への割り込みが発生しない場合があります。

8.9.4 DTCE ビットの設定

DTCE ビットの設定は、必ず BSET、BCLR などのビット操作命令を使ってリード/ライトしてください。ただし、初期設定に限り複数の起動要因を一度に設定するときには、割り込みを禁止して当該レジスタのダメージリードを行ってからライトすることができます。

8.9.5 チェイン転送

チェイン転送が実行された場合には、連結された最後のデータ転送時に起動要因または DTCER のクリアを行います。一方、SCI および高速 A/D 変換器の割り込み/起動要因は、所定のレジスタがリード/ライトされたときにクリアされます。

したがって、これらの割り込み/起動要因によって DTC を起動する場合、DTC によるデータ転送が当該レジスタのリード/ライトを含まない場合は、割り込み/起動要因が保持されます。

8.9.6 転送情報先頭アドレス/ソースアドレス/デスティネーションアドレス

ベクタテーブルへ指定する転送情報先頭アドレスは、必ず 4n 番地を指定してください。4n 番地以外は、アドレスの最下位 2 ビットは 0 と見なしてアクセスします。

SAR、DAR へ設定するソースアドレス、デスティネーションアドレスは、アドレスとデータサイズによりバスサイクルを分割してデータ転送します。

8.9.7 転送情報の書き換え

IBCCS=1 かつ DMAC を使用している状態で、DTC 転送終了による CPU の割り込み例外処理ルーチンで DTC の転送情報を書き換える場合、IBCCS ビットを 0 にクリアし、再度 1 にセットしてから行ってください。

8.9.8 エンディアン

DTC は、ビッグエンディアン形式に対応しています。転送情報をライトする際のエンディアン形式と DTC が転送情報をリードする際のエンディアン形式は同一にしてください。

9. I/O ポート

ポート機能一覧を表 9.1 に示します。各ポートは周辺モジュールの入出力端子や割り込み入力と端子を兼用しています。入出力ポートは入出力を制御するデータディレクションレジスタ (DDR)、出力データを格納するデータレジスタ (DR)、端子の状態をリードするポートレジスタ (PORT)、入力バッファのオン/オフを制御する入力バッファコントロールレジスタ (ICR) から構成されています。ポート 4、5 には DDR、DR はありません。

ポート D、H、J、K には、入力プルアップ MOS が内蔵されており、プルアップ MOS コントロールレジスタ (PCR) で入力プルアップ MOS のオン/オフを制御できます。

ポート 1、6、A、D、H には、端子プルダウン MOS が内蔵されており、端子状態設定レジスタ (PSR) で端子プルダウン MOS のオン/オフを制御できます。

ポート 1、6、A、D、H には、ドライブ能力切り替え機能が内蔵されており、ポートドライバビリティ設定レジスタ (DSR) で出力ドライバ能力の大/小を制御できます。

ポート 1、3、6、A、D、H、J、K は、1 個の TTL 負荷と 30pF の容量負荷を駆動することができます。

すべての入出力ポートは出力時にダーリントトランジスタを駆動することができます。

ポートの $\overline{\text{IRQ}}$ 入力、TPU 入力として使用するときシュミットトリガ入力端子となります。

表 9.1 ポート機能一覧

ポート名	概要	ピ ッ ト	機能			シュミットトリガ 入力端子*	入力プル アップ MOS 機能	オープン ドレイン 出力機能
			入出力	入力	出力			
ポート 1	割り込み入力、SCI 入出力、A/D 変換器 入力、TPU 入力、 RSPI 入出力と 汎用入出力 ポート	7	P17	$\overline{\text{IRQ7-A/ADTRG1/}}$ TCKLD-B	SSLD1	$\overline{\text{IRQ7-A/TCLKD-B}}$	—	RSPI 機能時
		6	P16/SCK3	$\overline{\text{IRQ6-A/TCLKC-B}}$	—	$\overline{\text{IRQ6-A/TCLKC-B}}$		
		5	P15	$\overline{\text{IRQ5-A/}}$ TCLKB-B/RxD3	—	$\overline{\text{IRQ5-A/TCLKB-B}}$		
		4	P14	$\overline{\text{IRQ4-A/TCLKA-B}}$	TxD3	$\overline{\text{IRQ4-A/TCLKA-B}}$		
		3	P13/SSLD0	$\overline{\text{IRQ3-A/ADTRG0}}$	—	$\overline{\text{IRQ3-A}}$		
		2	P12/RSPCKD	$\overline{\text{IRQ2-A}}$	—	$\overline{\text{IRQ2-A}}$		
		1	P11/MISOD	$\overline{\text{IRQ1-A}}$	—	$\overline{\text{IRQ1-A}}$		
		0	P10/MOSID	$\overline{\text{IRQ0}}$	—	$\overline{\text{IRQ0-A}}$		

ポート名	概要	ビット	機能			シュミットトリガ 入力端子*	入力プル アップ MOS 機能	オープン ドレイン 出力機能
			入出力	入力	出力			
ポート 3	PPG 出力、 TPU 入出力、 汎用入出力ポート	7	P37/TIOCB2	TIOCA2/TCLKD-A	PO15	全入力機能	—	—
		6	P36/TIOCA2	—	PO14	全入力機能		
		5	P35/TIOCB1	TIOCA1/TCLKC-A	PO13	全入力機能		
		4	P34/TIOCA1	—	PO12	全入力機能		
		3	P33/TIOCD0	TIOCC0/TCLKB-A/ —	PO11	P33/TIOCD0/ TIOCC0/TCLKB-A		
		2	P32/TIOCC0	TCLKA-A	PO10	P32/TIOCC0/ TCLKA-A		
		1	P31/TIOCB0	TIOCA0	PO9	P31/TIOCB0/ TIOCA0		
		0	P30/TIOCA0	—	PO8	P30/TIOCA0		
ポート 4	A/D 変換器入力と 兼用汎用入力 ポート	7	—	P47/AN11	—	—	—	—
		6	—	P46/AN10	—			
		5	—	P45/AN9	—			
		4	—	P44/AN8	—			
		3	—	P43/AN15	—			
		2	—	P42/AN14	—			
		1	—	P41/AN13	—			
		0	—	P40/AN12	—			
ポート 5	割り込み入力、 A/D 変換器入力、 兼用汎用入力 ポート	7	—	P57/ $\overline{\text{IRQ}}7\text{-B}/\text{AN}7$	—	$\overline{\text{IRQ}}7\text{-B}$	—	—
		6	—	P56/ $\overline{\text{IRQ}}6\text{-B}/\text{AN}6$	—	$\overline{\text{IRQ}}6\text{-B}$		
		5	—	P55/ $\overline{\text{IRQ}}5\text{-B}/\text{AN}5$	—	$\overline{\text{IRQ}}5\text{-B}$		
		4	—	P54/ $\overline{\text{IRQ}}4\text{-B}/\text{AN}4$	—	$\overline{\text{IRQ}}4\text{-B}$		
		3	—	P53/ $\overline{\text{IRQ}}3\text{-B}/\text{AN}3$	—	$\overline{\text{IRQ}}3\text{-B}$		
		2	—	P52/ $\overline{\text{IRQ}}2\text{-B}/\text{AN}2$	—	$\overline{\text{IRQ}}2\text{-B}$		
		1	—	P51/ $\overline{\text{IRQ}}1\text{-B}/\text{AN}1$	—	$\overline{\text{IRQ}}1\text{-B}$		
		0	—	P50/ $\overline{\text{IRQ}}0\text{-B}/\text{AN}0$	—	$\overline{\text{IRQ}}0\text{-B}$		

ポート名	概要	ピ ッ ト	機能			シュミットトリガ 入力端子*	入力プ ラップ MOS 機能	オーブ ン ドレイン 出力機能
			入出力	入力	出力			
ポート 6	SCI 入出力、 RCAN 入出力 割り込み入力と 兼用汎用入出力 ポート	6	P66	$\overline{\text{IRQ14}}/\text{CRx}_1$	—	$\overline{\text{IRQ14}}$	—	—
		5	P65	$\overline{\text{IRQ13}}/\text{CRx}_0\text{-A}/$ ($\text{CRx}_0, \text{CRx}_1$)-A	$\text{CTx}_0\text{-B}/$ (CTx_0 または CTx_1)-B	$\overline{\text{IRQ13}}$		
		4	P64	$\overline{\text{IRQ12}}/\text{CRx}_0\text{-B}/$ ($\text{CRx}_0, \text{CRx}_1$)-B	$\text{CTx}_0\text{-A}/$ (CTx_0 または CTx_1)-A	$\overline{\text{IRQ12}}$		
		3	P63	$\overline{\text{IRQ11}}$	—	$\overline{\text{IRQ11}}$		
		2	P62/SCK4	$\overline{\text{IRQ10}}$	CTx_1	$\overline{\text{IRQ10}}$		
		1	P61	$\text{RxD4}/\overline{\text{IRQ9}}$		$\overline{\text{IRQ9}}$		
		0	P60	$\overline{\text{IRQ8}}$	TxD4	$\overline{\text{IRQ8}}$		
ポート A	RSPI 入出力と Bφ 出力と兼用 汎用入出力ポート	7	—	PA7	Bφ/SSCL3	—	—	
		6	PA6	—	SSCL2			
		5	PA5	—	SSCL1			
		4	PA4/SSCL0	—	—			
		3	PA3/MOSIC	—	—			
		2	PA2/MISOC	—	—			
		1	PA1/RSPCKC	—	—			
ポート D	RSPI 入出力と 汎用入出力ポート	7	PD7/SSLB0	—	—	—	○	RSPI 機能時 のみ
		6	PD6/RSPCKB	—	—			
		5	PD5/MISOB	—	—			
		4	PD4/MOSIB	—	—			
		3	PD3/SSLA0	—	—			
		2	PD2/RSOCKA	—	—			
		1	PD1/MISOA	—	—			
ポート H	RSPI 出力、 汎用入出力ポート	7	PH7	—	SSLD3	—	○	RSPI 機能時 のみ
		6	PH6	—	SSLD2			
		5	PH5	—	SSLB3			
		4	PH4	—	SSLB2			
		3	PH3	—	SSLB1			
		2	PH2	—	SSLA3			
		1	PH1	—	SSLA2			
0	PH0	—	SSLA1					

ポート名	概要	ピ ッ ト	機能			シュミットトリガ 入力端子*	入カプル アップ MOS 機能	オープン ドレイン 出力機能
			入出力	入力	出力			
ポート J	TPU 入出力、 汎用入出力ポート	7	PJ7/TIOCB8	TIOCA8/TCLKH	—	PJ7、 TIOCA8/TIOCB8/ TCLKH	○	—
		6	PJ6/TIOCA8	—	—	PJ6、TIOCA8		
		5	PJ5/TIOCB7	TIOCA7/TCLKG	—	PJ5、TIOCA7/ TIOCB7-TCLKG		
		4	PJ4/TIOCA7	—	—	PJ4、TIOCA7		
		3	PJ3/TIOCD6	TIOCC6/TCLKF	—	PJ3、 TIOCC6/TIOCD6/ TCLKF		
		2	PJ2/TIOCC6	TCLKE	—	PJ2、TIOCC6/TCLKE		
		1	PJ1/TIOCB6	TIOCA6	—	PJ1、TIOCA6/TIOCB6		
		0	PJ0/TIOCA6	—	—	PJ0、TIOCA6		
ポート K	TPU 入出力、 汎用入出力ポート	7	PK7/TIOCB11	TIOCA11	—	PK7、TIOCA11/ TIOCB11	○	—
		6	PK6/TIOCA11	—	—	PK6、TIOCA11		
		5	PK5/TIOCB10	TIOCA10	—	PK5、TIOCA10/ TIOCB10		
		4	PK4/TIOCA10	—	—	PK4、TIOCA10		
		3	PK3/TIOCD9	TIOCC9	—	PK3、 TIOCD9/TIOCC9		
		2	PK2/TIOCC9	—	—	PK2、TIOCC9		
		1	PK1/TIOCB9	TIOCA9	—	PK1、 TIOCA9/TIOCB9		
		0	PK0/TIOCA9	—	—	PK0、TIOCA9		

【注】 * シュミットトリガ入力端子以外は、CMOS 入力端子になります。

9.1 レジスタの説明

各ポートのレジスタを表 9.2 に示します。

表 9.2 各ポートのレジスタ構成

ポート	端子数	レジスタ						
		DDR	DR	PORT	ICR	PCR	DSR	PSR
ポート1	8	○	○	○	○	—	○	○
ポート3	8	○	○	○	○	—	—	—
ポート4	8	—	—	○	○	—	—	—
ポート5	8	—	—	○	○	—	—	—
ポート6* ¹	7	○	○	○	○	—	○	○
ポートA* ²	7	○	○	○	○	—	○	○
ポートD	8	○	○	○	○	○	○	○
ポートH	8	○	○	○	○	○	○	○
ポートJ	8	○	○	○	○	○	—	—
ポートK	8	○	○	○	○	○	—	—

【記号説明】 ○：レジスタあり、—：レジスタなし

【注】 *1 下位7ビットが有効で、上位1ビットはリザーブビットです。ライト時は、初期値をライトしてください。

*2 上位7ビットが有効で、下位1ビットはリザーブビットです。ライト時は、初期値をライトしてください。

I/O ポートブロック図を図 9.1 に示します。

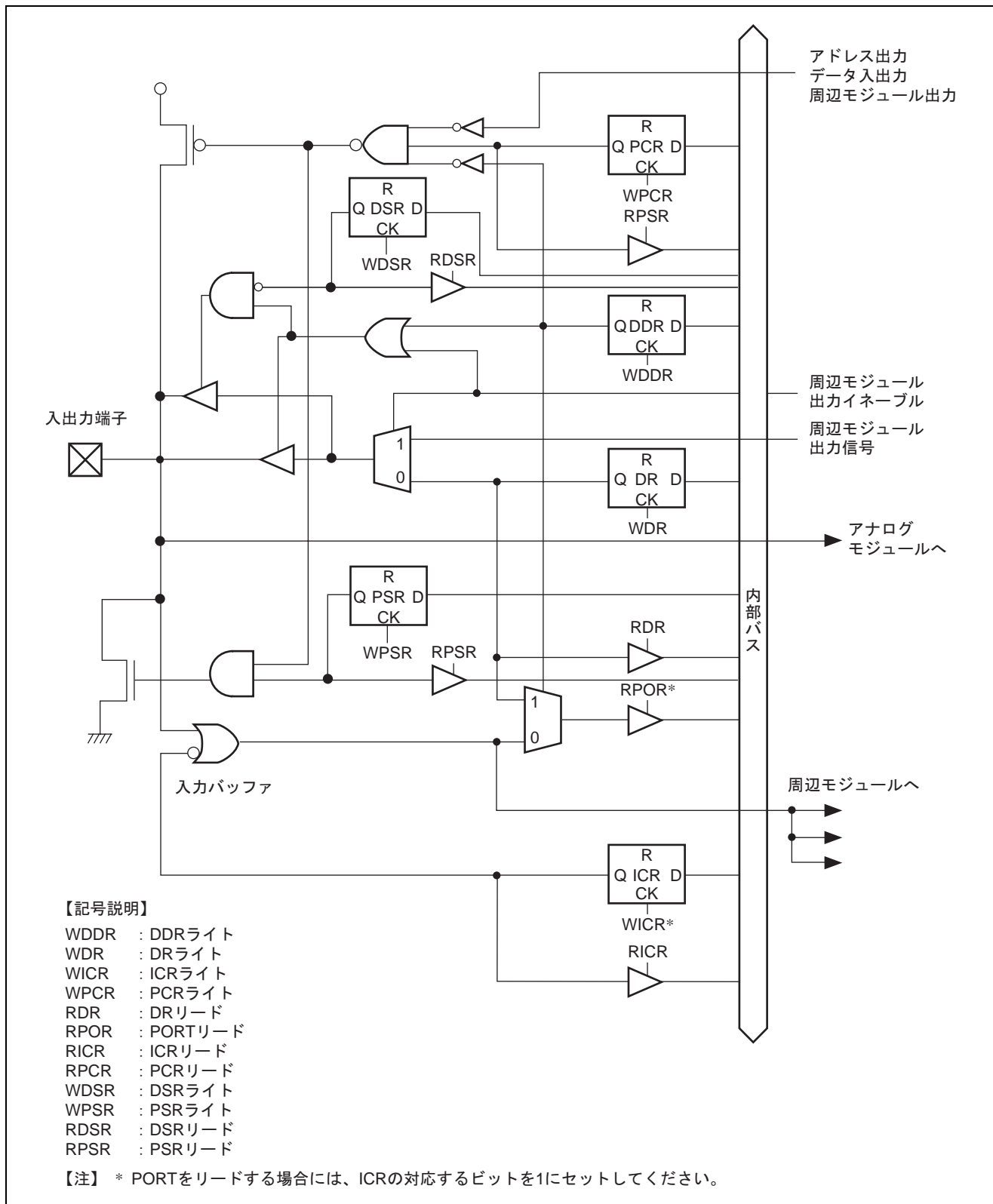


図 9.1 I/O ポートブロック図

9.1.1 データディレクションレジスタ (PnDDR) (n=1、3、6、A、D、H、J、K)

DDR は、ポートの入出力をビットごとに指定する 8 ビットのライト専用レジスタです。DDR のリードは無効です。リードすると不定値が読み出されます。DDR はライト専用のレジスタのため、ビット操作命令でライトしないでください。

汎用入出力ポートの機能が選択されているとき、DDR の当該ビットを 1 にセットすると対応する端子は出力ポートとなり、0 にクリアすると対応する端子は入力ポートになります。

ビット	7	6	5	4	3	2	1	0
ビット名	Pn7DDR	Pn6DDR	Pn5DDR	Pn4DDR	Pn3DDR	Pn2DDR	Pn1DDR	Pn0DDR
初期値:	0	0	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W

【注】 ポート6データディレクションレジスタ (P6DDR) は、下位7ビットが有効で、上位1ビットはリザーブビットです。
ポートAデータディレクションレジスタ (PADDDR) は、上位7ビットが有効で、下位1ビットはリザーブビットです。

9.1.2 データレジスタ (PnDR) (n=1、3、6、A、D、H、J、K)

DR は、汎用出力ポートとして使用する端子の出力データを格納する 8 ビットのリード/ライト可能なレジスタです。

DR の初期値は H'00 です。

ビット	7	6	5	4	3	2	1	0
ビット名	Pn7DR	Pn6DR	Pn5DR	Pn4DR	Pn3DR	Pn2DR	Pn1DR	Pn0DR
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】 ポート6データレジスタ (P6DR) は、下位7ビットが有効で、上位1ビットはリザーブビットです。
ポートAデータレジスタ (PADR) は、上位7ビットが有効で、下位1ビットはリザーブビットです。

9.1.3 ポートレジスタ (PORTn) (n=1、3、4、5、6、A、D、H、J、K)

PORT は、ポートの端子の状態を反映する 8 ビットのリード専用レジスタです。PORT のライトは無効です。

PORT をリードすると、DDR が 1 にセットされているビットは DR の値がリードされます。

PORT の初期値は不定です。ポートの端子状態により決定されます。

ビット	7	6	5	4	3	2	1	0
ビット名	Pn7	Pn6	Pn5	Pn4	Pn3	Pn2	Pn1	Pn0
初期値:	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

【注】 ポート6レジスタ (PORT6) は、下位7ビットが有効で、上位1ビットはリザーブビットです。
ポートAレジスタ (PORTA) は、上位7ビットが有効で、下位1ビットはリザーブビットです。

9.1.4 入力バッファコントロールレジスタ (PnICR) (n=1、3、4、5、6、A、D、H、J、K)

ICR は、ポートの入力バッファを制御する 8 ビットのリード/ライト可能なレジスタです。

ICR が 1 にセットされているビットは、対応する端子の入力バッファは有効になります。ICR が 0 にクリアされているビットは、対応する端子の入力バッファは無効となり、入力信号は **High** レベルに固定されます。

周辺モジュールの入力端子として使用する場合もしくは **PORT** をリードする場合には、対応するビットを 1 にセットしてください。入力として使用しない端子、およびアナログ入出力端子に対するビットへライトする場合は、初期値をライトしてください。

ICR の設定を変更するときに、端子の状態によって内部的にエッジが発生することがあります。ICR の設定の変更は、当該入力端子が使用されていないときに行ってください。たとえば、 $\overline{\text{IRQ}}$ 入力の場合、当該割り込みを禁止した状態で ICR の設定の変更を行い、割り込みコントローラの ISR の IRQF フラグを 0 にクリアし、その後当該割り込みを許可してください。ICR の設定の変更後にエッジが発生したときは、そのエッジをキャンセルするようにしてください。

ICR の初期値は H'00 です。

ビット	7	6	5	4	3	2	1	0
ビット名	Pn7ICR	Pn6ICR	Pn5ICR	Pn4ICR	Pn3ICR	Pn2ICR	Pn1ICR	Pn0ICR
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】 ポート6入力バッファコントロールレジスタ (P6ICR) は、下位7ビットが有効で、上位1ビットはリザーブビットです。ポートA入力バッファコントロールレジスタ (PAICR) は、上位7ビットが有効で、下位1ビットはリザーブビットです。

9.1.5 プルアップ MOS コントロールレジスタ (PnPCR) (n=D、H、J、K)

PCR は、ポートの入力プルアップ MOS のオン/オフを制御する 8 ビットのリード/ライト可能なレジスタです。

端子が入力状態のとき、PCR が 1 にセットされているビットに対応する端子の入力プルアップ MOS がオンします。表 9.3 に入力プルアップ MOS の状態を示します。

PCR の初期値は H'00 です。

ビット	7	6	5	4	3	2	1	0
ビット名	Pn7PCR	Pn6PCR	Pn5PCR	Pn4PCR	Pn3PCR	Pn2PCR	Pn1PCR	Pn0PCR
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

表 9.3 入力プルアップ MOS の状態

ポート	端子状態	リセット	ソフトウェア スタンバイモード	その他の動作
ポート D	周辺モジュール出力	OFF		
	ポート入力	OFF	ON/OFF	
ポート H	周辺モジュール出力	OFF		
	ポート入力	OFF	ON/OFF	
ポート J	周辺モジュール出力	OFF		
	ポート入力	OFF	ON/OFF	
ポート K	周辺モジュール出力	OFF		
	ポート入力	OFF	ON/OFF	

【記号説明】 OFF : 入力プルアップ MOS は常にオフ状態です。

ON/OFF : PCR を 1 にセットするとオン状態、PCR を 0 にクリアするとオフ状態です。

9.1.6 ドライバビリティ設定レジスタ (DSR) (n=1、6、A、D、H)

DSR は、ポートのドライブ能力を設定する 8 ビットのリード/ライト可能なレジスタです。

DSR のビットを 1 にすると、対応する端子のドライブ能力が減少します。

ビット	7	6	5	4	3	2	1	0
ビット名	Pn7DSR	Pn6DSR	Pn5DSR	Pn4DSR	Pn3DSR	Pn2DSR	Pn1DSR	Pn0DSR
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

9.1.7 端子状態設定レジスタ (PSR) (n=1、6、A、D、H)

PSR は、プルダウン/プルダウンオフを設定する 8 ビットのリード/ライト可能なレジスタです。

PSR のビットを 1 にすると、対応する端子がプルダウン状態に設定されます。

ビット	7	6	5	4	3	2	1	0
ビット名	Pn7PSR	Pn6PSR	Pn5PSR	Pn4PSR	Pn3PSR	Pn2PSR	Pn1PSR	Pn0PSR
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

9.2 出力バッファ制御

各端子の出力優先順位について説明します。

各周辺モジュールの端子は、端子名の後に「_OE」を付けて記載しています。これは（例：MIOCA4_OE）、対象となる機能の出力を有効にする設定（1）であるか、それ以外の設定（0）かを示しています。表 9.4 に各ポートの出力信号有効設定一覧を示します。当該する出力信号の詳細は、各周辺モジュールのレジスタの説明を参照してください。また、端子名の接尾に「A」、「B」が付いている端子は、ポートファンクションコントロールレジスタ（PFCR）によって端子機能を変更できます。詳細は、「9.3 ポートファンクションコントロール」を参照してください。

9.2.1 ポート 1

(1) P17/ $\overline{\text{IRQ7-A}}$ / $\overline{\text{ADTRG1}}$ /TCLKD-B/SSLD1

RSPI のレジスタの設定および P17DDR ビットにより次のように切り替わります。

モジュール名	端子機能	設定	
		RSPI	I/O ポート
		SSLD1_OE	P17DDR
RSPI	SSLD1 出力	1	—
I/O ポート	P17 出力	0	1
	P17 入力（初期値）	0	0

(2) P16/ $\overline{\text{IRQ6-A}}$ /TCLKC-B/SCK3

SCI_3 のレジスタの設定および P16DDR ビットの組み合わせにより次のように切り替わります。

モジュール名	端子機能	設定	
		SCI_3	I/O ポート
		SCK3_OE	P16DDR
SCI_3	SCK3 出力	1	—
I/O ポート	P16 出力	0	1
	P16 入力（初期値）	0	0

(3) P15/ $\overline{\text{IRQ5-A}}$ /TCLKB-B/RxD3

P15DDR ビットにより次のように切り替わります。

モジュール名	端子機能	設定	
		I/O ポート	
		P15DDR	
I/O ポート	P15 出力	1	
	P15 入力（初期値）	0	

(4) P14/ $\overline{\text{IRQ4-A}}$ /TCLKA-B/TxD3

SCI_3 のレジスタの設定および P14DDR ビットの組み合わせにより次のように切り替わります。

モジュール名	端子機能	設定	
		SCI_3	I/O ポート
		TxD3_OE	P14DDR
SCI_3	TxD3 出力	1	—
I/O ポート	P14 出力	0	1
	P14 入力 (初期値)	0	0

(5) P13/ $\overline{\text{IRQ3-A}}$ / $\overline{\text{ADTRG0}}$ /SSLD0

RSPI のレジスタ設定および P13DDR ビットにより次のように切り替わります。

モジュール名	端子機能	設定	
		RSPI	I/O ポート
		SSLD0_OE	P13DDR
RSPI	SSLD0 出力	1	—
I/O ポート	P13 出力	0	1
	P13 入力 (初期値)	0	0

(6) P12/ $\overline{\text{IRQ2-A}}$ /RSPCKD

RSPI のレジスタの設定および P12DDR ビットにより次のように切り替わります。

モジュール名	端子機能	設定	
		RSPI	I/O ポート
		RSPCKD_OE	P12DDR
RSPI	RSPCKD 出力	1	—
I/O ポート	P12 出力	0	1
	P12 入力 (初期値)	0	0

(7) P11/ $\overline{\text{IRQ1-A}}$ MISOD

RSPI、 のレジスタの設定および P11DDR ビットにより次のように切り替わります。

モジュール名	端子機能	設定	
		RSPI	I/O ポート
		MISOD_OE	P11DDR
RSPI	MISOD 出力	1	—
I/O ポート	P11 出力	0	1
	P11 入力 (初期値)	0	0

(8) P10/ $\overline{\text{IRQ0}}\text{-A/MOSID}$

RSPI のレジスタの設定および P10DDR ビットにより次のように切り替わります。

モジュール名	端子機能	設定	
		RSPI	I/O ポート
		MOSID_OE	P10DDR
RSPI	MOSID 出力	1	—
I/O ポート	P10 出力	0	1
	P10 入力 (初期値)	0	0

9.2.2 ポート 3

(1) P37/TIOCB2/TIOCA2/TCLKD-A/PO15

ポートファンクションコントロールレジスタ 9 (PFCR9)、TPU_2、PPG のレジスタの設定、および P37DDR ビットの組み合わせにより次のように切り替わります。

モジュール名	端子機能	設定		
		TPU_2	PPG	I/O ポート
		TIOCB2_OE	PO15_OE	P37DDR
TPU_2	TIOCB2 出力	1	—	—
PPG	PO15 出力	0	1	—
I/O ポート	P37 出力	0	0	1
	P37 入力 (初期値)	0	0	0

(2) P36/TIOCA2/PO14

ポートファンクションコントロールレジスタ 9 (PFCR9)、TPU_2、PPG のレジスタの設定、および P36DDR ビットの組み合わせにより次のように切り替わります。

モジュール名	端子機能	設定		
		TPU_2	PPG	I/O ポート
		TIOCA2_OE	PO14_OE	P36DDR
TPU_2	TIOCA2 出力	1	—	—
PPG	PO14 出力	0	1	—
I/O ポート	P36 出力	0	0	1
	P36 入力 (初期値)	0	0	0

(3) P35/TIOCB1/TIOCA1/TCLKC-A/PO13

ポートファンクションコントロールレジスタ 9 (PFCCR9)、TPU_1、PPG のレジスタの設定、および P35DDR ビットの組み合わせにより次のように切り替わります。

モジュール名	端子機能	設定		
		TPU_1	PPG	I/O ポート
		TIOCB1_OE	PO13_OE	P35DDR
TPU_1	TIOCB1 出力	1	—	—
PPG	PO13 出力	0	1	—
I/O ポート	P35 出力	0	0	1
	P35 入力 (初期値)	0	0	0

(4) P34/TIOCA1/PO12

ポートファンクションコントロールレジスタ 9 (PFCCR9)、TPU_1、PPG のレジスタの設定、および P34DDR ビットの組み合わせにより次のように切り替わります。

モジュール名	端子機能	設定		
		TPU_1	PPG	I/O ポート
		TIOCA1_OE	PO12_OE	P34DDR
TPU_1	TIOCA1 出力	1	—	—
PPG	PO12 出力	0	1	—
I/O ポート	P34 出力	0	0	1
	P34 入力 (初期値)	0	0	0

(5) P33/TIOCD0/TIOCC0/TCLKB-A/PO11

ポートファンクションコントロールレジスタ 9 (PFCCR9)、TPU_0、PPG のレジスタの設定、および P33DDR ビットの組み合わせにより次のように切り替わります。

モジュール名	端子機能	設定		
		TPU_0*	PPG	I/O ポート
		TIOCD0_OE	PO11_OE	P33DDR
TPU_0	TIOCD0 出力	1	—	—
PPG	PO11 出力	0	1	—
I/O ポート	P33 出力	0	0	1
	P33 入力 (初期値)	0	0	0

(6) P32/TIOCC0/TCLKA-A/PO10

ポートファンクションコントロールレジスタ 9 (PFCR9)、TPU_0、PPG のレジスタの設定、および P32DDR ビットの組み合わせにより次のように切り替わります。

モジュール名	端子機能	設定		
		TPU_0	PPG	I/O ポート
		TIOCC0_OE	PO10_OE	P32DDR
TPU_0	TIOCC0 出力	1	—	—
PPG	PO10 出力	0	1	—
I/O ポート	P32 出力	0	0	1
	P32 入力 (初期値)	0	0	0

(7) P31/TIOCB0/TIOCA0/PO9

ポートファンクションコントロールレジスタ 9 (PFCR9)、TPU_0、PPG のレジスタの設定、および P31DDR ビットの組み合わせにより次のように切り替わります。

モジュール名	端子機能	設定		
		TPU_0	PPG	I/O ポート
		TIOCB0_OE	PO9_OE	P31DDR
TPU_0	TIOCB0 出力	1	—	—
PPG	PO9 出力	0	1	—
I/O ポート	P31 出力	0	0	1
	P31 入力 (初期値)	0	0	0

(8) P30/TIOCA0/PO8

ポートファンクションコントロールレジスタ 9 (PFCR9)、TPU_0、PPG のレジスタの設定、および P30DDR ビットの組み合わせにより次のように切り替わります。

モジュール名	端子機能	設定		
		TPU_0	PPG	I/O ポート
		TIOCA0_OE	PO8_OE	P30DDR
TPU_0	TIOCA0 出力	1	—	—
PPG	PO8 出力	0	1	—
I/O ポート	P30 出力	0	0	1
	P30 入力 (初期値)	0	0	0

9.2.3 ポート 6

(1) P66/CRx_1/ $\overline{\text{IRQ14}}$

P66DDR ビットにより次のように切り替わります。

モジュール名	端子機能	設定	
		I/O ポート	
		P66DDR	
I/O ポート	P66 出力	1	
	P66 入力 (初期値)	0	

(2) P65/CRx_0-A/(CRx_0,CRx_1)-A/CTx_0-B/(CTx_0 または CTx_1)-B/ $\overline{\text{IRQ13}}$

ポートファンクションコントロールレジスタ 5 (PFCR5)、RCAN のレジスタの設定、および P65DDR ビットにより次のように切り替わります。

モジュール名	端子機能	設定		
		RCAN		I/O ポート
		CTx_0-B_OE	(CTx_0 または CTx_1)-B_OE	P65DDR
RCAN	CTx_0-B 出力	1	0	—
	(CTx_0 または CTx_1)-B 出力	0	1	—
I/O ポート	P65 出力	0	0	1
	P65 入力 (初期値)	0	0	0

(3) P64/CTx_0-A/(CTx_0 または CTx_1)-A/CRx_0-B/(CRx_0,CRx_1)-B/ $\overline{\text{IRQ12}}$

ポートファンクションコントロールレジスタ 5 (PFCR5)、RCAN のレジスタの設定、および P64DDR ビットにより次のように切り替わります。

モジュール名	端子機能	設定		
		RCAN		I/O ポート
		CTx_0-A_OE	(CTx_0 または CTx_1)-A_OE	P64DDR
RCAN	CTx_0-A 出力	1	0	—
	(CTx_0 または CTx_1)-A 出力	0	1	—
I/O ポート	P64 出力	0	0	1
	P64 入力 (初期値)	0	0	0

(4) P63/ $\overline{\text{IRQ11}}$

P63DDR ビットの組み合わせにより次のように切り替わります。

モジュール名	端子機能	設定	
		I/O ポート	
		P63DDR	
I/O ポート	P63 出力	1	
	P63 入力 (初期値)	0	

(5) P62/CTx_1/SCK4/ $\overline{\text{IRQ10}}$

ポートファンクションコントロールレジスタ 5 (PFCR5)、RCAN、SCI_4 のレジスタの設定、および P62DDR ビットの組み合わせにより次のように切り替わります。

モジュール名	端子機能	設定		
		RCAN	SCI_4	I/O ポート
		CTx_1_OE	SCK4_OE	P62DDR
RCAN	CTx_1 出力	1	—	—
SCI_4	SCK4 出力	0	1	—
I/O ポート	P62 出力	0	0	1
	P62 入力 (初期値)	0	0	0

(6) P61/RxD4/ $\overline{\text{IRQ9}}$

P61DDR ビットの組み合わせにより次のように切り替わります。

モジュール名	端子機能	設定	
		I/O ポート	
		P61DDR	
I/O ポート	P61 出力	1	
	P61 入力 (初期値)	0	

(7) P60/TxD4/ $\overline{\text{IRQ8}}$

SCI_4 のレジスタの設定および P60DDR ビットの組み合わせにより次のように切り替わります。

モジュール名	端子機能	設定	
		SCI_4	I/O ポート
		TxD4_OE	P60DDR
SCI_4	TxD4 出力	1	—
I/O ポート	P60 出力	0	1
	P60 入力 (初期値)	0	0

9.2.4 ポート A

(1) PA7/B ϕ /SSLC3

RSPI のレジスタの設定および PA7DDR ビットにより次のように切り替わります。

モジュール名	端子機能	設定	
		RSPI	I/O ポート
		SSLC3_OE	PA7DDR
RSPI	SSLC3 出力	1	—
I/O ポート	B ϕ 出力	0	1
	PA7 入力 (初期値)	0	0

(2) PA6/SSLC2

RSPI のレジスタの設定および PA6DDR ビットにより次のように切り替わります。

モジュール名	端子機能	設定	
		RSPI	I/O ポート
		SSLC2_OE	PA6DDR
RSPI	SSLC2 出力	1	—
I/O ポート	PA6 出力	0	1
	PA6 入力 (初期値)	0	0

(3) PA5/SSLC1

RSPI のレジスタの設定および PA5DDR ビットにより次のように切り替わります。

モジュール名	端子機能	設定	
		RSPI	I/O ポート
		SSLC1_OE	PA5DDR
RSPI	SSLC1 出力	1	—
I/O ポート	PA5 出力	0	1
	PA5 入力 (初期値)	0	0

(4) PA4/SSLC0

RSPI のレジスタの設定および PA4DDR ビットにより次のように切り替わります。

モジュール名	端子機能	設定	
		RSPI	I/O ポート
		SSLC0_OE	PA4DDR
RSPI	SSLC0 出力	1	—
I/O ポート	PA4 出力	0	1
	PA4 入力 (初期値)	0	0

(5) PA3/MOSIC

RSPI のレジスタの設定および PA3DDR ビットにより次のように切り替わります。

モジュール名	端子機能	設定	
		RSPI	I/O ポート
		MOSIC_OE	PA3DDR
RSPI	MOSIC 出力	1	—
I/O ポート	PA3 出力	0	1
	PA3 入力 (初期値)	0	0

(6) PA2/MISOC

RSPI のレジスタの設定および PA2DDR ビットにより次のように切り替わります。

モジュール名	端子機能	設定	
		RSPI	I/O ポート
		MISOC_OE	PA2DDR
RSPI	MISOC 出力	1	—
I/O ポート	PA2 出力	0	1
	PA2 入力 (初期値)	0	0

(7) PA1/RSPCKC

RSPI のレジスタの設定および PA1DDR ビットにより次のように切り替わります。

モジュール名	端子機能	設定	
		RSPI	I/O ポート
		RSPCKC_OE	PA1DDR
RSPI	RSPCKC 出力	1	—
I/O ポート	PA1 出力	0	1
	PA1 入力 (初期値)	0	0

9.2.5 ポート D

(1) PD7/SSLB0

RSPI のレジスタの設定および PD7DDR ビットにより次のように切り替わります。

モジュール名	端子機能	設定	
		RSPI	I/O ポート
		SSLB0_OE	PD7DDR
RSPI	SSLB0 出力	1	—
I/O ポート	PD7 出力	0	1
	PD7 入力 (初期値)	0	0

(2) PD6/RSPCKB

RSPI のレジスタの設定および PD6DDR ビットにより次のように切り替わります。

モジュール名	端子機能	設定	
		RSPI	I/O ポート
		RSPCKB_OE	PD6DDR
RSPI	RSPCKB 出力	1	—
I/O ポート	PD6 出力	0	1
	PD6 入力 (初期値)	0	0

(3) PD5/MISOB

RSPI のレジスタの設定および PD5DDR ビットにより次のように切り替わります。

モジュール名	端子機能	設定	
		RSPI	I/O ポート
		MISOB_OE	PD5DDR
RSPI	MISOB 出力	1	—
I/O ポート	PD5 出力	0	1
	PD5 入力 (初期値)	0	0

(4) PD4/MOSIB

RSPI のレジスタの設定および PD4DDR ビットにより次のように切り替わります。

モジュール名	端子機能	設定	
		RSPI	I/O ポート
		MOSIB_OE	PD4DDR
RSPI	MOSIB 出力	1	—
I/O ポート	PD4 出力	0	1
	PD4 入力 (初期値)	0	0

(5) PD3/SSLA0

RSPI のレジスタの設定および PD3DDR ビットにより次のように切り替わります。

モジュール名	端子機能	設定	
		RSPI	I/O ポート
		SSLA0_OE	PD3DDR
RSPI	SSLA0 出力	1	—
I/O ポート	PD3 出力	0	1
	PD3 入力 (初期値)	0	0

(6) PD2/RSPCKA

RSPI のレジスタの設定および PD2DDR ビットにより次のように切り替わります。

モジュール名	端子機能	設定	
		RSPI	I/O ポート
		RSPCKA_OE	PD2DDR
RSPI	RSPCKA 出力	1	—
I/O ポート	PD2 出力	0	1
	PD2 入力 (初期値)	0	0

(7) PD1/MISOA

RSPI のレジスタの設定および PD1DDR ビットにより次のように切り替わります。

モジュール名	端子機能	設定	
		RSPI	I/O ポート
		MISOA_OE	PD1DDR
RSPI	MISOA 出力	1	—
I/O ポート	PD1 出力	0	1
	PD1 入力 (初期値)	0	0

(8) PD0/MOSIA

RSPI のレジスタの設定および PD0DDR ビットにより次のように切り替わります。

モジュール名	端子機能	設定	
		RSPI	I/O ポート
		MOSIA_OE	PD0DDR
RSPI	MOSIA 出力	1	—
I/O ポート	PD0 出力	0	1
	PD0 入力 (初期値)	0	0

9.2.6 ポート H

(1) PH7/SSLD3、PH6/SSLD2、PH5/SSLB3、PH4/SSLB2、PH3/SSLB1、PH2/SSLA3、PH1/SSLA2、PH0/SSLA1

ポート H は、RSPI のレジスタの設定および PHnDDR ビットにより、次のように切り替わります。

モジュール名	端子機能	設定	
		RSPI	I/O ポート
		SSLxx_OE	PHnDDR
RSPI	SSLxx 出力	1	—
I/O ポート	PHn 出力	0	1
	PHn 入力 (初期値)	0	0

【記号説明】 n : 7~0

9.2.7 ポート J

(1) PJ7/TIOCA8/TIOCB8/TCLKH

ポートファンクションコントロールレジスタ A (PFCRA)、TPU_8 のレジスタの設定、および PJ7DDR ビットの組み合わせにより次のように切り替わります。

モジュール名	端子機能	設定	
		TPU_8	I/O ポート
		TIOCB8_OE	PJ7DDR
TPU_8	TIOCB8 出力	1	—
I/O ポート	PJ7 出力	0	1
	PJ7 入力 (初期値)	0	0

(2) PJ6/TIOCA8

ポートファンクションコントロールレジスタ A (PFCRA)、TPU_8 のレジスタの設定、および PJ6DDR ビットの組み合わせにより次のように切り替わります。

モジュール名	端子機能	設定	
		TPU_8	I/O ポート
		TIOCA8_OE	PJ6DDR
TPU_8	TIOCA8 出力	1	—
I/O ポート	PJ6 出力	0	1
	PJ6 入力 (初期値)	0	0

(3) PJ5/TIOCA7/TIOCB7/TCLKG

ポートファンクションコントロールレジスタ A (PFCRA)、TPU_7 のレジスタの設定、および PJ5DDR ビットの組み合わせにより次のように切り替わります。

モジュール名	端子機能	設定	
		TPU_7	I/O ポート
		TIOCB7_OE	PJ5DDR
TPU_7	TIOCB7 出力	1	—
I/O ポート	PJ5 出力	0	1
	PJ5 入力 (初期値)	0	0

(4) PJ4/TIOCA7

ポートファンクションコントロールレジスタ A (PFCRA)、TPU_7 のレジスタの設定、および PJ4DDR ビットの組み合わせにより次のように切り替わります。

モジュール名	端子機能	設定	
		TPU_7	I/O ポート
		TIOCA7_OE	PJ4DDR
TPU_7	TIOCA7 出力	1	—
I/O ポート	PJ4 出力	0	1
	PJ4 入力 (初期値)	0	0

(5) PJ3/TIOCC6/TIOCD6/TCLKF

ポートファンクションコントロールレジスタ A (PFCRA)、TPU_6 のレジスタの設定、および PJ3DDR ビットの組み合わせにより次のように切り替わります。

モジュール名	端子機能	設定	
		TPU_6	I/O ポート
		TIOCD6_OE	PJ3DDR
TPU_6	TIOCD6 出力	1	—
I/O ポート	PJ3 出力	0	1
	PJ3 入力 (初期値)	0	0

(6) PJ2/TIOCC6/TCLKE

ポートファンクションコントロールレジスタ A (PFCRA)、TPU_6 のレジスタの設定、および PJ2DDR ビットの組み合わせにより次のように切り替わります。

モジュール名	端子機能	設定	
		TPU_6	I/O ポート
		TIOCC6_OE	PJ2DDR
TPU_6	TIOCC6 出力	1	—
I/O ポート	PJ2 出力	0	1
	PJ2 入力 (初期値)	0	0

(7) PJ1/TIOCA6/TIOCB6

ポートファンクションコントロールレジスタ A (PFCRA)、TPU_6 のレジスタの設定、および PJ1DDR ビットの組み合わせにより次のように切り替わります。

モジュール名	端子機能	設定	
		TPU_6	I/O ポート
		TIOCB6_OE	PJ1DDR
TPU_6	TIOCB6 出力	1	—
I/O ポート	PJ1 出力	0	1
	PJ1 入力 (初期値)	0	0

(8) PJ0/TIOCA6

ポートファンクションコントロールレジスタ A (PFCRA)、TPU_6 のレジスタの設定、および PJ0DDR ビットの組み合わせにより次のように切り替わります。

モジュール名	端子機能	設定	
		TPU_6	I/O ポート
		TIOCA6_OE	PJ0DDR
TPU_6	TIOCA6 出力	1	—
I/O ポート	PJ0 出力	0	1
	PJ0 入力 (初期値)	0	0

9.2.8 ポート K

(1) PK7/TIOCB11/TIOCB11

ポートファンクションコントロールレジスタ A (PFCRA)、TPU_11 のレジスタの設定、および PK7DDR ビットの組み合わせにより次のように切り替わります。

モジュール名	端子機能	設定	
		TPU_11	I/O ポート
		TIOCB11_OE	PK7DDR
TPU_11	TIOCB11 出力	1	—
I/O ポート	PK7 出力	0	1
	PK7 入力 (初期値)	0	0

(2) PK6/TIOCA11

ポートファンクションコントロールレジスタ A (PFCRA)、TPU_11 のレジスタの設定、および PK6DDR ビットの組み合わせにより次のように切り替わります。

モジュール名	端子機能	設定	
		TPU_11	I/O ポート
		TIOCA11_OE	PK6DDR
TPU_11	TIOCA11 出力	1	—
I/O ポート	PK6 出力	0	1
	PK6 入力 (初期値)	0	0

(3) PK5/TIOCA10/TIOCB10

ポートファンクションコントロールレジスタ A (PFCRA)、TPU_10 のレジスタの設定、および PK5DDR ビットの組み合わせにより次のように切り替わります。

モジュール名	端子機能	設定	
		TPU_10	I/O ポート
		TIOCB10_OE	PK5DDR
TPU_10	TIOCB10 出力	1	—
I/O ポート	PK5 出力	0	1
	PK5 入力 (初期値)	0	0

(4) PK4/TIOCA10

ポートファンクションコントロールレジスタ A (PFCRA)、TPU_10 のレジスタの設定、および PK4DDR ビットの組み合わせにより次のように切り替わります。

モジュール名	端子機能	設定	
		TPU_10	I/O ポート
		TIOCA10_OE	PK4DDR
TPU_10	TIOCA10 出力	1	—
I/O ポート	PK4 出力	0	1
	PK4 入力 (初期値)	0	0

(5) PK3 PK3/TIOCC9/TIOCD9

ポートファンクションコントロールレジスタ A (PFCRA)、TPU_9 のレジスタの設定、および PK3DDR ビットの組み合わせにより次のように切り替わります。

モジュール名	端子機能	設定	
		TPU_9	I/O ポート
		TIOCD9_OE	PK3DDR
TPU_9	TIOCD9 出力	1	—
I/O ポート	PK3 出力	0	1
	PK3 入力 (初期値)	0	0

(6) PK2/TIOCC9

ポートファンクションコントロールレジスタ A (PFCRA)、TPU_9 のレジスタの設定、および PK2DDR ビットの組み合わせにより次のように切り替わります。

モジュール名	端子機能	設定	
		TPU_9	I/O ポート
		TIOCC9_OE	PK2DDR
TPU_9	TIOCC9 出力	1	—
I/O ポート	PK2 出力	0	1
	PK2 入力 (初期値)	0	0

(7) PK1/TIOCA9/TIOCB9

ポートファンクションコントロールレジスタ A (PFCRA)、TPU_9 のレジスタの設定、および PK1DDR ビットの組み合わせにより次のように切り替わります。

モジュール名	端子機能	設定	
		TPU_9	I/O ポート
		TIOCB9_OE	PK1DDR
TPU_9	TIOCB9 出力	1	—
I/O ポート	PK1 出力	0	1
	PK1 入力 (初期値)	0	0

(8) PK0/TIOCA9

ポートファンクションコントロールレジスタ A (PFCRA)、TPU_9 のレジスタの設定、および PK0DDR ビットの組み合わせにより次のように切り替わります。

モジュール名	端子機能	設定	
		TPU_9	I/O ポート
		TIOCA9_OE	PK0DDR
TPU_9	TIOCA9 出力	1	—
I/O ポート	PK0 出力	0	1
	PK0 入力 (初期値)	0	0

表 9.4 各ポートの出力信号有効設定一覧

Port	出力設定 信号名	出力信号名	信号選択 レジスタの設定	各内部モジュールの設定
P1	7	SSLD1_OE	SSLD1	RSPID.SPCR.MSTR=1, SPMS=0, RSPID.SPSR.MODF=0, RSPID.SPDCR.SLSEL0=0, RSPID.SPPCR.SPOM=0 (* ¹) か RSPID.SPCR.MSTR=1, SPMS=0, RSPID.SPSR.MODF=0, RSPID.SPDCR.SLSEL0=0, RSPID.SPPCR.SPOM=1, RSPID.SSLO1=0 (* ²)
	6	SCK3_OE	SCK3	SCI3.SMR.C/A=0, SCI3.SCR.CKE[1:0]=01 か SCI3.SMR.C/A=1, SCI3.SCR.CKE1=0
	4	TxD3_OE	TxD3	SCI3.SCR.TE=1
	3	SSLD0_OE	SSLD0	RSPID.SPCR.MSTR=1, MODFEN=0, SPMS=0, RSPID.SPSR.MODF=0, RSPID.SPPCR.SPOM=0 (* ¹) か RSPID.SPCR.MSTR=1, MODFEN=0, SPMS=0, RSPID.SPSR.MODF=0, RSPID.SPPCR.SPOM=1, RSPID.SSLO0=0 (* ²)
	2	RSPCKD_OE	RSPCKD	RSPID.SPCR.MSTR=1, RSPID.SPSR.MODF=0, RSPID.SPPCR.SPOM=0 (* ¹) か RSPID.SPCR.MSTR=1, RSPID.SPSR.MODF=0, RSPID.SPPCR.SPOM=1, RSPID.RSPCKO=0 (* ²)
	1	MISOD_OE	MISOD	RSPID.SPCR.SPE=1, MSTR=0, SPMS=0, RSPID.SPPCR.SPOM=0 (* ^{1*3}) か RSPID.SPCR.SPE=1, MSTR=0, SPMS=0, RSPID.SPPCR.SPOM=1, RSPID.MISOO=0 (* ^{2*3}) か RSPID.SPCR.SPE=1, MSTR=0, SPMS=1, RSPID.SPPCR.SPOM=0 (* ^{1*4}) か RSPID.SPCR.SPE=1, MSTR=0, SPMS=1, RSPID.SPPCR.SPOM=1, RSPID.MISOO=0 (* ^{2*4})
	0	MOSID_OE	MOSID	RSPID.SPCR.MSTR=1, RSPID.SPSR.MODF=0, RSPID.SPPCR.SPOM=0 (* ¹) か RSPID.SPCR.MSTR=1, RSPID.SPSR.MODF=0, RSPID.SPPCR.SPOM=1, RSPID.MOSIO=0 (* ²)
P3	7	TIOCB2_OE	TIOCB2	TPU0.TIOR_2.IOB3=0, IOB[1:0]=01/10/11, TPU0.TMDR.MD[2:0]=011, TPU0.TCR.CCLR[1:0]=00/01/11 か TPU0.TIOR_2.IOB3=0, IOB[1:0]=01/10/11, TPU0.TMDR.MD[2:0]=000/100/101/110/111
		PO15_OE	PO15	PPG.NDERH.NDER15=1
	6	TIOCA2_OE	TIOCA2	TPU0.TIOR_2.IOA3=0, IOA[1:0]=01/10/11, TPU0.TCR.CCLR[1:0]=00/10/11 か TPU0.TIOR_2.IOA3=0, IOA[1:0]=01/10/11, TPU0.TMDR.MD[2:0]=000/010/100/101/110/111
		PO14_OE	PO14	PPG.NDERH.NDER14=1
	5	TIOCB1_OE	TIOCB1	TPU0.TIOR_1.IOB3=0, IOB[1:0]=01/10/11, TPU0.TMDR.MD[2:0]=011, TPU0.TCR.CCLR[1:0]=00/01/11 か TPU0.TIOR_1.IOB3=0, IOB[1:0]=01/10/11, TPU0.TMDR.MD[2:0]=000/100/101/110/111
		PO13_OE	PO13	PPG.NDERH.NDER13=1
	4	TIOCA1_OE	TIOCA1	TPU0.TIOR_1.IOA3=0, IOA[1:0]=01/10/11, TPU0.TCR.CCLR[1:0]=00/10/11 か TPU0.TIOR_1.IOA3=0, IOA[1:0]=01/10/11, TPU0.TMDR.MD[2:0]=000/010/100/101/110/111
		PO12_OE	PO12	PPG.NDERH.NDER12=1

Port	出力設定 信号名	出力信号名	信号選択 レジスタの設定	各内部モジュールの設定	
P3	3	TIOCD0_OE	TIOCD0	TPU0.TIORL_0.IOD3=0, IOD[1:0]=01/10/11, TPU0.TMDR.MD[2:0]=011, BFB=0, TPU0.TCR.CCLR2=0 か TPU0.TIORL_0.IOD3=0, IOD[1:0]=01/10/11, TPU0.TMDR.MD[2:0]=011, BFB=0, TPU0.TCR.CCLR[1:0]=00/01/11 か TPU0.TIORL_0.IOD3=0, IOD[1:0]=01/10/11, TPU0.TMDR.MD[2:0]=000, BFB=0	
		PO11_OE	PO11	PPG.NDERH.NDER11=1	
	2	TIOCC0_OE	TIOCC0	TPU0.TIORL_0.IOC3=0, IOC[1:0]=01/10/11, TPU0.TMDR.BFA=0, TPU0.TCR.CCLR2=0 か TPU0.TIORL_0.IOC3=0, IOC[1:0]=01/10/11, TPU0.TMDR.BFA=0, TPU0.TCR.CCLR[1:0]=00/10/11 か TPU0.TIORL_0.IOC3=0, IOC[1:0]=01/10/11, TPU0.TMDR.MD[2:0]=000/010, BFA=0	
		PO10_OE	PO10	PPG.NDERH.NDER10=1	
	1	TIOCB0_OE	TIOCB0	TPU0.TIORH_0.IOB3=0, IOB[1:0]=01/10/11, TPU0.TMDR.MD[2:0]=011, TPU0.TCR.CCLR2=1 か TPU0.TIORH_0.IOB3=0, IOB[1:0]=01/10/11, TPU0.TMDR.MD[2:0]=011, TPU0.TCR.CCLR[1:0]=00/01/11 か TPU0.TIORH_0.IOB3=0, IOB[1:0]=01/10/11, TPU0.TMDR.MD[2:0]=000	
		PO9_OE	PO9	PPG.NDERH.NDER9=1	
	0	TIOCA0_OE	TIOCA0	TPU0.TIORH_0.IOA3=0, IOA[1:0]=01/10/11, TPU0.TCR.CCLR2=1 か TPU0.TIORH_0.IOA3=0, IOA[1:0]=01/10/11, TPU0.TCR.CCLR[1:0]=00/10/11 か TPU0.TIORH_0.IOA3=0, IOA[1:0]=01/10/11, TPU0.TMDR.MD[2:0]=000/010	
		PO8_OE	PO8	PPG.NDERH.NDER8=1	
P6	5	CTx0-B_OE	CTx0	PFCR5.RCANMD=0, RCAN0S=1	RCANGL.RCANMON0.RCANE=1
		(CTx0 または CTx1) -B_OE	(CTx0 または CTx1)	PFCR5.RCANMD=1, RCAN0S=1	RCANGL.RCANMON0.RCANE=1, RCANMON1.RCANE=1 (*5)
	4	CTx0-A_OE	CTx0	PFCR5.RCANMD=0, RCAN0S=0	RCANGL.RCANMON0.RCANE=1
		(CTx0 または CTx1) -A_OE	(CTx0 または CTx1)	PFCR5.RCANMD=1, RCAN0S=0	RCANGL.RCANMON0.RCANE=1, RCANMON1.RCANE=1 (*5)
	2	CTx1_OE	CTx1	PFCR5.RCANMD=0, RCAN1S=0	RCANGL.RCANMON1.RCANE=1
		SCK4_OE	SCK4		SCI4.SMR.C/A=0, SCI4.SCR.CKE[1:0]=01 か SCI4.SMR.C/A=1, SCI4.SCR.CKE1=0
0	TxD4_OE	TxD4		SCI4.SCR.TE=1	

Port	出力設定 信号名	出力信号名	信号選択 レジスタの設定	各内部モジュールの設定
PA	7	SSLC3_OE	SSLC3	RSPIC.SPCR.MSTR=1, SPMS=0, RSPIC.SPSR.MODF=0, RSPIC.SPDCR.SLSEL0=0, SLSEL1=0, RSPIC.SPPCR.SPOM=0 (* ¹) か RSPIC.SPCR.MSTR=1, SPMS=0, RSPIC.SPSR.MODF=0, RSPIC.SPDCR.SLSEL0=0, SLSEL1=0, RSPIC.SPPCR.SPOM=1, RSPIC.SSLO3=0 (* ²)
		Bφ_OE	Bφ	SYSC.SCKCR0.PSTOP1=0, PORT.PADDR.PA7DDR=1
	6	SSLC2_OE	SSLC2	RSPIC.SPCR.MSTR=1, SPMS=0, RSPIC.SPSR.MODF=0, RSPIC.SPDCR.SLSEL0=0, SLSEL1=0, RSPIC.SPPCR.SPOM=0 (* ¹) か RSPIC.SPCR.MSTR=1, SPMS=0, RSPIC.SPSR.MODF=0, RSPIC.SPDCR.SLSEL0=0, SLSEL1=0, RSPIC.SPPCR.SPOM=1, RSPIC.SSLO2=0 (* ²)
	5	SSLC1_OE	SSLC1	RSPIC.SPCR.MSTR=1, SPMS=0, RSPIC.SPSR.MODF=0, RSPIC.SPDCR.SLSEL0=0, RSPIC.SPPCR.SPOM=0 (* ¹) か RSPIC.SPCR.MSTR=1, SPMS=0, RSPIC.SPSR.MODF=0, RSPIC.SPDCR.SLSEL0=0, RSPIC.SPPCR.SPOM=1, RSPIC.SSLO1=0 (* ²)
	4	SSLC0_OE	SSLC0	RSPIC.SPCR.MSTR=1, MODFEN=0, SPMS=0, RSPIC.SPSR.MODF=0, RSPIC.SPPCR.SPOM=0 (* ¹) か RSPIC.SPCR.MSTR=1, MODFEN=0, SPMS=0, RSPIC.SPSR.MODF=0, RSPIC.SPPCR.SPOM=1, RSPIC.SSLO0=0 (* ²)
	3	MOSIC_OE	MOSIC	RSPIC.SPCR.MSTR=1, RSPIC.SPSR.MODF=0, RSPIC.SPPCR.SPOM=0 (* ¹) か RSPIC.SPCR.MSTR=1, RSPIC.SPSR.MODF=0, RSPIC.SPPCR.SPOM=1, RSPIC.MOSIO=0 (* ²)
	2	MISOC_OE	MISOC	RSPIC.SPCR.SPE=1, MSTR=0, SPMS=0, RSPIC.SPPCR.SPOM=0 (* ¹ * ³) か RSPIC.SPCR.SPE=1, MSTR=0, SPMS=0, RSPIC.SPPCR.SPOM=1, RSPIC.MISOO=0 (* ² * ³) か RSPIC.SPCR.SPE=1, MSTR=0, SPMS=1, RSPIC.SPPCR.SPOM=0 (* ¹ * ⁴) か RSPIC.SPCR.SPE=1, MSTR=0, SPMS=1, RSPIC.SPPCR.SPOM=1, RSPIC.MISOO=0 (* ² * ⁴)
	1	RSPCKC_OE	RSPCKC	RSPIC.SPCR.MSTR=1, RSPIC.SPSR.MODF=0, RSPIC.SPPCR.SPOM=0 (* ¹) か RSPIC.SPCR.MSTR=1, RSPIC.SPSR.MODF=0, RSPIC.SPPCR.SPOM=1, RSPIC.RSPCKO=0 (* ²)
PD	7	SSLB0_OE	SSLB0	RSPIB.SPCR.MSTR=1, MODFEN=0, SPMS=0, RSPIB.SPSR.MODF=0, RSPIB.SPPCR.SPOM=0 (* ¹) か RSPIB.SPCR.MSTR=1, MODFEN=0, SPMS=0, RSPIB.SPSR.MODF=0, RSPIB.SPPCR.SPOM=1, RSPIB.SSLO0=0 (* ²)
		RSPCKB_OE	RSPCKB	RSPIB.SPCR.MSTR=1, RSPIB.SPSR.MODF=0, RSPIB.SPPCR.SPOM=0 (* ¹) か RSPIB.SPCR.MSTR=1, RSPIB.SPSR.MODF=0, RSPIB.SPPCR.SPOM=1, RSPIB.RSPCKO=0 (* ²)
	5	MISOB_OE	MISOB	RSPIB.SPCR.SPE=1, MSTR=0, SPMS=0, RSPIB.SPPCR.SPOM=0 (* ¹ * ³) か RSPIB.SPCR.SPE=1, MSTR=0, SPMS=0, RSPIB.SPPCR.SPOM=1, RSPIB.MISOO=0 (* ² * ³) か RSPIB.SPCR.SPE=1, MSTR=0, SPMS=1, RSPIB.SPPCR.SPOM=0 (* ¹ * ⁴) か RSPIB.SPCR.SPE=1, MSTR=0, SPMS=1, RSPIB.SPPCR.SPOM=1, RSPIB.MISOO=0 (* ² * ⁴)
	4	MOSIB_OE	MOSIB	RSPIB.SPCR.MSTR=1, RSPIB.SPSR.MODF=0, RSPIB.SPPCR.SPOM=0 (* ¹) か RSPIB.SPCR.MSTR=1, RSPIB.SPSR.MODF=0, RSPIB.SPPCR.SPOM=1, RSPIB.MOSIO=0 (* ²)

Port		出力設定 信号名	出力信号名	信号選択 レジスタの設定	各内部モジュールの設定
PD	3	SSLA0_OE	SSLA0	PFCR8.RSPIISA=0	RSPIA.SPCR.MSTR=1, MODFEN=0, SPMS=0, RSPIA.SPSR.MODF=0, RSPIA.SPPCR.SPOM=0 (* ¹) か RSPIA.SPCR.MSTR=1, MODFEN=0, SPMS=0, RSPIA.SPSR.MODF=0, RSPIA.SPPCR.SPOM=1, RSPIA.SSLO0=0 (* ²)
	2	RSPCKA_OE	RSPCKA	PFCR8.RSPIISA=0	RSPIA.SPCR.MSTR=1, RSPIA.SPSR.MODF=0, RSPIA.SPPCR.SPOM=0 (* ¹) か RSPIA.SPCR.MSTR=1, RSPIA.SPSR.MODF=0, RSPIA.SPPCR.SPOM=1, RSPIA.RSPCKO=0 (* ²)
	1	MISOA_OE	MISOA	PFCR8.RSPIISA=0	RSPIA.SPCR.SPE=1, MSTR=0, SPMS=0, RSPIA.SPPCR.SPOM=0 (* ¹ * ³) か RSPIA.SPCR.SPE=1, MSTR=0, SPMS=0, RSPIA.SPPCR.SPOM=1, RSPIA.MISOO=0 (* ² * ³) か RSPIA.SPCR.SPE=1, MSTR=0, SPMS=1, RSPIA.SPPCR.SPOM=0 (* ¹ * ⁴) か RSPIA.SPCR.SPE=1, MSTR=0, SPMS=1, RSPIA.SPPCR.SPOM=1, RSPIA.MISOO=0 (* ² * ⁴)
	0	MOSIA_OE	MOSIA	PFCR8.RSPIISA=0	RSPIA.SPCR.MSTR=1, RSPIA.SPSR.MODF=0, RSPIA.SPPCR.SPOM=0 (* ¹) か RSPIA.SPCR.MSTR=1, RSPIA.SPSR.MODF=0, RSPIA.SPPCR.SPOM=1, RSPIA.MOSIO=0 (* ²)
PH	7	SSLD3_OE	SSLD3		RSPID.SPCR.MSTR=1, SPMS=0, RSPID.SPSR.MODF=0, RSPID.SPDCR.SLSEL0=0, SLSEL1=0, RSPID.SPPCR.SPOM=0 (* ¹) か RSPID.SPCR.MSTR=1, SPMS=0, RSPID.SPSR.MODF=0, RSPID.SPDCR.SLSEL0=0, SLSEL1=0, RSPID.SPPCR.SPOM=1, RSPID.SSLO3=0 (* ²)
	6	SSLD2_OE	SSLD2		RSPID.SPCR.MSTR=1, SPMS=0, RSPID.SPSR.MODF=0, RSPID.SPDCR.SLSEL0=0, SLSEL1=0, RSPID.SPPCR.SPOM=0 (* ¹) か RSPID.SPCR.MSTR=1, SPMS=0, RSPID.SPSR.MODF=0, RSPID.SPDCR.SLSEL0=0, SLSEL1=0, RSPID.SPPCR.SPOM=1, RSPID.SSLO2=0 (* ²)
	5	SSLB3_OE	SSLB3		RSPIB.SPCR.MSTR=1, SPMS=0, RSPIB.SPSR.MODF=0, RSPIB.SPDCR.SLSEL0=0, SLSEL1=0, RSPIB.SPPCR.SPOM=0 (* ¹) か RSPIB.SPCR.MSTR=1, SPMS=0, RSPIB.SPSR.MODF=0, RSPIB.SPDCR.SLSEL0=0, SLSEL1=0, RSPIB.SPPCR.SPOM=1, RSPIB.SSLO3=0 (* ²)
	4	SSLB2_OE	SSLB2		RSPIB.SPCR.MSTR=1, SPMS=0, RSPIB.SPSR.MODF=0, RSPIB.SPDCR.SLSEL0=0, SLSEL1=0, RSPIB.SPPCR.SPOM=0 (* ¹) か RSPIB.SPCR.MSTR=1, SPMS=0, RSPIB.SPSR.MODF=0, RSPIB.SPDCR.SLSEL0=0, SLSEL1=0, RSPIB.SPPCR.SPOM=1, RSPIB.SSLO2=0 (* ²)
	3	SSLB1_OE	SSLB1		RSPIB.SPCR.MSTR=1, SPMS=0, RSPIB.SPSR.MODF=0, RSPIB.SPDCR.SLSEL0=0, RSPIB.SPPCR.SPOM=0 (* ¹) か RSPIB.SPCR.MSTR=1, SPMS=0, RSPIB.SPSR.MODF=0, RSPIB.SPDCR.SLSEL0=0, RSPIB.SPPCR.SPOM=1, RSPIB.SSLO1=0 (* ²)
	2	SSLA3-A_OE	SSLA3	PFCR8.RSPIISA=0	RSPIA.SPCR.MSTR=1, SPMS=0, RSPIA.SPSR.MODF=0, RSPIA.SPDCR.SLSEL0=0, SLSEL1=0, RSPIA.SPPCR.SPOM=0 (* ¹) か RSPIA.SPCR.MSTR=1, SPMS=0, RSPIA.SPSR.MODF=0, RSPIA.SPDCR.SLSEL0=0, SLSEL1=0, RSPIA.SPPCR.SPOM=1, RSPIA.SSLO3=0 (* ²)

Port		出力設定 信号名	出力信号名	信号選択 レジスタの設定	各内部モジュールの設定
PD	1	SSLA2-A_OE	SSLA2	PFCCR8.RSPIISA=0	RSPIA.SPCR.MSTR=1, SPMS=0, RSPIA.SPSR.MODF=0, RSPIA.SPDCR.SLSEL0=0, SLSEL1=0, RSPIA.SPPCR.SPOM=0 (* ¹) か RSPIA.SPCR.MSTR=1, SPMS=0, RSPIA.SPSR.MODF=0, RSPIA.SPDCR.SLSEL0=0, SLSEL1=0, RSPIA.SPPCR.SPOM=1, RSPIA.SSLO2=0 (* ²)
	0	SSLA1-A_OE	SSLA1	PFCCR8.RSPIISA=0	RSPIA.SPCR.MSTR=1, SPMS=0, RSPIA.SPSR.MODF=0, RSPIA.SPDCR.SLSEL0=0, RSPIA.SPPCR.SPOM=0 (* ¹) か RSPIA.SPCR.MSTR=1, SPMS=0, RSPIA.SPSR.MODF=0, RSPIA.SPDCR.SLSEL0=0, RSPIA.SPPCR.SPOM=1, RSPIA.SSLO1=0 (* ²)
PJ	7	TIOCB8_OE	TIOCB8		TPU1.TIOR_2.IOB3=0, IOB[1:0]=01/10/11, TPU1.TMDR.MD[2:0]=011, TPU1.TCR.CCLR[1:0]=00/01/11 か TPU1.TIOR_2.IOB3=0, IOB[1:0]=01/10/11, TPU1.TMDR.MD[2:0]=000/100/101/110/111
	6	TIOCA8_OE	TIOCA8		TPU1.TIOR_2.IOA3=0, IOA[1:0]=01/10/11, TPU1.TCR.CCLR[1:0]=00/10/11 か TPU1.TIOR_2.IOA3=0, IOA[1:0]=01/10/11, TPU1.TMDR.MD[2:0]=000/010/100/101/110/111
	5	TIOCB7_OE	TIOCB7		TPU1.TIOR_1.IOB3=0, IOB[1:0]=01/10/11, TPU1.TMDR.MD[2:0]=011, TPU1.TCR.CCLR[1:0]=00/01/11 か TPU1.TIOR_1.IOB3=0, IOB[1:0]=01/10/11, TPU1.TMDR.MD[2:0]=000/100/101/110/111
	4	TIOCA7_OE	TIOCA7		TPU1.TIOR_1.IOA3=0, IOA[1:0]=01/10/11, TPU1.TCR.CCLR[1:0]=00/10/11 か TPU1.TIOR_1.IOA3=0, IOA[1:0]=01/10/11, TPU1.TMDR.MD[2:0]=000/010/100/101/110/111
	3	TIOCD6_OE	TIOCD6		TPU1.TIORL_0.IOD3=0, IOD[1:0]=01/10/11, TPU1.TMDR.MD[2:0]=011, BFB=0, TPU1.TCR.CCLR2=0 か TPU1.TIORL_0.IOD3=0, IOD[1:0]=01/10/11, TPU1.TMDR.MD[2:0]=011, BFB=0, TPU1.TCR.CCLR[1:0]=00/01/11 か TPU1.TIORL_0.IOD3=0, IOD[1:0]=01/10/11, TPU1.TMDR.MD[2:0]=000, BFB=0
	2	TIOCC6_OE	TIOCC6		TPU1.TIORL_0.IOC3=0, IOC[1:0]=01/10/11, TPU1.TMDR.BFA=0, TPU1.TCR.CCLR2=0 か TPU1.TIORL_0.IOC3=0, IOC[1:0]=01/10/11, TPU1.TMDR.BFA=0, TPU1.TCR.CCLR[1:0]=00/10/11 か TPU1.TIORL_0.IOC3=0, IOC[1:0]=01/10/11, TPU1.TMDR.MD[2:0]=000/010, BFA=0
	1	TIOCB6_OE	TIOCB6		TPU1.TIORH_0.IOB3=0, IOB[1:0]=01/10/11, TPU1.TMDR.MD[2:0]=011, TPU1.TCR.CCLR2=1 か TPU1.TIORH_0.IOB3=0, IOB[1:0]=01/10/11, TPU1.TMDR.MD[2:0]=011, TPU1.TCR.CCLR[1:0]=00/01/11 か TPU1.TIORH_0.IOB3=0, IOB[1:0]=01/10/11, TPU1.TMDR.MD[2:0]=000
	0	TIOCA6_OE	TIOCA6		TPU1.TIORH_0.IOA3=0, IOA[1:0]=01/10/11, TPU1.TCR.CCLR2=1 か TPU1.TIORH_0.IOA3=0, IOA[1:0]=01/10/11, TPU1.TCR.CCLR[1:0]=00/10/11 か TPU1.TIORH_0.IOA3=0, IOA[1:0]=01/10/11, TPU1.TMDR.MD[2:0]=000/010

Port	出力設定 信号名	出力信号名	信号選択 レジスタの設定	各内部モジュールの設定
PK	7	TIOCB11_OE	TIOCB11	TPU1.TIOR_5.IOB3=0, IOB[1:0]=01/10/11, TPU1.TMDR.MD[2:0]=011, TPU1.TCR.CCLR[1:0]=00/01/11 か TPU1.TIOR_5.IOB3=0, IOB[1:0]=01/10/11, TPU1.TMDR.MD[2:0]=000/100/101/110/111
	6	TIOCA11_OE	TIOCA11	TPU1.TIOR_5.IOA3=0, IOA[1:0]=01/10/11, TPU1.TCR.CCLR[1:0]=00/10/11 か TPU1.TIOR_5.IOA3=0, IOA[1:0]=01/10/11, TPU1.TMDR.MD[2:0]=000/010/100/101/110/111
	5	TIOCB10_OE	TIOCB10	TPU1.TIOR_4.IOB3=0, IOB[1:0]=01/10/11, TPU1.TMDR.MD[2:0]=011, TPU1.TCR.CCLR[1:0]=00/01/11 か TPU1.TIOR_4.IOB3=0, IOB[1:0]=01/10/11, TPU1.TMDR.MD[2:0]=000/100/101/110/111
	4	TIOCA10_OE	TIOCA10	TPU1.TIOR_4.IOA3=0, IOA[1:0]=01/10/11, TPU1.TCR.CCLR[1:0]=00/10/11 か TPU1.TIOR_4.IOA3=0, IOA[1:0]=01/10/11, TPU1.TMDR.MD[2:0]=000/010/100/101/110/111
	3	TIOCD9_OE	TIOCD9	TPU1.TIORL_3.IOD3=0, IOD[1:0]=01/10/11, TPU1.TMDR.MD[2:0]=011, BFB=0, TPU1.TCR.CCLR2=0 か TPU1.TIORL_3.IOD3=0, IOD[1:0]=01/10/11, TPU1.TMDR.MD[2:0]=011, BFB=0, TPU1.TCR.CCLR[1:0]=00/01/11 か TPU1.TIORL_3.IOD3=0, IOD[1:0]=01/10/11, TPU1.TMDR.MD[2:0]=000, BFB=0
	2	TIOCC9_OE	TIOCC9	TPU1.TIORL_3.IOC3=0, IOC[1:0]=01/10/11, TPU1.TMDR.BFA=0, TPU1.TCR.CCLR2=0 か TPU1.TIORL_3.IOC3=0, IOC[1:0]=01/10/11, TPU1.TMDR.BFA=0, TPU1.TCR.CCLR[1:0]=00/10/11 か TPU1.TIORL_3.IOC3=0, IOC[1:0]=01/10/11, TPU1.TMDR.MD[2:0]=000/010, BFA=0
	1	TIOCB9_OE	TIOCB9	TPU1.TIORH_3.IOB3=0, IOB[1:0]=01/10/11, TPU1.TMDR.MD[2:0]=011, TPU1.TCR.CCLR2=1 か TPU1.TIORH_3.IOB3=0, IOB[1:0]=01/10/11, TPU1.TMDR.MD[2:0]=011, TPU1.TCR.CCLR[1:0]=00/01/11 か TPU1.TIORH_3.IOB3=0, IOB[1:0]=01/10/11, TPU1.TMDR.MD[2:0]=000
	0	TIOCA9_OE	TIOCA9	TPU1.TIORH_3.IOA3=0, IOA[1:0]=01/10/11, TPU1.TCR.CCLR2=1 か TPU1.TIORH_3.IOA3=0, IOA[1:0]=01/10/11, TPU1.TCR.CCLR[1:0]=00/10/11 か TPU1.TIORH_3.IOA3=0, IOA[1:0]=01/10/11, TPU1.TMDR.MD[2:0]=000/010

- 【注】 *1 SPOM=0→CMOS 出力
 *2 SPOM=1→オープンドレイン出力
 *3 SPI 動作（4 線式）時。SSL の信号極性が SSLiP（RSPI.SSLP）で設定しているアクティブレベルのとき
 *4 クロック同期式動作（3 線式）時
 *5 1 チャンネル 64 メールボックス使用時、必ずこの設定にして使用

9.3 ポートファンクションコントローラ

ポートファンクションコントローラは、I/O ポートの制御を行います。
ポートファンクションコントローラには以下のレジスタがあります。

- ポートファンクションコントロールレジスタ5 (PFCR5)
- ポートファンクションコントロールレジスタ6 (PFCR6)
- ポートファンクションコントロールレジスタ8 (PFCR8)
- ポートファンクションコントロールレジスタ9 (PFCR9)
- ポートファンクションコントロールレジスタA (PFCRA)
- ポートファンクションコントロールレジスタB (PFCRB)
- ポートファンクションコントロールレジスタC (PFCRC)
- ポートファンクションコントロールレジスタD (PFCRD)

9.3.1 ポートファンクションコントロールレジスタ 5 (PFCR5)

PFCR5 は、RCAN の入出力端子を選択します。

ビット	7	6	5	4	3	2	1	0
ビット名	—	—	—	RCANMD	—	—	RCANS1	RCANS0
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

ビット	ビット名	初期値	R/W	説明
7~5	—	すべて0	R/W	リザーブビット 初期値を変更しないでください。
4	RCANMD	0	R/W	RCAN モードセレクト RCAN モードを選択します。 0 : 2CH×32 メールボックス モード CTx_0 を P64 または P65、CTx_1 を P62 から出力 CRx_0 を P65 または P64、CRx_1 を P66 から入力 1 : 1CH×64 メールボックス モード CTx_0&CTx_1 を P64 または P65 から出力 CRx_0、CRx_1 を P65 または P64 から入力
3~2	—	すべて0	R/W	リザーブビット 初期値を変更しないでください。
1	RCANS1	0	R/W	RCAN_1 制御端子セレクト (RCANMD=0 のときのみ有効) RCAN_1 を制御するための入出力ポートを選択します。 0 : P62 を CTx_1 として設定 1 : P62 を CTx_1 として使用禁止

ビット	ビット名	初期値	R/W	説明
0	RCANS0	0	R/W	RCAN_0 制御端子セレクト RCAN_0 を制御するための入出力ポートを選択します。 0 : P64 を CTx_0、P65 を CRx_0 として設定(RCANMD=0 のとき) P64 を CTx_0 または CTx_1、P65 を CRx_0、CRx_1 として設定 (RCANMD=1 のとき) 1 : P65 を CTx_0、P64 を CRx_0 として設定(RCANMD=0 のとき) P65 を CTx_0 または CTx_1、P64 を CRx_0、CRx_1 として設定 (RCANMD=1 のとき)

9.3.2 ポートファンクションコントロールレジスタ 6 (PFCR6)

PFCR6 は、TPU 用クロック入力端子の外部入力端子を選択します。

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

ビット	ビット名	初期値	R/W	説明
7	—	0	R/W	リザーブビット 初期値を変更しないでください。
6	—	1	R/W	リザーブビット 初期値を変更しないでください。
5	—	0	R/W	リザーブビット 初期値を変更しないでください。
4	—	0	R	リザーブビット リードすると常に 0 が読み出されます。 ライトは無効です。初期値を変更しないでください。
3	TCLKS	0	R/W	TPU (ユニット 0 用外部クロック入力端子セレクト TPU 用の外部クロックの入力端子を選択します。 0 : P32、P33、P35、P37 を外部クロック入力端子として設定 1 : P14~P17 を外部クロック入力端子として設定
2~0	—	すべて 0	R/W	リザーブビット 初期値を変更しないでください。

9.3.3 ポートファンクションコントロールレジスタ 8 (PFCR8)

PFCR8 は、HWLIN と連携する SCI チャンネルの選択、RSPIA の入出力端子を選択します。

ビット	7	6	5	4	3	2	1	0
ビット名	—	—	HLIS2	HLIS1	HLIS0	RSPISA	—	—
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	R/W

ビット	ビット名	初期値	R/W	説明
7~6	—	すべて 0	R/W	リザーブビット 初期値を変更しないでください。
5~3	HLIS2 HLIS1 HLIS0	1 0 0	R/W R/W R/W	HWLIN 制御セレクト HWLIN と連携の取れる SCI チャンネルを選択します。 000 : 設定禁止 001 : 設定禁止 010 : 設定禁止 011 : HWLIN と SCI チャンネル 3 の間で連携 100 : HWLIN と SCI チャンネル 4 の間で連携 上記以外 : 設定禁止
2	RSPISA	0	R/W	RSPI_A 制御端子セレクト RSPI_A を制御するための入出力ポートを選択します。 0 : PD0~PD3、PH0~PH2 を RSPI_A 制御端子として設定 1 : PD0~PD3、PH0~PH2 を RSPI_A 制御端子として使用禁止
1~0	—	すべて 0	R/W	リザーブビット 初期値を変更しないでください。

9.3.4 ポートファンクションコントロールレジスタ 9 (PFCCR9)

PFCCR9 は、TPU (ユニット 0) 入出力端子のマルチ機能を選択します。

ビット	7	6	5	4	3	2	1	0
ビット名	—	—	—	—	TPUMS2	TPUMS1	TPUMS0A	TPUMS0B
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

ビット	ビット名	初期値	R/W	説明
7~4	—	すべて 0	R/W	リザーブビット 初期値を変更しないでください。
3	TPUMS2	0	R/W	TPU 入出力端子マルチ機能セレクト TIOCA2 の機能を選択します。 0 : アウトプットコンペア出力、インプットキャプチャは P36 1 : インプットキャプチャ入力 P37、アウトプットコンペアは P36
2	TPUMS1	0	R/W	TPU 入出力端子マルチ機能セレクト TIOCA1 の機能を選択します。 0 : アウトプットコンペア出力、インプットキャプチャは P34 1 : インプットキャプチャ入力 P35、アウトプットコンペアは P34
1	TPUMS0A	0	R/W	TPU 入出力端子マルチ機能セレクト TIOCA0 の機能を選択します。 0 : アウトプットコンペア出力、インプットキャプチャは P30 1 : インプットキャプチャ入力 P31、アウトプットコンペアは P30
0	TPUMS0B	0	R/W	TPU 入出力端子マルチ機能セレクト TIOCC0 の機能を選択します。 0 : アウトプットコンペア出力、インプットキャプチャは P32 1 : インプットキャプチャ入力 P33、アウトプットコンペアは P32

9.3.5 ポートファンクションコントロールレジスタ A (PFCRA)

PFCRA は、TPU（ユニット 1）入出力端子のマルチ機能を選択します。

ビット	7	6	5	4	3	2	1	0
ビット名	TPUMS11	TPUMS10	TPUMS9A	TPUMS9B	TPUMS8	TPUMS7	TPUMS6A	TPUMS6B
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

ビット	ビット名	初期値	R/W	説明
7	TPUMS11	0	R/W	TPU 入出力端子マルチ機能セレクト TIOCA11 の機能を選択します。 0 : アウトプットコンペア出力、インプットキャプチャは PK6 1 : インプットキャプチャ入力は PK7、アウトプットコンペアは PK6
6	TPUMS10	0	R/W	TPU 入出力端子マルチ機能セレクト TIOCA10 の機能を選択します。 0 : アウトプットコンペア出力、インプットキャプチャは PK4 1 : インプットキャプチャ入力は PK5、アウトプットコンペアは PK4
5	TPUMS9A	0	R/W	TPU 入出力端子マルチ機能セレクト TIOCA9 の機能を選択します。 0 : アウトプットコンペア出力、インプットキャプチャは PK0 1 : インプットキャプチャ入力は PK1、アウトプットコンペアは PK0
4	TPUMS9B	0	R/W	TPU 入出力端子マルチ機能セレクト TIOCC9 の機能を選択します。 0 : アウトプットコンペア出力、インプットキャプチャは PK2 1 : インプットキャプチャ入力は PK3、アウトプットコンペアは PK2
3	TPUMS8	0	R/W	TPU 入出力端子マルチ機能セレクト TIOCA8 の機能を選択します。 0 : アウトプットコンペア出力、インプットキャプチャは PJ6 1 : インプットキャプチャ入力は PJ7、アウトプットコンペアは PJ6
2	TPUMS7	0	R/W	TPU 入出力端子マルチ機能セレクト TIOCA7 の機能を選択します。 0 : アウトプットコンペア出力、インプットキャプチャは PJ4 1 : インプットキャプチャ入力は PJ5、アウトプットコンペアは PJ4
1	TPUMS6A	0	R/W	TPU 入出力端子マルチ機能セレクト TIOCA6 の機能を選択します。 0 : アウトプットコンペア出力、インプットキャプチャは PJ0 1 : インプットキャプチャ入力は PJ1、アウトプットコンペアは PJ0

ビット	ビット名	初期値	R/W	説明
0	TPUMS6B	0	R/W	TPU 入出力端子マルチ機能セレクト TIOCC6 の機能を選択します。 0 : アウトプットコンペア出力、インプットキャプチャは PJ2 1 : インプットキャプチャ入力 PJ3、アウトプットコンペアは PJ2

9.3.6 ポートファンクションコントロールレジスタ B (PFCRB)

PFCRB は、 $\overline{\text{IRQ14}}$ ~ $\overline{\text{IRQ8}}$ 入力端子を選択します。

ビット	7	6	5	4	3	2	1	0
ビット名	—	ITS14	ITS13	ITS12	ITS11	ITS10	ITS9	ITS8
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

ビット	ビット名	初期値	R/W	説明
7	—	0	R/W	リザーブビット リードすると常に 0 が読み出されます。ライトする値は常に 0 にしてください。
6	ITS14	0	R/W	$\overline{\text{IRQ14}}$ 端子セレクト $\overline{\text{IRQ14}}$ の入力端子を選択します。 0 : P66 を $\overline{\text{IRQ14}}$ 入力端子として使用禁止 1 : P66 を $\overline{\text{IRQ14}}$ 入力端子として設定
5	ITS13	0	R/W	$\overline{\text{IRQ13}}$ 端子セレクト $\overline{\text{IRQ13}}$ の入力端子を選択します。 0 : P65 を $\overline{\text{IRQ13}}$ 入力端子として使用禁止 1 : P65 を $\overline{\text{IRQ13}}$ 入力端子として設定
4	ITS12	0	R/W	$\overline{\text{IRQ12}}$ 端子セレクト $\overline{\text{IRQ12}}$ の入力端子を選択します。 0 : P64 を $\overline{\text{IRQ12}}$ 入力端子として使用禁止 1 : P64 を $\overline{\text{IRQ12}}$ 入力端子として設定
3	ITS11	0	R/W	$\overline{\text{IRQ11}}$ 端子セレクト $\overline{\text{IRQ11}}$ の入力端子を選択します。 0 : P63 を $\overline{\text{IRQ11}}$ 入力端子として使用禁止 1 : P63 を $\overline{\text{IRQ11}}$ 入力端子として設定
2	ITS10	0	R/W	$\overline{\text{IRQ10}}$ 端子セレクト $\overline{\text{IRQ10}}$ の入力端子を選択します。 0 : P62 を $\overline{\text{IRQ10}}$ 入力端子として使用禁止 1 : P62 を $\overline{\text{IRQ10}}$ 入力端子として設定

ビット	ビット名	初期値	R/W	説明
1	ITS9	0	R/W	$\overline{\text{IRQ9}}$ 端子セレクト $\overline{\text{IRQ9}}$ の入力端子を選択します。 0 : P61 を $\overline{\text{IRQ9}}$ 入力端子として使用禁止 1 : P61 を $\overline{\text{IRQ9}}$ 入力端子として設定
0	ITS8	0	R/W	$\overline{\text{IRQ8}}$ 端子セレクト $\overline{\text{IRQ8}}$ の入力端子を選択します。 0 : P60 を $\overline{\text{IRQ8}}$ 入力端子として使用禁止 1 : P60 を $\overline{\text{IRQ8}}$ 入力端子として設定

9.3.7 ポートファンクションコントロールレジスタ C (PFCRC)

PFCRC は、 $\overline{\text{IRQ7}} \sim \overline{\text{IRQ0}}$ 入力端子を選択します。

ビット	7	6	5	4	3	2	1	0
ビット名	ITS7	ITS6	ITS5	ITS4	ITS3	ITS2	ITS1	ITS0
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

ビット	ビット名	初期値	R/W	説明
7	ITS7	0	R/W	$\overline{\text{IRQ7}}$ 端子セレクト $\overline{\text{IRQ7}}$ の入力端子を選択します。 0 : P17 を $\overline{\text{IRQ7-A}}$ 入力端子として設定 1 : P57 を $\overline{\text{IRQ7-B}}$ 入力端子として設定
6	ITS6	0	R/W	$\overline{\text{IRQ6}}$ 端子セレクト $\overline{\text{IRQ6}}$ の入力端子を選択します。 0 : P16 を $\overline{\text{IRQ6-A}}$ 入力端子として設定 1 : P56 を $\overline{\text{IRQ6-B}}$ 入力端子として設定
5	ITS5	0	R/W	$\overline{\text{IRQ5}}$ 端子セレクト $\overline{\text{IRQ5}}$ の入力端子を選択します。 0 : P15 を $\overline{\text{IRQ5-A}}$ 入力端子として設定 1 : P55 を $\overline{\text{IRQ5-B}}$ 入力端子として設定
4	ITS4	0	R/W	$\overline{\text{IRQ4}}$ 端子セレクト $\overline{\text{IRQ4}}$ の入力端子を選択します。 0 : P14 を $\overline{\text{IRQ4-A}}$ 入力端子として設定 1 : P54 を $\overline{\text{IRQ4-B}}$ 入力端子として設定

ビット	ビット名	初期値	R/W	説明
3	ITS3	0	R/W	$\overline{\text{IRQ3}}$ 端子セレクト $\overline{\text{IRQ3}}$ の入力端子を選択します。 0 : P13 を $\overline{\text{IRQ3}}$ -A 入力端子として設定 1 : P53 を $\overline{\text{IRQ3}}$ -B 入力端子として設定
2	ITS2	0	R/W	$\overline{\text{IRQ2}}$ 端子セレクト $\overline{\text{IRQ2}}$ の入力端子を選択します。 0 : P12 を $\overline{\text{IRQ2}}$ -A 入力端子として設定 1 : P52 を $\overline{\text{IRQ2}}$ -B 入力端子として設定
1	ITS1	0	R/W	$\overline{\text{IRQ1}}$ 端子セレクト $\overline{\text{IRQ1}}$ の入力端子を選択します。 0 : P11 を $\overline{\text{IRQ1}}$ -A 入力端子として設定 1 : P51 を $\overline{\text{IRQ1}}$ -B 入力端子として設定
0	ITS0	0	R/W	$\overline{\text{IRQ0}}$ 端子セレクト $\overline{\text{IRQ0}}$ の入力端子を選択します。 0 : P10 を $\overline{\text{IRQ0}}$ -A 入力端子として設定 1 : P50 を $\overline{\text{IRQ0}}$ -B 入力端子として設定

9.3.8 ポートファンクションコントロールレジスタ D (PFCRD)

PFCRD は全ポートのドライブ能力を設定します。

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

ビット	ビット名	初期値	R/W	説明
7~1	—	すべて 0	R	リザーブビット 初期値を変更しないでください。
0	DRVWDWNE	0	R/W	ドライブ能力ダウンイネーブル ドライブ能力ダウンの許可/禁止を選択します。 0 : 各ポートの DSR 設定によるドライブ能力 1 : 全ポートのドライブ能力ダウン

9.4 使用上の注意事項

9.4.1 入力バッファコントロールレジスタ（ICR）の設定

- ICRの設定を変更する場合、端子の状態によっては内部的にエッジが発生し意図しない動作をすることがあります。ICRの設定を変更する場合は、端子をHighレベルに固定した状態で行うか、当該端子に割り当てられている周辺モジュールの設定で、入力機能を無効にした状態で行ってください。
- ICRの設定で入力を有効にした場合、複数の入力機能が割り当てられている端子では、そのすべての入力機能に端子状態が反映されます。未使用の入力機能は各周辺モジュール側の設定に注意してください。
- 端子を出力端子として使用する場合、ICRの設定で入力を有効にすると出力データが端子状態として取り込まれます。出力として使用する端子は、ICRの設定で入力を無効にしてください。

9.4.2 ポートファンクションコントロールレジスタ（PFCR）の設定

- ポートファンクションコントローラは、I/Oポートの制御を行います。
各端子の入出力の設定は、入出力先を選択後に入出力を有効にしてください。
- 入力端子を変更する場合、変更前の端子レベルと変更後の端子レベルが異なると内部にエッジが発生し、意図しない動作をすることがあります。
変更する場合には以下の手順で行ってください。
 - 入力端子の変更
 - 変更しようとする端子機能に該当する周辺モジュールの設定で、入力機能を無効にする。
 - PFCR の設定により、入力端子を選択する。
 - 変更しようとする端子機能に該当する周辺モジュールの設定で、入力機能を有効にする。
- 一つの端子機能に対して、入出力先を変更する端子セレクトビット、および端子機能を有効にするイネーブルビットの両方が存在する場合、端子セレクトビットで端子の入出力先を設定した後に、イネーブルビットで端子機能を有効にしてください。
- IOポートのドライブ能力を小さくする設定は、DSRよりもPFCRDのDRV DWNEが優先され、以下のようになります。

DRV DWNE	DSR	ドライブ能力
1	—	小
0	0	大（初期値）
0	1	小

10. 16ビットタイマパルスユニット (TPU)

本 LSI は、6 チャンネルの 16 ビットタイマにより構成される 16 ビットタイマパルスユニット (TPU) を 2 ユニット (ユニット 0、ユニット 1)、合計 12 チャンネル内蔵しています。

ユニット 0 の機能一覧を表 10.1 に、ユニット 1 の機能一覧を表 10.2 に示します。また、ユニット 0 のブロック図を図 10.1 に、ユニット 1 のブロック図を図 10.2 に示します。

本文中では、ユニット 0 について説明します。ユニット 1 についても同一機能です。

10.1 特長

- 最大16本のパルス入出力が可能
- 各チャンネルごとに8種類のカウンタ入力クロックを選択可能
- 各チャンネルとも次の動作を設定可能：コンペアマッチによる波形出力、インプットキャプチャ機能、カウンタクリア動作、複数のタイマカウンタ (TCNT) への同時書き込み、コンペアマッチ/インプットキャプチャによる同時クリア、カウンタの同期動作による各レジスタの同期入出力、同期動作と組み合わせることによる最大15相のPWM出力

【注】 本 LSI では、チャンネル 3、チャンネル 4 およびチャンネル 5 の TIOCA3、TIOCB3、TIOCC3、TIOCDB3、TIOCA4、TIOCB4、TIOCA5、TIOCB5 入出力端子はありません。そのため、インプットキャプチャ入力、コンペアマッチによる波形出力 (0 出力/1 出力/トグル出力)、PWM 波形出力を行うことができません。

- チャンネル0、3はバッファ動作を設定可能
- チャンネル1、2、4、5は各々独立に位相計数モードを設定可能
- カスケード接続動作
- 内部16ビットバスによる高速アクセス
- 26種類の割り込み要因
- レジスタデータの自動転送が可能
- プログラマブルパルスジェネレータ (PPG) の出力トリガを生成可能 (ユニット0のみ)
- A/D変換器の変換スタートトリガを生成可能 (ユニット0のみ)
- モジュールストップモードの設定可能

表 10.1 TPU (ユニット0) の機能一覧

項目	チャンネル0	チャンネル1	チャンネル2	チャンネル3	チャンネル4	チャンネル5
カウントクロック	Pφ/1 Pφ/4 Pφ/16 Pφ/64 TCLKA TCLKB TCLKC TCLKD	Pφ/1 Pφ/4 Pφ/16 Pφ/64 Pφ/256 TCLKA TCLKB	Pφ/1 Pφ/4 Pφ/16 Pφ/64 Pφ/1024 TCLKA TCLKB TCLKC	Pφ/1 Pφ/4 Pφ/16 Pφ/64 Pφ/256 Pφ/1024 Pφ/4096 TCLKA	Pφ/1 Pφ/4 Pφ/16 Pφ/64 Pφ/1024 TCLKA TCLKC	Pφ/1 Pφ/4 Pφ/16 Pφ/64 Pφ/256 TCLKA TCLKC TCLKD
ジェネラルレジスタ (TGR)	TGRA_0 TGRB_0	TGRA_1 TGRB_1	TGRA_2 TGRB_2	TGRA_3 TGRB_3	TGRA_4 TGRB_4	TGRA_5 TGRB_5
ジェネラルレジスタ/ バッファレジスタ	TGRC_0 TGRD_0	—	—	TGRC_3 TGRD_3	—	—
入出力端子	TIOCA0 TIOCB0 TIOCC0 TIOCD0	TIOCA1 TIOCB1	TIOCA2 TIOCB2	TIOCA3 ^{*1} TIOCB3 ^{*1} TIOCC3 ^{*1} TIOCD3 ^{*1}	TIOCA4 ^{*1} TIOCB4 ^{*1}	TIOCA5 ^{*1} TIOCB5 ^{*1}
カウンタクリア 機能	TGR の コンペアマッチ または インプット キャプチャ	TGR の コンペアマッチ または インプット キャプチャ	TGR の コンペアマッチ または インプット キャプチャ	TGR の コンペアマッチ または インプット キャプチャ ^{*2}	TGR の コンペアマッチ または インプット キャプチャ ^{*2}	TGR の コンペアマッチ または インプット キャプチャ ^{*2}
コンペア マッチ 出力	0 出力 ○	○	○	○ ^{*1}	○ ^{*1}	○ ^{*1}
	1 出力 ○	○	○	○ ^{*1}	○ ^{*1}	○ ^{*1}
	トグル 出力 ○	○	○	○ ^{*1}	○ ^{*1}	○ ^{*1}
インプットキャプチャ 機能	○	○	○	○ ^{*2}	○ ^{*2}	○ ^{*2}
同期動作	○	○	○	○	○	○
PWM モード	○	○	○	○ ^{*1}	○ ^{*1}	○ ^{*1}
位相計数モード	—	○	○	—	○	○
バッファ動作	○	—	—	○	—	—
DTC の起動	TGR の コンペアマッチ または インプット キャプチャ	TGR の コンペアマッチ または インプット キャプチャ	TGR の コンペアマッチ または インプット キャプチャ	TGR の コンペアマッチ または インプット キャプチャ ^{*2}	TGR の コンペアマッチ または インプット キャプチャ ^{*2}	TGR の コンペアマッチ または インプット キャプチャ ^{*2}

【記号説明】

○：可能

—：不可

項 目	チャンネル0	チャンネル1	チャンネル2	チャンネル3	チャンネル4	チャンネル5
DMAC の起動	TGRA_0 の コンペアマッチ または インプット キャプチャ	TGRA_1 の コンペアマッチ または インプット キャプチャ	TGRA_2 の コンペアマッチ または インプット キャプチャ	TGRA_3 の コンペアマッチ または インプット キャプチャ ^{*2}	TGRA_4 の コンペアマッチ または インプット キャプチャ ^{*2}	TGRA_5 の コンペアマッチ または インプット キャプチャ ^{*2}
A/D 変換開始トリガ	TGRA_0 の コンペアマッチ または インプット キャプチャ	TGRA_1 の コンペアマッチ または インプット キャプチャ	TGRA_2 の コンペアマッチ または インプット キャプチャ	TGRA_3 の コンペアマッチ または インプット キャプチャ ^{*2}	TGRA_4 の コンペアマッチ または インプット キャプチャ ^{*2}	TGRA_5 の コンペアマッチ または インプット キャプチャ ^{*2}
PPG トリガ	TGRA_0、 TGRB_0 の コンペアマッチ または インプット キャプチャ	TGRA_1、 TGRB_1 の コンペアマッチ または インプット キャプチャ	TGRA_2、 TGRB_2 の コンペアマッチ または インプット キャプチャ	TGRA_3、 TGRB_3 の コンペアマッチ または インプット キャプチャ	—	—
割り込み要因	5 要因 ・コンペアマッチ ／インプット キャプチャ0A ・コンペアマッチ ／インプット キャプチャ0B ・コンペアマッチ ／インプット キャプチャ0C ・コンペアマッチ ／インプット キャプチャ0D ・オーバフロー	4 要因 ・コンペアマッチ ／インプット キャプチャ1A ・コンペアマッチ ／インプット キャプチャ1B ・オーバフロー ・アンダフロー	4 要因 ・コンペアマッチ ／インプット キャプチャ2A ・コンペアマッチ ／インプット キャプチャ2B ・オーバフロー ・アンダフロー	5 要因 ・コンペアマッチ ／インプット キャプチャ3A ・コンペアマッチ ／インプット キャプチャ3B ・コンペアマッチ ／インプット キャプチャ3C ・コンペアマッチ ／インプット キャプチャ3D ・オーバフロー	4 要因 ・コンペアマッチ ／インプット キャプチャ4A ・コンペアマッチ ／インプット キャプチャ4B ・オーバフロー ・アンダフロー	4 要因 ・コンペアマッチ ／インプット キャプチャ5A ・コンペアマッチ ／インプット キャプチャ5B ・オーバフロー ・アンダフロー

【注】 *1 本 LSI では、チャンネル 3、チャンネル 4 およびチャンネル 5 の TIOCA3、TIOCB3、TIOCC3、TIOCD3、TIOCA4、TIOCB4、TIOCA5、TIOCB5 入出力端子はありません。そのため、インプットキャプチャ入力、コンペアマッチによる波形出力（0 出力／1 出力／トグル出力）、PWM 波形出力を行うことができません。

*2 本 LSI では、チャンネル 3、チャンネル 4 およびチャンネル 5 でインプットキャプチャ機能はありません。

表 10.2 TPU (ユニット1) の機能一覧

項 目	チャンネル6	チャンネル7	チャンネル8	チャンネル9	チャンネル10	チャンネル11
カウントクロック	Pφ/1 Pφ/4 Pφ/16 Pφ/64 TCLKE TCLKF TCLKG TCLKH	Pφ/1 Pφ/4 Pφ/16 Pφ/64 Pφ/256 TCLKE TCLKF	Pφ/1 Pφ/4 Pφ/16 Pφ/64 Pφ/1024 TCLKE TCLKF TCLKG	Pφ/1 Pφ/4 Pφ/16 Pφ/64 Pφ/256 Pφ/1024 Pφ/4096 TCLKE	Pφ/1 Pφ/4 Pφ/16 Pφ/64 Pφ/1024 TCLKE TCLKG	Pφ/1 Pφ/4 Pφ/16 Pφ/64 Pφ/256 TCLKE TCLKG TCLKH
ジェネラルレジスタ (TGR)	TGRA_6 TGRB_6	TGRA_7 TGRB_7	TGRA_8 TGRB_8	TGRA_9 TGRB_9	TGRA_10 TGRB_10	TGRA_11 TGRB_11
ジェネラルレジスタ/ バッファレジスタ	TGRC_6 TGRD_6	—	—	TGRC_9 TGRD_9	—	—
入出力端子	TIOCA6 TIOCB6 TIOCC6 TIOCD6	TIOCA7 TIOCB7	TIOCA8 TIOCB8	TIOCA9 TIOCB9 TIOCC9 TIOCD9	TIOCA10 TIOCB10	TIOCA11 TIOCB11
カウンタクリア 機能	TGR の コンペアマッチ または インプット キャプチャ	TGR の コンペアマッチ または インプット キャプチャ	TGR の コンペアマッチ または インプット キャプチャ	TGR の コンペアマッチ または インプット キャプチャ	TGR の コンペアマッチ または インプット キャプチャ	TGR の コンペアマッチ または インプット キャプチャ
コンペア マッチ 出力	0 出力 1 出力 トグル 出力	○ ○ ○	○ ○ ○	○ ○ ○	○ ○ ○	○ ○ ○
インプットキャプチャ 機能	○	○	○	○	○	○
同期動作	○	○	○	○	○	○
PWM モード	○	○	○	○	○	○
位相計数モード	—	○	○	—	○	○
バッファ動作	○	—	—	○	—	—
DTC の起動	TGR の コンペアマッチ または インプット キャプチャ	TGR の コンペアマッチ または インプット キャプチャ	TGR の コンペアマッチ または インプット キャプチャ	TGR の コンペアマッチ または インプット キャプチャ	TGR の コンペアマッチ または インプット キャプチャ	TGR の コンペアマッチ または インプット キャプチャ

【記号説明】

○ : 可能

— : 不可

項 目	チャンネル 6	チャンネル 7	チャンネル 8	チャンネル 9	チャンネル 10	チャンネル 11
DMAC の起動	TGRA_6 の コンペアマッチ または インプット キャプチャ	TGRA_7 の コンペアマッチ または インプット キャプチャ	TGRA_8 の コンペアマッチ または インプット キャプチャ	TGRA_9 の コンペアマッチ または インプット キャプチャ	TGRA_10 の コンペアマッチ または インプット キャプチャ	TGRA_11 の コンペアマッチ または インプット キャプチャ
割り込み要因	5 要因 <ul style="list-style-type: none"> ・コンペアマッチ ／インプット キャプチャ 6A ・コンペアマッチ ／インプット キャプチャ 6B ・コンペアマッチ ／インプット キャプチャ 6C ・コンペアマッチ ／インプット キャプチャ 6D ・オーバフロー 	4 要因 <ul style="list-style-type: none"> ・コンペアマッチ ／インプット キャプチャ 7A ・コンペアマッチ ／インプット キャプチャ 7B ・オーバフロー ・アンダフロー 	4 要因 <ul style="list-style-type: none"> ・コンペアマッチ ／インプット キャプチャ 8A ・コンペアマッチ ／インプット キャプチャ 8B ・オーバフロー ・アンダフロー 	5 要因 <ul style="list-style-type: none"> ・コンペアマッチ ／インプット キャプチャ 9A ・コンペアマッチ ／インプット キャプチャ 9B ・コンペアマッチ ／インプット キャプチャ 9C ・コンペアマッチ ／インプット キャプチャ 9D ・オーバフロー 	4 要因 <ul style="list-style-type: none"> ・コンペアマッチ ／インプット キャプチャ 10A ・コンペアマッチ ／インプット キャプチャ 10B ・オーバフロー ・アンダフロー 	4 要因 <ul style="list-style-type: none"> ・コンペアマッチ ／インプット キャプチャ 11A ・コンペアマッチ ／インプット キャプチャ 11B ・オーバフロー ・アンダフロー

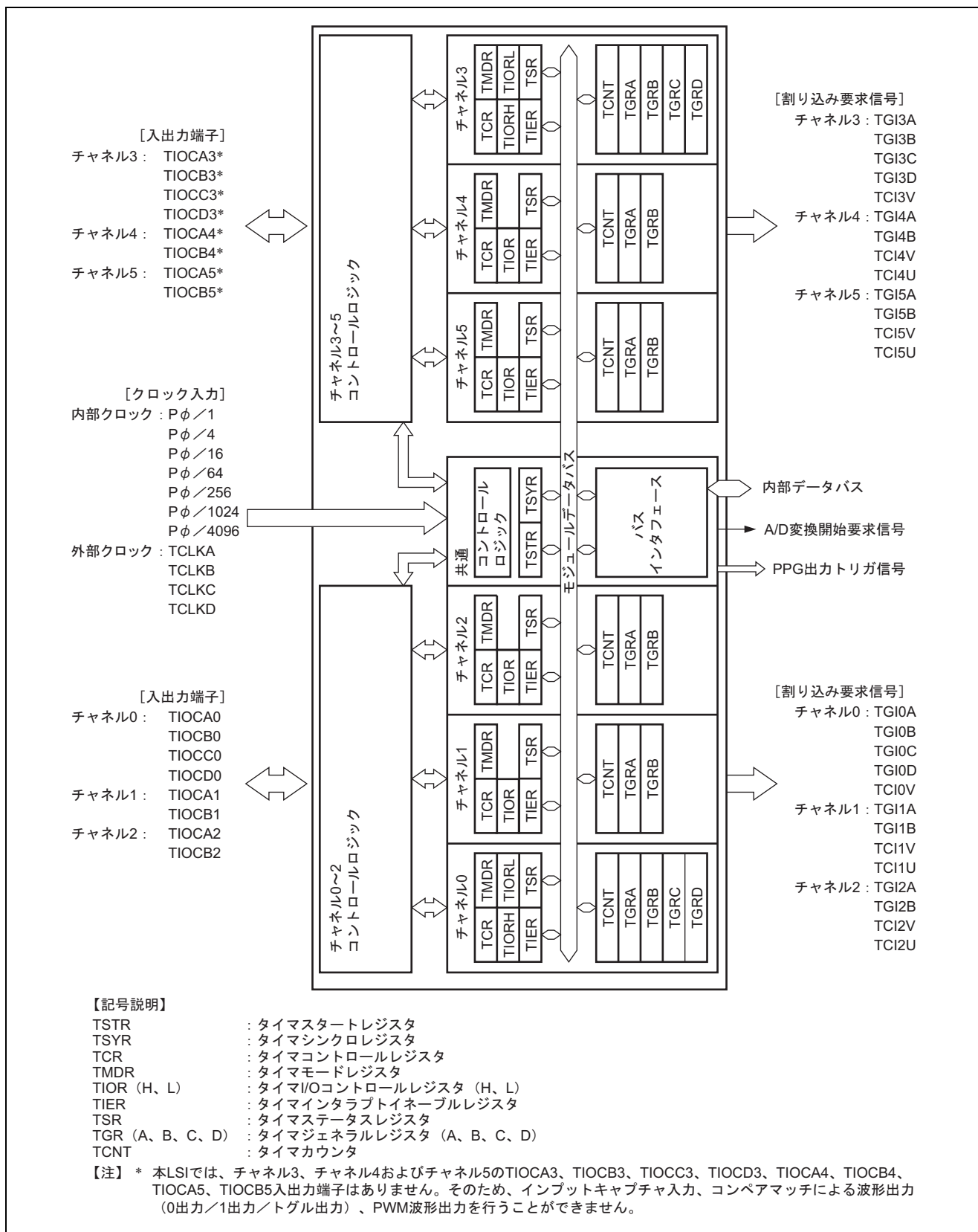


図 10.1 TPU (ユニット 0) のブロック図

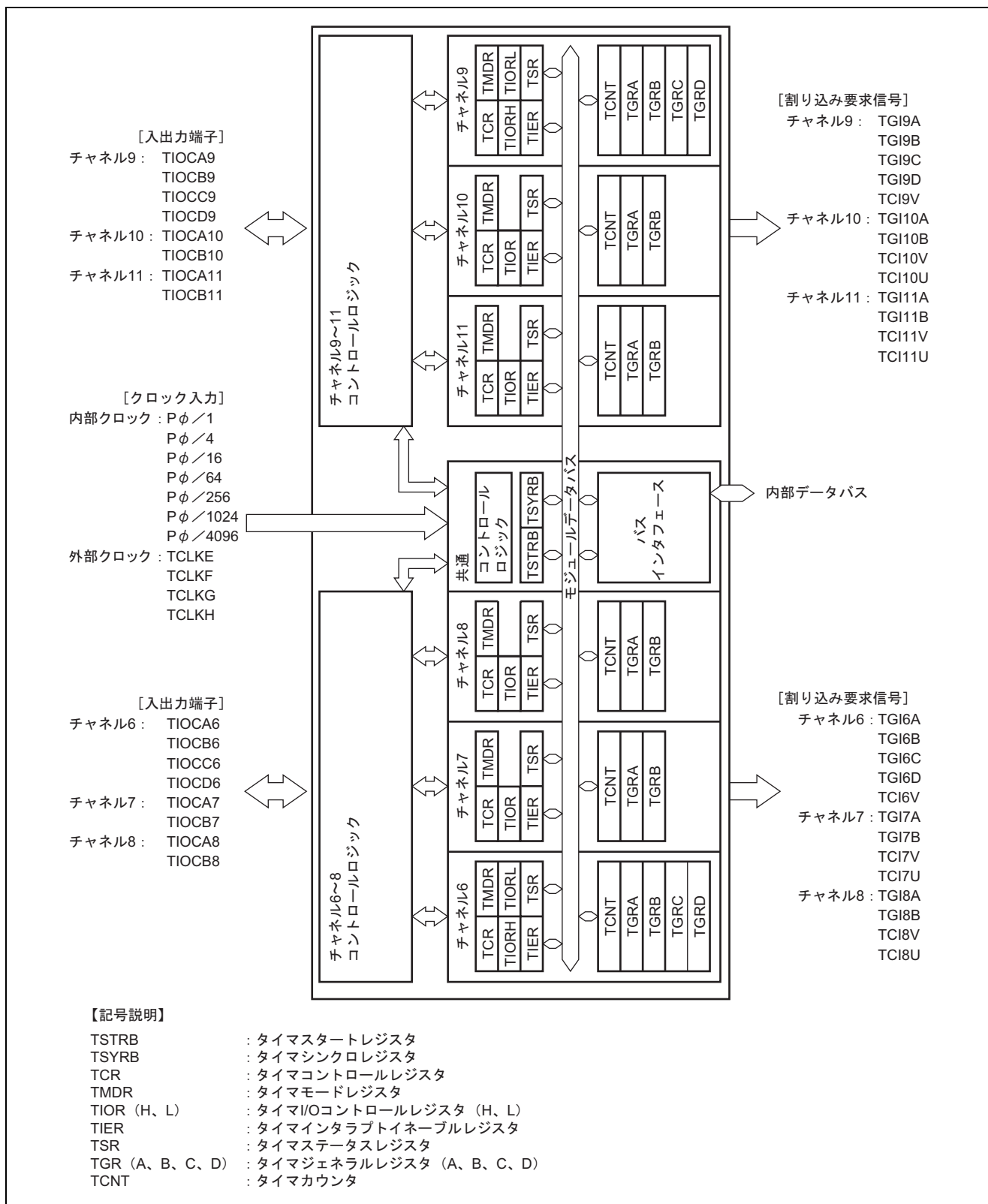


図 10.2 TPU (ユニット 1) のブロック図

10.2 入出力端子

TPU の端子構成を表 10.3 に示します。

表 10.3 端子構成

ユニット	チャンネル	名称	入出力	機能
0	共通	TCLKA	入力	外部クロック A 入力端子 (チャンネル 1、5 の位相計数モード A 相入力)
		TCLKB	入力	外部クロック B 入力端子 (チャンネル 1、5 の位相計数モード B 相入力)
		TCLKC	入力	外部クロック C 入力端子 (チャンネル 2、4 の位相計数モード A 相入力)
		TCLKD	入力	外部クロック D 入力端子 (チャンネル 2、4 の位相計数モード B 相入力)
	0	TIOCA0	入出力	TGRA_0 のインプットキャプチャ入力/アウトプットコンペア出力/ PWM 出力端子
		TIOCB0	入出力	TGRB_0 のインプットキャプチャ入力/アウトプットコンペア出力/ PWM 出力端子
		TIOCC0	入出力	TGRC_0 のインプットキャプチャ入力/アウトプットコンペア出力/ PWM 出力端子
		TIOCD0	入出力	TGRD_0 のインプットキャプチャ入力/アウトプットコンペア出力/ PWM 出力端子
	1	TIOCA1	入出力	TGRA_1 のインプットキャプチャ入力/アウトプットコンペア出力/ PWM 出力端子
		TIOCB1	入出力	TGRB_1 のインプットキャプチャ入力/アウトプットコンペア出力/ PWM 出力端子
	2	TIOCA2	入出力	TGRA_2 のインプットキャプチャ入力/アウトプットコンペア出力/ PWM 出力端子
		TIOCB2	入出力	TGRB_2 のインプットキャプチャ入力/アウトプットコンペア出力/ PWM 出力端子
	3	TIOCA3*	入出力	TGRA_3 のインプットキャプチャ入力/アウトプットコンペア出力/ PWM 出力端子
		TIOCB3*	入出力	TGRB_3 のインプットキャプチャ入力/アウトプットコンペア出力/ PWM 出力端子
		TIOCC3*	入出力	TGRC_3 のインプットキャプチャ入力/アウトプットコンペア出力/ PWM 出力端子
		TIOCD3*	入出力	TGRD_3 のインプットキャプチャ入力/アウトプットコンペア出力/ PWM 出力端子
	4	TIOCA4*	入出力	TGRA_4 のインプットキャプチャ入力/アウトプットコンペア出力/ PWM 出力端子
		TIOCB4*	入出力	TGRB_4 のインプットキャプチャ入力/アウトプットコンペア出力/ PWM 出力端子
	5	TIOCA5*	入出力	TGRA_5 のインプットキャプチャ入力/アウトプットコンペア出力/ PWM 出力端子
		TIOCB5*	入出力	TGRB_5 のインプットキャプチャ入力/アウトプットコンペア出力/ PWM 出力端子

ユニット	チャンネル	名称	入出力	機能
1	共通	TCLKE	入力	外部クロック E 入力端子 (チャンネル 7、11 の位相計数モード A 相入力)
		TCLKF	入力	外部クロック F 入力端子 (チャンネル 7、11 の位相計数モード B 相入力)
		TCLKG	入力	外部クロック G 入力端子 (チャンネル 8、10 の位相計数モード A 相入力)
		TCLKH	入力	外部クロック H 入力端子 (チャンネル 8、10 の位相計数モード B 相入力)
	6	TIOCA6	入出力	TGRA_6 のインプットキャプチャ入力/アウトプットコンペア出力/ PWM 出力端子
		TIOCB6	入出力	TGRB_6 のインプットキャプチャ入力/アウトプットコンペア出力/ PWM 出力端子
		TIOCC6	入出力	TGRC_6 のインプットキャプチャ入力/アウトプットコンペア出力/ PWM 出力端子
		TIOCD6	入出力	TGRD_6 のインプットキャプチャ入力/アウトプットコンペア出力/ PWM 出力端子
	7	TIOCA7	入出力	TGRA_7 のインプットキャプチャ入力/アウトプットコンペア出力/ PWM 出力端子
		TIOCB7	入出力	TGRB_7 のインプットキャプチャ入力/アウトプットコンペア出力/ PWM 出力端子
	8	TIOCA8	入出力	TGRA_8 のインプットキャプチャ入力/アウトプットコンペア出力/ PWM 出力端子
		TIOCB8	入出力	TGRB_8 のインプットキャプチャ入力/アウトプットコンペア出力/ PWM 出力端子
	9	TIOCA9	入出力	TGRA_9 のインプットキャプチャ入力/アウトプットコンペア出力/ PWM 出力端子
		TIOCB9	入出力	TGRB_9 のインプットキャプチャ入力/アウトプットコンペア出力/ PWM 出力端子
		TIOCC9	入出力	TGRC_9 のインプットキャプチャ入力/アウトプットコンペア出力/ PWM 出力端子
		TIOCD9	入出力	TGRD_9 のインプットキャプチャ入力/アウトプットコンペア出力/ PWM 出力端子
	10	TIOCA10	入出力	TGRA_10 のインプットキャプチャ入力/アウトプットコンペア出力/ PWM 出力端子
		TIOCB10	入出力	TGRB_10 のインプットキャプチャ入力/アウトプットコンペア出力/ PWM 出力端子
	11	TIOCA11	入出力	TGRA_11 のインプットキャプチャ入力/アウトプットコンペア出力/ PWM 出力端子
		TIOCB11	入出力	TGRB_11 のインプットキャプチャ入力/アウトプットコンペア出力/ PWM 出力端子

【注】 * 本 LSI では、チャンネル 3、チャンネル 4 およびチャンネル 5 の TIOCA3、TIOCB3、TIOCC3、TIOCD3、TIOCA4、TIOCB4、TIOCA5、TIOCB5 入出力端子はありません。

10.3 レジスタの説明

TPU には、以下のレジスタがあります。

2 ユニット (ユニット 0、ユニット 1) のレジスタは、TIER のビット 7 を除き同一機能です。本文中では、ユニット 0 のレジスタについて説明します。

- ユニット 0

- 【チャンネル 0】

- タイマコントロールレジスタ_0 (TCR_0)
- タイマモードレジスタ_0 (TMDR_0)
- タイマI/OコントロールレジスタH_0 (TIORH_0)
- タイマI/OコントロールレジスタL_0 (TIORL_0)
- タイマインタラプトイネーブルレジスタ_0 (TIER_0)
- タイマステータスレジスタ_0 (TSR_0)
- タイマカウンタ_0 (TCNT_0)
- タイマジェネラルレジスタA_0 (TGRA_0)
- タイマジェネラルレジスタB_0 (TGRB_0)
- タイマジェネラルレジスタC_0 (TGRC_0)
- タイマジェネラルレジスタD_0 (TGRD_0)

- 【チャンネル 1】

- タイマコントロールレジスタ_1 (TCR_1)
- タイマモードレジスタ_1 (TMDR_1)
- タイマI/Oコントロールレジスタ_1 (TIOR_1)
- タイマインタラプトイネーブルレジスタ_1 (TIER_1)
- タイマステータスレジスタ_1 (TSR_1)
- タイマカウンタ_1 (TCNT_1)
- タイマジェネラルレジスタA_1 (TGRA_1)
- タイマジェネラルレジスタB_1 (TGRB_1)

- 【チャンネル 2】

- タイマコントロールレジスタ_2 (TCR_2)
- タイマモードレジスタ_2 (TMDR_2)
- タイマI/Oコントロールレジスタ_2 (TIOR_2)
- タイマインタラプトイネーブルレジスタ_2 (TIER_2)
- タイマステータスレジスタ_2 (TSR_2)
- タイマカウンタ_2 (TCNT_2)
- タイマジェネラルレジスタA_2 (TGRA_2)
- タイマジェネラルレジスタB_2 (TGRB_2)

【チャンネル3】

- タイマコントロールレジスタ_3 (TCR_3)
- タイマモードレジスタ_3 (TMDR_3)
- タイマI/OコントロールレジスタH_3 (TIORH_3)
- タイマI/OコントロールレジスタL_3 (TIORL_3)
- タイマインタラプトイネーブルレジスタ_3 (TIER_3)
- タイマステータスレジスタ_3 (TSR_3)
- タイマカウンタ_3 (TCNT_3)
- タイマジェネラルレジスタA_3 (TGRA_3)
- タイマジェネラルレジスタB_3 (TGRB_3)
- タイマジェネラルレジスタC_3 (TGRC_3)
- タイマジェネラルレジスタD_3 (TGRD_3)

【チャンネル4】

- タイマコントロールレジスタ_4 (TCR_4)
- タイマモードレジスタ_4 (TMDR_4)
- タイマI/Oコントロールレジスタ_4 (TIOR_4)
- タイマインタラプトイネーブルレジスタ_4 (TIER_4)
- タイマステータスレジスタ_4 (TSR_4)
- タイマカウンタ_4 (TCNT_4)
- タイマジェネラルレジスタA_4 (TGRA_4)
- タイマジェネラルレジスタB_4 (TGRB_4)

【チャンネル5】

- タイマコントロールレジスタ_5 (TCR_5)
- タイマモードレジスタ_5 (TMDR_5)
- タイマI/Oコントロールレジスタ_5 (TIOR_5)
- タイマインタラプトイネーブルレジスタ_5 (TIER_5)
- タイマステータスレジスタ_5 (TSR_5)
- タイマカウンタ_5 (TCNT_5)
- タイマジェネラルレジスタA_5 (TGRA_5)
- タイマジェネラルレジスタB_5 (TGRB_5)

【共通】

- タイマスタートレジスタ (TSTR)
- タイマシンクロレジスタ (TSYR)

• ユニット1

【チャンネル6】

- タイマコントロールレジスタ_6 (TCR_6)
- タイマモードレジスタ_6 (TMDR_6)
- タイマI/OコントロールレジスタH_6 (TIORH_6)
- タイマI/OコントロールレジスタL_6 (TIORL_6)
- タイマインタラプトイネーブルレジスタ_6 (TIER_6)
- タイマステータスレジスタ_6 (TSR_6)
- タイマカウンタ_6 (TCNT_6)
- タイマジェネラルレジスタA_6 (TGRA_6)
- タイマジェネラルレジスタB_6 (TGRB_6)
- タイマジェネラルレジスタC_6 (TGRC_6)
- タイマジェネラルレジスタD_6 (TGRD_6)

【チャンネル7】

- タイマコントロールレジスタ_7 (TCR_7)
- タイマモードレジスタ_7 (TMDR_7)
- タイマI/Oコントロールレジスタ_7 (TIOR_7)
- タイマインタラプトイネーブルレジスタ_7 (TIER_7)
- タイマステータスレジスタ_7 (TSR_7)
- タイマカウンタ_7 (TCNT_7)
- タイマジェネラルレジスタA_7 (TGRA_7)
- タイマジェネラルレジスタB_7 (TGRB_7)

【チャンネル8】

- タイマコントロールレジスタ_8 (TCR_8)
- タイマモードレジスタ_8 (TMDR_8)
- タイマI/Oコントロールレジスタ_8 (TIOR_8)
- タイマインタラプトイネーブルレジスタ_8 (TIER_8)
- タイマステータスレジスタ_8 (TSR_8)
- タイマカウンタ_8 (TCNT_8)
- タイマジェネラルレジスタA_8 (TGRA_8)
- タイマジェネラルレジスタB_8 (TGRB_8)

【チャンネル9】

- タイマコントロールレジスタ_9 (TCR_9)
- タイマモードレジスタ_9 (TMDR_9)
- タイマI/OコントロールレジスタH_9 (TIORH_9)
- タイマI/OコントロールレジスタL_9 (TIORL_9)
- タイマインタラプトイネーブルレジスタ_9 (TIER_9)
- タイマステータスレジスタ_9 (TSR_9)
- タイマカウンタ_9 (TCNT_9)
- タイマジェネラルレジスタA_9 (TGRA_9)
- タイマジェネラルレジスタB_9 (TGRB_9)
- タイマジェネラルレジスタC_9 (TGRC_9)
- タイマジェネラルレジスタD_9 (TGRD_9)

【チャンネル10】

- タイマコントロールレジスタ_10 (TCR_10)
- タイマモードレジスタ_10 (TMDR_10)
- タイマI/Oコントロールレジスタ_10 (TIOR_10)
- タイマインタラプトイネーブルレジスタ_10 (TIER_10)
- タイマステータスレジスタ_10 (TSR_10)
- タイマカウンタ_10 (TCNT_10)
- タイマジェネラルレジスタA_10 (TGRA_10)
- タイマジェネラルレジスタB_10 (TGRB_10)

【チャンネル11】

- タイマコントロールレジスタ_11 (TCR_11)
- タイマモードレジスタ_11 (TMDR_11)
- タイマI/Oコントロールレジスタ_11 (TIOR_11)
- タイマインタラプトイネーブルレジスタ_11 (TIER_11)
- タイマステータスレジスタ_11 (TSR_11)
- タイマカウンタ_11 (TCNT_11)
- タイマジェネラルレジスタA_11 (TGRA_11)
- タイマジェネラルレジスタB_11 (TGRB_11)

【共通】

- タイマスタートレジスタ (TSTRB)
- タイマシンクロレジスタ (TSYRB)

10.3.1 タイマコントロールレジスタ (TCR)

TCR は各チャンネルの TCNT を制御します。TPU には、各チャンネルに 1 本、計 6 本の TCR があります。TCR の設定は、TCNT の動作が停止した状態で行ってください。

ビット	7	6	5	4	3	2	1	0
ビット名	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
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

ビット	ビット名	初期値	R/W	説 明
7	CCLR2	0	R/W	カウンタクリア 2、1、0
6	CCLR1	0	R/W	TCNT のカウンタクリア要因を選択します。詳細は表 10.4、表 10.5 を参照してください。
5	CCLR0	0	R/W	
4	CKEG1	0	R/W	
3	CKEG0	0	R/W	クロックエッジ 1、0 入力クロックのエッジを選択します。詳細は表 10.6 を参照してください。内部クロックを両エッジでカウントすると、入力クロックの周期が 1/2 になります (例 : $P\phi/4$ の両エッジ = $P\phi/2$ の立ち上がりエッジ)。チャンネル 1、2、4、5 で位相計数モードを使用する場合は、本設定は無視され、位相計数モードの設定が優先されます。内部クロックのエッジ選択は、入力クロックが $P\phi/4$ もしくはそれより遅い場合に有効です。入力クロックに $P\phi/1$ 、あるいは他のチャンネルのオーバフロー／アンダフローを選択した場合は本設定は無視されます。
2	TPSC2	0	R/W	タイマプリスケラ 2、1、0
1	TPSC1	0	R/W	TCNT のカウンタクロックを選択します。各チャンネル独立にクロックソースを選択することができます。詳細は表 10.7～表 10.12 を参照してください。クロックソースに外部クロックを選択する場合は、該当する端子の DDR ビットを 0 に、ICR ビットを 1 に設定してください。詳細は「第 9 章 I/O ポート」を参照してください。
0	TPSC0	0	R/W	

表 10.4 CCLR2~CCLR0 (チャンネル0、3)

チャンネル	ビット7	ビット6	ビット5	説明
	CCLR2	CCLR1	CCLR0	
0、3	0	0	0	TCNT のクリア禁止
	0	0	1	TGRA のコンペアマッチ/インプットキャプチャで TCNT クリア
	0	1	0	TGRB のコンペアマッチ/インプットキャプチャで TCNT クリア
	0	1	1	同期クリア/同期動作をしている他のチャンネルのカウンタクリアで TCNT をクリア*1
	1	0	0	TCNT のクリア禁止
	1	0	1	TGRC のコンペアマッチ/インプットキャプチャで TCNT クリア*2
	1	1	0	TGRD のコンペアマッチ/インプットキャプチャで TCNT クリア*2
	1	1	1	同期クリア/同期動作をしている他のチャンネルのカウンタクリアで TCNT をクリア*1

【注】 *1 同期動作の設定は、TSYR の SYNC ビットを 1 にセットすることにより行います。

*2 TGRC または TGRD をバッファレジスタとして使用している場合は、バッファレジスタの設定が優先され、コンペアマッチ/インプットキャプチャが発生しないため、TCNT はクリアされません。

表 10.5 CCLR2~CCLR0 (チャンネル1、2、4、5)

チャンネル	ビット7	ビット6	ビット5	説明
	リザーブ*2	CCLR1	CCLR0	
1、2、 4、5	0	0	0	TCNT のクリア禁止
	0	0	1	TGRA のコンペアマッチ/インプットキャプチャで TCNT クリア
	0	1	0	TGRB のコンペアマッチ/インプットキャプチャで TCNT クリア
	0	1	1	同期クリア/同期動作をしている他のチャンネルのカウンタクリアで TCNT をクリア*1

【注】 *1 同期動作の設定は、TSYR の SYNC ビットを 1 にセットすることにより行います。

*2 チャンネル1、2、4、5ではビット7はリザーブです。リードすると常に0がリードされます。ライトは無効です。

表 10.6 入力クロックエッジ選択

クロックエッジ選択		入力クロック	
CKEG1	CKEG0	内部クロック	外部クロック
0	0	立ち下がりエッジでカウント	立ち上がりエッジでカウント
0	1	立ち上がりエッジでカウント	立ち下がりエッジでカウント
1	x	両エッジでカウント	両エッジでカウント

【記号説明】 x : Don't care

表 10.7 TPSC2~TPSC0 (チャンネル0)

チャンネル	ビット2	ビット1	ビット0	説明
	TPSC2	TPSC1	TPSC0	
0	0	0	0	内部クロック : Pφ/1 でカウント
	0	0	1	内部クロック : Pφ/4 でカウント
	0	1	0	内部クロック : Pφ/16 でカウント
	0	1	1	内部クロック : Pφ/64 でカウント
	1	0	0	外部クロック : TCLKA 端子入力でカウント
	1	0	1	外部クロック : TCLKB 端子入力でカウント
	1	1	0	外部クロック : TCLKC 端子入力でカウント
	1	1	1	外部クロック : TCLKD 端子入力でカウント

表 10.8 TPSC2~TPSC0 (チャンネル1)

チャンネル	ビット2	ビット1	ビット0	説明
	TPSC2	TPSC1	TPSC0	
1	0	0	0	内部クロック : Pφ/1 でカウント
	0	0	1	内部クロック : Pφ/4 でカウント
	0	1	0	内部クロック : Pφ/16 でカウント
	0	1	1	内部クロック : Pφ/64 でカウント
	1	0	0	外部クロック : TCLKA 端子入力でカウント
	1	0	1	外部クロック : TCLKB 端子入力でカウント
	1	1	0	内部クロック : Pφ/256 でカウント
	1	1	1	TCNT2 のオーバフロー/アンダフローでカウント

【注】チャンネル1が位相計数モード時、この設定は無効になります。

表 10.9 TPSC2~TPSC0 (チャンネル2)

チャンネル	ビット2	ビット1	ビット0	説明
	TPSC2	TPSC1	TPSC0	
2	0	0	0	内部クロック : Pφ/1 でカウント
	0	0	1	内部クロック : Pφ/4 でカウント
	0	1	0	内部クロック : Pφ/16 でカウント
	0	1	1	内部クロック : Pφ/64 でカウント
	1	0	0	外部クロック : TCLKA 端子入力でカウント
	1	0	1	外部クロック : TCLKB 端子入力でカウント
	1	1	0	外部クロック : TCLKC 端子入力でカウント
	1	1	1	内部クロック : Pφ/1024 でカウント

【注】チャンネル2が位相計数モード時、この設定は無効になります。

表 10.10 TPSC2~TPSC0 (チャンネル3)

チャンネル	ビット2	ビット1	ビット0	説明
	TPSC2	TPSC1	TPSC0	
3	0	0	0	内部クロック : Pφ/1 でカウント
	0	0	1	内部クロック : Pφ/4 でカウント
	0	1	0	内部クロック : Pφ/16 でカウント
	0	1	1	内部クロック : Pφ/64 でカウント
	1	0	0	外部クロック : TCLKA 端子入力でカウント
	1	0	1	内部クロック : Pφ/1024 でカウント
	1	1	0	内部クロック : Pφ/256 でカウント
	1	1	1	内部クロック : Pφ/4096 でカウント

表 10.11 TPSC2~TPSC0 (チャンネル4)

チャンネル	ビット2	ビット1	ビット0	説明
	TPSC2	TPSC1	TPSC0	
4	0	0	0	内部クロック : Pφ/1 でカウント
	0	0	1	内部クロック : Pφ/4 でカウント
	0	1	0	内部クロック : Pφ/16 でカウント
	0	1	1	内部クロック : Pφ/64 でカウント
	1	0	0	外部クロック : TCLKA 端子入力でカウント
	1	0	1	外部クロック : TCLKC 端子入力でカウント
	1	1	0	内部クロック : Pφ/1024 でカウント
	1	1	1	TCNT5 のオーバフロー/アンダフローでカウント

【注】チャンネル4が位相計数モード時、この設定は無効になります。

表 10.12 TPSC2~TPSC0 (チャンネル5)

チャンネル	ビット2	ビット1	ビット0	説明
	TPSC2	TPSC1	TPSC0	
5	0	0	0	内部クロック : Pφ/1 でカウント
	0	0	1	内部クロック : Pφ/4 でカウント
	0	1	0	内部クロック : Pφ/16 でカウント
	0	1	1	内部クロック : Pφ/64 でカウント
	1	0	0	外部クロック : TCLKA 端子入力でカウント
	1	0	1	外部クロック : TCLKC 端子入力でカウント
	1	1	0	内部クロック : Pφ/256 でカウント
	1	1	1	外部クロック : TCLKD 端子入力でカウント

【注】チャンネル5が位相計数モード時、この設定は無効になります。

10.3.2 タイマモードレジスタ (TMDR)

TMDR は、各チャンネルの動作モードの設定を行います。TPU には、各チャンネルに 1 本、計 6 本の TMDR があります。TMDR の設定は、TCNT の動作が停止した状態で行ってください。

ビット	7	6	5	4	3	2	1	0
ビット名	—	—	BFB	BFA	MD3	MD2	MD1	MD0
初期値 :	1	1	0	0	0	0	0	0
R/W :	R	R	R/W	R/W	R/W	R/W	R/W	R/W

ビット	ビット名	初期値	R/W	説明
7	—	1	R	リザーブビット
6	—	1	R	これらのビットはリードのみ有効で、ライトは無効です。
5	BFB	0	R/W	バッファ動作 B TGRB を通常動作させるか、TGRB と TGRD を組み合わせてバッファ動作させるかを設定します。TGRD をバッファレジスタとして使用した場合は、TGRD の入力キャプチャ/アウトプットコンペアは発生しません。 TGRD を持たないチャンネル 1、2、4、5 ではこのビットはリザーブビットになります。リードすると常に 0 が読み出されます。ライトは無効です。 0 : TGRB は通常動作 1 : TGRB と TGRD はバッファ動作
4	BFA	0	R/W	バッファ動作 A TGRA を通常動作させるか、TGRA と TGRC を組み合わせてバッファ動作させるかを設定します。TGRC をバッファレジスタとして使用した場合は、TGRC の入力キャプチャ/アウトプットコンペアは発生しません。 TGRC を持たないチャンネル 1、2、4、5 ではこのビットはリザーブビットになります。リードすると常に 0 が読み出されます。ライトは無効です。 0 : TGRA は通常動作 1 : TGRA と TGRC はバッファ動作
3	MD3	0	R/W	モード 3~0
2	MD2	0	R/W	MD3~MD0 はタイマの動作モードを設定します。
1	MD1	0	R/W	MD3 はリザーブビットです。ライト時には常に 0 としてください。
0	MD0	0	R/W	詳細は表 10.13 を参照してください。

表 10.13 MD3~MD0

ビット3	ビット2	ビット1	ビット0	説 明
MD3* ¹	MD2* ²	MD1	MD0	
0	0	0	0	通常動作
0	0	0	1	リザーブ (設定禁止)
0	0	1	0	PWM モード 1
0	0	1	1	PWM モード 2
0	1	0	0	位相計数モード 1
0	1	0	1	位相計数モード 2
0	1	1	0	位相計数モード 3
0	1	1	1	位相計数モード 4
1	x	x	x	— (設定禁止)

【記号説明】 x : Don't care

【注】 *1 MD3 はリザーブビットです。ライト時には常に 0 としてください。

*2 チャンネル 0、3 では、位相計数モードの設定はできません。MD2 には常に 0 をライトしてください。

10.3.3 タイマ I/O コントロールレジスタ (TIOR)

TIOR は TGR を制御します。TPU には、チャンネル 0、3 に各 2 本、チャンネル 1、2、4、5 に各 1 本、計 8 本の TIOR があります。TIOR の設定は、TCNT の動作が停止した状態で行ってください。また、TIOR は TMDR の設定により影響を受けますので注意してください。

TIOR で指定した初期出力はカウンタが停止した状態 (TSTR の CST ビットを 0 にクリアした) で有効になります。また、PWM モード 2 の場合にはカウンタが 0 にクリアされた時点での出力を指定します。

TGRC、あるいは TGRD をバッファ動作に設定した場合は、本設定は無効となり、バッファレジスタとして動作します。

- TIORH_0、TIOR_1、TIOR_2、TIORH_3、TIOR_4、TIOR_5

ビット	7	6	5	4	3	2	1	0
ビット名	I OB3	I OB2	I OB1	I OB0	I OA3	I OA2	I OA1	I OA0
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

- TIORL_0、TIORL_3

ビット	7	6	5	4	3	2	1	0
ビット名	I OD3	I OD2	I OD1	I OD0	I OC3	I OC2	I OC1	I OC0
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

- TIORH_0、TIOR_1、TIOR_2、TIORH_3、TIOR_4、TIOR_5

ビット	ビット名	初期値	R/W	説明
7	IOB3	0	R/W	I/O コントロール B3~B0 TGRB の機能を設定します。詳細は表 10.14、表 10.16、表 10.17、表 10.18、表 10.20、表 10.21 を参照してください。
6	IOB2	0	R/W	
5	IOB1	0	R/W	
4	IOB0	0	R/W	
3	IOA3	0	R/W	I/O コントロール A3~A0 TGRA の機能を設定します。詳細は表 10.22、表 10.24、表 10.25、表 10.26、表 10.28、表 10.29 を参照してください。
2	IOA2	0	R/W	
1	IOA1	0	R/W	
0	IOA0	0	R/W	

- TIORL_0、TIORL_3

ビット	ビット名	初期値	R/W	説明
7	IOD3	0	R/W	I/O コントロール D3~D0 TGRD の機能を設定します。詳細は表 10.15、表 10.19 を参照してください。
6	IOD2	0	R/W	
5	IOD1	0	R/W	
4	IOD0	0	R/W	
3	IOC3	0	R/W	I/O コントロール C3~C0 TGRC の機能を設定します。詳細は表 10.23、表 10.27 を参照してください。
2	IOC2	0	R/W	
1	IOC1	0	R/W	
0	IOC0	0	R/W	

表 10.14 TIORH_0

ビット7	ビット6	ビット5	ビット4	説明	
IOB3	IOB2	IOB1	IOB0	TGRB_0の機能	TIOCB0端子の機能
0	0	0	0	アウトプットコンペアレジスタ	出力禁止
0	0	0	1		初期出力は0出力 コンペアマッチで0出力
0	0	1	0		初期出力は0出力 コンペアマッチで1出力
0	0	1	1		初期出力は0出力 コンペアマッチでトグル出力
0	1	0	0		出力禁止
0	1	0	1		初期出力は1出力 コンペアマッチで0出力
0	1	1	0		初期出力は1出力 コンペアマッチで1出力
0	1	1	1		初期出力は1出力 コンペアマッチでトグル出力
1	0	0	0	インプットキャプチャレジスタ	キャプチャ入力元は TIOCB0 端子 立ち上がりエッジでインプットキャプチャ
1	0	0	1		キャプチャ入力元は TIOCB0 端子 立ち下がりエッジでインプットキャプチャ
1	0	1	x		キャプチャ入力元は TIOCB0 端子 両エッジでインプットキャプチャ
1	1	x	x		キャプチャ入力元はチャネル1/カウントクロック TCNT_1のカウンタアップ/カウンタダウンでイン プットキャプチャ*

【記号説明】 x : Don't care

【注】 * TCR_1のTPSC2~TPSC0ビットをB'000とし、TCNT_1のカウントクロックにPφ/1を使用した場合は、インプットキャプチャは発生しません。

表 10.15 TIORL_0

ビット7	ビット6	ビット5	ビット4	説明	
IOD3	IOD2	IOD1	IOD0	TGRD_0の機能	TIOCD0 端子の機能
0	0	0	0	アウトプットコンペアレジスタ*2	出力禁止
0	0	0	1		初期出力は0出力 コンペアマッチで0出力
0	0	1	0		初期出力は0出力 コンペアマッチで1出力
0	0	1	1		初期出力は0出力 コンペアマッチでトグル出力
0	1	0	0		出力禁止
0	1	0	1		初期出力は1出力 コンペアマッチで0出力
0	1	1	0		初期出力は1出力 コンペアマッチで1出力
0	1	1	1		初期出力は1出力 コンペアマッチでトグル出力
1	0	0	0	インプットキャプチャレジスタ*2	キャプチャ入力元は TIOCD0 端子 立ち上がりエッジでインプットキャプチャ
1	0	0	1		キャプチャ入力元は TIOCD0 端子 立ち下がりエッジでインプットキャプチャ
1	0	1	x		キャプチャ入力元は TIOCD0 端子 両エッジでインプットキャプチャ
1	1	x	x		キャプチャ入力元はチャネル1/カウントクロック TCNT_1のカウンタアップ/カウンタダウンでイン プットキャプチャ*1

【記号説明】 x : Don't care

【注】 *1 TCR_1のTPSC2~TPSC0ビットをB'000とし、TCNT_1のカウントクロックにPφ/1を使用した場合は、インプットキャプチャは発生しません。

*2 TMDR_0のBFBビットを1にセットしてTGRD_0をバッファレジスタとして使用した場合は、インプットキャプチャ/アウトプットコンペアは発生しません。この場合、コンペアマッチで0/1/トグル出力を選択しないでください。

表 10.16 TIOR_1

ビット7	ビット6	ビット5	ビット4	説明	
IOB3	IOB2	IOB1	IOB0	TGRB_1の機能	TIOCB1端子の機能
0	0	0	0	アウトプットコンペアレジスタ	出力禁止
0	0	0	1		初期出力は0出力 コンペアマッチで0出力
0	0	1	0		初期出力は0出力 コンペアマッチで1出力
0	0	1	1		初期出力は0出力 コンペアマッチでトグル出力
0	1	0	0		出力禁止
0	1	0	1		初期出力は1出力 コンペアマッチで0出力
0	1	1	0		初期出力は1出力 コンペアマッチで1出力
0	1	1	1		初期出力は1出力 コンペアマッチでトグル出力
1	0	0	0	インプットキャプチャレジスタ	キャプチャ入力元はTIOCB1端子 立ち上がりエッジでインプットキャプチャ
1	0	0	1		キャプチャ入力元はTIOCB1端子 立ち下がりエッジでインプットキャプチャ
1	0	1	x		キャプチャ入力元はTIOCB1端子 両エッジでインプットキャプチャ
1	1	x	x		TGRC_0コンペアマッチ/インプットキャプチャ TGRC_0のコンペアマッチ/インプットキャプチャの発生でインプットキャプチャ

【記号説明】 x : Don't care

表 10.17 TIOR_2

ビット7	ビット6	ビット5	ビット4	説 明	
IOB3	IOB2	IOB1	IOB0	TGRB_2 の機能	TIOCB2 端子の機能
0	0	0	0	アウトプットコンペア レジスタ	出力禁止
0	0	0	1		初期出力は 0 出力 コンペアマッチで 0 出力
0	0	1	0		初期出力は 0 出力 コンペアマッチで 1 出力
0	0	1	1		初期出力は 0 出力 コンペアマッチでトグル出力
0	1	0	0		出力禁止
0	1	0	1		初期出力は 1 出力 コンペアマッチで 0 出力
0	1	1	0		初期出力は 1 出力 コンペアマッチで 1 出力
0	1	1	1		初期出力は 1 出力 コンペアマッチでトグル出力
1	x	0	0	インプットキャプチャ レジスタ	キャプチャ入力元は TIOCB2 端子 立ち上がりエッジでインプットキャプチャ
1	x	0	1		キャプチャ入力元は TIOCB2 端子 立ち下がりエッジでインプットキャプチャ
1	x	1	x		キャプチャ入力元は TIOCB2 端子 両エッジでインプットキャプチャ

【記号説明】 x : Don't care

表 10.18 TIORH_3

ビット7	ビット6	ビット5	ビット4	説明	
IOB3	IOB2	IOB1	IOB0	TGRB_3の機能	TIOCB3端子の機能
0	0	0	0	アウトプットコンペアレジスタ	出力禁止
0	0	0	1		初期出力は0出力 コンペアマッチで0出力
0	0	1	0		初期出力は0出力 コンペアマッチで1出力
0	0	1	1		初期出力は0出力 コンペアマッチでトグル出力
0	1	0	0		出力禁止
0	1	0	1		初期出力は1出力 コンペアマッチで0出力
0	1	1	0		初期出力は1出力 コンペアマッチで1出力
0	1	1	1		初期出力は1出力 コンペアマッチでトグル出力
1	0	0	0	インプットキャプチャレジスタ	キャプチャ入力元はTIOCB3端子 立ち上がりエッジでインプットキャプチャ
1	0	0	1		キャプチャ入力元はTIOCB3端子 立ち下がりエッジでインプットキャプチャ
1	0	1	x		キャプチャ入力元はTIOCB3端子 両エッジでインプットキャプチャ
1	1	x	x		キャプチャ入力元はチャネル4/カウントクロック TCNT_4のカウントアップ/カウントダウンでイン プットキャプチャ*

【記号説明】 x : Don't care

【注】 * TCR_4のTPSC2~TPSC0ビットをB'000とし、TCNT_4のカウントクロックにPφ/1を使用した場合は、インプットキャプチャは発生しません。

表 10.19 TIORL_3

ビット7	ビット6	ビット5	ビット4	説明	
IOD3	IOD2	IOD1	IOD0	TGRD_3の機能	TIOCD3端子の機能
0	0	0	0	アウトプットコンペアレジスタ*2	出力禁止
0	0	0	1		初期出力は0出力 コンペアマッチで0出力
0	0	1	0		初期出力は0出力 コンペアマッチで1出力
0	0	1	1		初期出力は0出力 コンペアマッチでトグル出力
0	1	0	0		出力禁止
0	1	0	1		初期出力は1出力 コンペアマッチで0出力
0	1	1	0		初期出力は1出力 コンペアマッチで1出力
0	1	1	1		初期出力は1出力 コンペアマッチでトグル出力
1	0	0	0	インプットキャプチャレジスタ*2	キャプチャ入力元は TIOCD3 端子 立ち上がりエッジでインプットキャプチャ
1	0	0	1		キャプチャ入力元は TIOCD3 端子 立ち下がりエッジでインプットキャプチャ
1	0	1	x		キャプチャ入力元は TIOCD3 端子 両エッジでインプットキャプチャ
1	1	x	x		キャプチャ入力元はチャネル4/カウントクロック TCNT_4のカウントアップ/カウントダウンでイン プットキャプチャ*1

【記号説明】 x : Don't care

【注】 *1 TCR_4のTPSC2~TPSC0ビットをB'000とし、TCNT_4のカウントクロックにPφ/1を使用した場合は、インプットキャプチャは発生しません。

*2 TMDR_3のBFBビットを1にセットしてTGRD_3をバッファレジスタとして使用した場合は、インプットキャプチャ/アウトプットコンペアは発生しません。この場合、コンペアマッチで0/1/トグル出力を選択しないでください。

表 10.20 TIOR_4

ビット7	ビット6	ビット5	ビット4	説明	
IOB3	IOB2	IOB1	IOB0	TGRB_4の機能	TIOCB4端子の機能
0	0	0	0	アウトプットコンペアレジスタ	出力禁止
0	0	0	1		初期出力は0出力 コンペアマッチで0出力
0	0	1	0		初期出力は0出力 コンペアマッチで1出力
0	0	1	1		初期出力は0出力 コンペアマッチでトグル出力
0	1	0	0		出力禁止
0	1	0	1		初期出力は1出力 コンペアマッチで0出力
0	1	1	0		初期出力は1出力 コンペアマッチで1出力
0	1	1	1		初期出力は1出力 コンペアマッチでトグル出力
1	0	0	0	インプットキャプチャレジスタ	キャプチャ入力元は TIOCB4 端子 立ち上がりエッジでインプットキャプチャ
1	0	0	1		キャプチャ入力元は TIOCB4 端子 立ち下がりエッジでインプットキャプチャ
1	0	1	x		キャプチャ入力元は TIOCB4 端子 両エッジでインプットキャプチャ
1	1	x	x		キャプチャ入力元は TGRC_3 コンペアマッチ/インプットキャプチャ TGRC_3のコンペアマッチ/インプットキャプチャの発生でインプットキャプチャ

【記号説明】 x : Don't care

表 10.21 TIOR_5

ビット7	ビット6	ビット5	ビット4	説 明	
IOB3	IOB2	IOB1	IOB0	TGRB_5の機能	TIOCB5 端子の機能
0	0	0	0	アウトプットコンペア レジスタ	出力禁止
0	0	0	1		初期出力は0出力 コンペアマッチで0出力
0	0	1	0		初期出力は0出力 コンペアマッチで1出力
0	0	1	1		初期出力は0出力 コンペアマッチでトグル出力
0	1	0	0		出力禁止
0	1	0	1		初期出力は1出力 コンペアマッチで0出力
0	1	1	0		初期出力は1出力 コンペアマッチで1出力
0	1	1	1		初期出力は1出力 コンペアマッチでトグル出力
1	x	0	0	インプットキャプチャ レジスタ	キャプチャ入力元は TIOCB5 端子 立ち上がりエッジでインプットキャプチャ
1	x	0	1		キャプチャ入力元は TIOCB5 端子 立ち下がりエッジでインプットキャプチャ
1	x	1	x		キャプチャ入力元は TIOCB5 端子 両エッジでインプットキャプチャ

【記号説明】 x : Don't care

表 10.22 TIORH_0

ビット3	ビット2	ビット1	ビット0	説明	
IOA3	IOA2	IOA1	IOA0	TGRA_0の機能	TIOCA0端子の機能
0	0	0	0	アウトプットコンペアレジスタ	出力禁止
0	0	0	1		初期出力は0出力 コンペアマッチで0出力
0	0	1	0		初期出力は0出力 コンペアマッチで1出力
0	0	1	1		初期出力は0出力 コンペアマッチでトグル出力
0	1	0	0		出力禁止
0	1	0	1		初期出力は1出力 コンペアマッチで0出力
0	1	1	0		初期出力は1出力 コンペアマッチで1出力
0	1	1	1		初期出力は1出力 コンペアマッチでトグル出力
1	0	0	0	インプットキャプチャレジスタ	キャプチャ入力元は TIOCA0 端子 立ち上がりエッジでインプットキャプチャ
1	0	0	1		キャプチャ入力元は TIOCA0 端子 立ち下がりエッジでインプットキャプチャ
1	0	1	x		キャプチャ入力元は TIOCA0 端子 両エッジでインプットキャプチャ
1	1	x	x		キャプチャ入力元はチャネル1/カウントクロック TCNT_1のカウントアップ/カウントダウンでイン プットキャプチャ

【記号説明】 x : Don't care

表 10.23 TIORL_0

ビット3	ビット2	ビット1	ビット0	説明	
IOC3	IOC2	IOC1	IOC0	TGRC_0の機能	TIOCC0の端子の機能
0	0	0	0	アウトプットコンペアレジスタ*	出力禁止
0	0	0	1		初期出力は0出力 コンペアマッチで0出力
0	0	1	0		初期出力は0出力 コンペアマッチで1出力
0	0	1	1		初期出力は0出力 コンペアマッチでトグル出力
0	1	0	0		出力禁止
0	1	0	1		初期出力は1出力 コンペアマッチで0出力
0	1	1	0		初期出力は1出力 コンペアマッチで1出力
0	1	1	1		初期出力は1出力 コンペアマッチでトグル出力
1	0	0	0	インプットキャプチャレジスタ*	キャプチャ入力元は TIOCC0 端子 立ち上がりエッジでインプットキャプチャ
1	0	0	1		キャプチャ入力元は TIOCC0 端子 立ち下がりエッジでインプットキャプチャ
1	0	1	x		キャプチャ入力元は TIOCC0 端子 両エッジでインプットキャプチャ
1	1	x	x		キャプチャ入力元はチャネル1/カウントクロック TCNT_1のカウントアップ/カウントダウンでイン プットキャプチャ

【記号説明】 x : Don't care

【注】 * TMDR_0のBFAビットを1にセットしてTGRC_0をバッファレジスタとして使用した場合は、インプットキャプチャ/アウトプットコンペアは発生しません。この場合、コンペアマッチで0/1/トグル出力を選択しないでください。

表 10.24 TIOR_1

ビット3	ビット2	ビット1	ビット0	説明	
IOA3	IOA2	IOA1	IOA0	TGRA_1の機能	TIOCA1端子の機能
0	0	0	0	アウトプットコンペアレジスタ	出力禁止
0	0	0	1		初期出力は0出力 コンペアマッチで0出力
0	0	1	0		初期出力は0出力 コンペアマッチで1出力
0	0	1	1		初期出力は0出力 コンペアマッチでトグル出力
0	1	0	0		出力禁止
0	1	0	1		初期出力は1出力 コンペアマッチで0出力
0	1	1	0		初期出力は1出力 コンペアマッチで1出力
0	1	1	1		初期出力は1出力 コンペアマッチでトグル出力
1	0	0	0	インプットキャプチャレジスタ	キャプチャ入力元は TIOCA1 端子 立ち上がりエッジでインプットキャプチャ
1	0	0	1		キャプチャ入力元は TIOCA1 端子 立ち下がりエッジでインプットキャプチャ
1	0	1	x		キャプチャ入力元は TIOCA1 端子 両エッジでインプットキャプチャ
1	1	x	x		キャプチャ入力元は TGRA_0 コンペアマッチ/インプットキャプチャ チャンネル0/TGRA_0のコンペアマッチ/インプットキャプチャの発生でインプットキャプチャ

【記号説明】 x : Don't care

表 10.25 TIOR_2

ビット3	ビット2	ビット1	ビット0	説明	
IOA3	IOA2	IOA1	IOA0	TGRA_2の機能	TIOCA2端子の機能
0	0	0	0	アウトプットコンペアレジスタ	出力禁止
0	0	0	1		初期出力は0出力 コンペアマッチで0出力
0	0	1	0		初期出力は0出力 コンペアマッチで1出力
0	0	1	1		初期出力は0出力 コンペアマッチでトグル出力
0	1	0	0		出力禁止
0	1	0	1		初期出力は1出力 コンペアマッチで0出力
0	1	1	0		初期出力は1出力 コンペアマッチで1出力
0	1	1	1		初期出力は1出力 コンペアマッチでトグル出力
1	x	0	0		インプットキャプチャレジスタ
1	x	0	1	キャプチャ入力元はTIOCA2端子 立ち下がりエッジでインプットキャプチャ	
1	x	1	x	キャプチャ入力元はTIOCA2端子 両エッジでインプットキャプチャ	

【記号説明】 x : Don't care

表 10.26 TIORH_3

ビット3	ビット2	ビット1	ビット0	説明	
IOA3	IOA2	IOA1	IOA0	TGRA_3の機能	TIOCA3端子の機能
0	0	0	0	アウトプットコンペアレジスタ	出力禁止
0	0	0	1		初期出力は0出力 コンペアマッチで0出力
0	0	1	0		初期出力は0出力 コンペアマッチで1出力
0	0	1	1		初期出力は0出力 コンペアマッチでトグル出力
0	1	0	0		出力禁止
0	1	0	1		初期出力は1出力 コンペアマッチで0出力
0	1	1	0		初期出力は1出力 コンペアマッチで1出力
0	1	1	1		初期出力は1出力 コンペアマッチでトグル出力
1	0	0	0	インプットキャプチャレジスタ	キャプチャ入力元はTIOCA3端子 立ち上がりエッジでインプットキャプチャ
1	0	0	1		キャプチャ入力元はTIOCA3端子 立ち下がりエッジでインプットキャプチャ
1	0	1	x		キャプチャ入力元はTIOCA3端子 両エッジでインプットキャプチャ
1	1	x	x		キャプチャ入力元はチャネル4/カウントクロック TCNT_4のカウントアップ/カウントダウンでイン プットキャプチャ

【記号説明】 x : Don't care

表 10.27 TIORL_3

ビット3	ビット2	ビット1	ビット0	説明	
IOC3	IOC2	IOC1	IOC0	TGRC_3の端子	TIOCC3端子の機能
0	0	0	0	アウトプットコンペアレジスタ*	出力禁止
0	0	0	1		初期出力は0出力 コンペアマッチで0出力
0	0	1	0		初期出力は0出力 コンペアマッチで1出力
0	0	1	1		初期出力は0出力 コンペアマッチでトグル出力
0	1	0	0		出力禁止
0	1	0	1		初期出力は1出力 コンペアマッチで0出力
0	1	1	0		初期出力は1出力 コンペアマッチで1出力
0	1	1	1		初期出力は1出力 コンペアマッチでトグル出力
1	0	0	0	インプットキャプチャレジスタ*	キャプチャ入力元はTIOCC3端子 立ち上がりエッジでインプットキャプチャ
1	0	0	1		キャプチャ入力元はTIOCC3端子 立ち下がりエッジでインプットキャプチャ
1	0	1	x		キャプチャ入力元はTIOCC3端子 両エッジでインプットキャプチャ
1	1	x	x		キャプチャ入力元はチャネル4/カウントクロック TCNT_4のカウントアップ/カウントダウンでイン プットキャプチャ

【記号説明】 x : Don't care

【注】 * TMDR_3のBFAビットを1にセットしてTGRC_3をバッファレジスタとして使用した場合は、インプットキャプチャ/アウトプットコンペアは発生しません。この場合、コンペアマッチで0/1/トグル出力を選択しないでください。

表 10.28 TIOR_4

ビット3	ビット2	ビット1	ビット0	説明	
IOA3	IOA2	IOA1	IOA0	TGRA_4の機能	TIOCA4端子の機能
0	0	0	0	アウトプットコンペアレジスタ	出力禁止
0	0	0	1		初期出力は0出力 コンペアマッチで0出力
0	0	1	0		初期出力は0出力 コンペアマッチで1出力
0	0	1	1		初期出力は0出力 コンペアマッチでトグル出力
0	1	0	0		出力禁止
0	1	0	1		初期出力は1出力 コンペアマッチで0出力
0	1	1	0		初期出力は1出力 コンペアマッチで1出力
0	1	1	1		初期出力は1出力 コンペアマッチでトグル出力
1	0	0	0	インプットキャプチャレジスタ	キャプチャ入力元は TIOCA4 端子 立ち上がりエッジでインプットキャプチャ
1	0	0	1		キャプチャ入力元は TIOCA4 端子 立ち下がりエッジでインプットキャプチャ
1	0	1	x		キャプチャ入力元は TIOCA4 端子 両エッジでインプットキャプチャ
1	1	x	x		キャプチャ入力元は TGRA_3 コンペアマッチ/イン プットキャプチャ TGRA_3のコンペアマッチ/インプットキャプチャ の発生でインプットキャプチャ

【記号説明】 x : Don't care

表 10.29 TIOR_5

ビット3	ビット2	ビット1	ビット0	説 明	
IOA3	IOA2	IOA1	IOA0	TGRA_5 の機能	TIOCA5 端子の機能
0	0	0	0	アウトプットコンペア レジスタ	出力禁止
0	0	0	1		初期出力は 0 出力 コンペアマッチで 0 出力
0	0	1	0		初期出力は 0 出力 コンペアマッチで 1 出力
0	0	1	1		初期出力は 0 出力 コンペアマッチでトグル出力
0	1	0	0		出力禁止
0	1	0	1		初期出力は 1 出力 コンペアマッチで 0 出力
0	1	1	0		初期出力は 1 出力 コンペアマッチで 1 出力
0	1	1	1		初期出力は 1 出力 コンペアマッチでトグル出力
1	x	0	0	インプットキャプチャ レジスタ	キャプチャ入力元は TIOCA5 端子 立ち上がりエッジでインプットキャプチャ
1	x	0	1		キャプチャ入力元は TIOCA5 端子 立ち下がりエッジでインプットキャプチャ
1	x	1	x		キャプチャ入力元は TIOCA5 端子 両エッジでインプットキャプチャ

【記号説明】 x : Don't care

10.3.4 タイマインタラプトイネーブルレジスタ (TIER)

TIER は、各チャンネルの割り込み要求の許可、禁止を制御します。TPU には、各チャンネルに 1 本、計 6 本の TIER があります。

ビット	7	6	5	4	3	2	1	0
ビット名	TTGE	—	TCIEU	TCIEV	TGIED	TCIEC	TGIEB	TGIEA
初期値 :	0	1	0	0	0	0	0	0
R/W :	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

ビット	ビット名	初期値	R/W	説明
7	TTGE	0	R/W	A/D 変換開始要求イネーブル TGRA のインプットキャプチャ/コンペアマッチによる A/D 変換器開始要求の発生を許可または禁止します。 0 : A/D 変換開始要求の発生を禁止 1 : A/D 変換開始要求の発生を許可
6	—	1	R	リザーブビット このビットはリードのみ有効で、ライトは無効です。
5	TCIEU	0	R/W	アンダフローインタラプトイネーブル チャンネル 1、2、4、5 で TSR の TCFU フラグが 1 にセットされたとき、TCFU フラグによる割り込み要求 (TCIU) を許可または禁止します。 チャンネル 0、3 ではリザーブビットです。 リードすると常に 0 がリードされます。ライトは無効です。 0 : TCFU による割り込み要求 (TCIU) を禁止 1 : TCFU による割り込み要求 (TCIU) を許可
4	TCIEV	0	R/W	オーバフローインタラプトイネーブル TSR の TCFV フラグが 1 にセットされたとき、TCFV フラグによる割り込み要求 (TCIV) を許可または禁止します。 0 : TCFV による割り込み要求 (TCIV) を禁止 1 : TCFV による割り込み要求 (TCIV) を許可
3	TGIED	0	R/W	TGR インタラプトイネーブル D チャンネル 0、3 で TSR の TGFD ビットが 1 にセットされたとき、TGFD ビットによる割り込み要求 (TGID) を許可または禁止します。チャンネル 1、2、4、5 ではリザーブビットです。リードすると常に 0 がリードされます。ライトは無効です。 0 : TGFD ビットによる割り込み要求 (TGID) を禁止 1 : TGFD ビットによる割り込み要求 (TGID) を許可

ビット	ビット名	初期値	R/W	説明
2	TGIEC	0	R/W	TGR インタラプトイネーブル C チャンネル 0、3 で TSR の TGFC ビットが 1 にセットされたとき、TGFC ビットによる割り込み要求 (TGIC) を許可または禁止します。 チャンネル 1、2、4、5 ではリザーブビットです。リードすると常に 0 がリードされます。ライトは無効です。 0 : TGFC ビットによる割り込み要求 (TGIC) を禁止 1 : TGFC ビットによる割り込み要求 (TGIC) を許可
1	TGIEB	0	R/W	TGR インタラプトイネーブル B TSR の TGFB ビットが 1 にセットされたとき、TGFB ビットによる割り込み要求 (TGIB) を許可または禁止します。 0 : TGFB ビットによる割り込み要求 (TGIB) を禁止 1 : TGFB ビットによる割り込み要求 (TGIB) を許可
0	TGIEA	0	R/W	TGR インタラプトイネーブル A TSR の TGFA ビットが 1 にセットされたとき、TGFA ビットによる割り込み要求 (TGIA) を許可または禁止します。 0 : TGFA ビットによる割り込み要求 (TGIA) を禁止 1 : TGFA ビットによる割り込み要求 (TGIA) を許可

10.3.5 タイマステータスレジスタ (TSR)

TSR は、各チャンネルのステータスの表示を行います。TPU には、各チャンネルに 1 本、計 6 本の TSR があります。

ビット	7	6	5	4	3	2	1	0
ビット名	TCFD	—	TCFU	TCFV	TGFD	TGFC	TGFB	TGFA
初期値 :	1	1	0	0	0	0	0	0
R/W :	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	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。

ビット	ビット名	初期値	R/W	説明
7	TCFD	1	R	カウント方向フラグ チャンネル 1、2、4、5 の TCNT のカウント方向を示すステータスフラグです。 チャンネル 0、3 ではリザーブビットです。リードすると常に 1 がリードされます。 ライトは無効です。 0 : TCNT はダウンカウント 1 : TCNT はアップカウント
6	—	1	R	リザーブビット このビットはリードのみ有効で、ライトは無効です。

ビット	ビット名	初期値	R/W	説明
5	TCFU	0	R/(W)*	<p>アンダフローフラグ</p> <p>チャンネル 1、2、4、5 が位相計数モードのとき、TCNT のアンダフローの発生を示すステータスフラグです。</p> <p>チャンネル 0、3 ではリザーブビットです。リードすると常に 0 がリードされます。ライトは無効です。</p> <p>[セット条件]</p> <p>TCNT の値がアンダフロー (H'0000→H'FFFF) したとき</p> <p>[クリア条件]</p> <p>TCFU=1 の状態で TCFU をリード後、TCFU に 0 をライトしたとき (割り込みを使用し、CPU によってクリアする場合、0 ライト後必ずフラグをリードしてください。)</p>
4	TCFV	0	R/(W)*	<p>オーバフローフラグ (TCFV)</p> <p>TCNT のオーバフローの発生を示すステータスフラグです。</p> <p>[セット条件]</p> <p>TCNT の値がオーバフローしたとき (H'FFFF→ H'0000)</p> <p>[クリア条件]</p> <p>TCFV=1 の状態で TCFV をリード後、TCFV に 0 をライトしたとき (割り込みを使用し、CPU によってクリアする場合、0 ライト後必ずフラグをリードしてください。)</p>
3	TGFD	0	R/(W)*	<p>インプットキャプチャ/アウトプットコンペアフラグ D</p> <p>チャンネル 0、3 の TGRD のインプットキャプチャまたはコンペアマッチの発生を示すステータスフラグです。</p> <p>チャンネル 1、2、4、5 ではリザーブビットです。リードすると常に 0 がリードされます。ライトは無効です。</p> <p>[セット条件]</p> <ul style="list-style-type: none"> • TGRD がアウトプットコンペアレジスタとして機能している場合、TCNT = TGRD になったとき • TGRD がインプットキャプチャとして機能している場合、インプットキャプチャ信号により TCNT の値が TGRD に転送されたとき <p>[クリア条件]</p> <ul style="list-style-type: none"> • TGID 割り込みにより DTC が起動され、DTC の MRB の DISSEL ビットが 0 のとき • TGFD=1 の状態で TGFD をリード後、TGFD に 0 をライトしたとき (割り込みを使用し、CPU によってクリアする場合、0 ライト後必ずフラグをリードしてください。)

ビット	ビット名	初期値	R/W	説明
2	TGFC	0	R/(W)*	<p>インプットキャプチャ／アウトプットコンペアフラグ C</p> <p>チャンネル 0、3 の TGRC のインプットキャプチャまたはコンペアマッチの発生を示すステータスフラグです。</p> <p>チャンネル 1、2、4、5 ではリザーブビットです。リードすると常に 0 がリードされます。ライトは無効です。</p> <p>[セット条件]</p> <ul style="list-style-type: none"> • TGRC がアウトプットコンペアレジスタとして機能している場合、TCNT = TGRC になったとき • TGRC がインプットキャプチャとして機能している場合、インプットキャプチャ信号により TCNT の値が TGRC に転送されたとき <p>[クリア条件]</p> <ul style="list-style-type: none"> • TGIC 割り込みにより DTC が起動され、DTC の MRB の DIESEL ビットが 0 のとき • TGFC=1 の状態で TGFC をリード後、TGFC に 0 をライトしたとき (割り込みを使用し、CPU によってクリアする場合、0 ライト後必ずフラグをリードしてください。)
1	TGFB	0	R/(W)*	<p>インプットキャプチャ／アウトプットコンペアフラグ B</p> <p>TGRB のインプットキャプチャまたはコンペアマッチの発生を示すステータスフラグです。</p> <p>[セット条件]</p> <ul style="list-style-type: none"> • TGRB がアウトプットコンペアレジスタとして機能している場合、TCNT = TGRB になったとき • TGRB がインプットキャプチャとして機能している場合、インプットキャプチャ信号により TCNT の値が TGRB に転送されたとき <p>[クリア条件]</p> <ul style="list-style-type: none"> • TGIB 割り込みにより DTC が起動され、DTC の MRB の DIESEL ビットが 0 のとき • TGFB=1 の状態で TGFB をリード後、TGFB に 0 をライトしたとき (割り込みを使用し、CPU によってクリアする場合、0 ライト後必ずフラグをリードしてください。)

ビット	ビット名	初期値	R/W	説明
0	TGFA	0	R/(W)*	インพุットキャプチャ/アウトプットコンペアフラグ A TGRA のインพุットキャプチャまたはコンペアマッチの発生を示すステータスフラグです。 [セット条件] ・ TGRA がアウトプットコンペアレジスタとして機能している場合、TCNT = TGRA になったとき ・ TGRA がインพุットキャプチャとして機能している場合、インพุットキャプチャ信号により TCNT の値が TGRA に転送されたとき [クリア条件] ・ TGIA 割り込みにより DTC が起動され、DTC の MRB の DISEL ビットが 0 のとき ・ TGFA=1 の状態で TGFA をリード後、TGFA に 0 をライトしたとき (割り込みを使用し、CPU によってクリアする場合、0 ライト後必ずフラグをリードしてください。)

【注】 * フラグをクリアするための 0 ライトのみ可能です。

10.3.6 タイマカウンタ (TCNT)

TCNT は 16 ビットのリード/ライト可能なカウンタです。各チャンネルに 1 本、計 6 本の TCNT があります。TCNT は、リセットまたはハードウェアスタンバイモード*時に H'0000 に初期化されます。TCNT の 8 ビット単位でのアクセスは禁止です。常に 16 ビット単位でアクセスしてください。

【注】 * 本 LSI には、ハードウェアスタンバイモードはありません。

ビット	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ビット名																
初期値:	0	0	0	0	0	0	0	0	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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

10.3.7 タイマジェネラルレジスタ (TGR)

TGR は 16 ビットのリード/ライト可能なアウトプットコンペア/インพุットキャプチャ兼用のレジスタです。チャンネル 0、3 に各 4 本、チャンネル 1、2、4、5 に各 2 本、計 16 本のジェネラルレジスタがあります。チャンネル 0、3 の TGRC と TGRD は、バッファレジスタとして動作設定することができます。TGR の 8 ビット単位でのアクセスは禁止です。常に 16 ビット単位でアクセスしてください。バッファ動作時の TGR とバッファレジスタの組み合わせは、TGRA-TGRC、TGRB-TGRD になります。

ビット	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ビット名																
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

10.3.8 タイマスタートレジスタ (TSTR)

TSTR は、チャンネル 0~5 の TCNT の動作/停止を選択します。TMDR へ動作モードを設定する場合や TCR へ TCNT のカウントクロックを設定する場合は、TCNT のカウンタ動作を停止してから行ってください。

ビット	7	6	5	4	3	2	1	0
ビット名	—	—	CST5	CST4	CST3	CST2	CST1	CST0
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

ビット	ビット名	初期値	R/W	説明
7	—	0	R/W	リザーブビット
6	—	0	R/W	リードすると常に 0 が読み出されます。ライトする値は常に 0 にしてください。
5	CST5	0	R/W	カウンタスタート 5~0
4	CST4	0	R/W	TCNT の動作または停止を選択します。
3	CST3	0	R/W	TIOC 端子を出力状態で動作中に、CST ビットに 0 をライトするとカウンタは
2	CST2	0	R/W	停止しますが、TIOC 端子のアウトプットコンペア出力レベルは保持されます。
1	CST1	0	R/W	CST ビットが 0 の状態で TIOR へのライトを行うと、設定した初期出力値に端
0	CST0	0	R/W	子の出力レベルが更新されます。 0 : TCNT_5~TCNT_0 のカウント動作は停止 1 : TCNT_5~TCNT_0 はカウント動作

10.3.9 タイマシンクロレジスタ (TSYR)

TSYR は、チャンネル 0~5 の TCNT の独立動作または同期動作を選択します。対応するビットを 1 にセットしたチャンネルが同期動作を行います。

ビット	7	6	5	4	3	2	1	0
ビット名	—	—	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0
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

ビット	ビット名	初期値	R/W	説明
7	—	0	R/W	リザーブビット
6	—	0	R/W	リードすると常に 0 が読み出されます。ライトする値は常に 0 にしてください。
5	SYNC5	0	R/W	タイマ同期 5~0
4	SYNC4	0	R/W	他のチャンネルとの独立動作または同期動作を選択します。
3	SYNC3	0	R/W	同期動作を選択すると、複数の TCNT の同期プリセットや、他チャンネルのカウントクリアによる同期クリアが可能となります。
2	SYNC2	0	R/W	
1	SYNC1	0	R/W	同期動作の設定には、最低 2 チャンネルの SYNC ビットを 1 にセットする必要があります。
0	SYNC0	0	R/W	同期クリアの設定には、SYNC ビットの他に TCR の CCLR2~CCLR0 ビットで、TCNT のクリア要因を設定する必要があります。
				0 : TCNT_5~TCNT_0 は独立動作 (TCNT のプリセット/クリアは他チャンネルと無関係)
				1 : TCNT_5~TCNT_0 は同期動作
				TCNT の同期プリセット/同期クリアが可能

10.4 動作説明

10.4.1 基本動作

各チャンネルには、TCNT と TGR があります。TCNT は、アップカウント動作を行い、フリーランニング動作、周期カウント動作、または外部イベントカウント動作が可能です。

TGR は、それぞれインプットキャプチャレジスタまたはアウトプットコンペアレジスタとして使用することができます。

(1) カウンタの動作

TSTR の CST0~CST5 ビットを1にセットすると、対応するチャンネルの TCNT はカウント動作を開始します。フリーランニングカウンタ動作、周期カウンタ動作などが可能です。

(a) カウント動作の設定手順例

カウント動作の設定手順例を図 10.3 に示します。

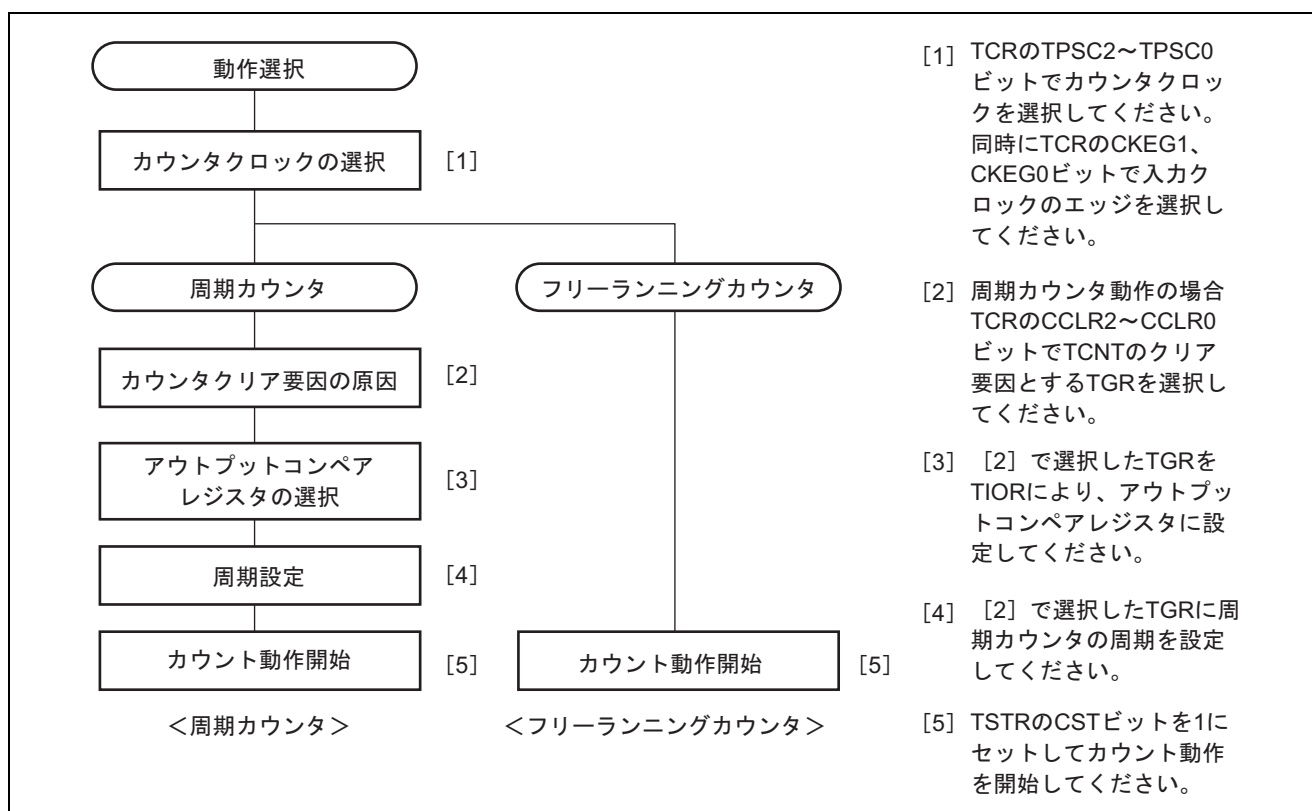


図 10.3 カウンタ動作設定手順例

(b) フリーランニングカウンタ動作と周期カウンタ動作

TPUのTCNTは、リセット直後はすべてフリーランニングカウンタの設定となっており、TSTRの対応するビットを1にセットするとフリーランニングカウンタとしてアップカウント動作を開始します。TCNTがオーバーフロー (H'FFFF→H'0000) すると、TSRのTCFVビットが1にセットされます。このとき、対応するTIERのTCIEVビットが1ならば、TPUは割り込みを要求します。TCNTはオーバーフロー後、H'0000からアップカウント動作を継続します。

フリーランニングカウンタの動作を図10.4に示します。

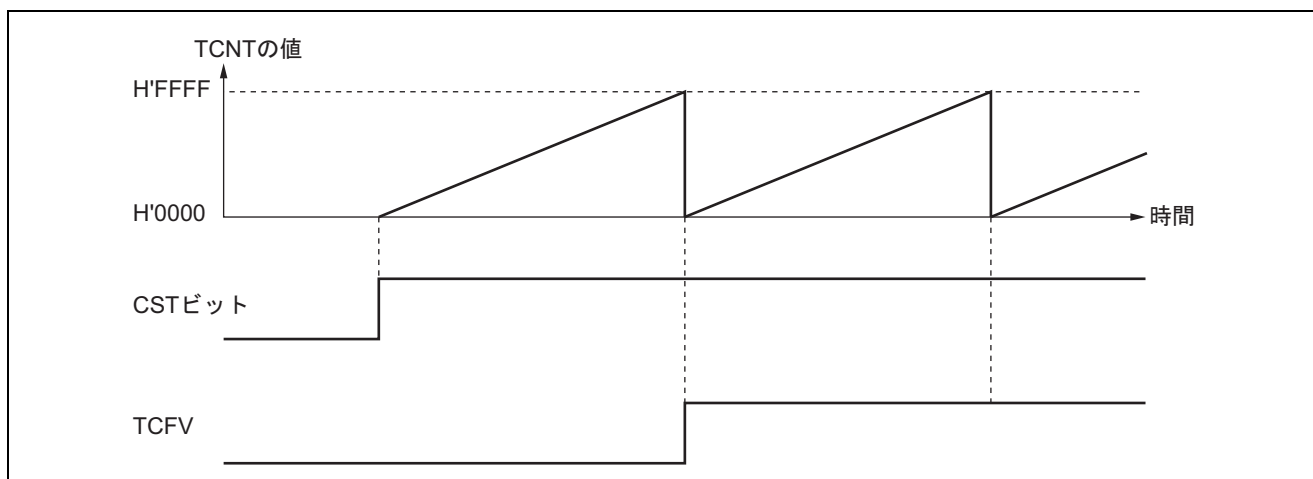


図 10.4 フリーランニングカウンタの動作

TCNTのクリア要因にコンペアマッチを選択したときは、対応するチャンネルのTCNTは周期カウンタ動作を行います。周期設定用のTGRをアウトプットコンペアレジスタに設定し、TCRのCCLR2～CCLR0ビットによりコンペアマッチによるカウンタクリアを選択します。設定後、TSTRの対応するビットを1にセットすると、周期カウンタとしてアップカウント動作を開始します。カウント値がTGRの値と一致すると、TSRのTGFビットが1にセットされ、TCNTはH'0000にクリアされます。

このとき対応するTIERのTGIEビットが1ならば、TPUは割り込みを要求します。TCNTはコンペアマッチ後、H'0000からアップカウント動作を継続します。

周期カウンタの動作を図 10.5 に示します。

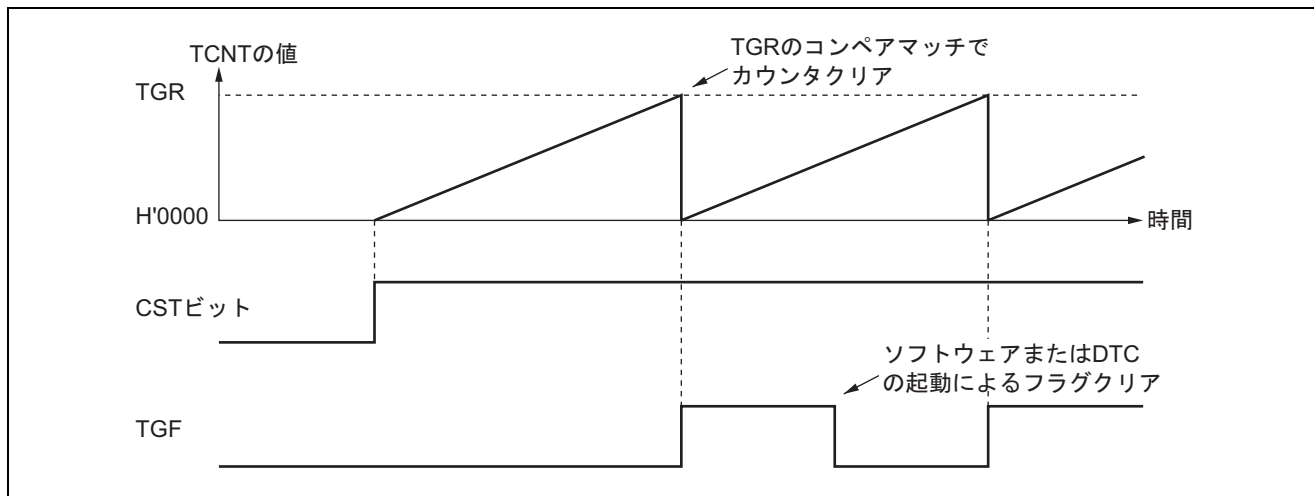


図 10.5 周期カウンタの動作

(2) コンペアマッチによる波形出力機能

TPU は、コンペアマッチにより対応する出力端子から 0 出力/1 出力/トグル出力を行うことができます。

【注】 本 LSI では、チャンネル 3、チャンネル 4 およびチャンネル 5 の TIOCA3、TIOCB3、TIOCC3、TIOCD3、TIOCA4、TIOCB4、TIOCA5、TIOCB5 入出力端子はありません。
 そのため、コンペアマッチによる波形出力（0 出力/1 出力/トグル出力）、PWM 波形出力を行うことができません。

(a) コンペアマッチによる波形出力動作の設定手順例

コンペアマッチによる波形出力動作の設定手順例を図 10.6 に示します。

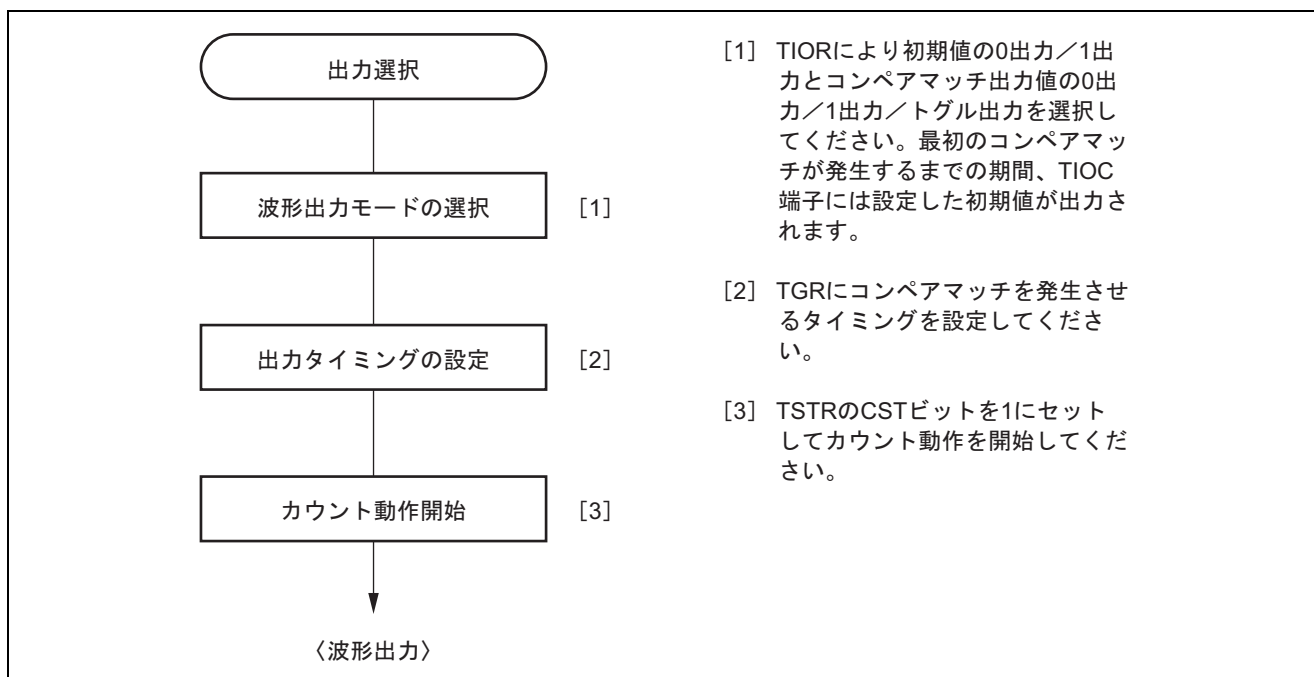


図 10.6 コンペアマッチによる波形出力動作例

(b) 波形出力動作例

0出力/1出力例を図10.7に示します。

TCNTをフリーランニングカウンタ動作とし、コンペアマッチAにより1出力、コンペアマッチBにより0出力となるように設定した場合の例です。設定したレベルと端子のレベルが一致した場合には、端子のレベルは変化しません。

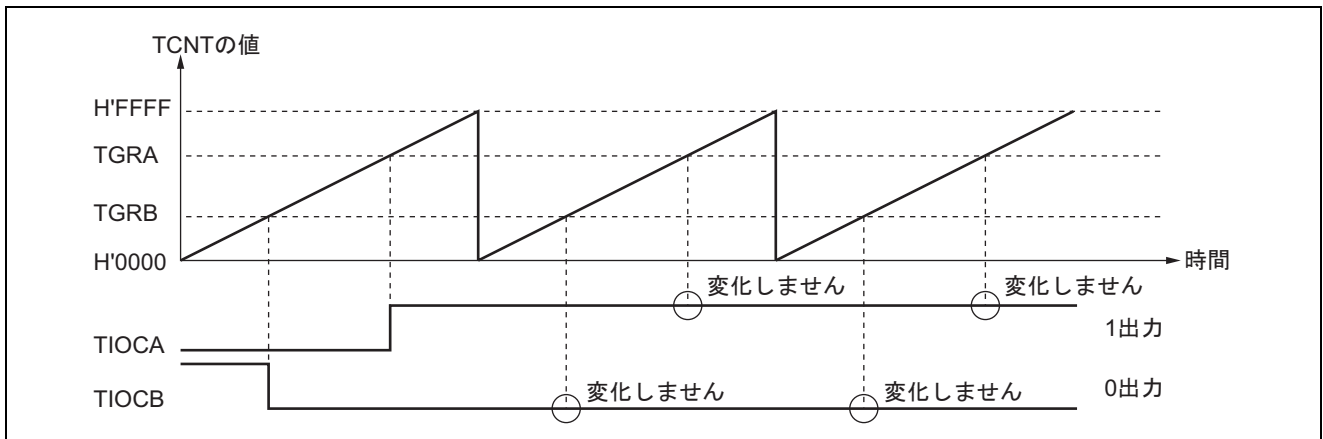


図10.7 0出力/1出力の動作例

トグル出力の例を図10.8に示します。

TCNTを周期カウンタ動作（コンペアマッチBによりカウンタクリア）に、コンペアマッチA、Bともトグル出力となるように設定した場合の例です。

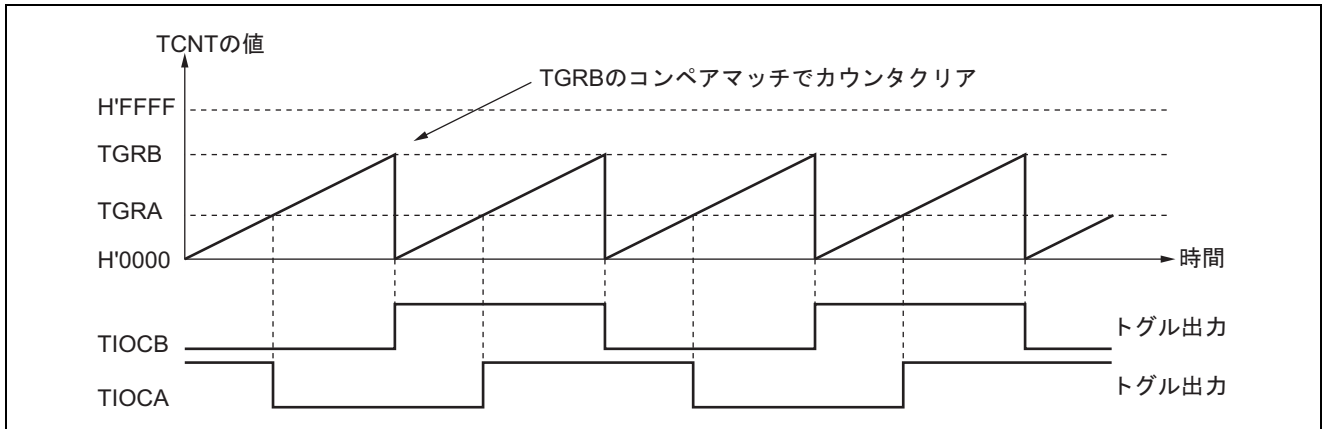


図10.8 トグル出力の動作例

(3) インพุットキャプチャ機能

TIOC 端子の入力エッジを検出して TCNT の値を TGR に転送することができます。

検出エッジは立ち上がりエッジ/立ち下がりエッジ/両エッジから選択できます。また、チャンネル 0、1、3、4 は別のチャンネルのカウンタ入力クロックやコンペアマッチ信号をインพุットキャプチャの要因とすることもできます。

【注】 チャンネル 0、3 で別のチャンネルのカウンタ入力クロックをインพุットキャプチャ入力とする場合は、インพุットキャプチャ入力とするカウンタ入力クロックに Pφ/1 を選択しないでください。Pφ/1 を選択した場合は、インพุットキャプチャは発生しません。

(a) インพุットキャプチャ動作の設定手順例

インพุットキャプチャ動作の設定手順例を図 10.9 に示します。

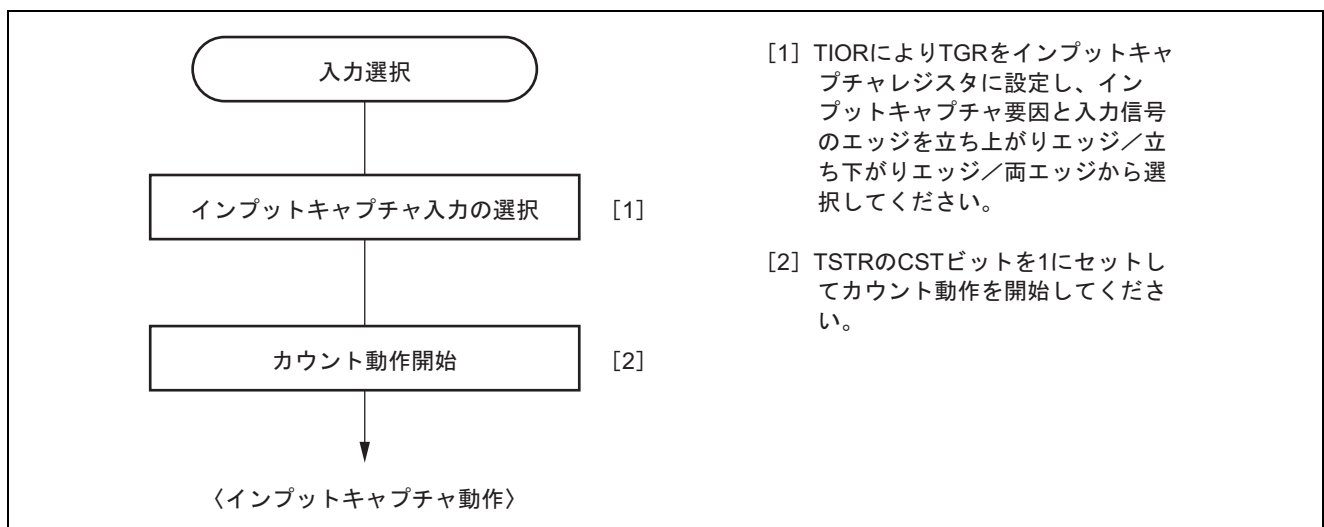


図 10.9 インพุットキャプチャ動作の設定例

(b) インพุットキャプチャ動作例

インพุットキャプチャ動作例を図 10.10 に示します。

TIOCA 端子のインพุットキャプチャ入力エッジは立ち上がり/立ち下りの両エッジ、また TIOCB 端子のインพุットキャプチャ入力エッジは立ち下りエッジを選択し、TCNT は TGRB のインพุットキャプチャでカウンタクリアされるように設定した場合の例です。

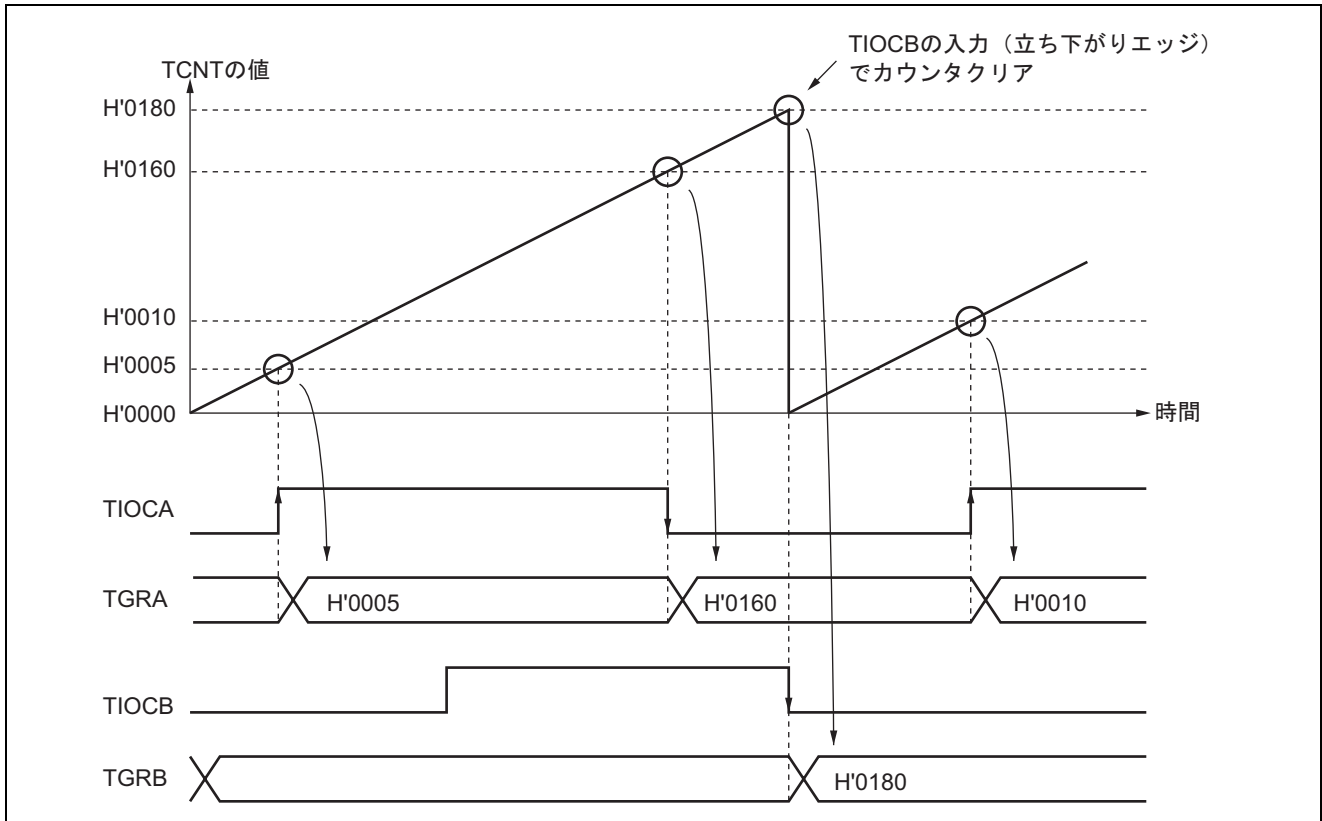


図 10.10 インพุットキャプチャ動作例

10.4.2 同期動作

同期動作は、複数の TCNT の値を同時に書き換えることができます (同期プリセット)。また、TCR の設定により複数の TCNT を同時にクリアすることができます (同期クリア)。

同期動作により、1つのタイムベースに対して TGR を増加することができます。

チャンネル 0~5 はすべて同期動作の設定が可能です。

(1) 同期動作の設定手順例

同期動作の設定手順例を図 10.11 に示します。

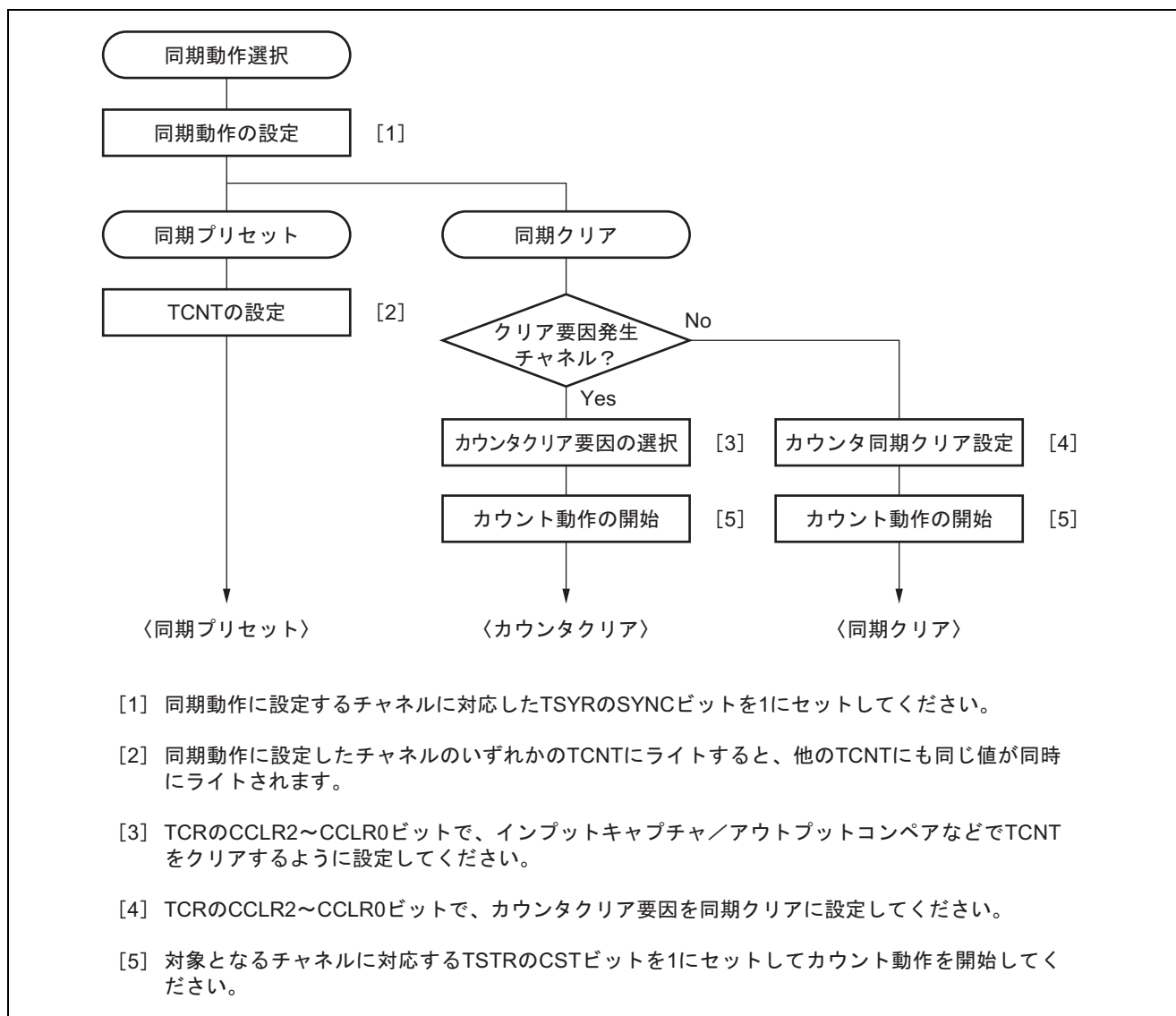


図 10.11 同期動作の設定手順例

(2) 同期動作の例

同期動作の例を図 10.12 に示します。

チャンネル 0~2 を同期動作かつ PWM モード 1 に設定し、チャンネル 0 のカウンタクリア要因を TGRB_0 のコンペアマッチ、またチャンネル 1、2 のカウンタクリア要因を同期クリアに設定した場合の例です。

3 相の PWM 波形を TIOCA0、TIOCA1、TIOCA2 端子から出力します。このとき、チャンネル 0~2 の TCNT は同期プリセット、TGRB_0 のコンペアマッチによる同期クリアを行い、TGRB_0 に設定したデータが PWM 周期となります。

PWM モードについては、「10.4.5 PWM モード」を参照してください。

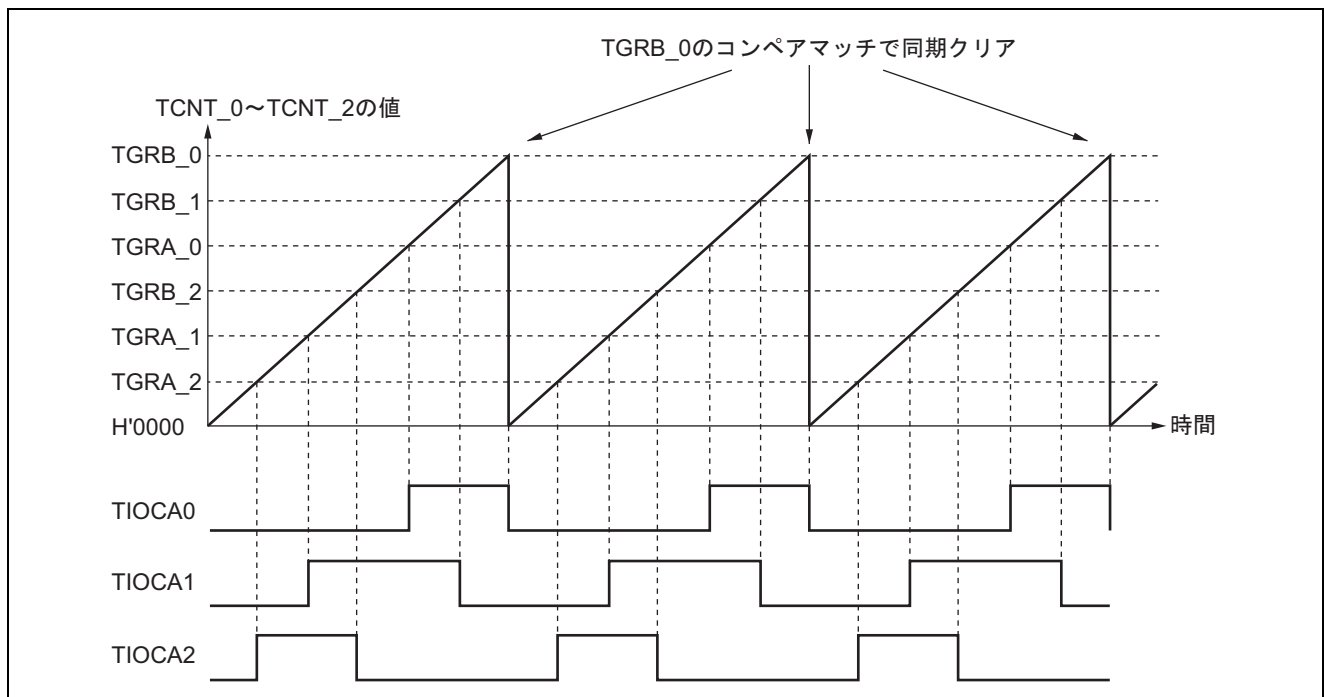


図 10.12 同期動作の動作例

10.4.3 バッファ動作

バッファ動作は、チャンネル0、3が持つ機能です。TGRCとTGRDをバッファレジスタとして使用することができます。

バッファ動作は、TGRをインプットキャプチャレジスタに設定した場合と、コンペアマッチレジスタに設定した場合のそれぞれで動作内容が異なります。

表 10.30 にバッファ動作時のレジスタの組み合わせを示します。

表 10.30 レジスタの組み合わせ

チャンネル	タイマジェネラルレジスタ	バッファレジスタ
0	TGRA_0	TGRC_0
	TGRB_0	TGRD_0
3	TGRA_3	TGRC_3
	TGRB_3	TGRD_3

• TGRがアウトプットコンペアレジスタの場合

コンペアマッチが発生すると、対応するチャンネルのバッファレジスタの値がタイマジェネラルレジスタに転送されます。

この動作を図 10.13 に示します。

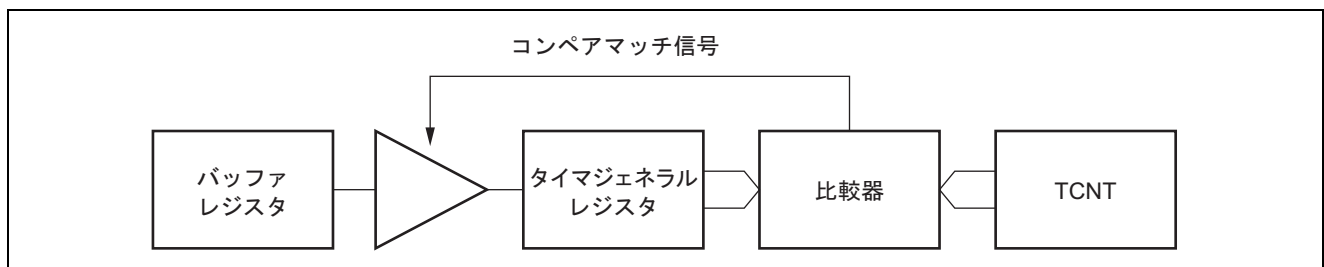


図 10.13 コンペアマッチバッファ動作

• TGRがインプットキャプチャレジスタの場合

インプットキャプチャが発生すると、TCNT の値を TGR に転送すると同時に、それまで格納されていた TGR の値をバッファレジスタに転送します。

この動作を図 10.14 に示します。

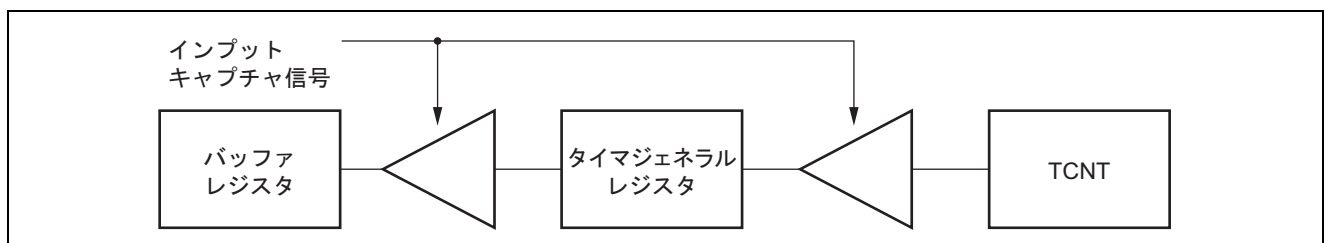


図 10.14 インプットキャプチャバッファ動作

(1) バッファ動作の設定手順例

バッファ動作の設定手順例を図 10.15 に示します。

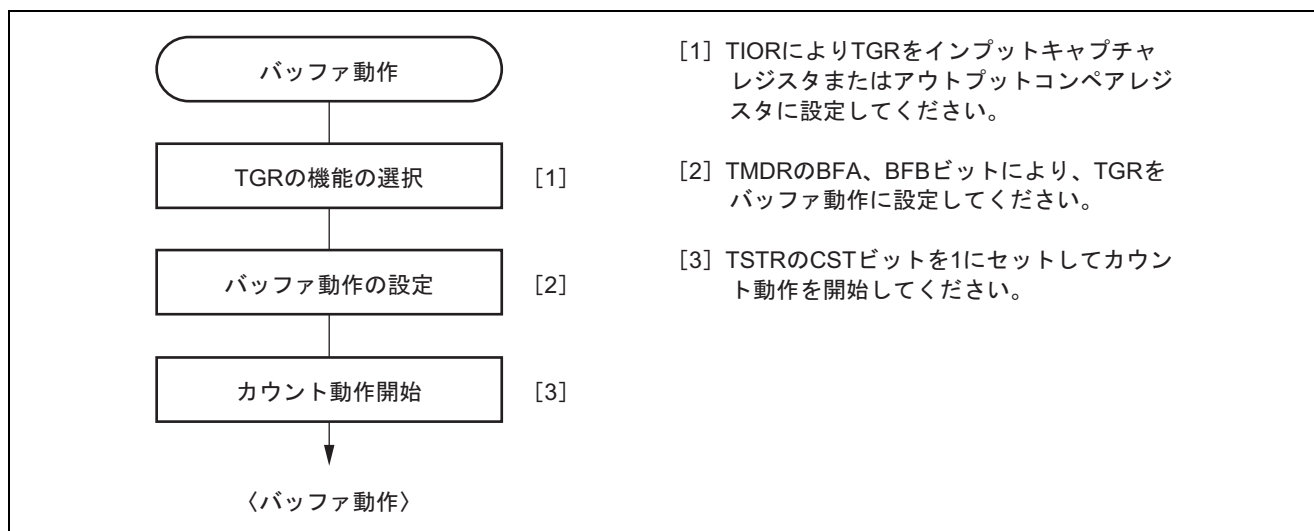


図 10.15 バッファ動作の設定手順例

(2) バッファ動作例

(a) TGR がアウトプットコンペアレジスタの場合

チャンネル0をPWMモード1に設定し、TGRAとTGRCをバッファ動作に設定した場合の動作例を図10.16に示します。TCNTはコンペアマッチBによりクリア、出力はコンペアマッチAで1出力、コンペアマッチBで0出力に設定した例です。

バッファ動作が設定されているため、コンペアマッチAが発生すると出力を変化させると同時に、バッファレジスタTGRCの値がタイマジェネラルレジスタTGRAに転送されます。この動作は、コンペアマッチAが発生する度に繰り返されます。

PWMモードについては、「10.4.5 PWMモード」を参照してください。

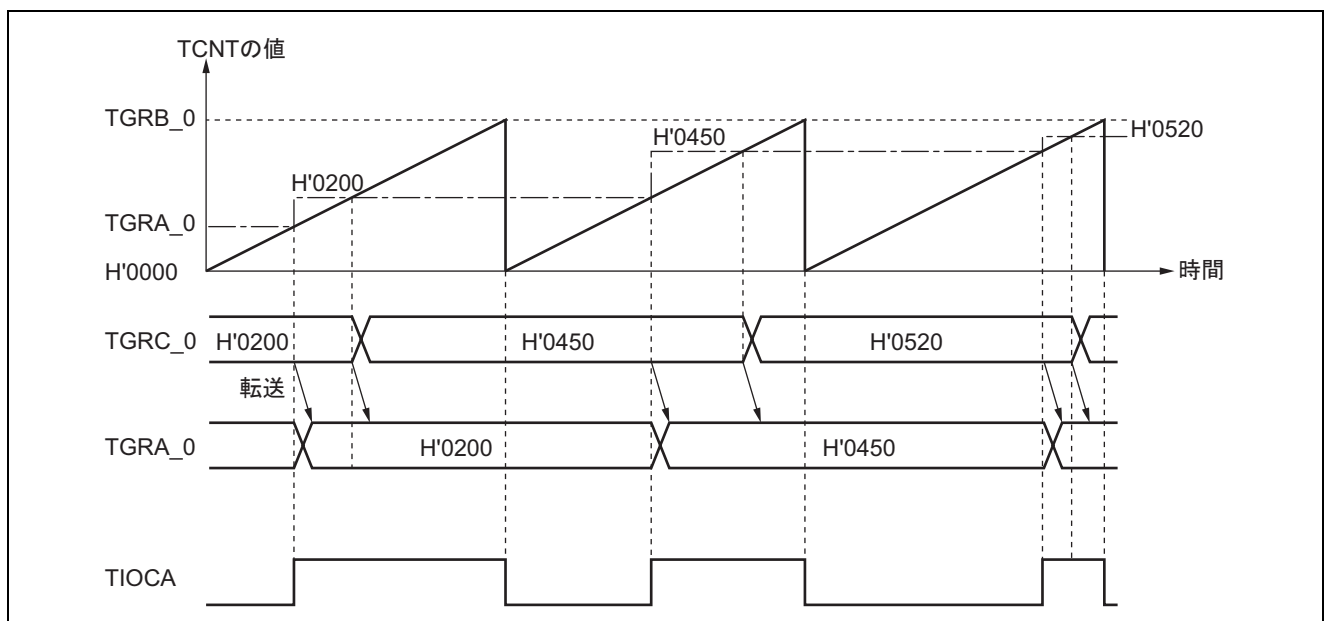


図 10.16 バッファ動作例 (1)

(b) TGR が入力キャプチャレジスタの場合

TGRA を入力キャプチャレジスタに設定し、TGRA と TGRC をバッファ動作に設定したときの動作例を図 10.17 に示します。

TCNT は TGRA の入力キャプチャでカウンタクリア、TIOCA 端子の入力キャプチャ入力エッジは立ち上がりエッジ/立ち下がりエッジの両エッジが選択されています。

バッファ動作が設定されているため、入力キャプチャ A により TCNT の値が TGRA に格納されると同時に、それまで TGRA に格納されていた値が TGRC に転送されます。

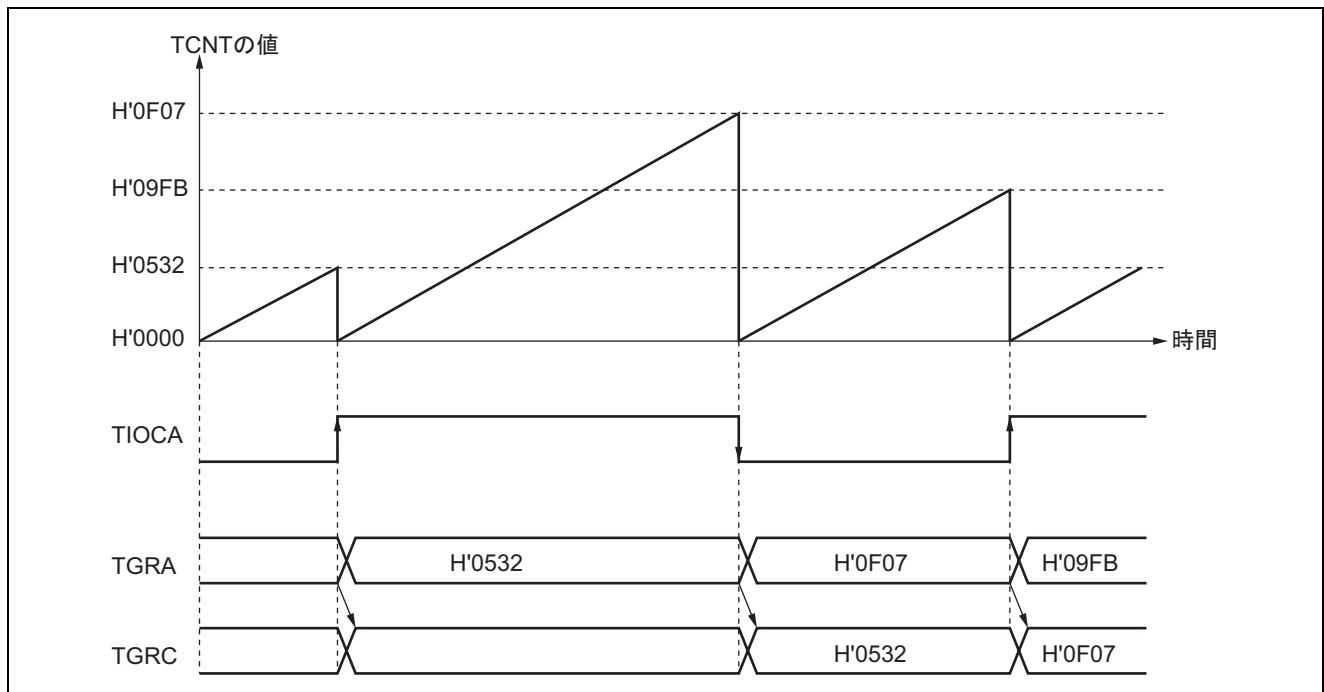


図 10.17 バッファ動作例 (2)

10.4.4 カスケード接続動作

カスケード接続動作は、2チャンネルの16ビットカウンタを接続して32ビットカウンタとして動作させる機能です。

この機能は、チャンネル1 (チャンネル4) のカウンタクロックをTCRのTPSC2~TPSC0ビットでTCNT_2 (TCNT_5) のオーバーフロー/アンダフローでカウントに設定することにより動作します。

アンダフローが発生するのは、下位16ビットのTCNTが位相計数モードのときのみです。

表10.31にカスケード接続の組み合わせを示します。

【注】 チャンネル1、4を位相計数モードに設定した場合は、カウンタクロックの設定は無効となり、独立して位相計数モードで動作します。

表 10.31 カスケード接続組み合わせ

組み合わせ	上位 16 ビット	下位 16 ビット
チャンネル1とチャンネル2	TCNT_1	TCNT_2
チャンネル4とチャンネル5	TCNT_4	TCNT_5

(1) カスケード接続動作の設定手順例

カスケード接続動作の設定手順例を図10.18に示します。

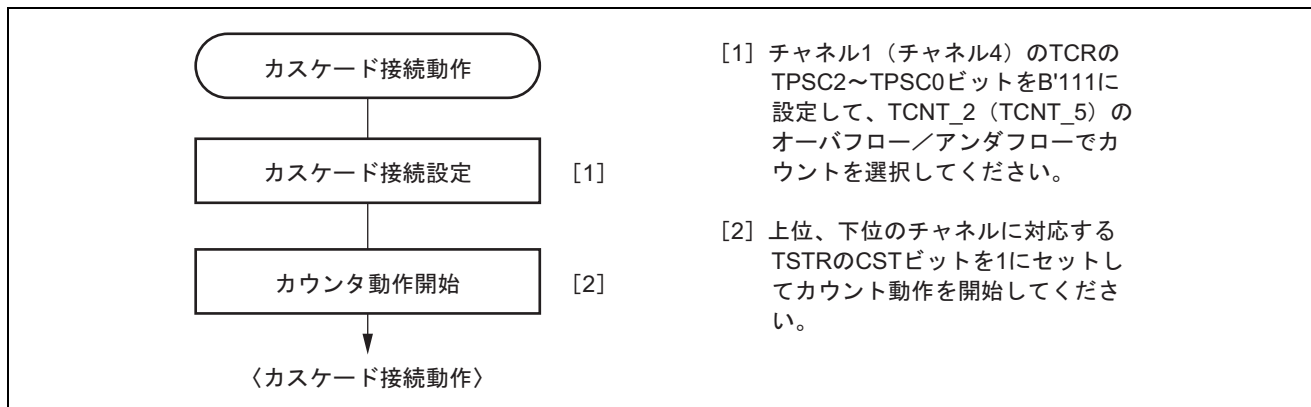


図 10.18 カスケード接続動作設定手順例

(2) カスケード接続動作例

TCNT_1 は TCNT_2 のオーバフロー/アンダフローでカウント、TGRA_1 と TGRA_2 をインプットキャプチャレジスタに設定し、TIOC 端子の立ち上がりエッジを選択した時の動作を図 10.19 に示します。

TIOCA1 端子と TIOCA2 端子に立ち上がりエッジを同時に入力することにより、TGRA_1 に上位 16 ビット、TGRA_2 に下位 16 ビットの 32 ビットデータが転送されます。

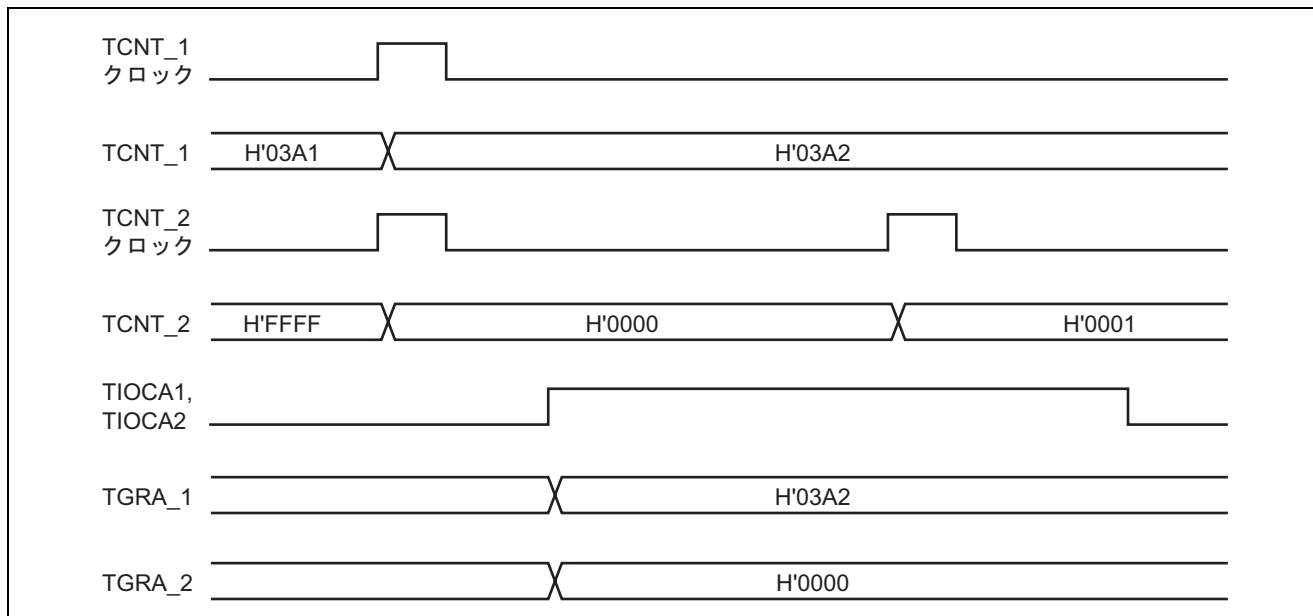


図 10.19 カスケード接続動作例 (1)

TCNT_1 は TCNT_2 のオーバフロー/アンダフローでカウント、チャンネル 2 を位相計数モードに設定した時の動作を図 10.20 に示します。

TCNT_1 は、TCNT_2 のオーバフローでアップカウント、TCNT_2 のアンダフローでダウンカウントされます。

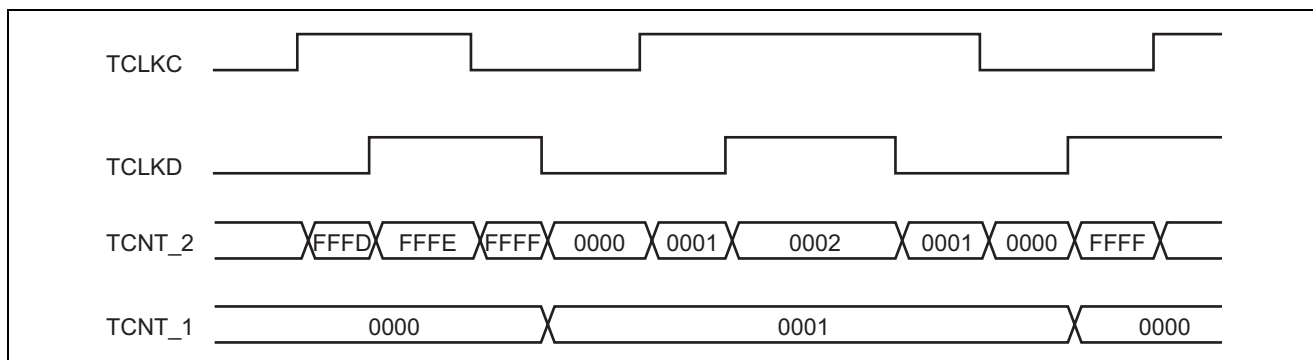


図 10.20 カスケード接続動作例 (2)

10.4.5 PWM モード

PWM モードは出力端子よりそれぞれ PWM 波形を出力するモードです。各 TGR のコンペアマッチによる出力レベルは 0 出力/1 出力/トグル出力の中から選択可能です。

各 TGR の設定により、デューティ 0~100% の PWM 波形が出力できます。

TGR のコンペアマッチをカウンタクリア要因とすることにより、そのレジスタに周期を設定することができます。全チャンネル独立に PWM モードに設定できます。同期動作も可能です。

PWM モードは次に示す 2 種類あります。

(a) PWM モード 1

TGRA と TGRB、TGRC と TGRD をペアで使用して、TIOCA、TIOCC 端子から PWM 出力を生成します。TIOCA、TIOCC 端子からコンペアマッチ A、C によって TIOR の IOA3~IOA0、IOC3~IOC0 ビットで指定した出力を、また、コンペアマッチ B、D によって TIOR の IOB3~IOB0、IOD3~IOD0 ビットで指定した出力を行います。初期出力値は TGRA、TGRC に設定した値になります。ペアで使用する TGR の設定値が同一の場合、コンペアマッチが発生しても出力値は変化しません。

PWM モード 1 では、最大 8 相の PWM 出力が可能です。

(b) PWM モード 2

TGR の 1 本を周期レジスタ、他の TGR をデューティレジスタに使用して PWM 出力を生成します。コンペアマッチによって、TIOR で指定した出力を行います。また、同期レジスタのコンペアマッチによるカウンタのクリアで各端子の出力値は TIOR で設定した初期値が出力されます。周期レジスタとデューティレジスタの設定値が同一の場合、コンペアマッチが発生しても出力値は変化しません。

PWM モード 2 では、同期動作と併用することにより最大 15 相の PWM 出力が可能です。

PWM 出力端子とレジスタの対応を表 10.32 に示します。

表 10.32 各 PWM 出力のレジスタと出力端子

チャンネル	レジスタ	出力端子	
		PWM モード 1	PWM モード 2
0	TGRA_0	TIOCA0	TIOCA0
	TGRB_0		TIOCB0
	TGRC_0	TIOCC0	TIOCC0
	TGRD_0		TIOCD0
1	TGRA_1	TIOCA1	TIOCA1
	TGRB_1		TIOCB1
2	TGRA_2	TIOCA2	TIOCA2
	TGRB_2		TIOCB2
3	TGRA_3	TIOCA3	TIOCA3
	TGRB_3		TIOCB3
	TGRC_3	TIOCC3	TIOCC3
	TGRD_3		TIOCD3
4	TGRA_4	TIOCA4	TIOCA4
	TGRB_4		TIOCB4
5	TGRA_5	TIOCA5	TIOCA5
	TGRB_5		TIOCB5

【注】 PWM モード 2 のとき、周期を設定した TGR の PWM 出力はできません。

(1) PWM モードの設定手順例

PWM モードの設定手順例を図 10.21 に示します。

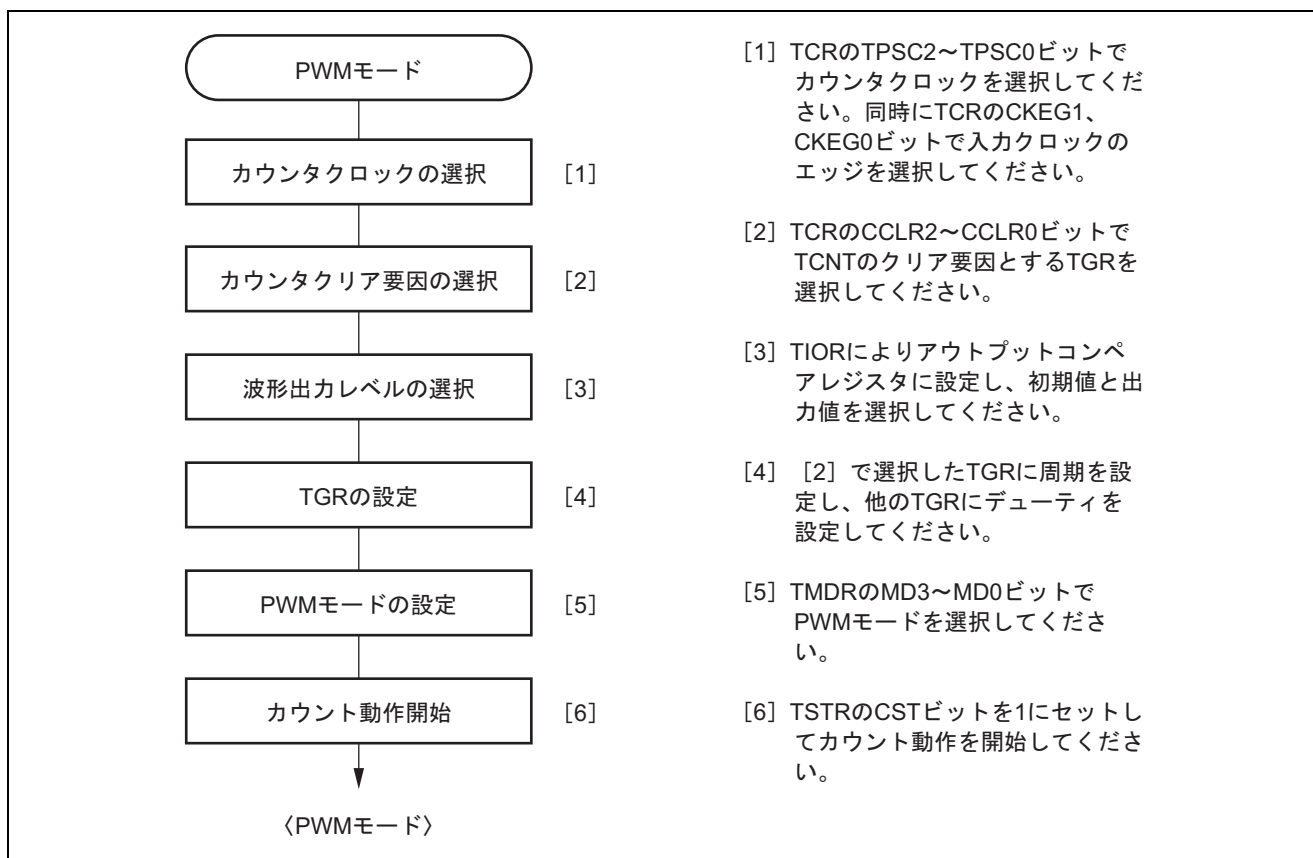


図 10.21 PWM モードの設定手順例

(2) PWM モードの動作例

PWM モード 1 の動作例を図 10.22 に示します。

この図は、TCNT のクリア要因を TGRA のコンペアマッチとし、TGRA の初期出力値と出力値を 0、TGRB の出力値を 1 に設定した場合の例です。

この場合、TGRA に設定した値が周期となり、TGRB に設定した値がデューティになります。

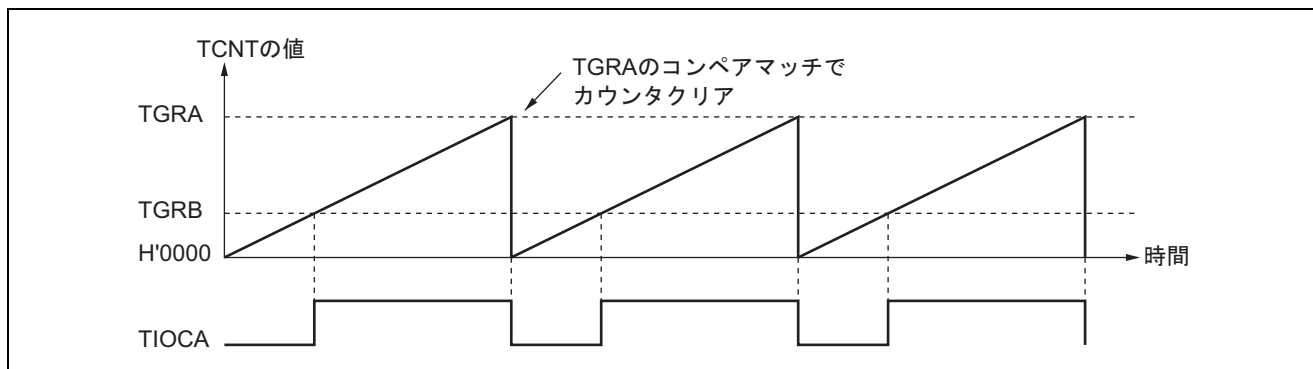


図 10.22 PWM モードの動作例 (1)

PWM モード2の動作例を図 10.23 に示します。

この図は、チャンネル0と1を同期動作させ、TCNTのクリア要因をTGRB_1のコンペアマッチとし、他のTGR (TGRA_0~TGRD_0、TGRA_1)の初期出力値を0、出力値を1に設定して5相のPWM波形を出力させた場合の例です。

この場合、TGRB_1に設定した値が周期となり、他のTGRに設定した値がデューティになります。

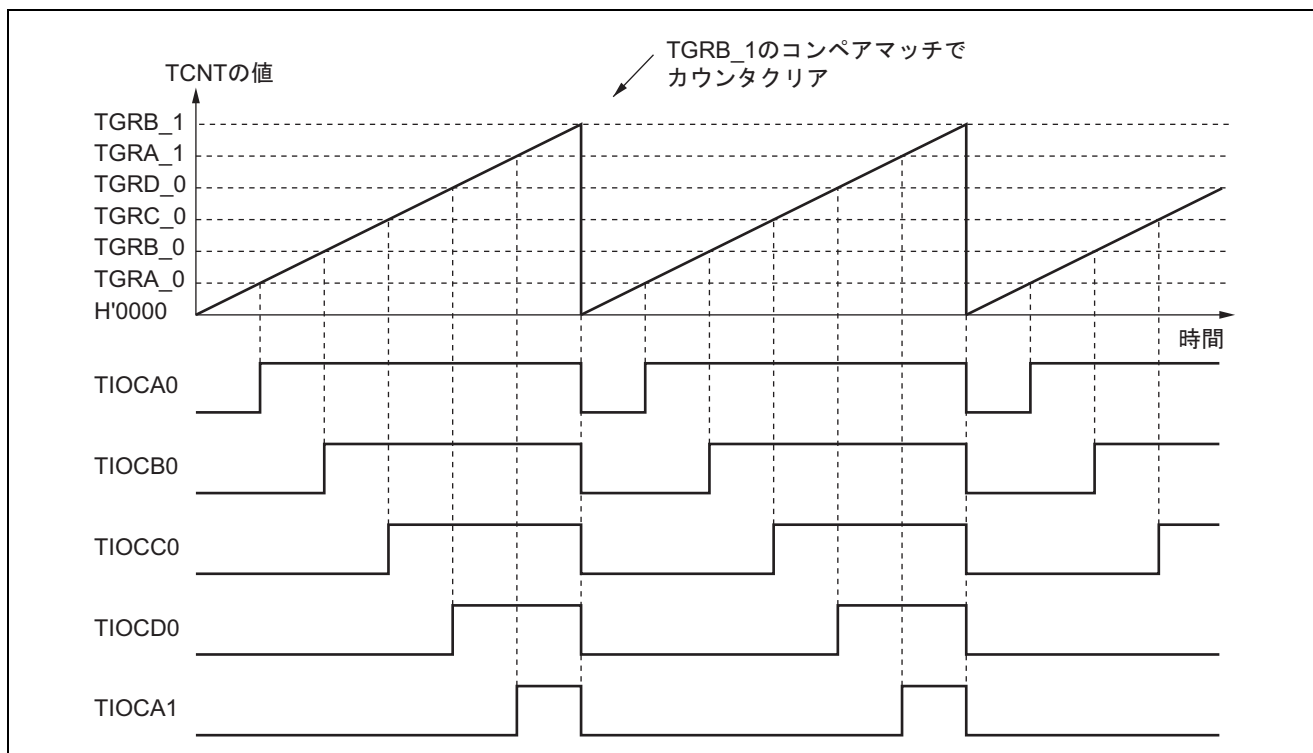


図 10.23 PWM モードの動作例 (2)

PWM モードで、デューティ 0%、デューティ 100%の PWM 波形を出力する例を図 10.24 に示します。

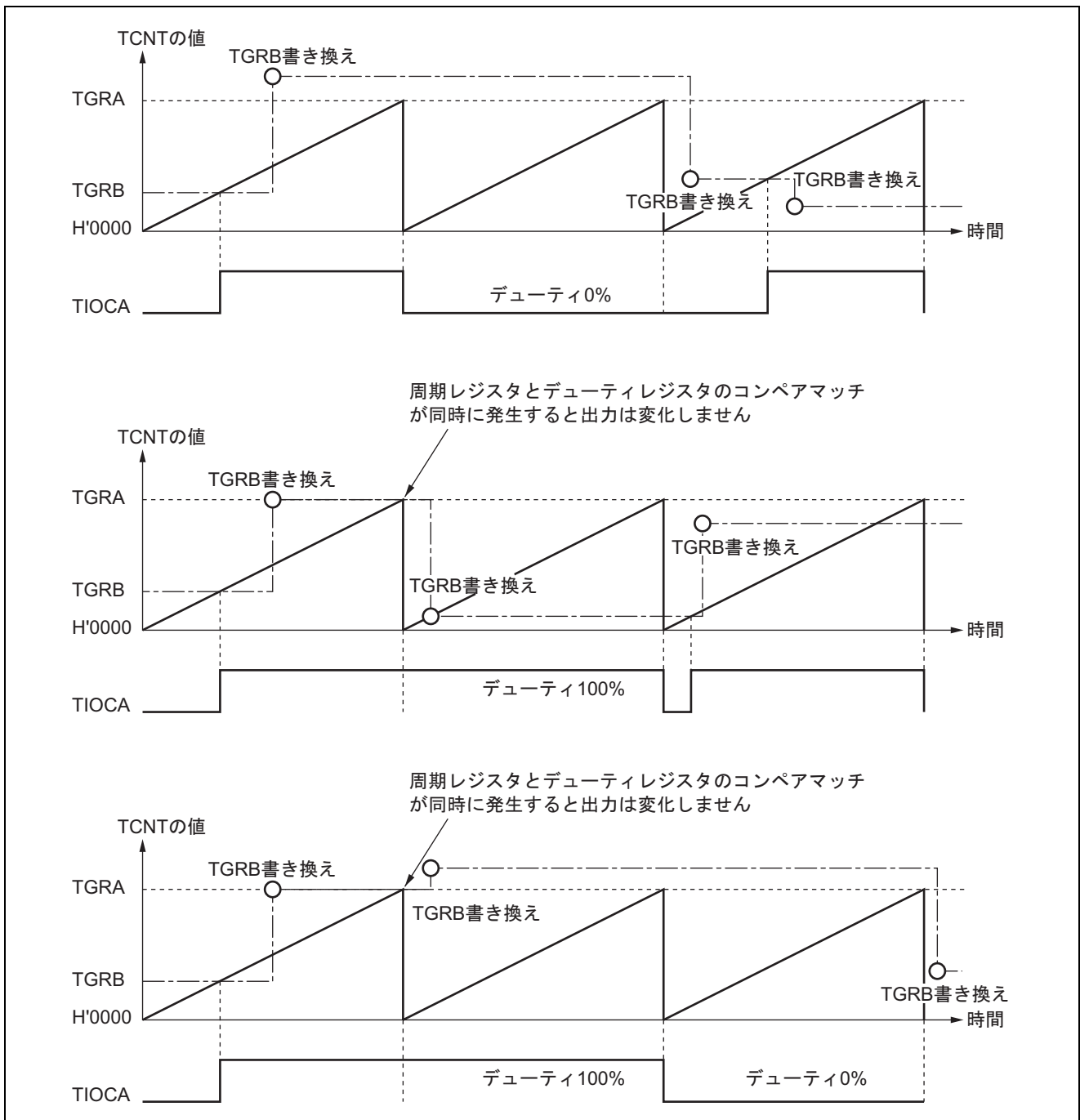


図 10.24 PWM モード動作例 (3)

10.4.6 位相計数モード

位相計数モードは、チャンネル 1、2、4、5 の設定により、2 本の外部クロック入力の位相差を検出し、TCNT をアップ/ダウンカウントします。

位相計数モードに設定すると、TCR の TPSC2~TPSC0 ビット、CKEG1、CKEG0 ビットの設定にかかわらずカウンタ入力クロックは外部クロックを選択し、TCNT はアップ/ダウンカウンタとして動作します。ただし、TCR の CCLR1、CCLR0 ビット、TIOR、TIER、TGR の機能は有効ですので、インプットキャプチャ/コンペアマッチ機能や割り込み機能は使用することができます。

2 相エンコーダパルスの入力として使用できます。

TCNT がアップカウント時、オーバフローが発生するとすると TSR の TCFV フラグがセットされます。また、ダウンカウント時にアンダフローが発生すると、TCFU フラグがセットされます。

TSR の TCFD ビットはカウント方向フラグです。TCFD フラグをリードすることにより、TCNT がアップカウントしているかダウンカウントしているかを確認することができます。

表 10.33 に外部クロック端子とチャンネルの対応を示します。

表 10.33 位相計数モードクロック入力端子

チャンネル	外部クロック端子	
	A 相	B 相
チャンネル 1 または 5 を位相計数モードとするとき	TCLKA	TCLKB
チャンネル 2 または 4 を位相計数モードとするとき	TCLKC	TCLKD

(1) 位相計数モードの設定手順例

位相計数モードの設定手順例を図 10.25 に示します。

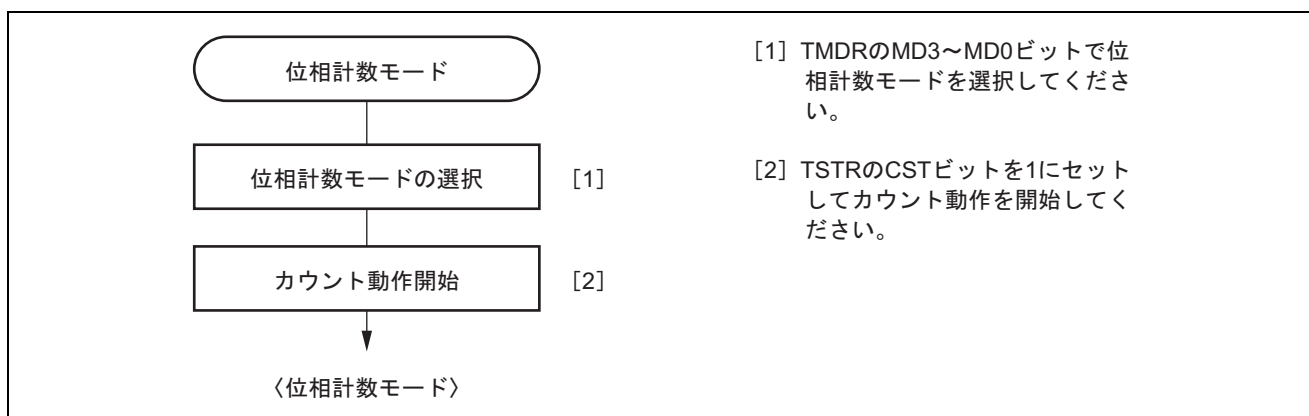


図 10.25 位相計数モードの設定手順例

(2) 位相計数モードの動作例

位相計数モードでは、2本の外部クロックの位相差で TCNT がアップ/ダウンカウントします。また、カウント条件により 4つのモードがあります。

(a) 位相計数モード 1

位相計数モード 1 の動作例を図 10.26 に、TCNT のアップ/ダウンカウント条件を表 10.34 に示します。

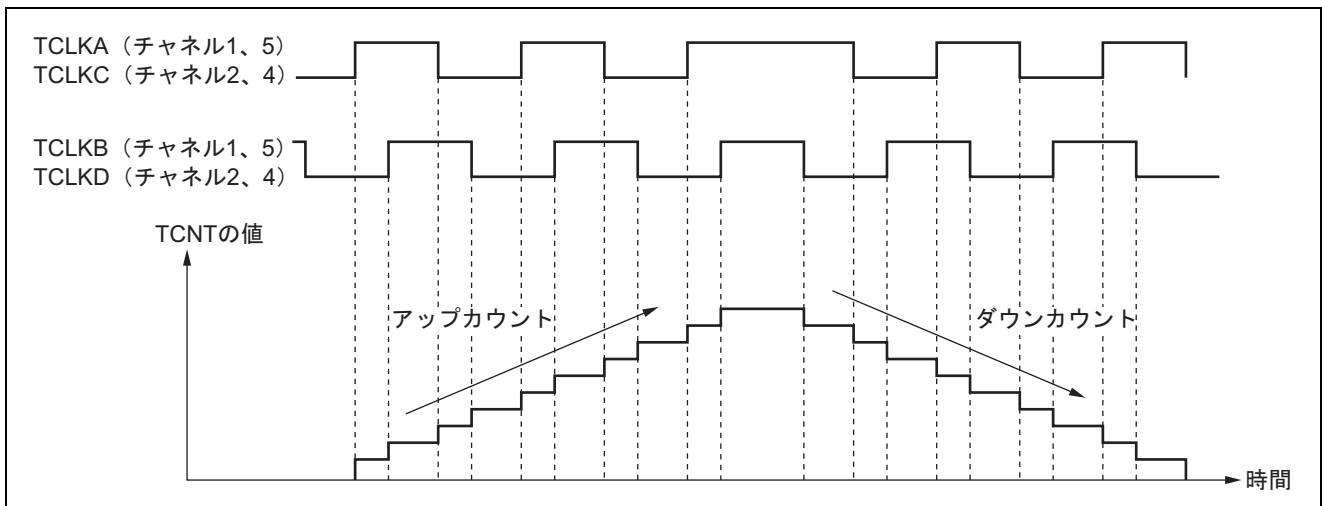


図 10.26 位相計数モード 1 の動作例

表 10.34 位相計数モード 1 のアップ/ダウンカウント条件

TCLKA (チャンネル 1、5) TCLKC (チャンネル 2、4)	TCLKB (チャンネル 1、5) TCLKD (チャンネル 2、4)	動作内容
High レベル	↑	アップカウント
Low レベル	↓	
↑	Low レベル	
↓	High レベル	
High レベル	↓	ダウンカウント
Low レベル	↑	
↑	High レベル	
↓	Low レベル	

【記号説明】

↑ : 立ち上がりエッジ

↓ : 立ち下がりエッジ

(b) 位相計数モード2

位相計数モード2の動作例を図10.27に、TCNTのアップ/ダウンカウント条件を表10.35に示します。

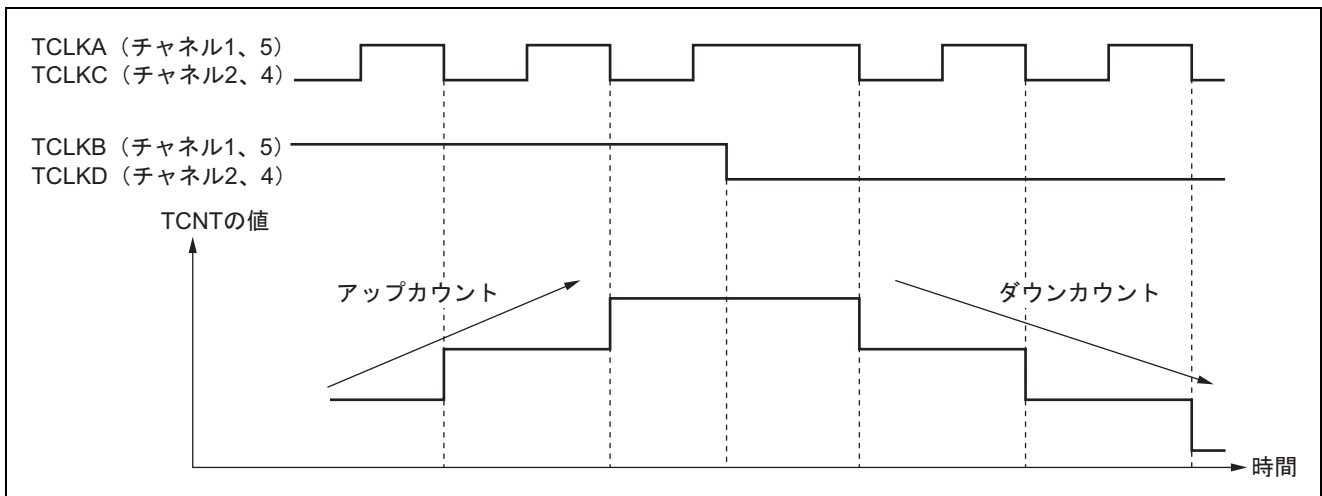


図 10.27 位相計数モード2の動作例

表 10.35 位相計数モード2のアップ/ダウンカウント条件

TCLKA (チャンネル1、5) TCLKC (チャンネル2、4)	TCLKB (チャンネル1、5) TCLKD (チャンネル2、4)	動作内容
High レベル	⬆	Don't care
Low レベル	⬇	Don't care
⬆	Low レベル	Don't care
⬇	High レベル	アップカウント
High レベル	⬇	Don't care
Low レベル	⬆	Don't care
⬆	High レベル	Don't care
⬇	Low レベル	ダウンカウント

【記号説明】

⬆: 立ち上がりエッジ

⬇: 立ち下がりエッジ

(c) 位相計数モード3

位相計数モード3の動作例を図10.28に、TCNTのアップ/ダウンカウント条件を表10.36に示します。

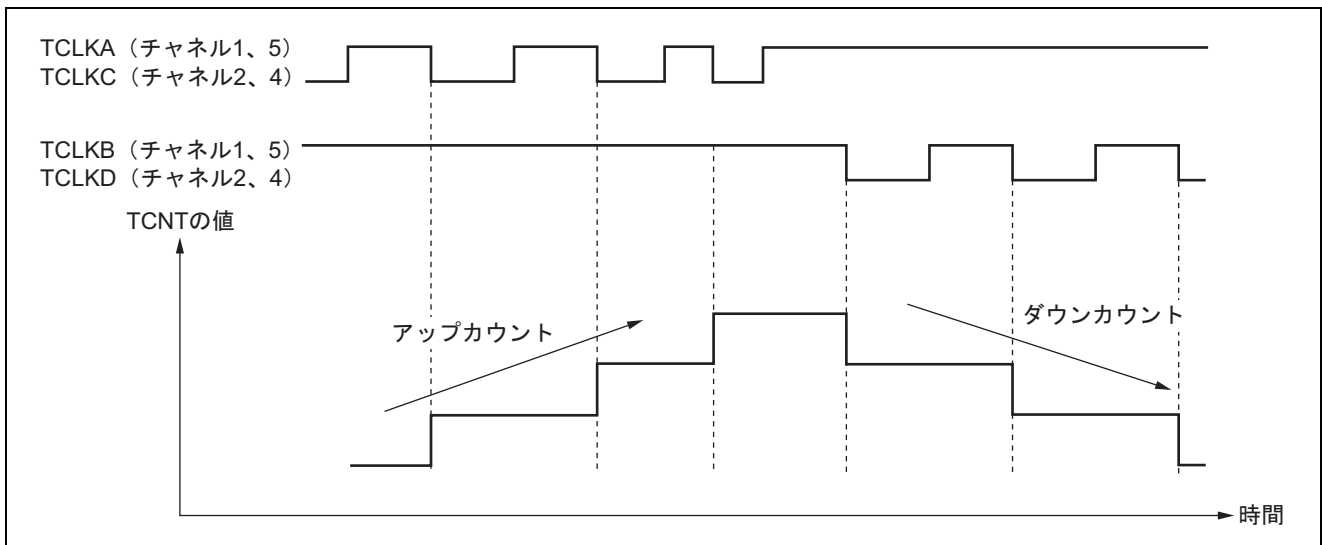


図 10.28 位相計数モード3の動作例

表 10.36 位相計数モード3のアップ/ダウンカウント条件

TCLKA (チャンネル 1, 5) TCLKC (チャンネル 2, 4)	TCLKB (チャンネル 1, 5) TCLKD (チャンネル 2, 4)	動作内容
High レベル	\uparrow	Don't care
Low レベル	\downarrow	Don't care
\uparrow	Low レベル	Don't care
\downarrow	High レベル	アップカウント
High レベル	\downarrow	ダウンカウント
Low レベル	\uparrow	Don't care
\uparrow	High レベル	Don't care
\downarrow	Low レベル	Don't care

【記号説明】

\uparrow : 立ち上がりエッジ

\downarrow : 立ち下がりエッジ

(d) 位相計数モード4

位相計数モード4の動作例を図10.29に、TCNTのアップ/ダウンカウント条件を表10.37に示します。

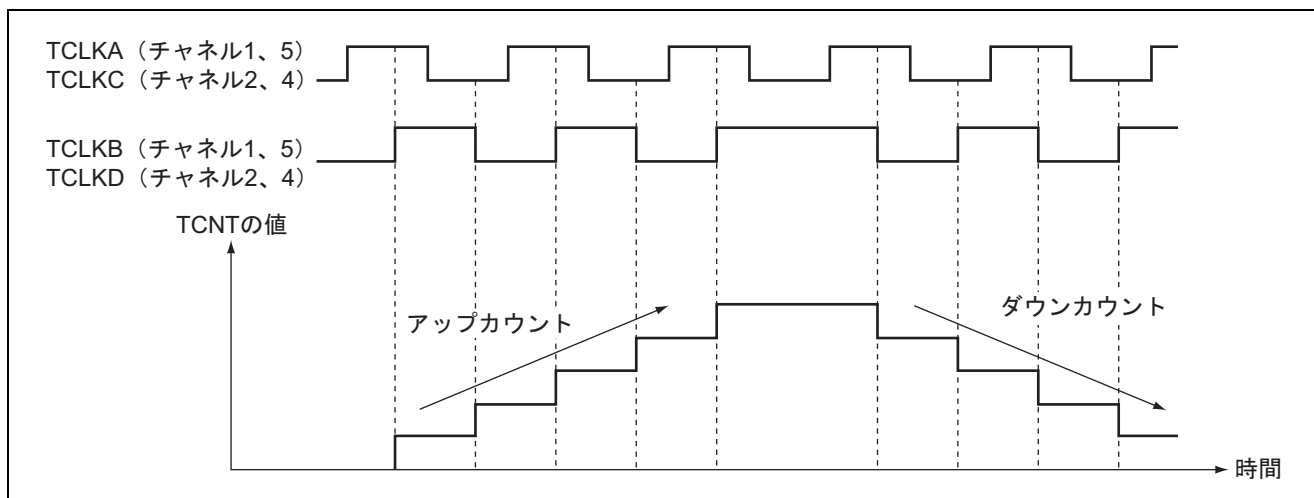


図 10.29 位相計数モード4の動作例

表 10.37 位相計数モード4のアップ/ダウンカウント条件

TCLKA (チャンネル 1、5) TCLKC (チャンネル 2、4)	TCLKB (チャンネル 1、5) TCLKD (チャンネル 2、4)	動作内容
High レベル	↑	アップカウント
Low レベル	↓	
↑	Low レベル	Don't care
↓	High レベル	
High レベル	↓	ダウンカウント
Low レベル	↑	
↑	High レベル	Don't care
↓	Low レベル	

【記号説明】

↑ : 立ち上がりエッジ

↓ : 立ち下がりエッジ

10.5 割り込み要因

TPU の割り込み要因には、TGR のインプットキャプチャ/コンペアマッチ、TCNT のオーバーフロー、アンダフローの3種類があります。各割り込み要因は、それぞれ専用のステータスフラグと、許可/禁止ビットを持っているため、割り込み要求信号の発生を独立に許可または禁止することができます。

割り込み要因が発生すると、TSR の対応するステータスフラグが1にセットされます。このとき TIER の対応する許可/禁止ビットが1にセットされていれば、割り込みを要求します。ステータスフラグを0にクリアすることで割り込み要求は解除されます。

表 10.38 に TPU の割り込み要因の一覧を示します。

表 10.38 TPU 割り込み一覧

チャンネル	名称	割り込み要因	割り込みフラグ	DTC の起動 (in_vnum)
0	TGI0A	TGRA_0 のインプットキャプチャ/コンペアマッチ	TGFA_0	可 (00000)
	TGI0B	TGRB_0 のインプットキャプチャ/コンペアマッチ	TGFB_0	可 (00001)
	TGI0C	TGRC_0 のインプットキャプチャ/コンペアマッチ	TGFC_0	可 (00010)
	TGI0D	TGRD_0 のインプットキャプチャ/コンペアマッチ	TGFD_0	可 (00011)
	TCI0V	TCNT_0 のオーバーフロー	TCFV_0	不可
1	TGI1A	TGRA_1 のインプットキャプチャ/コンペアマッチ	TGFA_1	可 (00100)
	TGI1B	TGRB_1 のインプットキャプチャ/コンペアマッチ	TGFB_1	可 (00101)
	TCI1V	TCNT_1 のオーバーフロー	TCFV_1	不可
	TCI1U	TCNT_1 のアンダフロー	TCFU_1	不可
2	TGI2A	TGRA_2 のインプットキャプチャ/コンペアマッチ	TGFA_2	可 (00110)
	TGI2B	TGRB_2 のインプットキャプチャ/コンペアマッチ	TGFB_2	可 (00111)
	TCI2V	TCNT_2 のオーバーフロー	TCFV_2	不可
	TCI2U	TCNT_2 のアンダフロー	TCFU_2	不可
3	TGI3A	TGRA_3 のインプットキャプチャ/コンペアマッチ	TGFA_3	可 (01000)
	TGI3B	TGRB_3 のインプットキャプチャ/コンペアマッチ	TGFB_3	可 (01001)
	TGI3C	TGRC_3 のインプットキャプチャ/コンペアマッチ	TGFC_3	可 (01010)
	TGI3D	TGRD_3 のインプットキャプチャ/コンペアマッチ	TGFD_3	可 (01011)
	TCI3V	TCNT_3 のオーバーフロー	TCFV_3	不可
4	TGI4A	TGRA_4 のインプットキャプチャ/コンペアマッチ	TGFA_4	可 (01100)
	TGI4B	TGRB_4 のインプットキャプチャ/コンペアマッチ	TGFB_4	可 (01101)
	TCI4V	TCNT_4 のオーバーフロー	TCFV_4	不可
	TCI4U	TCNT_4 のアンダフロー	TCFU_4	不可
5	TGI5A	TGRA_5 のインプットキャプチャ/コンペアマッチ	TGFA_5	可 (10000)
	TGI5B	TGRB_5 のインプットキャプチャ/コンペアマッチ	TGFB_5	可 (10001)
	TCI5V	TCNT_5 のオーバーフロー	TCFV_5	不可
	TCI5U	TCNT_5 のアンダフロー	TCFU_5	不可

- 【注】
1. リセット直後の初期状態について示しています。チャンネル間の優先順位は割り込みコントローラにより変更可能で
ず。
 2. 本 LSI では、チャンネル 3、チャンネル 4 およびチャンネル 5 でインプットキャプチャ機能はありません。

(1) インプットキャプチャ/コンペアマッチ割り込み

各チャンネルの TGR のインプットキャプチャ/コンペアマッチの発生により、TSR の TGF フラグが 1 にセットされたとき、TIER の TGIE ビットが 1 にセットされていれば、割り込みを要求します。TGF フラグを 0 にクリアすることで割り込み要求は解除されます。TPU には、チャンネル 0、3 に各 4 本、チャンネル 1、2、4、5 に各 2 本、計 16 本のインプットキャプチャ/コンペアマッチ割り込みがあります。

(2) オーバフロー割り込み

各チャンネルの TCNT のオーバフローの発生により、TSR の TCFV フラグが 1 にセットされたとき、TIER の TCIEV ビットが 1 にセットされていれば、割り込みを要求します。TCFV フラグを 0 にクリアすることで割り込み要求は解除されます。TPU には、各チャンネルに 1 本、計 6 本のオーバフロー割り込みがあります。

(3) アンダフロー割り込み

各チャンネルの TCNT のアンダフローの発生により、TSR の TCFU フラグが 1 にセットされたとき、TIER の TCIEU ビットが 1 にセットされていれば、割り込みを要求します。TCFU フラグを 0 にクリアすることで割り込み要求は解除されます。TPU には、チャンネル 1、2、4、5 に各 1 本、計 4 本のアンダフロー割り込みがあります。

10.6 DTC の起動

各チャンネルの TGR のインプットキャプチャ/コンペアマッチ割り込みによって、DTC を起動することができます。

TPU では、チャンネル 0、3 が各 4 本、チャンネル 1、2、4、5 が各 2 本、計 16 本のインプットキャプチャ/コンペアマッチ割り込みを DTC の起動要因とすることができます。

10.7 A/D 変換器の起動

各チャンネルの TGRA のインプットキャプチャ/コンペアマッチによって、A/D 変換器を起動することができます。

各チャンネルの TGRA のインプットキャプチャ/コンペアマッチの発生により、TSR の TGFA フラグが 1 にセットされたとき、TIER の TTGE ビットが 1 にセットされていれば、A/D 変換器に対して A/D 変換の開始を要求します。このとき A/D 変換器側で、TPU の変換開始トリガが選択されていれば、A/D 変換が開始されます。

TPU では、各チャンネル 1 本、計 6 本の TGRA のインプットキャプチャ/コンペアマッチ割り込みを A/D 変換器の変換開始要因とすることができます。

10.8 動作タイミング

10.8.1 入出力タイミング

(1) TCNT のカウントタイミング

内部クロック動作の場合の TCNT のカウントタイミングを図 10.31 に示します。また、外部クロック動作の場合の TCNT のカウントタイミングを図 10.32 に示します。

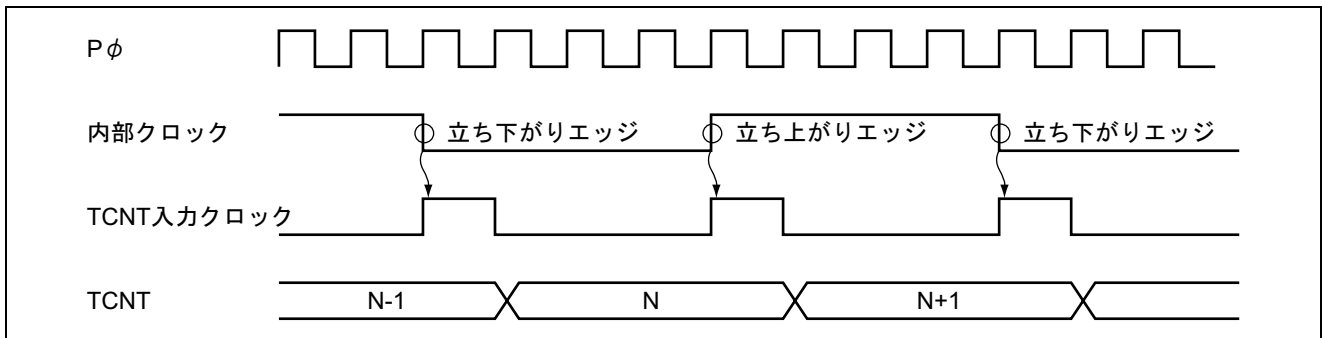


図 10.31 内部クロック動作時のカウントタイミング

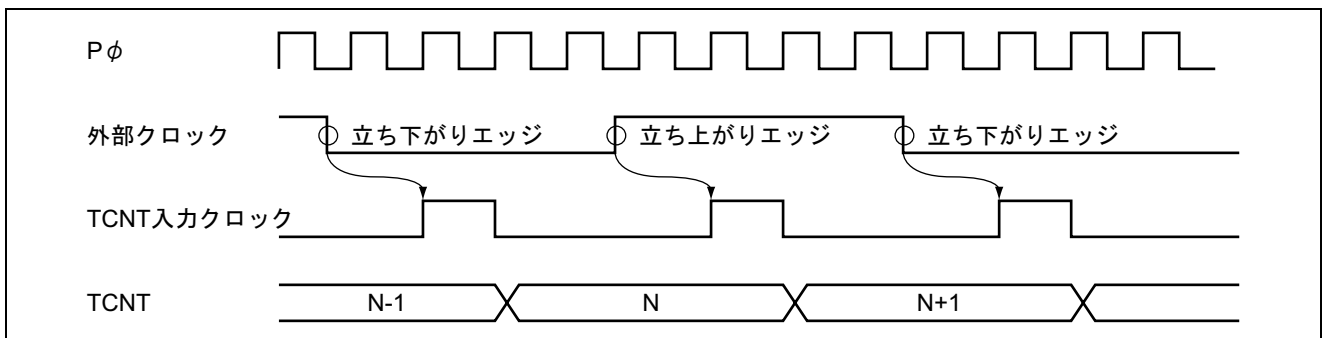


図 10.32 外部クロック動作時のカウントタイミング

(2) アウトプットコンペア出力タイミング

コンペアマッチ信号は、TCNT と TGR が一致した最後のステート (TCNT 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。コンペアマッチ信号が発生したとき、TIOR で設定した出力値がアウトプットコンペア出力端子 (TIOC 端子) に出力されます。TCNT と TGR が一致した後、TCNT 入力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。

アウトプットコンペア出力タイミングを図 10.33 に示します。

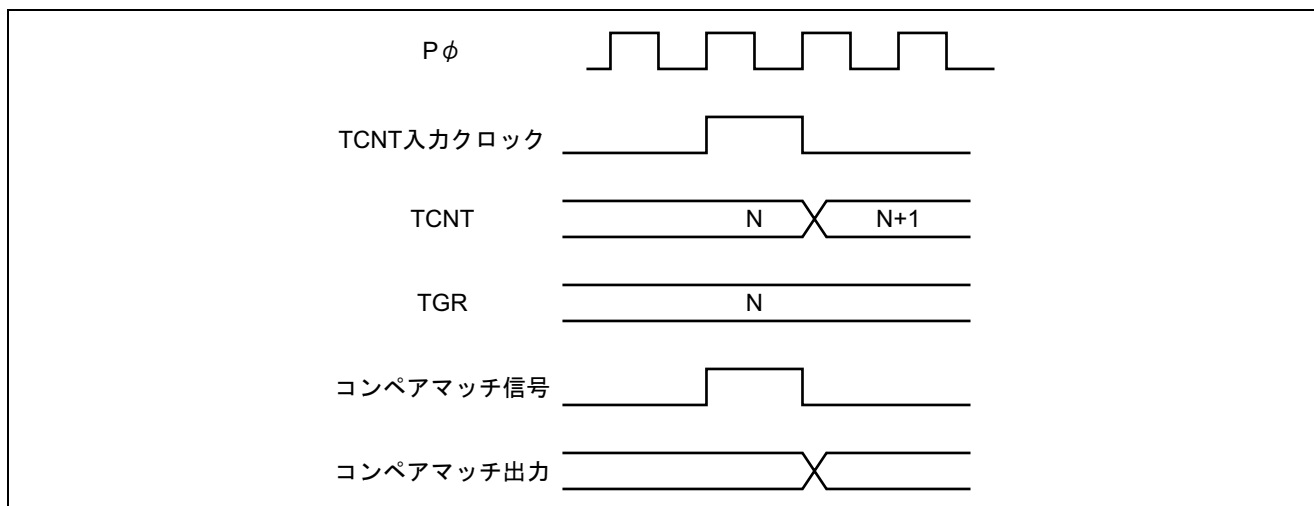


図 10.33 アウトプットコンペア出力タイミング

(3) インプットキャプチャ信号タイミング

インプットキャプチャのタイミングを図 10.34 に示します。

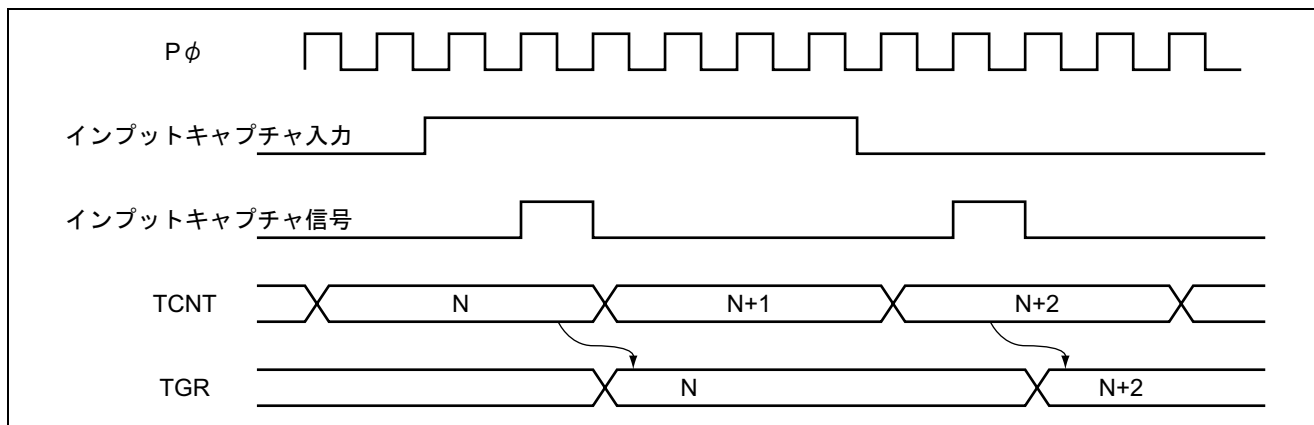


図 10.34 インプットキャプチャ入力信号タイミング

(4) コンペアマッチ/インプットキャプチャによるカウンタクリアタイミング

コンペアマッチの発生によるカウンタクリアを指定した場合のタイミングを図 10.35 に示します。

インプットキャプチャの発生によるカウンタクリアを指定した場合のタイミングを図 10.36 に示します。

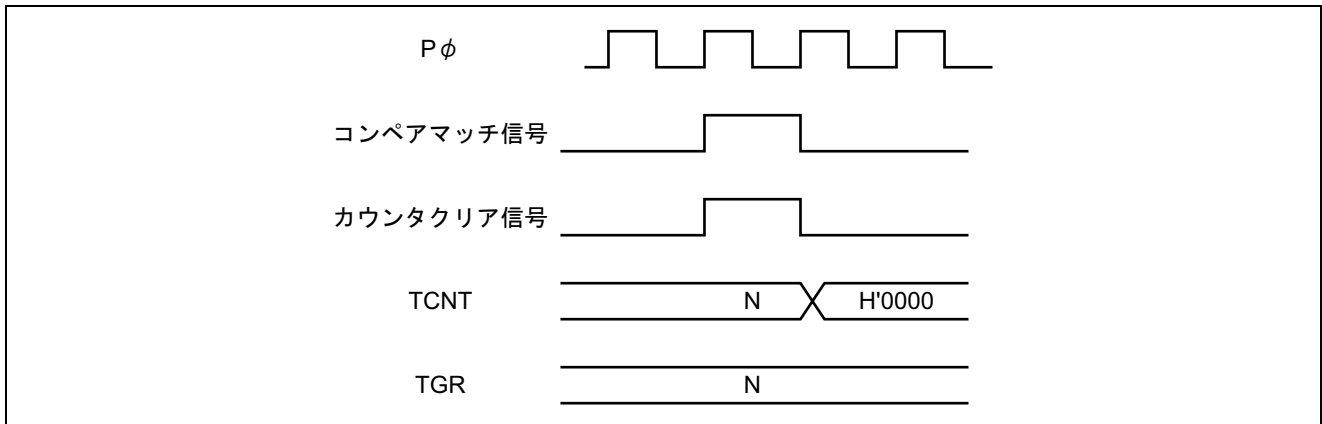


図 10.35 カウンタクリアタイミング (コンペアマッチ)

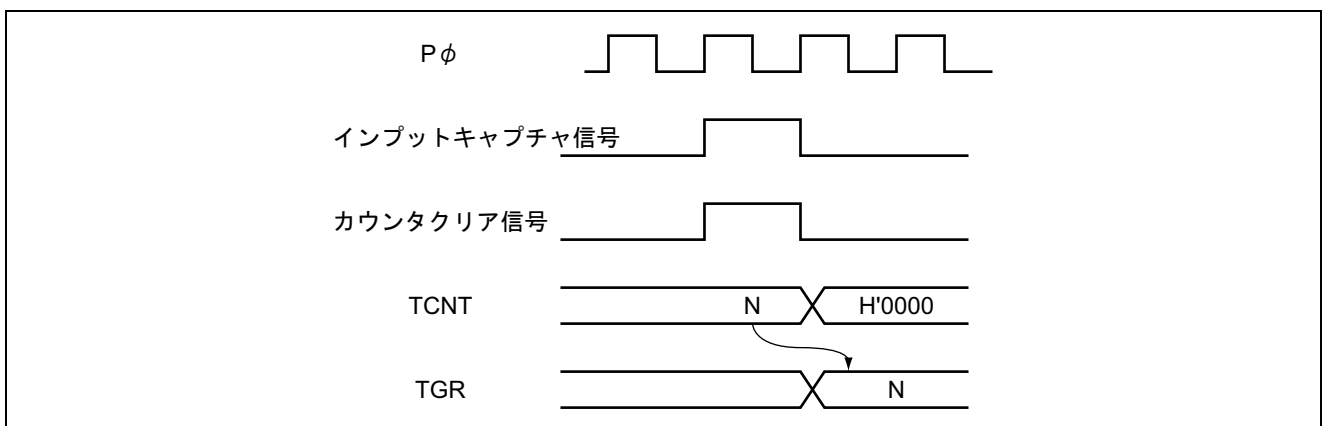


図 10.36 カウンタクリアタイミング (インプットキャプチャ)

(5) バッファ動作タイミング

バッファ動作の場合のタイミングを図 10.37、図 10.38 に示します。

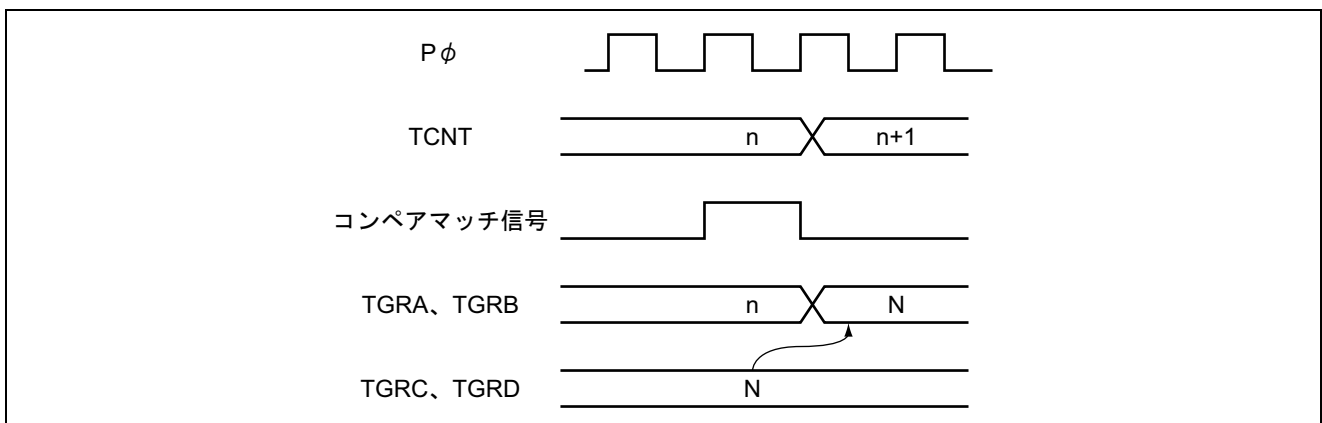


図 10.37 バッファ動作タイミング (コンペアマッチ)

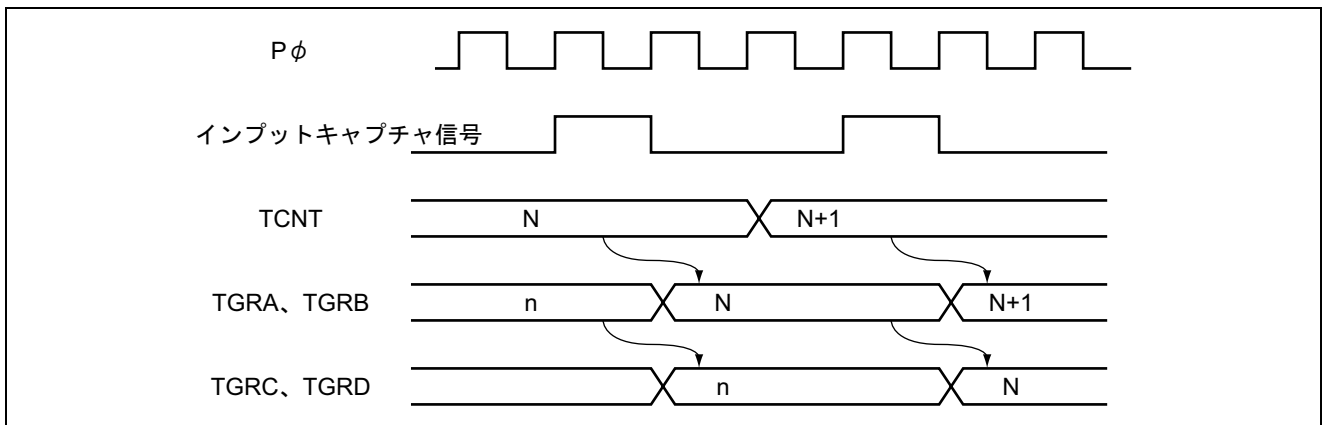


図 10.38 バッファ動作タイミング (インプットキャプチャ)

10.8.2 割り込み信号タイミング

(1) コンペアマッチ時の TGF フラグのセットタイミング

コンペアマッチの発生による TSR の TGF フラグのセットタイミングと、TGI 割り込み要求信号のタイミングを図 10.39 に示します。

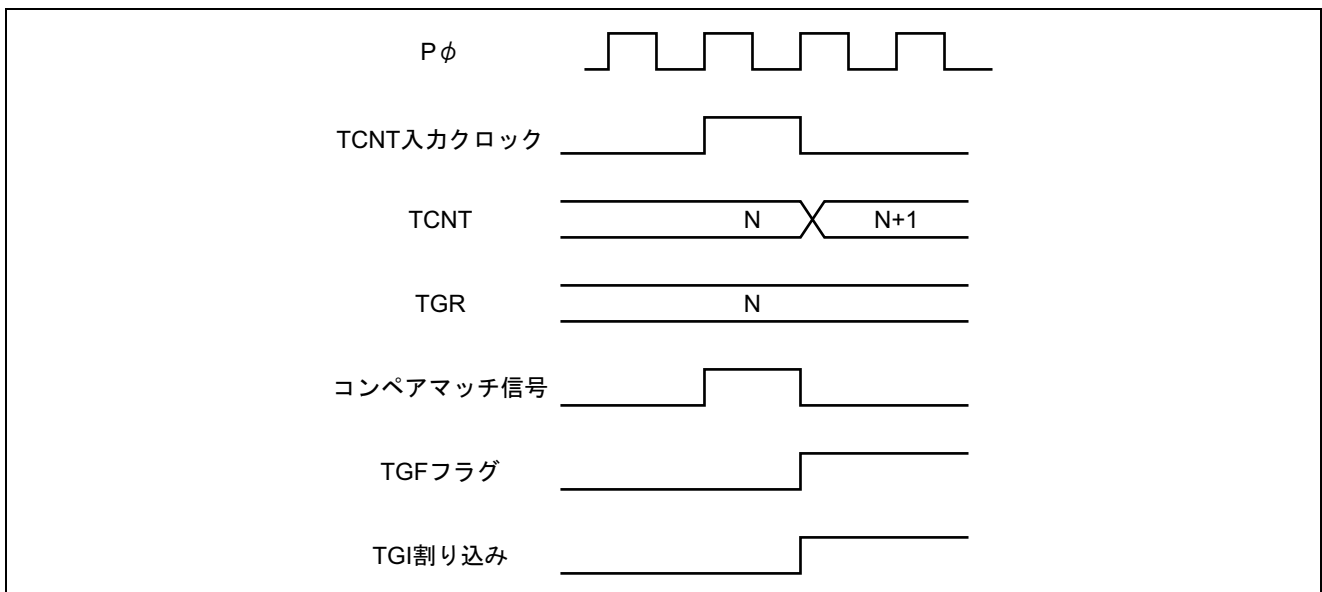


図 10.39 TGI 割り込みタイミング (コンペアマッチ)

(2) インพุットキャプチャ時の TGF フラグのセットタイミング

インพุットキャプチャの発生による TSR の TGF フラグのセットタイミングと、TGI 割り込み要求信号のタイミングを図 10.40 に示します。

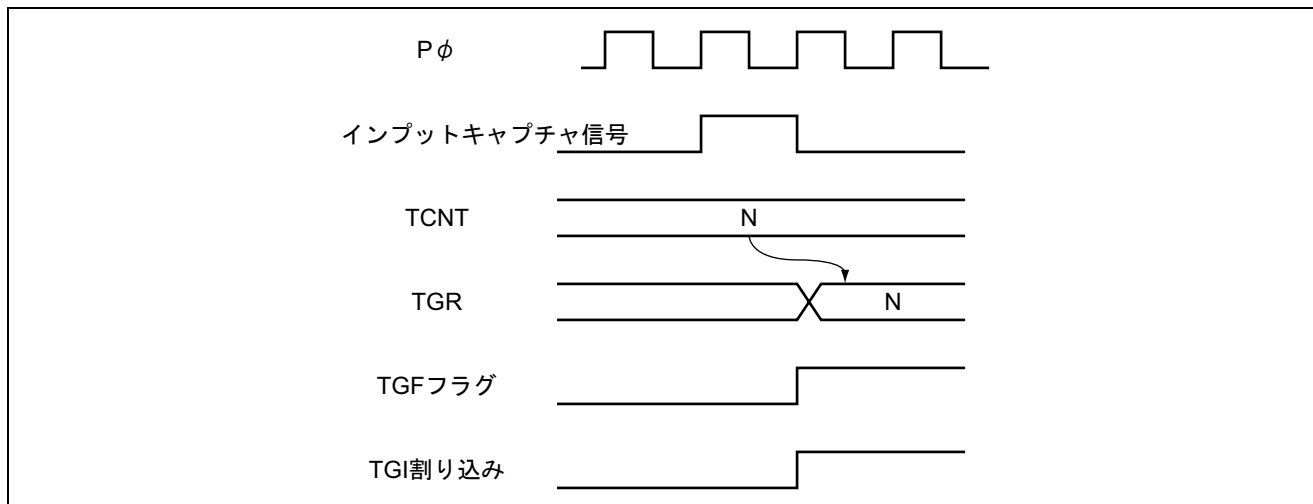


図 10.40 TGI 割り込みタイミング (インพุットキャプチャ)

(3) TCFV フラグ/TCFU フラグのセットタイミング

オーバフローの発生による TSR の TCFV フラグのセットタイミングと、TCIV 割り込み要求信号のタイミングを図 10.41 に示します。

アンダフローの発生による TSR の TCFU フラグのセットタイミングと、TCIU 割り込み要求信号のタイミングを図 10.42 に示します。

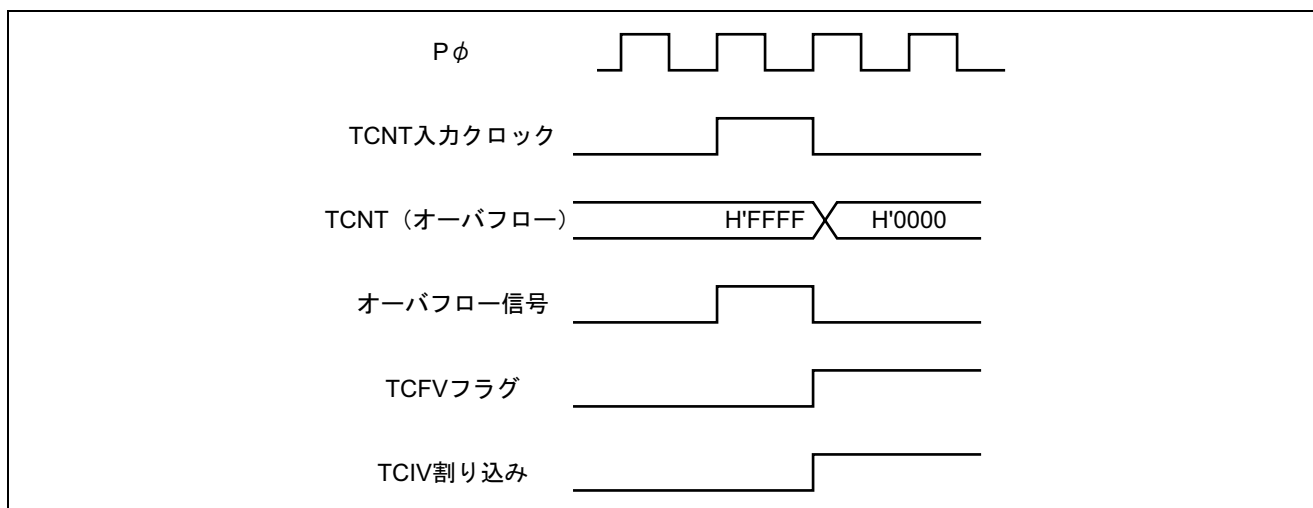


図 10.41 TCIV 割り込みのセットタイミング

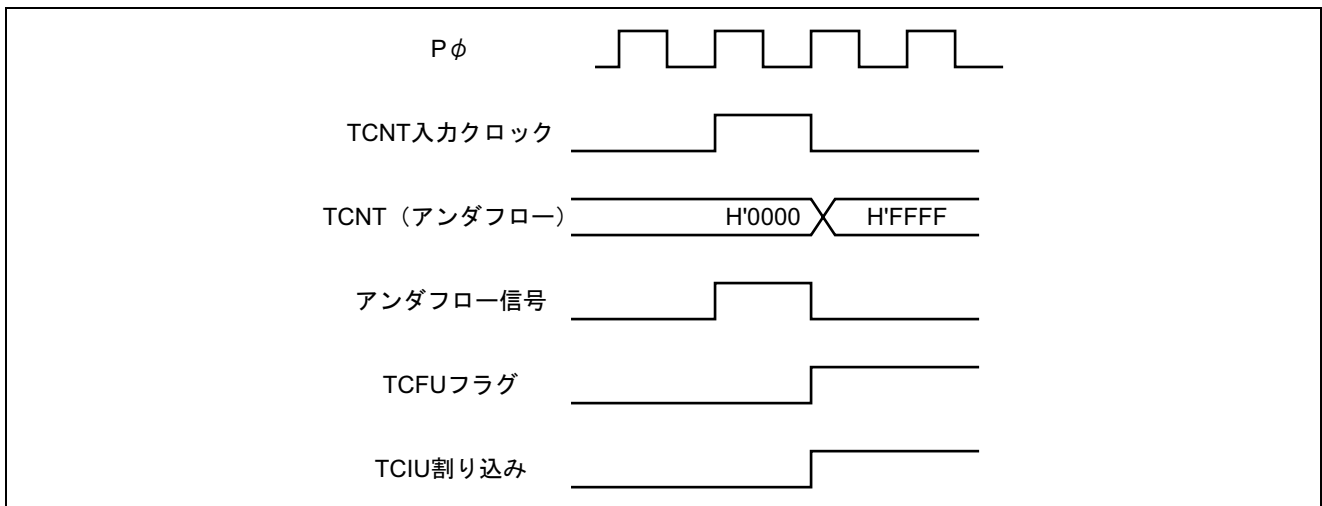


図 10.42 TCIU 割り込みのセットタイミング

(4) ステータスフラグのクリアタイミング

ステータスフラグはCPUが1の状態をリードした後、0をライトするとクリアされます。DTCを起動する場合は、自動的にクリアすることもできます。CPUによるステータスフラグのクリアタイミングを図10.43に、DTCによるステータスフラグのクリアのタイミングを図10.44に示します。

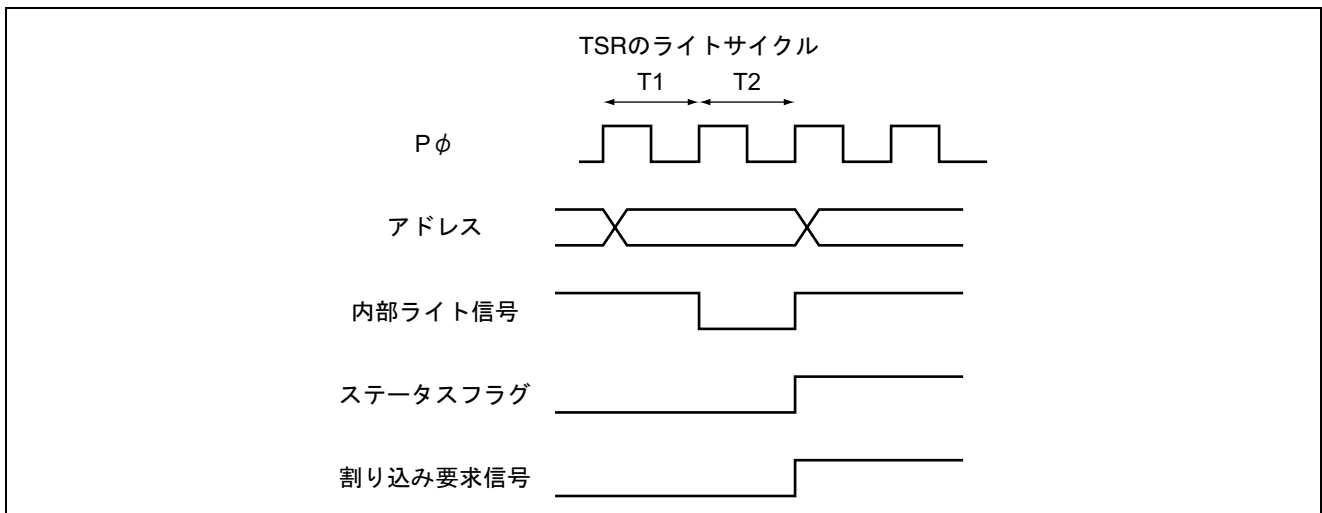


図 10.43 CPUによるステータスフラグのクリアタイミング

ステータスフラグ、割り込み要求信号のクリアは、**図 10.44** に示すように DTC の転送中に行われます。

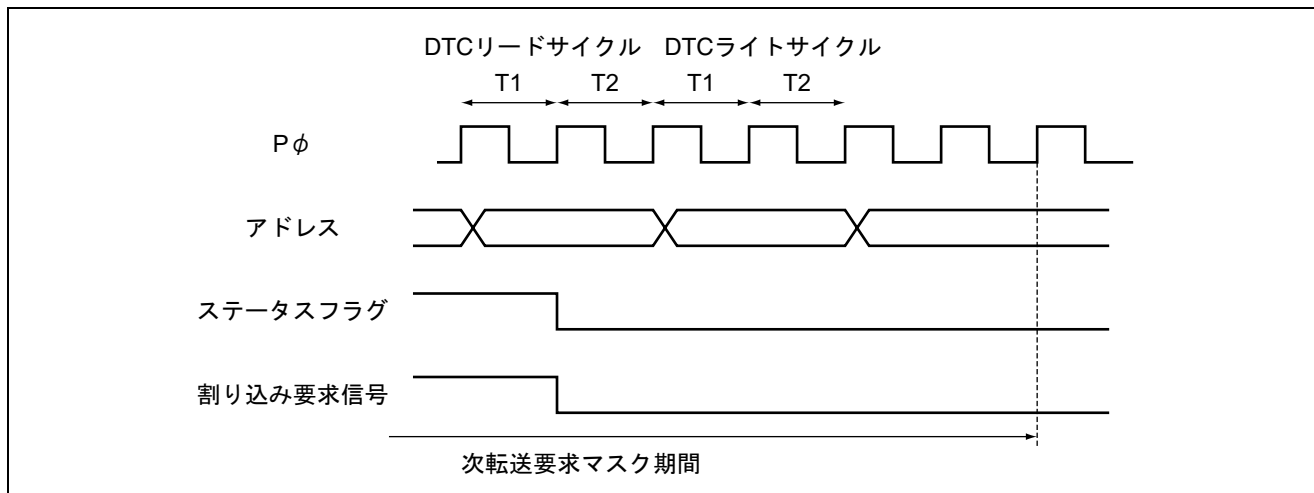


図 10.44 DTC の起動によるステータスフラグのクリアタイミング例

10.9 使用上の注意事項

10.9.1 モジュールストップモードの設定

モジュールストップコントロールレジスタにより、TPU の動作禁止/許可を設定することが可能です。初期値では、TPU の動作は停止します。モジュールストップモードを解除することにより、レジスタのアクセスが可能になります。詳細は「**第 23 章 低消費電力**」を参照してください。

10.9.2 入力クロックの制限事項

入力クロックのパルス幅は、単エッジの場合は 1.5 ステートクロック以上、両エッジの場合は 2.5 ステート以上が必要です。これ以下のパルス幅では正しく動作しませんのでご注意ください。

位相計数モードの場合は、2 本の入力クロックの位相差およびオーバーラップはそれぞれ 1.5 ステート以上、パルス幅は 2.5 ステート以上必要です。位相計数モードの入力クロックの条件を**図 10.45** に示します。

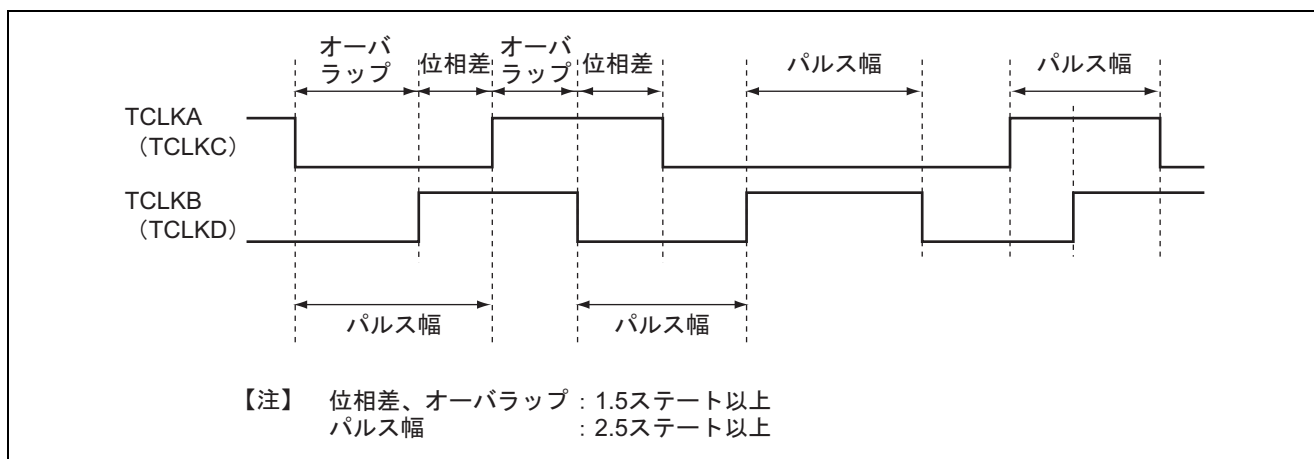


図 10.45 位相計数モード時の位相差、オーバーラップ、およびパルス幅

10.9.3 周期設定上の注意事項

コンペアマッチによるカウンタクリアを設定した場合、TCNTはTGRの値と一致した最後のステート(TCNTが一致したカウント値を更新するタイミング)でクリアされます。このため、実際のカウンタの周波数は次の式ようになります。

$$f = \frac{P\phi}{(N+1)}$$

f : カウンタ周波数

Pφ : 動作周波数

N : TGR の設定値

10.9.4 TCNTのライトとクリアの競合

TCNTのライトサイクル中のT2ステートでカウンタクリア信号が発生すると、TCNTへのライトは行われずにTCNTのクリアが優先されます。このタイミングを図10.46に示します。

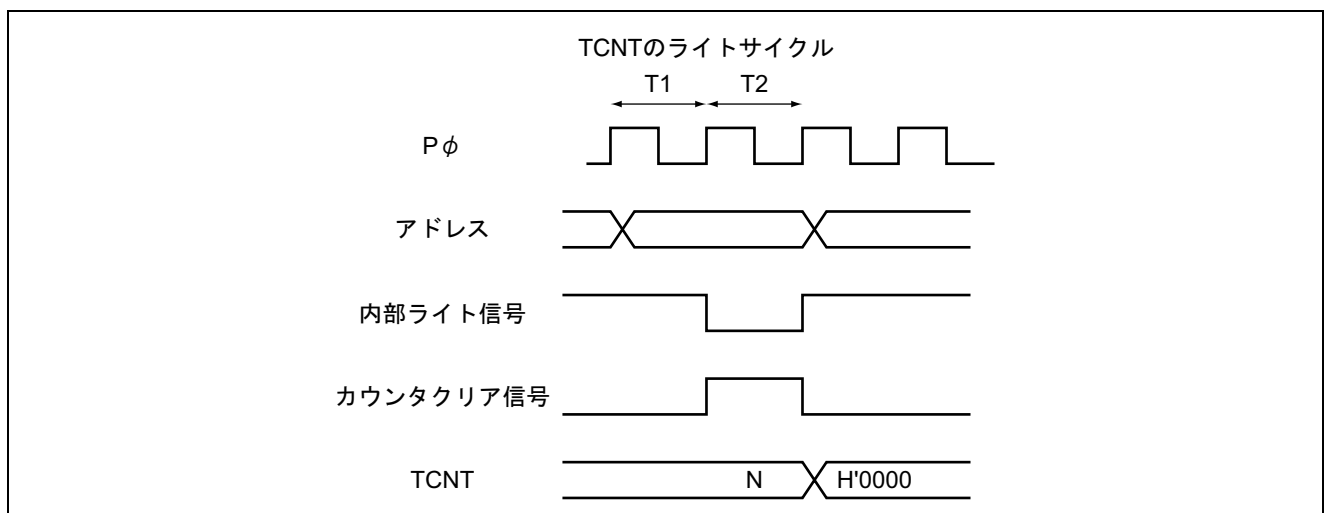


図 10.46 TCNTのライトとクリアの競合

10.9.5 TCNT のライトとカウントアップの競合

TCNT のライトサイクル中の T2 ステートでカウントアップが発生してもカウントアップされず、TCNT へのライトが優先されます。このタイミングを図 10.47 に示します。

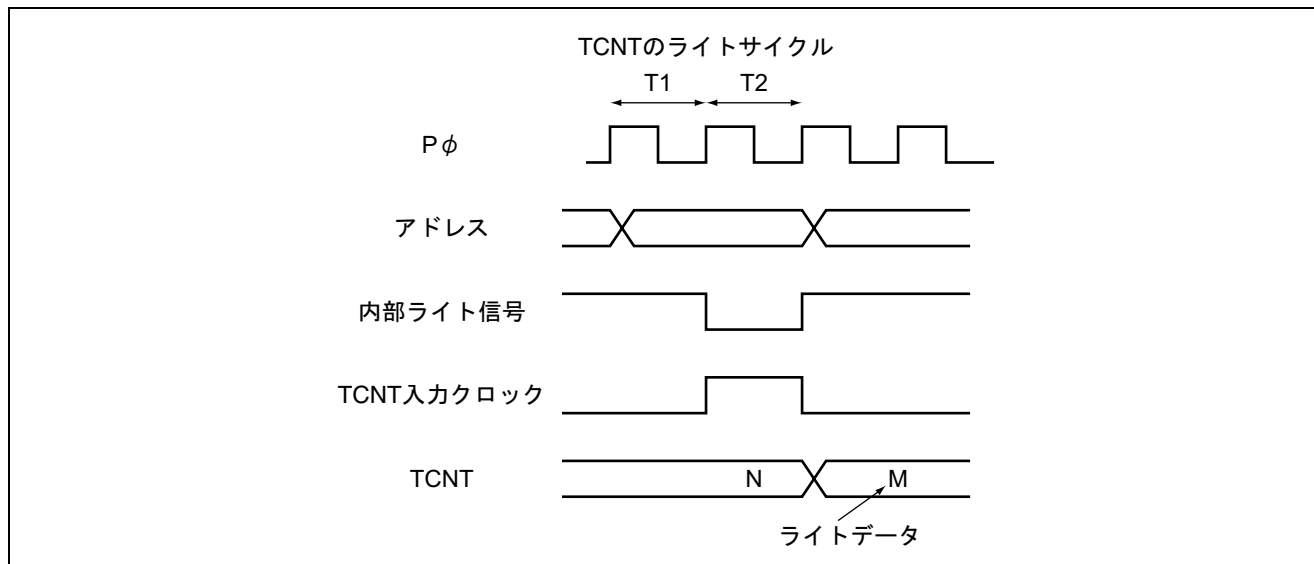


図 10.47 TCNT のライトとカウントアップの競合

10.9.6 TGR のライトとコンペアマッチの競合

TGR のライトサイクル中の T2 ステートでコンペアマッチが発生しても TGR のライトが優先され、コンペアマッチ信号は禁止されます。前回と同じ値をライトした場合でもコンペアマッチは発生しません。

このタイミングを図 10.48 に示します。

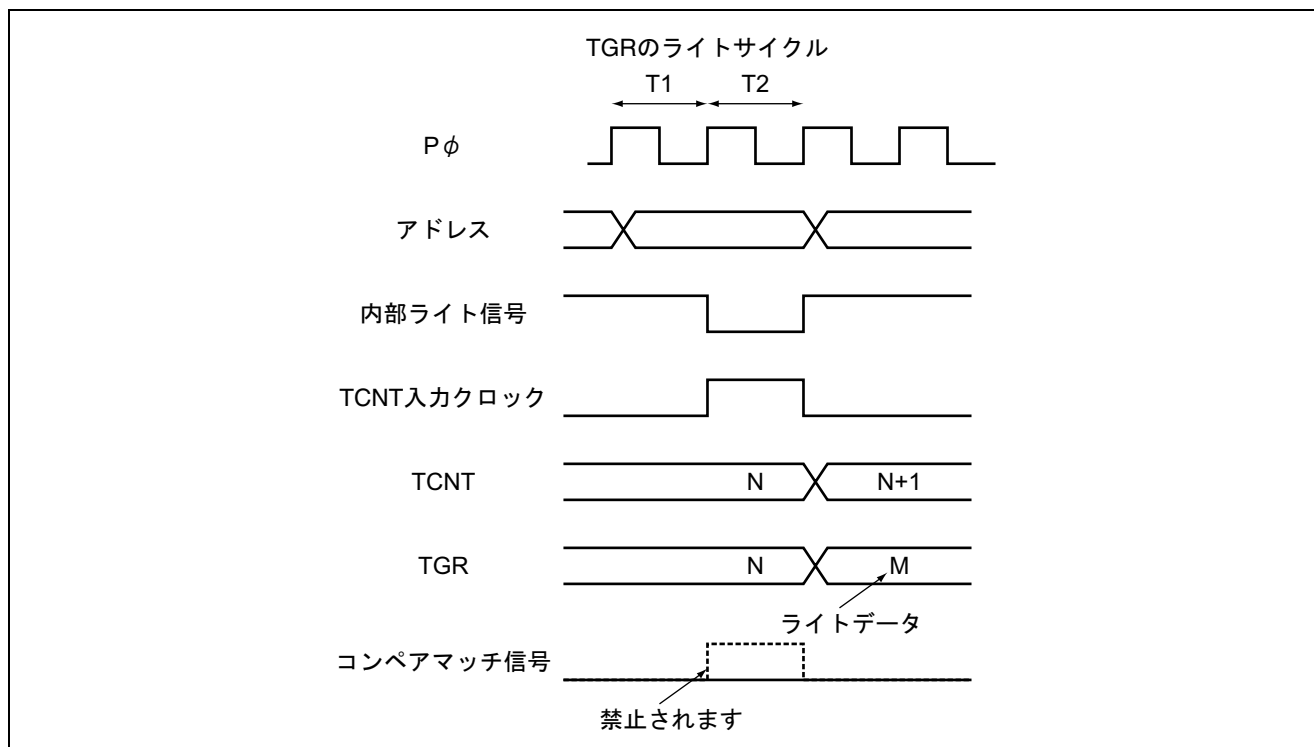


図 10.48 TGR のライトとコンペアマッチの競合

10.9.7 バッファレジスタのライトとコンペアマッチの競合

TGR のライトサイクル中の T2 ステートでコンペアマッチが発生すると、バッファ動作によって TGR に転送されるデータはライト前のデータとなります。

このタイミングを図 10.49 に示します。

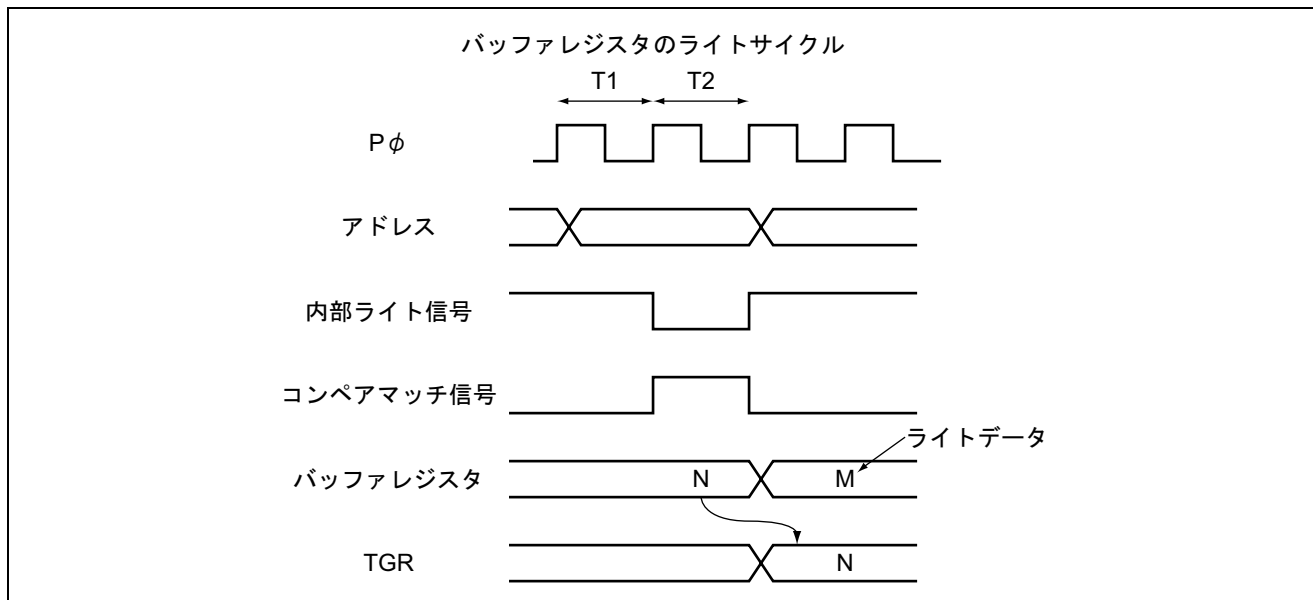


図 10.49 バッファレジスタのライトとコンペアマッチの競合

10.9.8 TGR のリードとインプットキャプチャの競合

TGR のリードサイクル中の T1 ステートでインプットキャプチャ信号が発生すると、リードされるデータはインプットキャプチャ転送前のデータとなります。

このタイミングを図 10.50 に示します。

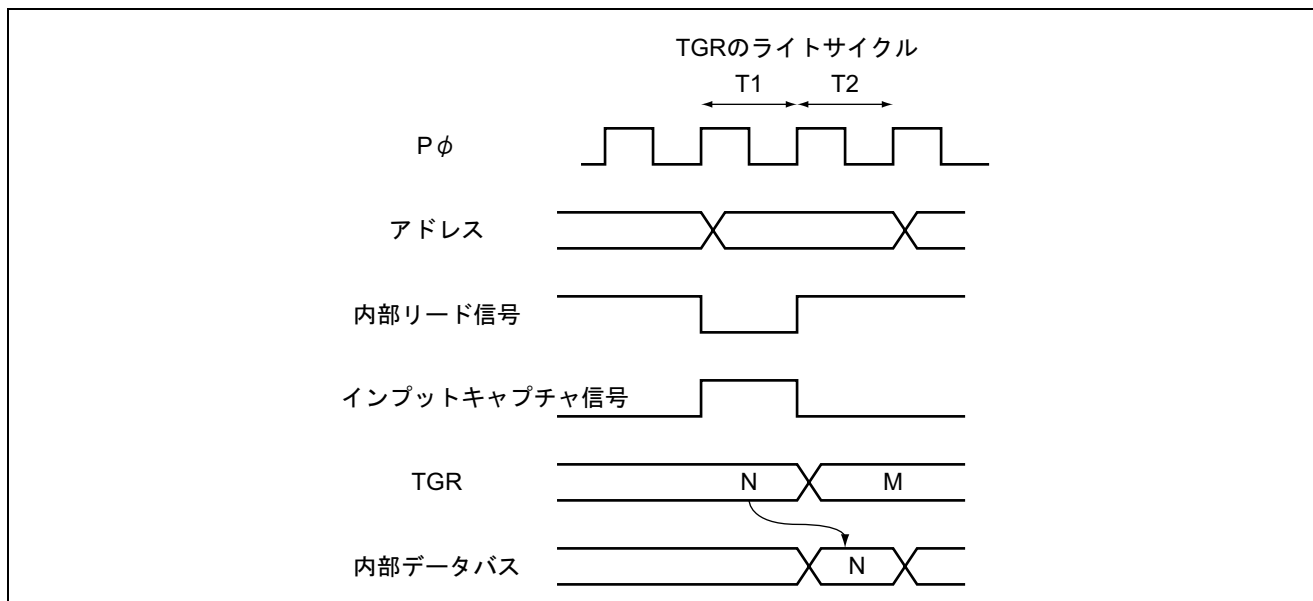


図 10.50 TGR のリードとインプットキャプチャの競合

10.9.9 TGR のライトとインプットキャプチャの競合

TGR のライトサイクル中の T2 ステートでインプットキャプチャ信号が発生すると、TGR へのライトは行われず、インプットキャプチャが優先されます。このタイミングを図 10.51 に示します。

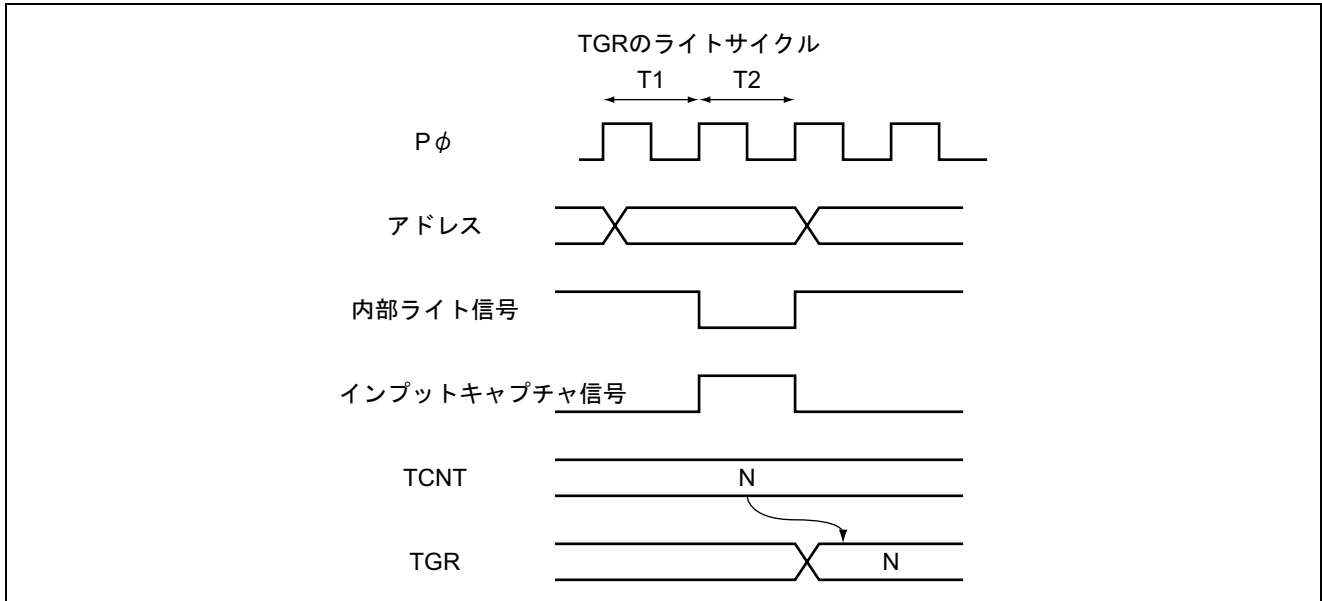


図 10.51 TGR のライトとインプットキャプチャの競合

10.9.10 バッファレジスタのライトとインプットキャプチャの競合

バッファレジスタのライトサイクル中の T2 ステートでインプットキャプチャ信号が発生すると、バッファレジスタへのライトは行われず、バッファ動作が優先されます。このタイミングを図 10.52 に示します。

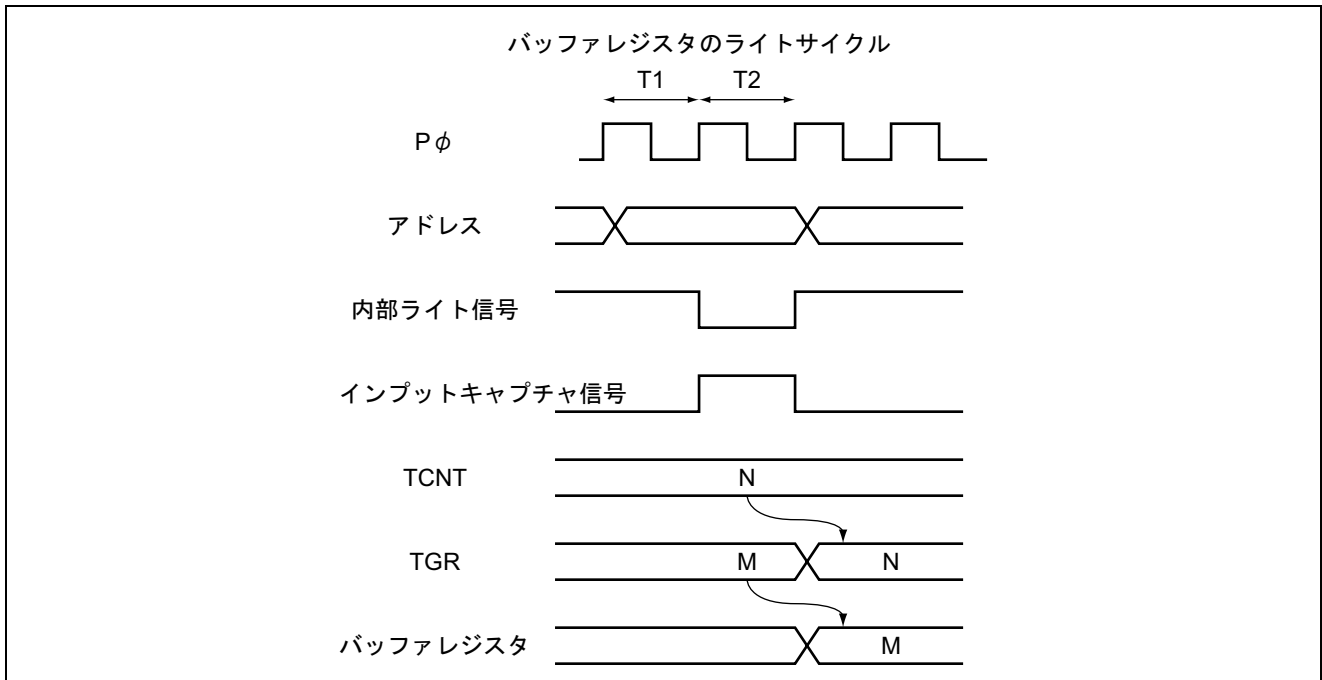


図 10.52 バッファレジスタのライトとインプットキャプチャの競合

10.9.11 オーバフロー／アンダフローとカウンタクリアの競合

オーバフロー／アンダフローとカウンタクリアが同時に発生すると、TSR の TCFV/TCFU フラグはセットされ、TCNT のクリアも行われます。

TGR のコンペアマッチをクリア要因とし、TGR に H'FFFF を設定した場合の動作タイミングを図 10.53 に示します。

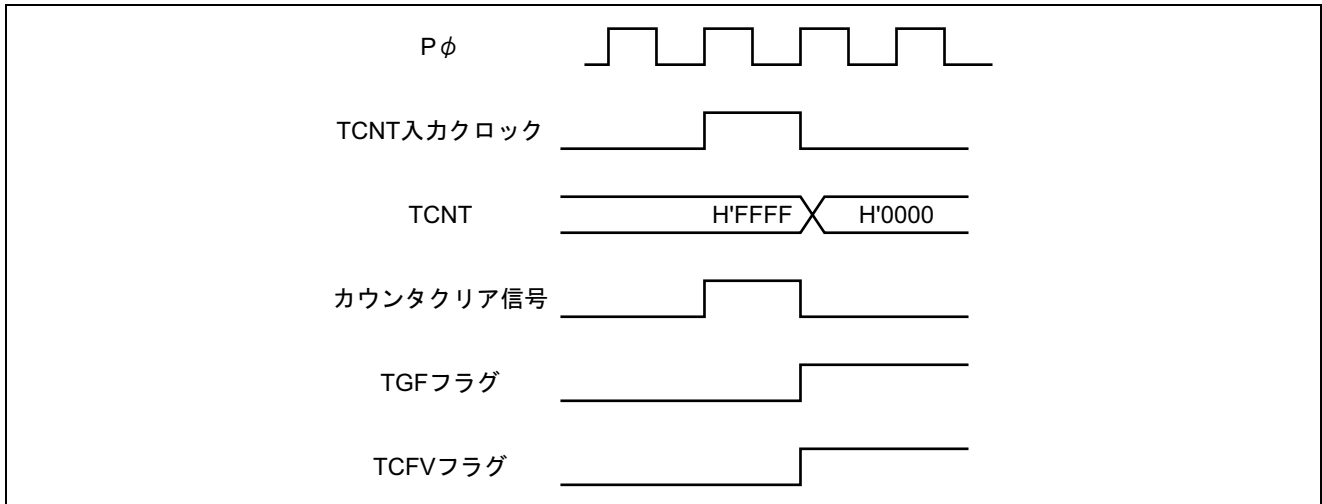


図 10.53 オーバフローとカウンタクリアの競合

10.9.12 TCNT のライトとオーバフロー／アンダフローの競合

TCNT のライトサイクル中の T2 ステートでカウントアップ／カウントダウンが発生し、オーバフロー／アンダフローが発生しても TCNT へのライトが優先され、TSR の TCFV/TCFU フラグはセットされません。

TCNT のライトとオーバフロー競合時の動作タイミングを図 10.54 に示します。

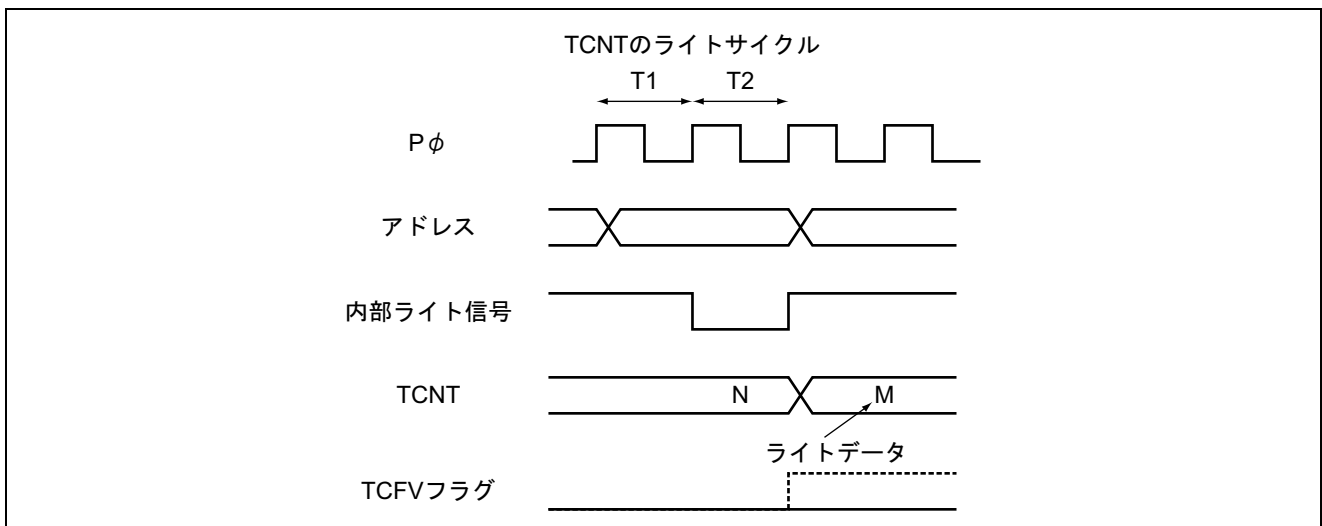


図 10.54 TCNT のライトとオーバフローの競合

10.9.13 モジュールストップ時の割り込み

割り込みが要求された状態でモジュールストップモードにすると、CPU の割り込み要因、DTC の起動要因のクリアができません。事前に割り込みをディスエーブルしてからモジュールストップモードとしてください。

11. プログラマブルパルスジェネレータ (PPG)

プログラマブルパルスジェネレータ (PPG) は 16 ビットタイマパルスユニット (TPU) をタイムベースとしてパルスを出力します。PPG は 4 ビット単位のパルス出力グループ 3、2 から構成されており、これらは同時に動作させることも、独立に動作させることもできます。PPG のブロック図を図 11.1 に示します。

11.1 特長

- 出力データ8ビット
- 2系統の出力可能
- 出力トリガ信号を選択可能
- ノンオーバーラップ動作可能
- データトランスファコントローラ (DTC) との連携動作可能
- 反転出力の指定可能
- モジュールストップモードの設定可能

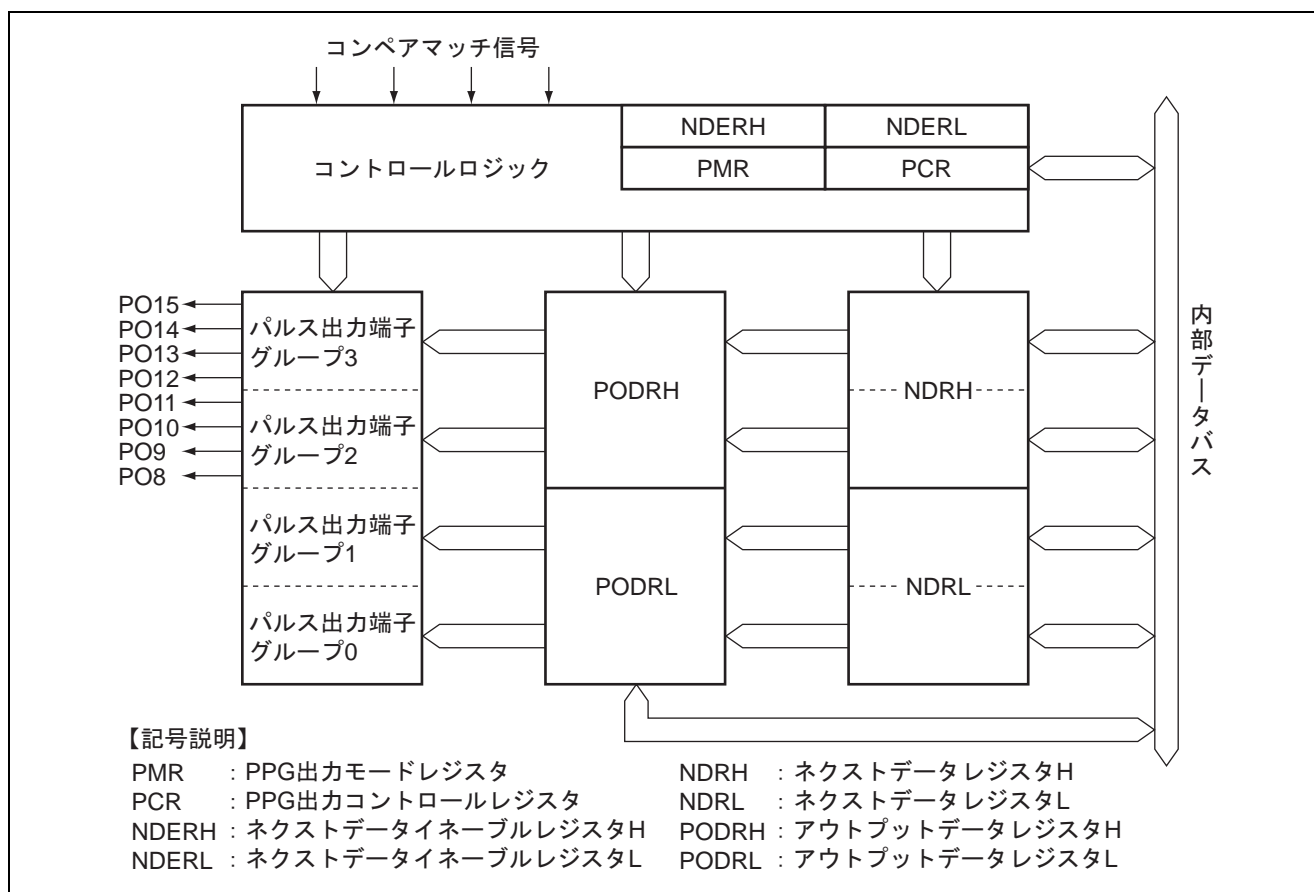


図 11.1 PPG のブロック図

11.2 入出力端子

PPG には表 11.1 の入出力端子があります。

表 11.1 PPG の入出力端子

端子名	入出力	機能
PO15	出力	パルス出力グループ 3 のパルス出力
PO14	出力	
PO13	出力	
PO12	出力	
PO11	出力	パルス出力グループ 2 のパルス出力
PO10	出力	
PO9	出力	
PO8	出力	

11.3 レジスタの説明

PPG には以下のレジスタがあります。

- ネクストデータイネーブルレジスタH (NDERH)
- ネクストデータイネーブルレジスタL (NDERL)
- アウトプットデータレジスタH (PODRH)
- アウトプットデータレジスタL (PODRL)
- ネクストデータレジスタH (NDRH)
- ネクストデータレジスタL (NDRL)
- PPG出力コントロールレジスタ (PCR)
- PPG出力モードレジスタ (PMR)

11.3.1 ネクストデータイネーブルレジスタ H、L (NDERH、NDERL)

NDERH、NDERL は、PPG によるパルス出力端子をビット単位で選択します。

・ NDERH

ビット	7	6	5	4	3	2	1	0
ビット名	NDER15	NDER14	NDER13	NDER12	NDER11	NDER10	NDER9	NDER8
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

・ NDERL

ビット	7	6	5	4	3	2	1	0
ビット名	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0
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

• NDERH

ビット	ビット名	初期値	R/W	説明
7	NDER15	0	R/W	ネクストデータイネーブル 15~8 1 にセットすると選択された出力トリガによって NDRH の対応するビットから PODRH ヘデータが転送されます。クリアされているビットは NDRH から PODRH へのデータ転送は行われません。
6	NDER14	0	R/W	
5	NDER13	0	R/W	
4	NDER12	0	R/W	
3	NDER11	0	R/W	
2	NDER10	0	R/W	
1	NDER9	0	R/W	
0	NDER8	0	R/W	

• NDERL

ビット	ビット名	初期値	R/W	説明
7	NDER7	0	R/W	ネクストデータイネーブル 7~0 1 にセットすると選択された出力トリガによって NDRL の対応するビットから PODRL ヘデータが転送されます。クリアされているビットは NDRL から PODRL へのデータ転送は行われません。
6	NDER6	0	R/W	
5	NDER5	0	R/W	
4	NDER4	0	R/W	
3	NDER3	0	R/W	
2	NDER2	0	R/W	
1	NDER1	0	R/W	
0	NDER0	0	R/W	

11.3.2 アウトプットデータレジスタ H、L (PODRH、PODRL)

PODRH、PODRL は、パルス出力値が格納されます。NDER によりパルス出力に設定されたビットはリード専用となり、ライトできません。

・ PODRH

ビット	7	6	5	4	3	2	1	0
ビット名	POD15	POD14	POD13	POD12	POD11	POD10	POD9	POD8
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

・ PODRL

ビット	7	6	5	4	3	2	1	0
ビット名	POD7	POD6	POD5	POD4	POD3	POD2	POD1	POD0
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

• PODRH

ビット	ビット名	初期値	R/W	説明
7	POD15	0	R/W	アウトプットデータレジスタ 15~8 NDERH によりパルス出力に設定されたビットは PPG 動作中、出力トリガによって NDRH の値がこのレジスタに転送されます。NDERH が 1 にセットされている期間 CPU からはライトできません。NDERH がクリアされている状態ではパルスの初期出力値を設定することができます。
6	POD14	0	R/W	
5	POD13	0	R/W	
4	POD12	0	R/W	
3	POD11	0	R/W	
2	POD10	0	R/W	
1	POD9	0	R/W	
0	POD8	0	R/W	

• PODRL

ビット	ビット名	初期値	R/W	説明
7	POD7	0	R/W	アウトプットデータレジスタ 7~0 NDERL によりパルス出力に設定されたビットは PPG 動作中、出力トリガによって NDRL の値がこのレジスタに転送されます。NDERL が 1 にセットされている期間 CPU からはライトできません。NDERL がクリアされている状態ではパルスの初期出力値を設定することができます。
6	POD6	0	R/W	
5	POD5	0	R/W	
4	POD4	0	R/W	
3	POD3	0	R/W	
2	POD2	0	R/W	
1	POD1	0	R/W	
0	POD0	0	R/W	

11.3.3 ネクストデータレジスタ H、L (NDRH、NDRL)

NDRH、NDRL は、パルス出力の次のデータを格納します。NDR のアドレスは、パルス出力グループの出力トリガを同一に設定した場合と、異なる出力トリガを選択した場合とで異なります。

・ NDRH

ビット	7	6	5	4	3	2	1	0
ビット名	NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9	NDR8
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

・ NDRL

ビット	7	6	5	4	3	2	1	0
ビット名	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1	NDR0
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

• NDRH

パルス出力グループ 2、3 の出力トリガを同一にすると、以下のように 8 ビットすべて同一アドレスにマッピングされ、8 ビット同時にアクセスできます。

ビット	ビット名	初期値	R/W	説明
7	NDR15	0	R/W	ネクストデータレジスタ 15~8 PCR で指定した出力トリガにより、このレジスタの内容が PODRH の対応するビットに転送されます。
6	NDR14	0	R/W	
5	NDR13	0	R/W	
4	NDR12	0	R/W	
3	NDR11	0	R/W	
2	NDR10	0	R/W	
1	NDR9	0	R/W	
0	NDR8	0	R/W	

パルス出力グループ 2 とパルス出力グループ 3 で異なる出力トリガを選択すると、以下のように上位 4 ビットと下位 4 ビットは異なるアドレスにマッピングされます。

ビット	ビット名	初期値	R/W	説明
7	NDR15	0	R/W	ネクストデータレジスタ 15~12 PCR で指定した出力トリガにより、このレジスタの内容が PODRH の対応するビットに転送されます。
6	NDR14	0	R/W	
5	NDR13	0	R/W	
4	NDR12	0	R/W	
3~0	—	すべて 1	R	リザーブビット 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	ビット名	初期値	R/W	説明
7~4	—	すべて1	—	リザーブビット 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。
3	NDR11	0	R/W	ネクストデータレジスタ 11~8
2	NDR10	0	R/W	PCR で指定した出カトリガにより、このレジスタの内容が PODRH の対応するビットに転送されます。
1	NDR9	0	R/W	
0	NDR8	0	R/W	

• NDRL

パルス出力グループ 0、1 の出カトリガを同一にすると、以下のように 8 ビットすべて同一アドレスにマッピングされ、8 ビット同時にアクセスできます。

ビット	ビット名	初期値	R/W	説明
7	NDR7	0	R/W	ネクストデータレジスタ 7~0
6	NDR6	0	R/W	PCR で指定した出カトリガにより、このレジスタの内容が PODRL の対応するビットに転送されます。
5	NDR5	0	R/W	
4	NDR4	0	R/W	
3	NDR3	0	R/W	
2	NDR2	0	R/W	
1	NDR1	0	R/W	
0	NDR0	0	R/W	

パルス出力グループ 0 とパルス出力グループ 1 で異なる出カトリガを選択すると、以下のように上位 4 ビットと下位 4 ビットは異なるアドレスにマッピングされます。

ビット	ビット名	初期値	R/W	説明
7	NDR7	0	R/W	ネクストデータレジスタ 7~4
6	NDR6	0	R/W	PCR で指定した出カトリガにより、このレジスタの内容が PODRL の対応するビットに転送されます。
5	NDR5	0	R/W	
4	NDR4	0	R/W	
3~0	—	すべて1	—	リザーブビット 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	ビット名	初期値	R/W	説明
7~4	—	すべて1	—	リザーブビット 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。
3	NDR3	0	R/W	ネクストデータレジスタ 3~0
2	NDR2	0	R/W	PCR で指定した出カトリガにより、このレジスタの内容が PODRL の対応するビットに転送されます。
1	NDR1	0	R/W	
0	NDR0	0	R/W	

11.3.4 PPG 出力コントロールレジスタ (PCR)

PCR は、パルス出力トリガ信号をグループ単位で選択します。出力トリガの選択については「11.3.5 PPG 出力モードレジスタ (PMR)」を併せて参照してください。

ビット	7	6	5	4	3	2	1	0
ビット名	G3CMS1	G3CMS0	G2CMS1	G2CMS0	—	—	—	—
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

ビット	ビット名	初期値	R/W	説明
7	G3CMS1	1	R/W	グループ 3 コンペアマッチセレクト 1、0 パルス出力グループ 3 の出力トリガを選択します。 00: TPU チャンネル 0 のコンペアマッチ 01: TPU チャンネル 1 のコンペアマッチ 10: TPU チャンネル 2 のコンペアマッチ 11: TPU チャンネル 3 のコンペアマッチ
6	G3CMS0	1	R/W	
5	G2CMS1	1	R/W	グループ 2 コンペアマッチセレクト 1、0 パルス出力グループ 2 の出力トリガを選択します。 00: TPU チャンネル 0 のコンペアマッチ 01: TPU チャンネル 1 のコンペアマッチ 10: TPU チャンネル 2 のコンペアマッチ 11: TPU チャンネル 3 のコンペアマッチ
4	G2CMS0	1	R/W	
3~0	—	すべて 1	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください。

11.3.5 PPG 出力モードレジスタ (PMR)

PMR は、PPG のパルス出力モードをグループ単位で設定します。反転出力に設定すると PODRH の値が 1 のとき端子に Low レベルを、PODRH の値が 0 のとき端子に High レベルを出力します。また、ノンオーバーラップ動作に設定すると PPG は、出力トリガとなる TPU のコンペアマッチ A、B で出力値を更新します。詳細は、「11.4.4 パルス出力ノンオーバーラップ動作」を参照してください。

ビット	7	6	5	4	3	2	1	0
ビット名	G3INV	G2INV	—	—	G3NOV	G2NOV	—	—
初期値 :	1	1	1	1	0	0	0	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ビット	ビット名	初期値	R/W	説明
7	G3INV	1	R/W	グループ 3 インバート パルス出力グループ 3 を直接出力させるか反転出力させるかを選択します。 0 : 反転出力 1 : 直接出力
6	G2INV	1	R/W	グループ 2 インバート パルス出力グループ 2 を直接出力させるか反転出力させるかを選択します。 0 : 反転出力 1 : 直接出力
5	—	1	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください。
4	—	1	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください。
3	G3NOV	0	R/W	グループ 3 ノンオーバーラップ パルス出力グループ 3 を通常動作させるか、ノンオーバーラップ動作させるかを選択します。 0 : 通常動作 (選択された TPU のコンペアマッチ A で出力値を更新) 1 : ノンオーバーラップ動作 (選択された TPU のコンペアマッチ A、B で出力値を更新)
2	G2NOV	0	R/W	グループ 2 ノンオーバーラップ パルス出力グループ 2 を通常動作させるか、ノンオーバーラップ動作させるかを選択します。 0 : 通常動作 (選択された TPU のコンペアマッチ A で出力値を更新) 1 : ノンオーバーラップ動作 (選択された TPU のコンペアマッチ A、B で出力値を更新)

ビット	ビット名	初期値	R/W	説明
1	—	0	R/W	リザーブビット リードすると常に0が読み出されます。ライトする値は常に0にしてください。
0	—	0	R/W	リザーブビット リードすると常に0が読み出されます。ライトする値は常に0にしてください。

11.4 動作説明

PPG 概要を図 11.2 に示します。PPG は、NDR の対応するビットをそれぞれ 1 にセットすることによりパルス出力状態となります。初期出力値は対応する PODR の初期設定値により決まります。その後、PCR で指定したコンペアマッチが発生すると、対応する NDR の値がそれぞれ PODR に転送されて出力値が更新されます。次のコンペアマッチが発生するまでに NDR に出力データをライトすることにより、コンペアマッチのたびに最大 8 ビットのデータを順次出力することができます。

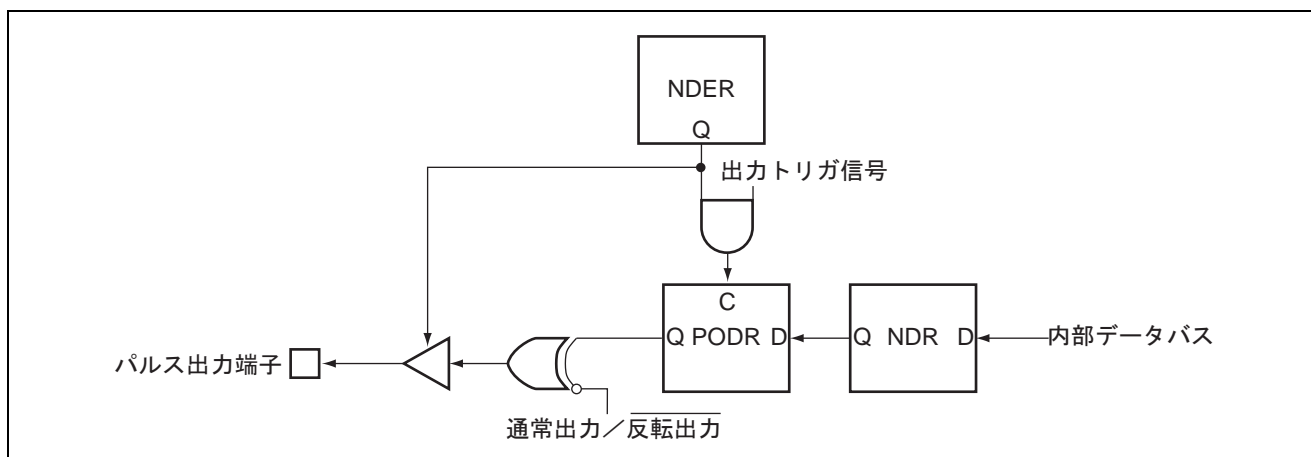


図 11.2 PPG 概要

11.4.1 出力タイミング

パルス出力許可状態で指定されたコンペアマッチが発生すると、NDR の内容が PODR に転送され、出力されます。このタイミングを図 11.3 に示します。コンペアマッチ A により、グループ 2、3 で通常出力を行った場合の例です。

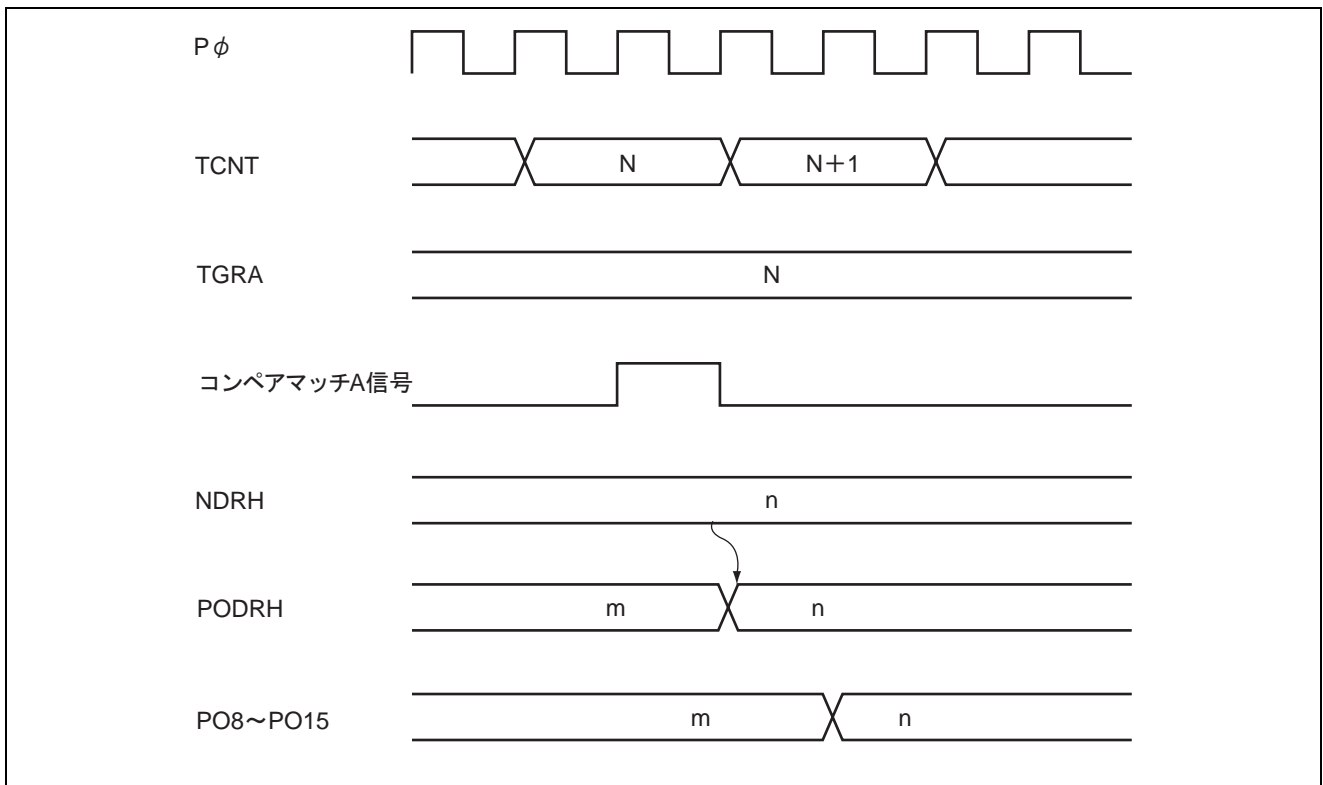


図 11.3 NDR の内容が転送・出力されるタイミング例

11.4.2 通常動作のパルス出力設定手順例

パルス出力通常動作の設定手順例を図 11.4 に示します。

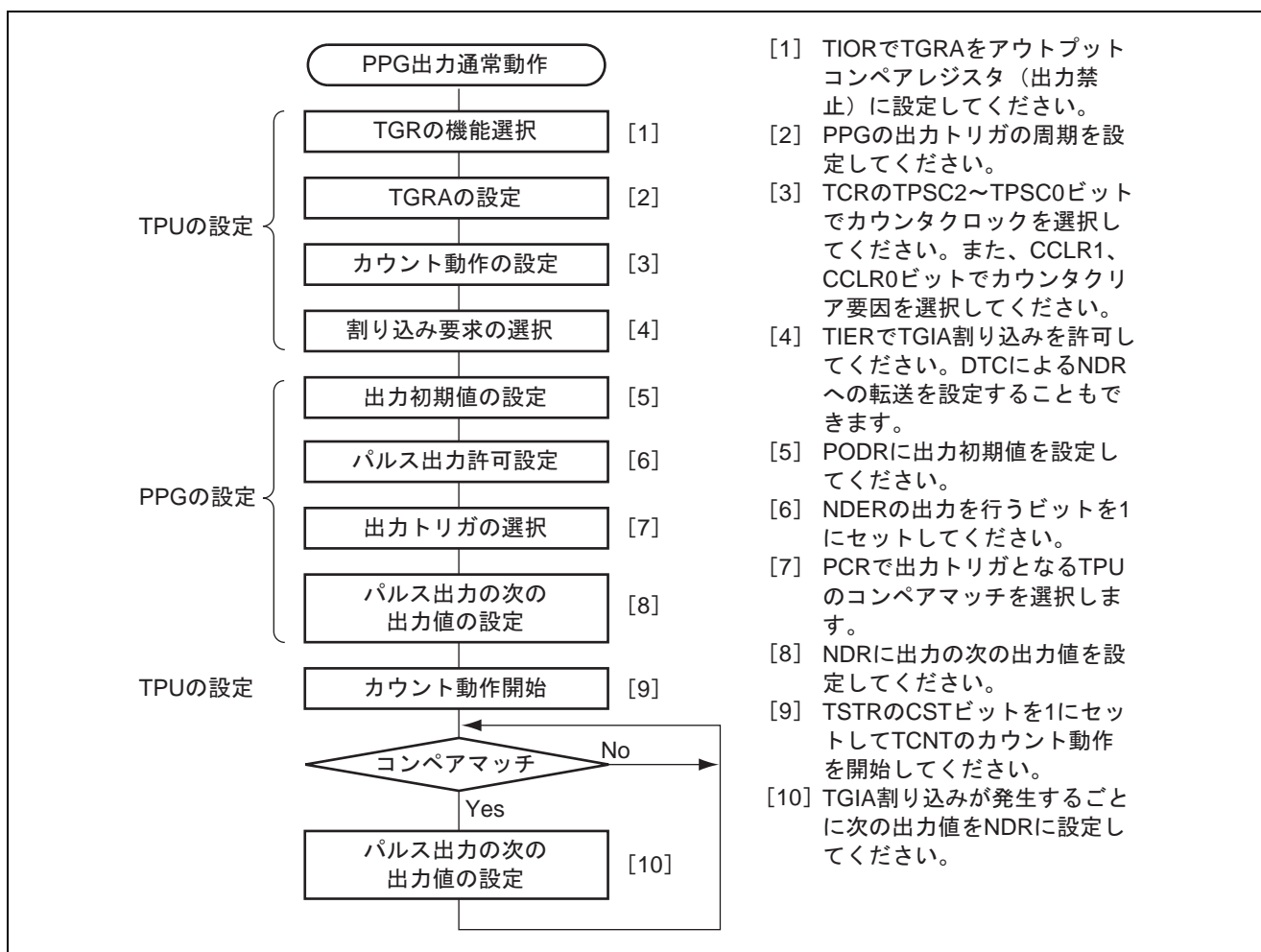


図 11.4 パルス出力通常動作の設定手順例

11.4.3 パルス出力通常動作例 (5相パルス出力例)

パルス出力を使用して一定周期で5相パルスを出力させた例を図 11.5 に示します。

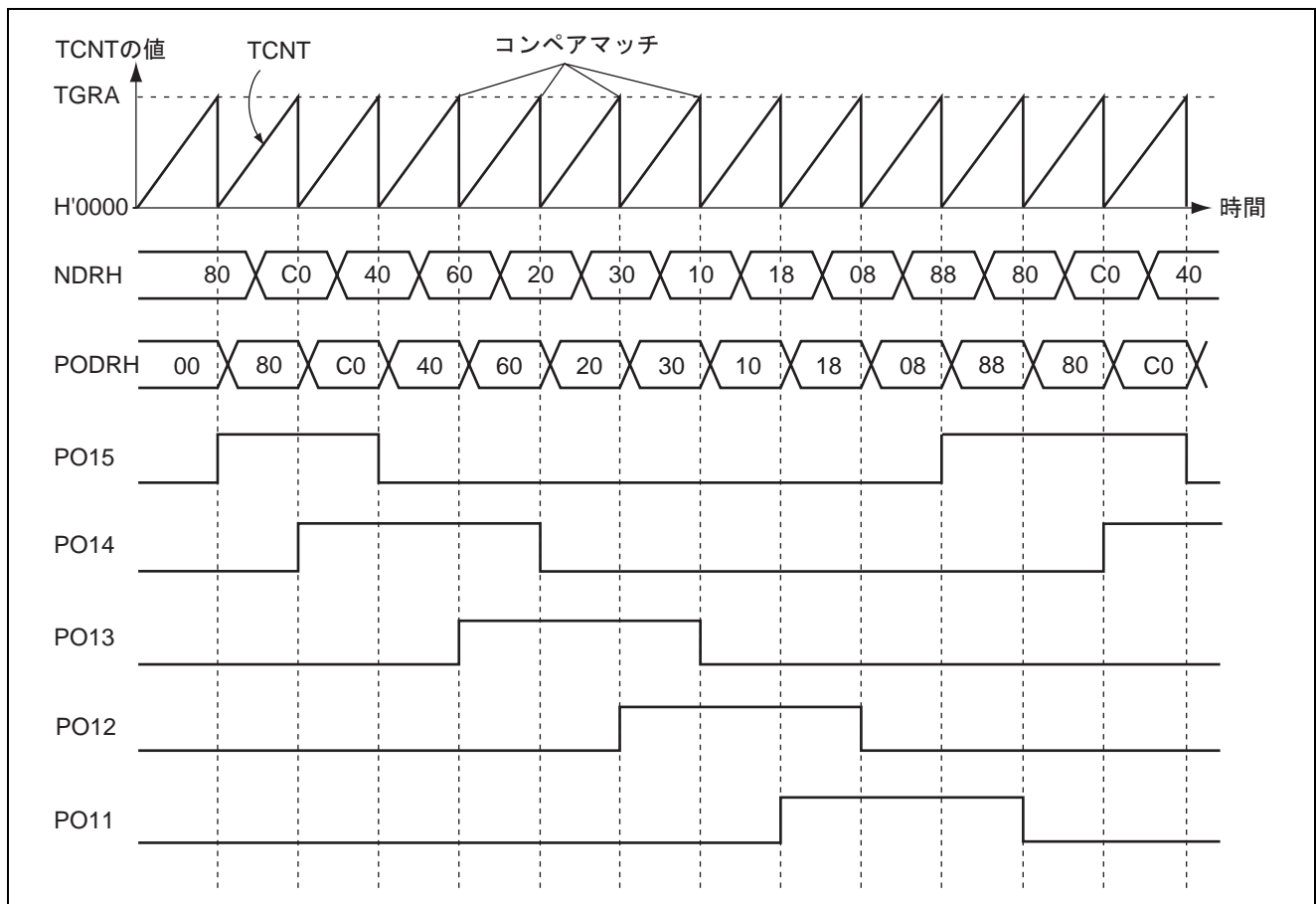


図 11.5 パルス出力通常動作例 (5相パルス出力例)

- 出力トリガとするTPUのTGRAをアウトプットコンペアレジスタに設定します。TGRAには周期を設定し、コンペアマッチAによるカウンタクリアを選択します。また、TIERのTGIEAビットを1にセットして、コンペアマッチ/インプットキャプチャA (TGIA) 割り込みを許可します。
- NDRHにH'F8をライトし、PCRのG3CMS1、G3CMS0ビットおよびG2CMS1、G2CMS0ビットにより、1で選択したTPUのコンペアマッチに出力トリガを設定します。NDRHに出力データH'80をライトします。
- TPU当該チャンネルの動作を開始しコンペアマッチAが発生すると、NDRHの内容がPODRHに転送され出力されます。TGIA割り込み処理でNDRHに次の出力データH'C0をライトします。
- 以後、TGIA割り込みで順次H'40、H'60、H'20、H'30、H'10、H'18、H'08、H'88……をライトすることで、5相の1-2相パルス出力を行うことができます。
TGIA割り込みでDTCを起動するように設定すれば、CPUの負荷なくパルス出力を行うことができます。

11.4.4 パルス出力ノンオーバーラップ動作

ノンオーバーラップ動作時の NDR から PODR への転送は以下のようになっています。

- コンペアマッチAではNDRの内容を常にPODRへ転送します。
- コンペアマッチBではNDRの転送するビットの内容が0のときのみ転送を行います。1のときは転送を行いません。

ノンオーバーラップ時のパルス出力動作を図 11.6 に示します。

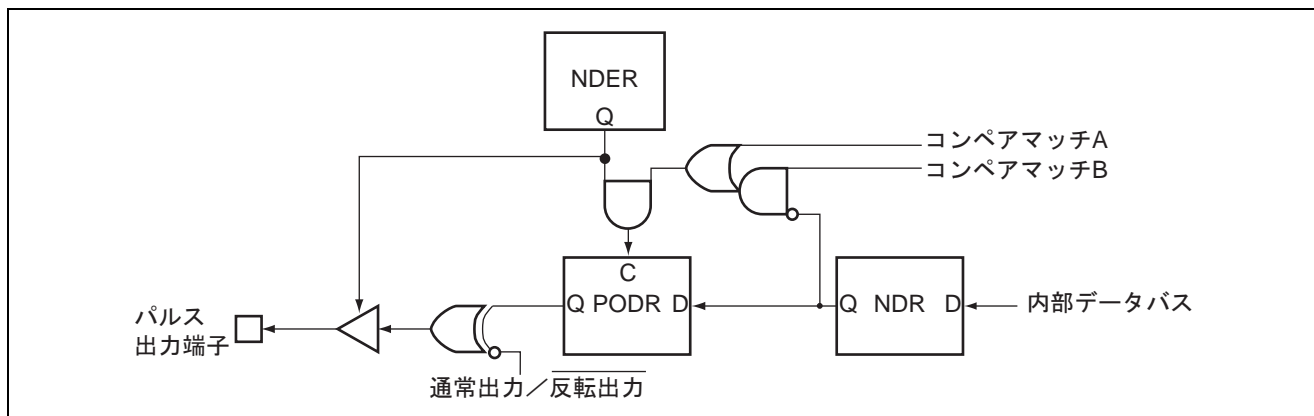


図 11.6 パルス出力ノンオーバーラップ動作

したがって、コンペアマッチ B をコンペアマッチ A よりも先に発生させることにより、0 データの転送を 1 データの転送に先立って行うことが可能です。

この場合、コンペアマッチ B が発生した後、コンペアマッチ A が発生するまで（ノンオーバーラップ期間）の間、NDR の内容を変更しないようにしてください。

これは TGIA 割り込みの割り込み処理ルーチンで NDR に次のデータをライトすることによって実現できます。また、TGIA 割り込みで DTC を起動することもできます。ただし、このライトは次のコンペアマッチ B が発生する前に行ってください。

このタイミングを図 11.7 に示します。

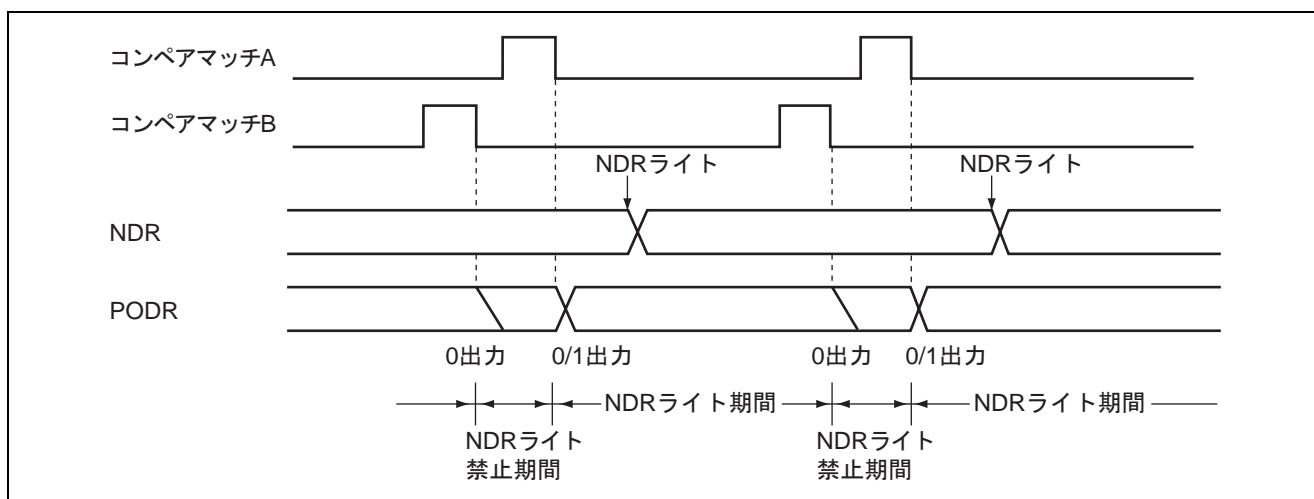


図 11.7 ノンオーバーラップ動作と NDR ライトタイミング

11.4.5 ノンオーバーラップ動作のパルス出力設定手順例

パルス出力ノンオーバーラップ動作の設定手順例を図 11.8 に示します。

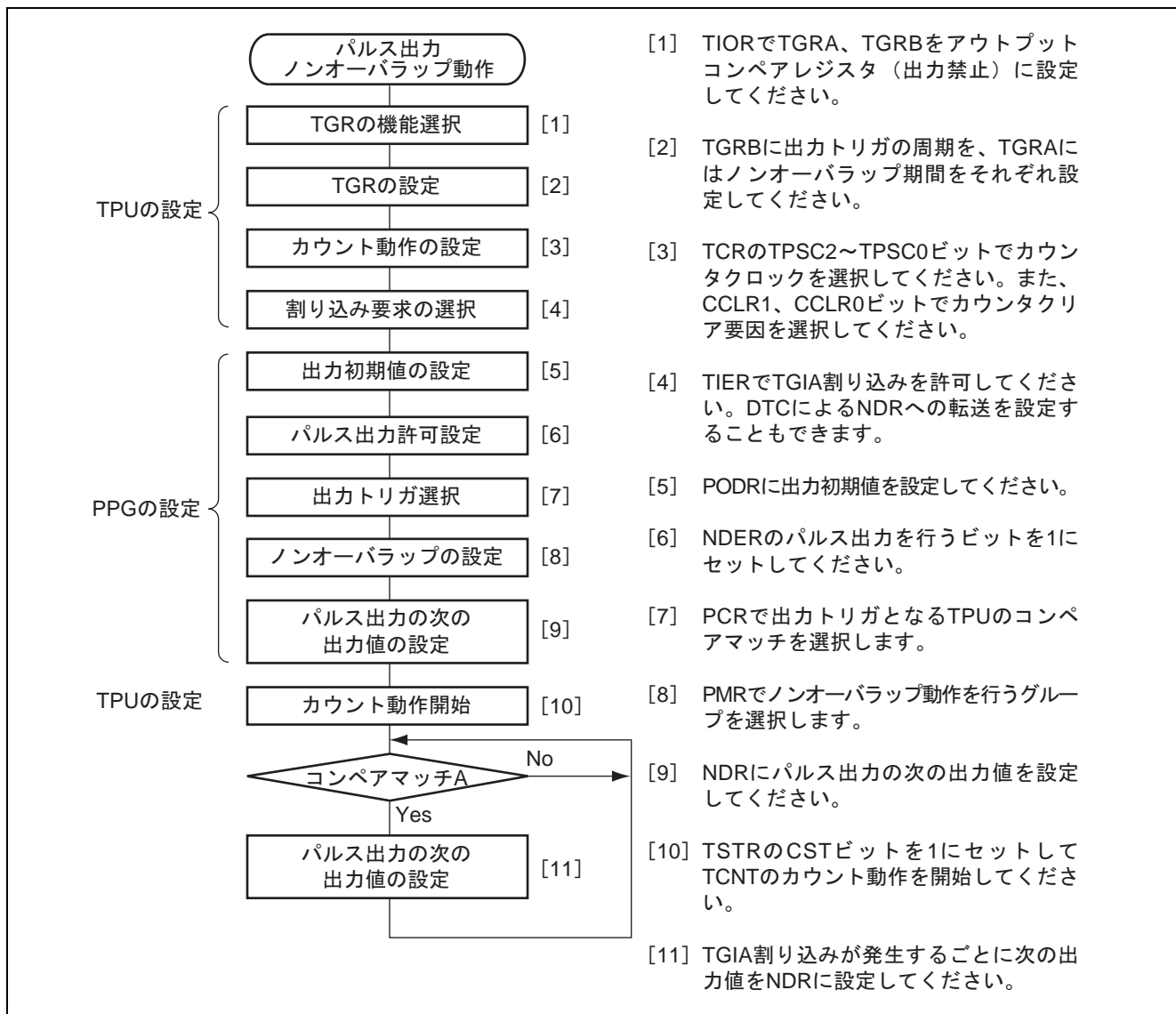


図 11.8 パルス出力ノンオーバーラップ動作の設定手順例

11.4.6 パルス出力ノンオーバーラップ動作例 (4相の相補ノンオーバーラップ出力例)

パルス出力を使用して4相の相補ノンオーバーラップのパルスを出力させた例を図11.9に示します。

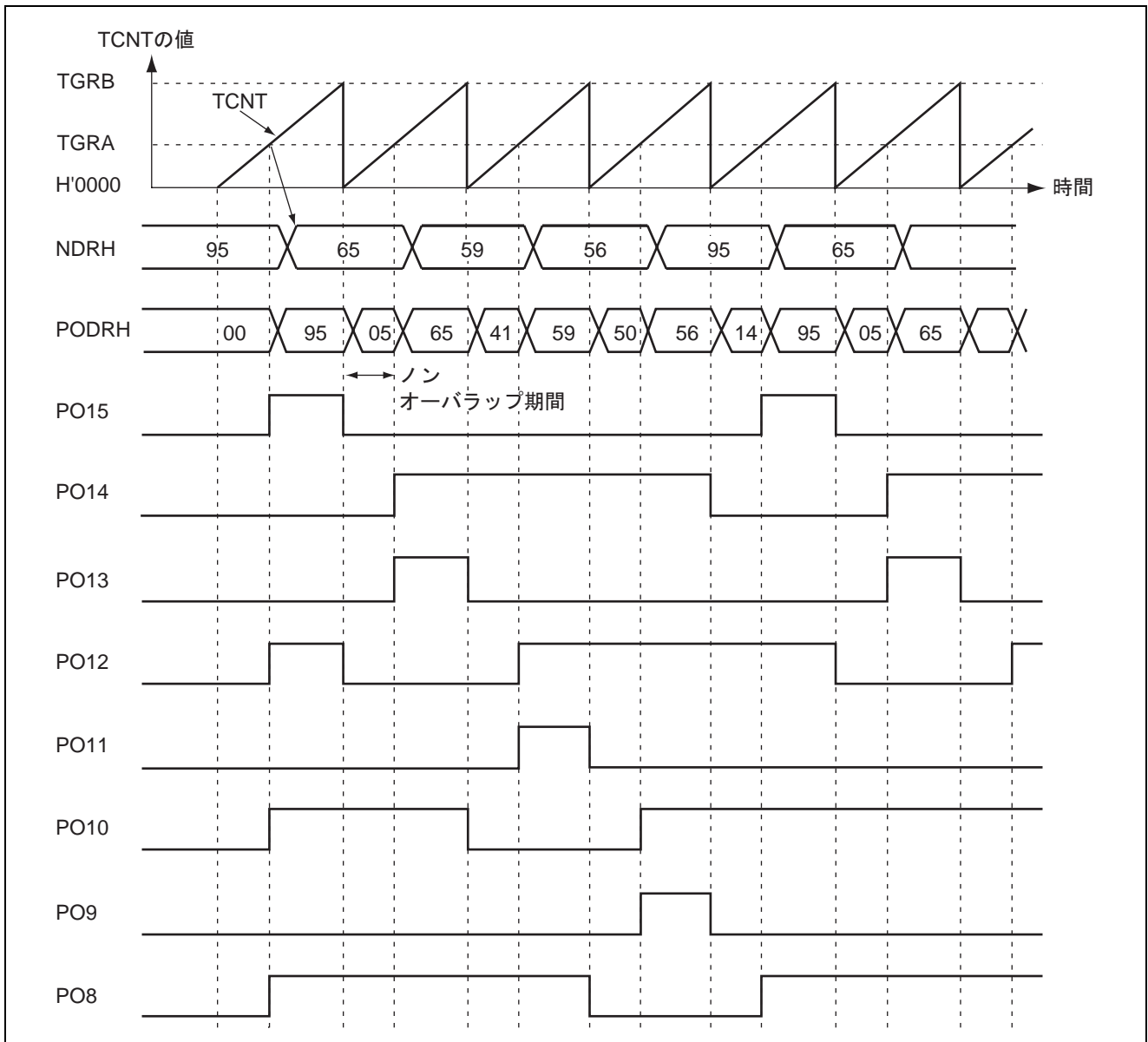


図 11.9 パルス出力ノンオーバーラップ動作例 (4相の相補ノンオーバーラップ出力)

1. 出力トリガとするTPUのTGRA、TGRBをアウトプットコンペアレジスタに設定します。TGRBには周期、TGRAにはノンオーバーラップ期間を設定し、コンペアマッチBによるカウンタクリアを選択します。また、TIERのTGIEAビットを1にセットして、TGIA割り込みを許可します。
2. NDERHにH'FFをライトし、PCRのG3CMS1、G3CMS0ビットおよびG2CMS1、G2CMS0ビットにより、1で選択したTPUのコンペアマッチに出力トリガを設定します。
PMRのG3NOV、G2NOVビットをそれぞれ1にセットして、ノンオーバーラップ動作を設定します。
NDRHに出力データH'95をライトします。
3. TPU当該チャネルの動作を開始すると、TGRBのコンペアマッチで1出力→0出力の変化、TGRAのコンペアマッチで0出力→1出力の変化を行います（0出力→1出力の変化はTGRAの設定値分遅延することになります）。
TGIA割り込み処理でNDRHに次の出力データH'65をライトします。
4. 以後、TGIA割り込みで順次H'59、H'56、H'95……をライトすることで、4相の相補ノンオーバーラップ出力を発生することができます。
TGIA割り込みでDTCを起動するように設定すれば、CPUの負荷なくパルス出力を行うことができます。

11.4.7 パルス反転出力

PMR の G3INV、および G2INV を 0 に設定すると、PODR の内容に対する反転値を端子出力することができます。

図 11.9 の設定で、さらに G3INV、G2INV を 0 にしたときの端子出力の様子を図 11.10 に示します。

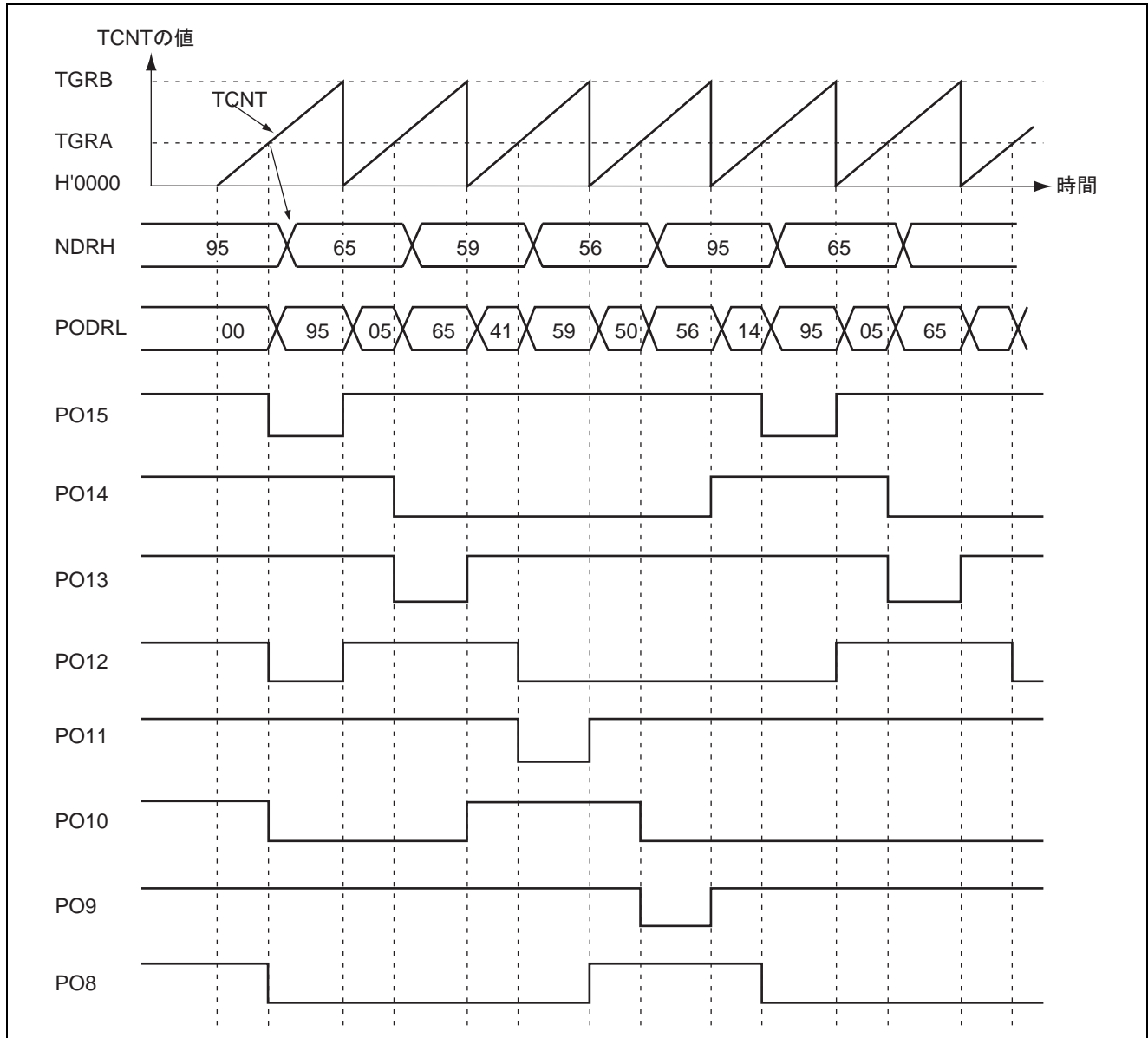


図 11.10 パルス反転出力例

11.4.8 インพุットキャプチャによるパルス出力

パルス出力は、TPU のコンペアマッチだけでなく、インพุットキャプチャによっても可能です。PCR によって選択された TPU の TGRA がインพุットキャプチャレジスタとして機能しているとき、インพุットキャプチャ信号によりパルス出力を行います。

このタイミングを図 11.11 に示します。

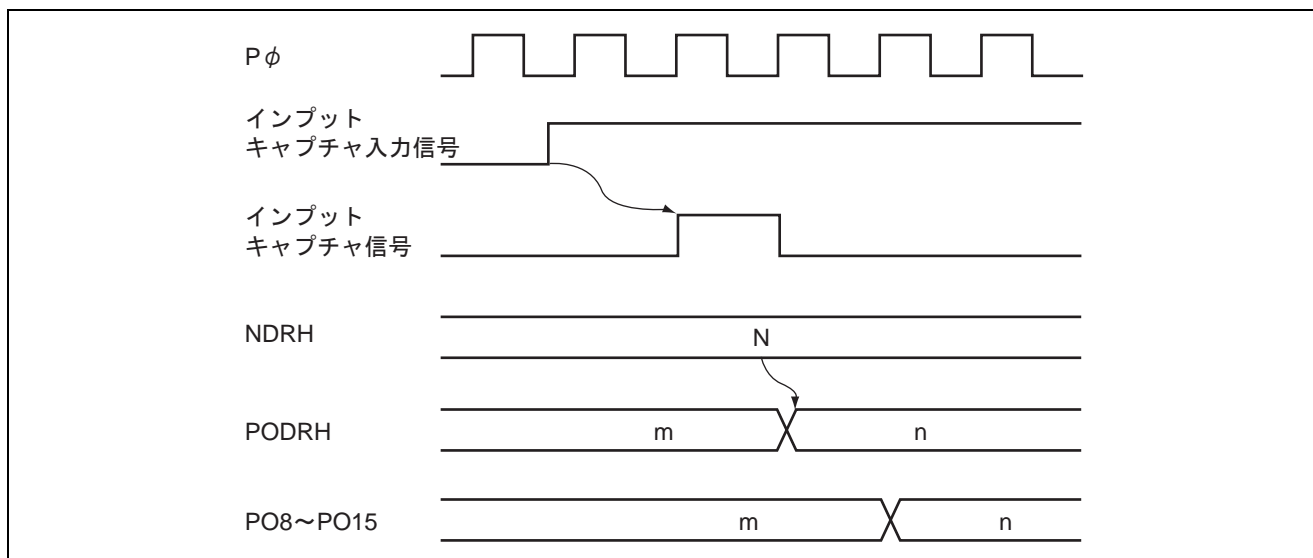


図 11.11 インพุットキャプチャによるパルス出力例

11.5 使用上の注意事項

11.5.1 モジュールストップモードの設定

モジュールストップコントロールレジスタにより、PPG の動作禁止/許可を設定することが可能です。初期値では、PPG の動作は停止します。モジュールストップモードを解除することにより、レジスタのアクセスが可能になります。詳細は「第 23 章 低消費電力」を参照してください。

11.5.2 パルス出力端子の動作

PO8~PO15 は TPU などの他の周辺機能の端子と兼用になっています。これらの端子は、他の周辺機能が出力許可状態になっているときは、パルス出力を行うことができません。ただし、NDR から PODR への転送は、端子の状態にかかわらず常に行うことが可能です。

端子機能の変更は、出力トリガが発生しない状態で行ってください。

12. ウォッチドッグタイマ (WDT)

ウォッチドッグタイマ (WDT) は 8 ビットのタイマで、システムの暴走などによりカウンタの値が書き換えられずにオーバーフローすると、本 LSI 内部をリセットすることができます。

ウォッチドッグタイマとして使用しない場合は、インターバルタイマとして使用することもできます。インターバルタイマとして使用する場合は、カウンタがオーバーフローするごとにインターバルタイマ割り込みを発生します。WDT のブロック図を図 12.1 に示します。

12.1 特長

- 8種類のカウンタ入力クロックを選択可能
- ウォッチドッグタイマモードとインターバルタイマモードを切り替え可能

ウォッチドッグタイマモード

- カウンタがオーバーフローしたときに、本LSI内部をリセット可能

インターバルタイマモード

- カウンタがオーバーフローすると、インターバルタイマ割り込み (WOVI) を発生

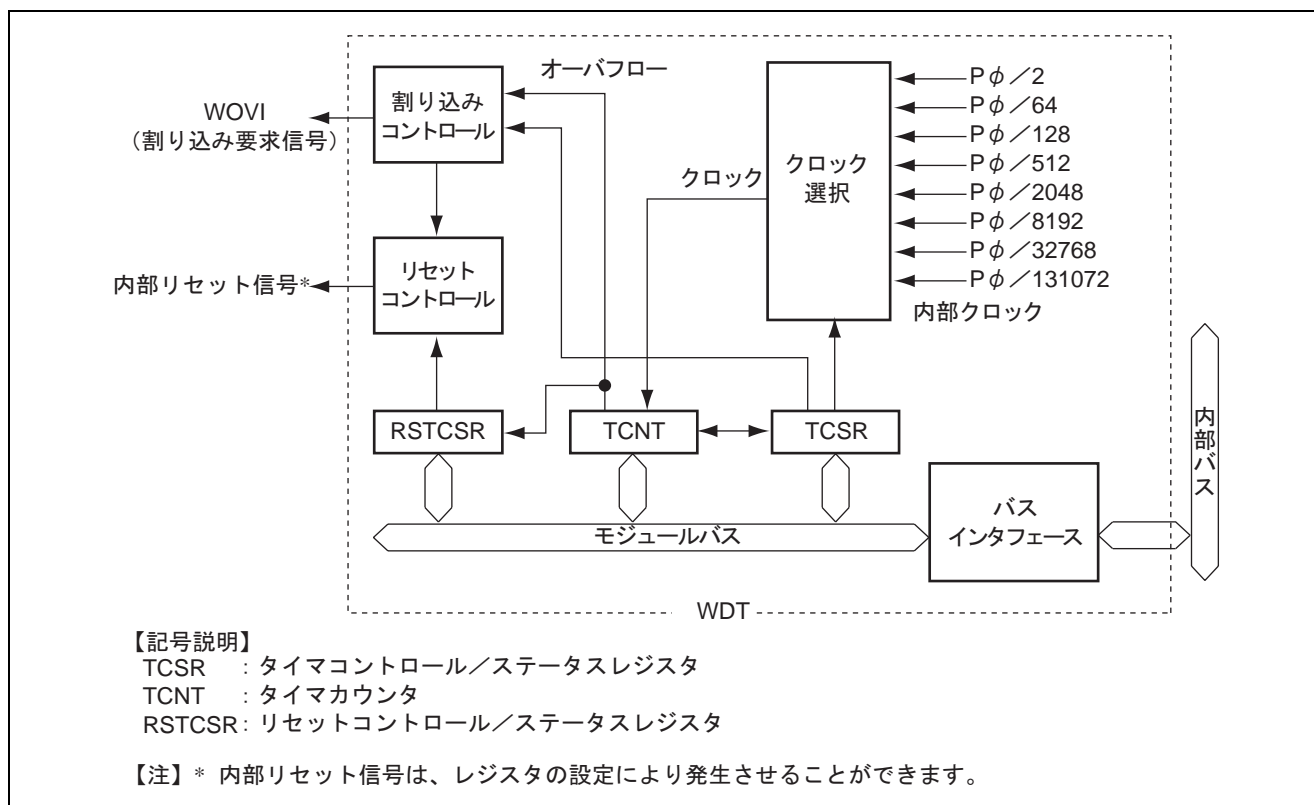


図 12.1 WDT のブロック図

12.2 レジスタの説明

WDTには、以下のレジスタがあります。TCNT、TCSR、RSTCSRは容易に書き換えられないように、ライト方法が一般のレジスタと異なっています。詳細は、「12.5.1 レジスタアクセス時の注意」を参照してください。

- タイマカウンタ (TCNT)
- タイマコントロール/ステータスレジスタ (TCSR)
- リセットコントロール/ステータスレジスタ (RSTCSR)

12.2.1 タイマカウンタ (TCNT)

TCNTは、リード/ライト可能な8ビットのアップカウンタです。TCNTは、タイマコントロール/ステータスレジスタ (TCSR) のTMEビットが0のとき、H'00に初期化されます。

ビット	7	6	5	4	3	2	1	0
ビット名								
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

12.2.2 タイマコントロール/ステータスレジスタ (TCSR)

TCSR は、TCNT に入力するクロック、モードの選択などを行います。

ビット	7	6	5	4	3	2	1	0
ビット名	OVF	WT/ \bar{I} T	TME	—	—	CKS2	CKS1	CKS0
初期値 :	0	0	0	1	1	0	0	0
R/W :	R/(W)*	R/W	R/W	R	R	R/W	R/W	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。

ビット	ビット名	初期値	R/W	説 明
7	OVF	0	R/(W)*	<p>オーバフローフラグ</p> <p>インターバルタイマモードで TCNT がオーバフローしたことを示します。フラグをクリアするための0クリアのみ可能です。</p> <p>[セット条件]</p> <ul style="list-style-type: none"> インターバルタイマモードで TCNT がオーバフロー (H'FF→H'00) したとき <p>ただし、ウォッチドッグタイマモードで、内部リセット要求を選択した場合は、セット後、内部リセットにより自動的にクリアされます。</p> <p>[クリア条件]</p> <ul style="list-style-type: none"> OVF=1 の状態で、TCSR をリード後、OVF に 0 をライトしたとき (割り込みを使用し、CPU によってクリアする場合、0 ライト後必ずフラグをリードしてください)
6	WT/ \bar{I} T	0	R/W	<p>タイマモードセレクト</p> <p>ウォッチドッグタイマとして使用するか、インターバルタイマとして使用するかを選択します。</p> <p>0 : インターバルタイマモード</p> <p>TCNT がオーバフローしたとき、インターバルタイマ割り込み (WOVI) を要求します。</p> <p>1 : ウォッチドッグタイマモード</p> <p>TCNT がオーバフロー、RSTE=1 のとき、LSI 内部をリセットします。</p>
5	TME	0	R/W	<p>タイマイネーブル</p> <p>このビットを 1 にセットすると TCNT がカウントを開始します。クリアすると TCNT はカウント動作を停止し、H'00 に初期化されます。</p>
4	—	1	R	リザーブビット
3	—	1	R	これらのビットはリードのみ有効で、ライトは無効です。

ビット	ビット名	初期値	R/W	説明
2	CKS2	0	R/W	クロックセレクト 2~0 TCNT に入力するクロックを選択します。() 内は Pφ=20MHz のときの オーバフロー周期を表します。 000 : クロック Pφ/2 (周期 25.6μs) 001 : クロック Pφ/64 (周期 819.2μs) 010 : クロック Pφ/128 (周期 1.6ms) 011 : クロック Pφ/512 (周期 6.6ms) 100 : クロック Pφ/2048 (周期 26.2ms) 101 : クロック Pφ/8192 (周期 104.9ms) 110 : クロック Pφ/32768 (周期 419.4ms) 111 : クロック Pφ/131072 (周期 1.68s)
1	CKS1	0	R/W	
0	CKS0	0	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。

12.2.3 リセットコントロール/ステータスレジスタ (RSTCSR)

RSTCSR は、TCNT のオーバーフローによる内部リセット信号の発生を制御し、内部リセット信号の種類を選択します。RSTCSR は、 $\overline{\text{RES}}$ 端子からのリセット信号で H'1F に初期化されます。WDT のオーバーフローによる内部リセット信号では初期化されません。

ビット	7	6	5	4	3	2	1	0
ビット名	WOVF	RSTE	—	—	—	—	—	—
初期値 :	0	0	0	1	1	1	1	1
R/W :	R/(W)*	R/W	R/W	R	R	R	R	R

【注】 * フラグをクリアするための0ライトのみ可能です。

ビット	ビット名	初期値	R/W	説明
7	WOVF	0	R/(W)*	ウォッチドッグタイマオーバーフローフラグ ウォッチドッグタイマモードで、TCNT がオーバーフローするとセットされます。インターバルタイマモードではセットされません。0 ライトのみ可能です。 [セット条件] • ウォッチドッグタイマモードで TCNT がオーバーフロー(H'FF→H'00)したとき [クリア条件] • 1 の状態をリードした後、0 をライトしたとき (割り込みを使用し CPU によってクリアする場合、0 ライト後に必ずフラグをリードしてください。)
6	RSTE	0	R/W	リセットイネーブル ウォッチドッグタイマモードで TCNT のオーバーフローにより LSI 内部をリセットするかどうかを選択します。 0 : TCNT がオーバーフローしても、内部はリセットされません。 (本 LSI 内部はリセットされませんが、WDT 内の TCNT、TCSR はリセットされます。) 1 : TCNT がオーバーフローすると内部がリセットされます。
5	—	0	R/W	リザーブビット リードすると常に 0 が読み出されます。ライトする値は常に 0 にしてください。
4~0	—	1	R	リザーブビット これらのビットはリードのみ有効で、ライトは無効です。

【注】 * フラグをクリアするための 0 ライトのみ可能です。

12.3 動作説明

12.3.1 ウォッチドッグタイマモード

ウォッチドッグタイマモードとして使用するときは、TCSR の $\overline{WT}/\overline{IT}$ ビット=1 に、TME ビット=1 に設定してください。

ウォッチドッグタイマモードで TCNT がオーバーフローすると、RSTCSR の WOVF ビットが 1 にセットされます。また、ウォッチドッグタイマモードで、RSTCSR の RSTE ビットを 1 にセットすると、システムの暴走などにより TCNT の値が書き換えられずオーバーフローしたとき、本 LSI の内部をリセットする信号が発生します。システムが正常に動作している間は、TCNT のオーバーフローは発生しません。TCNT がオーバーフローする前に必ず TCNT の値を書き換えて（通常は H'00 をライトする）、オーバーフローを発生させないようにプログラムしてください。

TCNT がオーバーフローしたときに、 \overline{RES} 端子からの入力信号によるリセットと WDT のオーバーフローによるリセットが同時に発生したときは、 \overline{RES} 端子によるリセットが優先され、RSTCSR の WOVF ビットは 0 にクリアされます。

内部リセット信号は、Pφ で 519 ステートの間出力されます。

RSTE ビット=1 のときは本 LSI の内部をリセットする信号が発生し、システムクロックコントロールレジスタ (SCKCR) がリセットされるため、Pφ の入力クロックに対する倍率は初期値になります。

RSTE ビット=0 のときは内部をリセットする信号が発生せず、SCKCR の設定が保持されるため Pφ の入力クロックに対する倍率は変化しません。

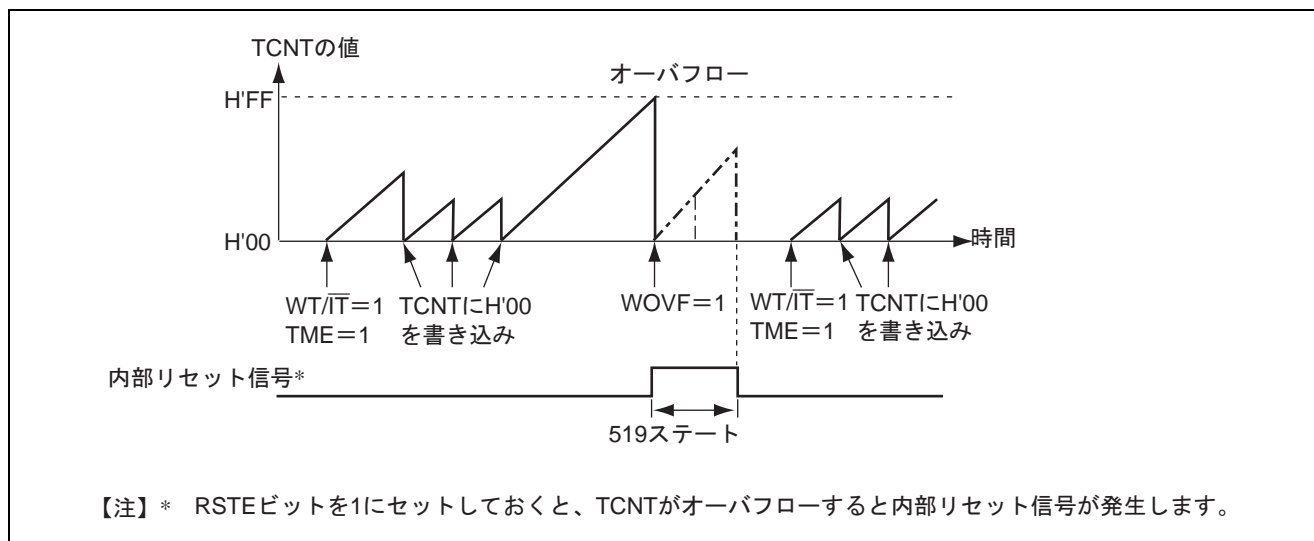


図 12.2 ウォッチドッグタイマモード時の動作

12.3.2 インターバルタイマモード

インターバルタイマとして使用するときは、TCSR の WT/\overline{IT} ビット=0 に、TME ビット=1 に設定してください。

インターバルタイマとして動作しているときは、TCNT がオーバーフローするごとにインターバルタイマ割り込み (WOVI) が発生します。したがって、一定時間ごとに割り込みを発生させることができます。

インターバルタイマモードで TCNT がオーバーフローすると、TCSR の OVF ビットが 1 にセットされ、同時にインターバルタイマ割り込み (WOVI) が要求されます。

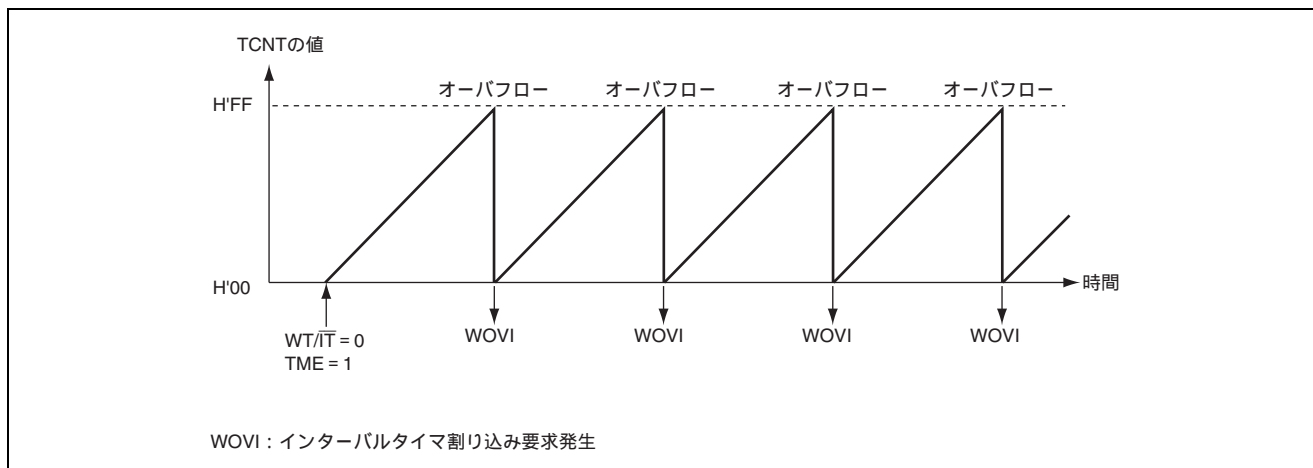


図 12.3 インターバルタイマモード時の動作

12.4 割り込み要因

インターバルタイマモード時、オーバーフローによりインターバルタイマ割り込み (WOVI) を発生します。インターバルタイマ割り込みは、TCSR の OVF フラグが 1 にセットされると常に要求されます。割り込み処理ルーチンで必ず OVF を 0 にクリアしてください。

表 12.1 WDT の割り込み要因

名称	割り込み要因	割り込みフラグ
WOVI	TCNT のオーバーフロー	OVF

12.5 使用上の注意事項

12.5.1 レジスタアクセス時の注意

TCNT、TCSR、RSTCSR は、容易に書き換えられないように、ライト方法が一般のレジスタと異なっています。次の方法で、リード/ライトを行ってください。

(1) TCNT、TCSR、RSTCSR へのライト

TCNT、TCSR へライトするときは、必ずワード転送命令を使用してください。バイト転送命令では、ライトできません。

ライト時は、TCNT と TCSR が同一アドレスに割り当てられています。このため、**図 12.4** に示すようにしてデータを転送してください。転送すると、下位バイトのデータが TCNT または TCSR へライトされます。

RSTCSR へライトするときは、アドレス (H'FFA6) に対してワード転送を行ってください。バイト転送命令では、ライトできません。

WOVF ビットへ 0 をライトする場合と、RSTE ビットにライトする場合は、ライトの方法が異なります。このため、**図 12.4** に示すようにしてデータを転送してください。

転送すると、WOVF ビットが 0 にクリアされます。このとき、RSTE ビットは影響を受けません。RSTE ビットにライトするときは、**図 12.4** に示すようにしてデータを転送してください。転送すると、下位バイトのビット 6 の値が RSTE ビットにライトされます。このとき、WOVF ビットは影響を受けません。

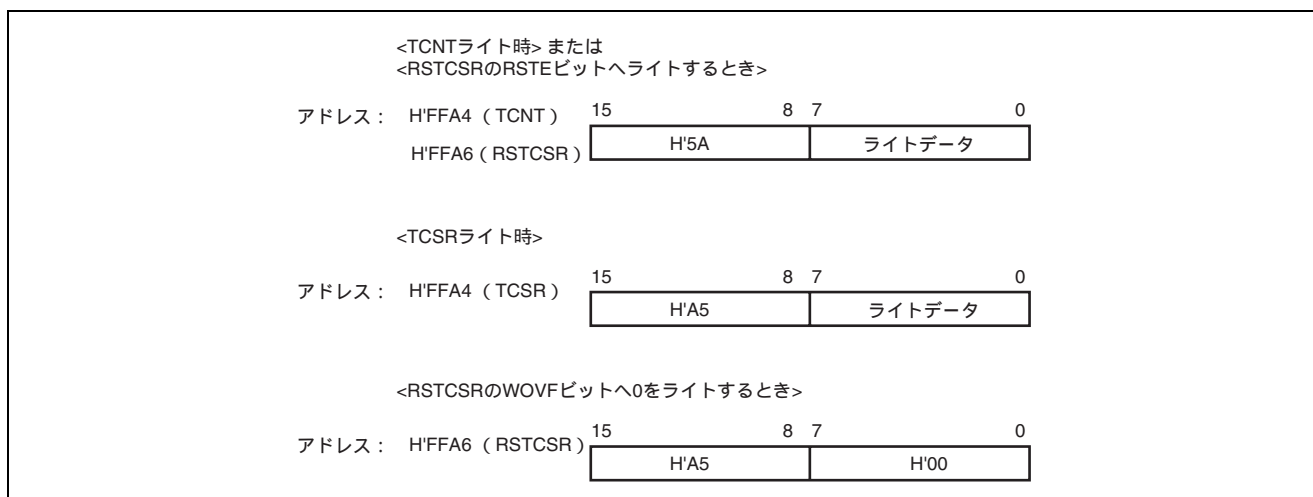


図 12.4 TCNT、TCSR、RSTCSR へのライト

(2) TCNT、TCSR、RSTCSR からのリード

リードは、一般のレジスタと同様の方法で行うことができます。TCSR はアドレス (H'FFA4) に、TCNT はアドレス (H'FFA5) に、RSTCSR はアドレス (H'FFA7) にそれぞれ割り当てられています。

12.5.2 タイマカウンタ (TCNT) のライトとカウントアップの競合

TCNT のライトサイクル中の T2 ステートでカウントアップが発生しても、カウントアップされずに TCNT へのカウンタライトが優先されます。これを図 12.5 に示します。

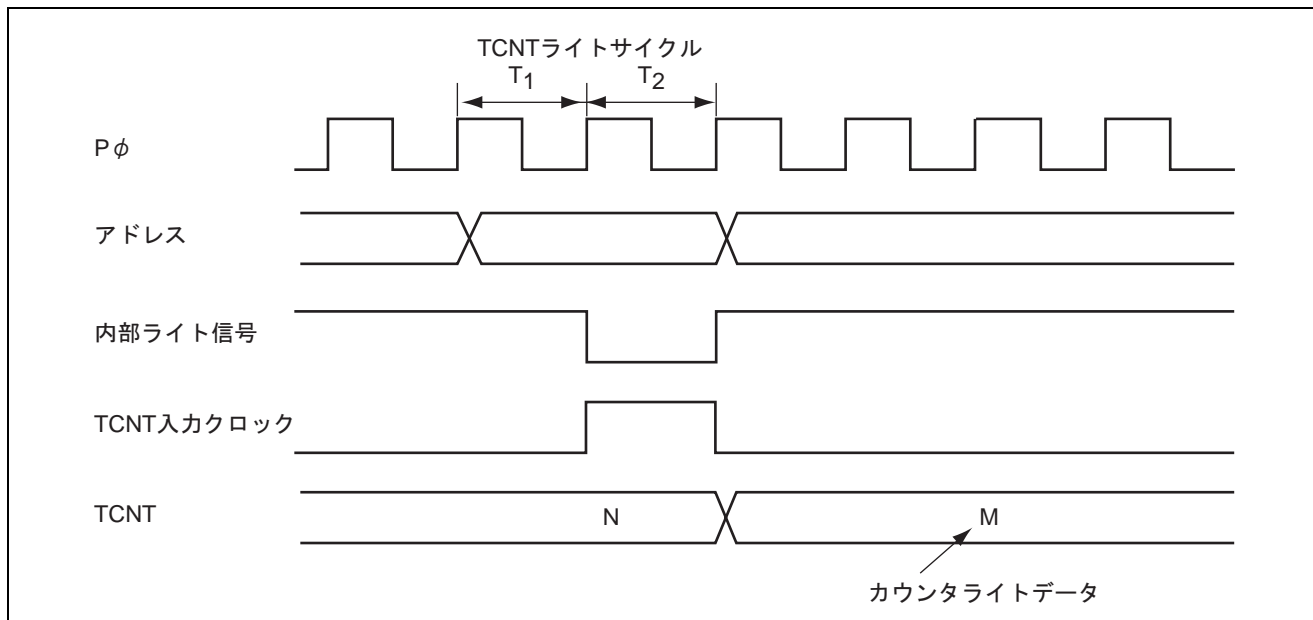


図 12.5 TCNT のライトとカウントアップの競合

12.5.3 CKS2～CKS0 ビットの書き換え

WDT の動作中に TCSR の CKS2～CKS0 ビットを書き換えると、カウントアップが正しく行われな場合があります。CKS2～CKS0 ビットを書き換えるときは、必ず WDT を停止させてから (TME ビットを 0 にクリアしてから) 行ってください。

12.5.4 ウォッチドッグタイマモードとインターバルタイマモードの切り替え

WDT の動作中にウォッチドッグタイマモードとインターバルタイマモードを切り替えると、正しい動作が行われな場合があります。タイマモードの切り替えは、必ず WDT を停止させてから (TME ビットを 0 にクリアしてから) 行ってください。

12.5.5 ウォッチドッグタイマモードとソフトウェアスタンバイモードへの遷移

ウォッチドッグタイマモードとして動作しているときは、SBYCR の SSBY ビットを 1 にセットした状態で SLEEP 命令を実行してもソフトウェアスタンバイモードには遷移せず、スリープモードに遷移します。

ソフトウェアスタンバイモードに遷移させる場合は、WDT を停止させてから (TME ビットを 0 にクリア) SLEEP 命令を実行してください。

インターバルタイマモードとして動作している場合は、SBYCR の SSBY ビットを 1 にセットした状態で SLEEP 命令を実行するとソフトウェアスタンバイモードに遷移します。

13. シリアルコミュニケーションインタフェース (SCI)

本 LSI は独立した 2 チャンネルのシリアルコミュニケーションインタフェース (SCI : Serial Communication Interface) を備えています。SCI は、調歩同期式とクロック同期式の 2 方式のシリアル通信が可能です。調歩同期方式では 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 とのシリアル通信ができます。調歩同期式モードでは複数のプロセッサ間のシリアル通信機能 (マルチプロセッサ通信機能) を備えています。

13.1 特長

- シリアルデータ通信モードを調歩同期式またはクロック同期式に設定可能
- 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、送信部と受信部はともにダブルバッファ構造になっていますので、連続送受信が可能です。
- 内蔵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として外部クロックの選択も可能です。
- 割り込み要因 : 4種類
送信終了、送信データエンプティ、受信データフル、受信エラーの割り込み要因があります。また、送信データエンプティ、受信データフル割り込み要因により DTC または DMAC を起動することができます。
- モジュールストップモードの設定可能

調歩同期式モード

- データ長 : 7ビット / 8ビット 選択可能
- ストップビット長 : 1ビット / 2ビット 選択可能
- パリティ : 偶数パリティ / 奇数パリティ / パリティなしから選択可能
- 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 RxD 端子のレベルを直接リードすることでブレークを検出可能

クロック同期式モード

- データ長 : 8ビット
- 受信エラーの検出 : オーバランエラー

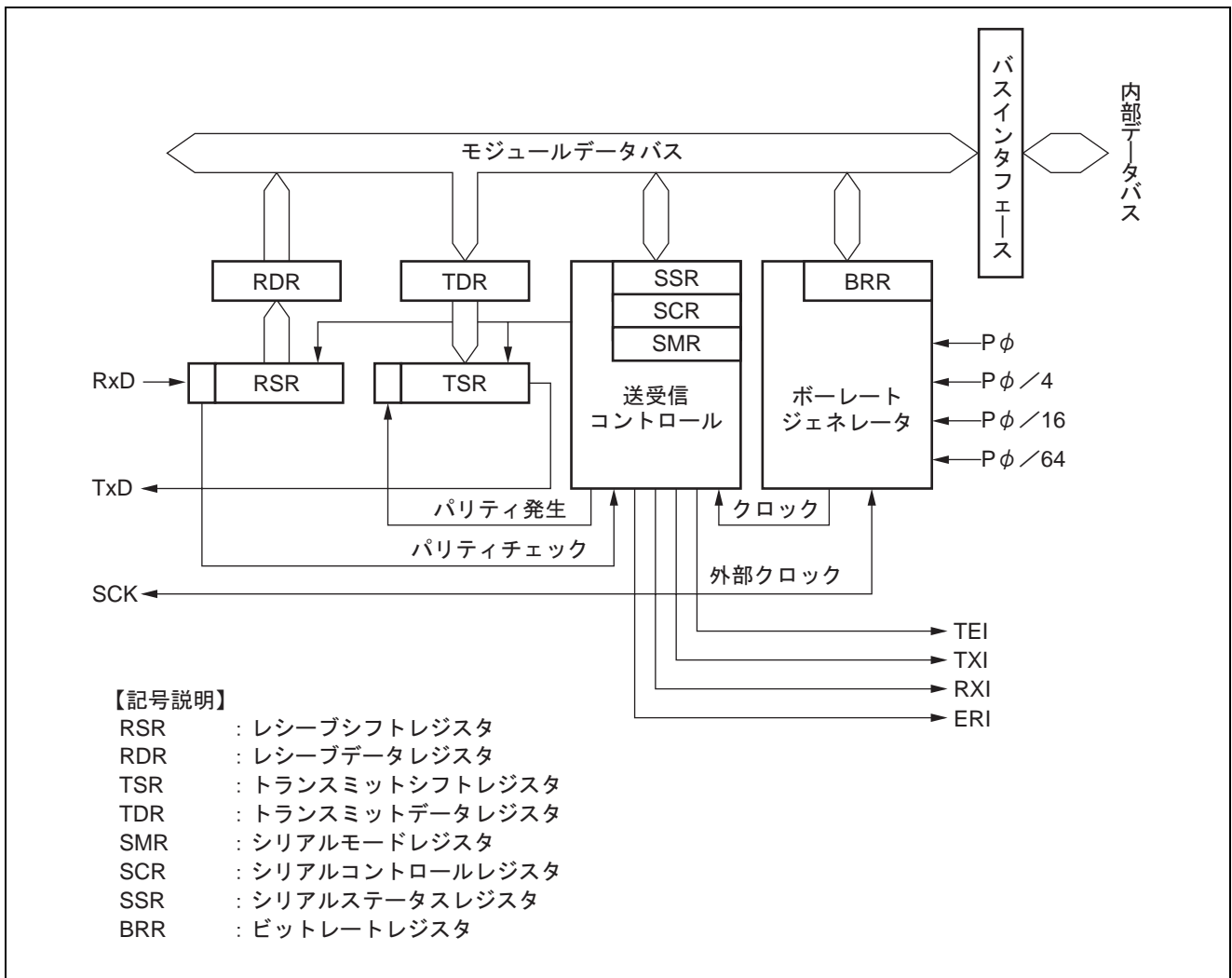


図 13.1 SCI のブロック図

13.2 入出力端子

SCI には、表 13.1 の入出力端子があります。

表 13.1 端子構成

チャンネル	端子名*	入出力	機能
3	SCK3	入出力	チャンネル 3 のクロック入出力端子
	RxD3	入力	チャンネル 3 の受信データ入力端子
	TxD3	出力	チャンネル 3 の送信データ出力端子
4	SCK4	入出力	チャンネル 4 のクロック入出力端子
	RxD4	入力	チャンネル 4 の受信データ入力端子
	TxD4	出力	チャンネル 4 の送信データ出力端子

【注】 * 本文中ではチャンネルを省略し、それぞれ SCK、RxD、TxD と略称します。

13.3 レジスタの説明

SCIには以下のレジスタがあります。

チャンネル3

- レシーブシフトレジスタ_3 (RSR_3)
- トランスミットシフトレジスタ_3 (TSR_3)
- レシーブデータレジスタ_3 (RDR_3)
- トランスミットデータレジスタ_3 (TDR_3)
- シリアルモードレジスタ_3 (SMR_3)
- シリアルコントロールレジスタ_3 (SCR_3)
- シリアルステータスレジスタ_3 (SSR_3)
- ビットレートレジスタ_3 (BRR_3)

チャンネル4

- レシーブシフトレジスタ_4 (RSR_4)
- トランスミットシフトレジスタ_4 (TSR_4)
- レシーブデータレジスタ_4 (RDR_4)
- トランスミットデータレジスタ_4 (TDR_4)
- シリアルモードレジスタ_4 (SMR_4)
- シリアルコントロールレジスタ_4 (SCR_4)
- シリアルステータスレジスタ_4 (SSR_4)
- ビットレートレジスタ_4 (BRR_4)

13.3.1 レシーブシフトレジスタ (RSR)

RSR は RxD 端子から入力されたシリアルデータをパラレル変換するための受信シフトレジスタです。1 フレーム分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 から直接アクセスすることはできません。

13.3.2 レシーブデータレジスタ (RDR)

RDR は受信データを格納するための 8 ビットのレジスタです。1 フレーム分のデータを受信すると RSR から受信データがこのレジスタへ転送され、RSR は次のデータを受信可能となります。RSR と RDR はダブルバッファ構造になっているため連続受信動作が可能です。RDR のリードは SSR の RDRF が 1 にセットされていることを確認して行ってください。RDR は CPU からライトできません。

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

13.3.3 トランスミットデータレジスタ (TDR)

TDR は送信データを格納するための 8 ビットのレジスタです。TSR に空きを検出すると TDR にライトされた送信データは TSR に転送されて送信を開始します。TDR と TSR はダブルバッファ構造になっているため連続送信動作が可能です。1 フレーム分のデータを送信したとき TDR に次の送信データがライトされていれば TSR へ転送して送信を継続します。TDR は CPU から常にリード/ライト可能です。TDR への送信データのライトは必ず SSR の TDRE が 1 にセットされていることを確認して行ってください。

ビット	7	6	5	4	3	2	1	0
ビット名								
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

13.3.4 トランスミットシフトレジスタ (TSR)

TSR はシリアルデータを送信するためのシフトレジスタです。TDR にライトされた送信データは自動的に TSR に転送され、TxD 端子に送出することでシリアルデータの送信を行います。CPU からは直接アクセスすることはできません。

13.3.5 シリアルモードレジスタ (SMR)

SMRは通信フォーマットと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ためのレジスタです。

ビット	7	6	5	4	3	2	1	0
ビット名	C/ \bar{A}	CHR	PE	O/ \bar{E}	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*

【注】 * TE=RE=0の場合のみライト可能です。

ビット	ビット名	初期値	R/W	説明
7	C/ \bar{A}	0	R/W*	コミュニケーションモード 0: 調歩同期式モードで動作します。 1: クロック同期式モードで動作します。
6	CHR	0	R/W*	キャラクタレングス (調歩同期式モードのみ有効) 0: データ長 8 ビットで送受信します。 1: データ長 7 ビットで送受信します。 送信では TDR の MSB (ビット 7) は送信されません。 クロック同期式モードではデータ長は 8 ビット固定です。
5	PE	0	R/W*	パリティイネーブル (調歩同期式モードのみ有効) このビットが 1 のとき、送信時はパリティビットを付加し、受信時はパリティチェックを行います。 マルチプロセッサフォーマットではこのビットの設定にかかわらずパリティビットの付加、チェックは行いません。
4	O/ \bar{E}	0	R/W*	パリティモード (調歩同期式モードで PE = 1 のときのみ有効) 0: 偶数パリティで送受信します。 1: 奇数パリティで送受信します。 偶数パリティでは、パリティビットと送受信キャラクタをあわせて、その中の 1 の数の合計が偶数個になるようにパリティビットを決定します。同様に、奇数パリティでは、パリティビットと送受信キャラクタをあわせて、その中の 1 の数の合計が奇数個になるようにパリティビットを決定します。
3	STOP	0	R/W*	ストップビットレングス (調歩同期式モードのみ有効) 0: 送信時 1 ストップビット 1: 送信時 2 ストップビット 受信時はこのビットの設定にかかわらずストップビットの 1 ビット目のみチェックし、2 ビット目が 0 の場合は次の送信フレームのスタートビットと見なします。

ビット	ビット名	初期値	R/W	説明
2	MP	0	R/W*	マルチプロセッサモード (調歩同期式モードのみ有効) このビットが1のとき、マルチプロセッサ通信機能がイネーブルになります。 マルチプロセッサモードではPE、O/E ビットの設定は無効です。
1 0	CKS1 CKS0	0 0	R/W* R/W*	クロックセレクト1、0 内蔵ポーレートジェネレータのクロックソースを選択します。 00 : Pφクロック (n=0) 01 : Pφ/4 (n=1) 10 : Pφ/16 (n=2) 11 : Pφ/64 (n=3) このビットの設定値とポーレートの関係については、「13.3.8 ビットレートレジスタ (BRR)」を参照してください。nは設定値の10進表示で、「13.3.8 ビットレートレジスタ (BRR)」中のnの値を表します。

【注】 * TE=RE=0 の場合のみライト可能です。

13.3.6 シリアルコントロールレジスタ (SCR)

SCR は以下の送受信制御と割り込み制御、送受信クロックソースの選択を行うためのレジスタです。各割り込み要求については「13.7 割り込み要因」を参照してください。

ビット	7	6	5	4	3	2	1	0
ビット名	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*1	R/W*1	R/W	R/W	R/W*2	R/W*2

ビット	ビット名	初期値	R/W	説明
7	TIE	0	R/W	トランスミットインタラプトイネーブル このビットを1にセットすると、TXI 割り込み要求がイネーブルになります。 TXI 割り込み要求の解除は、TDRE フラグから1をリードした後、0にクリアするか、TIE ビットを0にクリアすることで行うことができます。
6	RIE	0	R/W	レシーブインタラプトイネーブル このビットを1にセットすると、RXI および ERI 割り込み要求がイネーブルになります。 RXI および ERI 割り込み要求の解除は、RDRF、または FER、PER、ORER の各フラグから1をリードした後、0にクリアするか、RIE ビットを0にクリアすることで行うことができます。
5	TE	0	R/W*1	トランスミットイネーブル このビットを1にセットすると、送信動作が可能になります。この状態で、TDR に送信データをライトして、SSR の TDRE フラグを0にクリアするとシリアル送信を開始します。なお、TE ビットを1にセットする前に必ず SMR の設定を行い、送信フォーマットを決定してください。 このビットを0にして、送信動作を禁止すると、SSR の TDRE フラグは1に固定されます。
4	RE	0	R/W*1	レシーブイネーブル このビットを1にセットすると、受信動作が可能になります。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受信を開始します。なお、RE ビットを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。 このビットを0にして、受信動作を禁止しても RDRF、または FER、PER、ORER の各フラグは影響を受けず、状態を保持します。

ビット	ビット名	初期値	R/W	説明
3	MPIE	0	R/W	<p>マルチプロセッサインタラプトイネーブル (調歩同期式モードで SMR の MP=1 のとき有効)</p> <p>このビットを 1 にセットすると、マルチプロセッサビットが 0 の受信データは読みとばし、SSR の RDRF、FER、ORER の各ステータス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 のデータを受信すると、このビットは自動的にクリアされ通常の受信動作に戻ります。詳細は「13.5 マルチプロセッサ通信機能」を参照してください。</p> <p>SSR の MPB=0 を含む受信データを受信しているときは、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、ORER の各フラグのセットは行いません。MPB=1 を含む受信データを受信すると、SSR の MPB を 1 にセットし、MPIE ビットを自動的に 0 クリアし、RXI、ERI 割り込み要求 (SCR の RIE ビットが 1 にセットされている場合) と FER、ORER フラグのセットが許可されます。マルチプロセッサ通信機能を使用しない場合は、このビットには 0 をライトして使用してください。</p>
2	TEIE	0	R/W	<p>トランスミットエンドインタラプトイネーブル</p> <p>このビットを 1 セットすると TEI 割り込み要求がイネーブルになります。TEI 割り込み要求の解除は、TDRE フラグから 1 をリードした後、0 にクリアして TEND フラグを 0 にクリアするか、TEIE ビットを 0 にクリアすることで行うことができます。</p>
1 0	CKE1 CKE0	0 0	R/W*2 R/W*2	<p>クロックイネーブル 1、0</p> <p>クロックソースおよび SCK 端子の機能を選択します。</p> <p>調歩同期式の場合</p> <p>00 : 内蔵ポーレートジェネレータ (SCK 端子は入出力ポートとして使用できます)</p> <p>01 : 内蔵ポーレートジェネレータ (SCK 端子からビットレートと同じ周波数のクロックを出力します)</p> <p>1X : 外部クロック (SCK 端子からビットレートの 16 倍の周波数のクロックを入力してください。)</p> <p>クロック同期式の場合</p> <p>0X : 内部クロック (SCK 端子はクロック出力端子となります。)</p> <p>1X : 外部クロック (SCK 端子はクロック入力端子となります。)</p>

【記号説明】 X : Don't care

【注】 *1 TE=RE=0 の場合のみ、1 をライト可能です。いったん、TE、RE ビットのいずれかを 1 にセットした後は、TE=RE=0 のライトのみ可能になります。

*2 TE=RE=0 の場合のみライト可能です。

13.3.7 シリアルステータスレジスタ (SSR)

SSR は SCI のステータスフラグと送受信マルチプロセッサビットで構成されます。TDRE、RDRF、ORER、PER、FER はクリアのみ可能です。

ビット	7	6	5	4	3	2	1	0
ビット名	TDRE	RDRF	ORER	FER	PER	TEND	MPB	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。

ビット	ビット名	初期値	R/W	説明
7	TDRE	1	R/(W)*	<p>トランスミットデータレジスタエンプティ</p> <p>TDR 内の送信データの有無を表示します。</p> <p>[セット条件]</p> <ul style="list-style-type: none"> SCR の TE が 0 のとき TDR から TSR にデータが転送され、TDR にデータライトが可能になったとき <p>[クリア条件]</p> <ul style="list-style-type: none"> 1 の状態をリードした後、0 をライトしたとき (割り込みを使用し、CPU によってクリアする場合、0 ライト後必ずフラグをリードしてください。) TXI 割り込み要求により DMAC または DTC で TDR へ送信データを転送したとき
6	RDRF	0	R/(W)*	<p>レシーブデータレジスタフル</p> <p>RDR 内の受信データの有無を表示します。</p> <p>[セット条件]</p> <ul style="list-style-type: none"> 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 <p>[クリア条件]</p> <ul style="list-style-type: none"> 1 の状態をリードした後、0 をライトしたとき (割り込みを使用し、CPU によってクリアする場合、0 ライト後必ずフラグをリードしてください。) RXI 割り込みにより DMAC または DTC で RDR からデータを転送したとき <p>SCR の RE をクリアしても RDRF は影響を受けず状態を保持します。</p> <p>RDRF フラグが 1 にセットされたまま次のデータを受信完了するとオーバランエラーを発生し、受信データが失われますので注意してください。</p>

ビット	ビット名	初期値	R/W	説明
5	ORER	0	R/(W)*	<p>オーバランエラー</p> <p>受信時にオーバランエラーが発生して異常終了したことを表示します。</p> <p>[セット条件]</p> <ul style="list-style-type: none"> RDRF=1 の状態で次のデータを受信したとき RDR ではオーバランエラーが発生する前の受信データを保持し、後から受信したデータが失われます。ORER フラグに 1 がセットされた状態では、以降のシリアル受信を続けることはできません。なお、クロック同期モードでは、シリアル送信も続けることはできません。 <p>[クリア条件]</p> <ul style="list-style-type: none"> 1 の状態をリードした後、0 をライトしたとき (割り込みを使用し、CPU によってクリアする場合、0 ライト後必ずフラグをリードしてください。) <p>SCR の RE をクリアしても、ORER は影響を受けず以前の状態を保持します。</p>
4	FER	0	R/(W)*	<p>フレーミングエラー</p> <p>調歩同期モードで受信時にフレーミングエラーが発生して異常終了したことを表示します。</p> <p>[セット条件]</p> <ul style="list-style-type: none"> ストップビットが 0 のとき 2 ストップモードのときは、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しません。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 フラグはセットされません。さらに、FER フラグが 1 にセットされた状態では、以降のシリアル受信を続けることはできません。クロック同期モードでは、シリアル送信も続けることはできません。 <p>[クリア条件]</p> <ul style="list-style-type: none"> 1 の状態をリードした後、0 をライトしたとき (割り込みを使用し、CPU によってクリアする場合、0 ライト後必ずフラグをリードしてください。) <p>SCR の RE をクリアしても、FER は影響を受けず以前の状態を保持します。</p>

ビット	ビット名	初期値	R/W	説明
3	PER	0	R/(W)*	<p>パリティエラー</p> <p>調歩同期式モードで受信時にパリティエラーが発生して異常終了したことを表示します。</p> <p>[セット条件]</p> <ul style="list-style-type: none"> 受信中にパリティエラーを検出したとき <p>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 フラグはセットされません。なお、PER フラグが 1 にセットされた状態では、以降のシリアル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、シリアル送信も続けることはできません。</p> <p>[クリア条件]</p> <ul style="list-style-type: none"> 1 の状態をリードした後、0 をライトしたとき <p>(割り込みを使用し、CPU によってクリアする場合、0 ライト後必ずフラグをリードしてください。)</p> <p>SCR の RE をクリアしても、PER は影響を受けず以前の状態を保持します。</p>
2	TEND	1	R	<p>トランスミットエンド</p> <p>[セット条件]</p> <ul style="list-style-type: none"> SCR の TE が 0 のとき 送信キャラクタの最後尾ビットの送信時、TDRE が 1 のとき <p>[クリア条件]</p> <ul style="list-style-type: none"> TDRE=1 の状態をリードした後、TDRE フラグに 0 をライトしたとき TXI 割り込み要求により DMAC または DTC で TDR へ送信データをライトしたとき
1	MPB	0	R	<p>マルチプロセッサビット</p> <p>受信フレーム中のマルチプロセッサビットの値が格納されます。SCR の RE が 0 のときは変化しません。</p>
0	MPBT	0	R/W	<p>マルチプロセッサビットトランスファ</p> <p>送信フレームに付加するマルチプロセッサビットの値を設定します。</p>

【注】 * フラグをクリアするための 0 ライトのみ可能です。

13.3.8 ビットレートレジスタ (BRR)

BRR はビットレートを調整するための 8 ビットのレジスタです。SCI はチャンネルごとにボーレートジェネレータが独立しているため、異なるビットレートを設定できます。通常の調歩同期式モード、クロック同期式モードにおける BRR の設定値 N とビットレート B の関係を表 13.2 に示します。BRR の初期値は H'FF です。BRR は、CPU からのリードは常に可能ですが、ライトは TE=RE=0 の場合のみ可能です。

表 13.2 BRR の設定値 N とビットレート B の関係

モード	ビットレート	誤差
調歩同期式	$B = \frac{\phi \times 10^6}{64 \times 2^{2n-1} \times (N+1)}$	誤差 (%) = $\left\{ \frac{\phi \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
クロック同期式	$B = \frac{\phi \times 10^6}{8 \times 2^{2n-1} \times (N+1)}$	

【注】 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 ($0 \leq N \leq 255$)

ϕ : 動作周波数 (MHz)

n : 下表のとおり SMR の設定値によって決まります。

SMR の設定値		n
CKS1	CKS0	
0	0	0
0	1	1
1	0	2
1	1	3

通常の調歩同期式モードにおける BRR の値 N の設定例を表 13.3 に、各動作周波数における設定可能な最大ビットレートを表 13.4 に示します。また、クロック同期式モードにおける BRR の値 N の設定例を表 13.6 に示します。また、表 13.5、表 13.7 に外部クロック入力時の最大ビットレートを示します。

表 13.3 ビットレートに対する BRR の設定例 [調歩同期式モード] (1)

ビットレート (bit/s)	動作周波数 ϕ (MHz)											
	9.8304			10			12			12.288		
	n	N	誤差 (%)	n	N	誤差 (%)	n	N	誤差 (%)	n	N	誤差 (%)
110	2	174	-0.26	2	177	-0.25	2	212	0.03	2	217	0.08
150	2	127	0.00	2	129	0.16	2	155	0.16	2	159	0.00
300	1	255	0.00	2	64	0.16	2	77	0.16	2	79	0.00
600	1	127	0.00	1	129	0.16	1	155	0.16	1	159	0.00
1200	0	255	0.00	1	64	0.16	1	77	0.16	1	79	0.00
2400	0	127	0.00	0	129	0.16	0	155	0.16	0	159	0.00
4800	0	63	0.00	0	64	0.16	0	77	0.16	0	79	0.00
9600	0	31	0.00	0	32	-1.36	0	38	0.16	0	39	0.00
19200	0	15	0.00	0	15	1.73	0	19	-2.34	0	19	0.00
31250	0	9	-1.70	0	9	0.00	0	11	0.00	0	11	2.40
38400	0	7	0.00	0	7	1.73	0	9	-2.34	0	9	0.00

ビットレート (bit/s)	動作周波数 ϕ (MHz)											
	14			14.7456			16			17.2032		
	n	N	誤差 (%)	n	N	誤差 (%)	n	N	誤差 (%)	n	N	誤差 (%)
110	2	248	-0.17	3	64	0.70	3	70	0.03	3	75	0.48
150	2	181	0.16	2	191	0.00	2	207	0.16	2	223	0.00
300	2	90	0.16	2	95	0.00	2	103	0.16	2	111	0.00
600	1	181	0.16	1	191	0.00	1	207	0.16	1	223	0.00
1200	1	90	0.16	1	95	0.00	1	103	0.16	1	111	0.00
2400	0	181	0.16	0	191	0.00	0	207	0.16	0	223	0.00
4800	0	90	0.16	0	95	0.00	0	103	0.16	0	111	0.00
9600	0	45	-0.93	0	47	0.00	0	51	0.16	0	55	0.00
19200	0	22	-0.93	0	23	0.00	0	25	0.16	0	27	0.00
31250	0	13	0.00	0	14	-1.70	0	15	0.00	0	16	1.20
38400	-	-	-	0	11	0.00	0	12	0.16	0	13	0.00

表 13.3 ビットレートに対する BRR の設定例 [調歩同期式モード] (2)

ビットレート (bit/s)	動作周波数 ϕ (MHz)											
	18			19.6608			20			25		
	n	N	誤差 (%)	n	N	誤差 (%)	n	N	誤差 (%)	n	N	誤差 (%)
110	3	79	-0.12	3	86	0.31	3	88	-0.25	3	110	-0.02
150	2	233	0.16	2	255	0.00	3	64	0.16	3	80	-0.47
300	2	116	0.16	2	127	0.00	2	129	0.16	2	162	0.15
600	1	233	0.16	1	255	0.00	2	64	0.16	2	80	-0.47
1200	1	116	0.16	1	127	0.00	1	129	0.16	1	162	0.15
2400	0	233	0.16	0	255	0.00	1	64	0.16	1	80	-0.47
4800	0	116	0.16	0	127	0.00	0	129	0.16	0	162	0.15
9600	0	58	-0.69	0	63	0.00	0	64	0.16	0	80	-0.47
19200	0	28	1.02	0	31	0.00	0	32	-1.36	0	40	-0.76
31250	0	17	0.00	0	19	-1.70	0	19	0.00	0	24	0.00
38400	0	14	-2.34	0	15	0.00	0	15	1.73	0	19	1.73

ビットレート (bit/s)	動作周波数 ϕ (MHz)					
	30			33		
	n	N	誤差 (%)	n	N	誤差 (%)
110	3	132	0.13	3	145	0.33
150	3	97	-0.35	3	106	0.39
300	2	194	0.16	2	214	-0.07
600	2	97	-0.35	2	106	0.39
1200	1	194	0.16	1	214	-0.07
2400	1	97	-0.35	1	106	0.39
4800	0	194	0.16	0	214	-0.07
9600	0	97	-0.35	0	106	0.39
19200	0	48	-0.35	0	53	-0.54
31250	0	29	0	0	32	0
38400	0	23	1.73	0	26	-0.54

表 13.4 各動作周波数における最大ビットレート (調歩同期式モード)

Pφ (MHz)	最大ビットレート (bit/s)	n	N	Pφ (MHz)	最大ビットレート (bit/s)	n	N
8	250000	0	0	17.2032	537600	0	0
9.8304	307200	0	0	18	562500	0	0
10	312500	0	0	19.6608	614400	0	0
12	375000	0	0	20	625000	0	0
12.288	384000	0	0	25	781250	0	0
14	437500	0	0	30	937500	0	0
14.7456	460800	0	0	33	1031250	0	0
16	500000	0	0	35	1093750	0	0

表 13.5 外部クロック入力時の最大ビットレート (調歩同期式モード)

Pφ (MHz)	外部入力クロック (MHz)	最大ビットレート (bit/s)	Pφ (MHz)	外部入力クロック (MHz)	最大ビットレート (bit/s)
8	2.0000	125000	17.2032	4.3008	268800
9.8304	2.4576	153600	18	4.5000	281250
10	2.5000	156250	19.6608	4.9152	307200
12	3.0000	187500	20	5.0000	312500
12.288	3.0720	192000	25	6.2500	390625
14	3.5000	218750	30	7.5000	468750
14.7456	3.6864	230400	33	8.2500	515625
16	4.0000	250000	35	8.7500	546875

表 13.6 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]

ビット レート (bit/s)	動作周波数 Pφ (MHz)															
	8		10		16		20		25		30		33		35	
	n	N	n	N	n	N	n	N	n	N	n	N	n	N	n	N
110																
250	3	124	—	—	3	249										
500	2	249	—	—	3	124	—	—			3	233				
1k	2	124	—	—	2	249	—	—	3	97	3	116	3	128	3	136
2.5k	1	199	1	249	2	99	2	124	2	155	2	187	2	205	2	218
5k	1	99	1	124	1	199	1	249	2	77	2	93	2	102	2	108
10k	0	199	0	249	1	99	1	124	1	155	1	187	1	205	1	218
25k	0	79	0	99	0	159	0	199	0	249	1	74	1	82	1	87
50k	0	39	0	49	0	79	0	99	0	124	0	149	0	164	0	174
100k	0	19	0	24	0	39	0	49	0	62	0	74	0	82	0	87
250k	0	7	0	9	0	15	0	19	0	24	0	29	0	32	0	34
500k	0	3	0	4	0	7	0	9	—	—	0	14	—	—	—	—
1M	0	1			0	3	0	4	—	—	—	—	—	—	—	—
2.5M			0	0*			0	1	—	—	0	2	—	—	—	—
5M							0	0*	—	—	—	—	—	—	—	—

【記号説明】

空欄 : 設定できません。

— : 設定可能ですが誤差がでます。

* : 連続送信/連続受信はできません。

表 13.7 外部クロック入力時の最大ビットレート (クロック同期式モード)

Pφ (MHz)	外部入力クロック (MHz)	最大ビットレート (bit/s)	Pφ (MHz)	外部入力クロック (MHz)	最大ビットレート (bit/s)
8	1.3333	1333333.3	20	3.3333	3333333.3
10	1.6667	1666666.7	25	4.1667	4166666.7
12	2.0000	2000000.0	30	5.0000	5000000.0
14	2.3333	2333333.3	33	5.5000	5500000.0
16	2.6667	2666666.7	35	5.8336	5833625.0
18	3.0000	3000000.0			

13.4 調歩同期式モードの動作

調歩同期式シリアル通信の一般的なフォーマットを図 13.2 に示します。1 フレームは、スタートビット (Low レベル) から始まり送受信データ、パリティビット、ストップビット (High レベル) の順で構成されます。調歩同期式シリアル通信では、通信回線は通常マーク状態 (High レベル) に保たれています。SCI は通信回線を監視し、スペース (Low レベル) を検出するとスタートビットとみなしてシリアル通信を開始します。SCI 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、送信および受信中にデータのリード/ライトができ、連続送受信が可能です。

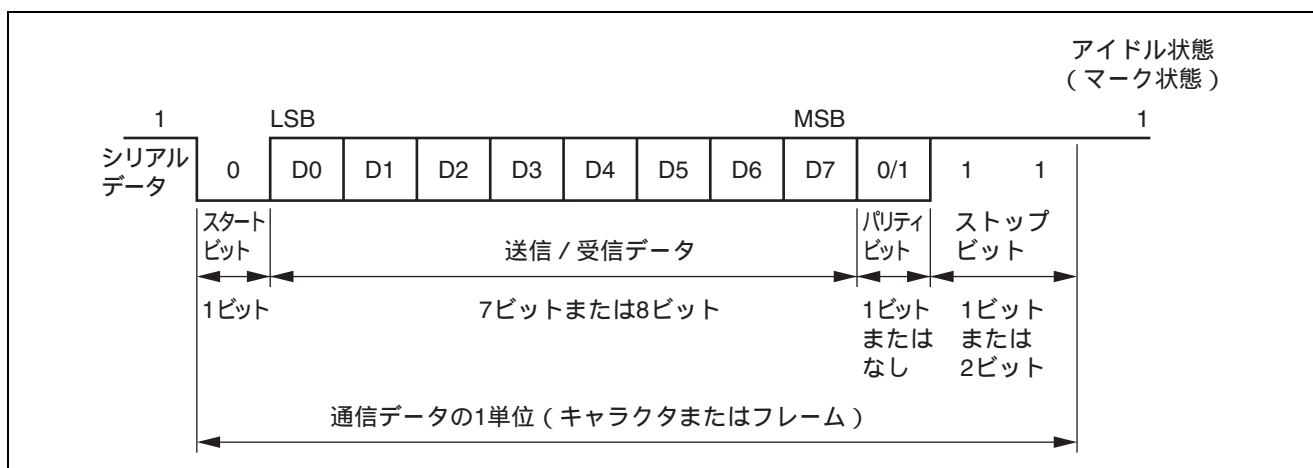


図 13.2 調歩同式通信のデータフォーマット (8 ビットデータ/パリティあり/2 ストップビットの例)

13.4.1 送受信フォーマット

調歩同期式モードで設定できる送受信フォーマットを、表 13.8 に示します。フォーマットは 12 種類あり、SMR の選定により選択できます。マルチプロセッサビットについては「13.5 マルチプロセッサ通信機能」を参照してください。

表 13.8 シリアル送信／受信フォーマット（調歩同期式モード）

SMRの設定				シリアル送信／受信フォーマットとフレーム長												
CHR	PE	MP	STOP	1	2	3	4	5	6	7	8	9	10	11	12	
0	0	0	0	S	8ビットデータ								STOP			
0	0	0	1	S	8ビットデータ								STOP	STOP		
0	1	0	0	S	8ビットデータ								P	STOP		
0	1	0	1	S	8ビットデータ								P	STOP	STOP	
1	0	0	0	S	7ビットデータ							STOP				
1	0	0	1	S	7ビットデータ							STOP	STOP			
1	1	0	0	S	7ビットデータ							P	STOP			
1	1	0	1	S	7ビットデータ							P	STOP	STOP		
0	—	1	0	S	8ビットデータ								MPB	STOP		
0	—	1	1	S	8ビットデータ								MPB	STOP	STOP	
1	—	1	0	S	7ビットデータ							MPB	STOP			
1	—	1	1	S	7ビットデータ							MPB	STOP	STOP		

【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

13.4.2 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、SCIはビットレートの16倍の周波数の基本クロックで動作します。受信時はスタートビットの立ち下がり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、図13.3に示すように受信データを基本クロックの8つ目の立ち上がりエッジでサンプリングすることで、各ビットの中央でデータを取り込みます。

したがって、調歩同期式モードでの受信マージンは式(1)のように表わすことができます。

$$M = \left\{ \left(0.5 - \frac{1}{2N} \right) - (L - 0.5) F - \frac{|D - 0.5|}{N} (1 + F) \right\} \times 100 \quad [\%] \quad \cdots \text{式 (1)}$$

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、

$$M = \left\{ 0.5 - \frac{1}{2 \times 16} \right\} \times 100 \quad [\%] = 46.875\%$$

となります。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。

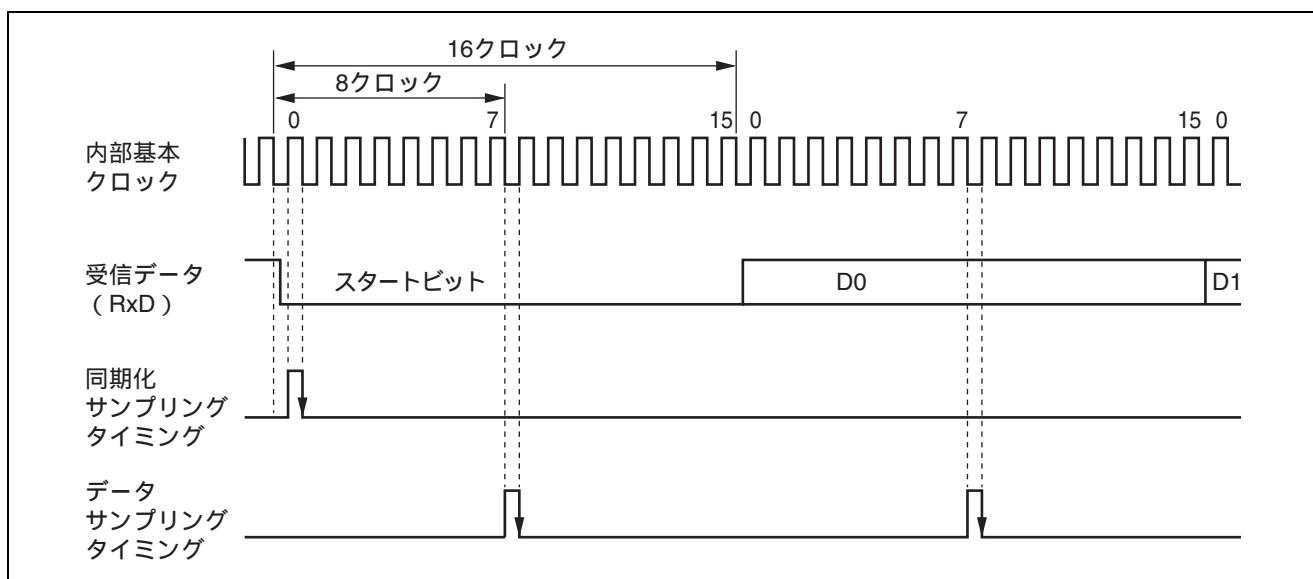


図 13.3 調歩同期式モードの受信データサンプリングタイミング

13.4.3 クロック

SCI の送受信クロックは、SMR の C/A ビットと SCR の CKE1、CKE0 ビットの設定により、内蔵ボーレートジェネレータの生成する内部クロックまたは SCK 端子から入力される外部クロックのいずれかを選択できます。外部クロックを使用する場合は、SCK 端子にビットレートの 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は SCK 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、送信時の位相は図 13.4 に示すように送信データの中央でクロックが立ち上がります。

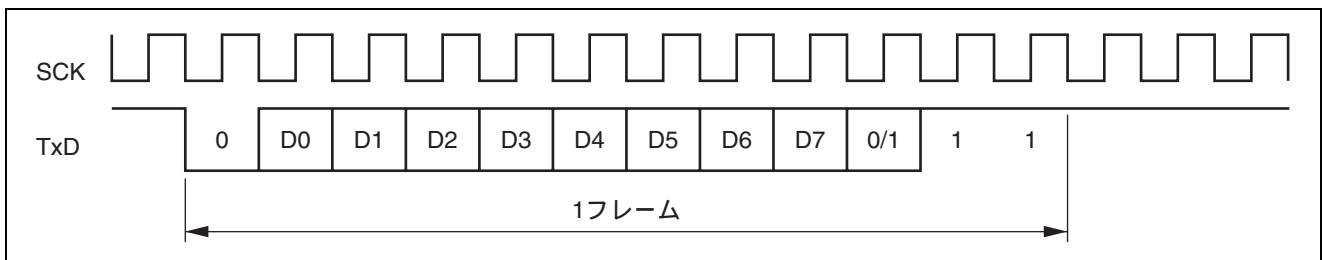


図 13.4 出力クロックと送信データの位相関係（調歩同期式モード）

13.4.4 SCI の初期化 (調歩同期式)

データの送受信前に、SCR の TE、RE ビットをクリアした後、図 13.5 のフローチャートの例に従って初期化してください。動作モードの変更、通信フォーマットの変更などの場合も必ず、TE ビットおよび RE ビットをいずれも 0 にクリアしてから変更を行ってください。TE を 0 にクリアすると、TDRE は 1 にセットされますが、RE を 0 にクリアしても、RDRF、PER、FER、ORER の各フラグ、および RDR は初期化されません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する場合は、初期化の期間も含めてクロックを供給してください。

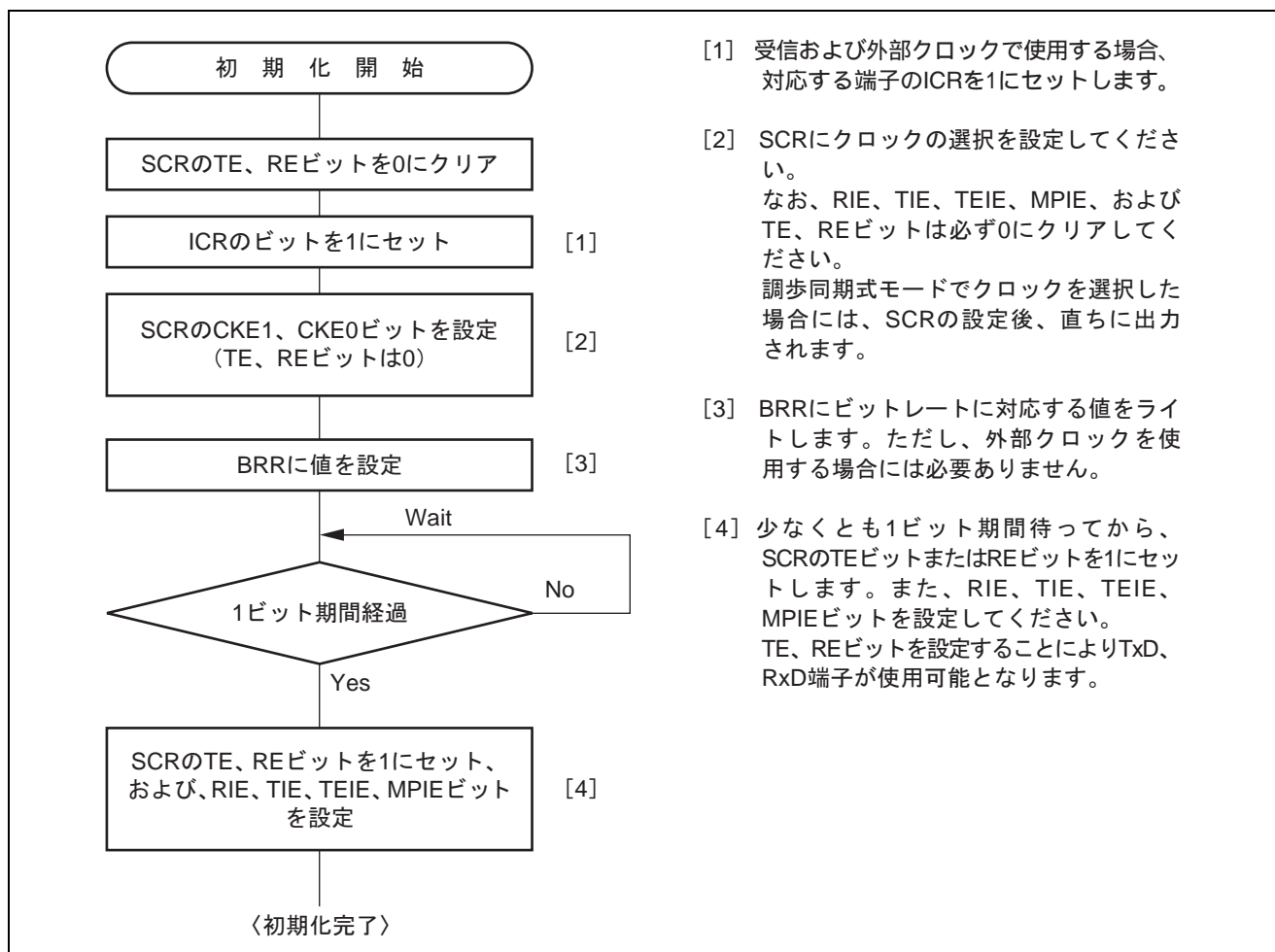


図 13.5 SCI の初期化フローチャートの例

13.4.5 シリアルデータ送信 (調歩同期式)

図 13.6 に調歩同期式モードの送信時の動作例を示します。データ送信時 SCI は以下のように動作します。

1. SCIはSSRのTDREを監視し、クリアされるとTDRにデータが書き込まれたと認識してTDRからTSRにデータを転送します。
2. TDRからTSRにデータを転送すると、TDREを1にセットして送信を開始します。このとき、SCRのTIEが1にセットされているとTXI割り込み要求を発生します。このTXI割り込みルーチンで、前に転送したデータの送信が終了するまでにTDRに次の送信データを書き込むことで連続送信が可能です。
3. TxD端子からスタートビット、送信データ、パリティビットまたはマルチプロセッサビット (フォーマットによってはない場合もあります)、ストップビットの順に送り出します。
4. 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。
5. TDREが0であると次の送信データを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。
6. TDREが1であるとSSRのTENDを1をセットし、ストップビット送出後、1を出力してマーク状態になります。このときSCRのTEIEが1にセットされているとTEIを発生します。

図 13.7 にデータ送信のフローチャートの例を示します。

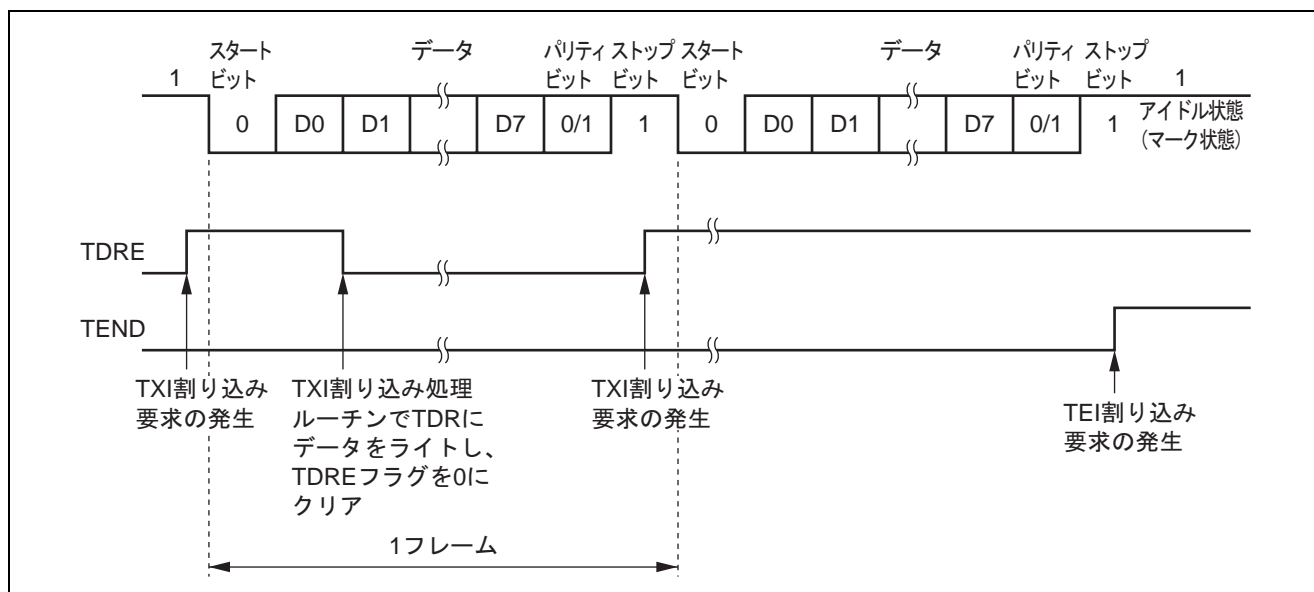


図 13.6 調歩同期式モードの送信時の動作例 (8 ビットデータ/パリティあり/1 ストップビットの例)

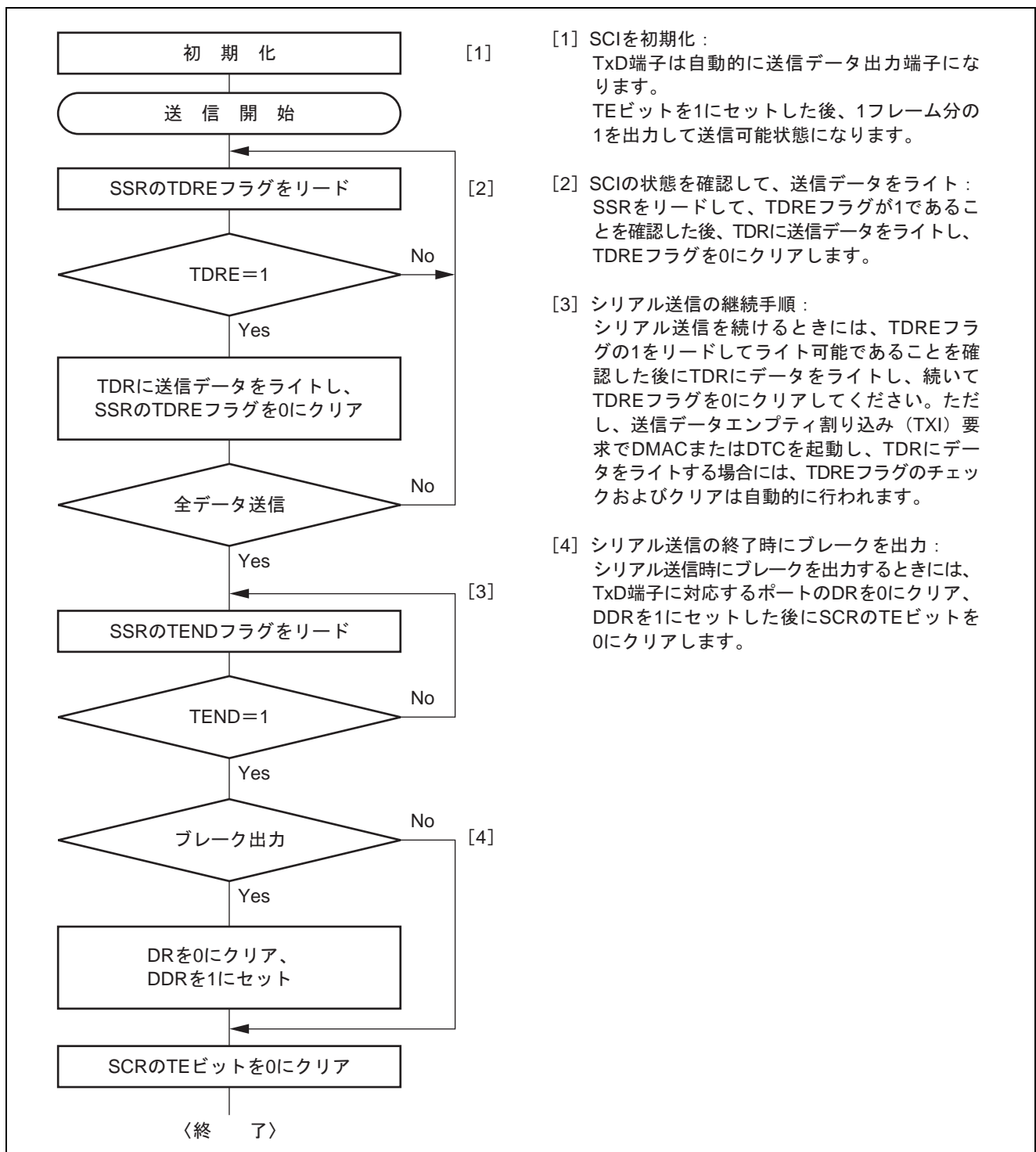


図 13.7 シリアル送信のフローチャートの例

13.4.6 シリアルデータ受信 (調歩同期式)

図 13.8 に調歩同期式モードの受信時の動作例を示します。データ受信時 SCI は以下のように動作します。

1. 通信回線を監視し、スタートビットを検出すると内部を同期化して受信データをRDRに取り込み、パリティビットとストップビットをチェックします。
2. オーバランエラーが発生したとき (SSRのRDRFが1にセットされたまま次のデータを受信完了したとき) はSSRのORERをセットします。このときSCRのRIEが1にセットされているとERI割り込み要求を発生します。受信データはRDRに転送しません。RDRFは1にセットされた状態を保持します。
3. パリティエラーを検出した場合はSSRのPERをセットし、受信データをRDRに転送します。このときSCRのRIEが1にセットされているとERI割り込み要求を発生します。
4. フレーミングエラー (ストップビットが0のとき) を検出した場合はSSRのFERをセットし、受信データをRDRに転送します。このときSCRのRIEが1にセットされているとERI割り込み要求を発生します。
5. 正常に受信したときはSSRのRDRFをセットし、受信データをRDRに転送します。このときSCRのRIEが1にセットされているとRXI割り込み要求を発生します。このRXI割り込み処理ルーチンでRDRに転送された受信データを次のデータ受信完了までにリードすることで連続受信が可能です。

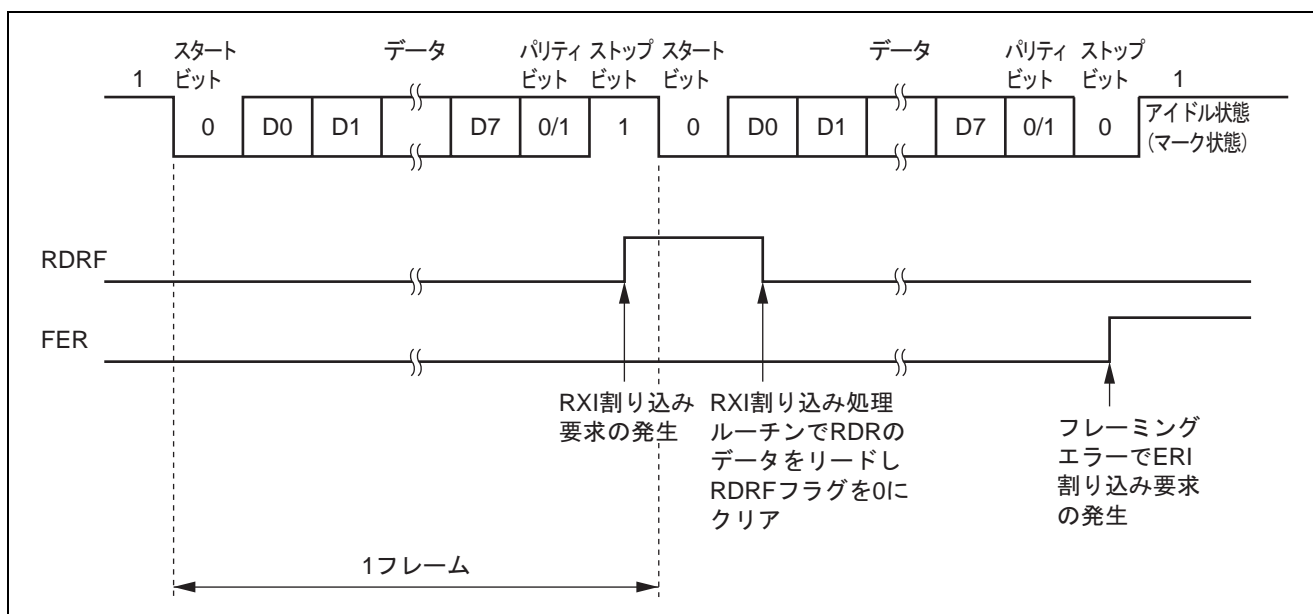


図 13.8 SCI の受信時の動作例 (8 ビットデータ/パリティあり/1 ストップビットの例)

受信エラーを検出した場合の SSR の各ステータスフラグの状態と受信データの処理を表 13.9 に示します。受信エラーを検出すると、RDRF はデータを受信する前の状態を保ちます。受信エラーフラグ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 ORER、FER、PER、および RDRF を 0 にクリアしてください。図 13.9 にデータ受信のためのフローチャートの例を示します。

表 13.9 SSR のステータスフラグの状態と受信データの処理

SSR のステータスフラグ				受信データ	受信エラーの状態
RDRF*	ORER	FER	PER		
1	1	0	0	消失	オーバランエラー
0	0	1	0	RDR へ転送	フレーミングエラー
0	0	0	1	RDR へ転送	パリティエラー
1	1	1	0	消失	オーバランエラー+フレーミングエラー
1	1	0	1	消失	オーバランエラー+パリティエラー
0	0	1	1	RDR へ転送	フレーミングエラー+パリティエラー
1	1	1	1	消失	オーバランエラー+フレーミングエラー+パリティエラー

【注】 * RDRF は、データ受信前の状態を保持します。

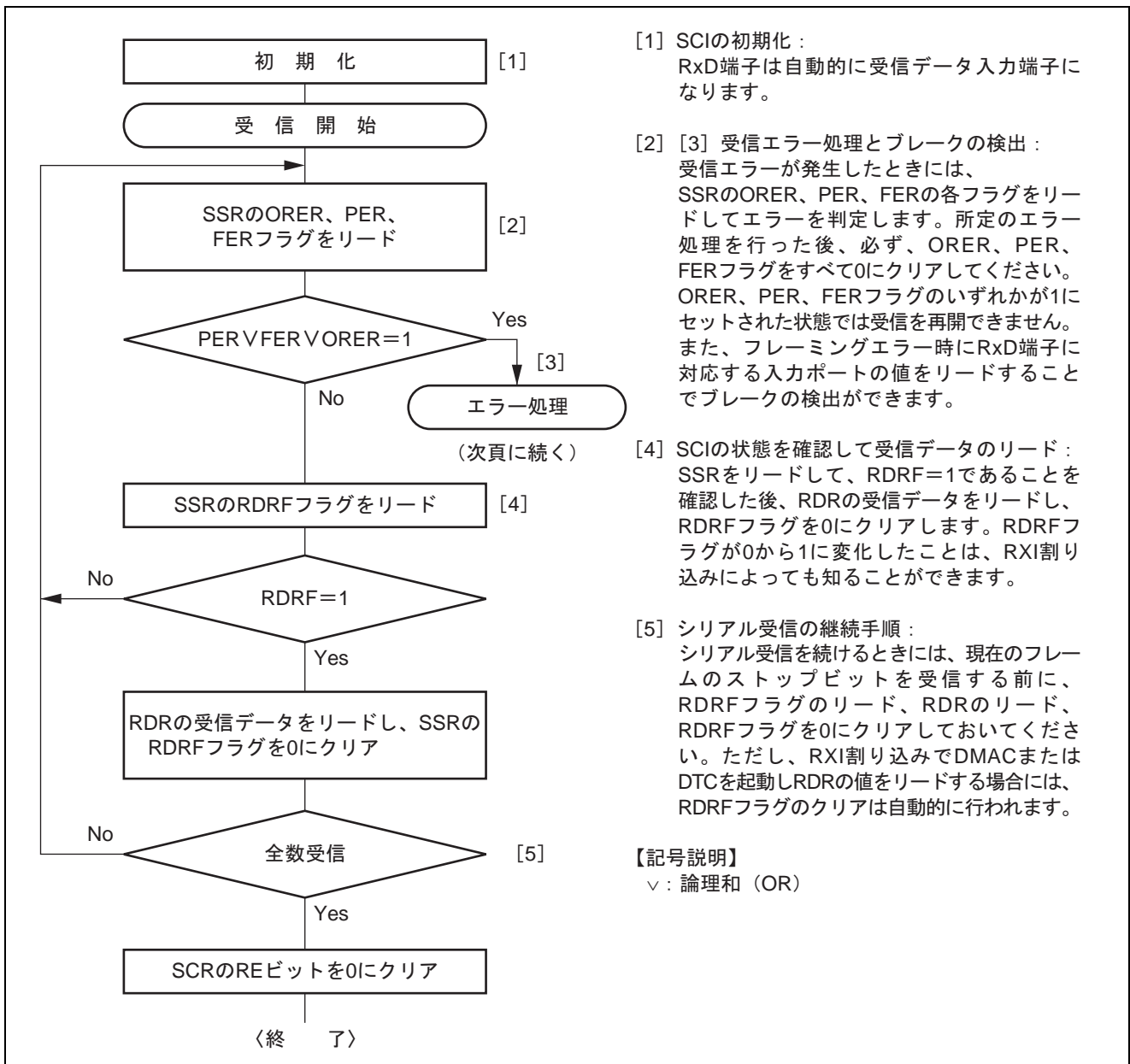


図 13.9 シリアル受信データフローチャートの例 (1)

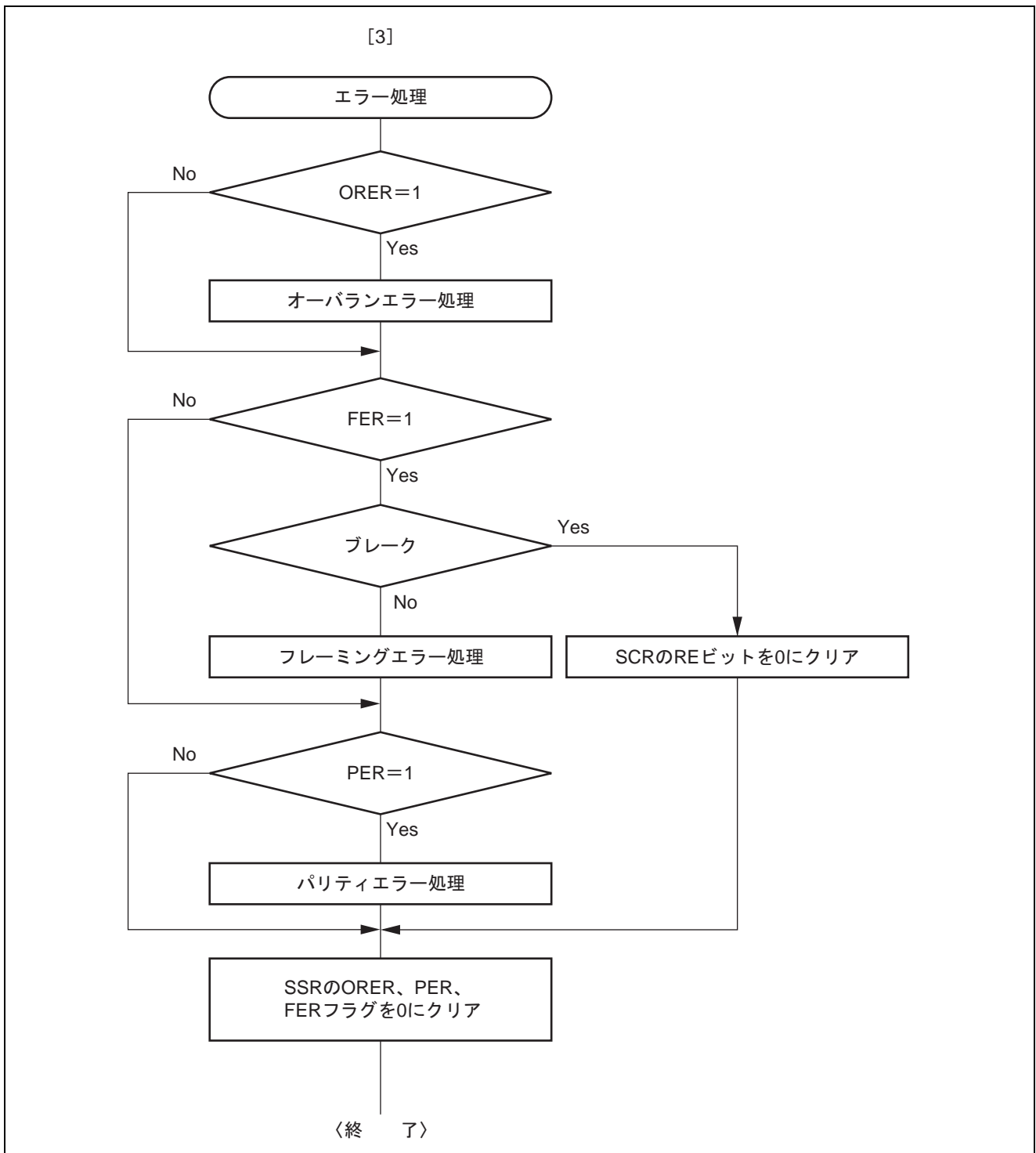


図 13.9 シリアル受信データフローチャートの例 (2)

13.5 マルチプロセッサ通信機能

マルチプロセッサ通信機能を使用すると、マルチプロセッサビットを付加した調歩同期式シリアル通信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ことができます。マルチプロセッサ通信では受信局に各々固有の ID コードを割り付けます。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に対するデータ送信サイクルで構成されます。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。図 13.10 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。送信局は、まず受信局の 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 の通信データを受信するまで通信データを読み飛ばします。

SCIはこの機能をサポートするため、SCRにMPIEビットが設けてあります。MPIEを1にセットすると、マルチプロセッサビットが1のデータを受け取るまで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、ORERの各ステータス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MPBが1にセットされるとともにMPIEが自動的にクリアされて通常の受信動作に戻ります。このときSCRのRIEがセットされているとRXI割り込みを発生します。MPIEが0にクリアされた状態では、マルチプロセッサビットの値に関係なく受信動作を行います。マルチプロセッサビットは、SSRのMPBに格納されます。

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。それ以外は通常の調歩同期式モードと変わりません。マルチプロセッサ通信を行うときのクロックも通常の調歩同期式モードと同一です。

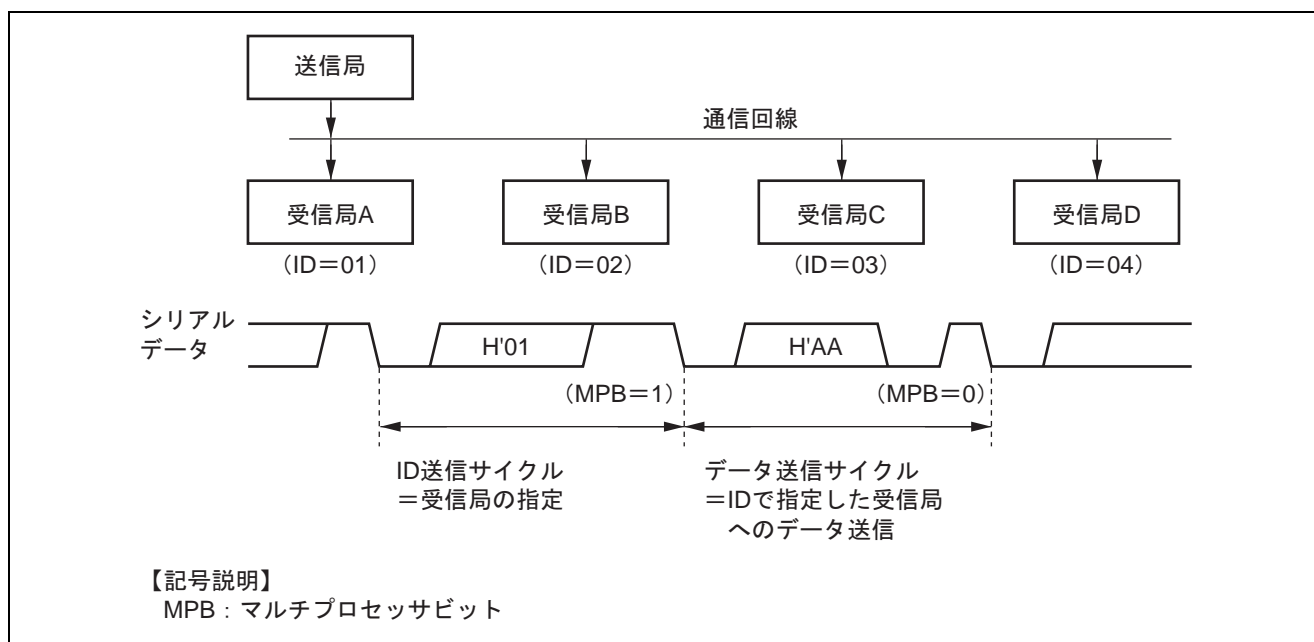


図 13.10 マルチプロセッサフォーマットを使用した通信例（受信局 A へのデータ H'AA の送金の例）

13.5.1 マルチプロセッサシリアルデータ送信

図 13.11 にマルチプロセッサデータ処理のフローチャートの例を示します。ID 送信サイクルでは SSR の MPBT を 1 にセットして送信してください。データ送信サイクルでは SSR の MPBT を 0 にクリアして送信してください。その他の動作は調歩同期式モードの動作と同じです。

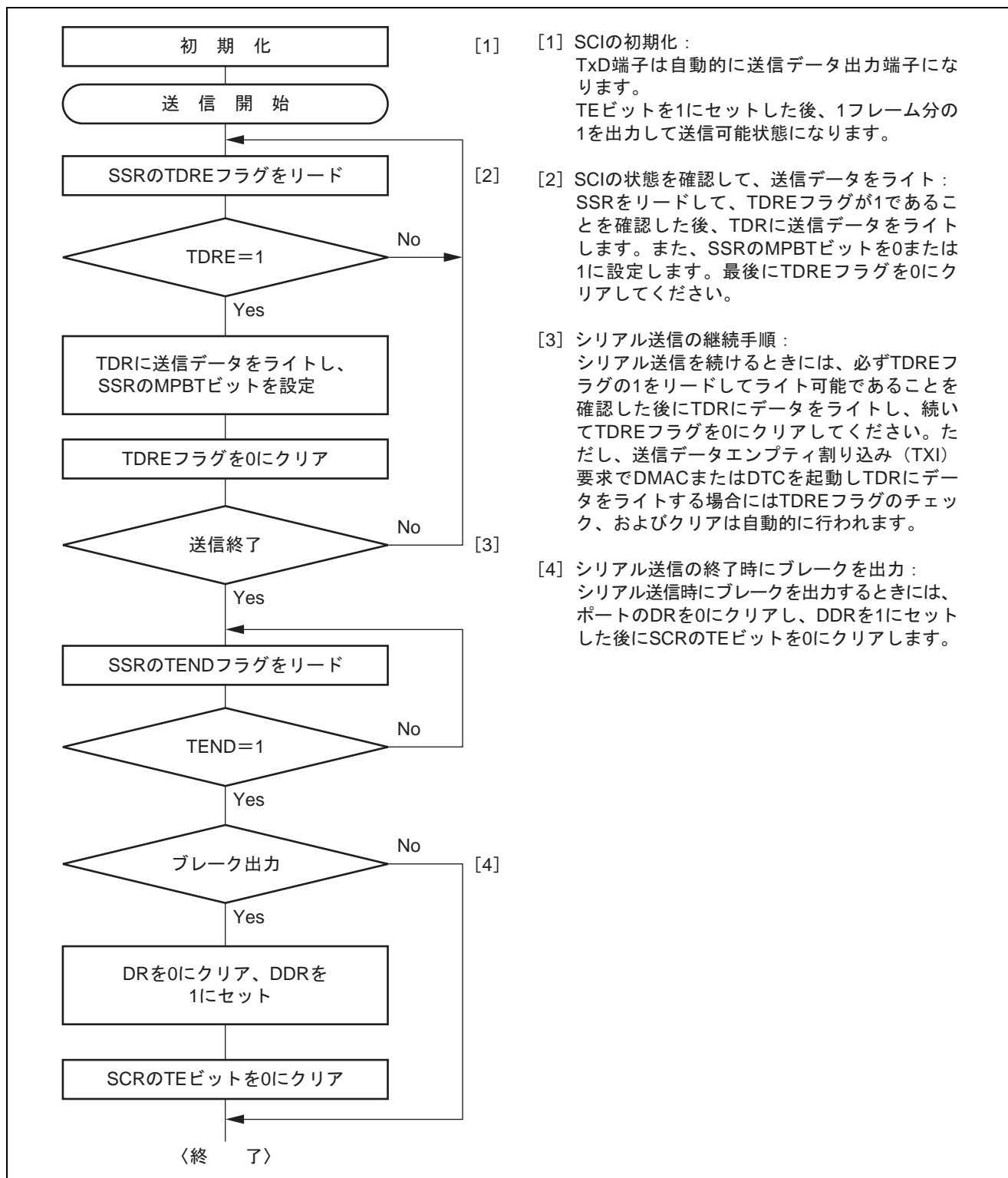


図 13.11 マルチプロセッサシリアル送信のフローチャートの例

13.5.2 マルチプロセッサシリアルデータ受信

図 13.13 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。SCR の MPIE を 1 にセットするとマルチプロセッサビットが 1 の通信データを受信するまで通信データを読みとばします。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受信データを RDR に転送します。このとき RXI 割り込み要求を発生します。その他の動作は調歩同期式モードの動作と同じです。図 13.12 に受信時の動作例を示します。

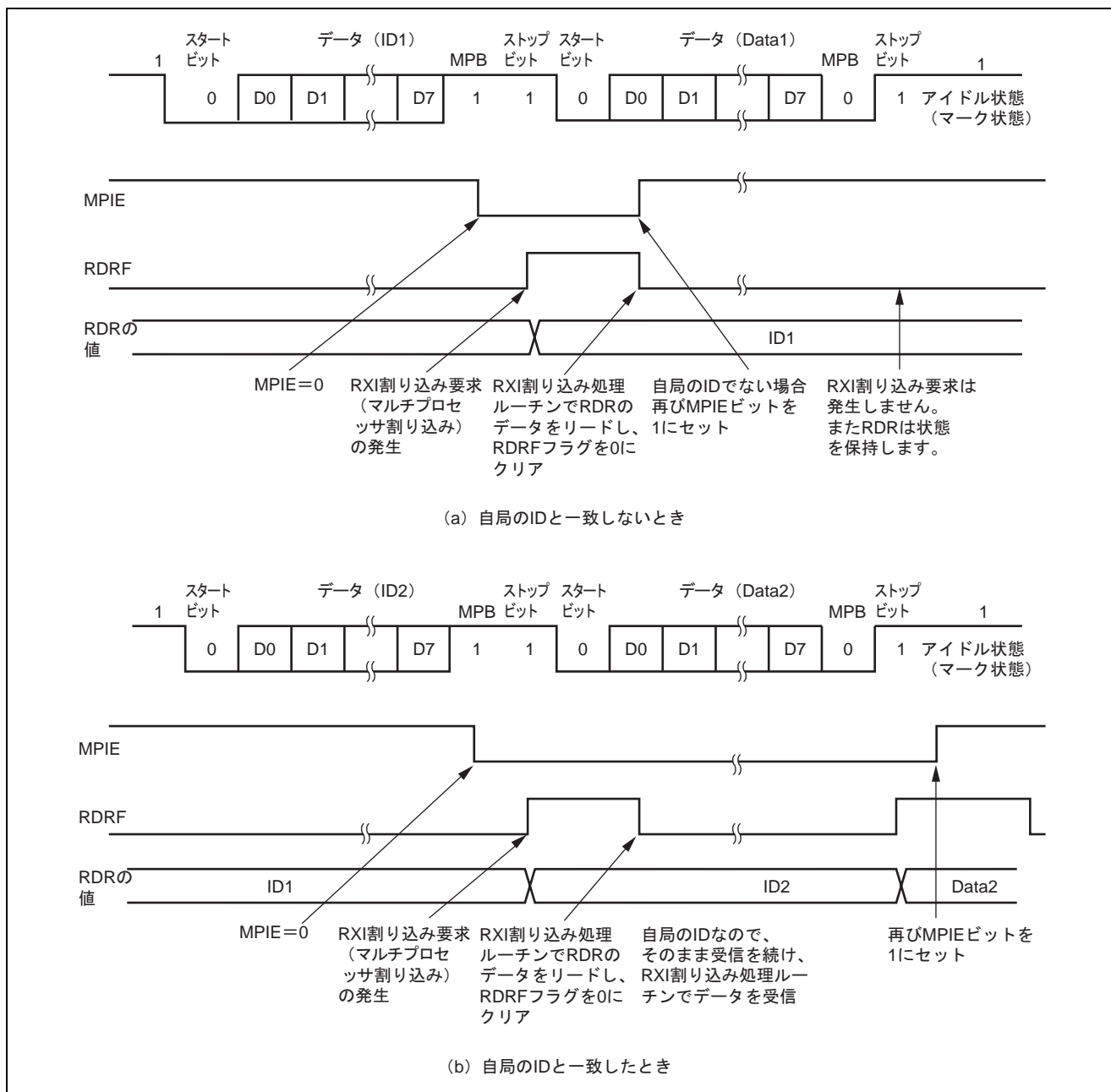
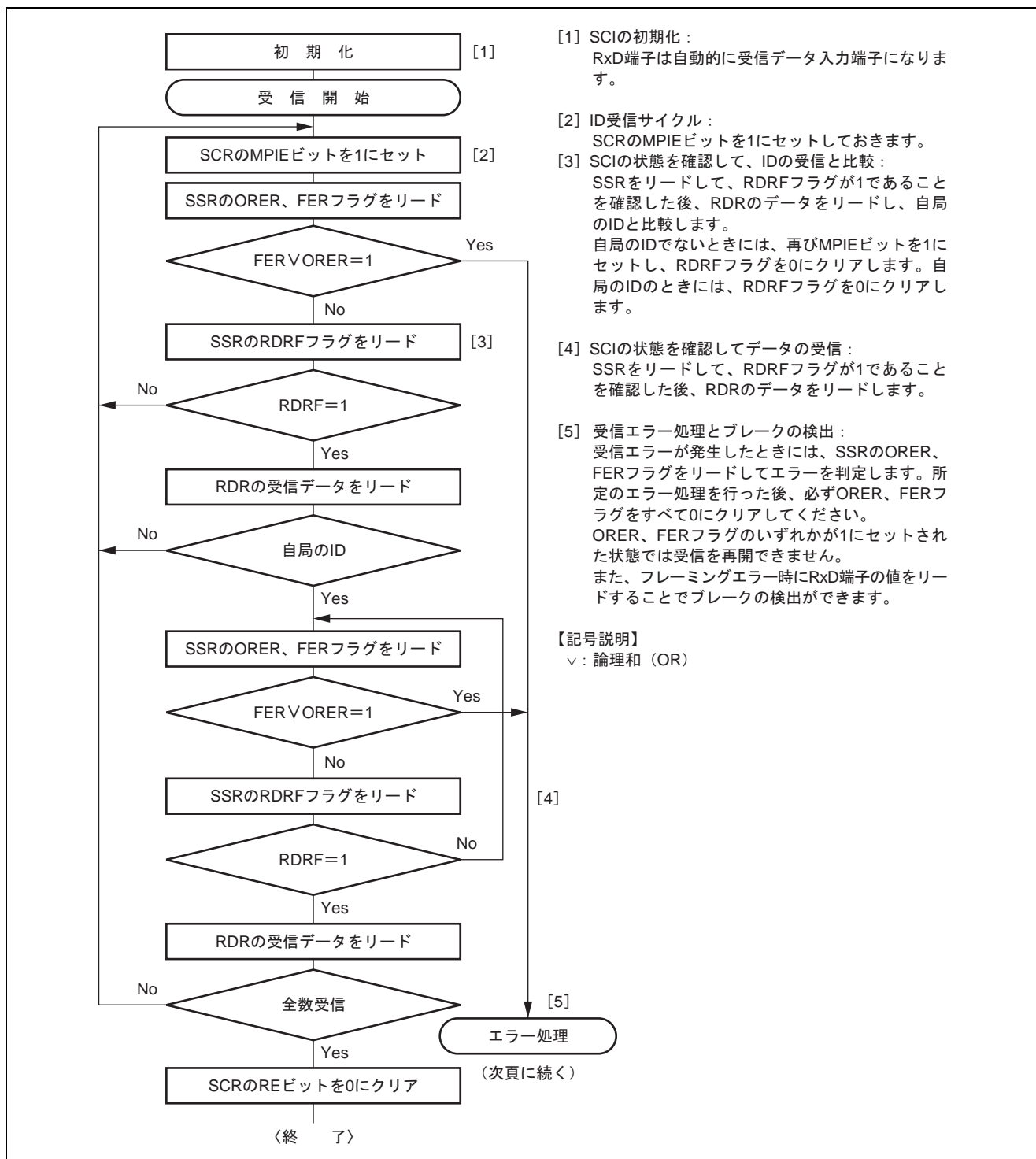


図 13.12 SCI の受信時の動作例 (8 ビットデータ/マルチプロセッサビットあり/1 ストップビットの例)



- [1] SCIの初期化：
RxD端子は自動的に受信データ入力端子になります。
- [2] ID受信サイクル：
SCRのMPIEビットを1にセットしておきます。
- [3] SCIの状態を確認して、IDの受信と比較：
SSRをリードして、RDRFフラグが1であることを確認した後、RDRのデータをリードし、自局のIDと比較します。
自局のIDでないときには、再びMPIEビットを1にセットし、RDRFフラグを0にクリアします。自局のIDのときには、RDRFフラグを0にクリアします。
- [4] SCIの状態を確認してデータの受信：
SSRをリードして、RDRFフラグが1であることを確認した後、RDRのデータをリードします。
- [5] 受信エラー処理とブレークの検出：
受信エラーが発生したときには、SSRのORER、FERフラグをリードしてエラーを判定します。所定のエラー処理を行った後、必ずORER、FERフラグをすべて0にクリアしてください。
ORER、FERフラグのいずれかが1にセットされた状態では受信を再開できません。
また、フレーミングエラー時にRxD端子の値をリードすることでブレークの検出ができます。

【記号説明】
∨：論理和 (OR)

図 13.13 マルチプロセッサシリアル受信のフローチャートの例 (1)

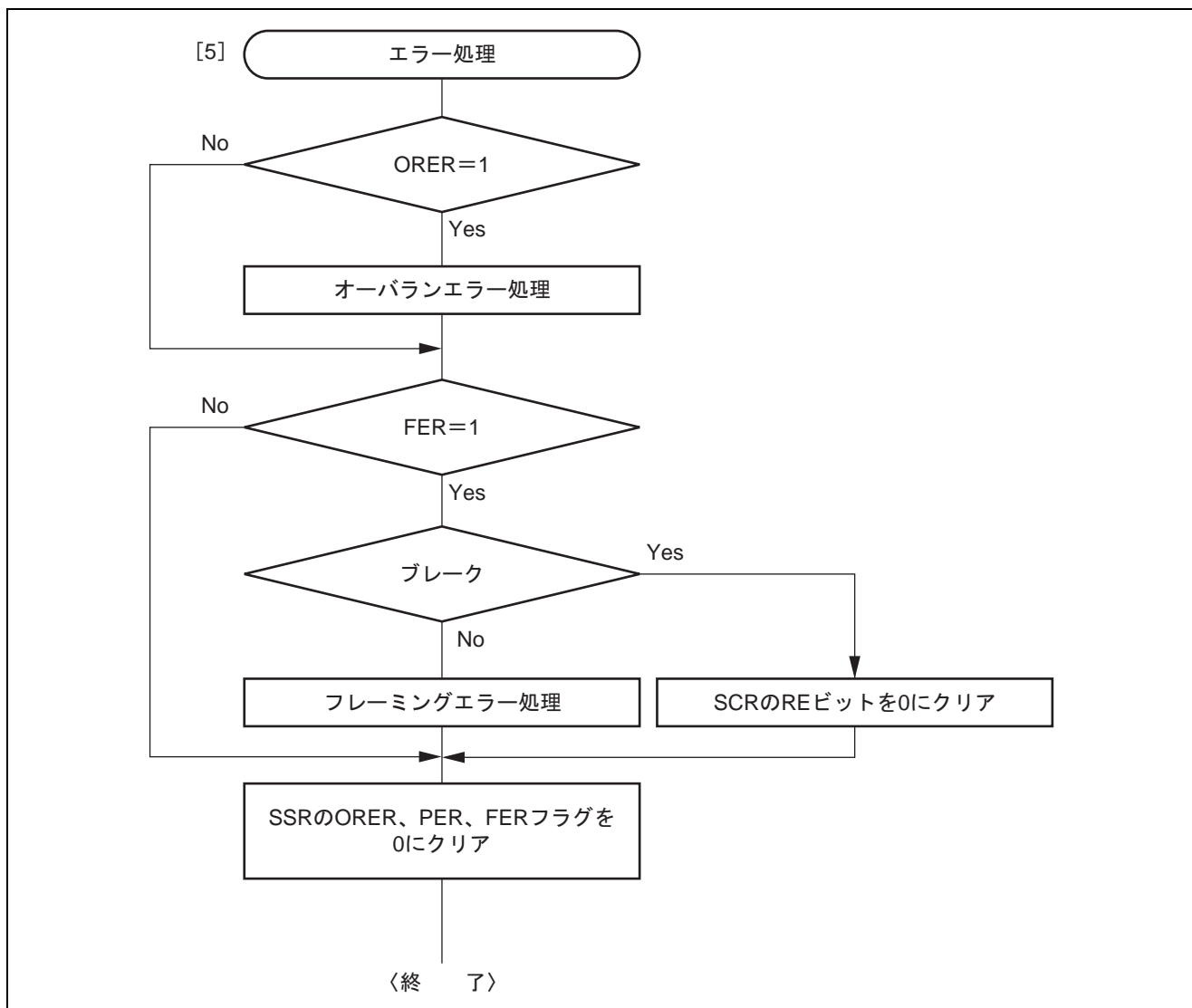


図 13.13 マルチプロセッサシリアル受信のフローチャートの例 (2)

13.6 クロック同期式モードの動作

クロック同期式通信の通信データのフォーマットを図 13.14 に示します。クロック同期式モードではクロックパルスに同期してデータを送受信します。通信データの 1 キャラクタは 8 ビットデータで構成されます。SCI はデータ送信時は同期クロックの立ち下がりから次の立ち下がりまで出力します。データ受信時は同期クロックの立ち上がりに同期してデータを取り込みます。8 ビット出力後の通信回線は最終ビット出力状態を保ちます。クロック同期式モードでは、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。SCI 内部では送信部と受信部が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、送信中に次の送信データのライト、受信中に前の受信データのリードを行うことで連続送受信が可能です。

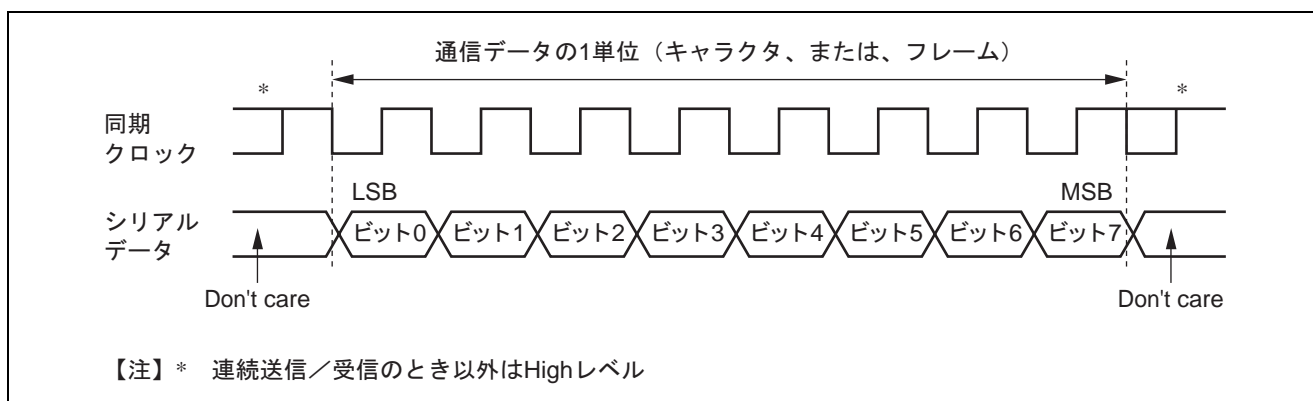


図 13.14 クロック同期式通信のデータフォーマット (LSB ファーストの場合)

13.6.1 クロック

SCR の CKE1、CKE0 の設定により、内蔵ポーレートジェネレータが生成する内部クロックまたは SCK 端子から入力される外部同期クロックを選択できます。内部クロックで動作させるとき、SCK 端子から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は High レベルに固定されます。

13.6.2 SCI の初期化 (クロック同期式)

データの送受信前に、SCR の TE、RE ビットをいずれも 0 にクリアした後、図 13.15 のフローチャートの例に従って初期化してください。動作モードの変更、通信フォーマットの変更などの場合も必ず、TE ビットおよび RE ビットをいずれも 0 にクリアしてから変更を行ってください。TE を 0 にクリアすると、TDRE は 1 にセットされますが、RE を 0 にクリアしても、RDRF、PER、FER、ORER の各フラグ、および RDR は初期化されませんので注意してください。

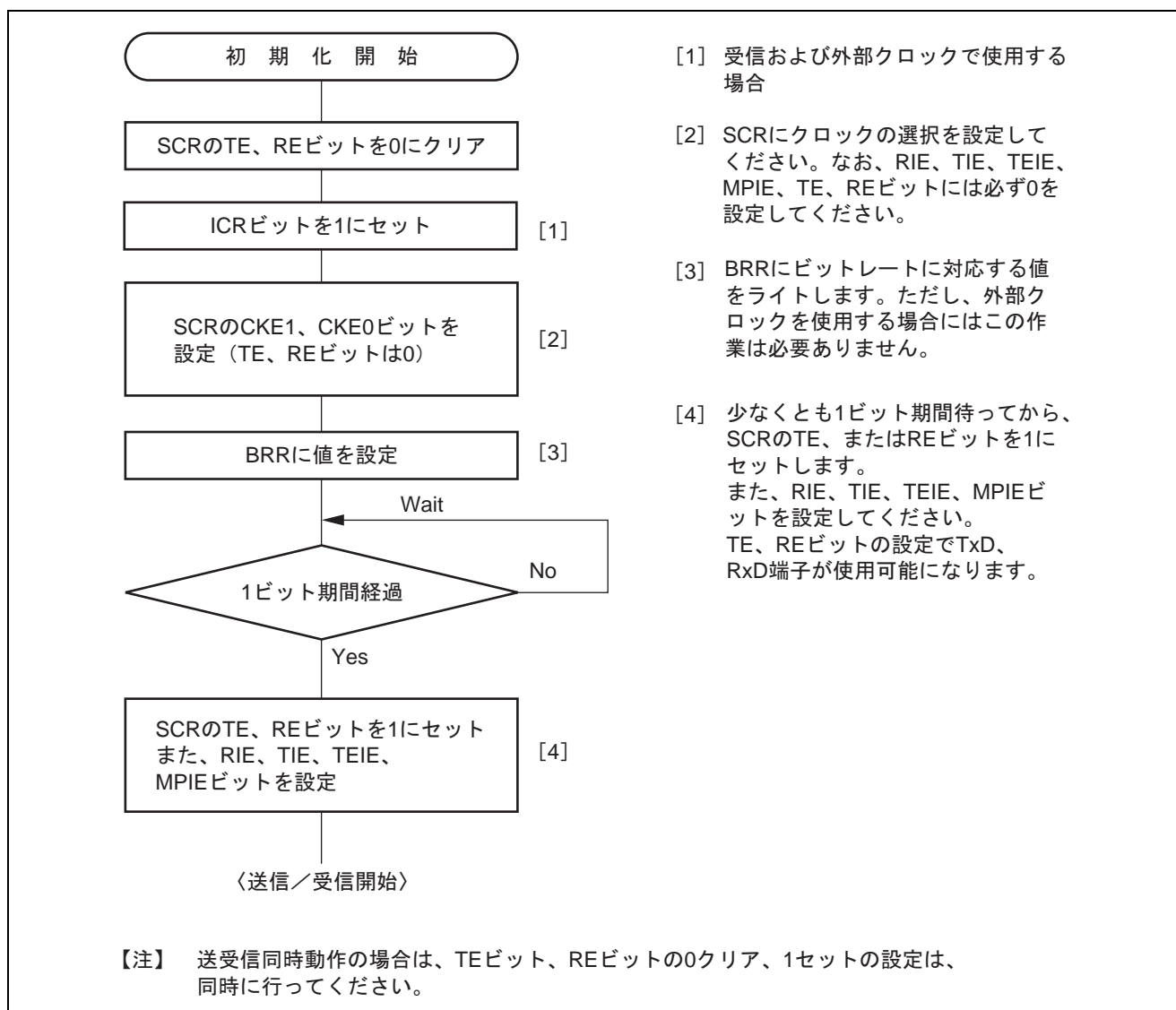


図 13.15 SCI の初期化フローチャートの例

13.6.3 シリアルデータ送信 (クロック同期式)

図 13.16 にクロック同期式モードの送信時の動作例を示します。データ送信時 SCI は以下のように動作します。

1. SCIはSSRのTDREを監視し、クリアされるとTDRにデータが書き込まれたと認識してTDRからTSRにデータを転送します。
2. TDRからTSRにデータを転送すると、TDREを1にセットして送信を開始します。このとき、SCRのTIEが1にセットされているとTXI割り込み要求を発生します。このTXI割り込みルーチンで、前に転送したデータの送信が終了するまでにTDRに次の送信データを書き込むことで連続送信が可能です。
3. クロック出力モードに設定したときには出力クロックに同期して、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、TxD端子から8ビットのデータを出力します。
4. 最終ビットを送り出すタイミングでTDREをチェックします。
5. TDREが0であると次の送信データを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。
6. TDREが1であるとSSRのTENDに1をセットし、最終ビット出力状態を保持します。このときSCRのTEIEが1にセットされているとTEIを発生します。SCK端子はHighレベルに固定されます。

図 13.17 にデータ送信のフローチャートの例を示します。受信エラーフラグ (ORER、FER、PER) が1にセットされた状態では TDRE をクリアしても送信を開始しません。送信開始の前に、必ず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、受信エラーフラグは RE ビットをクリアしただけではクリアされませんので注意してください。

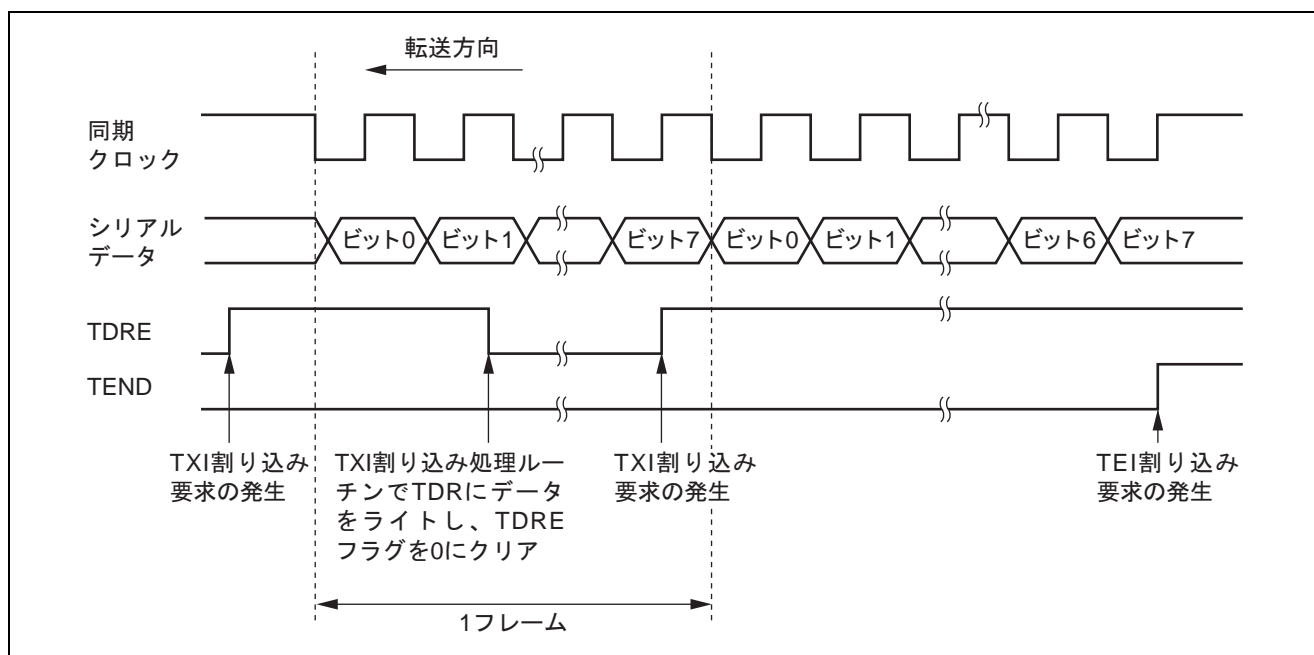


図 13.16 クロック同期式モードの送信時の動作例

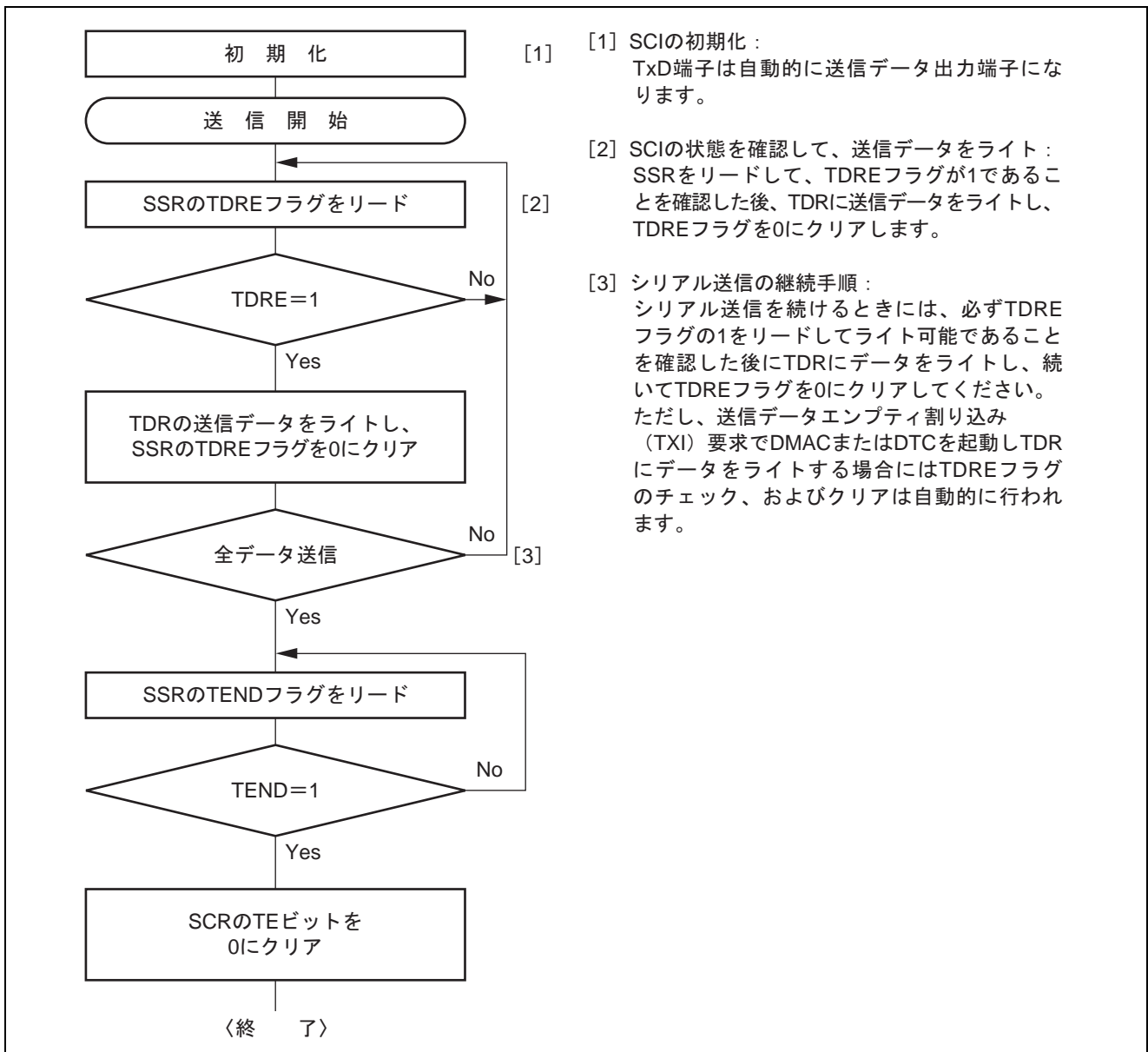


図 13.17 シリアル送信のフローチャートの例

13.6.4 シリアルデータ受信 (クロック同期式)

図 13.18 にクロック同期式モードの受信時の動作例を示します。データ受信時 SCI は以下のように動作します。

1. SCI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て受信を開始し、受信データをRDRに取り込みます。
2. オーバランエラーが発生したとき (SSRのRDRFが1にセットされたまま次のデータを受信完了したとき) はSSRのORERをセットします。このときSCRのRIEが1にセットされているとERI割り込み要求が発生します。受信データはRDRに転送しません。RDRFは1にセットされた状態を保持します。
3. 正常に受信したときはSSRのRDRFをセットし、受信データをRDRに転送します。このときSCRのRIEが1にセットされているとRXI割り込み要求が発生します。このRXI割り込み処理ルーチンでRDRのデータをリードし、RDRFフラグを0にクリア

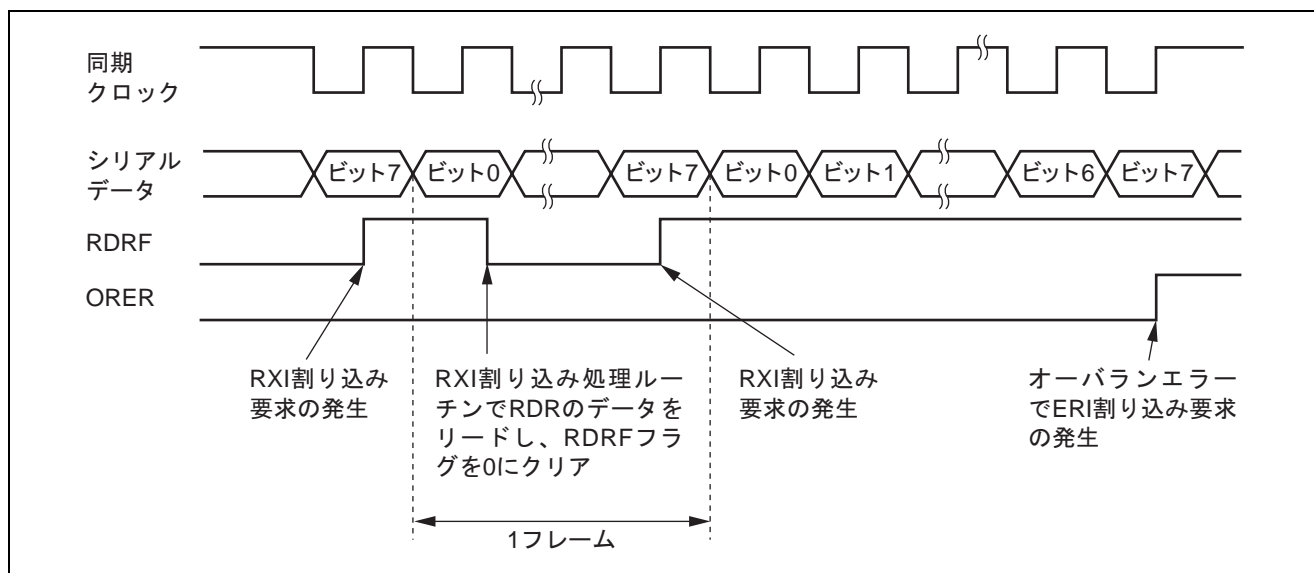


図 13.18 SCI の受信時の動作例

受信エラーフラグがセットされた状態では以後の送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 ORER、FER、PER、および RDRF を 0 にクリアしてください。図 13.19 にデータ受信のためのフローチャートの例を示します。

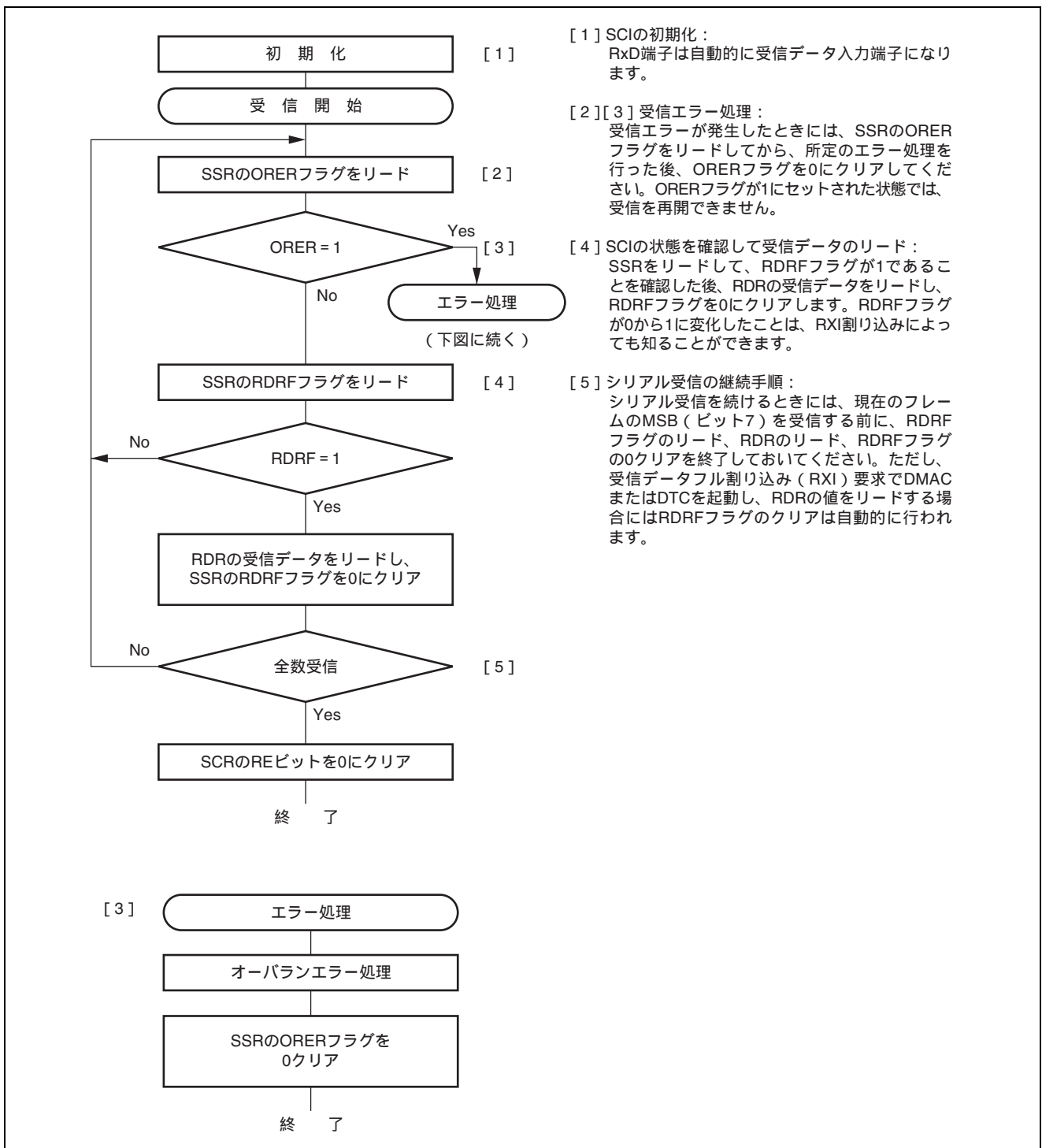


図 13.19 シリアルデータ受信フローチャートの例

13.6.5 シリアルデータ送受信同時動作 (クロック同期式)

図 13.20 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 SCI の初期化後、以下の手順に従って行ってください。

送信から同時送受信へ切り替えるときには、SCI が送信終了状態であること、TDRE および TEND が 1 にセットされていることを確認した後、TE を 0 にクリアしてから TE および RE を 1 命令で同時に 1 にセットしてください。

受信から同時送受信へ切り替えるときには、SCI が受信完了状態であることを確認し、RE を 0 にクリアしてから RDRF およびエラーフラグ (ORER、FER、PER) が 0 にクリアされていることを確認した後、TE および RE を 1 命令で同時に 1 にセットしてください。

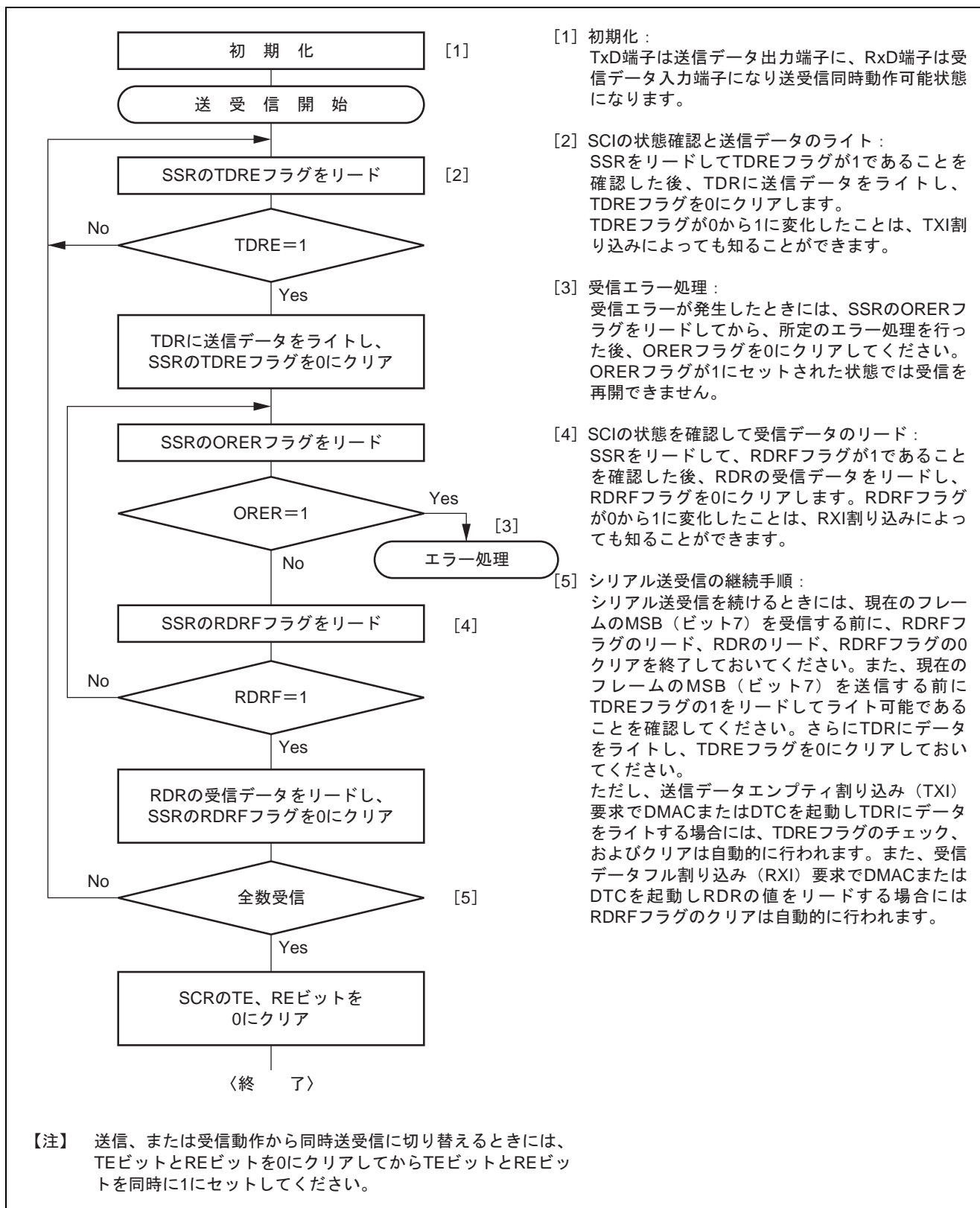


図 13.20 シリアル送受信同時動作のフローチャートの例

13.7 割り込み要因

13.7.1 通常のシリアルコミュニケーションインタフェースモードにおける割り込み

表 13.10 に通常のシリアルコミュニケーションインタフェースモードにおける割り込み要因を示します。各割り込み要因は独立した割り込み要求信号を出力しています（異なるベクタを割り当てることができます）。これらの割り込み要因は、SCR のイネーブルビットにより独立にイネーブルにすることができます。

SSR の TDRE フラグが 1 にセットされると、TXI 割り込み要求が発生します。また、SSR の TEND フラグが 1 にセットされると、TEI 割り込み要求が発生します。TXI 割り込み要求により DTC または DMAC を起動してデータ転送を行うことができます。TDRE フラグは DTC または DMAC によるデータ転送時に自動的に 0 にクリアされます。SSR の RDRF フラグが 1 にセットされると RXI 割り込み要求が発生します。SSR の ORER、PER、FER フラグのいずれかが 1 にセットされると、ERI 割り込み要求が発生します。RXI 割り込み要求で DTC または DMAC を起動してデータ転送を行うことができます。RDRF フラグは DTC または DMAC によるデータ転送時に自動的に 0 にクリアされます。

TEI 割り込みは TEIE ビットが 1 にセットされた状態で TEND フラグが 1 にセットされたとき発生します。

表 13.10 SCI 割り込み要因

名称	割り込み要因	割り込みフラグ	DTC・DMAC の起動	優先順位
ERI	受信エラー	ORER、FER、PER	不可	高 ↑ 低
RXI	受信データフル	RDRF	可	
TXI	送信データエンブティ	TDRE	可	
TEI	送信終了	TEND	不可	

13.8 使用上の注意事項

13.8.1 モジュールストップモードの設定

モジュールストップコントロールレジスタにより、SCIの動作禁止/許可を設定することが可能です。初期値では、SCIの動作は停止します。モジュールストップモードを解除することにより、レジスタのアクセスが可能になります。詳細は「第23章 低消費電力」を参照してください。

13.8.2 ブレークの検出と処理について

フレーミングエラー検出時に、RxD端子の値を直接リードすることでブレークを検出できます。ブレークではRxD端子からの入力がすべて0になりますので、FERがセットされ、またPERもセットされる可能性があります。SCI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にクリアしても再びFERが1にセットされますので注意してください。

13.8.3 マーク状態とブレークの送付

TEが0のとき、TxD端子はDRとDDRにより入出力方向とレベルが決まるI/Oポートになります。これを利用してTxD端子をマーク状態にしたりデータ送信時にブレークの送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端子はI/Oポートになり、TxD端子から0が出力されます。

13.8.4 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)

クロック同期式送受信同時動作時、受信エラーフラグ(ORER)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。

13.8.5 TDRへのライトとTDREフラグの関係について

SSRのTDREフラグはTDRからTSRに送信データの転送が行われたことを示すステータスフラグです。SCIがTDRからTSRにデータを転送すると、TDREフラグが1にセットされます。

TDRへのデータのライトは、TDREフラグの状態にかかわらず行うことができます。しかし、TDREフラグが0の状態新しいデータをTDRにライトすると、TDRに格納されていたデータはTSRに転送されていないため失われてしまいます。したがってTDRへの送信データのライトは、必ずTDREフラグが1にセットされていることを確認してから行ってください。

13.8.6 DMAC または DTC 使用上の制約事項

1. 同期クロックに外部クロックソースを使用する場合、DMACまたはDTCによるTDRの更新後、Pφクロックで5クロック以上経過した後に送信クロックを入力してください。TDRの更新後4クロック以内に送信クロックを入力すると誤動作することがあります（図13.21参照）。
2. DMACまたはDTCによりRDRのリードを行うときは、必ず起動要因を当該SCIの受信完了割り込み（RXI）に設定してください。

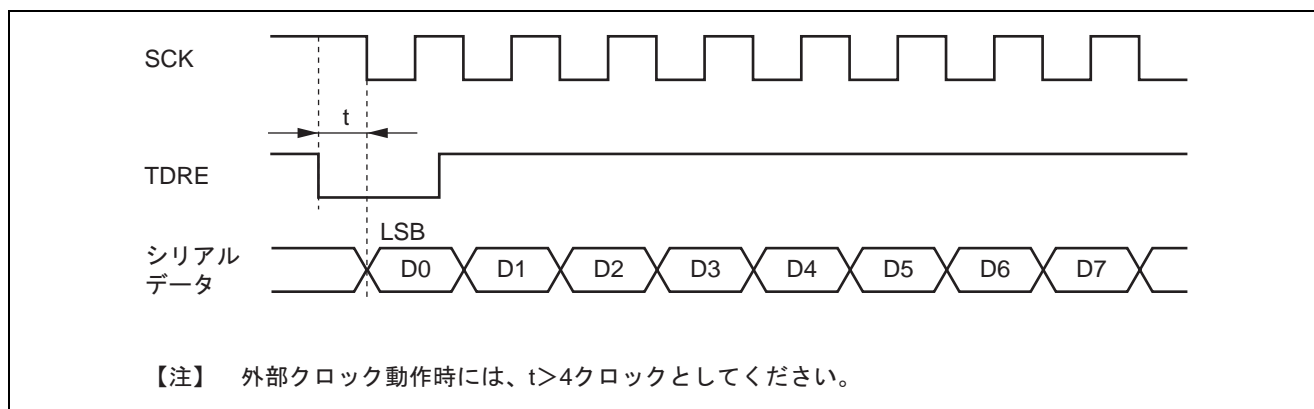


図 13.21 DMAC または DTC によるクロック同期式送信時の例

13.8.7 低消費電力状態遷移時の動作について

(1) 送信

モジュールストップモード、ソフトウェアスタンバイモード遷移は、動作を停止（TE=TIE=TEIE=0）してから行ってください。TE ビットを 0 にクリアすることにより、TSR、TDR および SSR はリセットされます。モジュールストップモード、ソフトウェアスタンバイモード期間中の出力端子の状態は、ポートの設定に依存し、送信中に遷移すると、送信中のデータは不確定になります。これらのモード解除後出力端子は High 出力となります。送信モードを変えないで送信する場合は、TE=1 に設定、TDR に送信データをライトし、TDRE フラグを 0 にクリアすることで、送信を開始します。送信モードを変えて送信する場合は、初期設定から行ってください。図 13.22 に送信時のモード遷移フローチャートの例を示します。なお、モード遷移時のポートの端子状態を図 13.23、図 13.24 に示します。

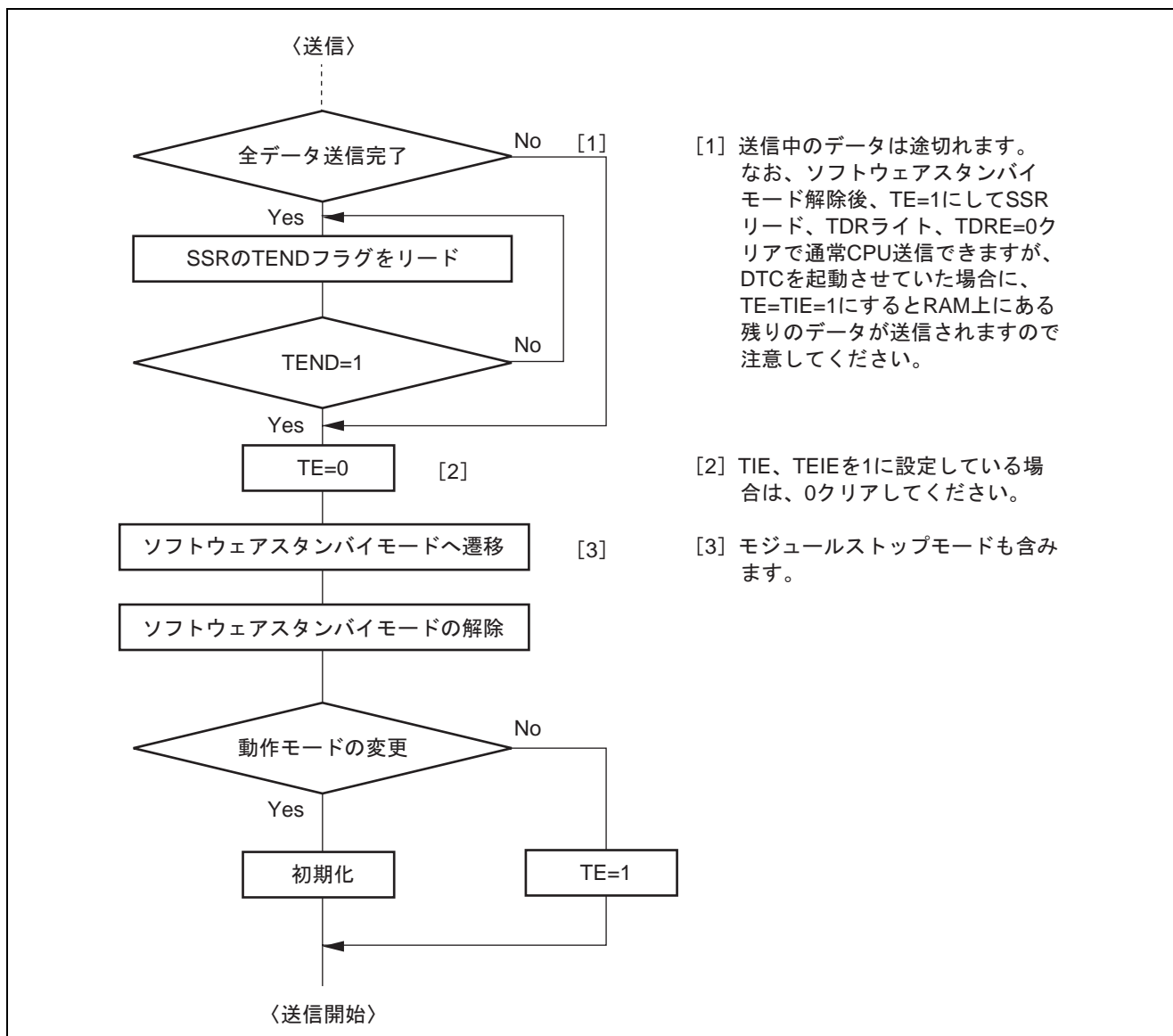
また、DTC 転送による送信からモジュールストップモード、ソフトウェアスタンバイモード遷移時には、動作を停止（TE=TIE=TEIE=0）してから行ってください。解除後 DTC による送信をする場合は TE=1、TIE=1 に設定すると TXI 割り込み要求が発生し、DTC による送信が始まります。

(2) 受信

モジュールストップモード、ソフトウェアスタンバイモード遷移は、受信動作を停止 (RE=0) してから行ってください。RE ビットを 0 にクリアすることにより、RSR、RDR および SSR はリセットされます。受信中に遷移すると、受信中のデータは無効になります。

これらのモード解除後、受信モードを変えないで受信する場合は、RE=1 に設定して受信を開始してください。受信モードを変えて受信する場合は、初期設定から行ってください。

図 13.25 に受信時のモード遷移フローチャートの例を示します。



[1] 送信中のデータは途切れます。なお、ソフトウェアスタンバイモード解除後、TE=1にしてSSRリード、TDRライト、TDRE=0クリアで通常CPU送信できますが、DTCを起動させていた場合に、TE=TIE=1にするとRAM上にある残りのデータが送信されますので注意してください。

[2] TIE、TEIEを1に設定している場合は、0クリアしてください。

[3] モジュールストップモードも含まれます。

図 13.22 送信時のモード遷移フローチャートの例

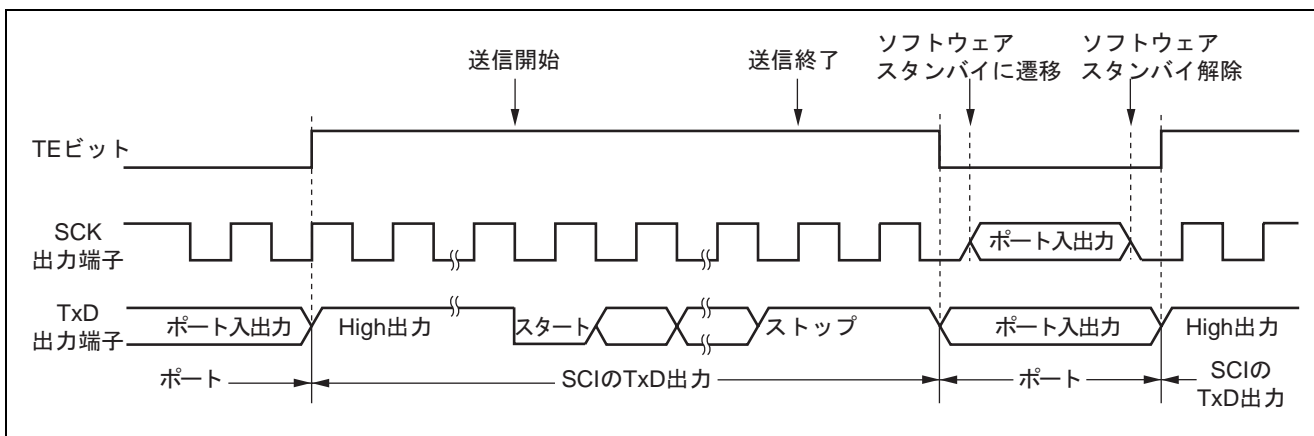
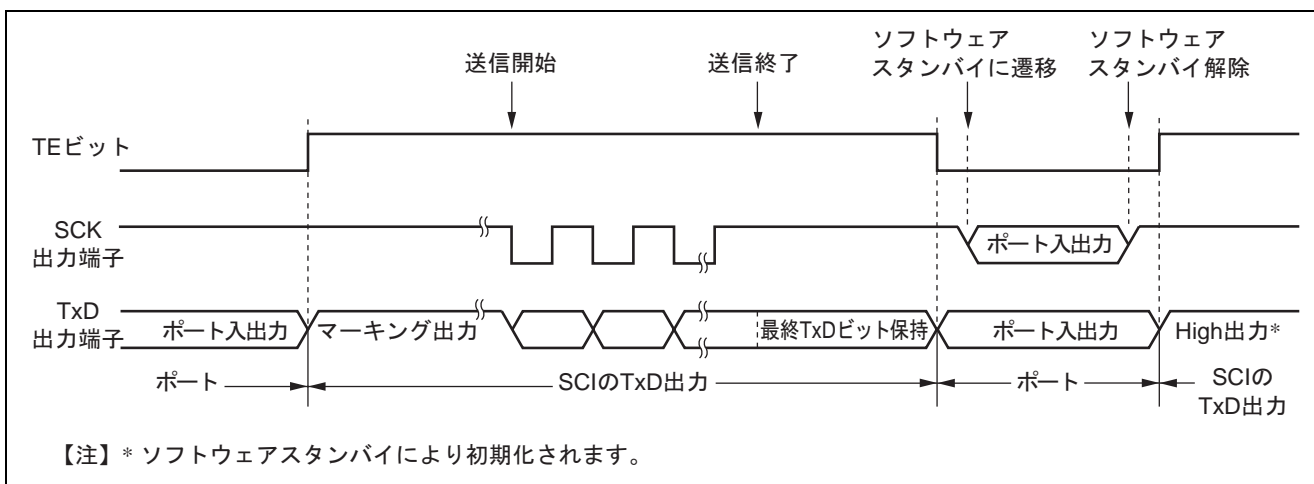


図 13.23 モード遷移時のポート端子状態 (内部クロック、調歩同期送信)



【注】 * ソフトウェアスタンバイにより初期化されます。

図 13.24 モード遷移時のポート端子状態 (内部クロック、クロック同期送信)

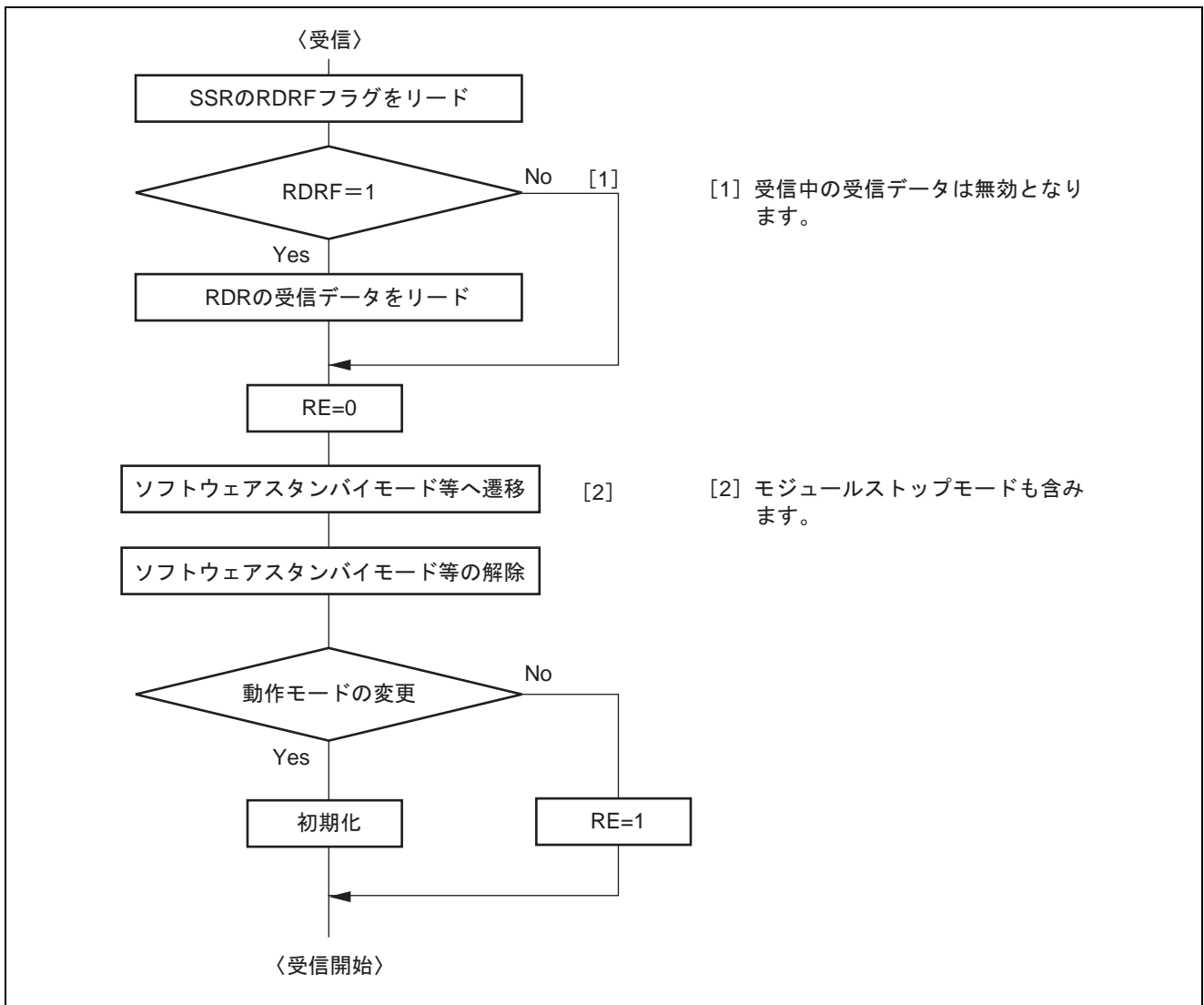


図 13.25 受信時のモード遷移フローチャートの例

13.8.8 クロック同期式モード外部クロック入力

クロック同期式モード時、外部クロック SCK 入力は、High パルス期間および Low パルス期間を 2 クロック以上、周期を 6 クロック以上としてください。

14. コントローラエリアネットワーク (RCAN-TL1)

コントローラエリアネットワーク (RCAN-TL1 : Renesas CAN Time Trigger Level 1) は自動車および産業機器システム等でのリアルタイム通信を目的とした CAN (Controller Area Network) を制御するためのモジュールです。

本章は RCAN-TL1 のプログラムインタフェースについて説明します。

また CAN のデータリンクコントローラの機能については下記参考文献を参照してください。

[参考文献]

1. CAN Specification Version 2.0 part A, Robert Bosch GmbH, 1991
2. CAN Specification Version 2.0 part B, Robert Bosch GmbH, 1991
3. Implementation Guide for the CAN Protocol, CAN Specification 2.0 Addendum, CAN In Automation, Erlangen, Germany, 1997
4. Road vehicles-Controller area network (CAN) : Part 1: Data link layer and physical signaling (ISO-11898-1, 2003)
5. Road vehicles - Controller area network (CAN): Part 4: Time triggered communication (ISO-11898-4, 2004)

14.1 特長

14.1.1 RCAN-TL1 の特長

- CAN規格2.0B対応
- ビットタイミングはISO-11898規格に準拠
- 32個のメールボックス
- クロック周波数 : 16~40MHz
- プログラム可能な31個の送受信メールボックスおよび1個の受信メールボックス
- 低消費電力のCANスリープモードおよびCANバスアクティビティを検出してCANスリープモードを自動解除
- すべてのメールボックスにも対応したプログラム可能な受信フィルタマスク (スタンダードIDおよびエクステンデッドID)
- 最大1Mbpsのプログラム可能なCANデータレート
- リアルタイムアプリケーション起因による優先順位逆転防止のために、優先順位の内部区分機能を備えた転送メッセージのキューを用意
- 豊富な割り込み要因
- テスト機能を内蔵 (リソンオンリモード、エラーパッシブモード)
- 16ビットのフリーランニングタイマ (多様なクロックソース、プリスケアラ、3個のタイマコンペアマッチレジスタ)
- タイムトリガ送信用の6ビットサイクルカウンタ (Basic Cycle)
- タイマのコンペアマッチレジスタによる割り込み
- タイマカウンタのクリアおよび設定が可能
- タイムトリガ用レジスタ : Local_Time、Cycle_time、Ref_Mark、Tx_Enable_Window、Ref_Trigger_Offset

- SOFでのフレキシブルなタイムスタンプ機能を送信、受信ともにサポート
- イベントトリガ送信に加え、タイムトリガ送信および定周期送信もサポート
- サイクルカウンタ (Basic Cycle) をCANフレームに組み込んで送信可能

14.1.2 本マイコンにおける特長

- 32メールボックスRCAN×2チャンネル搭載 (RCAN_0、1)
- RCAN2チャンネルはピンファンクションコントローラ (PFC) の設定により、次の接続による使用が可能
- 32メールボックス×2チャンネル (RCAN_0、1) 64メールボックス*×1チャンネル (RCAN_0、1) RCAN_0、1ともにメールボックス0の受信メッセージによりDMAC起動可能

【注】 * 64メールボックス構成においては使用上の注意事項があります。「14.12 使用上の注意事項」を参照ください。

14.2 構成

14.2.1 ブロック図

RCAN-TL1は、CAN2.0B ActiveとISO-11898をサポートするCANフレームを構成、制御する自由度の大きい洗練された方法を提供します。RCAN-TL1は機能的に、マイクロプロセッサインタフェース (MPI)、メールボックス、メールボックスコントロール、タイマ、およびCANインタフェースの5種類のブロックからなります。

図 14.1 に RCAN-TL1 のブロック図を示します。

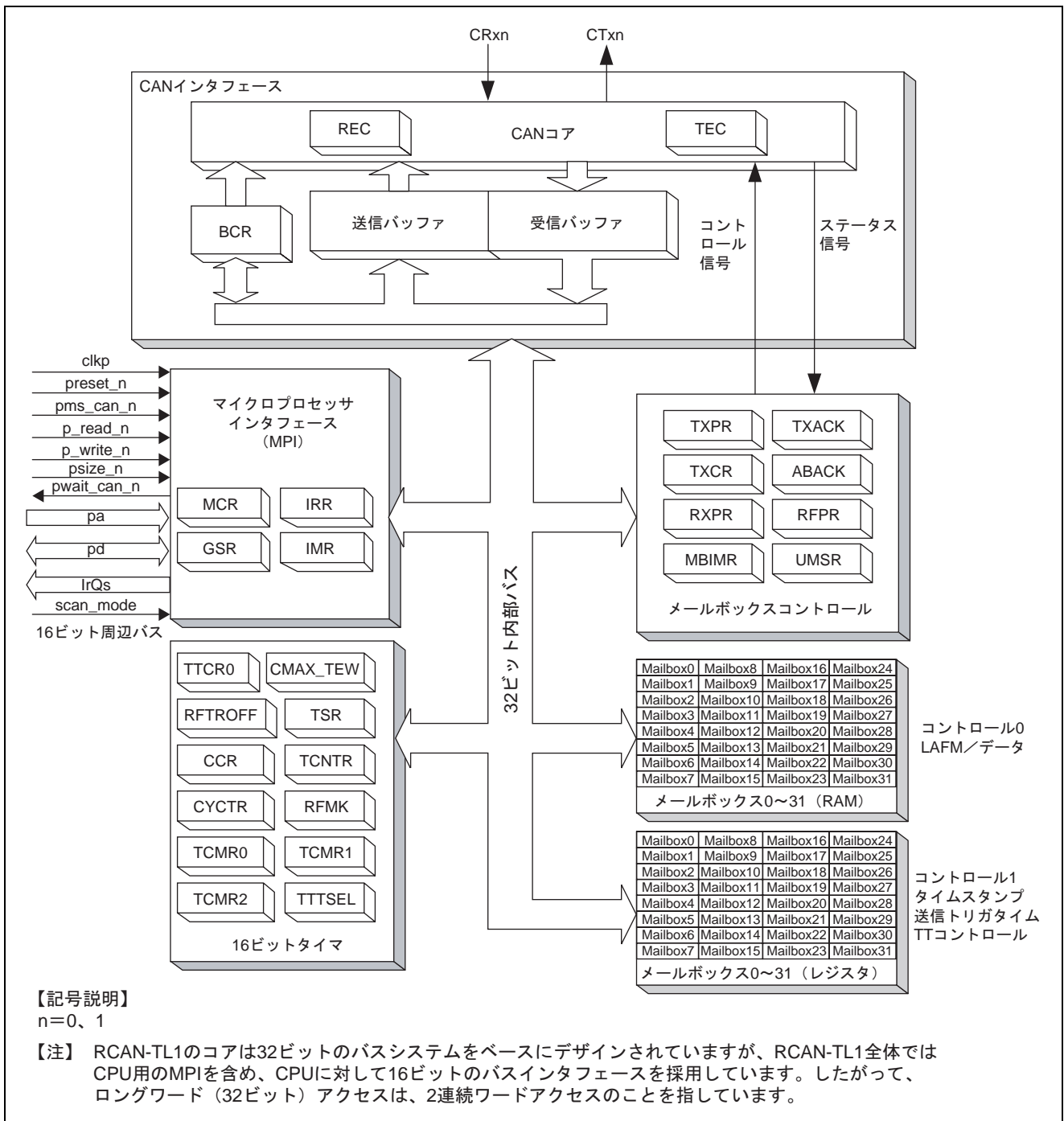


図 14.1 RCAN-TL1 のブロック図 (1 チャンネルあたり)

14.2.2 各ブロックの機能

(1) マイクロプロセッサインタフェース (MPI)

MPIは、CPUとRCAN-TL1のレジスタやメールボックスとの間の通信を可能にし、メモリインタフェースを制御します。またMPIはCANバスのアクティブを検出し、MPI自身やRCAN-TL1の他のモジュールにもCANバスのアクティブを通知するウェイクアップコントロールロジックを持っているので、RCAN-TL1は自動的にCANスリープモードを解除することができます。なおMPIのレジスタには、MCR、IRR、GSR、およびIMRがあります。

(2) メールボックス

メールボックスは、メッセージバッファとしてRAMおよびレジスタに配列されています。RAMとレジスタ内には、それぞれ32個のメールボックスがあり、以下の情報を格納します。

[RAM]

- CANメッセージコントロール (ID、RTR、IDEなど)
- CANメッセージデータ (CANデータフレーム用)
- 受信用のローカルアクセプタンスフィルタマスク (LAFM)

[レジスタ]

- CANメッセージコントロール (DLC)
- メッセージ送信/受信用タイムスタンプ
- 3ビットのメールボックスコンフィギュレーション、自動再送信無効ビット、リモートリクエスト用自動送信ビット、ニューメッセージコントロールビット
- 送信トリガタイム

(3) メールボックスコントロール

メールボックスコントロールは以下のような機能を持ちます。

- メッセージ受信時は、IDを比較しCANインタフェースからのメッセージをメールボックスに格納するためのRAMアドレスとデータを生成し、対応するレジスタをセット/クリアします。
- イベントトリガメッセージ送信時は、内部アービトレーションを動作させて正しい優先順位のメッセージを選択し、メールボックスからCANインタフェースの送信バッファにメッセージをロードします。その後、対応するレジスタをセット/クリアします。タイムトリガ送信の場合は、送信トリガタイムのコンペアマッチでメッセージをロードします。
- CPUとメールボックスコントロール間のメールボックスアクセスのアービトレーションを行います。
- レジスタは、TXPR、TXCR、TXACK、ABACK、RXPR、RFPR、MBIMR、およびUMSRがあります。

(4) タイマ

本タイマは、特定の時間枠でメッセージを送信し、結果を記録するための機能ブロックです。16ビットのフリーランニングアップカウンタで、CPUで制御することができます。ローカルタイムと比較する16ビットのコンペアマッチレジスタが1個と、サイクルタイムと比較するコンペアマッチレジスタが2個あります。これらのコンペアマッチレジスタは、割り込み信号を発生し、カウンタをクリアすることができます。本タイマのクロックは、システムクロックから生成する多数のクロック周期から選択することができますが、CANバスの1ビットタイミングでカウントするようにプログラムすることも可能です。本タイマのレジスタには、TCNTR、TTCR0、CMAX_TEW、RETROFF、TSR、CCR、CYCTR、RFMK、TCMR0、TCMR1、TCMR2、およびTTTSELがあります。

(5) CAN インタフェース

本ブロックは参考文献[2]と[4]のCANバスデータリンクコントローラ仕様をサポートしています。これはOSIモデルで規定されるデータリンクコントローラの全機能を満たします。また、CANバスに特化したレジスタやロジックも提供します。具体的には、受信エラーカウンタ、送信エラーカウンタ、ビットコンフィギュレーションレジスタ、種々のテストモードなどです。さらに、CANデータリンクコントローラの送受信を格納する機能もあります。

14.2.3 端子構成

RCAN-TL1の端子構成を表14.1に示します。

表 14.1 端子構成

名称	端子名	入出力	機能
送信データ端子	CTxn	出力	CANバス送信用端子です。
受信データ端子	CRxn	入力	CANバス受信用端子です。

【注】 n=0, 1

14.2.4 メモリマップ

RCAN-TL1のメモリマップを図14.2に示します。

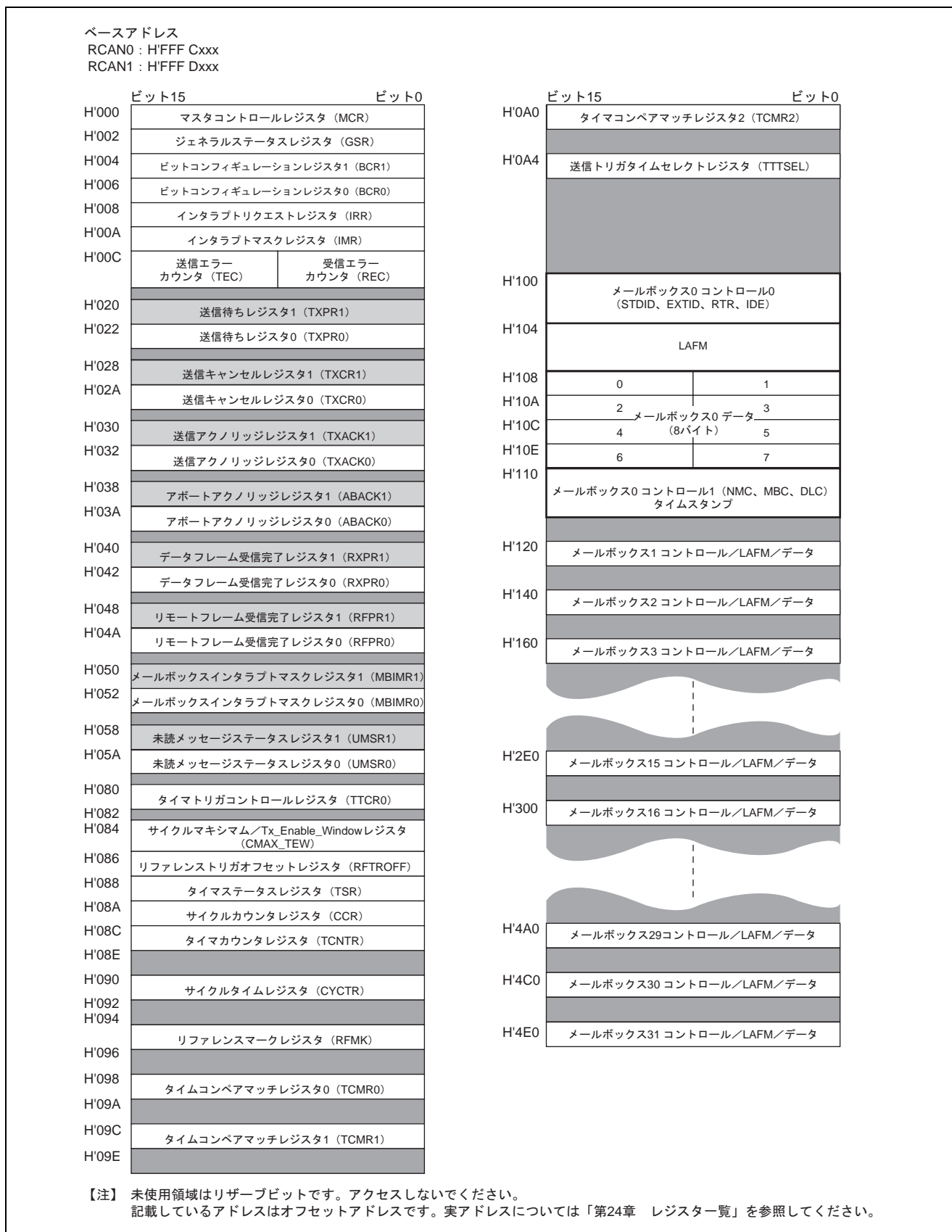


図 14.2 RCAN-TL1 のメモリマップ (1チャンネルあたり)

14.3 メールボックス

14.3.1 メールボックスの構成

メールボックスはCAN フレームを送受信するためのメッセージバッファとして動作します。各メールボックスは、メッセージコントロール、ローカルアクセプタンスフィルタマスク (LAFM)、メッセージデータの3個の格納フィールドからなります。さらに、タイムスタンプ、タイムトリガコンフィギュレーション、およびタイムトリガコントロールのフィールドを持つメールボックスもあります。

表 14.2 に各メールボックスに対するメッセージコントロール、LAFM、メッセージデータ、タイムスタンプ、送信トリガタイム、およびタイムトリガコントロールのアドレスマップを示します。

表 14.2 各メールボックスのアドレスマップ (1 チャンネルあたり)

メール ボックス	アドレス						
	コントロール 0	LAFM	データ	コントロール 1	タイム スタンプ	トリガタイム	TT コントロール
	4 バイト	4 バイト	8 バイト	2 バイト	2 バイト	2 バイト	2 バイト
0 (受信のみ)	H'100-H'103	H'104-H'107	H'108-H'10F	H'110-H'111	H'112-H'113	—	—
1	H'120-H'123	H'124-H'127	H'128-H'12F	H'130-H'131	H'132-H'133	—	—
2	H'140-H'143	H'144-H'147	H'148-H'14F	H'150-H'151	H'152-H'153	—	—
3	H'160-H'163	H'164-H'167	H'168-H'16F	H'170-H'171	H'172-H'173	—	—
4	H'180-H'183	H'184-H'187	H'188-H'18F	H'190-H'191	H'192-H'193	—	—
5	H'1A0-H'1A3	H'1A4-H'1A7	H'1A8-H'1AF	H'1B0-H'1B1	H'1B2-H'1B3	—	—
6	H'1C0-H'1C3	H'1C4-H'1C7	H'1C8-H'1CF	H'1D0-H'1D1	H'1D2-H'1D3	—	—
7	H'1E0-H'1E3	H'1E4-H'1E7	H'1E8-H'1EF	H'1F0-H'1F1	H'1F2-H'1F3	—	—
8	H'200-H'203	H'204-H'207	H'208-H'20F	H'210-H'211	H'212-H'213	—	—
9	H'220-H'223	H'224-H'227	H'228-H'22F	H'230-H'231	H'232-H'233	—	—
10	H'240-H'243	H'244-H'247	H'248-H'24F	H'250-H'251	H'252-H'253	—	—
11	H'260-H'263	H'264-H'267	H'268-H'26F	H'270-H'271	H'272-H'273	—	—
12	H'280-H'283	H'284-H'287	H'288-H'28F	H'290-H'291	H'292-H'293	—	—
13	H'2A0-H'2A3	H'2A4-H'2A7	H'2A8-H'2AF	H'2B0-H'2B1	H'2B2-H'2B3	—	—
14	H'2C0-H'2C3	H'2C4-H'2C7	H'2C8-H'2CF	H'2D0-H'2D1	H'2D2-H'2D3	—	—
15	H'2E0-H'2E3	H'2E4-H'2E7	H'2E8-H'2EF	H'2F0-H'2F1	H'2F2-H'2F3	—	—
16	H'300-H'303	H'304-H'307	H'308-H'30F	H'310-H'311	—	—	—
17	H'320-H'323	H'324-H'327	H'328-H'32F	H'330-H'331	—	—	—
18	H'340-H'343	H'344-H'347	H'348-H'34F	H'350-H'351	—	—	—
19	H'360-H'363	H'364-H'367	H'368-H'36F	H'370-H'371	—	—	—
20	H'380-H'383	H'384-H'387	H'388-H'38F	H'390-H'391	—	—	—
21	H'3A0-H'3A3	H'3A4-H'3A7	H'3A8-H'3AF	H'3B0-H'3B1	—	—	—
22	H'3C0-H'3C3	H'3C4-H'3C7	H'3C8-H'3CF	H'3D0-H'3D1	—	—	—

メール ボックス	アドレス						
	コントロール 0	LAFM	データ	コントロール 1	タイム スタンプ	トリガタイム	TT コントロール
	4 バイト	4 バイト	8 バイト	2 バイト	2 バイト	2 バイト	2 バイト
23	H'3E0-H'3E3	H'3E4-H'3E7	H'3E8-H'3EF	H'3F0-H'3F1	-	-	-
24	H'400-H'403	H'404-H'407	H'408-H'40F	H'410-H'411	-	H'414-H'415	H'416-H'417
25	H'420-H'423	H'424-H'427	H'428-H'42F	H'430-H'431	-	H'434-H'435	H'436-H'437
26	H'440-H'443	H'444-H'447	H'448-H'44F	H'450-H'451	-	H'454-H'455	H'456-H'457
27	H'460-H'463	H'464-H'467	H'468-H'46F	H'470-H'471	-	H'474-H'475	H'476-H'477
28	H'480-H'483	H'484-H'487	H'488-H'48F	H'490-H'491	-	H'494-H'495	H'496-H'497
29	H'4A0-H'4A3	H'4A4-H'4A7	H'4A8-H'4AF	H'4B0-H'4B1	-	H'4B4-H'4B5	H'4B6-H'4B7
30	H'4C0-H'4C3	H'4C4-H'4C7	H'4C8-H'4CF	H'4D0-H'4D1	H'4D2-H'4D3 (ローカル タイム)	H'4D4-H'4D5	-
31	H'4E0-H'4E3	H'4E4-H'4E7	H'4E8-H'4EF	H'4F0-H'4F1	H'4F2-H'4F3 (ローカル タイム)	-	-

メールボックス 0 は受信専用です。他のすべてのメールボックスは、メッセージコントロールの MBC (メールボックス構成) ビットの設定により、送信、受信ともに可能です。メールボックスの構成の詳細を図 14.3 ~ 図 14.5 に示します。

表 14.3 メールボックスの役割

	イベントトリガ		タイムトリガ		備考	
	送信	受信	送信	受信	タイムスタンプ レジスタ	送信トリガ タイム レジスタ
MB31	設定可	設定可	-	タイムリファレンス 受信	有	-
MB30	設定可	設定可	タイムマスタモード時 タイムリファレンス送信	タイムスレーブ モード時受信	有	有
MB29~MB24	設定可	設定可	OK	OK	-	有
MB23~MB16	設定可	設定可	- (ET)	OK	-	-
MB15~MB1	設定可	設定可	- (ET)	OK	有	-
MB0	-	設定可	-	OK	有	-

【注】 ET : タイムトリガモード時のマージドアービトラレーティングウィンドウ内で送信動作可能です。

・ MB0 (タイムスタンプ付き受信用メールボックス)

アドレス	データベース															アクセスサイズ	フィールド名
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1		
H'100	IDE	RTR	0	STDID[10:0]										EXTID[17:16]	16/32	コントロール0	
H'102	EXTID[15:0]																16
H'104	IDE LAFM	0	0	STDID_LAFM[10:0]										EXTID LAFM[17:16]	16/32	LAFM	
H'106	EXTID_LAFM[15:0]																16
H'108	MSG_DATA_0 (最初のRx/Tx バイト)							MSG_DATA_1							8/16/32	データ	
H'10A	MSG_DATA_2							MSG_DATA_3							8/16		
H'10C	MSG_DATA_4							MSG_DATA_5							8/16/32		
H'10E	MSG_DATA_6							MSG_DATA_7							8/16		
H'110	0	0	NMC	0	0	0	0	0	0	0	0	0	0	0	DLC[3:0]		8/16
H'112	TimeStamp[15:0] (CYCTR[15:0] or CCR[5:0]/CYCTR[15:6] at SOF)															16	タイムスタンプ

・ MB1~15 (タイムスタンプ付きメールボックス)

アドレス	データベース															アクセスサイズ	フィールド名
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1		
H'100+N*32	IDE	RTR	0	STDID[10:0]										EXTID[17:16]	16/32	コントロール0	
H'102+N*32	EXTID[15:0]																16
H'104+N*32	IDE LAFM	0	0	STDID_LAFM[10:0]										EXTID LAFM[17:16]	16/32	LAFM	
H'106+N*32	EXTID_LAFM[15:0]																16
H'108+N*32	MSG_DATA_0 (最初のRx/Tx バイト)							MSG_DATA_1							8/16/32	データ	
H'10A+N*32	MSG_DATA_2							MSG_DATA_3							8/16		
H'10C+N*32	MSG_DATA_4							MSG_DATA_5							8/16/32		
H'10E+N*32	MSG_DATA_6							MSG_DATA_7							8/16		
H'110+N*32	0	0	NMC	ATX	DART	MBC[2:0]	0	0	0	0	0	0	0	0	DLC[3:0]		8/16
H'112+N*32	TimeStamp[15:0] (CYCTR[15:0] or CCR[5:0]/CYCTR[15:6] at SOF)															16	タイムスタンプ

【注】 * MBC[1]=1固定

1. グレー表示のビットはリザーブビットです。書き込む値は0にしてください。読み出し値は必ずしも0ではなく、また保証もされません。
2. ATXとDARTはメールボックス0ではサポートされません。また、メールボックス0のMBCの設定値は限られています。
3. MCR15ビットの初期値が1のため、メッセージコントロールおよびLAFMのSTDID、RTR、IDE、EXTIDの順序はHCAN2と異なる順序です。
4. N : 1~15 (メールボックス番号)

図 14.3 メールボックス (0~15) の構成

・ MB16~23 (メールボックス : タイムスタンプなし)

アドレス	データベース														アクセスサイズ	フィールド名
	15	14	13	12	11	10	9	8	7	6	5	4	3	2		
H'100+N*32	IDE	RTR	0	STDID[10:0]										EXTID[17:16]	16/32	コントロール0
H'102+N*32	EXTID[15:0]														16	
H'104+N*32	IDE LAFM	0	0	STDID_LAFM[10:0]										EXTID_LAFM[17:16]	16/32	LAFM
H'106+N*32	EXTID_LAFM[15:0]														16	
H'108+N*32	MSG_DATA_0 (最初のRx/Tx バイト)							MSG_DATA_1							8/16/32	データ
H'10A+N*32	MSG_DATA_2							MSG_DATA_3							8/16	
H'10C+N*32	MSG_DATA_4							MSG_DATA_5							8/16/32	
H'10E+N*32	MSG_DATA_6							MSG_DATA_7							8/16	
H'110+N*32	0	0	NMC	ATX	DART	MBC[2:0]		0	0	0	0	DLC[3:0]		8/16	コントロール1	

・ MB24~29 (タイムトリガモード時タイムトリガ送信)

アドレス	データベース														アクセスサイズ	フィールド名	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2			1
H'100+N*32	IDE	RTR	0	STDID[10:0]										EXTID[17:16]	16/32	コントロール0	
H'102+N*32	EXTID[15:0]														16		
H'104+N*32	IDE LAFM	0	0	STDID_LAFM[10:0]										EXTID_LAFM[17:16]	16/32	LAFM	
H'106+N*32	EXTID_LAFM[15:0]														16		
H'108+N*32	MSG_DATA_0 (最初のRx/Tx バイト)							MSG_DATA_1							8/16/32	データ	
H'10A+N*32	MSG_DATA_2							MSG_DATA_3							8/16		
H'10C+N*32	MSG_DATA_4							MSG_DATA_5							8/16/32		
H'10E+N*32	MSG_DATA_6							MSG_DATA_7							8/16		
H'110+N*32	0	0	NMC	ATX	DART	MBC[2:0]		0	0	0	0	DLC[3:0]		8/16	コントロール1		
H'112+N*32	リザーブ														—	—	
H'114+N*32	送信トリガタイム (TTT)														16	トリガタイム	
H'116+N*32	TTW[1:0]	オフセット							0	0	0	0	0	Rep_Factor		16	TTコントロール

【注】 1. グレー表示のビットはリザーブビットです。書き込む値は0にしてください。読み出し値は必ずしも0ではなく、また保証もされません。
 2. MCR15ビットの初期値が1のため、メッセージコントロールおよびLAFMのSTDID、RTR、IDE、EXTIDの順序はHCAN2と異なる順序です。
 3. N : 16~29 (メールボックス番号)

図 14.4 メールボックス (16~29) の構成

・MB30 (タイムトリガモード時タイムリファレンス送信)

アドレス	データベース															アクセスサイズ	フィールド名
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1		
4C0	IDE	RTR	0	STDID[10:0]										EXTID[17:16]	16/32	コントロール0	
4C2	EXTID[15:0]																16
4C4	IDE LAFM	0	0	STDID_LAFM[10:0]										EXTID_LAFM[17:16]	16/32	LAFM	
4C6	EXTID_LAFM[15:0]																16
4C8	MSG_DATA_0 (最初のRx/Tx バイト)							MSG_DATA_1							8/16/32	データ	
4CA	MSG_DATA_2							MSG_DATA_3							8/16		
4CC	MSG_DATA_4							MSG_DATA_5							8/16/32		
4CE	MSG_DATA_6							MSG_DATA_7							8/16		
4D0	0	0	NMC	ATX	DART	MBC[2:0]	0	0	0	0	DLC[3:0]	8/16	コントロール1				
4D2	TimeStamp[15:0] (SOFでのTCNTR値)															16	タイムスタンプ
4D4	タイムリファレンスとしての送信トリガタイム (TTT)															16	トリガタイム

・MB31 (タイムトリガモード時タイムリファレンス受信)

アドレス	データベース															アクセスサイズ	フィールド名
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1		
4E0	IDE	RTR	0	STDID[10:0]										EXTID[17:16]	16/32	コントロール0	
4E2	EXTID[15:0]																16
4E4	IDE LAFM	0	0	STDID_LAFM[10:0]										EXTID_LAFM[17:16]	16/32	LAFM	
4E6	EXTID_LAFM[15:0]																16
4E8	MSG_DATA_0 (最初のRx/Tx バイト)							MSG_DATA_1							8/16/32	データ	
4EA	MSG_DATA_2							MSG_DATA_3							8/16		
4EC	MSG_DATA_4							MSG_DATA_5							8/16/32		
4EE	MSG_DATA_6							MSG_DATA_7							8/16		
4F0	0	0	NMC	ATX	DART	MBC[2:0]	0	0	0	0	DLC[3:0]	8/16	コントロール1				
4F2	TimeStamp[15:0] (SOFでのTCNTR値)															16	タイムスタンプ

【注】 1. グレー表示のビットはリザーブビットです。書き込む値は0にしてください。読み出し値は必ずしも0ではなく、また保証もされません。
 2. MCR15ビットの初期値が1のため、メッセージコントロールおよびLAFMのSTDID、RTR、IDE、EXTIDの順序はHCAN2と異なる順序です。
 3. N : 30、31 (メールボックス番号)

図 14.5 メールボックス (30、31) の構成

14.3.2 メッセージコントロールフィールド

レジスタ名	アドレス	ビット	ビット名	説明
MB[N]. CONTROL0H	H'100+N*32	15	IDE	ID 拡張 CAN データフレームとリモートフレームがスタンダードフォーマットか エクステンデッドフォーマットかを区別します。 0: スタンダードフォーマット 1: エクステンデッドフォーマット
		14	RTR	リモート送信リクエスト データフレームとリモートフレームを区別します。データフレームかリ モートフレームかによって受信 CAN フレームがこのビットを書き替え ます。 【重要】 MBC=B'001 でデータフレーム自動送信 (ATX) ビットをセット すると RTR はセットできません。リモートフレームを受信する と、CPU は対応する RFPR ビットまたは IRR2 (リモートフレ ーム受信割り込み) ビットによって通知されますが、RCAN-TL1 は現在のメッセージをデータフレームとして送信する必要がある ので RTR ビットは変化しません。 【重要】 MBC=B'001 で ATX=1 を設定し、リモートフレームを受信した メールボックスは自動的に送信設定されます。そのときデータ フレームが送信されるように、RTR は 0 に設定しなければなり ません。 0: データフレーム 1: リモートフレーム
		13	—	リザーブビット 初期値は不定です。書き込む値は常に 0 にしてください。
		12~2	STDID [10:0]	スタンダード ID データフレームとリモートフレームの ID (スタンダード ID) を設定する ビットです。
		1、0	EXTID [17:16]	エクステンデッド ID データフレームとリモートフレームの ID (エクステンデッド ID) を設定 するビットです。
MB[N]. CONTROL0L	H'102+N*32	15~0	EXTID [15:0]	

【記号説明】 N : 0~31 (メールボックス番号)

- メールボックス0

ビット	15	14	13	12	11	10	9	8
ビット名	0	0	NMC	0	0	MBC2	MBC1	MBC0
初期値:	0	0	0	0	0	1	1	1
R/W:	R	R	R/W	R	R	R/W	R	R/W
ビット	7	6	5	4	3	2	1	0
ビット名	0	0	0	0	DLC3	DLC2	DLC1	DLC0
初期値:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

【注】 MBC[1]の値は常に1です。

- メールボックス1~31

ビット	15	14	13	12	11	10	9	8
ビット名	0	0	NMC	ATX	DART	MBC2	MBC1	MBC0
初期値:	0	0	0	0	0	1	1	1
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W
ビット	7	6	5	4	3	2	1	0
ビット名	0	0	0	0	DLC3	DLC2	DLC1	DLC0
初期値:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

レジスタ名	アドレス	ビット	ビット名	説明
MB[N]. CONTROL1	H'110+N*32	15、14	—	リザーブビット 読み出すと常に0が読み出されます。書き込む値は常に0にしてください。
		13	NMC	<p>ニューメッセージコントロール</p> <p>このビットが0にセットされると、RXPR あるいは RFPR がすでにセットされているメールボックスは新しいメッセージを格納せず、前のメッセージを保持し UMSR の対応するビットをセットします。</p> <p>このビットが1にセットされると、RXPR あるいは RFPR がすでにセットされているメールボックスは新しいメッセージをオーバライトし、UMSR の対応するビットをセットします。</p> <p>【重要】 もしリモートフレームがデータフレームで書き換えられたり、あるいはその逆の場合には、RXPR および RFPR フラグは (USMR と一緒に) 同一メールボックスにセットされる可能性があります。この場合、メールボックスコントロールフィールド内にある RTR ビットも上書きされます。</p> <p>【重要】 タイムトリガモード使用時には、メールボックス 31 の NMC を必ず1にセットし、RXPR[31]がクリアされなくてもすべてのリファレンスメッセージを受信できるようにしてください。</p> <p>0 : オーバランモード 1 : オーバライトモード</p>

レジスタ名	アドレス	ビット	ビット名	説明
MB[N]. CONTROL1	H'110+N*32	12	ATX	<p>データフレーム自動送信</p> <p>本ビットが1にセットされ、リモートフレームがメールボックスで受信されると、DLCが格納され、TXPRは自動的にセットされます。同じメールボックスから書き換えられたDLCを使用し、自動的にデータフレームが送信されます。</p> <p>自動送信設定されたメールボックスは、メッセージ送信プライオリティビット (MCR2) の設定に従って、ID優先順位あるいはメールボックス優先順位にスケジューリングされます。本機能を使用するにはMBC[2:0]をB'001に設定してください。この機能を用いて送信を行うとき、用いるデータ長コード (DLC) は受信されたものが使用されます。これを適用するには、リモートフレームのDLCは要求されているデータフレームのDLCに対応したものであることが必要です。</p> <p>【重要】 1. ATXが使用され、MBCがB'001のとき、リモートフレームのIDは、応答メッセージの場合と同じく、データフレームのIDと完全に同じでなくてはならないため、IDEビットのフィルタは使用できません。</p> <p>2. 本機能を使用する場合、リモートフレームを受信するにもかかわらず、RTRビットはセットされません。リモートフレームを受信するとCPUはセット済みRFPRによって通知されますが、RCAN-TL1は現在のメッセージをデータフレームとして送信する必要があるためRTRビットは変更されません。</p> <p>3. リモートフレームの自動送信が開始されないこともあります。オーバランの状態(NMC=0のときのUMSRセット)に注意してください。</p> <p>4. ATX=1に設定されたメールボックスがリモートフレームによりオーバラン状態になった場合、旧メッセージの自動送信の要求が受け付けられる場合があります。</p> <p>0: データフレームの自動送信無効 1: データフレームの自動送信有効</p> <p>【注】 本ビットはメールボックス1~31にのみあります。メールボックス0ではリザーブビットとなります。</p>
		11	DART	<p>自動再送信無効</p> <p>このビットが1にセットされると、CANバスエラーのイベントが発生した場合や、CANバスのアービトレーションで負けた場合、メッセージの自動再送信を無効にします。実際に、本機能が使用されるときは、対応するTXCRビットが送信の最初に自動的にセットされます。このビットが0にセットされると、RCAN-TL1は、TXCRで送信リクエストをキャンセルされるか、送信が正常終了されるまで送信をリクエストします。</p> <p>0: 再送信有効 1: 再送信無効</p> <p>【注】 本ビットはメールボックス1~31にのみあります。メールボックス0ではリザーブビットとなります。</p>

レジスタ名	アドレス	ビット	ビット名	説明
MB[N]. CONTROL1	H'110+N*32	10~8	MBC[2:0]	<p>メールボックスコンフィギュレーション</p> <p>これらのビットは各メールボックスの機能を表 14.4 のように設定します。MBC が B'111 のとき、メールボックスはアクティブでなくなります。つまり TXPR やその他の設定にかかわらずメッセージの送受信を行いません。</p> <p>MBC を B'100、B'101、B'110 に設定することは禁止されています。MBC の値を B'000 とそれ以外に設定すれば、LAFM フィールドは使用可能となります。</p> <p>MBC が受信に設定されているとき、TXPR はセットしないでください。ハードウェア保護機能はなく、TXPR はセットされたままになります。メールボックス 0 の MBC[1]は受信専用のため、ハードウェアによって 1 に固定されています。</p>
		7~4	—	<p>リザーブビット</p> <p>読み出すと常に 0 が読み出されます。書き込む値は常に 0 にしてください。</p>
		3~0	DLC[3:0]	<p>データ長コード</p> <p>データフレームで送信されるデータのバイト数 (0~8) をエンコードします。</p> <p>0000 : データ長 0 バイト 0001 : データ長 1 バイト 0010 : データ長 2 バイト 0011 : データ長 3 バイト 0100 : データ長 4 バイト 0101 : データ長 5 バイト 0110 : データ長 6 バイト 0111 : データ長 7 バイト 1xxx : データ長 8 バイト</p> <p>【注】 x : Don't care</p>

【記号説明】 N : 0~31 (メールボックス番号)

表 14.4 メールボックスの機能の設定

MBC2	MBC1	MBC0	データ フレーム送信	リモート フレーム送信	データ フレーム受信	リモート フレーム受信	説明	
0	0	0	可	可	不可	不可	<ul style="list-style-type: none"> メールボックス 0 は使用不可能 タイムトリガ送信は使用可能 	
0	0	1	可	可	不可	可	<ul style="list-style-type: none"> ATX で使用可能* メールボックス 0 は使用不可能 LAFM は使用可能 	
0	1	0	不可	不可	可	可	<ul style="list-style-type: none"> メールボックス 0 は使用可能 LAFM は使用可能 	
0	1	1	不可	不可	可	不可	<ul style="list-style-type: none"> メールボックス 0 は使用可能 LAFM は使用可能 	
1	0	0	設定禁止					
1	0	1	設定禁止					
1	1	0	設定禁止					
1	1	1	メールボックスインアクティブ (初期値)					

【注】 * 自動再送信をサポートするために、MBC が B'001 で ATX が 1 のとき RTR は 0 にしてください。ATX が 1 の設定で使用するときは IDE のフィルタは使用しないでください。

14.3.3 ローカルアクセプタンスフィルタマスク (LAFM)

MBC が B'001、B'010、B'011 の値の場合、このフィールドは受信の LAFM として使用されます。LAFM はメールボックスが複数の受信 ID を受け入れることを許可します。LAFM は、[図 14.6](#) に示すとおり、2 つの 16 ビットの読み出し/書き込み可能なエリアから成ります。

レジスタ名	アドレス	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	アクセスサイズ	フィールド名
MB[N].LAFMH	H'104+N*32	IDE LAFM	0	0	STDID_LAFM[10:0]												EXTID LAFM[17:16]	16/32	LAFM
MB[N].LAFML	H'106+N*32	EXTID_LAFM[15:0]												16					

【記号説明】 N : 0~31 (メールボックス番号)

図 14.6 ローカルアクセプタンスフィルタマスク (LAFM)

1 つのビットが LAFM にセットされている場合、RCAN-TL1 が一致する CAN-ID でメールボックスを検索するとき、受信された CAN の ID の対応するビットは無視されます。ビットがクリアされている場合、受信された CAN の ID の対応するビットは格納されるメールボックスに設定されている STDID/IDE/EXTID と一致してはなりません。LAFM の構造はメールボックスのメッセージコントロールと同じです。この機能を使用しない場合は 0 で埋める必要があります。

- 【注】
1. RCAN-TL1 は、メールボックス 31 からメールボックス 0 まで、一致する ID の検索を行います。RCAN-TL1 は一致する ID を検知すると、そのメッセージは NMC や RXPR/RFPR フラグによらず、ただちに検索を終了します。これは、LAFM を使用していても受信メッセージは 1 つのメールボックスにのみ格納されることを示します。
 2. 1 つのメッセージが受信され一致するメールボックスが見つかったら、メッセージ全体がメールボックスに格納されます。LAFM を使用している場合、STDID、RTR、IDE、EXTID は受信されたメッセージの STDID、RTR、IDE、EXTID に更新されるので、受信前に設定されていたものと異なることがあります。

レジスタ名	アドレス	ビット	ビット名	説明
MB[N]. LAFMH	H'104+N*32	15	IDE_LAFM	IDE ビットのフィルタマスクビット 0 : 対応する IDE ビットが有効 1 : 対応する IDE ビットが無効
		14、13	—	リザーブビット 初期値は不定です。書き込む値は常に 0 にしてください。
		12~2	STDID_LAFM [10:0]	STDID[10:0]のフィルタマスクビット 0 : 対応する STDID ビットが有効 1 : 対応する STDID ビットが無効
		1、0	EXTID_LAFM [17:16]	EXTID[17:0]のフィルタマスクビット 0 : 対応する EXTID ビットが有効 1 : 対応する EXTID ビットが無効
MB[N]. LAFML	H'106+N*32	15~0	EXTID_LAFM [15:0]	1 : 対応する EXTID ビットが無効

【記号説明】 N : 0~31 (メールボックス番号)

14.3.4 メッセージデータフィールド

送受信される CAN メッセージを格納します。MSG_DATA_0 は送受信が行われる最初のデータバイトに対応します。CAN バス上のビットの並び順はビット 7 からビット 0 です。

• **タイムトリガ送信時の注意事項**

メールボックス 30 は、C_{MAX}≠B'111、MBC[30]=B'000 で TXPR[30]を 1 に設定すると、タイムリファレンス送信用に設定できます。この際、必ず DLC には 0 より大きい値を設定し、RTR は 0 に設定してください (TTCAN Level 1 にて規定)。メールボックス 30 の送信開始時に、データフィールドの先頭バイトの MSG_DATA_0[5:0]にサイクルカウンタ値 (CCR) が埋め込まれてメッセージを送信します。

MSG_DATA_0[7:6]は、メールボックスに格納された値が送信されます。

Next_is_Gap を送信する必要がある場合は、ユーザは MSG_DATA_0[7]を 1 に設定することで送信できます。

このとき、CCR の値は送信されるフレームに入りますが、メールボックス 30 のメッセージデータフィールドには格納されませんのでご注意ください。

メールボックス 31 は、C_{MAX}≠B'111、MBC[31]=B'011 で設定するとタイムリファレンス受信用に設定されます。有効なリファレンスメッセージを受信すると (ID 一致、DLC>0)、RCAN-TL1 は RFMK およびサイクルカウンタ (CCR) を書き換えることにより、Cycle_Time と Basic_Cycle を同期化します。

MB30, 31																	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
H'108+N*32	Next_is_Gap/Cycle_Counter (最初のRx/Txバイト)								MSG_DATA_1				8/16/32ビット				データ
H'10A+N*32	MSG_DATA_2								MSG_DATA_3				8/16ビット				
H'10C+N*32	MSG_DATA_4								MSG_DATA_5				8/16/32ビット				
H'10E+N*32	MSG_DATA_6								MSG_DATA_7				8/16ビット				
【記号説明】 N : 30, 31 (メールボックス番号)																	

図 14.7 メッセージデータフィールド

14.3.5 タイムスタンプ

送信/受信メッセージに記録されたタイムスタンプを格納します。タイムスタンプは、メッセージがスケジュールどおりに送信/受信されたかどうかモニタするのに役立ちます。

• **タイムスタンプ**

ビット	15	14	13	12	11	10	9	8
ビット名	TS15	TS14	TS13	TS12	TS11	TS10	TS9	TS8
初期値 :	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R
ビット	7	6	5	4	3	2	1	0
ビット名	TS7	TS6	TS5	TS4	TS3	TS2	TS1	TS0
初期値 :	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R

(1) メッセージ受信

メールボックス 0~15 の受信メッセージのタイムスタンプには、タイムトリガコントロールレジスタ 0 (TTCR0) のビット 14 の設定により、CYCTR[15:0] (サイクルタイムレジスタ) の値または CCR[5:0] (サイクルカウンタ) + CYCTR[15:6] (サイクルタイムレジスタ) の値が受信メッセージの SOF でキャプチャされ、タイムスタンプに格納されます。

メールボックス 30、31 の受信メッセージのタイムスタンプには、タイマカウンタレジスタ (TCNTR) の値が受信メッセージの SOF でキャプチャされ、タイムスタンプに格納されます。

(2) メッセージ送信

メールボックス 1~15 の送信メッセージのタイムスタンプには、タイムトリガコントロールレジスタ 0 (TTCR0) のビット 14 の設定により、CYCTR[15:0] (サイクルタイムレジスタ) の値または CCR[5:0] (サイクルカウンタ) + CYCTR[15:6] (サイクルタイムレジスタ) の値が送信メッセージの SOF でキャプチャされ、タイムスタンプに格納されます。

メールボックス 30、31 の送信メッセージのタイムスタンプには、タイマカウンタレジスタ (TCNTR) の値が送信メッセージの SOF でキャプチャされ、タイムスタンプに格納されます。

【重要】 タイムスタンプはテンポラリレジスタに格納されます。送信または受信が正常に行われると、その値がメールボックスのフィールドにコピーされます。また、オーバーランによって UMSR[N]がセットされるのと同時に CPU が RXPR[N]/RFPR[N]をクリアしたとき、タイムスタンプのみが更新されることがあります。したがって、RXPR[N]/RFPR[N]がクリアされる前の正しいタイムスタンプ値をリードすることができます。

【記号説明】 N : 0~15、30、31 (メールボックス番号)

14.3.6 送信トリガタイム (TTT) とタイムトリガコントロール

メールボックス 24~29 では、タイムトリガモード (C_{MAX}≠B'111) で MBC を B'000 に設定すると、送信トリガタイムはタイムウィンドウ間の境界を決定するための Time_Mark として機能します。

送信トリガタイム (TTT) とタイムトリガコントロール (TT コントロール) は、以下に示す 2 つの 16 ビットリード/ライトレジスタで構成されています。

メールボックス 30 にはタイムトリガコントロール (TT コントロール) がなく、Time_Ref として機能します。

メールボックス 24~30 をタイムトリガモード送信に使用しない場合は、受信用メールボックスとして使用することができます。ただし、タイムトリガモードを使用する場合は、イベントトリガ送信の対象にはなりません。

• 送信トリガタイム (TTT)

送信トリガタイム (TTT) は、サイクルタイム内でメッセージの送信を開始する時間を指定します。

ビット	15	14	13	12	11	10	9	8
ビット名	TTT15	TTT14	TTT13	TTT12	TTT11	TTT10	TTT9	TTT8
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

ビット	7	6	5	4	3	2	1	0
ビット名	TTT7	TTT6	TTT5	TTT4	TTT3	TTT2	TTT1	TTT0
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

• タイムトリガコントロール (TTコントロール)

タイムトリガコントロール (TT コントロール) は、ウィンドウの属性、送信を開始するシステムマトリックス内のサイクルカウント (Basic Cycle) および定周期送信の頻度を指定します。

ビット	15	14	13	12	11	10	9	8
ビット名	TTW1	TTW0	Offset5	Offset4	Offset3	Offset2	Offset1	Offset0
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

ビット	7	6	5	4	3	2	1	0
ビット名	0	0	0	0	0	rep_factor2	rep_factor1	rep_factor0
初期値:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

タイムトリガモードをサポートするすべてのメールボックス間の相違点を図 14.8 に示します。

MB24~29																	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
H'114+N*32	送信トリガタイム (サイクルタイム)														16ビット	トリガタイム	
H'116+N*32	TTW[1:0]		Offset[5:0]					0	0	0	0	0	rep_factor[2:0]		16ビット	TTコントロール	
MB30																	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
H'4D4	送信トリガタイム (サイクルタイム)														16ビット	トリガタイム	

【記号説明】 N : 24~29 (メールボックス番号)

図 14.8 送信トリガコントロールフィールド

• TTW[1:0] (タイムトリガウィンドウ)

タイムウィンドウの属性を示します。TTW=B'10 設定でスタートしたマージドアービトレイティングウィンドウ (Merged Arbitrating Window) は、必ず TTW=B'11 設定によって終了してください。マージドアービトレイティングウィンドウの先頭から最後まで間に TTW=B'10 設定のメッセージを数個使用することができます。

TTW[1]	TTW[0]	説明
0	0	エクスクルーシブウィンドウ : exclusive window (初期値)
0	1	アービトレイティングウィンドウ : arbitrating window
1	0	マージドアービトレイティングウィンドウの開始 : Start of Merged arbitrating window
1	1	マージドアービトレイティングウィンドウの終了 : End of Merged arbitrating window

TTT レジスタの値が CYCTR の値と一致し、Offset の値が CCR の値と一致すると、対応するメールボックスから送信を試みます。この機能を有効にするには、C_{MAX}≠B'111 に設定し、タイマ (TCNTR) を動作させ (TTCR0 のビット 15 が 1)、対応するメールボックスの MBC=B'000 に設定し、対応する TXPR のビットを 1 に設定してください。TXPR のビットをソフトウェアでセットした後は、定周期送信を続けるために、RCAN-TL1 はその TXPR のビット (メールボックス 24~30) をクリアしません。

定周期送信を停止するには、TXCR で TXPR をクリアしてください。この場合、送信完了直後に TXACK をクリアしないと、同じメールボックスの TXACK と ABACK の両方がセットされる可能性があります (図 14.9 参照)。

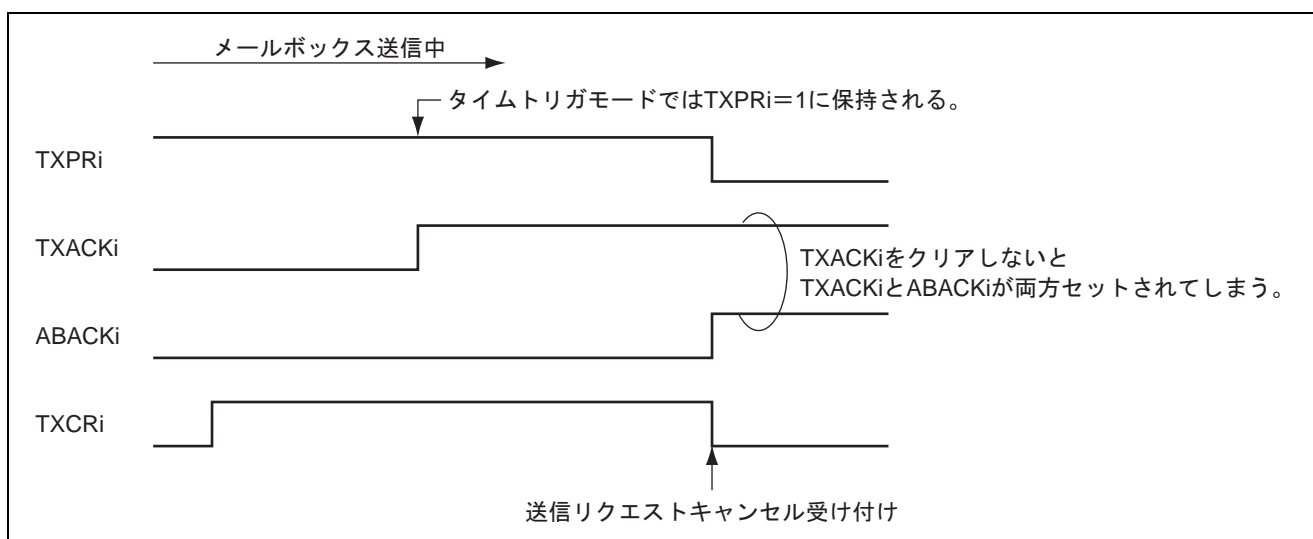


図 14.9 タイムトリガ送信時の TXACK と ABACK

なお、メールボックス 30 では TTW=B'01、Offset=B'000000、rep_factor=B'000 に固定です。以下の表に rep_factor と Offset の組み合わせを示します。

rep_factor	説明
B'000	サイクルカウントごと (初期値)
B'001	2 サイクルカウントごと
B'010	4 サイクルカウントごと
B'011	8 サイクルカウントごと
B'100	16 サイクルカウントごと
B'101	32 サイクルカウントごと
B'110	64 サイクルカウントごと (システムマトリックスごとに 1 回)
B'111	リザーブ

Offset フィールドは、タイムトリガメールボックスがメッセージの送信を開始できる最初のサイクルカウンタ値を決定します。

Offset	説明
B'000000	オフセット (Offset) = 最初のサイクルカウント (初期値)
B'000001	オフセット (Offset) = 2 番目のサイクルカウント
B'000010	オフセット (Offset) = 3 番目のサイクルカウント
B'000011	オフセット (Offset) = 4 番目のサイクルカウント
B'000100	オフセット (Offset) = 5 番目のサイクルカウント
...	...
B'111110	オフセット (Offset) = 63 番目のサイクルカウント
B'111111	オフセット (Offset) = 64 番目のサイクルカウント

以下の関係を満たさなければなりません。

$$\text{Cycle_Count_Maximum} + 1 \geq \text{Repeat_Factor} > \text{Offset}$$

$$\text{Cycle_Count_Maximum} = 2^{\text{C}_{\text{MAX}}} - 1$$

$$\text{Repeat_Factor} = 2^{\text{rep_factor}}$$

C_{MAX}、Repeat_Factor、および Offset はレジスタ値です。

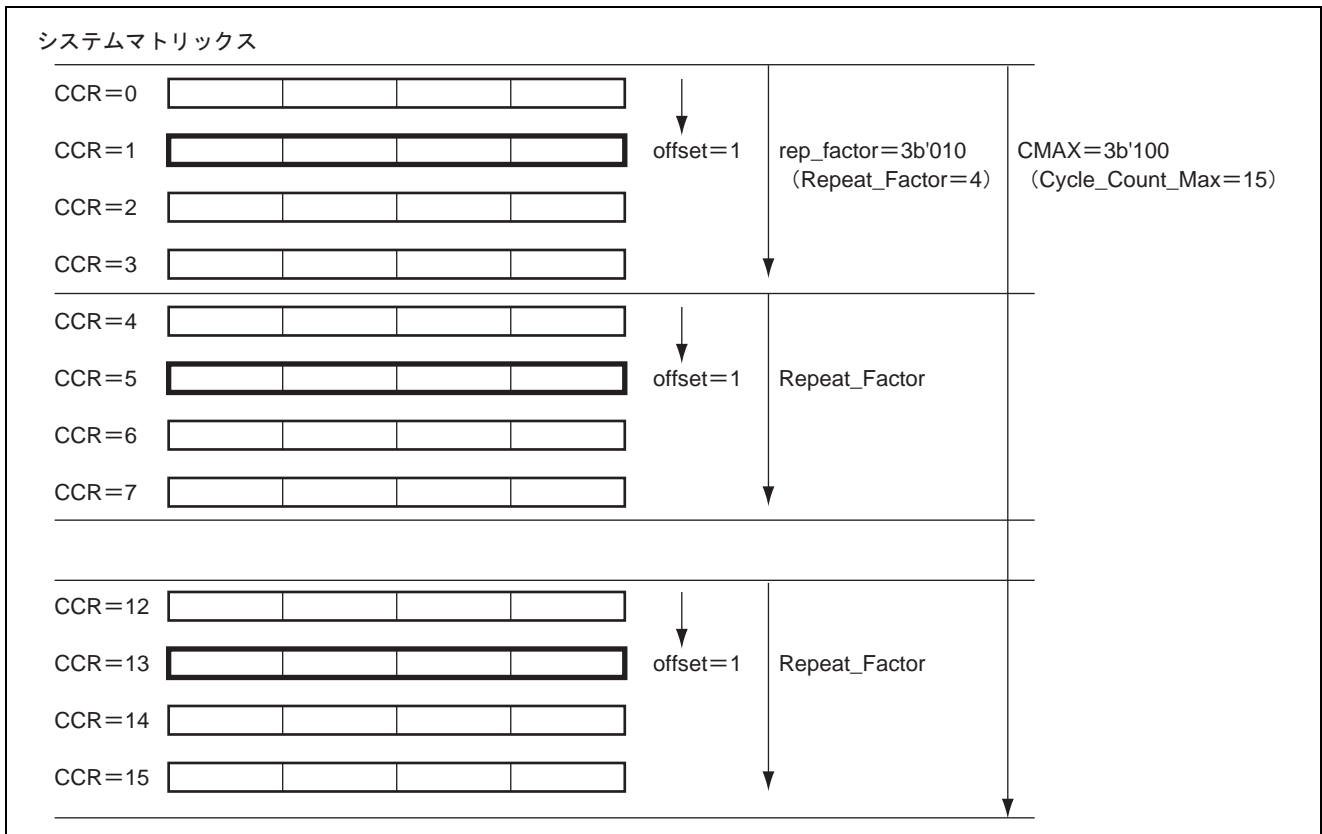


図 14.10 システムマトリックス

送信トリガタイムは昇順に設定してください。送信トリガタイム間の差は、 $\text{TTT}(\text{メールボックス } i) - \text{TTT}(\text{メールボックス } i-1) > \text{TEW} + \text{最大フレーム長} + 9$ を満たすように設定してください。

14.4 RCAN-TL1 のコントロールレジスタ

RCAN-TL1 のコントロールレジスタについて説明します。RCAN-TL1 のコントロールレジスタはワードサイズ (16 ビット) でのみアクセスできます。

表 14.5 に RCAN-TL1 のコントロールレジスタを示します。

表 14.5 RCAN-TL1 のコントロールレジスタの構成

レジスタ名	略称	アドレス	アクセスサイズ (ビット)
マスタコントロールレジスタ	MCR	H'000	16
ジェネラルステータスレジスタ	GSR	H'002	16
ビットコンフィギュレーションレジスタ 1	BCR1	H'004	16
ビットコンフィギュレーションレジスタ 0	BCR0	H'006	16
インタラプトリクエストレジスタ	IRR	H'008	16
インタラプトマスクレジスタ	IMR	H'00A	16
送信エラーカウンタ/受信エラーカウンタ	TEC/REC	H'00C	16

14.4.1 マスタコントロールレジスタ (MCR)

MCR は、16 ビットの読み出し/書き込み可能なレジスタで、RCAN-TL1 を制御します。

ビット	15	14	13	12	11	10	9	8
ビット名	MCR15	MCR14	—	—	—	TST2	TST1	TST0
初期値:	1	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R/W	R/W	R/W
ビット	7	6	5	4	3	2	1	0
ビット名	MCR7	MCR6	MCR5	—	—	MCR2	MCR1	MCR0
初期値:	0	0	0	0	0	0	0	1
R/W:	R/W	R/W	R/W	R	R	R/W	R/W	R/W

ビット	ビット名	初期値	R/W	説明
15	MCR15	1	R/W	<p>ID 並べ替え</p> <p>本ビットは HCAN2 と互換性のあるメッセージコントロールおよび LAFM の STDID、RTR、IDE、EXTID の順序を設定できます。なお本ビットはリセットモードでのみ変更可能です。ID 並び替えの順序については図 14.11 を参照してください。</p> <p>0: RCAN-TL1 と HCAN2 は同等の順序</p> <p>1: RCAN-TL1 と HCAN2 は異なる順序</p>

ビット	ビット名	初期値	R/W	説明
14	MCR14	0	R/W	<p>自動バスオフホルト</p> <p>本ビットと MCR6 がともに設定された場合、RCAN-TL1 がバスオフ状態に入ると MCR1 はただちに自動的にセットされます。本ビットはリセットモードでのみ変更可能です。</p> <p>0 : 通常の復帰シーケンス (128×11 レセシブビット) で RCAN-TL1 バスオフ状態を維持</p> <p>1 : MCR6 がセットされると RCAN-TL1 はバスオフ状態のあと、ただちにホルトモードに入ります</p>
13~11	—	すべて 0	R	<p>リザーブビット</p> <p>読み出すと常に 0 が読み出されます。書き込む値は常に 0 にしてください。</p>
10~8	TST[2:0]	000	R/W	<p>テストモード</p> <p>本ビットはテストモードを有効/無効にします。テストモードを起動する前に、RCAN-TL1 をホルトモードあるいはリセットモードにする必要がありますので注意してください。これは、テストモードへの遷移が進行中の送受信に影響することを回避するためです。詳細については「14.6.2 テストモードの設定」を参照してください。</p> <p>テストモードは診断およびテストのためだけに用いるもので、RCAN-TL1 が通常動作時には使用できません。</p> <p>000 : ノーマルモード</p> <p>001 : リスンオンリモード (受信専用モード)</p> <p>010 : セルフテストモード 1 (外部)</p> <p>011 : セルフテストモード 2 (内部)</p> <p>100 : ライトエラーカウンタ</p> <p>101 : エラーパッシブモード</p> <p>110 : 設定禁止</p> <p>111 : 設定禁止</p>
7	MCR7	0	R/W	<p>自動ウェイクモード</p> <p>本ビットは CAN スリープモードの自動ウェイクモードを有効/無効にします。本ビットが設定されると、RCAN-TL1 は CAN バスアクティビティ (ドミナントビット) を検出して自動的に CAN スリープモード (MCR5) を解除します。本ビットがクリアされると RCAN-TL1 は自動的に CAN スリープモードを解除しません。</p> <p>RCAN-TL1 は、RCAN-TL1 をウェイクアップしたメッセージを格納できません。</p> <p>0 : CAN バスのアクティビティによる自動ウェイクモードが無効</p> <p>1 : CAN バスのアクティビティによる自動ウェイクモードが有効</p> <p>【注】 CAN スリープモード中は MCR7 ビットを変更できません。</p>

ビット	ビット名	初期値	R/W	説明
6	MCR6	0	R/W	<p>バスオフ時ホルト</p> <p>本ビットは、バスオフ時に MCR1 がセットされるとただちにホルトモードに入ることを有効にするか無効にするかを設定します。本ビットはリセットあるいはホルトモードでのみ変更可能です。バスオフ時にホルトモードに入った場合、CAN コントローラもただちにエラーアクティブモードに復帰するので注意してください。</p> <p>0 : バスオフ時にホルトモードには入らず、復帰シーケンスが終了するのを待ちます</p> <p>1 : バスオフ時に MCR1 設定によるホルトモード遷移を有効にします</p>
5	MCR5	0	R/W	<p>CAN スリープモード</p> <p>本ビットは CAN スリープモードへの遷移を有効/無効にします。RCAN-TL1 がホルトモード時に本ビットがセットされると、CAN スリープモードへの遷移が有効になります。本ビットの設定は、ホルトモードに入った後に許可されます。2つのエラーカウンタ (REC、TEC) は CAN スリープモードの間は変化しません。CAN スリープモードを解除するには2つの方法があります。</p> <ul style="list-style-type: none"> • MCR5 ビットに 0 を書き込む • MCR7 が有効の場合、CAN バス上のドミナントビットを検出する <p>自動スリープ解除モードが無効であれば、CAN スリープモードが終了するまで RCAN-TL1 はすべての CAN バスの動作を無視します。CAN スリープモードを解除するときは、RCAN-TL1 は CAN バス動作に入る前に 11 レセツピビットをチェックすることで CAN バスと同期を取ります。これは 2 番目の方法が使用されているときに RCAN-TL1 が最初のメッセージを受信できないことを意味します。同様に CAN トランシーバもスタンバイモードを終了するときには最初のメッセージを受信できませんので、ソフトウェアはこの方法で設計する必要があります。</p> <p>CAN スリープモード中は MCR、GSR、IRR、IMR レジスタだけがアクセス可能です。詳細は「14.6.1 (3) CAN スリープモード」を参照してください。</p> <p>0 : CAN スリープモードが解除されています</p> <p>1 : CAN スリープモードへの遷移が有効です</p> <p>【注】 RCAN-TL1 は、最初にホルトモードに設定してから CAN スリープモードに遷移できます。これにより、CAN スリープモードに遷移する前に CPU が待ち状態のすべての割り込みをクリアすることができます。すべての割り込みがクリアされてしまうと同時に RCAN-TL1 はホルトモードから CAN スリープモードに遷移しなければなりません。(具体的には MCR5 に 1 を、MCR1 に 0 を同時にライトします。)</p>
4、3	—	すべて 0	R	<p>リザーブビット</p> <p>読み出すと常に 0 が読み出されます。書き込む値は常に 0 にしてください。</p>

ビット	ビット名	初期値	R/W	説明
2	MCR2	0	R/W	<p>メッセージ送信プライオリティ</p> <p>本ビットはペンディング中の送信データの送信順序を選択します。本ビットを1にセットした場合、送信データは送信待ちレジスタ (TXPR) のビット順で送信されます。</p> <p>送信はメールボックス 31 を最優先で開始し、メールボックス 1 まで続けます (メールボックスが送信用に設定されている場合)。</p> <p>本機能は、メールボックス 24~30 のタイムトリガ送信には使用できませんのでご注意ください。もし本ビットがクリアされると、すべての送信メッセージは (内部アービトレーションを走らせることにより) ID 優先順位どおり送信されます。最優先のメッセージは最小の数字のアービトレーションフィールド (STDID + IDE ビット + EXTID (IDE=1 の場合) + RTR ビット) を持ち、最初に送信されます。</p> <p>内部アービトレーションは RTR ビットと IDE ビットを含みます (内部アービトレーションは、2 つの CAN ノード間の CAN バス上にあるアービトレーションが同時に遷移を開始するのと同じ方法で動作します)。</p> <p>本ビットはリセットあるいはホルトモードでのみ変更可能です。</p> <p>0 : メッセージ ID 優先順に送信</p> <p>1 : メールボックス番号順 (メールボックス 31 → メールボックス 1) に送信</p>

ビット	ビット名	初期値	R/W	説明
1	MCR1	0	R/W	<p>ホルトリクエスト</p> <p>本ビットをセットすると CAN コントローラは現在の動作を終了したのちホルトモードに入ります（ここで CAN バスからは切り離されます）。RCAN-TL1 は本ビットがクリアされるまでホルトモードのままになります。</p> <p>ホルトモード中、CAN インタフェースは CAN バス動作に関係せず、メッセージの格納や送信も行いません。ホルト状態を CPU に通知するための IRR0 と GSR4 を除き、ユーザレジスタ（メールボックスの内容および TEC/REC を含みます）の内容は保持されます。</p> <p>CAN バスがアイドルまたはインタミッション状態の場合は、MCR6 にかかわりなく RCAN-TL1 は 1 ビット時間内にホルトモードになります。MCR6 がセットされていると、バスオフ中のホルトリクエストも 1 ビット時間内に動作します。それ以外ではバスオフ復帰シーケンスが完了するまではホルトモードに入りません。ホルトモードになると IRR0 と GSR4 によって通知されます。</p> <p>MCR14 と MCR6 がセットされていると、本ビットは RCAN-TL1 がバスオフ状態に遷移するとすぐに自動的にセットされます。</p> <p>ホルトモード中、RCAN-TL1 はバス動作に関係しないため、ビットタイミング設定を除き RCAN-TL1 の構成を変更することができます。CAN バス動作に再び参加するには、本ビットを 0 にクリアする必要があります。クリア後、RCAN-TL1 は 11 レセシブビットが検出されるまで待ち、CAN バスに参加します。</p> <p>0 : ホルトモードリクエストをクリア 1 : ホルトモード遷移リクエスト</p> <p>【注】 1. ホルトリクエストが発行された後は、CPU はホルトモードへの遷移が完了するまで (IRR0 と GSR4 で通知されます) TXPR と TXCR にアクセスおよび本ビットをクリアすることができません。 MCR1 がセットされた後、この状態はホルトモードに入ってからか、（ソフトウェアまたハードウェアでの）リセット動作でのみ解除できます。</p> <p>2. ホルトモードへのあるいはホルトモードからの遷移が可能なのは、BCR1 と BCR0 レジスタが適切なポーレートに設定されているときだけです。</p>

ビット	ビット名	初期値	R/W	説明
0	MCR0	1	R/W	<p>リセットリクエスト</p> <p>本ビットは RCAN-TL1 モジュールのリセットを制御します。本ビットが 0 から 1 に変わったとき、RCAN-TL1 コントローラはリセットルーチンに入り、内部ロジックを初期化して、リセットモードを通知するため GSR3 と IRR0 をセットします。すべてのユーザレジスタが初期化されます。</p> <p>本ビットがセットされている間、RCAN-TL1 は再構成することができます。CAN バスに参加するために本ビットは 0 をライトしてクリアする必要があります。クリア後、RCAN-TL1 モジュールは、11 レセシブビットの検出を待って CAN バスに参加します。</p> <p>CAN バス上の値をサンプリングするためにボーレートを適切な値にセットする必要があります。パワーオンリセット後には、このビットと GSR3 は常にセットされます。これはリセットがリクエストされ RCAN-TL1 を構成する必要があることを示します。</p> <p>リセットリクエストはパワーオンリセットに相当しますがソフトウェアで制御されます。</p> <p>0 : リセットモードリクエストをクリア [クリア条件] RCAN-TL1 をリセットした後に 0 が書き込まれたとき 1 : CAN インタフェースのリセットモード遷移リクエスト</p>

・ MCR15 (ID 並べ替え) = 0

アドレス	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	アクセスサイズ	フィールド名
H'100+N*32	0	STDID[10:0]											RTR	IDE	EXTID[17:16]	16/32	コントロール0	
H'102+N*32	EXTID[15:0]																	16
H'104+N*32	0	STDID_LAFM[10:0]											0	IDE_LAFM	EXTID_LAFM [17:16]	16/32	LAFMフィールド	
H'106+N*32	EXTID_LAFM[15:0]																	16

・ MCR15 (ID 並べ替え) = 1

アドレス	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	アクセスサイズ	フィールド名
H'100+N*32	IED	RTR	0	STDID[10:0]											EXTID[17:16]		16/32	コントロール0
H'102+N*32	EXTID[15:0]																16	
H'104+N*32	IDE_LAFM	0	0	STDID_LAFM[10:0]											EXTID_LAFM [17:16]		16/32	LAFMフィールド
H'106+N*32	EXTID_LAFM[15:0]																16	

【記号説明】 N : 0~31 (メールボックス番号)

図 14.11 ID 並べ替え

14.4.2 ジェネラルステータスレジスタ (GSR)

GSR は、16 ビットの読み出し専用レジスタで、RCAN-TL1 の状態を示します。

ビット	15	14	13	12	11	10	9	8
ビット名	—	—	—	—	—	—	—	—
初期値 :	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R
ビット	7	6	5	4	3	2	1	0
ビット名	—	—	GSR5	GSR4	GSR3	GSR2	GSR1	GSR0
初期値 :	0	0	0	0	1	1	0	0
R/W :	R	R	R	R	R	R	R	R

ビット	ビット名	初期値	R/W	説明
15~6	—	すべて 0	R	リザーブビット 読み出すと常に 0 が読み出されます。書き込む値は常に 0 にしてください。
5	GSR5	0	R	エラーパッシブステータス CAN インタフェースがエラーパッシブかどうかを示します。本ビットは RCAN-TL1 がエラーパッシブ状態になるとすぐに 1 にセットされ、再びエラーアクティブ状態になるとクリアされます。これは、エラーパッシブ中とバスオフ中は、GSR5 は 1 を保持することを意味します。したがって、正確な状態を知るには GSR5 と GSR0 の両方を調べてください。 0 : RCAN-TL1 はエラーパッシブあるいはバスオフ状態ではありません [クリア条件] RCAN-TL1 がエラーアクティブ状態の間 1 : RCAN-TL1 がエラーパッシブ (ただし GSR0=0 の場合) あるいはバスオフ (ただし GSR0=1 の場合) です [セット条件] TEC \geq 128 または REC \geq 128 またはテストモードでエラーパッシブモード選択時

ビット	ビット名	初期値	R/W	説明
4	GSR4	0	R	<p>ホルト/スリープステータス</p> <p>CAN コントローラがホルト/スリープかどうかを示します。本フラグのクリア時間は IRR12 のセッティング時間とは同じではありませんので留意してください。このフラグは CAN コントローラの状態を反映するもので、RCAN-TL1 の状態をフルに反映するものではありません。RCAN-TL1 は CAN スリープモードを終了し MCR5 がクリアされるとアクセス可能になります。CAN コントローラの CAN スリープモードは、転送クロック 2 ビット分経過後に終了します。</p> <p>0 : RCAN-TL1 はホルトモードでも CAN スリープモードでもありません 1 : ホルトモード (MCR1=1 の場合) または CAN スリープモード (MCR5=1 の場合) です</p> <p>[セット条件] MCR1 がセットされ CAN バスがインタミッションまたはアイドルの場合、または MCR5 がセットされ RCAN-TL1 がホルトモードのとき、または MCR14 と MCR6 が両者ともセットされて RCAN-TL1 がバスオフへ遷移したとき</p>
3	GSR3	1	R	<p>リセットステータス</p> <p>RCAN-TL1 がリセット状態かどうかを示します。</p> <p>0 : RCAN-TL1 はリセット状態ではありません 1 : RCAN-TL1 がリセット状態です</p> <p>[セット条件] RCAN-TL1 のソフトウェアまたはハードウェアリセットの後</p>
2	GSR2	1	R	<p>メッセージ送信中フラグ</p> <p>RCAN-TL1 がバスオフ状態にあるのか、メッセージを送信中なのかまたは送信中に検出されたエラー起因によるエラー/オーバーロードのフラグを送信中なのかを CPU に示すフラグです。TXACK をセットするタイミングと GSR2 をクリアするタイミングとは異なります。TXACK は EOF の 7 番目のビットにセットされます。GSR2 については、送信待ちのメッセージがない場合には、インタミッションの 3 番目のビットでセットされます。また、アービトレーションロスト、バスアイドル、受信、リセット、ホルト遷移でもセットされます。</p> <p>0 : RCAN-TL1 はバスオフ状態またはメッセージを送信中です 1 : [セット条件] バスオフ状態でない、またはメッセージを送信中でない</p>
1	GSR1	0	R	<p>送信/受信ワーニングフラグ</p> <p>エラーワーニングを示すフラグです。</p> <p>0 : [クリア条件] $TEC < 96$ かつ $REC < 96$ またはバスオフのとき 1 : [セット条件] $96 \leq TEC < 256$ または $96 \leq REC < 256$ のとき</p> <p>【注】 REC は、バスオフ復帰シーケンスに必要な 11 レセツピットの繰返し回数をカウントするために、バスオフ中は増加します。ただし、本ビットはバスオフ中にはセットされません。</p>

ビット	ビット名	初期値	R/W	説明
0	GSR0	0	R	バスオフフラグ RCAN-TL1 がバスオフ状態であることを示します。 0: [クリア条件] バスオフ状態から復帰またはハードウェアあるいはソフトウェアリセットの後 1: [セット条件] $TEC \geq 256$ (バスオフ状態)

14.4.3 ビットコンフィギュレーションレジスタ 0、1 (BCR0、BCR1)

BCR0、BCR1 は、それぞれ 16 ビットの読み出し/書き込み可能なレジスタで、CAN ビットタイミングパラメータと CAN インタフェースのボーレートプリスケアラを設定します。

以下、タイムクオンタを以下のとおり定義します。

$$\text{タイムクオンタ} = 2 \times \text{BRP} / f_{\text{clk}}$$

BRP (ボーレートプリスケアラ) は BCR0 格納値+1 の値です。 f_{clk} は周辺バスクロック周波数です。

(1) BCR1

TSEG1 と TSEG2 の設定については表 14.5 を参照してください。

ビット	15	14	13	12	11	10	9	8
ビット名	TSG1_3	TSG1_2	TSG1_1	TSG1_0	—	TSG2_2	TSG2_1	TSG2_0
初期値:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W
ビット	7	6	5	4	3	2	1	0
ビット名	—	—	SJW1	SJW0	—	—	—	BSP
初期値:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R	R/W

ビット	ビット名	初期値	R/W	説明
15~12	TSG1[3:0]	0000	R/W	<p>タイムセグメント 1</p> <p>これらのビットは、CAN バス上のエッジをポジティブフェーズエラーで補償するため、セグメント TSEG1 (=PRSEG + PHSEG1) を設定することに使用します。4~16 タイムカウンタが設定できます。</p> <p>0000 : 設定禁止 0001 : 設定禁止 0010 : 設定禁止 0011 : PRSEG + PHSEG1=4 タイムカウンタ 0100 : PRSEG + PHSEG1=5 タイムカウンタ : : 1111 : PRSEG + PHSEG1=16 タイムカウンタ</p>
11	—	0	R	<p>リザーブビット</p> <p>読み出すと常に 0 が読み出されます。書き込む値は常に 0 にしてください。</p>
10~8	TSG2[2:0]	000	R/W	<p>タイムセグメント 2</p> <p>これらのビットは、CAN バス上のエッジをネガティブフェーズエラーで補償するため、セグメント TSEG2 (=PRSEG2) を設定することに使用します。2~8 タイムカウンタが設定できます。</p> <p>000 : 設定禁止 001 : PHSEG2=2 タイムカウンタ (条件によっては設定禁止です。表 14.6 を参照してください) 010 : PHSEG2=3 タイムカウンタ 011 : PHSEG2=4 タイムカウンタ 100 : PHSEG2=5 タイムカウンタ 101 : PHSEG2=6 タイムカウンタ 110 : PHSEG2=7 タイムカウンタ 111 : PHSEG2=8 タイムカウンタ</p>
7、6	—	すべて 0	R	<p>リザーブビット</p> <p>読み出すと常に 0 が読み出されます。書き込む値は常に 0 にしてください。</p>
5、4	SJW[1:0]	00	R/W	<p>再同期ジャンプ幅</p> <p>同期ジャンプ幅を設定します。</p> <p>00 : 同期ジャンプ幅=1 タイムカウンタ 01 : 同期ジャンプ幅=2 タイムカウンタ 10 : 同期ジャンプ幅=3 タイムカウンタ 11 : 同期ジャンプ幅=4 タイムカウンタ</p>
3~1	—	すべて 0	R	<p>リザーブビット</p> <p>読み出すと常に 0 が読み出されます。書き込む値は常に 0 にしてください。</p>

ビット	ビット名	初期値	R/W	説明
0	BSP	0	R/W	ビットサンプルポイント データがサンプリングされるポイントを設定します。 0 : 1 か所でビットサンプリングが行われます (タイムセグメント1の最後) 1 : 3 か所でビットサンプリングが行われます (PHSEG1の最終3クロックサイクルの立ち上がりエッジ)

(2) BCR0

ビット	15	14	13	12	11	10	9	8
ビット名	—	—	—	—	—	—	—	—
初期値 :	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R
ビット	7	6	5	4	3	2	1	0
ビット名	BRP7	BRP6	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0
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

ビット	ビット名	初期値	R/W	説明
15~8	—	すべて0	R	リザーブビット 読み出すと常に0が読み出されます。書き込む値は常に0にしてください。
7~0	BRP[7:0]	0000000	R/W	ポーレートプリスケアラ これらのビットは、1タイムクオンタに対応する周辺バスクロック数を設定します。 0000000 : 2×周辺バスクロック 0000001 : 4×周辺バスクロック 0000010 : 6×周辺バスクロック : 2×(レジスタ値+1)×周辺バスクロック 1111111 : 512×周辺バスクロック

- ビットコンフィギュレーションレジスタについて

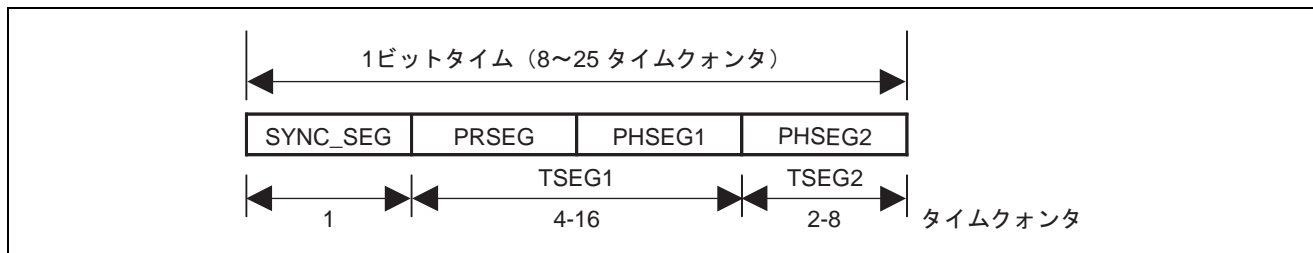


図 14.12 1 ビットタイムの構成

SYNC_SEG : CAN バス上のノードの同期をするセグメント

(通常のビットエッジ遷移がこのセグメントで発生します)

PRSEG : ネットワーク間の物理的な遅延を調整するセグメント

PHSEG1 : フェーズドリフト (正方向) のバッファセグメント (同期または再同期の際、拡張されます)

PHSEG2 : フェーズドリフト (負方向) のバッファセグメント (同期または再同期の際、縮小されます)

TSEG1 : TSG1 + 1

TSEG2 : TSG2 + 1

RCAN-TL1 ビットレートは以下のとおり計算されます。

$$\text{ビットレート} = f_{clk} / (2 \times (\text{BRP} + 1) \times (\text{TSEG1} + \text{TSEG2} + 1))$$

BRP はレジスタ値で与えられ、また TSEG1 および TSEG2 は上述の表から算出した値であって実際の設定値ではありません。前記ビットレート計算式のタイムセグメントの「+1」は SYNC_SEG の 1 タイムクオンタであることによります。

f_{clk} = 周辺バスクロック

BCR 設定上の制約となる事項

$$\text{TSEG1 (Min.)} > \text{TSEG2} \geq \text{SJW (Max.)} \quad (\text{SJW} = 1 \sim 4)$$

$$8 \leq \text{TSEG1} + \text{TSEG2} + 1 \leq 25 \text{ タイムクオンタ} \quad (\text{TSEG1} + \text{TSEG2} + 1 = 7 \text{ は不可})$$

$$\text{TSEG2} \geq 2$$

ビットコンフィギュレーションレジスタの TSEG1 および TSEG2 の値が表 14.6 に示される設定値の範囲であれば、上述の制限事項を満たします。表 14.6 の値は SJW の設定可能値を示したものです。「不可」で示されたところは TSEG1 と TSEG2 との組み合わせがとれません。

表 14.6 TSG と TSEG の設定

		001	010	011	100	101	110	111	TSG2
		2	3	4	5	6	7	8	TSEG2
TSG1	TSEG1								
0011	4	不可	1-3	不可	不可	不可	不可	不可	
0100	5	1-2	1-3	1-4	不可	不可	不可	不可	
0101	6	1-2	1-3	1-4	1-4	不可	不可	不可	
0110	7	1-2	1-3	1-4	1-4	1-4	不可	不可	
0111	8	1-2	1-3	1-4	1-4	1-4	1-4	不可	
1000	9	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1001	10	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1010	11	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1011	12	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1100	13	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1101	14	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1110	15	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1111	16	1-2	1-3	1-4	1-4	1-4	1-4	1-4	

例 1 : *fcik* が 40MHz でビットレートを 500kbps とする場合、BRP = 3、TSEG1 = 6、TSEG2 = 3 が条件を満たします。この場合、BCR1 には H'5200、BCR0 には H'0003 をライトすることになります。

例 2 : *fcik* が 35MHz でビットレートを 250kbps とする場合、BRP = 4、TSEG1 = 8、TSEG2 = 5 が条件を満たします。この場合、BCR1 には H'7400、BCR0 には H'0004 をライトすることになります。

例 3 : *fcik* が 32MHz でビットレートを 500kbps とする場合、BRP = 1、TSEG1 = 11、TSEG2 = 4 が条件を満たします。この場合、BCR1 には H'A300、BCR0 には H'0001 をライトすることになります。

例 4 : *fcik* が 20MHz でビットレートを 500kbps とする場合、BRP = 1、TSEG1 = 6、TSEG2 = 3 が条件を満たします。この場合、BCR1 には H'5200、BCR0 には H'0001 をライトすることになります。

14.4.4 インタラプトリクエストレジスタ (IRR)

IRR は、16 ビットの読み出し/書き込み可能なレジスタで、各種割り込み要因のステータスフラグで構成されています。

ビット	15	14	13	12	11	10	9	8
ビット名	IRR15	IRR14	IRR13	IRR12	IRR11	IRR10	IRR9	IRR8
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	R
ビット	7	6	5	4	3	2	1	0
ビット名	IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0
初期値 :	0	0	0	0	0	0	0	1
R/W :	R/W	R/W	R/W	R/W	R/W	R	R	R/W

ビット	ビット名	初期値	R/W	説明
15	IRR15	0	R/W	タイマコンペアマッチ割り込み 1 タイマコンペアマッチレジスタ 1 (TCMR1) のコンペアマッチが発生したことを示します。TCMR1 の設定値がサイクルタイムと一致すると (TCMR1=CYCTR)、本ビットがセットされます。 0 : TCMR1 のタイマコンペアマッチが発生していない [クリア条件] 1 を書き込む 1 : TCMR1 のタイマコンペアマッチが発生 [セット条件] TCMR1 がサイクルタイムと一致 (TCMR1=CYCTR)
14	IRR14	0	R/W	タイマコンペアマッチ割り込み 0 タイマコンペアマッチレジスタ 0 (TCMR0) のコンペアマッチが発生したことを示します。TCMR0 の設定値がローカルタイムと一致すると (TCMR0=TCNTR)、本ビットがセットされます。 0 : TCMR0 のタイマコンペアマッチが発生していない [クリア条件] 1 を書き込む 1 : TCMR0 のタイマコンペアマッチが発生 [セット条件] TCMR0 がタイム値と一致 (TCMR0=TCNTR)

ビット	ビット名	初期値	R/W	説明
13	IRR13	0	R/W	<p>タイマオーバーラン割り込み/Next_is_Gap 受信割り込み/メッセージエラー割り込み</p> <p>本割り込みは RCAN-TL1 のモードにより異なった意味を持ち、以下の状態を示します。</p> <ul style="list-style-type: none"> ・ イベントトリガモード (テストモードを含む) で動作中、タイマ (TCNTR) オーバーランが発生。 ・ タイムトリガモード (テストモードを含む) で動作中、Next_is_Gap がセットされたタイムリファレンスメッセージを受信。 ・ テストモードでメッセージエラーが発生。ただし、テストモード中にメッセージオーバーロード条件が発生しても、本ビットはセットされません。 <p>0: イベントトリガモード (テストモードを含む) でタイマ (TCNTR) オーバーランが発生していない タイムトリガモード (テストモードを含む) で Next_is_Gap 付きタイムリファレンスメッセージを受信していない テストモードでメッセージエラーが発生していない</p> <p>[クリア条件] 1 を書き込む</p> <p>1: [セット条件] イベントトリガモード(テストモードを含む)でタイマ (TCNTR) オーバーランが発生し、H'FFFF から H'0000 に変化 タイムトリガモード (テストモードを含む) で Next_is_Gap 付きタイムリファレンスメッセージを受信 テストモードでメッセージエラーが発生</p>
12	IRR12	0	R/W	<p>CAN スリープモード時バスアクティビティ</p> <p>本ビットは CAN バスアクティビティの存在を示します。RCAN-TL1 が CAN スリープモードのとき、CAN バス上でドミナントビットを検出すると、本ビットがセットされます。本ビットに 1 を書き込むと割り込みがクリアされます。0 を書き込むと無視されます。オートウェイクアップ機能を使用せず本割り込みを受け付けられない場合は、対応するマスクビットで必ず本割り込みを無効にしてください。オートウェイクアップ機能不使用時に本割り込みが要求された場合は、CAN スリープモードから復帰した後に本ビットをクリアしてください。これは、受信ライン上の新しい立ち下がりエッジにより、再び割り込みがセットされるのを防ぐためです。</p> <p>本割り込みのセット時間は、GSR4 のクリア時間と異なりますので注意してください。</p> <p>0: バスアイドル状態</p> <p>[クリア条件] 1 を書き込む</p> <p>1: CAN バスアクティビティを CAN スリープモード中に検出</p> <p>[セット条件]CAN スリープモード中に CRxn 上でドミナントへのビット状態変化を検出</p>

ビット	ビット名	初期値	R/W	説明
11	IRR11	0	R/W	<p>タイマコンペアマッチ割り込み 2</p> <p>タイマコンペアマッチレジスタ 2 (TCMR2) のコンペアマッチが発生したことを示します。TCMR2 の設定値がサイクルタイムと一致すると (TCMR2=CYCTR)、本ビットがセットされます。</p> <p>0: TCMR2 のタイマコンペアマッチが発生していない</p> <p>[クリア条件] 1 を書き込む</p> <p>1: TCMR2 のタイマコンペアマッチが発生</p> <p>[セット条件] TCMR2 がサイクルタイムと一致 (TCMR2=CYCTR)</p>
10	IRR10	0	R/W	<p>スタートシステムマトリックス割り込み</p> <p>次のシステムマトリックスが開始したことを示します。CCR=0 のとき、タイムリファレンスメッセージの送信/受信完了時に本ビットがセットされます。CMAx=0 の場合は、本割り込みはサイクルカウント (Basic Cycle) ごとにセットされます。</p> <p>0: 新しいシステムマトリックスの先頭でない</p> <p>[クリア条件] 1 を書き込む</p> <p>1: サイクルカウンタが 0 になった</p> <p>[セット条件]</p> <p>CMAx!=B'111 かつ CCR=0 のとき、タイムリファレンスメッセージの送信/受信が完了</p>
9	IRR9	0	R	<p>メッセージオーバーラン/オーバーライト割り込みフラグ</p> <p>本ビットは、メッセージを受信したが、対応する RXPR または RFPR がすでに 1 にセットされているためメールボックス内のメッセージが読み出されず、CPU によってクリアされていないことにより新しく受信されたメッセージは NMC ビットの設定によって、捨てられる (オーバーラン) か上書き (オーバーライト) されたことを示します。本ビットは UMSR レジスタの対応するビットに 1 を書き込むとクリアされます。また、MBIMR の対応するすべてのビットに 1 を書き込むことでもクリアできます。本ビットに対する書き込みは無効です。</p> <p>0: メッセージオーバーラン/オーバーライト通知がない</p> <p>[クリア条件] UMSR のすべてのビットがクリアまたはセットされたすべての UMSR に対応する MBIMR がセット</p> <p>1: オーバーランが発生したため、受信メッセージが捨てられた。またはメッセージがオーバーライトされた</p> <p>[セット条件] 対応する PXPR または RFPR=1 かつ MBIMR=0 のときにメッセージを受信</p>

ビット	ビット名	初期値	R/W	説明
8	IRR8	0	R	<p>メールボックスエンプティ割り込みフラグ</p> <p>送信用に設定されたメッセージの1つが正常に送信（対応する TXACK フラグがセット）または送信アボート（送信キャンセルが実行されたメッセージに対応する ABACK フラグがセット）されると、本ビットがセットされます。</p> <p>このときイベントトリガモードでは、対応する TXPR がクリアされ、メールボックスが次の送信用のメッセージデータを受け入れられるようになります。</p> <p>タイムトリガモードでは、プログラムされたサイクルカウント（Basic Cycle）ごとの送信を続けるため、メールボックス 24～30 の TXPR は送信完了後にクリアされません。</p> <p>実際には、本ビットは MBIMR フラグによってマスクされていない TXACK と ABACK ビットの OR を取った信号によってセットされます。したがって、すべての TXACK および ABACK ビットがクリアされると、本ビットは自動的にクリアされます。また、MBIMR の対応するすべてのビットに 1 を書き込むことでもクリアできます。本ビットに対する書き込みは無効です。</p> <p>0：送信または送信キャンセルするメッセージが処理中でない</p> <p>〔クリア条件〕 すべての TXACK および ABACK ビットがクリアまたは、セットされたすべての TXACK および ABACK ビットに対応する MBIMR がセット</p> <p>1：メッセージが送信または送信アボート（送信キャンセル）され、次のメッセージの格納が可能となった(タイムトリガモードでは、メールボックス 24～30 はアボート時のみ次のメッセージを格納可能)</p> <p>〔セット条件〕 TXACK または ABACK ビットがセットされたとき（対応する MBIMR=0 の場合）</p>
7	IRR7	0	R/W	<p>オーバーロードフレーム</p> <p>RCAN-TL1 がオーバーロードフレーム送信を検出したことを示すフラグです。IRR7 に 1 を書き込むとクリアされます。0 を書き込むと無視されます。</p> <p>0：〔クリア条件〕 1 を書き込む</p> <p>1：〔セット条件〕 オーバロード条件を検出</p>

ビット	ビット名	初期値	R/W	説明
6	IRR6	0	R/W	<p>バスオフ割り込みフラグ</p> <p>本ビットは、RCAN-TL1 がバスオフ状態になったとき、またはバスオフ状態からエラーアクティブ状態に戻ったときセットされます。したがって、ノードの $TEC \geq 256$、バスオフ復帰シーケンスの終了 (11 レセッシブビットを 128 回連続受信) またはバスオフから停止状態への遷移 (オートまたはマニュアル) が要因となります。</p> <p>本ビットは RCAN-TL1 がバスオフ解除となった後もセットされた状態となりますので、ソフトウェアでクリアする必要があります。ソフトウェアで GSR0 をリードして、RCAN-TL1 がバスオフ状態かエラーアクティブ状態か判定してください。本ビットはノードがまだバスオフ状態であっても 1 を書き込むとクリアされます。0 を書き込むと無視されます。</p> <p>0: [クリア条件] 1 を書き込む</p> <p>1: 送信エラーによりバスオフ状態に遷移した、またはバスオフ状態からエラーアクティブ状態に復帰した</p> <p>[セット条件] $TEC \geq 256$ または 11 レセッシブビットを 128 回連続受信した後バスオフ終了またはバスオフから停止状態へ遷移</p>
5	IRR5	0	R/W	<p>エラーパッシブ割り込みフラグ</p> <p>受信または送信エラーカウンタによるエラーパッシブ状態、またはテストモードによって強制的にエラーパッシブ状態になったことを示します。本ビットに 1 を書き込むとクリアされ、0 を書き込むと無視されます。本ビットがクリアされてもノードはエラーパッシブ状態のままとなる場合があります。ソフトウェアで GSR0 および GSR5 をチェックし、RCAN-TL1 がエラーパッシブ状態であるかバスオフ状態であるか判定する必要があります。</p> <p>0: [クリア条件] 1 を書き込む</p> <p>1: 送信/受信エラーによるエラーパッシブ状態</p> <p>[セット条件] $TEC \geq 128$ または $REC \geq 128$ またはエラーパッシブテストモードを使用</p>
4	IRR4	0	R/W	<p>受信エラーカウンタワーニング割り込みフラグ</p> <p>RCAN-TL1 がバスオフ状態でないときに受信エラーカウンタ (REC) が 95 を超えるとセットされます。本ビットに 1 を書き込むと割り込みはクリアされ、0 を書き込むと無視されます。</p> <p>0: [クリア条件] 1 を書き込む</p> <p>1: 受信エラーによるエラーワーニング状態</p> <p>[セット条件] RCAN-TL1 がバスオフ状態以外で $REC \geq 96$</p>

ビット	ビット名	初期値	R/W	説明
3	IRR3	0	R/W	<p>送信エラーカウンタワーニング割り込みフラグ</p> <p>RCAN-TL1 がバスオフ状態でないときに送信エラーカウンタ (TEC) が 95 を超えるとセットされます。本ビットに 1 を書き込むと割り込みはクリアされ、0 を書き込むと無視されます。</p> <p>0: [クリア条件] 1 を書き込む 1: 送信エラーによるエラーワーニング状態</p> <p>[セット条件] TEC ≥ 96</p>
2	IRR2	0	R	<p>リモートフレーム受信割り込みフラグ</p> <p>メールボックスがリモートフレームを受信したことを示します。対応する MBIMR がセットされていない少なくとも 1 つのメールボックスに、リモートフレーム送信リクエストが格納されているとセットされます。本ビットは、リモートフレーム受信完了レジスタ (RFPR) のすべてのビットがクリアされるとクリアされます。また、MBIMR の対応するすべてのビットに 1 を書き込むとクリアされます。本ビットに対する書き込みは無効です。</p> <p>0: [クリア条件] RFPR のすべてのビットがクリア 1: 少なくとも 1 つのリモートフレームリクエストが処理待ち状態</p> <p>[セット条件] 対応する MBIMR=0 のときリモートフレームを受信</p>
1	IRR1	0	R	<p>データフレーム受信割り込みフラグ</p> <p>処理待ち状態の受信データフレームが存在することを示します。本ビットが 1 のとき、少なくとも 1 つのメールボックスに未処理のメッセージが格納されています。本ビットは、データフレーム受信完了レジスタ (RXPR) のすべてのビットがクリアされると (どの受信メールボックスにも未処理のメッセージがない) クリアされます。MBIMR が 0 でない各受信メールボックスの RXPR フラグの論理和が設定されます。</p> <p>また、MBIMR の対応するすべてのビットに 1 を書き込むとクリアされます。本ビットに対する書き込みは無効です。</p> <p>0: [クリア条件] RXPR のすべてのビットがクリア 1: データフレームを受信しメールボックスに格納した</p> <p>[セット条件] 対応する MBIMR=0 のときデータを受信</p>

ビット	ビット名	初期値	R/W	説明
0	IRR0	1	R/W	<p>リセット/ホルト/スリープ割り込みフラグ</p> <p>下記の3つの要因によってセットされます。</p> <ul style="list-style-type: none"> ソフトウェアリセット (MCR0) またはハードウェアリセット後、リセットモードに遷移 ホルトリクエスト (MCR1) の後、ホルトモードに遷移 ホルトモードでスリープリクエスト (MCR5) 発行後、CAN スリープモードに遷移 <p>本ビットがセットされた後 GSR をリードして、RCAN-TL1 の状態を知ることができます。</p> <p>【注】スリープモードリクエストを発行する必要がある場合、事前にホルトモードを使用してください。MCR5 の説明および図 14.17 を参照してください。</p> <p>IRR0 は GSR3 または GSR4 が 0 から 1 に変化したとき、またはホルトモードから CAN スリープモードに遷移したときにセットされます。したがって、RCAN-TL1 がホルトモード解除の直後に再びホルトモードに遷移した場合は GSR4 がクリアされないため、IRR0 はセットされません。同様に、CAN スリープモードからホルトモードに直接遷移した場合も IRR0 はセットされません。ホルトモード/CAN スリープモードから送信/受信動作に遷移する際、GSR4 がクリアされるまでに[1 ビット時間-TSEG2]~[1 ビット時間×2-TSEG2] の時間が必要です。</p> <p>リセットモードの場合、IRR0 はセットされますが初期化によって IMR0 が自動的にセットされるため、CPU への割り込みはアサートされません。</p> <p>0: [クリア条件] 1 を書き込む</p> <p>1: ソフトウェアリセットモードまたはホルトモードまたは CAN スリープモードへ遷移</p> <p>[セット条件] リセット (MCR0 またはハードウェア) またはホルトモード (MCR1) または CAN スリープモード (MCR5) リクエストの後、リセット/ホルト/CAN スリープモードへの遷移が完了</p>

14.4.5 インタラプトマスクレジスタ (IMR)

IMR は、16 ビットの読み出し/書き込み可能なレジスタで、インタラプトリクエストレジスタ (IRR) の各ビットに対応する割り込みの IRQ 出力信号の生成をマスクします。ビットを 1 に設定すると対応する割り込み要求がマスクされます。IMR は IRQ の生成を直接制御しますが、IRR の対応するビットのセットを禁止しません。

ビット	15	14	13	12	11	10	9	8
ビット名	IMR15	IMR14	IMR13	IMR12	IMR11	IMR10	IMR9	IMR8
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
ビット	7	6	5	4	3	2	1	0
ビット名	IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	IMR0
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

ビット	ビット名	初期値	R/W	説明
15~0	IMR[15:0]	すべて 1	R/W	IRR[15:0]の各ビットに対応する割り込み要因マスク 本ビットをセットすると、IRR の対応するビットがセットされてもその割り込み信号は生成されません。 0: 対応する IRR をマスクしない (割り込み要因が発生すると IRQ が生成される) 1: IRR の対応する割り込みをマスクする

14.4.6 送信エラーカウンタ (TEC) / 受信エラーカウンタ (REC)

TEC および REC は、読み出し/条件付き書き込み可能な 16 ビットのレジスタで、CAN インタフェースにおける送信/受信メッセージエラー数を示すカウンタです。カウント値は前述した参考文献の[1]、[2]、[3]、[4]に規定されています。ライトエラーカウンタテストモード以外では本レジスタはリード専用で、CAN インタフェースによってのみ書き替え可能です。本レジスタは、リセットリクエスト (MCR0) またはバスオフ状態への遷移によってクリアされます。

ライトエラーカウンタテストモード (TST[2:0]=B'100) では、本レジスタへの書き込みが可能です。TEC および REC には同じ値しか書き込めません。TEC に書き込まれた値は TEC および REC に設定されます。本レジスタに書き込む際は、RCAN-TL1 をホルトモードにする必要があります。

ビット	15	14	13	12	11	10	9	8
ビット名	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0
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*
ビット	7	6	5	4	3	2	1	0
ビット名	REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0
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】 * MCR の TST[2:0]=B'100 によるテストモードのときのみ書き込み可能です。REC はバスオフ状態において、バスオフ復帰シーケンスに必要な 11 レセツピビットの受信回数をカウントします。

ビット	ビット名	初期値	R/W	説明
15~8	TEC[7:0]	すべて 0	R/W*	送信エラーカウンタ 送信中に CAN 仕様に基づくエラーが検出されると本レジスタの値はインクリメントされます。
7~0	REC[7:0]	すべて 0	R/W*	受信エラーカウンタ 受信中に CAN 仕様に基づくエラーが検出されると本レジスタの値はインクリメントされます。

14.5 RCAN-TL1 のメールボックスレジスタ

RCAN-TL1 のメールボックスレジスタについて説明します。RCAN-TL1 のメールボックスレジスタは、各メールボックスを制御し、メールボックスの状態を示します。

表 14.7 に RCAN-TL1 のメールボックスレジスタを示します。

【重要】 ロングワードアクセスは、2つの連続したワードアクセスとして行われます。

表 14.7 RCAN-TL1 のメールボックスレジスタ

レジスタ名	略称	アドレス	アクセスサイズ (ビット)
送信待ちレジスタ 1	TXPR1	H'020	32
送信待ちレジスタ 0	TXPR0	H'022	—
		H'024	
		H'026	
送信キャンセルレジスタ 1	TXCR1	H'028	16/32
送信キャンセルレジスタ 0	TXCR0	H'02A	16
		H'02C	
		H'02E	
送信アクリッジレジスタ 1	TXACK1	H'030	16/32
送信アクリッジレジスタ 0	TXACK0	H'032	16
		H'034	
		H'036	
アボートアクリッジレジスタ 1	ABACK1	H'038	16/32
アボートアクリッジレジスタ 0	ABACK0	H'03A	16
		H'03C	
		H'03E	
データフレーム受信完了レジスタ 1	RXPR1	H'040	16/32
データフレーム受信完了レジスタ 0	RXPR0	H'042	16
		H'044	
		H'046	
リモートフレーム受信完了レジスタ 1	RFPR1	H'048	16/32
リモートフレーム受信完了レジスタ 0	RFPR0	H'04A	16
		H'04C	
		H'04E	
メールボックスインタラプトマスクレジスタ 1	MBIMR1	H'050	16/32
メールボックスインタラプトマスクレジスタ 0	MBIMR0	H'052	16
		H'054	
		H'056	
未読メッセージステータスレジスタ 1	UMSR1	H'058	16/32

レジスタ名	略称	アドレス	アクセスサイズ (ビット)
未読メッセージステータスレジスタ 0	UMSR0	H'05A	16
		H'05C	
		H'05E	

14.5.1 送信待ちレジスタ 1、0 (TXPR1、TXPR0)

TXPR1 と TXPR0 は連結され、CAN モジュールの送信待ちフラグを格納する 32 ビットのレジスタを構成します。16 ビットバスインタフェースの場合、ロングワードアクセスは、2 つの連続したワードアクセスとして行われます。

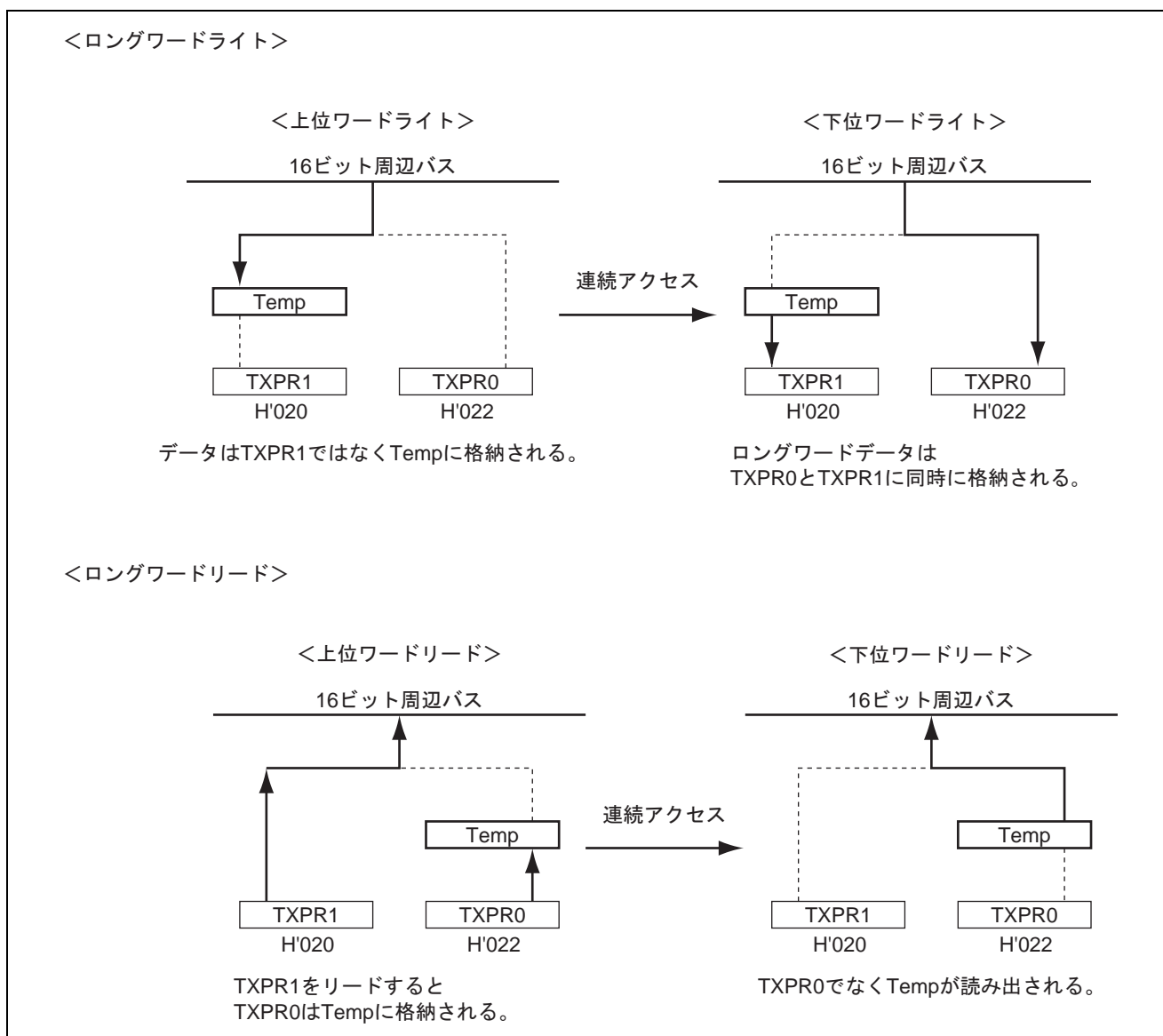


図 14.13 16 ビットバスインタフェース時のロングワードアクセス

TXPR1 レジスタはメールボックス 31~16 を制御し、TXPR0 レジスタはメールボックス 15~1 を制御します。CPU はビットに 1 を書き込むことで対応する送信メッセージに対して操作することができます。0 書き込みは無効で、TXPR をクリアするには TXCR の対応するビットをセットしなければなりません。CPU は TXPR をリードして、送信待ちのメールボックスおよび送信中のメールボックスを知ることができます。実際には、メールボックス 0 以外のすべてのメールボックスについて送信待ちビットが存在します。また、送信に設定されていないメールボックスに対するビットに 1 を書き込むことは禁止されています。

イベントトリガモードでは、メッセージが正常に送信された後または TXCR からの送信アボートが行われた後、RCAN-TL1 は対応する送信待ちフラグをクリアします。タイムトリガモードでは、プログラムされたサイクルカウント (Basic Cycle) ごとの送信を続けるため、メールボックス 30~24 の TXPR は送信完了後にクリアされません。TXPR のフラグは、CAN ノードがアービトレーションに負けたり CAN バス上にエラーが発生したために、メッセージが送信されなかった場合はクリアされません。このとき対応するメールボックスのメッセージコントロールフィールドの DART (自動再送信無効) ビットがセットされていなければ、RCAN-TL1 は自動的に再送信を試みます。DART がセットされていると送信はクリアされ、メールボックスエンプティ割り込みフラグ (IRR8) とアボートアクリッジレジスタ (ABACK) の対応するビットによって CPU に通知されます。

TXPR の状態が変化したときは、バスアービトレーションに負けたり CAN バス上にエラーが発生した場合でも、ID が示す優先順位 (MCR2=0) が最も高いメッセージが常に送信されるように処理します。詳細については「14.6 動作説明」を参照してください。

RCAN-TL1 が TXPR のビットの状態を 0 に変更したとき、メールボックスエンプティ割り込みフラグ (IRR8) が生成されることがあります。これはメールボックスの送信が正常終了したかアボートされたことを示します。メッセージの送信が正常終了した場合は、TXACK に示され、メッセージ送信がアボートされた場合は ABACK に示されます。これらのレジスタをチェックすることによって対応するメールボックスのメッセージデータフィールドの内容を次の送信用に書き替えることができます。

(1) TXPR1

ビット	15	14	13	12	11	10	9	8
ビット名	TXPR1_15	TXPR1_14	TXPR1_13	TXPR1_12	TXPR1_11	TXPR1_10	TXPR1_9	TXPR1_8
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*

ビット	7	6	5	4	3	2	1	0
ビット名	TXPR1_7	TXPR1_6	TXPR1_5	TXPR1_4	TXPR1_3	TXPR1_2	TXPR1_1	TXPR1_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*

【注】 * 送信用に設定されたメールボックスに対応するビットには、1 のみ書き込むことができます。

ビット	ビット名	初期値	R/W	説明
15~0	TXPR1[15:0]	H'0000	R/W	<p>対応するメールボックスに対して CAN フレーム送信をリクエストします。ビット 15~0 はメールボックス 31~16 に対応しています。複数のビットがセットされた場合の送信順序は、MCR2 ビットの設定によりメッセージ ID 優先順またはメールボックス番号順に送信されます。</p> <p>0: 対応するメールボックスが送信メッセージアイドル状態</p> <p>[クリア条件] メッセージ送信終了(イベントトリガメッセージの場合)またはメッセージ送信アボート (自動的にクリアされます)</p> <p>1: 対応するメールボックスに送信リクエストが発生</p>

(2) TXPR0

ビット	15	14	13	12	11	10	9	8
ビット名	TXPR0_15	TXPR0_14	TXPR0_13	TXPR0_12	TXPR0_11	TXPR0_10	TXPR0_9	TXPR0_8
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*
ビット	7	6	5	4	3	2	1	0
ビット名	TXPR0_7	TXPR0_6	TXPR0_5	TXPR0_4	TXPR0_3	TXPR0_2	TXPR0_1	—
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

【注】 * 送信用に設定されたメールボックスに対応するビットには、1のみ書き込むことができます。TXPR1/TXPR0 のリード/ライトは、必ずロングワードアクセスで行ってください。

ビット	ビット名	初期値	R/W	説明
15~1	TXPR0[15:1]	H'0000	R/W	<p>対応するメールボックスに CAN フレーム送信リクエストが発生していることを示します。ビット 15~1 はメールボックス 15~1 に対応しています。複数のビットがセットされた場合の送信順序は、MCR2 ビットの設定によりメッセージ ID 優先順またはメールボックス番号順に送信されます。</p> <p>0: 対応するメールボックスが送信メッセージアイドル状態</p> <p>[クリア条件] メッセージ送信終了(イベントトリガメッセージの場合)またはメッセージ送信アボート (自動的にクリアされます)</p> <p>1: 対応するメールボックスに送信リクエストが発生</p>
0	—	0	R	<p>リザーブビット</p> <p>受信専用メールボックスのため、本ビットは常に 0 となります。1 を書き込むと無視されます。読み出し値は 0 です。</p>

14.5.2 送信キャンセルレジスタ 1、0 (TXCR1、TXCR0)

TXCR1 と TXCR0 は、16 ビットの読み出し/条件付き書き込み可能なレジスタで、TXCR1 はメールボックス 31~16 を制御し、TXCR0 はメールボックス 15~1 を制御します。CPU は TXCR を使用して、TXPR の送信リクエストをキャンセルします。TXPR のビットをクリアするには、CPU から対応する TXCR のビットに 1 を書き込んでください。0 を書き込むと無視されます。

アボートが正常に行われると、CAN コントローラは対応する TXPR と TXCR のビットをクリアし、ABACK のビットをセットします。しかし、メールボックスが送信を開始した後は、本レジスタのビットでキャンセルすることはできません。このような場合、送信が正常終了すると CAN コントローラは TXPR と TXCR のビットをクリアし、TXACK のビットをセットします。しかし、アービトラジョンに負けたりバス上にエラーが発生したことによって送信が行われなかった場合は、CAN コントローラは対応する TXPR と TXCR のビットをクリアし、対応する ABACK のビットをセットします。CPU が送信待ち状態でないメールボックスの送信をクリアしようとしても無視されます。この場合、CPU は TXCR のフラグをセットすることはできません。

(1) TXCR1

ビット	15	14	13	12	11	10	9	8
ビット名	TXCR1_15	TXCR1_14	TXCR1_13	TXCR1_12	TXCR1_11	TXCR1_10	TXCR1_9	TXCR1_8
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*
ビット	7	6	5	4	3	2	1	0
ビット名	TXCR1_7	TXCR1_6	TXCR1_5	TXCR1_4	TXCR1_3	TXCR1_2	TXCR1_1	TXCR1_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*

【注】 * 送信リクエストを受けた送信メールボックスに対応するビットには 1 のみ書き込むことができます。

ビット	ビット名	初期値	R/W	説明
15~0	TXCR1[15:0]	H'0000	R/W	<p>送信用キューに入っているメールボックスに対し、送信キャンセルをリクエストします。ビット 15~0 はメールボックス 31~16 (および TXPR1[15:0]) に対応しています。</p> <p>0: 対応するメールボックスが送信メッセージキャンセルアイドル状態 [クリア条件] 送信メッセージキャンセルが終了 (自動的にクリアされます)</p> <p>1: 対応するメールボックスに対して送信キャンセルを要求</p>

(2) TXCR0

ビット	15	14	13	12	11	10	9	8
ビット名	TXCR0_15	TXCR0_14	TXCR0_13	TXCR0_12	TXCR0_11	TXCR0_10	TXCR0_9	TXCR0_8
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*
ビット	7	6	5	4	3	2	1	0
ビット名	TXCR0_7	TXCR0_6	TXCR0_5	TXCR0_4	TXCR0_3	TXCR0_2	TXCR0_1	—
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

【注】 * 送信リクエストを受けた送信メールボックスに対応するビットには1のみ書き込むことができます。

ビット	ビット名	初期値	R/W	説明
15~1	TXCR0[15:1]	H'0000	R/W	送信用キューに入っているメールボックスに対し、送信キャンセルをリクエストします。ビット 15~1 はメールボックス 15~1 (および TXPR0[15:1]) に対応しています。 0 : 対応するメールボックスが送信メッセージキャンセルアイドル状態 [クリア条件] 送信メッセージキャンセルが終了 (自動的にクリアされます) 1 : 対応するメールボックスに対して送信キャンセルを要求
0	—	0	R	リザーブビット 受信専用メールボックスのため、本ビットは常に0となります。1を書き込むと無視され、読み出し値は常に0です。

14.5.3 送信アクノリッジレジスタ 1、0 (TXACK1、TXACK0)

TXACK1 と TXACK0 は、16 ビットの読み出し/条件付き書き込み可能なレジスタで、メールボックスの送信が正常に行われたことを CPU に通知するために使用します。送信が正常に行われると、RCAN-TL1 は TXACK レジスタの対応するビットをセットします。CPU は、1 を書き込むことによって TXACK のビットをクリアすることができます。0 を書き込むと無視されます。

(1) TXACK1

ビット	15	14	13	12	11	10	9	8
ビット名	TXACK1_15	TXACK1_14	TXACK1_13	TXACK1_12	TXACK1_11	TXACK1_10	TXACK1_9	TXACK1_8
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*
ビット	7	6	5	4	3	2	1	0
ビット名	TXACK1_7	TXACK1_6	TXACK1_5	TXACK1_4	TXACK1_3	TXACK1_2	TXACK1_1	TXACK1_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*

【注】 * ビットをクリアするために 1 のみ書き込むことができます。

ビット	ビット名	初期値	R/W	説明
15~0	TXACK1[15:0]	H'0000	R/W	<p>対応するメールボックスに要求された送信が正常に行われたことを通知します。ビット 15~0 はメールボックス 31~16 に対応しています。</p> <p>0: [クリア条件] 1 を書き込む</p> <p>1: 対応するメールボックスのメッセージ (データフレームまたはリモートフレーム) が正常に送信された</p> <p>[セット条件] 対応するメールボックスのメッセージ送信が完了</p>

(2) TXACK0

ビット	15	14	13	12	11	10	9	8
ビット名	TXACK0_15	TXACK0_14	TXACK0_13	TXACK0_12	TXACK0_11	TXACK0_10	TXACK0_9	TXACK0_8
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*
ビット	7	6	5	4	3	2	1	0
ビット名	TXACK0_7	TXACK0_6	TXACK0_5	TXACK0_4	TXACK0_3	TXACK0_2	TXACK0_1	—
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*	—

【注】 * ビットをクリアするために1のみ書き込むことができます。

ビット	ビット名	初期値	R/W	説明
15~1	TXACK0[15:1]	H'0000	R/W	<p>対応するメールボックスに要求された送信が正常に行われたことを通知します。ビット 15~1 はメールボックス 15~1 に対応しています。</p> <p>0 : [クリア条件] 1 を書き込む</p> <p>1 : 対応するメールボックスのメッセージ (データフレームまたはリモートフレーム) が正常に送信された</p> <p>[セット条件] 対応するメールボックスのメッセージ送信が完了</p>
0	—	0	R	<p>リザーブビット</p> <p>受信専用メールボックスのため、本ビットは常に0となります。1を書き込むと無視され、読み出し値は常に0です。</p>

14.5.4 アポートアクノリッジレジスタ 1、0 (ABACK1、ABACK0)

ABACK1 と ABACK0 は、16 ビットの読み出し/条件付き書き込み可能なレジスタで、要求に応じてメールボックスの送信がアポートされたことを CPU に通知するために使用します。アポートが行われると、RCAN-TL1 は ABACK レジスタの対応するビットをセットします。CPU は、1 を書き込むことによって ABACK のビットをクリアすることができます。0 を書き込むと無視されます。RCAN-TL1 が ABACK のビットをセットすることにより、対応する TXCR ビットによって TXPR ビットがクリアされたことを認識します。

(1) ABACK1

ビット	15	14	13	12	11	10	9	8
ビット名	ABACK1_15	ABACK1_14	ABACK1_13	ABACK1_12	ABACK1_11	ABACK1_10	ABACK1_9	ABACK1_8
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*
ビット	7	6	5	4	3	2	1	0
ビット名	ABACK1_7	ABACK1_6	ABACK1_5	ABACK1_4	ABACK1_3	ABACK1_2	ABACK1_1	ABACK1_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*

【注】 * ビットをクリアするために 1 のみ書き込むことができます。

ビット	ビット名	初期値	R/W	説明
15~0	ABACK1[15:0]	H'0000	R/W	<p>対応するメールボックスに要求された送信キャンセルが正常に行われたことを通知します。ビット 15~0 はメールボックス 31~16 に対応しています。</p> <p>0: [クリア条件] 1 を書き込む</p> <p>1: 対応するメールボックスのメッセージ (データフレームまたはリモートフレーム) の送信がキャンセルされた</p> <p>[セット条件] 対応するメールボックスのメッセージ送信をキャンセル</p>

(2) ABACK0

ビット	15	14	13	12	11	10	9	8
ビット名	ABACK0_15	ABACK0_14	ABACK0_13	ABACK0_12	ABACK0_11	ABACK0_10	ABACK0_9	ABACK0_8
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*
ビット	7	6	5	4	3	2	1	0
ビット名	ABACK0_7	ABACK0_6	ABACK0_5	ABACK0_4	ABACK0_3	ABACK0_2	ABACK0_1	—
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

【注】 * ビットをクリアするために1のみ書き込むことができます。

ビット	ビット名	初期値	R/W	説明
15~1	ABACK0[15:1]	H'0000	R/W	対応するメールボックスに要求された送信キャンセルが正常に行われたことを通知します。ビット 15~1 はメールボックス 15~1 に対応しています。 0 : [クリア条件] 1 を書き込む 1 : 対応するメールボックスのメッセージ (データフレームまたはリモートフレーム) の送信がキャンセルされた [セット条件] 対応するメールボックスのメッセージ送信をキャンセル
0	—	0	R	リザーブビット 受信専用メールボックスのため、本ビットは常に0となります。1を書き込むと無視され、読み出し値は常に0です。

14.5.5 データフレーム受信完了レジスタ 1、0 (RXPR1、RXPR0)

RXPR1 と RXPR0 は、16 ビットの読み出し/条件付き書き込み可能なレジスタで、受信用に設定されたメールボックスがデータフレームを受信したことを示すフラグで構成されています。CAN データフレームが正常に受信メールボックスに格納されると、RXPR の対応するビットがセットされます。1 を書き込むと対応するビットがクリアされます。0 を書き込むと無効とされます。ただし、メールボックスが MBC (メールボックスコンフィギュレーション) によってデータフレームを受信するように設定されている場合のみビットがセットされます。RXPR のビットがセットされると、対応する MBIMR がセットされていなければ IRR1 (データフレーム受信割り込みフラグ) もセットされ、さらに IMR1 がセットされていなければ割り込み信号が生成されません。本レジスタのビットはデータフレームの受信によってのみセットされ、リモートフレーム受信ではセットされません。

(1) RXPR1

ビット	15	14	13	12	11	10	9	8
ビット名	RXPR1_15	RXPR1_14	RXPR1_13	RXPR1_12	RXPR1_11	RXPR1_10	RXPR1_9	RXPR1_8
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*

ビット	7	6	5	4	3	2	1	0
ビット名	RXPR1_7	RXPR1_6	RXPR1_5	RXPR1_4	RXPR1_3	RXPR1_2	RXPR1_1	RXPR1_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*

【注】 * ビットをクリアするために 1 のみ書き込むことができます。

ビット	ビット名	初期値	R/W	説明
15~0	RXPR1[15:0]	H'0000	R/W	ビット 15~0 はメールボックス番号 31~16 の受信用に設定されたメールボックスに対応します。 0 : [クリア条件] 1 を書き込む 1 : 対応するメールボックスが CAN データフレームを受信した [セット条件] 対応するメールボックスのデータフレーム受信終了

(2) RXPR0

ビット	15	14	13	12	11	10	9	8
ビット名	RXPR0_15	RXPR0_14	RXPR0_13	RXPR0_12	RXPR0_11	RXPR0_10	RXPR0_9	RXPR0_8
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*
ビット	7	6	5	4	3	2	1	0
ビット名	RXPR0_7	RXPR0_6	RXPR0_5	RXPR0_4	RXPR0_3	RXPR0_2	RXPR0_1	RXPR0_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*

【注】 * ビットをクリアするために1のみ書き込むことができます。

ビット	ビット名	初期値	R/W	説明
15~0	RXPR0[15:0]	H'0000	R/W	ビット 15~0 はメールボックス番号 15~0 の受信用に設定されたメールボックスに対応します。 0 : [クリア条件] 1 を書き込む 1 : 対応するメールボックスが CAN データフレームを受信した [セット条件] 対応するメールボックスのデータフレーム受信終了

14.5.6 リモートフレーム受信完了レジスタ 1、0 (RFPR1、RFPR0)

RFPR1 と RFPR0 は、16 ビットの読み出し/条件付き書き込み可能なレジスタで、受信用に設定されたメールボックスがリモートフレームを受信したことを示すフラグで構成されています。CAN リモートフレームが正常に受信メールボックスに格納されると、RFPR の対応するビットがセットされます。1 を書き込むと対応するビットがクリアされます。0 を書き込むと無効となります。すべてのメールボックスに対してビットが存在しますが、メールボックスが MBC (メールボックスコンフィギュレーション) によってリモートフレームを受信するように設定されている場合のみビットがセットされます。RFPR のビットがセットされると、対応する MBIMR がセットされていなければ IRR2 (リモートフレーム受信割り込みフラグ) もセットされ、さらに IMR2 がセットされていなければ割り込み信号が生成されます。本レジスタのビットはリモートフレームの受信によってのみセットされ、データフレーム受信ではセットされません。

(1) RFPR1

ビット	15	14	13	12	11	10	9	8
ビット名	RFPR1_15	RFPR1_14	RFPR1_13	RFPR1_12	RFPR1_11	RFPR1_10	RFPR1_9	RFPR1_8
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*
ビット	7	6	5	4	3	2	1	0
ビット名	RFPR1_7	RFPR1_6	RFPR1_5	RFPR1_4	RFPR1_3	RFPR1_2	RFPR1_1	RFPR1_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*

【注】 * ビットをクリアするために1のみ書き込むことができます。

ビット	ビット名	初期値	R/W	説明
15~0	RFPR1[15:0]	H'0000	R/W	メールボックス 31~16 のリモートリクエスト受信フラグです。 0: [クリア条件] 1 を書き込む 1: 対応するメールボックスが CAN リモートフレームを受信した [セット条件] 対応するメールボックスのリモートフレーム受信終了

(2) RFPR0

ビット	15	14	13	12	11	10	9	8
ビット名	RFPR0_15	RFPR0_14	RFPR0_13	RFPR0_12	RFPR0_11	RFPR0_10	RFPR0_9	RFPR0_8
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*
ビット	7	6	5	4	3	2	1	0
ビット名	RFPR0_7	RFPR0_6	RFPR0_5	RFPR0_4	RFPR0_3	RFPR0_2	RFPR0_1	RFPR0_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*

【注】 * ビットをクリアするために1のみ書き込むことができます。

ビット	ビット名	初期値	R/W	説明
15~0	RFPR0[15:0]	H'0000	R/W	メールボックス 15~0 のリモートリクエスト受信フラグです。 0: [クリア条件] 1 を書き込む 1: 対応するメールボックスが CAN リモートフレームを受信した [セット条件] 対応するメールボックスのリモートフレーム受信終了

14.5.7 メールボックスインタラプトマスクレジスタ 1、0 (MBIMR1、MBIMR0)

MBIMR1 と MBIMR0 は、16 ビットの読み出し/書き込み可能なレジスタです。MBIMR は、メールボックスの動作に関連する IRR (IRR1: データフレーム受信割り込み、IRR2: リモートフレーム受信割り込み、IRR8: メールボックスエンプティ割り込み、IRR9: メッセージオーバーラン/オーバーライト割り込み) をマスクするレジスタです。メールボックスが受信に設定されている場合、受信割り込みフラグ (IRR1、IRR2、IRR9) による割り込みをマスクします。RXPR、RFPR、UMSR ビットのセットには影響しません。メールボックスが送信に設定されている場合は、送信や送信アポート (IRR8) による割り込みやメールボックスエンプティ割り込みをマスクします。送信による TXPR/TXCR ビットのクリアと TXACK ビットのセット、送信アポートによる TXPR/TXCR ビットのクリアと ABACK ビットのセットには影響しません。

マスクするメールボックスに対応するビットに 1 を書き込むことでマスクが設定されます。リセット時はすべてのメールボックス割り込みがマスクされます

(1) MBIMR1

ビット	15	14	13	12	11	10	9	8
ビット名	MBIMR1_15	MBIMR1_14	MBIMR1_13	MBIMR1_12	MBIMR1_11	MBIMR1_10	MBIMR1_9	MBIMR1_8
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
ビット	7	6	5	4	3	2	1	0
ビット名	MBIMR1_7	MBIMR1_6	MBIMR1_5	MBIMR1_4	MBIMR1_3	MBIMR1_2	MBIMR1_1	MBIMR1_0
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

ビット	ビット名	初期値	R/W	説明
15~0	MBIMR1[15:0]	H'FFFF	R/W	メールボックス 31~16 からの割り込み要求を許可/禁止します。 0: IRR1、IRR2、IRR8、IRR9 の割り込み要求を許可 1: IRR1、IRR2、IRR8、IRR9 の割り込み要求を禁止

(2) MBIMR0

ビット	15	14	13	12	11	10	9	8
ビット名	MBIMR0_15	MBIMR0_14	MBIMR0_13	MBIMR0_12	MBIMR0_11	MBIMR0_10	MBIMR0_9	MBIMR0_8
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
ビット	7	6	5	4	3	2	1	0
ビット名	MBIMR0_7	MBIMR0_6	MBIMR0_5	MBIMR0_4	MBIMR0_3	MBIMR0_2	MBIMR0_1	MBIMR0_0
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

ビット	ビット名	初期値	R/W	説明
15~0	MBIMR0[15:0]	H'FFFF	R/W	メールボックス 15~0 からの割り込み要求を許可/禁止します。 0 : IRR1、IRR2、IRR8、IRR9 の割り込み要求を許可 1 : IRR1、IRR2、IRR8、IRR9 の割り込み要求を禁止

14.5.8 未読メッセージステータスレジスタ 1、0 (UMSR1、UMSR0)

UMSR1 と UMSR0 は、16 ビットの読み出し/条件付き書き込み可能なレジスタで、新しいメッセージを受信する際に CPU/DMAC/A-DMAC によって内容がアクセスされていない受信メールボックスを記録します。メールボックスの新しいメッセージを受信するときに、RXPR または RFPR の対応するビットが CPU によってクリアされていないと、UMSR のビットがセットされます。1 を書き込むとクリアされます。0 を書き込むと無効とされます。

メールボックスが送信に設定されている場合は、対応する UMSR のビットはセットされません。

(1) UMSR1

ビット	15	14	13	12	11	10	9	8
ビット名	UMSR1_15	UMSR1_14	UMSR1_13	UMSR1_12	UMSR1_11	UMSR1_10	UMSR1_9	UMSR1_8
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*
ビット	7	6	5	4	3	2	1	0
ビット名	UMSR1_7	UMSR1_6	UMSR1_5	UMSR1_4	UMSR1_3	UMSR1_2	UMSR1_1	UMSR1_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*

【注】 * ビットをクリアするために 1 のみ書き込むことができます。

ビット	ビット名	初期値	R/W	説明
15~0	UMSR1[15:0]	H'0000	R/W	<p>メールボックス 31~16 の未読受信メッセージがオーバーライトされたかオーバーランが発生したことを示します。</p> <p>0: [クリア条件] 1 を書き込む</p> <p>1: 未読受信メッセージが新しいメッセージでオーバーライトされた、またはオーバーランが発生した</p> <p>[セット条件] RXPR または RFPR がクリアされる前に新しいメッセージを受信</p>

(2) UMSR0

ビット	15	14	13	12	11	10	9	8
ビット名	UMSR0_15	UMSR0_14	UMSR0_13	UMSR0_12	UMSR0_11	UMSR0_10	UMSR0_9	UMSR0_8
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*
ビット	7	6	5	4	3	2	1	0
ビット名	UMSR0_7	UMSR0_6	UMSR0_5	UMSR0_4	UMSR0_3	UMSR0_2	UMSR0_1	UMSR0_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*

【注】 * ビットをクリアするために 1 のみ書き込むことができます。

ビット	ビット名	初期値	R/W	説明
15~0	UMSR0[15:0]	H'0000	R/W	<p>メールボックス 15~0 の未読受信メッセージがオーバーライトされたかオーバーランが発生したことを示します。</p> <p>0: [クリア条件] 1 を書き込む</p> <p>1: 未読受信メッセージが新しいメッセージでオーバーライトされた、またはオーバーランが発生した</p> <p>[セット条件] RXPR または RFPR がクリアされる前に新しいメッセージを受信</p>

14.5.9 タイマレジスタ

RCAN-TL1 のタイマは 16 ビットのタイマで、数種のクロックソースをサポートしています。プリスケールカウンタにより、クロックの速度を落とすことができます。また、3 個のコンペアマッチレジスタ (TCMR2、TCMR1、TCMR0) を備えています。アドレスマップを以下に示します。

【重要】タイマレジスタはすべてワード (16 ビット) アクセスのみ可能です。

表 14.8 RCAN-TL1 のタイマレジスタの構成

レジスタ名	略称	アドレス	アクセスサイズ (ビット)
タイムトリガコントロールレジスタ 0	TTCR0	H'080	16
サイクルマキシマム/Tx_Enable_Window レジスタ	CMAX_TEW	H'084	16
リファレンストリガオフセットレジスタ	RFTR0FF	H'086	16
タイマステータスレジスタ	TSR	H'088	16
サイクルカウンタレジスタ	CCR	H'08A	16
タイマカウンタレジスタ	TCNTR	H'08C	16
サイクルタイムレジスタ	CYCTR	H'090	16
リファレンスマークレジスタ	RFMK	H'094	16
タイマコンペアマッチレジスタ 0	TCMR0	H'098	16
タイマコンペアマッチレジスタ 1	TCMR1	H'09C	16
タイマコンペアマッチレジスタ 2	TCMR2	H'0A0	16
送信トリガタイムセレクトレジスタ	TTTSEL	H'0A4	16

14.5.10 タイムトリガコントロールレジスタ 0 (TTCR0)

TTCR0 は、16 ビットの読み出し/書き込み可能なレジスタです。

タイマの動作を制御します。定周期送信や監視用レジスタの設定は、本レジスタを設定し、タイマを動作させてから行ってください。

ビット	15	14	13	12	11	10	9	8
ビット名	TCR15	TCR14	TCR13	TCR12	TCR11	TCR10	—	—
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	R
ビット	7	6	5	4	3	2	1	0
ビット名	—	TCR6	TPSC5	TPSC4	TPSC3	TPSC2	TPSC1	TPSC0
初期値 :	0	0	0	0	0	0	0	0
R/W :	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ビット	ビット名	初期値	R/W	説明
15	TCR15	0	R/W	タイマイネーブル 本ビットをセットすると TCNTR は動作し、クリアすると TCNTR と CCR がクリアされます。 0 : TCNTR と CCR をクリアし、動作停止 1 : タイマ動作
14	TCR14	0	R/W	タイムスタンプ値 メールボックス 15~0 の送信および受信タイムスタンプとして、サイクルタイム (CYCTR) を使用するか CCR[5:0] + CYCTR[15:6] を使用するか指定します。本機能は、タイムトリガモードで Rx_Trigger をモニタするのに便利です。 本レジスタはメールボックス 30 および 31 のタイムスタンプには影響しません。 0 : メールボックス 15~0 のタイムスタンプに CYCTR[15:0] を使用 1 : メールボックス 15~0 のタイムスタンプに CCR[5:0] + CYCTR[15:6] を使用
13	TCR13	0	R/W	TCMR2 によるキャンセル 本ビットおよびビット 12 がセットされた状態で、RCAN-TL1 がホルトモード以外でコンペアマッチが発生したときに、当該 TXPR ビットに対応するすべての TXCR ビットを自動セットします。それにより送信キュー内のメッセージをキャンセルします。 0 : TCMR2 のコンペアマッチによる送信キャンセル禁止 1 : TCMR2 のコンペアマッチによる送信キャンセル許可

ビット	ビット名	初期値	R/W	説明
12	TCR12	0	R/W	コンペアマッチイネーブル 本ビットをセットすると TCMR2 のコンペアマッチで IRR11 がセットされます。 0 : TCMR2 のコンペアマッチで IRR11 がセットされない 1 : TCMR2 のコンペアマッチで IRR11 がセットされる
11	TCR11	0	R/W	コンペアマッチイネーブル 本ビットをセットすると TCMR1 のコンペアマッチで IRR15 がセットされます。 0 : TCMR1 のコンペアマッチで IRR15 がセットされない 1 : TCMR1 のコンペアマッチで IRR15 がセットされる
10	TCR10	0	R/W	コンペアマッチイネーブル 本ビットをセットすると TCMR0 のコンペアマッチで IRR14 がセットされます。 0 : TCMR0 のコンペアマッチで IRR14 がセットされない 1 : TCMR0 のコンペアマッチで IRR14 がセットされる
9~7	—	すべて 0	R	リザーブビット 読み出すと常に 0 が読み出されます。書き込む値は常に 0 にしてください。
6	TCR6	0	R/W	TCMR0 によるタイマクリア制御 TCNTR が TCMR0 と一致したときにタイマを H'0000 にクリアするか指定します。また、TCMR0 は IRR14 により CPU に対して割り込み信号を発生することもできます。 【注】RCAN-TL1 が TTCAN モードで動作中 (CMAX≠B'111) は、本ビットを 0 に設定しローカルタイムがクリアされないようにしてください。 0 : TCMR0 でタイマクリアしない 1 : TCMR0 でタイマクリア
5~0	TPSC5~ TPSC0	すべて 0	R/W	タイマプリスケアラ 本ビットの設定により、タイマのソースクロック (4×[RCAN-TL1 のシステムクロック]) を分周して使用することができます。プリスケアラはイベントトリガモードでのみ有効です。TTCAN モードで (CMAX≠B'111) は、自動的に 1 ビットタイミング (=CAN バス上の 1 ビット長) が TCNTR のソースクロックとして選択されます。 ソースクロック周期とタイマ周期の関係を以下に示します。 000000 : 1X ソースクロック 000001 : 2X ソースクロック 000010 : 3X ソースクロック 000011 : 4X ソースクロック 000100 : 5X ソースクロック : 111111 : 64X ソースクロック

14.5.11 サイクルマキシマム/Tx_Enable_Window レジスタ (CMAX_TEW)

CMAX_TEW は、16 ビットの読み出し/書き込み可能なレジスタです。

CMAX はタイムトリガ送信用のサイクルカウンタ (CCR) の最大値を指定し、これによりシステムマトリックスのサイクルカウント数 (Basic Cycle) が設定されます。サイクルカウンタが最大値になると (CCR=CMAX)、1 サイクルカウント後にサイクルカウンタは 0 にクリアされ、IRR10 割り込みが発生します。

TEW は Tx_Enable_Window 幅を指定します。

ビット	15	14	13	12	11	10	9	8
ビット名	—	—	—	—	—	CMAX2	CMAX1	CMAX0
初期値 :	0	0	0	0	0	1	1	1
R/W :	R	R	R	R	R	R/W	R/W	R/W
ビット	7	6	5	4	3	2	1	0
ビット名	—	—	—	—	TEW3	TEW2	TEW1	TEW0
初期値 :	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R/W	R/W	R/W	R/W

ビット	ビット名	初期値	R/W	説明
15~11	—	すべて 0	R	リザーブビット 読み出すと常に 0 が読み出されます。書き込む値は常に 0 にしてください。
10~8	CMAX[2:0]	111	R/W	<p>サイクルカウンタ最大値 CCR の最大値を示します。 サイクルカウンタ最大値 = $2^{\text{CMAX}} - 1$ (CMAX = レジスタ値) タイムトリガ送信用のマトリックスサイクル内で可能なサイクルカウント数は、(サイクルカウンタ最大値 + 1) となります。 CMAX ≠ B'111 設定時、RCAN-TL1 はタイムトリガモードであり、タイムトリガ機能を使用できます。 CMAX = B'111 設定時、RCAN-TL1 はイベントトリガモードです。</p> <p>000 : サイクルカウンタ最大値 = 0 001 : サイクルカウンタ最大値 = 1 010 : サイクルカウンタ最大値 = 3 011 : サイクルカウンタ最大値 = 7 100 : サイクルカウンタ最大値 = 15 101 : サイクルカウンタ最大値 = 31 110 : サイクルカウンタ最大値 = 63 111 : RCAN-TL1 はイベントトリガモード (CCR はクリア)</p> <p>【注】 イベントトリガモードを使用する場合は、CMAX = B'111 に設定してください。</p>

ビット	ビット名	初期値	R/W	説明
7~4	—	すべて 0	R	リザーブビット 読み出すと常に 0 が読み出されます。書き込む値は常に 0 にしてください。
3~0	TEW[3:0]	0000	R/W	<p>Tx_Enable_Window</p> <p>Tx_Enable_Window 幅を設定します。</p> <p>TEW=B'0000 のとき、ウィンドウ幅は 1 ビットタイミングとなります。</p> <p>B'0000~B'1111 のすべての値を設定することができます。</p> <p>0000 : Tx_Enable_Window 幅=1</p> <p>0001 : Tx_Enable_Window 幅=2</p> <p>0010 : Tx_Enable_Window 幅=3</p> <p>0011 : Tx_Enable_Window 幅=4</p> <p>：</p> <p>1111 : Tx_Enable_Window 幅=16</p> <p>【注】 RCAN-TL1 の CAN データリンクコントローラは、送信リクエストから送信を開始するのに常に CAN バス 1~2 ビットタイミングを必要とします。したがって、上記の値はこの精度を考慮していません。</p>

14.5.12 リファレンストリガオフセットレジスタ (RFTROFF)

RFTROFF は、8 ビットの読み出し／書き込み可能なレジスタです。

メールボックス 30 の送信トリガタイム (TTT) に -127 ~ +127 までオフセットを設定することができます。メールボックス 30 の TTT は、RFTROFF レジスタ値の符号拡張値を付加した後、CYCTR と比較されます。ただし、TTT の値は変更されません。本オフセット値はメールボックス 30 にのみ影響します。

ビット	15	14	13	12	11	10	9	8
ビット名	RFTROFF7	RFTROFF6	RFTROFF5	RFTROFF4	RFTROFF3	RFTROFF2	RFTROFF1	RFTROFF0
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
ビット	7	6	5	4	3	2	1	0
ビット名	—	—	—	—	—	—	—	—
初期値 :	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R

ビット	ビット名	初期値	R/W	説明
15~8	RFTROFF [7:0]	すべて 0	R/W	リファレンストリガオフセット値を示します。 00000000 : ref_trigger_offset =+0 00000001 : ref_trigger_offset =+1 00000010 : ref_trigger_offset =+2 : 01111111 : ref_trigger_offset =+127 : 11111111 : ref_trigger_offset =-1 11111110 : ref_trigger_offset =-2 : 10000001 : ref_trigger_offset =-127
7~0	—	すべて 0	R	リザーブビット 読み出すと常に 0 が読み出されます。書き込む値は常に 0 にしてください。

14.5.13 タイマステータスレジスタ (TSR)

TSR は、16 ビットの読み出し専用レジスタで、これにより CPU はタイマコンペアマッチ状態およびタイマオーバーラン状態を監視することができます。

ビット	15	14	13	12	11	10	9	8
ビット名	—	—	—	—	—	—	—	—
初期値 :	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R
ビット	7	6	5	4	3	2	1	0
ビット名	—	—	—	TSR4	TSR3	TSR2	TSR1	TSR0
初期値 :	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R

ビット	ビット名	初期値	R/W	説明
15~5	—	すべて 0	R	リザーブビット 読み出すと常に 0 が読み出されます。書き込む値は常に 0 にしてください。
4	TSR4	0	R*	スタートシステムマトリックス 次のシステムマトリックスが開始したことを示します。 CCR=0 のとき、本ビットはタイムリファレンスメッセージの送信/受信完了時にセットされます。 0 : 次のシステムマトリックスの先頭でない [クリア条件] IRR10 (サイクルカウンタオーバーフロー割り込みフラグ) に 1 を書き込む 1 : サイクルカウンタが 0 になった [セット条件] サイクルカウンタの値が最大値 (CMAX) から H'0 に変化 CMAX!=B'111 かつ CCR=0 のとき、タイムリファレンスメッセージの送信/受信が完了

ビット	ビット名	初期値	R/W	説明
3	TSR3	0	R*	<p>タイマコンペアマッチフラグ 2</p> <p>タイマコンペアマッチレジスタ 2 (TCMR2) のコンペアマッチが発生したことを示します。TCMR2 に設定した値がサイクルタイムレジスタと一致 (TCMR2=CYCTR) したとき、TTCR0 のビット 12 が 1 に設定されていれば本ビットがセットされます。本ビットは読み出し専用で、IRR11 (タイマコンペアマッチ割り込み 2 フラグ) がクリアされるとクリアされます。</p> <p>0: TCMR2 のタイマコンペアマッチが発生していない</p> <p>[クリア条件]</p> <p>IRR11 (タイマコンペアマッチ割り込み 2 フラグ) に 1 を書き込む</p> <p>1: TCMR2 のタイマコンペアマッチが発生</p> <p>[セット条件]</p> <p>TTCR0 のビット 12 が 1 のとき TCMR2 がサイクルタイムと一致 (TCMR2=CYCTR)</p>
2	TSR2	0	R*	<p>タイマコンペアマッチフラグ 1</p> <p>タイマコンペアマッチレジスタ 1 (TCMR1) のコンペアマッチが発生したことを示します。TCMR1 に設定した値がサイクルタイムレジスタと一致 (TCMR1=CYCTR) したとき、TTCR0 のビット 11 が 1 に設定されていれば本ビットがセットされます。本ビットは読み出し専用で、IRR15 (タイマコンペアマッチ割り込み 1 フラグ) がクリアされるとクリアされます。</p> <p>0: TCMR1 のタイマコンペアマッチが発生していない</p> <p>[クリア条件]</p> <p>IRR15 (タイマコンペアマッチ割り込み 1 フラグ) に 1 を書き込む</p> <p>1: TCMR1 のタイマコンペアマッチが発生</p> <p>[セット条件]</p> <p>TTCR0 のビット 11 が 1 のとき TCMR1 がサイクルタイムと一致 (TCMR1=CYCTR)</p>
1	TSR1	0	R*	<p>タイマコンペアマッチフラグ 0</p> <p>タイマコンペアマッチレジスタ 0 (TCMR0) のコンペアマッチが発生したことを示します。TCMR0 に設定した値がタイマ値と一致 (TCMR0=TCNTR) したとき、TTCR0 のビット 10 が 1 に設定されていれば本ビットがセットされます。本ビットは読み出し専用で、IRR14 (タイマコンペアマッチ割り込み 0 フラグ) がクリアされるとクリアされます。</p> <p>0: TCMR0 のタイマコンペアマッチが発生していない</p> <p>[クリア条件]</p> <p>IRR14 (タイマコンペアマッチ割り込み 0 フラグ) に 1 を書き込む</p> <p>1: TCMR0 のタイマコンペアマッチが発生</p> <p>[セット条件]</p> <p>TTCR0 のビット 10 が 1 のとき TCMR0 がタイマ値と一致 (TCMR0=TCNTR)</p>

ビット	ビット名	初期値	R/W	説明
0	TSR0	0	R*	<p>タイマオーバーラン/Next_is_Gap 受信/メッセージエラー</p> <p>本フラグは3つの異なる機能が割り付けられています。イベントトリガモードで動作中にタイマオーバーランが発生、タイムトリガモードで動作中にNext_is_Gap がセットされたタイムリファレンスメッセージを受信、およびテストモード中にCANバス上で検出されたエラーが発生したことを示します。テストモードは他の設定よりも高い優先度を持ちます。</p> <p>0: イベントトリガモードでタイマ(TCNTR)オーバーランが発生していない タイムトリガモードでNext_is_Gap 付きタイムリファレンスメッセージを受信していない テストモードでメッセージエラーが発生していない</p> <p>[クリア条件] IRR13に1を書き込む</p> <p>1: [セット条件] イベントトリガモードでタイマ(TCNTR)オーバーランが発生し、H'FFFFからH'0000に変化 タイムトリガモードでNext_is_Gap 付きタイムリファレンスメッセージを受信 テストモードでメッセージエラーが発生</p>

【注】 * 本ビットは読み出し専用で、CPUはサイクルカウンタ、タイマ、およびコンペアマッチレジスタの状態を監視することができます。本ビットに対する書き込みは無効です。

14.5.14 サイクルカウンタレジスタ (CCR)

CCR は、6 ビットの読み出し/書き込み可能なレジスタです。

タイムトリガ送信のサイクルカウント値 (Basic Cycle) を表示するためのレジスタです。

CCR の値は、RCAN-TL1 がポテンシャルタイムマスタとして動作するか、タイムスレーブとして動作するかにより、更新方法が変わります。CCR は以下のように更新されます。

RCAN-TL1 がタイムマスタ (ポテンシャル) として動作する場合：

- サイクルタイム (CYCTR) がメールボックス30の送信トリガタイムと一致するたびに+1増加
または
- 有効なリファレンスメッセージを受信するたびにメールボックス31のMSG_DATA_0[5:0]の値を上書き

RCAN-TL1 がタイムスレーブとして動作する場合：

- 有効なリファレンスメッセージを受信するたびにメールボックス31のMSG_DATA_0[5:0]の値で上書き

CMAX=B'111 かつ TTCR0[15]=0 の場合：

- CCRの値は常にB'000000

ビット	15	14	13	12	11	10	9	8
ビット名	—	—	—	—	—	—	—	—
初期値：	0	0	0	0	0	0	0	0
R/W：	R	R	R	R	R	R	R	R
ビット	7	6	5	4	3	2	1	0
ビット名	—	—	CCR5	CCR4	CCR3	CCR2	CCR1	CCR0
初期値：	0	0	0	0	0	0	0	0
R/W：	R	R	R/W	R/W	R/W	R/W	R/W	R/W

ビット	ビット名	初期値	R/W	説明
15~6	—	すべて0	R	リザーブビット 読み出すと常に0が読み出されます。書き込む値は常に0にしてください。
5~0	CCR[5:0]	すべて0	R/W	サイクルカウンタレジスタ タイムトリガ送信用マトリックスサイクルの現在のサイクルカウント数を示します。

14.5.15 タイマカウンタレジスタ (TCNTR)

TCNTR は、16 ビットの読み出し/書き込み可能なレジスタです。

TCNTR はフリーランニングタイマカウンタです。TTCR0[15]=1 に設定することでタイマがカウントします。タイマカウンタのプリスケアラは、RCAN-TL1 のトリガモードにより変わります。

- イベントトリガモード時 (CMAX=B'111) : TTCR0[5:0]のTPSC値
- タイムトリガモード時 (CMAX≠B'111) : CANバス1ビットタイミング

TTCR0[6]=1 を設定時、TCNTR の値がタイマコンペアマッチレジスタ 0 (TCMR0) と一致すると、TCNTR は H'0000 にクリアされカウントアップします。

タイムトリガモードでは、TCNTR はローカルタイム (Local time) として使用します。したがって、TTCR0[6]=0 に設定してフリーランニングタイマとして動作させます。

- 【注】 1. TTCR0[15]=0 の場合、TCNTR は常に H'0000 です。
2. タイマをイネーブル設定(TTCR0[15]=1)してから TCNTR がカウントを開始するまで、数クロックサイクルの遅延があります。これはプリスケアラの内部論理に起因するものです。

ビット	15	14	13	12	11	10	9	8
ビット名	TCNTR15	TCNTR14	TCNTR13	TCNTR12	TCNTR11	TCNTR10	TCNTR9	TCNTR8
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*

ビット	7	6	5	4	3	2	1	0
ビット名	TCNTR7	TCNTR6	TCNTR5	TCNTR4	TCNTR3	TCNTR2	TCNTR1	TCNTR0
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】 * 本レジスタへの書き込みは、タイマイネーブル設定 (TTCR0[15]=1) されたときのみ可能です。タイムトリガモード (CMAX が B'111 以外) では、書き込みは禁止です。

ビット	ビット名	初期値	R/W	説明
15~0	TCNTR[15:0]	H'0000	R/W	フリーランニングタイマの値を示します。

14.5.16 サイクルタイムレジスタ (CYCTR)

CYCTR は、サイクルタイムを示す 16 ビットの読み出し専用レジスタです。

$Cycle_Time (CYCTR) = Local_Time (TCNTR) - Reference_Mark (RFMK)$

イベントトリガモードでは、RFMK は 0 に固定されるので本レジスタは TCNTR のコピーとなります。

ビット	15	14	13	12	11	10	9	8
ビット名	CYCTR15	CYCTR14	CYCTR13	CYCTR12	CYCTR11	CYCTR10	CYCTR9	CYCTR8
初期値:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R
ビット	7	6	5	4	3	2	1	0
ビット名	CYCTR7	CYCTR6	CYCTR5	CYCTR4	CYCTR3	CYCTR2	CYCTR1	CYCTR0
初期値:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

ビット	ビット名	初期値	R/W	説明
15~0	CYCTR[15:0]	H'0000	R	サイクルタイムを示します。

14.5.17 リファレンスマークレジスタ (RFMK)

RFMK は、16 ビットの読み出し専用レジスタです。リファレンスメッセージが正常に送信または受信されたときに、リファレンスメッセージの SOF におけるローカルタイム (TCNTR) をキャプチャするためのレジスタです。イベントトリガモードでは本レジスタは使用せず、常に 0 にクリアされます。

ビット	15	14	13	12	11	10	9	8
ビット名	RFMK15	RFMK14	RFMK13	RFMK12	RFMK11	RFMK10	RFMK9	RFMK8
初期値:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R
ビット	7	6	5	4	3	2	1	0
ビット名	RFMK7	RFMK6	RFMK5	RFMK4	RFMK3	RFMK2	RFMK1	RFMK0
初期値:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

ビット	ビット名	初期値	R/W	説明
15~0	RFMK[15:0]	H'0000	R	リファレンスマークレジスタ タイムリファレンスメッセージの SOF における TCNTR の値を示します。

14.5.18 タイマコンペアマッチレジスタ 0~2 (TCMR0~2)

TCMR0~2 は、16 ビットの読み出し/書き込み可能なレジスタです。

割り込み信号の発生、タイマ値のクリア (TCMR0 のみサポート)、送信リクエストのクリア (TCMR2 のみサポート) を行うことが可能です。

TCMR0 は TCNTR と比較し、TCMR1 と TCMR2 は CYCTR と比較します。各コンペアマッチレジスタの比較値はそれぞれ独立に設定することができます。コンペアマッチのフラグをセットするには、TTCR0 のビット 12~10 をセットする必要があります。タイムトリガモードでは、TCMR0 は Init_Watch_Trigger、TCMR2 は Watch_Trigger に使用します。

(1) 割り込み機能

各レジスタの割り込みフラグは、TTCR0 のビット 12、ビット 11、ビット 10 を設定することで許可します。コンペアマッチが発生すると、IRR の対応する割り込みフラグ (ビット 11、ビット 15、ビット 14) をセットします。

割り込み信号の発生は、IMR のビット 11、ビット 15、ビット 14 で抑止することができます。

コンペアマッチ発生時に IRR11 (または IRR15 か IRR14) がセットされると、タイムステータスレジスタ (TSR) のビット 3、ビット 2、またはビット 1 もセットされます。また、IRR のビットをクリアすると、対応する TSR のビットもクリアされます。

(2) タイマクリア機能

TCMR0 のみコンペアマッチによるタイマ値 (TCNTR) のクリア機能があります。TTCR0[6]=1 設定時に有効です。TCMR1 と TCMR2 には本機能はありません。

(3) 送信リクエストされたメッセージのキャンセル機能

TCMR2 は、TTCR0[13:12]=B'11 設定時に、ホルトモード以外で TCMR2 のコンペアマッチ発生時に TXPR 設定に対応する TXCR 設定を行います。したがって、送信中のメッセージを除き、送信待ちリクエストをキャンセルします。

TCMR1 と TCMR0 には本機能はありません。

• TCMR0

ビット	15	14	13	12	11	10	9	8
ビット名	TCMR0_15	TCMR0_14	TCMR0_13	TCMR0_12	TCMR0_11	TCMR0_10	TCMR0_9	TCMR0_8
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
ビット	7	6	5	4	3	2	1	0
ビット名	TCMR0_7	TCMR0_6	TCMR0_5	TCMR0_4	TCMR0_3	TCMR0_2	TCMR0_1	TCMR0_0
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

ビット	ビット名	初期値	R/W	説明
15~0	TCMR0[15:0]	H'FFFF	R/W	タイマコンペアマッチレジスタ コンペアマッチ発生時の TCNTR の値を示します。

- TCMR1

ビット	15	14	13	12	11	10	9	8
ビット名	TCMR1_15	TCMR1_14	TCMR1_13	TCMR1_12	TCMR1_11	TCMR1_10	TCMR1_9	TCMR1_8
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
ビット	7	6	5	4	3	2	1	0
ビット名	TCMR1_7	TCMR1_6	TCMR1_5	TCMR1_4	TCMR1_3	TCMR1_2	TCMR1_1	TCMR1_0
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

ビット	ビット名	初期値	R/W	説明
15~0	TCMR1 [15:0]	H'FFFF	R/W	タイマコンペアマッチレジスタ コンペアマッチ発生時の CYCTR の値を示します。

- TCMR2

ビット	15	14	13	12	11	10	9	8
ビット名	TCMR2_15	TCMR2_14	TCMR2_13	TCMR2_12	TCMR2_11	TCMR2_10	TCMR2_9	TCMR2_8
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
ビット	7	6	5	4	3	2	1	0
ビット名	TCMR2_7	TCMR2_6	TCMR2_5	TCMR2_4	TCMR2_3	TCMR2_2	TCMR2_1	TCMR2_0
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

ビット	ビット名	初期値	R/W	説明
15~0	TCMR2[15:0]	H'FFFF	R/W	タイマコンペアマッチレジスタ コンペアマッチ発生時の CYCTR の値を示します。

14.5.19 送信トリガタイムセレクトレジスタ (TTTSEL)

TTTSEL は、16 ビットの読み出し／書き込み可能なレジスタで、サイクルタイムとコンペアマッチさせる送信トリガタイムを指定します。1 ビットのみセットできます。複数のビットをセットしないでください。またはすべてのビットをクリアしないでください。本レジスタは、コンフィギュレーションモードでのみ変更することができます。変更時のアルゴリズムを図 14.14 に示します。

TTTSEL はテストおよび診断専用です。通常動作時は書き込まないでください。また、読み出し値は保証されません。

ビット	15	14	13	12	11	10	9	8
ビット名	—	TTTSEL14	TTTSEL13	TTTSEL12	TTTSEL11	TTTSEL10	TTTSEL9	TTTSEL8
初期値 :	0	1	0	0	0	0	0	0
R/W :	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ビット	7	6	5	4	3	2	1	0
ビット名	—	—	—	—	—	—	—	—
初期値 :	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R

【注】 1 ビットのみセット可能。

ビット	ビット名	初期値	R/W	説明
15	—	0	R	リザーブビット 読み出すと常に 0 が読み出されます。書き込む値は常に 0 にしてください。
14~8	TTTSEL [14:8]	B'1000000	R/W	CYCTR とコンペアマッチさせる送信トリガタイムを指定します。ビット 14~8 は、メールボックス 30~24 に対応しています。
7~0	—	すべて 0	R	リザーブビット 読み出すと常に 0 が読み出されます。書き込む値は常に 0 にしてください。

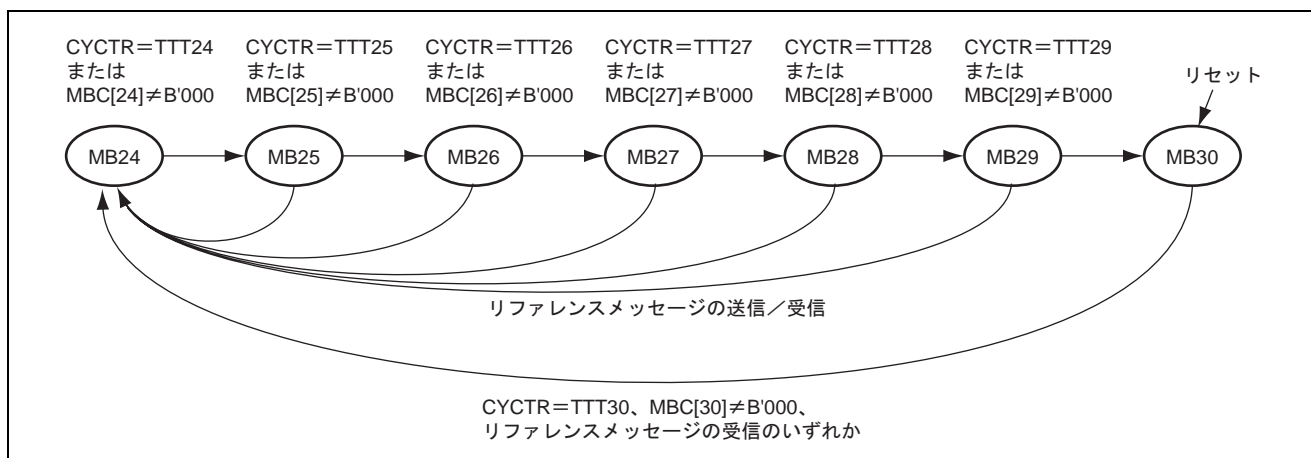


図 14.14 TTTSEL 変更アルゴリズム

14.6 動作説明

14.6.1 RCAN-TL1 の設定

ハードウェアリセット（パワーオンリセット）またはソフトウェアリセット（MCR0）後のコンフィギュレーションモードおよびホルトモード時の RCAN-TL1 の設定について説明します。どちらの場合も RCAN-TL1 は CAN バスアクティビティに参加できません。また、RCAN-TL1 の設定の変更が CAN バス上の通信に影響を与えることはありません。

(1) リセットシーケンス

図 14.15 にソフトウェアリセットまたはハードウェアリセット後の RCAN-TL1 の設定手順を示します。リセット後すべてのレジスタは初期化されます。したがって、CAN バスアクティビティに参加する前に RCAN-TL1 を設定する必要があります。詳細については図中の注を参照してください。

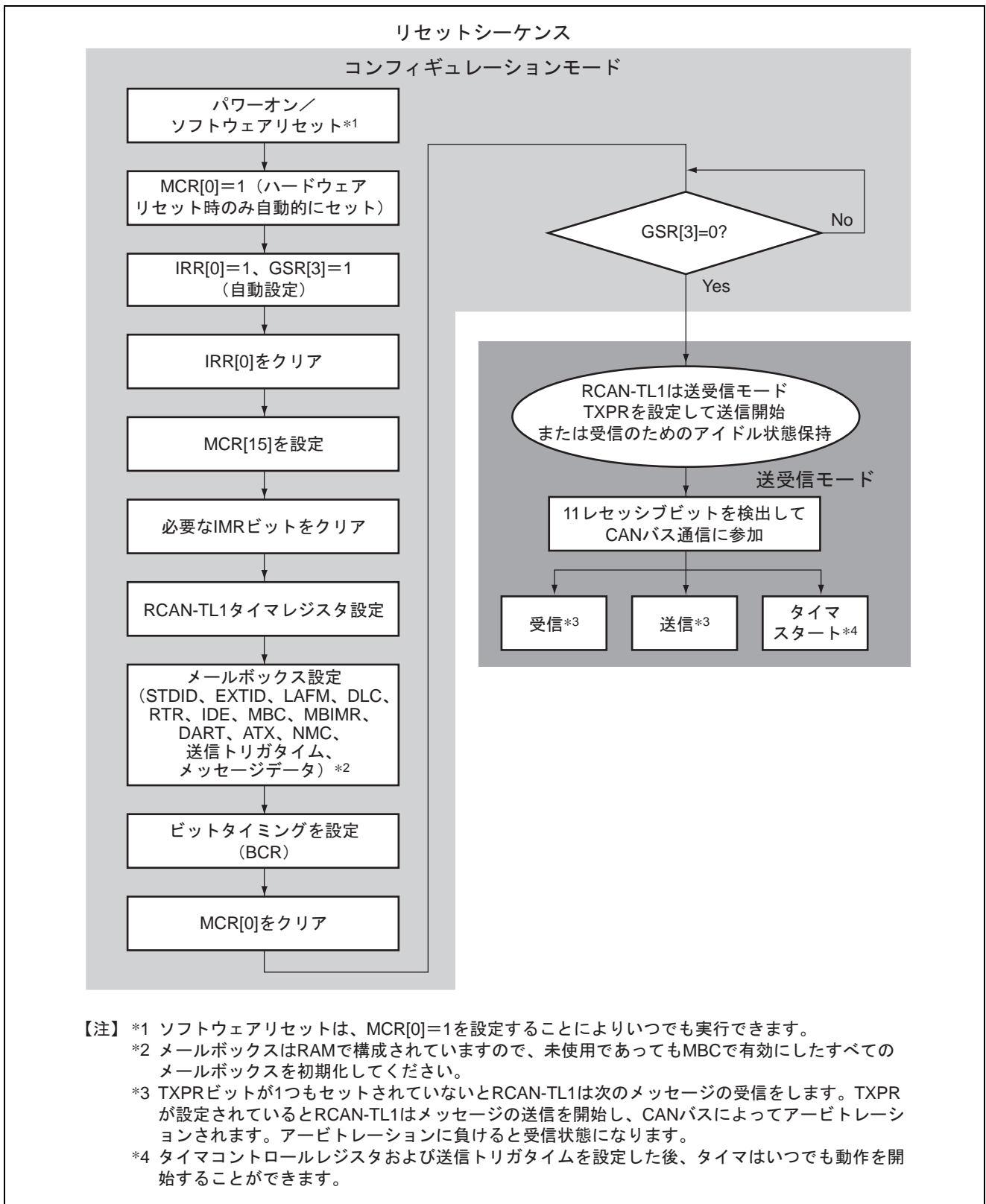


図 14.15 リセットシーケンス

(2) ホルトモード

RCAN-TL1 はホルトモードのとき、CAN バスアクティビティに参加することができません。したがってユーザは、CAN バス上の通信に影響を与えることなく必要なレジスタの設定を変更することができます。ここで重要なのは、レジスタを変更する前に RCAN-TL1 がホルトモードになるまで待つことです。ホルトモードへの遷移はかならずしも即時に行われるとはかぎりません (CAN バスがアイドルまたは休止状態のときに遷移します)。RCAN-TL1 がホルトモードに遷移すると GSR4 ビットがセットされます。

レジスタ設定終了後はホルトリクエストを解除する必要があります。ホルトリクエスト解除後 (MCR1=0 か GSR4=0) に RCAN-TL1 は CAN バス上で 11 レセッシブビットを検出した後 CAN バスアクティビティに参加します。

(3) CAN スリープモード

CAN スリープモード中では、RCAN-TL1 の主要なモジュールのクロックは消費電流を低減するために停止しますが、MCR、GSR、IRR、および IMR レジスタのみアクセスできます。なお、送信と受信に関連する割り込みは、CAN スリープモード中ではクリアできませんので、あらかじめクリアしてください。

図 14.16 に RCAN-TL1 の CAN スリープモードのフローチャートを示します。

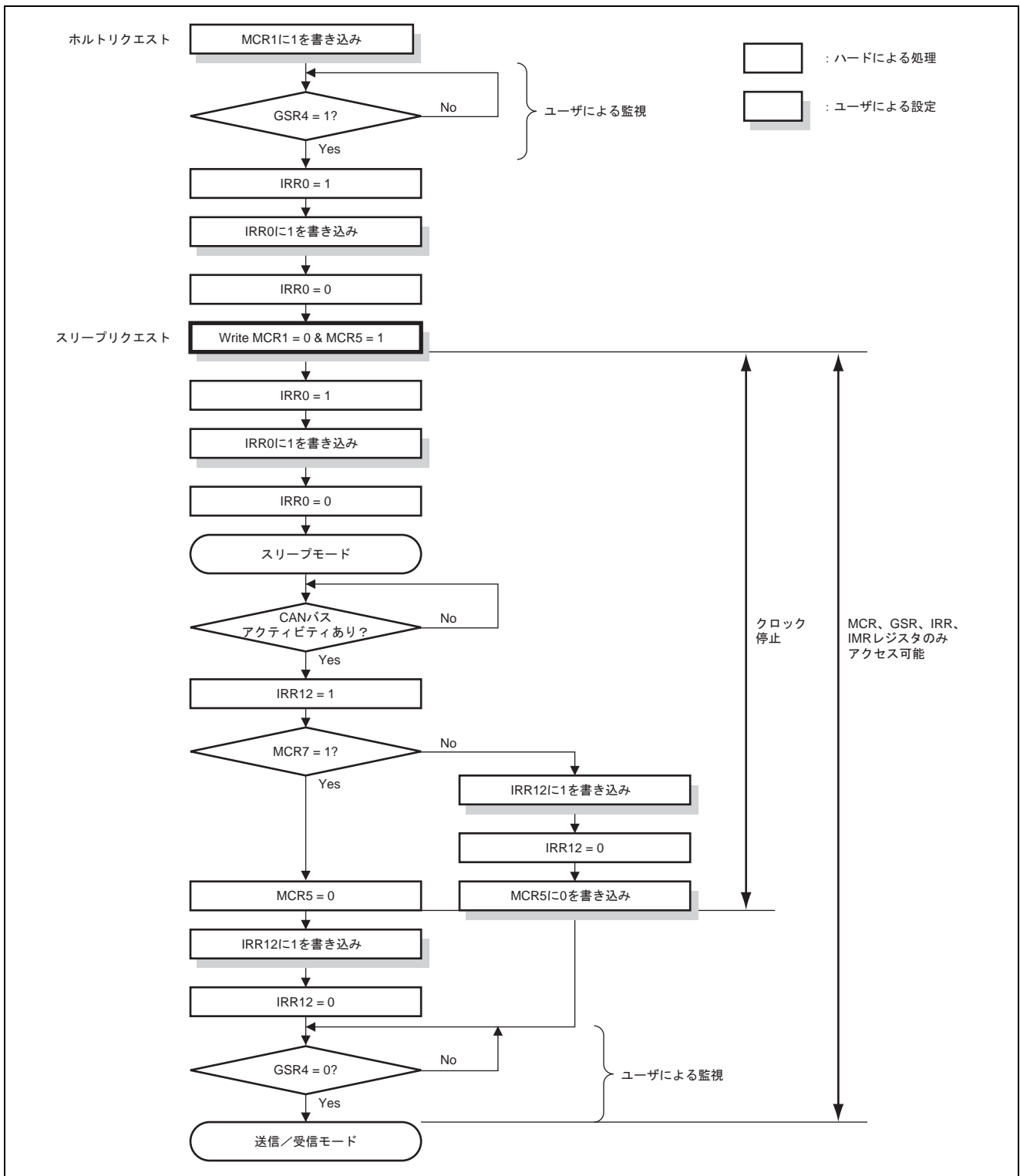


図 14.16 CAN スリープモードのフローチャート

図 14.17 に可能な状態遷移を示します。

なお、ホルトモード以外で MCR5 (CAN スリープモード) ビットをセットしないでください。

また、MCR1 ビットをセットした後は、MCR1 をクリアする前に必ず GSR4 のセットを確認して RCAN-TL1 をホルトモードにしてください。

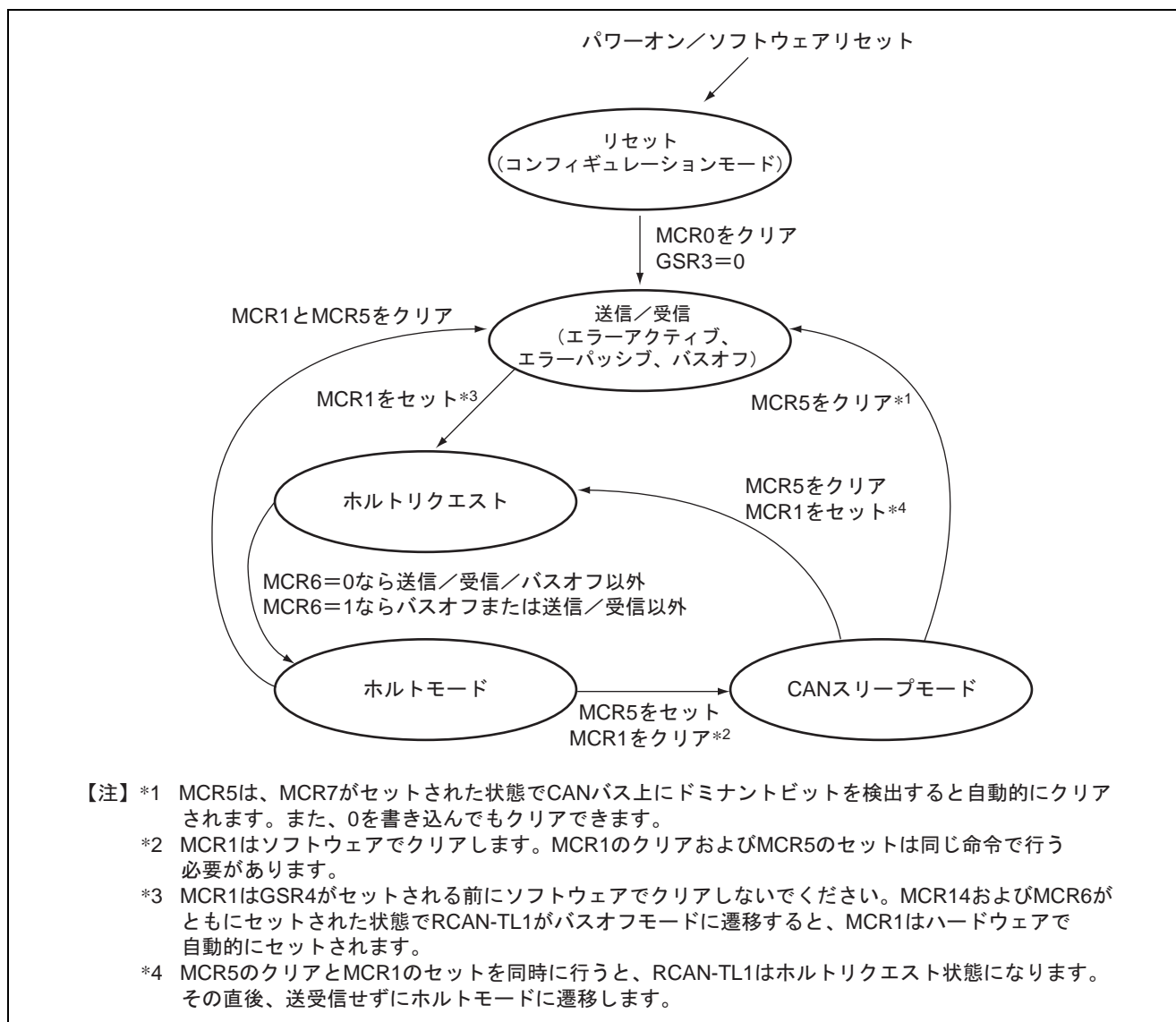


図 14.17 状態遷移図

各モードでのアクセスを許可する条件を表 14.9 に示します。

表 14.9 アクセス可能なレジスタ

ステータス モード	レジスタ									
	MCR、 GSR	IRR、 IMR	BCR	MBIMR タイマ TT レジスタ	フラグ レジスタ	メール ボックス (コント ロール 0、 LAFM)	メール ボックス (データ)	メール ボックス (コント ロール 1)	メール ボックス (トリガ タイム、 TT コントロ ール)	
リセット	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
送信/ 受信	Yes	Yes	No	Yes	Yes	No	Yes*	Yes*	No	Yes*
ホルト リクエスト	Yes	Yes	No	Yes	Yes	No	Yes*	Yes*	No	Yes*
ホルト モード	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
CAN スリープ モード	Yes	Yes	No	No	No	No	No	No	No	No

【記号説明】

Yes : アクセスを許可するレジスタ

No : アクセスを禁止するレジスタ

【注】 * TXPR0 がセットされていない場合

14.6.2 テストモードの設定

RCAN-TL1 には種々のテストモードがあります。テストモードの選択は MCR レジスタの TST[2:0] ビットで行います。RCAN-TL1 は、デフォルト (初期値) ではノーマルモードで動作します。

表 14.10 にテストモード設定を示します。

テストモードの選択はコンフィギュレーションモードでのみ可能です。選択したテストモードを実行するには、テストモードを選択した後コンフィギュレーションモードを解除 (BCR0/BCR1 が設定されていることを確認) してください。

表 14.10 テストモードの設定

TST2	TST1	TST0	モード
0	0	0	ノーマルモード
0	0	1	リスンオンリモード (受信専用モード)
0	1	0	セルフテストモード 1 (外部)
0	1	1	セルフテストモード 2 (内部)
1	0	0	ライトエラーカウンタ
1	0	1	エラーパッシブモード
1	1	0	設定禁止
1	1	1	設定禁止

- ノーマルモード
RCAN-TL1 は通常の動作をします。
- リスンオンリモード
ボーレート検出用に ISO-11898 で要求されているモードです。エラーカウンタはクリアされた後に動作禁止となり、TEC/REC の値が増加しないようになります。また、CTxn (n=A, B, C) 出力を禁止し、RCAN-TL1 によるエラーフレームやアクノリッジビットの生成を抑止します。メッセージエラーが発生すると IRR13 がセットされます。
- セルフテストモード (外部)
RCAN-TL1 自身でアクノリッジビットを生成し、必要ならばメッセージを RCAN-TL1 の受信メールボックスに格納します。CRxn/CTxn (n=0, 1) 端子は必ず CAN バスに接続してください。
- セルフテストモード (内部)
RCAN-TL1 自身でアクノリッジビットを生成し、必要ならばメッセージを RCAN-TL1 の受信メールボックスに格納します。内部 CTxn (n=0, 1) が内部 CRxn (n=0, 1) にループバックされるため、CRxn/CTxn (n=0, 1) 端子を CAN バスその他の外部デバイスに接続する必要はありません。CTxn (n=0, 1) 端子はレセツピビットのみ出力し、CRxn (n=0, 1) 端子は無効となります。
- ライトエラーカウンタ
TEC/REC は本モードで書き込み可能です。エラーカウンタに 127 より大きい値を書き込むことにより、RCAN-TL1 を強制的にエラーパッシブモードにすることができます。TEC に書き込まれた値は REC に書き込まれるので、TEC と REC は常に同じ値に設定されます。同様に、95 より大きい値を書き込むことにより、RCAN-TL1 を強制的にエラーワーニングモードにすることができます。
TEC/REC に書き込む際は RCAN-TL1 がホルトモードでなければなりません (エラーカウンタ書き込み時に MCRI=1)。さらに CAN 仕様により、ホルトモードを解除する前に本テストモードを解除して TEC/REC を再び動作可能にする必要があります。
- エラーパッシブモード
RCAN-TL1 は強制的にエラーパッシブモードにすることができます。
エラーパッシブモードを実行することによって REC の値が変わることはありませんが、一度エラーパッシブモードで動作すると、エラーを受信すれば REC の値は通常どおり増加します。本モードでは、TEC の値が 256 に達すると RCAN-TL1 はバスオフ状態になりますが、本モードを使用すると RCAN-TL1 はエラーアクティブになることができません。したがって RCAN-TL1 はバスオフ復帰シーケンスの最後に、エラーアクティブではなくエラーパッシブに遷移します。

メッセージエラー発生時には、すべてのテストモードで IRR13 ビットがセットされます。

14.6.3 メッセージ送信シーケンス

(1) メッセージ送信リクエスト

バス上に CAN フレームを送信するシーケンスの例を図 14.18 に示します。

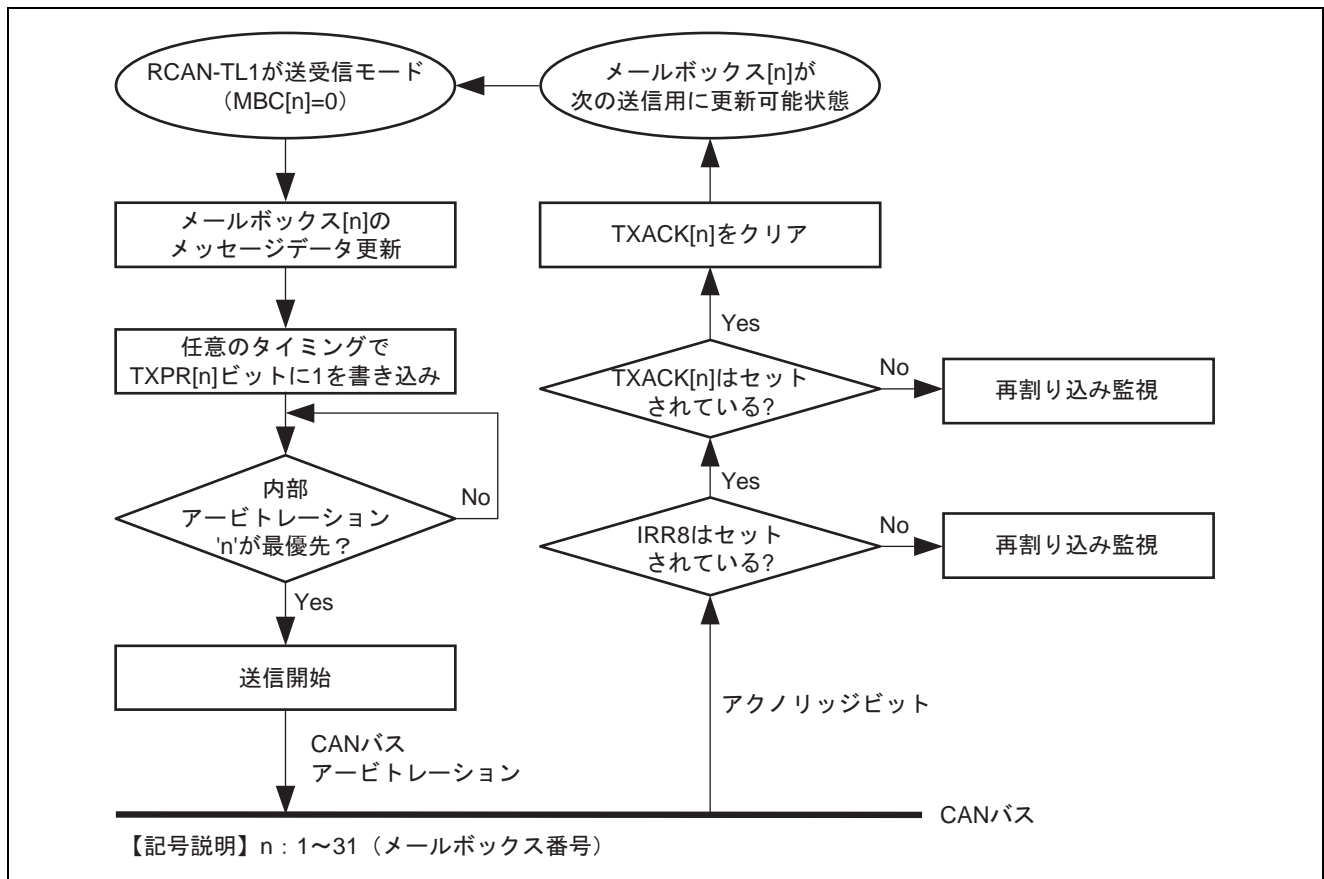


図 14.18 メッセージ送信リクエスト

レジスタの章で説明したとおり、TXACK または ABACK ビットのいずれかがセットされると、IRR8 がセットされることにご注意ください。これはいずれかのメールボックスが送信または送信アボートを終了し、次の送信用に更新可能状態であることを意味しています。一方、GSR2 は現在送信リクエストが発生していない (すべての TXPR フラグがセットされていない) ことを示しています。

(2) 送信用内部アービトレーション

図 14.19 は、RCAN-TL1 がどのようにしてメッセージ ID に従った順序で送信リクエストされたメッセージのスケジューリングを行うかを説明しています。内部アービトレーションでは、送信リクエストされたメッセージのうち最も優先度の高いものを取り出します。

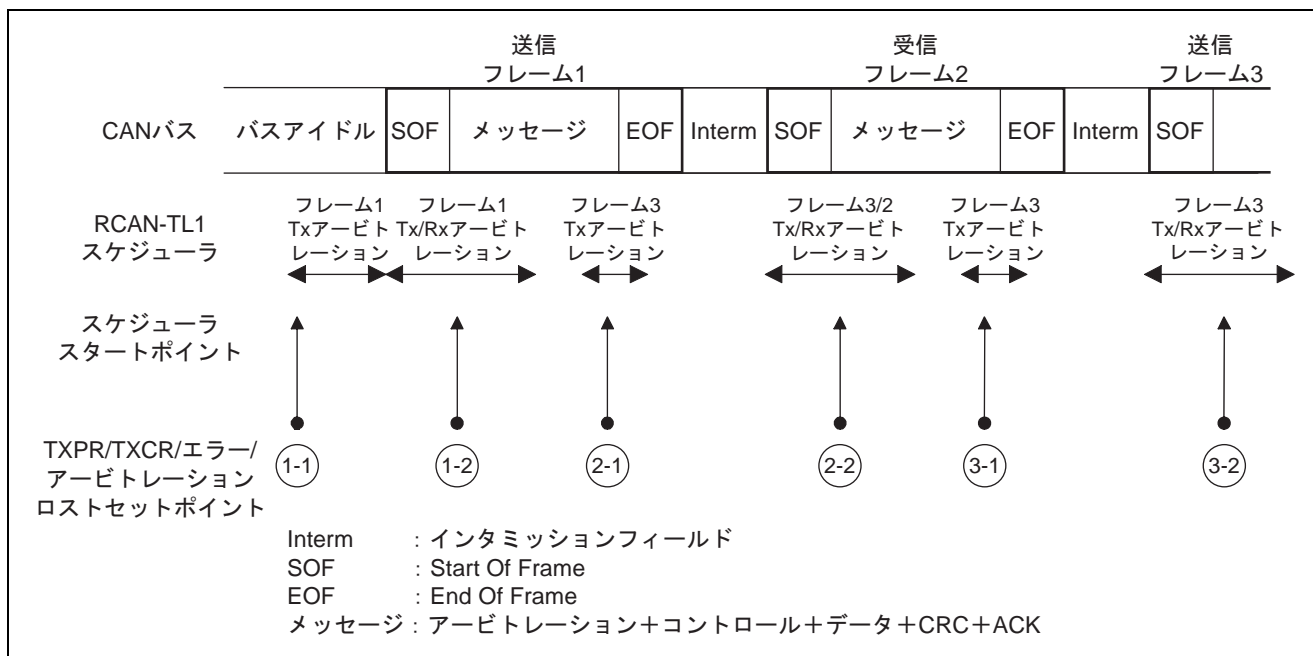


図 14.19 送信用内部アービトレーション

- 1-1 : CAN バスがアイドル状態のとき TXPR ビットがセットされると、ただちに内部アービトレーションを実行し、送信を開始します。
- 1-2 : SOF で送信、受信の両方の動作を開始します。受信フレームでないため、RCAN-TL1 は送信を行います。
- 2-1 : CRC のデリミタで内部アービトレーションが開始し、次に送信するメッセージを探します。
- 2-2 : SOF で送信、受信の両方の動作を開始します。優先度の高い受信フレームであるため、RCAN-TL1 は受信を行います。このため、フレーム 3 を送信せずに受信を行います。
- 3-1 : CRC のデリミタで内部アービトレーションが開始し、次に送信するメッセージを探します。
- 3-2 : SOF で送信、受信の両方の動作を開始します。送信フレームの優先度が受信フレームよりも高いため、RCAN-TL1 は送信を行います。

CAN バス上でエラーが検出された場合、次の送信用の内部アービトレーションは各エラーデリミタの先頭でも行われます。また、オーバーロードフレームの後のエラーデリミタの先頭でも行われます。

送信用アービトレーションは CRC のデリミタで行われるため、ATX=1 のメールボックスがリモートフレームリクエストを受信した場合、そのリクエストに応じて送信するメッセージが送信用アービトレーションに参加するタイミングは、その後のバスアイドルまたは CRC デリミタまたはエラーデリミタになります。

CAN バスの状態により、TXCR がセットされてから最大 1CAN フレームの遅延後に対応するメッセージのアポート処理が行われます。

(3) タイムトリガ送信

RCAN-TL1 は、ISO-11898-4 TTCAN Level 1 の仕様に準拠したタイムトリガモード通信を、ハードウェアでサポートしています。

ここでは、タイムトリガモードを使用するための基本的な手順を記します。

• タイムトリガモードの設定

タイムトリガモードをセットアップするには以下の設定が必要です。

- CMAX_TEWのCMAXをB'111以外の値に設定
- TTCR0のビット15をセットし、TCNTRの動作を開始
- コンペアマッチでTCNTRがクリアされないように、TTCR0のビット6をクリア
- タイムトリガ送信をするメールボックスのTXPRは、定周期送信を行うためにクリアされません。

• 各レジスタの役割

RCAN-TL1 のユーザレジスタを使用して、TTCAN 規格が要求する主な機能を扱うことができます。

TCNTR	Local_Time
RFMK	Ref_Mark
CYCTR	サイクルタイム = TCNTR - RFMK
RFTROFF	メールボックス 30 の Ref_Trigger_Offset
メールボックス 31	タイムリファレンスメッセージ専用受信メールボックス
メールボックス 30	タイムリファレンスメッセージ専用送信メールボックス (ポテンシャルタイムマスタとして動作する場合)
メールボックス 29~24	タイムトリガ送信対応のメールボックス
メールボックス 23~16	タイムスタンプなしの受信対応のメールボックス (イベントトリガ送信用メールボックスとして使用することもできます。)
メールボックス 15~0	タイムスタンプ付き受信対応のメールボックス (イベントトリガ送信用メールボックスとして使用することもできます。)
Tx-Trigger Time	メッセージを送信するタイミングを指定する Time_Mark
CMAX	ポテンシャルタイムマスタとして動作する場合のサイクルカウント数(Basic Cycle)の最大値を指定
TEW	Tx_Enable_Window 幅を指定
TCMR0	Init_Watch_Trigger (ローカルタイムとコンペアマッチ)
TCMR1	ユーザ指定イベント監視用にサイクルタイムとコンペアマッチ
TCMR2	Watch_Trigger (サイクルタイムとコンペアマッチ) 待ち状態にある送信をすべてアボートするように設定することができます。
TTW	送信に使用するタイムウィンドウの属性を指定
TTTSEL	次の送信待ちメールボックスを指定

• タイムマスタ/タイムスレーブ

RCAN-TL1 は、ネットワークのポテンシャルタイムマスタまたはタイムスレーブとして動作するように設定することができます。各モードに必要な設定と RCAN-TL1 が自動的に行う処理を下記の表に示します。

モード	設定	機能
タイムスレーブ	TXPR[30]=0 & MBC[30]≠B'000 & CMAx≠B'111 & MBC[31]=B'011	CANバス上でSOFが検出される度にTCNTRをサンプリングし、内部レジスタに格納します。メールボックス31が有効なタイムリファレンスメッセージを受信すると、TCNTRの値(SOFで格納)をRef_Mark(RFMK)にコピーします。受信したリファレンスメッセージのサイクルカウンタ値(CCR)を自ノードのサイクルカウンタレジスタ(CCR)にコピーします。Next_is_Gap=1ならばIRR13をセットします。
(ポテンシャル)タイムマスタ	TXPR[30]=1 & MBC[30]=B'000 & DLC[30]>0 & CMAx≠B'111 & MBC[31]=B'011	以下の2つの場合があります。 (1) メールボックス31が有効なタイムリファレンスメッセージを受信すると、SOFで内部レジスタに格納したTCNTRの値をRef_Mark(RFMK)にコピーします。受信したリファレンスメッセージのサイクルカウンタ値(CCR)を自ノードのサイクルカウンタレジスタ(CCR)にコピーし、Next_is_Gap=1ならばIRR13をセットします。 (2) メールボックス30からタイムリファレンスメッセージが送信されると、SOFで内部レジスタに格納したTCNTRの値をRef_Mark(RFMK)にコピーします。メールボックス30のTTTがCYCTRに一致するとサイクルカウンタ(CCR)をインクリメントします。サイクルカウンタ(CCR)はタイムリファレンスメッセージの最初のデータバイト(Data0[7:6]、CCR[5:0])に組み込まれます。

• 送信トリガタイムの設定

送信トリガタイム(TTT)は、以下に示すように昇順に設定してください。TTT間の差分は、次の式を満たす必要があります。式中のTEWはレジスタ値です。

$$\begin{aligned} & \text{TTT (メールボックス 24)} < \text{TTT (メールボックス 25)} < \text{TTT (メールボックス 26)} < \\ & \text{TTT (メールボックス 27)} < \text{TTT (メールボックス 28)} < \text{TTT (メールボックス 29)} < \\ & \text{TTT (メールボックス 30)} \end{aligned}$$

かつ

$$\text{TTT (メールボックス } i) - \text{TTT (メールボックス } i-1) > \text{TEW} + \text{最大フレーム長} + 9$$

ポテンシャルタイムマスタとして動作する場合、TTT (メールボックス 24) ~ TTT (メールボックス 29) は Time_Mark に、TTT (メールボックス 30) は Basic_Cycle length を示す Time_Ref に対応しています。

上記の制約は、タイムトリガ送信に設定されたメールボックスにのみ適用されます。

【重要】 送信トリガタイム設定の制約により、1つのタイムウィンドウに割り付けることのできるメールボックスは1つのみとなります。

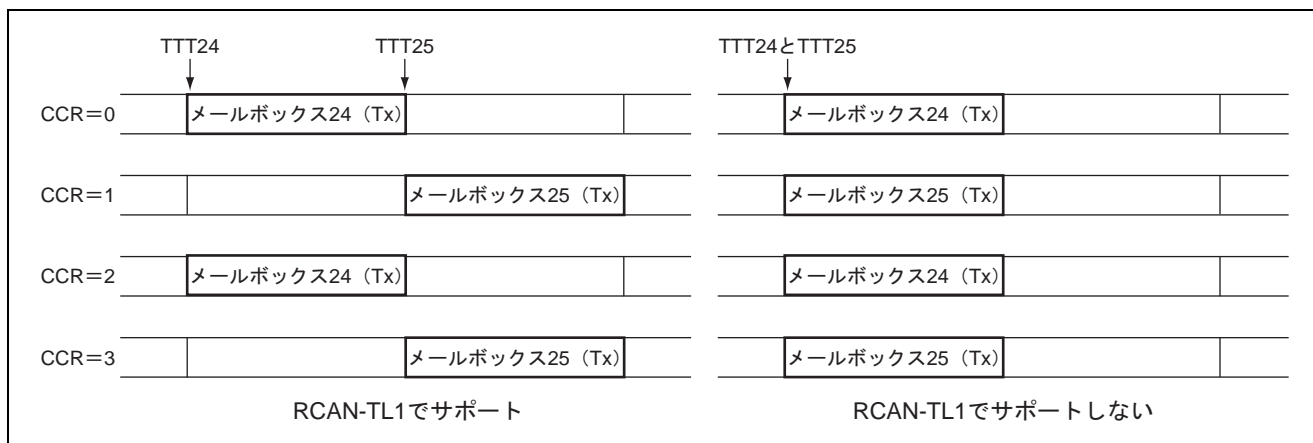


図 14.20 送信トリガタイムの制約

Watch_Trigger としての TCMR2 の値は、1Basic_Cycle length を示す TTT (メールボックス 30) より大きくなければなりません。

図 14.21、図 14.22 に、(ポテンシャル) タイムマスタおよびタイムスレーブの設定例を示します。図中の L は、タイムリファレンスメッセージの時間長です。

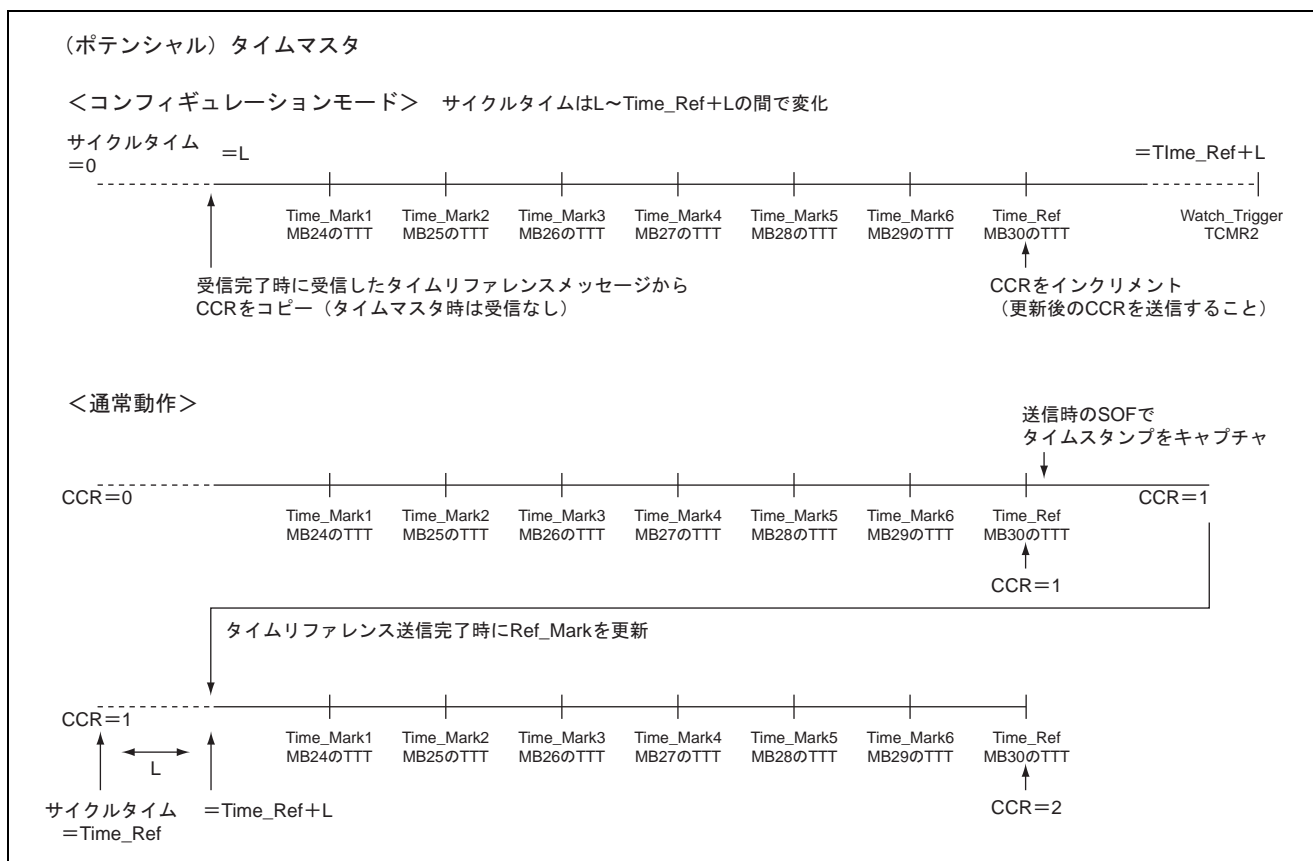


図 14.21 (ポテンシャル) タイムマスタ

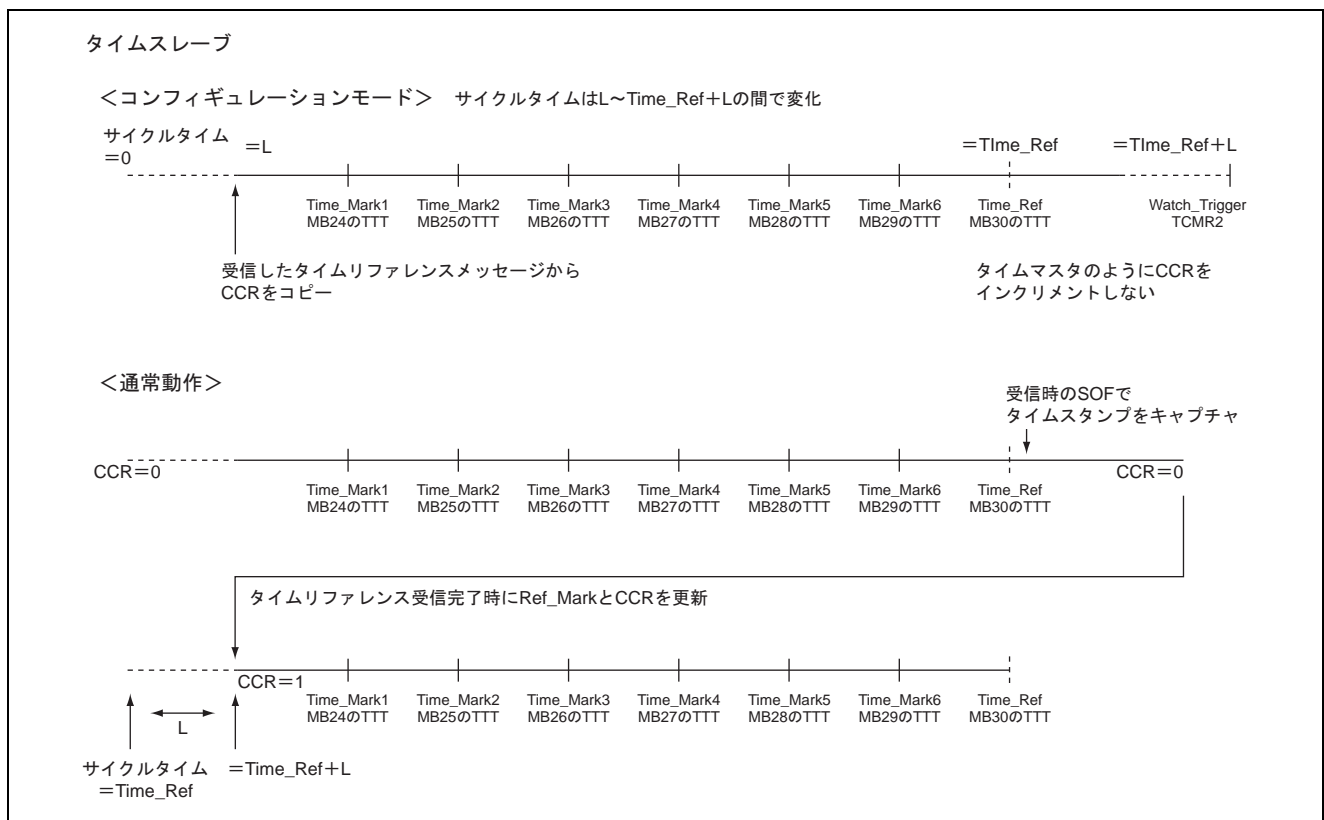


図 14.22 タイムスレーブ

- ソフトウェアで実装する機能

TTCAN の機能には、ソフトウェアで実装する必要のあるものがあります。主な機能を以下に示します。詳細は ISO-11898-4 を参照してください。

- Init_Watch_TriggerからWatch_Triggerへの変更

RCAN-TL1 では、Init_Watch_Trigger および Watch_Trigger のハードウェアサポートとしてそれぞれ TCMR0 レジスタと TCMR2 レジスタを用意しています。CAN バス上で最初のリファレンスメッセージが検出されるまで TCMR0 を有効かつ TCMR2 を無効にし、その後 TCMR0 を無効かつ TCMR2 を有効にする制御はソフトウェアで行います。(スケジュール同期化ステートマシーン)

Next_is_Gap 受信割り込みのみサポートしています。アプリケーションにて、対応する TXCR フラグをセットすることにより、現在の Basic_Cycle の終了時にすべての送信を停止する必要があります。

マスタ/スレーブモード制御

自動サイクルタイム同期と CCR のインクリメントのみサポートしています。

- メッセージステータスカウント

エクスクルーシブウィンドウの定周期メッセージに関するスケジュールエラーを、ソフトウェアでカウントする必要があります。

- タイムトリガ通信のメッセージ送信リクエスト

タイムトリガモードを使用する場合、ISO11898-4 の要求事項を満たさなければなりません。

以下の手順を行ってください。

1. RCAN-TL1をリセットまたはホルトモードにする
2. TCMR0にInit_Watch_Trigger (H'FFFF) を設定
3. TTCR0のビット10でTCMR0によるコンペアマッチを有効に設定
4. TCMR2に指定のWatch_Trigger 値を設定
5. TTCR0のビット12を0に保持し、TCMR2によるコンペアマッチを無効にする。
6. CMAXに必要な値を設定 (B'111以外)
7. TEWに必要な値を設定
8. メールボックスをタイムトリガ送信および受信に設定する
9. メールボックス31のLAFMを下位3ビットに対して設定
10. MCR、BCR1、BCR0に必要な値を設定
11. ポテンシャルタイムマスタとして動作する場合は下記を設定
 - RFTROFFに必要なInit_Ref_Offset値を設定
 - メールボックス30のTXPRをセット
 - TTTSELにH'4000を書き込む
12. TTCR0のビット15でタイマTCNTRをイネーブルにする
13. 通常モードに移行
14. 有効なリファレンスメッセージが送信または受信されるのを待つ、もしくはTCMR0のコンペアマッチを待つ
15. ローカルタイム (TCNTR) がTCMR0の値に一致したら、Init_Watch_Triggerに到達したことを意味し、アプリケーションでメールボックス30のTXCRをセットし再び開始する
16. リファレンスメッセージが送信されたら (TXACK[30]がセットされた)、RFTROFFを0に設定
17. 有効なリファレンスメッセージを受信したら (RXPR[31]がセットされた)、下記を行う
 - メールボックス31のIDの下位3ビットの優先度がメールボックス30の下位3ビットよりも高ければ (ポテンシャルタイムマスタとして動作する場合)、RFTROFFの値をInit_Ref_Offsetに保持
 - メールボックス31のIDの下位3ビットの優先度がメールボックス30の下位3ビットよりも低ければ (ポテンシャルタイムマスタとして動作する場合)、RFTROFFの値を1デクリメント
18. TTCR0のビット10をクリアし、TCMR0 によるコンペアマッチを無効にする
19. TTCR0のビット12をセットし、TCMR2 によるコンペアマッチを有効にする
20. CANバス上でリファレンスメッセージを2個検出 (送信または受信) した後、アプリケーションで他のタイムトリガメールボックスのTXPRをセットすることができます。

CANバス上でリファレンスメッセージが検出されずに、サイクルタイム CYCTR が TCMR2 の値に到達した場合はいつでも、RCAN-TL1 は待ち状態の送信 (リファレンスメッセージを含む) をすべて自動的にアボートします。

タイムトリガモードでさらに送信を要求するときのシーケンスを以下に示します。

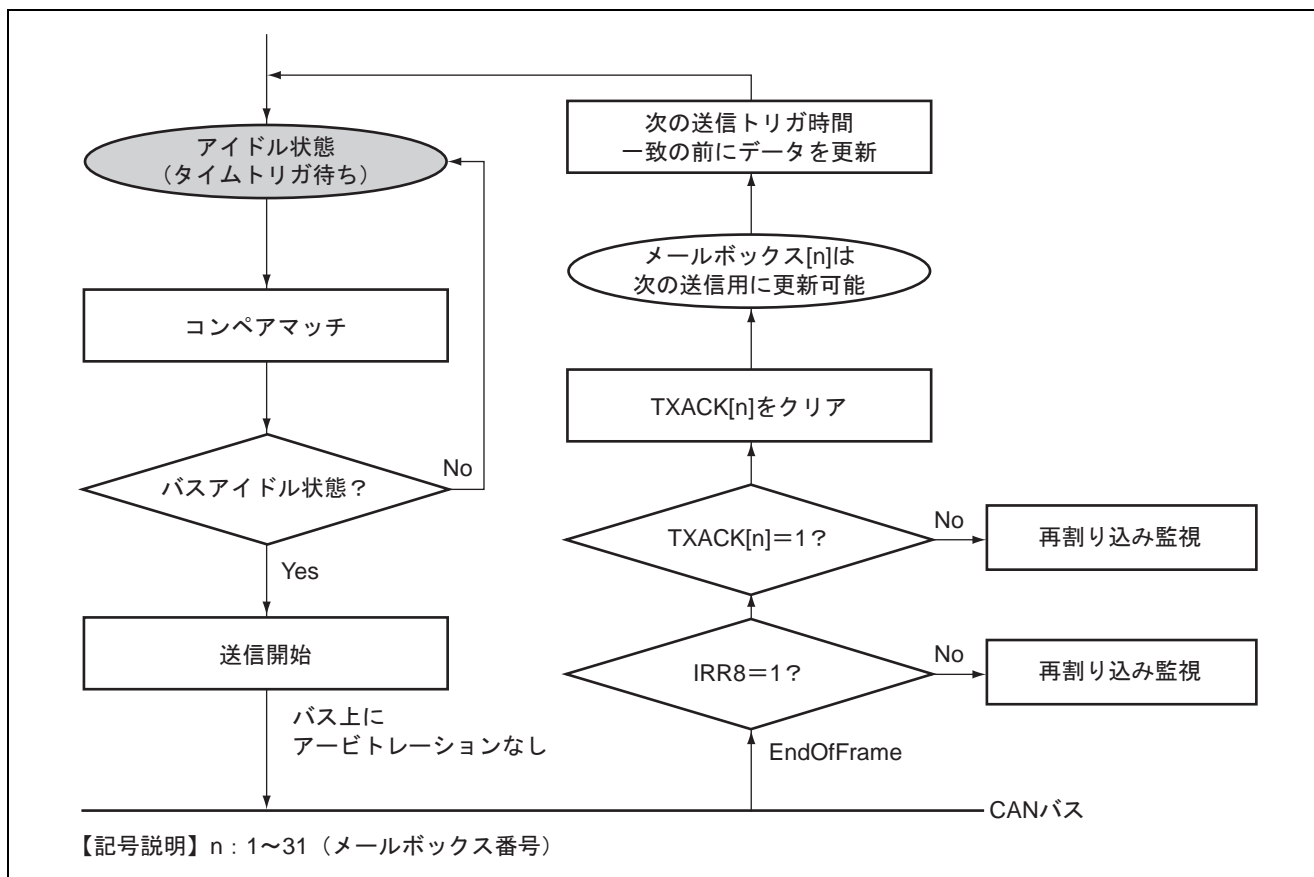


図 14.23 メッセージ送信リクエスト

ソフトウェアは、送信トリガが発生する前にメッセージの更新が確実に行われるようにしなければなりません。CYCTR がメールボックスの TTT (送信トリガタイム) に到達し、CCR がプログラムされた送信用サイクルに一致すると、RCAN-TL1 はただちに送信バッファにメッセージを転送します。

この時点で、RCAN-TL1 は指定された Tx_Enable Window 内で送信を試みます。このタイムスロットを逃すと、送信が定周期の場合 (メールボックス 24~30)、RCAN-TL1 は対応する TXPR ビットを 1 に保持し送信リクエストを次の送信トリガまで保留します。

RCAN-TL1 が上記のタイムスロットを逃す 3 つの要因があります。

1. CANバスが使用中
2. タイムトリガメッセージ送信中にCANバス上にエラーが発生
3. タイムトリガメッセージ送信中にアービトレーションロストが発生

マージドアービトレーティングウィンドウ (Merged Arbitrating Window) の場合、この送信スロットは、ウィンドウを開始した(TTW=B'10)メールボックスの送信トリガから、ウィンドウを終了する (TTW=B'11) メールボックスの TEW の最後までです。TXPR はいつでも変更可能です。RCAN-TL1 は、タイムトリガメッセージの送信が常に正しくスケジュールされるようにしますが、正確なスケジュールを保証するため、以下に示す重要な規則があります。

- TTT (送信トリガタイム) は、コンフィギュレーションモードで変更可能。
- Basic_Cycle length サイクルカウンタ (Basic Cycle) 長を指定する Time_Ref を超える TTT の設定は不可。違反すると、スケジューリングで問題が発生します。
- 定周期送信では、TXPR は自動的にクリアされません。定周期送信をキャンセルする必要がある場合は、アプリケーションで対応する TXCR ビットをセットします。

• タイムトリガシステムの例

タイムスレーブモードの RCAN-TL1 を使用して、タイムトリガシステムが動作する簡単な例を下図に示します。

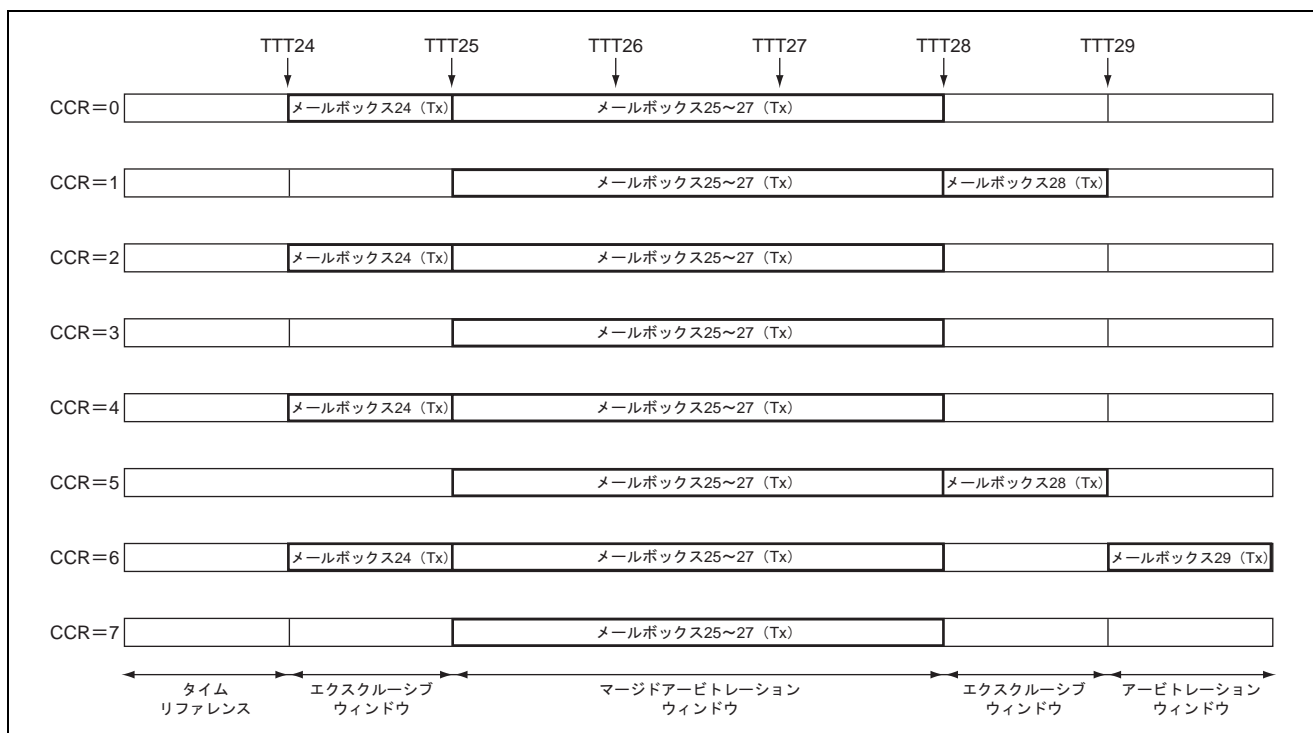


図 14.24 タイムスレーブとしてのタイムトリガシステム例

図 14.24 の例で使用する値を表 14.11 に示します。

表 14.11 例で使用する設定値

	rep_factor (レジスタ)	Offset		TTW[1:0]	MBC[2:0]
メールボックス 24	B'001	B'000000		B'00	B'000
メールボックス 25	B'000	B'000000		B'10	B'000
メールボックス 26	B'000	B'000000		B'10	B'000
メールボックス 27	B'000	B'000000		B'11	B'000
メールボックス 28	B'010	B'000001		B'00	B'000
メールボックス 29	B'011	B'000110		B'01	B'000
メールボックス 30	-	-		-	B'111
メールボックス 31	-	-		-	B'011

【注】 CMAX=B'011, TXPR[30]=0

マージドアービトレイティングウィンドウ内では、タイムトリガ送信リクエストは FCFS (First Come First Served) で処理されます。たとえば、メールボックス 25 が送信トリガタイム 25 (TTT25) と送信トリガタイム 26 (TTT26) の間で送信できなかった場合、TTT26-TTT28 間ではメールボックス 25 はメールボックス 26 より優先度が高くなります。

タイムトリガ送信を無効にするには、MBC を B'111 に設定します。RCAN-TL1 がタイムマスタの場合、MBC[30]=B'000 としなければならず、タイムリファレンスウィンドウは自動的にアービトレイティングウィンドウと認識されます。

• タイマ動作

図 14.25 にタイマのタイミング図を示します。送信トリガタイム=n と設定すると、タイムトリガ送信は、CYCTR=n+2 から CYCTR=n+3 の間に開始します。

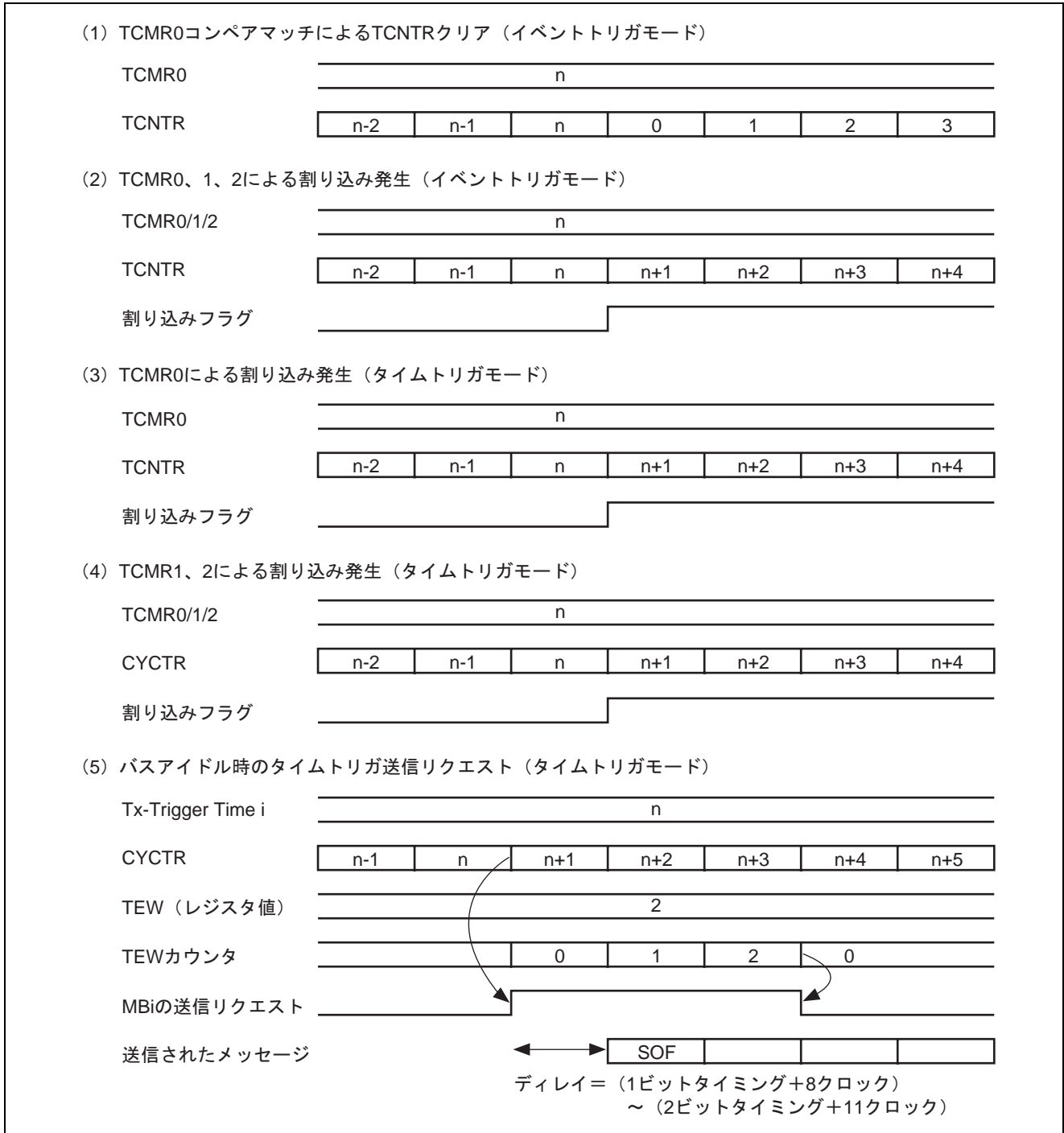


図 14.25 タイマのタイミング図

マージドアービトレイティングウィンドウ内では、イベントトリガ送信はタイマトリガ送信完了後に処理されます。たとえば、メールボックス 25 の送信が完了したときに **CYCTR** が **TTT26** に到達していないと、**MCR2** で指定したメッセージ送信の優先順位により、イベントトリガ送信が開始します。タイマトリガ送信の **TXPR** は送信完了後にクリアされませんが、イベントトリガ送信の **TXPR** は送信完了後にクリアされます。

マージドアービトレイティングウィンドウを閉じるメールボックスの **TXPR** がセットされない場合は、そのメールボックスの **TTT** の後に続く **TEW** の最後でマージドアービトレイティングウィンドウが閉じます。

「14.3.1 メールボックスの構成」の表 14.3 を参照してください。

14.6.4 メッセージ受信シーケンス

メッセージ受信シーケンスを図 14.26 に示します。

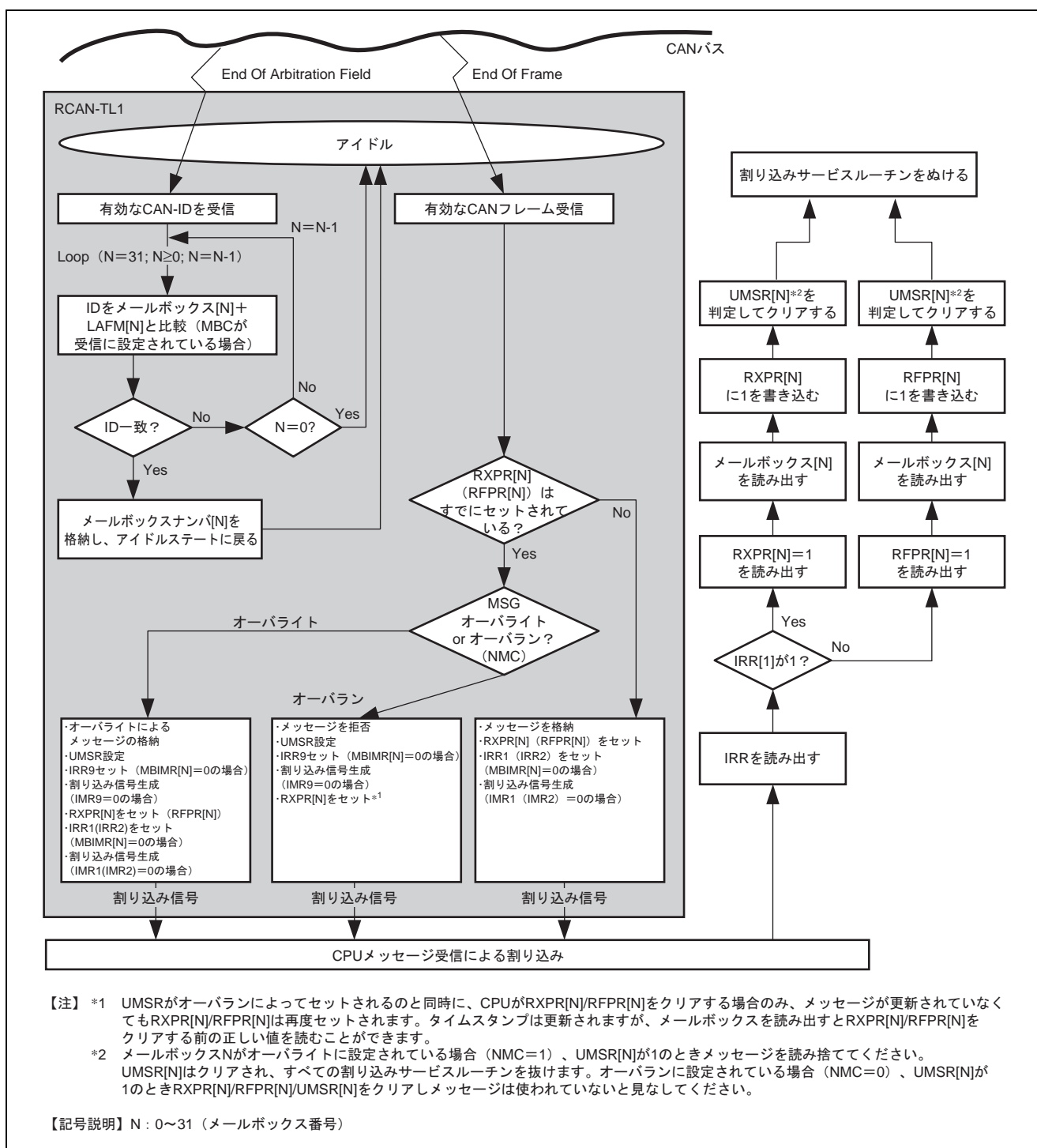


図 14.26 メッセージ受信シーケンス

メッセージを受信中に RCAN-TL1 がアービトレーションフィールドの最後を認識すると、受信した ID とメールボックスに設定された ID の比較を始めます。比較する順序はメールボックス 31 からメールボックス 0 の順です。まず MBC をチェックし、メールボックスが受信に設定されているか調べます。その後 LAFM を読み込み、続いてメールボックス 31 (受信用に設定されている場合) の ID を読み込み、受信した ID と比較します。一致しない場合は、メールボックス 30 (受信用に設定されている場合) に対して同じチェックを行います。一致する ID を見つけると RCAN-TL1 はそのメールボックス番号 (N) を内部バッファに格納し、サーチを停止した後アイドル状態に戻り EOF (End Of Frame) を待ちます。EOF の 6 ビット目が通知されると、受信メッセージは NMC ビットの設定により書き込まれるか廃棄されます。

通信中に RCAN-TL1 のメッセージ ID と LAFM の設定を変更することはできません。設定変更する手段の 1 つとしてホルトモードおよびコンフィギュレーションモードがあります。受信メッセージに対応するメールボックスに書き込むときに、メッセージ ID を含めて書き込まれるため、LAFM を使用するときは CAN-ID が異なる CAN-ID でオーバーライトされることがあります。これはまた、受信したメッセージの ID が複数のメールボックスの ID + LAFM と一致する場合には、受信メッセージは常にメールボックス番号の一番大きいメールボックスに格納され、小さい番号のメールボックスにはメッセージが格納されなくなってしまうことを意味しています。したがって、ID と LAFM の設定値は注意深く選択する必要があります。

図 14.11 に示すデータおよびリモートフレームの受信で、IRR をリードした後に UMSR フラグをクリアするのは、割り込みサービスルーチンの実行中にメッセージが同じメールボックスに格納された新しいメッセージでオーバーライトされること (NMC が 1 のとき) を検出するためです。UMSR の最後のチェック中にオーバーライトが検出された場合、メッセージを破棄し再度読み込む必要があります。

メールボックスがオーバーランに (NMC=0) 設定されているときは、UMSR がセットされた場合のメッセージは有効です。しかし、CAN バスでモニタされた最新のメッセージではなく古いメッセージです。メッセージの読み出しは、関連する RXPR/RFPR フラグをクリアする前に行ってください。

受信したリモートフレームがデータフレームでオーバーライトされた場合、リモートフレーム受信割り込み (IRR2) およびデータフレーム受信割り込み (IRR1) がセットされ、受信フラグ (RXPR、RFPR) もセットされますので、ご注意ください。同様にデータフレームがリモートフレームでオーバーライトされた場合も IRR2 と IRR1 がセットされます。

メッセージを受信しメールボックスに格納すると、受信されなかったデータフィールドはすべて 0 として格納されます。スタンダード ID を受信した場合も同様です。エクステンデッド ID (EXTID[17:0]) には 0 が書き込まれます。

14.6.5 メールボックスの再設定

メールボックスの再設定が必要な場合は、下記の手順に従ってください。

(1) 送信ボックスの設定変更

下記の2つの場合があります。

- ID、RTR、IDE、LAFM、データ、DLC、NMC、ATX、DARTの変更
MBC=B'000 の場合のみ変更可能です。対応する TXPR ビットがセットされていないことを確認してください。設定 (MBC を除く) はいつでも変更することができます。
- 送信ボックスから受信ボックスへの設定変更
対応する TXPR ビットがセットされていないことを確認してください。ホルトモードまたはリセット状態でのみ変更可能です。RCAN-TL1 がメッセージを受信中または送信中の場合、ホルト状態に遷移するのに時間がかかることがあります (受信/送信が終了するのを待ってからホルト状態に遷移するためです)。また、ホルト状態ではメッセージの送受信ができませんのでご注意ください。
RCAN-TL1 がバスオフ状態の場合、ホルト状態への遷移は MCR レジスタのビット 6 およびビット 14 の設定に従います。

(2) 受信ボックスの ID、RTR、IDE、LAFM、データ、DLC、NMC、ATX、DART、MBC の設定変更 および受信ボックスから送信ボックスへの変更

設定変更はホルトモードでのみ可能です。メッセージが CAN バス上にあり RCAN-TL1 が受信モードの場合、そのメッセージを逃すことはありません。RCAN-TL1 は現在行っている受信を完了してからホルトモードに遷移します。RCAN-TL1 がメッセージを受信中または送信中の場合、ホルト状態に遷移するのに時間がかかることがあります (受信/送信が終了するのを待ってからホルト状態に遷移するためです)。また、ホルト状態ではメッセージの送受信ができませんのでご注意ください。

RCAN-TL1 がバスオフ状態の場合、ホルト状態への遷移は MCR レジスタのビット 6 およびビット 14 の設定に従います。

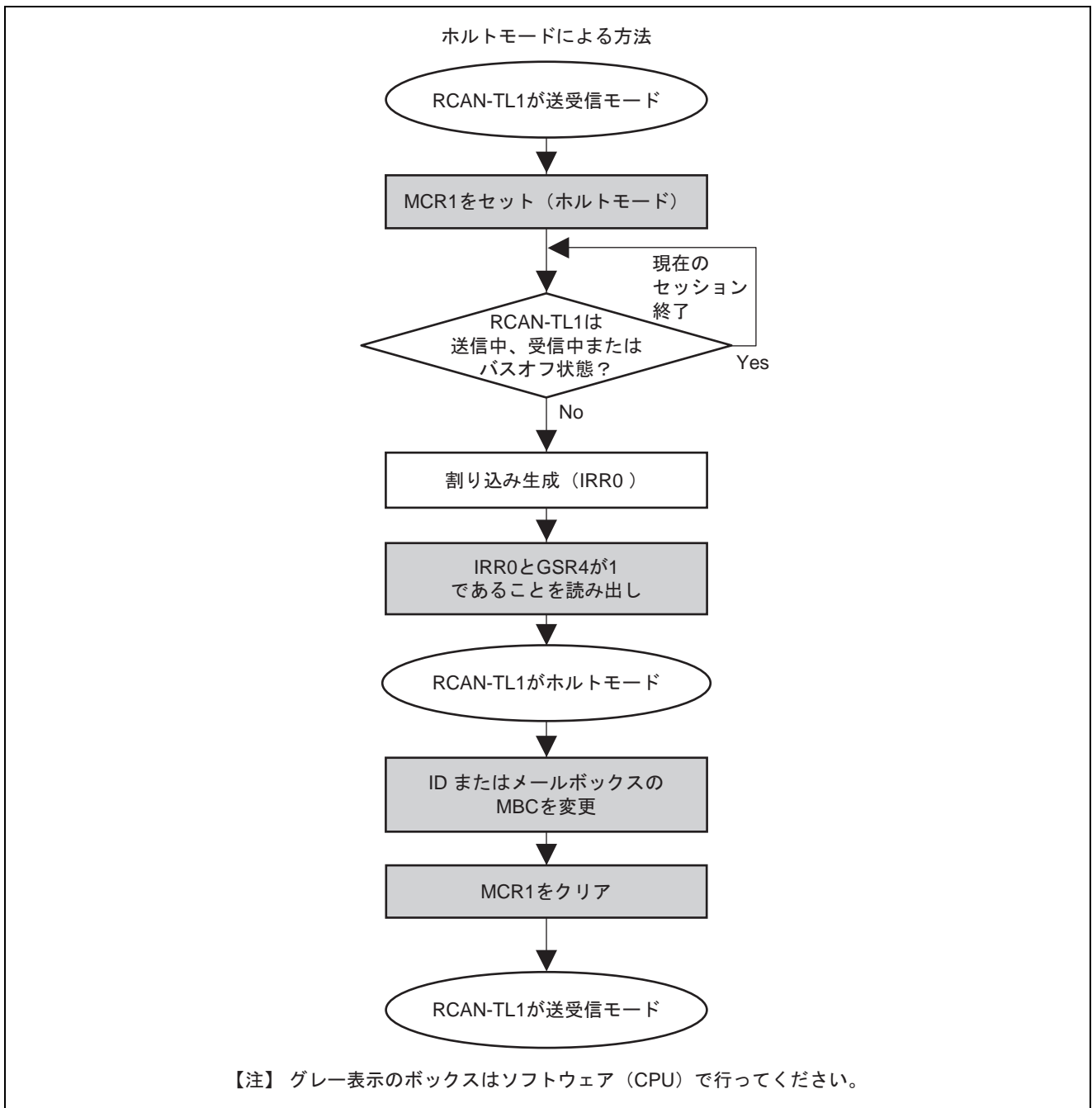


図 14.27 受信ボックスの ID 変更／受信ボックスから送信ボックスへの変更

14.7 割り込み要因

RCAN-TL1 には表 14.12 に示す割り込み要因があります。これらの要因はマスクすることができます。マスクには、メールボックスインタラプトマスクレジスタ (MBIMR) およびインタラプトマスクレジスタ (IMR) を使用します。各割り込み要求の割り込みベクタについては「第 5 章 割り込みコントローラ」を参照してください。

表 14.12 RCAN-TL1 の割り込み要因

モジュール名	名称	要 因	割り込みフラグ	DMAC の起動	
RCANn* ¹	ERSn* ¹	エラーパッシブ (TEC \geq 128 または REC \geq 128)	IRR5	不可	
		バスオフ (TEC \geq 256) /バスオフからの復帰	IRR6		
		エラーワーニング (TEC \geq 96)	IRR3		
		エラーワーニング (REC \geq 96)	IRR4		
	OVRn* ¹	リセット/ホルト/CAN スリープ遷移	IRR0		
		オーバロードフレーム送信	IRR7		
		未読メッセージのオーバーライト (オーバーラン)	IRR9		
		スタートシステムマトリックス	IRR10		
		TCMR2 コンペアマッチ	IRR11		
		CAN スリープ中 CAN バス動作の検出	IRR12		
		タイマオーバーラン/Next_is_Gap/メッセージエラー	IRR13		
		TCMR0 コンペアマッチ	IRR14		
		TCMR1 コンペアマッチ	IRR15		
	RMn0* ^{1*2}	データフレーム受信	IRR1* ³		可* ⁴
	RMn1* ^{1*2}	リモートフレーム受信	IRR2* ³		
SLEn* ¹	メッセージの送信/送信取り消し (スロットエンプティ)	IRR8	不可		

【注】 *1 n=0、1

- *2 RM0 はメールボックス 0 のリモートフレーム受信フラグ (RFPR0[0]) またはデータフレーム受信フラグ (RXPR0[0]) による割り込み、RM1 はメールボックス n (n=1~31) のリモートフレーム受信フラグ (RFPR0[n]) またはデータフレーム受信フラグ (RXPR0[n]) による割り込みです。
- *3 IRR1 はメールボックス 0~31 のデータフレーム受信フラグ、IRR2 はメールボックス 0~31 のリモートフレーム受信フラグです。
- *4 RMn0 割り込みのみ DMAC を起動できます。

14.8 DMAC インタフェース

各 RCAN-TL1 のメールボックス 0 にメッセージを受信すると、DMAC を起動することができます。なお、DMAC 起動を設定し、DMAC による転送が終了すると、自動的に RXPR0 と RFPR0 レジスタのフラグはクリアされます。このとき、RCAN-TL1 からの受信割り込みで CPU への割り込みは発生しません。図 14.28 に DMAC の転送フローチャートを示します。

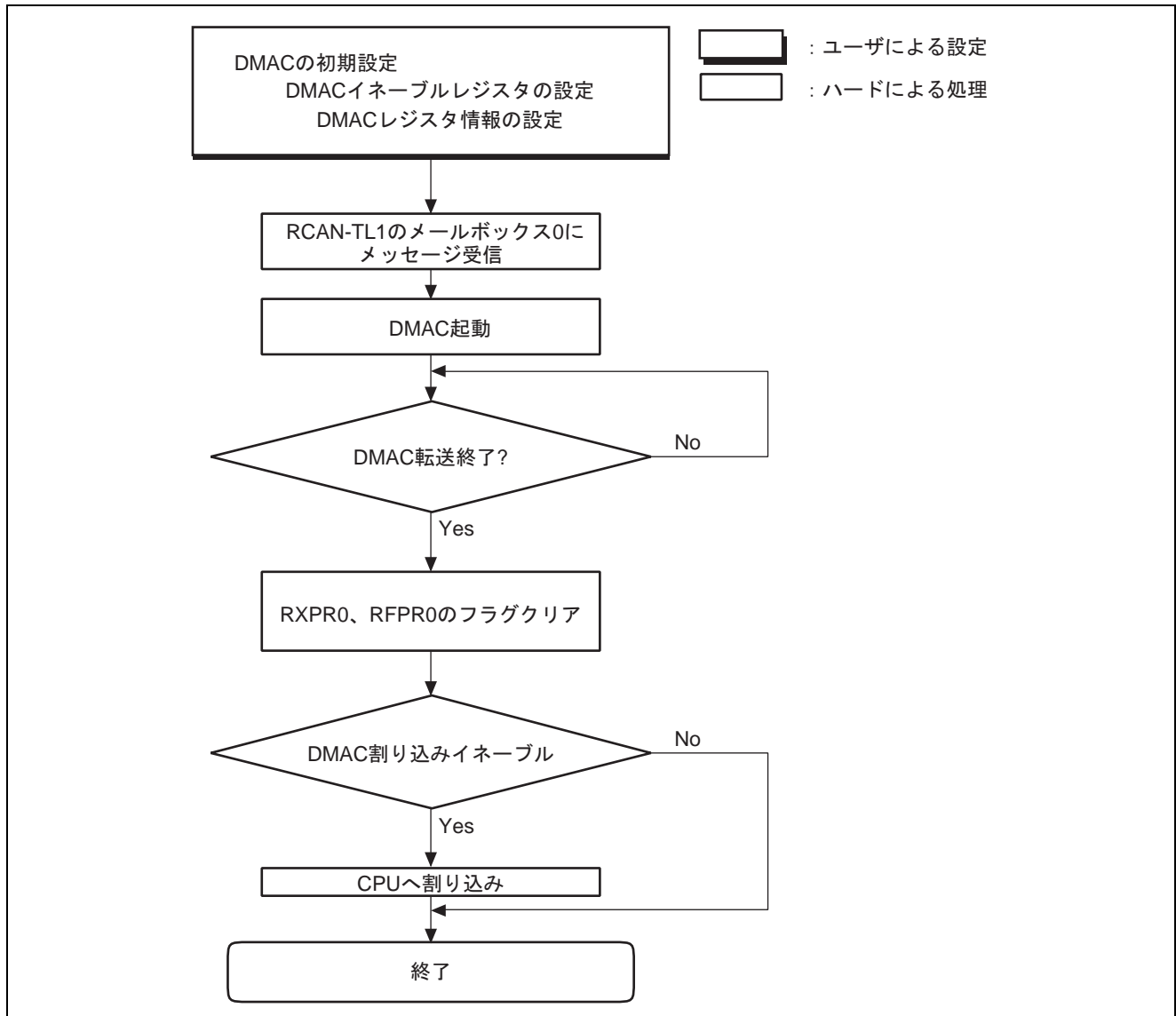


図 14.28 DMAC の転送フローチャート

14.9 PORT インタフェース

RCAN-TL1 の端子を制御する RCAN モニタレジスタ (RCANMON) があります。

14.9.1 RCAN モニタレジスタ (RCANMON)

RCAN-TL1 モニタレジスタ (RCANMON) は送信端子の送信停止制御、RCAN-TL1 の送受信端子の許可/禁止、RCAN-TL1 の端子の状態をモニタします。本レジスタのアクセスは CAN スリープの影響を受けません。また、ソフトウェアリセット (MCR0) による初期化の影響も受けません。

ビット	7	6	5	4	3	2	1	0
ビット名	—	CTxSTP	RCANE	—	—	—	CTxD	CRxD
初期値 :	0	0	0	0	0	0	不定	不定
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R	R

ビット	ビット名	初期値	R/W	説明
7	—	0	R	リザーブビット 読み出すと常に0が読み出されます。書き込む値は常に0にしてください。
6	CTxSTP	0	R/W	RCAN 送信停止 このビットを1にセットすると、RCAN-TL1 の送信データにかかわらず、CTx 端子のデータを1にします。
5	RCANE	0	R/W	RCAN-TL1 送受信端子イネーブル このビットを1にセットすると、RCAN-TL1 の CTx 端子、CRx 端子を有効にします。
4~2	—	0	R	リザーブビット 読み出すと常に0が読み出されます。書き込む値は常に0にしてください。
1	CTxD	不定	R	RCAN 送信データモニタ リードすると CTx の状態が読み出されます。ライトは無効です。
0	CRxD	不定	R	RCAN 受信データモニタ リードすると CRx の状態が読み出されます。ライトは無効です。

RCANMON 各ビットの PORT インタフェースの概要を図 14.29 に示します。

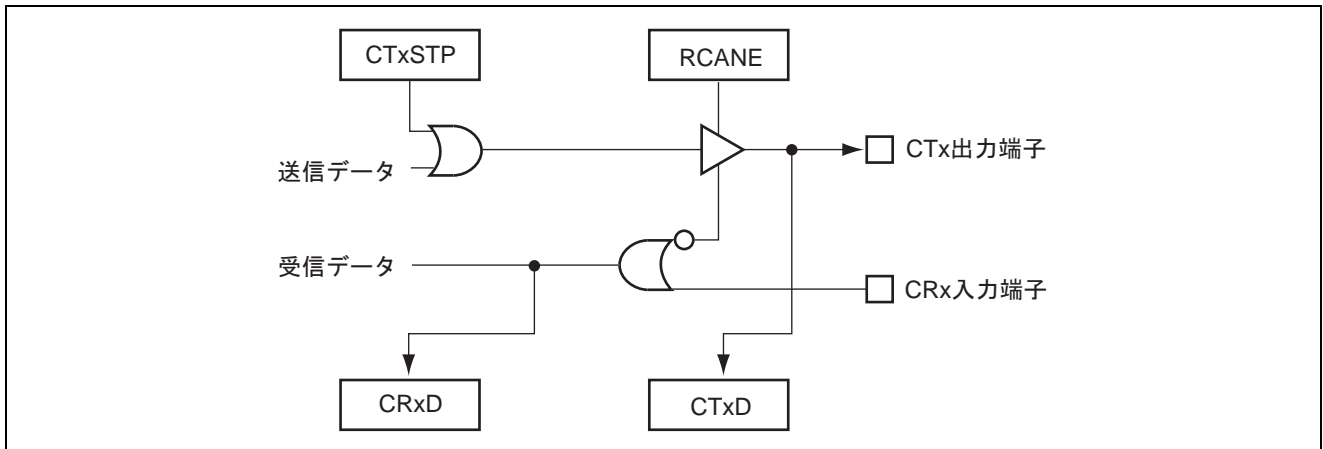


図 14.29 PORT インタフェース概要

14.10 CAN バスインタフェース

本 LSI と CAN バスを接続するためにはバストランシーバ IC が必要になります。トランシーバ IC はルネサス HA13721 を推奨します。HA13721 以外の製品を使用する場合は、HA13721 とコンパチブルな製品を使用してください。図 14.30 に接続例を示します。

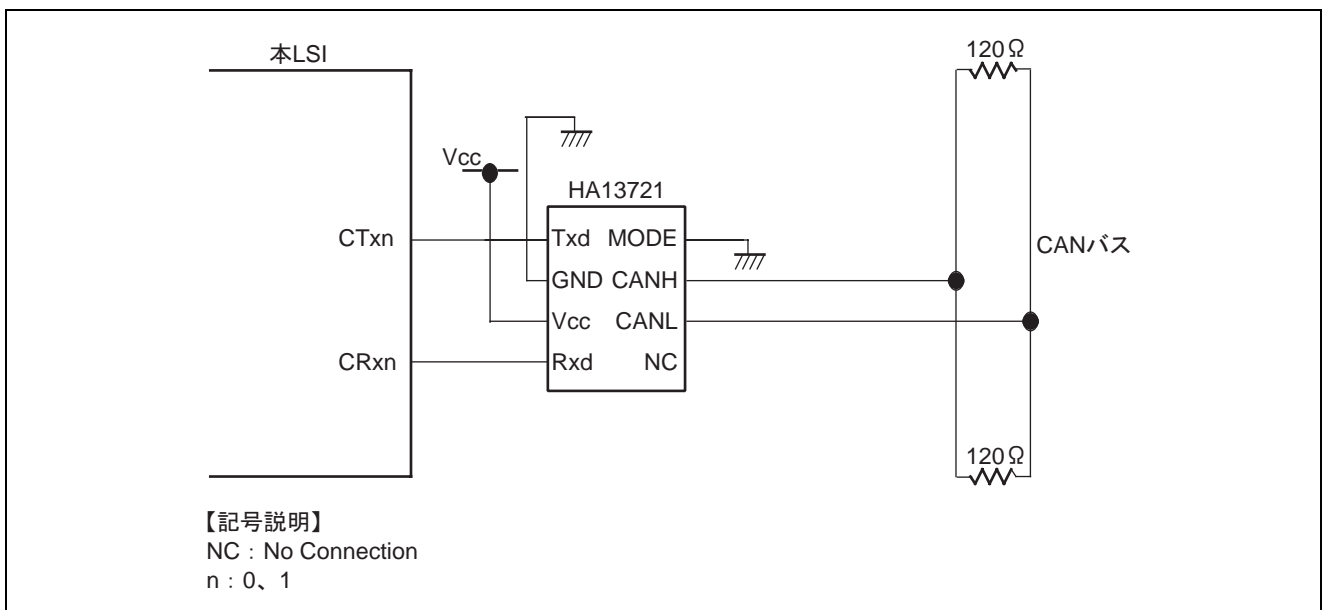


図 14.30 HA13721 を用いたハイスピード CAN インタフェース

14.11 RCAN-TL1 端子ポート設定

RCAN-TL1 ポート設定は、必ずコンフィギュレーションモード中またはそれ以前に行ってください。ポート設定方法の詳細は「9.3.1 ポートファンクションコントロールレジスタ 5 (PFCR5)」を参照してください。本 LSI は 2 チャンネルの RCAN-TL1 を内蔵しており、2 種類の使用方法があります。

- 2チャンネルの32メールボックス (RCAN0、1)
- 1チャンネルの64メールボックス (RCAN0+1)

【注】 64 メールボックス使用時には注意が必要です。「14.12.1 1チャンネル 64 メールボックスでのポート設定についての注意事項」を必ずお読みください。

図 14.31、図 14.32 に各ポート設定での接続例を示します。

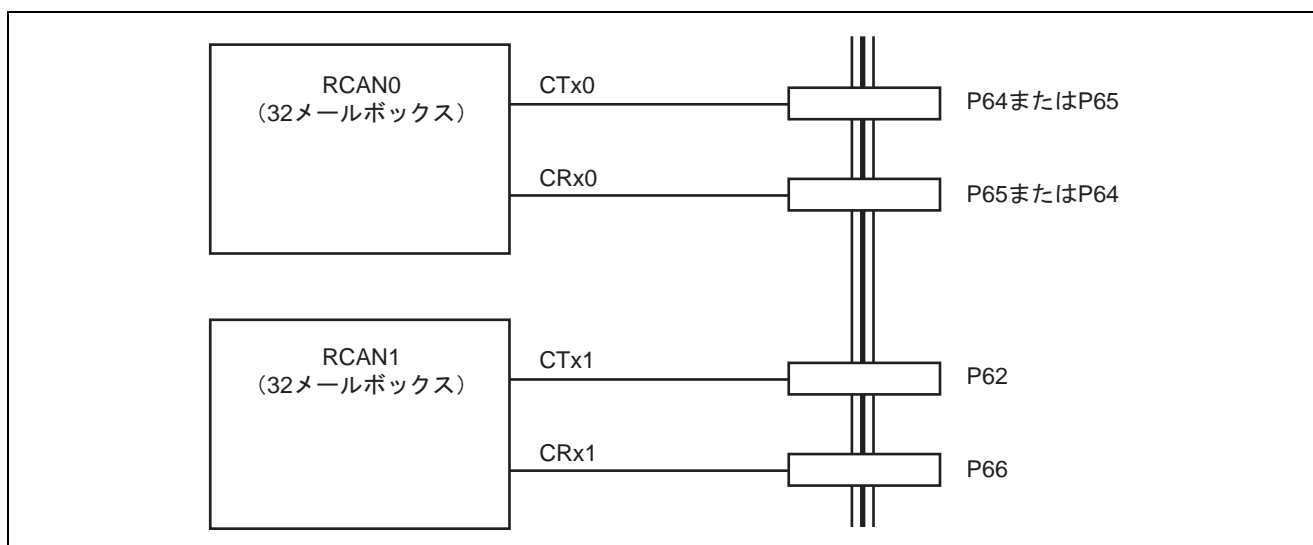


図 14.31 RCAN0、1 を個別のチャンネルとして使用するときの接続例

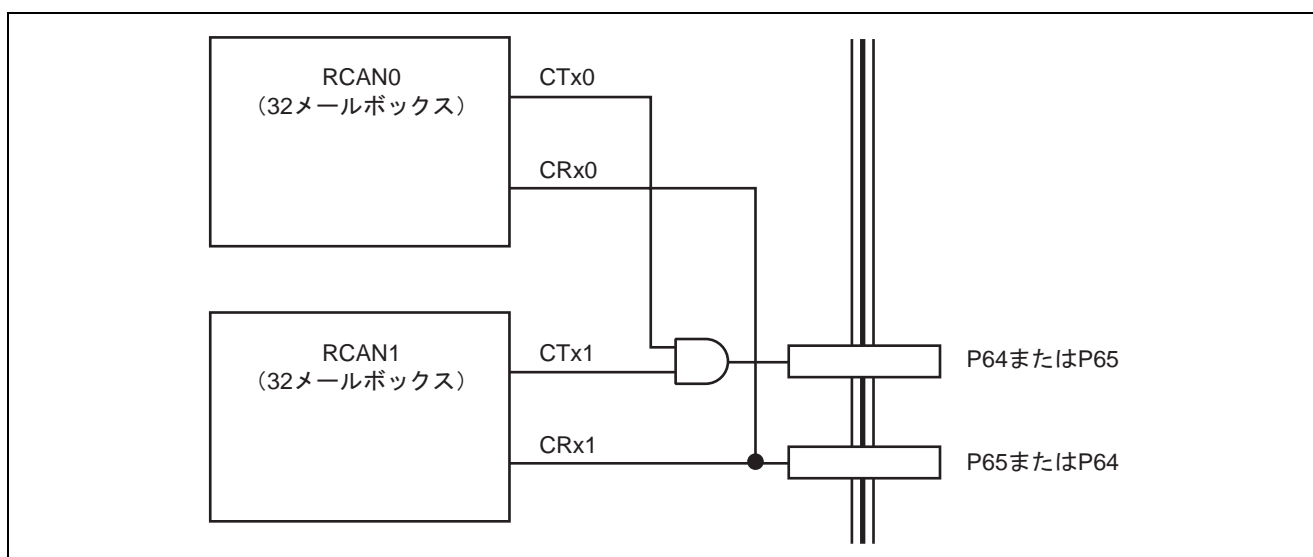


図 14.32 RCAN0、1 を 64 メールボックスの 1 チャンネルとして使用するときの接続例

14.12 使用上の注意事項

14.12.1 1チャンネル64メールボックスでのポート設定についての注意事項

本 LSI は、2チャンネルの RCAN-TL1 を内蔵しています。1チャンネルで64メールボックスの設定で使用する際、以下の注意事項があります。

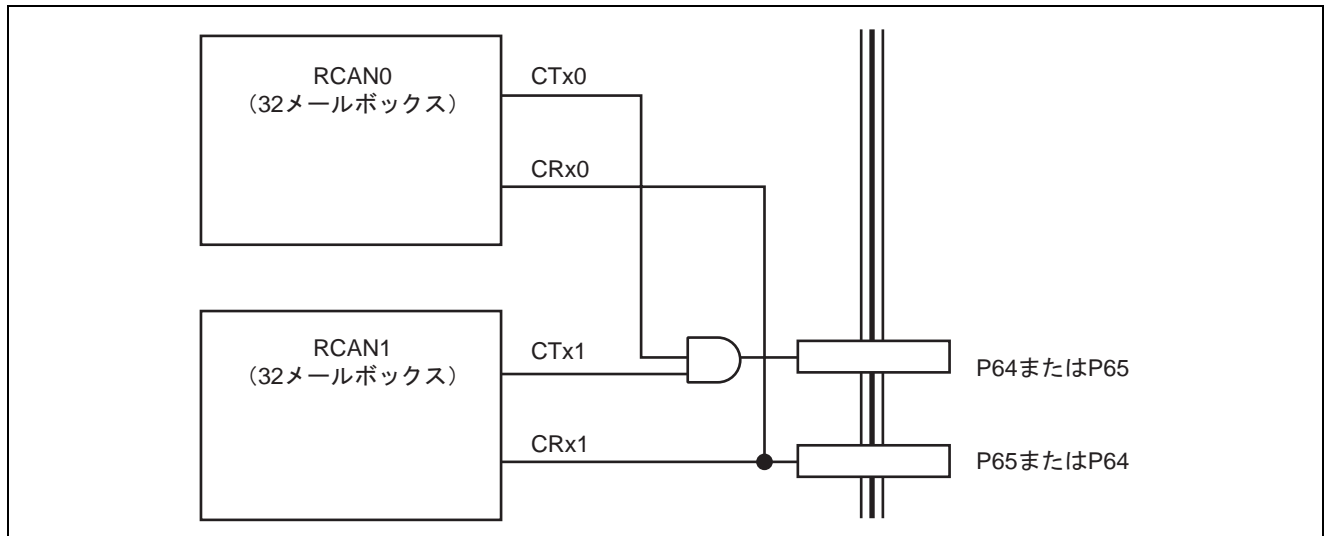


図 14.33 RCAN0、1 を 64 メールボックスの 1 チャンネルとして使用するときの接続例

1. CANバスに他ノードを接続しない状態でメッセージ送信した場合、ACKエラーが発生しません。上図の RCAN0 からメッセージを送信した場合、ACK フィールドで RCAN_B が ACK を送信するためです。RCAN1 は CANバス上のメッセージを受信しており、CAN プロトコルに従い ACK フィールドで ACK を送信し、その ACK を RCAN0 が受信します。
対応方法は、メッセージを送信しない RCAN_1 をテストモードのリスンオンリモード状態 (TST[2:0]=B'001) またはリセット状態 (MCR0=1 設定) にしてください。これにより、メッセージを送信しないチャンネルから ACK を送信しません。
2. 送信順位を決定する内部アービトレーションは、それぞれ RCAN0 と RCAN1 で独立に実施します。送信可能バッファは、RCAN-TL1 には 31 メールボックス/チャンネルありますが、62 バッファの範囲で内部アービトレーションは実施しません。
3. 同一送信メッセージ ID を RCAN0 と RCAN1 に設定しないでください。CANバス上でのアービトレーション実施後、2つのチャンネルからメッセージを送信することになります。

14.12.2 モジュールストップモード

モジュールストップコントロールレジスタ (MSTPCRE) により、RCAN-TL1 に対するクロックの動作/停止を設定することが可能です。初期値では RCAN-TL1 のクロックは停止しています。RCAN-TL1 のレジスタのアクセスは、モジュールストップモードを解除してから行ってください。

14.12.3 リセット

RCAN-TL1 のリセットにはハードウェアリセットとソフトウェアリセットがあります。

- **ハードウェアリセット**

パワーオンリセット、ハードウェアスタンバイ、モジュールストップ、ソフトウェアスタンバイでは RCAN-TL1 は初期化されます。

- **ソフトウェアリセット**

マスタコントロールレジスタ (MCR) の MCR0 ビットにより、MCR0 ビット以外のレジスタおよび CAN 通信機能が初期化されます。

リセット時にはインタラプトリクエストレジスタ (IRR) の IRR0 ビットが初期化によりセットされますので、リセットシーケンスに示されたコンフィギュレーションモード時にクリアしてください。

メールボックスのメッセージコントロールフィールド 1 (CONTROL1)、タイムスタンプ (TIMESTAMP)、送信トリガタイム (TTT)、タイムトリガコントロール (TTCONTROL) を除いた領域は RAM で構成されていますので、リセットにより初期化されません。パワーオンリセット後は、リセットシーケンスに示されたコンフィギュレーションモード時にすべてのメールボックスを初期設定してください。

14.12.4 CAN スリープモード

CAN スリープモードでは、主要な部分のクロックをモジュール内部で停止しています。このため、CAN スリープモードで MCR、GSR、IRR、IMR レジスタ以外へのアクセスはしないでください。

14.12.5 レジスタアクセス

RCAN-TL1 内部の CAN 通信機能が CAN バス受信フレームをメールボックスに格納している期間に、メールボックス領域をアクセスすると 0~5 周辺バスサイクル分のウェイトが発生します。

14.12.6 割り込み

メールボックス 0 受信割り込みは、表 14.12 に示したように DMAC の起動が可能です。メールボックス 0 受信割り込みを起動要因にして DMA 転送時の割り込み要因をクリアする設定を選択した場合には、ブロック転送モードなどを利用して、メールボックス 0 のタイムスタンプ (TIMESTAMP) までリードしてください。

CPU によって、表 14.12 の割り込み要因フラグをクリアするときは、割り込み要因フラグのクリアから割り込みコントローラでの割り込みがクリアされるまでの間に CPU の RTE 命令が実行されるのを防ぐため、割り込みサービスルーチン内でフラグをクリアした後にフラグをリードしてください。

15. ルネサスシリアルペリフェラルインタフェース (RSPI)

本 LSI は、独立した 4 チャンネルのルネサスシリアルペリフェラルインタフェース (RSPI) を備えています。RSPI は、全二重同期式のシリアル通信、送信のみの動作ができます。複数のプロセッサや周辺デバイスとの高速なシリアル通信機能を備えています。

15.1 特長

本 LSI の RSPI には次のような特長があります。

- RSPI 転送機能

MOSI (Master Out Slave In)、MISO (Master In Slave Out)、SSL (Slave Select)、RSPCK (RSPI Clock) 信号を使用して、SPI 動作 (4 線式) / クロック同期式動作 (3 線式) でシリアル通信が可能。

送信のみの動作が可能

マスタ/スレーブモードでのシリアル通信が可能。

モードフォルトエラー検出が可能。

オーバランエラー検出が可能。

シリアル転送クロックの極性を変更可能。

シリアル転送クロックの位相を変更可能。

- データフォーマット

MSB ファースト / LSB ファーストの切り替え可能。

転送ビット長を 8、9、10、11、12、13、14、15、16、20、24、32 ビットに変更可能。

送信/受信バッファは 128 ビット。

一度の送受信で最大 4 フレームを転送 (1 フレームは最大 32 ビット)。

- バッファ構成

送信/受信バッファ構成はダブルバッファ。

- SSL制御機能

RSPI 1 チャンネルあたり 4 本の SSL 信号 (SSL0~SSL3) あり。

シングルマスタ設定時には、SSL0~SSL3 信号を出力。

マルチマスタ設定時には、SSL0 信号は入力、SSL1~SSL3 信号は出力または Hi-Z。

スレーブ設定時には、SSL0 信号は入力、SSL1~SSL3 信号は Hi-Z。

SSL 出力のアサートから RSPCK 動作までの遅延 (RSPCK 遅延) を設定可能。

設定範囲: 1~8 RSPCK 設定単位: 1 RSPCK

RSPCK 停止から SSL 出力のネゲートまでの遅延 (SSL ネゲート遅延) を設定可能。

設定範囲: 1~8 RSPCK 設定単位: 1 RSPCK

次アクセスの SSL 出力アサートのウェイト (次アクセス遅延) を設定可能。

設定範囲: 1~8 RSPCK 設定単位: 1 RSPCK

SSL 極性変更機能。

- マスタ転送時の制御方式

最大 8 コマンドで構成された転送をシーケンシャルにループ実行可能。

各コマンドに設定可能な項目は以下のとおり。

SSL 信号値、ビットレート、RSPCK 極性/位相、転送データ長、LSB/MSB ファースト、バースト、RSPCK 遅延、SSL ネゲート遅延、次アクセス遅延。

送信バッファへのライトで転送を起動可能。

CPU による SPTEF ビットクリアで転送を起動可能。

SSL ネゲート時の MOSI 信号値を設定可能。

- 割り込み要因

マスカブルな割り込み要因あり。

RSPI 受信割り込み (受信バッファフル)。

RSPI 送信割り込み (送信バッファエンプティ)。

RSPI エラー割り込み (モードフォルト、オーバラン、パリティエラー)。

RSPI アイドル割り込み (RSPI アイドル)。

- その他

ループバックモードあり。

CMOS/オープンドレイン出力切り替え機能あり。

RSPI ディスエーブル (初期化) 機能あり。

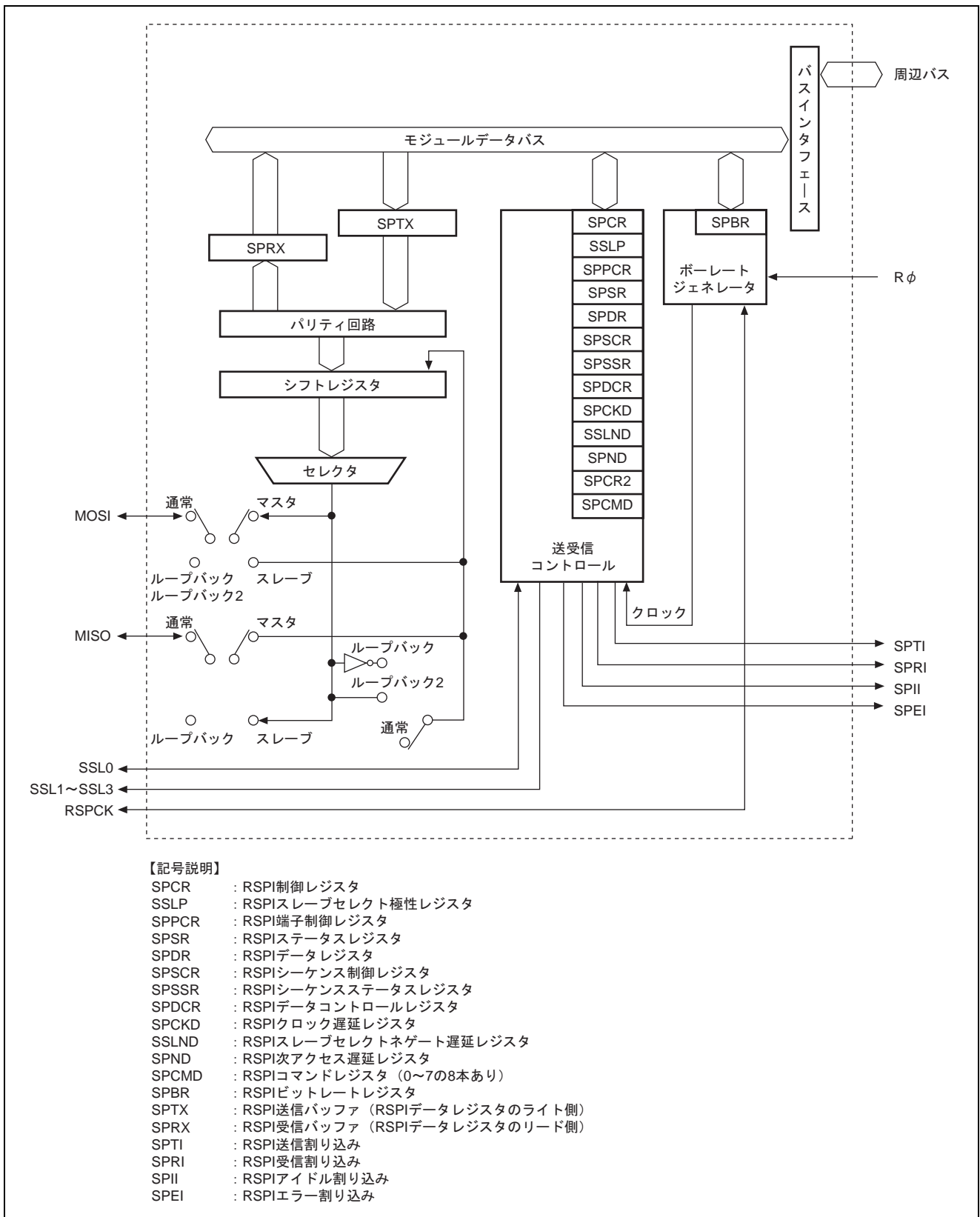


図 15.1 RSPI のブロック図 (1 チャンネル分)

15.2 入出力端子

RSPI は、チャンネルごとに表 15.1 に示すシリアル端子を持っています。SSL0 端子の入出力方向は、シングルマスタ設定の場合には出力、マルチマスタ設定とスレーブ設定の場合には入力に、RSPI が自動的に切り替えます。RSPCK、MOSI、MISO の入出力方向は、マスタ/スレーブ設定と SSL0 入力レベルに応じて、RSPI が自動的に切り替えます（「15.4.2 RSPI 端子の制御」を参照）。

表 15.1 端子構成

チャンネル	名称	端子名	入出力	機能
A	RSPI クロック端子	RSPCKA	入出力	RSPI のクロック入出力
	マスタ送出データ端子	MOSIA	入出力	RSPI のマスタ送出データ
	スレーブ送出データ端子	MISOA	入出力	RSPI のスレーブ送出データ
	スレーブセレクト 0 端子	SSLA0	入出力	RSPI のスレーブセレクト
	スレーブセレクト 1 端子	SSLA1	出力	RSPI のスレーブセレクト
	スレーブセレクト 2 端子	SSLA2	出力	RSPI のスレーブセレクト
	スレーブセレクト 3 端子	SSLA3	出力	RSPI のスレーブセレクト
B	RSPI クロック端子	RSPCKB	入出力	RSPI のクロック入出力
	マスタ送出データ端子	MOSIB	入出力	RSPI のマスタ送出データ
	スレーブ送出データ端子	MISOB	入出力	RSPI のスレーブ送出データ
	スレーブセレクト 0 端子	SSLB0	入出力	RSPI のスレーブセレクト
	スレーブセレクト 1 端子	SSLB1	出力	RSPI のスレーブセレクト
	スレーブセレクト 2 端子	SSLB2	出力	RSPI のスレーブセレクト
	スレーブセレクト 3 端子	SSLB3	出力	RSPI のスレーブセレクト
C	RSPI クロック端子	RSPCKC	入出力	RSPI のクロック入出力
	マスタ送出データ端子	MOSIC	入出力	RSPI のマスタ送出データ
	スレーブ送出データ端子	MISOC	入出力	RSPI のスレーブ送出データ
	スレーブセレクト 0 端子	SSLC0	入出力	RSPI のスレーブセレクト
	スレーブセレクト 1 端子	SSLC1	出力	RSPI のスレーブセレクト
	スレーブセレクト 2 端子	SSLC2	出力	RSPI のスレーブセレクト
	スレーブセレクト 3 端子	SSLC3	出力	RSPI のスレーブセレクト
D	RSPI クロック端子	RSPCKD	入出力	RSPI のクロック入出力
	マスタ送出データ端子	MOSID	入出力	RSPI のマスタ送出データ
	スレーブ送出データ端子	MISOD	入出力	RSPI のスレーブ送出データ
	スレーブセレクト 0 端子	SSLD0	入出力	RSPI のスレーブセレクト
	スレーブセレクト 1 端子	SSLD1	出力	RSPI のスレーブセレクト
	スレーブセレクト 2 端子	SSLD2	出力	RSPI のスレーブセレクト
	スレーブセレクト 3 端子	SSLD3	出力	RSPI のスレーブセレクト

【注】 本文中ではチャンネルを省略し、RSPICK、MOSI、MISO、SSL0～SSL3 と略称します。

15.3 レジスタの説明

RSPI には、表 15.2 に示す内部レジスタがあります。これらのレジスタにより、マスタ/スレーブモードの指定、転送フォーマットの指定、および送信部/受信部の制御を行うことができます。

表 15.2 レジスタ構成

チャンネル	名称	略称* ¹	R/W	初期値	アドレス	アクセス サイズ
A	RSPI 制御レジスタ A	SPCRA	R/W	H'00	H'FFD800	8、16
	RSPI スレーブセレクト極性 レジスタ A	SSLPA	R/W	H'00	H'FFD801	8、16
	RSPI 端子制御レジスタ A	SPPCRA	R/W	H'00	H'FFD802	8、16
	RSPI ステータスレジスタ A	SPSRA	R/(W)* ²	H'20	H'FFD803	8、16
	RSPI データレジスタ A	SPDRA	R/W	H'00000000	H'FFD804	16、32
	RSPI シーケンス制御レジスタ A	SPSCRA	R/W	H'00	H'FFD808	8、16
	RSPI シーケンスステータスレジスタ A	SPSSRA	R	H'00	H'FFD809	8、16
	RSPI ビットレートレジスタ A	SPBRA	R/W	H'FF	H'FFD80A	8、16
	RSPI データコントロールレジスタ A	SPDCRA	R/W	H'00	H'FFD80B	8、16
	RSPI クロック遅延レジスタ A	SPCKDA	R/W	H'00	H'FFD80C	8、16
	RSPI スレーブセレクトネゲート遅延 レジスタ A	SSLNDA	R/W	H'00	H'FFD80D	8、16
	RSPI 次アクセス遅延レジスタ A	SPNDA	R/W	H'00	H'FFD80E	8、16
	RSPI 制御レジスタ 2 A	SPCR2A	R/W	H'00	H'FFD80F	8、16
	RSPI コマンドレジスタ A0	SPCMDA0	R/W	H'070D	H'FFD810	16
	RSPI コマンドレジスタ A1	SPCMDA1	R/W	H'070D	H'FFD812	16
	RSPI コマンドレジスタ A2	SPCMDA2	R/W	H'070D	H'FFD814	16
	RSPI コマンドレジスタ A3	SPCMDA3	R/W	H'070D	H'FFD816	16
	RSPI コマンドレジスタ A4	SPCMDA4	R/W	H'070D	H'FFD818	16
	RSPI コマンドレジスタ A5	SPCMDA5	R/W	H'070D	H'FFD81A	16
	RSPI コマンドレジスタ A6	SPCMDA6	R/W	H'070D	H'FFD81C	16
RSPI コマンドレジスタ A7	SPCMDA7	R/W	H'070D	H'FFD81E	16	

チャンネル	名称	略称* ¹	R/W	初期値	アドレス	アクセス サイズ
B	RSPI 制御レジスタ B	SPCRB	R/W	H'00	H'FFD820	8、16
	RSPI スレーブセレクト極性レジスタ B	SSLPB	R/W	H'00	H'FFD821	8、16
	RSPI 端子制御レジスタ B	SPPCRB	R/W	H'00	H'FFD822	8、16
	RSPI ステータスレジスタ B	SPSRB	R/(W)* ²	H'20	H'FFD823	8、16
	RSPI データレジスタ B	SPDRB	R/W	H'00000000	H'FFD824	16、32
	RSPI シーケンス制御レジスタ B	SPSCRB	R/W	H'00	H'FFD828	8、16
	RSPI シーケンスステータスレジスタ B	SPSSRB	R	H'00	H'FFD829	8、16
	RSPI ビットレートレジスタ B	SPBRB	R/W	H'FF	H'FFD82A	8、16
	RSPI データコントロールレジスタ B	SPDCRB	R/W	H'00	H'FFD82B	8、16
	RSPI クロック遅延レジスタ B	SPCKDB	R/W	H'00	H'FFD82C	8、16
	RSPI スレーブセレクトネゲート遅延 レジスタ B	SSLNDB	R/W	H'00	H'FFD82D	8、16
	RSPI 次アクセス遅延レジスタ B	SPNDB	R/W	H'00	H'FFD82E	8、16
	RSPI 制御レジスタ 2 B	SPCR2B	R/W	H'00	H'FFD82F	8、16
	RSPI コマンドレジスタ B0	SPCMDB0	R/W	H'070D	H'FFD830	16
	RSPI コマンドレジスタ B1	SPCMDB1	R/W	H'070D	H'FFD832	16
	RSPI コマンドレジスタ B2	SPCMDB2	R/W	H'070D	H'FFD834	16
	RSPI コマンドレジスタ B3	SPCMDB3	R/W	H'070D	H'FFD836	16
	RSPI コマンドレジスタ B4	SPCMDB4	R/W	H'070D	H'FFD838	16
	RSPI コマンドレジスタ B5	SPCMDB5	R/W	H'070D	H'FFD83A	16
	RSPI コマンドレジスタ B6	SPCMDB6	R/W	H'070D	H'FFD83C	16
RSPI コマンドレジスタ B7	SPCMDB7	R/W	H'070D	H'FFD83E	16	

チャンネル	名称	略称* ¹	R/W	初期値	アドレス	アクセス サイズ
C	RSPI 制御レジスタ C	SPCRC	R/W	H'00	H'FFD840	8、16
	RSPI スレーブセレクト極性レジスタ C	SSLPC	R/W	H'00	H'FFD841	8、16
	RSPI 端子制御レジスタ C	SPPCRC	R/W	H'00	H'FFD842	8、16
	RSPI ステータスレジスタ C	SPSRC	R/(W)* ²	H'20	H'FFD843	8、16
	RSPI データレジスタ C	SPDRC	R/W	H'00000000	H'FFD844	16、32
	RSPI シーケンス制御レジスタ C	SPSCRC	R/W	H'00	H'FFD848	8、16
	RSPI シーケンスステータスレジスタ C	SPSSRC	R	H'00	H'FFD849	8、16
	RSPI ビットレートレジスタ C	SPBRC	R/W	H'FF	H'FFD84A	8、16
	RSPI データコントロールレジスタ C	SPDCRC	R/W	H'00	H'FFD84B	8、16
	RSPI クロック遅延レジスタ C	SPCKDC	R/W	H'00	H'FFD84C	8、16
	RSPI スレーブセレクトネゲート遅延 レジスタ C	SSLNDC	R/W	H'00	H'FFD84D	8、16
	RSPI 次アクセス遅延レジスタ C	SPNDC	R/W	H'00	H'FFD84E	8、16
	RSPI 制御レジスタ 2 C	SPCR2C	R/W	H'00	H'FFD84F	8、16
	RSPI コマンドレジスタ C0	SPCMDC0	R/W	H'070D	H'FFD850	16
	RSPI コマンドレジスタ C1	SPCMDC1	R/W	H'070D	H'FFD852	16
	RSPI コマンドレジスタ C2	SPCMDC2	R/W	H'070D	H'FFD854	16
	RSPI コマンドレジスタ C3	SPCMDC3	R/W	H'070D	H'FFD856	16
	RSPI コマンドレジスタ C4	SPCMDC4	R/W	H'070D	H'FFD858	16
	RSPI コマンドレジスタ C5	SPCMDC5	R/W	H'070D	H'FFD85A	16
	RSPI コマンドレジスタ C6	SPCMDC6	R/W	H'070D	H'FFD85C	16
RSPI コマンドレジスタ C7	SPCMDC7	R/W	H'070D	H'FFD85E	16	

チャンネル	名称	略称* ¹	R/W	初期値	アドレス	アクセス サイズ
D	RSPI 制御レジスタ D	SPCRD	R/W	H'00	H'FFD860	8、16
	RSPI スレーブセレクト極性レジスタ D	SSLPD	R/W	H'00	H'FFD861	8、16
	RSPI 端子制御レジスタ D	SPPCRD	R/W	H'00	H'FFD862	8、16
	RSPI ステータスレジスタ D	SPSRD	R/(W)* ²	H'20	H'FFD863	8、16
	RSPI データレジスタ D	SPDRD	R/W	H'00000000	H'FFD864	16、32
	RSPI シーケンス制御レジスタ D	SPSCRD	R/W	H'00	H'FFD868	8、16
	RSPI シーケンスステータスレジスタ D	SPSSRD	R	H'00	H'FFD869	8、16
	RSPI ビットレートレジスタ D	SPBRD	R/W	H'FF	H'FFD86A	8、16
	RSPI データコントロールレジスタ D	SPDCRD	R/W	H'00	H'FFD86B	8、16
	RSPI クロック遅延レジスタ D	SPCKDD	R/W	H'00	H'FFD86C	8、16
	RSPI スレーブセレクトネゲート遅延 レジスタ D	SSLNDD	R/W	H'00	H'FFD86D	8、16
	RSPI 次アクセス遅延レジスタ D	SPNDD	R/W	H'00	H'FFD86E	8、16
	RSPI 制御レジスタ 2 D	SPCR2D	R/W	H'00	H'FFD86F	8、16
	RSPI コマンドレジスタ D0	SPCMDD0	R/W	H'070D	H'FFD870	16
	RSPI コマンドレジスタ D1	SPCMDD1	R/W	H'070D	H'FFD872	16
	RSPI コマンドレジスタ D2	SPCMDD2	R/W	H'070D	H'FFD874	16
	RSPI コマンドレジスタ D3	SPCMDD3	R/W	H'070D	H'FFD876	16
	RSPI コマンドレジスタ D4	SPCMDD4	R/W	H'070D	H'FFD878	16
	RSPI コマンドレジスタ D5	SPCMDD5	R/W	H'070D	H'FFD87A	16
	RSPI コマンドレジスタ D6	SPCMDD6	R/W	H'070D	H'FFD87C	16
RSPI コマンドレジスタ D7	SPCMDD7	R/W	H'070D	H'FFD87E	16	

【注】 *1 本文中のチャンネルを省略してレジスタ名を表記しています。

*2 フラグをクリアするために0のみ書き込むことができます。

15.3.1 RSPI 制御レジスタ (SPCR)

SPCR は、RSPI の動作モードを設定するためのレジスタです。SPCR は、常に CPU による読み出し／書き込みが可能です。SPE ビットが 1 で RSPI 機能がイネーブルである状態において、MSTR ビット、MODFEN ビット、TXMD ビットの設定値を変更した場合には、以降の動作は保証されません。

ビット	7	6	5	4	3	2	1	0
ビット名	SPRIE	SPE	SPTIE	SPEIE	MSTR	MODFEN	TXMD	SPMS
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

ビット	ビット名	初期値	R/W	説明
7	SPRIE	0	R/W	<p>RSPI 受信割り込みイネーブル</p> <p>RSPI がシリアル転送完了後の受信バッファ書き込みを検出し、RSPI ステータスレジスタ (SPSR) の SPRF ビットを 1 にした場合の RSPI 受信割り込み要求の発生を許可／禁止するためのビットです。</p> <p>0 : RSPI 受信割り込み要求の発生を禁止する 1 : RSPI 受信割り込み要求の発生を許可する</p>
6	SPE	0	R/W	<p>RSPI 機能イネーブル</p> <p>本ビットを 1 にすることにより、RSPI 機能が有効になります。RSPI ステータスレジスタ (SPSR) の MODF ビットが 1 の場合には、SPE ビットを 1 に設定することはできません (「15.4.8 エラー検出」を参照)。SPE ビットを 0 にすると、RSPI 機能は無効化されて、モジュール機能の一部が初期化されます (「15.4.9 RSPI の初期化」を参照)。</p> <p>0 : RSPI 機能を無効化する 1 : RSPI 機能を有効化する</p>
5	SPTIE	0	R/W	<p>RSPI 送信割り込みイネーブル</p> <p>RSPI が送信バッファエンプティを検出し、RSPI ステータスレジスタ (SPSR) の SPTEF ビットを 1 にした場合の RSPI 送信割り込み要求の発生を許可／禁止するためのビットです。</p> <p>RSPI ディスエーブル (SPE ビットが 0) の状態では、SPTEF ビットが 1 になります。このため、RSPI ディスエーブル状態で SPTIE を 1 に設定すると、RSPI 送信割り込み要求が発生することに注意してください。</p> <p>0 : RSPI 送信割り込み要求の発生を禁止する 1 : RSPI 送信割り込み要求の発生を許可する</p>

ビット	ビット名	初期値	R/W	説明
4	SPEIE	0	R/W	<p>RSPI エラー割り込みイネーブル</p> <p>RSPI がモードフォルトエラーを検出して RSPI ステータスレジスタ (SPSR) の MODF ビットを 1 にした場合、または RSPI がオーバランエラーを検出して SPSR の OVRF ビットを 1 にした場合の RSPI エラー割り込み要求の発生を許可/禁止するためのビットです (「15.4.8 エラー検出」を参照)。</p> <p>0 : RSPI エラー割り込み要求の発生を禁止する 1 : RSPI エラー割り込み要求の発生を許可する</p>
3	MSTR	0	R/W	<p>RSPI マスタ/スレーブモード選択</p> <p>RSPI のマスタ/スレーブモードを選択するためのビットです。また、RSPI は MSTR ビットの設定に従って、RSPCK、MOSI、MISO、SSL1~SSL3 端子の方向を決定します。</p> <p>0 : スレーブモード 1 : マスタモード</p>
2	MODFEN	0	R/W	<p>モードフォルトエラー検出イネーブル</p> <p>モードフォルトエラーの検出を許可/禁止するためのビットです (「15.4.8 エラー検出」を参照)。また、RSPI は MODFEN ビットと MSTR ビットとの組み合わせに従って、SSL0 端子の入出力方向を決定します (「15.4.2 RSPI 端子の制御」を参照)。</p> <p>0 : モードフォルトエラー検出を禁止する 1 : モードフォルトエラー検出を許可する</p>
1	TXMD	0	R/W	<p>通信動作モード選択ビット</p> <p>全二重同期式のシリアル通信、送信のみの動作を選択するためのビットです。</p> <p>TXMD を 1 にして、通信を行う場合は、送信動作のみを行い、受信動作を行いません (「15.4.6 通信動作モード」を参照)。</p> <p>また、TXMD を 1 に設定した時、受信バッファフルの割り込み要求を使用することはできません。</p> <p>0 : 全二重同期式のシリアル通信 1 : 送信のみ動作</p>

ビット	ビット名	初期値	R/W	説明
0	SPMS	0	R/W	<p>RSPI モード選択</p> <p>SPI 動作 (4 線式) / クロック同期式動作 (3 線式) を選択するためのビットです。</p> <p>クロック同期式動作を行う場合は SSL 端子を使用せず、RSPCK 端子、MOSI 端子、MISO 端子の 3 端子を用いて通信を行います。また、マスタモード (MSTR=1) でクロック同期式動作を行う場合は、RSPI コマンドレジスタ (SPCMD) の CPHA を 0、1 どちらにも設定可能です。スレーブモード時 (MSTR=0) でクロック同期式動作を行う場合は CPHA を 1 に設定してください。スレーブモード (MSTR=0) でクロック同期式動作を行う場合に CPHA を 0 に設定した場合の動作は保証されません。</p> <p>0 : SPI 動作 (4 線式)</p> <p>1 : クロック同期式動作 (3 線式)</p>

15.3.2 RSPI スレーブセレクト極性レジスタ (SSLP)

SSLP は、RSPI の SSL0~3 信号の極性を設定するためのレジスタです。SSLP は、常に CPU による読み出し / 書き込みが可能です。RSPI 制御レジスタ (SPCR) の SPE ビットが 1 で RSPI 機能がイネーブルである状態において、CPU が SSLP を書き換えた場合には、以降の動作は保証されません。

ビット	7	6	5	4	3	2	1	0
ビット名	—	—	—	—	SSL3P	SSL2P	SSL1P	SSL0P
初期値 :	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R/W	R/W	R/W	R/W

ビット	ビット名	初期値	R/W	説明
7~4	—	すべて 0	R	<p>リザーブビット</p> <p>書き込む値は常に 0 にしてください。1 を書き込んだ場合の動作は保証しません。</p>
3	SSL3P	0	R/W	<p>SSL 信号極性設定</p> <p>SSL 信号の極性を設定するためのビットです。SSLiP (i=3~0) の設定値が、SSLi 信号のアクティブ極性を示します。</p> <p>0 : SSLi 信号は 0 アクティブ</p> <p>1 : SSLi 信号は 1 アクティブ</p>
2	SSL2P	0	R/W	
1	SSL1P	0	R/W	
0	SSL0P	0	R/W	

15.3.3 RSPI 端子制御レジスタ (SPPCR)

SPPCR は、RSPI の端子モードを設定するために使用されるレジスタです。SPPCR は、常に CPU による読み出し/書き込みが可能です。SPCR の SPE ビットが 1 で RSPI 機能がイネーブルである状態において、CPU が SPPCR を書き換えた場合には、以降の動作は保証されません。

ビット	7	6	5	4	3	2	1	0
ビット名	—	—	MOIFE	MOIFV	—	SPOM	SPLP2	SPLP
初期値 :	0	0	0	0	0	0	0	0
R/W :	R	R	R/W	R/W	R	R/W	R/W	R/W

ビット	ビット名	初期値	R/W	説明
7、6	—	すべて 0	R	リザーブビット 書き込む値は常に 0 にしてください。1 を書き込んだ場合の動作は保証しません。
5	MOIFE	0	R/W	MOSI アイドル値固定イネーブル マスタモードの RSPI が、SSL ネゲート期間（バースト転送における SSL 保持期間を含む）に MOSI 出力値を固定するために使用するビットです。 MOIFE が 0 の場合には、RSPI は SSL ネゲート期間中に前回のシリアル転送の最終データを MOSI に出力します。MOIFE が 1 の場合には、RSPI は MOIFV ビットに設定された固定値を MOSI に出力します。 0 : MOSI 出力値は前回転送の最終データ 1 : MOSI 出力値は MOIFV ビットの設定値
4	MOIFV	0	R/W	MOSI アイドル固定値 マスタモードで MOIFE ビットが 1 の場合には、RSPI は MOIFV ビットの設定に従って、SSL ネゲート期間（バースト転送における SSL 保持期間を含む）の MOSI 信号値を決定します。 0 : MOSI アイドル固定値は 0 1 : MOSI アイドル固定値は 1
3	—	0	R	リザーブビット 書き込む値は常に 0 にしてください。1 を書き込んだ場合の動作は保証しません。
2	SPOM	0	R/W	RSPI 出力端子モード RSPI の出力端子を CMOS 出力/オープンドレイン出力に設定するためのビットです。 0 : CMOS 出力 1 : オープンドレイン出力

ビット	ビット名	初期値	R/W	説明
1	SPLP2	0	R/W	RSPI ループバック 2 SPLP2 ビットを 1 にすると、RSPI は MISO 端子とシフトレジスタ間、MOSI 端子とシフトレジスタ間の経路を遮断し、シフトレジスタの入力経路と出力経路を接続します (ループバックモード)。 0: 通常モード 1: ループバックモード (送信データ=受信データ)
0	SPLP	0	R/W	RSPI ループバック SPLP ビットを 1 にすると、RSPI は MISO 端子とシフトレジスタ間、MOSI 端子とシフトレジスタ間の経路を遮断し、シフトレジスタの入力経路と出力経路 (反転) を接続します。 0: 通常モード 1: ループバックモード (送信データの反転=受信データ)

15.3.4 RSPI ステータスレジスタ (SPSR)

SPSR は、RSPI の動作状態を示すフラグを格納したレジスタです。SPSR は、常に CPU による読み出しが可能です。CPU から SPSR への書き込みは、一定条件下においてのみ有効です。

ビット	7	6	5	4	3	2	1	0
ビット名	SPRF	—	SPTEF	—	PERF	MODF	IDLNF	OVRF
初期値:	0	0	1	0	0	0	0	0
R/W:	R/(W)*	R	R/(W)*	R	R/(W)*	R/(W)*	R	R/(W)*

【注】 * フラグをクリアするため、1を読み出した後に0を書き込むことのみ可能です。

ビット	ビット名	初期値	R/W	説明
7	SPRF	0	R/(W)*	<p>RSPI 受信バッファフルフラグ</p> <p>RSPI データレジスタ (SPDR) の受信バッファの状態を示すビットです。RSPI 制御レジスタ (SPCR) の通信動作モード選択ビット (TXMD) が 0、SPRF ビットが 0 の状態でシリアル転送が終了すると、RSPI はシフトレジスタから SPDR に受信データを転送して、このビットを 1 にします。また、TXMD が 0 の時、RSPI は全二重同期式のシリアル通信を行いますので、送信データの最後尾ビットの送信時でもあります。SPRF ビットを 0 にクリアする条件は以下のとおりです。</p> <ul style="list-style-type: none"> • SPRF が 1 にされた状態の SPSR を CPU が読み出した後、CPU が SPRF に 0 を書き込む • SPDR から受信データを読み出す。 • システムリセット <p>SPRF ビットが 1 の状態でシリアル転送が終了した場合には、RSPI はシフトレジスタから SPDR に受信データを転送しません。SPSR の OVRF ビットが 1 の状態では、SPRF ビットを 0 から 1 に変化させることができません (「15.4.8 エラー検出」を参照)。</p> <p>0 : SPDR に有効な受信データなし 1 : SPDR に有効な受信データあり</p>
6	—	0	R	<p>リザーブビット</p> <p>書き込む値は常に 0 にしてください。1 を書き込んだ場合の動作は保証しません。</p>
5	SPTEF	1	R/(W)*	<p>RSPI 送信バッファエンプティフラグ</p> <p>RSPI データレジスタ (SPDR) の送信バッファの状態を示すビットです。RSPI の初期化後、または送信バッファからシフトレジスタに送信データが転送された場合に、RSPI が SPTEF ビットを 1 にします。SPTEF ビットを 0 にクリアする条件は以下のとおりです。SPTEF クリア状態で、かつシフトレジスタが空の場合、送信バッファからシフトレジスタへデータがコピーされます。</p> <ul style="list-style-type: none"> • SPTEF が 1 にされた状態の SPSR を CPU が読み出した後、CPU が SPTEF に 0 を書き込む • SPDR に送信データを書き込む <p>SPDR に対する書き込みは、SPTEF ビットが 1 の場合にのみ有効です。SPTEF が 0 の状態で、SPDR の送信バッファを書き込んだ場合には、送信バッファのデータは更新されません。</p> <p>0 : 送信バッファにデータあり 1 : 送信バッファにデータなし</p>
4	—	0	R	<p>リザーブビット</p> <p>書き込む値は常に 0 にしてください。1 を書き込んだ場合の動作は保証しません。</p>

ビット	ビット名	初期値	R/W	説明
3	PERF	0	R/(W) *	<p>パリティエラーフラグ</p> <p>パリティエラーの発生状況を示すビットです。SPCR の TXMD が 0、SPCR2 の SPPE が 1 の状態でシリアル転送が終了した場合に、RSPI はパリティエラーを検出します。パリティエラーが発生した時、PERF ビットを 1 にします。PERF ビットのクリア条件は、以下のとおりです。</p> <ul style="list-style-type: none"> • PERF が 1 にされた状態の SPSR を CPU が読み出した後、CPU が PERF に 0 を書き込む • システムリセット <p>0 : パリティエラーなし 1 : パリティエラー発生</p>
2	MODF	0	R/(W)*	<p>モードフォルトエラーフラグ</p> <p>モードフォルトエラーの発生状況を示すビットです。RSPI 制御レジスタ (SPCR) の MSTR ビットが 1、MODFEN ビットが 1 で RSPI がマルチマスタモードの場合に、SSL0 端子の入カレベルがアクティブレベルになると、RSPI がモードフォルトエラー検出し MODF ビットを 1 にします。また、MSTR ビットが 0 で RSPI がスレーブモードの場合に MODFEN ビットに 1 を設定すると、データ転送に必要な RSPCK サイクルが終了する前に SSL0 端子がネゲートされると、RSPI がモードフォルトエラーを検出します。なお、SSL0 信号のアクティブレベルは、RSPI スレーブセレクト極性レジスタ (SSLP) の SSL0P ビットによって決定されます。MODF ビットのクリア条件は以下のとおりです。</p> <ul style="list-style-type: none"> • MODF が 1 にされた状態の SPSR を CPU が読み出した後、CPU が MODF に 0 を書き込む • システムリセット <p>0 : モードフォルトエラーなし 1 : モードフォルトエラー発生</p>

ビット	ビット名	初期値	R/W	説明
1	IDLNF	0	R	<p>RSPI アイドルフラグ</p> <p>RSPI の転送状況を示すビットです。マスタモード時は SPSSR の SPCP ビットが 000 になりシーケンスの先頭である、かつ SPSR の SPTEF ビットが 1 で次転送のデータがセットされず、RSPI の転送が行われない状態で 0 にクリアされます。</p> <p>マスタ/スレーブモード時共に SPCR の SPE ビットが 0 で RSPI の機能無効時に 0 にクリアされます。</p> <p>マスタモード (シングル/マルチ) 時における IDLNF ビットのクリア条件は、以下の通りです。</p> <ul style="list-style-type: none"> • SPCR の SPE ビットが 0 (RSPI 初期化) <p>または、</p> <ul style="list-style-type: none"> • SPSR の SPTEF ビットが 1 (次転送データがセットされていない) • SPSSR の SPCP ビットが 000 (シーケンス制御がループ先頭コマンドポインタに位置) • RSPI 内部シーケンサがアイドル状態へ遷移 (次アクセス遅延までが動作完了された状態) <p>の条件が全て満たされた場合、IDLNF ビットはクリアされます。 上記条件が満たされない場合、IDLNF ビットはセットされます。</p> <p>スレーブモード時は IDLNF ビットのクリア条件は、以下の通りです。</p> <ul style="list-style-type: none"> • SPCR の SPE ビットが 0 (RSPI 初期化) SPE=1 にセットされると IDLNF ビットは 1 にセットされます。
0	OVRF	0	R/(W)*	<p>オーバランエラーフラグ</p> <p>オーバランエラーの発生状況を示すビットです。RSPI 制御レジスタ (SPCR) の通信動作モード選択ビット (TXMD) が 0、SPRF ビットが 1 にされた状態でシリアル転送が終了した場合に、RSPI はオーバランエラーを検出し、OVRF ビットを 1 にします。OVRF ビットのクリア条件は、以下のとおりです。</p> <ul style="list-style-type: none"> • OVRF が 1 にされた状態の SPSR を CPU が読み出した後、CPU が OVRF に 0 を書き込む • システムリセット <p>0 : オーバランエラーなし 1 : オーバランエラー発生</p>

【注】 * フラグをクリアするため、1 を読み出した後に 0 を書き込むことのみ可能です。

15.3.5 RSPI データレジスタ (SPDR)

SPDR は、RSPI 送受信のデータを格納するバッファです。

送信用バッファ (SPTX) と受信用バッファ (SPRX) は独立したバッファで、これらのバッファが SPDR にマッピングされています。

SPDR へのリード/ライトは、RSPI データコントロールレジスタ (SPDCR) の RSPI ロングワードアクセス/ワードアクセス設定ビット (SPLW) の設定によって、ワード/ロングワードで行ってください。SPLW が 0 のとき、SPDR は、64 ビットのバッファで最大 16 ビットの 4 フレームから構成され、SPLW が 1 のとき、SPDR は、128 ビットのバッファで最大 32 ビットの 4 フレームから構成されます。

SPDR の使用するフレーム長は RSPI データコントロールレジスタ (SPDCR) のフレーム数設定ビット (SPFC1 ~ SPFC0) によって決定され、使用するビット長は RSPI コマンドレジスタ (SPCMD) の RSPI データ長設定ビット (SPB3 ~ 0) によって決定されます。

SPDR への書き込みを行う場合には、RSPI ステータスレジスタ (SPSR) の SPTEF ビットが 1 にされていれば、RSPI が SPDR の送信バッファにデータを書き込みます。SPTEF ビットが 0 の状態では、RSPI は SPDR の送信バッファを更新しません。

SPDR からの読み出しを行う場合には、RSPI データコントロールレジスタ (SPDCR) の RSPI 受信/送信データ選択ビット (SPRDTD) が 0 であれば受信バッファを読み出し、1 であれば送信バッファを読み出します。

送信バッファを読み出す場合には、直前に書き込んだ値が読み出されます。また、RSPI ステータスレジスタ (SPSR) の SPTEF ビットが 0 の状態では、送信バッファを読み出しません。

通常の使用方法では、SPRDTD を 0 とし、SPSR の SPRF ビットの 1 状態 (受信バッファに未リードのデータが格納された状態) で、受信バッファ読み出しを実行します。SPSR の SPRF ビットまたは OVRF ビットが 1 の状態では、RSPI はシリアル転送終了時に SPDR の受信バッファを更新しません。

ビット	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ビット名	SPD31	SPD30	SPD29	SPD28	SPD27	SPD26	SPD25	SPD24	SPD23	SPD22	SPD21	SPD20	SPD19	SPD18	SPD17	SPD16
初期値:	0	0	0	0	0	0	0	0	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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ビット	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ビット名	SPD15	SPD14	SPD13	SPD12	SPD11	SPD10	SPD9	SPD8	SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
初期値:	0	0	0	0	0	0	0	0	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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15.3.6 RSPI シーケンス制御レジスタ (SPSCR)

SPSCR は、RSPI がマスタ動作する場合のシーケンス制御方式を設定するためのレジスタです。SPSCR は、常に CPU による読み出し/書き込みが可能です。RSPI 制御レジスタ (SPCR) の MSTR ビットと SPE ビットが 1 で、マスタモードの RSPI 機能がイネーブルである状態において、CPU が SPSCR を書き換えた場合には、以降の動作は保証されません。

ビット	7	6	5	4	3	2	1	0
ビット名	—	—	—	—	—	SPSLN2	SPSLN1	SPSLN0
初期値:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

ビット	ビット名	初期値	R/W	説明																								
7~3	—	すべて 0	R	リザーブビット 書き込む値は常に 0 にしてください。1 を書き込んだ場合の動作は保証しません。																								
2~0	SPSLN2~ SPSLN0	000	R/W	RSPI シーケンス長設定 マスタモードの RSPI がシーケンス動作する場合のシーケンス長を設定するためのビットです。マスタモードの RSPI は SPSLN2~SPSLN0 に設定されたシーケンス長に応じて、参照する RSPI コマンドレジスタ 0~7 (SPCMD0~7) と参照順を変更します。SPSLN2~SPSLN0 の設定値とシーケンス長、RSPI が参照する SPCMD0~7 の関係は以下のとおりです。なお、スレーブモードの RSPI では、常に SPCMD0 が参照されます。 <div style="text-align: center;"> シーケンス長 参照する SPCMD レジスタ (番号) </div> <table style="margin-left: auto; margin-right: auto;"> <tr><td>000 :</td><td>1</td><td>0→0→...</td></tr> <tr><td>001 :</td><td>2</td><td>0→1→0→...</td></tr> <tr><td>010 :</td><td>3</td><td>0→1→2→0→...</td></tr> <tr><td>011 :</td><td>4</td><td>0→1→2→3→0→...</td></tr> <tr><td>100 :</td><td>5</td><td>0→1→2→3→4→0→...</td></tr> <tr><td>101 :</td><td>6</td><td>0→1→2→3→4→5→0→...</td></tr> <tr><td>110 :</td><td>7</td><td>0→1→2→3→4→5→6→0→...</td></tr> <tr><td>111 :</td><td>8</td><td>0→1→2→3→4→5→6→7→0→...</td></tr> </table>	000 :	1	0→0→...	001 :	2	0→1→0→...	010 :	3	0→1→2→0→...	011 :	4	0→1→2→3→0→...	100 :	5	0→1→2→3→4→0→...	101 :	6	0→1→2→3→4→5→0→...	110 :	7	0→1→2→3→4→5→6→0→...	111 :	8	0→1→2→3→4→5→6→7→0→...
000 :	1	0→0→...																										
001 :	2	0→1→0→...																										
010 :	3	0→1→2→0→...																										
011 :	4	0→1→2→3→0→...																										
100 :	5	0→1→2→3→4→0→...																										
101 :	6	0→1→2→3→4→5→0→...																										
110 :	7	0→1→2→3→4→5→6→0→...																										
111 :	8	0→1→2→3→4→5→6→7→0→...																										

15.3.7 RSPI シーケンスステータスレジスタ (SPSSR)

SPSSR は、RSPI がマスタ動作する場合のシーケンス制御の状態を示すレジスタです。SPSSR は、常に CPU による読み出しが可能です。CPU から SPSSR への書き込みは無視されます。

ビット	7	6	5	4	3	2	1	0
ビット名	—	SPECM2	SPECM1	SPECM0	—	SPCP2	SPCP1	SPCP0
初期値 :	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R

ビット	ビット名	初期値	R/W	説明
7	—	0	R	リザーブビット 書き込む値は常に0にしてください。1を書き込んだ場合の動作は保証しません。
6~4	SPECM2~ SPECM0	すべて0	R	RSPI エラーコマンド RSPI のシーケンス制御で、エラー検出時にコマンドポインタ (SPCP2~SPCP0 ビット) で指されていた RSPI コマンドレジスタ 0~7 (SPCMD0~SPCMD7) を示すビットです。RSPI はエラー検出時にのみ SPECM2~SPECM0 を更新します。RSPI ステータスレジスタ (SPSR) の OVRF ビットと MODF ビットがともに0で、エラーが発生していない場合には、SPECM2~SPECM0 の値には意味がありません。SPECM2~SPECM0 の値と SPCMD7~SPCMD0 の対応は以下のとおりです。 なお、RSPI のエラー検出機能については、「15.4.8 エラー検出」を参照してください。また、RSPI のシーケンス制御については、「15.4.10 (1) マスタモード動作」を参照してください。 000 : SPCMD0 001 : SPCMD1 010 : SPCMD2 011 : SPCMD3 100 : SPCMD4 101 : SPCMD5 110 : SPCMD6 111 : SPCMD7
3	—	0	R	リザーブビット 書き込む値は常に0にしてください。1を書き込んだ場合の動作は保証しません。

ビット	ビット名	初期値	R/W	説明
2~0	SPCP2~ SPCP0	すべて 0	R	<p>RSPI コマンドポインタ</p> <p>RSPI のシーケンス制御で、現在ポインタで指されている RSPI コマンドレジスタ 7~0 (SPCMD7~0) を示すビットです。SPCP2~SPCP0 の値と SPCMD7~SPCMD0 の対応は以下のとおりです。</p> <p>なお、RSPI のシーケンス制御については、「15.4.10 (1) マスタモード動作」を参照してください。</p> <p>000 : SPCMD0 001 : SPCMD1 010 : SPCMD2 011 : SPCMD3 100 : SPCMD4 101 : SPCMD5 110 : SPCMD6 111 : SPCMD7</p>

15.3.8 RSPI ビットレートレジスタ (SPBR)

SPBR は、マスタモード時のビットレート設定に使用するレジスタです。SPBR は、常に CPU による読み出し/書き込みが可能です。RSPI 制御レジスタ (SPCR) の MSTR ビットと SPE ビットが 1 で、マスタモードの RSPI 機能がイネーブルである状態において、CPU が SPBR を書き換えた場合には、以降の動作は保証されません。

ビット	7	6	5	4	3	2	1	0
ビット名	SPR7	SPR6	SPR5	SPR4	SPR3	SPR2	SPR1	SPR0
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

RSPI をスレーブモードで使用する場合には、SPBR、BRDV の設定に関係なく、入力クロックのビットレートに依存します。

ビットレートは SPBR の設定値と RSPI コマンドレジスタ (SPCMD0~3) の BRDV1~0 ビットの設定値の組み合わせで決定されます。ビットレートの計算式は下記のとおりです。計算式中で n は SPBR の設定値 (0、1、2、……、255)、N は BRDV1~0 ビットの設定値 (0、1、2、3) です。

$$\text{ビットレート} = \frac{f(R\phi)}{2 \times (n+1) \times 2^N}$$

SPBR レジスタ、BRDV1~0 ビットの設定値とビットレートの関係の例を表 15.3 に示します。

表 15.3 SPBR レジスタ、BRDV1~0 ビットの設定値とビットレート

SPBR の設定値 (n)	BRDV1~0 の設定値 (N)	分周比	ビットレート	
			R ϕ = 32MHz	R ϕ = 40MHz
0	0	2	16.0 Mbps*	20.0 Mbps*
1	0	4	8.00 Mbps	10.0 Mbps
2	0	6	5.33 Mbps	6.67 Mbps
3	0	8	4.00 Mbps	5.00 Mbps
4	0	10	3.20 Mbps	4.00 Mbps
5	0	12	2.67 Mbps	3.33 Mbps
5	1	24	1.33 Mbps	1.67 Mbps
5	2	48	667 kbps	833 kbps
5	3	96	333 kbps	417 kbps
255	3	4096	7.81 kbps	9.78 kbps

【注】 * 本 LSI では設定できません。

15.3.9 RSPI データコントロールレジスタ (SPDCR)

SPDCR は、SPDR レジスタに格納できるフレーム数、SSL 端子出力制御、SPDR レジスタの読み出し、SPDR レジスタへのアクセス幅をロングワードアクセス/ワードアクセスに設定するためのレジスタです。

RSPI コマンドレジスタ (SPCMD) の RSPI データ長設定ビット (SPB3~0)、RSPI シーケンス制御レジスタ (SPSCR) のシーケンス長設定ビット (SPSLN2~0)、RSPI データコントロールレジスタ (SPDCR) のフレーム数設定ビット (SPFC1、0) の組み合わせから 1 回の送受信起動で最大 4 フレームを送受信できます。

SPDCR は、常に CPU による読み出し/書き込みが可能です。RSPI 制御レジスタ (SPCR) の SPE ビットが 1 で RSPI 機能がイネーブルである状態において、CPU が SPDCR を書き換えた場合には、以降の動作は保証されません。

ビット	7	6	5	4	3	2	1	0
ビット名	—	—	SPLW	SPRDTD	SLSEL1	SLSELO	SPFC1	SPFC0
初期値:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

ビット	ビット名	初期値	R/W	説明
7、6	—	すべて 0	R	リザーブビット 書き込む値は常に 0 にしてください。1 を書き込んだ場合の動作は保証しません。
5	SPLW	0	R/W	RSPI ロングワードアクセス/ワードアクセス設定 RSPI データレジスタ (SPDR) へのアクセス幅を設定します。SPLW が 0 のときはワードアクセス、SPLW が 1 のときはロングワードアクセスで SPDR にアクセスしてください。 また、SPLW が 0 のとき、RSPI コマンドレジスタ (SPCMD) の RSPI データ長設定ビット (SPB3~0) の設定は、8~16 ビットに設定してください。20、24、32 ビットに設定した場合の動作は保証されません。 0: SPDR レジスタへはワードアクセス 1: SPDR レジスタへはロングワードアクセス
4	SPRDTD	0	R/W	RSPI 受信/送信データ選択 RSPI データレジスタ (SPDR) の読み出す値を受信バッファとするか、送信バッファとするか選択します。 送信バッファを読み出した場合、SPDR へ直前に書き込んだ値が読み出されます。送信バッファの読み出しは、RSPI ステータスレジスタ (SPSR) の SPTEF ビットが 1 の状態であるときに行ってください。 0: SPDR は受信バッファを読み出す 1: SPDR は送信バッファを読み出す (ただし、SPTEF ビットが 1 のとき)

ビット	ビット名	初期値	R/W	説明																						
3	SLSEL1	0	R/W	SSL 端子出力選択 SSL 端子選択ビット(SLSEL1,0)は、マスタモード時に SSL 端子の出力制御を行います。SLSEL1,SLSEL0 の組み合わせにより、SSL 端子に割り当てられた IO を IO として用いるか、SSL 端子として用いるかを設定できます。以下に SLSEL1、SLSEL0 の組み合わせと端子状態について以下に示します。SLSEL1=1、SLSEL0=1 と設定した場合、以後の動作は保証されません。																						
2	SLSEL0	0	R/W																							
<table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th></th> <th>SLSEL1=0 SLSEL0=0</th> <th>SLSEL1=0 SLSEL0=1</th> <th>SLSEL1=1 SLSEL0=0</th> <th>SLSEL1=1 SLSEL0=1</th> </tr> </thead> <tbody> <tr> <td>SSL3</td> <td>出力</td> <td>IO</td> <td>IO</td> <td rowspan="4" style="text-align: center;">設定禁止</td> </tr> <tr> <td>SSL2</td> <td>出力</td> <td>IO</td> <td>IO</td> </tr> <tr> <td>SSL1</td> <td>出力</td> <td>IO</td> <td>出力</td> </tr> <tr> <td>SSL0</td> <td>出力</td> <td>出力</td> <td>出力</td> </tr> </tbody> </table>						SLSEL1=0 SLSEL0=0	SLSEL1=0 SLSEL0=1	SLSEL1=1 SLSEL0=0	SLSEL1=1 SLSEL0=1	SSL3	出力	IO	IO	設定禁止	SSL2	出力	IO	IO	SSL1	出力	IO	出力	SSL0	出力	出力	出力
	SLSEL1=0 SLSEL0=0	SLSEL1=0 SLSEL0=1	SLSEL1=1 SLSEL0=0	SLSEL1=1 SLSEL0=1																						
SSL3	出力	IO	IO	設定禁止																						
SSL2	出力	IO	IO																							
SSL1	出力	IO	出力																							
SSL0	出力	出力	出力																							
1	SPFC1	0	R/W	フレーム数設定 SPDR レジスタに格納できるフレーム数を設定するビットです。RSPI コマンドレジスタ (SPCMD) の RSPI データ長設定ビット (SPB3~0)、RSPI シーケンス制御レジスタ (SPSCR) の RSPI シーケンス長設定ビット (SPSLN2~0)、RSPI データコントロールレジスタ (SPDCR) のフレーム数設定ビット (SPFC1、0) の設定により 1 回の送受信起動で最大 4 フレームを送受信できます。また、SPFC1、0 は、RSPI ステータスレジスタ (SPSR) の RSPI 受信バッファフルフラグ (SPRF) をセットする受信データ数の設定を行います。表 15.4 および図 15.2 に、SPDR レジスタに格納できるフレームの構成と送受信設定の組み合わせ例を示します。組み合わせ例に示した以外の設定を行った場合、以後の動作は保証されません。																						
0	SPFC0	0	R/W																							

表 15.4 各ビットによるフレームの設定

設定	SPB3~0	SPSLN2~0	SPFC1,0	転送する フレーム数	SPRF を 1 にセット、 SPTEF を 0 にクリアする フレーム数
1-1	M	000	00	1	1
1-2	M	000	01	2	2
1-3	M	000	10	3	3
1-4	M	000	11	4	4
2-1	M、N	001	01	2	2
2-2	M、N	001	11	4	4
3	M、N、O	010	10	3	3
4	M、N、O、P	011	11	4	4
5	M、N、O、P、Q	100	00	5	1
6	M、N、O、P、Q、R	101	00	6	1
7	M、N、O、P、Q、R、S	110	00	7	1
8	M、N、O、P、Q、R、S、T	111	00	8	1

【記号説明】 M、N、O、P、Q、R、S、T : SPB3~0 で設定できるデータ長

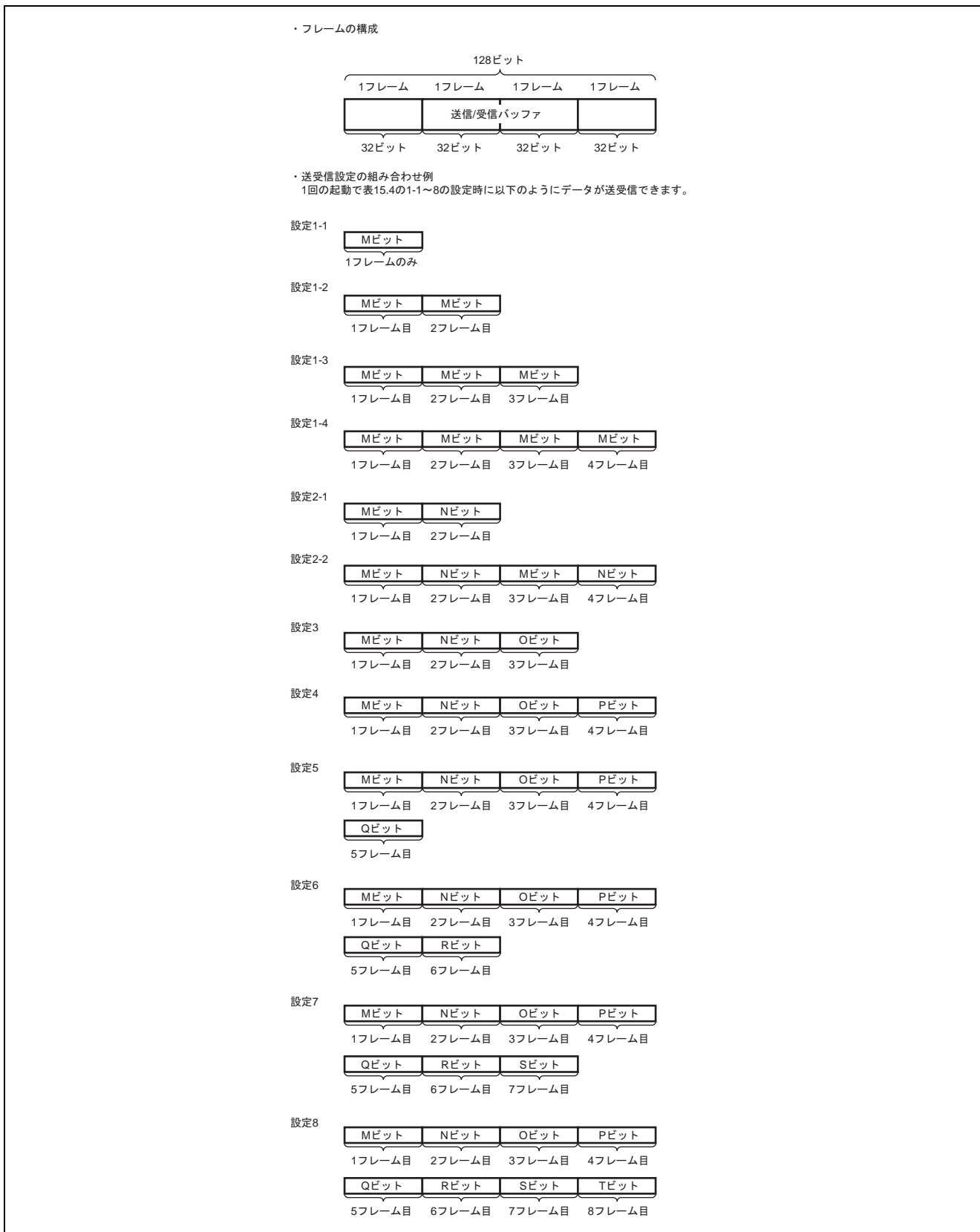


図 15.2 フレームの構成と送受信設定の組み合わせ例

15.3.10 RSPI クロック遅延レジスタ (SPCKD)

SPCKD は、RSPI コマンドレジスタ (SPCMD) の SCKDEN ビットが 1 の状態における、SSL 信号アサート開始から RSPCK 発振までの期間 (RSPCK 遅延) を設定するためのレジスタです。SPCKD は、常に CPU による読み出し/書き込みが可能です。RSPI 制御レジスタ (SPCR) の MSTR ビットと SPE ビットが 1 で、マスターモードの RSPI がイネーブルである状態において、CPU が SPCKD を書き換えた場合には、以降の動作は保証されません。

RSPI をスレーブモードで使用する場合には、SCKDL2~SCKDL0 に B'000 を設定してください。

ビット	7	6	5	4	3	2	1	0
ビット名	—	—	—	—	—	SCKDL2	SCKDL1	SCKDL0
初期値:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

ビット	ビット名	初期値	R/W	説明
7~3	—	すべて 0	R	リザーブビット 書き込む値は常に 0 にしてください。1 を書き込んだ場合の動作は保証しません。
2	SCKDL2	0	R/W	RSPCK 遅延設定 SPCMD の SCKDEN ビットが 1 の場合の RSPCK 遅延値を設定するためのビットです。 SCKDL2~SCKDL0 の設定値と RSPCK 遅延値の関係は以下のとおりです。 000 : 1RSPCK 001 : 2RSPCK 010 : 3RSPCK 011 : 4RSPCK 100 : 5RSPCK 101 : 6RSPCK 110 : 7RSPCK 111 : 8RSPCK
1	SCKDL1	0	R/W	
0	SCKDL0	0	R/W	

15.3.11 RSPI スレーブセレクトネゲート遅延レジスタ (SSLND)

SSLND は、マスタモードの RSPI がシリアル転送の最終 RSPCK エッジを送出してから SSL 信号をネゲートするまでの期間 (SSL ネゲート遅延) を設定するためのレジスタです。SSLND は、常に CPU による読み出し/書き込みが可能です。RSPI 制御レジスタ (SPCR) の MSTR ビットと SPE ビットが 1 で、マスタモードの RSPI がイネーブルである状態において、CPU が SSLND を書き換えた場合には、以降の動作は保証されません。

RSPI をスレーブモードで使用する場合には、SLNDL2~SLNDL0 に B'000 を設定してください。

ビット	7	6	5	4	3	2	1	0
ビット名	—	—	—	—	—	SLNDL2	SLNDL1	SLNDL0
初期値 :	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R/W	R/W	R/W

ビット	ビット名	初期値	R/W	説明
7~3	—	すべて 0	R	リザーブビット 書き込む値は常に 0 にしてください。1 を書き込んだ場合の動作は保証しません。
2	SLNDL2	0	R/W	SSL ネゲート遅延設定
1	SLNDL1	0	R/W	マスタモードの RSPI の SSL ネゲート遅延値を設定するためのビットです。
0	SLNDL0	0	R/W	SLNDL2~SLNDL0 の設定値と SS ネゲート遅延値の関係は以下のとおりです。 000 : 1RSPCK 001 : 2RSPCK 010 : 3RSPCK 011 : 4RSPCK 100 : 5RSPCK 101 : 6RSPCK 110 : 7RSPCK 111 : 8RSPCK

15.3.12 RSPI 次アクセス遅延レジスタ (SPND)

SPND は、RSPI コマンドレジスタ (SPCMD) の SPNDEN ビットが 1 の状態における、シリアル転送終了後の SSL 信号の非アクティブ期間 (次アクセス遅延) を設定するためのレジスタです。SPND は、常に CPU による読み出し/書き込みが可能です。RSPI 制御レジスタ (SPCR) の MSTR ビットと SPE ビットが 1 で、マスターモードの RSPI がイネーブルである状態において、CPU が SPND を書き換えた場合には、以降の動作は保証されません。

RSPI をスレーブモードで使用する場合には、SPNDL2~SPNDL0 に B'000 を設定してください。

ビット	7	6	5	4	3	2	1	0
ビット名	—	—	—	—	—	SPNDL2	SPNDL1	SPNDL0
初期値:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

ビット	ビット名	初期値	R/W	説明
7~3	—	すべて 0	R	リザーブビット 書き込む値は常に 0 にしてください。1 を書き込んだ場合の動作は保証しません。
2	SPNDL2	0	R/W	RSPI 次アクセス遅延設定 SPCMD の SPNDEN ビットが 1 の場合の次アクセス遅延を設定するためのビットです。 SPNDL2~SPNDL0 の設定値と次アクセス遅延値の関係は以下のとおりです。 000 : 1RSPCK+2Rφ 001 : 2RSPCK+2Rφ 010 : 3RSPCK+2Rφ 011 : 4RSPCK+2Rφ 100 : 5RSPCK+2Rφ 101 : 6RSPCK+2Rφ 110 : 7RSPCK+2Rφ 111 : 8RSPCK+2Rφ
1	SPNDL1	0	R/W	
0	SPNDL0	0	R/W	

15.3.13 RSPI 制御レジスタ 2 (SPCR2)

RSPI 制御レジスタ 2 (SPCR2) は、RSPI の動作モードを設定するためのレジスタです。SPCR2 は、常に CPU による読み出し/書き込みが可能です。SPE ビットが 1 で RSPI 機能がイネーブルである状態において、SPPE ビット、SPOE ビットの設定値を変更した場合には、以降の動作は保証されません。

ビット	7	6	5	4	3	2	1	0
ビット名	—	—	—	—	PTE	SPIIE	SPOE	SPPE
初期値 :	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R/W	R/W	R/W	R/W

ビット	ビット名	初期値	R/W	説明
7~4	—	すべて 0	R	リザーブビット 書き込む値は常に 0 にしてください。1 を書き込んだ場合の動作は保証しません。
3	PTE	0	R/W	パリティ自己診断ビット パリティ機能有効が正常であることを確認するために、パリティ回路の自己診断を有効にするビットです。 0 : パリティ回路の自己診断機能無効 1 : パリティ回路の自己診断機能有効
2	SPIIE	0	R/W	RSPI アイドル割り込みイネーブルビット RSPI がアイドル状態であることを検出し、RSPI ステータスレジスタ (SPSR) の IDLNF ビットを 0 にした場合の RSPI アイドル割り込み要求の発生を許可/禁止するためのビットです。 0 : アイドル割り込み要求の発生を禁止 1 : アイドル割り込み要求の発生を許可
1	SPOE	0	R/W	パリティモードビット 偶数パリティでは、パリティビットと送受信キャラクタをあわせて、1 の数の合計が偶数個になるようにパリティビットを決定します。同様に、奇数パリティでは、パリティビットと送受信キャラクタをあわせて、1 の数の合計が奇数個になるようにパリティビットを決定します。 SPOE は、SPCR2 の SPPE が 1 の時のみ有効です。 0 : 偶数パリティで送受信します 1 : 奇数パリティで送受信します

ビット	ビット名	初期値	R/W	説明
0	SPPE	0	R/W	<p>パリティイネーブルビット</p> <p>パリティ機能の有効、無効を選択するビットです。</p> <p>RSPI 制御レジスタ (SPCR) の通信動作モード選択ビット (TXMD) が 0、SPPE が 1 の時、送信データにパリティビットを付加し、受信データのパリティチェックを行います。</p> <p>TXMD が 1、SPPE が 1 の場合、送信データにパリティビットを付加しますが、受信データのパリティチェックを行いません。</p> <p>0 : 送信データにパリティビットを付加しません。受信データのパリティチェックを行いません</p> <p>1 : TXMD=0 のときは、送信データにパリティビットを付加し、受信データのパリティチェックを行います。</p> <p>TXMD=1 のときは、送信データにパリティビットを付加しますが、受信データのパリティチェックを行いません。</p>

15.3.14 RSPI コマンドレジスタ (SPCMD)

1 チャンネルの RSPI には、RSPI コマンドレジスタ (SPCMD) が 8 本あります (SPCMD0～SPCMD7)。SPCMD0～SPCMD7 は、マスターモードの RSPI の転送フォーマットを設定するために使用されます。また、SPCMD0 の一部のビットは、スレーブモードの RSPI の転送フォーマットを設定するためにも使用されます。マスターモードの RSPI は RSPI シーケンス制御レジスタ (SPSCR) の SPSLN2～SPSLN0 ビットの設定に従ってシーケンスに SPCMD0～7 を参照し、参照した SPCMD に設定されたシリアル転送を実行します。

SPCMD は、常に CPU による読み出し/書き込みが可能です。SPCMD レジスタの設定は RSPI ステータスレジスタ (SPSR) の SPTEF ビットが 1 の状態でその SPCMD を参照して送信するデータを設定する前に実施してください。マスターモードの RSPI が参照している SPCMD は、RSPI シーケンスステータスレジスタ (SPSSR) の SPCP2～0 ビットにより確認できます。また、スレーブモードの RSPI がイネーブルな状態において、SPCMD0 を CPU が書き換えた場合には、以降の動作は保証されません。

ビット	15	14	13	12	11	10	9	8
ビット名	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB3	SPB2	SPB1	SPB0
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	R/W
ビット	7	6	5	4	3	2	1	0
ビット名	SSLKP	—	SSLA1	SSLA0	BRDV1	BRDV0	CPOL	CPHA
初期値 :	0	0	0	0	1	1	0	1
R/W :	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

ビット	ビット名	初期値	R/W	説明
15	SCKDEN	0	R/W	<p>RSPCK 遅延設定イネーブル</p> <p>マスターモードの RSPI が、SSL 信号をアクティブにしてから RSPCK を発振するまでの期間 (RSPCK 遅延) を設定するためのビットです。SCKDEN が 0 の場合には、RSPI は RSPCK 遅延を 1RSPCK にします。SCKDEN が 1 の場合には、RSPI は RSPI クロック遅延レジスタ (SPCKD) の設定に従った RSPCK 遅延で RSPCK の発振を開始します。</p> <p>RSPI をスレーブモードで使用する場合には、SCKDEN に 0 を設定してください。</p> <p>0 : RSPCK 遅延は 1RSPCK</p> <p>1 : RSPCK 遅延は RSPCK 遅延レジスタ (SPCKD) の設定値</p>

ビット	ビット名	初期値	R/W	説明
14	SLNDEN	0	R/W	<p>SSL ネゲート遅延設定イネーブル</p> <p>マスタモードの RSPI が、RSPCK を発振停止してから SSL 信号を非アクティブにするまでの期間 (SSL ネゲート遅延) を設定するためのビットです。SLNDEN が 0 の場合には、RSPI は SSL ネゲート遅延を 1RSPCK にします。SLNDEN が 1 の場合には、RSPI はスレーブセレクトネゲート遅延レジスタ (SSLND) の設定に従った RSPCK 遅延で SSL をネゲートします。</p> <p>RSPI をスレーブモードで使用する場合には、SLNDEN に 0 を設定してください。</p> <p>0 : SSL ネゲート遅延は 1RSPCK 1 : SSL ネゲート遅延はスレーブセレクトネゲート遅延レジスタ (SSLND) の設定値</p>
13	SPNDEN	0	R/W	<p>RSPI 次アクセス遅延イネーブル</p> <p>マスタモードの RSPI がシリアル転送を終了して SSL 信号を非アクティブにしてから、次アクセスの SSL 信号アサートを可能にするまでの期間 (次アクセス遅延) を設定するためのビットです。SPNDEN が 0 の場合には、RSPI は次アクセス遅延を 1RSPCK+2Rφ にします。SPNDEN が 1 の場合には、RSPI は RSPI 次アクセス遅延レジスタ (SPND) の設定に従った次アクセス遅延を挿入します。</p> <p>RSPI をスレーブモードで使用する場合には、SPNDEN に 0 を設定してください。</p> <p>0 : 次アクセス遅延は 1RSPCK+2Rφ 1 : 次アクセス遅延は RSPI 次アクセス遅延レジスタ (SPND) の設定値</p>
12	LSBF	0	R/W	<p>RSPI LSB ファースト</p> <p>マスタモード/スレーブモードの RSPI のデータフォーマットを、MSB ファースト/LSB ファーストに設定するためのビットです。</p> <p>0 : MSB ファースト 1 : LSB ファースト</p>

ビット	ビット名	初期値	R/W	説明	
11	SPB3	0	R/W	RSPI データ長設定	
10	SPB2	1	R/W	マスタモード/スレーブモードの RSPI の転送データ長を設定するためのビットです。 0100~0111 : 8 ビット 1000 : 9 ビット 1001 : 10 ビット 1010 : 11 ビット 1011 : 12 ビット 1100 : 13 ビット 1101 : 14 ビット 1110 : 15 ビット 1111 : 16 ビット 0000 : 20 ビット 0001 : 24 ビット 0010、0011 : 32 ビット	
9	SPB1	1	R/W		
8	SPB0	1	R/W		
7	SSLKP	0	R/W		SSL 信号レベル保持 マスタモードの RSPI がシリアル転送する場合に、現コマンドに対応する SSL ネゲートタイミングから次コマンドに対応する SSL アサートタイミングの間に、現コマンドの SSL 信号レベルを保持するか、ネゲートするかを設定するビットです。 RSPI をスレーブモードで使用する場合には、SSLKP に 0 を設定してください。 0 : 転送終了時に全 SSL 信号をネゲート 1 : 転送終了後から次アクセス開始まで SSL 信号レベルを保持
6	—	0	R		リザーブビット 書き込む値は常に 0 にしてください。1 を書き込んだ場合の動作は保証しません。

ビット	ビット名	初期値	R/W	説明
5	SSLA1	0	R/W	SSL 信号アサート設定 マスタモードの RSPI がシリアル転送する場合の SSL 信号のアサートを制御するためのビットです。SSLA _i (i は 1, 0) の設定値が、SSL3~SSL0 信号のアサートを制御します。SSL 信号アサート時の信号極性は、RSPI スレーブセレクト極性レジスタ (SSLP) の設定値に依存します。マルチマスタモードで SSLA1、SSLA0 に B'00 を設定した場合には、全 SSL 信号がネゲート状態でシリアル転送が実行されます (SSL0 は入力になるため)。 RSPI をスレーブモードで使用する場合には、SSLA1、SSLA0 に B'00 を設定してください。 00 : SSL0 01 : SSL1 10 : SSL2 11 : SSL3
4	SSLA0	0	R/W	
3	BRDV1	1	R/W	ビットレート分周設定 ビットレートを決定するために使用するレジスタです。BRDV1、0 ビットと RSPI ビットレートレジスタ (SPBR) の設定値の組み合わせでビットレートを決定します (「15.3.8 RSPI ビットレートレジスタ (SPBR)」を参照)。SPBR の設定値は、ベースとなるビットレートを決定します。BRDV1~0 ビットの設定値は、ベースのビットレートに対して分周なし/2 分周/4 分周/8 分周したビットレートを選択するために使用します。SPCMD0~7 にはそれぞれ異なる BRDV1、0 の設定を行うことができます。このため、コマンドごとに異なるビットレートでシリアル転送を実行することが可能です。 00 : ベースのビットレートを選択 01 : ベースのビットレートの 2 分周を選択 10 : ベースのビットレートの 4 分周を選択 11 : ベースのビットレートの 8 分周を選択
2	BRDV0	1	R/W	
1	CPOL	0	R/W	RSPCK 極性設定 マスタモード/スレーブモードの RSPI の RSPCK 極性を設定するためのビットです。RSPI モジュール間のデータ通信を行う場合、モジュール間で同一の RSPCK 極性を設定する必要があります。 0 : アイドル時の RSPCK が 0 1 : アイドル時の RSPCK が 1
0	CPHA	1	R/W	RSPCK 位相設定 マスタモード/スレーブモードの RSPI の RSPCK 位相を設定するためのビットです。RSPI モジュール間のデータ通信を行う場合、モジュール間で同一の RSPCK 位相を設定する必要があります。 0 : 奇数エッジでデータサンプル、偶数エッジでデータ変化 1 : 奇数エッジでデータ変化、偶数エッジでデータサンプル

15.4 動作説明

本章では、シリアル転送期間という用語を、有効データのドライブ開始から最終有効データの取り込みまでの期間を意味する用語として使用しています。

15.4.1 RSPI 動作の概要

RSPI は、スレーブモード (SPI 動作)、シングルマスタモード (SPI 動作)、マルチマスタモード (SPI 動作)、スレーブモード (クロック同期式動作)、マスタモード (クロック同期式動作) での同期式のシリアル転送が可能です。RSPI のモードは、RSPI 制御レジスタ (SPCR) の MSTR ビットと MODFEN ビット、SPMS ビットによって設定可能です。表 15.5 に RSPI のモードと SPCR 設定の関係および各モードの概要をまとめます。

表 15.5 RSPI のモードと SPCR 設定の関係および各モードの概要

モード	スレーブ (SPI 動作)	シングルマスタ (SPI 動作)	マルチマスタ (SPI 動作)	スレーブ (クロック 同期式動作)	マスタ (クロック 同期式動作)
MSTR ビットの設定	0	1	1	0	1
MODFEN ビットの設定	0 or 1	0	1	0	0
SPMS ビットの設定	0	0	0	1	1
RSPCK 信号	入力	出力	出力/Hi-Z	入力	出力
MOSI 信号	入力	出力	出力/Hi-Z	入力	出力
MISO 信号	出力/Hi-Z	入力	入力	出力	入力
SSL0 信号	入力	出力	入力	Hi-Z	Hi-Z
SSL1~SSL3 信号	Hi-Z	出力/Hi-Z	出力/Hi-Z	Hi-Z	Hi-Z
出力端子モード	CMOS/ オープンドレイン	CMOS/ オープンドレイン	CMOS/ オープンドレイン	CMOS/ オープンドレイン	CMOS/ オープンドレイン
SSL 極性変更機能	あり	あり	あり	—	—
転送レート	~R ϕ /8	~R ϕ /2	~R ϕ /2	~R ϕ /8	~R ϕ /2
クロックソース	RSPCK 入力	内蔵ポーレート ジェネレータ	内蔵ポーレート ジェネレータ	RSPCK 入力	内蔵ポーレート ジェネレータ
クロック極性	2 種	2 種	2 種	2 種	2 種
クロック位相	2 種	2 種	2 種	1 種 (CPHA=1)	2 種
先頭転送ビット	MSB/LSB	MSB/LSB	MSB/LSB	MSB/LSB	MSB/LSB
転送データ長	8~32 ビット	8~32 ビット	8~32 ビット	8~32 ビット	8~32 ビット
バースト転送	可能 (CPHA=1)	可能 (CPHA=0,1)	可能 (CPHA=0,1)	—	—
RSPCK 遅延制御	なし	あり	あり	なし	あり
SSL ネゲート遅延制御	なし	あり	あり	なし	あり
次アクセス遅延制御	なし	あり	あり	なし	あり

モード	スレーブ (SPI 動作)	シングルマスタ (SPI 動作)	マルチマスタ (SPI 動作)	スレーブ (クロック 同期式動作)	マスタ (クロック 同期式動作)
転送起動方法	SSL 入力 アクティブ または RSPCK 発振	SPTEF=1 で 送信バッファ 書き込み	SPTEF=1 で 送信バッファ 書き込み	RSPCK 発振	SPTEF=1 で 送信バッファ 書き込み
シーケンス制御	なし	あり	あり	なし	あり
送信バッファエンプティ 検出	あり	あり	あり	あり	あり
受信バッファフル検出* ¹	あり	あり	あり	あり	あり
オーバーランエラー検出* ¹	あり	あり	あり	あり	あり
パリティエラー検出* ^{1,*2}	あり	あり	あり	あり	あり
モードフォルトエラー 検出	あり (MODFEN=1)	なし	あり	なし	なし

【注】 *1 SPCR レジスタの TXMD が 1 の時は、受信バッファフル検出、オーバーランエラー検出、パリティエラー検出を行いません。

*2 SPCR2 レジスタの SPPE が 0 の時は、パリティエラー検出を行いません。

15.4.2 RSPI 端子の制御

RSPI は、RSPI 制御レジスタ (SPCR) の MSTR ビット、MODFEN ビット、SPMS ビットと RSPI 端子制御レジスタ (SPPCR) の SPOM ビットの設定に従って、端子方向と出力モードを自動的に切り替えます。端子状態と各ビットの設定値の関係を表 15.6 に示します。

表 15.6 RSPI 端子の状態と制御ビット設定値の関係

モード	端子	端子状態*1	
		SPOM=0	SPOM=1
シングルマスタ (SPI 動作) (MSTR=1、MODFEN=0、 SPMS=0)	RSPCK	CMOS 出力	オープンドレイン出力
	SSL0	CMOS 出力	オープンドレイン出力
	SSL1~3	CMOS 出力/Hi-Z	オープンドレイン出力/Hi-Z
	MOSI	CMOS 出力	オープンドレイン出力
	MISO	入力	入力
マルチマスタ (SPI 動作) (MSTR=1、MODFEN=1、 SPMS=0)	RSPCK*2	CMOS 出力/Hi-Z	オープンドレイン出力/Hi-Z
	SSL0	入力	入力
	SSL1~3*2	CMOS 出力/Hi-Z	オープンドレイン出力/Hi-Z
	MOSI*2	CMOS 出力/Hi-Z	オープンドレイン出力/Hi-Z
	MISO	入力	入力
スレーブ (SPI 動作) (MSTR=0、SPMS=0)	RSPCK	入力	入力
	SSL0	入力	入力
	SSL1~3	Hi-Z	Hi-Z
	MOSI	入力	入力
	MISO*3	CMOS 出力/Hi-Z	オープンドレイン出力/Hi-Z
マスタ (クロック同期式動作) (MSTR=1、MODFEN=0、 SPMS=1)	RSPCK	CMOS 出力	オープンドレイン出力
	SSL0~3*4	Hi-Z	Hi-Z
	MOSI	CMOS 出力	オープンドレイン出力
	MISO	入力	入力
スレーブ (クロック同期式動作) (MSTR=0、SPMS=1)	RSPCK	入力	入力
	SSL0~3*4	Hi-Z	Hi-Z
	MOSI	入力	入力
	MISO	CMOS 出力	オープンドレイン出力

【注】 *1 RSPI 機能が選択されていないマルチファンクションピンには、RSPI の設定値は反映されません。

*2 SSL0 がアクティブレベルの場合、端子状態が Hi-Z になります。

*3 SSL0 が非アクティブレベルまたは SPCR の SPE ビットが 0 の場合、端子状態が Hi-Z になります。

*4 クロック同期式動作時は、SSL0~3 を I/O ポートとして使用可能です。

シングルマスタモード (SPI 動作)、マルチマスタモード (SPI 動作) の RSPI は、SPPCR の MOIFE ビットと MOIFV ビットの設定に従って、SSL ネゲート期間 (バースト転送における SSL 保持期間を含む) の MOSI 信号値を表 15.7 のように決定します。

表 15.7 SSL ネゲート期間の MOSI 信号値の決定方法

MOIFE	MOIFV	SSL ネゲート期間の MOSI 信号値
0	0、1	前回転送の最終データ
1	0	常に 0
1	1	常に 1

15.4.3 RSPI システム構成例

(1) シングルマスタ/シングルスレーブ (本 LSI=マスタ)

図 15.3 に、本 LSI をマスタとして使用した場合のシングルマスタ/シングルスレーブの RSPI システムの構成例を示します。シングルマスタ/シングルスレーブの構成では、本 LSI (マスタ) の SSL0~SSL3 出力は使用しません。RSPI スレーブの SSL 入力は 0 レベルに固定して、RSPI スレーブを常にセレクト状態にします。RSPI 制御レジスタ (SPCR) の CPHA ビットが 0 の場合に相当する転送フォーマットでは、SSL 信号をアクティブレベルに固定することができないスレーブデバイスも存在します。SSL 信号を固定にできない場合には、本 LSI の SSL 出力をスレーブデバイスの SSL 入りに接続してください。

本 LSI (マスタ) は、RSPCK と MOSI を常にドライブします。RSPI スレーブは、MISO を常にドライブします。

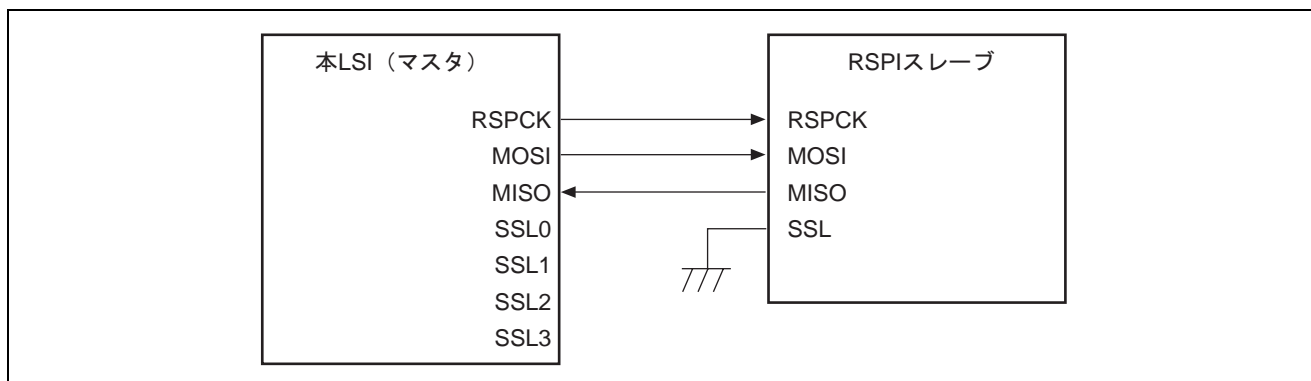


図 15.3 シングルマスタ/シングルスレーブの構成例 (本 LSI=マスタ)

(2) シングルマスタ/シングルスレーブ (本 LSI=スレーブ)

図 15.4 に、本 LSI をスレーブとして使用した場合のシングルマスタ/シングルスレーブの RSPI システム構成例を示します。本 LSI をスレーブとして使用する場合には、SSL0 端子を SSL 入力として使用します。RSPI マスタは、RSPCK と MOSI を常にドライブします。本 LSI (スレーブ) は、MISO を常にドライブします。SSL0 がアクティブレベルの場合、端子状態が Hi-Z になります。

RSPI コマンドレジスタ (SPCMD) の CPHA ビットを 1 に設定したシングルスレーブ構成の場合には、本 LSI (スレーブ) の SSL0 入力を 0 レベルに固定して本 LSI (スレーブ) を常に選択状態とし、シリアル転送を実行することも可能です (図 15.5)。

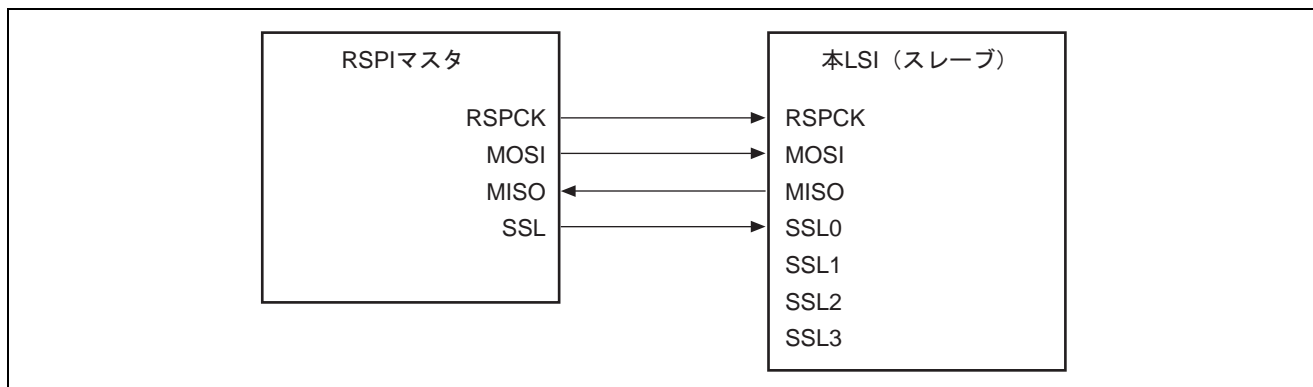


図 15.4 シングルマスタ/シングルスレーブの構成例 (本 LSI=スレーブ)

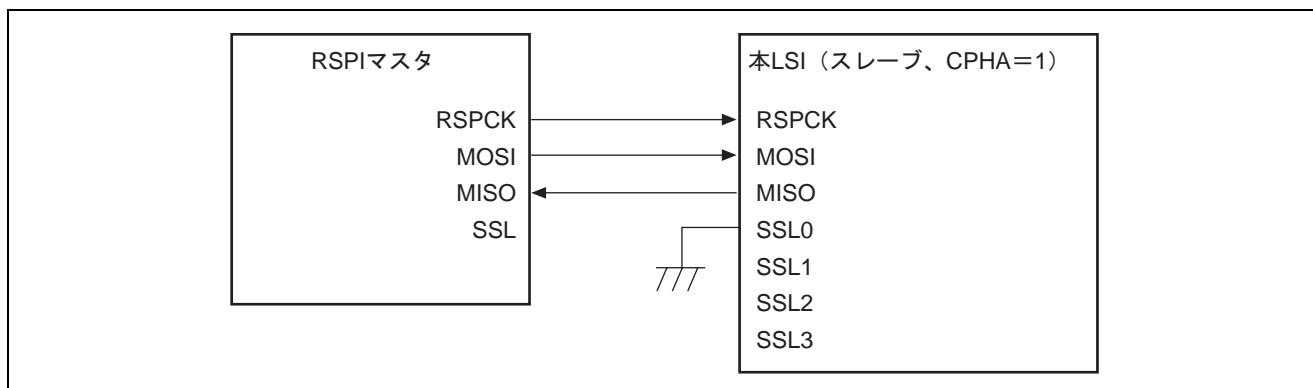


図 15.5 シングルマスタ/シングルスレーブの構成例 (本 LSI=スレーブ、CPHA=1)

(3) シングルマスタ/マルチスレーブ (本 LSI=マスタ)

図 15.6 に、本 LSI をマスタとして使用した場合のシングルマスタ/マルチスレーブの RSPI システム構成例を示します。図 15.5 の例では、本 LSI (マスタ) と 4 つのスレーブ (RSPI スレーブ 0~RSPI スレーブ 3) から RSPI システムを構成しています。

本 LSI (マスタ) の RSPCK 出力と MOSI 出力は、RSPI スレーブ 0~RSPI スレーブ 3 の RSPCK 入力と MOSI 入力に接続します。RSPI スレーブ 0~RSPI スレーブ 3 の MISO 出力は、すべて本 LSI (マスタ) の MISO 入力に接続します。本 LSI (マスタ) の SSL0~SSL3 出力は、それぞれ RSPI スレーブ 0~RSPI スレーブ 3 の SSL 入力に接続します。

本 LSI (マスタ) は、RSPCK、MOSI、SSL0~SSL3 を常にドライブします。RSPI スレーブ 0~RSPI スレーブ 3 のうち、SSL 入力が 0 レベルを入力されているスレーブが、MISO をドライブします。

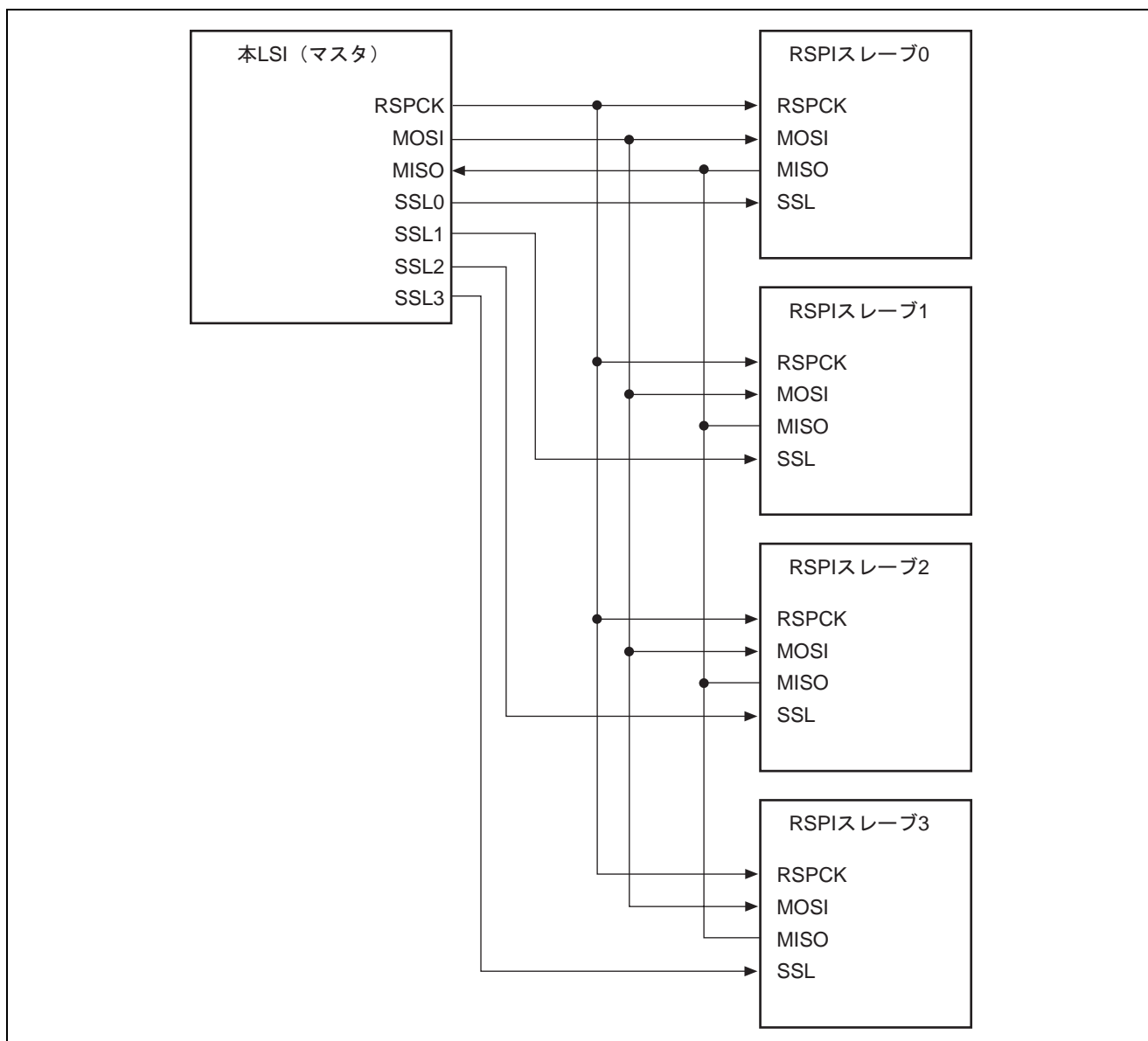


図 15.6 シングルマスタ/マルチスレーブの構成例 (本 LSI=マスタ)

(4) シングルマスタ/マルチスレーブ (本 LSI=スレーブ)

図 15.7 に、本 LSI をスレーブとして使用した場合のシングルマスタ/マルチスレーブの RSPI システム構成例を示します。図 15.7 の例では、RSPI マスタと 2 つの本 LSI (スレーブ X、スレーブ Y) から RSPI システムを構成しています。

RSPI マスタの RSPCK 出力と MOSI 出力は、本 LSI (スレーブ X、スレーブ Y) の RSPCK 入力と MOSI 入力に接続します。本 LSI (スレーブ X、スレーブ Y) の MISO 出力は、RSPI マスタの MISO 入力に接続します。RSPI マスタの SSLX 出力、SSLY 出力は、本 LSI (スレーブ X、スレーブ Y) の SSL0 入力に接続します。

RSPI マスタは、RSPCK、MOSI、SSLX、SSLY を常にドライブします。本 LSI (スレーブ X、スレーブ Y) のうち、SSL0 入力が 0 レベルを入力されているスレーブが、MISO をドライブします。

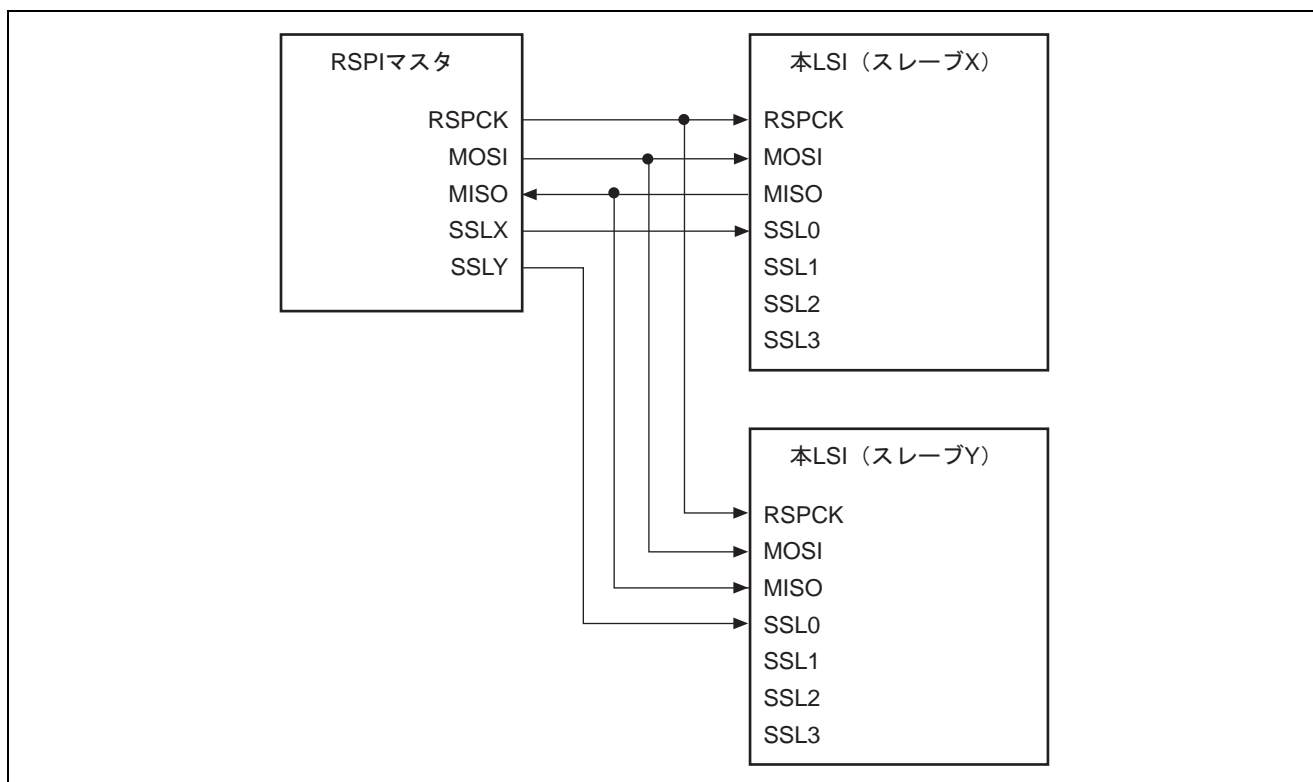


図 15.7 シングルマスタ/マルチスレーブの構成例 (本 LSI=スレーブ)

(5) マルチマスタ/マルチスレーブ (本 LSI=マスタ)

図 15.8 に、本 LSI をマスタとして使用した場合のマルチマスタ/マルチスレーブの RSPI システム構成例を示します。図 15.8 の例では、2 つの本 LSI (マスタ X、マスタ Y) と 2 つの RSPI スレーブ (RSPI スレーブ 1、RSPI スレーブ 2) から RSPI システムを構成しています。

本 LSI (マスタ X、マスタ Y) の RSPCK 出力と MOSI 出力は、RSPI スレーブ 1、RSPI スレーブ 2 の RSPCK 入力と MOSI 入力に接続します。RSPI スレーブ 1、RSPI スレーブ 2 の MISO 出力は、本 LSI (マスタ X、マスタ Y) の MISO 入力に接続します。本 LSI (マスタ X) の任意の汎用ポート Y 出力は、本 LSI (マスタ Y) の SSL0 入力に接続します。本 LSI (マスタ Y) の任意の汎用ポート X 出力は、本 LSI (マスタ X) の SSL0 入力に接続します。本 LSI (マスタ X、マスタ Y) の SSL1 出力と SSL2 出力は、RSPI スレーブ 1、RSPI スレーブ 2 の SSL 入力に接続します。この構成例では、SSL0 入力、スレーブ接続用の SSL1 出力、SSL2 出力のみでシステムを構成できるので、本 LSI の SSL3 出力を使用していません。

本 LSI は、SSL0 入力レベルが 1 の場合には、RSPCK、MOSI、SSL1、SSL2 をドライブします。SSL0 入力レベルが 0 の場合には、モードフォルトエラーを検出し、RSPCK、MOSI、SSL1、SSL2 を Hi-Z にして、他方のマスタに RSPI バス権を開放します。RSPI スレーブ 1、RSPI スレーブ 2 のうち、SSL 入力に 0 レベルを入力されているスレーブが、MISO をドライブします。

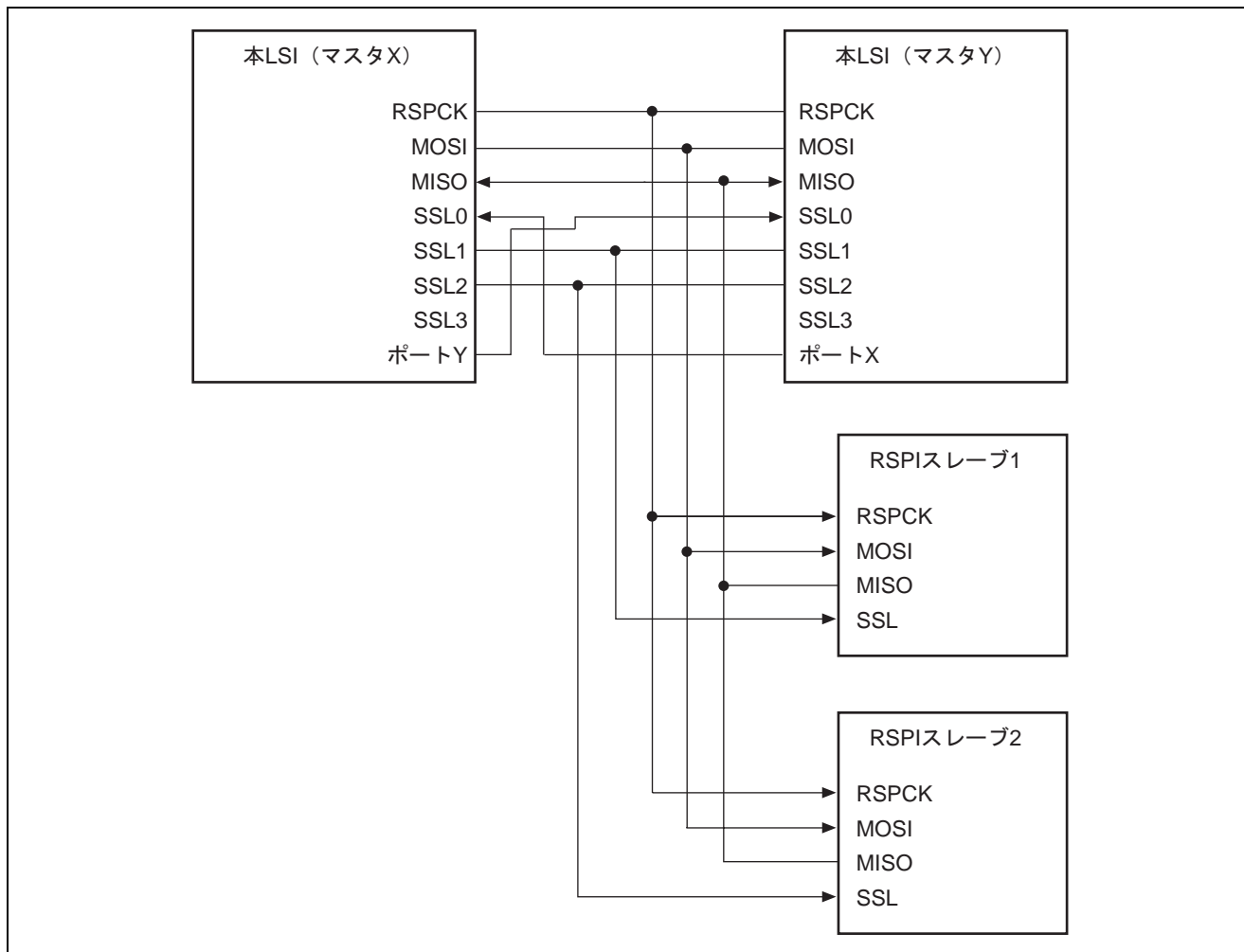


図 15.8 マルチマスタ/マルチスレーブの構成例 (本 LSI=マスタ)

(6) マスタ (クロック同期式動作) / スレーブ (クロック同期式動作) (本 LSI をマスタとして使用した場合)

図 15.9 に、本 LSI をマスタとして使用した場合のマスタ (クロック同期式動作) / スレーブ (クロック同期式動作) の RSPI システムの構成例を示します。マスタ (クロック同期式動作) / スレーブ (クロック同期式動作) の構成では、本 LSI (マスタ) の SSL0~SSL3 は使用しません。

本 LSI (マスタ) は、RSPCK と MOSI を常にドライブします。RSPI スレーブは、MISO を常にドライブします。

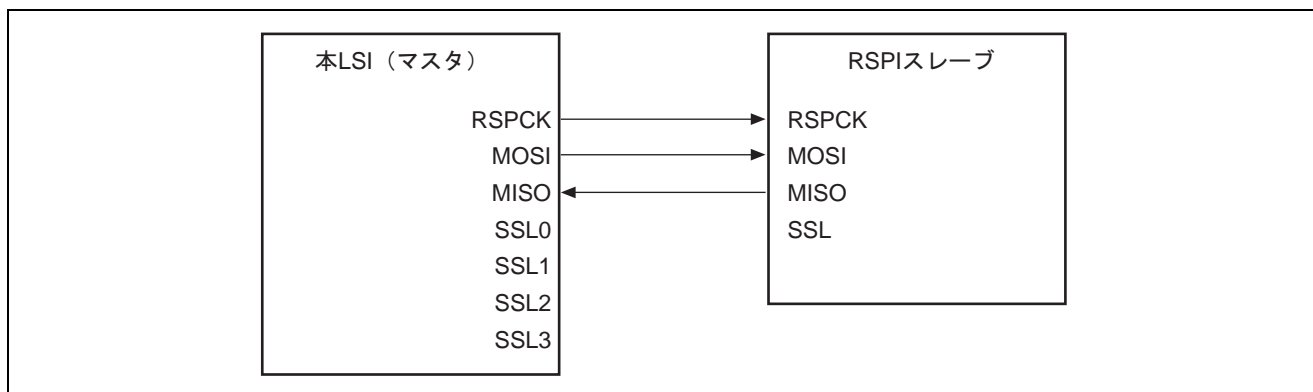


図 15.9 マスタ (クロック同期式動作) / スレーブ (クロック同期式動作) の構成例 (本 LSI = マスタ)

(7) マスタ (クロック同期式動作) / スレーブ (クロック同期式動作) (本 LSI をスレーブとして使用した場合)

図 15.10 に、本 LSI をスレーブとして使用した場合のマスタ (クロック同期式動作) / スレーブ (クロック同期式動作) の RSPI システム構成例を示します。本 LSI をスレーブ (クロック同期式動作) として使用する場合には、本 LSI (スレーブ) は、MISO を常にドライブし、RSPI マスタは、RSPCK と MOSI を常にドライブします。また、本 LSI (スレーブ) の SSL0~SSL3 は使用しません。

RSPI コマンドレジスタ (SPCMD) の CPHA ビットを 1 に設定したシングルスレーブ構成の場合のみ、本 LSI (スレーブ) はシリアル転送を実行することが可能です。

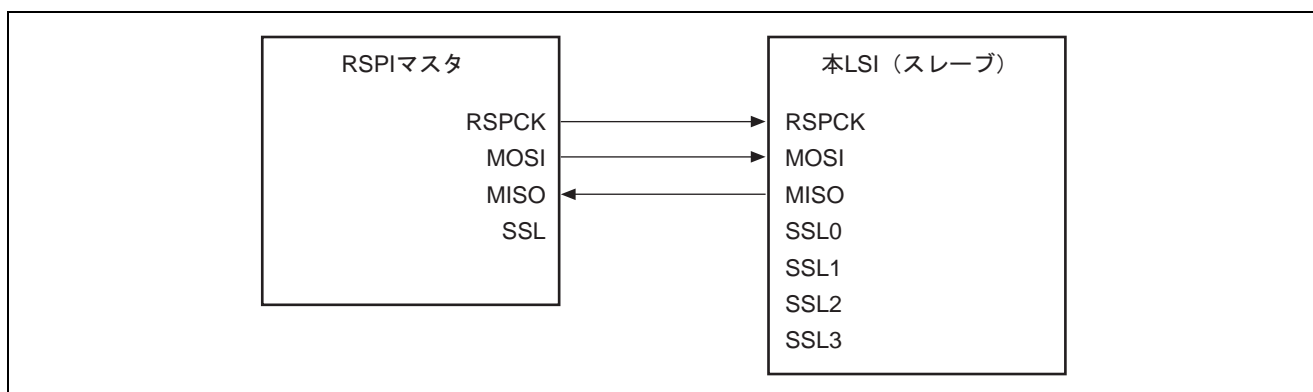


図 15.10 マスタ (クロック同期式動作) / スレーブ (クロック同期式動作) の構成例 (本 LSI = スレーブ、CPHA=1)

15.4.4 転送フォーマット

(1) CPHA=0 の場合

図 15.11 に RSPI コマンドレジスタ (SPCMD) の CPHA ビットが 0 の場合に、8 ビットのデータをシリアル転送した場合の転送フォーマット例を示します。ただし、RSPI がスレーブモード (MSTR=0) で、CPHA ビットが 0 の場合のクロック同期式動作 (RSPI 制御レジスタ (SPCR) の SPMS ビットが 1 の場合) は保証しません。図 15.11 において、RSPCK (CPOL=0) は SPCMD の CPOL ビットが 0 の場合、RSPCK (CPOL=1) は CPOL ビットが 1 の場合の RSPCK 信号波形です。サンプリングタイミングは、RSPI がシフトレジスタにシリアル転送データを取り込むタイミングを示しています。各信号の入出力方向は、RSPI の設定に依存します。詳細は「15.4.2 RSPI 端子の制御」を参照してください。

CPHA ビットが 0 の場合には、SSL 信号のアサートタイミングで、MOSI 信号と MISO 信号への有効データのドライブが開始されます。SSL 信号のアサート後に発生する最初の RSPCK 信号変化タイミングが最初の転送データ取り込みタイミングになり、このタイミング以降 1RSPCK 周期ごとにデータがサンプリングされます。MOSI 信号と MISO 信号の変化タイミングは、常に転送データ取り込みタイミングの $1/2$ RSPCK 周期後になります。CPOL ビットの設定値は、RSPCK 信号の動作タイミングには影響を与えず、信号極性のみに影響を与えます。

t_1 は、SSL 信号のアサートから RSPCK 発振までの期間 (RSPCK 遅延) です。 t_2 は、RSPCK 発振停止から SSL 信号のネゲートまでの期間 (SSL ネゲート遅延) です。 t_3 は、シリアル転送終了後に次転送のための SSL 信号アサートを抑制するための期間 (次アクセス遅延) です。 t_1 、 t_2 、 t_3 は、RSPI システム上のマスタデバイスによって制御されます。本 LSI の RSPI がマスタモードである場合の t_1 、 t_2 、 t_3 については、「15.4.10 (1) マスタモード動作」を参照してください。

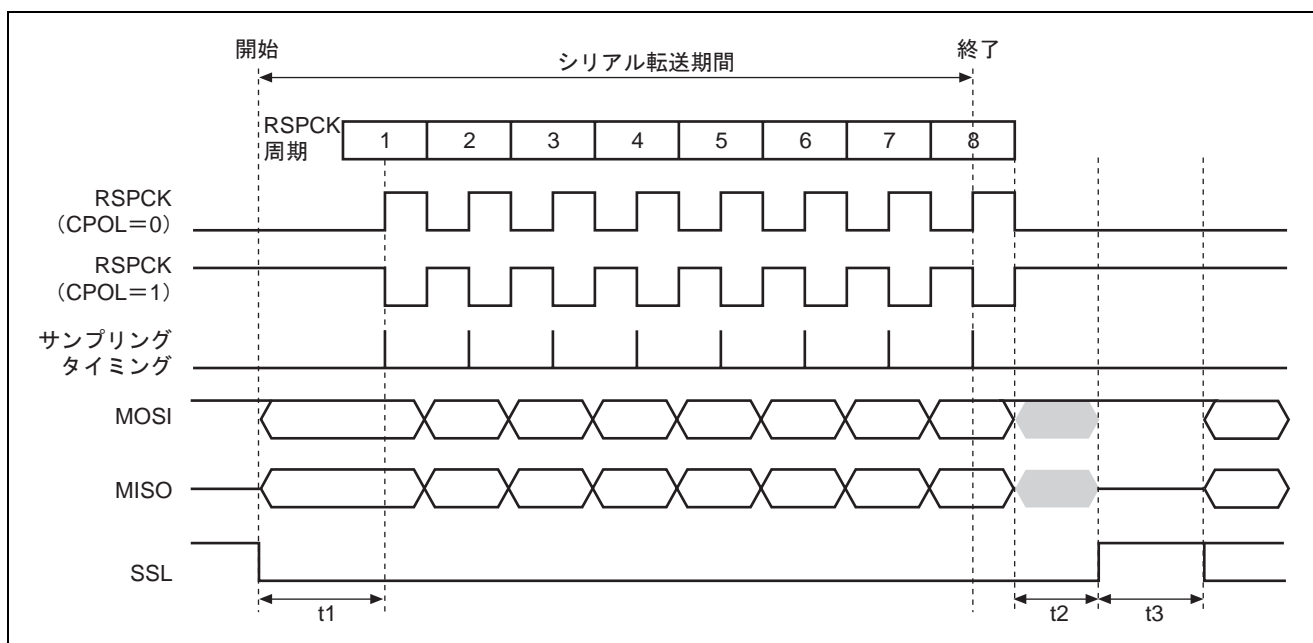


図 15.11 RSPI 転送フォーマット (CPHA=0)

(2) CPHA=1 の場合

図 15.12 に RSPI コマンドレジスタ (SPCMD) の CPHA ビットが 1 の場合に、8 ビットのデータをシリアル転送した場合の転送フォーマット例を示します。ただし、RSPI 制御レジスタ (SPCR) の SPMS ビットが 1 の場合は SSL 信号を用いず、RSPCK 信号、MOSI 信号、MISO 信号のみで通信を行います。図 15.12 において、RSPCK (CPOL=0) は SPCMD の CPOL ビットが 0 の場合、RSPCK (CPOL=1) は CPOL ビットが 1 の場合の RSPCK 信号波形です。サンプリングタイミングは、RSPI がシフトレジスタにシリアル転送データを取り込むタイミングを示しています。各信号の入出力方向は、RSPI のモード (マスタ/スレーブ) に依存します。詳細は「15.4.2 RSPI 端子の制御」を参照してください。

CPHA ビットが 1 の場合には、SSL 信号のアサートタイミングで、MOSI 信号と MISO 信号に無効データのドライブが開始されます。SSL 信号のアサート後に発生する最初の RSPCK 信号変化タイミングで、MOSI 信号と MISO 信号への有効データのドライブが開始され、このタイミング以降 1RSPCK 周期ごとにデータが更新されます。転送データの取り込みは、常にこのタイミングの 1/2RSPCK 周期後になります。CPOL ビットの設定値は RSPCK 信号の動作タイミングには影響を与えず、信号極性のみに影響を与えます。

t1、t2、t3 の内容は、CPHA=0 の場合と同様です。本 LSI の RSPI がマスタモードである場合の t1、t2、t3 については、「15.4.10 (1) マスタモード動作」を参照してください。

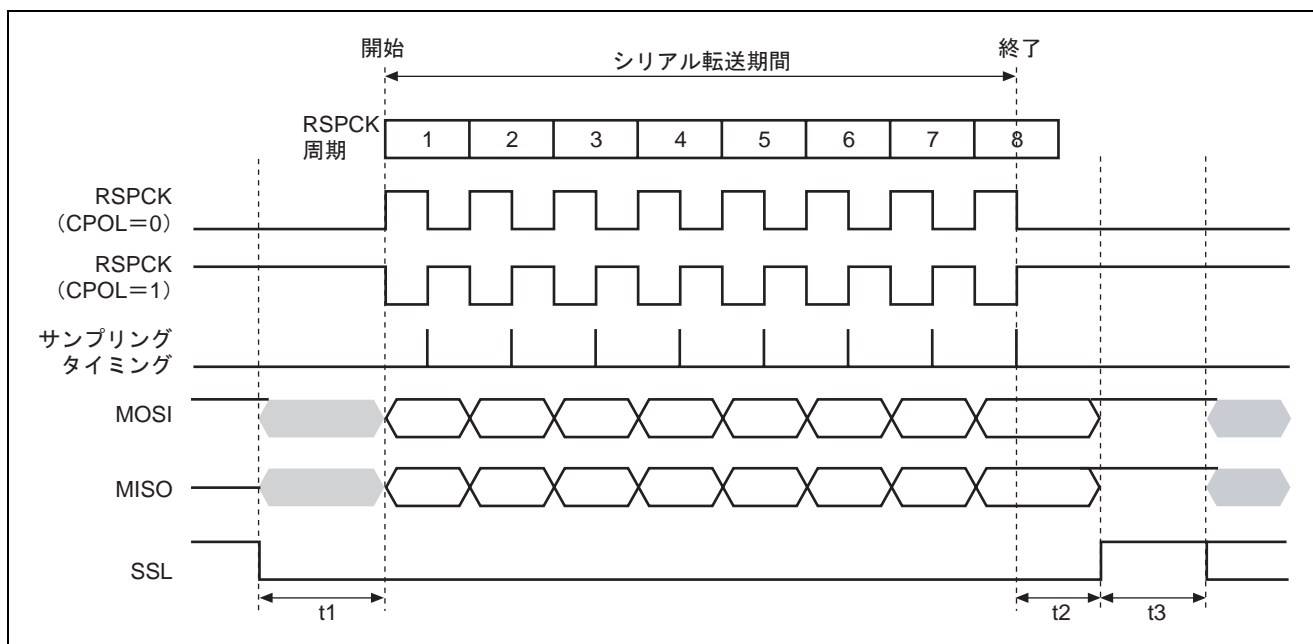


図 15.12 RSPI 転送フォーマット (CPHA=1)

15.4.5 データフォーマット

RSPI のデータフォーマットは、RSPI コマンドレジスタ (SPCMD) 、RSPI 制御レジスタ 2 (SPCR2) のパリティイネーブルビット (SPPE) の設定値に依存します。MSB/LSB ファーストにかかわらず、RSPI は RSPI データレジスタ (SPDR) の LSB から設定データ長分の範囲を転送データとして扱います。

(1) MSB ファースト転送 (32 ビットデータ)

(1-1) パリティ機能無効時 (SPPE=0)

図 15.13 に、パリティ機能無効時、RSPI がデータ長 32 ビットの MSB ファースト転送を実施する場合の RSPI データレジスタ (SPDR) とシフトレジスタの動作内容を示します。

CPU または DMAC によって、SPDR の送信バッファに T31~T00 を書き込みます。RSPI ステータスレジスタ (SPSR) の SPTEF ビットが 0 かつシフトレジスタが空であれば、RSPI が送信バッファのデータをシフトレジスタにコピーし、シフトレジスタをフルにします。シリアル転送が開始されると、RSPI はシフトレジスタの MSB (ビット 31) からデータを出力し、シフトレジスタの LSB (ビット 0) からデータをシフトインします。32 ビット分のシリアル転送に必要な RSPCK 周期を経過すると、シフトレジスタにはデータ R31~R00 が格納されます。この状態で、全二重同期式のシリアル通信動作 (TXMD=0) であれば、RSPI はシフトレジスタから SPDR の受信バッファにデータをコピーし、シフトレジスタを空にします。

なお、CPU または DMAC が SPDR の送信バッファを書き込む前に、次のシリアル転送が起動されると、受信データ R31~R00 がシフトレジスタからシフトアウトされます。

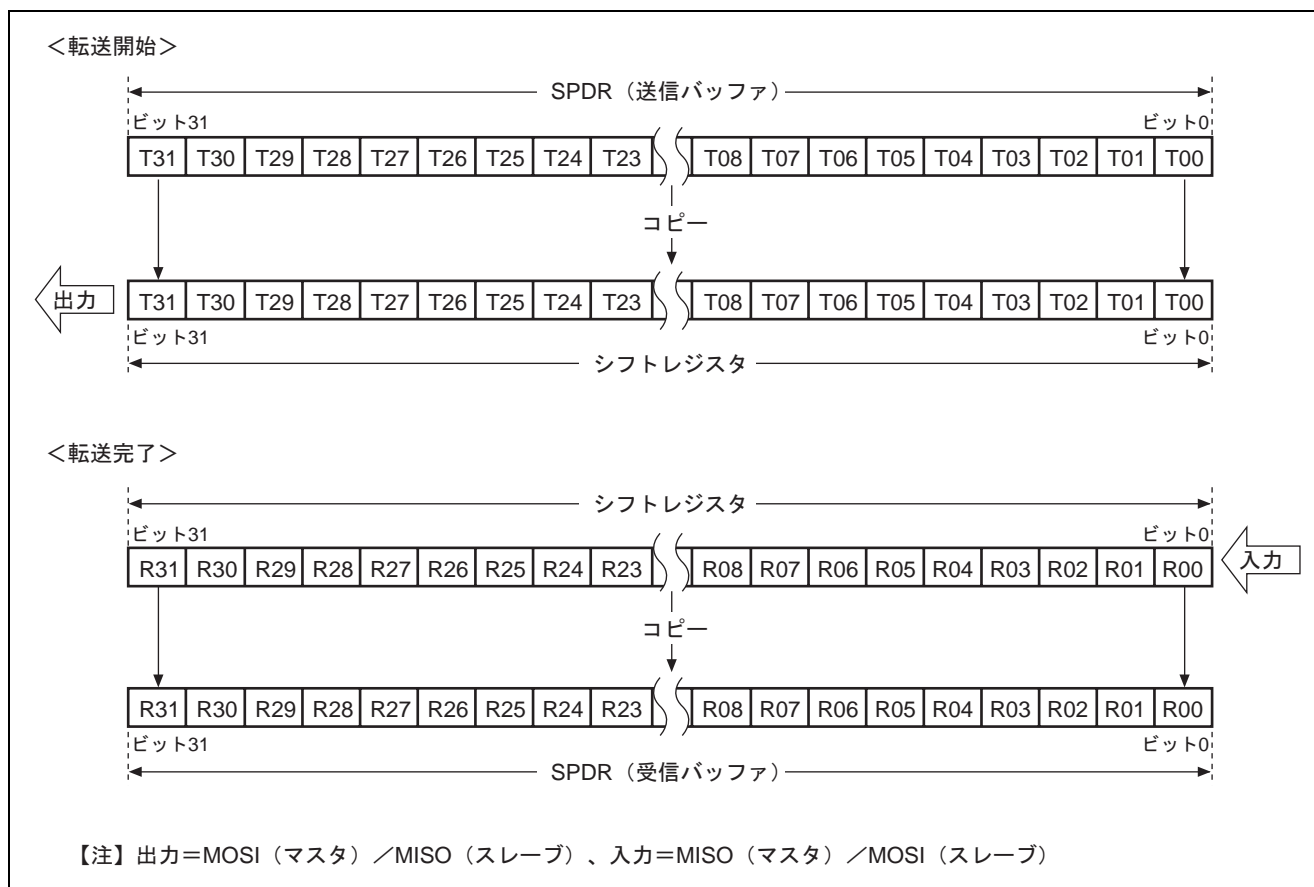


図 15.13 MSB ファースト転送 (32 ビットデータ / パリティ機能無効)

(1-2) パリティ機能有効時 (SPPE=1)

図 15.14 に、パリティ機能有効時、RSPI がデータ長 32 ビットの MSB ファースト転送を実施する場合の RSPI データレジスタ (SPDR) とシフトレジスタの動作内容を示します。

CPU または DMAC によって SPDR の送信バッファに T31~T00 を書き込みます。RSPI ステータスレジスタ (SPSR) の SPTEF ビットが 0 かつシフトレジスタが空であれば、RSPI が SPDR の送信バッファに格納されたデータの T00 をパリティビット (P) に変換します。パリティビット (P) を付加したデータをシフトレジスタにコピーし、シフトレジスタをフルにします。シリアル転送が開始されると、RSPI はシフトレジスタの MSB (ビット 31) からデータを出力し、シフトレジスタの LSB (ビット 0) からデータをシフトインします。

32 ビット分のシリアル転送に必要な RSPCK 周期を経過すると、シフトレジスタにはデータ R31~P が格納されます。この状態で、全二重同期式のシリアル通信動作 (TXMD=0) であれば、RSPI はシフトレジスタから SPDR の受信バッファにデータをコピーし、シフトレジスタを空にします。

なお、CPU または DMAC が SPDR の送信バッファを書き込む前に、次のシリアル転送が起動されると、受信データ R31~P がシフトレジスタからシフトアウトされます。

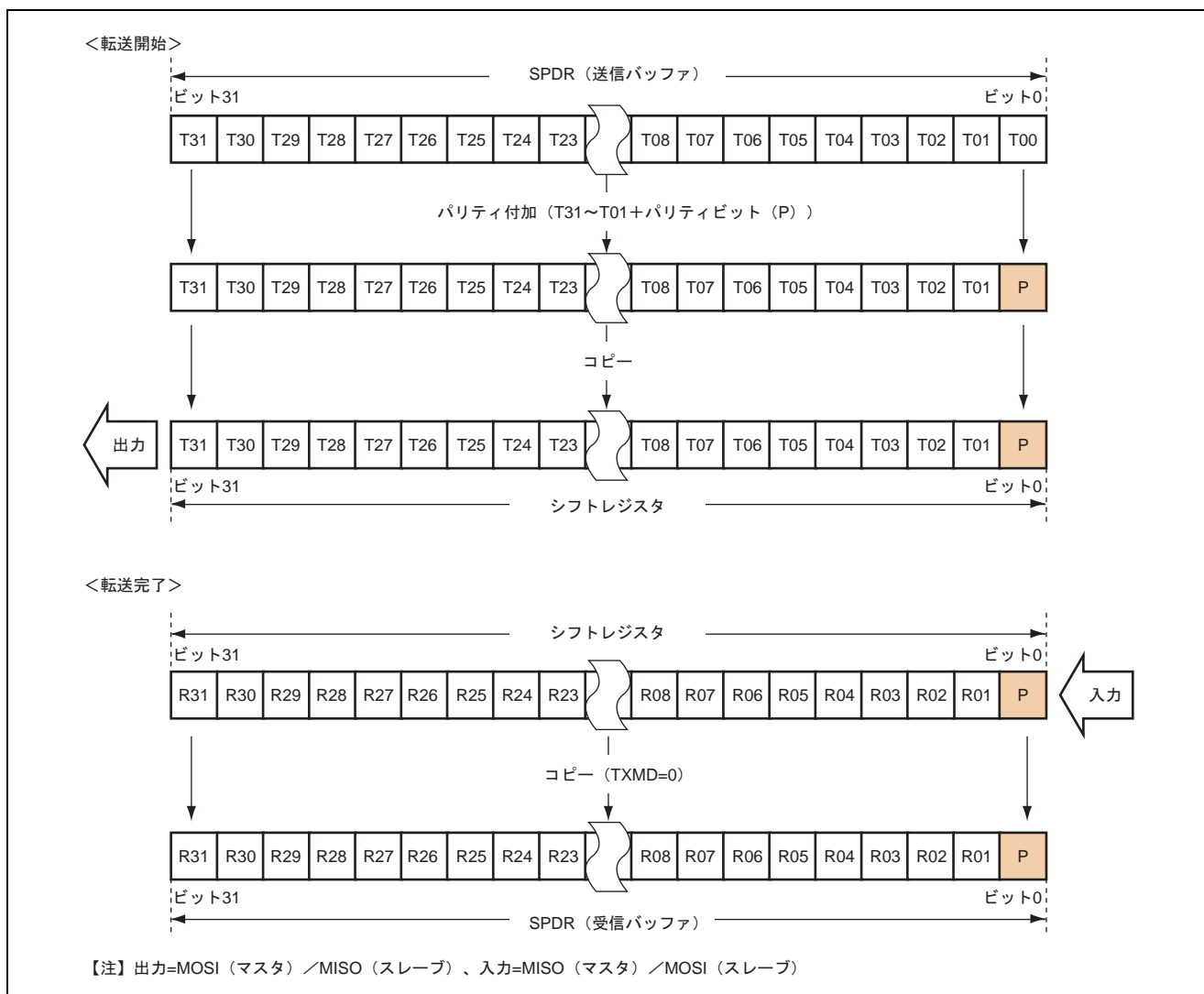


図 15.14 MSB ファースト転送 (32 ビットデータ / パリティ機能有効)

(2) MSB ファースト転送 (24 ビットデータ)

(2-1) パリティ機能無効時 (SPPE=0)

図 15.15 に、RSPI がパリティ機能無効時、32 ビット以外のデータを MSB ファースト転送する例として、24 ビットのデータ転送を実施する場合の RSPI データレジスタ (SPDR) とシフトレジスタの動作内容を示します。

CPU または DMAC は、SPDR の送信バッファに T31~T00 を書き込みます。RSPI ステータスレジスタ (SPSR) の SPTEF ビットが 0 かつシフトレジスタが空であれば、RSPI が SPDR の送信バッファのデータをシフトレジスタにコピーし、シフトレジスタをフルにします。シリアル転送が開始されると、RSPI はシフトレジスタのビット 23 からデータを出力し、シフトレジスタの LSB (ビット 0) からデータをシフトインします。24 ビット分のシリアル転送に必要な RSPCK 周期を経過すると、シフトレジスタのビット 23~0 には受信データ R23~R00 が格納されます。シリアル転送完了後のシフトレジスタのビット 31~24 には、転送前のデータが保持されています。この状態で、全二重同期式のシリアル通信動作 (TXMD=0) であれば、RSPI がシフトレジスタから SPDR の受信バッファにデータをコピーし、シフトレジスタを空にします。

なお、CPU または DMAC が SPDR の送信バッファを書き込む前に、次のシリアル転送が起動されると、受信データ R23~R00 がシフトレジスタからシフトアウトされます。

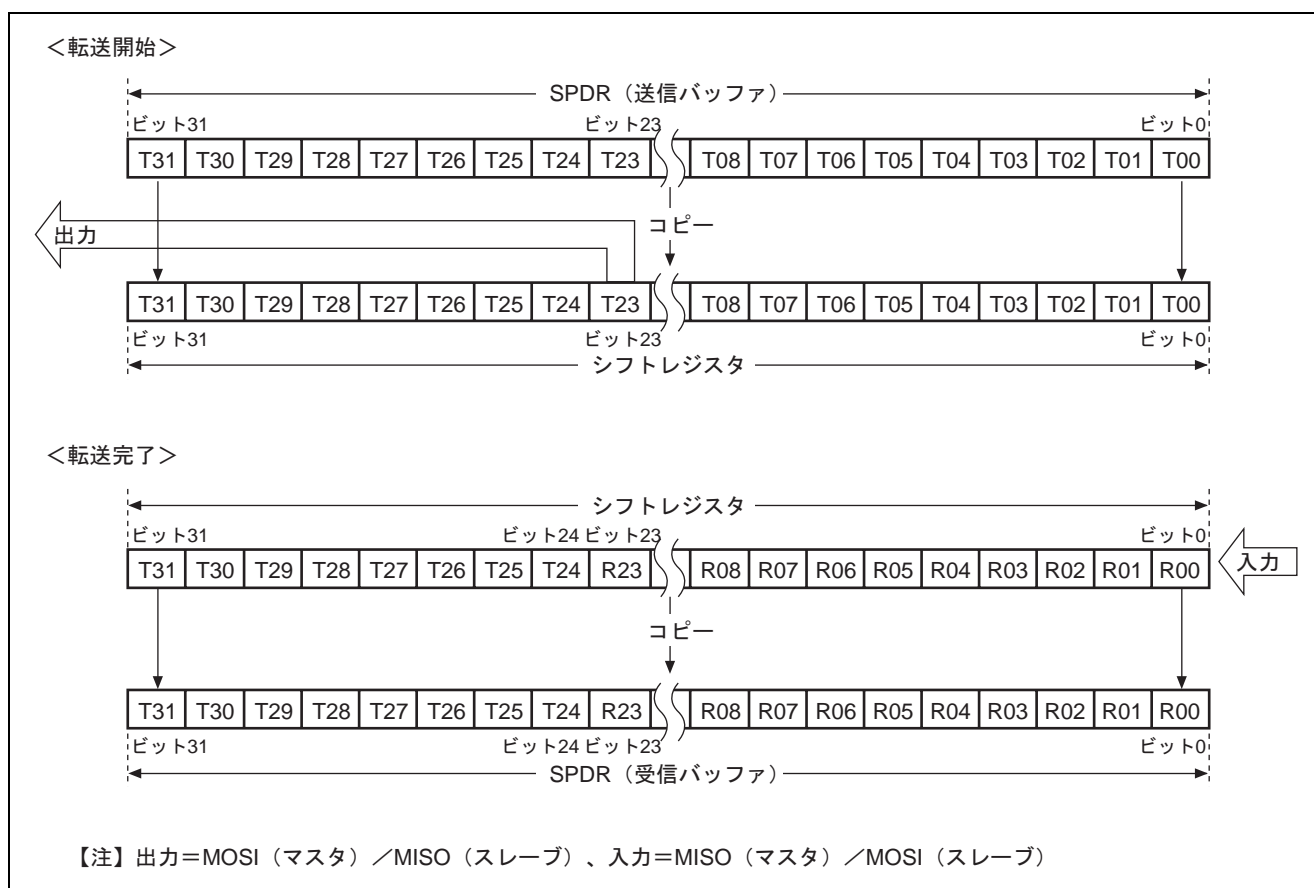


図 15.15 MSB ファースト転送 (24 ビットデータ / パリティ機能無効)

(2-2) パリティ機能有効時 (SPPE=1)

図 15.16 に、RSPI がパリティ機能有効時、32 ビット以外のデータを MSB ファースト転送する例として、24 ビットのデータ転送を実施する場合の RSPI データレジスタ (SPDR) とシフトレジスタの動作内容を示します。

CPU または DMAC は、SPDR の送信バッファに T31~T00 を書き込みます。RSPI ステータスレジスタ (SPSR) の SPTEF ビットが 0 かつシフトレジスタが空であれば、RSPI が SPDR の送信バッファに格納されたデータの T00 をパリティビット (P) に変換します。パリティビット (P) を付加したデータをシフトレジスタにコピーし、シフトレジスタをフルにします。シリアル転送が開始されると、RSPI はシフトレジスタのビット 23 からデータを出し、シフトレジスタの LSB (ビット 0) からデータをシフトインします。24 ビット分のシリアル転送に必要な RSPCK 周期を経過すると、シフトレジスタのビット 23~ビット 0 には受信データ R23~P が格納されます。シリアル転送完了後のシフトレジスタのビット 24~ビット 31 には、転送前のデータが保持されています。この状態で、全二重同期式のシリアル通信動作 (TXMD=0) であれば、RSPI がシフトレジスタから SPDR の受信バッファにデータをコピーし、シフトレジスタを空にします。

なお、CPU または DMAC が SPDR の送信バッファを書き込む前に、次のシリアル転送が起動されると、受信データ R23~P がシフトレジスタからシフトアウトされます。

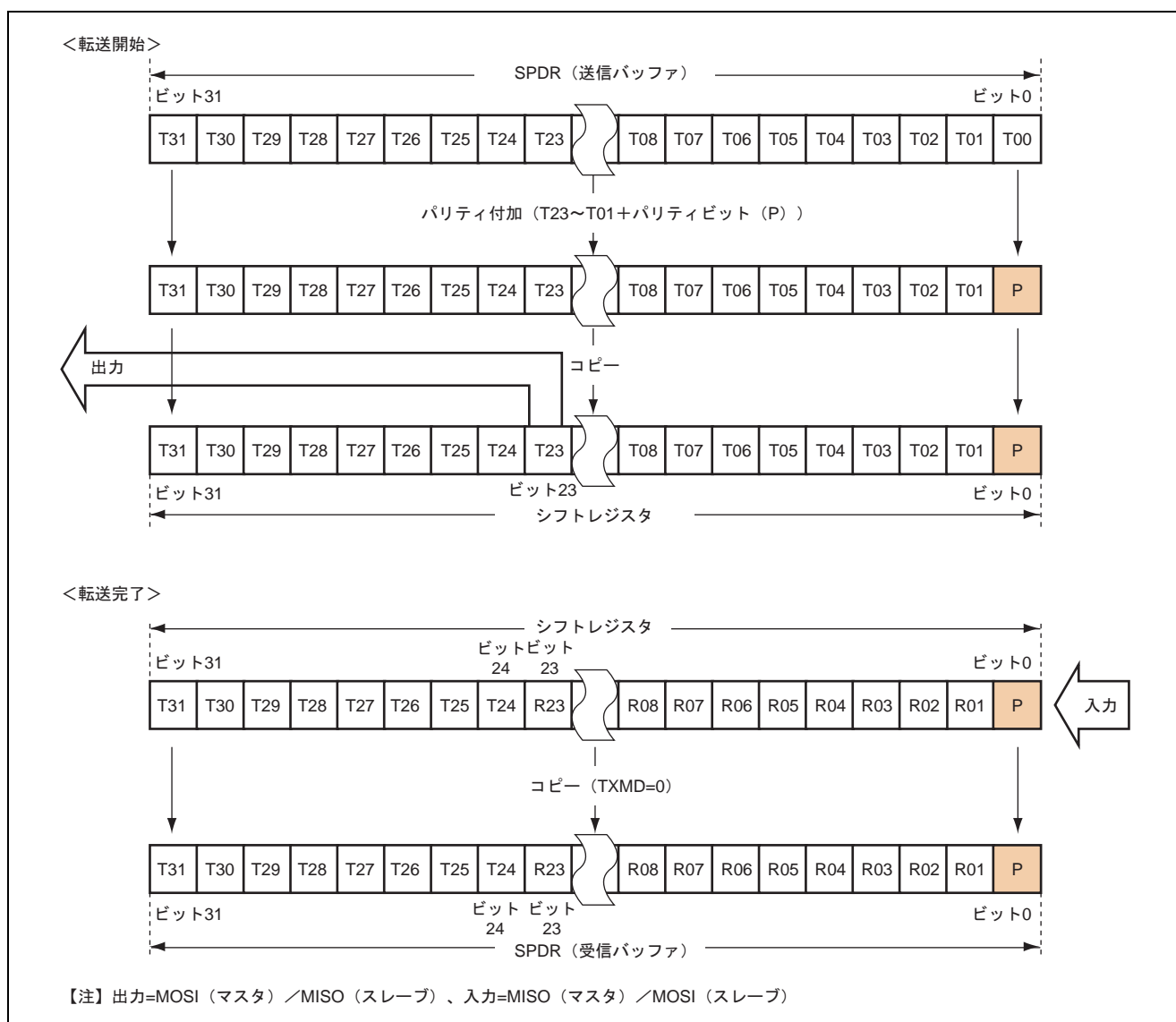


図 15.16 MSB ファースト転送 (24 ビットデータ / パリティ機能有効)

(3) 0LSB ファースト転送 (32 ビットデータ)

(3-1) パリティ機能無効時 (SPPE=0)

図 15.17 に、RSPI がパリティ機能無効時、データ長 32 ビットの LSB ファースト転送を実施する場合の RSPI データレジスタ (SPDR) とシフトレジスタの動作内容を示します。

CPU または DMAC は、SPDR の送信バッファに T31~T00 を書き込みます。RSPI ステータスレジスタ (SPSR) の SPTEF ビットが 0 かつシフトレジスタが空であれば、RSPI が SPDR の送信バッファのデータをシフトレジスタにビット順を逆転してコピーし、シフトレジスタをフルにします。シリアル転送が開始されると、RSPI はシフトレジスタの MSB (ビット 31) からデータを出力し、シフトレジスタの LSB (ビット 0) からデータをシフトインします。32 ビット分のシリアル転送に必要な RSPCK 周期を経過すると、シフトレジスタにはデータ R00~R31 が格納されます。この状態で、全二重同期式のシリアル通信動作 (TXMD=0) であれば、RSPI はシフトレジスタから SPDR の受信バッファにビット順を逆転したデータをコピーし、シフトレジスタを空にします。

なお、CPU または DMAC が SPDR の送信バッファを書き込む前に、次のシリアル転送が起動されると、受信データ R00~R31 がシフトレジスタからシフトアウトされます。

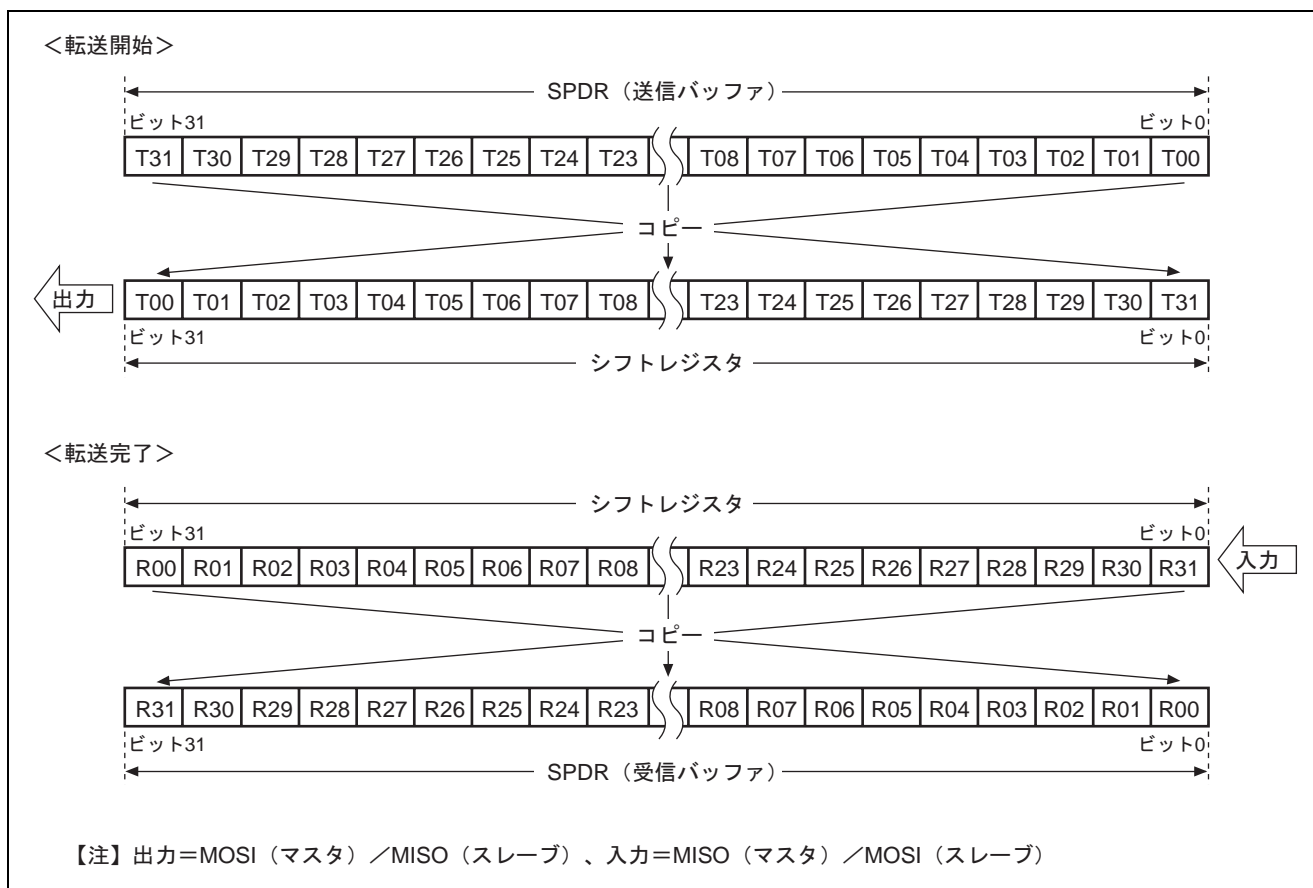


図 15.17 LSB ファースト転送 (32 ビットデータ / パリティ機能無効)

(3-2) パリティ機能有効時 (SPPE=1)

図 15.18 に、RSPI がパリティ機能有効時、データ長 32 ビットの LSB ファースト転送を実施する場合の RSPI データレジスタ (SPDR) とシフトレジスタの動作内容を示します。

CPU または DMAC は、SPDR の送信バッファに T31~T00 を書き込みます。RSPI が SPDR の送信バッファに格納されたデータの T31 をパリティビット (P) に変換します。RSPI ステータスレジスタ (SPSR) の SPTEF ビットが 0 かつシフトレジスタが空であれば、パリティビット (P) を付加したデータは、シフトレジスタへビット順を逆転してコピーし、シフトレジスタをフルにします。シリアル転送が開始されると、RSPI はシフトレジスタの MSB (ビット 31) からデータを出力し、シフトレジスタの LSB (ビット 0) からデータをシフトインします。32 ビット分のシリアル転送に必要な RSPCK 周期を経過すると、シフトレジスタにはデータ R00~P が格納されます。この状態で、全二重同期式のシリアル通信動作 (TXMD=0) であれば、RSPI はシフトレジスタから SPDR の受信バッファにビット順を逆転したデータをコピーし、シフトレジスタを空にします。

なお、CPU または DMAC が SPDR の送信バッファを書き込む前に、次のシリアル転送が起動されると、受信データ R00~P がシフトレジスタからシフトアウトされます。

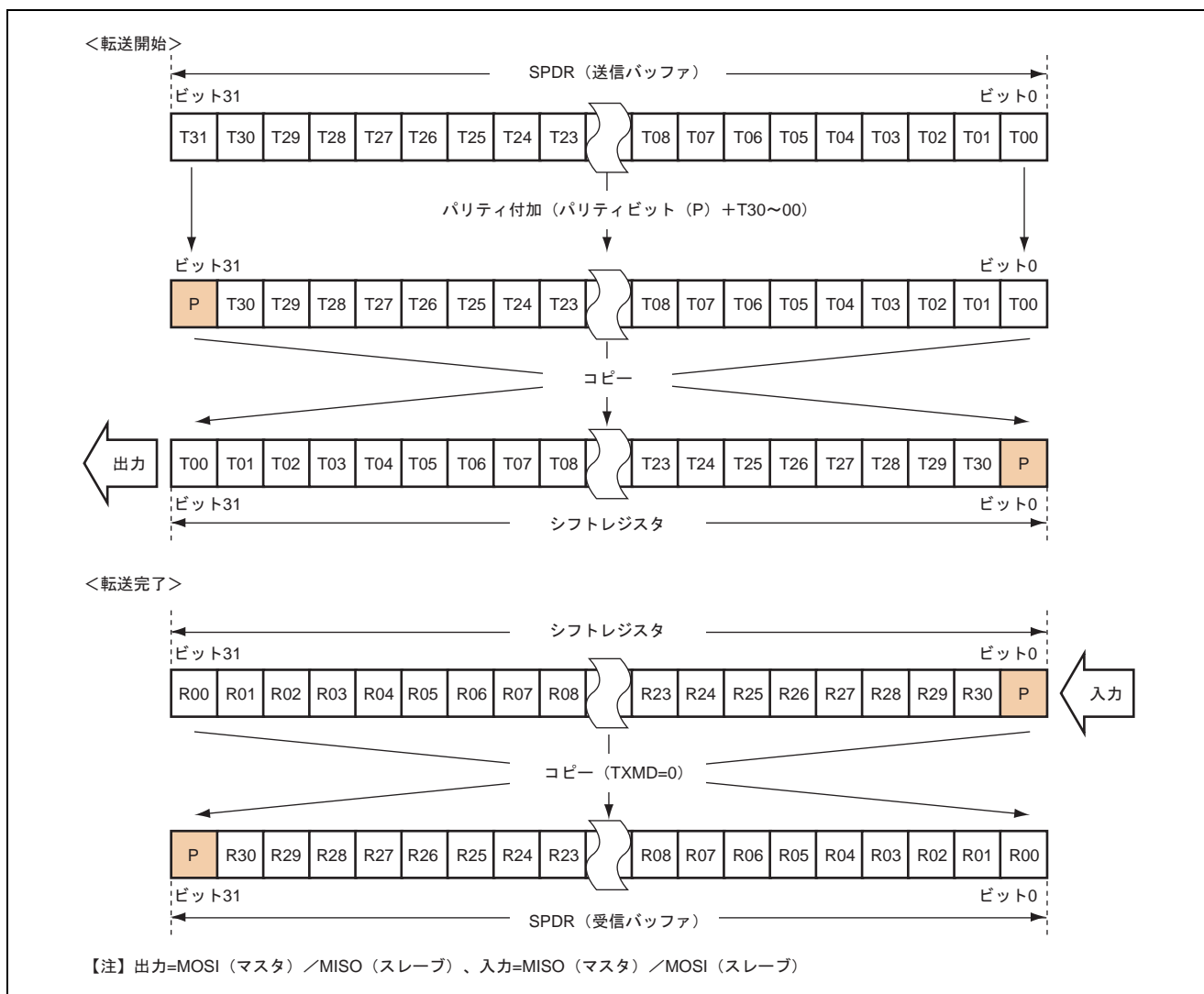


図 15.18 LSB ファースト転送 (32 ビットデータ / パリティ機能有効)

(4) LSB ファースト転送 (24 ビットデータ)

(4-1) パリティ機能無効時 (SPPE=0)

図 15.19 に、RSPI がパリティ機能無効時、32 ビット以外のデータを LSB ファースト転送する例として、24 ビットのデータ転送を実施する場合の RSPI データレジスタ (SPDR) とシフトレジスタの動作内容を示します。

CPU または DMAC は、SPDR の送信バッファに T31~T00 を書き込みます。RSPI ステータスレジスタ (SPSR) の SPTEF ビットが 0 かつシフトレジスタが空であれば、RSPI が SPDR の送信バッファのデータをシフトレジスタにビット順を逆転してコピーし、シフトレジスタをフルにします。シリアル転送が開始されると、RSPI はシフトレジスタの MSB (ビット 31) からデータを出力し、シフトレジスタのビット 8 からデータをシフトインします。24 ビット分のシリアル転送に必要な RSPCK 周期を経過すると、シフトレジスタのビット 31~8 には受信データ R00~R23 が格納されます。シリアル転送完了後のシフトレジスタのビット 7~0 には、転送前のデータが保持されています。この状態で、全二重同期式のシリアル通信動作 (TXMD = 0) であれば、RSPI がシフトレジスタから SPDR の受信バッファにビット順を逆転したデータをコピーし、シフトレジスタを空にします。

なお、CPU または DMAC が SPDR の送信バッファを書き込む前に、次のシリアル転送が起動されると、受信データ R00~R23 がシフトレジスタからシフトアウトされます。

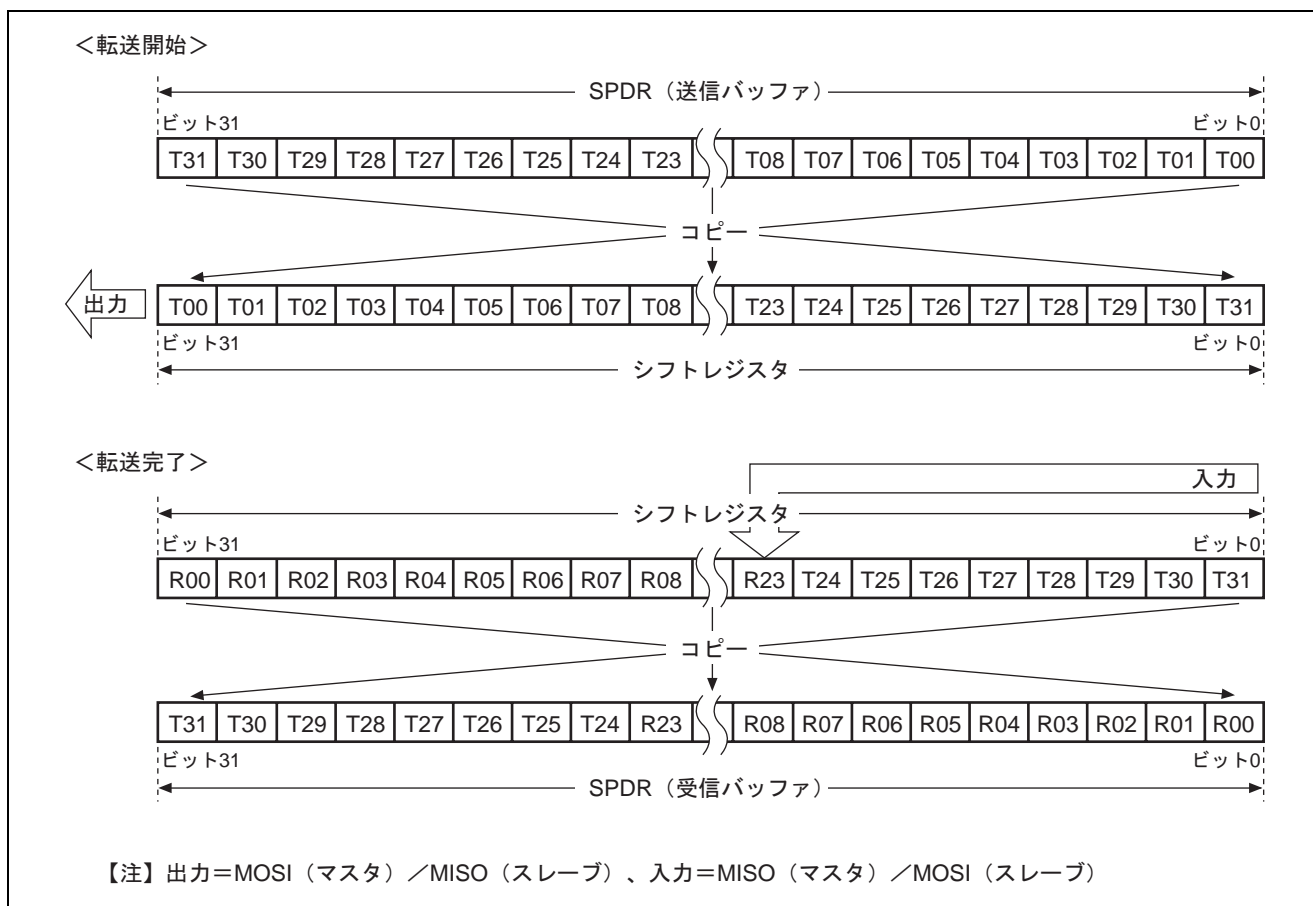


図 15.19 LSB ファースト (24 ビットデータ / パリティ機能無効)

(4-2) パリティ機能有効時 (SPPE=1)

図 15.20 に、RSPI がパリティ機能有効時、32 ビット以外のデータを LSB ファースト転送する例として、24 ビットのデータ転送を実施する場合の RSPI データレジスタ (SPDR) とシフトレジスタの動作内容を示します。

CPU または DMAC は、SPDR の送信バッファに T31~T00 を書き込みます。RSPI が SPDR の送信バッファに格納されたデータの T23 をパリティビット (P) に変換します。RSPI ステータスレジスタ (SPSR) の SPTEF ビットが 0 かつシフトレジスタが空であれば、パリティビット (P) を付加したデータは、シフトレジスタへビット順を逆転してコピーし、シフトレジスタをフルにします。シリアル転送が開始されると、RSPI はシフトレジスタの MSB (ビット 31) からデータを出力し、シフトレジスタのビット 8 からデータをシフトインします。24 ビット分のシリアル転送に必要な RSPCK 周期を経過すると、シフトレジスタのビット 31~ビット 8 には受信データ R00~P が格納されます。シリアル転送完了後のシフトレジスタのビット 7~ビット 0 には、転送前のデータが保持されています。この状態で、全二重同期式のシリアル通信動作 (TXMD=0) であれば、RSPI がシフトレジスタから SPDR の受信バッファにビット順を逆転したデータをコピーし、シフトレジスタを空にします。

なお、CPU または DMAC が SPDR の送信バッファを書き込む前に、次のシリアル転送が起動されると、受信データ R00~P がシフトレジスタからシフトアウトされます。

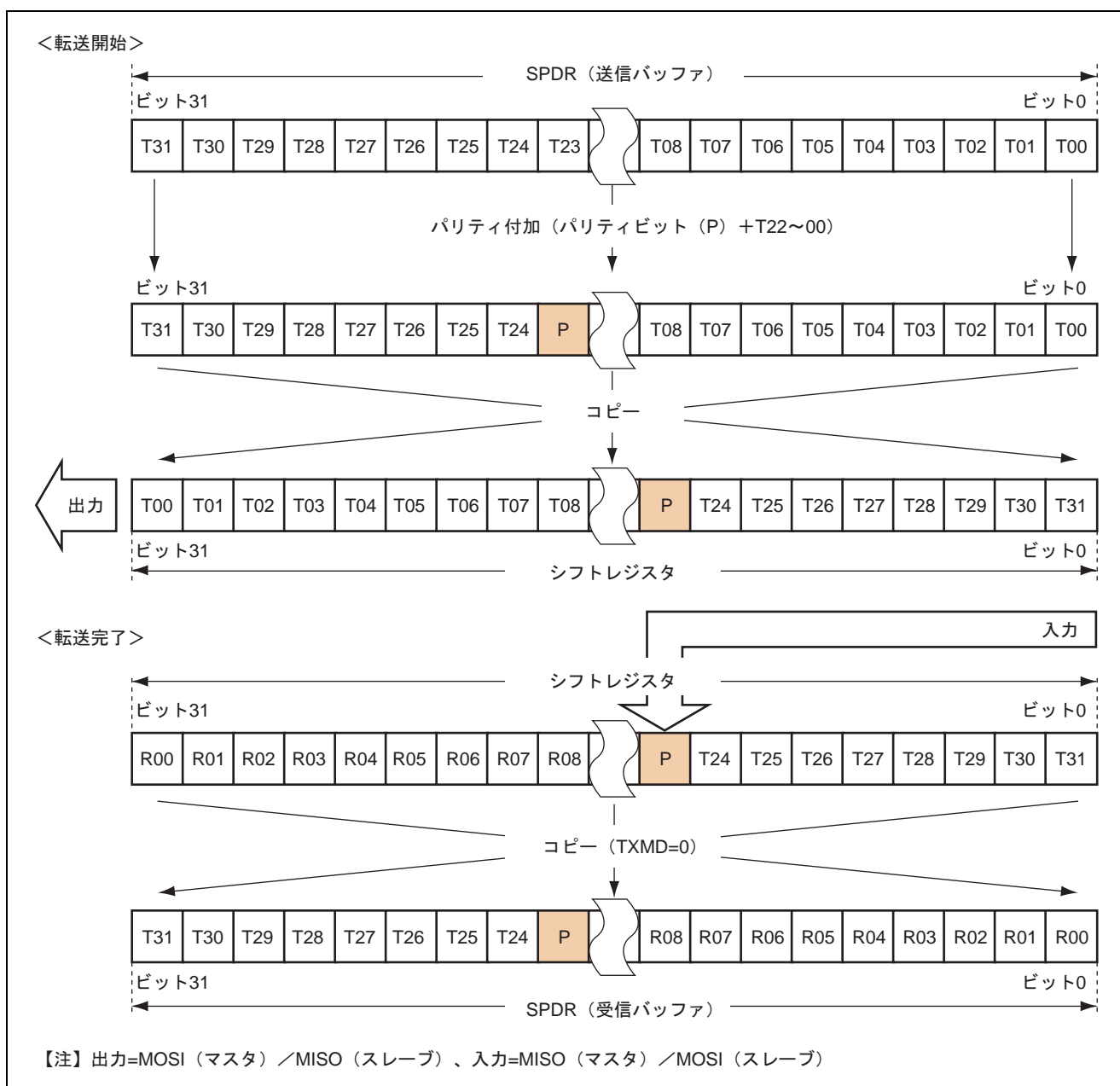


図 15.20 LSB ファースト (24 ビットデータ / パリティ機能有効)

15.4.6 通信動作モード

RSPI 制御レジスタ (SPCR) の通信動作モード選択ビット (TXMD) の設定により、全二重同期式シリアル通信、送信のみの動作を選択します。図 15.21、図 15.22 に記載した SPDR アクセスは、RSPI データレジスタ (SPDR) へのアクセス状況を示しています。I はアイドルサイクル、W は書き込みサイクルを示しています。

(1) 全二重同期式シリアル通信 (TXMD=0)

図 15.21 に、RSPI 制御レジスタ (SPCR) の通信動作モード選択ビット (TXMD) を 0 に設定した場合の動作例を示します。図 15.21 の例では、RSPI データコントロールレジスタ (SPDCR) の SPFC1、0 が 00、RSPI コマンドレジスタ (SPCMD) の CPHA が 1、CPOL が 0 の設定で、RSPI が 8 ビットのシリアル転送を実行しています。RSPCK 波形の下に記載した数字は RSPCK サイクル数 (= 転送ビット数) を示しています。

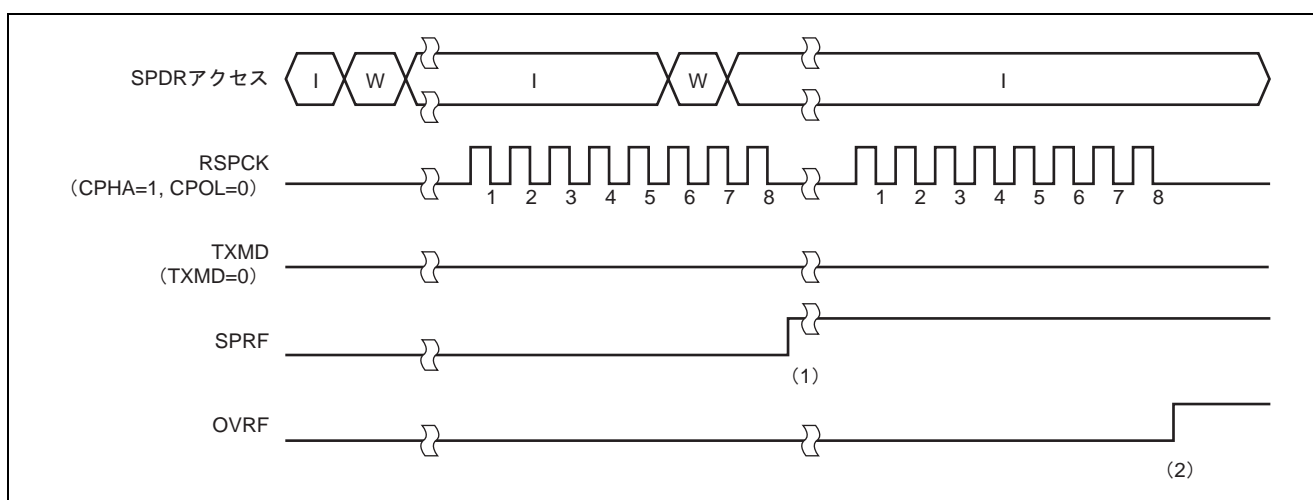


図 15.21 TXMD=0 の動作例

以下に、図中の (1)、(2) に示したタイミングでのフラグの動作内容を説明します。

1. SPDRの受信バッファが空の状態ではシリアル転送が終了すると、RSPIはSPRFを1にしてシフトレジスタの受信データを受信バッファにコピーします。
2. SPDRの受信バッファに以前の受信データがある状態でシリアル転送が終了すると、RSPIはOVRFを1にしてシフトレジスタの受信データを破棄します。

全二重同期式シリアル通信(TXMD=0)時は、送信データを送信し、受信データを受信します。そのため、SPRF、OVRFは、(1)、(2)それぞれのタイミングで1になります。

(2) 送信のみ動作 (TXMD=1)

図 15.22 に、RSPI 制御レジスタ (SPCR) の通信動作モード選択ビット (TXMD) を 1 に設定した場合の動作例を示します。図 15.22 の例では、RSPI データコントロールレジスタ (SPDCR) の SPFC1、0 が 00、RSPI コマンドレジスタ (SPCMD) の CPHA が 1、CPOL が 0 の設定で、RSPI が 8 ビットのシリアル転送を実行しています。RSPCK 波形の下に記載した数字は RSPCK サイクル数 (=転送ビット数) を示しています。

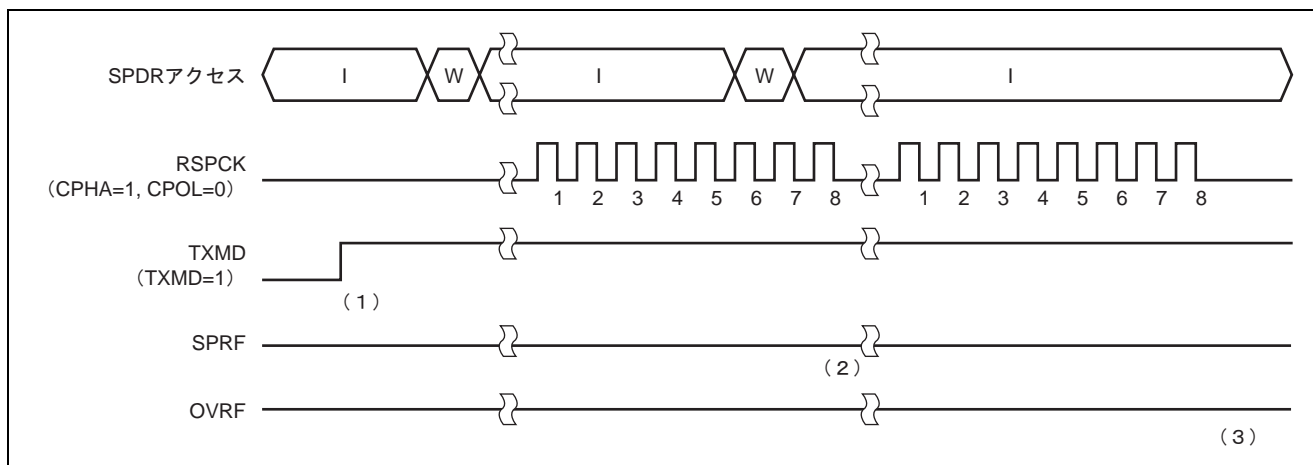


図 15.22 TXMD=1 の動作例

以下に、図中の (1) ~ (3) に示したタイミングでのフラグの動作内容を説明します。

1. 送信のみ動作 (TXMD=1) への遷移は、SPRF、OVRFが0であることを確認してから行ってください。
2. SPDRの受信バッファが空の状態ではシリアル転送が終了すると、送信のみ動作 (TXMD=1) の時は、SPRFは0を保持し、シフトレジスタのデータを受信バッファへコピーしません。
3. SPDRの受信バッファに以前の受信データは存在しないため、シリアル転送が終了しても、OVRFは0を保持し、シフトレジスタのデータを受信バッファへコピーしません。

送信のみ動作 (TXMD=1) の時は、送信データを送信し、受信データを受信しません。そのため、SPRF、OVRFは、(1)、(2)、(3)それぞれのタイミングで0を保持します。

15.4.7 送信バッファエンpty/受信バッファフルフラグ

図 15.23 に RSPI ステータスレジスタ (SPSR) の RSPI 送信バッファエンptyフラグ (SPTEF) と RSPI 受信バッファフルフラグ (SPRF) の動作例を示します。図 15.23 に記載した SPDR アクセスは、RSPI データレジスタ (SPDR) へのアクセス状況を示しています。I はアイドルサイクル、W は書き込みサイクル、R は読み出しサイクルを示しています。図 15.23 の例では、RSPI 制御レジスタ (SPCR) の TXMD が 0、RSPI データコントロールレジスタ (SPDCR) の SPFC1、0 が 00、RSPI コマンドレジスタ (SPCMD) の CPHA が 1、CPOL が 0 の設定で、RSPI が 8 ビットのシリアル転送を実行しています。RSPCK 波形の下に記載した数字は RSPCK サイクル数 (=転送ビット数) を示しています。

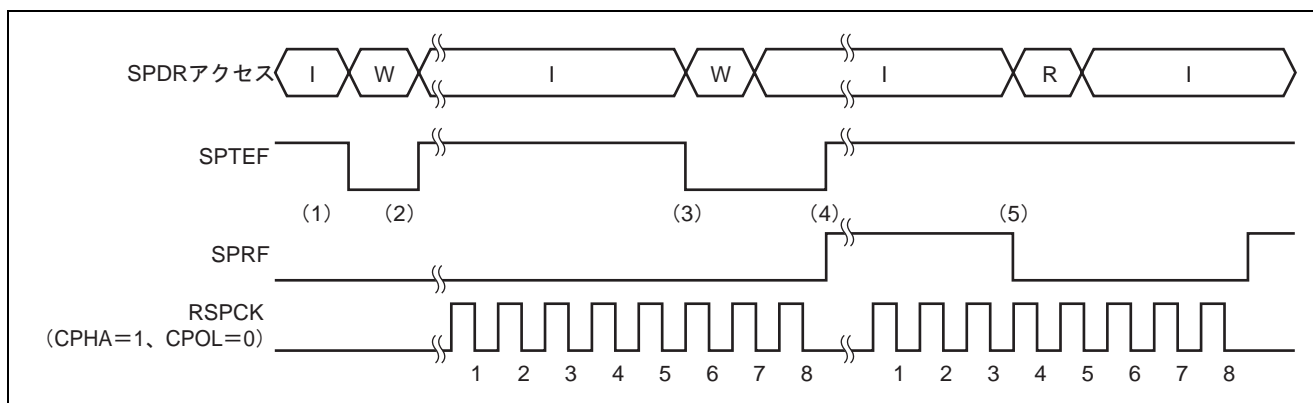


図 15.23 SPTEF、SPRF ビットの動作例

以下に、図中の (1) ~ (5) に示したタイミングでのフラグの動作内容を説明します。

1. SPDRの送信バッファが空の状態、SPDRに送信データを書き込むと、RSPIはSPTEFビットを0にして送信バッファにデータを書き込みます。SPRFフラグは変化しません。
2. シフトレジスタが空の場合には、RSPIはSPTEFビットを1にして送信バッファのデータをシフトレジスタにコピーします。SPRFフラグは変化しません。なお、シリアル転送の開始方法は、RSPIのモードに依存します。詳細は、「15.4.10 SPI動作」および「15.4.11 クロック同期式動作」を参照してください。
3. SPDRの送信バッファが空の状態、SPDRに送信データを書き込むと、RSPIがSPTEFビットを0にして送信バッファにデータを書き込みます。SPRFフラグは変化しません。シフトレジスタにはシリアル転送中のデータが格納されているため、RSPIは送信バッファのデータをシフトレジスタにコピーしません。
4. SPDRの受信バッファが空の状態、シリアル転送が終了すると、RSPIはSPRFを1にしてシフトレジスタの受信データを受信バッファにコピーします。また、シリアル転送が終了するとシフトレジスタが空になるため、シリアル転送が終了する前に送信バッファがフルであった場合には、RSPIがSPTEFビットを1にして送信バッファのデータをシフトレジスタにコピーします。なお、オーバランエラー発生状態で、シフトレジスタから受信バッファへ受信データをコピーしなかった場合でも、シリアル転送が終了するとRSPIはシフトレジスタを空であると判定し、送信バッファからシフトレジスタへのデータ転送は可能な状態になります。
5. 受信バッファフルの状態、SPDRを読み出すと、RSPIはSPRFを0にして、受信バッファのデータをチップ内部バスに送出します。

SPTEF ビットが 0 の状態で SPDR へ書き込んだ場合には、RSPI は送信バッファのデータを更新しません。SPDR を書き込む場合には、必ず SPTEF が 1 であることを確認してください。SPTEF が 1 であることは、SPSR の読み出し、あるいは RSPI 送信割り込みの利用によって確認できます。RSPI 送信割り込みを利用する場合には、SPCR の SPTIE ビットを 1 にしてください。

RSPI ディスエーブル (SPCR の SPE ビットが 0) の場合には、SPTEF ビットが 1 に初期化されます。このため RSPI ディスエーブル状態で SPCR の SPTIE ビットを 1 にすると、RSPI 送信割り込みが発生します。

SPRF ビットが 1 の状態で、シリアル転送が終了した場合には、RSPI はシフトレジスタから受信バッファへのデータのコピーを行わず、オーバランエラーを検出します (「15.4.8 エラー検出」を参照)。受信データのオーバランを防ぐためには、シリアル転送の終了よりも前に SPRF ビットを 0 にしてください。SPRF が 1 であることは、SPSR の読み出し、あるいは RSPI 受信割り込みの利用によって確認できます。RSPI 受信割り込みを利用する場合には、SPCR の SPRIE ビットを 1 にしてください。

15.4.8 エラー検出

通常の RSPI のシリアル転送では、RSPI データレジスタ (SPDR) の送信バッファに書き込んだデータがシリアル送信され、シリアル受信されたデータを SPDR の受信バッファから読み出すことができます。SPDR へアクセスした場合の送受信バッファの状態やシリアル転送の開始/終了時の RSPI の状態によっては、通常以外の転送が実行される場合があります。

一部の通常以外の転送動作が発生した場合には、RSPI はオーバランエラー、パリティエラーまたはモードフォルトエラーとして検出します。表 15.8 に、通常以外の転送動作と RSPI のエラー検出機能の関係を示します。

表 15.8 通常以外の転送の発生条件と RSPI のエラー検出機能

	発生条件	RSPI 動作	エラー検出
A	送信バッファフルの状態で SPDR を書き込み。	送信バッファ内容を保持。 書き込みデータ欠落。	なし
B	スレーブモードで送信データをシフトレジスタにセットしていない状態で、シリアル転送開始。	前回シリアル転送時の受信データをシリアル送信。	なし
C	受信バッファエンプティの状態で SPDR を読み出し。	前回シリアル受信データを出力。	なし
D	受信バッファフルの状態で、シリアル転送が終了。	受信バッファ内容を保持。 シリアル受信データ欠落。	オーバランエラー検出
E	全二重同期式シリアル通信時にパリティ機能が有効な状態で誤ったパリティビットを受信。	パリティエラーフラグのアサート。	パリティエラー検出
F	マルチマスタモードでシリアル転送アイドル時に SSL0 入力信号アサート。	RSPCK、MOSI、SSL1~3 出力信号のドライブ停止。 RSPI ディスエーブル。	モードフォルトエラー検出

	発生条件	RSPI 動作	エラー検出
G	マルチマスタモードでシリアル転送中に SSL0 入力信号アサート。	シリアル転送を中断。 送受信データ欠落。 RSPCK、MOSI、SSL1~3 出力信号のドライブ停止。 RSPI ディスエーブル。	モードフォルトエラー検出
H	スレーブモードでシリアル転送中に SSL0 入力信号がネゲート。	シリアル転送中断。 送受信データ欠落。 MISO 出力信号のドライブ停止。 RSPI ディスエーブル。	モードフォルトエラー検出

表 15.8 の A に示した動作に対しては、RSPI はエラーを検出しません。SPDR への書き込み時にデータを欠落させないために、必ず RSPI ステータスレジスタ (SPSR) の SPTEF ビットが 1 の状態で SPDR への書き込みを実施してください。

B に示した動作に対しても、RSPI はエラーを検出しません。RSPI では、シフトレジスタの更新前に起動されたシリアル転送において、前回シリアル転送時の受信データを送信し、B に示した動作をエラーとして扱いません。なお、前回シリアル転送時の受信データは SPDR の受信バッファに保持されているので、正しく読み出されます (シリアル転送が終了する前に SPDR を読み出さないと、オーバランエラーが発生します)。

C に示した動作に対しても、RSPI はエラーを検出しません。不必要なデータを読み出さないようにするためには、SPSR の SPRF ビットが 1 の状態で SPDR の読み出しを実行するようにしてください。

D に示したオーバランエラーについては、「15.4.8 (1) オーバランエラー」で、E に示したパリティエラーについては、「15.4.8 (2) パリティエラー」で詳しく説明します。また、F~H に示したモードフォルトエラーについては、「15.4.8 (3) モードフォルトエラー」で説明します。なお、SPSR の SPTEF ビットと SPRF ビットの動作については、「15.4.7 送信バッファエンプティ/受信バッファフルフラグ」を参照してください。

(1) オーバランエラー

RSPI データレジスタ (SPDR) の受信バッファフル状態でシリアル転送が終了すると、RSPI はオーバランエラーを検出して SPSR の OVRF ビットを 1 にします。OVRF ビットが 1 の状態では、RSPI はシフトレジスタのデータを受信バッファにコピーしないので、受信バッファにはエラー発生前のデータが保持されます。SPSR の OVRF ビットを 0 にするためには、システムリセットを実施するか、OVRF ビットが 1 にセットされた状態の SPSR を CPU が読み出した後に、OVRF に 0 を書き込む必要があります。

図 15.24 に、SPSR の SPRF ビットと OVRF ビットの動作を示します。図 15.24 に記載した SPSR アクセスと SPDR アクセスは、それぞれのアクセス状況を示しています。I はアイドル状態、W は書き込みサイクル、R は読み出しサイクルを示しています。図 15.24 の例では、RSPI コマンドレジスタ (SPCMD) の CPHA が 1、CPOL が 0 の設定で、RSPI が 8 ビットのシリアル転送を実行しています。RSPCK 波形の下に記載した数字は RSPCK サイクル数 (= 転送ビット数) を示しています。

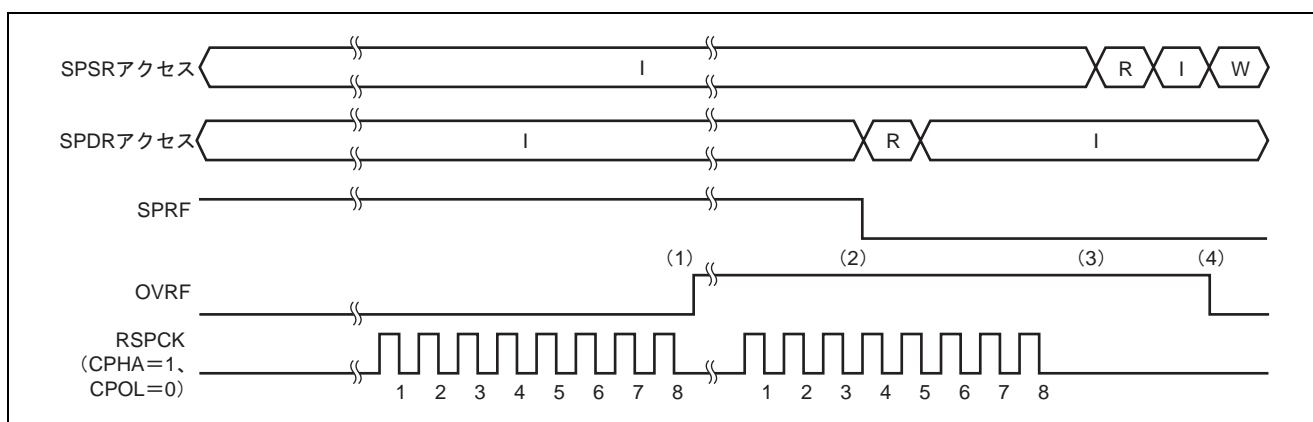


図 15.24 SPRF、OVRF ビットの動作例

以下に、図中の (1) ~ (4) に示したタイミングでのフラグの動作内容を説明します。

1. SPRF ビットが 1 の状態 (受信バッファフル) でシリアル転送が終了すると、RSPI がオーバランエラーを検出し、OVRF ビットを 1 にします。RSPI はシフトレジスタのデータを受信バッファにコピーしません。また、SPPE が 1 であっても、パリティエラーの検出は行いません。マスタモードの場合には、RSPI は RSPI シーケンスステータスレジスタ (SPSSR) の SPEC2~0 ビットに、RSPI コマンドレジスタ (SPCMD) に対するポインタの値をコピーします。
2. SPDR を読み出すと、RSPI は SPRF ビットを 0 にして受信バッファのデータを内部バスに出力します。受信バッファが空になっても、OVRF ビットはクリアされません。
3. OVRF ビットが 1 の状態 (オーバランエラー) でシリアル転送が終了した場合には、RSPI は SPRF ビットを 0 のまま更新しません。また、RSPI はシフトレジスタのデータを受信バッファにコピーしません。また、SPPE が 1 であっても、パリティエラーの検出は行いません。マスタモードの RSPI の場合に、RSPI は SPSSR の SPEC2~0 ビットを更新しません。オーバランエラー発生状態で、シフトレジスタから受信バッファへ受信データをコピーしなかった場合でも、シリアル転送が終了すると RSPI はシフトレジスタを空であると判定し、送信バッファからシフトレジスタへのデータ転送は可能な状態になります。
4. OVRF ビットが 1 の状態で CPU が SPSR を読み出した後、CPU が OVRF に 0 を書き込むと、RSPI は OVRF ビットをクリアします。

オーバーランの発生は、SPSR の読み出しあるいは RSPI エラー割り込みと SPSR の読み出しによって確認できます。RSPI エラー割り込みを利用する場合には、RSPI 制御レジスタ (SPCR) の SPEIE ビットを 1 にしてください。RSPI エラー割り込みを利用せずにシリアル転送を実行する場合には、SPDR の読み出し直後に SPSR を読み出すなどの方法で、オーバーランエラー発生を早期に検出できるように対処してください。RSPI をマスターモードで使用する場合、SPSSR の SPECM1、0 ビットを読み出すことで、エラー発生時の SPCMD に対するポインタ値を確認できます。

オーバーランエラーが発生して OVRF ビットが 1 になると、OVRF ビットをクリアするまで正常な受信動作ができなくなります。OVRF ビットを 0 にクリアする条件は以下のとおりです。

- OVRFが1にされた状態のSPSRをCPUが読み出した後、CPUがOVRFに0を書き込む
- システムリセット

(2) パリティエラー

RSPI 制御レジスタ (SPCR) の TXMD ビットが 0、RSPI 制御レジスタ 2 (SPCR2) の SPPE ビットが 1 の状態で全二重同期式シリアル通信を行い、転送が終了すると、パリティエラーの判定を行います。RSPI は、受信データにパリティエラーを検出すると、RSPI ステータスレジスタ (SPSR) の PERF ビットを 1 にします。OVRF ビットが 1 の状態では、RSPI はシフトレジスタのデータを受信バッファにコピーしないので、受信データに対するパリティエラーの検出は行いません。SPSR の PERF ビットを 0 にするためには、システムリセットを実施するか、PERF ビットが 1 にセットされた状態の SPSR を CPU が読み出した後に、PERF に 0 を書き込む必要があります。

図 15.25 に、SPSR の OVRF ビットと PERF ビットの動作を示します。図 15.25 に記載した SPSR アクセスは、SPSR へのアクセス状況を示しています。I はアイドル状態、W は書き込みサイクルを示しています。図 15.25 の例では、RSPI 制御レジスタ (SPCR) の TXMD ビットが 0、RSPI 制御レジスタ 2 (SPCR2) の SPPE ビットが 1 の状態で全二重同期式シリアル通信を行います。RSPI コマンドレジスタ (SPCMD) の CPHA が 1、CPOL が 0 の設定で、RSPI が 8 ビットのシリアル転送を実行しています。RSPCK 波形の下に記載した数字は RSPCK サイクル数 (=転送ビット数) を示しています。

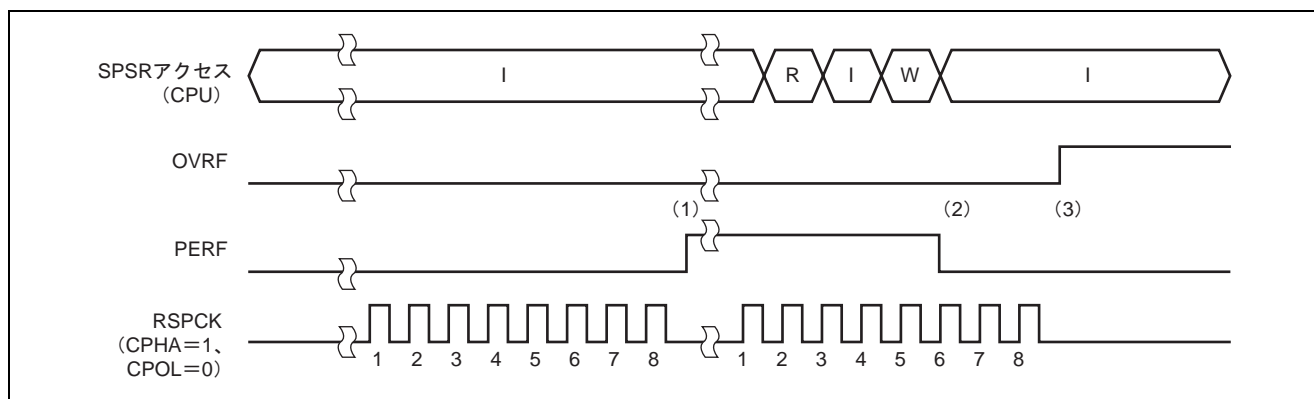


図 15.25 PERF ビットの動作例

以下に、図中の (1) ~ (3) に示したタイミングでのフラグの動作内容を説明します。

1. RSPIがオーバランエラーを検出せず、シリアル転送が終了すると、シフトレジスタのデータを受信バッファにコピーします。この時、RSPIが受信データを判定し、パリティエラーを検出するとPERFビットを1にします。マスタモードの場合には、RSPIはRSPIシーケンスステータスレジスタ (SPSSR) のSPECM2~0ビットに、RSPIコマンドレジスタ (SPCMD) に対するポインタの値をコピーします。
2. PERFビットが1の状態CPUがSPSRを読み出した後、CPUがPERFに0を書き込むと、RSPIはPERFビットをクリアします。
3. RSPIがオーバランエラーを検出し、シリアル転送が終了すると、シフトレジスタのデータを受信バッファにコピーしません。この時、RSPIはパリティエラーの検出行いません。

パリティエラーの発生は、SPSR の読み出し、あるいは RSPI エラー割り込みと SPSR の読み出しによって確認できます。RSPI エラー割り込みを利用する場合には、RSPI 制御レジスタ (SPCR) の SPEIE ビットを 1 にしてください。RSPI エラー割り込みを利用せずにシリアル転送を実行する場合には、SPSR を読み出すなどの方法で、パリティエラー発生を早期に検出できるように対処してください。RSPI をマスタモードで使用する場合、SPSSR の SPECM2~0 ビットを読み出すことで、エラー発生時の SPCMD に対するポインタ値を確認できます。

PERF ビットを 0 にクリアする条件は、以下の通りです。

- PERFが1にされた状態のSPSRをCPUが読み出した後、CPUがPERFに0を書き込む。
- システムリセット。

(3) モードフォルトエラー

RSPI 制御レジスタ (SPCR) の MSTR ビットが 1、SPMS ビットが 0、MODFEN ビットが 1 の場合には、RSPI はマルチマスタモードで動作します。マルチマスタモードの RSPI の SSL0 入力信号に対してアクティブレベルが入力されると、シリアル転送状態にかかわらず、RSPI はモードフォルトエラーを検出して RSPI ステータスレジスタ (SPSR) の MODF ビットを 1 にします。モードフォルトエラーを検出すると、RSPI は RSPI シーケンスステータスレジスタ (SPSSR) の SPECM1、0 ビットに、RSPI コマンドレジスタ (SPCMD) に対するポインタの値をコピーします。なお、SSL0 信号のアクティブレベルは、RSPI スレーブセレクト極性レジスタ (SSLP) の SSL0P ビットによって決定されます。

MSTR ビットが 0 の場合には、RSPI はスレーブモードで動作します。スレーブモードの RSPI の MODFEN ビットが 1、SPMS ビットが 0 の場合、シリアル転送期間 (有効データのドライブ開始から最終有効データの取り込みまで) に SSL0 入力信号がネゲートされると、RSPI はモードフォルトエラーを検出します。

RSPI はモードフォルトエラーを検出すると、出力信号のドライブ停止および SPCR レジスタの SPE ビットのクリアを実施します (「15.4.9 RSPI の初期化」を参照)。マルチマスタ構成の場合には、モードフォルトエラーを利用して出力信号のドライブと RSPI 機能を停止させ、マスタ権の解放を実現することが可能です。

モードフォルトエラーの発生は、SPSR の読み出し、あるいは RSPI エラー割り込みと SPSR の読み出しによって確認できます。RSPI エラー割り込みを利用する場合には、RSPI 制御レジスタ (SPCR) の SPEIE ビットを 1 にしてください。RSPI エラー割り込みを利用せずにモードフォルトエラーを検出するためには、SPSR をポーリングする必要があります。RSPI をマスタモードで使用する場合、SPSSR の SPECM1、0 ビットを読み出すことで、エラー発生時の SPCMD に対するポインタ値を確認できます。

MODF ビットが 1 の状態では、RSPI は CPU による SPE ビットへの 1 の書き込みを無視します。モードフォルトエラー検出後に RSPI 機能を有効にするためには、必ず MODF ビットを 0 にしてください。MODF ビットを 0 にクリアする条件は以下のとおりです。

- MODFが1にされた状態のSPSRをCPUが読み出した後、CPUがMODFに0を書き込む
- システムリセット

15.4.9 RSPI の初期化

CPU が RSPI 制御レジスタ (SPCR) の SPE ビットに 0 を書き込んだ場合、またはモードフォルトエラー検出により RSPI が SPE ビットを 0 にクリアした場合には、RSPI は RSPI 機能を無効化し、モジュール機能の一部を初期化します。また、システムリセットが発生した場合には、RSPI はモジュール機能をすべて初期化します。以下に、SPE ビットのクリアによる初期化とシステムリセットによる初期化について説明します。

(1) SPE ビットのクリアによる初期化

SPCR の SPE ビットがクリアされた場合には、RSPI は以下に示す初期化を実施します。

- 実行中のシリアル転送を中断
- スレーブモードの場合、出力信号のドライブ停止 (Hi-Z)
- RSPI 内部ステートの初期化
- RSPI ステータスレジスタ (SPSR) の SPTEF ビットの初期化

SPE ビットのクリアによる初期化では、RSPI の制御ビットは初期化されません。このため、CPU が SPE ビットに 1 を再設定すれば初期化前と同じ転送モードで RSPI を起動できます。

SPSR の SPRF ビット、OVRF ビット、MODF ビットの値は初期化されません。また、RSPI シーケンスステータスレジスタ (SPSSR) の値も初期化されません。このため、RSPI の初期化後も受信バッファのデータの読み出し、RSPI 転送時のエラー発生状況の確認が可能です。

SPSR の SPTEF ビットの値は、1 に初期化されます。このため、RSPI 初期化後に SPCR の SPTIE ビットが 1 に設定されていると、RSPI 送信割り込みが発生します。CPU で RSPI を初期化する場合に、RSPI 送信割り込みを禁止するためには、SPE ビットへの 0 書き込みと同時に SPTIE ビットにも 0 を書き込んでください。モードフォルトエラー検出後の RSPI 送信割り込みを禁止するためには、エラー処理ルーチンで SPTIE ビットに 0 を書き込んでください。

(2) システムリセット

システムリセットによる初期化では、「15.4.9 (1) SPE ビットのクリアによる初期化」に記載の事項に加え、RSPI 制御用の全ビットの初期化、ステータスビットの初期化、データレジスタの初期化が実施され、RSPI が完全に初期化されます。

15.4.10 SPI 動作

(1) マスタモード動作

シングルマスタモード動作とマルチマスタモード動作の違いは、モードフォルトエラー検出（「15.4.8 エラー検出」を参照）のみです。シングルマスタモードの RSPI ではモードフォルトエラーを検出しません。マルチマスタモードの RSPI ではモードフォルトエラーを検出します。本節では、シングル/マルチマスタモードで共通する動作について説明します。

(a) シリアル転送の開始

RSPI ステータスレジスタ (SPSR) の SPTEF ビットが 1 の状態で、RSPI データレジスタ (SPDR) へデータを書き込むと、RSPI は SPDR の送信バッファのデータを更新します。SPDR への書き込み、または CPU から SPTEF の 1 を読み出した後の 0 の書き込みによって SPTEF ビットを 0 にクリアした状態で、シフトレジスタが空の場合には、RSPI は送信バッファのデータをシフトレジスタにコピーしてシリアル転送を開始します。RSPI は、シフトレジスタに送信データをコピーするとシフトレジスタのステータスをフルに変更し、シリアル転送が終了するとシフトレジスタのステータスを空に変更します。シフトレジスタのステータスを CPU から参照することはできません。

なお、RSPI の転送フォーマットの詳細については「15.4.4 転送フォーマット」を参照してください。

(b) シリアル転送の終了

RSPI コマンドレジスタ (SPCMD) の CPHA ビットにかかわらず、RSPI は最終サンプリングタイミングに対応する RSPCK エッジを送出するとシリアル転送を終了します。RSPI ステータスレジスタ (SPSR) の SPRF ビットが 0 で受信バッファに空きがある場合には、シリアル転送終了後にシフトレジスタから RSPI データレジスタ (SPDR) の受信バッファにデータをコピーします。

なお、最終サンプリングタイミングは転送データのビット長に依存して変化します。マスタモードの RSPI のデータ長は、RSPI コマンドレジスタ (SPCMD) の SPB3~SPB0 ビットの設定値に依存します。RSPI の転送フォーマットの詳細については「15.4.4 転送フォーマット」を参照してください。

(c) シーケンス制御

マスタモード時の転送フォーマットは、RSPI シーケンス制御レジスタ (SPSCR)、RSPI コマンドレジスタ 0~7 (SPCMD0~7)、RSPI ビットレートレジスタ (SPBR)、RSPI クロック遅延値レジスタ (SPCKD)、RSPI スレーブセレクトネゲート遅延レジスタ (SSLND)、RSPI 次アクセス遅延値レジスタ (SPND) によって決定されます。

SPSCR は、マスタモードの RSPI で実行するシリアル転送のシーケンス構成を決定するためのレジスタです。SPCMD0~7 には、SSL 出力信号値、MSB/LSB ファースト、データ長、ビットレート設定の一部、RSPCK 極性/位相、SPCKD の参照要否、SSLND の参照要否、SPND の参照要否が設定されています。SPBR にはビットレート設定の一部、SPCKD には RSPI クロック遅延値、SSLND には SSL ネゲート遅延、SPND には次アクセス遅延値が設定されています。

RSPI は、SPSCR に設定されたシーケンス長に従って、SPCMD0~7 の一部/全部からなるシーケンスを構成します。RSPI には、シーケンスを構成している SPCMD に対するポインタが存在します。このポインタの値は、RSPI シーケンスステータスレジスタ (SPSSR) の SPCP2~0 ビットの読み出しによって CPU から確認可能です。RSPI 制御レジスタ (SPCR) の SPE ビットを 1 にして RSPI 機能をイネーブルにすると、RSPI はコマンドに対するポインタを SPCMD0 にセットし、シリアル転送の開始時に SPCMD0 の設定内容を転送フォーマットに反映します。RSPI は、各データ転送の次アクセス遅延期間が終了するたびにポインタをインクリメントします。シーケンスを構成している最終コマンドに対応するシリアル転送が終了すると、RSPI はポインタを SPCMD0 にセットするので、シーケンスは繰り返し実行されます。

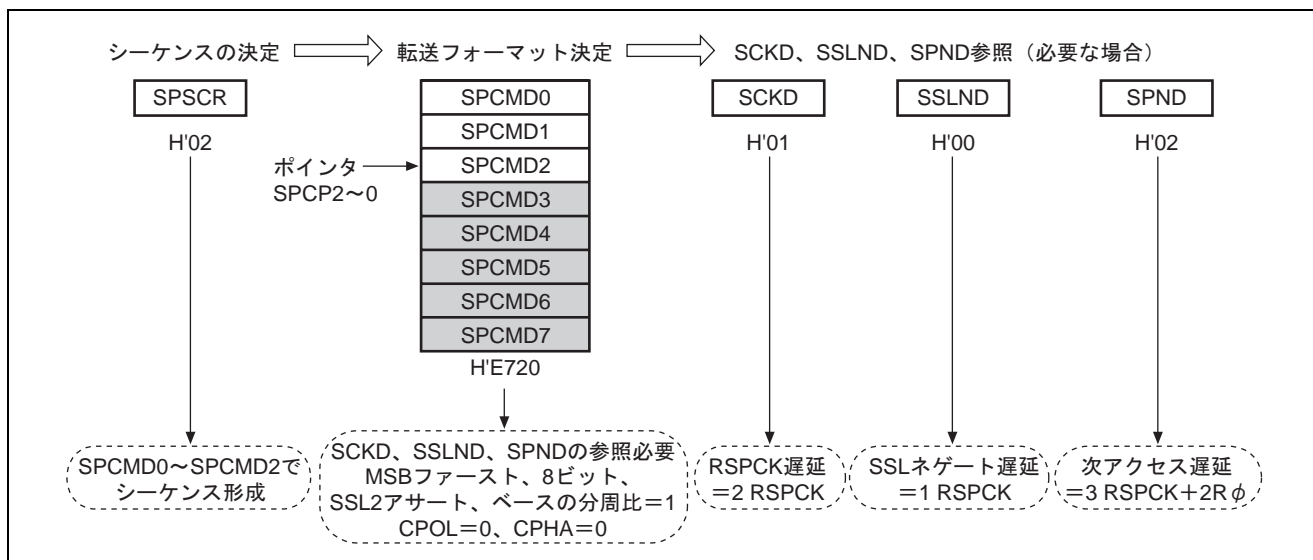


図 15.26 マスタモードでのシリアル転送方式の決定方法

(d) バースト転送

RSPI が現在のシリアル転送で参照している RSPI コマンドレジスタ (SPCMD) の SSLKP ビットが 1 の場合には、RSPI はシリアル転送中の SSL 信号レベルを次のシリアル転送の SSL 信号アサート開始まで保持します。次のシリアル転送での SSL 信号レベルが、現在のシリアル転送での SSL 信号レベルと同じであれば、RSPI は SSL 信号アサート状態を保持したまま連続的にシリアル転送を実行することができます (バースト転送)。

図 15.27 に、SPCMD0、1 の設定を使用してバースト転送を実現した場合の SSL 信号動作例を示します。図 15.27 に記載した (1) ~ (7) の RSPI 動作内容について、以下に説明します。なお、SSL 出力信号の極性は、RSPI スレーブセレクト極性レジスタ (SSLP) の設定値に依存します。

1. SPCMD0に従ったSSL信号のアサートとRSPCK遅延の挿入を実施します。
2. SPCMD0に従ったシリアル転送を実行します。
3. SSLネゲート遅延を挿入します。
4. SPCMD0のSSLKPビットが1であるため、SPCMD0でのSSL信号値を保持します。この期間は、最短の場合にはSPCMD0の次アクセス遅延と同じだけ継続されます。最短期間を経過後にシフトレジスタが空の場合には、次転送のための送信データがシフトレジスタに格納されるまで、この期間を継続します。
5. SPCMD1に従ったSSL信号のアサートとRSPCK遅延の挿入を実施します。
6. SPCMD1に従ったシリアル転送を実行します。
7. SPCMD1のSSLKPビットが0であるため、SSL信号をネゲートします。また、SPCMD1に従った次アクセス遅延が挿入されます。

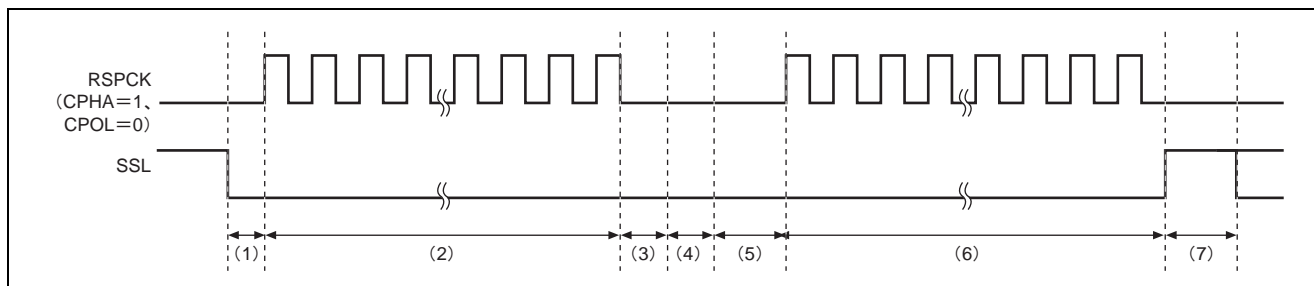


図 15.27 SSLKP ビットを利用したバースト転送動作の例

SSLKP ビットに 1 を設定した SPCMD での SSL 信号出力設定と、次転送で使用する SPCMD での SSL 信号出力設定が異なる場合、RSPI は次転送のコマンドに対応した SSL 信号のアサート時 (図 15.23 の (5)) に SSL 信号状態を切り替えます。このような SSL 信号の切り替えが発生した場合、MISO をドライブするスレーブが競合して信号レベルの衝突が発生する可能性があるので注意してください。

マスタモードの RSPI は、SSLKP を使用しない場合の SSL 信号動作をモジュール内部で参照しています。SPCMD の CPHA ビットが 0 の場合でも、RSPI は内部で検出した次転送の SSL 信号のアサートを使用してシリアル転送を正確に開始できます。このため、マスタモードのバースト転送は、CPHA ビットの設定値にかかわらず実行できます (「15.4.10 (2) スレーブモード動作」を参照)。

(e) RSPCK 遅延 (t1)

マスタモードの RSPI の RSPCK 遅延値は、RSPI コマンドレジスタ (SPCMD) の SCKDEN ビットの設定と RSPI クロック遅延レジスタ (SPCKD) の設定に依存します。RSPI は、シリアル転送で参照する SPCMD をポインタ制御によって決定し、選択した SPCMD の SCKDEN ビットと SPCKD を使用して、表 15.9 のようにシリアル転送時の RSPCK 遅延値を決定します。なお、RSPCK 遅延の定義については、「15.4.4 転送フォーマット」を参照してください。

表 15.9 SCKDEN、SPCKD と RSPCK 遅延値の関係

SCKDEN	SPCKD	RSPCK 遅延値
0	000~111	1RSPCK
1	000	1RSPCK
	001	2RSPCK
	010	3RSPCK
	011	4RSPCK
	100	5RSPCK
	101	6RSPCK
	110	7RSPCK
	111	8RSPCK

(f) SSL ネゲート遅延 (t2)

マスタモードの RSPI の SSL ネゲート遅延値は、RSPI コマンドレジスタ (SPCMD) の SLNDEN ビットの設定と SSL ネゲート遅延レジスタ (SSLND) の設定に依存します。RSPI は、シリアル転送で参照する SPCMD をポインタ制御によって決定し、選択した SPCMD の SLNDEN ビットと SSLND を使用して、表 15.10 のようにシリアル転送時の SSL ネゲート遅延値を決定します。なお、SSL ネゲート遅延の定義については、「15.4.4 転送フォーマット」を参照してください。

表 15.10 SSLND と SSL ネゲート遅延値の関係

SLNDEN	SSLND	SSL ネゲート遅延値
0	000~111	1RSPCK
1	000	1RSPCK
	001	2RSPCK
	010	3RSPCK
	011	4RSPCK
	100	5RSPCK
	101	6RSPCK
	110	7RSPCK
	111	8RSPCK

(g) 次アクセス遅延 (t3)

マスタモードの RSPI の次アクセス遅延は、RSPI コマンドレジスタ (SPCMD) の SPNDEN ビットの設定と RSPI 次アクセス遅延レジスタ (SPND) の設定に依存します。RSPI は、シリアル転送で参照する SPCMD をポインタ制御によって決定し、選択した SPCMD の SPNDEN ビットと SPND を使用して、表 15.11 のようにシリアル転送時の RSPCK 遅延を決定します。なお、次アクセス遅延の定義については、「15.4.4 転送フォーマット」を参照してください。

表 15.11 SPNDEN、SPND と次アクセス遅延値の関係

SPNDEN	SPND	次アクセス遅延値
0	000~111	1RSPCK+2Rφ
1	000	1RSPCK+2Rφ
	001	2RSPCK+2Rφ
	010	3RSPCK+2Rφ
	011	4RSPCK+2Rφ
	100	5RSPCK+2Rφ
	101	6RSPCK+2Rφ
	110	7RSPCK+2Rφ
	111	8RSPCK+2Rφ

(h) 初期化フロー

図 15.28 に、SPI 動作時、RSPI をマスターモードで使用する場合の初期化フローの例を示します。なお、割り込みコントローラ、DMAC、入出力ポートの設定方法については各ブロックの説明を参照してください。

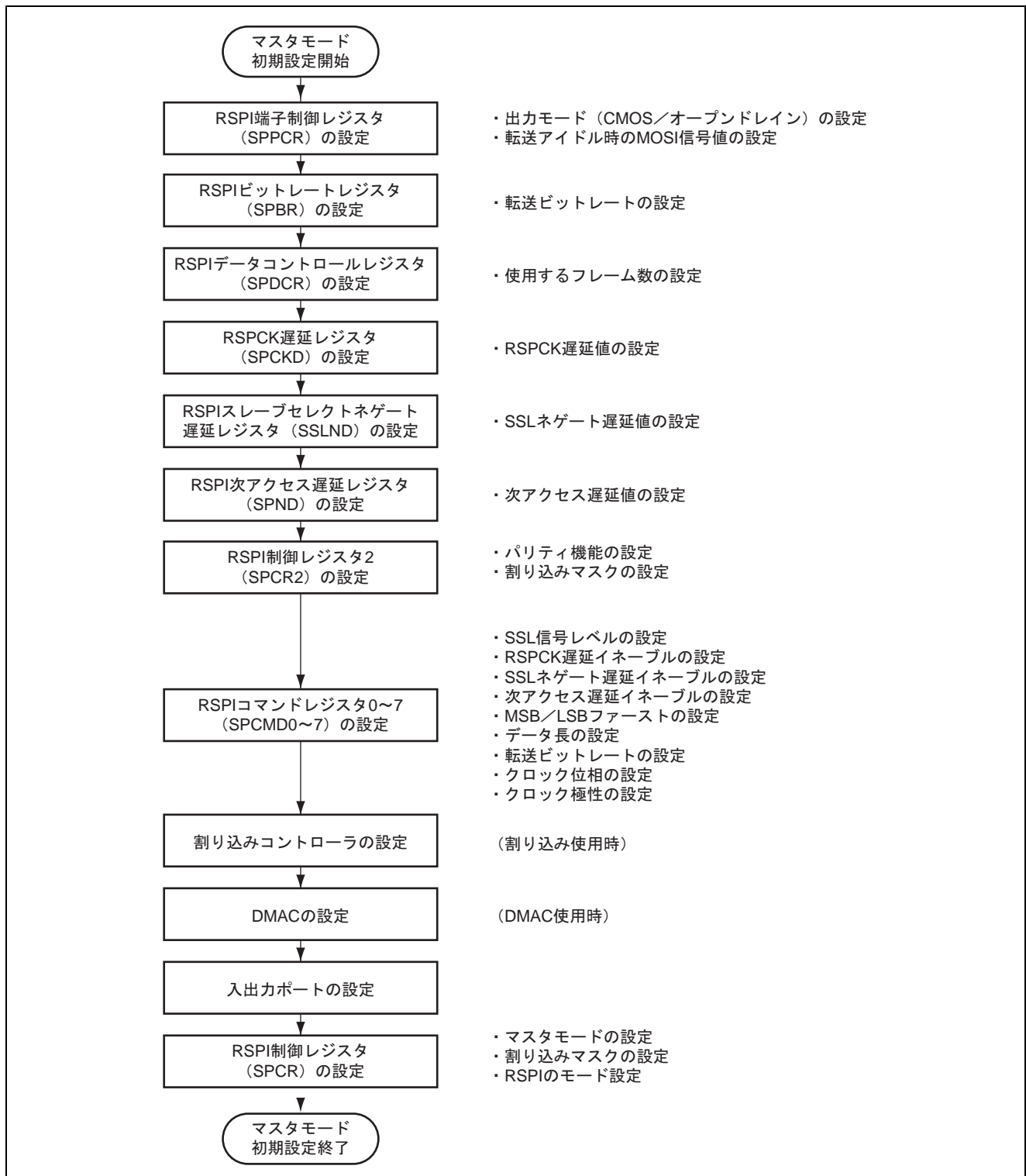


図 15.28 マスターモード時の初期化フロー例

(i) 転送動作フロー

図 15.29 に、SPI 動作時、マスタモードの RSPI の転送動作フローを示します。

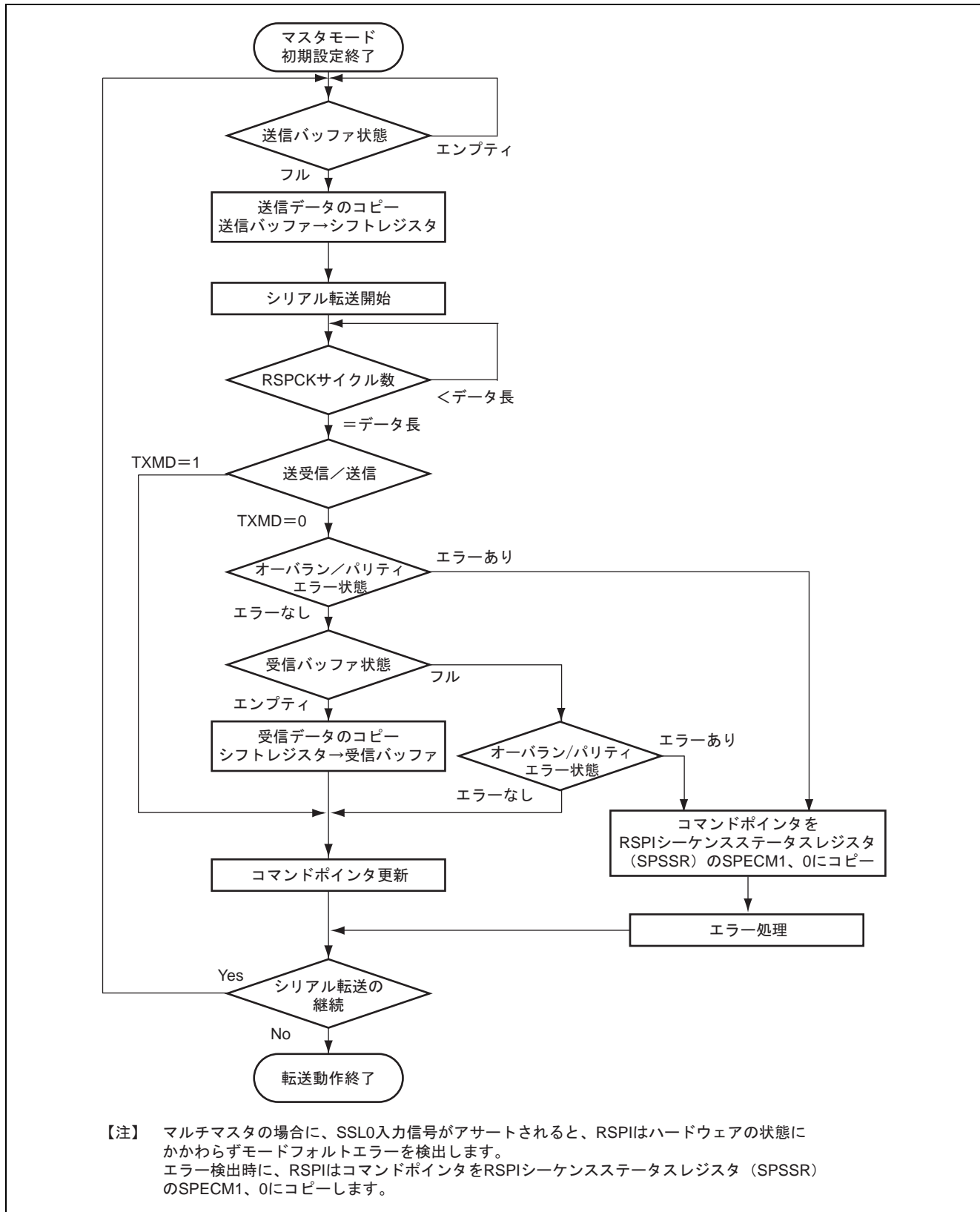


図 15.29 マスタモード時の転送動作フロー

(2) スレーブモード動作

(a) シリアル転送の開始

RSPI コマンドレジスタ 0 (SPCMD0) の CPHA ビットが 0 の場合、RSPI は SSL0 入力信号のアサートを検出すると、MISO 出力信号への有効データのドライブを開始する必要があります。このため、CPHA が 0 の場合には、SSL0 入力信号のアサートがシリアル転送開始のトリガになります。

CPHA ビットが 1 の場合には、RSPI は SSL0 入力信号のアサート状態で最初の RSPCK エッジを検出すると、MISO 出力信号への有効データのドライブを開始する必要があります。このため、CPHA が 1 の場合には、SSL0 信号アサート状態における最初の RSPCK エッジがシリアル転送開始のトリガになります。

RSPI は、シフトレジスタが空の状態ではシリアル転送の開始を検出した場合、シフトレジスタの状態をフルに変更し、シリアル転送中に送信バッファからシフトレジスタにデータがコピーできないようにします。シリアル転送の開始よりも前にシフトレジスタがフルであった場合、RSPI はシフトレジスタの状態をフルのまま変更しません。

CPHA ビットの設定に依存せず、RSPI が MISO 出力信号のドライブを開始するタイミングは、SSL0 信号アサートタイミングです。CPHA ビットの設定によって、RSPI が出力するデータの有効/無効が異なります。

なお、RSPI の転送フォーマットの詳細については、「15.4.4 転送フォーマット」を参照してください。SSL0 入力信号の極性は、RSPI スレーブセレクト極性レジスタ (SSLP) の SSL0P ビットの設定値に依存します。

(b) シリアル転送の終了

RSPI コマンドレジスタ 0 (SPCMD0) の CPHA ビットにかかわらず、RSPI は最終サンプリングタイミングに相当する RSPCK エッジを検出するとシリアル転送を終了します。RSPI ステータスレジスタ (SPSR) の SPRF ビットが 0 で受信バッファに空きがある場合には、シリアル転送の終了後に、RSPI はシフトレジスタから RSPI データレジスタ (SPDR) の受信バッファに受信データをコピーします。また、SPRF ビットの値にかかわらず、RSPI はシリアル転送の終了後にシフトレジスタの状態を空に変更します。シリアル転送開始からシリアル転送終了の間に RSPI が SSL0 入力信号のネゲートを検出するとモードフォルトエラーが発生します（「15.4.8 エラー検出」を参照）。

なお、最終サンプリングタイミングは転送データのビット長に依存して変化します。スレーブモードの RSPI のデータ長は SPCMD0 の SPB3~SPB0 ビットの設定値に依存します。SSL0 入力信号の極性は、RSPI スレーブセレクト極性レジスタ (SSLP) の SSL0P ビットの設定値に依存します。RSPI の転送フォーマットの詳細については、「15.4.4 転送フォーマット」を参照してください。

(c) シングルスレーブ時の注意点

RSPI コマンドレジスタ 0 (SPCMD0) の CPHA ビットが 0 の場合には、RSPI は SSL0 入力信号のアサートエッジを検出するとシリアル転送を開始します。図 15.4 の例に示したような構成で RSPI をシングルスレーブで使用する場合には、SSL0 入力信号が常にアクティブ状態に固定されるため、CPHA を 0 に設定した RSPI ではシリアル転送を正しく開始できません。SSL0 入力信号をアクティブ状態に固定する構成で、スレーブモード RSPI の送受信を正しく実行するためには、CPHA ビットを 1 にしてください。CPHA ビットを 0 にする必要がある場合には、SSL0 入力信号を固定しないでください。

(d) バースト転送

RSPI コマンドレジスタ 0 (SPCMD0) の CPHA ビットが 1 の場合には、SSL0 入力信号のアサート状態を保持したまま連続的なシリアル転送 (バースト転送) を実行することが可能です。CPHA ビットが 1 の場合には、SSL0 入力信号アクティブ状態における最初の RSPCK エッジから最終ビット受信のためのサンプリングタイミングまでが、シリアル転送期間に相当します。SSL0 入力信号がアクティブレベルのままであっても、アクセスの開始を検出可能であるので、バースト転送に対応可能です。

CPHA ビットが 0 の場合には、バースト転送の 2 回目以降のシリアル転送を正しく実行できません。

(e) 初期化フロー

図 15.30 に、SPI 動作時、RSPI をスレーブモードで使用する場合の初期化フローの例を示します。なお、割り込みコントローラ、DMAC、入出力ポートの設定方法については各ブロックの説明を参照してください。

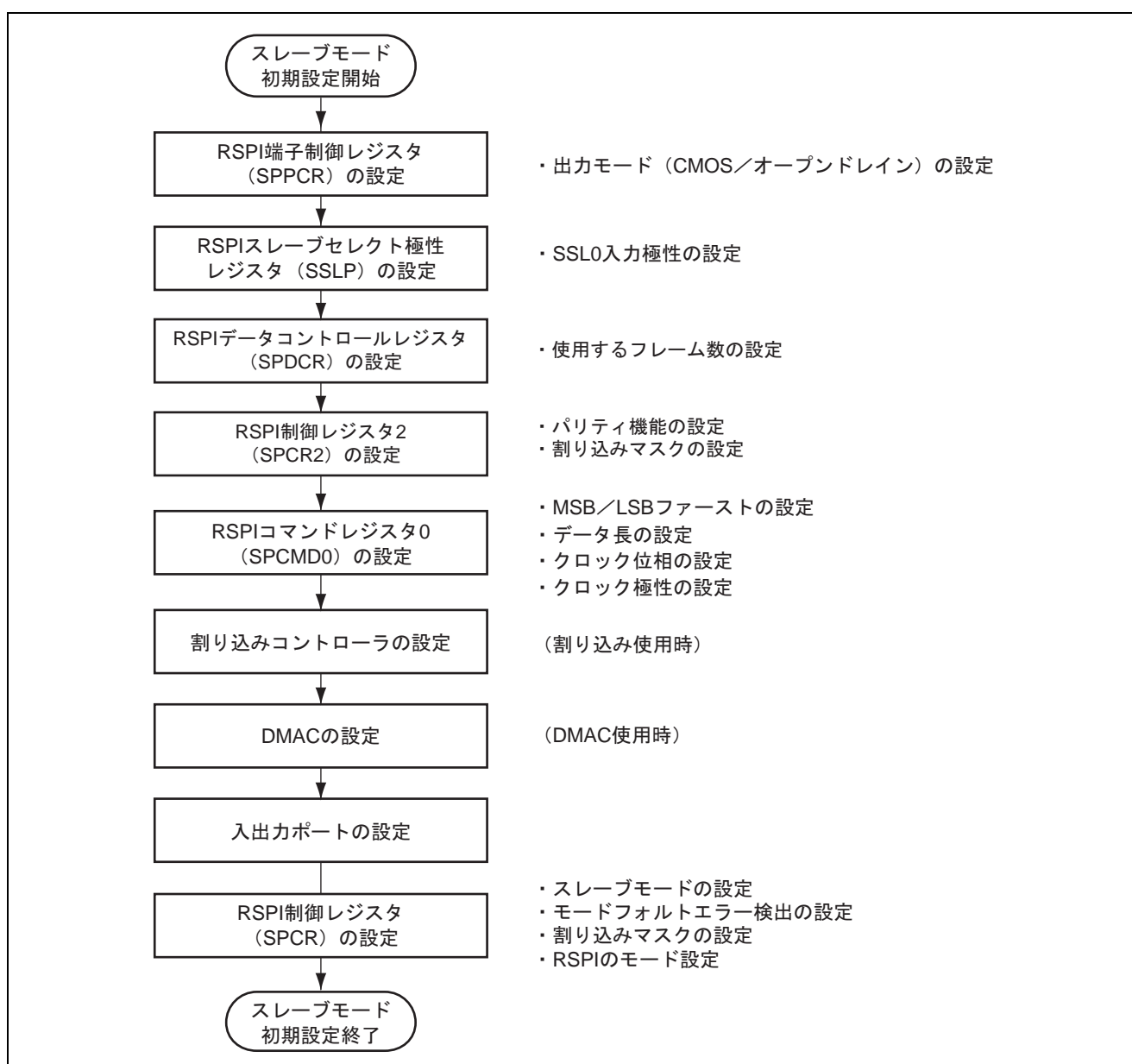


図 15.30 スレーブモード時の初期化フロー例

(f) 転送動作フロー (CPHA=0)

図 15.31 に、SPI 動作時、RSPI コマンドレジスタ 0 (SPCMD0) の CPHA ビットを 0 に設定したスレーブモードの RSPI の転送動作フローを示します。

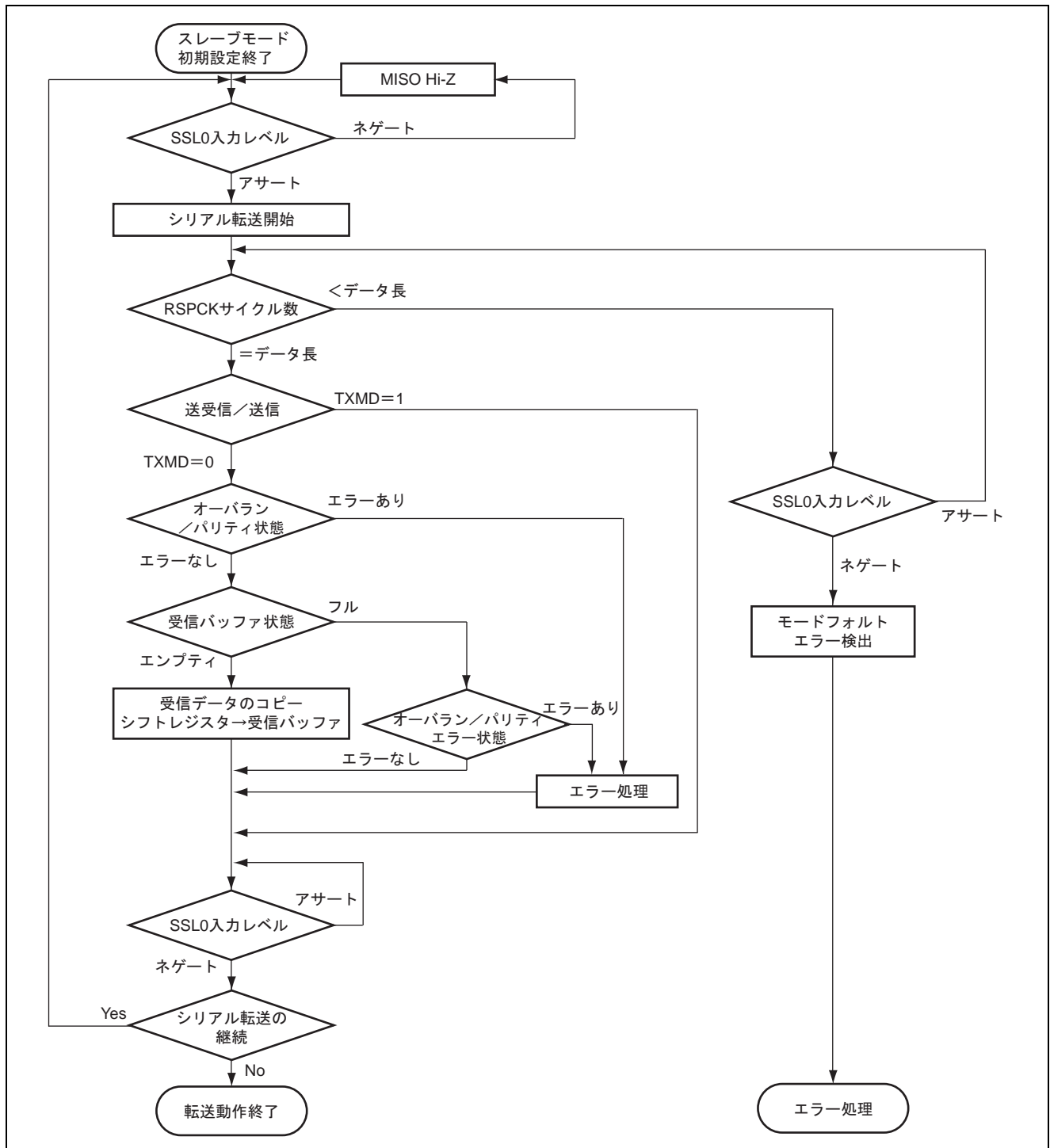


図 15.31 スレーブモード時の転送動作フロー (CPHA=0)

(g) 転送動作フロー (CPHA=1)

図 15.32 に、SPI 動作時、RSPI コマンドレジスタ 0 (SPCMD0) の CPHA ビットを 1、RSPI 制御レジスタ (SPCR) の MODFEN ビットを 1 に設定したスレーブモードの RSPI の転送動作フローを示します。MODFEN ビットを 0 に設定した状態でシリアル転送を開始し、RSPCK サイクル数がデータ長より短い状態で SSLO 入力レベルがネゲートされた場合、以降の動作は保証されません。

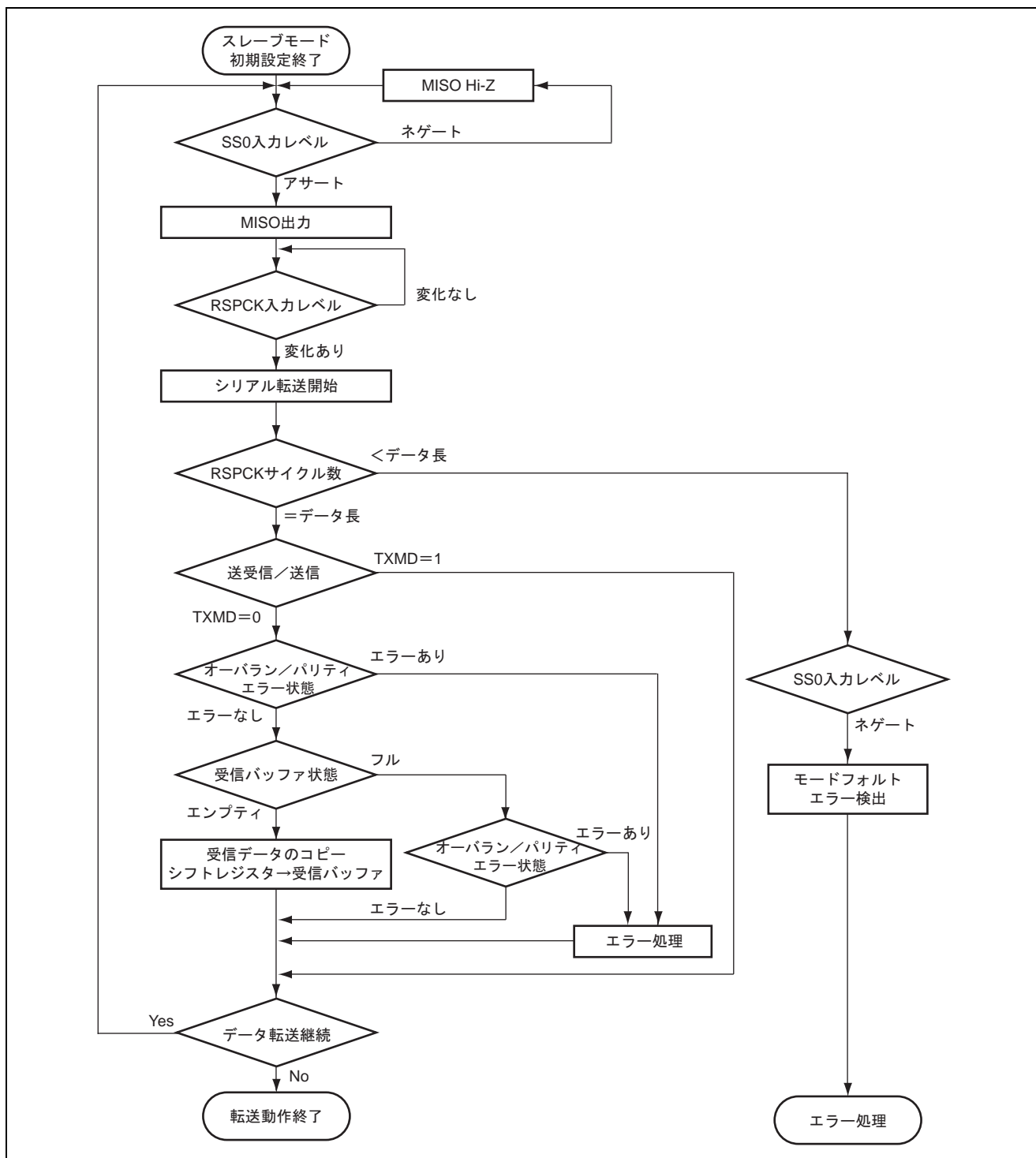


図 15.32 スレーブモード時の転送動作フロー (CPHA=1)

15.4.11 クロック同期式動作

RSPI は、RSPI 制御レジスタ (SPCR) の SPMS ビットが 1 であるとき、クロック同期式動作となります。クロック同期式動作は、SSL 端子を使用せず、RSPCK、MOSI、MISO の 3 本の端子を用いて通信を行い、SSL 端子は I/O ポートとして使用することができます。

クロック同期式動作は、SSL 端子を使用せず通信を行いますが、モジュール内部の動作は SPI 動作と同様の動作を行います。マスタ動作、スレーブ動作において、SPI 動作時と同様のフローで通信を行うことができますが、SSL 端子を使用しませんので、モードフォルトエラーの検出を行いません。

また、クロック同期式動作では、スレーブモード (MSTR=0) 時に RSPI コマンドレジスタ (SPCMD) の CPHA ビットを 0 に設定した場合の動作について保証していません。

(1) マスタモード動作

(a) シリアル転送の開始

RSPI ステータスレジスタ (SPSR) の SPTEF ビットが 1 の状態で RSPI データレジスタ (SPDR) ヘデータを書き込むと、RSPI は SPDR の送信バッファのデータを更新します。SPDR への書き込み、または CPU から SPTEF の 1 読み出し後の 0 書き込みによって SPTEF ビットを 0 にクリアした状態で、シフトレジスタが空の場合には、RSPI は送信バッファのデータをシフトレジスタにコピーしてシリアル転送を開始します。RSPI は、シフトレジスタに送信データをコピーするとシフトレジスタのステータスをフルに変更し、シリアル転送が終了するとシフトレジスタのステータスを空に変更します。シフトレジスタのステータスを CPU から参照することはできません。

なお、RSPI の転送フォーマットの詳細については、「15.4.4 転送フォーマット」を参照してください。

(b) シリアル転送の終了

RSPI はサンプリングタイミングに対応する RSPCK エッジを送出するとシリアル転送を終了します。RSPI ステータスレジスタ (SPSR) の SPRF ビットが 0 で受信バッファに空きがある場合には、シリアル転送終了後にシフトレジスタから RSPI データレジスタ (SPDR) の受信バッファにデータをコピーします。

なお、最終サンプリングタイミングは転送データのビット長に依存して変化します。マスタモードの RSPI のデータ長は、RSPI コマンドレジスタ (SPCMD) の SPB2~SPB0 ビットの設定値に依存します。RSPI の転送フォーマットの詳細については、「15.4.4 転送フォーマット」を参照してください。

(c) シーケンス制御

マスタモード時の転送フォーマットは、RSPI シーケンス制御レジスタ (SPSCR)、RSPI コマンドレジスタ 0~7 (SPCMD0~7)、RSPI ビットレートレジスタ (SPBR)、RSPI クロック遅延値レジスタ (SPCKD)、RSPI スレーブセレクトネゲート遅延レジスタ (SSLND)、RSPI 次アクセス遅延値レジスタ (SPND) によって決定されます。クロック同期式動作時は、SSL 信号の出力を行いませんが、これらの設定は有効です。

SPSCR は、マスタモードの RSPI で実行するシリアル転送のシーケンス構成を決定するためのレジスタです。SPCMD0~7 には、SSL 出力信号値、MSB/LSB ファースト、データ長、ビットレート設定の一部、RSPCK 極性/位相、SPCKD の参照要否、SSLND の参照要否、SPND の参照要否が設定されています。SPBR にはビットレート設定の一部、SPCKD には RSPI クロック遅延値、SSLND には SSL ネゲート遅延、SPND には次アクセス遅延値が設定されています。

RSPI は、SPSCR に設定されたシーケンス長に従って、SPCMD0~7 の一部/全部からなるシーケンスを構成します。RSPI には、シーケンスを構成している SPCMD に対するポインタが存在します。このポインタの値は、RSPI シーケンスステータスレジスタ (SPSSR) の SPCP2~0 ビットの読み出しによって CPU から確認可能です。RSPI 制御レジスタ (SPCR) の SPE ビットを 1 にして RSPI 機能をイネーブルにすると、RSPI はコマンド

に対するポインタを SPCMD0 にセットし、シリアル転送の開始時に SPCMD0 の設定内容を転送フォーマットに反映します。RSPI は、各データ転送の次アクセス遅延期間が終了するたびにポインタをインクリメントします。シーケンスを構成している最終コマンドに対応するシリアル転送が終了すると、RSPI はポインタを SPCMD0 にセットするので、シーケンスは繰り返し実行されます。

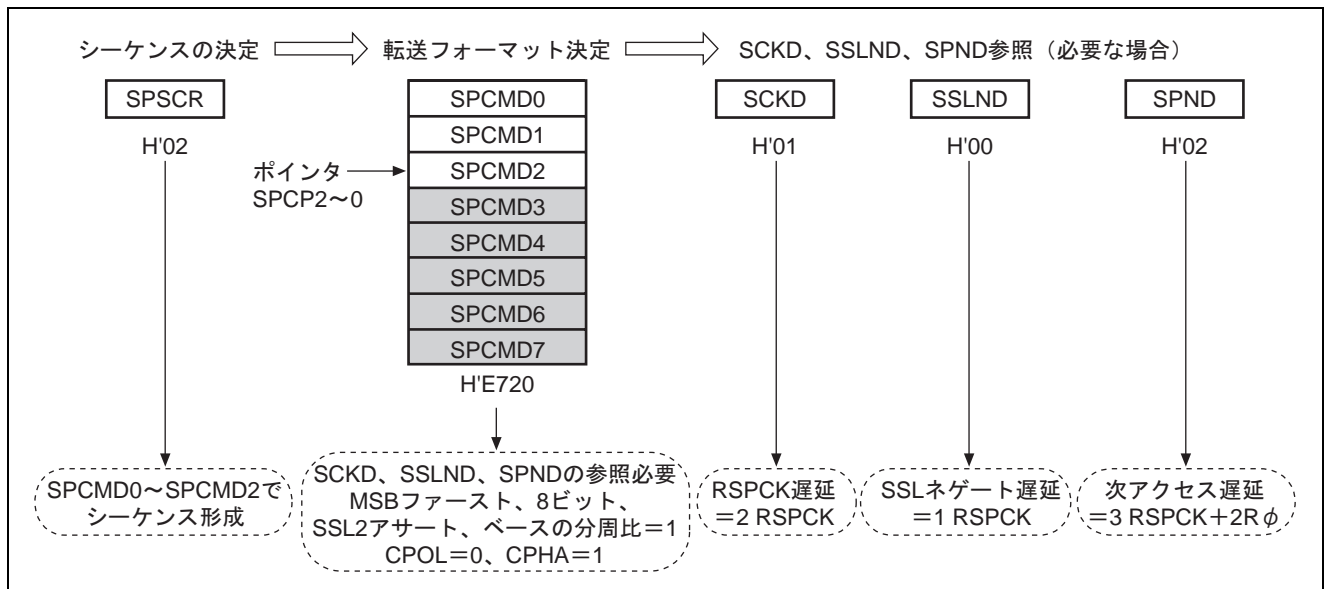


図 15.33 マスタモードでのシリアル転送方式の決定方法

(d) 初期化フロー

図 15.34 に、クロック同期式動作時の RSPI をマスターモードで使用する場合の初期化フローの例を示します。なお、割り込みコントローラ、DMAC、入出力ポートの設定方法については、各ブロックの説明を参照してください。

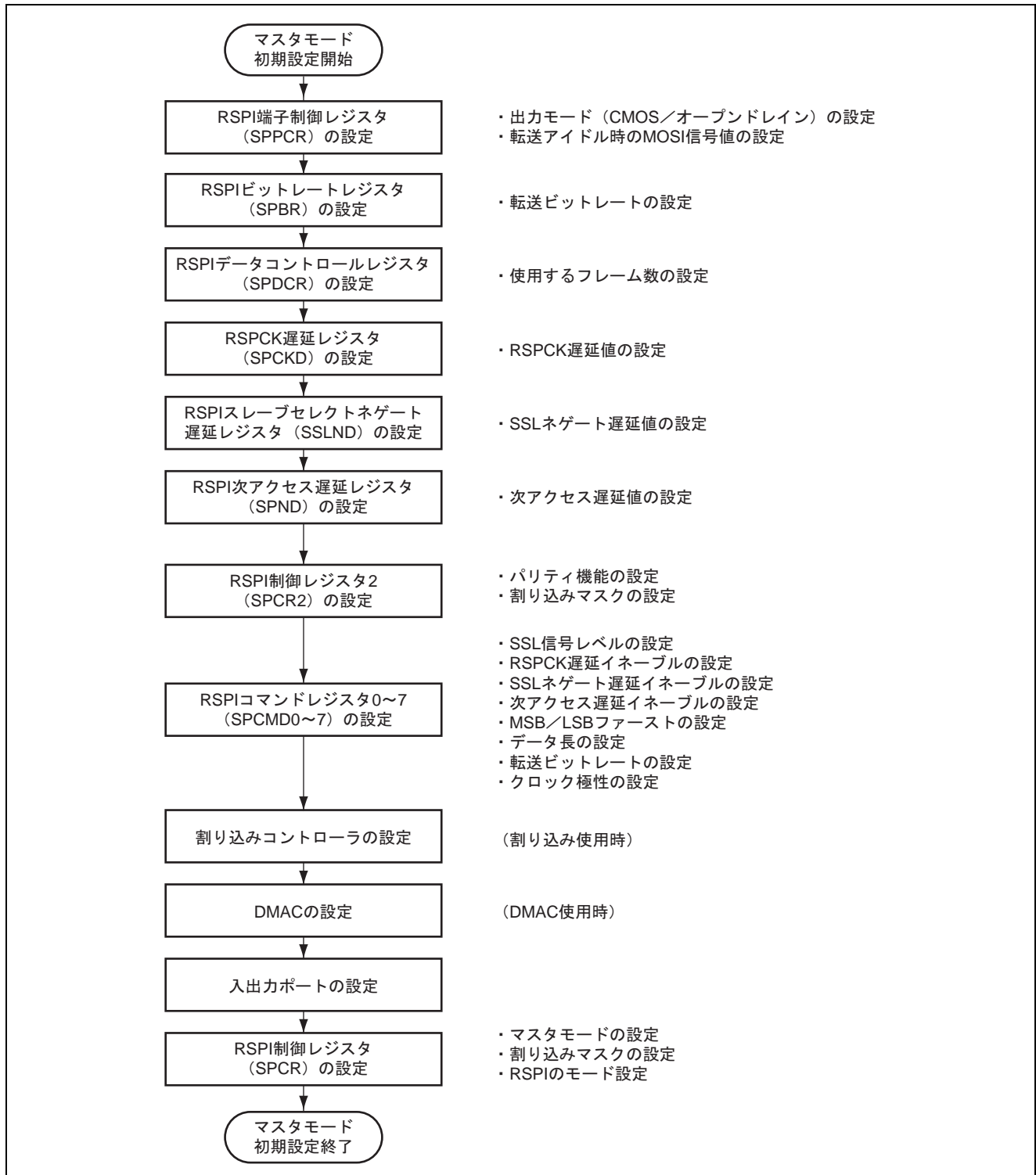


図 15.34 マスターモード時の初期化フロー例

(e) 転送動作フロー

図 15.35 に、クロック同期式動作時、マスタモードの転送動作フローを示します。

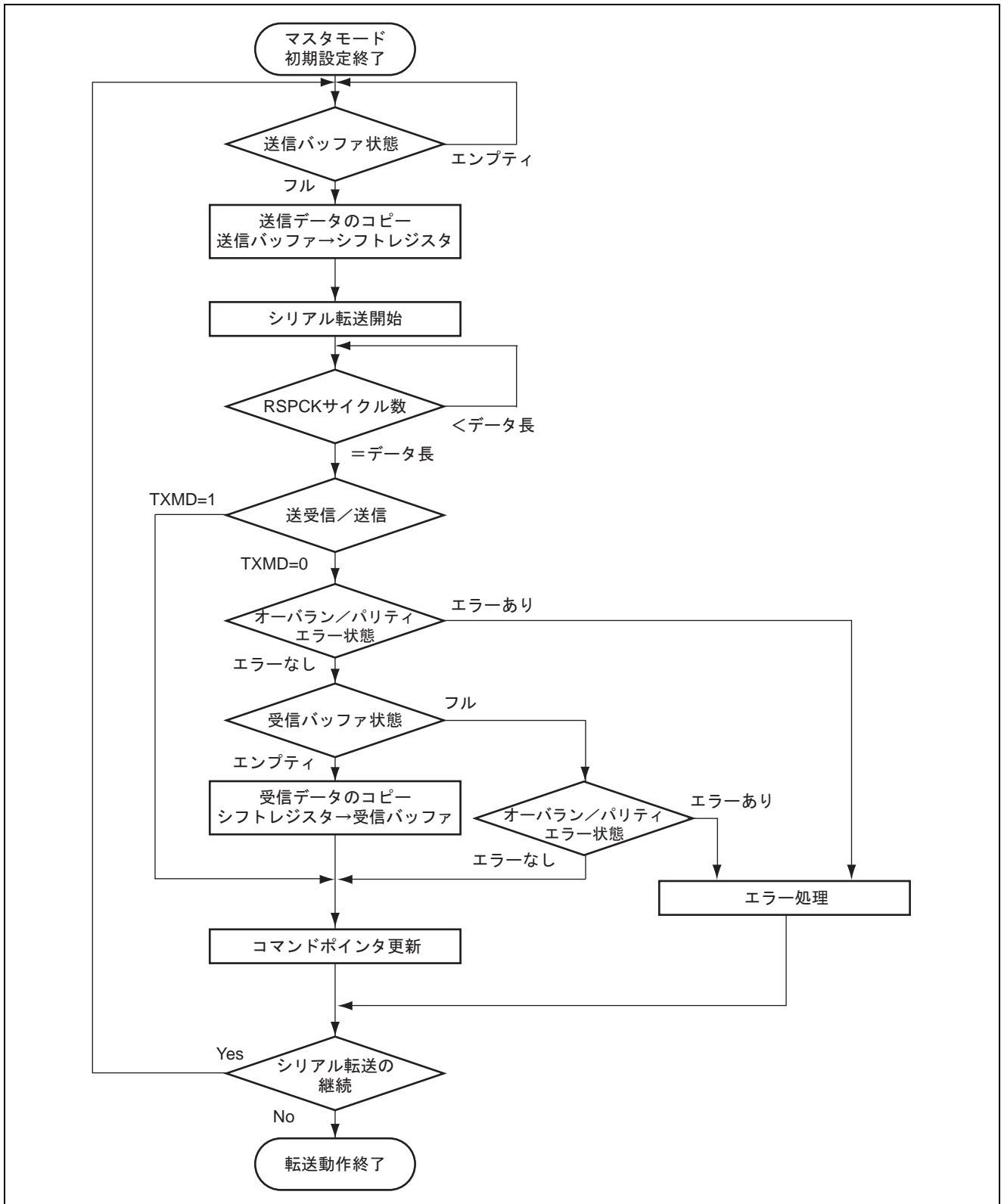


図 15.35 マスタモード時の転送動作フロー

(2) スレーブモード動作

(a) シリアル転送の開始

RSPI は、RSPI 制御レジスタ (SPCR) の SPMS ビットが 1 であるとき、最初の RSPCK エッジがシリアル転送開始のトリガになります。

RSPI は、シフトレジスタが空の状態ではシリアル転送の開始を検出した場合、シフトレジスタの状態をフルに変更し、シリアル転送中に送信バッファからシフトレジスタにデータがコピーできないようにします。シリアル転送の開始よりも前にシフトレジスタがフルであった場合、RSPI はシフトレジスタの状態をフルのまま変更しません。

SPMS ビットが 1 であるときは、RSPI は MISO 出力信号を常にドライブします。

なお、RSPI の転送フォーマットの詳細については、「15.4.4 転送フォーマット」を参照してください。

(b) シリアル転送の終了

RSPI は最終サンプリングタイミングに相当する RSPCK エッジを検出するとシリアル転送を終了します。RSPI ステータスレジスタ (SPSR) の SPRF ビットが 0 で受信バッファに空きがある場合には、シリアル転送の終了後に、RSPI はシフトレジスタから RSPI データレジスタ (SPDR) の受信バッファに受信データをコピーします。また、SPRF ビットの値にかかわらず、RSPI はシリアル転送の終了後にシフトレジスタの状態を空に変更します。なお、最終サンプリングタイミングは転送データのビット長に依存して変化します。スレーブモードの RSPI のデータ長は SPCMD0 の SPB3~SPB0 ビットの設定値に依存します。RSPI の転送フォーマットの詳細については、「15.4.4 転送フォーマット」を参照してください。

(c) 初期化フロー

図 15.36 に、クロック同期式動作時の RSPI をスレーブモードで使用する場合の初期化フローの例を示します。なお、割り込みコントローラ、DMAC、入出力ポートの設定方法については、各ブロックの説明を参照してください。

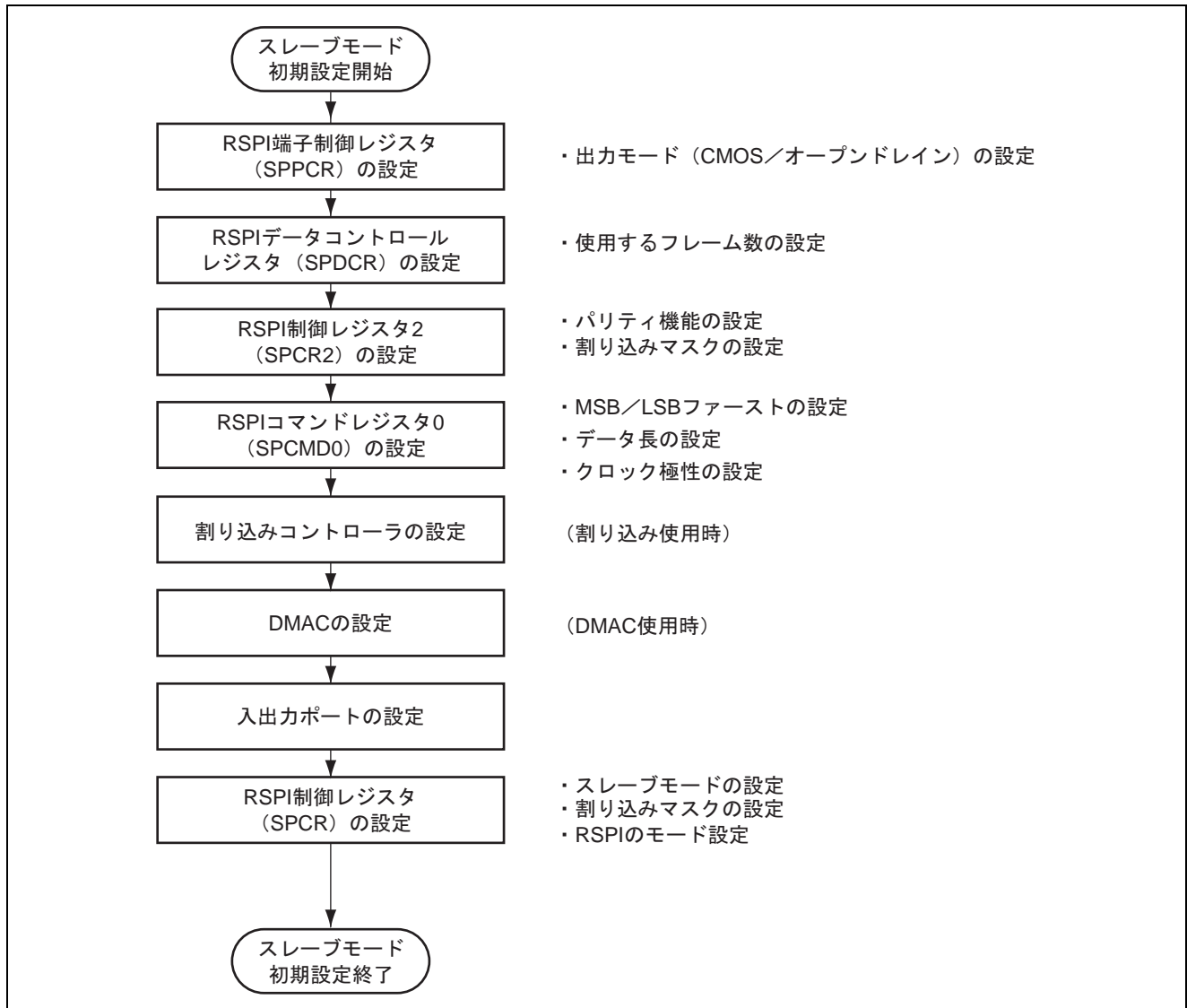


図 15.36 スレーブモード時の初期化フロー例

(d) 転送動作フロー

図 15.37 に、クロック同期式動作時の RSPI の転送動作フローを示します。

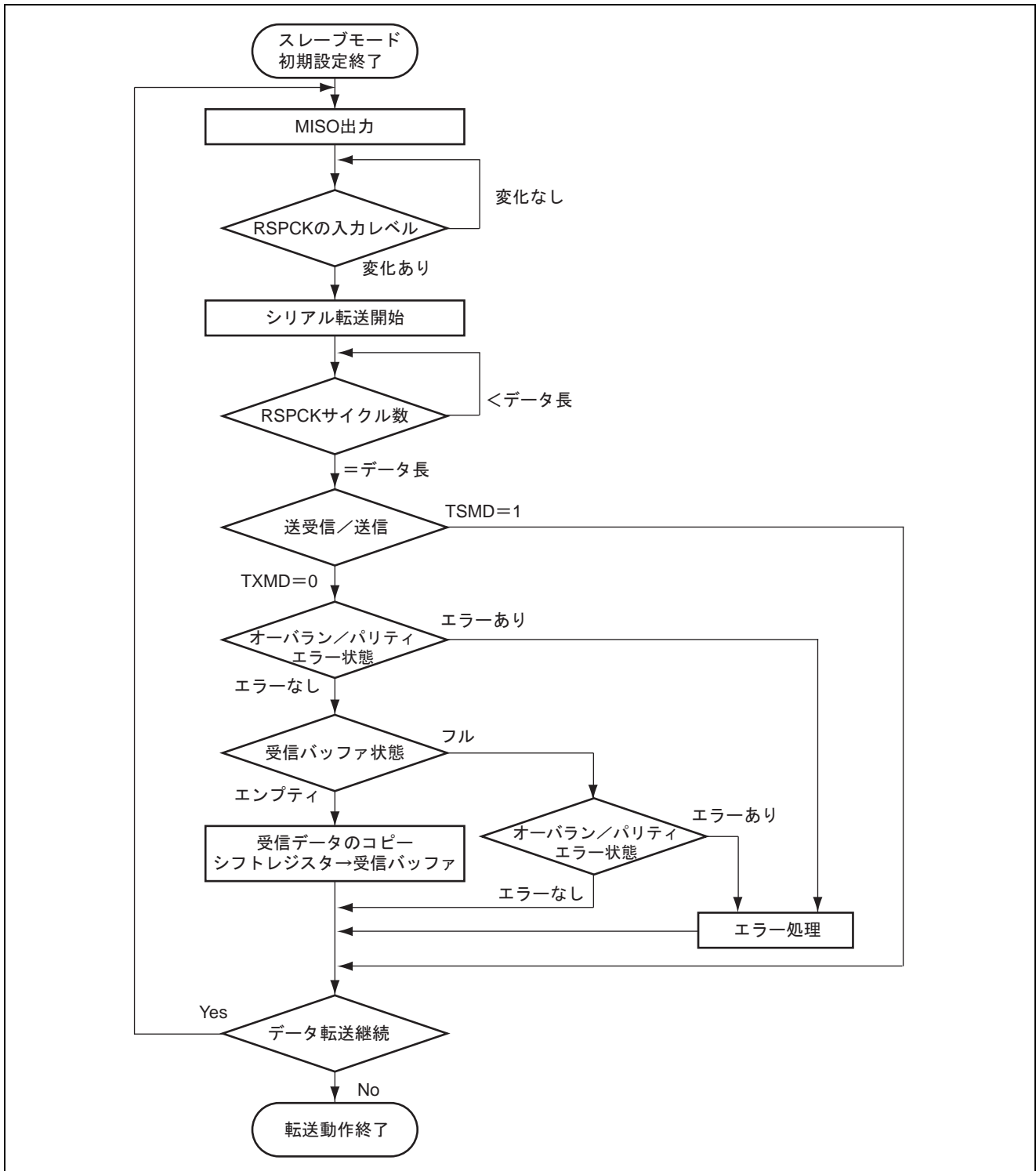


図 15.37 スレープモード時の転送動作フロー (CPHA=1)

15.4.12 エラー処理

図 15.38、図 15.39、図 15.40 に、RSPI のエラー処理を示します。マスタモード、スレーブモードで発生したエラーは、以下のエラー処理を行うことでエラー状態から復帰できます。

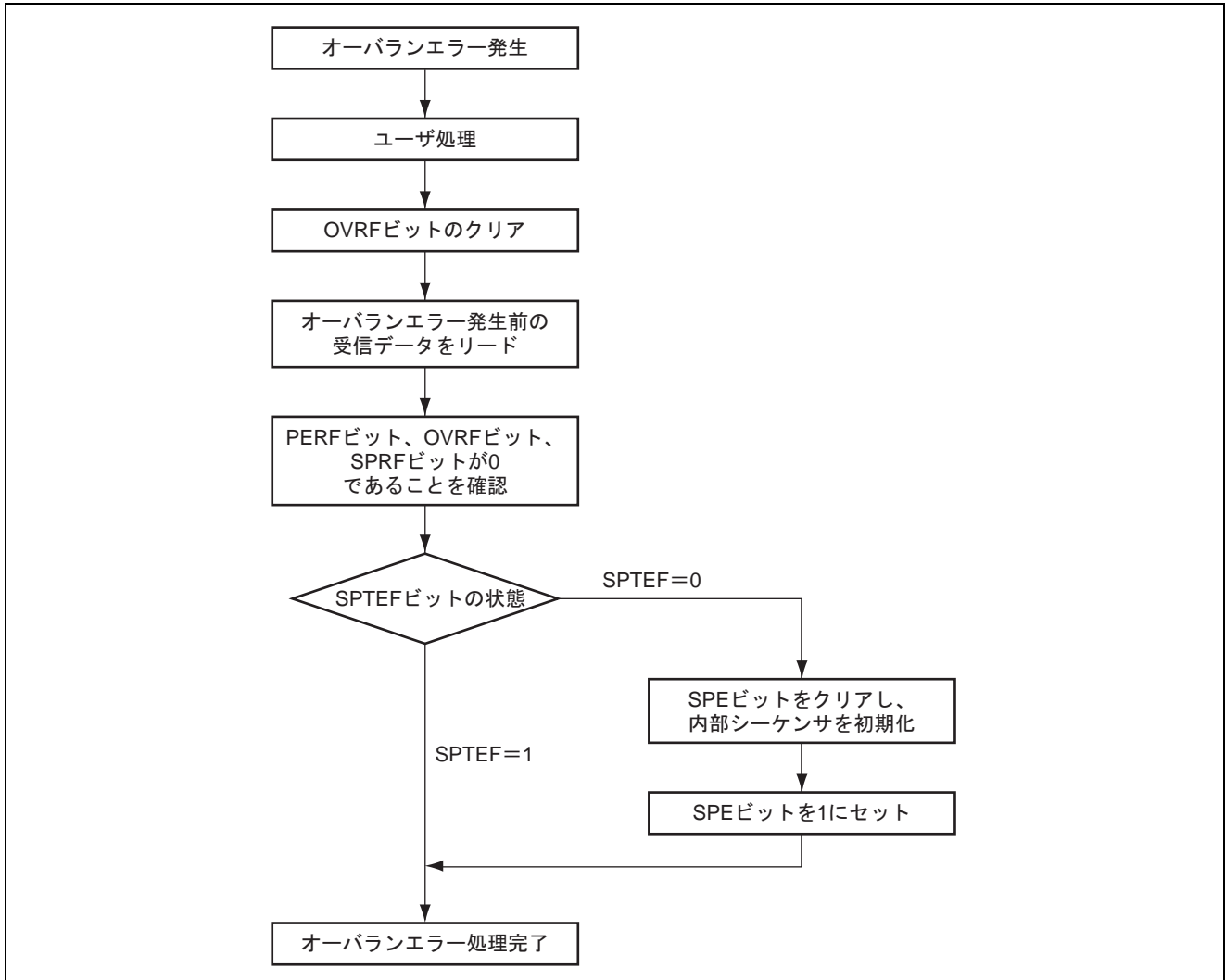


図 15.38 エラー処理（オーバランエラー）

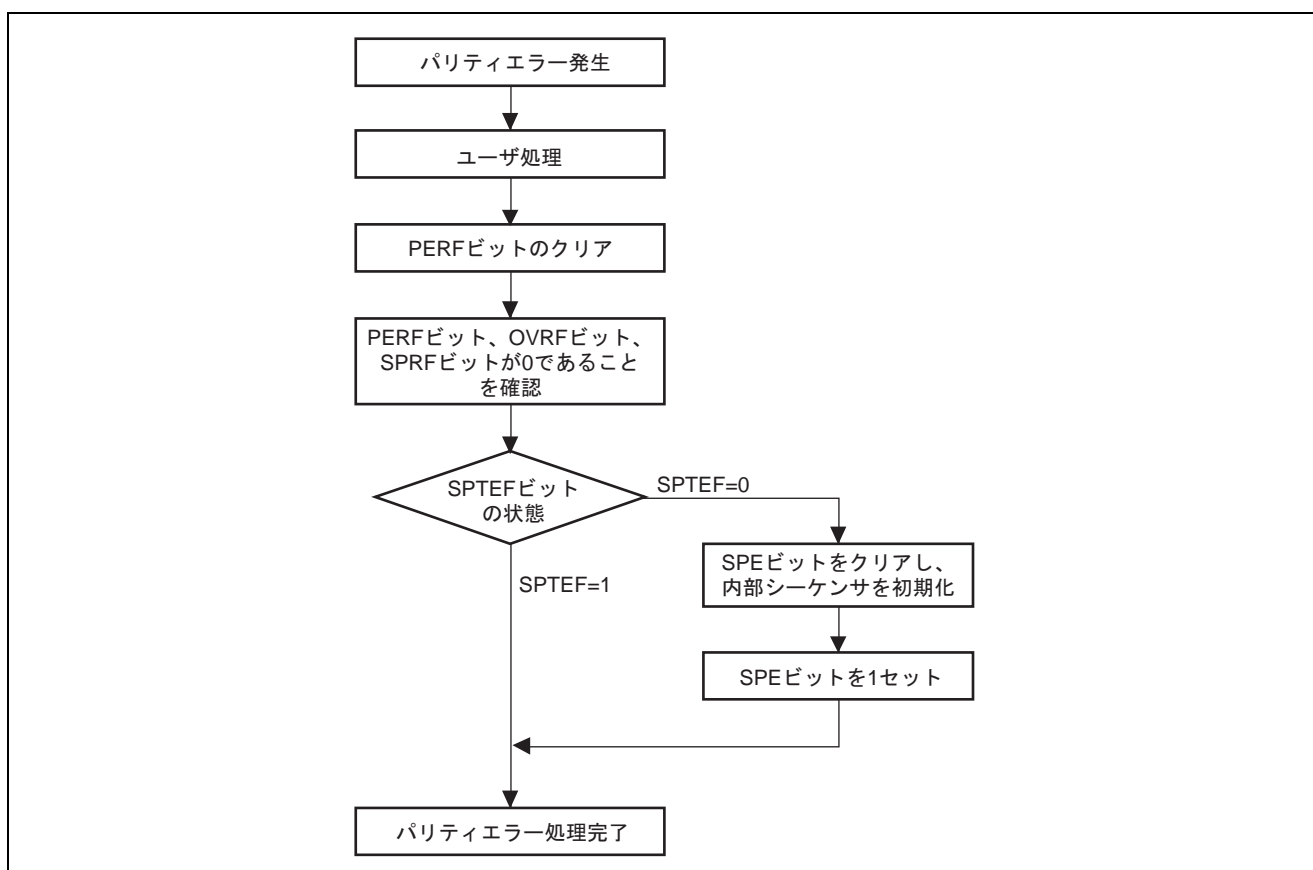


図 15.39 エラー処理 (パリティエラー)

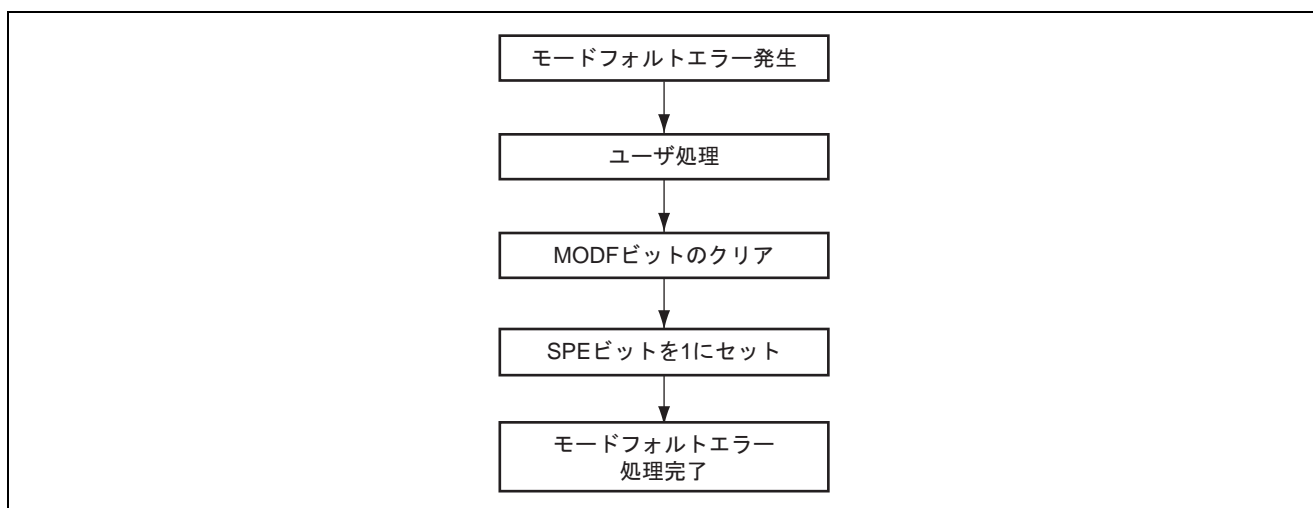


図 15.40 エラー処理 (モードフォルトエラー)

15.4.13 ループバックモード

CPU が RSPI 端子制御レジスタ (SPPCR) の SPLP2 ビットまたは SPLP ビットに 1 を書き込むと、RSPI は MISO 端子とシフトレジスタ間、MOSI 端子とシフトレジスタ間の経路を遮断し、シフトレジスタの入力経路と出力経路を接続します。これをループバックモードと呼びます。ループバックモードでシリアル転送を実行すると、RSPI の送信データまたは送信データの反転が RSPI の受信データになります。

SPLP2 ビット、SPLP ビットの設定と受信データの関係を表 15.12 に示します。

表 15.12 SPLP2 ビット、SPLP ビットの設定と受信データ

SPLP2	SPLP	受信データ
0	0	MOSI または MISO からの入力データ
0	1	送信データの反転
1	0	送信データ
1	1	送信データ

図 15.41 に、マスタモードの RSPI をループバックモード (SPLP2 = 0, SPLP = 1) に設定した場合のシフトレジスタ入出力経路の構成を示します。

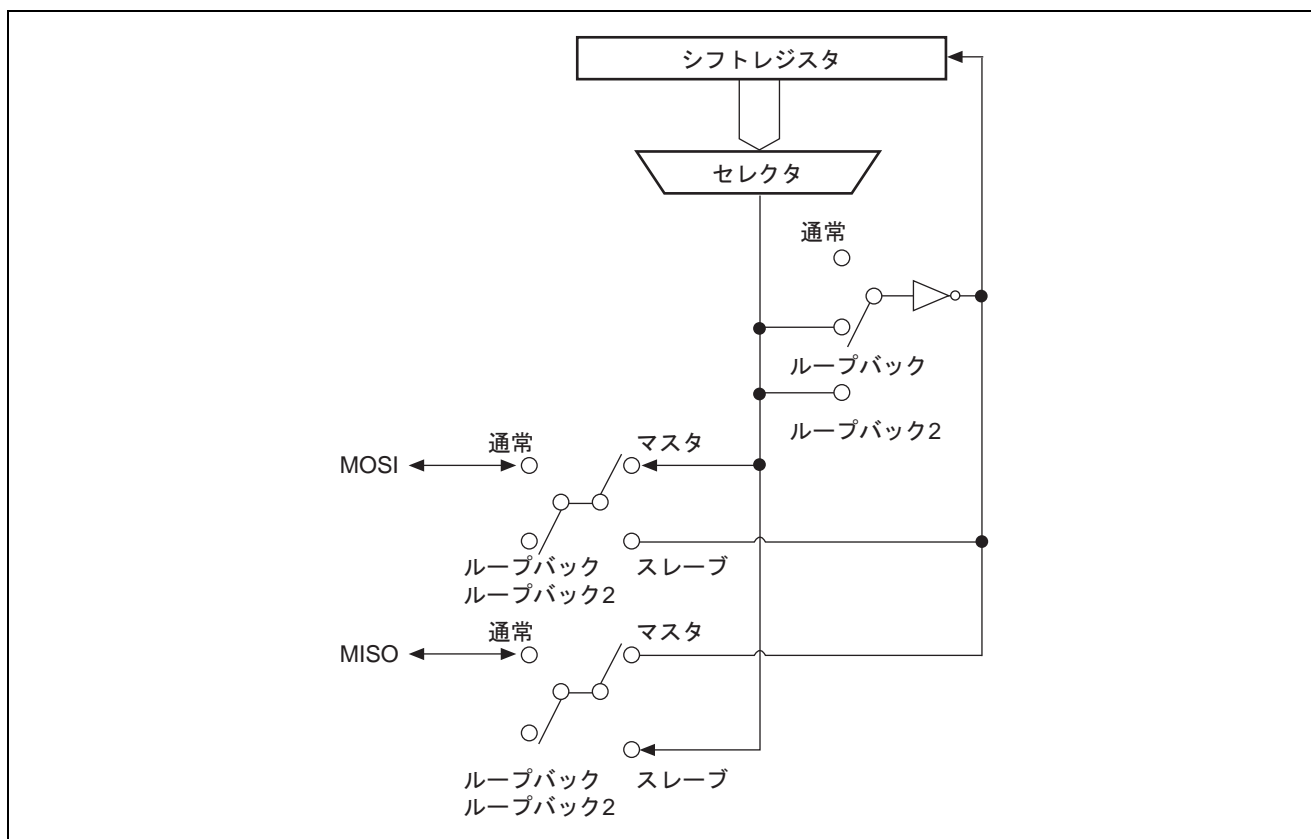


図 15.41 ループバックモード時のシフトレジスタ入出力構成 (マスタモード)

15.4.14 パリティ機能の自己診断

パリティ回路は、送信データに対するパリティ付加部と受信データに対するエラー検出部で構成されます。パリティ回路のパリティ付加部とエラー検出部の故障を検出するために、図 15.42 のフローに従い、パリティ回路の自己診断を行います。

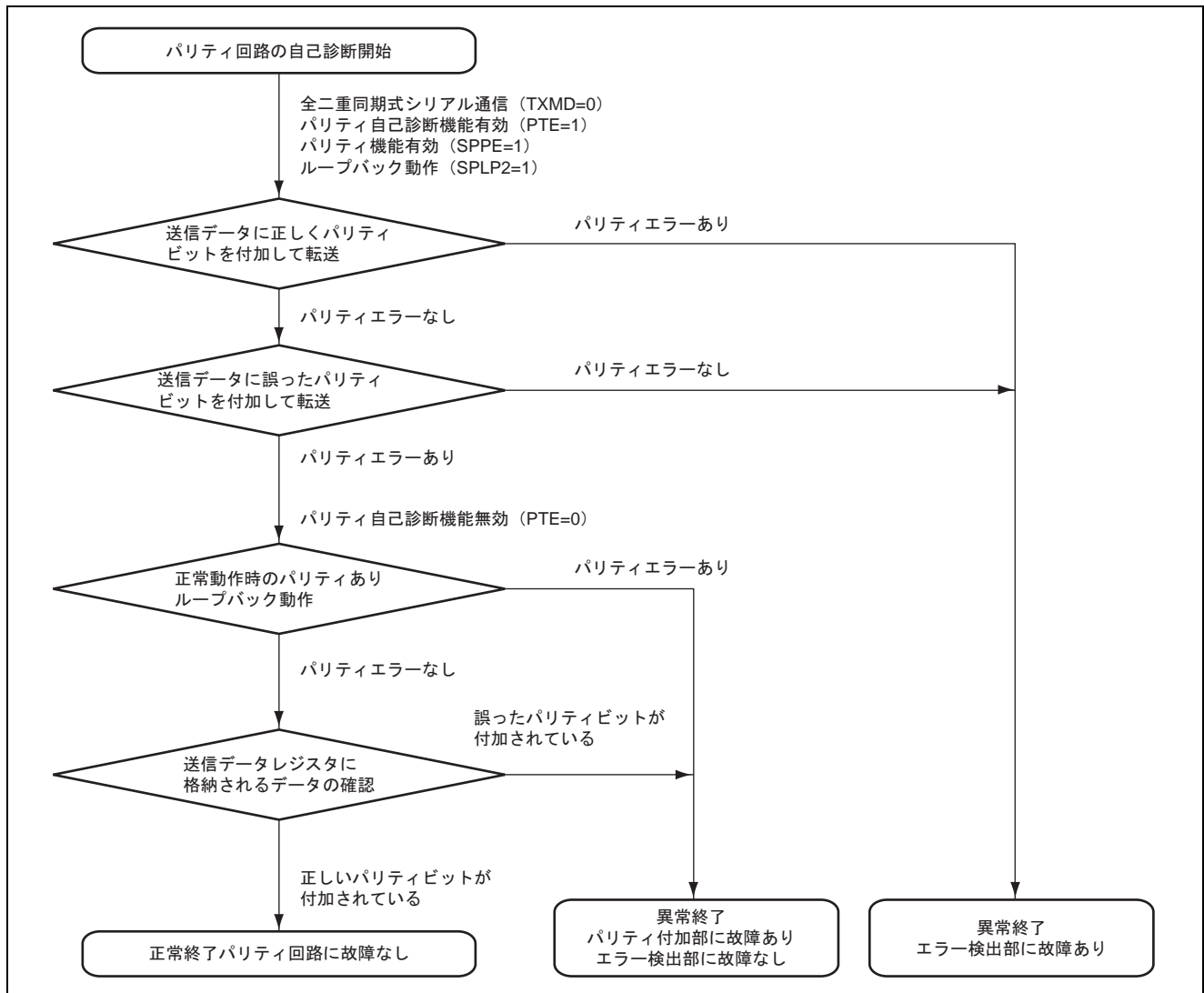


図 15.42 パリティ回路の自己診断フロー

15.4.15 割り込み要因

RSPI の割り込み要因には、受信バッファフル、送信バッファエンプティ、モードフォルト、オーバーランがあります。また、受信バッファフル、送信バッファエンプティの割り込み要求で DMAC を起動し、データ転送を行うことができます。

受信バッファフルの割り込み要求が `sp_rxint`、送信バッファエンプティの割り込み要求が `sp_txint`、モードフォルト、オーバーランの割り込み要求が `sp_errint` のベクタアドレスに割り付けられているため、フラグによる要因の判断が必要です。表 15.13 に、RSPI の割り込み要因を示します。

表 15.13 の割り込み条件が成立すると、割り込みが発生します。CPU または DMAC によるデータ転送で割り込み要因をクリアしてください。

表 15.13 RSPI の割り込み要因

名称	割り込み要因	略称	割り込み条件	DMAC 起動
<code>sp_rxint</code>	受信バッファフル	RXI	$(sprie=1) \cdot (sprf=1)$	○
<code>sp_txint</code>	送信バッファエンプティ	TXI	$(sptie=1) \cdot (sptef=1)$	○
<code>sp_errint</code>	モードフォルト	MOI	$(speie=1) \cdot (modf=1)$	—
	オーバーラン	OVI	$(speie=1) \cdot (ovrf=1)$	—
	パリティエラー	PRI	$(speie=1) \cdot (perf=1)$	—
<code>sp_idint</code>	RSPI アイドル	IDI	$(spiie=1) \cdot (idlnf=0)$	—

16. ハードウェア LIN (HWLIN)

ハードウェア LIN は、SCI と連携し、LIN 通信を行うものです。

16.1 特長

ハードウェア LIN には、以下の特長があります。また、**図 16.1** にハードウェア LIN のブロック図を示します。

【マスタモード】

- Sync Break発生
- バス衝突検出
- タイムアウト検出

【スレーブモード】

- Sync Break検出
- Sync Field計測
- Sync BreakおよびSync Field信号のSCI入力制御機能
- バス衝突検出
- タイムアウト検出

【注】 Wake Up 送信は SCI により送信、Wake Up 受信は IRQ により検出

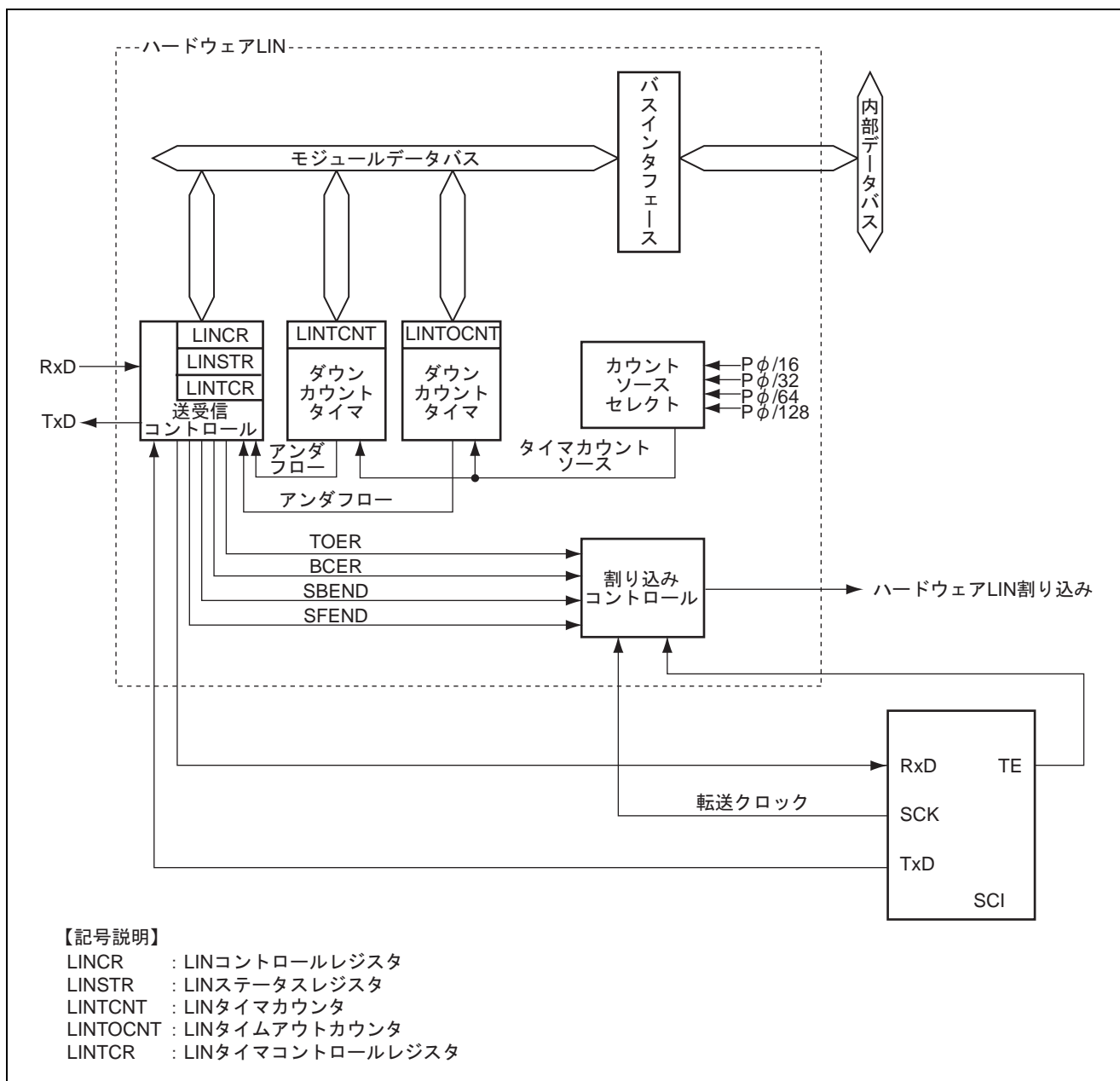


図 16.1 ハードウェア LIN のブロック図

16.2 入出力端子

ハードウェア LIN の端子構成を表 16.1 に示します。

表 16.1 端子構成

名称	略称	入出力	機能
レシーブデータ入力	RxD	入力	ハードウェア LIN の受信データ入力端子
トランスミットデータ出力	TxD	出力	ハードウェア LIN の送信データ出力端子

16.3 レジスタの説明

ハードウェア LIN には以下のレジスタがあります。

- LINコントロールレジスタ (LINCR)
- LINステータスレジスタ (LINSTR)
- LINタイマコントロールレジスタ (LINTCR)
- LINタイマカウンタ (LINTCNT)
- LINタイムアウトカウンタ (LINTOCNT)

16.3.1 LIN コントロールレジスタ (LINCR)

LINCR は、ハードウェア LIN の動作を制御します。

ビット	7	6	5	4	3	2	1	0
ビット名	LINE	MSS	RXDMRS	SBSTR	RXDSF	BCIE	SBIE	SFIE
初期値 :	0	0	0	0	0	0	0	0
R/W :	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

ビット	ビット名	初期値	R/W	説明
7	LINE	0	R/W	LIN イネーブル 0 : LIN は動作停止 1 : LIN は動作開始 (本ビットを 1 にした直後は、SCI への入力禁止) 0 を書き込むと、LINCR、LINSTR、LINTCR、LINTCNT、LINTOCNT およびハードウェア LIN 内部回路が初期化されます。
6	MSS	0	R/W	マスタ/スレーブセレクト 0 : スレーブモード (Sync Break 検出回路動作、Sync Field 計測回路動作) 1 : マスタモード (Sync Break 発生回路動作) LIN 動作モードを切り替える場合は、一度 LIN 動作を停止 (LINE=0) してください。
5	RXDMRS	0	R/W	RxD マスクリリースタイミングセレクト (スレーブモードのとき有効) RxD に加えたマスク制御を解除するタイミングを選択します。 0 : Sync Break 検出後に解除 1 : Sync Field 検出後に解除
4	SBSTR	0	R/W	Sync Break スタート (スレーブモードのとき有効) リードすると常に 0 が読み出されます。 SBSTR ビット設定後、RXDSF フラグが 1 になることを確認してから Sync Break を入力開始してください。 0 : Don't care 1 : SCI の RxD 入力をマスク開始

ビット	ビット名	初期値	R/W	説明
3	RXDSF	0	R	RxD 入力ステータスフラグ 0 : RxD 入力マスク解除状態 1 : RxD 入力マスク状態
2	BCIE	0	R/W	バスコンフリクトインタラプトイネーブル 0 : バス衝突検出割り込み禁止 1 : バス衝突検出割り込み許可
1	SBIE	0	R/W	Sync Break インタラプトイネーブル 0 : Sync Break 完了および Sync Break 検出割り込み禁止 1 : Sync Break 完了および Sync Break 検出割り込み許可
0	SFIE	0	R/W	Sync Field インタラプトイネーブル 0 : Sync Field 計測完了割り込み禁止 1 : Sync Field 計測完了割り込み許可

16.3.2 LIN ステータスレジスタ (LINSTR)

LINSTR は、ハードウェア LIN の動作制御とステータスフラグで構成されています。

ビット	7	6	5	4	3	2	1	0
ビット名	BCE	—	TOIE	RXDS	TOER	BCER	SBEND	SFEND
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】 * フラグをクリアするための1ライトのみ可能です。

ビット	ビット名	初期値	R/W	説明
7	BCE	0	R/W	バスコンフリクトディテクトイネーブル 0 : バス衝突検出回路停止 1 : バス衝突検出回路動作
6	—	0	R/W	リザーブビット 読み出すと常に 0 が読み出されます。書き込む値は常に 0 にしてください。
5	TOIE	0	R/W	タイムアウトインタラプトイネーブル 0 : タイムアウト検出割り込み禁止 1 : タイムアウト検出割り込み許可
4	RXDS	0	R/W	RxD セレクタ (スレーブモードのとき有効) RxD にマスク制御を加えるか加えないかを選択します。 0 : RxD 信号にマスク制御を加えない 1 : RxD 信号にマスク制御を加える

ビット	ビット名	初期値	R/W	説明
3	TOER	0	R/(W)*	タイムアウトエラー [セット条件] LINTOCNT がアンダフローしたとき [クリア条件] TOER=1 の状態をリードした後、TOER に 0 をライトしたとき
2	BCER	0	R/(W)*	バスコンフリクトエラー [セット条件] SCI の TE=1 のとき、LINE=1、BCE=1 の状態でバス衝突検出機能によりバスの衝突を検出したとき [クリア条件] BCER=1 の状態をリードした後、BCER に 0 をライトしたとき
1	SBEND	0	R/(W)*	Sync Break エンド [セット条件] LINTCNT がアンダフローしたとき [クリア条件] SBEND=1 の状態をリードした後、SBEND に 0 をライトしたとき
0	SFEND	0	R/(W)*	Sync Field エンド [セット条件] スレーブモード (MSS=0) のとき Sync Field のスタートビットおよび 0~6 ビットまでの期間を受信したとき [クリア条件] SFEND=1 の状態をリードした後、SFEND に 0 をライトしたとき

【注】 * フラグをクリアするための 1 ライトのみ可能です。

16.3.3 LIN タイマコントロールレジスタ (LINTCR)

LINTCR は、ハードウェア LIN のタイマ機能を制御します。

ビット	7	6	5	4	3	2	1	0
ビット名	TOCSTR	TCK1	TCK0	—	—	—	TCSTF	TCSTR
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	R/W

ビット	ビット名	初期値	R/W	説明
7	TOCSTR	0	R/W	タイムアウトカウンタスタート 0 : カウント停止 1 : カウント開始
6 5	TCK1 TCK0	0 0	R/W R/W	タイマカウントソース選択ビット TOCSTR もしくは TCSTR に 1 をライトするときはカウントソースの切り替えを同時に行わないでください。カウント動作中にカウントソースの切り替えを行わないでください。カウントソースの切り替えを行うときは、タイマのカウントを停止してください。 00 : Pφ/16 01 : Pφ/32 10 : Pφ/64 11 : Pφ/128
4~2	—	すべて 0	R/W	リザーブビット 読み出すと常に 0 が読み出されます。書き込む値は常に 0 にしてください。
1	TCSTF	0	R	タイマカウントステータスフラグ TCSTR に 1 ライト後、カウントが開始されると TCSTF が 1 になります。 また、TCSTR に 0 ライト後、カウントが停止されると TCSTF が 0 になります。 カウント開始後 TCSTF に 1 が読み込まれるまで、カウント停止後 TCSTF に 0 が読み込まれるまでは何もしないでください。 0 : カウント停止 1 : カウント中
0	TCSTR	0	R/W	タイマカウントスタート LIN タイマカウンタのカウント停止、カウント開始を選択します。本ビットが 0 のとき、LINTCNT はリロードされます。 0 : カウント停止 1 : カウント開始

16.3.4 LIN タイマカウンタ (LINTCNT)

LINTCNT は、リード/ライト可能な 16 ビットのダウンカウンタです。LINTCNT の 8 ビット単位のアクセスは禁止です。常に 16 ビット単位でアクセスしてください。

ビット	15	14	13	12	11	10	9	8
ビット名								
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
ビット	7	6	5	4	3	2	1	0
ビット名								
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

16.3.5 LIN タイムアウトカウンタ (LINTOCNT)

LINTOCNT は、リード/ライト可能な 16 ビットのダウンカウンタです。LINTOCNT の 8 ビット単位のアクセスは禁止です。常に 16 ビット単位でアクセスしてください。

ビット	15	14	13	12	11	10	9	8
ビット名								
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
ビット	7	6	5	4	3	2	1	0
ビット名								
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

16.3.6 LIN タイマカウンタと LIN タイムアウトカウンタの設定

1 ビット分の送信もしくは受信に必要とする LINTCNT と LINTOCNT の設定値 N とビットレート B の関係を表 16.2 に示します。

表 16.2 1 ビット分の LINTCNT と LINTOCNT の設定値 N とビットレート B の関係

設定値	誤差
$N = \frac{P\phi \times 10^6}{n \times B}$	$\text{誤差 (\%)} = \left\{ \frac{P\phi \times 10^6}{B \times n \times N} - 1 \right\} \times 100$

【記号説明】

B : ビットレート (bit/s)

N : 1 ビット分の送信もしくは受信に必要とする LINTCNT ・ LINTOCNT 値

Pφ : 動作周波数 (MHz)

n : 下表のとおり LINTCR の設定値によって決まります。

LINTCR の設定		n
TCK1	TCK0	
0	0	16
0	1	32
1	0	64
1	1	128

表 16.2 による計算結果と表 16.3、表 16.4 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余裕を持たせてください。

ビットレートが 2400/9600/10417/19200 のとき、1 ビット分の送信もしくは受信に必要とする LINTCNT・LINTCNT 値を表 16.3～表 16.6 に示します。

表 16.3 ビットレートが 2400bps のときの 1 ビット分の LINTCNT・LINTCNT 設定値

ビットレート (bit/s)	動作周波数 Pφ (MHz)	n	LINTCNT LINTCNT	誤差	ビットレート (bit/s)	動作周波数 Pφ (MHz)	n	LINTCNT LINTCNT	誤差
2400	8	16	208	0.16%	2400	20	16	521	-0.03%
		32	104	0.16%			32	260	0.16%
		64	52	0.16%			64	130	0.16%
		128	26	0.16%			128	65	0.16%
	10	16	260	0.16%		25	16	651	0.01%
		32	130	0.16%			32	326	-0.15%
		64	65	0.16%			64	163	-0.15%
		128	33	-1.36%			128	81	0.47%
	12	16	313	-0.16%		30	16	781	0.03%
		32	156	0.16%			32	391	-0.10%
		64	78	0.16%			64	195	0.16%
		128	39	0.16%			128	98	-0.35%
	14	16	365	-0.11%		35	16	911	0.05%
		32	182	0.16%			32	456	-0.06%
		64	91	0.16%			64	228	-0.06%
		128	46	-0.93%			128	114	-0.06%
	16	16	417	-0.08%		40	16	1042	-0.03%
		32	208	0.16%			32	521	-0.03%
		64	104	0.16%			64	260	0.16%
		128	52	0.16%			128	130	0.16%
	18	16	469	-0.05%					
		32	234	0.16%					
		64	117	0.16%					
		128	59	-0.69%					

表 16.4 ビットレートが 9600bps のときの 1 ビット分の LINTCNT・LINTOCNT 設定値

ビットレート (bit/s)	動作周波数 Pφ (MHz)	n	LINTCNT LINTOCNT	誤差	ビットレート (bit/s)	動作周波数 Pφ (MHz)	n	LINTCNT LINTOCNT	誤差
9600	8	16	52	0.16%	9600	20	16	130	0.16%
		32	26	0.16%			32	65	0.16%
		64	13	0.16%			64	33	-1.36%
		128	7	-6.99%			128	16	1.73%
	10	16	65	0.16%		25	16	163	-0.15%
		32	33	-1.36%			32	81	0.47%
		64	16	1.73%			64	41	-0.76%
		128	8	1.73%			128	20	1.73%
	12	16	78	0.16%		30	16	195	0.16%
		32	39	0.16%			32	98	-0.35%
		64	20	-2.34%			64	49	-0.35%
		128	10	-2.34%			128	24	1.73%
	14	16	91	0.16%		35	16	228	-0.06%
		32	46	-0.93%			32	114	-0.06%
		64	23	-0.93%			64	57	-0.06%
		128	11	3.57%			128	28	1.73%
	16	16	104	0.16%		40	16	260	0.16%
		32	52	0.16%			32	130	0.16%
		64	26	0.16%			64	65	0.16%
		128	13	0.16%			128	33	-1.36%
	18	16	117	0.16%					
		32	59	-0.69%					
		64	29	1.02%					
		128	15	-2.34%					

表 16.5 ビットレートが 10417bps のときの 1 ビット分の LINTCNT・LINTOCNT 設定値

ビットレート (bit/s)	動作周波数 Pφ (MHz)	n	LINTCNT LINTOCNT	誤差	ビットレート (bit/s)	動作周波数 Pφ (MHz)	n	LINTCNT LINTOCNT	誤差
10417	8	16	48	0.00%	10417	20	16	120	0.00%
		32	24	0.00%			32	60	0.00%
		64	12	0.00%			64	30	0.00%
		128	6	0.00%			128	15	0.00%
	10	16	60	0.00%		25	16	150	0.00%
		32	30	0.00%			32	75	0.00%
		64	15	0.00%			64	37	1.35%
		128	7	7.14%			128	19	-1.32%
	12	16	72	0.00%		30	16	180	0.00%
		32	36	0.00%			32	90	0.00%
		64	18	0.00%			64	45	0.00%
		128	9	0.00%			128	22	2.27%
	14	16	84	0.00%		35	16	210	0.00%
		32	42	0.00%			32	105	0.00%
		64	21	0.00%			64	52	0.96%
		128	10	5.00%			128	26	0.96%
	16	16	96	0.00%		40	16	240	0.00%
		32	48	0.00%			32	120	0.00%
		64	24	0.00%			64	60	0.00%
		128	12	0.00%			128	30	0.00%
	18	16	108	0.00%					
		32	54	0.00%					
		64	27	0.00%					
		128	13	3.84%					

表 16.6 ビットレートが 19200bps のときの 1 ビット分の LINTCNT・LINTOCNT 設定値

ビットレート (bit/s)	動作周波数 Pφ (MHz)	n	LINTCNT LINTOCNT	誤差	ビットレート (bit/s)	動作周波数 Pφ (MHz)	n	LINTCNT LINTOCNT	誤差
19200	8	16	26	0.16%	19200	20	16	65	0.16%
		32	13	0.16%			32	33	-1.36%
		64	7	-6.99%			64	16	1.73%
		128	3	8.51%			128	8	1.73%
	10	16	33	-1.36%		25	16	81	0.47%
		32	16	1.73%			32	41	-0.76%
		64	8	1.73%			64	20	1.73%
		128	4	1.73%			128	10	1.73%
	12	16	39	0.16%		30	16	98	-0.35%
		32	20	-2.34%			32	49	-0.35%
		64	10	-2.34%			64	24	1.73%
		128	5	-2.34%			128	12	1.73%
	14	16	46	-0.93%		35	16	114	-0.06%
		32	23	-0.93%			32	57	-0.06%
		64	11	3.57%			64	28	1.73%
		128	6	-5.06%			128	14	1.73%
	16	16	52	0.16%		40	16	130	0.16%
		32	26	0.16%			32	65	0.16%
		64	13	0.16%			64	33	-1.36%
		128	7	-6.99%			128	16	1.73%
	18	16	59	-0.69%					
		32	29	1.02%					
		64	15	-2.34%					
		128	7	4.63%					

16.4 動作説明

16.4.1 マスタモード

図 16.2 にマスタモードでの、ヘッダフィールドの送信時の動作例を示します。また、図 16.3～図 16.6 にヘッダフィールドの送信を行うためのフローチャートを示します。

ハードウェア LIN は、ヘッダフィールド送信時、以下のように動作します。

1. LINTCRのTCSTRビットに1を書き込むと、LINTCNTに設定された期間、TxD端子からLowレベルを出力します。
2. LINTCNTがアンダフローすると、TxD端子の出力を反転し、LINSTRのSBENDフラグが1にセットされます。また、LINCRのSBIEビットを1に設定している場合は、割り込みが発生します。
3. SCIにより、H'55を送信します。
4. SCIにより、H'55の送信が完了後、IDフィールドを送信します。
5. IDフィールドの送信完了後、レスポンスフィールドの通信を行います。

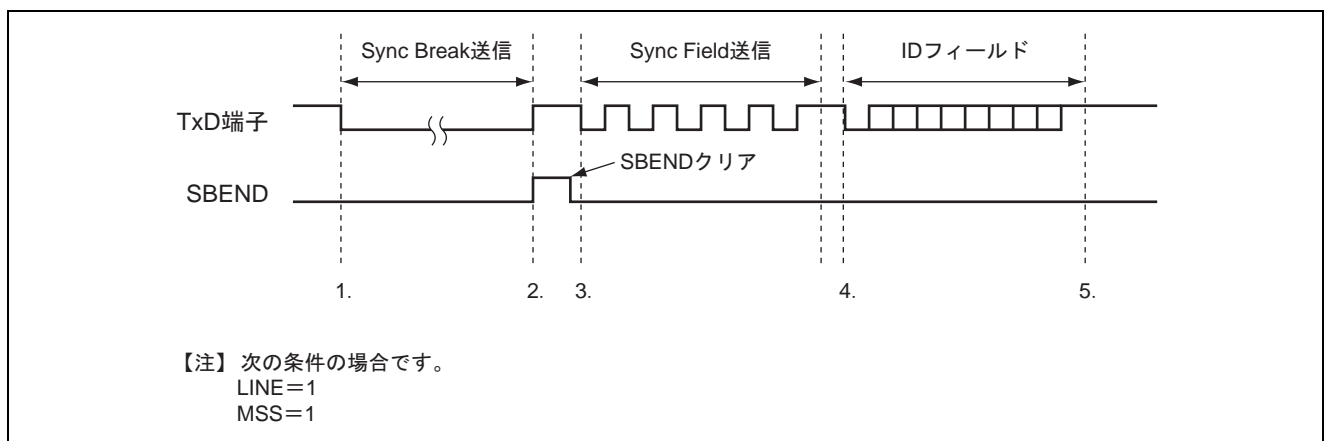


図 16.2 ヘッダフィールドの送信時の動作例

(1) マスタモードの初期設定

マスタモードでの初期設定例を図 16.3 に示します。

LIN 通信の前に LINCRC の LINE ビットをクリアした後、図 16.3 のフローチャートの例に従って初期設定してください。動作モードの変更の場合も必ず LINE ビット 0 にクリアしてから変更を行ってください。

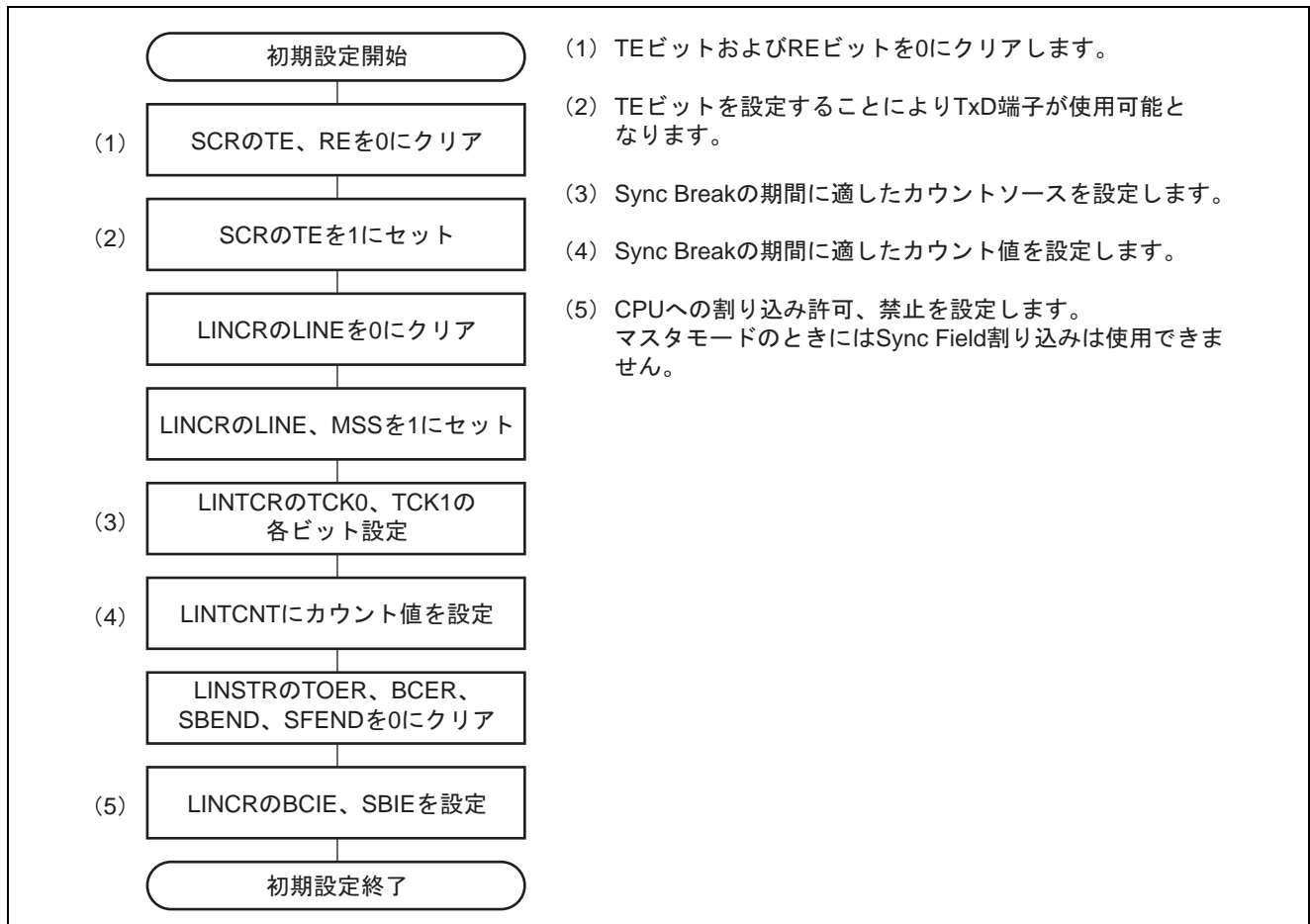


図 16.3 マスタモードの初期設定例

(2) マスタモードの Sync Break 送信

マスタモードでの Sync Break 送信設定例を図 16.4 に示します。

Sync Break 送信時にハードウェア LIN は以下のように動作します。

ハードウェア LIN は LINTCR の TCSTR を 1 にセットすることによって、LINTCR の TCK1、TCK0 で選択されたタイマカウントソースに同期して Sync Break が送信されます。

Sync Break の送信が開始されると自動的に LINTCR の TCSTF が 1 にセットされ、LINTCNT に設定した期間、TxD 端子から Low レベルが出力されます。

LINTCNT のカウントは LINTCR の TCK1、TCK0 で選択されたタイマカウントソースに同期してダウンカウントします。LINTCNT がアンダフローすると TxD 端子の出力を反転し、LINSTR の SBEND が 1 にセットされます。また、LINCR の SBIE を 1 にセットしている場合は Sync Break 完了を割り込み要因とする割り込みが発生します。

LINTCNT はアンダフローすると、設定した値にリロードを行います。リロード後も LINTCR の TCSTR が 1 にセットされている場合、LINTCNT はリロード後もダウンカウントを続けます。

LINSTR の SBEND が 1 にセットされたことを確認した後、LINTCR の TCSTR、LINSTR の SBEND をそれぞれ 0 にクリアしてください。LINTCNT は LINTCR の TCSTR を 0 にクリアすると、ダウンカウントを停止し、リロードします。

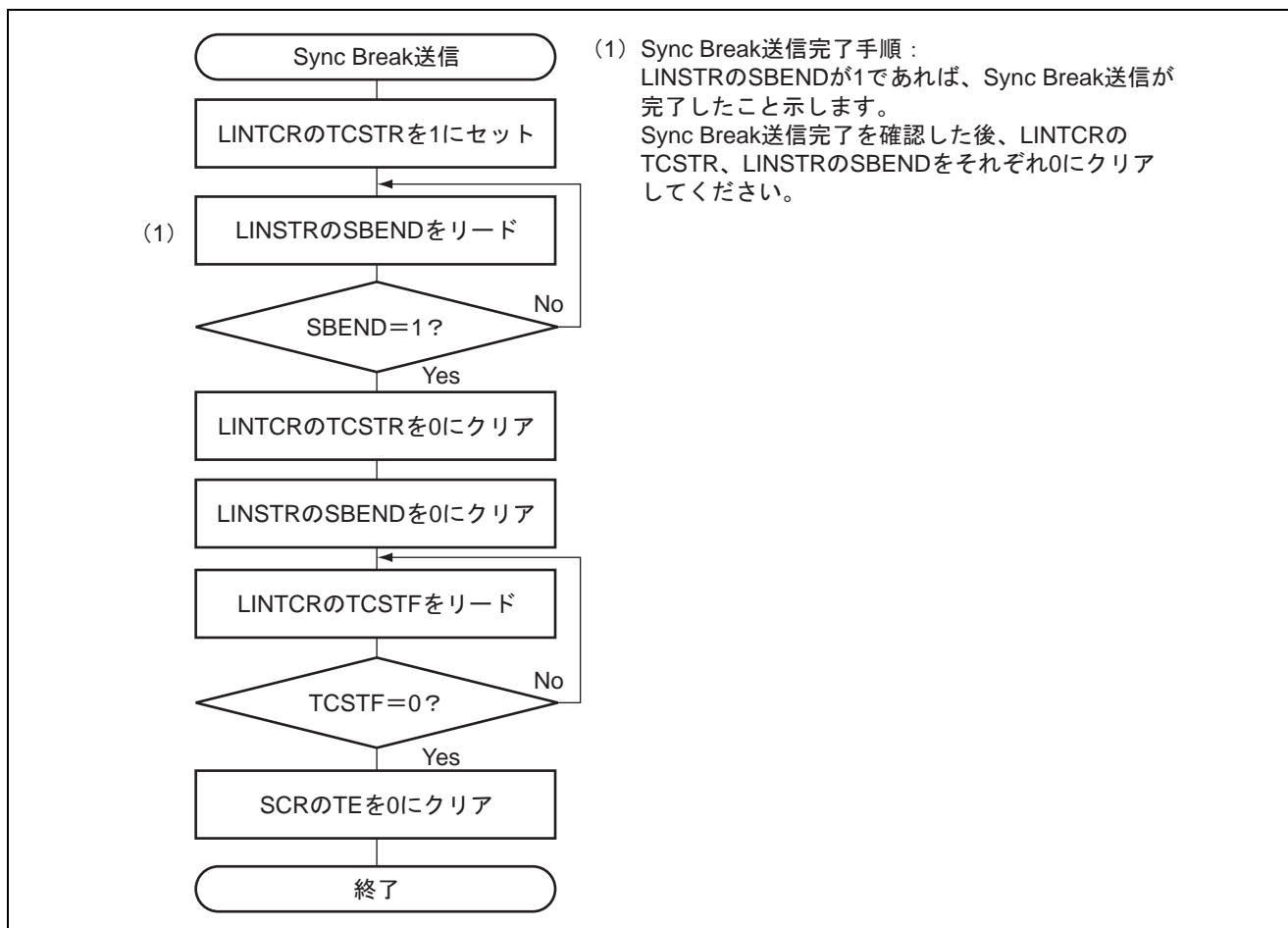


図 16.4 マスタモードの Sync Break 送信設定例

(3) マスタモードの Sync Field 送信

マスタモードでの Sync Field 送信設定例を図 16.5 に示します。

Sync Field 送信時にハードウェア LIN は以下のように動作します。

ハードウェア LIN は LINSTR の BCE が 1 にセットされている場合、バス衝突検出回路を動作させます。バス衝突検出機能の詳細は「16.5 バス衝突検出機能」を参照してください。

Sync Field 送信は SCI により H'55 を送信します。

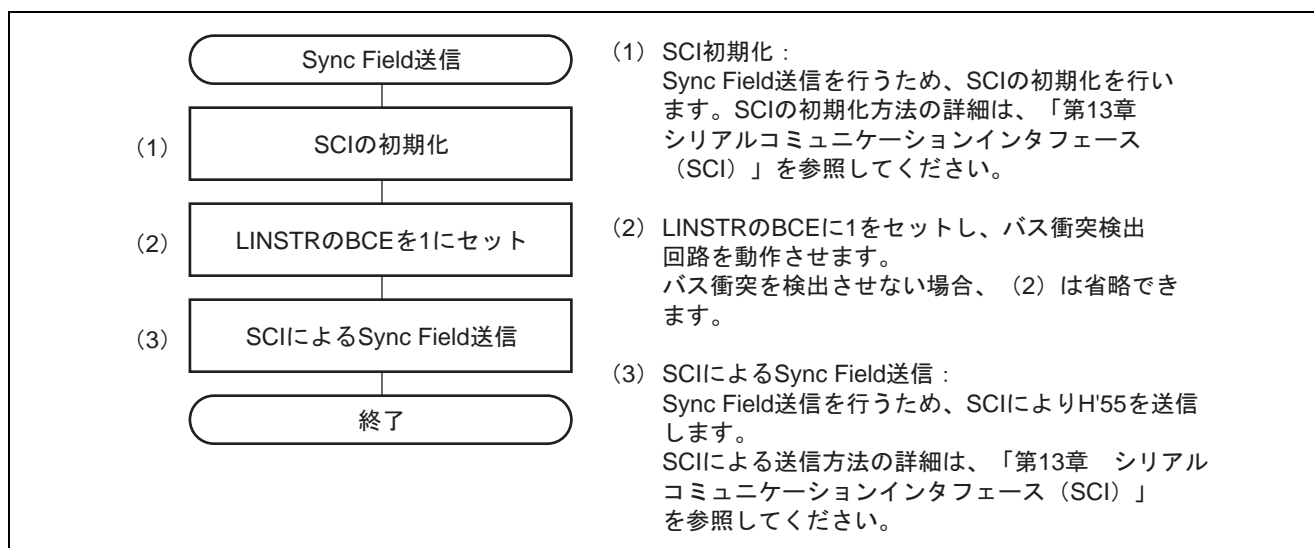


図 16.5 マスタモードの Sync Field 送信設定例

(4) マスタモードの ID フィールド送信

マスタモードでの ID フィールド送信設定例を図 16.6 に示します。

ID フィールドの送信は SCI により行います。

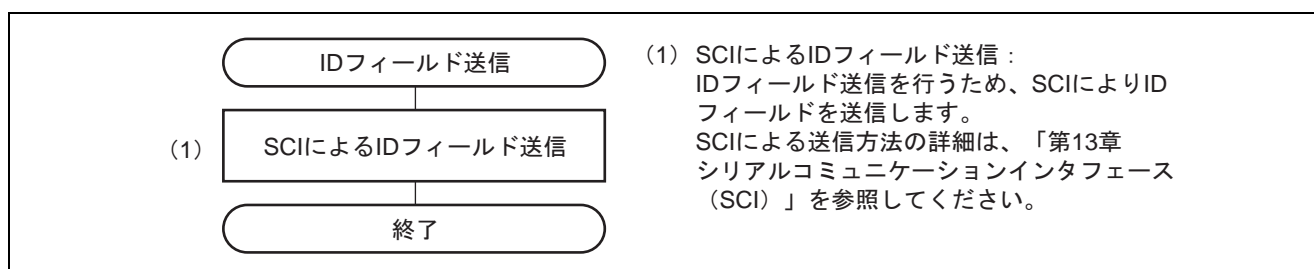


図 16.6 マスタモードの ID フィールド送信設定例

(5) マスタモードのレスポンスフィールド通信

マスタモードでのレスポンスフィールド通信設定例を図 16.7 に示します。

レスポンスフィールド通信は SCI により行います。

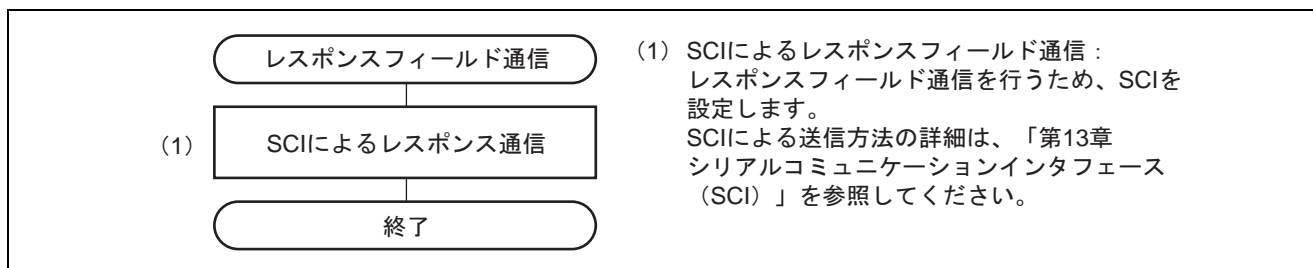


図 16.7 マスタモードのレスポンス通信設定例

16.4.2 スレーブモード

図 16.8 にスレーブモードでの、ヘッダフィールドの受信時の動作例を示します。また、図 16.9～図 16.12 にヘッダフィールドの受信を行うためのフローチャートを示します。

ハードウェア LIN は、ヘッダフィールド受信時、以下のように動作します。

1. LINCRCのSBSTRビットに1書き込むと、RxD入力のマスクが開始されます。
2. LINTCNTに設定した期間以上のLowレベルが入力されるとSync Breakとして検出します。このとき、LINSTRのSBENDフラグが1にセットされます。また、LINCRCのSBIEビットを1に設定している場合は、Sync Break検出を割り込み要因とする割り込みが発生します。
3. Sync Field(H'55)を受信します。このとき、LINTCNTにより、スタートビットおよび0～6ビットまでの期間を測定します。また、Sync Fieldの信号をSCIのRxDに入力するか禁止にするかをLINCRCのRXDMRSビットにより選択できます。
4. Sync Field計測が完了するとLINSTRのSFENDフラグが1にセットされます。また、LINCRCのSFIEビットを1に設定している場合は、Sync Field計測完了を割り込み要因とする割り込みが発生します。
5. Sync Field計測完了後、LINTCNTのカウント値から転送速度を算出し、SCIに設定します。そして、SCIにより、IDフィールドを受信します。
6. IDフィールドの受信完了後、レスポンスフィールドの通信を行います。

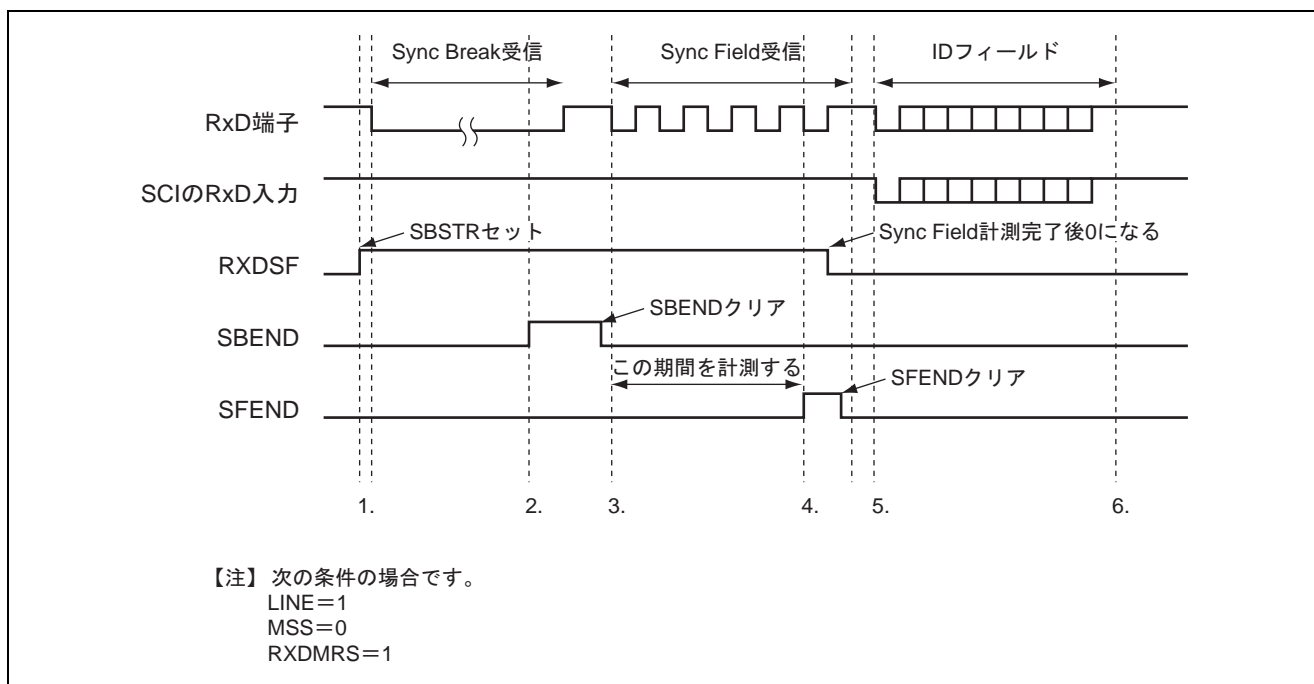


図 16.8 ヘッダフィールドの受信時の動作例

(1) スレーブモードの初期設定

スレーブモードでの初期設定例を図 16.9 に示します。

LIN 通信の前に LINCRC の LINE ビットをクリアした後、図 16.9 のフローチャートの例に従って初期設定してください。動作モードの変更の場合も必ず LINE ビットを 0 にクリアしてから変更を行ってください。

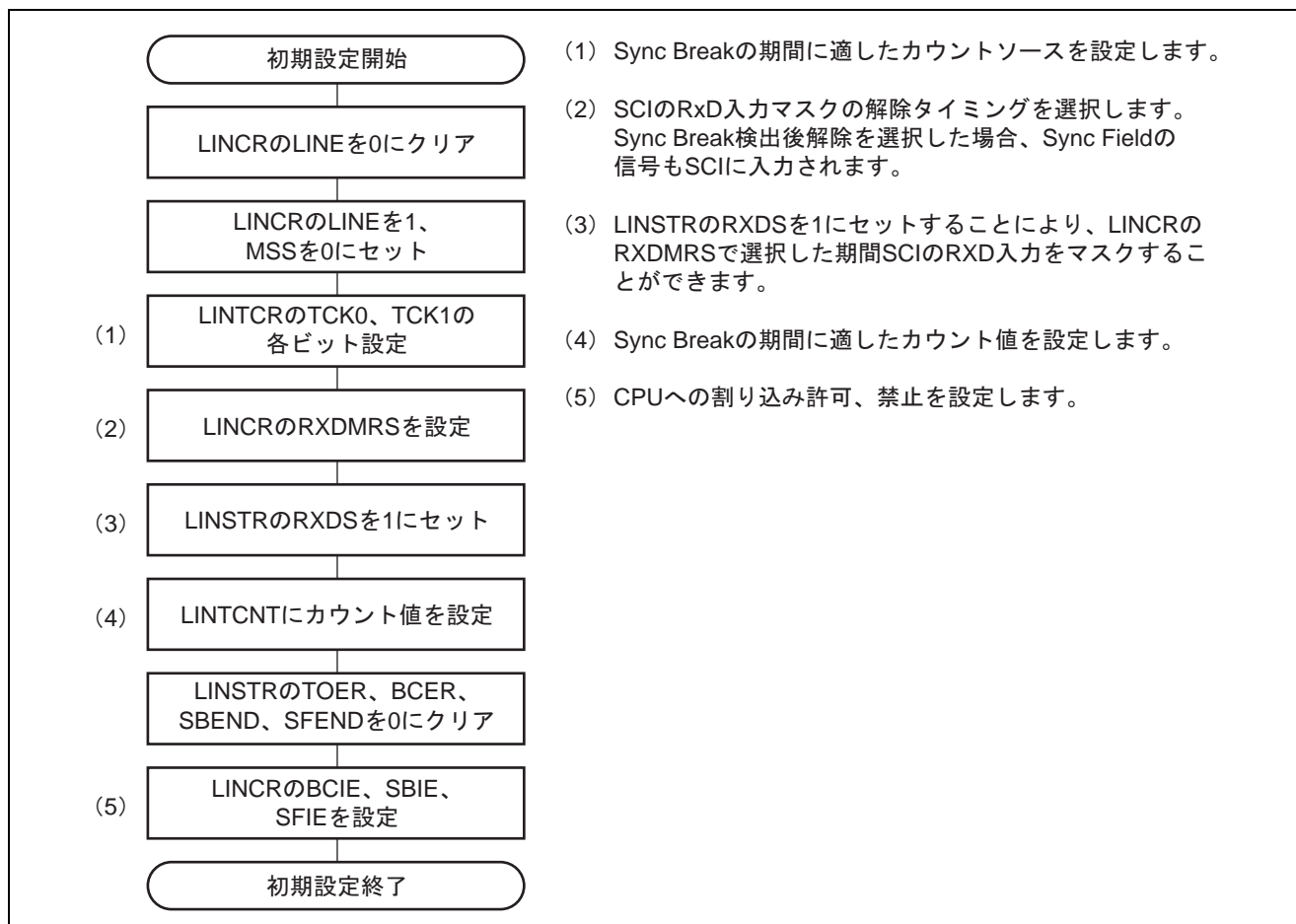


図 16.9 スレーブモードの初期設定例

(2) スレーブモードの Sync Break 受信

スレーブモードでの Sync Break 受信設定例を図 16.10 に示します。

Sync Break 受信時にハードウェア LIN は以下のように動作します。

ハードウェア LIN は LINTCR の TCSTR を 1 にセットすることによって、LINTCR の TCK1、TCK0 で選択されたタイマカウントソースに同期して Sync Break を受信できる状態に遷移します。

ダウンカウンタが動作可能状態になると自動的に LINTCR の TCSTF が 1 にセットされます。

ハードウェア LIN は LINCRC の SBSTR を 1 にセットすると自動的に SCI の RxD 入力をマスクします。RXDSF が 1 であることを確認するまでは、RxD 端子に Low レベルを入力しないでください。入力すると SCI に入力されてしまいます。

LINTCNT は Sync Break による RxD 端子に Low レベルが入力されると LINTCR の TCK1、TCK0 で選択されたタイマカウントソースに同期してダウンカウントします。LINTCNT がアンダフローすると LINSTR の SBEND が 1 にセットされます。また、LINCRC の SBIE を 1 にセットしている場合は、Sync Break 検出を割り込み要因とする割り込みが発生します。

LINSTR の SBEND が 1 にセットされたことを確認した後、LINSTR の SBEND を 0 にクリアしてください。

LINTCNT がアンダフローすると、LINTCNT は設定した値にリロードを行います。リロード後も RxD 端子

に Low レベルが入力されている場合、ダウンカウントを続けます。Sync Break 受信が終了し、RxD 端子から High レベルが入力されると、LINTCNT は設定した値にリロードし、ダウンカウントを停止します。ハードウェア LIN は Sync Field を受信できる状態に遷移します。

Sync Break による RxD 端子に入力される Low レベルの期間が LINTCNT に設定した期間に満たない場合、初めに設定した値をリロードし、次に Low レベルが入力されるまで待ちます。この場合 LINSTR の SBEND は 1 にセットされません。

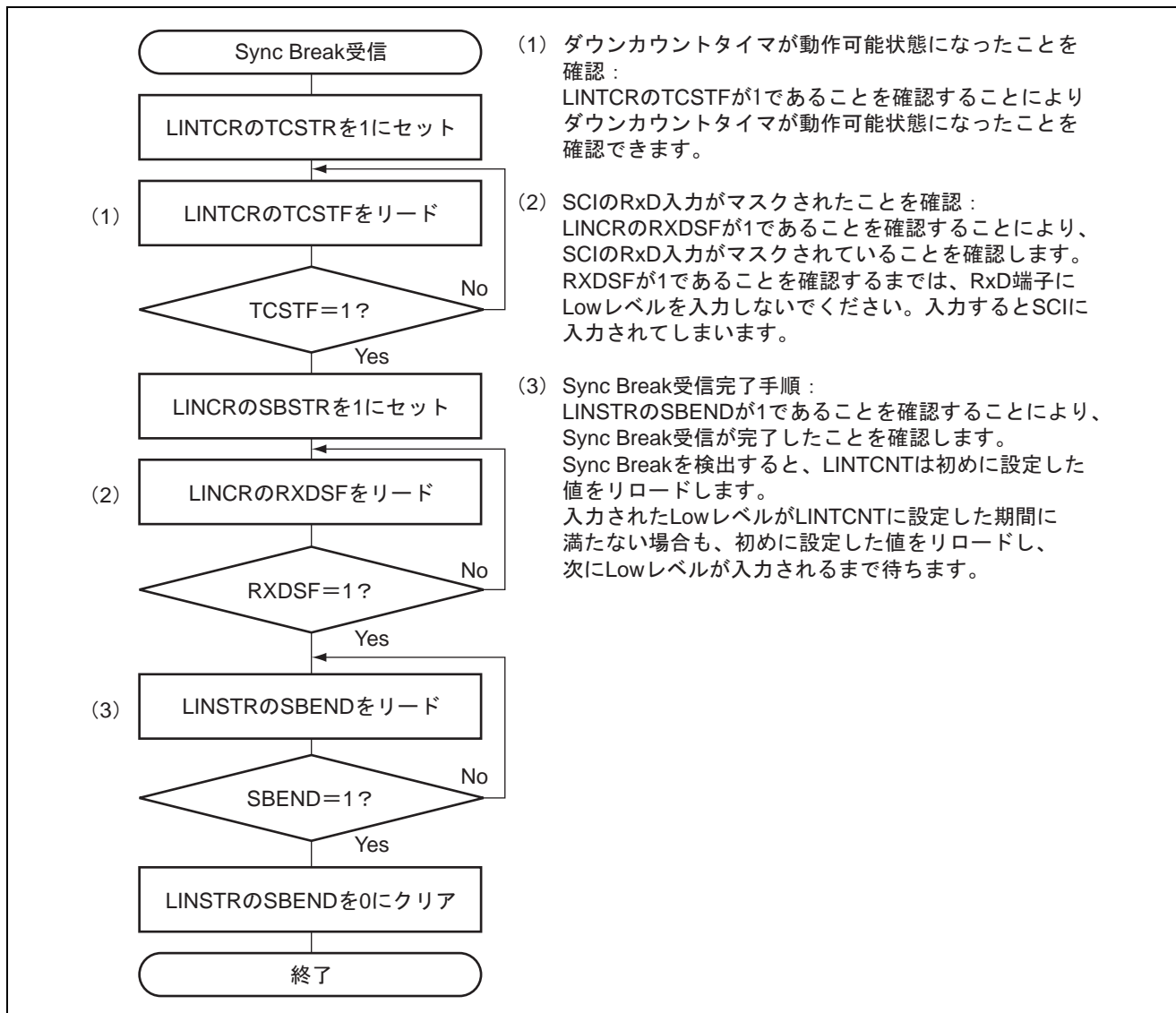


図 16.10 スレープモードの Sync Break 受信設定例

(3) スレーブモードの Sync Field 受信

スレーブモードでの Sync Field 受信設定例を図 16.11 に示します。

Sync Field 受信時にハードウェア LIN は以下のように動作します。

ハードウェア LIN は Sync Break 受信を完了し、Sync Field を受信する際、LINCRC の RXDMRS が 0 にクリアされている場合、Sync Field の受信データが SCI に入力されます。ハードウェア LIN は Sync Field のスタートビットおよび 0~6 ビットまでの期間 LINCRC の TCK1、TCK0 で選択されたタイマカウンタソースに同期して LINTCNT をダウンカウントします。

ハードウェア LIN は Sync Field のスタートビットおよび 0~6 ビットまでの期間の受信が終了すると LINSTR の SFEND が自動的に 1 にセットされます。また、LINCRC の SFIE を 1 にセットしている場合は、Sync Field 計測完了を割り込み要因とする割り込みが発生します。LINTCNT は Sync Field 計測が完了するとダウンカウントを停止します。

Sync Field のスタートビットおよび 0~6 ビットまでの期間の受信が LINTCNT で設定した期間以上の場合、LINTCNT がアンダフローし、LINSTR の SBEND が 1 にセットされます。また、LINCRC の SBIE を 1 にセットしている場合、Sync Break 検出を割り込み要因とする割り込みが発生します。

Sync Field 計測前の LINTCNT のカウンタ値と Sync Field 計測完了後の LINTCNT のカウンタ値から転送速度を算出し、SCI に設定します。LINTCNT のカウンタ値を読み出す場合、LINCRC の TCSTR を 0 にクリアする前に読み出してください。TCSTR を 0 にクリアすると、LINTCNT がリロードされます。

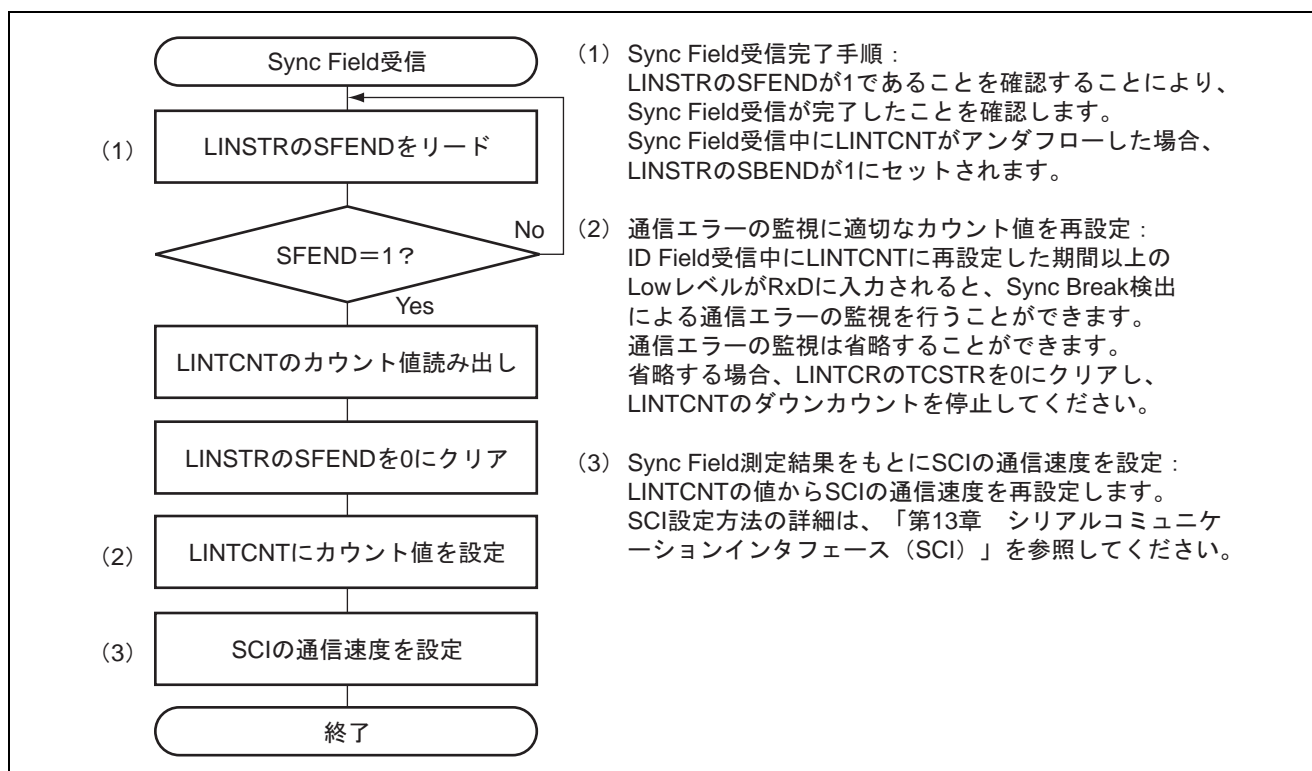


図 16.11 スレーブモードの Sync Field 受信設定例

(4) スレーブモードの ID フィールド受信

スレーブモードでの ID フィールド受信設定例を図 16.12 に示します。

ID フィールドの受信には SCI により行います。

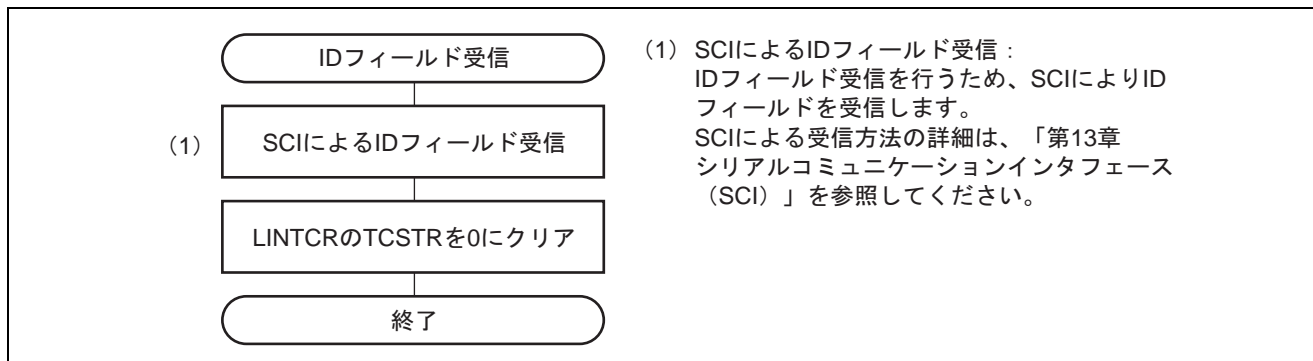


図 16.12 スレーブモードの ID フィールド受信設定例

(5) スレーブモードのレスポンスフィールド通信

スレーブモードでのレスポンスフィールド通信設定例を図 16.13 に示します。

レスポンスフィールドの通信には SCI により行います。

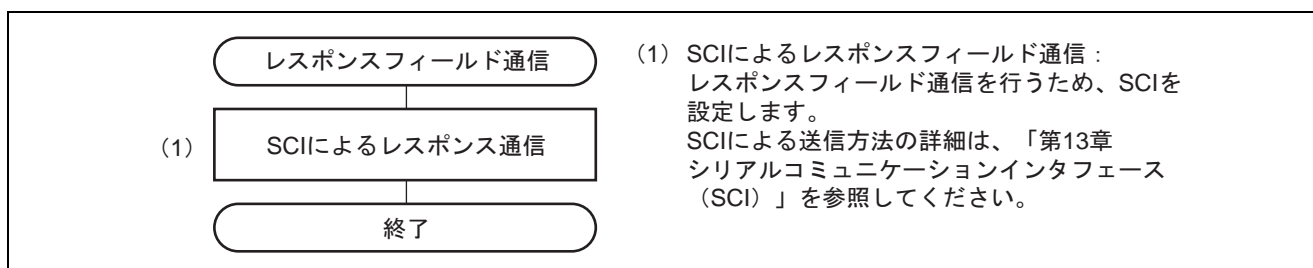


図 16.13 スレーブモードのレスポンス通信設定例

16.5 バス衝突検出機能

SCI が送信許可 (SCR の TE ビットが 1) の場合、バス衝突検出機能を使用することができます。

バス衝突検出機能を使用する場合は SCI から送信データを送信する Sync Field 送信、ID フィールドの送信、レスポンスフィールドの送信のときのみ LINSTR の BCE を 1 にセットし使用してください。それ以外の Sync Break 送信、Sync Break 受信、Sync Field 受信、ID フィールドの受信、レスポンスフィールドの受信のときにはバス衝突検出機能は使用できません。LINSTR の BCE を 0 にクリアし、バス衝突検出機能を停止してください。

図 16.14 にバス衝突検出時の動作例を示します。

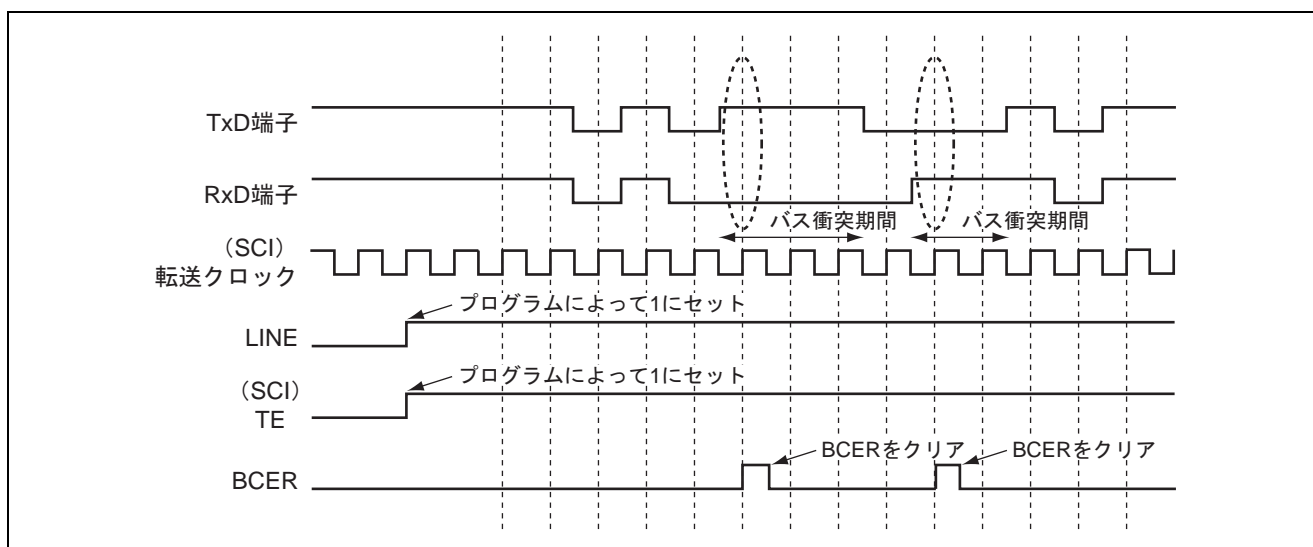


図 16.14 バス衝突検出時の動作例

バス衝突検出はバス衝突期間の最初の SCI の転送クロックの立ち上がりエッジで検出します。

ハードウェア LIN は、バス衝突を検出すると LINSTR の BCER が 1 にセットされます。また、LINCR の BCIE を 1 にセットしている場合は、バス衝突検出を割り込み要因とする割り込みが発生します。

LINSTR の BCER が 1 にセットされたことを確認した後、BCER を 0 にクリアしてください。

16.6 タイムアウト検出機能

タイムアウト検出機能はタイムアウト計測期間以内にフレームが完了しなかったときにエラーと判定する機能です。タイムアウト検出機能を使用する場合、マスタモード、スレーブモード共に Sync Break 完了もしくは Sync Break 検出の割り込みを起点に、ソフトウェアにより LINTCNT による計測を開始します。フレームが完了するまで LINTCNT による計測を継続し、フレーム完了と共に LINTCNT の計測を停止させます。LINTCNT がアンダフローすると LINSTR の TOER が 1 にセットされます。また、LINSTR の TOIE を 1 にセットしている場合は、タイムアウト検出を割り込み要因とする割り込みが発生します。

16.6.1 マスタモードでのタイムアウト検出

図 16.15 にマスタモード時にハードウェア LIN がタイムアウト計測を行うフレームの領域を示します。

LINTCNT のカウントは LINTCR の TOCSTR を 1 にセットすることによって、LINTCR の TCK1、TCK0 で選択されたタイマカウンタソースに同期してダウンカウントします。LINTCNT のカウント値には Sync Break 完了からフレームが完了するまでに必要なカウント値を設定してください。

マスタモード時にタイムアウト検出機能を使用する場合は Sync Break 完了の割り込みを検出後、LINTCR の TOCSTR を 1 にセットし LINTCNT による計測を開始してください。フレームが完了後、LINTCR の TOCSTR を 0 にクリアし LINTCNT の計測を停止させてください。TOCSTR を 0 にクリアしても LINTCNT はリロードされませんので注意してください。LINTCNT がアンダフローすると LINSTR の TOER が 1 にセットされます。また、LINSTR の TOIE を 1 にセットしている場合は、タイムアウト検出を割り込み要因とする割り込みが発生します。LINTCNT はアンダフローすると、設定した値にリロードし、リロード後も TOCSTR が 1 にセットされている場合、リロード後の値からダウンカウントを続けます。

また、マスタモード時は LINTCNT に設定する値を Sync Break 開始からフレームが完了するまでに必要なカウント値を設定し、Sync Break 送信開始直後に LINTCR の TOCSTR を 1 にセットすることで、Sync Break 送信開始からフレーム完了までをタイムアウトエラー検出対象とすることができます。

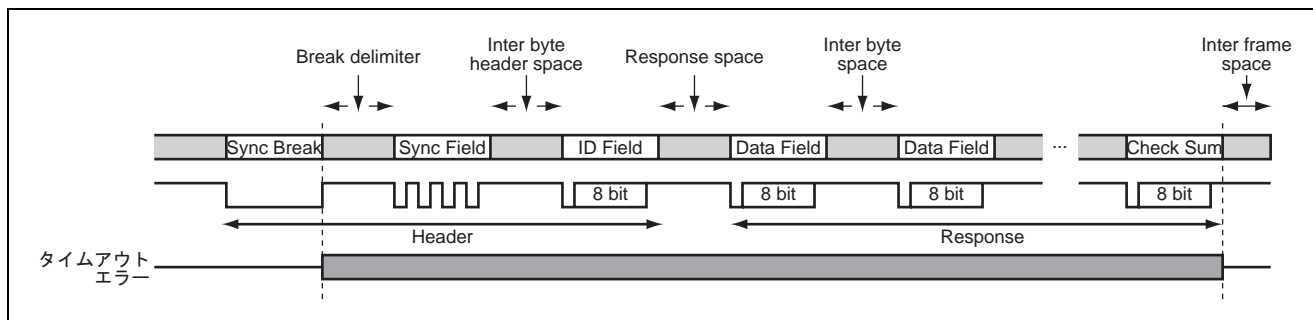


図 16.15 マスタモード時のタイムアウトエラー検出対象

16.6.2 スレーブモードでのタイムアウト検出

図 16.16 にスレーブモードでのハードウェア LIN がタイムアウト計測を行うフレームの領域を示します。

LINTCNT のカウントは LINTCR の TOCSTR を 1 にセットすることによって、LINTCR の TCK1、TCK0 で選択されたタイマカウントソースに同期してダウンカウントします。LINTCNT のカウント値には Sync Break 検出からフレームが完了するまでに必要なカウント値を設定してください。

スレーブモード時にタイムアウト検出機能を使用する場合は Sync Break 検出の割り込みを検出後、LINTCR の TOCSTR を 1 にセットし LINTCNT による計測を開始してください。フレームが完了後、LINTCR の TOCSTR を 0 にクリアし LINTCNT の計測を停止させてください。TOCSTR を 0 にクリアしても LINTCNT はリロードされませんので注意してください。LINTCNT がアンダフローすると LINSTR の TOER が 1 にセットされます。また、LINSTR の TOIE を 1 にセットしている場合は、タイムアウト検出を割り込み要因とする割り込みが発生します。LINTCNT はアンダフローすると、設定した値にリロードし、リロード後も TOCSTR が 1 にセットされている場合、リロード後の値からダウンカウントを続けます。

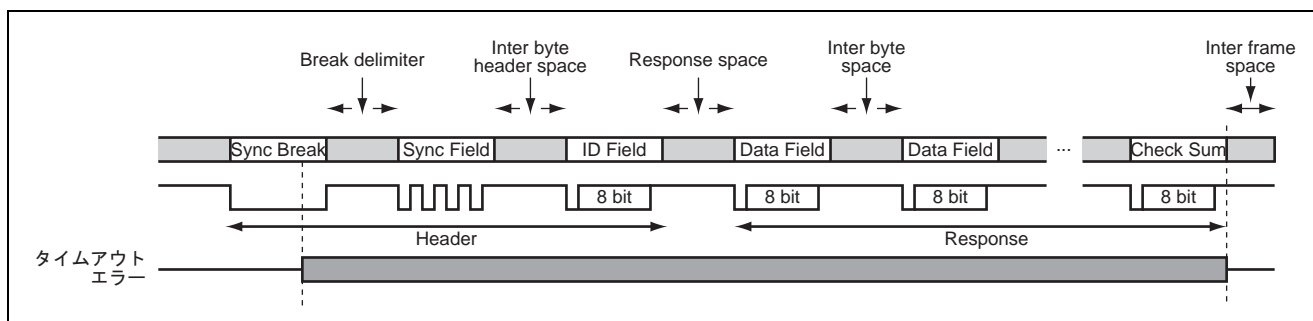


図 16.16 スレーブモード時のタイムアウトエラー検出対象

16.7 ウェイクアップ検出方法

ハードウェア LIN の RxD は、IRQn 入力と端子を兼用しています。ウェイクアップの検出は、RxD 入力の立ち下がりエッジを IRQn による割り込みで検出してください。また、ウェイクアップの送信には SCI を使用してください。

IRQn による割り込みの詳細は「第 5 章 割り込みコントローラ」を参照してください。

16.8 割り込み要求

マスタモードでハードウェア LIN が生成する割り込みの要因は 3 種類あり、スレーブモードでハードウェア LIN が生成する割り込み要因は 4 種類あります。表 16.7 にマスタモードでの割り込み要因の内容を示します。表 16.8 にスレーブモードでの割り込み要因の内容を示します。SBIE、SFIE、BCIE、TOIE が 1 にセットされている場合、SBEND、SFEND、BCER、TOER のいずれかが 1 にセットされると割り込みが発生します。

表 16.7 マスタモードでの割り込み要因

割り込み要因	ステータスフラグ	割り込み発生条件
Sync Break 完了	SBEND	Sync Break 出力の期間を完了し、LINTCNT がアンダフローしたとき
バス衝突検出	BCER	SCI が送信許可のとき、データラッチタイミングで RxD 入力と TxD 出力の値が異なったとき
タイムアウト検出	TOER	LINTCNT がアンダフローしたとき

表 16.8 スレーブモードでの割り込み要因

割り込み要因	ステータスフラグ	割り込み発生条件
Sync Break 検出	SBEND	RxD 入力の Low レベルの期間を計測し、LINTCNT がアンダフローしたとき
Sync Field 計測完了	SFEND	Sync Field の 8 ビット分の計測が完了したとき
バス衝突検出	BCER	SCI が送信許可のとき、データラッチタイミングで RxD 入力と TxD 出力の値が異なったとき
タイムアウト検出	TOER	LINTCNT がアンダフローしたとき

16.9 使用上の注意事項

16.9.1 モジュールストップモードの設定

モジュールストップコントロールレジスタにより、HWLIN の動作禁止/許可を設定することが可能です。初期値では、HWLIN の動作は停止します。モジュールストップモードを解除することにより、レジスタのアクセスが可能になります。詳細は、「第 23 章 低消費電力」を参照してください。

16.9.2 LIN タイマカウンタ (LINTCNT) のライトとカウントダウンの競合

LINTCNT のライトサイクル中の T2 ステートでカウントダウンが発生しても、カウントダウンされずに LINTCNT へのカウンタライトが優先されます。これを図 16.17 に示します。

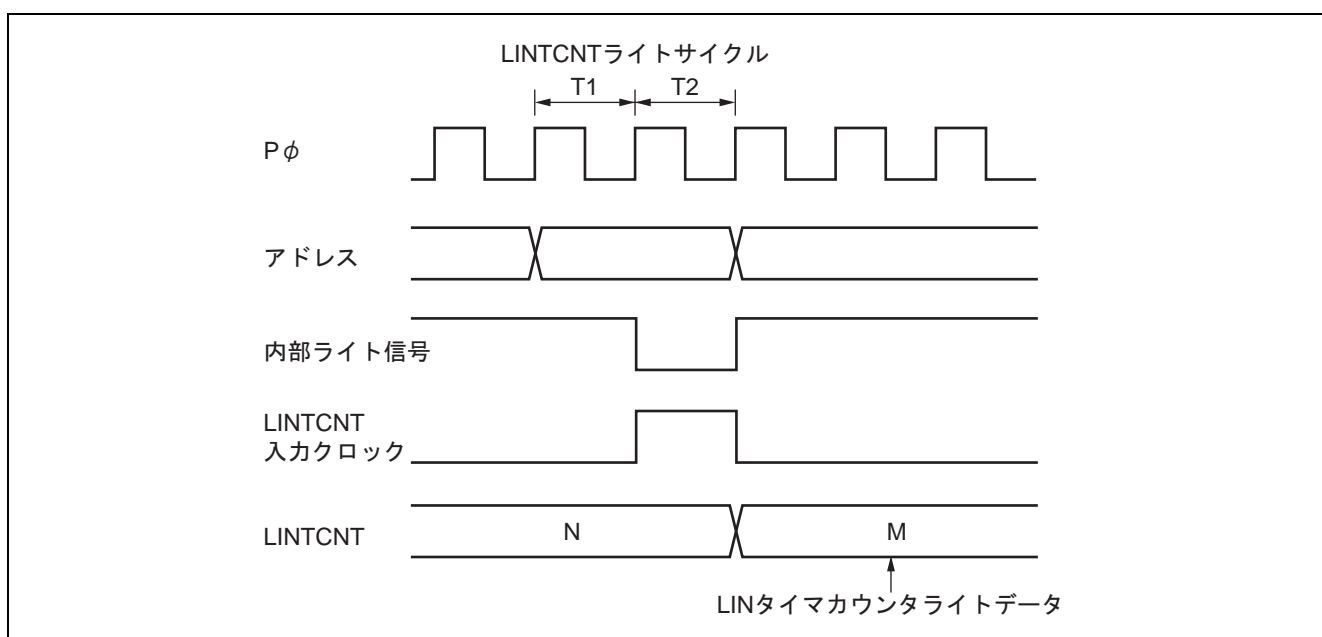


図 16.17 LINTCNT のライトとカウントダウンの競合

16.9.3 LIN タイムアウトカウンタ (LINTOCNT) のライトとカウントダウンの競合

LINTOCNT のライトサイクル中の T2 ステートでカウントダウンが発生しても、カウントダウンされずに LINTOCNT へのカウンタライトが優先されます。これを図 16.18 に示します。

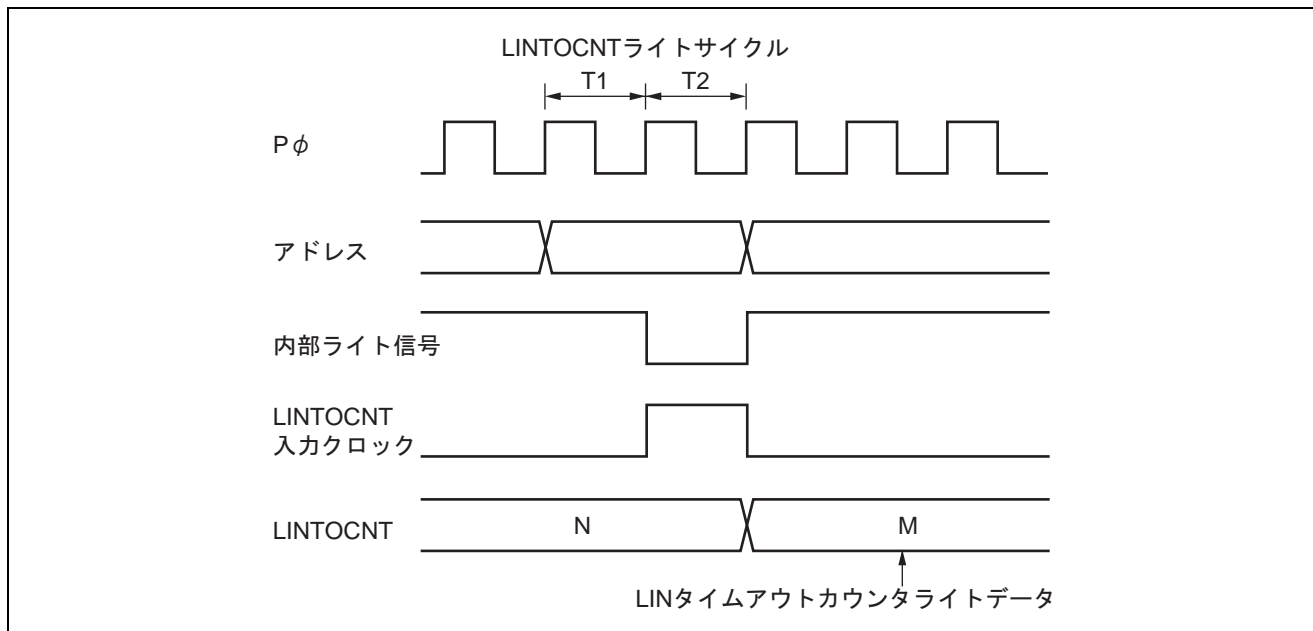


図 16.18 LINTOCNT のライトとカウントダウンの競合

16.9.4 LIN タイマカウンタ (LINTCNT) のライトとアンダフローの競合

LINTCNT のライトサイクル中の T2 ステートでカウントダウンが発生しアンダフローが発生した場合、LINTCNT はリロードされずに LINTCNT へのカウンタライトが優先されます。LINSTR の SBEND フラグはセットされます。これを図 16.19 に示します。

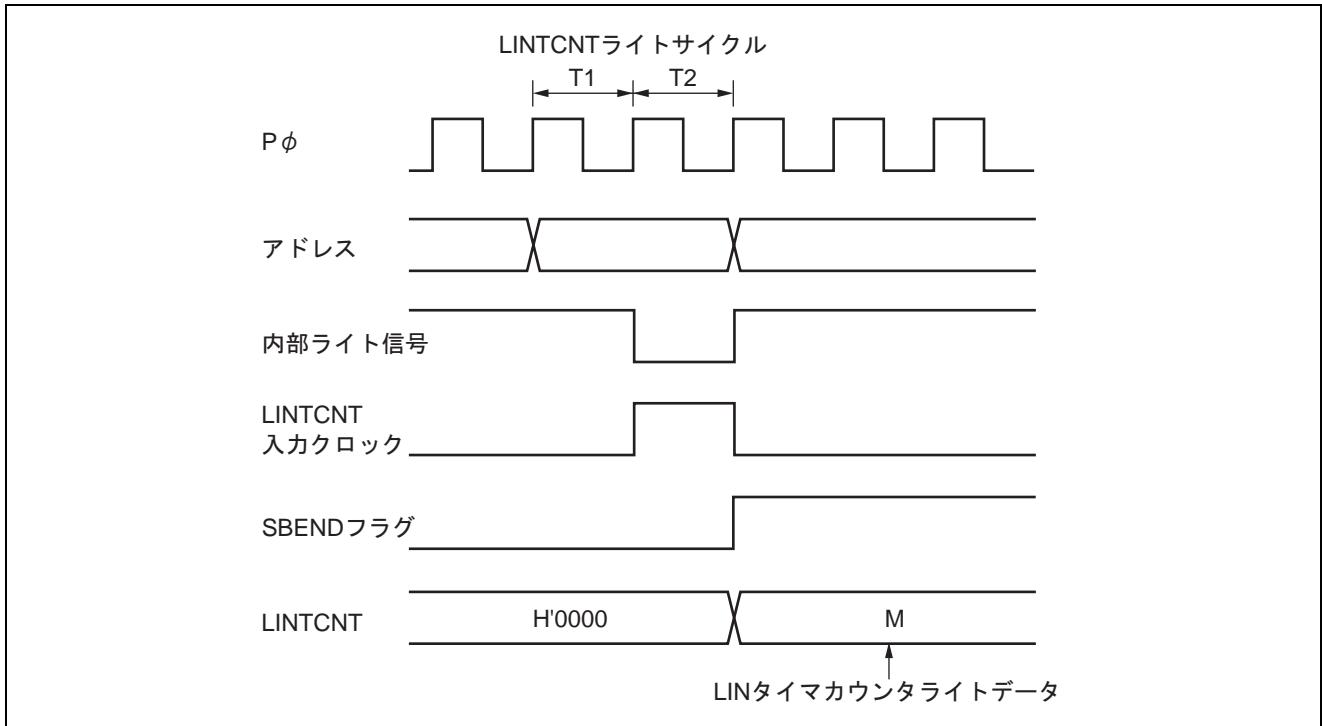


図 16.19 LINTCNT のライトとアンダフローの競合

16.9.5 LIN タイムアウトカウンタ (LINTOCNT) のライトとアンダフローの競合

LINTOCNT のライトサイクル中の T2 ステートでカウントダウンが発生しアンダフローが発生した場合、LINTOCNT はリロードされずに LINTOCNT へのカウンタライトが優先されます。LINSTR の TOER フラグはセットされます。これを図 16.20 に示します。

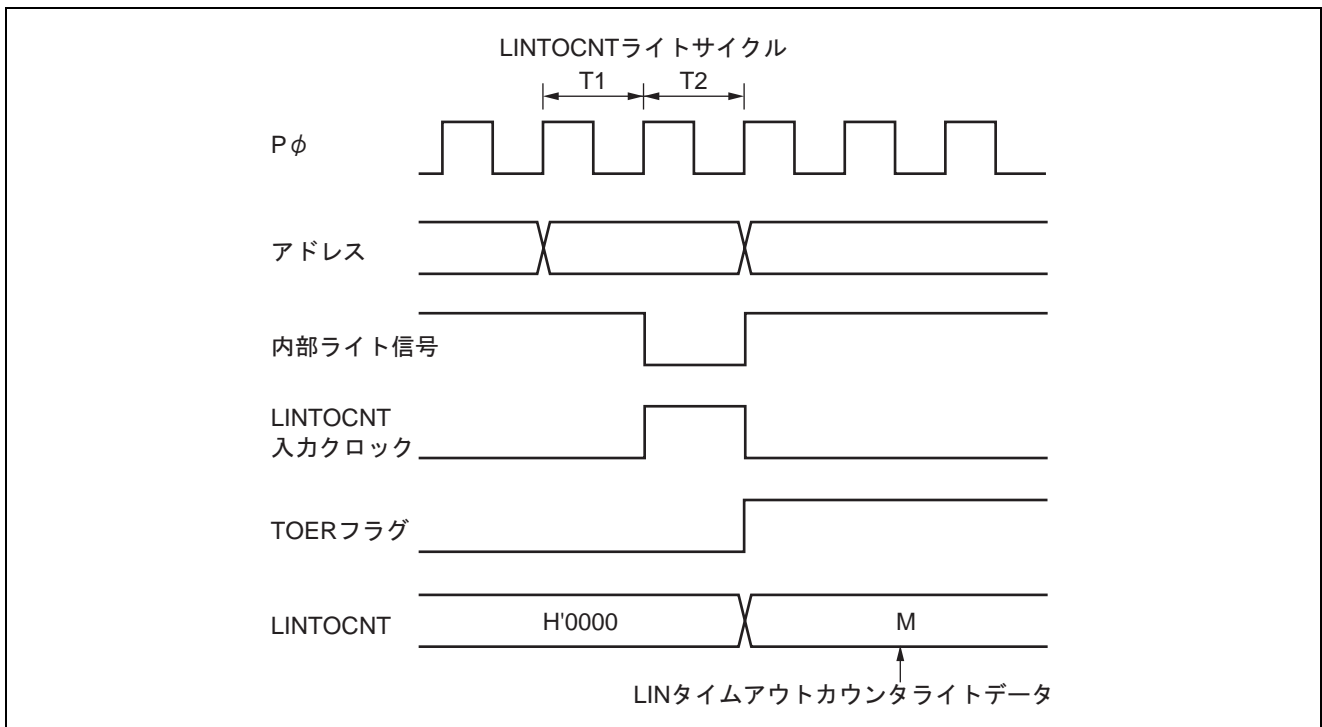


図 16.20 LINTOCNT のライトとアンダフローの競合

17. CRC 演算器

CRC (Cyclic Redundancy Check) 演算器は、データブロックの誤り検出を行います。

17.1 特長

- 8ビット単位の任意のデータ長に対してCRCコードを生成
- CRC演算は8ビットずつ並列に実行
- 生成多項式を3つの多項式から選択可能
- LSBファースト通信用CRCコード生成/MSBファースト通信用CRCコード生成の選択が可能

図 17.1 に CRC 演算器のブロック図を示します。

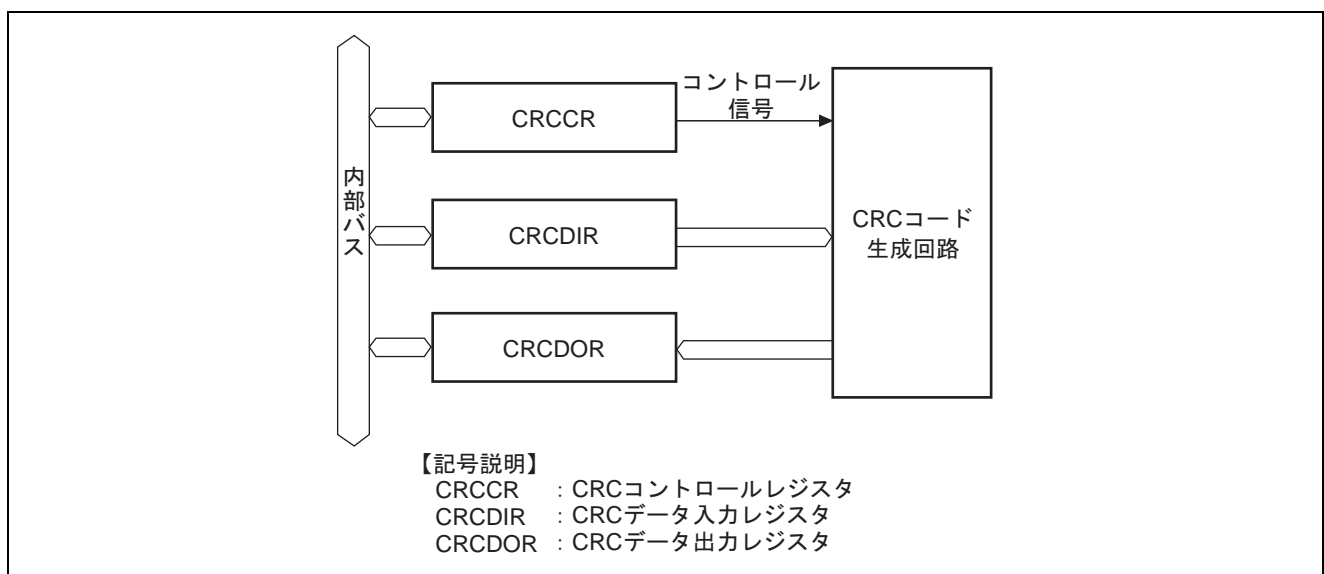


図 17.1 CRC 演算器のブロック図

17.2 レジスタの説明

CRC 演算器には以下のレジスタがあります。

- CRCコントロールレジスタ (CRCCR)
- CRCデータ入力レジスタ (CRCDIR)
- CRCデータ出力レジスタ (CRCDOR)

17.2.1 CRC コントロールレジスタ (CRCCR)

CRCCR は CRC 演算器の初期化、演算切り替え、生成多項式を選択します。

ビット	7	6	5	4	3	2	1	0
ビット名	DORCLR	—	—	—	—	LMS	G1	G0
初期値:	0	0	0	0	0	0	0	0
R/W:	W	R	R	R	R	R/W	R/W	R/W

ビット	ビット名	初期値	R/W	説明
7	DORCLR	0	W	CRCDOR クリア このビットを 1 にセットすると、CRCDOR が H'0000 にクリアされます。
6~3	—	0	R	リザーブビット 初期値を変更しないでください。
2	LMS	0	R/W	CRC 演算切り替え LSB ファースト通信用 CRC コード生成か、MSB ファースト通信用 CRC コード生成かを選択します。 0: LSB ファーストで通信する場合の CRC 演算を行います。CRCDOR の内容 (CRC コード) を 2 バイトに分けて送信する場合、下位バイト (ビット 7 ~0) を先に送信します。 1: MSB ファーストで通信する場合の CRC 演算を行います。CRCDOR の内容 (CRC コード) を 2 バイトに分けて送信する場合、上位バイト (ビット 15 ~8) を先に送信します。
1 0	G1 G0	0 0	R/W R/W	CRC 生成多項式切り替え 多項式を選択します。 00: リザーブ 01: $X^8 + X^2 + X + 1$ 10: $X^{16} + X^{15} + X^2 + 1$ 11: $X^{16} + X^{12} + X^5 + 1$

17.2.2 CRC データ入力レジスタ (CRCDIR)

CRCDIR は 8 ビットのリード/ライト可能なレジスタです。CRCDIR に CRC 演算対象のバイトをライトすると CRCDOR に結果が得られます。

ビット	7	6	5	4	3	2	1	0
ビット名								
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

17.2.3 CRC データ出力レジスタ (CRCDOR)

CRCDOR は 16 ビットのリード/ライト可能なレジスタです。CRCDOR クリア後、CRCDIR に CRC 演算対象のバイトをライトすると CRCDOR に結果が得られます。CRC 演算対象のバイトに CRC 演算結果を追加してライトした場合、CRC エラーがなければ結果は H'0000 になります。CRCCR ビット 1、0 を G1=0、G0=1 と指定した場合、下位バイトに結果が得られます。

ビット	7	6	5	4	3	2	1	0
ビット名								
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
ビット	7	6	5	4	3	2	1	0
ビット名								
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

17.3 CRC 演算器の動作説明

CRC 演算器は、LSB ファースト/MSB ファースト通信用 CRC コードを生成します。以下に CRCCR の G1、G0 ビットを B'11 として $X^{16}+X^{12}+X^5+1$ の多項式を使用し、16 進数 H'F0 データについて CRC コードを生成する使用例を示します。

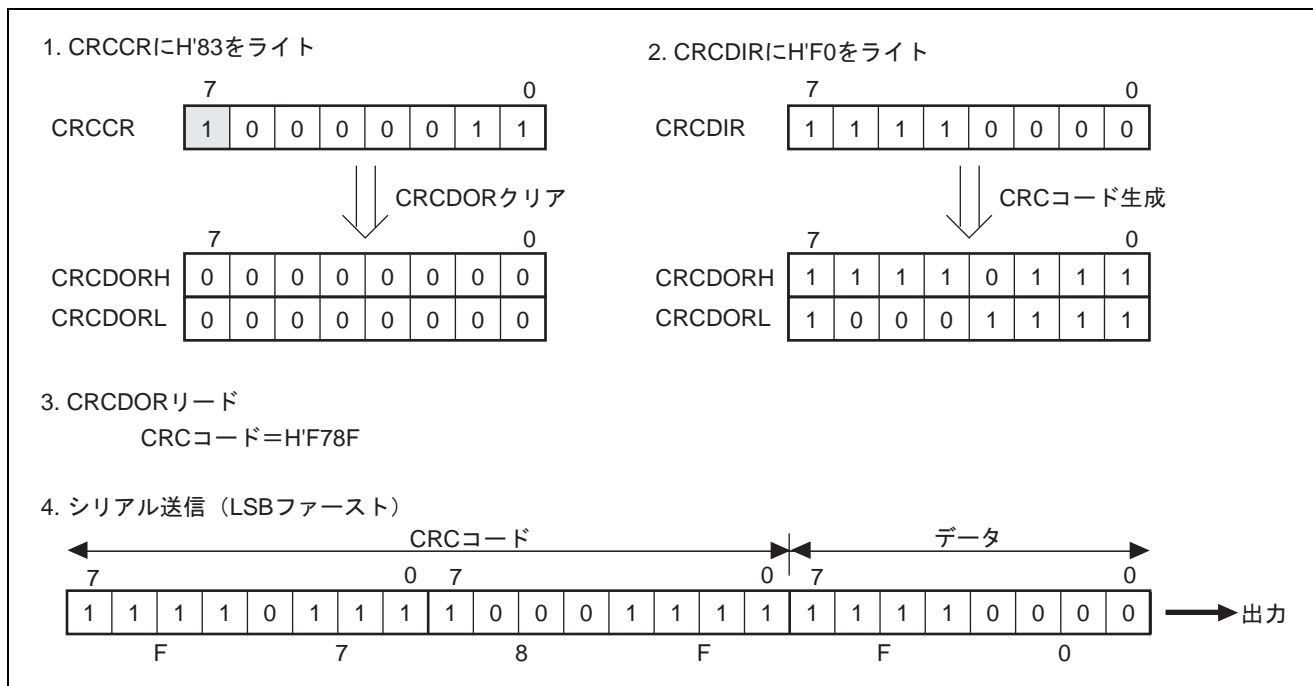


図 17.2 LSB ファーストでのデータ送信

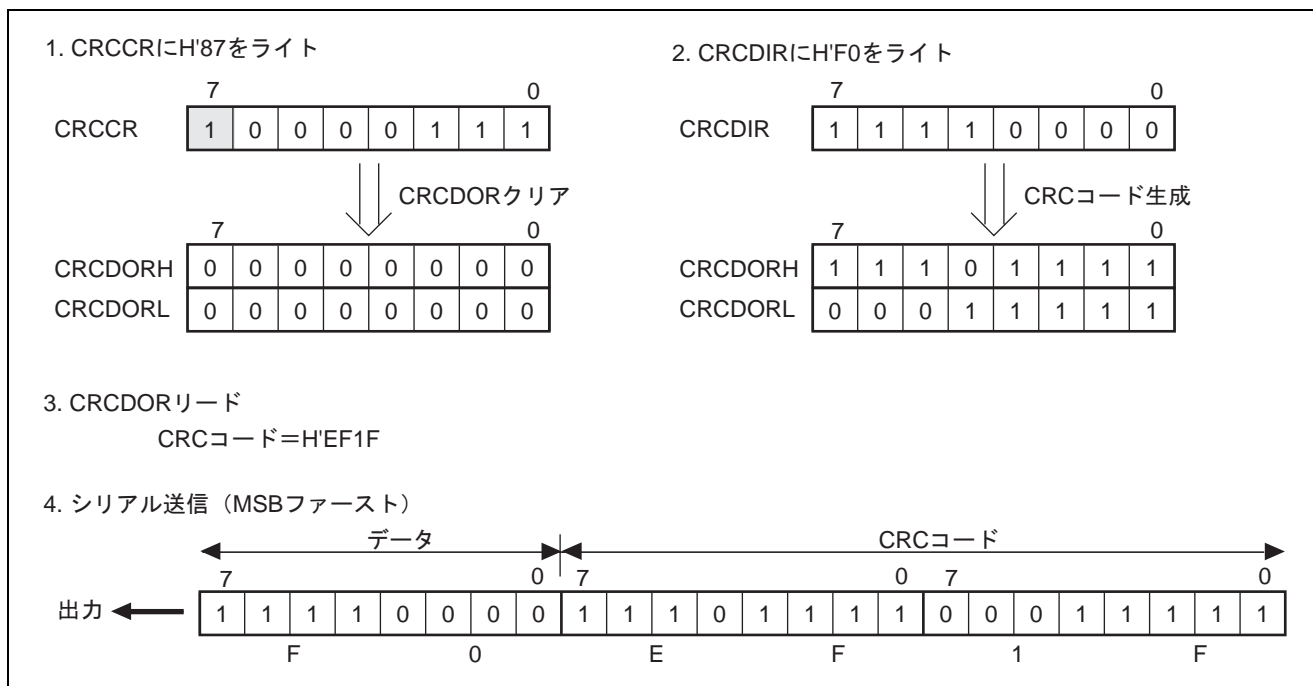


図 17.3 MSB ファーストでのデータ送信

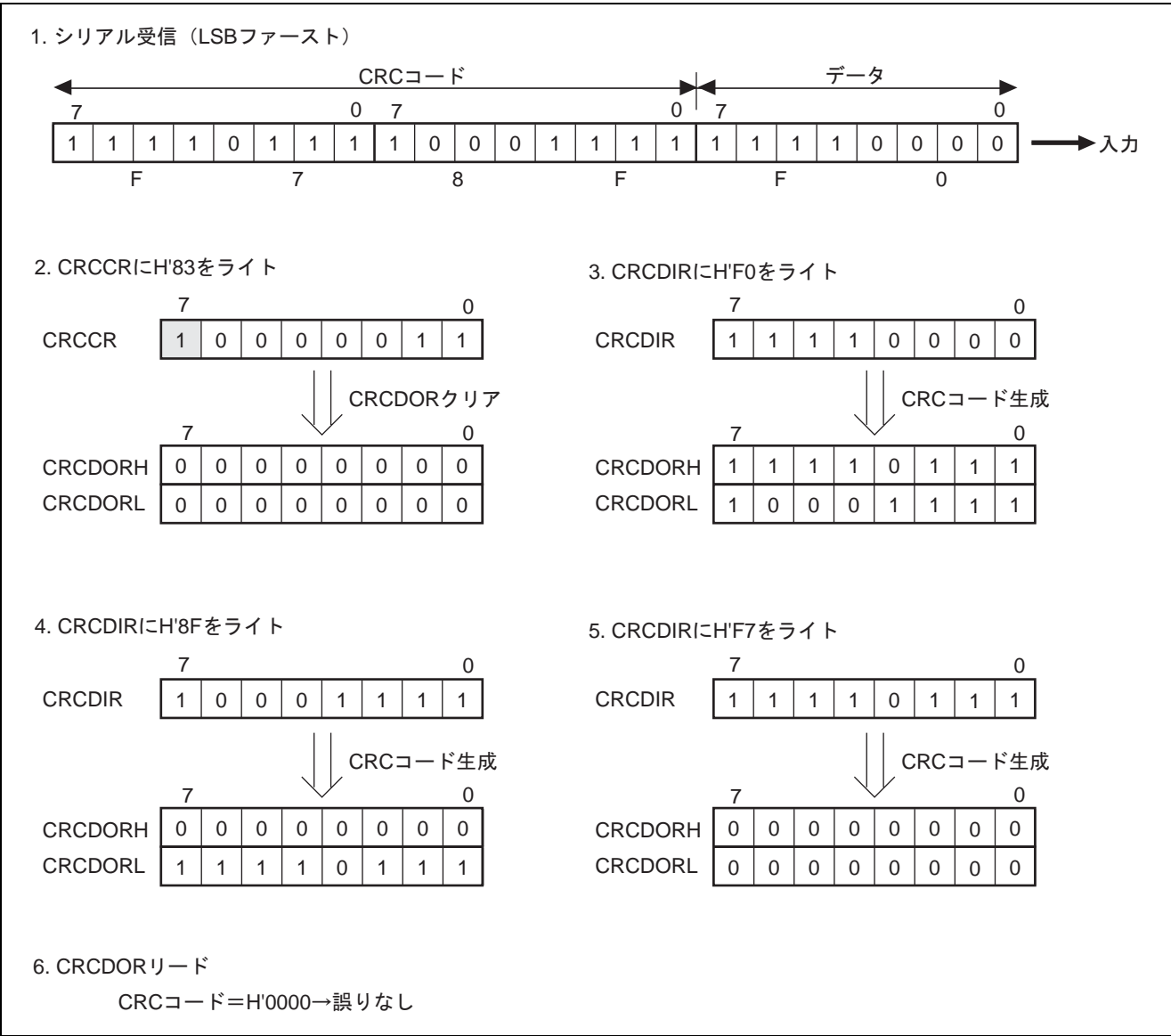


図 17.4 LSB ファーストでのデータ受信

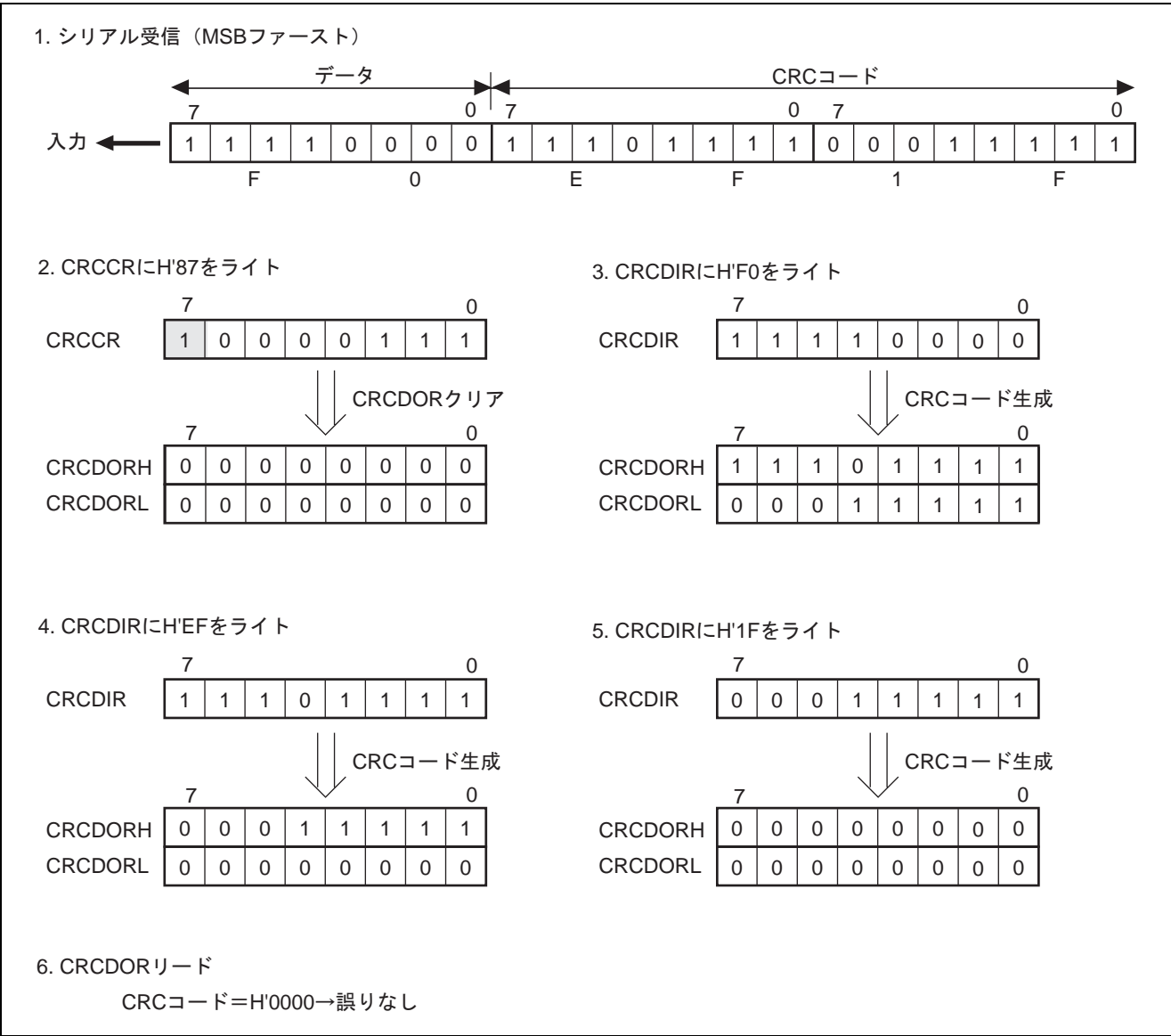


図 17.5 MSB ファーストでのデータ受信

17.4 CRC 演算器使用上の注意事項

LSB ファーストで送信する場合と MSB ファーストで送信する場合とは、CRC コードを送る順序が異なりますので、注意してください。

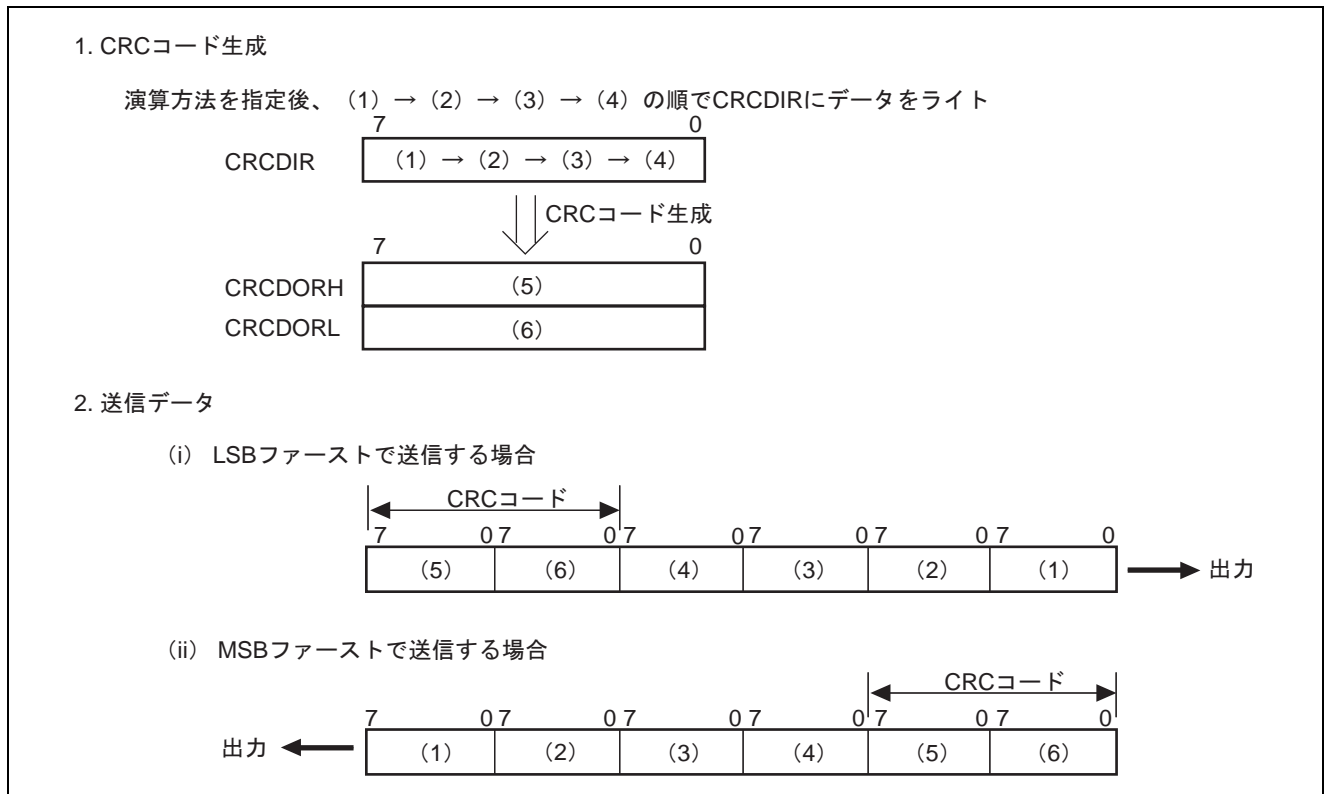


図 17.6 LSB ファーストと MSB ファーストの送信データ

18. A/D 変換器

本 LSI は、逐次比較方式の 10 ビットの A/D 変換器を 2 ユニット (ユニット 0、ユニット 1) 内蔵しています。各ユニットとも最大 8 チャンネルのアナログ入力を選択することができます。

ユニット 0 のブロック図を図 18.1 に、ユニット 1 のブロック図を図 18.2 に示します。

本文中は、ユニット 0 について説明します。ユニット 1 についても同一機能です。

18.1 特長

- 分解能：10ビット
- 入力チャンネル：8チャンネル (16チャンネル/2ユニット合計)
- 変換サイクル：40サイクル (A/D変換用クロック)
- 動作モード：2種類
 - シングルモード：1チャンネルの A/D 変換
 - スキャンモード：1~4チャンネルの連続 A/D 変換、または 1~8チャンネルの連続 A/D 変換
- A/D変換用クロックを個別に設定可能 (A ϕ 、A ϕ /2、A ϕ /4、A ϕ /8)
- データレジスタ：8本 (16本/2ユニット)
A/D 変換結果は各チャンネルに対応した 16 ビットデータレジスタに保持
- サンプル&ホールド機能付き
- A/D変換開始方法：3種類
 - ソフトウェア
 - 16 ビットタイムパルスユニット (TPU) による変換開始トリガ
 - 外部トリガ信号
- 割り込み要因
A/D 変換終了割り込み要求 (ADI) を発生
- モジュールストップモードの設定可能
- アナログポートのプルダウン機能により、アナログ入力信号の断線チェックが可能
- A/D変換器の自己診断が可能

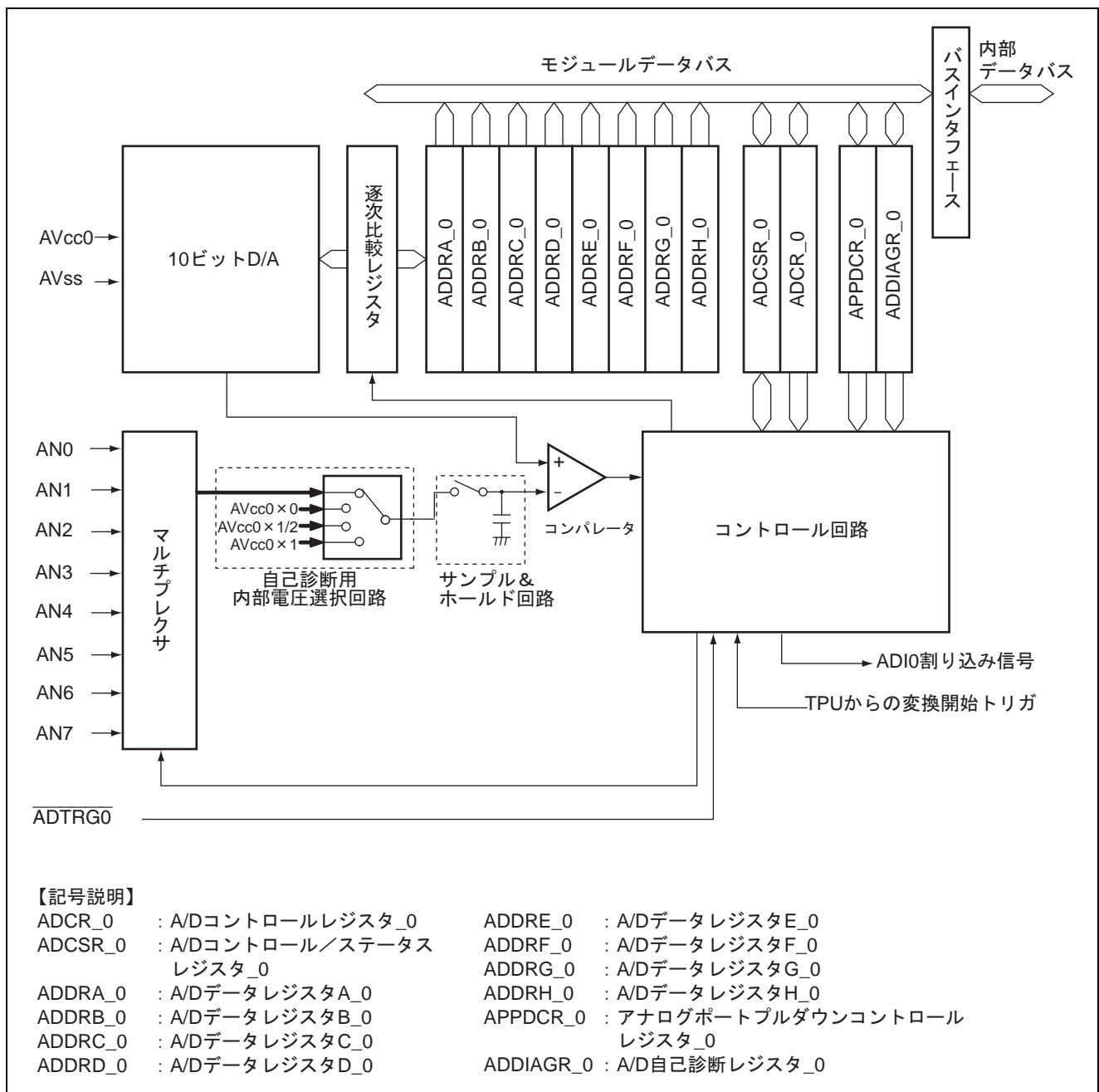


図 18.1 A/D 変換器 (ユニット 0/AD_0) のブロック図

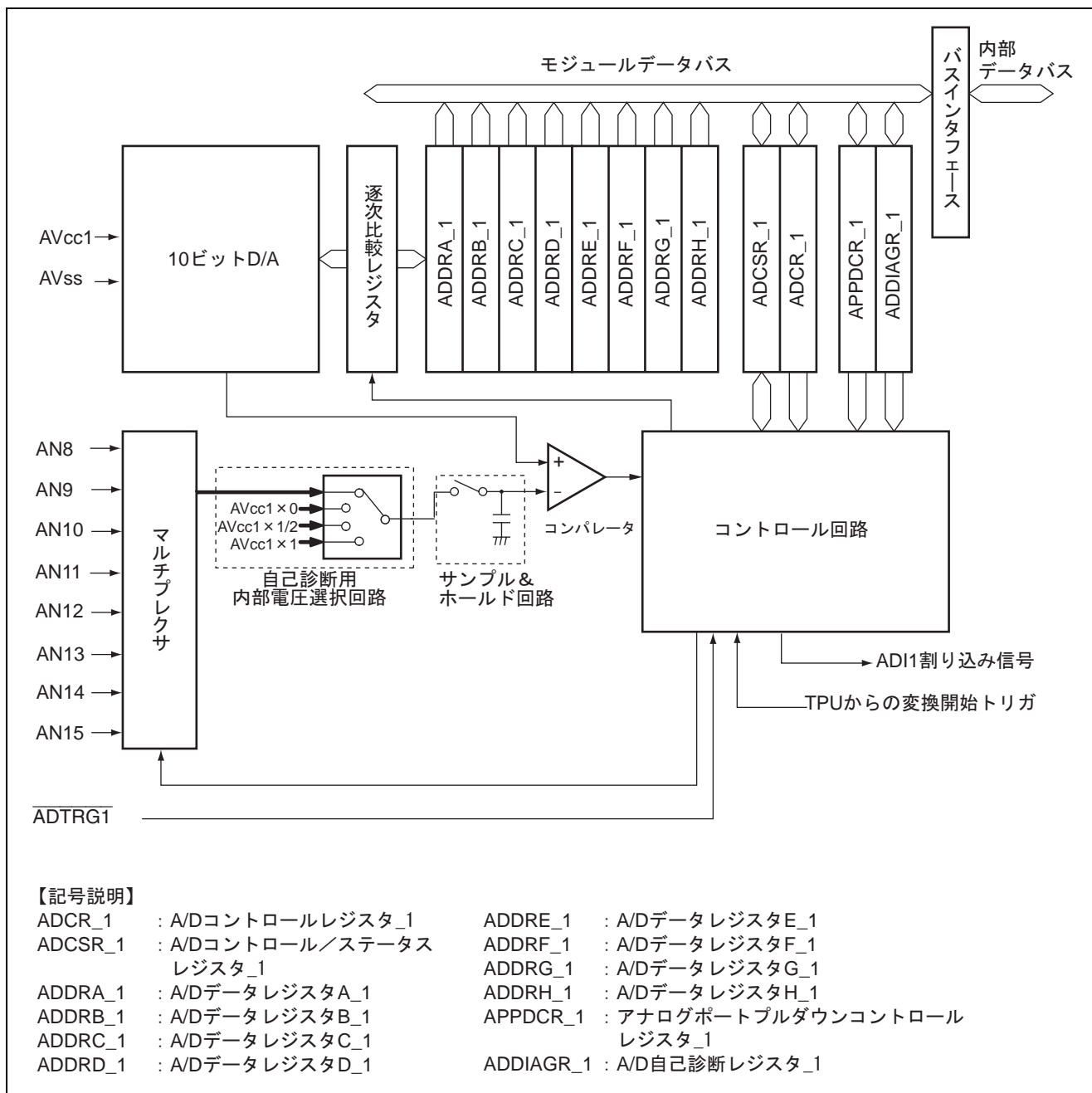


図 18.2 A/D 変換器 (ユニット 1/AD_1) のブロック図

18.2 入出力端子

A/D 変換器で使用する入力端子を表 18.1 に示します。

表 18.1 端子構成

ユニット	略称	端子名	記号	入出力	機能
0	AD_0	アナログ入力端子 0	AN0	入力	アナログ入力端子
		アナログ入力端子 1	AN1	入力	
		アナログ入力端子 2	AN2	入力	
		アナログ入力端子 3	AN3	入力	
		アナログ入力端子 4	AN4	入力	
		アナログ入力端子 5	AN5	入力	
		アナログ入力端子 6	AN6	入力	
		アナログ入力端子 7	AN7	入力	
		A/D 外部トリガ入力端子 0	$\overline{\text{ADTRG0}}$	入力	A/D 変換開始のための外部トリガ入力端子
		アナログ電源端子 0	AVcc0	入力	アナログ部の電源端子および基準電圧端子
1	AD_1	アナログ入力端子 8	AN8	入力	アナログ入力端子
		アナログ入力端子 9	AN9	入力	
		アナログ入力端子 10	AN10	入力	
		アナログ入力端子 11	AN11	入力	
		アナログ入力端子 12	AN12	入力	
		アナログ入力端子 13	AN13	入力	
		アナログ入力端子 14	AN14	入力	
		アナログ入力端子 15	AN15	入力	
		A/D 外部トリガ入力端子 1	$\overline{\text{ADTRG1}}$	入力	A/D 変換開始のための外部トリガ入力端子
		アナログ電源端子 1	Avcc1	入力	アナログ部の電源端子および基準電圧端子
共通		アナロググランド端子	AVss	入力	アナログ部のグランド端子

18.3 レジスタの説明

A/D 変換器には以下のレジスタがあります。

ユニット0 (A/D_0)、ユニット1 (A/D_1) とともに同一機能です。以下のレジスタ説明では AN8～AN15 は、AN0～AN7 にそれぞれ対応しています。

1. ユニット0 (A/D_0)

- A/DデータレジスタA_0 (ADDRA_0)
- A/DデータレジスタB_0 (ADDRB_0)
- A/DデータレジスタC_0 (ADDRC_0)
- A/DデータレジスタD_0 (ADDRD_0)
- A/DデータレジスタE_0 (ADDRE_0)
- A/DデータレジスタF_0 (ADDRF_0)
- A/DデータレジスタG_0 (ADDRG_0)
- A/DデータレジスタH_0 (ADDRH_0)
- A/Dコントロール/ステータスレジスタ_0 (ADCSR_0)
- A/Dコントロールレジスタ_0 (ADCR_0)
- アナログポートプルダウンコントロールレジスタ_0 (APPDCR_0)
- A/D自己診断レジスタ_0 (ADDIAGR_0)

2. ユニット1 (A/D_1)

- A/DデータレジスタA_1 (ADDRA_1)
- A/DデータレジスタB_1 (ADDRB_1)
- A/DデータレジスタC_1 (ADDRC_1)
- A/DデータレジスタD_1 (ADDRD_1)
- A/DデータレジスタE_1 (ADDRE_1)
- A/DデータレジスタF_1 (ADDRF_1)
- A/DデータレジスタG_1 (ADDRG_1)
- A/DデータレジスタH_1 (ADDRH_1)
- A/Dコントロール/ステータスレジスタ_1 (ADCSR_1)
- A/Dコントロールレジスタ_1 (ADCR_1)
- アナログポートプルダウンコントロールレジスタ_1 (APPDCR_1)
- A/D自己診断レジスタ_1 (ADDIAGR_1)

18.3.1 A/D データレジスタ A~H (ADDRA~ADDRH)

ADDR は、A/D 変換された結果を格納するための 16 ビットのリード専用レジスタです。ADDRA~ADDRH の 8 本あります。各アナログ入力チャネルの変換結果が格納される ADDR は表 18.2 のとおりです。

10 ビットの変換データは ADDR のビット 15 からビット 6 に格納されます。下位 6 ビットはリードすると常に 0 がリードされます。

CPU との間のデータバスは 16 ビット幅で、CPU から直接リードできます。ADDR をアクセスする場合は、16 ビット単位でアクセスしてください。8 ビット単位でのアクセスは禁止です。

ビット	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ビット名											—	—	—	—	—	—
初期値:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

表 18.2 アナログ入力チャネルと ADDR の対応

アナログ入力チャネル	変換結果が格納される A/D データレジスタ
AN0	ADDRA
AN1	ADDRB
AN2	ADDRC
AN3	ADDRD
AN4	ADDRE
AN5	ADDRF
AN6	ADDRG
AN7	ADDRH

18.3.2 A/D コントロール/ステータスレジスタ (ADCSR)

ADCSR は A/D 変換動作を制御します。

ビット	7	6	5	4	3	2	1	0
ビット名	ADF	ADIE	ADST	—	CH3	CH2	CH1	CH0
初期値	0	0	0	0	0	0	0	0
R/W	R/(W)*	R/W	R/W	R	R/W	R/W	R/W	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。

ビット	ビット名	初期値	R/W	説明
7	ADF	0	R/(W)*	A/D エンドフラグ A/D 変換の終了を示すステータスフラグです。 [セット条件] <ul style="list-style-type: none"> シングルモードで A/D 変換が終了したとき スキャンモードで選択されたすべてのチャンネルの A/D 変換が終了したとき [クリア条件] <ul style="list-style-type: none"> 1 の状態をリードした後、0 をライトしたとき (割り込みを使用し、CPU によってクリアする場合、0 ライト後必ずフラグをリードしてください。) ADI 割り込みにより DMAC または DTC が起動され、ADDR をリードしたとき
6	ADIE	0	R/W	A/D インタラプトイネーブル 1 にセットすると ADF による ADI 割り込みがイネーブルになります。
5	ADST	0	R/W	A/D スタート 0 にクリアすると A/D 変換を停止し、待機状態になります。1 にセットすると A/D 変換を開始します。シングルモードでは選択したチャンネルの A/D 変換が終了すると自動的にクリアされます。スキャンモードではソフトウェア、リセット、またはハードウェアスタンバイモード*によってクリアされるまで選択されたチャンネルを順次連続変換します。 【注】* 本 LSI には、ハードウェアスタンバイモードはありません。
4	—	0	R	リザーブビット このビットはリードのみ有効で、ライトは無効です。

ビット	ビット名	初期値	R/W	説 明
3	CH3	0	R/W	チャンネルセレクト 3~0
2	CH2	0	R/W	ADCR の SCANE ビット、SCANS ビットとともに、アナログ入力を選択します。
1	CH1	0	R/W	SCANE=0、
0	CH0	0	R/W	SCANE=1、
				SCANS=X のとき
				SCANS=0 のとき
				SCANS=1 のとき
				0000 : AN0
				0001 : AN1
				0010 : AN2
				0011 : AN3
				0100 : AN4
				0101 : AN5
				0110 : AN6
				0111 : AN7
				1XXX : 設定禁止

【記号説明】 X : Don't care

【注】 * フラグをクリアするための 0 ライトのみ可能です。

18.3.3 A/D コントロールレジスタ (ADCR)

ADCR は外部トリガによる A/D 変換開始をイネーブルにします。

ビット	7	6	5	4	3	2	1	0
ビット名	TRGS1	TRGS0	SCANE	SCANS	CKS1	CKS0	ADSTCLR	EXTRGS
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

ビット	ビット名	初期値	R/W	説 明
7	TRGS1	0	R/W	タイマトリガセレクト 1、0、拡張トリガセレクト
6	TRGS0	0	R/W	トリガ信号による A/D 変換開始をイネーブルにします。
0	EXTRGS	0	R/W	000 : 外部トリガによる A/D 変換開始を禁止
				001 : TPU1 からの変換トリガによる A/D 変換開始
				010 : TPU0 からの変換トリガによる A/D 変換開始
				011 : 設定禁止
				100 : 設定禁止
				101 : 設定禁止
				110 : ADTRG0 による A/D 変換開始
				111 : 設定禁止

ビット	ビット名	初期値	R/W	説明
5	SCANE	0	R/W	スキャンモード
4	SCANS	0	R/W	A/D 変換の動作モードを選択します。 0X : シングルモード 10 : スキャンモード (1~4 チャンネルの連続 A/D 変換) 11 : スキャンモード (1~8 チャンネルの連続 A/D 変換)
3	CKS1	0	R/W	クロックセレクト 1、0
2	CKS0	0	R/W	A/D 変換時に使用するクロック (ADCLK) を選択します。 本ビットの設定は ADCSR の ADST ビットが 0 のときに行い、その後変換モードの設定を行うようにしてください。また、SCKCR1.ACK=B'011 設定時、A ϕ /8 は選択しないでください。 00 : A ϕ 01 : A ϕ /2 10 : A ϕ /4 11 : A ϕ /8
1	ADSTCLR	0	R/W	A/D スタートクリア スキャンモード時に ADST ビットの自動クリアの設定をします。 0 : スキャンモードのとき、ADST ビットの自動的なクリアを禁止 1 : スキャンモードのとき、選択された全てのチャンネルの A/D 変換が終了すると自動的にクリアされます

【記号説明】 X : Don't care

【注】 * $\overline{\text{ADTRG}}$ による A/D 変換開始に設定する場合は、該当する端子の DDR ビットを 0 に、ICR ビットを 1 に設定してください。詳細は「第 9 章 I/O ポート」を参照してください。

18.3.4 アナログポートプルダウンコントロールレジスタ (APPDCR)

APPDCR は、アナログ入力端子のプルダウン MOS を制御します。

ビット	7	6	5	4	3	2	1	0
ビット名	AN7PD	AN6PD	AN5PD	AN4PD	AN3PD	AN2PD	AN1PD	AN0PD
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

ビット	ビット名	初期値	R/W	説明
7	AN7PD	0	R/W	アナログ入力端子のプルダウン MOS を制御します。 0 : プルダウン MOS はオフ 1 : プルダウン MOS はオン
6	AN6PD	0	R/W	
5	AN5PD	0	R/W	
4	AN4PD	0	R/W	
3	AN3PD	0	R/W	
2	AN2PD	0	R/W	
1	AN1PD	0	R/W	
0	AN0PD	0	R/W	

18.3.5 A/D 自己診断レジスタ (ADDIAGR)

ADDIAGR は A/D 変換器の故障を検出するためのレジスタです。A/D 変換の対象に、内部で生成する電圧値を選択できます。

ビット	7	6	5	4	3	2	1	0
ビット名	—	—	—	—	—	—	DIAG1	DIAG0
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

ビット	ビット名	初期値	R/W	説明
7~2	—	0	R/W	リザーブビット リードすると常に 0 が読み出されます。ライトする値は常に 0 にしてください。
1	DIAG1	0	R/W	A/D 自己診断ビット 00 : 自己診断オフ 01 : AV _{cc0} ×0 の電圧値の A/D 変換イネーブル 10 : AV _{cc0} ×1/2 の電圧値の A/D 変換イネーブル 11 : AV _{cc0} ×1 の電圧値の A/D 変換イネーブル
0	DIAG0	0	R/W	

18.4 動作説明

A/D 変換器は、逐次比較方式で分解能は 10 ビットです。動作モードには、シングルモード、スキャンモードがあります。動作モードやアナログ入力チャンネルの切り替えは、誤動作を避けるため ADCSR の ADST ビットが 0 の状態で行ってください。動作モードやアナログ入力チャンネルの変更と、ADST ビットを 1 にセットするのは同時に行うことができます。

18.4.1 シングルモード

シングルモードは、指定された 1 チャンネルのアナログ入力を以下のように 1 回 A/D 変換します。

1. ソフトウェア、TPU、または外部トリガ入力によって ADCSR の ADST ビットが 1 にセットされると、選択されたチャンネルの A/D 変換を開始します。
2. A/D 変換が終了すると、A/D 変換結果がそのチャンネルに対応する A/D データレジスタに転送されます。
3. A/D 変換終了後、ADCSR の ADF ビットが 1 にセットされます。このとき、ADIE ビットが 1 にセットされていると、ADI 割り込み要求が発生します。
4. ADST ビットは A/D 変換中は 1 を保持し、変換が終了すると自動的にクリアされて A/D 変換器は待機状態になります。A/D 変換中に ADST ビットを 0 にクリアすると A/D 変換を中止し、A/D 変換器は待機状態になります。

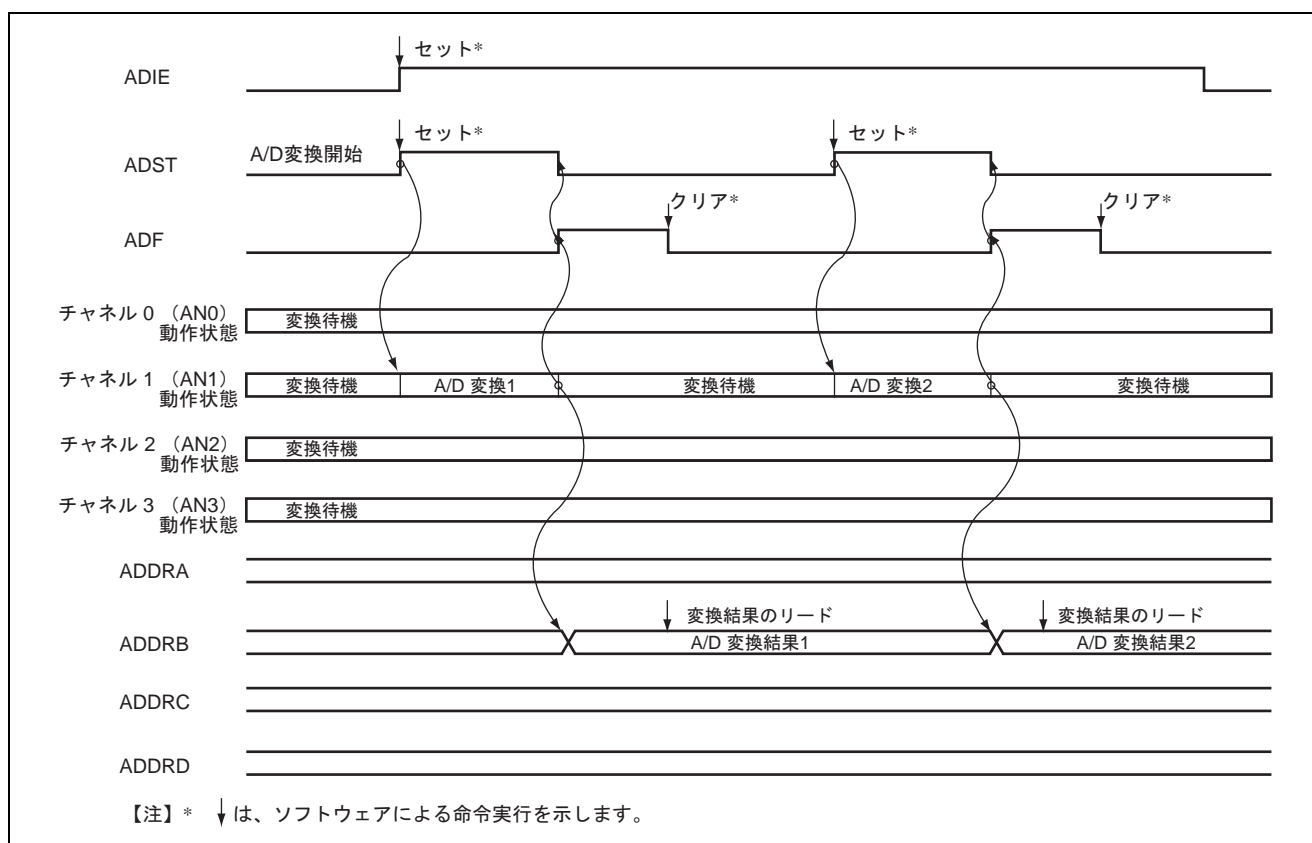


図 18.3 A/D 変換器の動作例（シングルチップモード、チャンネル 1 選択時）

18.4.2 スキャンモード

スキャンモードは、指定された最大4チャンネル、または最大8チャンネルのアナログ入力を以下のように順次連続してA/D変換します。

1. ソフトウェア、TPU、または外部トリガ入力によってADCSRのADSTビットが1にセットされると、選択されたチャンネルの第1チャンネルからA/D変換を開始します。最大4チャンネルの連続A/D変換（SCANE、SCANS=B'10）、または最大8チャンネルの連続A/D変換（SCANE、SCANS=B'11）を選択できます。4チャンネルの連続A/D変換の場合は、CH3、CH2=B'00のときAN0、CH3、CH2=B'01のときAN4からA/D変換を開始します。8チャンネルの連続A/D変換の場合は、CH3=B'0のときAN0からA/D変換を開始します。
2. それぞれのチャンネルのA/D変換が終了すると、A/D変換結果は順次そのチャンネルに対応するADDRに転送されます。
3. 選択されたすべてのチャンネルのA/D変換が終了すると、ADCSRのADFビットが1にセットされます。このときADIEビットが1にセットされていると、ADI割り込み要求が発生します。A/D変換器は再び第1チャンネルからA/D変換を開始します。
4. ADSTビットは自動的にクリアされず、1にセットされている間は2～3を繰り返します。ADSTビットを0にクリアするとA/D変換を中止し、A/D変換器は待機状態になります。その後、ADSTビットを1にセットすると再び第1チャンネルからA/D変換を開始します。
5. ADCRのADSTCLRビットが1にセットされている場合、ADSTビットは選択されたすべてのチャンネルのA/D変換が終了すると自動的にクリアされ、A/D変換を中止し、A/D変換器は待機状態になります。

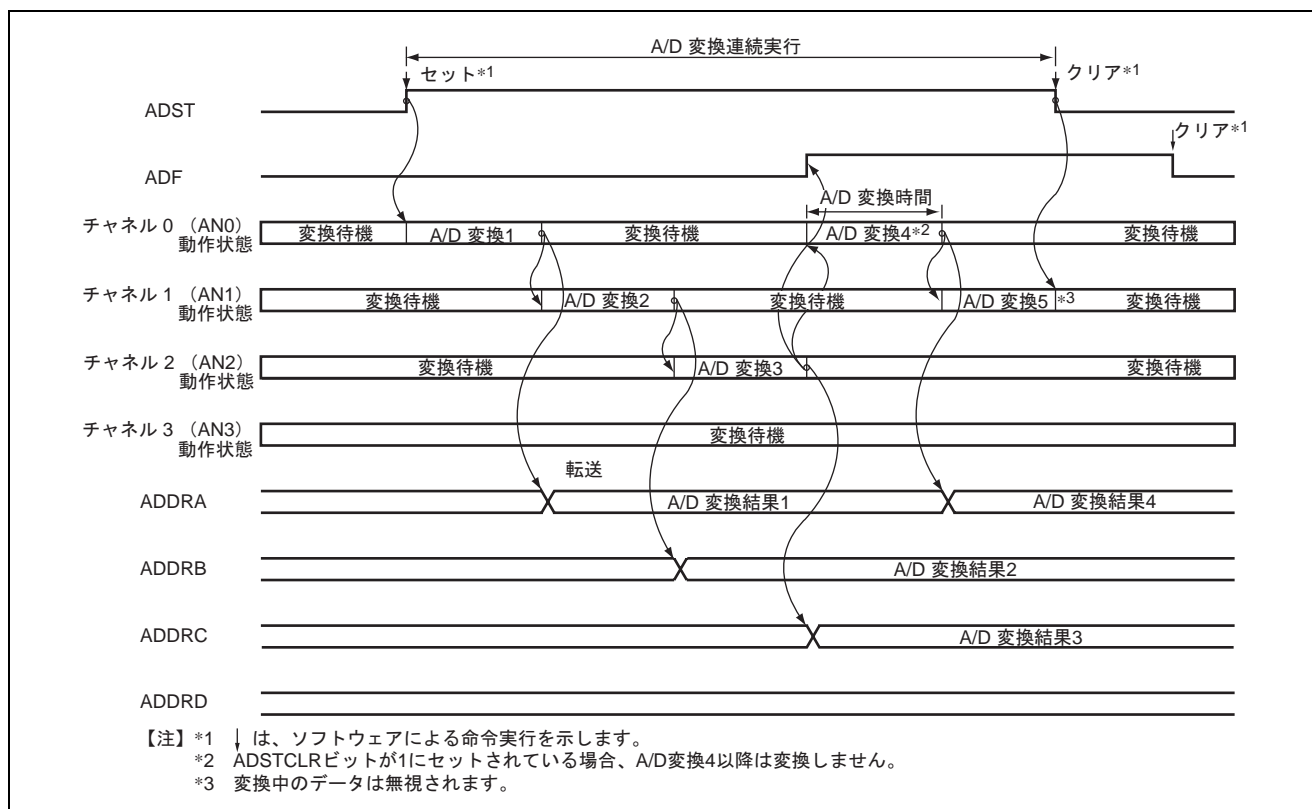


図 18.4 A/D 変換器の動作例（スキャンモード、AN0～AN2 の3チャンネル選択時）

18.4.3 入力サンプリングと A/D 変換時間

A/D 変換器には、サンプル&ホールド回路が内蔵されています。A/D 変換器は、ADCSR の ADST ビットが 1 にセットされてから A/D 変換開始遅延時間(t_D)経過後、入力のサンプリングを行い、その後 A/D 変換を開始します。A/D 変換のタイミングを図 18.5 に示します。また、A/D 変換時間を表 18.3 に示します。

A/D 変換時間(t_{CONV})は、図 18.5 に示すように、 t_D と入力サンプリング時間(t_{SPL})を含めた時間となります。ここで t_D は、ADCSR へのライトタイミングにより決まり、一定値とはなりません。そのため、変換時間は表 18.3 に示す範囲で変化します。

スキャンモードの変換時間は、表 18.3 に示す値が 1 回目の変換時間となります。2 回目以降の変換時間は表 18.4 に示す値となります。いずれの場合も、変換時間は A/D 変換特性に示す範囲となるように ADCR の CKS1、CKS0 ビットを設定してください。

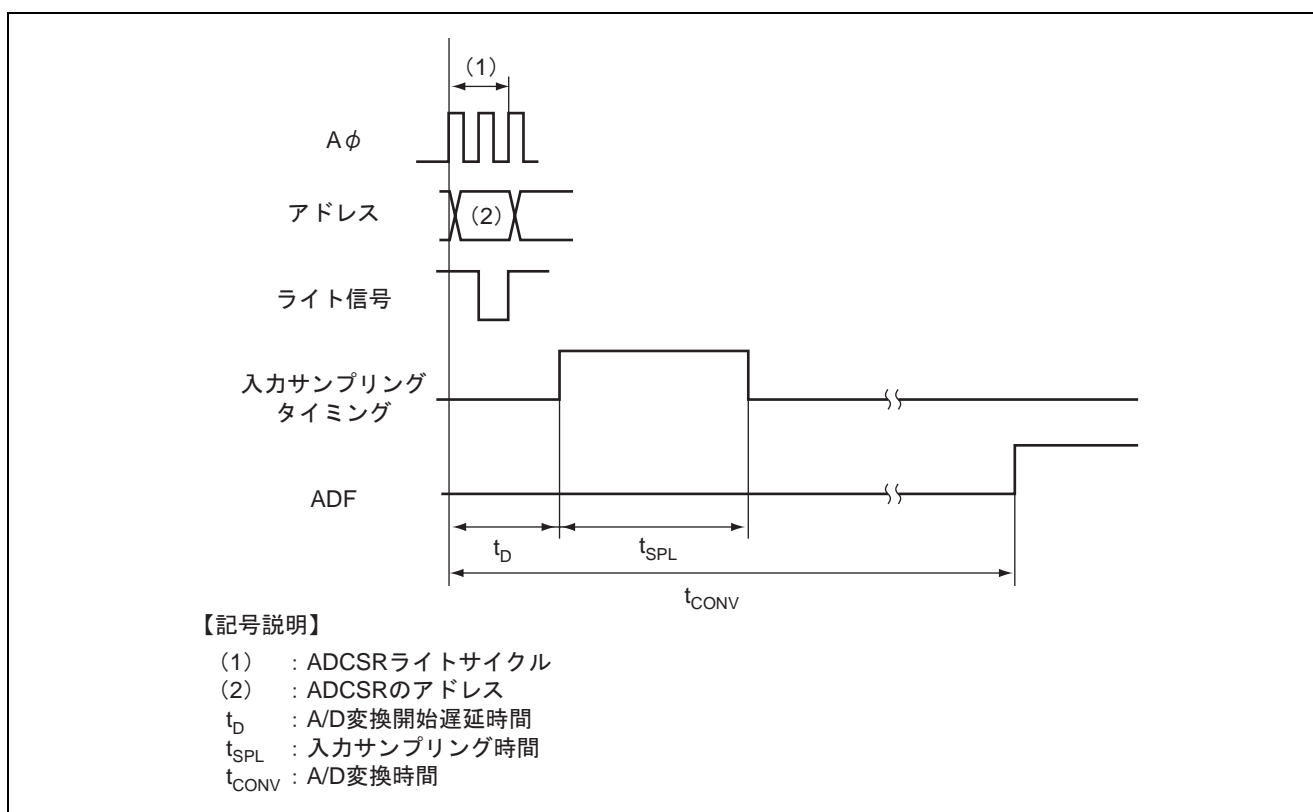


図 18.5 A/D 変換タイミング

表 18.3 A/D 変換特性

項目	記号	CKS1=0						CKS1=1					
		CKS0=0			CKS0=1			CKS0=0			CKS0=1		
		min	typ	max	min	typ	max	min	typ	max	min	typ	max
A/D 変換開始遅延時間	t_D	3	—	5	3	—	6	3	—	8	3	—	12
入力サンプリング時間	t_{SPL}	—	15	—	—	30	—	—	60	—	—	120	—
A/D 変換時間	t_{CONV}	45	—	47	85	—	88	165	—	170	325	—	334

【注】 表中の数値の単位はステートです。また、表中の値は暫定です。

表 18.4 A/D 変換時間（スキャンモード）

CKS1	CKS0	変換時間（ステート）
0	0	40（固定）
	1	80（固定）
1	0	160（固定）
	1	320（固定）

18.4.4 外部トリガ入力タイミング

A/D 変換は、外部トリガ入力により開始することも可能です。外部トリガ入力は、ADCR の TRGS1、TRGS0、EXTRGS ビットが B'110 にセットされているとき、 $\overline{ADTRG0}$ 端子から入力されます。 $\overline{ADTRG0}$ の立ち下がりエッジで、ADCSR の ADST ビットが 1 にセットされ、A/D 変換が開始されます。その他の動作は、シングルモード/スキャンモードによらず、ソフトウェアによって ADST ビットを 1 にセットした場合と同じです。このタイミングを図 18.6 に示します。

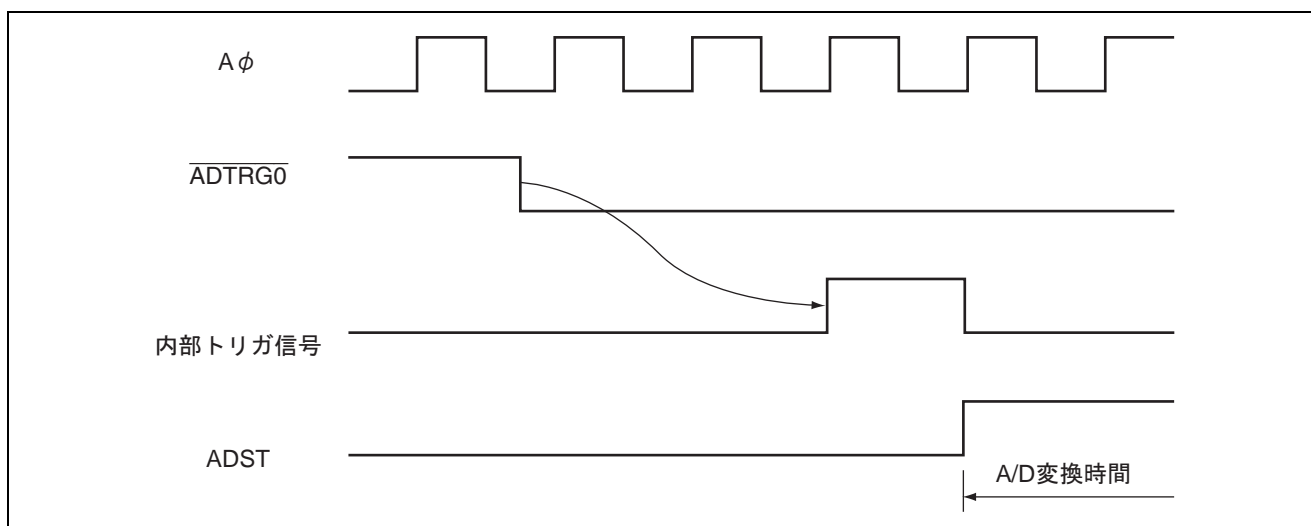


図 18.6 外部トリガ入力タイミング

18.5 割り込み要因

A/D 変換器は、A/D 変換が終了すると A/D 変換終了割り込み (ADI) を発生します。ADI 割り込み要求は、A/D 変換終了後 ADCSR の ADF が 1 にセットされ、このとき ADIE ビットが 1 にセットされるとイネーブルになります。ADI 割り込みで、データトランスファコントローラ (DTC) および DMA コントローラ (DMAC) の起動ができます。ADI 割り込みで変換されたデータのリードを DTC または DMAC で行くと、連続変換がソフトウェアの負担なく実現できます。

表 18.5 A/D 変換器の割り込み要因

名称	割り込み要因	割り込みフラグ	DTC の起動	DMAC の起動
ADI	A/D 変換終了	ADF	可	可

18.6 A/D 変換精度の定義

本 LSI の A/D 変換精度の定義は以下のとおりです。

- 分解能
A/D 変換器のデジタル出力コード数
- 量子化誤差
A/D 変換器が本質的に有する偏差であり、1/2LSB で与えられる (図 18.7)
- オフセット誤差
デジタル出力が最小電圧値 B'000000000 (H'000) から B'000000001 (H'001) に変化するときのアナログ入力電圧値の理想 A/D 変換特性からの偏差 (図 18.8)
- フルスケール誤差
デジタル出力が B'111111110 (H'3FE) から B'111111111 (H'3FF) に変化するときのアナログ入力電圧値の理想 A/D 変換特性からの偏差 (図 18.8)
- 非直線性誤差
ゼロ電圧からフルスケール電圧までの間の理想 A/D 変換特性からの誤差。ただし、オフセット誤差、フルスケール誤差、量子化誤差を含まない (図 18.8)
- 絶対精度
デジタル値とアナログ入力値との偏差。オフセット誤差、フルスケール誤差、量子化誤差および非直線誤差を含む

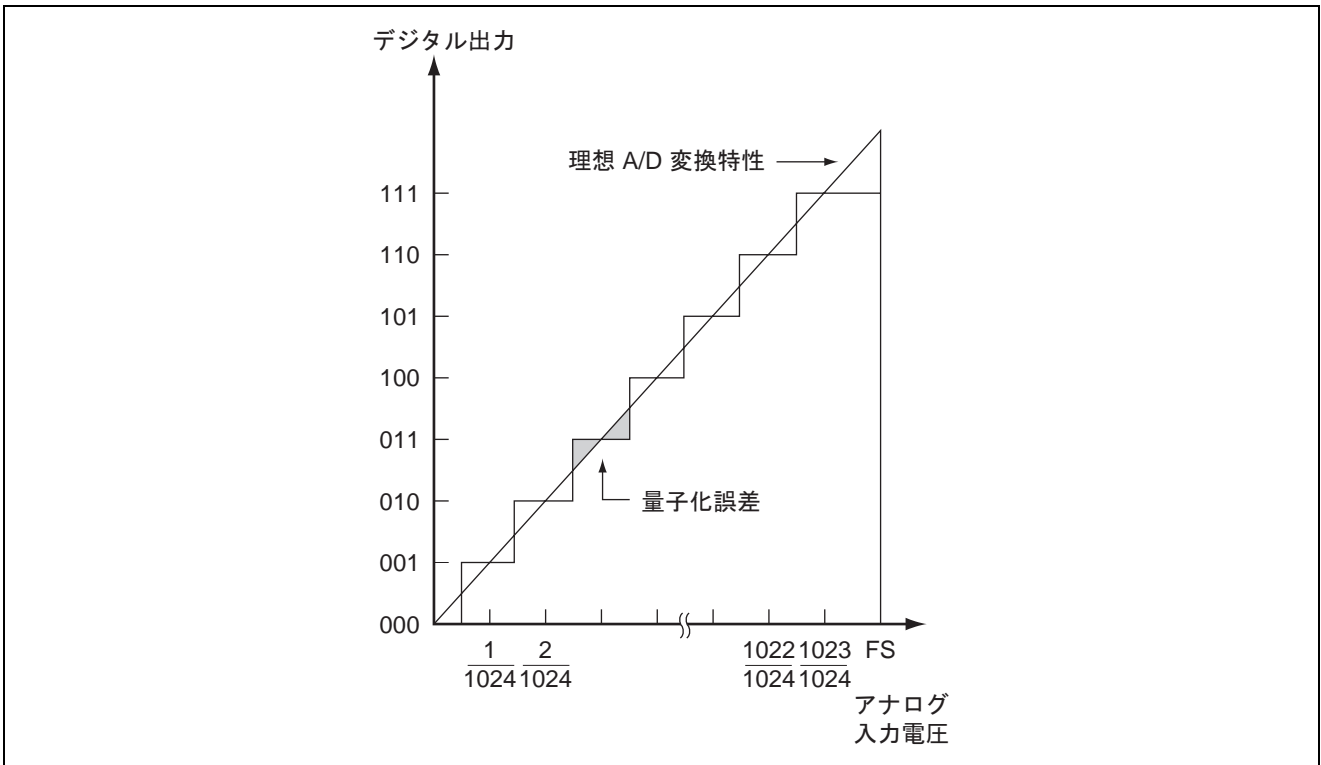


図 18.7 A/D 変換精度の定義

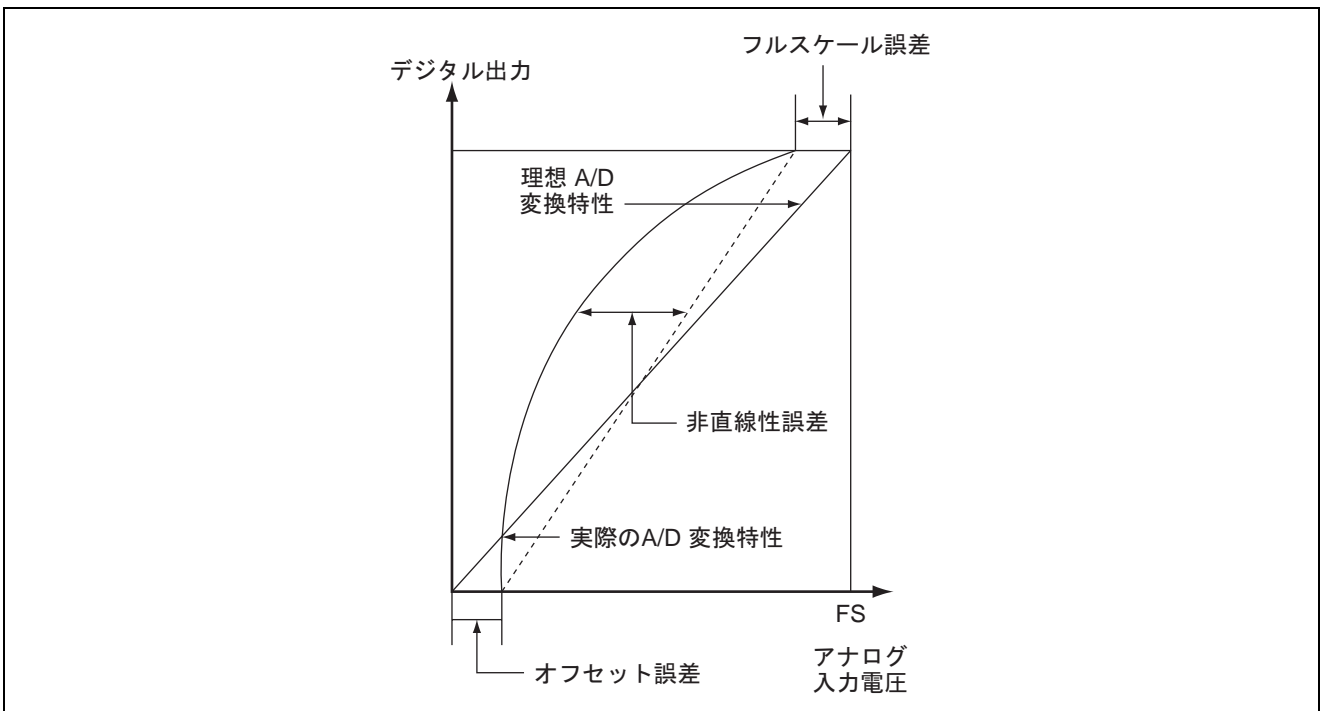


図 18.8 A/D 変換精度の定義

18.7 アナログポートプルダウン機能

アナログポートプルダウン機能はアナログポートそれぞれにプルダウン MOS がついており、APPDCR の AN7PD~AN0PD ビットに 1 (初期値 0) をセットすることにより、それぞれのアナログチャンネルに対応したプルダウン MOS が ON します。

これによりアナログポートに外部回路から電圧を印加している場合、A/D 変換により得られた結果からアナログポートと外部ユニット間の信号の断線チェックを行うことができます。

- 例

正常時：外部回路から印加された電圧に近い変換結果が得られる。

異常時（断線時）：AVss に近い変換結果が得られる。

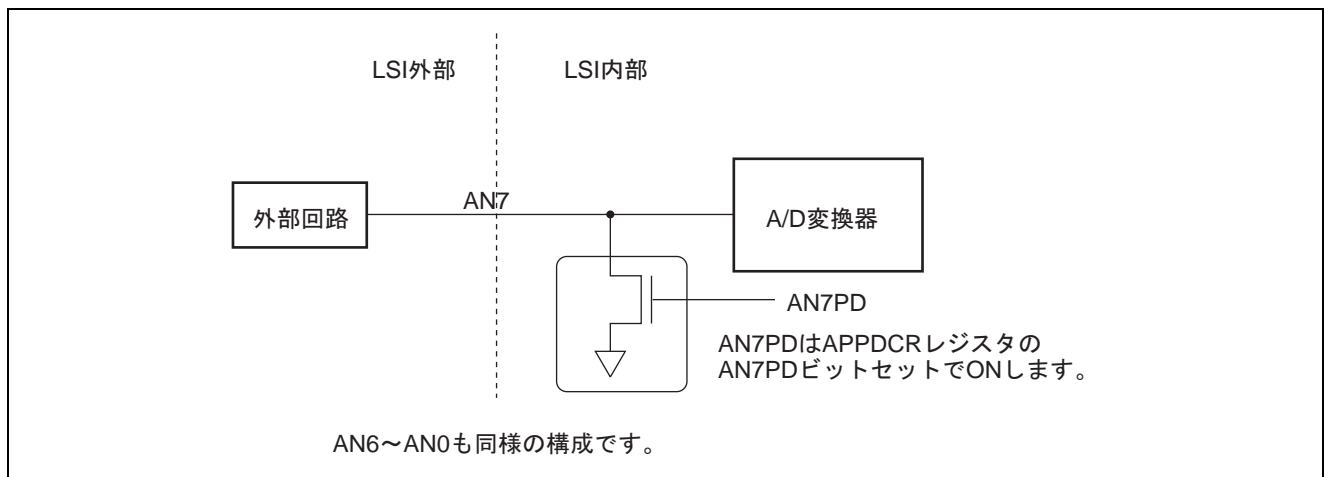


図 18.9 アナログポートプルダウン機能概略図

18.8 A/D 変換器の自己診断

ソフトウェアによる A/D 変換器の自己診断が可能です。

ADDIAGR で A/D 変換対象に内部電圧値を選択し、A/D 変換を行います。A/D 変換終了後、ADDR、ADDIAGR の値をソフトウェアで読み出すと、変換値が正常の範囲にある（正常）かない（異常）かを判定できます。

なお、A/D 変換対象に内部電圧値が選択されている場合、アナログ入力は無視されます。ADDIAGR の設定は、誤動作を避けるため ADCSR の ADST ビットが 0 の状態で行ってください。

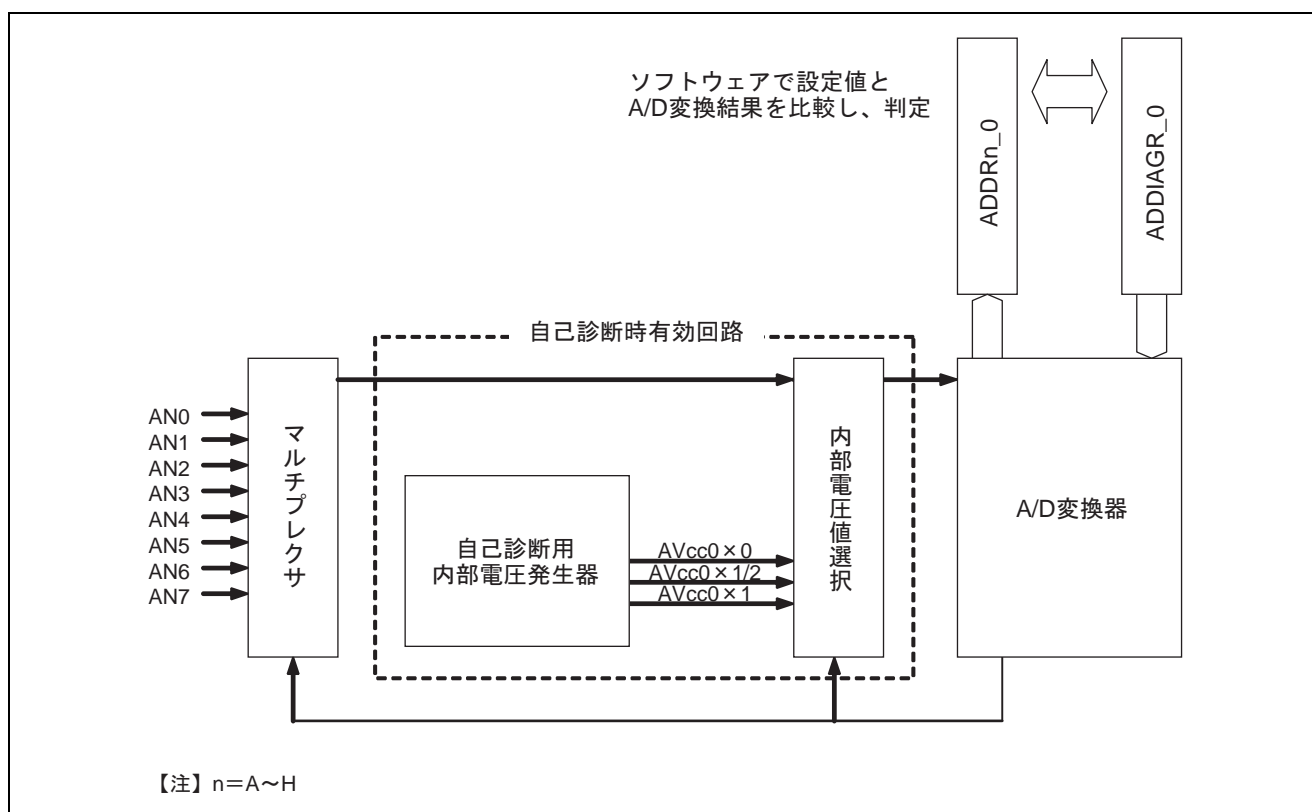


図 18.10 A/D 変換器の自己診断概略図

表 18.6 A/D 変換結果の理想値（正常範囲時の値）

選択した内部電圧値	ADDR の値(正常範囲時の値)
$AV_{CC0} \times 0$	H'0000(H'0000~H'0A00*)
$AV_{CC0} \times 1/2$	H'7FC0(H'75C0~H'89C0*)
$AV_{CC0} \times 1$	H'FFC0(H'F5C0~H'FFC0*)

【注】 * 内部電圧発生器の電圧誤差を含んだ値です。

18.9 使用上の注意事項

18.9.1 モジュールストップモードの設定

モジュールストップコントロールレジスタにより、A/D 変換器の動作禁止/許可を設定することが可能です。初期値では、A/D 変換器の動作は停止します。モジュールストップモードを解除することにより、レジスタのアクセスが可能になります。A/D 変換器の動作後、モジュールストップモードに遷移する場合は ADST、TRGS1、TRGS0、EXTRGS ビット、および APPDCR、ADDIAGR をすべて 0 にクリアして A/D 変換を禁止してください。その後、1 ワードダミーリードしてからモジュールストップコントロールレジスタを設定してください。モジュールストップコントロールレジスタの詳細は、「第 23 章 低消費電力」を参照してください。

18.9.2 ソフトウェアスタンバイ時の A/D 変換保持機能

A/D 変換を許可した状態で本 LSI がソフトウェアスタンバイモードになると A/D 変換は保持され、アナログ電源電流は A/D 変換中と同等になります。ソフトウェアスタンバイモードでアナログ電源電流を低減する必要がある場合は、ADST、TRGS1、TRGS0、EXTRGS ビット、および APPDCR、ADDIAGR をすべて 0 にクリアして A/D 変換を禁止してください。その後、1 ワードダミーリードしてソフトウェアスタンバイモードに遷移させてください。

18.9.3 許容信号源インピーダンスについて

本 LSI のアナログ入力は、信号源インピーダンスが $5\text{k}\Omega$ 以下の入力信号に対し、変換精度が保証される設計となっています。これは A/D 変換器のサンプル&ホールド回路の入力容量をサンプリング時間内に充電するための規格で、センサの出力インピーダンスが $5\text{k}\Omega$ を超える場合は、充電不足が生じて A/D 変換精度が保証できなくなります。シングルモードで変換を行うときに外部に大容量を設けている場合は、入力の負荷は実質的に内部入力抵抗の $10\text{k}\Omega$ だけになりますので、信号源インピーダンスは不要となります。ただし、ローパスフィルタとなりますので、微分係数の大きなアナログ信号（たとえば $5\text{mV}/\mu\text{s}$ 以上）には追従できないことがあります（図 18.11）。高速のアナログ信号を変換する場合や、スキャンモードで変換を行う場合には、低インピーダンスのバッファを入れてください。

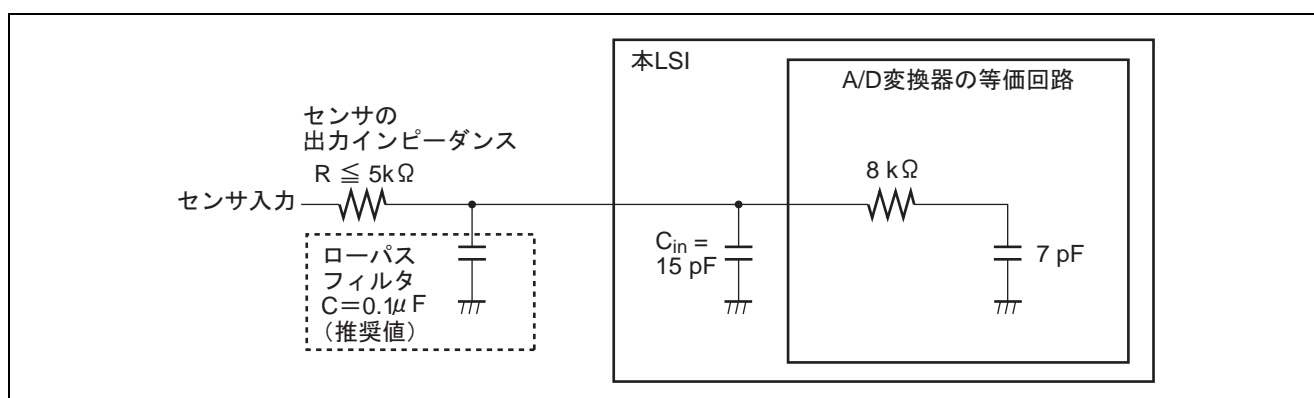


図 18.11 アナログ入力回路の例

18.9.4 絶対精度への影響

容量を付加することにより、GND とのカップリングを受けることとなりますので、GND にノイズがあると絶対精度が悪化する可能性がありますので、必ず AV_{SS} 等の電氣的に安定な GND に接続してください。

またフィルタ回路が実装基板上でデジタル信号と干渉したり、アンテナとならないように注意してください。

18.9.5 アナログ電源端子他の設定範囲

以下に示す電圧の設定範囲を超えて LSI を使用した場合、LSI の信頼性に悪影響を及ぼすことがあります。

- アナログ入力電圧の設定範囲

A/D 変換中、アナログ入力端子 AN_n に印加する電圧は $AV_{SS} \leq V_{AN} \leq AV_{CC0}$ 、 $AV_{SS} \leq V_{AN} \leq AV_{CC1}$ の範囲としてください。

- AV_{CC0} 、 AV_{CC1} 、 AV_{SS} と V_{CC} 、 V_{SS} の関係

AV_{CC0} 、 AV_{CC1} 、 AV_{SS} と V_{CC} 、 V_{SS} との関係は $AV_{CC0} = V_{CC} \pm 0.3V$ 、 $AV_{CC1} = V_{CC} \pm 0.3V$ かつ $AV_{SS} = V_{SS}$ としてください。A/D 変換器を使用しない場合、 $AV_{CC0} = V_{CC}$ 、 $AV_{CC1} = V_{CC}$ 、 $AV_{SS} = V_{SS}$ としてください。

18.9.6 ボード設計上の注意

ボード設計時には、デジタル回路とアナログ回路をできるだけ分離してください。また、デジタル回路の信号線とアナログ回路の信号配線を交差させたり、近接させないでください。誘導によりアナログ回路が誤動作し、A/D 変換値に悪影響を及ぼします。アナログ入力端子 ($AN0 \sim AN15$)、アナログ電源電圧 (AV_{CC0} 、 AV_{CC1}) は、アナロググランド (AV_{SS}) でデジタル回路と分離してください。さらに、アナロググランド (AV_{SS}) は、ボード上の安定したグランド (V_{SS}) に一点接続してください。

18.9.7 ノイズ対策上の注意

過大なサージなど異常電圧によるアナログ入力端子 ($AN0 \sim AN15$) の破壊を防ぐために、**図 18.12** に示すように $AV_{CC0} - AV_{SS}$ 、 $AV_{CC1} - AV_{SS}$ 間に保護回路を接続してください。 AV_{CC0} 、 AV_{CC1} に接続するバイパスコンデンサ、 $AN0 \sim AN15$ に接続するフィルタ用のコンデンサは、必ず AV_{SS} に接続してください。

なお、フィルタ用のコンデンサを接続すると、 $AN0 \sim AN15$ の入力電流が平均化されるため、誤差を生じることがあります。また、スキャンモードなどで A/D 変換を頻繁に行う場合、A/D 変換器内部のサンプル&ホールド回路の容量に充放電される電流が入力インピーダンス (R_{in}) を経由して入力される電流を上回ると、アナログ入力端子の電圧に誤差を生じます。したがって、回路定数は充分ご検討の上決定してください。

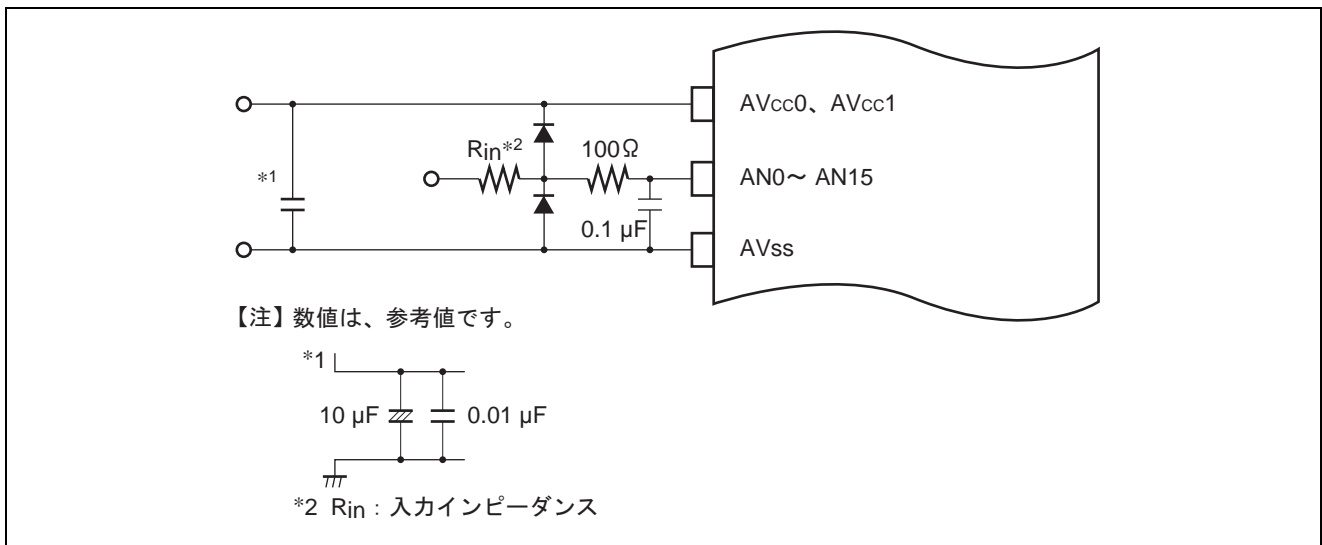


図 18.12 アナログ入力保護回路の例

表 18.7 アナログ端子の規格

項目	min	max	単位
アナログ入力容量	—	20	pF
許容信号源インピーダンス	—	5	k Ω

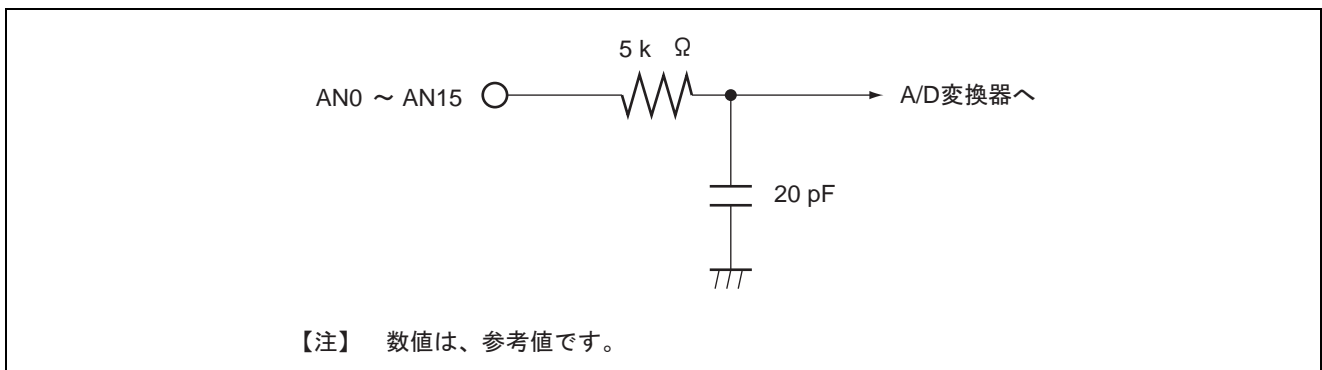


図 18.13 アナログ入力端子等価回路

18.9.8 A/D 変換器停止方法に関する注意事項

A/D 変換器の A/D 変換中にソフトウェアにより A/D スタート (ADST) ビットをクリアした場合、A/D 変換データ結果の格納先 (ADDR) を誤る、もしくは A/D 変換を再開した際に割り込みフラグの誤セット等の現象が発生する場合があります。

本現象は、下記の対策を行うことで回避できます。

(1) シングルモード時およびスキャンモード (1 サイクルモード) 時

ADST ビットは A/D 変換が終了すると自動的にクリアされますので、A/D 変換中にソフトウェアで ADST ビットをクリアしないでください。

(2) スキャンモード (連続スキャンモード) 時

(a) ソフトウェアによる A/D 変換器の起動時

A/D 変換中にソフトウェアにより ADST ビットをクリアしないでください。A/D 変換を停止させる場合は、SCANE ビットをスキャンモードからシングルモードに変更してください。これにより、ソフトウェアで ADST ビットをクリアすることなく A/D 変換器を停止できます。

ただし、SCANE ビットを変更後、A/D 変換が停止し A/D エンドフラグ (ADF) に 1 がセットされるまでに、最大で A/D 変換 1.5 チャンネル分の時間がかかる場合があります。また、A/D 変換停止後の ADDR の値は使用しないでください。

(b) 外部トリガによる A/D 起動時の場合

A/D 変換中にソフトウェアにより ADST ビットをクリアしないでください。A/D 変換を停止させる場合は、外部トリガによる A/D 変換開始を禁止してから、SCANE ビットをスキャンモードからシングルモードに変更してください。

これによりソフトウェアによって ADST ビットをクリアすることなく停止できます。

ただし、SCANE ビットを変更後、A/D 変換が停止し、A/D エンドフラグ (ADF) に 1 がセットされるまでに、最大で A/D 変換 1.5 チャンネル分の時間がかかる場合があります。また、A/D 変換停止後の ADDR の値は使用しないでください。

(2) の詳細な設定は、[図 18.14](#) および [図 18.15](#) を参照してください。

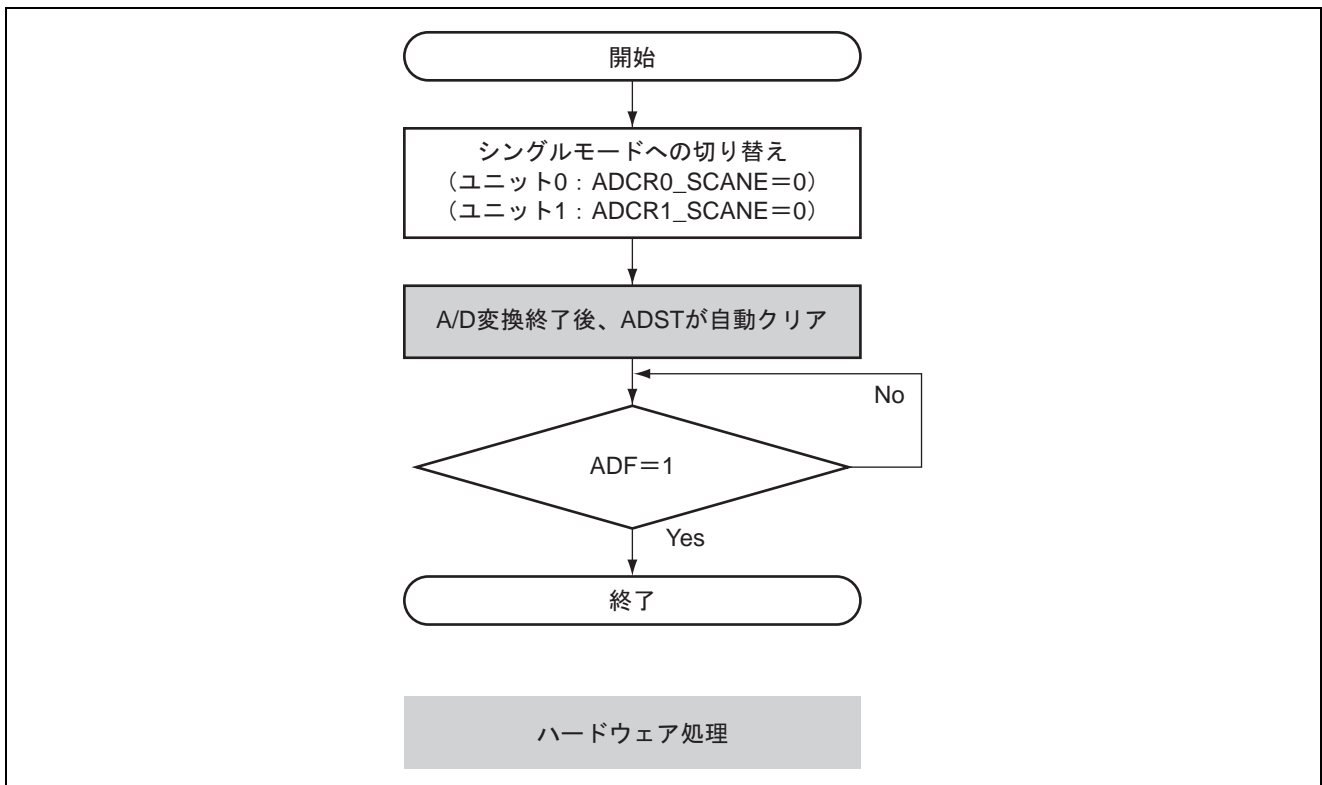


図 18.14 ソフトウェア起動による連続スキャンモード停止方法

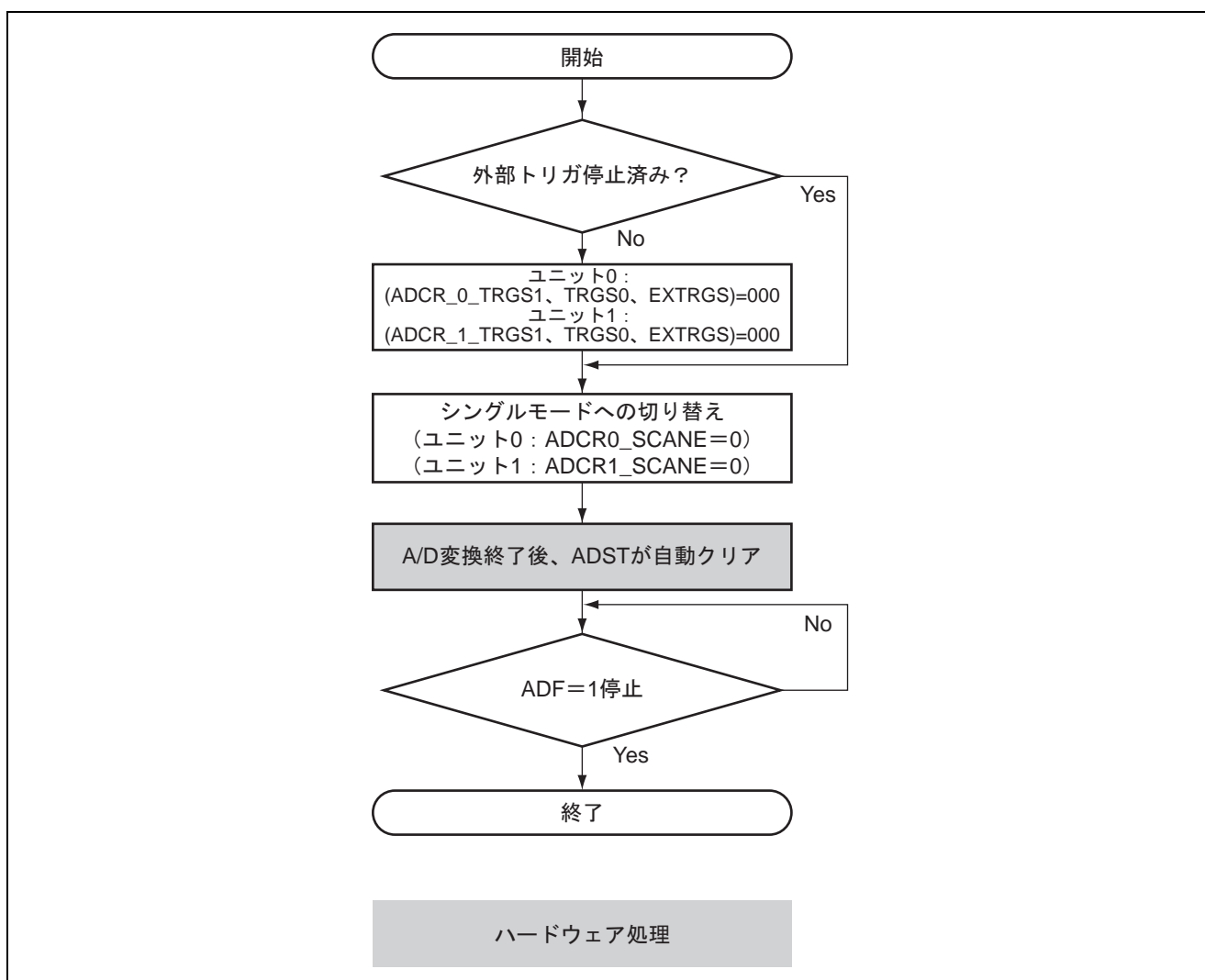


図 18.15 外部トリガによる連続スキャンモード

19. RAM

本 LSI は、最大 40KB の RAM を内蔵しています。内蔵 RAM は、32 ビットの内部バスと接続しており、バイトデータ、ワードデータ、ロングワードデータにかかわらず、リードは、I ϕ 1 ステート、ライトは、I ϕ 1 または 2 ステートでアクセスすることができます。

表 19.1 に内蔵 RAM アドレス空間、表 19.2 に内蔵 RAM エリアを示します。

本章の説明は 40KB になります。24KB 版ではエリア 3、4 がアクセス禁止空間になります。

19.1 特長

- **アクセス** : CPU、DMAC、DTCは、8、16、または32ビット幅で内蔵RAMをアクセスすることができます。内蔵RAMのデータは、高速アクセスが必要なプログラムエリアまたはスタックエリアやデータアクセスとしての使用に適しています。
- **ECC** : レジスタの設定によりECCエラー訂正機能の有効/無効を選択することができます。初期状態はECCエラー訂正機能が有効です。32ビットデータに対して1ビットの誤り訂正、2ビットの誤り検出が可能です。ECCエラー訂正機能が有効のときにライトは2サイクルとなります。
- **Parity** : ECCエラー訂正機能が無効のときは、パリティによる誤り検出機能が有効で8ビットデータに対して1ビットの誤り検出が可能です。ECCエラー訂正機能が無効のとき（パリティエラー検出機能が有効の時）は、ライトは1サイクル設定となります。
- **エラーフラグ** : RAMエラー（ECCエラー／パリティエラー）発生時にエラー発生を示すフラグを持ちます。
- **割り込み** : RAMエラー（ECCエラー／パリティエラー）発生時の割り込み要求の有無をレジスタで設定することができます。
- **RAMアクセスプロテクト** : 内蔵RAMアドレスのエリアごとに内蔵RAMアクセスのイネーブル／ディスエーブル設定が可能です。
- **RAM書き込みプロテクト** : 内蔵RAMアドレスのエリアごとに内蔵RAMの書き込みイネーブル／ディスエーブル設定が可能です。
- **制御レジスタの書き換え** : RAMの制御レジスタは容易に書き換えられないように書き込み方法が一般のレジスタと異なっています。

表 19.1 内蔵 RAM アドレス空間

製品分類		RAM 容量	RAM アドレス
フラッシュメモリ版	H8SX/1727S	40KB	H'FF2000~H'FFBFFF
	H8SX/1725S	24KB	H'FF6000~H'FFBFFF

表 19.2 内蔵 RAM エリア

RAM エリア	アドレス
エリア 4 (8KB) *	H'FF2000~H'FF3FFF
エリア 3 (8KB) *	H'FF4000~H'FF5FFF
エリア 2 (8KB)	H'FF6000~H'FF7FFF
エリア 1 (8KB)	H'FF8000~H'FF9FFF
エリア 0 (8KB)	H'FFA000~H'FFBFFF

【注】 * 24KB 版ではサポートしません。

19.2 レジスタの説明

RAM 関連レジスタには以下のレジスタがあります。

表 19.3 レジスタ構成

レジスタ名	略称	R/W	初期値	アドレス	アクセス サイズ
RAM イネーブルコントロールレジスタ	RAMEN	R/W	H'00FF	H'FFE400	8、(16) *
RAM 書き込みイネーブルコントロールレジスタ	RAMWEN	R/W	H'00FF	H'FFE402	8、(16) *
RAM ECC イネーブルコントロールレジスタ	RAMECC	R/W	H'0000	H'FFE404	8、(16) *
RAM エラーステータスレジスタ	RAMERR	R	H'00	H'FFE406	8
RAM エラー割り込みコントロールレジスタ	RAMINT	R/W	H'00	H'FFE410	8
RAM アクセスサイクル設定レジスタ	RAMACYC	R/W	H'0020	H'FFE412	8、(16) *

【注】 * 書き込むときは、上位バイトにキーデータを書き込んで必ずワードでアクセスしてください。

19.2.1 RAM イネーブルコントロールレジスタ (RAMEN)

RAMEN は、読み出し/書き込み可能な 16 ビットのレジスタで、内蔵 RAM へのアクセス許可/禁止を設定します。RAMEN は、リセット/スタンバイ時に H'00FF に初期化されます。書き込みはワードアクセスのみ可能で、読み出しはワード、バイトのどちらでも可能です。

アクセスを許可するエリアに対応する RAME ビットを 1 にセットすると内蔵 RAM が有効になります。0 にクリアするとアクセスできません。アクセス禁止状態では、対応するエリアのリードおよび命令フェッチは不定値が読み出され、ライトは無視されます。初期値は 1 です。

なお、RAME ビットを書き換える場合には、上位バイトを H'96 にし、下位バイトを書き込みデータにしてワードサイズで行ってください。

RAMEN の上位バイト (ビット 15~8) の読み出し時には常に H'00 が読み出されます。

RAMEN へのライト命令の直後に内蔵 RAM をアクセスするような命令を置かないでください。もし内蔵 RAM アクセス命令を置いた場合、正常なアクセスは保証できません。

RAME ビットを 1 にセットして内蔵 RAM を有効にする場合、RAMEN へのライト命令の直後に RAMEN のリード命令を置いてください。

ビット	15	14	13	12	11	10	9	8
ビット名	RNKEY7	RNKEY6	RNKEY5	RNKEY4	RNKEY3	RNKEY2	RNKEY1	RNKEY0
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)
ビット	7	6	5	4	3	2	1	0
ビット名	—	—	—	RAME4*	RAME3*	RAME2	RAME1	RAME0
初期値:	1	1	1	1	1	1	1	1
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W

【注】 本レジスタは、誤って書き換えられないように、書き込み方法が一般のレジスタと異なります。詳しくは、「19.2.7 レジスタアクセス時の注意」を参照してください。

* 24KB版ではサポートしません。

ビット	ビット名	初期値	R/W	説明
15	RNKEY7	0	R/(W)	RAME ビットの書き換え可否を設定するためのビットです。 H'96 : RAME7~0 ビットの書き込み許可。書き込みデータは保持されませんので読み出し値は常に H'00 です。 H'96 以外 : RAME7~0 ビットの書き込み不可
14	RNKEY6	0	R/(W)	
13	RNKEY5	0	R/(W)	
12	RNKEY4	0	R/(W)	
11	RNKEY3	0	R/(W)	
10	RNKEY2	0	R/(W)	
9	RNKEY1	0	R/(W)	
8	RNKEY0	0	R/(W)	
7~5	—	1	R	リザーブビット 読み出すと常に 1 が読み出されます。 書き込む値も常に 1 にしてください。
4	RAME4*	1	R/W	RAM イネーブル 4 内蔵 RAM のエリア 4 に対するアクセスの有効/無効を設定するためのビットです。 0 : 内蔵 RAM のエリア 4 が無効 1 : 内蔵 RAM のエリア 4 が有効 [クリア条件] • 0 書き込み (上位バイトに H'96 を同時書き込み) [セット条件] • リセット/スタンバイ • 1 書き込み (上位バイトに H'96 を同時書き込み)
3	RAME3*	1	R/W	RAM イネーブル 3 内蔵 RAM のエリア 3 に対するアクセスの有効/無効を設定するためのビットです。 0 : 内蔵 RAM のエリア 3 が無効 1 : 内蔵 RAM のエリア 3 が有効 [クリア条件] • 0 書き込み (上位バイトに H'96 を同時書き込み) [セット条件] • リセット/スタンバイ • 1 書き込み (上位バイトに H'96 を同時書き込み)

ビット	ビット名	初期値	R/W	説明
2	RAME2	1	R/W	<p>RAM イネーブル 2</p> <p>内蔵 RAM のエリア 2 に対するアクセスの有効/無効を設定するためのビットです。</p> <p>0 : 内蔵 RAM のエリア 2 が無効 1 : 内蔵 RAM のエリア 2 が有効</p> <p>[クリア条件]</p> <ul style="list-style-type: none"> 0 書き込み (上位バイトに H'96 を同時書き込み) <p>[セット条件]</p> <ul style="list-style-type: none"> リセット/スタンバイ 1 書き込み (上位バイトに H'96 を同時書き込み)
1	RAME1	1	R/W	<p>RAM イネーブル 1</p> <p>内蔵 RAM のエリア 1 に対するアクセスの有効/無効を設定するためのビットです。</p> <p>0 : 内蔵 RAM のエリア 1 が無効 1 : 内蔵 RAM のエリア 1 が有効</p> <p>[クリア条件]</p> <ul style="list-style-type: none"> 0 書き込み (上位バイトに H'96 を同時書き込み) <p>[セット条件]</p> <ul style="list-style-type: none"> リセット/スタンバイ 1 書き込み (上位バイトに H'96 を同時書き込み)
0	RAME0	1	R/W	<p>RAM イネーブル 0</p> <p>内蔵 RAM のエリア 0 に対するアクセスの有効/無効を設定するためのビットです。</p> <p>0 : 内蔵 RAM のエリア 0 が無効 1 : 内蔵 RAM のエリア 0 が有効</p> <p>[クリア条件]</p> <ul style="list-style-type: none"> 0 書き込み (上位バイトに H'96 を同時書き込み) <p>[セット条件]</p> <ul style="list-style-type: none"> リセット/スタンバイ 1 書き込み (上位バイトに H'96 を同時書き込み)

【注】 * 24KB 版ではサポートしません。

19.2.2 RAM 書き込みイネーブルコントロールレジスタ (RAMWEN)

RAMWEN は、読み出し／書き込み可能な 16 ビットのレジスタで、内蔵 RAM への書き込み許可／禁止を設定します。RAMWEN は、リセット／スタンバイ時に H'00FF に初期化されます。書き込みはワードアクセスのみ可能で、読み出しはワード、バイトのどちらでも可能です。

アクセスを許可するエリアに対応する RAMWE ビットを 1 にセットすると内蔵 RAM への書き込みが有効になります。0 にクリアすると内蔵 RAM への書き込みはできません。アクセス禁止状態では、内蔵 RAM へのライトは無視されます。初期値は 1 です。

なお、RAMWE ビットを書き換える場合には、上位バイトを H'69 にし、下位バイトを書き込みデータにしてワードサイズで行ってください。

RAMWEN の上位バイト（ビット 15～8）の読み出し時には常に H'00 が読み出されます。

RAMEN へのライト命令の直後に内蔵 RAM をアクセスするような命令を置かないでください。もし内蔵 RAM アクセス命令を置いた場合、正常なアクセスは保証できません。

RAMWE ビットを 1 にセットして内蔵 RAM への書き込みを有効にする場合、RAMWEN へのライト命令の直後に RAMWEN のリード命令を置いてください。

ビット	15	14	13	12	11	10	9	8
ビット名	RWNKEY7	RWNKEY6	RWNKEY5	RWNKEY4	RWNKEY3	RWNKEY2	RWNKEY1	RWNKEY0
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)
ビット	7	6	5	4	3	2	1	0
ビット名	—	—	—	RAMWE4*	RAMWE3*	RAMWE2	RAMWE1	RAMWE0
初期値:	1	1	1	1	1	1	1	1
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W

【注】 本レジスタは、誤って書き換えられないように、書き込み方法が一般のレジスタと異なっています。詳しくは、「19.2.7 レジスタアクセス時の注意」を参照してください。

* 24KB版ではサポートしません。

ビット	ビット名	初期値	R/W	説明
15	RWNKEY7	0	R/(W)	RAMWE ビットの書き換え可否を設定するためのビットです。
14	RWNKEY6	0	R/(W)	H'69 : RAMWE7~0 ビットへの書き込み許可。書き込みデータは保持されませんので読み出し値は常に H'00 です。 H'69 以外 : RAMWE7~0 ビットへの書き込み不可
13	RWNKEY5	0	R/(W)	
12	RWNKEY4	0	R/(W)	
11	RWNKEY3	0	R/(W)	
10	RWNKEY2	0	R/(W)	
9	RWNKEY1	0	R/(W)	
8	RWNKEY0	0	R/(W)	
7~5	—	1	R	リザーブビット 読み出すと常に 1 が読み出されます。 書き込む値も常に 1 にしてください。
4	RAMWE4*	1	R/W	RAM 書き込みイネーブル 4 内蔵 RAM のエリア 4 への書き込みを有効/無効にするためのビットです。 0 : 内蔵 RAM のエリア 4 への書き込みが無効 1 : 内蔵 RAM のエリア 4 への書き込みが有効 [クリア条件] • 0 書き込み (上位バイトに H'69 を同時書き込み) [セット条件] • リセット/スタンバイ 1 書き込み (上位バイトに H'69 を同時書き込み)
3	RAMWE3*	1	R/W	RAM 書き込みイネーブル 3 内蔵 RAM のエリア 3 への書き込みを有効/無効にするためのビットです。 0 : 内蔵 RAM のエリア 3 への書き込みが無効 1 : 内蔵 RAM のエリア 3 への書き込みが有効 [クリア条件] • 0 書き込み (上位バイトに H'69 を同時書き込み) [セット条件] • リセット/スタンバイ 1 書き込み (上位バイトに H'69 を同時書き込み)
2	RAMWE2	1	R/W	RAM 書き込みイネーブル 2 内蔵 RAM のエリア 2 への書き込みを有効/無効にするためのビットです。 0 : 内蔵 RAM のエリア 2 への書き込みが無効 1 : 内蔵 RAM のエリア 2 への書き込みが有効 [クリア条件] • 0 書き込み (上位バイトに H'69 を同時書き込み) [セット条件] • リセット/スタンバイ • 1 書き込み (上位バイトに H'69 を同時書き込み)

ビット	ビット名	初期値	R/W	説明
1	RAMWE1	1	R/W	<p>RAM 書き込みイネーブル 1</p> <p>内蔵 RAM のエリア 1 への書き込みを有効／無効にするためのビットです。</p> <p>0 : 内蔵 RAM のエリア 1 への書き込みが無効</p> <p>1 : 内蔵 RAM のエリア 1 への書き込みが有効</p> <p>[クリア条件]</p> <ul style="list-style-type: none"> • 0 書き込み (上位バイトに H'69 を同時書き込み) <p>[セット条件]</p> <ul style="list-style-type: none"> • リセット／スタンバイ • 1 書き込み (上位バイトに H'69 を同時書き込み)
0	RAMWE0	1	R/W	<p>RAM 書き込みイネーブル 0</p> <p>内蔵 RAM のエリア 0 への書き込みを有効／無効にするためのビットです。</p> <p>0 : 内蔵 RAM のエリア 0 への書き込みが無効</p> <p>1 : 内蔵 RAM のエリア 0 への書き込みが有効</p> <p>[クリア条件]</p> <ul style="list-style-type: none"> • 0 書き込み (上位バイトに H'69 を同時書き込み) <p>[セット条件]</p> <ul style="list-style-type: none"> • リセット／スタンバイ • 1 書き込み (上位バイトに H'69 を同時書き込み)

【注】 * 24KB 版ではサポートしません。

19.2.3 RAM ECC イネーブルコントロールレジスタ (RAMECC)

RAMECC は、ECC 誤り訂正の有効/無効を設定するためのレジスタです。RAMECC は、リセット/スタンバイによって初期化されます。

RAMECC への書き込みはワードアクセスのみ可能で、読み出しはワード、バイトのどちらでも可能です。RAMECC への書き込みは、上位バイトを H'76 にし、下位バイトを書き込みデータにしてワードで行ってください。

RAMECC の上位バイト (ビット 15~8) の読み出し時には常に H'00 が読み出されます。

RAMECC へのライト命令の直後に内蔵 RAM をアクセスするような命令を置かないでください。もし内蔵 RAM アクセス命令を置いた場合、正常なアクセスは保証できません。

RECCA ビットを 1 にセットして ECC エラー訂正を無効にする場合、RAMECC へのライト命令の直後に RAMECC のリード命令を置いてください。

ビット	15	14	13	12	11	10	9	8
ビット名	REKEY7	REKEY6	REKEY5	REKEY4	REKEY3	REKEY2	REKEY1	REKEY0
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)

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

【注】 本レジスタは、誤って書き換えられないように、書き込み方法が一般のレジスタと異なっています。詳しくは、「19.2.7 レジスタアクセス時の注意」を参照してください。

ビット	ビット名	初期値	R/W	説明
15	REKEY7	0	R/(W)	RECCA ビットの書き換え可否を設定するためのビットです。 H'76 : RECCA ビットへの書き込み可能。書き込みデータは保持されませんので読み出し値は常に H'00 です。 H'76 以外 : RECCA ビットへの書き込み不可
14	REKEY6	0	R/(W)	
13	REKEY5	0	R/(W)	
12	REKEY4	0	R/(W)	
11	REKEY3	0	R/(W)	
10	REKEY2	0	R/(W)	
9	REKEY1	0	R/(W)	
8	REKEY0	0	R/(W)	
7~1	—	すべて 0	R	リザーブビット 読み出すと常に 0 が読み出されます。書き込む値も常に 0 にしてください。

ビット	ビット名	初期値	R/W	説明
0	RECCA	0	R/W	<p>ECC エラー訂正機能の有効/無効を設定するためのビットです。</p> <p>0 : ECC エラー訂正機能が有効 1 : ECC エラー訂正機能が無効 (パリティエラー検出機能有効)</p> <p>[クリア条件]</p> <ul style="list-style-type: none">• リセット/スタンバイ• 0 書き込み (上位バイトに H'76 を同時書き込み) <p>[セット条件]</p> <ul style="list-style-type: none">• 1 書き込み (上位バイトに H'76 を同時書き込み)

19.2.4 RAM エラーステータスレジスタ (RAMERR)

RAMERR は、RAM ECC イネーブルコントロールレジスタ (RAMECC) により ECC エラー訂正機能を有効に設定したときには ECC エラーの発生の有無をモニタするためのレジスタで、RAMECC イネーブルコントロールレジスタ (RAMECC) により ECC エラー訂正機能を無効に設定したときにはパリティエラーの発生の有無をモニタするためのレジスタです。

RAMERR は、リセット/スタンバイによって初期化されます。書き込み/読み出しはバイトアクセスのみ可能です。

ECC エラー訂正機能が有効に設定された場合の RAM の読み出し時に ECC エラーが発生すると、RERRC ビットがセットされます。2 ビットの誤り検出が発生すると RDTCT がセットされます。また、ECC エラー訂正機能が無効に設定された場合の RAM の読み出し時にパリティエラーが発生すると、RERRC ビットがセットされます。RERRC ビットがセットされた後に RAM ECC イネーブルコントロールレジスタ (RAMECC) の設定で ECC エラー訂正機能を無効にした場合には、RERRC ビットはセットされたままとなります。

RAMERR のビット 7~1 の読み出し時には常に 0 が読み出されます。

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

ビット	ビット名	初期値	R/W	説明
7~2	—	すべて 0	R	リザーブビット 読み出すと常に 0 が読み出されます。書き込む値も常に 0 にしてください。
1	RDTCT	0	R/W	RAM 2 ビット誤り検出モニタビット ECC 誤り訂正有効時に 2 ビット誤り検出の発生有無をモニタするためのビットです。 0: 2 ビット誤り検出が未発生 1: 2 ビット誤り検出が発生 [クリア条件] • リセット/スタンバイ • 1 を読み出した後に 0 を書き込み [セット条件] • 2 ビット誤り検出が発生

ビット	ビット名	初期値	R/W	説明
0	RERRC	0	R/W	ECC エラーまたはパリティエラーの発生の有無をモニタするためのビットです。 0 : クリア後、ECC エラーまたはパリティエラーが未発生 1 : ECC エラーまたはパリティエラーが発生 [クリア条件] <ul style="list-style-type: none">• リセット/スタンバイ• 0 書き込み [セット条件] <ul style="list-style-type: none">• ECC エラーまたはパリティエラーが発生

19.2.5 RAM エラー割り込みコントロールレジスタ (RAMINT)

RAMINT は、RAM エラー割り込みの有効/無効を設定するためのレジスタです。ECC エラーとパリティエラーを合わせて RAM エラーと呼びます。

RAM ECC イネーブルコントロールレジスタ (RAMECC) により ECC エラー訂正機能を有効に設定したときには、RAMINT の RINTC を有効に設定したとき、1 ビット誤り訂正または 2 ビット誤り検出が発生すると割り込みが発生します。また、RAMINT の REDIE を有効に設定したときに、2 ビット誤り検出が発生すると割り込みが発生します。

RAM ECC イネーブルコントロールレジスタ (RAMECC) により ECC エラー訂正機能を無効に設定したときにはパリティエラーが発生した場合に割り込みが発生します。

RAMINT は、リセット/スタンバイによって初期化されます。

RAMINT への書き込み/読み出しはバイトアクセスのみ可能です。

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

ビット	ビット名	初期値	R/W	説明
7~2	—	すべて 0	R	リザーブビット 読み出すと常に 0 が読み出されます。書き込む値も常に 0 にしてください。
1	REDIE	0	R/W	RAM 2 ビット誤り検出割り込みビット ECC 誤り訂正有効時に 2 ビット誤り検出発生時の割り込みの有効/無効を設定するためのビットです。 0 : 2 ビット誤り検出時の割り込みが無効 1 : 2 ビット誤り検出時の割り込みが有効
0	RINTC	0	R/W	RAM エラー割り込みイネーブルビット ECC 誤り訂正有効時に 1 ビット誤り訂正発生時の割り込み、またはパリティエラーの有効/無効を設定するためのビットです。 2 ビット誤り検出時も割り込みが発生します。(RAM1 ビット誤り訂正とパリティエラーをまとめて RAM エラーと呼びます) 0 : RAM エラー割り込みが無効 1 : RAM エラー割り込みが有効 [クリア条件] • リセット/スタンバイ • 0 書き込み [セット条件] • 1 書き込み

19.2.6 RAM アクセスサイクル設定レジスタ (RAMACYC)

RAMACYC は、RAM のリード/ライトサイクルを設定するためのレジスタです。ECC エラー訂正機能の有効/無効によって、サイクル設定が決まりますので推奨の設定値でご使用ください。推奨設定値以外でご使用の場合の動作は保証できません。WRCYC ビットはリセット/スタンバイによって初期化されます。

RAMACYC への書き込みは、上位バイトを H'78 にし、下位バイトを書き込みデータにしてワードで行ってください。書き込みはワードアクセスのみ可能で、読み出しはワード、バイトのどちらでも可能です。

RAMACYC の上位バイト (ビット 15~8) の読み出し時には常に H'00 が読み出されます。

RAM アクセス中には RAMACYC への書き込みを行わないでください。

また、RAMACYC へのライト命令の直後に内蔵 RAM をアクセスするような命令を置かないでください。もし内蔵 RAM アクセス命令を置いた場合、正常なアクセスは保証できません。

WRCYC1、0 ビットを 1 にセットしてアクセスサイクルを設定する場合、RAMACYC へのライト命令の直後に RAMACYC のリード命令を置いてください。

ビット	15	14	13	12	11	10	9	8
ビット名	RAKEY7	RAKEY6	RAKEY5	RAKEY4	RAKEY3	RAKEY2	RAKEY1	RAKEY0
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)

ビット	7	6	5	4	3	2	1	0
ビット名	—	—	WRCYC1	WRCYC0	—	—	—	—
初期値:	0	0	1	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R	R

【注】 本レジスタは、誤って書き換えられないように、書き込み方法が一般のレジスタと異なっています。詳しくは、「19.2.7 レジスタアクセス時の注意」を参照してください。

ビット	ビット名	初期値	R/W	説明
15	RAKEY7	0	R/(W)	WRCYC[1]、WRCYC[0]ビットの書き換え可否を設定するためのビットです。 H'78 : WRCYC[1]、WRCYC[0]ビットへの書き込み可能。 書き込みデータは保持されませんので読み出し値は常に H'00 です。 H'78 以外 : WRCYC[1]、WRCYC[0]ビットへの書き込み不可
14	RAKEY6	0	R/(W)	
13	RAKEY5	0	R/(W)	
12	RAKEY4	0	R/(W)	
11	RAKEY3	0	R/(W)	
10	RAKEY2	0	R/(W)	
9	RAKEY1	0	R/(W)	
8	RAKEY0	0	R/(W)	
7、6	—	すべて 0	R	リザーブビット 読み出すと常に 0 が読み出されます。書き込む値も常に 0 にしてください。

ビット	ビット名	初期値	R/W	説明
5	WRCYC1	1	R/W	RAM のライトサイクルを設定するためのビットです。 00 : 設定禁止 01 : 設定禁止 10 : ライトアクセスを 2 サイクルに設定 (初期値 : ECC エラー訂正機能有効、パリティエラー検出機能無効) 11 : ライトアクセスを 1 サイクルに設定 (ECC エラー訂正機能無効、パリティエラー検出機能有効) [クリア条件] • リセット/スタンバイ • B'00 書き込み (上位バイトに H'78 を同時書き込み) [セット条件] • 上位バイトに H'78 を同時書き込み
4	WRCYC0	0	R/W	
3~0	—	すべて 0	R	リザーブビット 読み出すと常に 0 が読み出されます。書き込む値も常に 0 にしてください。

19.2.7 レジスタアクセス時の注意

RAM イネーブルコントロールレジスタ (RAMEN)、RAM 書き込みイネーブルコントロールレジスタ (RAMWEN)、RAM ECC イネーブルコントロールレジスタ (RAMECC) は、容易に書き換えられないように、書き込み方法が一般のレジスタと異なっています。次の方法で書き込みを行ってください。

RAMEN、RAMWEN、RAMECC、および RAMACYC レジスタへ書き込むときは必ずワードでアクセスしてください。バイトおよびロングワード命令では書き込みません。

図 19.1 示しますように、上位バイトにキーデータを書き込んでください。

- RAMENへ書き込むときは上位バイトをH'96にし、下位バイトを書き込みデータにして転送してください。
- RAMWENへ書き込むときは上位バイトをH'69にし、下位バイトを書き込みデータにして転送してください。
- RAMECCへ書き込むときは上位バイトをH'76にし、下位バイトを書き込みデータにして転送してください。
- RAMACYCへ書き込むときは上位バイトをH'78にし、下位バイトを書き込みデータにして転送してください。

RAMEN、RAMWEN、RAMECC、および RAMACYC の上位バイト (ビット 15~8) の読み出し時には常に H'00 が読み出されます。

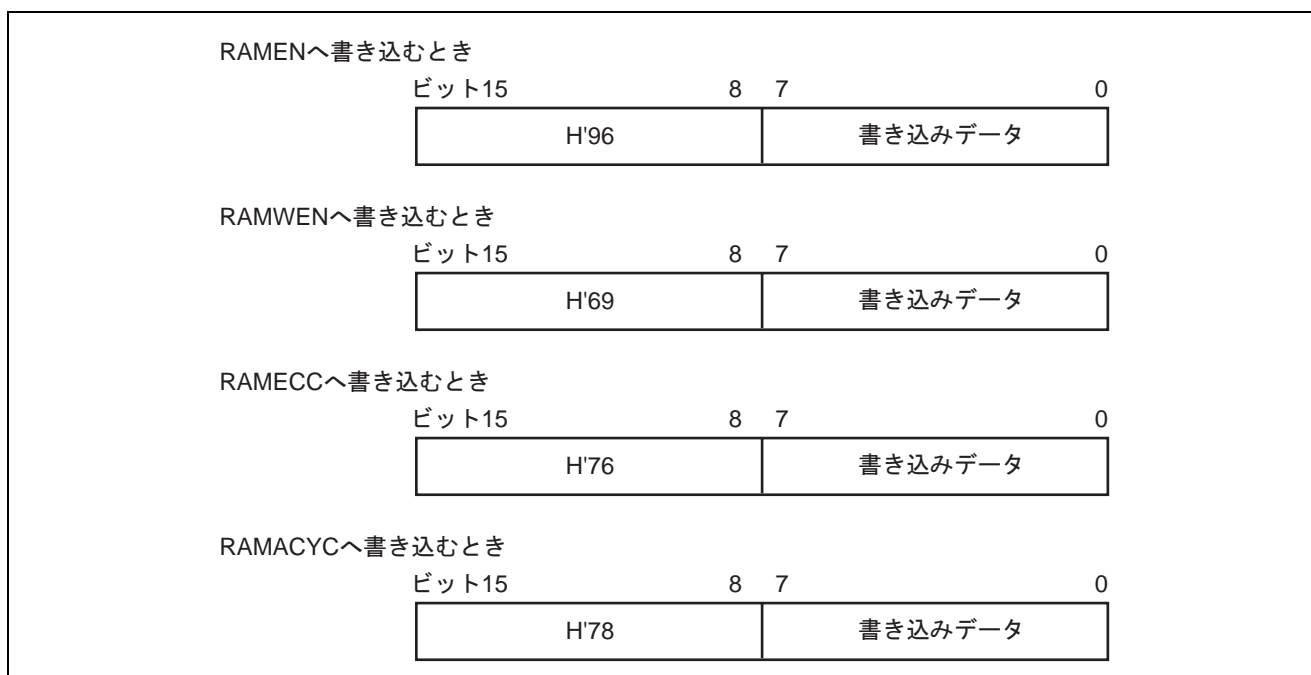


図 19.1 RAMEN、RAMWEN、RAMECC、および RAMACYC レジスタへの書き込み

19.3 動作説明

内蔵 RAM へのアクセスは、RAM イネーブルコントロールレジスタ (RAMEN) および RAM 書き込みイネーブルコントロールレジスタ (RAMWEN) で制御されます。

RAM アクセスのイネーブル/ディスエーブルは、RAM イネーブルコントロールレジスタ (RAMEN) の RAME ビットによって内蔵 RAM アドレスのエリアごとに設定できます。

RAMEN レジスタの RAME ビットを 0 にクリアすると内蔵 RAM はアクセスできません。読み出すと不定値が読み出され、書き込みは無効です。

RAM の書き込みイネーブル/ディスエーブルは、RAM 書き込みイネーブルコントロールレジスタ (RAMWEN) の RAMWE ビットによってエリアごとに設定できます。

レジスタの設定により ECC エラー訂正機能の有効/無効を選択することができます。初期状態では ECC エラー訂正機能は有効状態です。

ECC エラー訂正機能が有効のときに 1 ビットの誤り訂正、2 ビットの誤り検出

が可能です。ライトは 2 サイクル設定でご使用ください。ECC のエラー訂正があった場合、エラー訂正を示すフラグ (RAM エラーステータスレジスタ) を持ちます。RAM エラーステータスレジスタがセットされたときに割り込みを発生することができます。

ECC エラー訂正機能が無効のときにパリティによりエラー検出が可能です。RAM アクセスサイクル設定レジスタ (RAMACYC) の設定をライト 1 サイクル設定でご使用ください。パリティエラー発生時にエラーを示すフラグ (RAM エラーステータスレジスタ) を持ちます。RAM エラーステータスレジスタがセットされたときに割り込みを発生することができます。

割り込み発生のイネーブル/ディスエーブルは、RAM エラー割り込みコントロールレジスタで設定することができます。

19.4 RAM のデータ保持

19.4.1 リセット時のデータ保持

LSI の動作中に外部から $\overline{\text{RES}}$ 端子にローレベル信号を入力すると、LSI はパワーオンリセット状態に遷移します。このとき内蔵 RAM をアクセスしていると、バスサイクルが正常終了しないために当該アドレスの RAM データが破壊されることがあります。

LSI 外部から内蔵 RAM のアクセスを避けてリセットを入力することは困難なため、リセット時にすべてのデータを保持する必要がある場合は RAM イネーブルレジスタ (RAMEN) により、当該 RAM を無効にする必要があります。

19.5 使用上の注意事項

19.5.1 電源立ち上げ後の RAM 初期化

電源立ち上げ後は、ECC エラー訂正データやパリティを含むすべての RAM データが不定値です。そのため、RAM データとエラー訂正データやパリティの値の関係は整合が取れていない場合があります。

電源立ち上げ後には RAM データと誤り訂正データの整合をとる(初期化する)ため、使用するすべての RAM エリアについて何らかのデータをロングワード (32 ビット) で書き込む必要があります。もし、初期化せずに RAM の読み出しを行った場合は、ECC エラーまたはパリティエラーが発生することがあります。

なお、RAM の書き込みでは、ECC エラーまたはパリティエラーが発生することはありません。

20. フラッシュメモリ

H8SX/1727S は 512KB、H8SX/1725S は 256KB のコード格納用フラッシュメモリを内蔵しています。フラッシュメモリの特長を以下に示します。

20.1 特長

- フラッシュメモリマット
ユーザマット：512KB（H8SX/1727S）、256KB（H8SX/1725S）

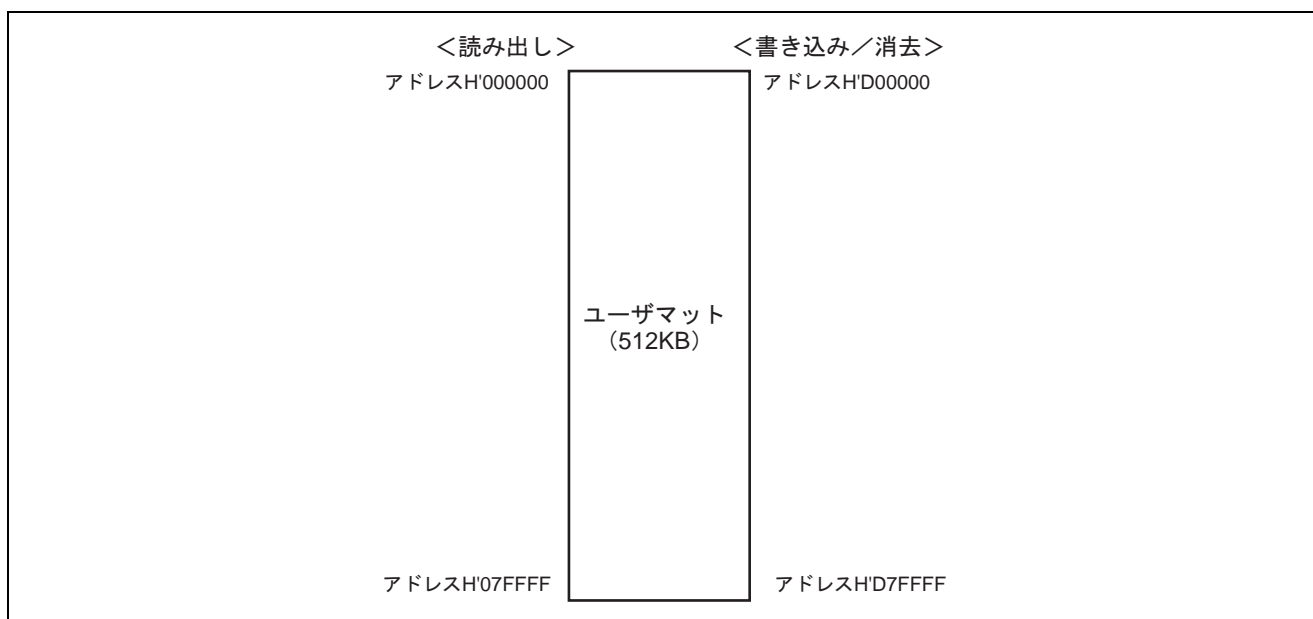


図 20.1 フラッシュメモリのメモリマット構成

- 【注】
- 本文中では 512KB のフラッシュメモリについてのみ記載しています。
 - 256KB のメモリマット構成に関しては「3.3.4 アドレスマップ」をご参照ください。

- 書き込み/消去方式

周辺バス経由でフラッシュメモリ/EEPROM 専用のシーケンサ (FCU) にコマンドを発行することにより、フラッシュメモリの書き込み/消去を実行可能です。FCU がフラッシュメモリの書き込み/消去を実行している期間でも、CPU はフラッシュメモリ以外の領域に配置したプログラムを実行可能です。FCU が EEPROM の書き込み/消去を実行している期間は、フラッシュメモリ領域に配置したプログラムを実行可能です。また、FCU によるフラッシュメモリ書き込み/消去動作を中断して CPU がフラッシュメモリ領域のプログラムを実行した後、フラッシュメモリ書き込み/消去を再開可能です。消去を中断した場合には、中断した消去の対象ではない領域に書き込みを実施することも可能です。

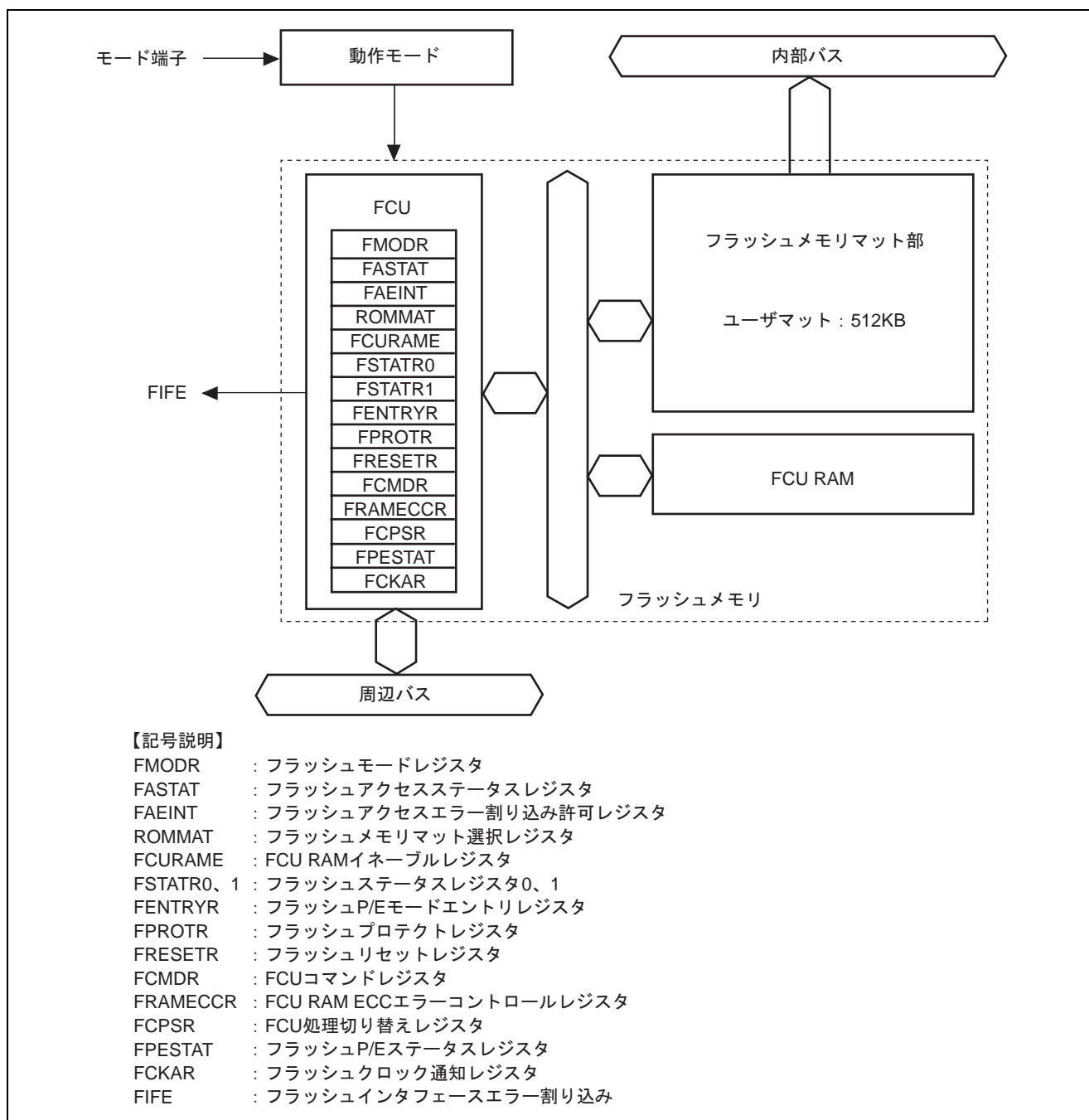


図 20.2 フラッシュメモリのブロック図

- 書き込み/消去単位

ユーザマットの書き込み単位は 128 バイトです。ライターモード以外では、ユーザマットをブロック単位で消去可能です。ライターモードのユーザマット消去単位は、ユーザマット全面です。

図 20.3 にユーザマットのブロック分割を示します。ユーザマットは、H8SX/1727S の場合、4KB (8 ブロック)、32KB (1 ブロック)、64KB (7 ブロック) に、H8SX/1725S の場合、4KB (8 ブロック)、32KB (1 ブロック)、64KB (3 ブロック) に分割されています。

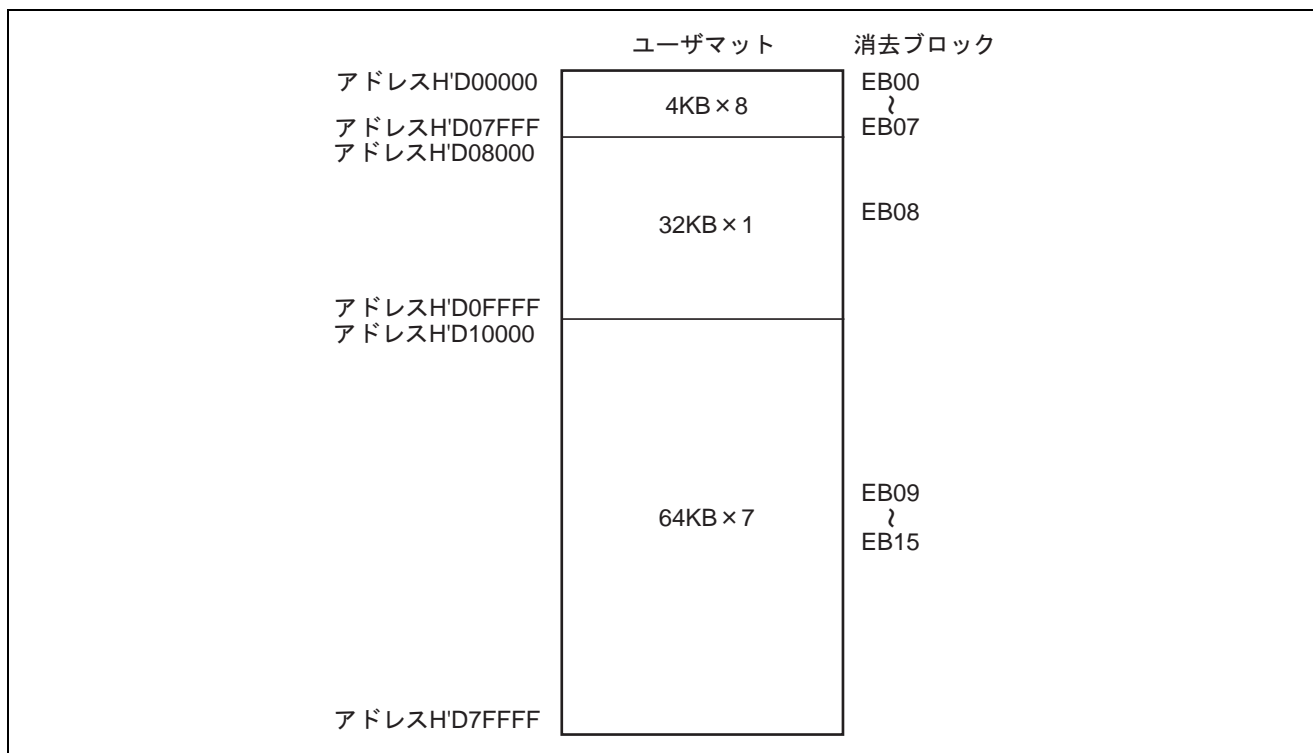


図 20.3 ユーザマットのブロック分割

- オンボードプログラミングモード (2種類)

ブートモード：SCI を使用してユーザマットを書き換え可能なプログラムモードです。ホストと本 LSI 間の SCI 通信のビットレートは自動調整可能です。

ユーザプログラムモード：任意のインタフェースで、ユーザマットを書き換え可能なプログラムモードです。

- オフボードプログラミングモード (1種類)

ライターモード：PROM ライタを使用してユーザマットを書き換え可能なプログラムモードです。

- プロテクトモード

モード端子によるハードウェアプロテクトと FENTRY0 ビット/ロックビットによるソフトウェアプロテクトの2種類のプロテクトモードがあり、書き込み/消去に対するプロテクト状態を設定することができます。FENTRY0 ビットはFCUによるフラッシュメモリ書き込み/消去処理の許可/禁止を制御するためのビットです。ロックビットはユーザマットの各消去ブロック内に1ビットずつ設置されている書き込み/消去プロテクト用のビットです。

書き込み/消去中に異常動作を検出した場合、書き込み/消去処理を中断する機能もあります。

- 書き込み時間/消去時間/書き換え回数

「第 25 章 電気的特性」を参照してください。

20.2 入出力端子

表 20.1 にフラッシュメモリ関連の入出力端子を示します。MD1、MD0 端子の組み合わせによって、フラッシュメモリのプログラミングモードを決定します（「20.4 フラッシュメモリ関連モード概要」を参照）。

表 20.1 端子構成

名称	端子名	入出力	機能
リセット	$\overline{\text{RES}}$	入力	この端子がローレベルになるとリセット状態になります。
モード設定	MD1、MD0	入力	動作モードを決定します。
SCI 受信データ	RxD4	入力	SCI の受信データ (チャンネル 4)
SCI 送信データ	TxD4	出力	SCI の送信データ (チャンネル 4)

20.3 レジスタの説明

表 20.2 にフラッシュメモリ関連のレジスタを示します。一部のレジスタは EEPROM 関連のビットも持ちますが、本章ではフラッシュメモリ機能に関連するビットの説明のみ記載しています。EEPROM 関連ビットの機能の詳細は、「第 21 章 データフラッシュ (EEPROM)」の「21.3 レジスタの説明」を参照してください。

表 20.2 レジスタ構成

レジスタ名	略称	R/W ^{*1}	初期値	アドレス	アクセス サイズ
フラッシュモードレジスタ	FMODR	R/W	H'00	H'FFE002	8
フラッシュアクセスステータスレジスタ	FASTAT	R/(W) ^{*2}	H'00	H'FFE010	8
フラッシュアクセスエラー割り込み許可レジスタ	FAEINT	R/W	H'9F	H'FFE011	8
フラッシュメモリマット選択レジスタ	ROMMAT	R/(W) ^{*3}	H'0000 H'0001	H'FFE020	8、16
FCU RAM イネーブルレジスタ	FCURAME	R/(W) ^{*3}	H'0000	H'FFE054	8、16
フラッシュステータスレジスタ 0	FSTATR0	R ^{*5}	H'80	H'FFE100	8、16
フラッシュステータスレジスタ 1	FSTATR1	R ^{*5}	H'00	H'FFE101	8、16
フラッシュ P/E モードエントリレジスタ	FENTRYR	R/(W) ^{*4*5}	H'0000	H'FFE102	8、16
フラッシュプロテクトレジスタ	FPROTR	R/(W) ^{*4*5}	H'0000	H'FFE104	8、16
フラッシュリセットレジスタ	FRESETR	R/(W) ^{*3}	H'0000	H'FFE106	8、16
FCU コマンドレジスタ	FCMDR	R ^{*5}	H'FFFF	H'FFE10A	8、16
FCU RAM ECC エラーコントロールレジスタ	FRAMECCR	R/W	H'02	H'FFE10C	8
FCU 処理切り替えレジスタ	FCPSR	R/W ^{*5}	H'0000	H'FFE118	8、16
フラッシュ P/E ステータスレジスタ	FPESTAT	R ^{*5}	H'0000	H'FFE11C	8、16
フラッシュロック通知レジスタ	FCKAR	R/W ^{*5}	H'0000	H'FFE138	16

【注】 *1 内蔵 ROM が無効なモードでは、フラッシュメモリ関連レジスタの読み出しデータは 0 に固定され、書き込みは無効化されます（本 LSI には内蔵 ROM 無効モードはありません）。

*2 フラグをクリアするために 0 のみ書き込むことができるビットと読み出し専用ビットによって構成されています。

*3 ワードアクセスで上位バイトに特定の値を書き込んだ場合のみ書き込み有効です。上位バイトへの書き込みデータは保持されません。

*4 ワードアクセスで上位バイトに特定の値を書き込んだ場合のみ書き込み有効で、それ以外の書き込みを行った場合には初期化されます。上位バイトへの書き込みデータは保持されません。

*5 $\overline{\text{RES}}$ 端子からのリセット以外に、FRESETR レジスタの FRESET ビットを 1 にすることも初期化できます。

20.3.1 フラッシュモードレジスタ (FMODER)

FMODER は、FCU の動作モードを指定するレジスタです。

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

ビット	ビット名	初期値	R/W	説明
7~5	—	すべて 0	R	リザーブビット 読み出すと常に 0 が読み出されます。書き込む値も常に 0 にしてください。
4	FRDMD	0	R/W	FCU リードモード選択ビット FCU を使用したフラッシュメモリ/EEPROM 読み出し処理の方法を選択するためのビットです。フラッシュメモリの場合には、ロックビットの確認方法を指定するビットとして使用します。(「20.6.1 FCU コマンド一覧」、「20.6.3 (12) ロックビットの読み出し」を参照)。EEPROM の場合には、ブランクチェックコマンド使用時に設定する必要があります(「第 21 章 データフラッシュ (EEPROM)」を参照)。 0 : メモリ領域リードモード フラッシュメモリロックビットリードモードでフラッシュメモリのロックビットを読み出す場合に、メモリ領域リードモードに設定します。 1 : レジスタリードモード ロックビットリード 2 コマンドを使用してフラッシュメモリのロックビットを読み出す場合に、レジスタリードモードに設定します。
3~0	—	すべて 0	R	リザーブビット 読み出すと常に 0 が読み出されます。書き込む値も常に 0 にしてください。

20.3.2 フラッシュアクセスステータスレジスタ (FASTAT)

FASTAT は、フラッシュメモリ/EEPROM に対するアクセス違反の有無を確認するためのレジスタです。FASTAT レジスタのいずれかのビットが 1 にセットされると、FCU はコマンドロック状態になります(「20.8.3 エラープロテクト」を参照)。コマンドロック状態を解除するためには、FASTAT レジスタを H'10 に設定した後、FCU にステータスクリアコマンドを発行する必要があります。

ビット	7	6	5	4	3	2	1	0
ビット名	ROMAE	—	—	CMDLK	EEPAE	—	EEPRPE	EEPWPE
初期値:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R	R	R	R/(W)*	R	R/(W)*	R/(W)*

【注】 * フラグをクリアするため、1を読み出した後に0を書き込むことのみ可能です。

ビット	ビット名	初期値	R/W	説明
7	ROMAE	0	R/(W)*	<p>フラッシュメモリアクセス違反ビット</p> <p>フラッシュメモリに対するアクセス違反の有無を示すビットです。ROMAE ビットが 1 になると、FSTATR0 レジスタの IGLERR ビットが 1 にセットされ、FCU はコマンドロック状態になります。</p> <p>0: フラッシュメモリアクセスエラーなし 1: フラッシュメモリアクセスエラーあり</p> <p>【セット条件】</p> <ul style="list-style-type: none"> FENTRYR レジスタの FENTRY0 ビットが 1 かつフラッシュメモリ P/E ノーマルモードの状態で、フラッシュメモリ書き込み/消去用アドレス H'D00000~H'D7FFFF に対してリードアクセスを発行 FENTRYR レジスタの FENTRY0 ビットが 0 の状態で、フラッシュメモリ書き込み/消去用アドレス H'D00000~H'D7FFFF に対するアクセスを発行 FENTRYR レジスタが H'0001 の状態で、フラッシュメモリ読み出し用アドレス H'000000~H'07FFFF に対してリードアクセスを発行 <p>【クリア条件】</p> <p>ROMAE=1 を読み出した後に、0 を書き込み</p>
6, 5	—	すべて 0	R	<p>リザーブビット</p> <p>読み出すと常に 0 が読み出されます。書き込む値も常に 0 にしてください。</p>

ビット	ビット名	初期値	R/W	説明
4	CMDLK	0	R	<p>FCU コマンドロックビット</p> <p>FCU がコマンドロック状態であることを示すビットです（「20.8.3 エラープロテクト」を参照）。</p> <p>0 : FCU はコマンドロック状態ではない</p> <p>1 : FCU はコマンドロック状態</p> <p>【セット条件】</p> <ul style="list-style-type: none"> FCU がエラーを検出してコマンドロック状態に遷移後 <p>【クリア条件】</p> <ul style="list-style-type: none"> FASTAT レジスタが H'10 の状態で、FCU がステータスクリアコマンドを処理後
3	EEPAE	0	R/(W)*	<p>EEPROM アクセス違反ビット</p> <p>「第 21 章 データフラッシュ (EEPROM)」を参照してください。</p>
2	—	0	R	<p>リザーブビット</p> <p>読み出すと常に 0 が読み出されます。書き込む値も常に 0 にしてください。</p>
1	EEPRPE	0	R/(W)*	<p>EEPROM リードプロテクト違反ビット</p> <p>「第 21 章 データフラッシュ (EEPROM)」を参照してください。</p>
0	EEPWPE	0	R/(W)*	<p>EEPROM 書き込み/消去プロテクト違反ビット</p> <p>「第 21 章 データフラッシュ (EEPROM)」を参照してください。</p>

【注】 * フラグをクリアするため、1 を読み出した後に 0 を書き込むことのみ可能です。

20.3.3 フラッシュアクセスエラー割り込み許可レジスタ (FAEINT)

FAEINT は、フラッシュインタフェースエラー割り込み (FIFE) の出力許可/禁止を設定するためのレジスタです。

ビット	7	6	5	4	3	2	1	0
ビット名	ROMAEIE	—	—	CMDLKIE	EPAEIE	—	EEPRPEIE	EWPWEIE
初期値:	1	0	0	1	1	1	1	1
R/W:	R/W	R	R	R/W	R/W	R	R/W	R/W

ビット	ビット名	初期値	R/W	説明
7	ROMAEIE	1	R/W	フラッシュメモリアクセス違反割り込みイネーブル フラッシュメモリアクセス違反が発生し、FASTAT レジスタの ROMAE ビットが 1 になった場合の FIFE 割り込み要求の発生を許可/禁止するためのビットです。 0: ROMAE=1 で FIFE 割り込み要求を発生しない 1: ROMAE=1 で FIFE 割り込み要求を発生する
6、5	—	すべて 0	R	リザーブビット 読み出すと常に 0 が読み出されます。書き込む値も常に 0 にしてください。
4	CMDLKIE	1	R/W	FCU コマンドロック割り込みイネーブル FCU コマンドロックが発生し、FASTAT レジスタの CMDLK ビットが 1 になった場合の FIFE 割り込み要求の発生を許可/禁止するためのビットです。 0: CMDLK=1 で FIFE 割り込み要求を発生しない 1: CMDLK=1 で FIFE 割り込み要求を発生する
3	EPAEIE	1	R/W	EEPROM アクセス違反割り込みイネーブル 「第 21 章 データフラッシュ (EEPROM)」を参照してください。
2	—	1	R	リザーブビット 読み出すと常に 1 が読み出されます。書き込む値も常に 1 にしてください。
1	EEPRPEIE	1	R/W	EEPROM リードプロテクト違反割り込みイネーブル 「第 21 章 データフラッシュ (EEPROM)」を参照してください。
0	EWPWEIE	1	R/W	EEPROM 書き込み/消去プロテクト違反割り込みイネーブル 「第 21 章 データフラッシュ (EEPROM)」を参照してください。

20.3.4 フラッシュメモリマット選択レジスタ (ROMMAT)

ROMMAT は、フラッシュメモリのマットを切り替えるために使用するレジスタです。本 LSI にユーザブートマットは存在しません。ROMSEL ビットへ書き込む値は常に 0 にしてください。

ビット	15	14	13	12	11	10	9	8
ビット名	RMKEY7	RMKEY6	RMKEY5	RMKEY4	RMKEY3	RMKEY2	RMKEY1	RMKEY0
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)*
ビット	7	6	5	4	3	2	1	0
ビット名	—	—	—	—	—	—	—	ROMSEL
初期値:	0	0	0	0	0	0	0	0/1
R/W:	R	R	R	R	R	R	R	R/W

【注】 * 書き込みデータは保持されません。

ビット	ビット名	初期値	R/W	説明
15~8	RMKEY7 ~ RMKEY0	すべて 0	R/(W)*	キーコード ROMSEL ビットの書き換えの可否を制御します。本ビットへの書き込みデータは保持されません。
7~1	—	すべて 0	R	リザーブビット 読み出すと常に 0 が読み出されます。書き込む値も常に 0 にしてください。
0	ROMSEL	0/1	R/W	フラッシュメモリマット選択ビット フラッシュメモリのマットを選択するためのビットです。ユーザブートモードで起動した場合には、初期値が 1 になります。それ以外の場合のモードで起動した場合には、初期値が 0 になります。 ROMSEL ビットへの書き込みは、ワードアクセスで RMKEY が H'3B の場合のみ有効です。 0 : ユーザマット選択 1 : ユーザブートマット選択 本 LSI にはユーザブートマットは存在しません。ROMSEL ビットへ書き込む値は常に 0 にしてください。

【注】 * 書き込みデータは保持されません。

20.3.5 FCU RAM イネーブルレジスタ (FCURAME)

FCURAME は、FCU RAM 領域へのアクセスを許可／禁止するためのレジスタです。

ビット	15	14	13	12	11	10	9	8
ビット名	FCKEY7	FCKEY6	FCKEY5	FCKEY4	FCKEY3	FCKEY2	FCKEY1	FCKEY0
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)*
ビット	7	6	5	4	3	2	1	0
ビット名	—	—	—	—	—	—	—	FCRME
初期値:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

【注】 * 書き込みデータは保持されません。

ビット	ビット名	初期値	R/W	説明
15~8	FCKEY7 ~ FCKEY0	すべて 0	R/(W)*	キーコード FCRME ビットの書き換えの可否を制御します。本ビットへの書き込みデータは保持されません。
7~1	—	すべて 0	R	リザーブビット 読み出すと常に 0 が読み出されます。書き込む値も常に 0 にしてください。
0	FCRME	0/1	R/W	FCU RAM イネーブル FCU RAM へのアクセスを許可／禁止するためのビットです。FCRME ビットへの書き込みは、ワードアクセスで FCKEY が H'C4 の場合のみ有効です。FCU RAM に書き込む場合は、FENTRYR レジスタを H'0000 に設定して FCU を停止してください。 0 : FCU RAM へのアクセス禁止 1 : FCU RAM へのアクセス許可

【注】 * 書き込みデータは保持されません。

20.3.6 フラッシュステータスレジスタ 0 (FSTATR0)

FSTATR0 は、FCU の状態を確認するためのレジスタです。FSTATR0 は、 $\overline{\text{RES}}$ 端子からのリセット以外に、FRESETR レジスタの FRESET ビットを 1 にすることでも初期化されます。

ビット	7	6	5	4	3	2	1	0
ビット名	FRDY	ILGLERR	ERSERR	PRGERR	SUSRDY	—	ERSSPD	PRGSPD
初期値:	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

ビット	ビット名	初期値	R/W	説明
7	FRDY	1	R	<p>フラッシュレディビット</p> <p>FCU の処理状態を確認するためのビットです。</p> <p>0: 書き込み/消去処理中 書き込み/消去の中断処理中 ロックビットリード 2 コマンド処理中 EEPROM のブランクチェック処理中 (「第 21 章 データフラッシュ (EEPROM)」を参照)</p> <p>1: 上記の処理を実行していない</p>
6	ILGLERR	0	R	<p>イリーガルコマンドエラービット</p> <p>FCU が不正なコマンドや不正なフラッシュメモリ/EEPROM アクセスなどを検出したことを示すビットです。このビットが 1 の場合には、FCU はコマンドロック状態になります (「20.8.3 エラープロテクト」を参照)。</p> <p>0: FCU は不正なコマンドやフラッシュメモリ/EEPROM アクセスを検出していない</p> <p>1: FCU は不正なコマンドやフラッシュメモリ/EEPROM アクセスを検出した</p> <p>【セット条件】</p> <ul style="list-style-type: none"> FCU が不正なコマンドを検出した FCU が不正なフラッシュメモリ/EEPROM アクセスを検出した (FASTAT レジスタの ROMAE、EEPAE、EEPRPE、EEPWPE ビットのいずれかが 1) FENTRYR の設定が不正 <p>【クリア条件】</p> <ul style="list-style-type: none"> FASTAT レジスタが H'10 の状態で FCU がステータスクリアコマンドを処理後

ビット	ビット名	初期値	R/W	説明
5	ERSERR	0	R	<p>消去エラービット</p> <p>FCUによるフラッシュメモリ/EEPROM消去処理の結果を示すビットです。このビットが1の場合には、FCUはコマンドロック状態になります（「20.8.3 エラープロテクト」を参照）。</p> <p>0：消去処理は正常終了 1：消去処理中にエラー発生</p> <p>【セット条件】</p> <ul style="list-style-type: none"> 消去中にエラーが発生した ロックビットでプロテクトされた領域に対するブロックイレースコマンドを発行した <p>【クリア条件】</p> <ul style="list-style-type: none"> FCUがステータスクリアコマンドを処理後
4	PRGERR	0	R	<p>書き込みエラービット</p> <p>FCUによるフラッシュメモリ/EEPROM書き込み処理の結果を示すビットです。このビットが1の場合には、FCUはコマンドロック状態になります（「20.8.3 エラープロテクト」を参照）。</p> <p>0：書き込み処理は正常終了 1：書き込み処理中にエラー発生</p> <p>【セット条件】</p> <ul style="list-style-type: none"> 書き込み中にエラーが発生した ロックビットでプロテクトされた領域に対する書き込みコマンドを発行した <p>【クリア条件】</p> <ul style="list-style-type: none"> FCUがステータスクリアコマンドを処理後
3	SUSRDY	0	R	<p>サスペンドレディビット</p> <p>FCUがP/Eサスペンドコマンドを受け付け可能であることを示すビットです。</p> <p>0：P/Eサスペンドコマンド受け付け不可能 1：P/Eサスペンドコマンド受け付け可能</p> <p>【セット条件】</p> <ul style="list-style-type: none"> 書き込み/消去処理を開始後、P/Eサスペンドコマンドが受付可能な状態に遷移した <p>【クリア条件】</p> <ul style="list-style-type: none"> P/Eサスペンドコマンドを受け付けた 書き込み/消去中にコマンドロック状態に遷移した
2	—	0	R	<p>リザーブビット</p> <p>読み出すと常に0が読み出されます。書き込む値も常に0にしてください。</p>

ビット	ビット名	初期値	R/W	説明
1	ERSSPD	0	R	<p>消去サスペンドステータスビット</p> <p>FCU が消去の中断処理中または消去サスペンド状態に遷移したことを示すビットです（「20.6.4 サスペンド動作」を参照）。</p> <p>0：下記以外の状態</p> <p>1：消去の中断処理中または消去サスペンド中</p> <p>【セット条件】</p> <ul style="list-style-type: none"> 消去の中断処理を開始した <p>【クリア条件】</p> <ul style="list-style-type: none"> レジュームコマンドを受け付けた
0	PRGSPD	0	R	<p>書き込みサスペンドステータスビット</p> <p>FCU が書き込みの中断処理中または書き込みサスペンド状態に遷移したことを示すビットです（「20.6.4 サスペンド動作」を参照）。</p> <p>0：下記以外の状態</p> <p>1：書き込み中の中断処理中または書き込みサスペンド中</p> <p>【セット条件】</p> <ul style="list-style-type: none"> 書き込みの中断処理を開始した <p>【クリア条件】</p> <ul style="list-style-type: none"> レジュームコマンドを受け付けた

20.3.7 フラッシュステータスレジスタ 1 (FSTATR1)

FSTATR1 は、FCU の状態を確認するためのレジスタです。FSTATR1 は、 $\overline{\text{RES}}$ 端子からのリセット以外に、FRESETR レジスタの FRESET ビットを 1 にすることでも初期化されます。

ビット	7	6	5	4	3	2	1	0
ビット名	FCUERR	—	—	FLOCKST	—	—	PRDTCT	FRCRCT
初期値 :	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R

ビット	ビット名	初期値	R/W	説明
7	FCUERR	0	R	FCU エラービット FCU 内部の CPU 処理においてエラーが発生したことを示すビットです。 0 : FCU の CPU 処理でエラー未発生 1 : FCU の CPU 処理でエラー発生 FCUERR が 1 の場合には、FRESET ビットを 1 にして、FCU を初期化してください。また、FCU ファームウェアを FCU ファーム領域から FCU RAM 領域へ再コピーしてください。
6、5	—	すべて 0	R	リザーブビット 読み出すと常に 0 が読み出されます。書き込む値も常に 0 にしてください。
4	FLOCKST	0	R	ロックビットステータスビット ロックビットリード 2 コマンドを使用した場合に、ロックビットのリードデータが反映されるビットです。ロックビットリード 2 コマンド発行後に、FRDY ビットが 1 になった時点で、FLOCKST ビットに有効なデータが格納されます。FLOCKST ビットの値は、次のロックビットリード 2 コマンドの終了まで保持されます。 0 : プロテクト状態 1 : 非プロテクト状態
3~2	—	すべて 0	R	リザーブビット 読み出すと常に 0 が読み出されます。書き込む値も常に 0 にしてください。

ビット	ビット名	初期値	R/W	説明
1	FRDTCT	0	R	<p>FCU RAM 2 ビット誤り検出モニタビット</p> <p>FCU RAM の読み出し時に 2 ビット誤りを検出したことを示すビットです。</p> <p>0 : 2 ビット誤りを検出していない 1 : 2 ビット誤りを検出した</p> <p>FRDTCT が 1 の場合には、FRESET ビットを 1 にして、FCU を初期化してください。また、FCU ファームウェアを FCU ファーム領域から FCURAM 領域へ再コピーしてください。</p>
0	FRCRCT	0	R	<p>FCU RAM 1 ビット誤り訂正モニタビット</p> <p>FCU RAM の読み出し時に 1 ビット誤りを訂正したことを示すビットです。</p> <p>0 : 1 ビット誤りを訂正していない 1 : 1 ビット誤りを訂正した</p> <p>FRCRCT が 1 の場合には、FRESET ビットを 1 にして、FCU を初期化してください。また、FCU ファームウェアを FCU ファーム領域から FCURAM 領域へ再コピーしてください。</p>

20.3.8 FCU RAM ECC エラーコントロールレジスタ (FRAMECCR)

FRAMECCR は、FCU RAM の読み出し時に 1 ビット誤り訂正、2 ビット誤り検出が発生した場合の FCU コマンドロックへの要求を、有効/無効に設定するためのレジスタです。FRAMECCR は、FCU コマンドロック要求の有効/無効を設定しますがフラッシュステータスレジスタ 1 (FSTATR1) の FRDTCT と FRCRCT のビットセットはコントロールしません。

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

ビット	ビット名	初期値	R/W	説明
7~2	—	すべて 0	R	リザーブビット 読み出すと常に 0 が読み出されます。書き込む値も常に 0 にしてください。
1	FRDCLE	1	R/W	FCU RAM 2 ビット誤り検出 FCU コマンドロックイネーブルビット FCU RAM の読み出し時に 2 ビット誤り検出が発生した場合の FCU コマンドロックへの要求を、有効/無効に設定するためのビットです。本ビットを 1 にして 2 ビット誤りが検出された場合、FASTAT レジスタの CMDLK ビットは 1 にセットされます。 0 : 2 ビット誤り検出時に FCU コマンドロック要求を発生しない 1 : 2 ビット誤り検出時に FCU コマンドロック要求を発生する
0	FRCCLE	0	R/W	FCU RAM 1 ビット誤り訂正 FCU コマンドロックイネーブルビット FCU RAM の読み出し時に 1 ビット誤り訂正が発生した場合の FCU コマンドロックへの要求を、有効/無効に設定するためのビットです。本ビットを 1 にして 1 ビット誤り訂正が発生した場合、FASTAT レジスタの CMDLK ビットは 1 にセットされます。 0 : 1 ビット誤り訂正時に FCU コマンドロック要求を発生しない 1 : 1 ビット誤り訂正時に FCU コマンドロック要求を発生する

20.3.9 フラッシュ P/E モードエントリレジスタ (FENTRYR)

FENTRYR は、フラッシュメモリ/EEPROM を P/E モードに設定するために使用するレジスタです。フラッシュメモリを P/E モードにして FCU のコマンド受付を可能にするためには、FENTRY0 ビットに 1 を設定する必要があります。FENTRYR は、 $\overline{\text{RES}}$ 端子からのリセット以外に、FRESETR レジスタの FRESET ビットを 1 にすることでも初期化されます。

ビット	15	14	13	12	11	10	9	8
ビット名	FEKEY7	FEKEY6	FEKEY5	FEKEY4	FEKEY3	FEKEY2	FEKEY1	FEKEY0
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)*
ビット	7	6	5	4	3	2	1	0
ビット名	FENTRYD	—	—	—	—	—	—	FENTRY0
初期値:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R/W

【注】 * 書き込みデータは保持されません。

ビット	ビット名	初期値	R/W	説明
15~8	FEKEY7 ~ FEKEY0	すべて 0	R/(W)*	キーコード FENTRYD、FENTRY0 ビットの書き換えの可否を制御します。本ビットへの書き込みデータは保持されません。
7	FENTRYD	0	R/W	EEPROM P/E モードエントリビット 「第 21 章 データフラッシュ (EEPROM)」を参照してください。
6~1	—	すべて 0	R	リザーブビット 読み出すと常に 0 が読み出されます。書き込む値も常に 0 にしてください。

ビット	ビット名	初期値	R/W	説明
0	FENTRY0	0	R/W	<p>フラッシュメモリ P/E モードエントリビット 0</p> <p>フラッシュメモリ 512KB (読み出し用アドレス : H'000000~H'07FFFF、書き込み/消去用アドレス : H'D00000~H'D7FFFF) を P/E モードに設定するためのビットです。</p> <p>0 : フラッシュメモリはリードモード 1 : フラッシュメモリは P/E モード</p> <p>【書き込み有効条件】</p> <p>以下の全条件を満たす場合</p> <ul style="list-style-type: none"> • 内蔵 ROM が有効なモード • FSTATR0 レジスタの FRDY ビットが 1 • ワードアクセスで FEKEY に H'AA 書き込み <p>【セット条件】</p> <ul style="list-style-type: none"> • 書き込み有効条件を満たし、かつ FENTRYR レジスタが H'0000 の状態で、FENTRY0 に 1 を書き込んだ場合 <p>【クリア条件】</p> <ul style="list-style-type: none"> • バイトアクセスで書き込んだ場合 • ワードアクセスで FEKEY が H'AA 以外の状態で書き込んだ場合 • 書き込み有効条件を満たした状態で、FENTRY0 に 0 を書き込んだ場合 • 書き込み有効条件を満たし、かつ FENTRYR レジスタが H'0000 以外の状態で、FENTRYR レジスタを書き込んだ場合

【注】 * 書き込みデータは保持されません。

20.3.10 フラッシュプロテクトレジスタ (FPROTR)

FPROTR は、ロックビットによる書き込み/消去プロテクト機能の有効/無効を設定するためのレジスタです。FPROTR は、 $\overline{\text{RES}}$ 端子からのリセット以外に、FRESETR レジスタの FRESET ビットを 1 にすることでも初期化されます。

ビット	15	14	13	12	11	10	9	8
ビット名	FPKEY7	FPKEY6	FPKEY5	FPKEY4	FPKEY3	FPKEY2	FPKEY1	FPKEY0
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)*
ビット	7	6	5	4	3	2	1	0
ビット名	—	—	—	—	—	—	—	FPROTCN
初期値:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

【注】 * 書き込みデータは保持されません。

ビット	ビット名	初期値	R/W	説明
15~8	FPKEY7 ~ FPKEY0	すべて 0	R/(W)*	キーコード FPROTCN ビットの書き換えの可否を制御します。本ビットへの書き込みデータは保持されません。
7~1	—	すべて 0	R	リザーブビット 読み出すと常に 0 が読み出されます。書き込む値も常に 0 にしてください。
0	FPROTCN	0	R/W	ロックビットプロテクトキャンセルビット ロックビットによる書き込み/消去プロテクトを有効/無効化するためのビットです。 0: ロックビットによるプロテクト有効 1: ロックビットによるプロテクト無効 【セット条件】 <ul style="list-style-type: none"> FENTRYR レジスタの値が H'0000 以外の状態で、ワードアクセスで FPKEY に H'55、FPROTCN に 1 を書き込んだ場合 【クリア条件】 <ul style="list-style-type: none"> バイトアクセスで書き込んだ場合 ワードアクセスで FPKEY が H'55 以外の状態で書き込んだ場合 ワードアクセスで FPKEY に H'55、FPROTCN に 0 を書き込んだ場合 FENTRYR レジスタの値が H'0000 の場合

【注】 * 書き込みデータは保持されません。

20.3.11 フラッシュリセットレジスタ (FRESETR)

FRESETR は、FCU を初期化するためのレジスタです。

ビット	15	14	13	12	11	10	9	8
ビット名	FRKEY7	FRKEY6	FRKEY5	FRKEY4	FRKEY3	FRKEY2	FRKEY1	FRKEY0
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)*
ビット	7	6	5	4	3	2	1	0
ビット名	—	—	—	—	—	—	—	FRESET
初期値:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

【注】 * 書き込みデータは保持されません。

ビット	ビット名	初期値	R/W	説明
15~8	FRKEY7 ~ FRKEY0	すべて 0	R/(W)*	キーコード FRESET ビットの書き換えの可否を制御します。本ビットへの書き込みデータは保持されません。
7~1	—	すべて 0	R	リザーブビット 読み出すと常に 0 が読み出されます。書き込む値も常に 0 にしてください。
0	FRESET	0	R/W	フラッシュリセットビット FRESET ビットを 1 に設定すると、フラッシュメモリ/EEPROM の書き込み/消去動作が強制終了され、FCU が初期化されます。書き込み/消去中のフラッシュメモリ/EEPROM のメモリには高電圧が印加されています。メモリに印加された電圧の降下に必要な時間を確保するために、FCU を初期化する場合には、FRESET を 1 にセットした状態を tRESW2 保持してください。FRESET を 1 に保持している期間はフラッシュメモリ/EEPROM へのアクセスは禁止してください。また、FRESET が 1 の状態では、FENTRYR レジスタが初期化されているため、FCU コマンドを使用することはできません。 FRESET ビットへの書き込みは、ワードアクセスで FRKEY が H'CC の場合のみ有効です。 0 : FCU はリセットされない 1 : FCU はリセットされる

【注】 * 書き込みデータは保持されません。

20.3.12 FCU コマンドレジスタ (FCMDR)

FCMDR は、FCU が受け付けたコマンドを格納するレジスタです。FCMDR は、 $\overline{\text{RES}}$ 端子からのリセット以外に、FRESETR レジスタの FRESET ビットを 1 にすることでも初期化されます。

ビット	15	14	13	12	11	10	9	8
ビット名	CMDR	CMDR	CMDR	CMDR	CMDR	CMDR	CMDR	CMDR
初期値 :	1	1	1	1	1	1	1	1
R/W :	R	R	R	R	R	R	R	R
ビット	7	6	5	4	3	2	1	0
ビット名	PCMDR	PCMDR	PCMDR	PCMDR	PCMDR	PCMDR	PCMDR	PCMDR
初期値 :	1	1	1	1	1	1	1	1
R/W :	R	R	R	R	R	R	R	R

ビット	ビット名	初期値	R/W	説明
15~8	CMDR	H'FF	R	コマンドレジスタ FCU が受け付けた最新のコマンドを格納するレジスタです。
7~0	PCMDR	H'FF	R	プレコマンドレジスタ FCU が受け付けた 1 つ前のコマンドを格納するレジスタです。

表 20.3 に各コマンド受け付け後の FCMDR レジスタの状態を示します。

表 20.3 各コマンド受け付け後の FCMDR レジスタの状態

コマンド	CMDR	PCMDR
ノーマルモード移行	H'FF	前回コマンド
ステータスリードモード移行	H'70	前回コマンド
ロックビットリードモード移行 (ロックビットリード 1)	H'71	前回コマンド
フラッシュクロック通知コマンド	H'E9	前回コマンド
プログラム	H'E8	前回コマンド
ブロックイレーズ	H'D0	H'20
P/E サスペンド	H'B0	前回コマンド
P/E レジューム	H'D0	前回コマンド
ステータスレジスタクリア	H'50	前回コマンド
ロックビットリード 2、ブランクチェック	H'D0	H'71
ロックビットプログラム	H'D0	H'77

20.3.13 FCU 処理切り替えレジスタ (FCPSR)

FCPSR は、FCU の消去サスペンドモードを指定するレジスタです。FCPSR は、 $\overline{\text{RES}}$ 端子からのリセット以外に、FRESETR レジスタの FRESET ビットを 1 にすることでも初期化されます。

ビット	15	14	13	12	11	10	9	8
ビット名	—	—	—	—	—	—	—	—
初期値 :	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R
ビット	7	6	5	4	3	2	1	0
ビット名	—	—	—	—	—	—	—	ESUSPMD
初期値 :	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R/W

ビット	ビット名	初期値	R/W	説明
15~1	—	すべて 0	R	リザーブビット 読み出すと常に 0 が読み出されます。書き込む値も常に 0 にしてください。
0	ESUSPMD	0	R/W	消去サスペンドモード FCU がフラッシュメモリ/EEPROM の消去処理を実行中に P/E サスペンドコマンドが発行された場合の消去中断処理モードを選択するためのビットです（「20.6.4 サスペンド動作」を参照）。 0 : サスペンド優先モード 1 : 消去優先モード

20.3.14 フラッシュ P/E ステータスレジスタ (FPESTAT)

FPESTAT は、フラッシュメモリ/EEPROM の書き込み/消去処理結果を示すレジスタです。FPESTAT は、 $\overline{\text{RES}}$ 端子からのリセット以外に、FRESETR レジスタの FRESET ビットを 1 にすることも初期化されます。

ビット	15	14	13	12	11	10	9	8
ビット名	—	—	—	—	—	—	—	—
初期値 :	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R
ビット	7	6	5	4	3	2	1	0
ビット名	PEERRST	PEERRST	PEERRST	PEERRST	PEERRST	PEERRST	PEERRST	PEERRST
初期値 :	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R

ビット	ビット名	初期値	R/W	説明
15~8	—	すべて 0	R	リザーブビット 読み出すと常に 0 が読み出されます。書き込む値は 0 にしてください。
7~0	PEERRST	H'00	R	P/E エラーステータスビット フラッシュメモリ/EEPROM の書き込み/消去処理中にエラーが発生した場合のエラー原因を示すビットです。PEERRST ビットの値は、FSTATR0 レジスタの PRGERR ビットまたは ERSERR ビットが 1 の状態でのみ有効です。ERSERR ビットと PRGERR ビットが 0 の場合の PEERRST ビットには、過去に発生したエラー原因の値が保持されます。 H'01 : ロックビットでプロテクトされた領域に対する書き込みエラー H'02 : ロックビットプロテクト以外の要因による書き込みエラー H'11 : ロックビットでプロテクトされた領域に対する消去によるエラー H'12 : ロックビットプロテクト以外の要因による消去エラー 上記以外 : 予約

20.3.15 フラッシュクロック通知レジスタ (FCKAR)

FCKAR レジスタは、フラッシュメモリ/EEPROM の書き込み/消去時にフラッシュクロック (FCLK) の周波数設定情報をシーケンサに通知するためのレジスタです。FCKAR レジスタは、 $\overline{\text{RES}}$ 端子からのリセット以外に、FRESETR レジスタの FRESET ビットを 1 にすることも初期化できます。

ビット	15	14	13	12	11	10	9	8
ビット名	—	—	—	—	—	—	—	—
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
ビット	7	6	5	4	3	2	1	0
ビット名	FCKA7	FCKA6	FCKA5	FCKA4	FCKA3	FCKA2	FCKA1	FCKA0
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

ビット	ビット名	初期値	R/W	説明
15~6	—	すべて 0	R/W	リザーブビット 読み出すと常に 0 が読み出されます。書き込む値も常に 0 にしてください。
7~0	FCKA[7:0]	すべて 0	R/W	フラッシュクロック通知ビット フラッシュメモリ/EEPROM の書き込み/消去時にフラッシュクロック (FCLK) を設定するためのビットです。書き込み/消去を行う前に FCKA[7:0] ビットに FCLK の周波数を設定して、フラッシュクロック通知コマンドを発行してください。フラッシュメモリ/EEPROM の書き込み/消去中は、周波数を変更しないでください。

FCKA[7:0] ビット (フラッシュクロック通知ビット) の設定値の算出は以下のようにしてください。

1. MHz 単位で表現した動作周波数を 2 進数に変換し、FCKA[7:0] ビットに書き込む。
2. 具体例として周辺クロックの動作周波数が 35.9MHz の場合には次のようになります。
3. 35.9 を切り上げ
4. 36 を 2 進数変換し、上位は H'00 で、下位は H'24 (0010 0100) を FCKA[7:0] ビットに設定する。

- 【注】
1. FCKA[7:0] ビットが 8MHz~40MHz の範囲外に設定された場合は、フラッシュメモリ/EEPROM に対する書き換えコマンドを発行しないでください。
 2. 実周波数と異なる周波数を FCKA[7:0] ビットに設定された場合、ROM/データフラッシュのデータが破壊する可能性があります。
 3. FCKA[7:0] ビットを活用しても、書き換え時間はある程度周波数に依存することを了承ください。

20.4 フラッシュメモリ関連モード概要

図 20.4 に本 LSI のモード遷移図（フラッシュメモリ関連）を示します。MD1、MD0 端子の設定値と本 LSI の動作モードの関係については「第 3 章 MCU 動作モード」を参照してください。

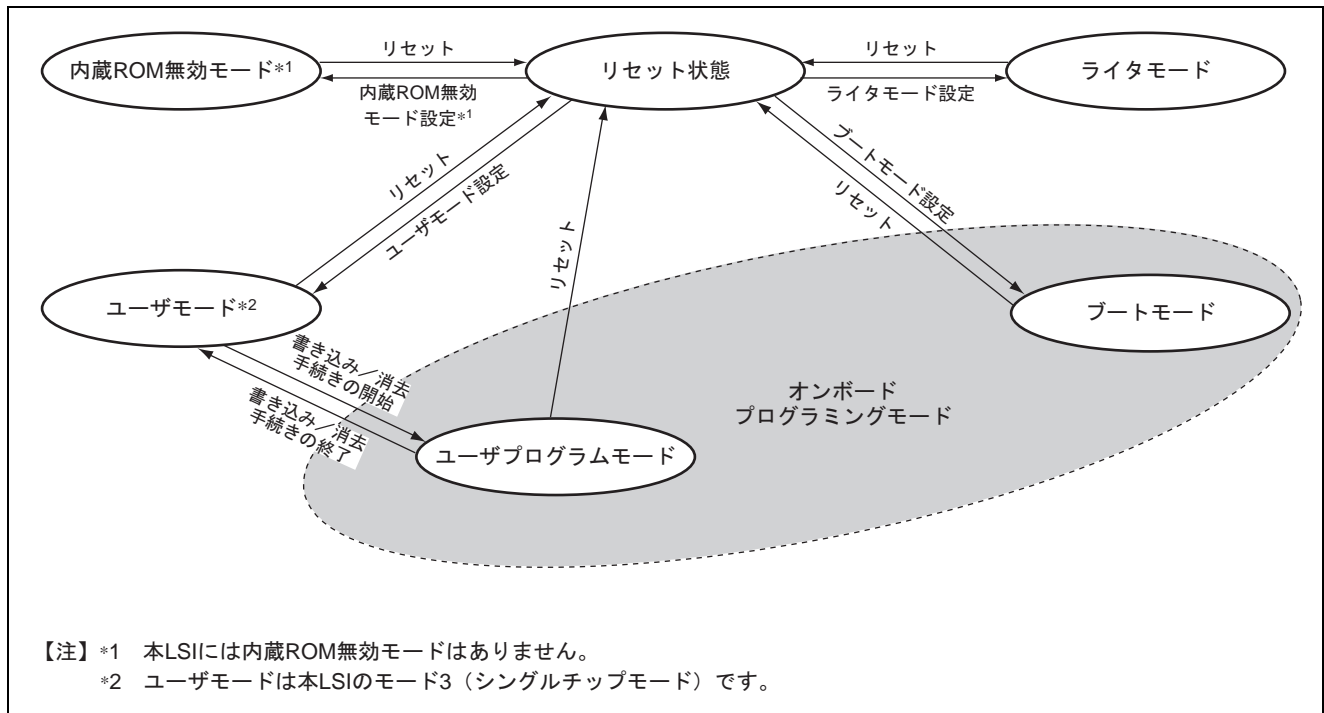


図 20.4 フラッシュメモリに関するモード遷移図

- 内蔵ROM無効モードでは、フラッシュメモリの読み出し／書き込み／消去は実施できません。本 LSI には内蔵 ROM 無効モードはありません。
- ユーザーモードでは、フラッシュメモリの読み出しは可能ですが、書き込み／消去は実施できません。
- ユーザープログラムモード／ブートモードでは、オンボードでフラッシュメモリの読み出し／書き込み／消去を実施できます。
- ライターモードでは、PROMライターを使用してフラッシュメモリの読み出し／書き込み／消去を実施できません。

表 20.4 に、ブートモード、ユーザプログラムモード、ライターモード書き込み／消去関連項目の比較表を示します。

表 20.4 プログラミングモードの比較

項目	ブートモード	ユーザプログラムモード* ²	ライターモード
書き込み／消去環境	オンボード プログラミング		オフボード プログラミング
書き込み／消去可能マツト	ユーザマツト	ユーザマツト	ユーザマツト
書き込み／消去制御	ホスト	FCU	ライター
全面消去	○（自動）	○	○（自動）
ブロック分割消去	○* ¹	○	×
書き込みデータ転送	ホストから SCI 経由	任意のデバイス から RAM 経由	ライター経由
リセット時の起動マツト	組み込みプログラム 格納マツト	ユーザマツト	組み込みプログラム 格納マツト
MCU 動作モードへの遷移	モード設定変更& リセット	書き込み／消去 手続きの終了	—

【注】 *1 起動時に全面消去されます。その後、特定ブロックの消去を実施可能です。

*2 本 LSI では、ユーザモードにて規定の書き込み／消去の手続きを開始したところから手続きの終了したところまでをユーザプログラムモードと規定します。書き込み／消去の手続きは、「20.6 ユーザプログラムモード」を参照してください。

- ブートモードでは、起動直後にユーザマツト／EEPROMのデータマツトが全面消去されます。その後、ホストからSCI経由でのユーザマツト／データマツトの書き込みが可能になります。フラッシュメモリ内容の読み出しも、起動直後の全面消去後に可能になります。
- ブートモード／ライターモードではブートプログラムで内蔵RAMを使用します。このため、RAMイネーブルレジスタ（RAMEN）の設定で内蔵RAMを無効にしてリセットし、ブートモード／ライターモードで起動した場合には、リセット前の内蔵RAMのデータは保持されません（「第19章 RAM」を参照）。

20.5 ブートモード

20.5.1 システム構成

ブートモードでは、ホストから制御コマンドや書き込みデータを送信してユーザマット/ユーザブートマットの書き込み/消去を実行可能です。ホストと本 LSI 間の通信には、本 LSI 内蔵の SCI を調歩同期モードで使用します。ホスト上には制御コマンドを送信するためのツールと書き込みデータを準備する必要があります。本 LSI をブートモードで起動すると、組み込みプログラム格納マット上のプログラムが実行されます。組み込みプログラム格納マット上のプログラムでは、SCI のビットレートの自動調整と制御コマンド方式でのホスト⇄本 LSI 間の通信が実現されます。

図 20.5 にブートモード時のシステム構成を示します。ブートモードでは NMI、 $\overline{\text{IRQ7}} \sim \overline{\text{IRQ0}}$ の割り込みは無視されますが、端子のレベルは非アクティブ状態に固定してください。

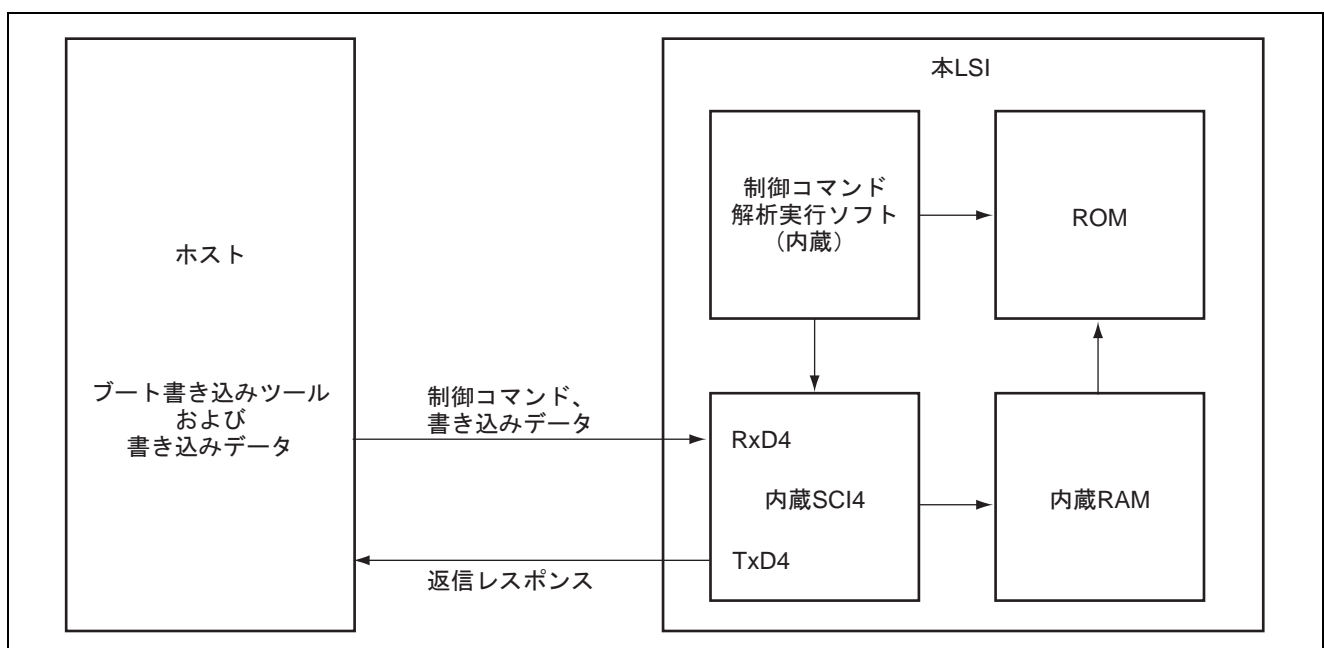


図 20.5 ブートモード時のシステム構成

20.5.2 ブートモードの状態遷移

図 20.6 にブートモードの状態遷移図を示します。

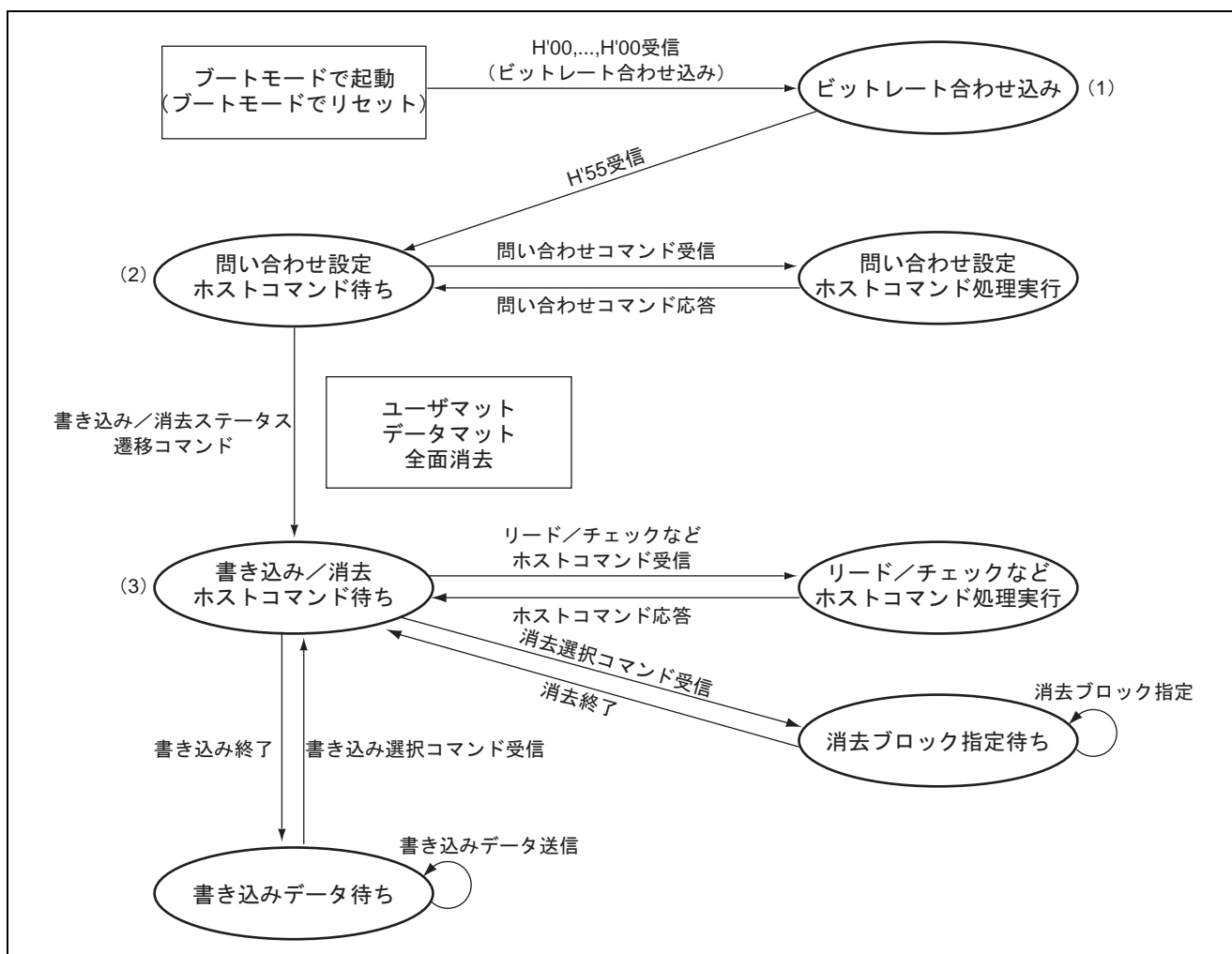


図 20.6 ブートモードの状態遷移図

(1) ビットレート合わせ込み

本 LSI をブートモードで起動すると、ホストと SCI4 のビットレート自動調整を実行します。ビットレートの自動調整が終了すると、本 LSI からホストへ H'00 を送信します。その後、ホストから送信された H'55 を本 LSI が正しく受信すると問い合わせ設定ホストコマンド待ち状態に遷移します。ビットレート合わせ込みの詳細は「20.5.3 ビットレートの自動調整」を参照してください。

(2) 問い合わせ設定ホストコマンド待ち

マットサイズ、マット構成、マット先頭アドレス、サポート状況などの問い合わせや、デバイス、クロックモード、ビットレートを選択するための状態です。ホストから書き込み消去ステータス遷移コマンドを発行すると、本 LSI はユーザマット/EEPROM のデータマットの全面消去を実行し、書き込み/消去ホストコマンド待ち状態に遷移します。問い合わせ設定ホストコマンドの詳細は「20.5.4 問い合わせ設定ホストコマンド待ち状態」を参照してください。

(3) 書き込み/消去ホストコマンド待ち

ホストからのコマンドにしたがって、書き込み/消去を実行する状態です。本 LSI が受信したコマンドに応じて、書き込みデータ待ち状態、消去ブロック指定待ち状態、リード/チェックなどコマンド処理実行状態に遷移します。

本 LSI が書き込み選択コマンドを受信した場合には、書き込みデータ待ち状態に遷移します。ホストから書き込み選択コマンドに続けて、書き込み先頭アドレス、書き込みデータを送信してください。書き込み先頭アドレスを **HFFFFFFF** と設定すると、書き込みが終了して書き込みデータ待ち状態から書き込み/消去コマンド待ち状態に遷移します。

本 LSI が消去選択コマンドを受信すると消去ブロック指定待ち状態に遷移します。ホストから消去選択コマンドに続けて、消去ブロック番号を送信してください。消去ブロック番号を **HFF** と設定すると、消去が終了して消去ブロック指定待ち状態から書き込み/消去ホストコマンド待ち状態に遷移します。ブートモードで起動してから書き込み/消去ホストコマンド待ち状態に遷移する間にユーザマット/EEPROM のデータマットの全面が消去されていますので、ブートモードで新たに書き込んだデータをリセットせずに消去したい場合以外には消去を実行する必要はありません。

書き込み/消去以外に、ユーザマットのサムチェック、ブランクチェック（消去チェック）、メモリリード、ステータス情報取得のためのホストコマンドもあります。書き込み/消去ホストコマンド待ち状態で実行できるホストコマンドの詳細は「**20.5.5 書き込み/消去ホストコマンド待ち状態**」を参照してください。

20.5.3 ビットレートの自動調整

本 LSI をブートモードで起動すると、ホストから連続送信される調歩同期式 SCI 通信のデータ H'00 のローレベル期間を測定します。ローレベル期間測定時のホストの SCI 送受信フォーマットは 8 ビットデータ、1 ストップビット、パリティなし、ビットレートは 9,600bps または 19,200bps に設定してください。本 LSI は測定したローレベル期間からホストの SCI のビットレートを計算し、ビットレート調整が終了すると H'00 をホストへ送信します。ホストが H'00 を正常に受信した場合には、ホストから本 LSI に H'55 を送信してください。H'00 を正常に受信できなかった場合には、本 LSI をブートモードで再起動し、ビットレートの自動調整を再実行してください。本 LSI は H'55 を正常に受信すると H'E6 を送信し、H'55 を正常に受信できなかった場合には H'FF を送信します。

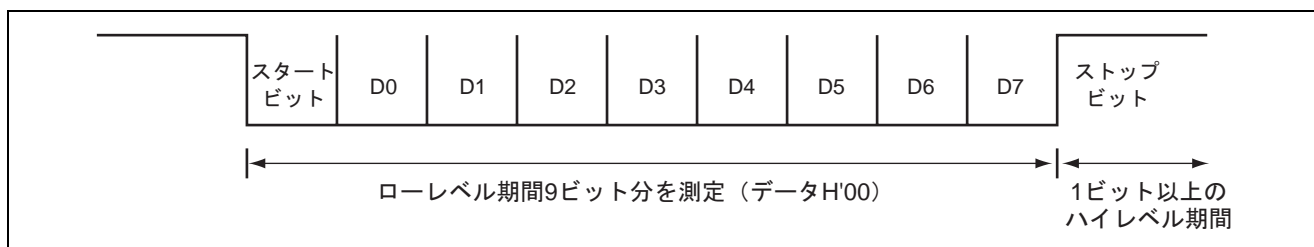


図 20.7 ビットレート自動調整時の SCI 送受信フォーマット

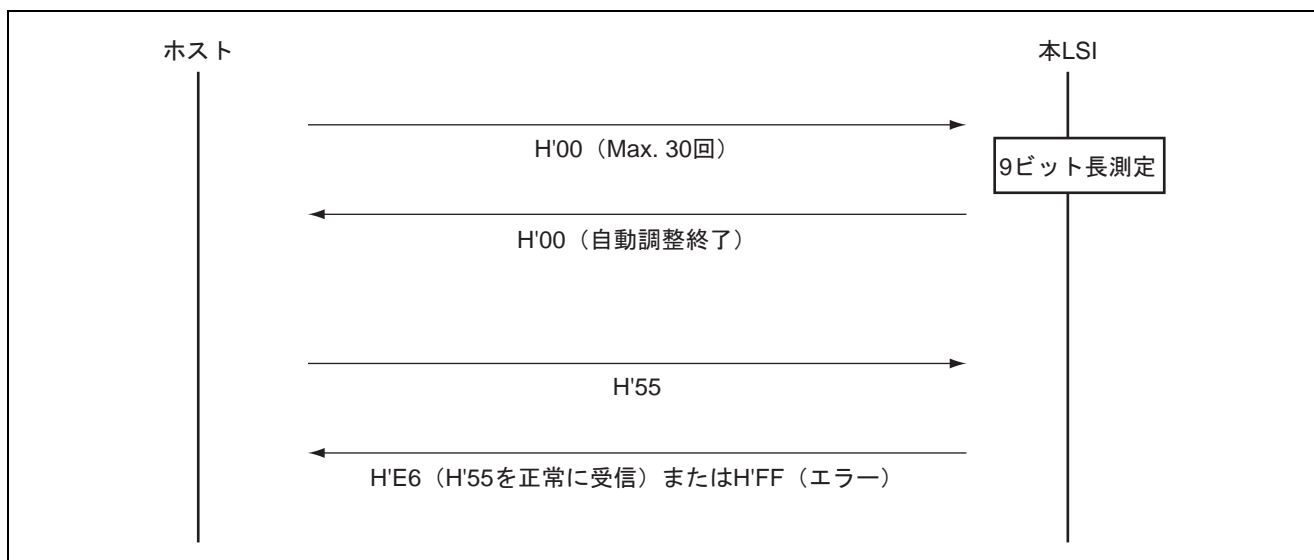


図 20.8 ホストと本 LSI 間の通信シーケンス

ホストの SCI のビットレートや本 LSI の周辺クロックの周波数に依存してビットレートを正常に調整できない場合がありますので、表 20.5 に示した条件で SCI の通信を行うようにしてください。

表 20.5 ビットレート自動調整が可能な条件

ホストのビットレート	本 LSI の周辺クロックの周波数	外部クロック入力周波数
9,600bps	16~20MHz	8~10MHz
19,200bps	16~20MHz	8~10MHz

20.5.4 問い合わせ設定ホストコマンド待ち状態

表 20.6 に、問い合わせ設定ホストコマンド待ち状態で使用可能なホストコマンドの一覧を示します。ブートプログラムステータス問い合わせコマンドは、書き込み/消去ホストコマンド待ち状態でも使用可能です。その他のコマンドは、問い合わせ設定ホストコマンド待ち状態でのみ使用可能です。

表 20.6 問い合わせ設定ホストコマンド

ホストコマンド名	機 能
サポートデバイス問い合わせ	デバイスコードと組み込みプログラム型名の問い合わせ
デバイス選択	デバイスコードの選択
クロックモード問い合わせ	クロックモードの問い合わせ
クロックモード選択	クロックモードの選択
通倍比問い合わせ	クロック種類、通倍比/分周比の種類、通倍比/分周比の問い合わせ
動作周波数問い合わせ	クロック種類、最大/最低動作周波数の問い合わせ
ユーザブートマット情報問い合わせ*	ユーザブートマットの個数、先頭/最終アドレスの問い合わせ
ユーザマット情報問い合わせ	ユーザマットの個数、先頭/最終アドレスの問い合わせ
消去ブロック情報問い合わせ	ブロック数、先頭/最終アドレスの問い合わせ
書き込みサイズ問い合わせ	書き込み時のデータ長の問い合わせ
2面同時書き込み情報問い合わせ	2面同時書き込み可否の問い合わせ
新ビットレート選択	ホスト⇄本 LSI 間の SCI 通信のビットレートを変更
書き込み消去ステータス遷移	ユーザマット/ユーザブートマット/EEPROM のデータマットの全面消去、および書き込み/消去ホストコマンド待ち状態への遷移
ブートプログラムステータス問い合わせ	本 LSI の状態の問い合わせ

【注】 * 本 LSI にはユーザブートマットは存在しません。

ホストが未定義のコマンドを送信した場合は、本 LSI がコマンドエラーのレスポンスを送信します。コマンドエラーのレスポンスの内容は以下の通りです。コマンドには、ホストが送信したコマンドの先頭バイトが格納されています。

エラーレスポンス

H'80

コマンド

問い合わせ設定ホストコマンド待ち状態では、問い合わせコマンドのレスポンスを参考にして、デバイス選択 → クロックモード選択 → 新ビットレート選択の順にホストから選択コマンドを送信し、本 LSI の設定を行ってください。また、サポートデバイス問い合わせ/クロックモード問い合わせ以外の問い合わせコマンドは、クロックモード選択コマンドを発行前には使用できません。誤った順番でコマンドを送信した場合には、本 LSI がコマンドエラーのレスポンスを送信します。図 20.9 に問い合わせ設定ホストコマンド待ち状態でのホストコマンド使用例を示します。

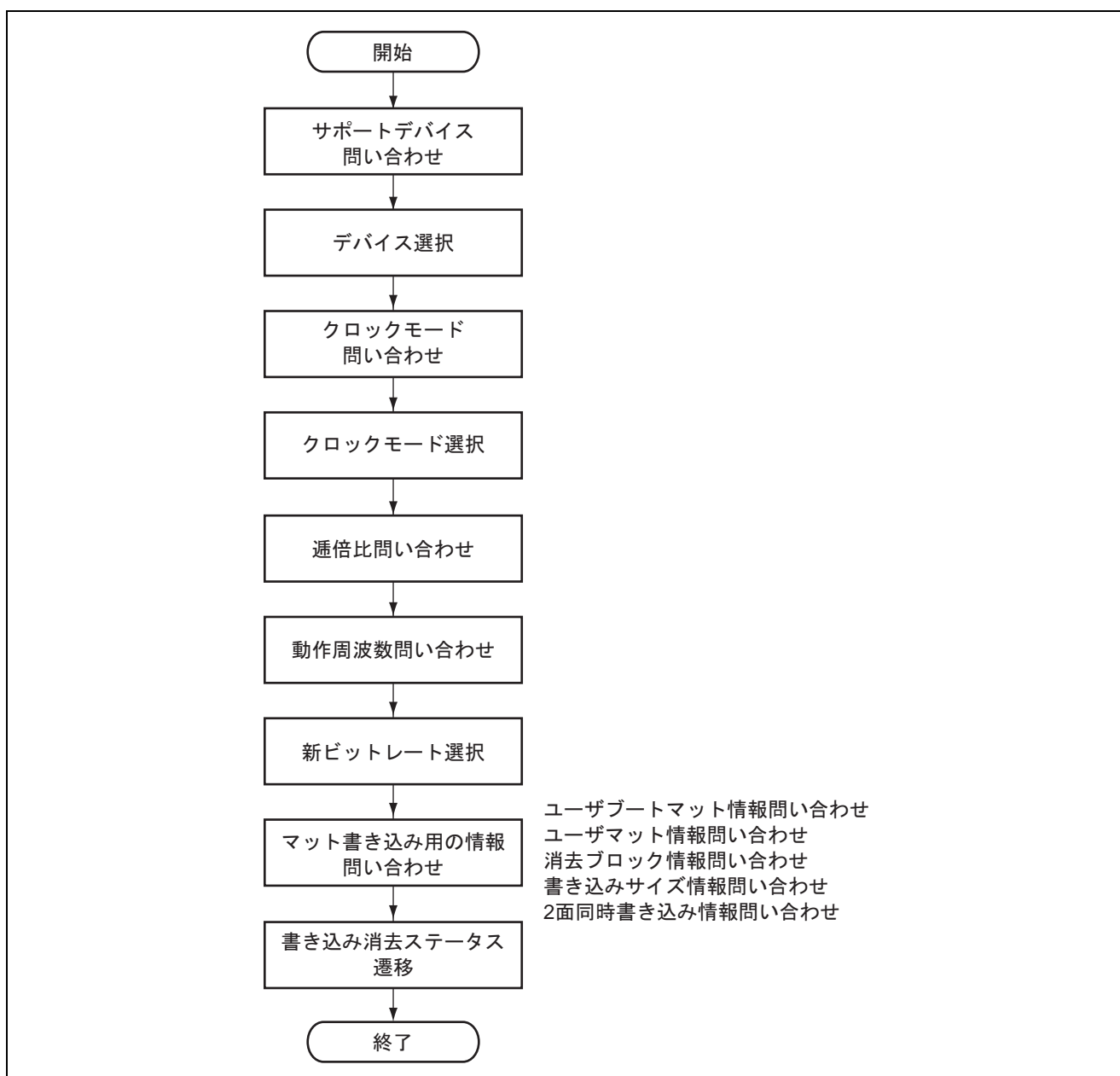


図 20.9 問い合わせ設定ホストコマンドの使用例

各ホストコマンドの詳細を以下に説明します。説明文中の「コマンド」はホストから本 LSI に送信するコマンド、「レスポンス」は本 LSI からホストに送信する応答です。「サムチェック」は、本 LSI が送信した各バイトを合計した場合に、H'00 になるように計算されたバイトデータを指します。

(1) サポートデバイス問い合わせ

ホストがサポートデバイス問い合わせコマンドを送信すると、ブートモード用の組み込みプログラムでサポート可能なデバイス情報を本 LSI が送信します。ホストがデバイスを選択した後に、サポートデバイス問い合わせコマンドを送信した場合には、本 LSI は選択したデバイスの情報のみ送信します。

コマンド	H'20		
レスポンス	H'30	サイズ	デバイス数
	文字数	デバイスコード	
	文字数	デバイスコード	
	
	文字数	デバイスコード	
	SUM		
	品名		
	品名		
	...		
	品名		

【記号説明】

サイズ (1 バイト) : デバイス数、文字数、デバイスコード、品名のデータの総バイト数

デバイス数 (1 バイト) : ブートモード用の組み込みプログラムがサポートする品種数

文字数 (1 バイト) : デバイスコードと品名の文字数

デバイスコード (4 バイト) : チップ品名の ASCII コード

品名 (n バイト) : サポートデバイス名の ASCII コード

SUM (1 バイト) : サムチェック

(2) デバイス選択

ホストがデバイス選択コマンドを送信すると、本 LSI は指定されたデバイスがサポート可能なデバイスかチェックします。サポート可能なデバイスの場合、本 LSI はサポートデバイスを指定したデバイスに変更し、レスポンス (H'06) を送信します。サポート可能なデバイスでなかった場合や、送信されたコマンドが不正であった場合には、本 LSI はエラーレスポンス (H'90) を送信します。

サポートデバイス問い合わせの結果、デバイス数が H'01 であった場合も、デバイス選択コマンドで、問い合わせ結果のデバイスコードの値を設定してください。

コマンド	H'10	サイズ	デバイスコード	SUM
レスポンス	H'06			
エラーレスポンス	H'90	エラー		

【記号説明】

サイズ (1 バイト) : デバイスコードの文字数 (固定値で 4)

デバイスコード (4 バイト) : チップ品名の ASCII コード (サポートデバイス問い合わせコマンドの応答と同一のコード)

SUM (1 バイト) : サムチェック

エラー (1 バイト) : エラーコード

H'11 : サムチェックエラー (コマンドが不正)

H'21 : デバイスコード不一致

(3) クロックモード問い合わせ

ホストがクロックモード問い合わせコマンドを送信すると、選択可能なクロックモードを本 LSI が送信します。ホストがクロックモードを選択した後に、クロックモード問い合わせコマンドを送信した場合には、本 LSI は選択したクロックモードの情報のみ送信します。

コマンド	H'21			
レスポンス	H'31	サイズ		
	モード	モード	...	モード
	SUM			

【記号説明】

サイズ (1 バイト) : モード数、モードのデータの総バイト数

モード (1 バイト) : 選択可能なクロックモード (例 : H'01 クロックモード 1)

SUM (1 バイト) : サムチェック

(4) クロックモード選択

ホストがクロックモード選択コマンドを送信すると、本 LSI は指定されたクロックモードがサポート可能なモードかをチェックします。サポート可能なモードの場合、本 LSI はクロックモードを指定したモードに変更し、レスポンス (H'06) を送信します。サポート可能なモードでなかった場合や、送信されたコマンドが不正であった場合には、本 LSI はエラーレスポンス (H'91) を送信します。

クロックモード選択コマンドは、デバイス選択コマンドを送信した後に送信してください。クロックモード問い合わせの結果、クロックモード数が H'00 または H'01 であった場合も、クロックモード選択コマンドで、問い合わせ結果のモードの値を設定してください。

コマンド	H'11	サイズ	モード	SUM
------	------	-----	-----	-----

レスポンス	H'06
-------	------

エラーレスポンス	H'91	エラー
----------	------	-----

【記号説明】

サイズ (1 バイト) : モードの文字数 (固定値で 1)

モード (1 バイト) : クロックモード (クロックモード問い合わせコマンドの応答と同一のモード)

SUM (1 バイト) : サムチェック

エラー (1 バイト) : エラーコード

H'11 : サムチェックエラー (コマンドが不正)

H'22 : クロックモード不一致

(5) 通倍比問い合わせ

ホストが通倍比問い合わせコマンドを送信すると、クロック種類、通倍比／分周比の種類、通倍比／分周比の情報を本 LSI が送信します。

コマンド	H'22			
レスポンス	H'32	サイズ	クロック数	
	通倍比種類	通倍比	通倍比	・・・ 通倍比
	通倍比種類	通倍比	通倍比	・・・ 通倍比
	・・・	・・・	・・・	・・・
	通倍比種類	通倍比	通倍比	・・・ 通倍比
	SUM			

【記号説明】

サイズ (1 バイト) : クロック数、通倍比種類、通倍比のデータの総バイト数

クロック数 (1 バイト) : クロックの種類 (例: H'02 内部クロックと周辺クロックの 2 種類)

通倍比種類 (1 バイト) : 選択可能な通倍比／分周比の種類

(例: H'03 内部クロックは 4 通倍、6 通倍、8 通倍の 3 種類)

通倍比 (1 バイト) : 通倍比 (例: H'04 = 4 4 通倍) ← 正の数で指定

分周比 (例: HFE = -2 2 分周) ← 負の数で指定

SUM (1 バイト) : サムチェック

(6) 動作周波数問い合わせ

ホストが動作周波数問い合わせコマンドを送信すると、各クロックの動作周波数の最小値と最大値の情報を本 LSI が送信します。

コマンド	H'23		
レスポンス	H'33	サイズ	クロック数
	最小周波数		最大周波数
	最小周波数		最大周波数

	最小周波数		最大周波数
	SUM		

【記号説明】

サイズ (1 バイト) : クロック数、最小周波数、最大周波数のデータの総バイト数

クロック数 (1 バイト) : クロックの種類 (例 : H'02 内部クロックと周辺クロックの 2 種類)

最小周波数 (2 バイト) : 動作周波数の最小値 (例 : H'07D0 20.00MHz)

周波数 (MHz) の小数点第 2 位までの値を 100 倍した値

最大周波数 (2 バイト) : 動作周波数の最大値

書式は最小周波数と同様

SUM (1 バイト) : サムチェック

(7) ユーザブートマット情報問い合わせ

本LSIにはユーザブートマットは存在しませんのでユーザブートマット情報の問い合わせ機能をサポートしていません。ホストがユーザブートマット情報を問い合わせると、ユーザブートマットのエリア数とアドレスの情報を本LSIが送信します。

コマンド	H'24		
レスポンス	H'34	サイズ	エリア数
	エリア先頭アドレス		
	エリア最終アドレス		
	エリア先頭アドレス		
	エリア最終アドレス		
	...		
	エリア先頭アドレス		
	エリア最終アドレス		
	SUM		

【記号説明】

サイズ (1 バイト) : エリア数、エリア先頭アドレス、エリア最終アドレスのデータの総バイト数

エリア数 (1 バイト) : ユーザブートマットのエリア数

(連続したエリアは1 エリアと数えます)

エリア先頭アドレス (4 バイト) : ユーザブートマットエリアの先頭アドレス

エリア最終アドレス (4 バイト) : ユーザブートマットエリアの最終アドレス

SUM (1 バイト) : サムチェック

(8) ユーザマット情報問い合わせ

ホストがユーザマット情報を問い合わせると、ユーザマットのエリア数とアドレスの情報を本 LSI が送信します。

コマンド	H'25		
レスポンス	H'35	サイズ	エリア数
	エリア先頭アドレス		
	エリア最終アドレス		
	エリア先頭アドレス		
	エリア最終アドレス		
	...		
	エリア先頭アドレス		
	エリア最終アドレス		
	SUM		

【記号説明】

サイズ (1 バイト) : エリア数、エリア先頭アドレス、エリア最終アドレスのデータの総バイト数

エリア数 (1 バイト) : ユーザマットのエリア数 (連続したエリアは 1 エリアと数えます)

エリア先頭アドレス (4 バイト) : ユーザマットエリアの先頭アドレス

エリア最終アドレス (4 バイト) : ユーザマットエリアの最終アドレス

SUM (1 バイト) : サムチェック

(9) 消去ブロック情報問い合わせ

ホストが消去ブロック情報を問い合わせると、ユーザマットの消去ブロック数とアドレスの情報を本 LSI が送信します。

コマンド	H'26		
レスポンス	H'36	サイズ	ブロック数
	ブロック先頭アドレス		
	ブロック最終アドレス		
	ブロック先頭アドレス		
	ブロック最終アドレス		
	...		
	ブロック先頭アドレス		
	ブロック最終アドレス		
	SUM		

【記号説明】

サイズ (1 バイト) : ブロック数、ブロック先頭アドレス、ブロック最終アドレスのデータの総バイト数

ブロック数 (1 バイト) : ユーザマットの消去ブロック数

エリア先頭アドレス (4 バイト) : 消去ブロックの先頭アドレス

エリア最終アドレス (4 バイト) : 消去ブロックの最終アドレス

SUM (1 バイト) : サムチェック

(10) 書き込みサイズ問い合わせ

ホストが書き込みサイズを問い合わせると、本 LSI が書き込みサイズの情報を送信します。

コマンド	H'27			
レスポンス	H'37	サイズ	書き込みサイズ	SUM

【記号説明】

サイズ (1 バイト) : 書き込みサイズの文字数 (固定値で 2)

書き込みサイズ (2 バイト) : 書き込み単位 (バイト数単位)

SUM (1 バイト) : サムチェック

(11) 2面同時書き込み情報問い合わせ

ホストが2面同時書き込みを問い合わせると、2面同時書き込みの可否と対象マットの先頭アドレスを本 LSI が送信します。本 LSI では、2面同時に書き込みを行うことができません。

コマンド	H'28		
レスポンス	H'38	サイズ	方式
	1 面目マット先頭アドレス		
	2 面目マット先頭アドレス		
	SUM		

【記号説明】

サイズ (1 バイト) : 方式、1 面目マット先頭アドレス、2 面目マット先頭アドレスのデータの総バイト数 (本 LSI では固定値で 5)

方式 (1 バイト) : 書き込み方式 (本 LSI では H'01)

H'01 : 1 面書き込み、H'10 : 2 面同時書き込み

1 面目マット先頭アドレス (1 バイト) : 1 面目マットの先頭アドレス (本 LSI では H'0000,0000)

2 面目マット先頭アドレス (1 バイト) : 2 面目マットの先頭アドレス (本 LSI では送信されない)

1 面書き込み方式の場合には送信されません

SUM (1 バイト) : サムチェック

(12) 新ビットレート選択

ホストが新ビットレート選択コマンドを送信すると、本 LSI は内蔵 SCI を指定された新ビットレートに設定可能かをチェックします。新ビットレートの設定が可能な場合、本 LSI はレスポンス (H'06) を送信し、SCI を新ビットレートに設定します。新ビットレートの設定ができない場合や、送信されたコマンドが不正であった場合には、本 LSI はエラーレスポンス (H'BF) を送信します。ホストはレスポンス (H'06) を受信すると、新ビットレート選択コマンド送信時のビットレートで 1 ビット期間ウェイトし、ホストのビットレートを新ビットレートに変更します。その後、ホストは新ビットレートで確認用のデータ (H'06) を送信し、本 LSI は確認データをレスポンス (H'06) を送信します。

新ビットレート選択コマンドは、クロックモード選択コマンドを送信した後に送信してください。



図 20.10 新ビットレート選択のシーケンス

コマンド	H'3F	サイズ	ビットレート	入力周波数
	クロック数	通倍比 1	通倍比 2	
	SUM			
レスポンス	H'06			
エラーレスポンス	H'BF	エラー		
確認	H'06			
レスポンス	H'06			

【記号説明】

- サイズ (1 バイト) : ビットレート、入力周波数、クロック数、通倍比のデータの総バイト数
- ビットレート (2 バイト) : 新ビットレート (例 : H'00C0 19200bps)
ビットレート値を 1/100 した値を設定
- 入力周波数 (2 バイト) : 本 LSI の入力周波数 (例 : H'03E8 10.00MHz)
入力周波数の小数点第 2 位までを 100 倍した値を設定
- クロック数 (1 バイト) : クロックの種類 (例 : H'02 内部クロックと周辺クロックの 2 種類)
- 通倍比 1 (1 バイト) : 入力周波数に対する内部クロックの通倍比/分周比
通倍比 (例 : H'04 = 4 4 通倍) ← 正の数で指定
分周比 (例 : HFE = -2 2 分周) ← 負の数で指定
- 通倍比 2 (1 バイト) : 入力周波数に対する周辺クロックの通倍比/分周比
通倍比 1 と同じフォーマット
- SUM (1 バイト) : サムチェック
- エラー : エラーコード H'11 : サムチェックエラー
H'24 : ビットレート選択不可エラー
H'25 : 入力周波数エラー
H'26 : 通倍比エラー
H'27 : 動作周波数エラー

- ビットレート選択不可エラー

新ビットレート選択コマンドで指定したビットレートを、本 LSI の SCI が誤差 4% 未満で設定できない場合にビットレート選択不可エラーが発生します。新ビットレート選択コマンドで指定したビットレートを B、入力周波数を f_{EX} 、通倍比 2 を $M_{P\phi}$ 、SCI4 の BRR_4 レジスタの設定値を N、SMR_4 レジスタの CKS1、CKS0 ビットの設定値を n とした場合のビットレート誤差は、以下の計算式で求められます。

$$\text{誤差 (\%)} = \left\{ \frac{f_{EX} \times M_{P\phi} \times 10^6}{(N+1) \times B \times 64 \times 2^{n-1}} - 1 \right\} \times 100$$

- 入力周波数エラー

新ビットレート選択コマンドで指定した入力周波数が、クロックモード選択コマンドで指定したクロックモードに対応する入力周波数の最小値と最大値の範囲外であった場合に、入力周波数エラーが発生します。

- 通倍比エラー

新ビットレート選択コマンドで指定した通倍比が、クロックモード選択コマンドで指定したクロックモードに対応する通倍比でなかった場合に、通倍比エラーが発生します。選択可能な通倍比を確認するためには通倍比問い合わせコマンドを使用してください。

- 動作周波数エラー

新ビットレート選択コマンドで指定した動作周波数で本 LSI が動作できない場合に動作周波数エラーが発生します。本 LSI は、新ビットレート選択コマンドで指定された入力周波数、通倍比から動作周波数を計算し、計算結果が各クロックの動作周波数の最小値から最大値の範囲内であるかをチェックします。各クロックの動作周波数の最小値と最大値を確認するためには、動作周波数問い合わせコマンドを使用してください。

(13) 書き込み消去ステータス遷移

ホストが書き込み消去ステータス遷移コマンドを送信すると、本 LSI はユーザマット/ユーザブートマット/EEPROM のデータマットを全面消去します。全面消去が完了すると、本 LSI はレスポンス (H'06) を送信し、書き込み消去ホストコマンド待ち状態に遷移します。エラーが発生して消去が完了しなかった場合には、本 LSI はエラーレスポンス (H'C0→H'51) を送信します。

デバイス選択、クロックモード選択、新ビットレート選択を実行する前に、書き込み消去ステータス遷移コマンドを発行しないでください。

コマンド	H'40	
レスポンス	H'06	
エラーレスポンス	H'C0	H'51

(14) ブートプログラムステータス問い合わせ

ホストがブートプログラムステータス問い合わせコマンドを送信すると、本 LSI は現在のステータスを送信します。ブートプログラムステータス問い合わせコマンドは、問い合わせ設定ホストコマンド待ち状態と書き込み消去ホストコマンド待ち状態で使用可能です。

コマンド	H'4F				
レスポンス	H'5F	サイズ	ステータス	エラー	SUM

【記号説明】

サイズ (1 バイト) : ステータス、エラーのデータの総バイト数 (固定値で 2)

ステータス (1 バイト) : 本 LSI の状態 (表 20.7 を参照)

エラー (1 バイト) : 本 LSI のエラー発生状況 (表 20.8 を参照)

SUM (1 バイト) : サムチェック

表 20.7 ステータスの内容

コード	内 容
H'11	デバイス選択待ち
H'12	クロックモード選択待ち
H'13	ビットレート選択待ち
H'1F	書き込み消去ホストコマンド待ち状態への遷移待ち（ビットレート選択完了）
H'31	ユーザマットの消去中
H'3F	書き込み消去ホストコマンド待ち
H'4F	書き込みデータ受信待ち
H'5F	消去ブロック指定待ち

表 20.8 エラーの内容

コード	内 容
H'00	エラーなし
H'11	サムチェックエラー
H'21	デバイスコード不一致エラー
H'22	クロックモード不一致エラー
H'24	ビットレート選択不可エラー
H'25	入力周波数エラー
H'26	通倍比エラー
H'27	動作周波数エラー
H'29	ブロック番号エラー
H'2A	アドレスエラー
H'2B	データ長エラー
H'51	消去エラー
H'52	未消去エラー
H'53	書き込みエラー
H'54	選択処理エラー
H'80	コマンドエラー
H'FF	ビットレート合わせ込み確認エラー

20.5.5 書き込み／消去ホストコマンド待ち状態

表 20.9 に、書き込み／消去ホストコマンド待ち状態で使用可能なホストコマンドの一覧を示します。

表 20.9 書き込み／消去ホストコマンド

ホストコマンド名	機 能
ユーザブートマット書き込み選択*	本 LSI はユーザブートマット書き込みプログラムを選択
ユーザマット書き込み選択	本 LSI はユーザマット書き込みプログラムを選択
ユーザマット 2 面同時書き込み選択	本 LSI はユーザマット 2 面同時書き込みプログラムを選択
128 バイト書き込み	128 バイト書き込み
消去選択	本 LSI は消去用プログラムを選択
ブロック消去	ブロックデータの消去
メモリリード	メモリの読み出し
ユーザブートマットサムチェック*	ユーザブートマットのサムチェック
ユーザマットサムチェック	ユーザマットのサムチェック
ユーザブートマットブランクチェック*	ユーザブートマットのブランクチェック
ユーザマットブランクチェック	ユーザマットのブランクチェック
リードロックビットステータス	ロックビットの読み出し
ロックビットプログラム	ロックビットの書き込み
ロックビット有効	ロックビットプロテクト有効設定
ロックビット無効	ロックビットプロテクト無効設定
ブートプログラムステータス問い合わせ	本 LSI の状態の問い合わせ

【注】 * 本 LSI にはユーザブートマットは存在しません。

ホストが未定義のコマンドを送信した場合は、本 LSI がコマンドエラーのレスポンスを送信します。コマンドエラーの内容は、「20.5.4 問い合わせ設定ホストコマンド待ち状態」を参照してください。

フラッシュメモリの書き込みを実行する場合には、ホストからユーザマット書き込み選択コマンドを送信後、128 バイト書き込みコマンドを送信します。ホストが書き込み選択コマンドを送信すると、本 LSI は書き込みデータ待ち状態になります（「20.5.2 ブートモードの状態遷移」を参照）。書き込みデータ待ちの状態、ホストが 128 バイト書き込みコマンドを送信すると、本 LSI はフラッシュメモリにデータを書き込みます。ホストが書き込み先のアドレスを H'FFFFFFF に設定して 128 バイト書き込みコマンドを送信すると、本 LSI は書き込み終了と判定し、書き込み／消去ホストコマンド待ち状態に遷移します。

フラッシュメモリの消去を実行する場合には、ホストから消去選択コマンドを送信後、ブロック消去コマンドを送信します。ホストが消去選択コマンドを送信すると、本 LSI は消去ブロック指定待ち状態になります（「20.5.2 ブートモードの状態遷移」を参照）。消去ブロック指定待ちの状態、ホストがブロック消去コマンドを送信すると、本 LSI はフラッシュメモリをブロック消去します。ホストがブロック番号に H'FF を設定してブロック消去コマンドを送信すると、本 LSI は消去終了と判定し、書き込み／消去ホストコマンド待ち状態に遷移します。

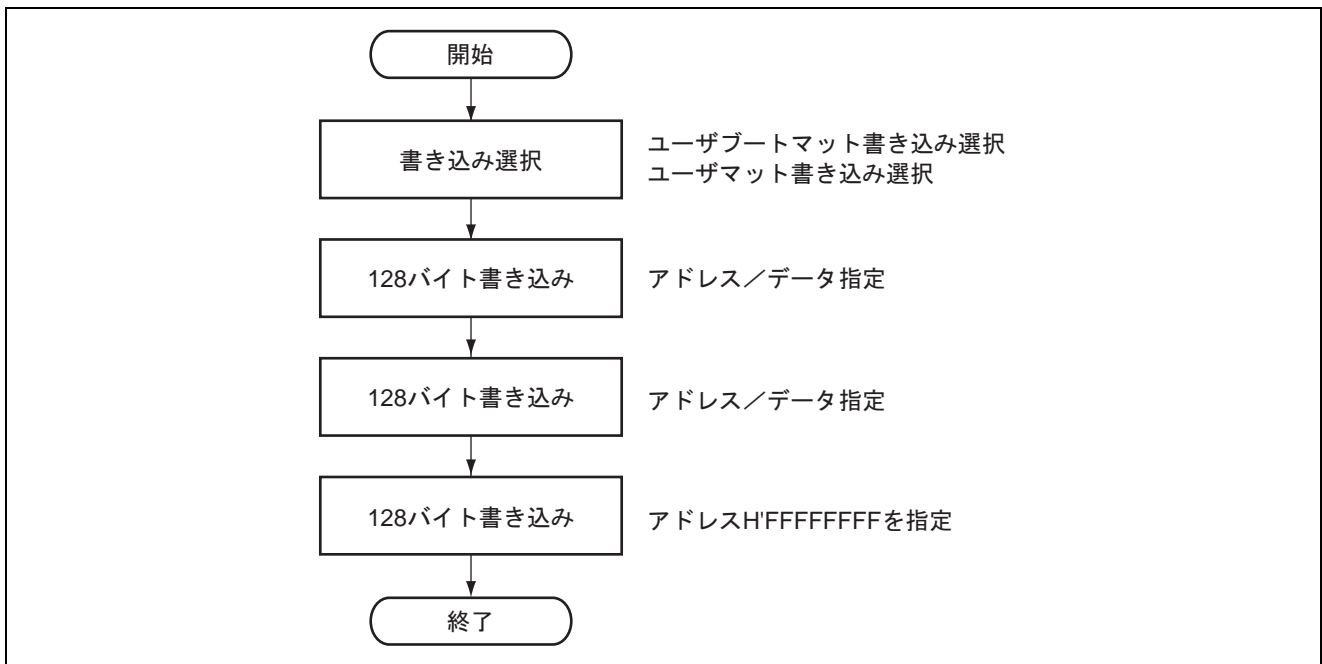


図 20.11 ブートモードでのフラッシュメモリ書き込み方法

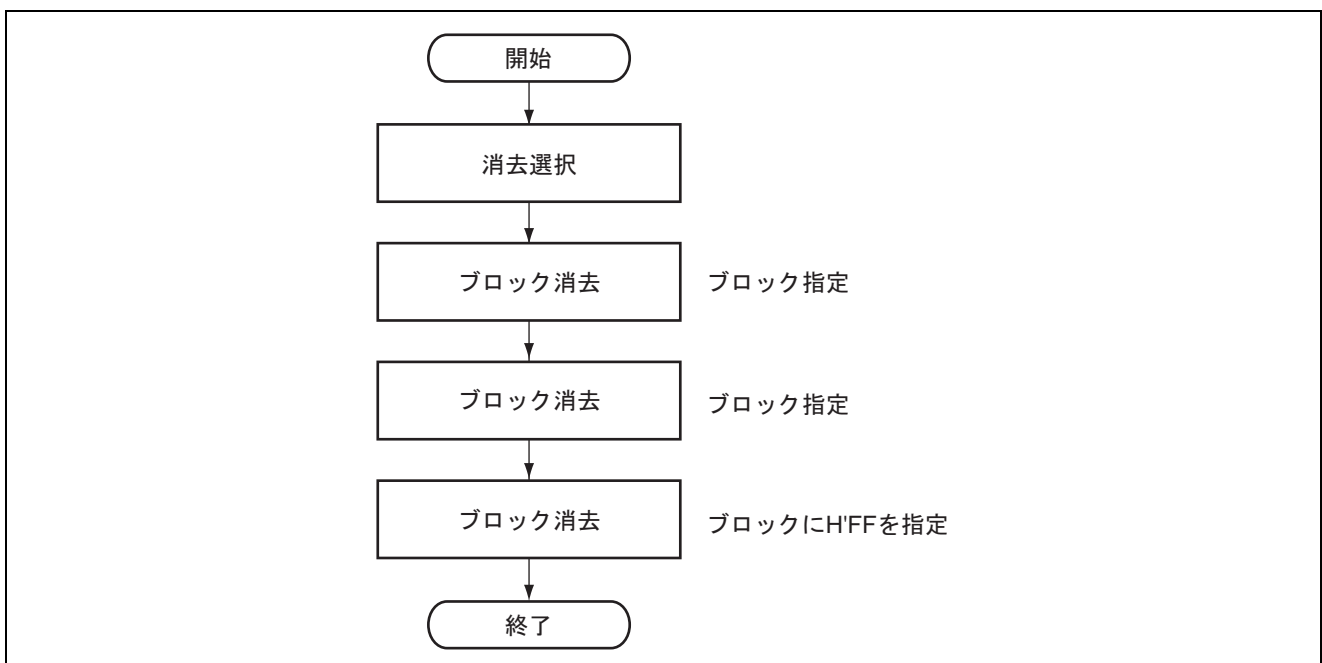


図 20.12 ブートモードでのフラッシュメモリ消去方法

各ホストコマンドの詳細を以下に説明します。説明文中の「コマンド」はホストから本 LSI に送信するコマンド、「レスポンス」は本 LSI からホストに送信する応答です。「サムチェック」は、本 LSI が送信した各バイトを合計した場合に、H'00 になるように計算されたバイトデータを指します。

(1) ユーザブートマット書き込み選択

本 LSI にはユーザブートマットは存在しませんのでユーザブートマット書き込み選択機能をサポートしていません。ホストがユーザブートマット書き込み選択コマンドを送信すると、本 LSI はユーザブートマット書き込みプログラムを選択し、書き込みデータ待ち状態になります。

コマンド

H'42

レスポンス

H'06

(2) ユーザマット書き込み選択

ホストがユーザマット書き込み選択コマンドを送信すると、本 LSI はユーザマット書き込みプログラムを選択し、書き込みデータ待ち状態になります。

コマンド

H'43

レスポンス

H'06

(3) ユーザマット 2 面同時書き込み選択

本 LSI ではユーザマット 2 面同時書き込み機能をサポートしていません。ホストがユーザマット 2 面同時書き込み選択コマンドを送信すると、本 LSI はコマンドエラー (H'80→H'44) を送信します。

コマンド

H'44

レスポンス

H'80	H'44
------	------

(4) 128 バイト書き込み

ホストが 128 バイト書き込みコマンドを送信すると、本 LSI はフラッシュメモリの書き込みを実行します。フラッシュメモリの書き込みが正常に終了すると、本 LSI はレスポンス (H'06) を送信します。書き込み処理中にエラーが発生すると、本 LSI はエラーレスポンス (H'D0) を送信します。

コマンド	H'50		書き込みアドレス	
	データ	データ	...	データ
	SUM			
レスポンス	H'06			
エラーレスポンス	H'D0	エラー		

【記号説明】

書き込みアドレス (4 バイト) : 書き込み先のアドレス

書き込み実行時には 128 バイト境界にアラインしたアドレス

書き込み終了を指定する場合には H'FFFFFFF を送信

データ (128 バイト) : 書き込みデータ

書き込み不要なバイトには H'FF を指定

書き込み終了を指定する場合にはデータの送信は不要 (書き込みアドレス→SUM の順で送信する)

SUM (1 バイト) : サムチェック

エラー (1 バイト) : エラーコード

H'11 : サムチェックエラー

H'2A : アドレスエラー (アドレスが指定のマット内でない)

H'53 : 書き込みエラーが発生し書き込めない

(5) 消去選択

ホストが消去選択コマンドを送信すると、本 LSI は消去プログラムを選択し、消去ブロック指定待ち状態になります。

コマンド	H'48
レスポンス	H'06

(6) ブロック消去

ホストがブロック消去コマンドを送信すると、本 LSI はフラッシュメモリの消去を実行します。フラッシュメモリの消去が正常に終了すると、本 LSI はレスポンス (H'06) を送信します。消去処理中にエラーが発生すると、本 LSI はエラーレスポンス (H'D8) を送信します。

コマンド	H'58	サイズ	ブロック	SUM
レスポンス	H'06			
エラーレスポンス	H'D8	エラー		

【記号説明】

サイズ (1 バイト) : ブロックのデータのバイト数 (固定値で 1)

ブロック (1 バイト) : 消去する消去ブロックの番号

消去終了を指定する場合には H'FF を送信

SUM (1 バイト) : サムチェック

エラー (1 バイト) : エラーコード

H'11 : サムチェックエラー

H'29 : ブロック番号エラー (ブロック番号が正しくない)

H'51 : 消去エラーが発生し消去できない

(7) メモリリード

ホストがメモリリードコマンドを送信すると、本 LSI はフラッシュメモリに対するリードを実行します。正常にリードが実行された場合には、本 LSI はメモリリードコマンドで指定されたアドレスのデータを送信します。リードが実行されなかった場合には、本 LSI はエラーレスポンス (H'D2) を送信します。

コマンド	H'52	サイズ	エリア	読み出し先頭アドレス	
	読み出しサイズ			SUM	
レスポンス	H'52	読み出しサイズ			
	データ	データ	...	データ	
	SUM				
エラーレスポンス	H'D2	エラー			

【記号説明】

サイズ (1 バイト) : エリア、読み出しアドレス、読み出しサイズのデータの総バイト数

エリア (1 バイト) : 読み出し対象のマット

H'00 : ユーザブートマット*

H'01 : ユーザマット

【注】 *本 LSI にはユーザブートマットは存在しません。

読み出し先頭アドレス (4 バイト) : 読み出し対象領域の先頭アドレス

読み出しサイズ (4 バイト) : 読み出すデータのサイズ (バイト単位)

SUM (1 バイト) : サムチェック

データ (1 バイト) : フラッシュメモリから読み出したデータ

エラー (1 バイト) : エラーコード

H'11 : サムチェックエラー

H'2A : アドレスエラー

- ・ エリアの選択で H'00、H'01 以外を指定
- ・ 読み出し先頭アドレスが指定したマットの領域外

H'2B : サイズエラー

- ・ 読み出しサイズの選択で H'00 を指定
- ・ 読み出しサイズがマットのサイズを超えている
- ・ 読み出し先頭アドレスと読み出しサイズから計算されたアドレスがマットの領域外

(8) ユーザブートマットサムチェック

本 LSI にはユーザブートマットは存在しませんのでユーザブートマットサムチェック機能をサポートしていません。ホストがユーザブートマットサムチェックコマンドを送信すると、本 LSI はユーザブートマットのデータをバイト単位で加算した結果（サムチェック）を送信します。

コマンド	H'4A			
レスポンス	H'5A	サイズ	マットのサムチェック	SUM

【記号説明】

サイズ (1 バイト) : マットのサムチェックのバイト数 (固定値で 4)

マットのサムチェック (4 バイト) : ユーザブートマットのサムチェック結果

SUM (1 バイト) : サムチェック (レスポンスデータのサムチェック)

(9) ユーザマットサムチェック

ホストがユーザマットサムチェックコマンドを送信すると、本 LSI はユーザマットのデータをバイト単位で加算した結果（サムチェック）を送信します。

コマンド	H'4B			
レスポンス	H'5B	サイズ	マットのサムチェック	SUM

【記号説明】

サイズ (1 バイト) : マットのサムチェックのバイト数 (固定値で 4)

マットのサムチェック (4 バイト) : ユーザマットのサムチェック結果

ユーザマットにはデバッグ機能認証用のキーコードも含まれています。加算結果にキーコード値が含まれることに注意してください。

SUM (1 バイト) : サムチェック (レスポンスデータのサムチェック)

(10) ユーザブートマットブランクチェック

本 LSI にはユーザブートマットは存在しませんのでユーザブートマットブランクチェック機能をサポートしていません。ホストがユーザブートマットブランクチェックコマンドを送信すると、本 LSI はユーザブートマットがすべて消去状態であるかをチェックします。ユーザブートマットが全て消去状態であった場合には、本 LSI はレスポンス (H'06) を送信します。ユーザブートマットに未消去領域が存在した場合には、本 LSI はエラーレスポンス (H'CC→H'52) を送信します。

コマンド	H'4C	
レスポンス	H'06	
エラーレスポンス	H'CC	H'52

(11) ユーザマットブランクチェック

ホストがユーザマットブランクチェックコマンドを送信すると、本 LSI はユーザマットがすべて消去状態であるかをチェックします。ユーザマットが全て消去状態であった場合には、本 LSI はレスポンス (H'06) を送信します。ユーザマットに未消去領域が存在した場合には、本 LSI はエラーレスポンス (H'CD→H'52) を送信します。

コマンド	H'4D	
レスポンス	H'06	
エラーレスポンス	H'CD	H'52

(12) リードロックビットステータス

ホストがリードロックビットステータスコマンドを送信すると、本 LSI はロックビットに対するリードを実行します。正常にリードが実行された場合には、本 LSI はリードロックビットステータスコマンドで指定されたアドレスのデータを送信します。リードが実行されなかった場合には、本 LSI はエラーレスポンス (H'F1) を送信します。

コマンド	H'71	サイズ	エリア	中位アドレス	上位アドレス	SUM
レスポンス	ステータス					
エラーレスポンス	H'F1	エラー				

【記号説明】

サイズ (1 バイト) : エリア、中位アドレス、上位アドレスのデータの総バイト数 (本 LSI では固定値で 3)

エリア (1 バイト) : 読み出し対象のマット

H'00 : ユーザブートマット*

H'01 : ユーザマット

【注】 *本 LSI にはユーザブートマットは存在しません。

中位アドレス (1 バイト) : 指定ブロックの最後尾のアドレスの中位アドレス (8~15 ビット)

上位アドレス (1 バイト) : 指定ブロックの最後尾のアドレスの上位アドレス (16~23 ビット)

SUM (1 バイト) : サムチェック

ステータス (1 バイト) : ビット 6 が "0" でロック状態

ビット 6 が "1" でアンロック状態

エラー (1 バイト) : エラーコード

H'11 : サムチェックエラー

H'2A : アドレスエラー (アドレスが指定のマット内でない)

(13) ロックビットプログラム

ホストがロックビットプログラムコマンドを送信すると、本 LSI はロックビットの書き込みを行い、指定ブロックをロック状態にします。正常にロックされた場合には、本 LSI はレスポンス (H'06) を送信します。ロックされなかった場合には、本 LSI はエラーレスポンス (H'F7) を送信します。

コマンド	H'77	サイズ	エリア	中位アドレス	上位アドレス	H'D0	SUM
レスポンス	H'06						
エラーレスポンス	H'F7	エラー					

【記号説明】

サイズ (1 バイト) : エリア、中位アドレス、上位アドレス、固定コード (H'D0) のデータの総バイト数 (本 LSI では固定値で 4)

エリア (1 バイト) : ロック対象のマット

H'00 : ユーザブートマット*

H'01 : ユーザマット

【注】 *本 LSI にはユーザブートマットは存在しません。

中位アドレス (1 バイト) : 指定ブロックの最後尾のアドレスの中位アドレス (8~15 ビット)

上位アドレス (1 バイト) : 指定ブロックの最後尾のアドレスの上位アドレス (16~23 ビット)

SUM (1 バイト) : サムチェック

エラー (1 バイト) : エラーコード

H'11 : サムチェックエラー

H'28 : アドレスエラー (アドレスが指定のマット内でない)

H'53 : 書き込みエラーが発生しロック状態にできない

(14) ロックビット有効

ホストがロックビット有効コマンドを送信すると、本 LSI はロックビットを有効にします。

コマンド	H'7A
レスポンス	H'06

(15) ロックビット無効

ホストがロックビット無効コマンドを送信すると、本 LSI はロックビットを無効にします。

コマンド	H'75
レスポンス	H'06

(16) ブートプログラムステータス問い合わせ

詳細については、「20.5.4 問い合わせ設定ホストコマンド待ち状態」を参照してください。

20.6 ユーザプログラムモード

20.6.1 FCU コマンド一覧

ユーザプログラムモードでは、FCU へ FCU コマンドを発行してユーザマットの書き込み/消去を実行します。表 20.10 に、フラッシュメモリ書き込み/消去で使用可能な FCU コマンドの一覧を示します。

表 20.10 FCU コマンド一覧（フラッシュメモリ関連）

コマンド	機能
ノーマルモード移行	ノーマルモードに遷移（「20.6.2 FCU コマンド受け付け条件」を参照）
ステータスリードモード移行	ステータスリードモードに遷移（「20.6.2 FCU コマンド受け付け条件」を参照）
ロックビットリードモード移行 （ロックビットリード1）	ロックビットリードモードに遷移（「20.6.2 FCU コマンド受け付け条件」を参照）
フラッシュロック通知	フラッシュロックの周波数を設定
プログラム	フラッシュメモリ書き込み（128 バイト単位）
ブロックイレーズ	フラッシュメモリ消去（ブロック単位。ロックビットも同時に消去）
P/E サスペンド	書き込み/消去の中断
P/E レジューム	書き込み/消去の再開
ステータスレジスタクリア	FSTATR0 レジスタの IGLERR/ERSERR/PRGERR ビットのクリアとコマンドロック状態の解除
ロックビットリード2	指定した消去ブロックのロックビット読み出し（FSTATR1 レジスタの FLOCKST ビットにロックビットを反映）
ロックビットプログラム	指定した消去ブロックのロックビットを書き込み

ロックビットリード2/ロックビットプログラム以外の FCU コマンドは、EEPROM 書き込み/消去でも使用します。EEPROM に対してロックビットコマンド2を発行した場合には、EEPROM のブランクチェックが実行されます。また、EEPROM に対してロックビットプログラムコマンドを発行した場合には不正コマンド検出によるエラーが発生します。（「第 21 章 データフラッシュ（EEPROM）」を参照）。

FCU へのコマンド発行は、フラッシュメモリ書き込み/消去用のアドレスに対する周辺バスライトアクセスで実現されます。表 20.11 に FCU コマンドのフォーマットを示します。表 20.11 に示した周辺バスライトアクセスを特定条件下で実行すると、FCU は各コマンドに対応した処理を実行します。FCU のコマンド受け付け条件については、「20.6.2 FCU コマンド受け付け条件」を参照してください。各 FCU コマンドの使用方法については、「20.6.3 FCU コマンド使用方法」を参照してください。

FRDMD ビットが 0（メモリ領域リードモード）で FCU コマンドの 1 サイクル目が H'71 の場合には、FCU はロックビットリードモード移行コマンド（ロックビットリード1）を受け付けます。ロックビットリードモードに移行後にフラッシュメモリ書き込み/消去用のアドレスに対して周辺バスリードアクセスを実行すると、アクセス先に対応する消去ブロックのロックビットが読み出しデータの全ビットにコピーされます。FRDMD ビットが 1（レジスタリードモード）で FCU コマンドの 1 サイクル目が H'71 の場合には、FCU はロックビットリード2 コマンドの 2 サイクル目のコマンド（H'D0）待ちの状態になります。この状態で、フラッシュメモリ書き込み/消去用のアドレスに対して周辺バスライトアクセスで H'D0 を書き込むと、FCU はアクセス先に対応する消去ブロックのロックビットを FSTATR1 レジスタの FLOCKST ビットにコピーします。

P/E サスペンドコマンド発行時のサスペンド動作には、サスペンド優先モードと消去優先モードがあります。各モードでの動作の詳細は「20.6.4 サスペンド動作」を参照してください。

表 20.11 FCU コマンドのフォーマット

コマンド	バス サイクル 数	1 サイクル目		2 サイクル目		3 サイクル目		4~5 サイクル目		6 サイクル目		7~66 サイクル目		67 サイクル目	
		アド レス	デー タ	アド レス	デー タ	アド レス	デー タ	アド レス	デー タ	アド レス	デー タ	アド レス	デー タ	アド レス	デー タ
ノーマルモード移行	1	RA	H'FF	-	-	-	-	-	-	-	-	-	-	-	-
ステータスリードモード移行	1	RA	H'70	-	-	-	-	-	-	-	-	-	-	-	-
ロックビットリードモード移行 (ロックビットリード1)	1	RA	H'71	-	-	-	-	-	-	-	-	-	-	-	-
フラッシュロック設定	6	RA	H'E9	RA	H'03	WA	H'0F0F	RA	H'0F0F	RA	H'D0	-	-	-	-
プログラム	67	RA	H'E8	RA	H'40	WA	WD1	RA	WDn	RA	WDn	RA	WDn	RA	H'D0
ブロックイレーズ	2	RA	H'20	BA	H'D0	-	-	-	-	-	-	-	-	-	-
P/E サスペンド	1	RA	H'B0	-	-	-	-	-	-	-	-	-	-	-	-
P/E レジューム	1	RA	H'D0	-	-	-	-	-	-	-	-	-	-	-	-
ステータスレジスタクリア	1	RA	H'50	-	-	-	-	-	-	-	-	-	-	-	-
ロックビットリード2	2	RA	H'71	BA	H'D0	-	-	-	-	-	-	-	-	-	-
ロックビットプログラム	2	RA	H'77	BA	H'D0	-	-	-	-	-	-	-	-	-	-

【記号説明】

RA : フラッシュメモリ書き込み/消去用アドレス

FENTRY0 が 1 の場合 : H'D00000~H'D7FFFF の任意アドレス

WA : フラッシュメモリ書き込み先頭アドレス

書き込みデータ 128 バイトの先頭アドレス

BA : フラッシュメモリ消去ブロックアドレス

対象消去ブロック内の任意アドレス (書き込み/消去用アドレスで指定)

WDn : 書き込みデータ n ワード目 (n=1~64)

20.6.2 FCU コマンド受け付け条件

FCU コマンドの受け付け可否は、FCU のモード／状態に依存します。図 20.13 に FCU のモード遷移図を示します。

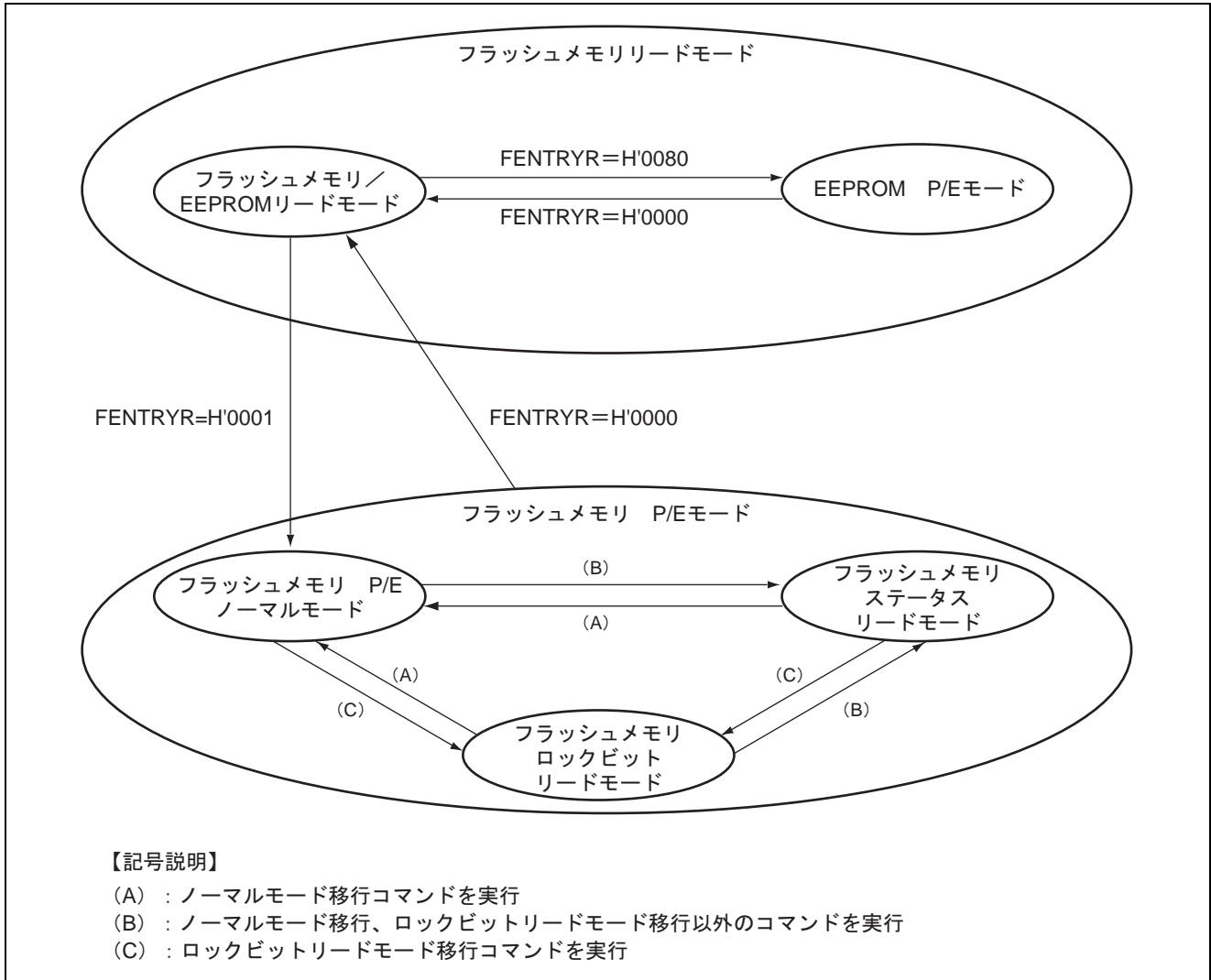


図 20.13 FCU のモード遷移図 (フラッシュメモリ関連)

(1) フラッシュメモリリードモード

• フラッシュメモリ/EEPROMリードモード

内部バス経由でフラッシュメモリを、周辺バス経由で EEPROM を高速読み出し可能なモードです。FCU コマンドは受け付けられません。FENTRYR レジスタの FENTRY0 ビットを 0、かつ FENTRYD ビットを 0 に設定した場合に、このモードに遷移します。

• EEPROM P/Eモード

内部バス経由でフラッシュメモリを高速読み出し可能なモードです。FCU は EEPROM 関連の FCU コマンドを受け付けますが、フラッシュメモリに対する FCU コマンドは受け付けません。FENTRY0 ビットを 0、かつ FENTRYD ビットを 1 に設定した場合に、このモードに遷移します。EEPROM P/E モードの詳細は、「20.6.2 FCU コマンド受け付け条件」を参照してください。

(2) フラッシュメモリ P/E モード

• フラッシュメモリ P/E ノーマルモード

フラッシュメモリリードモード時に FENTRYD ビットを 0、かつ FENTRY0 ビットを 1 に設定した場合、またはフラッシュメモリ P/E モードでノーマルモード移行コマンドを受け付けた場合に遷移するモードです。表 20.12 に受け付け可能なコマンドを示します。フラッシュメモリの高速読み出しは実行できません。FENTRY0 ビットが 1 の状態で H'D00000~H'D7FFFF に対して周辺バスリードアクセスを発行した場合には、フラッシュメモリアクセス違反が発生して FCU はコマンドロック状態になります（「20.8.3 エラープロテクト」を参照）。

• フラッシュメモリステータスリードモード

フラッシュメモリ P/E モードでノーマルモード移行、ロックビットリードモード移行以外のコマンドを受け付けた場合に遷移するモードです。FSTATR0 レジスタの FRDY が 0 の状態やエラー発生後のコマンドロック状態も、フラッシュメモリステータスリードモード中の状態です。表 20.12 に受け付け可能なコマンドを示します。フラッシュメモリの高速読み出しは実行できません。FENTRYR レジスタの値は、フラッシュメモリ P/E ノーマルモードと同じ値です。FENTRY0 ビットが 1 の状態で H'D00000~H'D7FFFF に対して周辺バスリードアクセスを発行した場合には、FSTATR0 レジスタの値が読み出されます。

• フラッシュメモリロックビットリードモード

フラッシュメモリ P/E モードでロックビットリードモード移行コマンドを受け付けた場合に遷移するモードです。表 20.12 に受け付け可能なコマンドを示します。フラッシュメモリの高速読み出しは実行できません。FENTRYR レジスタの値は、フラッシュメモリ P/E ノーマルモードと同じ値です。FENTRY0 ビットが 1 の状態で H'D00000~H'D7FFFF に対して周辺バスリードアクセスを発行した場合には、読み出しデータの全ビットがアクセス先の消去ブロックのロックビット値になります。

表 20.12 にフラッシュメモリ P/E モードの各モード／状態と受け付け可能なコマンドの関係を示します。受け付け不可能なコマンドが発行された場合には、FCU はコマンドロック状態になります（「20.8.3 エラープロテクト」を参照）。

FCU コマンドを確実に受け付けさせたい場合には、発行するコマンドが受け付け可能なモードに移行し、FSTATR0 レジスタの FRDY ビット、ILGLERR ビット、ERSERR ビット、PRGERR ビットと FSTATR1 の FCUERR ビット、FRDTCT ビット、FRCRCT ビットの値を確認した後に FCU コマンドを発行してください。FASTAT レジスタの CMDLK ビットの値は、FSTATR0 レジスタの ILGLERR ビット、ERSERR ビット、PRGERR ビットと FSTATR1 の FCUERR ビット、FRDTCT ビット、FRCRCT ビットの値の論理和です。このため、CMDLK ビットを確認して FCU のエラー発生状況を確認することもできます。表 20.12 ではエラー発生状況を表すビットに CMDLK ビットを使用しています。（ただし、FRAMECCR レジスタの FRDCLE ビットと FRCCLC ビットに 0 が設定されている場合、CMDLK ビットによる FIFE の割り込み要求は、発生しません。）「書き込み／消去の処理中」、「書き込み／消去の中断処理中」、「ロックビットリード 2 処理中」には FSTATR0 レジスタの FRDY ビットが 0 になります。FRDY ビットが 0 の場合で P/E サスペンドコマンドが受け付け可能な状態は、FSTATR0 レジスタの SUSRDY ビットが 1 の場合のみです。

表 20.12 では、表を簡素化するために ERSSPD ビット、PRGSPD ビット、FRDY ビットの値を"0/1"と表記します。ERSSPD ビットは、消去の中断処理中の場合には 1、書き込みの中断処理中には 0 になります。PRGSPD ビットは、書き込みの中断処理中には 1、消去の中断処理中には 0 になります。コマンドロック状態の FRDY ビットの値は、コマンドロック状態に移移する前の FRDY ビットの値が保持されます。

表 20.12 FCU のモード／状態と受け付け可能なコマンドの関係

項 目	P/E ノーマルモード			ステータスリードモード									ロックビットリードモード		
	書き込みサスペンド中	消去サスペンド中	その他の状態	書き込み／消去の処理中	消去サスペンド中の書き込み処理中	書き込み／消去の中断処理中	ロックビットリード 2 処理中	書き込みサスペンド中	消去サスペンド中	(FRDY=0) コマンドロック状態	(FRDY=1) コマンドロック状態	その他の状態	書き込みサスペンド中	消去サスペンド中	その他の状態
FSTATR0 レジスタの FRDY ビット	1	1	1	0	0	0	0	1	1	0	1	1	1	1	1
FSTATR0 レジスタの SUSRDY ビット	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
FSTATR0 レジスタの ERSSPD ビット	0	1	0	0	1	0/1	0/1	0	1	0/1	0/1	0	0	1	0
FSTATR0 レジスタの PRGSPD ビット	1	0	0	0	0	0/1	0/1	1	0	0/1	0/1	0	1	0	0
FASTAT レジスタの CMDLK ビット	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
ノーマルモード移行	○	○	○	×	×	×	×	○	○	×	×	○	○	○	○
ステータスリードモード移行	○	○	○	×	×	×	×	○	○	×	×	○	○	○	○
ロックビットリードモード移行 (ロックビットリード 1)	○	○	○	×	×	×	×	○	○	×	×	○	○	○	○
フラッシュクロック通知	×	×	○	×	×	×	×	×	×	×	×	○	×	×	○
プログラム	×	△	○	×	×	×	×	×	△	×	×	○	×	△	○
ブロックイレーズ	×	×	○	×	×	×	×	×	×	×	×	○	×	×	○
P/E サスペンド	×	×	×	○	×	×	×	×	×	×	×	×	×	×	×
P/E レジューム	○	○	×	×	×	×	×	○	○	×	×	×	○	○	×
ステータスレジスタクリア	○	○	○	×	×	×	×	○	○	×	○	○	○	○	○
ロックビットリード 2	○	○	○	×	×	×	×	○	○	×	×	○	○	○	○
ロックビットプログラム	×	△	○	×	×	×	×	×	△	×	×	○	×	△	○

【記号説明】

○：受け付け可能

△：消去中断したブロック以外への書き込みのみ受け付け可能。

×：受け付け不可能

20.6.3 FCU コマンド使用方法

FCU RAM へのファームウェア転送、および FCU コマンド発行時のユーザ処理フロー例を示します。本節で紹介するフローでは、FCU コマンド発行前の FCU 状態確認は実施せず、フロー終了前にコマンド実行結果を確認している場合があります。FCU コマンドを確実に受け付けさせたい場合には、フロー開始前に FCU 状態を確認してください（「20.6.2 FCU コマンド受け付け条件」を参照）。

本節で紹介するフローでは、FSTATR0 レジスタの FRDY ビット、ILGLERR ビット、ERSERR ビット、PRGERR ビット、SUSRDY ビット、ERSSPD ビット、PRGSPD ビットと FSTATR1 の FCUERR ビット、FRDTCT ビット、FRCRCT ビットを使用して、FCU のコマンド処理状況とエラー発生状況を確認しています。FSTATR0 レジスタと FSTATR1 レジスタはワードアクセスで同時に読み出し可能であるため、1 回のレジスタアクセスで FCU の状態を確認できます。FSTATR0 レジスタの FRDY ビットと FSTATR1 レジスタの CMDLK ビットを使用して FCU の状態を確認する方法を使用する場合には、2 回のレジスタアクセスが必要になりますが、CMDLK ビットのみでエラーの有無が確認可能です。

FCU がコマンド処理中に FCUERR ビットが 1 または FRDTCT ビット、FRCRCT ビットが 1 にセットされてコマンドロック状態に遷移した場合には、FRDY ビットは 0 を保持します。コマンドロック状態では FCU の処理が停止するため、FRDY ビットが 0 から 1 にセットされることはありません。書き込み/消去時間やサスペンド遅延時間（「第 25 章 電気的特性」を参照）よりも長時間にわたって FRDY ビットが 0 に保持される場合は、コマンドロック状態で FCU の処理が停止する等の異常動作が発生している可能性があるため、FCU のリセットによって FCU を初期化してください。FCU のコマンド処理が完了して FRDY ビットが 1 にセットされた場合には、FCUERR ビットと FRDTCT ビット、FRCRCT ビットは必ず 0 の状態です。このため、コマンド処理完了後のエラー発生状況は、ILGLERR ビット、ERSERR ビット、PRGERR ビットで確認可能です。

以下にFCUコマンドの概略フローを示します。詳細については次頁を参照してください。

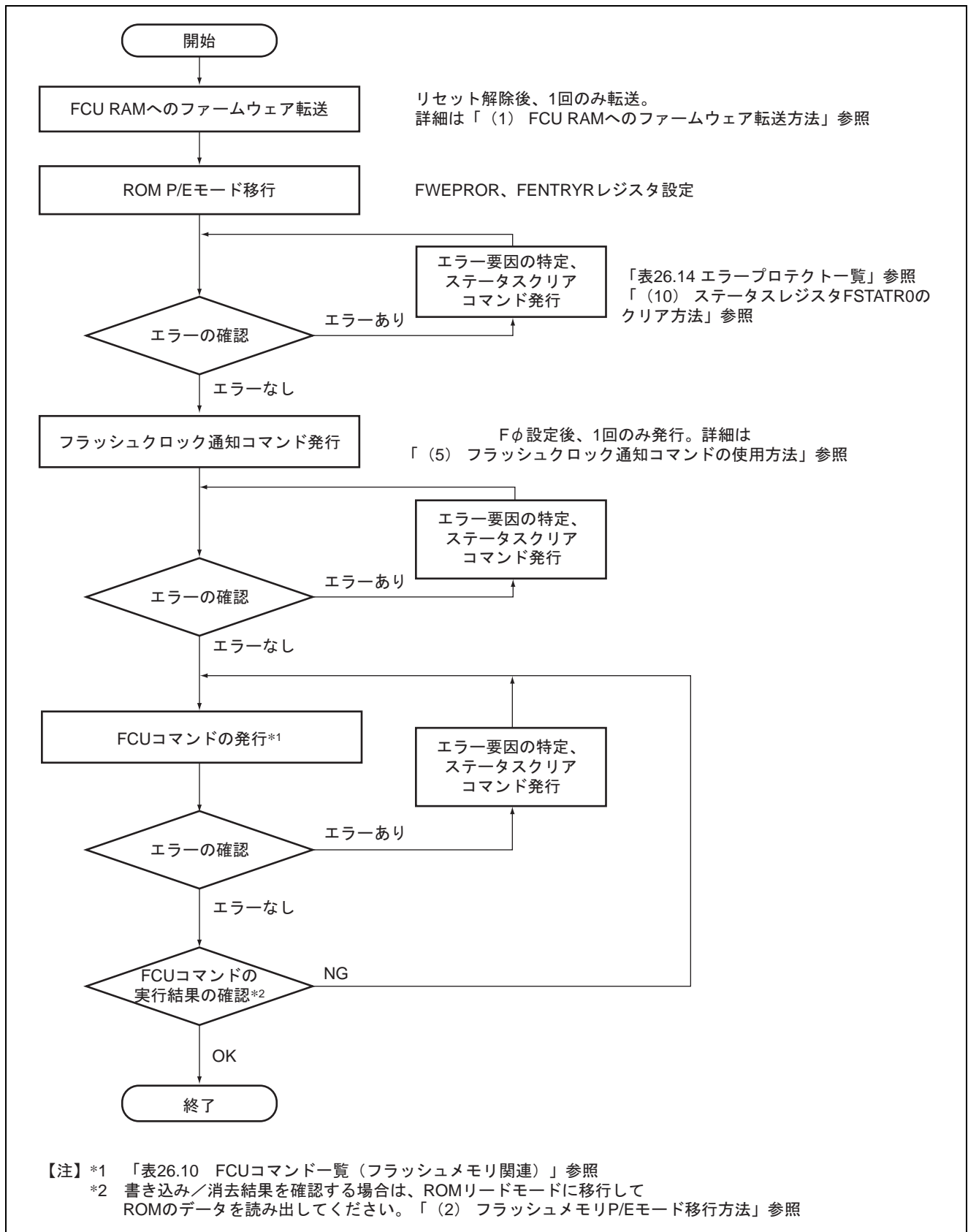


図 20.14 FCU コマンドの概略フロー

(1) FCU RAM へのファームウェア転送方法

FCU コマンドを使用するためには、FCU RAM に FCU 用のファームウェアを格納する必要があります。チップ起動時には FCU RAM に FCU のファームウェアが格納されていないため、FCU ファーム領域に格納された FCU ファームウェアを FCU RAM にコピーする必要があります。また、FSTATR1 レジスタの FCUERR ビットまたは FRDTCT ビット、FRCRCT ビットが 1 の場合には、FCU RAM に格納されたファームウェアが破壊されている可能性があるため、FCU をリセットし FCU ファームを再コピーする必要があります。

図 20.15 に FCU RAM へのファームウェア転送フローを示します。FCU RAM にデータを書き込む場合には、FENTRYR レジスタを H'0000 に設定して FCU を停止してください。DMAC 設定方法の詳細は、「第 7 章 DMA コントローラ (DMAC)」を参照してください。

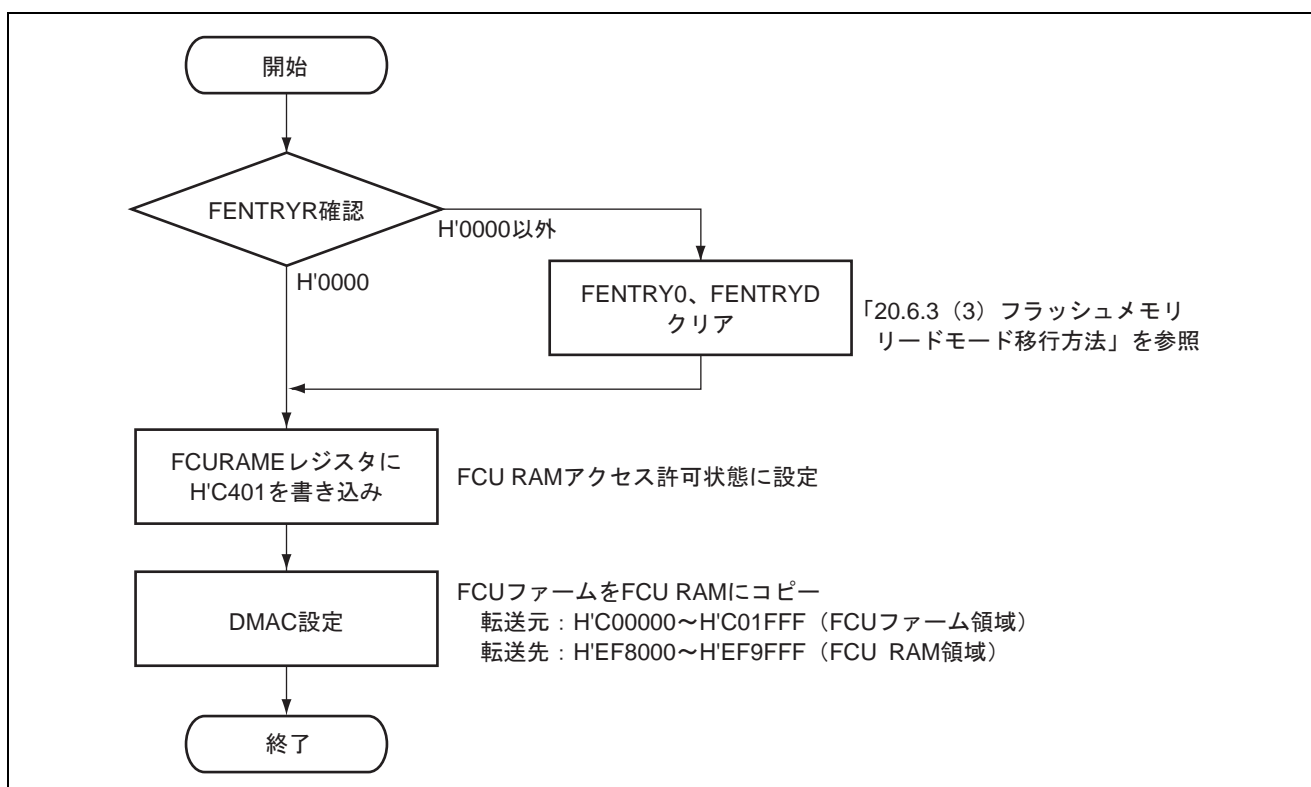


図 20.15 FCU RAM へのファームウェア転送フロー

(2) フラッシュメモリ P/E モード移行方法

フラッシュメモリ関連の FCU コマンドを実行するためには、FENTRYR レジスタの FENTRY0 ビットを設定して、FCU をフラッシュメモリ P/E モードに設定する必要があります（「20.6.2 FCU コマンド受け付け条件」を参照）。フラッシュメモリ（読み出しアドレス：H'000000～H'07FFFF、書き込み/消去用アドレス：H'D00000～H'D7FFFF）に対する FCU コマンドを使用したい場合には、FENTRY0 を 1 に設定します。FENTRY0 ビットの書き込み条件については、「20.3.9 フラッシュ P/E モードエントリレジスタ（FENTRYR）」を参照してください。

フラッシュメモリリードモードからフラッシュメモリ P/E モードに移行した場合には、フラッシュメモリ P/E ノーマルモードになります。

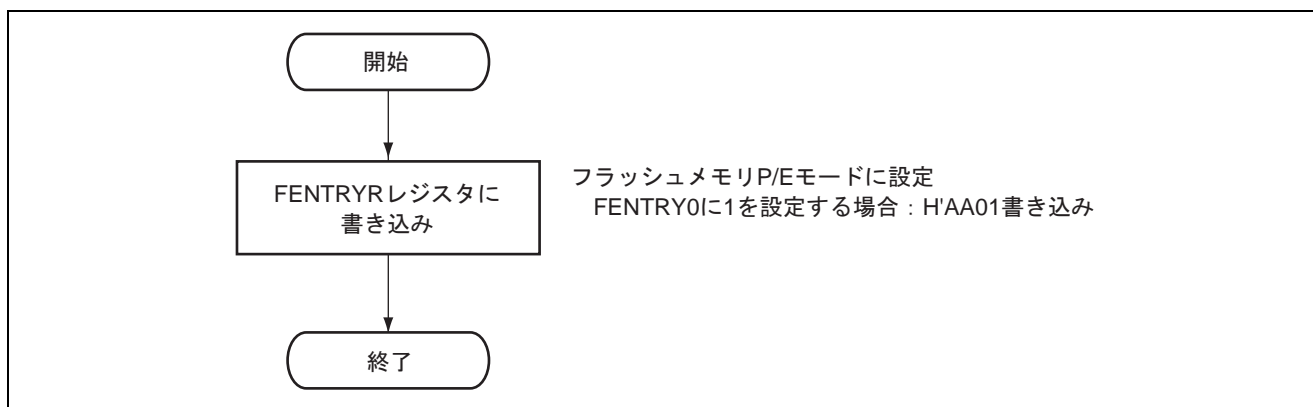


図 20.16 フラッシュメモリ P/E モード移行フロー

(3) フラッシュメモリリードモード移行方法

内部バス経由でフラッシュメモリを高速読み出しするためには、FENTRYR レジスタの FENTRY0 ビットをクリアして、FCU をフラッシュメモリリードモードに設定する必要があります（「20.6.2 FCU コマンド受け付け条件」を参照）。フラッシュメモリ P/E モードからフラッシュメモリリードモードへの移行は、FCU のコマンド処理が完了し、かつ FCU がエラー検出していない状態で実施してください。

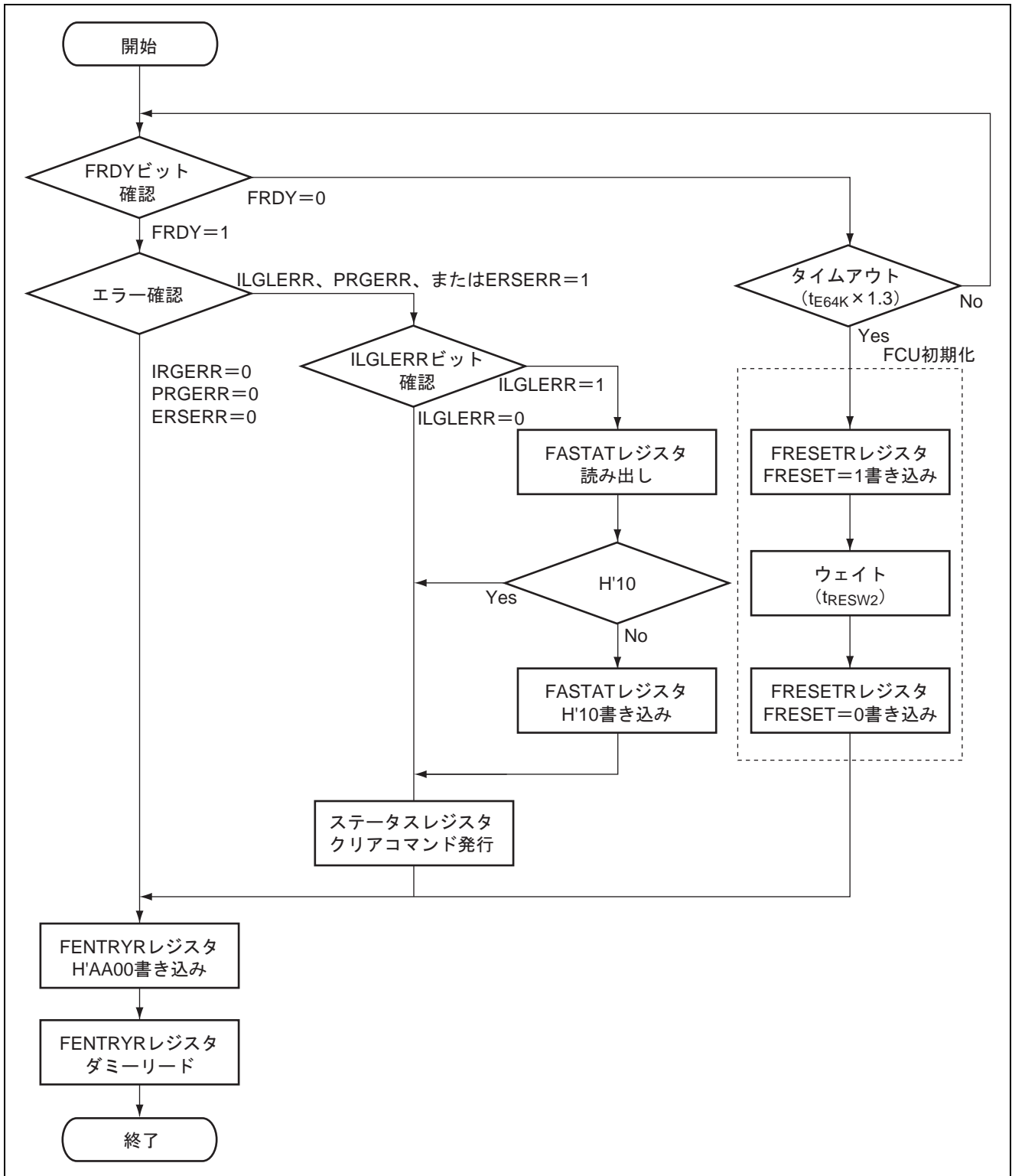


図 20.17 フラッシュメモリリードモード移行フロー

(4) フラッシュメモリ P/E ノーマルモード移行コマンド使用方法

フラッシュメモリ P/E ノーマルモードへの移行方法には、フラッシュメモリリードモード時に FENTRYR レジスタを設定する方法（「20.6.3 (1) FCU RAM へのファームウェア転送方法」を参照）とフラッシュメモリ P/E モード時にノーマルモード移行コマンドを発行する方法（図 20.18）があります。ステータスリードモード移行、ロックビットリードモード移行コマンドの使用方法もノーマルモード移行コマンドの使用方法と同様です。

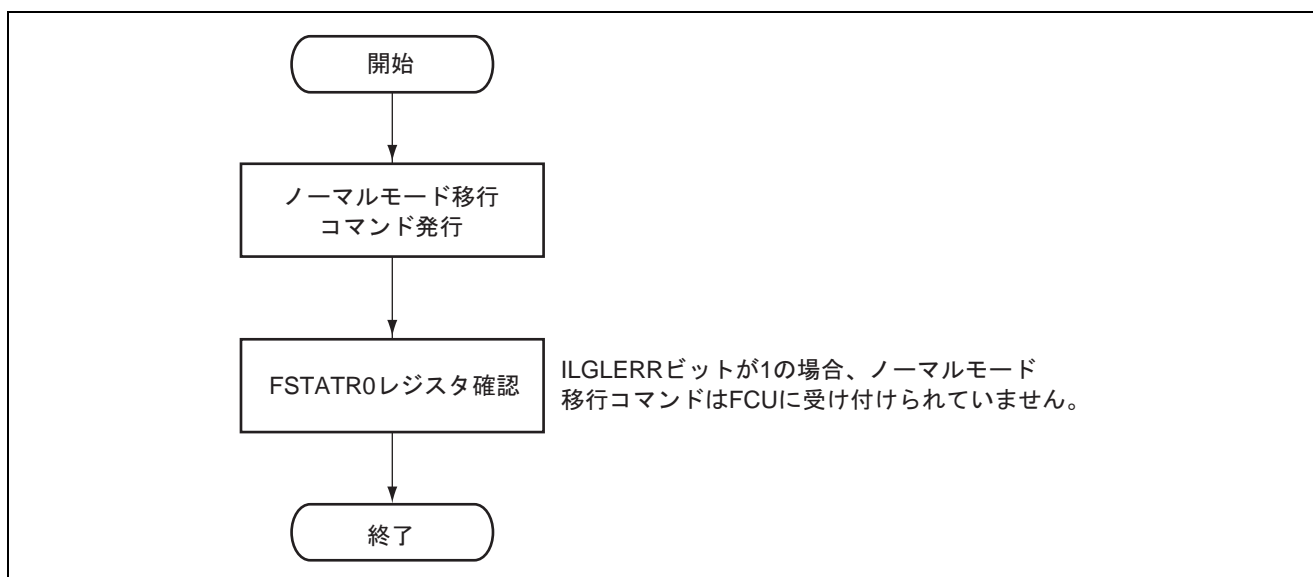


図 20.18 フラッシュメモリ P/E ノーマルモード移行コマンド使用方法

(5) フラッシュクロック通知コマンドの使用方法

フラッシュメモリへのデータ書き込み/消去前に使用しているフラッシュクロックの設定を行います。

最初に使用しているフラッシュクロックの周波数を FCKAR レジスタに設定します。設定可能な周波数の範囲は 8~40MHz です。この範囲以外は、設定しないでください。

次にフラッシュクロック通知コマンドを使用します。フラッシュクロック通知コマンドの第 1 サイクルでは HE9 を、第 2 サイクルでは H'03 を ROM 書き込み/消去用のアドレスにバイト書き込みします。コマンドの第 3 サイクル~第 5 サイクルでは、ワードサイズで周辺バスアクセスを実行します。この際、先頭アドレスは 4 バイト境界にアラインしたアドレスを使用してください。ROM 書き込み/消去用のアドレスに対して H'0F0F データの 3 回ワード書き込みを実行後、第 6 サイクルで ROM 書き込み/消去用のアドレスに対して H'D0 をバイト書き込みすると、FCU がフラッシュクロックの周波数設定処理を開始します。設定完了は、FSTATR0.FR DY ビットで確認可能です。

第 1 サイクル~第 6 サイクルで指定可能なアドレスは、FENTRY0 ビットを 1 に設定した場合にはアドレス H'D00000~H'D7FFFF を指定可能です。誤った FENTRY0 ビットとアドレス指定の組み合わせでコマンドを発行した場合には、FCU はエラーを検出しコマンドロック状態になります（「20.8.3 エラープロテクト」を参照）。

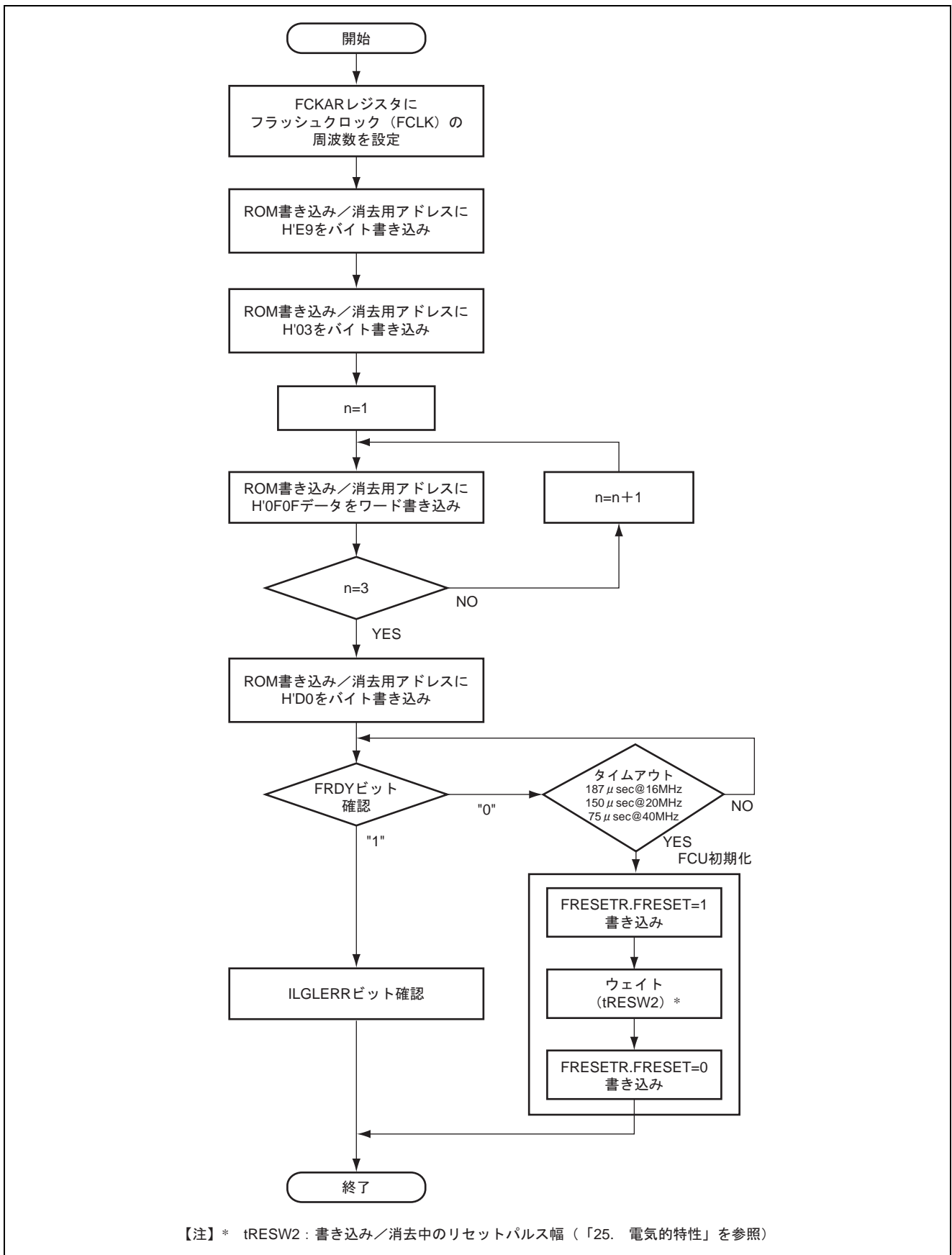


図 20.19 フラッシュロック通知コマンドの使用法

(6) 書き込み方法

フラッシュメモリへのデータ書き込みには、プログラムコマンドを使用します。プログラムコマンドの第1サイクルでは **H'E8** を、第2サイクルでは **H'40** を、フラッシュメモリ書き込み/消去用アドレスにバイト書き込みします。コマンドの第3~66サイクルでは、ワードサイズで周辺バスアクセスを実行します。第3サイクルのアクセスでは、フラッシュメモリ書き込み先頭アドレスに対して書き込みデータを書き込んでください。この際、先頭アドレスは128バイト境界にアラインしたアドレスを使用してください。フラッシュメモリ書き込み/消去用アドレスに対して63回のワード書き込みを実行後、第67サイクルでフラッシュメモリ書き込み/消去用アドレスに対して **H'D0** をバイト書き込みするとFCUがフラッシュメモリの書き込み処理を開始します。書き込みの完了は、**FSTATR0** レジスタの **FRDY** ビットで確認可能です。

第1~第67サイクルで指定可能なアドレスは、**FENTRYR** レジスタの **FENTRY0** を1に設定した場合にはアドレス **H'D00000**~**H'D7FFFF** を指定可能です。誤ったアドレス指定でコマンドを発行した場合には、FCUはエラーを検出しコマンドロック状態になります（「20.8.3 エラープロテクト」を参照）。

第3サイクル~第66サイクルでアクセスする領域に書き込み不要なアドレスが含まれる場合は、該当アドレスに対する書き込みデータを **H'FFFF** にしてください。ロックビットによるプロテクトを無効化して書き込みを実施したい場合には、**FPROTR** レジスタの **FPROTCN** ビットを1にセットしてから書き込みを行ってください。

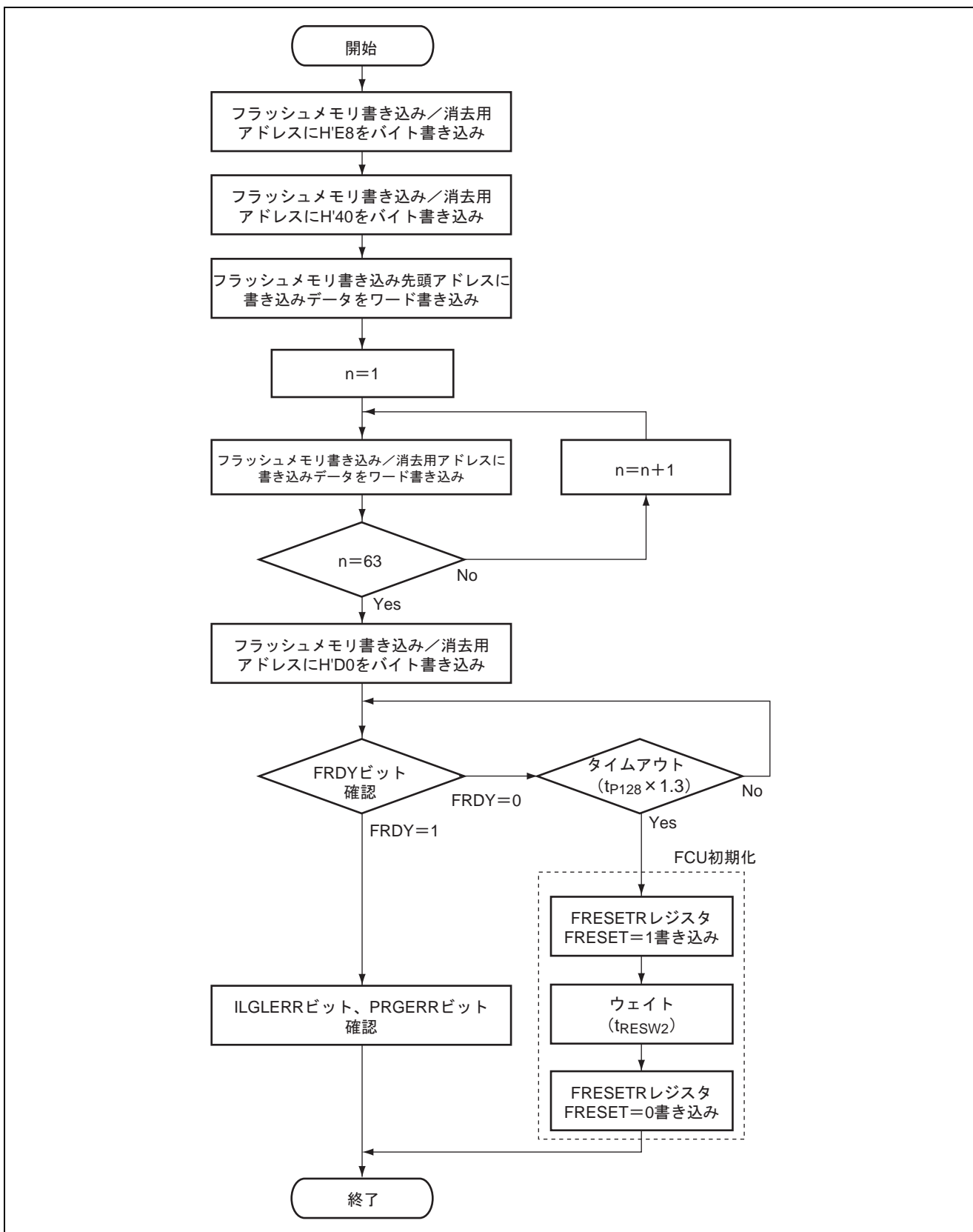


図 20.20 フラッシュメモリ書き込み方法

(7) 消去方法

フラッシュメモリの消去には、ブロックイレーズコマンドを使用します。ブロックイレーズコマンドの第1サイクルではH'20をフラッシュメモリ書き込み/消去用アドレスにバイト書き込みします。第2サイクルでH'D0を消去対象ブロック内の任意アドレスにバイト書き込みするとFCUがフラッシュメモリの消去処理を開始します。消去の完了は、FSTATR0レジスタのFRDYビットで確認可能です。

ロックビットによるプロテクトを無効化して消去を実施したい場合には、FPROTRレジスタのFPROTCNビットを1にセットしてから消去を行ってください。

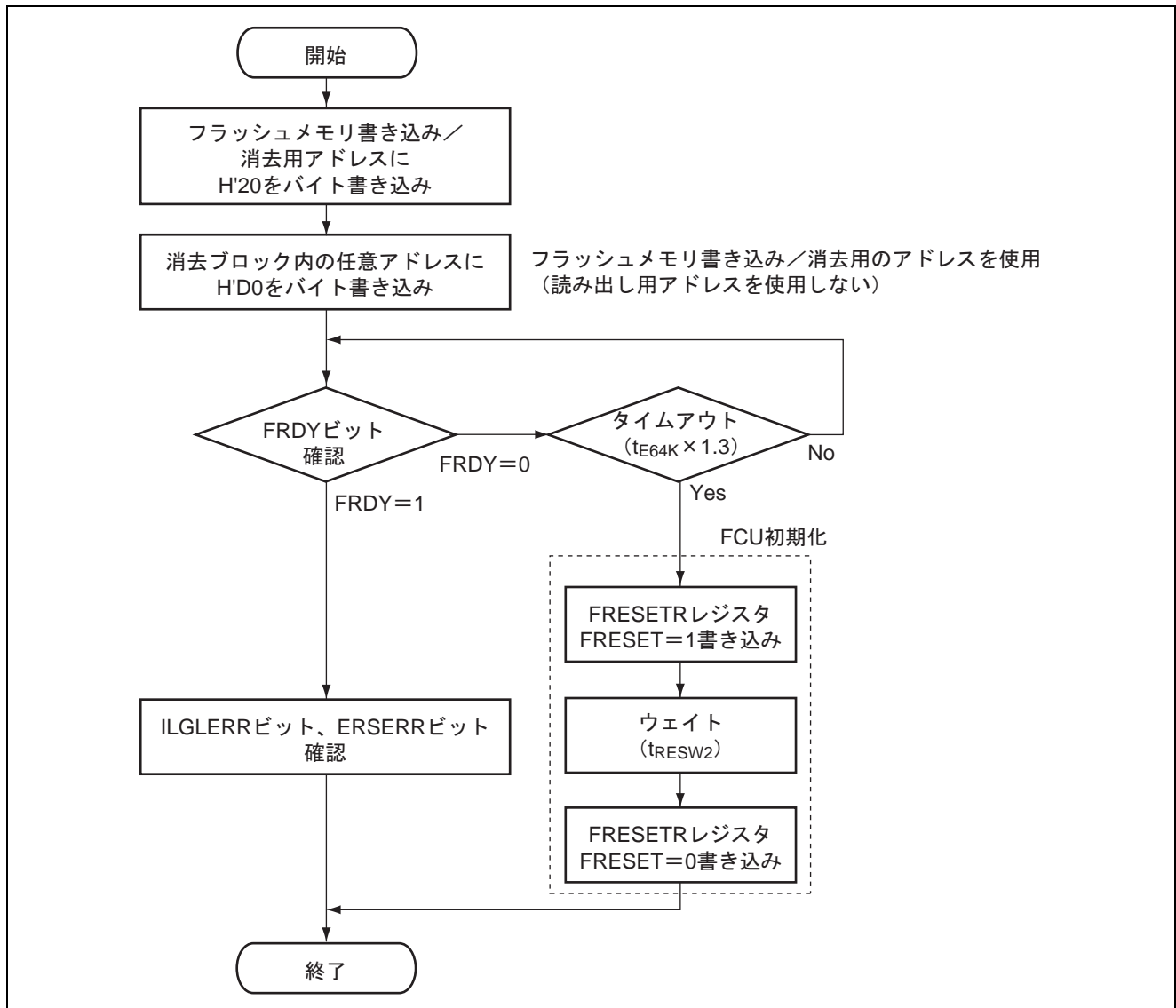


図 20.21 フラッシュメモリ消去方法

(8) 書き込み／消去のサスペンド方法

フラッシュメモリの書き込み／消去の中断には、P/E サスペンドコマンドを使用します。P/E サスペンドコマンドを発行する場合には、事前に FSTATR0 レジスタの ILGLERR ビット、PRGERR ビット、ERSERR ビットと FSTATR1 レジスタの FCUERR ビット、FRDTCT ビット、FRCRCT ビットが 0 で書き込み／消去処理が正常に実行されていることを確認してください。また、サスペンドコマンドが受け付け可能であることを確認するために、FSTATR1 レジスタの SUSRDY ビットが 1 であることを確認してください。P/E サスペンドコマンドの発行後は、FSTATR0 レジスタと FSTATR1 レジスタを読み出して、エラーが発生していないことを確認してください。書き込み／消去処理中に異状が発生した場合には、ILGLERR ビット、PRGERR ビット、ERSERR ビット、FCUERR ビット、FRDTCT ビット、FRCRCT ビットのうち少なくとも 1 つのビットが 1 になります。また、SUSRDY ビットが 1 であることを確認してから P/E サスペンドコマンドが受け付けられるまでの間に書き込み／消去処理が完了していた場合には、P/E サスペンドコマンドが不正コマンドとして検出されるため ILGLERR ビットが 1 になります。P/E サスペンドコマンドの受付と書き込み／消去処理の完了が同時であった場合には、エラーは発生せず、サスペンド状態にも遷移しません (FRDY ビットが 1 かつ ERSSPD ビットと PRGSPD ビットが 0)。P/E サスペンドコマンドが受け付けられて、書き込み／消去の中断処理が正常終了した場合には、FCU がサスペンド状態に遷移して FDRY ビットが 1 かつ ERSSPD ビットまたは PRGSPD ビットが 1 になります。P/E サスペンドコマンド発行後は、サスペンド状態に遷移していることを確認した後、後続するフローを決定してください。サスペンド状態に遷移していないにもかかわらず、後続するフローで P/E レジュームコマンドを発行すると、不正コマンドエラーが発生し FCU がコマンドロック状態に遷移します (「20.8.3 エラープロテクト」を参照)。

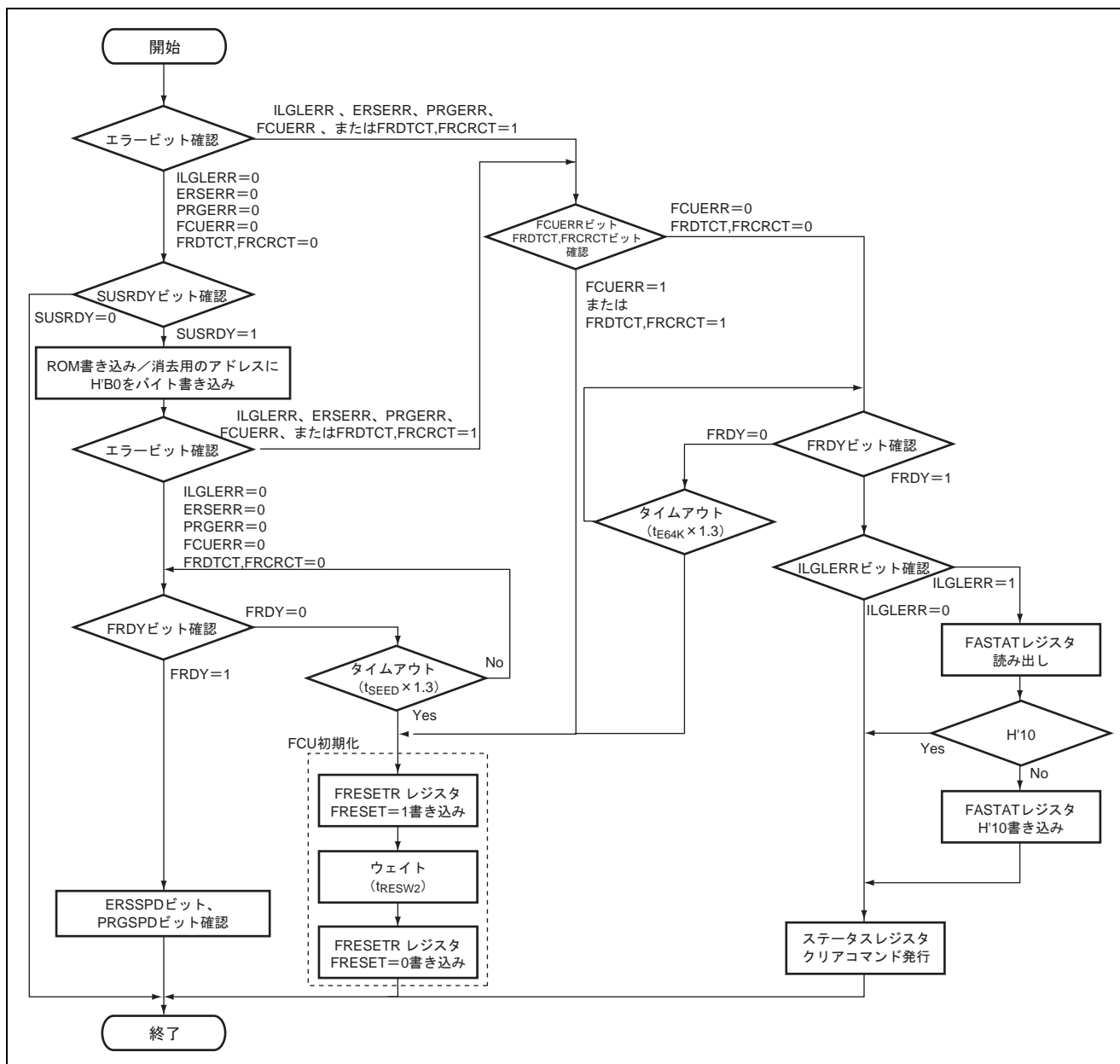


図 20.22 書き込み/消去のサスペンド方法

消去サスペンド状態に遷移した場合には、消去対象外のブロックに対する書き込みを実行することができます。また、書き込み/消去サスペンド状態ともに、FENTRYR レジスタをクリアすることにより、フラッシュメモリリードモードに遷移することも可能です。

なお、P/E サスペンドコマンド受け付け時の FCU 動作の内容については、「20.6.4 サスペンド動作」を参照してください。

(9) P/E レジューム

サスペンドした書き込み/消去処理を再開したい場合には、P/E レジュームコマンドを使用します。サスペンド中に FENTRYR レジスタの設定を変更した場合には、P/E レジュームコマンドを発行する前に FENTRYR を P/E サスペンドコマンド発行直前の値に再設定してください。

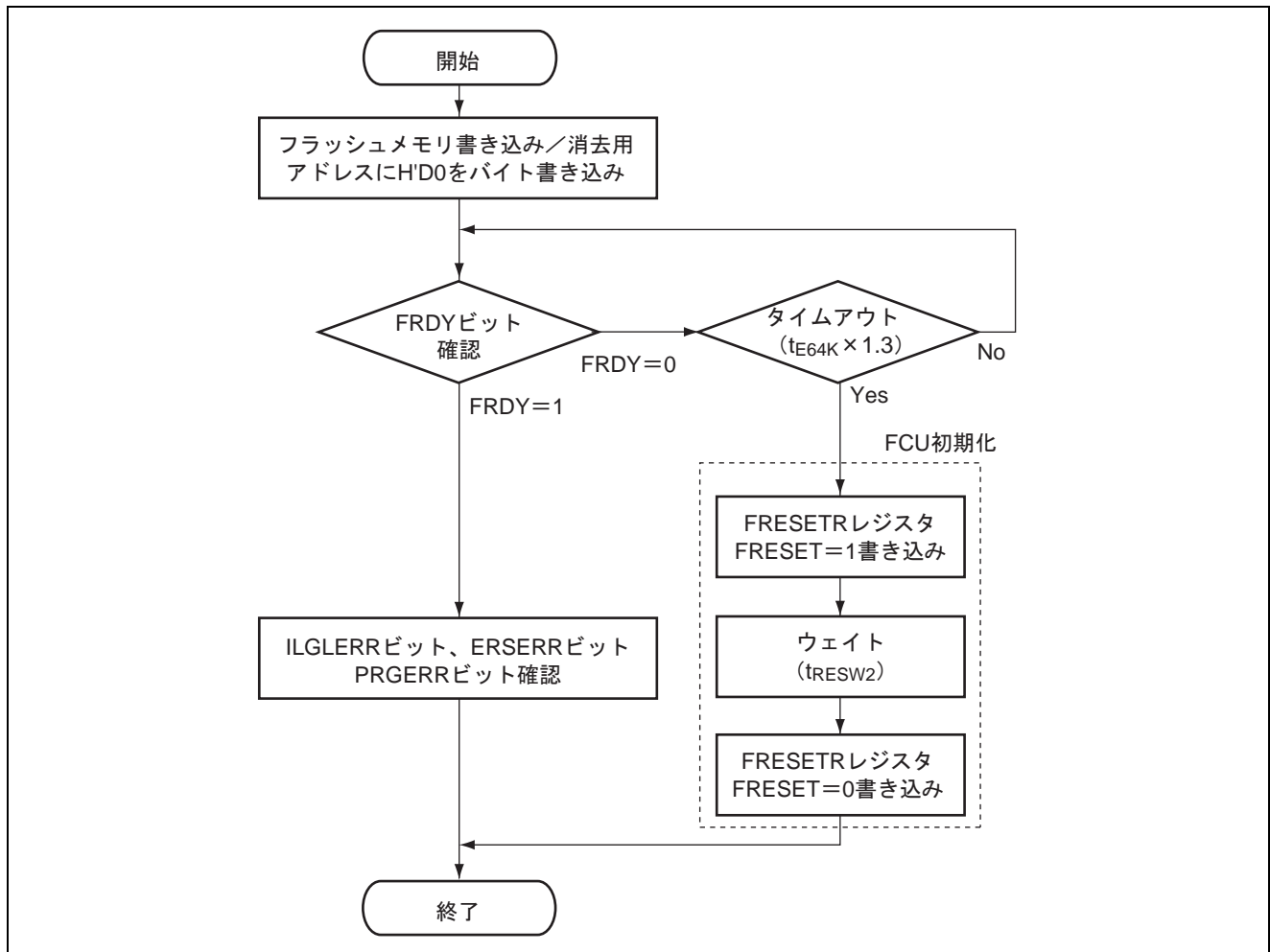


図 20.23 書き込み/消去のレジューム方法

(10) ステータスレジスタ 0 (FSTATR0) のクリア方法

FSTATR0 レジスタの ILGLERR ビット、PRGERR ビット、ERSERR ビットをクリアしたい場合には、ステータスレジスタクリアコマンドを使用します。ILGLERR ビット、PRGER ビット、ERSERR ビットのいずれかが 1 である場合、FCU はコマンドロック状態になりステータスレジスタクリアコマンド以外の FCU コマンドを受け付けません。ILGLERR ビットが 1 の場合には、FASTAT レジスタの ROMAE ビット、EEPAE ビット、EEPRPE ビット、および EEPWPE ビットの値も確認してください。これらのビットをクリアせずにステータスレジスタクリアコマンドを発行しても、ILGLERR ビットはクリアされません。

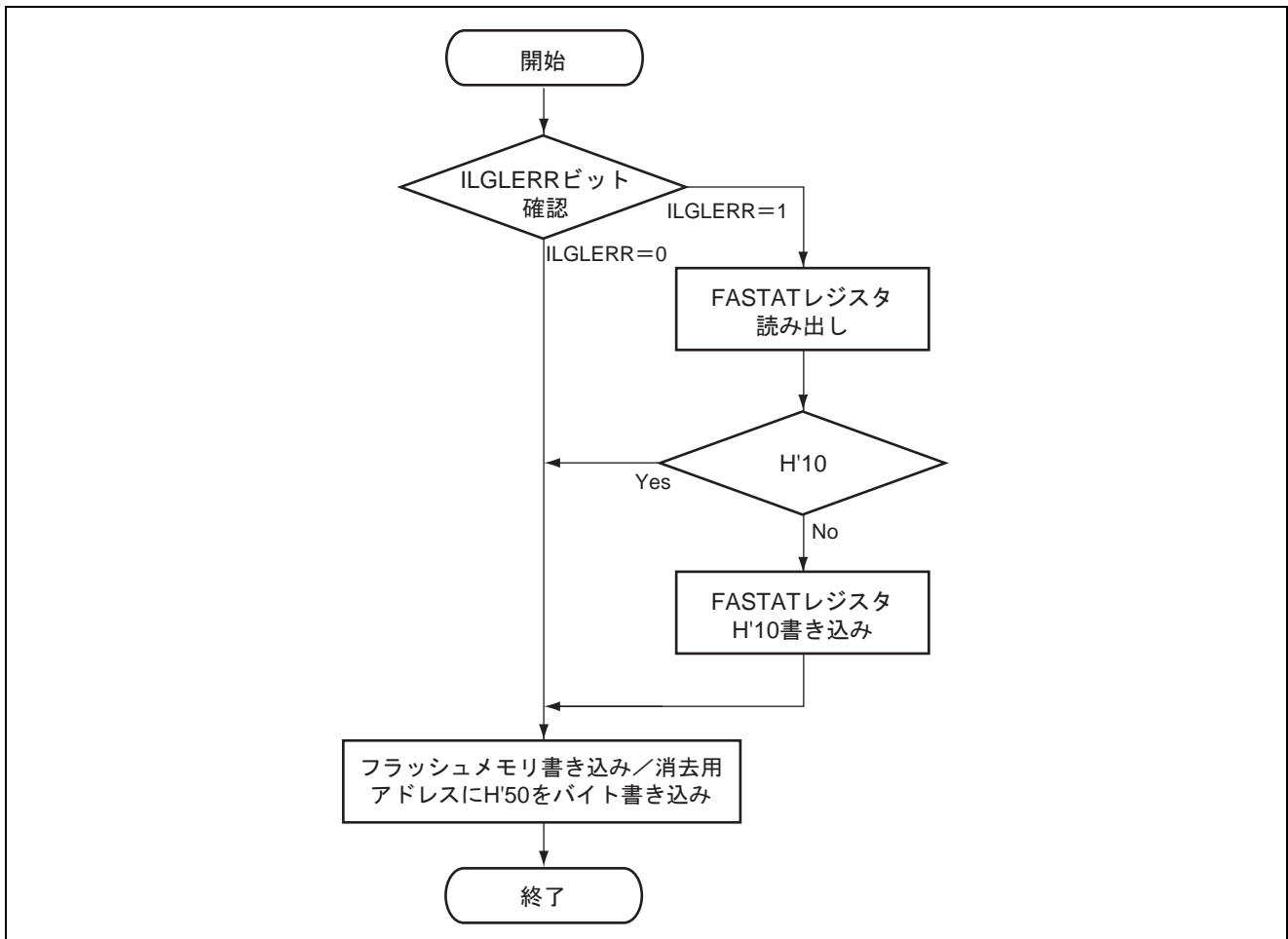


図 20.24 ステータスレジスタ 0 のクリア方法

(11) ステータスレジスタ 0 (FSTATR0) の確認方法

FSTATR0 レジスタの確認方法には、FSTATR0 を直接読み出す方法とフラッシュメモリステータスリードモードでフラッシュメモリ書き込み/消去用アドレスを読み出す方法があります。ノーマルモード移行、ロックビットリードモード移行以外の FCU コマンド発行後には、FCU はフラッシュメモリステータスリードモードに遷移しています。図 20.25 の例では、ステータスリードモード移行コマンドを発行してフラッシュメモリステータスリードモードに移行し、フラッシュメモリ書き込み/消去用アドレスを読み出して FSTATR0 の内容を確認しています。

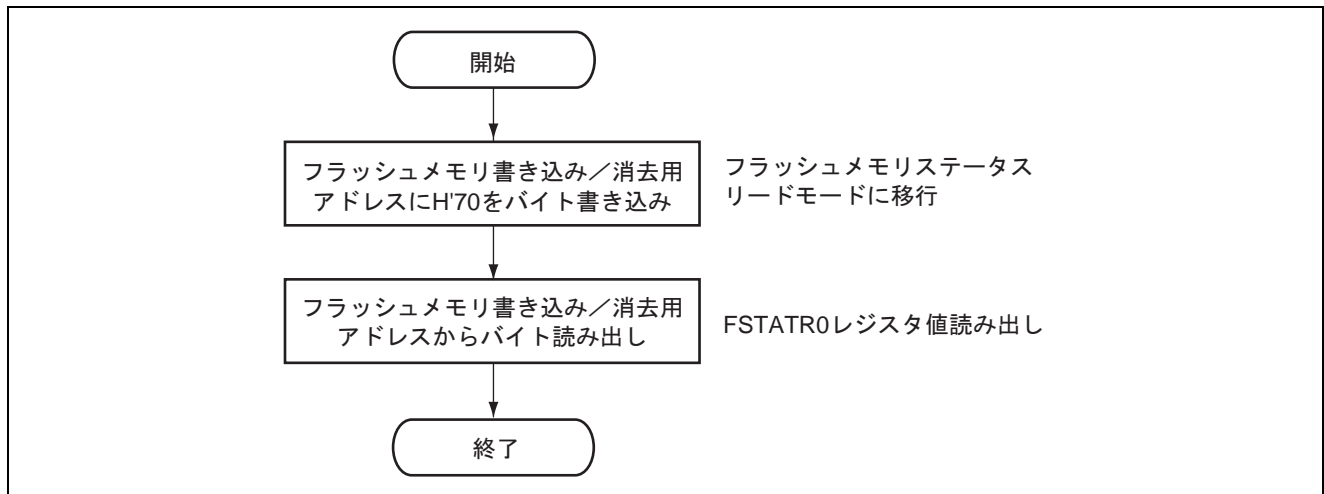


図 20.25 ステータスレジスタ 0 の確認方法

(12) ロックビットの読み出し

ユーザマットの各消去ブロックにはロックビットが内蔵されています。FPROTR レジスタの FPROTCN ビットが 0 の状態では、ロックビットが 0 に設定された消去ブロックを書き込み／消去することができません。

ロックビットの確認方法には、メモリ領域リードモードとレジスタリードモードがあります。メモリ領域リードモード（FMODR レジスタの FRDMD ビットが 0）の場合には、フラッシュメモリロックビットリードモードでフラッシュメモリ書き込み／消去用のアドレスに対する読み出しを実行すると、指定した消去ブロックのロックビットが周辺バス読み出しデータの全ビットにコピーされます。レジスタリードモード（FMODR の FRDMD ビットが 1）の場合には、ロックビットリード 2 コマンドを発行すると、指定した消去ブロックのロックビットが FSTATR1 レジスタの FLOCKST ビットにコピーされます。

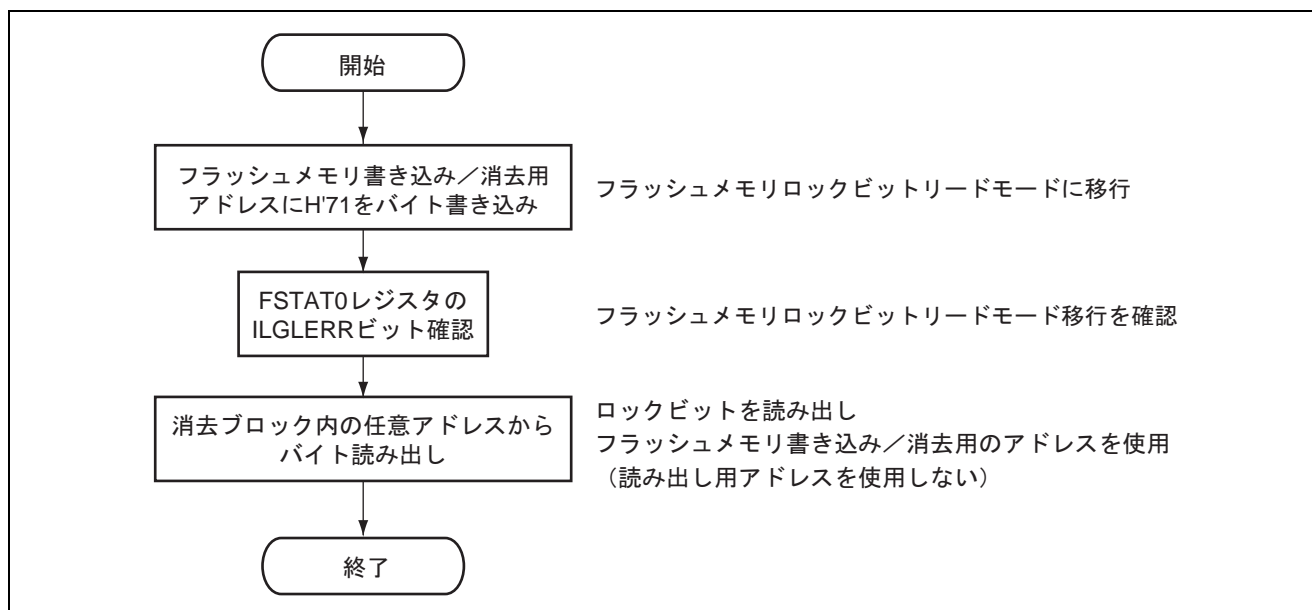


図 20.26 メモリ領域リードモードでロックビットを読み出す方法

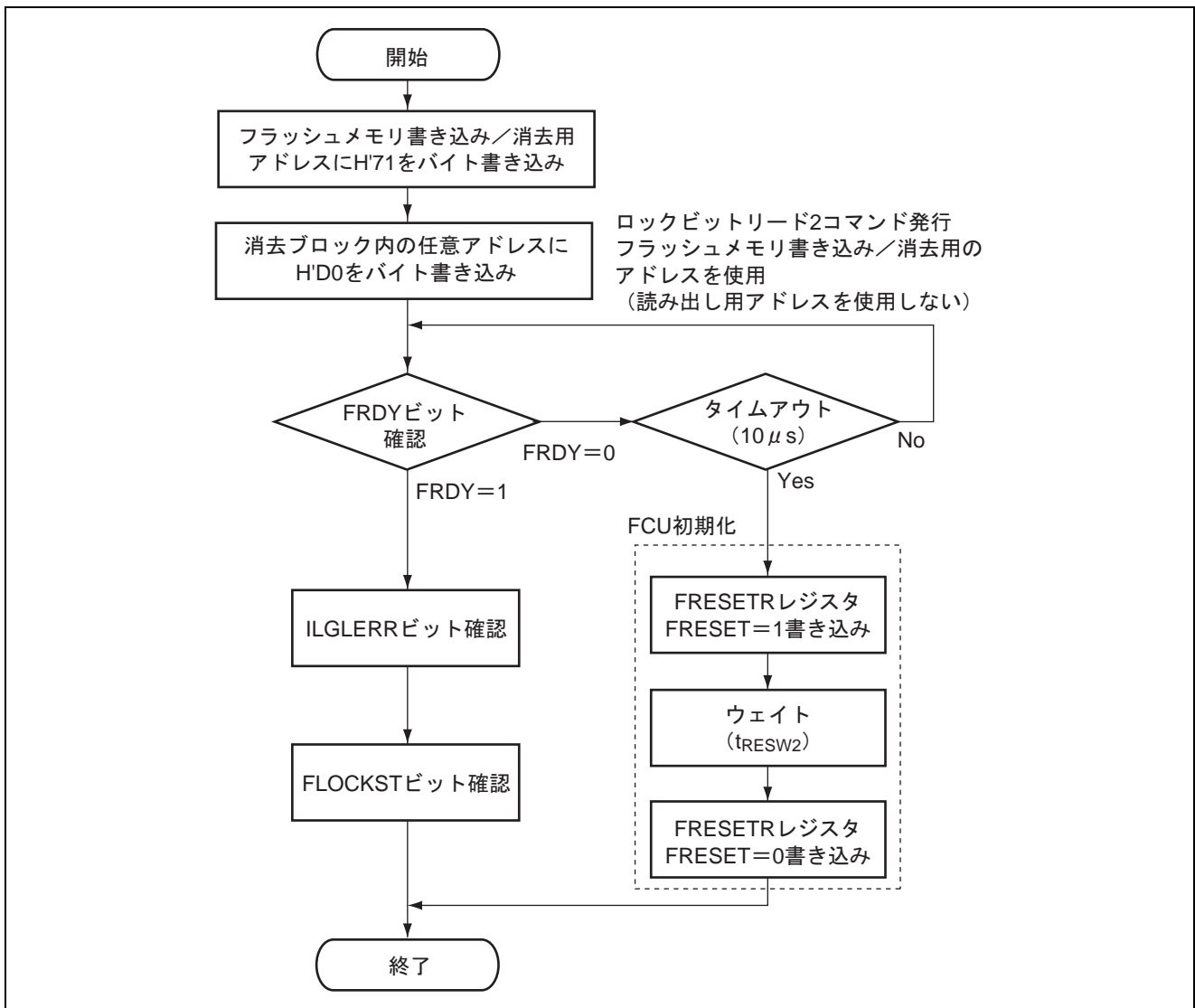


図 20.27 レジスタリードモードでロックビットを読み出す方法

(13) ロックビットの書き込み

ユーザマットの各消去ブロックにはロックビットが内蔵されています。ロックビットに書き込みを行いたい場合にはロックビットプログラムコマンドを使用します。ロックビットプログラムコマンドの第1サイクルではH'77をフラッシュメモリ書き込み/消去用アドレスにバイト書き込みします。第2サイクルでロックビットを書き込みたい消去ブロック内の任意アドレスに対してH'D0をバイト書き込みするとFCUがロックビットの書き込み処理を開始します。書き込みの完了は、FSTATR0レジスタのFRDYビットで確認可能です。

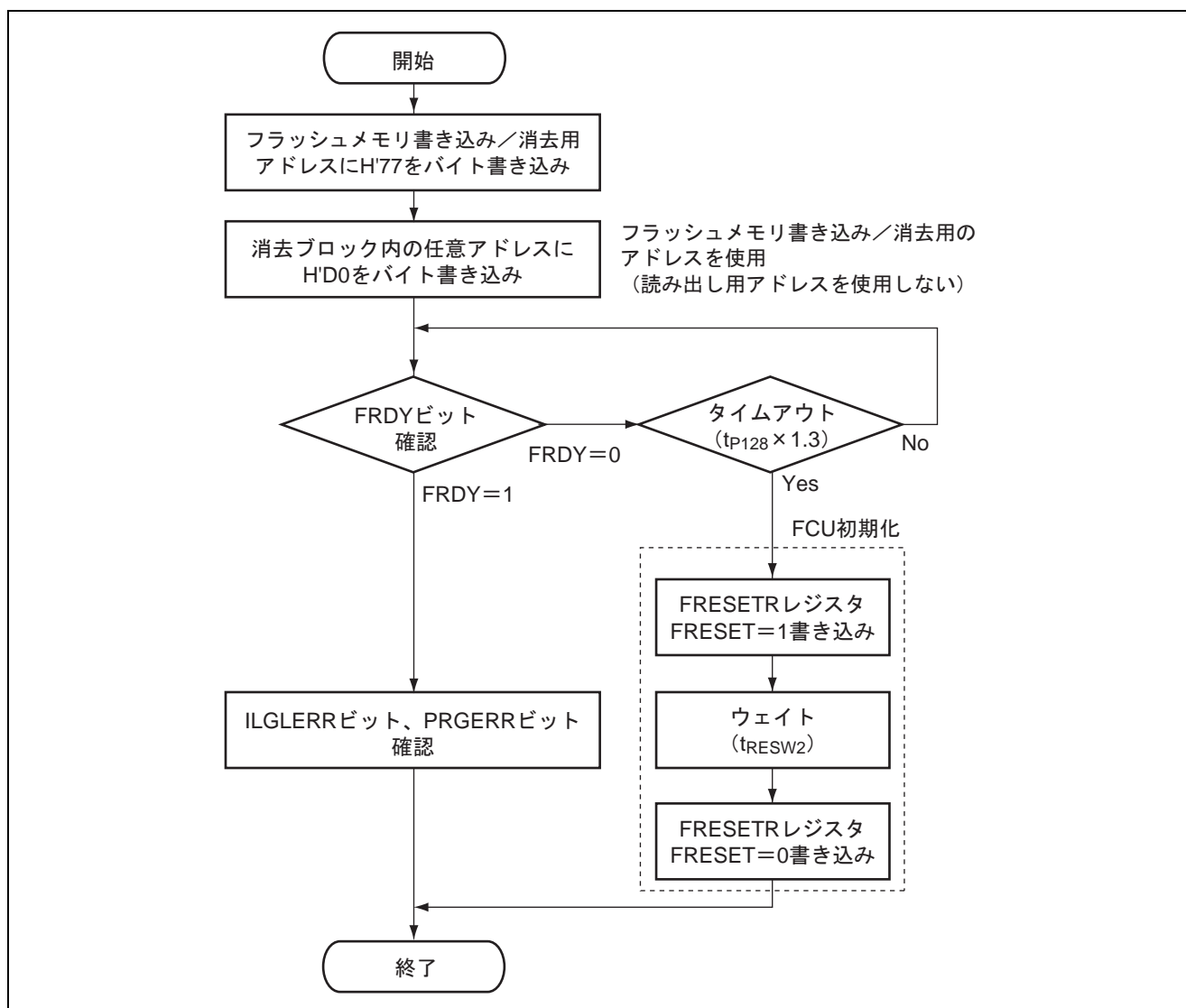


図 20.28 ロックビットのプログラム方法

ロックビットの消去には、ブロックイレーズコマンドを使用します。FPROTRレジスタのFPROTCNビットが0の状態ではロックビットが0に設定された消去ブロックを消去することができません。ロックビットを消去する場合には、FPROTCNビットを1にセットした状態でブロックイレーズコマンドを発行してください。ブロックイレーズコマンドを使用すると消去ブロック内の全データが消去されます。ロックビットのみを消去することはできません。

20.6.4 サスペンド動作

フラッシュメモリの書き込み/消去中に P/E サスペンドコマンドを発行すると、FCU は書き込み/消去処理を中断します。図 20.29 に書き込み処理の中断動作の概略を示します。FCU は書き込み系のコマンドを受け付けると FSTATR0 レジスタの FRDY ビットを 0 にクリアして書き込み処理を開始します。書き込み処理の開始後に FCU が P/E サスペンドコマンドを受け付け可能な状態に遷移すると SUSRDY ビットが 1 にセットされます。P/E サスペンドコマンドが発行されると、FCU はサスペンドコマンドを受け付けて SUSRDY ビットを 0 にクリアします。フラッシュメモリへの書き込みパルス印加中に、FCU が P/E サスペンドを受け付けた場合には、FCU はパルスの印加を継続します。所定のパルス印加時間を経過すると、FCU はパルス印加を完了し、書き込みの中断処理を開始して PRGSPD ビットを 1 にセットします。中断処理が完了すると FCU は FRDY ビットを 1 にセットして書き込みサスペンド状態に遷移します。書き込みサスペンド状態で FCU が P/E レジュームコマンドを受け付けた場合には、FCU は FRDY ビットと PRGSPD ビット 0 にクリアして書き込み処理を再開します。

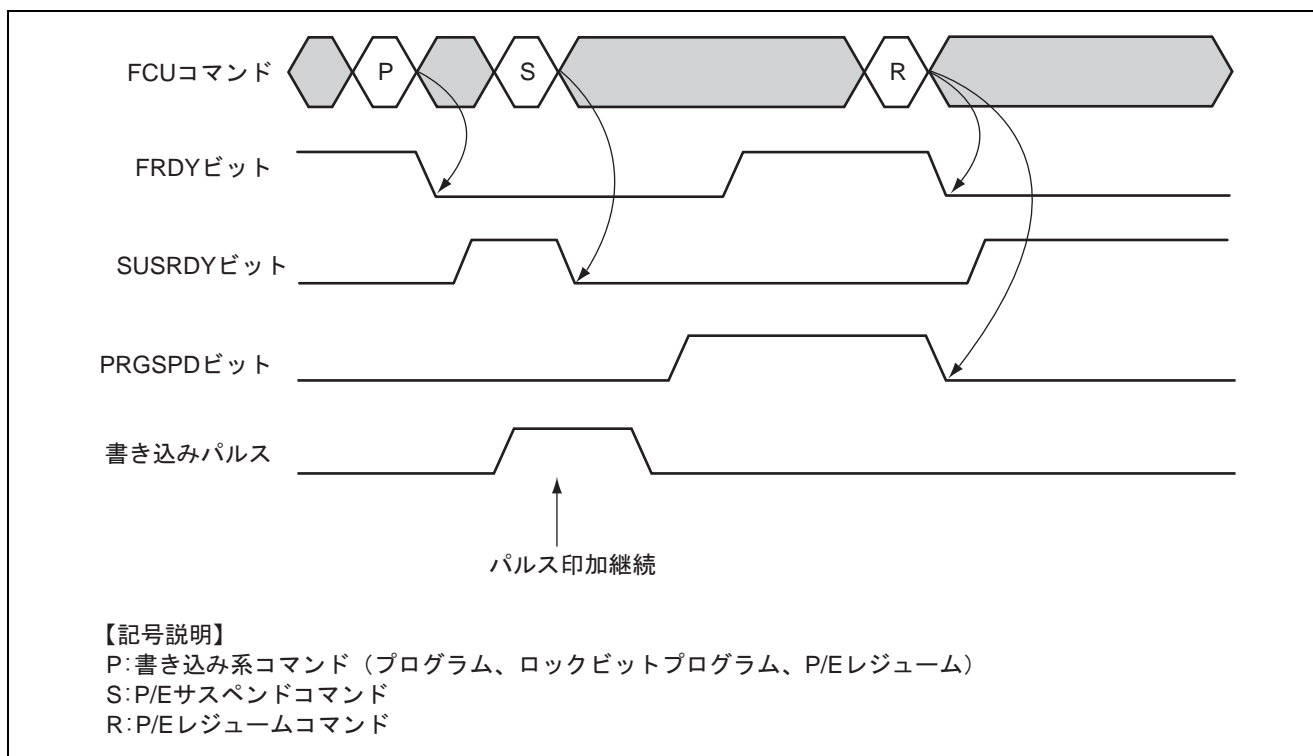


図 20.29 書き込み処理の中断動作

図 20.30 に消去サスペンドモードがサスペンド優先モード (FCPSR レジスタの ESUSPMD ビットが 0) の場合の消去処理の中断動作を示します。FCU は消去系のコマンドを受け付けると FRDY ビットを 0 にクリアして消去処理を開始します。消去処理の開始後に FCU が P/E サスペンドコマンドを受け付け可能な状態に遷移すると、SUSRDY ビットが 1 にセットされます。P/E サスペンドコマンドが発行されると、FCU はサスペンドコマンドを受け付けて SUSRDY ビットをクリアします。消去処理中にサスペンドコマンドを受け付けた場合には、FCU は消去パルス印加中でも中断処理を開始して ERSSPD ビットを 1 にセットします。中断処理が完了すると、FCU は FRDY ビットを 1 にセットして、消去サスペンド状態に遷移します。消去サスペンド状態で、FCU が P/E レジュームコマンドを受け付けた場合には、FCU は FRDY ビットと ERSSPD ビットを 0 にクリアして、消去処理を再開します。消去処理の中断/再開時の FRDY ビット、SUSRDY ビット、ERSSPD ビット動作は、消去サスペンドモードに依存せず同様です。

消去サスペンドモードの設定は、消去パルスの制御方式に影響を与えます。サスペンド優先モードでは、過去に中断されたことのない消去パルス A を印加中に FCU が P/E サスペンドコマンドを受け付けた場合には、消去パルス A の印加を中断して消去サスペンド状態に遷移します。P/E レジュームコマンドにより消去が再開され、消去パルス A を再印加している期間に、FCU が P/E サスペンドコマンドを受け付けた場合には、FCU は消去パルス A の印加を継続します。所定のパルス印加時間を経過すると、FCU は消去パルスの印加を完了して消去サスペンド状態に遷移します。次に FCU が P/E レジュームコマンドを受け付けて、新たな消去パルス B の印加が開始された後に、再び FCU が P/E サスペンドコマンドを受け付けた場合には、消去パルス B の印加は中断されます。サスペンド優先モードでは、1 パルスあたり 1 回の割合で消去パルスの印加を中断してサスペンド処理を優先するため、サスペンドの遅延を小さくできる場合があります。

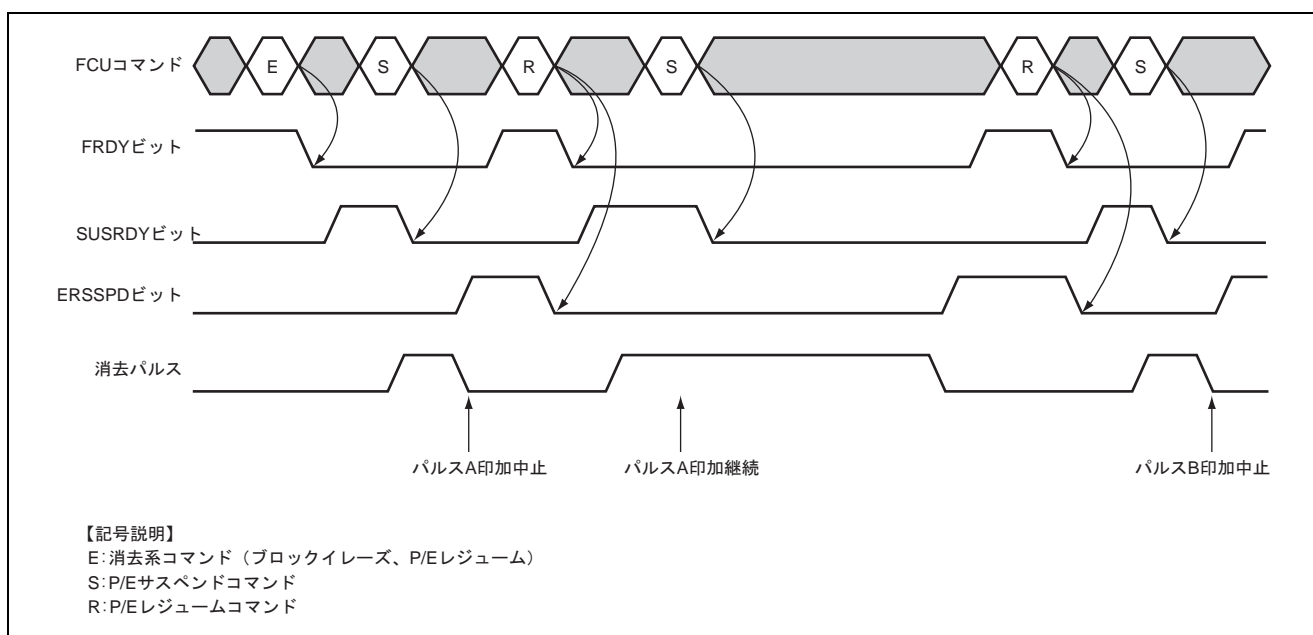


図 20.30 消去処理の中断動作 (サスペンド優先モード)

図 20.31 に消去優先モード (FCPSR レジスタの ESUSPMD ビットが 1) の場合の消去処理の中断動作を示します。消去優先モードの消去パルス制御方式は、書き込み中断処理の書き込みパルス制御方式と同様です。

FCU が消去パルス印加中に P/E サスペンドコマンドを受け付けた場合には、必ず消去パルスの印加を継続します。このモードでは消去パルスの再印加が発生しないため、サスペンド優先モードと比較して消去処理全体に必要な時間を短縮可能です。

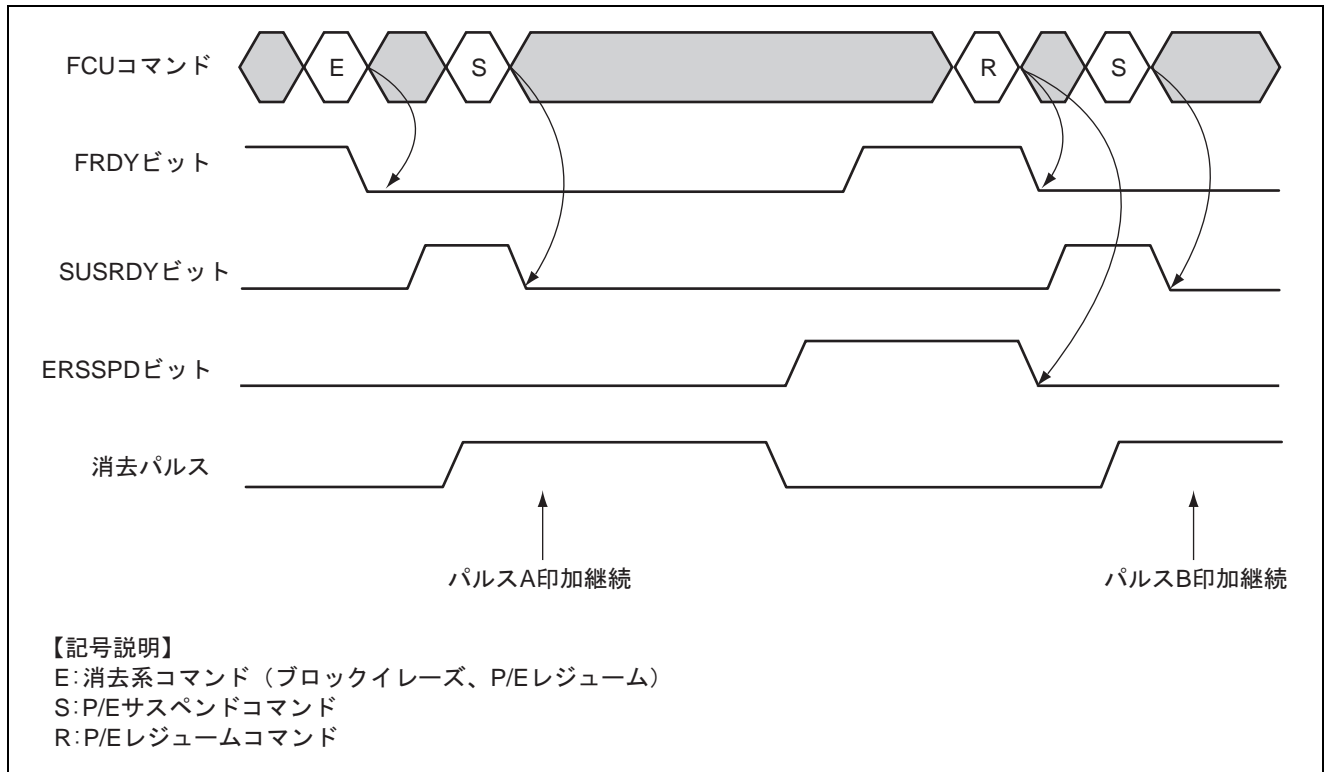


図 20.31 消去処理の中断動作 (消去優先モード)

20.7 ライタモード

ライタモードでは表 20.13 のデバイスタイプをサポートしている汎用 PROM ライタを用いて内蔵 ROM にプログラムを書き込むことができます。

表 20.13 ライタモードでサポートするデバイスタイプ

対象マット	容量	デバイスタイプ
ユーザマット	512K バイト	FZTAT512V5A
ユーザブートマット*	16K バイト	FZTAT USBT16V5A

【注】 * 本 LSI にはユーザブートマットは存在しません。

20.8 プロテクト

フラッシュメモリに対する書き込み/消去のプロテクトには、ハードウェアプロテクト、ソフトウェアプロテクト、エラープロテクトの 3 種類があります。

20.8.1 ハードウェアプロテクト

ハードウェアプロテクトは、チップ端子の設定によってフラッシュメモリに対する書き込み/消去が禁止された状態です。

(1) モード端子によるプロテクト

内蔵 ROM が無効な設定の場合には、フラッシュメモリの書き込み/消去/読み出しが禁止された状態になります（本 LSI には内蔵 ROM 無効モードはありません）。本 LSI の動作モードと端子設定の関係は「第 3 章 MCU 動作モード」を参照してください。

20.8.2 ソフトウェアプロテクト

ソフトウェアプロテクトは、制御レジスタ設定やユーザマットのロックビット設定によってフラッシュメモリに対する書き込み/消去が禁止された状態です。ソフトウェアプロテクトに違反して、フラッシュメモリに対する書き込み/消去系コマンドを発行した場合には、FCU がエラーを検出してコマンドロック状態になります。

(1) FENTRYR によるプロテクト

FENTRYR レジスタの FENTRY0 が 0 の場合には、フラッシュメモリ（読み出し用アドレス：H'000000～H'07FFFF、書き込み/消去用アドレス：H'D00000～H'D7FFFF）はフラッシュメモリリードモードになります。フラッシュメモリリードモードでは FCU コマンドが受け付けられないため、フラッシュメモリの書き込み/消去は禁止状態になります。フラッシュメモリリードモードで FCU コマンドを発行すると、FCU はイリーガルコマンドエラーを検出してコマンドロック状態になります（「20.8.3 エラープロテクト」を参照）。

(2) ロックビットによるプロテクト

ユーザマットの各消去ブロックにはロックビットが内蔵されています。FPROTR レジスタの FPROTCN ビットが 0 の場合には、ロックビットが 0 に設定された消去ブロックに対する書き込み/消去は禁止状態になります。ロックビットが 0 に設定された消去ブロックを書き込み/消去したい場合には、FPROTCN ビットを 1 に設定してください。ロックビットによるプロテクトに違反して、フラッシュメモリに対する書き込み/消去系コマンドを発行すると、FCU は書き込み/消去エラーを検出してコマンドロック状態になります(「20.8.3 エラープロテクト」を参照)。

20.8.3 エラープロテクト

エラープロテクトは、FCU コマンドの誤発行/禁止アクセスの発生/FCU の誤動作を検出して FCU コマンドの受け付けを禁止する状態(コマンドロック状態)です。FCU をコマンドロック状態にすることにより、フラッシュメモリの書き込み/消去が禁止されます。コマンドロック状態を解除するためには、FASTAT レジスタが H'10 の状態でステータスレジスタクリアコマンドを発行する必要があります。

FAEINT レジスタの CMDLKIE ビットが 1 の場合には、FCU がコマンドロック状態 (FASTAT レジスタの CMDLK ビットが 1) になるとフラッシュインタフェースエラー (FIFE) 割り込みが発生します。また、FAEINT レジスタの ROMAEIE が 1 の場合には、FASTAT レジスタの ROMAE ビットが 1 になると FIFE 割り込みが発生します。

表 20.14 にフラッシュメモリ専用およびフラッシュメモリと EEPROM 共通のエラープロテクト内容とエラー検出後のステータスビット値 (FSTATR0 レジスタの ILGLERR ビット、ERSERR ビット、PRGERR ビット、FSTATR1 レジスタの FCUERR ビット、FRDTCT ビット、FRCRCT ビット、FASTST レジスタの ROMAE ビット) の関係を示します。書き込み/消去処理中にサスペンド以外のコマンドが発行されてコマンドロック状態に遷移した場合には、FCU は書き込み/消去処理を継続します。この状態で P/E サスペンドコマンドを発行して書き込み/消去を中断することはできません。コマンドロック状態でコマンドが発行された場合には、ILGLERR ビットの値は 1 になり、その他のビットの値は以前のエラー検出時に設定された値を保持します。

表 20.14 エラープロテクト一覧

分類	内 容	ILGLERR	ERSERR	PRGERR	FCUERR	FRDTCT	FRCRCT	ROMAE
FENTRYR 設定エラ ー	FENTRYR レジスタに H'0001、H'0080 以外の値を設定	1	0	0	0	0	0	0
	サスペンド時とレジャーモード時で FENTRYR レジスタ設定が 不一致	1	0	0	0	0	0	0
不正 コマンド エラー	FCU コマンドの 1 サイクル目で未定義コードを指定	1	0	0	0	0	0	0
	複数サイクルの FCU コマンドの最終サイクルで H'D0 以外を指定	1	0	0	0	0	0	0
	書き込み/消去処理中にサスペンド以外のコマンドを発行	1	0	0	0	0	0	0
	書き込み/消去以外の処理中にサスペンドコマンドを発行	1	0	0	0	0	0	0
	サスペンド状態でサスペンドコマンドを発行	1	0	0	0	0	0	0
	サスペンド以外の状態でレジャーモードコマンドを発行	1	0	0	0	0	0	0
	書き込みサスペンド状態で書き込み/消去系（プログラム/ ロックビットプログラム/ブロックイレーズ（「20.6 ユーザ プログラムモード」を参照））コマンドを発行	1	0	0	0	0	0	0
	消去サスペンド状態でブロックイレーズコマンド発行	1	0	0	0	0	0	0
	消去サスペンド状態で消去サスペンド対象領域へのプログラム/ ロックビットプログラムを発行	1	0	0	0	0	0	0
	プログラムコマンドの 2 サイクル目で H'40 以外を指定	1	0	0	0	0	0	0
コマンドロック状態でコマンド発行	1	0/1	0/1	0/1	0/1	0/1	0/1	
消去 エラー	消去処理中のエラー発生	0	1	0	0	0	0	0
	FPROTR レジスタの FPROTCN ビットが 0 の場合に、ロックビットが 0 に設定された消去ブロックにブロックイレーズコマンドを発行	0	1	0	0	0	0	0
書き込み エラー	書き込み処理中のエラー発生	0	0	1	0	0	0	0
	FPROTR レジスタの FPROTCN ビットが 0 の場合に、ロックビットが 0 に設定された消去ブロックに対してプログラム/ロックビットプログラム コマンドを発行	0	0	1	0	0	0	0
FCU エラー	FCU 内部の CPU 処理でエラー発生	0	0	0	1	0	0	0
FCU RAM ECC エラー	FCU RAM 読み出し時に 1 ビット誤り訂正発生	0	0	0	0	0	1	0
	FCU RAM 読み出し時に 2 ビット誤り検出発生	0	0	0	0	1	0	0
ROM アクセス 違反	FENTRY0=1 かつフラッシュメモリ P/E ノーマルモードの場合に、H'D00000 ~H'D7FFFF に対するリードアクセスを発行。	1	0	0	0	0	0	1
	FENTRY0=0 で H'D00000~H'D7FFFF に対するアクセスを発行	1	0	0	0	0	0	1
	FENTRY0=1 で H'00000~H'07FFFF に対してリードアクセスを発行	1	0	0	0	0	0	1

20.9 使用上の注意事項

20.9.1 その他のご注意

(1) 割り込み無視状態

以下の状態では、NMI やマスカブル割り込みが発生しても無視されます。

- ブートモード動作中
- ライタモード動作中

(2) 書き込み／消去サスペンド対象領域

書き込み／消去サスペンド中の領域の格納データは不定です。不定データの読み出しが原因で発生する誤動作を回避するために、書き込み／消去サスペンド対象領域の命令実行や、データ読み出しが発生しないように注意してください。

(3) 従来の F-ZTAT H8SX マイコンとの書き込み／消去プログラムの互換性

従来の F-ZTAT H8SX マイコンで使用していたフラッシュメモリの書き込み／消去プログラムは、本 LSI では動作しません。

(4) 書き込み／消去中のリセット

書き込み／消去処理中に **FRESETR** レジスタの **FRESET** ビットをセットして **FCU** をリセットする場合には、リセット状態を **trESW2** 保持してください。書き込み／消去中のフラッシュメモリには高電圧が印加されているため、メモリに印加された電圧の降下に必要な期間を確保するために **FCU** のリセット状態を保持する必要があります。**FCU** をリセットしている期間はフラッシュメモリの読み出しは行わないでください。

書き込み／消去処理中に **RES** 端子のアサートによってリセットが発生させた場合には、リセット状態を **trESW2** 保持してください。リセットでは、メモリに印加された電圧の降下に必要な期間だけでなく、フラッシュメモリ用電源の初期化やフラッシュメモリの内部回路の初期化に必要な期間を確保する必要があります。

書き込み／消去処理中には **WDT** のカウンタのオーバフローによる内部リセットは発生しないようにしてください。**WDT** によるリセットでは、メモリ電圧の降下／フラッシュメモリ用電源の初期化／フラッシュメモリ内部の初期化に必要な期間を確保することができません。

書き込み／消去処理中に **RES** 端子のアサートによるリセット、**FRESETR** レジスタの **FRESET** ビットをセットしての **FCU** リセットを実行すると、書き込み／消去対象の全データは不定となります。

(5) 追加書き込み禁止

同一領域に 2 回以上の書き込みを行うことはできません。書き込み済みの領域を書き換えたい場合には、必ず該当領域を消去してください。

(6) 書き込み／消去サスペンドによる中断

書き込み／消去サスペンドコマンドによって書き込み／消去処理を中断した場合は、必ずレジュームコマンドにより動作を完了させてください。

21. データフラッシュ (EEPROM)

H8SX/1727S は 32KB、H8SX/1725S は 16KB は 16KB のデータ格納用のフラッシュメモリ (EEPROM) を内蔵しています。データフラッシュ (以降 EEPROM) の特長を以下に示します。

21.1 特長

- フラッシュメモリマツト

EEPROM には、同一アドレス空間に配置される 2 種類のメモリ空間 (以下メモリマツトと呼びます) があり、制御レジスタを使用したバンク切り替えでマツトを切り替えることができます。製品情報マツト選択時の場合でも、H'E00080~H'E07FFF 領域を読み出すとデータマツトの内容が読み出されます。製品情報マツトは書き込み/消去できません。

データマツト : 32KB (H8SX/1727S) 、16KB (H8SX/1725S)

製品情報マツト : 128 バイト

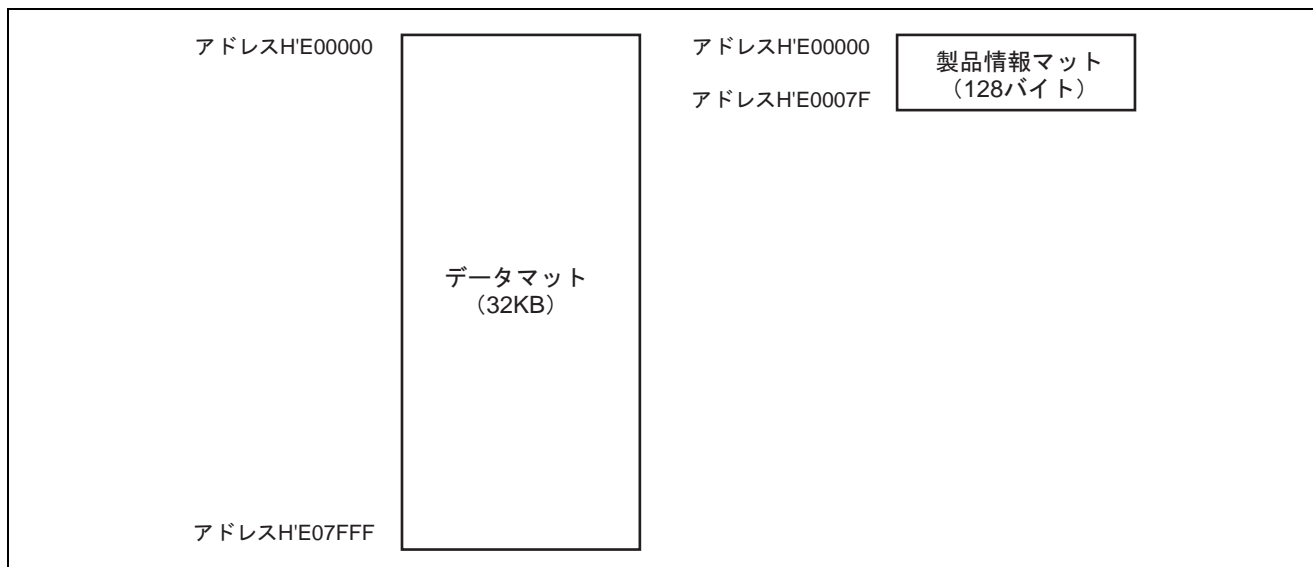


図 21.1 EEPROM のメモリマツト構成

- 【注】
1. 本 LSI は、ロックビットは使用できません。
 2. 本文中では 32KB の EEPROM についてのみ記載しています。
 3. 16KB のメモリマツト構成に関しては「3.3.4 アドレスマツプ」をご参照ください。

- 周辺バス経由で読み出し可能

データマツト、製品情報マツトともにバイト、ワードアクセス時には3周辺クロック (F ϕ) サイクル、ロングワードアクセス時には6F ϕ サイクルで、周辺バス経由の読み出しが可能です。

F ϕ クロックが20MHz以下のときにはSYSCR1レジスタのEEPWTビット (初期値=1) を0にすることでバイト、ワードアクセス時には2F ϕ サイクル、ロングワードアクセス時には4F ϕ サイクルで、周辺バス経由の読み出しが可能です。

表 21.1 周辺バス経由での読み出しサイクルの設定

XTAL	1 ϕ 通倍比	1 ϕ	F ϕ 通倍比	
			$\times 2$	$\times 4$
10MHz	$\times 4$	40MHz	F ϕ =20MHz EEPWT=0 でも可	F ϕ =40MHz EEPWT=1
	$\times 8$	80MHz	F ϕ =20MHz EEPWT=0 でも可	F ϕ =40MHz EEPWT=1
8MHz	$\times 4$	32MHz	F ϕ =16MHz EEPWT=0 でも可	F ϕ =32MHz EEPWT=1
	$\times 8$	64MHz	F ϕ =16MHz EEPWT=0 でも可	F ϕ =32MHz EEPWT=1

- 書き込み/消去方式

周辺バス経由でフラッシュメモリ/EEPROM専用のシーケンサ (FCU) にコマンドを発行することにより、データマツトの書き込み/消去を実行可能です。FCUがデータマツトの書き込み/消去を実行している期間でも、CPUはフラッシュメモリ/RAM/外部アドレス空間の領域に配置したプログラムを実行可能です。FCUがフラッシュメモリ/データマツトの書き込み/消去を実行している期間に、データマツトを読み出すことはできません。FCUによるデータマツトの書き込み/消去動作を中断してCPUでデータマツトを読み出した後、データマツトの書き込み/消去を再開することは可能です。消去を中断した場合には、中断した消去の対象ではない領域に書き込みを実施することも可能です。

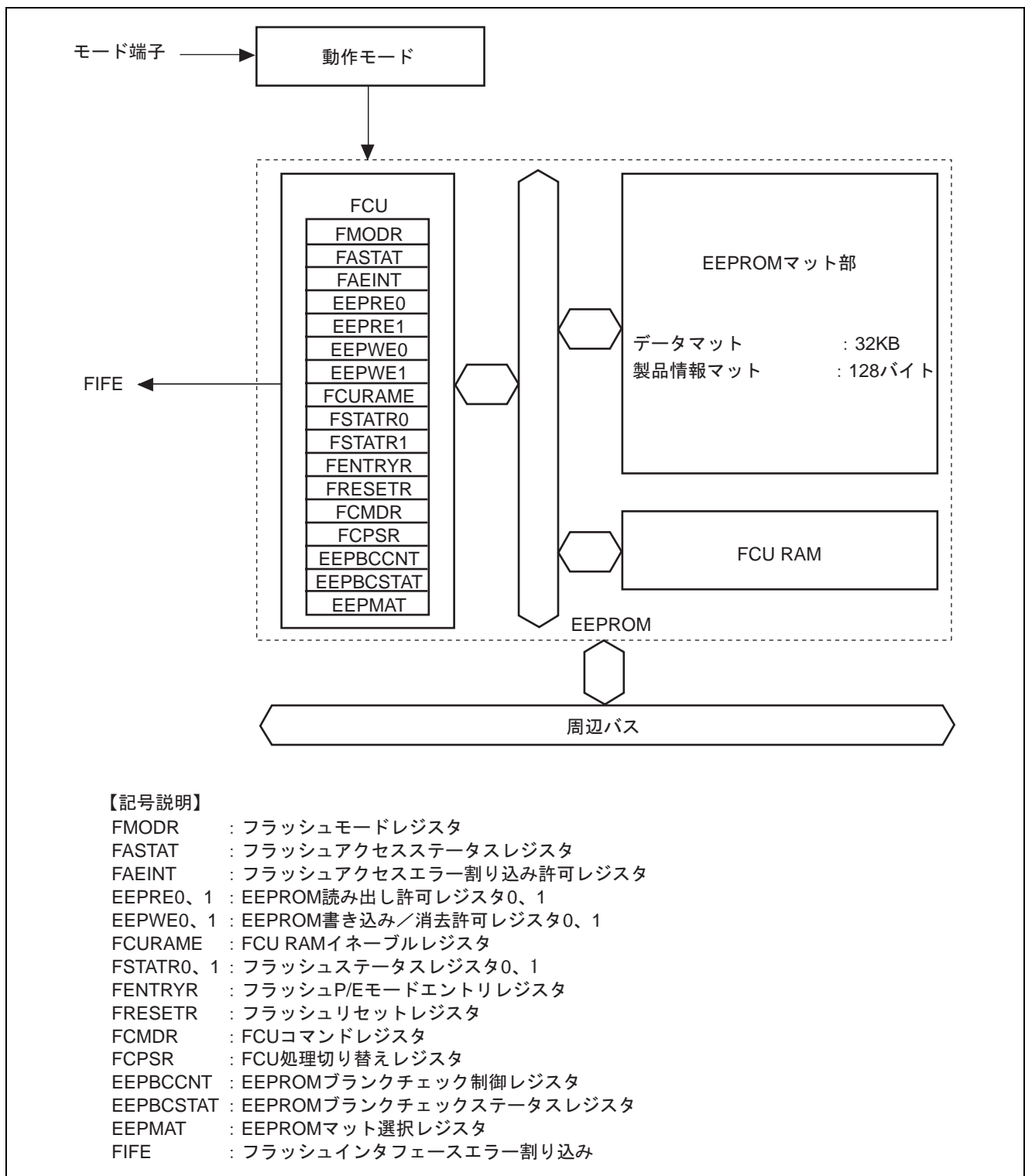


図 21.2 EEPROM のブロック図

- 書き込み/消去単位

ユーザモード/ユーザプログラムモードでのデータマットの書き込み単位は 8 バイトまたは 128 バイト、消去単位はブロック単位 (2KB) です。ブートモードでのデータマット書き込み単位は 128 バイト、消去単位はブロック単位 (2KB) です。製品情報マットは読み出し専用で、書き込み/消去はできません。図 21.3 に本 LSI のデータマットのブロック分割を示します。データマットは、H8SX/1727S の場合、2KB (16 ブロック : DB00~DB15) に分割され、H8SX/1725S の場合、2KB (8 ブロック : DB00~DB07) に分割されています。

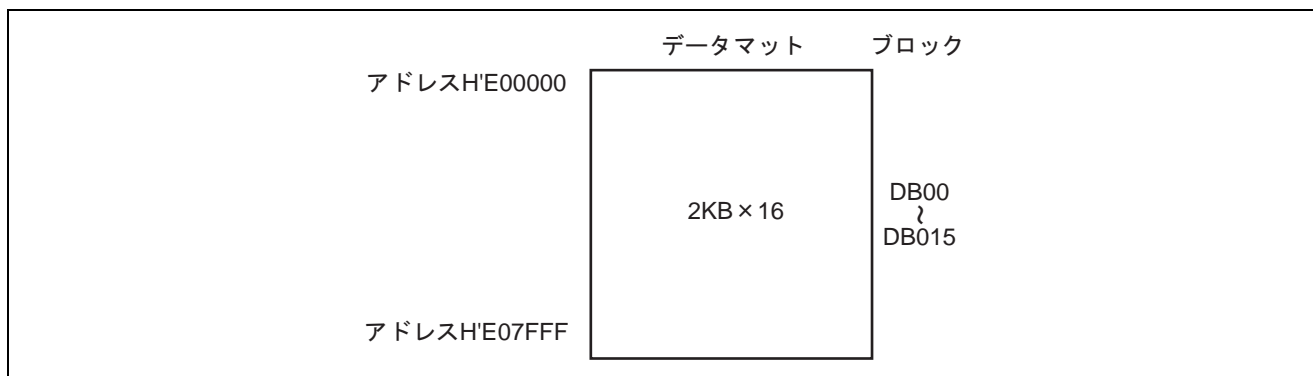


図 21.3 データマットのブロック分割

- ブランクチェック機能

消去状態の EEPROM を CPU から読み出すと不定データが読み出されます。FCU のブランクチェックコマンドを使用すると EEPROM が消去状態 (ブランク状態) であるか確認することができます。1 回のブランクチェックコマンドで確認可能な領域のサイズは、2K バイト (1 消去ブロック) または 8 バイトです。

- オンボードプログラミングモード (2種類)

ブートモード : SCI を使用してデータマットを書き換え可能なプログラムモードです。ホストと本 LSI 間の SCI 通信のビットレートは自動調整可能です。

ユーザモード/ユーザプログラムモード : 任意のインタフェースで、データマットを書き換え可能なプログラムモードです。

- プロテクトモード

モード端子によるハードウェアプロテクトと FENTRYD ビット/EEPWE0、1 レジスタ/EEPWE0、1 レジスタによるソフトウェアプロテクトの 2 種類のプロテクトモードがあり、書き込み/消去/読み出しに対するプロテクト状態を設定可能です。FENTRYD ビットは FCU によるデータマット書き込み/消去処理の許可/禁止を制御するためのビットです。EEPWE0、1 レジスタはデータマットの各ブロックの読み出しプロテクトを制御するためのレジスタです。EEPWE0、1 レジスタはデータマットの各ブロックの書き込み/消去プロテクトを制御するためのレジスタです。EEPWE0、1 レジスタの設定を変更してもデータマットの内容は保持されます。

書き込み/消去中に異常発生を検出した場合、書き込み/消去処理を中断する機能もあります。また、CPU が EEPROM 領域から命令をフェッチした場合、アドレスエラー例外処理が発生します。詳細については「第 4 章 例外処理」を参照してください。

- 書き込み時間/消去時間/書き換え回数

「第 25 章 電気的特性」を参照してください。

21.2 入出力端子

表 21.2 に EEPROM 関連の入出力端子を示します。MD1、MD0 端子の組み合わせによって、EEPROM のプログラミングモードを決定します（「21.4 EEPROM 関連モード概要」を参照）。

表 21.2 端子構成

名称	端子名	入出力	機能
リセット	$\overline{\text{RES}}$	入力	この端子がローレベルになるとリセット状態になります。
モード設定	MD1、MD0	入力	動作モードを決定します。
SCI 受信データ	RxD4	入力	SCI の受信データ (チャンネル 4)
SCI 送信データ	TxD4	出力	SCI の送信データ (チャンネル 4)

21.3 レジスタの説明

表 21.3 に EEPROM 関連のレジスタを示します。一部のレジスタはフラッシュメモリ関連のビットも持ちますが、本章では EEPROM 機能に関連するビットの説明のみ記載しています。フラッシュメモリ/EEPROM 共用ビットのみで構成されるレジスタ (FMODR、FCURAME、FSTATR0、FSTATR1、FRESETR、FCMDR、FCPSR) の仕様とフラッシュメモリ専用ビットの機能の詳細は、「第 20 章 フラッシュメモリ」の「20.3 レジスタの説明」を参照してください。

表 21.3 レジスタ構成

レジスタ名	略称	R/W* ¹	初期値	アドレス	アクセスサイズ
フラッシュモードレジスタ	FMODR	R/W	H'00	H'FFE002	8
フラッシュアクセスステータスレジスタ	FASTAT	R/(W)* ²	H'00	H'FFE010	8
フラッシュアクセスエラー割り込み許可レジスタ	FAEINT	R/W	H'9F	H'FFE011	8
EEPROM 読み出し許可レジスタ 0	EEPWE0	R/(W)* ³	H'0000	H'FFE040	8、16
EEPROM 読み出し許可レジスタ 1	EEPWE1	R/(W)* ³	H'0000	H'FFE042	8、16
EEPROM 書き込み/消去許可レジスタ 0	EEPWE0	R/(W)* ³	H'0000	H'FFE050	8、16
EEPROM 書き込み/消去許可レジスタ 1	EEPWE1	R/(W)* ³	H'0000	H'FFE052	8、16
FCU RAM イネーブルレジスタ	FCURAME	R/(W)* ³	H'0000	H'FFE054	8、16
フラッシュステータスレジスタ 0	FSTATR0	R* ⁵	H'80	H'FFE100	8、16
フラッシュステータスレジスタ 1	FSTATR1	R* ⁵	H'00	H'FFE101	8、16
フラッシュ P/E モードエントリレジスタ	FENTRYR	R/(W)* ⁴ * ⁵	H'0000	H'FFE102	8、16
フラッシュリセットレジスタ	FRESETR	R/(W)* ³	H'0000	H'FFE106	8、16
FCU コマンドレジスタ	FCMDR	R* ⁵	H'FFFF	H'FFE10A	8、16
FCU 処理切り替えレジスタ	FCPSR	R/W* ⁵	H'0000	H'FFE118	8、16
EEPROM ブランクチェック制御レジスタ	EEPBCCNT	R/W* ⁵	H'0000	H'FFE11A	8、16
EEPROM ブランクチェックステータスレジスタ	EEPBCSTAT	R* ⁵	H'0000	H'FFE11E	8、16
フラッシュロック通知レジスタ	FCKAR	R/W* ⁵	H'0000	H'FFE138	16
EEPROM マット選択レジスタ	EEPMAT	R/(W)* ³	H'0000	H'FFE380	8、16

【注】 *1 内蔵 ROM が無効なモードでは、フラッシュメモリ関連レジスタの読み出しデータは 0 に固定され、書き込みは無効化されます (本 LSI には内蔵 ROM 無効モードはありません)。

*2 フラグをクリアするために 0 のみ書き込むことができるビットと読み出し専用ビットによって構成されています。

*3 ワードアクセスで上位バイトに特定の値を書き込んだ場合のみ書き込み有効です。上位バイトへの書き込みデータは保持されません。

*4 ワードアクセスで上位バイトに特定の値を書き込んだ場合のみ書き込み有効で、それ以外の書き込みを行った場合には初期化されます。上位バイトへの書き込みデータは保持されません。

*5 $\overline{\text{RES}}$ 端子からのリセット以外に、FRESETR レジスタの FRESET ビットを 1 にすることで初期化できます。

21.3.1 フラッシュモードレジスタ (FMODER)

FMODER は FCU の動作モードを指定するレジスタです。

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

ビット	ビット名	初期値	R/W	説明
7~5	—	すべて 0	R	リザーブビット 読み出すと常に 0 が読み出されます。書き込む値も常に 0 にしてください。
4	FRDMD	0	R/W	FCU リードモード選択ビット FCU を使用したフラッシュメモリ/EEPROM 読み出し処理の方法を選択するためのビットです。EEPROM の場合には、EEPROM ロックビットリードモード移行処理かブランクチェック処理を選択するために使用します。（「21.6.1 FCU コマンド一覧」、「21.6.3 FCU コマンド使用方法」を参照）。フラッシュメモリの場合には、ロックビット読み出し方法を選択するために FRDMD 使用します（「第 20 章 フラッシュメモリ」を参照）。 0 : メモリ領域リードモード EEPROM ロックビットモードに移行する場合には、メモリ領域モードに設定します。EEPROM にはロックビットが存在しないため、ロックビットモードに移行して EEPROM 領域から読み出しを実行した場合、不定データが読み出されます。 1 : レジスタリードモード ブランクチェックコマンドを使用する場合には、レジスタリードモードに設定します。
3~0	—	すべて 0	R	リザーブビット 読み出すと常に 0 が読み出されます。書き込む値も常に 0 にしてください。

21.3.2 フラッシュアクセスステータスレジスタ (FASTAT)

FASTAT は、フラッシュメモリ/EEPROM に対するアクセス違反の有無を確認するためのレジスタです。FASTAT レジスタのいずれかのビットが 1 にセットされると、FCU はコマンドロック状態になります(「21.7.3 エラープロテクト」を参照)。コマンドロック状態を解除するためには、FASTAT レジスタを H'10 に設定した後、FCU にステータスクリアコマンドを発行する必要があります。

ビット	7	6	5	4	3	2	1	0
ビット名	ROMAE	—	—	CMDLK	EEPAE	—	EEPRPE	EEPWPE
初期値:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R	R	R	R/(W)*	R	R/(W)*	R/(W)*

【注】 * フラグをクリアするため、1を読み出した後に0を書き込むことのみ可能です。

ビット	ビット名	初期値	R/W	説明
7	ROMAE	0	R/(W)*	フラッシュメモリアクセス違反ビット 「第 20 章 フラッシュメモリ」を参照してください。
6、5	—	すべて 0	R	リザーブビット 読み出すと常に 0 が読み出されます。書き込む値も常に 0 にしてください。
4	CMDLK	0	R	FCU コマンドロックビット FCU がコマンドロック状態であることを示すビットです(「21.7.3 エラープロテクト」を参照)。 0: FCU はコマンドロック状態ではない 1: FCU はコマンドロック状態 【セット条件】 FCU がエラーを検出してコマンドロック状態に遷移後 【クリア条件】 FASTAT レジスタが H'10 の状態で、FCU がステータスクリアコマンドを処理後

ビット	ビット名	初期値	R/W	説明
3	EEPAAE	0	R/(W)*	<p>EEPROM アクセス違反ビット</p> <p>EEPROM に対するアクセス違反の有無を示すビットです。EEPAAE ビットが 1 になると FSTATR0 レジスタの ILGLERR ビットが 1 にセットされ、FCU はコマンドロック状態になります。</p> <p>0 : EEPROM アクセス違反なし 1 : EEPROM アクセス違反あり</p> <p>【セット条件】</p> <ul style="list-style-type: none"> • FENTRYR レジスタの FENTRYD ビットが 1、かつ EEPROM P/E ノーマルモードで、EEPROM 領域に対してリードアクセスを発行 • FENTRYD ビットが 0 の状態で、EEPROM 領域に対してライトアクセスを発行 • FENTRYR レジスタの FENTRY0 ビットが 1 の状態で、EEPROM 領域に対するアクセスを発行 <p>「第 20 章 フラッシュメモリ」を参照してください。</p> <p>【クリア条件】</p> <ul style="list-style-type: none"> • EEPAAE=1 を読み出した後に、0 を書き込み
2	—	0	R	<p>リザーブビット</p> <p>読み出すと常に 0 が読み出されます。書き込む値も常に 0 にしてください。</p>
1	EEPRPE	0	R/(W)*	<p>EEPROM リードプロテクト違反ビット</p> <p>EEPRE0、1 レジスタで設定した読み出しプロテクトに対する違反の有無を示すビットです。</p> <p>0 : EEPRE0、1 設定に違反した EEPROM 読み出しなし 1 : EEPRE0、1 設定に違反した EEPROM 読み出しあり</p> <p>【セット条件】</p> <p>EEPRE0、1 レジスタで読み出し禁止に設定した EEPROM 領域に対してリードアクセスを発行</p> <p>【クリア条件】</p> <p>EEPRPE=1 を読み出した後に、0 を書き込み</p>

ビット	ビット名	初期値	R/W	説明
0	EEPWPE	0	R/(W)*	EEPROM 書き込み/消去プロテクト違反ビット EEPWE0、1 レジスタで設定した書き込み/消去プロテクトに対する違反の有無を示すビットです。 0: EEPWE0、1 設定に違反した EEPROM 書き込み/消去系コマンドの発行なし 1: EEPWE0、1 設定に違反した EEPROM 書き込み/消去系コマンドの発行あり 【セット条件】 EEPWE0、1 レジスタで書き込み/消去禁止に設定した EEPROM 領域に対して書き込み/消去系コマンドを発行 【クリア条件】 EEPWPE=1 を読み出した後に、0 を書き込み

【注】 * フラグをクリアするため、1 を読み出した後に 0 を書き込むことのみ可能です。

21.3.3 フラッシュアクセスエラー割り込み許可レジスタ (FAEINT)

FAEINT は、フラッシュインタフェースエラー割り込み (FIFE) の出力許可/禁止を設定するためのレジスタです。

ビット	7	6	5	4	3	2	1	0
ビット名	ROMAEIE	—	—	CMDLKIE	EEPAEIE	—	EEPRPEIE	EEPWPEIE
初期値:	1	0	0	1	1	1	1	1
R/W:	R/W	R	R	R/W	R/W	R	R/W	R/W

ビット	ビット名	初期値	R/W	説明
7	ROMAEIE	1	R/W	フラッシュメモリアクセス違反割り込みイネーブル 「第 20 章 フラッシュメモリ」を参照してください。
6、5	—	すべて 0	R	リザーブビット 読み出すと常に 0 が読み出されます。書き込む値も常に 0 にしてください。
4	CMDLKIE	1	R/W	FCU コマンドロック割り込みイネーブル FCU コマンドロックが発生し、FASTAT レジスタの CMDLK ビットが 1 になった場合の FIFE 割り込み要求の発生を許可/禁止するためのビットです。 0: CMDLK=1 で FIFE 割り込み要求を発生しない 1: CMDLK=1 で FIFE 割り込み要求を発生する

ビット	ビット名	初期値	R/W	説明
3	EEPAAIE	1	R/W	EEPROM アクセス違反割り込みイネーブル EEPROM アクセス違反が発生し、FASTAT レジスタの EEPAAIE ビットが 1 になった場合の FIFE 割り込み要求の発生を許可/禁止するためのビットです。 0 : EEPAAIE=1 で FIFE 割り込み要求を発生しない 1 : EEPAAIE=1 で FIFE 割り込み要求を発生する
2	—	1	R	リザーブビット 読み出すと常に 1 が読み出されます。書き込む値も常に 1 にしてください。
1	EEPRPEIE	1	R/W	EEPROM リードプロテクト違反割り込みイネーブル EEPROM リードプロテクト違反が発生し、FASTAT レジスタの EEPRPE ビットが 1 になった場合の FIFE 割り込み要求の発生を許可/禁止するためのビットです。 0 : EEPRPE=1 で FIFE 割り込み要求を発生しない 1 : EEPRPE=1 で FIFE 割り込み要求を発生する
0	EEPWPEIE	1	R/W	EEPROM 書き込み/消去プロテクト違反割り込みイネーブル EEPROM 書き込み/消去プロテクト違反が発生し、FASTAT レジスタの EEPWPE ビットが 1 になった場合の FIFE 割り込み要求の発生を許可/禁止するためのビットです。 0 : EEPWPE=1 で FIFE 割り込み要求を発生しない 1 : EEPWPE=1 で FIFE 割り込み要求を発生する

21.3.4 EEPROM 読み出し許可レジスタ 0 (EEPRE0)

EEPRE0 は、データマットの DB00~DB07 ブロック (図 21.3 を参照) の読み出しを許可/禁止するためのレジスタです。

ビット	15	14	13	12	11	10	9	8
ビット名	REKEY07	REKEY06	REKEY05	REKEY04	REKEY03	REKEY02	REKEY01	REKEY00
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)*
ビット	7	6	5	4	3	2	1	0
ビット名	DBRE07	DBRE06	DBRE05	DBRE04	DBRE03	DBRE02	DBRE01	DBRE00
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】 * 書き込みデータは保持されません。

ビット	ビット名	初期値	R/W	説明
15~8	REKEY07 ~ REKEY00	すべて0	R/(W)*	キーコード DBRE07~DBRE00 ビットの書き換えの可否を制御します。本ビットへの書き込みデータは保持されません。
7~0	DBRE07 ~ DBRE00	0	R/W	DB07~DB00 ブロック読み出し許可ビット データマットの DB07~DB00 ブロックに対する読み出しの許可/禁止を設定するビットです。DBREi ビット (i=07~00) を DBi ブロックの読み出し制御に使用します。 DBRE07~DBRE00 ビットへの書き込みは、ワードアクセスで KEY が H'2D の場合のみ有効です。 0: 読み出し禁止 1: 読み出し許可

【注】 * 書き込みデータは保持されません。

21.3.5 EEPROM 読み出し許可レジスタ 1 (EEPRES1)

EEPRES1 は、データマットの DB08~DB15 ブロック (図 21.3 を参照) の読み出しを許可/禁止するためのレジスタです。

ビット	15	14	13	12	11	10	9	8
ビット名	REKEY15	REKEY14	REKEY13	REKEY12	REKEY11	REKEY10	REKEY09	REKEY08
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)*
ビット	7	6	5	4	3	2	1	0
ビット名	DBRE15	DBRE14	DBRE13	DBRE12	DBRE11	DBRE10	DBRE09	DBRE08
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】 * 書き込みデータは保持されません。

ビット	ビット名	初期値	R/W	説明
15~8	REKEY15 ~ REKEY08	すべて0	R/(W)*	キーコード DBRE15~DBRE08 ビットの書き換えの可否を制御します。本ビットへの書き込みデータは保持されません。
7~0	DBRE15 ~ DBRE08	0	R/W	DB15~DB08 ブロック読み出し許可ビット データマットの DB15~DB08 ブロックに対する読み出しの許可/禁止を設定するビットです。DBREi ビット (i=15~08) を DBi ブロックの読み出し制御に使用します。 DBRE15~DBRE08 ビットへの書き込みは、ワードアクセスで KEY が H'D2 の場合のみ有効です。 0: 読み出し禁止 1: 読み出し許可

【注】 * 書き込みデータは保持されません。

21.3.6 EEPROM 書き込み/消去許可レジスタ 0 (EEPWE0)

EEPWE0 は、データマットの DB00~DB07 ブロック (図 21.3 を参照) の書き込み/消去を許可/禁止するためのレジスタです。

ビット	15	14	13	12	11	10	9	8
ビット名	WEKEY07	WEKEY06	WEKEY05	WEKEY04	WEKEY03	WEKEY02	WEKEY01	WEKEY00
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)*

ビット	7	6	5	4	3	2	1	0
ビット名	DBWE07	DBWE06	DBWE05	DBWE04	DBWE03	DBWE02	DBWE01	DBWE00
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】 * 書き込みデータは保持されません。

ビット	ビット名	初期値	R/W	説明
15~8	WEKEY07 ~ WEKEY00	すべて0	R/(W)*	キーコード DBWE07~DBWE00 ビットの書き換えの可否を制御します。本ビットへの書き込みデータは保持されません。
7~0	DBWE07 ~ DBWE00	0	R/W	DB07~DB00 ブロック書き込み/消去許可ビット データマットの DB07~DB00 ブロックに対する書き込み/消去の許可/禁止を設定するビットです。DBWE _i ビット (i=07~00) を DB _i ブロックの書き込み/消去制御に使用します。 DBWE07~DBWE00 ビットへの書き込みは、ワードアクセスで KEY が H'1E の場合のみ有効です。 0: 書き込み/消去禁止 1: 書き込み/消去許可

【注】 * 書き込みデータは保持されません。

21.3.7 EEPROM 書き込み/消去許可レジスタ 1 (EEPWE1)

EEPWE1 は、データマットの DB08~DB15 ブロック (図 21.3 を参照) の書き込み/消去を許可/禁止するためのレジスタです。

ビット	15	14	13	12	11	10	9	8
ビット名	WEKEY15	WEKEY14	WEKEY13	WEKEY12	WEKEY11	WEKEY10	WEKEY09	WEKEY08
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)*
ビット	7	6	5	4	3	2	1	0
ビット名	DBWE15	DBWE14	DBWE13	DBWE12	DBWE11	DBWE10	DBWE09	DBWE08
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】 * 書き込みデータは保持されません。

ビット	ビット名	初期値	R/W	説明
15~8	WEKEY15 ~ WEKEY08	すべて0	R/(W)*	キーコード DBWE15~DBWE08 ビットの書き換えの可否を制御します。本ビットへの書き込みデータは保持されません。
7~0	DBWE15 ~ DBWE08	0	R/W	DB15~DB08 ブロック書き込み/消去許可ビット データマットの DB15~DB08 ブロックに対する書き込み/消去の許可/禁止を設定するビットです。DBWE _i ビット (i=15~08) を DB _i ブロックの書き込み/消去制御に使用します。 DBWE15~DBWE08 ビットへの書き込みは、ワードアクセスで KEY が H'E1 の場合のみ有効です。 0: 書き込み/消去禁止 1: 書き込み/消去許可

【注】 * 書き込みデータは保持されません。

21.3.8 フラッシュ P/E モードエントリレジスタ (FENTRYR)

FENTRYR は、フラッシュメモリ/EEPROM を P/E モードに設定するために使用するレジスタです。EEPROM を P/E モードにして FCU のコマンド受付を可能にするためには、FENTRYD ビットに 1 を設定する必要があります。

ビット	15	14	13	12	11	10	9	8
ビット名	FEKEY7	FEKEY6	FEKEY5	FEKEY4	FEKEY3	FEKEY2	FEKEY1	FEKEY0
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)*
ビット	7	6	5	4	3	2	1	0
ビット名	FENTRYD	—	—	—	—	—	—	FENTRY0
初期値:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R/W

【注】 * 書き込みデータは保持されません。

ビット	ビット名	初期値	R/W	説明
15~8	FEKEY7 ~ FEKEY0	すべて 0	R/(W)*	キーコード FENTRYD、FENTRY0 ビットの書き換えの可否を制御します。本ビットへの書き込みデータは保持されません。
7	FENTRYD	0	R/W	EEPROM P/E モードエントリビット EEPROM を P/E モードに設定するためのビットです。 0 : EEPROM はリードモード 1 : EEPROM は P/E モード 【書き込み有効条件】 以下の全条件を満たす場合 <ul style="list-style-type: none"> 内蔵 ROM が有効なモード FSTATR0 レジスタの FRDY ビットが 1 ワードアクセスで FEKEY に H'AA 書き込み 【セット条件】 <ul style="list-style-type: none"> 書き込み有効条件を満たし、かつ FENTRYR レジスタが H'0000 の状態で、FENTRYD に 1 を書き込んだ場合 【クリア条件】 <ul style="list-style-type: none"> バイトアクセスで書き込んだ場合 ワードアクセスで FEKEY が H'AA 以外の状態で書き込んだ場合 書き込み有効条件を満たした状態で、FENTRYD に 0 を書き込んだ場合 書き込み有効条件を満たし、かつ FENTRYR レジスタが H'0000 以外の状態で、FENTRYR レジスタを書き込んだ場合

ビット	ビット名	初期値	R/W	説明
6~1	—	すべて0	R	リザーブビット 読み出すと常に0が読み出されます。書き込む値も常に0にしてください。
0	FENTRY0	0	R/W	フラッシュメモリ P/E モードエントリビット0 「第20章 フラッシュメモリ」を参照してください。

【注】 * 書き込みデータは保持されません。

21.3.9 EEPROM ブランクチェック制御レジスタ (EEPBCCNT)

EEPBCCNT レジスタは、ブランクチェックコマンドのチェック対象領域のアドレスとサイズを指定するためのレジスタです。EEPBCCNT は、 \overline{RES} 端子からのリセット以外に、FRESETR レジスタの FRESET ビットを1にすることによっても初期化されます。

ビット	15	14	13	12	11	10	9	8
ビット名	—	—	BCADR	BCADR	BCADR	BCADR	BCADR	BCADR
初期値 :	0	0	0	0	0	0	0	0
R/W :	R	R	R/W	R/W	R/W	R/W	R/W	R/W
ビット	7	6	5	4	3	2	1	0
ビット名	BCADR	BCADR	BCADR	BCADR	BCADR	—	—	BCSIZE
初期値 :	0	0	0	0	0	0	0	0
R/W :	R/W	R/W	R/W	R/W	R/W	R	R	R/W

ビット	ビット名	初期値	R/W	説明
15~14	—	すべて0	R	リザーブビット 書き込む値は0にしてください。1を書き込んだ場合の動作は保証しません。
13~3	BCADR	すべて0	R/W	ブランクチェックアドレス設定ビット ブランクチェックコマンドのチェック対象領域のサイズが8バイト (BCSIZE ビットが0) の場合に、チェック対象領域のアドレスを設定するためのビットです。BCSIZE が0の場合には、EEPBCCNT レジスタの設定値 (BCADR の設定値を MSB 側に3ビットシフトした値) とブランクチェックコマンド発行時に指定した消去ブロック先頭アドレスを加算した値がチェック対象領域の先頭アドレスになります。
2~1	—	すべて0	R	リザーブビット 書き込む値は0にしてください。1を書き込んだ場合の動作は保証しません。

ビット	ビット名	初期値	R/W	説明
0	BCSIZE	0	R/W	ブランクチェックサイズ設定ビット ブランクチェックコマンドのチェック対象領域のサイズを設定するためのビットです。 0: ブランクチェック対象領域は8バイト 1: ブランクチェック対象領域は2Kバイト

21.3.10 EEPROM ブランクチェックステータスレジスタ (EEPBCSTAT)

EEPBCSTAT レジスタは、ブランクチェックコマンドの処理結果が格納されるレジスタです。EEPBCSTAT は、 $\overline{\text{RES}}$ 端子からのリセット以外に、FRESETR レジスタの FRESET ビットを 1 にすることによっても初期化されます。

ビット	15	14	13	12	11	10	9	8
ビット名	—	—	—	—	—	—	—	—
初期値:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R
ビット	7	6	5	4	3	2	1	0
ビット名	—	—	—	—	—	—	—	BCST
初期値:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

ビット	ビット名	初期値	R/W	説明
15~1	—	すべて0	R	リザーブビット 書き込む値は0にしてください。1を書き込んだ場合の動作は保証しません。
0	BCST	0	R	ブランクチェックステータスビット ブランクチェックの結果を示すビットです。 0: ブランクチェック対象領域は消去状態 (ブランク) 1: ブランクチェック対象領域は0データか1データが書き込まれた状態

21.3.11 EEPROM マット選択レジスタ (EEPSTAT)

EEPSTAT は、EEPROM のマットを切り替えるために使用するレジスタです。

ビット	15	14	13	12	11	10	9	8
ビット名	EMKEY7	EMKEY6	EMKEY5	EMKEY4	EMKEY3	EMKEY2	EMKEY1	EMKEY0
初期値:	0	0	0	0	0	0	0	0
R/W:	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1
ビット	7	6	5	4	3	2	1	0
ビット名	—	—	—	—	—	—	—	EEPSEL
初期値:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

【注】 *1 書き込みデータは保持されません。

ビット	ビット名	初期値	R/W	説明
15~8	EMKEY7 ~ EMKEY0	すべて0	R/(W)*	キーコード EEPSEL ビットの書き換えの可否を制御します。本ビットへの書き込みデータは保持されません。
7~1	—	すべて0	R	リザーブビット 読み出すと常に0が読み出されます。書き込む値も常に0にしてください。
0	EEPSEL	0	R/W	EEPROM マット選択ビット EEPROM のマットを選択するためのビットです。EEPSTAT ビットへの書き込みは、ワードアクセスで EMKEY が H'B3 の場合のみ有効です。 0: データマット選択 1: 製品情報マット選択*2

【注】 *1 書き込みデータは保持されません。

*2 製品情報マットは読み出し専用であり、書き込み/消去はできません。

21.4 EEPROM 関連モード概要

図 21.4 に本 LSI のモード遷移図 (EEPROM 関連) を示します。MD1、MD0 端子の設定値と本 LSI の動作モードの関係については「第 3 章 MCU 動作モード」を参照してください。

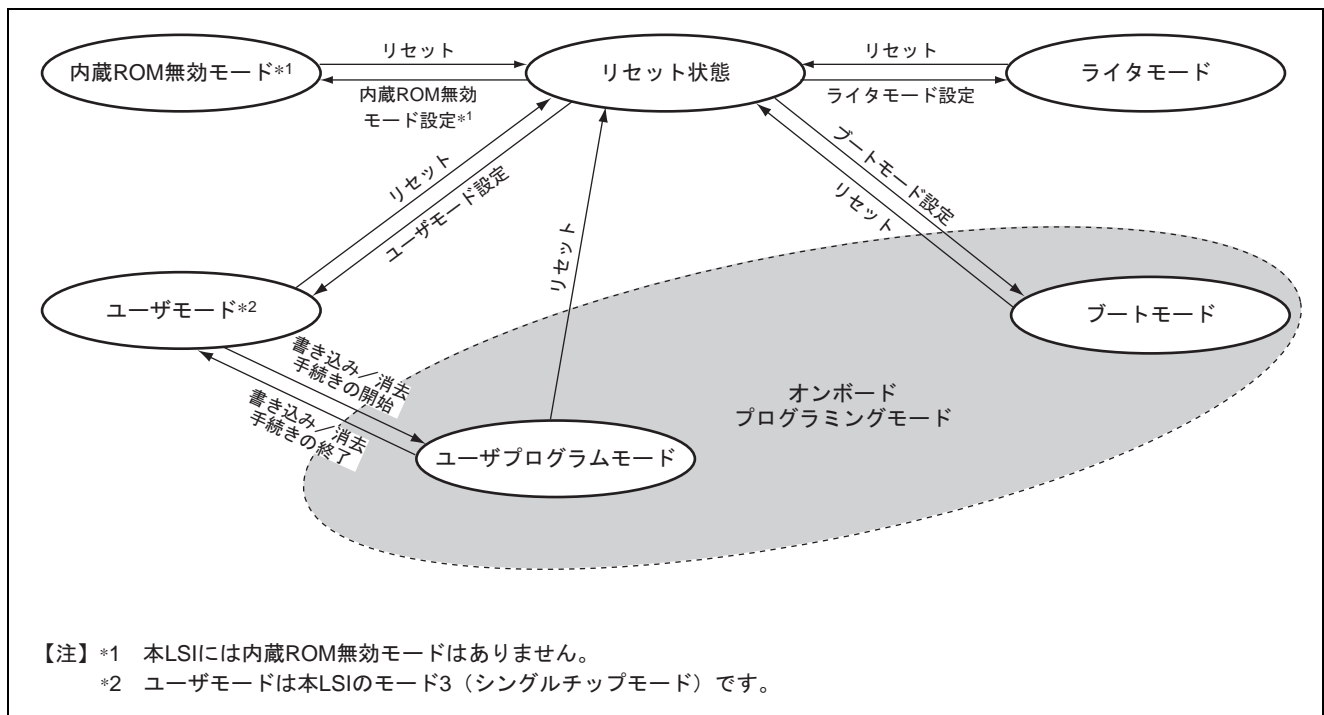


図 21.4 EEPROM に関するモード遷移図

- 内蔵ROM無効モード/ライターモードでは、EEPROMの読み出し/書き込み/消去は実施できません。本 LSI では内蔵 ROM 無効モードをサポートしていません。
- ユーザーモード/ユーザープログラムモード/ブートモードでは、オンボードでデータマットの読み出し/書き込み/消去を実施できます。
- ユーザーモードではフラッシュメモリを書き込み/消去できませんが、EEPROMを書き込み/消去できます。また、EEPROMの書き込み/消去中にはフラッシュメモリを読み出すことができます。このため、書き込み/消去プロテクトされたフラッシュメモリ上のアプリケーションプログラムを実行しながら、データをEEPROMに書き込むことができます。

表 21.4 に、ブートモード、ユーザモード、ユーザプログラムモード書き込み/消去関連項目の比較を示します。

表 21.4 プログラミングモードの比較

項目	ブートモード	ユーザモード	ユーザプログラムモード ^{*2}
書き込み/消去環境	オンボードプログラミング		
書き込み/消去可能マツト	データマツト	データマツト	データマツト
書き込み/消去制御	ホスト	FCU	FCU
全面消去	○ (自動)	○	○
ブロック分割消去	○ ^{*1}	○	○
書き込みデータ転送	ホストから SCI 経由	任意のデバイスから RAM 経由	任意のデバイスから RAM 経由
リセット時の起動マツト	組み込みプログラム 格納マツト	ユーザマツト	ユーザマツト

【注】 *1 起動時に全面消去されます。その後、特定ブロックの消去を実施可能です。

*2 本 LSI では、ユーザモードにて規定の書き込み/消去の手続きを開始したところから手続きの終了したところまでをユーザプログラムモードと規定します。書き込み/消去の手続きは、「21.6 ユーザモード/ユーザプログラムモード」を参照してください。

- ブートモードでは、起動直後にフラッシュメモリのユーザマツトとデータマツトが全面消去されます。その後、ホストからSCI経由でのデータマツトの書き込みが可能になります。データマツトの読み出しも、起動直後の全面消去後に可能になります。
- ブートモード/ライターモードではブートプログラムで内蔵RAMを使用します。このため、RAMイネーブルレジスタ (RAMEN) の設定で内蔵RAMを無効にしてリセットし、ブートモード/ライターモードで起動した場合には、リセット前の内蔵RAMのデータは保持されません (「第19章 RAM」を参照)。

21.5 ブートモード

ブートモードでは、ホストから制御コマンドや書き込みデータを送信してデータマットの書き込み／消去を実行可能です。ブートモードのシステム構成や使用方法の詳細は、「第 20 章 フラッシュメモリ」を参照してください。本節では、EEPROM 専用のコマンドの説明を記載します。

21.5.1 問い合わせ設定ホストコマンド

表 21.5 に、EEPROM 専用の問い合わせ設定ホストコマンド一覧を示します。データマット有無問い合わせ／データマット情報問い合わせコマンドは、「20.5.4 問い合わせ設定ホストコマンド待ち状態」の図 20.9 に示したフロー中の「マット書き込み用の情報問い合わせ」を実施する箇所で使用します。

表 21.5 問い合わせ設定ホストコマンド (EEPROM 専用)

ホストコマンド名	機 能
データマット有無問い合わせ	データマット有無の問い合わせ
データマット情報問い合わせ	データマットの個数、先頭／最終アドレスの問い合わせ

各ホストコマンドの詳細を以下に説明します。説明文中の「コマンド」はホストから本 LSI に送信するコマンド、「レスポンス」は本 LSI からホストに送信する応答です。「サムチェック」は、本 LSI が送信した各バイトを合計した場合に、H'00 になるように計算されたバイトデータを指します。

(1) データマット有無問い合わせ

ホストがデータマット有無問い合わせコマンドを送信すると、データマットがあることを示す情報を本 LSI が送信します。

コマンド	H'2A			
レスポンス	H'3A	サイズ	マット有無	SUM

【記号説明】

サイズ (1 バイト) : マット有無の文字数 (固定値で 1)

マット有無 (1 バイト) : データマットの有無 (固定値で H'01)

H'00 : データマットなし

H'01 : データマットあり

SUM (1 バイト) : サムチェック

(2) データマツト情報問い合わせ

ホストがデータマツト情報問い合わせコマンドを送信すると、データマツトのエリア数とアドレスの情報を本 LSI が送信します。

コマンド	H'2B		
レスポンス	H'3B	サイズ	エリア数
	エリア先頭アドレス		
	エリア最終アドレス		
	エリア先頭アドレス		
	エリア最終アドレス		
	...		
	エリア先頭アドレス		
	エリア最終アドレス		
	SUM		

【記号説明】

サイズ (1 バイト) : エリア数、エリア先頭アドレス、エリア最終アドレスのデータの総バイト数

エリア数 (1 バイト) : データマツトのエリア数 (連続したエリアは 1 エリアと数えます)

エリア先頭アドレス (4 バイト) : データマツトエリアの先頭アドレス

エリア最終アドレス (4 バイト) : データマツトエリアの最終アドレス

SUM (1 バイト) : サムチェック

データマツトのブロック構成の情報は、消去ブロック情報問い合わせコマンド (「20.5.4 問い合わせ設定ホストコマンド待ち状態」を参照) のレスポンスに含まれます。

21.5.2 書き込み/消去ホストコマンド

表 21.6 に、EEPROM 専用の書き込み/消去ホストコマンド一覧を示します。EEPROM 専用のホストコマンドはデータマツトのサムチェック/ブランクチェック用のコマンドのみで、書き込み/消去/読み出し用のコマンドはフラッシュメモリと共用です。

データマツトの書き込みを行う場合は、ユーザマツト書き込み選択コマンドを発行後、128 バイト書き込みコマンドで書き込みアドレスにデータマツトのアドレスを指定します。データマツトの消去を行う場合は、消去選択コマンド発行後、ブロック消去コマンドでデータマツトの消去ブロックを指定します。データマツトの消去ブロック情報は、消去ブロック問い合わせコマンドのレスポンスに含まれます。データマツトの読み出しを行う場合は、メモリリードコマンドでユーザマツトを選択し、読み出し対象アドレスにデータマツトのアドレスを指定します。

ユーザマツト書き込み選択コマンド/128 バイト書き込みコマンド/消去選択コマンド/ブロック消去コマンド/メモリリードコマンドの詳細は、「20.5.5 書き込み/消去ホストコマンド待ち状態」を参照してください。消去ブロック情報問い合わせコマンドの詳細は、「20.5.4 問い合わせ設定ホストコマンド待ち状態」を参照してください。

表 21.6 書き込み/消去ホストコマンド (EEPROM 専用)

ホストコマンド名	機 能
データマツサムチェック	データマツのサムチェック
データマツブランクチェック	データマツのブランクチェック

各ホストコマンドの詳細を以下に説明します。説明文中の「コマンド」はホストから本 LSI に送信するコマンド、「レスポンス」は本 LSI からホストに送信する応答です。「サムチェック」は、本 LSI が送信した各バイトを合計した場合に、H'00 になるように計算されたバイトデータを指します。

(1) データマツサムチェック

ホストがデータマツサムチェックコマンドを送信すると、本 LSI はデータマツのデータをバイト単位で加算した結果 (サムチェック) を送信します。

コマンド	H'61			
レスポンス	H'71	サイズ	マツのサムチェック	SUM

【記号説明】

サイズ (1 バイト) : マツのサムチェックのバイト数 (固定値で 4)

マツのサムチェック (4 バイト) : データマツのサムチェック結果

SUM (4 バイト) : サムチェック (レスポンスデータのサムチェック)

(2) データマツブランクチェック

ホストがデータマツブランクチェックコマンドを送信すると、本 LSI はデータマツがすべて消去状態であるかをチェックします。データマツが全て消去状態であった場合には、本 LSI はレスポンス (H'06) を送信します。データマツに未消去領域が存在した場合には、本 LSI はエラーレスポンス (H'E2→H'52) を送信します。

コマンド	H'62	
レスポンス	H'06	
エラー レスポンス	H'E2	H'52

21.6 ユーザモード/ユーザプログラムモード

21.6.1 FCU コマンド一覧

ユーザモード/ユーザプログラムモードでは、FCU へ FCU コマンドを発行してデータマットの書き込み/消去を実行します。表 21.7 に、EEPROM 書き込み/消去で使用可能な FCU コマンドの一覧を示します。

表 21.7 FCU コマンド一覧 (EEPROM 関連)

コマンド	機能
ノーマルモード移行	ノーマルモードに遷移 (「21.6.2 FCU コマンド受け付け条件」を参照)
ステータスリードモード移行	ステータスリードモードに遷移 (「21.6.2 FCU コマンド受け付け条件」を参照)
ロックビットリードモード移行 (ロックビットリード1)	ロックビットリードモードに遷移 (「21.6.2 FCU コマンド受け付け条件」を参照)
フラッシュロック通知	フラッシュロックの周波数を設定
プログラム	EEPROM 書き込み (8 バイトまたは 128 バイト単位)
ブロックイレーズ	EEPROM 消去 (ブロック単位)
P/E サスペンド	書き込み/消去の中断
P/E レジューム	書き込み/消去の再開
ステータスレジスタクリア	FSTATR0 レジスタの ILGLERR,ERSERR,PRGERR ビットのクリアとコマンドロック状態の解除
ブランクチェック	指定した領域が消去状態 (ブランク) であるか確認

表 21.7 のブランクチェックコマンド以外の FCU コマンドは、フラッシュメモリ書き込み/消去でも使用します。フラッシュメモリに対してブランクチェックコマンドを発行した場合には、フラッシュメモリのロックビット読み出しが実行されます。

FCU へのコマンド発行は、EEPROM 領域に対する周辺バスライトアクセスで実現されます。表 21.8 にプログラムコマンドとブランクチェックコマンドのフォーマットを示します。ブランクチェックコマンド以外の FCU コマンドのフォーマットは、「21.6.1 FCU コマンド一覧」を参照してください。表 21.8 に示した周辺バスアクセスを特定条件下で実行すると、FCU は各コマンドに対応した処理を実行します。FCU コマンドの受け付け条件については、「21.6.2 FCU コマンド受け付け条件」を参照してください。また、コマンドの使用方法については、「21.6.3 FCU コマンド使用方法」を参照してください。

FRDMD ビットが 0 (メモリ領域モード) で FCU コマンドの 1 サイクル目が H'71 の場合には、FCU はロックビットリードモード移行コマンド (ロックビットリード 1) を受け付けます。EEPROM にはロックビットが存在しないため、ロックビットリードモードに移行後に EEPROM 領域に対して周辺バスリードアクセスを実行すると不定データが読み出されます。この不定データ読み出し時には、FCU はエラーを検出しません。

FRDMD が 1 (レジスタリードモード) で FCU コマンドの 1 サイクル目が H'71 の場合には、FCU はブランクチェックコマンドの 2 サイクル目のコマンド (H'D0) 待ちの状態になります。この状態で、EEPROM 領域に対して周辺バスライトアクセスで H'D0 を書き込むと、FCU は EEPBCCNT レジスタの設定値に従ったブランクチェック処理を実行し、ブランクチェック完了後に EEPBCSTAT レジスタに結果を反映します。

P/E サスペンドコマンド発行時のサスペンド動作にはサスペンド優先モードと消去優先モードがあります。各モードでの動作の詳細は「20.6.4 サスペンド動作」を参照してください。

表 21.8 プログラムコマンドおよびブランクチェックコマンドのフォーマット

コマンド	バス サイクル数	1 サイクル目		2 サイクル目		3 サイクル目		4~N+2 サイクル目		N+3 サイクル目	
		アドレス	データ	アドレス	データ	アドレス	データ	アドレス	データ	アドレス	データ
プログラム (8 バイト書き込み: N=4)	7	EA	H'E8	EA	H'04	WA	WD1	EA	WDn	EA	H'D0
プログラム (128 バイト書き込み: N=64)	67	EA	H'E8	EA	H'40	WA	WD1	EA	WDn	EA	H'D0
ブランクチェック	2	EA	H'71	BA	H'D0	—	—	—	—	—	—

【記号説明】

EA : EEPROM 領域のアドレス

H'E00000~H'E07FFF の任意のアドレス

WA : EEPROM 書き込み先頭アドレス

8 バイト書き込みの場合 : 書き込みデータ 8 バイトの先頭アドレス

128 バイト書き込みの場合 : 書き込みデータ 128 バイトの先頭アドレス

BA : EEPROM 消去ブロックアドレス

対象消去ブロック内の任意アドレス (書き込み/消去用アドレスで指定)

WDn : 書き込みデータ n ワード目 (n=1~N)

21.6.2 FCU コマンド受け付け条件

FCU コマンドの受け付け可否は、FCU のモード／状態に依存します。図 21.5 に FCU のモード遷移図を示します。

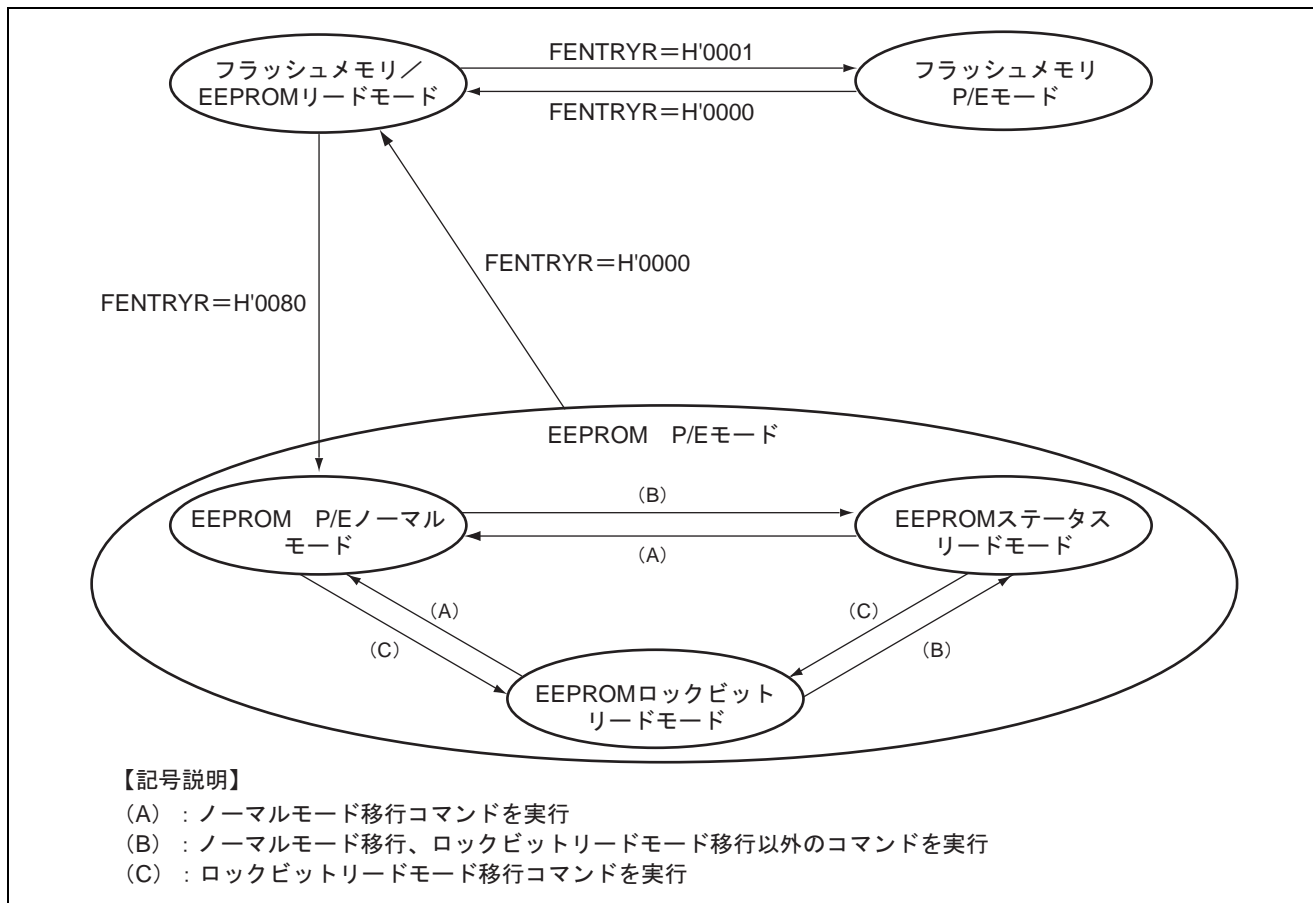


図 21.5 FCU のモード遷移図 (EEPROM 関連)

(1) フラッシュメモリ P/E モード

フラッシュメモリ書き込み／消去用の FCU コマンドを受け付け可能なモードです。EEPROM を読み出すことはできません。FENTRYR レジスタの FENTRYD ビットを 0、かつ FENTRY0 ビットを 1 に設定した場合に遷移するモードです。フラッシュメモリ P/E モードの詳細は、「21.6.2 FCU コマンド受け付け条件」を参照してください。

(2) フラッシュメモリ/EEPROM リードモード

周辺バス経由で EEPROM を、内部バス経由でフラッシュメモリを高速読み出し可能なモードです。FCU コマンドは受け付けられません。FENTRYD ビットを 0、かつ FENTRY0 ビットを 0 に設定した場合に、このモードに遷移します。

(3) EEPROM P/E モード

- EEPROM P/E ノーマルモード

フラッシュメモリ/EEPROM リードモードまたはフラッシュメモリ P/E モード時に FENTRYD ビットを 1 かつ FENTRY0 ビットを 0 に設定した場合、または EEPROM P/E モードでノーマルモード移行コマンドを受け付けた場合に遷移するモードです。表 21.9 に受け付け可能なコマンドを示します。EEPROM 領域に対して周辺バスリードアクセスを発行した場合には、EEPROM アクセス違反が発生して FCU はコマンドロック状態になります。フラッシュメモリは高速読み出し可能です。

- EEPROM ステータスリードモード

EEPROM P/E モードでノーマルモード移行、ロックビットリードモード移行以外のコマンドを受け付けた場合に遷移するモードです。FSTATR0 レジスタの FRDY が 0 の状態やエラー発生後のコマンドロック状態も、EEPROM ステータスリードモード中の状態です。表 21.9 に受け付け可能なコマンドを示します。EEPROM 領域に対して周辺バスリードアクセスを発行した場合には、FSTATR0 レジスタの値が読み出されます。フラッシュメモリは高速読み出し可能です。

- EEPROM ロックビットリードモード

EEPROM P/E モードでロックビットリードモード移行コマンドを受け付けた場合に遷移するモードです。表 21.9 に受け付け可能なコマンドを示します。EEPROM にはロックビットが存在しないため、EEPROM 領域に対して周辺バスリードアクセスを発行した場合には、読み出しデータは不定になりますが、EEPROM アクセス違反は発生しません。フラッシュメモリは高速読み出し可能です。

表 21.9 に EEPROM P/E モードの各モード/状態と受け付け可能なコマンドの関係を示します。受け付け不可能なコマンドが発行された場合には、FCU はコマンドロック状態になります（「21.7.3 エラープロテクト」を参照）。

FCU コマンドを確実に受け付けさせたい場合には、発行するコマンドを受付可能なモードに移行し、FSTATR0 レジスタの FRDY ビット、ILGLERR ビット、ERSERR ビット、PRGERR ビットと FSTATR1 レジスタの FCUERR ビットと FRDTCT ビット、FRCRCT ビットの値を確認した後に FCU コマンドを発行してください。FASTAT レジスタの CMDLK ビットの値は、FSTATR0 レジスタの ILGLERR ビット、ERSERR ビット、PRGERR ビットと FSTATR1 レジスタの FCUERR ビット、FRDTCT ビット、FRCRCT ビットの値の論理和です。このため、CMDLK ビットを確認して FCU エラー発生状況を確認することもできます。表 21.9 では、エラー発生状況を表すビットに CMDLK ビットを使用しています。"書き込み/消去の処理中"、"書き込み/中断処理の処理中"、"ブランクチェックの処理中"には FASTAT0 レジスタの FRDY ビットが 0 になります。FRDY ビットが 0 の場合で P/E サスペンドコマンドが受け付け可能な状態は、FSTATR0 レジスタの SUSRDY ビットが 1 の場合のみです。

表 21.9 では表を簡素化するために ERSSPD ビット、PRGSPD ビット、FRDY ビットの値を"0/1"と表記しています。ERSSPD ビットは、消去の中断処理中の場合には 1、書き込み処理の中断処理中の場合には 0 になります。PRGSPD ビットは書き込みの中断処理中には 1、消去の中断処理中には 0 になります。コマンドロック状態の FRDY ビットの値は、コマンドロックに遷移する前の FRDY ビットの値が保持されます。

表 21.9 FCU のモード/状態と受け付け可能なコマンドの関係

項 目	P/E ノーマルモード			ステータスリードモード									ロックビットリードモード		
	書き込みサスペンド中	消去サスペンド中	その他の状態	書き込み/消去の処理中	消去サスペンド中の書き込み処理中	書き込み/消去の中断処理中	ブランクチェック処理中	書き込みサスペンド中	消去サスペンド中	コマンドロック状態 (RDY=0)	コマンドロック状態 (RDY=1)	その他の状態	書き込みサスペンド中	消去サスペンド中	その他の状態
FSTATR0 レジスタの FRDY ビット	1	1	1	0	0	0	0	1	1	0	1	1	1	1	1
FSTATR0 レジスタの SUSRDY ビット	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
FSTATR0 レジスタの ERSSPD ビット	0	1	0	0	1	0/1	0/1	0	1	0/1	0/1	0	0	1	0
FSTATR0 レジスタの PRGSPD ビット	1	0	0	0	0	0/1	0/1	1	0	0/1	0/1	0	1	0	0
FASTAT レジスタの CMDLK ビット	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
ノーマルモード移行	○	○	○	×	×	×	×	○	○	×	×	○	○	○	○
ステータスリードモード移行	○	○	○	×	×	×	×	○	○	×	×	○	○	○	○
ロックビットリードモード移行 (ロックビットリード1)	○	○	○	×	×	×	×	○	○	×	×	○	○	○	○
フラッシュロック通知	×	×	○	×	×	×	×	×	×	×	×	○	×	×	○
プログラム	×	△	○	×	×	×	×	×	△	×	×	○	×	△	○
ブロックイレーズ	×	×	○	×	×	×	×	×	×	×	×	○	×	×	○
P/E サスペンド	×	×	×	○	×	×	×	×	×	×	×	×	×	×	×
P/E レジューム	○	○	×	×	×	×	×	○	○	×	×	×	○	○	×
ステータスレジスタクリア	○	○	○	×	×	×	×	○	○	×	○	○	○	○	○
ブランクチェック	○	○	○	×	×	×	×	○	○	×	×	○	○	○	○

【記号説明】

○ : 受け付け可能

△ : 消去中断したブロック以外への書き込みのみ受け付け可能。

× : 受け付け不可能

21.6.3 FCU コマンド使用方法

プログラムコマンドとブロックイレーズコマンドを使用して EEPROM を書き込み/消去する方法と、ブランクチェックコマンドを使用して EEPROM の消去状態を確認する方法を示します。FCU RAM へのファームウェア転送方法やその他の FCU コマンド使用方法については、「20.6.3 FCU コマンド使用方法」を参照してください。

FCU がコマンド処理中に FSTATR1 レジスタの FCUERR ビットが 1、または FRDTCT ビット、FRCRCT ビットが 1 にセットされてコマンドロック状態に遷移した場合には、FSTATR0 レジスタの FRDY ビットは 0 を保持します。コマンドロック状態では FCU の処理が停止するため、FRDY ビットが 0 から 1 にセットされることはありません。書き込み/消去時間やサスペンド遅延時間（「第 25 章 電気的特性」を参照）よりも長期間にわたって FRDY が 0 に保持される場合は、コマンドロック状態で FCU の処理が停止する等の異常動作が発生している可能性があるため、FCU リセットによって FCU を初期化してください。FCU のコマンド処理が完了して FRDY が 1 にセットされた場合には、FCUERR ビットと FRDTCT ビット、FRCRCT ビットは必ず 0 の状態です。このため、コマンド処理完了後のエラー発生状況は、ILGLERR ビット、ERSERR ビット、PRGERR ビットで確認可能です。

(1) 周辺クロック通知コマンドの使用方法

EEPROM へのデータ書き込み/消去前に使用しているフラッシュクロックの設定を行います。詳細はフラッシュメモリの「20.6.3 FCU コマンド使用方法」を参照してください。FENTRYR.FENTRYD ビットを 1 にセットして、アドレスはデータフラッシュ領域内のアドレスをさすようにみなして設定を行ってください。

(2) 書き込み方法

EEPROM へのデータ書き込みには、プログラムコマンドを使用します。プログラムコマンドの第 1 サイクルでは H'E8 を、第 2 サイクルでは書き込みワード数 (N) * を EEPROM 領域のアドレスにバイト書き込みします。コマンドの第 3~N+2 サイクルでは、ワードサイズで周辺バスアクセスを実行します。第 3 サイクルのアクセスでは、EEPROM 書き込み先頭アドレスに対して書き込みデータを書き込んでください。先頭アドレスは、8 バイト書き込みの場合には 8 バイト境界、128 バイト書き込みの場合には 128 バイト境界にアラインしてください。EEPROM 書き込み/消去用のアドレスに対して N-1 回のワード書き込みを実行後、第 N+3 サイクルで EEPROM 領域のアドレスに対して H'D0 をバイト書き込みすると FCU が EEPROM の書き込み処理を開始します。書き込みの完了は、FSTATR0 レジスタの FRDY ビットで確認可能です。

第 3 サイクル~第 N+2 サイクルでアクセスする領域に書き込み不要なアドレスが含まれる場合は、該当アドレスに対する書き込みデータを H'FFFF にしてください。EEPWE0、EEPWE1 レジスタによる書き込み/消去プロテクトを無効化して書き込みを実施したい場合には、書き込み対象ブロック用の書き込み/消去許可ビットを 1 に設定してから書き込みを行ってください。

図 21.6 に、EEPROM の書き込み方法を示します。

【注】 * 8 バイト書き込みの場合は N=H'04、128 バイト書き込みの場合は N=H'40 です。

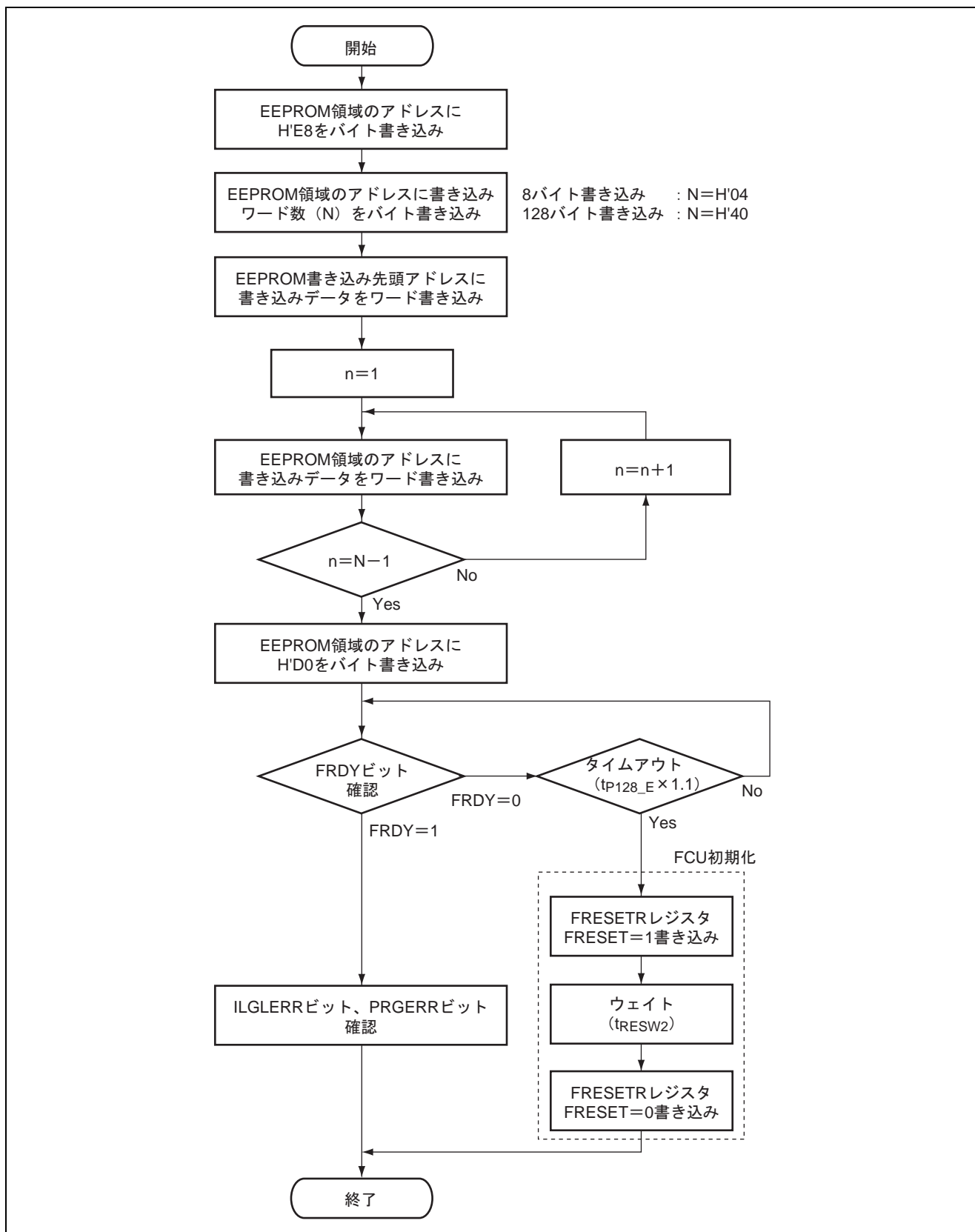


図 21.6 EEPROM の書き込み方法

(3) 消去方法

EEPROM の消去には、ブロックイレーズコマンドを使用します。ブロックイレーズコマンドを使用した消去方法は、フラッシュメモリの消去方法と同様です（「第 20 章 フラッシュメモリ」を参照）。EEPROM には、EEPWE0、EEPWE1 レジスタによる書き込み／消去プロテクト機能があることに注意してください。EEPWE0、EEPWE1 によるプロテクトを無効化して消去を実施したい場合には、消去対象ブロック用の書き込み／消去許可ビットを 1 に設定してから消去を行ってください。

(4) 消去状態の確認方法

CPU で消去状態の EEPROM を読み出すと不定値が読み出されますので、消去状態の確認にはブランクチェックコマンドを使用する必要があります。ブランクチェックコマンドを使用する場合には、事前に FMODR レジスタの FRDMD ビットを 1 に設定してブランクチェックコマンドが使用可能な状態にし、EEPBCCNT レジスタにチェック対象領域のサイズとアドレスを設定してください。EEPBCCNT レジスタの BCSIZE ビットが 1 の場合には、ブランクチェックコマンドの第 2 サイクルで指定した消去ブロック全体（2K バイト）のブランクチェックを実行可能です。BCSIZE ビットが 0 の場合には、ブランクチェックコマンドの第 2 サイクルで指定した消去ブロックの先頭アドレスと EEPBCCNT レジスタの値を加算したアドレスから 8 バイト分の領域のブランクチェックを実行可能です。ブランクチェックコマンドの第 1 サイクルでは、H'71 を EEPROM 領域のアドレスにバイト書き込みします。コマンドの第 2 サイクルでブランクチェック対象領域を含む消去ブロック内の任意アドレスに H'D0 をバイト書き込みすると、FCU が EEPROM のブランクチェック処理を開始します。ブランクチェックの完了は、FSTATR0 レジスタの FRDY ビットで確認可能です。ブランクチェックの完了後に EEPBCSTAT レジスタの BCST ビットの値を確認すると、チェック対象領域が消去状態であるか 0 データか 1 データを書き込んだ状態であるかを確認することができます。

図 21.7 に、EEPROM のブランクチェック方法を示します。

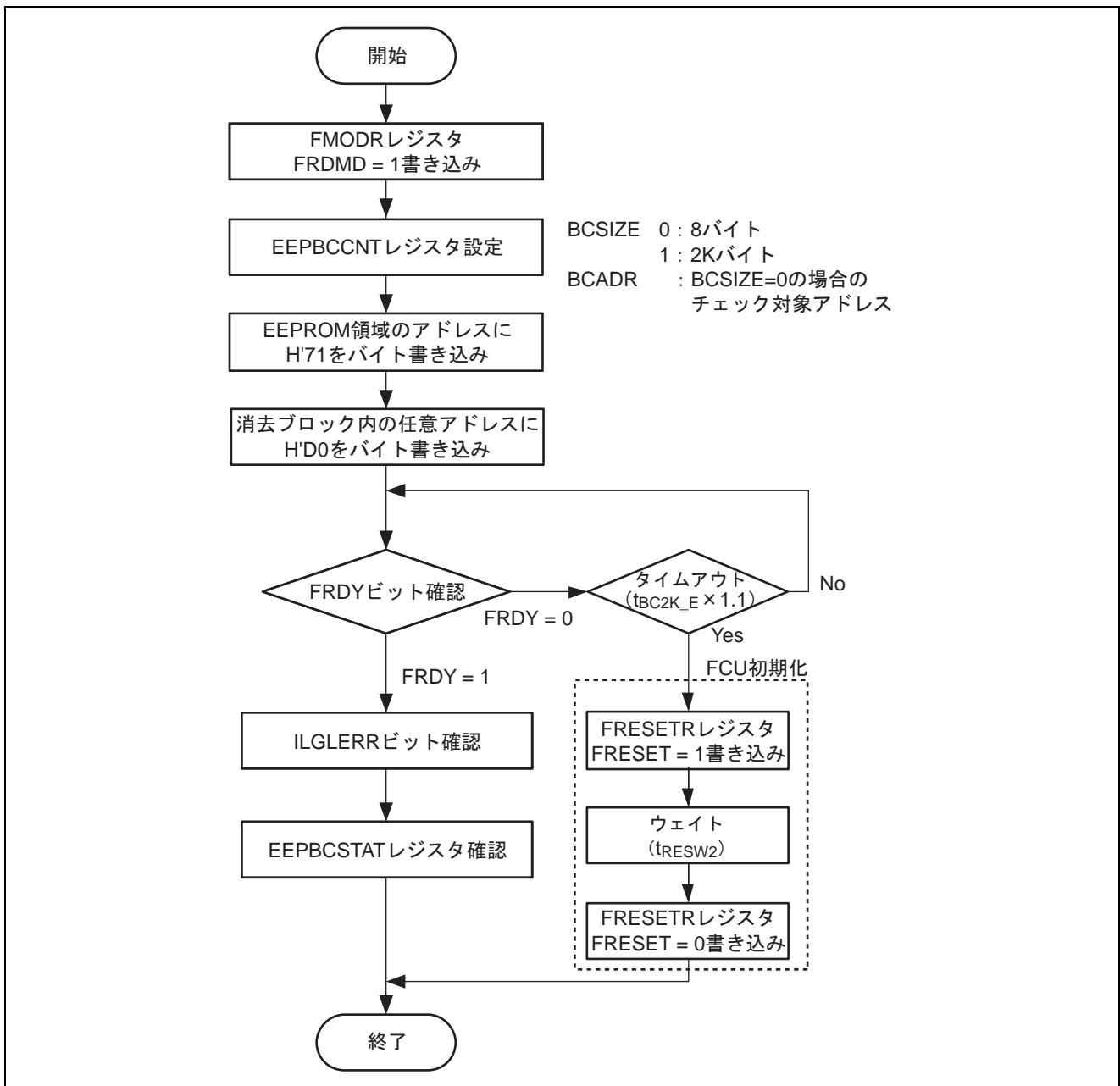


図 21.7 EEPROM のブランクチェック方法

21.7 プロテクト

EEPROM に対する書き込み／消去のプロテクトには、ハードウェアプロテクト、ソフトウェアプロテクト、エラープロテクトの3種類があります。

21.7.1 ハードウェアプロテクト

ハードウェアプロテクトは、モード端子の設定によって EEPROM に対する書き込み／消去が禁止された状態です。内蔵 ROM が無効な設定の場合には、EEPROM の書き込み／消去／読み出しが禁止された状態になります(本 LSI には内蔵 ROM 無効モードはありません)。本 LSI の動作モードと端子設定の関係は「第3章 MCU 動作モード」を参照してください。

21.7.2 ソフトウェアプロテクト

ソフトウェアプロテクトは、制御レジスタ設定によって EEPROM に対する書き込み／消去が禁止された状態です。ソフトウェアプロテクトに違反して、EEPROM に対する書き込み／消去系コマンドを発行した場合には、FCU がエラーを検出してコマンドロック状態になります。

(1) FENTRYR によるプロテクト

FENTRYR レジスタの FENTRYD ビットが 0 の場合には、EEPROM に対する FCU コマンドが受け付けられないため、EEPROM の書き込み／消去は禁止状態になります。FENTRYD ビットが 0 の状態で EEPROM に対する FCU コマンドを発行すると、FCU はイリーガルコマンドエラーを検出してコマンドロック状態になります(「21.7.3 エラープロテクト」を参照)。

(2) EEPWE0、EEPWE1 レジスタによるプロテクト

EEPWE0、EEPWE1 レジスタの DBWE_i (i=00~15) ビットが 0 の場合には、データマットの DB_i ブロックの書き込み／消去が禁止状態になります。DBWE_i ビットが 0 の状態で DB_i ブロックに対する書き込み／消去を実行すると、FCU は書き込み／消去プロテクト違反を検出してコマンドロック状態になります(「21.7.3 エラープロテクト」を参照)。

21.7.3 エラープロテクト

エラープロテクトは、FCU コマンドの誤発行/禁止アクセスの発生/FCU の誤動作を検出して FCU コマンドの受け付けを禁止する状態 (コマンドロック状態) です。FCU をコマンドロック状態にすることにより、EEPROM の書き込み/消去が禁止されます。コマンドロック状態を解除するためには、FASTAT レジスタが H'10 の状態でステータスレジスタクリアコマンドを発行する必要があります。

FAEINT レジスタの CMDLKIE ビットが 1 の場合には、FCU がコマンドロック状態 (FASTAT レジスタの CMDLK ビットが 1) になるとフラッシュインタフェースエラー (FIFE) 割り込みが発生します。また、FAEINT レジスタの EEPROM 関連の割り込みイネーブルビット (EEPAAIE/EEPRPEIE/EEPWPEIE ビット) が 1 の場合には、FASTAT レジスタの対応するビット (EEPAAE/EEPRPE/EEPWPE ビット) が 1 になると FIFE 割り込みが発生します。

表 21.10 に EEPROM 専用のエラープロテクト内容とエラー検出後のステータスビット値 (FSTATR0 レジスタの ILGLERR/ERSERR/PRGERR ビット、FASTAT レジスタの EEPAAE/EEPRPE/EEPWPE ビット) の関係を示します。フラッシュメモリ/EEPROM 共通のエラープロテクト内容 (FENTRYR 設定エラー、不正コマンドの大半、消去エラー、書き込みエラー、FCU エラー、FCU RAM ECC エラー) については、「21.7.3 エラープロテクト」を参照してください。

書き込み/消去処理中にサスペンド以外のコマンドが発行されてコマンドロック状態に遷移した場合には、FCU は書き込み/消去処理を継続します。この状態で P/E サスペンドコマンドを発行して書き込み/消去を中断することはできません。コマンドロック状態でコマンドが発行された場合には、ILGLERR ビット値は 1 になり、その他のビットの値は以前のエラー検出時に設定された値を保持します。

表 21.10 エラープロテクト一覧

分類	内 容	ILGLERR	ERSERR	PRGERR	EEPAE	EEPRPE	EEPWPE
不正コマンド	プログラムコマンドの 2 サイクル目で H'04、H'40 以外を指定	1	0	0	0	0	0
	FENTRYR レジスタの FENTRYD ビットが 1 の状態で、EEPROM 領域に対してロックビットプログラムコマンドを発行	1	0	0	0	0	0
EEPROM アクセス違反	FENTRYR レジスタの FENTRYD ビットが 1、かつ EEPROM P/E ノーマルモードで、EEPROM 領域に対してリードアクセスを発行	1	0	0	1	0	0
	FENTRYD ビットが 0 の状態で、EEPROM 領域に対してライトアクセスを発行	1	0	0	1	0	0
	FENTRYR レジスタの FENTRY0 ビットが 1 の状態で、EEPROM 領域に対するアクセスを発行	1	0	0	1	0	0
EEPROM リードプロテクト違反	EEPRE0、EEPRE1 レジスタで読み出し禁止に設定した EEPROM 領域に対してリードアクセスを発行	1	0	0	0	1	0
EEPROM ライトプロテクト違反	EEPWE0、EEPWE1 レジスタで書き込み/消去禁止に設定した EEPROM 領域に対して、プログラム/ブロックイレースコマンドを発行	1	0	0	0	0	1

21.8 製品情報マット

製品情報マットには、デバイス名/デバイスリビジョン/組み込みプログラムリビジョンが ASCII コードで書き込まれています。組み込みプログラムとは、ブートモード/ライターモード時の起動マット（「20.4 フラッシュメモリ関連モード概要」を参照）に格納されているプログラムです。表 21.11、表 21.12 に製品情報マットに格納される情報の格納アドレスとデータ例を示します。製品情報マット領域（H'E0,0000～H'E0,007F）のうち、表 21.11、表 21.12 にないアドレスは予約領域です。予約領域の読み出しデータは不定です。

表 21.11 製品情報マット格納データ (H8SX/1727S の場合)

情報	アドレス	データ例
デバイス名	H'E0,0000～H'E0,0008	H'523546363137323753=R5F61727S
デバイスリビジョン	H'E0,0010～H'E0,0011	H'3031=01
組み込みプログラムリビジョン	H'E0,0020～H'E0,0022	H'313030=100 (1.00)

表 21.12 製品情報マット格納データ (H8SX/1725S の場合)

情報	アドレス	データ例
デバイス名	H'E0,0000～H'E0,0008	H'523546363137323553=R5F61725S
デバイスリビジョン	H'E0,0010～H'E0,0011	H'3031=01
組み込みプログラムリビジョン	H'E0,0020～H'E0,0022	H'313030=100 (1.00)

21.9 使用上の注意事項

(1) リセット起動直後のデータマットプロテクト状態

EEPWE0,EEPWE1,EEPWE0,EEPWE1 レジスタの初期値が H'0000 であるため、リセット起動直後のデータマットの読み出し/書き込み/消去は禁止状態です。データマットの読み出しが必要な場合には EEPWE0,EEPWE1 レジスタを設定してからデータマットにアクセスしてください。また、データマットの書き込み/消去が必要な場合には、EEPWE0、EEPWE1 を設定してから書き込み/消去用の FCU コマンドを発行してください。レジスタを設定せずに読み出し/書き込み/消去を実行しようとする、FCU がエラーを検出してコマンドロック状態になります。

(2) 割り込み無視状態

以下の状態では、NMI やマスカブル割り込みが発生しても無視されます。

- ブートモード動作中

(3) 書き込み/消去サスペンド対象領域

書き込み/消去サスペンド中の領域の格納データは不定です。不定データの読み出しが原因で発生する誤動作を回避するために、書き込み/消去サスペンド対象領域のデータ読み出しが発生しないように注意してください。

(4) 従来の F-ZTAT H8SX マイコンとの書き込み/消去プログラムの互換性

従来の F-ZTAT H8SX マイコンで使用していたフラッシュメモリの書き込み/消去プログラムは、本 LSI では動作しません。

(5) 書き込み/消去中のリセット

書き込み/消去処理中に FRESETR レジスタの FRESET ビットをセットして FCU をリセットする場合には、リセット状態を t_{RESW2} 保持してください。書き込み/消去中の EEPROM には高電圧が印加されているため、メモリに印加された電圧の降下に必要な期間を確保するために FCU のリセット状態を保持する必要があります。FCU をリセットしている期間は EEPROM の読み出しは行わないでください。

書き込み/消去処理中に \overline{RES} 端子のアサートによってリセットが発生させた場合には、リセット状態を t_{RESW2} 保持してください。リセットでは、メモリに印加された電圧の降下に必要な期間だけでなく、EEPROM 用電源の初期化や EEPROM の内部回路の初期化に必要な期間を確保する必要があります。

書き込み/消去処理中には WDT のカウンタのオーバフローによる内部リセットは発生しないようにしてください。WDT によるリセットでは、メモリ電圧の降下/EEPROM 用電源の初期化/EEPROM 内部の初期化に必要な期間を確保することができません。

書き込み/消去処理中に \overline{RES} 端子のアサートによるリセット、FRESETR レジスタの FRESET ビットをセットしての FCU リセットを実行すると、書き込み/消去対象の全データは不定となります。

(6) 追加書き込み禁止

同一領域に2回以上の書き込みを行うことはできません。書き込み済みの領域を書き換えたい場合には、必ず該当領域を消去してください。

(7) 製品情報マットへの書き込み

製品情報マットは読み出し専用で、書き込み/消去はできません。EEPMAT レジスタの EEPSEL ビットを1にセットして書き込み/消去を行った場合、データマットへの書き込み/消去が実行されます。また、EEPROM アクセス違反などのエラーも発生しません。製品情報マットへの書き込み/消去は行わないようにしてください。

(8) 書き込み/消去サスペンドによる中断

書き込み/消去サスペンドコマンドによって書き込み/消去処理を中断した場合は、必ずレジュームコマンドにより動作を完了させてください。

22. クロック発振器

本 LSI は、クロック発振器 (CPG : Clock Pulse Generator) を内蔵しており、システムクロック ($I\phi$)、周辺モジュールクロック ($P\phi$)、外部バスクロック ($B\phi$)、FLASH クロック ($F\phi$)、A/D クロック ($A\phi$)、RSPI クロック ($R\phi$) を生成します。

クロック発振器は、発振器、外部発振停止検出回路、内部発振回路、メインクロック分周器、PLL (Phase Locked Loop) 回路、セクタ回路から構成されます。クロック発振器のブロック図を図 22.1 に示します。

クロック発振器内部の PLL 回路とメインクロック分周器によりクロック周波数を変更できます。クロック周波数の変更は、システムクロックコントロールレジスタ 0 (SCKCR0)、システムクロックコントロールレジスタ 1 (SCKCR1) の設定によりソフトウェアで行います。

クロックには CPU、バスマスタに供給されるシステムクロック、周辺モジュールに供給される周辺モジュールクロック、外部バスに供給される外部バスクロック、FLASH メモリ制御部供給される FLASH クロック、A/D に供給される A/D クロック、および RSPI に供給される RSPI クロックがあります。

システムクロック、周辺モジュールクロック、外部バスクロック、FLASH クロック、A/D クロック、および RSPI クロックは、それぞれ独立に周波数を設定することができますが、周辺モジュールクロック、外部バスクロック、FLASH クロック、A/D クロック、RSPI クロックは、システムクロック以下の周波数でしか動作できません。

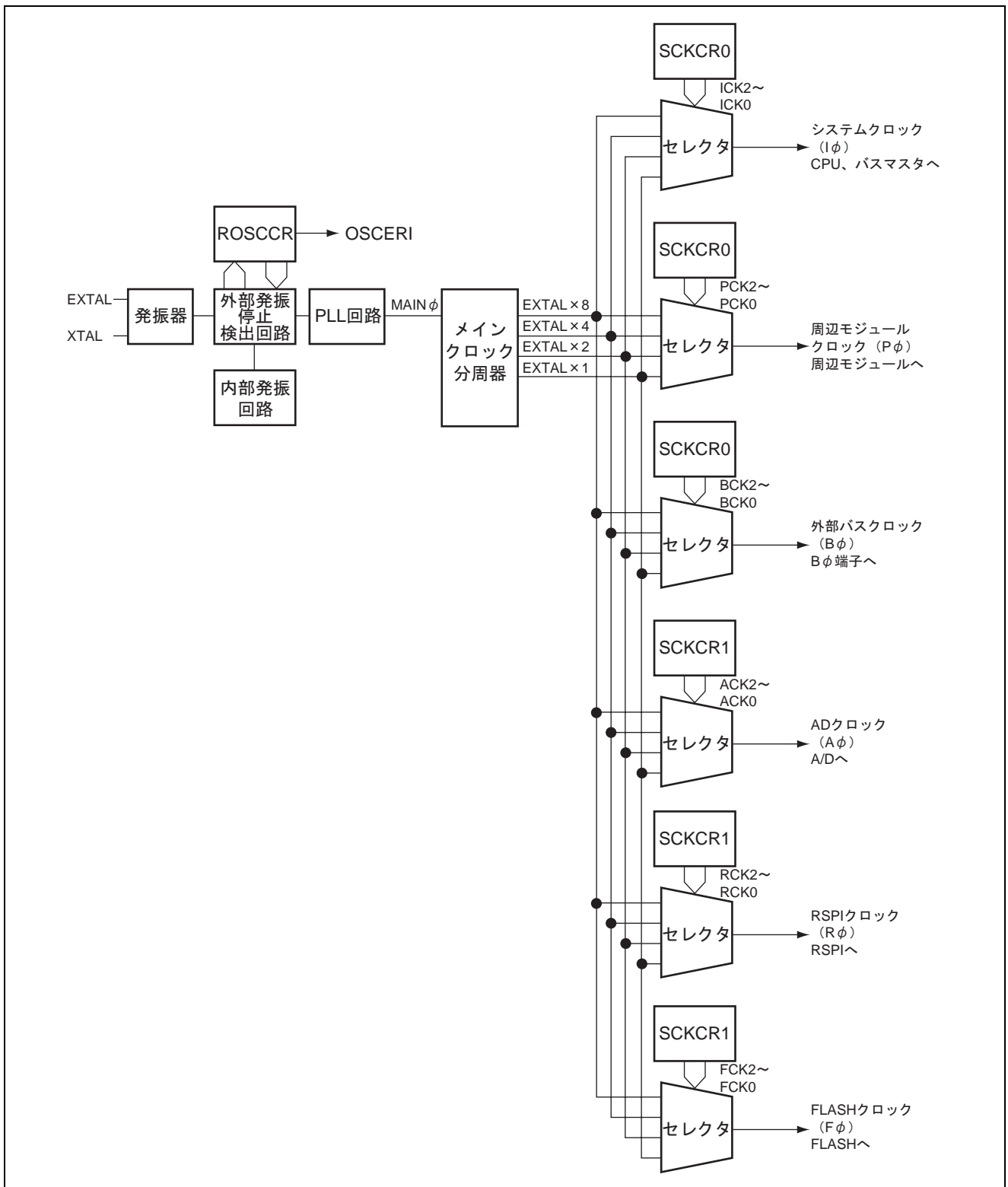


図 22.1 クロック発振器のブロック図

表 22.1 クロック発振器の選択 (推奨設定)

EXTAL 入力クロック 周波数	Iφ	Pφ	Bφ	Fφ	Aφ	Rφ
8MHz	64MHz (x8)	32MHz (x4)	16MHz (x2)	32MHz (x4)	32MHz (x4)	32MHz (x4)
		16MHz (x2)	16MHz (x2)	16MHz (x2)	16MHz (x2)	16MHz (x2)
10MHz	80MHz (x8)	40MHz (x4)	20MHz (x2)	40MHz (x4)	40MHz (x4)	40MHz (x4)
		20MHz (x2)	20MHz (x2)	20MHz (x2)	20MHz (x2)	20MHz (x2)
	40MHz (x4)	40MHz (x4)	20MHz (x2)	40MHz (x4)	40MHz (x4)	40MHz (x4)
		20MHz (x2)	20MHz (x2)	20MHz (x2)	20MHz (x2)	20MHz (x2)

22.1 レジスタの説明

クロック発振器には以下のレジスタがあります。

- システムクロックコントロールレジスタ0 (SCKCR0)
- システムクロックコントロールレジスタ1 (SCKCR1)
- リカバリーオシレータコントロールレジスタ (ROSCCR)

22.1.1 システムクロックコントロールレジスタ 0 (SCKCR0)

SCKCR0 は B ϕ 出力制御、B ϕ 出力選択とシステムクロック、周辺モジュールクロックおよび外部バスクロックの周波数の制御を行います。

ビット	15	14	13	12	11	10	9	8
ビット名	PSTOP1	—	POSEL1	—	—	ICK2	ICK1	ICK0
初期値 :	0	0	0	0	0	0	1	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ビット	7	6	5	4	3	2	1	0
ビット名	—	PCK2	PCK1	PCK0	—	BCK2	BCK1	BCK0
初期値 :	0	0	1	0	0	0	1	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ビット	ビット名	初期値	R/W	説明
15	PSTOP1	0	R/W	B ϕ 出力セレクトイネーブル PA7 からの ϕ 出力を制御します。 通常動作状態 0 : B ϕ 出力 1 : High レベル固定 ソフトウェアスタンバイモード x : High レベル固定 ハードウェアスタンバイモード* x : ハイインピーダンス 【注】* 本 LSI には、ハードウェアスタンバイモードはありません。
14	—	0	R/W	リザーブビット リードすると常に 0 が読み出されます。ライトする値は常に 0 にしてください。
13	POSEL1	0	R/W	B ϕ 出力セレクト 1 PA7 の B ϕ 出力を制御します。 0 : 外部バスクロック (B ϕ) 1 : 設定禁止
12, 11	—	すべて 0	R/W	リザーブビット リードすると常に 0 が読み出されます。ライトする値は常に 0 にしてください。

ビット	ビット名	初期値	R/W	説明
10	ICK2	0	R/W	システムクロック (I ϕ) セレクト CPU、DMAC、DTC モジュールとシステムクロックの周波数を選択します。入力クロックに対する倍率を示しています。 000 : $\times 8$ 001 : $\times 4$ 010 : $\times 2$ 011 : $\times 1$ 1xx : 設定禁止 周辺モジュールクロック、外部クロック、A/D クロックおよび RSPI クロックより低い周波数の設定を行うと、周辺モジュールクロックはシステムクロックと同じ周波数に変わります。
9	ICK1	1	R/W	
8	ICK0	0	R/W	
7	—	0	R/W	リザーブビット リードすると常に 0 が読み出されます。ライトする値は常に 0 にしてください。
6	PCK2	0	R/W	周辺モジュールクロック (P ϕ) セレクト 周辺モジュールクロックの周波数を選択します。入力クロックに対する倍率を示しています。 000 : $\times 8$ 001 : $\times 4$ 010 : $\times 2$ 011 : $\times 1$ 1xx : 設定禁止 システムクロックより高い周波数の設定を行わないでください。 レジスタのセットは行えますが、周波数はシステムクロックと同一になります。
5	PCK1	1	R/W	
4	PCK0	0	R/W	
3	—	0	R/W	リザーブビット リードすると常に 0 が読み出されます。ライトする値は常に 0 にしてください。
2	BCK2	0	R/W	外部バスクロック (B ϕ) セレクト 外部バスクロックの周波数を選択します。入力クロックに対する倍率を示しています。 000 : $\times 8$ 001 : $\times 4$ 010 : $\times 2$ 011 : $\times 1$ 1xx : 設定禁止 システムクロックより高い周波数の設定を行わないでください。 レジスタのセットは行えますが、周波数はシステムクロックと同一になります。
1	BCK1	1	R/W	
0	BCK0	0	R/W	

【注】 x : Don't care

22.1.2 システムクロックコントロールレジスタ 1 (SCKCR1)

SCKCR1 は FLASH クロック、A/D クロックおよび RSPI クロックの周波数の制御を行います。

ビット	15	14	13	12	11	10	9	8
ビット名	—	—	—	—	—	FCK2	FCK1	FCK0
初期値 :	0	0	0	0	0	0	1	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ビット	7	6	5	4	3	2	1	0
ビット名	—	ACK2	ACK1	ACK0	—	RCK2	RCK1	RCK0
初期値 :	0	0	1	0	0	0	1	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ビット	ビット名	初期値	R/W	説明
15~11	—	すべて 0	R/W	リザーブビット リードすると常に 0 が読み出されます。ライトする値は常に 0 にしてください。
10	FCK2	0	R/W	FLASH クロック (F ϕ) セレクト FLASH クロックの周波数を選択します。入力クロックに対する倍率を示しています。 000 : $\times 8$ 001 : $\times 4$ 010 : $\times 2$ 011 : $\times 1$ 1xx : 設定禁止 システムクロックより高い周波数の設定を行わないでください。 レジスタのセットは行えますが、周波数はシステムクロックと同一になります。
9	FCK1	1	R/W	
8	FCK0	0	R/W	
7	—	0	R/W	リザーブビット リードすると常に 0 が読み出されます。ライトする値は常に 0 にしてください。
6	ACK2	0	R/W	A/D クロック (A ϕ) セレクト A/D クロックの周波数を選択します。入力クロックに対する倍率を示しています。 000 : $\times 8$ 001 : $\times 4$ 010 : $\times 2$ 011 : $\times 1$ 1xx : 設定禁止 システムクロックより高い周波数の設定を行わないでください。 レジスタのセットは行えますが、周波数はシステムクロックと同一になります。
5	ACK1	1	R/W	
4	ACK0	0	R/W	

ビット	ビット名	初期値	R/W	説明
3	—	0	R/W	リザーブビット リードすると常に0が読み出されます。ライトする値は常に0にしてください。
2	RCK2	0	R/W	RSPI クロック (Rφ) セレクト RSPI クロックの周波数を選択します。入力クロックに対する倍率を示しています。 000 : ×8 001 : ×4 010 : ×2 011 : ×1 1xx : 設定禁止 システムクロックより高い周波数の設定を行わないでください。 レジスタのセットは行えますが、周波数はシステムクロックと同一になります。
1	RCK1	1	R/W	
0	RCK0	0	R/W	

【注】 x : Don't care

22.1.3 リカバリーオシレータコントロールレジスタ (ROSCCR)

ROSCCR は外部発振停止検出機能の制御を行います。

ビット	15	14	13	12	11	10	9	8
ビット名	INOSCE	OSCERR	OSCIE	—	—	—	—	ERRTEST
初期値:	1	0	0	0	0	0	0	0
R/W:	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
ビット	7	6	5	4	3	2	1	0
ビット名	—	—	—	—	—	—	—	—
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

ビット	ビット名	初期値	R/W	説明
15	INOSCE	1	R/W	外部発振停止検出機能イネーブル 外部発振停止検出機能の動作を制御します。 0: 外部発振停止検出機能は無効 1: 外部発振停止検出機能は有効
14	OSCERR	0	R	外部発振の状態フラグ 外部発振の状態を示すステータスフラグです。 INOSCE=1 のとき 0: 外部発振は正常動作 1: 外部発振の 1 または 0 スタックを検出 INOSCE=0 のとき リードすると常に 0 が読み出されます。 このビットはリードのみ有効でライトは無効です。
13	OSCIE	0	R/W	外部発振停止検出割り込みイネーブル OSCERR フラグが 1 にセットされたとき、割り込み要求の許可または禁止を行います。 0: OSCERR による割り込みを禁止 1: OSCERR による割り込みを許可
12~9	—	すべて 0	R/W	リザーブビット リードすると常に 0 が読み出されます。ライトする値は常に 0 にしてください。
8	ERRTEST	0	R/W	発振停止検出テスト 発振停止検出フラグ(OSCERR)を強制的に 1 セットします。 0: OSCERR フラグは状態保持 1: OSCERR フラグを 1 セット

ビット	ビット名	初期値	R/W	説明
7~0	—	すべて0	R/W	リザーブビット リードすると常に0が読み出されます。ライトする値は常に0にしてください。

22.2 発振器

クロックを供給する方法には、水晶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。

22.2.1 水晶発振子を接続する方法

水晶発振子を接続する場合の接続例を図 22.2 に示します。ダンピング抵抗 R_d は、表 22.2 に示すものを使用してください。

水晶発振子を接続してクロックを供給する場合、接続する水晶発振子は、8~10MHz としてください。

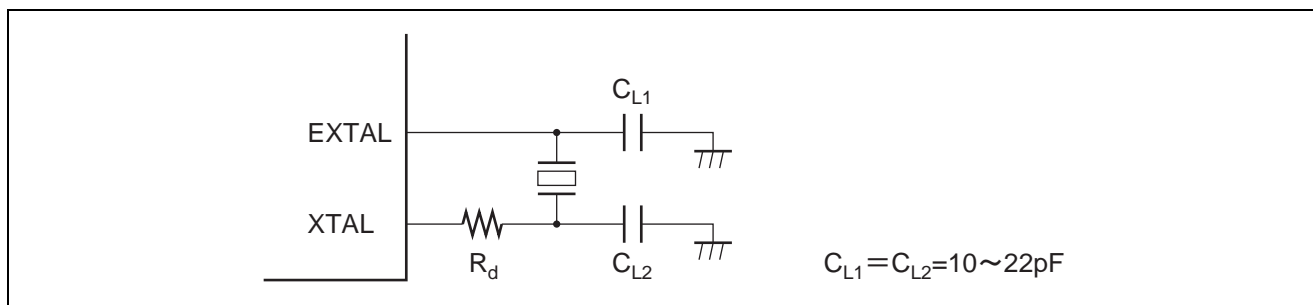


図 22.2 水晶発振子の接続例

表 22.2 ダンピング抵抗値

周波数 (MHz)	8	10
R_d (Ω)	200	0

水晶発振子の等価回路を図 22.3 に示します。水晶発振子は表 22.3 に示す特性のものを使用してください。

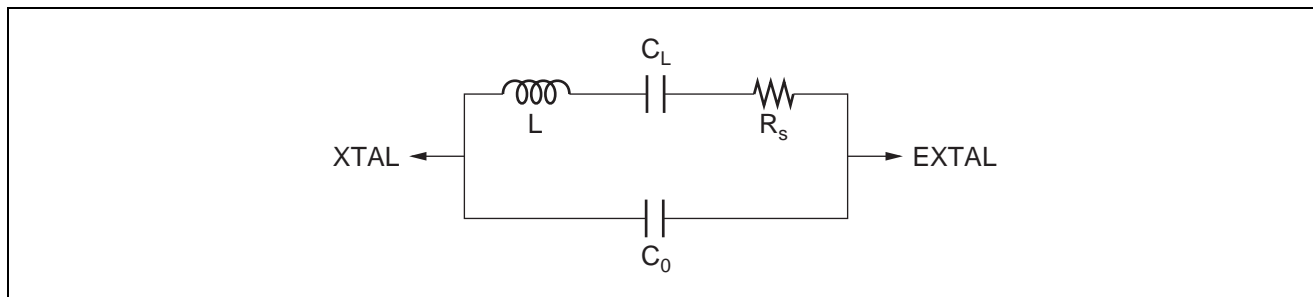


図 22.3 水晶発振子の等価回路

表 22.3 水晶発振子の特性

周波数 (MHz)	8	10
R_s max (Ω)	80	70
C_0 max (pF)	7	

22.2.2 外部クロックを入力する方法

外部クロック入力の接続例を図 22.4 に示します。XTAL 端子をオープン状態にする場合、寄生容量は 10pF 以下にしてください。XTAL 端子に逆相クロックを入力する場合、スタンバイモード時は外部クロックを High レベルにしてください。

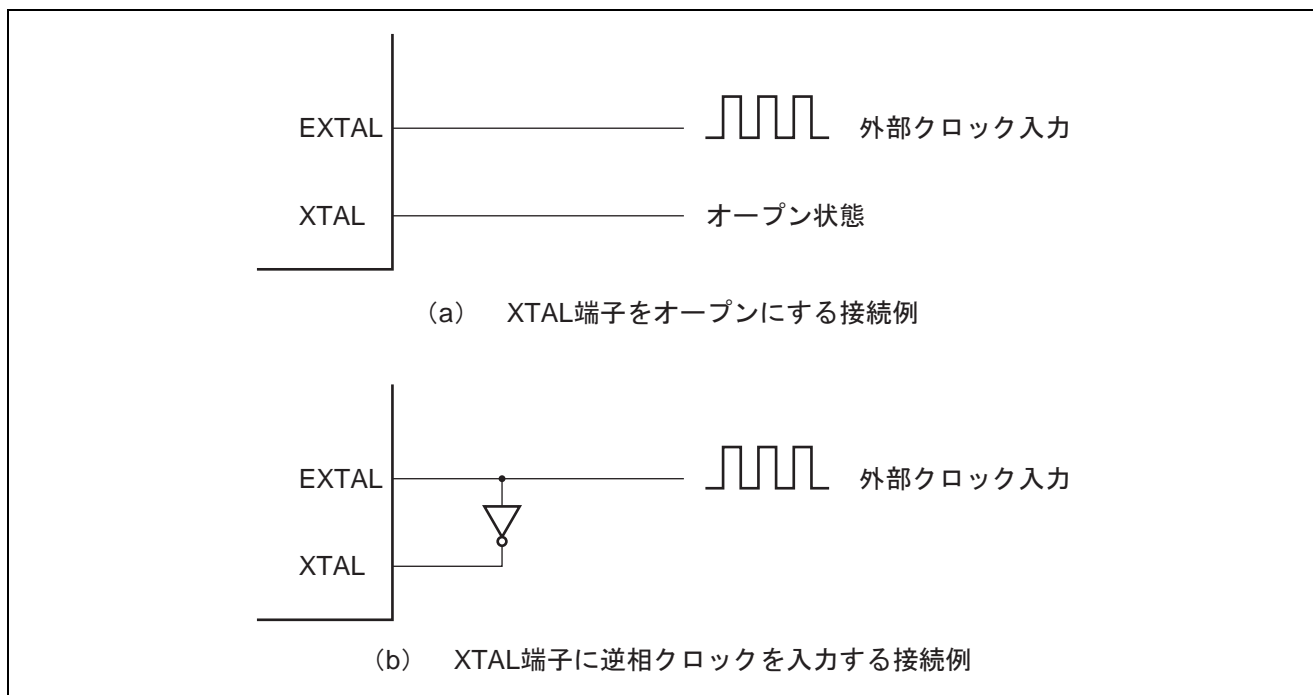


図 22.4 外部クロックの接続例

外部クロックの入力条件については、「25.3.1 クロックタイミング」を参照してください。入力する外部クロックは、8~10MHz としてください。

22.3 PLL 回路

PLL 回路は、発振器からの周波数を 8 倍に通倍する機能を持っています。周波数通倍率は固定です。このとき、内部クロックの立ち上がりエッジの相位は EXTAL 端子の立ち上がりエッジの相位に一致するように制御されます。

22.4 メインクロック分周器

メインクロック分周器は、PLL クロックを分周し、1/2、1/4、1/8 のクロックを生成します。ICK2~ICK0、PCK2~PCK0、BCK2~BCK0、FCK2~FCK0 ビット、ACK2~ACK0、RCK2~RCK0 ビット書き換え後に、変更後の周波数で本 LSI は動作します。

22.5 外部発振停止検出機能

22.5.1 概要

外部発振停止検出回路は、外部発振が何らかの理由で発振が停止のような異常動作をした場合には、ROSCCR の OSCERR ビットを 1 (初期値 0) にセットします。さらに、外部発振のクロックを内部発振回路のクロックに切り替えて LSI の動作を継続します。外部発振の異常動作を一度検出したら、次のリセットスタートまではその状態を保持します。

本 LSI は、リセットスタート、およびソフトウェアスタンバイ解除時には内部発振回路のクロックを使用し、バスマスタ、周辺モジュール、外部バスにクロックを供給します。発振安定時間内に外部発振のクロックが正常動作すれば、外部発振のクロックを使用しクロックを供給します。この外部発振停止検出機能を無効 (ROSCCR レジスタの INOSCE ビットを 0) にした場合には、外部発振の異常動作を検出しません。また、内部発振回路のクロックへの切り替えも行いません。

22.5.2 外部発振停止検出機能の設定方法

外部発振停止検出機能はリセットスタート直後から有効になります。この機能を無効にするには、リセット解除後、ROSCCR レジスタの INOSCE ビットを 0 (初期値 1) にクリアしてください。ROSCCR レジスタの INOSCE ビットは、リセット/ハードウェアスタンバイ*1 で 1 にセットされます。

外部発振停止検出機能が有効なとき、外部発振の発振周波数が異常動作により、ある一定期間クロックのエッジが入らない (表 25.12 を参照) 場合には、外部発振異常動作と判定し内部発振回路のクロック (表 22.4 を参照) に切り替えます。このときに LSI は ROSCCR レジスタの OSCERR ビット*2 を 1 (初期値 0) にセットします。内部発振回路のクロックに一度切り替わった場合は、その後、外部発振が正常に動作したとしても内部発振回路のクロックでの動作を継続します。外部発振を用いて動作させるためには、再度リセットスタートをしてください。

また、ROSCCR レジスタの OSCIE ビットを 1 (初期値 0) にセットすると外部発振の異常を検出した場合、外部発振停止検出割り込み (OSCERI) *2 を発生させることができます。外部発振停止検出割り込みを発生させた場合、割り込み処理中に LSI の異常処理を行ってください。外部発振停止検出割り込み処理からの復帰はリセットスタートを行ってください。

【注】 *1 本 LSI には、ハードウェアスタンバイモードはありません。

*2 ROSCCR レジスタの ERRTTEST ビットを 1 セットすることにより、外部発振の異常を検出しなくても、ソフトウェアで OSCERR を 1 セットできます。また、ROSCCR レジスタの OSCIE ビットを 1 (初期値 0) にセットすると外部発振停止検出割り込み (OSCERI) を発生させることができます。

表 22.4 内部発振回路使用時のシステムクロック (f_{ϕ}) の周波数

	min	typ	max
システムクロック (f_{ϕ}) 周波数 (8 通倍時)	$8 \times f_{\text{MAIN } \phi}$ (min)	$8 \times f_{\text{MAIN } \phi}$ (typ)	$8 \times f_{\text{MAIN } \phi}$ (max)
システムクロック (f_{ϕ}) 周波数 (4 通倍時)	$4 \times f_{\text{MAIN } \phi}$ (min)	$4 \times f_{\text{MAIN } \phi}$ (typ)	$4 \times f_{\text{MAIN } \phi}$ (max)
システムクロック (f_{ϕ}) 周波数 (2 通倍時)	$2 \times f_{\text{MAIN } \phi}$ (min)	$2 \times f_{\text{MAIN } \phi}$ (typ)	$2 \times f_{\text{MAIN } \phi}$ (max)
システムクロック (f_{ϕ}) 周波数 (1 通倍時)	$f_{\text{MAIN } \phi}$ (min)	$f_{\text{MAIN } \phi}$ (typ)	$f_{\text{MAIN } \phi}$ (max)

【注】 $f_{\text{MAIN } \phi}$ については、「25.6.3 内部発振周波数」を参照してください。

22.5.3 外部発振停止検出機能動作時の注意事項

外部発振が異常動作し内部発振を用いての動作時は、LSIの異常処理だけを行ってください。異常処理時、以下の動作は禁止です。行った場合の動作保証はいたしません。

- INOSCEビットのクリア
- 割り込み処理からの復帰
- ソフトウェアスタンバイモードへの遷移

22.6 使用上の注意事項

22.6.1 クロック発振器に関する使用上の注意事項

1. SCKCR0、SCKCR1の設定により各モジュールに供給される ϕ ($I\phi$: システムクロック、 $P\phi$: 周辺モジュールクロック、 $B\phi$: 外部バスクロック、 $F\phi$: FLASHクロック、 $A\phi$: A/Dクロック、 $R\phi$: RSPIクロック)の周波数が変わります。各クロックの周波数は表22.5の範囲内に収まるように選択してください。

表 22.5 各クロックの設定周波数の範囲

		設定周波数 [MHz]								
		↓	8	16	20	32	40	64	80	↑
クロック名	$I\phi$	設定禁止	min	—	—	—	—	—	max	設定禁止
	$P\phi$		min	—	—	—	max			
	$B\phi$		min	—	max					
	$F\phi$		min	—	—	—	max			
	$A\phi$		min	—	—	—	max			
	$R\phi$		min	—	—	—	max			

- 【注】
1. $I\phi$ より高い周波数の設定を行わないでください。
 2. $R\phi \geq P\phi \geq F\phi$ となるように設定してください。
 3. RCAN 使用時、 $P\phi \text{ min} = 16\text{MHz}$ としてください。
2. 周辺モジュール (DMAC、DTC、A/D、RSPIを除く)は、すべて $P\phi$ を基準に動作します。このため、周波数変更の前後でタイマやSCIなどの時間処理が変わりますので注意してください。
また、ソフトウェアスタンバイモード解除用の待機時間も周波数を変更することで変わります。詳細は「23.7.3 ソフトウェア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。
 3. システムクロック、周辺モジュールクロック、外部バスクロック、A/DクロックおよびRSPIクロックの間には、 $I\phi \geq P\phi$ 、 $I\phi \geq B\phi$ 、 $I\phi \geq F\phi$ 、 $I\phi \geq A\phi$ 、 $I\phi \geq R\phi$ の関係が成り立っており、かつシステムクロックの設定が優先されます。そのため、 $P\phi$ 、 $B\phi$ 、 $F\phi$ 、 $A\phi$ 、 $R\phi$ が、PCK2~PCK0、BCK2~BCK0、FCK2~FCK0、ACK2~ACK0、RCK2~RCK0のレジスタ設定の周波数ではなく、ICK2~ICK0で設定した周波数になることがあります。
 4. 図22.6にクロック変更タイミングを示します。SCKCR0あるいはSCKCR1に値をライトした後、現在実行しているバスサイクル終了を待ちます。そのバスサイクル終了後、外部クロックに対して最大1サイクル後に各クロックの周波数が変更されます。

5. SCKCR0、SCKCR1の設定が $I\phi > P\phi$ 、 $I\phi > F\phi$ 、 $I\phi > A\phi$ 、 $I\phi > R\phi$ の場合は、周辺モジュールからの信号とシステムクロックの同期をとる必要があります。CPUの命令により周辺モジュールの割り込み要因フラグをクリアする場合は、要因フラグを0にクリアした後にリードしてください。
6. $I\phi \geq 40\text{MHz}$ の周波数時に、システムクロック、周辺モジュールクロック、外部バスクロック、A/DクロックおよびRSPIクロックの周波数を変更する場合、SCKCR0、SCKCR1の設定直後、NOP命令を8回数実行してください。

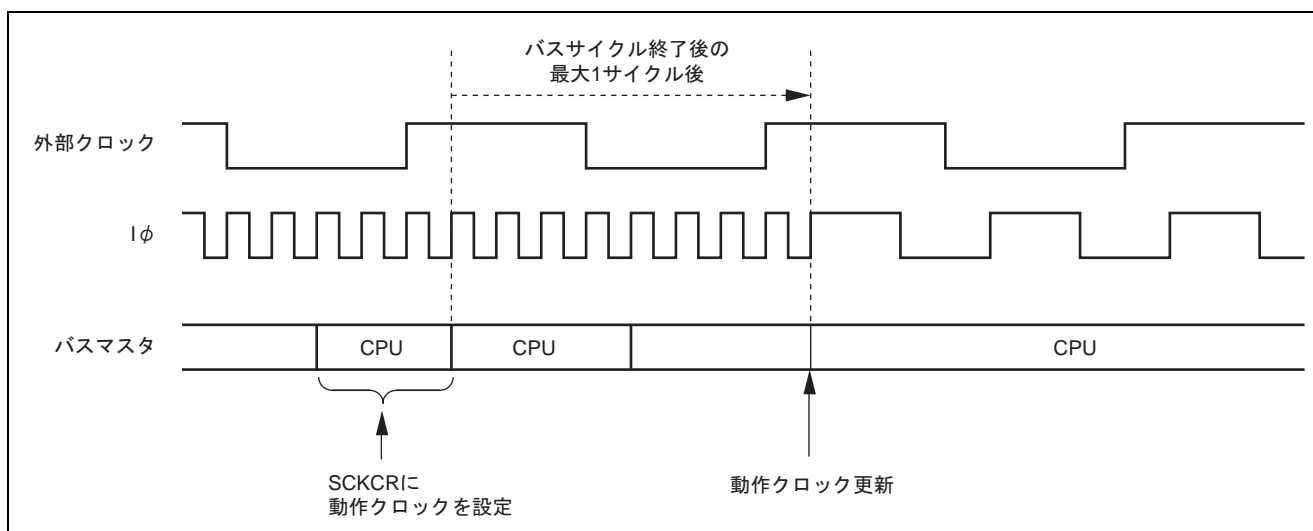


図 22.5 クロック変更タイミング

22.6.2 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、ユーザ側での十分な評価を実施してご使用願います。発振子の回路定数は発振子、実装回路の浮遊容量などにより異なるため、発振子メーカーと十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超えないようにしてください。

22.6.3 ボード設計上の注意

水晶発振子を使用する場合は、発振子および負荷容量はできるだけ XTAL、EXTAL 端子の近くに配置してください。図 22.6 に示すように発振回路の近くには信号線を通過させないでください。誘導により正しい発振ができなくなることがあります。

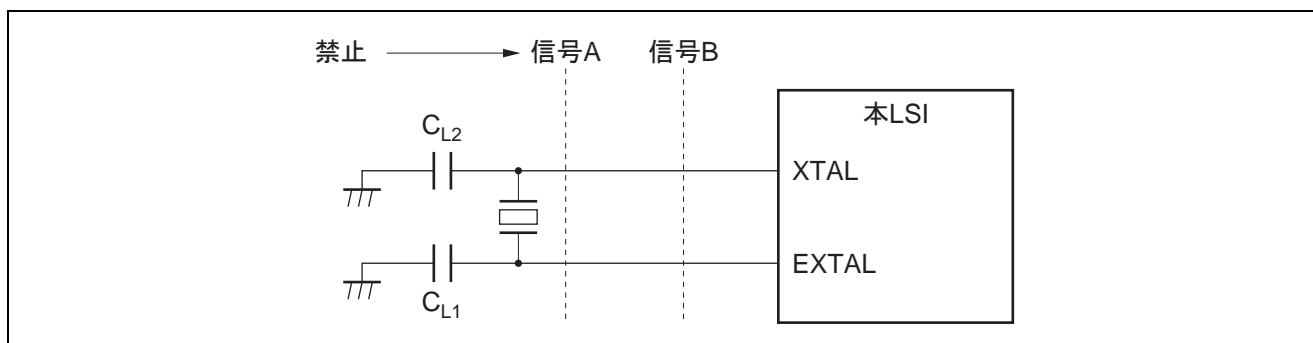


図 22.6 発振回路部のボード設計に関する注意事項

バイパスコンデンサの接続例を図 22.7 に示します。Vcc、Vss はボードの電源供給元から分離し、端子の近くにバイパスコンデンサ CB を必ず挿入してください。バイパスコンデンサの容量値はユーザのシステムボード設計において十分検討してください。

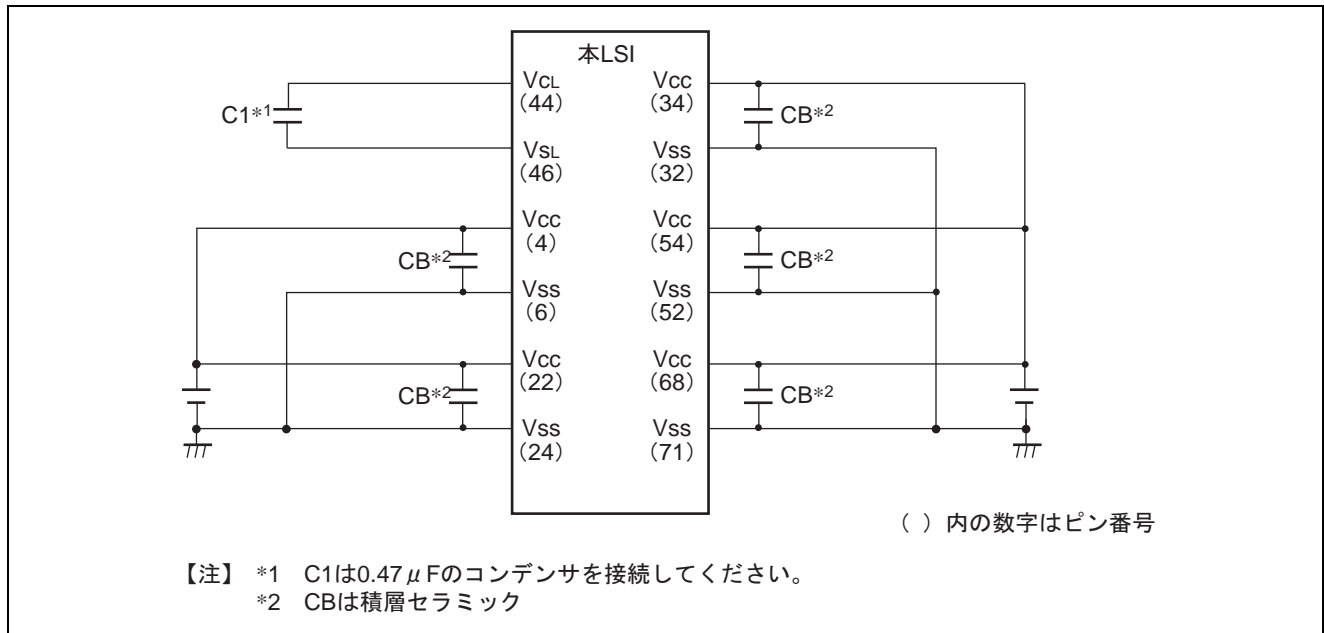


図 22.7 バイパスコンデンサの接続例

22.6.4 入力クロック周波数に関する注意

入力クロック周波数は PLL 回路内で 8 倍に逡倍されます。ノイズ低減のために、入力クロック周波数は、8 ~10MHz の間で低い周波数を推奨します。

23. 低消費電力

本 LSI には、消費電力低減機能としてマルチクロック機能、モジュールストップ機能、および低消費電力状態への遷移機能があります。

23.1 特長

- マルチクロック機能
システムクロック、周辺モジュールクロック、外部バスクロック、FLASH クロック、A/D クロック、RSPI クロックに対し独立に分周比を設定することが可能
- モジュールストップ機能
周辺モジュール毎に機能を停止し、低消費電力状態にすることが可能
- 低消費電力状態への遷移機能
CPU、周辺モジュール、発振器を停止する低消費電力状態にすることが可能
- 低消費電力モード：3種類
スリープモード
全モジュールクロックストップモード
ソフトウェアスタンバイモード

表 23.1 に、低消費電力状態への遷移する条件と CPU や周辺モジュールなどの状態、および各モードの解除方法を示します。

リセット後は、通常のプログラム動作で DMAC、DTC 以外のモジュールは停止状態です。

表 23.1 動作状態

動作状態	スリープ モード	全モジュールクロック ストップモード	ソフトウェア スタンバイモード
遷移条件	制御レジスタ+ 命令	制御レジスタ+ 命令	制御レジスタ+ 命令
解除方法	割り込み	割り込み* ¹	外部割り込み
発振器	動作	動作	停止
CPU	停止 (保持)	停止 (保持)	停止 (保持)
ウォッチドッグ タイマ	動作	動作	停止 (保持)
周辺モジュール	動作	停止* ²	停止* ²
I/O ポート	動作	保持	保持

【注】 停止（保持）は、内部レジスタ値保持、内部状態は動作中断を示します。

*1 外部割り込み、一部の内部割り込み（ウォッチドッグタイマ）

*2 RCAN はリセット状態、その他の周辺モジュールは状態を保持します。

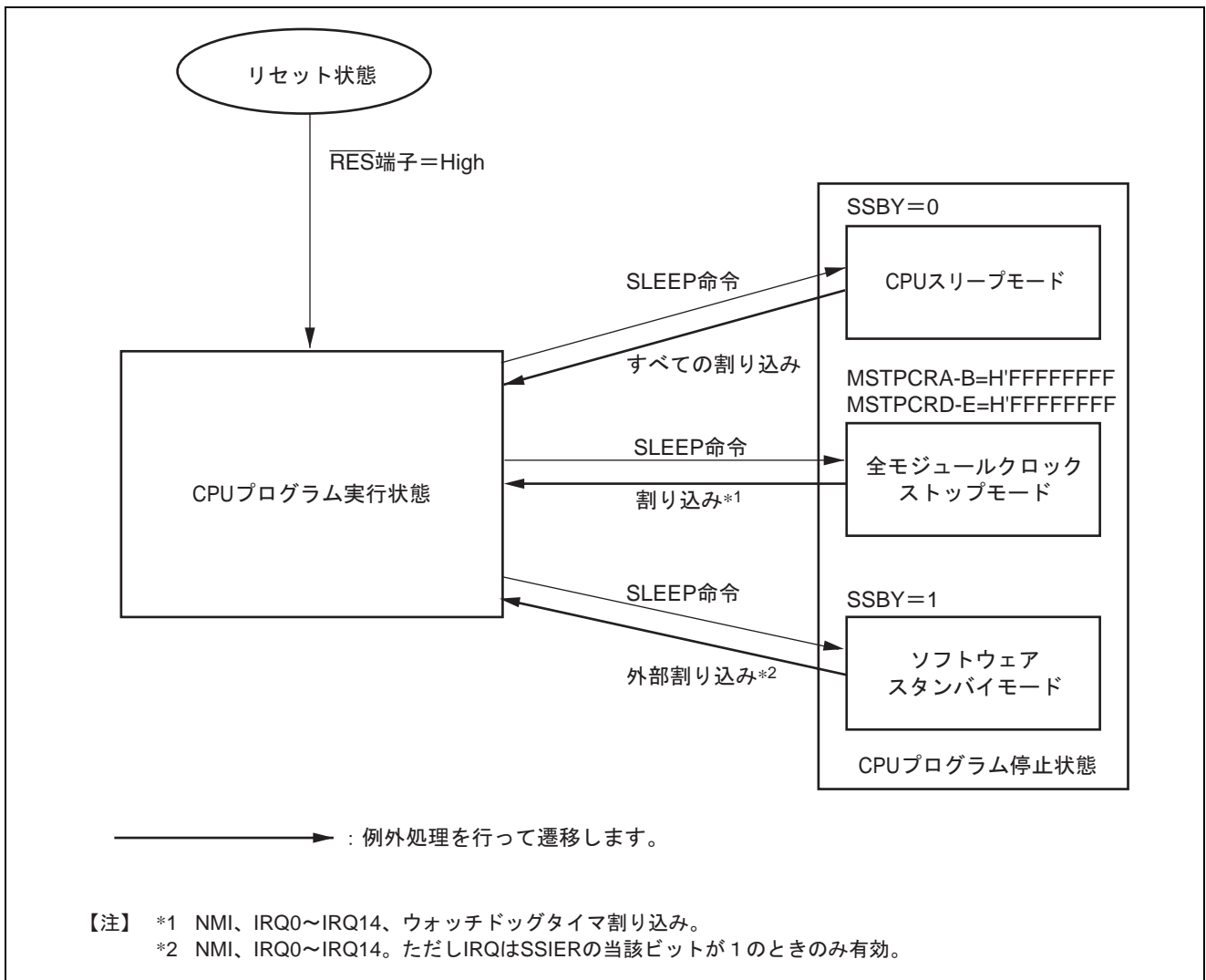


図 23.1 モード遷移図

23.2 レジスタの説明

低消費電力モードに関連するレジスタには以下のものがあります。システムクロックコントロールレジスタ 0 (SCKCR0)、システムクロックコントロールレジスタ 1 (SCKCR1) については「22.1.1 システムクロックコントロールレジスタ 0(SCKCR0)」および「22.1.2 システムクロックコントロールレジスタ 1(SCKCR1)」を参照してください。

- スタンバイコントロールレジスタ (SBYCR)
- モジュールストップコントロールレジスタA (MSTPCRA)
- モジュールストップコントロールレジスタB (MSTPCRB)
- モジュールストップコントロールレジスタC (MSTPCRC)
- モジュールストップコントロールレジスタD (MSTPCRD)
- モジュールストップコントロールレジスタE (MSTPCRE)

23.2.1 スタンバイコントロールレジスタ (SBYCR)

SBYCR は、ソフトウェアスタンバイモードの制御を行います。

ビット	15	14	13	12	11	10	9	8
ビット名	SSBY	OPE	SSBYF	STS4	STS3	STS2	STS1	STS0
初期値 :	0	1	0	0	1	1	1	1
R/W :	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
ビット	7	6	5	4	3	2	1	0
ビット名	—	—	—	—	—	—	—	—
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

ビット	ビット名	初期値	R/W	説明
15	SSBY	0	R/W	ソフトウェアスタンバイ SLEEP 命令実行後の遷移先を設定します。 0 : SLEEP 命令実行後、スリープモードに遷移 1 : SLEEP 命令実行後、ソフトウェアスタンバイモードに遷移 なお、外部割り込みによってソフトウェアスタンバイモードが解除され通常モードに遷移したときは、このビットは1にセットされたままです。クリアするときは0をライトしてください。WDT をウォッチドッグタイマとして使用しているときは、このビットの設定は無効になります。その場合、SLEEP 命令実行後は常にスリープモードあるいは全モジュールクロックストップモードに遷移します。

ビット	ビット名	初期値	R/W	説明
14	OPE	1	R/W	<p>出力ポートイネーブル</p> <p>ソフトウェアスタンバイモード時にアドレスバス、バス制御信号($\overline{CS0} \sim \overline{CS7}$、$\overline{AS}$、$\overline{RD}$、$\overline{HWR}$、$\overline{LWR}$、$\overline{UCAS}$、$\overline{LCAS}$)の出力を保持するか、ハイインピーダンスにするかを設定します。</p> <p>0: ソフトウェアスタンバイモード時、アドレスバス、バス制御信号はハイインピーダンス</p> <p>1: ソフトウェアスタンバイモード時、アドレスバス、バス制御信号は出力状態を保持</p>
13	SSBYF	0	R	<p>ソフトウェアスタンバイフラグ</p> <p>ソフトウェアスタンバイに遷移すると1にセットします。</p> <p>\overline{RES} 端子によるリセットにより0にクリアします。ソフトウェアスタンバイモードをNMI端子の立ち上がりエッジ入力などにより解除した場合は、1にセットしたまま0にクリアしません。</p> <p>0: ソフトウェアスタンバイモードに遷移していないか、ソフトウェアスタンバイモードに遷移後、\overline{RES} 端子によりリセットされている</p> <p>1: ソフトウェアスタンバイモードから、復帰後、\overline{RES} 端子によりリセットされていない</p>

ビット	ビット名	初期値	R/W	説明	
12	STS4	0	R/W	スタンバイタイムセレクト 4~0 外部割り込みによってソフトウェアスタンバイモードを解除する場合に、クロックが安定するまで MCU が待機する時間を選択します。水晶発振の場合、表 23.2 を参照し、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選択してください。外部クロックの場合においても、動作周波数に応じて発振安定時間が必要になります。表 23.2 を参照に待機時間を設定してください。 発振安定期間中は、Pφの周波数でカウントされます。マルチクロック機能時は注意してください。 00000 : リザーブ 00001 : リザーブ 00010 : リザーブ 00011 : リザーブ 00100 : リザーブ 00101 : 待機時間=64 ステート 00110 : 待機時間=512 ステート 00111 : 待機時間=1024 ステート 01000 : 待機時間=2048 ステート 01001 : 待機時間=4096 ステート 01010 : 待機時間=16384 ステート 01011 : 待機時間=32768 ステート 01100 : 待機時間=65536 ステート 01101 : 待機時間=131072 ステート 01110 : 待機時間=262144 ステート 01111 : 待機時間=524288 ステート 10000 : リザーブ 10001 : リザーブ 1001* : リザーブ 101** : リザーブ 11*** : リザーブ	
11	STS3	1	R/W		
10	STS2	1	R/W		
9	STS1	1	R/W		
8	STS0	1	R/W		
7~0	—	0	R/W		リザーブビット リードすると常に 0 が読み出されます。ライトする値は常に 0 にしてください。

【注】 * : Don't care

23.2.2 モジュールストップコントロールレジスタ A、B、C、D、E (MSTPCRA、MSTPCRB、MSTPCRC、MSTPCRD、E)

MSTPCRA、MSTPCRB、MSTPCRC、MSTPCRD、MSTPCRE はモジュールストップモードの制御を行います。1 のとき対応するモジュールはモジュールストップモードになり、クリアするとモジュールストップモードは解除されます。

• MSTPCRA

ビット	15	14	13	12	11	10	9	8
ビット名	ACSE	MSTPA14	MSTPA13	MSTPA12	MSTPA11	MSTPA10	MSTPA9	MSTPA8
初期値 :	0	0	0	0	1	1	1	1
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ビット	7	6	5	4	3	2	1	0
ビット名	MSTPA7	MSTPA6	—	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0
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

• MSTPCRB

ビット	15	14	13	12	11	10	9	8
ビット名	MSTPB15	MSTPB14	MSTPB13	MSTPB12	MSTPB11	MSTPB10	MSTPB9	MSTPB8
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

ビット	7	6	5	4	3	2	1	0
ビット名	MSTPB7	MSTPB6	MSTPB5	MSTPB4	MSTPB3	MSTPB2	MSTPB1	MSTPB0
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

• MSTPCRC

ビット	15	14	13	12	11	10	9	8
ビット名	MSTPC15	MSTPC14	MSTPC13	MSTPC12	MSTPC11	MSTPC10	MSTPC9	MSTPC8
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

ビット	7	6	5	4	3	2	1	0
ビット名	MSTPC7	MSTPC6	MSTPC5	MSTPC4	MSTPC3	MSTPC2	MSTPC1	MSTPC0
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

- MSTPCRD

ビット	15	14	13	12	11	10	9	8
ビット名	MSTPD15	MSTPD14	MSTPD13	MSTPD12	MSTPD11	MSTPD10	MSTPD9	MSTPD8
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
ビット	7	6	5	4	3	2	1	0
ビット名	MSTPD7	MSTPD6	MSTPD5	MSTPD4	MSTPD3	MSTPD2	MSTPD1	MSTPD0
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

- MSTPCRE

ビット	15	14	13	12	11	10	9	8
ビット名	MSTPE15	MSTPE14	MSTPE13	MSTPE12	MSTPE11	MSTPE10	MSTPE9	MSTPE8
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
ビット	7	6	5	4	3	2	1	0
ビット名	MSTPE7	MSTPE6	MSTPE5	MSTPE4	MSTPE3	MSTPE2	MSTPE1	MSTPE0
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

- MSTPCRA

ビット	ビット名	初期値	R/W	対象モジュール
15	ACSE	0	R/W	全モジュールクロックストップモードイネーブル MSTPCRで制御されるすべてのモジュールがモジュールストップモードに設定された上で、CPUがSLEEP命令を実行した場合にバスコントローラとI/Oポートも動作をストップして、消費電流を低減する全モジュールクロックストップモードの許可または禁止を設定します。 0: 全モジュールクロックストップモード禁止 1: 全モジュールクロックストップモード許可
14	MSTPA14	0	R/W	リザーブビット
13	MSTPA13	0	R/W	DMAコントローラ (DMAC)
12	MSTPA12	0	R/W	データトランスファコントローラ (DTC)
11	MSTPA11	1	R/W	リザーブビット リードすると常に1が読み出されます。ライトする値は常に1にしてください。
10	MSTPA10	1	R/W	リザーブビット リードすると常に1が読み出されます。ライトする値は常に1にしてください。
9	MSTPA9	1	R/W	リザーブビット リードすると常に1が読み出されます。ライトする値は常に1にしてください。
8	MSTPA8	1	R/W	リザーブビット リードすると常に1が読み出されます。ライトする値は常に1にしてください。
7	MSTPA7	1	R/W	リザーブビット リードすると常に1が読み出されます。ライトする値は常に1にしてください。
6	MSTPA6	1	R/W	リザーブビット リードすると常に1が読み出されます。ライトする値は常に1にしてください。
5	MSTPA5	1	R/W	リザーブビット リードすると常に1が読み出されます。ライトする値は常に1にしてください。
4	MSTPA4	1	R/W	A/D変換器 (ユニット1)
3	MSTPA3	1	R/W	A/D変換器 (ユニット0)
2	MSTPA2	1	R/W	リザーブビット リードすると常に1が読み出されます。ライトする値は常に1にしてください。
1	MSTPA1	1	R/W	16ビットタイマパルスユニット (TPUチャンネル11~6)
0	MSTPA0	1	R/W	16ビットタイマパルスユニット (TPUチャンネル5~0)

- MSTPCRB

ビット	ビット名	初期値	R/W	対象モジュール
15	MSTPB15	1	R/W	プログラマブルパルスジェネレータ (PPG)
14	MSTPB14	1	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください。
13	MSTPB13	1	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください
12	MSTPB12	1	R/W	シリアルコミュニケーションインタフェース_4 (SCI_4)
11	MSTPB11	1	R/W	シリアルコミュニケーションインタフェース_3 (SCI_3)
10	MSTPB10	1	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください
9	MSTPB9	1	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください
8	MSTPB8	1	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください
7	MSTPB7	1	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください
6	MSTPB6	1	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください
5	MSTPB5	1	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください
4	MSTPB4	1	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください。
3	MSTPB3	1	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください
2	MSTPB2	1	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください。
1	MSTPB1	1	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください。
0	MSTPB0	1	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください。

- MSTPCRC

ビット	ビット名	初期値	R/W	対象モジュール
15	MSTPC15	1	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください。
14	MSTPC14	1	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください。
13	MSTPC13	1	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください。
12	MSTPC12	1	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください。
11	MSTPC11	1	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください。
10	MSTPC10	1	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください。
9	MSTPC9	1	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください。
8	MSTPC8	1	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください。
7	MSTPC7	0	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください。
6	MSTPC6	0	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください。
5	MSTPC5	0	R/W	内蔵 RAM_4 (H'FFF2000~H'FFF3FFF)
4	MSTPC4	0	R/W	MSTPC5 と MSTPC4 は常に同じ値を設定してください。 本ビット設定中の RAM 読み出し値は保証しません。
3	MSTPC3	0	R/W	内蔵 RAM_3、2 (H'FFF4000~H'FFF7FFF)
2	MSTPC2	0	R/W	MSTPC3 と MSTPC2 は常に同じ値を設定してください。 本ビット設定中の RAM 読み出し値は保証しません。
1	MSTPC1	0	R/W	内蔵 RAM_1、0 (H'FFF8000~H'FFFBFFF)
0	MSTPC0	0	R/W	MSTPC1 と MSTPC0 は常に同じ値を設定してください。 本ビット設定中の RAM 読み出し値は保証しません。

- MSTPCR

ビット	ビット名	初期値	R/W	対象モジュール
15	MSTPD15	1	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください
14	MSTPD14	1	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください
13	MSTPD13	1	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください
12	MSTPD12	1	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください
11	MSTPD11	1	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください
10	MSTPD10	1	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください
9	MSTPD9	1	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください
8	MSTPD8	1	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください
7	MSTPD7	1	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください
6	MSTPD6	1	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください
5	MSTPD5	1	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください
4	MSTPD4	1	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください
3	MSTPD3	1	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください
2	MSTPD2	1	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください
1	MSTPD1	1	R/W	エラー訂正 (CRC)
0	MSTPD0	1	R/W	リザーブビット リードすると常に 1 が読み出されます。ライトする値は常に 1 にしてください。

- MSTPCRE

ビット	ビット名	初期値	R/W	対象モジュール
15	MSTPE15	1	R/W	コントロールエリアネットワーク_1 (RCAN-TL1_1)
14	MSTPE14	1	R/W	コントロールエリアネットワーク_0 (RCAN-TL1_0)
13	MSTPE13	1	R/W	ハードウェアローカルインターコネクトネットワーク (HW-LIN)
12	MSTPE12	1	R/W	リザーブビット リードすると常に1が読み出されます。ライトする値は常に1にしてください
11	MSTPE11	1	R/W	ルネサスシリアルペリフェラルインタフェース_D (RSPI_D)
10	MSTPE10	1	R/W	ルネサスシリアルペリフェラルインタフェース_C (RSPI_C)
9	MSTPE9	1	R/W	ルネサスシリアルペリフェラルインタフェース_B (RSPI_B)
8	MSTPE8	1	R/W	ルネサスシリアルペリフェラルインタフェース_A (RSPI_A)
7	MSTPE7	1	R/W	リザーブビット リードすると常に1が読み出されます。ライトする値は常に1にしてください。
6	MSTPE6	1	R/W	リザーブビット リードすると常に1が読み出されます。ライトする値は常に1にしてください。
5	MSTPE5	1	R/W	リザーブビット リードすると常に1が読み出されます。ライトする値は常に1にしてください。
4	MSTPE4	1	R/W	リザーブビット リードすると常に1が読み出されます。ライトする値は常に1にしてください。
3	MSTPE3	1	R/W	リザーブビット リードすると常に1が読み出されます。ライトする値は常に1にしてください。
2	MSTPE2	1	R/W	リザーブビット リードすると常に1が読み出されます。ライトする値は常に1にしてください。
1	MSTPE1	1	R/W	リザーブビット リードすると常に1が読み出されます。ライトする値は常に1にしてください。
0	MSTPE0	1	R/W	リザーブビット リードすると常に1が読み出されます。ライトする値は常に1にしてください。

23.3 マルチクロック機能

SCKCR0 の ICK2~ICK0 ビット、PCK2~PCK0 ビットと BCK2~BCK0 ビット、SCKCR1 の FCK2~FCK0 ビット、ACK2~ACK0 ビット、RCK2~RCK0 ビットを設定すると、そのバスサイクルの終了時点でマルチクロック機能になります。マルチクロック機能では、CPU とバスマスタは、ICK2~ICK0 ビットで設定した動作クロックで動作します。周辺モジュールは、PCK2~PCK0 ビット、FCK2~FCK0 ビット、ACK2~ACK0 ビット、RCK2~RCK0 ビットで設定した動作クロックで動作します。また、外部クロックは、BCK2~BCK0 ビットで設定した動作クロックで動作します。

ただし、ICK2~ICK0 ビットで設定した動作クロックより高い動作クロックに PCK2~PCK0 ビット、BCK2~BCK0 ビット、FCK2~FCK0 ビット、ACK2~ACK0 ビットおよび RCK2~RCK0 ビットを設定しても、設定値がクロックに反映されません。周辺モジュール、外部クロックは、ICK2~ICK0 ビットで設定した動作クロックに制限されます。

マルチクロック機能の解除は、ICK2~ICK0 ビット、PCK2~PCK0 ビット、BCK2~BCK0 ビット、FCK2~FCK0 ビット、ACK2~ACK0 ビットおよび RCK2~RCK0 ビットをすべて 0 にクリアすることによって行われ、そのバスサイクルの終了時点で通常状態に遷移し、マルチクロック機能は解除されます。

SBYCR の SSBY ビットを 0 にクリアした状態で SLEEP 命令を実行すると、スリープモードに遷移します。スリープモード期間中もマルチクロック機能は有効です。

また、SSBY ビットを 1 にセットした状態で SLEEP 命令を実行すると、ソフトウェアスタンバイモードに遷移します。ソフトウェアスタンバイモード期間中はマルチクロック機能は停止し、ソフトウェアスタンバイモードが外部割り込みによって解除されるとマルチクロック機能が有効な状態に復帰します。

$\overline{\text{RES}}$ 端子を Low レベルにするとリセット状態に遷移し、マルチクロック機能は解除されます。ウォッチドッグタイマのオーバフローによるリセットについても同様です。

23.4 モジュールストップ機能

モジュールストップ機能は、内蔵周辺機能のモジュール単位で設定することができます。

MSTPCRA、MSTPCRB、MSTPCRC、MSTPCRD、MSTPCRE の対応する MSTP ビットを 1 にセットすると、バスサイクルの終了時点でモジュールは動作を停止してモジュールストップモードへ遷移します。このとき CPU は独立して動作を継続します。

対応する MSTP ビットを 0 にクリアすることによって、モジュールストップモードは解除され、バスサイクルの終了時点でモジュールは動作を再開します。モジュールストップモードでは、モジュールの内部状態が保持されています。

リセット解除後は、DMAC、DTC を除くすべてのモジュールがモジュールストップモードになっています。モジュールストップモードに設定されたモジュールのレジスタは、リード/ライトできません。

23.5 スリープモード

23.5.1 スリープモードへの遷移

SBYCR の SSBY ビット=0 の状態で SLEEP 命令を実行すると、CPU はスリープモードになります。スリープモード時、CPU の動作は停止しますが、CPU の内部レジスタの内容は保持されます。CPU 以外の周辺機能は停止しません。

23.5.2 スリープモードの解除

スリープモードの解除は、すべての割り込み、 $\overline{\text{RES}}$ 端子、またはウォッチドッグタイマのオーバフローによるリセットによって行われます。

1. 割り込みによる解除

割り込みが発生すると、スリープモードは解除され、割り込み例外処理を開始します。割り込みが禁止されている場合、また、NMI以外の割り込みがCPUでマスクされている場合には、スリープモードは解除されません。

2. $\overline{\text{RES}}$ 端子による解除

$\overline{\text{RES}}$ 端子をLowレベルにすると、リセット状態になります。規定のリセット入力期間後、 $\overline{\text{RES}}$ 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。

3. ウォッチドッグタイマのオーバフローリセットによる解除

ウォッチドッグタイマのオーバフローの内部リセットにより、スリープモードが解除されます。

23.6 全モジュールクロックストップモード

ACSE を 1 に設定し、かつ MSTPCR で制御されるすべてのモジュールをストップ

(MSTPCRA、B、D、E=H'FFFFFFFFFFFFFFFF) したとき、SBYCR の SSBY ビットを 0 にクリアした状態で SLEEP 命令を実行すると、バスサイクルの終了時点でウォッチドッグタイマを除く全モジュール、バスコントローラ、および I/O ポートの動作を停止して全モジュールクロックストップモードへ遷移します。

全モジュールクロックストップモードの解除は、外部割り込み (NMI 端子、 $\overline{\text{IRQ0}} \sim \overline{\text{IRQ14}}$ 端子)、 $\overline{\text{RES}}$ 端子、内部割り込み (ウォッチドッグタイマ) によって行われ、例外処理状態を経て通常のプログラム実行状態へ遷移します。割り込みが禁止されている場合、また、NMI 以外の割り込みが CPU でマスクされている場合、または DTC の起動要因に設定した場合には、全モジュールクロックストップモードは解除されません。

23.7 ソフトウェアスタンバイモード

23.7.1 ソフトウェアスタンバイモードへの遷移

SBYCR の SSBY ビットを 1 にセットした状態で SLEEP 命令を実行すると、ソフトウェアスタンバイモードになります。このモードでは、CPU、周辺モジュール、および発振器のすべての機能が停止します。ただし、CPU の内部レジスタの内容と内蔵 RAM のデータ、SCI を除く内蔵周辺機能と、I/O ポートの状態は保持されます。アドレスバス、バス制御信号は、ハイインピーダンス状態とするか、出力状態を保持するかを、SBYCR の OPE ビットにより設定できます。本モードでは、発振器が停止するため、消費電力は著しく低減されます。

WDT をウォッチドッグタイマとして使用している場合、ソフトウェアスタンバイモードに遷移できません。SLEEP 命令を実行する前に WDT を停止させてください。

23.7.2 ソフトウェアスタンバイモードの解除

ソフトウェアスタンバイモードの解除は、外部割り込み (NMI 端子、 $\overline{\text{IRQ0}}\sim\overline{\text{IRQ14}}$ 端子*) $\overline{\text{RES}}$ 端子によって行われます。

1. 割り込みによる解除

NMI、 $\overline{\text{IRQ0}}\sim\overline{\text{IRQ14}}$ *割り込み要求信号が入力されると、クロックが発振を開始し、SBYCRのSTS4～STS0 ビットによって設定された時間が経過した後、安定したクロックが本LSI全体に供給されて、ソフトウェアスタンバイモードは解除され、割り込み例外処理を開始します。

$\overline{\text{IRQ0}}\sim\overline{\text{IRQ14}}$ *割り込みでソフトウェアスタンバイモードを解除する場合には、対応するイネーブルビットを 1 にセットし、かつ $\overline{\text{IRQ0}}\sim\overline{\text{IRQ14}}$ *割り込みより高い優先順位の割り込みが発生しないようにしてください。なお、CPU 側でマスクした場合、または DTC の起動要因に設定した場合には、ソフトウェアスタンバイモードは解除できません。

【注】 * SSIER の SSIn ビットを 1 にセットすることにより、 $\overline{\text{IRQ0}}\sim\overline{\text{IRQ14}}$ をソフトウェアスタンバイモードの解除要因として使用することができます。

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、クロックの発振が開始されます。クロックの発振開始と同時に、本LSIにクロックが供給されます。このとき $\overline{\text{RES}}$ 端子は必ずクロックの発振が安定するまでLowレベルに保持してください。 $\overline{\text{RES}}$ 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。

23.7.3 ソフトウェアスタンバイモード解除後の発振安定時間の設定

SBYCR の STS4~STS0 ビットの設定は、以下のようにしてください。

1. 水晶発振の場合

待機時間が発振安定時間以上となるようにSTS4~STS0ビットを設定してください。

表 23.2 に、動作周波数と STS4~STS0 ビットの設定に対する待機時間を示します。

2. 外部クロックの場合

水晶発振同様、発振安定時間が必要となります。表23.2を参照し待機時間を設定してください。

表 23.2 発振安定時間の設定

STS 4	STS 3	STS 2	STS 1	STS 0	待機時間	Pφ* [MHz]							単位	
						40	33	25	20	13	10	8		
0	0	0	0	0	リザーブ	—	—	—	—	—	—	—	μs	
				1	リザーブ	—	—	—	—	—	—	—		
			1	0	リザーブ	—	—	—	—	—	—	—		
				1	リザーブ	—	—	—	—	—	—	—		
		1	0	0	リザーブ	—	—	—	—	—	—	—		
				1	64	1.6	1.9	2.6	3.2	4.9	6.4	8.0		
			1	0	0	512	12.8	15.5	20.5	25.6	39.4	51.2		64.0
					1	1024	25.6	31.0	41.0	51.2	78.8	102.4		128.0
	0			0	2048	51.2	62.1	81.9	102.4	157.5	204.8	256.0		
				1	4096	0.10	0.12	0.16	0.20	0.32	0.41	0.51		ms
	1	0	0	16384	0.41	0.50	0.66	0.82	1.26	1.64	2.05			
			1	32768	0.82	0.99	1.31	1.64	2.52	3.28	4.10			
	1	0	0	0	65536	1.64	1.99	2.62	3.28	5.04	6.55	8.19		
				1	131072	3.28	3.97	5.24	6.55	10.08	13.11	16.38		
1		0	0	262144	6.55	7.94	10.49	13.11	20.16	26.21	32.77			
			1	524288	13.11	15.89	20.97	26.21	40.33	52.43	65.54			
1	0	0	0	0	1048576	26.21	31.78	41.94	52.43	80.66	104.86	131.07		

□ : 外部クロック使用時の推奨設定時間

■ : 水晶発振使用時の推奨設定時間

【注】 * Pφは周辺モジュール分周器の出力です。

23.7.4 ソフトウェアスタンバイモードの応用例

NMI 端子の立ち下がりエッジでソフトウェアスタンバイモードに遷移し、NMI 端子の立ち上がりエッジでソフトウェアスタンバイモードの解除を行う例を図 23.2 に示します。

この例では、INTCR の NMIEG ビットが 0 にクリアされている（立ち下がりエッジ設定）状態で、NMI 割り込みを受け付けた後、NMIEG ビットを 1 にセット（立ち上がりエッジ設定）、SSBY ビットを 1 にセットした後、SLEEP 命令を実行してソフトウェアスタンバイモードに遷移しています。

その後、NMI 端子の立ち上がりエッジでソフトウェアスタンバイモードが解除されます。

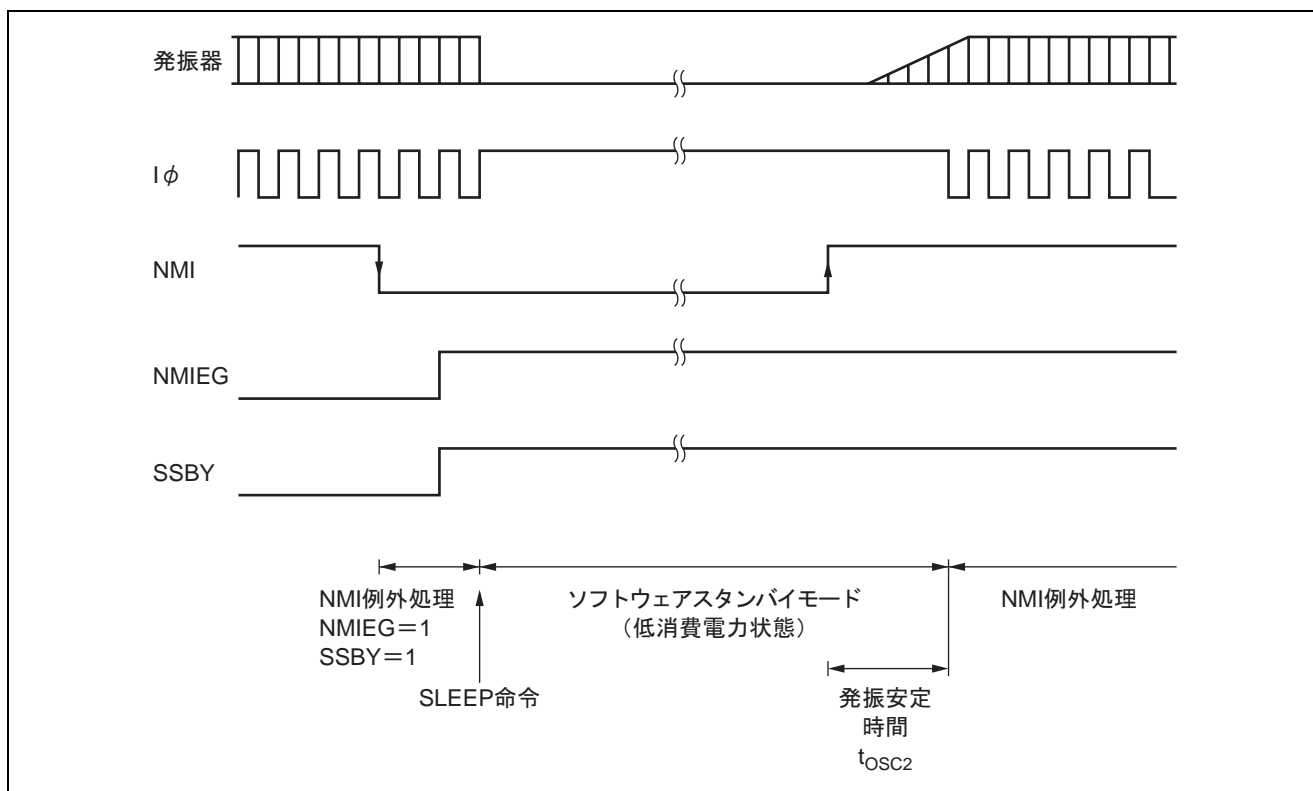


図 23.2 ソフトウェアスタンバイモードの応用例

23.8 φクロック出力制御

SCKCR0 の PSTOP1、POSEL1 ビットと対応する PA7 の DDR により、φクロック (Bφ) の出力を制御することができます。PSTOP1、POSEL1 ビットをともに 0 にクリアすると PA7 は Bφ 出力になります。PSTOP1 ビットを 1 にセットすると、バスサイクルの終了時点で PA7 の Bφ 出力は停止し、Bφ 出力は High レベルになります。また、PA7 の DDR を 0 にクリアすると、Bφ 出力は禁止され入力ポートになります。

表 23.3 に各処理状態における φ 端子の状態を示します。

表 23.3 各処理状態における φ 端子 (PA7) の状態

レジスタの設定値			通常動作 状態	スリープ	全モジュール クロック ストップ	ソフトウェア スタンバイ	
DDR	PSTOP1	POSEL1				OPE=0	OPE=1
0	x	x	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
1	0	0	Bφ 出力	Bφ 出力	Bφ 出力	Hi-Z	High
1	0	1	設定禁止	設定禁止	設定禁止	設定禁止	設定禁止
1	1	x	High	High	High	Hi-Z	High

23.9 使用上の注意事項

23.9.1 I/O ポートの状態

ソフトウェアスタンバイモードでは、I/O ポートの状態が保持されます。したがって、High レベルを出力している場合は出力電流分の消費電流は低減されません。

23.9.2 発振安定待機中の消費電流

発振安定待機中は消費電流が増加します。

23.9.3 DMAC、DTC のモジュールストップ

DMAC、DTC の動作状態によって、MSTPA13、12 ビットは 1 にセットされない場合があります。DMAC または DTC のモジュールストップモードの設定は、それぞれ起動されていない状態で行ってください。

詳細は「第 7 章 DMA コントローラ (DMAC)」、「第 8 章 データトランスファコントローラ (DTC)」を参照してください。

23.9.4 内蔵周辺モジュールの割り込み

モジュールストップモードでは当該割り込みの動作ができません。したがって、割り込みが要求された状態でモジュールストップとすると、CPU の割り込み要因または DMAC、DTC の起動要因のクリアができません。事前に割り込みをディスエーブルにするなどしてからモジュールストップモードにしてください。

23.9.5 MSTPCRA、MSTPCRB、MSTPCRC、MSTPCRD、MSTPCRE のライト

MSTPCRA、MSTPCRB、MSTPCRC、MSTPCRD および MSTPCRE は、CPU のみでライトしてください。

24. レジスタ一覧

レジスタ一覧では、内蔵レジスタのアドレス、ビット構成および動作モード別の状態に関する情報をまとめています。表記方法は下記のとおりです。

1. レジスタアドレス一覧（アドレス順）

- 割り付けアドレスの小さいレジスタから順に記載します。
- モジュール名称による分類をしています。
- アクセスステート数については、指定の基準クロックのステート数を示しています。
- 未定義・リザーブアドレスのアクセスは禁止します。これらのレジスタをアクセスしたときの動作および継続する動作については保証できませんので、アクセスしないようにしてください。

2. y レジスタビット一覧

- 「レジスタアドレス一覧（アドレス順）」の順序で、ビット構成を記載します。
- リザーブビットは、ビット名称部に「-」で表記しています。
- ビット名称部が空白のものは、そのレジスタ全体がカウンタやデータに割り付けられていることを示します。
- 16ビットまたは32ビットのレジスタの場合、MSB側のビットから記載しています。バイトの記載順序はビッグエンディアンを前提としています。

3. 各動作モードにおけるレジスタの状態

- 「レジスタアドレス一覧（アドレス順）」の順序で、レジスタの状態を記載します。
- 初期化の各ビットの状態は、該当する章のレジスタ説明を参照してください。
- 基本的な動作モード時のレジスタの状態を示しており、内蔵モジュール固有のリセットなどがある場合は、周辺モジュールの章を参照してください。

24.1 レジスタアドレス一覧（アドレス順）

レジスタ名称	略称	ビット数	アドレス*1	モジュール	データ バス幅	アクセス*2 ステート数 (リード/ライト)
マスタコントロールレジスタ_0	MCR_0	16	H'FFC000	RCAN_0	16	2Pφ/3Pφ
ジェネラルステータスレジスタ_0	GSR_0	16	H'FFC002	RCAN_0	16	2Pφ/3Pφ
ビットコンフィギュレーションレジスタ 1_0	BCR1_0	16	H'FFC004	RCAN_0	16	2Pφ/3Pφ
ビットコンフィギュレーションレジスタ 0_0	BCR0_0	16	H'FFC006	RCAN_0	16	2Pφ/3Pφ
インタラプトリクエストレジスタ_0	IRR_0	16	H'FFC008	RCAN_0	16	2Pφ/3Pφ
インタラプトマスクレジスタ_0	IMR_0	16	H'FFC00A	RCAN_0	16	2Pφ/3Pφ
送信エラーカウンタ_0	TEC_0	16	H'FFC00C	RCAN_0	16	2Pφ/3Pφ
受信エラーカウンタ_0	REC_0					
送信待ちレジスタ 1_0	TXPR1_0	16	H'FFC020	RCAN_0	32	4Pφ/6Pφ
送信待ちレジスタ 0_0	TXPR0_0	16				
送信キャンセルレジスタ 1_0	TXCR1_0	16	H'FFC028	RCAN_0	16	2Pφ/3Pφ
送信キャンセルレジスタ 0_0	TXCR0_0	16	H'FFC02A	RCAN_0	16	2Pφ/3Pφ
送信アクノリッジレジスタ 1_0	TXACK1_0	16	H'FFC030	RCAN_0	16	2Pφ/3Pφ
送信アクノリッジレジスタ 0_0	TXACK0_0	16	H'FFC032	RCAN_0	16	2Pφ/3Pφ
アポートアクノリッジレジスタ 1_0	ABACK1_0	16	H'FFC038	RCAN_0	16	2Pφ/3Pφ
アポートアクノリッジレジスタ 0_0	ABACK0_0	16	H'FFC03A	RCAN_0	16	2Pφ/3Pφ
データフレーム受信完了レジスタ 1_0	RXPR1_0	16	H'FFC040	RCAN_0	16	2Pφ/3Pφ
データフレーム受信完了レジスタ 0_0	RXPR0_0	16	H'FFC042	RCAN_0	16	2Pφ/3Pφ
リモートフレーム受信完了レジスタ 1_0	RFPR1_0	16	H'FFC048	RCAN_0	16	2Pφ/3Pφ
リモートフレーム受信完了レジスタ 0_0	RFPR0_0	16	H'FFC04A	RCAN_0	16	2Pφ/3Pφ
メールボックスインタラプトマスクレジスタ 1_0	MBIMR1_0	16	H'FFC050	RCAN_0	16	2Pφ/3Pφ
メールボックスインタラプトマスクレジスタ 0_0	MBIMR0_0	16	H'FFC052	RCAN_0	16	2Pφ/3Pφ
未読メッセージステータスレジスタ 1_0	UMSR1_0	16	H'FFC058	RCAN_0	16	2Pφ/3Pφ
未読メッセージステータスレジスタ 0_0	UMSR0_0	16	H'FFC05A	RCAN_0	16	2Pφ/3Pφ
タイムトリガコントロールレジスタ 0_0	TTCR0_0	16	H'FFC080	RCAN_0	16	2Pφ/3Pφ
サイクルマキシマム/ Tx_Enable_Window レジスタ_0	CMAX_TEW_0	16	H'FFC084	RCAN_0	16	2Pφ/3Pφ
リファレンストリガオフセットレジスタ_0	RFTR0FF_0	16	H'FFC086	RCAN_0	16	2Pφ/3Pφ
タイマステータスレジスタ_0	TSR_0	16	H'FFC088	RCAN_0	16	2Pφ/3Pφ
サイクルカウンタレジスタ_0	CCR_0	16	H'FFC08A	RCAN_0	16	2Pφ/3Pφ
タイマカウンタレジスタ_0	TCNTR_0	16	H'FFC08C	RCAN_0	16	2Pφ/3Pφ
サイクルタイムレジスタ_0	CYCTR_0	16	H'FFC090	RCAN_0	16	2Pφ/3Pφ
リファレンスマークレジスタ_0	RFMK_0	16	H'FFC094	RCAN_0	16	2Pφ/3Pφ
タイマコンペアマッチレジスタ 0_0	TCMR0_0	16	H'FFC098	RCAN_0	16	2Pφ/3Pφ

レジスタ名称	略称	ビット数	アドレス* ¹	モジュール	データ バス幅	アクセス* ² ステート数 (リード/ライト)	
タイマコンペアマッチレジスタ 1_0	TCMR1_0	16	H'FFC09C	RCAN_0	16	2Pφ/3Pφ	
タイマコンペアマッチレジスタ 2_0	TCMR2_0	16	H'FFC0A0	RCAN_0	16	2Pφ/3Pφ	
送信トリガタイムセレクトレジスタ_0	TTTSEL_0	16	H'FFC0A4	RCAN_0	16	2Pφ/3Pφ	
MB_0[0].	CONTROL0H	—	H'FFC100	RCAN_0	16	2+nPφ/3+nPφ	
	CONTROL0L	—	H'FFC102	RCAN_0	16	2+nPφ/3+nPφ	
	LAFMH	—	H'FFC104	RCAN_0	16	2+nPφ/3+nPφ	
	LAFML	—	H'FFC106	RCAN_0	16	2+nPφ/3+nPφ	
	MSG_DATA[0]	—	H'FFC108	RCAN_0	16	2+nPφ/3+nPφ	
	MSG_DATA[1]	—	H'FFC109	RCAN_0	16	2+nPφ/3+nPφ	
	MSG_DATA[2]	—	H'FFC10A	RCAN_0	16	2+nPφ/3+nPφ	
	MSG_DATA[3]	—	H'FFC10B	RCAN_0	16	2+nPφ/3+nPφ	
	MSG_DATA[4]	—	H'FFC10C	RCAN_0	16	2+nPφ/3+nPφ	
	MSG_DATA[5]	—	H'FFC10D	RCAN_0	16	2+nPφ/3+nPφ	
	MSG_DATA[6]	—	H'FFC10E	RCAN_0	16	2+nPφ/3+nPφ	
	MSG_DATA[7]	—	H'FFC10F	RCAN_0	16	2+nPφ/3+nPφ	
	CONTROL1H	—	H'FFC110	RCAN_0	16	2+nPφ/3+nPφ	
	CONTROL1L	—	H'FFC111	RCAN_0	16	2+nPφ/3+nPφ	
	TIMESTAMP	—	16	H'FFC112	RCAN_0	16	2Pφ/3Pφ
MB_0[1].	CONTROL0H	—	H'FFC120	RCAN_0	16	2+nPφ/3+nPφ	
	CONTROL0L	—	H'FFC122	RCAN_0	16	2+nPφ/3+nPφ	
	LAFMH	—	H'FFC124	RCAN_0	16	2+nPφ/3+nPφ	
	LAFML	—	H'FFC126	RCAN_0	16	2+nPφ/3+nPφ	
	MSG_DATA[0]	—	H'FFC128	RCAN_0	16	2+nPφ/3+nPφ	
	MSG_DATA[1]	—	H'FFC129	RCAN_0	16	2+nPφ/3+nPφ	
	MSG_DATA[2]	—	H'FFC12A	RCAN_0	16	2+nPφ/3+nPφ	
	MSG_DATA[3]	—	H'FFC12B	RCAN_0	16	2+nPφ/3+nPφ	
	MSG_DATA[4]	—	H'FFC12C	RCAN_0	16	2+nPφ/3+nPφ	
	MSG_DATA[5]	—	H'FFC12D	RCAN_0	16	2+nPφ/3+nPφ	
	MSG_DATA[6]	—	H'FFC12E	RCAN_0	16	2+nPφ/3+nPφ	
	MSG_DATA[7]	—	H'FFC12F	RCAN_0	16	2+nPφ/3+nPφ	
	CONTROL1H	—	H'FFC130	RCAN_0	16	2+nPφ/3+nPφ	
	CONTROL1L	—	H'FFC131	RCAN_0	16	2+nPφ/3+nPφ	
	MB_0[1].	TIMESTAMP	—	16	H'FFC132	RCAN_0	16

レジスタ名称	略称	ビット数	アドレス* ¹	モジュール	データ バス幅	アクセス* ² ステート数 (リード/ライト)	
MB_0[2].	CONTROL0H	—	H'FFC140	RCAN_0	16	2+nP φ/3+nP φ	
	CONTROL0L	—	H'FFC142	RCAN_0	16	2+nP φ/3+nP φ	
	LAFMH	—	H'FFC144	RCAN_0	16	2+nP φ/3+nP φ	
	LAFML	—	H'FFC146	RCAN_0	16	2+nP φ/3+nP φ	
	MSG_DATA[0]	—	H'FFC148	RCAN_0	16	2+nP φ/3+nP φ	
	MSG_DATA[1]	—	H'FFC149	RCAN_0	16	2+nP φ/3+nP φ	
	MSG_DATA[2]	—	H'FFC14A	RCAN_0	16	2+nP φ/3+nP φ	
	MSG_DATA[3]	—	H'FFC14B	RCAN_0	16	2+nP φ/3+nP φ	
	MSG_DATA[4]	—	H'FFC14C	RCAN_0	16	2+nP φ/3+nP φ	
	MSG_DATA[5]	—	H'FFC14D	RCAN_0	16	2+nP φ/3+nP φ	
	MSG_DATA[6]	—	H'FFC14E	RCAN_0	16	2+nP φ/3+nP φ	
	MSG_DATA[7]	—	H'FFC14F	RCAN_0	16	2+nP φ/3+nP φ	
	CONTROL1H	—	H'FFC150	RCAN_0	16	2+nP φ/3+nP φ	
	CONTROL1L	—	H'FFC151	RCAN_0	16	2+nP φ/3+nP φ	
	TIMESTAMP	—	16	H'FFC152	RCAN_0	16	2P φ/3P φ
MB_0[3].	CONTROL0H	—	H'FFC160	RCAN_0	16	2+nP φ/3+nP φ	
	CONTROL0L	—	H'FFC162	RCAN_0	16	2+nP φ/3+nP φ	
	LAFMH	—	H'FFC164	RCAN_0	16	2+nP φ/3+nP φ	
	LAFML	—	H'FFC166	RCAN_0	16	2+nP φ/3+nP φ	
	MSG_DATA[0]	—	H'FFC168	RCAN_0	16	2+nP φ/3+nP φ	
	MSG_DATA[1]	—	H'FFC169	RCAN_0	16	2+nP φ/3+nP φ	
	MSG_DATA[2]	—	H'FFC16A	RCAN_0	16	2+nP φ/3+nP φ	
	MSG_DATA[3]	—	H'FFC16B	RCAN_0	16	2+nP φ/3+nP φ	
	MSG_DATA[4]	—	H'FFC16C	RCAN_0	16	2+nP φ/3+nP φ	
	MSG_DATA[5]	—	H'FFC16D	RCAN_0	16	2+nP φ/3+nP φ	
	MSG_DATA[6]	—	H'FFC16E	RCAN_0	16	2+nP φ/3+nP φ	
	MSG_DATA[7]	—	H'FFC16F	RCAN_0	16	2+nP φ/3+nP φ	
	CONTROL1H	—	8	H'FFC170	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1L	—	8	H'FFC171	RCAN_0	16	2+nP φ/3+nP φ
	TIMESTAMP	—	16	H'FFC172	RCAN_0	16	2P φ/3P φ
MB_0[4].	CONTROL0H	—	16	H'FFC180	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL0L	—	16	H'FFC182	RCAN_0	16	2+nP φ/3+nP φ
	LAFMH	—	16	H'FFC184	RCAN_0	16	2+nP φ/3+nP φ
	LAFML	—	16	H'FFC186	RCAN_0	16	2+nP φ/3+nP φ

レジスタ名称	略称	ビット数	アドレス* ¹	モジュール	データ バス幅	アクセス* ² ステート数 (リード/ライト)	
MB_0[4].	MSG_DATA[0]	—	8	H'FFC188	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[1]	—	8	H'FFC189	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[2]	—	8	H'FFC18A	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[3]	—	8	H'FFC18B	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[4]	—	8	H'FFC18C	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[5]	—	8	H'FFC18D	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[6]	—	8	H'FFC18E	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	8	H'FFC18F	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1H	—	8	H'FFC190	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1L	—	8	H'FFC191	RCAN_0	16	2+nP φ/3+nP φ
	TIMESTAMP	—	16	H'FFC192	RCAN_0	16	2P φ/3P φ
MB_0[5].	CONTROL0H	—	16	H'FFC1A0	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL0L	—	16	H'FFC1A2	RCAN_0	16	2+nP φ/3+nP φ
	LAFMH	—	16	H'FFC1A4	RCAN_0	16	2+nP φ/3+nP φ
	LAFML	—	16	H'FFC1A6	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[0]	—	8	H'FFC1A8	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[1]	—	8	H'FFC1A9	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[2]	—	8	H'FFC1AA	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[3]	—	8	H'FFC1AB	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[4]	—	8	H'FFC1AC	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[5]	—	8	H'FFC1AD	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[6]	—	8	H'FFC1AE	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	8	H'FFC1AF	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1H	—	8	H'FFC1B0	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1L	—	8	H'FFC1B1	RCAN_0	16	2+nP φ/3+nP φ
	TIMESTAMP	—	16	H'FFC1B2	RCAN_0	16	2P φ/3P φ
MB_0[6].	CONTROL0H	—	16	H'FFC1C0	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL0L	—	16	H'FFC1C2	RCAN_0	16	2+nP φ/3+nP φ
	LAFMH	—	16	H'FFC1C4	RCAN_0	16	2+nP φ/3+nP φ
	LAFML	—	16	H'FFC1C6	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[0]	—	8	H'FFC1C8	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[1]	—	8	H'FFC1C9	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[2]	—	8	H'FFC1CA	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[3]	—	8	H'FFC1CB	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[4]	—	8	H'FFC1CC	RCAN_0	16	2+nP φ/3+nP φ

レジスタ名称	略称	ビット数	アドレス* ¹	モジュール	データ バス幅	アクセス* ² ステート数 (リード/ライト)	
MB_0[6].	MSG_DATA[5]	—	8	H'FFC1CD	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[6]	—	8	H'FFC1CE	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	8	H'FFC1CF	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1H	—	8	H'FFC1D0	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1L	—	8	H'FFC1D1	RCAN_0	16	2+nP φ/3+nP φ
	TIMESTAMP	—	16	H'FFC1D2	RCAN_0	16	2P φ/3P φ
MB_0[7].	CONTROL0H	—	16	H'FFC1E0	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL0L	—	16	H'FFC1E2	RCAN_0	16	2+nP φ/3+nP φ
	LAFMH	—	16	H'FFC1E4	RCAN_0	16	2+nP φ/3+nP φ
	LAFML	—	16	H'FFC1E6	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[0]	—	8	H'FFC1E8	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[1]	—	8	H'FFC1E9	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[2]	—	8	H'FFC1EA	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[3]	—	8	H'FFC1EB	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[4]	—	8	H'FFC1EC	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[5]	—	8	H'FFC1ED	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[6]	—	8	H'FFC1EE	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	8	H'FFC1EF	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1H	—	8	H'FFC1F0	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1L	—	8	H'FFC1F1	RCAN_0	16	2+nP φ/3+nP φ
	TIMESTAMP	—	16	H'FFC1F2	RCAN_0	16	2P φ/3P φ
	MB_0[8].	CONTROL0H	—	16	H'FFC200	RCAN_0	16
CONTROL0L		—	16	H'FFC202	RCAN_0	16	2+nP φ/3+nP φ
LAFMH		—	16	H'FFC204	RCAN_0	16	2+nP φ/3+nP φ
LAFML		—	16	H'FFC206	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[0]		—	8	H'FFC208	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[1]		—	8	H'FFC209	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[2]		—	8	H'FFC20A	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[3]		—	8	H'FFC20B	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[4]		—	8	H'FFC20C	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[5]		—	8	H'FFC20D	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[6]		—	8	H'FFC20E	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[7]		—	8	H'FFC20F	RCAN_0	16	2+nP φ/3+nP φ
CONTROL1H		—	8	H'FFC210	RCAN_0	16	2+nP φ/3+nP φ
CONTROL1L		—	8	H'FFC211	RCAN_0	16	2+nP φ/3+nP φ

レジスタ名称	略称	ビット数	アドレス* ¹	モジュール	データ バス幅	アクセス* ² ステート数 (リード/ライト)	
MB_0[8].	TIMESTAMP	—	16	H'FFC212	RCAN_0	16	2Pφ/3Pφ
MB_0[9].	CONTROL0H	—	16	H'FFC220	RCAN_0	16	2+nPφ/3+nPφ
	CONTROL0L	—	16	H'FFC222	RCAN_0	16	2+nPφ/3+nPφ
	LAFMH	—	16	H'FFC224	RCAN_0	16	2+nPφ/3+nPφ
	LAFML	—	16	H'FFC226	RCAN_0	16	2+nPφ/3+nPφ
	MSG_DATA[0]	—	8	H'FFC228	RCAN_0	16	2+nPφ/3+nPφ
	MSG_DATA[1]	—	8	H'FFC229	RCAN_0	16	2+nPφ/3+nPφ
	MSG_DATA[2]	—	8	H'FFC22A	RCAN_0	16	2+nPφ/3+nPφ
	MSG_DATA[3]	—	8	H'FFC22B	RCAN_0	16	2+nPφ/3+nPφ
	MSG_DATA[4]	—	8	H'FFC22C	RCAN_0	16	2+nPφ/3+nPφ
	MSG_DATA[5]	—	8	H'FFC22D	RCAN_0	16	2+nPφ/3+nPφ
	MSG_DATA[6]	—	8	H'FFC22E	RCAN_0	16	2+nPφ/3+nPφ
	MSG_DATA[7]	—	8	H'FFC22F	RCAN_0	16	2+nPφ/3+nPφ
	CONTROL1H	—	8	H'FFC230	RCAN_0	16	2+nPφ/3+nPφ
	CONTROL1L	—	8	H'FFC231	RCAN_0	16	2+nPφ/3+nPφ
	TIMESTAMP	—	16	H'FFC232	RCAN_0	16	2Pφ/3Pφ
MB_0[10].	CONTROL0H	—	16	H'FFC240	RCAN_0	16	2+nPφ/3+nPφ
	CONTROL0L	—	16	H'FFC242	RCAN_0	16	2+nPφ/3+nPφ
	LAFMH	—	16	H'FFC244	RCAN_0	16	2+nPφ/3+nPφ
	LAFML	—	16	H'FFC246	RCAN_0	16	2+nPφ/3+nPφ
	MSG_DATA[0]	—	8	H'FFC248	RCAN_0	16	2+nPφ/3+nPφ
	MSG_DATA[1]	—	8	H'FFC249	RCAN_0	16	2+nPφ/3+nPφ
	MSG_DATA[2]	—	8	H'FFC24A	RCAN_0	16	2+nPφ/3+nPφ
	MSG_DATA[3]	—	8	H'FFC24B	RCAN_0	16	2+nPφ/3+nPφ
	MSG_DATA[4]	—	8	H'FFC24C	RCAN_0	16	2+nPφ/3+nPφ
	MSG_DATA[5]	—	8	H'FFC24D	RCAN_0	16	2+nPφ/3+nPφ
	MSG_DATA[6]	—	8	H'FFC24E	RCAN_0	16	2+nPφ/3+nPφ
	MSG_DATA[7]	—	8	H'FFC24F	RCAN_0	16	2+nPφ/3+nPφ
	CONTROL1H	—	8	H'FFC250	RCAN_0	16	2+nPφ/3+nPφ
	CONTROL1L	—	8	H'FFC251	RCAN_0	16	2+nPφ/3+nPφ
	TIMESTAMP	—	16	H'FFC252	RCAN_0	16	2Pφ/3Pφ
MB_0[11].	CONTROL0H	—	16	H'FFC260	RCAN_0	16	2+nPφ/3+nPφ
	CONTROL0L	—	16	H'FFC262	RCAN_0	16	2+nPφ/3+nPφ
	LAFMH	—	16	H'FFC264	RCAN_0	16	2+nPφ/3+nPφ

レジスタ名称		略称	ビット数	アドレス* ¹	モジュール	データ バス幅	アクセス* ² ステート数 (リード/ライト)
MB_0[11].	LAFML	—	16	H'FFC266	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[0]	—	8	H'FFC268	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[1]	—	8	H'FFC269	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[2]	—	8	H'FFC26A	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[3]	—	8	H'FFC26B	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[4]	—	8	H'FFC26C	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[5]	—	8	H'FFC26D	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[6]	—	8	H'FFC26E	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	8	H'FFC26F	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1H	—	8	H'FFC270	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1L	—	8	H'FFC271	RCAN_0	16	2+nP φ/3+nP φ
	TIMESTAMP	—	16	H'FFC272	RCAN_0	16	2P φ/3P φ
MB_0[12].	CONTROL0H	—	16	H'FFC280	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL0L	—	16	H'FFC282	RCAN_0	16	2+nP φ/3+nP φ
	LAFMH	—	16	H'FFC284	RCAN_0	16	2+nP φ/3+nP φ
	LAFML	—	16	H'FFC286	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[0]	—	8	H'FFC288	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[1]	—	8	H'FFC289	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[2]	—	8	H'FFC28A	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[3]	—	8	H'FFC28B	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[4]	—	8	H'FFC28C	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[5]	—	8	H'FFC28D	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[6]	—	8	H'FFC28E	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	8	H'FFC28F	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1H	—	8	H'FFC290	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1L	—	8	H'FFC291	RCAN_0	16	2+nP φ/3+nP φ
	TIMESTAMP	—	16	H'FFC292	RCAN_0	16	2P φ/3P φ
	MB_0[13].	CONTROL0H	—	16	H'FFC2A0	RCAN_0	16
CONTROL0L		—	16	H'FFC2A2	RCAN_0	16	2+nP φ/3+nP φ
LAFMH		—	16	H'FFC2A4	RCAN_0	16	2+nP φ/3+nP φ
LAFML		—	16	H'FFC2A6	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[0]		—	8	H'FFC2A8	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[1]		—	8	H'FFC2A9	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[2]		—	8	H'FFC2AA	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[3]		—	8	H'FFC2AB	RCAN_0	16	2+nP φ/3+nP φ

レジスタ名称	略称	ビット数	アドレス* ¹	モジュール	データ バス幅	アクセス* ² ステート数 (リード/ライト)	
MB_0[13].	MSG_DATA[4]	—	8	H'FFC2AC	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[5]	—	8	H'FFC2AD	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[6]	—	8	H'FFC2AE	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	8	H'FFC2AF	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1H	—	8	H'FFC2B0	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1L	—	8	H'FFC2B1	RCAN_0	16	2+nP φ/3+nP φ
	TIMESTAMP	—	16	H'FFC2B2	RCAN_0	16	2P φ/3P φ
MB_0[14].	CONTROL0H	—	16	H'FFC2C0	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL0L	—	16	H'FFC2C2	RCAN_0	16	2+nP φ/3+nP φ
	LAFMH	—	16	H'FFC2C4	RCAN_0	16	2+nP φ/3+nP φ
	LAFML	—	16	H'FFC2C6	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[0]	—	8	H'FFC2C8	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[1]	—	8	H'FFC2C9	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[2]	—	8	H'FFC2CA	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[3]	—	8	H'FFC2CB	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[4]	—	8	H'FFC2CC	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[5]	—	8	H'FFC2CD	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[6]	—	8	H'FFC2CE	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	8	H'FFC2CF	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1H	—	8	H'FFC2D0	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1L	—	8	H'FFC2D1	RCAN_0	16	2+nP φ/3+nP φ
	TIMESTAMP	—	16	H'FFC2D2	RCAN_0	16	2P φ/3P φ
MB_0[15].	CONTROL0H	—	16	H'FFC2E0	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL0L	—	16	H'FFC2E2	RCAN_0	16	2+nP φ/3+nP φ
	LAFMH	—	16	H'FFC2E4	RCAN_0	16	2+nP φ/3+nP φ
	LAFML	—	16	H'FFC2E6	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[0]	—	8	H'FFC2E8	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[1]	—	8	H'FFC2E9	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[2]	—	8	H'FFC2EA	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[3]	—	8	H'FFC2EB	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[4]	—	8	H'FFC2EC	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[5]	—	8	H'FFC2ED	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[6]	—	8	H'FFC2EE	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	8	H'FFC2EF	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1H	—	8	H'FFC2F0	RCAN_0	16	2+nP φ/3+nP φ

レジスタ名称	略称	ビット数	アドレス* ¹	モジュール	データ バス幅	アクセス* ² ステート数 (リード/ライト)	
MB_0[15].	CONTROL1L	—	8	H'FFC2F1	RCAN_0	16	2+nP φ/3+nP φ
	TIMESTAMP	—	16	H'FFC2F2	RCAN_0	16	2P φ/3P φ
MB_0[16].	CONTROL0H	—	16	H'FFC300	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL0L	—	16	H'FFC302	RCAN_0	16	2+nP φ/3+nP φ
	LAFMH	—	16	H'FFC304	RCAN_0	16	2+nP φ/3+nP φ
	LAFML	—	16	H'FFC306	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[0]	—	8	H'FFC308	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[1]	—	8	H'FFC309	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[2]	—	8	H'FFC30A	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[3]	—	8	H'FFC30B	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[4]	—	8	H'FFC30C	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[5]	—	8	H'FFC30D	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[6]	—	8	H'FFC30E	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	8	H'FFC30F	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1H	—	8	H'FFC310	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1L	—	8	H'FFC311	RCAN_0	16	2+nP φ/3+nP φ
MB_0[17].	CONTROL0H	—	16	H'FFC320	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL0L	—	16	H'FFC322	RCAN_0	16	2+nP φ/3+nP φ
	LAFMH	—	16	H'FFC324	RCAN_0	16	2+nP φ/3+nP φ
	LAFML	—	16	H'FFC326	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[0]	—	8	H'FFC328	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[1]	—	8	H'FFC329	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[2]	—	8	H'FFC32A	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[3]	—	8	H'FFC32B	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[4]	—	8	H'FFC32C	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[5]	—	8	H'FFC32D	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[6]	—	8	H'FFC32E	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	8	H'FFC32F	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1H	—	8	H'FFC330	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1L	—	8	H'FFC331	RCAN_0	16	2+nP φ/3+nP φ
MB_0[18].	CONTROL0H	—	16	H'FFC340	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL0L	—	16	H'FFC342	RCAN_0	16	2+nP φ/3+nP φ
	LAFMH	—	16	H'FFC344	RCAN_0	16	2+nP φ/3+nP φ
	LAFML	—	16	H'FFC346	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[0]	—	8	H'FFC348	RCAN_0	16	2+nP φ/3+nP φ

レジスタ名称	略称	ビット数	アドレス* ¹	モジュール	データ バス幅	アクセス* ² ステート数 (リード/ライト)	
MB_0[18].	MSG_DATA[1]	—	8	H'FFC349	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[2]	—	8	H'FFC34A	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[3]	—	8	H'FFC34B	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[4]	—	8	H'FFC34C	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[5]	—	8	H'FFC34D	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[6]	—	8	H'FFC34E	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	8	H'FFC34F	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1H	—	8	H'FFC350	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1L	—	8	H'FFC351	RCAN_0	16	2+nP φ/3+nP φ
MB_0[19].	CONTROL0H	—	16	H'FFC360	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL0L	—	16	H'FFC362	RCAN_0	16	2+nP φ/3+nP φ
	LAFMH	—	16	H'FFC364	RCAN_0	16	2+nP φ/3+nP φ
	LAFML	—	16	H'FFC366	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[0]	—	8	H'FFC368	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[1]	—	8	H'FFC369	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[2]	—	8	H'FFC36A	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[3]	—	8	H'FFC36B	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[4]	—	8	H'FFC36C	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[5]	—	8	H'FFC36D	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[6]	—	8	H'FFC36E	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	8	H'FFC36F	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1H	—	8	H'FFC370	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1L	—	8	H'FFC371	RCAN_0	16	2+nP φ/3+nP φ
	MB_0[20].	CONTROL0H	—	16	H'FFC380	RCAN_0	16
CONTROL0L		—	16	H'FFC382	RCAN_0	16	2+nP φ/3+nP φ
LAFMH		—	16	H'FFC384	RCAN_0	16	2+nP φ/3+nP φ
LAFML		—	16	H'FFC386	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[0]		—	8	H'FFC388	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[1]		—	8	H'FFC389	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[2]		—	8	H'FFC38A	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[3]		—	8	H'FFC38B	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[4]		—	8	H'FFC38C	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[5]		—	8	H'FFC38D	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[6]		—	8	H'FFC38E	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[7]		—	8	H'FFC38F	RCAN_0	16	2+nP φ/3+nP φ

レジスタ名称	略称	ビット数	アドレス* ¹	モジュール	データ バス幅	アクセス* ² ステート数 (リード/ライト)	
MB_0[20].	CONTROL1H	—	8	H'FFC390	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1L	—	8	H'FFC391	RCAN_0	16	2+nP φ/3+nP φ
MB_0[21].	CONTROL0H	—	16	H'FFC3A0	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL0L	—	16	H'FFC3A2	RCAN_0	16	2+nP φ/3+nP φ
	LAFMH	—	16	H'FFC3A4	RCAN_0	16	2+nP φ/3+nP φ
	LAFML	—	16	H'FFC3A6	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[0]	—	8	H'FFC3A8	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[1]	—	8	H'FFC3A9	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[2]	—	8	H'FFC3AA	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[3]	—	8	H'FFC3AB	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[4]	—	8	H'FFC3AC	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[5]	—	8	H'FFC3AD	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[6]	—	8	H'FFC3AE	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	8	H'FFC3AF	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1H	—	8	H'FFC3B0	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1L	—	8	H'FFC3B1	RCAN_0	16	2+nP φ/3+nP φ
	MB_0[22].	CONTROL0H	—	16	H'FFC3C0	RCAN_0	16
CONTROL0L		—	16	H'FFC3C2	RCAN_0	16	2+nP φ/3+nP φ
LAFMH		—	16	H'FFC3C4	RCAN_0	16	2+nP φ/3+nP φ
LAFML		—	16	H'FFC3C6	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[0]		—	8	H'FFC3C8	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[1]		—	8	H'FFC3C9	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[2]		—	8	H'FFC3CA	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[3]		—	8	H'FFC3CB	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[4]		—	8	H'FFC3CC	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[5]		—	8	H'FFC3CD	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[6]		—	8	H'FFC3CE	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[7]		—	8	H'FFC3CF	RCAN_0	16	2+nP φ/3+nP φ
CONTROL1H		—	8	H'FFC3D0	RCAN_0	16	2+nP φ/3+nP φ
CONTROL1L		—	8	H'FFC3D1	RCAN_0	16	2+nP φ/3+nP φ
MB_0[23].		CONTROL0H	—	16	H'FFC3E0	RCAN_0	16
	CONTROL0L	—	16	H'FFC3E2	RCAN_0	16	2+nP φ/3+nP φ
	LAFMH	—	16	H'FFC3E4	RCAN_0	16	2+nP φ/3+nP φ
	LAFML	—	16	H'FFC3E6	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[0]	—	8	H'FFC3E8	RCAN_0	16	2+nP φ/3+nP φ

レジスタ名称	略称	ビット数	アドレス* ¹	モジュール	データ バス幅	アクセス* ² ステート数 (リード/ライト)	
MB_0[23].	MSG_DATA[1]	—	8	H'FFC3E9	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[2]	—	8	H'FFC3EA	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[3]	—	8	H'FFC3EB	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[4]	—	8	H'FFC3EC	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[5]	—	8	H'FFC3ED	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[6]	—	8	H'FFC3EE	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	8	H'FFC3EF	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1H	—	8	H'FFC3F0	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1L	—	8	H'FFC3F1	RCAN_0	16	2+nP φ/3+nP φ
MB_0[24].	CONTROL0H	—	16	H'FFC400	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL0L	—	16	H'FFC402	RCAN_0	16	2+nP φ/3+nP φ
	LAFMH	—	16	H'FFC404	RCAN_0	16	2+nP φ/3+nP φ
	LAFML	—	16	H'FFC406	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[0]	—	8	H'FFC408	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[1]	—	8	H'FFC409	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[2]	—	8	H'FFC40A	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[3]	—	8	H'FFC40B	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[4]	—	8	H'FFC40C	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[5]	—	8	H'FFC40D	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[6]	—	8	H'FFC40E	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	8	H'FFC40F	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1H	—	8	H'FFC410	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1L	—	8	H'FFC411	RCAN_0	16	2+nP φ/3+nP φ
	TTT	—	16	H'FFC414	RCAN_0	16	2P φ/3P φ
	TTCONTROL	—	16	H'FFC416	RCAN_0	16	2P φ/3P φ
	MB_0[25].	CONTROL0H	—	16	H'FFC420	RCAN_0	16
CONTROL0L		—	16	H'FFC422	RCAN_0	16	2+nP φ/3+nP φ
LAFMH		—	16	H'FFC424	RCAN_0	16	2+nP φ/3+nP φ
LAFML		—	16	H'FFC426	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[0]		—	8	H'FFC428	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[1]		—	8	H'FFC429	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[2]		—	8	H'FFC42A	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[3]		—	8	H'FFC42B	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[4]		—	8	H'FFC42C	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[5]		—	8	H'FFC42D	RCAN_0	16	2+nP φ/3+nP φ

レジスタ名称	略称	ビット数	アドレス* ¹	モジュール	データ バス幅	アクセス* ² ステート数 (リード/ライト)	
MB_0[25].	MSG_DATA[6]	—	8	H'FFC42E	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	8	H'FFC42F	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1H	—	8	H'FFC430	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1L	—	8	H'FFC431	RCAN_0	16	2+nP φ/3+nP φ
	TTT	—	16	H'FFC434	RCAN_0	16	2P φ/3P φ
	TTCONTROL	—	16	H'FFC436	RCAN_0	16	2P φ/3P φ
MB_0[26].	CONTROL0H	—	16	H'FFC440	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL0L	—	16	H'FFC442	RCAN_0	16	2+nP φ/3+nP φ
	LAFMH	—	16	H'FFC444	RCAN_0	16	2+nP φ/3+nP φ
	LAFML	—	16	H'FFC446	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[0]	—	8	H'FFC448	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[1]	—	8	H'FFC449	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[2]	—	8	H'FFC44A	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[3]	—	8	H'FFC44B	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[4]	—	8	H'FFC44C	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[5]	—	8	H'FFC44D	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[6]	—	8	H'FFC44E	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	8	H'FFC44F	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1H	—	8	H'FFC450	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1L	—	8	H'FFC451	RCAN_0	16	2+nP φ/3+nP φ
	TTT	—	16	H'FFC454	RCAN_0	16	2P φ/3P φ
	TTCONTROL	—	16	H'FFC456	RCAN_0	16	2P φ/3P φ
MB_0[27].	CONTROL0H	—	16	H'FFC460	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL0L	—	16	H'FFC462	RCAN_0	16	2+nP φ/3+nP φ
	LAFMH	—	16	H'FFC464	RCAN_0	16	2+nP φ/3+nP φ
	LAFML	—	16	H'FFC466	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[0]	—	8	H'FFC468	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[1]	—	8	H'FFC469	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[2]	—	8	H'FFC46A	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[3]	—	8	H'FFC46B	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[4]	—	8	H'FFC46C	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[5]	—	8	H'FFC46D	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[6]	—	8	H'FFC46E	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	8	H'FFC46F	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1H	—	8	H'FFC470	RCAN_0	16	2+nP φ/3+nP φ

レジスタ名称	略称	ビット数	アドレス* ¹	モジュール	データ バス幅	アクセス* ² ステート数 (リード/ライト)	
MB_0[27].	CONTROL1L	—	8	H'FFC471	RCAN_0	16	2+nP φ/3+nP φ
	TTT	—	16	H'FFC474	RCAN_0	16	2P φ/3P φ
	TTCONTROL	—	16	H'FFC476	RCAN_0	16	2P φ/3P φ
MB_0[28].	CONTROL0H	—	16	H'FFC480	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL0L	—	16	H'FFC482	RCAN_0	16	2+nP φ/3+nP φ
	LAFMH	—	16	H'FFC484	RCAN_0	16	2+nP φ/3+nP φ
	LAFML	—	16	H'FFC486	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[0]	—	8	H'FFC488	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[1]	—	8	H'FFC489	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[2]	—	8	H'FFC48A	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[3]	—	8	H'FFC48B	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[4]	—	8	H'FFC48C	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[5]	—	8	H'FFC48D	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[6]	—	8	H'FFC48E	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	8	H'FFC48F	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1H	—	8	H'FFC490	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1L	—	8	H'FFC491	RCAN_0	16	2+nP φ/3+nP φ
	TTT	—	16	H'FFC494	RCAN_0	16	2P φ/3P φ
	TTCONTROL	—	16	H'FFC496	RCAN_0	16	2P φ/3P φ
	MB_0[29].	CONTROL0H	—	16	H'FFC4A0	RCAN_0	16
CONTROL0L		—	16	H'FFC4A2	RCAN_0	16	2+nP φ/3+nP φ
LAFMH		—	16	H'FFC4A4	RCAN_0	16	2+nP φ/3+nP φ
LAFML		—	16	H'FFC4A6	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[0]		—	8	H'FFC4A8	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[1]		—	8	H'FFC4A9	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[2]		—	8	H'FFC4AA	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[3]		—	8	H'FFC4AB	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[4]		—	8	H'FFC4AC	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[5]		—	8	H'FFC4AD	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[6]		—	8	H'FFC4AE	RCAN_0	16	2+nP φ/3+nP φ
MSG_DATA[7]		—	8	H'FFC4AF	RCAN_0	16	2+nP φ/3+nP φ
CONTROL1H		—	8	H'FFC4B0	RCAN_0	16	2+nP φ/3+nP φ
CONTROL1L		—	8	H'FFC4B1	RCAN_0	16	2+nP φ/3+nP φ
TTT		—	16	H'FFC4B4	RCAN_0	16	2P φ/3P φ
TTCONTROL		—	16	H'FFC4B6	RCAN_0	16	2P φ/3P φ

レジスタ名称	略称	ビット数	アドレス* ¹	モジュール	データ バス幅	アクセス* ² ステート数 (リード/ライト)
MB_0[30].	CONTROL0H	—	H'FFC4C0	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL0L	—	H'FFC4C2	RCAN_0	16	2+nP φ/3+nP φ
	LAFMH	—	H'FFC4C4	RCAN_0	16	2+nP φ/3+nP φ
	LAFML	—	H'FFC4C6	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[0]	—	H'FFC4C8	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[1]	—	H'FFC4C9	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[2]	—	H'FFC4CA	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[3]	—	H'FFC4CB	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[4]	—	H'FFC4CC	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[5]	—	H'FFC4CD	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[6]	—	H'FFC4CE	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	H'FFC4CF	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1H	—	H'FFC4D0	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1L	—	H'FFC4D1	RCAN_0	16	2+nP φ/3+nP φ
	TIMSTAMP	—	H'FFC4D2	RCAN_0	16	2P φ/3P φ
	TTT	—	H'FFC4D4	RCAN_0	16	2P φ/3P φ
MB_0[31].	CONTROL0H	—	H'FFC4E0	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL0L	—	H'FFC4E2	RCAN_0	16	2+nP φ/3+nP φ
	LAFMH	—	H'FFC4E4	RCAN_0	16	2+nP φ/3+nP φ
	LAFML	—	H'FFC4E6	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[0]	—	H'FFC4E8	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[1]	—	H'FFC4E9	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[2]	—	H'FFC4EA	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[3]	—	H'FFC4EB	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[4]	—	H'FFC4EC	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[5]	—	H'FFC4ED	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[6]	—	H'FFC4EE	RCAN_0	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	H'FFC4EF	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1H	—	H'FFC4F0	RCAN_0	16	2+nP φ/3+nP φ
	CONTROL1L	—	H'FFC4F1	RCAN_0	16	2+nP φ/3+nP φ
	TIMSTAMP	—	H'FFC4F2	RCAN_0	16	2P φ/3P φ
	RCAN-TL1 モニタレジスタ_0	RCANMON_0	8	H'FFC800	RCAN_0	16
RCAN-TL1 モニタレジスタ_1	RCANMON_1	8	H'FFC801	RCAN_1	16	2P φ/2P φ
マスタコントロールレジスタ_1	MCR_1	16	H'FFD000	RCAN_1	16	2P φ/3P φ
ジェネラルステータスレジスタ_1	GSR_1	16	H'FFD002	RCAN_1	16	2P φ/3P φ
ビットコンフィギュレーションレジスタ 1_1	BCR1_1	16	H'FFD004	RCAN_1	16	2P φ/3P φ

レジスタ名称	略称	ビット数	アドレス* ¹	モジュール	データ バス幅	アクセス* ² ステート数 (リード/ライト)
ビットコンフィギュレーションレジスタ 0_1	BCR0_1	16	H'FFD006	RCAN_1	16	2Pφ/3Pφ
インタラプトリクエストレジスタ_1	IRR_1	16	H'FFD008	RCAN_1	16	2Pφ/3Pφ
インタラプトマスクレジスタ_1	IMR_1	16	H'FFD00A	RCAN_1	16	2Pφ/3Pφ
送信エラーカウンタ_1	TEC_1	16	H'FFD00C	RCAN_1	16	2Pφ/3Pφ
受信エラーカウンタ_1	REC_1					
送信待ちレジスタ 1_1	TXPR1_1	16	H'FFD020	RCAN_1	32	4Pφ/6Pφ
送信待ちレジスタ 0_1	TXPR0_1	16				
送信キャンセルレジスタ 1_1	TXCR1_1	16	H'FFD028	RCAN_1	16	2Pφ/3Pφ
送信キャンセルレジスタ 0_1	TXCR0_1	16	H'FFD02A	RCAN_1	16	2Pφ/3Pφ
送信アクノリッジレジスタ 1_1	TXACK1_1	16	H'FFD030	RCAN_1	16	2Pφ/3Pφ
送信アクノリッジレジスタ 0_1	TXACK0_1	16	H'FFD032	RCAN_1	16	2Pφ/3Pφ
アボートアクノリッジレジスタ 1_1	ABACK1_1	16	H'FFD038	RCAN_1	16	2Pφ/3Pφ
アボートアクノリッジレジスタ 0_1	ABACK0_1	16	H'FFD03A	RCAN_1	16	2Pφ/3Pφ
データフレーム受信完了レジスタ 1_1	RXPR1_1	16	H'FFD040	RCAN_1	16	2Pφ/3Pφ
データフレーム受信完了レジスタ 0_1	RXPR0_1	16	H'FFD042	RCAN_1	16	2Pφ/3Pφ
リモートフレーム受信完了レジスタ 1_1	RFPR1_1	16	H'FFD048	RCAN_1	16	2Pφ/3Pφ
リモートフレーム受信完了レジスタ 0_1	RFPR0_1	16	H'FFD04A	RCAN_1	16	2Pφ/3Pφ
メールボックスインタラプトマスクレジスタ 1_1	MBIMR1_1	16	H'FFD050	RCAN_1	16	2Pφ/3Pφ
メールボックスインタラプトマスクレジスタ 0_1	MBIMR0_1	16	H'FFD052	RCAN_1	16	2Pφ/3Pφ
未読メッセージステータスレジスタ 1_1	UMSR1_1	16	H'FFD058	RCAN_1	16	2Pφ/3Pφ
未読メッセージステータスレジスタ 0_1	UMSR0_1	16	H'FFD05A	RCAN_1	16	2Pφ/3Pφ
タイムトリガコントロールレジスタ 0_1	TTCR0_1	16	H'FFD080	RCAN_1	16	2Pφ/3Pφ
サイクルマキシマム/Tx_Enable_Window レジスタ_1	CMAX_TEW_1	16	H'FFD084	RCAN_1	16	2Pφ/3Pφ
リファレンストリガオフセットレジスタ_1	RFTR0FF_1	16	H'FFD086	RCAN_1	16	2Pφ/3Pφ
タイマステータスレジスタ_1	TSR_1	16	H'FFD088	RCAN_1	16	2Pφ/3Pφ
サイクルカウンタレジスタ_1	CCR_1	16	H'FFD08A	RCAN_1	16	2Pφ/3Pφ
タイマカウンタレジスタ_1	TCNTR_1	16	H'FFD08C	RCAN_1	16	2Pφ/3Pφ
サイクルタイムレジスタ_1	CYCTR_1	16	H'FFD090	RCAN_1	16	2Pφ/3Pφ
リファレンスマークレジスタ_1	RFMK_1	16	H'FFD094	RCAN_1	16	2Pφ/3Pφ
タイマコンペアマッチレジスタ 0_1	TCMR0_1	16	H'FFD098	RCAN_1	16	2Pφ/3Pφ
タイマコンペアマッチレジスタ 1_1	TCMR1_1	16	H'FFD09C	RCAN_1	16	2Pφ/3Pφ
タイマコンペアマッチレジスタ 2_1	TCMR2_1	16	H'FFD0A0	RCAN_1	16	2Pφ/3Pφ
送信トリガタイムセレクトレジスタ_1	TTTSEL_1	16	H'FFD0A4	RCAN_1	16	2Pφ/3Pφ

レジスタ名称	略称	ビット数	アドレス* ¹	モジュール	データ バス幅	アクセス* ² ステート数 (リード/ライト)	
MB_1[0].	CONTROL0H	—	H'FFD100	RCAN_1	16	2+nP φ/3+nP φ	
	CONTROL0L	—	H'FFD102	RCAN_1	16	2+nP φ/3+nP φ	
	LAFMH	—	H'FFD104	RCAN_1	16	2+nP φ/3+nP φ	
	LAFML	—	H'FFD106	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[0]	—	H'FFD108	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[1]	—	H'FFD109	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[2]	—	H'FFD10A	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[3]	—	H'FFD10B	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[4]	—	H'FFD10C	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[5]	—	H'FFD10D	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[6]	—	H'FFD10E	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[7]	—	H'FFD10F	RCAN_1	16	2+nP φ/3+nP φ	
	CONTROL1H	—	H'FFD110	RCAN_1	16	2+nP φ/3+nP φ	
	CONTROL1L	—	H'FFD111	RCAN_1	16	2+nP φ/3+nP φ	
TIMESTAMP	—	16	H'FFD112	RCAN_1	16	2P φ/3P φ	
MB_1[1].	CONTROL0H	—	H'FFD120	RCAN_1	16	2+nP φ/3+nP φ	
	CONTROL0L	—	H'FFD122	RCAN_1	16	2+nP φ/3+nP φ	
	LAFMH	—	H'FFD124	RCAN_1	16	2+nP φ/3+nP φ	
	LAFML	—	H'FFD126	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[0]	—	H'FFD128	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[1]	—	H'FFD129	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[2]	—	H'FFD12A	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[3]	—	H'FFD12B	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[4]	—	H'FFD12C	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[5]	—	H'FFD12D	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[6]	—	H'FFD12E	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[7]	—	H'FFD12F	RCAN_1	16	2+nP φ/3+nP φ	
	CONTROL1H	—	8	H'FFD130	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1L	—	8	H'FFD131	RCAN_1	16	2+nP φ/3+nP φ
TIMESTAMP	—	16	H'FFD132	RCAN_1	16	2P φ/3P φ	
MB_1[2].	CONTROL0H	—	16	H'FFD140	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL0L	—	16	H'FFD142	RCAN_1	16	2+nP φ/3+nP φ
	LAFMH	—	16	H'FFD144	RCAN_1	16	2+nP φ/3+nP φ
	LAFML	—	16	H'FFD146	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[0]	—	8	H'FFD148	RCAN_1	16	2+nP φ/3+nP φ

レジスタ名称	略称	ビット数	アドレス* ¹	モジュール	データ バス幅	アクセス* ² ステート数 (リード/ライト)		
MB_1[2].	MSG_DATA[1]	—	8	H'FFD149	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[2]	—	8	H'FFD14A	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[3]	—	8	H'FFD14B	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[4]	—	8	H'FFD14C	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[5]	—	8	H'FFD14D	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[6]	—	8	H'FFD14E	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[7]	—	8	H'FFD14F	RCAN_1	16	2+nP φ/3+nP φ	
	CONTROL1H	—	8	H'FFD150	RCAN_1	16	2+nP φ/3+nP φ	
	CONTROL1L	—	8	H'FFD151	RCAN_1	16	2+nP φ/3+nP φ	
	TIMESTAMP	—	16	H'FFD152	RCAN_1	16	2P φ/3P φ	
MB_1[3].	CONTROL0H	—	16	H'FFD160	RCAN_1	16	2+nP φ/3+nP φ	
	CONTROL0L	—	16	H'FFD162	RCAN_1	16	2+nP φ/3+nP φ	
	LAFMH	—	16	H'FFD164	RCAN_1	16	2+nP φ/3+nP φ	
	LAFML	—	16	H'FFD166	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[0]	—	8	H'FFD168	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[1]	—	8	H'FFD169	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[2]	—	8	H'FFD16A	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[3]	—	8	H'FFD16B	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[4]	—	8	H'FFD16C	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[5]	—	8	H'FFD16D	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[6]	—	8	H'FFD16E	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[7]	—	8	H'FFD16F	RCAN_1	16	2+nP φ/3+nP φ	
	CONTROL1H	—	8	H'FFD170	RCAN_1	16	2+nP φ/3+nP φ	
	CONTROL1L	—	8	H'FFD171	RCAN_1	16	2+nP φ/3+nP φ	
	TIMESTAMP	—	16	H'FFD172	RCAN_1	16	2P φ/3P φ	
	MB_1[4].	CONTROL0H	—	16	H'FFD180	RCAN_1	16	2+nP φ/3+nP φ
		CONTROL0L	—	16	H'FFD182	RCAN_1	16	2+nP φ/3+nP φ
LAFMH		—	16	H'FFD184	RCAN_1	16	2+nP φ/3+nP φ	
LAFML		—	16	H'FFD186	RCAN_1	16	2+nP φ/3+nP φ	
MSG_DATA[0]		—	8	H'FFD188	RCAN_1	16	2+nP φ/3+nP φ	
MSG_DATA[1]		—	8	H'FFD189	RCAN_1	16	2+nP φ/3+nP φ	
MSG_DATA[2]		—	8	H'FFD18A	RCAN_1	16	2+nP φ/3+nP φ	
MSG_DATA[3]		—	8	H'FFD18B	RCAN_1	16	2+nP φ/3+nP φ	
MSG_DATA[4]		—	8	H'FFD18C	RCAN_1	16	2+nP φ/3+nP φ	
MSG_DATA[5]		—	8	H'FFD18D	RCAN_1	16	2+nP φ/3+nP φ	

レジスタ名称	略称	ビット数	アドレス* ¹	モジュール	データ バス幅	アクセス* ² ステート数 (リード/ライト)	
MB_1[4].	MSG_DATA[6]	—	8	H'FFD18E	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	8	H'FFD18F	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1H	—	8	H'FFD190	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1L	—	8	H'FFD191	RCAN_1	16	2+nP φ/3+nP φ
	TIMESTAMP	—	16	H'FFD192	RCAN_1	16	2P φ/3P φ
MB_1[5].	CONTROL0H	—	16	H'FFD1A0	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL0L	—	16	H'FFD1A2	RCAN_1	16	2+nP φ/3+nP φ
	LAFMH	—	16	H'FFD1A4	RCAN_1	16	2+nP φ/3+nP φ
	LAFML	—	16	H'FFD1A6	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[0]	—	8	H'FFD1A8	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[1]	—	8	H'FFD1A9	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[2]	—	8	H'FFD1AA	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[3]	—	8	H'FFD1AB	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[4]	—	8	H'FFD1AC	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[5]	—	8	H'FFD1AD	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[6]	—	8	H'FFD1AE	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	8	H'FFD1AF	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1H	—	8	H'FFD1B0	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1L	—	8	H'FFD1B1	RCAN_1	16	2+nP φ/3+nP φ
	TIMESTAMP	—	16	H'FFD1B2	RCAN_1	16	2P φ/3P φ
MB_1[6].	CONTROL0H	—	16	H'FFD1C0	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL0L	—	16	H'FFD1C2	RCAN_1	16	2+nP φ/3+nP φ
	LAFMH	—	16	H'FFD1C4	RCAN_1	16	2+nP φ/3+nP φ
	LAFML	—	16	H'FFD1C6	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[0]	—	8	H'FFD1C8	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[1]	—	8	H'FFD1C9	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[2]	—	8	H'FFD1CA	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[3]	—	8	H'FFD1CB	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[4]	—	8	H'FFD1CC	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[5]	—	8	H'FFD1CD	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[6]	—	8	H'FFD1CE	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	8	H'FFD1CF	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1H	—	8	H'FFD1D0	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1L	—	8	H'FFD1D1	RCAN_1	16	2+nP φ/3+nP φ
	TIMESTAMP	—	16	H'FFD1D2	RCAN_1	16	2P φ/3P φ

レジスタ名称	略称	ビット数	アドレス* ¹	モジュール	データ バス幅	アクセス* ² ステート数 (リード/ライト)
MB_1[7].	CONTROL0H	—	H'FFD1E0	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL0L	—	H'FFD1E2	RCAN_1	16	2+nP φ/3+nP φ
	LAFMH	—	H'FFD1E4	RCAN_1	16	2+nP φ/3+nP φ
	LAFML	—	H'FFD1E6	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[0]	—	H'FFD1E8	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[1]	—	H'FFD1E9	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[2]	—	H'FFD1EA	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[3]	—	H'FFD1EB	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[4]	—	H'FFD1EC	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[5]	—	H'FFD1ED	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[6]	—	H'FFD1EE	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	H'FFD1EF	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1H	—	H'FFD1F0	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1L	—	H'FFD1F1	RCAN_1	16	2+nP φ/3+nP φ
TIMESTAMP	—	16	H'FFD1F2	RCAN_1	16	2P φ/3P φ
MB_1[8].	CONTROL0H	—	H'FFD200	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL0L	—	H'FFD202	RCAN_1	16	2+nP φ/3+nP φ
	LAFMH	—	H'FFD204	RCAN_1	16	2+nP φ/3+nP φ
	LAFML	—	H'FFD206	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[0]	—	H'FFD208	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[1]	—	H'FFD209	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[2]	—	H'FFD20A	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[3]	—	H'FFD20B	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[4]	—	H'FFD20C	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[3]	—	H'FFD20D	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[6]	—	H'FFD20E	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	H'FFD20F	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1H	—	H'FFD210	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL0H	—	H'FFD211	RCAN_1	16	2+nP φ/3+nP φ
TIMESTAMP	—	16	H'FFD212	RCAN_1	16	2P φ/3P φ
MB_1[9].	CONTROL0H	—	H'FFD220	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL0L	—	H'FFD222	RCAN_1	16	2+nP φ/3+nP φ
	LAFMH	—	H'FFD224	RCAN_1	16	2+nP φ/3+nP φ
	LAFML	—	H'FFD226	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[0]	—	H'FFD228	RCAN_1	16	2+nP φ/3+nP φ

レジスタ名称	略称	ビット数	アドレス* ¹	モジュール	データ バス幅	アクセス* ² ステート数 (リード/ライト)	
MB_1[9].	MSG_DATA[1]	—	8	H'FFD229	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[2]	—	8	H'FFD22A	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[3]	—	8	H'FFD22B	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[4]	—	8	H'FFD22C	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[5]	—	8	H'FFD22D	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[6]	—	8	H'FFD22E	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	8	H'FFD22F	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1H	—	8	H'FFD230	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1L	—	8	H'FFD231	RCAN_1	16	2+nP φ/3+nP φ
	TIMESTAMP	—	16	H'FFD232	RCAN_1	16	2P φ/3P φ
MB_1[10].	CONTROL0H	—	16	H'FFD240	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL0L	—	16	H'FFD242	RCAN_1	16	2+nP φ/3+nP φ
	LAFMH	—	16	H'FFD244	RCAN_1	16	2+nP φ/3+nP φ
	LAFML	—	16	H'FFD246	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[0]	—	8	H'FFD248	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[1]	—	8	H'FFD249	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[2]	—	8	H'FFD24A	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[3]	—	8	H'FFD24B	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[4]	—	8	H'FFD24C	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[5]	—	8	H'FFD24D	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[6]	—	8	H'FFD24E	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	8	H'FFD24F	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1H	—	8	H'FFD250	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1L	—	8	H'FFD251	RCAN_1	16	2+nP φ/3+nP φ
	TIMESTAMP	—	16	H'FFD252	RCAN_1	16	2P φ/3P φ
	MB_1[11].	CONTROL0H	—	16	H'FFD260	RCAN_1	16
CONTROL0L		—	16	H'FFD262	RCAN_1	16	2+nP φ/3+nP φ
LAFMH		—	16	H'FFD264	RCAN_1	16	2+nP φ/3+nP φ
LAFML		—	16	H'FFD266	RCAN_1	16	2+nP φ/3+nP φ
MSG_DATA[0]		—	8	H'FFD268	RCAN_1	16	2+nP φ/3+nP φ
MSG_DATA[1]		—	8	H'FFD269	RCAN_1	16	2+nP φ/3+nP φ
MSG_DATA[2]		—	8	H'FFD26A	RCAN_1	16	2+nP φ/3+nP φ
MSG_DATA[3]		—	8	H'FFD26B	RCAN_1	16	2+nP φ/3+nP φ
MSG_DATA[4]		—	8	H'FFD26C	RCAN_1	16	2+nP φ/3+nP φ
MSG_DATA[5]		—	8	H'FFD26D	RCAN_1	16	2+nP φ/3+nP φ

レジスタ名称	略称	ビット数	アドレス* ¹	モジュール	データ バス幅	アクセス* ² ステート数 (リード/ライト)	
MB_1[11].	MSG_DATA[6]	—	8	H'FFD26E	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	8	H'FFD26F	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1H	—	8	H'FFD270	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1L	—	8	H'FFD271	RCAN_1	16	2+nP φ/3+nP φ
	TIMESTAMP	—	16	H'FFD272	RCAN_1	16	2P φ/3P φ
MB_1[12].	CONTROL0H	—	16	H'FFD280	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL0L	—	16	H'FFD282	RCAN_1	16	2+nP φ/3+nP φ
	LAFMH	—	16	H'FFD284	RCAN_1	16	2+nP φ/3+nP φ
	LAFML	—	16	H'FFD286	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[0]	—	8	H'FFD288	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[1]	—	8	H'FFD289	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[2]	—	8	H'FFD28A	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[3]	—	8	H'FFD28B	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[4]	—	8	H'FFD28C	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[5]	—	8	H'FFD28D	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[6]	—	8	H'FFD28E	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	8	H'FFD28F	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1H	—	8	H'FFD290	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1L	—	8	H'FFD291	RCAN_1	16	2+nP φ/3+nP φ
	TIMESTAMP	—	16	H'FFD292	RCAN_1	16	2P φ/3P φ
MB_1[13].	CONTROL0H	—	16	H'FFD2A0	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL0L	—	16	H'FFD2A2	RCAN_1	16	2+nP φ/3+nP φ
	LAFMH	—	16	H'FFD2A4	RCAN_1	16	2+nP φ/3+nP φ
	LAFML	—	16	H'FFD2A6	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[0]	—	8	H'FFD2A8	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[1]	—	8	H'FFD2A9	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[2]	—	8	H'FFD2AA	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[3]	—	8	H'FFD2AB	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[4]	—	8	H'FFD2AC	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[5]	—	8	H'FFD2AD	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[6]	—	8	H'FFD2AE	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	8	H'FFD2AF	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1H	—	8	H'FFD2B0	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1L	—	8	H'FFD2B1	RCAN_1	16	2+nP φ/3+nP φ
	TIMESTAMP	—	16	H'FFD2B2	RCAN_1	16	2P φ/3P φ

レジスタ名称	略称	ビット数	アドレス* ¹	モジュール	データ バス幅	アクセス* ² ステート数 (リード/ライト)	
MB_1[14].	CONTROL0H	—	16	H'FFD2C0	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL0L	—	16	H'FFD2C2	RCAN_1	16	2+nP φ/3+nP φ
	LAFMH	—	16	H'FFD2C4	RCAN_1	16	2+nP φ/3+nP φ
	LAFML	—	16	H'FFD2C6	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[0]	—	8	H'FFD2C8	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[1]	—	8	H'FFD2C9	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[2]	—	8	H'FFD2CA	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[3]	—	8	H'FFD2CB	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[4]	—	8	H'FFD2CC	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[5]	—	8	H'FFD2CD	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[6]	—	8	H'FFD2CE	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	8	H'FFD2CF	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1H	—	8	H'FFD2D0	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1L	—	8	H'FFD2D1	RCAN_1	16	2+nP φ/3+nP φ
TIMESTAMP	—	16	H'FFD2D2	RCAN_1	16	2P φ/3P φ	
MB_1[15].	CONTROL0H	—	16	H'FFD2E0	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL0L	—	16	H'FFD2E2	RCAN_1	16	2+nP φ/3+nP φ
	LAFMH	—	16	H'FFD2E4	RCAN_1	16	2+nP φ/3+nP φ
	LAFML	—	16	H'FFD2E6	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[0]	—	8	H'FFD2E8	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[1]	—	8	H'FFD2E9	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[2]	—	8	H'FFD2EA	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[3]	—	8	H'FFD2EB	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[4]	—	8	H'FFD2EC	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[5]	—	8	H'FFD2ED	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[6]	—	8	H'FFD2EE	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	8	H'FFD2EF	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1H	—	8	H'FFD2F0	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1L	—	8	H'FFD2F1	RCAN_1	16	2+nP φ/3+nP φ
TIMESTAMP	—	16	H'FFD2F2	RCAN_1	16	2P φ/3P φ	
MB_1[16].	CONTROL0H	—	16	H'FFD300	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL0L	—	16	H'FFD302	RCAN_1	16	2+nP φ/3+nP φ
	LAFMH	—	16	H'FFD304	RCAN_1	16	2+nP φ/3+nP φ
	LAFML	—	16	H'FFD306	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[0]	—	8	H'FFD308	RCAN_1	16	2+nP φ/3+nP φ

レジスタ名称	略称	ビット数	アドレス* ¹	モジュール	データ バス幅	アクセス* ² ステート数 (リード/ライト)	
MB_1[16].	MSG_DATA[1]	—	8	H'FFD309	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[2]	—	8	H'FFD30A	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[3]	—	8	H'FFD30B	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[4]	—	8	H'FFD30C	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[5]	—	8	H'FFD30D	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[6]	—	8	H'FFD30E	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	8	H'FFD30F	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1H	—	8	H'FFD310	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1L	—	8	H'FFD311	RCAN_1	16	2+nP φ/3+nP φ
MB_1[17].	CONTROL0H	—	16	H'FFD320	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL0L	—	16	H'FFD322	RCAN_1	16	2+nP φ/3+nP φ
	LAFMH	—	16	H'FFD324	RCAN_1	16	2+nP φ/3+nP φ
	LAFML	—	16	H'FFD326	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[0]	—	8	H'FFD328	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[1]	—	8	H'FFD329	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[2]	—	8	H'FFD32A	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[3]	—	8	H'FFD32B	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[4]	—	8	H'FFD32C	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[5]	—	8	H'FFD32D	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[6]	—	8	H'FFD32E	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	8	H'FFD32F	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1H	—	8	H'FFD330	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1L	—	8	H'FFD331	RCAN_1	16	2+nP φ/3+nP φ
	MB_1[18].	CONTROL0H	—	16	H'FFD340	RCAN_1	16
CONTROL0L		—	16	H'FFD342	RCAN_1	16	2+nP φ/3+nP φ
LAFMH		—	16	H'FFD344	RCAN_1	16	2+nP φ/3+nP φ
LAFML		—	16	H'FFD346	RCAN_1	16	2+nP φ/3+nP φ
MSG_DATA[0]		—	8	H'FFD348	RCAN_1	16	2+nP φ/3+nP φ
MSG_DATA[1]		—	8	H'FFD349	RCAN_1	16	2+nP φ/3+nP φ
MSG_DATA[2]		—	8	H'FFD34A	RCAN_1	16	2+nP φ/3+nP φ
MSG_DATA[3]		—	8	H'FFD34B	RCAN_1	16	2+nP φ/3+nP φ
MSG_DATA[4]		—	8	H'FFD34C	RCAN_1	16	2+nP φ/3+nP φ
MSG_DATA[5]		—	8	H'FFD34D	RCAN_1	16	2+nP φ/3+nP φ
MSG_DATA[6]		—	8	H'FFD34E	RCAN_1	16	2+nP φ/3+nP φ
MSG_DATA[7]		—	8	H'FFD34F	RCAN_1	16	2+nP φ/3+nP φ

レジスタ名称	略称	ビット数	アドレス* ¹	モジュール	データ バス幅	アクセス* ² ステート数 (リード/ライト)	
MB_1[18].	CONTROL1H	—	8	H'FFD350	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1L	—	8	H'FFD351	RCAN_1	16	2+nP φ/3+nP φ
MB_1[19].	CONTROL0H	—	16	H'FFD360	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL0L	—	16	H'FFD362	RCAN_1	16	2+nP φ/3+nP φ
	LAFMH	—	16	H'FFD364	RCAN_1	16	2+nP φ/3+nP φ
	LAFML	—	16	H'FFD366	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[0]	—	8	H'FFD368	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[1]	—	8	H'FFD369	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[2]	—	8	H'FFD36A	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[3]	—	8	H'FFD36B	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[4]	—	8	H'FFD36C	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[5]	—	8	H'FFD36D	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[6]	—	8	H'FFD36E	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	8	H'FFD36F	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1H	—	8	H'FFD370	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1L	—	8	H'FFD371	RCAN_1	16	2+nP φ/3+nP φ
	MB_1[20].	CONTROL0H	—	16	H'FFD380	RCAN_1	16
CONTROL0L		—	16	H'FFD382	RCAN_1	16	2+nP φ/3+nP φ
LAFMH		—	16	H'FFD384	RCAN_1	16	2+nP φ/3+nP φ
LAFML		—	16	H'FFD386	RCAN_1	16	2+nP φ/3+nP φ
MSG_DATA[0]		—	8	H'FFD388	RCAN_1	16	2+nP φ/3+nP φ
MSG_DATA[1]		—	8	H'FFD389	RCAN_1	16	2+nP φ/3+nP φ
MSG_DATA[2]		—	8	H'FFD38A	RCAN_1	16	2+nP φ/3+nP φ
MSG_DATA[3]		—	8	H'FFD38B	RCAN_1	16	2+nP φ/3+nP φ
MSG_DATA[4]		—	8	H'FFD38C	RCAN_1	16	2+nP φ/3+nP φ
MSG_DATA[5]		—	8	H'FFD38D	RCAN_1	16	2+nP φ/3+nP φ
MSG_DATA[6]		—	8	H'FFD38E	RCAN_1	16	2+nP φ/3+nP φ
MSG_DATA[7]		—	8	H'FFD38F	RCAN_1	16	2+nP φ/3+nP φ
CONTROL1H		—	8	H'FFD390	RCAN_1	16	2+nP φ/3+nP φ
CONTROL1L		—	8	H'FFD391	RCAN_1	16	2+nP φ/3+nP φ
MB_1[21].		CONTROL0H	—	16	H'FFD3A0	RCAN_1	16
	CONTROL0L	—	16	H'FFD3A2	RCAN_1	16	2+nP φ/3+nP φ
	LAFMH	—	16	H'FFD3A4	RCAN_1	16	2+nP φ/3+nP φ
	LAFML	—	16	H'FFD3A6	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[0]	—	8	H'FFD3A8	RCAN_1	16	2+nP φ/3+nP φ

レジスタ名称	略称	ビット数	アドレス* ¹	モジュール	データ バス幅	アクセス* ² ステート数 (リード/ライト)		
MB_1[21].	MSG_DATA[1]	—	8	H'FFD3A9	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[2]	—	8	H'FFD3AA	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[3]	—	8	H'FFD3AB	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[4]	—	8	H'FFD3AC	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[5]	—	8	H'FFD3AD	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[6]	—	8	H'FFD3AE	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[7]	—	8	H'FFD3AF	RCAN_1	16	2+nP φ/3+nP φ	
	CONTROL1H	—	8	H'FFD3B0	RCAN_1	16	2+nP φ/3+nP φ	
	CONTROL1L	—	8	H'FFD3B1	RCAN_1	16	2+nP φ/3+nP φ	
MB_1[22].	CONTROL0H	—	16	H'FFD3C0	RCAN_1	16	2+nP φ/3+nP φ	
	CONTROL0L	—	16	H'FFD3C2	RCAN_1	16	2+nP φ/3+nP φ	
	LAFMH	—	16	H'FFD3C4	RCAN_1	16	2+nP φ/3+nP φ	
	LAFML	—	16	H'FFD3C6	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[0]	—	8	H'FFD3C8	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[1]	—	8	H'FFD3C9	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[2]	—	8	H'FFD3CA	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[3]	—	8	H'FFD3CB	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[4]	—	8	H'FFD3CC	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[5]	—	8	H'FFD3CD	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[6]	—	8	H'FFD3CE	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[7]	—	8	H'FFD3CF	RCAN_1	16	2+nP φ/3+nP φ	
	CONTROL1H	—	8	H'FFD3D0	RCAN_1	16	2+nP φ/3+nP φ	
	CONTROL1L	—	8	H'FFD3D1	RCAN_1	16	2+nP φ/3+nP φ	
	MB_1[23].	CONTROL0H	—	16	H'FFD3E0	RCAN_1	16	2+nP φ/3+nP φ
		CONTROL0L	—	16	H'FFD3E2	RCAN_1	16	2+nP φ/3+nP φ
LAFMH		—	16	H'FFD3E4	RCAN_1	16	2+nP φ/3+nP φ	
LAFML		—	16	H'FFD3E6	RCAN_1	16	2+nP φ/3+nP φ	
MSG_DATA[0]		—	8	H'FFD3E8	RCAN_1	16	2+nP φ/3+nP φ	
MSG_DATA[1]		—	8	H'FFD3E9	RCAN_1	16	2+nP φ/3+nP φ	
MSG_DATA[2]		—	8	H'FFD3EA	RCAN_1	16	2+nP φ/3+nP φ	
MSG_DATA[3]		—	8	H'FFD3EB	RCAN_1	16	2+nP φ/3+nP φ	
MSG_DATA[4]		—	8	H'FFD3EC	RCAN_1	16	2+nP φ/3+nP φ	
MSG_DATA[5]		—	8	H'FFD3ED	RCAN_1	16	2+nP φ/3+nP φ	
MSG_DATA[6]		—	8	H'FFD3EE	RCAN_1	16	2+nP φ/3+nP φ	
MSG_DATA[7]		—	8	H'FFD3EF	RCAN_1	16	2+nP φ/3+nP φ	

レジスタ名称	略称	ビット数	アドレス*1	モジュール	データ バス幅	アクセス*2 (リード/ライト)		
MB_1[23].	CONTROL1H	—	8	H'FFD3F0	RCAN_1	16	2+nP φ/3+nP φ	
	CONTROL1L	—	8	H'FFD3F1	RCAN_1	16	2+nP φ/3+nP φ	
MB_1[24].	CONTROL0H	—	16	H'FFD400	RCAN_1	16	2+nP φ/3+nP φ	
	CONTROL0L	—	16	H'FFD402	RCAN_1	16	2+nP φ/3+nP φ	
	LAFMH	—	16	H'FFD404	RCAN_1	16	2+nP φ/3+nP φ	
	LAFML	—	16	H'FFD406	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[0]	—	8	H'FFD408	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[1]	—	8	H'FFD409	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[2]	—	8	H'FFD40A	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[3]	—	8	H'FFD40B	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[4]	—	8	H'FFD40C	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[5]	—	8	H'FFD40D	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[6]	—	8	H'FFD40E	RCAN_1	16	2+nP φ/3+nP φ	
	MSG_DATA[7]	—	8	H'FFD40F	RCAN_1	16	2+nP φ/3+nP φ	
	CONTROL1H	—	8	H'FFD410	RCAN_1	16	2+nP φ/3+nP φ	
	CONTROL1L	—	8	H'FFD411	RCAN_1	16	2+nP φ/3+nP φ	
	TTT	—	16	H'FFD414	RCAN_1	16	2P φ/3P φ	
	TTCONTROL	—	16	H'FFD416	RCAN_1	16	2P φ/3P φ	
	MB_1[25].	CONTROL0H	—	16	H'FFD420	RCAN_1	16	2+nP φ/3+nP φ
		CONTROL0L	—	16	H'FFD422	RCAN_1	16	2+nP φ/3+nP φ
		LAFMH	—	16	H'FFD424	RCAN_1	16	2+nP φ/3+nP φ
LAFML		—	16	H'FFD426	RCAN_1	16	2+nP φ/3+nP φ	
MSG_DATA[0]		—	8	H'FFD428	RCAN_1	16	2+nP φ/3+nP φ	
MSG_DATA[1]		—	8	H'FFD429	RCAN_1	16	2+nP φ/3+nP φ	
MSG_DATA[2]		—	8	H'FFD42A	RCAN_1	16	2+nP φ/3+nP φ	
MSG_DATA[3]		—	8	H'FFD42B	RCAN_1	16	2+nP φ/3+nP φ	
MSG_DATA[4]		—	8	H'FFD42C	RCAN_1	16	2+nP φ/3+nP φ	
MSG_DATA[5]		—	8	H'FFD42D	RCAN_1	16	2+nP φ/3+nP φ	
MSG_DATA[6]		—	8	H'FFD42E	RCAN_1	16	2+nP φ/3+nP φ	
MSG_DATA[7]		—	8	H'FFD42F	RCAN_1	16	2+nP φ/3+nP φ	
CONTROL1H		—	8	H'FFD430	RCAN_1	16	2+nP φ/3+nP φ	
CONTROL1L		—	8	H'FFD431	RCAN_1	16	2+nP φ/3+nP φ	
TTT		—	16	H'FFD434	RCAN_1	16	2P φ/3P φ	
TTCONTROL		—	16	H'FFD436	RCAN_1	16	2P φ/3P φ	

レジスタ名称	略称	ビット数	アドレス* ¹	モジュール	データ バス幅	アクセス* ² ステート数 (リード/ライト)
MB_1[26].	CONTROL0H	—	H'FFD440	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL0L	—	H'FFD442	RCAN_1	16	2+nP φ/3+nP φ
	LAFMH	—	H'FFD444	RCAN_1	16	2+nP φ/3+nP φ
	LAFML	—	H'FFD446	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[0]	—	H'FFD448	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[1]	—	H'FFD449	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[2]	—	H'FFD44A	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[3]	—	H'FFD44B	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[4]	—	H'FFD44C	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[5]	—	H'FFD44D	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[6]	—	H'FFD44E	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	H'FFD44F	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1H	—	H'FFD450	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1L	—	H'FFD451	RCAN_1	16	2+nP φ/3+nP φ
	TTT	—	H'FFD454	RCAN_1	16	2P φ/3P φ
	TTCONTROL	—	H'FFD456	RCAN_1	16	2P φ/3P φ
MB_1[27].	CONTROL0H	—	H'FFD460	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL0L	—	H'FFD462	RCAN_1	16	2+nP φ/3+nP φ
	LAFMH	—	H'FFD464	RCAN_1	16	2+nP φ/3+nP φ
	LAFML	—	H'FFD466	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[0]	—	H'FFD468	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[1]	—	H'FFD469	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[2]	—	H'FFD46A	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[3]	—	H'FFD46B	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[4]	—	H'FFD46C	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[5]	—	H'FFD46D	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[6]	—	H'FFD46E	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	H'FFD46F	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1H	—	H'FFD470	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1L	—	H'FFD471	RCAN_1	16	2+nP φ/3+nP φ
	TTT	—	H'FFD474	RCAN_1	16	2P φ/3P φ
	TTCONTROL	—	H'FFD476	RCAN_1	16	2P φ/3P φ
MB_1[28].	CONTROL0H	—	H'FFD480	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL0L	—	H'FFD482	RCAN_1	16	2+nP φ/3+nP φ
	LAFMH	—	H'FFD484	RCAN_1	16	2+nP φ/3+nP φ

レジスタ名称	略称	ビット数	アドレス* ¹	モジュール	データ バス幅	アクセス* ² ステート数 (リード/ライト)
MB_1[28].	LAFML	—	H'FFD486	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[0]	—	H'FFD488	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[1]	—	H'FFD489	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[2]	—	H'FFD48A	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[3]	—	H'FFD48B	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[4]	—	H'FFD48C	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[5]	—	H'FFD48D	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[6]	—	H'FFD48E	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	H'FFD48F	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1H	—	H'FFD490	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1L	—	H'FFD491	RCAN_1	16	2+nP φ/3+nP φ
	TTT	—	H'FFD494	RCAN_1	16	2P φ/3P φ
	TTCONTROL	—	H'FFD496	RCAN_1	16	2P φ/3P φ
MB_1[29].	CONTROL0H	—	H'FFD4A0	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL0L	—	H'FFD4A2	RCAN_1	16	2+nP φ/3+nP φ
	LAFMH	—	H'FFD4A4	RCAN_1	16	2+nP φ/3+nP φ
	LAFML	—	H'FFD4A6	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[0]	—	H'FFD4A8	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[1]	—	H'FFD4A9	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[2]	—	H'FFD4AA	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[3]	—	H'FFD4AB	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[4]	—	H'FFD4AC	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[5]	—	H'FFD4AD	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[6]	—	H'FFD4AE	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	H'FFD4AF	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1H	—	H'FFD4B0	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1L	—	H'FFD4B1	RCAN_1	16	2+nP φ/3+nP φ
	TTT	—	H'FFD4B4	RCAN_1	16	2P φ/3P φ
TTCONTROL	—	H'FFD4B6	RCAN_1	16	2P φ/3P φ	
MB_1[30].	CONTROL0H	—	H'FFD4C0	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL0L	—	H'FFD4C2	RCAN_1	16	2+nP φ/3+nP φ
	LAFMH	—	H'FFD4C4	RCAN_1	16	2+nP φ/3+nP φ
	LAFML	—	H'FFD4C6	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[0]	—	H'FFD4C8	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[1]	—	H'FFD4C9	RCAN_1	16	2+nP φ/3+nP φ

レジスタ名称	略称	ビット数	アドレス* ¹	モジュール	データ バス幅	アクセス* ² ステート数 (リード/ライト)	
MB_1[30].	MSG_DATA[2]	—	8	H'FFD4CA	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[3]	—	8	H'FFD4CB	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[4]	—	8	H'FFD4CC	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[5]	—	8	H'FFD4CD	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[6]	—	8	H'FFD4CE	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	8	H'FFD4CF	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1H	—	8	H'FFD4D0	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1L	—	8	H'FFD4D1	RCAN_1	16	2+nP φ/3+nP φ
	TIMSTAMP	—	16	H'FFD4D2	RCAN_1	16	2P φ/3P φ
	TTT	—	16	H'FFD4D4	RCAN_1	16	2P φ/3P φ
MB_1[31].	CONTROL0H	—	16	H'FFD4E0	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL0L	—	16	H'FFD4E2	RCAN_1	16	2+nP φ/3+nP φ
	LAFMH	—	16	H'FFD4E4	RCAN_1	16	2+nP φ/3+nP φ
	LAFML	—	16	H'FFD4E6	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[0]	—	8	H'FFD4E8	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[1]	—	8	H'FFD4E9	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[2]	—	8	H'FFD4EA	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[3]	—	8	H'FFD4EB	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[4]	—	8	H'FFD4EC	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[5]	—	8	H'FFD4ED	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[6]	—	8	H'FFD4EE	RCAN_1	16	2+nP φ/3+nP φ
	MSG_DATA[7]	—	8	H'FFD4EF	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1H	—	8	H'FFD4F0	RCAN_1	16	2+nP φ/3+nP φ
	CONTROL1L	—	8	H'FFD4F1	RCAN_1	16	2+nP φ/3+nP φ
	TIMSTAMP	—	16	H'FFD4F2	RCAN_1	16	2P φ/3P φ
RSPI 制御レジスタ A	SPCRA	8	H'FFD800	RSPI	16	2R φ/2R φ	
RSPI スレーブセレクト極性レジスタ A	SSLPA	8	H'FFD801	RSPI	16	2R φ/2R φ	
RSPI 端子制御レジスタ A	SPPCRA	8	H'FFD802	RSPI	16	2R φ/2R φ	
RSPI ステータスレジスタ A	SPSRA	8	H'FFD803	RSPI	16	2R φ/2R φ	
RSPI データレジスタ A	SPDRA	32	H'FFD804	RSPI	16	2R φ/2R φ	
RSPI シーケンス制御レジスタ A	SPSCRA	8	H'FFD808	RSPI	16	2R φ/2R φ	
RSPI シーケンスステータスレジスタ A	SPSSRA	8	H'FFD809	RSPI	16	2R φ/2R φ	
RSPI ビットレートレジスタ A	SPBRA	8	H'FFD80A	RSPI	16	2R φ/2R φ	
RSPI データコントロールレジスタ A	SPDCRA	8	H'FFD80B	RSPI	16	2R φ/2R φ	
RSPI クロック遅延レジスタ A	SPCKDA	8	H'FFD80C	RSPI	16	2R φ/2R φ	
RSPI スレーブセレクトネゲート遅延レジスタ A	SSLNDA	8	H'FFD80D	RSPI	16	2R φ/2R φ	

レジスタ名称	略称	ビット数	アドレス* ¹	モジュール	データ バス幅	アクセス* ² ステート数 (リード/ライト)
RSPI 次アクセス遅延レジスタ A	SPNDA	8	H'FFD80E	RSPI	16	2Rφ/2Rφ
RSPI 制御レジスタ 2 A	SPCR2A	8	H'FFD80F	RSPI	16	2Rφ/2Rφ
RSPI コマンドレジスタ A0	SPCMDA0	16	H'FFD810	RSPI	16	2Rφ/2Rφ
RSPI コマンドレジスタ A1	SPCMDA1	16	H'FFD812	RSPI	16	2Rφ/2Rφ
RSPI コマンドレジスタ A2	SPCMDA2	16	H'FFD814	RSPI	16	2Rφ/2Rφ
RSPI コマンドレジスタ A3	SPCMDA3	16	H'FFD816	RSPI	16	2Rφ/2Rφ
RSPI コマンドレジスタ A4	SPCMDA4	16	H'FFD818	RSPI	16	2Rφ/2Rφ
RSPI コマンドレジスタ A5	SPCMDA5	16	H'FFD81A	RSPI	16	2Rφ/2Rφ
RSPI コマンドレジスタ A6	SPCMDA6	16	H'FFD81C	RSPI	16	2Rφ/2Rφ
RSPI コマンドレジスタ A7	SPCMDA7	16	H'FFD81E	RSPI	16	2Rφ/2Rφ
RSPI 制御レジスタ B	SPCRB	8	H'FFD820	RSPI	16	2Rφ/2Rφ
RSPI スレーブセレクト極性レジスタ B	SSLPB	8	H'FFD821	RSPI	16	2Rφ/2Rφ
RSPI 端子制御レジスタ B	SPPCRB	8	H'FFD822	RSPI	16	2Rφ/2Rφ
RSPI ステータスレジスタ B	SPSRB	8	H'FFD823	RSPI	16	2Rφ/2Rφ
RSPI データレジスタ B	SPDRB	32	H'FFD824	RSPI	16	2Rφ/2Rφ
RSPI シーケンス制御レジスタ B	SPSCRB	8	H'FFD828	RSPI	16	2Rφ/2Rφ
RSPI シーケンスステータスレジスタ B	SPSSRB	8	H'FFD829	RSPI	16	2Rφ/2Rφ
RSPI ビットレートレジスタ B	SPBRB	8	H'FFD82A	RSPI	16	2Rφ/2Rφ
RSPI データコントロールレジスタ B	SPDCRB	8	H'FFD82B	RSPI	16	2Rφ/2Rφ
RSPI クロック遅延レジスタ B	SPCKDB	8	H'FFD82C	RSPI	16	2Rφ/2Rφ
RSPI スレーブセレクトネゲート遅延レジスタ B	SSLNDB	8	H'FFD82D	RSPI	16	2Rφ/2Rφ
RSPI 次アクセス遅延レジスタ B	SPNDB	8	H'FFD82E	RSPI	16	2Rφ/2Rφ
RSPI 制御レジスタ 2 B	SPCR2B	8	H'FFD82F	RSPI	16	2Rφ/2Rφ
RSPI コマンドレジスタ B0	SPCMDB0	16	H'FFD830	RSPI	16	2Rφ/2Rφ
RSPI コマンドレジスタ B1	SPCMDB1	16	H'FFD832	RSPI	16	2Rφ/2Rφ
RSPI コマンドレジスタ B2	SPCMDB2	16	H'FFD834	RSPI	16	2Rφ/2Rφ
RSPI コマンドレジスタ B3	SPCMDB3	16	H'FFD836	RSPI	16	2Rφ/2Rφ
RSPI コマンドレジスタ B4	SPCMDB4	16	H'FFD838	RSPI	16	2Rφ/2Rφ
RSPI コマンドレジスタ B5	SPCMDB5	16	H'FFD83A	RSPI	16	2Rφ/2Rφ
RSPI コマンドレジスタ B6	SPCMDB6	16	H'FFD83C	RSPI	16	2Rφ/2Rφ
RSPI コマンドレジスタ B7	SPCMDB7	16	H'FFD83E	RSPI	16	2Rφ/2Rφ
RSPI 制御レジスタ C	SPCRC	8	H'FFD840	RSPI	16	2Rφ/2Rφ
RSPI スレーブセレクト極性レジスタ C	SSLPC	8	H'FFD841	RSPI	16	2Rφ/2Rφ
RSPI 端子制御レジスタ C	SPPCRC	8	H'FFD842	RSPI	16	2Rφ/2Rφ
RSPI ステータスレジスタ C	SPSRC	8	H'FFD843	RSPI	16	2Rφ/2Rφ
RSPI データレジスタ C	SPDRC	32	H'FFD844	RSPI	16	2Rφ/2Rφ

レジスタ名称	略称	ビット数	アドレス* ¹	モジュール	データ バス幅	アクセス* ² ステート数 (リード/ライト)
RSPI シーケンス制御レジスタ C	SPSCRC	8	H'FFD848	RSPI	16	2Rφ/2Rφ
RSPI シーケンスステータスレジスタ C	SPSSRC	8	H'FFD849	RSPI	16	2Rφ/2Rφ
RSPI ビットレートレジスタ C	SPBRC	8	H'FFD84A	RSPI	16	2Rφ/2Rφ
RSPI データコントロールレジスタ C	SPDCRC	8	H'FFD84B	RSPI	16	2Rφ/2Rφ
RSPI クロック遅延レジスタ C	SPCKDC	8	H'FFD84C	RSPI	16	2Rφ/2Rφ
RSPI スレーブセレクトネゲート遅延レジスタ C	SSLNDC	8	H'FFD84D	RSPI	16	2Rφ/2Rφ
RSPI 次アクセス遅延レジスタ C	SPNDC	8	H'FFD84E	RSPI	16	2Rφ/2Rφ
RSPI 制御レジスタ 2 C	SPCR2C	8	H'FFD84F	RSPI	16	2Rφ/2Rφ
RSPI コマンドレジスタ C0	SPCMDC0	16	H'FFD850	RSPI	16	2Rφ/2Rφ
RSPI コマンドレジスタ C1	SPCMDC1	16	H'FFD852	RSPI	16	2Rφ/2Rφ
RSPI コマンドレジスタ C2	SPCMDC2	16	H'FFD854	RSPI	16	2Rφ/2Rφ
RSPI コマンドレジスタ C3	SPCMDC3	16	H'FFD856	RSPI	16	2Rφ/2Rφ
RSPI コマンドレジスタ C4	SPCMDC4	16	H'FFD858	RSPI	16	2Rφ/2Rφ
RSPI コマンドレジスタ C5	SPCMDC5	16	H'FFD85A	RSPI	16	2Rφ/2Rφ
RSPI コマンドレジスタ C6	SPCMDC6	16	H'FFD85C	RSPI	16	2Rφ/2Rφ
RSPI コマンドレジスタ C7	SPCMDC7	16	H'FFD85E	RSPI	16	2Rφ/2Rφ
RSPI 制御レジスタ D	SPCRD	8	H'FFD860	RSPI	16	2Rφ/2Rφ
RSPI スレーブセレクト極性レジスタ D	SSLPD	8	H'FFD861	RSPI	16	2Rφ/2Rφ
RSPI 端子制御レジスタ D	SPPCRD	8	H'FFD862	RSPI	16	2Rφ/2Rφ
RSPI ステータスレジスタ D	SPSRD	8	H'FFD863	RSPI	16	2Rφ/2Rφ
RSPI データレジスタ D	SPDRD	32	H'FFD864	RSPI	16	2Rφ/2Rφ
RSPI シーケンス制御レジスタ D	SPSCRD	8	H'FFD868	RSPI	16	2Rφ/2Rφ
RSPI シーケンスステータスレジスタ D	SPSSRD	8	H'FFD869	RSPI	16	2Rφ/2Rφ
RSPI ビットレートレジスタ D	SPBRD	8	H'FFD86A	RSPI	16	2Rφ/2Rφ
RSPI データコントロールレジスタ D	SPDCRD	8	H'FFD86B	RSPI	16	2Rφ/2Rφ
RSPI クロック遅延レジスタ D	SPCKDD	8	H'FFD86C	RSPI	16	2Rφ/2Rφ
RSPI スレーブセレクトネゲート遅延レジスタ D	SSLNDD	8	H'FFD86D	RSPI	16	2Rφ/2Rφ
RSPI 次アクセス遅延レジスタ D	SPNDD	8	H'FFD86E	RSPI	16	2Rφ/2Rφ
RSPI 制御レジスタ 2 D	SPCR2D	8	H'FFD86F	RSPI	16	2Rφ/2Rφ
RSPI コマンドレジスタ D0	SPCMDD0	16	H'FFD870	RSPI	16	2Rφ/2Rφ
RSPI コマンドレジスタ D1	SPCMDD1	16	H'FFD872	RSPI	16	2Rφ/2Rφ
RSPI コマンドレジスタ D2	SPCMDD2	16	H'FFD874	RSPI	16	2Rφ/2Rφ
RSPI コマンドレジスタ D3	SPCMDD3	16	H'FFD876	RSPI	16	2Rφ/2Rφ
RSPI コマンドレジスタ D4	SPCMDD4	16	H'FFD878	RSPI	16	2Rφ/2Rφ
RSPI コマンドレジスタ D5	SPCMDD5	16	H'FFD87A	RSPI	16	2Rφ/2Rφ
RSPI コマンドレジスタ D6	SPCMDD6	16	H'FFD87C	RSPI	16	2Rφ/2Rφ

レジスタ名称	略称	ビット数	アドレス* ¹	モジュール	データ バス幅	アクセス* ² ステート数 (リード/ライト)
RSPI コマンドレジスタ D7	SPCMDD7	16	H'FFD87E	RSPI	16	2Rφ/2Rφ
LIN コントロールレジスタ	LINCR	8	H'FFD900	HWLIN	16	2Pφ/2Pφ
LIN ステータスレジスタ	LINSTR	8	H'FFD901	HWLIN	16	2Pφ/2Pφ
LIN タイマコントロールレジスタ	LINTCR	8	H'FFD902	HWLIN	16	2Pφ/2Pφ
	—		H'FFD903	HWLIN	16	
LIN タイマカウンタ	LINTCNT	16	H'FFD904	HWLIN	16	2Pφ/2Pφ
LIN タイムアウトカウンタ	LINTCNT	16	H'FFD906	HWLIN	16	2Pφ/2Pφ
	—	—	H'FFD908~ H'FFD9FF	HWLIN	16	
CRC コントロールレジスタ	CRCCR	8	H'FFDA4C	CRC	16	2Pφ/2Pφ
CRC データ入力レジスタ	CRCDIR	8	H'FFDA4D	CRC	16	2Pφ/2Pφ
CRC データ出力レジスタ	CRCDOR	16	H'FFDA4E	CRC	16	2Pφ/2Pφ
フラッシュモードレジスタ	FMODR	8	H'FFE002	FLASH/EEPROM	8	2Fφ/3Fφ
フラッシュアクセスステータスレジスタ	FASTAT	8	H'FFE010	FLASH/EEPROM	8	2Fφ/3Fφ
フラッシュアクセスエラー割り込み許可 レジスタ	FAEINT	8	H'FFE011	FLASH/EEPROM	8	2Fφ/3Fφ
フラッシュメモリマット選択レジスタ	ROMMAT	16	H'FFE020	FLASH/EEPROM	16	2Fφ/3Fφ
EEPROM 読み出し許可レジスタ 0	EEPRE0	16	H'FFE040	FLASH/EEPROM	16	2Fφ/3Fφ
EEPROM 読み出し許可レジスタ 1	EEPRE1	16	H'FFE042	FLASH/EEPROM	16	2Fφ/3Fφ
EEPROM 書き込み/消去許可レジスタ 0	EEPWE0	16	H'FFE050	FLASH/EEPROM	16	2Fφ/3Fφ
EEPROM 書き込み/消去許可レジスタ 1	EEPWE1	16	H'FFE052	FLASH/EEPROM	16	2Fφ/3Fφ
FCU RAM イネーブルレジスタ	FCURAME	16	H'FFE054	FLASH/EEPROM	16	2Fφ/3Fφ
フラッシュステータスレジスタ 0	FSTATR0	8	H'FFE100	FLASH/EEPROM	8	2Fφ/3Fφ
フラッシュステータスレジスタ 1	FSTATR1	8	H'FFE101	FLASH/EEPROM	8	2Fφ/3Fφ
フラッシュ P/E モードエントリレジスタ	FENTRYR	16	H'FFE102	FLASH/EEPROM	16	2Fφ/3Fφ
フラッシュプロテクトレジスタ	FPROTR	16	H'FFE104	FLASH/EEPROM	16	2Fφ/3Fφ
フラッシュリセットレジスタ	FRESETR	16	H'FFE106	FLASH/EEPROM	16	2Fφ/3Fφ
FCU コマンドレジスタ	FCMDR	16	H'FFE10A	FLASH/EEPROM	16	2Fφ/3Fφ
FCU RAM ECC エラーコントロールレジスタ	FRAMECCR	8	H'FFE10C	FLASH/EEPROM	8	2Fφ/3Fφ
FCU 処理切り替えレジスタ	FCPSR	16	H'FFE118	FLASH/EEPROM	16	2Fφ/3Fφ
EEPROM ブランクチェック制御レジスタ	EEPBCCNT	16	H'FFE11A	FLASH/EEPROM	16	2Fφ/3Fφ
フラッシュ P/E ステータスレジスタ	FPESTAT	16	H'FFE11C	FLASH/EEPROM	16	2Fφ/3Fφ
EEPROM ブランクチェックステータスレジスタ	EEPBCSTAT	16	H'FFE11E	FLASH/EEPROM	16	2Fφ/3Fφ
フラッシュロック通知レジスタ	FCKAR	16	H'FFE138	FLASH/EEPROM	16	2Fφ/3Fφ
EEPROM マット選択レジスタ	EEPSTAT	16	H'FFE380	FLASH/EEPROM	16	2Fφ/2Fφ
RAM イネーブルコントロールレジスタ	RAMEN	16	H'FFE400	RAM	16	2Pφ/2Pφ
RAM 書き込みイネーブルコントロールレジスタ	RAMWEN	16	H'FFE402	RAM	16	2Pφ/2Pφ

レジスタ名称	略称	ビット数	アドレス* ¹	モジュール	データ バス幅	アクセス* ² ステート数 (リード/ライト)
RAM_ECC イネーブルコントロールレジスタ	RAMECC	16	H'FFE404	RAM	16	2Pφ/2Pφ
RAM エラーステータスレジスタ	RAMERR	8	H'FFE406	RAM	8	2Pφ/2Pφ
RAM エラー割り込みコントロールレジスタ	RAMINT	8	H'FFE410	RAM	8	2Pφ/2Pφ
RAM アクセスサイクル設定レジスタ	RAMACYC	16	H'FFE412	RAM	16	2Pφ/2Pφ
A/D データレジスタ A_0	ADDRA_0	16	H'FFE500	A/D_0	16	2Aφ/2Aφ
A/D データレジスタ B_0	ADDRB_0	16	H'FFE502	A/D_0	16	2Aφ/2Aφ
A/D データレジスタ C_0	ADDRC_0	16	H'FFE504	A/D_0	16	2Aφ/2Aφ
A/D データレジスタ D_0	ADDRD_0	16	H'FFE506	A/D_0	16	2Aφ/2Aφ
A/D データレジスタ E_0	ADDRE_0	16	H'FFE508	A/D_0	16	2Aφ/2Aφ
A/D データレジスタ F_0	ADDRF_0	16	H'FFE50A	A/D_0	16	2Aφ/2Aφ
A/D データレジスタ G_0	ADDRG_0	16	H'FFE50C	A/D_0	16	2Aφ/2Aφ
A/D データレジスタ H_0	ADDRH_0	16	H'FFE50E	A/D_0	16	2Aφ/2Aφ
A/D コントロール/ステータスレジスタ_0	ADCSR_0	8	H'FFE510	A/D_0	16	2Aφ/2Aφ
A/D コントロールレジスタ_0	ADCR_0	8	H'FFE511	A/D_0	16	2Aφ/2Aφ
アナログポートブルダウコントロールレジスタ_0	APPDCR_0	8	H'FFE51C	A/D_0	16	2Aφ/2Aφ
A/D 自己診断レジスタ_0	ADDIAGR_0	8	H'FFE51D	A/D_0	16	2Aφ/2Aφ
A/D データレジスタ A_1	ADDRA_1	16	H'FFE520	A/D_1	16	2Aφ/2Aφ
A/D データレジスタ B_1	ADDRB_1	16	H'FFE522	A/D_1	16	2Aφ/2Aφ
A/D データレジスタ C_1	ADDRC_1	16	H'FFE524	A/D_1	16	2Aφ/2Aφ
A/D データレジスタ D_1	ADDRD_1	16	H'FFE526	A/D_1	16	2Aφ/2Aφ
A/D データレジスタ E_1	ADDRE_1	16	H'FFE528	A/D_1	16	2Aφ/2Aφ
A/D データレジスタ F_1	ADDRF_1	16	H'FFE52A	A/D_1	16	2Aφ/2Aφ
A/D データレジスタ G_1	ADDRG_1	16	H'FFE52C	A/D_1	16	2Aφ/2Aφ
A/D データレジスタ H_1	ADDRH_1	16	H'FFE52E	A/D_1	16	2Aφ/2Aφ
A/D コントロール/ステータスレジスタ_1	ADCSR_1	8	H'FFE530	A/D_1	16	2Aφ/2Aφ
A/D コントロールレジスタ_1	ADCR_1	8	H'FFE531	A/D_1	16	2Aφ/2Aφ
アナログポートブルダウコントロールレジスタ_1	APPDCR_1	8	H'FFE53C	A/D_1	16	2Aφ/2Aφ
A/D 自己診断レジスタ_1	ADDIAGR_1	8	H'FFE53D	A/D_1	16	2Aφ/2Aφ
タイマスタートレジスタ	TSTRB	8	H'FFF00	TPU (ユニット 1)	16	2Pφ/2Pφ
タイマシンクロレジスタ	TSYRB	8	H'FFF01	TPU (ユニット 1)	16	2Pφ/2Pφ
タイマコントロールレジスタ_6	TCR_6	8	H'FFF10	TPU_6	16	2Pφ/2Pφ
タイマモードレジスタ_6	TMDR_6	8	H'FFF11	TPU_6	16	2Pφ/2Pφ
タイマ I/O コントロールレジスタ H_6	TIORH_6	8	H'FFF12	TPU_6	16	2Pφ/2Pφ
タイマ I/O コントロールレジスタ L_6	TIORL_6	8	H'FFF13	TPU_6	16	2Pφ/2Pφ
タイマインタラプトイネーブルレジスタ_6	TIER_6	8	H'FFF14	TPU_6	16	2Pφ/2Pφ
タイマステータスレジスタ_6	TSR_6	8	H'FFF15	TPU_6	16	2Pφ/2Pφ

レジスタ名称	略称	ビット数	アドレス* ¹	モジュール	データ バス幅	アクセス* ² ステート数 (リード/ライト)
タイマカウンタ_6	TCNT_6	16	H'FFFB16	TPU_6	16	2Pφ/2Pφ
タイマジェネラルレジスタ A_6	TGRA_6	16	H'FFFB18	TPU_6	16	2Pφ/2Pφ
タイマジェネラルレジスタ B_6	TGRB_6	16	H'FFFB1A	TPU_6	16	2Pφ/2Pφ
タイマジェネラルレジスタ C_6	TGRC_6	16	H'FFFB1C	TPU_6	16	2Pφ/2Pφ
タイマジェネラルレジスタ D_6	TGRD_6	16	H'FFFB1E	TPU_6	16	2Pφ/2Pφ
タイマコントロールレジスタ_7	TCR_7	8	H'FFFB20	TPU_7	16	2Pφ/2Pφ
タイマモードレジスタ_7	TMDR_7	8	H'FFFB21	TPU_7	16	2Pφ/2Pφ
タイマ I/O コントロールレジスタ_7	TIOR_7	8	H'FFFB22	TPU_7	16	2Pφ/2Pφ
	—		H'FFFB23	TPU_7	16	
タイマインタラプトイネーブルレジスタ_7	TIER_7	8	H'FFFB24	TPU_7	16	2Pφ/2Pφ
タイマステータスレジスタ_7	TSR_7	8	H'FFFB25	TPU_7	16	2Pφ/2Pφ
タイマカウンタ_7	TCNT_7	16	H'FFFB26	TPU_7	16	2Pφ/2Pφ
タイマジェネラルレジスタ A_7	TGRA_7	16	H'FFFB28	TPU_7	16	2Pφ/2Pφ
タイマジェネラルレジスタ B_7	TGRB_7	16	H'FFFB2A	TPU_7	16	2Pφ/2Pφ
	—		H'FFFB2C~ H'FFFB2F	TPU_7	16	
タイマコントロールレジスタ_8	TCR_8	8	H'FFFB30	TPU_8	16	2Pφ/2Pφ
タイマモードレジスタ_8	TMDR_8	8	H'FFFB31	TPU_8	16	2Pφ/2Pφ
タイマ I/O コントロールレジスタ_8	TIOR_8	8	H'FFFB32	TPU_8	16	2Pφ/2Pφ
	—		H'FFFB33	TPU_8	16	
タイマインタラプトイネーブルレジスタ_8	TIER_8	8	H'FFFB34	TPU_8	16	2Pφ/2Pφ
タイマステータスレジスタ_8	TSR_8	8	H'FFFB35	TPU_8	16	2Pφ/2Pφ
タイマカウンタ_8	TCNT_8	16	H'FFFB36	TPU_8	16	2Pφ/2Pφ
タイマジェネラルレジスタ A_8	TGRA_8	16	H'FFFB38	TPU_8	16	2Pφ/2Pφ
タイマジェネラルレジスタ B_8	TGRB_8	16	H'FFFB3A	TPU_8	16	2Pφ/2Pφ
	—		H'FFFB3C~ H'FFFB3F	TPU_8	16	
タイマコントロールレジスタ_9	TCR_9	8	H'FFFB40	TPU_9	16	2Pφ/2Pφ
タイマモードレジスタ_9	TMDR_9	8	H'FFFB41	TPU_9	16	2Pφ/2Pφ
タイマ I/O コントロールレジスタ H_9	TIORH_9	8	H'FFFB42	TPU_9	16	2Pφ/2Pφ
タイマ I/O コントロールレジスタ L_9	TIORL_9	8	H'FFFB43	TPU_9	16	2Pφ/2Pφ
タイマインタラプトイネーブルレジスタ_9	TIER_9	8	H'FFFB44	TPU_9	16	2Pφ/2Pφ
タイマステータスレジスタ_9	TSR_9	8	H'FFFB45	TPU_9	16	2Pφ/2Pφ
タイマカウンタ_9	TCNT_9	16	H'FFFB46	TPU_9	16	2Pφ/2Pφ
タイマジェネラルレジスタ A_9	TGRA_9	16	H'FFFB48	TPU_9	16	2Pφ/2Pφ
タイマジェネラルレジスタ B_9	TGRB_9	16	H'FFFB4A	TPU_9	16	2Pφ/2Pφ
タイマジェネラルレジスタ C_9	TGRC_9	16	H'FFFB4C	TPU_9	16	2Pφ/2Pφ

レジスタ名称	略称	ビット数	アドレス* ¹	モジュール	データ バス幅	アクセス* ² ステート数 (リード/ライト)
タイマジェネラルレジスタ D_9	TGRD_9	16	H'FFFB4E	TPU_9	16	2Pφ/2Pφ
タイマコントロールレジスタ_10	TCR_10	8	H'FFFB50	TPU_10	16	2Pφ/2Pφ
タイマモードレジスタ_10	TMDR_10	8	H'FFFB51	TPU_10	16	2Pφ/2Pφ
タイマ I/O コントロールレジスタ_10	TIOR_10	8	H'FFFB52	TPU_10	16	2Pφ/2Pφ
	—		H'FFFB53	TPU_10	16	
タイマイントラプトイネーブルレジスタ_10	TIER_10	8	H'FFFB54	TPU_10	16	2Pφ/2Pφ
タイマステータスレジスタ_10	TSR_10	8	H'FFFB55	TPU_10	16	2Pφ/2Pφ
タイマカウンタ_10	TCNT_10	16	H'FFFB56	TPU_10	16	2Pφ/2Pφ
タイマジェネラルレジスタ A_10	TGRA_10	16	H'FFFB58	TPU_10	16	2Pφ/2Pφ
タイマジェネラルレジスタ B_10	TGRB_10	16	H'FFFB5A	TPU_10	16	2Pφ/2Pφ
	—		H'FFFB5C~ H'FFFB5F	TPU_10	16	
タイマコントロールレジスタ_11	TCR_11	8	H'FFFB60	TPU_11	16	2Pφ/2Pφ
タイマモードレジスタ_11	TMDR_11	8	H'FFFB61	TPU_11	16	2Pφ/2Pφ
タイマ I/O コントロールレジスタ_11	TIOR_11	8	H'FFFB62	TPU_11	16	2Pφ/2Pφ
	—		H'FFFB63	TPU_11	16	
タイマイントラプトイネーブルレジスタ_11	TIER_11	8	H'FFFB64	TPU_11	16	2Pφ/2Pφ
タイマステータスレジスタ_11	TSR_11	8	H'FFFB65	TPU_11	16	2Pφ/2Pφ
タイマカウンタ_11	TCNT_11	16	H'FFFB66	TPU_11	16	2Pφ/2Pφ
タイマジェネラルレジスタ A_11	TGRA_11	16	H'FFFB68	TPU_11	16	2Pφ/2Pφ
タイマジェネラルレジスタ B_11	TGRB_11	16	H'FFFB6A	TPU_11	16	2Pφ/2Pφ
	—		H'FFFB6C~ H'FFFB6F	TPU_11	16	
ポート 1 データディレクションレジスタ	P1DDR	8	H'FFFB80	I/O ポート	8	2Pφ/2Pφ
ポート 3 データディレクションレジスタ	P3DDR	8	H'FFFB82	I/O ポート	8	2Pφ/2Pφ
ポート 6 データディレクションレジスタ	P6DDR	8	H'FFFB85	I/O ポート	8	2Pφ/2Pφ
ポート A データディレクションレジスタ	PADDR	8	H'FFFB89	I/O ポート	8	2Pφ/2Pφ
ポート D データディレクションレジスタ	PDDDR	8	H'FFFB8C	I/O ポート	8	2Pφ/2Pφ
ポート 1 入力バッファコントロールレジスタ	P1ICR	8	H'FFFB90	I/O ポート	8	2Pφ/2Pφ
ポート 3 入力バッファコントロールレジスタ	P3ICR	8	H'FFFB92	I/O ポート	8	2Pφ/2Pφ
ポート 4 入力バッファコントロールレジスタ	P4ICR	8	H'FFFB93	I/O ポート	8	2Pφ/2Pφ
ポート 5 入力バッファコントロールレジスタ	P5ICR	8	H'FFFB94	I/O ポート	8	2Pφ/2Pφ
ポート 6 入力バッファコントロールレジスタ	P6ICR	8	H'FFFB95	I/O ポート	8	2Pφ/2Pφ
ポート A 入力バッファコントロールレジスタ	PAICR	8	H'FFFB99	I/O ポート	8	2Pφ/2Pφ
ポート D 入力バッファコントロールレジスタ	PDICR	8	H'FFFB9C	I/O ポート	8	2Pφ/2Pφ
ポート H レジスタ	PORTH	8	H'FFFBAA0	I/O ポート	8	2Pφ/2Pφ
ポート J レジスタ	PORTJ	8	H'FFFBAA2	I/O ポート	8	2Pφ/2Pφ

レジスタ名称	略称	ビット数	アドレス* ¹	モジュール	データ バス幅	アクセス* ² ステート数 (リード/ライト)
ポート K レジスタ	PORTK	8	H'FFFBA3	I/O ポート	8	2Pφ/2Pφ
ポート H データレジスタ	PHDR	8	H'FFFBA4	I/O ポート	8	2Pφ/2Pφ
ポート J データレジスタ	PJDR	8	H'FFFBA6	I/O ポート	8	2Pφ/2Pφ
ポート K データレジスタ	PKDR	8	H'FFFBA7	I/O ポート	8	2Pφ/2Pφ
ポート H データディレクションレジスタ	PHDDR	8	H'FFFBA8	I/O ポート	8	2Pφ/2Pφ
ポート J データディレクションレジスタ	PJDDR	8	H'FFFBA A	I/O ポート	8	2Pφ/2Pφ
ポート K データディレクションレジスタ	PKDDR	8	H'FFFBA B	I/O ポート	8	2Pφ/2Pφ
ポート H 入力バッファコントロールレジスタ	PHICR	8	H'FFFBA C	I/O ポート	8	2Pφ/2Pφ
ポート J 入力バッファコントロールレジスタ	PJICR	8	H'FFFBA E	I/O ポート	8	2Pφ/2Pφ
ポート K 入力バッファコントロールレジスタ	PKICR	8	H'FFFBA F	I/O ポート	8	2Pφ/2Pφ
ポート D プルアップ MOS コントロールレジスタ	PDPCR	8	H'FFFBB4	I/O ポート	8	2Pφ/2Pφ
ポート H プルアップ MOS コントロールレジスタ	PHPCR	8	H'FFFBB8	I/O ポート	8	2Pφ/2Pφ
ポート J プルアップ MOS コントロールレジスタ	PJPCR	8	H'FFFBB A	I/O ポート	8	2Pφ/2Pφ
ポート K プルアップ MOS コントロールレジスタ	PKPCR	8	H'FFFBB B	I/O ポート	8	2Pφ/2Pφ
ポートファンクションコントロールレジスタ 5	PFCR5	8	H'FFFBC5	I/O ポート	8	2Pφ/3Pφ
ポートファンクションコントロールレジスタ 6	PFCR6	8	H'FFFBC6	I/O ポート	8	2Pφ/3Pφ
ポートファンクションコントロールレジスタ 8	PFCR8	8	H'FFFBC8	I/O ポート	8	2Pφ/3Pφ
ポートファンクションコントロールレジスタ 9	PFCR9	8	H'FFFBC9	I/O ポート	8	2Pφ/3Pφ
ポートファンクションコントロールレジスタ A	PFCRA	8	H'FFFBCA	I/O ポート	8	2Pφ/3Pφ
ポートファンクションコントロールレジスタ B	PFCRB	8	H'FFFBCB	I/O ポート	8	2Pφ/3Pφ
ポートファンクションコントロールレジスタ C	PFCRC	8	H'FFFBC C	I/O ポート	8	2Pφ/3Pφ
ポートファンクションコントロールレジスタ D	PFCRD	8	H'FFFBC D	I/O ポート	8	2Pφ/3Pφ
ソフトウェアスタンバイ解除 IRQ イネーブルレジスタ	SSIER	16	H'FFFBC E	INTC	16	2Iφ/3Iφ
ポート 1 ドライバビリティ設定レジスタ	P1DSR	8	H'FFFBE0	I/O ポート	8	2Pφ/2Pφ
ポート 6 ドライバビリティ設定レジスタ	P6DSR	8	H'FFFBE2	I/O ポート	8	2Pφ/2Pφ
ポート A ドライバビリティ設定レジスタ	PADSR	8	H'FFFBE3	I/O ポート	8	2Pφ/2Pφ
ポート D ドライバビリティ設定レジスタ	PDDSR	8	H'FFFBE4	I/O ポート	8	2Pφ/2Pφ
ポート H ドライバビリティ設定レジスタ	PHDSR	8	H'FFFBE6	I/O ポート	8	2Pφ/2Pφ
ポート 1 端子状態設定レジスタ	P1PSR	8	H'FFFBE8	I/O ポート	8	2Pφ/2Pφ
ポート 6 端子状態設定レジスタ	P6PSR	8	H'FFFBEA	I/O ポート	8	2Pφ/2Pφ
ポート A 端子状態設定レジスタ	PAPSR	8	H'FFFBE B	I/O ポート	8	2Pφ/2Pφ
ポート D 端子状態設定レジスタ	PDPSR	8	H'FFFBE C	I/O ポート	8	2Pφ/2Pφ
ポート H 端子状態設定レジスタ	PHPSR	8	H'FFFBE E	I/O ポート	8	2Pφ/2Pφ
DMA ソースアドレスレジスタ_0	DSAR_0	32	H'FFFC00	DMAC_0	16	2Iφ/2Iφ
DMA デスティネーションアドレスレジスタ_0	DDAR_0	32	H'FFFC04	DMAC_0	16	2Iφ/2Iφ
DMA オフセットレジスタ_0	DOFR_0	32	H'FFFC08	DMAC_0	16	2Iφ/2Iφ

レジスタ名称	略称	ビット数	アドレス* ¹	モジュール	データ バス幅	アクセス* ² (リード/ライト)
DMA 転送カウントレジスタ_0	DTCR_0	32	H'FFFC0C	DMAC_0	16	21φ/21φ
DMA ブロックサイズレジスタ_0	DBSR_0	32	H'FFFC10	DMAC_0	16	21φ/21φ
DMA モードコントロールレジスタ_0	DMDR_0	32	H'FFFC14	DMAC_0	16	21φ/21φ
DMA アドレスコントロールレジスタ_0	DACR_0	32	H'FFFC18	DMAC_0	16	21φ/21φ
DMA ソースアドレスレジスタ_1	DSAR_1	32	H'FFFC20	DMAC_1	16	21φ/21φ
DMA デスティネーションアドレスレジスタ_1	DDAR_1	32	H'FFFC24	DMAC_1	16	21φ/21φ
DMA オフセットレジスタ_1	DOFR_1	32	H'FFFC28	DMAC_1	16	21φ/21φ
DMA 転送カウントレジスタ_1	DTCR_1	32	H'FFFC2C	DMAC_1	16	21φ/21φ
DMA ブロックサイズレジスタ_1	DBSR_1	32	H'FFFC30	DMAC_1	16	21φ/21φ
DMA モードコントロールレジスタ_1	DMDR_1	32	H'FFFC34	DMAC_1	16	21φ/21φ
DMA アドレスコントロールレジスタ_1	DACR_1	32	H'FFFC38	DMAC_1	16	21φ/21φ
DMA ソースアドレスレジスタ_2	DSAR_2	32	H'FFFC40	DMAC_2	16	21φ/21φ
DMA デスティネーションアドレスレジスタ_2	DDAR_2	32	H'FFFC44	DMAC_2	16	21φ/21φ
DMA オフセットレジスタ_2	DOFR_2	32	H'FFFC48	DMAC_2	16	21φ/21φ
DMA 転送カウントレジスタ_2	DTCR_2	32	H'FFFC4C	DMAC_2	16	21φ/21φ
DMA ブロックサイズレジスタ_2	DBSR_2	32	H'FFFC50	DMAC_2	16	21φ/21φ
DMA モードコントロールレジスタ_2	DMDR_2	32	H'FFFC54	DMAC_2	16	21φ/21φ
DMA アドレスコントロールレジスタ_2	DACR_2	32	H'FFFC58	DMAC_2	16	21φ/21φ
DMA ソースアドレスレジスタ_3	DSAR_3	32	H'FFFC60	DMAC_3	16	21φ/21φ
DMA デスティネーションアドレスレジスタ_3	DDAR_3	32	H'FFFC64	DMAC_3	16	21φ/21φ
DMA オフセットレジスタ_3	DOFR_3	32	H'FFFC68	DMAC_3	16	21φ/21φ
DMA 転送カウントレジスタ_3	DTCR_3	32	H'FFFC6C	DMAC_3	16	21φ/21φ
DMA ブロックサイズレジスタ_3	DBSR_3	32	H'FFFC70	DMAC_3	16	21φ/21φ
DMA モードコントロールレジスタ_3	DMDR_3	32	H'FFFC74	DMAC_3	16	21φ/21φ
DMA アドレスコントロールレジスタ_3	DACR_3	32	H'FFFC78	DMAC_3	16	21φ/21φ
DMA モジュールリクエストセレクトレジスタ_0	DMRSR_0	8	H'FFFD20	DMAC_0	16	21φ/21φ
DMA モジュールリクエストセレクトレジスタ_1	DMRSR_1	8	H'FFFD21	DMAC_1	16	21φ/21φ
DMA モジュールリクエストセレクトレジスタ_2	DMRSR_2	8	H'FFFD22	DMAC_2	16	21φ/21φ
DMA モジュールリクエストセレクトレジスタ_3	DMRSR_3	8	H'FFFD23	DMAC_3	16	21φ/21φ
インタラプトプライオリティレジスタ A	IPRA	16	H'FFFD40	INTC	16	21φ/31φ
インタラプトプライオリティレジスタ B	IPRB	16	H'FFFD42	INTC	16	21φ/31φ
インタラプトプライオリティレジスタ C	IPRC	16	H'FFFD44	INTC	16	21φ/31φ
インタラプトプライオリティレジスタ D	IPRD	16	H'FFFD46	INTC	16	21φ/31φ
インタラプトプライオリティレジスタ E	IPRE	16	H'FFFD48	INTC	16	21φ/31φ
インタラプトプライオリティレジスタ F	IPRF	16	H'FFFD4A	INTC	16	21φ/31φ
インタラプトプライオリティレジスタ G	IPRG	16	H'FFFD4C	INTC	16	21φ/31φ

レジスタ名称	略称	ビット数	アドレス* ¹	モジュール	データ バス幅	アクセス* ² ステート数 (リード/ライト)
インタラプトプライオリティレジスタ I	IPRI	16	H'FFFD50	INTC	16	21φ/31φ
インタラプトプライオリティレジスタ J	IPRJ	16	H'FFFD52	INTC	16	21φ/31φ
インタラプトプライオリティレジスタ L	IPRL	16	H'FFFD56	INTC	16	21φ/31φ
インタラプトプライオリティレジスタ M	IPRM	16	H'FFFD58	INTC	16	21φ/31φ
インタラプトプライオリティレジスタ N	IPRN	16	H'FFFD5A	INTC	16	21φ/31φ
インタラプトプライオリティレジスタ O	IPRO	16	H'FFFD5C	INTC	16	21φ/31φ
インタラプトプライオリティレジスタ P	IPRP	16	H'FFFD5E	INTC	16	21φ/31φ
IRQ センスコントロールレジスタ H	ISCRH	16	H'FFFD68	INTC	16	21φ/31φ
IRQ センスコントロールレジスタ L	ISCR L	16	H'FFFD6A	INTC	16	21φ/31φ
DTC ベクタベースレジスタ	DTCVBR	32	H'FFFD80	BSC	16	21φ/31φ
バスコントロールレジスタ 2	BCR2	8	H'FFFD94	BSC	16	21φ/31φ
モードコントロールレジスタ	MDCR	16	H'FFFD C0	SYSTEM	16	21φ/31φ
システムコントロールレジスタ 0	SYSCR0	16	H'FFFD C2	SYSTEM	16	21φ/31φ
システムクロックコントロールレジスタ 0	SCKCR0	16	H'FFFD C4	SYSTEM	16	21φ/31φ
スタンバイコントロールレジスタ	SBYCR	16	H'FFFD C6	SYSTEM	16	21φ/31φ
モジュールストップコントロールレジスタ A	MSTPCRA	16	H'FFFD C8	SYSTEM	16	21φ/31φ
モジュールストップコントロールレジスタ B	MSTPCRB	16	H'FFFD CA	SYSTEM	16	21φ/31φ
モジュールストップコントロールレジスタ C	MSTPCRC	16	H'FFFD CC	SYSTEM	16	21φ/31φ
リカバリーオシレータコントロールレジスタ	ROSCCR	16	H'FFFD E0	SYSTEM	16	21φ/31φ
システムコントロールレジスタ 1	SYSCR1	16	H'FFFD E2	SYSTEM	16	21φ/31φ
システムクロックコントロールレジスタ 1	SCKCR1	16	H'FFFD E4	SYSTEM	16	21φ/31φ
モジュールストップコントロールレジスタ D	MSTPCRD	16	H'FFFD E8	SYSTEM	16	21φ/31φ
モジュールストップコントロールレジスタ E	MSTPCRE	16	H'FFFD EA	SYSTEM	16	21φ/31φ
シリアルモードレジスタ_3	SMR_3	8	H'FFFE88	SCI_3	8	2Pφ/2Pφ
ビットレートレジスタ_3	BRR_3	8	H'FFFE89	SCI_3	8	2Pφ/2Pφ
シリアルコントロールレジスタ_3	SCR_3	8	H'FFFE8A	SCI_3	8	2Pφ/2Pφ
トランスミットデータレジスタ_3	TDR_3	8	H'FFFE8B	SCI_3	8	2Pφ/2Pφ
シリアルステータスレジスタ_3	SSR_3	8	H'FFFE8C	SCI_3	8	2Pφ/2Pφ
レシーブデータレジスタ_3	RDR_3	8	H'FFFE8D	SCI_3	8	2Pφ/2Pφ
シリアルモードレジスタ_4	SMR_4	8	H'FFFE90	SCI_4	8	2Pφ/2Pφ
ビットレートレジスタ_4	BRR_4	8	H'FFFE91	SCI_4	8	2Pφ/2Pφ
シリアルコントロールレジスタ_4	SCR_4	8	H'FFFE92	SCI_4	8	2Pφ/2Pφ
トランスミットデータレジスタ_4	TDR_4	8	H'FFFE93	SCI_4	8	2Pφ/2Pφ
シリアルステータスレジスタ_4	SSR_4	8	H'FFFE94	SCI_4	8	2Pφ/2Pφ
レシーブデータレジスタ_4	RDR_4	8	H'FFFE95	SCI_4	8	2Pφ/2Pφ
タイマコントロールレジスタ_4	TCR_4	8	H'FFFE E0	TPU_4	16	2Pφ/2Pφ

レジスタ名称	略称	ビット数	アドレス* ¹	モジュール	データ バス幅	アクセス* ² ステート数 (リード/ライト)
タイマモードレジスタ_4	TMDR_4	8	H'FFFEE1	TPU_4	16	2Pφ/2Pφ
タイマ I/O コントロールレジスタ_4	TIOR_4	8	H'FFFEE2	TPU_4	16	2Pφ/2Pφ
	—		H'FFFEE3	TPU_4	16	
タイマインタラプトイネーブルレジスタ_4	TIER_4	8	H'FFFEE4	TPU_4	16	2Pφ/2Pφ
タイマステータスレジスタ_4	TSR_4	8	H'FFFEE5	TPU_4	16	2Pφ/2Pφ
タイマカウンタ_4	TCNT_4	16	H'FFFEE6	TPU_4	16	2Pφ/2Pφ
タイマジェネラルレジスタ A_4	TGRA_4	16	H'FFFEE8	TPU_4	16	2Pφ/2Pφ
タイマジェネラルレジスタ B_4	TGRB_4	16	H'FFFEEA	TPU_4	16	2Pφ/2Pφ
	—		H'FFFEEC~ H'FFFEED	TPU_4	16	
タイマコントロールレジスタ_5	TCR_5	8	H'FFFEF0	TPU_5	16	2Pφ/2Pφ
タイマモードレジスタ_5	TMDR_5	8	H'FFFEF1	TPU_5	16	2Pφ/2Pφ
タイマ I/O コントロールレジスタ_5	TIOR_5	8	H'FFFEF2	TPU_5	16	2Pφ/2Pφ
	—		H'FFFEF3	TPU_5	16	
タイマインタラプトイネーブルレジスタ_5	TIER_5	8	H'FFFEF4	TPU_5	16	2Pφ/2Pφ
タイマステータスレジスタ_5	TSR_5	8	H'FFFEF5	TPU_5	16	2Pφ/2Pφ
タイマカウンタ_5	TCNT_5	16	H'FFFEF6	TPU_5	16	2Pφ/2Pφ
タイマジェネラルレジスタ A_5	TGRA_5	16	H'FFFEF8	TPU_5	16	2Pφ/2Pφ
タイマジェネラルレジスタ B_5	TGRB_5	16	H'FFFefa	TPU_5	16	2Pφ/2Pφ
	—		H'FFFefc~ H'FFFefd	TPU_5	16	
DTC イネーブルレジスタ A	DTCERA	16	H'FFFF20	DTC	16	2Iφ/3Iφ
DTC イネーブルレジスタ B	DTCERB	16	H'FFFF22	DTC	16	2Iφ/3Iφ
DTC イネーブルレジスタ C	DTCERC	16	H'FFFF24	DTC	16	2Iφ/3Iφ
DTC イネーブルレジスタ D	DTCERD	16	H'FFFF26	DTC	16	2Iφ/3Iφ
DTC イネーブルレジスタ E	DTCERE	16	H'FFFF28	DTC	16	2Iφ/3Iφ
DTC イネーブルレジスタ F	DTCERF	16	H'FFFF2A	DTC	16	2Iφ/3Iφ
DTC イネーブルレジスタ G	DTCERG	16	H'FFFF2C	DTC	16	2Iφ/3Iφ
DTC コントロールレジスタ	DTCCR	8	H'FFFF30	DTC	16	2Iφ/3Iφ
割り込みコントロールレジスタ	INTCR	8	H'FFFF32	INTC	16	2Iφ/3Iφ
CPU プライオリティコントロールレジスタ	CPUPCR	8	H'FFFF33	INTC	16	2Iφ/3Iφ
IRQ イネーブルレジスタ	IER	16	H'FFFF34	INTC	16	2Iφ/3Iφ
IRQ ステータスレジスタ	ISR	16	H'FFFF36	INTC	16	2Iφ/3Iφ
ポート 1 レジスタ	PORT1	8	H'FFFF40	I/O ポート	8	2Pφ/-
ポート 3 レジスタ	PORT3	8	H'FFFF42	I/O ポート	8	2Pφ/-
ポート 4 レジスタ	PORT4	8	H'FFFF43	I/O ポート	8	2Pφ/-
ポート 5 レジスタ	PORT5	8	H'FFFF44	I/O ポート	8	2Pφ/-

レジスタ名称	略称	ビット数	アドレス* ¹	モジュール	データ バス幅	アクセス* ² ステート数 (リード/ライト)
ポート 6 レジスタ	PORT6	8	H'FFFF45	I/O ポート	8	2Pφ/-
ポート A レジスタ	PORTA	8	H'FFFF49	I/O ポート	8	2Pφ/-
ポート D レジスタ	PORTD	8	H'FFFF4C	I/O ポート	8	2Pφ/-
ポート 1 データレジスタ	P1DR	8	H'FFFF50	I/O ポート	8	2Pφ/2Pφ
ポート 3 データレジスタ	P3DR	8	H'FFFF52	I/O ポート	8	2Pφ/2Pφ
ポート 6 データレジスタ	P6DR	8	H'FFFF55	I/O ポート	8	2Pφ/2Pφ
ポート A データレジスタ	PADR	8	H'FFFF59	I/O ポート	8	2Pφ/2Pφ
ポート D データレジスタ	PDDR	8	H'FFFF5C	I/O ポート	8	2Pφ/2Pφ
PPG 出力コントロールレジスタ	PCR	8	H'FFFF76	PPG	8	2Pφ/2Pφ
PPG 出力モードレジスタ	PMR	8	H'FFFF77	PPG	8	2Pφ/2Pφ
ネクストデータインーブルレジスタ H	NDERH	8	H'FFFF78	PPG	8	2Pφ/2Pφ
ネクストデータインーブルレジスタ L	NDERL	8	H'FFFF79	PPG	8	2Pφ/2Pφ
アウトプットデータレジスタ H	PODRH	8	H'FFFF7A	PPG	8	2Pφ/2Pφ
アウトプットデータレジスタ L	PODRL	8	H'FFFF7B	PPG	8	2Pφ/2Pφ
ネクストデータレジスタ H* ³	NDRH	8	H'FFFF7C	PPG	8	2Pφ/2Pφ
ネクストデータレジスタ L* ³	NDRL	8	H'FFFF7D	PPG	8	2Pφ/2Pφ
ネクストデータレジスタ H* ³	NDRH	8	H'FFFF7E	PPG	8	2Pφ/2Pφ
ネクストデータレジスタ L* ³	NDRL	8	H'FFFF7F	PPG	8	2Pφ/2Pφ
タイマコントロール/ステータスレジスタ	TCSR	8	H'FFFFA4	WDT	16	2Pφ/3Pφ
タイマカウンタ	TCNT	8	H'FFFFA5	WDT	16	2Pφ/3Pφ
リセットコントロール/ステータスレジスタ	RSTCSR	8	H'FFFFA7	WDT	16	2Pφ/3Pφ
タイマスタートレジスタ	TSTR	8	H'FFFFBC	TPU (ユニット 0)	16	2Pφ/2Pφ
タイマシンクロレジスタ	TSYR	8	H'FFFFBD	TPU (ユニット 0)	16	2Pφ/2Pφ
タイマコントロールレジスタ_0	TCR_0	8	H'FFFFC0	TPU_0	16	2Pφ/2Pφ
タイマモードレジスタ_0	TMDR_0	8	H'FFFFC1	TPU_0	16	2Pφ/2Pφ
タイマ I/O コントロールレジスタ H_0	TIORH_0	8	H'FFFFC2	TPU_0	16	2Pφ/2Pφ
タイマ I/O コントロールレジスタ L_0	TIORL_0	8	H'FFFFC3	TPU_0	16	2Pφ/2Pφ
タイマインタラプトインーブルレジスタ_0	TIER_0	8	H'FFFFC4	TPU_0	16	2Pφ/2Pφ
タイマステータスレジスタ_0	TSR_0	8	H'FFFFC5	TPU_0	16	2Pφ/2Pφ
タイマカウンタ_0	TCNT_0	16	H'FFFFC6	TPU_0	16	2Pφ/2Pφ
タイマジェネラルレジスタ A_0	TGRA_0	16	H'FFFFC8	TPU_0	16	2Pφ/2Pφ
タイマジェネラルレジスタ B_0	TGRB_0	16	H'FFFFCA	TPU_0	16	2Pφ/2Pφ
タイマジェネラルレジスタ C_0	TGRC_0	16	H'FFFFCC	TPU_0	16	2Pφ/2Pφ
タイマジェネラルレジスタ D_0	TGRD_0	16	H'FFFFCE	TPU_0	16	2Pφ/2Pφ
タイマコントロールレジスタ_1	TCR_1	8	H'FFFFD0	TPU_1	16	2Pφ/2Pφ
タイマモードレジスタ_1	TMDR_1	8	H'FFFFD1	TPU_1	16	2Pφ/2Pφ

レジスタ名称	略称	ビット数	アドレス* ¹	モジュール	データ バス幅	アクセス* ² ステート数 (リード/ライト)
タイマ I/O コントロールレジスタ_1	TIOR_1	8	H'FFFFD2	TPU_1	16	2Pφ/2Pφ
	—		H'FFFFD3	TPU_1	16	
タイマインタラプトイネーブルレジスタ_1	TIER_1	8	H'FFFFD4	TPU_1	16	2Pφ/2Pφ
タイマステータスレジスタ_1	TSR_1	8	H'FFFFD5	TPU_1	16	2Pφ/2Pφ
タイマカウンタ_1	TCNT_1	16	H'FFFFD6	TPU_1	16	2Pφ/2Pφ
タイマジェネラルレジスタ A_1	TGRA_1	16	H'FFFFD8	TPU_1	16	2Pφ/2Pφ
タイマジェネラルレジスタ B_1	TGRB_1	16	H'FFFFDA	TPU_1	16	2Pφ/2Pφ
	—		H'FFFFDC~ H'FFFFDF	TPU_1	16	
タイマコントロールレジスタ_2	TCR_2	8	H'FFFFE0	TPU_2	16	2Pφ/2Pφ
タイマモードレジスタ_2	TMDR_2	8	H'FFFFE1	TPU_2	16	2Pφ/2Pφ
タイマ I/O コントロールレジスタ_2	TIOR_2	8	H'FFFFE2	TPU_2	16	2Pφ/2Pφ
	—		H'FFFFE3	TPU_2	16	
タイマインタラプトイネーブルレジスタ_2	TIER_2	8	H'FFFFE4	TPU_2	16	2Pφ/2Pφ
タイマステータスレジスタ_2	TSR_2	8	H'FFFFE5	TPU_2	16	2Pφ/2Pφ
タイマカウンタ_2	TCNT_2	16	H'FFFFE6	TPU_2	16	2Pφ/2Pφ
タイマジェネラルレジスタ A_2	TGRA_2	16	H'FFFFE8	TPU_2	16	2Pφ/2Pφ
タイマジェネラルレジスタ B_2	TGRB_2	16	H'FFFFEA	TPU_2	16	2Pφ/2Pφ
	—		H'FFFFEC~ H'FFFFEF	TPU_2	16	
タイマコントロールレジスタ_3	TCR_3	8	H'FFFFF0	TPU_3	16	2Pφ/2Pφ
タイマモードレジスタ_3	TMDR_3	8	H'FFFFF1	TPU_3	16	2Pφ/2Pφ
タイマ I/O コントロールレジスタ H_3	TIORH_3	8	H'FFFFF2	TPU_3	16	2Pφ/2Pφ
タイマ I/O コントロールレジスタ L_3	TIORL_3	8	H'FFFFF3	TPU_3	16	2Pφ/2Pφ
タイマインタラプトイネーブルレジスタ_3	TIER_3	8	H'FFFFF4	TPU_3	16	2Pφ/2Pφ
タイマステータスレジスタ_3	TSR_3	8	H'FFFFF5	TPU_3	16	2Pφ/2Pφ
タイマカウンタ_3	TCNT_3	16	H'FFFFF6	TPU_3	16	2Pφ/2Pφ
タイマジェネラルレジスタ A_3	TGRA_3	16	H'FFFFF8	TPU_3	16	2Pφ/2Pφ
タイマジェネラルレジスタ B_3	TGRB_3	16	H'FFFFFA	TPU_3	16	2Pφ/2Pφ
タイマジェネラルレジスタ C_3	TGRC_3	16	H'FFFFFC	TPU_3	16	2Pφ/2Pφ
タイマジェネラルレジスタ D_3	TGRD_3	16	H'FFFFFE	TPU_3	16	2Pφ/2Pφ

【注】 *1 アドレスの下位 24 ビットを示しています。

*2 RCAN-TL1_0、RCAN-TL1_1 のメールボックス領域のアクセスは、0~5Pφのウェイトが発生することがあります。

*3 PCR の設定によりパルス出力グループ 2 とパルス出力グループ 3 の出力トリガが同一の場合は、NDRH のアドレスは H'FFF7C となり、出力トリガが異なる場合は、グループ 2 に対応する NDRH のアドレスは H'FFF7E、グループ 3 に対する NDRH のアドレスは H'FFF7C となります。同様に、PCR の設定によりパルス出力グループ 0 とパルス出力グループ 1 の出力トリガが同一の場合は、NDRL のアドレスは H'FFF7D となり、出力トリガが異なる場合は、グループ 0 に対応する NDRL のアドレスは H'FFF7F、グループ 1 に対する NDRL のアドレスは H'FFF7D となります。

24.2 レジスタビット一覧

周辺モジュールのレジスタのアドレスとビット名を以下に示します。

16 ビット、32 ビットレジスタは、8 ビットずつ 2 段または 4 段で表しています。

レジスタ 略称	ビット 31/23/15/7	ビット 30/22/14/6	ビット 29/21/13/5	ビット 28/20/12/4	ビット 27/19/11/3	ビット 26/18/10/2	ビット 25/17/9/1	ビット 24/16/8/0	モジュール
MCR_0	MCR15	MCR14	—	—	—	TST2	TST1	TST0	RCAN- TL1_0
	MCR7	MCR6	MCR5	—	—	MCR2	MCR1	MCR0	
GSR_0	—	—	—	—	—	—	—	—	
	—	—	GSR5	GSR4	GSR3	GSR2	GSR1	GSR0	
BCR1_0	TSG13	TSG12	TSG11	TSG10	—	TSG22	TSG21	TSG20	
	—	—	SJW1	SJW0	—	—	—	BSP	
BCR0_0	—	—	—	—	—	—	—	—	
	BRP7	BRP6	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0	
IRR_0	IRR15	IRR14	IRR13	IRR12	IRR11	IRR10	IRR9	IRR8	
	IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0	
IMR_0	IMR15	IMR14	IMR13	IMR12	IMR11	IMR10	IMR9	IMR8	
	IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	IMR0	
TEC_0	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0	
REC_0	REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0	
TXPR1_0	TXPR1_15	TXPR1_14	TXPR1_13	TXPR1_12	TXPR1_11	TXPR1_10	TXPR1_9	TXPR1_8	
	TXPR1_7	TXPR1_6	TXPR1_5	TXPR1_4	TXPR1_3	TXPR1_2	TXPR1_1	TXPR1_0	
TXPR0_0	TXPR0_15	TXPR0_14	TXPR0_13	TXPR0_12	TXPR0_11	TXPR0_10	TXPR0_9	TXPR0_8	
	TXPR0_7	TXPR0_6	TXPR0_5	TXPR0_4	TXPR0_3	TXPR0_2	TXPR0_1	—	
TXCR1_0	TXCR1_15	TXCR1_14	TXCR1_13	TXCR1_12	TXCR1_11	TXCR1_10	TXCR1_9	TXCR1_8	
	TXCR1_7	TXCR1_6	TXCR1_5	TXCR1_4	TXCR1_3	TXCR1_2	TXCR1_1	TXCR1_0	
TXCR0_0	TXCR0_15	TXCR0_14	TXCR0_13	TXCR0_12	TXCR0_11	TXCR0_10	TXCR0_9	TXCR0_8	
	TXCR0_7	TXCR0_6	TXCR0_5	TXCR0_4	TXCR0_3	TXCR0_2	TXCR0_1	—	
TXACK1_0	TXACK1_15	TXACK1_14	TXACK1_13	TXACK1_12	TXACK1_11	TXACK1_10	TXACK1_9	TXACK1_8	
	TXACK1_7	TXACK1_6	TXACK1_5	TXACK1_4	TXACK1_3	TXACK1_2	TXACK1_1	TXACK1_0	
TXACK0_0	TXACK0_15	TXACK0_14	TXACK0_13	TXACK0_12	TXACK0_11	TXACK0_10	TXACK0_9	TXACK0_8	
	TXACK0_7	TXACK0_6	TXACK0_5	TXACK0_4	TXACK0_3	TXACK0_2	TXACK0_1	—	
ABACK1_0	ABACK1_15	ABACK1_14	ABACK1_13	ABACK1_12	ABACK1_11	ABACK1_10	ABACK1_9	ABACK1_8	
	ABACK1_7	ABACK1_6	ABACK1_5	ABACK1_4	ABACK1_3	ABACK1_2	ABACK1_1	ABACK1_0	
ABACK0_0	ABACK0_15	ABACK0_14	ABACK0_13	ABACK0_12	ABACK0_11	ABACK0_10	ABACK0_9	ABACK0_8	
	ABACK0_7	ABACK0_6	ABACK0_5	ABACK0_4	ABACK0_3	ABACK0_2	ABACK0_1	—	
RXPR1_0	RXPR1_15	RXPR1_14	RXPR1_13	RXPR1_12	RXPR1_11	RXPR1_10	RXPR1_9	RXPR1_8	
	RXPR1_7	RXPR1_6	RXPR1_5	RXPR1_4	RXPR1_3	RXPR1_2	RXPR1_1	RXPR1_0	

レジスタ 略称	ビット 31/23/15/7	ビット 30/22/14/6	ビット 29/21/13/5	ビット 28/20/12/4	ビット 27/19/11/3	ビット 26/18/10/2	ビット 25/17/9/1	ビット 24/16/8/0	モジュール
RXPR0_0	RXPR0_15	RXPR0_14	RXPR0_13	RXPR0_12	RXPR0_11	RXPR0_10	RXPR0_9	RXPR0_8	RCAN- TL1_0
	RXPR0_7	RXPR0_6	RXPR0_5	RXPR0_4	RXPR0_3	RXPR0_2	RXPR0_1	RXPR0_0	
RFPR1_0	RFPR1_15	RFPR1_14	RFPR1_13	RFPR1_12	RFPR1_11	RFPR1_10	RFPR1_9	RFPR1_8	
	RFPR1_7	RFPR1_6	RFPR1_5	RFPR1_4	RFPR1_3	RFPR1_2	RFPR1_1	RFPR1_0	
RFPR0_0	RFPR0_15	RFPR0_14	RFPR0_13	RFPR0_12	RFPR0_11	RFPR0_10	RFPR0_9	RFPR0_8	
	RFPR0_7	RFPR0_6	RFPR0_5	RFPR0_4	RFPR0_3	RFPR0_2	RFPR0_1	RFPR0_0	
MBIMR1_0	MBIMR1_15	MBIMR1_14	MBIMR1_13	MBIMR1_12	MBIMR1_11	MBIMR1_10	MBIMR1_9	MBIMR1_8	
	MBIMR1_7	MBIMR1_6	MBIMR1_5	MBIMR1_4	MBIMR1_3	MBIMR1_2	MBIMR1_1	MBIMR1_0	
MBIMR0_0	MBIMR0_15	MBIMR0_14	MBIMR0_13	MBIMR0_12	MBIMR0_11	MBIMR0_10	MBIMR0_9	MBIMR0_8	
	MBIMR0_7	MBIMR0_6	MBIMR0_5	MBIMR0_4	MBIMR0_3	MBIMR0_2	MBIMR0_1	MBIMR0_0	
UMSR1_0	UMSR1_15	UMSR1_14	UMSR1_13	UMSR1_12	UMSR1_11	UMSR1_10	UMSR1_9	UMSR1_8	
	UMSR1_7	UMSR1_6	UMSR1_5	UMSR1_4	UMSR1_3	UMSR1_2	UMSR1_1	UMSR1_0	
UMSR0_0	UMSR0_15	UMSR0_14	UMSR0_13	UMSR0_12	UMSR0_11	UMSR0_10	UMSR0_9	UMSR0_8	
	UMSR0_7	UMSR0_6	UMSR0_5	UMSR0_4	UMSR0_3	UMSR0_2	UMSR0_1	UMSR0_0	
TTCR0_0	TCR15	TCR14	TCR13	TCR12	TCR11	TCR10	—	—	
	—	TCR6	TPSC5	TPSC4	TPSC3	TPSC2	TPSC1	TPSC0	
CMAX_	—	—	—	—	—	CMAX2	CMAX1	CMAX0	
TEW_0	—	—	—	—	TEW3	TEW2	TEW1	TEW0	
RFTR0FF_0	RFTR0FF7	RFTR0FF6	RFTR0FF5	RFTR0FF4	RFTR0FF3	RFTR0FF2	RFTR0FF1	RFTR0FF0	
	—	—	—	—	—	—	—	—	
TSR_0	—	—	—	—	—	—	—	—	
	—	—	—	TSR4	TSR3	TSR2	TSR1	TSR0	
CCR_0	—	—	—	—	—	—	—	—	
	—	—	CCR5	CCR4	CCR3	CCR2	CCR1	CCR0	
TCNTR_0	TCNTR15	TCNTR14	TCNTR13	TCNTR12	TCNTR11	TCNTR10	TCNTR9	TCNTR8	
	TCNTR7	TCNTR6	TCNTR5	TCNTR4	TCNTR3	TCNTR2	TCNTR1	TCNTR0	
CYCTR_0	CYCTR15	CYCTR14	CYCTR13	CYCTR12	CYCTR11	CYCTR10	CYCTR9	CYCTR8	
	CYCTR7	CYCTR6	CYCTR5	CYCTR4	CYCTR3	CYCTR2	CYCTR1	CYCTR0	
RFMK_0	RFMK15	RFMK14	RFMK13	RFMK12	RFMK11	RFMK10	RFMK9	RFMK8	
	RFMK7	RFMK6	RFMK5	RFMK4	RFMK3	RFMK2	RFMK1	RFMK0	
TCMR0_0	TCMR0_15	TCMR0_14	TCMR0_13	TCMR0_12	TCMR0_11	TCMR0_10	TCMR0_9	TCMR0_8	
	TCMR0_7	TCMR0_6	TCMR0_5	TCMR0_4	TCMR0_3	TCMR0_2	TCMR0_1	TCMR0_0	
TCMR1_0	TCMR1_15	TCMR1_14	TCMR1_13	TCMR1_12	TCMR1_11	TCMR1_10	TCMR1_9	TCMR1_8	
	TCMR1_7	TCMR1_6	TCMR1_5	TCMR1_4	TCMR1_3	TCMR1_2	TCMR1_1	TCMR1_0	
TCMR2_0	TCMR2_15	TCMR2_14	TCMR2_13	TCMR2_12	TCMR2_11	TCMR2_10	TCMR2_9	TCMR2_8	
	TCMR2_7	TCMR2_6	TCMR2_5	TCMR2_4	TCMR2_3	TCMR2_2	TCMR2_1	TCMR2_0	

レジスタ 略称	ビット 31/23/15/7	ビット 30/22/14/6	ビット 29/21/13/5	ビット 28/20/12/4	ビット 27/19/11/3	ビット 26/18/10/2	ビット 25/17/9/1	ビット 24/16/8/0	モジュール
TTTSEL_0	TTTSEL15	TTTSEL14	TTTSEL13	TTTSEL12	TTTSEL11	TTTSEL10	TTTSEL9	TTTSEL8	RCAN- TL1_0
	TTTSEL7	TTTSEL6	TTTSEL5	TTTSEL4	TTTSEL3	TTTSEL2	TTTSEL1	TTTSEL0	
MB_0[0]. CONTROL0H	IDE	RTR	—	STDID10	STDID9	STDID8	STDID7	STDID6	
MB_0[0]. CONTROL0L	EXTID15	EXTID14	EXTID13	EXTID12	EXTID11	EXTID10	EXTID9	EXTID8	
MB_0[0]. LAFMH	IDE_LAFM	—	—	STDID_	STDID_	STDID_	STDID_	STDID_	
				LAFM10	LAFM9	LAFM8	LAFM7	LAFM6	
	STDID_	STDID_	STDID_	STDID_	STDID_	STDID_	EXTID_	EXTID_	
MB_0[0]. LAFML	LAFM5	LAFM4	LAFM3	LAFM2	LAFM1	LAFM0	LAFM17	LAFM16	
	EXTID_	EXTID_	EXTID_	EXTID_	EXTID_	EXTID_	EXTID_	EXTID_	
	LAFM15	LAFM14	LAFM13	LAFM12	LAFM11	LAFM10	LAFM9	LAFM8	
	EXTID_	EXTID_	EXTID_	EXTID_	EXTID_	EXTID_	EXTID_	EXTID_	
	LAFM7	LAFM6	LAFM5	LAFM4	LAFM3	LAFM2	LAFM1	LAFM0	
MB_0[0]. MSG_DATA[0]	MSG_DATA0								
MB_0[0]. MSG_DATA[1]	MSG_DATA1								
MB_0[0]. MSG_DATA[2]	MSG_DATA2								
MB_0[0]. MSG_DATA[3]	MSG_DATA3								
MB_0[0]. MSG_DATA[4]	MSG_DATA4								
MB_0[0]. MSG_DATA[5]	MSG_DATA5								
MB_0[0]. MSG_DATA[6]	MSG_DATA6								
MB_0[0]. MSG_DATA[7]	MSG_DATA7								
MB_0[0]. CONTROL1H	—	—	NMC	—	—	MBC2	MBC1	MBC0	
MB_0[0]. CONTROL1L	—	—	—	—	DLC3	DLC2	DLC1	DLC0	
MB_0[0]. TIMESTAMP	TS15	TS14	TS13	TS12	TS11	TS10	TS9	TS8	
	TS7	TS6	TS5	TS4	TS3	TS2	TS1	TS0	

レジスタ 略称	ビット 31/23/15/7	ビット 30/22/14/6	ビット 29/21/13/5	ビット 28/20/12/4	ビット 27/19/11/3	ビット 26/18/10/2	ビット 25/17/9/1	ビット 24/16/8/0	モジュール
MB_0[1]. CONTROL0H	IDE	RTR	—	STDID10	STDID9	STDID8	STDID7	STDID6	RCAN- TL1_0
	STDID5	STDID4	STDID3	STDID2	STDID1	STDID0	EXTID17	EXTID16	
MB_0[1]. CONTROL0L	EXTID15	EXTID14	EXTID13	EXTID12	EXTID11	EXTID10	EXTID9	EXTID8	
	EXTID7	EXTID6	EXTID5	EXTID4	EXTID3	EXTID2	EXTID1	EXTID0	
MB_0[1]. LAFMH	IDE_LAFM	—	—	STDID_	STDID_	STDID_	STDID_	STDID_	
				LAFM10	LAFM9	LAFM8	LAFM7	LAFM6	
	STDID_	STDID_	STDID_	STDID_	STDID_	STDID_	EXTID_	EXTID_	
	LAFM5	LAFM4	LAFM3	LAFM2	LAFM1	LAFM0	LAFM17	LAFM16	
MB_0[1]. LAFML	EXTID_	EXTID_	EXTID_	EXTID_	EXTID_	EXTID_	EXTID_	EXTID_	
	LAFM15	LAFM14	LAFM13	LAFM12	LAFM11	LAFM10	LAFM9	LAFM8	
	EXTID_	EXTID_	EXTID_	EXTID_	EXTID_	EXTID_	EXTID_	EXTID_	
	LAFM7	LAFM6	LAFM5	LAFM4	LAFM3	LAFM2	LAFM1	LAFM0	
MB_0[1]. MSG_DATA[0]	MSG_DATA0								
MB_0[1]. MSG_DATA[1]	MSG_DATA1								
MB_0[1]. MSG_DATA[2]	MSG_DATA2								
MB_0[1]. MSG_DATA[3]	MSG_DATA3								
MB_0[1]. MSG_DATA[4]	MSG_DATA4								
MB_0[1]. MSG_DATA[5]	MSG_DATA5								
MB_0[1]. MSG_DATA[6]	MSG_DATA6								
MB_0[1]. MSG_DATA[7]	MSG_DATA7								
MB_0[1]. CONTROL1H	—	—	NMC	ATX	DART	MBC2	MBC1	MBC0	
MB_0[1]. CONTROL1L	—	—	—	—	DLC3	DLC2	DLC1	DLC0	
MB_0[1]. TIMESTAMP	TS15	TS14	TS13	TS12	TS11	TS10	TS9	TS8	
	TS7	TS6	TS5	TS4	TS3	TS2	TS1	TS0	

レジスタ 略称	ビット 31/23/15/7	ビット 30/22/14/6	ビット 29/21/13/5	ビット 28/20/12/4	ビット 27/19/11/3	ビット 26/18/10/2	ビット 25/17/9/1	ビット 24/16/8/0	モジュール
MB_0[2]. CONTROL0H ~ MB_0[2]. TIMESTAMP			MB_0[1].CONTROL0H~MB_0[1].TIMESTAMP と同じビット構成						RCAN- TL1_0
↓ (繰り返し)									
MB_0[15]. CONTROL0H ~ MB_0[15]. TIMESTAMP			MB_0[1].CONTROL0H~MB_0[1].TIMESTAMP と同じビット構成						
MB_0[16]. CONTROL0H	IDE	RTR	—	STDID10	STDID9	STDID8	STDID7	STDID6	
MB_0[16]. CONTROL0L	STDID5	STDID4	STDID3	STDID2	STDID1	STDID0	EXTID17	EXTID16	
MB_0[16]. CONTROL0L	EXTID15	EXTID14	EXTID13	EXTID12	EXTID11	EXTID10	EXTID9	EXTID8	
MB_0[16]. LAFMH	EXTID7	EXTID6	EXTID5	EXTID4	EXTID3	EXTID2	EXTID1	EXTID0	
	IDE_LAFM	—	—	STDID_	STDID_	STDID_	STDID_	STDID_	STDID_
				LAFM10	LAFM9	LAFM8	LAFM7	LAFM6	
	STDID_	STDID_	STDID_	STDID_	STDID_	STDID_	EXTID_	EXTID_	
	LAFM5	LAFM4	LAFM3	LAFM2	LAFM1	LAFM0	LAFM17	LAFM16	
MB_0[16]. LAFML	EXTID_	EXTID_	EXTID_	EXTID_	EXTID_	EXTID_	EXTID_	EXTID_	
	LAFM15	LAFM14	LAFM13	LAFM12	LAFM11	LAFM10	LAFM9	LAFM8	
	EXTID_	EXTID_	EXTID_	EXTID_	EXTID_	EXTID_	EXTID_	EXTID_	
	LAFM7	LAFM6	LAFM5	LAFM4	LAFM3	LAFM2	LAFM1	LAFM0	
MB_0[16]. MSG_DATA[0]	MSG_DATA0								
MB_0[16]. MSG_DATA[1]	MSG_DATA1								
MB_0[16]. MSG_DATA[2]	MSG_DATA2								
MB_0[16]. MSG_DATA[3]	MSG_DATA3								
MB_0[16]. MSG_DATA[4]	MSG_DATA4								
MB_0[16]. MSG_DATA[5]	MSG_DATA5								
MB_0[16]. MSG_DATA[6]	MSG_DATA6								

レジスタ 略称	ビット 31/23/15/7	ビット 30/22/14/6	ビット 29/21/13/5	ビット 28/20/12/4	ビット 27/19/11/3	ビット 26/18/10/2	ビット 25/17/9/1	ビット 24/16/8/0	モジュール
MB_0[16]. MSG_DATA[7]									RCAN- TL1_0
MB_0[16]. CONTROL1H	—	—	NMC	ATX	DART	MBC2	MBC1	MBC0	
MB_0[16]. CONTROL1L	—	—	—	—	DLC3	DLC2	DLC1	DLC0	
MB_0[17]. CONTROL0H ~ MB_0[17]. CONTROL1L	MB_0[16].CONTROL0H~MB_0[16].CONTROL1L と同じビット構成								
↓	(繰り返し)								
MB_0[23]. CONTROL0H ~ MB_0[23]. CONTROL1L	MB_0[16].CONTROL0H~MB_0[16].CONTROL1L と同じビット構成								
MB_0[24]. CONTROL0H	IDE	RTR	—	STDID10	STDID9	STDID8	STDID7	STDID6	
MB_0[24]. CONTROL0L	STDID5	STDID4	STDID3	STDID2	STDID1	STDID0	EXTID17	EXTID16	
MB_0[24]. LAFMH	EXTID15	EXTID14	EXTID13	EXTID12	EXTID11	EXTID10	EXTID9	EXTID8	
MB_0[24]. LAFML	EXTID7	EXTID6	EXTID5	EXTID4	EXTID3	EXTID2	EXTID1	EXTID0	
MB_0[24]. MSG_DATA[0]	IDE_LAFM	—	—	STDID_	STDID_	STDID_	STDID_	STDID_	
MB_0[24]. MSG_DATA[1]	LAFM5	LAFM4	LAFM3	LAFM2	LAFM1	LAFM0	LAFM17	LAFM16	
MB_0[24]. MSG_DATA[2]	EXTID_	EXTID_	EXTID_	EXTID_	EXTID_	EXTID_	EXTID_	EXTID_	
MB_0[24]. MSG_DATA[3]	LAFM15	LAFM14	LAFM13	LAFM12	LAFM11	LAFM10	LAFM9	LAFM8	
	EXTID_	EXTID_	EXTID_	EXTID_	EXTID_	EXTID_	EXTID_	EXTID_	
	LAFM7	LAFM6	LAFM5	LAFM4	LAFM3	LAFM2	LAFM1	LAFM0	
MB_0[24]. MSG_DATA[0]	MSG_DATA0								
MB_0[24]. MSG_DATA[1]	MSG_DATA1								
MB_0[24]. MSG_DATA[2]	MSG_DATA2								
MB_0[24]. MSG_DATA[3]	MSG_DATA3								

レジスタ 略称	ビット 31/23/15/7	ビット 30/22/14/6	ビット 29/21/13/5	ビット 28/20/12/4	ビット 27/19/11/3	ビット 26/18/10/2	ビット 25/17/9/1	ビット 24/16/8/0	モジュール
MB_0[24]. MSG_DATA[4]					MSG_DATA4				RCAN- TL1_0
MB_0[24]. MSG_DATA[5]					MSG_DATA5				
MB_0[24]. MSG_DATA[6]					MSG_DATA6				
MB_0[24]. MSG_DATA[7]					MSG_DATA7				
MB_0[24]. CONTROL1H	—	—	NMC	ATX	DART	MBC2	MBC1	MBC0	
MB_0[24]. CONTROL1L	—	—	—	—	DLC3	DLC2	DLC1	DLC0	
MB_0[24]. TTT	TTT15	TTT14	TTT13	TTT12	TTT11	TTT10	TTT9	TTT8	
	TTT7	TTT6	TTT5	TTT4	TTT3	TTT2	TTT1	TTT0	
MB_0[24]. TT_CONTROL	TTW1	TTW0	OFFSET5	OFFSET4	OFFSET3	OFFSET2	OFFSET1	OFFSET0	
	—	—	—	—	—	REP_ FACTOR2	REP_ FACTOR1	REP_ FACTOR0	
MB_0[25]. CONTROL0H ~ MB_0[25]. TT_CONTROL	MB_0[24].CONTROL0H~MB_0[24]. TT_CONTROL と同じビット構成								
↓	(繰り返し)								
MB_0[29]. CONTROL0H ~ MB_0[29]. TT_CONTROL	MB_0[24].CONTROL0H~MB_0[24]. TT_CONTROL と同じビット構成								
MB_0[30]. CONTROL0H	IDE	RTR	—	STDID10	STDID9	STDID8	STDID7	STDID6	
	STDID5	STDID4	STDID3	STDID2	STDID1	STDID0	EXTID17	EXTID16	
MB_0[30]. CONTROL0L	EXTID15	EXTID14	EXTID13	EXTID12	EXTID11	EXTID10	EXTID9	EXTID8	
	EXTID7	EXTID6	EXTID5	EXTID4	EXTID3	EXTID2	EXTID1	EXTID0	
MB_0[30]. LAFMH	IDE_LAFM	—	—	STDID_ LAFM10	STDID_ LAFM9	STDID_ LAFM8	STDID_ LAFM7	STDID_ LAFM6	
	STDID_ LAFM5	STDID_ LAFM4	STDID_ LAFM3	STDID_ LAFM2	STDID_ LAFM1	STDID_ LAFM0	EXTID_ LAFM17	EXTID_ LAFM16	

レジスタ 略称	ビット 31/23/15/7	ビット 30/22/14/6	ビット 29/21/13/5	ビット 28/20/12/4	ビット 27/19/11/3	ビット 26/18/10/2	ビット 25/17/9/1	ビット 24/16/8/0	モジュール
MB_0[30]. LAFML	EXTID_ LAFM15	EXTID_ LAFM14	EXTID_ LAFM13	EXTID_ LAFM12	EXTID_ LAFM11	EXTID_ LAFM10	EXTID_ LAFM9	EXTID_ LAFM8	RCAN- TL1_0
	EXTID_ LAFM7	EXTID_ LAFM6	EXTID_ LAFM5	EXTID_ LAFM4	EXTID_ LAFM3	EXTID_ LAFM2	EXTID_ LAFM1	EXTID_ LAFM0	
MB_0[30]. MSG_DATA[0]	MSG_DATA0								
MB_0[30]. MSG_DATA[1]	MSG_DATA1								
MB_0[30]. MSG_DATA[2]	MSG_DATA2								
MB_0[30]. MSG_DATA[3]	MSG_DATA3								
MB_0[30]. MSG_DATA[4]	MSG_DATA4								
MB_0[30]. MSG_DATA[5]	MSG_DATA5								
MB_0[30]. MSG_DATA[6]	MSG_DATA6								
MB_0[30]. MSG_DATA[7]	MSG_DATA7								
MB_0[30]. CONTROL1H	—	—	NMC	ATX	DART	MBC2	MBC1	MBC0	
MB_0[30]. CONTROL1L	—	—	—	—	DLC3	DLC2	DLC1	DLC0	
MB_0[30]. TIMESTAMP	TS15 TS7	TS14 TS6	TS13 TS5	TS12 TS4	TS11 TS3	TS10 TS2	TS9 TS1	TS8 TS0	
MB_0[30]. TTT	TTT15 TTT7	TTT14 TTT6	TTT13 TTT5	TTT12 TTT4	TTT11 TTT3	TTT10 TTT2	TTT9 TTT1	TTT8 TTT0	
MB_0[31]. CONTROLOH	IDE STDID5	RTR STDID4	— STDID3	STDID10 STDID2	STDID9 STDID1	STDID8 STDID0	STDID7 EXTID17	STDID6 EXTID16	
MB_0[31]. CONTROL0L	EXTID15 EXTID7	EXTID14 EXTID6	EXTID13 EXTID5	EXTID12 EXTID4	EXTID11 EXTID3	EXTID10 EXTID2	EXTID9 EXTID1	EXTID8 EXTID0	
MB_0[31]. LAFMH	IDE_LAFM STDID_ LAFM5	— STDID_ LAFM4	— STDID_ LAFM3	STDID_ LAFM10 LAFM2	STDID_ LAFM9 LAFM1	STDID_ LAFM8 LAFM0	STDID_ LAFM7 LAFM17	STDID_ LAFM6 LAFM16	

レジスタ 略称	ビット 31/23/15/7	ビット 30/22/14/6	ビット 29/21/13/5	ビット 28/20/12/4	ビット 27/19/11/3	ビット 26/18/10/2	ビット 25/17/9/1	ビット 24/16/8/0	モジュール
MB_0[31]. LAFML	EXTID_ LAFM15	EXTID_ LAFM14	EXTID_ LAFM13	EXTID_ LAFM12	EXTID_ LAFM11	EXTID_ LAFM10	EXTID_ LAFM9	EXTID_ LAFM8	RCAN- TL1_0
	EXTID_ LAFM7	EXTID_ LAFM6	EXTID_ LAFM5	EXTID_ LAFM4	EXTID_ LAFM3	EXTID_ LAFM2	EXTID_ LAFM1	EXTID_ LAFM0	
MB_0[31]. MSG_DATA[0]	MSG_DATA0								
MB_0[31]. MSG_DATA[1]	MSG_DATA1								
MB_0[31]. MSG_DATA[2]	MSG_DATA2								
MB_0[31]. MSG_DATA[3]	MSG_DATA3								
MB_0[31]. MSG_DATA[4]	MSG_DATA4								
MB_0[31]. MSG_DATA[5]	MSG_DATA5								
MB_0[31]. MSG_DATA[6]	MSG_DATA6								
MB_0[31]. MSG_DATA[7]	MSG_DATA7								
MB_0[31]. CONTROL1H	—	—	NMC	ATX	DART	MBC2	MBC1	MBC0	
MB_0[31]. CONTROL1L	—	—	—	—	DLC3	DLC2	DLC1	DLC0	
MB_0[31]. TIMESTAMP	TS15 TS7	TS14 TS6	TS13 TS5	TS12 TS4	TS11 TS3	TS10 TS2	TS9 TS1	TS8 TS0	
RCANMON_0	—	CTxSTP	RCANE	—	—	—	CTxD	CRxD	
RCANMON_1	—	CTxSTP	RCANE	—	—	—	CTxD	CRxD	RCAN- TL1_1
MCR_1	RCAN-TL1_1 は RCAN-TL1_0 と同じビット構成								
↓									
MB_1[31]. CONTROL0H									
~									
MB_1[31]. TIMESTAMP									

レジスタ 略称	ビット 31/23/15/7	ビット 30/22/14/6	ビット 29/21/13/5	ビット 28/20/12/4	ビット 27/19/11/3	ビット 26/18/10/2	ビット 25/17/9/1	ビット 24/16/8/0	モジュール
SPCRA	SPRIE	SPE	SPTIE	SPEIE	MSTR	MODFEN	TXMD	SPMS	RSPI
SSLPA	—	—	—	—	SSL3P	SSL2P	SSL1P	SSL0P	
SPPCRA	—	—	MOIFE	MOIFV	—	SPOM	SPLP2	SPLP	
SPSRA	SPRF	—	SPTIEF	—	PERF	MODF	IDLNF	OVRF	
SPDRA	SPD31	SPD30	SPD29	SPD28	SPD27	SPD26	SPD25	SPD24	
	SPD23	SPD22	SPD21	SPD20	SPD19	SPD18	SPD17	SPD16	
	SPD15	SPD14	SPD13	SPD12	SPD11	SPD10	SPD9	SPD8	
	SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0	
SPSCRA	—	—	—	—	—	SPSLN2	SPSLN1	SPSLN0	
SPSSRA	—	SPECM2	SPECM1	SPECM0	—	SPCP2	SPCP1	SPCP0	
SPBRA	SPR7	SPR6	SPR5	SPR4	SPR3	SPR2	SPR1	SPR0	
SPDCRA	—	—	SPLW	SPRDTD	SLSEL1	SLSEL0	SPFC1	SPFC0	
SPCKDA	—	—	—	—	—	SCKDL2	SCKDL1	SCKDL0	
SSLNDA	—	—	—	—	—	SLNDL2	SLNDL1	SLNDL0	
SPNDA	—	—	—	—	—	SPNDL2	SPNDL1	SPNDL0	
SPCR2A	—	—	—	—	PTE	SPIIE	SPOE	SPPE	
SPCMDA0	SCKDEN_0	SLNDEN_0	SPNDEN_0	LSBF_0	SPB3_0	SPB2_0	SPB1_0	SPB0_0	
	SSLKP_0	—	SSLA1_0	SSLA0_0	BRDV1_0	BRDV0_0	CPOL_0	CPHA_0	
SPCMDA1	SCKDEN_1	SLNDEN_1	SPNDEN_1	LSBF_1	SPB3_1	SPB2_1	SPB1_1	SPB0_1	
	SSLKP_1	—	SSLA1_1	SSLA0_1	BRDV1_1	BRDV0_1	CPOL_1	CPHA_1	
SPCMDA2	SCKDEN_2	SLNDEN_2	SPNDEN_2	LSBF_2	SPB3_2	SPB2_2	SPB1_2	SPB0_2	
	SSLKP_2	—	SSLA1_2	SSLA0_2	BRDV1_2	BRDV0_2	CPOL_2	CPHA_2	
SPCMDA3	SCKDEN_3	SLNDEN_3	SPNDEN_3	LSBF_3	SPB3_3	SPB2_3	SPB1_3	SPB0_3	
	SSLKP_3	—	SSLA1_3	SSLA0_3	BRDV1_3	BRDV0_3	CPOL_3	CPHA_3	
SPCMDA4	SCKDEN_4	SLNDEN_4	SPNDEN_4	LSBF_4	SPB3_4	SPB2_4	SPB1_4	SPB0_4	
	SSLKP_4	—	SSLA1_4	SSLA0_4	BRDV1_4	BRDV0_4	CPOL_4	CPHA_4	
SPCMDA5	SCKDEN_5	SLNDEN_5	SPNDEN_5	LSBF_5	SPB3_5	SPB2_5	SPB1_5	SPB0_5	
	SSLKP_5	—	SSLA1_5	SSLA0_5	BRDV1_5	BRDV0_5	CPOL_5	CPHA_5	
SPCMDA6	SCKDEN_6	SLNDEN_6	SPNDEN_6	LSBF_6	SPB3_6	SPB2_6	SPB1_6	SPB0_6	
	SSLKP_6	—	SSLA1_6	SSLA0_6	BRDV1_6	BRDV0_6	CPOL_6	CPHA_6	
SPCMDA7	SCKDEN_7	SLNDEN_7	SPNDEN_7	LSBF_7	SPB3_7	SPB2_7	SPB1_7	SPB0_7	
	SSLKP_7	—	SSLA1_7	SSLA0_7	BRDV1_7	BRDV0_7	CPOL_7	CPHA_7	
SPCRB	SPRIE	SPE	SPTIE	SPEIE	MSTR	MODFEN	TXMD	SPMS	
SSLPB	—	—	—	—	SSL3P	SSL2P	SSL1P	SSL0P	
SPPCRB	—	—	MOIFE	MOIFV	—	SPOM	SPLP2	SPLP	
SPSRB	SPRF	—	SPTIEF	—	PERF	MODF	IDLNF	OVRF	

レジスタ 略称	ビット 31/23/15/7	ビット 30/22/14/6	ビット 29/21/13/5	ビット 28/20/12/4	ビット 27/19/11/3	ビット 26/18/10/2	ビット 25/17/9/1	ビット 24/16/8/0	モジュール
SPDRB	SPD31	SPD30	SPD29	SPD28	SPD27	SPD26	SPD25	SPD24	RSPI
	SPD23	SPD22	SPD21	SPD20	SPD19	SPD18	SPD17	SPD16	
	SPD15	SPD14	SPD13	SPD12	SPD11	SPD10	SPD9	SPD8	
	SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0	
SPSCR	—	—	—	—	—	SPSLN2	SPSLN1	SPSLN0	
SPSSRB	—	SPECM12	SPECM1	SPECM0	—	SPCP2	SPCP1	SPCP0	
SPBRB	SPR7	SPR6	SPR5	SPR4	SPR3	SPR2	SPR1	SPR0	
SPDCRB	—	—	SPLW	SPRDTD	SLSEL1	SLSEL0	SPFC1	SPFC0	
SPCKDB	—	—	—	—	—	SCKDL2	SCKDL1	SCKDL0	
SSLNDB	—	—	—	—	—	SLNDL2	SLNDL1	SLNDL0	
SPNDB	—	—	—	—	—	SPNDL2	SPNDL1	SPNDL0	
SPCR2B	—	—	—	—	PTE	SPIIE	SPOE	SPPE	
SPCMDB0	SCKDEN_0	SLNDEN_0	SPNDEN_0	LSBF_0	SPB3_0	SPB2_0	SPB1_0	SPB0_0	
	SSLKP_0	—	SSLA1_0	SSLA0_0	BRDV1_0	BRDV0_0	CPOL_0	CPHA_0	
SPCMDB1	SCKDEN_1	SLNDEN_1	SPNDEN_1	LSBF_1	SPB3_1	SPB2_1	SPB1_1	SPB0_1	
	SSLKP_1	—	SSLA1_1	SSLA0_1	BRDV1_1	BRDV0_1	CPOL_1	CPHA_1	
SPCMDB2	SCKDEN_2	SLNDEN_2	SPNDEN_2	LSBF_2	SPB3_2	SPB2_2	SPB1_2	SPB0_2	
	SSLKP_2	—	SSLA1_2	SSLA0_2	BRDV1_2	BRDV0_2	CPOL_2	CPHA_2	
SPCMDB3	SCKDEN_3	SLNDEN_3	SPNDEN_3	LSBF_3	SPB3_3	SPB2_3	SPB1_3	SPB0_3	
	SSLKP_3	—	SSLA1_3	SSLA0_3	BRDV1_3	BRDV0_3	CPOL_3	CPHA_3	
SPCMDB4	SCKDEN_4	SLNDEN_4	SPNDEN_4	LSBF_4	SPB3_4	SPB2_4	SPB1_4	SPB0_4	
	SSLKP_4	—	SSLA1_4	SSLA0_4	BRDV1_4	BRDV0_4	CPOL_4	CPHA_4	
SPCMDB5	SCKDEN_5	SLNDEN_5	SPNDEN_5	LSBF_5	SPB3_5	SPB2_5	SPB1_5	SPB0_5	
	SSLKP_5	—	SSLA1_5	SSLA0_5	BRDV1_5	BRDV0_5	CPOL_5	CPHA_5	
SPCMDB6	SCKDEN_6	SLNDEN_6	SPNDEN_6	LSBF_6	SPB3_6	SPB2_6	SPB1_6	SPB0_6	
	SSLKP_6	—	SSLA1_6	SSLA0_6	BRDV1_6	BRDV0_6	CPOL_6	CPHA_6	
SPCMDB7	SCKDEN_7	SLNDEN_7	SPNDEN_7	LSBF_7	SPB3_7	SPB2_7	SPB1_7	SPB0_7	
	SSLKP_7	—	SSLA1_7	SSLA0_7	BRDV1_7	BRDV0_7	CPOL_7	CPHA_7	
SPCRC	SPRIE	SPE	SPTIE	SPEIE	MSTR	MODFEN	TXMD	SPMS	
SSLPC	—	—	—	—	SSL3P	SSL2P	SSL1P	SSL0P	
SPPCRC	—	—	MOIFE	MOIFV	—	SPOM	SPLP2	SPLP	
SPSRC	SPRF	—	SPTIEF	—	PERF	MODF	IDLNF	OVRF	
SPDRB	SPD31	SPD30	SPD29	SPD28	SPD27	SPD26	SPD25	SPD24	
	SPD23	SPD22	SPD21	SPD20	SPD19	SPD18	SPD17	SPD16	
	SPD15	SPD14	SPD13	SPD12	SPD11	SPD10	SPD9	SPD8	
	SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0	

レジスタ 略称	ビット 31/23/15/7	ビット 30/22/14/6	ビット 29/21/13/5	ビット 28/20/12/4	ビット 27/19/11/3	ビット 26/18/10/2	ビット 25/17/9/1	ビット 24/16/8/0	モジュール
SPSCRC	—	—	—	—	—	SPSLN2	SPSLN1	SPSLN0	RSPI
SPSSRC	—	SPECM02	SPECM1	SPECM0	—	SPCP2	SPCP1	SPCP0	
SPBRC	SPR7	SPR6	SPR5	SPR4	SPR3	SPR2	SPR1	SPR0	
SPDCRC	—	—	SPLW	SPRDTD	SLSEL1	SLSEL0	SPFC1	SPFC0	
SPCKDC	—	—	—	—	—	SCKDL2	SCKDL1	SCKDL0	
SSLNDC	—	—	—	—	—	SLNDL2	SLNDL1	SLNDL0	
SPNDC	—	—	—	—	—	SPNDL2	SPNDL1	SPNDL0	
SPCR2C	—	—	—	—	PTE	SPIIE	SPOE	SPPE	
SPCMDC0	SCKDEN_0	SLNDEN_0	SPNDEN_0	LSBF_0	SPB3_0	SPB2_0	SPB1_0	SPB0_0	
	SSLKP_0	—	SSLA1_0	SSLA0_0	BRDV1_0	BRDV0_0	CPOL_0	CPHA_0	
SPCMDC1	SCKDEN_1	SLNDEN_1	SPNDEN_1	LSBF_1	SPB3_1	SPB2_1	SPB1_1	SPB0_1	
	SSLKP_1	—	SSLA1_1	SSLA0_1	BRDV1_1	BRDV0_1	CPOL_1	CPHA_1	
SPCMDC2	SCKDEN_2	SLNDEN_2	SPNDEN_2	LSBF_2	SPB3_2	SPB2_2	SPB1_2	SPB0_2	
	SSLKP_2	—	SSLA1_2	SSLA0_2	BRDV1_2	BRDV0_2	CPOL_2	CPHA_2	
SPCMDC3	SCKDEN_3	SLNDEN_3	SPNDEN_3	LSBF_3	SPB3_3	SPB2_3	SPB1_3	SPB0_3	
	SSLKP_3	—	SSLA1_3	SSLA0_3	BRDV1_3	BRDV0_3	CPOL_3	CPHA_3	
SPCMDC4	SCKDEN_4	SLNDEN_4	SPNDEN_4	LSBF_4	SPB3_4	SPB2_4	SPB1_4	SPB0_4	
	SSLKP_4	—	SSLA1_4	SSLA0_4	BRDV1_4	BRDV0_4	CPOL_4	CPHA_4	
SPCMDC5	SCKDEN_5	SLNDEN_5	SPNDEN_5	LSBF_5	SPB3_5	SPB2_5	SPB1_5	SPB0_5	
	SSLKP_5	—	SSLA1_5	SSLA0_5	BRDV1_5	BRDV0_5	CPOL_5	CPHA_5	
SPCMDC6	SCKDEN_6	SLNDEN_6	SPNDEN_6	LSBF_6	SPB3_6	SPB2_6	SPB1_6	SPB0_6	
	SSLKP_6	—	SSLA1_6	SSLA0_6	BRDV1_6	BRDV0_6	CPOL_6	CPHA_6	
SPCMDC7	SCKDEN_7	SLNDEN_7	SPNDEN_7	LSBF_7	SPB3_7	SPB2_7	SPB1_7	SPB0_7	
	SSLKP_7	—	SSLA1_7	SSLA0_7	BRDV1_7	BRDV0_7	CPOL_7	CPHA_7	
SPCRD	SPRIE	SPE	SPTIE	SPEIE	MSTR	MODFEN	TXMD	SPMS	
SSLPD	—	—	—	—	SSL3P	SSL2P	SSL1P	SSL0P	
SPPCRD	—	—	MOIFE	MOIFV	—	SPOM	SPLP2	SPLP	
SPSRD	SPRF	—	SPTF	—	PERF	MODF	IDLNF	OVRF	
SPDRD	SPD31	SPD30	SPD29	SPD28	SPD27	SPD26	SPD25	SPD24	
	SPD23	SPD22	SPD21	SPD20	SPD19	SPD18	SPD17	SPD16	
	SPD15	SPD14	SPD13	SPD12	SPD11	SPD10	SPD9	SPD8	
	SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0	
SPSCRD	—	—	—	—	—	SPSLN12	SPSLN1	SPSLN0	
SPSSRD	—	SPECM12	SPECM1	SPECM0	—	SPCP12	SPCP1	SPCP0	
SPBRD	SPR7	SPR6	SPR5	SPR4	SPR3	SPR2	SPR1	SPR0	
SPDCRD	—	—	SPLW	SPRDTD	SLSEL1	SLSEL0	SPFC1	SPFC0	

レジスタ 略称	ビット 31/23/15/7	ビット 30/22/14/6	ビット 29/21/13/5	ビット 28/20/12/4	ビット 27/19/11/3	ビット 26/18/10/2	ビット 25/17/9/1	ビット 24/16/8/0	モジュール
SPCKDD	—	—	—	—	—	SCKDL2	SCKDL1	SCKDL0	RSPI
SSLNDD	—	—	—	—	—	SLNDL2	SLNDL1	SLNDL0	
SPNDD	—	—	—	—	—	SPNDL2	SPNDL1	SPNDL0	
SPCR2D	—	—	—	—	PTE	SPIIE	SPOE	SPPE	
SPCMDD0	SCKDEN_0	SLNDEN_0	SPNDEN_0	LSBF_0	SPB3_0	SPB2_0	SPB1_0	SPB0_0	
	SSLKP_0	—	SSLA1_0	SSLA0_0	BRDV1_0	BRDV0_0	CPOL_0	CPHA_0	
SPCMDD1	SCKDEN_1	SLNDEN_1	SPNDEN_1	LSBF_1	SPB3_1	SPB2_1	SPB1_1	SPB0_1	
	SSLKP_1	—	SSLA1_1	SSLA0_1	BRDV1_1	BRDV0_1	CPOL_1	CPHA_1	
SPCMDD2	SCKDEN_2	SLNDEN_2	SPNDEN_2	LSBF_2	SPB3_2	SPB2_2	SPB1_2	SPB0_2	
	SSLKP_2	—	SSLA1_2	SSLA0_2	BRDV1_2	BRDV0_2	CPOL_2	CPHA_2	
SPCMDD3	SCKDEN_3	SLNDEN_3	SPNDEN_3	LSBF_3	SPB3_3	SPB2_3	SPB1_3	SPB0_3	
	SSLKP_3	—	SSLA1_3	SSLA0_3	BRDV1_3	BRDV0_3	CPOL_3	CPHA_3	
SPCMDD4	SCKDEN_4	SLNDEN_4	SPNDEN_4	LSBF_4	SPB3_4	SPB2_4	SPB1_4	SPB0_4	
	SSLKP_4	—	SSLA1_4	SSLA0_4	BRDV1_4	BRDV0_4	CPOL_4	CPHA_4	
SPCMDD5	SCKDEN_5	SLNDEN_5	SPNDEN_5	LSBF_5	SPB3_5	SPB2_5	SPB1_5	SPB0_5	
	SSLKP_5	—	SSLA1_5	SSLA0_5	BRDV1_5	BRDV0_5	CPOL_5	CPHA_5	
SPCMDD6	SCKDEN_6	SLNDEN_6	SPNDEN_6	LSBF_6	SPB3_6	SPB2_6	SPB1_6	SPB0_6	
	SSLKP_6	—	SSLA1_6	SSLA0_6	BRDV1_6	BRDV0_6	CPOL_6	CPHA_6	
SPCMDD7	SCKDEN_7	SLNDEN_7	SPNDEN_7	LSBF_7	SPB3_7	SPB2_7	SPB1_7	SPB0_7	
	SSLKP_7	—	SSLA1_7	SSLA0_7	BRDV1_7	BRDV0_7	CPOL_7	CPHA_7	
LINCR	LINE	MSS	RXDMRS	SBSTR	RXDSF	BCIE	SBIE	SFIE	HWLIN
LINSTR	BCE	—	TOIE	RXDS	TOER	BCER	SBEND	SFEND	
LINTCR	TOCSTR	TCK1	TCK0	—	—	—	TCSTF	TCSTR	
LINTCNT									
LINTCNT									
CRCR	DORCLR					LMS	G1	G0	CRC
CRCDIR									
CRCDOR									
FMODR	—	—	—	FRDMD	—	—	—	—	FLASH/ EEPROM
FASTAT	ROMAE	—	—	CMDLK	EEPAAE	—	EEPRPE	EEPWPE	
FAEINT	ROMAEIE	—	—	CMDLKIE	EEPAAEIE	—	EEPRPEIE	EEPWPEIE	
ROMMAT	RMKEY7	RMKEY6	RMKEY5	RMKEY4	RMKEY3	RMKEY2	RMKEY1	RMKEY0	
	—	—	—	—	—	—	—	ROMSEL	

レジスタ 略称	ビット 31/23/15/7	ビット 30/22/14/6	ビット 29/21/13/5	ビット 28/20/12/4	ビット 27/19/11/3	ビット 26/18/10/2	ビット 25/17/9/1	ビット 24/16/8/0	モジュール
EEPRE0	REKEY07	REKEY06	REKEY05	REKEY04	REKEY03	REKEY02	REKEY01	REKEY00	FLASH/ EEPROM
	DBRE07	DBRE06	DBRE05	DBRE04	DBRE03	DBRE02	DBRE01	DBRE00	
EEPRE1	REKEY15	REKEY14	REKEY13	REKEY12	REKEY11	REKEY10	REKEY09	REKEY08	
	DBRE15	DBRE14	DBRE13	DBRE12	DBRE11	DBRE10	DBRE09	DBRE08	
EEPWE0	WEKEY07	WEKEY06	WEKEY05	WEKEY04	WEKEY03	WEKEY02	WEKEY01	WEKEY00	
	DBWE07	DBWE06	DBWE05	DBWE04	DBWE03	DBWE02	DBWE01	DBWE00	
EEPWE1	WEKEY15	WEKEY14	WE0KEY13	WEKEY12	WEKEY11	WEKEY10	WEKEY09	WEKEY08	
	DBWE15	DBWE14	DBWE13	DBWE12	DBWE11	DBWE10	DBWE09	DBWE08	
FCURAME	FCKEY7	FCKEY6	FCKEY5	FCKEY4	FCKEY3	FCKEY2	FCKEY1	FCKEY0	
	—	—	—	—	—	—	—	FCRME	
FSTSTR0	FRDY	ILGLERR	ERSERR	PRGERR	SUSRDY	—	ERSSPD	PRGSPD	
FSTATR1	FCUERR	—	—	FLOCKST	—	—	FRDTCT	FRCRCT	
FENTRYR	FEKEY7	FEKEY6	FEKEY5	FEKEY4	FEKEY3	FEKEY2	FEKEY1	FEKEY0	
	FENTRYD	—	—	—	—	—	—	FENTRY0	
FPROTR	FPKEY7	FPKEY6	FPKEY5	FPKEY4	FPKEY3	FPKEY2	FPKEY1	FPKEY0	
	—	—	—	—	—	—	—	FPROTCN	
FRESETR	FRKEY7	FRKEY6	FRKEY5	FRKEY4	FRKEY3	FRKEY2	FRKEY1	FRKEY0	
	—	—	—	—	—	—	—	FRESET	
FCMDR	CMDR7	CMDR6	CMDR5	CMDR4	CMDR3	CMDR2	CMDR1	CMDR0	
	PCMDR7	PCMDR6	PCMDR5	PCMDR4	PCMDR3	PCMDR2	PCMDR1	PCMDR0	
FRAMECCR	—	—	—	—	—	—	FRDCLE	FRCCLE	
FCPSR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	ESUSPMD	
EEPBCCNT	—	—	BCADR13	BCADR12	BCADR11	BCADR10	BCADR09	BCADR08	
	BCADR07	BCADR06	BCADR05	BCADR04	BCADR03	—	—	BCSIZE	
FPESTAT	—	—	—	—	—	—	—	—	
	PEERRST7	PEERRST6	PEERRST5	PEERRST4	PEERRST3	PEERRST2	PEERRST1	PEERRST0	
EEPBCSTAT	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	BCST	
FCKAR	—	—	—	—	—	—	—	—	
	FCKA7	FCKA6	FCKA5	FCKA4	FCKA3	FCKA2	FCKA1	FCKA0	
EEPMAT	EMKEY7	EMKEY6	EMKEY5	EMKEY4	EMKEY3	EMKEY2	EMKEY1	EMKEY0	
	—	—	—	—	—	—	—	EEPSEL	
RAMEN	RNKEY7	RNKEY6	RNKEY5	RNKEY4	RNKEY3	RNKEY2	RNKEY1	RNKEY0	RAM
	—	—	—	RAME4	RAME3	RAME2	RAME1	RAME0	

レジスタ 略称	ビット 31/23/15/7	ビット 30/22/14/6	ビット 29/21/13/5	ビット 28/20/12/4	ビット 27/19/11/3	ビット 26/18/10/2	ビット 25/17/9/1	ビット 24/16/8/0	モジュール
RAMWEN	RWNKEY7	RWNKEY6	RWNKEY5	RWNKEY4	RWNKEY3	RWNKEY2	RWNKEY1	RWNKEY0	RAM
	—	—	—	RAMWE4	RAMWE3	RAMWE2	RAMWE1	RAMWE0	
RAMECC	REKEY7	REKEY6	REKEY5	REKEY4	REKEY3	REKEY2	REKEY1	REKEY0	
	—	—	—	—	—	—	—	RECCA	
RAMERR	—	—	—	—	—	—	RDTCT	RERRC	
RAMINT	—	—	—	—	—	—	REDIE	RINTC	
RAMCYC	RAKEY7	RAKEY6	RAKEY5	RAKEY4	RAKEY3	RAKEY2	RAKEY1	RAKEY0	
	—	—	WRCYC1	WRCYC0	—	—	—	—	
ADDRA_0			—	—	—	—	—	—	A/D_0
ADDRB_0			—	—	—	—	—	—	
ADDRC_0			—	—	—	—	—	—	
ADDRD_0			—	—	—	—	—	—	
ADDRE_0			—	—	—	—	—	—	
ADDRF_0			—	—	—	—	—	—	
ADDRG_0			—	—	—	—	—	—	
ADDRH_0			—	—	—	—	—	—	
ADCSR_0	ADF	ADIE	ADST	—	CH3	CH2	CH1	CH0	
ADCR_0	TRGS1	TRGS0	SCANE	SCANS	CKS1	CKS0	ADSTCLR	EXTRGS	
APPDCR_0	AN7PD	AN6PD	AN5PD	AN4PD	AN3PD	AN2PD	AN1PD	AN0PD	
ADDIAGR_0							DIAG1	DIAG0	
ADDRA_1			—	—	—	—	—	—	A/D_1
ADDRB_1			—	—	—	—	—	—	
ADDRC_1			—	—	—	—	—	—	
ADDRD_1			—	—	—	—	—	—	
ADDRE_1			—	—	—	—	—	—	
ADDRF_1			—	—	—	—	—	—	
ADDRG_1			—	—	—	—	—	—	
ADDRH_1			—	—	—	—	—	—	
ADCSR_1	ADF	ADIE	ADST	—	CH3	CH2	CH1	CH0	
ADCR_1	TRGS1	TRGS0	SCANE	SCANS	CKS1	CKS0	ADSTCLR	EXTRGS	
APPDCR_1	AN15PD	AN14PD	AN13PD	AN12PD	A11PD	AN10PD	AN9PD	AN8PD	

レジスタ 略称	ビット 31/23/15/7	ビット 30/22/14/6	ビット 29/21/13/5	ビット 28/20/12/4	ビット 27/19/11/3	ビット 26/18/10/2	ビット 25/17/9/1	ビット 24/16/8/0	モジュール
ADDIAGR_1							DIAG1	DIAG0	A/D_1
TSTRB	—	—	CST5	CST4	CST3	CST2	CST1	CST0	TPU
TSYRB	—	—	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0	(ユニット1)
TCR_6	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_6
TMDR_6	—	—	BFB	BFA	MD3	MD2	MD1	MD0	
TIORH_6	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIORL_6	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0	
TIER_6	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
TSR_6	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TCNT_6	_____								
TGRA_6	_____								
TGRB_6	_____								
TGRC_6	_____								
TGRD_6	_____								
TCR_7	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_7
TMDR_7	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_7	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_7	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_7	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_7	_____								
TGRA_7	_____								
TGRB_7	_____								
TCR_8	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_8
TMDR_8	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_8	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_8	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_8	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	

レジスタ 略称	ビット 31/23/15/7	ビット 30/22/14/6	ビット 29/21/13/5	ビット 28/20/12/4	ビット 27/19/11/3	ビット 26/18/10/2	ビット 25/17/9/1	ビット 24/16/8/0	モジュール
TCNT_8									TPU_8
TGRA_8									
TGRB_8									
TCR_9	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_9
TMDR_9	—	—	BFB	BFA	MD3	MD2	MD1	MD0	
TIORH_9	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIORL_9	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0	
TIER_9	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
TSR_9	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TCNT_9									
TGRA_9									
TGRB_9									
TGRC_9									
TGRD_9									
TCR_10	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_10
TMDR_10	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_10	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_10	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_10	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_10									
TGRA_10									
TGRB_10									
TCR_11	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_11
TMDR_11	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_11	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	

レジスタ 略称	ビット 31/23/15/7	ビット 30/22/14/6	ビット 29/21/13/5	ビット 28/20/12/4	ビット 27/19/11/3	ビット 26/18/10/2	ビット 25/17/9/1	ビット 24/16/8/0	モジュール	
TIER_11	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	TPU_11	
TSR_11	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA		
TCNT_11	_____									
TGRA_11	_____									
TGRB_11	_____									
P1DDR	P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR	I/O ポート	
P3DDR	P37DDR	P36DDR	P35DDR	P34DDR	P33DDR	P32DDR	P31DDR	P30DDR		
P6DDR	—	P66DDR	P65DDR	P64DDR	P63DDR	P62DDR	P61DDR	P60DDR		
PADDR	PA7DDR	PA6DDR	PA5DDR	PA4DDR	PA3DDR	PA2DDR	PA1DDR	—		
PDDDR	PD7DDR	PD6DDR	PD5DDR	PD4DDR	PD3DDR	PD2DDR	PD1DDR	PD0DDR		
P1ICR	P17ICR	P16ICR	P15ICR	P14ICR	P13ICR	P12ICR	P11ICR	P10ICR		
P3ICR	P37ICR	P36ICR	P35ICR	P34ICR	P33ICR	P32ICR	P31ICR	P30ICR		
P4ICR	P47ICR	P46ICR	P45ICR	P44ICR	P43ICR	P42ICR	P41ICR	P40ICR		
P5ICR	P57ICR	P56ICR	P55ICR	P54ICR	P53ICR	P52ICR	P51ICR	P50ICR		
P6ICR	—	P66ICR	P65ICR	P64ICR	P63ICR	P62ICR	P61ICR	P60ICR		
PAICR	PA7ICR	PA6ICR	PA5ICR	PA4ICR	PA3ICR	PA2ICR	PA1ICR	—		
PDICR	PD7ICR	PD6ICR	PD5ICR	PD4ICR	PD3ICR	PD2ICR	PD1ICR	PD0ICR		
PORTH	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0		
PORTJ	PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0		
PORTK	PK7	PK6	PK5	PK4	PK3	PK2	PK1	PK0		
PHDR	PH7DR	PH6DR	PH5DR	PH4DR	PH3DR	PH2DR	PH1DR	PH0DR		
PJDR	PJ7DR	PJ6DR	PJ5DR	PJ4DR	PJ3DR	PJ2DR	PJ1DR	PJ0DR		
PKDR	PK7DR	PK6DR	PK5DR	PK4DR	PK3DR	PK2DR	PK1DR	PK0DR		
PHDDR	PH7DDR	PH6DDR	PH5DDR	PH4DDR	PH3DDR	PH2DDR	PH1DDR	PH0DDR		
PJDDR	PJ7DDR	PJ6DDR	PJ5DDR	PJ4DDR	PJ3DDR	PJ2DDR	PJ1DDR	PJ0DDR		
PKDDR	PK7DDR	PK6DDR	PK5DDR	PK4DDR	PK3DDR	PK2DDR	PK1DDR	PK0DDR		
PHICR	PH7ICR	PH6ICR	PH5ICR	PH4ICR	PH3ICR	PH2ICR	PH1ICR	PH0ICR		
PJICR	PJ7ICR	PJ6ICR	PJ5ICR	PJ4ICR	PJ3ICR	PJ2ICR	PJ1ICR	PJ0ICR		
PKICR	PK7ICR	PK6ICR	PK5ICR	PK4ICR	PK3ICR	PK2ICR	PK1ICR	PK0ICR		
PDPCR	PD7PCR	PD6PCR	PD5PCR	PD4PCR	PD3PCR	PD2PCR	PD1PCR	PD0PCR		
PHPCR	PH7PCR	PH6PCR	PH5PCR	PH4PCR	PH3PCR	PH2PCR	PH1PCR	PH0PCR		
PJPCR	PJ7PCR	PJ6PCR	PJ5PCR	PJ4PCR	PJ3PCR	PJ2PCR	PJ1PCR	PJ0PCR		
PKPCR	PK7PCR	PK6PCR	PK5PCR	PK4PCR	PK3PCR	PK2PCR	PK1PCR	PK0PCR		

レジスタ 略称	ビット 31/23/15/7	ビット 30/22/14/6	ビット 29/21/13/5	ビット 28/20/12/4	ビット 27/19/11/3	ビット 26/18/10/2	ビット 25/17/9/1	ビット 24/16/8/0	モジュール
PFCR5	—	—	—	RCANMD	—	—	RCAN1S	RCAN0S	I/O ポート
PFCR6	—	—	—	—	TCLKS	—	—	—	
PFCR8	—	—	HLIS2	HLIS1	HLIS0	RSPISA	—	—	
PFCR9	—	—	—	—	TPUMS2	TPUMS1	TPUMS0A	TPUMS0B	
PFCRA	TPUMS11	TPUMS10	TPUMS9A	TPUMS9B	TPUMS8	TPUMS7	TPUMS6A	TPUMS6B	
PFCRB	—	ITS14	ITS13	ITS12	ITS11	ITS10	ITS9	ITS8	
PFCRC	ITS7	ITS6	ITS5	ITS4	ITS3	ITS2	ITS1	ITS0	
PFCRD	—	—	—	—	—	—	—	DRVDWNE	
SSIER	—	SSI14	SSI13	SSI12	SSI11	SSI10	SSI9	SSI8	INTC
	SSI7	SSI6	SSI5	SSI4	SSI3	SSI2	SSI1	SSI0	
P1DSR	P17DSR	P16DSR	P15DSR	P14DSR	P13DSR	P12DSR	P11DSR	P10DSR	I/O ポート
P6DSR	—	P66DSR	P65DSR	P64DSR	P63DSR	P62DSR	P61DSR	P60DSR	
PADSR	PA7DSR	PA6DSR	PA5DSR	PA4DSR	PA3DSR	PA2DSR	PA1DSR	—	
PDDSR	PD7DSR	PD6DSR	PD5DSR	PD4DSR	PD3DSR	PD2DSR	PD1DSR	PD0DSR	
PHDSR	PH7DSR	PH6DSR	PH5DSR	PH4DSR	PH3DSR	PH2DSR	PH1DSR	PH0DSR	
P1PSR	P17PSR	P16PSR	P15PSR	P14PSR	P13PSR	P12PSR	P11PSR	P10PSR	
P6PSR	—	P66PSR	P65PSR	P64PSR	P63PSR	P62PSR	P61PSR	P60PSR	
PAPSR	PA7PSR	PA6PSR	PA5PSR	PA4PSR	PA3PSR	PA2PSR	PA1PSR	—	
PDPSR	PD7PSR	PD6PSR	PD5PSR	PD4PSR	PD3PSR	PD2PSR	PD1PSR	PD0PSR	
PHPSR	PH7PSR	PH6PSR	PH5PSR	PH4PSR	PH3PSR	PH2PSR	PH1PSR	PH0PSR	
DSAR_0									DMAC_0
DDAR_0									
DOFR_0									
DTCR_0									

レジスタ 略称	ビット 31/23/15/7	ビット 30/22/14/6	ビット 29/21/13/5	ビット 28/20/12/4	ビット 27/19/11/3	ビット 26/18/10/2	ビット 25/17/9/1	ビット 24/16/8/0	モジュール
DBSR_0	BKSZH31	BKSZH30	BKSZH29	BKSZH28	BKSZH27	BKSZH26	BKSZH25	BKSZH24	DMAC_0
	BKSZH23	BKSZH22	BKSZH21	BKSZH20	BKSZH19	BKSZH18	BKSZH17	BKSZH16	
	BKSZ15	BKSZ14	BKSZ13	BKSZ12	BKSZ11	BKSZ10	BKSZ9	BKSZ8	
	BKSZ7	BKSZ6	BKSZ5	BKSZ4	BKSZ3	BKSZ2	BKSZ1	BKSZ0	
DMDR_0	DTE	DACKE	TENDE	—	DREQS	NRD	—	—	
	ACT	—	—	—	ERRF	—	ESIF	DTIF	
	DTSZ1	DTSZ0	MDS1	MDS0	TSEIE	—	ESIE	DTIE	
	DTF1	DTF0	DTA	—	—	DMAP2	DMAP1	DMAP0	
DACR_0	AMS	DIRS	—	—	—	RPTIE	ARS1	ARS0	
	—	—	SAT1	SAT0	—	—	DAT1	DAT0	
	SARIE	—	—	SARA4	SARA3	SARA2	SARA1	SARA0	
	DARIE	—	—	DARA4	DARA3	DARA2	DARA1	DARA0	
DSAR_1								DMAC_1	
DDAR_1									
DOFR_1									
DTCR_1									
DBSR_1	BKSZH31	BKSZH30	BKSZH29	BKSZH28	BKSZH27	BKSZH26	BKSZH25	BKSZH24	
	BKSZH23	BKSZH22	BKSZH21	BKSZH20	BKSZH19	BKSZH18	BKSZH17	BKSZH16	
	BKSZ15	BKSZ14	BKSZ13	BKSZ12	BKSZ11	BKSZ10	BKSZ9	BKSZ8	
	BKSZ7	BKSZ6	BKSZ5	BKSZ4	BKSZ3	BKSZ2	BKSZ1	BKSZ0	
DMDR_1	DTE	DACKE	TENDE	—	DREQS	NRD	—	—	
	ACT	—	—	—	—	—	ESIF	DTIF	
	DTSZ1	DTSZ0	MDS1	MDS0	TSEIE	—	ESIE	DTIE	
	DTF1	DTF0	DTA	—	—	DMAP2	DMAP1	DMAP0	

レジスタ 略称	ビット 31/23/15/7	ビット 30/22/14/6	ビット 29/21/13/5	ビット 28/20/12/4	ビット 27/19/11/3	ビット 26/18/10/2	ビット 25/17/9/1	ビット 24/16/8/0	モジュール
DACR_1	AMS	DIRS	—	—	—	RPTIE	ARS1	ARS0	DMAC_1
	—	—	SAT1	SAT0	—	—	DAT1	DAT0	
	SARIE	—	—	SARA4	SARA3	SARA2	SARA1	SARA0	
	DARIE	—	—	DARA4	DARA3	DARA2	DARA1	DARA0	
DSAR_2									DMAC_2
DDAR_2									
DOFR_2									
DTCR_2									
DBSR_2	BKSZH31	BKSZH30	BKSZH29	BKSZH28	BKSZH27	BKSZH26	BKSZH25	BKSZH24	
	BKSZH23	BKSZH22	BKSZH21	BKSZH20	BKSZH19	BKSZH18	BKSZH17	BKSZH16	
	BKSZ15	BKSZ14	BKSZ13	BKSZ12	BKSZ11	BKSZ10	BKSZ9	BKSZ8	
	BKSZ7	BKSZ6	BKSZ5	BKSZ4	BKSZ3	BKSZ2	BKSZ1	BKSZ0	
DMDR_2	DTE	DACKE	TENDE	—	DREQS	NRD	—	—	
	ACT	—	—	—	—	—	ESIF	DTIF	
	DTSZ1	DTSZ0	MDS1	MDS0	TSEIE	—	ESIE	DTIE	
	DTF1	DTF0	DTA	—	—	DMAP2	DMAP1	DMAP0	
DACR_2	AMS	DIRS	—	—	—	RPTIE	ARS1	ARS0	
	—	—	SAT1	SAT0	—	—	DAT1	DAT0	
	SARIE	—	—	SARA4	SARA3	SARA2	SARA1	SARA0	
	DARIE	—	—	DARA4	DARA3	DARA2	DARA1	DARA0	
DSAR_3									DMAC_3

レジスタ 略称	ビット 31/23/15/7	ビット 30/22/14/6	ビット 29/21/13/5	ビット 28/20/12/4	ビット 27/19/11/3	ビット 26/18/10/2	ビット 25/17/9/1	ビット 24/16/8/0	モジュール
DDAR_3									DMAC_3
DOFR_3									
DTCR_3									
DBSR_3	BKSZH31	BKSZH30	BKSZH29	BKSZH28	BKSZH27	BKSZH26	BKSZH25	BKSZH24	
	BKSZH23	BKSZH22	BKSZH21	BKSZH20	BKSZH19	BKSZH18	BKSZH17	BKSZH16	
	BKSZ15	BKSZ14	BKSZ13	BKSZ12	BKSZ11	BKSZ10	BKSZ9	BKSZ8	
	BKSZ7	BKSZ6	BKSZ5	BKSZ4	BKSZ3	BKSZ2	BKSZ1	BKSZ0	
DMDR_3	DTE	DACKE	TENDE	—	DREQS	NRD	—	—	
	ACT	—	—	—	—	—	ESIF	DTIF	
	DTSZ1	DTSZ0	MDS1	MDS0	TSEIE	—	ESIE	DTIE	
	DTF1	DTF0	DTA	—	—	DMAP2	DMAP1	DMAP0	
DACR_3	AMS	DIRS	—	—	—	RPTIE	ARS1	ARS0	
	—	—	SAT1	SAT0	—	—	DAT1	DAT0	
	SARIE	—	—	SARA4	SARA3	SARA2	SARA1	SARA0	
	DARIE	—	—	DARA4	DARA3	DARA2	DARA1	DARA0	
DMRSR_0								DMAC_0	
DMRSR_1								DMAC_1	
DMRSR_2								DMAC_2	
DMRSR_3								DMAC_3	
IPRA	—	IPRA14	IPRA13	IPRA12	—	IPRA10	IPRA9	IPRA8	INTC
	—	IPRA6	IPRA5	IPRA4	—	IPRA2	IPRA1	IPRA0	
IPRB	—	IPRB14	IPRB13	IPRB12	—	IPRB10	IPRB9	IPRB8	
	—	IPRB6	IPRB5	IPRB4	—	IPRB2	IPRB1	IPRB0	
IPRC	—	IPRC14	IPRC13	IPRC12	—	IPRC10	IPRC9	IPRC8	
	—	IPRC6	IPRC5	IPRC4	—	IPRC2	IPRC1	IPRC0	
IPRD	—	IPRD14	IPRD13	IPRD12	—	IPRD10	IPRD9	IPRD8	
	—	IPRD6	IPRD5	IPRD4	—	IPRD2	IPRD1	IPRD0	

レジスタ 略称	ビット 31/23/15/7	ビット 30/22/14/6	ビット 29/21/13/5	ビット 28/20/12/4	ビット 27/19/11/3	ビット 26/18/10/2	ビット 25/17/9/1	ビット 24/16/8/0	モジュール
IPRE	—	IPRE14	IPRE13	IPRE12	—	IPRE10	IPRE9	IPRE8	INTC
	—	IPRE6	IPRE5	IPRE4	—	IPRE2	IPRE1	IPRE0	
IPRF	—	IPRF14	IPRF13	IPRF12	—	IPRF10	IPRF9	IPRF8	
	—	IPRF6	IPRF5	IPRF4	—	IPRF2	IPRF1	IPRF0	
IPRG	—	IPRG14	IPRG13	IPRG12	—	IPRG10	IPRG9	IPRG8	
	—	IPRG6	IPRG5	IPRG4	—	IPRG2	IPRG1	IPRG0	
IPRI	—	IPRI14	IPRI13	IPRI12	—	IPRI10	IPRI9	IPRI8	
	—	IPRI6	IPRI5	IPRI4	—	IPRI2	IPRI1	IPRI0	
IPRJ	—	IPRJ14	IPRJ13	IPRJ12	—	IPRJ10	IPRJ9	IPRJ8	
	—	IPRJ6	IPRJ5	IPRJ4	—	IPRJ2	IPRJ1	IPRJ0	
IPRL	—	IPRL14	IPRL13	IPRL12	—	IPRL10	IPRL9	IPRL8	
	—	IPRL6	IPRL5	IPRL4	—	IPRL2	IPRL1	IPRL0	
IPRM	—	IPRM14	IPRM13	IPRM12	—	IPRM10	IPRM9	IPRM8	
	—	IPRM6	IPRM5	IPRM4	—	IPRM2	IPRM1	IPRM0	
IPRN	—	IPRN14	IPRN13	IPRN12	—	IPRN10	IPRN9	IPRN8	
	—	IPRN6	IPRN5	IPRN4	—	IPRN2	IPRN1	IPRN0	
IPRO	—	IPRO14	IPRO13	IPRO12	—	IPRO10	IPRO9	IPRO8	
	—	IPRO6	IPRO5	IPRO4	—	IPRO2	IPRO1	IPRO0	
IPRP	—	IPRP14	IPRP13	IPRP12	—	IPRP10	IPRP9	IPRP8	
	—	IPRP6	IPRP5	IPRP4	—	IPRP2	IPRP1	IPRP0	
ISCRH	—	—	IRQ14SR	IRQ14SF	IRQ13SR	IRQ13SF	IRQ12SR	IRQ12SF	
	—	IRQ11SR	IRQ11SF	IRQ10SR	IRQ10SF	IRQ9SR	IRQ9SF	IRQ8SR	
ISURL	IRQ7SR	IRQ7SF	IRQ6SR	IRQ6SF	IRQ5SR	IRQ5SF	IRQ4SR	IRQ4SF	
	IRQ3SR	IRQ3SF	IRQ2SR	IRQ2SF	IRQ1SR	IRQ1SF	IRQ0SR	IRQ0SF	
DTCVBR									BSC
BCR2	—	—	—	IBCCS	—	—	—	PWDBE	
MDCR	—	—	—	—	MDS3	MDS2	MDS1	MDS0	SYSTEM
	—	—	—	—	—	—	—	—	
SYSCR0	—	—	MACS	—	—	—	—	RAME	
	—	—	—	—	—	—	—	—	
SCKCR0	PSTOP1	—	POSEL1	—	—	ICK2	ICK1	ICK0	
	—	PCK2	PCK1	PCK0	—	BCK2	BCK1	BCK0	

レジスタ 略称	ビット 31/23/15/7	ビット 30/22/14/6	ビット 29/21/13/5	ビット 28/20/12/4	ビット 27/19/11/3	ビット 26/18/10/2	ビット 25/17/9/1	ビット 24/16/8/0	モジュール
SBYCR	SSBY	OPE	SSBYF	STS4	STS3	STS2	STS1	STS0	SYSTEM
	SLPIE	—	—	—	—	—	—	—	
MSTPCRA	ACSE	MSTPA14	MSTPA13	MSTPA12	MSTPA11	MSTPA10	MSTPA9	MSTPA8	
	MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0	
MSTPCRB	MSTPB15	MSTPB14	MSTPB13	MSTPB12	MSTPB11	MSTPB10	MSTPB9	MSTPB8	
	MSTPB7	MSTPB6	MSTPB5	MSTPB4	MSTPB3	MSTPB2	MSTPB1	MSTPB0	
MSTPCRC	MSTPC15	MSTPC14	MSTPC13	MSTPC12	MSTPC11	MSTPC10	MSTPC9	MSTPC8	
	MSTPC7	MSTPC6	MSTPC5	MSTPC4	MSTPC3	MSTPC2	MSTPC1	MSTPC0	
ROSCCR	INOSCE	OSCERR	OSCIE	—	—	—	—	ERRTEST	
	—	—	—	—	—	—	—	—	
SYSCR1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	EEPWT	
SCKCR1	—	—	—	—	—	FCK2	FCK1	FCK0	
	—	ACK2	ACK1	ACK0	—	RCK2	RCK1	RCK0	
MSTPCRD	MSTPD15	MSTPD14	MSTPD13	MSTPD12	MSTPD11	MSTPD10	MSTPD9	MSTPD8	
	MSTPD7	MSTPD6	MSTPD5	MSTPD4	MSTPD3	MSTPD2	MSTPD1	MSTPD0	
MSTPCRE	MSTPE15	MSTPE14	MSTPE13	MSTPE12	MSTPE11	MSTPE10	MSTPE9	MSTPE8	
	MSTPE7	MSTPE6	MSTPE5	MSTPE4	MSTPE3	MSTPE2	MSTPE1	MSTPE0	
SMR_3	C/ \bar{A}	CHR	PE	O/ \bar{E}	STOP	MP	CKS1	CKS0	SCI_3
BRR_3									
SCR_3	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_3									
SSR_3	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
RDR_3									
SMR_4	C/ \bar{A}	CHR	PE	O/ \bar{E}	STOP	MP	CKS1	CKS0	SCI_4
BRR_4									
SCR_4	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_4									
SSR_4	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
RDR_4									
TCR_4	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_4
TMDR_4	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_4	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_4	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_4	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	

レジスタ 略称	ビット 31/23/15/7	ビット 30/22/14/6	ビット 29/21/13/5	ビット 28/20/12/4	ビット 27/19/11/3	ビット 26/18/10/2	ビット 25/17/9/1	ビット 24/16/8/0	モジュール
TCNT4									TPU_4
TGRA_4									
TGRB_4									
TCR_5	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_5
TMDR_5	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_5	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_5	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_5	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_5									
TGRA_5									
TGRB_5									
DTCERA	DTCE15	DTCE14	DTCE13	DTCE12	DTCE11	DTCE10	DTCE9	DTCE8	DTC
	DTCE7	DTCE6	DTCE5	DTCE4	DTCE3	DTCE2	DTCE1	DTCE0	
DTCERB	DTCE15	DTCE14	DTCE13	DTCE12	DTCE11	DTCE10	DTCE9	DTCE8	
	DTCE7	DTCE6	DTCE5	DTCE4	DTCE3	DTCE2	DTCE1	DTCE0	
DTCERC	DTCE15	DTCE14	DTCE13	DTCE12	DTCE11	DTCE10	DTCE9	DTCE8	
	DTCE7	DTCE6	DTCE5	DTCE4	DTCE3	DTCE2	DTCE1	DTCE0	
DTCERD	DTCE15	DTCE14	DTCE13	DTCE12	DTCE11	DTCE10	DTCE9	DTCE8	
	DTCE7	DTCE6	DTCE5	DTCE4	DTCE3	DTCE2	DTCE1	DTCE0	
DTCERE	DTCE15	DTCE14	DTCE13	DTCE12	DTCE11	DTCE10	DTCE9	DTCE8	
	DTCE7	DTCE6	DTCE5	DTCE4	DTCE3	DTCE2	DTCE1	DTCE0	
DTCERF	DTCE15	DTCE14	DTCE13	DTCE12	DTCE11	DTCE10	DTCE9	DTCE8	
	DTCE7	DTCE6	DTCE5	DTCE4	DTCE3	DTCE2	DTCE1	DTCE0	
DTCERG	DTCE15	DTCE14	DTCE13	DTCE12	DTCE11	DTCE10	DTCE9	DTCE8	
	DTCE7	DTCE6	DTCE5	DTCE4	DTCE3	DTCE2	DTCE1	DTCE0	
DTCCR	—	—	—	—	—	—	—	—	
INTCR	—	—	INTM1	INTM0	NMIEG	—	—	—	INTC
CPUPCR	CPUPCE	DTCP2	DTCP1	DTCP0	IPSETE	CPUP2	CPUP1	CPUP0	
IER	IRQ15E	IRQ14E	IRQ13E	IRQ12E	IRQ11E	IRQ10E	IRQ9E	IRQ8E	
	IRQ7E	IRQ6E	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E	

レジスタ 略称	ビット 31/23/15/7	ビット 30/22/14/6	ビット 29/21/13/5	ビット 28/20/12/4	ビット 27/19/11/3	ビット 26/18/10/2	ビット 25/17/9/1	ビット 24/16/8/0	モジュール	
ISR	IRQ15F	IRQ14F	IRQ13F	IRQ12F	IRQ11F	IRQ10F	IRQ9F	IRQ8F	INTC	
	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F		
PORT1	P17	P16	P15	P14	P13	P12	P11	P10	I/O ポート	
PORT3	P37	P36	P35	P34	P33	P32	P31	P30		
PORT4	P47	P46	P45	P44	P43	P42	P41	P40		
PORT5	P57	P56	P55	P54	P53	P52	P51	P50		
PORT6	—	P66	P65	P64	P63	P62	P61	P60		
PORTA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	—		
PORTD	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0		
P1DR	P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR	P10DR		
P3DR	P37DR	P36DR	P35DR	P34DR	P33DR	P32DR	P31DR	P30DR		
P6DR	—	P66DR	P65DR	P64DR	P63DR	P62DR	P61DR	P60DR		
PADR	PA7DR	PA6DR	PA5DR	PA4DR	PA3DR	PA2DR	PA1DR	—		
PDDR	PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR		
PCR	G3CMS1	G3CMS0	G2CMS1	G2CMS0	—	—	—	—	PPG	
PMR	G3INV	G2INV	—	—	G3NOV	G2NOV	—	—		
NDERH	NDER15	NDER14	NDER13	NDER12	NDER11	NDER10	NDER9	NDER8		
NDERL	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0		
PODRH	POD15	POD14	POD13	POD12	POD11	POD10	POD9	POD8		
PODRL	POD7	POD6	POD5	POD4	POD3	POD2	POD1	POD0		
NDRH	NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9	NDR8		
NDRL	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1	NDR0		
TCSR	OVF	WT/IT	TME	—	—	CKS2	CKS1	CKS0		WDT
TCNT										
RSTCSR	WOVF	RSTE	—	—	—	—	—	—		TPU
TSTR	—	—	CST5	CST4	CST3	CST2	CST1	CST0		
TSYR	—	—	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0	(ユニット0)	
TCR_0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_0	
TMDR_0	—	—	BFB	BFA	MD3	MD2	MD1	MD0		
TIORH_0	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0		
TIORL_0	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0		
TIER_0	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA		
TSR_0	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA		
TCNT_0										

レジスタ 略称	ビット 31/23/15/7	ビット 30/22/14/6	ビット 29/21/13/5	ビット 28/20/12/4	ビット 27/19/11/3	ビット 26/18/10/2	ビット 25/17/9/1	ビット 24/16/8/0	モジュール
TGRA_0									TPU_0
TGRB_0									
TGRC_0									
TGRD_0									
TCR_1	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_1
TMDR_1	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_1	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_1	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_1	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_1									
TGRA_1									
TGRB_1									
TCR_2	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_2
TMDR_2	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_2	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_2	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_2	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_2									
TGRA_2									
TGRB_2									
TCR_3	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_3
TMDR_3	—	—	BFB	BFA	MD3	MD2	MD1	MD0	
TIORH_3	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIORL_3	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0	
TIER_3	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
TSR_3	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	

レジスタ 略称	ビット 31/23/15/7	ビット 30/22/14/6	ビット 29/21/13/5	ビット 28/20/12/4	ビット 27/19/11/3	ビット 26/18/10/2	ビット 25/17/9/1	ビット 24/16/8/0	モジュール
TCNT_3									TPU_3
TGRA_3									
TGRB_3									
TGRC_3									
TGRD_3									

24.3 各動作モードにおけるレジスタの状態

レジスタ 略称	リセット	スリープ モード	サブ クロック モード*	モジュール ストップ モード	全モジュール クロック ストップモード	ソフトウェア スタンバイ モード	ハードウェア スタンバイ モード*	モジュール
MCR_0	初期化	—	—	初期化	初期化	初期化	初期化	RCAN_0
GSR_0	初期化	—	—	初期化	初期化	初期化	初期化	
BCR1_0	初期化	—	—	初期化	初期化	初期化	初期化	
BCR0_0	初期化	—	—	初期化	初期化	初期化	初期化	
IRR_0	初期化	—	—	初期化	初期化	初期化	初期化	
IMR_0	初期化	—	—	初期化	初期化	初期化	初期化	
TEC_0	初期化	—	—	初期化	初期化	初期化	初期化	
REC_0	初期化	—	—	初期化	初期化	初期化	初期化	
TXPR1_0	初期化	—	—	初期化	初期化	初期化	初期化	
TXPR0_0	初期化	—	—	初期化	初期化	初期化	初期化	
TXCR1_0	初期化	—	—	初期化	初期化	初期化	初期化	
TXCR0_0	初期化	—	—	初期化	初期化	初期化	初期化	
TXACK1_0	初期化	—	—	初期化	初期化	初期化	初期化	
TXACK0_0	初期化	—	—	初期化	初期化	初期化	初期化	
ABACK1_0	初期化	—	—	初期化	初期化	初期化	初期化	
ABACK0_0	初期化	—	—	初期化	初期化	初期化	初期化	
RXPR1_0	初期化	—	—	初期化	初期化	初期化	初期化	
RXPR0_0	初期化	—	—	初期化	初期化	初期化	初期化	
RFPR1_0	初期化	—	—	初期化	初期化	初期化	初期化	
RFPR0_0	初期化	—	—	初期化	初期化	初期化	初期化	
MBIMR1_0	初期化	—	—	初期化	初期化	初期化	初期化	
MBIMR0_0	初期化	—	—	初期化	初期化	初期化	初期化	
UMSR1_0	初期化	—	—	初期化	初期化	初期化	初期化	
UMSR0_0	初期化	—	—	初期化	初期化	初期化	初期化	
TTCR0_0	初期化	—	—	初期化	初期化	初期化	初期化	
CMAX_TEW_0	初期化	—	—	初期化	初期化	初期化	初期化	
RFTROFF_0	初期化	—	—	初期化	初期化	初期化	初期化	
TSR_0	初期化	—	—	初期化	初期化	初期化	初期化	
CCR_0	初期化	—	—	初期化	初期化	初期化	初期化	
TCNTR_0	初期化	—	—	初期化	初期化	初期化	初期化	
CYCTR_0	初期化	—	—	初期化	初期化	初期化	初期化	
RFMK_0	初期化	—	—	初期化	初期化	初期化	初期化	
TCMR0_0	初期化	—	—	初期化	初期化	初期化	初期化	

レジスタ 略称	リセット	スリープ モード	サブ クロック モード*	モジュール ストップ モード	全モジュール クロック ストップモード	ソフトウェア スタンバイ モード	ハードウェア スタンバイ モード*	モジュール
TCMR1_0	初期化	—	—	初期化	初期化	初期化	初期化	RCAN_0
TCMR2_0	初期化	—	—	初期化	初期化	初期化	初期化	
TTTSEL_0	初期化	—	—	初期化	初期化	初期化	初期化	
MB_0[0]. CONTROL0H	—	—	—	—	—	—	—	
MB_0[0]. CONTROL0L	—	—	—	—	—	—	—	
MB_0[0]. LAFMH	—	—	—	—	—	—	—	
MB_0[0]. LAFML	—	—	—	—	—	—	—	
MB_0[0]. MSG_DATA[0]	—	—	—	—	—	—	—	
MB_0[0]. MSG_DATA[1]	—	—	—	—	—	—	—	
MB_0[0]. MSG_DATA[2]	—	—	—	—	—	—	—	
MB_0[0]. MSG_DATA[3]	—	—	—	—	—	—	—	
MB_0[0]. MSG_DATA[4]	—	—	—	—	—	—	—	
MB_0[0]. MSG_DATA[5]	—	—	—	—	—	—	—	
MB_0[0]. MSG_DATA[6]	—	—	—	—	—	—	—	
MB_0[0]. MSG_DATA[7]	—	—	—	—	—	—	—	
MB_0[0]. CONTROL1H	初期化	—	—	初期化	初期化	初期化	初期化	
MB_0[0]. CONTROL1L	初期化	—	—	初期化	初期化	初期化	初期化	
MB_0[0]. TIMESTAMP	初期化	—	—	初期化	初期化	初期化	初期化	
MB_0[1]. CONTROL0H	—	—	—	—	—	—	—	

レジスタ 略称	リセット	スリープ モード	サブ クロック モード*	モジュール ストップ モード	全モジュール クロック ストップモード	ソフトウェア スタンバイ モード	ハードウェア スタンバイ モード*	モジュール
MB_0[1]. CONTROL0L	—	—	—	—	—	—	—	RCAN_0
MB_0[1]. LAFMH	—	—	—	—	—	—	—	
MB_0[1]. LAFML	—	—	—	—	—	—	—	
MB_0[1]. MSG_DATA[0]	—	—	—	—	—	—	—	
MB_0[1]. MSG_DATA[1]	—	—	—	—	—	—	—	
MB_0[1]. MSG_DATA[2]	—	—	—	—	—	—	—	
MB_0[1]. MSG_DATA[3]	—	—	—	—	—	—	—	
MB_0[1]. MSG_DATA[4]	—	—	—	—	—	—	—	
MB_0[1]. MSG_DATA[5]	—	—	—	—	—	—	—	
MB_0[1]. MSG_DATA[6]	—	—	—	—	—	—	—	
MB_0[1]. MSG_DATA[7]	—	—	—	—	—	—	—	
MB_0[1]. CONTROL1H	初期化	—	—	初期化	初期化	初期化	初期化	
MB_0[1]. CONTROL1L	初期化	—	—	初期化	初期化	初期化	初期化	
MB_0[1]. TIMESTAMP	初期化	—	—	初期化	初期化	初期化	初期化	

レジスタ 略称	リセット	スリープ モード	サブ クロック モード*	モジュール ストップ モード	全モジュール クロック ストップモード	ソフトウェア スタンバイ モード	ハードウェア スタンバイ モード*	モジュール
MB_0[2]. CONTROL0H ~ MB_0[2]. TIMESTAMP								RCAN_0
↓ (繰り返し)								
MB_0[15]. CONTROL0H ~ MB_0[15]. TIMESTAMP								
MB_0[16]. CONTROL0H	-	-	-	-	-	-	-	
MB_0[16]. CONTROL0L	-	-	-	-	-	-	-	
MB_0[16]. LAFMH	-	-	-	-	-	-	-	
MB_0[16]. LAFML	-	-	-	-	-	-	-	
MB_0[16]. MSG_DATA[0]	-	-	-	-	-	-	-	
MB_0[16]. MSG_DATA[1]	-	-	-	-	-	-	-	
MB_0[16]. MSG_DATA[2]	-	-	-	-	-	-	-	
MB_0[16]. MSG_DATA[3]	-	-	-	-	-	-	-	
MB_0[16]. MSG_DATA[4]	-	-	-	-	-	-	-	
MB_0[16]. MSG_DATA[5]	-	-	-	-	-	-	-	
MB_0[16]. MSG_DATA[6]	-	-	-	-	-	-	-	
MB_0[16]. MSG_DATA[7]	-	-	-	-	-	-	-	

レジスタ 略称	リセット	スリープ モード	サブ クロック モード*	モジュール ストップ モード	全モジュール クロック ストップモード	ソフトウェア スタンバイ モード	ハードウェア スタンバイ モード*	モジュール
MB_0[16]. CONTROL1H	初期化	—	—	初期化	初期化	初期化	初期化	RCAN_0
MB_0[16]. CONTROL1L	初期化	—	—	初期化	初期化	初期化	初期化	
MB_0[17]. CONTROL0H ~ MB_0[17]. CONTROL1L	MB_0[16].CONTROL0H~MB_0[16].CONTROL1L と同じ							
↓	(繰り返し)							
MB_0[23]. CONTROL0H ~ MB_0[23]. CONTROL1L	MB_0[16].CONTROL0H~MB_0[16].CONTROL1L と同じ							
MB_0[24]. CONTROL0H	—	—	—	—	—	—	—	
MB_0[24]. CONTROL0L	—	—	—	—	—	—	—	
MB_0[24]. LAFMH	—	—	—	—	—	—	—	
MB_0[24]. LAFML	—	—	—	—	—	—	—	
MB_0[24]. MSG_DATA[0]	—	—	—	—	—	—	—	
MB_0[24]. MSG_DATA[1]	—	—	—	—	—	—	—	
MB_0[24]. MSG_DATA[2]	—	—	—	—	—	—	—	
MB_0[24]. MSG_DATA[3]	—	—	—	—	—	—	—	
MB_0[24]. MSG_DATA[4]	—	—	—	—	—	—	—	
MB_0[24]. MSG_DATA[5]	—	—	—	—	—	—	—	

レジスタ 略称	リセット	スリープ モード	サブ クロック モード*	モジュール ストップ モード	全モジュール クロック ストップモード	ソフトウェア スタンバイ モード	ハードウェア スタンバイ モード*	モジュール
MB_0[24]. MSG_DATA[6]	—	—	—	—	—	—	—	RCAN_0
MB_0[24]. MSG_DATA[7]	—	—	—	—	—	—	—	
MB_0[24]. CONTROL1H	初期化	—	—	初期化	初期化	初期化	初期化	
MB_0[24]. CONTROL1L	初期化	—	—	初期化	初期化	初期化	初期化	
MB_0[24]. TTT	初期化	—	—	初期化	初期化	初期化	初期化	
MB_0[24]. TT_CONTROL	初期化	—	—	初期化	初期化	初期化	初期化	
MB_0[25]. CONTROL0H ~ MB_0[25]. TT_CONTROL	MB_0[24].CONTROL0H~MB_0[24]. TT_CONTROL と同じ							
↓	(繰り返し)							
MB_0[29]. CONTROL0H ~ MB_0[29]. TT_CONTROL	MB_0[24].CONTROL0H~MB_0[24]. TT_CONTROL と同じ							
MB_0[30]. CONTROL0H	—	—	—	—	—	—	—	
MB_0[30]. CONTROL0L	—	—	—	—	—	—	—	
MB_0[30]. LAFMH	—	—	—	—	—	—	—	
MB_0[30]. LAFML	—	—	—	—	—	—	—	
MB_0[30]. MSG_DATA[0]	—	—	—	—	—	—	—	
MB_0[30]. MSG_DATA[1]	—	—	—	—	—	—	—	

レジスタ 略称	リセット	スリープ モード	サブ クロック モード*	モジュール ストップ モード	全モジュール クロック ストップモード	ソフトウェア スタンバイ モード	ハードウェア スタンバイ モード*	モジュール
MB_0[30]. MSG_DATA[2]	—	—	—	—	—	—	—	RCAN_0
MB_0[30]. MSG_DATA[3]	—	—	—	—	—	—	—	
MB_0[30]. MSG_DATA[4]	—	—	—	—	—	—	—	
MB_0[30]. MSG_DATA[5]	—	—	—	—	—	—	—	
MB_0[30]. MSG_DATA[6]	—	—	—	—	—	—	—	
MB_0[30]. MSG_DATA[7]	—	—	—	—	—	—	—	
MB_0[30]. CONTROL1H	初期化	—	—	初期化	初期化	初期化	初期化	
MB_0[30]. CONTROL1L	初期化	—	—	初期化	初期化	初期化	初期化	
MB_0[30]. TIMESTAMP	初期化	—	—	初期化	初期化	初期化	初期化	
MB_0[30]. TTT	初期化	—	—	初期化	初期化	初期化	初期化	
MB_0[31]. CONTROL0H	—	—	—	—	—	—	—	
MB_0[31]. CONTROL0L	—	—	—	—	—	—	—	
MB_0[31]. LAFMH	—	—	—	—	—	—	—	
MB_0[31]. LAFML	—	—	—	—	—	—	—	
MB_0[31]. MSG_DATA[0]	—	—	—	—	—	—	—	
MB_0[31]. MSG_DATA[1]	—	—	—	—	—	—	—	
MB_0[31]. MSG_DATA[2]	—	—	—	—	—	—	—	

レジスタ 略称	リセット	スリープ モード	サブ クロック モード*	モジュール ストップ モード	全モジュール クロック ストップモード	ソフトウェア スタンバイ モード	ハードウェア スタンバイ モード*	モジュール
MB_0[31]. MSG_DATA[3]	—	—	—	—	—	—	—	RCAN_0
MB_0[31]. MSG_DATA[4]	—	—	—	—	—	—	—	
MB_0[31]. MSG_DATA[5]	—	—	—	—	—	—	—	
MB_0[31]. MSG_DATA[6]	—	—	—	—	—	—	—	
MB_0[31]. MSG_DATA[7]	—	—	—	—	—	—	—	
MB_0[31]. CONTROL1H	初期化	—	—	初期化	初期化	初期化	初期化	
MB_0[31]. CONTROL1L	初期化	—	—	初期化	初期化	初期化	初期化	
MB_0[31]. TIMESTAMP	初期化	—	—	初期化	初期化	初期化	初期化	
RCANMON_0	初期化	—	—	—	—	—	初期化	
RCANMON_1	初期化	—	—	—	—	—	初期化	RCAN_1
MCR_1				RCAN_1 は RCAN_0 と同じ				
↓								
MB_1[31]. CONTROL0H								
~								
MB_1[31]. TIMESTAMP								
SPCRA	初期化	—	—	—	—	—	初期化	RSPI
SSLPA	初期化	—	—	—	—	—	初期化	
SPPCRA	初期化	—	—	—	—	—	初期化	
SPSRA	初期化	—	—	—	—	—	初期化	
SPDRA	初期化	—	—	—	—	—	初期化	
SPSCRA	初期化	—	—	—	—	—	初期化	
SPSSRA	初期化	—	—	—	—	—	初期化	
SPBRA	初期化	—	—	—	—	—	初期化	
SPDCRA	初期化	—	—	—	—	—	初期化	
SPCKDA	初期化	—	—	—	—	—	初期化	

レジスタ 略称	リセット	スリープ モード	サブ クロック モード*	モジュール ストップ モード	全モジュール クロック ストップモード	ソフトウェア スタンバイ モード	ハードウェア スタンバイ モード*	モジュール
SSLNDA	初期化	—	—	—	—	—	初期化	RSPI
SPNDA	初期化	—	—	—	—	—	初期化	
SPCR2A	初期化	—	—	—	—	—	初期化	
SPCMDA0	初期化	—	—	—	—	—	初期化	
SPCMDA1	初期化	—	—	—	—	—	初期化	
SPCMDA2	初期化	—	—	—	—	—	初期化	
SPCMDA3	初期化	—	—	—	—	—	初期化	
SPCMDA4	初期化	—	—	—	—	—	初期化	
SPCMDA5	初期化	—	—	—	—	—	初期化	
SPCMDA6	初期化	—	—	—	—	—	初期化	
SPCMDA7	初期化	—	—	—	—	—	初期化	
SPCRB	初期化	—	—	—	—	—	初期化	
SSLPB	初期化	—	—	—	—	—	初期化	
SPPCRB	初期化	—	—	—	—	—	初期化	
SPSRB	初期化	—	—	—	—	—	初期化	
SPDRB	初期化	—	—	—	—	—	初期化	
SPSCRB	初期化	—	—	—	—	—	初期化	
SPSSRB	初期化	—	—	—	—	—	初期化	
SPBRB	初期化	—	—	—	—	—	初期化	
SPDCRB	初期化	—	—	—	—	—	初期化	
SPCKDB	初期化	—	—	—	—	—	初期化	
SSLNDB	初期化	—	—	—	—	—	初期化	
SPNDB	初期化	—	—	—	—	—	初期化	
SPCR2B	初期化	—	—	—	—	—	初期化	
SPCMDB0	初期化	—	—	—	—	—	初期化	
SPCMDB1	初期化	—	—	—	—	—	初期化	
SPCMDB2	初期化	—	—	—	—	—	初期化	
SPCMDB3	初期化	—	—	—	—	—	初期化	
SPCMDB4	初期化	—	—	—	—	—	初期化	
SPCMDB5	初期化	—	—	—	—	—	初期化	
SPCMDB6	初期化	—	—	—	—	—	初期化	
SPCMDB7	初期化	—	—	—	—	—	初期化	
SPCRC	初期化	—	—	—	—	—	初期化	
SSLPC	初期化	—	—	—	—	—	初期化	
SPPCRC	初期化	—	—	—	—	—	初期化	

レジスタ 略称	リセット	スリープ モード	サブ クロック モード*	モジュール ストップ モード	全モジュール クロック ストップモード	ソフトウェア スタンバイ モード	ハードウェア スタンバイ モード*	モジュール
SPSRC	初期化	—	—	—	—	—	初期化	RSPI
SPDRC	初期化	—	—	—	—	—	初期化	
SPSCRC	初期化	—	—	—	—	—	初期化	
SPSSRC	初期化	—	—	—	—	—	初期化	
SPBRC	初期化	—	—	—	—	—	初期化	
SPDCRC	初期化	—	—	—	—	—	初期化	
SPCKDC	初期化	—	—	—	—	—	初期化	
SSLNDC	初期化	—	—	—	—	—	初期化	
SPNDC	初期化	—	—	—	—	—	初期化	
SPCR2C	初期化	—	—	—	—	—	初期化	
SPCMDC0	初期化	—	—	—	—	—	初期化	
SPCMDC1	初期化	—	—	—	—	—	初期化	
SPCMDC2	初期化	—	—	—	—	—	初期化	
SPCMDC3	初期化	—	—	—	—	—	初期化	
SPCMDC4	初期化	—	—	—	—	—	初期化	
SPCMDC5	初期化	—	—	—	—	—	初期化	
SPCMDC6	初期化	—	—	—	—	—	初期化	
SPCMDC7	初期化	—	—	—	—	—	初期化	
SPCRD	初期化	—	—	—	—	—	初期化	
SSLPD	初期化	—	—	—	—	—	初期化	
SPPCRD	初期化	—	—	—	—	—	初期化	
SPSRD	初期化	—	—	—	—	—	初期化	
SPDRD	初期化	—	—	—	—	—	初期化	
SPSCRD	初期化	—	—	—	—	—	初期化	
SPSSRD	初期化	—	—	—	—	—	初期化	
SPBRD	初期化	—	—	—	—	—	初期化	
SPDCRD	初期化	—	—	—	—	—	初期化	
SPCKDD	初期化	—	—	—	—	—	初期化	
SSLNDD	初期化	—	—	—	—	—	初期化	
SPNDD	初期化	—	—	—	—	—	初期化	
SPCR2D	初期化	—	—	—	—	—	初期化	
SPCMDD0	初期化	—	—	—	—	—	初期化	
SPCMDD1	初期化	—	—	—	—	—	初期化	
SPCMDD2	初期化	—	—	—	—	—	初期化	
SPCMDD3	初期化	—	—	—	—	—	初期化	

レジスタ 略称	リセット	スリープ モード	サブ クロック モード*	モジュール ストップ モード	全モジュール クロック ストップモード	ソフトウェア スタンバイ モード	ハードウェア スタンバイ モード*	モジュール
SPCMDD4	初期化	—	—	—	—	—	初期化	RSPI
SPCMDD5	初期化	—	—	—	—	—	初期化	
SPCMDD6	初期化	—	—	—	—	—	初期化	
SPCMDD7	初期化	—	—	—	—	—	初期化	
LINCR	初期化	—	—	—	—	—	初期化	HWLIN
LINSTR	初期化	—	—	—	—	—	初期化	
LINTCR	初期化	—	—	—	—	—	初期化	
LINTCNT	初期化	—	—	—	—	—	初期化	
LINTCNT	初期化	—	—	—	—	—	初期化	
CRCCR	初期化	—	—	—	—	—	初期化	CRC
CRCDIR	初期化	—	—	—	—	—	初期化	
CRCDOR	初期化	—	—	—	—	—	初期化	
FMODR	初期化	—	—	—	—	—	初期化	FLASH/ EEROM
FASTAT	初期化	—	—	—	—	—	初期化	
FAEINT	初期化	—	—	—	—	—	初期化	
ROMMAT	初期化	—	—	—	—	—	初期化	
EEPWE0	初期化	—	—	—	—	—	初期化	
EEPWE1	初期化	—	—	—	—	—	初期化	
EEPWE0	初期化	—	—	—	—	—	初期化	
EEPWE1	初期化	—	—	—	—	—	初期化	
FCURAME	初期化	—	—	—	—	—	初期化	
FSTATR0	初期化	—	—	—	—	—	初期化	
FSTATR1	初期化	—	—	—	—	—	初期化	
FENTRYR	初期化	—	—	—	—	—	初期化	
FPROTR	初期化	—	—	—	—	—	初期化	
FRESETR	初期化	—	—	—	—	—	初期化	
FCMDR	初期化	—	—	—	—	—	初期化	
FRAMECCR	初期化	—	—	—	—	—	初期化	
FCPSR	初期化	—	—	—	—	—	初期化	
EEPBCCNT	初期化	—	—	—	—	—	初期化	
FPESTAT	初期化	—	—	—	—	—	初期化	
EEPBCSTAT	初期化	—	—	—	—	—	初期化	
FCKAR	初期化	—	—	—	—	—	初期化	
EEPSTAT	初期化	—	—	—	—	—	初期化	

レジスタ 略称	リセット	スリープ モード	サブ クロック モード*	モジュール ストップ モード	全モジュール クロック ストップモード	ソフトウェア スタンバイ モード	ハードウェア スタンバイ モード*	モジュール
RAMEN	初期化	—	—	—	—	—	初期化	RAM
RAMWEN	初期化	—	—	—	—	—	初期化	
RAMECC	初期化	—	—	—	—	—	初期化	
RAMERR	初期化	—	—	—	—	—	初期化	
RAMINT	初期化	—	—	—	—	—	初期化	
RAMACYC	初期化	—	—	—	—	—	初期化	
ADDRA_0	初期化	—	—	—	—	—	初期化	A/D_0
ADDRB_0	初期化	—	—	—	—	—	初期化	
ADDRC_0	初期化	—	—	—	—	—	初期化	
ADDRD_0	初期化	—	—	—	—	—	初期化	
ADDRE_0	初期化	—	—	—	—	—	初期化	
ADDRF_0	初期化	—	—	—	—	—	初期化	
ADDRG_0	初期化	—	—	—	—	—	初期化	
ADDRH_0	初期化	—	—	—	—	—	初期化	
ADCSR_0	初期化	—	—	—	—	—	初期化	
ADCR_0	初期化	—	—	—	—	—	初期化	
APPDCR_0	初期化	—	—	—	—	—	初期化	
ADDIAGR_0	初期化	—	—	—	—	—	初期化	
ADDRA_1	初期化	—	—	—	—	—	初期化	A/D_1
ADDRB_1	初期化	—	—	—	—	—	初期化	
ADDRC_1	初期化	—	—	—	—	—	初期化	
ADDRD_1	初期化	—	—	—	—	—	初期化	
ADDRE_1	初期化	—	—	—	—	—	初期化	
ADDRF_1	初期化	—	—	—	—	—	初期化	
ADDRG_1	初期化	—	—	—	—	—	初期化	
ADDRH_1	初期化	—	—	—	—	—	初期化	
ADCSR_1	初期化	—	—	—	—	—	初期化	
ADCR_1	初期化	—	—	—	—	—	初期化	
APPDCR_1	初期化	—	—	—	—	—	初期化	
ADDIAGR_1	初期化	—	—	—	—	—	初期化	
TSTRB	初期化	—	—	—	—	—	初期化	TPU
TSYRB	初期化	—	—	—	—	—	初期化	(ユニット1)
TCR_6	初期化	—	—	—	—	—	初期化	TPU_6
TMDR_6	初期化	—	—	—	—	—	初期化	
TIORH_6	初期化	—	—	—	—	—	初期化	

レジスタ 略称	リセット	スリープ モード	サブ クロック モード*	モジュール ストップ モード	全モジュール クロック ストップモード	ソフトウェア スタンバイ モード	ハードウェア スタンバイ モード*	モジュール
TIORL_6	初期化	—	—	—	—	—	初期化	TPU_6
TIER_6	初期化	—	—	—	—	—	初期化	
TSR_6	初期化	—	—	—	—	—	初期化	
TCNT_6	初期化	—	—	—	—	—	初期化	
TGRA_6	初期化	—	—	—	—	—	初期化	
TGRB_6	初期化	—	—	—	—	—	初期化	
TGRC_6	初期化	—	—	—	—	—	初期化	
TGRD_6	初期化	—	—	—	—	—	初期化	
TCR_7	初期化	—	—	—	—	—	初期化	TPU_7
TMDR_7	初期化	—	—	—	—	—	初期化	
TIOR_7	初期化	—	—	—	—	—	初期化	
TIER_7	初期化	—	—	—	—	—	初期化	
TSR_7	初期化	—	—	—	—	—	初期化	
TCNT_7	初期化	—	—	—	—	—	初期化	
TGRA_7	初期化	—	—	—	—	—	初期化	
TGRB_7	初期化	—	—	—	—	—	初期化	
TCR_8	初期化	—	—	—	—	—	初期化	TPU_8
TMDR_8	初期化	—	—	—	—	—	初期化	
TIOR_8	初期化	—	—	—	—	—	初期化	
TIER_8	初期化	—	—	—	—	—	初期化	
TSR_8	初期化	—	—	—	—	—	初期化	
TCNT_8	初期化	—	—	—	—	—	初期化	
TGRA_8	初期化	—	—	—	—	—	初期化	
TGRB_8	初期化	—	—	—	—	—	初期化	
TCR_9	初期化	—	—	—	—	—	初期化	TPU_9
TMDR_9	初期化	—	—	—	—	—	初期化	
TIORH_9	初期化	—	—	—	—	—	初期化	
TIORL_9	初期化	—	—	—	—	—	初期化	
TIER_9	初期化	—	—	—	—	—	初期化	
TSR_9	初期化	—	—	—	—	—	初期化	
TCNT_9	初期化	—	—	—	—	—	初期化	
TGRA_9	初期化	—	—	—	—	—	初期化	
TGRB_9	初期化	—	—	—	—	—	初期化	
TGRC_9	初期化	—	—	—	—	—	初期化	
TGRD_9	初期化	—	—	—	—	—	初期化	

レジスタ 略称	リセット	スリープ モード	サブ クロック モード*	モジュール ストップ モード	全モジュール クロック ストップモード	ソフトウェア スタンバイ モード	ハードウェア スタンバイ モード*	モジュール
TCR_10	初期化	—	—	—	—	—	初期化	TPU_10
TMDR_10	初期化	—	—	—	—	—	初期化	
TIOR_10	初期化	—	—	—	—	—	初期化	
TIER_10	初期化	—	—	—	—	—	初期化	
TSR_10	初期化	—	—	—	—	—	初期化	
TCNT_10	初期化	—	—	—	—	—	初期化	
TGRA_10	初期化	—	—	—	—	—	初期化	
TGRB_10	初期化	—	—	—	—	—	初期化	
TCR_11	初期化	—	—	—	—	—	初期化	TPU_11
TMDR_11	初期化	—	—	—	—	—	初期化	
TIOR_11	初期化	—	—	—	—	—	初期化	
TIER_11	初期化	—	—	—	—	—	初期化	
TSR_11	初期化	—	—	—	—	—	初期化	
TCNT_11	初期化	—	—	—	—	—	初期化	
TGRA_11	初期化	—	—	—	—	—	初期化	
TGRB_11	初期化	—	—	—	—	—	初期化	
P1DDR	初期化	—	—	—	—	—	初期化	I/O ポート
P3DDR	初期化	—	—	—	—	—	初期化	
P6DDR	初期化	—	—	—	—	—	初期化	
PADDR	初期化	—	—	—	—	—	初期化	
PDDDR	初期化	—	—	—	—	—	初期化	
P1ICR	初期化	—	—	—	—	—	初期化	
P3ICR	初期化	—	—	—	—	—	初期化	
P4ICR	初期化	—	—	—	—	—	初期化	
P5ICR	初期化	—	—	—	—	—	初期化	
P6ICR	初期化	—	—	—	—	—	初期化	
PAICR	初期化	—	—	—	—	—	初期化	
PDICR	初期化	—	—	—	—	—	初期化	
PORTH	—	—	—	—	—	—	—	
PORTJ	—	—	—	—	—	—	—	
PORTK	—	—	—	—	—	—	—	
PHDR	初期化	—	—	—	—	—	初期化	
PJDR	初期化	—	—	—	—	—	初期化	
PKDR	初期化	—	—	—	—	—	初期化	
PHDDR	初期化	—	—	—	—	—	初期化	

レジスタ 略称	リセット	スリープ モード	サブ クロック モード*	モジュール ストップ モード	全モジュール クロック ストップモード	ソフトウェア スタンバイ モード	ハードウェア スタンバイ モード*	モジュール
PJDDR	初期化	—	—	—	—	—	初期化	I/O ポート
PKDDR	初期化	—	—	—	—	—	初期化	
PHICR	初期化	—	—	—	—	—	初期化	
PJICR	初期化	—	—	—	—	—	初期化	
PKICR	初期化	—	—	—	—	—	初期化	
PDPCR	初期化	—	—	—	—	—	初期化	
PHPCR	初期化	—	—	—	—	—	初期化	
PJPCR	初期化	—	—	—	—	—	初期化	
PKPCR	初期化	—	—	—	—	—	初期化	
PFCR5	初期化	—	—	—	—	—	初期化	
PFCR6	初期化	—	—	—	—	—	初期化	
PFCR8	初期化	—	—	—	—	—	初期化	
PFCR9	初期化	—	—	—	—	—	初期化	
PFCRA	初期化	—	—	—	—	—	初期化	
PFCRB	初期化	—	—	—	—	—	初期化	
PFCRC	初期化	—	—	—	—	—	初期化	
PFCRD	初期化	—	—	—	—	—	初期化	
SSIER	初期化	—	—	—	—	—	初期化	INTC
P1DSR	初期化	—	—	—	—	—	初期化	I/O ポート
P6DSR	初期化	—	—	—	—	—	初期化	
PADSR	初期化	—	—	—	—	—	初期化	
PDDSR	初期化	—	—	—	—	—	初期化	
PHDSR	初期化	—	—	—	—	—	初期化	
P1PSR	初期化	—	—	—	—	—	初期化	
P6PSR	初期化	—	—	—	—	—	初期化	
PAPSR	初期化	—	—	—	—	—	初期化	
PDPSR	初期化	—	—	—	—	—	初期化	
PHPSR	初期化	—	—	—	—	—	初期化	
DSAR_0	初期化	—	—	—	—	—	初期化	DMAC_0
DDAR_0	初期化	—	—	—	—	—	初期化	
DOFR_0	初期化	—	—	—	—	—	初期化	
DTCR_0	初期化	—	—	—	—	—	初期化	
DBSR_0	初期化	—	—	—	—	—	初期化	
DMDR_0	初期化	—	—	—	—	—	初期化	
DACR_0	初期化	—	—	—	—	—	初期化	

レジスタ 略称	リセット	スリープ モード	サブ クロック モード*	モジュール ストップ モード	全モジュール クロック ストップモード	ソフトウェア スタンバイ モード	ハードウェア スタンバイ モード*	モジュール
DSAR_1	初期化	—	—	—	—	—	初期化	DMAC_1
DDAR_1	初期化	—	—	—	—	—	初期化	
DOFR_1	初期化	—	—	—	—	—	初期化	
DTCR_1	初期化	—	—	—	—	—	初期化	
DBSR_1	初期化	—	—	—	—	—	初期化	
DMDR_1	初期化	—	—	—	—	—	初期化	
DACR_1	初期化	—	—	—	—	—	初期化	
DSAR_2	初期化	—	—	—	—	—	初期化	DMAC_2
DDAR_2	初期化	—	—	—	—	—	初期化	
DOFR_2	初期化	—	—	—	—	—	初期化	
DTCR_2	初期化	—	—	—	—	—	初期化	
DBSR_2	初期化	—	—	—	—	—	初期化	
DMDR_2	初期化	—	—	—	—	—	初期化	
DACR_2	初期化	—	—	—	—	—	初期化	
DSAR_3	初期化	—	—	—	—	—	初期化	DMAC_3
DDAR_3	初期化	—	—	—	—	—	初期化	
DOFR_3	初期化	—	—	—	—	—	初期化	
DTCR_3	初期化	—	—	—	—	—	初期化	
DBSR_3	初期化	—	—	—	—	—	初期化	
DMDR_3	初期化	—	—	—	—	—	初期化	
DACR_3	初期化	—	—	—	—	—	初期化	
DMRSR_0	初期化	—	—	—	—	—	初期化	DMAC_0
DMRSR_1	初期化	—	—	—	—	—	初期化	DMAC_1
DMRSR_2	初期化	—	—	—	—	—	初期化	DMAC_2
DMRSR_3	初期化	—	—	—	—	—	初期化	DMAC_3
IPRA	初期化	—	—	—	—	—	初期化	INTC
IPRB	初期化	—	—	—	—	—	初期化	
IPRC	初期化	—	—	—	—	—	初期化	
IPRD	初期化	—	—	—	—	—	初期化	
IPRE	初期化	—	—	—	—	—	初期化	
IPRF	初期化	—	—	—	—	—	初期化	
IPRG	初期化	—	—	—	—	—	初期化	
IPRI	初期化	—	—	—	—	—	初期化	
IPRJ	初期化	—	—	—	—	—	初期化	
IPRL	初期化	—	—	—	—	—	初期化	

レジスタ 略称	リセット	スリープ モード	サブ クロック モード*	モジュール ストップ モード	全モジュール クロック ストップモード	ソフトウェア スタンバイ モード	ハードウェア スタンバイ モード*	モジュール
IPRM	初期化	—	—	—	—	—	初期化	INTC
IPRN	初期化	—	—	—	—	—	初期化	
IPRO	初期化	—	—	—	—	—	初期化	
IPRP	初期化	—	—	—	—	—	初期化	
ISCRH	初期化	—	—	—	—	—	初期化	
ISCR_L	初期化	—	—	—	—	—	初期化	
DTCVBR	初期化	—	—	—	—	—	初期化	BSC
BCR2	初期化	—	—	—	—	—	初期化	
MDCR	初期化	—	—	—	—	—	初期化	SYSTEM
SYSCR0	初期化	—	—	—	—	—	初期化	
SCKCR0	初期化	—	—	—	—	—	初期化	
SBYCR	初期化	—	—	—	—	—	初期化	
MSTPCRA	初期化	—	—	—	—	—	初期化	
MSTPCRB	初期化	—	—	—	—	—	初期化	
MSTPCRC	初期化	—	—	—	—	—	初期化	
ROSCCR	初期化	—	—	—	—	—	初期化	
SYSCR1	初期化	—	—	—	—	—	初期化	
SCKCR1	初期化	—	—	—	—	—	初期化	
MSTPCRD	初期化	—	—	—	—	—	初期化	
MSTPCRE	初期化	—	—	—	—	—	初期化	
SMR_3	初期化	—	—	—	—	—	初期化	SCI_3
BRR_3	初期化	—	—	—	—	—	初期化	
SCR_3	初期化	—	—	—	—	—	初期化	
TDR_3	初期化	—	—	—	—	—	初期化	
SSR_3	初期化	—	—	—	—	—	初期化	
RDR_3	初期化	—	—	—	—	—	初期化	
SMR_4	初期化	—	—	—	—	—	初期化	SCI_4
BRR_4	初期化	—	—	—	—	—	初期化	
SCR_4	初期化	—	—	—	—	—	初期化	
TDR_4	初期化	—	—	—	—	—	初期化	
SSR_4	初期化	—	—	—	—	—	初期化	
RDR_4	初期化	—	—	—	—	—	初期化	
TCR_4	初期化	—	—	—	—	—	初期化	TPU_4
TMDR_4	初期化	—	—	—	—	—	初期化	
TIOR_4	初期化	—	—	—	—	—	初期化	

レジスタ 略称	リセット	スリープ モード	サブ クロック モード*	モジュール ストップ モード	全モジュール クロック ストップモード	ソフトウェア スタンバイ モード	ハードウェア スタンバイ モード*	モジュール
TIER_4	初期化	—	—	—	—	—	初期化	TPU_4
TSR_4	初期化	—	—	—	—	—	初期化	
TCNT_4	初期化	—	—	—	—	—	初期化	
TGRA_4	初期化	—	—	—	—	—	初期化	
TGRB_4	初期化	—	—	—	—	—	初期化	
TCR_5	初期化	—	—	—	—	—	初期化	TPU_5
TMDR_5	初期化	—	—	—	—	—	初期化	
TIOR_5	初期化	—	—	—	—	—	初期化	
TIER_5	初期化	—	—	—	—	—	初期化	
TSR_5	初期化	—	—	—	—	—	初期化	
TCNT_5	初期化	—	—	—	—	—	初期化	
TGRA_5	初期化	—	—	—	—	—	初期化	
TGRB_5	初期化	—	—	—	—	—	初期化	
DTCERA	初期化	—	—	—	—	—	初期化	DTC
DTCERB	初期化	—	—	—	—	—	初期化	
DTCERC	初期化	—	—	—	—	—	初期化	
DTCERD	初期化	—	—	—	—	—	初期化	
DTCERE	初期化	—	—	—	—	—	初期化	
DTCERF	初期化	—	—	—	—	—	初期化	
DTCERG	初期化	—	—	—	—	—	初期化	
DTCCR	初期化	—	—	—	—	—	初期化	
INTCR	初期化	—	—	—	—	—	初期化	INTC
CPUPCR	初期化	—	—	—	—	—	初期化	
IER	初期化	—	—	—	—	—	初期化	
ISR	初期化	—	—	—	—	—	初期化	
PORT1	—	—	—	—	—	—	—	I/O ポート
PORT3	—	—	—	—	—	—	—	
PORT4	—	—	—	—	—	—	—	
PORT5	—	—	—	—	—	—	—	
PORT6	—	—	—	—	—	—	—	
PORTA	—	—	—	—	—	—	—	
PORTD	—	—	—	—	—	—	—	
P1DR	初期化	—	—	—	—	—	初期化	
P3DR	初期化	—	—	—	—	—	初期化	
P6DR	初期化	—	—	—	—	—	初期化	

レジスタ 略称	リセット	スリープ モード	サブ クロック モード*	モジュール ストップ モード	全モジュール クロック ストップモード	ソフトウェア スタンバイ モード	ハードウェア スタンバイ モード*	モジュール
PADR	初期化	—	—	—	—	—	初期化	I/O ポート
PDDR	初期化	—	—	—	—	—	初期化	
PCR	初期化	—	—	—	—	—	初期化	PPG
PMR	初期化	—	—	—	—	—	初期化	
NDERH	初期化	—	—	—	—	—	初期化	
NDERL	初期化	—	—	—	—	—	初期化	
PODRH	初期化	—	—	—	—	—	初期化	
PODRL	初期化	—	—	—	—	—	初期化	
NDRH	初期化	—	—	—	—	—	初期化	
NDRL	初期化	—	—	—	—	—	初期化	
TCSR	初期化	—	—	—	—	—	初期化	WDT
TCNT	初期化	—	—	—	—	—	初期化	
RSTCSR	初期化	—	—	—	—	—	初期化	
TSTR	初期化	—	—	—	—	—	初期化	TPU
TSYR	初期化	—	—	—	—	—	初期化	(ユニット 0)
TCR_0	初期化	—	—	—	—	—	初期化	TPU_0
TMDR_0	初期化	—	—	—	—	—	初期化	
TIORH_0	初期化	—	—	—	—	—	初期化	
TIORL_0	初期化	—	—	—	—	—	初期化	
TIER_0	初期化	—	—	—	—	—	初期化	
TSR_0	初期化	—	—	—	—	—	初期化	
TCNT_0	初期化	—	—	—	—	—	初期化	
TGRA_0	初期化	—	—	—	—	—	初期化	
TGRB_0	初期化	—	—	—	—	—	初期化	
TGRC_0	初期化	—	—	—	—	—	初期化	
TGRD_0	初期化	—	—	—	—	—	初期化	
TCR_1	初期化	—	—	—	—	—	初期化	TPU_1
TMDR_1	初期化	—	—	—	—	—	初期化	
TIOR_1	初期化	—	—	—	—	—	初期化	
TIER_1	初期化	—	—	—	—	—	初期化	
TSR_1	初期化	—	—	—	—	—	初期化	
TCNT_1	初期化	—	—	—	—	—	初期化	
TGRA_1	初期化	—	—	—	—	—	初期化	
TGRB_1	初期化	—	—	—	—	—	初期化	

レジスタ 略称	リセット	スリープ モード	サブ クロック モード*	モジュール ストップ モード	全モジュール クロック ストップモード	ソフトウェア スタンバイ モード	ハードウェア スタンバイ モード*	モジュール
TCR_2	初期化	—	—	—	—	—	初期化	TPU_2
TMDR_2	初期化	—	—	—	—	—	初期化	
TIOR_2	初期化	—	—	—	—	—	初期化	
TIER_2	初期化	—	—	—	—	—	初期化	
TSR_2	初期化	—	—	—	—	—	初期化	
TCNT_2	初期化	—	—	—	—	—	初期化	
TGRA_2	初期化	—	—	—	—	—	初期化	
TGRB_2	初期化	—	—	—	—	—	初期化	
TCR_3	初期化	—	—	—	—	—	初期化	
TMDR_3	初期化	—	—	—	—	—	初期化	
TIORH_3	初期化	—	—	—	—	—	初期化	
TIORL_3	初期化	—	—	—	—	—	初期化	
TIER_3	初期化	—	—	—	—	—	初期化	
TSR_3	初期化	—	—	—	—	—	初期化	
TCNT_3	初期化	—	—	—	—	—	初期化	
TGRA_3	初期化	—	—	—	—	—	初期化	
TGRB_3	初期化	—	—	—	—	—	初期化	
TGRC_3	初期化	—	—	—	—	—	初期化	
TGRD_3	初期化	—	—	—	—	—	初期化	

【注】 * 本 LSI には、ハードウェアスタンバイモードおよびサブクロックモードはありません。

25. 電気的特性

25.1 絶対最大定格

表 25.1 絶対最大定格

項目	記号	定格値	単位
電源電圧	V _{cc}	-0.3~+7.0	V
入力電圧 (ポート 4、ポート 5 以外)	V _{in}	-0.3~V _{cc} +0.3	V
入力電圧 (ポート 4)	V _{in}	-0.3~AV _{cc1} +0.3	V
入力電圧 (ポート 5)	V _{in}	-0.3~AV _{cc0} +0.3	V
アナログ電源電圧	AV _{cc0}	-0.3~+7.0	V
	AV _{cc1}	-0.3~+7.0	V
アナログ入力電圧 (ポート 4)	V _{AN}	-0.3~AV _{cc1} +0.3	V
アナログ入力電圧 (ポート 5)	V _{AN}	-0.3~AV _{cc0} +0.3	V
動作温度	T _{opr}	-40~+85	°C
保存温度	T _{stg}	-55~+125	°C

【使用上の注意】 絶対最大定格を超えて LSI を使用した場合、LSI 永久破壊となることがあります。

25.2 DC 特性

表 25.2 DC 特性 (1)

条件 : $V_{CC}=4.5\sim 5.5V$ 、 $AV_{CC0}=4.5\sim 5.5V$ 、 $AV_{CC1}=4.5\sim 5.5V$ 、
 $V_{SS}=AV_{SS}=0V^{*1}$
 $T_a=-40\sim +85^{\circ}C$

項目		記号	min	typ	max	単位	測定条件
シュミット トリガ 入力電圧	\overline{IRQ} 入力端子、	V_{T^-}	$V_{CC} \times 0.2$	—	—	V	
	TPU 入力端子	V_{T^+}	—	—	$V_{CC} \times 0.7$		
	ポート 3、J、K	$V_{T^+} - V_{T^-}$	0.4	—	—		
入力 High レベル電圧 (シュミット トリガ入力 端子を除く)	EMLE、MD 端子、 \overline{RES} 、NMI	V_{IH}	$V_{CC} - 0.7$	—	$V_{CC} + 0.3$	V	
	EXTAL		$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$		
	その他の入力端子		$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$		
	ポート 4		$AV_{CC1} \times 0.7$	—	$AV_{CC1} + 0.3$		
	ポート 5		$AV_{CC0} \times 0.7$	—	$AV_{CC0} + 0.3$		
入力 Low レベル電圧 (シュミット トリガ入力 端子を除く)	EMLE、 \overline{RES} 、 MD 端子、NMI	V_{IL}	-0.3	—	$V_{CC} \times 0.1$	V	
	EXTAL		-0.3	—	$V_{CC} \times 0.2$		
	その他の入力端子		-0.3	—	$V_{CC} \times 0.2$		
	ポート 4		-0.3	—	$AV_{CC1} \times 0.2$		
	ポート 5		-0.3	—	$AV_{CC0} \times 0.2$		
出力 High レベル電圧	全出力端子	V_{OH}	$V_{CC} - 0.5$	—	—	V	$I_{OH} = -200 \mu A$
			$V_{CC} - 1.0$	—	—		$I_{OH} = -1mA$
出力 Low レベル電圧	全出力端子	V_{OL}	—	—	0.4	V	$I_{OL} = 1.6mA$
入力リーク 電流	EMLE、 \overline{RES} 、 NMI、MD 端子	$ I_{in} $	—	—	1.0	μA	$V_{in} = 0.5 \sim V_{CC} - 0.5V$
	ポート 4		—	—	1.0		$V_{in} = 0.5 \sim AV_{CC1} - 0.5V$
	ポート 5		—	—	1.0		$V_{in} = 0.5 \sim AV_{CC0} - 0.5V$

表 25.2 DC 特性 (2)

条件 : $V_{cc}=4.5\sim 5.5V$ 、 $AV_{cc0}=4.5\sim 5.5V$ 、 $AV_{cc1}=4.5\sim 5.5V$ 、 $V_{ss}=AV_{ss}=0V^{*1}$ $T_a=-40\sim +85^{\circ}C$

項目		記号	min	typ	max	単位	測定条件
スリーステート リーク電流 (オフ状態)	ポート 1、6、 A、D、H、J、K	$ I_{rs} $	—	—	1.0	μA	$V_{in}=0.5\sim V_{cc}-0.5V$
入力プルアップ MOS 電流	ポート D、H、J、K	$-I_p$	50	—	300	μA	$V_{in}=0V$
入力プルダウン MOS 電流	ポート 1、6、A、D、H	I_p	45	—	300	μA	$V_{in}=V_{cc}$
アナログポート プルダウン MOS 電流	ポート 4、5	I_p	5	—	50	μA	$V_{in}=AV_{cc0}$ 、 AV_{cc1}
入力容量	全入力端子	C_{in}	—	—	15	pF	$V_{in}=0V$ $f=1MHz$ $T_a=25^{\circ}C$
消費電流 ^{*2}	通常動作時	I_{cc}^{*4}	—	50	70	mA	Max 条件 $V_{cc}=5.5V$ EXTAL 10MHz $I\phi=80MHz$ $R\phi=P\phi=A\phi=F\phi=40MHz$ $B\phi=20MHz$
消費電流 ^{*2}	スリープ時	I_{cc}^{*4}	—	38	55	mA	Max 条件 $V_{cc}=5.5V$ EXTAL 10MHz $I\phi=80MHz$ $R\phi=P\phi=A\phi=F\phi=40MHz$ $B\phi=20MHz$
	スタンバイ時 ^{*3}		—	10	25	mA	$T_a\leq 50^{\circ}C$
			—	—	—	25	mA
全モジュールクロック ストップ時 ^{*5}			—	36	45		

項目		記号	min	typ	max	単位	測定条件
アナログ 電源電流	A/D 変換中	Alcc0	—	3.5	5	mA	AVcc0=5.0V
	A/D 変換待機時		—	10	100	μA	
	A/D 変換中	Alcc1	—	3.5	5	mA	AVcc1=5.0V
	A/D 変換待機時		—	10	100	μA	
RAM スタンバイ電圧		V _{RAM}	3.0	—	—	V	
Vcc 開始電圧	Vcc 端子	VCCSTART	0	—	0.2	V	
Vcc 立ち上がり 勾配		SVcc	—	—	0.01	V/us	

- 【注】 *1 A/D 変換器未使用時に AVcc0、AVcc1、AVss 端子を開放しないでください。
AVcc0、AVcc1 端子は Vcc に、AVss 端子は Vss にそれぞれ接続してください。
- *2 消費電流値は、V_{IH}=AVcc0（ポート 5）、AVcc1（ポート 4）、Vcc（その他）、V_{IL}=0V の条件下で、すべての出力端子を無負荷状態にして、さらに内蔵プルアップ MOS をオフ状態にした場合の値です。
- *3 V_{RAM} ≤ Vcc < 4.5V のとき、V_{IHmin}=Vcc-0.1V、V_{ILmax}=0.1V とした場合の値です。
- *4 ICC は下記の式に従って Vcc と f に依存します。
I_{ccmax} = 21 (mA) + 0.26 (mA/MHz) × f + 5.0 (mA/V) × Vcc（通常動作時）
I_{ccmax} = 21 (mA) + 0.35 (mA/MHz) × f + 1.0 (mA/V) × Vcc（スリープ時）
- *5 参考値です。

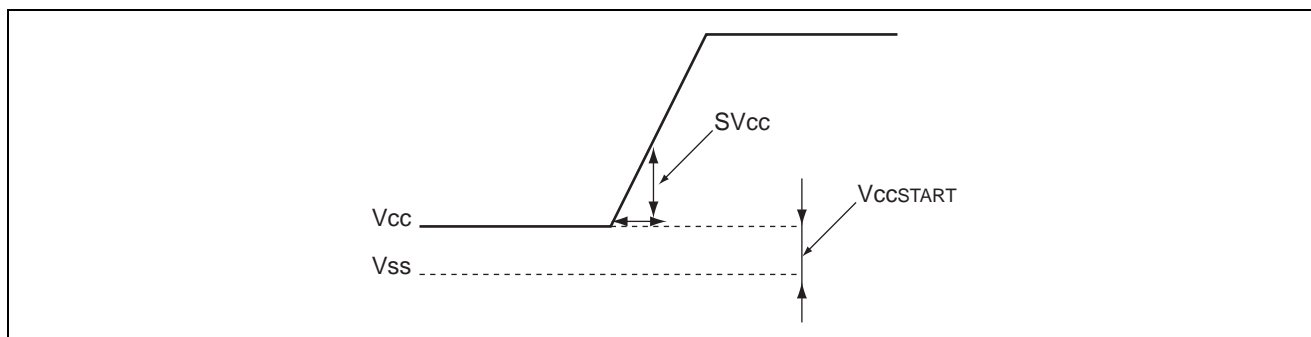


図 25.1 Vcc 電源立ち上がり

表 25.3 出力許容電流値

条件 : $V_{cc}=4.5\sim 5.5V$ 、 $AV_{cc0}=4.5\sim 5.5V$ 、 $AV_{cc1}=4.5\sim 5.5V$ 、
 $V_{ss}=AV_{ss}=0V^*$
 $T_a=-40\sim +85^\circ C$

項目		記号	min	typ	max	単位	測定条件
出力 Low レベル許容電流 (1 端子あたり)	全出力端子	I_{OL}	—	—	10	mA	
出力 Low レベル許容電流 (総和)	全出力端子	ΣI_{OL}	—	—	120	mA	
出力 High レベル許容電流 (1 端子あたり)	全出力端子	$-I_{OH}$	—	—	2.0	mA	
出力 High レベル許容電流 (総和)	全出力端子	$-\Sigma I_{OH}$	—	—	40	mA	

【使用上の注意】 LSI の信頼性を確保するため、出力電流値は表 25.3 の値を超えないようにしてください。

【注】 * A/D 変換器未使用時に AV_{cc0} 、 AV_{cc1} 、 AV_{ss} 端子を開放しないでください。
 AV_{cc0} 、 AV_{cc1} 端子は V_{cc} に、 AV_{ss} 端子は V_{ss} にそれぞれ接続してください。

25.3 AC 特性

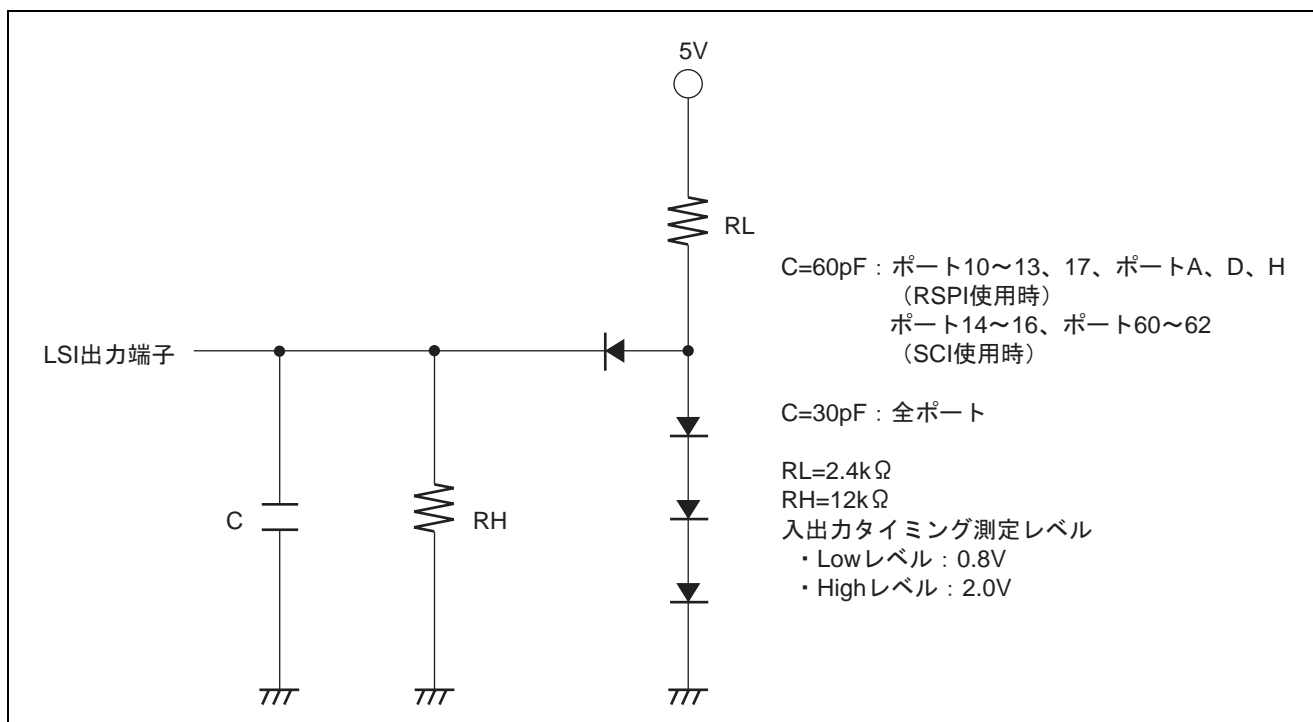


図 25.2 出力負荷回路

25.3.1 クロックタイミング

表 25.4 にクロックタイミングを示します。

表 25.4 クロックタイミング

条件 : $V_{cc}=4.5\sim 5.5V$ 、 $AV_{cc0}=4.5\sim 5.5V$ 、 $AV_{cc1}=4.5\sim 5.5V$ 、
 $V_{ss}=AV_{ss}=0V$
 $T_a=-40\sim +85^\circ C$

項目	記号	Min.	Max.	単位	参照図
外部入力クロック周波数	f_{EX}	8	10	MHz	図 25.3
外部入力クロックサイクル時間	t_{EXCyc}	100	125	ns	
外部入力クロックハイレベルパルス幅	t_{EXH}	40		ns	
外部入力クロックローレベルパルス幅	t_{EXL}	40	—	ns	
外部入力クロック立ち上がり時間	t_{EXr}	—	5	ns	
外部入力クロック立ち下がり時間	t_{EXf}	—	5	ns	
リセット発振安定時間 (水晶)	t_{OSC1}	20	—	ms	図 25.5
ソフトウェアスタンバイ発振安定時間 (水晶)	t_{OSC2}	10	—	ms	図 25.4
外部クロック発振安定時間	t_{DEXT}	2	—	ms	図 25.5
システムクロック (I ϕ) 周波数	f_I	8	80	MHz	図 25.6
システムクロック (I ϕ) サイクル時間	t_{Icyc}	12.5	125	ns	
周辺モジュールクロック (P ϕ) 周波数	f_P	8	40	MHz	図 25.7
周辺モジュールクロック (P ϕ) サイクル時間	t_{Pcyc}	25	125	ns	
外部バスクロック (B ϕ) 周波数	f_{op}	8	20	MHz	図 25.8
外部バスクロック (B ϕ) サイクル時間	t_{cyc}	50	125	ns	
外部バスクロック (B ϕ) ハイレベルパルス幅	t_{CH}	15	—	ns	
外部バスクロック (B ϕ) ローレベルパルス幅	t_{CL}	15	—	ns	
外部バスクロック (B ϕ) 立ち上がり時間	t_{Cr}	—	10	ns	
外部バスクロック (B ϕ) 立ち下がり時間	t_{Cf}	—	10	ns	
FLASH クロック (F ϕ) 周波数	f_F	8	40	MHz	図 25.9
FLASH クロック (F ϕ) サイクル時間	t_{Fcyc}	25	125	ns	
A/D クロック (A ϕ) 周波数	f_A	8	40	MHz	図 25.10
A/D クロック (A ϕ) サイクル時間	t_{Acyc}	25	125	ns	
RSPI クロック (R ϕ) 周波数	f_R	8	40	MHz	図 25.11
RSPI クロック (R ϕ) サイクル時間	t_{Rcyc}	25	125	ns	

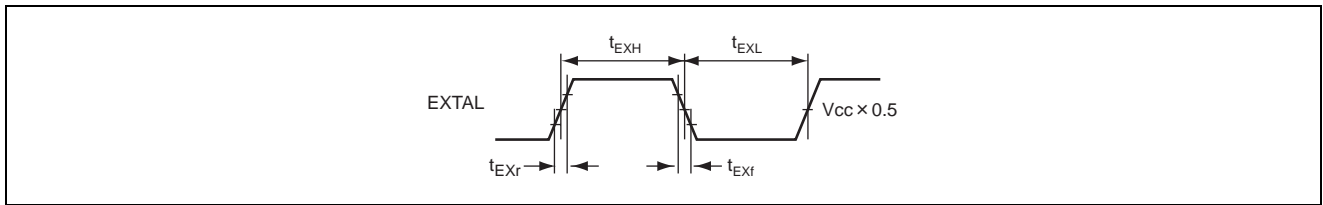


図 25.3 外部入力クロックタイミング

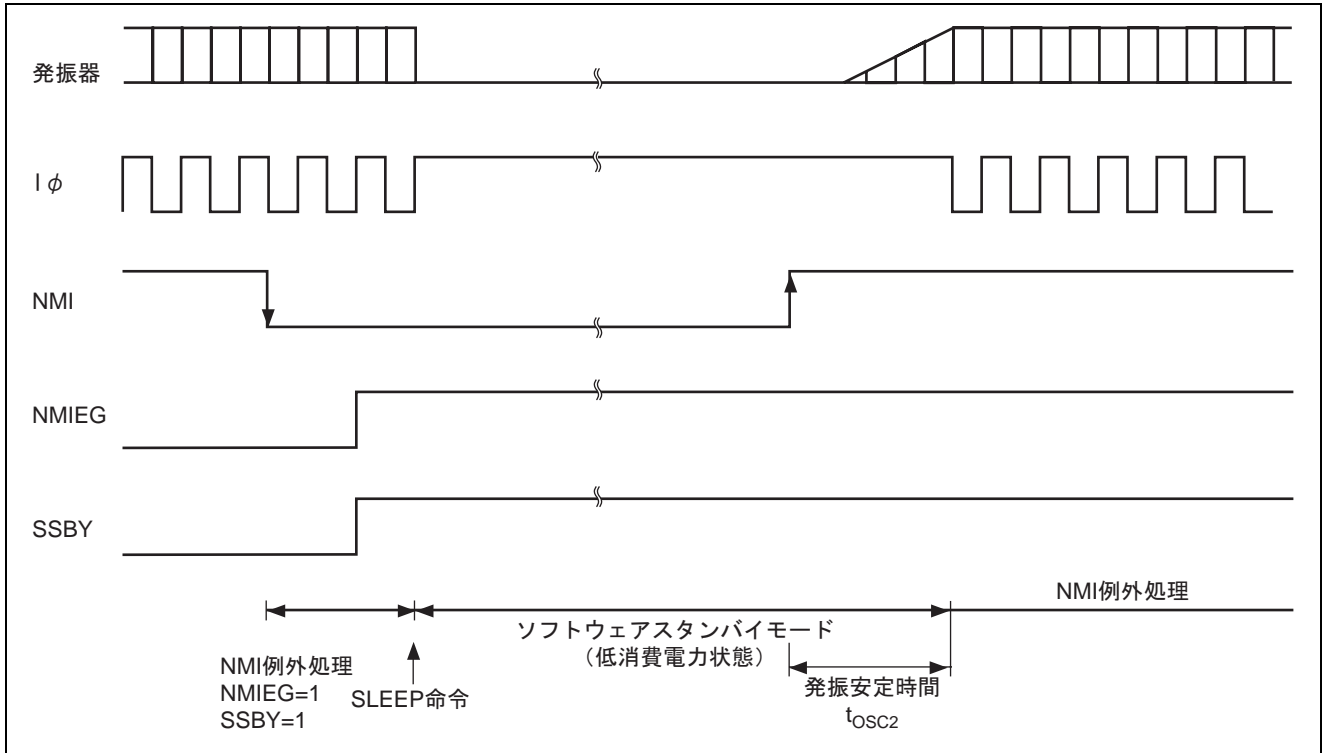


図 25.4 ソフトウェアスタンバイ発振安定時間タイミング

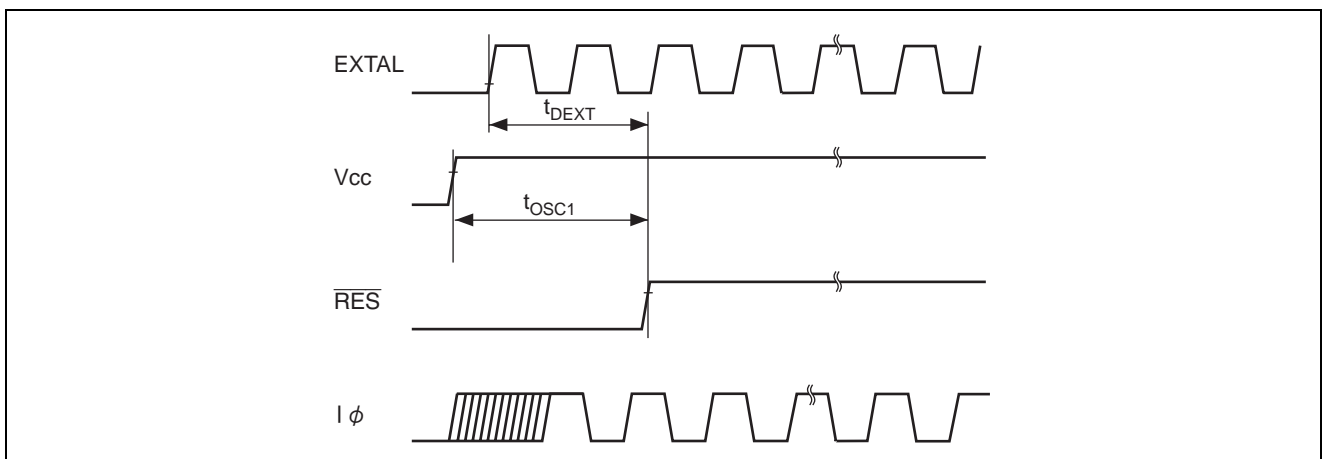


図 25.5 発振安定時間タイミング

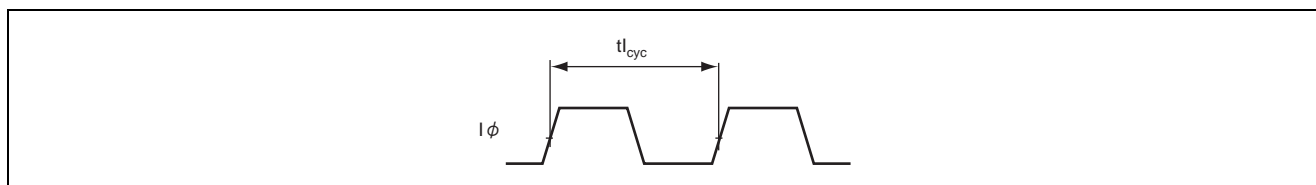


図 25.6 システムクロックタイミング

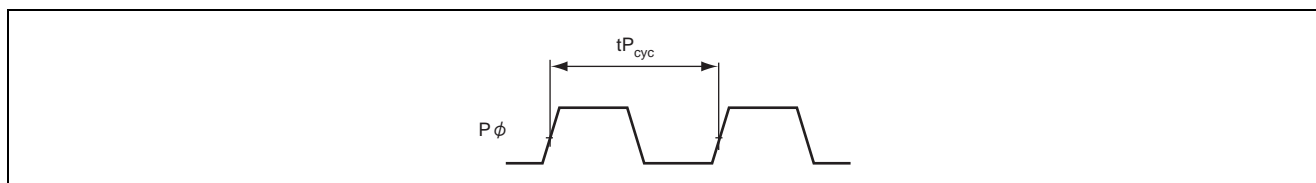


図 25.7 周辺モジュールクロックタイミング

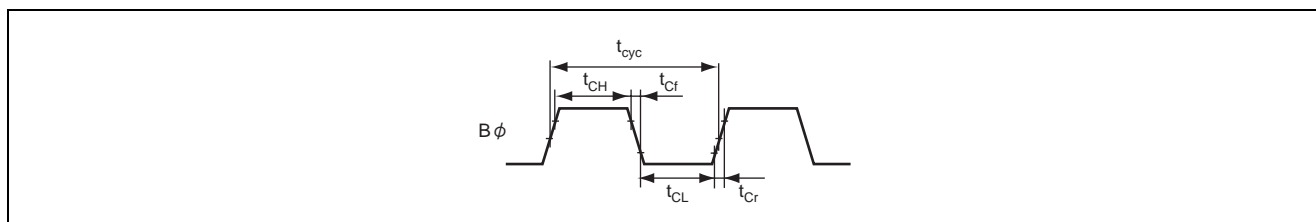


図 25.8 外部バスクロックタイミング

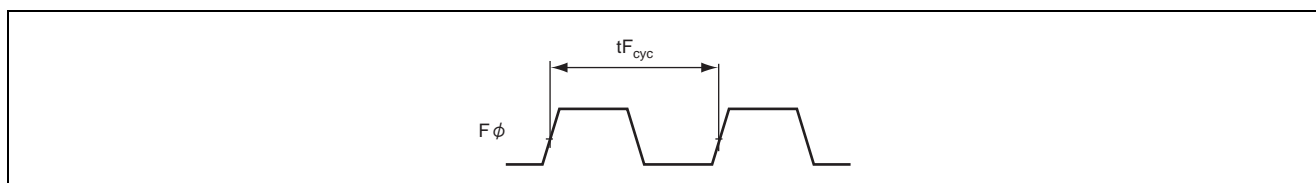


図 25.9 FLASH クロックタイミング

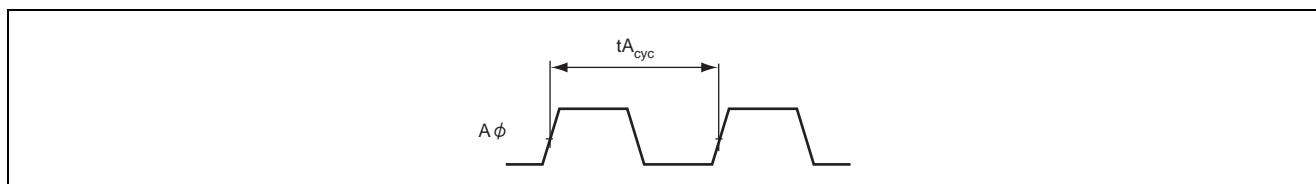


図 25.10 A/D クロックタイミング

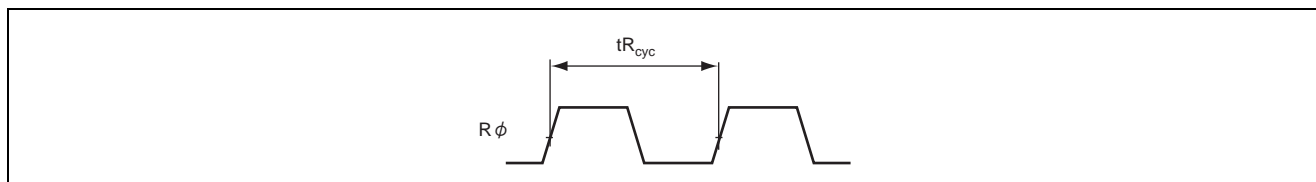


図 25.11 RSPI クロックタイミング

25.3.2 制御信号タイミング

表 25.5 制御信号タイミング

条件 : $V_{cc}=4.5\sim 5.5V$ 、 $AV_{cc0}=4.5\sim 5.5V$ 、 $AV_{cc1}=4.5\sim 5.5V$ 、
 $V_{ss}=AV_{ss}=0V$ 、 $f_{\phi}=8\sim 80MHz$
 $T_a=-40\sim +85^{\circ}C$

項目	記号	min	Max	単位	測定条件
\overline{RES} セットアップ時間	t_{RESS}	1000	—	ns	図 25.12
\overline{RES} パルス幅 (フラッシュメモリ書き込み/消去を除く)	t_{RESW}	100	—	t_{cyc}	
\overline{RES} パルス幅 (フラッシュメモリ書き込み/消去中) *	t_{RESW2}	20	—	μs	
NMI セットアップ時間	t_{NMIS}	150	—	ns	図 25.13
NMI ホールド時間	t_{NMIH}	10	—	ns	
NMI パルス幅 (ソフトウェアスタンバイモードからの復帰時)	t_{NMIW}	200	—	ns	
\overline{IRQ} セットアップ時間	t_{IRQS}	150	—	ns	
\overline{IRQ} ホールド時間	t_{IRQH}	10	—	ns	
\overline{IRQ} パルス幅 (ソフトウェアスタンバイモードからの復帰時)	t_{IRQW}	200	—	ns	

【注】 * 内蔵フラッシュの書き込み/消去中に発生する LSI 内部の高電圧を放電するための待ち時間です。

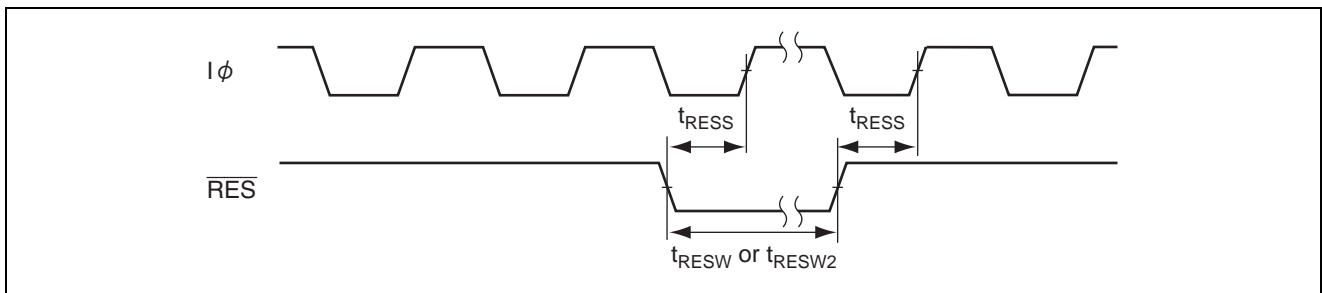


図 25.12 リセット入力タイミング

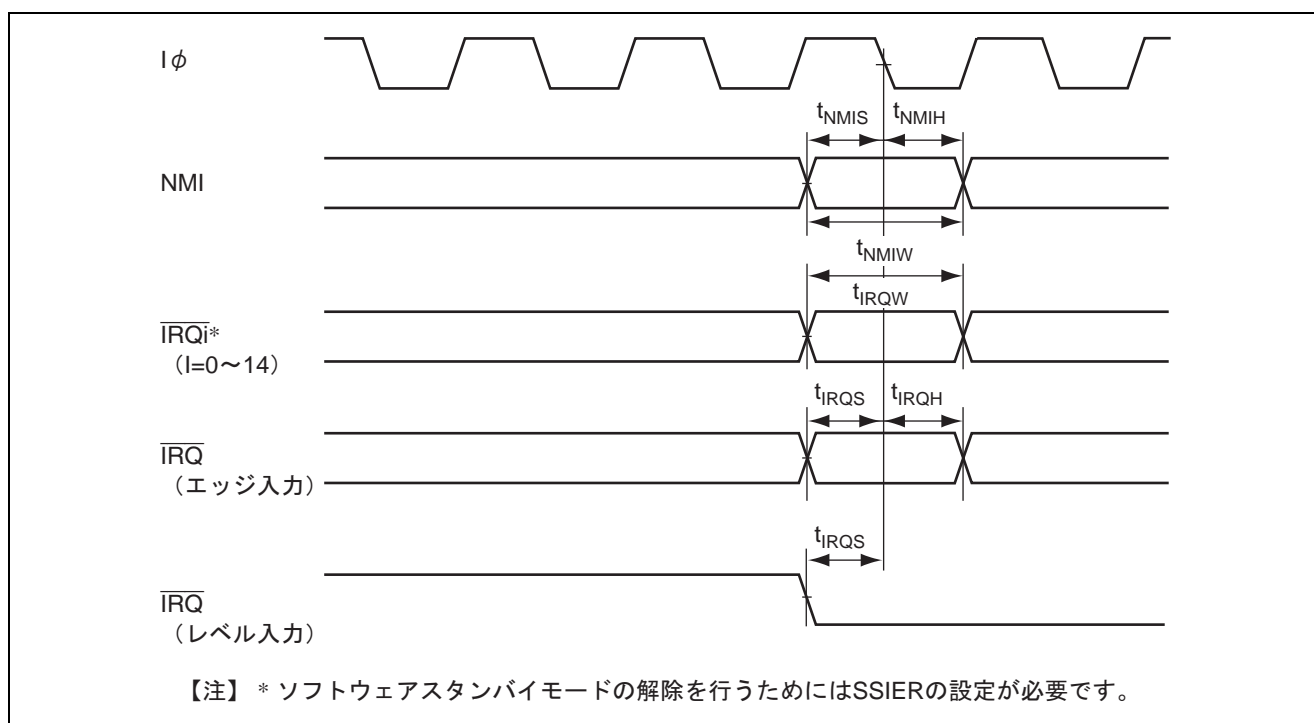


図 25.13 割り込み入力タイミング

25.3.3 内蔵周辺モジュールタイミング

表 25.6 内蔵周辺モジュールタイミング (1)

条件 : $V_{cc}=4.5\sim 5.5V$ 、 $AV_{cc0}=4.5\sim 5.5V$ 、 $AV_{cc1}=4.5\sim 5.5V$ 、
 $V_{ss}=AV_{ss}=0V$ 、 $P\phi=8\sim 40MHz$ 、 $A\phi=8\sim 40MHz$
 $T_a=-40\sim +85^\circ C$

項目		記号	min	max	単位	測定条件	
I/O ポート	出力データ遅延時間	High* ¹	-	40	ns	図 25.14	
		Low* ²		90			
	入力データセットアップ時間		t _{PRS}	25	-		ns
	入力データホールド時間		t _{PRH}	25	-		ns
TPU	タイマ出力遅延時間	High* ¹	-	40	ns	図 25.15	
		Low* ²		90			
	タイマ入力セットアップ時間		t _{TICS}	25	-	ns	
	タイマクロック入力セットアップ時間		t _{TCKS}	25	-	ns	図 25.16
	タイマクロックパルス幅	単エッジ指定	t _{TCKWH}	1.5	-	t _{pcyc}	
両エッジ指定		t _{TCKWL}	2.5	-	t _{pcyc}		
PPG	パルス出力遅延時間	High* ¹	-	40	ns	図 25.17	
		Low* ²		90			
SCI	入カクロックサイクル	調歩同期	t _{Scyc}	4	-	t _{pcyc}	図 25.18
		クロック同期		6	-		
	入カクロックパルス幅		t _{SCKW}	0.4	0.6	t _{Scyc}	
	入カクロック立ち上がり時間		t _{SCKr}	-	20	ns	図 25.18
	入カクロック立ち下がり時間		t _{SCKf}	-	20	ns	電圧参照レベル V _{cc} × 0.3V ~ V _{cc} × 0.7V
	出カクロックサイクル	調歩同期	t _{Scyc}	30	-	t _{pcyc}	図 25.18
		クロック同期		4	-		
	出カクロックパルス幅		t _{SCKW}	0.4	0.6	t _{Scyc}	
	出カクロック立ち上がり時間	High* ¹	t _{SCKr}	-	20	ns	図 25.18
		Low* ²		-	40	ns	電圧参照レベル
	出カクロック立ち下がり時間	High* ¹	t _{SCKf}	-	20	ns	V _{cc} × 0.3V ~
		Low* ²		-	40	ns	V _{cc} × 0.7V
	送信データ遅延時間	High* ¹	t _{TXD}	-	40	ns	図 25.19
Low* ²		-		90	ns		
受信データセットアップ時間 (クロック同期)		t _{RXS}	30	-	ns		
受信データホールド時間 (クロック同期)		t _{RXH}	30	-	ns		

項目		記号	min	max	単位	測定条件
A/D 変換器	トリガ入力セットアップ時間	t_{TRGS}	30	—	ns	図 25.20

- 【注】 *1 High : I/O ポート ドライブ能力 大
 *2 Low : I/O ポート ドライブ能力 小

表 25.6 内蔵周辺モジュールタイミング (2)

条件 : $V_{CC}=4.5\sim 5.5V$ 、 $AV_{CC0}=4.5\sim 5.5V$ 、 $AV_{CC1}=4.5\sim 5.5V$ 、
 $V_{SS}=AV_{SS}=0V$ 、 $P\phi=16\sim 40MHz$ 、 $T_a=-40\sim +85^\circ C$

項目			記号	min	max	単位	測定条件
RCAN-TL1*1	送信データ遅延時間	High*2	t_{CTXD}	—	100	ns	図 25.21
		Low*3		—	200	ns	
	受信データセットアップ時間		t_{CRXS}	100	—	ns	
	受信データホールド時間		t_{CRXH}	100	—	ns	

- 【注】 *1 RCAN-TL1 の入出力信号は非同期信号ですが、図 25.21 に示された Pφ クロック立ち上がりで同期化されます。
 *2 High : I/O ポート ドライブ能力 大
 *3 Low : I/O ポート ドライブ能力 小

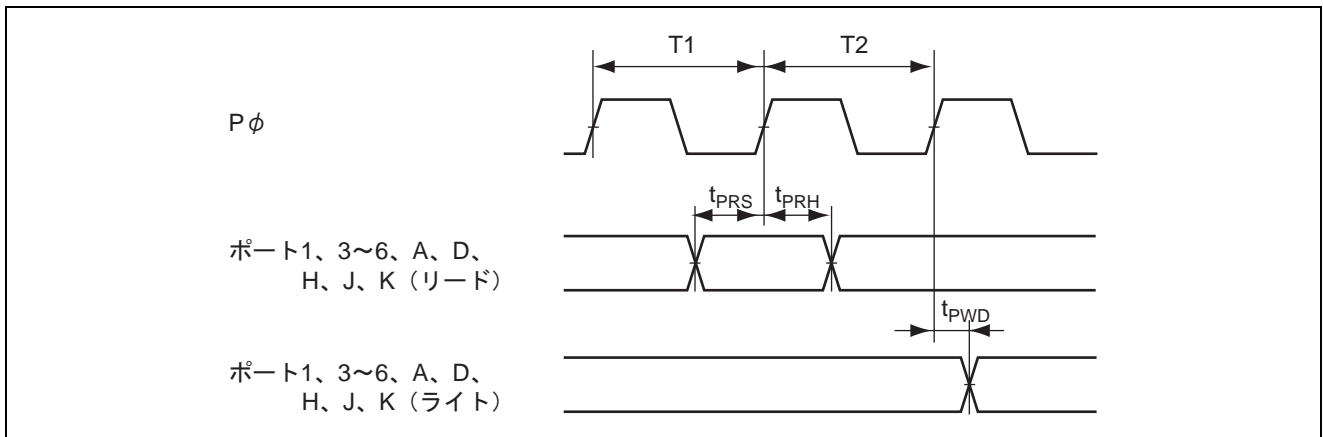


図 25.14 I/O ポート入出力タイミング

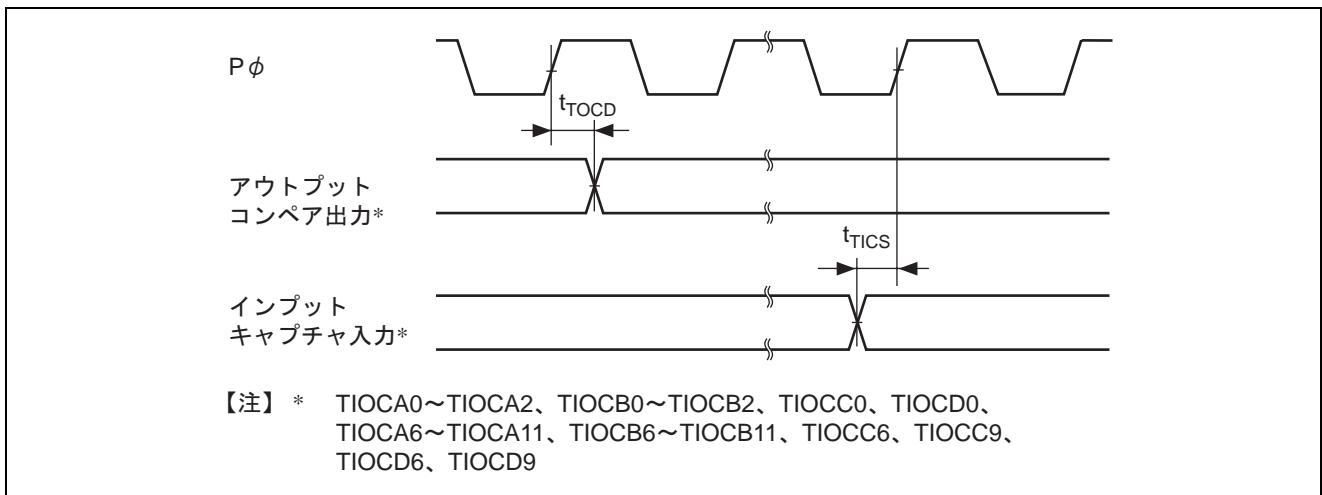


図 25.15 TPU 入出力タイミング

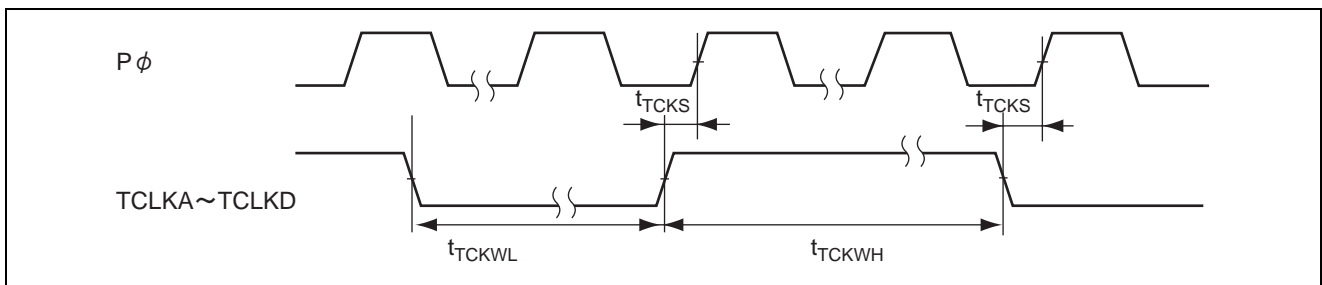


図 25.16 TPU クロック入力タイミング

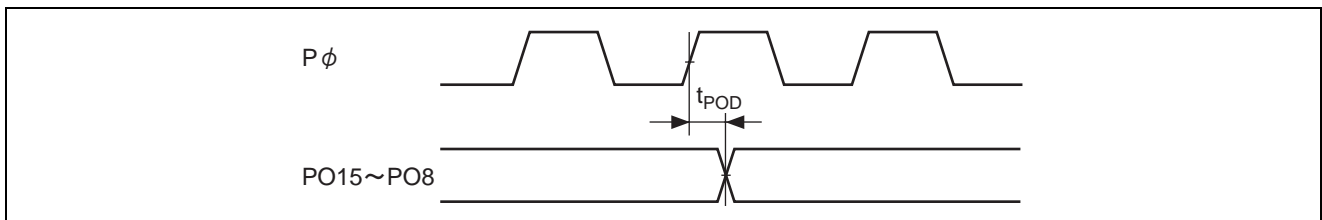


図 25.17 PPG 出力タイミング

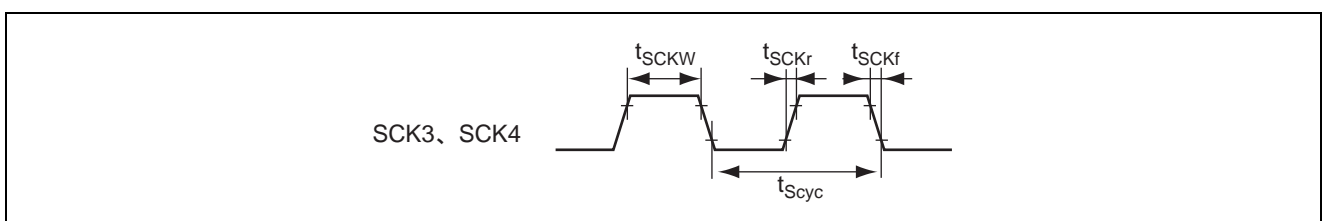


図 25.18 SCK クロック入出力タイミング

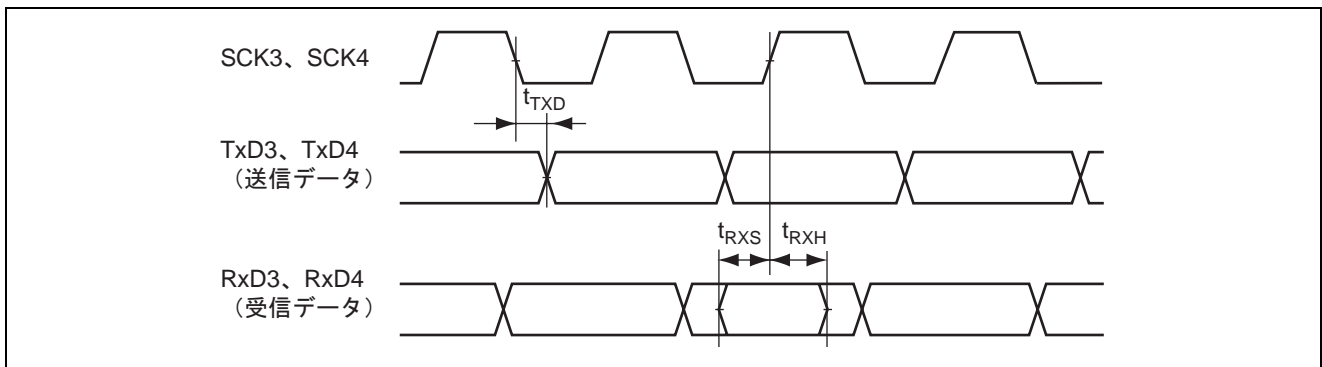


図 25.19 SCI 入出力タイミング/クロック同期式モード

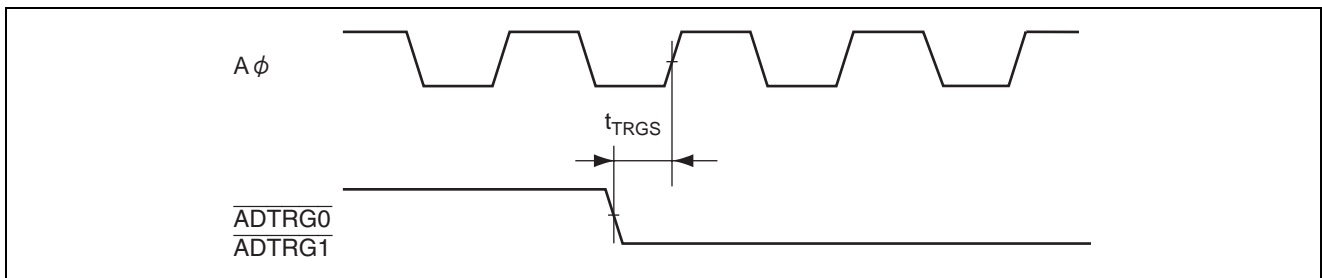


図 25.20 A/D 変換器外部トリガ入力タイミング

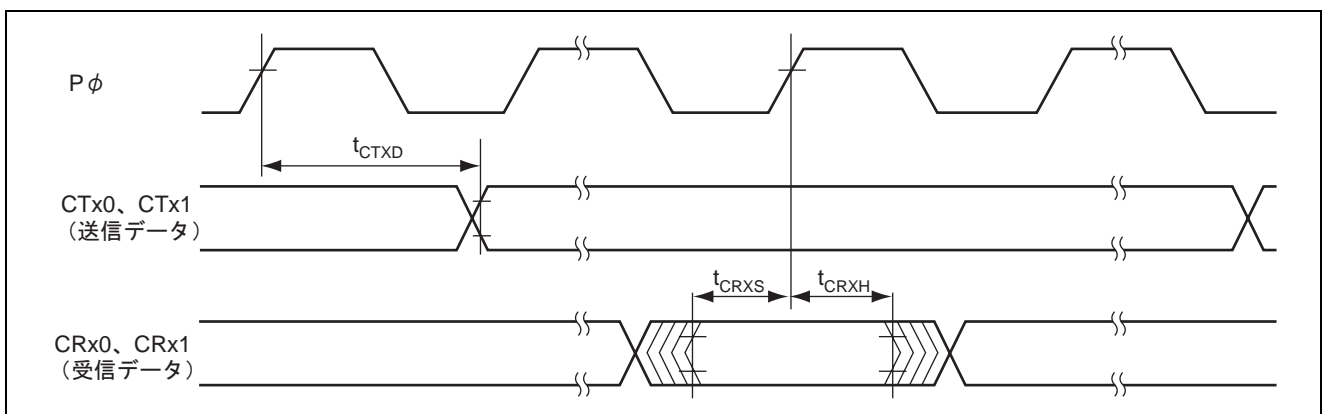


図 25.21 RCAN-TL1 入出力タイミング

表 25.7 RSPI タイミング

条件 : $V_{cc}=4.5\sim 5.5V$ 、 $AV_{cc0}=4.5\sim 5.5V$ 、 $AV_{cc1}=4.5\sim 5.5V$ 、
 $V_{ss}=AV_{ss}=0V$ 、 $R\phi=8\sim 40MHz$
 $T_a=-40\sim +85^\circ C$

項 目		記号	Min.	Typ.	Max.	単位	測定条件
RSPCK クロックサイクル	マスタ	tSPcyc	4	—	4096	tRcyc	図 25.22
	スレーブ		8	—	4096		
RSPCK クロックハイレベルパルス幅	マスタ	tSPCKWH	$(tSPcyc - tSPCKR - tSPCKF) / 2 - 3$	—	—	ns	
	スレーブ		$(tSPcyc - tSPCKR - tSPCKF) / 2$	—	—		
RSPCK クロックローレベルパルス幅	マスタ	tSPCKWL	$(tSPcyc - tSPCKR - tSPCKF) / 2 - 3$	—	—	ns	
	スレーブ		$(tSPcyc - tSPCKR - tSPCKF) / 2$	—	—		
RSPCK クロック 立ち上がり/立ち下がり時間	出力	High* ¹	—	3	5	ns	
		Low* ²	—	20	40		
	入力	—	—	1	μs		
データ入力セットアップ時間	マスタ	High* ¹	25	—	—	ns	
		Low* ²	75	—	—		
	スレーブ	$20 \cdot 2 \times tRcyc$	—	—			
データ入力ホールド時間	マスタ	tH	0	—	—	ns	
	スレーブ		20	—	—		
SSL セットアップ時間	マスタ	tLEAD	1	—	8	tSPcyc	
	スレーブ		4	—	—	tRcyc	
SSL ホールド時間	マスタ	tLAG	1	—	8	tSPcyc	
	スレーブ		4	—	—	tRcyc	
データ出力遅延時間	マスタ	High* ¹	—	—	25	ns	
		Low* ²	—	—	75	ns	
	スレーブ	High* ¹	—	—	$3 \times tRcyc + 40$	ns	
		Low* ²	—	—	$3 \times tRcyc + 90$		
データ出力ホールド時間	マスタ	tOH	0	—	—	ns	
	スレーブ		0	—	—		
連続送信遅延時間	マスタ	tTD	$tSPcyc + 2 \times tRcyc$	—	$8 \times tSPcyc + 2 \times tRcyc$	ns	
	スレーブ		$4 \times tRcyc$	—	—		

項目		記号	Min.	Typ.	Max.	単位	測定条件
MOSI、MISO 立ち上がり ／立ち下がり時間	出力	High* ¹	—	—	5	ns	図 25.23～ 25.26
		Low* ²	—	—	55		
	入力		—	—	1	μs	
SSL 立ち上がり／立ち下がり 時間	出力	High* ¹	—	—	5	ns	
		Low* ²	—	—	55		
	入力		—	—	1	μs	
スレーブアクセス時間		tSA	—	—	4	tRcyc	図 25.25、 25.26
スレーブ出力解放時間		tREL	—	—	3	tRcyc	

【注】 *1 High : I/O ポート ドライブ能力 大

*2 Low : I/O ポート ドライブ能力 小

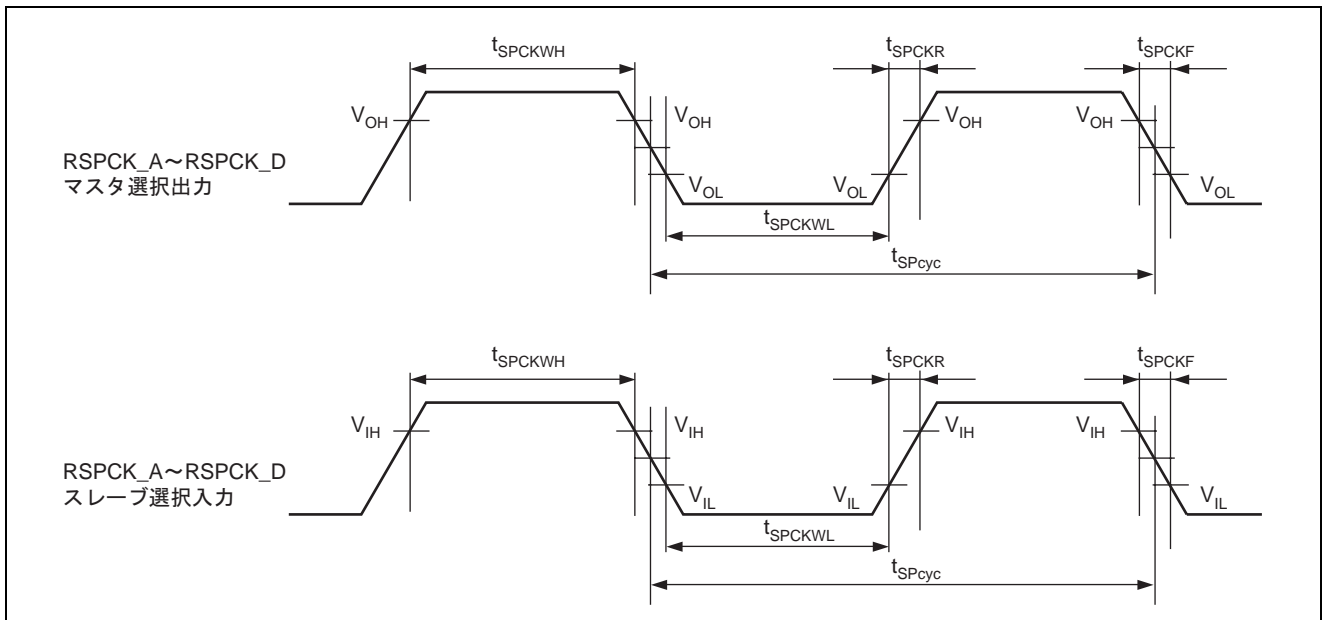


図 25.22 RSPI クロックタイミング

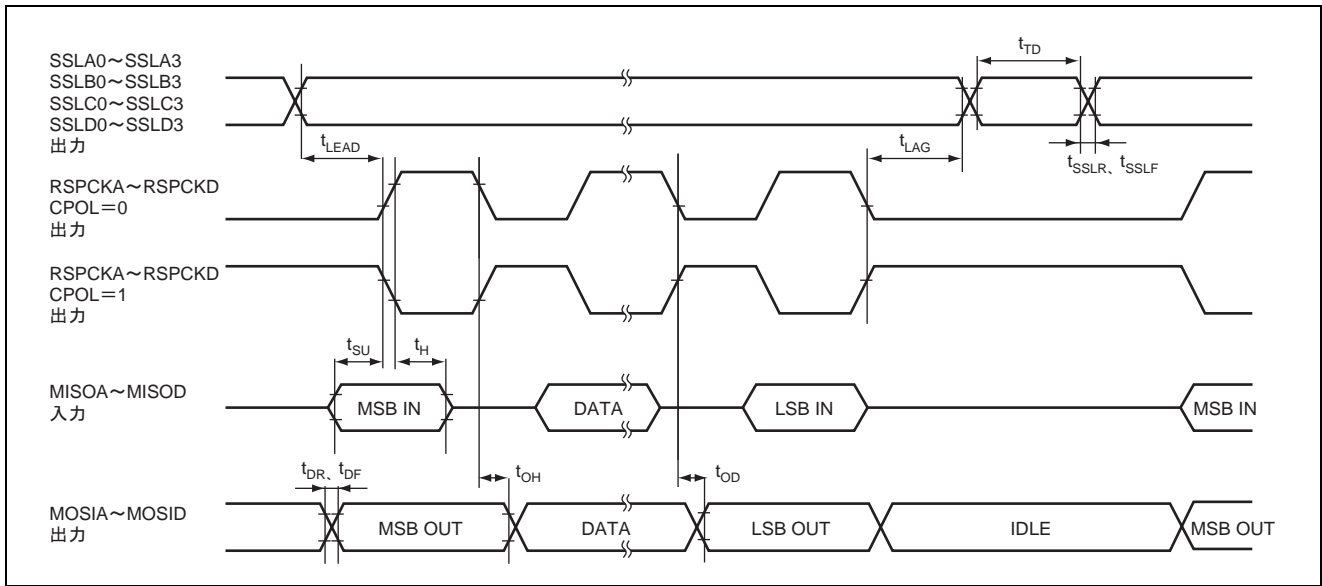


図 25.23 RSPI タイミング (マスタ、CPHA=0)

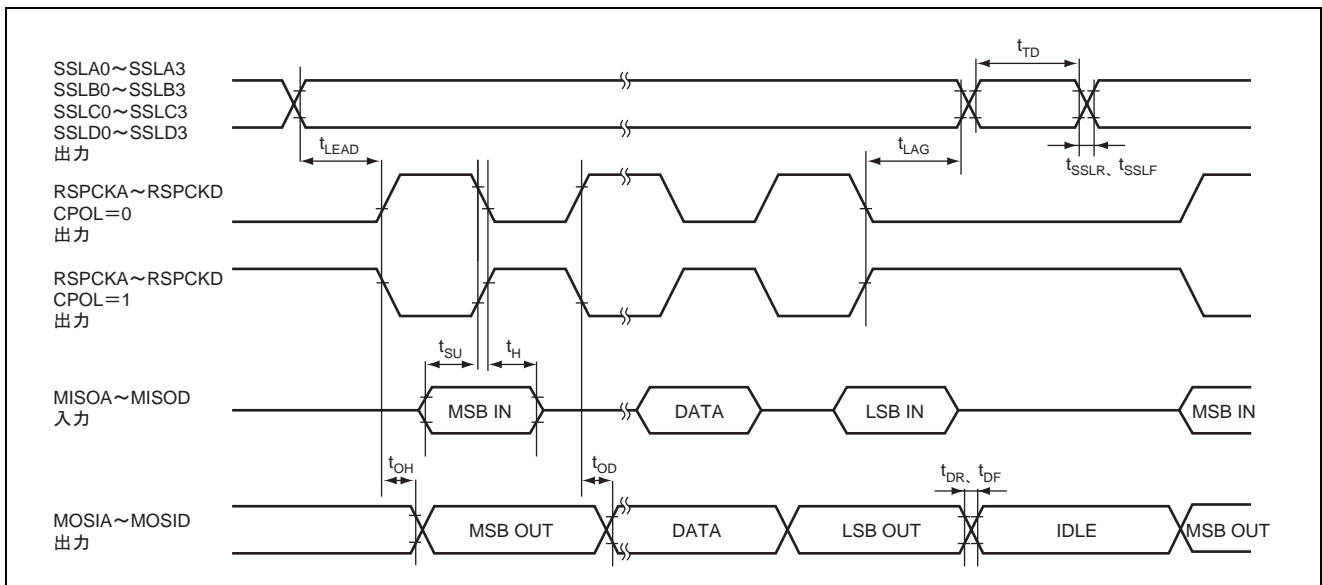


図 25.24 RSPI タイミング (マスタ、CPHA=1)

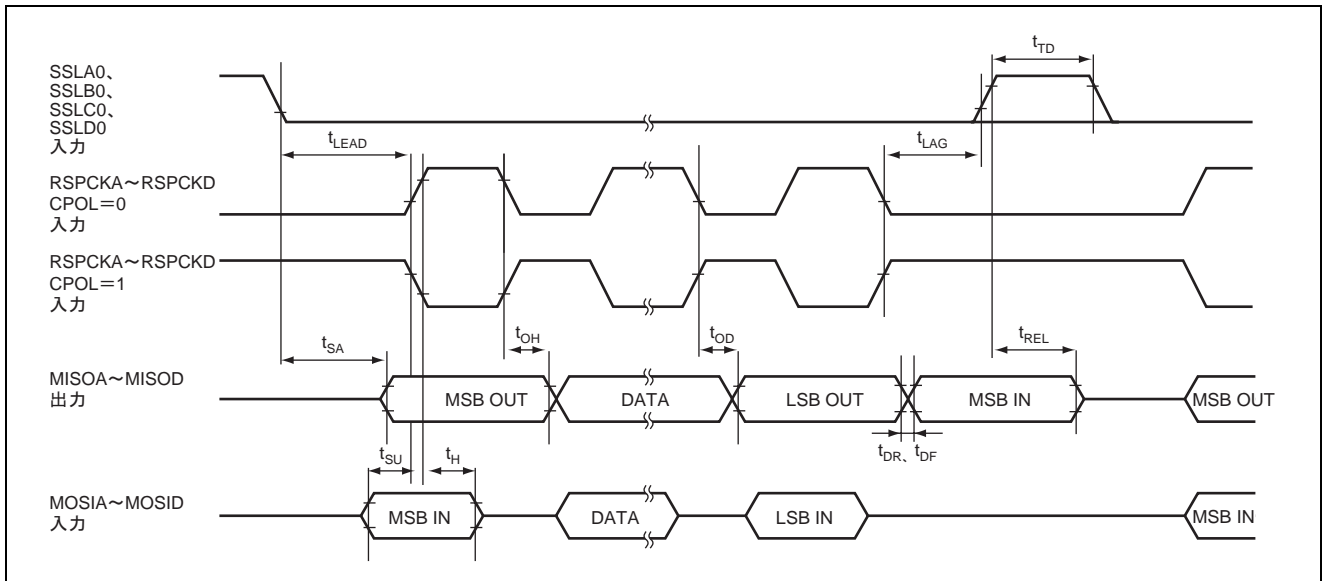


図 25.25 RSPI タイミング (スレーブ、CPHA=0)

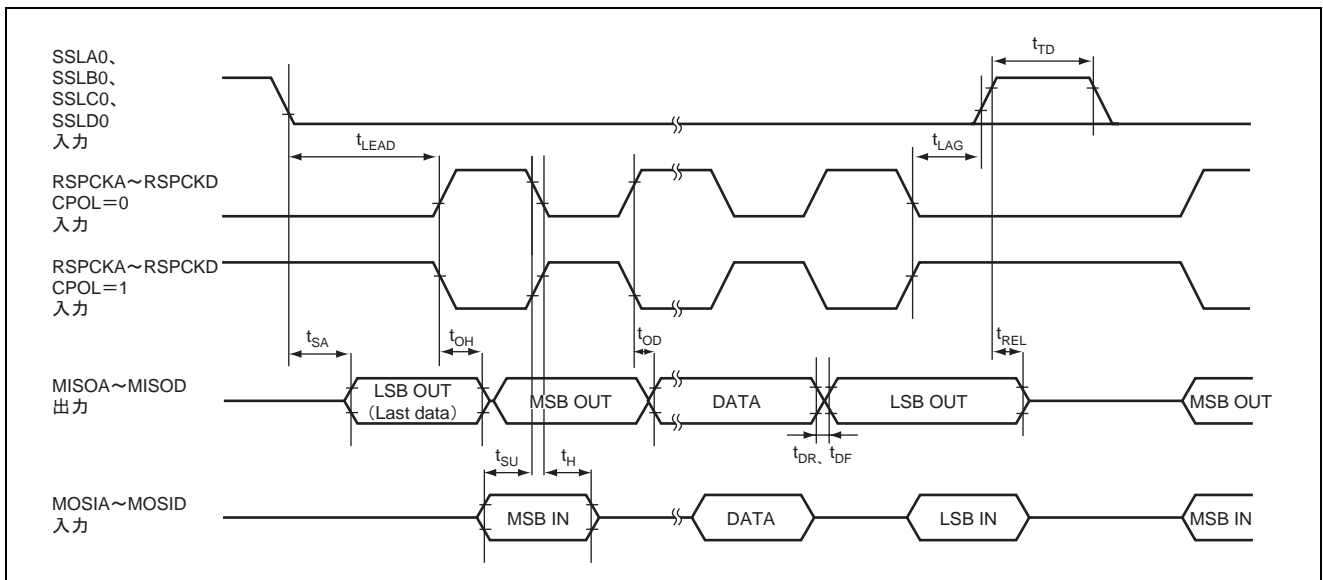


図 25.26 RSPI タイミング (スレーブ、CPHA=1)

25.3.4 A/D 変換特性

表 25.8 A/D 変換特性

条件 : $V_{cc}=4.5\sim 5.5V$ 、 $AV_{cc0}=4.5\sim 5.5V$ 、 $AV_{cc1}=4.5\sim 5.5V$ 、
 $V_{ss}=AV_{ss}=0V$ 、 $A\phi=8\sim 40MHz$
 $T_a=-40\sim +85^{\circ}C$

項 目	min	typ	max	単位
分解能	10	10	10	ビット
変換時間 (40Aφサイクル)	1	—	5	μs
アナログ入力容量	—	—	20	pF
許容信号源インピーダンス	—	—	5	KΩ
非直線性誤差	—	—	±3.5	LSB
オフセット誤差	—	—	±3.5	LSB
フルスケール誤差	—	—	±3.5	LSB
量子化誤差	—	±0.5	—	LSB
絶対精度	—	—	±4.0	LSB
自己診断時の絶対精度	—	—	±40	LSB

25.4 フラッシュメモリ特性

表 25.9 フラッシュメモリ特性

条件 : $V_{CC}=4.5\sim 5.5V$ 、 $AV_{CC0}=4.5\sim 5.5V$ 、 $AV_{CC1}=4.5\sim 5.5V$ 、
 $V_{SS}=AV_{SS}=0V$ 、 $I\phi=8\sim 80MHz$ 、 $F\phi=8\sim 40MHz$
 $T_a=-40\sim +85^{\circ}C$

項 目	記号	Min.	Typ.	Max.	単位	測定条件	
書き込み時間*1、*2、*4	128 バイト	tp128	—	1	10	ms	
	256KB	tp256K	—	1.5	3.2	s	
消去時間*1、*2、*4	4KB	te4K	—	25	60	ms	
	32KB	te32K	—	200	480	ms	
	64KB	te64K	—	400	875	ms	
	256KB	te256K	—	1.6	3.8	s	
再書き込み/消去サイクル	NPEC	100*3	—	—	—	回	
書き込み中のサスペンド遅延時間	tSPD	—	—	120	—	μs	図 25.27
消去中の 1 回目のサスペンド遅延時間 (サスペンド優先モード時)	tsESD1	—	—	120	—	μs	
消去中の 2 回目のサスペンド遅延時間 (サスペンド優先モード時)	tsESD2	—	—	1.7	—	ms	
消去中のサスペンド遅延時間 (消去優先モード時)	tSEED	—	—	1.7	—	ms	
データ保持時間*4	tDRP	20	—	—	—	年	

【注】 *1 書き込み、消去時間はデータに依存します。

*2 書き込み、消去時間にはデータ転送時間は含みません。

*3 書き換え後のすべての特性を保証する min 回数です (保証は 1~min 値の範囲です)。

*4 書き換えが min 値を含む仕様範囲内で行われたときの特性です。

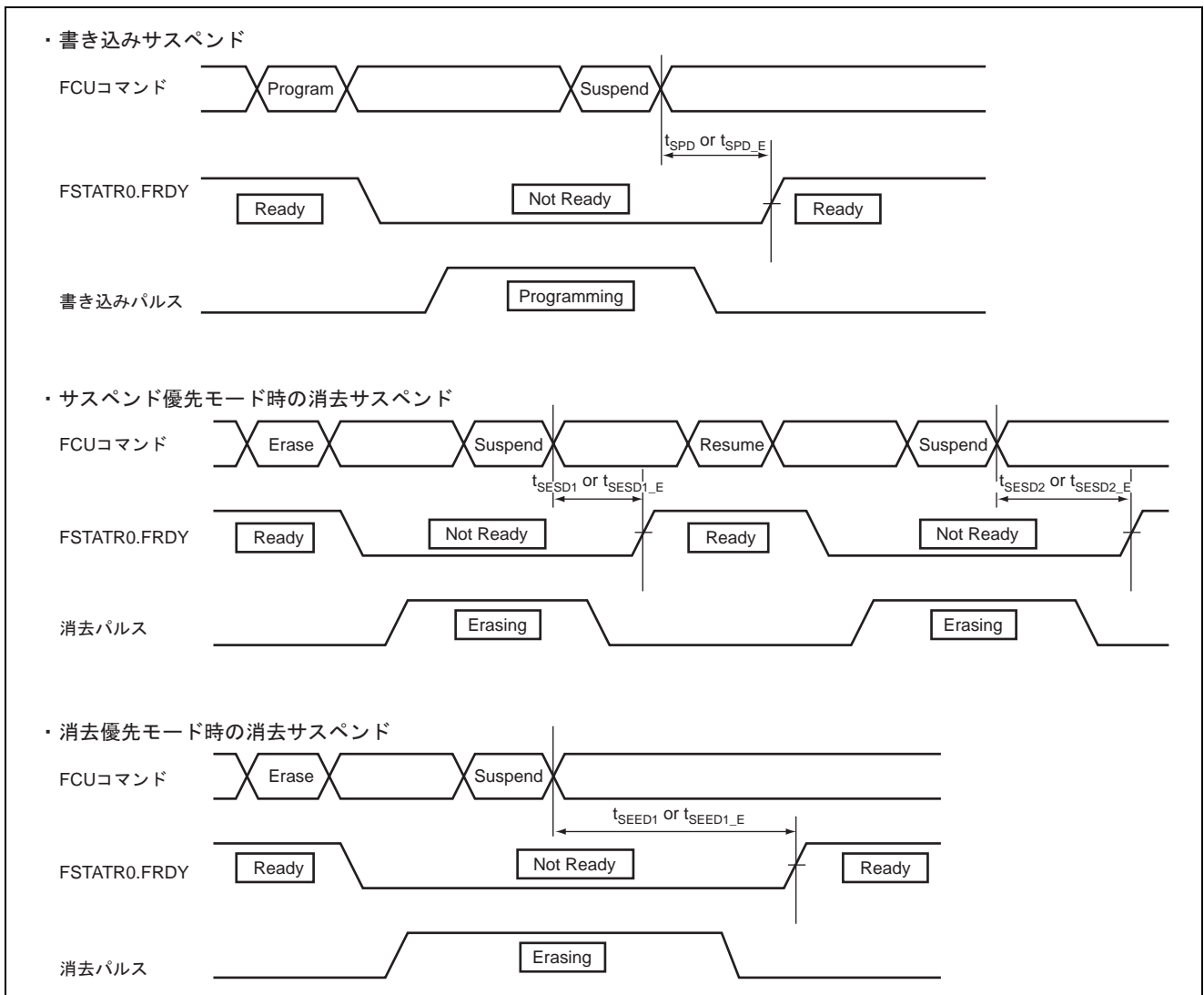


図 25.27 フラッシュメモリ書き込み/消去サスペンドタイミング

25.5 EEPROM 特性

表 25.10 に EEPROM 特性を示します。

表 25.10 EEPROM 特性

条件 : $V_{cc}=4.5\sim 5.5V$ 、 $AV_{cc0}=4.5\sim 5.5V$ 、 $AV_{cc1}=4.5\sim 5.5V$ 、
 $V_{ss}=AV_{ss}=0V$ 、 $I_{\phi}=8\sim 80MHz$ 、 $F_{\phi}=8\sim 40MHz$
 $T_a=-40\sim +85^{\circ}C$

項目	記号	Min.	Typ.	Max.	単位	参照図	
書き込み時間*1、*2、*4	8 バイト	t_{P8_E}	—	0.5	2.6	ms	
	128 バイト	t_{P128_E}	—	1.3	13	ms	
消去時間*1、*2、*4	2K バイト	t_{E2K_E}	—	70	280	ms	
ブランクチェック時間*1、*4	8 バイト	t_{BC8_E}	—	—	30	μs	
	2K バイト	t_{BC2K_E}	—	—	0.7	ms	
再書き込み/消去サイクル	N_{PEC_E}	30000*3	—	—	—	回	
書き込み中のサスペンド遅延時間	t_{SPD_E}	—	—	120	—	μs	図 25.27
消去中の 1 回目のサスペンド遅延時間 (サスペンド優先モード時)	t_{SESD1_E}	—	—	120	—	μs	
消去中の 2 回目のサスペンド遅延時間 (サスペンド優先モード時)	t_{SESD2_E}	—	—	1.7	—	ms	
消去中のサスペンド遅延時間 (消去優先モード時)	t_{SEED_E}	—	—	1.7	—	ms	
データ保持時間*4	t_{DRP_E}	15	—	—	—	年	

【注】 *1 書き込み、消去時間はデータに依存します。

*2 書き込み、消去時間にはデータ転送時間は含みません。

*3 書き換え後のすべての特性を保証する min 回数です (保証は 1~min 値の範囲です)。

*4 書き換えが min 値を含む仕様範囲内で行われたときの特性です。

25.6 その他の特性

25.6.1 外部発振停止検出条件

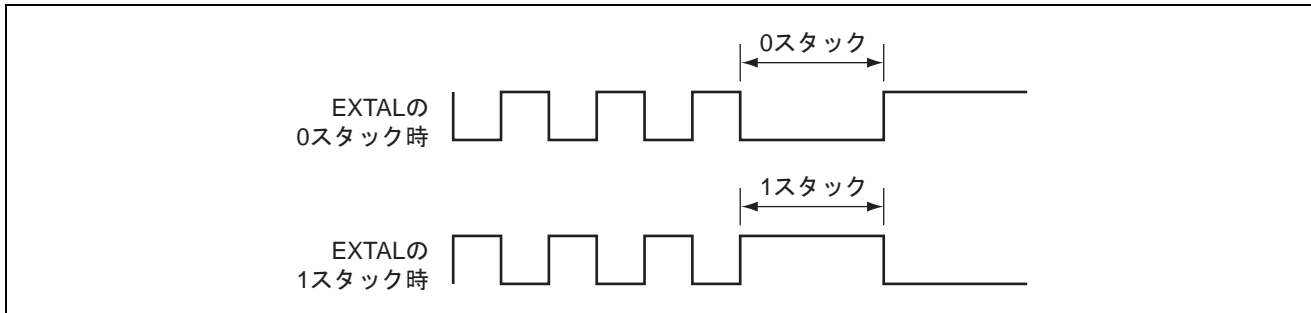


図 25.28 外部発振停止検出条件

表 25.11 外部発振停止検出条件

条件 : $V_{cc}=4.5\sim 5.5V$ 、 $AV_{cc0}=4.5\sim 5.5V$ 、 $AV_{cc1}=4.5\sim 5.5V$ 、 $V_{ss}=AV_{ss}=0V$ 、 $T_a=-40\sim +85^{\circ}C$

項目	測定条件	参照図
発振停止検出条件	0 または 1 スタック時	図 25.28

25.6.2 外部発振停止検出タイミング

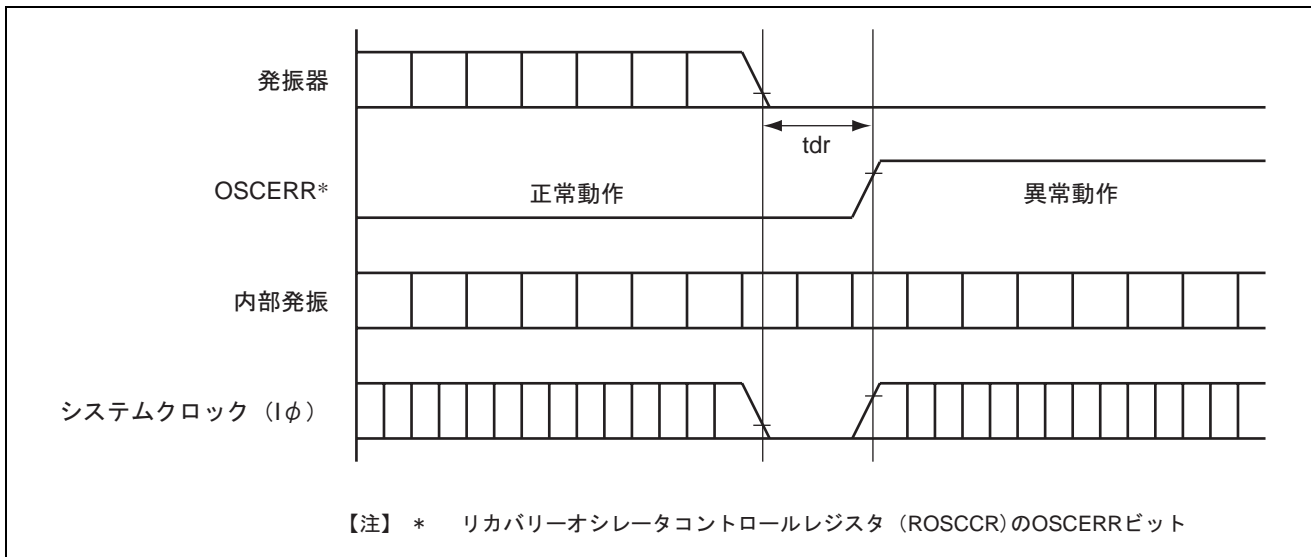


図 25.29 外部発振停止検出タイミング

表 25.12 外部発振停止検出回路の異常動作判定検出

条件 : $V_{CC}=4.5\sim 5.5V$ 、 $AV_{CC0}=4.5\sim 5.5V$ 、 $AV_{CC1}=4.5\sim 5.5V$ 、 $V_{SS}=AV_{SS}=0V$ 、 $T_a=-40\sim +85^{\circ}C$

	記号	min	typ	max	単位	参照図
検出時間	tdr	—	—	1.0	ms	図 25.29

25.6.3 内部発振周波数

表 25.13 内部発振周波数

条件 : $V_{CC}=4.5\sim 5.5V$ 、 $AV_{CC0}=4.5\sim 5.5V$ 、 $AV_{CC1}=4.5\sim 5.5V$ 、 $V_{SS}=AV_{SS}=0V$ 、 $T_a=-40^{\circ}C\sim 85^{\circ}C$

項目	記号	min	typ	max	単位	参照図
内部発振周波数	$f_{MAIN\phi}$	1.00	2.50	4.00	MHz	図 25.29

付録

A. 各処理状態におけるポートの状態

表 A.1 各処理状態におけるポートの状態

ポート名	MCU 動作モード	リセット	ソフトウェアスタンバイモード
ポート 1	全モード	HiZ	Keep
ポート 3	全モード	HiZ	Keep
ポート 4	全モード	HiZ	HiZ
ポート 5	全モード	HiZ	HiZ
ポート 6	全モード	HiZ	Keep
ポート A	全モード	HiZ	Keep
ポート D	全モード	HiZ	Keep
ポート H	全モード	HiZ	Keep
ポート J	全モード	HiZ	Keep
ポート K	全モード	HiZ	Keep

B. 型名一覧

製品分類	製品型名	マーク型名	パッケージ	コード
H8SX/1727S	R5F61727S	R5F61727S	QFP-100	PLQP0100KB-A (FP-100U)
H8SX/1725S	R5F61725S	R5F61725S		

C. 外形寸法図

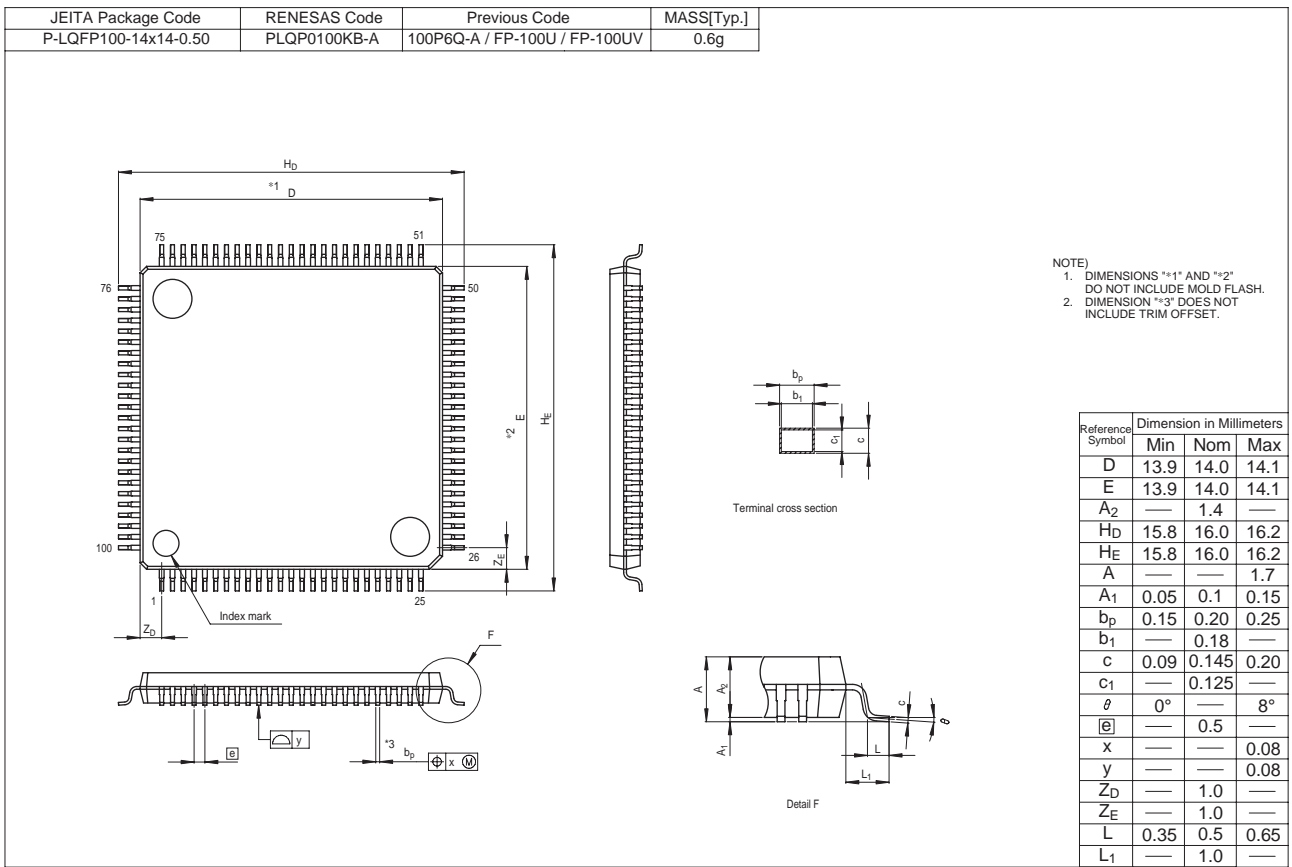
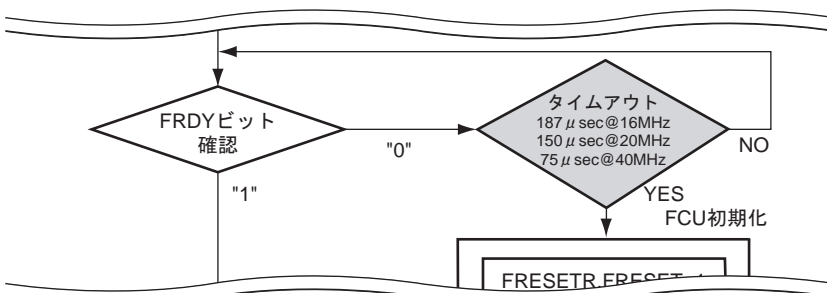
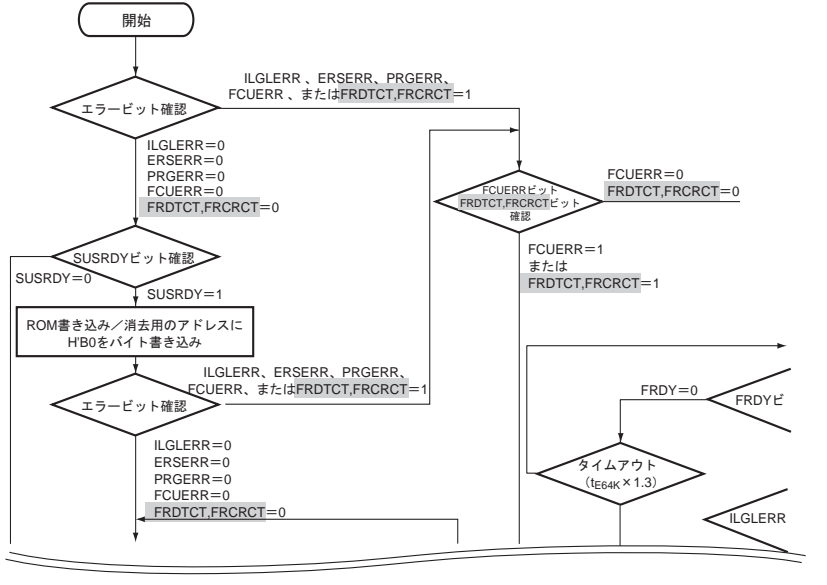


図 C.1 外形寸法図 (PLQP0100KB-A)

本版で修正または追加された箇所

項目	ページ	修正箇所
5.8.1 割り込みの発生とディスエーブルとの競合	5-35	追加 命令実行終了後にその割り込み例外処理を開始します。割り込みモード2の場合、命令実行終了後、EXRのI2~I0ビットで設定したCPUの割り込みマスクレベル以下の割り込み例外処理を実行する場合があります。ただし、その割り込みより優先順位の高い割り込み要求がある場合には優先順位の高い割り込み例外処理を実行し、その割り込みは無視されます。割り込み要因フラグを0にクリアする場合も同様です。割り込みモード2の場合、IPRのレジスタ書き換えで、割り込みをディスエーブルにする場合も同様です。TPUのTIERのTCIEVビットを0にクリアする場合の例を図5.7に示します。なお、割り込みをマスクした状態でイネーブルビット、または割り込み要因フラグを0にクリアまたは、IPRレジスタの書き換えをすれば、上記の競合は発生しません。本競合を回避するための割り込みマスク方法を図5.8に示します。
図5.8 割り込み競合の回避方法	5-36	図を追加
5.8.6 周辺モジュール割り込み要因フラグ	5-37	追加および修正 システムクロックに対して周辺モジュールのクロックの分周によらず、CPUによって周辺モジュールの割り込み要因フラグ、割り込みイネーブルビットをクリアするときは、周辺モジュールと同期を取るために、割り込み処理ルーチン内でフラグおよび、割り込みイネーブルレジスタをクリアした後にかならず当該レジスタをリードしてください。
表20.12 FCUのモード／状態と受け付け可能なコマンドの関係	20-61	追加および削除 表20.12ではエラー発生状況を表すビットにCMDLKビットを使用しています。(ただし、FRAMECCRレジスタのFRDCLEビットとFRCCLCビットに0が設定されている場合、CMDLKビットによるFIFEの割り込み要求は、発生しません。) ...コマンドロック状態の FRDY FRDYビットの値は、コマンドロック状態に遷移する前のFRDYビットの値が保持されます。

項目	ページ	修正箇所																																																																																																																																																																																																																																																																																																																
表 20.12FCU のモード /状態と受け付け可能 なコマンドの関係	20-61	追加 <table border="1" data-bbox="603 315 1433 1653"> <thead> <tr> <th rowspan="2">項 目</th> <th colspan="3">P/E ノーマル モード</th> <th colspan="8">ステータスリードモード</th> <th colspan="3">ロックビット リードモード</th> </tr> <tr> <th>書き込み サスペンド中</th> <th>消去サ スペンド中</th> <th>その 他の状 態</th> <th>書き込 み/消 去の 処理 中</th> <th>消去サ スペ ンド 中の 書き 込み 処理 中</th> <th>消去サ スペ ンド 中の 断 断 理 中</th> <th>書き込 み/消 去の 断 断 理 中</th> <th>ロッ クビ ット リ ード 2 処 理 中</th> <th>ロッ クビ ット リ ード 1 処 理 中</th> <th>消去サ スペ ンド 中</th> <th>消去サ スペ ンド 中</th> <th>コ マ ン ド ロ ック 状 態 (RDY=0)</th> <th>コ マ ン ド ロ ック 状 態 (RDY=1)</th> <th>そ の 他 の 状 態</th> <th>書き込 みサ スペ ンド 中</th> <th>消去サ スペ ンド 中</th> <th>その 他の 状 態</th> </tr> </thead> <tbody> <tr> <td>FSTATR0 レジスタの FRDY ビット</td> <td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td> </tr> <tr> <td>FSTATR0 レジスタの SUSRDY ビット</td> <td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> <tr> <td>FSTATR0 レジスタの ERSSPD ビット</td> <td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0/1</td><td>0/1</td><td>0</td><td>1</td><td>0/1</td><td>0/1</td><td>0/1</td><td>0</td><td>0</td><td>1</td><td>0</td> </tr> <tr> <td>FSTATR0 レジスタの PRGSPD ビット</td> <td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0/1</td><td>0/1</td><td>1</td><td>0</td><td>0/1</td><td>0/1</td><td>0/1</td><td>0</td><td>1</td><td>0</td><td>0</td> </tr> <tr> <td>FASTAT レジスタの CMDLK ビット</td> <td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> <tr> <td>ノーマルモード移行</td> <td>○</td><td>○</td><td>○</td><td>×</td><td>×</td><td>×</td><td>×</td><td>○</td><td>○</td><td>×</td><td>×</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td> </tr> <tr> <td>ステータスリード モード移行</td> <td>○</td><td>○</td><td>○</td><td>×</td><td>×</td><td>×</td><td>×</td><td>○</td><td>○</td><td>×</td><td>×</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td> </tr> <tr> <td>ロックビットリード モード移行 (ロックビットリード1)</td> <td>○</td><td>○</td><td>○</td><td>×</td><td>×</td><td>×</td><td>×</td><td>○</td><td>○</td><td>×</td><td>×</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td> </tr> <tr> <td>フラッシュロック 通知</td> <td>×</td><td>×</td><td>○</td><td>×</td><td>×</td><td>×</td><td>×</td><td>×</td><td>×</td><td>×</td><td>×</td><td>○</td><td>×</td><td>×</td><td>×</td><td>○</td> </tr> <tr> <td>プログラム</td> <td>×</td><td>△</td><td>○</td><td>×</td><td>×</td><td>×</td><td>×</td><td>×</td><td>△</td><td>×</td><td>×</td><td>○</td><td>×</td><td>△</td><td>○</td><td>○</td> </tr> <tr> <td>ブロックイレーズ</td> <td>×</td><td>×</td><td>○</td><td>×</td><td>×</td><td>×</td><td>×</td><td>×</td><td>×</td><td>×</td><td>×</td><td>○</td><td>×</td><td>×</td><td>×</td><td>○</td> </tr> <tr> <td>P/E サスペンド</td> <td>×</td><td>×</td><td>×</td><td>○</td><td>×</td><td>×</td><td>×</td><td>×</td><td>×</td><td>×</td><td>×</td><td>×</td><td>×</td><td>×</td><td>×</td><td>×</td> </tr> <tr> <td>P/E レジューム</td> <td>○</td><td>○</td><td>×</td><td>×</td><td>×</td><td>×</td><td>×</td><td>○</td><td>○</td><td>×</td><td>×</td><td>×</td><td>○</td><td>○</td><td>○</td><td>×</td> </tr> <tr> <td>ステータスレジスタ クリア</td> <td>○</td><td>○</td><td>○</td><td>×</td><td>×</td><td>×</td><td>×</td><td>○</td><td>○</td><td>×</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td> </tr> <tr> <td>ロックビットリード2</td> <td>○</td><td>○</td><td>○</td><td>×</td><td>×</td><td>×</td><td>×</td><td>○</td><td>○</td><td>×</td><td>×</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td> </tr> <tr> <td>ロックビットプログラム</td> <td>×</td><td>△</td><td>○</td><td>×</td><td>×</td><td>×</td><td>×</td><td>×</td><td>△</td><td>×</td><td>×</td><td>○</td><td>×</td><td>△</td><td>○</td><td>○</td> </tr> </tbody> </table>	項 目	P/E ノーマル モード			ステータスリードモード								ロックビット リードモード			書き込み サスペンド中	消去サ スペンド中	その 他の状 態	書き込 み/消 去の 処理 中	消去サ スペ ンド 中の 書き 込み 処理 中	消去サ スペ ンド 中の 断 断 理 中	書き込 み/消 去の 断 断 理 中	ロッ クビ ット リ ード 2 処 理 中	ロッ クビ ット リ ード 1 処 理 中	消去サ スペ ンド 中	消去サ スペ ンド 中	コ マ ン ド ロ ック 状 態 (RDY=0)	コ マ ン ド ロ ック 状 態 (RDY=1)	そ の 他 の 状 態	書き込 みサ スペ ンド 中	消去サ スペ ンド 中	その 他の 状 態	FSTATR0 レジスタの FRDY ビット	1	1	1	0	0	0	0	0	1	1	0	1	1	1	1	1	FSTATR0 レジスタの SUSRDY ビット	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	FSTATR0 レジスタの ERSSPD ビット	0	1	0	0	1	0/1	0/1	0	1	0/1	0/1	0/1	0	0	1	0	FSTATR0 レジスタの PRGSPD ビット	1	0	0	0	0	0/1	0/1	1	0	0/1	0/1	0/1	0	1	0	0	FASTAT レジスタの CMDLK ビット	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	ノーマルモード移行	○	○	○	×	×	×	×	○	○	×	×	○	○	○	○	○	ステータスリード モード移行	○	○	○	×	×	×	×	○	○	×	×	○	○	○	○	○	ロックビットリード モード移行 (ロックビットリード1)	○	○	○	×	×	×	×	○	○	×	×	○	○	○	○	○	フラッシュロック 通知	×	×	○	×	×	×	×	×	×	×	×	○	×	×	×	○	プログラム	×	△	○	×	×	×	×	×	△	×	×	○	×	△	○	○	ブロックイレーズ	×	×	○	×	×	×	×	×	×	×	×	○	×	×	×	○	P/E サスペンド	×	×	×	○	×	×	×	×	×	×	×	×	×	×	×	×	P/E レジューム	○	○	×	×	×	×	×	○	○	×	×	×	○	○	○	×	ステータスレジスタ クリア	○	○	○	×	×	×	×	○	○	×	○	○	○	○	○	○	ロックビットリード2	○	○	○	×	×	×	×	○	○	×	×	○	○	○	○	○	ロックビットプログラム	×	△	○	×	×	×	×	×	△	×	×	○	×	△	○	○
項 目	P/E ノーマル モード			ステータスリードモード								ロックビット リードモード																																																																																																																																																																																																																																																																																																						
	書き込み サスペンド中	消去サ スペンド中	その 他の状 態	書き込 み/消 去の 処理 中	消去サ スペ ンド 中の 書き 込み 処理 中	消去サ スペ ンド 中の 断 断 理 中	書き込 み/消 去の 断 断 理 中	ロッ クビ ット リ ード 2 処 理 中	ロッ クビ ット リ ード 1 処 理 中	消去サ スペ ンド 中	消去サ スペ ンド 中	コ マ ン ド ロ ック 状 態 (RDY=0)	コ マ ン ド ロ ック 状 態 (RDY=1)	そ の 他 の 状 態	書き込 みサ スペ ンド 中	消去サ スペ ンド 中	その 他の 状 態																																																																																																																																																																																																																																																																																																	
FSTATR0 レジスタの FRDY ビット	1	1	1	0	0	0	0	0	1	1	0	1	1	1	1	1																																																																																																																																																																																																																																																																																																		
FSTATR0 レジスタの SUSRDY ビット	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																		
FSTATR0 レジスタの ERSSPD ビット	0	1	0	0	1	0/1	0/1	0	1	0/1	0/1	0/1	0	0	1	0																																																																																																																																																																																																																																																																																																		
FSTATR0 レジスタの PRGSPD ビット	1	0	0	0	0	0/1	0/1	1	0	0/1	0/1	0/1	0	1	0	0																																																																																																																																																																																																																																																																																																		
FASTAT レジスタの CMDLK ビット	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0																																																																																																																																																																																																																																																																																																		
ノーマルモード移行	○	○	○	×	×	×	×	○	○	×	×	○	○	○	○	○																																																																																																																																																																																																																																																																																																		
ステータスリード モード移行	○	○	○	×	×	×	×	○	○	×	×	○	○	○	○	○																																																																																																																																																																																																																																																																																																		
ロックビットリード モード移行 (ロックビットリード1)	○	○	○	×	×	×	×	○	○	×	×	○	○	○	○	○																																																																																																																																																																																																																																																																																																		
フラッシュロック 通知	×	×	○	×	×	×	×	×	×	×	×	○	×	×	×	○																																																																																																																																																																																																																																																																																																		
プログラム	×	△	○	×	×	×	×	×	△	×	×	○	×	△	○	○																																																																																																																																																																																																																																																																																																		
ブロックイレーズ	×	×	○	×	×	×	×	×	×	×	×	○	×	×	×	○																																																																																																																																																																																																																																																																																																		
P/E サスペンド	×	×	×	○	×	×	×	×	×	×	×	×	×	×	×	×																																																																																																																																																																																																																																																																																																		
P/E レジューム	○	○	×	×	×	×	×	○	○	×	×	×	○	○	○	×																																																																																																																																																																																																																																																																																																		
ステータスレジスタ クリア	○	○	○	×	×	×	×	○	○	×	○	○	○	○	○	○																																																																																																																																																																																																																																																																																																		
ロックビットリード2	○	○	○	×	×	×	×	○	○	×	×	○	○	○	○	○																																																																																																																																																																																																																																																																																																		
ロックビットプログラム	×	△	○	×	×	×	×	×	△	×	×	○	×	△	○	○																																																																																																																																																																																																																																																																																																		
20.6.3 FCU コマンド 使用方法 (5) フラッシュクロ ック通知コマンドの使用 方法	20-67	削除および追加 フラッシュメモリへのデータ書き込み/消去前に使用しているフラッシュクロ ックの設定を行います。最初に使用しているフラッシュクロックの周波数を FCKAR レジスタに設定します。設定可能な周波数の範囲は 8~40MHz です。 この範囲に設 定しなかった場合には、FCU はエラーを検出しコマンドロック状態になります (「20.8.3 エラープロテクト」を参照)。 この範囲以外は、設定しないでくださ い。																																																																																																																																																																																																																																																																																																																

項目	ページ	修正箇所																		
図 20.19 フラッシュク ロック通知コマンドの 使用方法	20-68	修正  <p>The flowchart starts with a decision diamond labeled 'FRDYビット確認' (FRDY bit check). If the bit is '1', it proceeds to a 'FRESETR, ERASE' block. If the bit is '0', it goes to a 'タイムアウト' (Timeout) diamond with the following specifications: 187 μsec@16MHz, 150 μsec@20MHz, and 75 μsec@40MHz. If the timeout occurs (NO), it loops back to the FRDY bit check. If it does not (YES), it proceeds to 'FCU初期化' (FCU initialization).</p>																		
図 20.22 書き込み/消 去のサスペンド方法	20-73	修正  <p>The flowchart begins with '開始' (Start) leading to an 'エラービット確認' (Error bit check) diamond. The conditions for this check are: ILGLERR=0, ERSERR=0, PRGERR=0, FCUERR=0, and FRDTCT, FRCRCT=0. If an error is detected (ILGLERR, ERSERR, PRGERR, FCUERR, or FRDTCT, FRCRCT=1), it goes to another 'エラービット確認' diamond. If no error is detected, it proceeds to 'SUSRDYビット確認' (SUSRDY bit check). If SUSRDY=0, it goes to 'ROM書き込み/消去用のアドレスに HB0をバイト書き込み' (Write HB0 to ROM address for write/erase). If SUSRDY=1, it goes to the second error check diamond. After writing HB0, it checks for errors again. If no error, it goes to a 'タイムアウト' (Timeout) diamond with conditions: FRDY=0 and (tE64K × 1.3). If the timeout occurs, it goes to 'ILGLERR'. If not, it goes to a 'FCUERRビット FRDTCT, FRCRCTビット確認' (FCUERR bit FRDTCT, FRCRCT bit check) diamond. If FCUERR=0 and FRDTCT, FRCRCT=0, it exits. If FCUERR=1 or FRDTCT, FRCRCT=1, it loops back to the error bit check after the error bit check diamond.</p>																		
表 20.14 エラープロテ クト一覧	20-85	修正 <table border="1" data-bbox="670 1321 1356 1433"> <thead> <tr> <th>分類</th> <th>内容</th> <th>ILGLERR</th> <th>ERSERR</th> <th>PRGERR</th> <th>FCUERR</th> <th>FRDTCT</th> <th>FRCRCT</th> <th>ROMAE</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table>	分類	内容	ILGLERR	ERSERR	PRGERR	FCUERR	FRDTCT	FRCRCT	ROMAE									
分類	内容	ILGLERR	ERSERR	PRGERR	FCUERR	FRDTCT	FRCRCT	ROMAE												

項目	ページ	修正箇所																																																																																																																																																																																																																																																																														
表 21.9 FCU のモード ／状態と受け付け可能なコマンドの関係	21-29	<p>追加</p> <table border="1"> <thead> <tr> <th rowspan="2">項目</th> <th colspan="3">P/E ノーマルモード</th> <th colspan="8">ステータスリードモード</th> <th colspan="3">ロックビットリードモード</th> </tr> <tr> <th>書き込みサスペンド中</th> <th>消去サスペンド中</th> <th>その他の状態</th> <th>書き込み／消去の処理中</th> <th>書き込みサスペンド中の書き込み処理中</th> <th>書き込み／消去の中断</th> <th>ブランクチェック処理中</th> <th>書き込みサスペンド中</th> <th>消去サスペンド中</th> <th>(RDY=0) コマンドロック状態</th> <th>(RDY=1) コマンドロック状態</th> <th>その他の状態</th> <th>書き込みサスペンド中</th> <th>消去サスペンド中</th> <th>その他の状態</th> </tr> </thead> <tbody> <tr> <td>FSTATR0 レジスタの FRDY ビット</td> <td>1</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> </tr> <tr> <td>FSTATR0 レジスタの SUSRDY ビット</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>FSTATR0 レジスタの ERSSPD ビット</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>1</td> <td>0/1</td> <td>0/1</td> <td>0</td> <td>1</td> <td>0/1</td> <td>0/1</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>FSTATR0 レジスタの PRGSPD ビット</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0/1</td> <td>0/1</td> <td>1</td> <td>0</td> <td>0/1</td> <td>0/1</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>FASTAT レジスタの CMDLK ビット</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>ノーマルモード移行</td> <td>○</td> <td>○</td> <td>○</td> <td>×</td> <td>×</td> <td>×</td> <td>×</td> <td>○</td> <td>○</td> <td>×</td> <td>×</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ステータスリードモード移行</td> <td>○</td> <td>○</td> <td>○</td> <td>×</td> <td>×</td> <td>×</td> <td>×</td> <td>○</td> <td>○</td> <td>×</td> <td>×</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ロックビットリードモード移行 (ロックビットリード1)</td> <td>○</td> <td>○</td> <td>○</td> <td>×</td> <td>×</td> <td>×</td> <td>×</td> <td>○</td> <td>○</td> <td>×</td> <td>×</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>フラッシュロック通知</td> <td>×</td> <td>×</td> <td>○</td> <td>×</td> <td>×</td> <td>×</td> <td>×</td> <td>×</td> <td>×</td> <td>×</td> <td>×</td> <td>○</td> <td>×</td> <td>×</td> <td>○</td> </tr> <tr> <td>プログラム</td> <td>×</td> <td>△</td> <td>○</td> <td>×</td> <td>×</td> <td>×</td> <td>×</td> <td>×</td> <td>△</td> <td>×</td> <td>×</td> <td>○</td> <td>×</td> <td>△</td> <td>○</td> </tr> <tr> <td>ブロックイレーズ</td> <td>×</td> <td>×</td> <td>○</td> <td>×</td> <td>×</td> <td>×</td> <td>×</td> <td>×</td> <td>×</td> <td>×</td> <td>×</td> <td>○</td> <td>×</td> <td>×</td> <td>○</td> </tr> <tr> <td>P/E サスペンド</td> <td>×</td> <td>×</td> <td>×</td> <td>○</td> <td>×</td> <td>×</td> <td>×</td> <td>×</td> <td>×</td> <td>×</td> <td>×</td> <td>×</td> <td>×</td> <td>×</td> <td>×</td> </tr> <tr> <td>P/E レジューム</td> <td>○</td> <td>○</td> <td>×</td> <td>×</td> <td>×</td> <td>×</td> <td>×</td> <td>○</td> <td>○</td> <td>×</td> <td>×</td> <td>×</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ステータスレジスタクリア</td> <td>○</td> <td>○</td> <td>○</td> <td>×</td> <td>×</td> <td>×</td> <td>×</td> <td>○</td> <td>○</td> <td>×</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ブランクチェック</td> <td>○</td> <td>○</td> <td>○</td> <td>×</td> <td>×</td> <td>×</td> <td>×</td> <td>○</td> <td>○</td> <td>×</td> <td>×</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> </tr> </tbody> </table>	項目	P/E ノーマルモード			ステータスリードモード								ロックビットリードモード			書き込みサスペンド中	消去サスペンド中	その他の状態	書き込み／消去の処理中	書き込みサスペンド中の書き込み処理中	書き込み／消去の中断	ブランクチェック処理中	書き込みサスペンド中	消去サスペンド中	(RDY=0) コマンドロック状態	(RDY=1) コマンドロック状態	その他の状態	書き込みサスペンド中	消去サスペンド中	その他の状態	FSTATR0 レジスタの FRDY ビット	1	1	1	0	0	0	0	1	1	0	1	1	1	1	1	FSTATR0 レジスタの SUSRDY ビット	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	FSTATR0 レジスタの ERSSPD ビット	0	1	0	0	1	0/1	0/1	0	1	0/1	0/1	0	0	1	0	FSTATR0 レジスタの PRGSPD ビット	1	0	0	0	0	0/1	0/1	1	0	0/1	0/1	0	1	0	0	FASTAT レジスタの CMDLK ビット	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	ノーマルモード移行	○	○	○	×	×	×	×	○	○	×	×	○	○	○	○	ステータスリードモード移行	○	○	○	×	×	×	×	○	○	×	×	○	○	○	○	ロックビットリードモード移行 (ロックビットリード1)	○	○	○	×	×	×	×	○	○	×	×	○	○	○	○	フラッシュロック通知	×	×	○	×	×	×	×	×	×	×	×	○	×	×	○	プログラム	×	△	○	×	×	×	×	×	△	×	×	○	×	△	○	ブロックイレーズ	×	×	○	×	×	×	×	×	×	×	×	○	×	×	○	P/E サスペンド	×	×	×	○	×	×	×	×	×	×	×	×	×	×	×	P/E レジューム	○	○	×	×	×	×	×	○	○	×	×	×	○	○	×	ステータスレジスタクリア	○	○	○	×	×	×	×	○	○	×	○	○	○	○	○	ブランクチェック	○	○	○	×	×	×	×	○	○	×	×	○	○	○	○
項目	P/E ノーマルモード			ステータスリードモード								ロックビットリードモード																																																																																																																																																																																																																																																																				
	書き込みサスペンド中	消去サスペンド中	その他の状態	書き込み／消去の処理中	書き込みサスペンド中の書き込み処理中	書き込み／消去の中断	ブランクチェック処理中	書き込みサスペンド中	消去サスペンド中	(RDY=0) コマンドロック状態	(RDY=1) コマンドロック状態	その他の状態	書き込みサスペンド中	消去サスペンド中	その他の状態																																																																																																																																																																																																																																																																	
FSTATR0 レジスタの FRDY ビット	1	1	1	0	0	0	0	1	1	0	1	1	1	1	1																																																																																																																																																																																																																																																																	
FSTATR0 レジスタの SUSRDY ビット	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0																																																																																																																																																																																																																																																																	
FSTATR0 レジスタの ERSSPD ビット	0	1	0	0	1	0/1	0/1	0	1	0/1	0/1	0	0	1	0																																																																																																																																																																																																																																																																	
FSTATR0 レジスタの PRGSPD ビット	1	0	0	0	0	0/1	0/1	1	0	0/1	0/1	0	1	0	0																																																																																																																																																																																																																																																																	
FASTAT レジスタの CMDLK ビット	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0																																																																																																																																																																																																																																																																	
ノーマルモード移行	○	○	○	×	×	×	×	○	○	×	×	○	○	○	○																																																																																																																																																																																																																																																																	
ステータスリードモード移行	○	○	○	×	×	×	×	○	○	×	×	○	○	○	○																																																																																																																																																																																																																																																																	
ロックビットリードモード移行 (ロックビットリード1)	○	○	○	×	×	×	×	○	○	×	×	○	○	○	○																																																																																																																																																																																																																																																																	
フラッシュロック通知	×	×	○	×	×	×	×	×	×	×	×	○	×	×	○																																																																																																																																																																																																																																																																	
プログラム	×	△	○	×	×	×	×	×	△	×	×	○	×	△	○																																																																																																																																																																																																																																																																	
ブロックイレーズ	×	×	○	×	×	×	×	×	×	×	×	○	×	×	○																																																																																																																																																																																																																																																																	
P/E サスペンド	×	×	×	○	×	×	×	×	×	×	×	×	×	×	×																																																																																																																																																																																																																																																																	
P/E レジューム	○	○	×	×	×	×	×	○	○	×	×	×	○	○	×																																																																																																																																																																																																																																																																	
ステータスレジスタクリア	○	○	○	×	×	×	×	○	○	×	○	○	○	○	○																																																																																																																																																																																																																																																																	
ブランクチェック	○	○	○	×	×	×	×	○	○	×	×	○	○	○	○																																																																																																																																																																																																																																																																	
表 21.11 製品情報マトリックス格納データ (H8SX/1727S の場合)	21-36	<p>修正および追加</p> <table border="1"> <thead> <tr> <th>情報</th> <th>アドレス</th> <th>データ例</th> </tr> </thead> <tbody> <tr> <td>デバイス名</td> <td>H'E0,0000~H'E0,0008</td> <td>H'523546363137323753=R5F61727S</td> </tr> </tbody> </table>	情報	アドレス	データ例	デバイス名	H'E0,0000~H'E0,0008	H'523546363137323753=R5F61727S																																																																																																																																																																																																																																																																								
情報	アドレス	データ例																																																																																																																																																																																																																																																																														
デバイス名	H'E0,0000~H'E0,0008	H'523546363137323753=R5F61727S																																																																																																																																																																																																																																																																														
表 21.12 製品情報マトリックス格納データ (H8SX/1725S の場合)	21-36	<p>修正および追加</p> <table border="1"> <thead> <tr> <th>情報</th> <th>アドレス</th> <th>データ例</th> </tr> </thead> <tbody> <tr> <td>デバイス名</td> <td>H'E0,0000~H'E0,0008</td> <td>H'523546363137323553=R5F61725S</td> </tr> </tbody> </table>	情報	アドレス	データ例	デバイス名	H'E0,0000~H'E0,0008	H'523546363137323553=R5F61725S																																																																																																																																																																																																																																																																								
情報	アドレス	データ例																																																																																																																																																																																																																																																																														
デバイス名	H'E0,0000~H'E0,0008	H'523546363137323553=R5F61725S																																																																																																																																																																																																																																																																														
22.2.1 水晶発振子を接続する方法	22-10	<p>削除</p> <p>水晶発振子を接続する場合の接続例を図 22.2 に示します。ダンピング抵抗 Rd は、表 22.2 に示すものを使用してください。また、水晶発振子は、AT カット 並列共振形を使用してください。</p>																																																																																																																																																																																																																																																																														
図 22.3 水晶発振子の等価回路	22-10	<p>削除</p> <p style="text-align: right;">ATカット 並列共振形</p>																																																																																																																																																																																																																																																																														

項目	ページ	修正箇所																																																																																																																																																																																																																																																																																									
23.8 φクロック出力制御	23-20	削除 表 23.3、 表 23.4 に各処理状態におけるφ端子の状態を示します。																																																																																																																																																																																																																																																																																									
24.2 レジスタビット一覧	24-44、24-45	修正 <table border="1"> <thead> <tr> <th>レジスタ 略称</th> <th>ビット 31/23/15/7</th> <th>ビット 30/22/14/6</th> <th>ビット 29/21/13/5</th> <th>ビット 28/20/12/4</th> <th>ビット 27/19/11/3</th> <th>ビット 26/18/10/2</th> <th>ビット 25/17/9/1</th> <th>ビット 24/16/8/0</th> </tr> </thead> <tbody> <tr> <td rowspan="2">TXPR1_0</td> <td>TXPR1_15</td> <td>TXPR1_14</td> <td>TXPR1_13</td> <td>TXPR1_12</td> <td>TXPR1_11</td> <td>TXPR1_10</td> <td>TXPR1_9</td> <td>TXPR1_8</td> </tr> <tr> <td>TXPR1_7</td> <td>TXPR1_6</td> <td>TXPR1_5</td> <td>TXPR1_4</td> <td>TXPR1_3</td> <td>TXPR1_2</td> <td>TXPR1_1</td> <td>TXPR1_0</td> </tr> <tr> <td rowspan="2">TXPR0_0</td> <td>TXPR0_15</td> <td>TXPR0_14</td> <td>TXPR0_13</td> <td>TXPR0_12</td> <td>TXPR0_11</td> <td>TXPR0_10</td> <td>TXPR0_9</td> <td>TXPR0_8</td> </tr> <tr> <td>TXPR0_7</td> <td>TXPR0_6</td> <td>TXPR0_5</td> <td>TXPR0_4</td> <td>TXPR0_3</td> <td>TXPR0_2</td> <td>TXPR0_1</td> <td>—</td> </tr> <tr> <td rowspan="2">TXCR1_0</td> <td>TXCR1_15</td> <td>TXCR1_14</td> <td>TXCR1_13</td> <td>TXCR1_12</td> <td>TXCR1_11</td> <td>TXCR1_10</td> <td>TXCR1_9</td> <td>TXCR1_8</td> </tr> <tr> <td>TXCR1_7</td> <td>TXCR1_6</td> <td>TXCR1_5</td> <td>TXCR1_4</td> <td>TXCR1_3</td> <td>TXCR1_2</td> <td>TXCR1_1</td> <td>TXCR1_0</td> </tr> <tr> <td rowspan="2">TXCR0_0</td> <td>TXCR0_15</td> <td>TXCR0_14</td> <td>TXCR0_13</td> <td>TXCR0_12</td> <td>TXCR0_11</td> <td>TXCR0_10</td> <td>TXCR0_9</td> <td>TXCR0_8</td> </tr> <tr> <td>TXCR0_7</td> <td>TXCR0_6</td> <td>TXCR0_5</td> <td>TXCR0_4</td> <td>TXCR0_3</td> <td>TXCR0_2</td> <td>TXCR0_1</td> <td>—</td> </tr> <tr> <td rowspan="2">TXACK1_0</td> <td>TXACK1_15</td> <td>TXACK1_14</td> <td>TXACK1_13</td> <td>TXACK1_12</td> <td>TXACK1_11</td> <td>TXACK1_10</td> <td>TXACK1_9</td> <td>TXACK1_8</td> </tr> <tr> <td>TXACK1_7</td> <td>TXACK1_6</td> <td>TXACK1_5</td> <td>TXACK1_4</td> <td>TXACK1_3</td> <td>TXACK1_2</td> <td>TXACK1_1</td> <td>TXACK1_0</td> </tr> <tr> <td rowspan="2">TXACK0_0</td> <td>TXACK0_15</td> <td>TXACK0_14</td> <td>TXACK0_13</td> <td>TXACK0_12</td> <td>TXACK0_11</td> <td>TXACK0_10</td> <td>TXACK0_9</td> <td>TXACK0_8</td> </tr> <tr> <td>TXACK0_7</td> <td>TXACK0_6</td> <td>TXACK0_5</td> <td>TXACK0_4</td> <td>TXACK0_3</td> <td>TXACK0_2</td> <td>TXACK0_1</td> <td>—</td> </tr> <tr> <td rowspan="2">ABACK1_0</td> <td>ABACK1_15</td> <td>ABACK1_14</td> <td>ABACK1_13</td> <td>ABACK1_12</td> <td>ABACK1_11</td> <td>ABACK1_10</td> <td>ABACK1_9</td> <td>ABACK1_8</td> </tr> <tr> <td>ABACK1_7</td> <td>ABACK1_6</td> <td>ABACK1_5</td> <td>ABACK1_4</td> <td>ABACK1_3</td> <td>ABACK1_2</td> <td>ABACK1_1</td> <td>ABACK1_0</td> </tr> <tr> <td rowspan="2">ABACK0_0</td> <td>ABACK0_15</td> <td>ABACK0_14</td> <td>ABACK0_13</td> <td>ABACK0_12</td> <td>ABACK0_11</td> <td>ABACK0_10</td> <td>ABACK0_9</td> <td>ABACK0_8</td> </tr> <tr> <td>ABACK0_7</td> <td>ABACK0_6</td> <td>ABACK0_5</td> <td>ABACK0_4</td> <td>ABACK0_3</td> <td>ABACK0_2</td> <td>ABACK0_1</td> <td>—</td> </tr> <tr> <td rowspan="2">RXPR1_0</td> <td>RXPR1_15</td> <td>RXPR1_14</td> <td>RXPR1_13</td> <td>RXPR1_12</td> <td>RXPR1_11</td> <td>RXPR1_10</td> <td>RXPR1_9</td> <td>RXPR1_8</td> </tr> <tr> <td>RXPR1_7</td> <td>RXPR1_6</td> <td>RXPR1_5</td> <td>RXPR1_4</td> <td>RXPR1_3</td> <td>RXPR1_2</td> <td>RXPR1_1</td> <td>RXPR1_0</td> </tr> <tr> <td rowspan="2">RXPR0_0</td> <td>RXPR0_15</td> <td>RXPR0_14</td> <td>RXPR0_13</td> <td>RXPR0_12</td> <td>RXPR0_11</td> <td>RXPR0_10</td> <td>RXPR0_9</td> <td>RXPR0_8</td> </tr> <tr> <td>RXPR0_7</td> <td>RXPR0_6</td> <td>RXPR0_5</td> <td>RXPR0_4</td> <td>RXPR0_3</td> <td>RXPR0_2</td> <td>RXPR0_1</td> <td>RXPR0_0</td> </tr> <tr> <td rowspan="2">RFPR1_0</td> <td>RFPR1_15</td> <td>RFPR1_14</td> <td>RFPR1_13</td> <td>RFPR1_12</td> <td>RFPR1_11</td> <td>RFPR1_10</td> <td>RFPR1_9</td> <td>RFPR1_8</td> </tr> <tr> <td>RFPR1_7</td> <td>RFPR1_6</td> <td>RFPR1_5</td> <td>RFPR1_4</td> <td>RFPR1_3</td> <td>RFPR1_2</td> <td>RFPR1_1</td> <td>RFPR1_0</td> </tr> <tr> <td rowspan="2">RFPR0_0</td> <td>RFPR0_15</td> <td>RFPR0_14</td> <td>RFPR0_13</td> <td>RFPR0_12</td> <td>RFPR0_11</td> <td>RFPR0_10</td> <td>RFPR0_9</td> <td>RFPR0_8</td> </tr> <tr> <td>RFPR0_7</td> <td>RFPR0_6</td> <td>RFPR0_5</td> <td>RFPR0_4</td> <td>RFPR0_3</td> <td>RFPR0_2</td> <td>RFPR0_1</td> <td>RFPR0_0</td> </tr> <tr> <td rowspan="2">MBIMR1_0</td> <td>MBIMR1_15</td> <td>MBIMR1_14</td> <td>MBIMR1_13</td> <td>MBIMR1_12</td> <td>MBIMR1_11</td> <td>MBIMR1_10</td> <td>MBIMR1_9</td> <td>MBIMR1_8</td> </tr> <tr> <td>MBIMR1_7</td> <td>MBIMR1_6</td> <td>MBIMR1_5</td> <td>MBIMR1_4</td> <td>MBIMR1_3</td> <td>MBIMR1_2</td> <td>MBIMR1_1</td> <td>MBIMR1_0</td> </tr> <tr> <td rowspan="2">MBIMR0_0</td> <td>MBIMR0_15</td> <td>MBIMR0_14</td> <td>MBIMR0_13</td> <td>MBIMR0_12</td> <td>MBIMR0_11</td> <td>MBIMR0_10</td> <td>MBIMR0_9</td> <td>MBIMR0_8</td> </tr> <tr> <td>MBIMR0_7</td> <td>MBIMR0_6</td> <td>MBIMR0_5</td> <td>MBIMR0_4</td> <td>MBIMR0_3</td> <td>MBIMR0_2</td> <td>MBIMR0_1</td> <td>MBIMR0_0</td> </tr> <tr> <td rowspan="2">UMSR1_0</td> <td>UMSR1_15</td> <td>UMSR1_14</td> <td>UMSR1_13</td> <td>UMSR1_12</td> <td>UMSR1_11</td> <td>UMSR1_10</td> <td>UMSR1_9</td> <td>UMSR1_8</td> </tr> <tr> <td>UMSR1_7</td> <td>UMSR1_6</td> <td>UMSR1_5</td> <td>UMSR1_4</td> <td>UMSR1_3</td> <td>UMSR1_2</td> <td>UMSR1_1</td> <td>UMSR1_0</td> </tr> <tr> <td rowspan="2">UMSR0_0</td> <td>UMSR0_15</td> <td>UMSR0_14</td> <td>UMSR0_13</td> <td>UMSR0_12</td> <td>UMSR0_11</td> <td>UMSR0_10</td> <td>UMSR0_9</td> <td>UMSR0_8</td> </tr> <tr> <td>UMSR0_7</td> <td>UMSR0_6</td> <td>UMSR0_5</td> <td>UMSR0_4</td> <td>UMSR0_3</td> <td>UMSR0_2</td> <td>UMSR0_1</td> <td>UMSR0_0</td> </tr> </tbody> </table>	レジスタ 略称	ビット 31/23/15/7	ビット 30/22/14/6	ビット 29/21/13/5	ビット 28/20/12/4	ビット 27/19/11/3	ビット 26/18/10/2	ビット 25/17/9/1	ビット 24/16/8/0	TXPR1_0	TXPR1_15	TXPR1_14	TXPR1_13	TXPR1_12	TXPR1_11	TXPR1_10	TXPR1_9	TXPR1_8	TXPR1_7	TXPR1_6	TXPR1_5	TXPR1_4	TXPR1_3	TXPR1_2	TXPR1_1	TXPR1_0	TXPR0_0	TXPR0_15	TXPR0_14	TXPR0_13	TXPR0_12	TXPR0_11	TXPR0_10	TXPR0_9	TXPR0_8	TXPR0_7	TXPR0_6	TXPR0_5	TXPR0_4	TXPR0_3	TXPR0_2	TXPR0_1	—	TXCR1_0	TXCR1_15	TXCR1_14	TXCR1_13	TXCR1_12	TXCR1_11	TXCR1_10	TXCR1_9	TXCR1_8	TXCR1_7	TXCR1_6	TXCR1_5	TXCR1_4	TXCR1_3	TXCR1_2	TXCR1_1	TXCR1_0	TXCR0_0	TXCR0_15	TXCR0_14	TXCR0_13	TXCR0_12	TXCR0_11	TXCR0_10	TXCR0_9	TXCR0_8	TXCR0_7	TXCR0_6	TXCR0_5	TXCR0_4	TXCR0_3	TXCR0_2	TXCR0_1	—	TXACK1_0	TXACK1_15	TXACK1_14	TXACK1_13	TXACK1_12	TXACK1_11	TXACK1_10	TXACK1_9	TXACK1_8	TXACK1_7	TXACK1_6	TXACK1_5	TXACK1_4	TXACK1_3	TXACK1_2	TXACK1_1	TXACK1_0	TXACK0_0	TXACK0_15	TXACK0_14	TXACK0_13	TXACK0_12	TXACK0_11	TXACK0_10	TXACK0_9	TXACK0_8	TXACK0_7	TXACK0_6	TXACK0_5	TXACK0_4	TXACK0_3	TXACK0_2	TXACK0_1	—	ABACK1_0	ABACK1_15	ABACK1_14	ABACK1_13	ABACK1_12	ABACK1_11	ABACK1_10	ABACK1_9	ABACK1_8	ABACK1_7	ABACK1_6	ABACK1_5	ABACK1_4	ABACK1_3	ABACK1_2	ABACK1_1	ABACK1_0	ABACK0_0	ABACK0_15	ABACK0_14	ABACK0_13	ABACK0_12	ABACK0_11	ABACK0_10	ABACK0_9	ABACK0_8	ABACK0_7	ABACK0_6	ABACK0_5	ABACK0_4	ABACK0_3	ABACK0_2	ABACK0_1	—	RXPR1_0	RXPR1_15	RXPR1_14	RXPR1_13	RXPR1_12	RXPR1_11	RXPR1_10	RXPR1_9	RXPR1_8	RXPR1_7	RXPR1_6	RXPR1_5	RXPR1_4	RXPR1_3	RXPR1_2	RXPR1_1	RXPR1_0	RXPR0_0	RXPR0_15	RXPR0_14	RXPR0_13	RXPR0_12	RXPR0_11	RXPR0_10	RXPR0_9	RXPR0_8	RXPR0_7	RXPR0_6	RXPR0_5	RXPR0_4	RXPR0_3	RXPR0_2	RXPR0_1	RXPR0_0	RFPR1_0	RFPR1_15	RFPR1_14	RFPR1_13	RFPR1_12	RFPR1_11	RFPR1_10	RFPR1_9	RFPR1_8	RFPR1_7	RFPR1_6	RFPR1_5	RFPR1_4	RFPR1_3	RFPR1_2	RFPR1_1	RFPR1_0	RFPR0_0	RFPR0_15	RFPR0_14	RFPR0_13	RFPR0_12	RFPR0_11	RFPR0_10	RFPR0_9	RFPR0_8	RFPR0_7	RFPR0_6	RFPR0_5	RFPR0_4	RFPR0_3	RFPR0_2	RFPR0_1	RFPR0_0	MBIMR1_0	MBIMR1_15	MBIMR1_14	MBIMR1_13	MBIMR1_12	MBIMR1_11	MBIMR1_10	MBIMR1_9	MBIMR1_8	MBIMR1_7	MBIMR1_6	MBIMR1_5	MBIMR1_4	MBIMR1_3	MBIMR1_2	MBIMR1_1	MBIMR1_0	MBIMR0_0	MBIMR0_15	MBIMR0_14	MBIMR0_13	MBIMR0_12	MBIMR0_11	MBIMR0_10	MBIMR0_9	MBIMR0_8	MBIMR0_7	MBIMR0_6	MBIMR0_5	MBIMR0_4	MBIMR0_3	MBIMR0_2	MBIMR0_1	MBIMR0_0	UMSR1_0	UMSR1_15	UMSR1_14	UMSR1_13	UMSR1_12	UMSR1_11	UMSR1_10	UMSR1_9	UMSR1_8	UMSR1_7	UMSR1_6	UMSR1_5	UMSR1_4	UMSR1_3	UMSR1_2	UMSR1_1	UMSR1_0	UMSR0_0	UMSR0_15	UMSR0_14	UMSR0_13	UMSR0_12	UMSR0_11	UMSR0_10	UMSR0_9	UMSR0_8	UMSR0_7	UMSR0_6	UMSR0_5	UMSR0_4	UMSR0_3	UMSR0_2	UMSR0_1	UMSR0_0
レジスタ 略称	ビット 31/23/15/7	ビット 30/22/14/6	ビット 29/21/13/5	ビット 28/20/12/4	ビット 27/19/11/3	ビット 26/18/10/2	ビット 25/17/9/1	ビット 24/16/8/0																																																																																																																																																																																																																																																																																			
TXPR1_0	TXPR1_15	TXPR1_14	TXPR1_13	TXPR1_12	TXPR1_11	TXPR1_10	TXPR1_9	TXPR1_8																																																																																																																																																																																																																																																																																			
	TXPR1_7	TXPR1_6	TXPR1_5	TXPR1_4	TXPR1_3	TXPR1_2	TXPR1_1	TXPR1_0																																																																																																																																																																																																																																																																																			
TXPR0_0	TXPR0_15	TXPR0_14	TXPR0_13	TXPR0_12	TXPR0_11	TXPR0_10	TXPR0_9	TXPR0_8																																																																																																																																																																																																																																																																																			
	TXPR0_7	TXPR0_6	TXPR0_5	TXPR0_4	TXPR0_3	TXPR0_2	TXPR0_1	—																																																																																																																																																																																																																																																																																			
TXCR1_0	TXCR1_15	TXCR1_14	TXCR1_13	TXCR1_12	TXCR1_11	TXCR1_10	TXCR1_9	TXCR1_8																																																																																																																																																																																																																																																																																			
	TXCR1_7	TXCR1_6	TXCR1_5	TXCR1_4	TXCR1_3	TXCR1_2	TXCR1_1	TXCR1_0																																																																																																																																																																																																																																																																																			
TXCR0_0	TXCR0_15	TXCR0_14	TXCR0_13	TXCR0_12	TXCR0_11	TXCR0_10	TXCR0_9	TXCR0_8																																																																																																																																																																																																																																																																																			
	TXCR0_7	TXCR0_6	TXCR0_5	TXCR0_4	TXCR0_3	TXCR0_2	TXCR0_1	—																																																																																																																																																																																																																																																																																			
TXACK1_0	TXACK1_15	TXACK1_14	TXACK1_13	TXACK1_12	TXACK1_11	TXACK1_10	TXACK1_9	TXACK1_8																																																																																																																																																																																																																																																																																			
	TXACK1_7	TXACK1_6	TXACK1_5	TXACK1_4	TXACK1_3	TXACK1_2	TXACK1_1	TXACK1_0																																																																																																																																																																																																																																																																																			
TXACK0_0	TXACK0_15	TXACK0_14	TXACK0_13	TXACK0_12	TXACK0_11	TXACK0_10	TXACK0_9	TXACK0_8																																																																																																																																																																																																																																																																																			
	TXACK0_7	TXACK0_6	TXACK0_5	TXACK0_4	TXACK0_3	TXACK0_2	TXACK0_1	—																																																																																																																																																																																																																																																																																			
ABACK1_0	ABACK1_15	ABACK1_14	ABACK1_13	ABACK1_12	ABACK1_11	ABACK1_10	ABACK1_9	ABACK1_8																																																																																																																																																																																																																																																																																			
	ABACK1_7	ABACK1_6	ABACK1_5	ABACK1_4	ABACK1_3	ABACK1_2	ABACK1_1	ABACK1_0																																																																																																																																																																																																																																																																																			
ABACK0_0	ABACK0_15	ABACK0_14	ABACK0_13	ABACK0_12	ABACK0_11	ABACK0_10	ABACK0_9	ABACK0_8																																																																																																																																																																																																																																																																																			
	ABACK0_7	ABACK0_6	ABACK0_5	ABACK0_4	ABACK0_3	ABACK0_2	ABACK0_1	—																																																																																																																																																																																																																																																																																			
RXPR1_0	RXPR1_15	RXPR1_14	RXPR1_13	RXPR1_12	RXPR1_11	RXPR1_10	RXPR1_9	RXPR1_8																																																																																																																																																																																																																																																																																			
	RXPR1_7	RXPR1_6	RXPR1_5	RXPR1_4	RXPR1_3	RXPR1_2	RXPR1_1	RXPR1_0																																																																																																																																																																																																																																																																																			
RXPR0_0	RXPR0_15	RXPR0_14	RXPR0_13	RXPR0_12	RXPR0_11	RXPR0_10	RXPR0_9	RXPR0_8																																																																																																																																																																																																																																																																																			
	RXPR0_7	RXPR0_6	RXPR0_5	RXPR0_4	RXPR0_3	RXPR0_2	RXPR0_1	RXPR0_0																																																																																																																																																																																																																																																																																			
RFPR1_0	RFPR1_15	RFPR1_14	RFPR1_13	RFPR1_12	RFPR1_11	RFPR1_10	RFPR1_9	RFPR1_8																																																																																																																																																																																																																																																																																			
	RFPR1_7	RFPR1_6	RFPR1_5	RFPR1_4	RFPR1_3	RFPR1_2	RFPR1_1	RFPR1_0																																																																																																																																																																																																																																																																																			
RFPR0_0	RFPR0_15	RFPR0_14	RFPR0_13	RFPR0_12	RFPR0_11	RFPR0_10	RFPR0_9	RFPR0_8																																																																																																																																																																																																																																																																																			
	RFPR0_7	RFPR0_6	RFPR0_5	RFPR0_4	RFPR0_3	RFPR0_2	RFPR0_1	RFPR0_0																																																																																																																																																																																																																																																																																			
MBIMR1_0	MBIMR1_15	MBIMR1_14	MBIMR1_13	MBIMR1_12	MBIMR1_11	MBIMR1_10	MBIMR1_9	MBIMR1_8																																																																																																																																																																																																																																																																																			
	MBIMR1_7	MBIMR1_6	MBIMR1_5	MBIMR1_4	MBIMR1_3	MBIMR1_2	MBIMR1_1	MBIMR1_0																																																																																																																																																																																																																																																																																			
MBIMR0_0	MBIMR0_15	MBIMR0_14	MBIMR0_13	MBIMR0_12	MBIMR0_11	MBIMR0_10	MBIMR0_9	MBIMR0_8																																																																																																																																																																																																																																																																																			
	MBIMR0_7	MBIMR0_6	MBIMR0_5	MBIMR0_4	MBIMR0_3	MBIMR0_2	MBIMR0_1	MBIMR0_0																																																																																																																																																																																																																																																																																			
UMSR1_0	UMSR1_15	UMSR1_14	UMSR1_13	UMSR1_12	UMSR1_11	UMSR1_10	UMSR1_9	UMSR1_8																																																																																																																																																																																																																																																																																			
	UMSR1_7	UMSR1_6	UMSR1_5	UMSR1_4	UMSR1_3	UMSR1_2	UMSR1_1	UMSR1_0																																																																																																																																																																																																																																																																																			
UMSR0_0	UMSR0_15	UMSR0_14	UMSR0_13	UMSR0_12	UMSR0_11	UMSR0_10	UMSR0_9	UMSR0_8																																																																																																																																																																																																																																																																																			
	UMSR0_7	UMSR0_6	UMSR0_5	UMSR0_4	UMSR0_3	UMSR0_2	UMSR0_1	UMSR0_0																																																																																																																																																																																																																																																																																			

項目	ページ	修正箇所																		
24.2 レジスタビット 一覧	24-45	修正																		
		<table border="1"> <thead> <tr> <th>レジスタ</th> <th>ビット</th> <th>ビット</th> <th>ビット</th> <th>ビット</th> <th>ビット</th> <th>ビット</th> <th>ビット</th> <th>ビット</th> </tr> </thead> <tbody> <tr> <td>略称</td> <td>31/23/15/7</td> <td>30/22/14/6</td> <td>29/21/13/5</td> <td>28/20/12/4</td> <td>27/19/11/3</td> <td>26/18/10/2</td> <td>25/17/9/1</td> <td>24/16/8/0</td> </tr> </tbody> </table>	レジスタ	ビット	ビット	ビット	ビット	ビット	ビット	ビット	ビット	略称	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
		レジスタ	ビット	ビット	ビット	ビット	ビット	ビット	ビット	ビット										
		略称	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0										

		TCMR0_0	TCMR0_15	TCMR0_14	TCMR0_13	TCMR0_12	TCMR0_11	TCMR0_10	TCMR0_9	TCMR0_8										
			TCMR0_7	TCMR0_6	TCMR0_5	TCMR0_4	TCMR0_3	TCMR0_2	TCMR0_1	TCMR0_0										
		TCMR1_0	TCMR1_15	TCMR1_14	TCMR1_13	TCMR1_12	TCMR1_11	TCMR1_10	TCMR1_9	TCMR1_8										
			TCMR1_7	TCMR1_6	TCMR1_5	TCMR1_4	TCMR1_3	TCMR1_2	TCMR1_1	TCMR1_0										
		TCMR2_0	TCMR2_15	TCMR2_14	TCMR2_13	TCMR2_12	TCMR2_11	TCMR2_10	TCMR2_9	TCMR2_8										
			TCMR2_7	TCMR2_6	TCMR2_5	TCMR2_4	TCMR2_3	TCMR2_2	TCMR2_1	TCMR2_0										

索引

【数字／記号】	
0 出力／1 出力.....	10-47
16 ビットタイマパルスユニット (TPU)	10-1
【A】	
A/D 変換器.....	18-1
A/D 変換精度.....	18-15
AC 特性.....	25-5
【C】	
CAN インタフェース.....	14-5
CAN スリープモード.....	14-80
CAN バスインタフェース.....	14-104
CRC 演算器.....	17-1
【D】	
DC 特性.....	25-2
DMAC インタフェース.....	14-102
DMA コントローラ (DMAC)	7-1
DTC ベクタアドレス.....	8-12
DTC ベクタアドレスオフセット	8-12, 8-13, 8-14
【E】	
ERI.....	13-42
【F】	
FCU コマンド一覧.....	20-57
FCU コマンド使用方法.....	20-62, 21-30
【I】	
I/O ポート.....	9-1
ID 並べ替え.....	14-30
IRQn 割り込み.....	5-16
【M】	
MCU 動作モード.....	3-1
【N】	
NMI 割り込み.....	5-16
【P】	
PLL 回路.....	22-11
【R】	
RAM.....	19-1
RAM のデータ保持.....	19-18
RCAN-TL1 のコントロールレジスタ.....	14-25
RCAN-TL1 のタイマレジスタ.....	14-63
RCAN-TL1 のメールボックスレジスタ.....	14-47
RCAN-TL1 のメモリマップ.....	14-6
RCAN-TL1 の割り込み要因.....	14-101
RCAN-TL1 の設定.....	14-78
RCAN-TL1 端子ポート設定.....	14-105
RSPI システム構成例.....	15-38
RSPI データフォーマット.....	15-46
RSPI のエラー検出機能.....	15-58
RSPI のモードと SPCR 設定の関係.....	15-35
RSPI の初期化.....	15-63
RSPI 端子の制御.....	15-37
RSPI 転送フォーマット.....	15-44, 15-45
RXI.....	13-42
【S】	
SSL ネゲート期間の MOSI 信号値の決定方法.....	15-38
【T】	
TEI.....	13-42
TSG と TSEG の設定.....	14-37
TTW[1:0] (タイムトリガウィンドウ)	14-22
TXI.....	13-42
【あ】	
アドレスエラー.....	4-7
アドレスマップ.....	3-6, 3-7
アドレスモード.....	7-20
インターバルタイマモード.....	12-7
ウォッチドッグタイマ (WDT)	12-1
ウォッチドッグタイマモード.....	12-6
エラープロテクト.....	21-35
エラープロテクト.....	20-84
エラープロテクト一覧.....	20-85, 21-35
オーバフロー.....	12-6

オーバランエラー	13-25
オフセット加算	7-34
オフセット誤差	18-15

【か】

外形寸法図	付録-2
外部クロック	22-11
外部バスクロック (Bφ)	22-1
外部割り込み	5-16
書き込み/消去ホストコマンド待ち状態	20-47
各処理状態におけるポートの状態	付録-1
拡張リピートエリア	7-18
拡張リピートエリア機能	7-32
各動作モードにおけるレジスタの状態	24-72
各ポートの出力信号有効設定一覧	9-27
各ポートのレジスタ構成	9-5
カスケード接続動作	10-56
型名一覧	付録-1
クロック同期式モード	13-34
クロック発振器	22-1
コントローラエリアネットワーク (RCAN-TL1)	14-1

【さ】

サイクルスチールモード	7-30
サスペンド動作	20-80
サンプル&ホールド回路	18-13
システムクロック (Iφ)	6-4, 22-1
システムマトリックス	14-24
周辺モジュールクロック (Pφ)	6-4, 22-1
出力トリガ	11-12
出力バッファ制御	9-10
状態遷移図	2-42
ショートアドレスモード	8-10
処理状態	2-41
シリアルコミュニケーションインタフェース (SCI)	13-1
シングルアドレスモード	7-21
シングルモード	18-11
水晶発振子	22-10
スキャンモード	18-12
スリープモード	23-2, 23-15

スレープモード動作	15-71
製品情報マット	21-36
絶対最大定格	25-1
絶対精度	18-15
全モジュールクロックストップモード	23-2, 23-16
送信トリガコントロールフィールド	14-22
送信トリガタイム (TTT)	14-21
送信バッファエンプティノ	
受信バッファフルフラグ	15-57
送信用内部アービトレーション	14-86
ソフトウェアスタンバイモード	23-2, 23-17
ソフトウェアプロテクト	21-34
ソフトウェアプロテクト	20-83

【た】

タイムスタンプ	14-19
タイムスレープ	14-90
タイムトリガコントロール (TT コントロール)	14-21
タイムトリガシステムの例	14-93
タイムトリガ送信	14-87
端子機能	1-7
チェイン転送	8-23
調歩同期式モード	13-18
低消費電力	23-1
データディレクションレジスタ	9-7
データトランスファコントローラ (DTC)	8-1
データレジスタ	9-7
テストモードの設定	14-83
デュアルアドレスモード	7-20
電気的特性	25-1
転送情報	8-10
転送情報ライトバックスキップ機能	8-19
転送情報リードスキップ機能	8-19
転送モード	7-24
同期クリア	10-50
同期プリセット	10-50
動作モード別ピン配置一覧	1-4
トグル出力	10-47
トラップ命令例外処理	4-10
トレース例外処理	4-6

【な】

内蔵 RAM アドレス空間	19-2
内部システムバス 1	6-3
内部周辺バス	6-3
内部バス	6-5
内部バス構成	6-3
内部ブロック図	1-2
内部割り込み	5-17
入力バッファコントロールレジスタ	9-8
ノーマル転送モード	7-24, 8-20
ノンオーバーラップ動作	11-13

【は】

バーストモード	7-31
ハードウェアプロテクト	21-34
ハードウェアプロテクト	20-83
バスアービトラクション	6-7
バスコントローラ (BSC)	6-1
バスサイクル分割	8-17
バスモード	7-30
バス権解放状態	2-41
パッケージ	1-1
発振器	22-10
パリティエラー	13-25
非直線性誤差	18-15
ビットレート	13-13
ピン配置図	1-3
ブートモード	20-28, 21-22
ブートモードの状態遷移	20-29
不当命令例外処理	4-11
フラッシュメモリ	20-1
フリーランニングカウント動作	10-45
プルアップ MOS コントロールレジスタ	9-8
フルアドレスモード	8-10
フルスケール誤差	18-15
ブレーク	13-43
フレーミングエラー	13-25
プログラマブルパルスジェネレータ (PPG)	11-1
プログラム実行状態	2-41
プログラム停止状態	2-41
ブロック転送モード	7-26, 8-22

プロテクト	21-34
プロテクト	20-83
分解能	18-15
分周器	22-11
ベクタテーブルアドレス	4-2
ベクタテーブルアドレスオフセット	4-2
ポートファンクションコントローラ	9-33
ポートレジスタ	9-7
ホルトモード	14-80

【ま】

マーク状態	13-43
マイクロプロセッサインタフェース (MPI)	14-4
マスタモード動作	15-64
マルチクロック機能	23-14
メールボックス	14-4, 14-7
メールボックスコントロール	14-4
メールボックスの機能の設定	14-17
メールボックスの再設定	14-99
メールボックスの役割	14-8
メッセージコントロールフィールド	14-12
メッセージ受信シーケンス	14-96
メッセージ送信リクエスト	14-85, 14-92
メッセージデータフィールド	14-19
モード 2	3-5
モード 3	3-5
モード遷移図	23-3
モード端子	3-1
モジュールストップモード	23-15

【や】

ユーザプログラムモード	20-57
-------------	-------

【ら】

ライターモード	20-83
ライトデータバッファ機能	6-6
リセット	4-4
リセットシーケンス	14-79
リセット時の内蔵 RAM データ保持	19-18
リセット状態	2-41
リピート転送モード	7-25, 8-21
量子化誤差	18-15

ループバックモード	15-84	DTCVBR	8-10
ルネサスシリアルペリフェラルインタフェース		ECPBCCNT	21-17
(RSPI)	15-1	ECPBCSTAT	21-18
例外処理	4-1	ECPMAT	21-19
例外処理後のスタックの状態	4-12	ECPRE0	21-12
例外処理状態	2-41	ECPRE1	21-13
例外処理ベクタテーブル	4-2	ECPWE0	21-14
レジスタ		ECPWE1	21-15
ABACK0	14-56	EXR	2-13
ABACK1	14-55	FAEINT	20-9, 21-10
ADCR	18-8	FASTAT	20-7, 21-8
ADCSR	18-7	FCPSR	20-23
ADDIAGR	18-10	FCURAME	20-11
ADDR	18-6	FENTRYR	20-18, 21-16
APPDCR	18-10	FMODR	20-6, 21-7
BCR0	14-35	FPROTR	20-20
BCR1	14-33	FRESETR	20-21
BCR2	6-2	FSTATR0	20-12
BRR	13-13	FSTATR1	20-15
CCR	2-12, 14-72	GSR	14-31
CMAX_TEW	14-66	ICR	9-8
CPUPCR	5-5	IER	5-9
CRA	8-6	IMR	14-45
CRB	8-7	INTCR	5-4
CRCCR	17-2	IPR	5-7
CRCDIR	17-3	IRR	14-38
CRCDOR	17-3	ISCR	5-10
CYCTR	14-74	ISR	5-14
DACR	7-14	LINCR	16-3
DAR	8-6	LINSTR	16-4
DBSR	7-7	LINTCNT	16-7
DDAR	7-5	LINTCR	16-6
DDR	9-5, 9-7	LINTOCNT	16-7
DMDR	7-8	MAC	2-14
DMRSR	7-19	MBIMR0	14-61
DOFR	7-5	MBIMR1	14-60
DR	9-7	MCR	14-25
DSAR	7-4	MDCR	3-2
DTCCR	8-9	MRA	8-4
DTCER	8-8	MRB	8-5
DTCR	7-6	MSTPCR	23-7

NDERH	11-3	SPBR	15-21
NDERL	11-3	SPCKD	15-26
NDRH	11-5	SPCMD	15-31
NDRL	11-5	SPCR	15-9
PC	2-12	SPDCR	15-22
PCR	9-8, 11-7	SPDR	15-17
PFCR6	9-34	SPND	15-28
PFCR8	9-35	SPPCR	15-12
PFCR9	9-36	SPSCR	15-18
PFCRA	9-37	SPSR	15-13
PFCRB	9-38	SPSSR	15-19
PFCRC	9-39	SSIER	5-15
PMR	11-8	SSLND	15-27
PODRH	11-4	SSLP	15-11
PODRL	11-4	SSR	13-10
PORT	9-7	SYSCR	3-3, 3-4
RAMACYC	19-14	TCMR0~2	14-75
RAMECC	19-9	TCNT	12-2
RAMEN	19-3	TCNT(TPU)	10-41
RAMERR	19-11	TCNTR	14-73
RAMINT	19-13	TCR	10-14
RAMWEN	19-6	TCSR	12-3
RDR	13-5	TDR	13-5
REC	14-46	TEC	14-46
RFMK	14-74	TGR	10-41
RFPR0	14-59	TIER	10-37
RFPR1	14-59	TIOR	10-19
RFTROFF	14-68	TMDR	10-18
ROMMAT	20-10	TSR	13-5, 14-69
ROSCCR	22-8	TSR(TPU)	10-38
RSR	13-5	TSTR	10-42
RSTCSR	12-5	TSYR	10-43
RXPR0	14-58	TTCR0	14-64
RXPR1	14-57	TTTSEL	14-77
SAR	8-6	TXACK0	14-54
SBR	2-14	TXACK1	14-53
SBYCR	23-4	TXCR0	14-52
SCKCR0	22-4	TXCR1	14-51
SCKCR1	22-6	TXPR0	14-50
SCR	13-8	TXPR1	14-49
SMR	13-6	UMSR0	14-62

UMSR1.....	14-61
VBR.....	2-14
汎用レジスタ.....	2-11
レジスタアドレス一覧（アドレス順）.....	24-2
レジスタビット一覧.....	24-44
ローカルアクセプタンスフィルタマスク （LAFM）.....	14-18

【わ】

割り込み.....	4-9, 5-16
割り込み応答時間.....	5-29
割り込みコントローラ.....	5-1
割り込み制御モード0.....	5-24
割り込み制御モード2.....	5-26
割り込み要因とベクタアドレスオフセット.....	5-18
割り込み例外処理シーケンス.....	5-28

H8SX/1720Sグループ ユーザーズマニュアル
ハードウェア編

発行年月日 2010年2月9日 Rev.1.00

2012年5月16日 Rev.2.00

発行 ルネサス エレクトロニクス株式会社

〒211-8668 神奈川県川崎市中原区下沼部1753



ルネサスエレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所・電話番号は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス販売株式会社 〒100-0004 千代田区大手町2-6-2（日本ビル）

(03)5201-5307

■技術的なお問合せおよび資料のご請求は下記へどうぞ。

総合お問合せ窓口：<http://japan.renesas.com/contact/>

H8SX/1720S グループ