本ドキュメントは Cypress (サイプレス) 製品に関する情報が記載されております。本ドキュメントには、「MB」から始まるシリーズ名，品名およびオーダ型格が記載されておりますが、これらはすべて「CY」から始まるシリーズ名，品名およびオーダ型格として、新規および既存のお客様に引き続き提供してまいります。

オーダ型格の調べ方について

1. www.cypress.com/pcnにアクセスしてください。
2. SEARCH PCNS フィールドに、オーダ型格などのキーワードを入力し、「Apply」をクリックしてください。
3. 該当するタイトル(Title)をクリックしてください。
4. 「Affected Parts List」ファイルを開いてください。
   当該ファイルに記載されている各種変更情報をご利用ください。

詳しいお問い合わせ先

Cypress 製品およびそのソリューションの詳細につきましては、お近くの営業所へお問い合わせください。

サイプレスについて

サイプレスは、世界で最も革新的な車載や産業機器、スマート家電、民生機器および医療機器製品向けに、最先端の組み込みシステム ソリューションを提供するリーディング カンパニーです。サイプレスのマイクロコントローラーや、アナログ IC、ワイヤレスおよび USB ベースのコネクティビティ ソリューション、高い信頼性と高性能を提供するメモリ製品は、各種機器メーカーの差異化製品の開発と早期市場参入を支援します。サイプレスは、ベストクラスのサポートと開発リソースをグローバルに提供することで、彼らが従来市場を破壊しまったく新しい製品カテゴリを歴史的なスピードで市場投入できるよう支援します。詳細はサイプレスのウェブサイト (japan.cypress.com) をご覧ください。

F²MC-16FX 16ビットマイクロコントローラ

MB96300シリーズ ハードウェアマニュアル

Doc. No. 002-07019 Rev. *A

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
www.cypress.com

Copyrights

© Cypress Semiconductor Corporation, 2006-2017. 本書面は， Cypress Semiconductor Corporation 及び Spansion LLC を含むその子会社（以下「Cypress」という。）に帰属する財産である。本書面（本書面に含まれ又は言及されているあらゆるソフトウェア若しくはファームウェア（以下「本ソフトウェア」という。）を含む）は，アメリカ合衆国及び世界のその他の国における知的財産法令及び条約に基づき Cypress が所有する。Cypress はこれらの法令及び条約に基づく全ての権利を留保し，本段落で特に記載されているものを除き，その特許権，著作権，商標権又はその他の知的財産権のライセンスを一切許諾しない。本ソフトウェアにライセンス契約書が伴っておらず，かつ Cypress との間で別途本ソフトウェアの使用方法を定める書面による合意がない場合，Cypress は，(1) 本ソフトウェアの著作権に基づき，(a) ソースコード形式で提供されている本ソフトウェアについて，Cypress ハードウェア製品と共に用いるためにのみ，かつ組織内部でのみ，本ソフトウェアの修正及び複製を行うこと，並びに (b) Cypress のハードウェア製品ユニットに用いるためにのみ，（直接又は再販売者及び販売代理店を介して間接のいずれかで）本ソフトウェアをバイナリーコード形式で外部エンドユーザーに配布すること，並びに (2) 本ソフトウェア（Cypress により提供され，修正がなされていないもの）が抵触する Cypress の特許権のクレームに基づき，Cypress ハードウェア製品と共に用いるためにのみ，本ソフトウェアの作成，利用，配布及び輸入を行うことについての非独占的で譲渡不能な一身専属的ライセンス（サブライセンスの権利を除く）を付与する。本ソフトウェアのその他の使用，複製，修正，変換又はコンパイルを禁止する。

適用される法律により許される範囲内で，Cypress は，本書面又はいかなる本ソフトウェア若しくはこれに伴うハードウェアに関しても，明示又は黙示をとわず，いかなる保証（商品性及び特定の目的への適合性の黙示の保証を含むがこれらに限られない）も行わない。いかなるコンピューティングデバイスも絶対に安全ということはない。従って，Cypress のハードウェアまたはソフトウェア製品に講じられたセキュリティ対策にもかかわらず，Cypress は，Cypress 製品への権限のないアクセスまたは使用といったセキュリティ違反から生じる一切の責任を負わない。加えて，本書面に記載された製品には，エラッタと呼ばれる設計上の欠陥またはエラーが含まれている可能性があり，公表された仕様とは異なる動作をする場合がある。適用される法律により許される範囲内で，Cypress は，別途通知することなく，本書面を変更する権利を留保する。Cypress は，本書面に記載のある，いかなる製品若しくは回路の適用又は使用から生じる一切の責任を負わない。本書面で提供されたあらゆる情報（あらゆるサンプルデザイン情報又はプログラムコードを含む）は，参照目的のためのみに提供されたものである。この情報で構成するあらゆるアプリケーション及びその結果としてのあらゆる製品の機能性及び安全性を適切に設計，プログラム，かつテストすることは，本書面のユーザーの責任において行われるものとする。Cypress 製品は，兵器，兵器システム，原子力施設，生命維持装置若しくは生命維持システム，蘇生用の設備及び外科的移植を含むその他の医療機器若しくは医療システム，汚染管理若しくは有害物質管理の運用のために設計され若しくは意図されたシステムの重要な構成部分としての使用，又は装置若しくはシステムの不具合が人身傷害，死亡若しくは物的損害を生じさせるようなその他の使用（以下「本目的外使用」という。）のためには設計，意図又は承認されていない。重要な構成部分とは，それの不具合が装置若しくはシステムの不具合を生じさせるか又はその安全性若しくは実効性に影響すると合理的に予想できるような装置若しくはシステムのあらゆる構成部分をいう。Cypress 製品のあらゆる本目的外使用から生じ，若しくは本目的外使用に関連するいかなる請求，損害又はその他の責任についても，Cypress はその全部又は一部をとわず一切の責任を負わず，かつ Cypress はそれら一切から本書により免除される。Cypress は Cypress 製品の本目的外使用から生じ又は本目的外使用に関連するあらゆる請求，費用，損害及びその他の責任（人身傷害又は死亡に基づく請求を含む）から免責補償される。

Cypress，Cypressのロゴ，Spansion，Spansionのロゴ及びこれらの組み合わせ，WICED，PSoC，CapSense，EZ-USB，F-RAM，及びTraveoは，米国及びその他の国におけるCypressの商標又は登録商標である。Cypressのより完全な商標のリストは，cypress.comを参照すること。その他の名称及びブランドは，それぞれの権利者の財産として権利主張がなされている可能性がある。

はじめに

1. 本書の目的と対象読者

Cypress 製品につきましては , 平素より格別のご愛顧を賜り厚くお礼申し上げます。
MB96300 シリーズは , ASIC (Application Specific IC) 対応が可能なオリジナル 16 ビット ワンチップマイクロコントローラである F2MC-16FX ファミリの汎用品の 1 つとして開発された製品です。

本書は, 実際にこの半導体を使って製品を設計する技術者の方を対象に, MB96300シリーズの機能や動作について記載しています。 本書をご一読ください。

2. 本書の全体構成

1. 概要

MB96300 スーパーシリーズはF^2MC-16FX マイコンのファミリメンバーに対応しています。多様なアプリケーションに対応するさまざまなマイコンシリーズで構成されます。スーパーシリーズのすべてのメンバーの間でプログラミングは共通です。

2. CPU

CPU について説明します。

3. 割込み

割込みの機能と動作について説明します。

4. DMA

DMA の機能と動作について説明します。

5. 遅延割込み

遅延割込みの機能と動作について説明します。

6. クロック

F2MC-16FX ファミリのマイクロコントローラで使用されるクロックについて説明します。

7. クロックモジュレータ

クロックモジュレータの概要と特長について説明します。ここでは , クロックモジュレータのレジスタ構造と動作について説明します。クロックモジュレータは設定可能な組合せに制限がありますので,ご使用になる場合は営業担当へお問い合わせください。

8. リセットとスタートアップ

F2MC-16FX ファミリのマイクロコントローラのリセットとスタートアップについて説明します。

9. スタンバイモードおよび電圧レギュレータ制御回路

スタンバイモード制御回路の機能と動作および内部電圧レギュレータの制御について説明します。レギュレータの制御は特にスタンバイモードでの電力消費を最適化できます。

10. ソースクロックタイマ

3 つのソースクロックタイマ (CR クロックタイマ , メインクロックタイマ , サブクロックタイマ) の機能と動作について説明します。

11. ウォッチドッグタイマとウォッチドッグリセット

ウォッチドッグタイマおよびリセットの機能と動作について説明します。

12. 外部バスインタフェース

外部バスインタフェースの機能と動作について説明します。

13. I/Oポート

I/O ポートの機能と動作について説明します。

14. 16ビット入出力タイマ

16 ビット入出力タイマの機能と動作について説明します。

15. 16ビットリロードタイマ(イベントカウント機能付き)

16 ビットリロードタイマ (イベントカウント機能付き) の機能と動作について説明します。

16. PPG

PPG の機能と動作について説明します。

17. 外部割込み

外部割込みの機能と動作について説明します。

18. A/Dコンバータ

A/D コンバータの機能と動作について説明します。

19. アラームコンパレータ

アラームコンパレータの機能と動作について説明します。

20. USART

LIN-USART の機能と動作について説明します。

21. 400 kHz I2Cインタフェース

高速 I^2C インタフェースの機能と動作について説明します。

22. CANコントローラ

CAN コントローラの機能と動作を説明します。

23. クロック出力機能

クロック出力機能とその動作を説明します。

24. リアルタイムクロック

リアルタイムクロックの機能と動作を説明します。

25. クロック補正ユニット

クロック補正ユニットの機能および動作について説明します。

26. LCDコントローラ/ドライバ

LCD コントローラ / ドライバの機能および動作について説明します。

27. ステッパモータコントローラ

ステッパモータコントローラの機能および動作について説明します。

28. サウンドジェネレータ

サウンドジェネレータの機能および動作について説明します。

29. USBファンクション

USB ファンクションの機能と動作について説明します。

30. USBホスト

USB ホストの機能と動作について説明します。

31. メモリパッチ機能

メモリパッチ機能を説明し , そしてデータパッチまたはデバッグ機能がどのようにメモリパッチ機能とともに動作するかを説明します。

32. ROM/RAMミラー機能選択モジュール

ROM ミラー機能選択モジュールについて説明します。

33. フラッシュメモリ

フラッシュメモリの機能および動作について説明します。

34. マスクROMメモリ

マスク ROM メモリの機能と動作について説明します。

35. ROM/フラッシュセキュリティ機能

ROM/ フラッシュセキュリティの機能および動作について説明します。

36. シリアルプログラミングの接続例

フラッシュメモリの回路内シリアルプログラミングのための MCU の接続方法について説明します。

付録

I/O マップ , 命令などを掲載します。

目次

はじめに 3

第 1 章 概要 23

1.1 特長 23
1.2 スーパーシリーズ品種構成 25
1.2.1 スーパーシリーズ品種構成 25
1.3 MB96V300 のブロックダイヤグラム 26
1.3.1 MB96V300 のブロックダイヤグラム 26
1.4 本書の使用に関する一般的な注意事項 27
1.4.1 デバイス依存のリソース 27
1.4.2 説明しているリソース 27
1.5 16 ビット I/O タイマの構成 28
1.5.1 インプットキャプチャユニットとフリーランタイマの接続 28
1.5.2 アウトプットコンペアユニットとフリーランタイマの接続 28
1.5.3 タイマクリア機能と OCU 間およびコンペアクリアとフリーランタイマ関係 28
1.6 LIN-USART のインプットキャプチャユニットのソースの選択 29
1.6.1 LIN-SUART と ICU の接続の概要 29
1.6.2 インプットキャプチャユニットソースの選択 30
1.7 周辺リソース端子のリロケーション 35
1.7.1 周辺リソース端子リロケーションレジスタの概要 35
1.7.2 周辺リソース端子リロケーションレジスタ 0(PRRR0) 36
1.7.3 周辺リソース端子リロケーションレジスタ 1(PRRR1) 37
1.7.4 周辺リソース端子リロケーションレジスタ 2(PRRR2) 38
1.7.5 周辺リソース端子リロケーションレジスタ 3(PRRR3) 39
1.7.6 周辺リソース端子リロケーションレジスタ 4(PRRR4) 40
1.7.7 周辺リソース端子リロケーションレジスタ 5(PRRR5) 41
1.7.8 周辺リソース端子リロケーションレジスタ 6(PRRR6) 42
1.7.9 周辺リソース端子リロケーションレジスタ 7(PRRR7) 43
1.7.10 周辺リソース端子リロケーションレジスタ 8(PRRR8) 44
1.7.11 周辺リソース端子リロケーションレジスタ 9(PRRR9) 45
1.7.12 周辺リソース端子リロケーションレジスタ 10(PRRR10) 46
1.7.13 周辺リソース端子リロケーションレジスタ 11(PRRR11) 47
1.7.14 周辺リソース端子リロケーションレジスタ 12(PRRR12) 48
1.7.15 周辺リソース端子リロケーションレジスタ 13(PRRR13) 49

第 2 章 CPU 51
2.1 CPU の概要 51
2.2 ハードウェア構造 52
2.2.1 CPU のハードウェア構造 52
2.2.2 16FX コアのハードウェア構造 54
2.3 メモリ空間 56
2.3.1 メモリ領域 56
2.3.2 リニア方式によるアドレス指定 59
2.3.3 バンク方式によるアドレス指定 60
2.3.4 メモリ空間上の多バイトデータ 62
2.4 専用レジスタ 63
2.4.1 アキュムレータ (A) 65
2.4.2 ユーザスタックポインタ (USP) とシステムスタックポインタ (SSP) 67
2.4.3 プロセッサステータス (PS) 68
2.4.4 プログラムカウンタ (PC) 72
2.4.5 ダイレクトページレジスタ (DPR) 73
2.4.6 バンクレジスタ (PCB, DTB, ADB, USB, SSB) 74
2.5 汎用レジスタ 75
2.5.1 レジスタバンク 75
2.5.2 汎用レジスタのアドレス指定 77
2.6 プリフィックスコード 78
2.6.1 バンクセレクトプリフィックス 78
2.6.2 コモンレジスタバンクプリフィックス (CMR) 80
2.6.3 フラグ変化抑止プリフィックス (NCC) 81
2.6.4 プリフィックスコードに関する制約 82

第 3 章 割込み 85
3.1 割込みの概要 85
3.1.1 ハードウェア割込み 85
3.1.2 ソフトウェア割込み 85
3.1.3 例外 86
3.1.4 ダイレクトメモリアクセス (DMA) 86
3.2 割込みベクタ 87
3.2.1 割込みベクタ 87
3.2.2 割込みベクタテーブル 87
3.3 割込み制御レジスタ (ICR) 89
3.3.1 割込み制御レジスタ (ICR) 89
3.4 マスク不可能割込み (NMI) 91
3.4.1 NMI 制御ステータスレジスタ (NMI) 91
3.5 割込みフロー 93
3.5.1 割込みフロー 93
3.6 ハードウェア割込み 95
3.6.1 ハードウェア割込み 95

3.6.2 ハードウェア割込みシステムの構造 95
3.6.3 ハードウェア割込みの動作 96
3.6.4 ハードウェア割込みの処理時間 97
3.7 ソフトウェア割込み 98
3.7.1 ソフトウェア割込み 98
3.7.2 ソフトウェア割込みシステムの構造 98
3.7.3 ソフトウェア割込みの動作 99
3.8 多重割込み 100
3.8.1 多重ハードウェア割込み 100
3.8.2 多重ソフトウェア割込み 100
3.8.3 割込みの優先順位 101
3.9 例外 103
3.9.1 ソフトウェア例外 (op コード) 103
3.9.2 ハードウェア例外 (マスク不可能割込み) 104

第 4 章 DMA 107
4.1 概要 107
4.1.1 割込み処理と比較した DMA 107
4.1.2 DMA 機能 109
4.1.3 構造 109
4.2 DMA レジスタ 111
4.2.1 DMA 割込み要求選択レジスタ (DISEL) 112
4.2.2 DMA ステータスレジスタ (DSR) 113
4.2.3 DMA 停止ステータスレジスタ (DSSR) 114
4.2.4 DMA 許可レジスタ (DER) 115
4.3 DMA ディスクリプタ 116
4.3.1 DMA I/O アドレスポインタバンク選択レジスタ (IOABK)*1 118
4.3.2 データカウントレジスタ (DCT) 119
4.3.3 I/O レジスタアドレスポインタ (IOA) 120
4.3.4 DMA 制御レジスタ (DMACS) 121
4.3.5 バッファアドレスポインタ (BAP) 123
4.4 DMA コントローラの動作 124
4.5 DMA 転送の例 127

第 5 章 遅延割込み 133
5.1 遅延割込みモジュールの概要 133
5.2 遅延割込みレジスタ 134
5.3 遅延割込みの動作 135

第 6 章　クロック 137
6.1　クロック 137
6.2　クロック制御レジスタ 141
6.2.1　クロック選択レジスタ (CKSR) 142
6.2.2　クロックモニタレジスタ (CKMR) 145
6.2.3　クロック安定化選択レジスタ (CKSSR) 148
6.2.4　クロック周波数制御レジスタ (CKFCR) 152
6.2.5　PLL 制御レジスタ (PLLCR) 155
6.2.6　クロック入力および LVD 制御レジスタ (CILCR) 159
6.3　クロックモード 161
6.3.1　クロックモードの定義 161
6.3.2　クロックソースの切換え 161
6.3.3　クロックソースの起動と禁止 162
6.4　PLL の構成 163
6.4.1　PLL クロック逓倍回路のコンポーネント 163
6.4.2　VCO クロック逓倍選択ビット VMS[2:0] の設定 164
6.5　発振安定待ち時間 165
6.5.1　発振安定待ち時間 165
6.6　発振器または外部クロックとマイクロコントローラとの接続 167

第 7 章　クロックモジュレータ 169
7.1　概要 169
7.2　レジスタの説明 170
7.2.1　クロックモジュレータ制御レジスタ (CMCR) 170
7.2.2　クロック変調パラメータレジスタ (CMPR) 174
7.3　アプリケーションノート 177

第 8 章　リセットとスタートアップ 179
8.1　リセット 179
8.1.1　リセット要因 179
8.2　リセット , システムクロック , および安定待ち時間 182
8.2.1　リセット要因と安定待ち時間 182
8.2.2　外部リセット発生時の安定待ち時間 182
8.2.3　メイン発振器の安定 183
8.2.4　サブ発振器の安定 183
8.3　電源リセットおよび外部リセット後のスタートアップ 184
8.3.1　リセット延長 184
8.3.2　ソースクロックタイマとクロック準備完了モニタビット 185
8.3.3　電源リセットまたは外部リセットが発生した後でのプログラム実行の開始 186
8.3.4　メインクロックモードへの移行 186
8.3.5　PLL クロックモードへの移行 186

8.3.6 サブクロックモードへの移行 186
8.4 ブート ROM プログラムの実行と動作モード , および ROM 構成ブロック 187
8.4.1 外部バス 187
8.4.2 モード端子 187
8.4.3 動作モード 188
8.4.4 ROM 構成ブロック (RCB) 189
8.4.5 RCB マーカの機能説明 190
8.4.6 内部ベクタモードのフローチャート 193
8.4.7 外部ベクタモードのフローチャート 194
8.5 リセット制御レジスタ 195
8.5.1 リセット構成レジスタ (RCR) 195
8.5.2 リセット要因およびクロック状態レジスタ (RCCSR/RCCSRC) 198
8.5.3 クロック入力および LVD 制御レジスタ (CILCR) 203
8.6 クロック停止検出機能およびリセットの動作 205
8.6.1 クロック停止検出回路の機能 205
8.6.2 クロック停止検出回路の構成 206
8.6.3 クロック欠落フラグ MCMF と SCMF 207
8.6.4 クロック停止検出リセット 207
8.6.5 クロック停止検出リセットおよびスタンバイモード 208
8.6.6 クロック停止検出リセットとウォッチドッグタイマ 209
8.7 低電圧リセット機能の動作 209
8.7.1 低電圧リセットの機能 209
8.7.2 低電圧検出およびリセット回路の構成 209

第 9 章 スタンバイモードおよび電圧レギュレータ制御回路 211
9.1 CPU 動作モードの概要 211
9.1.1 CPU 動作モードと消費電流 211
9.1.2 状態遷移図 212
9.1.3 ランモード 213
9.1.4 スリープモード 214
9.1.5 タイマモード 215
9.1.6 ストップモード 215
9.2 スタンバイモード制御レジスタ (SMCR) 216
9.3 電圧レギュレータ制御レジスタ (VRCR) 218
9.4 スタンバイモード 222
9.4.1 スリープモード (CR スリープ , メインスリープ , PLL スリープ , およびサブスリープモード) 223
9.4.2 タイマモード (CR タイマ , メインタイマ , PLL タイマ , およびサブタイマモード) 226
9.4.3 ストップモード 229
9.5 モード切換え表と動作状態 232
9.6 スタンバイモードに関する使用上の注意事項 234
9.7 電圧レギュレータの動作 236
9.7.1 電圧レギュレータの動作モードの変更 237

9.7.2 出力電圧の電圧レギュレータの設定 239

第 10 章 ソースクロックタイマ 241
10.1 概要 241
10.2 CR クロックタイマ 242
10.2.1 CR クロックタイマ制御レジスタ (RCTCR) 243
10.2.2 CR クロックタイマの動作 247
10.3 メインクロックタイマ 248
10.3.1 メインクロックタイマ制御レジスタ (MCTCR) 249
10.3.2 メインクロックタイマの動作 253
10.4 サブクロックタイマ 254
10.4.1 サブクロックタイマ制御レジスタ (SCTCR) 255
10.4.2 サブクロックタイマの動作 258

第 11 章 ウォッチドッグタイマとウォッチドッグリセット 259
11.1 ウォッチドッグタイマおよびリセットの概要 259
11.2 ウォッチドッグタイマ制御レジスタ 261
11.2.1 ウォッチドッグタイマ構成レジスタ (WDTC) 262
11.2.2 ウォッチドッグタイマクリアパターンレジスタ (WDTCP) 266
11.3 ウォッチドッグタイマの動作 267
11.2.1 ウォッチドッグタイマの起動と停止 267
11.2.2 ウォッチドッグカウンタのクリア 267
11.2.3 ウォッチドッグカウンタの停止 267
11.2.4 ウォッチドッグカウンタのクロックソースの選択 267
11.2.5 スタンバイモード 268
11.2.6 ウォッチドッグのリセット 268
11.2.7 ウォッチドッグタイマインターバルの設定 268
11.2.8 ウォッチドッグタイマの動作モードの選択 268
11.2.9 ウォッチドッグタイマリセット要因 270

第 12 章 外部バスインタフェース 271
12.1 外部バスの概要 271
12.1.1 外部バスインタフェースの特長 271
12.1.2 用語 272
12.1.3 外部バス領域 272
12.1.4 各バスモードのメモリ空間 273
12.2 外部バス構成レジスタ 274
12.2.1 外部バスモードレジスタ (EBM) 278
12.2.2 外部バスクロックおよび機能レジスタ (EBCF) 280

12.2.3 外部バスアドレス出力許可レジスタ (EBAE2 ～ EBAE0) 284
12.2.4 外部バス制御信号レジスタ (EBCS) 285
12.2.5 外部領域構成レジスタ (EACH[5:0], EACL[5:0]) 287
12.2.6 外部領域選択レジスタ (EAS2 ～ EAS5) 295
12.3 外部メモリアクセス制御信号の動作 298
12.3.1 マルチプレクス外部バスモード 298
12.3.2 ノンマルチプレクス外部バスモード 305
12.3.3 レディ機能 310
12.3.4 ホールド機能 313
12.4 外部バスの使用に関する注意事項 316
12.5 外部ブートベクタフェッチ 320
12.6 MCU 状態毎の端子状態 322
12.6.1 外部バス端子の状態 322

第 13 章 I/O ポート 325
13.1 I/O ポート 325
13.2 I/O ポートレジスタ 327
13.2.1 ポートデータレジスタ (PDRnn) 329
13.2.2 外部端子状態レジスタ (EPSRnn) 331
13.2.3 データ方向レジスタ (DDRnn) 332
13.2.4 ポート入力許可レジスタ (PIERnn) 333
13.2.5 ポート入力レベルレジスタ (PILRnn) と拡張ポート入力レベルレジスタ (EPILRnn) 334
13.2.6 ポート出力駆動レジスタ (PODRnn) 336
13.2.7 ポート high 駆動レジスタ (PHDRnn) 338
13.2.8 プルアップ制御レジスタ (PUCRnn) 339
13.3 レジスタの用途 340

第 14 章 16 ビット入出力タイマ 341
14.1 16 ビット入出力タイマの概要 341
14.2 16 ビット入出力タイマのレジスタ 343
14.3 16 ビットフリーランタイマ 345
14.3.1 データレジスタ (TCDTn) 346
14.3.2 コントロールステータスレジスタ (TCCSLn) 347
14.3.3 16 ビットフリーランタイマの動作 350
14.4 アウトプットコンペアユニット 352
14.4.1 アウトプットコンペアレジスタ (OCCP(2n) / OCCP(2n+1)) 354
14.4.2 アウトプットコンペアのコントロールステータスレジスタ (OCS(2n) / OCS(2n+1)) 355
14.4.3 16 ビットアウトプットコンペア動作 360
14.5 インプットキャプチャユニット 365
14.5.1 インプットキャプチャユニットレジスタの詳細 366
14.5.2 16 ビットインプットキャプチャの動作 371

第 15 章　16 ビットリロードタイマ (イベントカウント機能付き) 373
15.1　16 ビットリロードタイマ (イベントカウント機能付き) の概要 373
15.2　16 ビットリロードタイマ (イベントカウント機能付き) 375
15.2.1　タイマ制御状態レジスタ (TMCSRn) 377
15.2.2　16 ビットタイマレジスタ (TMRn) / 16 ビットリロードレジスタ (TMRLRn) のレジスタ構成 380
15.3　16 ビットリロードタイマの内部クロックカウンタ動作と外部イベントカウンタ動作 382
15.4　16 ビットリロードタイマのアンダフロー動作 384
15.5　16 ビットリロードタイマの出力端子機能 385
15.6　カウンタの動作状態 386
15.7　16 ビットリロードタイマのカスケード接続 387

第 16 章　PPG 389
16.1　PPG の概要 389
16.2　レジスタ 392
16.2.1　レジスター覧 392
16.2.2　PPG 制御ステータスレジスタ (PCNn) 393
16.2.3　汎用制御レジスタ 1(GCN1g) 396
16.2.4　汎用制御レジスタ 2(GCN2g) 397
16.2.5　PPG 周期設定レジスタ (PCSRn) 398
16.2.6　PPG デューティ設定レジスタ (PDUTn) 399
16.2.7　PPG タイマレジスタ (PTMRn) 399
16.3　PPG の動作 400
16.3.1　PWM 動作 400
16.3.2　ワンショット動作 401
16.3.3　再起動動作 402
16.4　使用上の注意 403

第 17 章　外部割込み 405
17.1　外部割込みの概要 405
17.2　外部割込みレジスタ 407
17.3　外部割込み機能の使用上の注意 410

第 18 章　A/D コンバータ 411
18.1　A/D コンバータの概要 411
18.2　A/D コンバータのレジスタ 413
18.2.1　A/D 制御ステータスレジスタ (ADCS) 415
18.2.2　データレジスタ (ADCR) 419
18.2.3　A/D 設定レジスタ (ADSR) 420
18.2.4　A/D 拡張構成レジスタ (ADECR) 423

18.2.5 アナログ入力許可レジスタ (ADERx) 426
18.3 A/D コンバータの動作 427
18.3.1 単発変換モード 1/2 427
18.3.2 連続変換モード 427
18.3.3 停止変換モード 427
18.4 DMA を使用した変換 428
18.4.1 DMA を使用した変換 428
18.5 変換データ保護機能 429
18.5.1 A/D 変換データ保護機能の説明 429
18.5.2 A/D 変換結果が CPU によって読み出される場合のデータ保護機能 430
18.5.3 A/D 変換結果が DMA によって転送される場合のデータ保護機能 430
18.5.4 変換データ保護機能のフロー例 (DMA 使用) 431
18.5.5 使用上の注意 432

第 19 章 アラームコンパレータ 433
19.1 アラームコンパレータの概要 433
19.2 アラームコンパレータのレジスタ 435
19.2.1 アラームコンパレータ制御 / 状態レジスタ (ACSRn) 435
19.2.2 アラームコンパレータ拡張制御 / 状態レジスタ (AECSRn) 437
19.3 アラームコンパレータの動作モード 438

第 20 章 USART 441
20.1 USART の概要 441
20.2 USART の構成 443
20.2.1 USART のブロックダイヤグラム 443
20.3 USART の端子 448
20.4 USART のレジスタ 449
20.4.1 シリアル制御レジスタ (SCRn) 450
20.4.2 シリアルモードレジスタ (SMRn) 452
20.4.3 シリアルステータスレジスタ (SSRn) 454
20.4.4 受信データレジスタ / 送信データレジスタ (RDRn/TDRn) 456
20.4.5 拡張ステータス制御レジスタ (ESCRn) 457
20.4.6 拡張通信制御レジスタ (ECCRn) 459
20.4.7 ボーレート / リロードカウンタレジスタ (BGRn) 461
20.4.8 拡張シリアル割込みレジスタ (ESIRn) 462
20.5 USART の割込み 464
20.5.1 LIN-USART 割込み 464
20.5.2 USART DMA 機能 466
20.5.3 受信割込みの生成とフラグのセットタイミング 466
20.5.4 送信割込みの生成とフラグのセットタイミング 467
20.6 USART のボーレート 469

20.6.1 USART のボーレート選択 469
20.6.2 ボーレートの設定 471
20.6.3 リロードカウンタのリスタート 474
20.7 USART の動作 475
20.7.1 非同期モード (動作モード 0, 1) での動作 477
20.7.2 同期モード (動作モード 2) での動作 479
20.7.3 LIN 機能 (動作モード 3) の動作 482
20.7.4 シリアル端子の直接アクセス 485
20.7.5 双方向通信機能 (ノーマルモード) 486
20.7.6 マスタ / スレーブ通信機能 (マルチプロセッサモード) 488
20.7.7 LIN 通信機能 491
20.7.8 LIN 通信での USART(動作モード 3) のフローチャート例 492
20.8 USART の使用上の注意 494
20.8.1 USART の使用上の注意 494

第 21 章 400 kHz I²C インタフェース 499
21.1 I²C インタフェースの概要 499
21.2 I²C インタフェースのレジスタ 501
21.2.1 バスステータスレジスタ (IBSRn) 503
21.2.2 バスコントロールレジスタ (IBCRn) 506
21.2.3 10 ビットスレーブアドレスレジスタ (ITBAn) 514
21.2.4 10 ビットスレーブアドレスマスクレジスタ (ITMKn) 515
21.2.5 7 ビットスレーブアドレスレジスタ (ISBAn) 516
21.2.6 7 ビットスレーブアドレスマスクレジスタ (ISMKn) 517
21.2.7 データレジスタ (IDARn) 518
21.2.8 クロックコントロールレジスタ (ICCRn) 519
21.3 I²C インタフェースの動作 522
21.4 プログラミングフローチャート 525

第 22 章 CAN コントローラ 527
22.1 概要 527
22.2 機能の説明 528
22.3 各レジスタの説明 533
22.4 CAN デバイス関連のレジスタ 537
22.4.1 CAN 出力許可レジスタ (COERn) 537
22.5 CAN プロトコル関連のレジスタ 538
22.5.1 CAN 制御レジスタ (CTRLRn) 538
22.5.2 ステータスレジスタ (STATRn) 541
22.5.3 エラーカウンタ (ERRCNTn) 543
22.5.4 ビットタイミングレジスタ (BTRn) 544
22.5.5 テストレジスタ (TESTRn) 545

22.5.6 BRP 拡張レジスタ (BRPERn) 547
22.6 メッセージインタフェースレジスタセット 548
22.6.1 IFx コマンドリクエストレジスタ (IFxCREQn) 561
22.6.2 IFx コマンドマスクレジスタ (IFxCMSKn) 551
22.6.3 IFx マスクレジスタ (IFxMSK1n, IFxMSK2n) 553
22.6.4 IFx アービトレーションレジスタ (IFxARB1n, IFxARB2n) 554
22.6.5 IFx メッセージ制御レジスタ (IFxMCTRn) 555
22.6.6 IFx データ A およびデータ B レジスタ (IFxDTA1n, IFxDTA2n, IFxDTB1n, IFxDTB2n) 556
22.7 メッセージメモリ内のメッセージオブジェクト 557
22.8 メッセージハンドラのレジスタ 561
22.8.1 割込みレジスタ (INTRn) 561
22.8.2 送信リクエストレジスタ (TREQR1n, TREQR2n) 563
22.8.3 新規データレジスタ (NEWDT1n, NEWDT2n) 565
22.8.4 割込み保留レジスタ (INTPND1n, INTPND2n) 567
22.8.5 メッセージ有効レジスタ (MSGVAL1n, MSGVAL2n) 569
22.9 CAN アプリケーション 571

第 23 章 クロック出力機能 581
23.1 クロック出力機能の概要 581
23.2 クロック出力の構成と起動レジスタ 582
23.3 クロック出力機能の動作 588

第 24 章 リアルタイムクロック 589
24.1 リアルタイムクロックの概要 589
24.2 リアルタイムクロックのレジスタ 591
24.2.1 タイマ制御レジスタ (WTCR) 593
24.2.2 タイマ制御拡張レジスタ (WTCER) 596
24.2.3 クロック選択レジスタ (WTCKSR) 597
24.2.4 サブセカンドレジスタ 598
24.2.5 時 / 分 / 秒レジスタ (WTHR, WTMR, WTSR) 599
24.3 動作 601
24.4 注意事項 603
24.4.1 注意事項 603

第 25 章 クロック補正ユニット 605
25.1 概要 605
25.2 レジスタの説明 608
25.2.1 クロック補正ユニットレジスタ 608
25.2.2 クロック補正ユニット制御レジスタ (CUCR) 608

25.2.3 クロック補正ユニット測定時間タイマデータレジスタ (CUTD) 610
25.2.4 クロック補正ユニット補正タイマデータレジスタ (24 ビット)(CUTR) 612
25.3 アプリケーションノート 613
25.3.1 32.768kHz 613
25.3.2 100kHz 613
25.3.3 精度 614
25.3.4 電力消費 614
25.3.5 測定限界 614

第 26 章 LCD コントローラ / ドライバ 615
26.1 LCD コントローラ / ドライバの構成 615
26.1.1 LCD コントローラ / ドライバ分割抵抗 618
26.2 LCD コントローラレジスタ 621
26.2.1 LCD 制御レジスタ (LCR) 621
26.2.2 LCD 共通端子切換えレジスタ 623
26.2.3 LCD 拡張制御レジスタ (LECR) 624
26.3 LCD 許可レジスタ (LCDER, LCDVER) 625
26.4 LCD コントローラ / ドライバの表示用 RAM 627
26.4.1 表示データ設定のためのメモリ領域 (VRAM) 627
26.4.2 VRAM とコモンセグメント端子間の対応関係 627
26.5 LCD コントローラ / ドライバの動作 628
26.5.1 LCD コントローラ / ドライバ (LCDC) の動作 628
26.5.2 LCD コントローラ / ドライバ動作中 (1/2 デューティサイクル) の出力波形 629
26.5.3 LCD コントローラ / ドライバ動作中 (1/3 デューティサイクル) の出力波形 631
26.5.3 LCD コントローラ / ドライバ動作中 (1/4 デューティサイクル) の出力波形 633
26.6 注意事項 635

第 27 章 ステッパモータコントローラ 637
27.1 ステッパモータコントローラの概要 637
27.2 ステッパモータコントローラのレジスタ 638
27.3 PWM 制御レジスタ (PWCn) 639
27.3.1 PWM 制御レジスタ (PWCn) 639
27.4 PWM 拡張制御レジスタ (PWECn) 640
27.4.1 PWM 拡張制御レジスタ (PWECn) 640
27.5 PWM1, PWM2 比較レジスタ (PWC1n, PWC2n) 641
27.5.1 PWM1, PWM2 比較レジスタ (PWC1n, PWC2n) 641
27.6 PWM1, PWM2 選択レジスタ (PWS1n, PWS2n) 644
27.6.1 PWM1, PWM2 選択レジスタ (PWS1n, PWS2n) 644
27.6.2 PWM1 選択レジスタ (PWS1n) 644
27.6.3 PWM2 選択レジスタ (PWS2n) 645
27.7 ステッパモータコントローラの動作 646

27.8 ステッパモータコントローラの使用上の注意 648

第 28 章 サウンドジェネレータ 649
28.1 サウンドジェネレータの概要 649
28.2 サウンドジェネレータのレジスタ 650
28.2.1 サウンドコントロールレジスタ (SGCRn) 651
28.2.2 振幅データレジスタ (SGARn) 655
28.2.3 周波数データレジスタ (SGFRn) 657
28.2.4 トーンカウントレジスタ (SGTRn) 658
28.2.5 デクリメントグレードレジスタ (SGDRn) 659

第 29 章 USB ファンクション 661
29.1 USB ファンクションの概要 661
29.2 USB ファンクションのブロックダイヤグラム 662
29.3 USB ファンクションのレジスタ 663
29.3.1 UDC 制御レジスタ (UDCCn) 667
29.3.2 EP0 制御レジスタ (EP0Cn) 669
29.3.3 EP1 ～ EP5 制御レジスタ (EP1Cn ～ EP5Cn) 671
29.3.4 タイムスタンプレジスタ (TMSPn) 675
29.3.5 UDC ステータスレジスタ (UDCSn) 676
29.3.6 UDC 割込み許可レジスタ (UDCIEn) 678
29.3.7 EP0I ステータスレジスタ (EP0ISn) 680
29.3.8 EP0O ステータスレジスタ (EP0OSn) 682
29.3.9 EP1 ～ EP5 ステータスレジスタ (EP1Sn ～ EP5Sn) 684
29.3.10 EP0 ～ EP5 データレジスタ (EP0DTn ～ EP5DTn) 688
29.4 USB ファンクションの動作説明 689
29.4.1 USB ファンクションの動作 689
29.4.2 接続検出と切断検出 692
29.4.3 コマンド応答時の各レジスタ動作 695
29.4.4 STALL 応答と解除 697
29.4.5 サスペンドモード機能 701
29.4.6 ウェイクアップ機能 702
29.4.7 DMA 転送機能 703
29.4.8 NULL 転送機能 707

第 30 章 USB ホスト 709
30.1 USB ホストの特長 709
30.2 USB ホストの制限事項 710
30.3 USB ホストのブロックダイヤグラム 711

30.4 USB ホストのレジスタ 712
30.4.1 ホストコントロールレジスタ (HCNTn) 713
30.4.2 ホスト割込みレジスタ (HIRQn) 717
30.4.3 ホストエラーステータスレジスタ (HERRn) 720
30.4.4 ホスト状態ステータスレジスタ (HSTATEn) 723
30.4.5 SOF 割込み FRAME 比較レジスタ (HFCOMPn) 725
30.4.6 リトライタイマ設定レジスタ (HRTIMERLn, HRTIMERMn, HRTIMERHn) 726
30.4.7 ホストアドレスレジスタ (HADRn) 727
30.4.8 EOF 設定レジスタ (HEOFn) 728
30.4.9 FRAME 設定レジスタ (HFRAMEn) 729
30.4.10 ホストトークンエンドポイントレジスタ (HTOKENn) 730
30.5 USB ホストの動作 732
30.5.1 デバイスの接続 732
30.5.2 USB バスのリセット 734
30.5.3 トークンパケット 735
30.5.4 データパケット 737
30.5.5 ハンドシェークパケット 738
30.5.6 リトライ機能 739
30.5.7 SOF 割込み 740
30.5.8 エラーステータス 742
30.5.9 パケット終了 743
30.5.10 サスペンド・レジューム 744
30.5.11 デバイスの切断 747
30.6 トークンフローチャート 748

第 31 章 メモリパッチ機能 751
31.1 メモリパッチ機能の概要 751
31.2 メモリパッチ機能のレジスタ 753
31.3 メモリパッチ機能の動作 763

第 32 章 ROM /RAM ミラー機能選択モジュール 765
32.1 ROM/RAM ミラー機能選択モジュールの概要 765
32.2 ROM ミラー機能選択レジスタ (ROMM) 766

第 33 章 フラッシュメモリ 771
33.1 フラッシュメモリの概要 771
33.1.1 フラッシュメモリのタイプ 771
33.1.2 フラッシュメモリの特長 772

33.1.3 フラッシュメモリへの書込み / 消去 772
33.2 フラッシュメモリのセクタ構成 773
33.3 フラッシュメモリモード 776
33.4 プログラム / データフラッシュ (メインフラッシュ) メモリ制御レジスタ 777
33.4.1 フラッシュメモリ制御ステータスレジスタ (MCSRA, MCSRB) 778
33.4.2 フラッシュメモリタイミング設定レジスタ (MTCRA, MTCRB) 781
33.4.3 フラッシュメモリ書込み制御レジスタ 0 ～ 5 (FMWC0 ～ FMWC5) 786
33.5 データフラッシュメモリ制御レジスタ 788
33.5.1 データフラッシュメモリ制御ステータスレジスタ (DFCSA, DFCSB) 789
33.5.2 データフラッシュメモリ書込みコマンドシーケンサ制御レジスタ (DFWCA, DFWCB) 791
33.5.3 データフラッシュメモリ書込みコマンドシーケンサステータスレジスタ (DFWSA, DFWSB) 793
33.6 フラッシュメモリ自動アルゴリズムの起動 796
33.7 自動アルゴリズム実行状態の確認 800
33.7.1 データポーリングフラグ (DQ7) 802
33.7.2 トグルビットフラグ (DQ6) 804
33.7.3 タイミングリミット超過フラグ (DQ5) 805
33.7.4 セクタ消去タイマフラグ (DQ3) 806
33.7.5 トグルビット 2 フラグ (DQ2) 807
33.8 フラッシュメモリの書込み / 消去の詳細説明 809
33.8.1 読出し / リセット状態の設定 810
33.8.2 書込みコマンドシーケンスでのデータ書込み 811
33.8.3 書込みコマンドシーケンサを使用したデータフラッシュへの書込み 813
33.8.4 全データ消去 (チップ消去) 815
33.8.5 任意データ消去 (セクタ消去) 816
33.8.6 セクタ消去一時停止 818
33.8.7 セクタ消去再開 819
33.8.8 セクタ消去 / 書込み保護 820
33.9 フラッシュメモリ使用上の注意 823
33.9.1 フラッシュメモリ使用上の注意 823
33.10 フラッシュメモリプログラム例 824

第 34 章 マスク ROM メモリ 831
34.1 概要 831
34.2 マスク ROM メモリ制御レジスタ 832

第 35 章 ROM/ フラッシュセキュリティ機能 837
35.1 ROM/ フラッシュセキュリティの概要 837
35.2 ROM/ フラッシュセキュリティの使用方法 838

第 36 章　シリアルプログラミングの接続例 841
36.1　シリアルプログラミング接続の基本設定 841
36.2　フラッシュマイクロコントローラをプログラミングするための PC の接続例 843
36.3　フラッシュマイクロコントローラをプログラミングするためのプログラミングツールの接続例 844

第 37 章　付録 845
37.1　MB96V300 の I/O マップ 845
37.1.1　MB96V300 の I/O マップ 845
37.2　命令 882
37.2.1　命令の種類 882
37.2.2　アドレス指定 883
37.2.3　直接アドレス指定 885
37.2.4　間接アドレス指定 891
37.2.5　実行サイクル数 897
37.2.6　実効アドレスフィールド 898
37.2.7　命令一覧の読み方 899
37.2.8　F^2MC-16FX 命令一覧 902
37.2.9　命令マップ 914

改訂履歴 936

1. 概要

MB96300スーパーシリーズはF^{2}MC-16FXマイコンのファミリメンバーに対応しています。多様なアプリケーションに対応するさまざまなマイコンシリーズで構成されます。スーパーシリーズのすべてのメンバーの間でプログラミングは共通です。

1.1 特長

F^{2}MC-16FX ファミリの MB96300スーパーシリーズは,自動車関連の用途向けに開発された16ビットマイクロコントローラです。

1.1.1 特長

- 16ビットのコア CPU (最大56 MHzのCPUクロック)を内蔵, 17.9 ns の命令サイクル時間
- 0.18 mm CMOS プロセステクノロジ
- 豊富なデータタイプ(ビット・バイト・ワード・ロングワードデータタイプ; 23種類のアドレッシングモード;バレルシフト;各種ポインタ)
- 8バイト命令実行キュー
- 符号付き乗算 (16 ビット ×16 ビット) および除算 (32 ビット/16 ビット) 命令を使用可能
- 内部電圧レギュレータにより内部 MCU 電圧を抑制でき, EMI および電力消費を小さくします
- コードセキュリティ機能
- 最大 5 つの FULL-CANインタフェース：Version 2.0 Part A および Part B に準拠, ISO16845 認定済み。
- 強力な割込み機能 (8プライオリティレベル：最大 16 の外部割込み)
- 高速割込み処理
- 最大 16 チャネルの DMA - CPU に依存しない自動転送機能。リソースに自由に割当て可能。
- 3 つの独立したクロックタイマ (23ビット RC クロックタイマ, 23ビットのメインクロックタイマ, 17ビットのサブクロックタイマ)
- ウォッチドッグタイマ
- 最大 10 チャネルの全二重 USART (SCI/LIN)
- USB機能(USBフルスピード対応),最大6エンドポイントのUSBホスト対応
- 最大 2 チャネルの I^{2}C, 400 kbps
- A/D コンバータ用の最大 40 チャネルのアナログ入力 (分解能 10 ビットまたは 8 ビット)
- 最大 6 チャネルの 16ビットリロードタイマ
- 最大 12 チャネルの16 ビットインプットキャプチャ (ICU)
- 最大 12 チャネルの 16 ビットアウトプットコンペア (OCU)
- 最大 4 チャネルの 16ビットフリーランタイマ
- 最大 20 チャネルの16ビットプログラマブルパルスジェネレータ

- 大電流ポート付き最大 6 チャネルのステッパモータコントローラ
- 最大 4 COM × 72 SEG, LCD コントローラ, 内部または外部LCD駆動用電圧の生成
- 組込みデバッグサポートの実装のためにメモリパッチ機能も使用可能
- 低消費電力モード
  (ラン, スリープ, タイマ, ストップなど13通りの動作モード)
- 3 MHz ～ 16 MHz 外部クロック, (高速クロック入力モード時 3.5 ～ 56 MHz), プログラム可能な乗算係数1 ～ 16 のオンチップ PLL
- 32 kHz のサブシステムクロック
- 100 kHz/2 MHz の内部 CR クロック
- 最大 6 チップのセレクト信号付き外部バスインタフェース, 8 ビットまたは 16 ビットデータ, 24 ビットアドレス, マルチプレックスまたはノンマルチプレックス,
  プログラマブルタイミング
- 最大 2 チャネルのアラームコンパレータ
- プログラマブルな入力レベル (AUTOMOTIVE / CMOSヒステリシス/ TTL)
- すべてのポートに対してプログラマブルなプルアップ抵抗
- EMI 最適化のためのプログラマブルな出力駆動力
- パッケージ : LQFP48 ピン / LQFP64 ピン/ LQFP80 ピン / LQFPまたはQFP 100 ピン / LQFP120 ピン/ LQFP144 ピン
- コントローラエリアネットワーク(CAN) - Robert Bosch GmbHのライセンス

1.2 スーパーシリーズ品種構成

Table 1-1 にMB96300 スーパーシリーズの概要を示します。

1.2.1 スーパーシリーズ品種構成

F^{2}MC-16FX MB96300 スーパーシリーズは ,Table 1-1 に示す F^{2}MC-16FX シリーズを含みます。

Table 1-1. MB96300スーパーシリーズ品種構成

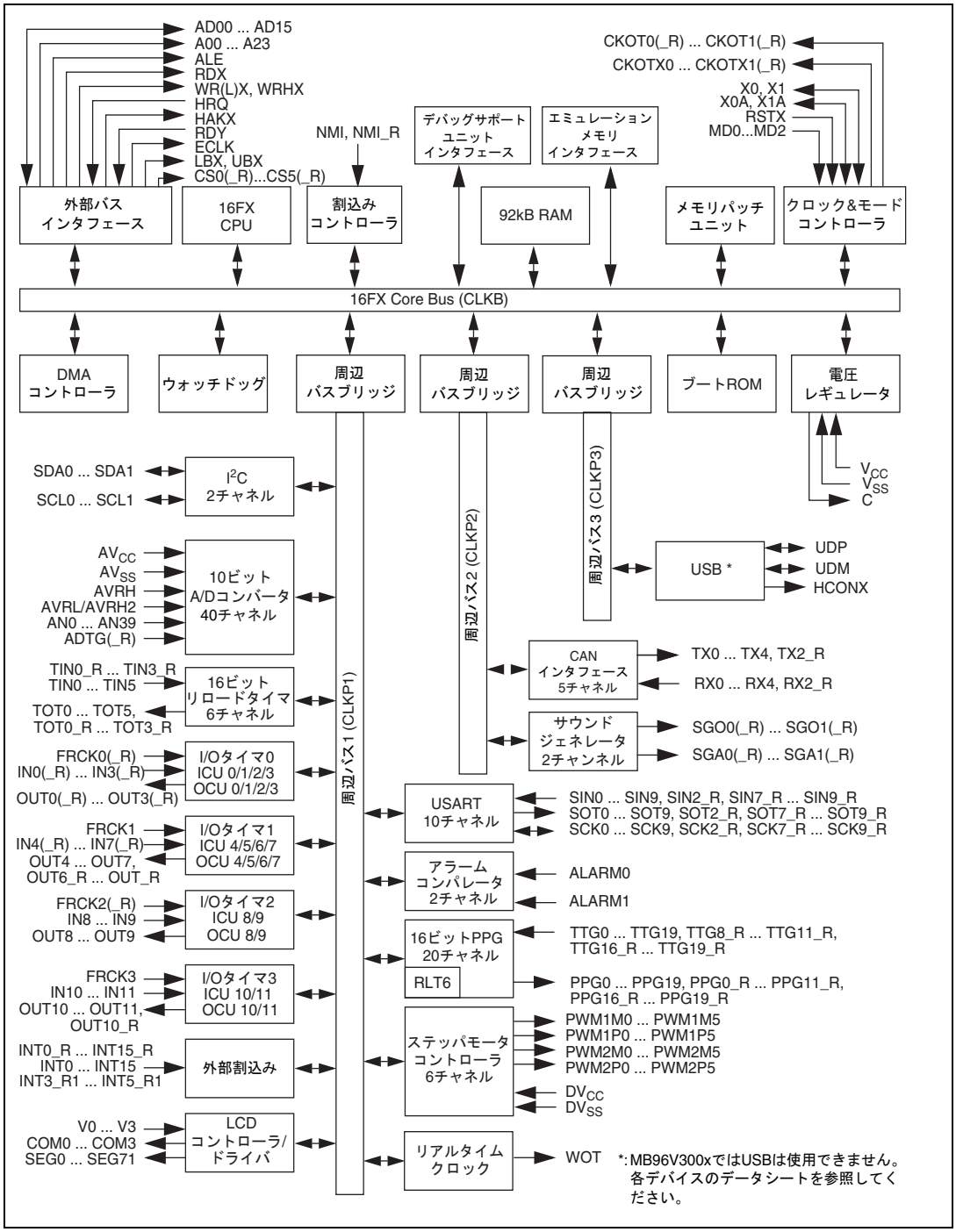
エバチップ	MB96V300x							
シリーズ	MB96310	MB96320	MB96330	MB96340	MB96350	MB96370	MB96380	MB96390
フラッシュ品	MB96F31x	MB96F32x	MB96F33x	MB96F34x	MB96F35x	MB96F37x	MB96F38x	MB96F39x
マスク ROM 品	-	-	-	MB9634x	-	-	MB9638x	-
パッケージ	FPT-48P-M26	FPT-80P-M21	FPT-144P-M08	FPT-100P-M20 FPT-100P-M22	FPT-64P-M23 FPT-64P-M24	FPT-144P-M08 FPT-144P-M12	FPT-120P-M21	FPT-100P-M20

1.3 MB96V300Bのブロックダイヤグラム

Figure 1-1 にMB96V300シリーズ のブロックダイヤグラムを示します。

1.3.1 MB96V300Bのブロックダイヤグラム

Figure 1-1. MB96V300Bのブロックダイヤグラム



1.4 本書の使用に関する一般的な注意事項

本書に関する一般的な注意事項を示します。

1.4.1 デバイス依存のリソース

マイクロコントローラシリーズの派生製品は , 異なるリソースのセット (例 , 特長 , チャネル数 , 相互接続) を含む場合があります。個別のデバイスに関する詳細は , 下記の文書に記載されています。

- リソースおよび使用可能なチャネル数: シリーズのデータシート
- 派生品種の中のリソースの特長 (例,リソース間の相互接続): スーパーマニュアルの「第1章 概要」

1.4.2 説明しているリソース

マイクロコントローラのリソースに関するすべての章 (USART, リロードタイマ , 汎用ポートなどに関する章) は , このリソースの 1 つのチャネルの機能だけを説明しています。同じ種類の他のすべてのリソースの動作は同じです。例外がある場合は ,「第 1 章 概要」か , 対応する章で説明しています。

ハードウェアマニュアルの各章では , リソースチャネル番号 (例 , レジスタ名) にプレースホルダ "n" を使用しています。このプレースホルダは , 使用するリソースのリソースチャネル番号に置き換えてください。

Table 1-2. 例：リロードタイマ0のレジスタ

「リロードタイマ」の章でのレジスタ名	ヘッダファイルでリロードタイマ 0 に定義されているレジスタ名	ヘッダファイルでリロードタイマ 1 に定義されているレジスタ名
TMCSRH**n**	TMCSRH**0**	TMCSRH**1**
TMCSRL**n**	TMRL**0**	TMRL**1**
TMRH**n**	TMRH**0**	TMRH**1**
TMRL**n**	TMRL**0**	TMRL**1**
TMRLRH**n**	TMRLRH**0**	TMRLRH**1**
TMRLRL**n**	TMRLRL**0**	TMRLRL**1**

1.5 16ビットI/Oタイマの構成

使用可能なフリーランタイマ,インプットキャプチャユニットおよびアウトプットコンペアユニットの数はデバイスによって異なります。複数のインプットキャプチャユニットおよびアウトプットコンペアユニットが1つのフリーランタイマに接続されます。これらのモジュールの間の関係を説明します。

1.5.1 インプットキャプチャユニットとフリーランタイマの接続

Table 1-3. インプットキャプチャユニットとフリーランタイマの接続

フリーランタイマ	インプットキャプチャユニット								
	MB96V300x	MB9631x	MB9632x	MB9633x	MB9634x	MB9635x	MB9637x	MB9638x	MB9639x
FRT 0	ICU 0/1/2/3	ICU 0/1	ICU 0/1/2/3			ICU 0/1	ICU 0/1/2/3		ICU 0/1
FRT 1	ICU 4/5/6/7	ICU 4/5/6	ICU 4/5/6/7						ICU 6/7
FRT 2	ICU 8/9	ICU 9	ICU 8/9		-	ICU 9	-		
FRT 3	ICU 10/11	ICU 10	ICU 10/11	-		ICU 10	-		

1.5.2 アウトプットコンペアユニットとフリーランタイマの接続

Table 1-4. アウトプットコンペアユニットとフリーランタイマの接続

フリーランタイマ	インプットキャプチャユニット								
	MB96V300x	MB9631x	MB9632x	MB9633x	MB9634x	MB9635x	MB9637x	MB9638x	MB9639x
FRT 0	OCU 0/1/2/3	-		OCU 0/1/2/3		-	OCU 0/1/2/3		
FRT 1	OCU 4/5/6/7	-	OCU 4/5/6/7				OCU 4/5	-	
FRT 2	OCU 8/9	-	-	OCU 8/9	-				
FRT 3	OCU 10/11	-	ICU 10/11	ICU 10/11	-				

1.5.3 タイマクリア機能とOCU間およびコンペアクリアとフリーランタイマ関係

Table 1-5. OCU, コンペアクリアおよびフリーランタイマのタイマクリアの関係

フリーランタイマ	タイマが下記のコンペアレジスタとの一致でクリアされます
FRT 0	OCCP0
FRT 1	OCCP4
FRT 2	OCCP8
FRT 3	OCCP10

OCCPx のコンペア機能は アウトプットコンペア制御ステータスレジスタ (OCS(2n)) の CST[1:0] ビットによって使用可能になります。

1.6 LIN-USARTのインプットキャプチャユニットのソースの選択

インプットキャプチャユニットのインプットソースは，外部端子 (INn) と LIN-USARTのシンクフィールドアウトプットのどちらかを選択できます。この章では　インプットキャプチャのインプットを選択する方法とどのLIN-USART をどの ICU に接続できるかを説明します。

1.6.1 LIN-USARTとICUの接続の概要

Table 1-6. LIN-USARTとICUの接続

LIN-USART	ICU	ICUのフリーランタイマ	ICUソース選択ビット
USART0	ICU0	FRT0	ICE01:ICUS0
USART1	ICU1		ICE01:ICUS1
USART2	ICU6	FRT1	ICE67:ICUS6
USART3	ICU7		ICE67:ICUS7
USART4	ICU4		ICE45:ICUS4
USART5	ICU5		ICE45:ICUS5
USART6	ICU8	FRT2	ICE89:ICUS8
USART7	ICU9		ICE89:ICUS9
USART8	ICU10	FRT3	ICE1011:ICUS10
USART9	ICU11		ICE1011:ICUS11

＜注意事項＞

USART, インプットキャプチャユニットおよびフリーランタイマのチャネルの搭載可否については , データシートを参照してください。

1.6.2 インプットキャプチャユニットソースの選択

ICU のソース選択は , インプットキャプチャエッジレジスタ ICExy によって行います。

Figure 1-2. インプットキャプチャエッジレジスタ(ICE01)

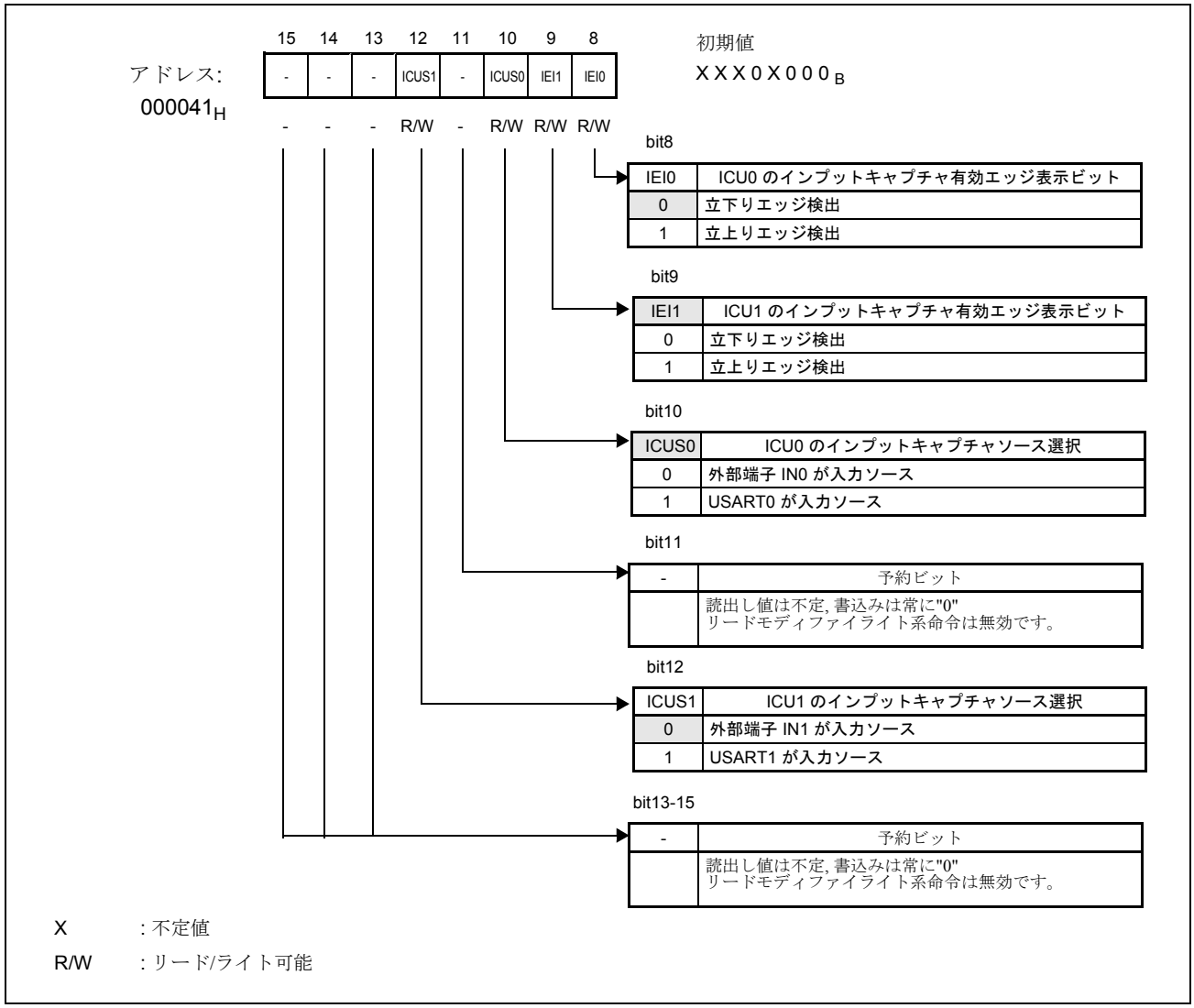


Figure 1-3. インプットキャプチャエッジレジスタ (ICE45)

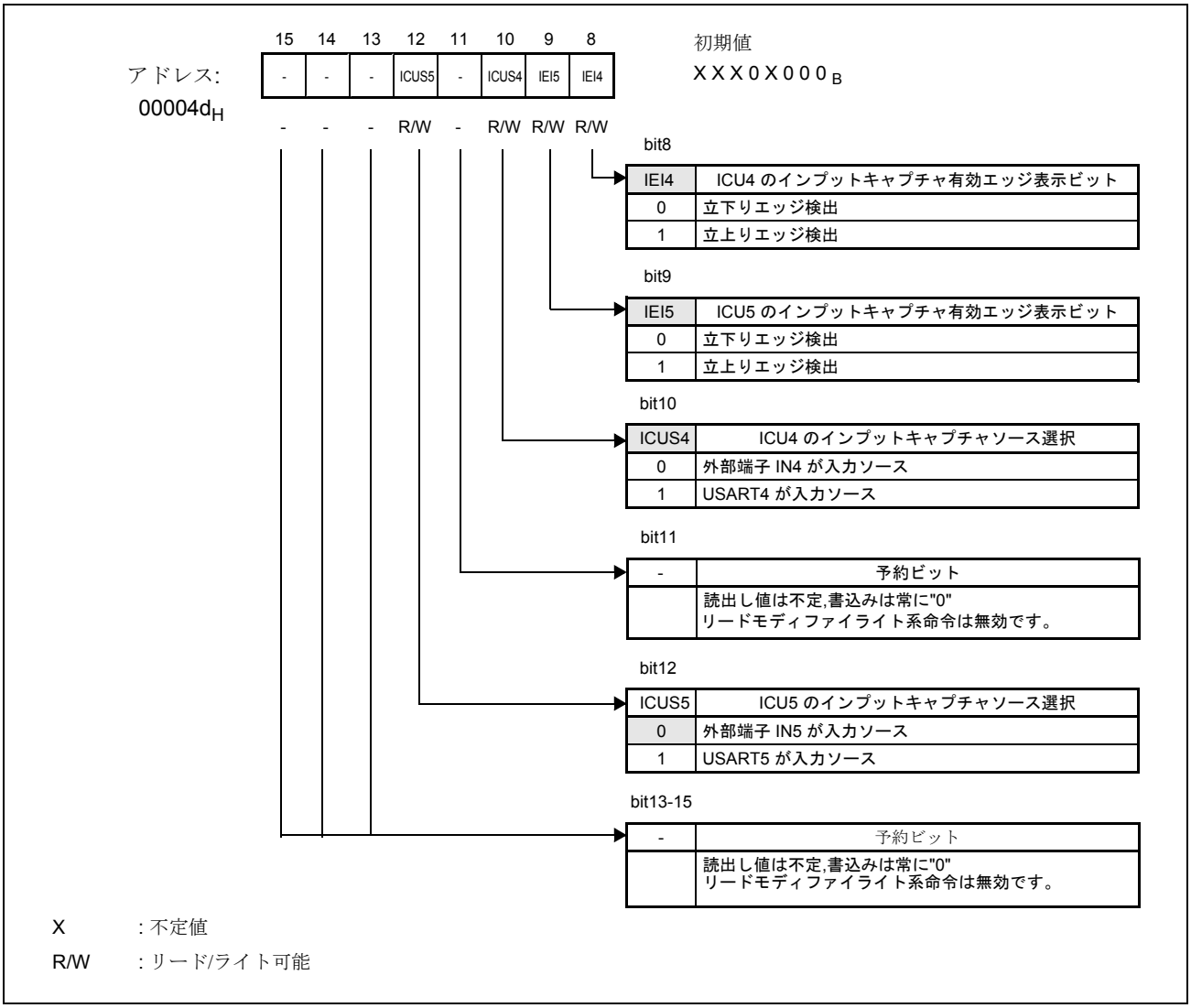


Figure 1-4. インプットキャプチャエッジレジスタ (ICE67)

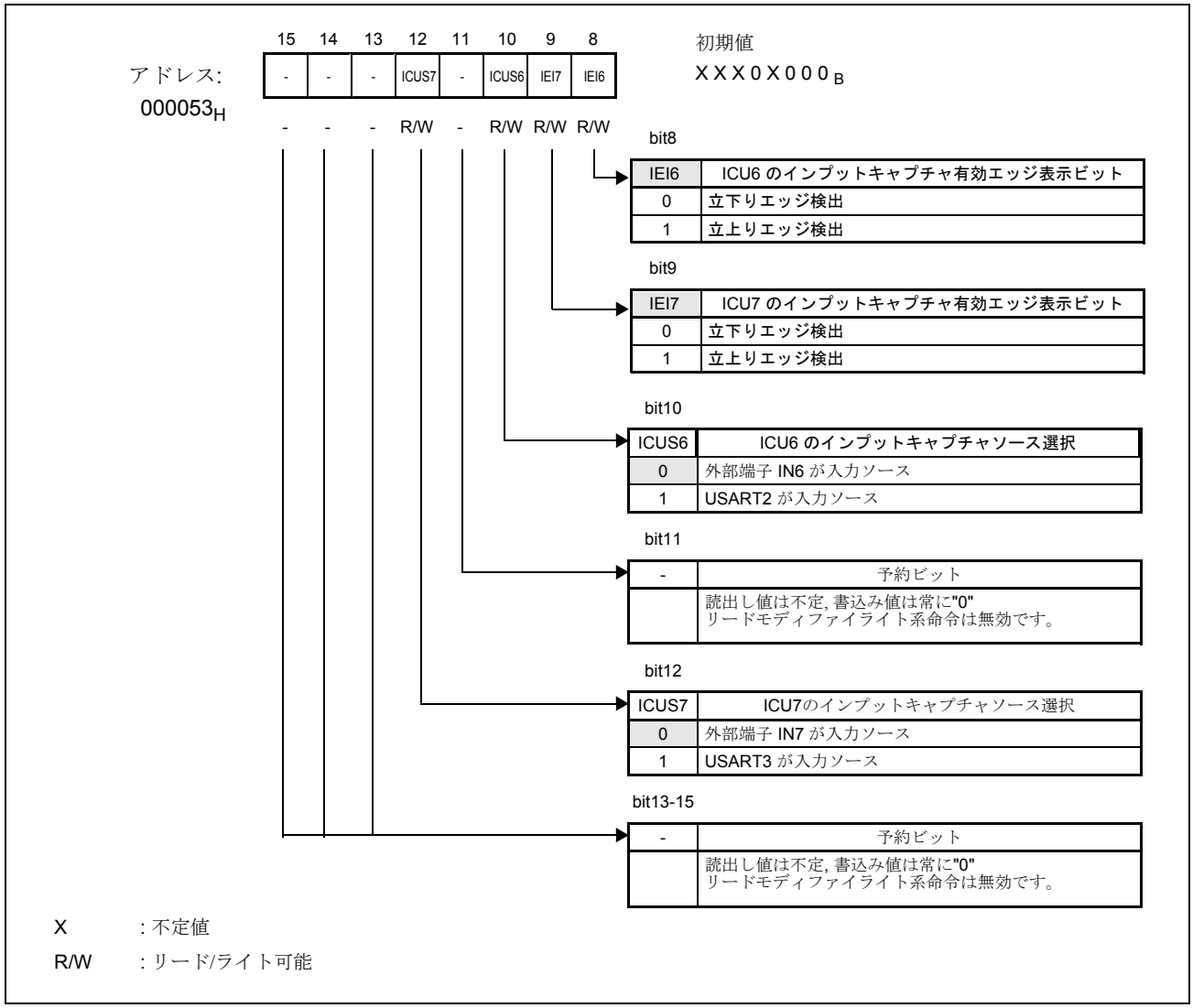


Figure 1-5. インプットキャプチャエッジレジスタ (ICE89)

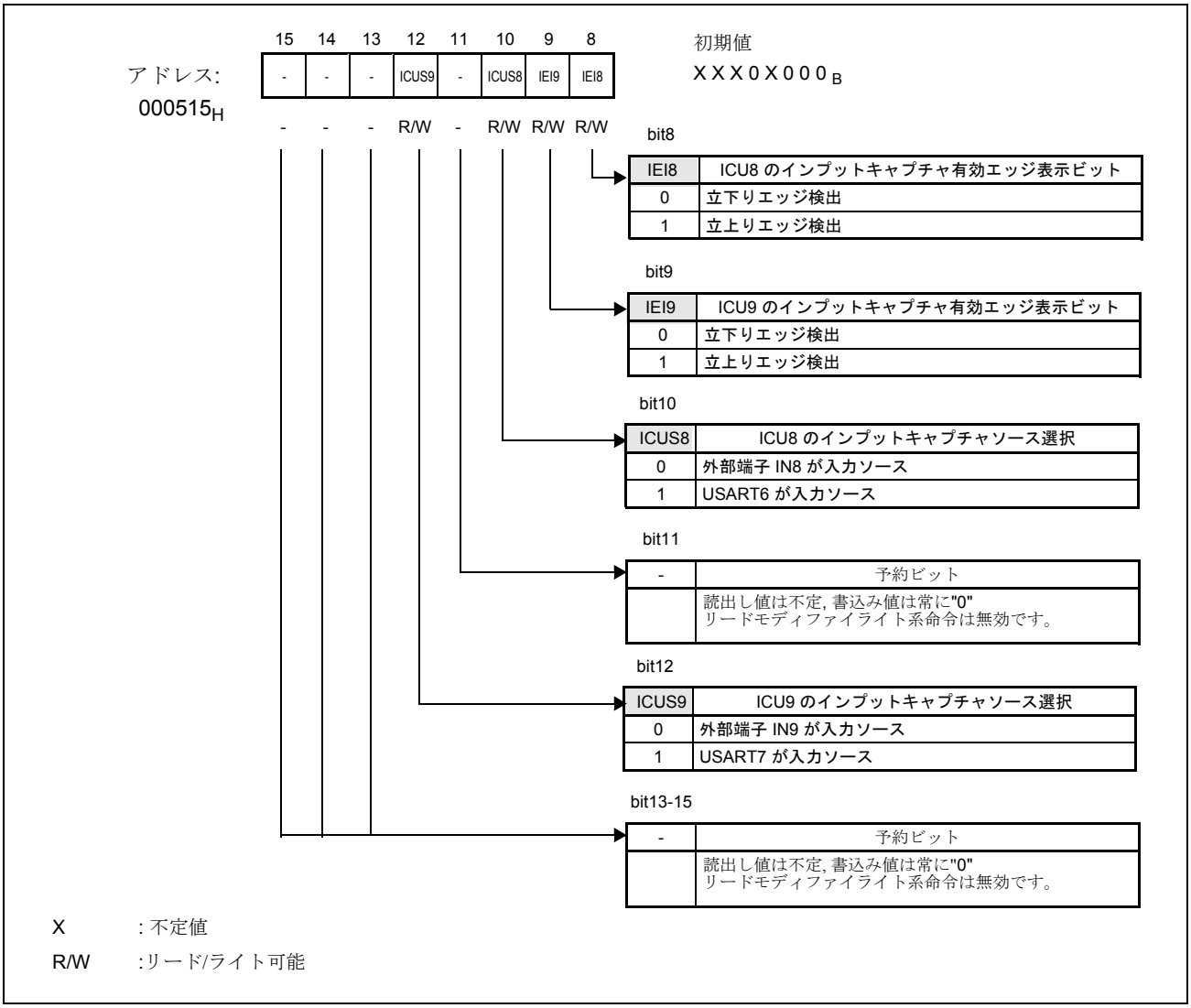
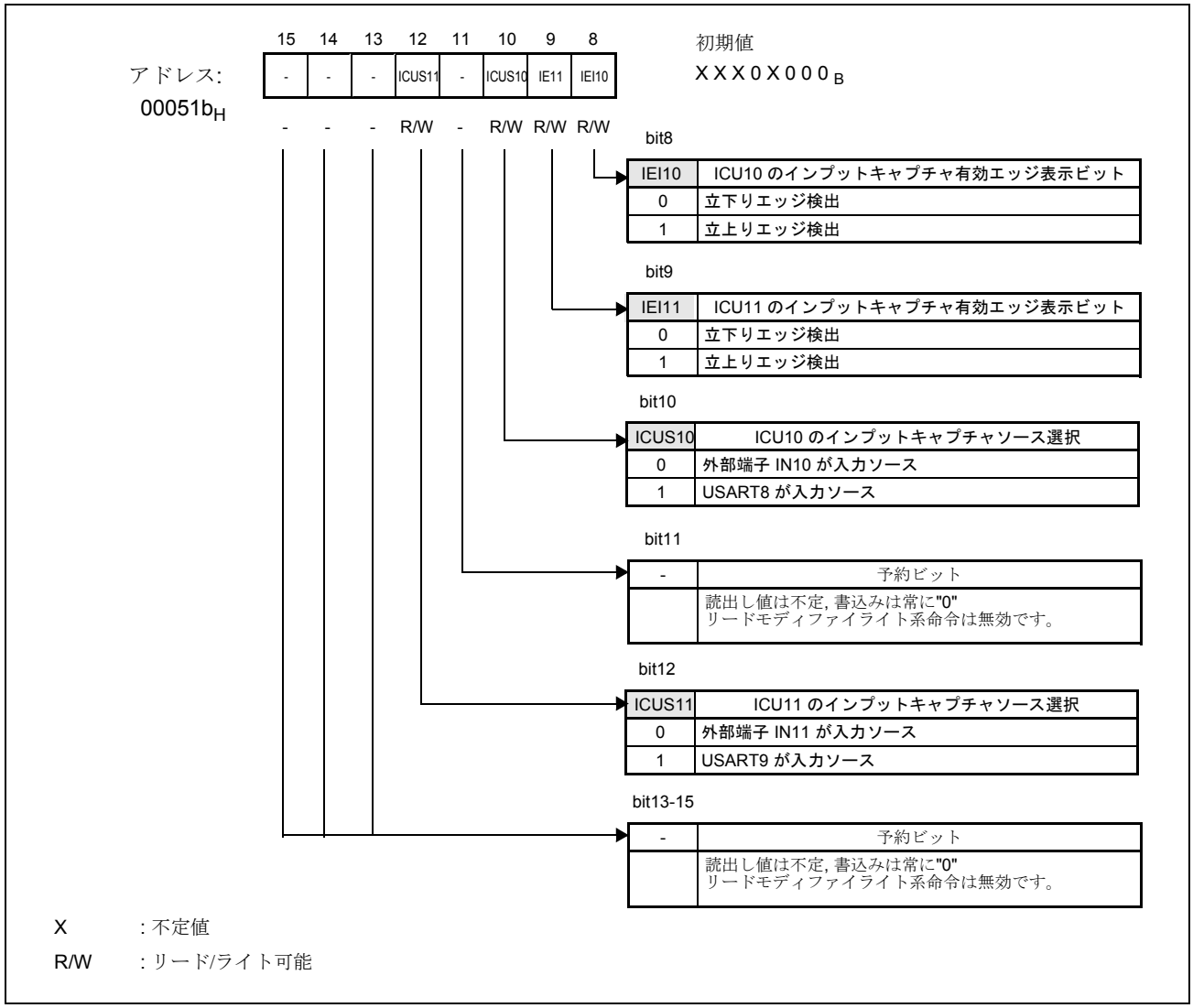


Figure 1-6. インプットキャプチャエッジレジスタ (ICE1011)



1.7 周辺リソース端子のリロケーション

一部のリソースの入力または出力端子をリロケートできます。これらのリソース端子のロケーションは,周辺リソース端子リロケーションレジスタPRRR0～PRRR13によって定義されます。

1.7.1 周辺リソース端子リロケーションレジスタの概要

Table 1-7. 周辺リソース端子リロケーションレジスタ (PRRR0 ～ PRRR13)

アドレス	レジスタ名	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	説明
0004d6	PRRR0	INT7_R	INT6_R	INT5_R	INT4_R	INT3_R	INT2_R	INT1_R	INT0_R	周辺リソース端子リロケーションレジスタ 0
0004d7	PRRR1	INT15_R	INT14_R	INT13_R	INT12_R	INT11_R	INT10_R	INT9_R	INT8_R	周辺リソース端子リロケーションレジスタ 1
0004d8	PRRR2	PPG7_R	PPG6_R	PPG5_R	PPG4_R	PPG3_R	PPG2_R	PPG1_R	PPG0_R	周辺リソース端子リロケーションレジスタ 2
0004d9	PRRR3	TOT3_R	TIN3_R	TOT2_R	TIN2_R	TOT1_R	TIN1_R	TOT0_R	TIN0_R	周辺リソース端子リロケーションレジスタ 3
0004da	PRRR4	IN7_R	IN6_R	IN5_R	IN4_R	IN3_R	IN2_R	IN1_R	IN0_R	周辺リソース端子リロケーションレジスタ 4
0004db	PRRR5	OUT7_R	OUT6_R	-	-	OUT3_R	OUT2_R	OUT1_R	OUT0_R	周辺リソース端子リロケーションレジスタ 5
0004dc	PRRR6	CKOTX1_R	CKOT1_R	SCK2_R	SOT2_R	SIN2_R	FRCK0_R	SGA0_R	SGO0_R	周辺リソース端子リロケーションレジスタ 6
0004dd	PRRR7	TX2_R	RX2_R	INT5_R1	INT4_R1	INT3_R1	CS3_R	NMI_R	ADTG_R	周辺リソース端子リロケーションレジスタ 7
0004de	PRRR8	SOT9_R	SIN9_R	SCK8_R	SOT8_R	SIN8_R	SCK7_R	SOT7_R	SIN7_R	周辺リソース端子リロケーションレジスタ 8
0004df	PRRR9	-	-	CKOT0_R	OUT10_R	FRCK2_R	SGA1_R	SGO1_R	SCK9_R	周辺リソース端子リロケーションレジスタ 9
000660	PRRR10	TTG11_R	TTG10_R	TTG9_R	TTG8_R	PPG11_R	PPG10_R	PPG9_R	PPG8_R	周辺リソース端子リロケーションレジスタ 10
000661	PRRR11	TTG19_R	TTG18_R	TTG17_R	TTG16_R	PPG19_R	PPG18_R	PPG17_R	PPG16_R	周辺リソース端子リロケーションレジスタ 11
000662	PRRR12	-	-	-	CS5_R	CS4_R	CS2_R	CS1_R	CS0_R	周辺リソース端子リロケーションレジスタ 12
000663	PRRR13	-	-	-	-	-	-	-	-	周辺リソース端子リロケーションレジスタ 13

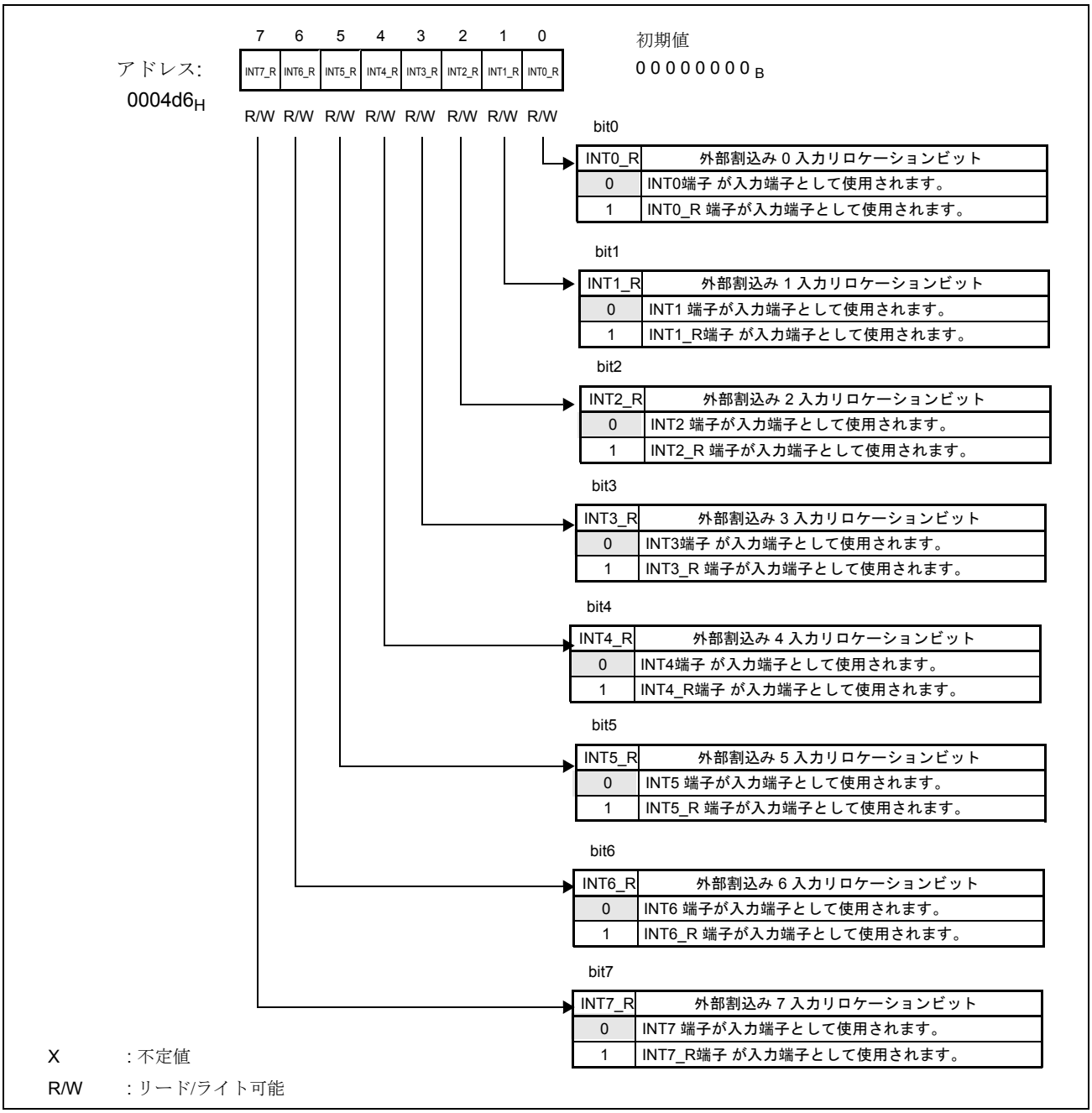
リソースリロケーション端子の使用可否

リソースリロケーション端子がない場合 , 対応するビットは未定義ビットとして扱わなければなりません。この場合対応するビットには常に "0" を書き込むことを推奨します。またこの対応するビットのリード値は不定ですから , レジスタへのリードモディファイライト系命令は使用できません。

リソースリロケーション端子の使用可否についてはデータシートを参照してください。

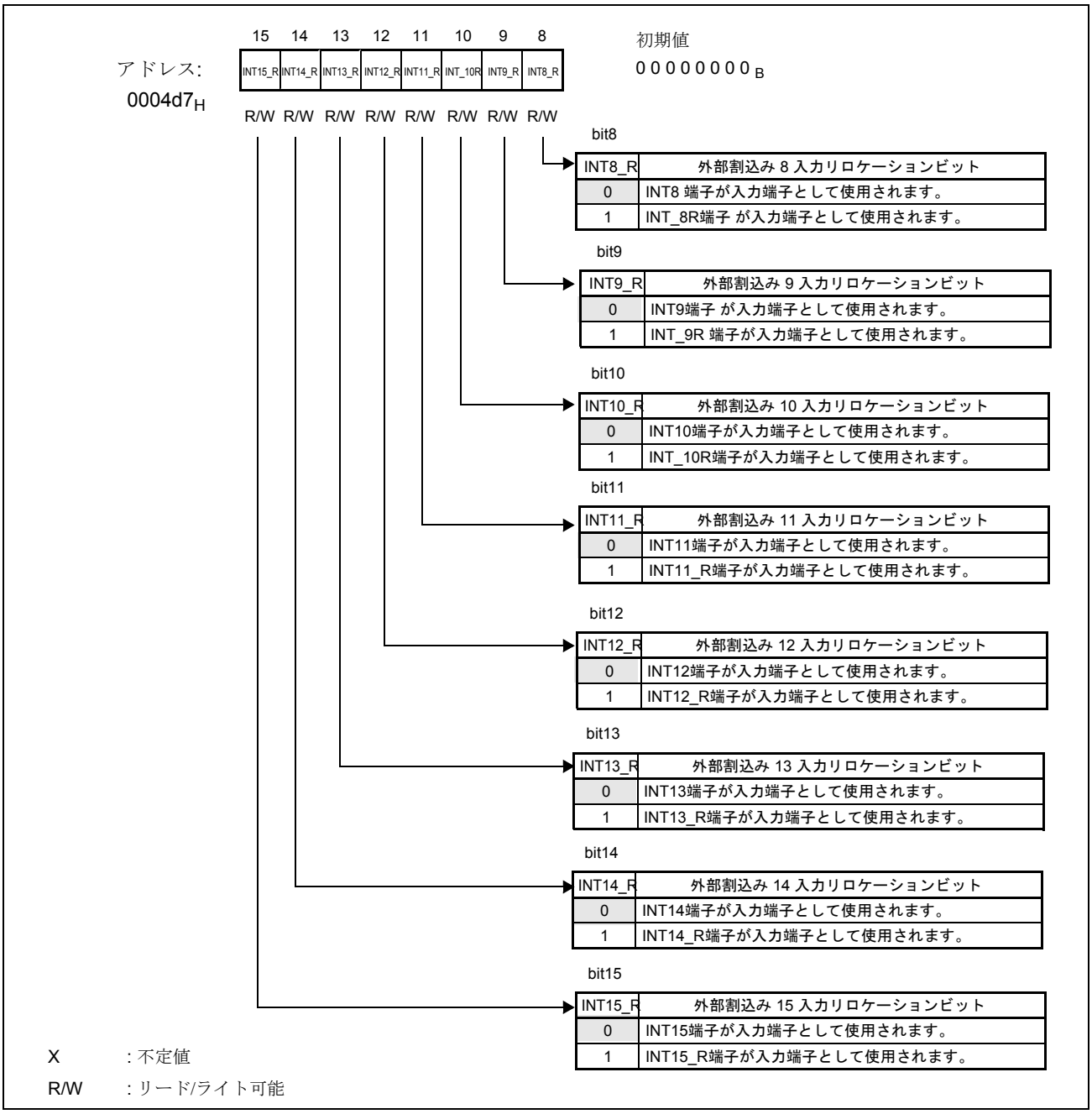
1.7.2 周辺リソース端子リロケーションレジスタ 0 (PRRR0)

Figure 1-7. 周辺リソース端子リロケーションレジスタ 0 (PRRR0)



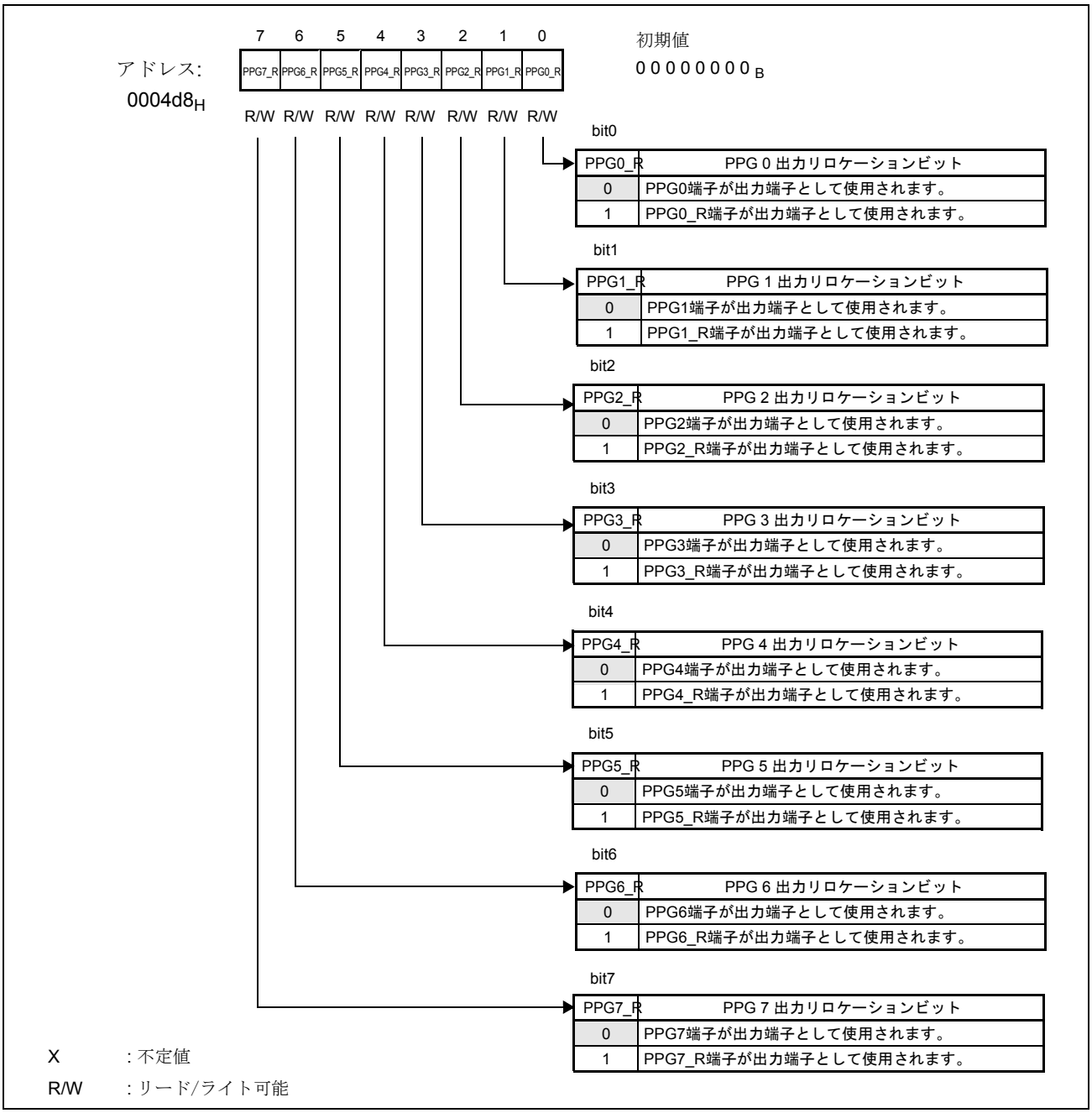
1.7.3 周辺リソース端子リロケーションレジスタ 1 (PRRR1)

Figure 1-8. 周辺リソース端子リロケーションレジスタ 1 (PRRR1)



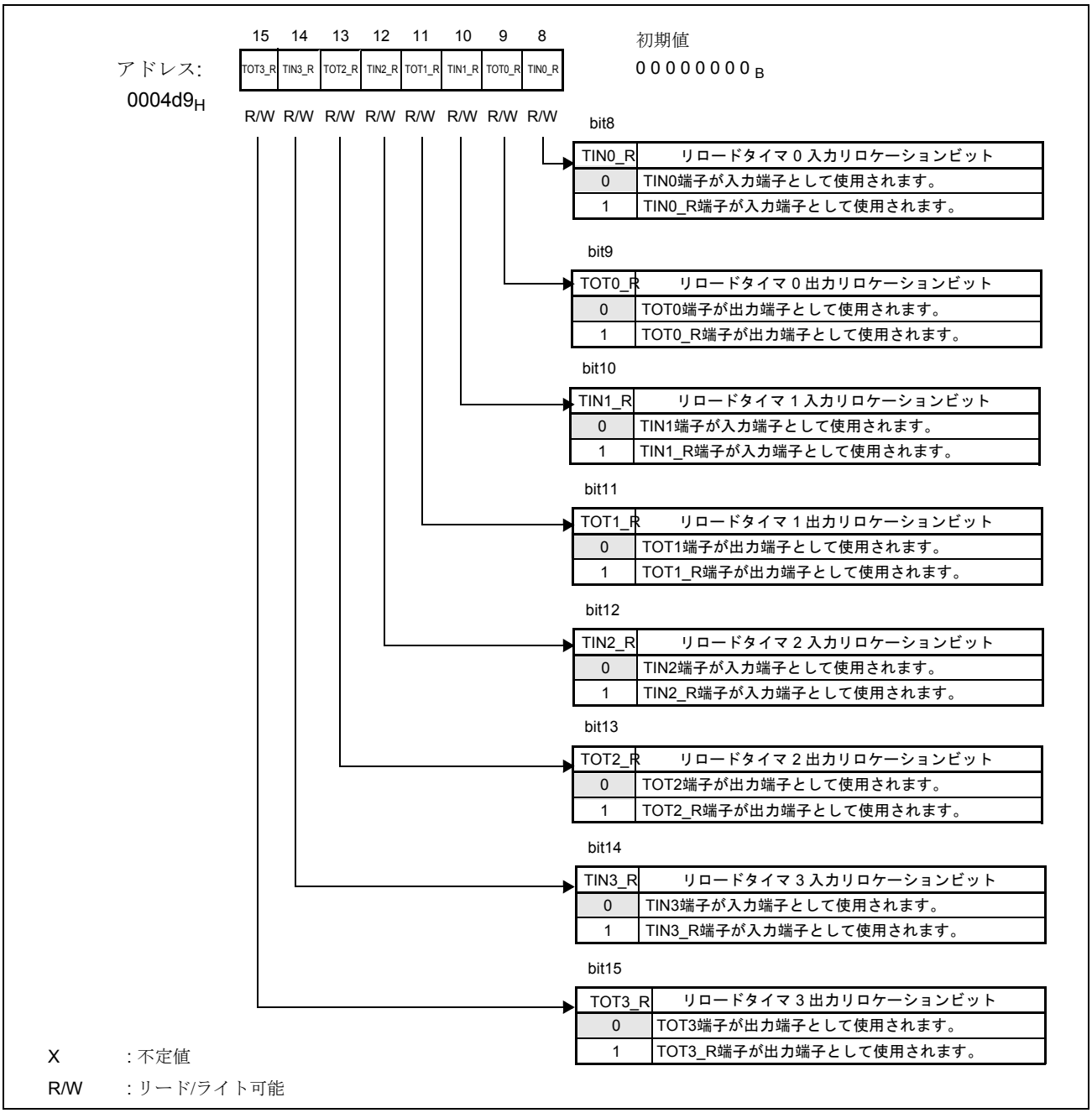
1.7.4 周辺リソース端子リロケーションレジスタ 2 (PRRR2)

Figure 1-9. 周辺リソース端子リロケーションレジスタ 2 (PRRR2)



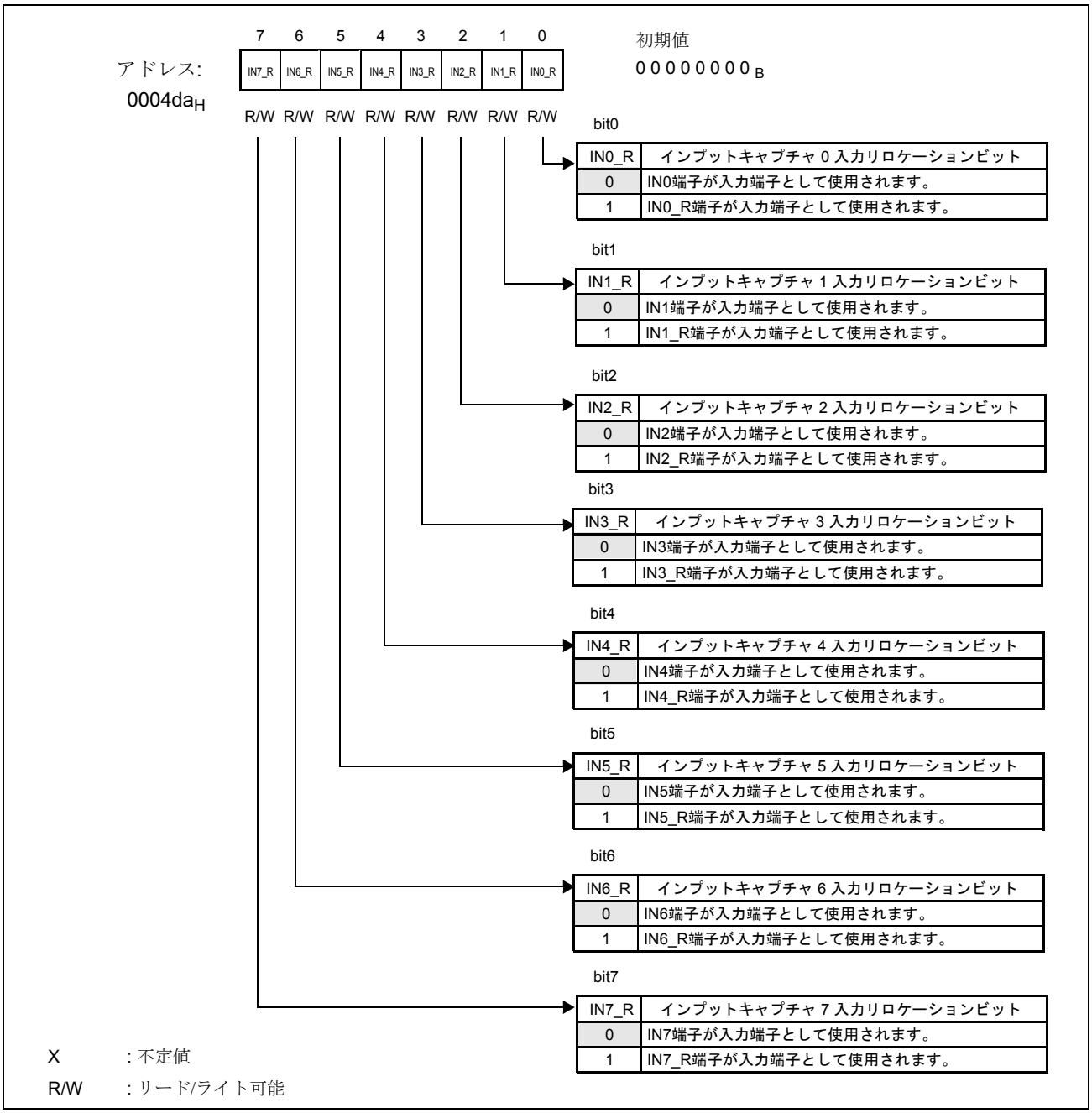
1.7.5 周辺リソース端子リロケーションレジスタ 3 (PRRR3)

Figure 1-10. 周辺リソース端子リロケーションレジスタ 3 (PRRR3)



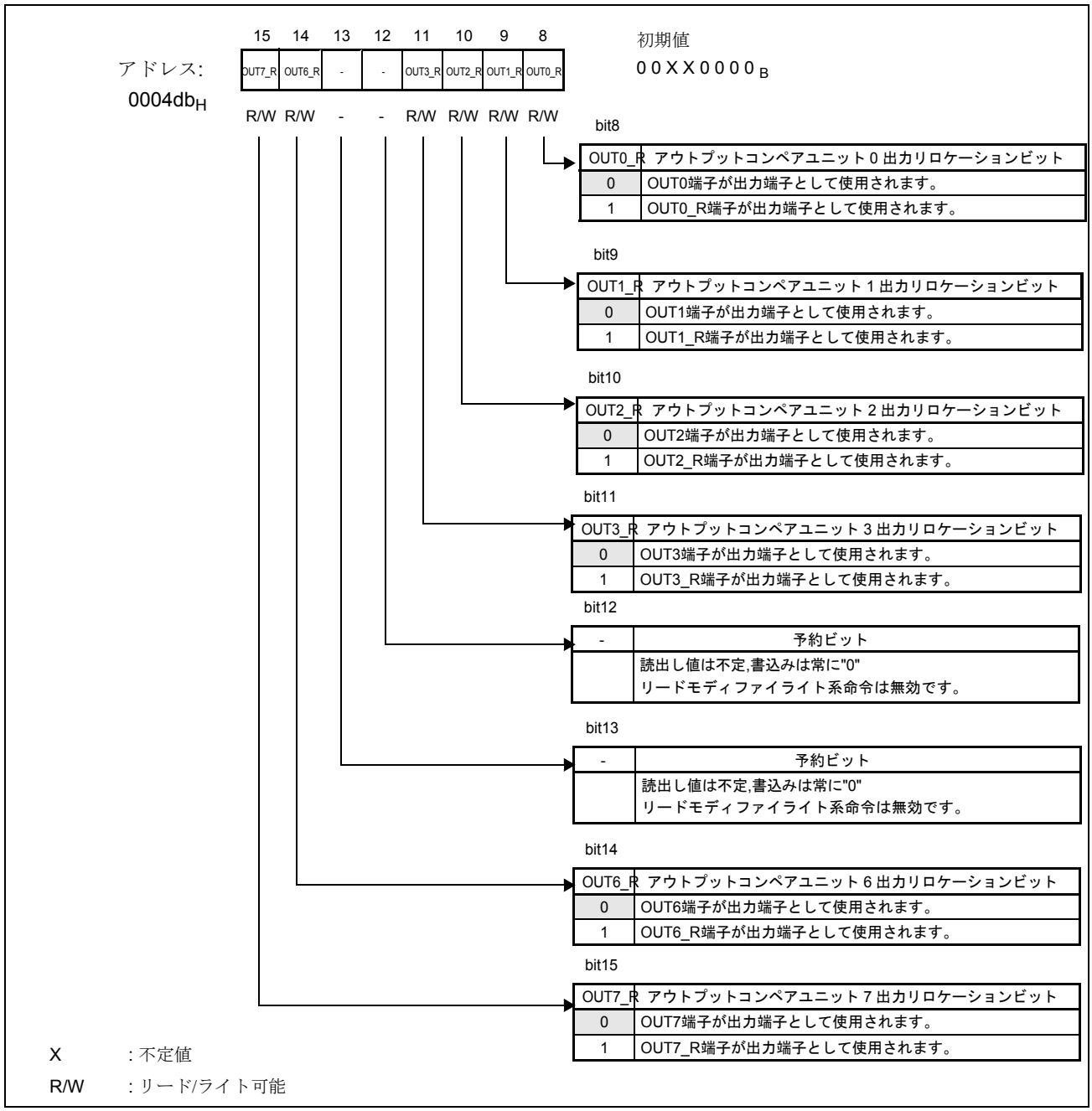
1.7.6 周辺リソース端子リロケーションレジスタ 4 (PRRR4)

Figure 1-11. 周辺リソース端子リロケーションレジスタ 4 (PRRR4)



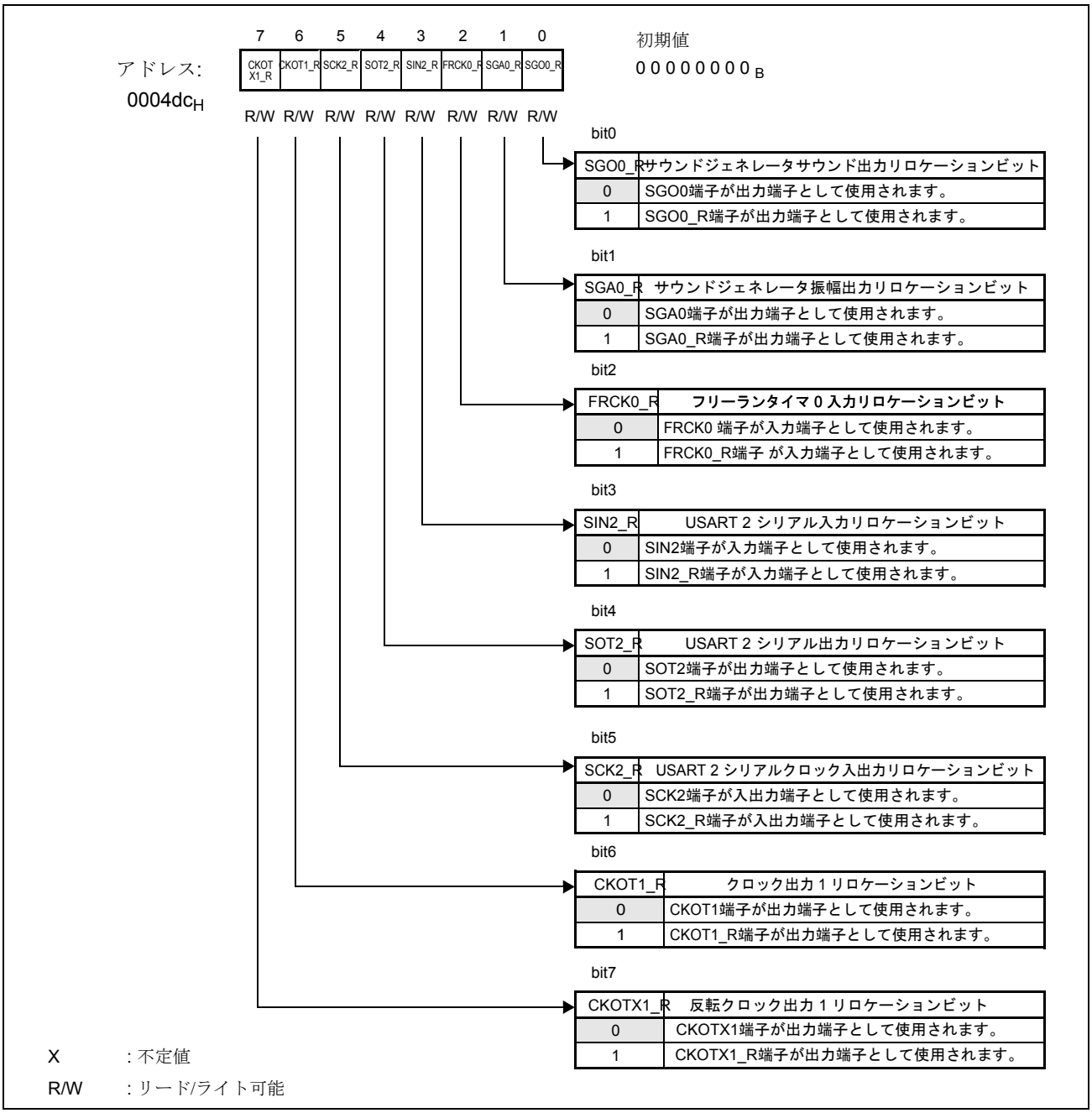
1.7.7 周辺リソース端子リロケーションレジスタ 5 (PRRR5)

Figure 1-12. 周辺リソース端子リロケーションレジスタ 5 (PRRR5)



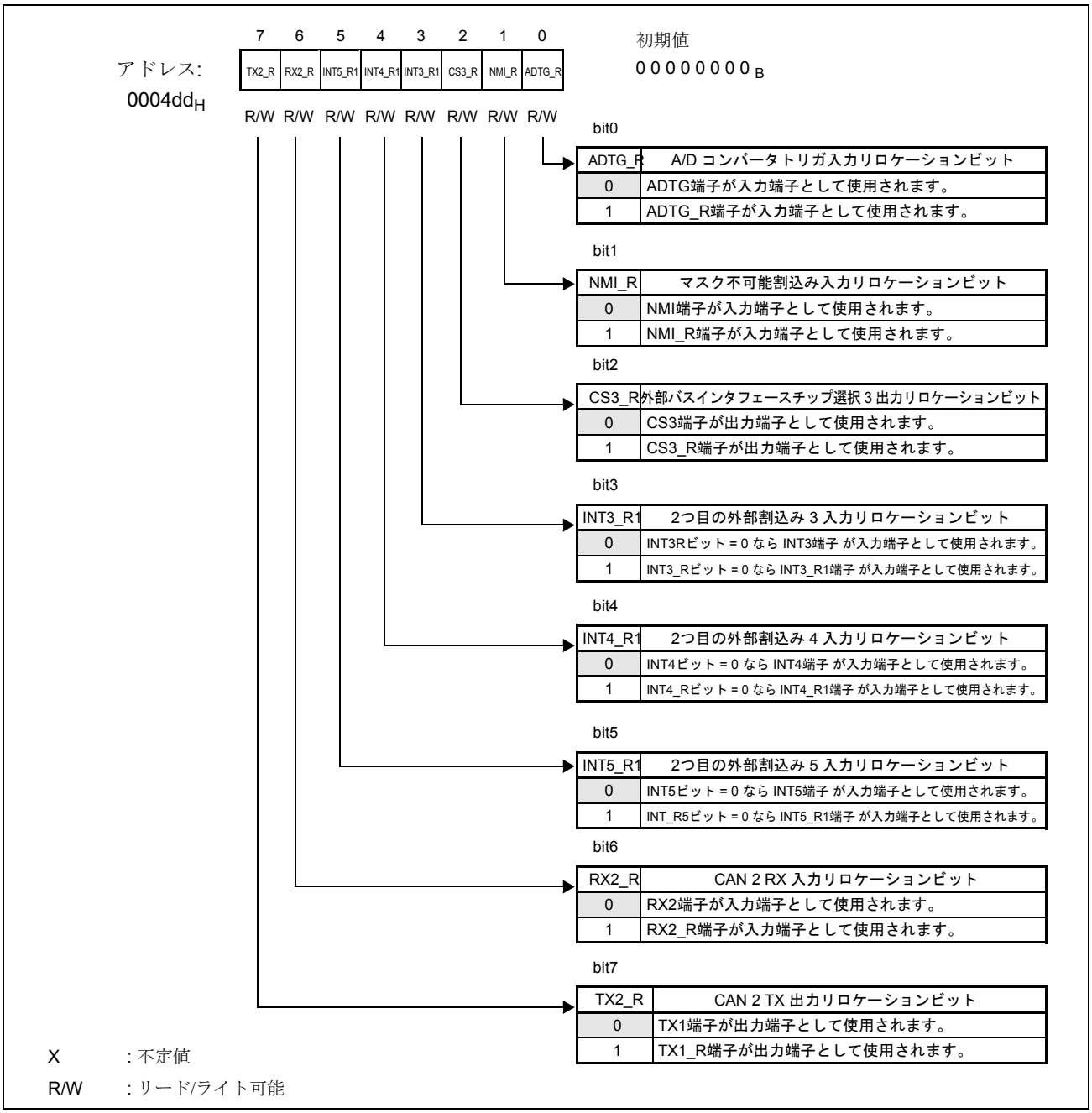
1.7.8 周辺リソース端子リロケーションレジスタ 6 (PRRR6)

Figure 1-13. 周辺リソース端子リロケーションレジスタ 6 (PRRR6)



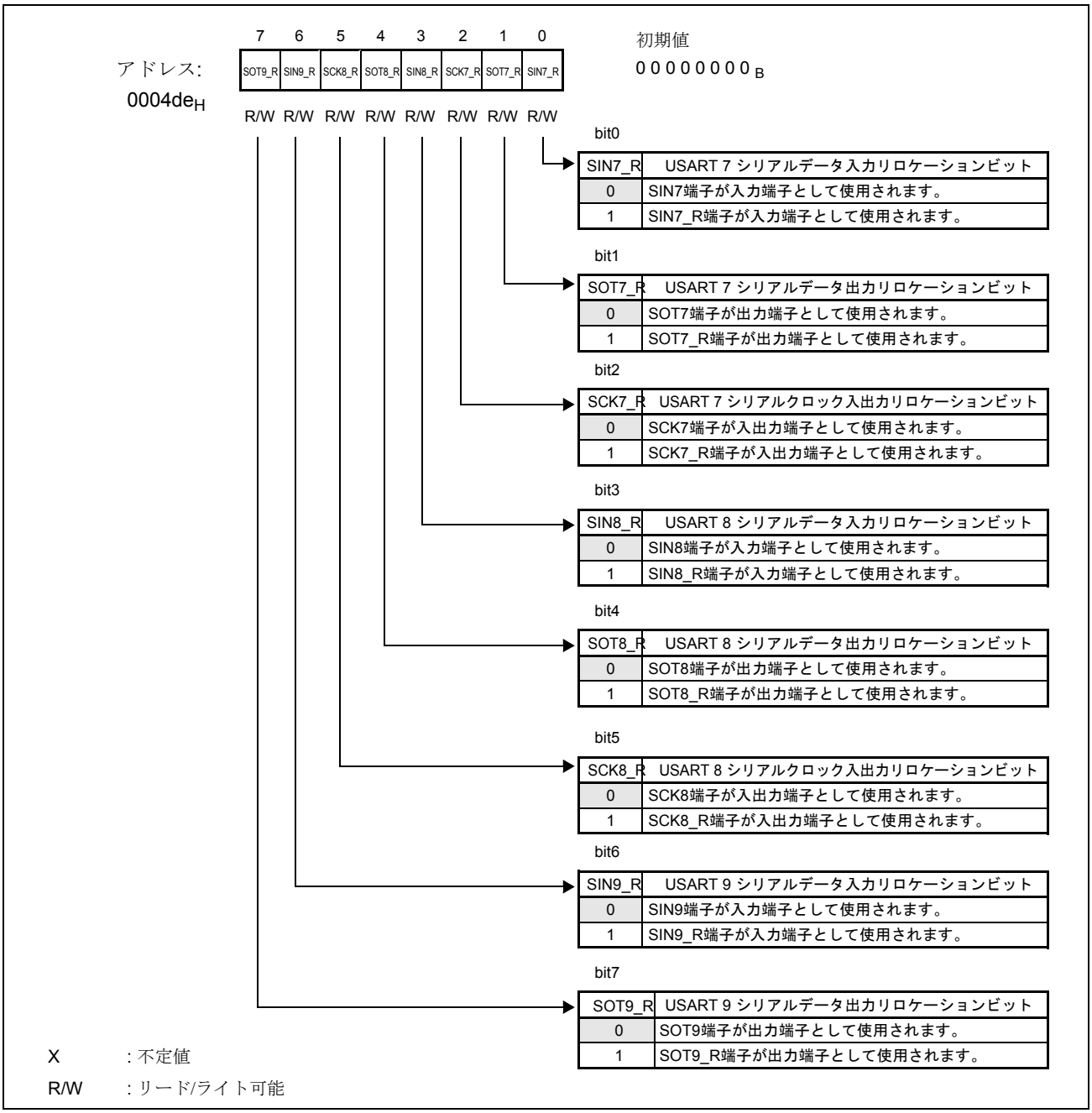
1.7.9 周辺リソース端子リロケーションレジスタ 7 (PRRR7)

Figure 1-14. 周辺リソース端子リロケーションレジスタ 7 (PRRR7)



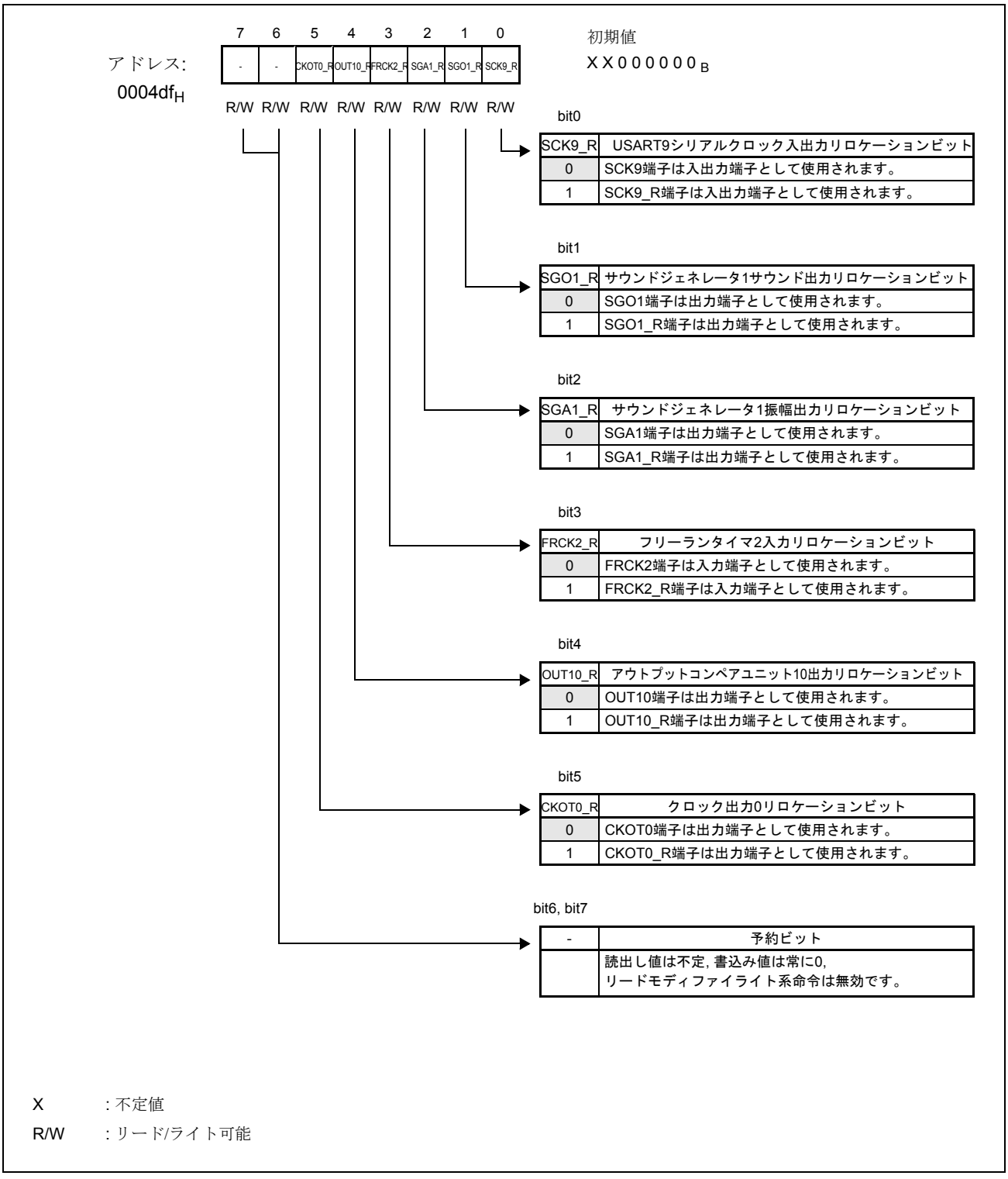
1.7.10 周辺リソース端子リロケーションレジスタ 8 (PRRR8)

Figure 1-15. 周辺リソース端子リロケーションレジスタ 8 (PRRR8)



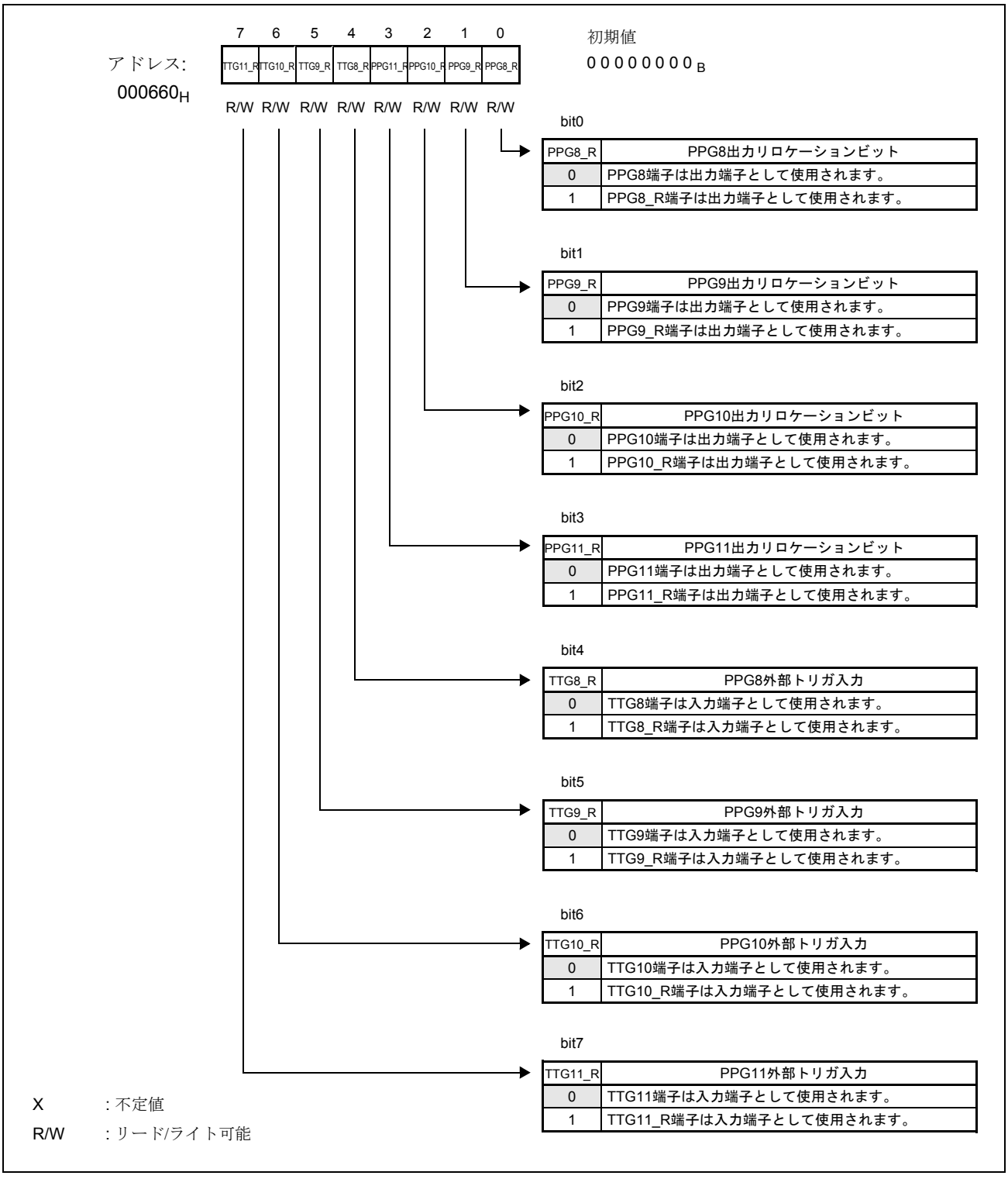
1.7.11 周辺リソース端子リロケーションレジスタ 9 (PRRR9)

Figure 1-16. 周辺リソース端子リロケーションレジスタ 9 (PRRR9)



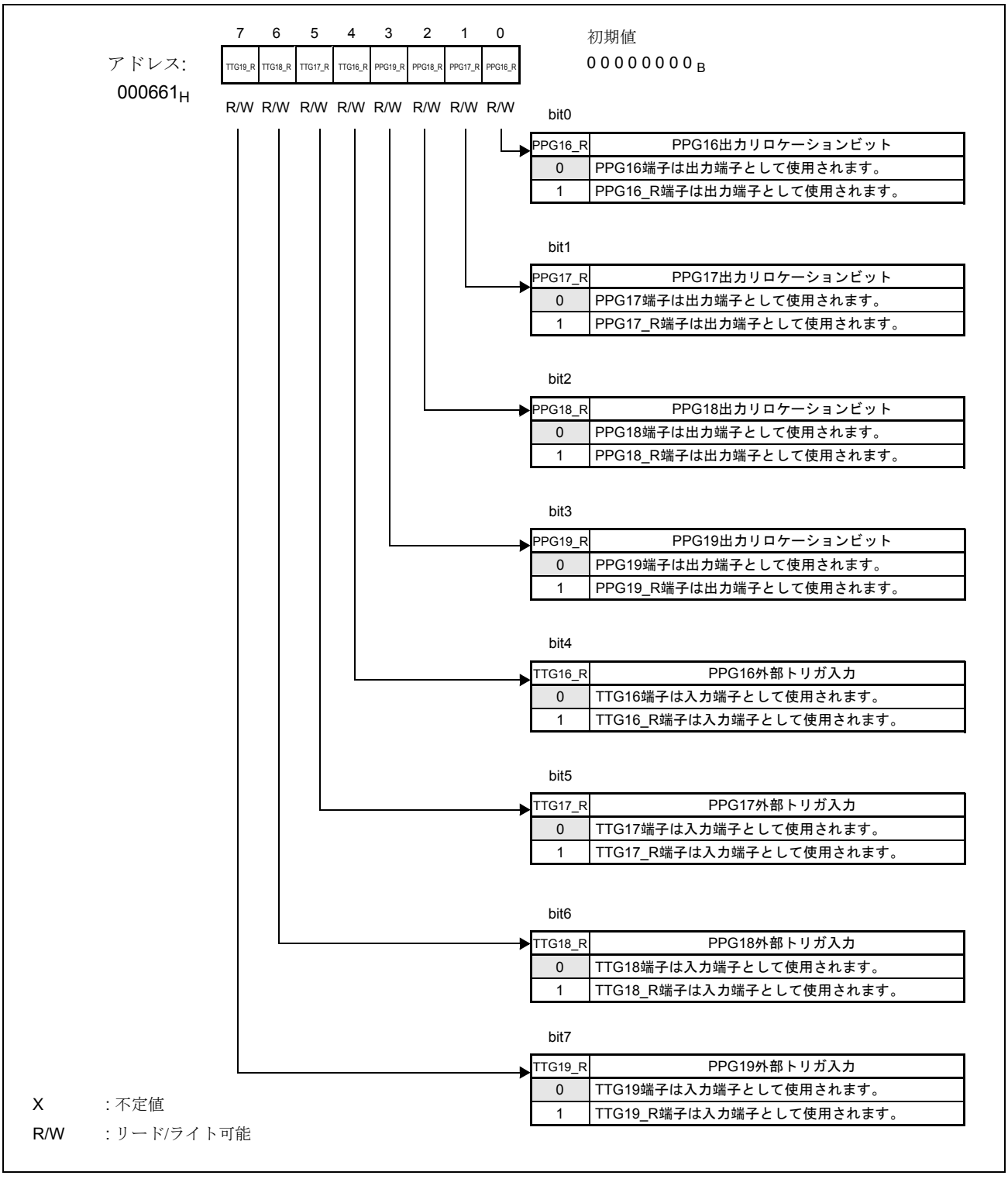
1.7.12 周辺リソース端子リロケーションレジスタ 10 (PRRR10)

Figure 1-17. 周辺リソース端子リロケーションレジスタ 10 (PRRR10)



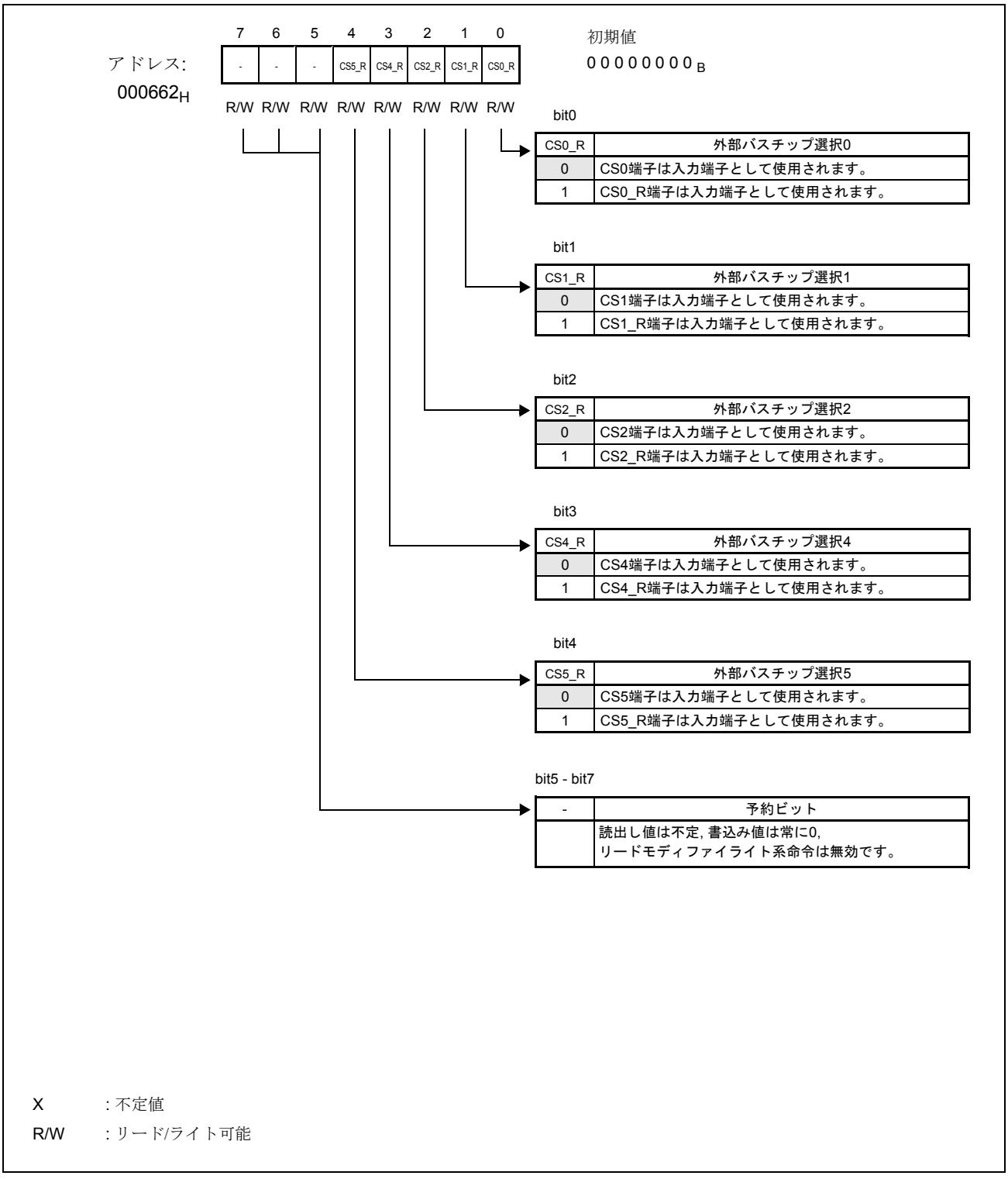
1.7.13 周辺リソース端子リロケーションレジスタ 11 (PRRR11)

Figure 1-18. 周辺リソース端子リロケーションレジスタ 11 (PRRR11)



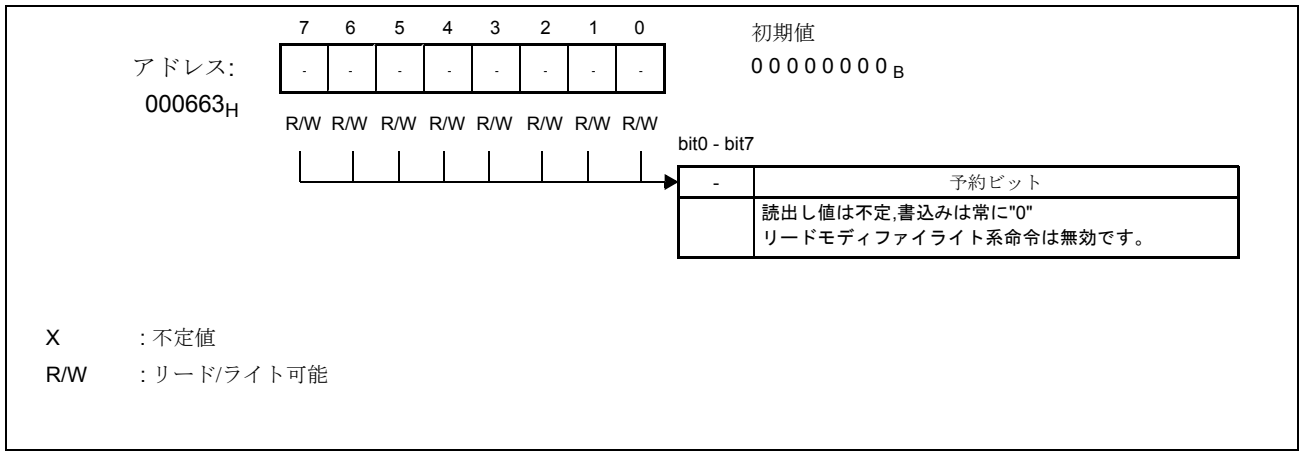
1.7.14 周辺リソース端子リロケーションレジスタ 12 (PRRR12)

Figure 1-19. 周辺リソース端子リロケーションレジスタ 12 (PRRR12)



1.7.15 周辺リソース端子リロケーションレジスタ 13 (PRRR13)

Figure 1-20. 周辺リソース端子リロケーションレジスタ 13 (PRRR13)



2. CPU

CPUについて説明します。

2.1 CPUの概要

F^{2}MC-16FX CPUコアは,民生用・車載用機器などの高速リアルタイム処理が要求される用途向けに設計された16ビットCPUです。F^{2}MC-16FXの命令体系はコントローラ用途向けに設計されており, 高速・高効率な制御処理が可能です。

2.1.1 CPUの概要

F^{2}MC-16FX CPU コアは , 16 ビットデータ処理はもちろん , 内部に 32 ビットアキュムレータを搭載しているため , 32 ビットデータ処理も可能です。ただし , 32 ビットデータが処理可能なのは一部の命令です。メモリ空間は最大 16M バイトで , リニア方式およびバンク方式のいずれかにてアクセス可能です。命令体系は , F^{2}MC-16LX と互換性があります。また , 高水準言語に対応しており , アドレッシングモード , 乗除算命令 , およびビット処理が充実しています。次に , F^{2}MC-16FX CPU の特長を示します。

- 高速実行
  - 最小命令実行時間16 ns (64 MHz動作時: 4 MHz発振, 16逓倍)
  - 基本命令は1サイクルで実行
  - 5段階パイプラインによる高速処理
  - 8バイトの命令キュー
- **汎用レジスタ: 32バンク×8ワード×16ビット**
- メモリ空間: 16Mバイト, リニアまたはバンク方式にてアクセス
- コントローラ用途向けに最適化された命令体系
  - 高いコード効率
  - 豊富なデータタイプ:ビット, バイト, ワード, ロングワード
  - 拡張アドレッシングモード: 23種類
  - 32ビットアキュムレータによる高精度演算(32ビット長)
  - 符号付きおよび符号なし乗除算命令
- 強力な割込み機能
  - 高速応答速度(約10クロックサイクル(CLKB))
  - 8つのプライオリティレベル(プログラマブル)
  - マスク不可割込み(NMI)
  - DMA転送により, CPUを使わずに割込み要求を処理可能(最大16チャネル)

- 高水準言語(C言語)/マルチタスク処理に対応した命令体系
  - システムスタックポインタ
  - バランスのとれた命令体系
  - シフト命令

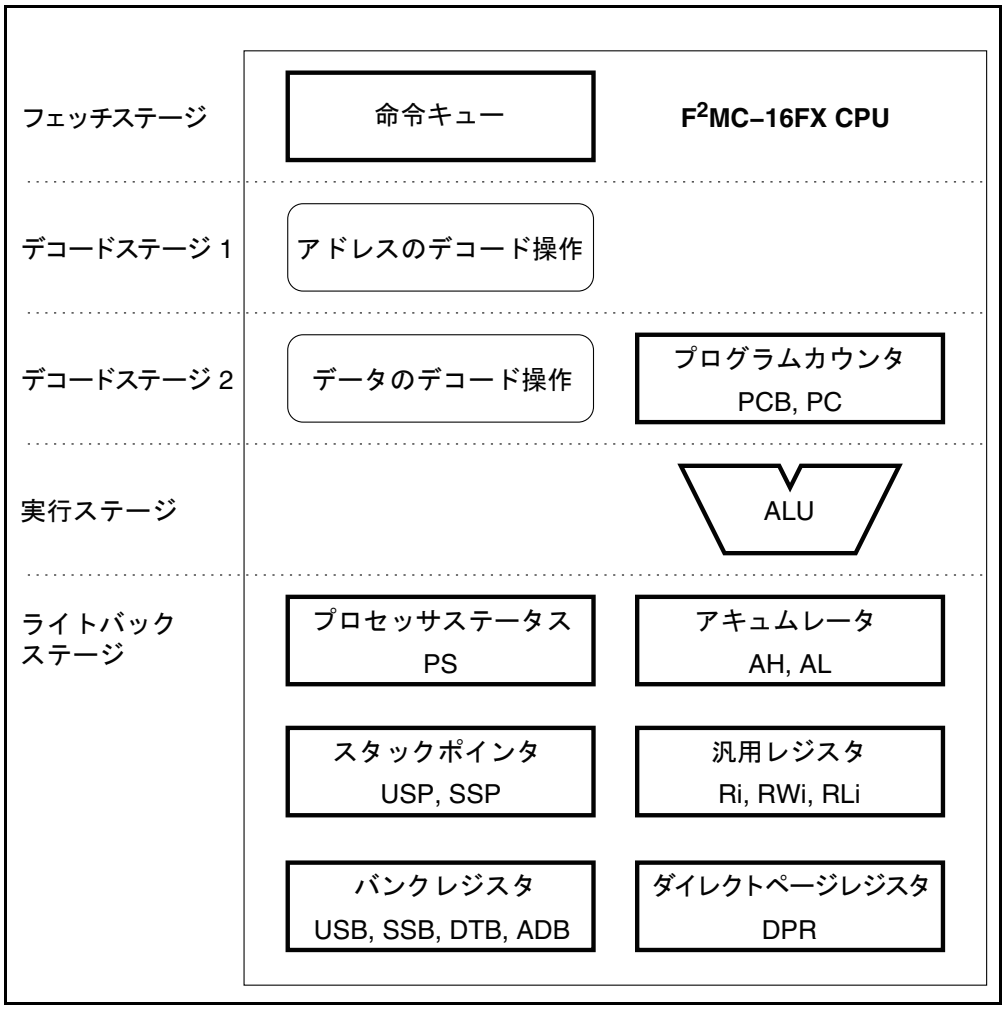
2.2 ハードウェア構造

CPUおよび16FXコアのハードウェア構造を説明します。

2.2.1 CPUのハードウェア構造

CPUのブロックダイヤグラム

Figure 2-1. CPUのブロックダイヤグラム

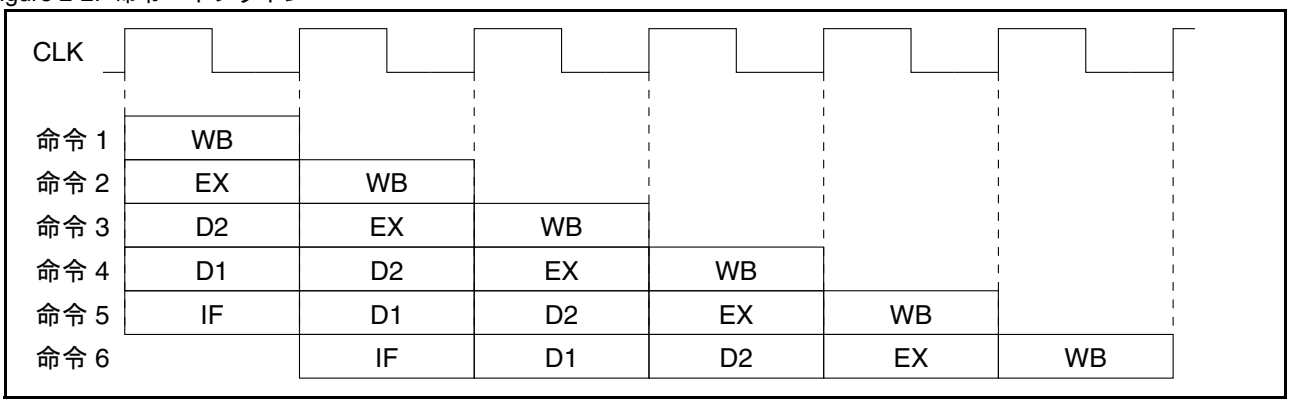


CPUパイプラインの動作

CPU には 5 段の命令パイプラインが採用されており , ほとんどの命令を 1 クロック周期で実行することができます。パイプラインは以下のステージから構成されています。

- 命令フェッチ(IF):命令キューから命令をフェッチします。
- 命令デコード1 (D1):命令をデコードして, アドレス演算についての制御を行います。
- 命令デコード2 (D2):命令をデコードして, オペランドおよびデータ演算の選択を行います。
- 実行(EX): 演算を実行します。
- ライトバック(WB): 演算結果をレジスタまたはメモリ領域に書き込みます。

Figure 2-2. 命令パイプライン



命令は , 順不同で実行されることはありません。つまり , 命令 A が命令 B の前にパイプラインに入ると , 命令 A は必ず命令 B の前にライトバックステージに達するということです。

命令の実行速度は , 原則として 1 サイクルあたり 1 命令です。ただし , メモリ待ちを伴う転送系命令 , 分岐系命令 , および複数サイクル命令の場合 , 命令の実行に複数のサイクルが必要となります。また , コードフェッチ中で命令の転送が遅れる場合にも命令の実行速度が低下します。

命令キュー

CPU には 8 バイトの命令キューがあります。

命令キューには , フェッチユニットによって命令が補充されます。連続したアドレスのコードフェッチにはプリフェッチが使用されます。このプリフェッチの使用により , 16FX コアの CPU とシステムバス間のパイプライン実装で発生する待ち時間の問題が解消されています。

プログラムカウンタ

プログラムカウンタバンク (PCB, プログラムアドレスの上位 8 ビット) およびプログラムカウンタ (PC, プログラムアドレスの下位 16 ビット) は , デコードステージ 1 によって制御されます。

{PCB, PC} の連結 24 ビットアドレスによって次に実行される命令が指定されます。

ALU

ALU は , デコードステージ 2 によって制御されます。このステージで , ALU の演算モードが選択され , オペランドがロードされます。実際の演算は次のサイクルで実行されます。

ALU は , 論理演算および算術演算 (乗除算を含む) に使用します。

CPUレジスタとメモリアクセス

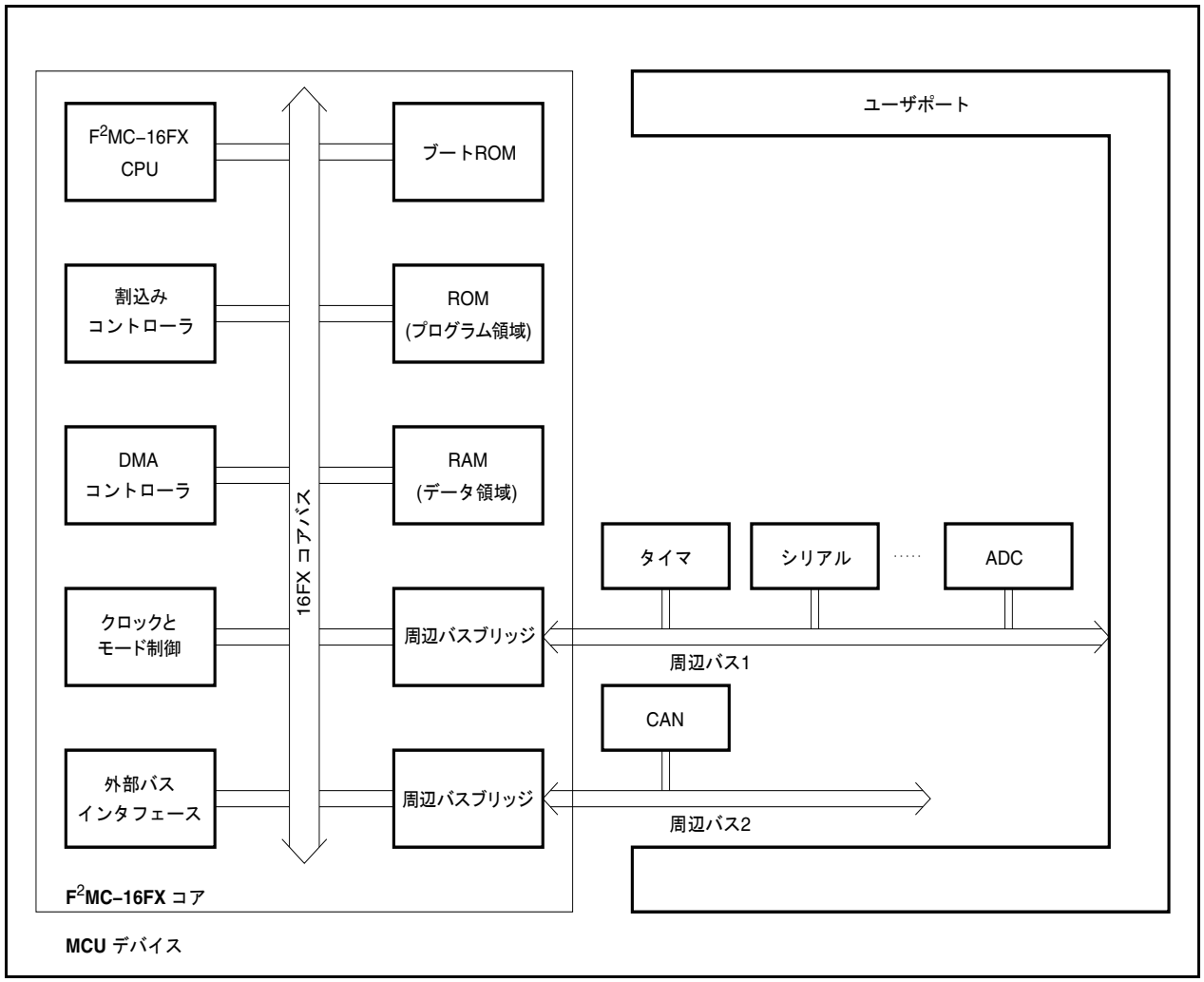
ライトバックステージでは , 演算の結果が CPU レジスタやメモリ領域に書き込まれます。プログラムカウンタ以外のすべての CPU 記録はパイプライン最終ステージに割り当てられます。

2.2.2 16FXコアのハードウェア構造

ブロックダイヤグラム

16FX コアをベースにした MCU デバイスの機能の配置と構成の例を Figure 2-3 に示します。

Figure 2-3. 16FXコアをベースにしたMCUデバイス



割込みコントローラ

割込みコントローラは , 入力される割込み要求 (IRQ) の優先度を判断し , 最も高い優先度の割込み番号を選択します。割込みが認められると , 選択された割込みサービスが CPU によって処理されます。各ハードウェア IRQ には , 優先度を制御するための割込みレベルレジスタがあります。

DMAコントローラ

DMA コントローラは, CPUが現在実行しているプログラムを中断することなくIRQを処理することもできます。これにより , 周辺とメモリ間のデータ転送を自動化することもできます。

デバイスによって , 最大 16 の DMA チャネルを使用することができます。各 DMA チャネルでは実行する IRQ 番号を選択できます。

クロックとモード制御

このユニットは , 動作モードを制御し , デバイスが正確に動作しているかを監視します。また , すべてのユニットに対して , それぞれの動作モードに合わせて適切なクロックを提供します。

外部バスインタフェース

外部バスインタフェースはオプションです。使用されるかどうかはデバイスの構成によります。

ブートROM

リセットによるデバイス初期化の後 , プログラムカウンタはブート ROM を指し示し , CPU によりブート ROM プログラムの実行が開始されます。これ以降 , デバイスが初期化されるとリセットベクタがフェッチされ , ブート ROM のコードはリセットベクタで分岐し , ユーザプログラムを実行するようになります。

周辺バスブリッジ

周辺バスブリッジは , 16FX コアのシステムバスと周辺バス間のインタフェースとなります。この周辺バスは , MCU 内部のすべての周辺リソースを接続しています。

また , 周辺バスブリッジによってコアクロックと周辺クロックドメインが同期します。

2.3 メモリ空間

F²MC-16FX CPUには16MBのメモリ空間があります。

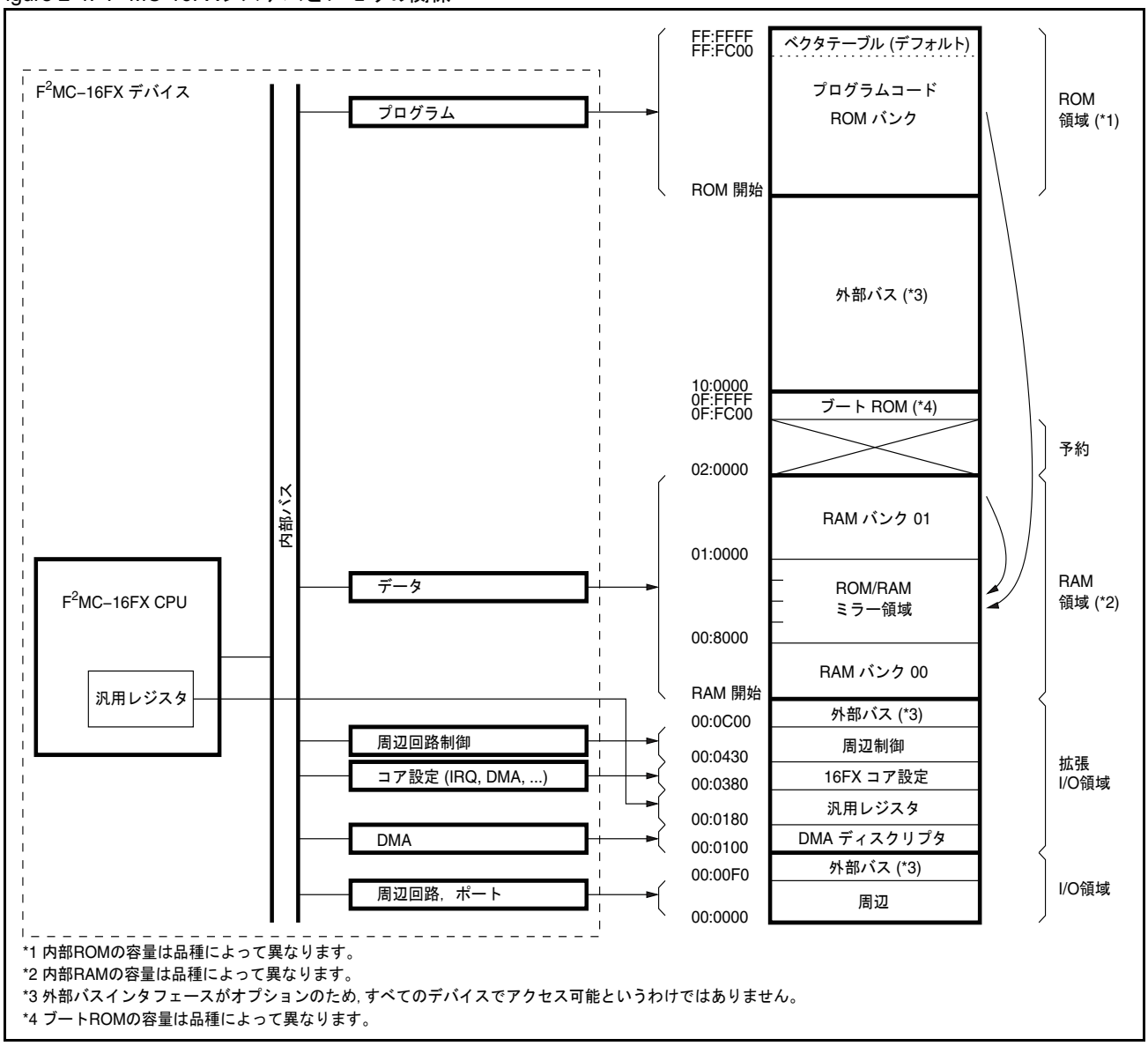
2.3.1 メモリ領域

すべてのI/Oアドレス, プログラムおよびデータは, F²MC-16FX CPUの16Mバイトのメモリ空間に配置されます。CPUは, 24ビットのアドレスバスで示されるアドレスを使って, リソースにアクセスできます。

2.3.1.1 F²MC-16FXシステムのメモリ領域

Figure 2-4 は , F²MC-16FX システムとメモリマップの関係例を示します。

Figure 2-4. F²MC-16FXシステムとメモリの関係



2.3.1.2 I/O領域

I/O 領域のアドレスは , "00:0000$_H$" ～ "00:00FF$_H$" です。このアドレスは , 命令と共にオペランドとして 8 ビットアドレスに指定することで , 直接 I/O アドレス指定に使用することができます。直接 I/O アドレス指定方式を使う命令の場合 , ダイレクトページレジスタ (DPR) またはデータバンクレジスタ (DTB) の指定値に関係なくレジスタにアクセスします。

- 周辺機能制御レジスタのI/O領域(アドレス: "00:0000$_H$"～"00:00EF$_H$")

このレジスタは , 内蔵周辺機能およびポートを制御します。

- 外部I/O領域(アドレス: "00:00F0$_H$"～"00:00FF$_H$")

このアドレスでは , 外部バス領域の I/O アドレスにアクセスします。

外部バスインタフェースはオプション部品です。

2.3.1.3 拡張I/O領域

- DMAディスクリプタ領域(アドレス: "00:0100$_H$"～"00:017F$_H$")

この領域には , 転送モード , I/O アドレス , 転送数およびバッファアドレスを格納します。

この領域は通常の RAM として使用することができます。

- 汎用レジスタ領域(アドレス: "00:0180$_H$"～"00:037F$_H$")

8 ビット , 16 ビット , 32 ビットの算術演算や転送に使用する汎用レジスタ (GPR) はこの領域にマッピングされます。この領域は通常の RAM として使用することができます。

GPR はメモリにマッピングされますが , GPR を使用した命令によるダイレクトアクセスの速度は 16LX と比較して大幅に向上しています。

- 16FXコア制御レジスタ(アドレス: "00:0380$_H$"～" 00:042F$_H$")

この領域は , CPU コアの内部構成レジスタが使用します。

- 周辺機能制御レジスタの拡張I/O領域(アドレス: "00:0430$_H$"～"00:0BFF$_H$")

この領域のレジスタで内蔵の周辺機能を制御します。

- 外部バス領域(アドレス: "00:0C00$_H$"～"00:0FFF$_H$")

この領域は , バンク 0 にある外部バス領域の拡張 I/O 範囲にアクセスします。

28K バイト未満の RAM が実装された場合 (これは MCU デバイスによって異なります), 外部バス領域は RAM 先頭アドレスから 1 を減算したアドレスまで拡張可能です。

外部バスインタフェースはオプションです。

2.3.1.4 RAM領域

■ バンク00のデータ領域(アドレス: "$00:1000_H$"～"$00:7FFF_H$")

この領域には , 28K バイトまでのスタティック RAM を実装します。
内部 RAM の容量はデバイスによって異なります。そのため , RAM の先頭アドレスは実装メモリの容量で決まります。

■ バンク00のミラー領域(アドレス: "$00:8000_H$"～"$00:FFFF_H$")

この領域は, デフォルトではROMの先頭から32Kバイトの先頭セクションにアクセスする場合に使用されます。
ROM ミラー機能は容量によって変化します。ROM ミラーには , 8K バイトのセグメントを 4 つまで選択することができます。バンク 00 でアクセスする ROM バンクは , ROMM レジスタで選択することができます。
ROM ミラー機能がオフになっている場合は , バンク 00 を介してバンク 01 の RAM にアクセスできます。
バンク 00 のミラー領域を使えば , C コンパイラのスモールモデルをサポートできます。最下位の 16 ビットが同一であるため , 選択された ROM または RAM バンクへのデータアクセスに対応する部分はポインタ宣言で far 指定をせずに参照することができます。

■ バンク1のデータ領域(アドレス: "$01:0000_H$"～"$01:FFFF_H$")

この領域には , 64K バイトまでのスタティック RAM を実装します。
内部 RAM の容量はデバイスによって異なります。

2.3.1.5 ROM領域

■ プログラム領域(アドレス: "～$FF:FFFF_H$")

内部プログラム領域として ROM が内蔵されています。
内部 ROM の容量と先頭アドレスはデバイスによって異なります。
デフォルトでは , "$FF:FC00_H$" ～ "$FF:FFFF_H$" はベクタテーブルによって割り当てられます。

2.3.1.6 ベクタテーブル領域

■ ベクタテーブル領域

この領域は , 割込みベクタおよびリセットベクタのベクタテーブルとして使用します。対応する処理ルーチンの先頭アドレスをそれぞれのベクタテーブルのエントリにデータとして設定します。
ベクタテーブル領域は , テーブルベースレジスタ (TBR) のユーザプログラムによって変更することができます。この領域は , RAM, ROM, または外部領域に配置することができます。この領域は , TBR によって定義されたアドレスで始まる 1K バイト領域になります。ベクタテーブル領域のアドレスは以下のとおりです 。

$VEC_{START} = TBR \times 100_H$ ～
$VEC_{END} = TBR \times 100_H + 3FF_H$

デフォルトでは , ベクタテーブル領域は "$FF:FC00_H$" ～ "$FF:FFFF_H$" になります。
ベクタコール命令のベクタテーブルは , PCB レジスタによって指定される現在のプログラムバンクに存在します。CALLV 命令は , "$PCB:FFE0_H$" ～ "$PCB:FFFF_H$" の範囲のプログラムバンクの先頭に位置するこの 32 バイト領域を使用します。TBR は CALLV 命令が使用するベクタの位置には影響しません。
割込みベクタ領域(割込みベクタテーブルの上位8つのエントリ)がプログラムバンクの先頭にある場合, CALLV 命令を使用する際には注意が必要です。両方のベクタテーブルが同じ場所を共有するためです。

2.3.2 リニア方式によるアドレス指定

リニア方式によるアドレス指定では , 24 ビットアドレス全体を命令で指定します。リニア方式には以下に示す 2 つの種類があります。

- 24ビットオペランド指定: オペランドで直接24ビットのアドレスを指定します。
- 32ビットレジスタ間接指定: 32ビットの汎用レジスタの下位24ビットをアドレスとして間接指定します。

2.3.2.1 24ビットオペランド指定

Figure 2-5 に, 24ビットオペランド指定の例を示します。Figure 2-6 では, 32ビットレジスタ間接指定の例を示します。

Figure 2-5. リニア方式の例 (24ビットオペランド指定)

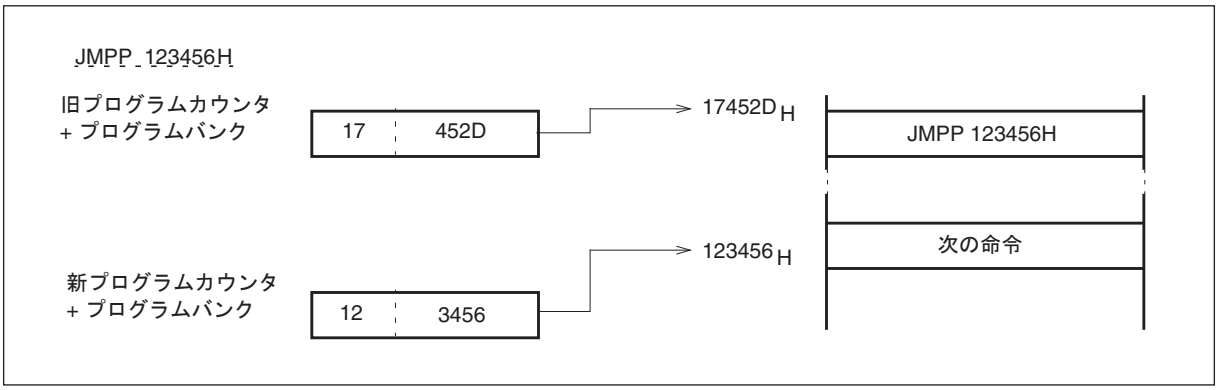
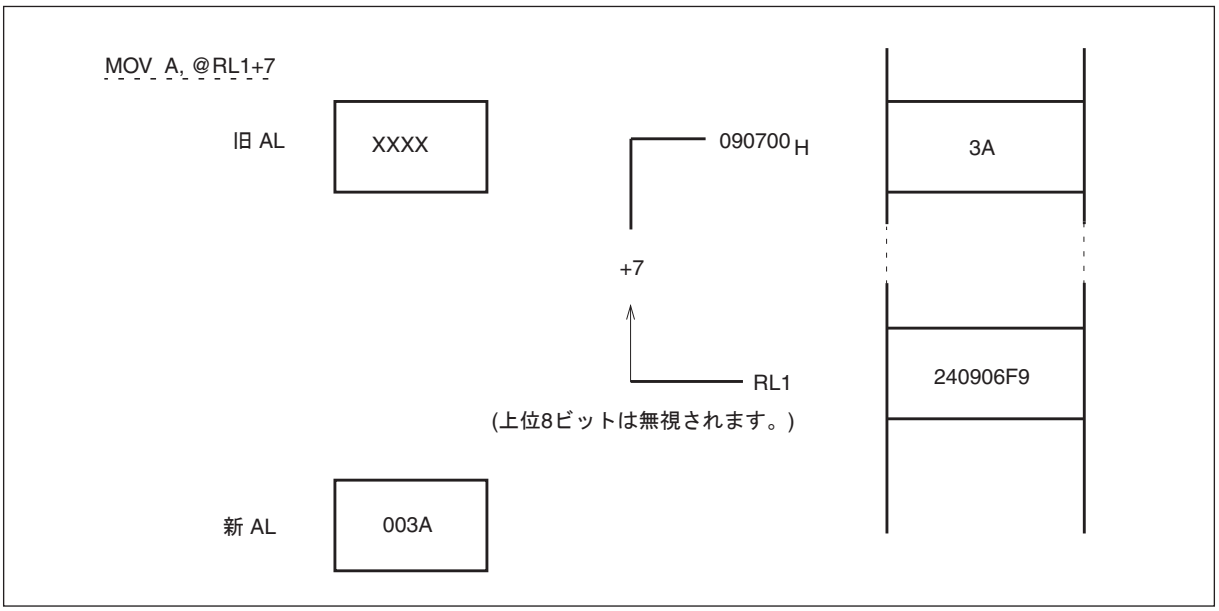


Figure 2-6. リニア方式の例 (32ビットレジスタ間接指定)



2.3.3 バンク方式によるアドレス指定

バンク方式によるアドレス指定では, アドレスの上位8ビットはバンクレジスタで指定し, 残りの下位16ビットは命令で指定します。

2.3.3.1 バンク方式によるアドレス指定

バンク方式によるアドレス指定では , 16M バイトの空間を 64K バイトごとの 256 バンクに分割します。以下に示す 5 つのバンクレジスタで , 各空間に対応するバンク (アドレスの上位 8 ビット) を指定します。

- プログラムカウンタバンクレジスタ(PCB)

PCB によって指定される 64K バイトのバンクはプログラム (PC) 空間とよばれます。PC 空間には , 命令コード , CALLV 命令のベクタテーブル , 即値データなどが入ります。

- データバンクレジスタ(DTB)

DTB によって指定される 64K バイトのバンクはデータ (DT) 空間とよばれます。DT 空間には , リード / ライト可能データおよび内外リソースの制御 / データレジスタが入ります。

- ユーザスタックバンクレジスタ(USB)/システムスタックバンクレジスタ(SSB)

USB または SSB によって指定される 64K バイトのバンクはスタック (SP) 空間とよばれます。SP 空間は , プッシュ / ポップ命令や割込みのレジスタ退避などの際にスタックアクセスが生じたときにアクセスされる領域です。USB と SSB のどちらの空間が使用されるかは , コンディションコードレジスタ中の S フラグの値で決まります。

- アディショナルデータバンクレジスタ(ADB)

ADB によって指定される 64K バイトのバンクはアディショナルデータ (AD) 空間とよばれます。AD 空間には , DT 空間に入りきらなかったデータなどが入ります。

Table 2-1は，各アドレッシングモードで使用されるデフォルト空間を示しています。これは，命令のコード効率を高めるために事前に設定されています。

Table 2-1. デフォルト空間

デフォルト空間	アドレッシングモード
プログラム空間	PC 間接 , プログラムアクセス , 分岐系
データ空間	@RW0, @RW1, @RW4, @RW5, @A, addr16, または dir を用いたアドレッシングモード
スタック空間	PUSHW, POPW, @RW3, または @RW7 を用いたアドレッシングモード
アディショナルデータ空間	@RW2 または @RW6 を用いたアドレッシングモード

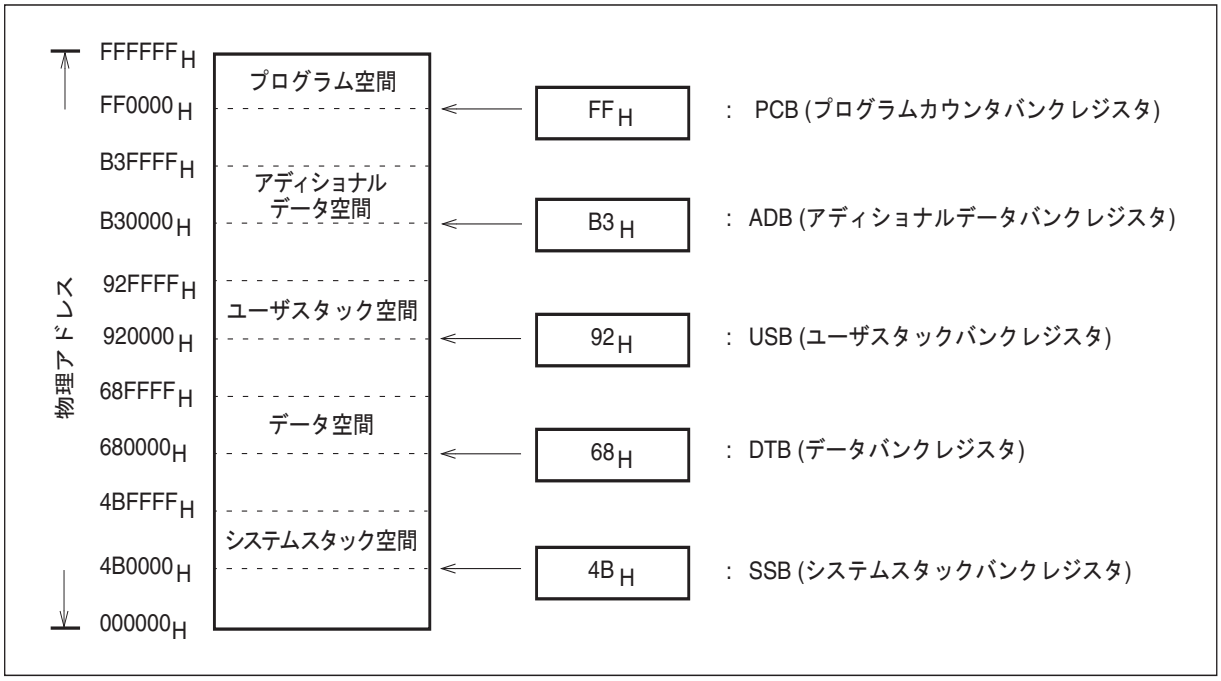
あるアドレッシングモードにデフォルト以外の空間を使用したいときは, 各バンクに対応しているプリフィックスコードを命令に先行して指定することにより, そのプリフィックスコードに対応した任意のバンク空間をアクセスすることができるようになります。

Table 2-2. バンクセレクトプリフィックス

バンクセレクトプリフィックス	選択される空間
PCB	プログラム空間
DTB	データ空間
ADB	アディショナルデータ空間
SPB	選択時のスタックフラグの状態によってシステムスタック空間またはユーザスタック空間が使用されます。

Table 2-7は, レジスタバンクに分割されたメモリ空間の例で, 各空間の物理アドレスが示されています。

Figure 2-7. バンク方式でアドレス指定した場合のメモリ空間およびバンクレジスタの値



2.3.3.2 バンクレジスタの初期化

リセットを行うと , DTB, USB, SSB, および ADB は "00_H" に初期化されます。リセットアドレスがブート ROM プログラムの先頭アドレス "$0F:FC00_H$" に固定されているため ,PCB は "$0F_H$" に初期化されます。

外部ベクタモードでは , ブート ROM コードの実行後 , "$FF:FFDC_H$" 番地のリセットベクタで指定された値を読み出して設定します。内部ベクタモードでは , 製品に設定されている固定値と共に PCB を読み出して設定します。

指定されたリセットベクタアドレスからユーザプログラムを開始する時点で, DT, SPおよびAD空間はバンク00_H("000000_H"～"$00FFFF_H$")に割り当てられ, PC空間はリセットベクタによって指定されたバンクに割り当てられます。

Table 2-3. バンクレジスタの初期化

バンクレジスタ	リセットによる初期化	ブートROMによる初期化	
		外部ベクタモード	内部ベクタモード
DTB	00_H	00_H	00_H
USB	00_H	00_H	00_H
SSB	00_H	00_H	00_H
ADB	00_H	00_H	00_H
PCB	$0F_H$	$FF:FFDE_H$から読み出された値.	製品に設定された値

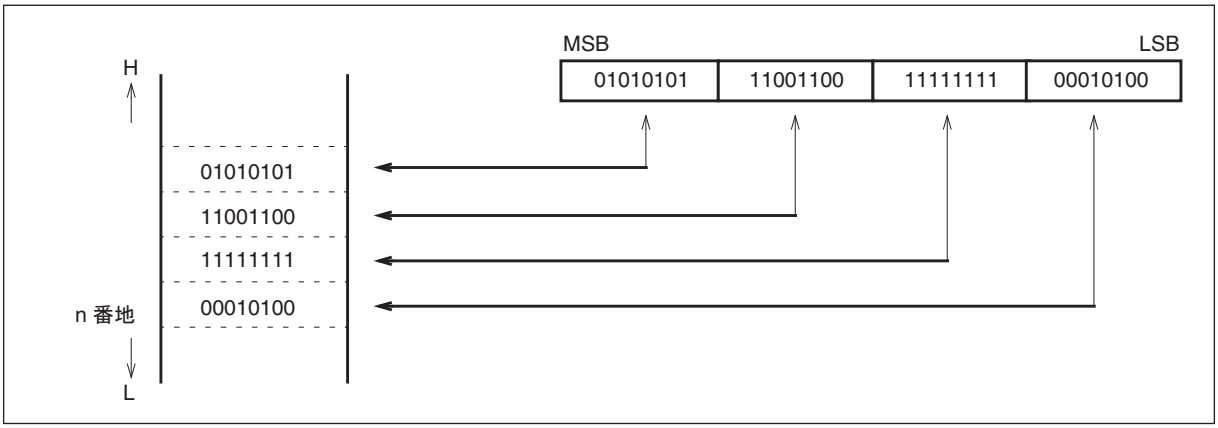
2.3.4 メモリ空間上の多バイトデータ

データは下位アドレスからメモリに書き込まれます。そのため，32ビットデータの場合，上位16ビットの前に下位16ビットが転送されます。
このとき，下位ビットを書き込んだ直後にリセット信号が入力された場合，上位ビットが書き込まれないことがあります。

2.3.4.1 メモリ空間での多バイトデータの配置

Figure 2-8は，メモリ上の多バイトデータの構成を示した図です。データの下位8ビットがn番地に，次の8ビットがn+1番地の順に配置されます。

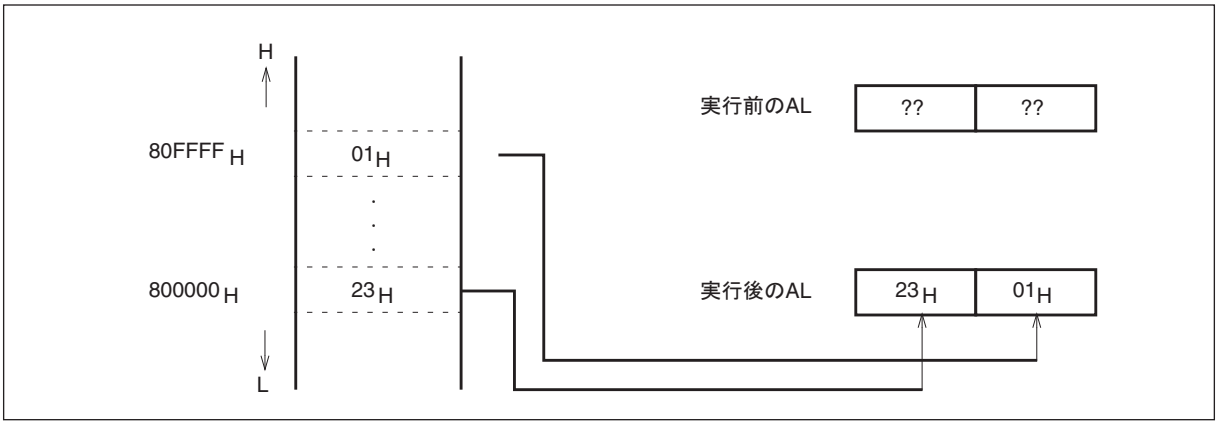
Figure 2-8. メモリ上の多バイトデータの配置サンプル



2.3.4.2 多バイトデータのアクセス

基本的に , アクセスは 1 つのバンク内で行われます。多バイトデータにアクセスする命令の場合 , "$FFFF_H$" 番地の後に , 同じバンクの "0000_H" 番地が続きます。Figure 2-9 は , 多バイトデータにアクセスする命令の例です。

Figure 2-9. MOVW A, 080FFFFHの実行



2.4 専用レジスタ

F^{2}MC-16FX CPUのレジスタには, 専用レジスタと汎用レジスタの2種類があります。
F^{2}MC-16FX CPUの専用レジスタについて説明します。専用レジスタは, CPU内に内蔵されている専用ハードウェアで, 用途がCPUのアーキテクチャ上で限定されているものです。このタイプのレジスタは, アドレスを使用せずにアクセスすることができます。レジスタの動作は, 専用の命令によって規定します。

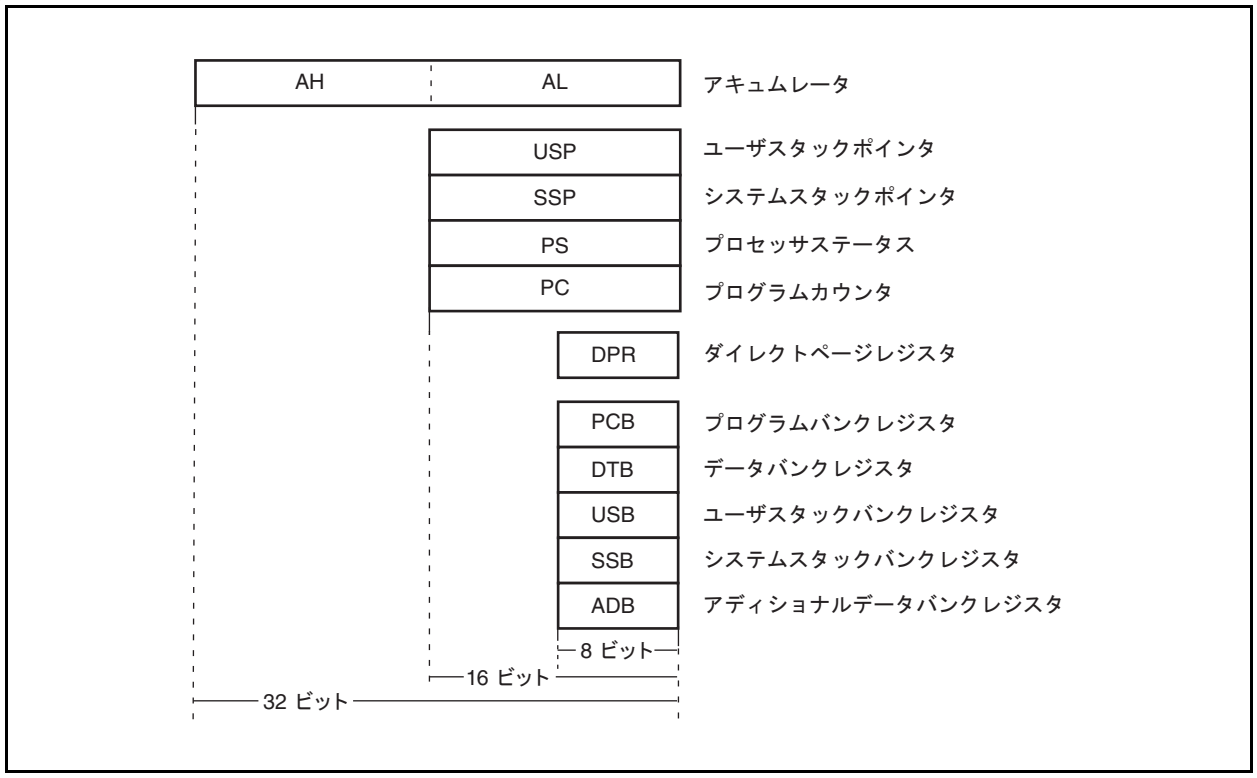
専用レジスタ

F^{2}MC-16FX CPU には以下の専用レジスタがあります。

- アキュムレータ(A=AH:AL): 16ビット×2本のアキュムレータ(合計32ビットのアキュムレータとしても使用可能)
- ユーザスタックポインタ(USP): 16ビットのユーザスタックポインタ
- システムスタックポインタ(SSP): 16ビットのシステムスタックポインタ
- プロセッサステータス(PS):システムの状態を示す16ビットのレジスタ
- プログラムカウンタ(PC):次に実行される命令のアドレスを格納する16ビットのレジスタ
- プログラムバンクレジスタ(PCB):プログラムバンクを示す8ビットのレジスタ
- データバンクレジスタ(DTB):データバンクを示す8ビットのレジスタ
- ユーザスタックバンクレジスタ(USB):ユーザスタックバンクを示す8ビットのレジスタ
- システムスタックバンクレジスタ(SSB):システムスタックバンクを示す8ビットのレジスタ
- アディショナルデータバンクレジスタ(ADB):アディショナルデータバンクを示す8ビットのレジスタ
- ダイレクトページレジスタ(DPR):ダイレクトアクセスするページを示す8ビットのレジスタ

Figure 2-10に, 専用レジスタを示します。

Figure 2-10. 専用レジスタ



2.4.1 アキュムレータ (A)

アキュムレータ(A)レジスタは, 2つの16ビット算術演算レジスタ(AHおよびAL)から構成されています。演算結果やデータ転送の一時記憶に使用されるレジスタです。

2.4.1.1 アキュムレータ (A)

アキュムレータ (A) は , 2 つの 16 ビット算術演算レジスタ (AH および AL) から構成されています。アキュムレータ (A) は , 演算結果やデータ転送の一時記憶に使用されます。32 ビットデータ処理時は , AH と AL を連結して使用します。16 ビットデータ処理モードのワード処理や 8 ビットデータ処理モードのバイト処理のときは AL のみが使用されます (Figure 2-11 および Figure 2-12 を参照)。アキュムレータ (A) 中のデータは , メモリまたはレジスタ (Ri, RWi, RLi) 中のデータと各種演算ができます。また , F^2MC-8L のときと同様 , ワード長以下のデータを AL へ転送すると , 転送前の AL 中のデータが自動的に AH に転送されます (データ保持機能)。このデータ保持機能と AL-AH 間演算により , 処理効率の向上が可能になっています。

バイト長以下のデータを AL に転送する際は , データは符号拡張またはゼロ拡張されて 16 ビット長となって AL に格納されます。AL 中のデータは , ワード長としてもバイト長としても扱えます。

AL にバイト処理の算術演算命令を実行すると , 演算前の AL の上位 8 ビットは無視されます。演算後の上位 8 ビットはすべて "0" になります。

アキュムレータ (A) はリセットでは初期化されず , リセット直後は不定値になります。

Figure 2-11. 32ビットデータ転送の例

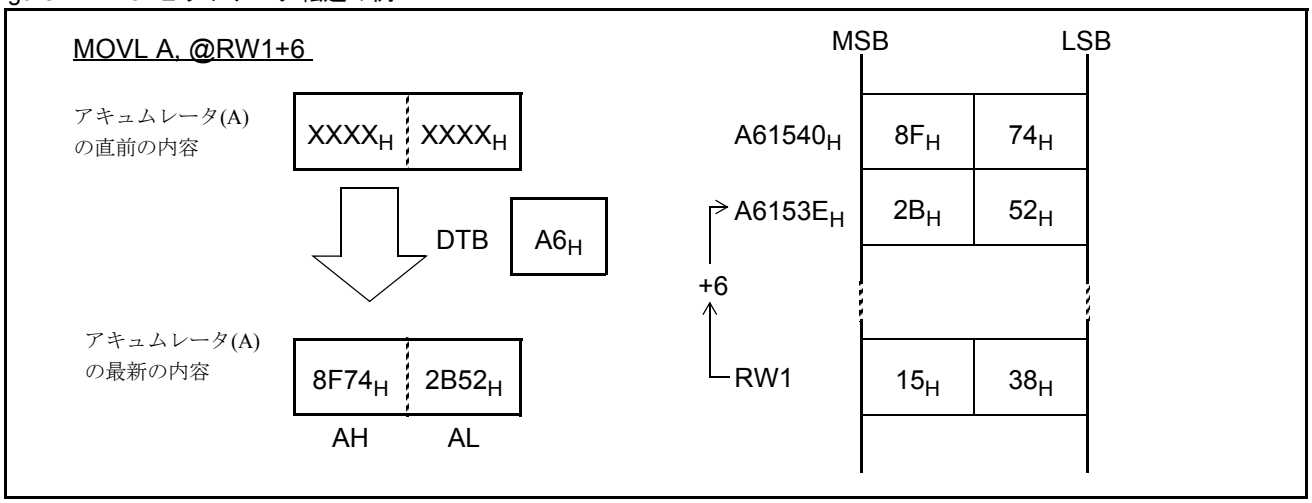
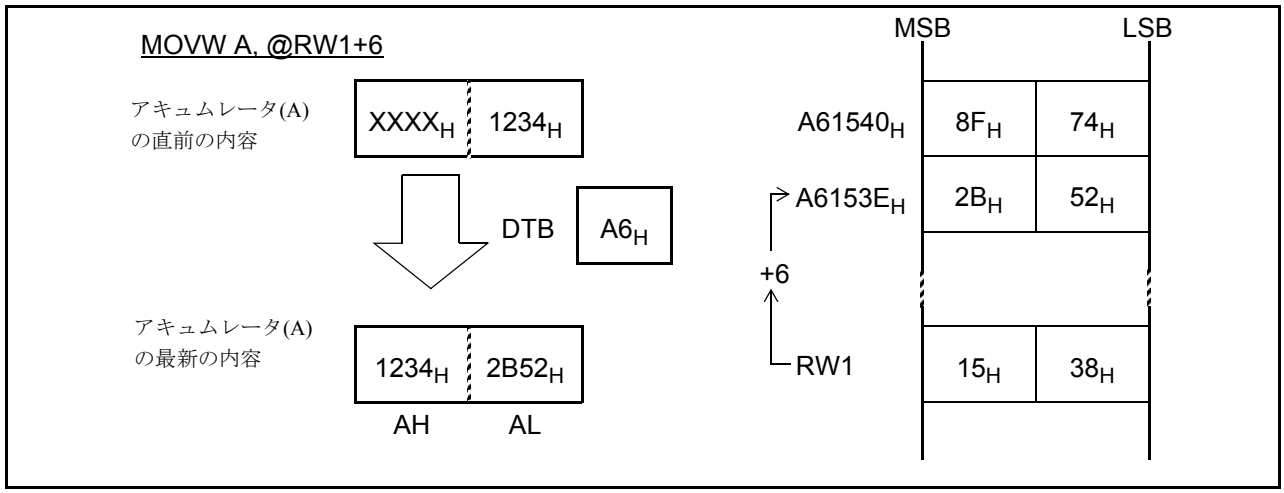


Figure 2-12. データ保持機能を使ったAL-AH転送の例



2.4.2 ユーザチェックポインタ(USP)とシステムスタックポインタ(SSP)

USPおよびSSPは, プッシュ/ポップ命令およびサブルーチン実行時のデータ退避/復帰のメモリアドレスを示す16ビットレジスタです。

2.4.2.1 ユーザスタックポインタ(USP)とシステムスタックポインタ(SSP)

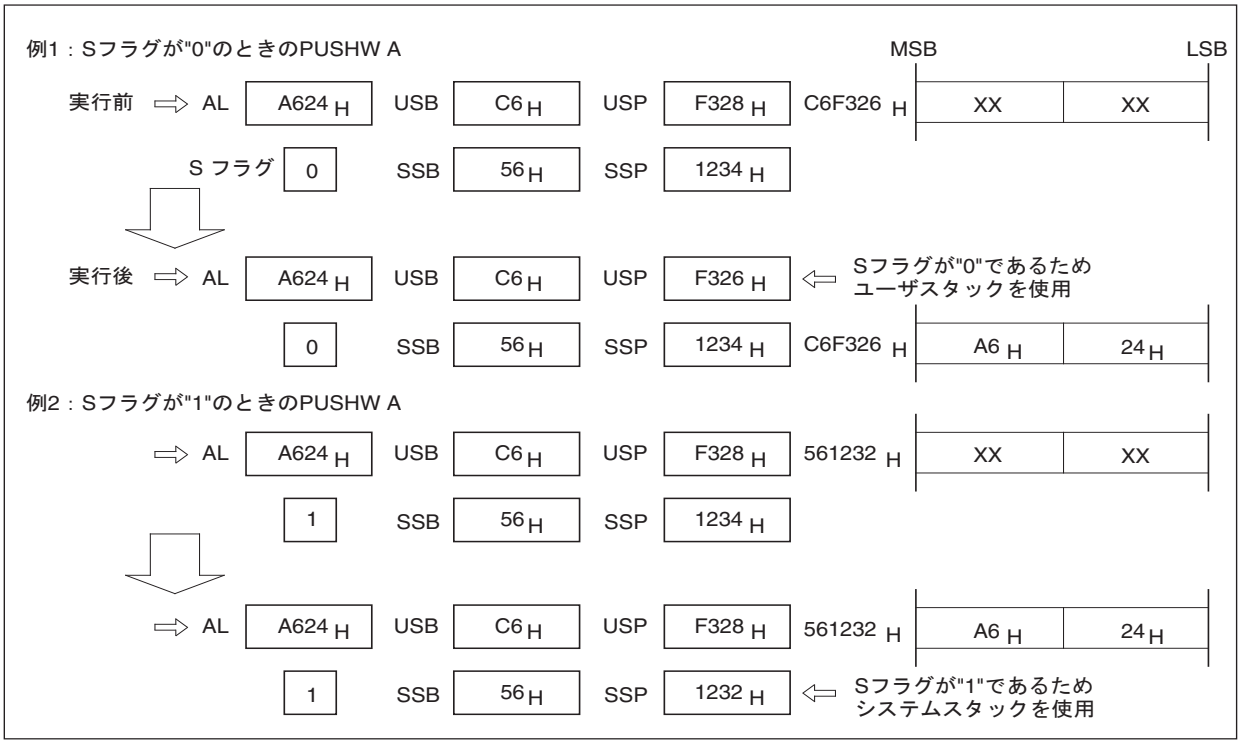
USP および SSP は , プッシュ / ポップ命令およびサブルーチン実行時のデータ退避 / 復帰のメモリアドレスを示す 16 ビットのレジスタです。USP および SSP レジスタはスタック命令で使用します。

プロセッサステータスレジスタ中の S フラグが "0" のときは USP レジスタが有効になり , S フラグが "1" のときは SSP レジスタが有効になります (Figure 2-13 を参照)。また , 割込みが受け付けられると S フラグがセットされるため , 割込み時のレジスタ退避は必ず SSP の示すメモリ領域で行われます。割込みルーチンでのスタック処理は SSP で , 割込みルーチン以外のスタック処理は USP が用いられます。スタック空間を分けない場合は SSP だけをお使いください。

スタック時のアドレスの上位 8 ビットは , SSP の場合は SSB, USP の場合は USB により示されます。

また , USP および SSP はリセットでは初期化されず , 不定値になります。

Figure 2-13. スタック操作命令とスタックポインタ



＜注意事項＞

スタックポインタに設定する値は , 原則として偶数アドレスを使用してください。奇数値を使うとスタック動作のパフォーマンス低下の原因となります。

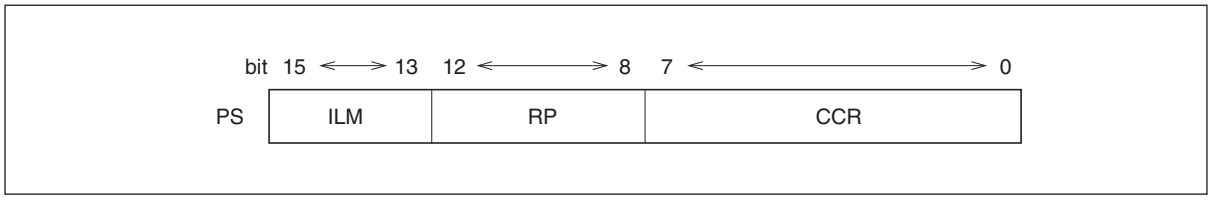
2.4.3 プロセッサステータス(PS)

プロセッサステータス(PS)は, CPUの動作制御を行うビットとCPUの状態を示すビットより構成されています。

2.4.3.1 プロセッサステータス(PS)

Figure 2-14 に示すように, プロセッサステータス (PS) の上位バイトはレジスタバンクポインタ (RP) とインタラプトレベルマスクレジスタ (ILM) より構成されています。レジスタバンクポインタ (RP) は , レジスタバンクの先頭アドレスを示します。PS レジスタの下位バイトは命令実行結果および割込み発生などによりセット / リセットされる各種フラグを持つコンディションコードレジスタ (CCR) より構成されています。

Figure 2-14. プロセッサステータス(PS)の構造



2.4.3.2 コンディションコードレジスタ(CCR)

Figure 2-15 に, コンディションコードレジスタ(CCR)の構成図を示します。

Figure 2-15. コンディションコードレジスタ(CCR)の構成

bit	7	6	5	4	3	2	1	0	
	P	I	S	T	N	Z	V	C	PS: CCR
	0	0	1	0	0	0	0	0	リセット後の初期値
	1	0	1	X	X	X	X	X	ブートROM実行後の値

■ P: 特権モードフラグ

P = 1 はユーザモード , P = 0 は特権モードを示します。

P フラグはリセットによってクリアされますが , ブート ROM コードの実行中にセットされます。

P フラグをクリアして , ほかのすべてのハードウェア割込みを無効にすることができるのは , NMI, HW-INT9 (EDSU) および DSU 割込みだけです。P フラグがクリアされた場合 , ILM が特権モード (P0 ～ P7) のシステムインタラプトレベルを決定します。これらの割込みレベルは , ユーザモード (U0 ～ U7) のどの ILM 設定よりも優先度が高くなります。

P フラグは , 専用の命令 (OR CCR #imm, POPW PS) またはプロセッサステータスの復帰 (RETI, JCTX @A) によってセットすることができます。P が "1" になっている場合は , P=0 への復帰は認められません。

■ I: 割込み許可フラグ

ソフトウェア割込み以外の割込みに対し , I フラグに "1" が設定されている場合は割込みが許可され , I フラグが "0" の場合は割込みがマスクされます。I フラグはリセットによって "0" にクリアされます。

■ S: スタックフラグ

S フラグが "0" の場合は , スタックポインタとして USP が有効になります。

S フラグが "1" の場合は , スタックポインタとして SSP が有効になります。

S フラグは , 割込みが受け付けられた場合 , またはリセットされた場合に "1" にセットされます。

■ T: スティッキィビットフラグ

論理右シフト命令または算術右シフト命令を実行した場合に , キャリによってシフトアウトされたデータに 1 つ以上 "1" があれば T フラグに "1" が設定され , "1" がなければ "0" が設定されます。

また , T フラグはシフト量がゼロの場合でも "0" に設定されます。

■ N: ネガティブフラグ

N フラグは , 演算結果の MSB が "1" の場合に "1" にセットされ , "0" の場合は "0" にクリアされます。

■ Z: ゼロフラグ

Z フラグは , 演算結果がすべて "0" の場合に "1" にセットされ , それ以外の場合は "0" にクリアされます。

■ V: オーバフローフラグ

V フラグは , 演算の実行で符号付き数値としてオーバフローが発生した場合に "1" にセットされ , オーバフローが発生しなかった場合は "0" にクリアされます。

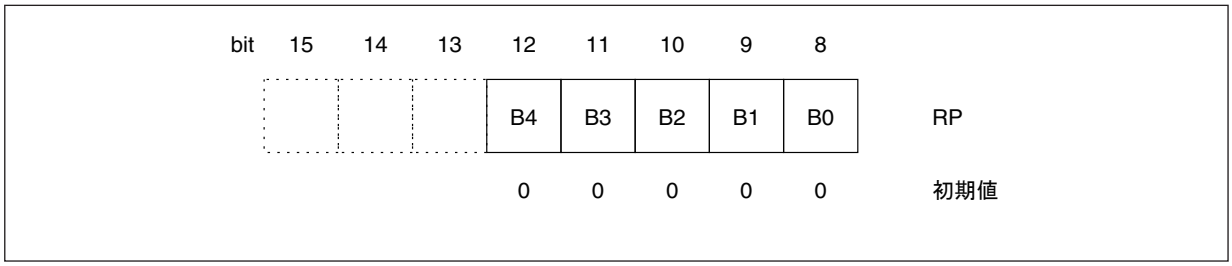
■ C: キャリフラグ

C フラグは , 演算の実行で MSB からの桁上りまたは LSB からの桁下りが発生した場合に "1" にセットされ , 発生しなかった場合は "0" にクリアされます。

2.4.3.3 レジスタバンクポインタ(RP)

RP レジスタは , F^{2}MC-16FX の汎用レジスタと内部 RAM アドレスとの関係を示します。具体的には , RP レジスタは現在使用しているレジスタバンクの先頭メモリアドレスを次の変換式 [00180_H + (RP)×10_H] を使って示します。RP レジスタは 5 ビットで構成され ,
"00_H" ～ "$1F_H$" の値をとります。レジスタバンクは , メモリ中の "000180_H" ～ "$00037F_H$" 番地に割り当てることができます。

Figure 2-16. レジスタバンクポインタ(RP)

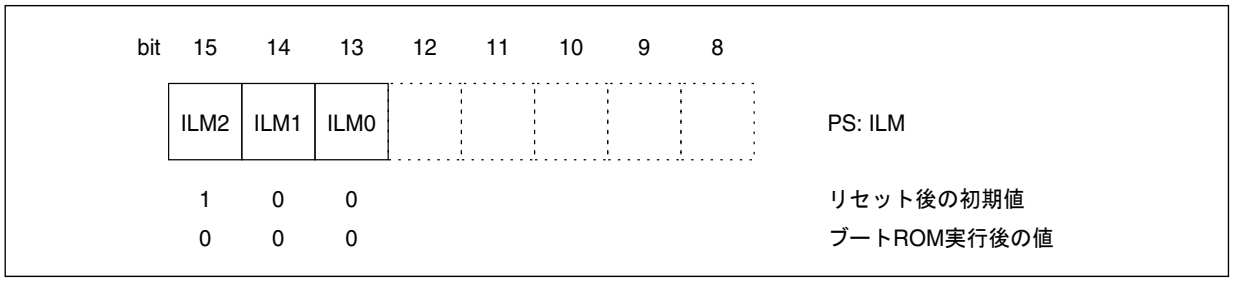


RP レジスタはリセットによりすべて "0" に初期化されます。命令上では RP レジスタに 8 ビットの即値が転送できますが , 実際に使用されるのはそのデータの下位 5 ビットのみです。

2.4.3.4 インタラプトレベルマスクレジスタ(ILM)

ILM レジスタは 3 ビットで構成されており , CPU の割込みマスクのレベルを示します。割込み要求は , ILM レジスタおよび P フラグよりも高い割込みの場合のみ受け付けられます。割込みの優先度が最も高いのが "P0" で , 最も低いものが "U7" になります。したがって , 割込みが受け付けられるには , 現状の ILM の保持値より小さい値の要求でなければなりません (Figure 2-17 を参照)。また , P フラグについても考慮する必要があります。割込みが受け付けられると , その割込みのレベル値が P フラグと ILM レジスタにセットされ , これ以降の同じレベルおよびそれより優先順位の低い割込みは受け付けられなくなります。

Figure 2-17. インタラプトレベルマスクレジスタ(ILM)



ILM はリセットにより "100_B" に初期化されますが , ブート ROM プログラムの実行中に ILM は "000_B" にセットされます。

命令上では ILM レジスタに 8 ビットの即値が転送できますが , 実際に使用されるのはそのデータの下位 3 ビットのみ (MOV ILM #imm, POPW PS, RETI, JCTX @A) です。P=1 (ユーザレベル) の場合 , ILM にはどのような変更でも加えることができます。P=0 (特権レベル) の場合 , (P=1 で) 新しい値をユーザレベル U0 ～ U7 に設定する場合 , または特権レベル (P0 ～ P7) を上げる場合にのみ ILM への変更が受け付けられます。特権モード P0 ～ P7 では , 上位レベルから実行する命令で下位レベルに行くことはできません。P フラグに 0 を書き込んで P=0 の状態でレベルを下げることができるのは NMI, HW-INT9 および DSU 割込みだけです。

＜注意事項＞

P フラグは ILM の拡張ビットと考えることができます。次に , 割込みレベルマスク {P, ILM} の最上位ビットを設定します。

リセットによる初期化の後 , CPU はレベル P4 になります。これにより , DSU を除くすべての割込み (NMI を含む) が無効になります。ブート ROM プログラムの実行後 , CPU はレベル U0 になります。周辺割込みは無効になります。

P0 ～ P7 のすべての特権モードレベルはロックされ , 命令によってレベルを入力したり下げたりできなくなります。つまり , P0 ～ P7 のレベルは上げることしかできません。これにより , HW-INT9, NMI および DSU 操作が保護されることになります。この場合 , NMI を中断することができ , デバッグセッション中に NMI の受付けをマスクできるのは DSU だけになります。

ユーザレベル U0 ～ U7 は , F^2MC-16LX の割込みレベル 0 ～ 7 と下位互換性があります。P フラグは , ユーザレベルでは書き込めません。

Table 2-4. Pフラグおよびインラプトレベルマスク(ILM)レジスタにより示されるレベル

レベル	Pフラグ	ILMの値	受付可能な割込みレベル		
P0	0	0	なし	割込み禁止	
P1	0	1	レベル < P1	I割込み禁止	
P2	0	2	レベル < P2	I割込み禁止	
P3	0	3	レベル < P3	DSU	
P4	0	4	レベル < P4	DSU	
P5	0	5	レベル < P5	NMI, DSU	
P6	0	6	レベル < P6	NMI, DSU	
P7	0	7	レベル < P7	HW-INT9, NMI, DSU	
U0	1	0	レベル < U0	ユーザ割込み禁止	HW-INT9, NMI, DSU
U1	1	1	レベル < U1	ユーザレベル 0	
U2	1	2	レベル < U2	ユーザレベル 0-1	
U3	1	3	レベル < U3	ユーザレベル 0-2	
U4	1	4	レベル < U4	ユーザレベル 0-3	
U5	1	5	レベル < U5	ユーザレベル 0-4	
U6	1	6	レベル < U6	ユーザレベル 0-5	
U7	1	7	レベル < U7	ユーザレベル 0-6	

2.4.4 プログラムカウンタ(PC)

プログラムカウンタ(PC)は, CPUが実行する命令コードのメモリアドレスの下位16ビットを示す16ビットカウンタです。

2.4.4.1 プログラムカウンタ(PC)

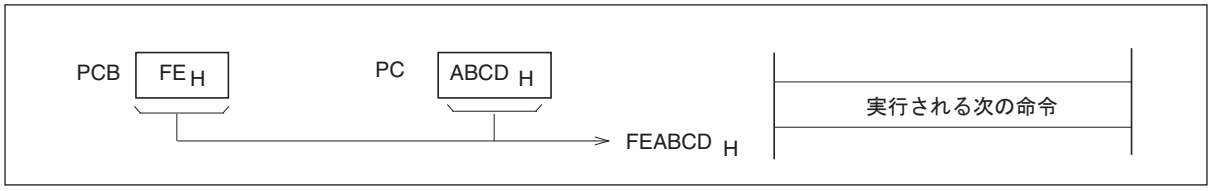
プログラムカウンタ (PC) は , CPU が実行する命令コードのメモリアドレスの下位 16 ビットを示す 16 ビットカウンタです。アドレスの上位 8 ビットはプログラムバンクレジスタ (PCB) で示します。

PC レジスタは , 分岐命令 , サブルーチンコール命令 , 割込み , リセットで内容が更新されます。リニアプログラムセグメントの中では , PC は最後の命令のバイト数分だけ増加します。

PC レジスタは , オペランドにアクセスする際のベースポインタとしても使用できます。

Figure 2-18にプログラムカウンタを示します。

Figure 2-18. プログラムカウンタ



リセットアドレスは , ブート ROM プログラムの先頭アドレス "0F:FC00$_H$" に固定されています。リセット時 , PC は "FC00$_H$" に初期化され , PCB は "0F$_H$" に初期化されます。

外部ベクタモードでは , ブート ROM コードの実行後 , "FF:FFDC$_H$" 番地のリセットベクタで指定された値を読み出し設定されます。内部ベクタモードでは , 製品に設定されている固定値を読み出して , PCB と PC 共に設定されます。

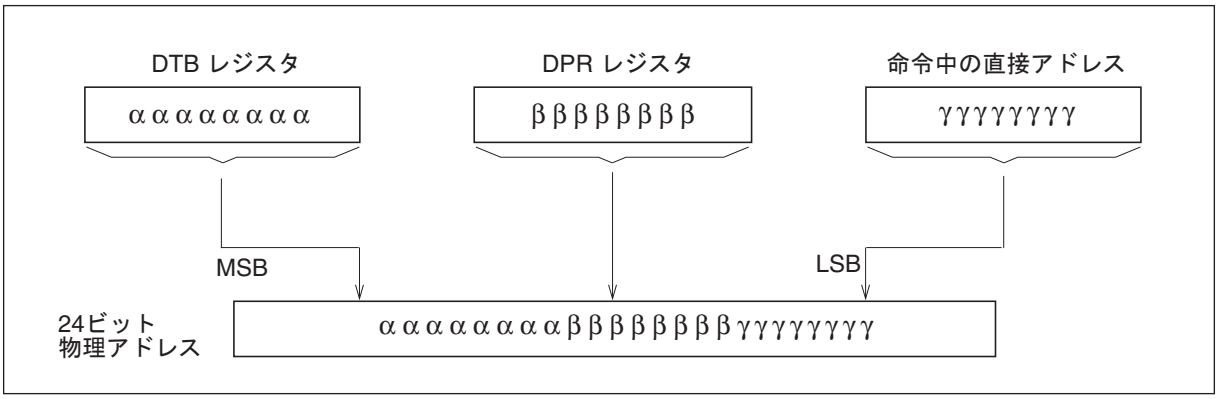
2.4.5 ダイレクトページレジスタ(DPR)

ダイレクトページレジスタ(DPR)は, 直接アドレス指定方式の命令に使用されるオペランドアドレスのbit8～bit15 (addr8～addr15)を指定します。

2.4.5.1 ダイレクトページレジスタ(DPR)

Figure 2-19に示すようにDPRは, 直接アドレス指定方式の命令オペランドのbit8～bit15を指定します。

Figure 2-19. 直接アドレス指定方式における物理アドレスの生成



DPR は 8 ビット長で , リセット時には "01_H" に初期化されます。DPR は命令で読み書き可能です。

2.4.6 バンクレジスタ(PCB, DTB, ADB, USB, SSB)

各バンクレジスタは, プログラム空間, データ空間, ユーザスタック空間, およびアディショナルデータ空間に配置されるメモリバンクを示します。

2.4.6.1 バンクレジスタ

すべてのバンクレジスタはバイト長です。各バンクレジスタ (PCB, DTB, USP, SSP, ADB) は , PC, DT, SP(ユーザ), SP(システム) または AD 空間が割り当てられるメモリバンクを示します。

PCB 以外のバンクレジスタは読み書き可能です。PCB は , 読出しはできますが書込みはできません。PCB レジスタは , JMPP または CALLP 命令実行時 , 16M バイト全空間への分岐時 , RETP または RETI 命令実行時 , または割込み時に書き換わります。

バンクレジスタの動作の詳細は , 「2.3.3 バンク方式によるアドレス指定」を参照してください。

- プログラムカウンタバンクレジスタ(PCB)
  初期値 : リセット後は $0F_H$, その後はユーザプログラム開始時のリセットベクタの値。
- データバンクレジスタ(DTB)
  初期値 : 00_H
- ユーザスタックバンクレジスタ(USB)
  初期値 : 00_H
- システムスタックバンクレジスタ(SSB)
  初期値 : 00_H
- アディショナルデータバンクレジスタ(ADB)
  初期値 : 00_H

2.5 汎用レジスタ

F^{2}MC-16FX CPUのレジスタには, 専用レジスタと汎用レジスタの2種類があります。
ここでは, F^{2}MC-16FX CPUの汎用レジスタ(GPR)について説明します。
GPRは, 専用レジスタと同様, アドレスを使用せずにアクセスできます。レジスタの動作は専用の命令によって,定義されます。さらに, GPRはCPUアドレス空間内でもアクセスできます。また, 通常のメモリとして使用することもできます。

2.5.1 レジスタバンク

レジスタバンクは8ワードで構成され, 算術演算用の汎用レジスタとして使用できます。

2.5.1.1 レジスタバンク

レジスタバンクは 8 ワードで構成され , 以下のような算術演算用の汎用レジスタとして使用できます。

- バイトレジスタR0～R7 (8ビット)
- ワードレジスタRW0～RW7 (16ビット)
- ロングワードレジスタRL0～RL3 (32ビット)

さらに , レジスタバンクのレジスタはポインタおよびカウンタとして使用できます。

Table 2-5 に , レジスタの機能一覧を示します。Table 2-6 では , レジスタ間の関係を示しています。

Table 2-5. レジスタの機能

R0 ～ R7	各種命令のオペランドとして使用。 (注意事項) R0 はシフトまたはノーマライズ (正規化) 命令のカウンタとしても使用します。
RW0 ～ RW7	ポインタとして使用。 各種命令のオペランドとして使用。 (注意事項) RW0 はストリング命令のカウンタとしても使用します。
RL0 ～ RL3	ロングポインタとして使用。 各種命令のオペランドとして使用。

Table 2-6. レジスタ間の関係

	RW0	RL0
	RW1	
	RW2	RL1
	RW3	
R0	RW4	RL2
R1		
R2	RW5	
R3		
R4	RW6	RL3
R5		
R6	RW7	
R7		

通常の RAM 領域と同様，レジスタバンクの値はリセットで初期化されず，リセット前の状態が保持されます。電源投入時にはレジスタバンクの値は不定値となります。

2.5.2 汎用レジスタのアドレス指定

F^{2}MC-16FXの汎用レジスタは, レジスタバンクポインタ(RP)を使って現在使用中のレジスタバンクを指定します。レジスタバンクは, メモリ空間の"00:0180$_H$"～"00:037F$_H$"でアドレス指定することができます。

2.5.2.1 汎用レジスタのアドレス指定

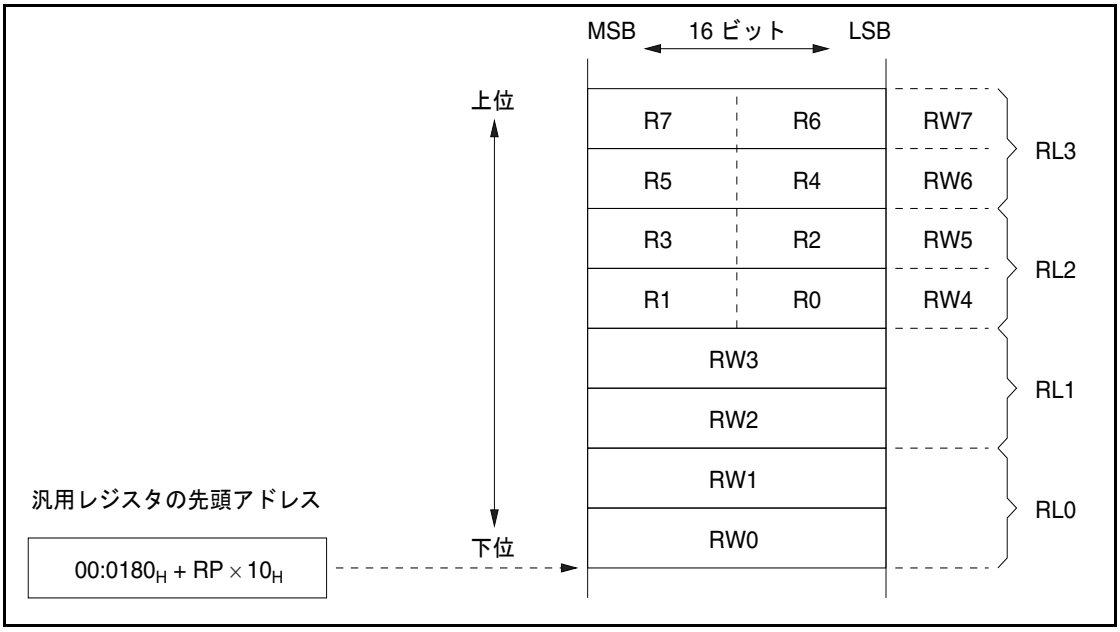
F^{2}MC-16FX の汎用レジスタは , メモリ空間の "000180$_H$" ～ "00037F$_H$" に配置され , レジスタバンクポインタ (RP) によってこのアドレスのどの部分が現在使用中のレジスタバンクであるかを指定します。各バンクには以下に示す 3 種類のレジスタが存在します。これらのレジスタは , Figure 2-20 に示すように相互依存関係があります。

R0 ～ R7: 8 ビットの汎用レジスタ

RW0 ～ RW7: 16 ビットの汎用レジスタ

RL0 ～ RL3: 32 ビットの汎用レジスタ

Figure 2-20. 汎用レジスタ



バイトレジスタとワードレジスタの上位 / 下位バイトの関係は以下の式で表すことができます。

$RW_{(i+4)} = R_{(i\times2+1)} \times 256 + R_{(i\times2)}$ [i=0 ～ 3]

RLi と RW の上位 / 下位バイトの関係は以下の式で表すことができます。

$RL_{(i)} = RW_{(i\times2+1)} \times 65536 + RW_{(i\times2)}$ [i=0 ～ 3]

2.6 プリフィックスコード

命令の前にプリフィックスコードを置くことで , 命令動作の一部を変更することが可能です。プリフィックスコードには以下に示す 3 つの種類があります。

- バンクセレクトプリフィックス
- コモンレジスタバンクプリフィックス
- フラグ変化抑止プリフィックス

各プリフィックスについて説明します。

2.6.1 バンクセレクトプリフィックス

バンクセレクトプリフィックスを命令の前に設定した場合は, 命令がアクセスするメモリ空間をアドレス指定方式とは無関係に設定できます。

2.6.1.1 バンクセレクトプリフィックス

データアクセスに使用するメモリ空間は , アドレス指定方式別に指定します。

バンクセレクトプリフィックスを命令の前に設定した場合は, 命令でデータアクセスに使用するメモリ空間をアドレス指定方式とは無関係に選択できます。

Table 2-7に, バンクセレクトプリフィックスと対応するメモリ空間を示します。

Table 2-7. バンクセレクトプリフィックス

バンクセレクトプリフィックス	選択される空間
PCB	PC空間
DTB	データ空間
ADB	AD空間
SPB	スタックフラグの値によってSSPまたはUSP空間が使用されます。

以下の命令では , プリフィックスの効果が各命令で異なることに注意してください。

- 転送命令(I/Oアクセス)

MOV A, io	MOV io, A	MOVW A, io	MOVW io, A
MOV io, #imm8	MOV io, imm16	MOVX A, io	

指定されているバンクセレクトプリフィックスに関係なく I/O 空間がアクセスされます。

- ビット操作命令(I/Oアクセス)

MOVB A, io:bp	MOVB io:bp, A	SETB io:bp	CLRB io:bp
BBC io:bp, rel	BBS io:bp, rel	WBTC io:bp	WBTS io:bp

指定されているバンクセレクトプリフィックスに関係なく , I/O 空間がアクセスされます。

■ 分岐命令

RETI			

指定されているバンクセレクトプリフィックスに関係なく , システムスタックバンク (SSB) が使用されます。

■ ストリング操作命令

FILS	FILSW	SCEQ	SCWEQ
MOVS	MOVSW		

指定されているバンクセレクトプリフィックスに関係なく , オペランドで指定されているバンクレジスタが使用されます。

■ スタック操作命令

PUSHW	POPW	POPW PS
LINK	UNLINK	

指定されているバンクセレクトプリフィックスに関係なく , S フラグの設定に従ってシステムスタックバンク (SSB) またはユーザスタックバンク (USB) が使用されます。

■ プロセッサステータスワード変更/割込み制御命令

AND CCR, #imm8	OR CCR, #imm8	MOV ILM, #imm8	POPW PS

指定されたバンクセレクトプリフィックスの効果は次の命令まで及びます。

2.6.2 コモンレジスタバンクプリフィックス(CMR)

コモンレジスタバンクプリフィックス(CMR)をレジスタバンクにアクセスする命令の前に設定すると, 命令の対象となっているレジスタをコモンレジスタバンクに変更することができます。これは, 現在のレジスタバンクポインタ(RP)の値に関係なく, RP=0のときに選択されるレジスタバンクになります。

2.6.2.1 コモンレジスタバンクプリフィックス(CMR)

複数のタスク間でのデータ交換を容易にするためには , そのときの RP 値に関係なく既定の同じレジスタバンクに比較的簡単にアクセスできる手段を用意することが必要です。この目的のため , F^2MC-16FX では各タスクが共通して使用できる「コモンレジスタバンク」とよばれるレジスタバンクが提供されています。コモンバンクは , $00:0180_H$ ～ $00:018F_H$ 番地にあります。RP の値が "0" に設定されているときに選択されます。

CMR がレジスタバンクにアクセスする命令の前に置かれた場合 , 現在の RP 値に関係なく , その命令はコモンバンクにアクセスします。

以下の命令では , プリフィックスの効果が各命令で異なることに注意してください。

- ストリング操作命令

FILS	FILSW	SCEQ	SCWEQ
MOVS	MOVSW		

上記のストリング命令の前に CMR プリフィックスを置いても無視されます。CMR プリフィックスが指定されているかどうかに関係なく , レジスタバンクポインタ RP の実際の値を使って , カウンタレジスタ RW0 が常に参照されます。

- スタック操作命令

LINK	UNLINK

指定されているバンクセレクトプリフィックスに関係なく , S フラグの設定に従ってシステムスタックバンク (SSB) またはユーザスタックバンク (USB) が使用されます。

- プロセッサステータスワード変更/割込み制御命令

AND CCR, #imm8	OR CCR, #imm8	MOV ILM, #imm8	POPW PS

指定されたプリフィックスの効果は次の命令まで及びます。

2.6.3 フラグ変化抑止プリフィックス(NCC)

命令の実行に伴うフラグの変化は, その命令の前にフラグ変化抑止プリフィックスコード(NCC)を置くことによって抑止できます。

2.6.3.1 フラグ変化抑止プリフィックス(NCC)

フラグ変化を抑止するためには , フラグ変化抑止プリフィックスコード (NCC) を用います。NCC を命令の前に置くことで , 命令実行に伴うフラグ変化を抑止できます。

以下の命令では , プリフィックスの効果が各命令で異なることに注意してください。

- ストリング操作命令

FILS	FILSW	SCEQ	SCWEQ

上記のストリング命令の前に NCC プリフィックスを置いても無視されます。NCC プリフィックスが指定されているかどうかに関係なく , CCR のフラグは命令の指定によって変化します。

- プロセッサステータスワード変更/割込み制御命令

AND CCR, #imm8	OR CCR, #imm8	MOV ILM, #imm8	POPW PS

指定されたプリフィックスの効果は次の命令まで及びます。

- 分岐命令

INT #vct8	INT9	INT addr16	INTP addr24
RETI			

NCC プリフィックスが指定されているかどうかに関係なく , CCR は命令の指定によって変化します。

- コンテキストスイッチ命令

JCTX @A

プリフィックスに関係なく , CCR は命令の指定によって変化します。

2.6.4 プリフィックスコードに関する制約

実行中に割込み要求を受け付けない命令について記載します。プリフィックスコードが割込みを受け付けない命令の前に置かれた場合, この後に最初の命令が実行されるまでプリフィックスコードの設定は有効になります。
競合するプリフィックスコードが連続した場合, 最後のプリフィックスコードのみが有効となります。

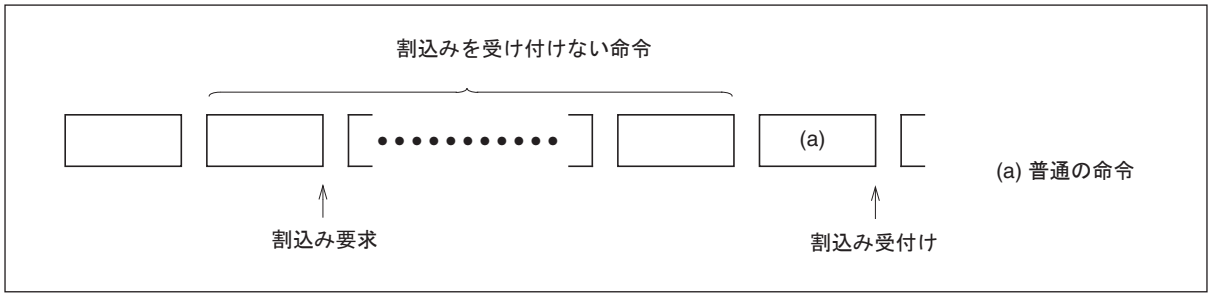
2.6.4.1 割込みを受け付けない命令

割込み要求は, 以下の命令およびプリフィックスコードの実行中は受け付けられません。

MOV ILM, #imm8	AND CCR, #imm8	OR CCR, #imm8	POPW PS
PCB	ADB	DTB	SPB
NCC	CMR		

上記の命令のいずれかの実行中に有効な割込み要求が発生した場合, 上記以外の命令が実行されるまで割込みは延期されます。詳細は , Figure 2-21 を参照してください。

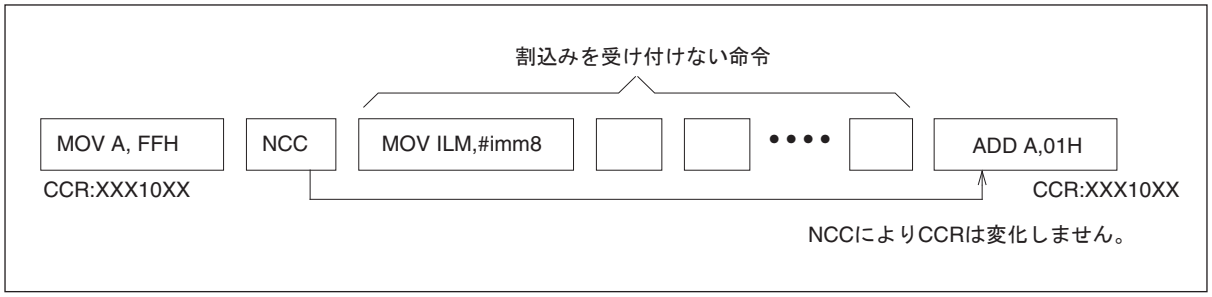
Figure 2-21. 割込みを受け付けない命令



2.6.4.2 割込みを受け付けない命令およびプリフィックスコードに関する制約

プリフィックスコードが割込みを受け付けない命令の前に置かれた場合 , プリフィックスコードは , 割込み受付可能な , 最初の命令に影響します。詳細は ,Figure 2-22 を参照してください。

Figure 2-22. 割込みを受け付けない命令およびプリフィックスコード

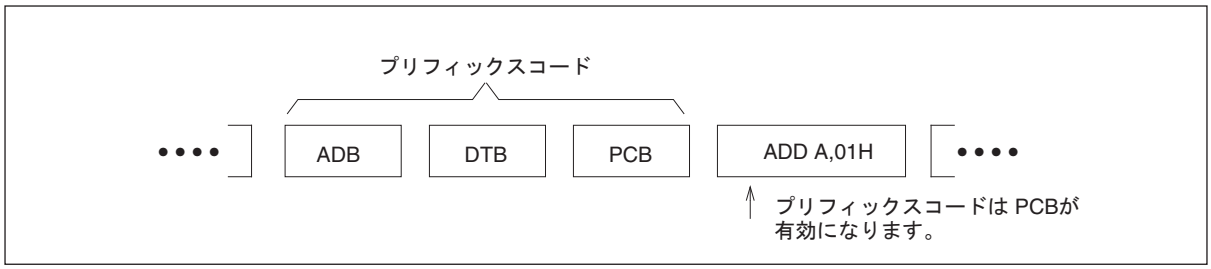


2.6.4.3 プリフィックスコードの連続

競合するプリフィックスコードが連続した場合 , 最後のものが有効となります。

下図では , 競合するプリフィックスコードは PCB, ADB, DTB および SPB になります。詳細は , Figure 2-23 を参照してください。

Figure 2-23. プリフィックスコードの連続



プリフィックスCMRおよびNCCは, バンクセレクトプリフィックスPCB, ADB, DTBおよびSPBと競合しません。また, CMRはNCCとも競合しません。

3. 割込み

割込みの機能と動作について説明します。

3.1 割込みの概要

F^{2}MC-16FX は , イベントなどの発生により現在実行中のプログラムを中断して , 別のプログラムに制御を移す割込み機能があります。割込み機能は以下の 4 つに分類することができます。

- ハードウェア割込み:内部リソースのイベント発生による割込み処理
- ソフトウェア割込み:ソフトウェアのイベント発生(命令)による割込み処理
- 例外:動作例外事項の発生による処理
- DMA:内部リソースのイベント発生によるCPUとのやり取りを伴わないデータ転送

3.1.1 ハードウェア割込み

ハードウェア割込みは , 内部リソースからの割込み要求によって発生します。ハードウェア割込み要求は , 内部リソース内の割込み要求フラグと割込み許可フラグがセットされている場合に発生します。

■ 割込みレベルの指定

ハードウェア割込みには , 割込みレベルを指定することができます。割込みレベルを指定するには , 割込み制御レジスタ ICR のレベル設定ビット (IL0, IL1, IL2) を使用します。

ハードウェア割込みには , 個別に割込みレベル (IL) を指定することができます。専用 IL に対するアクセスは , IX インデックスを設定することで実行できます。IX と IL は共に , 割込み制御レジスタ ICR を介してアクセスできます。

■ ハードウェア割込み要求マスク

ハードウェア割込み要求は , I フラグおよび ILM ビット (ILM0, ILM1, ILM2) を使用してマスクすることができます。割込みは , I フラグがセットされ , 割込みレベル IL の値が割込みレベルマスク ILM よりも小さい場合にのみ実行されます。さらに , ハードウェア割込みを受け付けるには P フラグがセットされている必要があります。P, I, および ILM は , CPU のプロセッサステータスワード PS の一部です。

マスクのかかっていない割込み要求が発生すると , CPU は PS, PC, PCB, DTB, ADB, DPR, A の各レジスタからなる 12 バイトのデータをシステムスタックバンクおよびポインタレジスタ (SSB と SSP) が示すメモリ領域に退避させます。

3.1.2 ソフトウェア割込み

ソフトウェア割込みは INT 命令の実行により要求されます。INT 命令により要求される割込みには , 割込み要求フラグや割込み許可フラグはありません。INT 命令の実行により , 常に割込み要求が発生します。

INT 命令には割込みレベルの割当てもありません。このため , INT 命令使用時には ILM は更新されません。その代わり , I フラグがクリアされ , 以降の割込み要求が保留状態になります。

3.1.3 例外

以下のソフトウェア例外が処理されます。

- 未定義命令
- INT9
- INTE(EVAデバイスのみで利用可能)

以下のハードウェア例外が処理されます。

- NMI
- HW-INT9(埋込みデバッグサポート)
- DSUブレーク要因(EVAデバイスのみで利用可能)

例外処理は基本的に割込みと同じであり , 命令の実行中に例外事項が発生したことを検出した段階で , 例外処理を行います。一般的に , 例外処理は予想外の動作の結果に発生するものですので , プログラムのデバッグや緊急時のリカバリソフトウェアの起動などに限定して使用することをお勧めします。

3.1.4 ダイレクトメモリアクセス(DMA)

■ DMA機能

F^2MC-16FX には , 割込み時に周辺リソースとメモリ間で自動的にデータ転送する DMA 機能があります。

DMA チャネルの数はデバイスに依存します。

指定された数の DMA データ転送が完了すると , 転送元の IRQ チャネルに対して割込み処理プログラムが自動実行されます。そのような DMA 完了時の割込み処理方法は , 標準のハードウェア割込みと同じです。

DMA の詳細は , 「第 4 章 DMA」を参照してください。

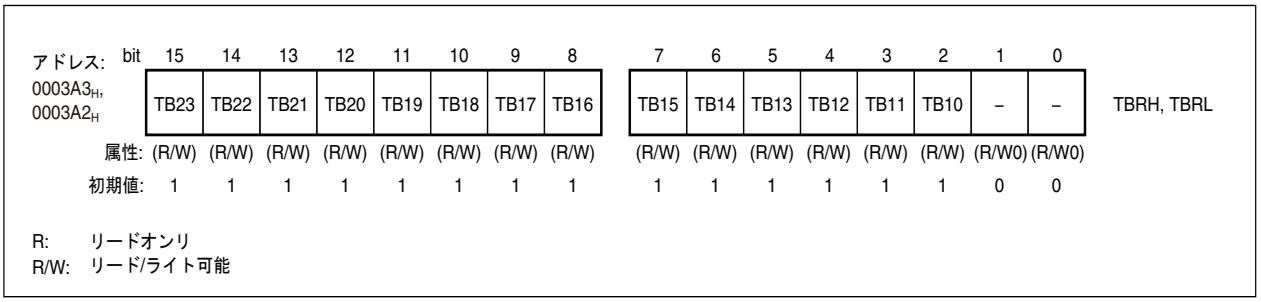
3.2 割込みベクタ

ハードウェア割込みとソフトウェア割込みは同じベクタテーブルを使用します。割込み処理ルーチンは , IRQ ラインをアサート , または INT 命令の実行によって割込みベクタ番号を指定することで実行できます。割込みベクタは , Table 3-2 で示されている番地の間に割り当てられます。割込みベクタテーブルの位置はテーブルベースレジスタで選択できます。

3.2.1 割込みベクタ

- 割込みベクタテーブルベースレジスタ(TBR)

Figure 3-1. 割込みベクタテーブルベースレジスタ (TBR)



テーブルベースレジスタを使えば , 1K バイト単位で割込みベクタテーブルを任意のメモリ位置に再配置できます。

TBR の値によって , 割込みベクタテーブルの開始アドレス 24 ビット中の最上位 14 ビット TB23 ～ TB10 が決まります。最下位ビット TB9 ～ TB0 は "0" に固定となります。

テーブルベースレジスタ TBR はリセット時に "$FFFC_H$" に初期化され , その結果 , 初期テーブルベース TB は "$FFFC00_H$" となります。

割込みベクタテーブルは1Kバイト(256ベクタ)の容量があります。

Table 3-1. TBRの例

TBRの値	割込みベクタテーブルの開始アドレス (テーブルベース)	割込みベクタテーブルの終了	コメント
$FFFC_H$	FF.FC00H	FF.FFFFH	F2MC-16LXの場合と同じ (初期値)
$FB00_H$	FB.0000H	FB.03FFH	ROMバンクFBの始点
$00FC_H$	00.FC00H	00.FFFFH	バンク00(外部メモリ可)の終点
0010_H	00.1000H	00.13FFH	RAM領域内部
0000_H	00.0000H	00.03FFH	IO領域のため使用禁止

3.2.2 割込みベクタテーブル

割込み処理中に参照される割込みベクタテーブルは , メモリ領域の "256×TBR" ～ "$256×TBR+3FF_H$" に割り当てられます。リセット時のデフォルトは , ベクタテーブルの "$FFFC00_H$" ～ "$FFFFFF_H$" になります。ベクタテーブルを ROM メモリの先頭に置かない場合は , 別の TBR 値を設定する必要があります。

ハードウェア割込み , 例外 , ソフトウェア割込みはすべて同じベクタテーブルを使用します。そのため , 割込みサービスルーチンはハードウェア割込みと対応するソフトウェア割込みのどちらからでも呼び出すことができます。

割込みサービスルーチンの開始アドレスの3バイトはすべて適切な割込みベクタ(VecAddr = 4×(255-INT#) + 256×TBR)に書き込む必要があります。

Table 3-2. 割込みベクタテーブル

割込み / ベクタ番号	ベクタアドレス	ICR中のレベルレジスタのインデックス	ハードウェアIRQ / 割込み要因
INT 0 CALLV 0/1 (*1)	TB+3FC$_H$	--	--
INT 1 CALLV 2/3 (*1)	TB+3F8$_H$	--	--
INT 2 CALLV 4/5 (*1)	TB+3F4$_H$	--	--
INT 3 CALLV 6/7 (*1)	TB+3F0$_H$	--	--
INT 4 CALLV 8/9 (*1)	TB+3EC$_H$	--	--
INT 5 CALLV 10/11 (*1)	TB+3E8$_H$	--	--
INT 6 CALLV 12/13 (*1)	TB+3E4$_H$	--	--
INT 7 CALLV 14 (*1)	TB+3E0$_H$	--	--
INT 8 MODE Byte	TB+3DC$_H$	--	リセット
INT 9	TB+3D8$_H$	--	INT9命令
INT 10	TB+3D4$_H$	--	未定義命令例外
INT 11	TB+3D0$_H$	--	NMI
INT 12	TB+3CC$_H$	IL12	遅延割込み
INT 13	TB+3C8$_H$	IL13	CRクロックタイマ
INT 14	TB+3C4$_H$	IL14	メインクロックタイマ
INT 15	TB+3C0$_H$	IL15	サブクロックタイマ
INT 16	TB+3BC$_H$	IL16	<PLLロック解除用に予約>
INT 17	TB+3B8$_H$	IL17	デバイス特有の周辺機能
INT 18	TB+3B4$_H$	IL18	
INT 19	TB+3B0$_H$	IL19	
...	...	...	
INT 254	TB+004$_H$	--	--
INT 255	TB+000$_H$	--	--

(*1)プログラムバンクレジスタ(PCB)がTBRHと同じ場合,CALLV命令のベクタ領域はINT#0～INT#7命令のベクタテーブルと重なります。この場合，CALLV命令がINT#0～INT#7命令と同じアドレスを使用しないようにするか, INT#0～INT#7を使用しないようにしてください。

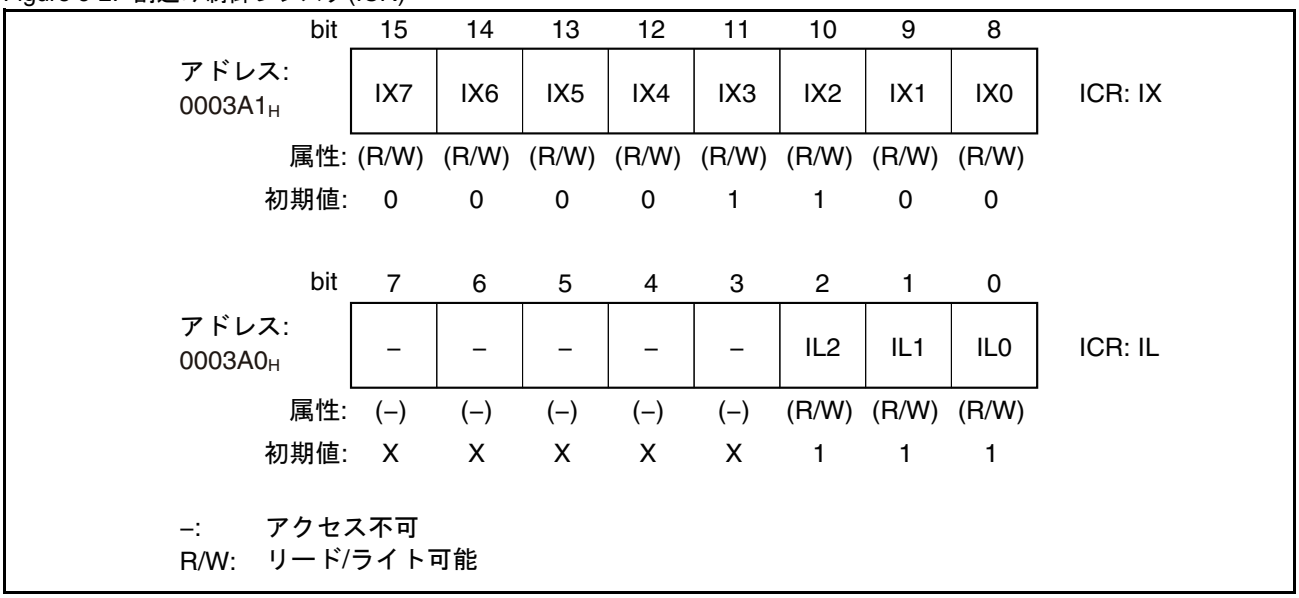
3.3 割込み制御レジスタ(ICR)

割込み機能を持つ周辺リソースにはそれぞれ割込み制御レジスタ(ICR)があります。割込み制御レジスタは, 割り当てられる周辺リソースに割込みレベル(IL)を設定します。

3.3.1 割込み制御レジスタ(ICR)

Figure 3-2は, 割込み制御レジスタのビット構成図です。

Figure 3-2. 割込み制御レジスタ(ICR)



ICR [bit15 ～ bit8] IX7 ～ IX0 : アクセスする割込みレベル (IL) のインデックス

内部リソースに対応する割込みレベルのインデックスを指定する読み書き可能なビットです。このインデックスでアクセスする割込みレベルを選択します。IL[n] は周辺割込み要求番号 IRQ[n] に属し , IL[n] および IRQ[n] は共に割込み INT[n] に関連付けられています。

システム割込み INT0 ～ INT11 の優先度は固定で , 割込みレベルが存在しません。インデックス 12 以下の割込みレベルに対して書込みを行うと無視され , 読出しを行うと不定値が返されます。これと同じことが , 利用できないハードウェア割込み (デバイス依存の最大割込み番号より上の番号) にもいえます。

IX はリセット時に 06_H に初期化されます。

Figure 3-3 に , IX ポインタを使ってレベルレジスタにアクセスする例を示します。点線で囲まれた IX および選択 IL は , ICR の実際の内容を表しています。

レベル設定の読み書きは , IX が指し示す IL に対して行います。IL にレベル設定を書き込むには , ワードアクセスによって事前または同時に IX に対してインデックスの書き込みを行います。IL の読出しを行うには , 読み出す前に IX に書き込みを行う必要があります。

並行タスクを使用する場合の注意事項 :

割込みレベル情報にアクセスする並行タスクの場合 , インデックスアクセスの処理には注意が必要です。

- ICR:IX および ICR:IL に情報を同時に書き込むには , ワードアクセスを使用します。
- 読出しアクセス時には , ワードアクセスを使用してインデックス ICR:IX をセットし , ICR レジスタ全体を読み出します。ICR:IX の値と読み出すはずのインデックスが一致していることを確認して , ICR:IL が正しく入力されたかどうかを確認してください。

ICR [bit7 ～ bit3] 未使用ビット

読出しアクセスを行うと不定値が返されます。

これらのビットには常に "0" を書き込みます。

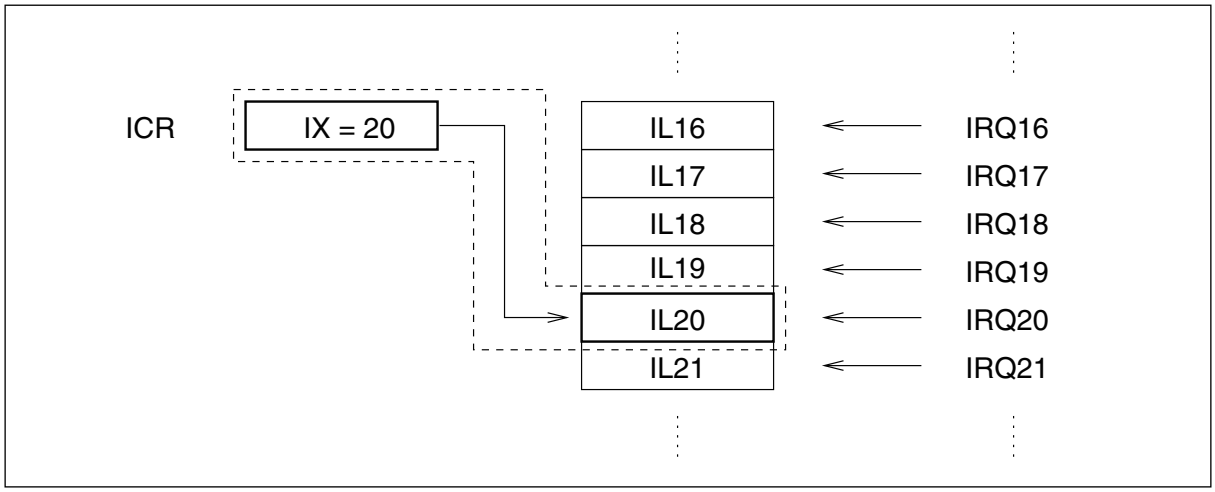
ICR [bit2 ～ bit0] IL2 ～ IL0 : 割込みレベル設定ビット

対応する内部リソースの割込みレベルを指定する読み書き可能ビットです。リセット時，レベル7 (割込みなし)に初期化されます。Table 3-3に, 割込みレベル設定ビットと割込みレベルの関係を示します。

Table 3-3. 割込みレベル設定ビットと割込みレベル

IL2	IL1	IL0	レベル
0	0	0	0 (最も強い)
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6 (最も弱い)
1	1	1	7 (割込みなし)

Figure 3-3. インデックス(IX), レベル(IL)およびIRQ番号の間の関係, IX = 20の場合の例



3.4 マスク不可能割込み(NMI)

F^{2}MC-16FX CPUには, マスク不可割込みがあります。外部NMI端子の機能を有効化, レベルの設定, NMI要求を終了するフラグの使用ができます。

3.4.1 NMI制御ステータスレジスタ(NMI)

Figure 3-2は, NMI制御ステータスレジスタのビット構成図です。

Figure 3-4. NMI制御ステータスレジスタ(NMI)

	bit	15	14	13	12	11	10	9	8
アドレス: 0003A5$_H$		–	–	–	–	FIX9	LEV	EN	FLAG
	属性:	(–)	(–)	(–)	(–)	(R/W)	(R/W)	(R/W1)	(R/W0)
	初期値:	–	–	–	–	0	1	0	X

– :アクセス不可
R/W :リード/ライト可能
R/W1 :リード可能。ビットはセットのみ可能
R/W0 :リード可能。ビットはクリアのみ可能

NMI [bit15 ～ bit12] 未使用ビット

読出しアクセスを行うと不定値が返されます。
これらのビットには常に "0" を書き込みます。

NMI [bit11] FIX9: INT9 ベクタの位置の固定

FIX9 ビットは , INT9 の割込みベクタへの代替位置に変わります。もし , "1" に設定されていると , 割込みベクタは, アドレス0F:FFD8$_H$より得られます。もし, "0"に設定されていれば, INT9ベクタの位置は, TBR+3D8$_H$により定義されます。リセットで , FIX9 ビットは , "0" に初期化されます。この機能は , デバイス起動時に実行されるファームウェアにより使用されます。TBR から INT9 の割込みベクタを独立させることで , 組込みデバッグ支援ユニット (EDSU) の信頼性が向上します。

NMI [bit10] LEV: NMI 端子の信号アクティビティレベル

LEV ビットでは , NMI 端子信号のアクティビティレベルを指定します。値を "1" にすると NMI 端子は正論理入力 , "0" にすると負論理入力であることを意味します。

EN ビットがセットされていない場合 , LEV ビットは読み書き可能となります。EN ビットがセットされている場合 , LEV の状態はロックされます。その場合 , LEV は読出し専用となり , EN=1 のときに LEV に書き込んでも無視されます。

リセット時に LEV ビットは "1" に初期化されるため , NMI 端子はデフォルトでは正論理となります。

NMI [bit9] EN: マスク不可能割込み機能許可ビット

EN ビットによって , 専用 NMI 端子機能が使用できるようになります。EN が "0" に設定されている場合 , デバイスに NMI は存在しません。この場合 , NMI 入力に信号レベルの変更があっても CPU は反応しません。端子はその他の目的に使用できます。EN が "1" に設定されている場合 , NMI 端子が有効になります。CPU は , (LEV ビットで指定された)NMI 端子でアクティブ信号レベルが検出されると NMI の例外処理に分岐します。

EN が "1" に設定されると , LEV および EN ビットは書込み不可になります。いったん NMI 機能が有効にな

ると , 信号レベルの変更も NMI の無効化もできなくなります。この状態では , EN ビットを "0" に変更できるのはデバイスリセットだけとなります。

EN ビットはリセット時に "0" に初期化されるため , デフォルトでは NMI 機能は無効になっています。

LEV および EN ビットは , 同時に起動 (同じアクセスで変更) してはいけません。EN は , 後で個別に有効にする必要があります。そうしないと , スパイクフィルタの緩和時間が原因で NMI が発生することがあります。

NMI [bit8] FLAG: マスク不可能割込みフラグ

FLAG ビットには , NMI 端子で NMI が発生したときの非同期イベントが格納されます。

スパイクを抑制するため , 短いパルスを除去するスパイクフィルタが使用されています。パルスの極性は , LEV ビットの設定によって決まります。

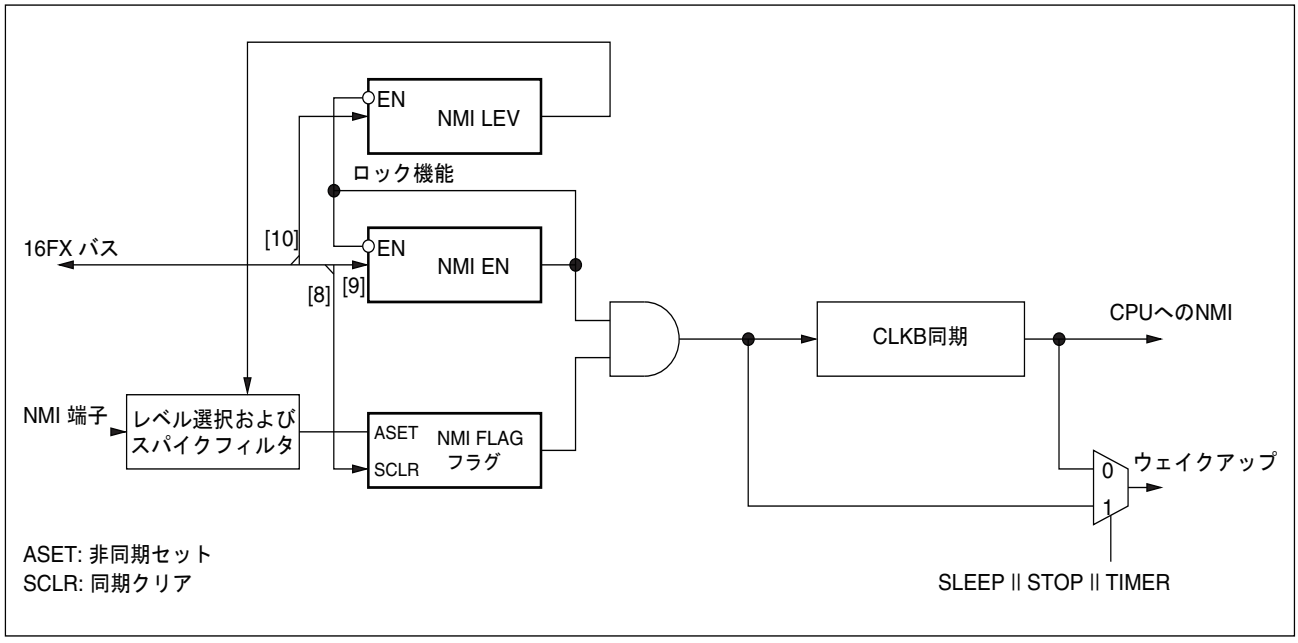
FLAG ビットは , ハードウェアのイベント (NMI 発生) によってセットされ , ソフトウェアによってクリアします。クリアされると割込みは終了します。割込みは , EN ビットと FLAG の両方がセットされている場合にのみ発生します。

FLAG ビットは読出しとクリアができます。FLAG に "1" を書き込んでも無視されます。

FLAG ビットのリセット後の値は不定です。このフラグは , NMI を有効にする前にクリアする必要があります。

ビット操作では , このフラグにリードモディファイライトで読出しを行うと常に "1" が返されます。

Figure 3-5. NMI制御ステータスレジスタの操作



3.5 割込みフロー

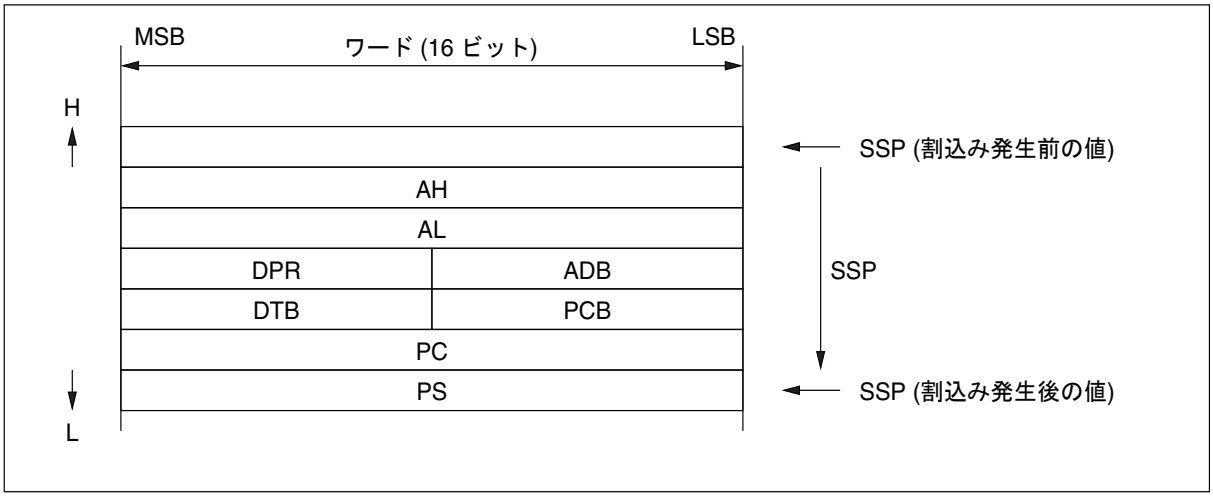
Figure 3-7に, 割込みフローを示します。

3.5.1 割込みフロー

ハードウェア割込み, ソフトウェア割込み, 例外が発生すると, 割込み処理フローが実行されます。割込みフローチャートの詳細は , Figure 3-7 を参照してください。

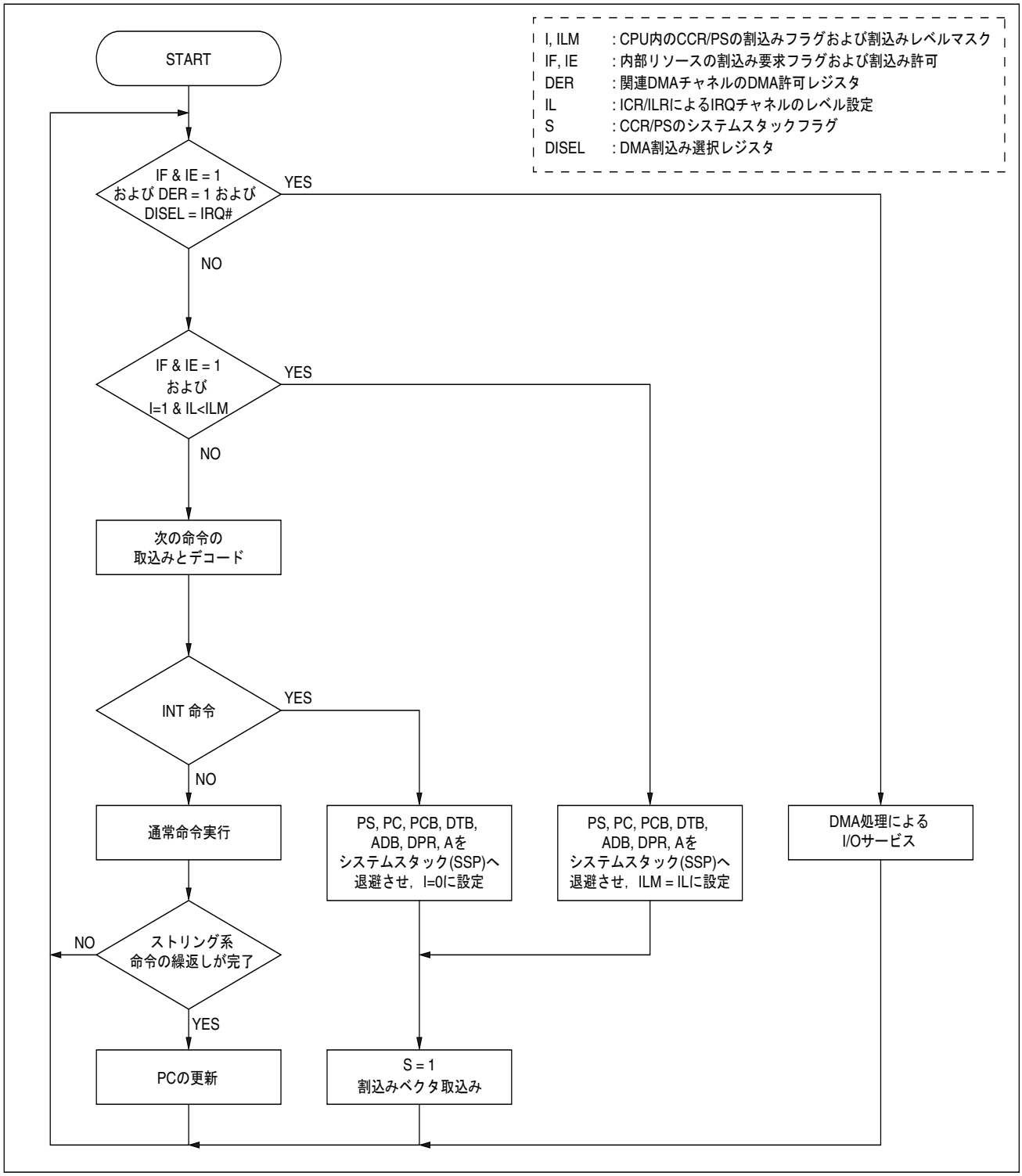
割込み処理の前に , CPU 専用レジスタがスタックに退避します (Figure 3-6 参照してください)。

Figure 3-6. 割込み処理中のレジスタ退避



割込み処理の終了時 , RETI 命令が実行される中で CPU レジスタのコンテキストが復帰します。CPU は通常のプログラム実行状態に戻ります。

Figure 3-7. 割込みフロー



3.6 ハードウェア割込み

内部リソースからの割込み要求信号に対応して, CPUは現在実行中のプログラムを一時中断し, ユーザの定義した割込み処理用プログラムへ制御を移行させます。

3.6.1 ハードウェア割込み

ハードウェア割込みは , 以下の 2 つの処理の結果 , 条件が満たされると発生します。

- 割込み要求レベル(IL)と, CPU内のPSに設定された割込みレベルマスクレジスタ(ILM)値の比較
- PSのIフラグ値に対するハードウェア参照

ハードウェア割込みが発生すると , CPU は以下の処理を行います。

- CPU内部のPC, PS, AH, AL, PCB, DTB, ADB, DPRレジスタ値のシステムスタックへの退避
- Sフラグのセット
- PSレジスタ内のILMの設定。現在要求されている割込みレベル(IL)が自動的に設定される。
- 対応する割込みベクタの読出しおよびその値で示された処理への分岐

デバイスがスタンバイモードのとき , IL<7 のハードウェア割込みがあるとクロックおよびモード制御ユニットに対するウェイクアップイベントが発生します。

3.6.2 ハードウェア割込みシステムの構造

割込みの状態は , 内部リソース , 割込みコントローラの ICR, および CPU の PS 値によって示されます。ハードウェア割込みを使用するには以下の設定を行います。

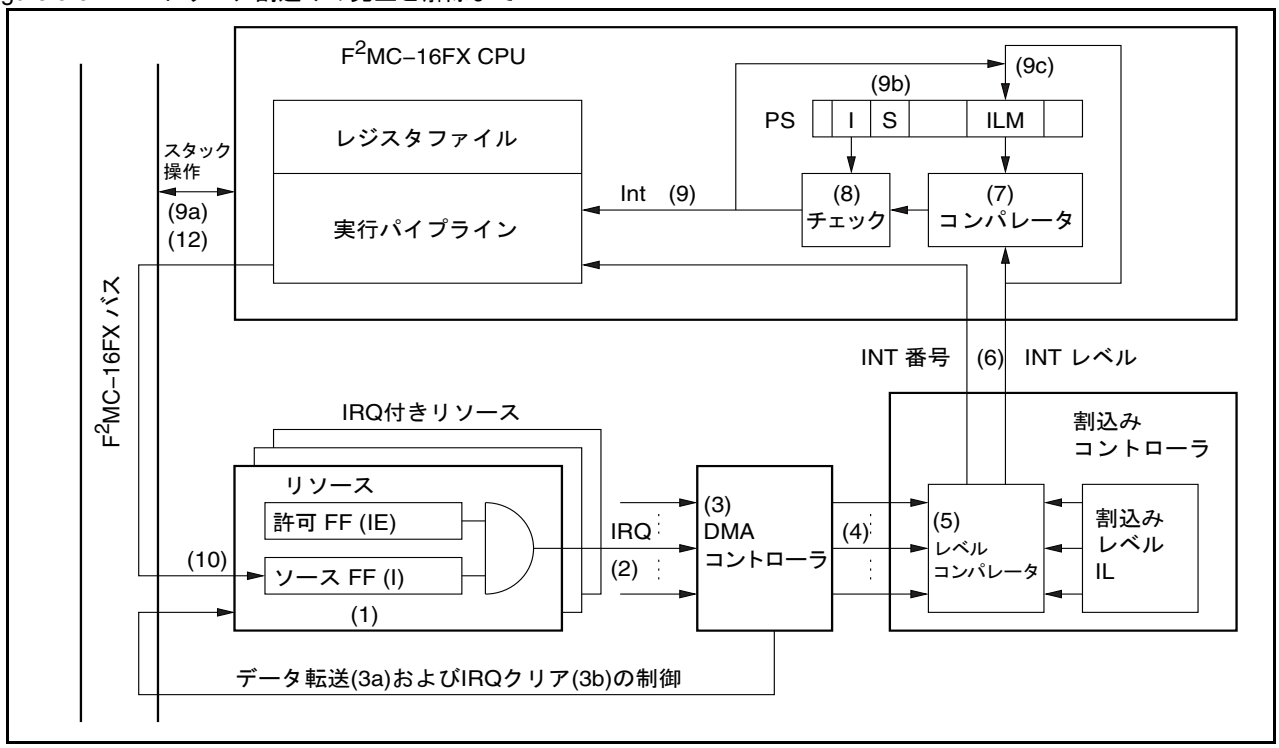
- 割込みベクタ(メモリ内)
  - ベクタテーブルをデフォルト以外の箇所に配置する場合はTBR値を考慮します。
  - 割込みサービスルーチンの開始アドレスは, 対応する割込みベクタ(VecAddr = 4×(255 - INT#) + 256×TBR)に書き込む必要があります。
- 周辺リソース
  - 割込み許可ビットおよび要求ビットを使用して, 周辺リソースからの割込み要求を制御します。
- 割込みコントローラ
  - 発生する可能性のある割込みに対して割込みレベル(ICR:IL)を指定します。
  - 複数の割込みが同時発生した場合, 割込みレベルが小さいものほど優先度が高くなります。IL=7では割込みが無効になります。
  - 同じレベルの要求が複数ある場合, 割込みコントローラは割込み番号が最も小さい要求を選択します。同じレベルが設定されている場合, 割込み番号とは関係なく, 遅延割込みの優先度が最も低くなります。
  - 割込み要求とILの間には一定の関係があります。各ハードウェア割込み要求IRQ[n]のレベルはIL[n] によって指定することができます（n >= 12です）。
- CPU
  - PSレジスタ内のILMとIは,要求された割り込みレベル(IL)と現在の割り込みレベルマスク(ILM)の比較と, 割り込み許可(I)の判定に使用されます。ハードウェア割込みの受付けは, Iフラグがセットされていること, ILMがILよりも大きいことが条件になります。
  - 割込み処理中, CPUは12バイトをSSBおよびSSPが示すメモリ領域に退避させます。そのため, 割込みを使用する前にシステムスタックポインタを初期化する必要があります。
  - CPUは割込みベクタの3バイトを取り出してPCとPCBに設定します。これは,割込み処理ルーチンがその位置から開始する必要があるためです。その結果, 次に実行する命令は,ユーザの定義した割込み処理プログラムになります。RETI命令の実行で通常動作が再開します。

3.6.3 ハードウェア割込みの動作

周辺リソースからの割込み要求 (IRQ) は , 割込みコントローラで処理を行う前に DMA コントローラに出力されます。DMA コントローラは , DMA チャネル設定 (DMA 割込み要求選択レジスタ DISEL および DMA 許可 DER:ENx ビット) に従って , IRQ を DMA 転送で処理するか , 割込みコントローラで処理するかを決定します。DMA 転送は , I フラグと割込みレベルの状態に関係なく受け付けられます。DMA コントローラでは優先度の順位が固定しており , チャネル 0 で最も高い優先度 , チャネル 15 で最も低い優先度となります。

Figure 3-8 に , 割込み処理プログラムによるハードウェア割込みの発生から割込み要求の解除までの処理フローを示します。

Figure 3-8. ハードウェア割込みの発生と解除まで



(1) リソース内で割込み要因が発生します。

(2) リソース内の割込み許可ビットを参照します。割込み許可になっていれば , リソースが割込み要求 (IRQ) を発生します。

(3) DMA コントローラが IRQ を DMA で処理すべきかチェックします。DMA コントローラは、各 DMA チャンネル毎に IRQ の割込み番号が選択されているか、または DMA が許可になっているかを判定します。

- 判定が「DMA で処理を行う」場合 , 転送は DMA (3a) が処理します。判定が「DMA で処理を行わない」となった場合 , 割込み処理は割込みコントローラが行うことになり , (4) に続きます。
- DMA 転送の終了後 , 割込みビットがリソース (3b) でクリアされます。
- 最終転送カウントに達すると , 割込みコントローラで DMA 完了割込みが処理されます。

(4) 割込みコントローラが割込み要求を受け取ります。

(5) 割込みコントローラは同時に要求のあった割込みの優先順位を判定します。

(6) 割込みコントローラは最も優先度の高い割込みレベルおよび対応する割込み番号を CPU に転送します。

(7) 割込みコントローラが要求した割込みレベルとプロセッサステータスレジスタの ILM 値が比較されます。

(8) 比較の結果 , 要求されている割込み処理のレベルが現在のレベルより優先順位が低い (IL<ILM) 場合 , 同じプロセッサステータスレジスタ内の I フラグの内容がチェックされます。

(9) 8 のチェックで I フラグが割込み許可状態だった場合 , 現在実行中の命令の実行が終了し次第 , 割込み処理が実行されます。

- CPUの状態を退避させるため, 専用CPUレジスタがシステムスタック(9a)に転送されます。
- Sフラグが"1"に設定されます(9b)。
- 要求されたレベルがILMビット(9c)に書き込まれます。
- 割込みベクタが取り出されます。
- 制御が割込み処理ルーチンに移ります(割込みベクタとして読み出されたアドレスへの分岐)。

(10) ユーザの割込み処理ルーチン内で (1) の割込み要因がソフトウェアによってクリアされた場合 , 割込み要求は終了します。

(11) RETI 命令を最後の命令として , 割込み処理ルーチンから復帰します。

(12) システムスタックに退避していた CPU の状態が復元され , 通常のプログラム実行が再開します。

3.6.4 ハードウェア割込みの処理時間

以下に , CPU が割込み処理 (スタック操作 , 割込みベクタ取込み , 割込みベクタへの分岐) に必要な時間を示します。以下は , スタック操作および割込みベクタ取込みがサイクルのウェイト時間なしで実行された場合の数値です。

- 割込み起動: 10サイクル + c
- 割込み復帰: 9サイクル + c (RETI命令)

Table 3-4. 割込み処理時のサイクル数の補正値(c)

スタックポインタが示しているアドレス	補正値
内部領域が偶数アドレスの場合	0
内部領域が奇数アドレスの場合	+2

また , バス転送のサイクルのウェイト時間がある場合 (例 : 遅い ROM メモリまたは外部領域のベクタテーブルへのアクセス), それも加算する必要があります。

3.7 ソフトウェア割込み

一定の命令が実行されると, CPUは実行中のプログラムからユーザ定義の割込み処理用プログラムに制御を移します。これをソフトウェア割込み機能とよびます。ソフトウェア割込みはソフトウェア割込み命令を実行すると必ず発生します。

3.7.1 ソフトウェア割込み

INT 命令により要求されるソフトウェア割込みには , 割込み要求フラグや割込み許可フラグはありません。ソフトウェア割込み要求は , INT 命令の実行により常に発生し , 常に受け付けられます。

INT 命令には割込みレベルはありません。したがって , INT 命令は ILM を更新せず , I フラグをクリアして以降のハードウェア割込み要求を保留にします。

ソフトウェア割込みが発生すると , CPU は以下の処理を行います。

- CPU内部のPC, PS, AH, AL, PCB, DTB, ADB, DPRレジスタのシステムスタックへの退避
- Sフラグのセット
- PSレジスタ内のIのクリア。ハードウェア割込みは自動的に無効になります。
- 対応する割込みベクタの取り出しおよびその値で示された処理への分岐

3.7.2 ソフトウェア割込みシステムの構造

ソフトウェア割込みは CPU 内ですべて処理されます。ソフトウェア割込みを使用するには , 以下の設定を行います。

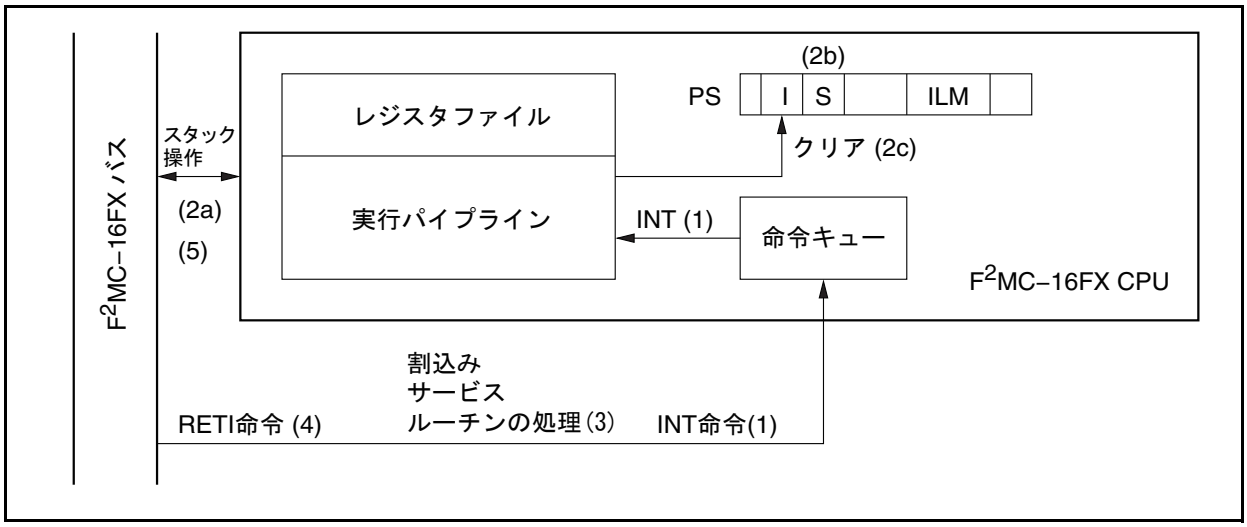
- 割込みベクタ(メモリ内)
  - ベクタテーブルをデフォルト以外の箇所に配置する場合はTBR値を考慮します。
  - 割込みサービスルーチンの開始アドレスは, 対応する割込みベクタ(VecAddr = 4×(255 - INT#) + 256×TBR）に書き込む必要があります。
- CPU
  - 割込み処理中, CPUは12バイトをSSBおよびSSPが示すメモリ領域に退避させます。そのため, 割込みを使用する前にシステムスタックポインタを初期化する必要があります。
  - CPUは割込みベクタの3バイトをフェッチし, PCとPCBに取り出します。これは,割込み処理ルーチンがその位置から開始する必要があるためです。その結果, 次に実行する命令は,ユーザの定義した割込み処理プログラムになります。RETI命令の実行で通常動作が再開します。

3.7.3 ソフトウェア割込みの動作

CPU がソフトウェア割込み命令をフェッチして実行すると , ソフトウェア割込み処理シーケンスが始まります。このシーケンスは , SSB と SSP の示すメモリ領域へ 12 バイト (PS, PC, PCB, DTB, ADB, DPR, A) を退避させた後 , 割込みベクタを 3 バイト取込み , PC と PCB へ格納し , I フラグをリセット , S フラグをセットして分岐処理を行います。その結果 , 次にユーザ指定の割込み処理プログラムが実行されます。

Figure 3-9 に , ソフトウェア割込みの発生から割込み要求の解除までの処理フローを示します。

Figure 3-9. ソフトウェア割込みの発生と解除まで



(1) ソフトウェア割込み命令を実行します。

(2) ソフトウェア割込み命令に対応した割込み処理が CPU によって実行されます。

- CPUの状態を退避させるため, 専用CPUレジスタがシステムスタック(2a)に転送されます。
- Sビットが"1"に設定されます(2b)。
- Iフラグがクリアされ, ハードウェア割込みが無効になります(2c)。
- 割込みベクタが取り出されます。
- 制御が割込み処理ルーチンに移ります(割込みベクタとして読み出されたアドレスへの分岐)。

(3) 割込みサービスルーチンが CPU によって処理されます。

(4) ユーザの割込み処理ルーチン内の RETI 命令で割込み処理が終了します。

(5) CPU は , 専用レジスタのコンテキストをシステムスタックから復元させます。

(6) INT 命令の後 , CPU は次の命令を実行してプログラムを続行します。

3.8 多重割込み

F^{2}MC-16FX CPUは, 多重割込み(同時発生する割込み処理および入れ子になった割込み処理)をサポートしています。

3.8.1 多重ハードウェア割込み

割込み処理実行中に優先度の高いハードウェア割込み (レベル値が小さい) が発生した場合は , 現在実行中の命令が終了した後に優先度の高い割込みに制御が移ります。優先度の高い割込み処理が終了した後は , 最初に実行していた割込み処理が再開します。

割込み処理実行中に , 現在実行中の割込みと同等か , 低い優先度の割込みが発生した場合は , I フラグまたは割込みレベルマスクレジスタ (ILM) を命令で変更しないかぎり , 現在の割込み処理が終了するまで新しい割込み要求は保留になります。

DMA 転送は中断不可で , 多重起動できません。DMA 転送の処理中は , ほかの DMA 要求はすべて保留されます。DMA コントローラに対する要求が同時発生した場合 , 最も小さいチャネル番号が最初に処理されます。

DMA の詳細は , 「第 4 章 DMA」を参照してください。

3.8.2 多重ソフトウェア割込み

ソフトウェア割込みが同時発生することはありません。ソフトウェア割込みは INT 命令を実行することで発生し , 常に受け付けられます。ただし , INT 命令を割込みサービスルーチン内に置けば入れ子になったソフトウェア割込みを実行することができます。

3.8.3 割込みの優先順位

以下の表に, すべての割込みの受付け条件を記載しています。

Table 3-5. 割込み優先順位の制御

イベント	INT#	種類	レベル	受付け条件	受け付けられた場合の動作	
ハードウェアイベント	-	命令ブレーク (VEIB) *システム予約*	P2	現在の命令の実行が終了して以下が成立する場合 ILM>2 || P == 1	CPU の状態をシステムスタックに退避 S = 1 割込みベクタに分岐	P = 0 ILM = 2
	-	ツールブレーク (VENMI) *システム予約*	P2	ストリング命令に割込みが行われている , または , 現在の命令の実行が終了して以下が成立する場合 ILM>2 || P == 1		P = 0 ILM = 2
	11	NMI	P4	ストリング命令に割込みが行われている , または , 現在の命令の実行が終了して以下が成立する場合 ILM>4 || P == 1		P = 0 ILM = 4
	9	アドレス一致検出 (HW-INT9)	P6	ストリング命令に割込みが行われている , または , 現在の命令の実行が終了して以下が成立する場合 ILM>6 || P == 1		P = 0 ILM = 6
	13 以降	周辺 IRQ	IL U0...U7	ストリング命令に割込みが行われている , または , 現在の命令の実行が終了して以下が成立する場合 ILM > IL P == 1 I == 1 同じ IL の要求が複数ある場合は , 最も小さい IRQ 番号の要求が受け付けられる。		ILM = IL
	12	遅延 INT	IL U0...U7	ストリング命令に割込みが行われている , または , 現在の命令の実行が終了して以下が成立する場合 ILM > IL P == 1 I == 1 同じ IL で保留中の周辺 IRQ がない。		ILM = IL
ソフトウェアイベント (命令)	-	ソフトウェア命令ブレーク (INTE) *システム予約*	P2	常に受付け		P = 0 ILM = 2 I = 0
	9	INT9	-			I = 0
	10	未定義命令による例外	-			I = 0
	すべて	INT 命令	-			I = 0
	-	RETI 命令	-		CPU の状態の復元 (P, I, S, ILM を含む)	

IL および ILM : 割込みレベルおよびマスク
I : 割込み許可フラグ (周辺割込み)
S : システムスタックフラグ
P : 特権モードフラグ (PS:CCR の bit7)

以下の表に，割込みレベルの名称，対応するPフラグおよびILM値を示します。また，割込みレベルを要求することがある割込み要因も記載します。

Table 3-6. 割込みレベル

名称	カテゴリ	P フラグ	ILM の値	優先度	使用命令
P0	特権モード	0	0	最高	-
P1		0	1		-
P2		0	2		DSU
P3		0	3		-
P4		0	4		NMI
P5		0	5		-
P6		0	6		HW-INT9
P7		0	7		-
U0	ユーザモード	1	0		周辺
U1		1	1		
U2		1	2		
U3		1	3		
U4		1	4		
U5		1	5		
U6		1	6		
U7		1	7	最低	要求なし

3.9 例外

F^{2}MC-16FXでは, 様々なソフトウェアおよびハードウェアイベントの発生時に例外処理が行われます。

3.9.1 ソフトウェア例外(opコード)

ソフトウェア例外は常に受け付けられます。ソフトウェア割込みと同じく , ソフトウェア例外はすべてのハードウェア割込みの受付けを無効にします。ソフトウェア例外は , 以下のような OP コードの実行によって発生します。

3.9.1.1 未定義命令の実行

命令マップで定義されていないコードはすべて未定義命令として処理されます。未定義命令が実行されると , ソフトウェア割込み命令 INT#10 と同じような処理が実行されます。具体的には , スタックに退避している PC 値のアドレスに未定義命令が格納されます。処理は RETI 命令によって復帰しますが , 同じ例外が再度発生するので復帰不能になります。

動作 :

(SSP)<-(SSP)-2, ((SSP))<-(AH)

(SSP)<-(SSP)-2, ((SSP))<-(AL)

(SSP)<-(SSP)-2, ((SSP))<-(DPR):(ADB)

(SSP)<-(SSP)-2, ((SSP))<-(DTB):(PCB)

(SSP)<-(SSP)-2, ((SSP))<-(PC)

(SSP)<-(SSP)-2, ((SSP))<-(PS)

(S)<-1, (I)<-0

(PCB)<- ベクタ #10 のアドレス (上位バイト)

(PC)<- ベクタ #10 のアドレス (下位ワード)

3.9.1.2 INT9

この命令により , ベクタ #9 によって示される割込み処理ルーチンに分岐します。割込みルーチンで RETI 命令を実行することで , INT9 命令後の処理に復帰します。

動作 :

(SSP)<-(SSP)-2, ((SSP))<-(AH)

(SSP)<-(SSP)-2, ((SSP))<-(AL)

(SSP)<-(SSP)-2, ((SSP))<-(DPR):(ADB)

(SSP)<-(SSP)-2, ((SSP))<-(DTB):(PCB)

(SSP)<-(SSP)-2, ((SSP))<-(PC)+1

(SSP)<-(SSP)-2, ((SSP))<-(PS)

(S)<-1, (I)<-0

(PCB)<- ベクタ #9 のアドレス (上位バイト)

(PC)<- ベクタ #9 のアドレス (下位ワード)

3.9.1.3 INTE (システム予約。DSUのみで利用可能)

INTE は , インサーキットエミュレータ (ICE) を使用したデバッグシステムでソフトウェアブレークポイントを挿入する際に使用します。ソフトウェア命令ブレークの挿入時に , 元の命令の最初のバイトが INTE と置き換わります。

この命令によって , DSU が定義する固定ベクタで示された割込み処理ルーチンに分岐します。INTE は , スタックに退避している PC の値のアドレスに格納されます。割込みルーチンで RETI 命令を実行すると , そこで処理が復帰します (ソフトウェアブレークポイントを削除すれば INTE と元の命令を置き換えることができます)。

特権モードフラグ (P フラグ) はクリアされ , ILM レジスタは 2 に設定されます (レベル P2)。これにより , すべてのハードウェア割込みおよび例外が無効になります。PフラグおよびILMは, RETI命令の実行時に復帰します。

動作 :

(SSP)<-(SSP)-2, ((SSP))<-(AH)

(SSP)<-(SSP)-2, ((SSP))<-(AL)

(SSP)<-(SSP)-2, ((SSP))<-(DPR):(ADB)

(SSP)<-(SSP)-2, ((SSP))<-(DTB):(PCB)

(SSP)<-(SSP)-2, ((SSP))<-(PC)

(SSP)<-(SSP)-2, ((SSP))<-(PS)

(S)<-1, (I)<-0, (P)<-0, (ILM)<-2

(PCB)<-DSU 指定の固定ベクタ (上位バイト , 00_H, DSU はアドレスを無視)

(PC)<-DSU 指定の固定ベクタ (下位ワード , 0400_H, DSU はアドレスを無視)

DSU がなければ , INTE は未定義命令による例外と同じように処理されます。割込みベクタ #10 が参照され , P フラグはクリアされず , ILM は更新されません。

3.9.2 ハードウェア例外(マスク不可能割込み)

ハードウェア例外は外部イベントのため , ソフトウェア命令ではマスクできません。現在処理されているものよりもレベル番号が高いハードウェア例外は , RETI 命令の実行によって以前のレベルに戻るまで保留されます。さらに , ハードウェア例外はすべてのハードウェア割込みの受付けを無効にします。

複数のハードウェア例外が同時に発生する場合, VEIB > VENMI > NMI > HW-INT9の優先順位で受け付けられます。ハードウェア例外は , 現在の割込みレベルマスクおよび P フラグの設定で許可されている場合 , 各命令の実行終了後およびストリング命令の実行中に受け付けられます。

3.9.2.1 HW-INT9

HW-INT9 は , アドレス一致検出機能で使用されます。この機能により , 内蔵デバッグサポート (オペランドアドレスブレークまたはデータ値ブレーク) や簡単なメモリ保護が実現されています。

特権モードフラグ (P フラグ) はクリアされ , ILM は 6 に設定されます (レベル P6)。これにより , 周辺からのすべてのハードウェア割込みが無効になります。P フラグおよび ILM は , RETI 命令の実行時に復帰します。

動作 :

(SSP)<-(SSP)-2, ((SSP))<-(AH)

(SSP)<-(SSP)-2, ((SSP)<-(AL)

(SSP)<-(SSP)-2, ((SSP))<-(DPR):(ADB)

(SSP)<-(SSP)-2, ((SSP))<-(DTB):(PCB)

(SSP)<-(SSP)-2, ((SSP))<-(PC)

(SSP)<-(SSP)-2, ((SSP))<-(PS)

(S)<-1, (P)<-0, (ILM)<-6

(PCB)<- ベクタ #9 のアドレス (上位バイト)

(PC)<- ベクタ #9 のアドレス (下位ワード)

3.9.2.2 NMI

NMI は , 外部ハードウェアの例外処理を行います。

特権モードフラグ (P フラグ) はクリアされ , ILM は 4 に設定されます (レベル P4)。これにより , 周辺からのすべてのハードウェア割込みおよび HW-INT9 が無効になります。P フラグおよび ILM は , RETI 命令の実行時に復帰します。

動作 :

(SSP)<-(SSP)-2, ((SSP))<-(AH)

(SSP)<-(SSP)-2, ((SSP))<-(AL)

(SSP)<-(SSP)-2, ((SSP))<-(DPR):(ADB)

(SSP)<-(SSP)-2, ((SSP))<-(DTB):(PCB)

(SSP)<-(SSP)-2, ((SSP))<-(PC)

(SSP)<-(SSP)-2, ((SSP))<-(PS)

(S)<-1, (P)<-0, (ILM)<-4

(PCB)<- ベクタ #11 のアドレス (上位バイト)

(PC)<- ベクタ #11 のアドレス (下位ワード)

3.9.2.3 ツールブレーク(VENMI。システム予約。DSUのみで利用可能)

VENMI は DSU を使ったデバッグ専用で , 様々なブレーク要因を実装しています。

特権モードフラグ (P フラグ) はクリアされ , ILM は 2 に設定されます (レベル P2)。これにより , すべてのハードウェア割込みおよび例外が無効になります。P フラグおよび ILM は , RETI 命令の実行時に復帰します。

動作 :

(SSP)<-(SSP)-2, ((SSP))<-(AH)

(SSP)<-(SSP)-2, ((SSP))<-(AL)

(SSP)<-(SSP)-2, ((SSP))<-(DPR):(ADB)

(SSP)<-(SSP)-2, ((SSP))<-(DTB):(PCB)

(SSP)<-(SSP)-2, ((SSP))<-(PC)

(SSP)<-(SSP)-2, ((SSP))<-(PS)

(S)<-1, (P)<-0, (ILM)<-2

(PCB)<-DSU 指定の固定ベクタ (上位バイト , 00_H, DSU はアドレスを無視)

(PC)<-DSU 指定の固定ベクタ (下位ワード , 0400_H, DSU はアドレスを無視)

3.9.2.4 命令ブレーク(VEIB。システム予約。DSUのみで利用可)

VEIB は , DSU を使ったデバッグ専用で , 実行前の命令ブレークを実装しています。VEIB は他のハードウェア例外とは反対に , ストリング命令の実行中は受け付けられません。

4. DMA

DMAの機能と動作について説明します。

4.1 概要

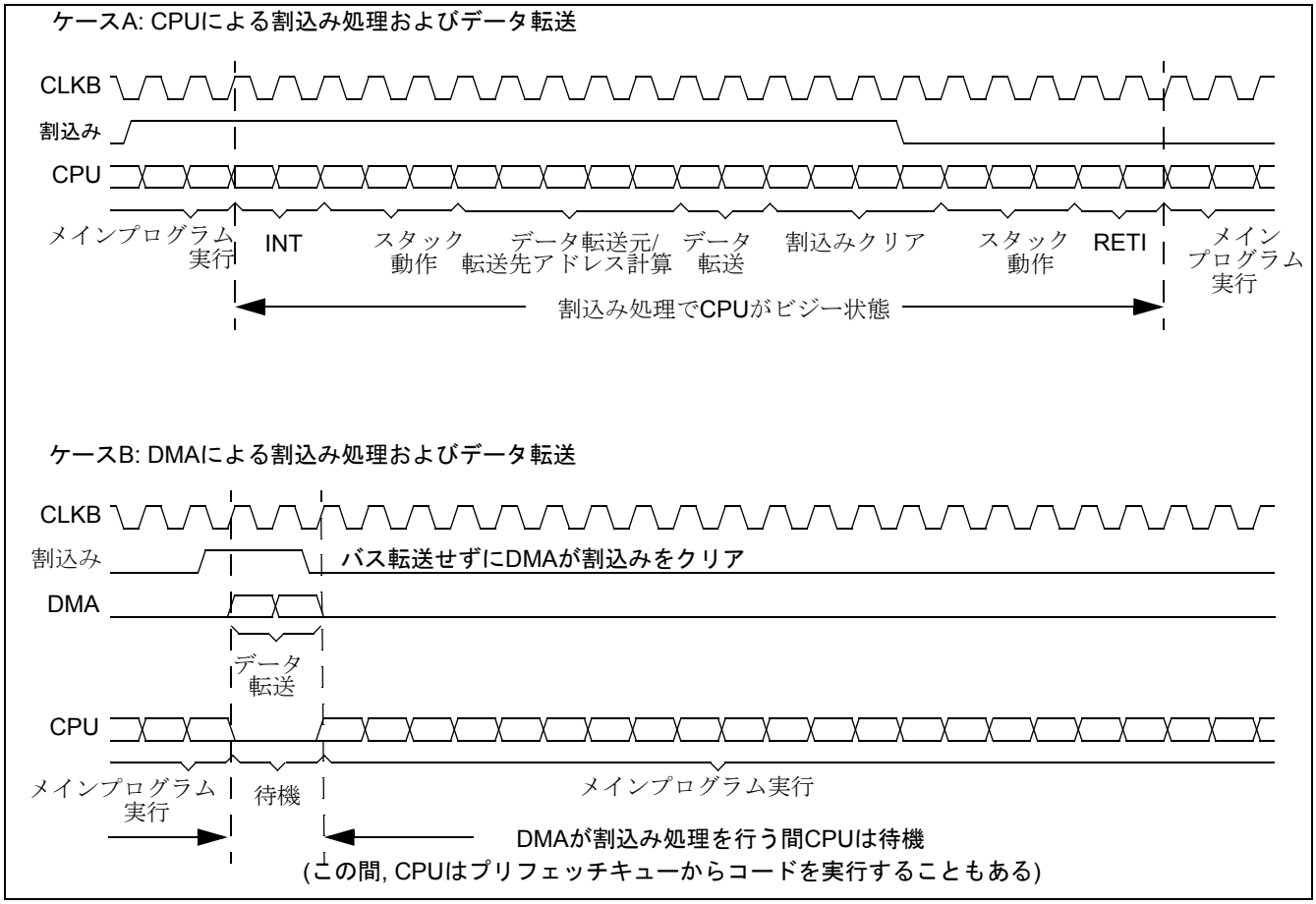
DMAは, CPUを介さずにメモリとメモリ/周辺リソースレジスタ間の自動データ転送を可能にします。DMAは16LXファミリのEI^{2}OSの性能を向上させ, 機能を拡張させたものです。
DMA機能について説明します。

4.1.1 割込み処理と比較したDMA:

DMA には割込みによる転送と比べて以下の利点があります (Figure 4-1 参照してください)。

- DMAは割込み処理を行うCPUの負荷を軽減させる。
- CPUレジスタが転送に使用されないため, レジスタを退避させる必要がなく, その結果として転送速度が上がる。
- CPUバスは転送動作だけにしか使用されないため, 転送速度が上がる。

Figure4-1.CPUとDMAの割込み処理の比較(原則のみ。クロックサイクル数はプログラムコードによって異なる場合がある。)



4.1.2 DMA機能

DMA コントローラには以下の機能があります。

- 周辺リソース(I/O)とメモリ間の自動データ転送
- DMA転送中のCPUバス割当て
- 優先度が固定されたDMA転送チャネルを16チャネルまで使用可能(DMAチャネル番号が小さいほど要求の優先度が高くなります)。
- 転送元および転送先アドレスをインクリメントまたはデクリメントするかどうかを選択可能(バッファアドレスまたはI/Oレジスタアドレスをインクリメントするか,デクリメントするか, そのままにするかを選択可能)。
- 選択可能な周辺リソースの割込みによってDMA転送を開始可能(ハードウェアIRQ)。
- DMA転送は以下によって制御されます。
  - DMA許可レジスタ(DER, bit0...bit15)
  - DMA割込み要求選択レジスタ(DISEL bit0...bit15)
  - DMA停止ステータスレジスタ(DSSR, bit0...bit15)
  - DMAステータスレジスタ(DSR, bit0...bit15)
  - DMAディスクリプタ(各8バイト。DCT, IOA, DMACS, BAP)
  - DMA IOA バンクレジスタ(IOABK)*1
- DMA転送は, 周辺リソース内のエラー条件によって停止させることができます。
- DMA転送の終わりに, 周辺リソースの割込みサービスルーチンに処理が自動的に分岐します。

*1: IOA バンクレジスタは , すべてのデバイスでは使用できません。データシートを参照してください。

4.1.3 構造

DMA コントローラには以下の機能があります。

- 周辺リソース(I/O)とメモリ間の自動データ転送
- DMA転送中のCPUバス割当て
- 優先度が固定されたDMA転送チャネルを16チャネルまで使用可能(DMAチャネル番号が小さいほど要求の優先度が高くなります)。
- 転送元および転送先アドレスをインクリメントまたはデクリメントするかどうかを選択可能(バッファアドレスまたはI/Oレジスタアドレスをインクリメントするか,デクリメントするか, そのままにするかを選択可能)。
- 選択可能な周辺リソースの割込みによってDMA転送を開始可能(ハードウェアIRQ)。
- DMA転送は以下によって制御されます。
  - DMA許可レジスタ(DER, bit0...bit15)
  - DMA割込み要求選択レジスタ(DISEL bit0...bit15)
  - DMA停止ステータスレジスタ(DSSR, bit0...bit15)
  - DMAステータスレジスタ(DSR, bit0...bit15)
  - DMAディスクリプタ(各8バイト。DCT, IOA, DMACS, BAP)
  - DMA IOA バンクレジスタ(IOABK)*1

- DMA転送は, 周辺リソース内のエラー条件によって停止させることができます。
- DMA転送の終わりに, 周辺リソースの割込みサービスルーチンに処理が自動的に分岐します。

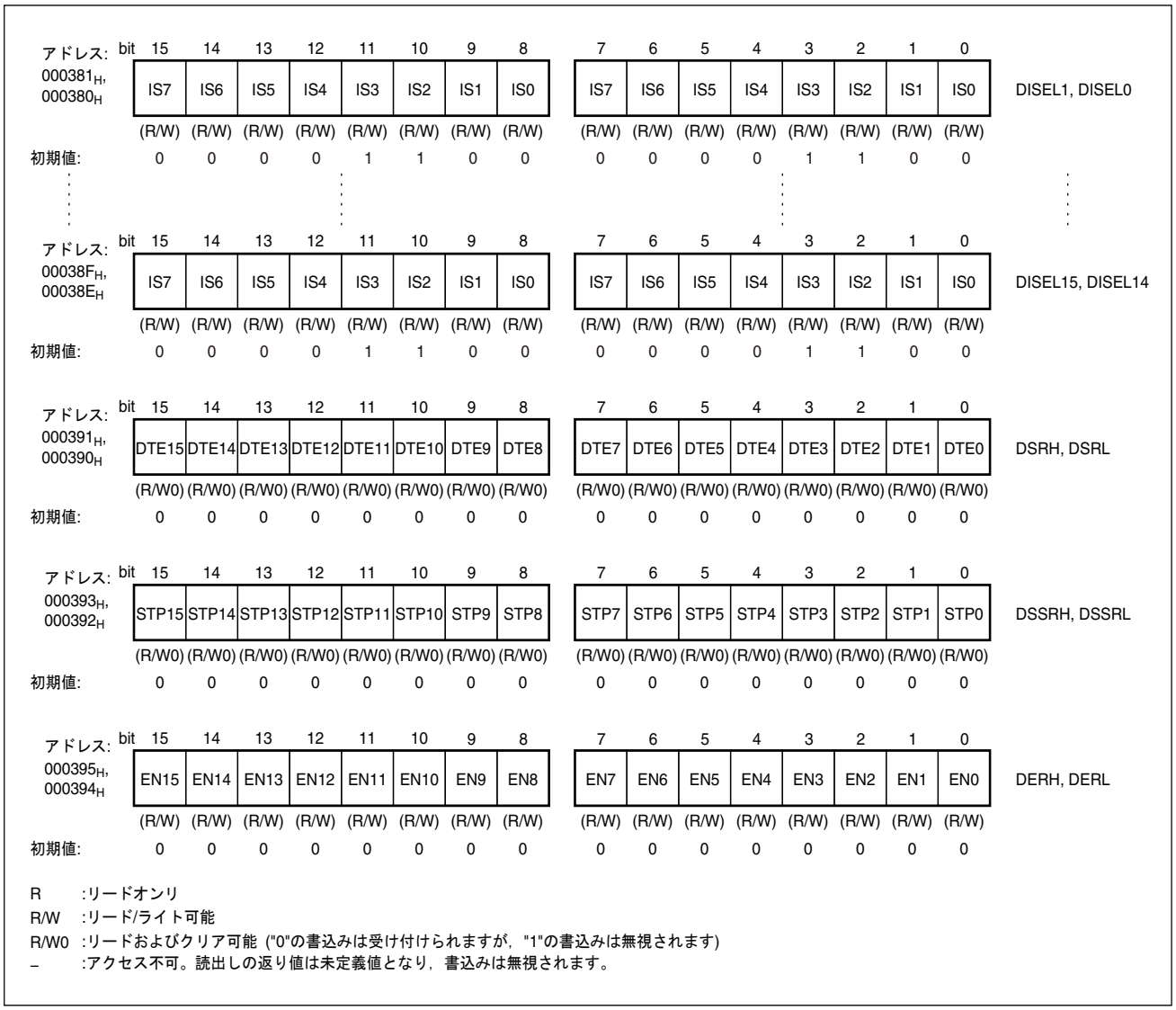
*1: IOA バンクレジスタは , すべてのデバイスでは使用できません。データシートを参照してください。

4.2 DMAレジスタ

DMAコントローラには, 3つのレジスタ(DER, DSR, DSSR。各DMAチャネルにつき1ビット)および割込みを選択する1つのレジスタ(DISEL。各DMAチャネルに1バイト)があります。
DMAディスクリプタは, DMA転送の設定に使用します。詳細は「4.3 DMAディスクリプタ」で説明します。

DMAのレジスター覧

Figure 4-2. DMAのレジスター覧



デバイスによって利用できないチャネルに関する注意事項:

搭載される DMA のチャネル数は , デバイスによって異なります。ご使用になるデバイスのデータシートを参照してください。

読出し操作時 , 使用できない DSR, DSSR および DER ビットは不定値を返します。

読出し操作時 , 使用できない DISEL レジスタは不定値を返します。

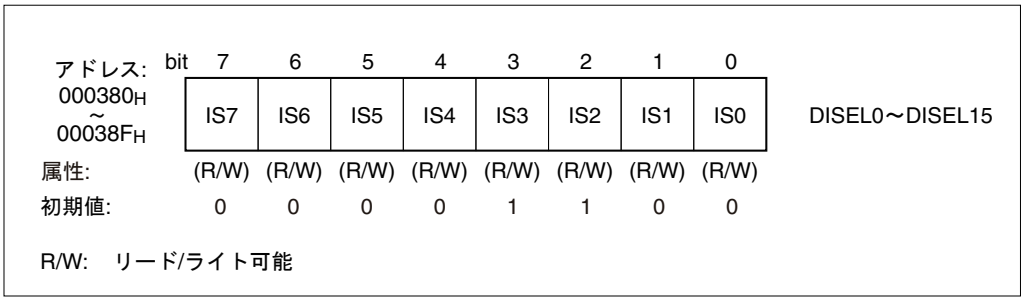
使用できないビット / レジスタには , "0" を書き込んでください。

4.2.1 DMA割込み要求選択レジスタ(DISEL)

各DMAチャネルには割込み要求選択レジスタ(DISEL0～DISEL15)が1つあります。このレジスタは, 対応するチャネルでDMA転送を起動させるIRQ番号を指定します。

4.2.1.1 DMA割込み要求選択レジスタ(DISEL)

Figure 4-3. DMA割込み要求選択レジスタ(DISEL)の構成



[bit7 ～ bit0] IS: 割込みの選択

各 DMA チャネルには DISELx レジスタが存在しており , このレジスタで当該チャネルの DMA 転送を起動させる IRQ 番号を指定します。選択可能な IRQ 番号の範囲は , "$0C_H$(12) " ～デバイス依存の最大数になります。

DMA 動作を起動できるのはハードウェア割込みだけです。

＜注意事項＞

DISEL レジスタのリセット後の値は "$0C_H$(12)" に初期化されます。使用しない（ディセーブル状態の) チャネルの DISEL レジスタには , 有効なチャネルの DISEL レジスタの値と同じ値を設定しないでください。初期値 "$0C_H$(12)" のままであれば問題ありません。

搭載される DMAC のチャネル数はデバイスによって異なります。搭載されておらず , 利用できないチャネルの DISEL レジスタは , 読出し操作時 , 不定値を返します。

複数の有効な DMA チャネルで同じ割込み番号を設定しないでください。

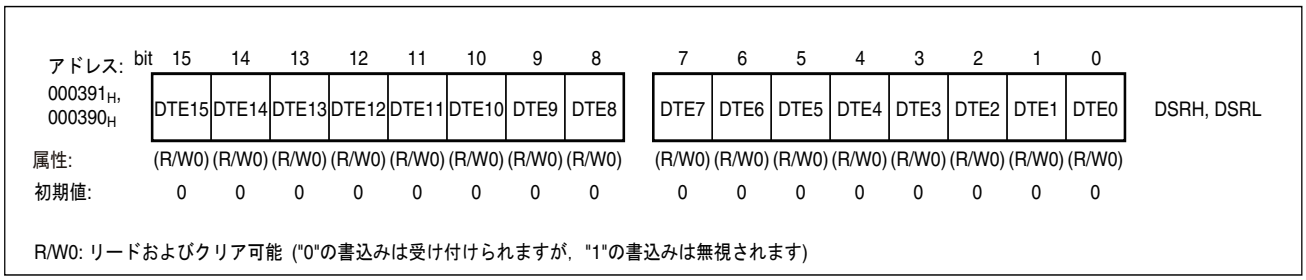
DISEL レジスタを更新する場合 , 関連 IRQ がアクティブでないこと , および DMA チャネルが無効になっていることを確認してください。

4.2.2 DMAステータスレジスタ(DSR)

DMAステータスレジスタ(DSR)について説明します。

4.2.2.1 DMAステータスレジスタ(DSR)

Figure 4-4. DMAステータスレジスタ(DSR)の構成



[bit15 ～ bit0] DTEx: データ転送終了割込み要求

各 DMA チャネルに DTE ビットが存在しており , データ転送の終了を示します。転送終了は , データカウントレジスタ (DCT) が 0 に達した場合 , あるいは周辺リソースからの停止要求が出された場合に成立します。DMA コントローラのハードウェアで DTE ビットをセットすることで , 割込みが要求されます。

DTExビット	機能
0 [初期値]	Iチャネルxに対する割込み要求は存在していないことを示します。
1	IDMA転送が完了または停止されていること,および割込みが要求されていることを示します。DTExがセットされると, このチャネルのDMA転送は無効になります。

＜注意事項＞

読出し時には , DSR レジスタの有効なチャネルの , すべての DTE ビットの状態が返されます。

読出し動作時 , 使用できないチャネルの DTE ビットは不定値を返します。

リードモディファイライト命令では , 読出し時にすべての DTE ビットで "1" が返されます。

このビットをセットできるのはハードウェアだけです。このビットに "1" を書き込んでも無視されます。

このビットに "0" を書き込むとクリアされます。

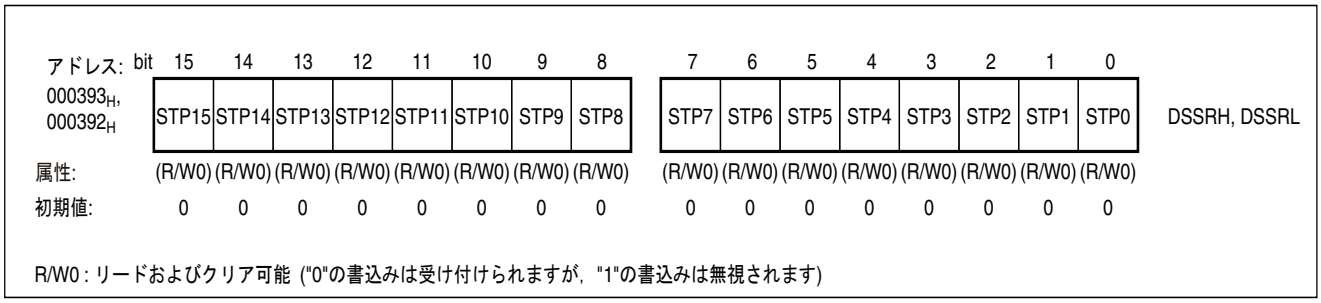
使用できないチャネルの DTE ビットには , "0" を書き込んでください。

4.2.3 DMA停止ステータスレジスタ(DSSR)

DMA停止ステータスレジスタ(DSSR)について説明します。

4.2.3.1 DMA停止ステータスレジスタ(DSSR)

Figure 4-5. DMA停止ステータスレジスタ(DSSR)の構成



[bit15 ～ bit0] STPx: DMA 停止状態 : 停止要求の発生

STPxビット	機能
0 [初期値]	チャネルx でDMA 転送が実行中にSTOP 要求は、発生しなかった。またはDMACS:SE ビットでSTOP 要求が許可されていません。
1	DMA転送中に周辺リソースから出されたSTOP要求によってDMA転送が停止したことを示します。

＜注意事項＞

読出し時には , DSSR レジスタの有効なチャネルの , すべての STP ビットの状態が返されます。

読出し動作時 , 使用できないチャネルの STP ビットは不定値を返します。

リードモディファイライト命令では , 読出し時にすべての STP ビットで "1" が返されます。

このビットをセットできるのはハードウェアだけです。このビットに "1" を書き込んでも無視されます。

このビットに "0" を書き込むとクリアされます。

使用できないチャネルの STP ビットには , "0" を書き込んでください。

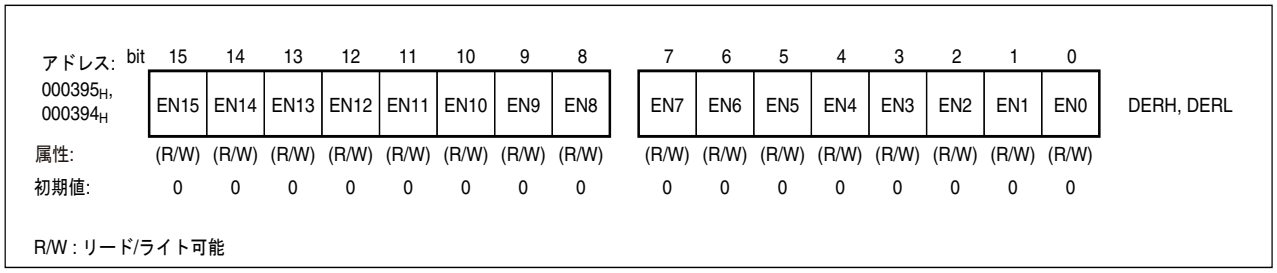
DMA チャネルはすべて STOP 要求をサポートしていますが , すべての IRQ 発生元 (周辺リソース) にこの機能があるというわけではありません。

4.2.4 DMA許可レジスタ(DER)

DMA許可レジスタ(DER)について説明します。

4.2.4.1 DMA許可レジスタ(DER)

Figure 4-6. DMA許可レジスタ(DER)の構成



[bit15 ～ bit0] ENx: DMA 許可

ENxビット	機能
0 [初期値]	リソースからの割込み要求はDMA開始要求としては処理されず, 割込みコントローラに渡されます。
1	リソースからの割込み要求はDMA開始要求として処理されます。その結果, 割込みはDMAコントローラで処理されることになります。DMA転送終了時(DCTが0に達した場合)または周辺からの停止要求時, 割込みコントローラに割込み要求が出力されます。

＜注意事項＞

レジスタは読み書き可能です。

読出しおよびリードモディファイライトの読出し時 , 使用できないチャネルの EN ビットは不定値を返します。

使用できないチャネルの EN には , "0" を書き込んでください。

これらのビットは , スタンバイモードに入る前にすべて "0" にクリアする必要があります。クリアされていない場合, DMAチャネルに関連付けられている割込みをウェイクアップに使用できなくなります。

EN ビットがセットされている場合 , DMA チャネルの設定は変更しないでください。

4.3 DMAディスクリプタ

各DMAチャネルのDMAディスクリプタは, 8バイトで構成されます。加えて,各DMAチャネルは 1 バイトの拡張設定を提供しています。DMAディスクリプタと拡張設定の追加はDMA転送の設定に使用します。

DMAディスクリプタと拡張設定の構成

各 DMA チャネルには 8 バイトの DMA ディスクリプタと拡張設定レジスタ IOABK があります。DMA ディスクリプタは , "000100_H" ～ "$00017F_H$" 番地に位置しています。拡張設定レジスタは , "$000A00_H$" ～ "$000A0F_H$" 番地に位置しています。下の表に , DMA チャネルとディスクリプタと拡張設定のアドレスの関係を示します。

使用できないチャネルのディスクリプタと拡張設定領域にはアクセスできません。使用できないチャネルのディスクリプタ領域には , "0" を書き込んでください。読出しアクセスを行うと不定値が返されます。

DMAチャネルの設定を変更する前に, 関連するIRQがアクティブでないことを確認する必要があります。または , 更新するチャネルのENビットをクリアする必要があります。

Table 4-1. DMAチャネル番号とDMAディスクリプタと拡張設定のアドレスの関係

DMAチャネル	ディスクリプタ開始アドレス	ディスクリプタ終了アドレス	IOABK
0	000100_H	000107_H	$000A00_H$
1	000108_H	$00010F_H$	$000A01_H$
2	000110_H	000117_H	$000A02_H$
3	000118_H	$00011F_H$	$000A03_H$
4	000120_H	000127_H	$000A04_H$
5	000128_H	$00012F_H$	$000A05_H$
6	000130_H	000137_H	$000A06_H$
7	000138_H	$00013F_H$	$000A07_H$
8	000140_H	000147_H	$000A08_H$
9	000148_H	$00014F_H$	$000A09_H$
10	000150_H	000157_H	$000A0A_H$
11	000158_H	$00015F_H$	$000A0B_H$
12	000160_H	000167_H	$000A0C_H$
13	000168_H	$00016F_H$	$000A0D_H$
14	000170_H	000177_H	$000A0E_H$
15	000178_H	$00017F_H$	$000A0F_H$

各チャネルの構造をFigure 4-7に示します。

Figure 4-7. DMAディスクリプタと拡張設定



拡張設定

アドレス:		属性	初期値:
$000A00_H$ + ch	I/O レジスタアドレスポインタのバンク選択(IOABK)	(R/W)	0

DMA ディスクプリタ

アドレス:		属性	初期値:
000107_H + 8 × ch	データカウンタの上位8ビット(DCTH)	(R/W)	X
000106_H + 8 × ch	データカウンタの下位8ビット(DCTL)	(R/W)	X
000105_H + 8 × ch	I/Oレジスタアドレスポインタの上位8ビット(IOAH)	(R/W)	X
000104_H + 8 × ch	I/Oレジスタアドレスポインタの下位8ビット(IOAL)	(R/W)	X
000103_H + 8 × ch	DMA制御レジスタ(DMACS)	(R/W)	X
000102_H + 8 × ch	バッファアドレスポインタの上位8ビット(BAPH)	(R/W)	X
000101_H + 8 × ch	バッファアドレスポインタの中位8ビット(BAPM)	(R/W)	X
000100_H + 8 × ch	バッファアドレスポインタの下位8ビット(BAPL)	(R/W)	X

R/W : リード/ライト可能

"ch" および "8×ch" はチャネルインデックス "ch" によってディスクリプタと拡張設定のアドレスオフセットを指定する用語です。

DMAディスクリプタの各レジスタ

DMA ディスクリプタと拡張設定に含まれる各レジスタについては以降の節で説明します。DMA ディスクリプタレジスタは初期値が未定義のため , 初期化してから DER:ENx を "1" に設定する必要があります。拡張設定レジスタは , リセット時に初期化されます。

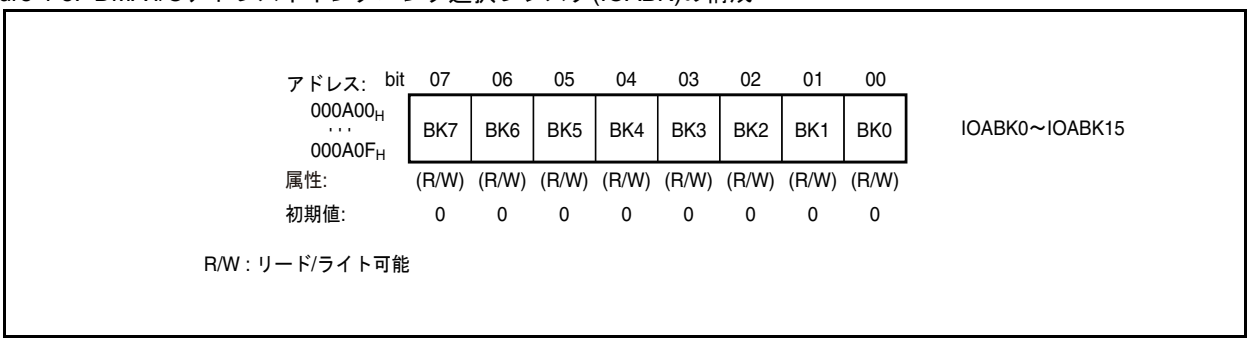
4.3.1 DMA I/Oアドレスポインタバンク選択レジスタ(IOABK)[1]

各DMAチャネルには, 1つのI/Oアドレスポインタバンク選択レジスタ(IOABK0～IOABK15)があります。
このレジスタは, DMA転送の間, どのバンクをI/O空間のアドレスポインタとして使用するかを定義します。

4.3.1.1 DMA I/Oアドレスポインタバンク選択レジスタ(IOABK)

Figure 4-8にDMA I/O アドレスポインタバンク選択レジスタ(IOABK)の構成を示します。

Figure 4-8. DMA I/Oアドレスポインタバンク選択レジスタ(IOABK)の構成



[bit7 ～ bit0] BK: バンク選択

各 DMA チャネルに IOABKx レジスタが存在しており , チャネルで DMA 転送をする間 IO レジスタのアドレッシング用バンクのアドレスを定義します。

＜注意事項＞

IOABK レジスタは読み書き可能です。

IOABK レジスタはすべてのリセットで "00_H" に初期化されます。

読出し動作時 , 使用できないチャネルの IOABK レジスタは不定値を返します。

使用できないチャネルの IOABK レジスタは "0" を書き込んでください。

DMA 転送の間 , IOABK レジスタは変化しません。多重バンク間の転送はできません。

*1: IOA バンクレジスタはすべてのデバイスに利用可能ではありません。データシートを参照してください。

1. IOA bank registers are not available for all devices. Please refer to the Datasheet.

4.3.2 データカウントレジスタ(DCT)

データカウントレジスタ(DCT) について説明します。

4.3.2.1 データカウントレジスタ(DCT)

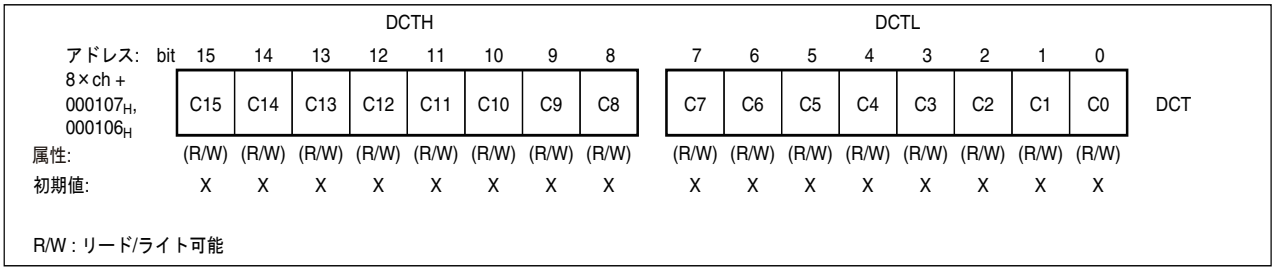
データカウントレジスタ (DCT) は , 転送するバイト数を格納する 16 ビットのレジスタです。1 回のデータ転送が終了するごとに, データカウントレジスタの値はバイト転送の場合は1, ワード転送の場合は2だけデクリメントします。

DCT に "0" を設定すると , 最大データ転送数 (65536 バイト) に設定されます。

データカウントレジスタが "0" になると , DMA 転送は終了です。

Figure 4-9に, データカウントレジスタ(DCT)の構成を示します。

Figure 4-9. データカウントレジスタ(DCT)の構成



4.3.2.2 DCTの転送タイプとデクリメント

DCT の転送タイプとデクリメント値は主に DMACS:BW ビットによって決まります。また , IOA, BAP, および DCT 自身の構成と状態も考慮されます。

Table 4-2に,IOA,BAP,DCTおよびBWの設定(状態)によって行われる転送を示します。BAPのインクリメント/デクリメント動作はDMACS:BPDビットによって決まります。

Table 4-2. ディスクリプタの設定によって実行される転送タイプ

設定/状態				転送タイプ		転送後の更新			転送サイクル[a]
BW	DCT	IOA	BAP	I/O	バッファ	DCT	DMACS:IF = 1のときのIOA	DMACS:BF = 1のときのBAP	
0 (バイト)	>0	-	-	バイト	バイト	-1	+1	+/-1	2
1 (ワード)	>1	偶数	偶数	ワード	ワード	-2	+2	+/-2	2
1 (ワード)	>1	偶数	奇数	ワード	2 x バイト	-2	+2	+/-2	3
1 (ワード)	>1	奇数	偶数	2 x バイト	ワード	-2	+2	+/-2	3
1 (ワード)	>1	奇数	奇数	2 x バイト	2 x バイト	-2	+2	+/-2	4
1 (ワード)	=1	-	-	バイト	バイト	-1	+1	+/-1	2

a. 1転送サイクルは, 1バスアクセスと同様です。

4.3.3 I/Oレジスタアドレスポインタ(IOA)

I/Oレジスタアドレスポインタ(IOA)について説明します。

4.3.3.1 I/Oレジスタアドレスポインタ(IOA)

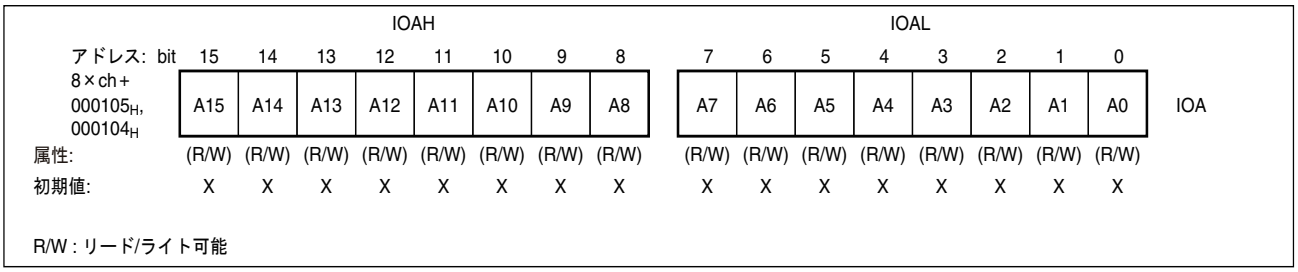
I/O レジスタアドレスポインタ (IOA) は , I/O レジスタの下位アドレス (A15 ～ A0) を示す 16 ビットのレジスタです。I/O レジスタの上位アドレス (A23 ～ A16) は IOABK レジスタに設定できます。

IOABK のリセット値は , バンク 0 になります。

DMA 制御レジスタの DMACS:IF ビットで「更新を実行」を指定した場合 , IOA はバイト転送で 1, ワード転送で 2 だけインクリメントします。DMACS:IF ビットで「更新を実行しない」を指定した場合 , IOA は固定になります (IOA は変化しません)。

Figure 4-10に, I/Oレジスタアドレスポインタ(IOA)の構成を示します。

Figure 4-10. I/Oレジスタアドレスポインタ(IOA)の構成



4.3.4 DMA制御レジスタ(DMACS)

DMA制御レジスタ(DMACS)について説明します。

4.3.4.1 DMA制御レジスタ(DMACS)

DMA 制御レジスタ (DMACS) は 8 ビット長のレジスタで , 以下の用途に使用されます。

- バッファアドレスポインタをインクリメントするか, デクリメントするかを指定する(BPD)
- バッファアドレスポインタおよびI/Oレジスタアドレスポインタ(BF, IF)を更新するか, 固定にするかを指定する
- 転送データフォーマットをバイトまたはワードに指定する(BW)
- 転送方向を指定する(DIR)
- STOP要求を受け付けるかを指定する(SE)

Figure 4-11に, DMACSの構成を示します。

Figure 4-11. DMACSの構成

アドレス: 8×ch + 000103_H	bit	15	14	13	12	11	10	9	8
		–	–	BPD	IF	BW	BF	DIR	SE
属性:		(–)	(–)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
初期値:		X	X	X	X	X	X	X	X

R/W : リード/ライト可能
– : アクセス不可。読出しの返り値は未定義値となります。
このビットには必ず"0"を書き込んでください。

[bit13] BPD:バッファポインタデクリメントビット

BPDビット	機能
0	BF=0の場合, 転送が終了するごとにBAPの値がインクリメントします。
1	BF=0 の場合 , 転送が終了するごとに BAP の値がデクリメントします。 BPDとBWを共に"1" (逆順のワード転送)に設定する場合, IOA, BAPおよびDCTに偶数の値を指定してください。

[bit12] IF: IOAを更新するか, 固定にするかを選択

IFビット	機能
0	1回のデータ転送ごとにIOAの値がインクリメントします。
1	IOAは更新されません(固定I/Oアドレス)。

[bit11] BW:転送データのデータ長を指定

BWビット	機能
0	バイト転送が実行されます。
1	ワード転送が実行されます。

[bit10] BF: BAPを更新するか, 固定にするかを選択

BFビット	機能
0	1回のデータ転送ごとにBAPの値が更新されます。
1	BAPは更新されません(固定バッファアドレス)。

[bit9] DIR:データ転送の方向

DIRビット	機能
0	IOABKおよびIOAで指定されるアドレスからBAPで指定されるアドレスへ転送します(@IOABK.IOA -> @BAP)。
1	BAPで指定されるアドレスからIOABKおよびIOAで指定されるアドレスへ転送します(@BAP -> @IOABK.IOA)。

[bit8] SE: DMA STOP要求の許可

SEビット	機能
0	リソースからのDMA STOP要求に対して停止しません。
1	DMA転送は周辺機能(UART-RX)によって停止させることができます。

4.3.5 バッファアドレスポインタ(BAP)

バッファアドレスポインタ(BAP)について説明します。

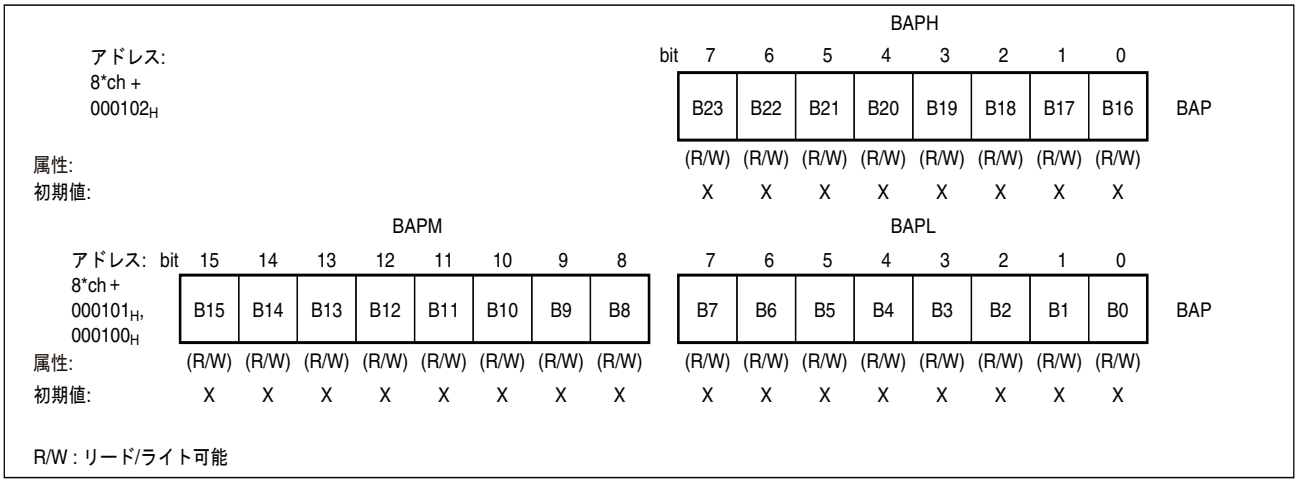
4.3.5.1 バッファアドレスポインタ(BAP)

バッファアドレスポインタ (BAP) は , DMA 転送に使用するアドレスを格納する 24 ビットレジスタです。任意のアドレスを指定できます。

DMACS:BF ビットで「更新を実行」を指定した場合 , BAP の下位 16 ビット (BAPM, BAPL) はバイト転送では "1", ワード転送では "2" だけインクリメントします。上位 8 ビットは変わりません。DMACS:BPD がセットされている場合 , BAP はインクリメントではなく , "1" または "2" だけデクリメントします。DMACS:BF ビットで「更新を実行しない」を指定した場合 , BAP は固定になります (BAP は変化しません)。

Figure 4-12に, バッファアドレスポインタ(BAP)の構成を示します。

Figure 4-12. バッファアドレスポインタ(BAP)



＜注意事項＞

I/O アドレスバンクレジスタ (IOABK) および I/O レジスタアドレスポインタ (IOA) で指定できる領域は , "000000_H" ～ "$FFFFFF_H$" です。IOABK はリセット後 , バンク "00_H" になります。

バッファアドレスポインタ (BAP) で指定できる領域は , "000000_H" ～ "$FFFFFF_H$" です。

DMA コントローラの内部レジスタ (DISEL0 ～ DISEL15, DSR, DSSR, DER) のアドレス , DMA ディスクリプタのアドレス "000100_H" ～ "$00017F_H$" および DMA IOABK レジスタのアドレス "$000A00_H$" ～ "$000A0F_H$" は , IOA と BAP には指定できません。

汎用レジスタ領域のアドレス "000180_H" ～ "$00037F_H$" は IOA と BAP には指定できません。

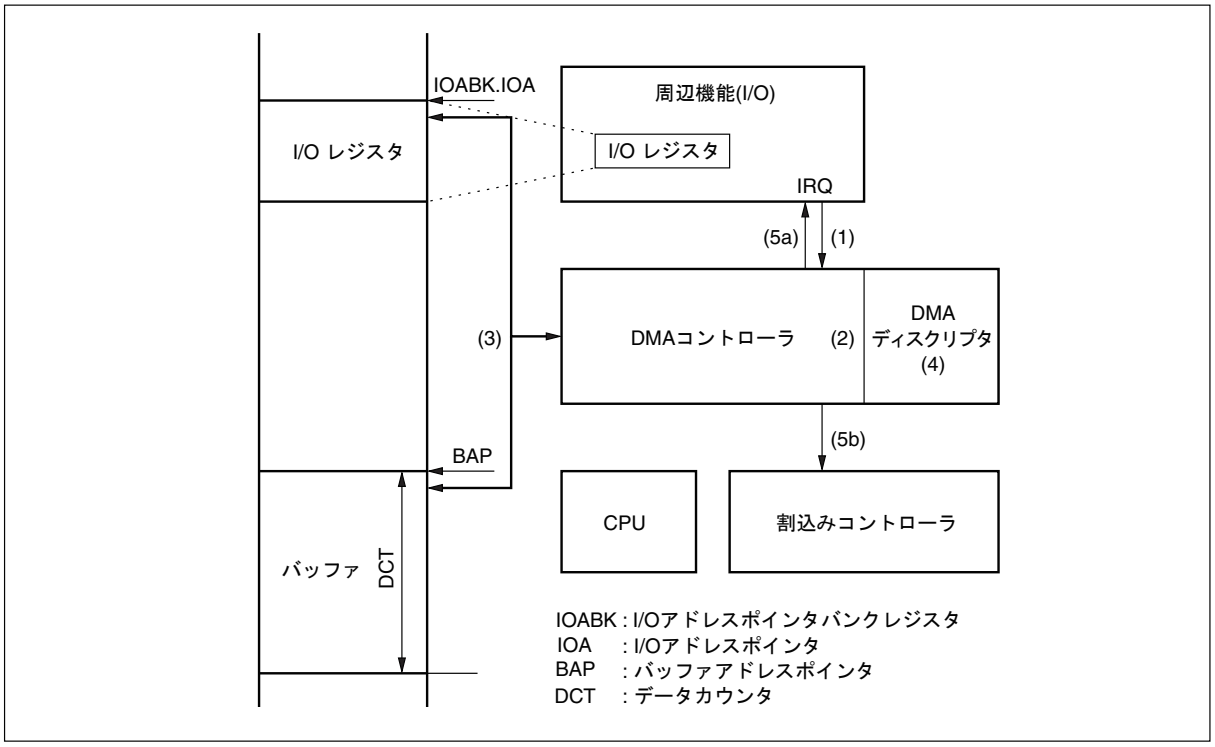
4.4 DMAコントローラの動作

DMAコントローラの動作について説明します。

DMAコントローラの動作

Figure 4-13に, DMAコントローラの動作を示します。

Figure 4-13. DMACの動作



DMA を使用したデータ転送は以下の順序で実行されます。

(1) 周辺リソース (I/O) が割込みを発生させて DMA 転送を要求します。

(2) DMA 許可レジスタ (DER:EN) の対応ビットが "1" で割込み番号が DISEL レジスタの設定と一致する場合 , DMACはディスクリプタから転送元アドレス, 転送先アドレス, カウントおよびチャネル設定を参照します。

(3) I/O とメモリ間で DMA データ転送が開始されます。

(4) DMA ディスクリプタが更新されます。

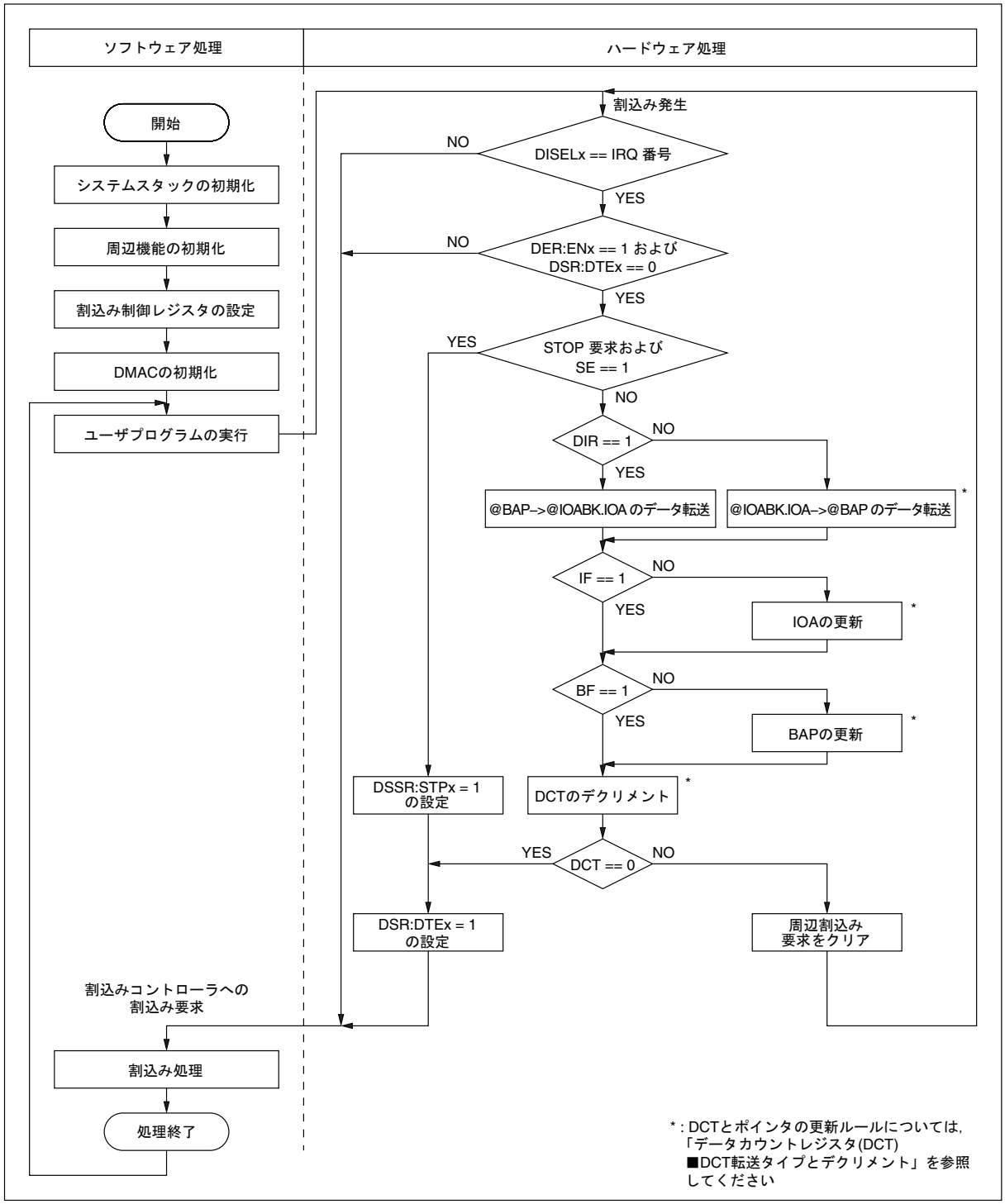
(5) データ (バイトデータまたはワードデータ) を 1 つ転送した後

(a) 転送が完了しなかった (DCT が 0 に達しなかった) 場合 :
DMAC は周辺リソースの DMA 転送要求 (割込み) をクリアします。

(b) 転送終了 (DCT が 0 に達した) 時 :
DMA 転送の完了後 , 転送の完了を示すフラグが DMA ステータスレジスタ (DSR:DTE ビット) にセットされ , 割込み要求が割込みコントローラに出力されます。

DMACを使用した手順

Figure 4-14. DMACを使用した手順



データ転送のサイクル数

転送サイクル数 (DMA 転送中のバスサイクル) は , DCT が 0 に達するまでの転送回数をすべて合計した数です。1 回の DMA 転送のサイクル数については , Table 4-2 の「転送サイクル」列を参照してください。

4.5 DMA転送の例

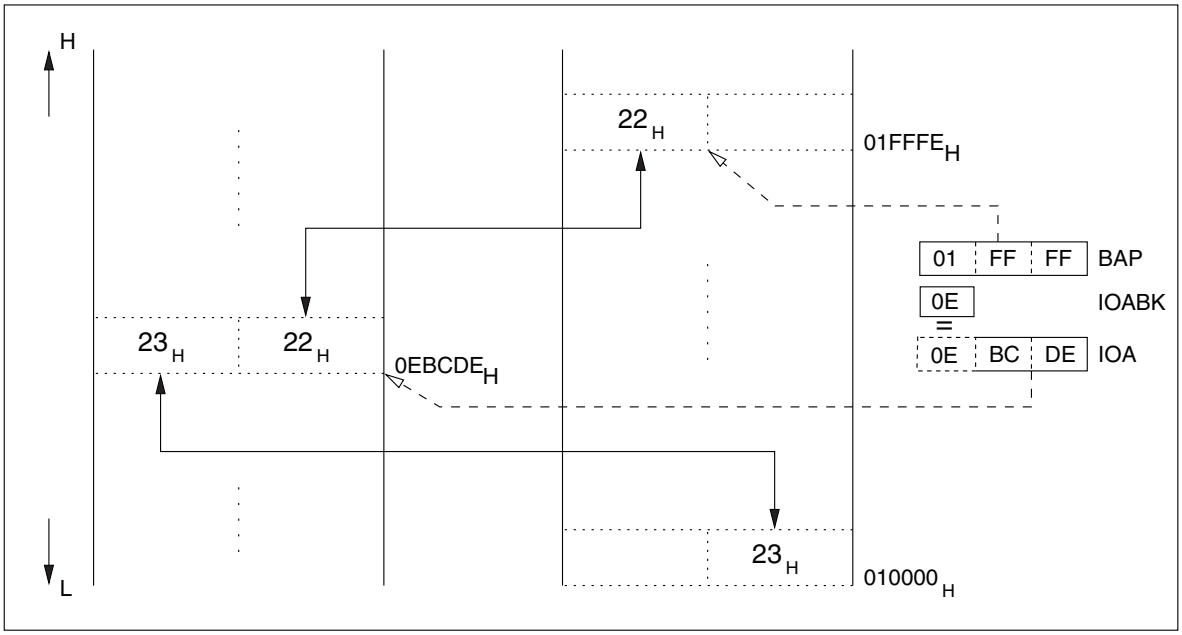
あまり一般的でないDMA転送の種類や状況をいくつか説明します。

単純化のため , 以下の例では IOABK で選択されたメモリバンクの偶数アドレスを IOA として使用します。以下の例は , IOA と BAP のアドレスを交換しても維持されます。ただし , BAP がデクリメントする場合は唯一の例外となります。

バンク境界でのラップアラウンド

DMA 転送は , どのような状況であってもバンク境界をまたいで行われることはありません。IOA または BAP が修正される場合の更新値にも同じことがいえます。バンクの先頭アドレスからワードを読み出す必要がある場合 , 下位バイトはバンクの先頭アドレスから , 上位バイトはバンク内のアドレス 0 から読み出されます。先頭アドレスへの書込みも同様に処理されます (Figure 4-15 を参照してください)。

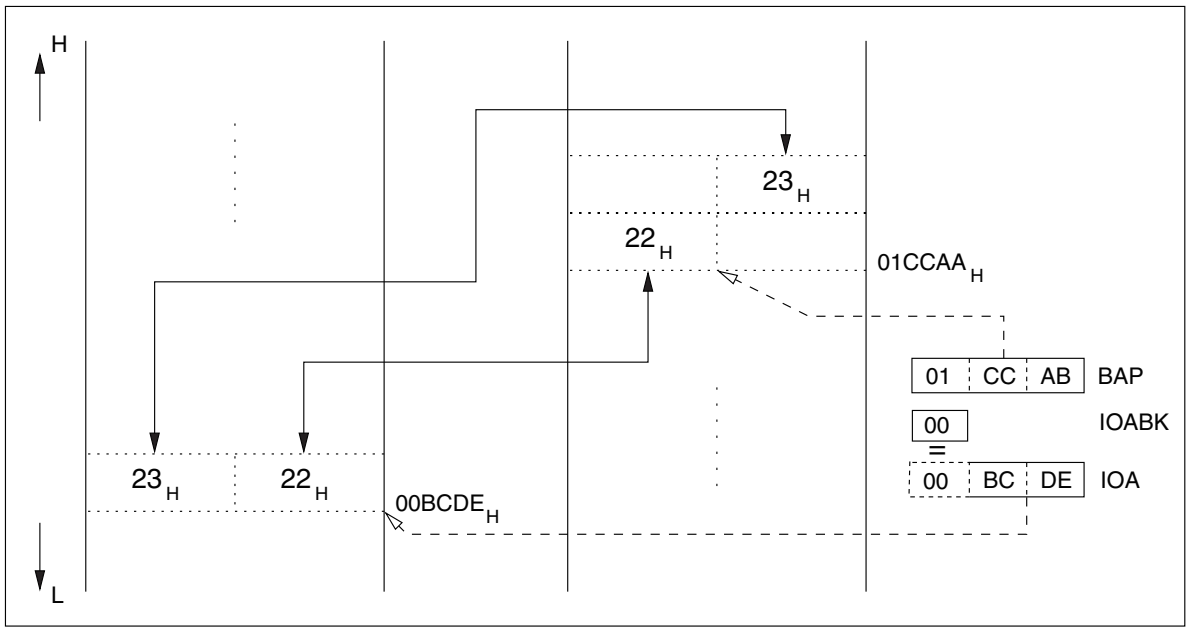
Figure 4-15. バンク01_Hの先頭アドレスを転送元 / 転送先とした1ワード転送



奇数アドレスへのワード転送

奇数アドレスを含むワード転送は , Table 4-2 で説明されているようにバイト転送に分割されます。このような転送の結果を , Figure 4-16 に示します。

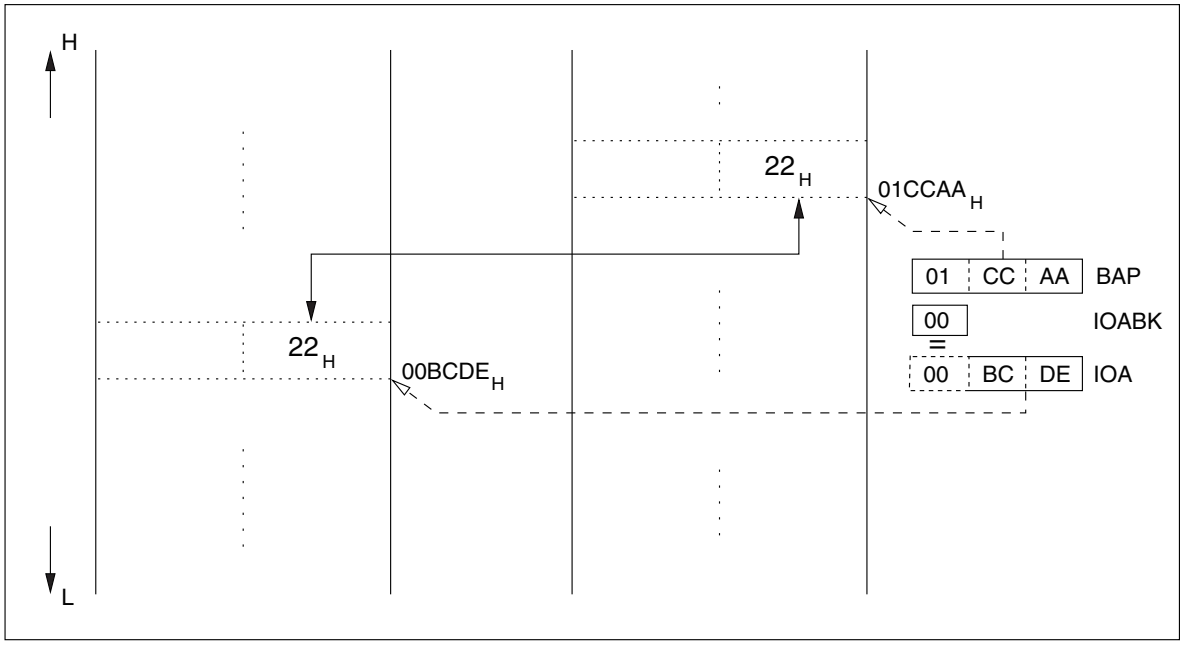
Figure 4-16. 奇数アドレスを転送元 / 転送先とする1ワードの転送



奇数バイト数のワード転送

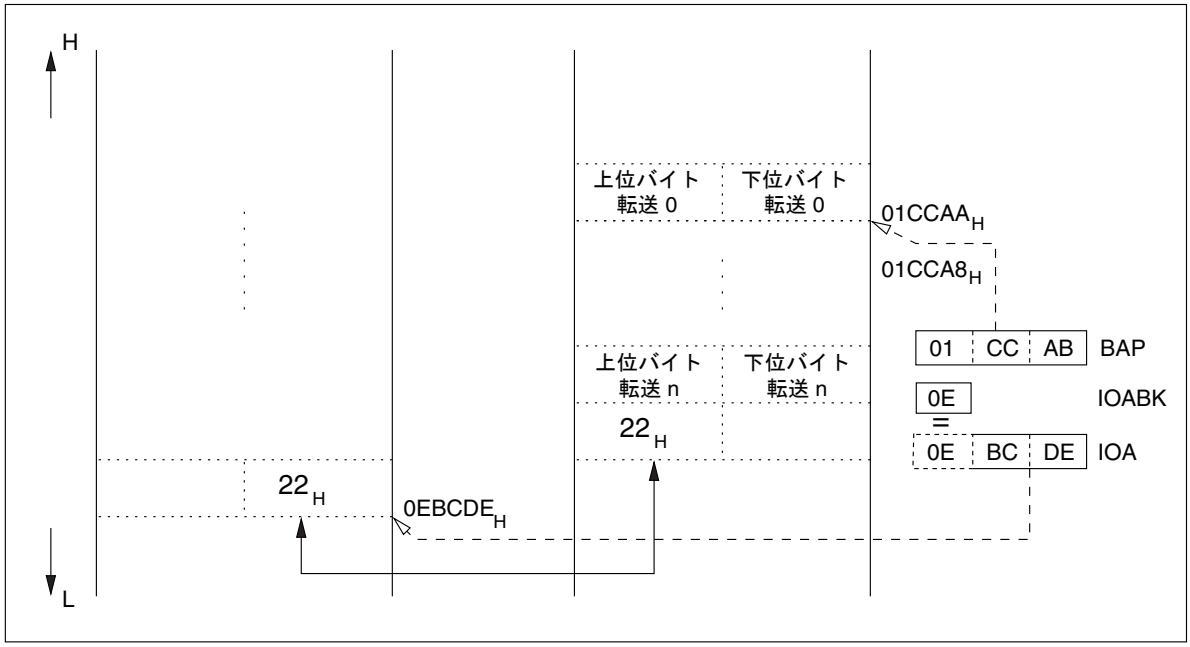
DCT を奇数バイト数に設定しているときにワード転送を使用する場合, 最後の転送で移動するのは 1 バイトだけになります。デクリメントしない BAP の場合 (つまり , 固定の場合またはインクリメントする場合), 転送元の下位バイトは転送先の下位バイトに移動します (Figure 4-17 を参照してください)。

Figure 4-17. ワード転送用に設定されたチャネルの最終バイト転送



BAP がデクリメントする場合 , BAP が参照するワードの上位バイトに対して転送バイトの読み書きが行われます (Figure 4-18 を参照してください)。

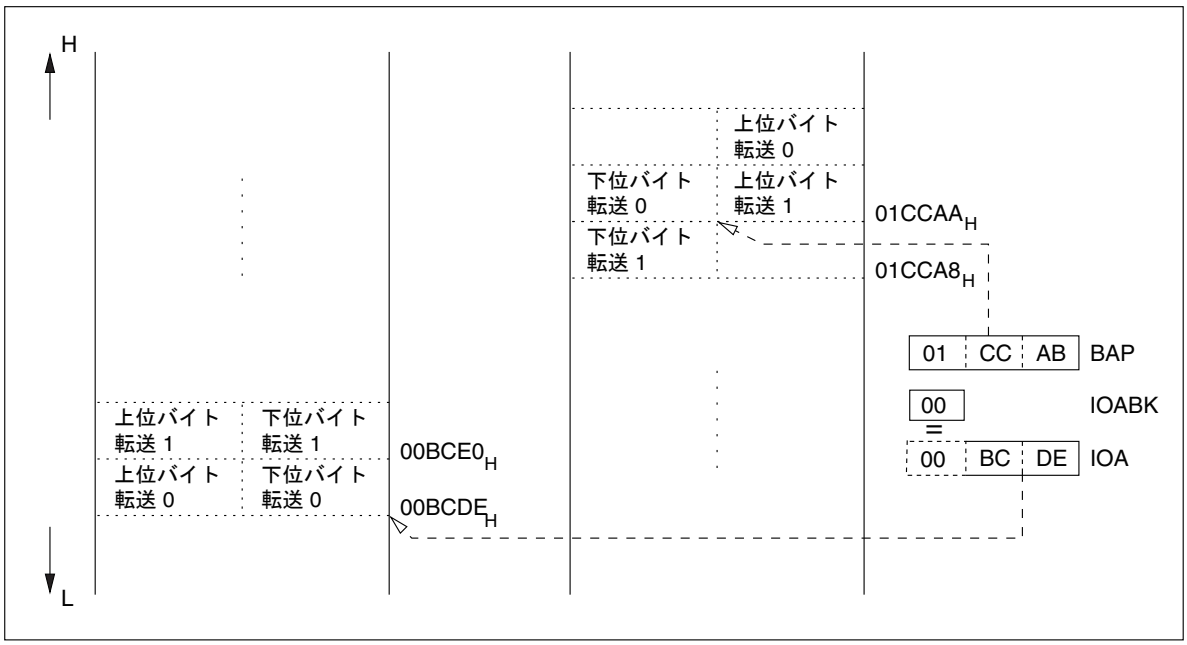
Figure 4-18. ワード転送とBAPデクリメント用に設定されたチャネルの最終バイト転送



IOAをインクリメント, BAPをデクリメントするワード転送

IOA がインクリメント , BAP がデクリメントするように設定されている場合 , 転送されるワードの順序は逆になりますが , ワード内部のバイトの順序は変わりません。Figure 4-19 は , 2 つの転送が完了した後のメモリ内容の例です。

Figure 4-19. IOAをインクリメント, BAPをデクリメントするワード転送



5. 遅延割込み

遅延割込みの機能と動作について説明します。

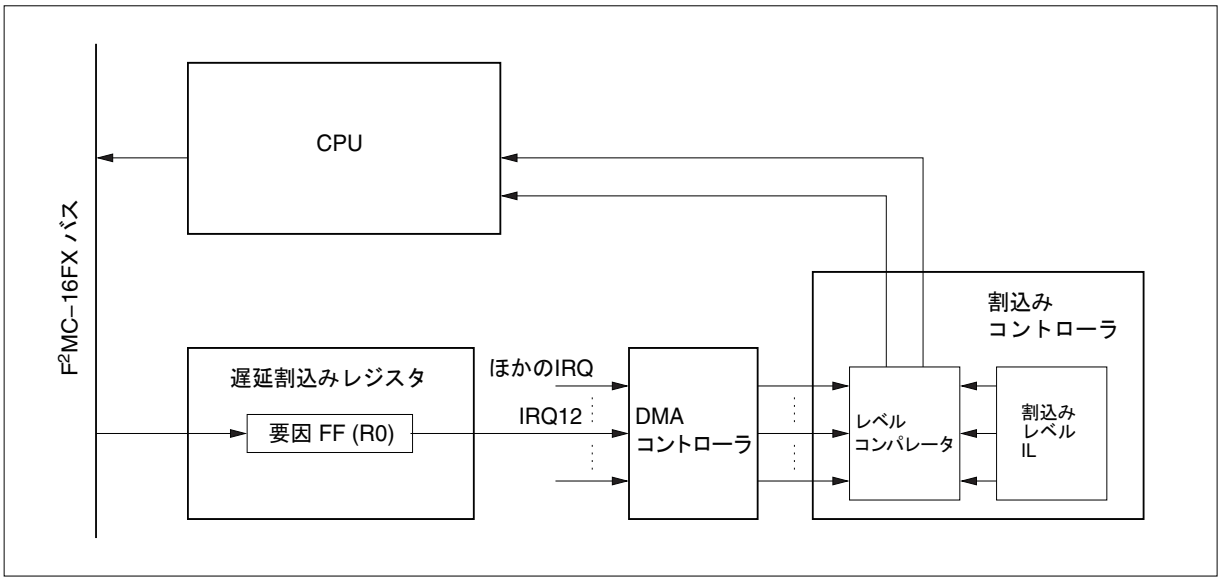
5.1 遅延割込みモジュールの概要

遅延割込み発生モジュールは , タスク切換え用の割込みを発生するためのモジュールです。このモジュールを使えば , F^{2}MC-16FX CPU に対するハードウェア割込み要求の発生または解除をソフトウェアから行うことができます。

遅延割込みのブロックダイヤグラム

Figure 5-1に, 遅延割込み発生モジュールのブロックダイヤグラムを示します。

Figure 5-1. 遅延割込みのブロックダイヤグラム



動作上の注意

遅延割込み信号は DIRR の R0:bit0 に "1" を書き込むことにより発生し , "0" を書き込むことにより解除されます。そのため , DIRR の割込みビットは割込み処理ルーチン内で "0" にクリアしておく必要があります。クリアしない場合 , 最初の割込み処理が完了した後すぐに同じ割込みが発生します。

ICR レジスタに設定されているレベルが同じ場合 , 遅延割込みはハードウェア割込みの中で最も低い優先順位となります。遅延割込みでの優先順位は割込み番号とは関係ありません。遅延割込みは INT #12 の割込みベクタを使用します。

5.2 遅延割込みレジスタ

DIRRは遅延割込み要求の発生や解除を行います。このレジスタに"1"を書き込むと遅延割込み要求が発生し, "0"を書き込むと解除されます。

遅延割込み要求発生 / 解除レジスタ(DIRR: 遅延割込み要求レジスタ)

Figure 5-2. 遅延割込み要求発生 / 解除レジスタ(DIRR)

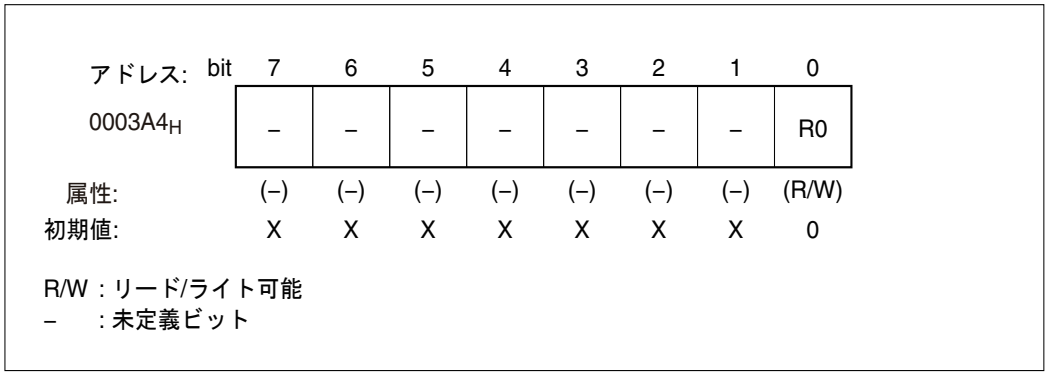


Table 5-1. 遅延割込み要因発生 / 解除レジスタ(DIRR)の各ビットの機能説明

ビット名		機能
bit 7 ～ bit 1	-: 未定義ビット	• このビットをリードしたときの値は不定です。 • このビットには必ず"0"を書き込んでください。
bit 0	R0: 遅延割込み要求発生ビット	• このビットで遅延割込み要求の発生または解除を行います。 • "1" に設定した場合 , 遅延割込み要求が発生します。 • "0" に設定した場合 , 遅延割込み要求が解除されます。 • このビットはリセット時に初期化されます。

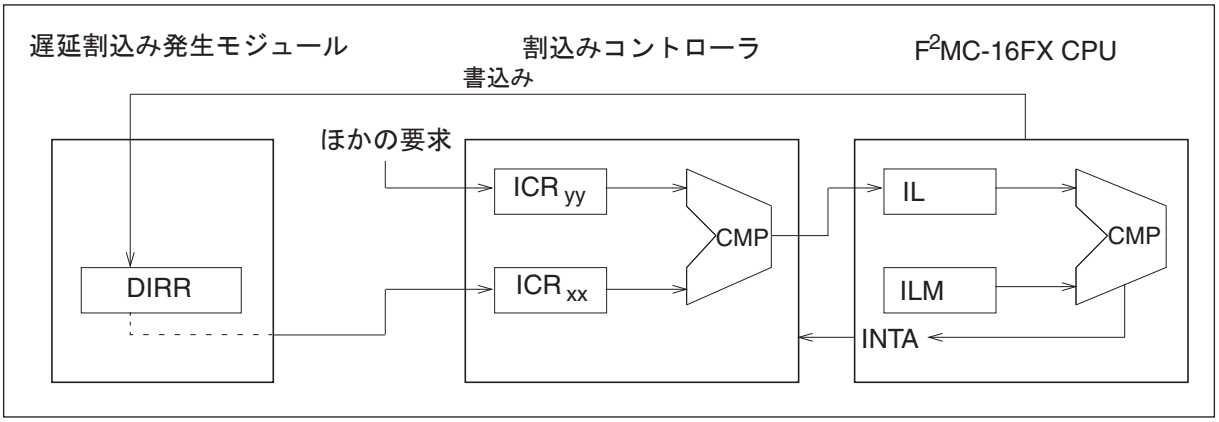
5.3 遅延割込みの動作

CPU がソフトウェアによってDIRRの該当ビット(R0:bit0)に"1"を設定した場合, 遅延割込み発生モジュールの要求ラッチがセットされ, 割込みコントローラに対する割込み要求が発生します。

遅延割込みの発生

CPU がソフトウェアによって DIRR の R0:bit0 に "1" を書き込むと , 遅延割込み発生モジュールの要求ラッチがセットされ , 割込みコントローラに対する割込み要求が発生します。ほかの割込み要求よりも割込み優先順位が高い場合 , 割込みコントローラは F^{2}MC-16FX CPU に対して割込み要求を発生します。F^{2}MC-16FX CPU は内蔵 CCR レジスタの ILM ビットと割込み要求を比較して , そこで割込み要求レベルが ILM ビットのレベルより高い場合は , 現在実行中の命令が完了するとすぐにハードウェア割込み処理マイクロプログラムの実行を開始します。以上のようにして , 当該割込みに対する割込み処理ルーチンが実行されます。

Figure 5-3. 遅延割込みの発生



6. クロック

F^{2}MC-16FX ファミリのマイクロコントローラで使用されるクロックについて説明します。

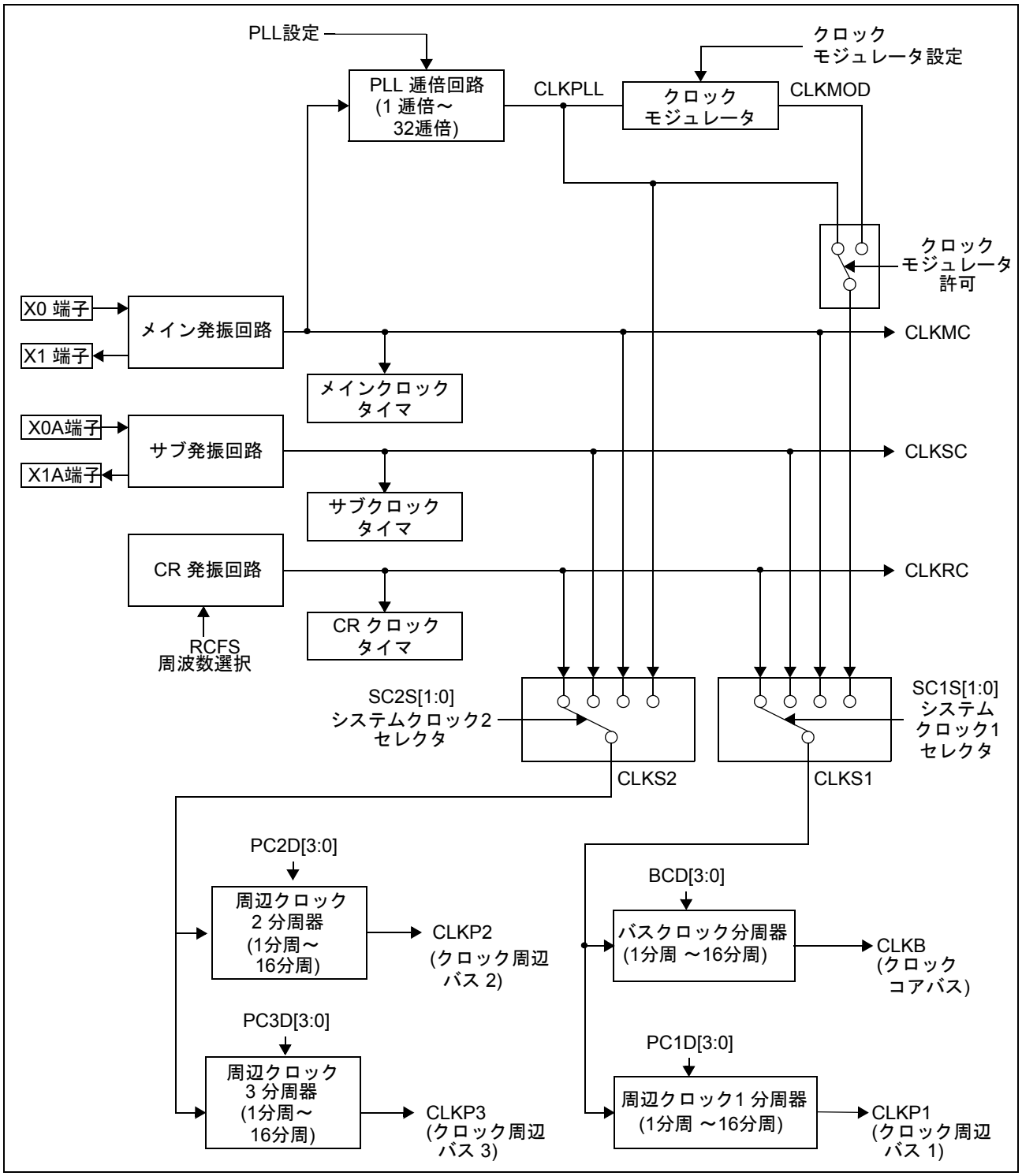
6.1 クロック

F^{2}MC-16FX MCU には , 最大 4 種類のクロックソース (CR クロック , メインクロック , PLL クロック , およびサブクロック) が用意されています。クロック分周器は , バスクロック (CPU とメモリを結ぶ内部バス用) と周辺クロック周波数を個別に設定できます。
サブ発振器はすべての F^{2}MC-16FX MCU に搭載されているわけではありません。本書に記載されているサブ発振器とサブクロックに関する機能は , サブ発振器が未搭載の MCU には使用できません。また , これに対応する設定は効果がありません。

ブロックダイヤグラム

F^{2}MC-16FX MCU におけるクロックソース , 内部クロック生成 , およびソースクロックタイマを Figure 6-1 のブロックダイヤグラムに示します。ブロックダイヤグラムの下に各種クロックの説明があります。どのモジュールをどのクロックに接続するかを示すブロックダイヤグラムについては , 「第 1 章 概要」を参照してください。

Figure 6-1. クロック生成とソースクロックタイマのブロックダイヤグラム



CLKP3 については , データシートを参照してください。

ソースクロック

以下のクロック信号は F²MC-16FX MCU のクロックソースです。

- **CR クロック (CLKRC)**

CR クロック CLKRC は内部 CR 発振器の出力クロックです。CR 発振器は CKFCR レジスタの RCFS (CR クロック周波数選択) ビットで選択可能な 2 種類のクロック周波数 (公称 2 MHz および 100 kHz, データシートを参照してください) を生成できます。

- **メインクロック (CLKMC)**

メインクロック CLKMC はメイン発振回路の出力クロックです。メイン発振回路には外部発振子または外部クロックのいずれかを接続することができます。

- **PLL クロック (CLKPLL)**

PLL クロック CLKPLL は , メインクロック CLKMC を内部 PLL クロック逓倍回路 (PLL 発振回路) にて逓倍されたクロックになります。逓倍率は , 1 ～ 32 逓倍を利用できます。
このクロックを利用できるのは , メインクロック CLKMC が動作中の場合に限ります。

- **サブクロック (CLKSC)**

サブクロック CLKSC はサブ発振回路の出力クロックです。サブ発振回路には外部発振子または外部クロックのいずれかを接続することができます。
サブクロックはすべての F²MC-16FX MCU に搭載されているわけではありません。

内部生成クロック

以下のクロック信号は , ソースクロックから内部生成されます。

- **変調 PLL クロック (CLKMOD)**

電磁放射量を少なくするために , 内蔵クロックモジュレータで PLL クロック CLKPLL を変調することができます。このクロックモジュレータの出力クロックを CLKMOD とよびます。

- **システムクロック 1 (CLKS1)**

システムクロック 1 (CLKS1) は F²MC-16FX MCU のマスタクロックです。バスクロック (CLKB) と周辺クロック 1 (CLKP1) 用のクロック分周器から生成されます。

CKSR レジスタの SC1S1, SC1S0 (システムクロック 1 選択) ビットに応じて , 以下の 4 つのクロックのいずれかをシステムクロック 1 として選択することができます。CLKRC (CR クロック), CLKMC (メインクロック), CLKPLL/CLKMOD (非変調または変調 PLL クロック), または CLKSC (サブクロック)。

- **バスクロック (CLKB)**

バスクロック CLKB は内部バス , CPU, 内部メモリ , DMA コントローラ , および外部バスインタフェース用のクロックソースです。このクロックは , すべてのスタンバイモード (スリープ , タイマ , およびストップモード) で停止します。システムクロック 1 (CLKS1) のバスクロック分周器から生成されます。

- **周辺クロック 1 (CLKP1)**

周辺クロック 1 (CLKP1) は , 変調クロックで動作可能なすべての周辺リソース用のクロックソースです。このクロックは , タイマモードとストップモードで停止します。システムクロック 1 (CLKS1) の周辺クロック分周器 1 から生成されます。

- **システムクロック 2 (CLKS2)**

システムクロック 2 (CLKS2) は , 変調することができないマスタクロックになります。周辺クロック 2 (CLKP2) 専用のクロック分周器を搭載しています。

CKSR レジスタの SC2S1, SC2S0 (システムクロック 2 選択) ビットに応じて , 以下の 4 つのクロックのいずれか

をシステムクロック 2 として選択することができます。CLKRC (CR クロック), CLKMC (メインクロック), CLKPLL (非変調 PLL クロック), または CLKSC (サブクロック)。

これらのビットの設定は , SC1S1, SC1S0 でのみ定義される CPU のクロックモードには影響を与えません。

- **周辺クロック 2 (CLKP2)**

周辺クロック 2 (CLKP2) は , 変調クロックで動作することのない周辺リソース用のクロックソースです。このクロックは , タイマモードとストップモードで停止します。システムクロック 2 (CLKS2) の周辺クロック分周器 2 から生成されます。

- **周辺クロック 3 (CLKP3)**

周辺クロック 3 (CLKP3) はオプションのクロックソースであり , 変調クロックで動作することがなく CLKP2 に接続するものとは別のクロック周波数を必要とする周辺リソース用のクロックソースです。CLKP3 は , タイマモードとストップモードで停止します。システムクロック 2 (CLKS2) の周辺クロック分周器 3 から生成されます。

CLKP3 クロックはすべてのデバイスで使用できるわけではありません。データシートを参照してください。

USB 機能を使用する場合 , CLKP3 は 48MHz に設定する必要があります。

ソースクロックタイマ

メインクロック , CR クロック , およびサブクロックの発振器は , 選択されたシステムクロック , バスクロック , および周辺クロックとは別に動作する専用ソースクロックタイマ (メインクロックタイマ , CR クロックタイマ , およびサブクロックタイマ) を搭載しています。

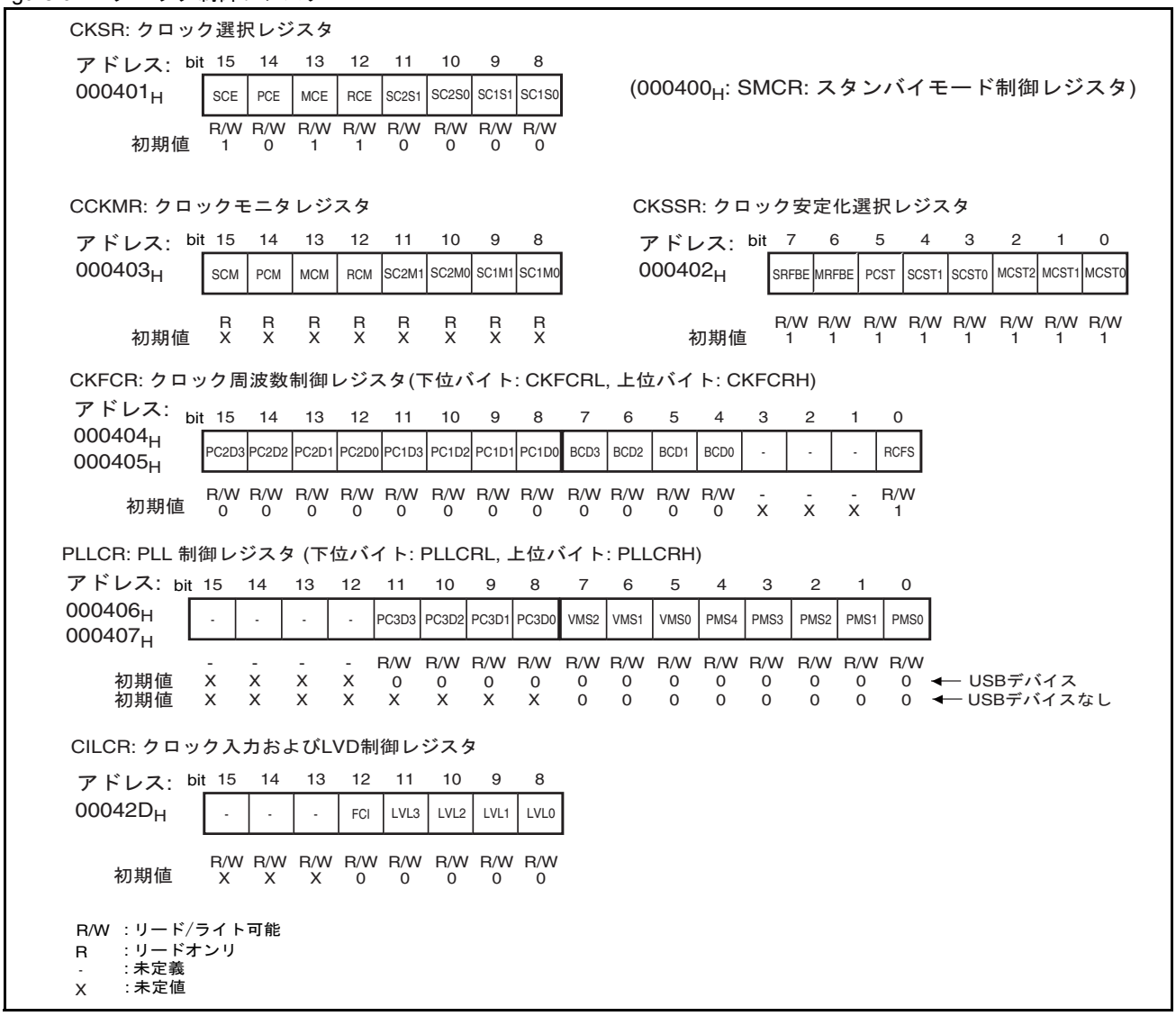
6.2 クロック制御レジスタ

クロック制御レジスタを一覧表示し, 各レジスタの機能について詳しく説明します。

クロック制御レジスタ

すべてのクロック制御レジスタの概要を Figure 6-2 に示します。

Figure 6-2. クロック制御レジスタ



6.2.1 クロック選択レジスタ(CKSR)

クロック選択レジスタ (CKSR) は, システムクロックセレクタ 1 および 2 と発振回路の制御に使用します。

クロック選択レジスタ(CKSR)の構成

クロック選択レジスタ (CKSR) の構成を Figure 6-3 に示し , 各ビットの機能を Table 6-1 で説明します。

Figure 6-3. クロック選択レジスタ(CKSR)の構成

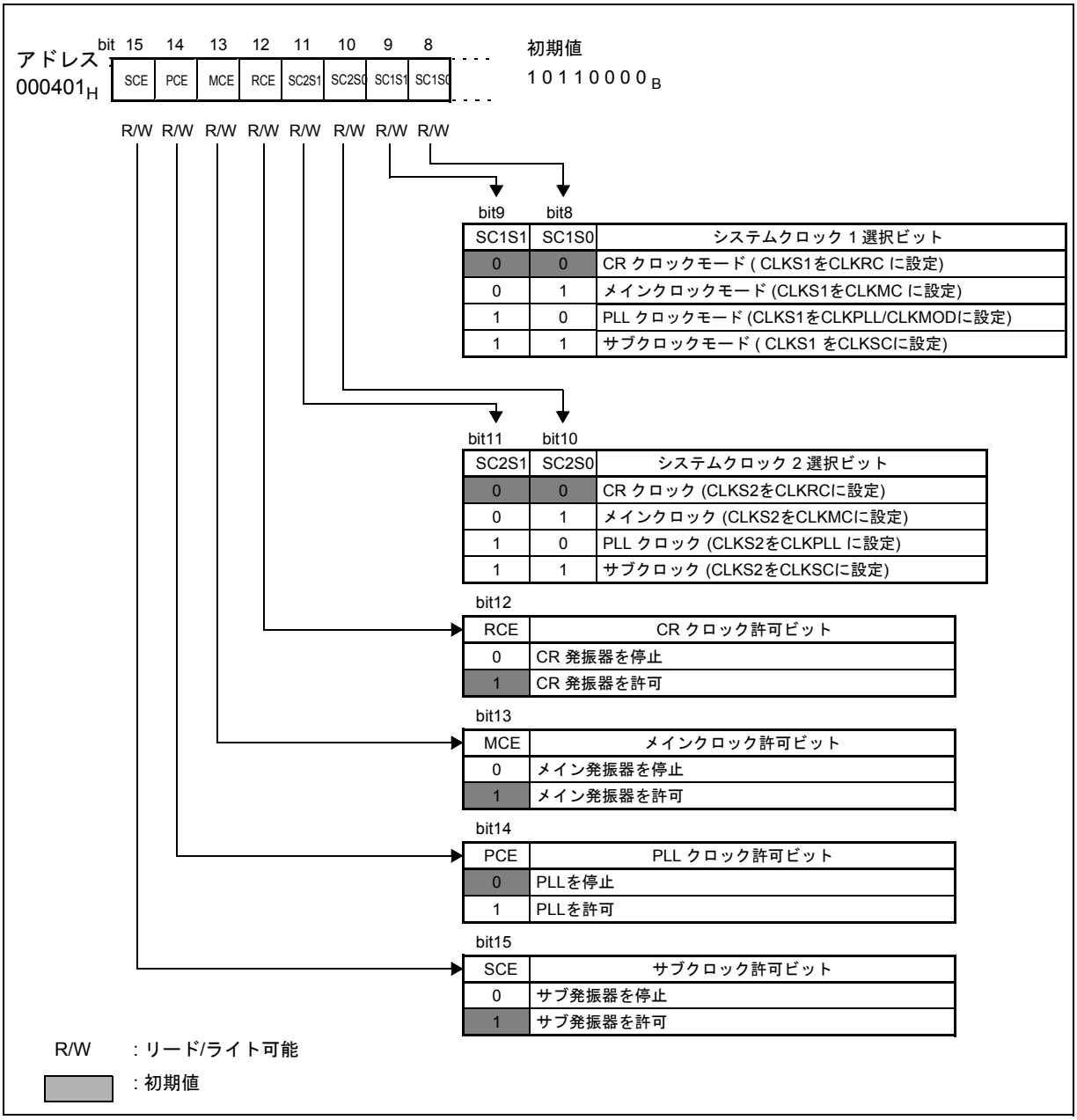


Table 6-1. クロック選択レジスタ(CKSR)の各ビットの機能説明 (Sheet 1 of 2)

<table>
<tr><th colspan="2">ビット名</th><th>機能</th></tr>
<tr><td>bit 8, bit 9</td><td>SC1S0 , SC1S1:
システムクロック 1
選択ビット</td><td>
• これらのビットは , 以下の表に従ってバスクロックと周辺クロック 1 のソース用のシステムクロック 1 セレクタを制御します。
<table>
<tr><th colspan="2">bit9 bit8</th><th>クロックモード</th><th>CLKS1 (システムクロック 1)</th></tr>
<tr><td>SC1S1</td><td>SC1S0</td><td></td><td></td></tr>
<tr><td>0</td><td>0</td><td>CR クロックモード</td><td>CLKRC (CR クロック) に設定</td></tr>
<tr><td>0</td><td>1</td><td>メインクロックモード</td><td>CLKMC (メインクロック) に設定</td></tr>
<tr><td>1</td><td>0</td><td>PLL クロックモード</td><td>CLKPLL または CLKMOD
(PLL クロック) に設定</td></tr>
<tr><td>1</td><td>1</td><td>サブクロックモード</td><td>CLKSC (サブクロック) に設定</td></tr>
</table>
• これらのビットは CPU クロックモードを制御します。
• リセットが発生すると , これらのビットが "00" (CR クロックモード) に初期化されます。
• 異なるクロックモードへの移行は , 選択されたクロックが安定する (クロックモニタレジスタ CKMR の対応するクロック準備完了モニタビットが設定される) まで延期されます。
• 選択されたクロックの準備が整っている場合は , 同期メカニズムを起動するとクロックモードの移行が行われます。クロックモニタレジスタのシステムクロック 1 モニタビット (SC1M[1:0]) は , クロックの切換えが終了したことを示します。
• 事前に選択されたモードへの切換えが終了する前にこのレジスタに別の値を書き込むと , 最初の移行要求が取り消されます。
• 使用禁止になっているクロックに移行することはできません。
(注意事項)
• ストップモードにある場合, システムクロック 1 セレクタは, 対応するクロック準備完了モニタビットに関係なく, SC1S[1:0] ビットで選択されたクロックに直接切り換わります (ストップモードでは, クロックモニタビットがすべてクリアされます)。このため, ストップモードに切り換える前に, 使用禁止になっているクロックを選択しないでください。
</td></tr>
<tr><td>bit 10, bit 11</td><td>SC2S0 , SC2S1:
システムクロック 2
選択ビット</td><td>
• これらのビットは , 以下の表に従って周辺クロック 2 のソース用のシステムクロック 2 セレクタを制御します。
<table>
<tr><th colspan="2">bit11 bit10</th><th>CLKS2 (システムクロック 2)</th></tr>
<tr><td>SC2S1</td><td>SC2S0</td><td></td></tr>
<tr><td>0</td><td>0</td><td>CLKRC (CR クロック) に設定</td></tr>
<tr><td>0</td><td>1</td><td>CLKMC (メインクロック) に設定</td></tr>
<tr><td>1</td><td>0</td><td>CLKPLL (非変調 PLL クロック) に設定</td></tr>
<tr><td>1</td><td>1</td><td>CLKSC (サブクロック) に設定</td></tr>
</table>
• リセットが発生すると , これらのビットが "00" (CR クロック) に初期化されます。
• 異なるクロックソースへの切換えは , 選択されたクロックが安定する (クロックモニタレジスタ CKMR 内の対応するクロック準備完了モニタビットが設定される) まで延期されます。
• 選択されたクロックが安定している場合は , 同期メカニズムを起動するとクロックソースの移行が行われます。クロックモニタレジスタのシステムクロック 2 モニタビット (SC2M[1:0]) は , クロックの切換えが終了したことを示します。
• 事前に選択されたモードへの切換えが終了する前にこのレジスタに別の値を書き込むと , 最初の移行要求が取り消されます。
• 使用禁止になっているクロックには移行することはできません。
(注意事項)
• ストップモードにある場合, システムクロック 2 セレクタは, 対応するクロック準備完了モニタビットに関係なく, SC2S[1:0] ビットで選択されたクロックに直接切り換わります (ストップモードでは, クロックモニタビットがすべてクリアされます)。このため, ストップモードに切り換える前に, 使用禁止になっているクロックを選択しないでください。
</td></tr>
</table>

Table 6-1. クロック選択レジスタ(CKSR)の各ビットの機能説明 (Sheet 2 of 2)

ビット名		機能
bit 12	RCE: CR クロック 許可ビット	• このビットは , 内部 CR 発振器の許可 / 停止を制御します。 • このビットに "1" を書き込むと CR 発振器が許可され , "0" を書き込むと CR 発振器が停止します。 • このビットはリセットごとに "1" (発振器を許可) に初期化されます。 • CR 発振器が許可されると CR 発振安定時間に入ります。この時間は 64 CR クロック周期に固定されています。この時間が経過すると , CKMR レジスタ内の CR クロックモニタビット (RCM) が設定されます。 • CLKRC を CLKS1/CLKS2 用のクロックソースとして選択すると , RCE ビットの設定に関係なく CR 発振器が起動します。CR 発振器を停止できるのは , 2 つのシステムクロックが別々のクロックソースで動作している場合に限ります。 (注意事項) • クロック停止検出リセットが許可されている場合は , CR 発振器を禁止しないでください。この場合に RCE を "0" に設定すると , クロック停止検出リセットが発生します。 • ウォッチドッグ用のクロックソースが CR クロックに設定されている場合は, CR 発振器を禁止しないでください。この場合に RCE を "0" に設定すると, ウォッチドッグリセットが発生します。
bit 13	MCE: メインクロック 許可ビット	• このビットは , メイン発振回路の許可 / 停止を制御します。 • このビットに "1" を書き込むとメイン発振器が許可され , "0" を書き込むとメイン発振器が停止します。 • このビットはリセットごとに "1" (発振器を許可) に初期化されます。 • メイン発振器が許可されるとメイン発振安定時間に入ります。この時間は , CKSSR レジスタの MCST ビットで設定します。この時間が経過すると , CKMR レジスタ内のメインクロックモニタビット (MCM) が設定されます。 • CLKMC, CLKPLL, または CLKMOD を CLKS1/CLKS2 用のクロックソースとして選択すると , MCE ビットの設定に関係なくメイン発振器が起動します。メイン発振器を停止できるのは , 2 つのシステムクロックが別々のクロックソースで動作している場合に限ります。 • このビットを "0" に設定すると PLL クロックが禁止され (PCE の設定に無関係), PCM が "0" に設定されます。ただし , PLL クロックがシステムクロック 1 または 2 として使用されている場合を除きます。 (注意事項) • ウォッチドッグ用のクロックソースがメインクロックに設定されている場合は, メイン発振器を禁止しないでください。この場合に MCE を "0" に設定すると, ウォッチドッグリセットが発生します。
bit 14	PCE: PLL クロック 許可ビット	• このビットは , PLL 発振回路の許可 / 停止を制御します。 • このビットに "1" を書き込むと PLL が許可され , "0" を書き込むと PLL が停止します。 • PLL はメインクロックを入力クロックとして使用しているため , PLL を許可する場合は MCE も "1" に設定する必要があります。MCE =0 の場合 , PCE の設定は効果がありません。 • このビットはリセットごとに "0" (PLL が停止) に初期化されます。 • メインクロックが安定し (MCM =1), PLL が許可されると PLL 安定時間に入ります。この時間は , CKSSR レジスタの PCST ビットで設定します。この時間が経過すると , CKMR レジスタ内の PLL クロックモニタビット (PCM) が設定されます。 • CLKPLLまたはCLKMODをCLKS1/CLKS2用のクロックソースとして選択するとPCEビットの設定に関係なく PLL が起動します。PLL を停止できるのは, 2 つのシステムクロックが別々のクロックソースで動作している場合に限ります。
bit 15	SCE: サブクロック 許可ビット	• このビットは , サブ発振回路の許可 / 停止を制御します。 • このビットに "1" を書き込むとサブ発振器が許可され , "0" を書き込むとサブ発振器が停止します。 • このビットはリセットごとに "1" (発振器を許可) に初期化されます。 • サブ発振器が許可されるとサブ発振安定時間に入ります。この時間は , CKSSR レジスタの SCST ビットで設定します。この時間が経過すると , CKMR レジスタ内のサブクロックモニタビット (SCM) が設定されます。 • CLKSC を CLKS1/CLKS2 用のクロックソースとして選択すると , SCE ビットの設定に関係なくサブ発振器が起動します。サブ発振器を停止できるのは , 2 つのシステムクロックが別々のクロックソースで動作している場合に限ります。 (注意事項) • ウォッチドッグ用のクロックソースがサブクロックに設定されている場合は, サブ発振器を禁止しないでください。この場合に SCE を "0" に設定すると, ウォッチドッグリセットが発生します。

6.2.2 クロックモニタレジスタ(CKMR)

クロックモニタレジスタ (CKMR) は, システムクロック (クロックモード) の現在の状態と発振回路の状態を確認する目的に使用します。

クロックモニタレジスタ(CKMR)の構成

クロックモニタレジスタ (CKMR) の構成を Figure 6-4 に示し , 各ビットの機能を Table 6-2 で説明します。

Figure 6-4. クロックモニタレジスタ(CKMR)の構成

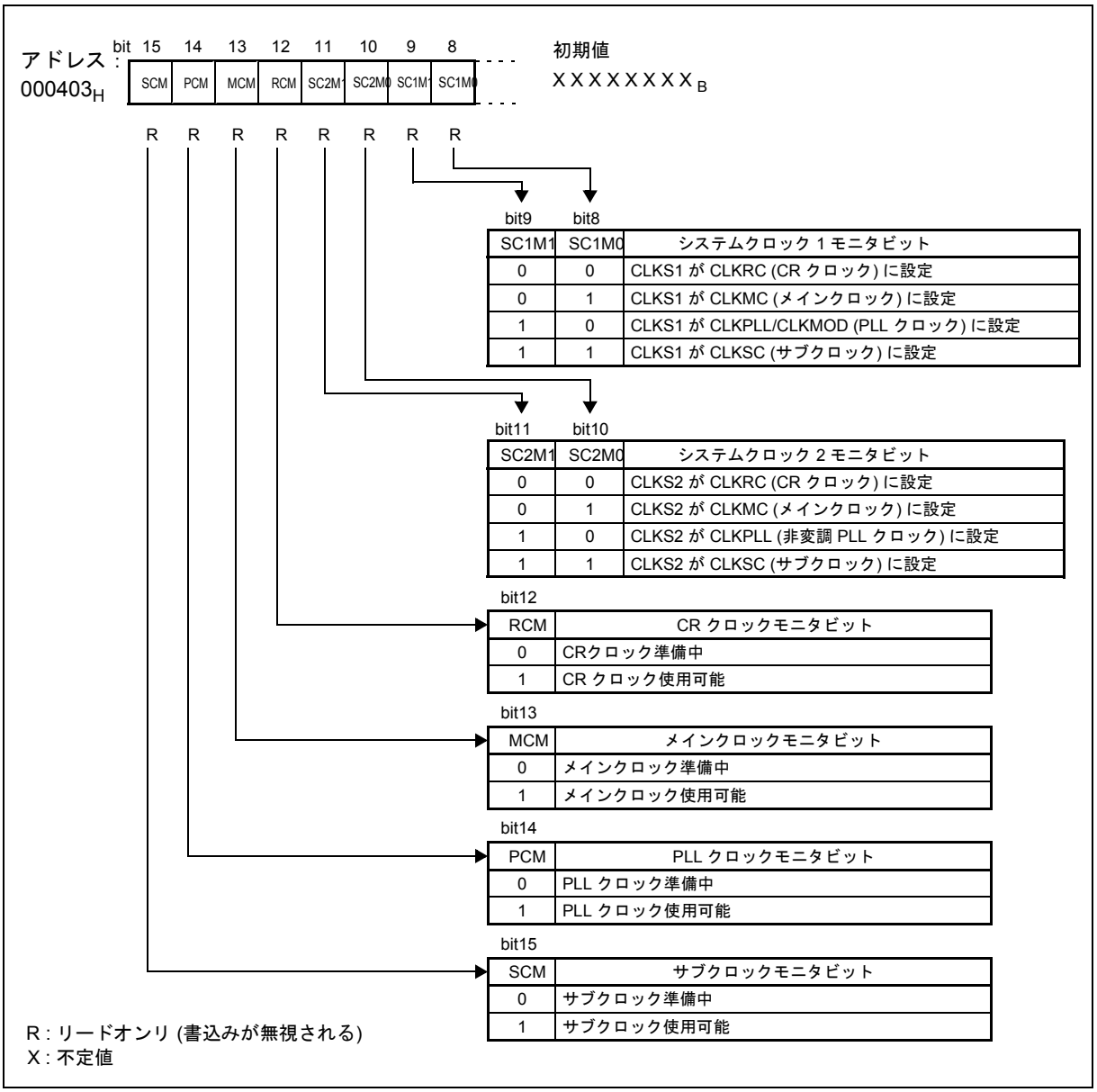


Table 6-2. クロックモニタレジスタ(CKMR)の各ビットの機能説明 (Sheet 1 of 2)

<table>
<tr><th colspan="2">ビット名</th><th>機能</th></tr>
<tr><td>bit 8, bit 9</td><td>SC1M1, SC1M0:
システムクロック
1 モニタビット</td><td>
• これらのビットは , 以下の表に従ってシステムクロック 1 として現在使用しているクロックを示します。
<table>
<tr><th colspan="2">bit9 bit8</th><th></th></tr>
<tr><th>SC1M1</th><th>SC1M0</th><th>システムクロック 1 モニタビット</th></tr>
<tr><td>0</td><td>0</td><td>CLKS1 が CLKRC (CR クロック) に設定</td></tr>
<tr><td>0</td><td>1</td><td>CLKS1 が CLKMC (メインクロック) に設定</td></tr>
<tr><td>1</td><td>0</td><td>CLKS1 が CLKPLL/CLKMOD (PLL クロック) に設定</td></tr>
<tr><td>1</td><td>1</td><td>CLKS1 が CLKSC (サブクロック) に設定</td></tr>
</table>
• これらのビットは現在動作中の CPU クロックモードを示します。
• SC1M[1:0] ビットの読出し値は SC1S[1:0] システムクロック 1 選択ビットに書き込まれた値と異なる場合があります。これは , 要求されたクロックモードの移行がまだ終了していないことを示します。
• クロックモードの移行 (同期メカニズムが起動中) は以下になります。SC1S[1:0] ビットで選択されたクロックの準備が整っていても (クロックモニタビット 1), SC1M[1:0] ビットで示された動作中のクロックモードは SC1S[1:0] の設定と異なります。SC1M[1:0] ビットが更新されるのは , クロックモードの移行が終了した後です。
• 選択されたクロックの準備が整っていない (SC1S[1:0] ビットで選択されたクロックのクロックモニタビットが "0" に設定されている) 場合は, クロックモードを切り換えることができません。
</td></tr>
<tr><td>bit 10, bit 11</td><td>SC2M1, SC2M0:
システムクロック
2 モニタビット</td><td>
• これらのビットは , 以下の表に従ってシステムクロック 2 として現在使用されているクロックを示します。
<table>
<tr><th colspan="2">bit11 bit10</th><th></th></tr>
<tr><th>SC2M1</th><th>SC2M0</th><th>システムクロック 2 モニタビット</th></tr>
<tr><td>0</td><td>0</td><td>CLKS2 が CLKRC (CR クロック) に設定</td></tr>
<tr><td>0</td><td>1</td><td>CLKS2 が CLKMC (メインクロック) に設定</td></tr>
<tr><td>1</td><td>0</td><td>CLKS2 が CLKPLL (非変調 PLL クロック) に設定</td></tr>
<tr><td>1</td><td>1</td><td>CLKS2 が CLKSC (サブクロック) に設定</td></tr>
</table>
• SC2M[1:0] ビットの読出し値は SC2S[1:0] システムクロック 2 選択ビットに書き込まれた値と異なる場合があります。これは , 要求されたクロックモードの移行がまだ終了していないことを示します。
• クロックモードの移行 (同期メカニズムが起動中) は以下になります。SC2S[1:0] ビットで選択されたクロックが安定していても (クロックモニタビット 1), SC2M[1:0] ビットで示された動作中のクロックモードは SC2S[1:0] の設定と異なります。SC2M[1:0] ビットが更新されるのは , クロックモードの移行が終了した後です。
• 選択されたクロックが安定していない (SC2S[1:0]ビットで選択されたクロックのクロックモニタビットが "0" に設定されている) 場合は, クロックモードを切り換えることができません。
</td></tr>
<tr><td>bit 12</td><td>RCM:
CR クロック
モニタビット</td><td>
• このビットは , 内部 CR クロックが安定しているかどうかを示します。
• RCM =1 は , CR クロックが使用可能であることを示します。RCE が "0" に設定されていても RCM =1 の場合は , CR クロックがシステムクロック 1 または 2 として使用されているため , CR クロックは禁止されていません。
• RCM =0 は , CR クロックが禁止されているか , または CR 発振安定時間に入っていることを示します。
• リセットが発生すると, このビットが "0" に初期化され, MCU の動作が停止します。CR 発振安定待ち時間が経過すると, このビットが "1" に設定され, リセットシーケンスの実行に伴って MCU の動作が再開します。
</td></tr>
</table>

Table 6-2. クロックモニタレジスタ(CKMR)の各ビットの機能説明 (Sheet 2 of 2)

ビット名		機能
bit 13	MCM: メインクロック モニタビット	• このビットは , メインクロックが安定しているかどうかを示します。 • MCM =1 は , メインクロックが使用可能であることを示します。MCE が "0" に設定されていても MCM =1 の場合は , メインクロック /PLL クロックがシステムクロック 1 または 2 として使用されているため , メインクロックは禁止されていません。 • MCM =0 は , メイン発振器が禁止されているか , またはメイン発振安定時間に入っていることを示します。 • 電源リセットまたは外部リセット (RSTX 立下りエッジ) とメインクロック停止検出リセットが発生すると , このビットが "0" に初期化されます。RSTX の影響の詳細は , 「8.3 電源リセットおよび外部リセット後のスタートアップ」を参照してください。 • サブクロック停止検出リセット , ソフトウェアリセット , またはウォッチドッグリセットが発生しても , このビットはリセットされません。 (注意事項) メインクロック安定時間選択ビット CKSSR:MCST[2:0]は, リセットが発生すると, "111" (2^{18}/CLKMC 周期) にリセットされます。したがって, メイン発振器が起動してから 2^{18}/CLKMC 周期以内にリセットが発生した場合は, MCM が"0" にクリアされます。
bit 14	PCM: PLL クロック モニタビット	• このビットは , PLL クロックが安定しているかどうかを示します。 • PCM =1 は , PLL クロックが使用可能であることを示します。PCE が "0" に設定されていても PCM = 1 の場合は, PLLクロックがシステムクロック 1 または 2 として使用されているため, PLLクロックは禁止されていません。 • PCM = 0 は , PLL が禁止されているか , または PLL 安定時間に入っていることを示します。 • リセットが発生すると, このビットが "0" (PLLの禁止) に初期化されます。
bit 15	SCM: サブクロック モニタビット	• このビットは , サブクロックが安定しいるかどうかを示します。 • SCM =1 は , サブクロックが使用可能であることを示します。SCE が "0" に設定されていても SCM =1 の場合は, サブクロックがシステムクロック 1 または 2 として使用されているため, サブクロックは禁止されていません。 • SCM =0 は , サブクロックが禁止されているか , またはサブクロック発振安定時間に入っていることを示します。 • 電源リセットまたは外部リセットとサブクロック停止検出リセットが発生すると , このビットが "0" に初期化されます。 • メインクロック停止検出リセット , ソフトウェアリセット , またはウォッチドッグリセットが発生しても , このビットはリセットされません。 (注意事項) サブクロック安定時間選択ビット CKSSR:SCST[1:0] は,リセットが発生すると, "11" (2^{16}/CLKSC 周期) にリセットされます。したがって, サブ発振器が起動してから 2^{16}/CLKSC 周期以内にリセットが発生した場合は, SCM が "0" にクリアされます。

6.2.3 クロック安定化選択レジスタ(CKSSR)

クロック安定化選択レジスタ (CKSSR) は, PLL 発振回路の安定時間の選択とメインおよびサブ発振回路の帰還抵抗の制御に使用します。

クロック安定化選択レジスタ(CKSSR)の構成

クロック安定化選択レジスタ (CKSSR) の構成を Figure 6-5 に示し , 各ビットの機能を Table 6-3 で説明します。

Figure 6-5. クロック安定化選択レジスタ(CKSSR)の構成

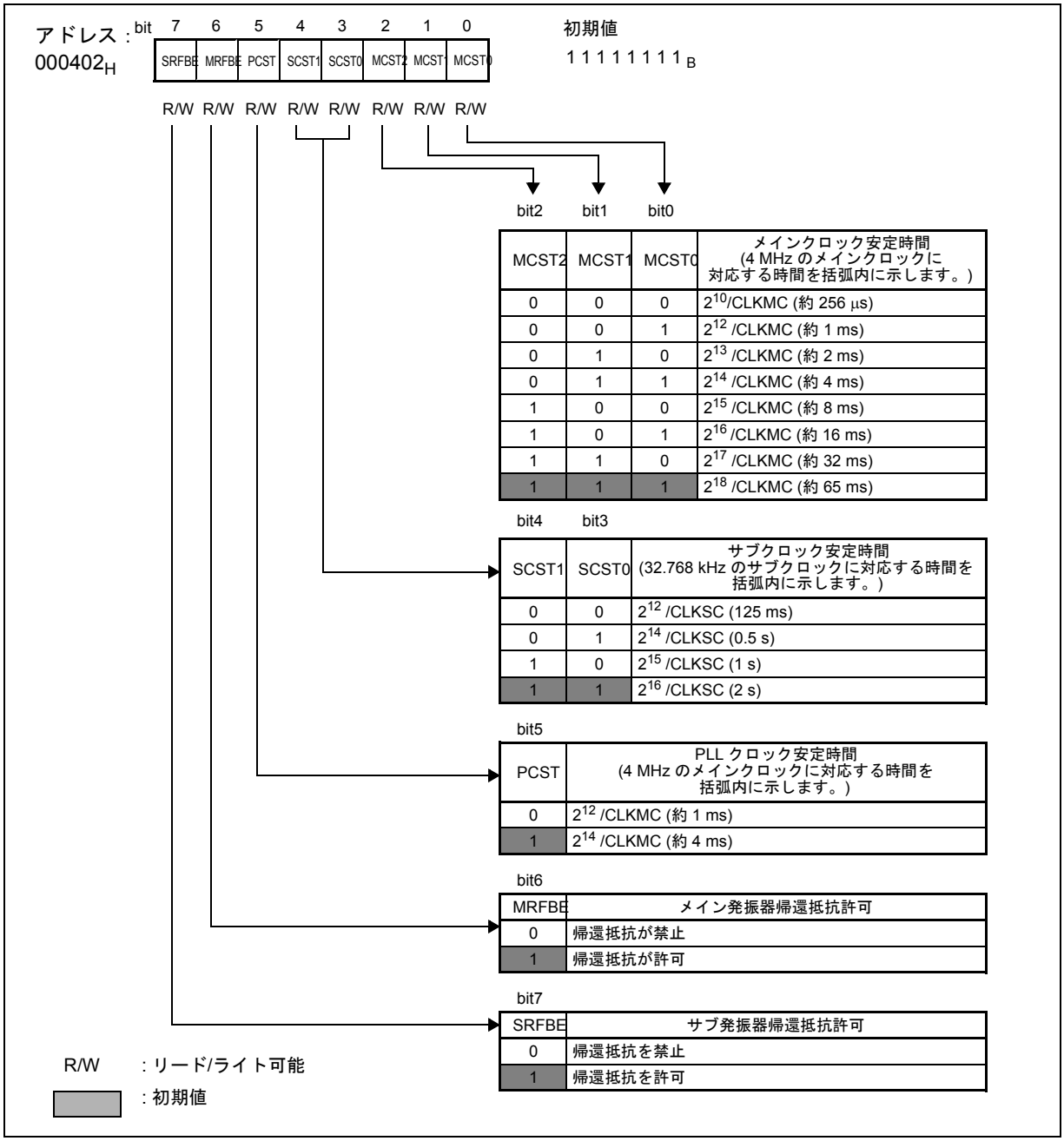


Table 6-3. クロック安定化選択レジスタ(CKSSR)の各ビットの機能説明 (Sheet 1 of 2)

<table>
<tr><th colspan="2">ビット名</th><th>機能</th></tr>
<tr><td>bit 0～bit 2</td><td>MCST2 ～ MCST0:
メインクロック安定時間選択ビット</td><td>
• これらのビットは , 以下の表に従ってメイン発振回路の安定時間を選択します。
<table>
<tr><th>bit2</th><th>bit1</th><th>bit0</th><th></th></tr>
<tr><th>MCST2</th><th>MCST1</th><th>MCST0</th><th>メインクロック安定時間
(4 MHz のメインクロックに対応する時間を括弧内に示します。)</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>2^{10}/CLKMC (約 256 µs)</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>2^{12}/CLKMC (約 1 ms)</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>2^{13}/CLKMC (約 2 ms)</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>2^{14}/CLKMC (約 4 ms)</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>2^{15}/CLKMC (約 8 ms)</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>2^{16} /CLKMC (約 16 ms)</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>2^{17}/CLKMC (約 32 ms)</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>2^{18}/CLKMC (約 65 ms)</td></tr>
</table>
• リセットが発生すると , このビットが "111" に初期化されます。
• これらのビットは , 上記のメインクロック (CLKMC) 周期が経過した後にメインクロックモニタ (MCM) ビットを設定するかを定義します。これにより , メインクロックがシステムクロックとして使用可能になります。
(注意事項)
発振安定待ち時間は, 使用している発振子に対して適切な値に設定する必要があります。
これらのビットを "111" に初期化すると, メイン発振が起動してから 2^{18}/CLKMC 周期以内にリセットが発生した場合に CKMR:MCM ビットがクリアされます。
</td></tr>
<tr><td>bit 3, bit 4</td><td>SCST1, SCST0:
サブクロック安定時間選択ビット</td><td>
• これらのビットは , 以下の表に従ってサブ発振回路の安定時間を選択します。
<table>
<tr><th>bit4</th><th>bit3</th><th></th></tr>
<tr><th>SCST1</th><th>SCST0</th><th>サブクロック安定時間
(32.768 kHz のサブクロックに対応する時間を括弧内に示します。)</th></tr>
<tr><td>0</td><td>0</td><td>2^{12}/CLKSC (125 ms)</td></tr>
<tr><td>0</td><td>1</td><td>2^{14}/CLKSC (0.5 s)</td></tr>
<tr><td>1</td><td>0</td><td>2^{15}/CLKSC (1 s)</td></tr>
<tr><td>1</td><td>1</td><td>2^{16}/CLKSC (2 s)</td></tr>
</table>
• リセットが発生すると , このビットが "11" に初期化されます。
• これらのビットは , 上記のサブクロック (CLKSC) 周期が経過した後にサブクロックモニタ (SCM) ビットを設定するかを定義します。これにより , サブクロックがシステムクロックとして使用可能になります。
(注意事項)
発振安定待ち時間は, 使用している発振子に対して適切な値に設定する必要があります。
これらのビットを "11" に初期化すると, サブ発振が起動してから 2^{16}/CLKSC 周期以内にリセットが発生した場合に CKMR:SCM ビットがクリアされます。
</td></tr>
</table>

Table 6-3. クロック安定化選択レジスタ(CKSSR)の各ビットの機能説明 (Sheet 2 of 2)

<table>
<tr><th colspan="2">ビット名</th><th>機能</th></tr>
<tr><td>bit 5</td><td>PCST:
PLLクロック安定時間選択ビット</td><td>

このビットは , 以下の表に従って PLL クロックの安定時間を選択します。

bit5
<table>
<tr><th>PCST</th><th>PLL クロック安定時間
(4 MHz のメインクロックに対応する時間を括弧内に示します。)</th></tr>
<tr><td>0</td><td>2^{12}/CLKMC (約 1 ms)</td></tr>
<tr><td>1</td><td>2^{14}/CLKMC (約 4 ms)</td></tr>
</table>

リセットが発生すると , このビットが "1" に初期化されます。
このビットは , 上記のメインクロック (CLKMC) 周期が経過した後に PLL クロックモニタ (PCM) ビットを設定します。これにより , PLL クロックがシステムクロックとして使用可能になります。
メインクロックが安定し (MCM =1), PCE が "1" に設定されると, PLL クロック安定時間のカウントが始まります。

</td></tr>
<tr><td>bit 6</td><td>MRFBE:
メイン発振器帰還抵抗許可ビット</td><td>

このビットは , メイン発振器の帰還抵抗を制御します。
このビットに "1" を書き込むと帰還抵抗が許可され , "0" を書き込むと帰還抵抗が禁止されます。
リセットが発生すると, このビットが "1" (抵抗を許可) に初期化されます。

</td></tr>
<tr><td>bit 7</td><td>SRFBE:
サブ発振器帰還抵抗許可ビット</td><td>

このビットは , サブ発振器の帰還抵抗を制御します。
このビットに "1" を書き込むと帰還抵抗が許可され , "0" を書き込むと帰還抵抗が禁止されます。
リセットが発生すると, このビットが "1" (抵抗を許可) に初期化されます。

</td></tr>
</table>

6.2.4 クロック周波数制御レジスタ(CKFCR)

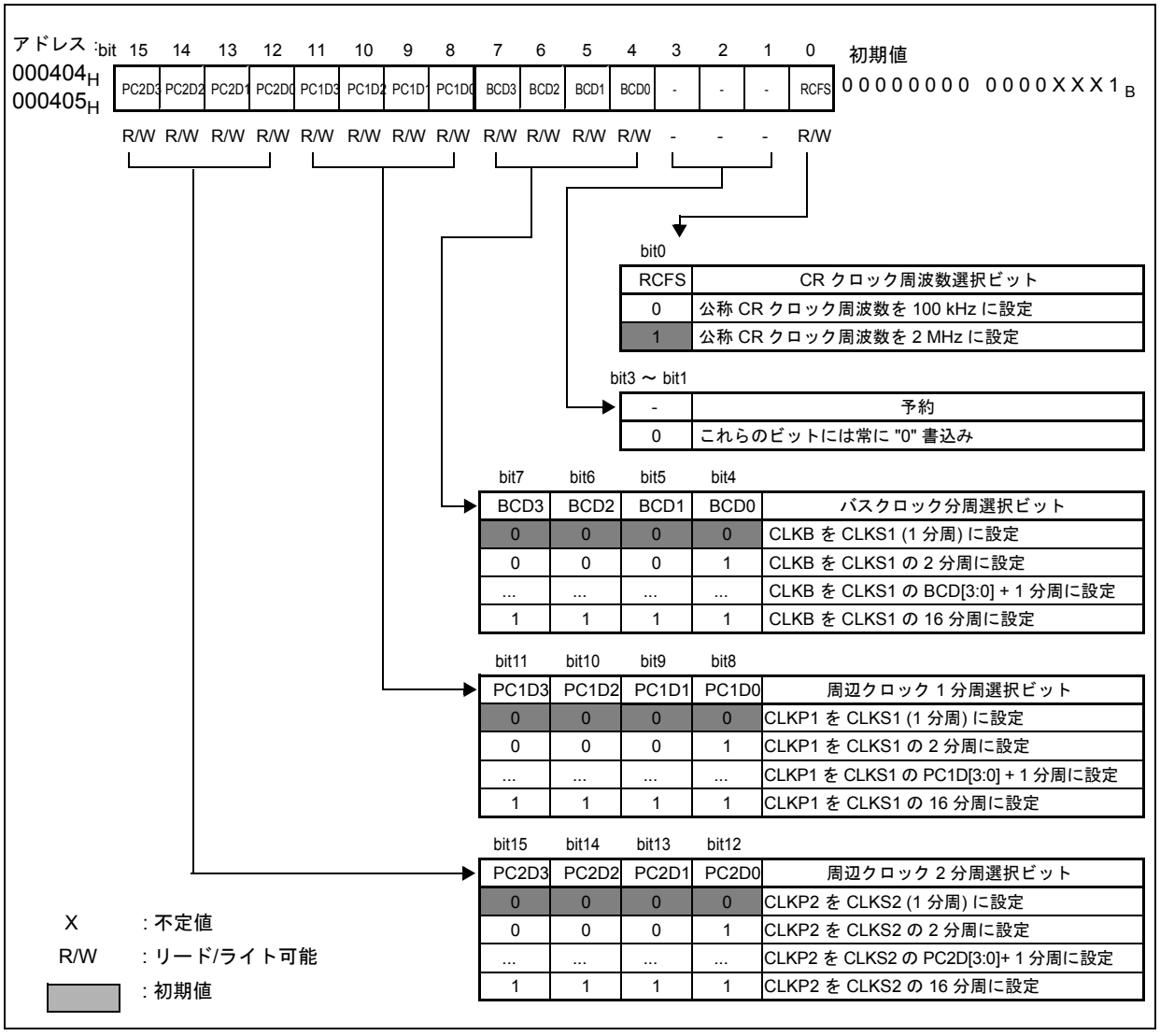
クロック周波数制御レジスタ (CKFCR) は, 周辺クロック分周器 (1 および 2), バスクロック分周器, および CR 発振器の周波数の制御に使用します。

クロック周波数制御レジスタ(CKFCR)の構成

クロック周波数制御レジスタ (CKFCR) の構成を Figure 6-6 に示し , 各ビットの機能を Table 6-4 で説明します。

このレジスタには 16 ビット幅 (CKFCR) および 8 ビット幅 (下位バイト :CKFCRL, 上位バイト : CKFCRH) でアクセス可能です。

Figure 6-6. クロック周波数制御レジスタ(CKFCR)の構成



bit0

RCFS	CR クロック周波数選択ビット
0	公称 CR クロック周波数を 100 kHz に設定
1	公称 CR クロック周波数を 2 MHz に設定

bit3 ～ bit1

-	予約
0	これらのビットには常に "0" 書込み

bit7	bit6	bit5	bit4	
BCD3	BCD2	BCD1	BCD0	バスクロック分周選択ビット
0	0	0	0	CLKB を CLKS1 (1 分周) に設定
0	0	0	1	CLKB を CLKS1 の 2 分周に設定
...	...	...	...	CLKB を CLKS1 の BCD[3:0] + 1 分周に設定
1	1	1	1	CLKB を CLKS1 の 16 分周に設定

bit11	bit10	bit9	bit8	
PC1D3	PC1D2	PC1D1	PC1D0	周辺クロック 1 分周選択ビット
0	0	0	0	CLKP1 を CLKS1 (1 分周) に設定
0	0	0	1	CLKP1 を CLKS1 の 2 分周に設定
...	...	...	...	CLKP1 を CLKS1 の PC1D[3:0] + 1 分周に設定
1	1	1	1	CLKP1 を CLKS1 の 16 分周に設定

bit15	bit14	bit13	bit12	
PC2D3	PC2D2	PC2D1	PC2D0	周辺クロック 2 分周選択ビット
0	0	0	0	CLKP2 を CLKS2 (1 分周) に設定
0	0	0	1	CLKP2 を CLKS2 の 2 分周に設定
...	...	...	...	CLKP2 を CLKS2 の PC2D[3:0]+ 1 分周に設定
1	1	1	1	CLKP2 を CLKS2 の 16 分周に設定

Table 6-4. クロック周波数制御レジスタ(CKFCR)の各ビットの機能説明 (Sheet 1 of 2)

ビット名		機能
bit 0	RCFS: CR クロック 周波数選択ビット	• このビットは , 内部 CR 発振器のクロック周波数の設定に使用されます。 • このビットに "1" を書き込むと公称 CR クロック周波数が 2 MHz に設定され , "0" を書き込むと公称周波数が 100 kHz に設定されます。 • 両方の設定の最小周波数と最大周波数については , データシートを参照してください。 • このビットはリセットごとに "1" (2 MHz) に初期化されます。 (注意事項) CR クロック周波数は随時変更可能です。ただし, クロック停止検出回路およびウォッチドッグリセットがCRクロックで動作可能であることとCRクロック周波数を変更するとこれらの回路の動作が影響を受けることを考慮してください。ウォッチドッグリセットモジュールが動作モードにある場合, CR クロック周波数は変更不可であり, 変更するとウォッチドッグリセットが発生します。詳細は, 「第12章 ウォッチドッグリセットとクロック停止検出リセット」の章の説明を参照してください。

Table 6-4. クロック周波数制御レジスタ(CKFCR)の各ビットの機能説明 (Sheet 2 of 2)

<table>
<tr><th colspan="2">ビット名</th><th>機能</th></tr>
<tr><td>bit 1 ～ bit 3</td><td>予約</td><td>• これらのビットに常に "0" を書き込みます。
• これらのビットの読出し値は未定義です。
• これらのレジスタへのリードモディファイライト系命令は無効です。</td></tr>
<tr><td>bit 4 ～ bit 7</td><td>BCD3 ～ BCD0:
バスクロック分周選択ビット</td><td>• これらのビットは , 以下の表に従ってバスクロック (CLKB) 用のクロック分周器を制御します。
bit7 bit6 bit5 bit4
BCD3 | BCD2 | BCD1 | BCD0 | バスクロック分周選択ビット
0 | 0 | 0 | 0 | CLKB が CLKS1 (1 分周) になる
0 | 0 | 0 | 1 | CLKB が CLKS1 の 2 分周になる
0 | 0 | 1 | 0 | CLKB が CLKS1 の 3 分周になる
– | – | – | – | CLKB が CLKS1 の BCD[3:0] + 1 分周になる
1 | 1 | 1 | 0 | CLKB が CLKS1 の 15 分周になる
1 | 1 | 1 | 1 | CLKB が CLKS1 の 16 分周になる
• これらのビットはリセットごとに "0000" (CLKB = CLKS1) に初期化されます。</td></tr>
<tr><td>bit 8 ～ bit 11</td><td>PC1D3 ～ PC1D0:
周辺クロック 1 分周選択ビット</td><td>• これらのビットは , 以下の表に従って周辺クロック (CLKP1) 用のクロック分周器を制御します。
bit11 bit10 bit9 bit8
PC1D3 | PC1D2 | PC1D1 | PC1D0 | 周辺クロック 1 分周選択ビット
0 | 0 | 0 | 0 | CLKP1 が CLKS1 (1 分周) になる
0 | 0 | 0 | 1 | CLKP1 が CLKS1 の 2 分周になる
0 | 0 | 1 | 0 | CLKP1 が CLKS1 の 3 分周になる
– | – | – | – | CLKP1 が CLKS1 の PC1D[3:0] + 1 分周になる
1 | 1 | 1 | 0 | CLKP1 が CLKS1 の 15 分周になる
1 | 1 | 1 | 1 | CLKP1 が CLKS1 の 16 分周になる
• これらのビットはリセットごとに "0000" (CLKP1 = CLKS1) に初期化されます。</td></tr>
<tr><td>bit 12～ bit 15</td><td>PC2D3 ～ PC2D0:
周辺クロック 2 分周選択ビット</td><td>• これらのビットは , 以下の表に従って周辺クロック (CLKP2) 用のクロック分周器を制御します。
bit15 bit14 bit13 bit12
PC2D3 | PC2D2 | PC2D1 | PC2D0 | 周辺クロック 2 分周選択ビット
0 | 0 | 0 | 0 | CLKP2 が CLKS2 (1 分周) になる
0 | 0 | 0 | 1 | CLKP2 が CLKS2 の 2 分周になる
0 | 0 | 1 | 0 | CLKP2 が CLKS2 の 3 分周になる
– | – | – | – | CLKP2 が CLKS2 の PC2D[3:0] + 1 分周になる
1 | 1 | 1 | 0 | CLKP2 が CLKS2 の 15 分周になる
1 | 1 | 1 | 1 | CLKP2 が CLKS2 の 16 分周になる
• これらのビットはリセットごとに "0000" (CLKP2 = CLKS2) に初期化されます。</td></tr>
</table>

6.2.5 PLL制御レジスタ(PLLCR)

PLL制御レジスタ(PLLCR)は,PLL逓倍回路のすべての機能とCLKP3の周辺クロック分周器の制御に使用します。

PLL制御レジスタ(PLLCR)の構成

PLL 制御レジスタ (PLLCR) の構成を Figure 6-7 に示し , 各ビットの機能を Table 6-5 で説明します。

このレジスタには 16 ビット幅 (PLLCR) および 8 ビット幅 (下位バイト : PLLCRL, 上位バイト : PLLCRH) でアクセス可能です。

Figure 6-7. PLL制御レジスタ(PLLCR)の構成

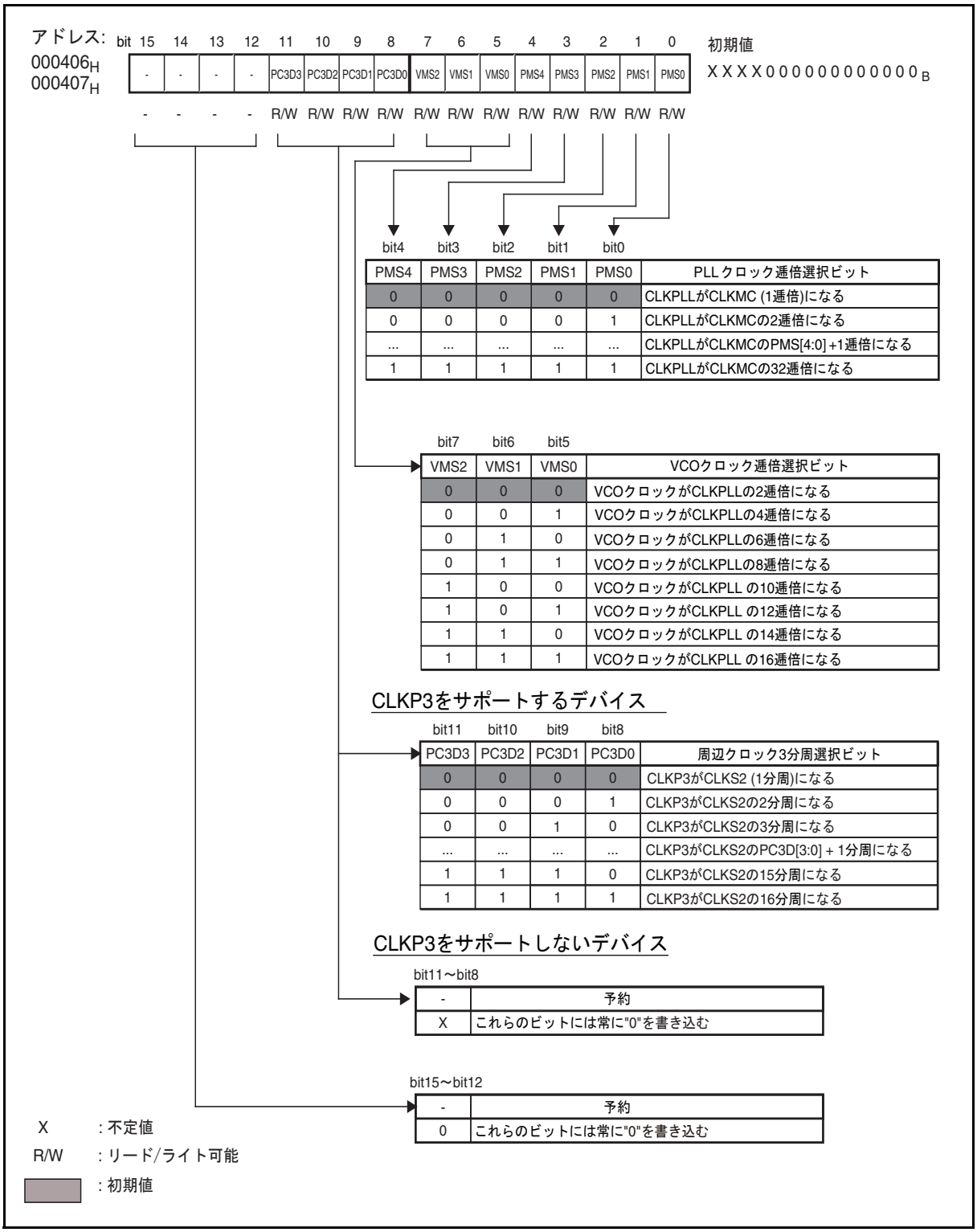


Table 6-5. PLL制御レジスタ(PLLCR)の各ビットの機能説明 (Sheet 1 of 2)

<table>
<tr><th colspan="2">ビット名</th><th>機能</th></tr>
<tr><td>bit 0 ～ bit 4</td><td>PMS4 ～ PMS0:
PLL クロック
逓倍選択ビット</td><td>
• これらのビットは , 以下の表に従って PLL クロック逓倍係数を制御します。
<table>
<tr><th>bit4</th><th>bit3</th><th>bit2</th><th>bit1</th><th>bit0</th><th></th></tr>
<tr><th>PMS4</th><th>PMS3</th><th>PMS2</th><th>PMS1</th><th>PMS0</th><th>PLL クロック逓倍選択ビット</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>CLKPLL が CLKMC (1 逓倍) になる</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>CLKPLL が CLKMC の 2 逓倍になる</td></tr>
<tr><td>～</td><td>～</td><td>～</td><td>～</td><td>～</td><td>CLKPLL が CLKMC の PMS[4:0] + 1 逓倍になる</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>CLKPLL が CLKMC の 32 逓倍になる</td></tr>
</table>
• これらのビットはリセットごとに "00000" に初期化されます。
• PLL 逓倍設定では , メインクロック周波数 (CLKMC) と PLL 出力周波数 (CLKPLL) との比を定義します。
• 必ずこれらのビットを設定した後で , CKSR: PCE ビットに "1" を書き込むか , PLL クロックを CLKS1 または CLKS2 用のソースとして選択することによって PLL を許可してください。
• PLL を起動した後で , これらのビットの設定を変更しないでください。
• 設定範囲はCLKVCOおよびCLKS1/CLKS2 (データシートを参照してください)に許可された周波数領域と使用するメインクロックの周波数とに依存します。
</td></tr>
<tr><td>bit 5 ～ bit 7</td><td>VMS2 ～ VMS0:
VCO クロック
逓倍選択ビット</td><td>
• これらのビットは , 以下の表に従って VCO クロック逓倍係数を制御します。
<table>
<tr><th>bit7</th><th>bit6</th><th>bit5</th><th></th></tr>
<tr><th>VMS2</th><th>VMS1</th><th>VMS0</th><th>VCO クロック逓倍選択ビット</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>VCO クロックが CLKPLL の 2 逓倍になる</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>VCO クロックが CLKPLL の 4 逓倍になる</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>VCO クロックが CLKPLL の 6 逓倍になる</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>VCO クロックが CLKPLL の 8 逓倍になる</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>VCO クロックが CLKPLL の 10 逓倍になる</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>VCO クロックが CLKPLL の 12 逓倍になる</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>VCO クロックが CLKPLL の 14 逓倍になる</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>VCO クロックが CLKPLL の 16 逓倍になる</td></tr>
</table>
• これらのビットはリセットごとに "000" に初期化されます。
• VCO クロック逓倍設定では , PLL 出力クロック周波数 (CLKPLL) と PLL 内部 VCO の出力周波数との関係を定義します。
• 必ずこれらのビットを設定した後で , CKSR: PCE ビットに "1" を書き込むか , PLL クロックを CLKS1 または CLKS2 用のソースとして選択することによって PLL を許可してください。
PLL を起動した後で, これらのビットの設定を変更しないでください。
VCOクロック逓倍係数は,PLL入力クロックCLKMCとPLL逓倍設定に合わせて設定する必要があります。詳細は, 「6.4 PLLの構成」を参照してください。
</td></tr>
</table>

Table 6-5. PLL制御レジスタ(PLLCR)の各ビットの機能説明 (Sheet 2 of 2)

<table>
<tr><th colspan="2">ビット名</th><th>機能</th></tr>
<tr><td>bit 8 ～ bit 11</td><td>PC3D0 ～ PC3D3:
周辺クロック
3分周選択ビット</td><td>
CLKP3 をサポートするデバイス
• これらのビットは以下の表に従って周辺クロック 3(CLKP3) の分周を制御します。
bit11 bit10 bit9 bit8
<table>
<tr><th>PC3D3</th><th>PC3D2</th><th>PC3D1</th><th>PC3D0</th><th>周辺クロック 3 分周選択ビット</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>CLKP3 = CLKS2 (1 分割)</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>CLKP3 = CLKS2 2 分割</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>CLKP3 = CLKS2 3 分割</td></tr>
<tr><td>…</td><td>…</td><td>…</td><td>…</td><td>CLKP3 = CLKS2 PC3D[3:0] + 1 分割</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>CLKP3 = CLKS2 15 分割</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>CLKP3 = CLKS2 16 分割</td></tr>
</table>
• これらのビットはリセットごとに "0000" (CLKP3 = CLKS2) に初期化されます。
CLKP3 をサポートしないデバイス
これらのビットは予約です。
• これらのビットに常に "0" を書き込みます。
• これらのビットの読出し値は未定義です。
• これらのレジスタへのリードモディファイライト系命令は無効です。
</td></tr>
<tr><td>bit 12～ bit 15</td><td>予約</td><td>
• これらのビットに常に "0" を書き込みます。
• これらのビットの読出し値は未定義です。
• これらのレジスタへのリードモディファイライト系命令は無効です。
</td></tr>
</table>

6.2.6 クロック入力およびLVD制御レジスタ(CILCR)

クロック入力およびLVD制御レジスタ(CILCR) は発振回路と低電圧検出レベルの追加機能に使用します。

クロック入力およびLVD制御レジスタ(CILCR)の構成

Figure 6-8 にクロック入力およびLVD制御レジスタ(CILCR)の構成を示し, Table 6-6に各ビットの機能を説明します。

このレジスタは 16 ビット幅 (VRCR) および 8 ビット幅 (下位バイト : VRCR, 上位バイト CILCR) にアクセスできます。

Figure 6-8. クロック入力およびLVD制御レジスタ(CILCR)の構成

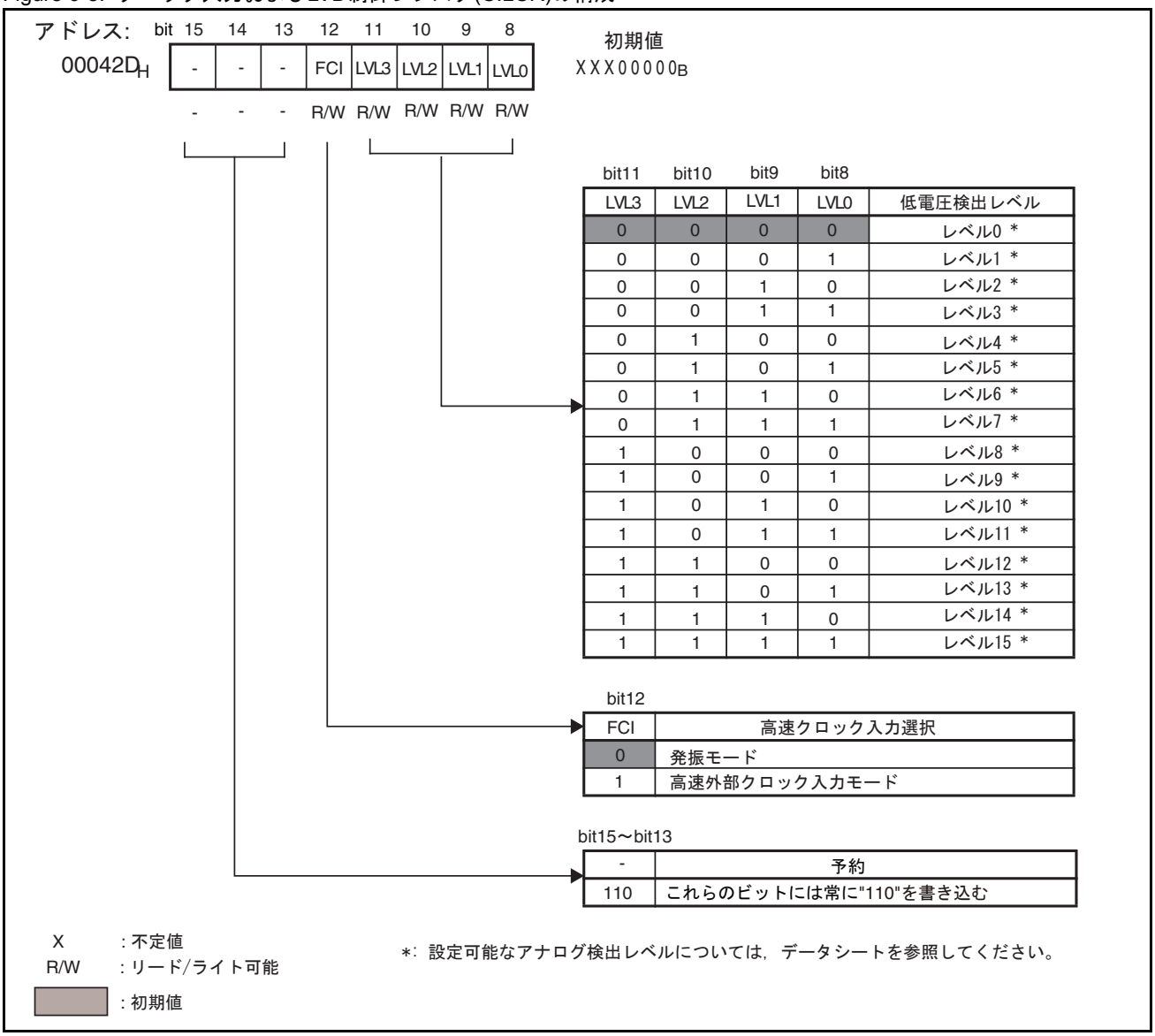


Table 6-6. クロック入力およびLVD制御レジスタ(CILCR)の各ビットの機能説明

<table>
<tr><th colspan="2">ビット名</th><th>機能</th></tr>
<tr><td>bit 8～ bit 11</td><td>LLVL3 ～ LVL0:
低電圧検出レベル選択ビット</td><td>
• T これらのビットは , 低電圧検出器 (LVD) のアナログしきい値を制御します。
<table>
<tr><th>LVL3</th><th>LVL2</th><th>LVL1</th><th>LVL0</th><th>低電圧検出レベル</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>レベル 0 *</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>レベル 1 *</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>レベル 2 *</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>レベル 3 *</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>レベル 4 *</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>レベル 5 *</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>レベル 6 *</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>レベル 7 *</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>レベル 8 *</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>レベル 9 *</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>レベル 10 *</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>レベル 11 *</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>レベル 12 *</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>レベル 13 *</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>レベル 14 *</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>レベル 15 *</td></tr>
</table>
• これらのビットはリセットごとに "000" に初期化されます。
• この機能が有効で外部供給電圧降下が選択したしきい値を超えた場合 , 低電圧検出回路はリセット要求を出力します。
• *: 設定可能なアナログしきい値については, データシートを参照してください。詳細については「第8章 リセットとスタートアップ」を参照してください。
</td></tr>
<tr><td>bit 12</td><td>FCI:
高速クロック入力選択ビット</td><td>
• このビットはリセットで "0" に初期化されます。
• このビットはメイン発振が発振モードか高速クロック入力モードかを選択します。
<table>
<tr><th>FCI</th><th>高速クロック入力制御</th></tr>
<tr><td>0</td><td>発振モード</td></tr>
<tr><td>1</td><td>高速外部クロック入力モード</td></tr>
</table>
• メイン発振用の端子 X0 と X1 に水晶発振器か共振器を接続するときは , このビットを "0" に設定します。このモードでは X0/X1 端子に外部低周波クロックを接続することも可能です。
• 外部低周波クロックを入力する場合は , 外部クロック (メイン) に切り換える前に , このビットを "1" に設定してください。
• 両モードの設定周波数範囲は , データシートに記載されています。
• この機能はすべてのデバイスで使用できません。詳細はデータシートを参照してください。
</td></tr>
<tr><td>bit 13 ～ bit 15</td><td>予約ビット</td><td>
• これらのビットに常に "110" を書き込みます。
• これらのビットの読出し値は未定義です。
• これらのレジスタへのリードモディファイライト系命令は無効です。
</td></tr>
</table>

6.3 クロックモード

クロックモードには, CR クロックモード, メインクロックモード, PLL クロックモード, およびサブクロックモードの 4 種類があります。

6.3.1 クロックモードの定義

MCU のクロックモードは , SC1S[1:0] ビットで選択されたシステムクロック 1 (CLKS1) 用のソースで定義します。CLKS1 は , CPU, メモリ , DMA コントローラ , および外部バスインタフェースを結ぶ内部バスに供給するバスクロック (CLKB) のマスタクロックになるだけでなく , 大部分の周辺モジュールに供給する周辺クロック 1 (CLKP1) のマスタクロックにもなります。

システムクロック 2 (CLKS2) の選択は , クロックモードに関係なく行うことができます。

- CR クロックモード

CR クロックモードでは , 内部 CR 発振器から生成されたクロックがシステムクロック 1 (CLKS1) に使用されます。このため , バスクロック (CLKB) と周辺クロック 1 (CLKP1) は CR 発振器から供給されます。CR クロック周波数は , 公称 100 kHz と 2 MHz 間の CR クロック周波数選択ビット (CKFCR: RCFS) で選択可能です。

メインクロック , PLL クロック , およびサブクロックを禁止するには , それぞれ MCE ビット , PCE ビット , および SCE ビットを使用します (これらがシステムクロック 2 やウォッチドッグ用に使用されていない限り)。これらのクロックを禁止すると , 対応するソースクロックタイマも禁止されます。

- メインクロックモード

メインクロックモードでは , メイン発振器の出力信号がシステムクロック 1 (CLKS1) に使用されます。このため , バスクロック (CLKB) と周辺クロック 1 (CLKP1) は CLKMC から供給されます。

CR クロック , PLL クロック , およびサブクロックを禁止するには , それぞれ RCE ビット , PCE ビット , および SCE ビットを使用します (これらがシステムクロック 2, ウォッチドッグ, クロック停止検出機能に使用されていない限り)。これらのクロックを禁止すると , 対応するソースクロックタイマも禁止されます。

- PLL クロックモード

PLL クロックモードでは , 変調 PLL クロックまたは非変調 PLL クロックのいずれかがシステムクロック 1 (CLKS1) に使用されます。このため , バスクロック (CLKB) と周辺クロック 1 (CLKP1) は CLKPLL または CLKMOD から供給されます。クロックモジュレータの設定によって , CLKPLL を使用するか CLKMOD を使用するかが決まります。詳細は , 「第 7 章 クロックモジュレータ」を参照してください。

このクロック信号の周波数は , PLL 制御レジスタとメインクロックの設定によって異なります。

CR クロックとサブクロックを禁止するには, それぞれ RCE ビットと SCE ビットを使用します (これらがシステムクロック 2, ウォッチドッグ , クロック停止検出機能に使用されていない限り)。これらのクロックを禁止すると , 対応するソースクロックタイマも禁止されます。PLL クロックモードでメインクロックを禁止することはできません。

- サブクロックモード

サブクロックモードでは , サブ発振器 CLKSC の出力信号がシステムクロック 1 (CLKS1) に使用されます。このため , バスクロック (CLKB) と周辺クロック 1 (CLKP1) は CLKSC から供給されます。

CR クロック , メインクロック , および PLL クロックを禁止するには , それぞれ RCE ビット , MCE ビット , および PCE ビットを使用します (これらがシステムクロック 2, ウォッチドッグ , クロック停止検出機能に使用されていない限り)。これらのクロックを禁止すると , 対応するソースクロックタイマも禁止されます。

6.3.2 クロックソースの切換え

クロックソースの切換えは , CLKS1 または CLKS2 用のクロックソースを変更することを意味します。

- クロックソースの切換え
  - クロックソースの選択は, CLKS1 と CLKS2 それぞれ独立に行うことができます。
  - クロックソースを切り換えるには, システムクロック選択ビット (システムクロック 1 の場合は SC1S[1:0], システムクロック 2 の場合は SC2S[1:0]) の値を書き換えます。

- 異なるクロックソースに移行できるのは, 選択されたクロックの準備が整っている (つまり, クロックモニタレジスタCKMR内の対応する準備完了フラグRCM,MCM,PCM,またはSCMが設定されている)場合に限ります。
- 選択されたクロックの準備が整っている場合は, 同期メカニズムを起動するとクロックソースの移行が行われます。このクロック切換えが終了すると, クロックモニタレジスタのシステムクロックモニタビット (システムクロック 1 の場合は SC1M, システムクロック 2 の場合は SC2M) が新しい値に変わります。
- 事前に選択されたモードへの切換えが終了する前にシステムクロック選択ビットに別の値を書き込むと, 最初の移行要求が取り消されます。
- 選択されたクロックが使用禁止であるか, または安定していない場合は, 選択されたクロックが使用可能になり,かつ安定する(クロック準備完了フラグCKMR:RCM,MCM,PCM, またはSCMが設定される)まで新しいクロックモードへの切換えが延期されます。
- 割込みによってストップモードが解除されると, システムクロックセレクタは, クロック準備完了モニタビットに関係なく SC1S[1:0]/SC2S[1:0] ビットで定義されたクロックをそのまま選択します。使用禁止になっているクロックは, 割込みによる再起動に使用されるため, ストップモードに切り換える前に選択してはなりま
  せん。

■ リセットおよびクロックソース

両方のシステムクロック (CLKS1 と CLKS2) 用のクロックソースは , 何らかのリセットが発生すると CR クロック (CLKRC) に設定されます。それぞれのリセットが終了するごとに 64 CR クロック周期の CR クロック安定時間に入ります。

■ CLKP2またはCLKP3をクロックソースとして使用している周辺リソースへのアクセス

SC1S または SC2S ビットの設定を変更した直後に , 周辺クロック 2 (CLKP2) をクロックソースとして使用しているリソースにアクセス (読出し / 書込み) しないでください。必ず SC1M/SC2M モニタビットを読み出すことで , 要求されたクロック移行が終了しているかどうかを確認してください。CLKP2 をクロックソースとして使用しているリソースにアクセスできるのは , SC1M[1:0] と SC1S[1:0] が一致しており , かつ SC2M[1:0] と SC2S[1:0] が一致している場合に限ります。

6.3.3 ソースクロックの起動と禁止

- それぞれのリセットが行われた後, メイン発振器, サブ発振器, および CR 発振器は許可されますが, PLL 逓倍回路は禁止されます。
- システムクロック CLKS1 または CLKS2 用に選択されたソースクロックが常に起動します。
- これらのシステムクロック用に使用されていないソースクロックを許可/禁止するには, CKSR レジスタ内の対応する許可ビットを使用します。
- 退避中のソースクロックを禁止するには, CKSR レジスタ内の対応する許可ビットを "0" に設定します。
- ソースクロックが特定機能または迅速なシステムクロックの切換えに必要となる場合にソースクロックを許可するには, CKSR レジスタ内の対応する許可ビットを "1" に設定します。起動したクロックが安定すると, CKMR レジスタ内の対応するクロックモニタビットが設定され, このクロックが「準備完了」であることを示します。
- メインクロックを許可する必要があるのは, PLL クロックを許可する場合です (MCE が "0" に設定されている場合, PCE の設定は効果がありません)。
- メインクロックは PLL 逓倍回路用の入力信号であるため, MCE を "0" に設定すると, システムクロック 1 または 2 がメインクロック/PLL クロックに設定されている場合にメイン発振器が禁止されません。

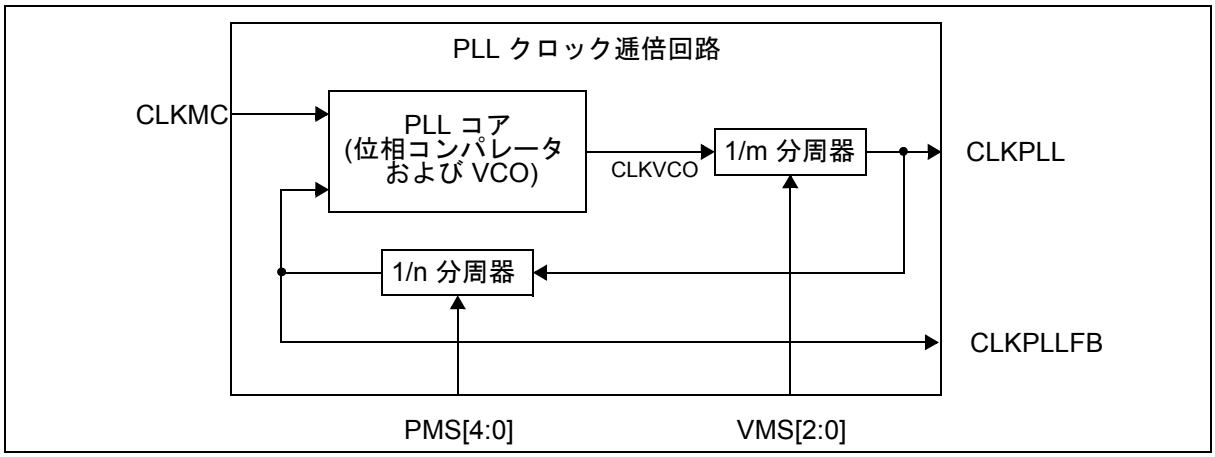
6.4 PLLの構成

PLL 逓倍回路は, メインクロックからの PLL クロックの生成に使用されます。
32 種類の逓倍値 (1逓倍～32逓倍) が利用可能です。

6.4.1 PLLクロック逓倍回路のコンポーネント

以下のブロックダイヤグラムは PLL のモジュールを示します。

Figure 6-9. PLL回路のブロックダイヤグラム



■ PLL コア (位相コンパレータおよび VCO)

PLL コアは , 電圧制御発振器 (VCO) を制御する位相コンパレータで構成されます。位相コンパレータは , 入力クロック CLKMC (メインクロック) と "1/n 分周器 " の出力クロックを比較します。VCO から生成される出力クロック CLKVCO の周波数は , "1/m 分周器 " と "1/n 分周器 " の設定によって異なります。CLKVCO = CLKMC × m × n

CLKVCO の周波数領域は , データシートを参照してください。

■ "1/m 分周器" (VCO クロック分周器)

"1/m 分周器 " は , VCO クロック逓倍選択ビット (VMS[2:0]) で制御されます。これにより , PLL 出力クロック CLKPLL と VCO 出力クロック CLKVCO との周波数関係が定義されます。デューティ周期が 50% の PLL 出力クロックを保証するために最小分周値は "2" になります。動作範囲内に VCO を維持する目的にも使用されます。

CLKVCO 周波数がいつもデータシートに記載されている許容範囲内の場合 , これらのビットはメイン発振周波数 CLKMC と CLKPLL に応じて設定してください。さもなければ , PLL はロックされず , 結果として CLKPLL 周波数は不定になります。

PLL 起動後に , これらのビットの設定を変えないでください。

■ "1/n 分周器" (PLL クロック分周器)

"1/n 分周器 " は , PLL クロック逓倍選択ビット (PMS[4:0]) で制御されます。PLL クロック逓倍回路の周波数逓倍値 (CLKMC と CLKPLL との関係) を定義します。1 ～ 32 の逓倍値が利用可能です。ただし , CLKMC, CLKVCO および PLKPLL (->CLKS1/CLKS2) に許可された周波数の最小値と最大値は厳守しなければなりません。PLL を起動した後で , これらのビットの設定を変更しないでください。

6.4.2 VCOクロック逓倍選択ビットVMS[2:0]の設定

"1/m 分周器" は, データシートに規定された広範な PLL クロック周波数に対応した動作範囲内に VCO を維持するようにプログラム可能です。有効な CLKVCO クロック周波数をもたらす VMS[2:0] ビットに値を書いてください。表の下に , 4MHz メインクロック用の推薦設定を示しています。ほかのメインクロック周波数については , 次の公式に基づいて VMS[2:0[ビットの設定を調整してください。

CLKVCO=CLKMC × m × n

Table 6-7. PMS[3:0]ビットとVMS[2:0]ビットの推奨設定 (CLKMC=4MHz)

メイン発振周波数CLKMC	要求されたPLL出力周波数CLKPLL	PMS[4:0]ビット(n 分周値)の設定	VMS[2:0]ビット(m分周値)の設定	VCO出力周波数CLKVCO
4 MHz	4 MHz	"00000" (div 1)	"111" (div 16)	64 MHz
4 MHz	8 MHz	"00001" (div 2)	"101" (div 12)	96 MHz
4 MHz	12 MHz	"00010" (div 3)	"011" (div 8)	96 MHz
4 MHz	16 MHz	"00011" (div 4)	"010" (div 6)	96 MHz
4 MHz	20 MHz	"00100" (div 5)	"010" (div 6)	120 MHz
4 MHz	24 MHz	"00101" (div 6)	"001" (div 4)	96 MHz
4 MHz	28 MHz	"00110" (div 7)	"001" (div 4)	112 MHz
4 MHz	32 MHz	"00111" (div 8)	"001" (div 4)	128 MHz
4 MHz	36 MHz	"01000" (div 9)	"000" (div 2)	72 MHz
4 MHz	40 MHz	"01001" (div 10)	"000" (div 2)	80 MHz
4 MHz	44 MHz	"01010" (div 11)	"000" (div 2)	88 MHz
4 MHz	48 MHz	"01011" (div 12)	"000" (div 2)	96 MHz
4 MHz	52 MHz	"01100" (div 13)	"000" (div 2)	104 MHz
4 MHz	56 MHz	"01101" (div 14)	"000" (div 2)	112 MHz
4 MHz	60 MHz	"01110" (div 15)	"000" (div 2)	120 MHz
4 MHz	64 MHz	"01111" (div 16)	"000" (div 2)	128 MHz
4 MHz	68 MHz	"10000" (div 17)	"000" (div 2)	136 MHz
4 MHz	72 MHz	"10001" (div 18)	"000" (div 2)	144 MHz
4 MHz	76 MHz	"10010" (div 19)	"000" (div 2)	152 MHz
4 MHz	80 MHz	"10011" (div 20)	"000" (div 2)	160 MHz
4 MHz	84 MHz	"10100" (div 21)	"000" (div 2)	168 MHz
4 MHz	88 MHz	"10101" (div 22)	"000" (div 2)	176 MHz
4 MHz	92 MHz	"10110" (div 23)	"000" (div 2)	184 MHz
4 MHz	96 MHz	"10111" (div 24)	"000" (div 2)	192 MHz
4 MHz	100 MHz	"11000" (div 25)	"000" (div 2)	200 MHz

6.5 発振安定待ち時間

電源をオンにした場合，ストップモードを解除した場合，または禁止されたクロックを許可した場合は，クロックが使用可能になるまでに発振安定待ち時間を要します。

6.5.1 発振安定待ち時間

通常 , X0/X1 および X0A/X1A 端子に接続可能なセラミック発振器と水晶発振器には , 発振開始時に固有振動数 (発振周波数) で安定するために数ミリ秒が必要となります。また , 内部 PLL 逓倍回路と CR 発振器には , 起動後に一定の安定時間が必要となります。このため , 起動したクロックを伴う CPU 動作は発振が開始した直後には許可されず , 発振が完全に安定した後にのみ許可されます。

発振安定待ち時間が経過すると , 対応するクロック準備完了モニタビットが設定されるため , クロックをシステムクロックとして選択することができます。

メイン発振器とサブ発振器の発振安定時間は発振器の種類 (水晶 , セラミックなど) によって異なるため , 使用する発振器に対して適切な発振安定待ち時間を選択する必要があります。発振安定待ち時間を選択するには , クロック安定化選択レジスタ (CKSSR) を設定します。CR 発振器の安定時間は固定されており , PLL の安定時間は PLL クロック周波数とメインクロック周波数に合わせて選択できます。

クロックソースを切り換えた場合 , 新たに選択されたクロックが安定するまで MCU はそれまで選択されていたクロックで動作します。対応するクロック準備完了フラグが設定されると , MCU は指定のクロックに切り換わります。

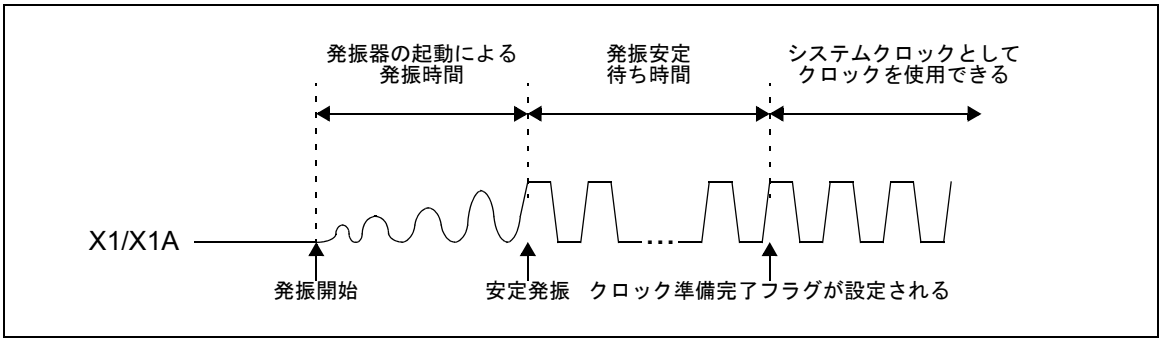
電源リセットまたは外部リセットが発生すると , ソースクロックタイマとクロック準備完了モニタビットがすべてクリアされ , 対応する発振安定待ち時間に入ります。

ソフトウェアリセットおよびウォッチドッグリセットが発生すると , ソースクロックタイマとメイン / サブクロックモニタビットは影響を受けませんが (リセットの前にクロックが許可された場合は , メイン / サブ発振安定待ち時間に入らない), CR/PLL クロックモニタビットがクリアされます (PLL が禁止され , CR 発振安定時間に入る)。

クロック停止リセットが発生すると , リセットの原因となったクロックのソースクロックタイマとクロックモニタビットがクリアされます。このクロックをこれ以上使用してはなりません。PLL が禁止され , CR 発振安定時間に入ります。

起動直後の発振器の動作を Figure 6-10 に示します。

Figure 6-10. 発振起動直後の動作



■ CR クロック安定時間

CR クロック安定時間は 64 CR クロック周期に固定されています。ただし , 電源リセットまたは外部リセット (RSTX 立下りエッジ) が発生すると , リセット延長回路により 700 CR クロック周期 (電源リセット) または 700 CR クロック周期 (外部リセット) の延長待ち時間に入ります (詳細は , 「8.3 電源リセットおよび外部リセット後のスタートアップ」を参照してください)。

■ メインクロック安定時間

メインクロック安定時間は , CKSSR レジスタの MCST[2:0] ビットで選択できます。Table 6-3 に示されているように 8 通りの設定が可能です。

■ サブクロック安定時間

サブクロック安定時間は , CKSSR レジスタの SCST[1:0] ビットで選択できます。Table 6-3 に示されているように 4 通りの設定が可能です。

■ PLL クロック安定時間

PLL クロック安定時間は , CKSSR レジスタの PCST ビットで選択できます。Table 6-3 に示されているように 2 通りの設定が可能です。

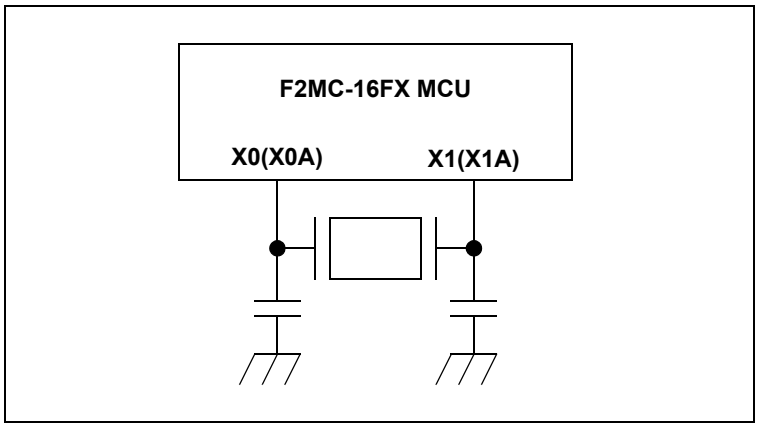
6.6 発振器または外部クロックとマイクロコントローラとの接続

F^{2}MC-16FX マイクロコントローラは, 外部発振器用の最大 2 つのクロック生成回路, メイン発振回路, およびサブ発振回路で構成されます。これらの回路に外部発振器を接続すると, メインクロックとサブクロックが生成されます。また, 外部で生成されたクロックをマイクロコントローラに入力することもできます。

水晶発振器またはセラミック発振器をマイクロコントローラに接続する例

Figure 6-11 の例に示されているように , 水晶発振器またはセラミック発振器を接続します。

Figure 6-11. 水晶発振器またはセラミック発振器をマイクロコントローラに接続する例



高速クロック入力が可能で, デバイスがこの接続を使用する場合 , CILCR: FCI ビットを "0" に設定してください。この機能の利用情報についてはデータシートを参照してください。

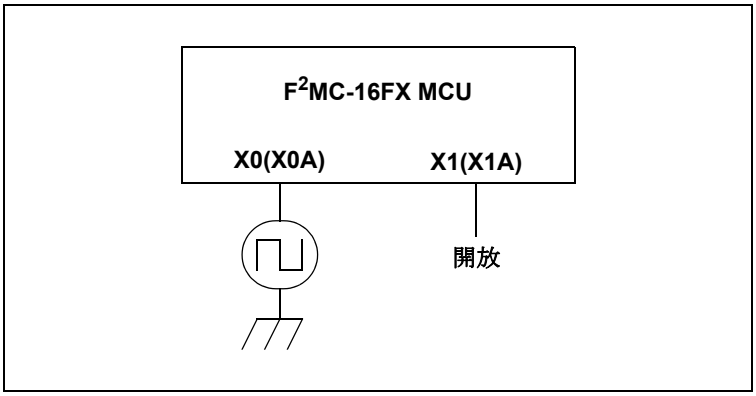
外部クロックをマイクロコントローラに接続する例

Figure 6-12 の例に示されているように , 外部クロックを X0 (X0A) 端子に接続します。X1 (X1A) 端子は必ず開放されていなければなりません。

オプションの高速クロック入力機能が可能なデバイスは , X0 端子が高速クロック周波数として使えます (入力クロック周波数の有効範囲や高速クロック入力機能の利用についてはデータシートを参照してください)。この機能は , デバイスを外部メインクロックに切り換える前に , CILCR:FCI ビットに "1" を設定すると有効になります。

メイン発振を外部クロックに接続する場合 , 弊社は高速クロック入力機能の使用を推奨します。

Figure 6-12. 外部クロックをマイクロコントローラに接続する例



7. クロックモジュレータ

クロックモジュレータの概要と特長について説明します。ここでは , クロックモジュレータのレジスタ構造と動作について説明します。
クロックモジュレータは設定可能な組合せに制限がありますので , ご使用になる場合は営業担当へお問い合わせください。

7.1 概要

クロックモジュレータの概要を示します。

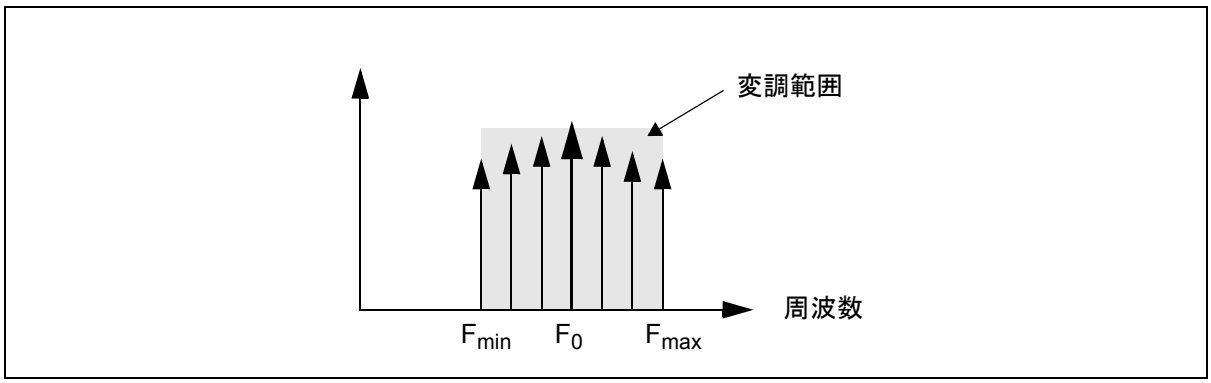
概要

クロックモジュレータはクロック信号のスペクトラムを広い範囲に分散させることで周囲への電磁ノイズの影響 (EMI) を減らします。

このモジュールには , PLL 回路から周波数 F0 の非変調基準クロックが供給されます。この基準クロックが不規則信号で制御され , 周波数変調されます。

変調クロックの平均周波数は基準クロック周波数 F0 に相当します。

Figure 7-1. 変調クロックの周波数帯域(基本周波数のみ)



変調度および周波数分解能

変調クロックの最大周波数と最小周波数 (F_{max} と F_{min}) は , 変調度パラメータで定義します。また , 変調範囲の分解能は低 (1) から高 (7) までの 7 段階で選択可能です。分解能が高くなるにつれて , 変調クロックの帯域における拡散周波数の精度は高くなりますが , 許容変調度は小さくなります。

通常 , 最大許容周波数分解能と最大許容変調度を組み合せると , 最大の EMI 低減効果が得られます。一部のケースでは , 変調度が低いほうが良好な EMI 効果が得られる場合もあります。可能な設定については , Table 7-4 を参照してください。

7.2 レジスタの説明

クロックモジュレータレジスタを一覧表示し, 各レジスタの機能について詳しく説明します。

クロックモジュレータレジスタ

Figure 7-2. クロックモジュレータレジスタ

CMCR:クロックモジュレータ制御レジスタ

アドレス：	bit	7	6	5	4	3	2	1	0
000418_H		予約	予約	予約	予約	予約	MOD-RUN	MODEN	PDX
初期値		-	R/W	R/W	R/W	-	R	R/W	R/W
		X	0	0	1	X	0	0	0

CMPRL：クロック変調パラメータレジスタ (下位)

アドレス：	bit	7	6	5	4	3	2	1	0
$00041A_H$		MP7	MP6	MP5	MP4	MP3	MP2	MP1	MP0
初期値		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		1	1	1	1	1	1	0	1

CMPRH：クロック変調パラメータレジスタ (上位)

アドレス：	bit	15	14	13	12	11	10	9	8
$00041B_H$		予約	予約	MP13	MP12	MP11	MP10	MP9	MP8
初期値		-	-	R/W	R/W	R/W	R/W	R/W	R/W
		X	X	0	0	0	0	1	0

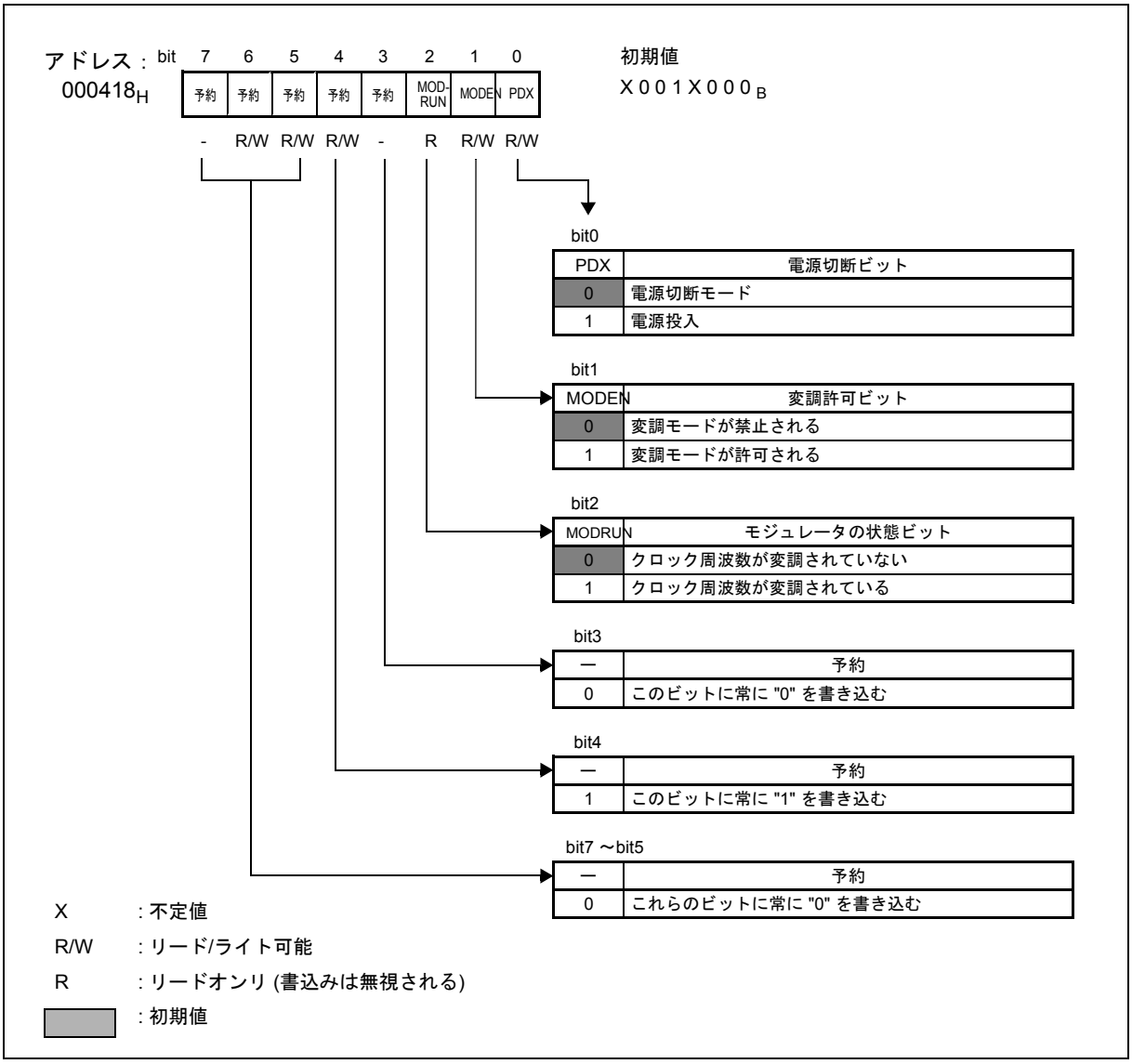
7.2.1 クロックモジュレータ制御レジスタ(CMCR)

制御レジスタ (CMCR) には以下の機能があります。

- モジュレータを電源切断モードに設定する
- モジュレータを許可/禁止する
- モジュレータの状態を示す

クロックモジュレータ制御レジスタ(CMCR)

Figure 7-3. クロックモジュレータ制御レジスタ(CMCR)の構成



クロックモジュレータ制御レジスタの内容

Table 7-1. クロックモジュレータ制御レジスタの各ビットの機能 (Sheet 1 of 2)

ビット名		機能
bit 0	PDX: 電源切断 ビット	"0": 電源切断モード "1": 電源投入 • PDX はモジュレータ用の電源切断信号です。変調を許可する前に , 必ずこのビットを "1" に設定し, 6 µs のスタートアップ時間が経過するまで待機しなければなりません。推奨するスタートアップシーケンスについては「7.3 アプリケーションノート」を参照してください。 • 電源切断モードに切り換える (PDX=0) 前に,モジュレータを禁止しておく必要があります (MOD=0 およびMODRUN=0)。
bit 1	MODEN: 変調許可 ビット	"0": 変調を禁止します。 "1": 変調を許可します。 • 変調を許可するには , MODEN を "1 " に設定する必要があります。 • 変調を許可する前に , 必ず PLL から安定した基準クロックを供給しなければなりません (PLL ロック時間経過後に設定する必要があります)。 • 変調用に指定された PLL 周波数範囲は 16 MHz ～ 64 MHz です。 • PLL 出力周波数ごとに一連の許容変調パラメータが用意されています。選択された設定 (CMPR レジスタ) と PLL 周波数が一致している必要があります。 CMPR レジスタの説明を参照してください。 • PLL 出力周波数を変更する場合や PLL を (例えば , 電源切断モードで)OFF する場合は , 必ず事前にモジュレータを禁止する必要があります (MODEN=0 および MODRUN=0)。 • 変調を許可する前に , 必ず PDX を "1" に設定してモジュレータを電源切断モードからアクティブモードに切り換え , 6 µs のスタートアップ時間が経過するまで待機しなければなりません。 推奨するスタートアップシーケンスについては「7.3 アプリケーションノート」を参照してください。 • 変調を許可する前に , クロック変調パラメータレジスタ CMPR を使用して適切な設定にしておく必要があります。 • MODEN を "1" に設定して変調を許可すると , モジュレータが調整されます。この期間中は , クロックが変調されません。このため , 出力クロックが変調クロックにすぐに切り換わりません。クロックの状態 (周波数変調 / 非変調) は MODRUN 状態ビットで示されます。MODRUN ビットの説明を参照してください。 • MODEN 信号の同期と非変調クロックへの同期切換えにより , 変調が禁止されてからクロックが非変調クロックに切り換わるまでに要する時間は 9×T0 (入力クロック期間) 未満になります。変調は随時禁止することができます。 • クロック変調パラメータレジスタ (CMPR) を変更する前に, モジュレータを禁止しておく必要があります(MODEN=0 および MODRUN=0)。
bit 2	MODRUN: モジュレータ 状態ビット	"0": MCU が非変調クロックで動作しています。 "1": MCU が変調クロックで動作しています。 • MODRUN は , モジュレータ出力クロックの状態を示します。出力クロックが変調されている場合は MODRUN が "1" に設定され , それ以外の場合は MODRUN が "0" に設定されます。 • MODEN を "1" に設定して変調モードを許可すると , モジュレータが調整されます。この期間中は , クロックが変調されません。このため , 出力クロックが変調クロックに切り換わり , MODRUN ビットが "1" に設定されるまで数 µs かかります。調整時間はメインクロックタイマで測定され , 256 ～ 512 のメインクロック周期 (4 MHz メインクロック周波数で 64 ～ 128 µs) が必要となります。調整中はメインクロックタイマをクリアしないでください。 • MODEN 信号の同期と非変調クロックへの同期切換えにより , 変調が禁止されてから MODRUN が "0" に変わり , クロックが非変調クロックに切り換わるまでに要する時間は 9×T0 (入力クロック期間) 未満になります。 • MODRUN ビットは読出し専用です。MODRUN への書込みは影響がありません。 • パラメータレジスタ CMPR を変更する前に必ずモジュレータを禁止しておく必要があります (MODEN=0およびMODRUN=0)。

Table 7-1. クロックモジュレータ制御レジスタの各ビットの機能 (Sheet 2 of 2)

ビット名		機能
bit 3	予約	• このビットに常に "0" を書き込みます。 • このビットの読出し値は未定義です。 • このレジスタへのリードモディファイライト系命令は無効です。
bit 4	予約	• このビットに常に "1" を書き込みます。 • このビットを読み出すと, 書込み値が戻されます。
bit 5, bit 6	未定義	• これらのビットに常に "0" を書き込みます。 • これらのビットを読み出すと, 書込み値が戻されます。
bit 7	予約	• このビットに常に "0" を書き込みます。 • このビットの読出し値は未定義です。 • このレジスタへのリードモディファイライト系命令は無効です。

モジュレータの状態を以下の表にまとめてあります。

Table 7-2. モジュレータの状態

	MODEN	PDX	MODRUN (リードオンリ)
モジュレータ禁止	0	0	0
モジュレータ電源投入 モジュレータスタートアップ時間待機中(> 6 µs)	0	1	0
モジュレータ許可, モジュレータの調整中, 変調が動作中でない	1	1	0
変調が動作中	1	1	1
	上記以外設定不可		

7.2.2 クロック変調パラメータレジスタ(CMPR)

クロック変調パラメータレジスタ (CMPR) によって変調度が決まります。

クロック変調パラメータレジスタ

Figure 7-4. クロック変調パラメータレジスタ

CMPRL：クロック変調パラメータレジスタ (下位)

アドレス：	bit	7	6	5	4	3	2	1	0
00041A$_H$		MP7	MP6	MP5	MB4	MP3	MP2	MP1	MP0
初期値		R/W 1	R/W 1	R/W 1	R/W 1	R/W 1	R/W 1	R/W 0	R/W 1

CMPRH：クロック変調パラメータレジスタ (上位)

アドレス：	bit	15	14	13	12	11	10	9	8
00041B$_H$		予約	予約	MB13	MP12	MP11	MP10	MP9	MP8
初期値		- X	- X	R/W 0	R/W 0	R/W 0	R/W 0	R/W 1	R/W 0

X : 不定値
R/W : リード/ライト可能

- 変調パラメータによって, 変調度と変調クロックで発生する最大および最小周波数が決まります。最適な設定を選択する手段については,「7.3 アプリケーションノート」を参照してください。
- 一組の許容変調パラメータはそれぞれ, 特定の PLL 周波数を表します。PLL 周波数と選択されたパラメータは必ず一致している必要があります。可能な設定については以下の表を参照してください。

＜注意事項＞

変調パラメータの変更は , モジュレータが禁止されており , かつ RUN フラグが "0" に設定されている (MODEN=0, MODRUN=0) 場合にのみ行ってください。

クロック変調パラメータレジスタの内容

Table 7-3. クロック変調パラメータレジスタ(CMPR)の各ビットの機能

ビット名		機能
bit 15, 14	未定義	
bit 13 ～ bit 0	MP13 ～ MP0: 変調パラメータビット	PLL周波数に合わせて以下の変調パラメータ設定を行うことができます。対応するCMPRレジスタ値が右端の列に示されます。

F0: 非変調入力クロックの周波数 (PLL 周波数)

T0: 非変調入力クロックの期間 (PLL クロック期間)

分解能 : 変調クロックにおける周波数の分解能 (低 (1) ～ 高 (7))

F_{min}: 変調クロックで発生する最小周波数

F_{max}: 変調クロックで発生する最大周波数

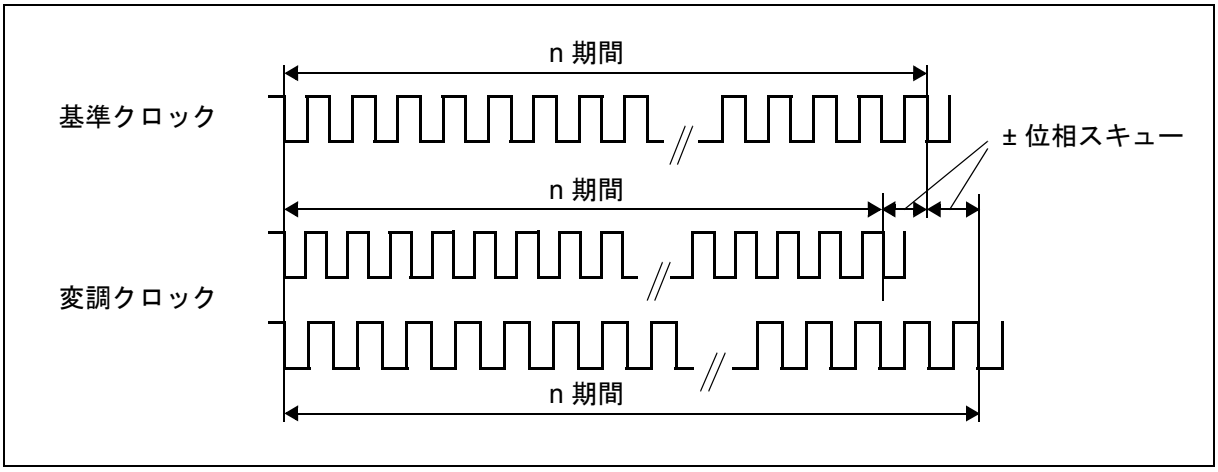
位相スキュー : 非変調クロックのクロック期間での非変調基準クロックに対する変調クロックの最大位相シフト

例 : 位相スキュー = 1.44
ワーストケースでは , 変調クロックの一連の n 期間は , 非変調基準クロックの一連の n 期間より 1.44×T0 だけ短くなるか 1.44×T0 だけ長くなる可能性があります。

n は , > 50 期間の任意の数字になります。

位相スキュー 50: n <= 50 期間のシーケンスを表す位相スキュー

CMPR: CMPR レジスタのレジスタ設定



MCU クロックと変調パラメータの推奨設定値を以下の表に示します。

Table 7-4. 変調パラメータの推奨設定

F0 (MHz)	分解能	変調度	F_{min} (MHz)	F_{max} (MHz)	+/- 位相スキュー 50 [期間]	+/- 位相スキュー 最小 / 最大 [期間]	CMPR
16	3	1	13.47	19.69	1.5625	3.7187	026F
16	11	1	12.19	23.72	2.25	7.875	036B
16	3	2	14.22	18.29	1.0625	2.875	046E
16	5	2	12.8	21.3	1.5625	4.0625	04AC
16	7	2	11.63	25.6	3.125	7.4375	04EA
16	3	3	13.47	19.69	1.5937	4.3125	066D
20	3	1	18.82	21.33	0.5312	1.4375	026F
20	3	2	17.77	22.85	1.0625	2.875	046E
20	5	2	16	26.67	1.5625	4.0625	04AC
24	3	1	22.59	25.6	0.53125	1.4375	026F
24	5	1	21.33	27.43	0.78125	2.03125	02AE
24	3	2	21.33	27.43	1.0625	2.875	046E
28	3	1	26.35	29.87	0.53125	1.4375	026F
28	5	1	24.89	32	0.78125	2.03125	02AE
28	5	2	22.4	37.33	1.5625	4.0625	04AC
28	3	3	23.58	34.46	1.59375	4.3125	066D
28	5	3	20.36	44.8	2.34375	6.09375	06AA
32	3	2	28.44	36.57	1.0625	2.875	046E
32	3	3	26.95	39.38	1.59375	4.3125	066D
32	5	3	23.27	51.2	2.34375	6.09375	06AA
48	3	1	45.18	51.2	0.53125	1.4375	026F
52	3	1	48.94	55.47	0.5312	1.4375	026F

7.3 アプリケーションノート

クロックモジュレータのスタートアップ/ストップシーケンスと変調パラメータ設定について説明します。

推奨スタートアップ / ストップシーケンス

スタート	1.	モジュレータを電源切断モードから電源投入モードに切り換えます。PDX=1 を設定します。
	2.	PLL をオンにします。
	3.	PLL ロック時間が経過するまで待機します (CKMR: PCM ビットを参照してください)。
	4.	CMPR レジスタを適切な値に設定します。
	5.	MODEN=1 を設定してクロック変調を許可します。 調整が終了すると , 非変調クロックから変調クロックに切り換わり , MODRUN フラグが "1" に変わります。
		... 実行中
ストップ	6.	MODEN=0 を設定してモジュレータを禁止します。
	7.	MODRUN が "0" に変わるまで待機します。
	8.	PDX=0 を設定して電源切断モードに切り換えます。
	9.	PLL を禁止し , 別の動作モード (スタンバイモードなど) に切り換えます。

＜注意事項＞

PLL ロック時間が経過するまでモジュレータを許可しないでください。また , モジュレータの実行中は , PLL を禁止しないでください。

変調パラメータ

特定の EMI 低減を実現するために特定の変調パラメータ設定を推奨することはできません。最適な設定は , 実際のアプリケーション , システム全体 , および条件に大きく依存します。

最適な変調パラメータ設定を確認するには, 以下の手段を推奨します。

1.	性能条件に基づいて必要な PLL 周波数を定義します。	例：16 MHz
2.	MCU の最大許容クロック周波数を決定します。	例：32 MHz
3.	最大周波数が MCU の最大許容クロック周波数を下回っている最大分解能と最大変調度の設定を選択します。	例：分解能 : 7, 変調度 : 2, CMPR = $5F2_H$ (F_{max} = 30.34 MHz)
4.	EMI 測定を実行します。	
5.	EMI 測定が条件を満たしていない場合は , 以下のいずれかを行うことができます。	
	同じ周波数分解能で変調度を小さくする (これにより , > 100 MHz の上位周波数帯域の低減度は向上しますが , < 100 MHz の基本周波数の低減度は低下する可能性がある)。	例：分解能 : 7, 変調度 : 1, CMPR = $3F9_H$
	または 低い周波数分解能で変調度を大きくする (これにより , < 100 MHz の基本周波数の低減度は向上しますが , > 100 MHz の上位周波数帯域の低減度は低下する可能性がある)。	または 例：分解能 : 5, 変調度 : 3, CMPR = 0771_H
6.	新しい設定で項目 3. をやり直し , 最適な設定が見つかるまで繰り返します。	

＜注意事項＞

- すべてのデバイスに対してすべての設定が許可されているわけではありません。
  MCU の実際の最大許容クロック周波数を考慮してください (データシートを参照してください)。
- フラッシュデバイスの設定制限。
  クロックモジュレータを使用する際 , フラッシュデバイスの取り扱いには注意が必要です。フラッシュメモリインタフェースのウェイトサイクル挿入数には限りがあるため , クロックモジュレータパラメータの割り当てを設定することはできません。最大許容周波数はフラッシュアクセススピードに依存します。詳しくは , 「第 33 章 フラッシュメモリ」を参照してください。

8. リセットとスタートアップ

F²MC-16FX ファミリのマイクロコントローラのリセットとスタートアップについて説明します。

8.1 リセット

リセットが発生すると , CPU は現在実行中のプロセスを即時停止し , リセットが解除されるまで待機します。この後 , CPU はブート ROM プログラムの実行を開始します。ブート ROM プログラムは , モード端子の設定に応じて様々な動作モードに分岐し , ユーザプログラムの実行を開始します。
リセット要因には以下の 5 種類があります。

- 電源リセット (電源投入または低電圧)
- RSTX端子経由の外部リセット要求
- クロック停止検出
- ソフトウェアリセット要求
- ウォッチドッグタイマのオーバフロー

8.1.1 リセット要因

リセット要因を Table 8-1 に示します。

Table 8-1. リセット要因

リセットの種類	要因	リセット解除後のシステムクロック (CLKS1 および CLKS2)	モード端子設定の読出し	RAMの内容
電源	電源が投入された場合または Vcc が低電圧検出器レベル(CILCRレジスタのLVL)を下回った場合。データシートのアナログ値を参照してください。	CR クロック CLKRC	あり	保証されない
外部端子	RSTX端子への L レベル入力			維持される*
クロック停止検出	外部発振器の故障		なし	保証されない
ソフトウェア	リセット構成レジスタ (RCR) の SRSTG ビットに"1"が書き込まれる			維持される
ウォッチドッグタイマ	ウォッチドッグタイマのオーバフロー			

＜注意事項＞

* アドレス $7FFC_H$ ～ $7FFF_H$ のバイトを除く。
それぞれのリセット後 , 汎用レジスタ (GPR) のバンク 00, 01 内 , ならびに上記 4 つの RAM バイトのレジスタの内容は未定義になります。それ以外の汎用レジスタ (GPR) の内容は維持されるか , または未定義になります（上記のリセット要因によって異なる）。

電源リセット

電源リセットが発生するのは , データシートに記載された電源投入立上り時間に電源が投入された場合 (パワーオンリセット) または電源 Vcc が CILCR レジスタの LVL ビットに記載された特定の値を下回っていることを低電圧検出器が検出した場合 (低電圧リセット) です。データシートの低電圧検出の特性を参照してください。電源リセット延長回路によって電源リセットが内部で延期される時間は , 最低 700 CR クロックです。

電源リセットが発生した後 , 内部 RAM の内容と初期値が "X" のすべてのレジスタの内容は不定になります。

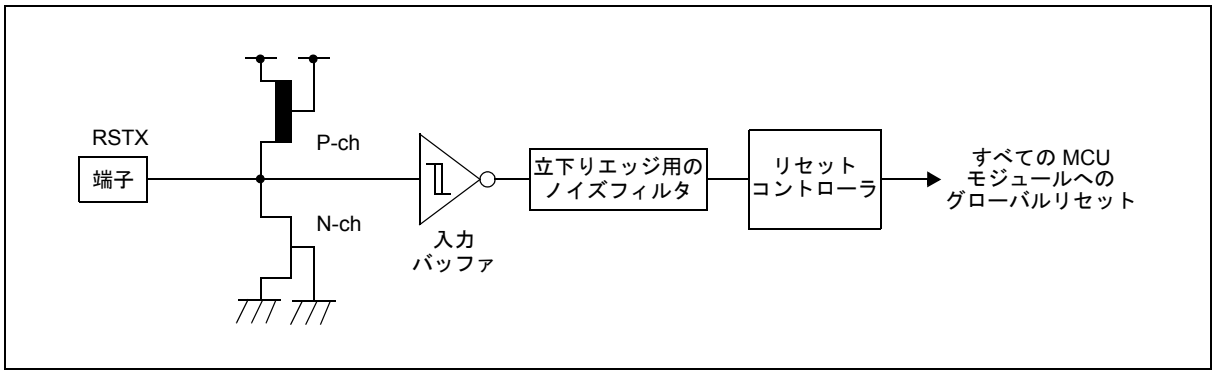
外部端子リセット

外部端子リセットは , 外部リセット端子 (RSTX 端子) への L レベル入力によって発生します。外部リセット延長回路によってこの信号が内部で延期される時間は , 最低 700 CR クロックです。

外部リセットは非同期であるため , デバイスの同期がとられているかどうかに関係なく , アサート後にほとんど遅延なく (10 ns 程度) 有効になります。

内部 RAM の内容とリセットされていないレジスタ (初期値 "X") の内容はアドレス$7FFFC_H$～$7FFF_H$の4バイトを除き維持されます。

Figure 8-1. 外部リセット端子のブロックダイヤグラム



クロック停止検出リセット

クロック停止検出リセットは , 現在の動作モードとリセット構成レジスタ (RCR) の設定に応じてメインクロック発振器またはサブクロック発振器のどちらかが故障した場合に発生します。詳細は ,「8.6 クロック停止検出機能およびリセットの動作」を参照してください。

クロック停止検出リセットが発生した後 , 内部 RAM の内容とリセットされていないレジスタ (初期値 "X") の内容は保証できません。

ソフトウェアリセット

ソフトウェアリセットは , リセット構成レジスタの (RCR) の SRSTG ビットに "1" を書き込むことによって生成される内部リセットです。

内部 RAM の内容とリセットされていないレジスタ (初期値 "X") の内容は維持されます。

ウォッチドッグタイマリセット

ウォッチドッグタイマリセットが発生するのは , ウォッチドッグタイマが起動した後 , 決められた期間内にクリアされなかった場合です (「第 11 章 ウォッチドッグタイマとウォッチドッグリセット」を参照してください)。

内部 RAM の内容とリセットされていないレジスタ (初期値 "X") の内容は維持されます。

＜注意事項＞

リセット時の端子の状態 : すべての端子は , Hi-Z 状態になります。Table 12-11 on page 322 を参照してください。

8.2 リセット，システムクロック，安定待ち時間

F²MC-16FXファミリには5種類のリセット要因があります。システムクロック(CLKS1とCLKS2)は,リセットが発生するごとにCRクロックに設定されます。安定待ち時間は,リセット要因とリセット発生時のCR発振器の状態によって異なります。

8.2.1 リセット要因と安定待ち時間

それぞれのリセット要因に適用される安定時間を Table 8-2 にまとめてあります。

Table 8-2. リセット要因と安定待ち時間

リセット要因	安定待ち時間
電源リセット	700 + 64 = 764 CR クロック (公称 2MHz CR クロック周波数で約 380µs)。この待ち時間は, 有効な電源 (外部電圧および内部電圧) に到達した後に始まります。700 CR クロックは電源リセット延長時間になり, 64 周期は CR クロック安定時間になります。
外部リセット	700+64=764 CR クロック (公称 2MHz CR クロック周波数で約 380µs)。700 CR クロックは RSTX 入力信号の立下りエッジで既に始まっている外部リセット延長時間であり, 64 周期は RSTX解除後に始まる CR クロック安定時間です。
クロック停止検出リセット	64 CR クロックの CR クロック安定時間が適用されます。
ソフトウェアリセット	
ウォッチドッグタイマリセット	

リセットが発生するごとに CR 発振器, メイン発振器, およびサブ発振器が起動し, PLL が停止し, クロック安定化選択レジスタ (CKSSR) がリセットされます。

電源リセットまたは外部リセットが発生すると , クロック準備完了モニタビットがすべてクリアされます。

それ以外のリセットが発生した場合は , PLL および CR クロック準備完了モニタビットだけがクリアされ , メインおよびサブクロック準備完了モニタビットはクリアされません。動作中のクロックはそのまま動作を続け , 非動作中のクロックは起動してから , CKSSR レジスタに定義された安定時間が経過した後に準備完了状態になります。

選択されたクロックと発振安定待ち時間の詳細は , 「第 6 章 クロック」を参照してください。

8.2.2 外部リセット発生時の安定待ち時間

外部リセット発生時の安定待ち時間は , 外部リセット延長時間と CR クロック安定時間の 2 部で構成されます。外部リセット延長時間は RSTX (立下りエッジ) のアサートからカウントが開始され , CR クロック安定時間は RSTX の解除からカウントが開始されます。このため , プログラム実行の開始は , 以下に示すように RSTX のアサート時間によって異なります。

700超過のCRクロックでアサートされた外部リセット

この場合は, 外部リセット延長時間が既に満了しています。このため, CPU は外部リセットが解除されてから 64 CR クロックの CR クロック安定時間が経過した後にブート ROM プログラムの実行を開始します。

700未満のCRクロックでアサートされた外部リセット

この場合は , 外部リセット延長時間がまだ満了していません。このため , CPU でのブート ROM プログラムの実行は , 外部リセット延長時間と 64 CR クロックの CR クロック安定時間が満了するまで延期されます。

8.2.3 メイン発振器の安定

メイン発振安定待ち時間セレクタの初期値は 2^{18}/ メインクロック (MCST[2:0] = 111_B) に設定されます。この値は, ユーザプログラム実行の開始直後に上書きできます (MCU が CR クロックモードでまだ動作している場合)。使用している発振器に適した値を選択してください。

メイン発振安定待ち時間の測定に使用されるのは, 電源リセットおよび外部リセット延長時間が満了した時点でカウントを開始するメインクロックカウンタです。このため , 700 超過の CR クロックで RSTX がアサートされる場合は , RSTX がまだアサート中であっても既にメイン発振安定待ち時間に入っています。

これによって , RSTX でメイン発振安定待ち時間を制御することが可能になります。メイン発振器が安定した後に RSTX を解除し , MCST[2:0] を既に満了している値に設定します。ただし , (MCE ビットに書き込むか , ストップモードに入ることによって) メイン発振器を非動作にする前に , MCST[2:0] ビットに適切な値を書き込む必要があります。この理由として , この値はメイン発振器の再起動後に使用されるからです。

8.2.4 サブ発振器の安定

サブ発振安定待ち時間セレクタの初期値は 2^{16}/サブクロック (SCST[1:0] =11) に設定されます。この値は, ユーザプログラム実行の開始直後に上書きできます (MCU が CR クロックモードでまだ動作している場合)。使用している発振器に適した値を選択してください。

サブ発振安定待ち時間の測定に使用されるのは , RSTX がクリアされ , かつ電源リセットおよび外部リセット延長時間が満了した時点でカウントを開始するサブクロックカウンタです。

8.3 電源リセットおよび外部リセット後のスタートアップ

電源リセットまたは外部リセットイベントが発生すると, MCU は電源と CR 発振器が安定するまで待機します。この後, MCU は CR クロックをクロックソースとしてブート ROM プログラムの実行を開始します (CR ランモード)。
別のクロックモード (メインクロック, PLL クロック, またはサブクロック) に移行できるのは, それぞれのクロックが安定した後です。 外部発振器の安定時間は, 接続されている発振器の特性に合わせて調整することができます。

8.3.1 リセット延長

電源リセットおよび外部リセットイベントは , プログラムの実行が開始されるまで , 電圧調整器と電圧比較器の安定を保証するためにリセット延長回路によって延長されます。

電源リセット延長

電源リセット延長カウンタは , 電源投入検出器と低電圧リセットで初期化され , かつ CR クロック CLKRC によって同期がとられる 10 ビットのカウンタです。

このカウンタは , 最後の電源投入または低電圧イベントが解除された後 , 700 CR クロックの間だけ内部電源リセット信号 PRST をアクティブ状態のまま維持します。PRST が非アクティブになるのは, 少なくとも 700 CR クロック (公称 2MHz CR クロック周波数で約 350µs) の間に電源が安定していた場合です。

外部リセット延長

外部リセット延長カウンタは, RSTX入力端子の立下りエッジで初期化され, かつ CR クロック CLKRC によって同期がとられる 10 ビットのカウンタです。また, 電源リセットが発生すると , デバイスが電源リセットでのみ起動される場合でも , 決められたスタートアップ時間を実現するためにこのカウンタが初期化されます。

このカウンタは RSTX入力端子の最後の立下りエッジ (公称 2MHz CR クロック周波数で 約 350 µs) の後, 700 CR クロックの間だけ信号 ERST をアクティブ状態のまま維持します。

700 超過の CR クロックで RSTX がアサートされた場合 , この機能は働きません。

リセット延長回路のブロックダイヤグラムを Figure 8-2 に示し , 外部リセット延長回路の機能を Figure 8-3 に示します。

Figure 8-2. リセット延長回路のブロックダイヤグラム

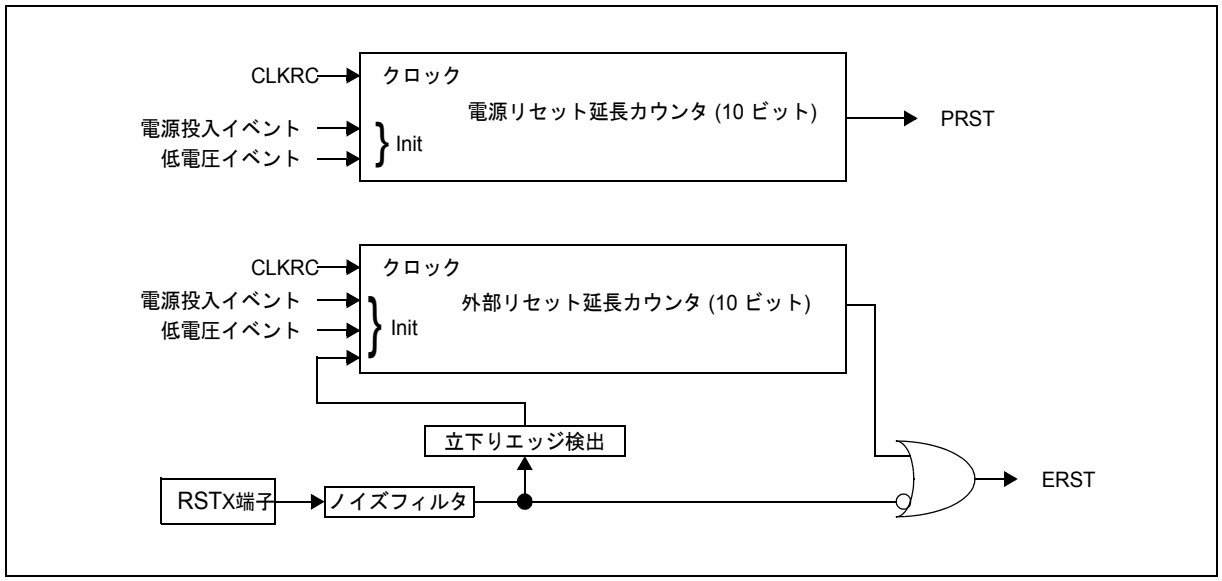
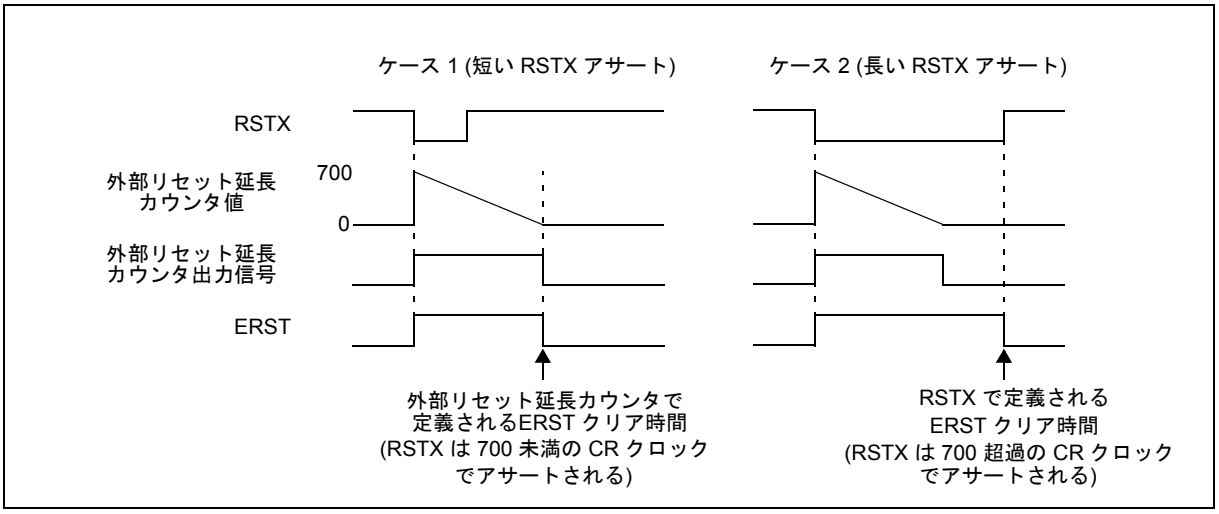


Figure 8-3. 外部リセット延長回路の機能



8.3.2 ソースクロックタイマとクロック準備完了モニタビット

ソースクロックタイマ (CR クロックタイマ , メインクロックタイマ , およびサブクロックタイマ) とクロックモニタビット (RCM, MCM, PCM, および SCM) はすべて , 3 つの発振器が許可されている場合に電源リセットまたは外部リセットイベントでクリアされます。

CRクロックタイマ, サブクロックタイマ, RCMビットおよびSCMビット

CR クロックタイマとサブクロックタイマは PRST または ERST でクリアおよび停止されます。PRST と ERST が両方とも解除された時点でカウントが始まります。このため , ERST または PRST がアクティブである限り , CR クロック準備完了モニタビット RCM とサブクロック準備完了モニタビット SCM を設定してはなりません。

メインクロックタイマとMCMビット

メインクロックタイマのクリアと停止は , 外部リセットまたは電源リセット延長時間の間にのみ行われます。このため , メインクロックタイマは , RSTX 立下りエッジの後に 700 CR クロックで起動するか , または電源投入 / 低電圧イベントが解除された後に 700 CR クロックで起動します。したがって , RSTX がまだアサート中の場合でもメイン発振安定待ち時間に入ります。

PCMビット

PLL は常に禁止されており , PLL クロック準備完了モニタビット PCM はリセットでクリアされます。ユーザプログラムが起動した後にソフトウェアで PLL を許可する必要があります。

8.3.3 電源リセットまたは外部リセットが発生した後でのプログラム実行の開始

CR クロックタイマは , PRST と ERST が非アクティブ状態になった時点でカウントを開始します。64 CR クロック (公称 2MHz CR クロック周波数で約 30µs) が経過すると , CR クロック準備完了モニタビット RCM が設定され, CPU 用の内部リセット信号が解除されます。リセットが解除された後, CPU は必ず, CR クロック周波数が公称 2MHz に設定された CR クロックモードに入り , ブート ROM プログラムの実行を開始します。

低電圧検出レベル (CILCR レジスタの LVL ビット) は可能な限り低い値に初期化されます。より高い検出値を必要とするなら , それに従って , レジスタ値を変えることができます。

詳細は, 「8.2 リセット, システムクロック, および安定待ち時間」を参照してください。

8.3.4 メインクロックモードへの移行

メインクロックモードに移行するには , CKSR レジスタのシステムクロック選択ビットへの書込みを行います。

ただし , メインクロックモニタビット MCM ("1" に設定されている) に示されたメインクロックが安定するまで移行が延期されます。

メイン発振器の安定時間は , 「8.2 リセット , システムクロック , および安定待ち時間」に記載されているように , CKSSR: MCST[2:0] ビットで定義します。

＜注意事項＞

いくつかのデバイスでオプションの高速クロック入力機能を使用する際 , メインクロックモードに切り換わる前に , CILCR レジスタの FCI ビットを "1" に設定してください。
発振端子の入力特性についてデータシートを参照してください。

8.3.5 PLLクロックモードへの移行

PLL はソフトウェアで構成および許可する必要があります。PLL が起動した後およびメインクロック (MCM=1) が安定した後で PLL 安定待ち時間に入ります。

PLL クロックモードに移行できるのは , PLL クロック準備完了モニタビット PCM が設定されている場合です。

CR クロックモードから PLL クロックモードに直接切り換えることもメインクロックモードを経由して切り換えることも可能です。

8.3.6 サブクロックモードへの移行

リセットごとにサブ発振器が許可され , PRST と ERST が解除された後にサブクロック安定時間が始まります。

サブ発振器の安定時間は , サブクロック準備完了モニタビット SCM が設定される時間を定義する CKSSR: SCST[1:0] によって決まります。サブクロックモードへの移行は , SCM が "1" に設定されるまで延期されます。

8.4 ブートROMプログラムの実行と動作モード, およびROM構成ブロック

リセット信号が解除されると,MCUは内部のブートROMプログラムの実行を開始します。ブートROMプログラムは, MCUの動作モードを定義するモード端子(MD2～MD0)の状態を読み出します。モード端子の設定に従って, MCUはパラレルフラッシュ書込みモード, シリアル通信モードの起動, あるいはユーザプログラムの実行を開始します。
MCU 動作モードの起動後, フラッシュ/ROM内のいわゆるROM構成ブロックに保存された構成によって, ブートROMプログラムの今後の実行が定義されます。またこれによって, ブートベクタ (ユーザプログラムの開始アドレス) も決定されます。
ブートROMプログラムはこのブートベクタからユーザプログラムの実行を開始します。

8.4.1 外部バス

初期設定では , ブートベクタ (ユーザプログラムの開始アドレス) は固定アドレス (FF: $FFDC_H$) から読み出されます。

外部バスインタフェースを搭載したデバイスでは , 外部メモリからブートベクタの読み出しが可能となります。モード端子の設定に従って, ブートベクタの読出し時に3つの異なった外部バス構成を選択することができます。

ブート ROM プログラムは , ブートベクタとモードバイトを固定アドレスから読み出します。バス構成は提供されたモードバイトに応じて変更され , ユーザプログラムはその提供アドレス (ブートベクタ) から開始されます。

8.4.2 モード端子

モード端子 MD2 ～ MD0 は MCU の動作モードを定義します。モード端子の設定は内部的にサンプリングされ , 電源リセットまたは外部リセットによってのみ変更可能です (Figure 8-2 で定義された信号 PRST と ERST が両方とも停止された時に , モード端子のデータがサンプリングされます)。

したがって , 動作モードはクロック停止検出 , ソフトウェアあるいはウォッチドッグタイマリセットでは変更できません。

サンプリングされたモード端子信号はリセット解除後にブート ROM プログラムで読み出されますが , これによって以下の表に従って動作モードが決定されます。

Table 8-3. モード端子の設定

モード端子の設定			動作モード	ブートベクタのソース	ブートベクタ読出し時の外部バス構成	備考
MD2	MD1	MD0				
0	0	0	外部ベクタモード 0	外部メモリ	8 ビット, アドレス/データマルチプレックス	外部バスインタフェース搭載のデバイスのみ使用可能
0	0	1	外部ベクタモード 1	外部メモリ	16 ビット, アドレス/データマルチプレックス	外部バスインタフェース搭載のデバイスのみ使用可能
0	1	0	シリアル通信モード	-	-	シリアルフラッシュ書込みに使用
0	1	1	内部ベクタモード	内部メモリまたは固定値	-	すべてのデバイスに使用可能
1	0	0	予約 (テストモード)	-	-	
1	0	1	予約 (テストモード)	-	-	
1	1	0	外部ベクタモード 2	外部メモリ	8 ビット, アドレス/データノンマルチプレックス	外部バスインタフェース搭載のデバイスのみ使用可能
1	1	1	パラレルフラッシュ書込みモード	-	-	フラッシュデバイスのみ使用可能

8.4.3 動作モード

■ **外部ベクタモード (MD2～MD0=000, 001または 110)**

外部ベクタモードとは , 外部バスインタフェース経由で FF:FFDC$_H$ にある外部メモリからブートベクタとモードバイトの読出しが行われる CPU 実行モードです。本モードは外部バスインタフェース搭載のデバイスでのみ利用可能です。

ROM 構成ブロック内に BDM 起動マーカが設定されている場合 , 必要に応じて内蔵モニタデバッガカーネルを初期化することができます。なお , 読出しセキュリティ機能が許可されている場合 , 外部ベクタモードで内部 ROM や内蔵モニタデバッガカーネルを使用することはできません。

詳細については ,「第 12 章 外部バスインタフェース」を参照してください。

■ **内部ベクタモード (MD2～MD0=011)**

内部ベクタモードとは , FF:FFDC$_H$ にある内部メモリからブートベクタ (ユーザプログラムの開始アドレス) の読み出しが行われる CPU 実行モードです。

固定ベクタ起動マーカが ROM 構成ブロックに設定されている場合 , 必要に応じて固定ブートベクタが選択できます。固定ブートベクタによって , ブートローダアプリケーションを導入することが可能になります。

主要リセット終了からユーザプログラム開始までに , ブート ROM は一時的に , シリアル通信モード * にあるような外部通信要求に対して専用 UART チャネルをスキャンします。ROM 構成ブロック内に UART スキャン停止マーカが設定されている場合 , 必要に応じてこのスキャン処理を禁止することができます。

ROM 構成ブロック内に BDM 起動マーカが設定されている場合 , 必要に応じて内蔵モニタデバッガカーネルを初期化することができます。
*: ただし , すべての製品で対応しておりません。

内部ベクタモードはすべての MCU タイプで利用可能であり , 固定ブートベクタは
DF: 0080$_H$ で提供されます。

■ **シリアル通信モード (MD2～MD0=010)**

本モードによって , シリアル通信リード / ライトコマンドにより任意のメモリアドレスの書込みと読出しが可能となります。

内部フラッシュのプログラミングは RAM の小さなプログラムを実行することで可能になります。

専用リード / ライトコマンドにかかわらず , 内部 RAM エリア 7C00$_H$ ～ 7FFF$_H$ はほかのコマンドによっても変わることがあります (例 : CRC チェック)。

ROM/ フラッシュのセキュリティ機能が許可されている場合 , 正しいセキュリティキーの送信後に書込みや読出しができます。

詳細については ,「第 33 章 フラッシュメモリ」と「第 35 章 ROM/ フラッシュセキュリティ機能」を参照してください。

■ **パラレルフラッシュ書込みモード (MD2～MD0=111)**

本モードによって , パラレルフラッシュプログラマ上での内部フラッシュの書込みが可能となります。本モードはフラッシュデバイスでのみ利用できます。

詳細については ,「第 33 章 フラッシュメモリ」を参照してください。

■ **テストモード (MD2～MD0=100 または 101)**

本モードはテスト機能用に予約されています。

8.4.4 ROM構成ブロック(RCB)

ROM 構成ブロックは , フラッシュ A/ROM 内の固定領域であり , フラッシュ B, データフラッシュ A, およびデータフラッシュ B が存在していればユーザにブート ROM の動作が構成できるようにします。指定された構成データ以外は , ユーザプログラムでプログラム用に割り当ててはいけません。

ROM 構成ブロック A (RCBA) は , アドレス DF:0000$_H$ ～ DF:007F$_H$ (フラッシュ A または ROM の開始アドレス) に置かれています。ROM 構成ブロック B(RCBB) は , アドレス DE:0000$_H$ ～ DE:002F$_H$ (存在する場合はフラッシュ B の開始アドレス) に置かれています。データフラッシュ構成ブロック A(DFCBA) のためのデータフラッシュ A の ROM 構成ブロック はアドレス 0E:FF00$_H$ ～ 0E:FF2F$_H$ (存在する場合) に置かれています。データフラッシュ構成ブロック B(DFCBB) のためのデータフラッシュ B の ROM 構成ブロックは , アドレス 0E:FE00$_H$ ～ 0E:FE2F$_H$ (存在する場合) に置かれています。ブート ROM によって読み出され , 処理されます。どんな方法でもフラッシュメモリの書込みや消去の内容を設定し , 変更することができます。

RCBA, RCBB, DFCBA, および DFCBB は帰属するフラッシュ /ROM またはデータフラッシュのセキュリティ設定を構成します。さらに , RCBA はブートシーケンスとバックグラウンドデバッグモードも構成します。

Table 8-4. ROM構成ブロックA (RCBA)の構造

オフセット	サブブロック	マーカ	備考
00$_H$	セキュリティ保護構成ブロック	MSBA	99$_H$によるフラッシュA/ROM保護, 66$_H$によるアクセス許可
01$_H$		予約	
02$_H$-11$_H$		MSUKA	保護済みフラッシュA/ROM用解除キー
12$_H$-1B$_H$		予約	
1C$_H$-1F$_H$		FWPAMA	292D3A7B$_H$によるフラッシュ書込み保護の起動
20$_H$-27$_H$		FWPSMA	フラッシュ書込み保護用のセクタ選択
28$_H$-2F$_H$		予約	
30$_H$-33$_H$	ブートシーケンス構成ブロック	FBVAM	292D3A7B$_H$による内部ベクタモードでの固定ブートベクタの起動
34$_H$-37$_H$		USDM	292D3A7B$_H$による内部ベクタモードでのUARTスキャンの停止
38$_H$-3F$_H$		予約	
40$_H$-43$_H$	バックグラウンドデバッグ構成ブロック	BDMAM	292D3A7B$_H$によるBDMの起動
44$_H$-7F$_H$		予約	BDMの構成

Table 8-5. ROM構成ブロックB (RCBB)の構造

オフセット	サブブロック	マーカ	備考
00H	セキュリティ保護構成ブロック	MSBB	99$_H$によるフラッシュB保護, 66$_H$によるアクセス許可
01H		予約	
02H-11H		MSUKB	保護済みフラッシュB用解除キー
12H-1BH		予約	
1CH-1FH		FWPAMB	292D3A7B$_H$によるフラッシュ書込み保護の起動
20H-27H		FWPSMB	フラッシュ書込み保護用のセクタ選択
28H-2FH		予約	

Table 8-6. データフラッシュ構成ブロックA (DFCBA)の構造

オフセット	サブブロック	マーカ	備考
00H	セキュリティ保護構成ブロック	MSBDA	99_HによるデータフラッシュA保護, 66_Hによるアクセス許可
01H		予約	
02H-11H		MSUKDA	保護済みデータフラッシュA用解除キー
12H-2FH		予約	

Table 8-7. データフラッシュ構成ブロックB(DFCBB)の構造

オフセット	サブブロック	マーカ	備考
00H	セキュリティ保護構成ブロック	MSBDB	99_HによるデータフラッシュB保護, 66_Hによるアクセス許可
01H		予約	
02H-11H		MSUKDB	保護済みデータフラッシュB用解除キー
12H-2FH		予約	

8.4.5 RCBマーカの機能説明

■ セキュリティ保護構成ブロック(SPCB)

読出しセキュリティ機能はフラッシュ ROM 内容の不正読出しを防止し , 内部ベクタモードでのアプリケーションによる読出しアクセス以外すべてのケースを対象とします。

詳細については , 「35.2 ROM/ フラッシュセキュリティの使用方法」を参照してください。

書込み保護機能は, 内部ベクタモードおよび外部ベクタモードでのアプリケーションによる意図的でないフラッシュメモリの書込みを防止します。書込み保護機能は , フラッシュ A とフラッシュ B でのみサポートされています。

詳細については , 「33.8.8 セクタ消去 / 書込み保護」を参照してください。

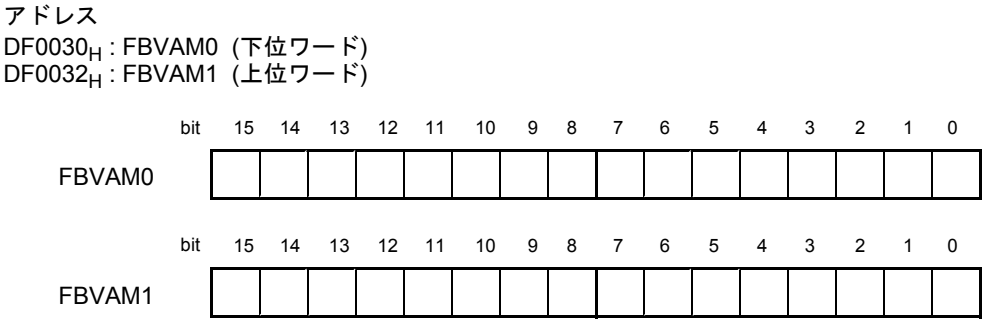
なお , 内蔵モニタデバッガカーネルが起動されている場合 , 読出しセキュリティ設定はブートシーケンスにも影響します。

■ ブートシーケンス構成ブロック(BSCB)

・固定ブートベクタの許可

内部ベクタモードでは FF:FFDC$_H$ からの読出し値の代わりに , ブートベクタ (ユーザプログラムの開始アドレス) を固定値 DF:0080$_H$ に設定することができます。この固定ブートベクタは小さなフラッシュセクタでプログラム実行する事により , ブートローダの実行も可能です。

Figure 8-4. 固定ブートベクタ起動マーカの構成

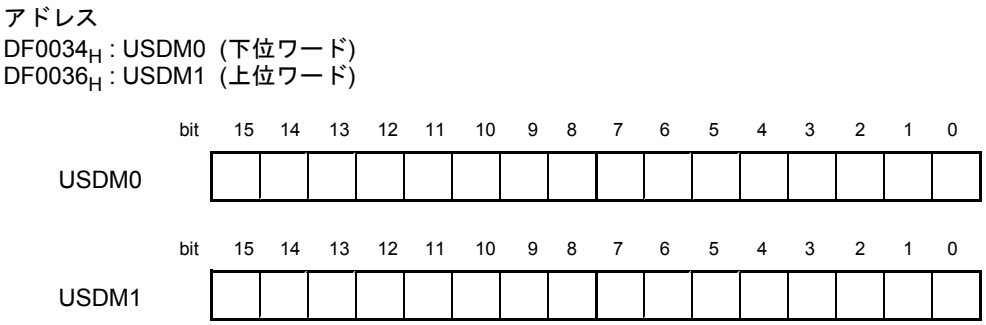


{FBVAM1, FBVAM0}の値が$292D3A7B_H$と同じ場合は, 固定ブートベクタが使用されます。{FBVAM1, FBVAM0}以外の値の場合は , ブートベクタはブートベクタ位置から読み出されます。

・一時的 UART スキャンの禁止

内部ベクタモードでのメインリセット後 , モード端子の設定を変更することなしに , シリアル通信モードにあるようなシリアル通信要求に対して , 専用 UART チャネルがスキャンされます。起動時間の短縮のために , このスキャン処理を禁止することができます。

Figure 8-5. UARTスキャン非起動マーカの構成



{USDM1, USDM0}の値が$292D3A7B_H$と同じ場合は, 一時的なUARTスキャンは行われません。{USDM1, USDM0}以外の値の場合は , ブート ROM はメインリセット後の一定期間 , 専用 UART チャネルをスキャンします。

■ BDM構成ブロック(BDCB)

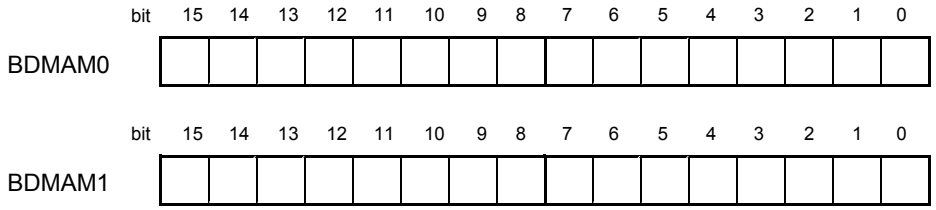
内部ベクタモードおよび外部ベクタモードでは , 内蔵モニタデバッガカーネルを起動することができます (バックグラウンドデバッグモード :BDM)。しかし , バックグラウンドデバッグモード (BDM) やフォアグラウンドデバッグモード (FDM) で MCU を動作するには , 外部ツールが必要です。また , デバッガ動作は BDM 構成ブロックでの設定によって最適化する必要があります。適切なツールのサポートなしに本機能を使用しないでください。

・内蔵モニタデバッガカーネルの許可

内部ベクタモードおよび外部ベクタモードでは , 内蔵モニタデバッガカーネルを起動することができます。

Figure 8-6. BDM起動マーカの構成

アドレス
$DF0040_H$: BDMAM0 (下位ワード)
$DF0042_H$: BDMAM1 (上位ワード)

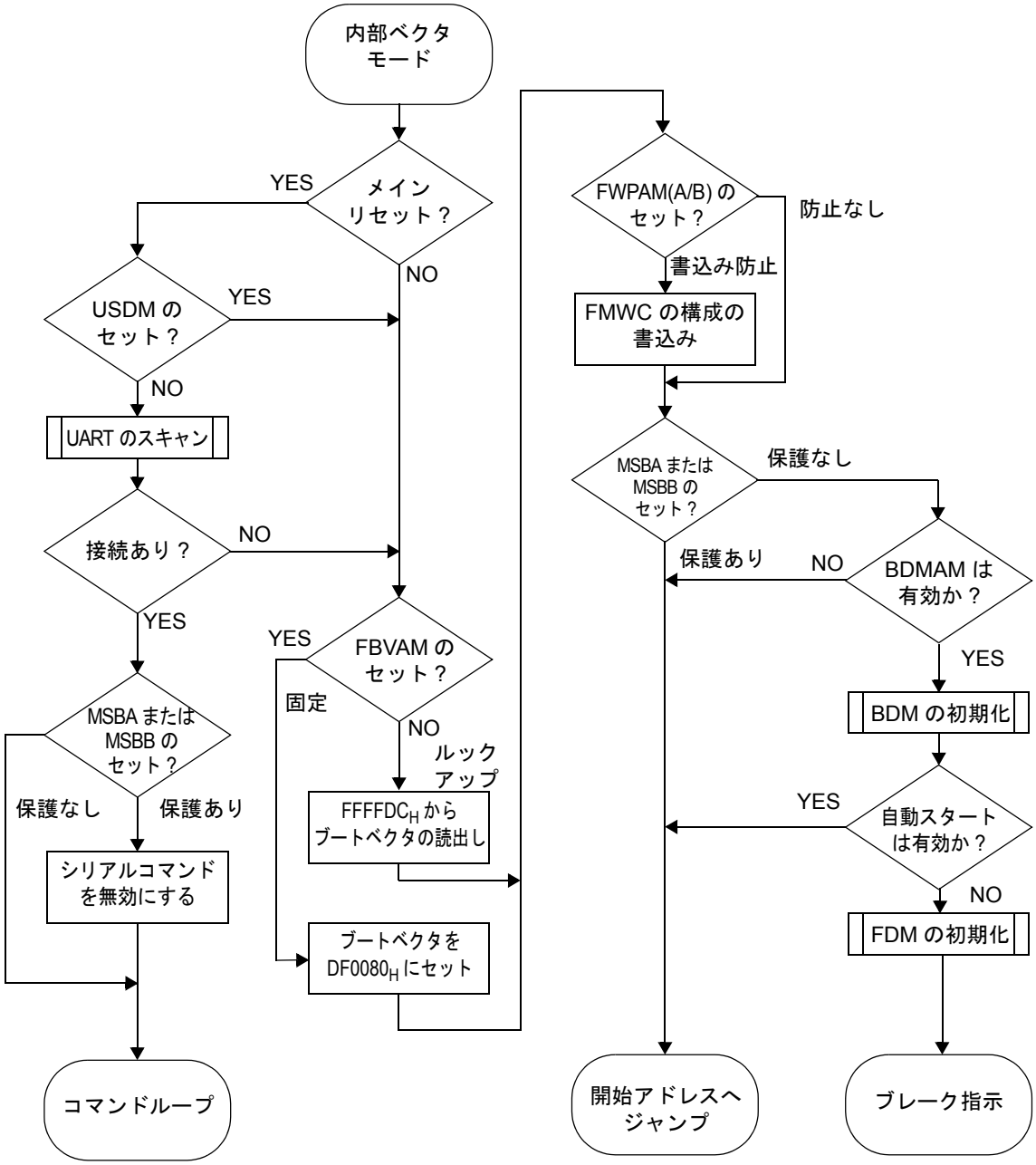


{BDMAM1, BDMAM0} の値が $292D3A7B_H$ と同じ場合は , BDM が起動されます。{BDMAM1, BDMAM0} 以外の値の場合は , BDM は起動されません。

なお , ROM/ フラッシュメモリのセキュリティが設定されている場合 , BDM は起動されません。

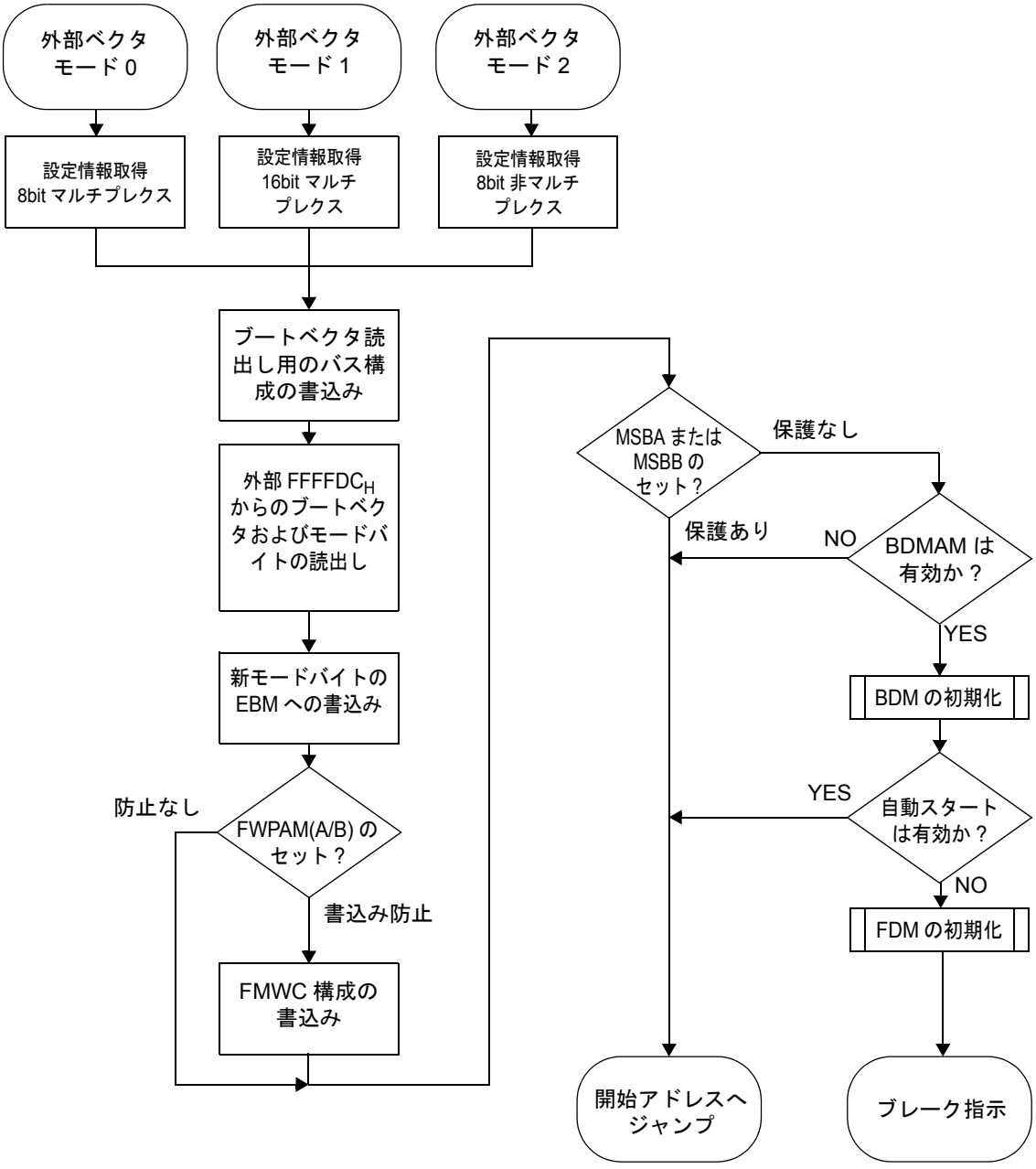
8.4.6 内部ベクタモードのフローチャート

Figure 8-7. 内部ベクタモードのブートROMシーケンス



8.4.7 外部ベクタモードのフローチャート

Figure 8-8. 外部ベクタモードのブートROMシーケンス



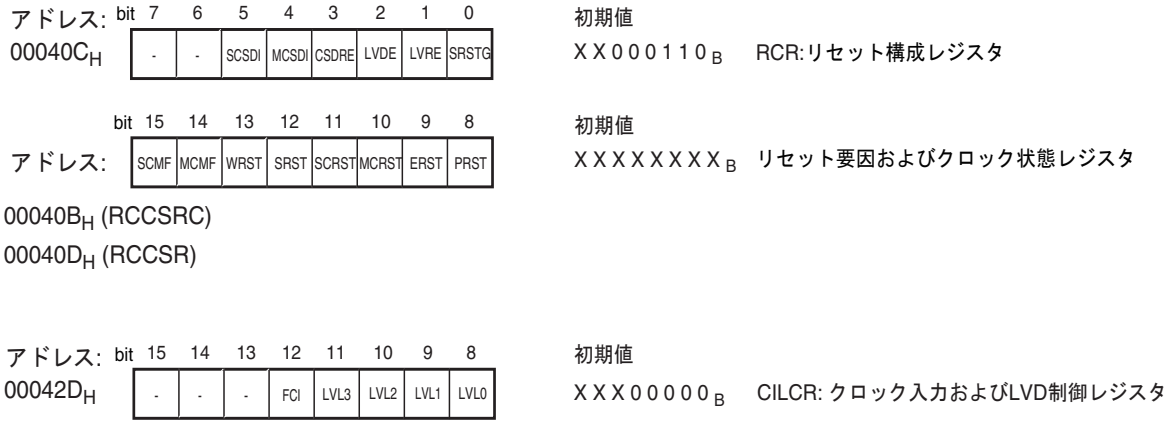
8.5 リセット制御レジスタ

リセット制御レジスタを一覧表示し, 各レジスタの機能について詳しく説明します。

リセット制御レジスタ

リセットコントローラには , リセット構成レジスタ (RCR) とリセット要因およびクロック状態レジスタ (RCCSR/RCCSRC) の 2 種類のレジスタがあります。リセット制御レジスタの概要を Figure 8-9 に示します。

Figure 8-9. リセット制御レジスタ



8.5.1 リセット構成レジスタ(RCR)

リセット構成レジスタ (RCR) は, ソフトウェアリセットのアサート, 低電圧リセットと低電圧検出器の構成, およびクロック停止検出回路の構成に使用します。

リセット構成レジスタ(RCR)の構成

リセット構成レジスタ (RCR) の構成を Figure 8-10 に示し , 各ビットの機能について Table 8-8 で説明します。

Figure 8-10. リセット構成レジスタ(RCR)の構成

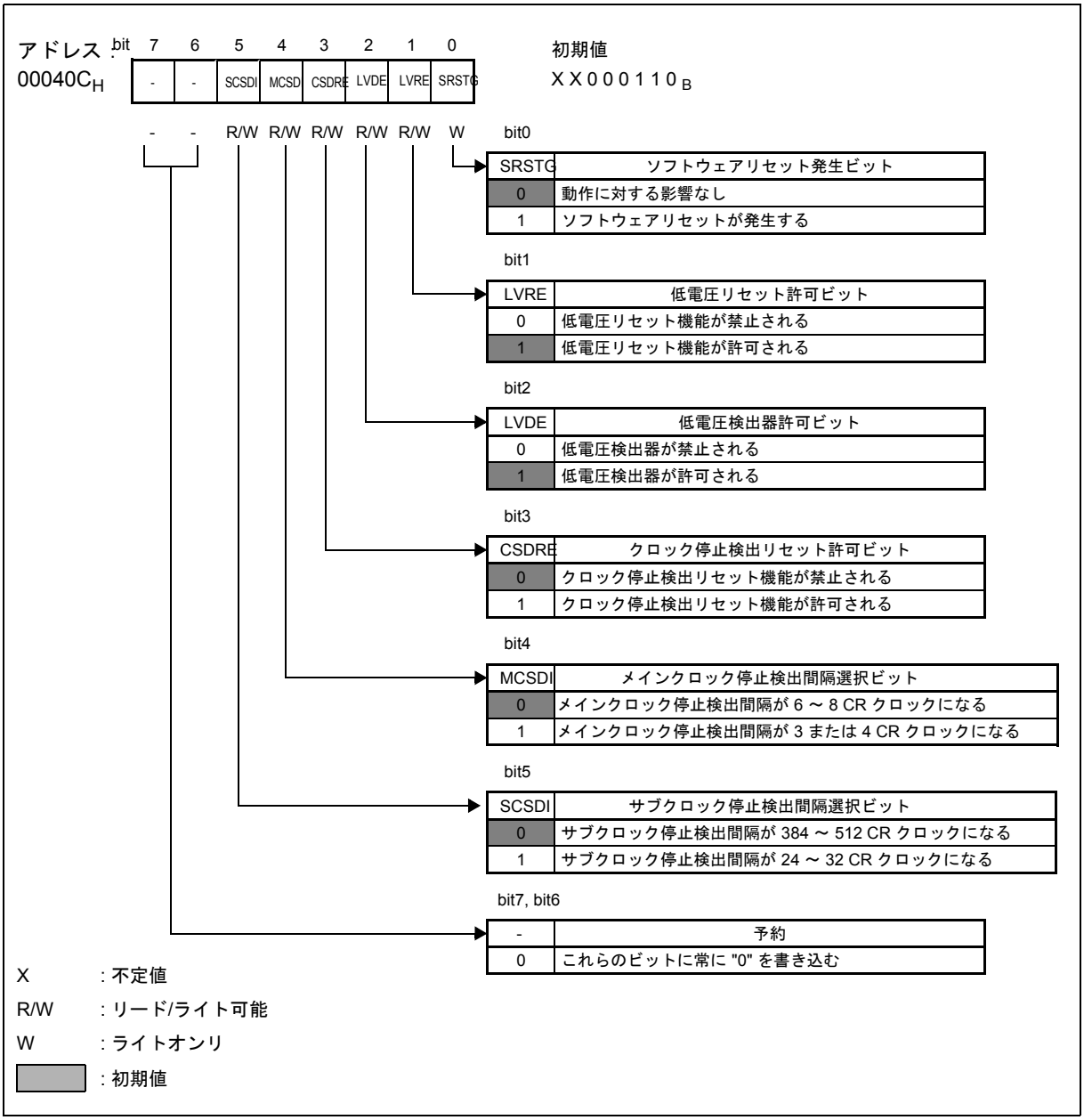


Table 8-8. リセット構成レジスタ(RCR)の各ビットの機能説明

ビット名		機能
bit 0	SRSTG: ソフトウェア リセット発生 ビット	• このビットに "1" を書き込むと , 内部ソフトウェアリセットが発生します。 • このビットに "0" を書き込んだ場合は , 何も影響がありません。 • このビットの読出し値は常に "0" になります。
bit 1	LVRE: 低電圧リセット 許可ビット	• このビットは , 電源リセットのリセット要因の 1 つとなる低電圧リセット機能を制御します。 • このビットを "1" に設定すると , 低電圧リセット機能が許可されます。 • このビットに "0" を書き込むと , 低電圧リセット機能が禁止されます。 • このビットは , 外部リセットおよび電源 (電源投入) リセットが発生した場合にのみ "1" (低電圧リセットが許可される) に初期化されます。 • このビットが "1" に設定されている限り , (LVDE ビットへの "0" の書込みによって) 低電圧検出器を禁止しないでください。 • 低電圧検出器が許可されて安定するまで , このビットを "0" から "1" に設定しないでください。 • 低電圧リセット回路が内部ベクタモードで常に許可されているデバイスでは , "011" (永続的低電圧リセット機能) に設定されたモード端子に対してこのビットの設定は影響を与えません。
bit 2	LVDE: 低電圧検出器許可ビット	• このビットは , 低電圧リセット機能に使用される低電圧検出器を制御します。 • このビットを "1" に設定すると , 低電圧検出器が許可されます。 • このビットに "0" を書き込むと , 低電圧検出器が禁止されます。 • このビットは , 外部リセットおよび電源 (電源投入) リセットが発生した場合にのみ "1" (低電圧検出器が許可される) に初期化されます。 • LVRE が "1" に設定されている限り , このビットを "0" に設定しないでください。 • このビットを "1" に設定すると , 低電圧検出器が許可されます。ただし , データシートに記載されているように , この回路では許可された後の安定時間が必要となります。この時間が経過するまで低電圧リセット機能を許可しないでください。 • 低電圧リセット回路が内部ベクタモードで常に許可されているデバイスでは , "011" (永続的低電圧リセット機能) に設定されたモード端子に対してこのビットの設定は影響を与えません。
bit 3	CSDRE: クロック停止検出リセット許可 ビット	• このビットは , クロック停止検出リセット機能を許可します。 • このビットへの書込みができるのは , システムクロック 1 が CR クロックに設定されている場合に限ります。 • このビットを "1" に設定すると , クロック停止検出リセットが許可されます。 • このビットに "0" を書き込むと , クロック停止検出リセットが禁止されます。 • このビットはリセットごとに "0" (クロック停止検出リセットが禁止される) に初期化されます。 • 許可された後 , クロック停止検出リセットは以下の 3 通りの場合にアサートされます。 1. システムクロック CLKS1/CLKS2 またはウォッチドッグタイマとしてメイン /PLL クロックが選択されている状態でメインクロックの欠落が検出された。 2. システムクロック CLKS1/CLKS2 またはウォッチドッグタイマとしてサブクロックが選択されている状態でサブクロックの欠落が検出された。 CR 発振器を禁止するために CKSR: RCE ビットに "0" が書き込まれた。
bit 4	MCSDI: メインクロック 停止検出間隔選択ビット	• このビットは , メインクロック停止検出回路の測定間隔を制御します。 • このビットはリセットごとに "0" に初期化されます。 • このビットに "0" を書き込むと , 間隔が 6 ～ 8 CR クロックに設定されます。 • このビットに "1" を書き込むと , 間隔が 3 または 4 CR クロックに設定されます。 • メインクロック停止検出回路では , 選択された間隔内で監視対象メインクロック入力信号 (CLKMC)の立上りエッジが検出されなかった場合にメインクロック欠落フラグ(RCCSR:MCMF)が設定されます。
bit 5	SCSDI: サブクロック停止検出 間隔選択 ビット	• このビットは , サブクロック停止検出回路の測定間隔を制御します。 • このビットはリセットごとに "0" に初期化されます。 • このビットに "0" を書き込むと , 間隔が 384 ～ 512 CR クロックに設定されます。 • このビットに "1" を書き込むと , 間隔が 24 ～ 32 CR クロックに設定されます。 • サブクロック停止検出回路では , 選択された間隔内で監視対象サブクロック入力信号 (CLKSC) の立上りエッジが検出されなかった場合にサブクロック欠落フラグ (RCCSR:SCMF)が設定されます。
bit 6, bit 7	予約	• これらのビットに常に "0" を書き込みます。 • これらのビットの読出し値は未定義です。 • このレジスターへのリードモディファイライト系命令は無効です。

8.5.2 リセット要因およびクロック状態レジスタ(RCCSR/RCCSRC)

RCCSR/RCCSRC レジスタは, メインおよびサブクロックのリセット要因と状態を示します。

8.5.2.1 リセット要因およびクロック状態レジスタ(RCCSR/RCCSRC)の構成

リセット要因およびクロック状態レジスタ (RCCSR/RCCSRC) には , 2 つのアドレスを介してアクセスできます。アドレス $00040B_H$ (RCCSRC) に読出しアクセスした場合は読出し後にこのレジスタのビットがすべてクリアされますが , アドレス $00040D_H$ (RCCSR) に読出しアクセスした場合はこのレジスタの状態が変わりません。RCCSR/RCCSRC レジスタへの書込みは無視されます。RCCSR/RCCSRC レジスタの構成を Figure 8-11 に示し , 各ビットの機能について Table 8-9 で説明します。

Figure 8-11. リセット要因およびクロック状態レジスタ(RCCSR/RCCSRC)の構成

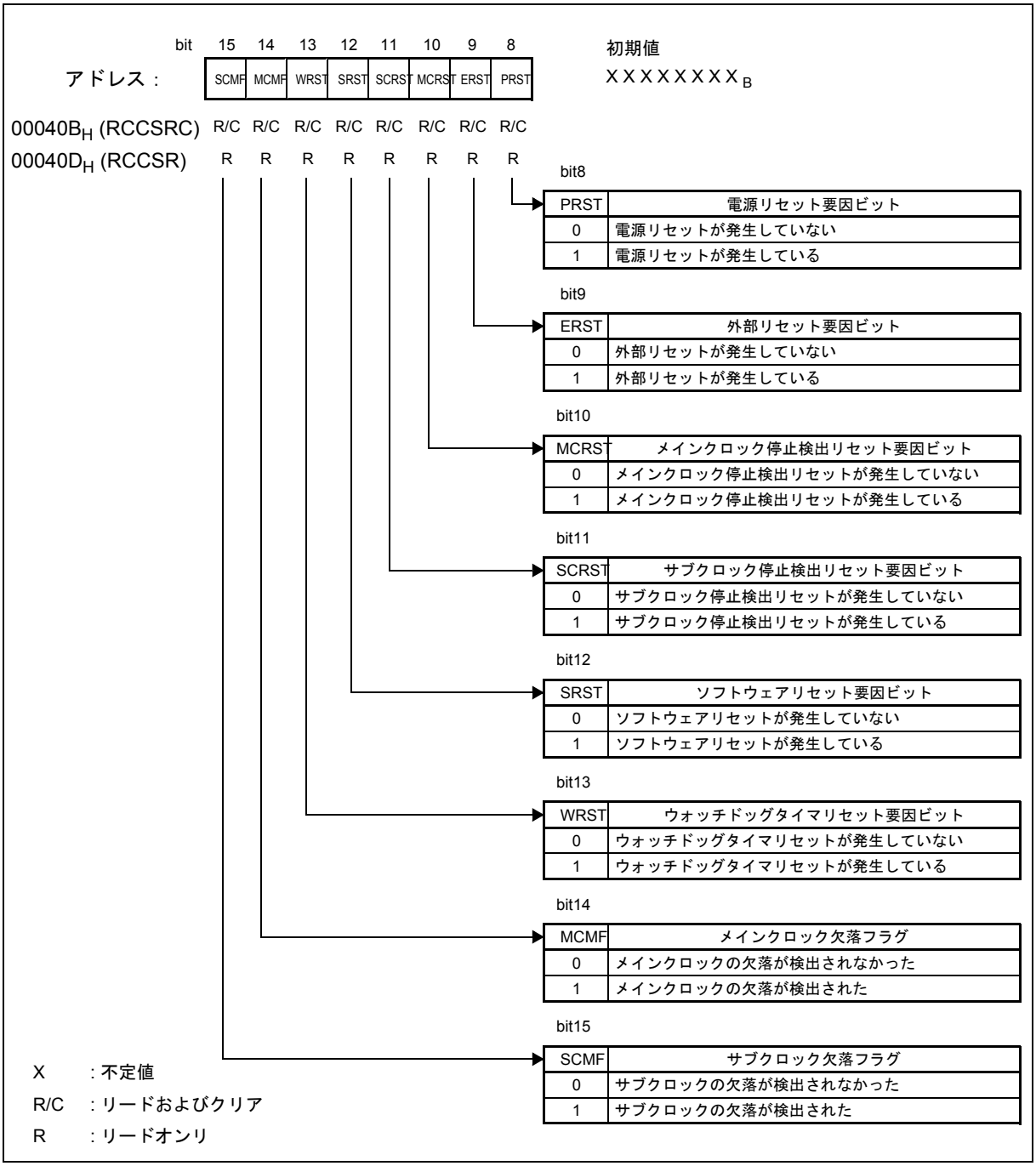


Table 8-9. リセット要因およびクロック状態レジスタ(RCCSR)の各ビットの機能説明 (Sheet 1 of 2)

ビット名		機能
bit 8	PRST: 電源リセット 要因ビット	• このビットは , 電源リセットが発生しているかどうかを示します。 • 電源投入後に , このビットは "1" に設定されます。 • このビットは , アドレス $00040B_H$ RCCSRC への読出しアクセスを行うことによってのみクリアできます。 • このビットが "1" に設定されるのは , 電源リセットが発生した場合です。考えられるリセット要因は以下のとおりです。 1. 電源投入イベント：データシートに記載されているプロファイルに従って電源が投入された。 低電圧イベント：低電圧リセット機能が許可され, かつ電源がデータシートに記載されている値を下回った。
bit 9	ERST: 外部リセット 要因ビット	• このビットは , 外部リセットが発生しているかどうかを示します。 • このビットが "1" に設定されるのは , RSTX を "0" にアサートすることによって外部リセットが発生した場合です。 • 電源リセットが発生した場合にも , このビットが "1" に設定されます。 • このビットは, アドレス $00040B_H$ RCCSRC への読出しアクセスを行うことによってのみクリアできます。
bit 10	MCRST: メインクロック停止検出 リセット要因ビット	• このビットは , メインクロック停止検出リセットが発生しているかどうかを示します。 • 電源リセットが発生しても , このビットは初期化されません。 • このビットは , アドレス $00040B_H$ RCCSRC への読出しアクセスを行うことによってのみクリアできます。 • このビットが "1" に設定されるのは , メインクロック停止検出リセットが発生した場合です。考えられるリセット要因は以下のとおりです。 1. システムクロック CLKS1/CLKS2 またはウォッチドッグタイマとしてメイン /PLL クロックが選択されている状態でメインクロックの欠落が検出された。 クロック停止検出リセットが許可されている状態で CR クロックの許可ビット (CKSR: RCE) に "0" が書き込まれた。
bit 11	SCRST: サブクロック 停止検出 リセット要因ビット	• このビットは , サブクロック停止検出リセットが発生しているかどうかを示します。 • 電源リセットが発生しても , このビットは初期化されません。 • このビットは , アドレス $00040B_H$ RCCSRC への読出しアクセスを行うことによってのみクリアできます。 • このビットが "1" に設定されるのは , サブクロック停止検出リセットが発生した場合です。考えられるリセット要因は以下のとおりです。 1. システムクロック CLKS1/CLKS2 またはウォッチドッグタイマとしてサブクロックが選択されている状態でサブクロックの欠落が検出された。 クロック停止検出リセットが許可されている状態で CR クロックの許可ビット (CKSR: RCE) に "0" が書き込まれた。

Table 8-9. リセット要因およびクロック状態レジスタ(RCCSR)の各ビットの機能説明 (Sheet 2 of 2)

ビット名		機能
bit 12	SRST: ソフトウェア リセット要因 ビット	• このビットは , ソフトウェアリセットが発生しているかどうかを示します。 • 電源リセットが発生しても , このビットは初期化されません。 • このビットは , アドレス $00040B_H$ RCCSRC への読出しアクセスを行うことによってのみクリアできます。 • このビットが "1" に設定されるのは, RCR: SRSTG ビットへの "1" の書込みによってソフトウェアリセットが発生した場合です。
bit 13	WRST: ウォッチドッグタイマリセット要因ビット	• このビットは , ウォッチドッグタイマリセットが発生しているかどうかを示します。 • 電源リセットが発生しても , このビットは初期化されません。 • このビットは , アドレス $00040B_H$ RCCSRC への読出しアクセスを行うことによってのみクリアできます。 • このビットが "1" に設定されるのは , ウォッチドッグタイマリセットが発生した場合です。考えられるリセット要因は以下のとおりです。 1. ウォッチドッグタイマは起動しているが , 選択された間隔内でクリアされなかった。 2. ウォッチドッグタイマクリアパターンレジスタに不正な値が書き込まれた。 3. ウォッチドッグタイマのソースクロックとして使用されているクロックを禁止するために, 該当するクロックの許可ビット (CKSR: RCE, MCE または SCE) に "0" が書き込まれた。 4. ウォッチドッグタイマクロック選択ビットが "01" (CR クロック周波数の変更が許可されない) に設定されている状態で CR クロック周波数が変更された。 WDTC:RSTP ビットが"1"に設定されている状態でストップモードへの移行が要求された。
bit 14	MCMF: メインクロック欠落フラグ	• このビットは , メインクロックの欠落が検出されたかどうかを示します。 • 電源リセットが発生しても , このビットは初期化されません。 • このビットが "1" に設定されるのは , メイン発振器が許可されており , かつ RCR: MCSDI ビットで定義された間隔内で監視対象メインクロック入力信号 (CLKMC) の立上りエッジが検出されなかった場合です。 • このビットをクリアするには, アドレス $00040B_H$ RCCSRC への読出しアクセスを行います。
bit 15	SCMF: サブクロック 欠落フラグ	• このビットは , サブクロックの欠落が検出されたかどうかを示します。 • 電源リセットが発生しても , このビットは初期化されません。 • このビットが "1" に設定されるのは, サブ発振器が許可されており, かつ RCR: SCSDI ビットで定義された間隔内で監視対象サブクロック入力信号 (CLKSC) の立上りエッジが検出されなかった場合です。 • このビットをクリアするには, アドレス $00040B_H$ RCCSRC への読出しアクセスを行います。

8.5.2.2 リセット要因ビットに関する注意事項

■ リセット要因レジスタを読み出す前の複数のリセット要因

リセット要因レジスタを読み出す前に複数のリセット要因が発生した場合は,RCCSR/RCCSRCレジスタの対応するリセット要因ビットがすべて"1"に設定されます。 例えば,RSTX端子経由の外部リセット要求とウォッチドッグタイマリセットが同時に発生した場合は,ERSTビットとWRSTビットが両方とも"1"に設定されます。 これらの対応関係を以下の表に示します。

Table 8-10. リセット要因ビットとリセット要因との対応関係

リセット要因	PRST	ERST	MCRST	SCRST	SRST	WRST
電源リセット (電源投入または低電圧)	1	1	X	X	X	X
RSTX端子経由の外部リセット要求	*	1	*	*	*	*
メインクロック停止検出リセット	*	*	1	*	*	*
サブクロック停止検出リセット	*	*	*	1	*	*
ソフトウェアリセット要求	*	*	*	*	1	*
ウォッチドッグタイマのオーバフロー	*	*	*	*	*	1

*: 前の状態が維持される
X: 不定

■ 電源リセット

PRST = 1の場合は, それ以外のリセット要因ビットをすべて無視するようにソフトウェアをプログラミングする必要があります。

■ リセット要因ビットのクリア

リセット要因ビットがクリアされるのは , アドレス $00040B_H$ のリセット要因およびクロック状態レジスタRCCSRC を読み出した場合に限ります。既に発生しているリセット要因に対応したビットは, 別のリセットが発生してもクリアされません ("1" の設定が保持される)。

電源リセットが発生したら, すべてのビットを初期化するために読出しによってレジスタをクリアする必要があります。

＜注意事項＞

パワーオンリセットが発生していない (データシートに記載されている電源投入プロファイルに一致していない) 状況で電源を投入した場合 , RCCSR レジスタの値は保証されません。

8.5.2.3 クロック欠落フラグに関する注意事項

クロック欠落フラグ MCMF および SCMF の詳細は ,「8.6 クロック停止検出機能およびリセットの動作」を参照してください。

8.5.3 クロック入力およびLVD制御レジスタ(CILCR)

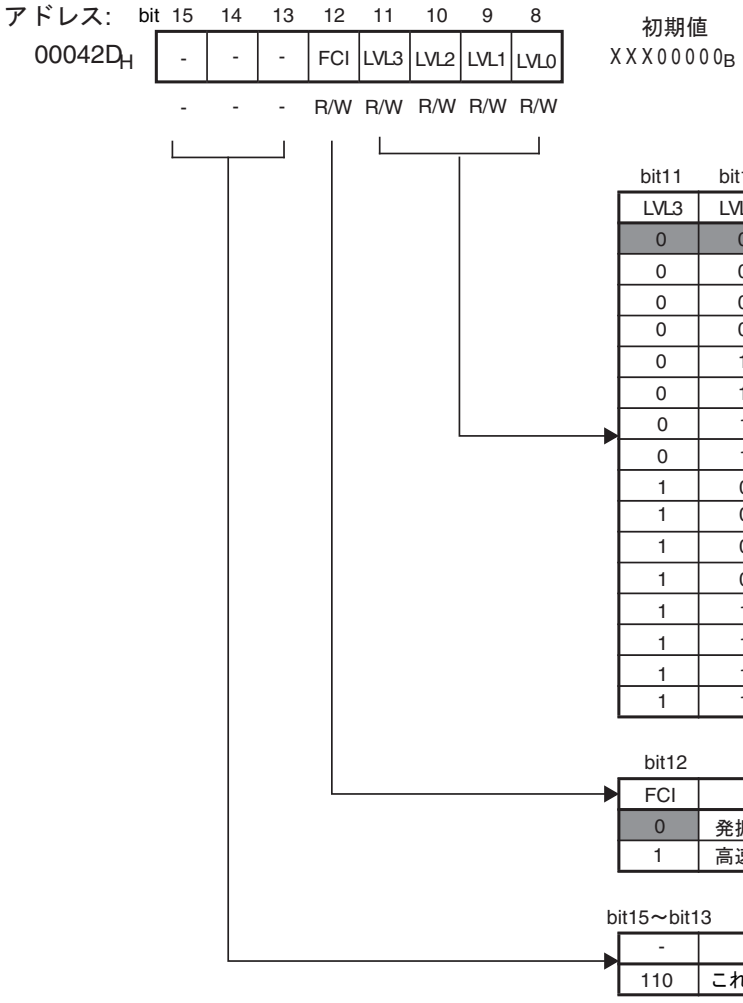
クロック入力およびLVD制御レジスタ(CILCR) は発振回路と低電圧検出レベルの制御に使用します。

クロック入力およびLVD制御レジスタ(CILCR)の構成

Figure 8-12 にクロック入力およびLVD制御レジスタ(CILCR)の構成を示し, Table 8-11に各ビットの機能を説明します。

このレジスタは 16 ビット幅 (VRCR) および 8 ビット幅 (下位バイト : VRCR, 上位バイト CILCR) にアクセスできます。

Figure 8-12. クロック入力およびLVD制御レジスタ(CILCR)の構成



bit11	bit10	bit9	bit8	
LVL3	LVL2	LVL1	LVL0	低電圧検出レベル
0	0	0	0	レベル0 *
0	0	0	1	レベル1 *
0	0	1	0	レベル2 *
0	0	1	1	レベル3 *
0	1	0	0	レベル4 *
0	1	0	1	レベル5 *
0	1	1	0	レベル6 *
0	1	1	1	レベル7 *
1	0	0	0	レベル8 *
1	0	0	1	レベル9 *
1	0	1	0	レベル10 *
1	0	1	1	レベル11 *
1	1	0	0	レベル12 *
1	1	0	1	レベル13 *
1	1	1	0	レベル14 *
1	1	1	1	レベル15 *

bit12

FCI	高速クロック入力選択
0	発振モード
1	高速外部クロック入力モード

bit15～bit13

-	予約
110	これらのビットには常に"110"を書き込む

*: 設定可能なアナログ検出レベルについては, データシートを参照してください。

Table 8-11. クロック入力およびLVD制御レジスタ(CILCR)の各ビットの機能説明

<table>
<tr><th colspan="2">ビット名</th><th>機能</th></tr>
<tr><td>bit 7 ～ bit 5</td><td>LVL3 ～ LVL0:
低電圧検出レベル選択ビット</td><td>
• これらのビットは , 低電圧検出器 (LVD) のアナログしきい値を制御します。
<table>
<tr><th>LVL3</th><th>LVL2</th><th>LVL1</th><th>LVL0</th><th>低電圧検出レベル</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>レベル 0 *</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>レベル 1 *</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>レベル 2 *</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>レベル 3 *</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>レベル 4 *</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>レベル 5 *</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>レベル 6 *</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>レベル 7 *</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>レベル 8 *</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>レベル 9 *</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>レベル 10 *</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>レベル 11 *</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>レベル 12 *</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>レベル 13 *</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>レベル 14 *</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>レベル 15 *</td></tr>
</table>
• これらのビットはリセットごとに "0000" に初期化されます。
• この機能が有効で外部供給電圧降下が選択したしきい値を超えた場合 , 低電圧検出回路はリセット要求を出力します。

*: 設定可能なアナログしきい値については, データシートを参照してください。詳細については「第8章 リセットとスタートアップ」を参照してください。
</td></tr>
<tr><td>bit 12</td><td>FCI
高速クロック入力選択ビット</td><td>
• このビットはリセットで "0" に初期化されます。
• このビットはメイン発振が発振モードか高速クロック入力モードかを選択します。
<table>
<tr><th>FCI</th><th>高速クロック入力制御</th></tr>
<tr><td>0</td><td>発振モード</td></tr>
<tr><td>1</td><td>高速外部クロック入力モード</td></tr>
</table>
• メイン発振用の端子 X0 と X1 に水晶発振器か振動子を接続するときは , このビットを "0" に設定します。このモードでは X0/X1 端子に外部低周波クロックを接続することも可能です。
• 外部低周波クロックを入力する場合は , 外部クロック (メイン) に切り換える前に , このビットを "1" に設定してください。
• 両モードの設定周波数範囲は , データシートに記載されています。
• この機能はすべてのデバイスで使用できません。詳細はデータシートを参照してください。
</td></tr>
<tr><td>bit 13 ～ bit 15</td><td>予約ビット</td><td>
• これらのビットに常に "110" を書き込みます。
• これらのビットの読出し値は未定義です。
• これらのレジスタへのリードモディファイライト系命令は無効です。
</td></tr>
</table>

8.6 クロック停止検出機能およびリセットの動作

外部メインまたはサブ発振器の故障を検出するクロック停止検出回路の動作について説明します。

8.6.1 クロック停止検出回路の機能

クロック停止検出回路は , メインクロックとサブクロックの発振を監視します。選択された間隔内で監視対象クロックの立上りエッジが検出されなかった場合は, このクロックが許可されていても, 対応するクロック欠落フラグが設定されます。この機能は , CR 発振器が許可されている限り常にアクティブになります。

MCU のハングアップを防止するために故障したクロックがシステムクロック CLKS1/CLKS2 またはウォッチドッグタイマとして現在使用されている場合は , 別のクロック停止検出リセットをアサートすることができます。このリセットが発生すると , MCU は必ず CR クロックモードに切り換わります。このモードにある場合 , CPU はリセット要因をチェックし , MCU を安全な状態に移すことができます。

8.6.2 クロック停止検出回路の構成

クロック停止検出回路には 3 つの構成ビットがあります。1 つはクロック停止リセット機能を許可するビットであり , それ以外の 2 つは検出間隔を選択するビットです。

■ **クロック停止検出リセット許可ビット (RCR: CSDRE)**

リセットが発生すると, CSDRE の初期値は "0" (クロック停止検出リセットが禁止される) になります。このビットを "1" に設定するとクロック停止検出リセット機能が許可され , "0" に設定すると禁止されます。ただし , CSDRE ビットへの書込みができるのは ,
システムクロック 1 (CLKS1) が CR クロックに設定されている場合に限ります。

■ **クロック停止検出間隔選択ビット (RCR: MCSDI および SCSDI)**

クロックの欠落が検出されるのは, 特定の間隔内で監視対象クロックの立上りエッジが検出されなかった場合です。この間隔は特定数の CR クロックであるため , 選択された CR クロック周波数によって異なります。2 つのビットを使用して , 該当する間隔を選択し , 選択された CR クロック周波数と監視対象外部クロック周波数に合わせて監視時間を調整することができます。

Table 8-12. クロック停止検出間隔

監視対象クロック	選択ビット	設定	クロック停止検出間隔		
			CRクロック	2MHz の CR クロック周波数に要する時間 (最小値 - 最大値) / (最小外部周波数)	100kHzのCRクロック周波数に要する時間 (最小値 - 最大値) / (最小外部周波数)
メインクロック	MCSDI	0 (初期値)	6 - 8	1.5µs - 8µs / (~700kHz)	30µs - 160µs / (~35kHz)
		1	3 - 4	0.75µs - 4µs / (~1.4MHz)	15µs - 80µs / (~70kHz)
サブクロック	SCSDI	0 (初期値)	384 - 512	96µs - 512µs / (~11kHz)	1.92ms - 10.2ms / (~0.55kHz)
		1	24 - 32	6µs - 32µs / (~170kHz)	12µs - 640µs / (~8.5kHz)

最短時間を指定すると, 監視対象クロックの 2 つの立上りエッジ間でこの時間が経過した後に停止検出が実行されます。最長時間を指定すると , この時間が経過した後にクロック停止が検出されます。最短時間は監視対象クロックの周期より長く設定してください (監視対象クロックの周波数は必ず Table 8-12 に示された最小外部周波数より高くなければなりません)。
クロック停止検出機能の動作中に CR クロック周波数を変更してもかまいませんが , これによって検出間隔が変更されます。

8.6.3 クロック欠落フラグMCMFとSCMF

メインクロック欠落フラグ MCMF はメインクロックの欠落を示し , サブクロック欠落フラグ SCMF はサブクロックの欠落を示します。

■ **電源投入後の状態**

スタートアップ後にクロック欠落フラグ MCMF と SCMF は未定義になります。

■ **クロック欠落フラグのクリア**

MCMF ビットと SCMF ビットは , アドレス $00040B_H$ を介して RCCSRC レジスタへの読出しアクセスを行うことによってクリアできます。

■ **クロック欠落フラグの設定**

メインクロック停止検出回路が許可されるのは , CR クロックが動作中であり , かつメイン発振器がアクティブになっている場合です (アクティブの意味は , メインクロック許可ビット CKSR: MCE が "1" に設定されているか , またはメインクロックモニタビット CKMR: MCM が "1" に設定されていることです)。選択された間隔内でメインクロック入力信号の立上りエッジが検出されなかった場合は , メインクロック欠落フラグが "1" に設定されます。

サブクロック停止検出回路が許可されるのは , CR クロックが動作中であり , かつサブ発振器がアクティブになっている場合です (アクティブの意味は , サブクロック許可ビット CKSR: SCE が "1" に設定されているか , またはサブクロックモニタビット CKMR: SCM が "1" に設定されていることです)。選択された間隔内でサブクロック入力信号の立上りエッジが検出されなかった場合は , サブクロック欠落フラグが "1" に設定されます。

8.6.4 クロック停止検出リセット

システムクロック (CLKS1/CLKS2) またはウォッチドッグタイマ用のソースとして使用されているクロックの故障が検出されると , クロック停止検出リセット機能によってリセットが発生します

■ **クロック停止検出リセットの許可**

クロック停止検出リセットを許可するには , RCR: CSDRE ビットを "1" に設定します。

■ **クロック停止検出リセットの発生**

以下の場合にクロック停止検出リセットが発生します。クロック停止検出リセット許可ビット CSDRE は必ず "1" (リセットが許可される) に設定しなければなりません。

- メインクロックが安定しており (CKMR:MCM=1),かつシステムクロック CLKS1/CLKS2 またはウォッチドッグタイマとしてメイン/PLL クロックが使用されている状態でメインクロックの欠落が検出された。 この場合には, MCRST (メインクロック停止検出リセット要因) ビットが設定され, リセットが発生します。
- サブクロックが安定しており (CKMR:SCM=1),かつシステムクロック CLKS1/CLKS2 またはウォッチドッグタイマとしてサブクロックが使用されている状態でサブクロックの欠落が検出された。 この場合には, SCRST (サブクロック停止検出リセット要因) ビットが設定され, リセットが発生します。
- CKSR: RCE ビットに "0" (CR クロックが禁止される) が書き込まれた。 この場合には, 両方のクロック停止検出リセット要因ビットが設定され, リセットが発生します。 クロック停止検出リセットが許可されている限り , CR クロックを禁止しないでください。

＜注意事項＞

クロック停止検出リセットが発生するのは , 故障したクロックがシステムクロック 1/2 またはウォッチドッグタイマとして使用されている場合だけです。故障したクロックが 1 つのリソース (メインクロックタイマ , サブクロックタイマなど) でのみ使用されており ,
システムクロックが別のクロックに設定されている場合は , リセットが発生しません。これによってシステムがハングアップする可能性がある場合は , 復帰目的にウォッチドッグタイマを使用してください。

■ クロック停止検出リセットの影響

クロック停止検出リセットが発生すると , グローバルリセット信号がアサートされ , 対応するクロック停止検出リセット要因ビットが設定され , MCU がリセット状態に移されます。

また , クロック停止検出リセットによって , 故障したクロックのソースクロックタイマとクロック準備完了モニタビットがクリアされます。

リセットが解除されると , MCU が CR クロックモードで再起動するため , リセット要因ビットの読出しが可能になります。MCE ビットまたは SCE ビットを "0" に設定してリセット要因となったクロックを禁止することを推奨します。

クロック停止検出リセットが発生した後 , 内部 RAM の内容とリセットされていないレジスタ (初期値 "X") の内容は保証できません。

8.6.5 クロック停止検出リセットおよびスタンバイモード

■ スリープおよびタイマモード

クロック停止検出リセット機能はスリープモードとタイマモードで動作します。つまり , メインクロック停止リセットは , システムクロック CLKS1/CLKS2 がメインまたは PLL クロックモード (CKMR レジスタで示されている) に設定されている場合にラン , スリープ , およびタイマモードでアサートすることができます。また , サブクロック停止リセットは , システムクロック CLKS1/CLKS2 がサブクロックモードに設定されている場合にラン , スリープ , およびタイマモードでアサートすることができます。

■ ストップモード

すべての発振器 (CR 発振器を含む) はストップモードで禁止されるため , クロック停止検出を実行できません。

ストップモードで割込みがアサートされると , CKSR レジスタに指定されているクロックで MCU がランモードに切り換わります。ランモードへの移行は , CKSR: SC1S1, SC1S0 ビットで選択されたクロックが安定するまで延期されます。また , システムクロック 2 の動作は , CKSR: SC2S[1:0] ビットで選択されたクロックが安定するまで延期されます。

＜注意事項＞

クロックが安定している間に誤ってリセットがアサートされるのを防ぐために , この期間はクロック停止検出リセット機能が禁止されます。つまり , ストップモードにある間または発振器の安定時間内にクロックが故障した場合は , クロック停止検出リセットが発生しません。

この状況を回避するためにストップモードに切り換える場合は , 必ず両方のシステムクロックを CR クロックに設定してください (CKSR: SC1S[1:0] および SC2S[1:0] =00)。該当するクロックが再起動しなかったためにクロック準備完了モニタビットを設定できなかった場合は , CR クロックモードで起動した後に CLKS1/CLKS2 がメインクロック , PLL クロック , サブクロックに切り換わりません。ソフトウェアでは , CR クロックタイマを用いたタイムアウト条件でこの状況を検出する必要があります。

8.6.6 クロック停止検出リセットとウォッチドッグタイマ

クロック停止検出リセットは, ウォッチドッグタイマ用のソースとして使用されているメインまたはサブクロックが故障した場合にもアサートされます。 ただし, クロックが安定している間 (ストップモードの解除後) は, このリセット機能が禁止されます。

8.7 低電圧リセット機能の動作

低電圧リセット機能の動作について説明します。

8.7.1 低電圧リセットの機能

低電圧リセット機能では , 電源電圧 V_{CC} と内部で発生する基準電圧を比較する低電圧検出器を使用しています。この基準電圧は , CILCR: LVL[3:0] ビットでプログラマブルします。

リセット機能と低電圧検出器が低電圧リセット許可 (LVRE) ビットと低電圧検出器許可 (LVDE) ビットで許可されており , かつ V_{CC} が 選択された検出レベル V_{DL} を下回っている場合は , 電源リセット延長カウンタが初期化され , PRST フラグが設定されます。選択された検出レベル V_{DL} の仕様については , データシートを参照してください。

電源電圧が復帰すると , 「8.3 電源リセットおよび外部リセット後のスタートアップ」に記載されているように電源リセット延長カウンタが解除され , スタートアップが実行されます。

8.7.2 低電圧検出およびリセット回路の構成

■ 低電圧リセット

低電圧リセットは , 低電圧リセット許可 (LVRE) ビットと CILCR:LVL レベル選択ビットで制御されます。LVDE ビットを "0" に設定すると低電圧リセット機能が禁止され , "1" に設定すると許可されます。電源投入後または RSTX のアサート時に , このビットは常に "1" (リセット機能が許可される) に設定されます。CILCR: LVL レベル選択ビットは , リセットごとに "0000" にリセットされます (レベル 0)。

低電圧検出器を無効にする時は , 必ず LVRE ビットを "0" に設定してください。低電圧検出器に電源を投入する場合は , 低電圧検出器が安定時間に入ってから , LVRE を "1" に設定してください。このように操作しないと , 低電圧検出器から間違った値が出力されるため , 低電圧リセットが誤ってアサートされる可能性があります。低電圧検出器の安定時間の仕様については , データシートを参照してください。

低電圧リセット回路が内部ベクタモードで常に許可されているデバイスでは , "011" に設定されたモード端子に対して LVRE ビットの設定は影響を与えません。

■ 低電圧検出器

低電圧検出器は , 低電圧検出器許可 (LVDE) ビットで制御されます。このビットを "0" に設定すると低電圧検出器が禁止され , "1" に設定すると許可されます。電源投入後または RSTX のアサート時に , このビットは常に "1" (低電圧検出器が許可される) に設定されます。 レベル選択ビット (CILCR: LVL) はどんなリセットでも "0000"(レベル 0) にリセットされます。

LVDE を "0" に設定した場合は , 必ず同時またはその前に LVRE ビットも "0" に設定しなければなりません。低電圧検出器に電源を投入した場合は, 低電圧検出器の安定時間に入った後で LVRE を "1" に設定する必要があります。

低電圧リセット回路が内部ベクタモードで常に許可されているデバイスでは , "011" (低電圧検出器が常に許可される) に設定されたモード端子に対して LVDE ビットの設定は影響を与えません。

リセット後 , 検出レベルは常に “ レベル 0” に設定されます。このレベルでは , VCC が MCU の安全な動作に必要な最低値を下回った時に , リセットを発生させることができます。CILCR:LVL ビットを設定することで , 起動後にこのレベルを高くすることができます。これにより , システムが必要とする場合は , VCC レベルが既に高い低電圧リセットを発生させることが可能です。

■ 消費電流への影響

低電圧検出器を起動すると , 低電圧検出器から電流が流れます (詳細はデータシートを参照)。この電流の流れは , 選択された動作モードに関係しません。このことがスタンバイモードで受け付けられない場合は , スタンバイモードに切り換える前に低電圧検出器と低電圧リセットを禁止してください。この場合には低電圧リセット機能を使用できないことに注意してください。

低電圧リセット回路が内部ベクタモードで常に許可されているデバイスでは , モード端子が "011" に設定されている場合に低電圧検出器と低電圧リセットを禁止することができません。

■ 低電圧リセット機能を使用する際の制限事項

低電圧検出器を許可した後 , 低電圧検出器の安定時間に入っている間は出力値が不定義になります。したがって , この時間中に低電圧リセットを使用してはなりません。

低電圧リセット機能には , データシートに記載されているように AC 仕様に関する制限事項があります。

LVRE ビットと LVDE ビットで低電圧リセット機能を禁止することができるデバイスや動作モードでは , 低電圧リセット機能を使用した場合に電源投入後の安全なスタートアップを保証できません。この場合には , データシートに記載されているパワーオンリセットまたは外部リセットのいずれかを MCU の起動に適用する必要があります。

9. スタンバイモードおよび電圧レギュレータ制御回路

スタンバイモード制御回路の機能と動作および内部電圧レギュレータの制御について説明します。
レギュレータの制御は特にスタンバイモードでの電力消費を最適化できます。

9.1 CPU動作モードの概要

F^{2}MC-16FX ファミリには , 以下の CPU 動作モードがあります。

- ランモード
- スリープモード
- タイマモード
- ストップモード

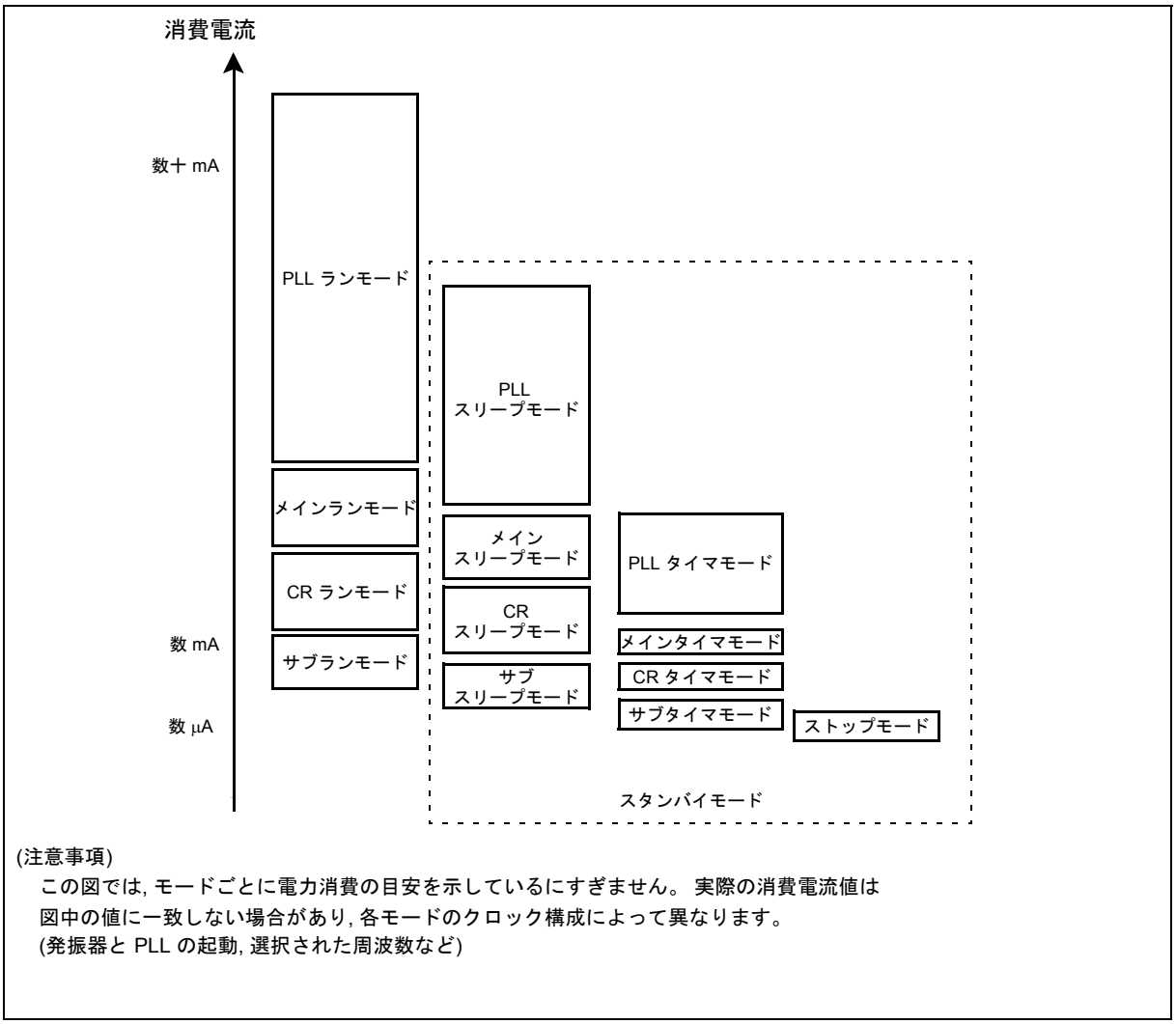
スリープモード, タイマモード, およびストップモードをまとめてスタンバイモードとよびます。 ストップモードでは, クロックがすべて停止します。 それ以外の CPU 動作モードでは, クロックソースと有効な発振器/PLL が選択可能です。

9.1.1 CPU動作モードと消費電流

CPU 動作モードと消費電流との関係を Figure 9-1 に示します。

動作モードの消費電力は内部電圧レギュレータの設定に影響されます。

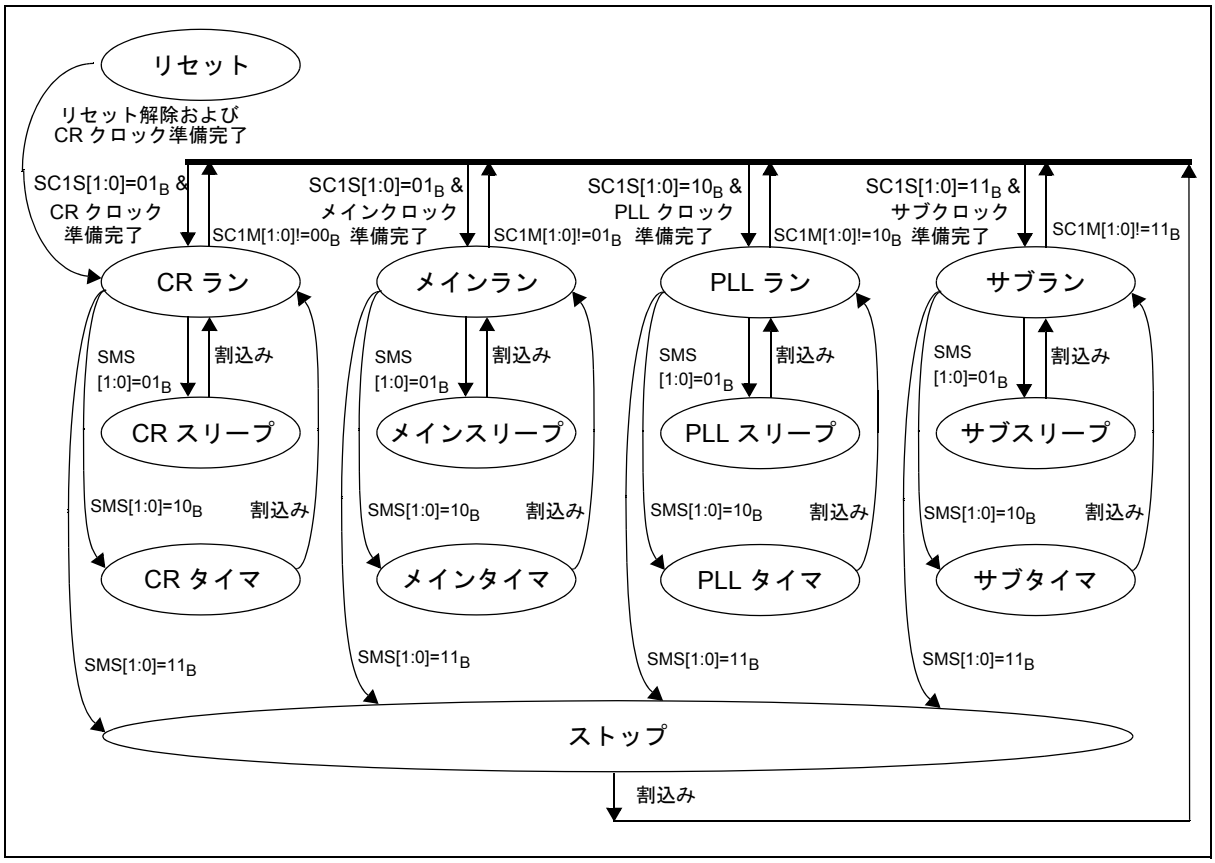
Figure 9-1. CPU動作モードと消費電流



9.1.2 状態遷移図

動作モードの状態遷移図を Figure 9-2 に示します。

Figure 9-2. モード切替え図



9.1.3 ランモード

ランモードは , 必ずシステムクロック 1 (CLKS1) 用のソースとして選択されたクロックで定義されます。システムクロック 2 セレクタで選択されたシステムクロック 2 (CLKS2) は , このランモードとは関係ありません。

したがって , 周辺クロック 2 (CLKP2) に接続されているモジュールは , 周辺クロック 2 がシステムクロック 2 から導出されているため , 別のクロックソースで動作することができます。

■ CR ランモード

このモードでは, 内部 CR 発振クロック CLKRC がシステムクロック 1 として使用され, CPU と大部分の周辺機能を動作させます。CR 発振器に対して 2 通りの周波数設定が可能です (公称 2 MHz または 100 kHz)。CPU を動作させるバスクロック (CLKB) と大部分の周辺機能を動作させる周辺クロック 1 (CLKP1) 用の独立したクロック分周器が CR クロックから供給され , 柔軟な周波数設定を可能にします。

メイン発振器 , PLL 逓倍回路 , およびサブ発振器の状態は , これらがシステムクロック 2 として使用されていない限り , それぞれ MCE, PCE および SCE ビットの設定に従います。

このモードは , リセット解除後に必ずアクティブになります。

■ メインランモード

このモードでは, メイン発振クロック CLKMC がシステムクロック 1 として使用され, CPU と大部分の周辺機能を動作させます。CPU を動作させるバスクロック (CLKB) と大部分の周辺機能を動作させる周辺クロック 1 (CLKP1) 用の独立したクロック分周器がメインクロックから供給され , 柔軟な周波数設定を可能にします。

PLL 逓倍回路, CR 発振器, およびサブ発振器の状態は, これらがシステムクロック 2 として使用されていない限り , それぞれ PCE, RCE, および SCE ビットの設定に従います。

■ PLL ランモード

このモードでは, メイン発振クロックを逓倍した PLL クロックがシステムクロック 1 として使用され, CPU と大部分の周辺機能を動作させます。CPU を動作させるバスクロック (CLKB) と大部分の周辺機能を動作させる周辺クロック 1 (CLKP1) 用の独立したクロック分周器が PLL出力クロックから供給され, 柔軟な周波数設定を可能にします。非変調 (CLKPLL) または変調 (CLKMOD) PLL クロックのどちらかを使用できます。

CR 発振器とサブ発振器の状態は , これらがシステムクロック 2 として使用されていない限り , それぞれ RCE ビットと SCE ビットの設定に従います。

■ サブランモード

このモードでは , サブ発振クロック CLKSC がシステムクロック 1 として使用され , CPU と大部分の周辺機能を動作させます。CPU を動作させるバスクロック (CLKB) と大部分の周辺機能を動作させる周辺クロック 1 (CLKP1) 用の独立したクロック分周器がサブクロックから供給され , 柔軟な周波数設定を可能にします。

メイン発振器 , PLL 逓倍回路 , および CR 発振器の状態は , これらがシステムクロック 2 として使用されていない限り , それぞれ MCE, PCE および RCE ビットの設定に従います。

＜注意事項＞

詳細は , 「第 6 章 クロック」と「第 7 章 クロックモジュレータ」を参照してください。

9.1.4 スリープモード

このモードに入ると , スタンバイ制御回路はバスクロック (CLKB) を停止します。これにより , CPU, 内部メモリ , DMA コントローラ , および外部バスインタフェースが無効になるため , 消費電力が削減されます。

■ CR スリープモード

CR スリープモードをアクティブにすると , CLKB クロックが CR クロックモードで停止します。周辺クロック 1 に接続されているリソースは CR クロックで動作します。

周辺クロック 2 に接続されているリソースは , システムクロック 2 セレクタで選択されたクロックで動作します。

メイン発振器 , PLL 逓倍回路 , およびサブ発振器の状態は , これらがシステムクロック 2 として使用されていない限り , それぞれ MCE, PCE および SCE ビットの設定に従います。

■ メインスリープモード

メインスリープモードをアクティブにすると , CLKB クロックがメインクロックモードで停止します。周辺クロック 1 に接続されているリソースはメインクロックで動作します。

周辺クロック 2 に接続されているリソースは , システムクロック 2 セレクタで選択されたクロックで動作します。

PLL 逓倍回路, CR 発振器, およびサブ発振器の状態は, これらがシステムクロック 2 として使用されていない限り , それぞれ PCE, RCE および SCE ビットの設定に従います。

■ PLL スリープモード

PLL スリープモードをアクティブにすると, CLKB クロックが PLL クロックモードで停止します。周辺クロック 1 に接続されているリソースは PLL クロックで動作します。

周辺クロック 2 に接続されているリソースは , システムクロック 2 セレクタで選択されたクロックで動作します。

CR 発振器とサブ発振器の状態は , これらがシステムクロック 2 として使用されていない限り , それぞれ RCE

ビットと SCE ビットの設定に従います。

■ サブスリープモード

サブスリープモードをアクティブにすると , CLKB クロックがサブクロックモードで停止します。周辺クロック 1 に接続されているリソースはサブクロックで動作します。

周辺クロック 2 に接続されているリソースは , システムクロック 2 セレクタで選択されたクロックで動作します。

メイン発振器 , PLL 逓倍回路 , および CR 発振器の状態は , これらがシステムクロック 2 として使用されていない限り , それぞれ MCE, PCE, および RCE ビットの設定に従います。

9.1.5 タイマモード

このモードに入ると , スタンバイ制御回路はシステムクロック CLKS1 と CLKS2 の供給を停止します。これにより , バスクロックとすべての周辺クロックが停止するため , 消費電力がさらに削減されます。この場合は , 発振器 , PLL, および対応するソースクロックタイマを除くすべての機能が停止します。

■ CR タイマモード

CR タイマモードでは , CR 発振器と CR クロックタイマが常にアクティブになります。ただし , PLL, メイン , およびサブ発振器の状態と対応するソースクロックタイマの状態は , それぞれ PCE, MCE, および SCE ビットの設定に従います。

■ メインタイマモード

メインタイマモードでは , メイン発振器とメインクロックタイマが常にアクティブになります。ただし , PLL, CR, およびサブ発振器の状態と対応するソースクロックタイマの状態は , それぞれ PCE, RCE, および SCE ビットの設定に従います。

■ PLL タイマモード

PLL タイマモードでは , メイン発振器 , PLL, およびメインクロックタイマが常にアクティブになります。ただし , CR およびサブ発振器の状態と対応するソースクロックタイマの状態は , それぞれ RCE および SCE ビットの設定に従います。

■ サブタイマモード

サブタイマモードでは , サブ発振器とサブクロックタイマが常にアクティブになります。ただし , PLL, メイン , および CR 発振器の状態と対応するソースクロックタイマの状態は , それぞれ PCE, MCE, および RCE ビットの設定に従います。

＜注意事項＞

PCE, MCE, SCE, および RCE ビットのうちの 1 つ以上が "1" に設定されている場合は , それぞれのタイマモードの動作が同じになることがあります。ただし , 割込みによるタイマモードからのウェイクアップ後にデバイスが移行するランモードは , このデバイスが 4 つのタイマモードのうちのどれに入っているかによって異なります。Table 9-4 を参照してください。

9.1.6 ストップモード

このモードに入ると , スタンバイ制御回路はすべての発振を停止します。すべての機能が稼動しない状態になります。

＜注意事項＞

ストップモードではすべての発振クロックが停止するため , 最低消費電力でデータを保持できます。

9.2 スタンバイモード制御レジスタ(SMCR)

このレジスタはスタンバイモードに切り換わり， タイマモードとストップモードで端子機能を制御し， 電圧レギュレータを低電力モードに設定します。

スタンバイモード制御レジスタ(SMCR)

スタンバイモード制御レジスタ (SMCR) の構成を Figure 9-3 に示します。

Figure 9-3. スタンバイモード制御レジスタ(SMCR)の構成

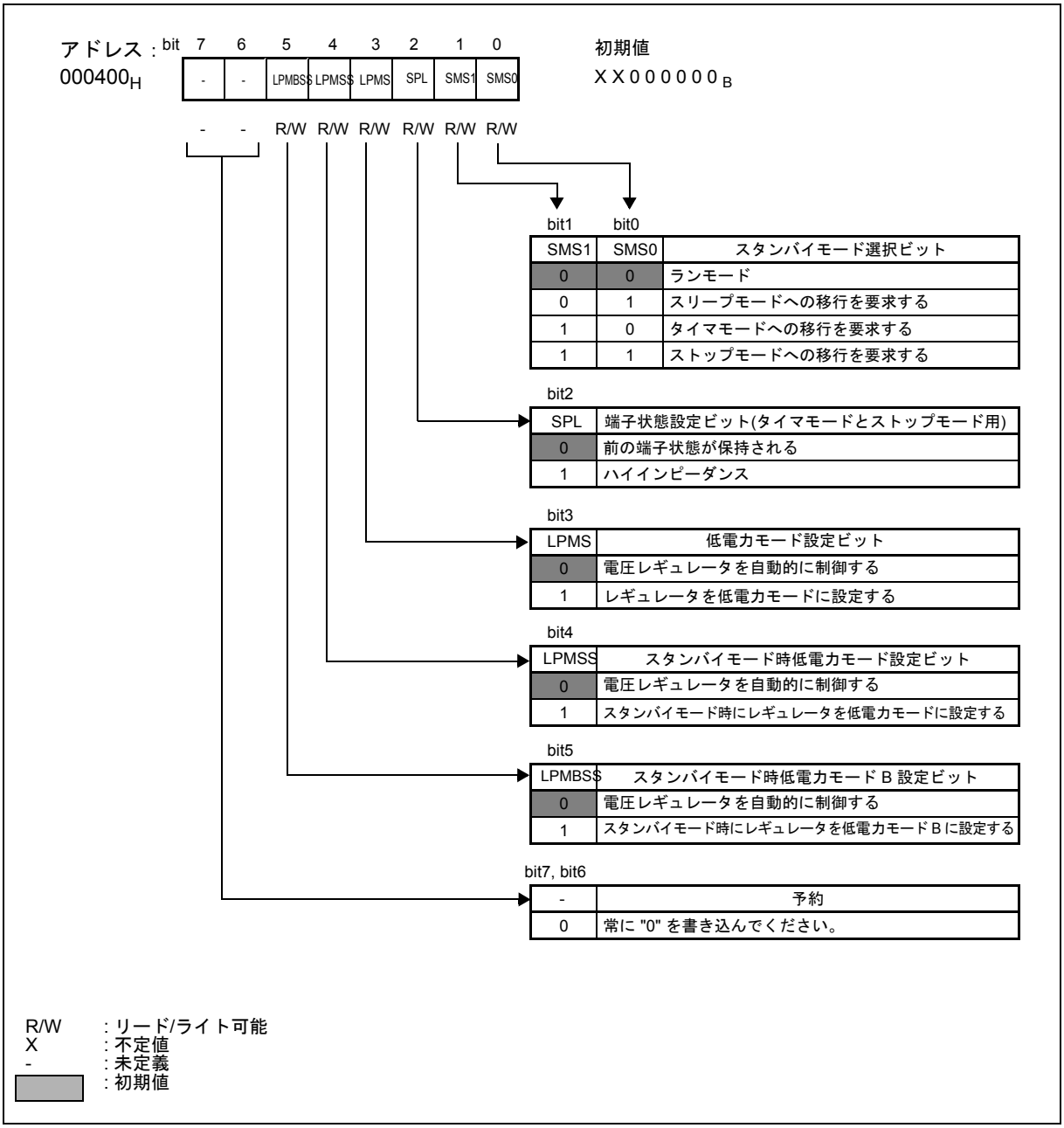


Table 9-1. スタンバイモード制御レジスタ(SMCR)の各ビットの機能説明

<table>
<tr><th colspan="2">ビット名</th><th>機能</th></tr>
<tr><td>bit 0 - bit 1</td><td>SMS1 および SMS0:
スタンバイモード
選択ビット</td><td>
• これらのビットは , 以下の表に従ってスタンバイモードへの切換えを要求します。
<table>
<tr><th colspan="2">bit1 bit0</th><th></th></tr>
<tr><th>SMS1</th><th>SMS0</th><th>スタンバイモード選択ビット</th></tr>
<tr><td>0</td><td>0</td><td>ランモード</td></tr>
<tr><td>0</td><td>1</td><td>スリープモードへの移行を要求する</td></tr>
<tr><td>1</td><td>0</td><td>タイマモードへの移行を要求する</td></tr>
<tr><td>1</td><td>1</td><td>ストップモードへの移行を要求する</td></tr>
</table>
• これらのビットに "01" を書き込むと , スリープモードへの切換えが要求されます。
• これらのビットに "10" を書き込むと , タイマモードへの切換えが要求されます。
• これらのビットに "11" を書き込むと , ストップモードへの切換えが要求されます。
• 3 つの値のいずれかを書き込むと, これらのビットがロックされます。別のスタンバイモードまたはランモードは選択できません。
• これらのビットは , 保留中のハードウェア割込み要求 (7 未満のレベル), NMI (利用可能な場合), およびリセットによって "00" にクリアされます。
• これらのビットの読出し値は, スタンバイモード移行要求の状態を示します。 "00" 以外の値が読み出された場合は選択されたスタンバイモードへの移行がまだ保留中であることを示し, "00" が読み出された場合は保留中のスタンバイモード移行要求が存在しないことを示します。
</td></tr>
<tr><td>bit 2</td><td>SPL:
端子状態設定ビット (タイマモードと
ストップモード用)</td><td>
• このビットの設定は , タイマモードとストップモードにのみ影響を与えます。
• このビットを "0" に設定すると , 外部端子のレベルが保持されます。
• このビットを "1" に設定すると , 外部端子の状態がハイインピーダンスに切り換わります。
• このビットはリセットごとに "0" に初期化されます。
</td></tr>
<tr><td>bit 3</td><td>LPMS:
低電力モード設定ビット</td><td>
• このビットは , いずれのモードでも電圧レギュレータを制御します。詳細については , 「9.7 電圧レギュレータの動作」を参照してください。
• このビットはリセットによりクリアされます。
• このビットが "0" に設定されると (初期値), 電圧レギュレータは自動的に制御されます (RC 発振器またはメイン発振器がアクティブな場合の高電力モードのレギュレータ)。
• このビットを "1" に設定すると , 電圧レギュレータは即座に低電力モード A に強制的に設定されます (レギュレータが常に低電力モード B に設定されるストップモードに移行しない限り)。これは , VRCR:LPMA[2:0] ビットによってレギュレータの出力電圧が定義されることを意味します。
• 「7.1 電圧レギュレータの動作モード」に記載された一定のモードでのみ , このビットの設定が許可されています。
• このビットを "0" に変更しなおした後は, より高い消費電流のモードに変更する前に, アプリケーションプログラムによって ~40µs の安定時間をとる必要があります。
</td></tr>
<tr><td>bit 4</td><td>LPMSS:
スタンバイモード用低電力モード設定ビット</td><td>
• このビットは , 各スタンバイモード (RC またはメインクロックでのスリープモードまたはタイマモード) で電圧レギュレータを制御します。詳細については , 「9.7 電圧レギュレータの動作」を参照してください。
• このビットはリセットによりクリアされます。
• このビットが "0" に設定されると (初期値), 電圧レギュレータは自動的に制御されます (RC 発振器またはメイン発振器がアクティブな場合の高電力モードのレギュレータ)。
• このビットを "1" に設定すると , スリープモードまたはタイマモードへの遷移後 , 電圧レギュレータは低電力モード A に強制的に設定されます。これは , VRCR:LPMA[2:0] ビットによってレギュレータの出力電圧が定義されることを意味します。
• 「9.7 電圧レギュレータの動作」に記載された一定のモードでのみ , このビットの設定が許可されています。
• このビットを "1" に設定した場合には, レギュレータを高電力モードに再び切り換えるために, スリープモードまたはタイマモードからのウェイクアップに最大40µsの遅延が自動的に発生します。
</td></tr>
</table>

Table 9-1. スタンバイモード制御レジスタ(SMCR)の各ビットの機能説明

ビット名		機能
bit 5	LPMBSS: スタンバイモード用低電力モードB 設定ビット	• このビットは, 各スタンバイモード (スリープモードまたはタイマモード) で電圧レギュレータを制御します。詳細については , 「9.7 電圧レギュレータの動作」を参照してください。 • このビットはリセットによりクリアされます。 • このビットを "0" に設定した場合には (初期値), 低電力モード B はストップモードのみで使用されます。ほかのすべてのモードでは , レギュレータは常に高電力モードまたは低電力モード A です。 • このビットを "1" に設定すると , 電圧レギュレータはスリープモードまたはタイマモードに遷移後 , 強制的に低電力モード B に設定されます (低電力モードがハードウェアまたは LPMS/LPMSS ビットによって起動された場合)。これは , VRCR: LPMB[2:0] ビットによってレギュレータの出力電圧が定義されることを意味します。 • 「9.7 電圧レギュレータの動作」に記載された一定のモードでのみ , このビットの設定が許可されています。 • このビットを "1" に設定した場合には , レギュレータを高電力モードまたは低電力モード A に再び切り換えるために, スリープモードまたはタイマモードからのウェイクアップに最大 40µs の遅延が自動的に発生します。 (注意事項) 本LPMBSSビットは"0"固定で使用してください。CPU動作モードがストップモード時のみ電圧レギュレータの設定を低電力モードBに切換え可能です。
bit 6 ～ bit 7	予約	• 常に "0" を書き込んでください。 • これらのビットの読出し値は未定義です。 • このレジスタへのリードモディファイライト系命令は無効です。

＜注意事項＞

アクティブな外部割込み入力端子 , NMI 入力端子 , および外部バスの HRQ ホールド要求入力端子を除くすべての入力端子はタイマモードとストップモードで自動的に無効になります。PIE ポート入力許可ビットの設定は , タイマモードとストップモードでは機能しません。

9.3 電圧レギュレータ制御レジスタ(VRCR)

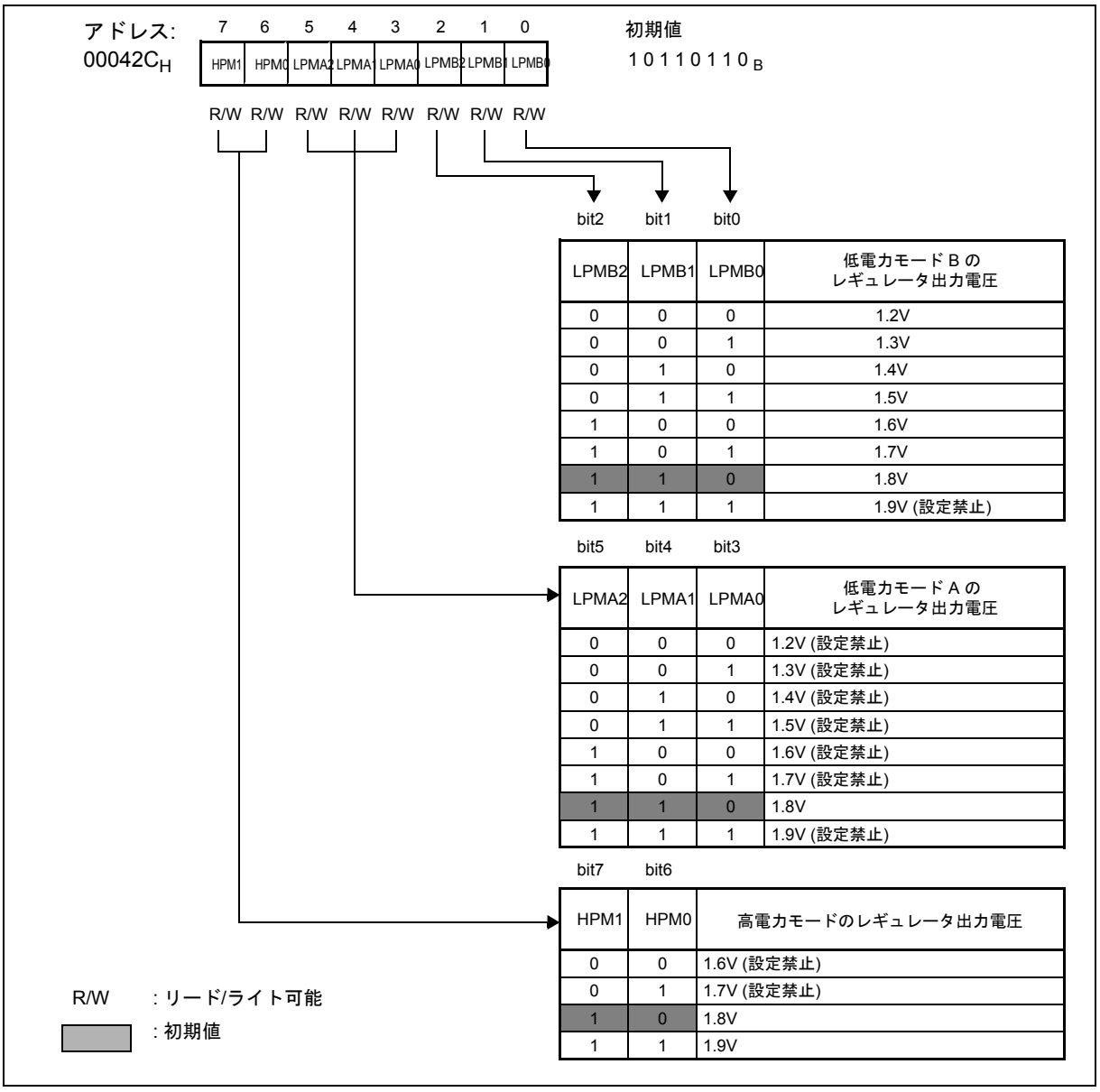
電圧レギュレータ制御レジスタは, 内部電圧レギュレータの各種動作モードの出力電圧を定義します。

電圧レギュレータ制御レジスタ(VRCR)

Figure 9-4 では電圧レギュレータ制御レジスタ (VRCR) の構成を示し, Table 9-2 では各ビットの機能について説明します。

詳細は , 「9.7 電圧レギュレータの動作」を参照してください。

Figure 9-4. 電圧レギュレータ制御レジスタ(VRCR)の構成



LPMB2	LPMB1	LPMB0	低電力モードBのレギュレータ出力電圧
0	0	0	1.2V
0	0	1	1.3V
0	1	0	1.4V
0	1	1	1.5V
1	0	0	1.6V
1	0	1	1.7V
1	1	0	1.8V
1	1	1	1.9V (設定禁止)

LPMA2	LPMA1	LPMA0	低電力モードAのレギュレータ出力電圧
0	0	0	1.2V (設定禁止)
0	0	1	1.3V (設定禁止)
0	1	0	1.4V (設定禁止)
0	1	1	1.5V (設定禁止)
1	0	0	1.6V (設定禁止)
1	0	1	1.7V (設定禁止)
1	1	0	1.8V
1	1	1	1.9V (設定禁止)

HPM1	HPM0	高電力モードのレギュレータ出力電圧
0	0	1.6V (設定禁止)
0	1	1.7V (設定禁止)
1	0	1.8V
1	1	1.9V

Table 9-2. 電圧レギュレータ制御レジスタ(VRCR)の各ビットの機能説明 (Sheet 1 of 2)

<table>
<tr><th colspan="2">ビット名</th><th>機能</th></tr>
<tr><td>bit 0 ～ bit 2</td><td>LPMB2 ～ LPMB0:
低電力モード B 選択ビット</td><td>これらのビットは , 低電力モード B のレギュレータ出力電圧を次の表のように選択します。
<table>
<tr><th></th><th></th><th></th><th></th></tr>
<tr><td>bit2</td><td>bit1</td><td>bit0</td><td></td></tr>
<tr><td>LPMB2</td><td>LPMB1</td><td>LPMB0</td><td>低電力モード B の
レギュレータ出力電圧</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1.2V</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1.3V</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1.4V</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1.5V</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1.6V</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1.7V</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1.8V</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1.9V (設定禁止)</td></tr>
</table>
• どのリセットでも , これらのビットは "110"(1.8V) に初期化されます。
• ストップモードでは , レギュレータの低電力モード B が使用されます。特定の条件下で SMCR:LPMBSS ビットにより , 低電力モード B を選択することもできます。
(注意事項)
1.9Vの設定は使用できません。</td></tr>
<tr><td>bit 3 ～ bit 5</td><td>LPMA2 ～ LPMA0:
低電力モード A 選択ビット</td><td>これらのビットは , 低電力モード A のレギュレータ出力電圧を次の表のように選択します。
<table>
<tr><th></th><th></th><th></th><th></th></tr>
<tr><td>bit5</td><td>bit4</td><td>bit3</td><td></td></tr>
<tr><td>LPMA2</td><td>LPMA1</td><td>LPMA0</td><td>低電力モード A の
レギュレータ出力電圧</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1.2V(設定禁止)</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1.3V(設定禁止)</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1.4V(設定禁止)</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1.5V(設定禁止)</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1.6V(設定禁止)</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1.7V(設定禁止)</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1.8V</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1.9V(設定禁止)</td></tr>
</table>
• どのリセットでも , これらのビットは "110"(1.8V) に初期化されます。
• メインおよび CR の両方の振動子が停止している場合は , サブモード (サブ RUN, サブスリープ , サブタイマモード) ではレギュレータの低電力モード A が使用されます。特定の条件下で SMCR:LPMS と LPMSS ビットにより , 低電力モード B を選択することもできます。
(注意事項)
これらのビットは初期値のまま変更せずに使用してください。</td></tr>
</table>

Table 9-2. 電圧レギュレータ制御レジスタ(VRCR)の各ビットの機能説明 (Sheet 2 of 2)

<table>
<tr><th colspan="2">ビット名</th><th>機能</th></tr>
<tr><td>bit 6, bit 7</td><td>HPM1 , HPM0:
高電力モード選択ビット</td><td>これらのビットは , 高電力モードのレギュレータ出力電圧を次の表のように選択します。
<table>
<tr><th>bit7
HPM1</th><th>bit6
HPM0</th><th>高電力モードのレギュレータ出力電圧</th></tr>
<tr><td>0</td><td>0</td><td>1.6V(設定禁止)</td></tr>
<tr><td>0</td><td>1</td><td>1.7V(設定禁止)</td></tr>
<tr><td>1</td><td>0</td><td>1.8V</td></tr>
<tr><td>1</td><td>1</td><td>1.9V</td></tr>
</table>
• どのリセットでも , これらのビットは "10"(1.8V) に初期化されます。
• CR 発振 またはメイン発振のいずれかの動作中は , 常にレギュレータの高電力モードが標準動作モードとなります (低電力モードが SMCR:LPMS または LPMSS ビットで選択されていない限り)。
• 1.9V のコア電圧でのみ許可されるクロック周波数への変更の前に , フラッシュインタフェースの FMCS:RD19V ビットをともにこれらのビットを "11"(1.9V) に設定する必要があります。
(注意事項)
1.6V および 1.7V の設定使用できません。</td></tr>
</table>

9.4 スタンバイモード

スタンバイモードには, スリープモード (CR スリープ, メインスリープ, PLL スリープ, サブスリープ), タイマモード (CR タイマ, メインタイマ, PLL タイマ, サブタイマ), およびストップモードがあります。

スタンバイモードにある間の動作状態

それぞれのスタンバイモードにある間のクロック, CPU, 周辺機能, および外部端子の状態を Table 9-3 に示します。

Table 9-3. スタンバイモードにある間の動作状態

スタンバイモード		切換えの条件	CR クロック	メインクロック	PLL クロック	サブクロック	バスクロック	周辺クロック1	周辺クロック2	端子	解除イベント
スリープモード	CR スリープモード	SC1M=00 SMS=01	アクティブ	MCEビットとSC2Mビットに従う	PCE, MCE, およびSC2Mビットに従う	SCEビットとSC2Mビットに従う	停止	CRクロック	SC2M ビットに従う	アクティブ	電源リセット, 外部リセット, ウォッチドッグリセット, クロック停止リセット, 割込み
	メインスリープモード	SC1M=01 SMS=01	RCEビットとSC2Mビットに従う	アクティブ	PCEビットとSC2Mビットに従う	SCEビットとSC2Mビットに従う		メインクロック			
	PLL スリープモード	SC1M=10 SMS=01	RCEビットとSC2Mビットに従う	アクティブ	アクティブ	SCEビットとSC2Mビットに従う		PLLクロック			
	サブスリープモード	SC1M=11 SMS=01	RCEビットとSC2Mビットに従う	MCEビットとSC2Mビットに従う	PCE, MCE, およびSC2Mビットに従う	アクティブ		サブクロック			
タイマモード	CR タイマモード	SC1M=00 SMS=10	アクティブ	MCEビットとSC2Mビットに従う	PCE, MCE, およびSC2Mビットに従う	SCEビットとSC2Mビットに従う		停止(*1)		SPL ビットに従う (Hi-Z または最後の値が保持される)	
	メインタイマモード	SC1M=01 SMS=10	RCEビットとSC2Mビットに従う	アクティブ	PCEビットとSC2Mビットに従う	SCEビットとSC2Mビットに従う					
	PLL タイマモード	SC1M=10 SMS=10	RCEビットとSC2Mビットに従う	アクティブ	アクティブ	SCEビットとSC2Mビットに従う					
	サブタイマモード	SC1M=11 SMS=10	RCEビットとSC2Mビットに従う	MCEビットとSC2Mビットに従う	PCE, MCE, およびSC2Mビットに従う	アクティブ					
ストップモード	ストップモード	SMS=11	停止					停止			電源リセット, 外部リセット, 割込み

*: 対応するクロックが停止していない場合は, CR クロックタイマ, メインクロックタイマ, およびサブクロックタイマが動作します。
SC1M: クロックモニタレジスタ (CKMR) のシステムクロック 1 モニタビット
SC2M: クロックモニタレジスタ (CKMR) のシステムクロック 2 モニタビット
SMS: スタンバイモード制御レジスタ (SMCR) のスタンバイモード選択ビット
SPL: スタンバイモード制御レジスタ (SMCR) の端子状態設定ビット
Hi-Z: ハイインピーダンス

9.4.1 スリープモード(CRスリープ, メインスリープ, PLLスリープ, およびサブスリープモード)

このモードでは, ほかのコンポーネントが動作し続けている間, CPU 動作クロック (バスクロック: CLKB) が停止します。 スリープモードに移行するには, スタンバイモード制御レジスタ (SMCR) への書込みを行います。(SC1M システムクロック 1 モニタビットに示されている) 現在アクティブなクロックモードに応じて, MCU が CR スリープ, メインスリープ, PLL スリープ, またはサブスリープのいずれかのモードに切り換わります。

9.4.1.1 スリープモードの機能

■ CPU, 内部メモリ, およびデータ保持

スリープモードでは , CPU と内部メモリに供給しているクロック (CLKB) が停止します。ただし , 専用レジスタ (アキュムレータなど) と内部 RAM の内容は保持されます。

■ リソース

リソースクロック CLKP1 と CLKP2 はスリープモードでアクティブになり , アクティブなリソースは最終の設定で動作します。

■ ソースクロックタイマ

発振器と対応するソースクロックタイマ (CR クロックタイマ , メインクロックタイマ , およびサブクロックタイマ) はスリープモードの影響を受けません。これらの発振器とタイマは , スリープモードに移行前の設定に従って動作を続けます。

■ リセットと割込み

リセットと割込みはスリープモードで有効となるため , スリープモードの解除に使用できます。

■ 外部バスとホールド機能

外部バスはスリープモードで停止しますが , 外部バスホールド機能はアクティブになります (ホールド機能が許可されている場合は , 外部バス端子を HRQ 端子で制御されている Hi-Z に設定します)。

■ 端子の状態

スリープモード中は , すべての端子 (外部バスに使用されているものを除く) でスリープモード移行前の機能が保持されます。

9.4.1.2 スリープモードへの切換え

スリープモードへの切換えを要求するには , スタンバイモード制御レジスタ (SMCR) の SMS ビットに "01" を書き込みます。クロックモニタレジスタ (CKMR) の SC1M ビットに示されている現在のクロックモードに応じて , 対応するスリープモードへの移行が要求されます (CR ラン -> CR スリープ, メインラン -> メインスリープ, PLL ラン -> PLL スリープ , サブラン -> サブスリープモード)。スリープモードへの移行は , 要求から数クロック周期以内に行われます。

＜注意事項＞

- SMS 書込み命令の次の命令が (スリープモードへの移行前ではなく) スリープモードからのウェイクアップ後に実行されているかどうかを確認するには , "01" に設定した後で SMS ビットをポーリングします。次の命令に分岐するのは , SMS ビットが "00" にクリアされている場合だけです。"01" が読み出された場合は , スリープモードへの移行がまだ行われていない (移行要求がまだ保留中である) ことを示します。
- (CKSR レジスタのシステムクロック 1 選択 "SC1S" ビットまたはシステムクロック 2 選択 "SC2S" ビットの値を書き換えることで) クロックモードを切り換えた後にスリープモードに移行する場合は , クロックモードの移行が SMS ビットの書込み前に行われているかどうかを確認

してください。クロックモードの移行は , クロック安定および同期メカニズムによって延期されます。必ず SMS ビットを設定する前に CKMR レジスタの SC1M/SC2M (システムクロック 1/2 モニタ) ビットを読み出してクロックモードの正否を確認してください。
SC1S1, SC1S0 / SC2S1, SC2S0 ビットの設定を変更するのは , SMS ビットが "00" (スタンバイモード移行要求が保留中でない) に設定されている場合に限ります。

■ スリープモードに切り換える際の割込み要求

ハードウェア割込みが保留中の場合 , SMS ビットへの書込みは無効です。

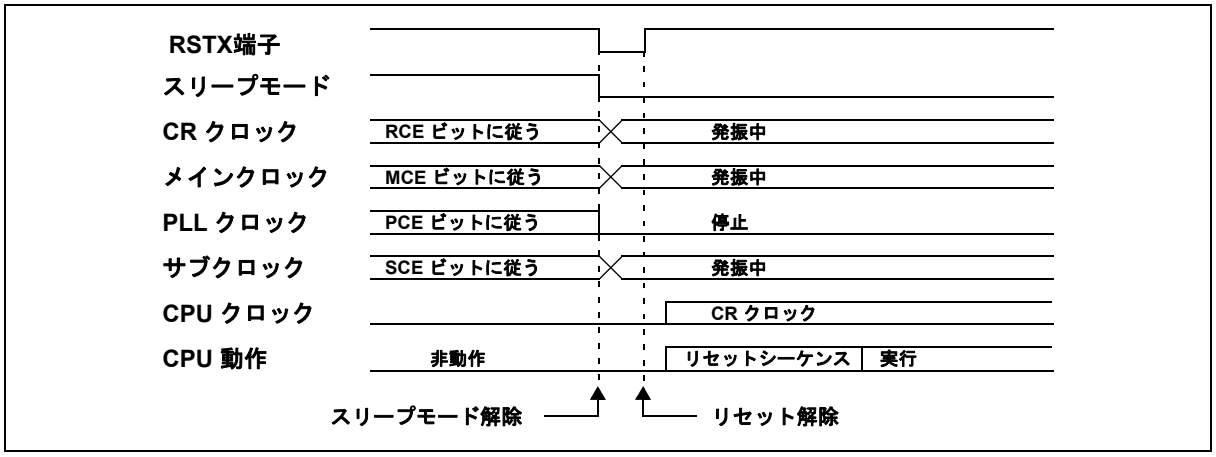
9.4.1.3 スリープモードの解除

リセットまたは割込みが発生すると , スタンバイモード制御回路によってスリープモードが解除されます。概要については , Table 9-4 を参照してください。

■ リセットによる復帰

リセット (電源 , 外部 , クロック停止検出 , またはウォッチドッグリセット) が発生した場合は , 最後に選択されたクロックモードに関係なく MCU が CR ランモードに切り換わります。

Figure 9-5. 外部リセットによるスリープモードの解除



■ 割込みによる復帰

スリープモードにある間に NMI 割込みまたはレベル 7 より高い別の割込み要求が周辺回路から発生した場合 , MCU はスリープモードを出て対応するランモードに復帰します (CR スリープ -> CR ラン , メインスリープ -> メインラン , PLL スリープ -> PLL ラン , サブスリープ -> サブラン)。スリープモードが解除された後の割込みは通常の割込みとして処理されます。

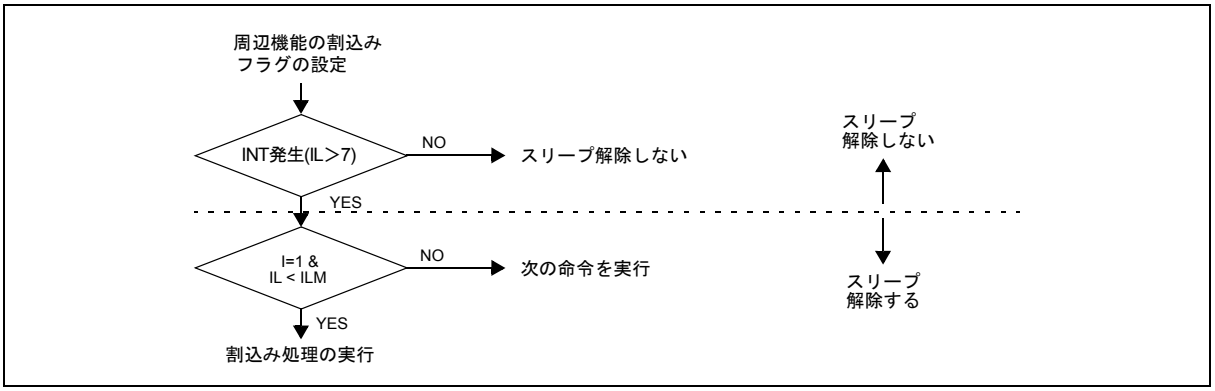
コンディションコードレジスタ (CCR)の I フラグ, 割込みレベルマスクレジスタ (ILM), および割込み制御レジスタ (ICR) の設定に従って割込みが受け付けられた場合 , CPU はパイプラインに格納された最後の命令を終了した後に割込みルーチンを実行します。この割込みから復帰した後, CPU は割込みルーチンに入る直前に実行した命令の次の命令を実行します。

この割込みが受け付けられなかった場合 , CPU はスリープモードに入る直前に実行した命令の次の命令を実行します。

電圧レギュレータはスリープモード時に手動で低電力モードに設定された場合は(SMCR:LMPSS ビットで設定), ウェイクアップを標準 40µs 延期します。

(NMI ではなく) 割込みが発生した場合のスリープモードの解除を Figure 9-6 に示します。

Figure 9-6. 割込みによるスリープモードの解除



9.4.2 タイマモード(CRタイマ, メインタイマ, PLLタイマ, サブタイマモード)

タイマモードでは, 発振器, PLL, およびソースクロックタイマを除くすべての機能が停止します。 タイマモードに移行するには,スタンバイモード制御レジスタ (SMCR)への書込みを行います。 CKMR：SC1Mビットに示されている現在のクロックモードに従って, MCU が CR タイマ, メインタイマ, PLL タイマ, またはサブタイマのいずれかのモードに切り換わります。

9.4.2.1 タイマモードの機能

■ CPU, 内部メモリ, およびデータ保持

タイマモードでは , CPU と内部メモリ (CLKB) に供給しているクロックが停止します。ただし , アキュムレータなど専用レジスタと内部 RAM の内容は保持されます。

■ リソース

リソースクロック CLKP1 と CLKP2 はタイマモードで停止し , すべてのリソースは最後の状態のまま停止します。

■ ソースクロックタイマ

発振器と対応するソースクロックタイマ (CR クロックタイマ, メインクロックタイマ, およびサブクロックタイマ) はタイマモードの影響を受けません。これらの発振器とタイマは , タイマモード移行前の設定に従って動作を続けます。

■ リセットと割込み

リセットと割込みはタイマモードで有効となるため , タイマモードの解除に使用できます。

■ 外部バスホールド機能

タイマモード中は外部バスは停止し , 外部バスホールド機能が有効になります (ホールド機能が許可されている場合は , 外部バス端子を HRQ 端子で制御されている Hi-Z に設定します)。

■ 端子状態

タイマモード中の外部端子の状態は , スタンバイモード制御レジスタ (SMCR) の SPL ビットによって , タイマモードに切り換わる直前の状態を保持するか , ハイインピーダンス状態にするかを設定できます。

9.4.2.2 タイマモードへの切換え

タイマモードへの切換えを要求するには , スタンバイモード制御レジスタ (SMCR) の SMS ビットに "10" を書き込みます。クロックモニタレジスタ (CKMR) の SC1M ビットに示されている現在のクロックモードに応じて, 対応するタイマモードへの移行が要求されます (CR ラン -> CR タイマ , メインラン -> メインタイマ , PLL ラン -> PLL タイマ , サブラン -> サブタイマモード)。タイマモードへの切換えは , 要求から数クロック周期以内に行われます。

＜注意事項＞

- SMS 書込み命令の次の命令が (タイマモードへの移行前ではなく) タイマモードからのウェイクアップ後に実行されているかどうかを確認するには , "10" に設定した後で SMS ビットをポーリングします。次の命令に分岐するのは , SMS ビットが "00" にクリアされている場合だけです。"10" が読み出された場合は , タイマモードへの移行がまだ行われていない (移行要求がまだ保留中である) ことを示します。
- (CKSR レジスタのシステムクロック 1 選択 "SC1S" ビットまたはシステムクロック 2 選択 "SC2S" ビットの値を書き換えることで) クロックモードを切り換えた後にタイマモードに移行する場合は , クロックモードの移行が SMS ビットの書込み前に行われているかどうかを確認してください。クロックモードの移行は , クロック安定および同期メカニズムによって延期されま

す。必ず SMS ビットを設定する前に CKMR レジスタの SC1M/SC2M (システムクロック 1/2 モニタ) ビットを読み出してクロックモードの正否を確認してください。
SC1S1, SC1S0 / SC2S1, SC2S0 ビットの設定を変更するのは , SMS ビットが "00" (スタンバイモード移行要求が保留中でない) に設定されている場合に限ります。

■ タイマモードに切り換える際の割込み要求

ハードウェア割込みが保留中の場合 , SMS ビットへの書込みは無効です。

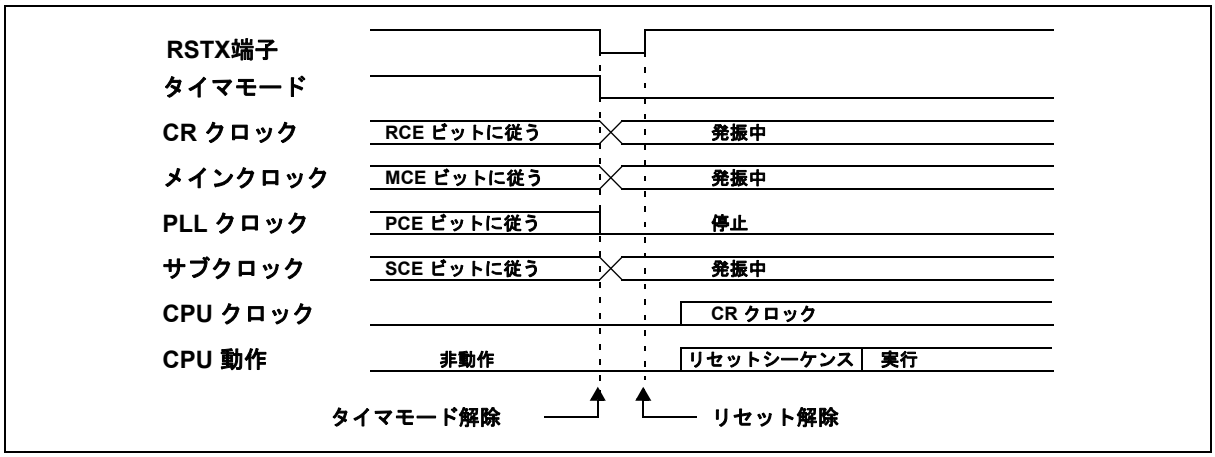
9.4.2.3 タイマモードからの復帰

リセットまたは割込みが発生すると , スタンバイモード制御回路によってタイマモードが解除されます。概要については , Table 9-4 を参照してください。

■ リセットによる復帰

リセット (電源 , 外部 , クロック停止検出 , またはウォッチドッグリセット) が発生した場合は , 最後に選択されたクロックモードに関係なく MCU が CR ランモードに切り換わります。

Figure 9-7. 外部リセットによるタイマモードの解除



■ 割込みによる復帰

NMI 割込みがアサートされた場合やタイマモード中にソースクロックタイマまたは外部割込みから割込みレベル (IL) が 7 より強い割込み要求が発生した場合は , タイマモードが解除され , MCU が対応するランモードに復帰します (CR タイマ -> CR ラン , メインタイマ -> メインラン , PLL タイマ -> PLL ラン , サブタイマ -> サブラン)。タイマモードが解除された後の割込みは通常の割込みとして処理されます。

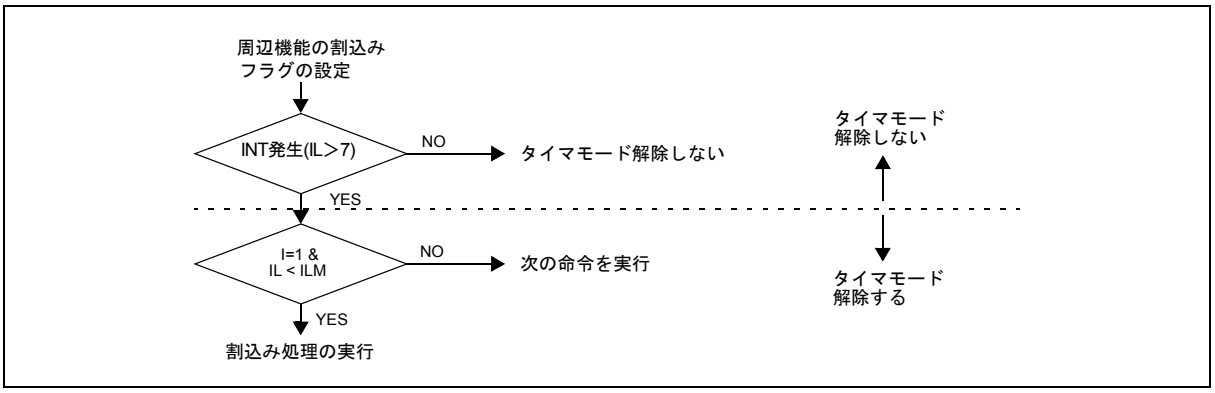
コンディションコードレジスタ (CCR)の I フラグ, 割込みレベルマスクレジスタ (ILM), および割込み制御レジスタ (ICR) の設定に従って割込みが受け付けられた場合 , CPU はパイプラインに格納された最後の命令を終了した後に割込みルーチンを実行します。この割込みから復帰した後, CPU は割込みルーチンに入る直前に実行した命令の次の命令を実行します。

この割込みが受け付けられなかった場合 , CPU はタイマモードに入る直前に実行した命令の次の命令を実行します。

タイマモード中に , 電圧レギュレータを手動で低電力モードに設定した場合 (SMCR: LMPSS ビットで設定), ウェイクアップは標準で 40µs 遅延されます。

(NMI ではなく) 割込みが発生した場合のタイマモードの解除を Figure 9-6 に示します。

Figure 9-8. 割込みによるタイマモードの解除



9.4.3 ストップモード

ストップモードでは,すべての発振器が停止し,すべての機能が稼動しない状態になるため,最低消費電力でデータを保持できます。

9.4.3.1 ストップモードの機能

■ CPU, 内部メモリ, およびデータ保持

ストップモードでは , CPU と内部メモリ (CLKB) を供給しているクロックが停止します。ただし , アキュムレータなど専用レジスタと内部 RAM の内容は保持されます。

■ リソース

リソースクロック CLKP1 と CLKP2 が停止し , すべてのリソースが最後の状態のまま停止します。

■ ソースクロックタイマとクロック

すべてのソースクロックタイマ (CR クロックタイマ , メインクロックタイマ , およびサブクロックタイマ) だけでなくすべての発振器と PLL も停止してクリアされます。すべての発振器準備完了フラグと PLL 準備完了フラグがクリアされます。

■ リセットと割込み

リセットと割込みはストップモードで有効となるため , ストップモードの解除に使用できます。

■ 外部バスホールド機能

ストップモード中は外部バスは停止し , 外部バスホールド機能が有効になります (ホールド機能が許可されている場合は , 外部バス端子を HRQ 端子で制御されている Hi-Z に設定します)。

■ 端子状態

ストップモード中の外部端子の状態は , スタンバイモード制御レジスタ (SMCR) の SPL ビットによって , ストップモードに切り換わる直前の状態を保持するか , ハイインピーダンス状態にするかを設定できます。

9.4.3.2 ストップモードへの切換え

現在のクロックモードに関係なくストップモードへの切換えを要求するには , スタンバイモード制御レジスタ (SMCR) の SMS ビットに "11" を書き込みます。ストップモードへの移行は , 要求から数クロック周期以内に行われます。

＜注意事項＞

- SMS 書込み命令の次の命令が (ストップモードへの移行前ではなく) ストップモードからのウェイクアップ後に実行されているかどうかを確認するには , "11" に設定した後で SMS ビットをポーリングします。次の命令に分岐するのは , SMS ビットが "00" にクリアされている場合だけです。"11" が読み出された場合は , ストップモードへの移行がまだ行われていない (移行要求がまだ保留中である) ことを示します。
- SMS ビットを "11" (ストップモード) に設定することと同時に , (CKSR レジスタの SC1S[1:0] またはSC2S[1:0] システムクロック選択ビットの値を書き換えて) クロックモードを切り換えることができます。新しいクロック設定は , ストップモードから復帰した直後に有効となります。スリープモードやタイマモードとは異なり , CKMR レジスタの SC1M または SC2M (システムクロックモニタ) ビットを読み出す必要も SMS ビットを設定する前にクロックモードの移行を待機する必要もありません。ただし , SMS ビットへの書込み後 (スタンバイモード移行要求が保留中の場合) は , SC1S[1:0] または SC2S[1:0] ビットの設定を変更しないでください。
  外部クロックが故障した場合でも信頼のおける MCU スタートアップを保証するために , ストップモードに切り換える際には両方のシステムクロックを CR クロックに設定する必要があります。このことは , クロック停止検出リセット機能を使用している場合に必須となります。
  SC1S[1:0] または SC2S[1:0] ビットを存在しないクロック (例えば , デバイスにサブ発振器が備

わっていない場合やサブ発振器端子に水晶 / 振動子が接続されていない場合に「サブクロックモード」) に設定しないでください。

■ ストップモードに切り換える際の割込み要求

ハードウェア割込みが保留中の場合 , SMS ビットへの書込みは効果がありません。

9.4.3.3 ストップモードからの復帰

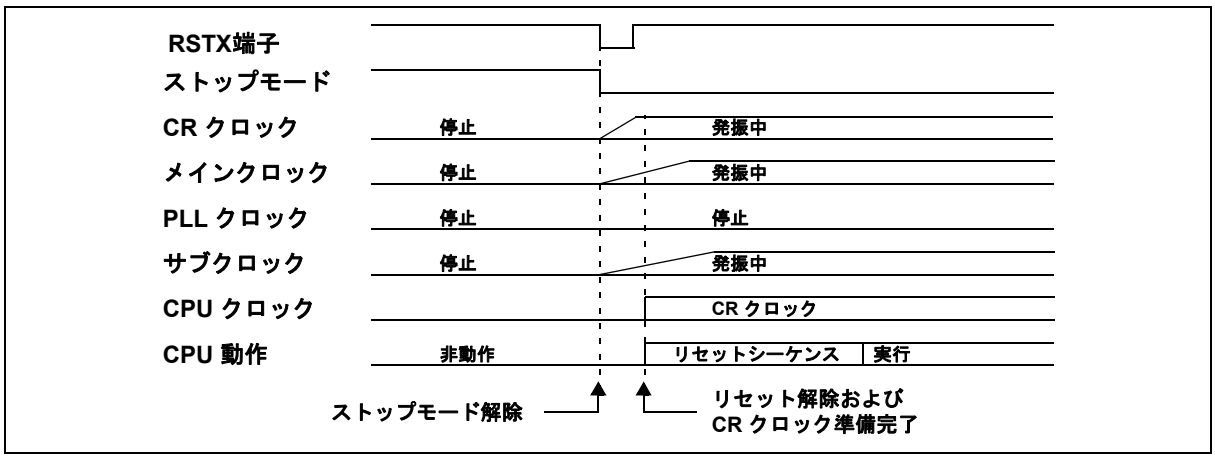
リセットまたは割込みが発生すると , スタンバイモード制御回路によってストップモードが解除されます。概要については , Table 9-4 を参照してください。

■ リセットによる復帰

電源リセットまたは外部リセットが発生した場合は , 最後に選択されたクロックモードに関係なく MCU が CR ランモードに切り換わります。

それ以外のリセット (クロック停止 , ソフトウェア , ウォッチドッグ) はストップモード中は動作しません。

Figure 9-9. 外部リセットによるストップモードの解除



■ 割込みによる復帰

NMI 割込みがアサートされた場合やストップモード中に外部割込みから割込みレベル (IL) が 7 より強い割込み要求が発生した場合は, ストップモードが解除され, MCU が SC1S[1:0]システムクロック 1 選択ビットで選択されたランモードに復帰します (該当するクロック安定待ち時間に入った後)。CKSR レジスタの構成に従って発振が許可されます。ストップモードが解除された後の割込みは通常の割込みとして処理されます。

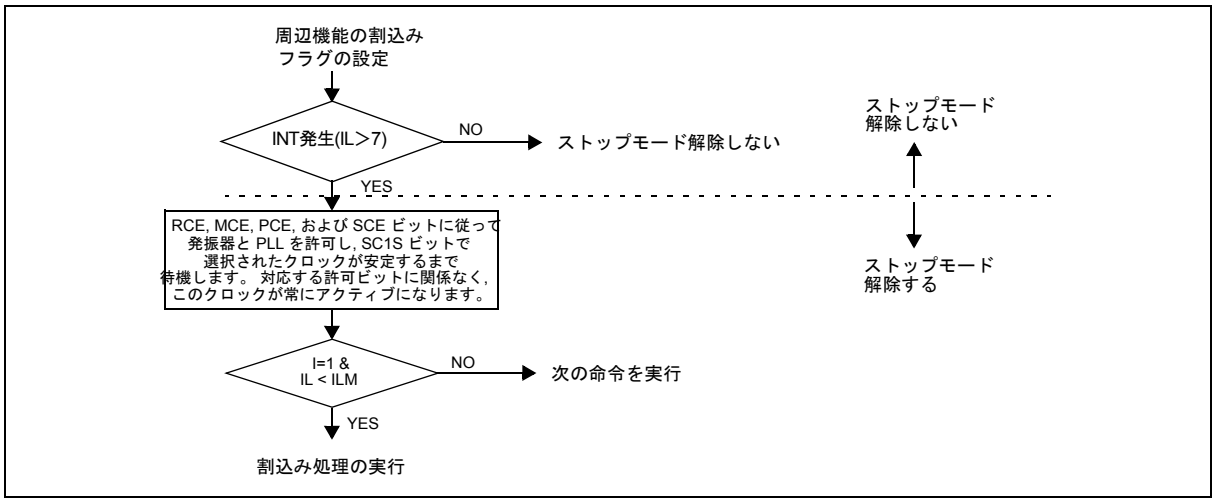
コンディションコードレジスタ (CCR)の I フラグ, 割込みレベルマスクレジスタ (ILM), および割込み制御レジスタ (ICR) の設定に従って割込みが受け付けられた場合 , CPU はパイプラインに格納された最後の命令を終了した後に割込みルーチンを実行します。この割込みから復帰した後, CPU は割込みルーチンに入る直前に実行した命令の次の命令を実行します。

この割込みが受け付けられなかった場合 , CPU はストップモードに入る直前に実行した命令の次の命令を実行します。

ストップモード中に電圧レギュレータを手動で低電力モードに設定した場合(SMCR:LMPSS ビットで設定), ウェイクアップは標準で 40µs 遅延されます。

(NMI ではなく) 割込みが発生した場合のストップモードの解除を Figure 9-10 に示します。

Figure 9-10. 割込みによるストップモードの解除



9.5 モード切替表と動作状態

F^{2}MC-16FX ファミリのモード切換え表を Table 9-4 に示し, それぞれの動作モードでの動作状態を Table 9-5 に示します。

モード切替表

Table 9-4. モード切替表

(SC1M ビットに示されている) 現行モード	リセット発生後	割込み発生後	CPU コマンド実行後
CRラン	RC Run	CRラン	メインラン, PLL ラン, サブラン, CR スリープ, CR タイマ, ストップ
メインラン		メインラン	CR ラン, PLL ラン, サブラン, メインスリープ, メインタイマ, ストップ
PLLラン		PLLラン	CR ラン , メインラン , サブラン , PLL スリープ, PLL タイマ, ストップ
サブラン		サブラン	CR ラン , メインラン , PLL ラン , サブスリープ, サブタイマ, ストップ
CRスリープ		CRラン	-
メインスリープ		メインラン	-
PLLスリープ		PLLラン	-
サブスリープ		サブラン	-
CRタイマ		CRラン	-
メインタイマ		メインラン	-
PLLタイマ		PLLラン	-
サブタイマ		サブラン	-
ストップ		CR ラン , メインラン PLL ラン, サブラン (SC1S[1:0] ビットに最後に書き込まれた値によって異なる)	-

各動作モードでの動作状態

各動作モードでの動作状態を Table 9-5 に示します。

Table 9-5. 各動作モードでの動作状態

動作モード	CR 発振器と CR クロックタイマ	メイン発振器とメインクロックタイマ	PLL クロック	サブ発振器とサブクロックタイマ	バスクロック CLKB 用のクロックソース	周辺クロック用のクロックソース	
						CLKP1	CLKP2
CRラン	アクティブ	MCE ビットと SC2M ビットに従う	PCE, MCE, および SC2M ビットに従う	SCE ビットと SC2M ビットに従う	CRクロック	CRクロック	SC2Mビットに従う
CRスリープ					停止		
CRタイマ						停止	
メインラン	RCE ビットと SC2M ビットに従う	アクティブ	PCE ビットと SC2Mビットに従う	SCE ビットと SC2M ビットに従う	メインクロック	メインクロック	SC2Mビットに従う
メインスリープ					停止		
メンタイマ						停止	
PLLラン	RCE ビットと SC2M ビットに従う	アクティブ	アクティブ	SCE ビットと SC2M ビットに従う	PLLクロック	PLLクロック	SC2Mビットに従う
PLLスリープ					停止		
PLLタイマ						停止	
サブラン	RCE ビットと SC2M ビットに従う	MCE ビットと SC2M ビットに従う	PCE, MCE, および SC2M ビットに従う	アクティブ	サブクロック	サブクロック	SC2Mビットに従う
サブスリープ					停止		
サブタイマ						停止	
ストップ	停止						
外部リセットがアクティブになる	CR 発振器が許可され, CR クロックタイマが停止する	メイン発振器が許可され, メインクロックタイマがアクティブになる	停止	サブ発振器が許可され, サブクロックタイマが停止する	停止		

9.6 スタンバイモードに関する使用上の注意事項

スタンバイモードを使用する際には以下の点に注意してください。

- スタンバイモードへの切換えと割込み
- 割込みによるスタンバイモードの解除
- 割込みによるストップモードの解除
- ストップモード解除後の発振安定待ち時間
- クロックモードの切換え
- スタンバイモードへの切換えと割込み

CPU に対して保留中の割込み要求がある場合 , SMCR レジスタの SMS ビットへの書込みは効果がありません。つまり , 割込みサービスが終了した後でも目的のモード移行が行われません。割込み要求が既に解除されており , かつ保留中の要求がほかに存在しない場合は , 目的のモード移行を行うことができます。この動作は , PS レジスタの設定によって CPU での割込み要求の受付けが可能になるかどうかに関係しません。

- 割込みによるスタンバイモードの解除

スリープ , タイマ , ストップのいずれかのモードにある間に NMI 割込みまたは割込みレベル (IL) が 7 より強い割込み要求が周辺機能から発生した場合は , CPU が割込みを受け付けるかどうかに関係なくスタンバイモードが解除されます。

割込みによってスタンバイモードが解除された後は , 通常の処理が行われます。

割込みが受け付けられた場合 , まず CPU はスタンバイモードに入る前の状態でパイプラインに残っている命令を実行します。この後 , 通常の割込み処理を続けます。

割込みが受け付けられなかった場合 , CPU はパイプラインにある命令とその後に続く命令の実行を続けます。

スリープモードまたはタイマモード中に電圧レギュレータを手動で低電力モードに設定した場合 (SMCR:LMPSS ビットで設定), ウェイクアップは標準で 40µs 遅延します。

- **割込みによるストップモードの解除**

ストップモードは , 外部割込みまたは NMI で解除することができます。外部割込み入力要因として , H レベル信号 , L レベル信号 , 立上りエッジ , または立下りエッジを選択できます。

- ストップモード解除後の発振安定待ち時間

ストップモードで発振器は停止するため , 発振安定待ち時間が必要となります。CKSR レジスタの SC1S[1:0] と SC2S[1:0] ビットの設定によって , 割込みによるストップモードからのウェイクアップ後に使用されるクロックが決まります。RCE, MCE, PCE, SCE, SC1S[1:0]およびSC2S[1:0]ビットに応じて発振器と PLL クロックが許可され , クロック安定化選択レジスタ (CKSSR) の設定に従って安定時間が適用されます。CPU は , SC1S[1:0] ビットで選択されたシステムクロック 1 が安定した後に動作を開始します。システムクロック 2 を使用しているモジュールは , システムクロック 2 が安定した後に動作を開始します。SC1S[1:0] と SC2S[1:0] で選択されたクロックは , クロック許可ビット (RCE, MCE, PCE および SCE) の設定に関係なく常に許可されます。

＜注意事項＞

ストップモードが解除されると , CPU は , SC1S[1:0] ビットで選択されたシステムクロック 1 が安定した後に動作を開始します。別のクロックをシステムクロック 2 として選択できますが , 安定時間が長くなります。CPU は , 周辺クロック 2 によって同期がとられている周辺リソースへのアクセスを試行した場合 , CLKP2 が安定し , アクセス可能になるまでホールド状態に入ります。

ストップモードが解除されてから , 新たに選択されたクロックの準備が完了する (SC2S[1:0] ビットで選択された

クロックのクロック準備完了モニタビットが "1" に設定される) まで , ストップモード移行前に使用されていたクロックが SC2M ビットに示されます。この理由として , このレジスタはシステムクロック 2 だけが更新されるからです。

＜注意事項＞

SC1S[1:0] で選択されたクロックが安定するまでプログラムの実行が延期されるため , SC1S[1:0] / SC2S[1:0] を存在しないクロック (例えば , デバイスにサブ発振器が備わっていない場合やサブ発振器端子に水晶 / 振動子が接続されていない場合に「サブクロックモード」) に設定してはなりません。ストップモードに切り換える場合は , 常に CR クロックを両方のシステムクロックとして選択することを推奨します。

ストップモードがリセットで解除されると, クロックモードは必ず CR クロックに設定され, CR 発振器が安定した後にリセットシーケンスが実行されます。レジスタと機能はすべて初期状態にリセットされます。

■ クロックモードの切換え

SC1S[1:0] / SC2S[1:0] ビットへの書込みによってクロックモードを切り換えた場合は , クロックモードの切換えが終了するまで, スリープモードとタイマモードに移行しないでください。クロックモニタレジスタ (CKMR) の SC1M/SC2M ビットを参照して , クロックモードの切換えが終了しているかどうかを確認します。

ただし , ストップモードへの移行は , クロックモードの切換えが終了するまで待機せずに実行できます。

SC1S[1:0] / SC2S[1:0] ビットの設定を変更するのは , SMS ビットが "00" (スタンバイモード移行要求が保留中でない) に設定されている場合に限ります。

9.7 電圧レギュレータの動作

F2MC-16FX MCU は , 外部電源からコアロジック電源を生成するオンチップ電圧レギュレータを装備しています。このレギュレータには , 出力電圧をプログラムできるいくつかの動作モードがあります。この機能を利用することで , 特にスタンバイモードでの電力消費を最適化できます。

レギュレータのいくつかの動作モード, 動作モードの変更方法、および出力電圧の設定方法について説明しています。

高電力モード

高電力モードは , 電圧レギュレータの初期設定での動作モードです。このモードでは , レギュレータは MCU のどのモードでも最大消費電流を提供できます。このモードの出力電圧は個別にプログラミングできます。初期設定での出力電圧は 1.8V (製造テクノロジに合わせて調整), レギュレータ自体の消費電流は数百マイクロアンペアです。

CR 振動子またはメイン振動子の動作中は , 電圧レギュレータはこのモードで動作します (SMCR:LMPS または SMCR:LPMSS ビットが設定されていない限り)。

低電力モード

低電力モードは , 電圧レギュレータのもう 1 つの動作モードです。低電力モードでは , レギュレータは 100 µA の最大過渡電流を供給できます。レギュレータ自体の消費電流は 10 µA まで低減されます。フラッシュ書込み / 消去はこのモードで許可されていません。

低電力モードには 2 種類あります。各モードの出力電圧は個別にプログラミングできます。どちらのモードも初期設定での出力電圧は 1.8V です。

- 低電力モード A

CR 振動子とメイン発振器の両方が停止し , サブ振動子のみが動作している場合 (CR およびメイン振動子停止状態のサブ RUN/ サブスリープ / サブタイマモード) にのみ電圧レギュレータはこのモードで動作します。

RC 振動子またはメイン振動子が許可されている場合でも , このモードは SMCR:LPMS または SMCR:LPMSS ビットの設定により選択できます。

- 低電力モード B

すべての振動子が停止している場合 (ストップモード) にのみ電圧レギュレータはこのモードで動作します。

このモードは SMCR:LPMBSS ビットの設定により選択できます。

＜注意事項＞

CPU 動作モードがストップモード時以外は電圧レギュレータを低電力モード B に切換えることは禁止です。すなわち SMCR:LPMBSS ビットは "0" (初期値) のままで使用してください。

9.7.1 電圧レギュレータの動作モードの変更

電圧レギュレータの動作モードは, 指定された CPU の動作モードに合わせて, ハードウェアによって自動制御されます。しかし一部のモードや一定条件のもとでは, 電流をさらに節約するために, レギュレータを高電力モードから低電力モードに手動で変更することができます。

9.7.1.1 概要

MCU コアの消費電流は , MCU の動作モード (ラン , スリープ , タイマまたはストップの各モード), 起動中の発振器と PLL, および選択された周波数 (CLKS1, CLKS2, CLKB, CLKP1, CLKP2) に左右されます。

電圧レギュレータの自動制御ロジックは , メインあるいは CR 発振器が起動するたびに , レギュレータを高電力モードに設定します。この高電力モードでは , レギュレータはコアによって消費されるあらゆる電流を駆動できます。これにより , 選択された CPU の動作モードや , CR またはメインクロックの周波数から独立した安全な MCU 動作が保証されます。

しかし一定のアプリケーションでは, 専用の低電力の CPU 動作モードにおいて , 非常に少量の MCU 消費電流を必要とします。これは , メインまたは CR 発振器がアクティブである場合でも , 電圧レギュレータを低電力モードに手動で設定することで実現できます。これにより , MCU の消費電力は数百 µA 低下します。

この機能の使用は , ダイナミックな消費電流が少ない CPU の動作モードだけで許可されています。電圧レギュレータの自動制御では , 発振器の安定期間中にレギュレータの安定時間が発生します。したがって , 電圧レギュレータの安定がそれ以上遅延することはありません。

9.7.1.2 ユーザプログラムによる電圧レギュレータの低電力モードへの設定

消費電力を削減する必要のある CPU 動作モードによっては , SMCR:LPMS ビットと SMCR:LPMSS ビットを制御することで , 電圧レギュレータを低電力モードに設定できます。

- SMCR:LPMS ビットによる低電力モードへの切換え

SMCR:LPMS ビットを "1" に設定すると , 電圧レギュレータが低電力モードに即座に切り換わります。これは PLL が無効状態で , メインクロックの最大周波数が 4MHz である CR ランモード (100kHz) だけに許可されています。MCU の動作モード (ランモードまたはスタンバイモード) には関係なく , このビットがクリアされるまでは , レギュレータは低電力モードのままです。

SC1S/SC2S ビットをセットして , SC1M/SC2M ビットを読み出すことでその遷移を確認して , MCU を CR ランモードに設定し , CLKS2 を CR またはサブクロックに設定する必要があります。そうすることで , SMCR:LPMS ビットを "1" に設定することができ , 電圧レギュレータは低電力モードに切り換わります。

CPU の動作モードを消費電量のより高い設定に変更する場合は , 事前にレギュレータを高電力モードに戻しておく必要があります。これは以下の順序で行われなければなりません。

- SMCR:LPMS ビットをクリアする。
- 電圧レギュレータの安定のために 40ms の待ち時間をとる。
- 新しい動作モードに切り換える (SC1S/SC2S ビットの変更, PLL の許可など)。

- SMCR:LPMSS ビットによる低電力モードへの切換え

スリープモードやタイマモードで電圧レギュレータを低電力モードに切り換えるのに, このビットを使用できます。

SMCR:LPMSS ビットを "1" に設定すると , 次回のスタンバイモードへの遷移時に , 電圧レギュレータは低電力モードに切り換わります。MCU がランモードである限り , このビットは動作に影響を与えません。SMCR:SMS

スタンバイモード要求ビットの設定前あるいは設定と同時に , SMCR:LPMSS ビットを設定できます。

ウェイクアップ割込みにより , 電圧レギュレータは再び高電力モードに自動的に切り換わります (メインまたは CR 発振器が許可されている場合)。ランモードへの遷移は , レギュレータの安定時間の分 (~40µs) だけ自動的に遅延します。

SMCR:LPMS を "1" に設定すると , SMCR:LPMSS ビットの設定が解除されます。

- SMCR:LPMBSS ビットによる低電力モード B への切換え

＜注意事項＞

CPU 動作モードがストップモード時以外は電圧レギュレータを低電力モード B に切換えることは禁止です。すなわち SMCR:LPMBSS ビットは "0" (初期値) のままで使用してください。

SMCR:LPMBSS ビットを "1" に設定すると , 次回のスタンバイモードへの遷移後に , 電圧レギュレータは低電力モード A ではなく , 低電力モード B に切り換わります。これは , VRCR:LPMA[2:0] ビットではなく VRCR:LPMB[2:0] ビットによって出力電圧が制御されることを意味します。SMCR:SMS スタンバイモード要求ビットの設定前あるいは設定と同時に , SMCR:LPMBSS ビットを設定できます。

デバイスがスタンバイモード状態であり , ハードウェアまたは SMCR:LPMS, SMCR:LPMSS ビットによってレギュレータが低電力モードに設定される場合にのみ , このビットは動作に影響を与えます。

ウェイクアップ割込みにより , 電圧レギュレータは高電力モードまたは低電力モード A に再び切り換わります。ランモードへの遷移は , レギュレータの安定時間の分 (~40µs) だけ自動的に遅延します。

9.7.1.3 電圧レギュレータの低電力モードを使用許可された構成

電圧レギュレータは以下の構成で低電力モードに切り換えることができます(SMCR:LPMSまたはLMPSSを"1"に設定)。

Table 9-6. 低電力モードAを使用許可された構成

動作モード	CR発振器	メイン発振器	PLL	サブ発振器
CRラン (*1)	アクティブ (100kHz に設定)	fosc ≦ 4MHz でアクティブ, あるいは禁止	禁止	アクティブ, あるいは禁止
CRスリープ (*1)				
CRタイマ	アクティブ (100kHz または 2MHz に設定)			
メインタイマ	アクティブ (100kHz または 2MHz に設定), あるいは禁止	fosc ≦ 4MHz でアクティブ		
サブラン (*1)		fosc ≦ 4MHz でアクティブ, あるいは禁止		アクティブ
サブスリープ (*1)				
サブタイマ				

*1: システムクロック 2 は , CR クロックまたはサブクロックへの設定が必要です。CR クロックの 2MHz 設定は , ランモードまたはスリープモードのシステムクロック 2 では禁止されています。

9.7.2 出力電圧の電圧レギュレータの設定

レギュレータの出力電圧をプログラムして, 必要な性能または消費電流に応じてコア電圧を調節できます。

9.7.2.1 高電力モードの電圧設定

レギュレータの初期設定での出力電圧は 1.8 V です。コア電圧に応じた内部クロック周波数 (CLKS1, CLKS2, CLKB, CLKP1 および CLKP2) については , データシートを参照してください。

- 1.9 V へのコア電圧の変更

最大の性能を必要とする用途では, 高速クロックに変更する前に VRCR:HPM ビットを "11" に設定してコア電圧を 1.9 V に設定する必要があります。

高周波数の PLL を許可する前にレギュレータを 1.9V に設定してください。

フラッシュデバイスでは , 高速クロックへの変更前に FMCS:RD19V ビットを "1" に設定することも必要です。VRCR および FMCS レジスタへの書込みアクセスの順序は問われません。

- 1.6 V と 1.7 V の設定

これらの設定は , 使用しないでください。

9.7.2.2 低電力モードの電圧設定

低電力モード A と B におけるレギュレータの初期設定での出力電圧は 1.8V です。

- 低電力モード A におけるコア電圧の変更

弊社では , 低電力モード A におけるレギュレータの出力電圧の変更を承認していません。

- 低電力モード B におけるコア電圧の変更

ストップモードで少量の消費電流を必要とするアプリケーションの場合 , ストップモードに移行する前に VRCR:LPMB[2:0] ビットを "000" に設定することで , コア電圧を 1.2V まで低下できます。これにより , MCU のリーク電流が削減されます。

コア電圧が 1.2V と 1.8V の場合のストップモード電流の絶対値については , データシートを参照してください。

10. ソースクロックタイマ

3 つのソースクロックタイマ(CR クロックタイマ, メインクロックタイマ, サブクロックタイマ)の機能と動作について説明します。

10.1 概要

MB96300スーパー シリーズでは, 3 つの独立したソースクロックタイマ(CR クロックタイマ, メインクロックタイマ, サブクロックタイマ)を提供しています。これらのタイマは指定したインターバルで割込みを発行でき, 発振安定時間を測定するために使用されます。

CRクロックタイマ

CR クロックタイマは, 内部 CR 発振器(CLKRC) の出力信号をクロックとして使用し, CR 発振器が使用可能のとき常に実行しています。このタイマはまた , CR クロックの安定待ち時間を測定するためにも使用します。

メインクロックタイマ

メインクロックタイマは , メイン発振回路 (CLKMC) の出力信号をクロックとして使用し , メイン発振器が使用可能のとき常に実行しています。このタイマはまた , メインクロックの安定待ち時間を測定するためにも使用します。

サブクロックタイマ

サブクロックタイマは, サブ発振回路 (CLKSC) の出力信号をクロックとして使用し, サブ発振器が使用可能のとき常に実行しています。このタイマはまた , サブクロックの安定待ち時間を測定するためにも使用します。

10.2 CRクロックタイマ

CR クロックタイマは23 ビットカウンタと制御レジスタによって構成されます。23 ビットカウンタはCR クロック CLKRC を分周します。CR クロックタイマは,CR クロックカウンタのキャリ信号により,指定したインターバルで割込みを発行します。

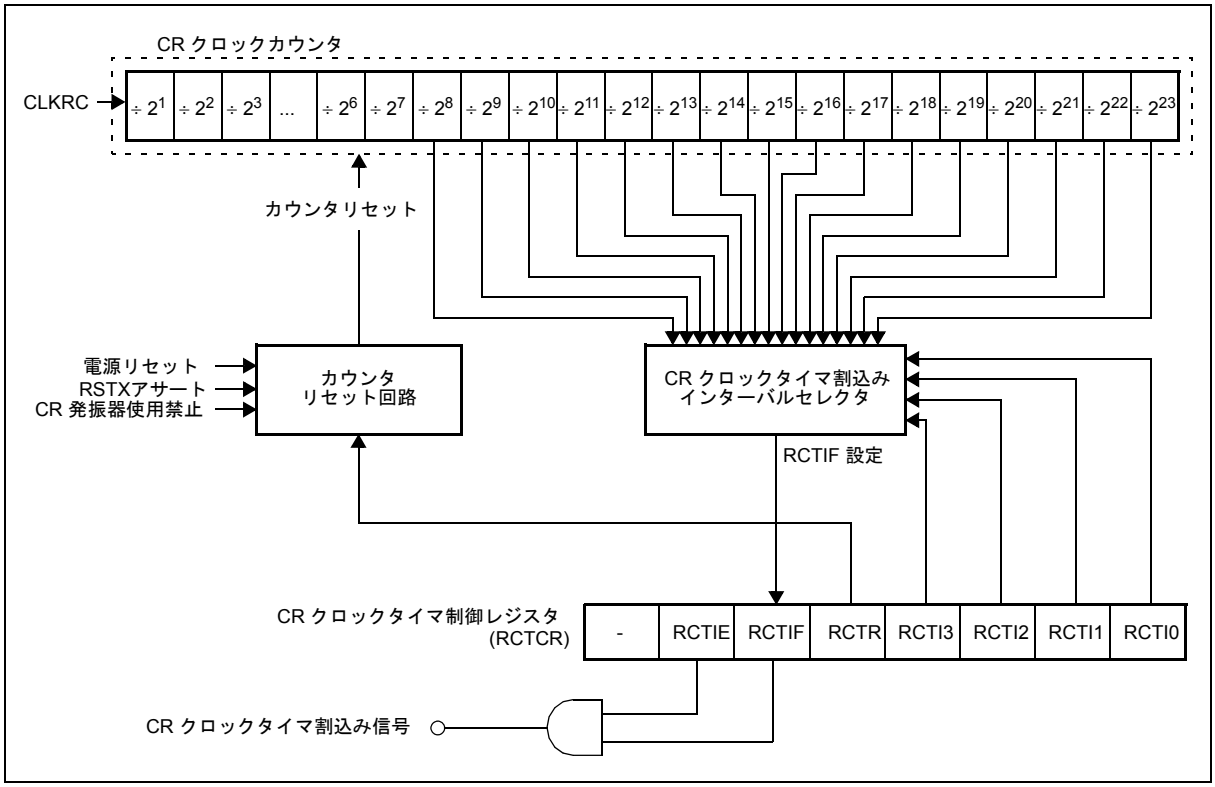
CRクロックタイマの概要

電源または外部リセットによって CR クロックタイマは 0 に初期化されます。CR クロックタイマは , CR クロックが禁止されたとき(ストップモードへの遷移時, または RCE を"0"にセット), または RCTCR レジスタの RCTR ビットに "0" を書き込むことによって初期化されます。CR クロックタイマは CR クロックが供給されている間はカウントを継続します。CR クロックタイマは CR クロックの安定待ち時間を測定するために使用し, インターバル割込みを生成するためにも使用できます。

CRクロックタイマのブロックダイヤグラム

Figure 10-1 は , CR クロックタイマのブロックダイヤグラムを示しています。

Figure 10-1. CRクロックタイマのブロックダイヤグラム



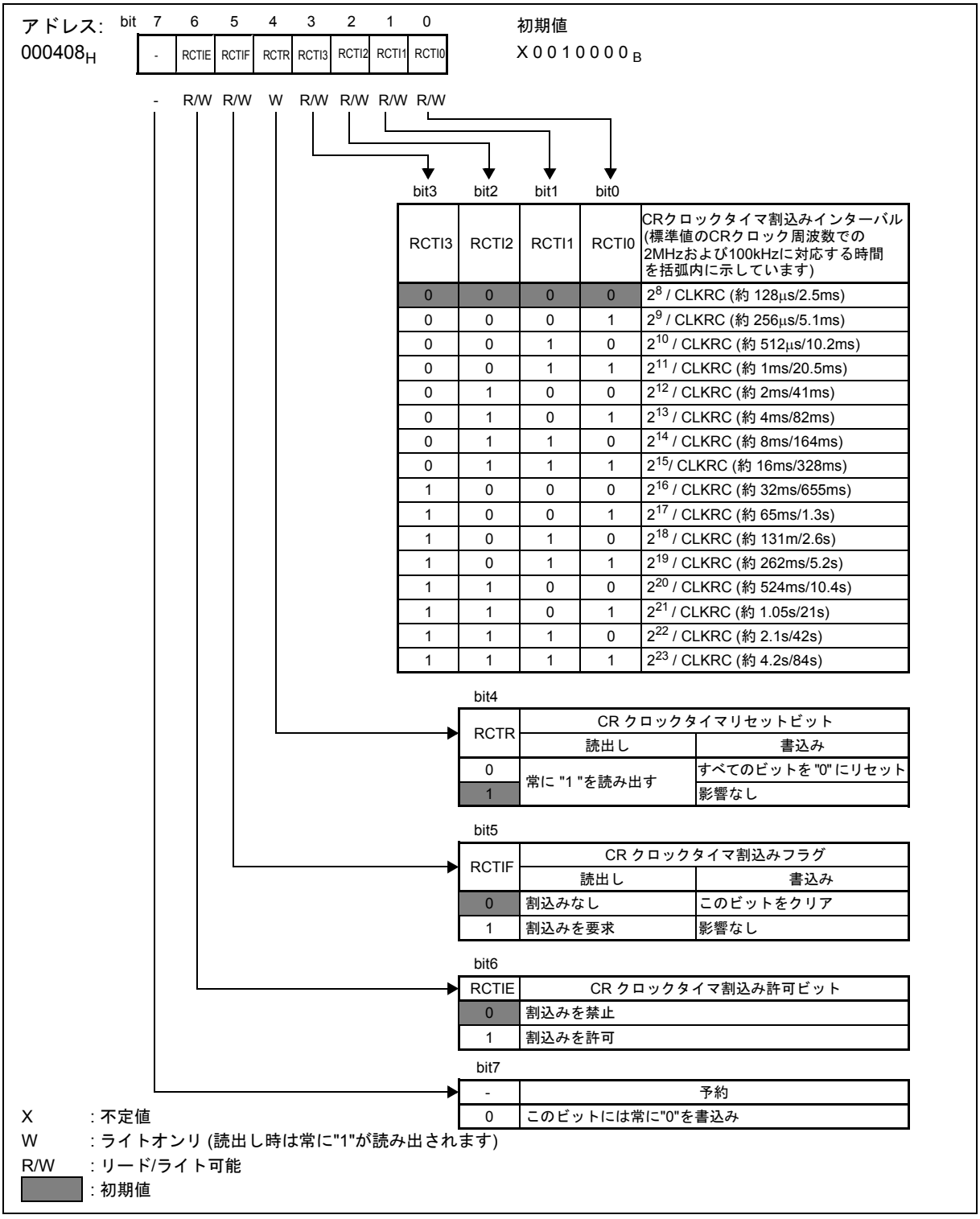
10.2.1 CRクロックタイマ制御レジスタ(RCTCR)

CR クロックタイマ制御レジスタ (RCTCR) は, CR クロックタイマインターバル割込み機能を制御し, CR クロックタイマをリセットするために使用します。

CRクロックタイマ制御レジスタ(RCTCR)の構成

Figure 10-2 は, CR クロックタイマ制御レジスタ (RCTCR) の構成を示し, Table 10-1 は各ビットの機能を説明しています。

Figure 10-2. CRクロックタイマ制御レジスタ(RCTCR)の構成



RCTI3	RCTI2	RCTI1	RCTI0	CRクロックタイマ割込みインターバル (標準値のCRクロック周波数での 2MHzおよび100kHzに対応する時間 を括弧内に示しています)
0	0	0	0	2^8 / CLKRC (約 128µs/2.5ms)
0	0	0	1	2^9 / CLKRC (約 256µs/5.1ms)
0	0	1	0	2^{10} / CLKRC (約 512µs/10.2ms)
0	0	1	1	2^{11} / CLKRC (約 1ms/20.5ms)
0	1	0	0	2^{12} / CLKRC (約 2ms/41ms)
0	1	0	1	2^{13} / CLKRC (約 4ms/82ms)
0	1	1	0	2^{14} / CLKRC (約 8ms/164ms)
0	1	1	1	2^{15}/ CLKRC (約 16ms/328ms)
1	0	0	0	2^{16} / CLKRC (約 32ms/655ms)
1	0	0	1	2^{17} / CLKRC (約 65ms/1.3s)
1	0	1	0	2^{18} / CLKRC (約 131m/2.6s)
1	0	1	1	2^{19} / CLKRC (約 262ms/5.2s)
1	1	0	0	2^{20} / CLKRC (約 524ms/10.4s)
1	1	0	1	2^{21} / CLKRC (約 1.05s/21s)
1	1	1	0	2^{22} / CLKRC (約 2.1s/42s)
1	1	1	1	2^{23} / CLKRC (約 4.2s/84s)

bit4

RCTR	CR クロックタイマリセットビット	
	読出し	書込み
0	常に "1 "を読み出す	すべてのビットを "0" にリセット
1		影響なし

bit5

RCTIF	CR クロックタイマ割込みフラグ	
	読出し	書込み
0	割込みなし	このビットをクリア
1	割込みを要求	影響なし

bit6

RCTIE	CR クロックタイマ割込み許可ビット
0	割込みを禁止
1	割込みを許可

bit7

-	予約
0	このビットには常に"0"を書込み

Table 10-1. CRクロックタイマ制御レジスタ(RCTCR)の各ビットの機能の説明 (Sheet 1 of 2)

<table>
<tr><th colspan="2">ビット名</th><th>機能</th></tr>
<tr><td>bit 0 ～
bit 3</td><td>RCTI3 ～ RCTI0:
CR クロック
タイマ割込み
インターバル
選択ビット</td><td>
• これらのビットは , 次の表に従って CR クロックタイマ割込みインターバルを制御します。
<table>
<tr><th colspan="4">bit3 bit2 bit1 bit0</th><th>CR クロックタイマ割込みインターバル
(標準値の CR クロック周波数での 2MHz および 100kHz
に対応する時間を括弧内に示しています)</th></tr>
<tr><td>RCTI3</td><td>RCTI2</td><td>RCTI1</td><td>RCTI0</td><td></td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>2^8 / CLKRC (約 128µs/2.5ms)</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>2^9 / CLKRC (約 256µs/5.1ms)</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>2^{10} / CLKRC (約 512µs/10.2ms)</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>2^{11}/ CLKRC (約 1ms/20.5ms)</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>2^{12}/ CLKRC (約 2ms/41ms)</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>2^{13}/ CLKRC (約 4ms/82ms)</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>2^{14}/ CLKRC (約 8ms/164ms)</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>2^{15}/ CLKRC (約 16ms/328ms)</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>2^{16}/ CLKRC (約 32ms/655ms)</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>2^{17}/ CLKRC (約 65ms/1.3s)</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>2^{18}/ CLKRC (約 131m/2.6s)</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>2^{19}/ CLKRC (約 262ms/5.2s)</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>2^{20}/ CLKRC (約 524ms/10.4s)</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>2^{21}/ CLKRC (約 1.05s/21s)</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>2^{22}/ CLKRC (約 2.1s/42s)</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>2^{23}/ CLKRC (約 4.2s/84s)</td></tr>
</table>
• これらのビットは , どのリセットでも "0000" に初期化されます。
• これらのビットにデータが書き込まれるとき , 同時にbit5 (RCTIF) をクリアしなければなりません。
• 表の中の時間の計算値は, 標準値の CR クロック周波数で算出しています。しかし, 実際の周波数およびインターバル時間は異なります。CR クロック周波数の精度の詳細についてはデータシートを参照してください。
</td></tr>
</table>

Table 10-1. CRクロックタイマ制御レジスタ(RCTCR)の各ビットの機能の説明 (Sheet 2 of 2)

ビット名		機能
bit 4	RCTR: CR クロック タイマリセット ビット	• このビットは , CR クロックカウンタのすべてのビットをクリアします。 • "0" を書き込むと CR クロックカウンタがクリアされます。 • "1" を書き込んでも影響はありません。 • このビットからは常に "1" が読み出されます。 (注意事項) このビットに "0" を書き込んで CRクロックタイマをクリアする場合, RCTIF ビットにも "0" を書き込んでCRクロックタイマの割込みフラグをクリアしてください。
bit 5	RCTIF: CR クロック タイマ割込み フラグ	• これは CR クロックタイマの割込み要求フラグです。 • このビットは RCTI3 ～ RCTI0 ビットによって指定された各インターバルで "1" にセットされます。 • 割込み許可ビット RCTIE が "1" にセットされている場合は , このビットが "1" にセットされたときに割込み要求が発行されます。 • このビットは "0" の書込み , およびいずれかのリセットによってクリアされます。 • CR クロックタイマをリセットする場合 , このビットも必ずクリアしてください。 • CR クロックタイマ割込みインターバルビット RCTI[3:0] はいずれのリセットでも "0000" (2^8/CLKRC サイクル) にリセットされます。したがって , いずれのリセット後でも 2^8/CLKRC サイクルで , RCTIF ビットは "1" に設定されます。 • "1" を書き込んでも影響はありません。 • リードモディファイライト系命令では常に"1"が読み出されます。
bit 6	RCTIE: CR クロック タイマ割込み 許可ビット	• このビットは , CR クロックタイマによるインターバル割込みを許可するために使用します。 • このビットに "1" を書き込むと割込みが許可され , "0" を書き込むと割込みが禁止されます。 • どのリセットでも"0"に初期化されます。
bit 7	予約	• このビットには常に "0" を書き込んでください。 • このビットの読出し値は未定義です。 • このレジスタへのリードモディファイライト系命令は無効です。

10.2.2 CRクロックタイマの動作

CR クロックタイマは指定されたインターバルで割込みを生成するためのインターバルタイマとして機能します。

CRクロックカウンタ

CR クロックカウンタは , CR クロック CLKRC をクロックとして使用する 23 ビットカウンタによって構成されます。CR クロックがアクティブの時 , CR クロックカウンタは常にカウントを継続します。CR クロック周波数は CKFCR: RCFS ビット によって設定されます。

CR クロックカウンタは , CR クロックが停止されたとき (ストップモードへの遷移 , または CKSR: RCE ビットによる CR クロックの禁止と CKMR: RCM ビットによる確認), RCTCR レジスタの RCTR ビットへの "0" の書込みによって , または電源リセットおよび外部リセットによってクリアされます。

＜注意事項＞

外部リセットがアサートされている間は CR クロックカウンタはクリアされ , 停止されます。このとき , CR クロック安定時間カウンタも停止されます。そのため CR クロック安定時間は RSTX によって引き延ばされ , 外部リセットがアサート解除された後でカウントが開始されます。この動作はメインクロックカウンタと異なります。

インターバル割込み機能

CR クロックカウンタのキャリ信号に従って , 指定したインターバルで割込みが生成されます。RCTCR レジスタの RCTI3 ～ RCTI0 ビットで指定されたインターバルで RCTIF フラグがセットされます。CR クロックタイマをクリアし , カウントが開始されたときに , インターバル時間は開始します。

CR クロック安定時間カウンタは CR クロックカウンタに接続されます。したがって , CR クロックが停止されたとき (ストップモードへの遷移 , または CKSR: RCE ビットによる CR クロックの禁止と CKMR: RCM ビットによる確認), CR クロックカウンタが直ちにクリアされます。

10.3 メインクロックタイマ

メインクロックタイマは23ビットカウンタと制御レジスタによって構成されます。23ビットカウンタはメインクロックCLKMCを分周します。メインクロックタイマは,メインクロックカウンタのキャリ信号により,指定したインターバルで割込みを発行します。

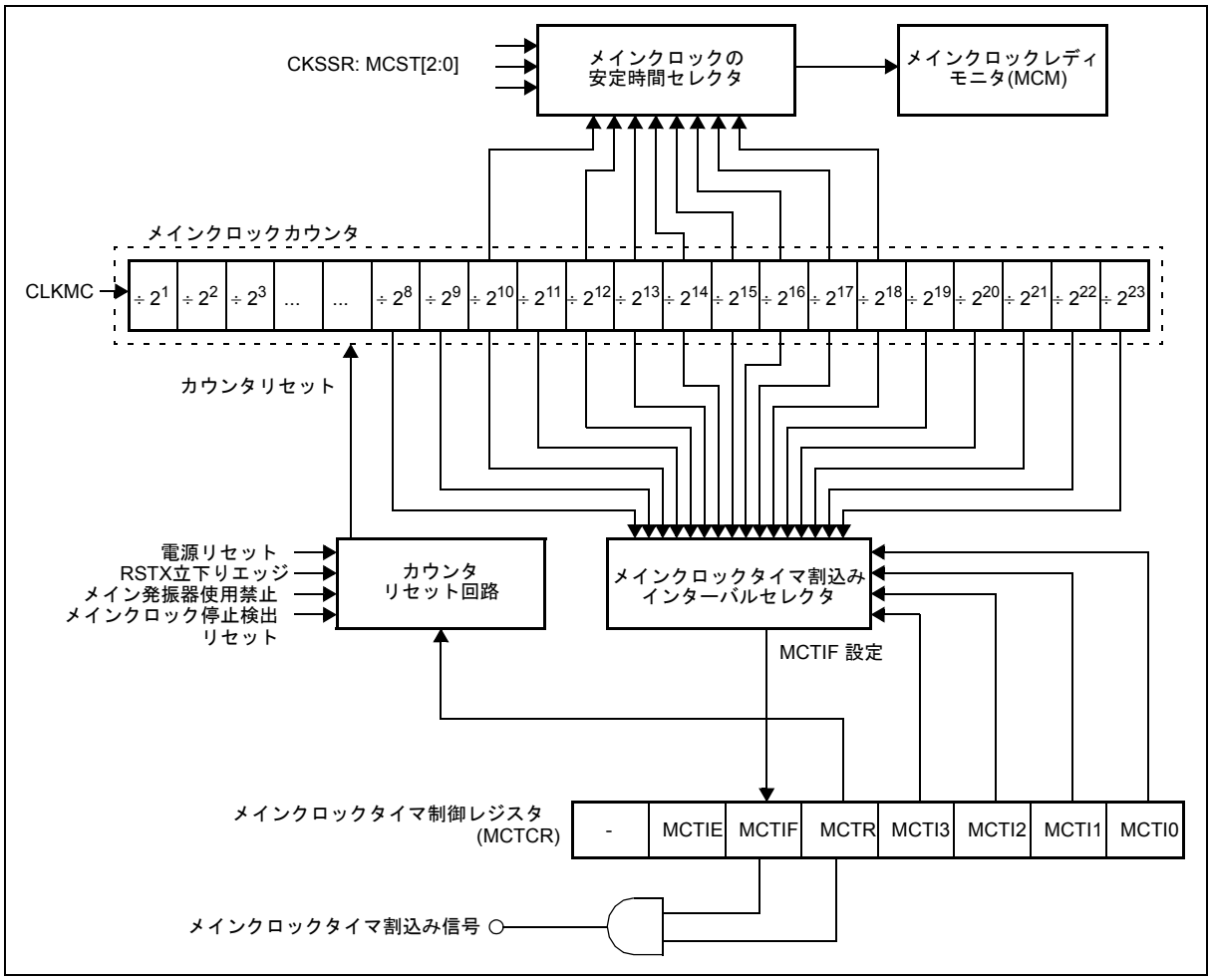
メインクロックタイマの概要

電源のリセットまたはRSTX入力端子の立下りエッジでメインクロックタイマが0に初期化されます。メインクロックタイマは, メインクロックが禁止されたとき (ストップモードへの遷移時 , または MCE を "0" にセット), または MCTCR レジスタの MCTR ビットに "0" を書き込むことによって初期化されます。メインクロックタイマはメインクロックが供給されている間はカウントを継続します。メインクロックタイマはメインクロックの安定待ち時間を測定するために使用し , インターバル割込みを生成するためにも使用できます。

メインクロックタイマのブロックダイヤグラム

Figure 10-3 は , メインクロックタイマのブロックダイヤグラムを示しています。

Figure 10-3. メインクロックタイマのブロックダイヤグラム



10.3.1 メインクロックタイマ制御レジスタ(MCTCR)

メインクロックタイマ制御レジスタ (MCTCR) は, メインクロックタイマインターバル割込み機能を制御し, メインクロックタイマをリセットするために使用します。

メインクロックタイマ制御レジスタ(MCTCR)の構成

Figure 10-4 は , メインクロックタイマ制御レジスタ (MCTCR) の構成を示し , Table 10-2 は各ビットの機能を説明しています。

Figure 10-4. メインクロックタイマ制御レジスタ(MCTCR)の構成

アドレス: 000409$_H$

bit	15	14	13	12	11	10	9	8
	-	MCTIE	MCTIF	MCTR	MCTI3	MCTI2	MCTI1	MCTI0
	-	R/W	R/W	W	R/W	R/W	R/W	R/W

初期値
X 0 0 1 0 0 0 0 $_B$

bit11 bit10 bit9 bit8

MCTI3	MCTI2	MCTI1	MCTI0	メインクロックタイマ 割込みインターバル (4MHz のメインクロックの対応する時間を括弧内に示しています)
0	0	0	0	2^8 / CLKMC (約 64μs)
0	0	0	1	2^9 / CLKMC (約 128μs)
0	0	1	0	2^{10} / CLKMC (約 256μs)
0	0	1	1	2^{11} / CLKMC (約 512μs)
0	1	0	0	2^{12} / CLKMC (約 1ms)
0	1	0	1	2^{13} / CLKMC (約 2ms)
0	1	1	0	2^{14} / CLKMC (約 4ms)
0	1	1	1	2^{15}/ CLKMC (約 8ms)
1	0	0	0	2^{16} / CLKMC (約 16ms)
1	0	0	1	2^{17} / CLKMC (約 32ms)
1	0	1	0	2^{18} / CLKMC (約 65ms)
1	0	1	1	2^{19} / CLKMC (約 131ms)
1	1	0	0	2^{20} / CLKMC (約 262ms)
1	1	0	1	2^{21} / CLKMC (約 524ms)
1	1	1	0	2^{22} / CLKMC (約 1.049s)
1	1	1	1	2^{23} / CLKMC (約 2.097s)

bit12

MCTR	メインクロックタイマリセットビット	
	読出し	書込み
0	常に"1" を読み出す	すべてのビットを "0" にリセット
1		影響なし

bit13

MCTIF	メインクロックタイマ割込みフラグ	
	読出し	書込み
0	割込みなし	このビットをクリア
1	割込みを要求	影響なし

bit14

MCTIE	メインクロックタイマ割込み許可ビット
0	割込みを禁止
1	割込みを許可

bit15

-	予約
0	このビットに常に"0"を書込み

X : 不定値
W : ライトオンリ (読出し時は常に"1"が読み出されます)
R/W : リード/ライト可能
: 初期値

Table 10-2. メインクロックタイマ制御レジスタ(MCTCR)の各ビットの機能の説明 (Sheet 1 of 2)

<table>
<tr><th colspan="2">ビット名</th><th>機能</th></tr>
<tr><td>bit 8 ～ bit 11</td><td>MCTI3 ～ MCTI0:
メインクロック
タイマ割込み
インターバル
選択ビット</td><td>
• これらのビットは , 次の表に従ってメインクロックタイマ割込みインターバルを制御します。
<table>
<tr><th>bit11</th><th>bit10</th><th>bit9</th><th>bit8</th><th>メインクロックタイマ割込みインターバル
(4MHz のメインクロックの対応する時間を
括弧内に示しています)</th></tr>
<tr><td>MCTI3</td><td>MCTI2</td><td>MCTI1</td><td>MCTI0</td><td></td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>2^8 / CLKMC (約 64µs)</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>2^9 / CLKMC (約 128µs)</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>2^{10} / CLKMC (約 256µs)</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>2^{11} / CLKMC (約 512µs)</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>2^{12}/ CLKMC (約 1ms)</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>2^{13}/ CLKMC (約 2ms)</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>2^{14}/ CLKMC (約 4ms)</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>2^{15}/ CLKMC (約 8ms)</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>2^{16}/ CLKMC (約 16ms)</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>2^{17}/ CLKMC (約 32ms)</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>2^{18}/ CLKMC (約 65ms)</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>2^{19}/ CLKMC (約 131ms)</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>2^{20}/ CLKMC (約 262ms)</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>2^{21} / CLKMC (約 524ms)</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>2^{22}/ CLKMC (約 1.049s)</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>2^{23}/ CLKMC (約 2.097s)</td></tr>
</table>
• これらのビットは , どのリセットでも "0000" に初期化されます。
• これらのビットにデータが書き込まれるとき, 同時にbit13 (MCTIF) をクリアしなければなりません。
</td></tr>
</table>

Table 10-2. メインクロックタイマ制御レジスタ(MCTCR)の各ビットの機能の説明 (Sheet 2 of 2)

ビット名		機能
bit 12	MCTR: メインクロック タイマリセット ビット	• このビットは , メインクロックカウンタのすべてのビットをクリアします。 • "0" を書き込むとメインクロックカウンタがクリアされます。 • "1" を書き込んでも影響はありません。 • このビットからは常に "1" が読み出されます。 (注意事項) このビットに"0"を書き込んでメインクロックタイマをクリアする場合は, MCTIF ビットにも "0" を書き込んでメインクロックタイマの割込みフラグをクリアしてください。
bit 13	MCTIF: メインクロック タイマ割込み フラグ	• これはメインクロックタイマの割込み要求フラグです。 • このビットは MCTI3～MCTI0 ビットによって指定された各インターバルで"1"にセットされます。 • 割込み許可ビット MCTIE が "1" にセットされている場合は , このビットが "1" にセットされたときに割込み要求が発行されます。 • このビットは "0" の書込み , およびいずれかのリセットによってクリアされます。 • メインクロックタイマをリセットする場合 , このビットも必ずクリアしてください。 • メインクロックタイマ割込みインターバルビット MCTI[3:0] はいずれのリセットでも "0000" (2^8/CLKMC サイクル) にリセットされます。したがって , いずれのリセット後でも 2^8/CLKMC サイクルで , MCTIF ビットは "1" に設定されます。 • "1" を書き込んでも影響はありません。 • リードモディファイライト系命令では常に"1"が読み出されます。
bit 14	MCTIE: メインクロック タイマ割込み 許可ビット	• このビットは , メインクロックタイマによるインターバル割込みを許可するために使用します。 • このビットに "1" を書き込むと割込みが許可され , "0" を書き込むと割込みが禁止されます。 • どのリセットでも"0"に初期化されます。
bit 15	予約	• このビットに常に "0" を書き込んでください。 • このビットの読出し値は未定義です。 • このレジスタへのリードモディファイライト系命令は無効です。

10.3.2 メインクロックタイマの動作

メインクロックタイマは指定されたインターバルで割込みを生成するためのインターバルタイマとして， また, メイン発振の安定待ちタイマとして機能します。

メインクロックカウンタ

メインクロックカウンタは , メインクロック CLKMC をクロックとして使用する 23 ビットカウンタによって構成されます。メインクロックがアクティブであるとき , メインクロックカウンタは常にカウントを継続します。

メインクロックカウンタは , メインクロックが停止されたとき (ストップモードへの遷移 , または CKSR: MCE ビットによるメインクロックの禁止と CKMR: MCM ビットによる確認), MCTCR レジスタの MCTR ビットへの "0" の書込み , 電源リセット , 外部リセット , およびメインクロック停止検出リセットによってクリアされます。

＜注意事項＞

外部リセットは , RSTX 入力端子の立下りエッジを検出することによってメインクロックカウンタをクリアおよび停止します。CR クロックで 700 サイクル後に (外部リセット延長時間が終了して), メインクロックカウンタリセットが解除され , 外部リセットがまだアサートされている場合でもカウンタはカウントを開始します。したがって , メインクロック安定待ち時間は , RSTX がまだアクティブであるときでも開始します。この動作は CR クロックカウンタおよびサブクロックカウンタと異なります。

インターバル割込み機能

メインクロックカウンタのキャリ信号に従って , 指定したインターバルで割込みが生成されます。MCTCR レジスタの MCTI[3:0] ビットで指定されたインターバルで MCTIF フラグがセットされます。メインクロックタイマをクリアし , カウントが開始されたとき , インターバル時間が開始されます。

メインクロックが停止されたとき (ストップモードへの遷移 , または CKSR: MCE ビットによるメインクロックの禁止と CKMR: MCM ビットによる確認), メインクロックタイマが復帰時のメイン発振の安定待ちタイマとして使用されます。したがって, メインクロックが停止したときメインクロックカウンタは直ちにクリアされます。

10.4 サブクロックタイマ

サブクロックタイマは 17 ビットカウンタと制御レジスタによって構成されます。17 ビットカウンタはサブクロック CLKSC を分周します。サブクロックタイマは, サブクロックカウンタのキャリ信号により, 指定したインターバルで割込みを発行します。

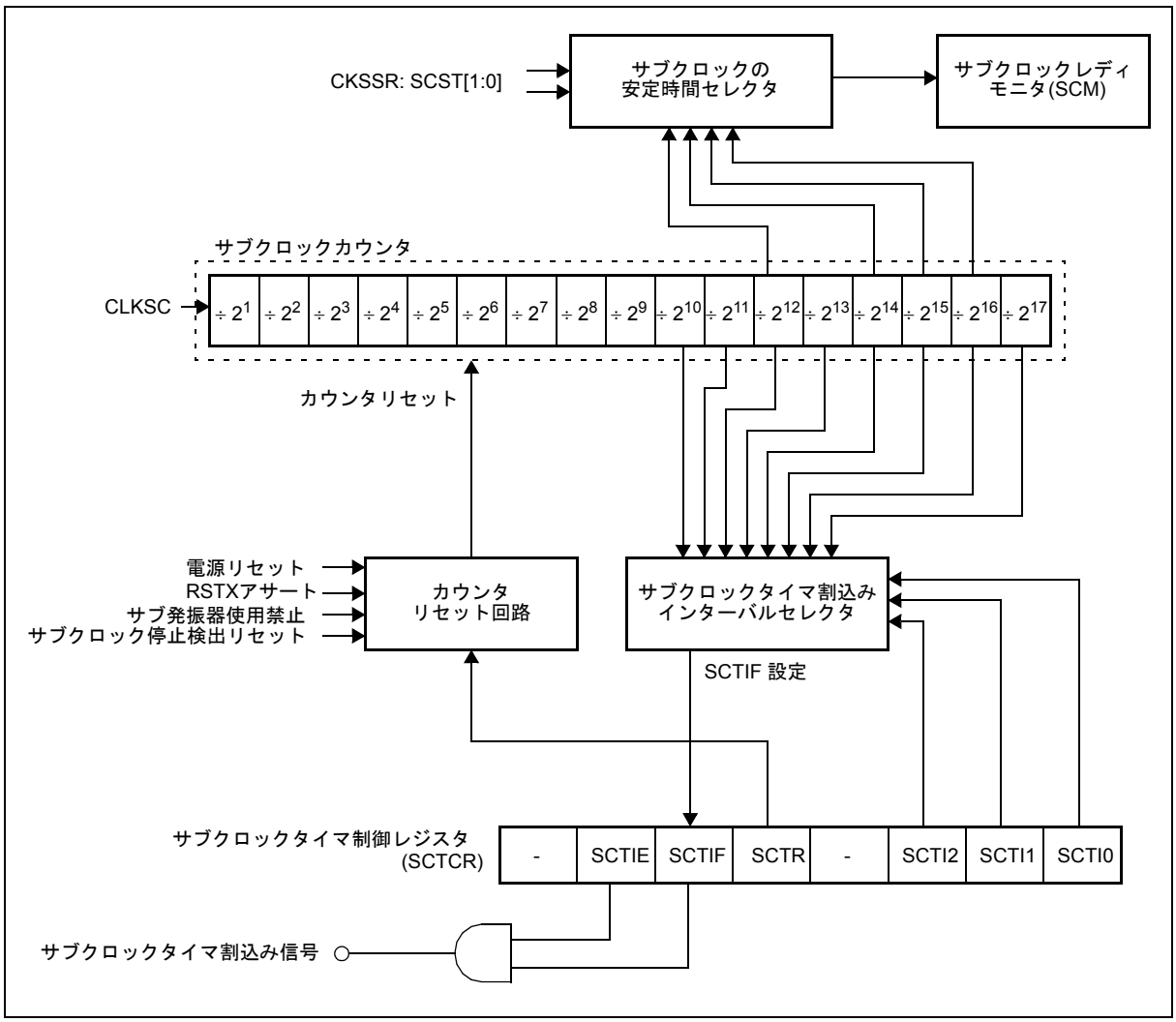
サブクロックタイマの概要

電源または外部リセットによってサブクロックタイマは 0 に初期化されます。サブクロックタイマは , サブクロックが禁止されたとき (ストップモードへの遷移時 , または SCE を "0" にセット), または SCTCR レジスタの SCTR ビットに "0" を書き込むことによって初期化されます。サブクロックタイマはサブクロックが供給されている間はカウントを継続します。サブクロックタイマはサブクロックの安定待ち時間を測定するために使用し , インターバル割込みを生成するためにも使用できます。

サブクロックタイマのブロックダイヤグラム

Figure 10-5 は , サブクロックタイマのブロックダイヤグラムを示しています。

Figure 10-5. サブクロックタイマのブロックダイヤグラム



10.4.1 サブクロックタイマ制御レジスタ(SCTCR)

サブクロックタイマ制御レジスタ (SCTCR) は, サブクロックタイマインターバル割込み機能を制御し, サブクロックタイマをリセットするために使用します。

サブクロックタイマ制御レジスタ(SCTCR)の構成

Figure 10-6 は , サブクロックタイマ制御レジスタ (SCTCR) の構成を示し , Table 10-3 は各ビットの機能を説明しています。

Figure 10-6. サブクロックタイマ制御レジスタ(SCTCR)の構成

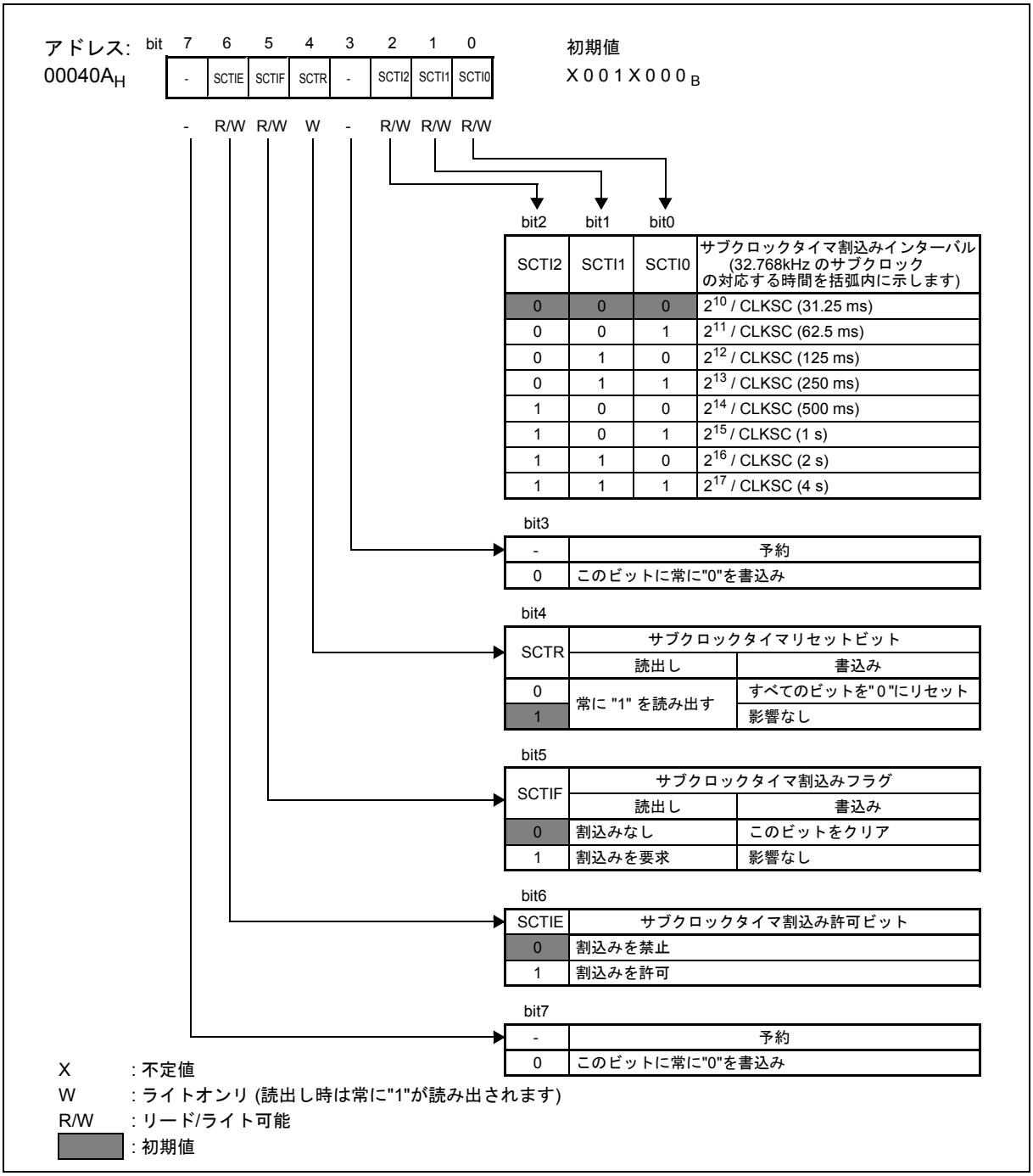


Table 10-3. サブクロックタイマ制御レジスタ(SCTCR)の各ビットの機能の説明

<table>
<tr><th colspan="2">ビット名</th><th>機能</th></tr>
<tr><td>bit 0 ～ bit 2</td><td>SCTI2 ～ SCTI0:
サブクロック
タイマ割込み
インターバル
選択ビット</td><td>
• これらのビットは , 次の表に従ってサブクロックタイマ割込みインターバルを制御します。
<table>
<tr><th colspan="3">bit2 bit1 bit0</th><th rowspan="2">サブクロックタイマ割込みインターバル
(32.768kHz のサブクロックの対応する時間を括弧内に示しています)</th></tr>
<tr><th>SCTI2</th><th>SCTI1</th><th>SCTI0</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>2^{10}/ CLKSC (31.25 ms)</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>2^{11}/ CLKSC (62.5 ms)</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>2^{12} / CLKSC (125 ms)</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>2^{13} / CLKSC (250 ms)</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>2^{14} / CLKSC (500 ms)</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>2^{15} / CLKSC (1 s)</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>2^{16}/ CLKSC (2 s)</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>2^{17}/ CLKSC (4 s)</td></tr>
</table>
• これらのビットは , どのリセットでも "000" に初期化されます。
• これらのビットにデータが書き込まれるとき, 同時にbit5 (SCTIF) をクリアしてください。
</td></tr>
<tr><td>bit 3</td><td>予約</td><td>• このビットに常に "0" を書き込んでください。
• このビットの読出し値は未定義です。
• このレジスタへのリードモディファイライト系命令は無効です。</td></tr>
<tr><td>bit 4</td><td>SCTR:
サブクロック
タイマリセット
ビット</td><td>• このビットは , サブクロックカウンタのすべてのビットをクリアします。
• "0" を書き込むとサブクロックカウンタがクリアされます。
• "1" を書き込んでも影響はありません。
• このビットからは常に "1" が読み出されます。
(注意事項)
このビットに"0"を書き込んでサブクロックタイマをクリアする場合は, SCTIF ビットにも "0" を書き込んでサブクロックタイマの割込みフラグをクリアしてください。</td></tr>
<tr><td>bit 5</td><td>SCTIF:
サブクロック
タイマ割込み
フラグ</td><td>• これはサブクロックタイマの割込み要求フラグです。
• このビットは SCTI2～SCTI0 ビットによって指定された各インターバルで"1"にセットされます。
• 割込み許可ビット SCTIE が "1" にセットされている場合 , このビットが "1" にセットされたとき割込み要求が発行されます。
• このビットは "0" の書込み , およびいずれかのリセットによってクリアされます。
• サブクロックタイマをリセットするときに , このビットも必ずクリアしてください。
• "1" を書き込んでも影響はありません。
• リードモディファイライト命令では常に"1"が読み出されます。</td></tr>
<tr><td>bit 6</td><td>SCTIE:
サブクロック
タイマ割込み
許可ビット</td><td>• このビットは , サブクロックタイマを基にインターバル割込みを許可するために使用します。
• このビットに "1" を書き込むと割込みが許可され , "0" を書き込むと割込みが禁止されます。
• どのリセットでも"0"に初期化されます。</td></tr>
<tr><td>bit 7</td><td>予約</td><td>• このビットに常に "0" を書き込んでください。
• このビットの読出し値は未定義です。
• このレジスタへのリードモディファイライト系命令は無効です。</td></tr>
</table>

10.4.2 サブクロックタイマの動作

サブクロックタイマは指定されたインターバルで割込みを生成するためのインターバルタイマとして, また, サブ発振の安定待ちタイマとして機能します。

サブクロックカウンタ

サブクロックカウンタは , サブクロック CLKSC をクロックとして使用する 17 ビットカウンタによって構成されます。サブクロックがアクティブであるとき , サブクロックカウンタは常にカウントを継続します。

サブクロックカウンタは, サブクロックが停止されたとき (ストップモードへの遷移, または CKSR: SCE ビットによるサブクロックの禁止と CKMR: SCM ビットによる確認), SCTCR レジスタの SCTR ビットへの "0" の書込み , 電源リセット , 外部リセット , およびサブクロック停止検出リセットによってクリアされます。

＜注意事項＞

外部リセットがアサートされている間はサブクロックカウンタはクリアされ , 停止されます。そのためサブクロック安定時間は RSTX によって引き延ばされ , 外部リセットがアサート解除された後でカウントが開始されます。この動作はメインクロックカウンタと異なります。

インターバル割込み機能

サブクロックカウンタのキャリ信号に従って , 指定したインターバルで割込みが生成されます。SCTCR レジスタの SCTI[2:0] ビットで指定されたインターバルで SCTIF フラグがセットされます。サブクロックタイマをクリアし , カウントが開始されたとき , インターバル時間が開始されます。

サブクロックが停止されたとき(ストップモードへの遷移, または CKSR: SCE ビットによるサブクロックの禁止と CKMR: SCM ビットによる確認), サブクロックタイマが復帰時のサブ発振の安定待ちタイマとして使用されます。したがって, サブクロックが停止したときサブクロックカウンタは直ちにクリアされます。

11. ウォッチドッグタイマとウォッチドッグリセット

ウォッチドッグタイマおよびリセットの機能と動作について説明します。

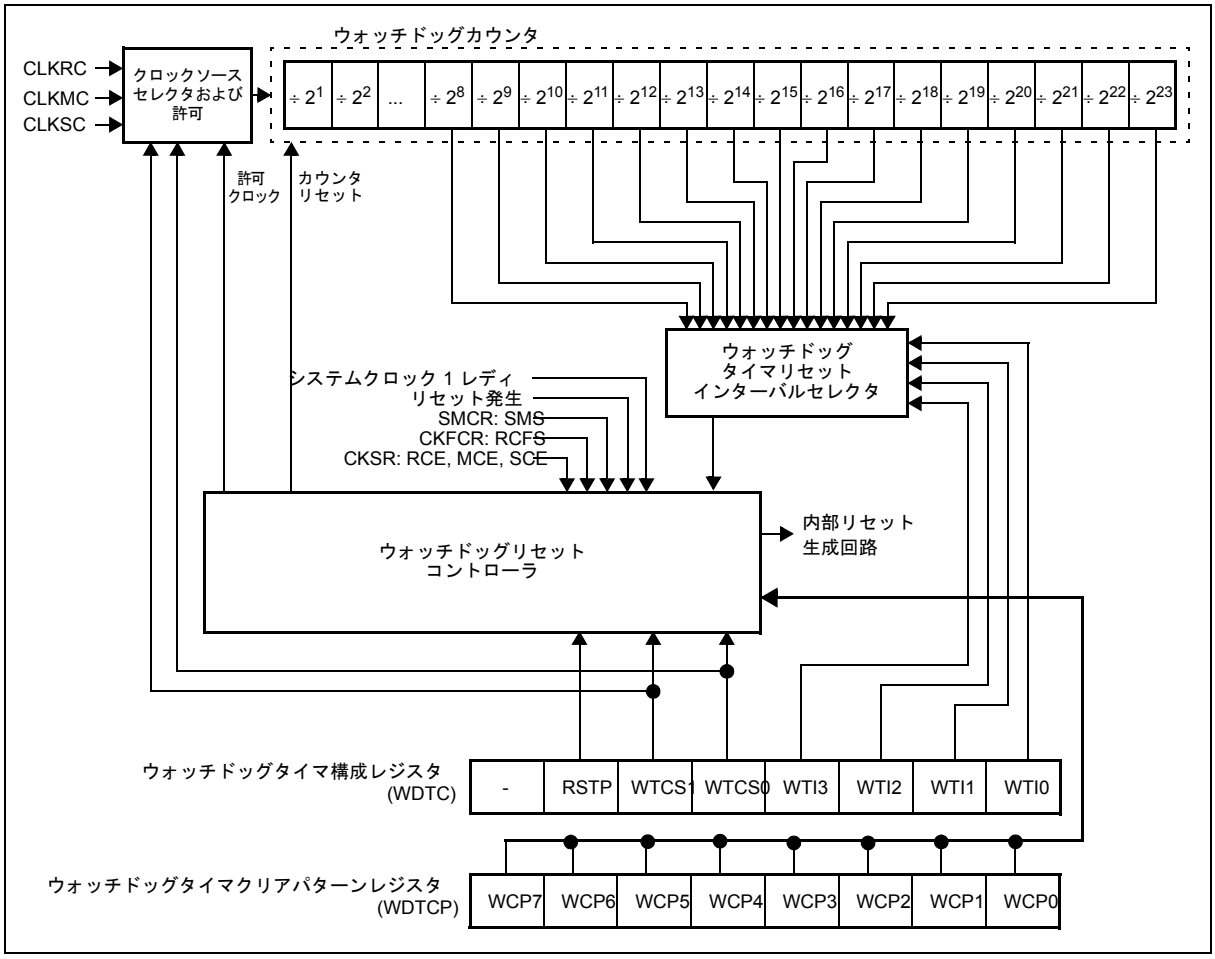
11.1 ウォッチドッグタイマおよびリセットの概要

ウォッチドッグ回路は，23 ビットのウォッチドッグカウンタ，複数の制御レジスタ，およびウォッチドッグリセットコントローラで構成されています。23 ビットのウォッチドッグカウンタは, クロックソースとして CR クロック, メインクロック, またはサブクロックのいずれかを使用します。

ウォッチドッグ回路のブロックダイヤグラム

Figure 11-1 は , ウォッチドッグ回路のブロックダイヤグラムを示しています。

Figure 11-1. ウォッチドッグ回路のブロックダイヤグラム



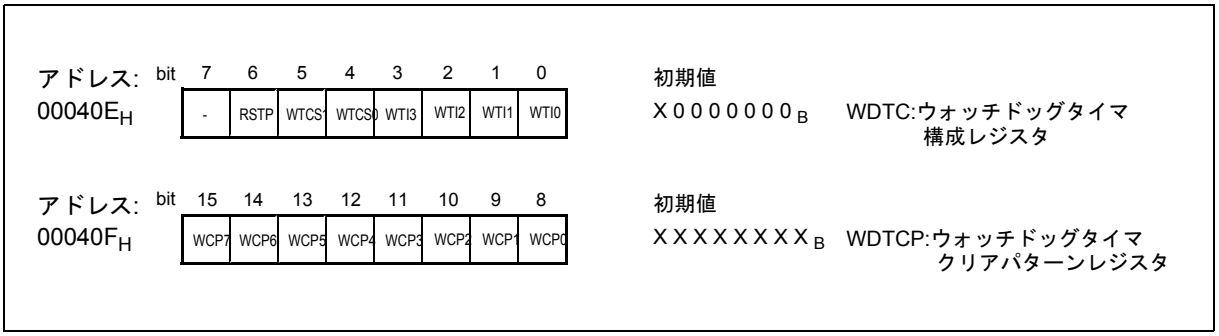
11.2 ウォッチドッグタイマ制御レジスタ

本項では, ウォッチドッグタイマ制御レジスタの一覧および, 各レジスタの機能について詳細に説明します。

ウォッチドッグタイマ制御レジスタ

ウォッチドッグタイマには , ウォッチドッグタイマ構成レジスタ (WDTC) とウォッチドッグタイマクリアパターンレジスタ (WDTCP) の 2 つの制御レジスタがあります。Figure 11-2 は , ウォッチドッグタイマ制御レジスタの概要を示しています。

Figure 11-2. ウォッチドッグタイマ制御レジスタ



11.2.1 ウォッチドッグタイマ構成レジスタ(WDTC)

ウォッチドッグタイマ構成レジスタ (WDTC) は, クロックソースおよびウォッチドッグインターバルの選択,およびウォッチドッグタイマ起動のために使用します。

ウォッチドッグタイマ構成レジスタ(WDTC)の構成

Figure 11-3 は, ウォッチドッグタイマ構成レジスタ (WDTC) の構成を示し, Table 11-1 は, 各ビットの機能を説明しています。

Figure 11-3. ウォッチドッグタイマ構成レジスタ(WDTC)の構成

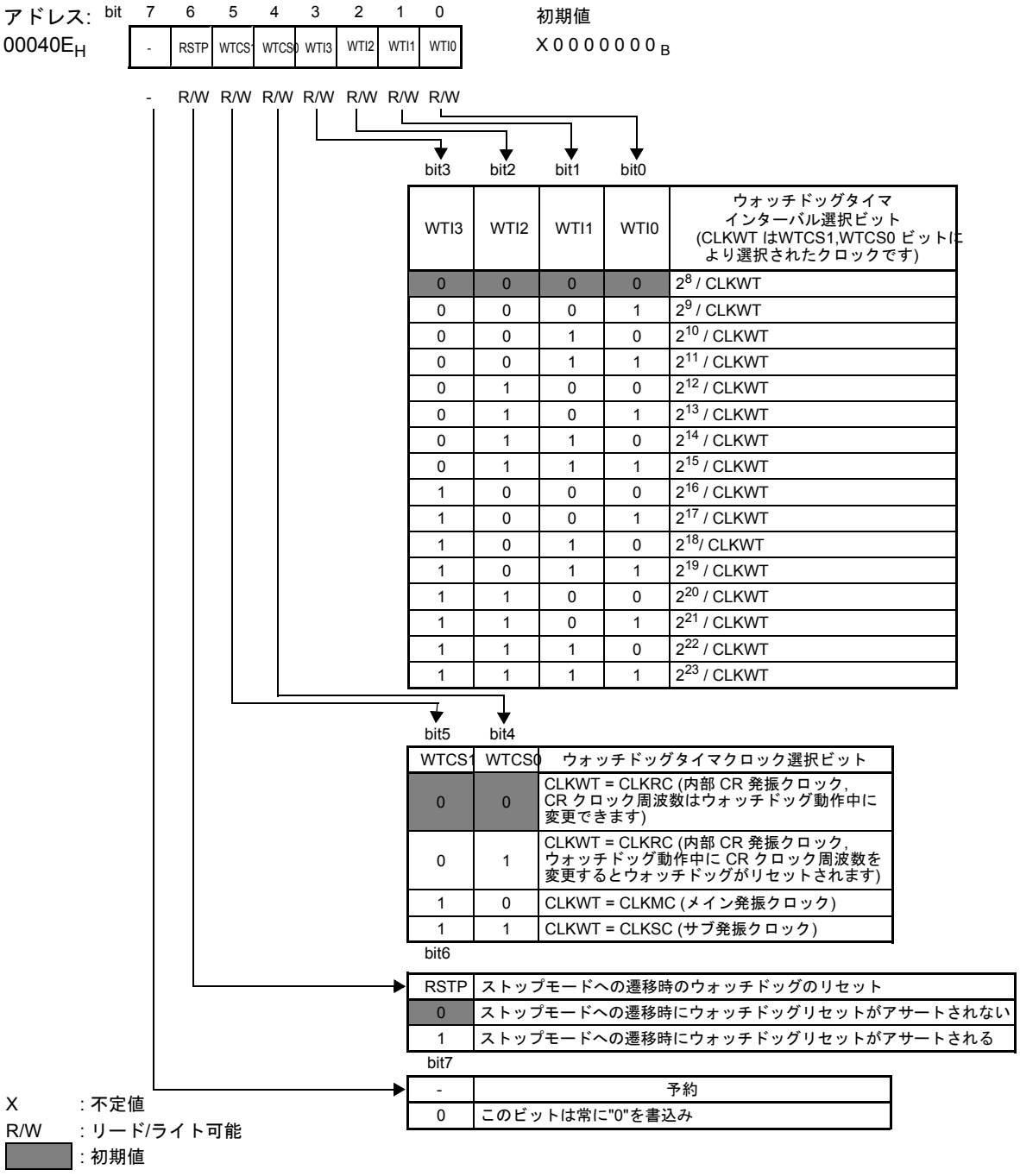


Table 11-1. ウォッチドッグタイマ構成レジスタ(WDTC)の各ビットの機能説明 (Sheet 1 of 2)

<table>
<tr><th colspan="2">ビット名</th><th>機能</th></tr>
<tr><td>bit 0 ～ bit 3</td><td>WTI3 ～ WTI0:
ウォッチドッグタイマ
インターバル
選択ビット</td><td>

これらのビットは , 下の表に従ってウォッチドッグタイマインターバルを選択します。

bit3 bit2 bit1 bit0
<table>
<tr><th rowspan="2">WTI3</th><th rowspan="2">WTI2</th><th rowspan="2">WTI1</th><th rowspan="2">WTI0</th><th colspan="3">ウォッチドッグタイマインターバルは選択クロックソースによって決まります。</th></tr>
<tr><th>CR クロック選択時 (2MHz/100kHz の通常の CR クロック周波数の対応する時間)</th><th>メインクロック選択時 (4MHz のメインクロック周波数の対応する時間)</th><th>サブクロック選択時 (32.768kHz のサブクロック周波数の対応する時間)</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>2^8 / CLKRC (最大 128µs/2.5ms)</td><td>2^8 / CLKMC (最大 64µs)</td><td>2^8 / CLKSC (7.8ms)</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>2^9 / CLKRC (最大 256µs/5.1ms)</td><td>2^9 / CLKMC (最大 128µs)</td><td>2^9 / CLKSC (15.6ms)</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>2^{10} / CLKRC (最大 512µs/10.2ms)</td><td>2^{10} / CLKMC (最大 256µs)</td><td>2^{10} / CLKSC (31.25ms)</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>2^{11} / CLKRC (最大 1ms/20.5ms)</td><td>2^{11} / CLKMC (最大 512µs)</td><td>2^{11} / CLKSC (62.5ms)</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>2^{12} / CLKRC (最大 2ms/41ms)</td><td>2^{12} / CLKMC (最大 1ms)</td><td>2^{12} / CLKSC (125ms)</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>2^{13} / CLKRC (最大 4ms/82ms)</td><td>2^{13} / CLKMC (最大 2ms)</td><td>2^{13} / CLKSC (250ms)</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>2^{14} / CLKRC (最大 8ms/164ms)</td><td>2^{14} / CLKMC (最大 4ms)</td><td>2^{14} / CLKSC (500ms)</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>2^{15} / CLKRC (最大 16ms/328ms)</td><td>2^{15} / CLKMC (最大 8ms)</td><td>2^{15} / CLKSC (1s)</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>2^{16} / CLKRC (最大 32ms/655ms)</td><td>2^{16} / CLKMC (最大 16ms)</td><td>2^{16} / CLKSC (2s)</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>2^{17} / CLKRC (最大 65ms/1.3s)</td><td>2^{17} / CLKMC (最大 32ms)</td><td>2^{17} / CLKSC (4s)</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>2^{18} / CLKRC (最大 131ms/2.6s)</td><td>2^{18} / CLKMC (最大 65ms)</td><td>2^{18} / CLKSC (8s)</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>2^{19} / CLKRC (最大 262ms/5.2s)</td><td>2^{19} / CLKMC (最大 131ms)</td><td>2^{19} / CLKSC (16s)</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>2^{20} / CLKRC (最大 524ms/10.4s)</td><td>2^{20} / CLKMC (最大 262ms)</td><td>2^{20} / CLKSC (32s)</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>2^{21} / CLKRC (最大 1.05s/21s)</td><td>2^{21} / CLKMC (最大 524ms)</td><td>2^{21} / CLKSC (64s)</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>2^{22} / CLKRC (最大 2.1s/42s)</td><td>2^{22} / CLKMC (最大 1.049s)</td><td>2^{22} / CLKSC (128s)</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>2^{23} / CLKRC (最大 4.2s/84s)</td><td>2^{23} / CLKMC (最大 2.097s)</td><td>2^{23} / CLKSC (256s)</td></tr>
</table>

これらのビットは , どのリセットによっても "0000" に初期化されます。
(最初にウォッチドッグタイマクリアパターンレジスタに書き込むことによって) ウォッチドッグリセット機能を起動した後でこれらのビットの設定を変更することはできません。16 ビットアクセスによって , これらのビットとウォッチドッグタイマリセット起動設定を同時に行うことは可能です。
リセットの後, これらのビットは再び書込み可能になります。

</td></tr>
</table>

Table 11-1. ウォッチドッグタイマ構成レジスタ(WDTC)の各ビットの機能説明 (Sheet 2 of 2)

<table>
<tr><th colspan="2">ビット名</th><th>機能</th></tr>
<tr><td>bit 4 ～ bit 5</td><td>WTCS1, WTCS0:
ウォッチドッグタイマクロック選択ビット</td><td>
• これらのビットは , 下の表に従ってウォッチドッグタイマのクロックソースを選択します。
<table>
<tr><th colspan="2">bit5 bit4</th><th>ウォッチドッグタイマクロック選択ビット</th></tr>
<tr><td>WTCS1</td><td>WTCS0</td><td></td></tr>
<tr><td>0</td><td>0</td><td>CLKWT = CLKRC (内部 CR 発振クロック , CR クロック周波数はウォッチドッグ動作中に変更可能)</td></tr>
<tr><td>0</td><td>1</td><td>CLKWT = CLKRC (内部 CR 発振クロック , ウォッチドッグ動作中に CR クロック周波数を変更すると , ウォッチドッグがリセットされる)</td></tr>
<tr><td>1</td><td>0</td><td>CLKWT = CLKMC (メイン発振クロック)</td></tr>
<tr><td>1</td><td>1</td><td>CLKWT = CLKSC (サブ発振クロック)</td></tr>
</table>
• これらのビットは , どのリセットでも "00" に初期化されます。
• (最初にウォッチドッグタイマクリアパターンレジスタに書き込むことによって) ウォッチドッグリセット機能を起動した後でこれらのビットの設定を変更することはできません。16 ビットアクセスによって , これらのビットとウォッチドッグタイマリセット起動設定を同時に行うことは可能です。
• リセットの後 , これらのビットは再び書込み可能になります。
• WTCS[1:0] を "00" にセットすると内部 CR 発振器が選択され , ウォッチドッグリセット機能の起動後に (CKFCR: RCFS ビットに書き込むことによって) CR クロック周波数を変更できます。CR クロック周波数を変更すると , ウォッチドッグインターバル時間も変更されます。
• WTCS[1:0]を"01"にセットすると内部 CR 発振器が選択されますが,ウォッチドッグリセット機能の起動後に(CKFCR: RCFS ビットに書き込むことによって)CR クロック周波数を変更することは禁止です。この場合, CR クロック周波数を変更するとウォッチドッグがリセットされます。
</td></tr>
<tr><td>bit 6</td><td>RSTP:
ストップモードへの遷移時のウォッチドッグのリセット</td><td>
• どのリセットでも "0" に初期化されます。
• (最初にウォッチドッグタイマクリアパターンレジスタに書き込むことによって) ウォッチドッグリセット機能を起動した後でこのビットの設定を変更することはできません。16 ビットアクセスによって , これらのビットとウォッチドッグタイマリセット起動設定を同時に行うことは可能です。
• リセットの後 , このビットは再び書込み可能になります。
• このビットを "0" にセットすると , ストップモードへの遷移が許可されます。
• このビットを "1" にセットし , SMCR: SMS (ストップモード) に "11" を書き込むと , ウォッチドッグがリセットされます。
• アプリケーションでストップモードを使用しない場合だけ, このビットを "1" にセットします。
</td></tr>
<tr><td>bit 7</td><td>予約</td><td>
• このビットに常に "0" を書き込みます。
• このビットのリード値は不定です。
• このレジスタへのリードモディファイライト系命令は無効です。
</td></tr>
</table>

11.2.2 ウォッチドッグタイマクリアパターンレジスタ(WDTCP)

ウォッチドッグタイマクリアパターンレジスタ (WDTCP) は, ウォッチドッグリセット機能の起動や, ウォッチドッグカウンタをクリアするために使用します。

ウォッチドッグタイマクリアパターンレジスタ(WDTCP)の構成

Figure 11-4 は, ウォッチドッグタイマクリアパターンレジスタ (WDTCP) の構成を示し, Table 11-2 は, 各ビットの機能を説明しています。

Figure 11-4. ウォッチドッグタイマクリアパターンレジスタ(WDTCP)の構成

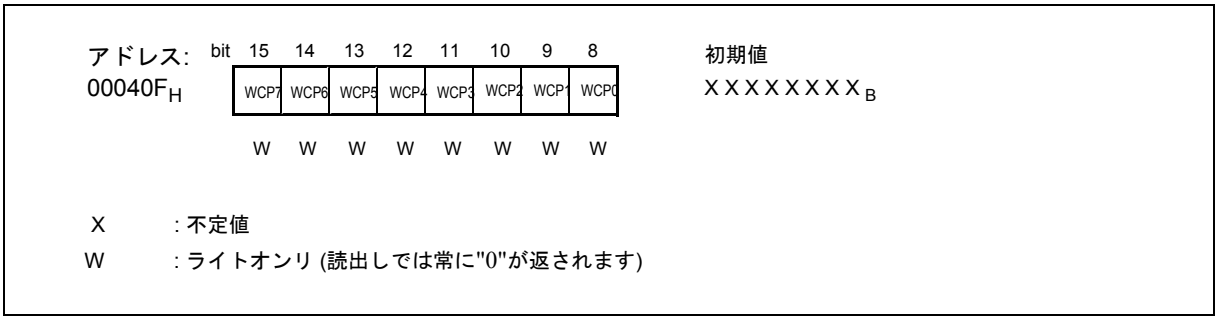


Table 11-2. ウォッチドッグタイマクリアパターンレジスタ(WDTCP)の各ビットの機能説明

ビット名		機能
bit 8 ～ bit 15	WCP7 ～ WCP0: ウォッチドッグクリアパターンビット	• ウォッチドッグリセット機能はこのレジスタへの書込みによって起動されます。 • ウォッチドッグリセット機能が起動された後 , この機能はリセット (どのタイプのリセットでもかまいません) によってのみ無効にできます。 • このレジスタへの書込みの後では , ウォッチドッグタイマ構成レジスタ (WDTC) の内容を変更できなくなります。 • リセット後初めて WDTCP レジスタへ書き込まれた値に従って , ウォッチドッグタイマは以下のいずれかのウォッチドッグ動作モードで動作します。 1. "00_H でクリア , ほかの値でリセットをアサート " WDTCP に 00_H を書込み : このモードでは , ウォッチドッグカウンタは , WDTCP レジスタへの 00_H の書込みによってクリアされます。00_H 以外の値を書き込むと , ウォッチドッグがリセットされます。 2. " 補数データでクリア " WDTCP への 00_H 以外の値の書込み : このモードでは , 最初に WDTCP レジスタに書き込まれた値が記憶されます。ウォッチドッグカウンタのクリアは , 前に書き込まれたレジスタ値の補数データの書込みによってのみ可能です。前と同じ値を書き込むと無効となり , ほかの値を書き込むとウォッチドッグがリセットされます。詳細については , 次の章を参照してください。 • このレジスタのリードでは常に 00_H が返されます。

11.3 ウォッチドッグタイマの動作

ウォッチドッグタイマおよびリセット機能を使用してユーザプログラムのハングアップを検出できます。
プログラムのハングアップなどの原因でウォッチドッグカウンタが指定した時間内にクリアされなかった場合, ウォッチドッグタイマはシステムをリセットします。

11.3.1 ウォッチドッグタイマの起動と停止

ウォッチドッグタイマは, ウォッチドッグタイマクリアパターンレジスタ WDTCP への書込みによって起動されます。リセット後に最初にこのレジスタに書き込まれたデータと , 同時またはその前にウォッチドッグタイマ構成レジスタ WDTC に書き込まれたデータによってウォッチドッグの機能が決まります。

ウォッチドッグタイマが起動された後 , このタイマはリセット (どのタイプのリセットでもかまいません) によってのみ停止できます。

11.3.2 ウォッチドッグカウンタのクリア

ウォッチドッグカウンタのクリアは , WDTCP レジスタに該当データを書き込むことによってのみ可能です。このデータは , ウォッチドッグ動作モードによって異なります。カウンタが WDTC: WTI[3:0] ビットで選択したインターバルの期限切れより前にクリアされない場合 , ウォッチドッグリセットがアサートされます。

11.3.3 ウォッチドッグカウンタの停止

ウォッチドッグカウンタの停止は , MCU がストップモードに移行するときにだけ可能です (このカウンタはクリアできません)。

■ ストップモードへの遷移

ストップモードではすべての発振器が無効にされます。したがって , ウォッチドッグカウンタも停止されます (クリアされません)。

■ ストップモードの解除

割込みによるウェイクアップ時に, ウォッチドッグのソースとして選択されているクロックが安定化し , CPU クロック CLKB が再起動されたとき (WDTC: WTCS[1:0] および CKSR: SC1S[1:0] 選択ビットによって選択されたクロックのクロックレディフラグがセットされる) にウォッチドッグカウンタがカウントを開始します。

＜注意事項＞

ウォッチドッグを常にアクティブにする必要がある場合は , ストップモードを使用しないでください。代わりに電流消費量が最小であるモードとして CR タイマモードを使用し , メインおよびサブ発振器を無効にし , CR クロックでウォッチドッグを実行することを推奨します。誤ってストップモードに遷移した場合に , ウォッチドックリセットさせるために , WDTC: RSTP を "1" に設定してください。

11.3.4 ウォッチドッグカウンタのクロックソースの選択

ウォッチドッグカウンタは , クロックソースとして CR クロック (CLKRC), メインクロック (CLKMC), またはサブクロック (CLKSC) のいずれかを使用できます。システム要求に合わせ , WDTC:WTCS[1:0] ビット設定することで , クロックソースの選択を行います。ウォッチドッグタイマを起動した後でクロックソースを変更することはできません。

CR クロックをクロックソースとして使用する必要があり , システムが動作中に CR クロック周波数の変更を必要とする場合は , WDTC: WTCS[1:0] ビットを "00" に設定します。ただし , CR クロック周波数を変更するとウォッチドッグインターバルも変更されます。

CR クロックをクロックソースとして使用する必要があり , 動作中に CR クロック周波数を変更しない場合は , WDTC: WTCS[1:0] ビットを "01" に設定します。誤って CR クロック周波数が変更された場合 , ウォッチドッグリセットがアサートされます。
ウォッチドッグカウンタは , 選択したクロックが安定化した (CKMR: RCM, MCM または SCM ビットが "1") ときにのみ動作できます。クロックが安定化するまでの間 (ストップモードが解除された後 , またはクロックソースが安定化する前にウォッチドッグカウンタが許可にされたとき), ウォッチドッグカウンタは停止します。

＜注意事項＞

ウォッチドッグカウンタは , 選択したクロックが安定化した (CKMR: RCM, MCM または SCM ビットが "1") ときにのみ動作できます。クロックが安定化するまでの間 (ストップモードが解除された後 , またはクロックソースが安定化する前にウォッチドッグカウンタが許可にされたとき), ウォッチドッグカウンタは停止します。クロックをウォッチドッグのための使用する前に , クロックが安定化していることを必ず確認してください。

11.3.5 スタンバイモード

スリープおよびタイマモードでは , ウォッチドッグタイマはその構成に従って動作し , クリアまたは停止されません。ストップモードでは , ウォッチドッグタイマは停止されますがクリアされません。ストップモード解除の後 , ウォッチドッグタイマは , ウォッチドッグタイマクロックソースと CLKS1 のクロックソースの両方が安定化されたとき動作を再開します。

11.3.6 ウォッチドッグのリセット

ウォッチドッグカウンタのクロックソースとして使用する発振器のクロック許可ビット (CKSR: RCE, MCE, SCE) を "0" (発振器を無効にする) にセットすると , ウォッチドッグがリセットされます。

11.3.7 ウォッチドッグタイマインターバルの設定

ウォッチドッグタイマインターバルを WDTC: WTI[3:0] ビットによって選択できます。ウォッチドッグタイマを起動した後は , インターバルを変更することはできません。

クロックソースとして CR 発振器を使用する場合 , インターバルの長さを計算するときこの発振器の周波数偏差を考慮してください。Table 11-1 に示す絶対時間は , 公称発振周波数に基づいています。CR クロック周波数の精度の詳細についてはデータシートを参照してください。

11.3.8 ウォッチドッグタイマの動作モードの選択

リセットの後 WDTCP レジスタへの最初の書込みアクセスによってウォッチドッグタイマが起動され, 動作モードが決定します。最初の書込みアクセス時に WDTCP に書き込まれたデータに従って , 次の 2 つの動作モードのどちらかが選択されます。

- 00_Hの書込みでクリア, ほかの値の書込みでウォッチドッグリセットをアサート

このモードは, リセット後の最初の書込みアクセス時に WDTCP レジスタに 00_Hを書き込むことによって選択されます。
このモードでは , ウォッチドッグカウンタは , WDTCP レジスタへの 00_H の書込みによってクリアされます。00_H 以外の値を書き込むと , ウォッチドッグがリセットされます。

- 補数データの書込みでクリア, 無効な値の書込みでウォッチドッグリセットをアサート

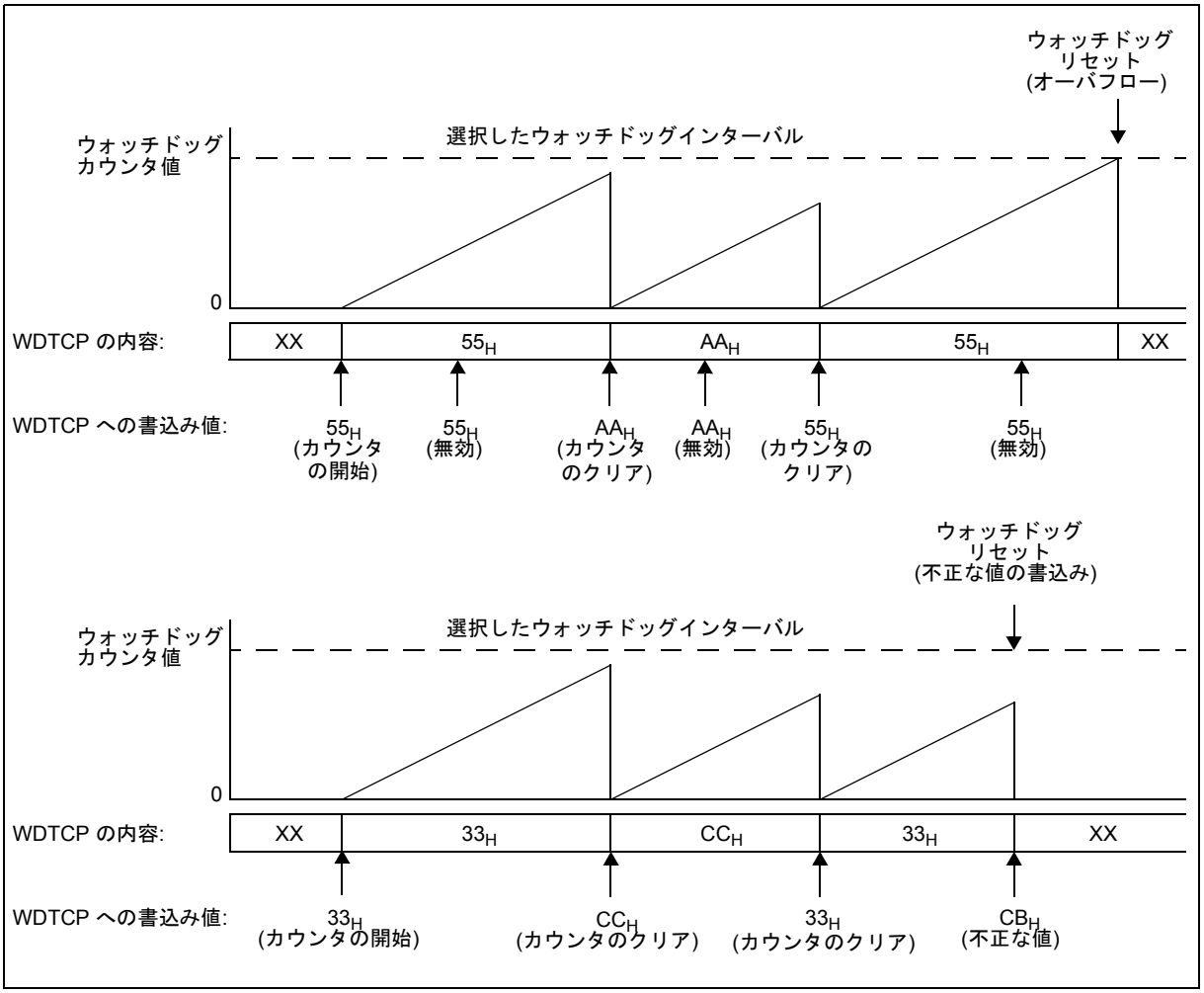
このモードは , リセットの後の最初の書込みアクセス時に WDTCP レジスタに 00_H 以外の値を書き込むことによって選択されます。
この最初の書込みアクセス時に書き込まれたデータが WDTCP レジスタに格納されます。
WDTCP レジスタに現在格納されているデータと同じデータを書き込んでも無効です。

現在の WDTCP の内容に対する補数データを書き込むと , ウォッチドッグカウンタがクリアされ , WDTCP レジスタの現在の値が置換されます。その後ウォッチドッグタイマをクリアするには , そのたびに補数データを書き込む必要があります (元のデータ - 補数データ - 元のデータ - 補数データ ...)。

現在のレジスタの内容またはその補数値以外のデータを WDTCP レジスタに書き込むと , ウォッチドッグがリセットされます。

下の図は , このモードでのウォッチドッグタイマの動作の例を示しています。

Figure 11-5. 補数モードでのウォッチドッグタイマの動作



11.3.9 ウォッチドッグタイマリセット要因

以下の場合にウォッチドッグタイマリセットがアサートされます。

- ウォッチドッグタイマが選択したインターバル内にクリアされなかった。
- WDTCP レジスタに無効のデータが書き込まれた。
- ウォッチドッグタイマ(CKSR:RCE,MCEまたはSCE)のソースクロックとして使用されるクロックの許可ビットに"0"が書き込まれた。
- WDTC:WTCS[1:0]が"01"にセットされているにもかかわらず, CRクロック周波数が変更された。
- WDTC:RSTPが"1"にセットされているにもかかわらず, ストップモードへの遷移を行った。

12. 外部バスインタフェース

外部バスインタフェースの機能と動作について説明します。

12.1 外部バスの概要

F^{2}MC-16FX MCU の外部バスインタフェースでは, 種々のアクセス方法およびアクセス領域を利用できます。

12.1.1 外部バスインタフェースの特長

外部バスインタフェースの特長の要約を示します。

外部バスインタフェースの特長

- 16bit長
- 最大24ビット長のアドレス出力(最大値はデバイスによって異なります)
- マルチプレクスモード
- ノンマルチプレクスモード(端子数の多いデバイスでの対応)
- 最大 6 つのチップセレクト信号, 4 つはメモリ領域でプログラミング可能
- レディ/ホールド機能
- 外部バスマスタが利用可能
- WRLX, WRHXプロトコルおよび WRX, UBX, LBX プロトコルをサポート
- タイミング設定可能

12.1.2 用語

ここでは, 外部バスインタフェースの基本的な用語について説明します。

■ バスモード

バスモードとは , 内部 ROM 動作および外部アクセス機能を制御するモードです。バスモードは , 外部バスモードレジスタ (EBM) の EAE[5:0] および ERE ビットによって設定されます。

■ アクセスモード

アクセスモードとは , 外部バス幅を制御します。

外部バスのデータ幅は , 外部領域構成レジスタ (EACL5 ～ EACL0) の BW ビットによって指定されます。

外バスインターフェース公開品の場合 , 外部バスモードレジスタの NMS ビットの設定によって , マルチプレクス / ノンマルチプレクスモードを切り換えることができます。この設定は , すべての外部領域に対して有効です。

■ 外部ブートベクタ

外部バスインターフェース公開品では , ブートベクタ (ユーザプログラム開始アドレス) およびモードバイトを , 内部データから (内部ベクタモード) ではなく , 外部メモリからフェッチできます。ブートベクタと共に読み出されるモードバイトは , 外部バスモードレジスタ (EBM) にコピーされ , 初期値を上書きします。

12.1.3 外部バス領域

チップセレク信号により最大6 つの外部バス領域に設定することが可能です。
16MBの領域をそれぞれに対応したチップセレクト信号により6つの外部領域に分割することが可能です。

外部バス領域

F^{2}MC-16FX MCU の 16MB アドレス領域は , 6 つの外部バス領域にそれぞれ設定レジスタおよびチップセレクト信号により分割されます。外部バス領域 0 および 1 のアドレス範囲は固定されていますが , ほかの領域のアドレス範囲は外部領域選択 (EAS) レジスタおよび外部領域構成 (EACH) レジスタの外部領域サイズ選択ビットにより , 上位 15 MB アドレス空間内で設定できます。

領域外のアドレスをアクセスしても, 外部バス機能は動作しません。

Table 12-1. 外部バス領域のアドレス範囲

外部領域	利用可能なアドレス領域	対応するチップセレクト	アドレス領域の初期値
外部領域 0	h'00.00f0...h'00.00ff	CS0	h'00.00f0...h'00.00ff (固定)
外部領域 1	h'00.0c00...(RAM-Start - 1)	CS1	h'00.0c00...(RAM-Start - 1) (固定)
外部領域 2	h'10.0000...h'ff.ffff	CS2	h'10.0000...h'3f.ffff (3MB)
外部領域 3		CS3	h'40.0000...h'7f.ffff (4MB)
外部領域 4		CS4	h'80.0000...h'bf.ffff (4MB)
外部領域 5		CS5	h'c0.0000...h'ff.ffff (=4MB, 外部リセットベクタを含む)

*: RAM-Start は , 内部 RAM の開始アドレスです (デバイスによって異なります)。

RAM のメモリサイズと開始アドレスについては , データシートを参照してください。

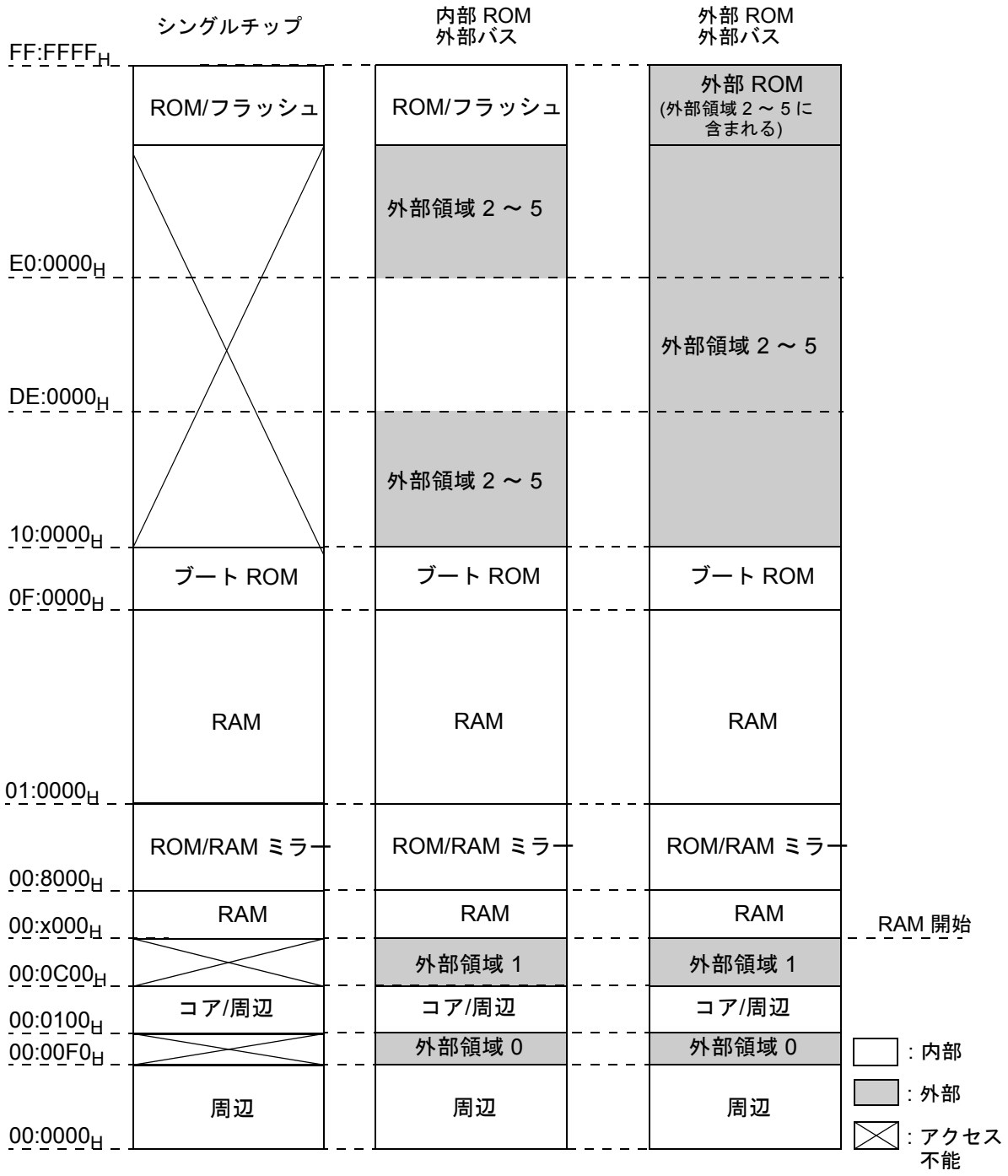
以下の章にデフォルトアドレス範囲の外部バス領域について説明します。

12.1.4 各バスモードのメモリ空間

Figure 12-1 は, 各バスモードのアクセス領域と物理アドレスの対応関係を示しています。

各バスモードのメモリ空間

Figure 12-1. 各バスモードのアクセス領域と物理アドレスの関係



12.2 外部バス構成レジスタ

外部バス機能の対応の為に , 下記のレジスタがあります。

- 外部バスモードレジスタ (EBM)
- 外部バスクロックおよび機能レジスタ (EBCF)
- 外部バスアドレス出力許可レジスタ [2:0] (EBAE[2:0])
- 外部バス制御信号レジスタ (EBCS)
- 外部領域 [5:0] 構成レジスタ (EACH/EACL[5:0])
- 外部領域選択レジスタ [5:2] (EAS[5:2])

メモリ内の外部バス構成レジスタの配列

Table 12-2. 外部バスレジスタ

アドレス	上位バイト	下位バイト
$00:06E0/1_H$	EACH0	EACL0
$00:06E2/3_H$	EACH1	EACL1
$00:06E4/5_H$	EACH2	EACL2
$00:06E6/7_H$	EACH3	EACL3
$00:06E8/9_H$	EACH4	EACL4
$00:06EA/B_H$	EACH5	EACL5
$00:06EC/D_H$	EAS3	EAS2
$00:06EE/F_H$	EAS5	EAS4
$00:06F0/1_H$	EBCF	EBM
$00:06F2/3_H$	EBAE1	EBAE0
$00:06F4/5_H$	EBCS	EBAE2

外部バス構成レジスタの概要

Figure 12-2. 外部バス構成レジスタ

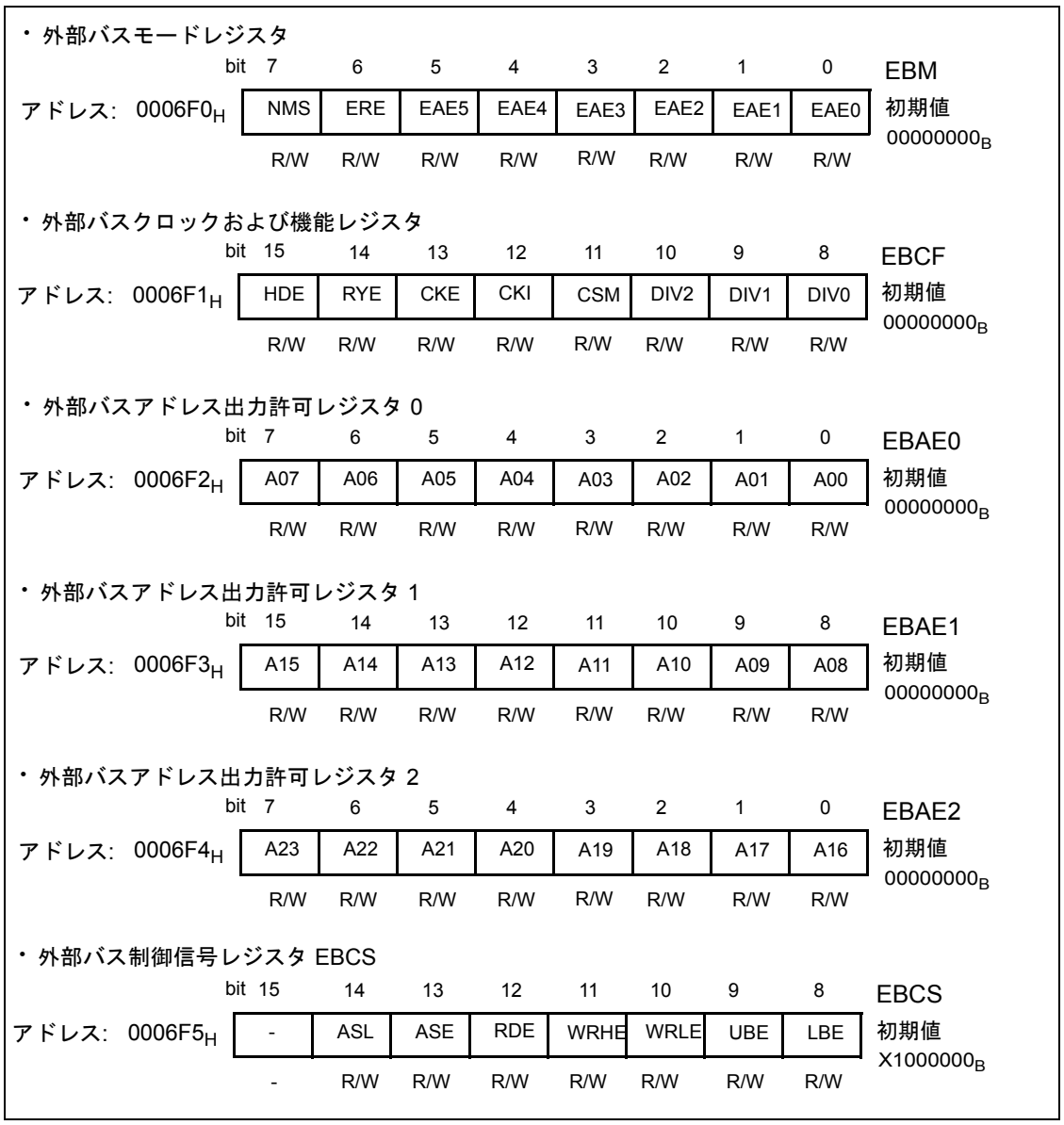


Figure 12-3. 外部バス構成レジスタ

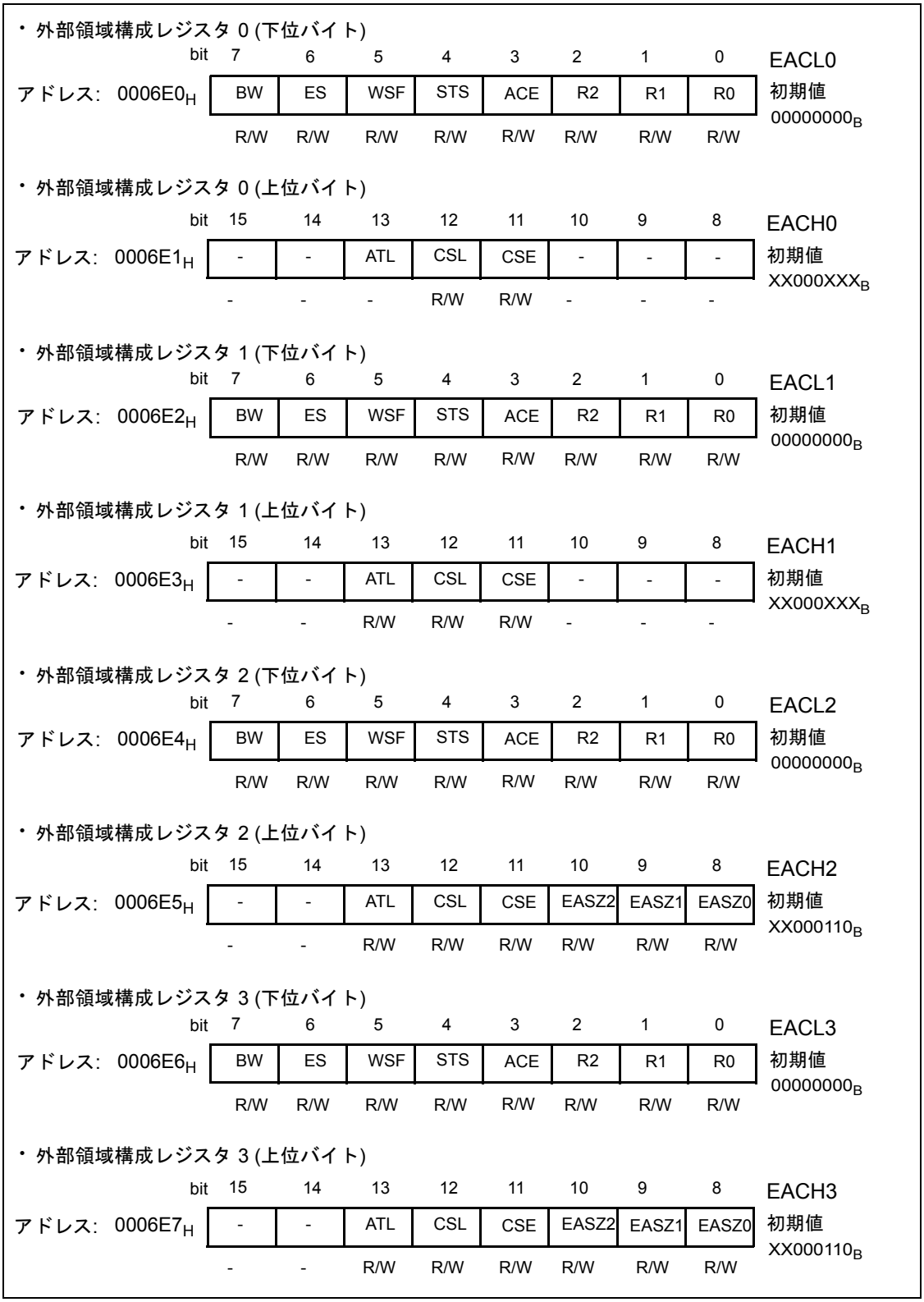
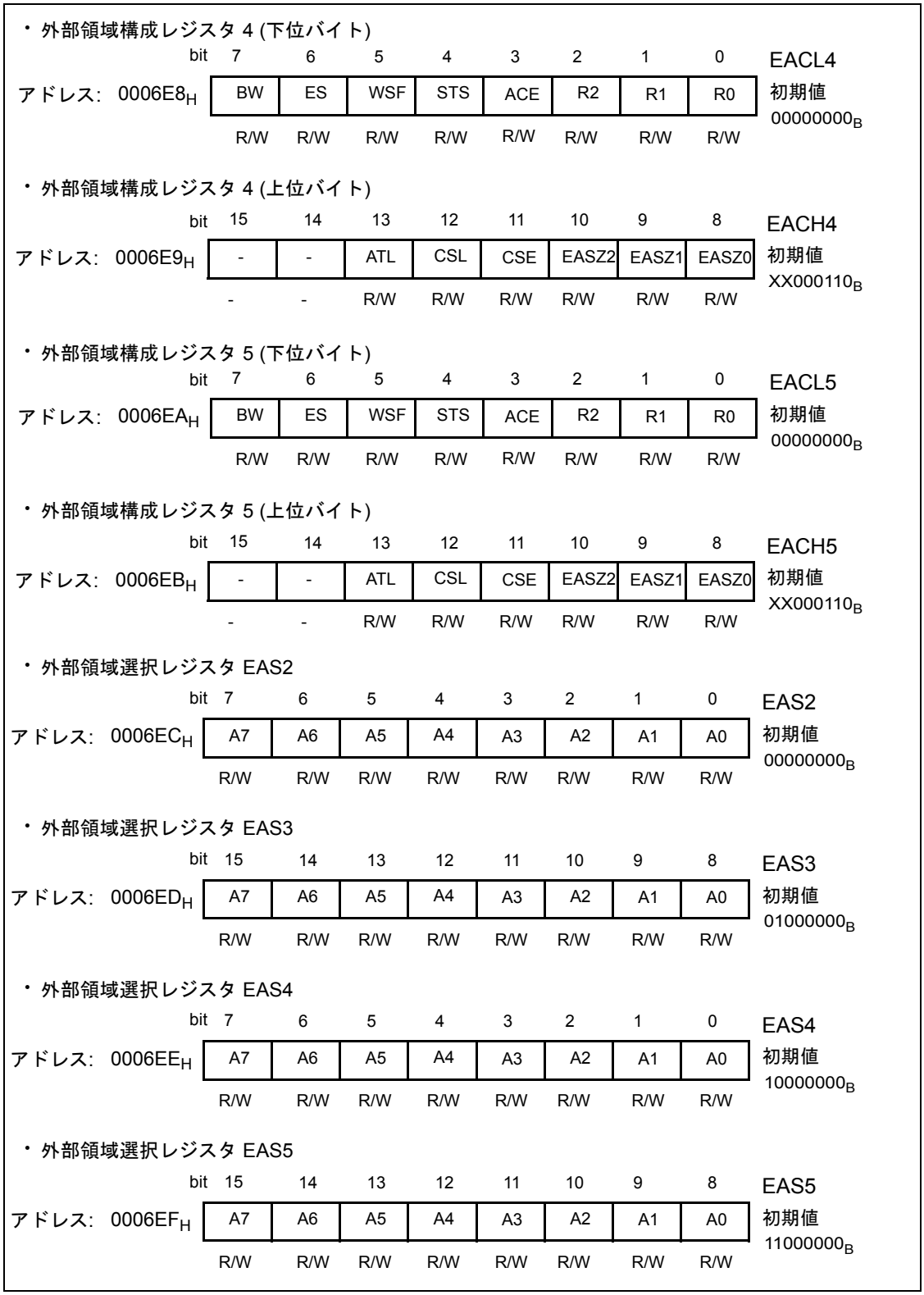


Figure 12-4. 外部バス構成レジスタ



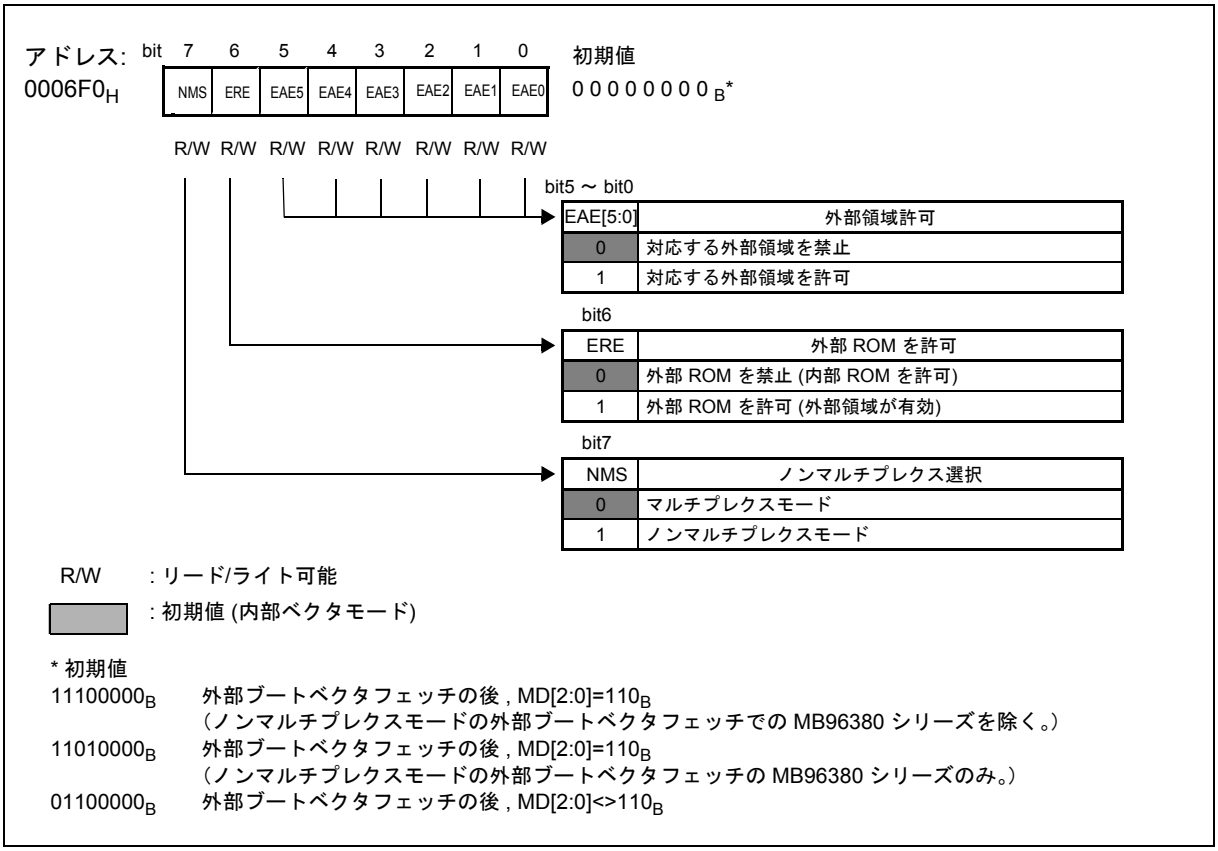
12.2.1 外部バスモードレジスタ(EBM)

外部バスモードレジスタは , 外部バスインタフェースのバスモードを定義します。外部ブートベクタフェッチでは , 外部バスモードバイトがアドレス $FFFFDF_H$ から外部的に読み出され , ブートベクタの読出し後に直接レジスタに書き込まれます。
EBM レジスタでは以下の設定を行うことができます。

- すべての外部バス領域の起動
- 内部 ROM アクセスと外部アクセスの切換え
- マルチプレクス/ノンマルチプレクスモード

外部バスモードレジスタ

Figure 12-5. 外部バスモードレジスタ(EBM)



＜注意事項＞

外部バスモードレジスタ (EBM) で , 外部領域アクセスを許可設定した直後に , 該当する外部アドレス領域 (CS 領域) に分岐 , またはリード・ライトすると正しく命令実行されません。外部バスモードレジスタ (EBM) に設定する命令の後 , 最低 1 つの NOP あるいは外部アドレス領域をアクセスしない命令を配置してください。

ビット名		機能
bit [5:0]	EAE5 ～ EAE0: 外部領域 許可	• EAE (外部領域許可) ビットは , 外部アドレス領域を有効にするのに使用します。 • "0" : 対応する外部アドレス領域を無効にする。この領域に対するほかの設定は無効になります。 • "1" : 対応する外部アドレス領域を有効にする。アドレス領域は , 対応する EACH/EAC および EAS レジスタの設定により定義され , EAE ビットを "1" にする前に , 設定する必要があります。 • 外部 ROM は , 外部領域が許可されている場合にのみ使用できます。 • EAE5 ～ EAE0 のビットの初期値は "0" です。 • 外部ブートベクタフェッチの前に , EAE5 は "1" にセットされます (MB96380 シリーズを除き , ノンマルチプレクスモードのスタートアップ時 , EAE4 は "1" にセットされます。) • ブートベクタフェッチの後, EBM レジスタは外部より読み込まれた外部バスモードバイトによって上書きされます。「12.5 外部ブートベクタフェッチ」の項も参照。
bit 6	ERE: 外部 ROM を 許可	• このビットにより , 内部 ROM のアドレス領域に内部 ROM アクセスするか外部アクセスかを選択します。 • "0" : 内部 ROM (フラッシュ) にアクセス (内部 ROM モード) • "1" : 外部メモリにアクセス (外部 ROM モード) • 外部バスアクセスは , 現在のアクセスのアドレスを含む EAE ビットによって外部アドレス領域がアクティブ化されている場合にのみ実行されます。 • 初期値は, "0"です。外部ブートベクタフェッチ時に, このビットは, "1" にセットされます。外部ブートベクタフェッチの後, EBM レジスタは外部的に読み出された外部バスモードバイトによって上書きされます。「12.5 外部ブートベクタフェッチ」の項も参照。
bit 7	NMS: ノンマルチプレクス バスモード	• このビットは , 外部領域へのマルチプレクス / ノンマルチプレクスモードを選択します。 • "0" : マルチプレクスモードを選択 • "1" : ノンマルチプレクスモードを選択 • 利用可能なモードはデバイスによって異なります。 • 初期値は, "0"です。外部ブートベクタフェッチ時は, このビットは, モード端子 MD[2:0] によって選択されたアクセスモードに従ってセットされます。ブートベクタフェッチの後, EBM レジスタは外部的に読み出された外部バスモードバイトによって上書きされます。「12.5 外部ブートベクタフェッチ」の項も参照。

外部ベクタモードでは , このレジスタの設定は , (ブートベクタの読出し後) 外部バスモードバイトをアドレス $FFFFDF_H$ から自動的に読出し , このデータをレジスタに書き込むことによって行われます (詳細は , 「12.5 外部ブートベクタフェッチ」を参照)。

外部バスモードレジスタは , ソフトウェアによって書き換え可能です。

MCU が内部ベクタモードで起動する場合は , レジスタの初期値は保持されます。

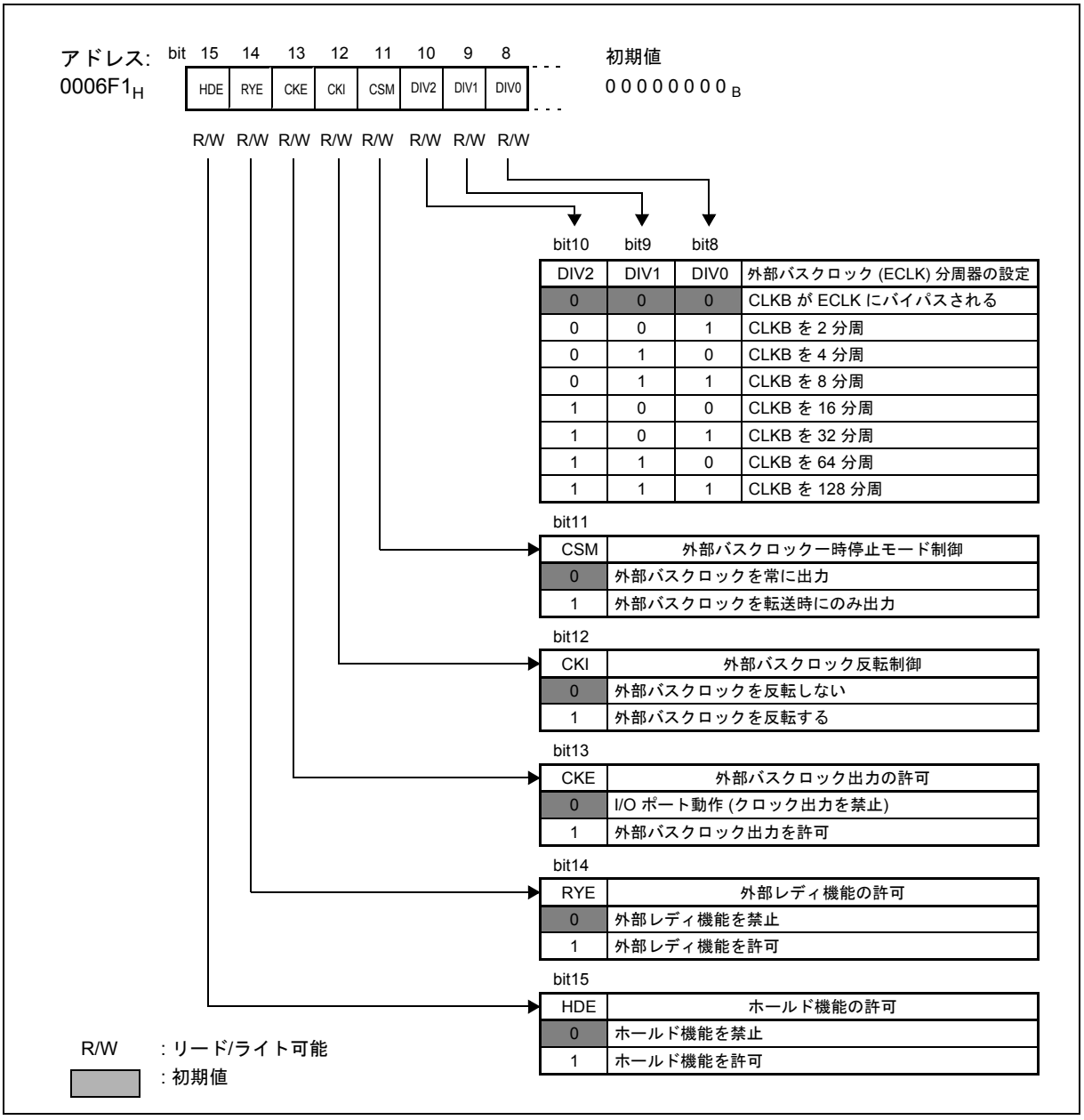
12.2.2 外部バスクロックおよび機能レジスタ(EBCF)

外部バスクロックおよび機能レジスタは , 外部バスクロック , 外部レディおよびホールド機能を制御します。

- ホールド機能
- 外部レディ機能
- クロック出力の許可
- クロックのアクティブエッジ
- クロック出力モード
- クロック分周器

外部バスクロックおよび機能レジスタ

Figure 12-6. 外部バスクロックおよび機能レジスタ(EBCF)



このレジスタは, MCU が外部ベクタモードで起動した場合には変更されません。
動作中にクロック構成ビット (EBCF:DIV) を変更した場合, 外部バス機能は保証されません。「6.2.4 クロック周波数制御レジスタ(CKFCR)」にある CKFCR:BCD ビットの構成も参照してください。

Table 12-3. 外部バスクロックおよび機能レジスタの各ビットの機能説明

ビット名		機能
bit15	HDE: ホールド機能の許可	• このビットは , 外部バスのホールド機能を許可します。 • "0" : ホールド機能を禁止する。 • "1" : ホールド機能を許可する。 • 初期値は , "0" です。 • ホールド機能が許可されている場合 , ホールドアクノリッジ端子 (HAKX) の出力が常に許可されます。入力許可レジスタ (PIER) およびデータ方向レジスタ (DDR) の対応する PIER を "1" に , DDR を "0" に設定し HRQ 端子を入力可能状態にしなければなりません。 • ホールド機能の詳細は , 「12.3.2 ホールド機能」を参照。
bit14	RYE: 外部レディ機能の許可	• このビットは , 外部バスの外部レディ機能を許可します。 • "0" : すべての外部領域の外部レディ機能を禁止する。 • "1" : すべての外部領域の外部レディ機能を許可する。 • 初期値は , "0" です。 • このビットの設定では RDY 端子の入力は許可されません。入力許可レジスタ (PIER) およびデータ方向レジスタ (DDR) で対応する PIER を "1" に , DDR を "0" に設定し , 入力許可状態にする必要があります。 • レディ機能の詳細は , 「12.3.1 レディ機能」を参照。
bit13	CKE: 外部 バスクロック 出力の許可	• このビットは , 外部バスクロック信号端子 (CLK) の出力を制御します。 • "0" : 外部バスクロック出力を禁止する。CLK 端子は DDR レジスタの設定により制御されます。 • "1" : 外部バスクロック出力を許可する。 • 初期値は , "0" です。 • 他のすべてのクロック設定が完了するまで外部バスクロックを許可してはいけません。
bit12	CKI: 外部 バスクロックの反転制御	• このビットは外部バスクロックの有効エッジを制御します。 • "0" : 立上りエッジが有効 • "1" : 立下りエッジが有効 • 初期値は , "0" です。 • CLK 端子への外部バスクロックの出力は , CKE および CSM ビットの設定によっても異なります。 • このビットの設定は , 対応する端子へのクロックの出力にのみ影響し , 内部機能には影響しません。
bit11	CSM: 外部 バスクロック 一時停止モード 制御	• このビットは , 対応する端子での外部バスクロック (ECLK) の発振を制御します。 • 初期値は , "0" です。 • "0" : 外部バスクロック (ECLK) は常に発振し , ホールド状態でも出力します。 • "1" : EBCF:DIV[2:0] の設定に依存します。 - EBCF:DIV[2:0] ≠ "000": 外部バスが非アクティブ (クロック一時停止モード) であるときに , 外部バスクロック (ECLK) は非アクティブレベルに設定されます。ホールド状態では , ECLK 端子はハイ・インピーダンスに設定されます。 - EBCF:DIV[2:0] = "000": 外部バスへのアクセスがない場合でも , 外部バスクロックは発振しています (クロック一時停止モードは利用不可)。しかしホールド状態では , DIV[2:0] の他の設定の場合 , ECLK 端子はハイインピーダンスに設定されます (詳細は「12.4 " 外部バスの使用に関する注意事項 "」を参照)。 • 非アクティブレベルは , CKI (外部バスクロック反転制御ビット) の設定によって異なります。 • このビットの設定は , 対応する端子へのクロックの出力にのみ影響します。

<table>
<tr><th colspan="2">ビット名</th><th>機能</th></tr>
<tr><td>bit10
～
bit8</td><td>DIV2 ～ DIV0:
外部
バスクロック
分周設定</td><td>

これらのビットは , 外部バスクロック信号端子 (ECLK) のクロック分周を制御します。リファレンスクロックは , クロック制御レジスタによって選択された CPU クロック (CLKB) です。外部バスのすべての動作は , このクロックの出力が許可されていない場合でも , 分周クロック (ECLK) を基にして行われます。クロック分周は全ての外部領域に対して共通して変更されます。

<table>
<tr><th>DIV2 ～ DIV0</th><th>外部バスクロック (ECLK) 分周設定</th></tr>
<tr><td>000</td><td>CLKB が ECLK にバイパスされる (1 分周)</td></tr>
<tr><td>001</td><td>ECLK = CLKB / 2</td></tr>
<tr><td>010</td><td>ECLK = CLKB / 4</td></tr>
<tr><td>011</td><td>ECLK = CLKB / 8</td></tr>
<tr><td>100</td><td>ECLK = CLKB / 16</td></tr>
<tr><td>101</td><td>ECLK = CLKB / 32</td></tr>
<tr><td>110</td><td>ECLK = CLKB / 64</td></tr>
<tr><td>111</td><td>ECLK = CLKB / 128</td></tr>
</table>

初期値は , "000" (ECLK = CLKB) です。
クロック分周が 1 分周以外に設定されている場合 , 外部バス上の転送は複数のシステムバスサイクルを必要とします。そのため , 外部バス転送時にシステムバスが停止します。つまり , システムバス上のすべての内部動作 (CPU, DMA) は , 外部バスアクセスが完了するまで待機します。

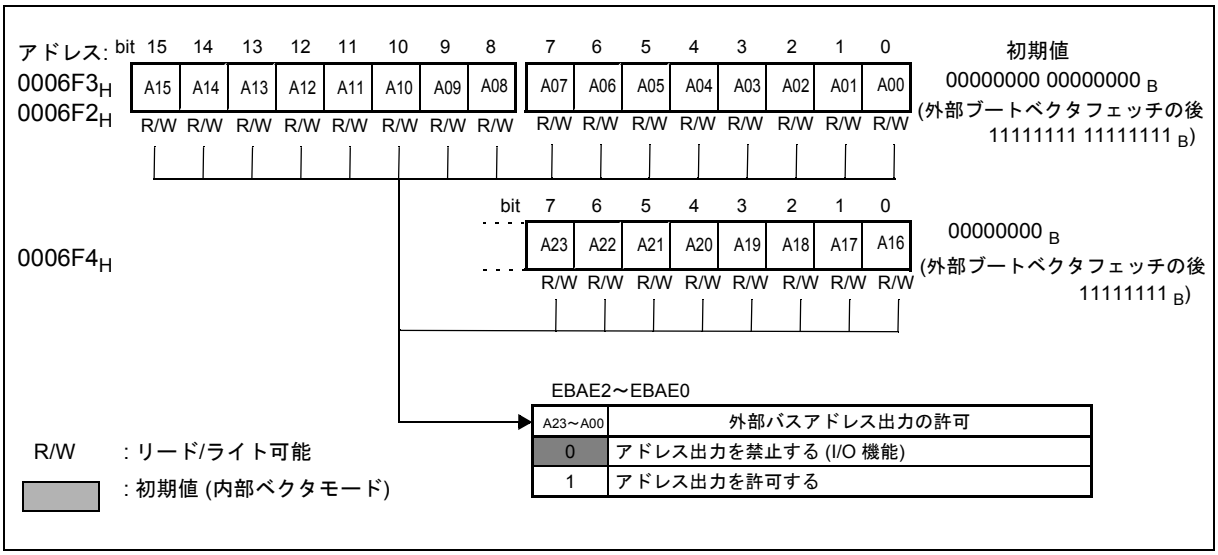
</td></tr>
</table>

12.2.3 外部バスアドレス出力許可レジスタ(EBAE[2:0])

外部バスアドレス出力許可レジスタは, 各アドレスラインの出力機能を制御します。

外部バスアドレス出力許可レジスタ

Figure 12-7. 外部バスアドレス出力許可レジスタ(EBAE[2:0])



ビット名		機能
bit [23:00]	A23 ～ A00: 外部バス アドレス 出力の許可	• これらのビットは , 対応するアドレスラインの出力機能を許可します。 • マルチプレクスバスモードでは , EBAE[1:0] はアドレス / データ共有端子 AD[15:00] のアドレス出力を制御します。ノンマルチプレクスバスモードでは , これらのレジスタは , アドレス端子 A[15:00] のアドレス出力を制御します。 • "0" : 対応するアドレスラインを端子に出力しない。代わりに端子の I/O 機能が許可されるか , またはほかのリソースが使用可能になります。 • "1" : 外部バスアクセス時に , 対応するアドレスラインを端子に出力します。 • 初期値は "0" です。外部ブートベクタフェッチにより "1" に変更されます。 • デバイスを外部ベクタモードで起動した場合, すべてのアドレス出力が許可されます。

デフォルトではアドレスの出力は禁止されます。したがって , 外部バスを使用するためには , それを許可する必要があります。MCU を外部ベクタモードで起動した場合 , 外部ブートベクタを読み出す前に , ブート ROM プログラムがすべてのアドレスラインを許可します。

アドレス出力に使用するすべての端子を , EBAE レジスタの対応するビットによって有効にする必要があります。これはマルチプレクスバスモードでのアドレス / データ共有端子を含みます。

すべてのアドレスおよびアドレス / データ端子のアドレス出力は , 外部バスアクセス時にのみアクティブになります。2 つの外部バスアクセスの間の一時停止時には , アドレスはドライブされません。したがってアクセスの一時停止時のアドレス端子の状態は , 対応するポート機能 DDR および PDR レジスタの設定によって異なります。これらの設定に従って , アドレス端子を Hi-Z に設定するか , "1" または "0" にドライブすることができます。内部プルアップ抵抗をアクティブにすることも可能です。

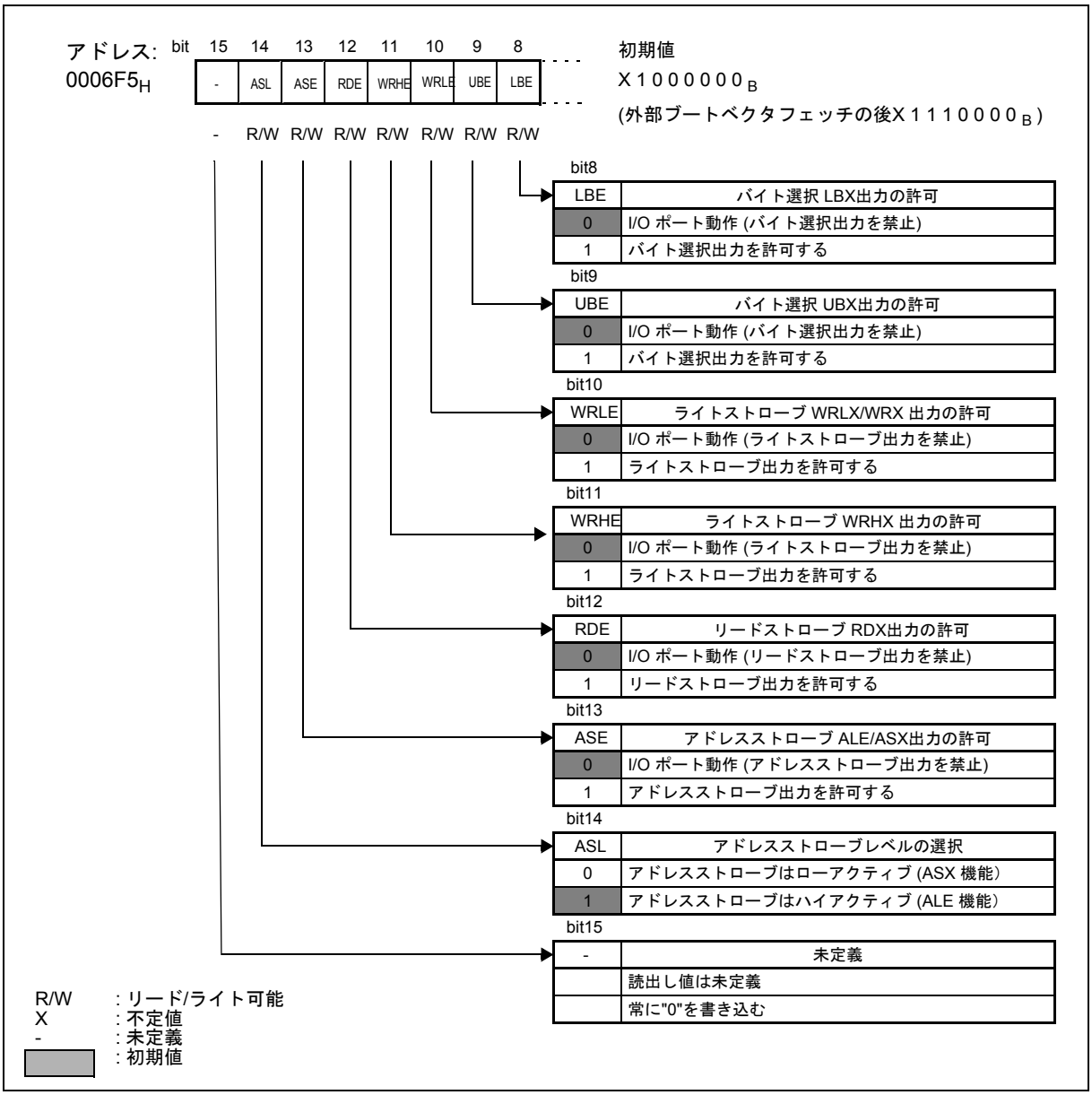
12.2.4 外部バス制御信号レジスタ(EBCS)

このレジスタは , 次のような外部バス制御信号を構成するために使用されます。

- ライトストローブ, リードストローブ, およびバイト選択出力の起動
- アドレスストローブ出力の起動とアクティブレベルの選択

外部バス制御信号レジスタ

Figure 12-8. 外部バス制御信号レジスタ(EBCS)



ビット名		機能
bit 8	LBE バイト選択 LBX 出力の許可	• このビットは , LBX バイト選択信号の出力を許可します。 • "0" : LBX バイト選択信号の出力を禁止 (I/O ポート機能を許可)。 • "1" : LBX バイト選択信号の出力を許可。 • 初期値は , "0" です。したがって LBX により制御される外部デバイスをリセットベクタの読出しのために使用することはできません。 • LBE =1 の場合, データ方向レジスタ (DDR) の対応するビットとは無関係に LBX 端子が 強制的に出力されます。
bit 9	UBE: バイト選択 UBX 出力の許可	• このビットは , UBX バイト選択信号の出力を許可します。 • "0" : UBX バイト選択信号の出力を禁止 (I/O ポート機能を許可)。 • "1" : UBX バイト選択信号の出力を許可。 • 初期値は , "0" です。したがって UBX により制御される外部デバイスをリセットベクタの読出しのために使用することはできません。 • UBE =1の場合, データ方向レジスタ (DDR) の対応するビットとは無関係に UBX 端子は 強制的に出力されます。
bit 10	WRLE: ライトストローブ WRLX/WRX 出力の 許可	• このビットは , WRLX/WRX ライトストローブ信号の出力を許可します。 • "0" : WRLX/WRX ライトストローブ信号の出力を禁止 (I/O ポート機能を許可)。 • "1" : WRLX/WRX ライトストローブ信号の出力を許可。 • 初期値は , "0" です。 • WRLE =1 の場合, データ方向レジスタ (DDR) の対応するビットとは無関係に WRLX/WRX 端子が 強制的に出力されます。
bit 11	WRHE: ライトストローブ WRHX 出力の許可	• このビットは , WRHX ライトストローブ信号の出力を許可します。 • "0" : WRHX ライトストローブ信号の出力を禁止 (I/O ポート機能を許可)。 • "1" : WRHX ライトストローブ信号の出力を許可。 • 初期値は , "0" です。 • WRHE =1 の場合, データ方向レジスタ (DDR) の対応するビットとは無関係に WRHX 端子が強制的に出力されます。
bit 12	RDE: リードストローブ RDX 出力の許可	• このビットは , RDX リードストローブ信号の出力を許可します。 • "0" : RDX リードストローブ信号の出力を禁止 (I/O ポート機能を許可)。 • "1" : RDX リードストローブ信号の出力を許可。 • 初期値は "0" です。外部ブートベクタフェッチにより "1" に変更されます。 • RDE =1の場合, データ方向レジスタ (DDR) の対応するビットとは無関係に RDX 端子が 強制的に出力されます。
bit 13	ASE: アドレスストローブ ALE/ASX出力の 許可	• このビットは , ALE/ASX アドレスストローブ信号の出力を許可します。 • "0" : ALE/ASX アドレスストローブ信号の出力を禁止 (I/O ポート機能を許可)。 • "1" : ALE/ASX アドレスストローブ信号の出力が許可される。 • 初期値は "0" です。外部ブートベクタフェッチモードにより "1" に変更されます。 • ASE =1の場合, データ方向レジスタ (DDR) の対応するビットとは無関係に ALE/ASX 端子が強制的に出力されます。
bit 14	ASL: アドレス ストローブレベル の選択	このビットは , アドレスストローブ信号のアクティブレベルを選択します。 • "0" : アクティブレベルは "L", 非アクティブレベルは "H" (ASX 機能)。 • "1" : アクティブレベルは "H", 非アクティブレベルは "L" (ALE 機能)。 • 初期値は"1"です。
bit 15	未定義	• 読出し値は未定義です。 • 常に "0" を書き込んでください。 • リードモディファイライト系命令は無効です。

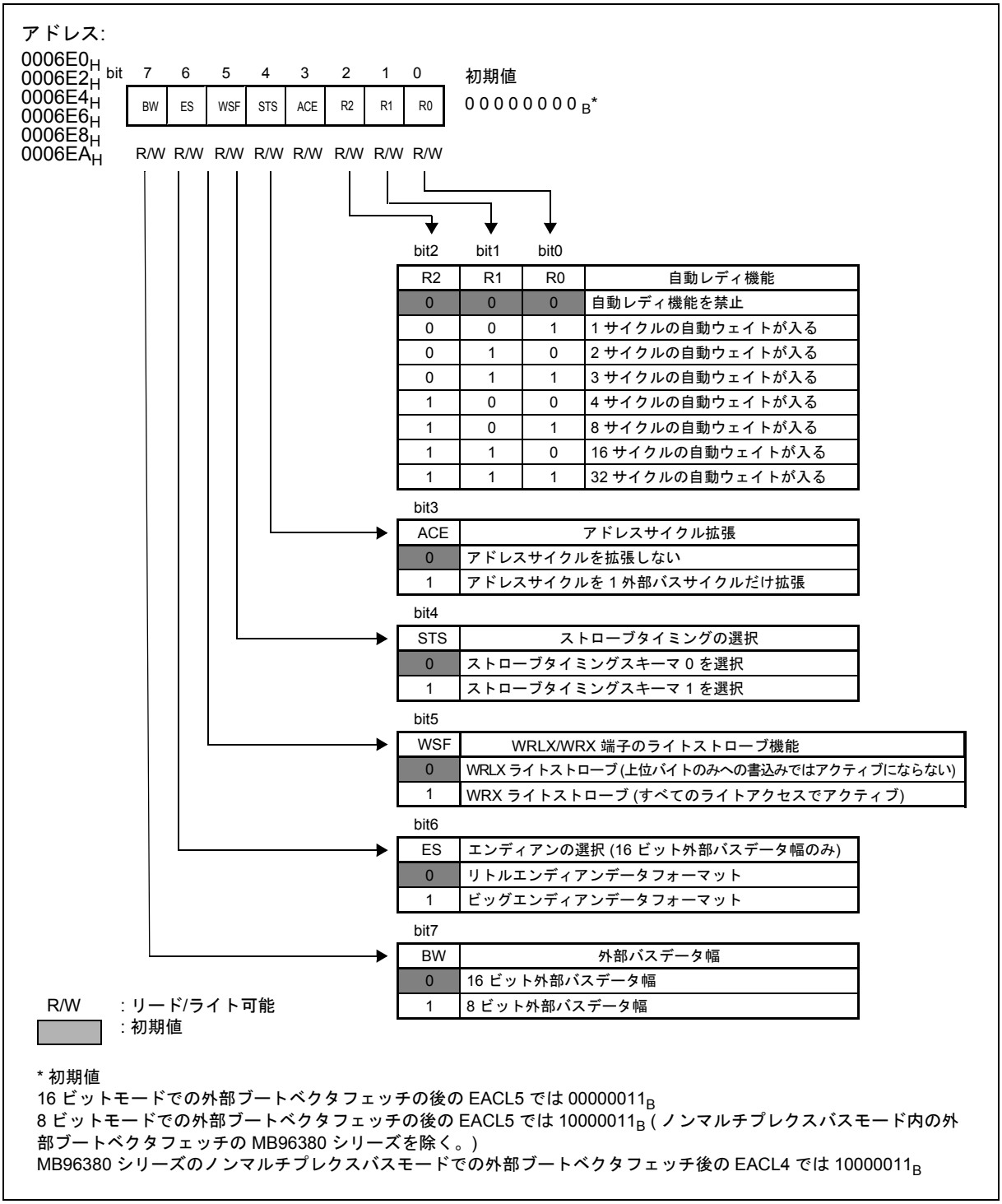
12.2.5 外部領域構成レジスタ(EACH/EACL[5:0])

外部領域構成レジスタは , 各外部領域の以下の設定を構成するために使用します。

- 自動レディ機能
- アドレスサイクル長
- ストローブ信号タイミング
- ライトストローブ機能
- リトル/ビッグエンディアンデータフォーマット
- 外部バスデータ幅
- 外部領域サイズ選択
- チップセレクト出力の許可
- チップセレクトレベルの選択
- アクセスタイプ制限 (外部領域からのコード読出しの禁止が可能)

外部領域構成レジスタ(下位バイト) EACL[5:0]

Figure 12-9. 外部領域構成レジスタ(下位バイト) EACL[5:0]



* 初期値
16 ビットモードでの外部ブートベクタフェッチの後の EACL5 では 00000011_B
8 ビットモードでの外部ブートベクタフェッチの後の EACL5 では 10000011_B (ノンマルチプレクスバスモード内の外部ブートベクタフェッチの MB96380 シリーズを除く。)
MB96380 シリーズのノンマルチプレクスバスモードでの外部ブートベクタフェッチ後の EACL4 では 10000011_B

<table>
<tr><th colspan="2">ビット名</th><th>機能</th></tr>
<tr><td>bit [2:0]</td><td>R2 ～ R0:
自動レディ機能</td><td>

これらのビットは , 自動レディ機能を許可し , 外部領域のウェイトサイクル数を選択します。

bit2 bit1 bit0
<table>
<tr><th>R2</th><th>R1</th><th>R0</th><th>自動レディ機能</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>自動レディ機能を禁止</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1 サイクルの自動ウェイトが入る</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>2 サイクルの自動ウェイトが入る</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>3 サイクルの自動ウェイトが入る</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>4 サイクルの自動ウェイトが入る</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>8 サイクルの自動ウェイトが入る</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>16 サイクルの自動ウェイトが入る</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>32 サイクルの自動ウェイトが入る</td></tr>
</table>

初期値は, "000" です。ただし, 外部ブートベクタフェッチの場合, EACL5 の R[2:0]は, ブート ROM プログラムにより"011" に変更されます。(MB96380シリーズを除き, ノンマルチプレクスバスモードでの外部ブートベクタフェッチでEACL4のR[2:0]は"011"にセットされます。)

</td></tr>
<tr><td>bit 3</td><td>ACE:
アドレスサイクル拡張</td><td>

このビットは , マルチプレクスバスモードでのアドレスサイクルの長さを選択します。
"0" : 標準のアドレスサイクルタイミング。
"1" : アドレスサイクルが 1 外部バスクロックサイクルだけ拡張される。
初期値は , "0" です。
このビットの設定は , ノンマルチプレクスバスモードには影響を与えません。
詳細は, 「12.3 外部メモリアクセス制御信号の動作」のタイミングダイヤグラムを参照してください。

</td></tr>
</table>

ビット名		機能
bit 4	STS: ストローブ タイミングの選択	• このビットは , 異なる外部デバイスに対して入出力タイミングを調整するためにアドレスのタイミングやライトストローブを選択します。 • 初期値は , "0" です。 • 2 つのタイミングに関する詳細は,「12.3 外部メモリアクセス制御信号の動作」のタイミングダイヤグラムを参照してください。
bit 5	WSF: WRLX/WRX 端子のライトストローブ機能	• このビットは , 対応する外部領域の WRLX/WRX 端子のライトストローブ機能を選択します。 • "0" : WRLがアクティブになる:この信号は, 上位バイトへのライトアクセス時にはアクティブになりません。外部デバイスに対して上位バイトと下位バイトに別々のライトストローブ入力が使用される場合はこのビットを "0" に設定してください。 • "1" : WR がアクティブになる : すべてのライトアクセス時に信号がアクティブになります。外部デバイスに対して上位バイトと下位バイトに共通のライトイネーブル入力が使用される場合はこのビットを "1" に設定してください。 • 初期値は, "0"です。
bit 6	ES: エンディアンの 選択	• このビットは , 対応するチップセレクト信号により示されるアドレス領域に対して 16 モードのデータフォーマットを選択します。 • "0" : リトルエンディアン • "1" : ビッグエンディアン • 初期値は , "0" です。 • 詳細は Table 12-4を参照してください。
bit 7	BW: 外部バスデータ幅	• このビットは , 対応するアドレス領域の外部バスデータ幅を選択します。 • "0" : 16 ビットデータ幅を選択。 • "1" : 8 ビットデータ幅を選択 • 初期値は, "0"です。ただし, 8 ビット幅の外部リセットベクタフェッチの場合, EACL5 の BW はブート ROM プログラムにより"1" に変更されます。(MB96380シリーズを除き, ノンマルチプレクスバスモードでの外部ブートベクタフェッチでEACL4のBWは"1"にセットされます。)

下の表は, 異なる外部バスフォーマットのエンディアン選択の影響を示しています。

Table 12-4. エンディアンの選択

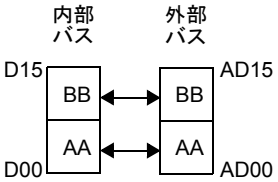
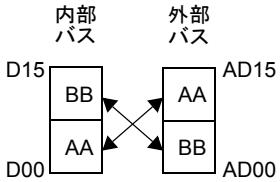
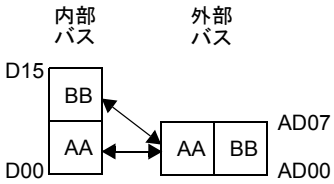
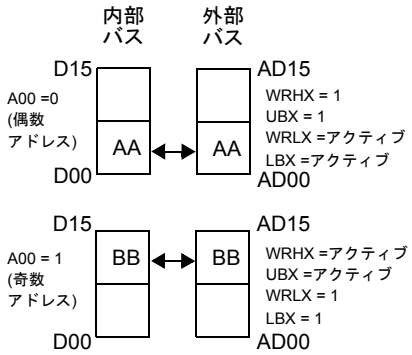
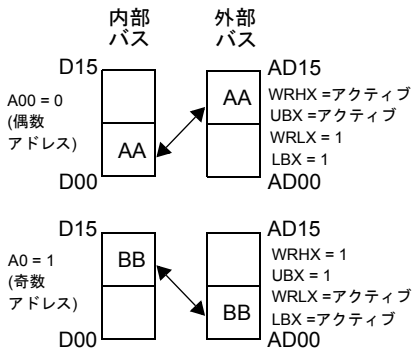
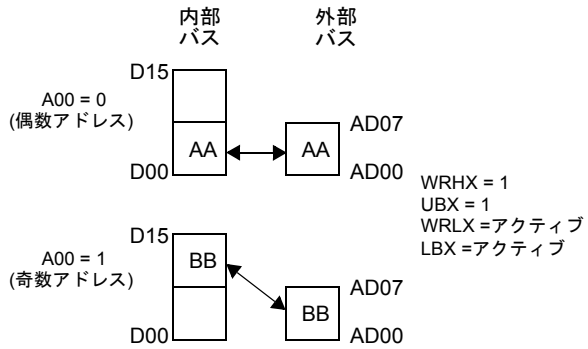
<table>
<tr><th rowspan="2">内部バスでのアクセス幅</th><th rowspan="2">外部バスの選択したバス幅</th><th colspan="2">外部バス上のシーケンス</th></tr>
<tr><th>リトルエンディアン (ES = '0')</th><th>ビッグエンディアン (ES = '1')</th></tr>
<tr><td rowspan="2">16ビット</td><td rowspan="2">16ビット</td><td>
</td><td>
</td></tr>
<tr><td colspan="2">• 書込み時に , WRHX と WRLX が共にアクティブになります。
• すべての読出し / 書込み時に UBX と LBX の両方が起動されます。
• アドレスは常に偶数です (A00 =0)。</td></tr>
<tr><td rowspan="2">16ビット</td><td rowspan="2">8ビット</td><td colspan="2">
</td></tr>
<tr><td colspan="2">• 内部 16 ビットアクセスは , 次の 2 つの 8 ビットアクセスに分割されます。
1. AA: 外部アドレスは内部アドレスと同じです (A00 =0)。
2. BB: 外部アドレスは , 1 増加します。(A00=1)。
• WRLX は書込み時にアクティブとなり , WRHX は "1" に固定されます。
• LBXは書込みおよび読出し時にアクティブとなり , UBXは"1"に固定されます。</td></tr>
</table>

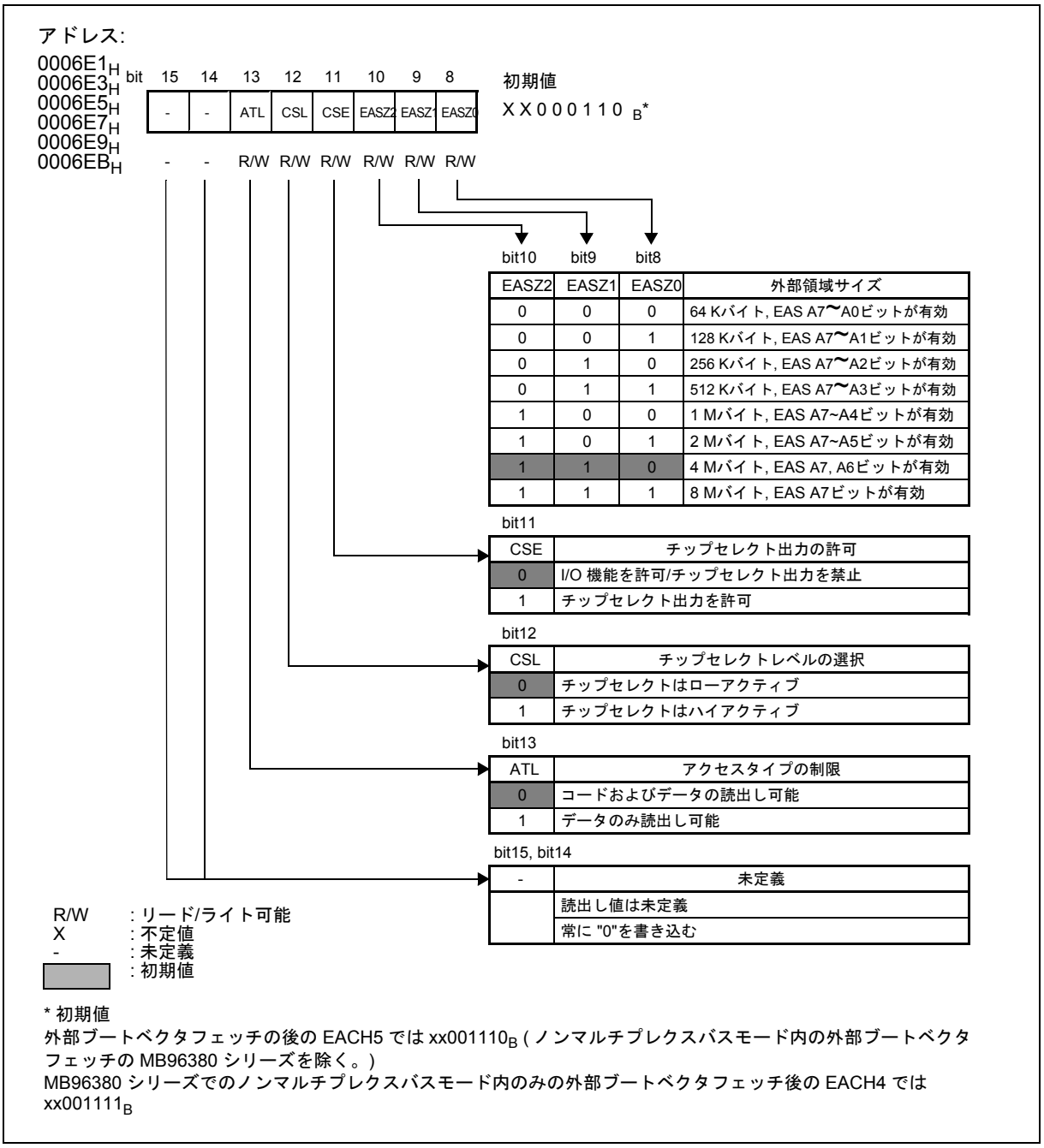
Table 12-4. エンディアンの選択

<table>
<tr><th rowspan="2">内部バスでのアクセス幅</th><th rowspan="2">外部バスの選択したバス幅</th><th colspan="2">外部バス上のシーケンス</th></tr>
<tr><th>リトルエンディアン (ES = '0')</th><th>ビッグエンディアン (ES = '1')</th></tr>
<tr><td rowspan="2">8ビット</td><td rowspan="2">16ビット</td><td>
</td><td>
</td></tr>
<tr><td colspan="2">• WRHXおよびUBXは, 外部バスデータ端子 AD[15:08]でデータがアクセスされるとき使用されます。
• WRLXおよびLBXは, 外部バスデータ端子 AD[07:00]でデータがアクセスされるとき使用されます。</td></tr>
<tr><td rowspan="2">8ビット</td><td rowspan="2">8ビット</td><td colspan="2">
</td></tr>
<tr><td colspan="2">• WRLX は書込み時にアクティブとなり , WRHX は "1" に固定されます。
• LBXは書込みおよび読出し時にアクティブとなり, UBXは"1"に固定されます。</td></tr>
</table>

- エンディアン選択は, 外部バスのデータ端子 AD[15:00], ライトストローブ信号 WRHX/WRLX およびバイト選択信号 UBX/LBX に影響を与えます。
- エンディアン選択は, ほかの信号 (アドレスなど) には影響を与えません。マルチプレクスバスモードでは, アドレス/データ端子のデータサイクルにのみ影響します。
- WSF=lでは,WRLX/WRX端子は,その機能をWRXに変更します(WRLXまたはWRHXがアサートされると"0"にアサートされます)。

外部領域構成レジスタ(上位バイト) EACH[5:0]

Figure 12-10. 外部領域構成レジスタ(上位バイト) EACH[5:0]



＜注意事項＞

8 〜 10 ビットの外部領域サイズは EACH0 および EACH1 レジスタに定義されていません。

<table>
<tr><th colspan="2">ビット名</th><th>機能</th></tr>
<tr><td>bit[10:8]</td><td>EASZ2 ～ EASZ0:
外部領域サイズ</td><td>

これらのビットは , 外部領域のサイズを定義するために使用します。この設定によって , 対応する外部領域選択レジスタ (EAS) のどのビットをアドレス比較に使用するかを定義します。

bit10 bit9 bit8
<table>
<tr><th>EASZ2</th><th>EASZ1</th><th>EASZ0</th><th>外部領域サイズ</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>64 K バイト , EAS A7 ～ A0 ビットが有効</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>128 K バイト , EAS A7 ～ A1 ビットが有効</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>256 K バイト , EAS A7 ～ A2 ビットが有効</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>512 K バイト , EAS A7 ～ A3 ビットが有効</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1 M バイト , EAS A7 ～ A4 ビットが有効</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>2 M バイト , EAS A7 ～ A5 ビットが有効</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>4 M バイト , EAS A7, A6 ビットが有効</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>8 M バイト , EAS A7 ビットが有効</td></tr>
</table>

初期値は , "110" です。つまり , 対応する EAS レジスタの上位 2 ビットだけが評価され , 外部領域のサイズは 4 M バイト です。
これらのビットは , 外部領域構成レジスタ EACH[5:2] にのみ存在します。外部領域 0 および 1 のアドレスサイズは , 固定されています。
MB96380シリーズでのノンマルチプレクスバスモードの外部ブートベクタフェッチではEASZ[2:0]は"111" (8Mバイト)にセットされます。

</td></tr>
<tr><td>bit 11</td><td>CSE:
チップセレクト出力の許可</td><td>

このビットは , 対応するチップセレクト信号の出力を許可します。
"0" : チップセレクト出力を禁止 (I/O ポート機能を許可)。
"1" : チップセレクト出力を許可。
初期値は, "0"です。ただし, 外部ブートベクタフェッチの場合, EACH5 の CSE はブート ROM プログラムにより"1"に変更されます。(MB96380シリーズでのノンマルチプレクスバスモードのスタートアップを除いては, EACH4のCSEは"1"にセットされます。)

</td></tr>
<tr><td>bit 12</td><td>CSL:
チップセレクトレベルの選択</td><td>

このビットは , 対応するチップセレクト信号のアクティブレベルを選択します。
"0" : アクティブレベルは , "L", 非アクティブレベルは "H"。
"1" : アクティブレベルは , "H", 非アクティブレベルは "L"。
対応するチップセレクト信号 CSx がアクティブレベルで , 対応するアドレス領域が許可されている場合 , この領域へのアクセスが行われます。アドレスが内部 ROM/ フラッシュ領域内であり , 外部バスアクセスを開始する場合 , EBM レジスタで外部 ROM 許可ビットもセットする必要があります。
初期値は, "0"です。

</td></tr>
<tr><td>bit 13</td><td>ATL:
アクセスタイプの制限</td><td>

このビットは外部領域からのコード読出しが可能かどうかを選択します。
"0" : コード読出しを許可。
"1" : コード読出しは不可能。リードアクセスでは , 未定義の命令例外が返されます。
外部領域からのデータ読出しは常に可能です。
このビットを "1" にセットした後 , このビットを "0" に戻すことはできません。リセットによってのみクリアできます。
初期値は, "0" です。

</td></tr>
<tr><td>bit 14, bit 15</td><td>未定義</td><td>

読出し値は未定義です。
常に "0" を書き込んでください。
リードモディファイライト系命令は無効です。

</td></tr>
</table>

12.2.6 外部領域選択レジスタ(EAS[5:2])

外部バスインタフェースは , 次の 2 種類の外部アドレス領域をサポートします。

- 固定アドレス範囲をもつ領域 (外部領域 0 および 1)
- 選択可能なアドレス範囲をもつ領域 (外部領域 2 ～ 5)

概要

外部バスインタフェースは , 次の 2 種類のアドレス領域をサポートします。

- 固定アドレス範囲 (外部領域 0 および 1):
  - 外部領域 0 はアドレス範囲$00:00F0_H$～$00:00FF_H$に固定されています。
  - 外部領域 1 はアドレス範囲 $00：0C00_H$～RAMの開始アドレス に固定されています。
    RAMの開始アドレスはデバイス構成によって異なり, 同じシリーズでも型格によって異なります。
    ソフトウェアによって変更することはできません。
    EVA デバイス上のRAMの開始アドレスは構成可能であり, エミュレートするデバイスによって異なります。
- プログラム可能なアドレス範囲 (外部領域 2 ～ 5):
  - 各領域の外部アドレス選択レジスタ EAS は, 対応する外部領域の上位 8 アドレスビットを定義しますから, 64KB の範囲内のアドレスの指定が可能です。
  - 対応する外部領域構成レジスタ (EACH[5:2]) の EASZ[2:0]ビットは, アドレス比較に使用する EAS[5:2]レジスタのビットの数を定義します。したがって 64 KB ～ 8 MB の範囲の外部領域を選択できます。
  - ただし $10:0000_H$ より下のアドレス領域を選択することはできません。

外部領域選択レジスタ(EAS[5:2])

Figure 12-11. 外部領域選択レジスタ(EAS[5:2])

アドレス: EAS2:0006EC$_H$
EAS3:0006ED$_H$
EAS4:0006EE$_H$
EAS5:0006EF$_H$

bit	15/7	14/6	13/5	12/4	11/3	10/2	9/1	8/0	
	A7	A6	A5	A4	A3	A2	A1	A0	初期値
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	********

R/W : リード/ライト可能

これらのレジスタは , 外部領域 2 ～ 5 のアドレスデコード領域を設定するために使用します。これらのレジスタはアドレス値の上位 8 ビットを指定しますから , 64 KB の範囲内のアドレスの指定が可能です。

誤動作を防止するために , EAS レジスタを変更するときは外部バスモードレジスタ (EBM) の対応する外部領域を禁止します。

EAS[5:2]の初期値

EAS[5:2] レジスタには種々の初期値があります。初期値によって , $10:0000_H$ より上のメモリ領域が 4 つのセクションに分割されます。外部リセットベクタフェッチ時には , 外部領域 5 および対応するチップセレクト CS5 が使用されます。(MB96380 シリーズは例外で , ノンマルチプレクスバスモードの外部ブートベクタフェッチで外部領域 4 および CS4 が使用されます。) EAS[5:2] の値はいつでも , ソフトウェア的に変更できます。

Table 12-5. EAS[5:2]の初期値

外部領域	EAS レジスタの初期値	デフォルトメモリ領域	チップセレクト信号
外部領域 2	EAS2= 00_H (EACH2: EASZ = "110")	h'10.0000...h'3f.ffff (3MByte)	CS2
外部領域 3	EAS3= 40_H (EACH3: EASZ = "110")	h'40.0000...h'7f.ffff (4MByte)	CS3
外部領域 4	MB96(F)38x ノンマルチプレクスバスモードでの外部ブートベクタフェッチ : EAS4 = 80_H (EACH4: EASZ[2:0] =111)	$80:0000_H$～$FF:FFFF_H$ (8MB,外部リセットベクタを含む)	CS4
	他の全てのデバイスおよびモード : EAS4 = 80_H (EACH4: EASZ[2:0] =110)	h'80.0000...h'bf.ffff (4MByte)	
外部領域 5	EAS5= $C0_H$ (EACH5: EASZ = "110")	$C0:0000_H$～$FF:FFFF_H$ (=4MB, 外部リセットベクタを含む)	CS5

外部アドレス領域選択の動作

CPU がプログラムまたはデータにアクセスするとき, アドレスの上位 8 ビットといずれかの間の一致が検出された場合, 外部バスアクセスが実行されます (EBM レジスタで対応する外部領域が起動されている必要があります)。EACH[5:0]レジスタの対応するEASZ[2:0]ビットの設定によっては, EAS レジスタのいくつかの LSB ビットが無視され, 64 KB ～ 8 MB の領域のデコードが可能になります。

例:

EAS2= $F0_H$
EACH2:EASZ =000_B
- デコードされないバンク : 00 ～ EF, F1 ～ FF
- デコードされるバンク : F0

EAS2= $F0_H$
EACH2:EASZ =001_B
- デコードされないバンク : 00 ～ EF, F2 ～ FF
- デコードされるバンク : F0, F1

EAS2= 17_H
EACH2:EASZ =000_B
- デコードされないバンク : 00 ～ 16, 18 ～ FF
- デコードされるバンク : 17

EAS2= 00_H
EACH2:EASZ =110_B
- デコードされないバンク : 40 ～ FF
- デコードされるバンク : 00 ～ 3F
  00 ～ 0F: 外部領域 2 ～ 5 が選択できないためアクセスできない。
  10～3F:アクセス可能

外部バス領域内のアドレスへアクセスされた場合でも , そのアドレスが選択されたメモリ領域内にない場合 , そのアクセスは無効です (リードアクセスでは "0" が返され , ライトアクセスは無視されます)。
外部領域 2 ～ 5 のアドレス範囲がオーバラップしてはいけません。アドレスが複数の外部領域に含まれる場合 , 番号が最も小さい外部領域の構成が優先されます。ほかの EACL[5:0] レジスタの設定は使用されません。
現在のアクセスと一致するすべての領域のチップセレクト信号が出力されます (許可されている場合)。

12.3 外部メモリアクセス制御信号の動作

外部バス動作時の制御信号について説明します。

12.3.1 マルチプレクス外部バスモード

- すべてのタイミングチャートでは, 下記の設定をしています。
  - CSx アクティブレベルは"L": EACHx:CSL=0
  - すべてのアドレス出力を許可 : EBAE:A[23:00]=11...11
  - 外部クロックの非アクティブレベルは"0": EBCF:CKI = 0
- 以下のタイミングダイヤグラムは, 下記の設定の影響を示します。
  - Figure 12-12 は, 16 ビット外部バスモードでの標準マルチプレクスアクセスを示しています。
  - Figure 12-13 および Figure 12-15 は, ストローブ時間選択ビット (EACL:STS) の影響を示しています。
  - Figure 12-14 および Figure 12-15 は, アドレスサイクル拡張 (EACL:ACE)の影響を示しています。
  - Figure 12-16 は, 8 ビット外部バスモードでの標準マルチプレクスアクセスを示しています。
  - Figure 12-17 は, 自動レディ機能 (1 ウェイトサイクルを挿入) の影響を示しています。
- マルチプレクスバスモードのタイミングチャート

Figure 12-12. マルチプレクス16ビットバス: EACL:STS='0'およびEACL:ACE='0'の場合のワードおよびバイトアクセス

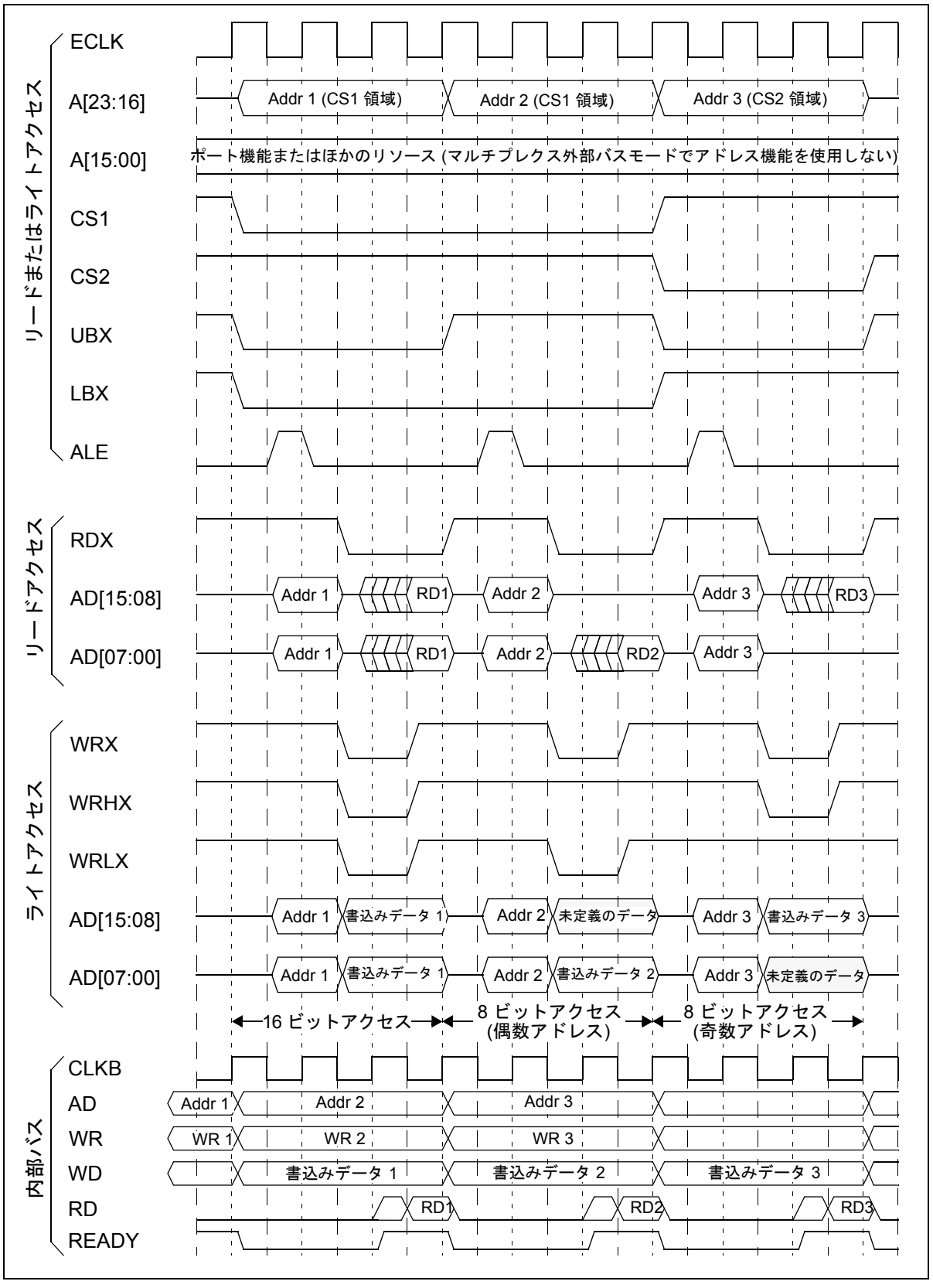


Figure 12-13. マルチプレクス16ビットバス: EACL:STS='1'およびEACL:ACE='0'の場合のワードおよびバイトアクセス

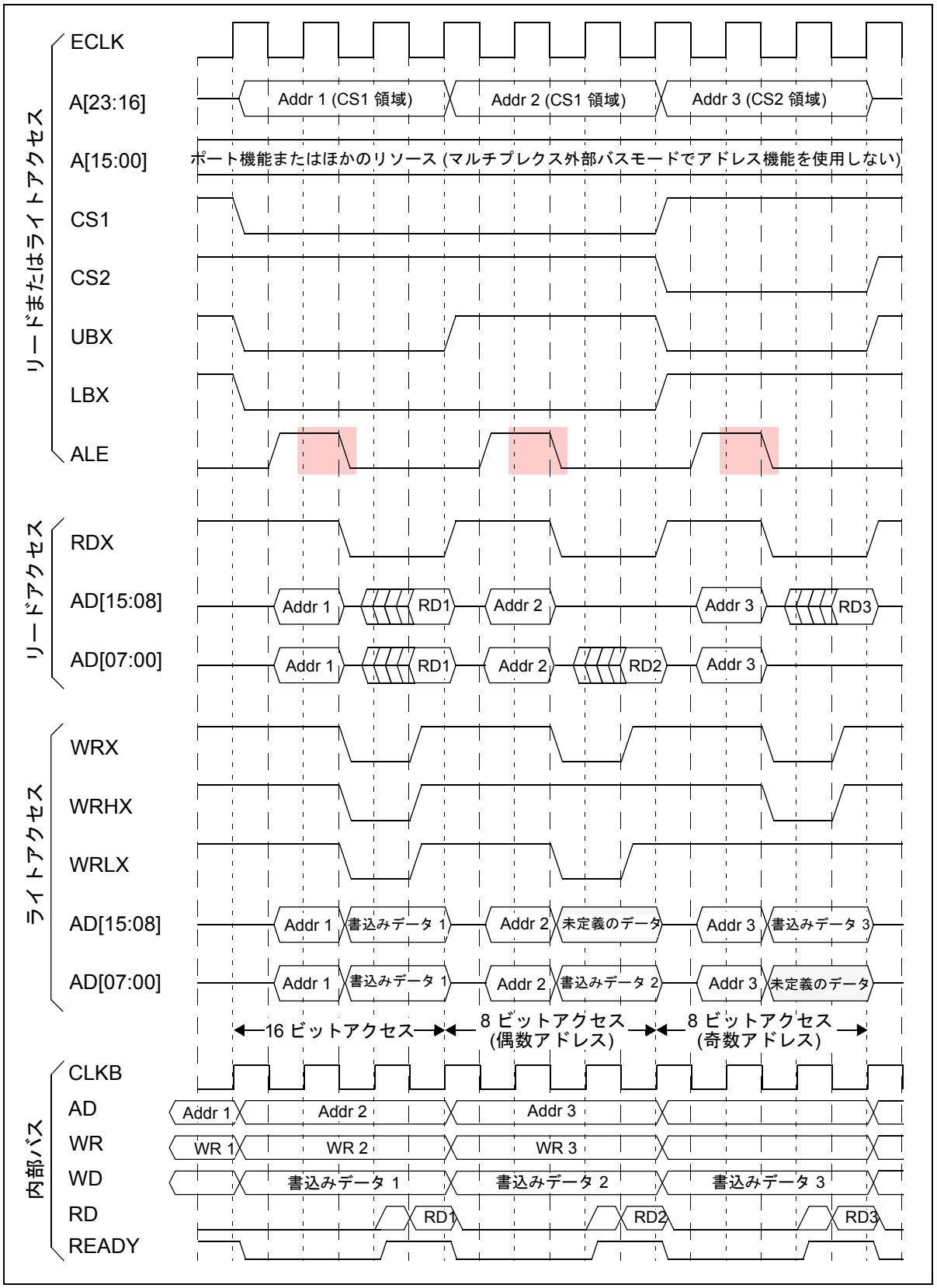


Figure 12-14. マルチプレクス16ビットバス: EACL:STS='0'およびEACL:ACE='1'の場合のワードおよびバイトアクセス

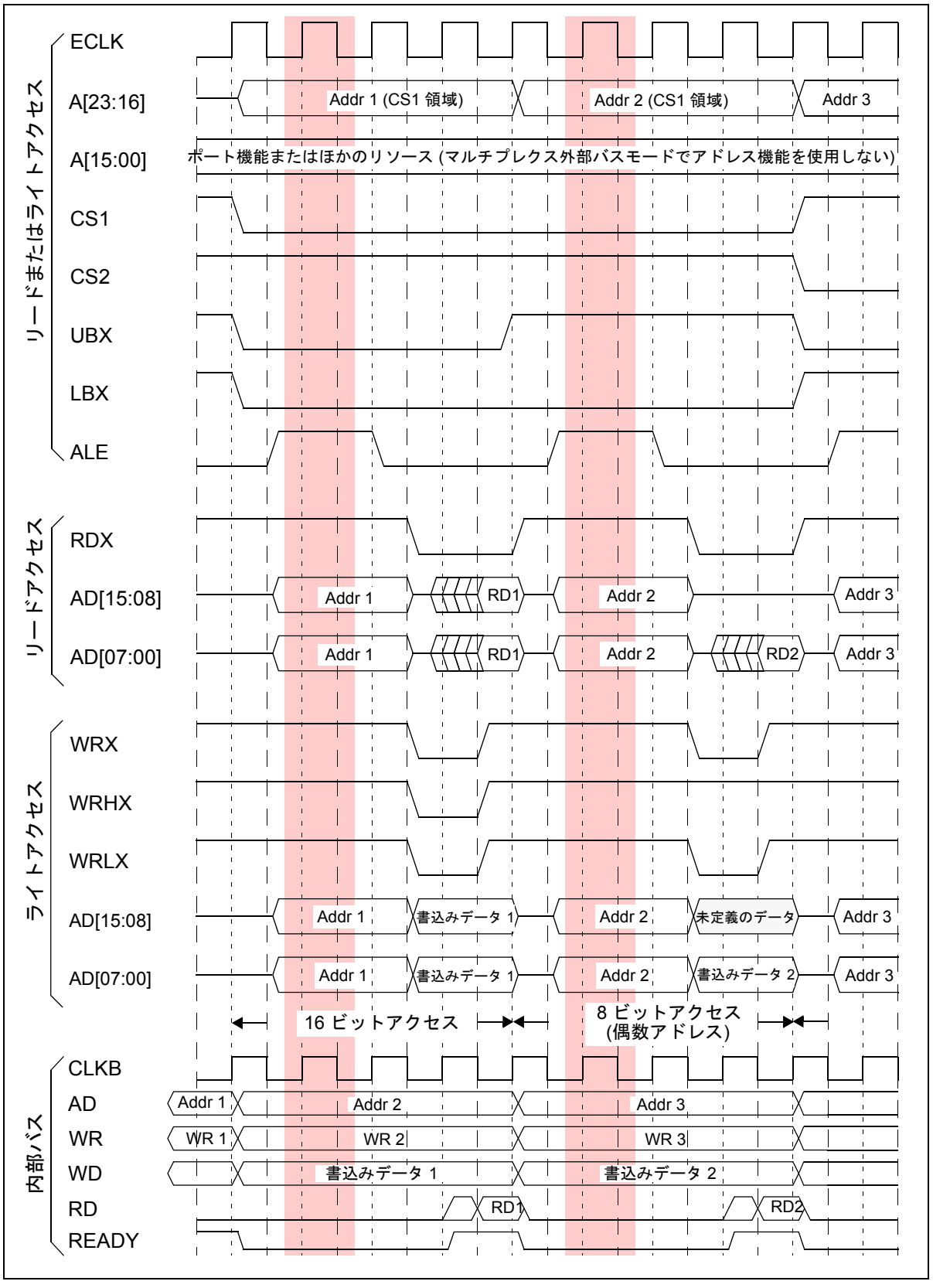


Figure 12-15. マルチプレクス16ビットバス: EACL:STS='1'およびEACL:ACE='1'の場合のワードおよびバイトアクセス

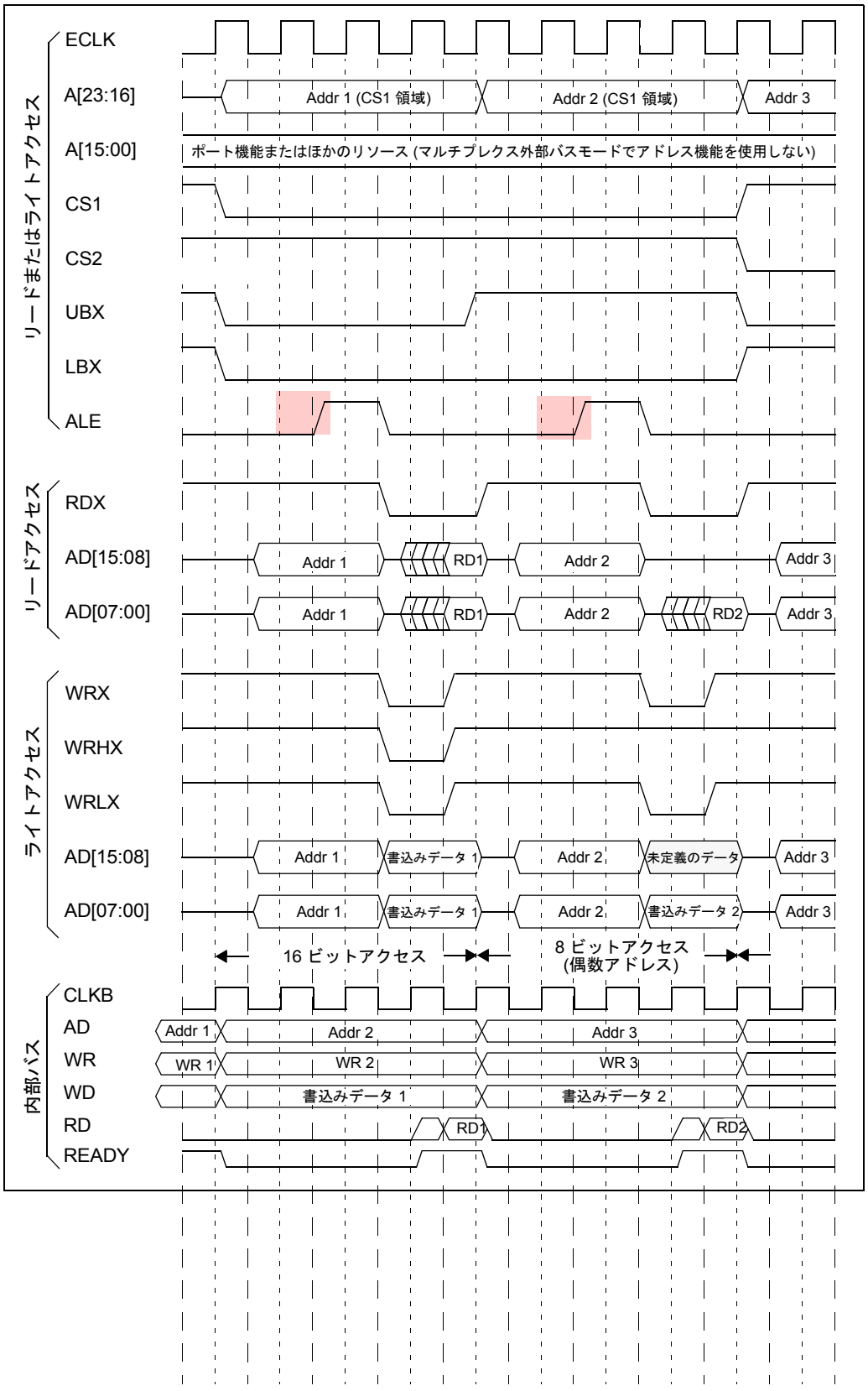


Figure 12-16. マルチプレクス8ビットバス: EACL:STS='0'およびEACL:ACE='0'の場合のワードおよびバイトアクセス

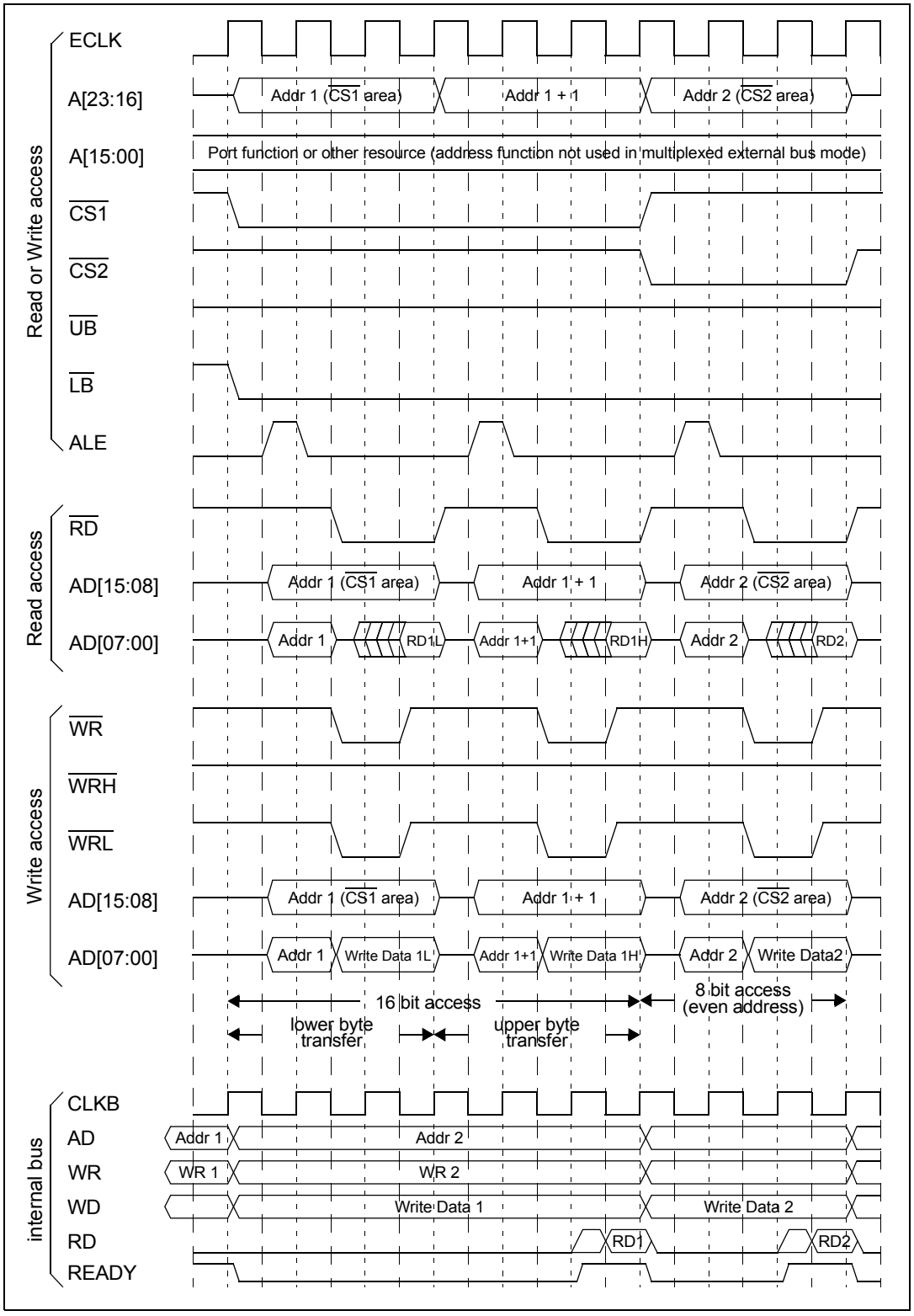
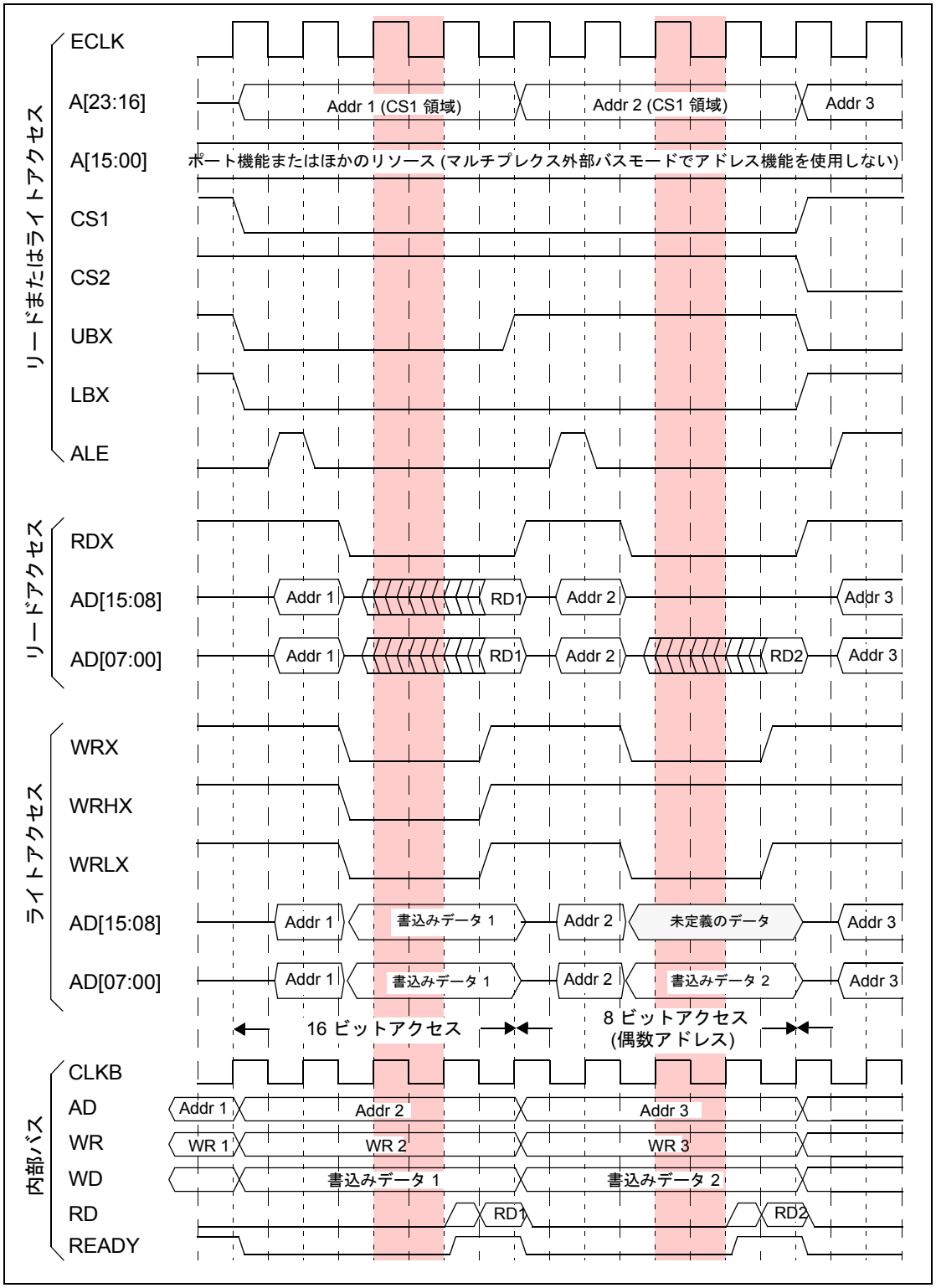


Figure 12-17. マルチプレクス16ビットバス: 自動レディ機能による1ウェイトサイクルの追加(EACL:R[2:0]='001', EACL:STS='0', EACL:ACE='0)'



12.3.2 ノンマルチプレクス外部バスモード

- すべてのタイミングチャートでは, 下記の設定をしています。
  - CSx アクティブレベルは"L": EACHx:CSL=0
  - すべてのアドレス出力を許可: EBAE:A[23:00]=11...11
  - 外部クロックの非アクティブレベルは"0": EBCF:CKI = 0
- 以下のタイミングダイヤグラムは, 下記の設定の影響を示します。
  - Figure 12-18 は, 16 ビット外部バスモードでの標準ノンマルチプレクスアクセスを示しています。
  - Figure 12-19 は, ストローブ時間選択ビット (EACL:STS) の影響を示しています。
  - Figure 12-20 は, 8 ビット外部バスモードでの標準ノンマルチプレクスアクセスを示しています。
  - Figure 12-21 は, 自動レディ機能 (1 ウェイトサイクルを挿入) の影響を示しています。

■ ノンマルチプレクスバスモードのタイミングチャート

Figure 12-18. ノンマルチプレクス16ビットバス: EACL:STS='0'の場合のワードおよびバイトアクセス

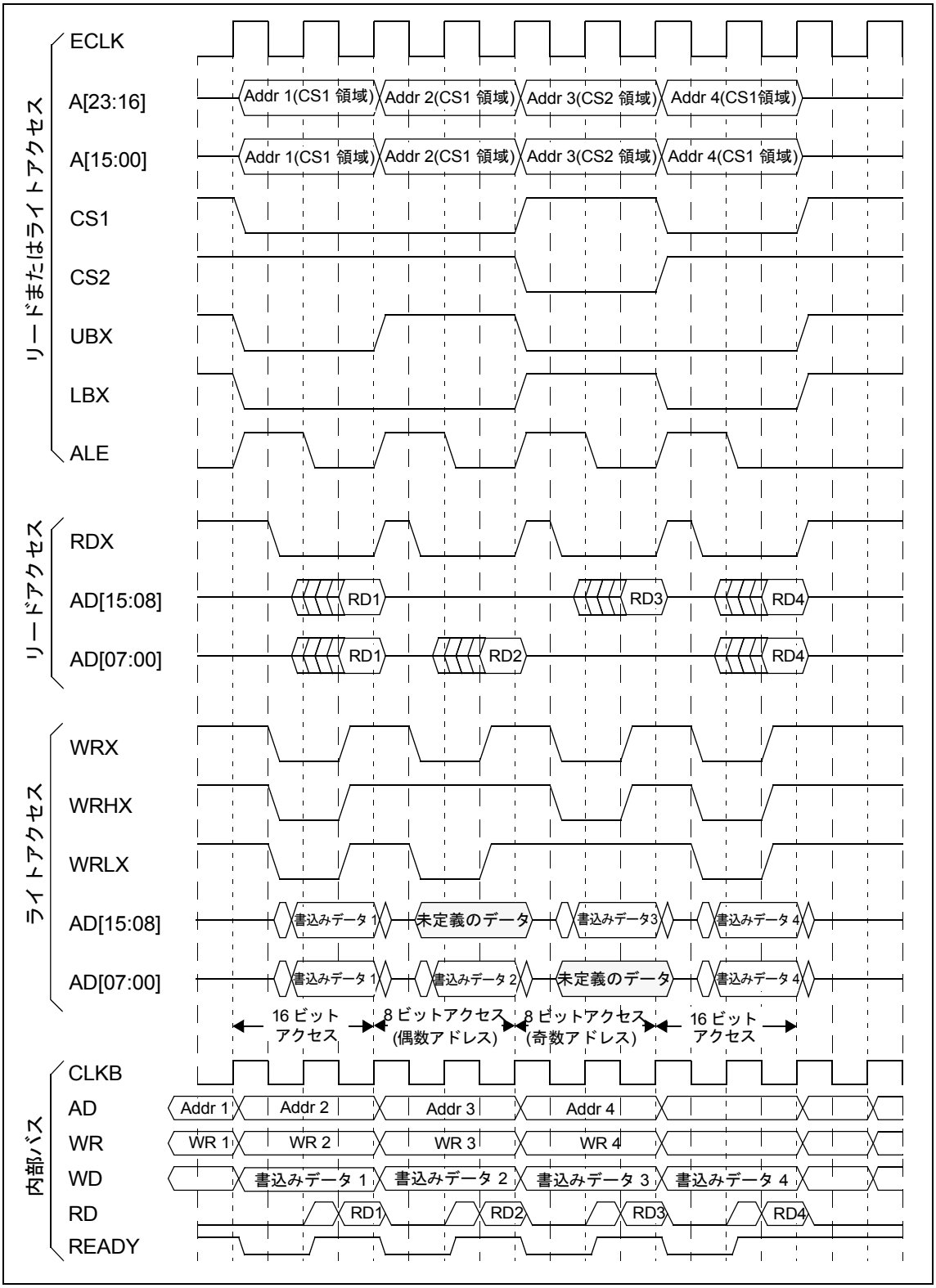


Figure 12-19. ノンマルチプレクス16ビットバス: EACL:STS='1'の場合のワードおよびバイトアクセス

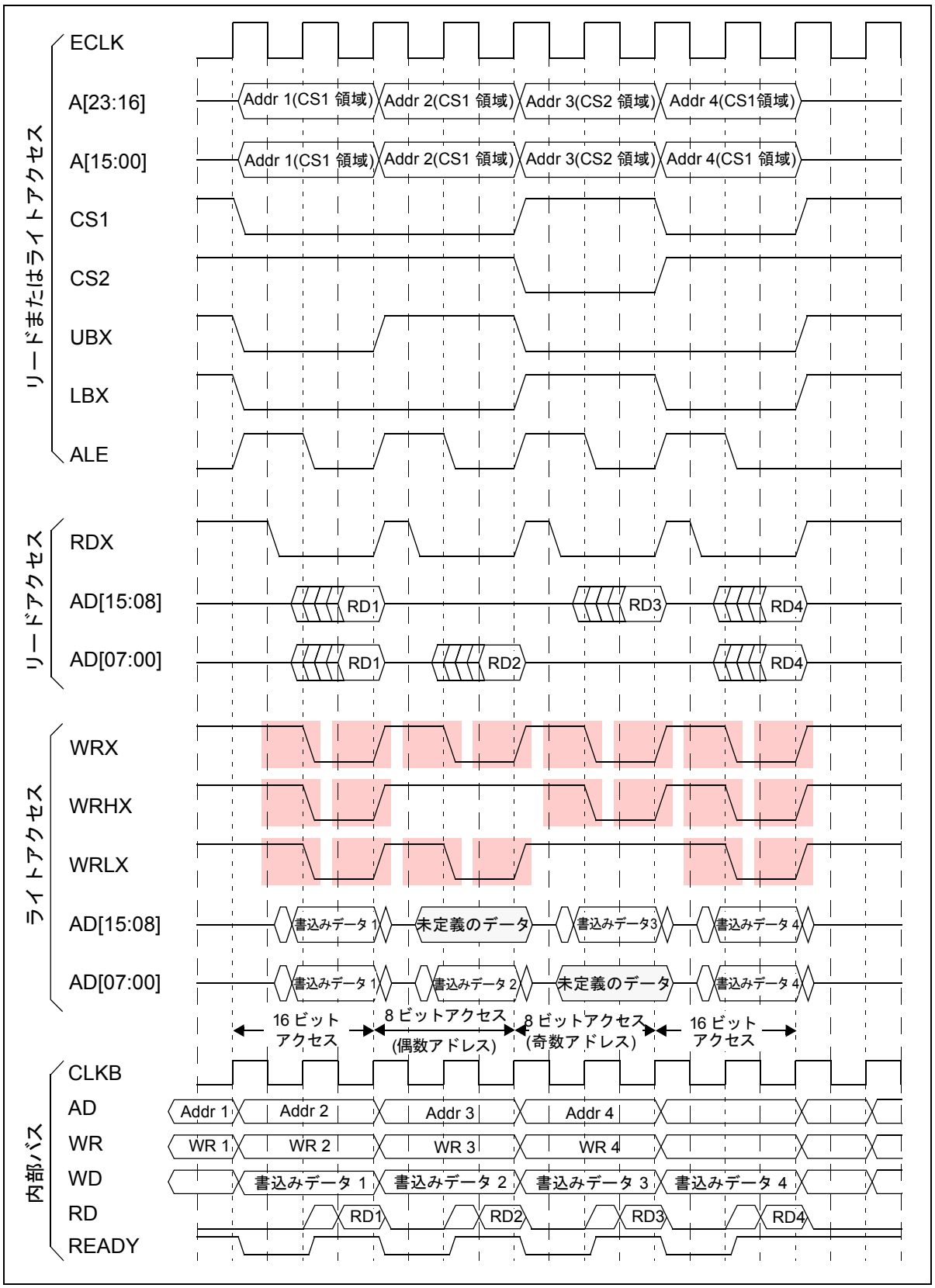


Figure 12-20. ノンマルチプレクス8ビットバス: EACL:STS='0'およびEACL:ACE='0'の場合のワードおよびバイトアクセス

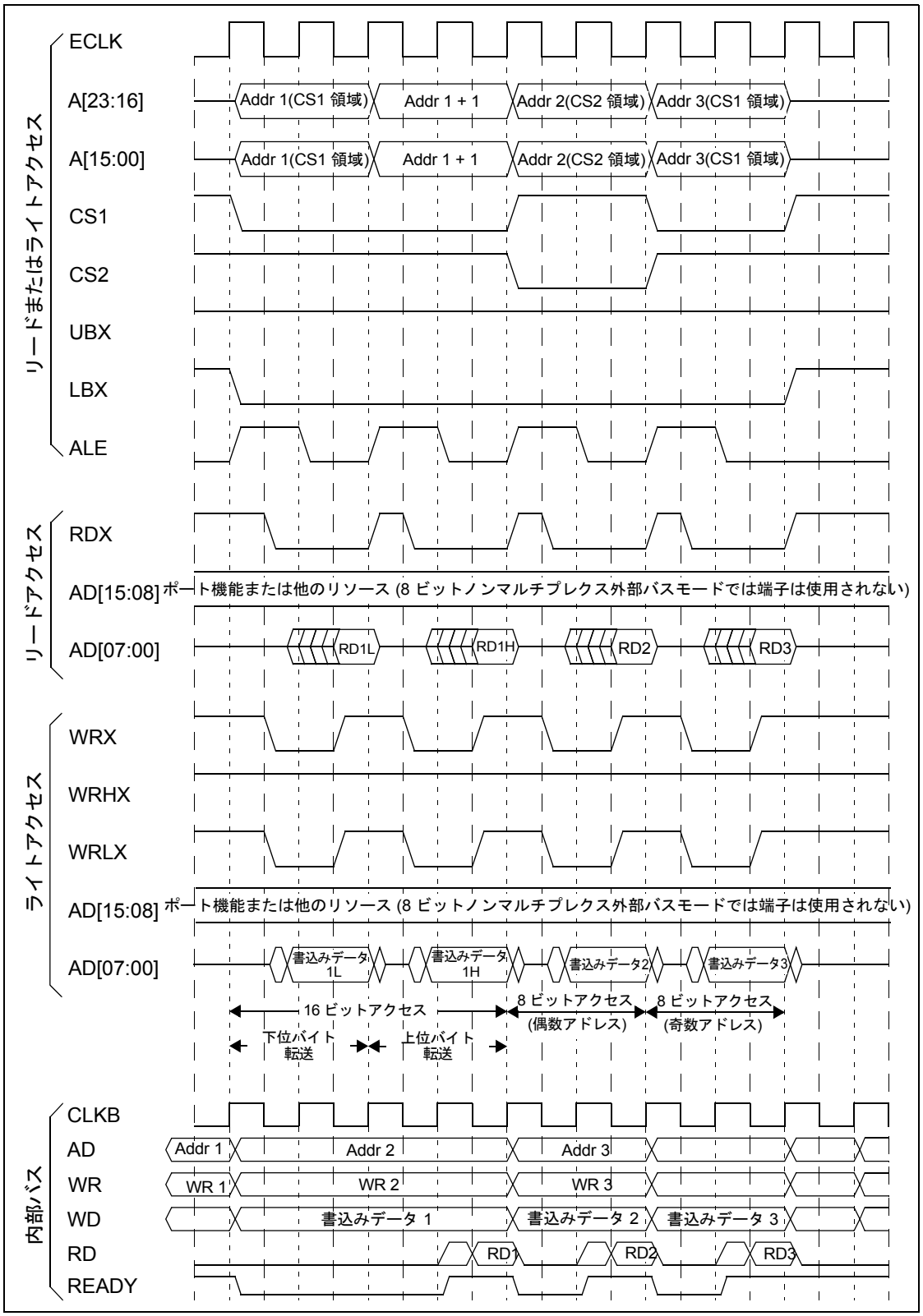
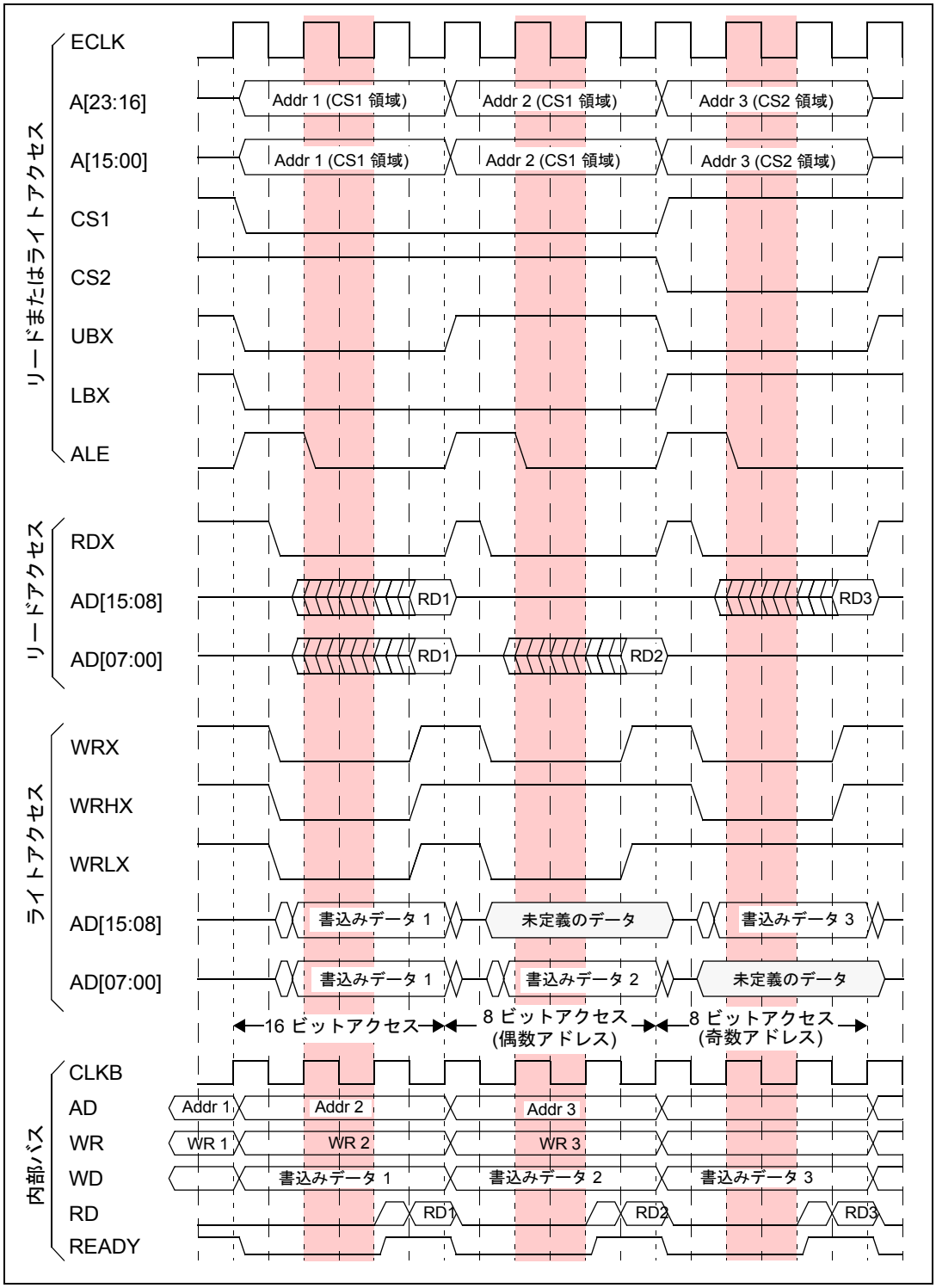


Figure 12-21. ノンマルチプレクス16ビットバス: 自動レディ機能による1ウェイトサイクルの追加(EACL:R[2:0]='001', EACL:STS='0', EACL:ACE='0)'



12.3.3 レディ機能

外部レディ機能 (RDY 端子) および (または) 自動ウェイト機能を有効にすると, 低速メモリおよび周辺回路へのアクセスが可能になります。

12.3.3.1 レディ機能

外部デバイスへのリードおよびライトアクセスはレディ機能により拡張できます。F^{2}MC-16FX MCU では , 外部アクセスのデータサイクルを拡張するために , 自動レディ機能および外部レディ機能の 2 つの方法を利用できます。

レディ機能によって外部バスを停止すると , 外部バス動作が完了するまで内部システムバスも停止します。

■ **自動レディ機能**

自動レディ機能は , 外部回路なしにデータサイクルを拡張するために , 1 ～ 32 のウェイトサイクルを挿入します。挿入するウェイトサイクルの数は , 各外部領域について個別に , EACL:R[2:0] ビットによって選択できます。自動レディ機能の影響を示すタイミングダイヤグラムを Figure 12-17 および Figure 12-21 に示しています。

■ **外部レディ機能**

外部レディ機能は , EBCF レジスタの RYE ビットを "1" に設定し , 対応する入力許可レジスタ (PIER00 ～ PIER17) によって RDY 端子入力機能を有効にすることで使用できます。

RDY 端子は , 外部バスアクセスの最後のサイクルの始めでサンプリングされます。RDY=0 の場合 , RDY が "1" に戻るまでデータサイクルが拡張されます。

外部レディ機能と自動レディ機能を合わせて使用することができます。この場合 , 最後の外部バスサイクルの始めで , 自動ウェイトサイクルが終了した後の RDY 端子の値がサンプリングされます。

Figure 12-22 および Figure 12-23 は , 外部レディ機能のタイミングダイヤグラムを示しています。

Figure 12-22. マルチプレクス16ビットバス: 外部レディ機能によって拡張されたデータサイクル

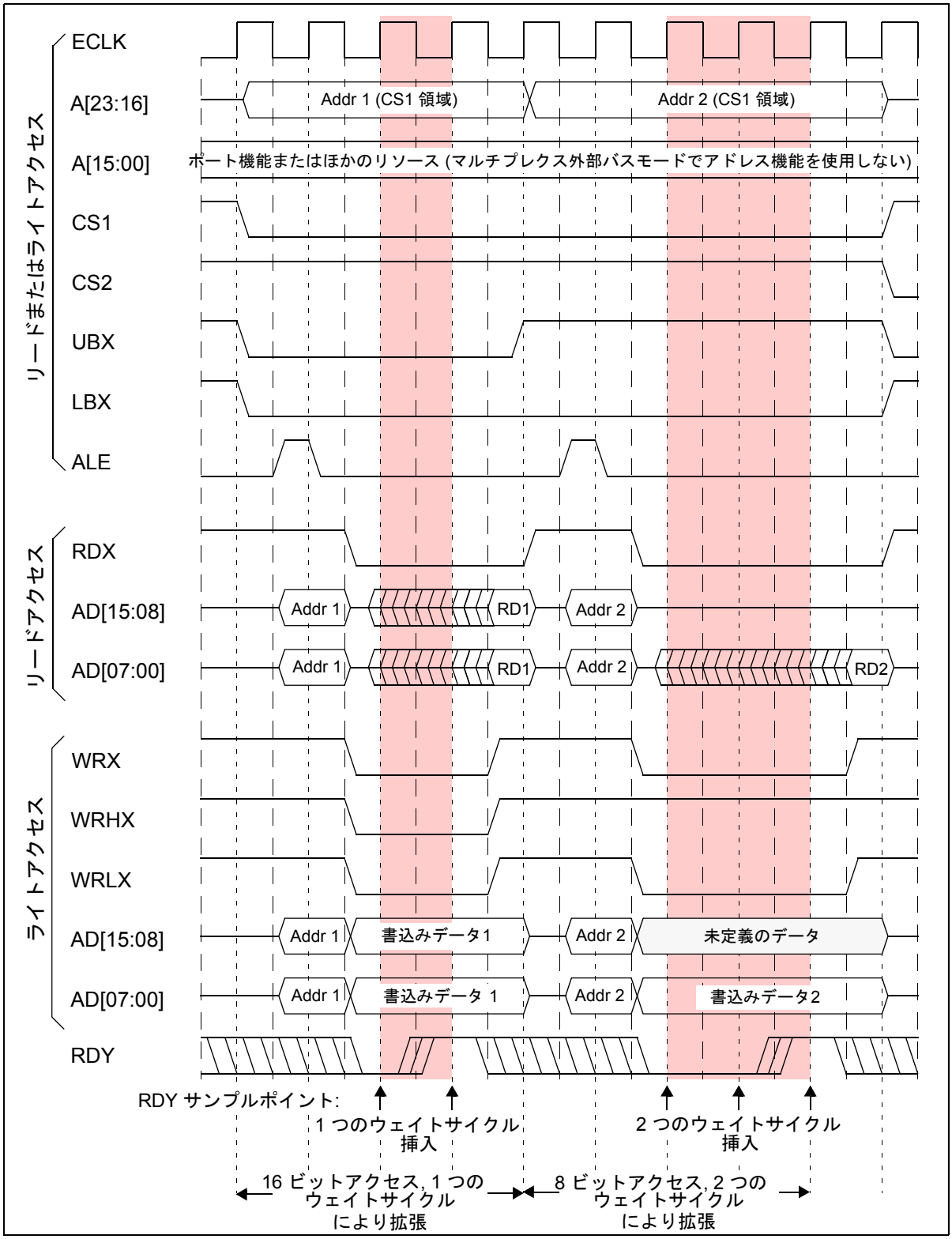
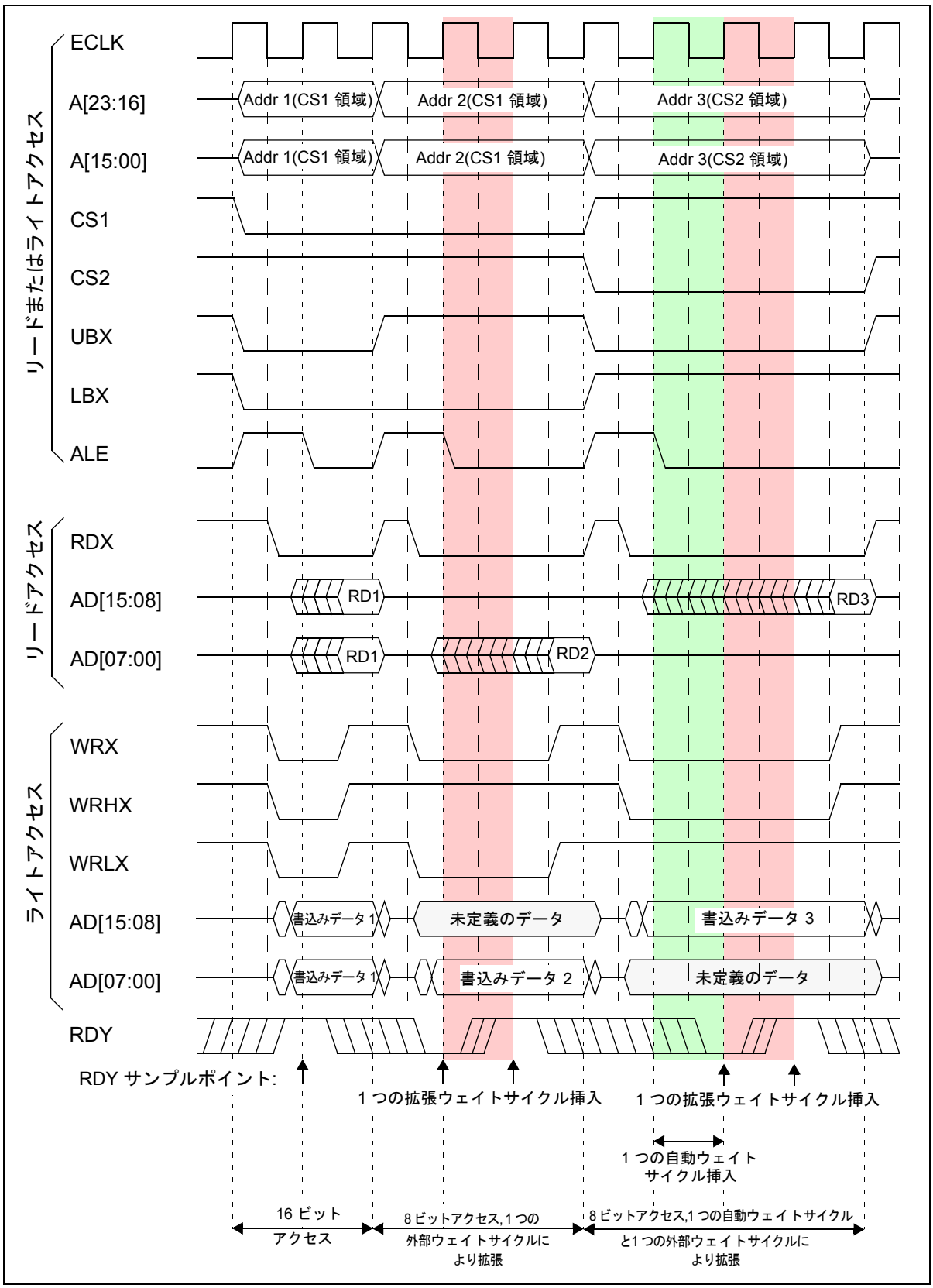


Figure 12-23. ノンマルチプレクス16ビットバス: 自動および外部レディ機能によって拡張されたデータサイクル



12.3.4 ホールド機能

ホールド機能は, 複数のマスタデバイスが単一の外部バスにアクセスする場合に使用されます。外部バスクロック/機能レジスタ(EBCF) の HDE ビットを"1"にセットすることによりホールド機能は有効となります。HRQ 入力端子に"1"を入力すると, 現在実行中の外部バスアクセスが完了した後, 外部バスの端子はHi-Z状態となります。本デバイスの外部バス端子がHi-Zの状態では, ほかのデバイスが外部バスにアクセスできます。ホールド状態は HAKX端子により確認が可能です。

12.3.4.1 ホールド機能

ホールド機能は , 複数のデバイスが単一の外部バスに接続されていて , その中の複数のデバイスがバスマスタとして動作し得る場合に必要です。この場合 , バスマスタとなるデバイス以外の出力を禁止し , それらの外部バス端子の状態をハイインピーダンスとする必要があります。

本デバイス以外のマスタデバイスからのバス権要求は , 本デバイスの HREQ 端子へのハイレベル入力によって示されます。次に , 必要なすべての外部バス端子が Hi-Z に設定され , HAKX 端子が "0" にセットされて , 本デバイス以外のマスタデバイスがその外部バスアクセスを開始できるようになります。以上の動作を行うためには , EBCF:HDE ビットを "1" にセットすることによって本デバイスのホールド機能を許可する必要があります。

外部バス動作の処理中にホールド要求が発生した場合 , 外部バス動作の完了後にホールド状態となり , HAKX 端子が "0" にセットされます。8 ビット外部バスモードで 16 ビットアクセスを行っている場合 (これは 2 つのアクセスに分割されます), 両方のアクセス完了後となります。

MCU が外部アクセスの開始を試みるまでは , MCU は外部バス端子のホールド状態を継続します。

MCU がホールド状態の自身の外部バス領域に (CPU または DMA によって) アクセスしようとすると , HRQ 端子が解放されるまで ("0") 内部システムバスが停止されます。ホールド状態が解除されるまで , 内部バス動作を実行できません。

ホールド状態 (HAKX 端子が "0" を出力する) では , 以下の端子がハイインピーダンス状態になります (外部バス出力または外部バス双方向信号として使用される場合)。

- アドレス出力:
  - マルチプレクスおよびノンマルチプレクスモードの A23 ～ A16
  - ノンマルチプレクスモードの A15 ～ A00 (マルチプレクスモードではこれらの端子を使用しません)
- アドレス/データ入出力:
  - すべてのモードの AD15 ～ AD00
- バス制御信号:
  - ストローブ信号 : ALE/ASX, RDX, WRLX/WRX, WRHX
  - バイト選択信号 : LBX, UBX
  - チップセレクト信号 : CS[5:0]

バスクロック (ECLK) は , EBCF:CSM =1 (クロック一時停止モード) の場合だけホールド状態で Hi-Z に設定されます。

ハイインピーダンス状態になるためには, 対応するデータ方向レジスタ (DDR) を "0" (入力) にセットし, 同じ端子にマッピングされているほかのすべてのリソースを禁止する必要があります。一部の端子がほかのマスタによってドライブされない場合, 内部または外部プルアップまたはプルダウン抵抗を使って浮動制御信号を防止することができます。ホールド状態のときに, 選択した端子を "1" または"0"にドライブするには, 対応する DDR レジスタを "出力" に設定し, 必要な端子レベルを PDR レジスタに書き込みます。HRQ 端子を"0"に戻すことによりホールド状態が保持されます。このとき HAKX端子はハイレベルを出力し, ほかのすべての出力端子はホールド状態の前にドライブしていた値を回復します。

12.3.4.2 スタンバイモード時のホールド要求

ホールド要求は , すべてのスタンバイモード (スリープ , タイマおよびストップ モード) で認識可能です。ホールド要求入力信号 (HRQ) によって , 外部バス端子は非同期に Hi-Z 状態となります (入力信号は , 同期動作のランモードのように ECLK でサンプリングされません)。

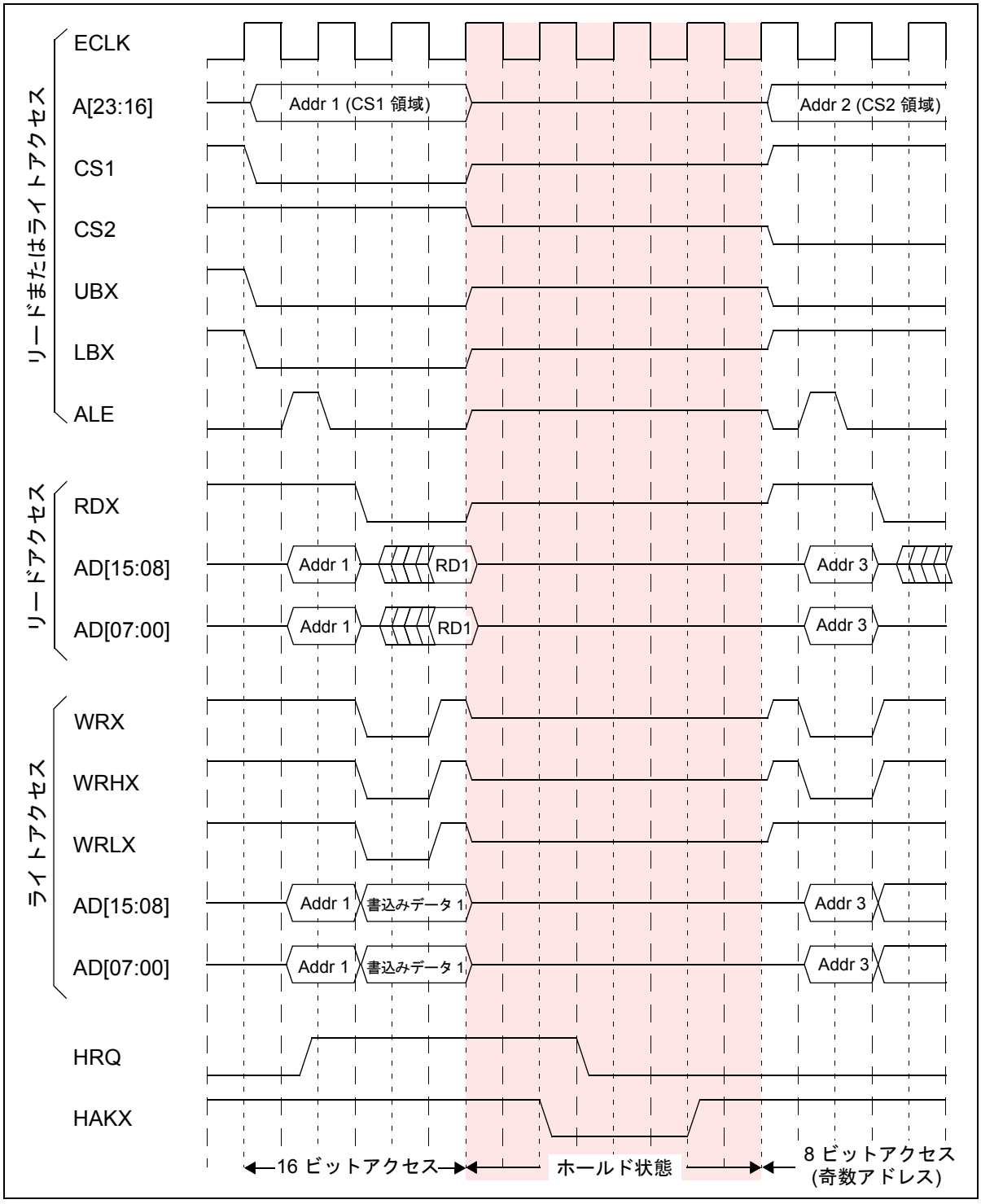
12.3.4.3 ウォッチドッグタイマ

ホールド機能を使用することにより,CPUまたはDMAが外部バスにアクセスしようとした場合にMCUの内部動作が停止することがあります。そのような場合, ウォッチドッグタイマをクリアするためのソフトウェアルーチンを実行できません。

12.3.4.4 タイミングダイヤグラム

Figure 12-24 は , ホールド機能のタイミングを示しています。

Figure 12-24. 16ビットマルチプレクスバスモードでのホールド要求



12.4 外部バスの使用に関する注意事項

外部バスの端子の操作は, 他のリソースの端子と同じです。ここでは, 外部バスインタフェースを使用するために必要な端子を設定する方法を説明します。

入力端子のセットアップ

ホールド機能または外部レディ機能を使用 , または双方を使用する場合 , 対応する入力端子 (HRQ および RDY) を対応する DDR (データ方向レジスタ) および PIER (入力許可レジスタ) で入力に設定する必要があります。EBCF: HDE ビット および EBCF: RYE ビット は, 外部バスモジュールの対応する機能を許可するだけであり , 対応する端子の入力を許可しません。

これらの端子に接続されている他のリソースの出力機能を禁止する必要があります。

出力端子のセットアップ

データ端子の出力機能 (AD[15:08] および AD[07:00] 端子のデータサイクル) は , 選択したバスモード (8 または 16 ビットデータ幅) に従って自動的に有効となります。

全モードのアドレス端子について , 外部バスアドレス出力許可レジスタ (EBAE[2:0]) でアドレス信号の出力を許可する必要があります。外部バスアクセス時にはアドレスは端子にドライブされるだけです。したがって 2 つの外部バスアクセス間の一時停止時のアドレス端子の端子状態は , 対応する DDR および PDR レジスタの設定によって異なります。

すべての制御信号 (ALE/ASX, RDX, WRLX/WRX, WRHX, LBX, UBX, CS[5:0]）およびクロック (ECLK) は , その専用の出力許可信号によって有効となります。HAKX 出力機能は , EBCF:HDE ビットによって有効となります。

これらの制御信号の出力が対応するビットで有効となった場合 , 外部バスがホールド状態でない限り , これらの端子の DDR の値は無視されます。外部バス端子をホールド状態で Hi-Z に設定するには , DDR を "0"(入力) にセットしてください。

使用している外部バス出力端子と兼用される別のリソースは常に動作禁止としてください。

外部バス機能として使用しない出力端子は , ポート I/O として , または同じ端子に割り当てられている別のリソース機能として使用可能です。

入出力端子

AD15 ～ AD00 端子はデータの読出しおよび書込み , およびマルチプレクスモードでのアドレス出力に使用します。

使用する端子 (8 ビット外部バスモードと 16 ビット外部バスモードで異なります) の入力機能は , 対応する入力許可レジスタ (PIER) のビットを "1" にセットすることによって有効となります。また , データ方向レジスタ (DDR) の対応するビットを入力に設定し , これらの端子に接続されているほかのリソースの出力機能を禁止する必要があります。

外部バス端子およびその設定

下の表は, 外部バス端子と, 外部バスインタフェースの各動作モードにおける端子の状態を示しています。

Table 12-6. 外部バス端子の状態

端子名	状態	動作モード			
		8 ビット ノンマルチプレクス	16 ビット ノンマルチプレクス	8 ビット マルチプレクス	16 ビット マルチプレクス
AD[07:00]	アクセス時	データ入出力		データ入出力 アドレス出力 (EBAE0有効時)	
	バスアイドル状態	Hi-Z			
	ホールド状態				
AD[15:08]	アクセス時	未使用	データ入出力	アドレス出力 (EBAE1有効時)	データ入出力 アドレス出力 (EBAE1有効時)
	バスアイドル状態		Hi-Z	Hi-Z	Hi-Z
	ホールド状態				
A[07:00]	アクセス時	アドレス出力 (EBAE0有効時)		未使用	
	バスアイドル状態	Hi-Z			
	ホールド状態				
A[15:08]	アクセス時	アドレス出力 (EBAE1有効時)		未使用	
	バスアイドル状態	Hi-Z			
	ホールド状態				
A[23:16]	アクセス時	アドレス出力 (EBAE2有効時)			
	バスアイドル状態	Hi-Z			
	ホールド状態				
ECLK	アクセス時	外部バスクロック出力 (EBCF:CKE, CKIに依存)			
	バスアイドル状態	外部バスクロック出力 (EBCF:CKE, CMSに依存)			
	ホールド状態	EBCF:CSM=0 かつ EBCF:CKE=1: 外部バスクロック出力 EBCF:CSM=1または EBCF:CKE=0: Hi-Z			
RDY	アクセス時	RDY 入力 (EBCF:RYE 有効時)			
	バスアイドル状態				
	ホールド状態				
$\overline{WRH}$	アクセス時	未使用 (EBCS:WRHE 有効時 "1" を出力)	WRHX出力 (EBCS:WRHE 有効時)	未使用 (EBCS:WRHE 有効時 "1" を出力)	WRHX出力 (EBCS:WRHE有効時)
	バスアイドル状態	EBCS:WRHE有効時"1"を出力			
	ホールド状態	Hi-Z			
$\overline{WRL}$/$\overline{WR}$	アクセス時	WRLX/WRX出力 (EBCS:WRLE 有効時)			
	バスアイドル状態	EBCS:WRLE有効時"1"を出力			
	ホールド状態	Hi-Z			

Table 12-6. 外部バス端子の状態

端子名	状態	動作モード			
		8 ビット ノンマルチプレクス	16 ビット ノンマルチプレクス	8 ビット マルチプレクス	16 ビット マルチプレクス
$\overline{RD}$	アクセス時	RDX出力 (EBCS:RDE有効時)			
	バスアイドル状態	EBCS:RDE 有効時"1"を出力			
	ホールド状態	Hi-Z			
ALE/$\overline{AS}$	アクセス時	ALE/ASX出力 (EBCS:ASE 有効時)			
	バスアイドル状態	EBCS:ASE 有効時, 非アクティブレベルを出力			
	ホールド状態	Hi-Z			
$\overline{UB}$	アクセス時	未使用 (EBCS:UBE 有効時 "1"を出力)	UBX出力 (EBCS:UBE有効時)	未使用 (EBCS:UBE 有効時 "1" を出力)	UBX出力 (EBCS:UBE 有効時)
	バスアイドル状態	EBCS:UBE 有効時"1"を出力			
	ホールド状態	Hi-Z			
$\overline{LB}$	アクセス時	LBX出力 (EBCS:LBE 有効時)			
	バスアイドル状態	EBCS:LBE 有効時"1"を出力			
	ホールド状態	Hi-Z			
HRQ	アクセス時	HRQ 入力機能 (EBCF:HDE 有効時)			
	バスアイドル状態				
	ホールド状態				
$\overline{HAK}$	アクセス時	ホールド機能が EBCF:HDE有効時"1"を出力			
	バスアイドル状態				
	ホールド状態	"0"を出力			
CS[5:0]	アクセス時	CS[5:0] 出力 (EACH[5:0]:CSE 有効時)			
	バスアイドル状態	EACH[5:0]:CSE 有効時, 非アクティブレベルを出力			
	ホールド状態	Hi-Z			

表の説明

未使用	• 端子は外部バスモジュールによる駆動はおこなわれず, 汎用 I/O として, または同じ端子に割り当てられるリソースの入出力として使用できます。
データ	• PIER を "1" にセットすることによって対応する入力を有効としてください。そうしなければデータを入力できません。 • 外部バスインタフェースは , 書込みサイクル時に書込みデータを端子に出力します。 • 同じ端子を共有するほかのリソース出力を禁止する必要があります。 • 対応する DDR を "0" (入力)にセットする必要があります。そうしなければデータを入力できません。

xxx 出力 (Exxx有効時)	• この端子は , 外部バス出力信号として使用できます。 • 外部バス機能の出力は , Exxx レジスタの対応する出力許可ビットによって許可 / 禁止できます。 • 端子を外部バス出力として使用する場合 , 同じ端子を共有するほかの出力リソースを禁止する必要があります。 • 端子を外部バス出力として使用し , ホールド機能を使用する場合は , 対応する DDR を入力に設定します。DDR を入力に設定しない場合 , ホールド状態では端子状態は Hi-Z とはならずに , PDR 値が出力されます。 • 端子を外部バス出力として使用しない場合, その端子をポート端子またはほかのリソースの出力として使用できます。この場合, ホールド機能は無効です。
xxx 入力	• この端子は , 外部バス入力端子として使用できます。 • PIER を"1"にセットすることによって対応する入力を許可する必要があります。 • 同じ端子を共有するほかのリソース出力を禁止する必要があります。 • 対応する DDR を"0" (入力)にセットする必要があります。
Hi-Z	• 対応する DDR を設定し , 入力状態 /Hi-Z 出力としてください。使用するすべての外部バス出力端子と同様に , 端子を共有するリソース出力を禁止する必要があります。 • DDR を出力に設定した場合, Hi-Z ではなく, 対応する PDR レジスタ値が出力されます。

ホールド状態または外部バスアクセス一時停止時に端子出力を Hi-Z 以外とする場合 , DDR を出力に設定し , 必要な値を対応する端子の PDR レジスタに書き込みます。外部バスが端子状態を Hi-Z に変更するため , その出力を無効としたとき , PDR 値が端子に出力されます。

＜注意事項＞

端子が外部バスモジュールによって出力駆動されていないとき , プルアップ抵抗を使用してその端子状態を終端処理することもできます。

12.5 外部ブートベクタフェッチ

外部バスインタフェースを備えたデバイスでは, 外部バスインタフェースを通じてブートベクタ(ユーザプログラムの開始アドレス)を読み出すことができます。本機能により, MD2～MD0モード端子の設定に従って, 3通りの外部バス動作モードを実現します。
ブート ROM プログラムは, モード端子設定に従って外部バスインタフェースを初期化します。ブートベクタと共に外部バスモードバイトが読み取られ, ユーザプログラムの実行の開始前に外部バスモードレジスタ EBM に書き込まれます。

外部ブートベクタフェッチでのモード端子の設定

外部ブートベクタフェッチで, 以下のモード端子設定を使用できます。

Table 12-7. 外部ベクタモードでのモード端子の設定

MD[2:0]	動作モード
000	8 ビットデータ幅 アドレス/データマルチプレクス
001	16 ビットデータ幅 アドレス/データマルチプレクス
110	8 ビットデータ幅 アドレス/データノンマルチプレクス

ブートベクタおよび外部バスモードバイトの位置

モード端子 MD2～MD0によって外部ベクタモードが選択されている場合, 開始データ(ブートベクタおよび外部バスモードバイト)は, 外部バスのアドレス FF:FFDC$_H$～FF:FFDF$_H$ から読み出されます。当該アドレス範囲のデータ配置は以下となります。

Table 12-8. ブートベクタおよび外部バスモードバイト

アドレス	上位バイト bit[15:08]	下位バイト bit[07:00]
FF:FFDD/DC$_H$	開始アドレスビットbit15～bit8	開始アドレスビットbit7～bit0
FF:FFDF/DE$_H$	外部バスモードバイト	開始アドレスビットbit23～bit16

外部バスモードバイトは , ユーザプログラムの実行の開始前に外部バスモードレジスタ EBM に書き込まれます。
外部バスモードバイトの構成は , 外部バスインタフェースの外部バスモードレジスタの構成と同じです。詳細は「12.2.1 外部バスモードレジスタ (EBM)」を参照してください。

外部ブートベクタフェッチの順序

1. 内部リセットが解除されます。
2. CR クロックによって MCU がウェイクアップし , ブート ROM スタートアッププログラムを起動します。
3. ブート ROM プログラムがモード端子設定を読み出します -> 外部ベクタモードが検出されます。
4. ブート ROM プログラムは , モード端子設定に従って以下のレジスタを設定します。

Table 12-9. ブートROMプログラムによる外部開始ベクタ読出し前のレジスタ設定内容
(MB96380シリーズ ノンマルチプレクスバスモード(MD[2:0]='110')を除く)

レジスタ	構成ビット	説明
EBM	EAE[5]='1'	外部領域 5 をアクティブ化
	ERE='1'	外部 ROM を許可
	MD2～MD0=110_B の場合: NMS=1	MD2～MD0=110_Bの場合, 外部バスをノンマルチプレクスモードに設定
EBAE[2:0]	A[23:0]='11...11'	すべてのアドレス出力端子を許可
EBCS	RDE='1'	リードストローブ出力を許可
	ASE='1'	アドレスストローブ出力を許可
EACL5	R[2:0]='011'	自動レディ機能を 3 ウェイトサイクルに設定
	MD2～MD0=000_B または 110_B の場合: BW=1	MD2～MD0=000_Bまたは 110_Bの場合, 外部バス幅を "8 ビット" に設定
EACH5	CSE='1'	チップセレクト CS5 出力を許可
PIER (AD[15:00]端子用)	PIER[AD[07:00]]='1'	AD[07:00]データ端子の入力を許可
	MD2～MD0=001_B の場合: PIER(AD15 ～ AD08) = 1	16 ビット外部開始ベクタフェッチモードの場合, AD[15:08]データ端子の入力を許可

Table 12-10. ブートROMプログラムによる外部開始ベクタ読出し前のレジスタ設定内容
(MB96380シリーズ ノンマルチプレクスバスモード(MD[2:0]='110'))

レジスタ	構成ビット	説明
EBM	EAE[4]='1'	外部領域 4 をアクティブ化
	ERE='1'	外部 ROM を許可
	NMS='1'	外部バスをノンマルチプレクスモードに設定
EBAE[2:0]	A[23:0]='11...11'	すべてのアドレス出力端子を許可
EBCS	RDE='1'	リードストローブ出力を許可
	ASE='1'	アドレスストローブ出力を許可
EACL4	R[2:0]='011'	自動レディ機能を 3 ウェイトサイクルに設定
	BW='1'	MD2～MD0=110_Bの場合, 外部バス幅を "8 ビット" に設定
EACH4	CSE='1'	チップセレクト CS4 出力を許可
PIER (AD[15:00]端子用)	PIER[AD[07:00]]='1'	AD[07:00]データ端子の入力を許可

5. ブート ROM プログラムが外部バスからアドレス領域 $FF:FFDC_H$ ～ $FF:FFFF_H$ を読み出します。
6. ブート ROM プログラムが外部バスモードバイトを EBM レジスタに書込みます。ここで外部バスインタフェースは , 外部バスモードバイトの定義に従い設定されます。
7. ブート ROM プログラムは , 外部バスから読み出されたユーザプログラム開始アドレスにジャンプします。
8. ユーザプログラムが実行されます。外部バスインタフェースのすべてのレジスタは , いつでもソフトウェアによって書き換えできます。

＜注意事項＞

外部ブートベクタフェッチ時には , 外部バスクロック (ECLK) およびバイト選択信号 UBX/LBX は有効となりません。

12.6 MCU状態ごとの端子状態

ここでは , 以下の MCU 状態での外部バス端子の状態を説明します。

- リセット
- 外部バスアクセスを使用する ランモード
- 内部アクセスを使用する ランモード
- スタンバイモード
- ホールド状態

12.6.1 外部バス端子の状態

Table 12-6 は , MCU 状態毎の各外部バス端子の状態を説明しています。この説明は , 外部バスが許可されている場合にだけ有効です。外部バスモードでの端子の使用については , Table 12-6. を参照してください。外部バス端子の Hi-Z 状態を保証するために ,「12.4 外部バスの使用に関する注意事項」に示している推奨手順に従ってください。

Table 12-11. 外部バスの各端子の状態

<table>
<tr><th rowspan="2">端子名</th><th rowspan="2">リセット</th><th colspan="3">ランモード</th><th rowspan="2">スタンバイモード + SMCR:SPL='0'*1</th><th rowspan="2">外聞バスホールド状態</th></tr>
<tr><th colspan="2">外部バスアクセス</th><th>内部アクセス</th></tr>
<tr><td>AD[07:00]</td><td rowspan="15">入力禁止 /Hi-Z を 出力</td><td colspan="2">データおよび (または) アドレスに使用</td><td colspan="3">Hi-Z を出力</td></tr>
<tr><td>AD[15:08]</td><td colspan="2">データおよび (または) アドレスに使用可能 (バスモードによって異なる)</td><td colspan="3">外部バスに使用する場合は, Hi-Z を出力</td></tr>
<tr><td>A[07:00]</td><td colspan="2" rowspan="2">許可されている場合, 出力 (ノンマルチプレクスバスモードのみ)</td><td colspan="3" rowspan="2">Hi-Z を出力(ノンマルチプレクスバスモードのみ)</td></tr>
<tr><td>A[15:08]</td></tr>
<tr><td>A[23:16]</td><td colspan="2" rowspan="8">許可されている場合は 出力</td><td>Hi-Z を出力</td><td rowspan="5">Hi-Z を出力</td></tr>
<tr><td>ALE/$\overline{AS}$</td><td>許可されている場合は非アクティブレベルを出力</td></tr>
<tr><td>$\overline{RD}$</td><td rowspan="3">許可されている場合は"H" を出力</td></tr>
<tr><td>$\overline{WRL}$/$\overline{WR}$</td></tr>
<tr><td>$\overline{WRH}$</td></tr>
<tr><td>$\overline{UB}$/$\overline{LB}$</td><td rowspan="2">許可されている場合は非アクティブレベルを出力</td></tr>
<tr><td>CS[5:0]</td></tr>
<tr><td>ECLK</td><td>許可されている場合は出力 (状態およびアクティビティはクロック構成によって異なる)</td><td>許可されている場合は "H" または "L" を出力 (EBCF:CSM=1および DIV[2:0] ≠ 000_B の場合は非アクティブレベル)</td><td>EBCF:CSM=0 および CKE=1: アクティブクロック そうでない場合は Hi-Z</td></tr>
<tr><td>RDY</td><td colspan="2">許可されている場合は 入力</td><td colspan="3">(未使用の入力)</td></tr>
<tr><td>HRQ</td><td colspan="4">許可されている場合は入力</td><td>入力</td></tr>
<tr><td>$\overline{HAK}$</td><td colspan="4">許可されている場合は"H" を出力</td><td>"L"出力</td></tr>
</table>

*1: スタンバイモードには, 次の種類があります: スリープ, タイマおよび ストップ モード; 外部バスはホールド状態ではない

ECLKに関する注意事項

外部バスアクセスを実行していない場合,ECLKの状態はEBCF:DVI[2:0],EBCF:CKIおよびEBCF:CSMビットの設定によって異なります。MCU がスタンバイモードに設定されている場合, 外部バスインタフェースを提供しているクロックが停止し, その結果, 外部バスクロック (ECLK) も停止します。したがってスタンバイモードでの ECLK の状態は,スタンバイモードへの遷移時のECLKの状態によって異なります。クロック一時停止モードが禁止されている場合またはECLKクロック分周器が禁止されている場合(EBCF:DVI[2:0]="000"), この状態は, "H"または"L"となります。クロック一時停止モードが許可されている場合, これは常に EBCF:CKI によって定義された非アクティブレベルとなります。

13. I/Oポート

I/O ポートの機能と動作について説明します。

13.1 I/Oポート

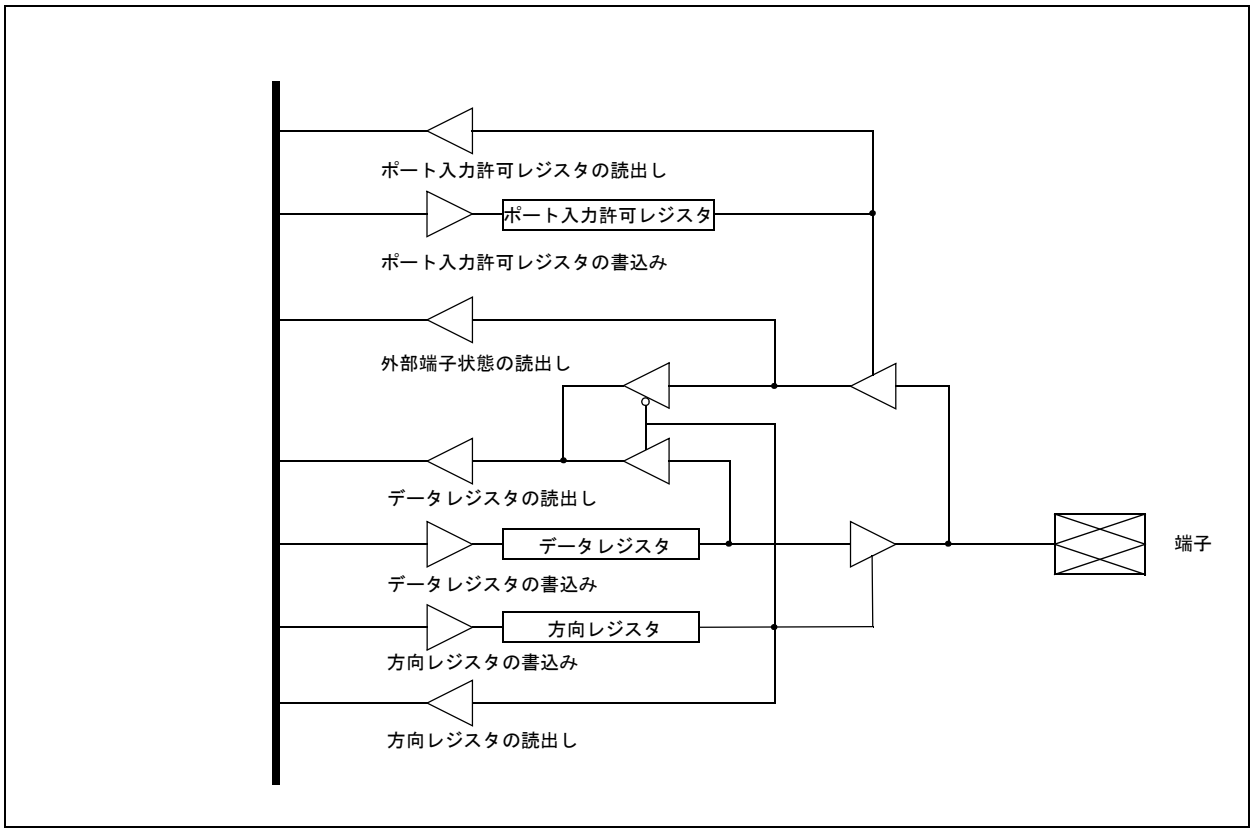
ポートの各端子は, 対応する周辺が端子を用いない設定となっているとき, 方向レジスタによって入力,出力の指定を行うことができます。入力として端子を使用する前に, ポート入力許可レジスタの設定により入力を許可する必要があります。端子が入力と指定されている場合は, 端子の論理レベルが読み出されます。端子が出力と指定されている場合は, データレジスタ値が読み出されます。上記は, リードモディファイライト系命令の読出しでも同じです。

I/Oポート

端子が他の周辺機能の出力として使用されている場合, 方向レジスタ値とは無関係に周辺の出力値が読み出されます。

通常は , ポートを出力に設定するために先立って , リードモディファイライト系命令をデータレジスタの設定に使用しないことを推奨します。これは , この場合のリードモディファイライト系命令がレジスタ値ではなく , ポートの論理レベルを読み出すことになるからです。

Figure 13-1. I/Oポートのブロックダイヤグラム



13.2 I/Oポートレジスタ

各汎用ポート端子 GPxx は , 9 つのタイプのレジスタにより制御されます。これらのレジスタは以下のとおりです。

- ポートデータレジスタ (PDR00 ～ PDRnn)
- 外部端子状態レジスタ (EPSR00 ～ EPSRnn)
- ポート方向レジスタ (DDR00 ～ DDRnn)
- ポート入力許可レジスタ (PIER00 ～ PIERnn)
- ポート入力レベルレジスタ (PILR00 ～ PILRnn)
- 拡張ポート入力レベルレジスタ (EPILR00 ～ EPILRnn)
- ポート出力駆動レジスタ (PODR00 ～ PODRnn)
- ポートhigh駆動レジスタ (PHDR00 ～ PHDRnn)
- プルアップ制御レジスタ (PUCR00 ～ PUCRnn)

I/Oポートレジスタ

汎用ポート端子 GPxx_y は I/O ポートレジスタ xx の bit y によって制御されます。例えば , ポート端子 GPxx_y のデータ方向は , DDRxx_y によって制御されます。

Figure 13-1. I/Oポートレジスタ(ポート00のみ)

ポート 0 データレジスタ (PDR00)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
P7	P6	P5	P4	P3	P2	P1	P0

ポート 0 外部端子状態レジスタ (EPSR00)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
PS7	PS6	PS5	PS4	PS3	PS2	PS1	PS0

ポート 0 方向レジスタ (DDR00)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
D7	D6	D5	D4	D3	D2	D1	D0

ポート 0 入力許可レジスタ (PIER00)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
IE7	IE6	IE5	IE4	IE3	IE2	IE1	IE0

ポート 0 入力レベルレジスタ (PILR00)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
IL7	IL6	IL5	IL4	IL3	IL2	IL1	IL0

ポート 0 拡張入力レベルレジスタ (EPILR00)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
EIL7	EIL6	EIL5	EIL4	EIL3	EIL2	EIL1	EIL0

ポート 0 出力駆動レジスタ (PODR00)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
OD7	OD6	OD5	OD4	OD3	OD2	OD1	OD0

ポート 0 high 駆動レジスタ (PHDR00)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
HD7	HD6	HD5	HD4	HD3	HD2	HD1	HD0

ポート 0 プルアップ制御レジスタ (PUCR00)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0

13.2.1 ポートデータレジスタ(PDRnn)

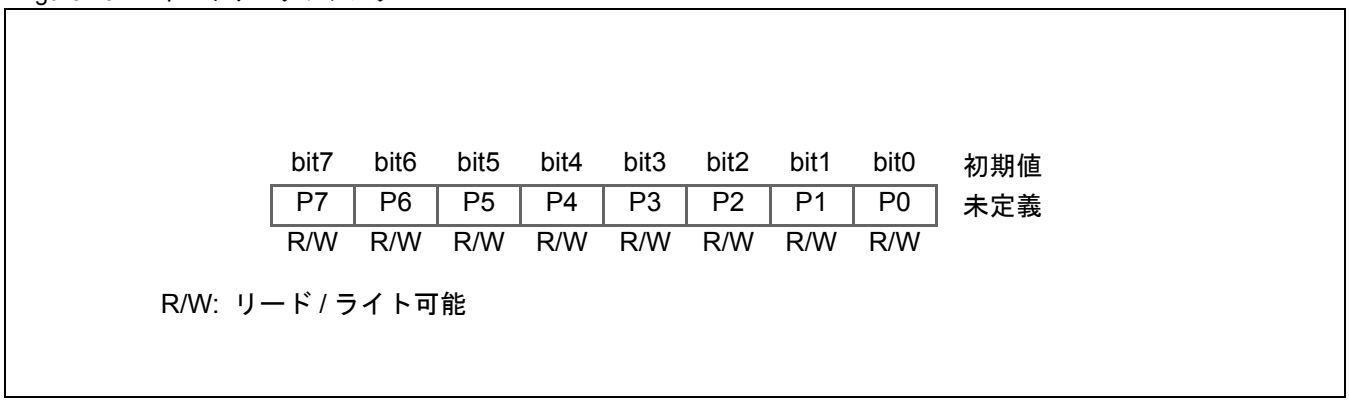
I/O ポートのリード / ライトは , 以下の点でメモリのリード / ライトと異なります。

■ 入力モード
リード : 対応する端子のレベルが読み出されます。
ライト : 出力用のラッチに書き込まれます。

■ 出力モード
リード : データレジスタラッチ値が読み出されます。
ライト : 出力用のラッチに書き込まれ , 対応する端子に出力されます。

13.2.1.1 ポートデータレジスタ

Figure 13-2. ポートデータレジスタ



13.2.1.2 ポートデータレジスタの読出し

Bits Px (PDR00 ～ PDRnn)

ポートデータレジスタの読出し値は,データ方向レジスタの対応するビット, 同じ端子に接続されているリソース(もしあれば)の現在の状態, および使用している命令(RMW かそれ以外か) によって異なります。以下の組み合わせとなります。

Table 13-1. ポートデータレジスタの読出し

DDR値	リソース出力	端子値	'リード命令'の読出し値	'RMW系命令'の読出し値
0 (入力)	許可	リソース出力の値	入力端子状態	PDR の値
	禁止	ポート入力	入力端子状態	PDR の値
1 (出力)	許可	リソース出力の値	PDR の値	PDR の値
	禁止	PDR の値	PDR の値	PDR の値

RMW 系命令による PDR 値の読出しでは , 誤って PDR の単一ビットが上書きされないように注意してください。端子のリソース機能が選択されていて , PDR 値の代わりに端子値が読み出される場合にそのような上書きが発生することがあります。

F^2MC-16LX と F^2MC-16FX の RMW 系命令による PDR の読出しの違いについて, 以下の例を参照してください。

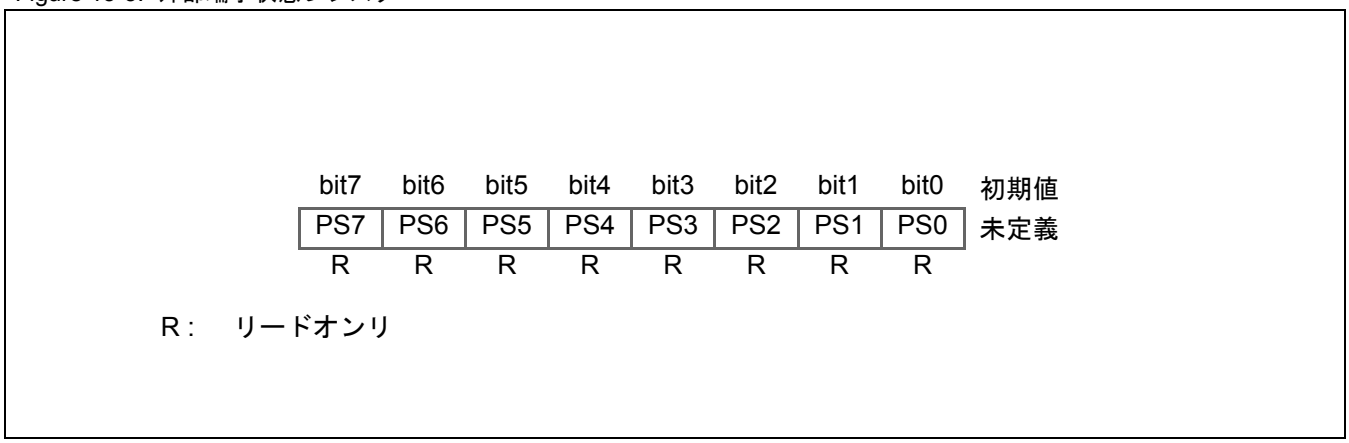
Bit #	7654.3210
DDR	1111.1111 (= 出力)
PDR (= 最後に PDR に書き込まれた値)	0000.0000
OE (リソース出力許可)	0010.0000
リソース出力	0010.0000
現在の端子値	0010.0000
16LXでのPDRからのRMW系命令による読出し	0010.0000 -> 例：OR PDR,#0 => #b'0010.0000 を書き戻す => PDR が変更される
16FXでのPDRからのRMW系命令による読出し	0000.0000 -> 例：OR PDR,#0 => #b'0000.0000 を書き戻す => PDR は変更されない

13.2.2 外部端子状態レジスタ(EPSRnn)

このレジスタによって外部端子の現在の状態を読み出すことができます。

13.2.2.1 外部端子状態レジスタ

Figure 13-3. 外部端子状態レジスタ



13.2.2.2 外部端子状態データレジスタへのアクセス

Bits PSx (EPSR00 ～ EPSRnn)

これらのビットは外部ポート端子の状態のミラーとなります。

このレジスタへの書込みは無効です。

ポート入力許可レジスタの対応するビットが"1"に設定されると外部端子状態レジスタは外部ポート端子の状態のミラーとなります。

13.2.3 データ方向レジスタ(DDRnn)

端子を汎用ポートとして使用するとき, このレジスタは対応する端子を入力モードまたは出力モードに切り換えます。

13.2.3.1 データ方向レジスタ

Figure 13-4. データ方向レジスタ

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
D7	D6	D5	D4	D3	D2	D1	D0	00000000_B
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W: リード / ライト可能

13.2.3.2 データ方向レジスタの読出し

Bits Dx (DDR00 ～ DDRnn)

これらのビットは , 各ポート端子のデータ方向を設定します。

端子をポートとして使用するとき, 対応する端子は下記のように制御されます。

Table 13-2. モード選択

0	入力モード
1	出力モード

＜注意事項＞

データ方向レジスタは, 対応するリソースの状態と関係なく読み出すことができます。しかし, DDRの値は , ポートデータレジスタの読出しアクセスの結果に影響します。
端子をアナログ入力またはアナログ出力として使用する場合は , この値は無視されます。

13.2.4 ポート入力許可レジスタ(PIERnn)

端子を汎用ポートとして使用するとき, このレジスタによってデジタル入力が許可または禁止されます。デジタル入力が禁止された場合, 貫通電流は発生しません。

13.2.4.1 ポート入力許可レジスタ(PIER)

Figure 13-5. ポート入力許可レジスタ

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
IE7	IE6	IE5	IE4	IE3	IE2	IE1	IE0	00000000_B
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W ：リード / ライト可能

Bits IEx (PIER00 ～ PIERnn)

これらのビットは , 各ポート端子のデジタル入力を許可または禁止します。

この端子を入力ポートとして使用するとき, 対応する端子は下記のように制御されます。

Table 13-3. ポート許可機能

0	デジタル入力を禁止
1	デジタル入力を許可

＜注意事項＞

- 端子をアナログ入力またはアナログ出力として使用する場合は ， デジタル入力機能は無効にされます。
- MCU がストップモードまたはタイマモードになると , PIER の設定にかかわらず , 外部割込みが起動されているポートを除くすべてのポートの入力が無効にされます。

13.2.5 ポート入力レベルレジスタ(PILRnn)と拡張ポート入力レジスタ(EPILRnn)

これらの2つのレジスタによって各端子のデジタル入力レベルをCMOSヒステリシスからオートモーティブヒステリシスまたはTTL レベルに設定を変更することができます。

13.2.5.1 ポート入力レベルレジスタ(PILRnn) と拡張ポート入力レベルレジスタ(EPILRnn)

Figure 13-6. 入力レベルレジスタと拡張入力レベルレジスタ

PILRnn

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
IL7	IL6	IL5	IL4	IL3	IL2	IL1	IL0	00000000_B
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

EPILRnn

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
EIL7	EIL6	EIL5	EIL4	EIL3	EIL2	EIL1	EIL0	00000000_B
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W ：リード / ライト可能

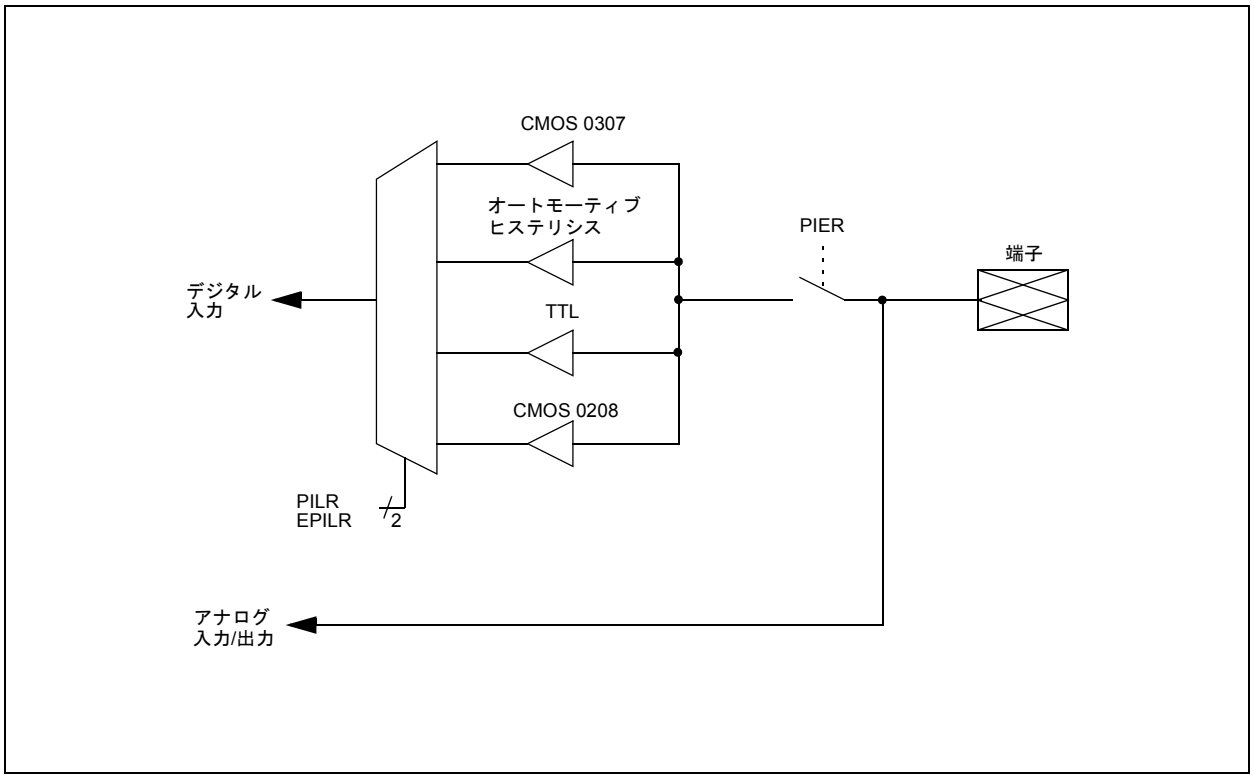
Bits ILx (PILR00 ～ PILRnn) と EILx (EPILR00 ～ EPILRnn)

これらのビットでデジタル入力レベルを設定します。

Table 13-4. 入力レベル選択

ILx	EILx	入力レベル	V_{IL}	V_{IH}
0	0	CMOS (0307)	0.3 x VDD	0.7 x VDD
1	0	オートモーティブヒステリシス	0.5 x VDD	0.8 x VDD
0	1	TTL	0.8V	2.1V
1	1	CMOS(0208)	0.2 x VDD	0.8 x VDD

Figure 13-7. デジタルおよびアナログ入力



13.2.6 ポート出力駆動レジスタ(PODRnn)

このレジスタで出力駆動の設定を変更することができます。
出力駆動力として通常電流と低減電流のどちらかを選択できます。

13.2.6.1 ポート出力駆動レジスタ(PODRnn)

Figure 13-8. ポート出力駆動レジスタ

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
OD7	OD6	OD5	OD4	OD3	OD2	OD1	OD0	00000000_B
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W ：リード / ライト可能

Bits ODx (PODR00 ～ PODRnn)

これらのビットは, 各ポート端子の出力電流について通常値と低減値の切換えを行います。
電気的特性についてはデータシートを参照してください。

Table 13-5. 出力駆動の選択

0	通常電流出力駆動
1	低減電流出力駆動

13.2.7 ポートhigh駆動レジスタ(PHDR)

このレジスタでは high 駆動オプションを許可または禁止することができます。

13.2.7.1 ポートhigh駆動レジスタ

Figure 13-9. ポートhigh駆動レジスタ

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
HD7	HD6	HD5	HD4	HD3	HD2	HD1	HD0	00000000_B
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W ：リード / ライト可能

Bits HDx (PHDR00 ～ PHDRnn)

これらのビットは, 各ポート端子の high 駆動オプションを許可または禁止します。
電気的特性についてはデータシートを参照してください。

Table 13-6. high駆動許可機能

0	high 駆動オプションを禁止 (駆動力は PODR設定によって決定)
1	high 駆動オプションを許可 (PODR 値を無視)

＜注意事項＞

このオプションは , 一部のポートでは使用できません。詳細についてはデータシートを参照してください。

13.2.8 プルアップ制御レジスタ(PUCR)

本レジスタはプルアップ抵抗の使用/不使用を制御します。

13.2.8.1 プルアップ制御レジスタ(PUCR)

Figure 13-10. プルアップ制御レジスタ

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0	00000000_B
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W ：リード / ライト可能

Bits PUx (PUCR00 ～ PUCRnn)

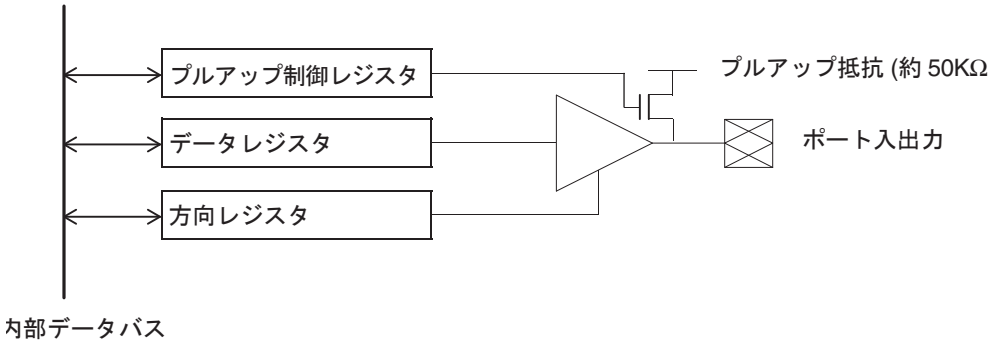
これらのビットは, 各ポート端子のプルアップ抵抗の使用/不使用を設定します。

Table 13-7. プルアップ機能

0	入力モードではプルアップ抵抗なし
1	入力モードではプルアップ抵抗あり

13.2.8.2 プルアップ制御レジスタ(PUCR)のブロックダイヤグラム

Figure 13-11. プルアップ制御レジスタ(PUCR)のブロックダイヤグラム



＜注意事項＞

出力モードでは , このレジスタは無効です (プルアップ抵抗なし)。ストップモードおよびタイマモード (SMCR:SPL=1) では , プルアップ抵抗なしの状態になります (ハイインピーダンス)。

13.3 レジスタの用途

ここでは I/O レジスタの用法の一覧を示します。

ポートの種々の用法に対応するレジスタの設定

Table 13-8. ポートの種々の用法に対応するレジスタの設定

レジスタ/ポートの用途	DDR	PIER	EPSR	PILR	EPILR	PUCR	PODR	PHDR (使用可能な場合)	PDR (リード)	PDR (ライト)
初期状態	"0" (入力)	0 (off)	未定義	0	0	0 (off)	0 (通常)	0 (off)	未定義	PDRに格納される値
汎用入力	"0" にセット (入力)	"1 "にセット (ON)	端子状態	ユーザ設定	ユーザ設定	ユーザ設定	無効	無効	端子状態	値はPDR に格納
汎用出力	"1" にセット (出力)	"1 "にセット (ON) (EPSR)	端子状態 (PIER= 1 の場合)	ユーザ設定	ユーザ設定	無視 (off)	ユーザ設定	ユーザ設定	PDR	値はPDR に格納 端子に出力される
周辺リソース入力 (デジタル)	"0 "にセット (入力)	"1 "にセット (ON)	端子状態	ユーザ設定	ユーザ設定	ユーザ設定	無効	無効	端子状態	値はPDR にのみ格納
周辺リソース出力 (デジタル)	無視	"1 "にセット (ON) (EPSR)	端子状態 (PIER= 1 の場合)	ユーザ設定	ユーザ設定	無視 (off)[a]	ユーザ設定	ユーザ設定	PDR	値はPDR に格納
アナログ I/O	無視	無視 (入力は OFF)	未定義	無視	無視	無視 (off)	無視	無視	未定義	値はPDR に格納

a. 例外： I^2CバスのSDA/SCLとして使用する場合I^2Cバスコントローラを有効に設定するとPUCRはユーザ設定に依存せず,プルアップ抵抗ありとなります。

＜注意事項＞

外部ベクタフェッチ時に外部バスで使用される端子では , 初期状態は PILR=0, EPILR=1 (TTL) です。
周辺リソース出力は , 対応する周辺リソースレジスタによって制御されます。
アナログ I/O は , アナログ周辺リソースのレジスタによって (例 , ADC の ADER によって) 制御されます。
「本ハードウェアマニュアルのアナログ周辺リソース関連の章」を参照してください。

14. 16ビット入出力タイマ

16 ビット入出力タイマの機能と動作について説明します。

14.1 16ビット入出力タイマの概要

16 ビット入出力タイマは, 16 ビットフリーランタイマ, アウトプットコンペアユニット, インプットキャプチャユニットで構成されています。このタイマは, パルスシーケンスの生成および外部イベント間の時間の測定に使用します。

16ビットフリーランタイマ

- 16 ビットフリーランタイマは,アップカウンタ, 制御レジスタ, 16 ビットコンペアレジスタ,およびプリスケーラで構成されています。このカウンタから取得される出力値は, インプットキャプチャユニットの基本タイマとして使用されます。
- カウンタ動作のためのクロックは, 8種類から選択できます。
- カウンタオーバフロー時, またはコンペアレジスタ 0 および 4 との一致時に割込みを発生できます。
- リセット時, ソフトウェアクリア時, または16ビットフリーランタイマとコンペアレジスタ値の一致により, カウンタ値を"0000_H"に初期化できます。

アウトプットコンペアユニット(1モジュールあたり2チャネル)

アウトプットコンペアユニットは , 2 本の 16 ビットコンペアレジスタ , 2 本のコンペア出力用ラッチ , および 2 本のコントロールレジスタから構成されています。

16 ビットフリーランタイマ値が対応するコンペアレジスタ値と一致したときに出力レベルを反転するとともに割込みを発生できます。

- 2 つのコンペアレジスタを各アウトプットコンペアユニットに対して独立して使用できます。
- 2本のコンペアレジスタをペアにして出力端子を制御できます。
- 出力端子の初期値を設定することができます。
- コンペア一致時に割込みを生成できます。

インプットキャプチャユニット(1モジュールあたり2チャネル)

インプットキャプチャは , 独立した 2 本の外部入力端子と対応したキャプチャレジスタ ,
コントロールレジスタから構成されています。外部入力端子から入力された信号の任意エッジを検出することにより , フリーランタイマ値をキャプチャレジスタに保持し , 同時に割込みを発生します。

- 外部入力信号の検出エッジを指定できます。

立上り , 立下り , または両エッジから選択できます。

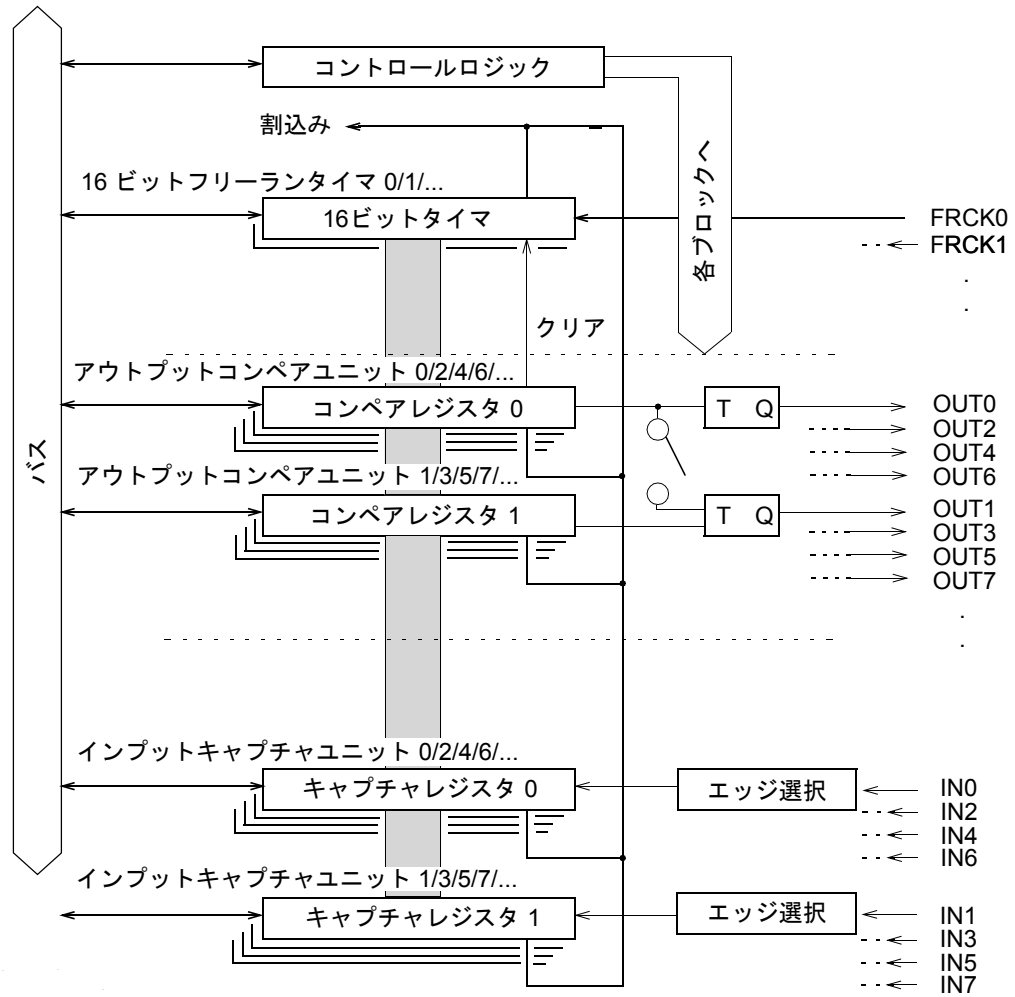
- 2本のインプットキャプチャは独立して動作可能です。
- 外部入力信号の有効エッジ検出により割込み要求が発生します。

インプットキャプチャ割込み時に DMA を起動できます。

16ビットI/Oタイマのブロックダイヤグラム

Figure 14-1 は , 16 ビット I/O タイマのブロックダイヤグラムを示しています。

Figure 14-1. 16ビットI/Oタイマのブロックダイヤグラム



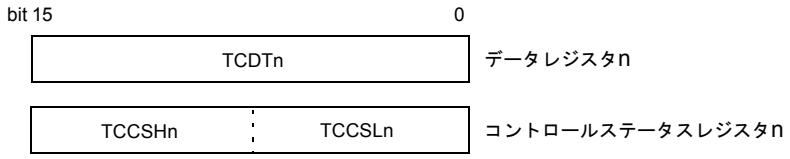
注意：「1.5 16 ビット I/O タイマの構成」を参照して , インプットキャプチャユニット / アウトプットコンペアユニットと , フリーランタイマ — との接続を確認してください。

14.2 16ビット入出力タイマのレジスタ

16 ビット I/O タイマは以下のレジスタで構成されています。

- 16 ビットフリーランタイマのレジスタ
- 16 ビットアウトプットコンペアのレジスタ
- 16 ビットインプットキャプチャのレジスタ

16ビットフリーランタイマのレジスタ

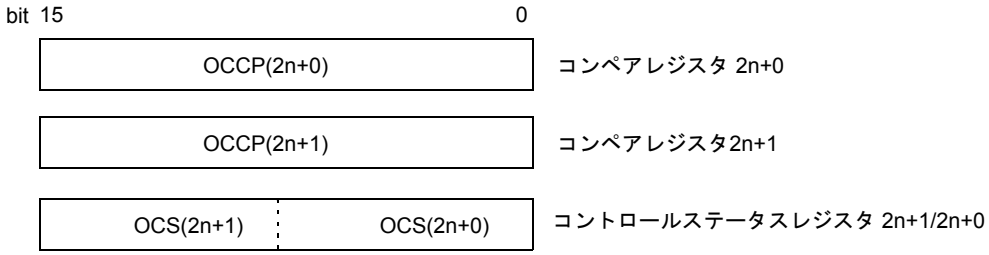


＜注意事項＞

サフィックスの "n" は , フリーランタイマの番号 (0, 1, 2, ...) を表しています。レジスタ名はレジスタタイプ名とサフィックスで構成されます。したがって以下のレジスタ名が使用できます。

- フリーランタイマ 0 の場合 : TCDT0, TCCSH0, TCCSL0
- フリーランタイマ 1 の場合 : TCDT1, TCCSH1, TCCSL1
- 以下同様。

16ビットアウトプットコンペアユニットのレジスタ

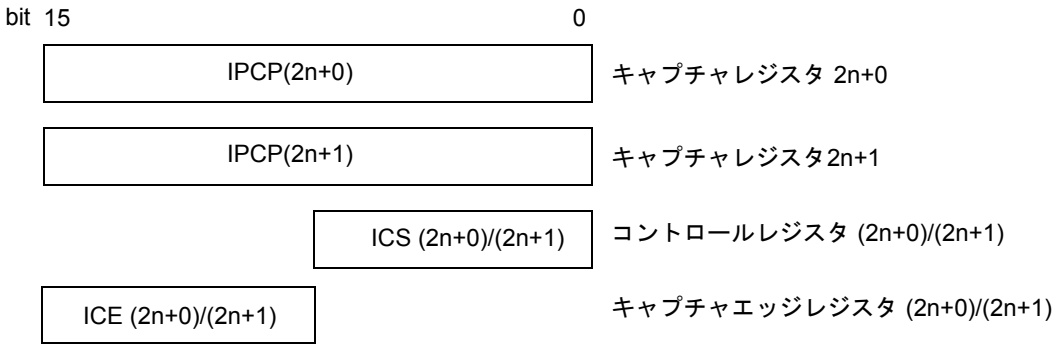


＜注意事項＞

サフィックスの "n" は , アウトプットコンペアユニットの番号 (0, 1, 2, ...) を表しています。レジスタ名はレジスタタイプ名とサフィックスで構成されます。したがって以下のレジスタ名が使用できます。

- アウトプットコンペアユニット 0 の場合 : OCCP0, OCCP1, OCS0, OCS1
- アウトプットコンペアユニット 1 の場合 : OCCP2, OCCP3, OCS2, OCS3
- 以下同様。

16ビットインプットキャプチャレジスタ



＜注意事項＞

サフィックスの "n" は , インプットキャプチャユニットの番号 (0, 1, 2, ...) を表しています。レジスタ名はレジスタタイプ名とサフィックスで構成されます。したがって以下のレジスタ名が使用できます。

- インプットキャプチャユニット 0 の場合 : IPCP0, IPCP1, ICS01, ICE01
- インプットキャプチャユニット 1 の場合 : IPCP2, IPCP3, ICS23, ICE23
- 以下同様。

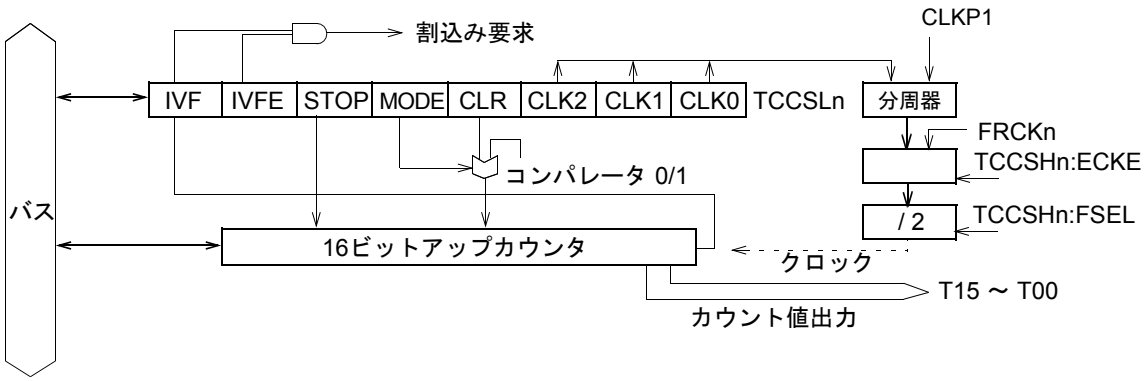
14.3 16ビットフリーランタイマ

16 ビットフリーランタイマは , 16 ビットアップカウンタとコントロールステータスレジスタで構成されています。16 ビットフリーランタイマのカウント値は , アウトプットコンペアユニットおよびインプットキャプチャユニットの基準時間として使用されます。

- カウンタ動作のためのクロックは, 8種類から選択できます。
- カウンタオーバフロー時,またはコンペアレジスタ 0 および 4 との一致時に割込みを発生できます。
- リセット時, ソフトウェアクリア時, また16ビットフリーランタイマとコンペアレジスタ値の一致により, カウンタ値を"0000_H" に初期化できます。

16ビットフリーランタイマのブロックダイヤグラム

Figure 14-2. 16ビットフリーランタイマのブロックダイヤグラム



(注意事項)上の図はすべてのタイマに有効です。

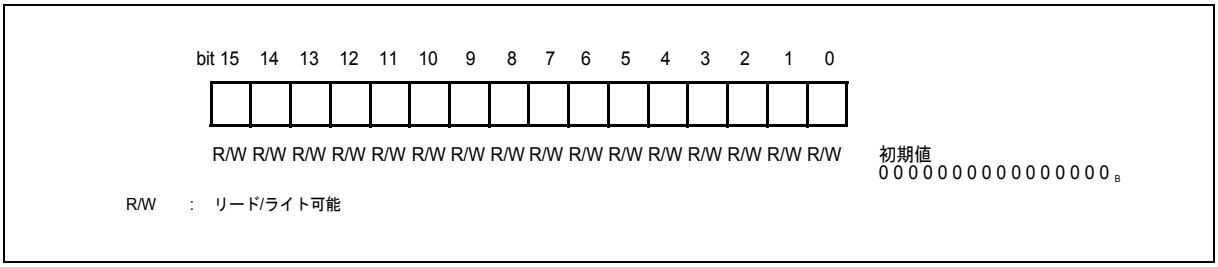
ICU と OCU の接続については「第1章　概要」を参照してください。

14.3.1 データレジスタ(TCDTn)

データレジスタ (TCDTn) は, 16 ビットフリーランタイマのカウント値を読み出すことができます。カウンタ値は, リセット時に 0000_H にクリアされます。このレジスタへの値の書込みによってタイマ値を設定できますが, 必ず停止(STOP=1)状態で書き込んでください。
データレジスタにはワードアクセスしてください。

フリーランタイマのデータレジスタ(TCDTn)

Figure 14-3. フリーランタイマのデータレジスタ(TCDTn)



16 ビットフリーランタイマは , 以下のときに初期化されます。

- リセット
- コントロールステータスレジスタ(TCCSL)のクリアビット (CLR)
- コンペアレジスタとタイマカウンタ値の一致

14.3.2 コントロールステータスレジスタ(TCCSLn)

コントロールステータスレジスタ (TCCSLn) は, 16 ビットフリーランタイマの動作モードを設定し, 16 ビットフリーランタイマを起動および停止, 割込みを制御します。

フリーランタイマのコントロールステータスレジスタ(TCCSLn)

Figure 14-4. フリーランタイマのコントロールステータスレジスタ(TCCSLn)

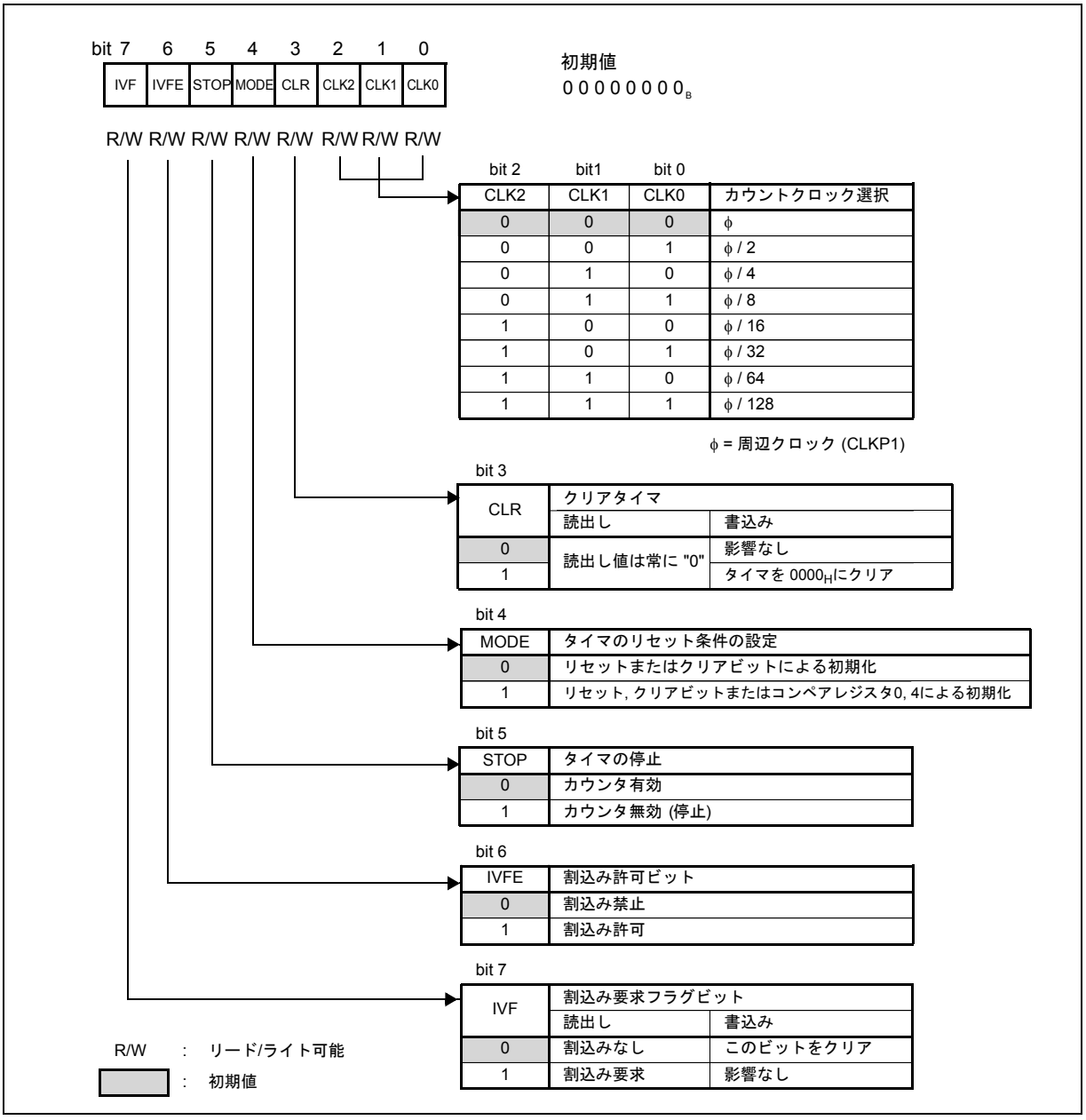


Table 14-1. フリーランタイマのコントロールステータスレジスタ(TCCSLn)

<table>
<tr><th colspan="2">ビット名</th><th>機能</th></tr>
<tr><td>bit 7</td><td>IVF:
割込み要求
フラグビット</td><td>• このビットは , フリーランタイマの割込み要求フラグです。
• "0" を書き込んだ場合 : フラグクリア
• "1" を書き込んだ場合 : 影響ありません。
• リードモディファイライト (RMW) 系命令では常に "1" が読み出されます。
• フリーランタイマがオーバフローを起こしたとき,またはMODEが"1"に設定されている状態でフリーランタイマとコンペアレジスタ0か4と一致した場合にこのビットは"1"にセットされます。</td></tr>
<tr><td>bit 6</td><td>IVFE:
割込み許可ビット</td><td>• このビットは , フリーランタイマの割込み許可ビットです。
• "0" を書き込んだ場合 : 割込み禁止。
• "1" を書き込んだ場合:割込み許可。</td></tr>
<tr><td>bit 5</td><td>STOP:
カウンタ許可ビット</td><td>• このビットは , フリーランタイマのカウント許可または禁止を設定します。
• "0" を書き込んだ場合 : カウント許可 (動作)
• "1" を書き込んだ場合:カウンタ禁止(停止)</td></tr>
<tr><td>bit 4</td><td>MODE:
リセット条件設定ビット</td><td>• "0": リセットまたはクリアビットによって初期化
• "1": リセット, クリアビット, コンペアレジスタ 0 またはコンペアレジスタ 4との一致 によって初期化</td></tr>
<tr><td>bit 3</td><td>CLR:
クリアビット</td><td>• CLR ビットは , 動作中のフリーランタイマを値 0000_H に初期化します。
• "0" を書き込んだ場合 : 影響ありません。
• "1" を書き込んだ場合 : カウンタが初期化されます。
(注意事項)
タイマが停止している (STOP = 1)ときカウンタ値を初期化するには, データレジスタ (TCDT) に 0000_H を書き込みます。</td></tr>
<tr><td>bit 2
～
bit 0</td><td>CLK2 ～ CLK0:
カウント
クロック選択</td><td>これらのビットは, 16 ビットフリーランタイマのカウントクロックを選択します。このビットに書き込み後,すぐにクロックが更新されますので,アウトプットコンペア,インプットキャプチャが停止状態のときに変更してください。
<table>
<tr><th>CLK2</th><th>CLK1</th><th>CLK0</th><th>カウントクロック</th><th>ϕ = 20 MHz</th><th>ϕ = 16 MHz</th><th>ϕ = 8 MHz</th><th>ϕ = 4 MHz</th><th>ϕ = 1 MHz</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>ϕ</td><td>50 ns</td><td>62.5 ns</td><td>125 ns</td><td>0.25 µs</td><td>1 µs</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>ϕ / 2</td><td>100 ns</td><td>125 ns</td><td>0.25 µs</td><td>0.5 µs</td><td>2 µs</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>ϕ / 4</td><td>0.2 µs</td><td>0.25 µs</td><td>0.5 µs</td><td>1 µs</td><td>4 µs</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>ϕ / 8</td><td>0.4 µs</td><td>0.5 µs</td><td>1 µs</td><td>2 µs</td><td>8 µs</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>ϕ / 16</td><td>0.8 µs</td><td>1 µs</td><td>2 µs</td><td>4 µs</td><td>16 µs</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>ϕ / 32</td><td>1.6 µs</td><td>2 µs</td><td>4 µs</td><td>8 µs</td><td>32 µs</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>ϕ / 64</td><td>3.2 µs</td><td>4 µs</td><td>8 µs</td><td>16 µs</td><td>64 µs</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>ϕ / 128</td><td>6.4 µs</td><td>8 µs</td><td>16 µs</td><td>32 µs</td><td>128 µs</td></tr>
</table></td></tr>
</table>

フリーランタイマのコントロールステータスレジスタ(TCCSHn)

Figure 14-5. フリーランタイマのコントロールステータスレジスタ(TCCSHn)

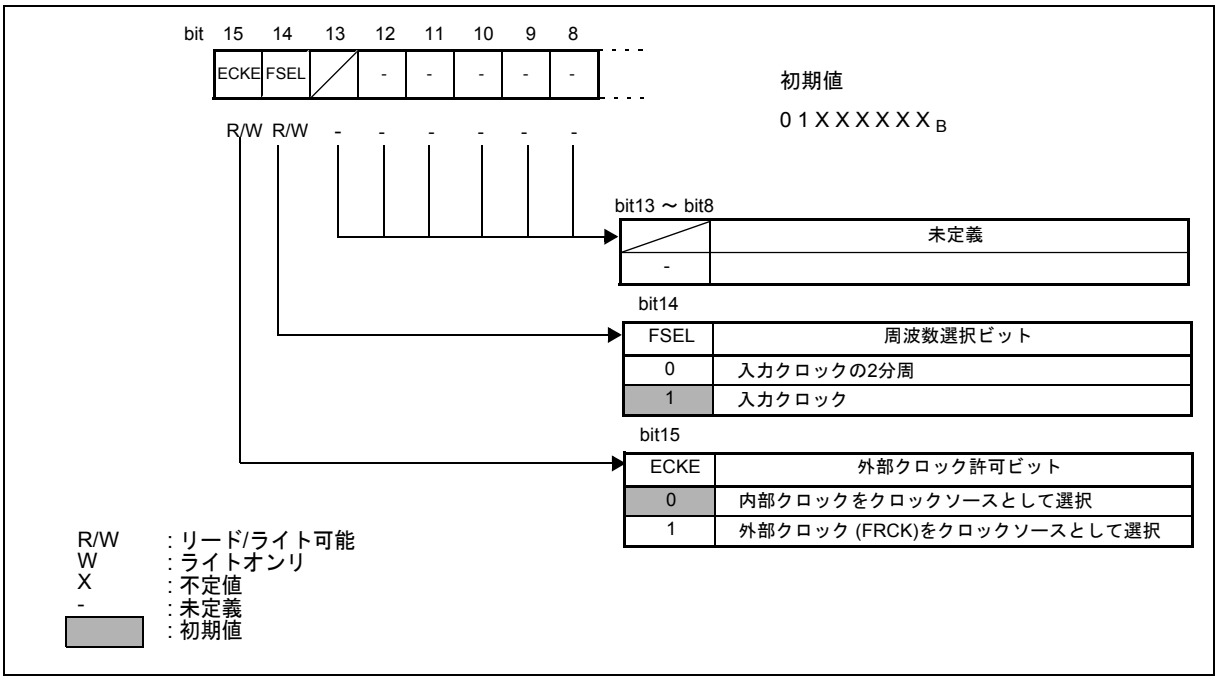


Table 14-2. フリーランタイマのコントロールステータスレジスタ(TCCSHn)の各ビットの機能説明

ビット名		機能
bit 15	ECKE: 外部 クロック許可 ビット	このビットは , 内部または外部クロックソースを選択します。このビットの変更はアウトプットコンペアまたはインプットキャプチャが停止しているときに行ってください。このビットに書き込んだ後にクロックが変更される可能性があります。 (注意事項) 内部クロックを選択した場合 , CLK2 ～ CLK0 ビット をカウントクロック値に変更されます。このカウントクロックが基本クロックとして使用されます。 クロックソースが FRCK からの入力である場合, 対応するポートのデータ方向レジスタ (DDR) を"入力" に設定し, 入力許可レジスタ (PIER) を "許可" に設定します。
bit 14	FSEL: 周波数 選択 ビット	このビットは , カウンタクロックソースの分周を制御するために使用します。このビットステータスの変更は , このビットの変更は , アウトプットコンペアとインプットキャプチャが停止しているときに行ってください。FSEL ビットは , カウンタクロック分周比を指定するために使用します。 FSEL が "0" にセットされると, カウンタクロック選択ビット (TCCSL: CLK2 ～ CLK0ビット) によって指定されたカウントクロックサイクルが 2 分周されます。
bit 13 ～ bit 8	未定義	これらのビットの読出し値は未定義です。 これらのビットは常に "0" を書き込んでください。 リードモディファイライト系命令は無効です。

14.3.3 16ビットフリーランタイマの動作

16 ビットフリーランタイマは, リセットが解除された後, カウンタ値 0000_H からカウントを開始します。このカウンタ値は, 16 ビットアウトプットコンペアと 16 ビットインプットキャプチャの動作の基準時間として使用されます。

14.3.3.1 16ビットフリーランタイマの動作

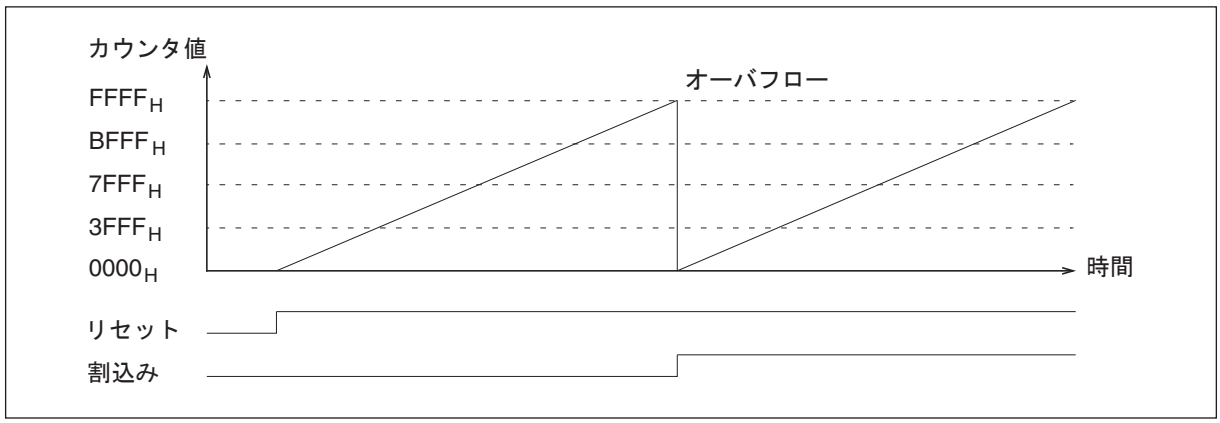
以下の条件でカウンタ値がクリアされます。

- オーバフローが発生したとき
- アウトプットコンペアレジスタ 0 またはアウトプットコンペアレジスタ 4 との一致が起こった(モード設定が必要)。
- 動作時に TCCSL レジスタの CLR ビットに "1" が書き込まれたとき
- 停止時に TCDT レジスタに 0000_H が書き込まれたとき
- リセット

オーバフローが発生した場合, またはコンペアレジスタ 0 または 4 との一致によってカウンタがクリアされた場合に割込みが生成されます (コンペア一致割込みは , 対応するモードでのみ使用できます)。

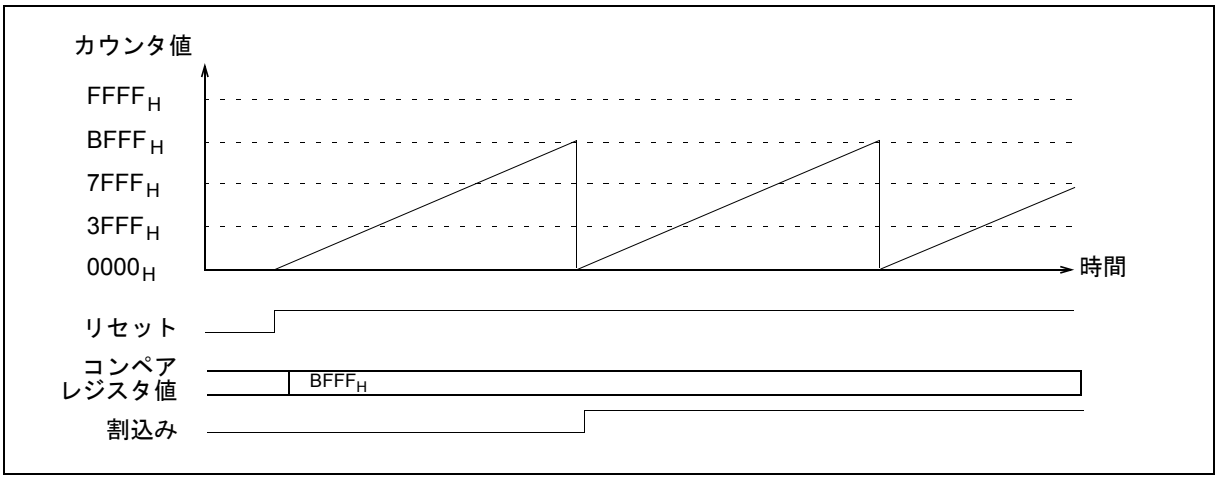
14.3.3.2 オーバフローによるカウンタのクリア

Figure 14-6. オーバフローによるカウンタのクリア



14.3.3.3 コンペアレジスタ値一致時のカウンタのクリア

Figure 14-7. コンペアレジスタ値が16ビットフリーランタイマ値と一致する場合のカウンタのクリア

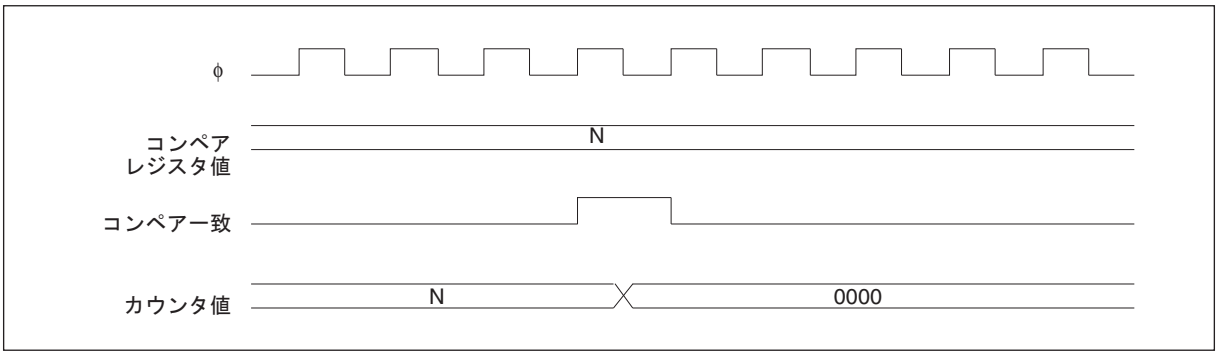


14.3.3.4 16ビットフリーランタイマタイミング

16ビットフリーランタイマクリアタイミング(コンペアレジスタ0か4と一致)

リセット時 , ソフトウェアクリア時 , またはコンペアレジスタ 0 か 4 との一致時にカウンタをクリアできます。リセットでは , カウンタは即座にクリアされます。コンペアレジスタ 0 またはコンペアレジスタ 4 値との一致時 , またはソフトウェアクリア時 (TCCSL: CLR = 1) には , カウンタはカウントタイミングと同時にクリアされます。

Figure 14-8. 16ビットフリーランタイマクリアタイミング(コンペアレジスタ0か4と一致)



14.4 アウトプットコンペアユニット

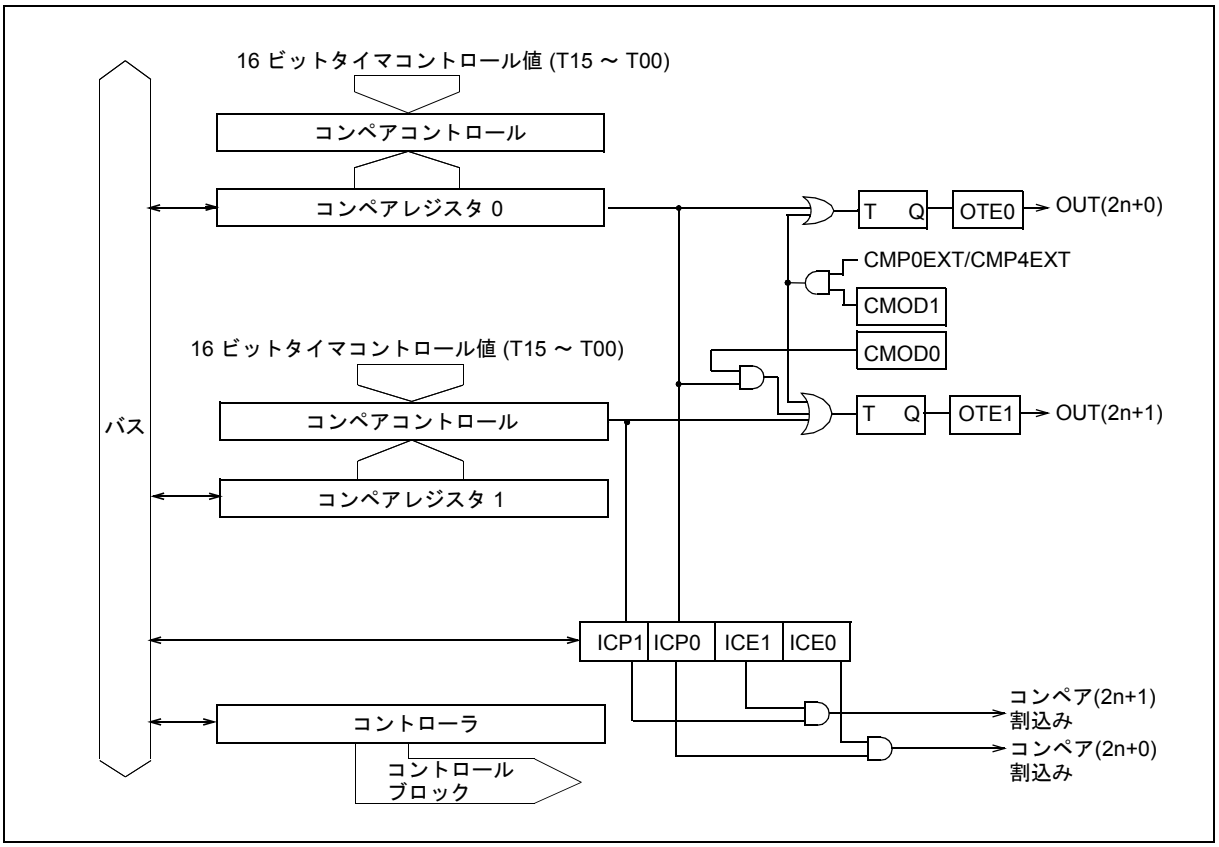
アウトプットコンペアユニット (OCU) は, 2本の16 ビットコンペアレジスタ, 2本のコンペアアウトプット端子, および 1 本のコントロールレジスタで構成されています。アウトプットコンペアは,設定されたコンペアレジスタ値とフリーランタイマとの値を比較して,一致していた場合に割込み要求を発生するとともに,出力レベルを反転します。

アウトプットコンペアユニット(OCU)の特長

- 各アウトプットコンペアユニットには, 2 つのコンペアレジスタがあり,独立して使用できます。モード設定によっては, 2 つのコンペアレジスタを使用して端子の出力を制御することが使用できます。
- 各端子出力の初期値を別々に指定できます。
- 比較の結果一致している時, 割込みを発生できます。
- 各OCUは,2本のコンペアレジスタを使って1つのPWM信号を生成できます(OUT0およびOUT4端子は除く)。
- 1本のアウトプットコンペアと3本のコンペアレジスタにより2つのPWM信号が生成できます。
- コンペアレジスタを使用して 1 つのサイクル波形信号を生成できます。

アウトプットコンペアユニットのブロックダイヤグラム

Figure 14-9. アウトプットコンペアユニットのブロックダイヤグラム

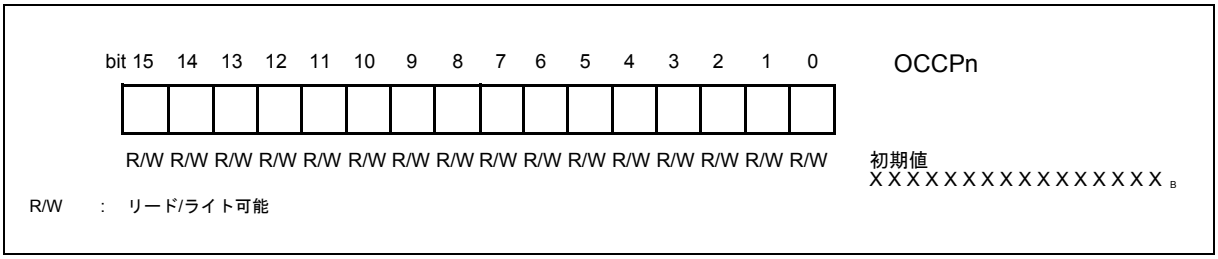


14.4.1 アウトプットコンペアレジスタ(OCCP(2n) / OCCP(2n+1))

16 ビットアウトプットコンペアレジスタは, 16 ビットフリーランタイマと比較されます。レジスタの初期値が未定義なので, 動作を有効にする前に適切な値を設定します。これらのレジスタにはワードアクセス命令によってアクセスする必要があります。レジスタの値が 16 ビットフリーランタイマの値と一致する場合, コンペア信号が生成され, アウトプットコンペア割込みフラグがセットされます。出力が有効である場合, コンペアレジスタに対応する出力レベルが反転されます。
コンペア一致がこのレジスタへの書込み動作と同時に発生した場合のコンペア動作は予測不能です。そのため, このレジスタへの書込みを行う前に, 必ずフリーランタイマのカウンタ値を確認するか, またはタイマを初期化します。

アウトプットコンペアレジスタ(OCCP(2n) / OCCP(2n+1))

Figure 14-10. アウトプットコンペアレジスタ(OCCP(2n) / OCCP(2n+1))



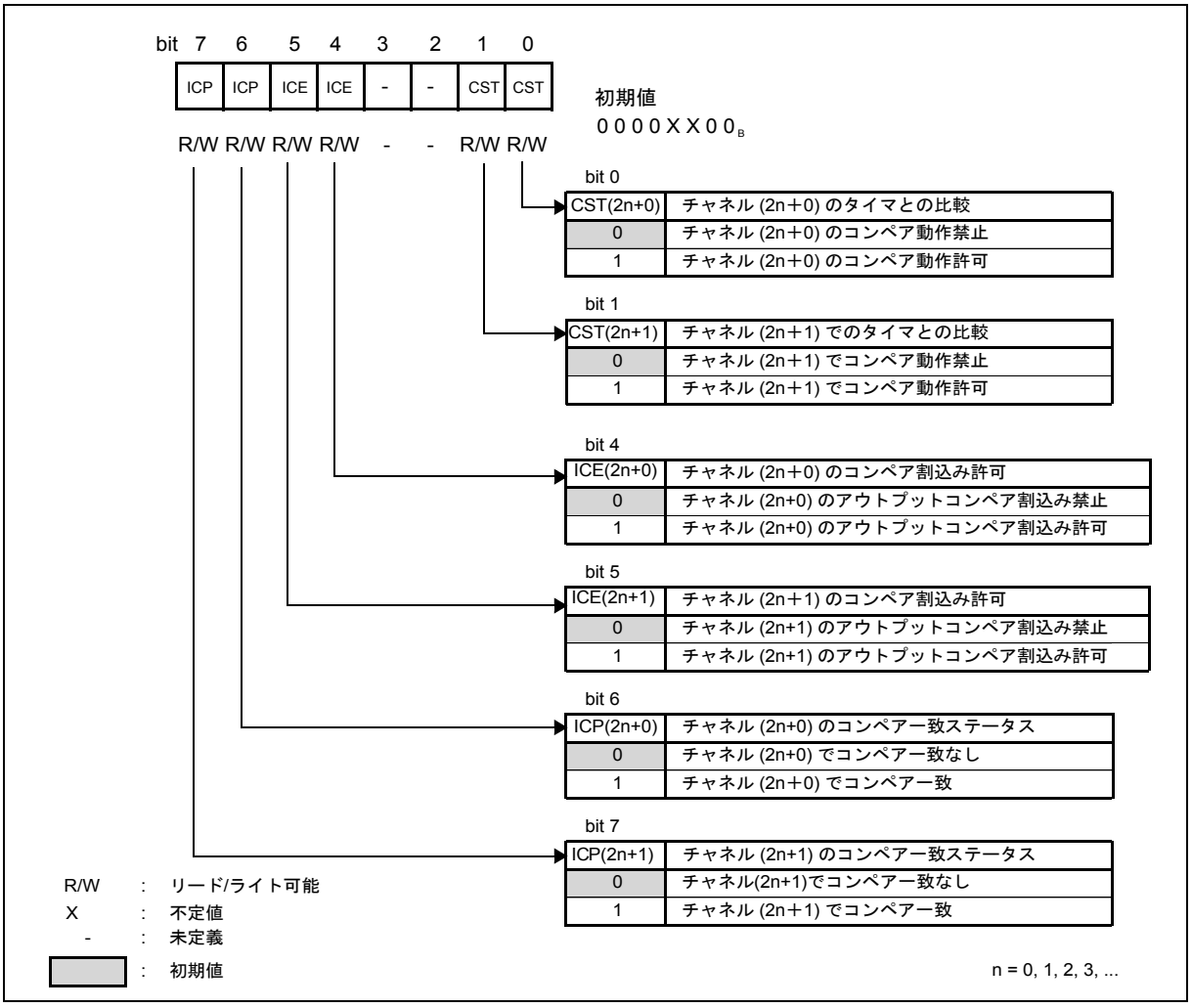
このレジスタは , 16 ビットワードとしてのみアクセスできます。

14.4.2 アウトプットコンペアのコントロールステータスレジスタ(OCS(2n) / OCS(2n+1))

コントロールステータスレジスタ (OCS(2n) / OCS(2n+1)) は, アウトプットコンペアの動作モードを設定し, アウトプットコンペアを開始および停止し, 割込みを制御し, 外部出力端子を設定します。

アウトプットコンペアコントロールステータスレジスタ(OCS(2n))

Figure 14-11. アウトプットコンペアコントロールステータスレジスタ(OCS(2n))



＜注意事項＞

サフィックスの "n" は , アウトプットコンペアユニットの番号 (0, 1, 2, ...) を表しています。レジスタ名およびビット名はレジスタ / ビットタイプ名とサフィックスで構成されます。

例 :

- アウトプットコンペアユニット 0 の場合 : n = 0, したがって OCS0 はビット CST0, CST1, ICE0, ICE1, ICP0, ICP1 を含みます。
- アウトプットコンペアユニット 1 の場合 : n = 1, したがって OCS2 はビット CST2, CST3, ICE2, ICE3, ICP2, ICP3 などを含みます。

Table 14-3. アウトプットコンペア制御ステータスレジスタ(OCS(2n))

ビット名		機能
bit 7	ICP(2n+1)	• これらのビットは , コンペア一致ステータスフラグとして使用されます。コンペアレジスタ値が 16 ビットフリーランタイマ値と一致する場合 , このビットは "1" にセットされます。 • コンペア一致ステータスビット ICP(2n+1) または ICP(2n+0) ビットがセットされ, 対応する割込み許可ビット ICE(2n+1) または ICE(2n+0) が許可されている場合 , アウトプットコンペア割込みが発生します。 • これらのビットは "0" の書込みによってクリアされます。 • "0": コンペア一致なし。 • "1": コンペア一致。 • "1" を書き込んでも影響はありません。 • リードモディファイライト系命令によって常に "1" が読み出されます。
bit 6	ICP(2n+0)	
bit 5	ICE(2n+1)	• これらのビットは , アウトプットコンペア割込み許可フラグとして使用されます。 • コンペア一致ステータスビット ICP(2n+1) または ICP(2n+0) ビットがセットされ, 対応する割込み許可ビット ICE(2n+1) または ICE(2n+0) が許可されている場合 , アウトプットコンペア割込みが発生します。 • "0" を書き込んだ場合 : アウトプットコンペア割込みが禁止されます。 • "1" を書き込んだ場合:アウトプットコンペア割込みが許可されます。
bit 4	ICE(2n+0)	
bit 3～bit 2	未定義	• これらのビットには常に "0" を書き込んでください。 • 読出し値は未定です。 • リードモディファイライト系命令は無効です。
bit 1	CST(2n+1)	• これらのビットはコンペア動作を許可するために使用します。 • "0" を書き込んだ場合 : コンペア動作が禁止されます。 • "1" を書き込んだ場合 : コンペア動作が許可されます。 (注意事項) コンペア動作を許可する前に, コンペアレジスタに値が書き込まれていることを確認してください。アウトプットコンペアは 16 ビットフリーランタイマクロックと同期化されていますから, 16 ビットフリーランタイマを停止するとコンペア動作が停止します。
bit 0	CST(2n+0)	

n = 0, 1, 2, 3, ... は , アウトプットコンペアユニットの番号を表しています。ビット名はビットタイプ名とサフィックスで構成されます。

例 : n = 0 の場合 : OCS0 で CST(2n+0) = CST0, CST(2n+1) = CST1

n = 1 の場合 : OCS2 で CST(2n+0) = CST2, CST(2n+1) = CST3

アウトプットコンペア制御ステータスレジスタ(OCS(2n+1))

Figure 14-12. アウトプットコンペア制御ステータスレジスタ(OCS(2n+1))

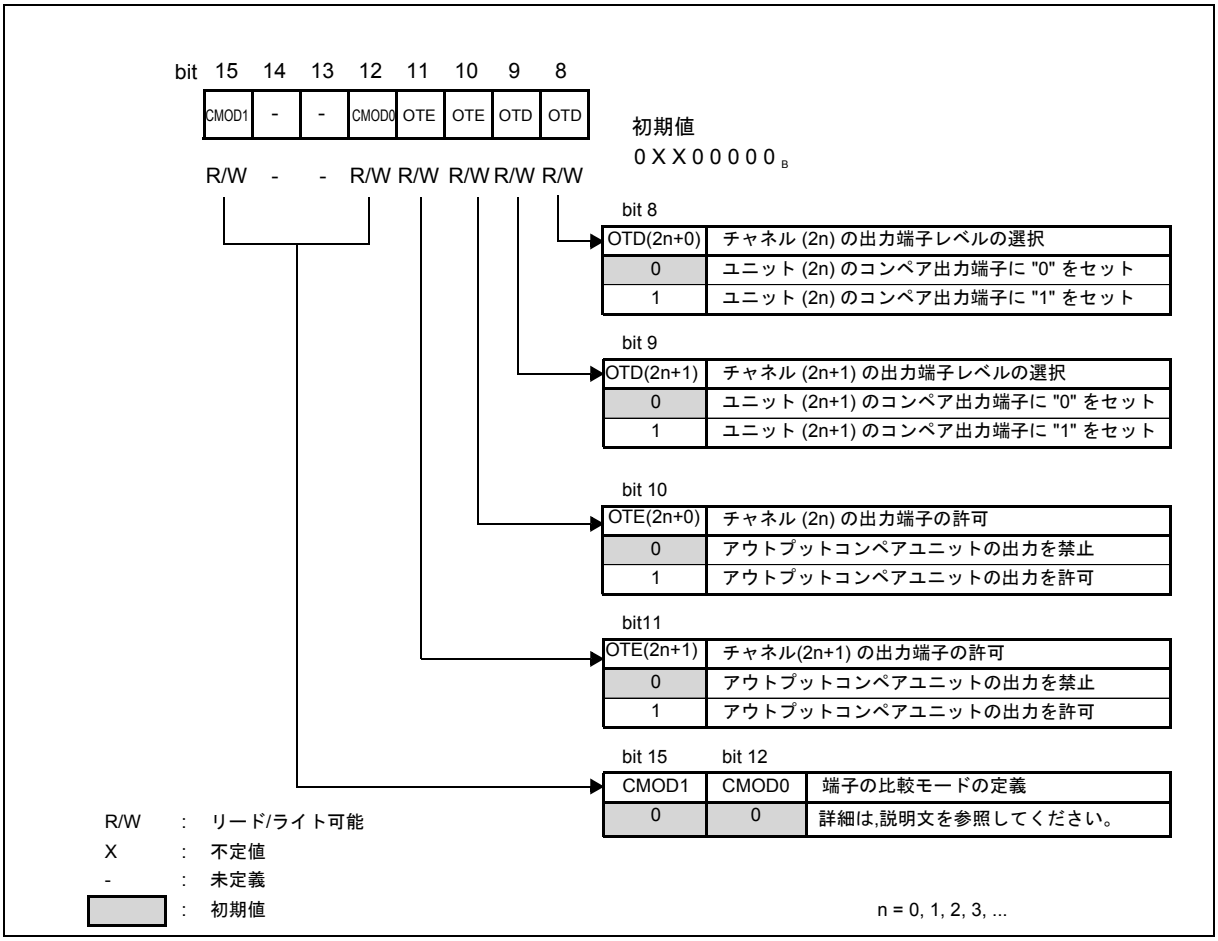


Table 14-4. アウトプットコンペア制御ステータスレジスタ(OCS(2n+1)) (Sheet 1 of 2)

ビット名		機能
bit 15, 12	CMOD0, CMOD1	これらのビットは端子出力信号の動作モードを定義します。定義されたモードによって, 種々のコンペアレジスタとの一致時にレベルがトグルされます。 Table 14-5 および「14.4.3 16ビットアウトプットコンペア動作」 を参照してください。
bit 14, 13	未定義	• これらのビットには常に "0" を書き込んでください。 • 読出し値は不定です。 • リードモディファイライト系命令は無効です。

Table 14-4. アウトプットコンペア制御ステータスレジスタ(OCS(2n+1)) (Sheet 2 of 2)

ビット名		機能
bit 11	OTE(2n+1)	これらのビットは , アウトプットコンペアユニット出力端子を許可するために使用されます。これらのビットの初期値は , "0" です。 • "0": アウトプットコンペアユニットの出力を禁止。対応する端子を汎用ポートとして , またはほかの周辺機能として使用できます。 • "1": アウトプットコンペアユニットの出力を許可。
bit 10	OTE(2n+0)	
bit 9	OTD(2n+1)	これらのビットは , アウトプットコンペアユニット出力端子が許可の時に端子出力レベルを変更するために使用されます。アウトプットコンペアユニット出力端子の初期値は , "0" です。値を書き込む前に , コンペア動作が停止されていることを確認してください。読出し時に , これらのビットはアウトプットコンペアユニットへの端子出力値を示します。 • "0" を書き込んだ場合 : アウトプットコンペアユニット出力端子に "0" をセットします。 • "1" を書き込んだ場合:アウトプットコンペアユニット出力端子に "1" をセットします。
bit 8	OTD(2n+0)	

n = 0, 1, 2, 3, ... は , アウトプットコンペアユニットの番号を表しています。ビット名はビットタイプ名とサフィックスで構成されます。
例 : n=0 の場合 : OCS(2n+1) = OCS1 のビット OTD(2n+0) = OTD0, OTD(2n+1) = OTD1
n=1 の場合 : OCS(2n+1) = OCS3 のビット OTD(2n+0) = OTD2, OTD(2n+1) = OTD3

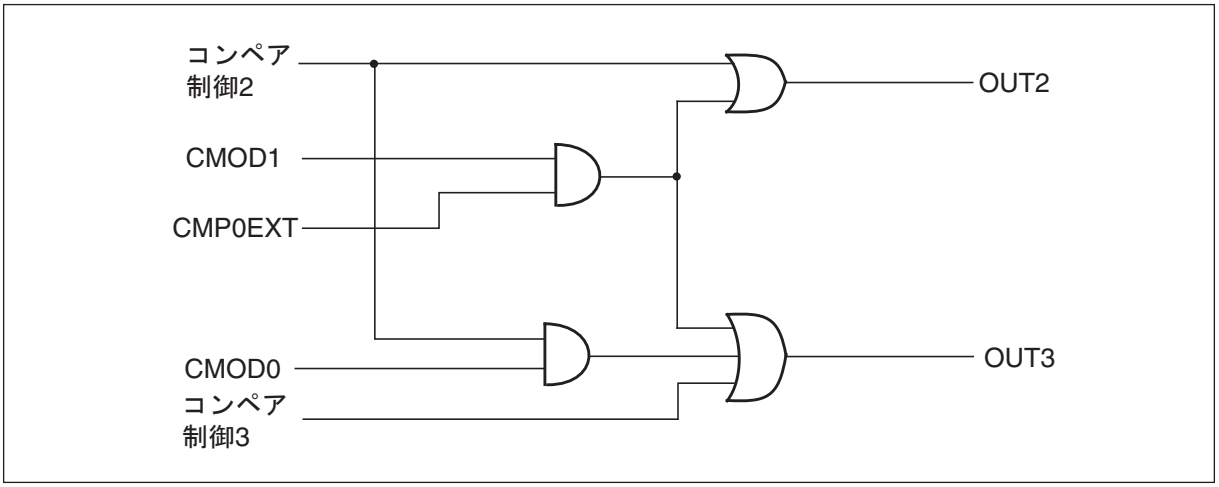
Table 14-5. CMOD1およびCMOD0ビットの機能

端子出力値は, コンペアレジスタ番号と一致したときに反転します。
表の読み方:

OCS(2n+1)		レジスタ OCCPx	
CMOD1	CMOD0	出力端子OUT(2n+0)	出力端子OUT(2n+1)
0/1/x	0/1/x	リストにある OCCP 番号のレジスタの値と一致したときに OUT(2n+0) がトグルします。	リストにあるOCCP番号のレジスタの値と一致したときに OUT(2n+1) がトグルします。
0/1/x	0/1/x	リストにあるOCCP番号のレジスタの値と一致したときに OUT(2n+0) がトグルします。	リストにあるOCCP番号のレジスタの値と一致したときに OUT(2n+1) がトグルします。

OCS1		OCCPxの一致時にトグル	
CMOD1	CMOD0	OUT0	OUT1
x	0	x = 0	x = 1
x	1	x = 0	x = 0, x = 1
OCS3		OCCPxの一致時にトグル	
CMOD1	CMOD0	OUT2	OUT3
0	0	x = 2	x = 3
0	1	x = 2	x = 2, x = 3
1	0	x = 0, x = 2	x = 0, x = 3
1	1	x = 0, x = 2	x = 0, x = 2, x = 3
OCS5		OCCPxの一致時にトグル	
CMOD1	CMOD0	OUT4	OUT5
x	0	x = 4	x = 5
x	1	x = 4	x = 4, x = 5
OCS7		OCCPxの一致時にトグル	
CMOD1	CMOD0	OUT6	OUT7
0	0	x = 6	x = 7
0	1	x = 6	x = 6, x = 7
1	0	x = 4, x = 6	x = 4,x = 7
1	1	x = 4, x = 6	x = 4, x = 6, x = 7
OCS9		OCCPxの一致時にトグル	
CMOD1	CMOD0	OUT8	OUT9
x	0	x = 8	x = 9
x	1	x = 8	x = 8, x = 9
OCS11		OCCPxの一致時にトグル	
CMOD1	CMOD0	OUT10	OUT11
x	0	x = 10	x = 11
x	1	x = 10	x = 10, x = 11

Figure 14-13. 出力選択(OCUモジュール1)のブロックダイヤグラム



アウトプットコンペアレジスタ 0 との一致が要求される OCU モジュール 1 では , CMOD[1:0] =10_B なら , モジュール 0 からの比較結果が CMP0EXT 信号として内部に送られます。これはモジュール 0 そのものには適用されません。この場合 OCCP0 および OCCP1 以外のレジスタを使用することは , できません。

同じ状況が OCU モジュール 3 にも当てはまります。この場合 CMP4EXT としてモジュール 2 からの結果が必要となります。

14.4.3 16ビットアウトプットコンペア動作

16 ビットアウトプットコンペア動作では，割込み要求フラグを設定することができ，指定したコンペアレジスタ値が16ビットフリーランタイマ値と一致するとき出力レベルをトグルさせることができます。CMOD0およびCMOD1 ビットを使って各端子に対応するコンペアレジスタを定義することができます。

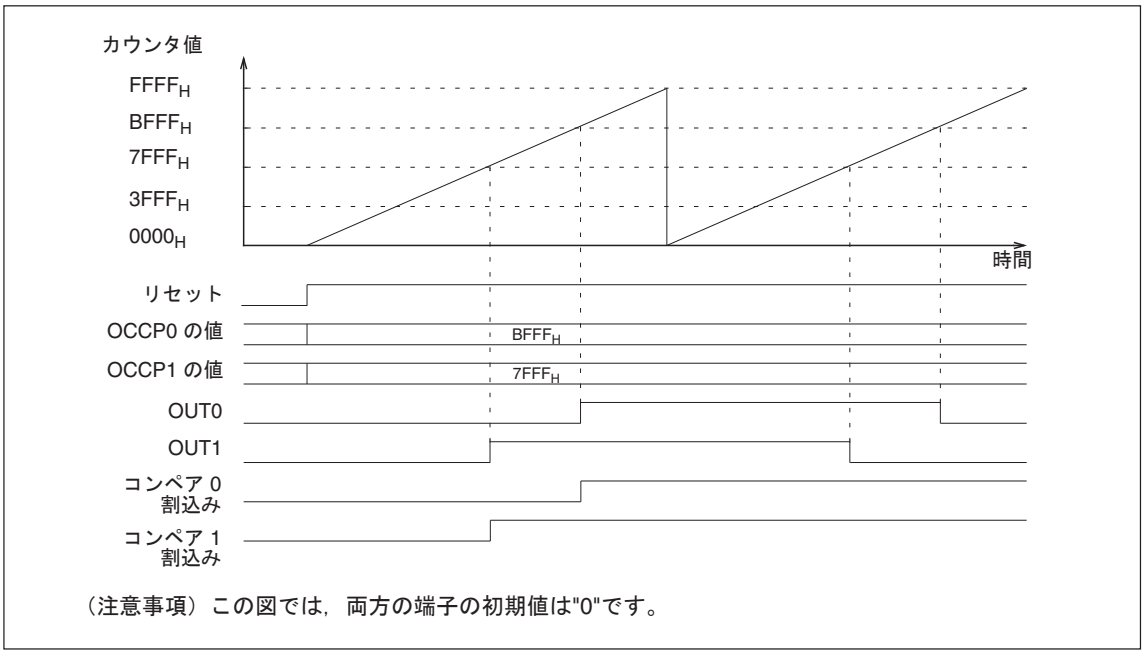
14.4.3.1 CMOD[1:0] = "00_B"の場合の出力波形の例

CMOD[1:0]=00_B であるとき , コンペアレジスタに対応する端子の出力レベルがレジスタ値と一致するたびに反転されます。各出力値は 1 つのコンペアレジスタによって制御されます。

OUT0: このレベルは , コンペアレジスタ 0 との一致によってのみ反転されます。

OUT1: このレベルは , コンペアレジスタ 1 との一致によってのみ反転されます。

Figure 14-14. CMOD[1:0] = "00_B"のときの出力波形の例



14.4.3.2 CMOD[1:0] = "01_B"の場合の2つのコンペアレジスタによる出力波形の例

CMOD[1:0]=01_B のとき , コンペアレジスタ 0 (2) に対応する端子の出力レベルがレジスタ値と一致するたびに反転します。これは CMOD[1:0] =00_B の場合の動作と同じです。ただし 2 番目の端子の出力レベルは , コンペアレジスタ 0 またはコンペアレジスタ 1 (3) のどちらかと一致したときに反転します。これによって , パルス信号を生成する際に , 一方のエッジをコンペアレジスタ 0 の値によって定義し , 他方のエッジをコンペアレジスタ 1 (3) によって定義する (またはその逆) ことができます。両方のコンペアレジスタが同じ値である場合 , 動作はCMOD[1:0]=00_B の場合と同じです。

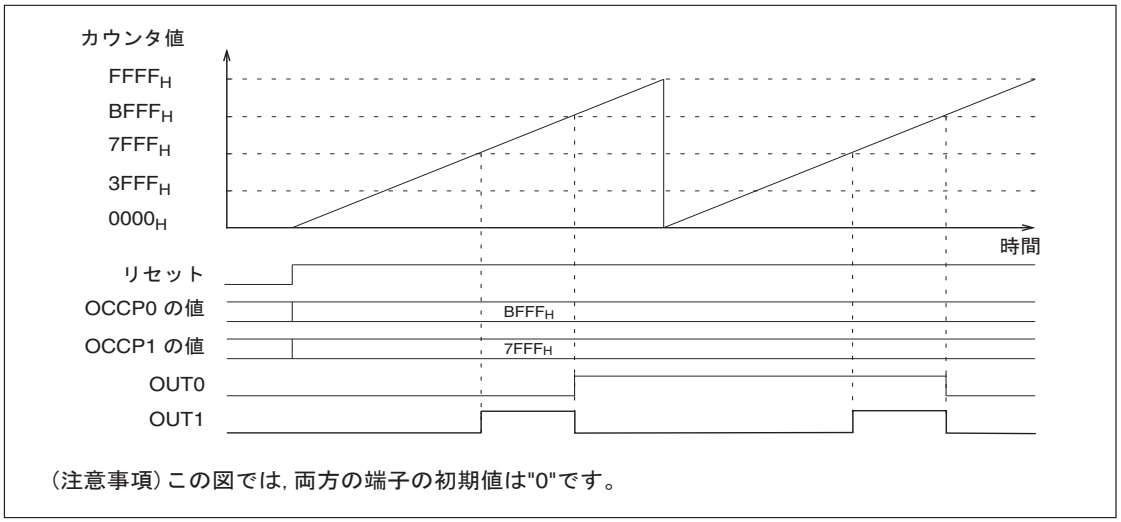
このモードと , フリーランタイマのコンペアレジスタ一致によるリセットのオプション (TCCSLx レジスタの MODE ビット) と合わせて使用することによって , 周波数が異なるパルス幅変調信号を生成できます。

OUT0 (2): このレベルは , コンペアレジスタ 0 (2) との一致によってのみ反転されます。

OUT1 (3): このレベルは , コンペアレジスタ 0 (2) またはコンペアレジスタ 1 (3) との一致によって反転されます。

OUT4, OUT5, OUT6, および OUT7 では , コンペアレジスタ 4 は , 上記のコンペアレジスタ 0 と同じ働きをします。

Figure 14-15. CMOD[1:0] = "01_B"のとき(一致によるタイマリセットなし)の出力波形の例



14.4.3.3 CMOD[1:0] = "10_B"の場合の出力波形の例

CMOD[1:0]=10_B によって設定される動作モードは , 各フリーランタイマに対して 3 つ のパルス幅変調信号 (2 つではなく) を使用することを想定しています。OCU モジュール 1 にこのモードがセットされている場合 , タイマ値とコンペアレジスタ 0 との一致によって OUT2 と OUT 3 の両方が反転します。3 番目のパルス信号では , OCU モジュール 0 の CMOD[1:0] を , 01_B に設定する必要があります。

レジスタ OCS1 の場合 : CMOD[1:0]=01_B

OUT0: このレベルは , コンペアレジスタ 0 との一致によってのみ反転されます。

OUT1: このレベルは , コンペアレジスタ 0 またはコンペアレジスタ 1 との一致によって反転します。

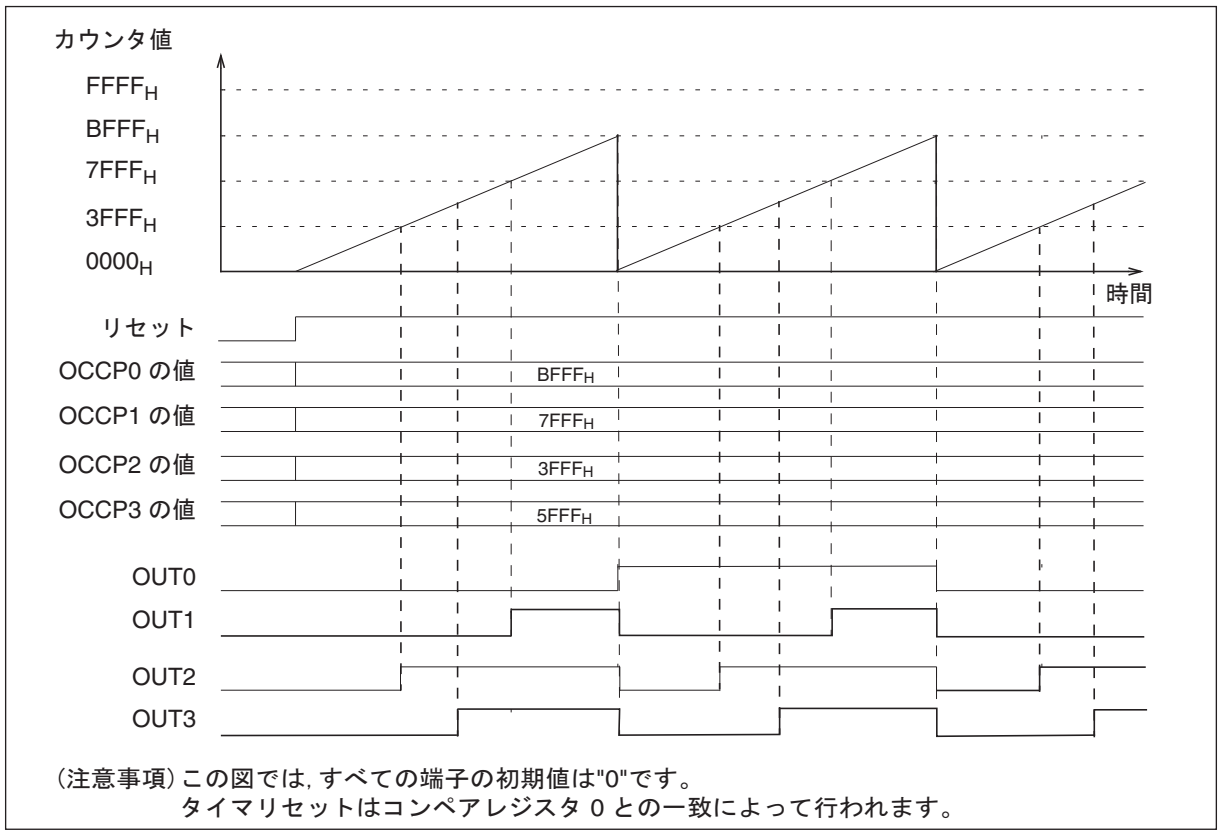
レジスタ OCS3 の場合 : CMOD[1:0] =10_B

OUT2: このレベルは , コンペアレジスタ 0 またはコンペアレジスタ 2 との一致によって反転します。

OUT3: このレベルは , コンペアレジスタ 0 またはコンペアレジスタ 3 との一致によって反転します。

OUT4, OUT5, OUT6, および OUT7 では , コンペアレジスタ 4 は , 上記のコンペアレジスタ 0 と同じ働きをします。

Figure 14-16. OCS1:CMOD[1:0] = "01_B"およびOCS3:CMOD[1:0] = "10_B"の場合の出力波形



14.4.3.4 CMOD[1:0] = "11_B"の場合の出力波形の例

CMOD[1:0]=11_B のとき , OUT3 (OUT7) 端子の出力レベルはコンペアレジスタ 0, 2, または 3 (4, 6, または 7) によって反転されます。端子 OUT1 (OUT5) では , この設定は CMOD[1:0]=01_B と同じです (表 14.4-3 も参照してください)。

OUT0: このレベルは , コンペアレジスタ 0 との一致によってのみ反転します。

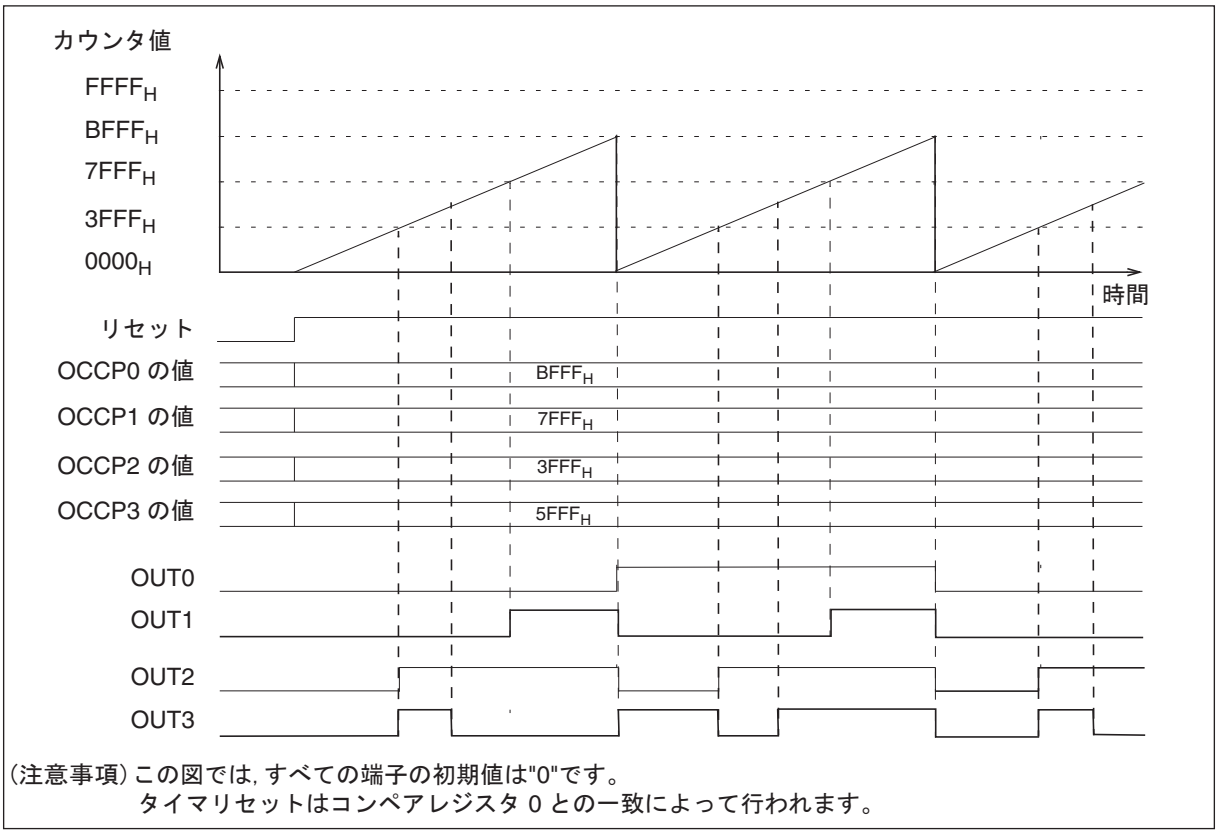
OUT1: このレベルは , コンペアレジスタ 0 またはコンペアレジスタ 1 との一致によって反転します。

OUT2: このレベルは , コンペアレジスタ 0 またはコンペアレジスタ 2 との一致によって反転します。

OUT3: このレベルは , コンペアレジスタ 0, コンペアレジスタ 2 またはコンペアレジスタ 3 との一致によって反転されます。

OUT4, OUT5, OUT6, および OUT7 では , コンペアレジスタ 4 は , 上記のコンペアレジスタ 0 と同じ働きをします。

Figure 14-17. OCS1:CMOD[1:0] = "11_B"およびOCS3:CMOD[1:0] = "11_B"の場合の出力波形



(注意事項) この図では, すべての端子の初期値は"0"です。
タイマリセットはコンペアレジスタ 0 との一致によって行われます。

14.4.3.5 アウトプットコンペアタイミング

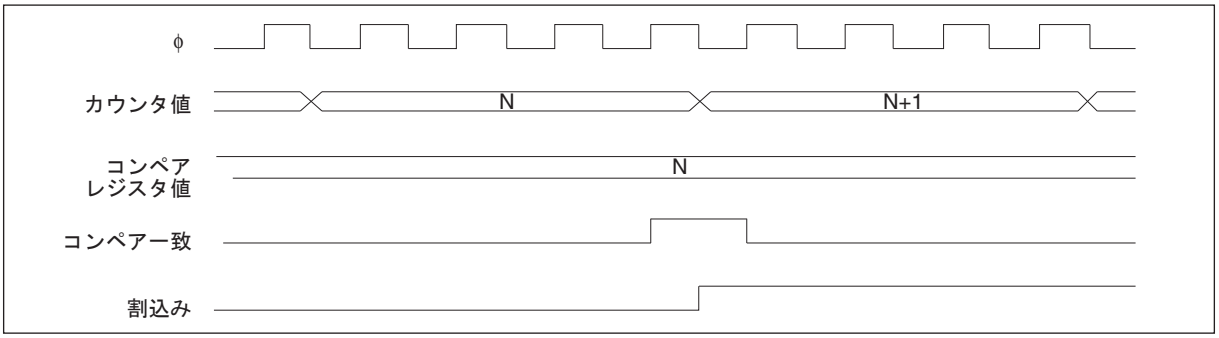
アウトプットコンペア動作では , フリーランタイマ値が指定されたコンペアレジスタ値と一致したとき , コンペア一致信号が生成されます。出力値を反転でき , 割込みを発行することができます。コンペア一致時の出力反転のタイミングは , カウンタのタイミングと同期化されます。

■ コンペアレジスタの更新時のコンペア動作

コンペアレジスタが更新されると , カウンタ値の比較は行いません。

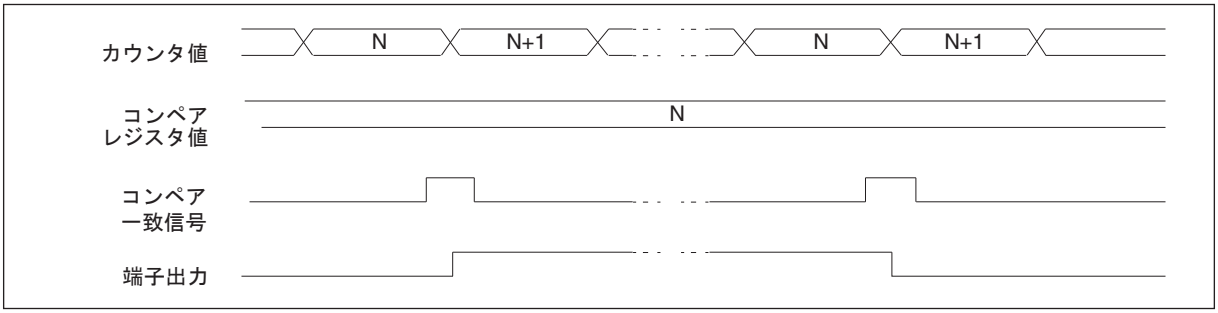
■ 割込みタイミング

Figure 14-18. 割込みタイミング



■ 出力端子変更タイミング

Figure 14-19. 出力端子変更タイミング



14.5 インプットキャプチャユニット

インプットキャプチャユニット (ICU) は, 外部入力信号の立上りエッジ, 立下りエッジ, または両方のエッジを検出し, その時点での 16 ビットフリーランタイマ値をレジスタに格納します。また, インプットキャプチャユニットは, エッジの検出時に割込みを生成することができます。1 つのインプットキャプチャユニットは, 2 つのインプットキャプチャレジスタと 1 つの制御レジスタで構成されます。

インプットキャプチャユニットの特長

- 外部入力の有効なエッジを以下の 3 つのタイプから選択できます。

Table 14-6. 外部入力エッジのタイプ

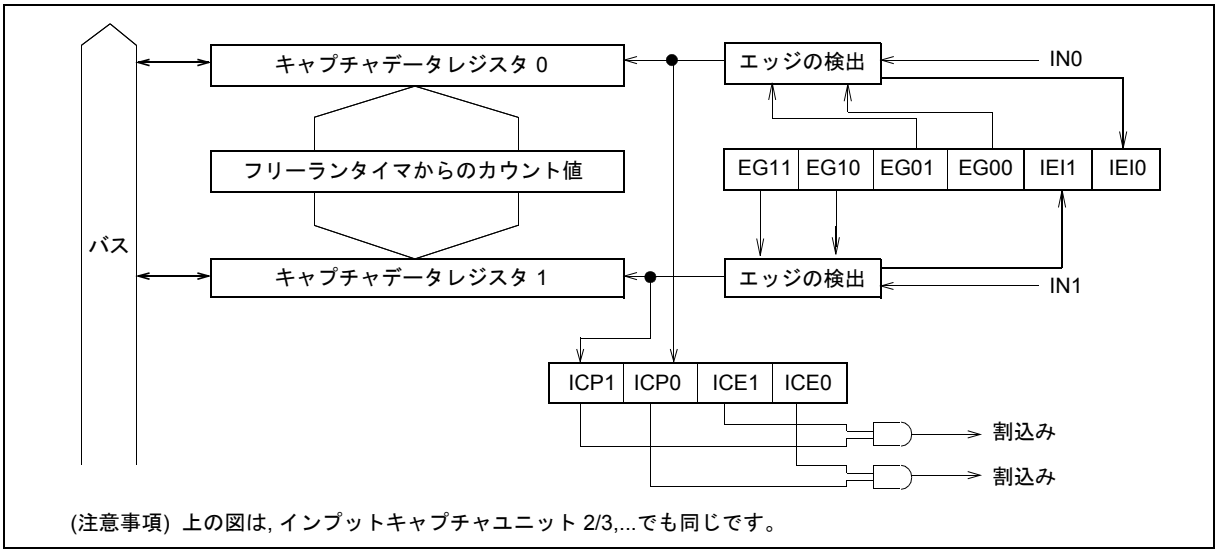
立ち上がりエッジ	↑
立下りエッジ	↓
両方のエッジ	↑↓

- 外部入力の有効なエッジの検出時に割込みを生成することができます。

インプットキャプチャユニットのブロックダイヤグラム

Figure 14-20 は , インプットキャプチャユニットのブロックダイヤグラムを示しています。

Figure 14-20. インプットキャプチャユニットのブロックダイヤグラム



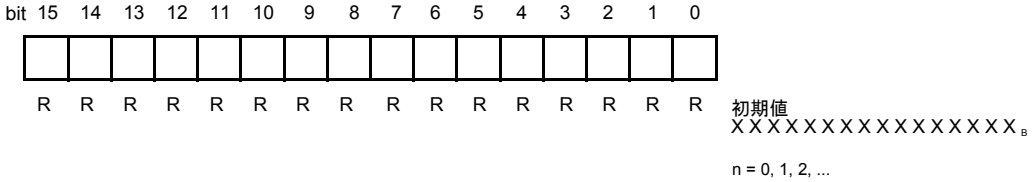
(注意事項) 上の図は, インプットキャプチャユニット 2/3,...でも同じです。

14.5.1 インプットキャプチャユニットレジスタの詳細

インプットキャプチャユニットは, 制御ステータスレジスタ (ICS(2n)(2n+1)) とエッジレジスタ (ICE(2n)(2n+1)) で構成されています。
インプットキャプチャユニットは, 16 ビットデータレジスタ (IPCPn) を備えています。このレジスタは, 対応する外部端子入力波形の有効なエッジが検出されたとき, 16 ビットフリーランタイマからの値を格納します。データの一貫性のため, このレジスタにはワードモードでのみアクセスしてください。

インプットキャプチャデータレジスタ(IPCPn)

Figure 14-21. インプットキャプチャデータレジスタ(IPCPn)

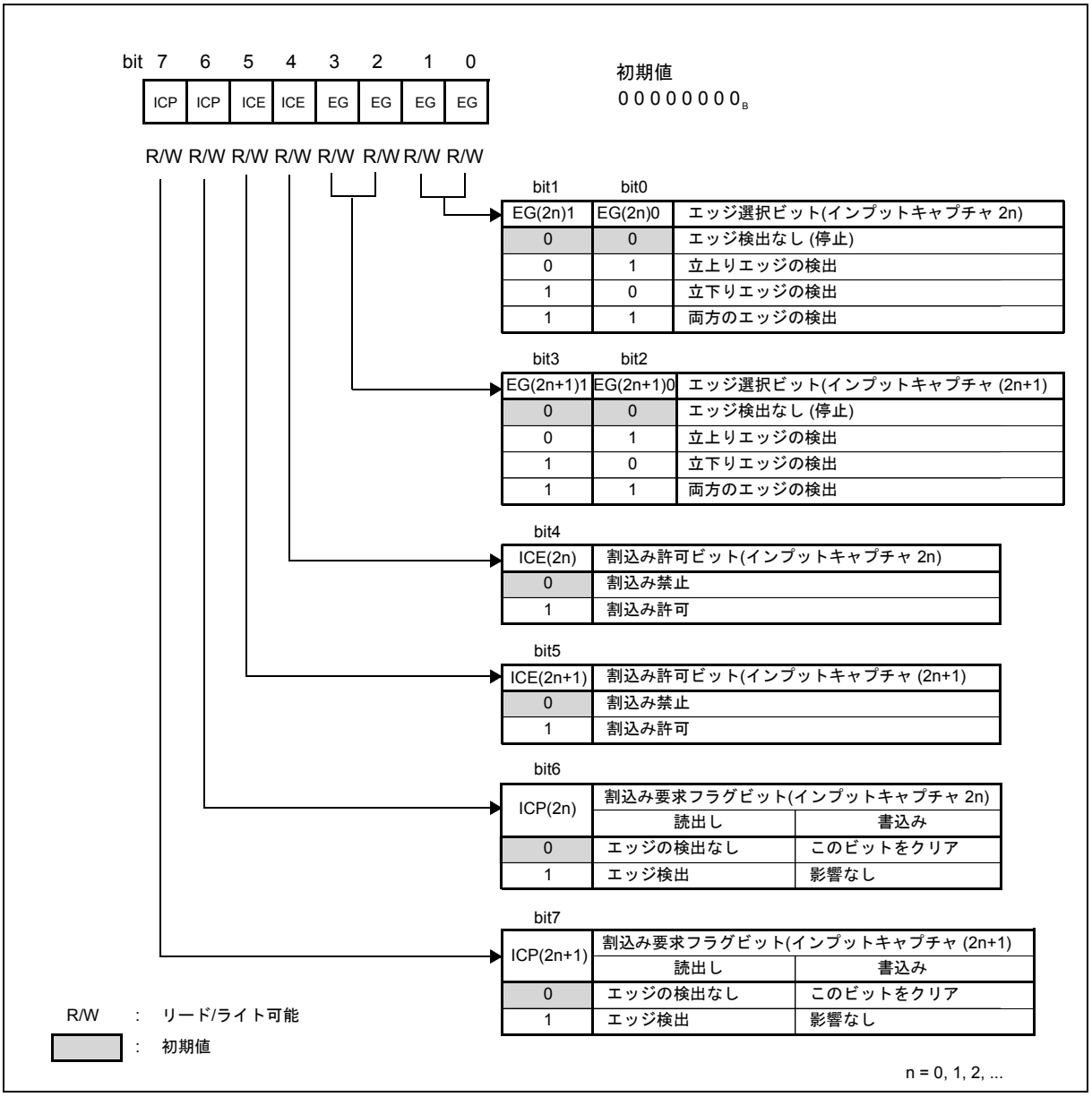


＜注意事項＞

上位バイトと下位バイトのデータの一貫性のため , このレジスタは常にワードモードで読み出します。

インプットキャプチャ制御ステータスレジスタ(ICS(2n)(2n+1))

Figure 14-22. インプットキャプチャ制御ステータスレジスタ(ICS(2n)(2n+1))



＜注意事項＞

サフィックスの "n" は , インプットキャプチャユニットの番号 (0, 1, 2, ...) を表しています。レジスタ名およびビット名はレジスタ / ビットタイプ名とサフィックスで構成されます。例 :

- インプットキャプチャユニット 0 の場合 : n = 0, したがって ICS01 はビット EG01, EG00, EG11, EG10, ICE0, ICE1, ICP0, ICP1 を含みます。
- インプットキャプチャユニット 1 の場合 : n = 1, したがって ICS23 はビット EG21, EG20, EG31, EG30, ICE2, ICE3, ICP2, ICP3 を含みます。
- 以下同様。

Table 14-7. インプットキャプチャユニット制御ステータスレジスタのビット

ビット名		機能
bit7	ICP(2n+1): 割込み要求フラグ ビット (インプットキャプチャ 2n+1)	• このビットは インプットキャプチャユニット n, 第 2 チャネルの割込み要求フラグとして使用されます。 • 外部入力端子の有効なエッジの検出時にこのビットに "1" がセットされます。 • 割込み許可ビット (ICE(2n+1)) がセットされている時 , 有効なエッジの検出時に割込みを生成します。 • "0" を書き込むとこのビットがクリアされます。 • "1" を書き込んでも影響はありません。 • リードモディファイライト系命令の場合は, 常に "1" が読み出されます。
bit6	ICP(2n): 割込み要求フラグ ビット (インプットキャプチャ 2n)	• このビットは インプットキャプチャユニット n, 第 1 チャネルの割込み要求フラグとして使用されます。 • 外部入力端子の有効なエッジの検出時にこのビットに "1" がセットされます。 • 割込み許可ビット (ICE(2n)) がセットされている時 , 有効なエッジの検出時に割込みを生成します。 • "0" を書き込むとこのビットがクリアされます。 • "1" を書き込んでも影響はありません。 • リードモディファイライト系命令の場合は, 常に "1" が読み出されます。
bit5	ICE(2n+1): 割込み要求許可 ビット (インプットキャプチャ 2n+1)	• このビットは , インプットキャプチャユニット n, 第 2 チャネルのインプットキャプチャ割込み要求を許可するために使用します。 • このビットに "1" を書き込んだ場合, 割込みフラグ (ICP(2n+1)) がセットされたときにインプットキャプチャ割込みが生成されます。
bit4	ICE(2n): 割込み要求許可 ビット (インプットキャプチャ 2n)	• このビットは , インプットキャプチャユニット n, 第 1 チャネルのインプットキャプチャ割込み要求を許可するために使用します。 • このビットに "1" を書き込んだ場合, 割込みフラグ (ICP(2n)) がセットされたときにインプットキャプチャ割込みが生成されます。
bit3, bit2	EG(2n+1)1, EG(2n+1)0 エッジ選択ビット	• これらのビットは , インプットキャプチャユニット n, 第 2 チャネルの外部入力の有効なエッジの極性を指定するために使用します。 • これらのビットは, インプットキャプチャ動作を有効にするためにも使用します。
bit1, bit0	EG(2n)1, EG(2n)0 エッジ選択ビット	• これらのビットは , インプットキャプチャユニット n, 第 1 チャネルの外部入力の有効なエッジの極性を指定するために使用します。 • これらのビットは, インプットキャプチャ動作を有効にするためにも使用します。

サフィックスの n= 0, 1, 2, 3, ... は , インプットキャプチャユニットの番号を表しています。ビット名はビットタイプ名とサフィックスで構成されます。したがって ,
n = 0 の場合 : ICP1, ICP0, ICE1, ICE0, EG11, EG10, EG01, EG00
n = 1 の場合 : ICP3, ICP2, ICE3, ICE2, EG31, EG30, EG21, EG20。以下同様。

インプットキャプチャユニットエッジレジスタ(ICE(2n)(2n+1))

インプットキャプチャユニットエッジレジスタ (ICE(2n)(2n+1)) にはデバイス依存構成ビットが含まれます。これらのレジスタの詳細については「1.6 LIN-USART のインプットキャプチャユニットのソースの選択」を参照してください。

Figure 14-23. インプットキャプチャユニットエッジレジスタ(ICE(2n)(2n+1))

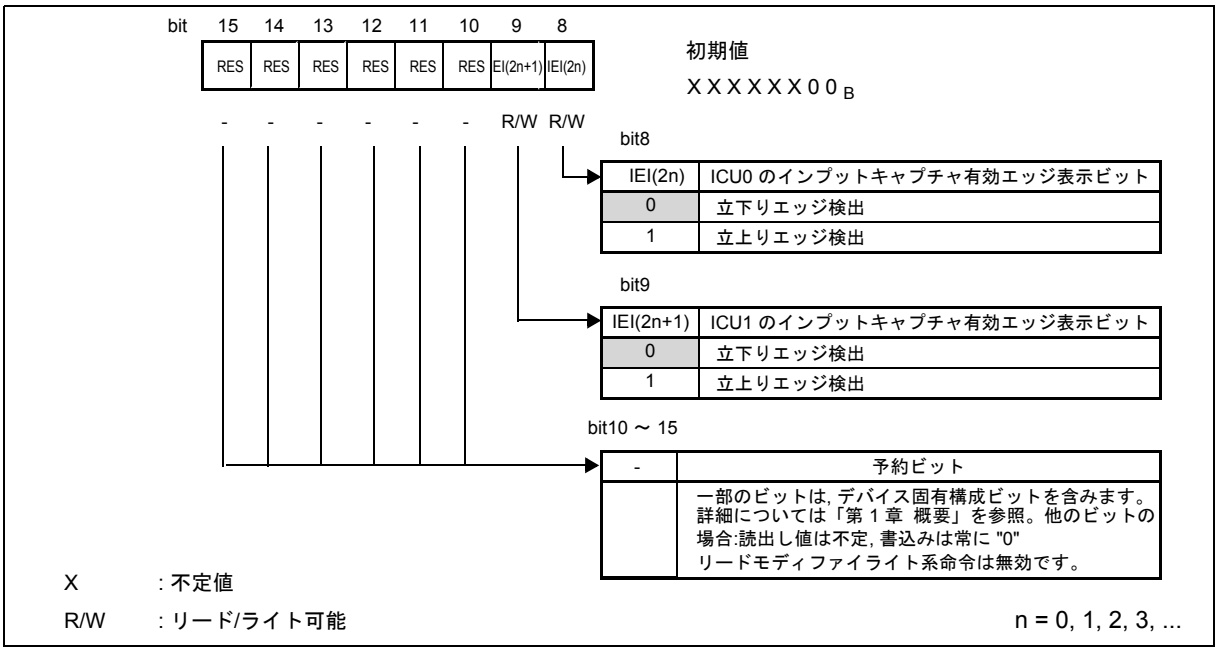


Table 14-8. インプットキャプチャユニットエッジレジスタビット(上位および下位)

ビット名		機能
bit15 ～ bit10	予約	一部のビットは, デバイス固有構成ビットを含みます。(詳細は「1.6 LIN-USARTのインプットキャプチャユニットのソースの選択」を参照)。
bit9	IEI(2n+1): 有効エッジ表示ビット	このビットは立上りまたは立下りエッジが検出されたことを示すためのキャプチャレジスタ IPCP(2n+1) のエッジ表示ビットです。 • "0": 立下りエッジ検出 • "1": 立上りエッジ検出 • このビットはリードオンリです。 (注意事項) EG(2n+1)1, EG(2n+1)0 =00 である場合は , 読出し値は無効です。 読出し値は, EG(2n+1)1 および EG(2n+1)0 によって選択したエッジに依存しません。
bit8	IEI(2n): 有効エッジ表示ビット	このビットは立上りまたは立下りエッジが検出されたことを示すためのキャプチャレジスタ IPCP0, IPCP2, IPCP4 および IPCP6 のエッジ表示ビットです。 • "0": 立下りエッジ検出 • "1": 立上りエッジ検出 • このビットはリードオンリです。 (注意事項) EG(2n)1, EG(2n)0 =00 である場合は , 読出し値は無効です。 読出し値は, EG(2n)1 および EG(2n)0 によって選択したエッジに依存しません。

サフィックスの n= 0, 1, 2, 3, ... は , インプットキャプチャユニットの番号を表しています。ビット名はビットタイプ名とサフィックスで構成されます。したがって ,
n = 0 の場合 : IEI1, IEI0
n = 1 の場合 : IEI3, IEI2。以下同様。

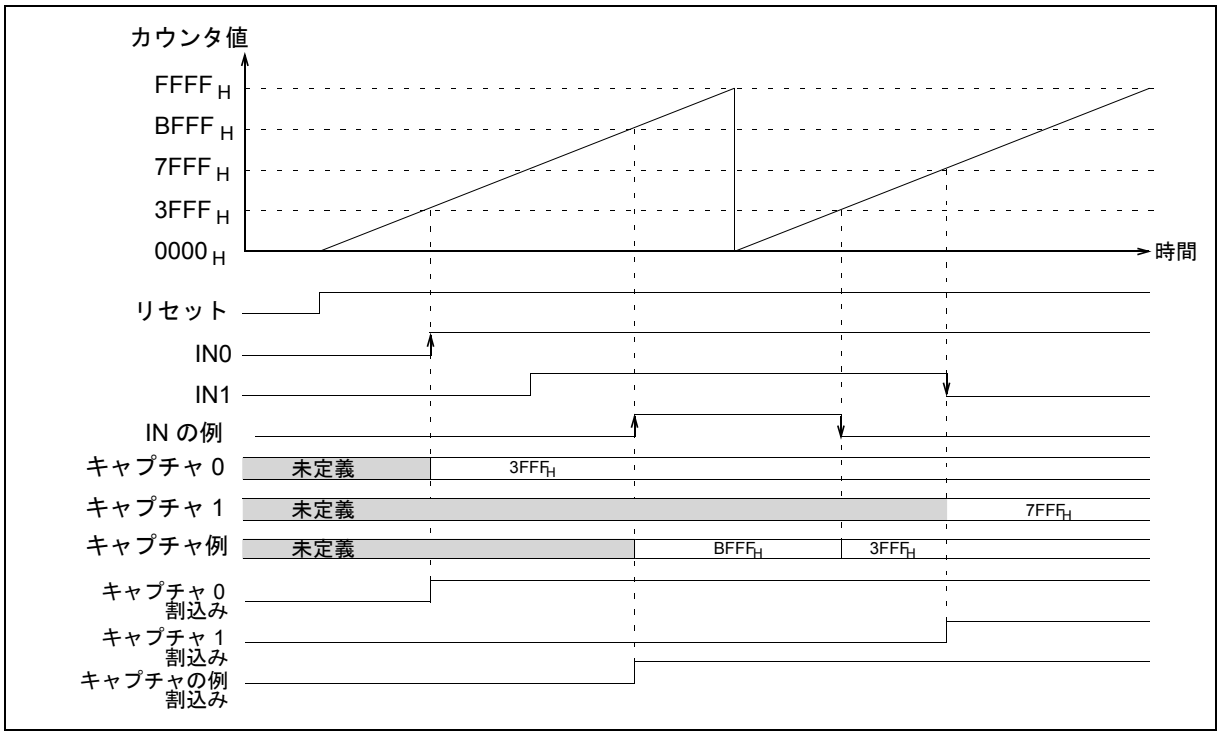
14.5.2 16ビットインプットキャプチャの動作

16 ビットインプットキャプチャの動作では, 指定したエッジの検出時に割込みを生成でき, 16 ビットフリーランタイマ値をフェッチし, それをキャプチャデータレジスタに書き込むことができます。

14.5.2.1 インプットキャプチャフェッチタイミングの例

- キャプチャ 0:立上りエッジ
- キャプチャ 1:立下りエッジ
- キャプチャの例:両方のエッジ

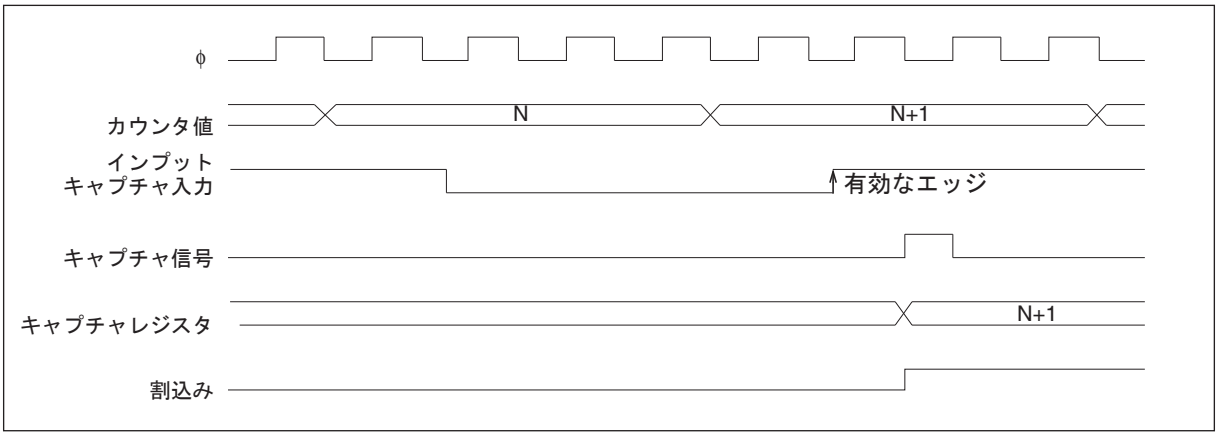
Figure 14-24. インプットキャプチャフェッチタイミングの例



14.5.2.2 インプットキャプチャインプットタイミング

入力信号のキャプチャタイミング

Figure 14-25. 入力信号のキャプチャタイミング



15. 16ビットリロードタイマ(イベントカウント機能付き)

16 ビットリロードタイマ(イベントカウント機能付き)の機能と動作について説明します。

15.1 16ビットリロードタイマ(イベントカウント機能付き)の概要

16 ビットリロードタイマは, 16 ビットダウンカウンタ, 16 ビットリロードレジスタ, 入力端子 1 本(TINn), 出力端子 1 本(TOTn), コントロールレジスタから構成されます。

16ビットリロードタイマ(イベントカウント機能付き)の概要

16 ビットリロードタイマには次の機能があります。

- 外部と内部のクロック/イベントソース
- トリガ信号として立上り, 立下り, 両エッジをプログラム可能
- ゲートカウント機能
- ワンショットカウンタモードとリロードカウンタモード
- カウンタ状態を外部端子に出力可能
- 内部クロックに対して 6 つ, 外部クロックに対して 2 つの設定を有するプリスケーラ
- 複数のリロードタイマのカスケード接続による長いリロードタイマの構成

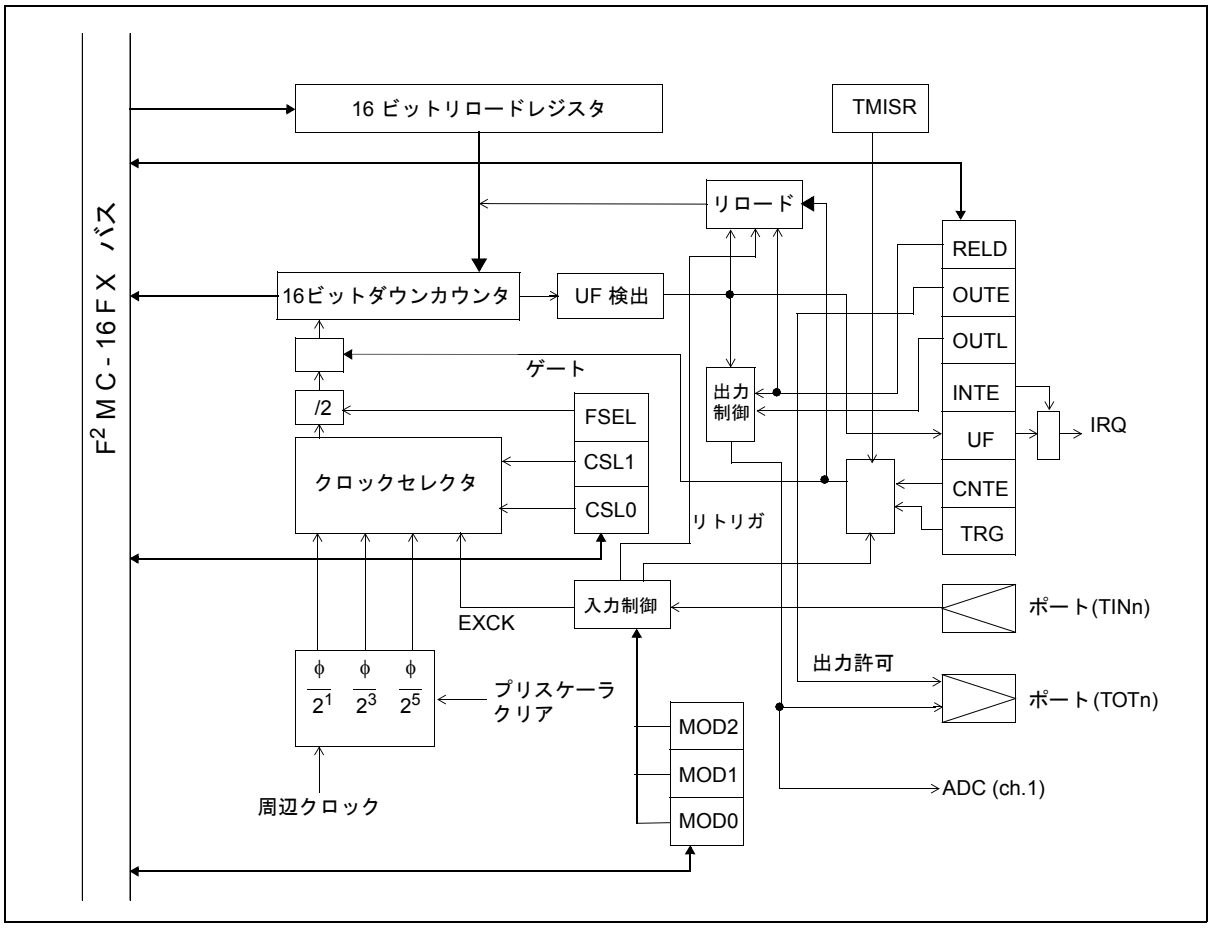
DMAと割込み

本タイマは DMA 転送の開始に使用可能な割込みを発生できます。

16ビットリロードタイマのブロックダイヤグラム

Figure 15-1 に , 16 ビットリロードタイマのブロックダイヤグラムを示します。

Figure 15-1. 16ビットリロードタイマのブロックダイヤグラム



＜注意事項＞

- サフィックス "n" はリロードタイマ番号を表します。
- リロードタイマ 1 の出力で A/D コンバータの変換を開始することができます。
- リロードタイマ 6 を PPG クロックソースとして使用できます。TIN/TOT 端子はありませんが , その制約以外では , PPG 動作に必要でない場合は , 通常のリロードタイマとして使用できます。

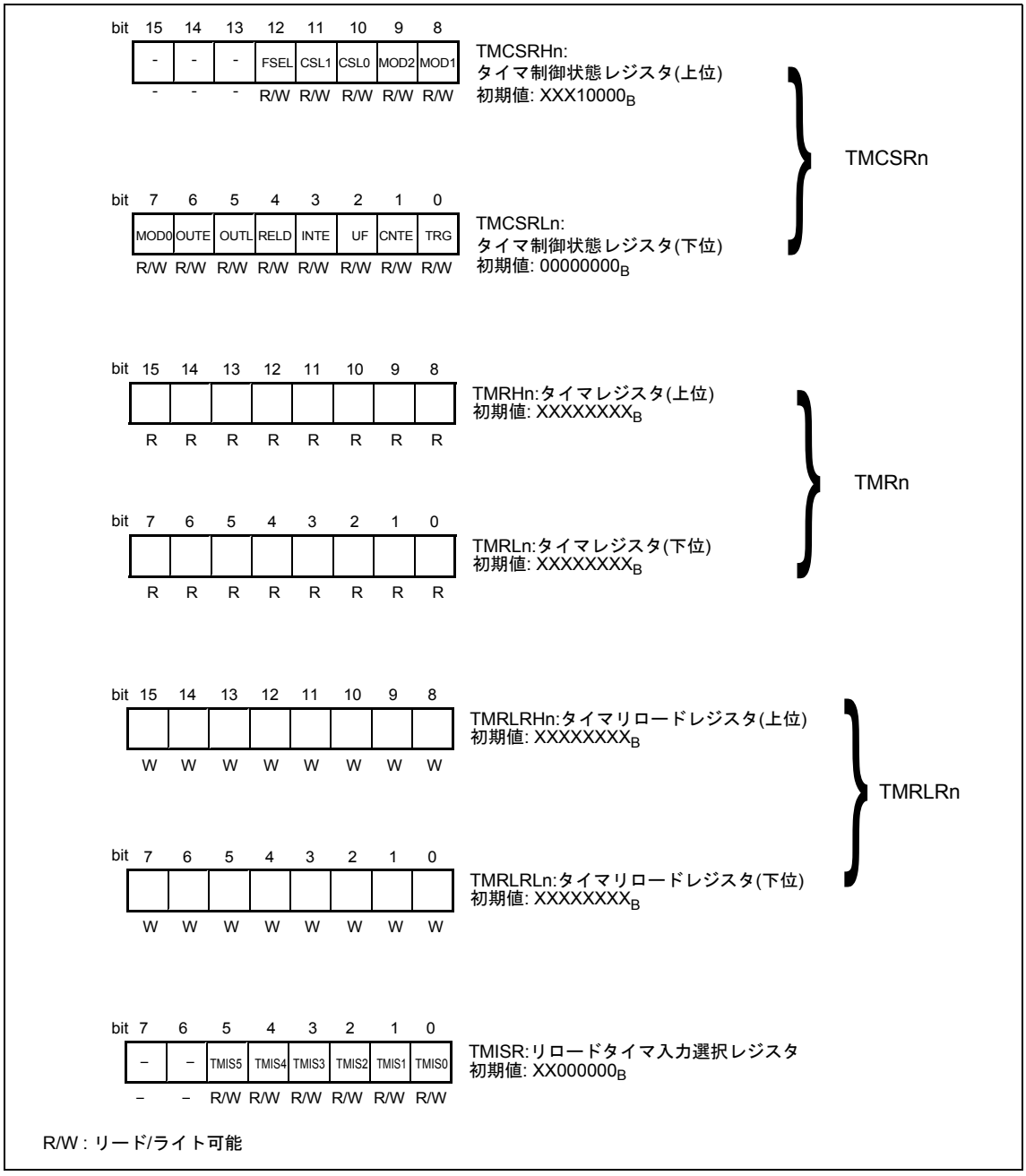
15.2 16ビットリロードタイマ(イベントカウント機能付き)

16 ビットリロードタイマには次のレジスタがあります。

- タイマ制御状態レジスタ(TMCSRn)
- 16 ビットタイマレジスタ(TMRn) / 16 ビットリロードレジスタ(TMRLRn)
- リロードタイマ入力選択レジスタ(TMISR)

16ビットリロードタイマレジスタ

Figure 15-2. 16ビットリロードタイマレジスタ



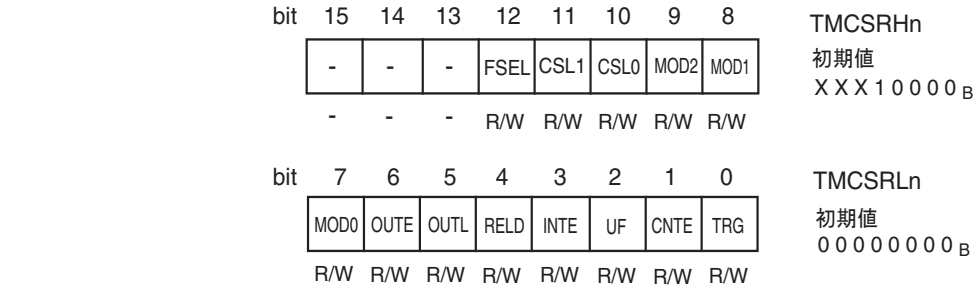
- 8 ビットレジスタの TMCSRHn と TMCSRLn は, 16 ビットレジスタの TMCSRn としてアクセスできます。
- 8 ビットレジスタの TMRHn と TMRLn は, 16 ビットレジスタの TMRn としてアクセスできます。
- 8 ビットレジスタの TMRLRHn と TMRLRLn は, 16 ビットレジスタの TMRLRn としてアクセスできます。

15.2.1 タイマ制御状態レジスタ(TMCSRn)

タイマ制御状態レジスタは, 16 ビットリロードタイマの動作モードと割込みを制御します。

15.2.1.1 タイマ制御状態レジスタ(TMCSRn)のレジスタ構成

Figure 15-3. タイマ制御状態レジスタ(TMCSRn)のレジスタ構成



R/W : リード/ライト可能

X : 不定値

- : 未定義

15.2.1.2 タイマ制御状態レジスタ(TMCSRn)の内容

[Bit 12] FSEL (カウントクロック分周制御)

カウントクロックの分周比を設定します。

FSEL が "0" に設定された場合には, カウントクロック選択ビット(CSL1, CSL0) で指定されたカウントクロックは 2 分周されます。

FSEL が "1" に設定された場合には, カウントクロック選択ビット (CSL1, CSL0) で指定されたカウントクロックは分周されません (初期値)。

[Bits 11, 10] CSL1, CSL0 (クロック選択1, 0)

入力されるクロック / イベントソースとクロックの分周比を設定します。

Table 15-1 に , FSEL と CSL0/CSL1 の設定によって選択されるクロックソースを示します。

Table 15-1. CSL0/1ビットとFSELビットの設定によって 選択されるクロックソース

FSEL	CSL1	CSL0	クロックソース(周辺クロックCLKP1 = 24 MHz)
1	0	0	CLKP1 / 2^1 (0.083 µs)
0	0	0	CLKP1 / 2^2 (0.167 µs)
1	0	1	CLKP1 / 2^3 (0.333µs)
0	0	1	CLKP1 / 2^4 (0.667µs)
1	1	0	CLKP1 / 2^5 (1.3 µs)
0	1	0	CLKP1 / 2^6 (2.6 µs)
1	1	1	外部イベントカウントモード
0	1	1	外部イベントカウントモード / 2

[Bits 9, 8, 7] MOD2, MOD1, MOD0 (動作モードおよびTINn機能)

16 ビットリロードタイマの動作モードと入力端子 (TINn) の機能を設定します。

MOD2 ビットで入力端子 (TINn) の機能を選択します。MOD2 =0 の場合は TINn はトリガ入力として機能します。この場合 , TINn 端子に有効なエッジが入力されるとリロードレジスタの内容がカウンタにロードされ , カウント動作が継続されます。MOD2 =1 の場合には , タイマ動作はゲートカウンタモードとなり , TINn 端子はゲート入力として機能します。このモードでは , TINn 端子に有効なレベルが入力されている間のみカウント動作が行われます。

Table 15-2 と Table 15-3 に MOD2/MOD1/MOD0 ビットの設定を示します。

Table 15-2. 内部クロックモードでのMOD2/1/0ビットの設定 (CSL0/1 = “00_B”, “01_B”, or “10_B”)

MOD2	MOD1	MOD0	入力端子の機能	有効なエッジまたはレベル
0	0	0	トリガ禁止	-
0	0	1	トリガ入力	立上りエッジ
0	1	0		立下りエッジ
0	1	1		両エッジ
1	x	0	ゲート入力	"L"レベル
1	x	1		"H"レベル

Table 15-3. イベントカウンタモードでのMOD2/1/0ビットの設定 (CSL0/1 = “11_B”)

MOD2	MOD1	MOD0	入力端子の機能	有効なエッジまたはレベル
x	0	0	-	-
	0	1	イベント入力	立上りエッジ
	1	0		立下りエッジ
	1	1		両エッジ

(注意事項)表中のXは, 任意の値に設定できます。

[Bit 6] OUTE (出力許可)

"1" に設定すると, TOTn 端子はリロードタイマ出力として使用されます。"0" に設定すると, タイマ出力 TOTn は禁止されます。

[Bit 5] OUTL (出力レベル)

TOTn 端子への出力レベルを設定します。

[Bit 4] RELD (リロード)

リロード動作を有効にします。

RELD を "1" に設定すると , タイマはリロードモードで動作します。このモードでは , アンダフローが発生 (カウンタの値が 0000_H から $FFFF_H$ に変化) すると , リロードレジスタの内容がカウンタにロードされ , カウント動作を継続します。

RELD を "0" に設定すると , タイマはワンショットモードで動作します。このモードでは , カウンタの値が 0000_H から $FFFF_H$ に変化してアンダフローが発生すると , カウント動作は停止します。

Table 15-4. OUTE, OUTL, RELDの設定

OUTE	OUTL	RELD	出力波形
0	x	x	タイマ出力禁止
1	0	0	カウント動作中に "H" レベルのパルスを出力します。
1	1	0	カウント動作中に "L" レベルのパルスを出力します。
1	0	1	"L" レベル出力から始めてトグル出力を行い, タイマのリロードごとに出力レベルを変化させます。
1	1	1	"H" レベル出力から始めてトグル出力を行い, タイマのリロードごとに出力レベルを変化させます。

[Bit 3] INTE (割込み許可)

タイマ割込み要求許可ビットです。

"1" に設定すると , UF ビットが "1" に変化した際に割込み要求が発生します。

"0" に設定すると , UF ビットが "1" に変化しても割込み要求は発生しません。

[Bit 2] UF (アンダフロー)

タイマ割込み要求フラグです。アンダフローが発生 (カウンタの値が 0000_H から $FFFF_H$ に変化) すると , "1" に設定されます。"0" の書込みまたは DMA コントローラによってクリアされます。"1" の書込みは動作に影響しません。リードモディファイライト命令時は "1" が読み出されます。

[Bit 1] CNTE (カウント許可)

タイマカウント許可ビットです。CNTE に "1" を書き込むと , タイマはトリガを待機します。"0" を書き込むとカウント動作を停止します。

[Bit 0] TRG (トリガ)

ソフトウェアトリガビットです。"1" を書き込むとソフトウェアトリガが発生し , リロードレジスタの内容がカウンタにロードされ , カウントが開始されます。

"0" の書込みは動作に影響を与えません。読出し時は常に "0" が読み出されます。このレジスタを使用したトリガは CNTE =1 の場合にのみ有効です。

CNTE =0 の場合には , "1" の書込みは動作に影響を与えません。

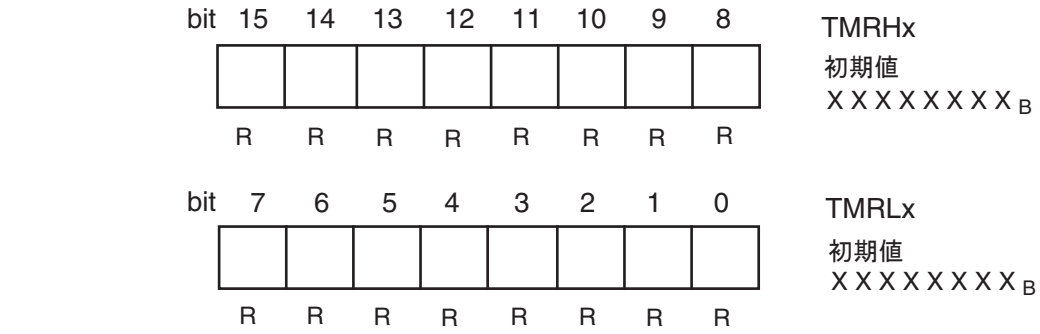
ゲート入力モードでこのビットを設定すると, カウント開始前にリロードレジスタの内容がロードされます。

15.2.2 16ビットタイマレジスタ(TMRn)/16ビットリロードレジスタ(TMRLRn)のレジスタ構成

- TMRn の機能
  このレジスタを読み出すと , 16 ビットリロードタイマのカウント値が読み出されます。初期値は未定義です。
- TMRLRn の機能
  16 ビットリロードレジスタはリロード値を保持します。初期値は未定義です。

16ビットタイマレジスタ(TMRn)のレジスタ構成

Figure 15-4. 16ビットタイマレジスタ(TMRn)のレジスタ構成

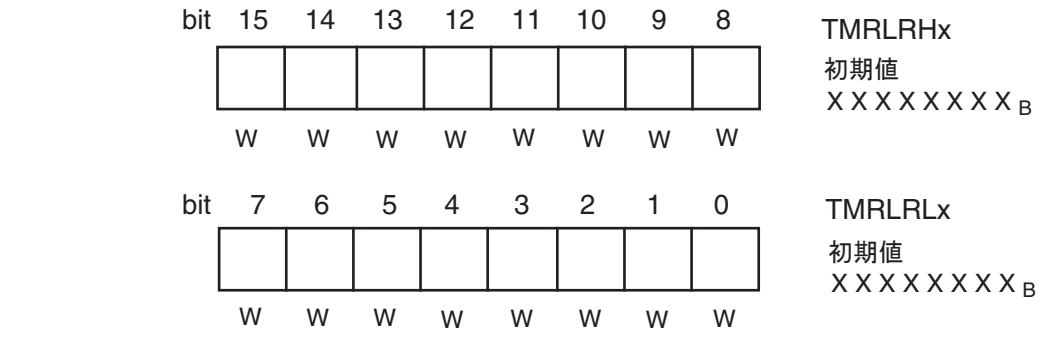


R : リードオンリ

X : 不定値

16ビットリロードレジスタ(TMRLRn)のレジスタ構成

Figure 15-5. 16ビットリロードレジスタ(TMRLRn)のレジスタ構成



W : ライトオンリ

X : 不定値

15.3 16ビットリロードタイマの内部クロックカウンタ動作と外部イベントカウンタ動作

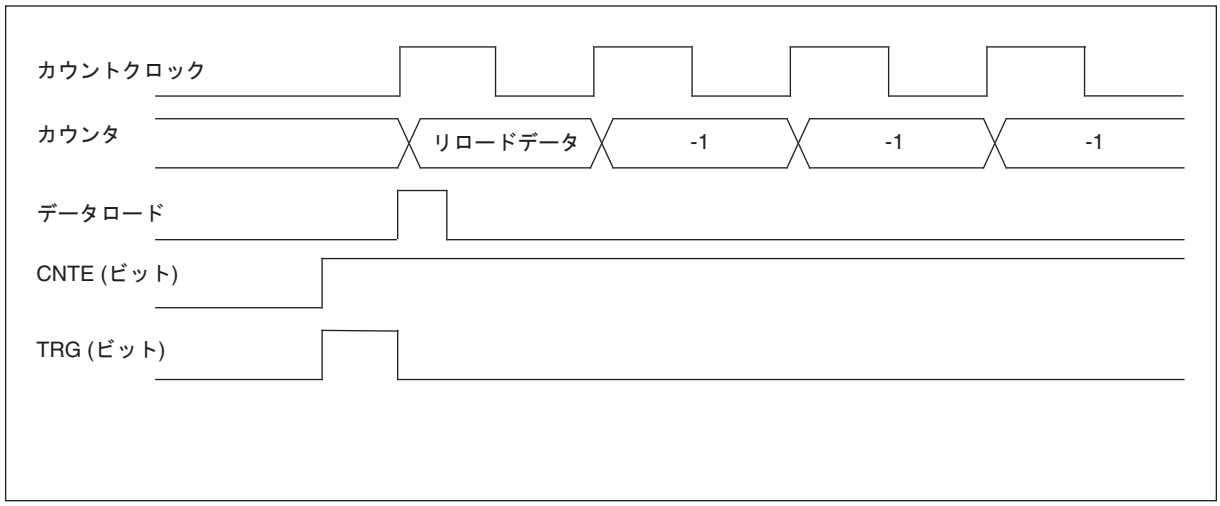
内部クロックモードでは, 各分周設定の周辺クロック (CLKP1) はリロードタイマを動作させるクロックソースとして選択できます。外部入力端子(TINn)はレジスタ設定によってトリガ入力またはゲート入力に設定できます。イベントカウンタモードでは, TINn 端子は外部イベント入力端子として使用されます。この端子の各アクティブエッジ (立上り, 立下り, 両エッジ)でカウンタがデクリメントされます。
FSEL = 0 の場合には, 1 つおきのイベントがカウントされます。

16ビットリロードタイマの内部クロック動作

制御状態レジスタの CNTE ビットと TRG ビットの両方に "1" を書き込むと , カウントの許可と開始が同時に行われます。タイマが許可されている (CNTE = 1) 場合には , 動作モードにかかわらず , TRG ビットをトリガ入力として使用できます。

Figure 15-6 にカウンタの起動とカウンタ動作を示します。

Figure 15-6. 16ビットリロードタイマカウンタの起動と動作

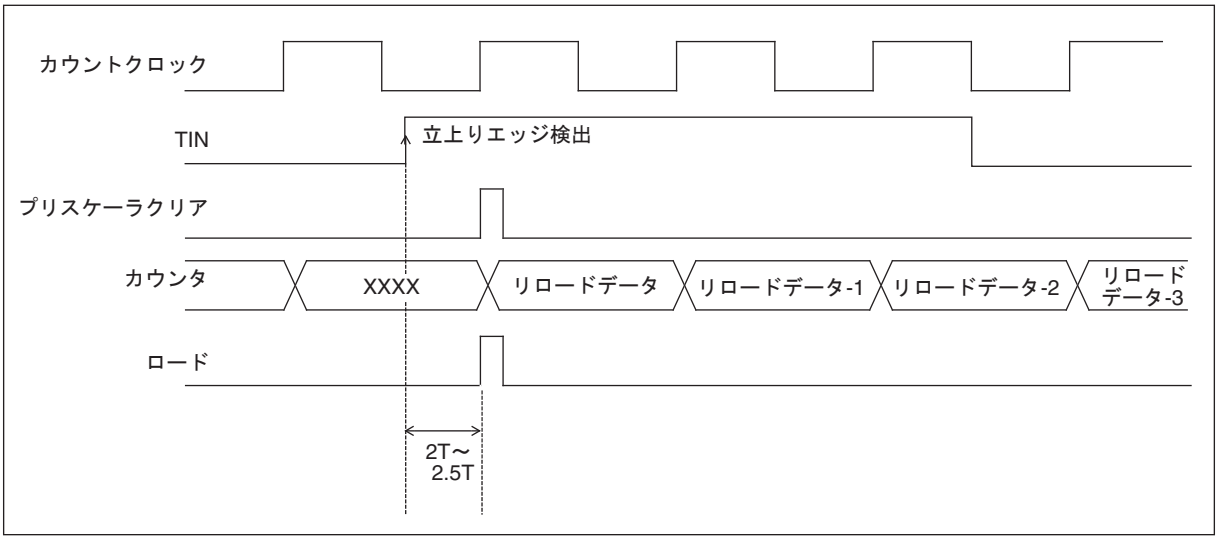


16ビットリロードタイマの入力端子の機能(内部クロックモード)

クロックソースとして内部クロックが選択された場合には , TINn 端子はトリガ入力またはゲート入力として使用できます。トリガ入力として使用された場合には , アクティブエッジによってリロードレジスタの内容がタイマにロードされ , 内部プリスケーラがリセットされます。その後カウント動作を開始します。TIN に必要とされる最小パルス幅は 2T+200ns です(T: 周辺クロック (CLKP1) の周期 , 200ns は入力のノイズフィルタで遮断されない最小のパルス幅です)。

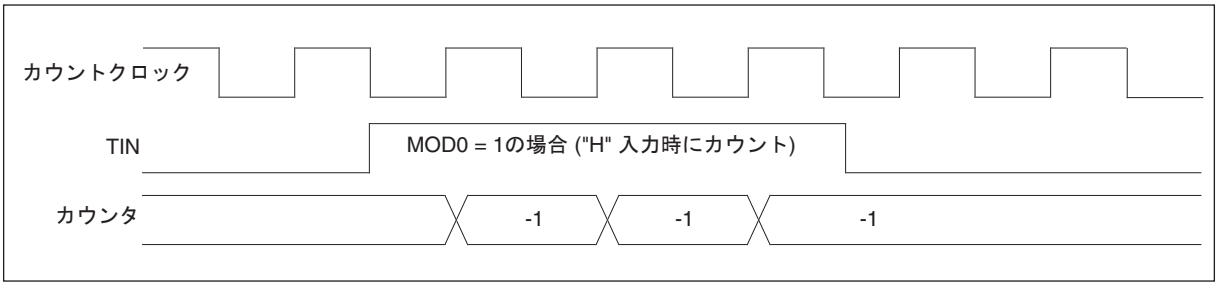
Figure 15-7 にトリガ入力の動作を示します。

Figure 15-7. 16ビットリロードタイマのトリガ入力動作



ゲート入力として使用される場合には, 制御レジスタの MOD0 ビットで指定されたアクティブレベルが TINn 端子に入力されている時にのみカウントが行われます。この場合 , カウントクロックが停止しないかぎりカウントし続けます。ゲートモードでは , ゲートレベルにかかわらずソフトウェアトリガを使用できます。TINn 端子には 2T +200ns 以上の幅のパルスを入力してください (T : 周辺クロック CLKP1 の周期)。Figure 15-8 にゲート入力の動作を示します。

Figure 15-8. 16ビットリロードタイマのゲート入力動作



外部イベントカウンタ

外部イベントカウントモードが選択された場合には , TINn 端子は外部イベント入力として使用されます。カウンタは TMCSRn で設定されたアクティブエッジをカウントします。TINn 端子への入力パルス幅は最小 4T+200ns です (T: 周辺クロック (CLKP1) の周期)。

15.4 16ビットリロードタイマのアンダフロー動作

16 ビットリロードタイマのアンダフローとは, カウンタ値が 0000_H から $FFFF_H$ に変化する時を指します。そのためアンダフローは, (リロードレジスタ設定 + 1) カウント後に発生します。

16ビットリロードタイマのアンダフロー動作

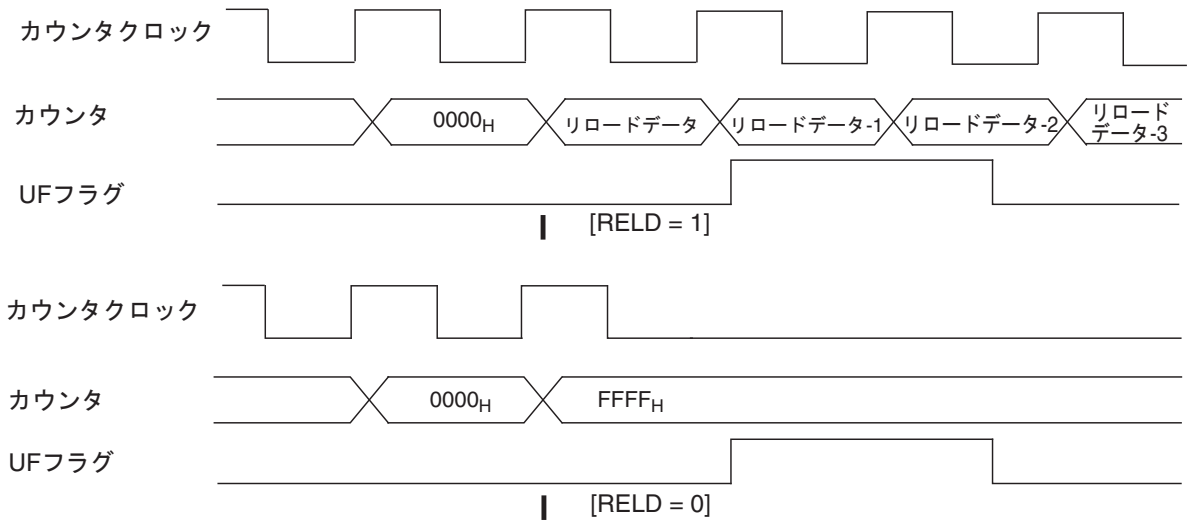
制御状態レジスタの RELD ビットが "1" に設定された状態でアンダフローが発生すると, リロードレジスタの内容がカウンタにロードされ , カウントが継続されます。

制御状態レジスタの RELD ビットが "0" に設定されている場合には , カウンタが $FFFF_H$ に達するとカウントが停止します。

アンダフローが発生すると , 制御状態レジスタの UF ビットがセットされます。このとき INTE ビットが "1" であると , 割込み要求が発生します。

Figure 15-9 に , アンダフロー発生時の動作を示します。

Figure 15-9. 16ビットリロードタイマのアンダフロー動作



15.5 16ビットリロードタイマの出力端子機能

リロードモードでは, TOTn 端子はトグル出力(アンダフローごとに反転)を行います。ワンショットモードでは, カウント中に設定されたレベルを示すパルス出力として TOTn端子を使用します。

16ビットリロードタイマの出力端子機能

制御状態レジスタの OUTL ビットで出力極性を設定します。

OUTL =0 の場合にはトグル出力の初期値は "L" であり , ワンショットパルス出力はカウント中に "H" になります。

OUTL =1 の場合には , 出力波形は反転します。

Figure 15-10 と Figure 15-11 に出力端子機能を示します。

Figure 15-10. リロードモードの16ビットリロードタイマの出力端子機能

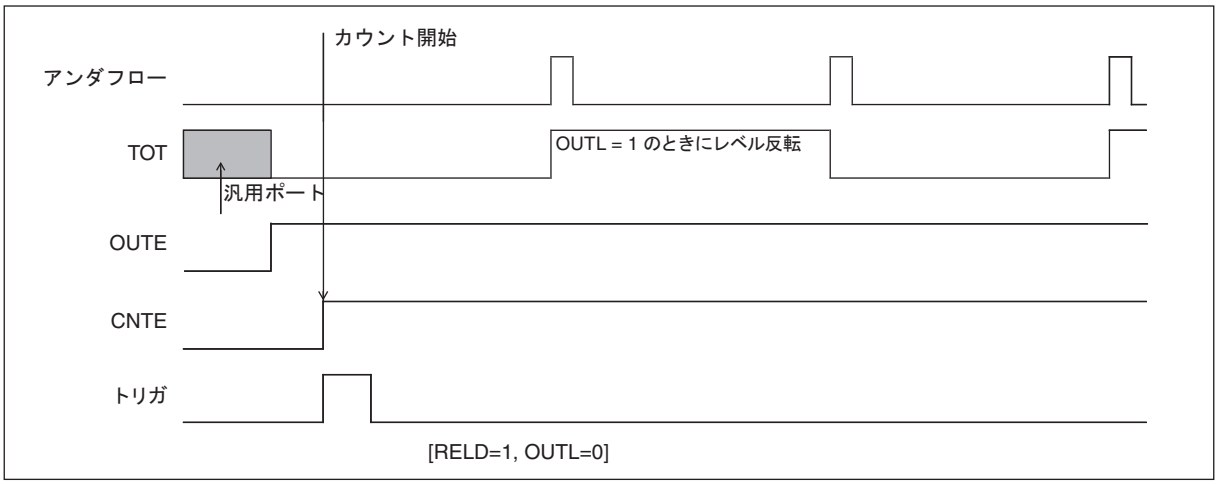
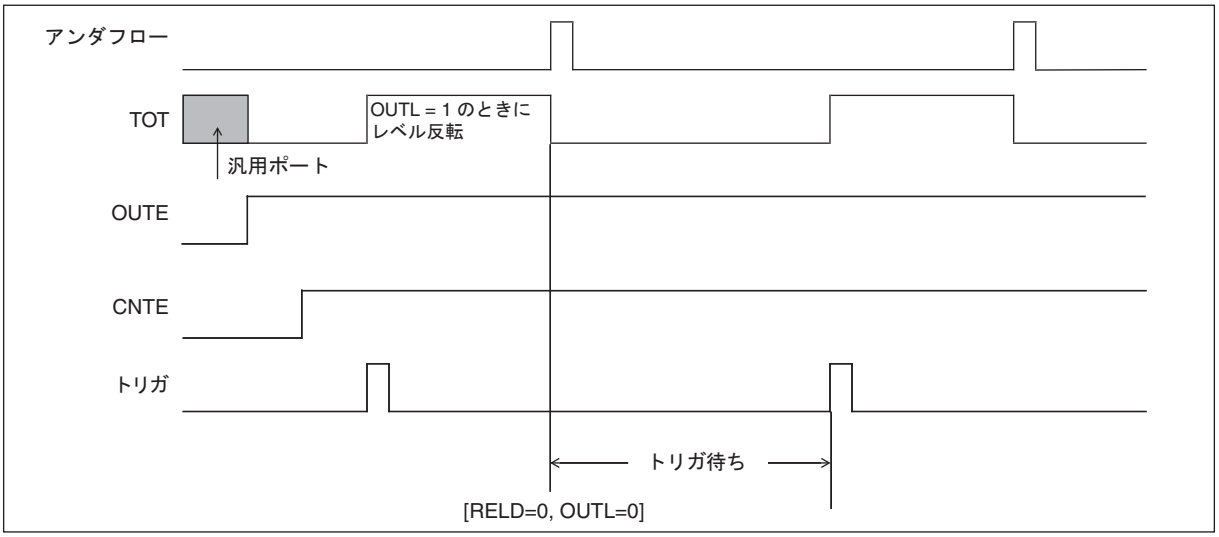


Figure 15-11. ワンショットモードの16ビットリロードタイマの出力端子機能



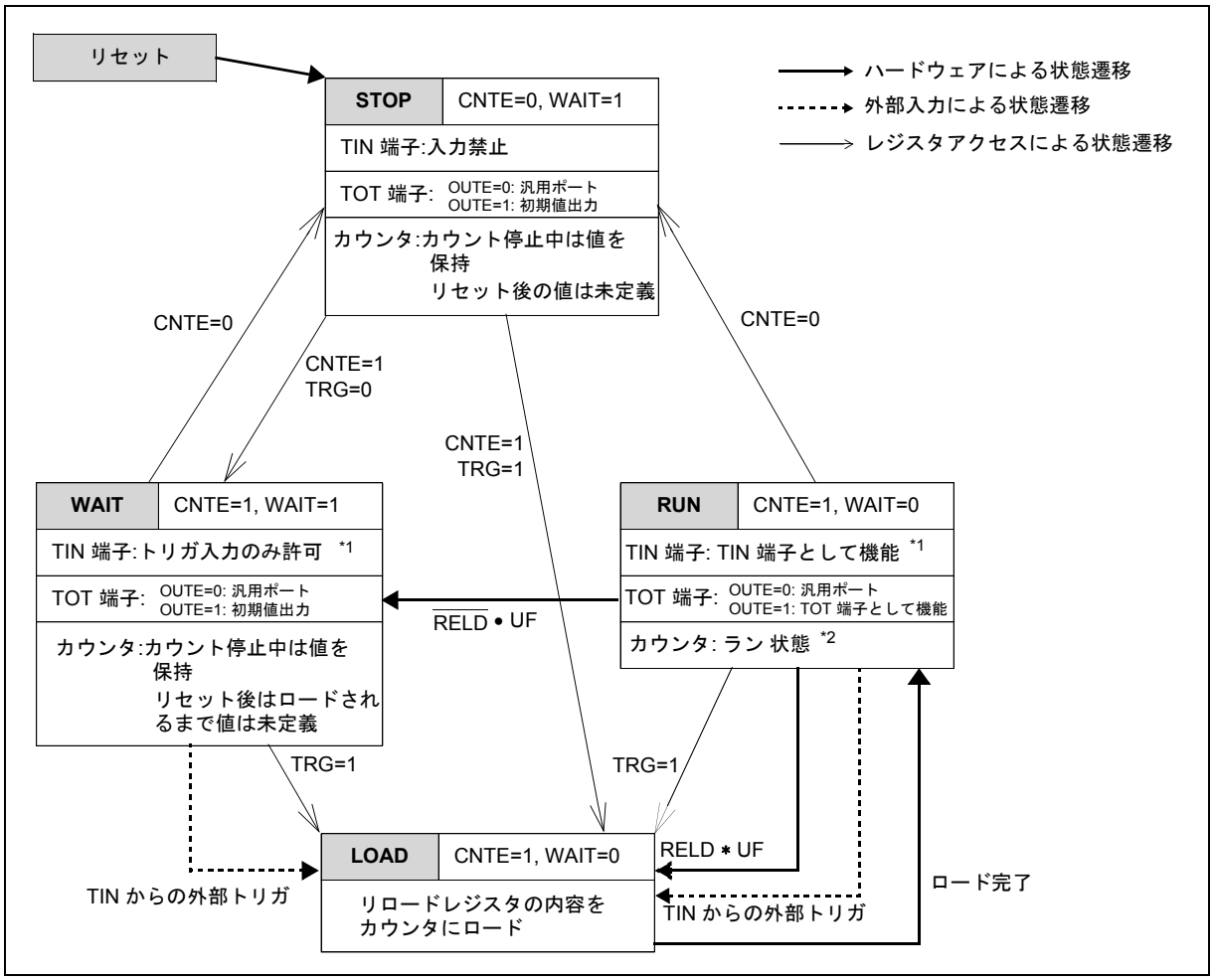
15.6 カウンタの動作状態

カウンタの状態は,制御状態レジスタのCNTEビットと内部WAIT信号で決定されます。動作状態としては,CNTE=0＆WAIT=1(ストップ状態),CNTE=1＆WAIT=1(トリガWAIT状態),CNTE=1＆WAIT=0(ラン状態)があります。

カウンタの動作状態

Figure 15-12 に , 各状態の遷移を示します。

Figure 15-12. カウンタの状態遷移



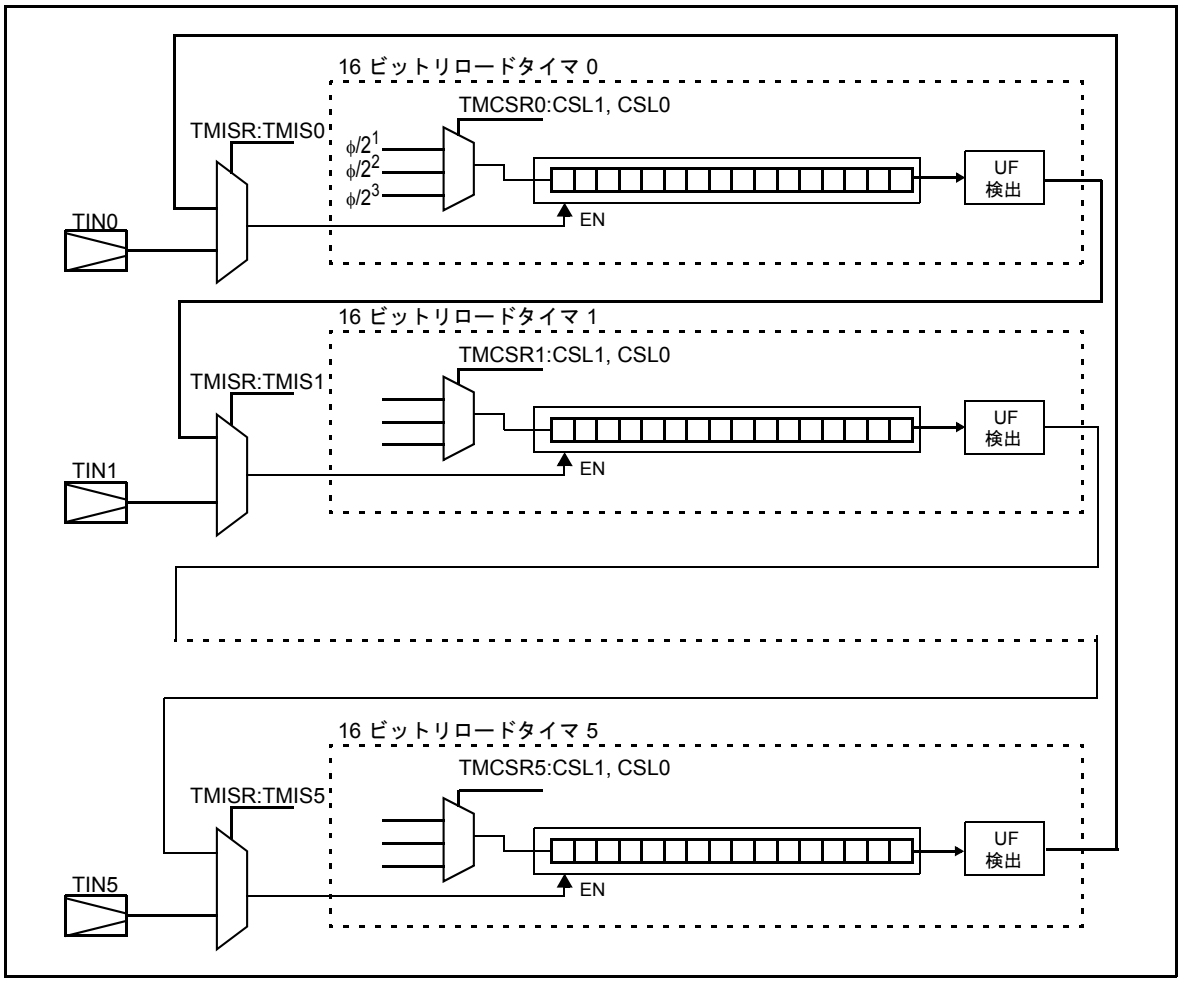
*1: TIN 端子を使用する前に, DDR レジスタの対応するビットを "0" に, PIER レジスタの対応するビットを "1" に設定する必要があります。
*2: ゲート入力モードでは, カウントは TIN に影響されます。

15.7 16ビットリロードタイマのカスケード接続

隣接する複数の 16 ビットリロードタイマをカスケード接続すると, n ´ 16 ビットリロードタイマを構成できます(例: 3 つの隣接するリロードタイマがデバイス上に存在)。

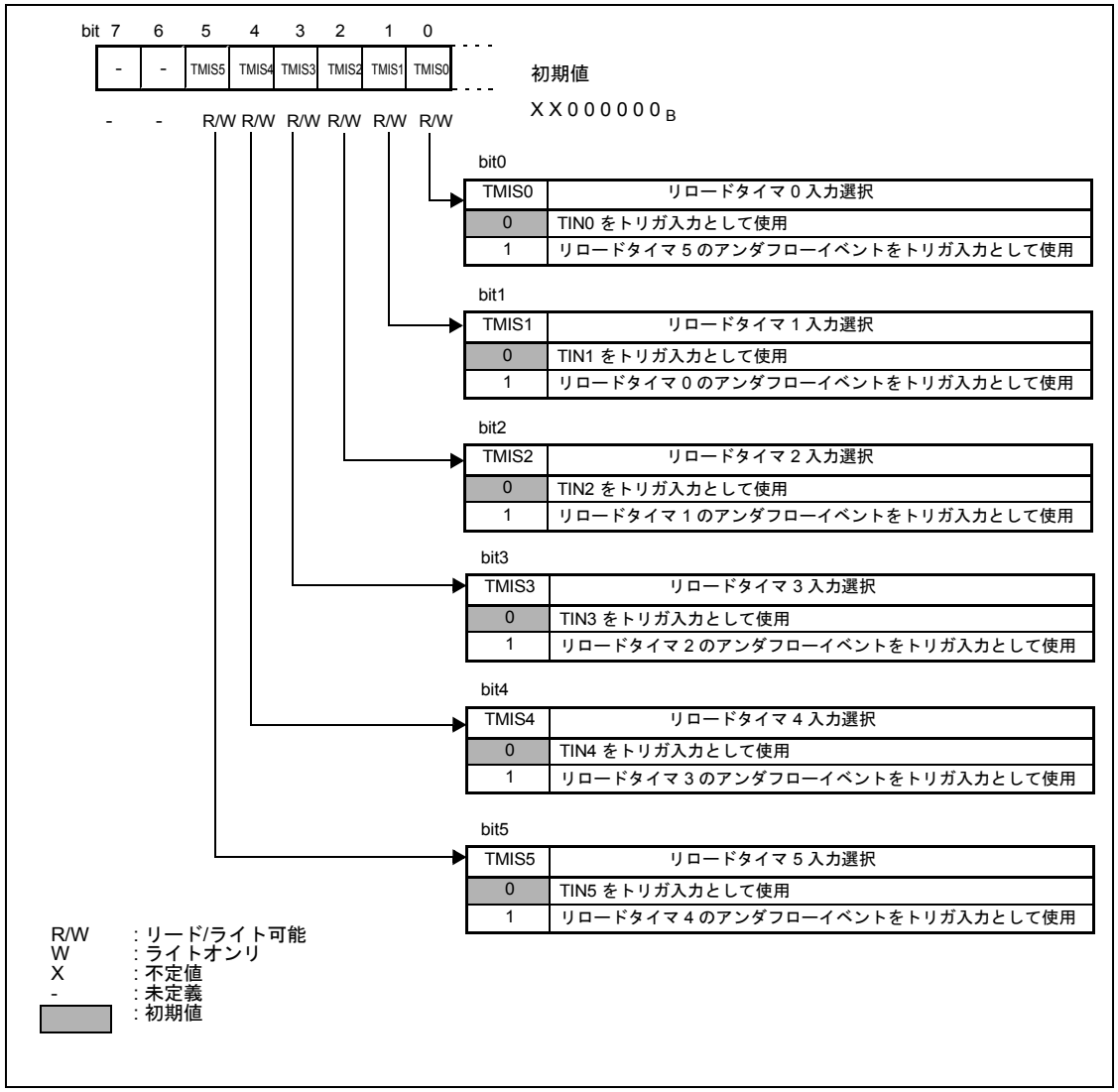
カスケード接続の概要

Figure 15-13. 16ビットリロードタイマのカスケード接続のブロックダイヤグラム (3つのリロードタイマがデバイス上に存在する例)



リロードタイマ入力選択レジスタ(TMISR)

Figure 15-14. リロードタイマ入力選択レジスタ(TMISR)



動作説明

カスケード接続された最初のリロードタイマは, カウンタクロックとして内部クロックまたは外部イベントを使用できます。このリロードタイマは , 自由に設定できます。
連結された , 2 番目以降のリロードタイマは , 必ず次のように設定してください。

- TMCSRn:MOD[2:0]=101_B :ゲートカウントモード "H" レベル
- TMCSRn:CSL[1:0]=00_BおよびTMCSRn:FSEL=1:CLKP1は2分周されますが(表15.2-1 を参照),CLKP1の周期と等しいUF信号を確実にサンプリングできるよう, 最初の RLT 以外のすべての RLT は, CLKP1 を 2 分周したクロックが供給されます。
- TMISR:TMISx=1 :前のタイマのアンダフロー信号をカウント許可信号として使用するすべてのリロードタイマは, リロード値をロードして ラン 状態に移行するために, カウント開始前にトリガ入力してください(Figure 15-12参照)。

16. PPG

PPG の機能と動作について説明します。

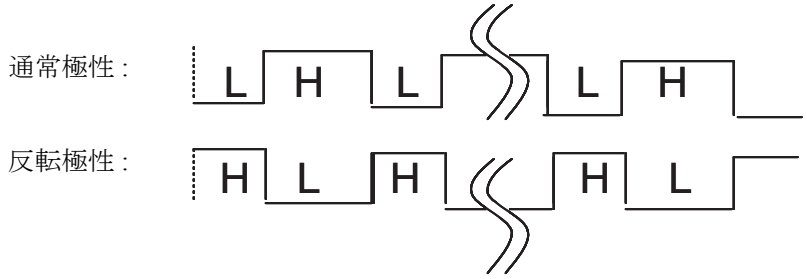
16.1 PPGの概要

プログラマブルパルスジェネレータ(PPG)は, ワンショット(矩形波)出力またはパルス幅変調(PWM)出力を得るために使用します。周期とデューティをソフトウェアでプログラム可能なことにより, PPG は広範なアプリケーションに容易に適合できます。

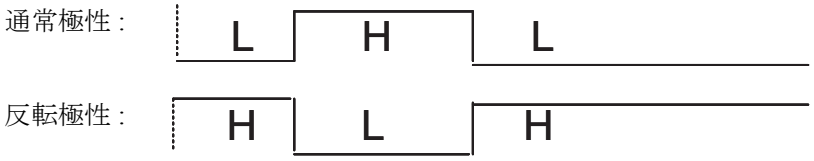
PPGの特長

(出力波形 : PPG は次の種類の波形を出力できます。)

PWM 波形



ワンショット波形(矩形波)



固定出力 :

- 通常極性: "L" 固定出力
- 反転極性: "H" 固定出力

カウントクロック : 次の 8 種類から選択 :

- 周辺クロック(CLKP1) またはリロードタイマ 6 のアンダフローの 1, 1/4, 1/16 (「第15章 "16 ビットリロードタイマ (イベントカウント機能付き)"」参照)

周期 : 設定範囲 = デューティ値 ～ 65535 (16 ビットレジスタで指定)

- 周期 = カウントクロック×(PCSR レジスタ値 + 1)
- (例) カウントクロック = 32MHz (31.25ns), PCSR 値 = 63999
- 周期 = 31.25ns×(63999+1) = 2ms

デューティ : 設定範囲 = 0 ～ 周期値 (16 ビットレジスタで指定)
- デューティ = カウントクロック(PDUT レジスタ値 + 1)

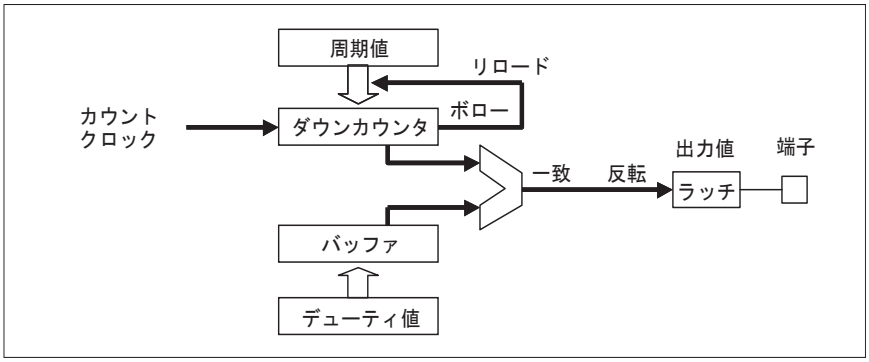
割込み : 次の 4 種類から選択 :
- ソフトウェアトリガまたは外部トリガ(TTGx 端子)
- カウンタのボロー発生(周期一致)
- デューティ一致発生
- カウンタのボロー発生(周期一致), またはデューティ一致発生

起動トリガ
- ソフトウェアトリガ
- 内部トリガ
- 外部トリガ(TTG 端子)

追加プリスケーラ入力としての自由に構成可能なリロードタイマを選択できます。

PPGの単純化されたブロックダイヤグラム

Figure 16-1. PPGの単純化されたブロックダイヤグラム

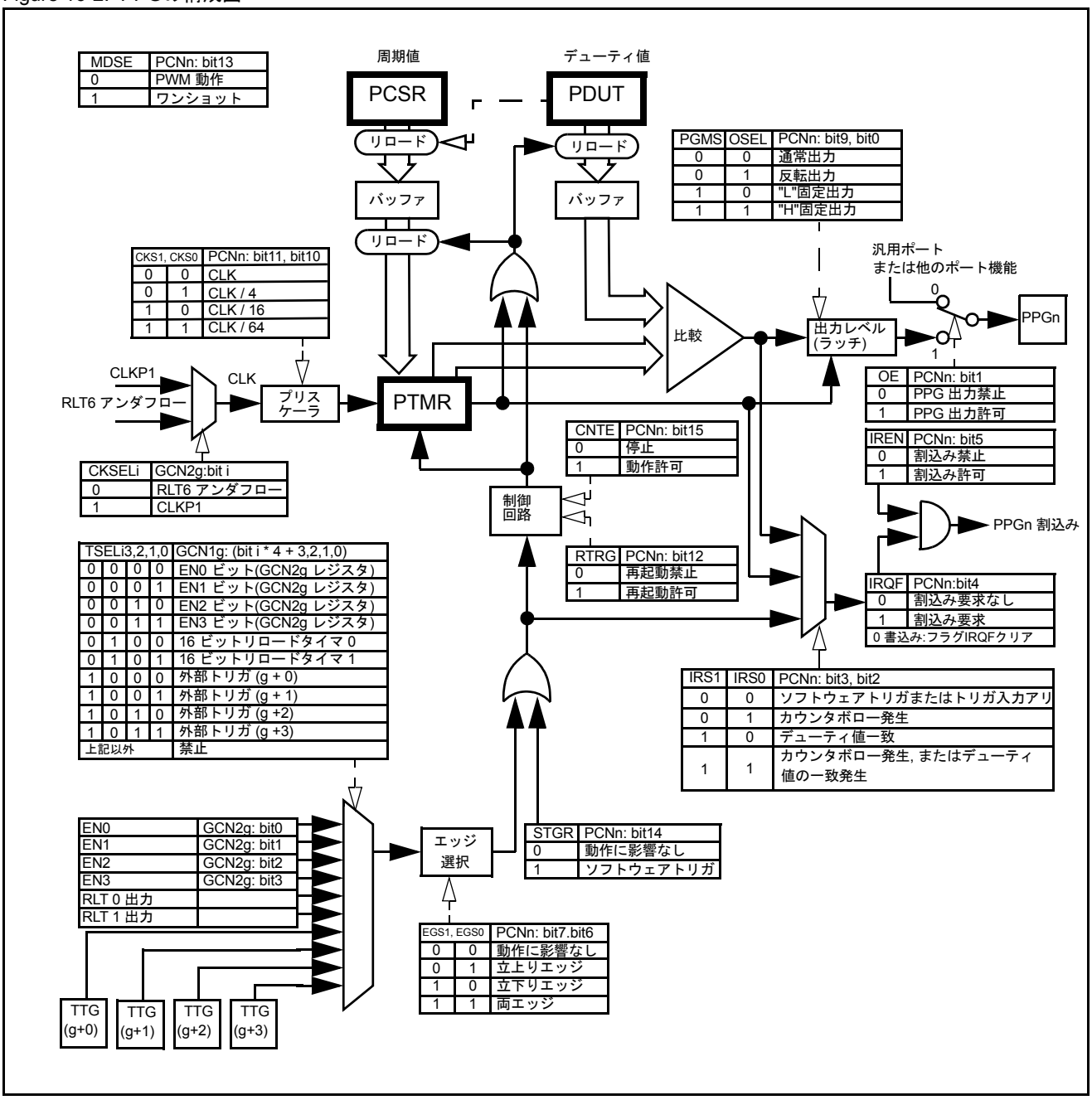


PPGの構成

4 つの PPG が共通の GCN レジスタによってグループ化されます。Figure 16-2 PPG の構成図では , 以下の表記が使用されています。

- n : 個別のPPG番号を示します。
- g : 4本のPPGからなるグループを示します。
- i : グループ内の PPG のインデックスを示します。

Figure 16-2. PPGの構成図



＜注意事項＞

本ダイヤグラムはほかの PPG に対しても当てはまります。RLT6 の構成については , 「16 ビットリロードタイマ (イベントカウント機能付き)」を参照してください。

RLT6 は PPG 動作で必要でなければ , 通常の RLT として使用できます。

16.2 レジスタ

PPG には次のレジスタがあります。

- PPG 制御ステータスレジスタ(PCNn)
- 汎用制御レジスタ 1 (GCN1g)
- 汎用制御レジスタ 2 (GCN2g)
- PPG 周期設定レジスタ(PCSRn)
- PPG デューティ設定レジスタ(PDUTn)
- PPG タイマレジスタ(PTMRn)

16.2.1 レジスター覧

Figure 16-3. レジスター覧

	bit15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PCNn	CNTE	STGR	MDSE	RTRG	CKS1	CKS0	PGMS	-	EGS1	EGS0	IREN	IRQF	IRS1	IRS0	OE	OSEL
GCN1g	TSEL33	TSEL32	TSEL31	TSEL30	TSEL23	TSEL22	TSEL21	TSEL20	TSEL13	TSEL12	TSEL11	TSEL10	TSEL03	TSEL02	TSEL01	TSEL00
GCN2g	-	-	-	-	CKSEL3	CKSEL2	CKSEL1	CKSEL0	-	-	-	-	EN3	EN2	EN1	EN0
PCSRn	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
PDUTn	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
PTMRn	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

(注意事項) n は個別 PPG を表し, g は 4 本の PPG からなるグループを表します。

16.2.2 PPG制御ステータスレジスタ(PCNn)

PPG 制御状態レジスタ(PCNn)は PPG の動作制御および状態を表示するレジスタです。

Figure 16-4. PPG制御状態レジスタ(PCNn)

PCNHn

bit	15	14	13	12	11	10	9	8
	CNTE	STGR	MDSE	RTRG	CKS1	CKS0	PGMS	-
属性:	R/W	R0/W	R/W	R/W	R/W	R/W	R/W	RX/WX
初期値:	0	0	0	0	0	0	0	X
動作中の書換え:	O	O	X	X	X	X	X	-

PCNLn

bit	7	6	5	4	3	2	1	0
	EGS1	EGS0	IREN	IRQF	IRS1	IRS0	OE	OSEL
属性:	R/W	R/W	R/W	R(RM1)/W	R/W	R/W	R/W	R/W
初期値:	0	0	0	0	0	0	0	0
動作中の書換え:	X	X	O	O	X	X	O	X

R/W : リード/ライト可能
X : 初期値不定
O : 書換え可能
X : 書換え不可

Bit 15: Timer動作許可

CNTE	動作
0	停止
1	動作

PPG の動作を許可するビットです

Bit 14: ソフトウェアトリガ

STGR	動作
0	動作は書込みに影響されない(読出し値は常に "0")
1	ソフトウェアトリガ起動

ソフトウェアトリガビットが "1" に設定されると , 内部トリガまたは外部トリガ (EN ビット , リロードタイマ出力 , TTG 入力) とは独立してソフトウェアトリガが発生し , PPG が起動します。このトリガはエッジ選択ビット EGS1 および EGS0 の設定には影響されません。

Bit 13: モード選択

MDSE	モード
0	PWM動作
1	ワンショット動作

モード選択ビットを "0" に設定すると PWM 動作が許可され , 連続したパルスを発生します。
モード選択ビットを "1" に設定すると , 1 回だけパルス出力が行われます。

Bit 12: 再起動許可

RTRG	動作
0	再起動を禁止
1	再起動を許可

再起動許可ビットを "1" に設定すると , トリガ (ソフトウェア / 内部 / 外部) により PPG の動作が再起動します (トリガの構成に依存)。

Bits 11-10: カウンタクロック選択

CKS1	CKS0	ダウンカウンタのカウントクロック選択
0	0	CKSEL で選択されたクロック
0	1	CKSEL で選択されたクロックを 4 分周したクロック
1	0	CKSEL で選択されたクロックを 16 分周したクロック
1	1	CKSEL で選択されたクロックを 64 分周したクロック

Bit 9: PPG出力マスク選択

PGMS	動作
0	出力マスクなし
1	出力マスク(OSEL =0 のとき, 出力は "L" レベルに保持)

PPG の出力マスク選択ビットを "1" に設定すると , PPG 出力をモード , 周期 , デューティの設定にかかわらず "L" または "H" に固定できます。

出力レベルは出力極性指定ビット (PCN:OSEL) で指定できます。

Bit 8: 未定義

常に "0" を書き込んでください。読出し値は不定です。リードモディファイライト命令は無効です。

Bits 7-6: トリガ入力エッジ選択

EGS1	EGS0	選択されるエッジ
0	0	エッジの選択なし。PCNHn:STGR によるトリガのみ可能
0	1	立上りエッジ
1	0	立下りエッジ
1	1	両エッジ(立上りまたは立下り)

EGS1=0 および EGS0=0 の場合には , PCNn:STGR による PPG のトリガのみが可能です。PPGn に対するトリガ EN[3:0], リロードタイマ , 外部トリガは禁止されます。

EGS1 と EGS0 の他の設定は , トリガ EN[3:0], リロードタイマ , 外部トリガにのみ影響します。PCNn:STGR への "1" の書込みによる対応する PPG のトリガは , PCNn: EGS1 と PCNn:EGS0 の設定には影響されません。

Bit 5: 割込み要求の許可

IREN	動作
0	割込み要求を禁止
1	割込み要求を許可

Bit 4: 割込み要求フラグ

IRQF	読出し操作	書込み操作
0	割込み要求なし	割込み要求フラグをクリア
1	割込み要求あり	"1" の書込みは動作に影響なし

割込み要求フラグ (IRQF) が "1" になったときと同時に "0" を書き込んだ場合には , ハードウェアによる割込み要求フラグの設定 (IRQF=1) が優先します。

Bit 3-2: 割込み要求選択

IRS1	IRS0	選択
0	0	ソフトウェアトリガまたは外部トリガ入力
0	1	カウンタのボロー発生
1	0	カウンタとデューティ値の一致
1	1	カウンタのボロー発生, またはカウンタとデューティ値の一致

割込み要求を発生する動作を選択します。

Bit 1: PPG出力許可

OE	動作
0	出力禁止
1	出力許可

Bit 0: PPG出力極性選択

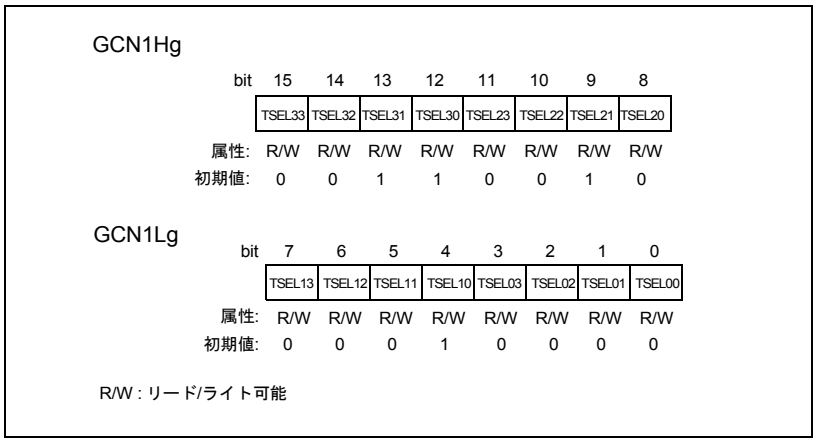
OSEL	動作
0	通常極性
1	I反転極性

PPG 出力マスク選択ビット (PCN:PGMS) が "1" に設定されている場合には , 出力極性選択ビット (OSEL) を "0" に設定すると , 出力は "L" にクランプされ , 出力極性選択ビットを "1" に設定すると出力は "H" にクランプされます。

16.2.3 汎用制御レジスタ1 (GCN1g)

汎用制御レジスタ 1 (GCN1g) は , 4 つの PPG からなる PPG グループへのトリガ入力を選択します。

Figure 16-5. 汎用制御レジスタ1 (GCN1g)



下記の表内で, インデックス "i" は PPG グループ内の PPG チャネル番号を表します(i = 0 ～ 3)。

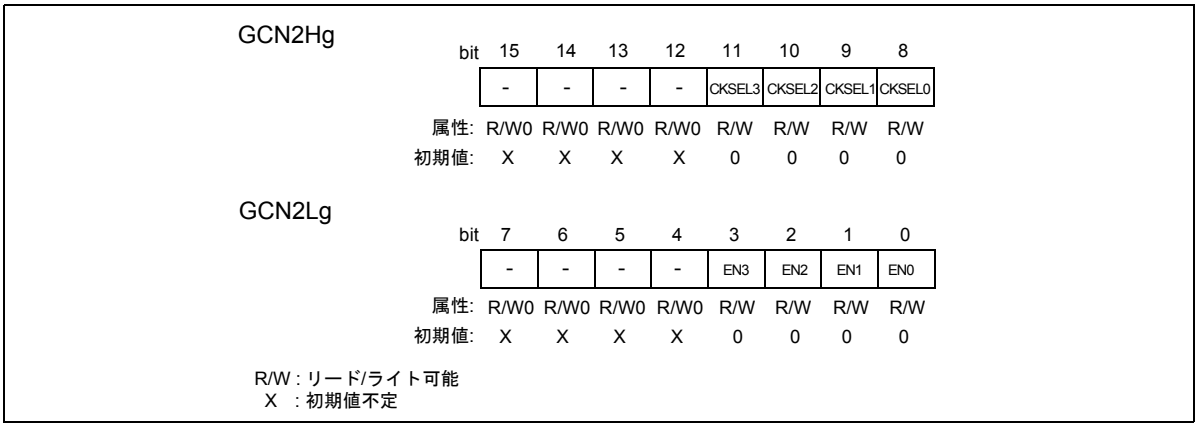
TSELi3	TSELi2	TSELi1	TSELi0	起動トリガ選択
0	0	0	0	EN0 ビット(GCN2 レジスタ)
0	0	0	1	EN1 ビット(GCN2 レジスタ)
0	0	1	0	EN2 ビット(GCN2 レジスタ)
0	0	1	1	EN3 ビット(GCN2 レジスタ)
0	1	0	0	16 ビットリロードタイマ 0 出力
0	1	0	1	16 ビットリロードタイマ 1 出力
1	0	0	0	外部トリガ (g + 0)
1	0	0	1	外部トリガ (g + 1)
1	0	1	0	外部トリガ (g + 2)
1	0	1	1	外部トリガ (g + 3)
上記以外				設定禁止

選択された PPG0 ～ PPGn は , トリガ入力エッジ選択ビット (PCNn:EGS[1:0]) で指定されたエッジが , 指定された起動トリガで検出されると起動します。

16.2.4 汎用制御レジスタ2 (GCN2g)

汎用制御レジスタ 2 (GCN2g) は , 4 つの PPG からなる PPG グループに対する内部トリガをソフトウェアで発生します。

Figure 16-6. 汎用制御レジスタ2 (GCN2g)



Bits 15-12: 未定義

常に "0" を書き込んでください。リードモディファイライト系命令は無効です。

Bits 11-8: プリスケーラ入力選択

CKSEL0..3	動作
0	周辺クロックCLKP1 をプリスケーラに入力
1	リロードタイマ6 RLT6のアンダフローをプリスケーラに入力

Bits 7-4: 未定義

常に "0" を書き込んでください。リードモディファイライト系命令は無効です。

Bit 3-0: EN0～EN3 トリガ入力

EN0～EN3	内部トリガ(EN0, EN1, EN2, EN3)
0	レベルを "L" に設定
1	レベルを "H" に設定

内部トリガ (EN0 ～ EN3) のレベルを設定します。

PPGn のトリガ指定ビット (GCN1g:TSEL0[3:0], GCN1g:TSEL1[3:0], GCN1g:TSEL2[3:0], GCN1g:TSEL3[3:0]) によっていずれかの EN トリガ入力 (EN0, EN1, EN2, EN3) が選択されている場合には , 選択された EN は PPG トリガ入力ビットとなります。

トリガ入力エッジ選択ビット (EGS[1:0]) で選択された状態がトリガ入力ビット (選択された EN0, EN1, EN2, EN3) を使用してソフトウェアで発生された場合には , PPG を起動するための起動トリガとなります。

16.2.5 PPG周期設定レジスタ(PCSRn)

PPG 周期設定レジスタ (PCSRn) は PPG の周期を制御します。

Figure 16-7. PPG周期設定レジスタ(PCSRn)

PCSRHn

bit	15	14	13	12	11	10	9	8
	D15	D14	D13	D12	D11	D10	D9	D8
属性:	W	W	W	W	W	W	W	W
初期値:	X	X	X	X	X	X	X	X

PCSRLn

bit	7	6	5	4	3	2	1	0
	D7	D6	D5	D4	D3	D2	D1	D0
属性:	W	W	W	W	W	W	W	W
初期値:	X	X	X	X	X	X	X	X

W：ライトオンリ
X：不定値

PPG 周期設定レジスタはバッファ付きです。バッファからカウンタへの転送は , 自動的にカウンタのボロー発生時に行われます。
PPG 周期レジスタの書換え後は , 必ず PPG デューティ設定レジスタ (PDUT) の設定を行ってください。

16.2.6 PPGデューティ設定レジスタ(PDUTn)

PPG デューティ設定レジスタ (PDUTn) は , PPG 出力波形のデューティを設定するためのレジスタです。

Figure 16-8. PPGデューティ設定レジスタ(PDUTn)

PDUTHn

bit	15	14	13	12	11	10	9	8
	D15	D14	D13	D12	D11	D10	D9	D8
属性:	W	W	W	W	W	W	W	W
初期値:	X	X	X	X	X	X	X	X

PDUTLn

bit	7	6	5	4	3	2	1	0
	D7	D6	D5	D4	D3	D2	D1	D0
属性:	W	W	W	W	W	W	W	W
初期値:	X	X	X	X	X	X	X	X

W : ライトオンリ
X : 初期値不定

PPG デューティ設定レジスタはバッファ付きです。バッファからカウンタへの転送は , 自動的にカウンタのボロー発生時に行われます。

PPG デューティ設定レジスタには , PPG 周期設定レジスタ (PCSR) に設定した値よりも小さな値を設定してください。

PPG デューティ設定レジスタに PPG 周期設定レジスタと同じ値を設定すると次のようになります。

- 通常極性(OSEL=0)の場合には, 常に "H" が出力されます。
- 反転極性(OSEL=1)の場合には, 常に "L" が出力されます。
- (OSEL ビットは PPG 制御レジスタ(PCN)の出力極性指定ビットです。)

16.2.7 PPGタイマレジスタ(PTMRn)

PPG タイマレジスタ (PTMRn) には PPGn のカウント値が読み出されます。

Figure 16-9. PPGタイマレジスタ(PTMRn)

PTMRHn

bit	15	14	13	12	11	10	9	8
	D15	D14	D13	D12	D11	D10	D9	D8
属性:	R	R	R	R	R	R	R	R
初期値:	1	1	1	1	1	1	1	1

PTMRLn

bit	7	6	5	4	3	2	1	0
	D7	D6	D5	D4	D3	D2	D1	D0
属性:	R	R	R	R	R	R	R	R
初期値:	1	1	1	1	1	1	1	1

R : リードオンリ, 書込みは動作に影響なし

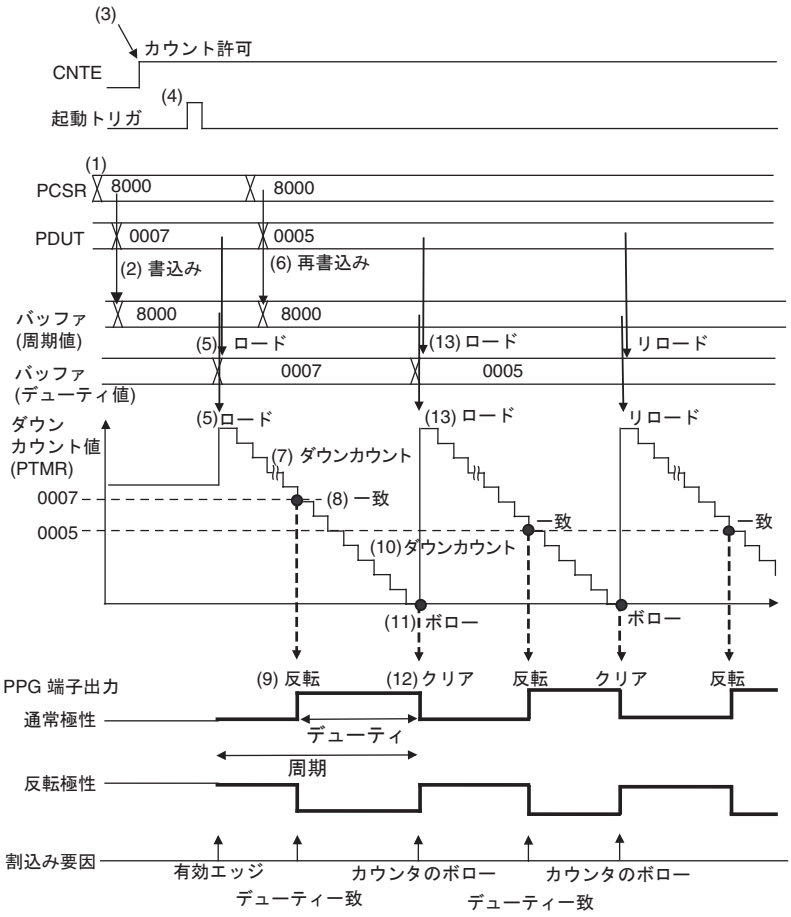
16 ビットダウンカウンタのカウント値を読み出すことができます。

16.3 PPGの動作

プログラマブルパルスジェネレータ(PPG)は，独立または連動して，プログラマブルなパルス出力を行います。各動作モードについて以下に説明します。

16.3.1 PWM動作

PWM 動作では , PPG 端子から可変デューティパルスが出力されます。



(1) 周期値の書込み

(2) デューティ値の書込み , バッファへの周期値の転送

(3) PPG の動作許可

(4) 起動トリガ発生

(5) 周期値とデューティ値のロード

(6) デューティ値の再書込み , バッファへの周期値の転送

(7) カウンタのダウンカウント

(8) ダウンカウント値とデューティ値の一致

(9) PPG 端子の出力レベルを反転

(10) カウンタのダウンカウント

(11) カウンタのボロー発生

(12) PPG 端子出力レベルのクリア (通常状態に復帰)

(13) 周期値のリロード

(14) デューティ値のリロード

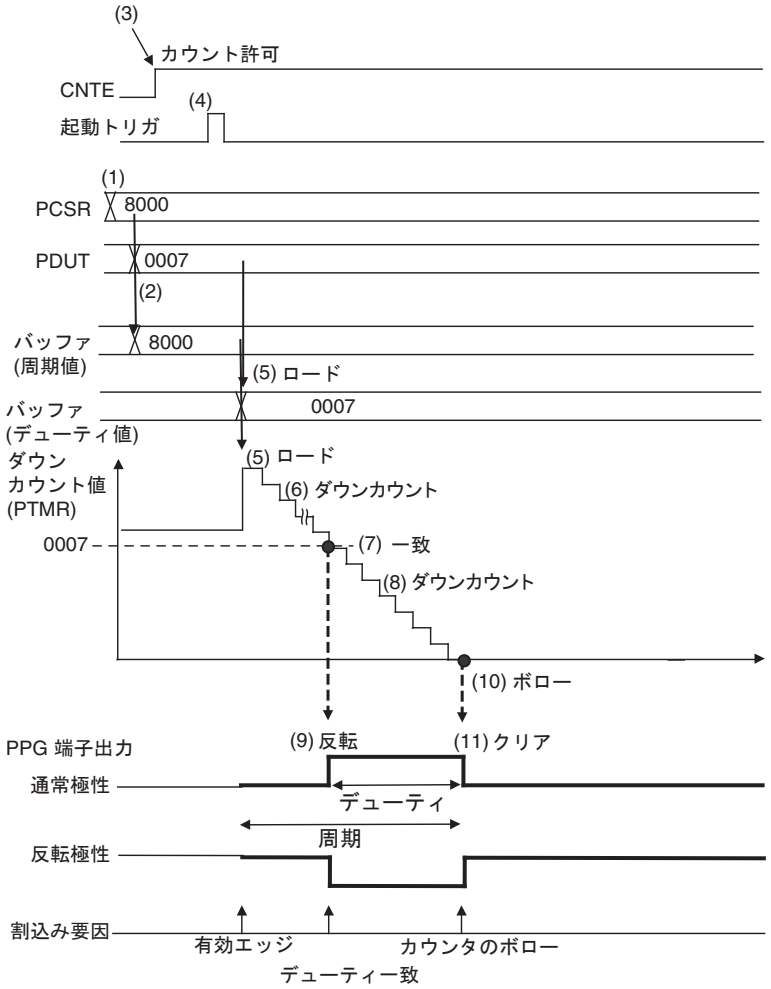
(15) (7) から (14) までを反復

計算式 :

- 周期 = {周期値(PCSR) + 1}×カウントクロック
- デューティ = {デューティ値(PDUT) + 1}×カウントクロック
- パルス出力までの幅 = {周期値(PCSR) - デューティ値(PDUT)}×カウントクロック

16.3.2 ワンショット動作

ワンショット動作では , PPG 端子からワンショットパルスが出力されます。



(1) 周期値の書込み

(2) デューティ値の書込み , バッファへの周期値の転送

(3) PPG の動作許可

(4) 起動トリガ発生

(5) 周期値とデューティ値のロード

(6) カウンタのダウンカウント

(7) ダウンカウント値とデューティ値の一致

(8) PPG 端子出力レベルの反転

(9) カウンタのダウンカウント

(10) カウンタのボロー発生

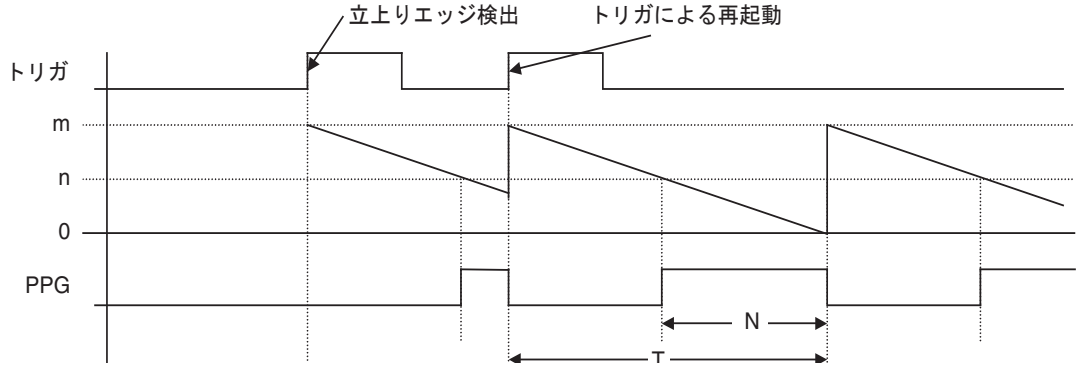
(11) PPG 端子の出力レベルのクリア (通常状態に復帰)

(12) 動作シーケンス終了

16.3.3 再起動動作

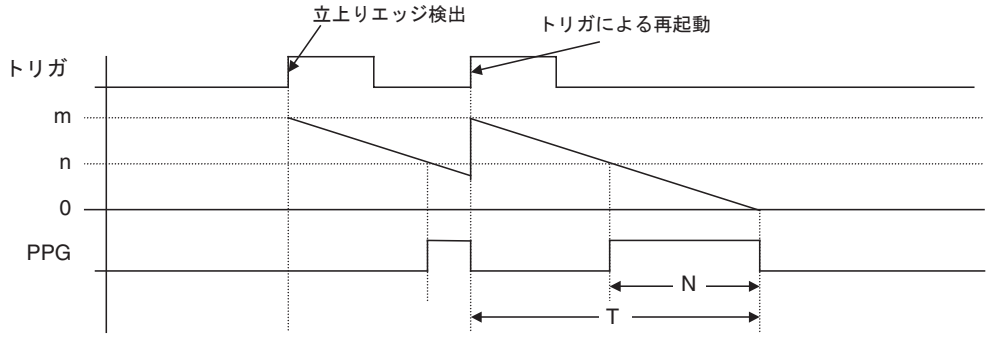
再起動時の動作を次に示します。

PWM動作中の再起動:



N = デューティ , T = 周期

ワンショット動作中の再起動:



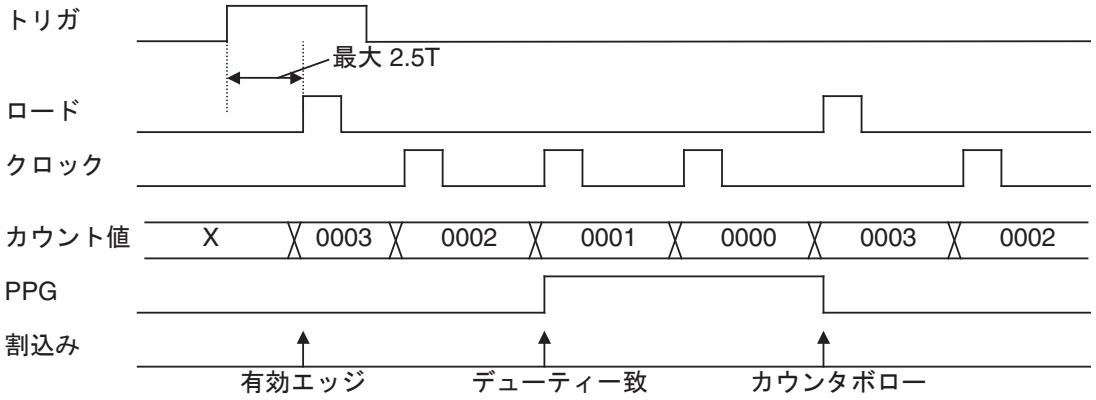
PWM 動作とワンショット動作のいずれでも,再起動がディスエーブルな場合には 2 番目以降のトリガは動作に影響しません (ダウンカウンタのシャットダウン後の 2 番目以降のトリガは機能します)。

16.4 使用上の注意

PPG を使用する時の注意事項を説明します。

使用上の注意

- 割込み要求フラグ(PCN:IRQF)が "1" になったときと同時に割込み要求フラグを "0" に設定すると, 割込み要求フラグへの"1"の設定が,フラグクリア要求より優先されます。
- 最初のロードは起動トリガの後, 最大 2.5T 遅延します。(T: カウントクロック)
  ダウンカウンタへのロードとカウントが同時に発生した場合には, ロード操作が優先されます。



- デューティ値 PDUT は, 周期 PCSR が初期化および書換えされた後に書き込んでください。
  (必ず, (1) PCSR, (2) PDUT の順序で書き込んでください。)
  PDUT は, デューティの書換えの場合にのみ書込みできます。
- デューティ値(PDUT)は周期値(PCSR)以下である必要があります。PCSR より大きな値を設定した場合には, デューティ値を小さな値に書き換える前に PPG の動作を禁止してください。
- PPG の周期設定レジスタ(PCSR)とデューティ設定レジスタ(PDUT)は, 必ずワード(16 ビット)フォーマットでアクセスしてください。これらのレジスタにバイトアクセスした場合は, 上側と下側のビット位置に値が書き込まれません。
- PPG を起動するには, PPG の動作を許可する前または同時にタイマ動作許可ビット(PCN:CNTE)に "1" を設定する必要があります。
- PPG の動作中には, モード値(MDSE), 再起動許可(RTRG), カウントクロック(CKS1, CKS0), トリガ入力エッジ(EGS[1:0]), 割込み要因(IRS), 内部トリガ(TSEL), 出力極性指定(OSEL)は変更しないでください。
  PPG の動作中にこれらの値を変更した場合には, レジスタをリロードする前に PPG の動作を禁止してください。
- GCN2 に値を書き込む際には, 上位 4 ビットの未定義部には "0" を書き込んでください。
  "1" を書き込んだ場合には, レジスタをリロードする前に PPG の動作を禁止してください。
- 起動トリガ選択ビット(TSEL0[3:0]), (TSEL1[3:0]), (TSEL2[3:0]), (TSEL3[3:0])に指定範囲外(0110, 0111, 1000～1111)の値を設定した場合は, レジスタを正常化するには PPG の動作を禁止した後に指定範囲内の値を書き
  込んでください。
- PPG の動作中にタイマ動作許可ビット(PCN:CNTE)に "0" を設定して PPGn を禁止した場合には, PPG はその状態(カウントおよび出力レベル)を維持したまま停止します。
  その後, タイマ動作許可ビット(PCN:CNTE)を "1" に設定して PPG の動作を許可すると, 停止した時点の状態から動作を再開します。

17. 外部割込み

外部割込みの機能と動作について説明します。

17.1 外部割込みの概要

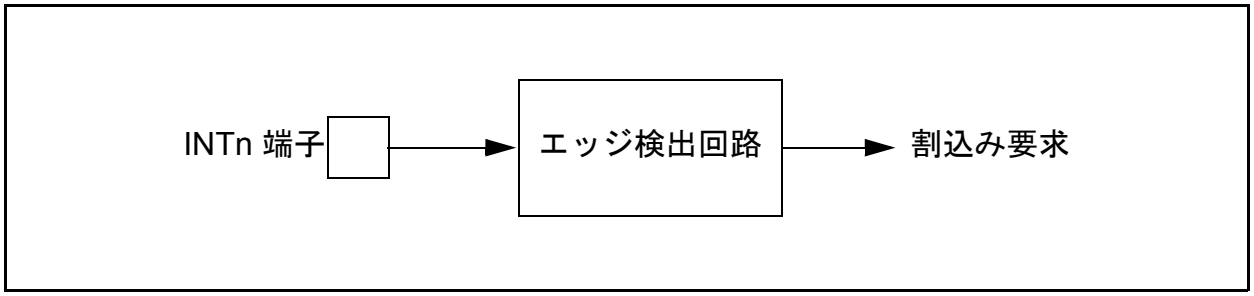
外部割込みモジュールは外部割込み端子に入力された信号を検出し, 割込み要求を発生します。

外部割込み

外部割込み要求は , "H", "L", 立上りエッジ , 立下りエッジの , 4 つの要求レベルがあります。

外部割込みのブロックダイヤグラム

Figure 17-1. 外部割込みのブロックダイヤグラム



外部割込みレジスタ

	bit	7	6	5	4	3	2	1	0	初期値
ENIR0		EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0	00000000_B
ENIR1	bit	7	6	5	4	3	2	1	0	
		EN15	EN14	EN13	EN12	EN11	EN10	EN9	EN8	00000000_B
EIRR0	bit	15	14	13	12	11	10	9	8	
		ER7	ER6	ER5	ER4	ER3	ER2	ER1	ER0	00000000_B
EIRR1	bit	15	14	13	12	11	10	9	8	
		ER15	ER14	ER13	ER12	ER11	ER10	ER9	ER8	00000000_B
ELVR0	bit	7	6	5	4	3	2	1	0	
		LB3	LA3	LB2	LA2	LB1	LA1	LB0	LA0	00000000_B
ELVR0	bit	15	14	13	12	11	10	9	8	
		LB7	LA7	LB6	LA6	LB5	LA5	LB4	LA4	00000000_B
ELVR1	bit	7	6	5	4	3	2	1	0	
		LB11	LA11	LB10	LA10	LB9	LA9	LB8	LA8	00000000_B
ELVR1	bit	15	14	13	12	11	10	9	8	
		LB15	LA15	LB14	LA14	LB13	LA13	LB12	LA12	00000000_B

17.2 外部割込みレジスタ

外部割込みモジュールには次のレジスタがあります。

- 外部割込み許可レジスタ(ENIRn: 外部割込み要求許可レジスタ)
- 外部割込みフラグ(EIRRn: 外部割込み要求レジスタ)
- 要求レベル設定レジスタ(ELVRn: 外部割込みレベルレジスタ)

外部割込み許可レジスタ(ENIRn: 外部割込み要求許可レジスタ)

Figure 17-2. 外部割込み要求許可レジスタ(ENIRn)

	bit	7	6	5	4	3	2	1	0	初期値
ENIR0		EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0	00000000_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
ENIR1	bit	7	6	5	4	3	2	1	0	
		EN15	EN14	EN13	EN12	EN11	EN10	EN9	EN8	00000000_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : リード/ライト可能

ENIRn レジスタには , 外部割込みモジュールの割込み許可ビットを設定します。ENx を " 許可 " に設定すると対応する外部割込み端子に対して割込みイベントが検出された際に割込み要求が発生します。これらのビットに "1" を設定すると割込み要求が許可され , "0" を設定すると禁止されます。

＜注意事項＞

外部割込みを " 許可 " に設定した場合は , 対応するポートのポート入力許可レジスタ (PIER) も設定してください (「13.2.4 " ポート入力許可レジスタ (PIERnn)"」参照)。

外部割込みフラグ(EIRRn: 外部割込み要求レジスタ)

Figure 17-3. 外部割込み要求レジスタ(EIRRn)

	bit	15	14	13	12	11	10	9	8	初期値
EIRR0		ER7	ER6	ER5	ER4	ER3	ER2	ER1	ER0	00000000_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
EIRR1	bit	15	14	13	12	11	10	9	8	
		ER15	ER14	ER13	ER12	ER11	ER10	ER9	ER8	00000000_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : リード/ライト可能

EIRRn レジスタは外部割込み要求を保持するレジスタです。

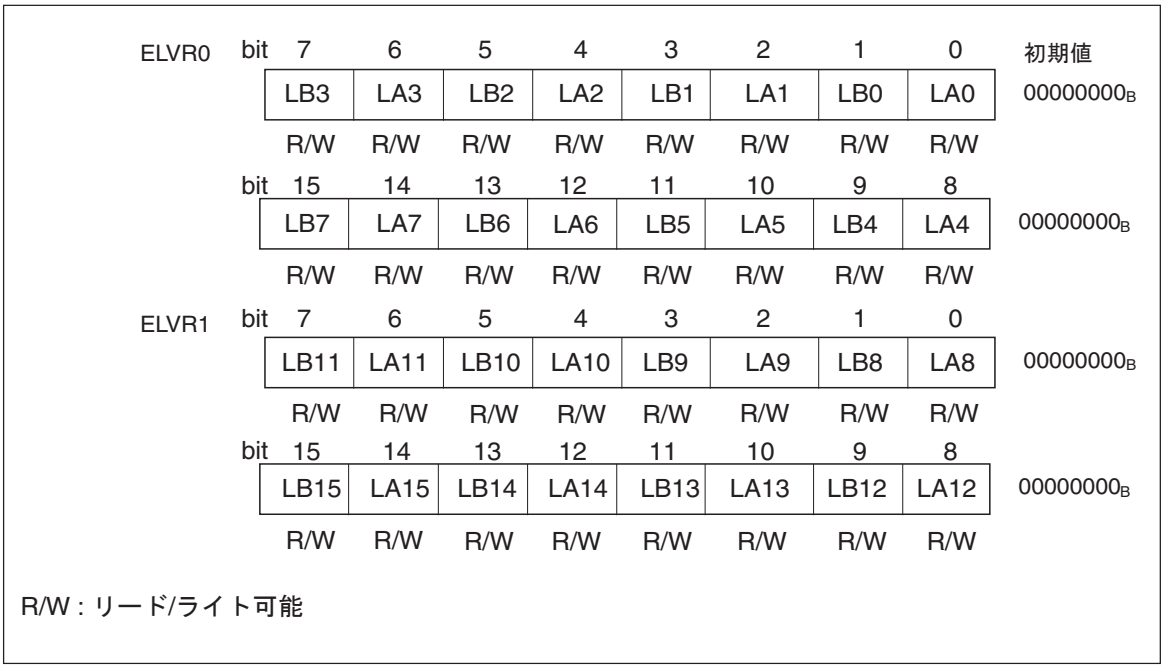
外部割込み端子に対して割込みイベントが検出されると , 対応する ERx フラグビットに "1" が設定されます。ENx ビットが "1" の場合には割込みコントローラに対して割込み要求が送られます。これらのビットに "1" を書き込んだ場合は動作に影響せず , "0" を書き込んだ場合はフラグがクリアされます。リードモディファイライト命令では , これらのビットからは常に "1" が読み出されます。

＜注意事項＞

外部割込みフラグをクリアする際には , 割込みの原因となったフラグのみをクリアしてください。

要求レベル設定レジスタ(ELVRn: 外部割込みレベルレジスタ)

Figure 17-4. 外部割込みレベルレジスタ(ELVRn)



ELVRn は外部割込み端子に対して要求イベントを定義します。各端子には, 表 17.2-1 に記載のように 2 つのビットが割り当てられます。要求イベントが入力レベルで検出される場合, 割込みフラグがソフトウェアでリセットされても, 入力が指定されたレベルである間は割込みフラグはセットされたままになります。

Table 17-1. 外部端子の割込み要求検出要因

LBx	LAx	割込み要求検出要因
0	0	"L" レベル端子入力
0	1	"H" レベル端子入力
1	0	立上りエッジ端子入力
1	1	立下りエッジ端子入力

17.3 外部割込み機能の使用上の注意

外部割込み機能を使用する際には , 次の点に留意する必要があります。

- DMA を使用するための外部回路の動作条件
- 割込みフラグのクリア
- 外部割込み要求レベル

DMAを使用するための外部回路の動作条件

DMA を使用する外部回路は , 要求された DMA 転送が実行された後に要求信号をインアクティブにする必要があります。DMA 転送を開始するための外部割込みの適切な使用法については , アプリケーションノート MCU-AN-300203-e を参照してください。

割込みフラグのクリア

- 割込みフラグは割込み処理ルーチン内でクリアされる必要があります。さもなければ, 最初の割込み処理の終了後に再び同じ処理が実行されます。
- イベント入力としてレベル検出が設定されている場合,入力端子にアクティブレベルが保持し続けている, 割込みフラグをクリアしても再度セットされます。この場合は,外部要因をクリアするかまたは割込み許可ビットをクリアする必要があります。

外部割込み要求レベル

- イベント入力としてエッジ検出が設定されている場合,入力エッジはノイズフィルタによりフィルタされない最低パルス幅以上である必要があります。最低パルス幅長についてはデータシートを参照してください。
- イベント入力としてレベル検出が設定されている場合,一度指定されたレベルが入力されると, 入力信号がインアクティブレベルに変化した後も割込みフラグはアクティブの状態のままになります(Figure 17-6 参照)。割込み要求をクリアするためには, 割込みフラグをクリアする必要があります。

Figure 17-5. レベル検出での割込み要因レジスタのクリア

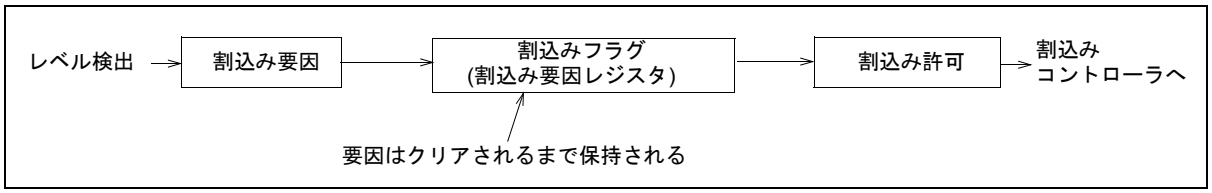
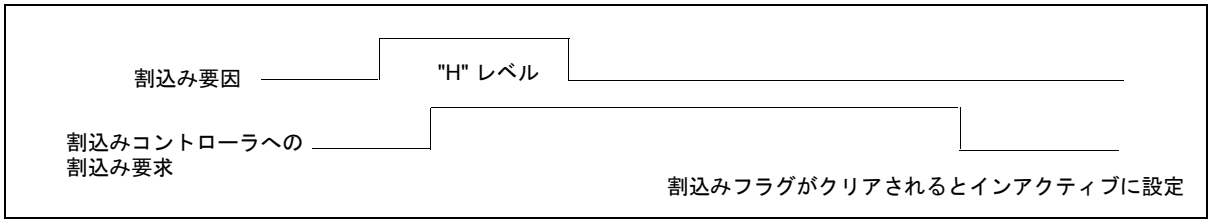


Figure 17-6. 割込みが許可されている場合の割込み要因と割込みコントローラへの割込み要求



18. A/Dコンバータ

A/D コンバータの機能と動作について説明します。

18.1 A/Dコンバータの概要

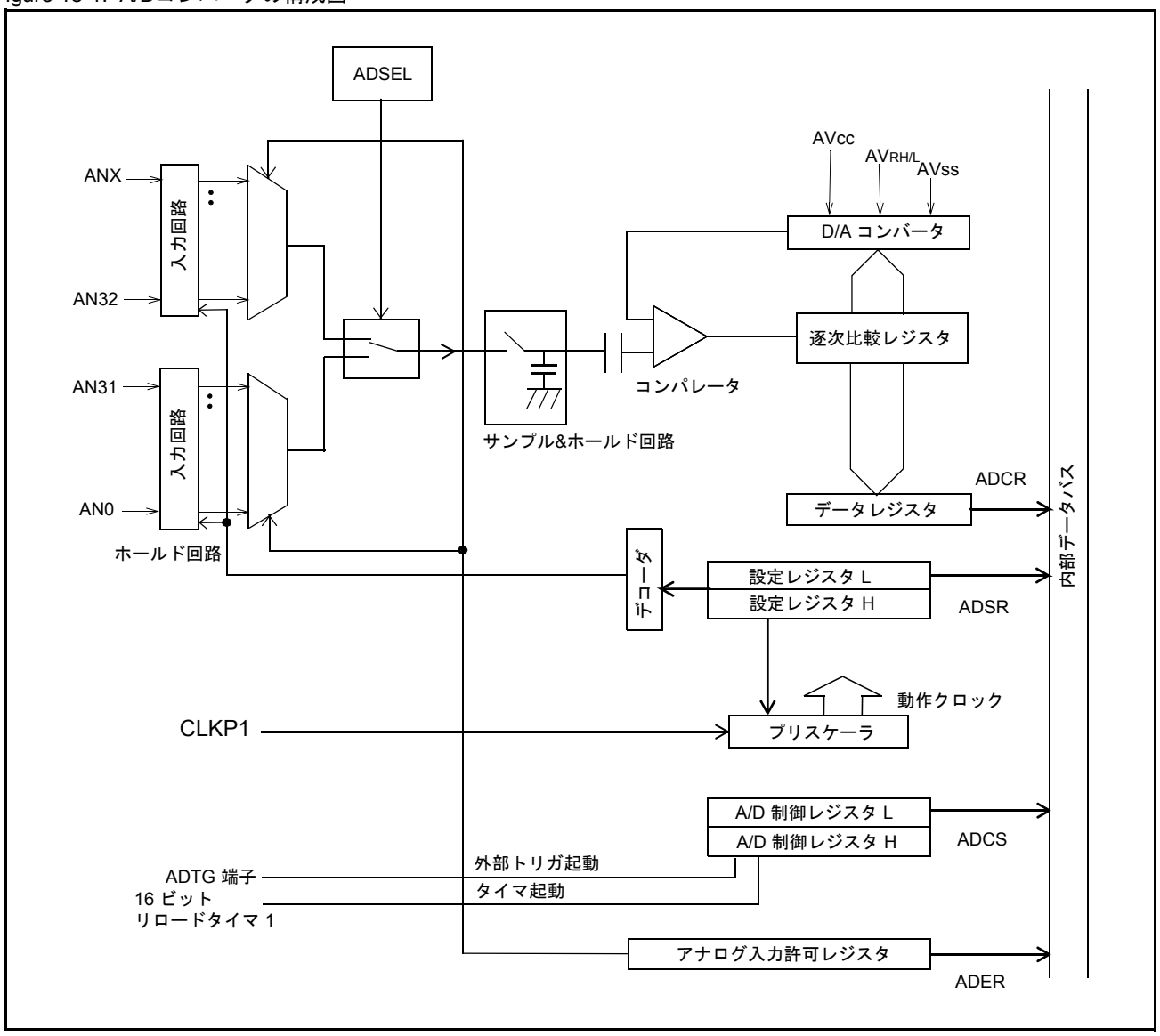
A/D コンバータはアナログ入力電圧をデジタル値に変換します。

A/Dコンバータの概要

- 変換時間: 1 チャネルにつき最小 1.9 ms (マシンクロックが 24 MHz の場合)
- サンプル&ホールド回路付き RC 型逐次変換比較方式
- 分解能: 10 ビットまたは 8 ビット
- ソフトウェア選択可能なアナログ入力端子(チャネル)
  - シングル変換動作:1チャネルを選択して変換します。
  - スキャン変換動作:連続した複数のチャネルを変換します。
- モード設定
  - 単発変換モード:指定チャネルを1サイクル変換します。
  - 連続変換モード:指定チャネルを繰り返し変換します。
  - 停止変換モード:指定チャネルを変換したら一時停止して次の起動が掛かるまで待機します(変換開始の同期が可能)。
- 割込み要求
  - A/D 変換終了時には,割込みコントローラ にA/D 変換終了の割込み要求を発生します。この割込みによってDMAを起動し, AD変換の結果をメモリに転送します。
- 選択可能な起動要因
  - 起動要因
    ソフトトリガ
    リロードタイマ1
    外部トリガ(立下りエッジ)

A/Dコンバータの構成

Figure 18-1. A/Dコンバータの構成図



＜注意事項＞

使用する端子をアナログ入力に設定にするには , アナログ入力許可レジスタ (ADER) のビットを "1" に設定します。

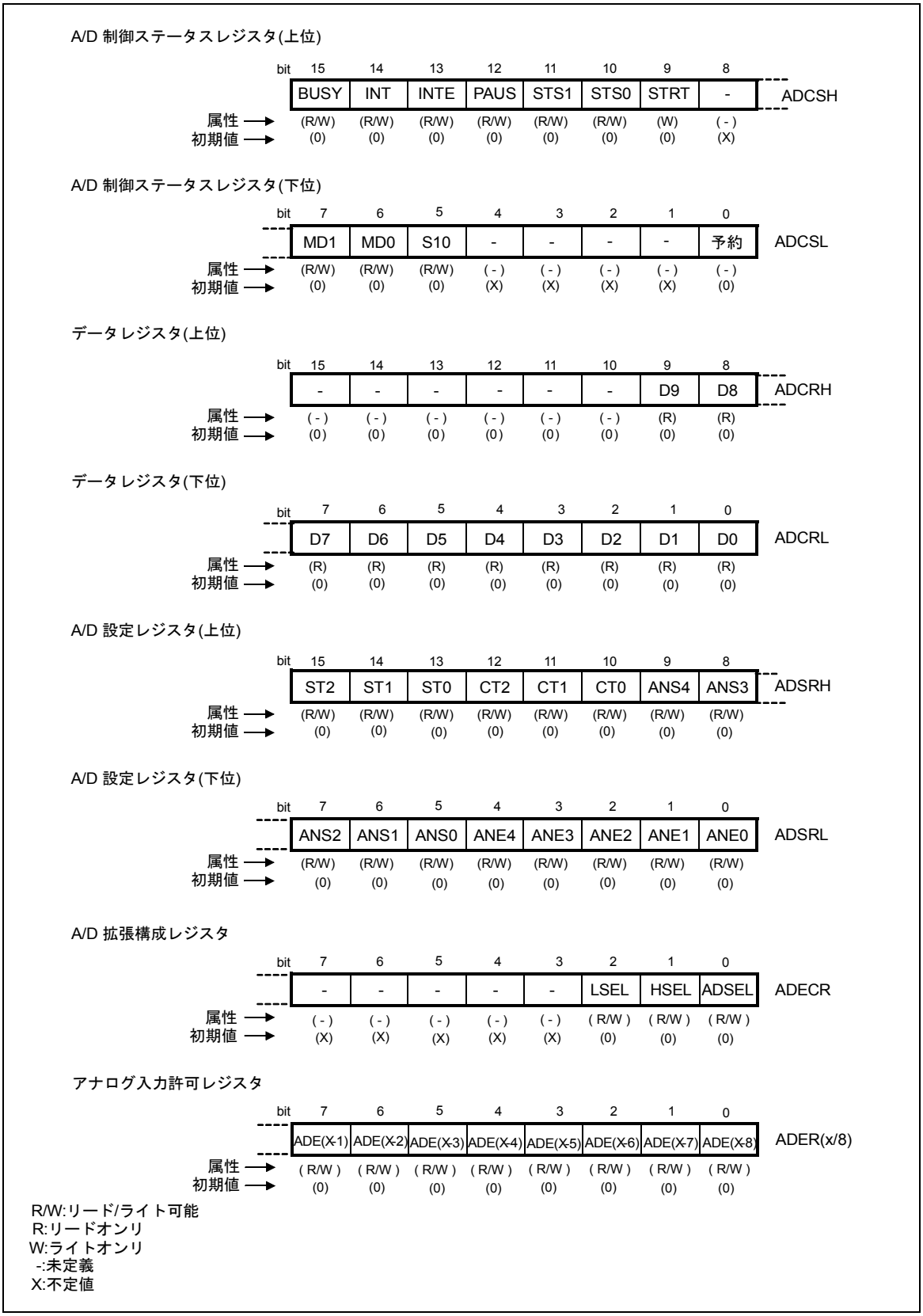
18.2 A/Dコンバータのレジスタ

A/D コンバータには次のレジスタがあります。

- 制御ステータスレジスタ: ADCS
- データレジスタ: ADCR
- 設定レジスタ: ADSR
- 拡張構成レジスタ: ADECR
- アナログ入力許可レジスタ: ADERx

A/Dコンバータのレジスタ

Figure 18-2. A/Dコンバータのレジスタ

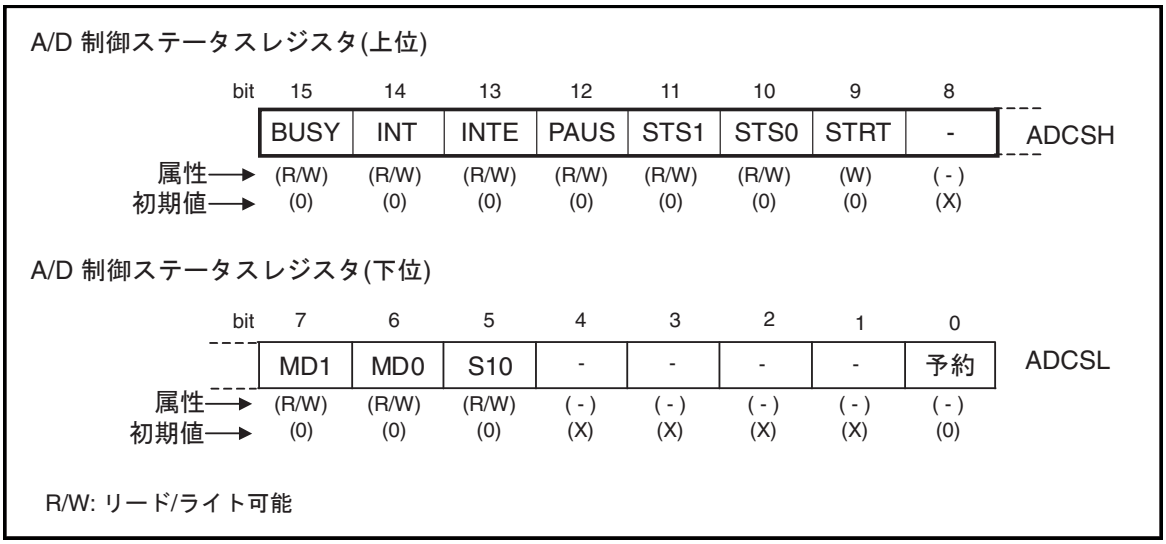


18.2.1 A/D制御ステータスレジスタ(ADCS)

A/Dコンバータの制御及びステータス表示を行うレジスタです。A/D変換中にADCSを書き換えないでください。

18.2.1.1 A/D制御ステータスレジスタ(ADCS)

Figure 18-3. A/D制御ステータスレジスタ(ADCS)



18.2.1.2 制御ステータスレジスタ(ADCSH)

[bit 15] BUSY (強制停止指示/動作確認)

- 読出し操作:
  - A/D コンバータの動作状態を示します。
  - A/D 変換の開始時にセットされ, 終了時にクリアされます。
- 書込み操作:
  - A/D 変換中に "0" を書き込むと, 変換は強制的に停止されます。
  - "1" の書込みは動作に影響しません。

＜注意事項＞

- **リードモディファイライト (RMW) 系命令では "1" が読み出されます。**
- **単発変換モードでは A/D 変換終了でクリアされます。**
- **連続変換モード , 停止変換モードでは , "0" を書き込むまで A/D 変換は停止しません。**
- **リセットで "0" に初期化されます。**
- **強制停止と起動を同時に行わないでください (ソフトウェア (BUSY = 0, STRT = 1), 外部トリガ , タイマ)。**

[bit 14] INT (割込み)

A/D 変換結果が A/D データレジスタ (ADCR) に格納されると INT ビットに "1" がセットされます。

割込み要求を許可した場合に (INTE=1) 割込み要求フラグビットがセットされると (INT=1) 割込み要求が発生します。DMA が許可されている場合には , DMA が起動されます。"1" の書込みは動作に影響しません。"0" の書込みまたは DMA からの割込みクリア信号でクリアされます。

＜注意事項＞

- "0" 書込みでクリアする場合は , A/D 変換中でないことを確認してください。
- リセットで "0" に初期化されます。
- リードモディファイライト (RMW) 系命令では "1" が読み出されます。

[bit 13] INTE (割込み許可)

割込みを許可および禁止します。

- 0: 割込み禁止
- 1: 割込み許可

DMA を使用する場合は , "1" にセットします (DMA は割込み要求によって起動)。

リセットで "0" に初期化されます。

このビットはリセットで "0" に初期化されます。

[bit 12] PAUS (A/D変換一時停止)

PAUS ビットは A/D 変換データ保護機能が有効であることを示します。PAUS ビットは割込み要求の出力を許可 (ADCS:INTE=1) に設定している場合のみ有効です。
A/D 変換データ保護機能が働いた場合 , "1" にセットされます。

"0" を書き込んだときにのみ , クリアできます。

"1" を書き込んで設定します。

割込みが許可された状態で (ADCS:INTE=1) に A/D 変換を実行した場合 , A/D 変換が終了して , 割込み要求フラグビット (ADCS:INT) が設定されると同時に , 割込み要求が発生します。割込み要求フラグビット (ADCS:INT) をクリアしないまま次の
A/D 変換が終了した場合 , 前データが上書き , 破壊されることを防ぐために A/D 変換動作が一時停止します (A/D 変換データ保護機能)。A/D 変換動作が一時停止されると , PAUS ビットは "1" に設定されます。

割込み要求フラグビット (ADCS:INT) がクリアされると , A/D 変換は一時停止状態を解除し , A/D 変換動作を再開します。

割込み要求フラグビット (ADCS:INT) は , "0" を書き込むことでクリアされます。また , DMA を使用して A/D 変換結果を A/D データレジスタから転送する場合 , A/D 変換結果の転送が完了したときに , DMA コントローラによって割込み要求フラグビット (ADCS:INT) がクリアされます。

＜注意事項＞

- 変換データ保護機能の詳細については , 「18.5 変換データ保護機能」を参照してください。
- 一時停止状態が解除されても ADCS:PAUS ビットは自動的にクリアされません。PAUS ビットに "0" を書き込んでクリアしてください。

[bit 11 and bit 10] STS1 and STS0 (起動トリガ選択)

リセットで00_B に初期化されます。
A/D 変換を起動する要因を選択します。

Table 18-1. 機能設定

STS1	STS0	機能
0	0	ソフトウェア起動
0	1	外部端子トリガ/ ソフトウェア起動
1	0	タイマ/ ソフトウェア起動
1	1	外部端子トリガ/タイマ/ソフトウェア起動

複数の起動要因があるモードでは , 先に発生した要因で起動します。A/D 変換中に設定を変更した場合には , 結果は即座に反映されます。そのため , A/D 変換中の設定変更は推奨しません。

＜注意事項＞

- 外部端子トリガは立下りエッジで検出されます。
  外部トリガ入力レベルが "L" のときに外部端子トリガを設定すると , A/D 変換が開始されます。
- タイマを選択した場合には, 16 ビットリロードタイマ 1 が選択され, タイマの出力が "1" になったときに変換が起動されます。

[bit 9] STRT (開始)

"1" を書き込むと A/D 変換を開始します。

A/D 変換を再起動するには , 再度 "1" を書き込みます。

リセット時には , "0" に初期化されます。

常に "0" が読みだされます。

変換中の再起動は単発変換モード

1 でのみ可能であり , 単発変換モード 2, 連続変換モード , 停止変換モードはサポートしていません。後者の 3 つのモードでは , "1" を書き込む前に BUSY ビットを確認してください。

ソフトウェアで , A/D 変換の強制停止と起動を同時に行わないでください (BUSY=0, STRT=1)。

[bit 8] 未使用ビット

常に "0" を書き込んでください。

このビットを読み出した場合には不定値が返されます。

リードモディファイライト系命令は無効です。

18.2.1.3 制御ステータスレジスタ(ADCSL)

[bit 7, bit 6] MD1, MD0 (A/D変換モード設定)

Table 18-2. 動作モード設定

MD1	MD0	動作
0	0	単発変換モード 1 (A/D 変換中の再起動可能)
0	1	単発変換モード 2 (A/D 変換中の再起動不可)
1	0	連続変換モード(A/D 変換中の再起動不可)
1	1	停止変換モード(A/D 変換中の再起動不可)

- 単発変換モード
  開始チャネル(ANS4～ANS0)から終了チャネル(ANE4～ANE0)までを連続してA/D変換していきます。終了チャネルのA/D 変換が終了すると, A/D 変換動作を停止します。
- 連続変換モード
  開始チャネル(ANS4 ～ ANS0)から 終了チャネル(ANE4 ～ ANE0)までの A/D 変換を繰り返し実行します。
- 停止変換モード
  開始チャネル(ANS4 ～ ANS0)から 終了チャネル(ANE4 ～ ANE0)までの A/D 変換を実行しますが,チャンネルごとにA/D変換が終了すると一時停止します。起動すると再開します。

＜注意事項＞

- 連続変換モードまたは停止変換モードの A/D 変換は , BUSY ビットによって停止されるまで続きます。
- A/D 変換を停止するには BUSY ビットに "0" を書き込んでください。
- 単発変換 2, 連続変換モード , 停止変換モードでは再起動はできません。これはすべての起動に適用されます。

[bit 5] S10

変換の分解能を指定します。このビットに "0" が書き込まれると , 10 ビット A/D 変換が実行されます。このビットに "1" が書き込まれると 8 ビット A/D 変換が実行され , 変換結果は D7 ～ D0 に格納されます。

[bits 4 ～ bit 1] 未定義ビット

これらのビットへの書込みは動作に影響しません。これらのビットを読み出すと常に "1" が返されます。

[bits 0] 予約 (予約ビット)

予約ビットです。常に "0" を書き込んでください。

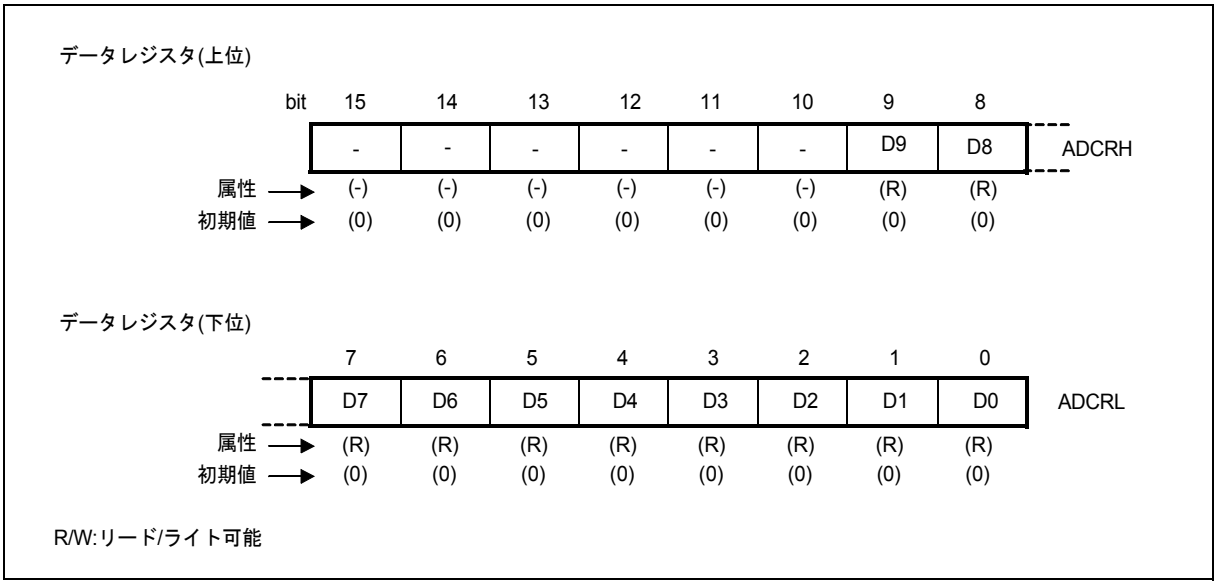
このビットを読み出すと常に "0" が返されます。

18.2.2 データレジスタ(ADCR)

データレジスタ(ADCR)には変換結果のデジタル値が格納されます。このレジスタの値は変換が終了するたびに書き換えられます。通常, 最後の変換結果値がレジスタに格納されます。

18.2.2.1 データレジスタ(ADCR)

Figure 18-4. データレジスタ(ADCR)



ADCRH の bit10 ～ bit15 を読み出すと常に "0" が返されます。

ADCSL の S10 ビットが "1" の場合には 8 ビットモードが選択され , 変換されたデータは bit7 ～ bit0 に格納されます。この場合 , bit9 , bit8 を読み出すと常に "0" が返されます。

変換データ保護機能の詳細については , 「18.5 変換データ保護機能」を参照してください。

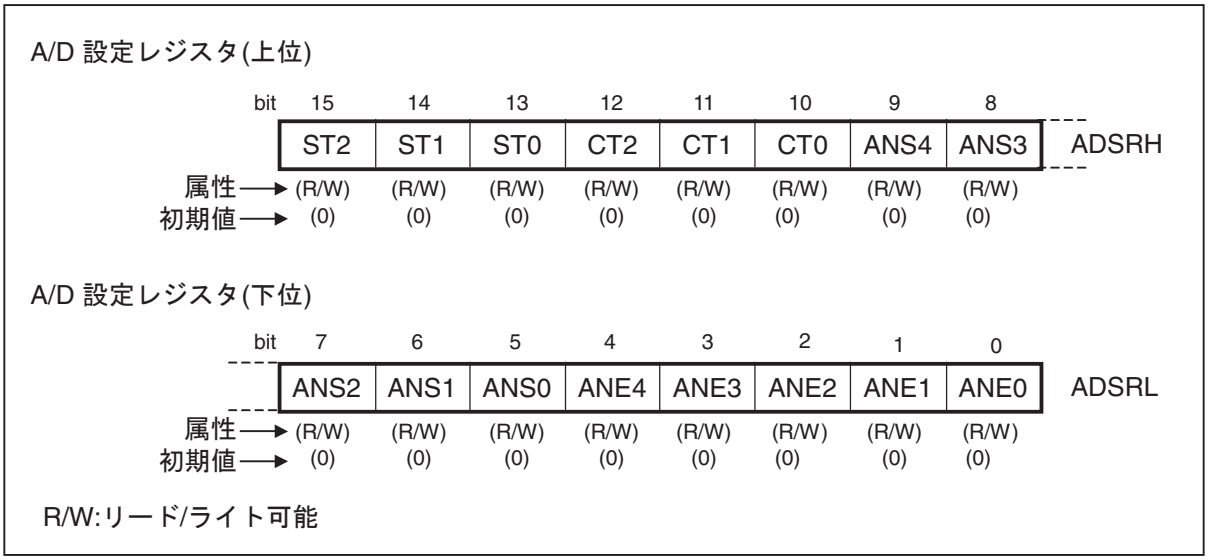
このレジスタには書込みを行わないでください。

18.2.3 A/D設定レジスタ(ADSR)

A/D設定レジスタ(ADSR)は, A/D 変換時間とサンプリングチャネルの設定, および現行サンプリングチャネルの表示に使用されます。

18.2.3.1 A/D設定レジスタ(ADSR)

Figure 18-5. A/D設定レジスタ(ADSR)



[bit 15 ～ bit 13] ST2, ST1, ST0 (サンプリング時間)

Table 18-3. 機能設定

ST2	ST1	ST0	機能
0	0	0	4 マシンクロック (8 MHz, 0.5 µs)
0	0	1	6 マシンクロック(8 MHz, 0.75 µs)
0	1	0	8マシンクロック(16 MHz, 0.5 µs)
0	1	1	12マシンクロック (24 MHz, 0.5 µs)
1	0	0	24 マシンクロック(8 MHz, 3 µs)
1	0	1	36 マシンクロック(16 MHz, 2.25 µs)
1	1	0	48マシンクロック(16 MHz, 3.0 µs)
1	1	1	128 マシンクロック (24 MHz, 5.3 µs)

アナログ入力信号のサンプリング時間を定義します。これらのビットの設定は , 次の条件を考慮して注意深く行う必要があります。 さもなければ , 結果の変換精度が保証されません。

[bit 12 ～ bit 10] CT2, CT1, CT0 (比較時間)

Table 18-4. 機能設定

CT2	CT1	CT0	機能
0	0	0	22マシンクロック(8 MHz, 2.8 µs)
0	0	1	33マシンクロック(16 MHz, 2.1 µs)
0	1	0	44マシンクロック(20 MHz, 2.2 µs)
0	1	1	66マシンクロック(24 MHz, 2.8 µs)
1	0	0	88マシンクロック(8 MHz, 11.0 µs)
1	0	1	132マシンクロック(16 MHz, 8.3 µs)
1	1	0	176マシンクロック(20 MHz, 8.8 µs)
1	1	1	264マシンクロック(24 MHz, 11.0 µs)

逐次比較変換の変換時間を定義します。設定は以下の条件に従う必要があります。さもなければ , 結果の変換精度が保証されません。

≧ 1.0µs (4.5V ≦ Avcc ≦ 5.5V の場合)

≧ 2.0µs (3.0V ≦ Avcc ＜ 4.5V の場合)

[bit 9 ～ bit 5] ANS4, ANS3, ANS2, ANS1, ANS0 (開始チャネル設定)

A/D 変換を開始するチャネルの設定 , および現行アナログ入力チャネルの表示に使用します。

A/D 変換が起動すると, 指定されたチャネルから A/D 変換を開始します。

Table 18-5. 開始チャネル設定

ANS4	ANS3	ANS2	ANS1	ANS0	ADECR:ADSEL=0の場合の開始チャネル	ADECR:ADSEL=1の場合の開始チャネル
0	0	0	0	0	AN0	AN32
0	0	0	0	1	AN1	AN33
0	0	0	1	0	AN2	AN34
0	0	0	1	1	AN3	AN35
					•	
					•	
					•	
1	1	1	0	0	AN28*	AN60*
1	1	1	0	1	AN29*	AN61*
1	1	1	1	0	AN30*	AN62*
1	1	1	1	1	AN31*	AN63*

*: デバイス上に存在する最上位ADCチャンネル番号については, 対応するデータシートを参照してください。

- 読出し動作

  A/D 変換のための現行アナログ入力チャネルが示されます。変換が停止された場合には , 最後のチャネルが示されます。

  新たな値が設定された後であっても , A/D 変換開始前には以前の変換チャネルが読み出されます。

- リセット時: 00000_B に初期化されます。

[bit 4 ～ bit 0] ANE4, ANE3, ANE2, ANE1, ANE0 (終了チャネル設定)

A/D 変換を終了するチャネルを設定します。

Table 18-6. 終了チャネル設定

ANE4	ANE3	ANE2	ANE1	ANE0	ADECR:ADSEL=0の場合の終了チャネル	ADECR:ADSEL=1の場合の終了チャネル
0	0	0	0	0	AN0	AN32
0	0	0	0	1	AN1	AN33
0	0	0	1	0	AN2	AN34
0	0	0	1	1	AN3	AN35
					・ ・ ・	
1	1	1	0	0	AN28*	AN60*
1	1	1	0	1	AN29*	AN61*
1	1	1	1	0	AN30*	AN62*
1	1	1	1	1	AN31*	AN63*

*: デバイス上に存在する最上位 ADC チャネル番号については,「第1章 "概要"」を参照してください。

■ リセット時には, これらのビットは00000_Bに初期化されます。

＜注意事項＞

- このレジスタに書き込む際は , 必ずワードアクセスを使用してください。このレジスタに対してバイト書込みまたはリードモディファイライトが実行されると , 意図しないチャネルから A/D 変換が開始される場合があります。
- ANE4 ～ ANE0 ビットと ANS4 ～ ANS0 ビットに同一のチャネルを書き込んだ場合は , 1 つのチャネルに対してのみ変換が行われます (単一チャネル変換)。
- 連続変換モードまたは停止変換モードにおいて , ANE4 ～ ANE0 で設定したアナログ終了チャネルの変換が終了した場合には , ANS4 ～ ANS0 で設定したアナログ開始チャネルに戻ります。
- ANS > ANE のチャネル設定をした場合には , 変換はアナログ開始チャネルから開始され , チャネル 31 に到達すると AN0 に戻って続けられ , アナログ終了チャネルで終了します。A/D コンバータのチャネル数が 32 以下の製品では , 変換結果が不正になる場合があります。これを抑止するには , ANS に ANE より大きな値を設定しないでください。
- 開始チャネル < チャンネル数が 32 ≧ を 32 に設定することはできません。このような設定が必要な場合は , 次の手順を使用してください。
  - ADECR:ADSEL = 0 に設定して , 開始チャネルからチャネル 31 までを変換します。
  - ADECR:ADSEL = 1 に設定してチャネル 32 から終了チャネルまでを変換します。
- ANEx ビットと ANSx ビットには, デバイス上に存在しないチャネル番号を設定しないでください。A/D コンバータのチャネル数を確認するには , 製品のデータシートを参照してください。

18.2.4 A/D拡張構成レジスタ(ADECR)

A/D拡張構成レジスタ(ADECR)は, A/D コンバータの入力チャネル 32 以降のチャネルを有効にするために使用します。
また,A/Dコンバータの「高位基準電圧」「低位基準電圧」は,AVRH+AVRL,AVRH+AVSS,AVRH2+AVSS間で切り換えることができます。基準電圧を切換えることでA/D コンバータのダイナミックレンジが調整可能です。

18.2.4.1 A/D拡張構成レジスタ(ADECR)

Figure 18-6. A/D拡張構成レジスタ(ADECR)

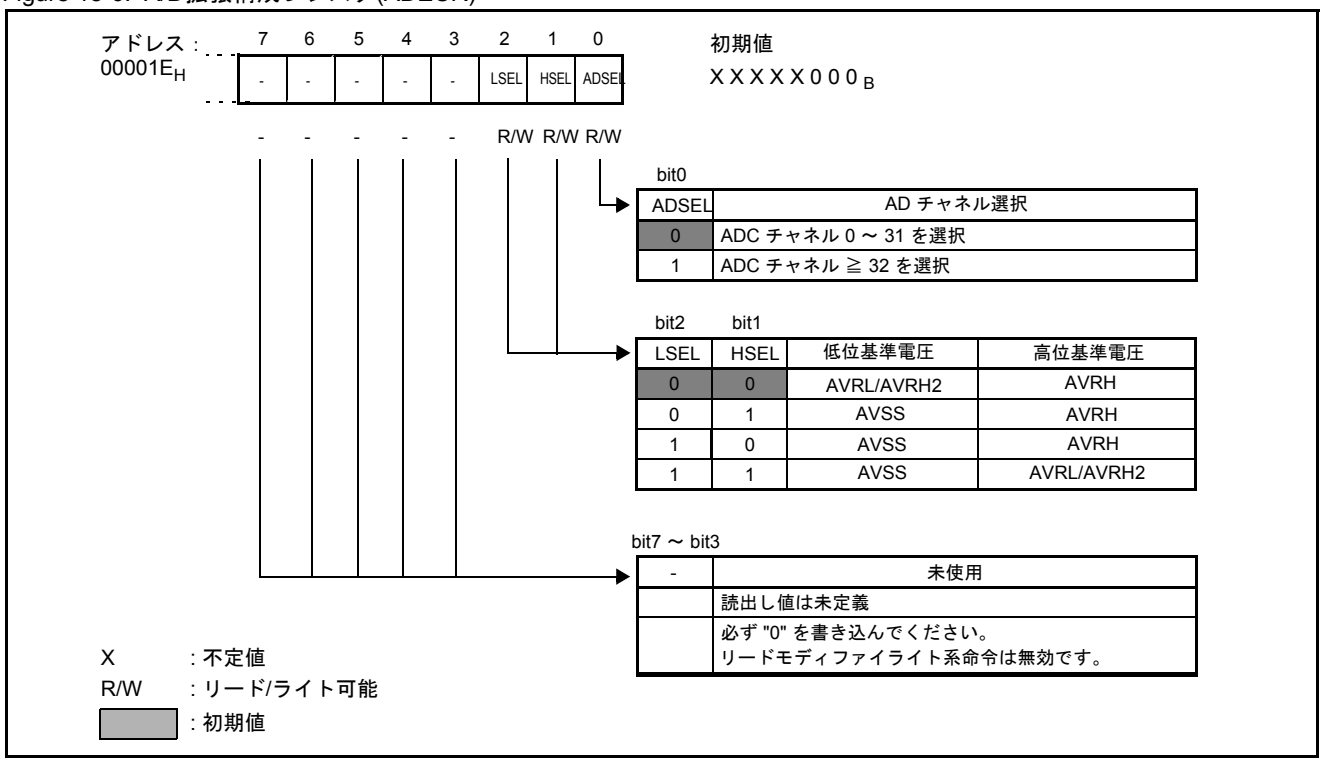
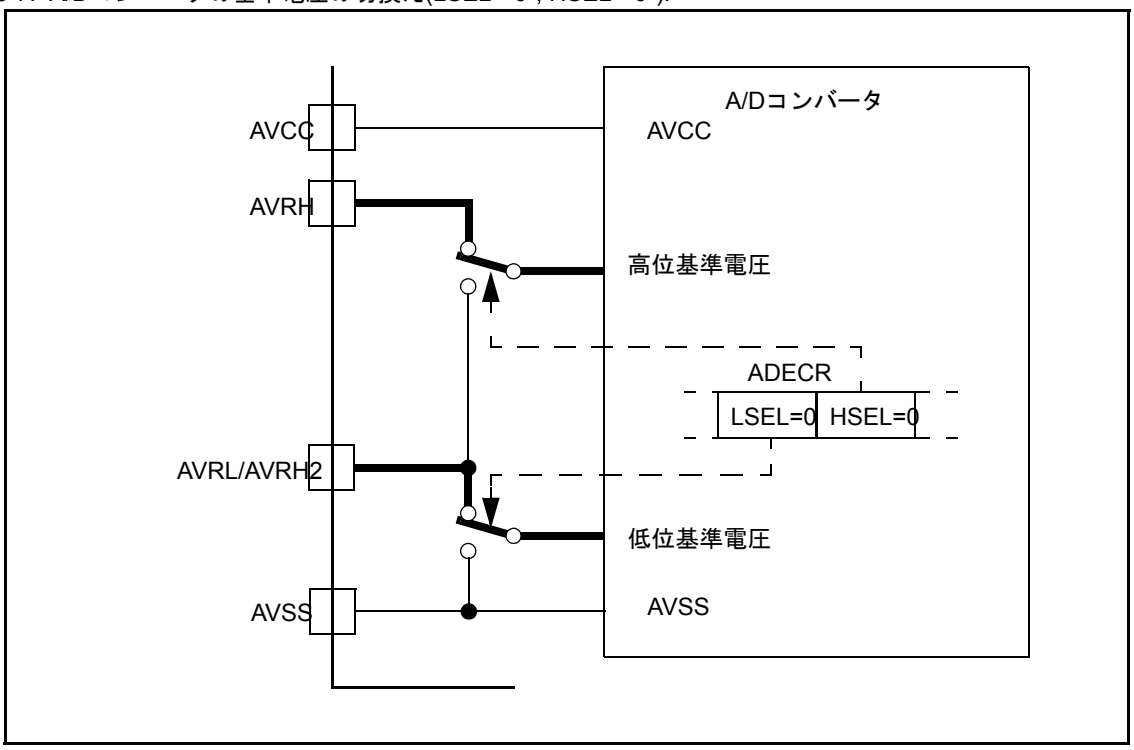


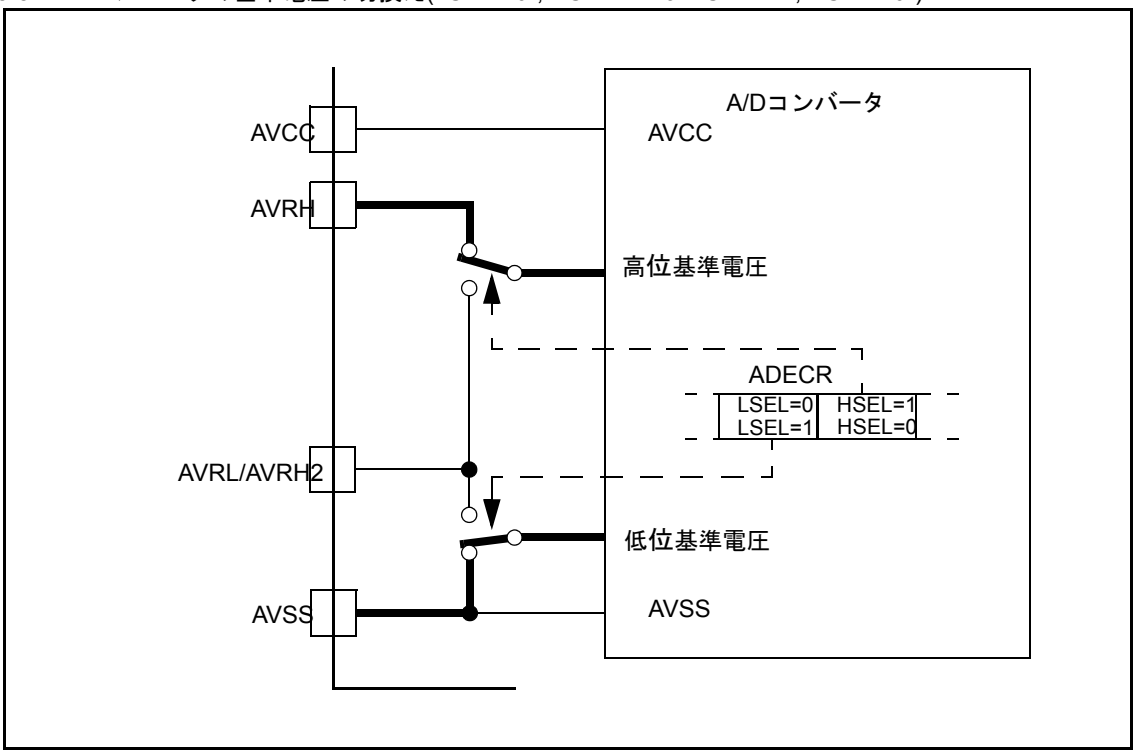
Figure 18-7. A/Dコンバータの基準電圧の切換え(LSEL="0", HSEL="0").



＜注意事項＞

本設定は , 16LX の A/D コンバータと互換性があります。A/D コンバータの低位基準電圧を AVSS の電位とは別のものにできます。

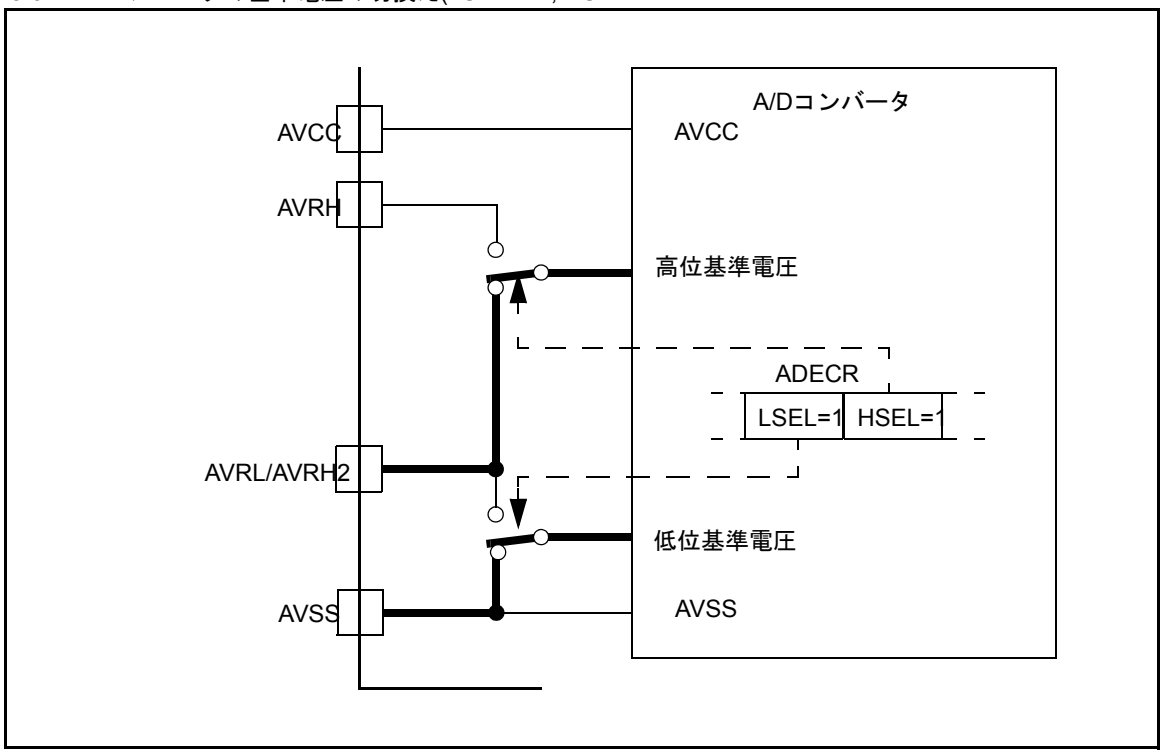
Figure 18-8. A/Dコンバータの基準電圧の切換え(LSEL="0", HSEL="1" or LSEL="1", HSEL="0")



＜注意事項＞

A/D コンバータの低位基準電圧が AVSS の電位 (システムグランドの電位など) と同等な場合に本設定を使用できます。

Figure 18-9. A/Dコンバータの基準電圧の切換え(LSEL="1", HSEL="1"



＜注意事項＞

A/D コンバータの低位基準電圧が AVSS の電位 (システムグランドの電位など) と同等な場合に , 本設定を使用できます。また , AVRH の電位とは違った高位基準電圧を供給するために , AVRL/AVRH2 端子を使用できます。

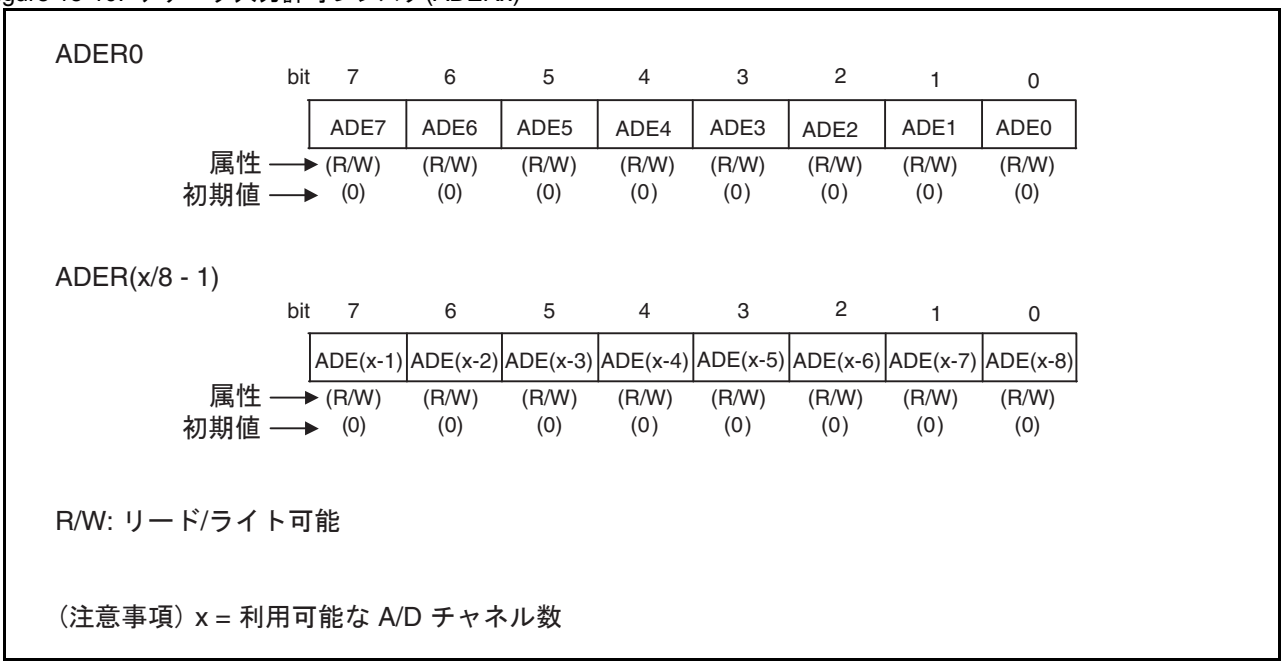
AVRH と AVRH2 間で A/D コンバータの高位基準電圧を切り換える際に必要な安定待ち時間については , データシートを参照してください。

18.2.5 アナログ入力許可レジスタ(ADERx)

このレジスタは A/D コンバータのアナログ入力を許可します。

18.2.5.1 アナログ入力許可レジスタ(ADERx)

Figure 18-10. アナログ入力許可レジスタ(ADERx)



[bit 7 ～ bit 0] ADEx (アナログ入力許可ビット)

外部端子を A/D コンバータ用のアナログ入力として使用する場合は , 対応するビットを "1" に設定する必要があります。

Table 18-7. モード選択

0	ポート入出力モード
1	アナログ入力モード

18.3 A/Dコンバータの動作

A/D コンバータは逐次比較変換方式で動作し, 変換の分解能は10 ビットと 8 ビットを選択できます。変換結果を格納するレジスタを　1つしか有しないため(10ビットの変換結果データレジスタADCR),レジスタは変換が終了するたびに書き換えられます。そのため, A/D コンバータ単独では連続変換モードに適さず, DMA による変換結果データのメモリへの転送を併用することを推奨します。

18.3.1 単発変換モード1 / 2

単発変換モードでは , ANS ビットで設定した開始チャネルから ANE ビットで設定した終了チャネルまでのアナログ入力が順次変換され , 終了チャネルの変換が終了すると A/D 変換は停止します。開始チャネルと終了チャネルが同一 (ANS = ANE) の場合には , ANS ビットで設定したチャネルの変換のみが行われます。

例 :

ANS = 00000_B, ANE = 00011_B:

開始 --> AN0 --> AN1 --> AN2 --> AN3 --> 終了

ANS = 00010_B, ANE = 00010_B:

開始 --> AN2 --> 終了

18.3.2 連続変換モード

連続変換モードでは , BUSY ビットに "0" が書き込まれるまで A/D 変換を繰り返します。

(BUSY ビットに "0" が書き込まれると , A/D 変換を強制停止します)。なお A/D 変換を強制停止すると , 変換途中で停止する場合があります。この場合 , 変換結果レジスタには前回変換が完了したときのデータが残ります。

18.3.3 停止変換モード

停止変換モードでは , ANS ビットで設定した開始チャネルから ANE ビットで設定した終了チャネルまでのアナログ入力を順次変換しますが , 1 つのチャネルの変換が終了するたびに変換が停止します。ANE ビットで設定した終了チャネルまで変換が終わると ANS のアナログ入力に戻り変換動作を続けます。開始チャネルと終了チャネルが同一 (ANS = ANE) の場合には , 1 チャネルの変換のみが行われます。

例 :

ANS = 00000_B, ANE = 00011_B:

開始 --> AN0 --> 停止 --> 開始 --> AN1 --> 停止 --> 開始 --> AN2 --> 停止 -->
開始 --> AN3 --> 停止　--> 開始 --> AN0 -->-->--> シーケンスを反復

ANS = 00010_B, ANE = 00010_B:

開始 --> AN2 --> 停止 --> 開始 --> AN2 --> 停止 --> 開始 --> AN2 -->-->--> シーケンスを反復

上記のシーケンスは , STS1 ビットと STS0 ビットで起動要因を設定する必要があります。

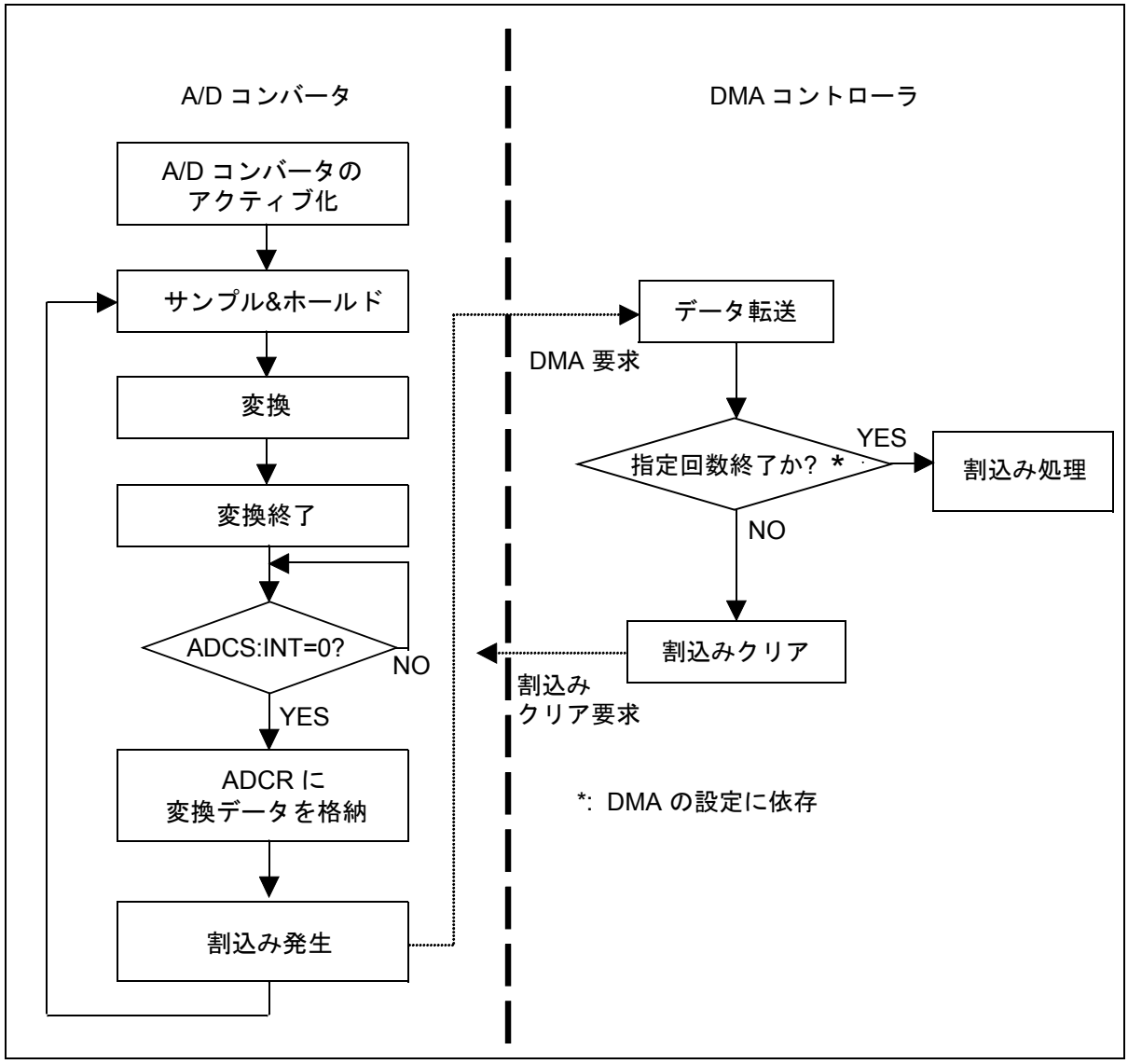
変換の開始は , 変換モードで同期させます。

18.4 DMAを使用した変換

Figure 18-11 に連続変換モードで変換したデータを転送する手順例を示します。

18.4.1 DMAを使用した変換

Figure 18-11. 連続変換モードで変換したデータを転送する手順例



18.5 変換データ保護機能

割込み要求出力を許可している場合に A/D 変換が行われると,データ保護機能が起動します。

18.5.1 A/D変換データ保護機能の説明

A/D 変換データ保護機能は , 一部の A/D 変換データの予期せぬ取得失敗を防止します。

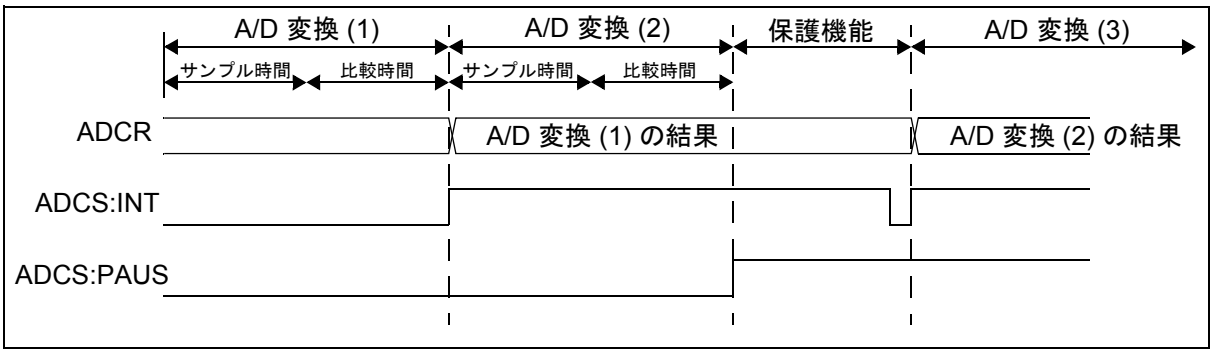
A/D コンバータには , 変換済みのデータを格納するデータレジスタ (ADCR) が 1 つと , A/D 変換中のデータを格納する逐次比較回路が 1 つあります。A/D 変換の実行中に , A/D コンバータは逐次比較回路に変換データを 1 ビットずつ格納し , A/D 変換の完了時にその A/D 変換結果をデータレジスタに格納します。

A/D 変換データ保護機能を使用した場合と , 使用しなかった場合の A/D コンバータの動作を以下に示します。

- 割込み許可ビット (ADCS:INTE) が設定されていない場合, データ保護機能は無効となります。この場合に A/D 変換が連続して実行されると, A/D コンバータは変換を終了するたびに, その変換結果をデータレジスタ (ADCR) に格納します。これは, 最新の変換データが常に格納されることを意味します。
- 割込み許可ビット (ADCS:INTE) が設定されている場合, データ保護機能は有効となります。この状態で A/D 変換が連続して実行されると, 最初の変換が完了したときに, 割込み要求フラグビットがADCS:INT=1 に設定されます。そして次の A/D 変換が実行されます。変換が ADCS:INT=1 の状態で終わると, A/D コンバータは変換データが上書きされないように, 逐次比較回路からの変換結果をデータレジスタ (ADCR) に転送する直前に, 「一時停止」状態に遷移します。制御ステータスレジスタの一時停止フラグビット (ADCS:PAUS) はこの時に "1" に設定されます。一時停止状態で割込み要求フラグビット (ADCS:INT) が "0" にクリアされると, 逐次比較回路に格納されたデータがデータレジスタに転送されます (Figure 18-12を参照)。

■

Figure 18-12. A/D変換データ保護機能の動作



18.5.2 A/D変換結果がCPUによって読み出される場合のデータ保護機能

- アナログ入力が A/D 変換された後, A/D 変換結果がデータレジスタ (ADCR) に格納されたときに,制御ステータスレジスタの割込み要求フラグビット (ADCS:INT) が "1" に設定されます。
- 前のA/D 変換の完了時にセットされた割込み要求フラグビット (ADCS:INT) が, 次の A/D 変換の終了時にも設定されたままの場合に, 割込み要求が許可されていれば (ADCS:INTE=1), A/D 変換動作は一時停止して, データレジスタへの新しいデータによって前のデータが上書きされるのを防止します。
- 制御ステータスレジスタ割込み要求が許可されていて(ADCS:INTE=1),INT ビットが設定されると割込み要求が発生します。INT ビットがクリアされると, A/D 変換動作の一時停止状態が解除されます。
- A/D 変換が連続して実行される場合, A/D コンバータは次の A/D 変換動作を開始します。この場合, 一時停止フラグビット (ADCS:PAUS) は自動的に "0" にはクリアされません。クリアするには, このビットに "0" を書き込みます。

＜注意事項＞

一時停止状態で割込み要求出力が禁止されている場合 (ADCS:INTE=0), A/D 変換が開始して , データレジスタ内のデータが書き換えられることがあります。

A/D 変換が 2 回以上連続して実行される場合 , 割込み要求フラグビット (ADCS:INT) をクリアする前に , データレジスタに格納されたデータを読み出します。A/D 変換が一時停止状態のときに , データレジスタに格納されたデータを読み出す前に割込み要求フラグビット (ADCS:INT) がクリアされると , 最初に格納された変換データは次の変換データによって上書きされ , 破壊されます。

18.5.3 A/D変換結果がDMAによって転送される場合のデータ保護機能

A/D 変換後, DMA を使用してデータレジスタから A/D 変換結果が転送されている最中に次の A/D 変換が終了すると , データレジスタへの新しいデータを上書きする直前に , データ保護のため , A/D 変換動作は一時停止します。A/D 変換動作が一時停止すると , 制御ステータスレジスタの一時停止フラグビット (ADCS: PAUS) が "1" に設定されます。

DMA を使用して A/D 変換結果の転送が完了すると , A/D 変換の一時停止状態は解除されます。A/D 変換が連続して実行される場合には , A/D 変換動作が再開されます。この場合 , 一時停止フラグビット (ADCS:PAUS) は自動的に "0" にはクリアされません。クリアするには , このビットに "0" を書き込みます。

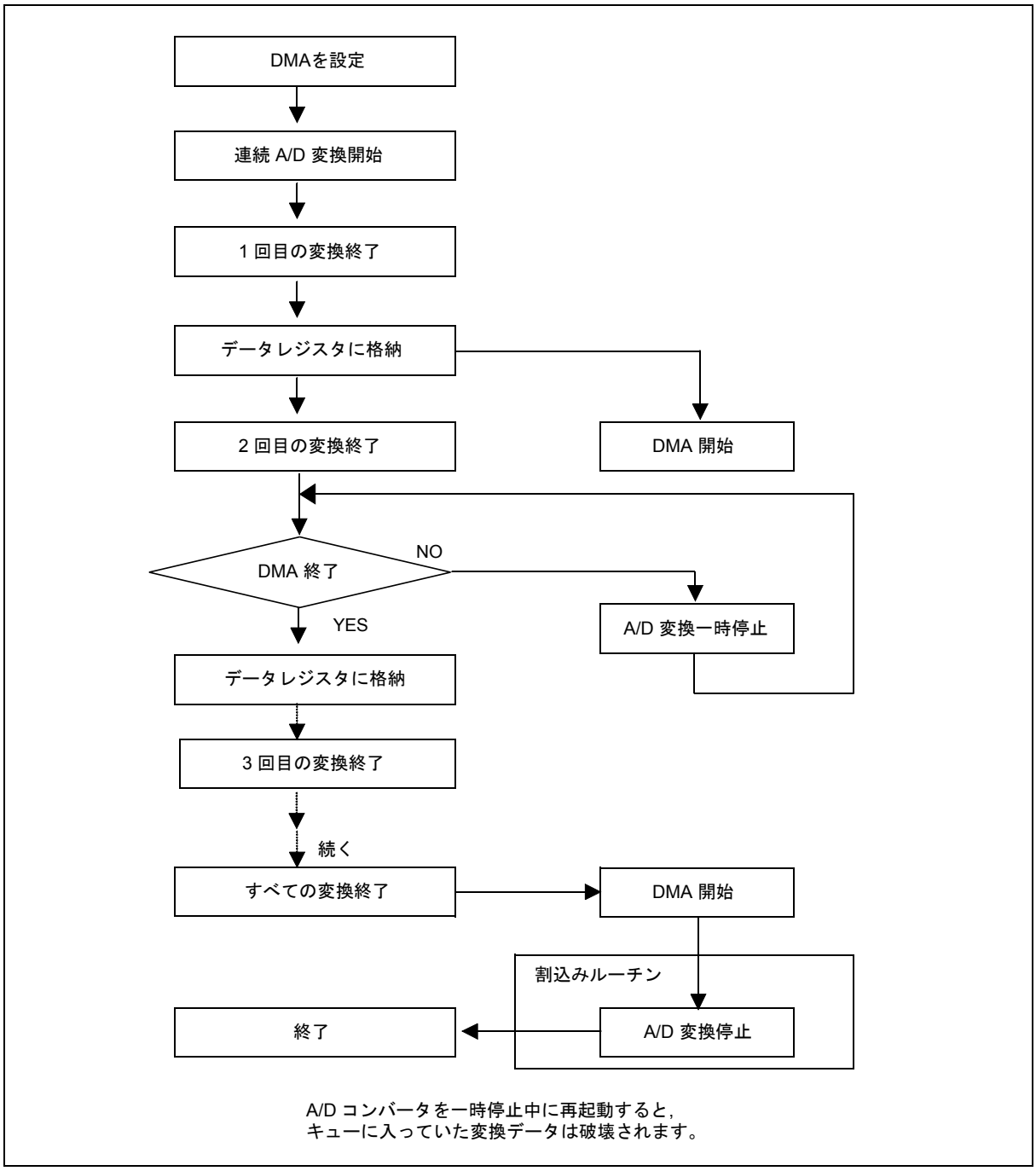
＜注意事項＞

DMA を使用して A/D 変換結果が転送される際は , CPU から割込み要求フラグビット (ADCS:INT=0) をクリアしないでください。クリアすると , 転送処理中のデータレジスタのデータが書き換えられることがあります。DMA を使用して A/D 変換結果が転送される際は , 割込み要求出力を禁止しないでください。一時停止状態で割込み要求出力が禁止されると (ADCS:INTE=0), A/D 変換が開始して , A/D データレジスタ内のデータが書き換えられることがあります。

DMA を使用して A/D 変換結果が転送されているときは , 再起動しないでください。変換の一時停止状態で A/D コンバータを再起動すると , 変換結果が破壊されることがあります。

18.5.4 変換データ保護機能のフロー例(DMA使用)

Figure 18-13. 変換データ保護機能のフロー例(DMA使用)



18.5.5 使用上の注意

A/D コンバータの起動要因として外部トリガまたは内部タイマを選択するには , ADCSH レジスタの A/D 開始要因選択ビット (STS1 ビットと STS0 ビット) を使用します。この場合には , 外部トリガまたは内部タイマの入力値が「非アクティブ」であることを保証してください。値が「アクティブ」の場合には , A/D 変換が即座に開始される場合があります。

STS1 ビットと STS0 ビットを設定する際には, 必ず ADTG 端子を "1" (入力)に設定し, 内部タイマ(リロードタイマ 1)を "0" (出力)に設定してください。

19. アラームコンパレータ

アラームコンパレータの機能と動作について説明します。

19.1 アラームコンパレータの概要

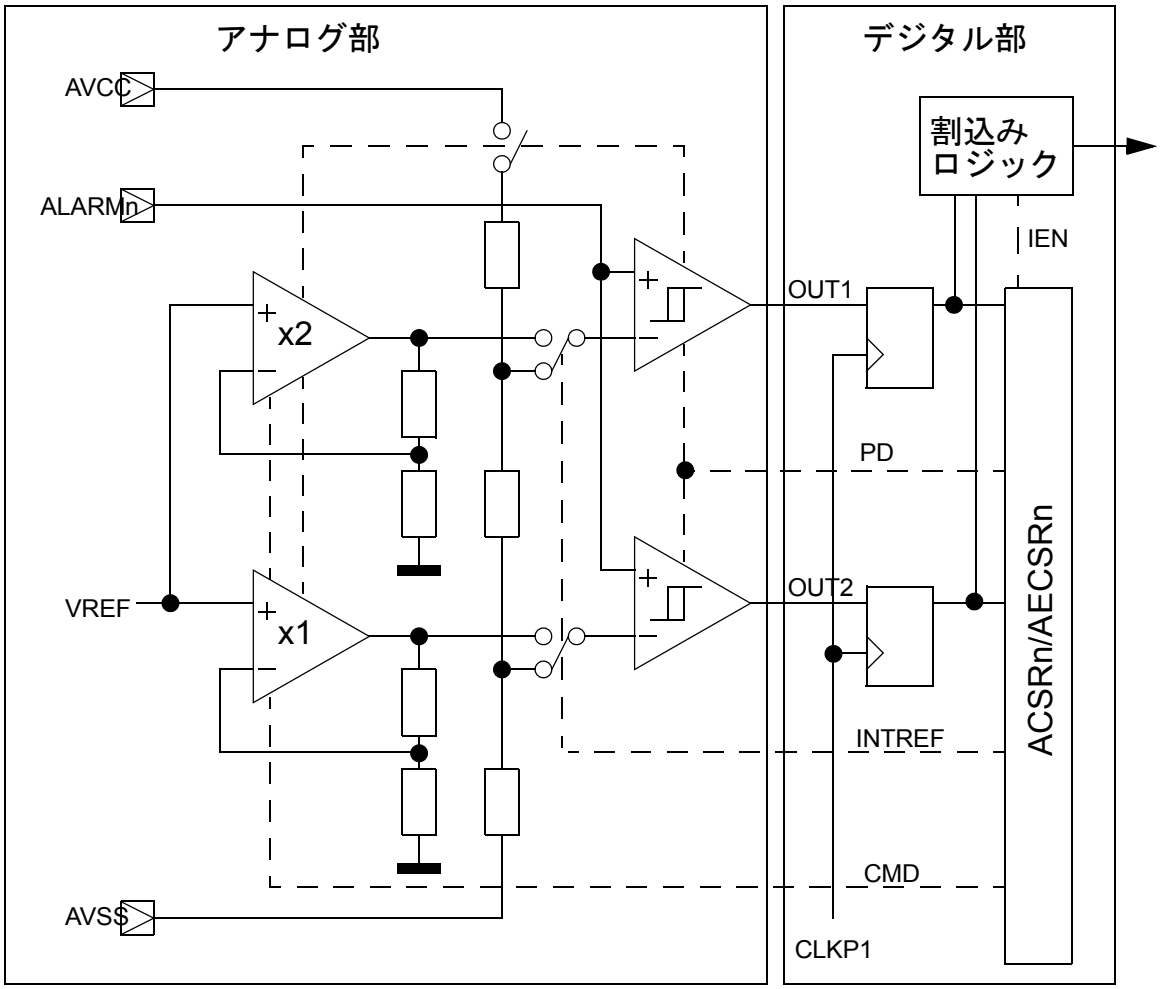
本章では, アラームコンパレータ(制限外電圧検出)の概要, レジスタ構成, アラームコンパレータのモードと動作を説明します。

アラームコンパレータの概要

- アラームコンパレータはアナログ部とデジタル部からなります。
- アナログ部は入力電圧をAVCC,AVSSまたは内部リファレンス電圧のVREFと比較し,デジタル出力信号OUT1と OUT2 をデジタル部のために生成します。
- 入力端子の電圧レベルが下位スレショルドより高い場合は, OUT2 が "H" レベルになります。
- 入力端子の電圧レベルが上位スレショルドより高い場合は, OUT1 が "H" レベルになります。
- デジタル部はOUT1 と OUT2 を使用して, 電圧が上位スレショルドより高い場合または下位スレショルドより低い場合に CPU への割込みを発生します。
- アラームコンパレータは ACSRnレジスタと AECSRn レジスタによって構成および制御できます。
- アラームコンパレータは, 消費電力を節減するために高速比較モードと低速比較モードを切り換えることができます。
- アラームコンパレータはパワーダウンモードに切り換えることができます。

アラームコンパレータのブロックダイヤグラム

Figure 19-1. アラームコンパレータのブロックダイヤグラム



＜注意事項＞

サフィックス n はアラームコンパレータモジュールの番号を表します。

19.2 アラームコンパレータのレジスタ

アラームコンパレータには次の 2 つのレジスタがあります。

- アラームコンパレータ制御/状態レジスタ(ACSRn)
- アラームコンパレータ拡張制御/状態レジスタ(AECSRn)

19.2.1 アラームコンパレータ制御/状態レジスタ(ACSRn)

Figure 19-2. アラームコンパレータ制御/状態レジスタの構成

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
	CMD	OVEN	UVEN	OUT2	OUT1	IRQ	IEN	PD	$011XXX00_B$
	R/W	R/W	R/W	R	R	R/W	R/W	R/W	

R/W: リード / ライト可能
R: リードオンリ

＜注意事項＞

サフィックス n はアラームコンパレータモジュールの番号を表します。

Bit 7: CMD (比較モード)

0	低速比較モード(低消費電力) [初期値]
1	高速比較モード(高消費電力)

消費電力およびアラームコンパレータの比較時間についてはデータシートを参照してください。

Bit 6: OVEN (上限外電圧許可)

0	過電圧時に割込みを発生しない
1	過電圧時の割込みを許可する [初期値]

Bit 5: UVEN (下限外電圧許可)

0	低電圧時に割込みを発生しない
1	低電圧時の割込みを許可する [初期値]

Bit 4: OUT2 (アラームコンパレータUV出力の同期出力)

0	アナログ入力電圧 < V_{EVTL} (AECSRn:INTREF = 0 の場合), または アナログ入力電圧 < V_{IVTL} (AECSRn:INTREF = 1 の場合)
1	アナログ入力電圧 > V_{EVTL} (AECSRn1:INTREF = 0 の場合), または アナログ入力電圧 > V_{IVTL} (AECSRn:INTREF = 1 の場合)

Bit 3: OUT1 (アラームコンパレータOV出力の同期出力)

0	アナログ入力電圧 < V_{EVTH} (AECSRn:INTREF = 0 の場合), または アナログ入力電圧 < V_{IVTH} (AECSRn:INTREF = 1 の場合)
1	アナログ入力電圧 > V_{EVTH} (AECSRn:INTREF = 0 の場合), または アナログ入力電圧 > V_{IVTH} (AECSRn:INTREF = 1 の場合)

Bit 2: IRQ (割込み要求)

0	低電圧または過電圧を検出しない
1	低電圧または過電圧を検出する

割込み要求をクリアするには IRQ に "0" を書き込みます。IRQ への "1" の書込みは動作に影響を与えません。リードモディファイライト命令が使用された際には , "1" が読み出されます。

＜注意事項＞

IRQ のクリアと同時にアラームコンパレータが割込みを要求した場合には , 割込み要求が高い優先度を持っており , IRQ は "1" に設定されます。電圧が永続的に範囲外の状況で割込み要因をクリアするには , 最初に低電圧割込みまたは過電圧割込みの発生を禁止し (OVEN = 0 または UVEN = 0, あるいはその両方), その後 IRQ ビットをクリアします。

Bit 1: IEN (割込み許可)

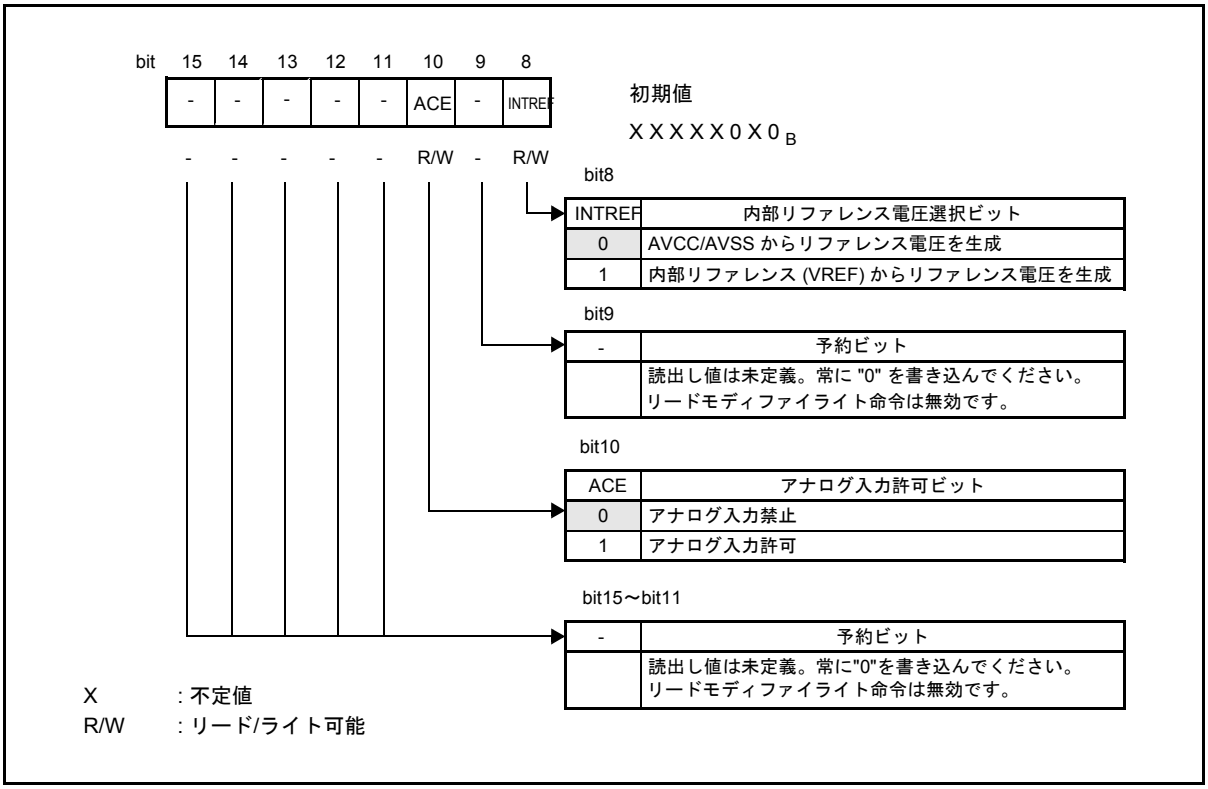
0	割込み発生禁止 [初期値]
1	割込み発生許可

Bit 0: PD (パワーダウン)

0	ランモード(アナログ部) [初期値]
1	パワーダウンモード (アナログ部)

19.2.2 アラームコンパレータ拡張制御/状態レジスタ(AECSRn)

Figure 19-3. アラームコンパレータ拡張制御/状態レジスタの構成



Bit 10: ACE (アナログ入力許可)

0	アナログ入力禁止
1	アナログ入力許可

Bit 8: INTREF (内部リファレンス電圧選択)

0	リファレンス電圧として AVCC/AVSS を選択[初期値]
1	内部リファレンス電圧 (VREF) を選択

19.3 アラームコンパレータの動作モード

アラームコンパレータ回路は割込みモードまたはポーリングモードで動作できます。内部割込みロジックは, IEN ビットの設定にかかわらず各割込みイベントを検出します。

割込みモード (IEN=1)

有効な割込みイベントを次表に示します。

Table 19-1. 有効な割込みイベント

INTREF	OUT2	OUT1	IRQ	アナログ入力電圧範囲
0	1	1	1	Vin > V_{EVTH} (過電圧)
0	1	0	0	V_{EVTL} < Vin < V_{EVTH} (通常動作)
0	0	0	1	Vin < V_{EVTL} (低電圧)
1	1	1	1	Vin > V_{IVTH} (過電圧)
1	1	0	0	V_{IVTL} < Vin < V_{IVTH} (通常動作)
1	0	0	1	Vin < V_{IVTL} (低電圧)

割込みビット IRQ は , 割込みイベント検出後の CLKP1 の次の立上りでセットされます。IEN=1 の場合には , CPU に対する割込み要求が発生します。両方の割込みが有効である場合 , アサートされている割込み要因を判断するには , 割込みサービスルーチン内で即座に ACSR レジスタを読み出す必要があります。OUT2 と OUT1 には常に電圧コンパレータ出力の実ステータスを保有しているため , 割込みトリガイベントは保存されていません。

ポーリングモード (IEN=0)

IRQ ビットはアクティブな割込みイベントでセットされ, ACSR レジスタへの書込みでリセットできます。ACSR を連続的にポーリングすると, アラームコンパレータの電圧比較入力に入力される電圧を監視できます。

IRQフラグビットのセットとリセット

ACSR の IRQ ビットは "0" を書き込むことでリセットできます。ACSR の IRQ ビットへの "1" の書込みでは何も起こりません。IRQ はハードウェア (コンパレータ回路の出力) でのみ "1" にセットできます。IRQ は "0" を書き込んでも , 有効な割込み状態が検出されている間は 有効なままになります。

ACSR に対してビットセット命令を実行すると , リードモディファイライトアクセスとなります。リードモディファイライト命令のリードアクセスでは , IRQ フラグからは常に "1" が読み出されます。これらにより , ソフトウェアで IRQ フラグビットを設定することによって割込みイベントの検出における無駄を回避できます。

パワーダウンモードおよびアラームコンパレータの比較モード

アラームコンパレータ回路の各動作モードを次表に示します。

Table 19-2. アラームコンパレータの動作モード

ストップ	タイマ	スリープ	デジタル部
0	0	0	動作
0	0	1	動作
0	1	0	停止
1	0	0	停止

ACSR:PD	アナログ部
0	ランモード
1	パワーダウン

＜注意事項＞

- アラームコンパレータ (アナログ部) の出力は , パワーオン後および ランモードへの移行後は最小 3µs の間 , 不定のままになります。アラームコンパレータのアナログ部が不安定な間は割込みの発生を禁止してください。
- MCU がスタンバイモード (ストップ , タイマ , スリープ) の間にアラームコンパレータの動作を禁止する必要がある場合は , SMCR レジスタで MCU のスタンバイモードを設定する前に ACSR の PD ビットを設定してください。
- アラームコンパレータをパワーダウンモードに移行するには , 以下の手順を使用してください。
  1. 割込み発生を禁止 (ACSRn:IEN = 0)
  2. アラームコンパレータをパワーダウン (ACSRn:PD = 1)
- アラームコンパレータをパワーダウンモードから ランモードに移行するには , 以下の手順を使用してください。
  1. アラームコンパレータをパワーダウンモードからランモードに切換え (ACSRn:PD = 0)
  2. 3µs 待機
  3. 割込みを許可 (ACSRn:IEN = 1) するか , またはアラームコンパレータをポーリングモードで使用

20. USART

LIN-USART の機能と動作について説明します。

20.1 USARTの概要

LIN (Local Interconnect Network)機能付き USART は, 外部デバイスと同期通信または非同期通信を行うための汎用シリアルデータ通信インタフェースです。 USART は双方向通信機能(ノーマルモード), マスタ/スレーブ通信機能(マスタ/スレーブシステムでのマルチプロセッサモード)に加えて LIN バスに対応するための特別な機能(マスタデバイスまたはスレーブデバイスとして動作)もサポートしています。

USARTの機能

USARTは,ほかのCPUおよび周辺デバイスとの間でシリアルデータを送受信するための汎用シリアルデータ通信インタフェースです。 USART の機能を Table 20-1 に示します。

Table 20-1. USARTの機能 (Sheet 1 of 2)

項目	機能
データバッファ	全二重方式
シリアル入力	5 回オーバサンプリング(非同期モード)
転送モード	• クロック同期 (調歩同期 , またはスタート / ストップビットオプション) • クロック非同期(スタート/ストップビットを使用)
ボーレート	• 16 ビットリロードカウンタ搭載の専用ボーレートジェネレータが搭載されている • 外部クロックの入力ができ、リロードカウンタでの分周が可能
データ長	• 7 ビット (同期モードと LIN モード以外) • 8 ビット
信号方式	Non-return to zero (NRZ)
スタートビットタイミング	非同期モード時は,スタートビットの立下りエッジに同期
受信エラー検出	• フレーミングエラー • オーバランエラー • パリティエラーモード1では未サポート
割込み要求	• 受信割込み (受信完了 , 受信エラー検出 , LIN synch break 検出) • 送信割込み (送信データエンプティ) • ICU (LIN Synch field 検出)への割込み要求(DMA 機能のための送受信サポート)
マスタ / スレーブ通信機能 (マルチプロセッサモード)	1 対 n 通信 (1 つのマスタ対 n 個のスレーブ) (マスタおよびスレーブシステムの両方をサポート)
同期モード	マスタ USART またはスレーブ USART として機能

Table 20-1. USARTの機能 (Sheet 2 of 2)

項目	機能
送受信端子	直接アクセス可能
LIN バスオプション	• マスタデバイス動作 • スレーブデバイス動作 • LIN Synch break 生成 • LIN Synch break 検出 • ICU は LIN Synch field 内で検出されたスタート/ストップエッジのビット時間を測定するために使用できます。 LIN USART の Synch field 信号への ICU の割当てについては,「1.6 LIN-USARTのインプットキャプチャユニットのソースの選択」を参照してください。
同期シリアルクロック	同期通信時に,スタートビットとストップビットを使って通信するため, SCKの常時出力が可能です。
クロック遅延オプション	遅延クロック用特殊同期クロックモード (シリアルペリフェラルインタフェース(SPI)に有用)

USARTの動作モード

USARTは4つの異なる動作モードで動作します。動作モードはシリアルモードレジスタ(SMRn)のMD0,MD1ビットにより決定されます。 モード 0 と モード 2 は双方向シリアル通信, モード 1 はマスタ/スレーブ通信, モード 3 は LIN マスタ/スレーブ通信に使用されます。

Table 20-2. USARTの動作モード

動作モード		データ長		同期方式	ストップビット長	データビット方向 *1
		パリティなし	パリティあり			
0	ノーマルモード	7 または 8ビット		非同期	1 または 2ビット	L/M
1	マルチプロセッサモード	7 または 8ビット + 1 *2	--	非同期	1 または 2ビット	L/M
2	ノーマルモード	8ビット		同期	0, 1 または 2ビット	L/M
3	LIN モード	8ビット	--	非同期	1ビット	L

*1: データビット転送フォーマットが LSB ファーストか MSB ファーストかを示します。
*2: "+1" は,マルチプロセッサモードで通信制御に使用されるアドレス/データ形式選択ビットです。

＜注意事項＞

モード 1 は , マスタとスレーブ動作の両方でサポートしています。

モード 3 では , 通信フォーマットが 8 ビットデータ , パリティなし , ストップビット 1, LSB ファーストに固定されます。

モードが変更された場合には , USART はすべての送受信を切断した後 , 待機状態となります。

USART の動作モードは , シリアルモードレジスタ (SMRn) の MD1 ビットと MD0 ビットにより , 次のように決定されます。

Table 20-3. モードビット設定

MD1	MD0	モード	種類
0	0	0	非同期(ノーマルモード)
0	1	1	非同期(マルチプロセッサモード)
1	0	2	同期(ノーマルモード)
1	1	3	非同期(LIN モード)

20.2 USARTの構成

USARTの構成するブロックについて説明します。

20.2.1 USARTのブロックダイヤグラム

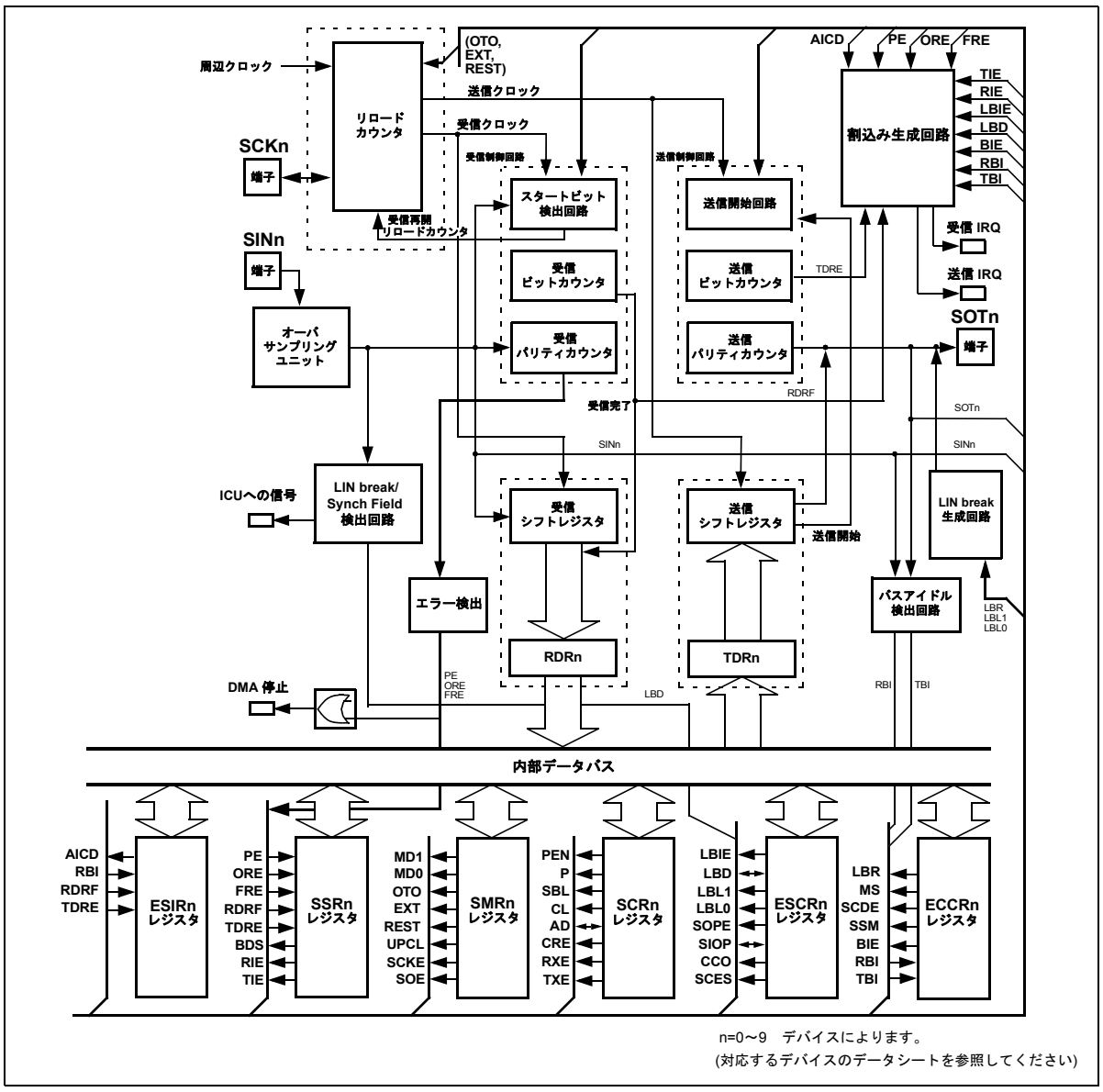
USART は以下のブロックで構成されます。

- リロードカウンタ
- 受信制御回路
- 受信シフトレジスタ
- 受信データレジスタ(RDRn)
- 送信制御回路
- 送信シフトレジスタ
- 送信データレジスタ(TDRn)
- エラー検出回路
- オーバサンプリングユニット
- 割込み生成回路
- LIN synch break/Synch Field 検出
- バスアイドル検出回路
- LIN-USART シリアルモードレジスタ(SMRn)
- シリアル制御レジスタ(SCRn)
- シリアルステータスレジスタ(SSRn)
- 拡張通信制御レジスタ(ECCRn)
- 拡張ステータス/制御レジスタ(ESCRn)
- 拡張シリアル割込みレジスタ(ESIRn)

＜注意事項＞

サフィックス "n" は USART 番号を表します。

Figure 20-1. USARTのブロックダイヤグラム



リロードカウンタ

リロードカウンタは専用ボーレートジェネレータとして機能します。クロックの送受信用に , 外部入力クロックまたは内部クロックを選択できます。リロードカウンタには ,
リロード値用の 16 ビットレジスタが内臓されています。BGRn レジスタを使用して , 送信リロードカウンタのカウンタ値を読み出すことができます。

受信制御回路

受信制御回路は, 受信ビットカウンタ , スタートビット検出回路, 受信パリティカウンタから構成されます。受信ビットカウンタは受信データビットをカウントします。指定されたデータ長のデータ受信が完了すると , 受信ビットカウンタは受信データレジスタフルフラグをセットします。スタートビット検出回路はシリアル入力信号からスタートビットを検出し , リロードカウンタへ信号を送信して , スタートビットの立下りエッジに信号を

同期します。受信パリティカウンタは受信データのパリティを計算します。

受信シフトレジスタ

受信シフトレジスタは , ビット単位でデータをシフトしながら SINn 端子から受信データ入力を取り込みます。受信が完了すると , 受信シフトレジスタは受信データを RDRn レジスタに転送します。

受信データレジスタ(RDRn)

受信データを格納します。シリアル入力データは , 変換後 , このレジスタに格納されます。

送信制御回路

送信制御回路は, 送信ビットカウンタ, 送信開始回路, 送信パリティカウンタから構成されます。送信ビットカウンタは送信データビットをカウントします。指定されたデータ長のデータ送信が完了すると , 送信ビットカウンタは送信データレジスタフルフラグをセットします。データが TDRn に書き込まれると , 送信開始回路は送信を開始します。送信パリティカウンタは , パリティありの場合 , 送信するデータのパリティビットを生成します。

送信シフトレジスタ

送信シフトレジスタは TDRn レジスタに書き込まれたデータを自身に転送し , ビット単位でデータをシフトしながら , SOTn 端子に出力します。

送信データレジスタ(TDRn)

送信データレジスタには送信データをセットします。このレジスタに書き込まれたレジスタはシリアルデータに変換され , 出力されます。

エラー検出回路

エラー検出回路は , 受信エラーの有無を確認します。エラーが検出された場合は , 該当するエラーフラグがセットされます。

オーバサンプリングユニット

オーバサンプリングユニットは , 非同期モード時に SINn 端子からの受信データを 5 回オーバサンプルします。同期モードでは停止します。

割込み生成回路

割込み生成回路は , すべての受信または送信割込みを管理します。該当するイネーブルフラグが設定されている場合は , 割込みが発生すると , 直ちに割込みが発生します。

LIN synch Break / Synchronization Field検出回路

LIN break/Synch field 検出回路は, LIN マスタノードがメッセージヘッダを送信すると, LIN break を検出します。LIN break が検出されると, LBD フラグビットがセットされます。インプットキャプチャユニットにより ,synch field の1 番目と5 番目の立下りエッジが検出され, マスタノードが送信した実際のシリアルクロック時間が測定されます。

LIN Synch Break生成回路

LIN break 生成回路は , 設定された長さの LIN break を生成します。

バスアイドル検出回路

バスアイドル検出回路は , 送信も受信も実行されていない状態を検出します。送受信が実行されていない場合 , この回路によって TBI, RBI フラグビットがセットされます。

LIN-USARTシリアルモードレジスタ(SMRn)

以下に動作を示します。

- LIN-USART 動作モード選択
- クロック入力ソースの選択
- 外部クロックが「1対1」接続か,またはリロードカウンタ接続であるか選択
- 専用リロードタイマのリセット
- LIN-USART のリセット(レジスタ設定を保持)
- シリアルデータ端子への出力許可/禁止設定
- クロック端子への出力許可/禁止設定

シリアル制御レジスタ(SCRn)

以下に動作を示します。

- パリティビットの有無の設定
- パリティビットの選択
- ストップビット長の設定
- データ長の設定
- モード 1 でのフレームデータ形式の選択
- エラーフラグのクリア
- 送信の許可/禁止
- 受信の許可/禁止

シリアルステータスレジスタ(SSRn)

以下に機能を示します。

- 送受信動作とエラーの状態表示
- LSB ファースト/MSB ファーストの設定
- 受信割込みの許可/禁止
- 送信割込みの許可/禁止

拡張ステータス制御レジスタ(ESCRn)

以下に機能を示します。

- LIN synch break 割込みの許可/禁止
- LIN synch break 検出
- LIN synch break 長の設定
- SINn 端子および SOTn 端子の直接アクセス
- 連続クロック出力設定
- サンプリングクロックエッジの設定

拡張通信制御レジスタ(ECCRn)

以下に機能を示します。

- バスアイドル状態を示す
- 同期クロックの設定
- LIN synch break 生成の設定

拡張シリアル割込みレジスタ(ESIRn)

以下に機能を示します。

- DMA との連携を可能にするための割込み処理の変更

20.3 USARTの端子

USART の端子について説明し, 端子のブロックダイヤグラムを示します。

USARTの端子

USARTの端子は汎用ポートと共用されています。Table 20-4に端子の機能,入出力フォーマット,USARTを使用するために必要な設定を示します。

Table 20-4. USARTの端子

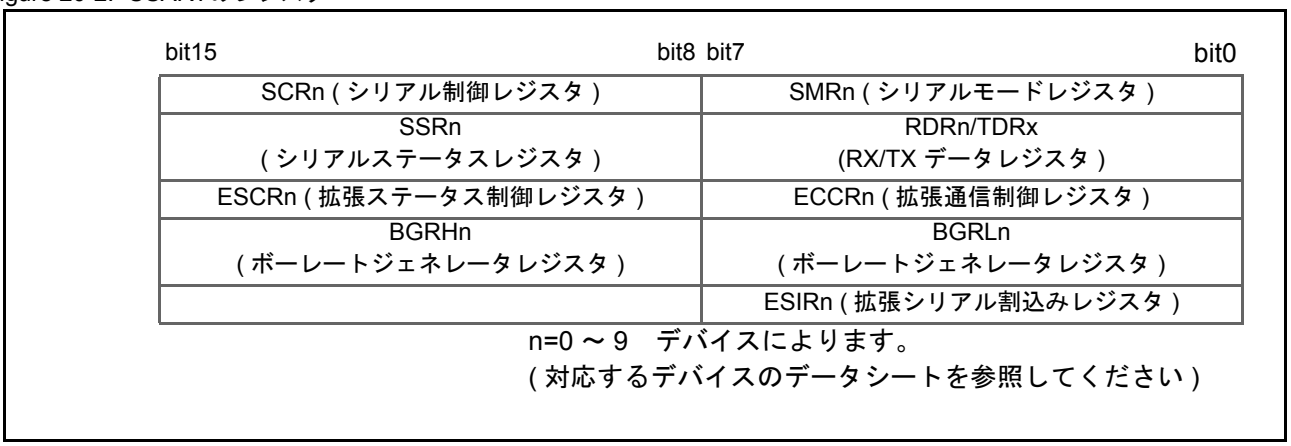
端子名	端子機能	入出力フォーマット	プルアップ	スタンバイ制御	端子を使用するために必要な設定
Pxx_i/SINn	ポート I/O またはシリアル データ入力	使用可能なフォーマットの選択については,「第13章 I/O ポート」を参照してください。	プログラマブルプルアップ抵抗 (プルアップ抵抗をオンにする方法については,「第13章 I/Oポート」を参照してください)	あり	対応する汎用ポートを「入力」に設定します。 DDRxx:Di = 0 PIERxx:IEi = 1
Pyy_j/SOTn	ポート I/O またはシリアル データ出力				出力許可モードを設定。 (SMRn: SOE = 1)
Pzz_k/SCKn	ポート I/O またはシリアルクロック入出力				クロックを入力する場合は , 対応する汎用ポートを「入力」に設定。 DDRzz:Dk = 0 PIERzz:IEk = 1
					クロックを出力する場合は , 出力許可モードに設定。 SMRn:SCKE = 1

20.4 USARTのレジスタ

下図に USART のレジスタを示します。

USARTのレジスタ

Figure 20-2. USARTのレジスタ



20.4.1 シリアル制御レジスタ(SCRn)

シリアル制御レジスタ(SCRn)は,パリティビットの設定,ストップビット長やデータ長の選択,モード1でのフレームデータ形式の選択,受信エラーフラグのクリア,送受信動作の許可または禁止の設定を行います。

シリアル制御レジスタ(SCRn)

Figure 20-3. シリアル制御レジスタ(SCRn)の構成

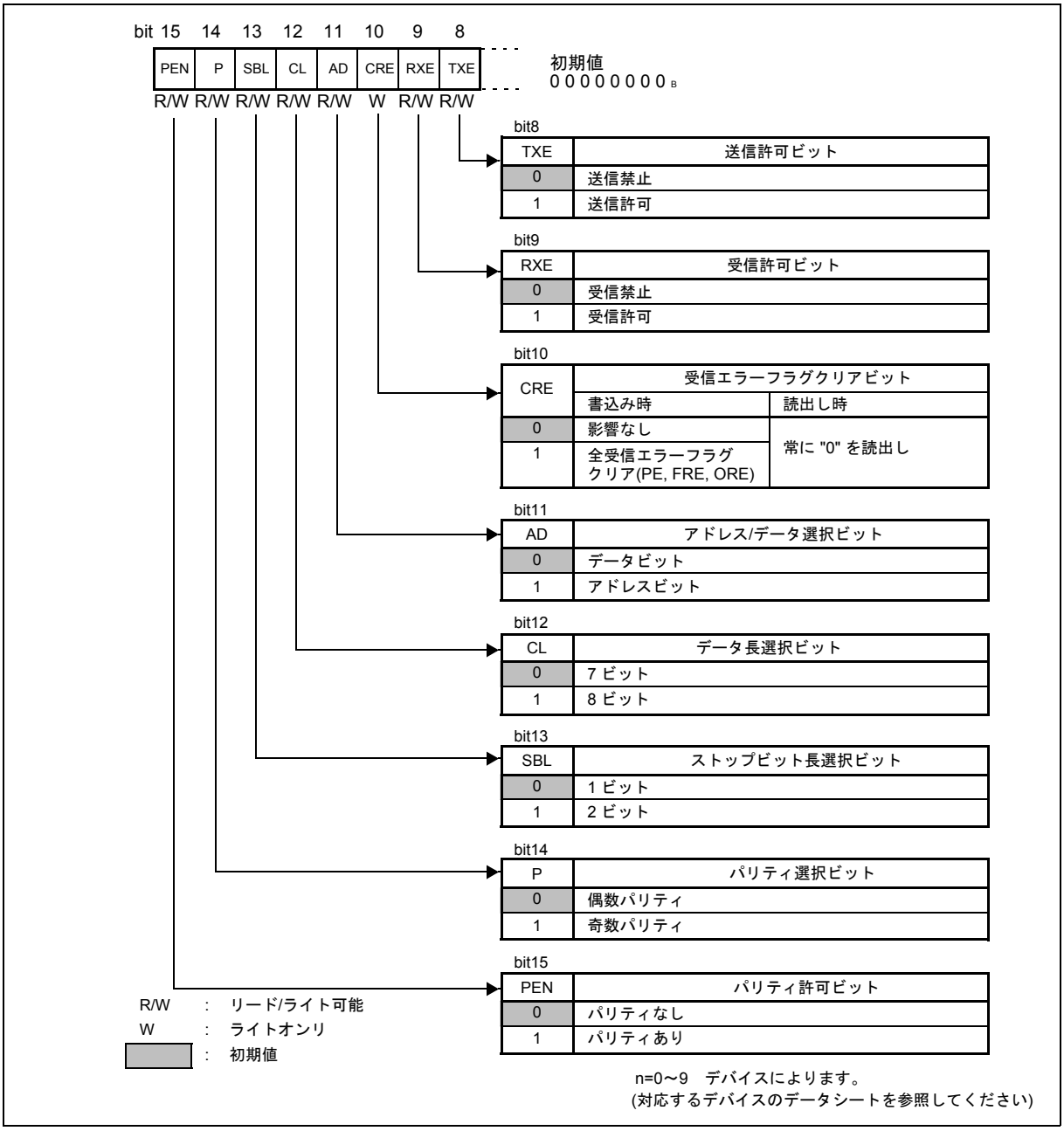


Table 20-5. 制御レジスタ(SCRn)の各ビットの機能

ビット名		機能
bit15	PEN: パリティ許可ビット	送信時のパリティビットの付加と受信時のパリティビットの検出を行うかどうかを設定します。 パリティビットはモード0,およびモード2でECCRnのSSMをセットした場合にのみ付加されます。 モード1 (マルチプロセッサ)およびモード3 (LIN)では, このビットは"0" (パリティなし)に固定されます。
bit14	P: パリティ選択ビット	パリティが使用可能であり, パリティビットありに設定した場合に,奇数パリティ(1)か偶数パリティ(0)のいずれかに設定します。
bit13	SBL: ストップビット長選択ビット	非同期データフレームまたはECCRnのSSMがセットされている場合の同期フレームのストップビット長を選択します。 本ビットは, モード3 (LIN)では, "0" (1ビット)に固定されます。
bit12	CL: データ長選択ビット	送受信データの長さを設定します。 本ビットは, モード2およびモード3では"1"(8ビット)に固定されます。
bit11	AD: アドレス/ データ選択ビット	マルチプロセッサモード (モード1) でのデータ形式を指定します。マスタ側は本ビットに書き込み , スレーブ側は , 本ビットを読み出します。"1" に設定した場合は , アドレスデータのフレームに設定され , "0" に設定した場合は , データフレームに設定されます。 (注意事項) 本ビットを使用するにあたっては,「20.8 USARTの使用上の注意」 を参照してください。
bit10	CRE: 受信エラーフラグクリアビット	シリアルステータスレジスタ (SSRn) の FRE, ORE, PE フラグをクリアし , 進行中の受信動作をリセットします。 "1" を書き込むとエラーフラグがクリアされます。 "0" の書込みは動作に影響を与えません。 読み出した場合には常に "0" が返されます。 (注意事項) エラーフラグのクリアは受信動作終了後に行ってください。
bit9	RXE: 受信許可ビット	LIN-USART の受信動作を許可または禁止します。 "0" に設定すると , データフレームの受信が禁止されます。 "1" に設定すると , データフレームの受信が許可されます。 モード 3 での LIN synch break 検出は影響されません。 (注意事項) 受信中に受信動作を禁止 (RXE=0) した場合には , 直ちに受信動作が停止します。この場合 , データは保証されません。 割込みルーチン内でシリアル制御レジスタ(SCRn) の CRE ビットをクリアした後, 実際にCREビットがクリアされるのを待つ為に, 周辺バス1 に接続されている周辺機能のレジスタを一度ダミーリードした後で, 割込みルーチンから復帰してください。
bit8	TXE: 送信許可ビット	LIN-USART の送信動作を許可または禁止します。 "0" に設定すると , データフレームの送信が禁止されます。 "1" に設定すると , データフレームの送信が許可されます。 (注意事項) 送信中に送信動作を禁止(TXE=0)した場合には,直ちに送信動作が停止します。この場合,データは保証されません。

20.4.2 シリアルモードレジスタ(SMRn)

シリアルモードレジスタ (SMRn) は , 動作モードの選択 , ボーレートクロックの選択 , シリアルデータとクロックの端子への出力許可または禁止の設定を行います。

シリアルモードレジスタ(SMRn)

Figure 20-4. シリアルモードレジスタ(SMRn)の構成

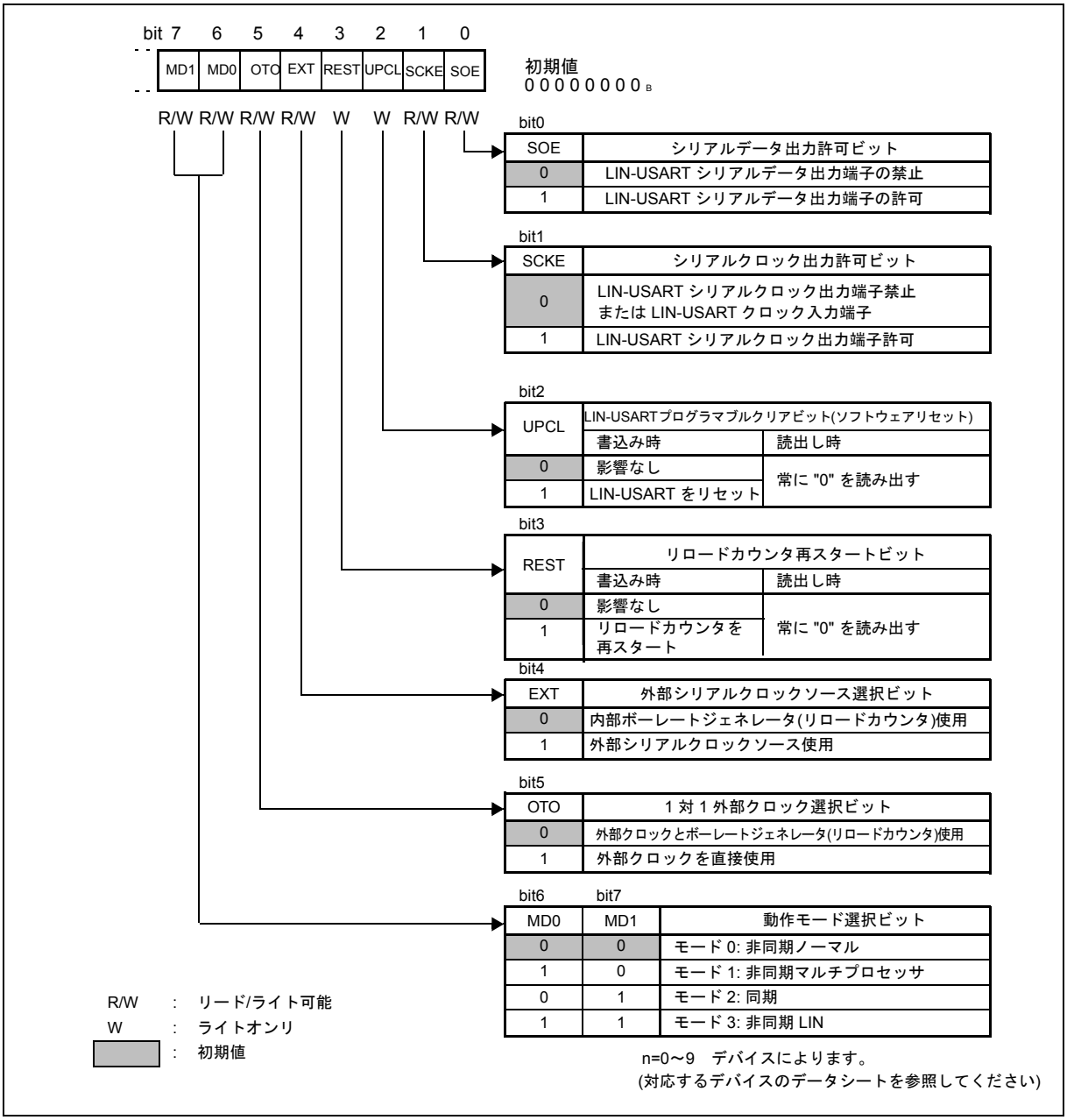


Table 20-6. シリアルモードレジスタ(SMRn)の各ビットの機能

ビット名		機能
bit7, bit6	MD1 および MD0: 動作モード選択ビット	USART の動作モードを設定します。 (注意事項) MD1, MD0 ビットを変更すると, 進行中の送受信は中断されます。
bit5	OTO: 1 対 1 外部クロック選択ビット	外部クロックを LIN-USART のシリアルクロックとして直接使用します。 本機能は動作モード 2 (同期)スレーブモード動作で使用します。
bit4	EXT: 外部クロック選択ビット	クロック入力を選択します。 "0" 設定 : 内部ボーレートジェネレータ (リロードカウンタ) を使用 "1"設定:外部シリアルクロックソースを使用
bit3	REST: 送信リロードカウンタリスタートビット	"1" を書き込んだ場合には , リロードカウンタを再スタートします。 "0" を書き込んだ場合は , 影響はありません。 常に0が読み出されます。
bit2	UPCL: USART プログラマブルクリアビット (ソフトウェアリセット)	"1" を書き込んだ場合には , LIN-USART を即時リセットします。ただし , レジスタの設定は維持されます。その際 , 送受信は中断されます。 すべての送受信割込み要因 (TDRE, RDRF, LBD, PE, ORE, FRE) は解除され , 受信データレジスタ (RDRn) の内容は 00_H になります。"0" を書き込んだ場合は , 動作に影響はありません。 常に"0"が読み出されます。送受信割込みを禁止に設定した後, LIN-USARTのリセットを行ってください。
bit1	SCKE: シリアルクロック出力許可ビット	• シリアルクロックの入出力を制御するビットです。 • "0" の場合は、SCKn 端子は汎用 I/O ポート , またはシリアルクロック入力端子として機能します。"1" に設定し , ほかの周辺機能がこの端子を出力として使用しない場合は , シリアルクロック出力端子として機能し , 動作モード 2(同期) の場合にクロックを出力します。 (注意事項) SCKn 端子をシリアルクロック入力 (SCKE=0) として使用する場合 , 汎用入出力ポートの対応する DDR ビットを入力ポートに設定してください。さらに , 外部クロック選択ビットを選択 (EXT=1) にしてください。 割込みルーチン内でシリアルモードレジスタ (SMRn) の UPCL ビットをクリアした後 , 実際に TDRE, RDRF, LBD, PE, ORE, FRE ビットがクリアされるのを待つ為に , 周辺バス 1 に接続されている周辺機能のレジスタを一度ダミーリードした後で , 割込みルーチンから復帰してください。 (参考) SCKn端子がシリアルクロック出力(SCKE=1)に設定され,ほかの周辺機能がこの端子を出力に使用しない場合は,汎用入出力ポートの状態にかかわらず,シリアルクロック出力端子として機能します。
bit0	SOE: シリアルデータ出力許可ビット	• シリアルデータの出力を許可または禁止します。 • "0" に設定すると , SOTn 端子は汎用 I/O 端子として機能します。"1" に設定し , ほかの周辺機能がこの端子を出力として使用しない場合は, SOTn 端子はシリアルデータ出力端子(SOT) として機能します。 (参考) シリアルデータ出力(SOE=1)で,ほかの周辺機能がこの端子を出力として使用しない場合,汎用入出力ポートの状態にかかわらず,端子はシリアルデータ出力端子(SOT)として機能します。

20.4.3 シリアルステータスレジスタ(SSRn)

シリアルステータスレジスタ(SSRn)は,送受信やエラーの状態の確認,割込みの許可または禁止の設定を行います。

シリアルステータスレジスタ(SSRn)

Figure 20-5. シリアルステータスレジスタ(SSRn)の構成

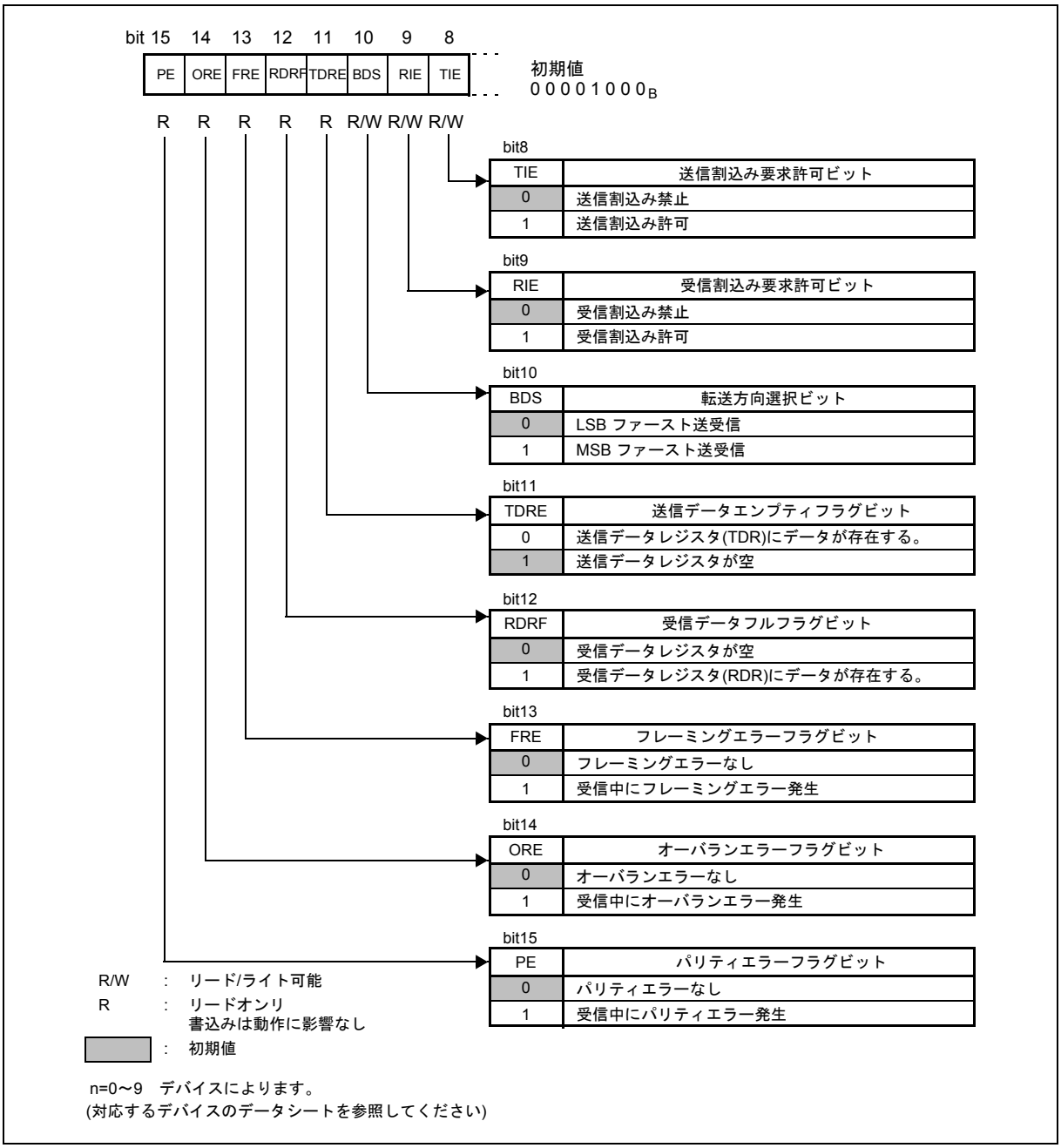


Table 20-7. ステータスレジスタ(SSRn)の各ビットの機能

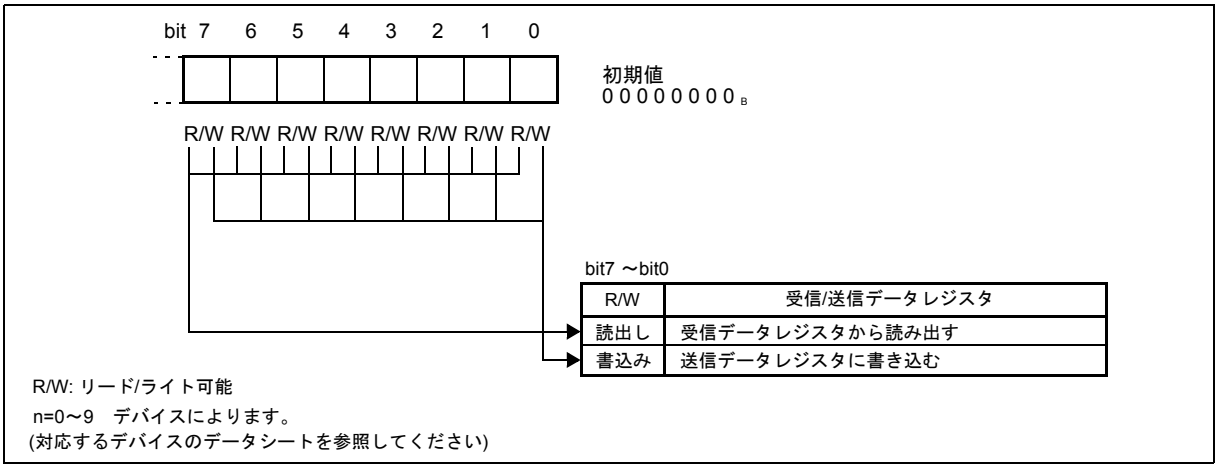
Bit name		Function
bit15	PE: パリティ エラーフラグビット	• PE=1 での受信中にパリティエラーが発生すると "1" にセットされます。 • シリアル制御レジスタ (SCRn) の CRE ビットに "1" を書き込むとクリアされます。 • モード 3 では , ESCRn:LBD = 1 の書込みでもクリアされます。 • PE ビットと RIE ビットが "1" に設定された場合 , 受信割込み要求を出力します。 • このフラグがセットされている場合は, 受信データレジスタ(RDRn)内のデータは無効です。
bit14	ORE: オーバラン エラーフラグビット	• 受信中にオーバランエラーが発生すると "1" にセットされます。 • シリアル制御レジスタ (SCRn) の CRE ビットに "1" を書き込むとクリアされます。 • モード 3 では , ESCRn:LBD = 1 の書込みでもクリアされます。 • ORE ビットと RIE ビットが "1" に設定された場合 , 受信割込み要求を出力します。 • このフラグがセットされている場合は, 受信データレジスタ(RDRn)内のデータは無効です。
bit13	FRE: フレーミング エラーフラグビット	• 受信中にフレーミングエラーが発生すると "1" にセットされます。 • シリアル制御レジスタ 1(SCRn) の CRE ビットに "1" を書き込むとクリアされます。 • モード 3 では , ESCRn:LBD = 1 の書込みでもクリアされます。 • FRE ビットと RIE ビットが "1" に設定された場合 , 受信割込み要求を出力します。 • このフラグがセットされている場合は, 受信データレジスタ(RDRn)内のデータは無効です。
bit12	RDRF: 受信データ フルフラグビット	• 受信データレジスタ (RDRn) の状態を示します。 • 受信データが RDRn にロードされると "1" にセットされます。 • 受信データレジスタ (RDRn) を読み出すとクリアされます。 • モード 3 では , ESCRn:LBD = 1 の書込みでもクリアされます。 • RDRF ビットと RIE ビットが "1" に設定された場合 , 受信割込み要求を出力します。 (注意事項) ESIR:RDRF は本ビットと同様の動作をしますが , 受信データレジスタ (RDRn) の読出しではクリアされません。 割込みルーチン内でRDRnレジスタをリードしてステータスレジスタ(SSRn) の RDRF ビットをクリアした後, 実際にRDRFビットがクリアされるのを待つ為に, 周辺バス1 に接続されている周辺機能のレジスタを一度ダミーリードした後で, 割込みルーチンから復帰してください。
bit11	TDRE: 送信データ エンプティ フラグビット	• 送信データレジスタ (TDRn) の状態を示します。 • TDRn に送信データを書き込むと "0" になり , データシフトレジスタにロードされて送信が開始されると , "1" になります。 • TDRE ビットと , TIE ビットが "1" の場合に送信割込み要求を出力します。 • TDRE ビットが "1" の場合に ECCRn レジスタの LBR ビットが "1" に設定されると , 本ビットは一度 "0" に変わります。LIN synch break 生成終了後に , TDRE ビットは "1" に戻ります。 (注意事項) • 本ビットの初期値は "1" (TDRn エンプティ) です。 ESIR:TDREは本ビットと同様の動作をしますが, TDRn への送信データの書込みではクリアされません。
bit10	BDS: 転送方向選択ビット	• 転送シリアルデータを最下位ビットから転送するか (LSB ファースト , BDS=0) または最上位ビットから転送するか (MSB ファースト , BDS=1) を選択します。 (注意事項) • シリアルデータレジスタの読出しまたは書込みの際にシリアルデータの上位側と下位側が入れ替えられます。 そのため, データが 受信データレジスタ(RDRn) に書き込まれた後に本ビットを書き換えるとデータは無効になります。 本ビットはモード 3 (LIN)では"0"に固定されます。
bit9	RIE: 受信割込み要求許可ビット	• 受信割込みを許可または禁止します。RDRF, PE, ORE, FRE ビットのいずれかがセットされたときに本ビットが "1" であると , 受信割込みが割込みコントローラに通知されます。
bit8	TIE: 送信割込み要求許可ビット	• 送信割込みを許可または禁止します。 • このビットと TDRE ビットが "1" にセットされると, 送信割込み要求が発生します。

20.4.4 受信データレジスタ / 送信データレジスタ(RDRn/TDRn)

受信データレジスタ(RDRn)は受信データを保持します。 送信データレジスタ(TDRn)は送信データを保持します。RDRnレジスタとTDRnレジスタは同一アドレスに配置されています。

20.4.4.1 受信データレジスタ / 送信データレジスタ(RDRn/TDRn)

Figure 20-6. 受信データレジスタ / 送信データレジスタ(RDRn/TDRn)



受信:

RDRn は受信データを保持するレジスタです。SINn 端子に送られてきたシリアルデータ信号は, シフトレジスタで変換されて受信データレジスタに格納されます。データ長が 7 ビットの場合には , 最上位ビット (D7) は , "0" になります。受信が完了するとデータは本レジスタに格納され , 受信データフルフラグビット (SSRn:RDRF) が "1" にセットされます。この時点で受信割込み要求が許可されていると , 受信割込みが発生します。

RDRn はステータスレジスタ (SSRn) の RDRF ビットが "1" のときに読み出してください。RDRn を読み出すと , RDRF ビットは自動的に "0" にクリアされます。また,受信割込みが許可されており , エラーが発生していない場合にも受信割込みはクリアされます。受信エラーが発生した場合 (SSRn の PE, ORE, FRE のいずれかが "1") には , RDRn 内のデータは無効になります。

送信:

送信許可状態の送信データレジスタに送信するデータを書き込むと, データは送信シフトレジスタに転送されてシリアルデータに変換され , シリアルデータ出力端子 (SOTn 端子) から送信されます。データ長が 7 ビットの場合は , 最上位ビット (D7) は送信されません。

送信データを本レジスタに書き込むと , 送信データエンプティフラグビット (SSRn:TDRE) は "0" にクリアされます。送信シフトレジスタへの転送が完了して送信が開始されると , このビットは " 1" にセットされます。TDRE ビットが "1" の場合には , 次の送信データを書き込むことができます。送信割込み要求が許可されている場合には , 送信割込みが生成されます。次の送信データは送信割込みが発生したとき , または TDRE ビットが " 1" のときに書き込んでください。

＜注意事項＞

送信データレジスタ (TDRn) は書込み専用のレジスタで , 受信データレジスタ (RDRn) は読込み専用のレジスタです。これら 2 つのレジスタは同一アドレスに配置されているため , 書込み値と読出し値が異なります。したがって , INC/DEC 命令などのリードモディファイライト (RMW) 系の動作をする命令を使用できません。割込みルーチン内で RDRn レジスタをリードしてステータスレジスタ (SSRn) の RDRF ビットをクリアした後 , 実際に RDRF ビットがクリアされるのを待つ為に , 周辺バス 1 に接続されている周辺機能のレジスタを一度ダミーリードした後で , 割込みルーチンから復帰してください。

20.4.5 拡張ステータス制御レジスタ(ESCRn)

拡張ステータス制御レジスタ(ESCRn)は, LIN synch break 割込み許可/禁止, LIN synch break 長選択, LIN synch break 検出, SINn, SOTn 端子への直接アクセス, LIN-USART 同期クロックモードでの連続クロック出力, サンプリングクロックエッジの設定があります。

拡張ステータス制御レジスタ(ESCRn)

Figure 20-7. 拡張ステータス制御レジスタ(ESCRn)の構成

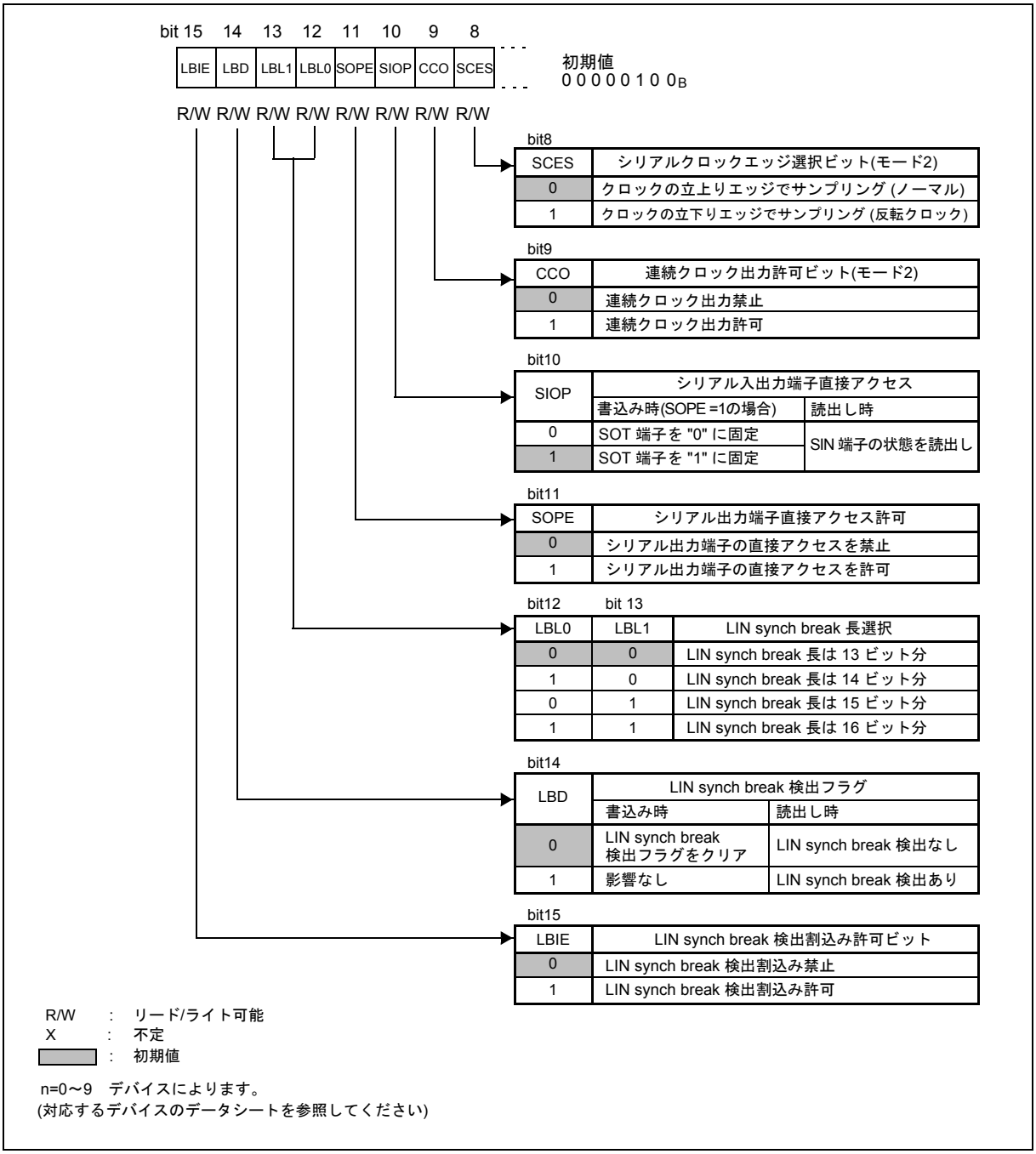


Table 20-8. 拡張ステータス制御レジスタ(ESCRn)の各ビットの機能

ビット名		機能
bit15	LBIE: LIN synch break 検出割込み許可 ビット	• LIN synch break 割込みを許可 / 禁止します。 • LIN synch break 割込みは受信割込みに接続されています。 モード 3 で LBD ビットがセットされたときにこのビットが "1" であると, 受信割込みが割込みコントローラに通知されます。 動作モード 1 と 2 では, 本ビットは "0" に固定されます。
bit14	LBD: LIN synch break 検出フラグ	• 動作モード 3 で LIN synch break が検出されると "1" になります。"0" を書き込むと , 本ビットおよび対応する割込みがクリアされます。 • 本ビットを使用する前に , SCRn の RXE ビットに "0" を書き込むことを推奨します。 • リードモディファイライト(RMW)系命令を実行すると,常に"1"が読み出されますが, LIN synch breakの検出によるものではありませんので注意してください。
bit13, bit12	LBL1/LBL0: LIN synch break 長 選択ビット	• これらのビットは , LIN synch break の生成時間を何ビット分にするかを設定します。受信 LIN synch break 長は , 常に 11 ビットです。
bit11	SOPE: シリアル出力端子直接 アクセス許可*	• SMRn:SOE = 1 の場合に本ビットを"1"に設定すると, SOTn 端子に直接書き込むことができます。*
bit10	SIOP: シリアル入出力端子 直接アクセス*	• 通常の読出し命令では SINn 端子の実際の値を返します。 SOPE = 1 の場合に書き込むと , 設定したビット値が SOTn 端子に出力されます。 • リードモディファイライト命令では, リードサイクルには SOTn 値が返されます。 *
bit9	CCO: 連続クロック 出力許可ビット	• USART がマスタモード 2 (同期) で動作しており , SCKn 端子がクロック出力として構成されている場合に , SCKn 端子への連続シリアルクロック出力を許可します。 (注意事項) CCO ビットが "1" の場合には, ECCRn:SSM ビットを "1" に設定してください。
bit8	SCES: シリアルクロック エッジ選択ビット	• 動作モード 2 (同期通信) でのシリアルクロック信号を反転します。 • 受信データは内部クロックの立下りエッジでサンプリングされます。ECCRn:MS ビットが "0" (マスタモード) であり , SMRn:SCKE ビットが "1" (クロック出力許可) の場合には , 出力クロック信号も反転されます。 • 動作モード 0, 1, 3 では, このビットを"0"に設定してください。

* : Table 20-9 参照。.

Table 20-9. SOPEとSIOPの相互作用説明

SOPE	SIOP	SIOPへの書込み	SIOPからの読出し
0	R/W	SOTn への影響なし(ただし, 書込み値は保持)	SINn の現在の値を返す
1	R/W	SOTn に "0" または "1" を書込み	SINn の現在の値を返す
1	RMW	SOTn の現在の値を読み出して, それを書き戻す	

20.4.6 拡張通信制御レジスタ(ECCRn)

拡張通信制御レジスタでは,バスアイドル検出の設定,同期クロックの設定,および LIN break の生成を行います。

拡張通信制御レジスタ(ECCRn)

Figure 20-8. 拡張通信制御レジスタ(ECCRn)の構成

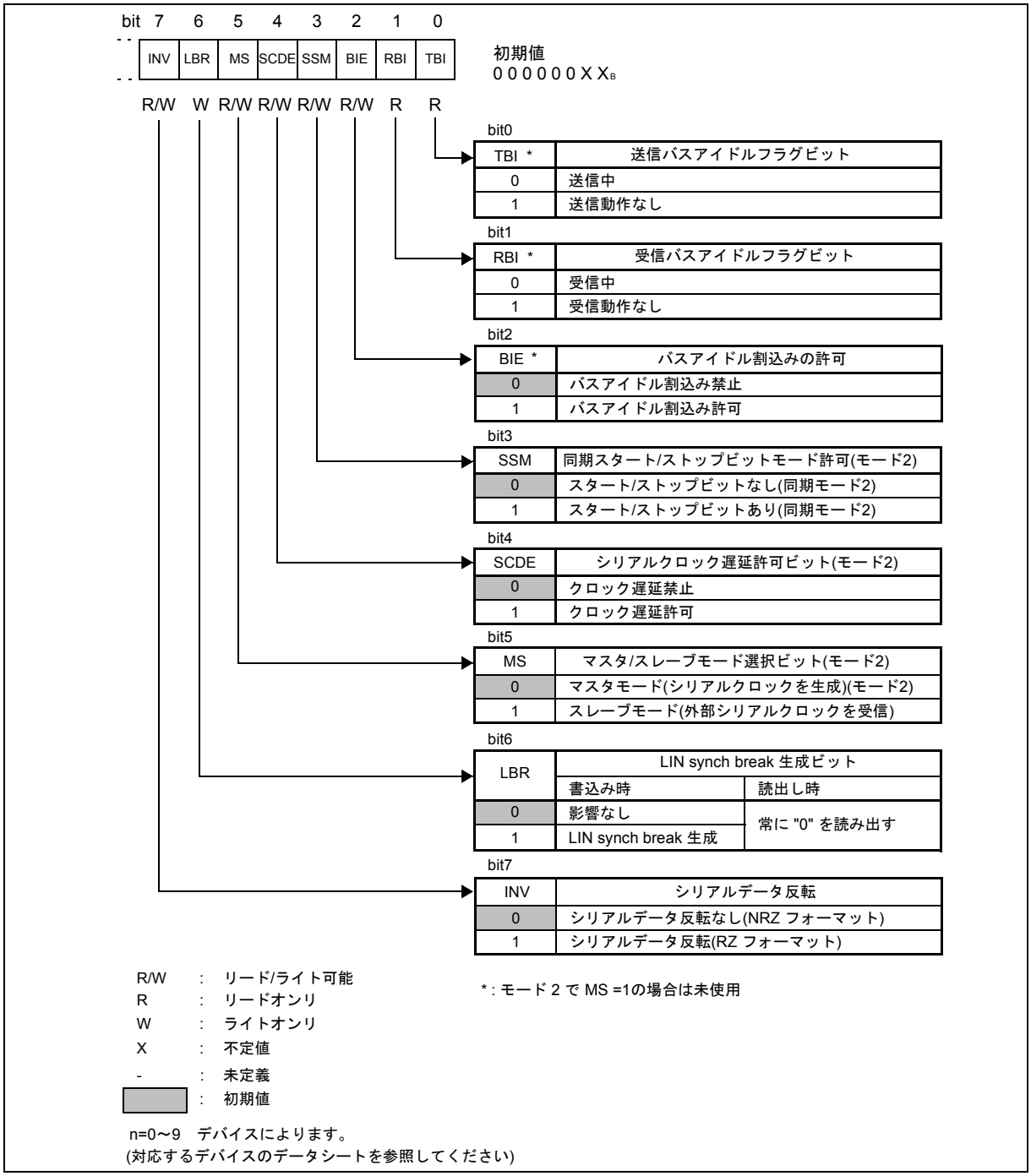


Table 20-10. 拡張通信制御レジスタ(ECCRn)の各ビットの機能

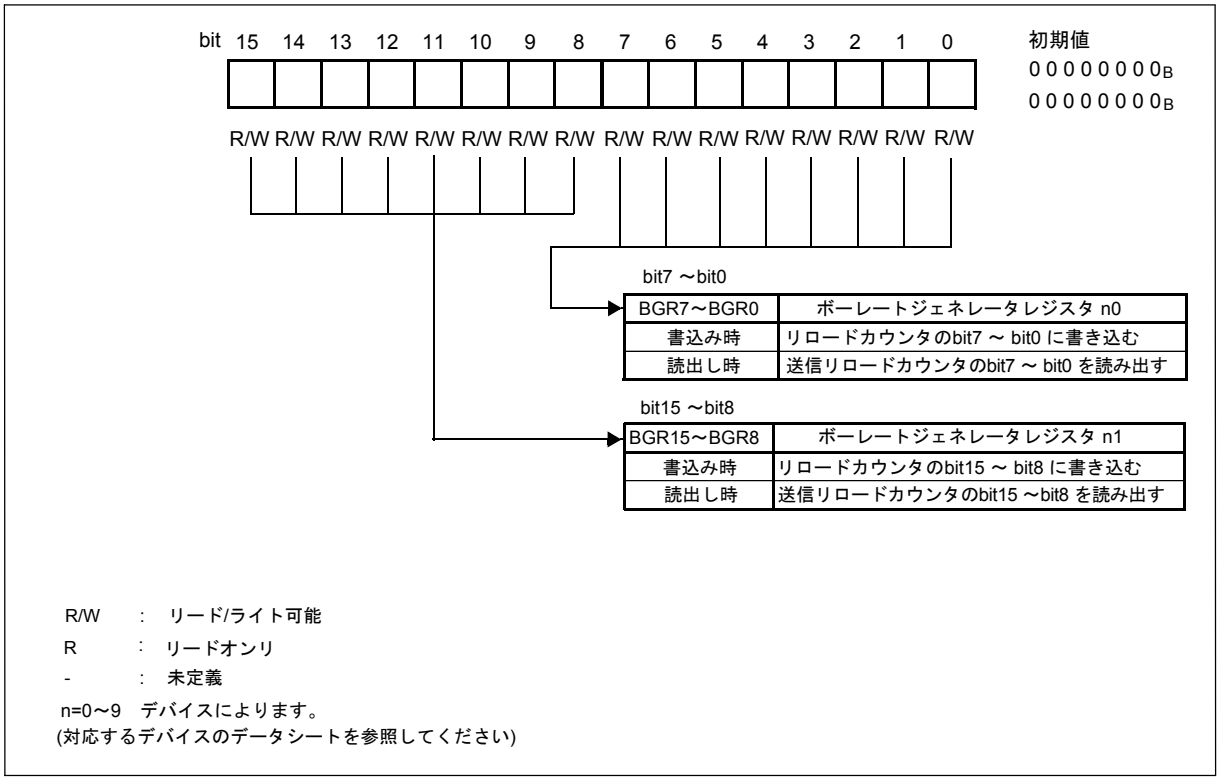
ビット名		機能
bit7	INV: シリアルデータ 反転	• SINn 端子と SOTn 端子でシリアルデータを反転します。SCKn は影響されません (ESCRn: SCES 参照)。 • "0" を書込み : シリアルデータフォーマットは NRZ (初期値) • "1" を書込み : シリアルデータは反転 (RZ フォーマット) • RMW 命令は本ビットに影響しません。
bit6	LBR: LIN synch break 生成ビット	• 動作モード 3 が選択されている場合に 1 を書き込むと, ESCRnの LBL0/LBL1 ビットで指定された長さの LIN synch break が生成されます。 動作モード 0 では "0" に設定してください。
bit5	MS: マスタ/スレーブモード選択 ビット	• 同期モード 2 でのマスタ / スレーブモードを選択します。 • マスタモードを選択した場合には , USART は同期クロックを自身で生成します。 • スレーブモードを選択した場合には , USART は外部シリアルクロックを受信します。 • 動作モード 0, 1, 3 では , 本ビットは "0" に固定されます。 (注意事項) スレーブモードを選択した場合には, クロックソースを外部に設定し, 「1 対 1」(SMRn:SCKE = 0, EXT = 1, OTO = 1)に設定する必要があります。
bit4	SCDE: シリアル クロック遅延 許可ビット	• このビットをセットすると, USART がマスタモード 2 で動作する場合に, Figure 20-20 に示すように遅延したクロックが出力されます。
bit3	SSM: スタート/ストップビットモード許可	• 動作モード 2 の同期データフォーマットにスタートビットとストップビットを付加します。モード 0, 1, 3 では無視されます。
bit2	BIE: バスアイドル 割込み許可	• バスアイドル検出による割込みを許可します。 • バスアイドルを通知するために , 送受信が進行中でない場合 (RBI = 1, TBI = 1) には , 受信割込みが許可されます。 • モード 2 で MS = 1 の場合には本ビットを使用しないでください。 (注意事項) バスアイドル割込み機能を使用する場合は, バスがアイドル状態でなくなったときに割込みが自動的にクリアされるのを抑止するために, ESIR:AICD =1 に設定してください。
bit1	RBI: 受信バスアイドルフラグビット	• SINn 端子が受信動作をしていない時に "1" になります。 • SINn 端子で受信動作が進行中には "0" になります。 • モード 2 の場合には本ビットを使用しないでください。
bit0	TBI: 送信バスアイドルフラグビット	• SOTn 端子が送信動作をしていない時に "1" になります。 • モード 2 で MS = 1 の場合には本ビットを使用しないでください。

20.4.7 ボーレート/リロードカウンタレジスタ(BGRn)

ボーレート/リロードカウンタレジスタにはシリアルクロックの分周比を設定します。 また, 送信リロードカウンタのカウント値を読み出すこともできます。

ボーレート/リロードカウンタレジスタ(BGRn)

Figure 20-9. ボーレート/リロードカウンタレジスタ(BGRn)



ボーレート/リロードカウンタレジスタ

ボーレート / リロードカウンタレジスタはシリアルクロックの分周比を決定します。

両レジスタはバイトアクセスまたはワードアクセスで読み書きできます。

20.4.8 拡張シリアル割込みレジスタ(ESIRn)

拡張シリアル割込みレジスタは USART の割込み処理を変更し, 割込み処理の改善および DMA による USART のデータ転送を可能にするコントロールビットを含みます。

拡張シリアル割込みレジスタ(ESIRn)

Figure 20-10. 拡張シリアル割込みレジスタ(ESIRn)の構成

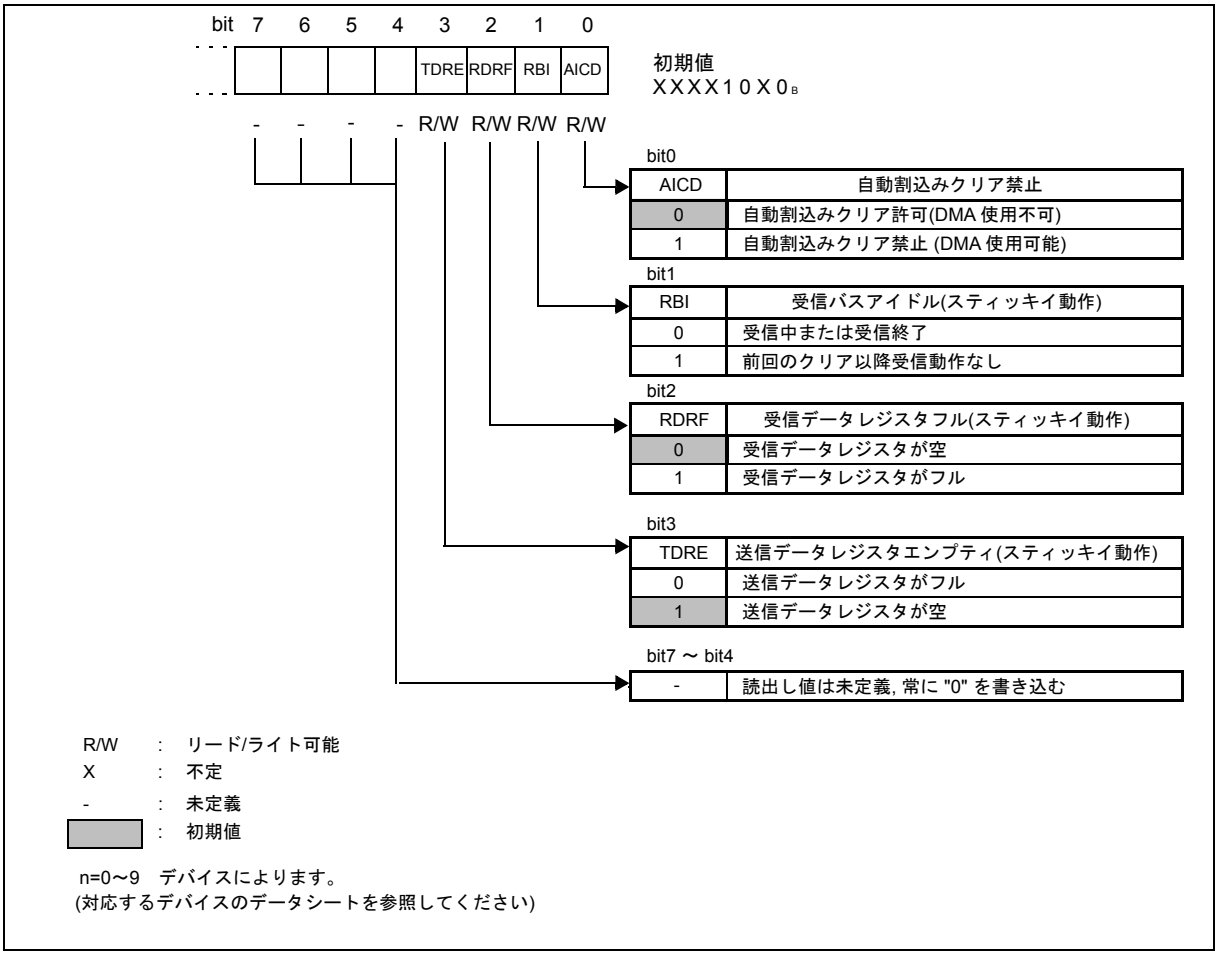


Table 20-11. 拡張シリアル割込みレジスタ(ESIRn)の各ビットの機能

ビット名		機能
bit7～ bit4	-	• これらのビットは未定義です。 • つねに"0"を書き込んでください。リードモディファイライト系命令は無効です。
bit3	TDRE: 送信データ レジスタ エンプティ (スティッキイ 動作)	• SSR:TDRE と同一の機能ですが , データが送信データレジスタ (TDR) に書き込まれてもクリアされません。 • 送信データレジスタ (TDR) の状態を示します。 • データが送信シフトレジスタにロードされ , 送信が開始されると "1" にセットされます。 • 本ビットは "0" の書込みでクリアする必要があります。 • AICD =0 で SSR:TIE =1 の場合には , 送信割込みを発生するために SSR:TDRE が使用されます。 • AICD =1 で SSR:TIE =1 の場合には , 送信割込みを発生するために ESIR:TDRE が使用されます。 (注意事項) 本ビットの初期値は"1" (TDR エンプティ)です。
bit2	RDRF: 受信データ レジスタフル (スティッキイ 動作)	• SSR:RDRF と同一の機能ですが , データを受信データレジスタ (RDR) から読み出してもクリアされません。 • 受信データレジスタ (RDR) の状態を示します。 • 受信データが RDR にロードされると "1" にセットされます。 • 本ビットは "0" の書込みでクリアする必要があります。 • AICD =0 で SSR:RIE =1 の場合には , 受信割込みを発生するために SSR:RDRF が使用されます。 • AICD =1で SSR:RIE =1の場合には, 受信割込みを発生するために ESIR:RDRF が使用されます。
bit1	RBI: 受信バスアイドル(ス ティッキイ 動作)	• ECCR:RBI と同一機能ですが , 受信バスがアイドル状態でなくなってもクリアされません。 • 受信動作が進行中でない場合に "1" にセットされます。 • 本ビットは "0" の書込みでクリアする必要があります。 • AICD =0 で ECCR:BIE =1 の場合には , 受信割込みを発生するために ECCR:RBI が使用されます。 • AICD =1 で ECCR:BIE =1 の場合には , 受信割込みを発生するために ESIR:RBI が使用されます。 (注意事項) バスアイドル状態で割込みを発生させるには, AICD =1に設定してください。 AICD =0の場合には ,ECCR:RBIが割込みの発生に使用されます。 バスがアイドル状態でなくなるとすぐにECCR:RBI は "0" になり, 割込み処理ルーチンが応答する前であっても, 割込みが取り除かれる場合があります。
bit0	AICD: 自動割込みクリア禁止	• "0" に設定すると , SSR:TDRE, SSR:RDRF, ECCR:RBI が割込みの発生に使用されます。このモードでは , USART は DMA と併用できません。 • "1" に設定すると , ESIR:TDRE, ESIR:RDRF, ESIR:RBI が割込みの発生に使用されます。このモードでは , USART は DMA と併用できます。 (注意事項) AICD=0の場合には,USARTはDMAと併用できません。 USARTのデータ転送にDMAを使用する場合は, AICD =1に設定してください。

20.5 USARTの割込み

USART は受信割込みと送信割込みを使用します。割込み要求は次のいずれの要因でも生成できます。

- 受信データレジスタ(RDRn)への受信データの転送, または受信エラーの発生。
- 送信データレジスタ(TDRn)から送信シフトレジスタへの送信データの転送および送信開始。
- LIN break の検出

これらの割込みに対しては DMA を使用可能です。

20.5.1 LIN-USART割込み

Table 20-12. LIN-USARTの割込み制御ビットと割込み要因

受信 / 送信 /ICU	割込み要求フラグビット	フラグレジスタ	動作モード				割込み要因	割込み要因許可ビット	割込み要求のクリア方法
			0	1	2	3			
受信	RDRF	SSRn	❍	❍	❍	❍	RDRnへの受信データの書込み	SSRn:RIE	受信データの読出し
	ORE	SSRn	❍	❍	❍	❍	オーバランエラー		受信エラークリアビット(SCRn:CRE)への "1" の書込み
	FRE	SSRn	❍	❍	1	❍	フレーミングエラー		
	PE	SSRn	❍	×	1	×	パリティエラー		
	LBD	ESCRn	×	×	×	❍	LIN synch break 検出	ESCRn:LBIE	ESCRn:LBD への "0" の書込み
送信	TDRE	SSRn	❍	❍	❍	❍	TDRnエンプティ	SSRn:TIE	TDRnへのデータの書込み
バスアイドル[3]	RBI, TBI	ECCRn,ESIRn	❍	❍	❍	❍	送受信動作なし	ECCRn:BIE	ESIRn:RBI への "0" の書込み, TDRn へのデータの書込み
インプットキャプチャ	ICPy[2]	ICSy[2]	×	×	×	❍	LIN synch field の最初の立下り	ICSy:ICEy[2]	一時的な ICPy の禁止
							LIN synch field の 5 回目の立下り		ICPy の禁止

❍: 使用可能
×: 使用不可
*1: ECCRn/SSM = 1 の場合のみ使用可能
*2: インプットキャプチャユニット数 (y) については ,「1.6 LIN-USART のインプットキャプチャユニットのソースの選択」を参照してください。
*3: 受信動作が開始されるとすぐに割込みがクリアされるため, ESIR:AICD = 0 でのバスアイドル割込みの使用は推奨されません。

受信割込み

受信モードで以下のイベントのいずれかが発生すると , シリアルステータスレジスタ (SSRn) と拡張シリアル割込みレジスタ (ESIRn) の対応するフラグビットが "1" にセットされます。

- データ受信完了。すなわち受信データがシリアル入力シフトレジスタから受信データレジスタ(RDRn)に転送され, データの読出しが可能になった場合:SSRn:RDRF, ESIRn:RDRF
- オーバランエラー。すなわちSSRn:RDRF=1でRDRがCPUによって読み出されておらず,次のシリアルデータを受信した場合:SSRn:ORE
- フレーミングエラー。すなわちストップビットが予期される時点で "0" を受信した場合: SSRn:FRE
- パリティエラー。すなわち不正なパリティビットが検出された場合:SSRn:PE

ESIR:AICD = 0 の場合には , SSRn:RDRF, SSRn:ORE, SSRn:FRE, SSRn:PE のいずれかが "1" になり , 受信割込みが許可されていると (SSRn:RIE = 1), 受信割込み要求が生成されます。

ESIR:AICD = 1 の場合には, ESIRn:RDRF, SSRn:ORE, SSRn:FRE, SSRn:PE のいずれかが "1" になり , 受信割込みが許可されていると (SSRn:RIE = 1), 受信割込み要求が生成されます。
受信データレジスタ (RDRn) が読み出されると , SSRn:RDRF フラグは自動的に "0" にクリアされます。なお , これは SSRn:RDRF フラグをリセットする唯一の方法です。
ESIRn:RDRF フラグは "1" を書き込むと "0" にクリアされます。
シリアル制御レジスタ (SCRn) の受信エラークリア (CRE) フラグビットに "1" を書き込むと , エラーフラグは "0" にクリアされます。RDRn には , RDRF フラグが "1" でエラービットがセットされていない場合にのみ有効なデータが含まれます。
なお , CRE フラグはライトオンリであり , "1" を書き込むと 1 周辺クロック (CLKP1) サイクルの間 , 内部的に "1" に保持されます。

送信割込み

送信データが送信データレジスタ (TDRn) から送信シフトレジスタに転送されて送信が開始されると , シリアルステータスレジスタ (SSR) の送信データレジスタエンプティフラグビット (TDRE) が "1" にセットされます。この場合 , SSR の送信割込み許可 (TIE) ビットが "1" にセットされていると , 割込み要求が生成されます。

なお , TDRE の初期値 (ハードウェアリセット後またはソフトウェアリセット後) は "1" です。そのため , TIE フラグを "1" にセットすると即座に割込みが生成されます。また , TDRE フラグをリセットする唯一の方法は送信データレジスタ (TDRn) へのデータの書込みです。

バスアイドル割込み

SINn 端子で受信動作が行われていない場合には , ECCRn:RBI ビットと ESIRn:RBI ビットが "1" にセットされます。ECCRn:RBI は受信が開始されると即座にクリアされるのに対して , ESIRn:RBI は "0" を書き込むことで明示的にクリアする必要があります。

SOTn 端子で送信動作が行われていない場合には , ECCRn:TBI が "1" にセットされます。送信が開始されると , ECCRn:TBI はクリアされます。

ESIRn:AICD=0 で ECCRn:BIE=1 の場合には , ECCRn:RBI と ECCRn:TBI の両方が "1" になると割込みが生成されます。ECCRn:RBI または ECCRn:TBI のいずれかがクリアされると , 割込みはクリアされます。割込み処理ソフトウェアが割込みをクリアできるようになる前に ECCRn:RBI がクリアされる場合があるため , AICD =0 でバスアイドル割込みを使用しないでください。

ESIRn:AICD =1 で ECCRn:BIE =1 の場合には , ESIRn:RBI と ECCRn:TBI の両方が "1" になると割込みが生成されます。ESIRn:RBI または ECCRn:TBI のいずれかがクリアされると , 割込みはクリアされます。ESIRn:RBI と ECCRn:TBI の両方が割込みサービスルーチンによって確認されるため , バスアイドル割込みは ESIRn:AICD = 1 で使用してください。

LIN Synchronization Break割込み

この段落は , USART がモード 3 で LIN スレーブとして動作している場合にのみ関係します。

バス (シリアル入力) が 11 ビット時間以上 "0" (ドミナント) になると , 拡張ステータス制御レジスタ (ESCRn) の LIN break 検出 (LBD) フラグビットが "1" にセットされます。なお , この場合には 9 ビット時間が経過したところで受信エラーフラグが "1" にセットされるため , LIN synch break 検出のみを望む場合には RXE フラグを "0"

にセットしておく必要があります。

割込みおよびESCRn:LBDフラグはLBDフラグへの"0"の書込みでクリアされます。これはLIN synch fieldに対するインプットキャプチャ割込みの前に実行する必要があります。

LIN Synchronization Fieldエッジ検出割込み

この段落は, USARTがモード3でLINスレーブとして動作している場合にのみ関係します。LIN break検出の後, 受信バスの次の立下りエッジがUSARTによって示されます。同時に, LIN-USARTに関連付けられたICUに接続された内部信号が"1"にセットされます(どのICUがどのLIN-USARTに関連付けられるかについては,「1.6 "LIN-USART のインプットキャプチャユニットのソースの選択 "」の項を参照)。

この信号はLIN synch fieldの5番目の立下りエッジの後に"0"にリセットされます。どちらの場合も ,「両エッジ検出」とICU割込みが許可されている場合には, 関連付けられたICUが割込みを生成します。ICUカウンタ値の相違はシリアルクロックの8倍です。これを8で割ると専用リロードカウンタのためのボーレートを算出できます。この値から1を引いた値をボーレート生成レジスタ (BGRn) に書き込みます。リロードカウンタはスタートビットの立下りエッジが検出されると自動的にリセットされるため, リロードカウンタをリスタートする必要はありません。

20.5.2 USART DMA機能

USARTは, 受信または送信のためにDMAと連携できます。DMAの詳細については,「第4章 DMA」を参照してください。

＜注意事項＞

USARTとDMAを連携させる場合は, ESIRn:AICD =1に設定してください。
また受信中にエラーが発生すると , 対応するDMAチャネルのDMACSレジスタのSEビットが設定されている場合, 実行中のDMA転送が停止されます。

20.5.3 受信割込みの生成とフラグのセットタイミング

受信割込みの要因としては, 受信完了(SSRn:RDRF)と受信エラーの発生(SSRn:PE, ORE, FRE)があります。

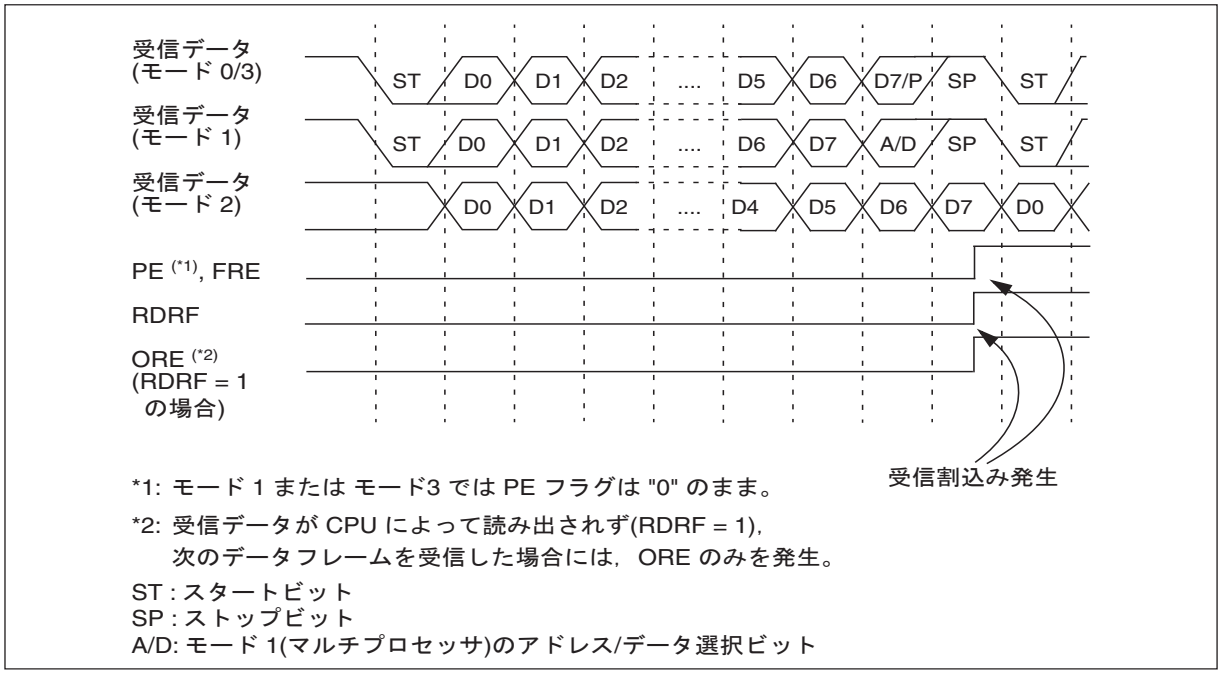
受信割込みの生成とフラグのセットタイミング

データの受信が完了し (ESIRn:AICD =0 の場合にはSSRn:RDRF=1, ESIRn:AICD =1 の場合にはESIRn:RDRF =1), シリアルステータスレジスタ (SSRn) の受信割込み許可 (RIE) フラグビットが"1"に設定されている場合には, 一般に受信割込みが生成されます。この割込みは, モード0, 1, 2 (SSM = 1 の場合), 3では最初のストップビットが検出されたとき, モード2 (SSM = 0 の場合) では最後のデータビットが読み取られたときに生成されます。

＜注意事項＞

受信エラーが発生した場合には, いずれのモードでも受信データレジスタ (RDRn) には不正なデータが含まれます。

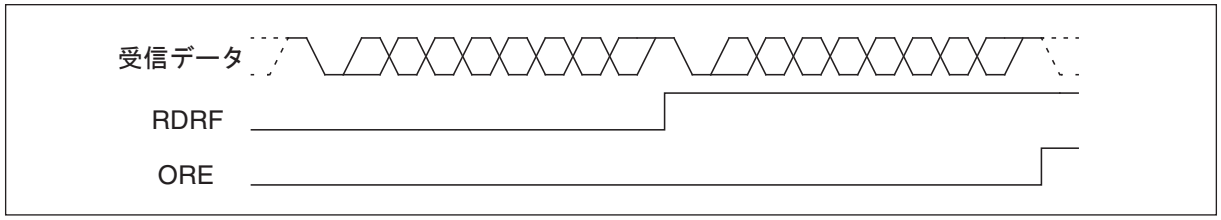
Figure 20-11. 受信動作とフラグのセットタイミング



＜注意事項＞

Figure 20-11 の例には，モード 0 とモード 3 におけるすべての受信オプションを表しているわけではありません。"7p1" と "8N1"(p は "E"[偶数] または "O"[奇数]) の例のみ示されています。

Figure 20-12. OREのセットタイミング:



20.5.4 送信割込みの生成とフラグセットタイミング

送信割込みは， 送信データが送信データレジスタ(TDRn)から送信シフトレジスタに転送されて送信が開始されたときに生成されます。

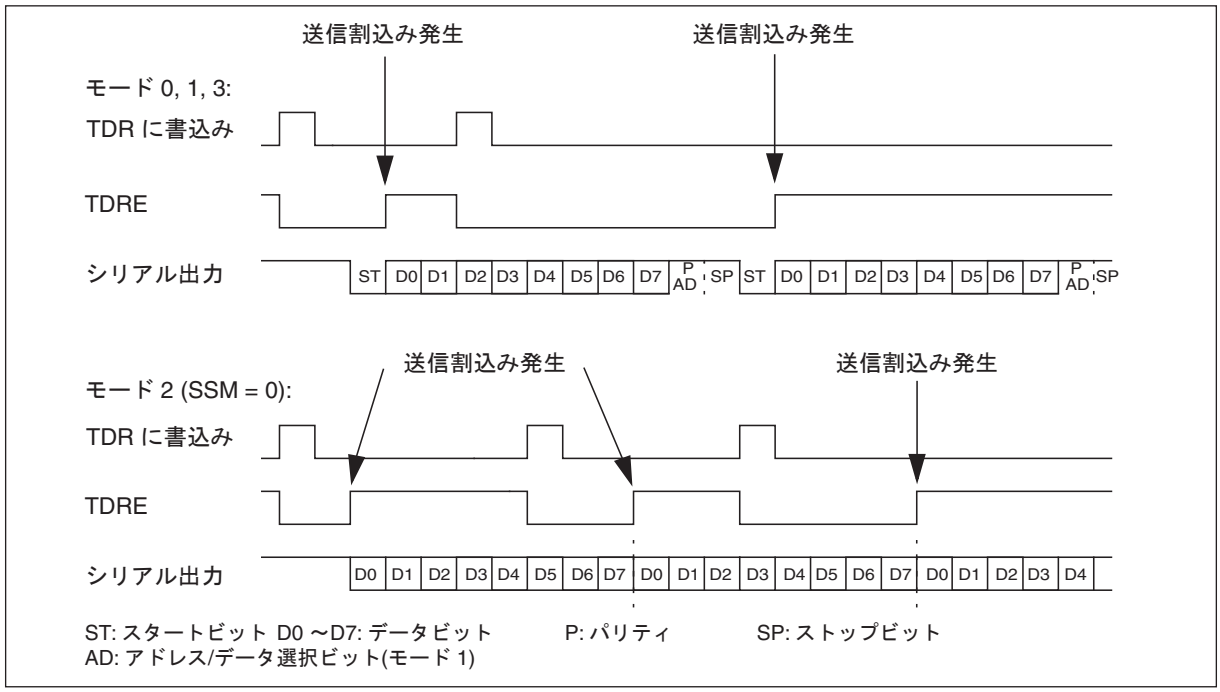
送信割込みの生成とフラグのセットタイミング

送信割込みは, 送信する次のデータを送信データレジスタ (TDRn) に書き込む準備ができたとき, すなわち TDRn が空であり , シリアルステータスレジスタ (SSRn) の送信割込み許可 (TIE) ビットが "1" に設定されることで送信割込みが許可されている場合に生成されます。

SSRn の送信データレジスタエンプティ(TDRE) フラグビットは TDRn が空であることを示します。TDRn にデータを書き込むと TDRE ビットがクリアされます。

USART の 4 つのモードでの送信動作とフラグのセットタイミングを下図に示します。

Figure 20-13. 送信動作とフラグのセットタイミング



＜注意事項＞

Figure 20-13 の例には , モード 0 におけるすべての送信オプションを表しているわけではありません。"8p1" (p は "E"[偶数] または "O"[奇数]) についてのみ示されています。
SSM = 0 の場合には , モード 3 とモード 2 ではパリティは使用できません。

送信割込み要求の生成タイミング

送信割込みが許可されている場合 (SSRn: TIE=1) に TDRE フラグが "1" にセットされると , 送信割込み要求が生成されます。

＜注意事項＞

TDRE ビットは初期状態で "1" にセットされているため , 送信割込み許可 (TIE=1) 直後に送信完了割込みが生成されます。送信割込みを許可するタイミングに注意してください。

20.6 USARTのボーレート

USART のシリアルクロックソースとしては , 以下のいずれかを選択できます。

- 専用ボーレートジェネレータ(リロードカウンタ)
- 外部クロック(SCKn 端子へのクロック入力)
- ボーレートジェネレータ(リロードカウンタ)に接続された外部クロック

20.6.1 USARTのボーレート選択

ボーレート選択回路を下図に示します。以下の 3 つのタイプのボーレートのうちの 1 つを選択できます。

専用ボーレートジェネレータ(リロードカウンタ)によるボーレート

USART には送信と受信のシリアルクロック用に , 2 つの独立した内部リロードカウンタがあります。ボーレートは , ボーレートジェネレータレジスタ (BGRn) に設定された 16 ビットのリロード値で指定されます。

リロードカウンタは , ボーレートジェネレータレジスタに設定された値で周辺クロック (CLKP1) を分周します。

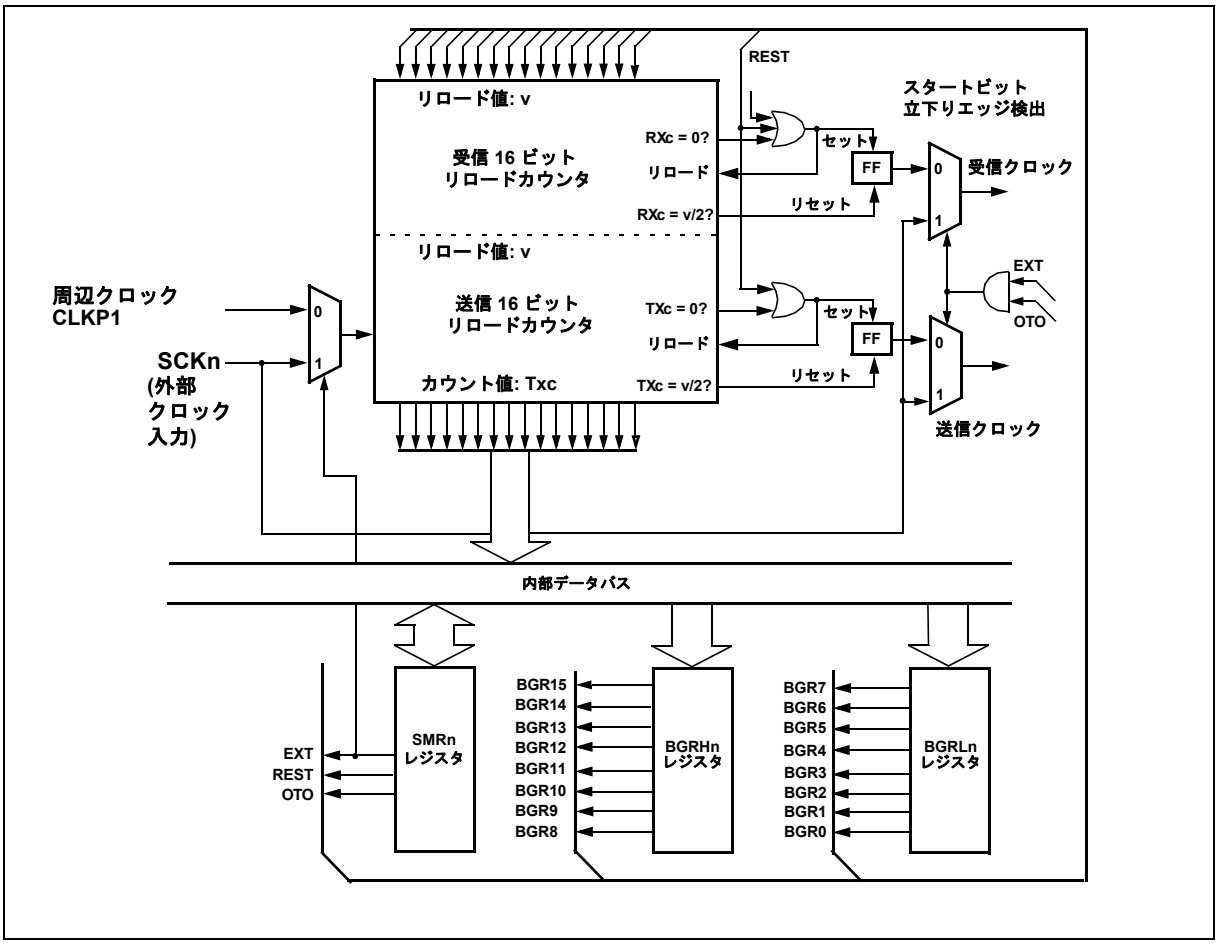
外部クロック(1対1モード)によるボーレート

USART クロックパルス入力端子 (SCK) から入力されたクロックが使用されます (同期モード)。CLKP1 を 4 分周したクロック未満の周波数で , 分周可能な任意のボーレートのクロックを外部から入力できます。

専用ボーレートジェネレータと外部クロックによるボーレート

外部クロックソースは内部でリロードカウンタに接続することもできます。このモードでは , 外部クロックが内部周辺クロック (CLKP1) の代わりに使用されます。これは , 分周することのできる特別な周波数の , 水晶振動子を使用するために設計されています。

Figure 20-14. ボーレート選択回路(リロードカウンタ)



20.6.2 ボーレートの設定

ボーレートの設定方法とシリアルクロック周波数の計算結果を示します。

20.6.2.1 ボーレートの計算

2 つの 16 ビットリロードカウンタは , ボーレートジェネレータレジスタ (BGRn) で設定します。目的のボーレートを設定するには , 次の計算式を使用します。

リロード値 :

v = [ϕ / b] - 1,

ここで , ϕ は周辺クロック (CLKP1), b はボーレート , [] はガウス括弧 (数学的丸め関数)

■ **計算例**

周辺クロック (CLKP1) が 16 MHz で目的のボーレートが 19200 bps の場合は , リロード値 v は次のように計算します。

$v = [16 \times 10^6 / 19200] - 1 = 832.3333$

正確なボーレートは次のように再計算できます。$b_{exact} = \phi / (v + 1)$ に数値を当てはめた結果は : $16 \times 10^6 / 833 = 19207.6831$

＜注意事項＞

リロード値を "0" に設定するとリロードカウンタは停止します。このため , 最小分周比は "2" です。非同期通信では内部的に 5 回オーバサンプリングが行われるため , リロード値は "4" 以上である必要があります。

20.6.2.2 各種周辺クロック(CLKP)周波数に対する分周比とボーレート

各種周辺クロック(CLKP)周波数に対する分周比とボーレートに対する推奨設定を以下に示します。

Table 20-13. 各種周辺クロック(CLKP)周波数に対するボーレートとリロード値

ボーレート	8 MHz		10 MHz		16 MHz		20 MHz		24 MHz	
	値	偏差	値	偏差	値	偏差	値	偏差	値	偏差
4M	-	-	-	-	-	-	4	0	5	0
2M	-	-	4	0	7	0	9	0	11	0
1M	7	0	9	0	15	0	19	0	23	0
500000	15	0	19	0	31	0	39	0	47	0
460800	-	-	-	-	-	-	-	-	51	-0.16
250000	31	0	39	0	63	0	79	0	95	0
230400	-	-	-	-	-	-	-	-	103	-0.16
153600	51	-0.16	64	-0.16	103	-0.16	129	-0.16	155	-0.16
125000	63	0	79	0	127	0	159	0	191	0
115200	68	-0.64	86	0.22	138	0.08	173	0.22	207	-0.16
76800	103	-0.16	129	-0.16	207	-0.16	259	-0.16	311	-0.16
57600	138	0.08	173	0.22	277	0.08	346	-0.06	416	0.08
38400	207	-0.16	259	-0.16	416	0.08	520	0.03	624	0
28800	277	0.08	346	<0.01	554	-0.01	693	-0.06	832	-0.03
19200	416	0.08	520	0.03	832	-0.03	1041	0.03	1249	0
10417	767	<0.01	959	<0.01	1535	<0.01	1919	<0.01	2303	<0.01
9600	832	0.04	1041	0.03	1666	0.02	2083	0.03	2499	0
7200	1110	<0.01	1388	<0.01	2221	<0.01	2777	<0.01	3332	<0.01
4800	1666	0.02	2082	-0.02	3332	<0.01	4166	<0.01	4999	0
2400	3332	<0.01	4166	<0.01	6666	<0.01	8332	<0.01	9999	0
1200	6666	<0.01	8334	0.02	13332	<0.01	16666	<0.01	19999	0
600	13332	<0.01	16666	<0.01	26666	<0.01	-	-	-	-
300	26666	<0.01	-	-	-	-	-	-	-	-

＜注意事項＞

- 1) 偏差は % 単位です。
- 最大同期ボーレート : 周辺クロック (CLKP1) 周波数 / 5

20.6.2.3 外部クロックの使用

SMRn レジスタの EXT ビットをセットすると , SCKn 端子に接続された外部クロックが選択されます。外部クロックは , 周辺クロック (CLKP1) と同様にしてボーレートリロードカウンタに入力されます。

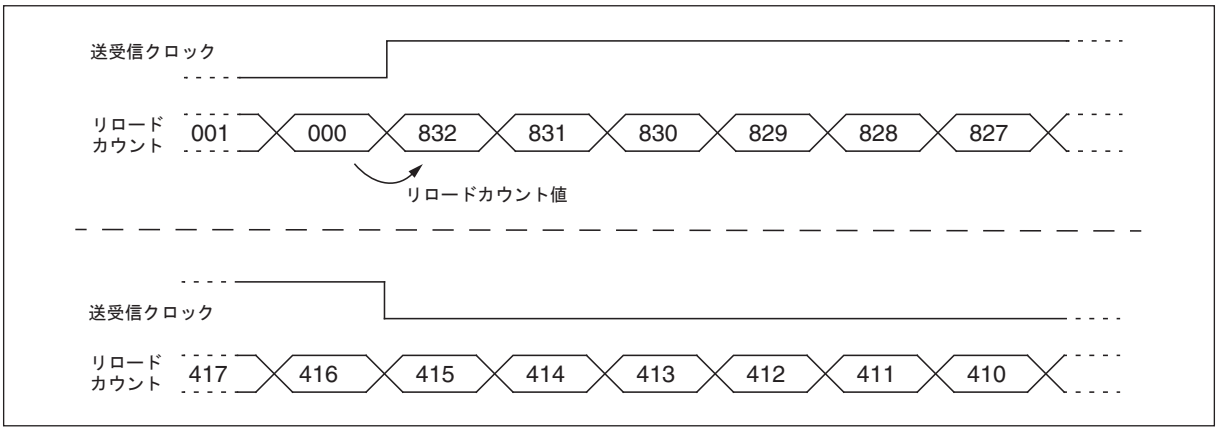
1 対 1 外部クロック入力モード (SMRn:OTO) を選択した場合には , SCKn 信号は USART のシリアルクロック入力に直接接続されます。これは USART が同期モード 2 でスレーブデバイスとして動作する場合に必要とされます。

なお , いずれのケースでも最終的に USART モジュール内で使用されるクロック信号は , 周辺クロック (CLKP1) に同期されます。このため , 分周不能なクロックレートを入力すると位相が不安定な信号を生じる結果となります。

20.6.2.4 カウント例

リロード値は 832 であるとします。Figure 20-15 に , 両リロードカウンタの動作を示します。

Figure 20-15. リロードカウンタのカウント例



＜注意事項＞

シリアルクロック信号の立下りエッジは , 常に { (v + 1) / 2 } 後に発生します。

20.6.3 リロードカウンタのリスタート

リロードカウンタは以下の場合にリスタートされます。
送信リロードカウンタおよび受信リロードカウンタ :

- MCU 全体のリセット
- USART のプログラマブルクリア(SMRn:UPCL ビット)
- ユーザプログラマブルリスタート(SMRn:REST ビット)

受信リロードカウンタ :

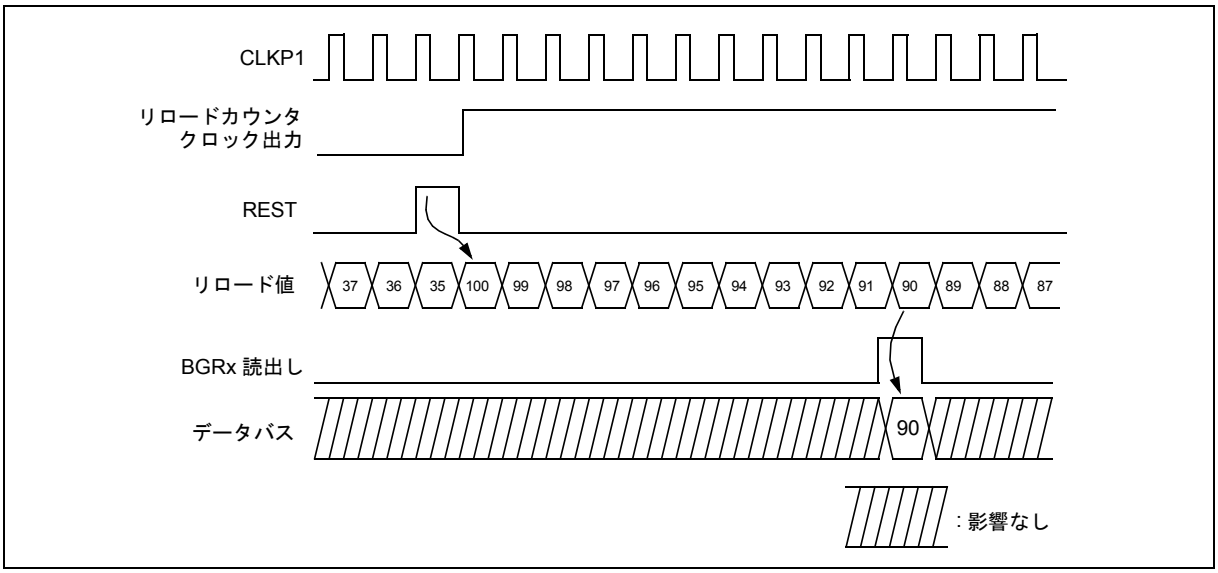
- 非同期モードでのスタートビットの立下りエッジの検出

20.6.3.1 プログラマブルリスタート

シリアルモードレジスタ (SMRn) の REST ビットをユーザがセットすると , 両リロードカウンタは次のクロックサイクルでリスタートされます。この機能は , 送信リロードカウンタを小さなタイマとして使用することを意図しています。

この機能の使用法を下図に示します (リロード値は 100 であると仮定)。

Figure 20-16. リロードカウンタのリスタート例



この例では , REST 後の周辺クロック (CLKP1) サイクル数 (cyc) は次の値になります。

cyc = v - c + 1 = 100 - 90 + 1 = 11

ここで v はリロード値であり , c はリロードカウンタ値です。

＜注意事項＞

SMRn:UPCL のセットによって USART をリセットした場合もリロードカウンタはリスタートされます。

リロードカウンタのクリア

ボーレートリロードカウンタレジスタ (BGRn) とボーレートリロードカウンタは , MCU 全体のリセットで "0" にクリアされ , カウンタは停止します。リロードカウンタは , SMRn レジスタの UPCL ビットへの "1" の書込みで "0" にクリアされます。リロードレジスタに格納されている値は代わりませんが , カウンタは即座にリロード値からカウントを開始します。REST ビットへの "0" の書込みではカウンタはクリアされず , 即座にリロード値からリスタートされます。

20.7 USARTの動作

USART は, モード 0 では通常の双方向シリアル通信用に動作し, モード 2 とモード 3 ではマスタまたはスレーブとして双方向通信を行い, モード 1 ではマルチプロセッサ通信でのマスタまたはスレーブとして動作します。

USARTの動作

■ 動作モード

USART には, モード 0 ～ 3 の 4 つの動作モードがあります。Table 20-14 に示すように, 動作モードは通信方式に従って選択できます。

Table 20-14. USARTの動作モード

動作モード		データ長		同期方式	ストップビット長	データビット方向 *1
		パリティなし	パリティあり			
0	ノーマルモード	7 ビットまたは 8 ビット		非同期	1 ビットまたは 2 ビット	L/M
1	マルチプロセッサモード	7 ビットまたは 8 + 1 ビット*2	-	非同期	1 ビットまたは 2 ビット	L/M
2	ノーマルモード	8ビット		同期	なし, 1 ビット, 2 ビット	L/M
3	LINモード	8ビット	-	非同期	1ビット	L

*1: データビット転送フォーマットが LSB ファーストか MSB ファーストかを示します。
*2:"+1" はパリティではなく, マルチプロセッサモードでのアドレス/データ選択ビットを意味します。

＜注意事項＞

モード 1 は , マスタ / スレーブ接続システムでの USART のマスタとスレーブのいずれの動作にも対応します。モード 3 では , USART の機能は 8N1, LSB ファーストに固定されます。

モードが変更された場合には , USART はすべての送受信を打ち切って待機状態となります。

CPU間接続方式

外部クロック 1 対 1 接続 (ノーマルモード) とマスタ / スレーブ接続 (マルチプロセッサモード) を選択できます。どちらの接続方式でも, データ長, パリティの有無, 同期方式はすべての CPU で同一である必要があります。動作モードは以下のように選択します。

- 1 対 1 接続方式では, 両 CPU は動作モード 0 または 2 を使用する必要があります。 非同期転送モードではモード 0, 同期転送モードではモード 2 を選択してください。
  なお同期モード 2 では, 片方の CPU をマスタに, 他方をスレーブに構成する必要があります。
- マスタ/スレーブ接続方式では動作モード 1 を選択し, マスタまたはスレーブとして使用します。

同期方式

非同期動作では , USART の受信クロックは受信したスタートビットの立下りエッジに自動的に同期されます。

同期モードでは , 同期はマスタデバイスのクロック信号またはマスタとして動作する USART 自身によってなされます。

信号方式

USART はデータのみを NRZ (Non Return Zero) フォーマットで扱えます。

動作許可ビット

USART は送信と受信を , 送信用 (SCRn: TXE) と受信用 (SCRn: RXE) の動作許可ビットで制御します。

- 受信中(受信シフトレジスタにデータ入力中)に受信動作を禁止する場合には, フレームの受信が終了するのを待って受信データレジスタ(RDRn)の受信データを読み出し, その後受信動作を停止してください。
- 送信中(送信シフトレジスタからデータ送出中)に送信動作を禁止する場合は, 送信データレジスタ(TDRn)内のデータがなくなるのを待ってから送信動作を停止してください。

20.7.1 非同期モード(動作モード0, 1)での動作

USART を動作モード 0 (ノーマルモード)または動作モード 1 (マルチプロセッサモード)で使用する場合は, 非同期転送モードが選択されます。

20.7.1.1 非同期モードの動作

■ T転送データフォーマット

一般に非同期モードでの各データ転送はスタートビットバス上の ("L" レベル) で始まり , 少なくとも 1 ビットのストップビット ("H" レベル) で終了します。ビットストリームの方向 (LSB ファーストまたは MSB ファースト) はシリアルステータスレジスタ (SSRn) の BDS ビットで設定します。パリティビット (使用する場合) は常に , 最後のデータビットと (最初の) ストップビットの間に置かれます。

動作モード 0 では , データフレームの長さは 7 または 8 ビット , パリティありまたはなし , ストップビットは 1 または 2 ビットです。

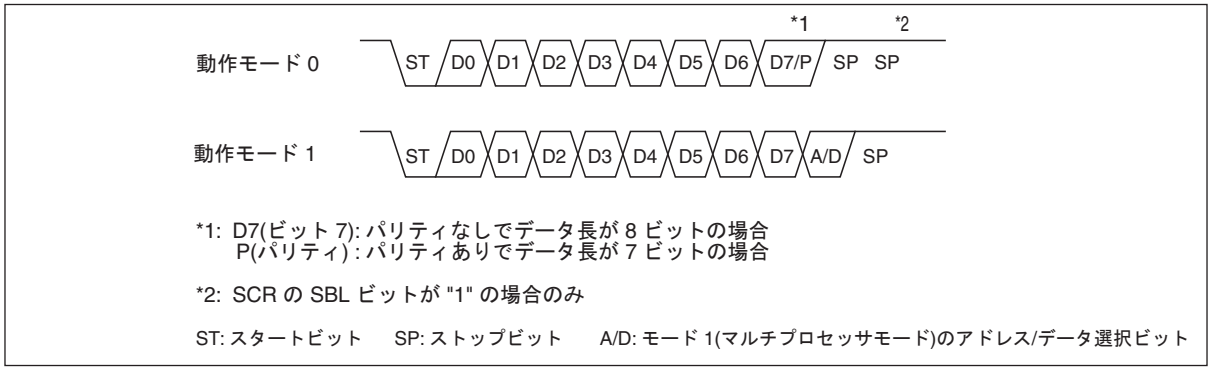
動作モード 1 では , データフレームの長さは 7 または 8 ビットで , パリティビットの代わりにアドレス / データ選択ビットが付随します。ストップビットは 1 または 2 ビットを選択できます。

転送フレームのビット長の計算式は以下のようになります。

長さ = 1 + d + p + s

(d = データビット数 [7 または 8], p = パリティ [0 または 1], s = ストップビット数 [1 または 2])

Figure 20-17. 転送データフォーマット(動作モード0 および1))



＜注意事項＞

- シリアルステータスレジスタ (SSRn) の BDS ビットが "1" にセットされていると (MSB ファースト), ビットストリームは D7, D6, ..., D1, D0, (P) という順序になります。

受信中ストップビット 2 ビットが選択されている場合には , 両方のストップビットが検出されます。ただし , 受信データレジスタフル (RDRF) フラグは最初のストップビットで "1" になります。バスアイドルフラグ (ECCRn の RBI) は , スタートビットが検出されない場合には 2 つ目のストップビットの後で "1" になります (2 つ目のストップビットはマークレベルですが , 「バスアクティビティ」 に属します)。

■ 送信動作

シリアルステータスレジスタ (SSRn) の送信データレジスタエンプティ (TDRE) フラグビットが "1" の場合には , 送信データを送信データレジスタ (TDRn) に書き込むことができます。データを書き込むと , TDRE フラグは "0" になります。シリアル制御レジスタ (SCRn) の TXE ビット ("1") で送信動作が許可されていると , データは送信シフトレジスタに書き込まれ , 次のシリアルクロックサイクルで , スタートビットから送信が開始されます。それにより , TDRE フラグが "1" になると , 新たなデータを TDRn に書き込むことができます。

送信割込みが許可されている場合 (TIE = 1) には , TDRE フラグによって割込みが生成されます。なお , TDRE フラグの初期値は "1" であるため , TIE が "1" にセットされると即座に割込みが発生します。

キャラクタ長が 7 ビットに設定されている場合(CL=0)には, BDS ビットの方向設定(LSB ファーストまたは MSB ファースト) にかかわらず , TDRn の使用されないビットは常に MSB です。

■ 受信動作

受信動作は , SCRn の受信許可 (RXE) フラグビットで許可されている場合に実行されます。スタートビットが検出されると, SCRn に設定されたフォーマットに従ってデータフレームが受信されます。エラー発生時には, 対応するエラーフラグがセットされます (PE, ORE, FRE)。データフレームの受信後に , データはシリアルシフトレジスタから受信データレジスタ (RDRn) に転送され , SSRn レジスタと ESIRn レジスタの受信データレジスタフル (RDRF) フラグビットがセットされます。

受信割込みが許可されていて(RIE = 1), ESIRn:AICD =0の場合には, SSRn:RDRF によって割込みが生成されます。

受信割込みが許可されていて (RIE = 1), ESIRn:AICD =1 の場合には , ESIRn:RDRF によって割込みが生成されます。

その後データは CPU によって読み出される必要があります。そうすることにより , SSRn:RDRF フラグがクリアされます。

ESIRn:AICD =0 の場合には , これによって割込みもクリアされます。

ESIRn:AICD =1 の場合には , 割込みは ESIRn:RDRF への "1" の書込みでクリアする必要があります。

キャラクタ長が 7 ビットに設定されている場合(CL=0)には, BDS ビットの方向設定(LSB ファーストまたは MSB ファースト) にかかわらず , RDRn の使用されないビットは常に MSB です。

＜注意事項＞

RDRF フラグビットがセットされており , エラーが発生していない場合にのみ , 受信データレジスタ (RDRn) のデータが有効です。

■ 使用されるクロック

内部クロックまたは外部クロックを使用します。使用するボーレートにしたがってボーレートジェネレータを選択します (SMRn:EXT = 0 または 1, SMRn:OTO = 0)。

■ ストップビット, エラー検出, パリティ:

ストップビット長は , SCRn レジスタの SBL ビットにより , 1 または 2 を設定できます。ストップビット 2 ビットが指定されている場合の受信では, 最初のストップビットに加えて 2 つ目のストップビットもチェックされます。RBI (バスアイドル) フラグは 2 つ目のストップビットの後でセットされます。ただし , RDRF フラグは最初のストップビットを受信したときにセットされます。モード 0 では , パリティエラー, オーバランエラー, フレーミングエラーがチェックされます。モード 1 では , パリティチェックは行われず , オーバランエラーとフレーミングエラーがチェックされます。モード 0 では SCRn レジスタの PEN ビットでパリティビットの有無を指定し , P ビットで偶数パリティまたは奇数パリティを指定します。

20.7.2 同期モード(動作モード2)での動作

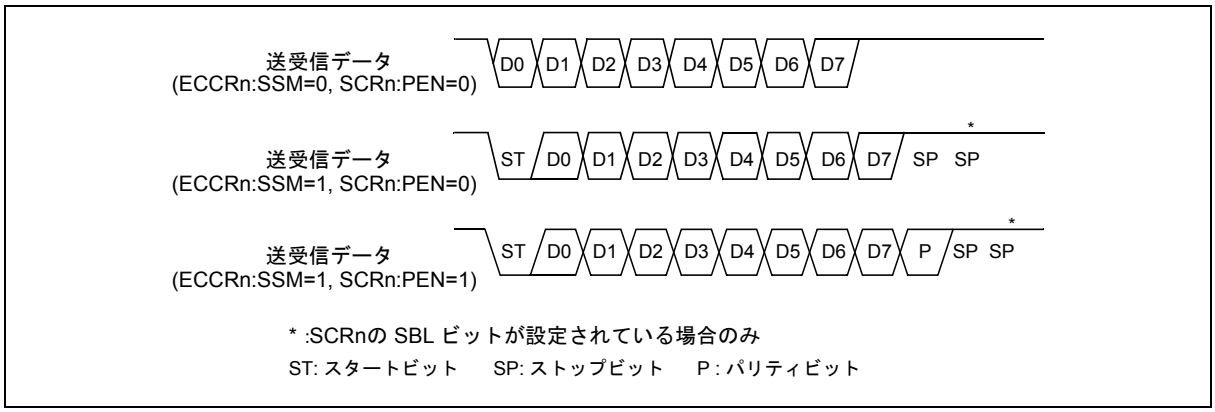
USART の動作モード 2 (ノーマルモード)では, クロック同期転送方式が使用されます。

20.7.2.1 同期モード(動作モード2) での動作

■ 転送データフォーマット

同期モードでは , 拡張通信制御レジスタ (ECCRn) の SSM ビットが "0" の場合には , スタートビットとストップビットなしで 8 ビットデータが転送されます。同期動作モードでのデータ転送フォーマットを下図に示します。

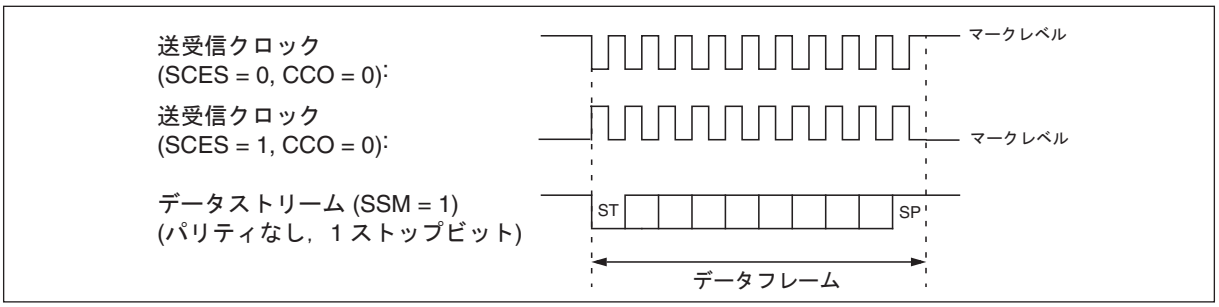
Figure 20-18. 転送データフォーマット(動作モード2)



■ モード 2 でのクロック反転とスタート/ストップビット

拡張ステータス制御レジスタ (ESCRn) の SCES ビットをセットすると , シリアルクロックが反転します。そのため , スレーブモードの USART は , 受信したシリアルクロックの立下りエッジでデータビットをサンプリングします。なお , SCES をセットすると , マスタモードのクロック信号のマークレベルは "0" になります。拡張通信制御レジスタ (ECCRn) の SSM ビットをセットすると , 非同期モードのように , データフォーマットにスタートビットとストップビットが付加されます。

Figure 20-19. クロック反転した転送データフォーマット



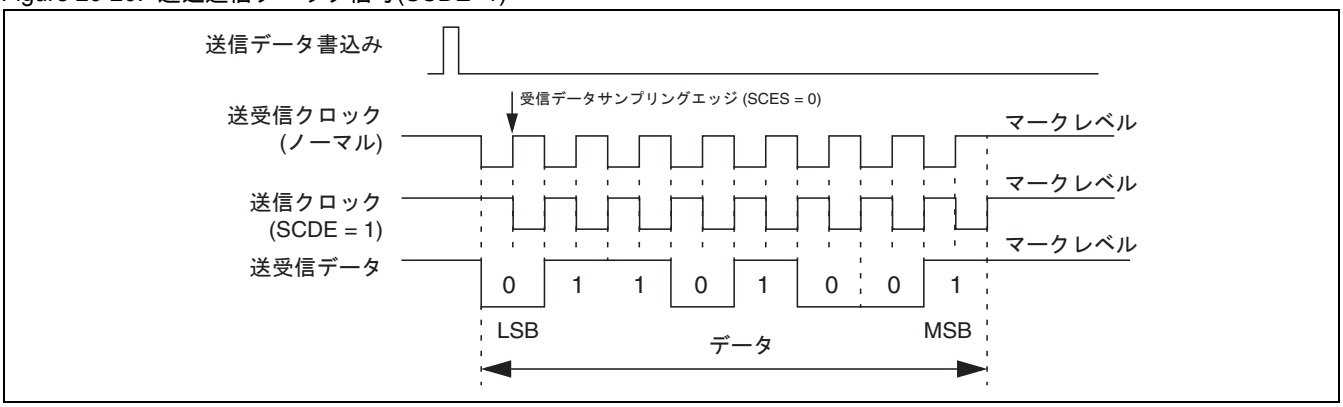
■ クロック供給

動作モード 2 では , クロック信号のクロックサイクル数は , スタートビットとストップビットを含むデータのビット数と同一である必要があります。ECCRn レジスタの MS ビットが "0" (マスタモード) で SMRn レジスタの SCKE ビットが "1" (クロック出力許可) の場合には , 安定したクロックサイクルが自動的に生成されます。ECCRn レジスタの MS ビットが "1" (スレーブモード) の場合には , ほかの通信デバイスで適切なクロックサイ

クルが生成されるようにしてください。通信が行われていない時には , クロック信号はマークレベルの "1" に維持される必要があります。

ECCRn レジスタの SCDE ビットを "1" に設定すると , Figure 20-20 に示すようにクロック出力信号はシリアルクロックの半サイクル分遅延します。この動作は, シリアルクロック信号の立下りエッジでデータサンプリングを行う通信デバイスのために用意されています。

Figure 20-20. 遅延送信クロック信号(SCDE=1)



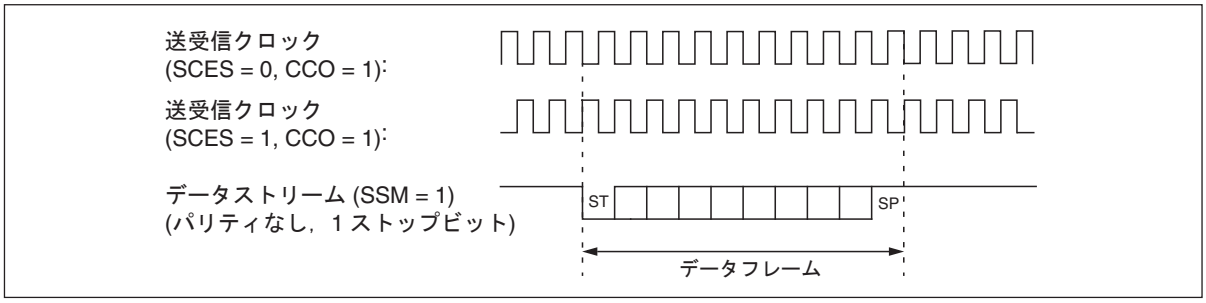
lESCRnレジスタの SCES ビットを "1" に設定すると, シリアルクロック信号は反転します。受信データはシリアルクロックの立下りエッジでサンプリングされます。

ECCRn レジスタの MS ビットが "0" (マスタモード) で SMRn レジスタの SCKE ビットが "1" (クロック出力許可) の場合には , 出力クロック信号も反転されます。

通信が行われていないときには , クロック信号はマークレベルの "0" に維持される必要があります。

ESCRn レジスタの CCO ビットを "1" に設定すると , シリアルクロックはデータ通信が行われていない時にも出力されます。そのため , Figure 20-21 に示すようにスタート / ストップビットの使用が推奨されます。

Figure 20-21. モード2での連続クロック出力



■ エラー検出:

スタート / ストップビットが選択されない場合 (ECCRn: SSM = 0) には , オーバランエラーのみが検出されます。

■ 通信:

同期モードを初期化するには , 以下の設定を行う必要があります。

ボーレートジェネレータレジスタ (BGRn):

専用ボーレートリロードカウンタへのリロード値の設定。

シリアルモード制御レジスタ (SMRn):

MD1, MD0: "10_B" (モード 2)

SCKE: "1" (専用ボーレートリロードカウンタ使用)

"0" (外部クロック入力使用)

SOE: "1" (送受信)

"0" (受信のみ)

シリアル制御レジスタ (SCRn):

RXE, TXE: 両フラグを "0" に設定

A/D: アドレス / データ選択なし - 無効

CL: 自動的に 8 ビットデータに固定 - 無効

CRE: 受信エラーフラグをクリアするために "1"

-- SSM=0 の場合 (初期値):

PEN, P, SBL: 無効

-- SSM=1 の場合 :

PEN: パリティビットを付加 / 検出する場合は "1", 使用しない場合は "0"

P: 偶数パリティの場合は "0", 奇数パリティの場合は "1"

SBL: 2 ストップビットの場合は "1", 1 ストップビットの場合は "0"

シリアルステータスレジスタ (SSRn):

BDS: LSB ファーストの場合は "0", MSB ファーストの場合は "1"

RIE: 割込みを使用する場合は "1", 受信割込みを禁止する場合は "0"

TIE: 割込みを使用する場合は "1", 送信割込みを禁止する場合は "0"

拡張通信制御レジスタ (ECCRn):

SSM: スタート / ストップビットを使用しない場合 (通常) は "0", 付加する場合 (特別) は "1"

MS: マスタモード (USART がシリアルクロックを生成) では "0", スレーブモード (USART はマスタデバイスからシリアルクロックを受信) では "1"

シリアル制御レジスタ (SCRn):

RXE, TXE: 通信を開始するには片方または両方の制御ビットを "1" に設定します。

20.7.3 LIN機能(動作モード3)での動作

USARTはLINマスタまたはLINスレーブとして動作します。LIN機能では特別なモードが使用されます。USARTをモード 3 に設定すると, データフォーマットは 8N1, LSB ファーストに設定されます。

20.7.3.1 非同期LINモード(動作モード3) での動作

■ LIN マスタとしての動作

LIN マスタモードでは , マスタがサブバス全体のボーレートを決定するため , スレーブデバイスはマスタに同期する必要があります。そのため初期化後のマスタの動作によってボーレートは固定されます。

拡張通信制御レジスタ (ECCRn) の LBR ビットへの "1" の書込みにより , 13 ～
16 ビット時間の間 "L" レベルが SOT 端子に出力されます。これは LIN メッセージの開始を示す LIN synch break です。これによりシリアルステータスレジスタ (SSRn) の TDRE フラグが "0" になり , break 後に "1" にリセットされ , CPU に対して送信割込みが生成されます (SSRn の TIE が "1" の場合)。

送信される Synch break の長さは ESCRn の LBL1/LBL0 ビットで以下のように決定されます。

Table 20-15. LIN break長

LBL1	LBL0	Break長
0	0	13ビット時間
1	0	14ビット時間
0	1	15ビット時間
1	1	16ビット時間

LIN break 後にはバイトデータ 55_H の Synch field が送信されます。送信割込みを抑止するため , 55_H は LBR ビットに "1" を書いた直後に TDRE フラグが "0" の状態で TDRn に書き込むことができます。内部送信シフトレジスタは LIN break の終了を待機し , その後 TDRn 内のデータを送出します。この場合 , LIN break の後 , スタートビット送出前に割込みは生成されません。

■ LIN スレーブとしての動作

LIN スレーブモードでは , USART はマスタのボーレートに同期する必要があります。受信が禁止されている (RXE = 0)が, LIN break 割込みが許可されている(LBIE =1)場合には, USART は LIN マスタからの LIN synch break を検出すると受信割込みを生成し , ESCRn の LBD フラグに示します。このビットに "0" を書き込むと受信割込み要求がクリアされます。LIN スレーブは synch field からボーレートを計算する必要がある場合があります。この場合 , synch field の最初の立下りエッジから 5 番目の立下りエッジまでの時間がインプットキャプチャモジュールによって測定されます。このため , インプットキャプチャモジュールは内部的に LIN-USART に接続されています。この内部信号は最初の立下りエッジで "0" から "1" に変化し , 5 番目の立下りエッジで "1" から "0" に変化します。そのためインプットキャプチャモジュールは , 立上りエッジと立下りエッジの両方を検出するように設定する必要があります。また , LIN-USART からの信号を選択する必要もあります。インプットキャプチャモジュールで測定される時間は , ボーレートクロックサイクルの 8 倍です。

そのため , ボーレート設定値は次の値となります。

タイマオーバフローなし : BGRn 値 = (b-a)/8

タイマオーバフローあり : BGRn 値 = (max + b-a)/8

max はオーバフロー発生時のタイマの最大値

a は最初の割込み後の ICU カウンタレジスタの値

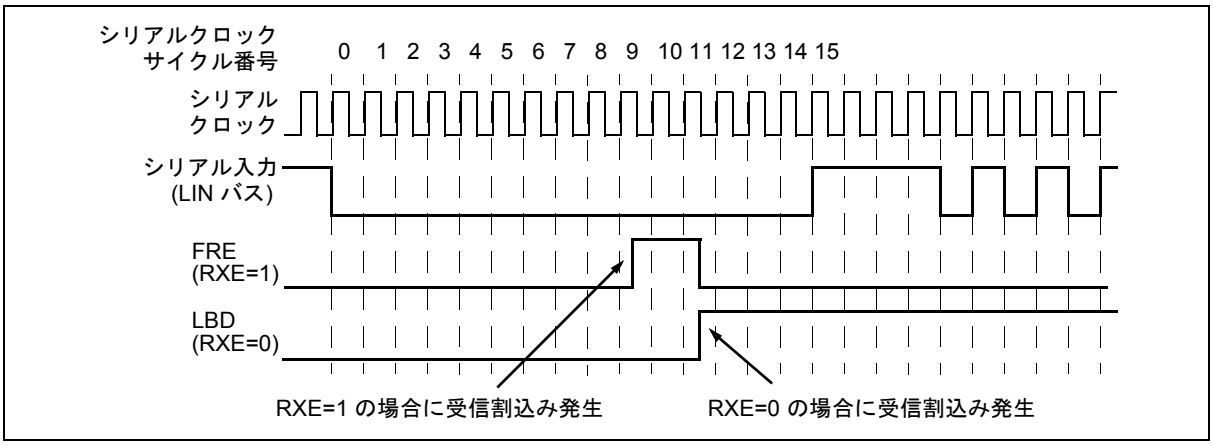
b は 2 回目の割込み後の ICU カウンタレジスタの値

このほかの USART と ICU の関係については , 「14.3 16 ビットフリーランタイマ」を参照してください。

■ LIN synch break 検出割込みとフラグ

スレーブモードで LIN synch break が検出されると , ESCRn の LIN break 検出 (LBD) フラグが "1" にセットされます。この場合 , LIN break 割込み許可 (LBIE) ビットがセットされていると割込みが発生します。

Figure 20-22. LIN synch break検出とフラグのセットタイミング



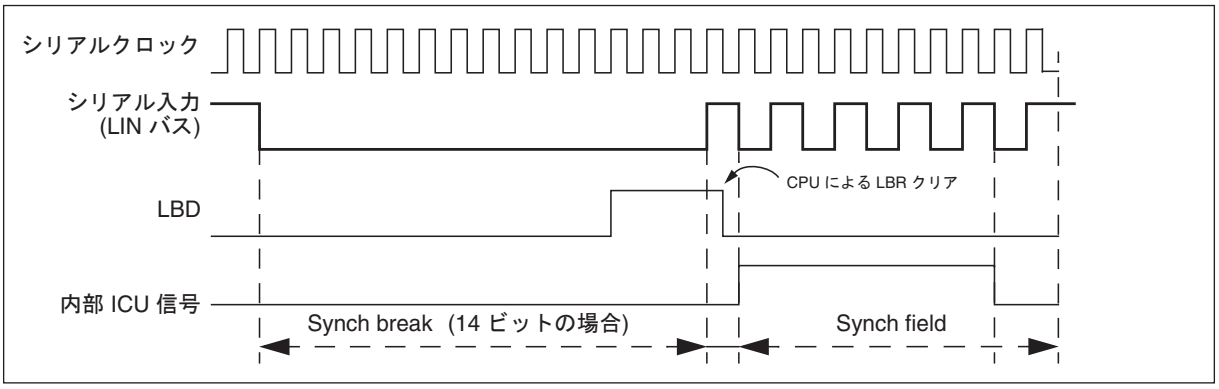
上図には LIN synch break 検出とフラグのセットタイミングが示されています。

なお , 受信が許可されていて (RXE = 1), 受信割込みも許可されている (RIE = 1) 場合には , SSRn の受信データフレーミングエラー(FRE) フラグビットが , LIN break 割込みよりも 2 ビット時間 ("8N1") 早く受信割込みを生成するため , LIN break を使用する場合は RXE をオフにすることを推奨します。

LBD は動作モード 3 でのみサポートされます。

Figure 20-23 に , 標準的な LIN メッセージフレームの開始と USART の動作を示します。

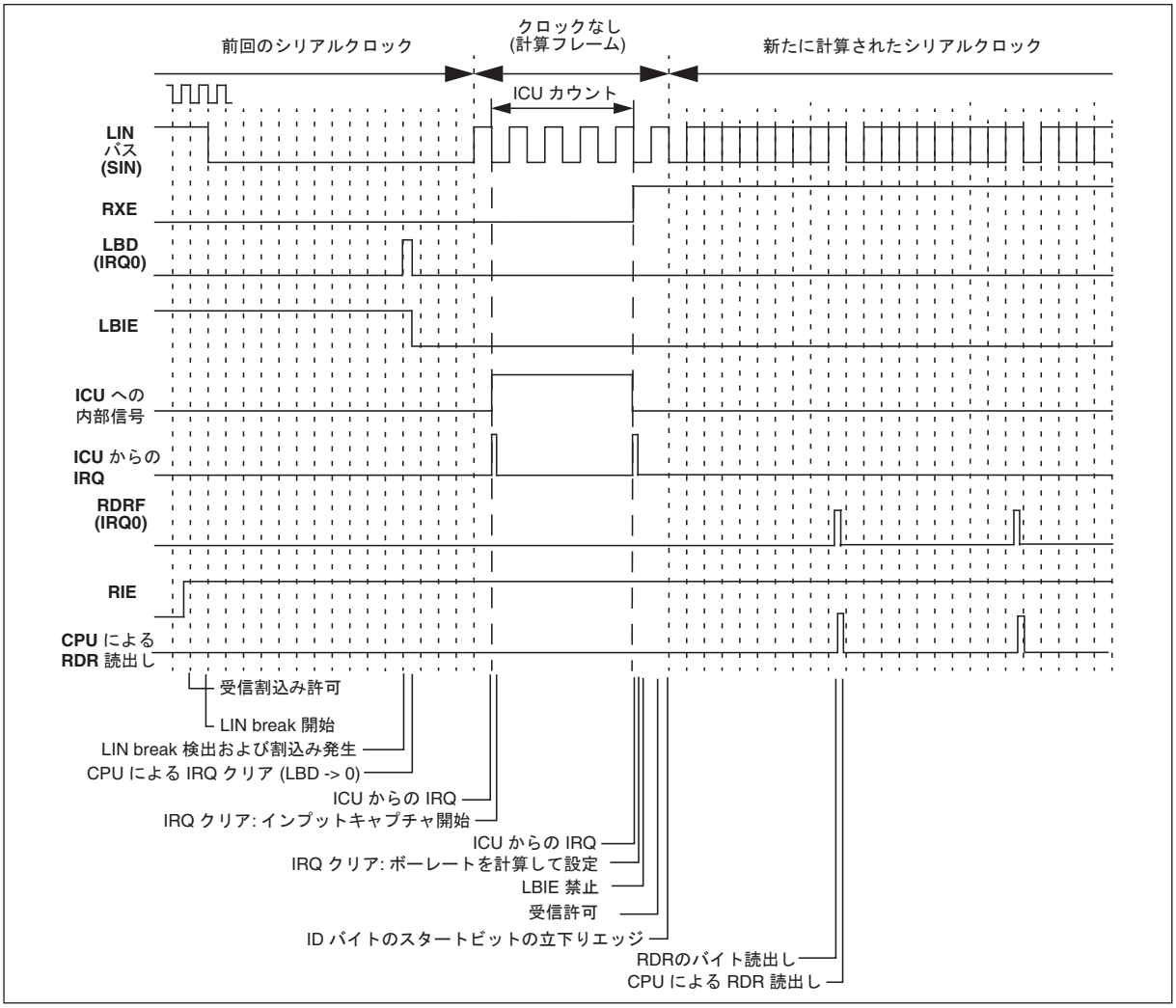
Figure 20-23. LINスレーブモードでのUSARTの動作



LIN Synch Break 検出とフラグ

■ LIN バスタイミング

Figure 20-24. LINバスタイミングとUSART信号



20.7.4 シリアル端子の直接アクセス

送信端子(SOTn)と受信端子(SINn)には直接アクセスできます。

USART直接端子アクセス

USART では , シリアル入力端子と出力端子にソフトウェアで直接アクセスできます。ソフトウェアでは ESCRn の SIOP ビットを読み出すことで , 入力シリアルデータを常時監視できます。ESCRn のシリアル出力端子直接アクセス許可 (SOPE) ビットをセットすると , ソフトウェアで強制的に SOTn 端子のレベルを変更できます。なお , このアクセスは送信シフトレジスタが空の (送信が行われていない) 場合にのみ可能です。

LIN モードでは , この機能を使用して自身の送信を読み取り , シングルワイヤ LIN バスにおいて物理的障害が発生した場合のエラー処理に使用できます。

＜注意事項＞

- ESCRn:SIOP には以前に書き込まれた値が保持されているため , 予期しないレベルの出力を抑止するために , 出力端子アクセスを許可する前に ESCRn:SIOP に目的の値を書き込んでください。
- 通常の読出し命令で ESCRn:SIOP ビットを読み出すと SINn 端子の状態が読み出されますが , リードモディファイライト命令では SOTn 端子の状態が読み出されます。

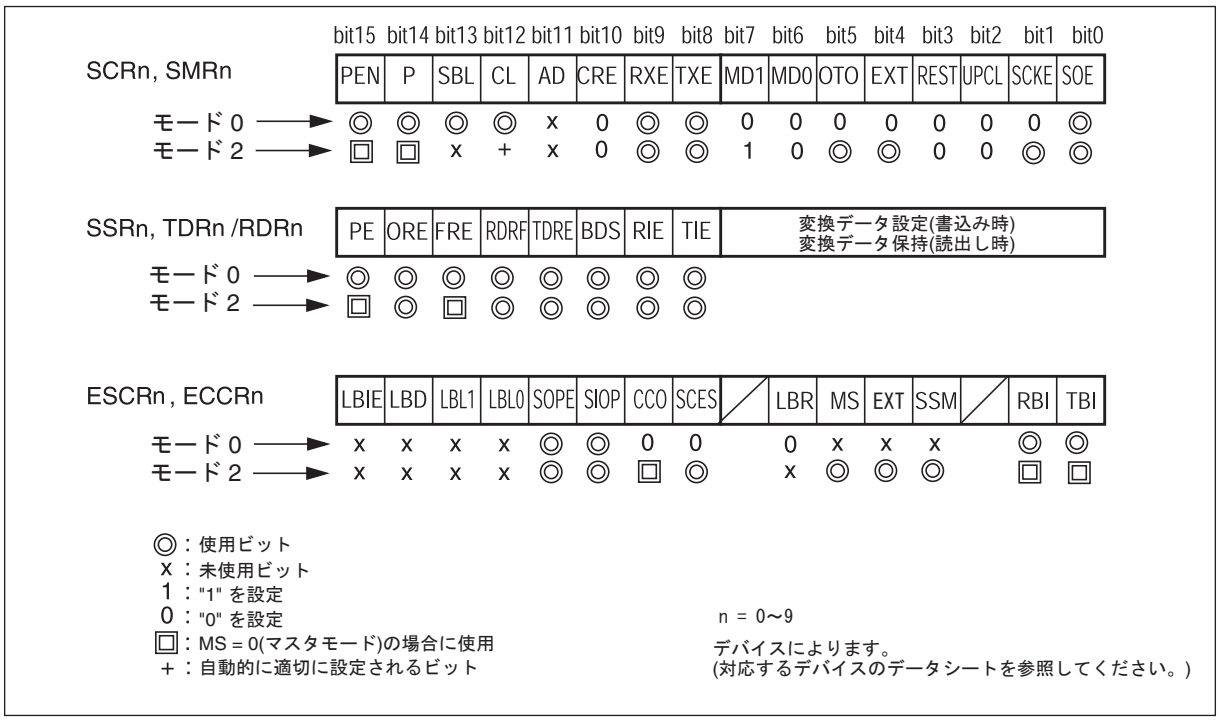
20.7.5 双方向通信機能(ノーマルモード)

動作モード 0 または 2 では, 通常の双方向シリアル通信ができます。 非同期通信ではモード 0, 同期通信ではモード 2 を選択してください。

20.7.5.1 双方向通信機能

USART をノーマルモード (動作モード 0 または 2) で動作させるために必要な設定を Figure 20-25 に示します。

Figure 20-25. USART動作モード0 または2の設定



■ CPU 間接続

Figure 20-26 に示すように , USART モード 2 で 2 つの CPU を相互接続します。

Figure 20-26. USARTモード2双方向通信での接続例

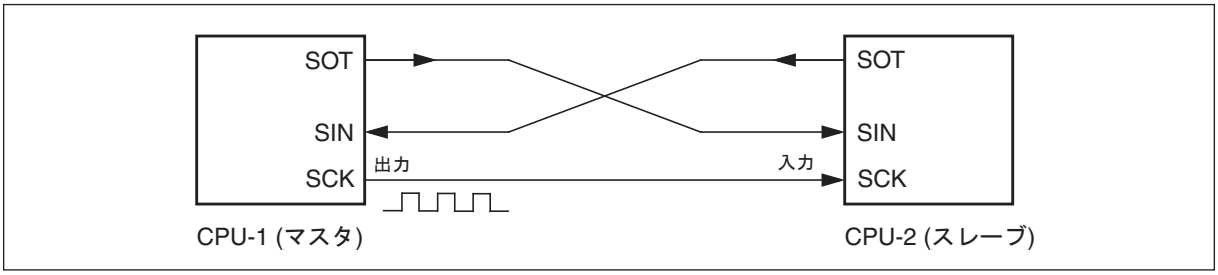
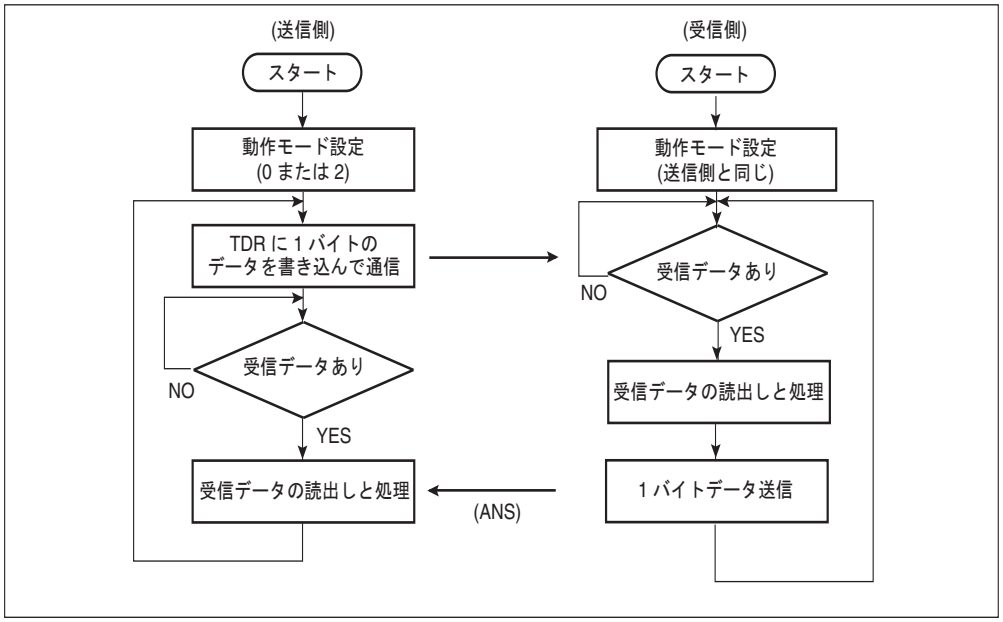


Figure 20-27. マスタ / スレーブ通信のフローチャート例



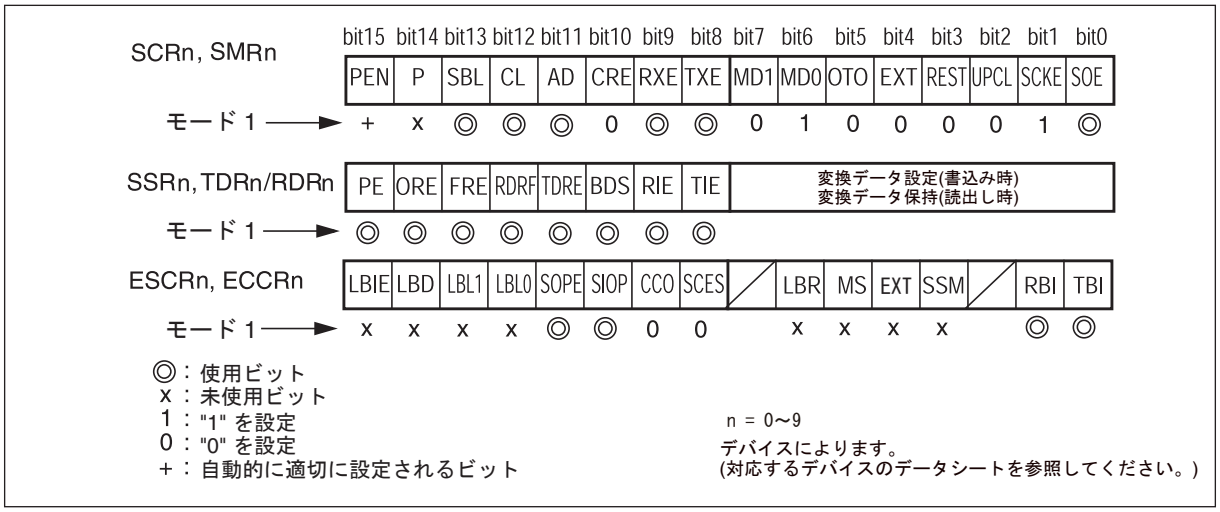
20.7.6 マスタ / スレーブ通信機能(マルチプロセッサモード)

USART は複数の CPU がマスタ/スレーブモードで接続された通信において, マスタシステムまたはスレーブシステムとして使用できます。

20.7.6.1 マスタ / スレーブ通信機能

USART をマルチプロセッサモード (動作モード 1) で動作させるために必要な設定を Figure 20-28 に示します。

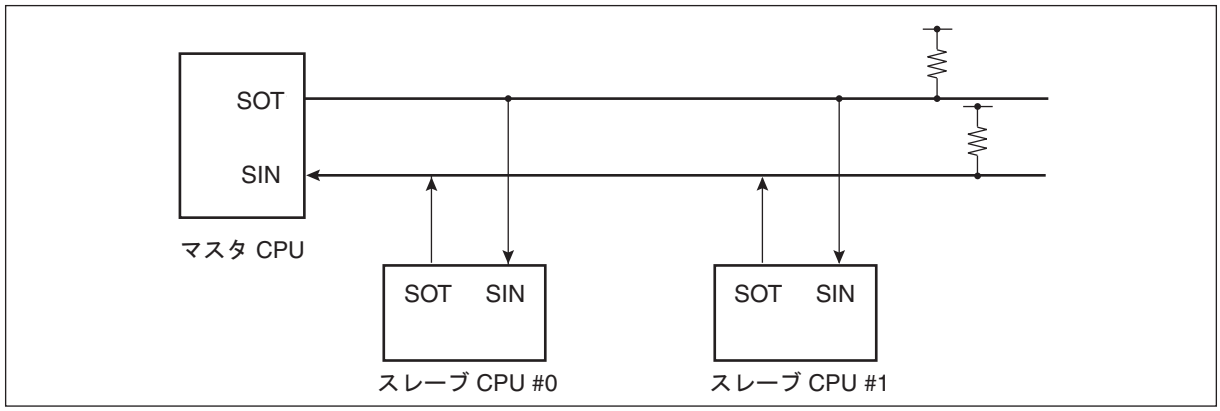
Figure 20-28. USART動作モード1の設定



■ CPU 間接続

Figure 20-29 に示すように , 通信システムは 2 本の通信ラインで接続された 1 つのマスタ CPU と複数のスレーブ CPU からなります。USART はマスタ CPU またはスレーブ CPU として使用できます。

Figure 20-29. USARTマスタ / スレーブ通信の接続例



■ 機能選択

Table 20-16 に示すように, マスタ/スレーブ通信のための動作モードとデータ転送モードを選択します。

Table 20-16. マスタ / スレーブ通信機能の選択

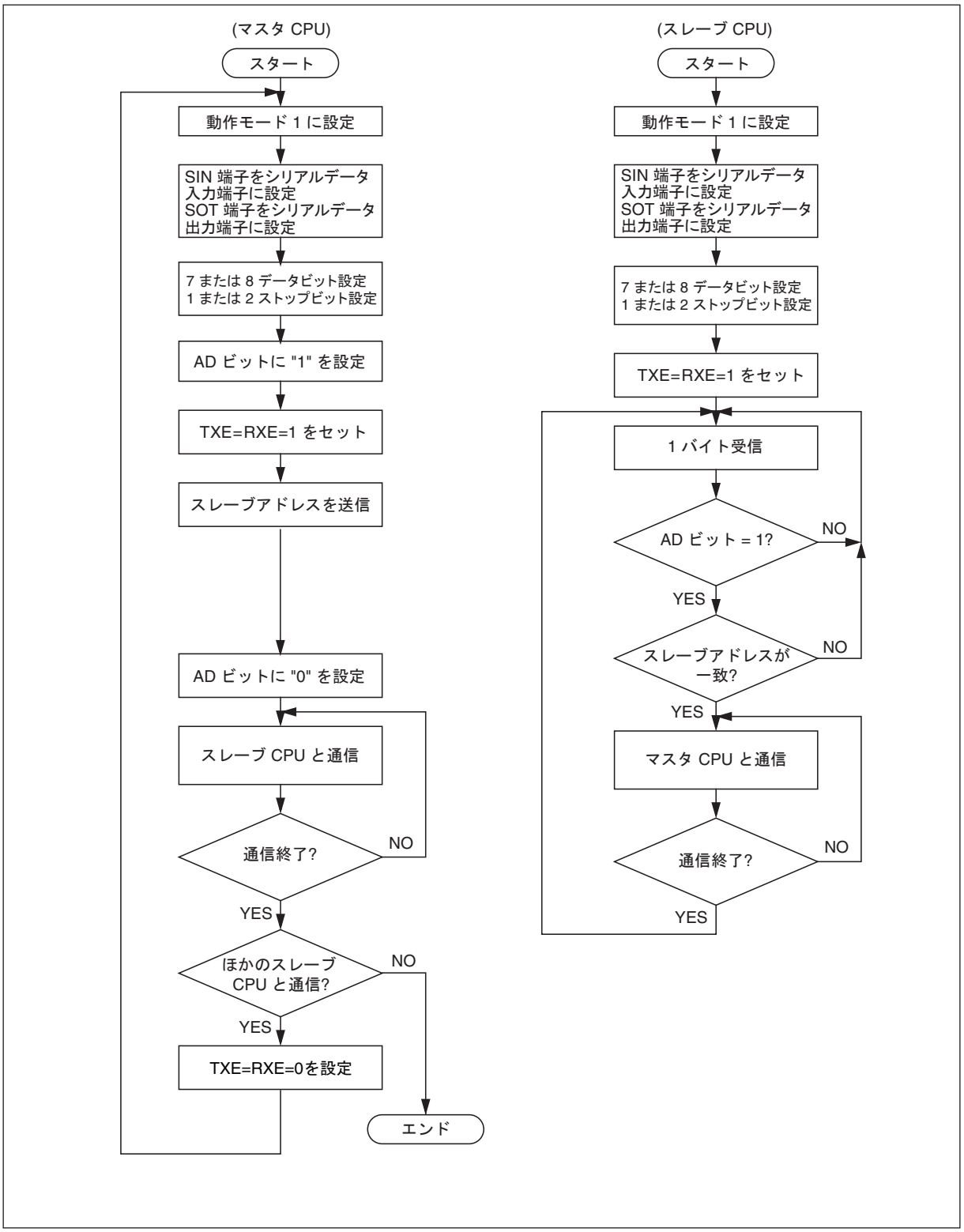
	動作モード		データ	パリティ	同期方式	ストップビット	ビット方向
	マスタCPU	スレーブCPU					
アドレス送受信	モード 1 (AD ビット送受信)	モード 1 (AD ビット送受信)	AD=1+7 ビットまたは 8 ビットアドレス	なし	非同期	1 ビットまたは 2 ビット	LSB ファーストまたは MSB ファースト
データ送受信			AD=0+7 ビットまたは 8 ビットデータ				

通信手順

マスタ CPU がアドレスデータを送信すると , 通信が開始されます。アドレスデータの A/D ビットは 1 にセットされており , 宛先スレーブ CPU が選択されます。各スレーブ CPU はプログラムでアドレスデータをチェックします。アドレスデータがスレーブ CPU に割り当てられているアドレスを示している場合には , そのスレーブ CPU はマスタ CPU と通信します。

Figure 20-30 に , マスタ / スレーブ通信のフローチャートを示します (マルチプロセッサモード)。

Figure 20-30. マスタ / スレーブ通信のフローチャート



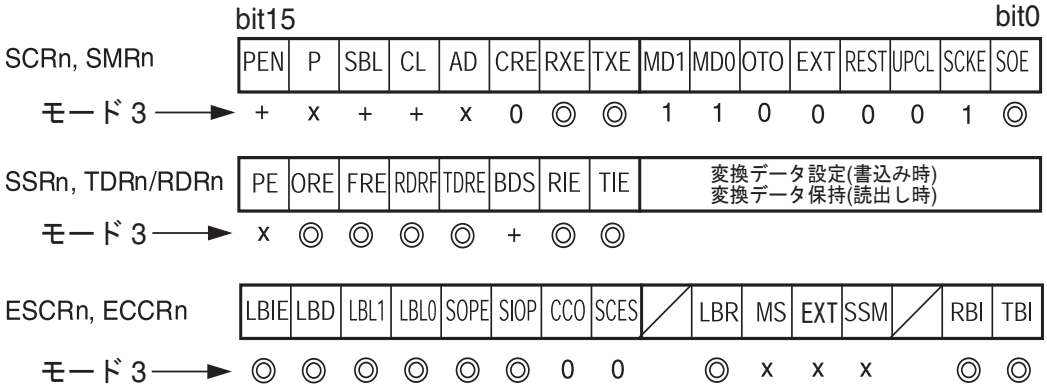
20.7.7 LIN通信機能

USART は LIN デバイスとの通信では LIN マスタシステムまたは LIN スレーブシステムとして使用できます。

20.7.7.1 LIN マスタ / スレーブ通信機能

USART を LIN 通信モード (動作モード 3) で動作させるために必要な設定を下図に示します。

Figure 20-31. USART動作モード3(LIN)の設定



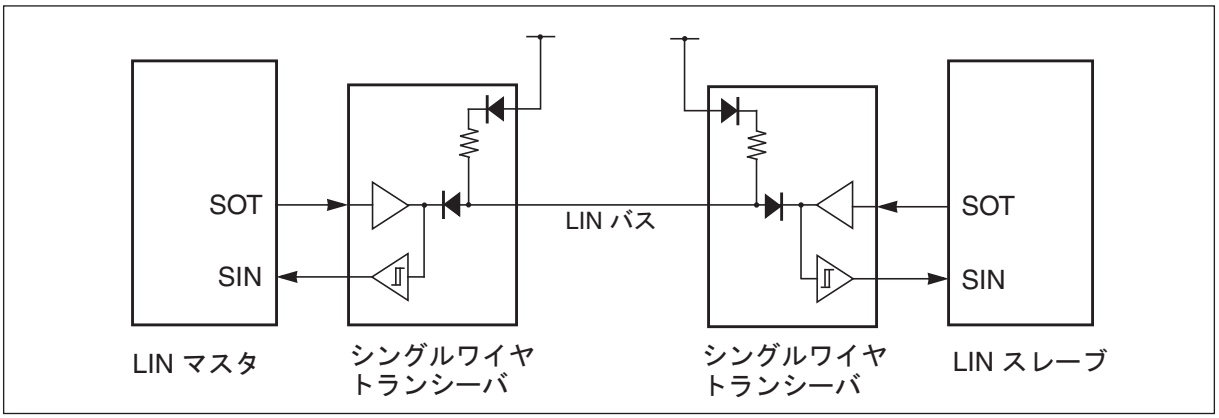
◎ : 使用ビット
x : 未使用ビット
1 : "1" を設定
0 : "0" を設定
+ : 自動的に適切に設定されるビット

n = 0～9
デバイスによります。
(対応するデバイスのデータシートを参照してください。)

■ LIN デバイス接続

1 つの LIN マスタデバイスと 1 つの LIN スレーブデバイスからなる通信システムを下図に示します。USART は LIN マスタまたは LIN スレーブとして動作できます。

Figure 20-32. 小さなLINバスシステムの接続例

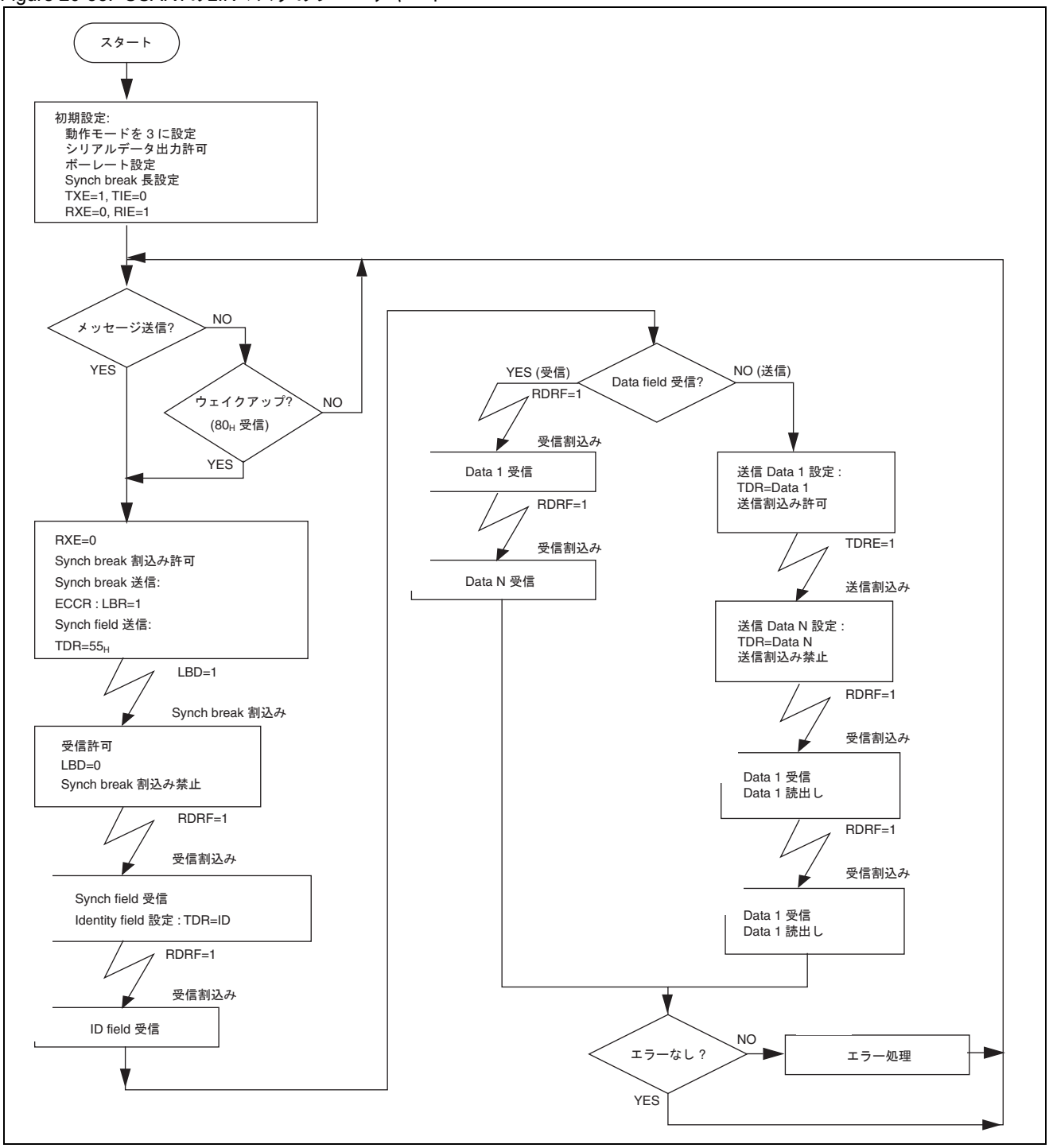


20.7.8 LIN通信でのUSART(動作モード3)のフローチャート例

LIN 通信での USART のためのフローチャート例を示します。

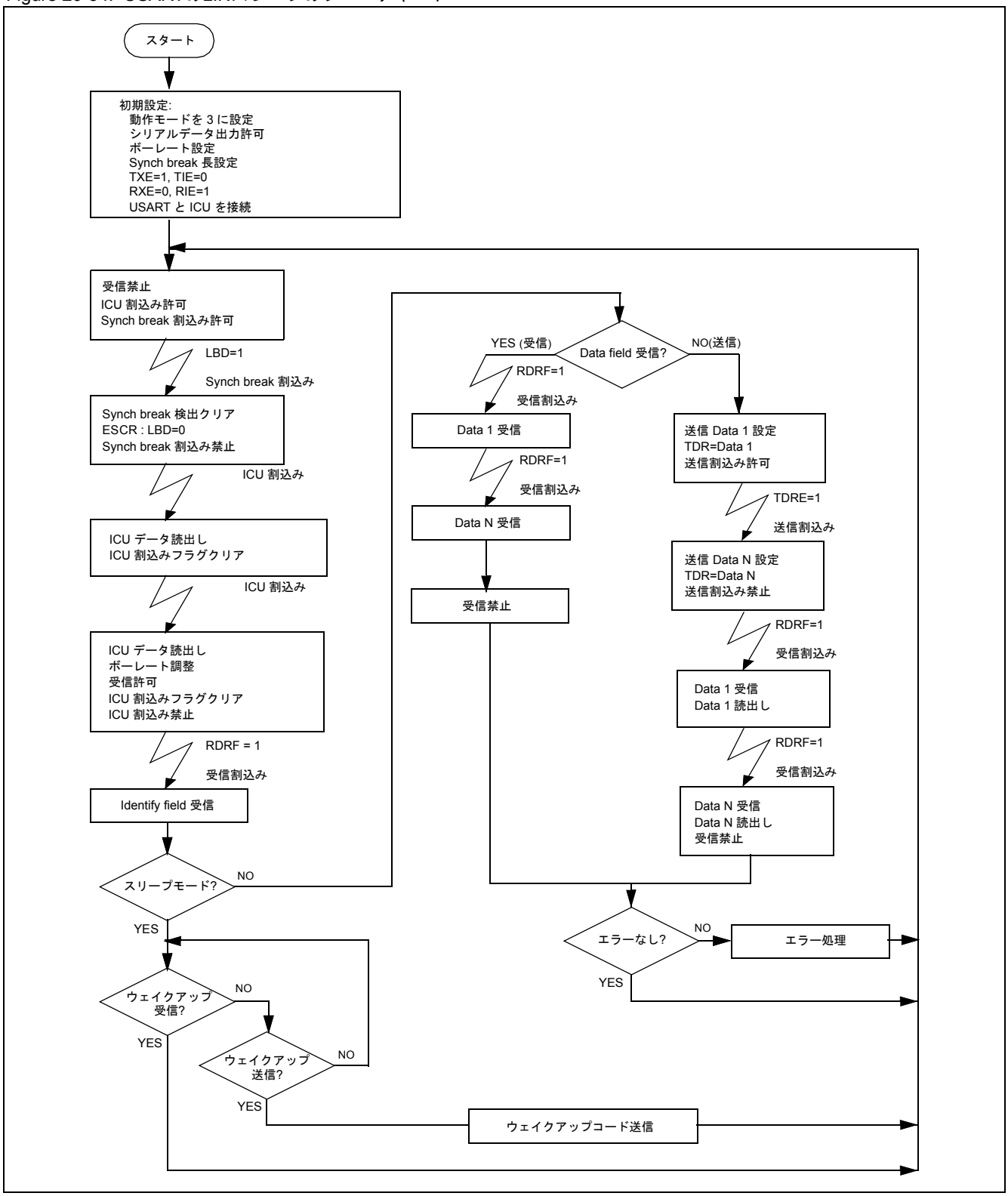
20.7.8.1 マスタデバイス動作

Figure 20-33. USARTのLINマスタのフローチャート



20.7.8.2 スレーブデバイス動作

Figure 20-34. USARTのLINスレーブのフローチャート



20.8 USARTの使用上の注意

USART を使用する場合の注意点を示します。

20.8.1 USARTの使用上の注意

■ 動作の許可

USART の制御レジスタ (SCRn) には動作許可ビットの TXE(送信) と RXE(受信) があります。送信動作と受信動作はデフォルト値 (初期値) では禁止されているため , 通信開始前に許可する必要があります。これらのビットを禁止することで動作を停止することもできます。

■ 通信モード設定

通信モードはシステムが動作していない時に設定してください。送受信中にモードを変更すると送受信は停止し , データを喪失する可能性があります。

■ 送信割込みの許可タイミング

送信データエンプティフラグビット (SSRn:TDRE) のデフォルト (初期値) は "1"(送信データなし , 送信データ書込み可能状態) です。送信割込み要求が許可されると (SSRn:TIE=1), 即座に送信割込み要求が生成されます。即座の割込みを回避するには , 送信データを書き込んだ後に TIE フラグを "1" に設定してください。

■ LIN 動作モード 3 の使用

LIN 機能はモード 3 で使用可能ですが (LIN synch break の送受信), モード 3 では USART のデータフォーマットは自動的に LIN フォーマット (8N1, LSB ファースト) になります。なお , 送信する LIN synch break の長さは可変ですが , 受信は 11 ビット長に固定です。

■ 動作設定の変更

動作設定を変更した後には USART をリセットすることを強く推奨します。特にデータフォーマットへのスタートビットまたはストップビットの追加や削除を行った場合はリセットしてください。

USART の設定またはモードを変更する場合, または USART を初期化する場合は, 通信を禁止する (RXE =0, TXE =0) ことを推奨します。

＜注意事項＞

シリアルモードレジスタ (SMRn) の設定と同時に , USART をリセットするために UPCL ビットをセットしないでください。この場合, 正しい動作設定は保証されません。SMRnのビットを設定した後に, 再度 UPCL ビットを加えた値をセットしてください。

■ LIN スレーブ設定

スレーブ動作のための最初の LIN synch break を受信する前にボーレートを設定してください。さもなければ , LIN synch break の長さが LIN 仕様の最小要件 (マスタは 13 ビット時間 , スレーブは 11 ビット時間) を満たすかどうかを正しく確認できません。

■ バスアイドル機能

バスアイドル機能は , モード 2 の同期スレーブモードでは使用できません。

■ AD ビット(シリアル制御レジスタ(SCRn): アドレス/データ形式選択ビット)

シリアル制御レジスタの AD ビット (マルチプロセッサモード 1 用のアドレス / データビット) を使用する際には特別な注意が必要です。このビットは , 書き込むと送信用の AD ビット設定となり , 読み出すと最後に受信した AD ビットを返すため , 制御ビットとフラグビットの両方です。内部的には送信値と受信値は異なるレジスタに格納されますが , リードモディファイライト命令では受信値が読み出され , モディファイの後に送信値に書き戻されます。このため , リードモディファイライト命令で同一レジスタ内の他のビットがアクセスされた場合には , AD ビットに不正な値が書き込まれる場合があります。

■ 受信エラーのクリア

受信エラーをクリアすると , 受信ステートマシンがリセットされます。そのため , 進行中の受信を擾乱しないためには , 次のスタートビットまたはスタート条件成立前にすべての受信エラーを確認する必要があります。

■ LIN synch field のウェイトステート

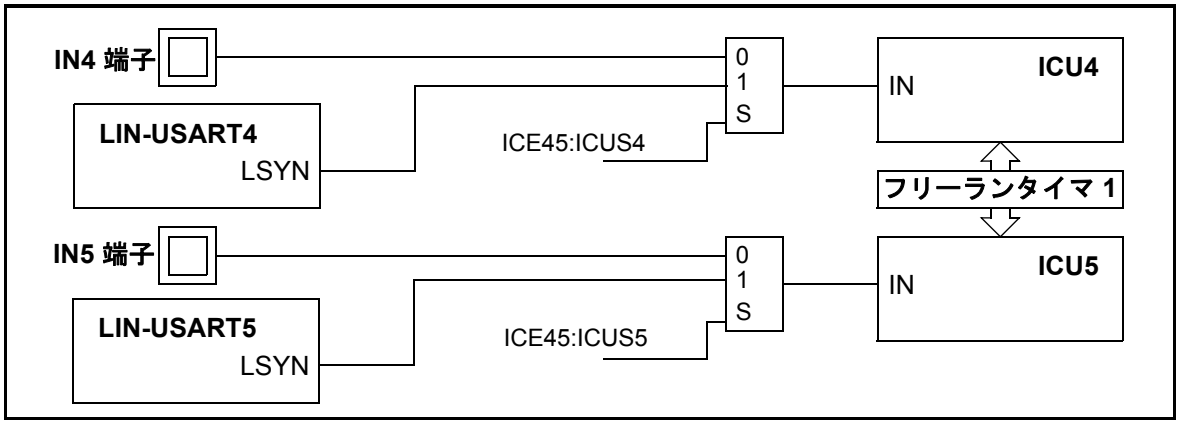
モード 3(LIN 動作) では , 入力信号が 11 ビット時間以上 "0" を維持すると , ESCRn レジスタの LBD ビットが "1" にセットされます。すると , USART は次にくる LIN synch field の受信を待機します。USART が LIN synch break 以外の理由でこの状態になった場合には , ソフトウェアリセット (SMRn, SCRn:UPCL=1) で初期化する必要があります。

モード 3 では , LIN break 検出は常にバックグラウンドで動作しています。バスエラーの場合には注意してください (バスの優先順位は高い)。LIN break 検出フラグ (LBD) は ,
LIN break 検出割込みが許可されているかどうかにかかわらず , 11 ビット時間の "0" の後には "1" になるかまたは "1" のままです。LIN break 検出割込みを使用する場合は, 受信割込みハンドラ内で必ずこのフラグを確認してクリアしてください。

■ インプットキャプチャユニット (ICU)を使用したボーレート検出

USART は ICU に接続可能な信号 LSYN を提供しますが, これはボーレートを引き出すために LSYN のパルス幅を測定できるようにするためです。LSYN 信号の ICU への接続は , インプットキャプチャエッジレジスタ ICE(2n)(2n+1) によって制御されます。ICU および USART の割り当てについては , 「1.6 "LIN-USART のインプットキャプチャユニットのソースの選択 "」を確認してください。

Figure 20-35. インプットキャプチャユニットを使用したボーレート検出



ICE(2n)(2n+1) レジスタの ICUSx ビットがクリアされ , 周辺リソース入力が許可されると , ICU はその対応する入力端子 IN に接続されます。ICUSx ビットが "1" に設定されると , 対応する LIN-USART が ICU に接続されます。

ユーザは , 「異なる複数の ICU が 1 つのフリーランタイマ (プリスケーラ) を共有する」ということに考慮する必要があります。

受信エラーの影響と CRE ビット

CRE は受信ステートマシンをリセットし , SINn の次の立下りエッジで新しいバイトの受信が開始します。したがって , データストリームの非同期化を防止するため , エラーを受信した直後 (1/2 ビット時間内) に CRE ビットを設定するか, あるいはエラー受信後にアプリケーションの依存時間を待ってから, SINn がアイドル状態のときに CRE ビットを設定してください。

Figure 20-36. CREビットのタイミング

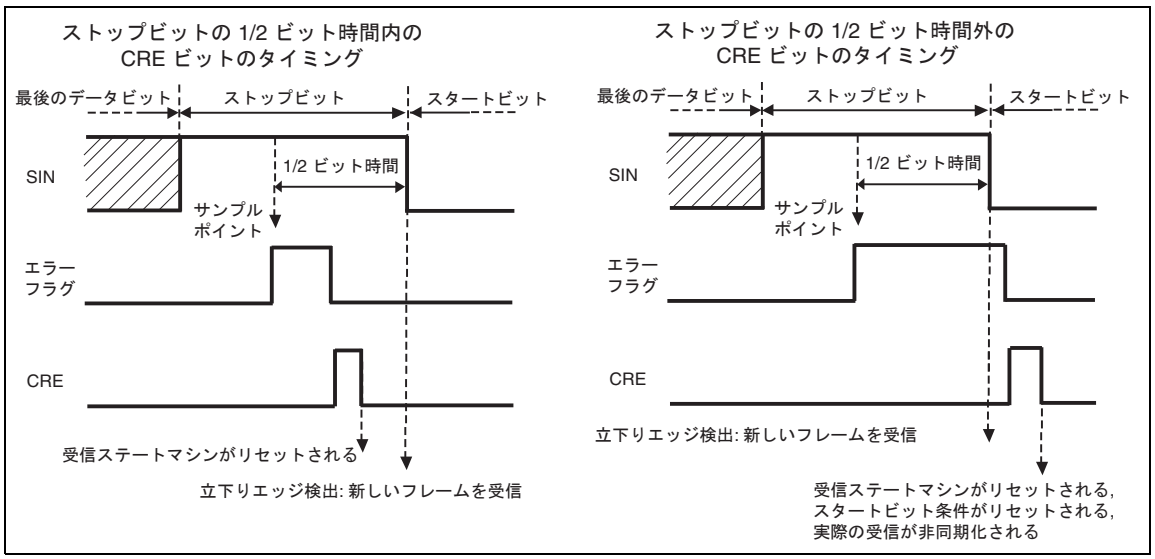
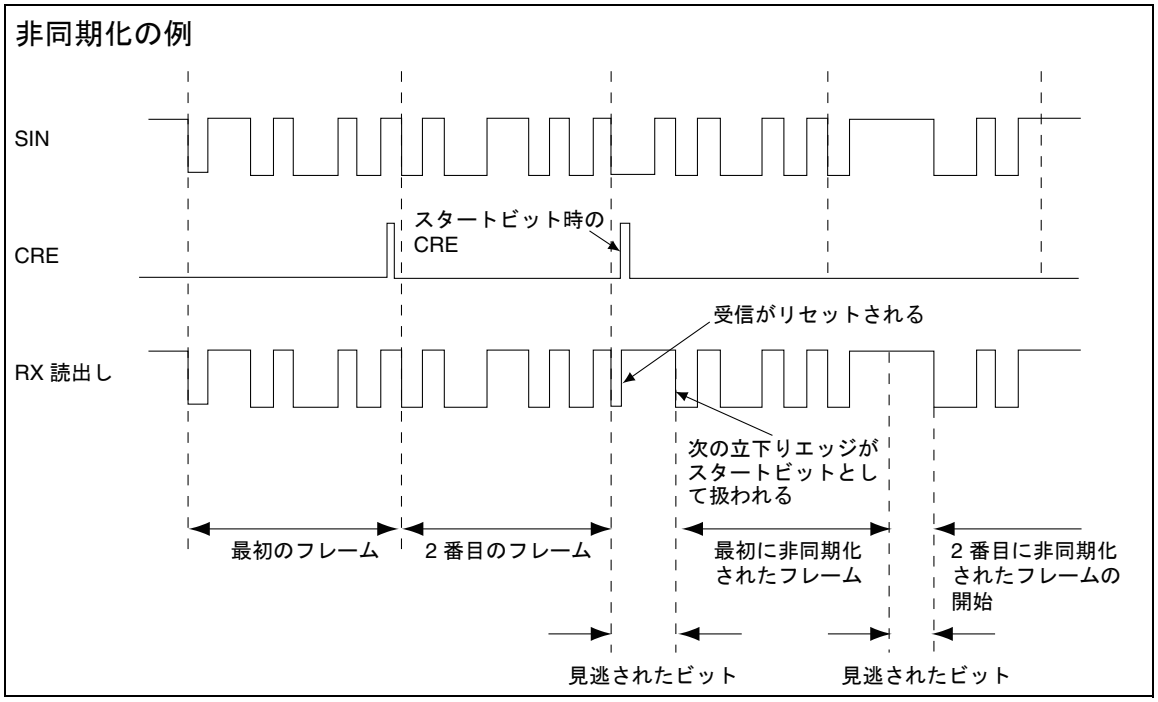


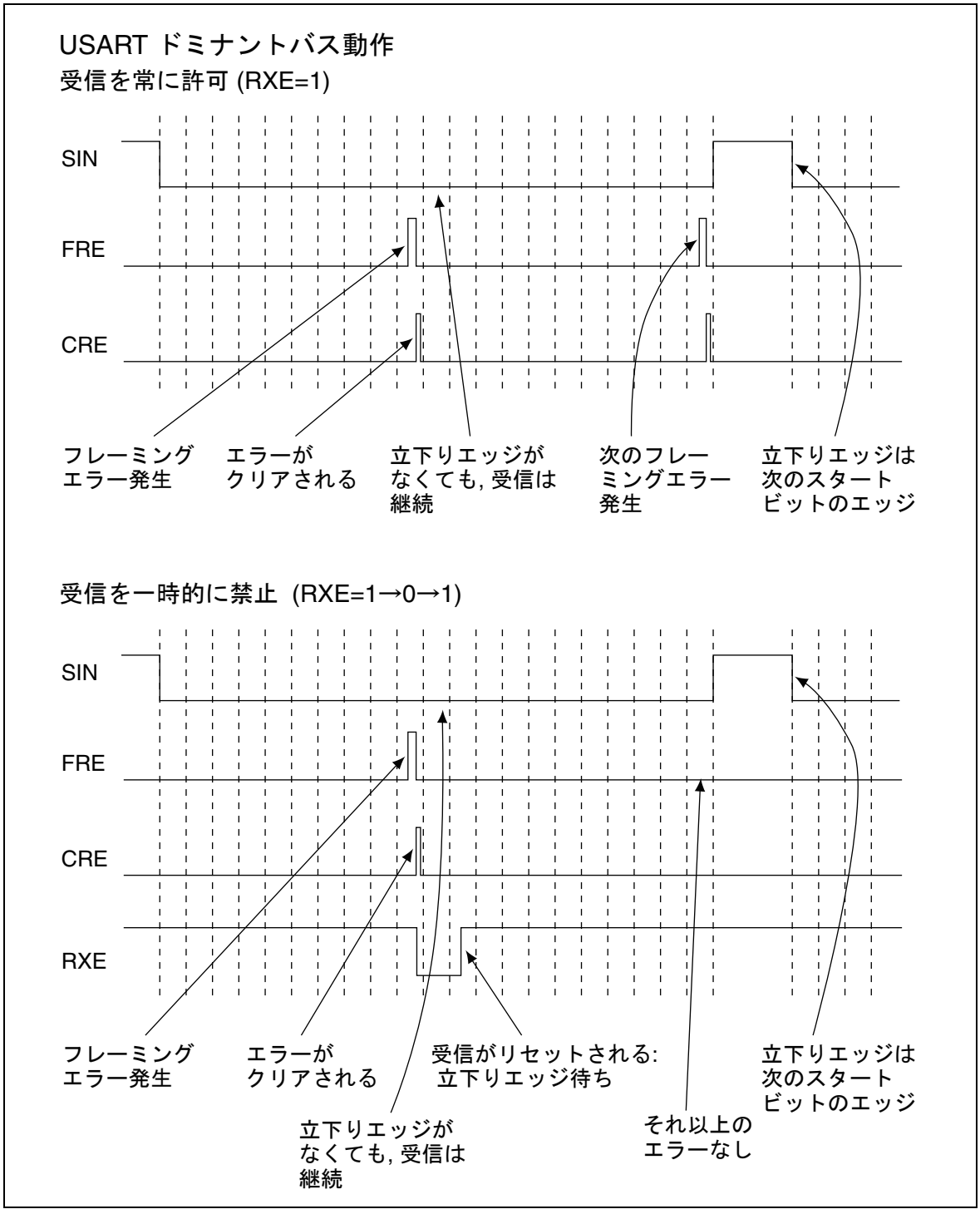
Figure 20-37. データストリームの同期化



＜注意事項＞

フレーミングエラーが発生 (ストップビット : SINn = "0") した直後に , 次のスタートビット (SINn = "0") が続く場合 , 事前に立下りエッジがなかったとしても , このスタートビットは認識されます。これは , リセッシブなストップビットが期待されている場合に , 次のフレーミングエラーを生成することで , USART がデータストリームに同期化し続け , バスの優先順位が常に高いエラーを判断するために使用します (Figure 20-38「USART ドミナントバス動作」の上図を参照)。この動作を希望しない場合は , フレーミングエラー後に受信を一時的に禁止してください (RXE = 1 -> 0 -> 1)。その場合 , 受信は SINn の次の立下りエッジまで継続します (Figure 20-38「USART ドミナントバス動作」の下図を参照)。

Figure 20-38. USARTドミナントバス動作



21. 400 kHz I²Cインタフェース

高速I²Cインタフェースの機能と動作について説明します。

21.1 I²Cインタフェースの概要

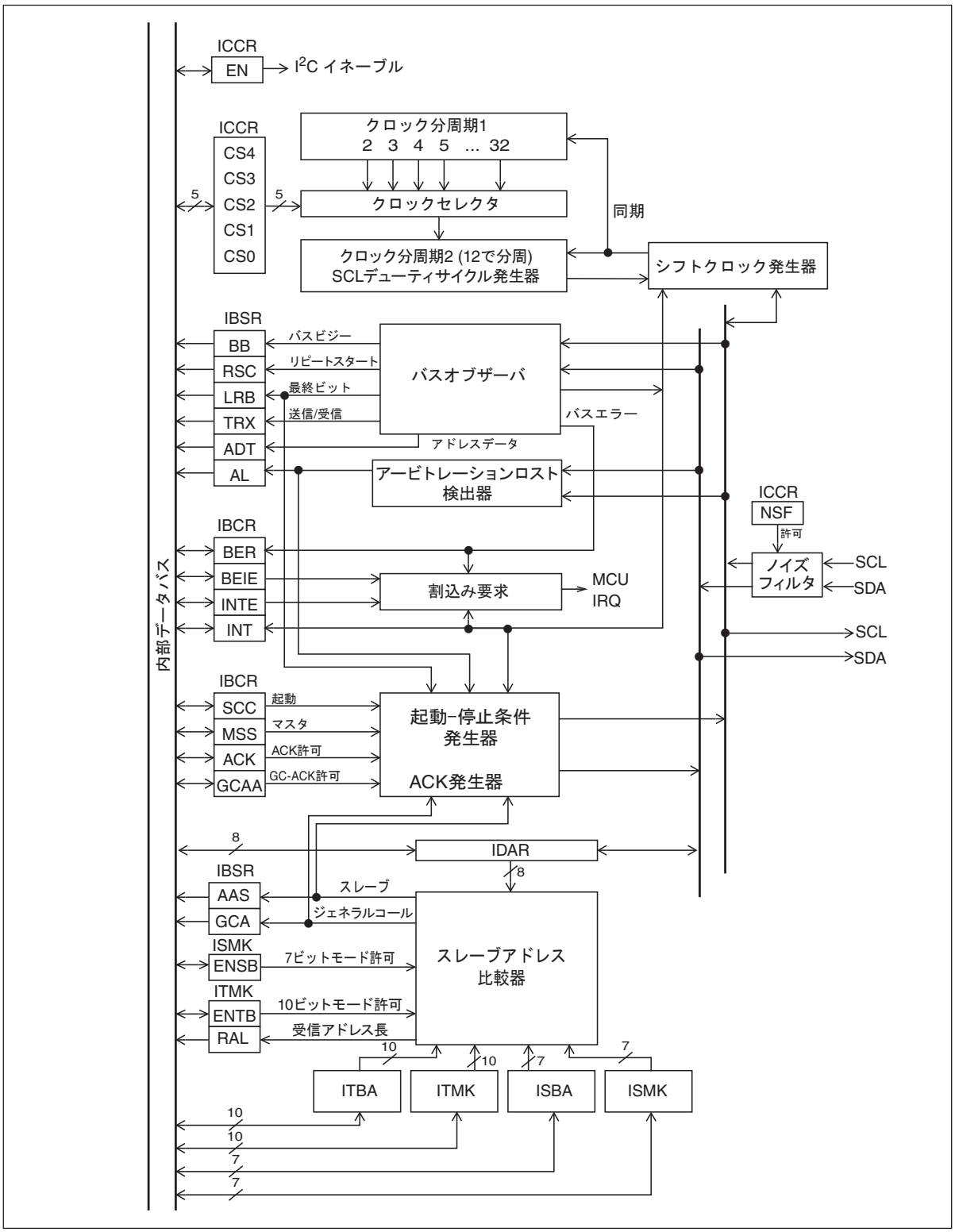
このI²CインタフェースはInter ICバスに対応したシリアルI/Oポートで, I²C上のマスタ/スレーブデバイスとして動作します。

特長

- マスタ/スレーブ送受信機能
- アービトレーション機能
- クロック同期機能
- ジェネラルコールアドレッシングをサポート
- 伝送方向検出機能
- リピートスタート条件の発生と検出機能
- バスエラー検出機能
- 7ビットアドレッシング(マスタおよびスレーブ)
- 10ビットアドレッシング(マスタおよびスレーブ)
- 7ビットおよび10ビットスレーブアドレスのインタフェースが可能
- スレーブアドレス受信時のアクノリッジを禁止できる(マスタのみの動作)
- 複数のスレーブアドレスをインタフェースするためのアドレスマスキング(7ビットおよび10ビットモード)
- 最大400 Kビットの転送速度
- SDAおよびSCL用の内蔵型ノイズフィルタとして使用可能
- 周辺クロック(CLKP1)が6 MHz以上のときは, プリスケーラの設定に関係なく 400 Kビットでのデータ受信が可能
- 送信およびバスエラー発生時のMCU割込み発生可能
- ビットレベルとバイトレベルでのスレーブによる減速をサポート

このI²Cインタフェースは, 周辺クロック CLKP1 が 6 MHz 以上のときは, プリスケーラの設定に関係なく 400 K ビットでのデータ転送速度で受信可能なため , ビットレベルでの SCL クロック引伸ばしに対応しません。ただし,割込み中(IBCR レジスタのINT=1) はSCLが"L" になるので, バイトレベルでのクロック引延しが行われます。

Figure 21-1. 400kHz I²Cインタフェースのブロックダイヤグラム

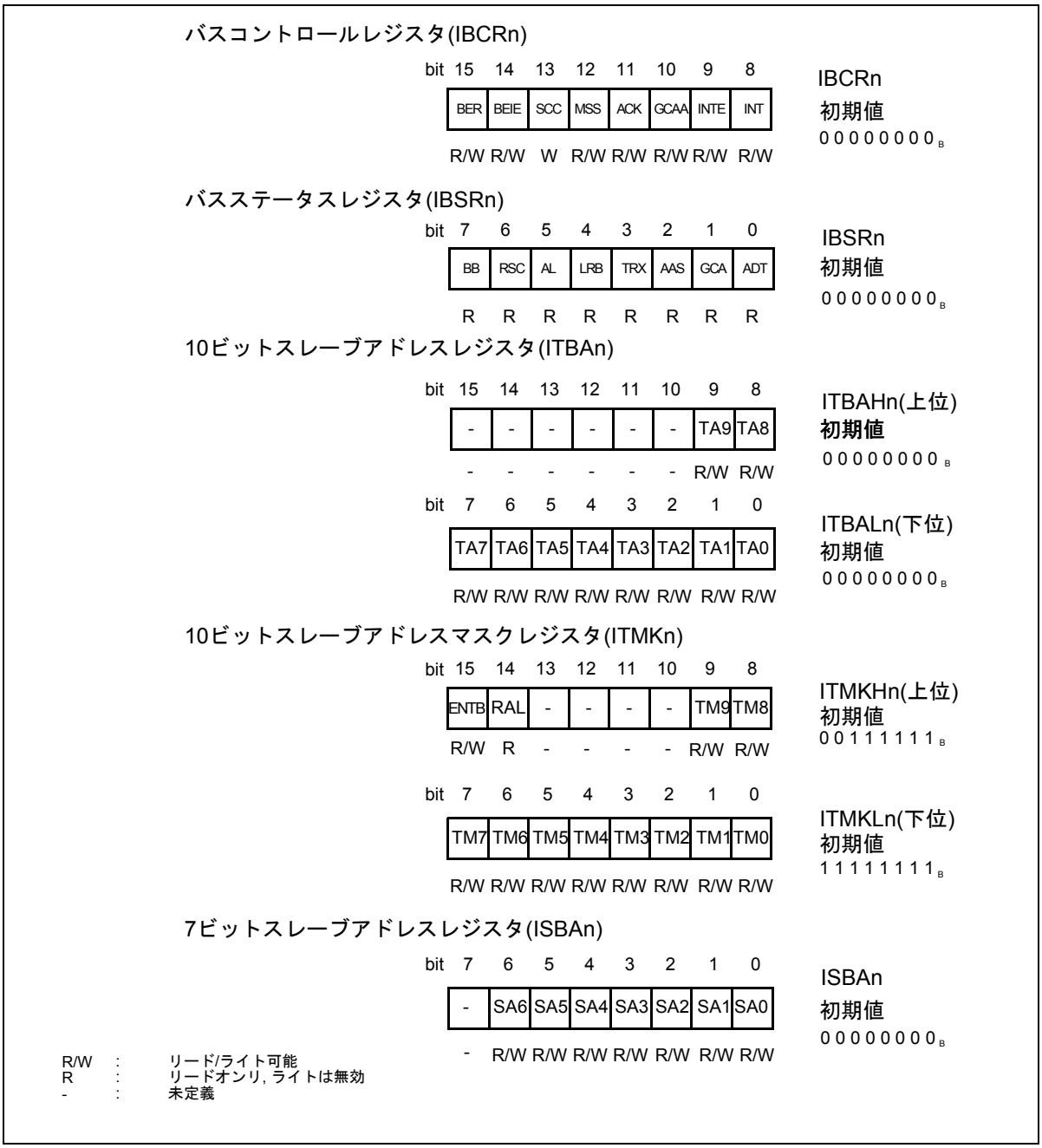


21.2 I²Cインタフェースのレジスタ

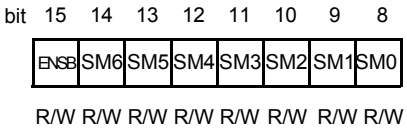
I²Cインタフェースの各レジスタについて詳細に説明します。

I²Cインタフェースのレジスタ

Figure 21-2. I²Cインタフェースのレジスタ

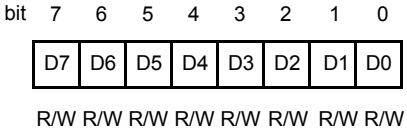


7ビットスレーブアドレスマスクレジスタ(ISMKn)



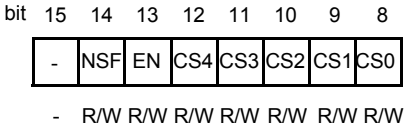
ISMKn
初期値
01111111_B

データレジスタ(IDARn)



IDARn
初期値
00000000_B

クロックコントロールレジスタ(ICCRn)



ICCRn
初期値
00011111_B

R/W : リード/ライト可能
- : 未定義

21.2.1 バスステータスレジスタ(IBSRn)

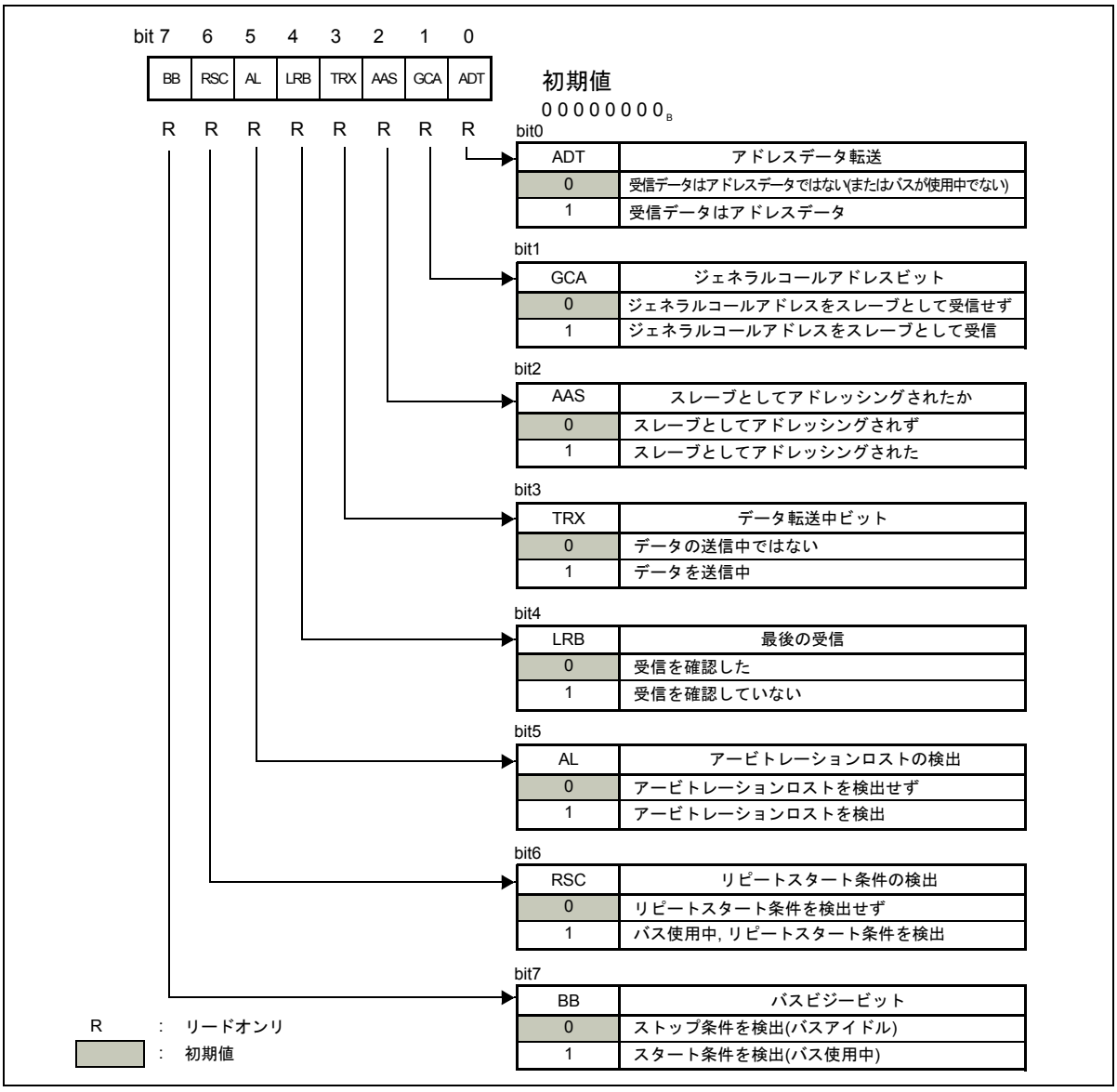
バスステータスレジスタ (IBSRn) には次の機能があります。

- バスビジーの検出
- リピートスタート条件の検出
- アービトレーションロストの検出
- アクノリッジの検出
- データ転送方向の検出
- スレーブとしてのアドレッシングの検出
- ジェネラルコールアドレスの検出
- アドレスデータ転送の検出

バスステータスレジスタ(IBSRn)

このレジスタはリードオンリで , すべてのビットはハードウェアによって制御されます。インタフェースを禁止すると (ICCRn の EN =0), すべてのビットがクリアされます。

Figure 21-3. バスステータスレジスタ(IBSRn)の構成



バスステータスレジスタ(IBSRn)の内容

Table 21-1. バスステータスレジスタ(IBSRn)の各ビットの機能 (Sheet 1 of 2)

ビット名		機能
bit 7	BB: バスビジービット	I^2C バスの状態を示します。 "0": ストップ条件を検出 (バスアイドル) "1": スタート条件を検出 (バス使用中) スタート条件が検出されると, このビットが"1"に設定されます。ストップ条件が検出されるとリセットされます。
bit 6	RSC: リピートスタート条件の検出	リピートスタート条件が検出されたかどうかを示します。 "0": リピートスタート条件検出されず。 "1": リピートスタート条件を検出 (バス使用中)。 このビットは, アドレスデータの転送後(ADT=0)またはストップ条件が検出されるとクリアされます。

Table 21-1. バスステータスレジスタ(IBSRn)の各ビットの機能 (Sheet 2 of 2)

ビット名		機能
bit 5	AL: アービトレーションロストの検出	アービトレーションロストを示します。 "0": アービトレーションロスト検出されず。 "1": マスタが送信している間にアービトレーションロストが発生。 このビットは , IBCR レジスタの INT ビットに "0" を書き込むか , MSS ビットに "1" を書き込むとクリアされます。 次の場合にはアービトレーションロストが発生します。 • 送信データが , SCL 端子の立上りエッジで読み出した SDA ラインのデータと一致しない場合 • データバイトの最初のビットで , 別のマスタによるリピートスタート条件が発生した場合 事前に別のスレーブによりSCL 端子ラインが"1"から"0" となったために,インタフェースが起動条件,または停止条件を発生できなかった場合
bit 4	LRB: 最後の受信検出	受信側からのアクノリッジを示すために使用されます。 "0": 受信を確認した "1": 受信を確認しなかった ハードウェアがbit9 (アクノリッジビット) を受信すると,このビットは変更され,スタート条件,またはストップ条件によりクリアされます。
bit 3	TRX: 送信動作表示ビット	このビットはデータ送信動作を示します。 "0": データの送信中ではない。 "1": データを送信中。 次の場合に "1" にセットされます。 - マスタモードでスタート条件が発生した場合 - 最初のバイト転送の終了 , およびスレーブとしての読出しアクセス , またはマスタとしてのデータ送信の場合 次の場合に "0" にセットされます。 - バスがアイドル状態になった (BB=0)。 - アービトレーションロストが発生した。 - マスタ割込み中 (MSS=1 および INT=1) に SCC ビットに "1" が書き込まれた。 - マスタ割込み中 (MSS=1 および INT=1) に MSS ビットがクリアされた。 - インタフェースがスレーブモードで , 最後に転送されたバイトが確認されなかった。 - インタフェースがスレーブモードでデータを受信中。 - インタフェースがマスタモードでスレーブからのデータを読出し中。
bit 2	AAS: スレーブとしてアドレッシング検出	スレーブアドレッシングが検出されたことを示します。 "0": スレーブとしてアドレッシングされず。 "1": スレーブとしてアドレッシングされた。 このビットは, (リピート)スタート/ストップ条件が検出されるとクリアされます。また, インタフェースが7 ビットスレーブアドレスと10 ビットスレーブアドレスの両方, またはいずれかを検出するとセットされます。
bit 1	GCA: ジェネラルコール検出	ジェネラルコールアドレス (00_H) が検出されたことを示します。 "0": スレーブとしてジェネラルコールアドレスを検出せず。 "1": スレーブとしてジェネラルコールアドレスを検出。 このビットは, (リピート)スタート/ストップ条件が検出されるとクリアされます。
bit 0	ADT: アドレスデータ転送検出	アドレスデータの転送が検出されたことを示します。 "0": 受信データはアドレスデータではない (またはバスが使用されていない)。 "1": 受信データはアドレスデータである。 このビットは , スタート条件が検出されると "1" にセットされます。また , ライトアクセスの 10 ビットスレーブアドレスヘッダが検出された場合は第 2 バイトの後にクリアされます。それ以外の場合は第 1 バイトの後にクリアされます。 「第 1/ 第 2 バイトの後」が意味するものは , 以下のとおりです。 - マスタ割込み中 (IBCR の MSS=1 および INT=1) に MSS ビットに "0" が書き込まれた。 - マスタ割込み中 (IBCR の MSS=1 および INT=1) に SCC ビットに "1" が書き込まれた。 - INT ビットがクリアされた。 - インタフェースがマスタ,またはスレーブとして現在の転送にかかわっていない場合は,すべてのバイト転送が開始されます。

21.2.2 バスコントロールレジスタ(IBCRn)

バスコントロールレジスタ (IBCRn) には次の機能があります。

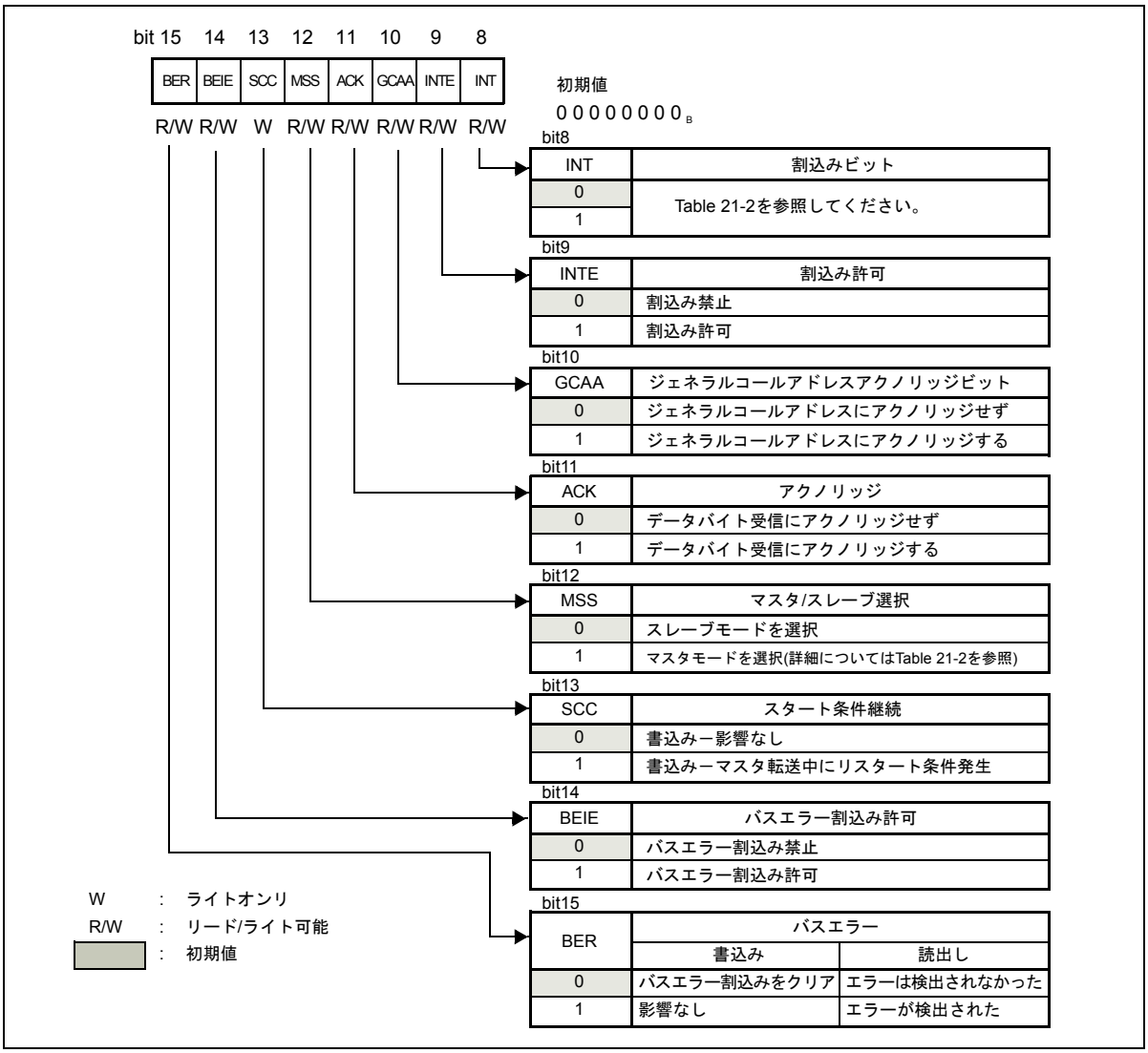
- 割込み許可フラグ
- 割込み発生フラグ
- バスエラー検出フラグ
- リピートスタート条件の生成
- マスタ/スレーブモードの選択
- ジェネラルコールアクノリッジ発生許可
- データバイトアクノリッジ発生許可

21.2.2.1 バスコントロールレジスタ(IBCRn)

このレジスタへの書込みアクセスは , INT=1, または転送が開始される場合にのみ発生します。ACK ビット , または GCAA ビットを変更するとバスエラーが発生することがあるので , ユーザによるこのレジスタへの転送中の書込みは禁止されます。

インタフェースが許可されない場合 (ICCR の EN=0) は , BER ビットと BEIE ビットを除き , このレジスタのすべてのビットがクリアされます。

Figure 21-4. バスコントロールレジスタ(IBCRn)の構成



21.2.2.2 バスコントロールレジスタ(IBCRn)の内容

Table 21-2. バスコントロールレジスタ(IBCRn)の各ビットの機能 (Sheet 1 of 2)

ビット名		機能
bit 15	BER: バスエラー	バスエラー割込みフラグです。ハードウェアによってセットされユーザがクリアします。 リードモディファイライトアクセスでは常に "1" が読み出されます。 ライトアクセスの場合 : "0": バスエラー割込みフラグのクリア "1": 影響なし リードアクセスの場合 : "0": バスエラーは検出されなかった。 "1": 下記のいずれかのエラー状態が検出された このビットを設定すると , ICCR レジスタの EN ビットがクリアされます。I^2C インタフェースが一時停止状態となってデータ転送が中断され , IBSR レジスタと IBCR レジスタの BER と BEIE を除くすべてのビットがクリアされます。インタフェースを再度許可する前に , BER ビットをクリアする必要があります。 • このビットは , 以下の場合に "1" に設定されます。 - アドレスデータの転送中 , または bit2 ～ bit9 (アクノリッジビット) の転送中に , スタート条件 , またはストップ条件が誤った位置で検出された - 10 ビットの書込みアクセス以前に,読出しアクセスを伴う10ビットアドレスヘッダを受信した 転送中にインタフェースが許可された場合は,誤ったバスエラー報告を防ぐために,最初の停止条件を受信した後で, 上記条件の最初の2 つを検出することが許可されます。
bit 14	BEIE: バスエラー 割込み許可	バスエラー割込みを許可します。変更できるのはユーザだけです。 "0": バスエラー割込み禁止 "1": バスエラー割込み許可 本ビットを"1"にセットした場合は, BERビットが"1"にセットされるとMCUによる割込み発生が許可されます。
bit 13	SCC: スタート 条件継続	リピートスタート条件の生成に使用されます。このビットは書込み専用で , 常に "0" が読み出されます。 ライトアクセスの場合 : "0": 影響なし "1": マスタ転送時にリピートスタート条件を発生する。 マスタモード(MSS=1, INT=1) での割込み中にこのビットに"1" が書き込まれるとリピートスタート条件が生成され, INTビットは自動的にクリアされます。
bit 12	MSS: マスタ/ スレーブ選択	マスタ / スレーブモードを選択するビットです。セットできるのはユーザだけですが , ユーザとハードウェアの両方からクリアすることができます。 "0": スレーブモードに設定する。 "1": マスタモードに設定し , スタート条件を生成して IDAR レジスタ内のアドレスデータバイトを送信する。 マスタの送信中にアービトレーションロストイベントが発生するとクリアされます。 マスタ割込み中 (MSS=1 および INT=1) に本ビットに "0" を書き込むと , INT ビットが自動的にクリアされてストップ条件が生成され , データ転送は終了します。MSS ビットを直ちにリセットし , IBSR レジスタの BB ビットをポーリングすることによりストップ条件が生成されたかどうかをチェックすることができます。 バスのアイドル時 (MSS=0 および BB=0) に本ビットに "1" を書き込むと , スタート条件が生成され , IDAR レジスタの内容 (アドレスデータ) が送信されます。 バスが使用されている (IBSR: BB=1 および TRX=0, IBCR: MSS=0) ときに MSS ビットに "1" を書き込むと , インタフェースはバスが開放されるまで待機して , 送信を開始します。 送信中にインタフェースがライトアクセス(データ受信)のスレーブとしてアドレッシングされた場合は, その転送が終了してバスが再度開放されてから送信が開始されます。このときインタフェースがスレーブとしてデータを送信していた場合は(IBSR: AAS=1およびTRX=1), 再度開放されてもデータ送信を開始しません。次の割込み時に, インタフェースがスレーブとしてアドレッシングされたかどうか(IBSR: AAS=1), データバイトが正しく送信されたか(IBCR: MSS=1), データバイトの送信に失敗したか(IBSR: AL=1)をチェックすることが重要です。

Table 21-2. バスコントロールレジスタ(IBCRn)の各ビットの機能 (Sheet 2 of 2)

ビット名		機能
bit 11	ACK: アクノリッジ	データバイト受信時のアクノリッジの生成を許可します。変更できるのはユーザだけです。 "0": データバイト受信時にアクノリッジは発生されません。 "1": データバイト受信時にアクノリッジが発生されます。 本ビットはスレーブモードでのアドレスバイトの受信時には無効です。インタフェースは 7 または 10 ビットのスレーブアドレスを検出すると , 対応する許可ビット (ITMK: ENTB または ISMK: ENSB) がセットされていればアクノリッジを返します。 本ビットへのライトアクセスは, 割込み時(INT=1)またはバスアイドル時(IBSR: BB=0)にのみ発生します。本ビットへのライトアクセスは, インタフェースが許可されていて(ICCR レジスタの EN=1), さらにバスエラーでない場合(IBCR レジスタのBER=0) にのみ可能です。
bit 10	GCAA: ジェネラルコールアドレスアクノリッジビット	ジェネラルコールアドレスを受信したときにアクノリッジの生成を行うことができます。変更できるのはユーザだけです。 "0": ジェネラルコールアドレスを受信したときにアクノリッジの生成を行いません。 "1": ジェネラルコールアドレスを受信したときにアクノリッジの生成を行います。 本ビットへのライトアクセスは, 割込み時(INT=1), バスアイドル時(IBSR: BB=0), またはインタフェースが禁止されているとき(ICCR: EN=0)にのみ発生します。このビットへの書込みアクセスは , インタフェースが許可されていて(ICCR レジスタのEN=1), さらにバスエラーでない場合(IBCRレジスタのBER=0) にのみ可能です。
bit 9	INTE: 割込み許可	MCU での割込み生成を許可します。変更できるのはユーザだけです。 "0": 割込み禁止 "1": 割込み許可 本ビットを"1"にセットすると, (ハードウェアによって)INTビットが"1"にセットされたときにMCUによる割込みが生成されます。
bit 8	INT: 割込みフラグ	転送終了割込み要求フラグです。ハードウェアによってセットされ , ユーザがクリアすることができます。リードモディファイライトアクセスでは常に "1" が読み出されます。 ライトアクセスの場合 : "0": 転送終了割込み要求フラグをクリアする。 "1": 影響なし リードアクセスの場合 : "0": 転送未終了 , 現在の転送に関係がない , バスがアイドル状態のいずれか。 "1": 次の 4 条件下で , アクノリッジビットも含め 1 バイトのデータ送受信後に "1" になります。 デバイスがバスマスタである。 デバイスがスレーブとしてアドレッシングされた。 ジェネラルコールアドレスを受信した。 アービトレーションロストが発生した。 デバイスがスレーブとしてアドレッシングされた場合は , アクノリッジビットを含むアドレスデータ受信終了時 (7 ビットアドレスを受信した場合は最初のバイトの後 , 10 ビットアドレスを受信した場合は 2 番目のバイトの後) にセットされます。 本ビットが"1"の間, SCLラインは"L"レベルに保持されます。本ビットに"0"を書き込むとSCLラインが解放され, 次のバイトの転送が実行されるかリピートスタート/ストップ条件が生成されます。また, SCCビットに"1"を書き込むかMSSビットをクリアすると本ビットはクリアされます。

21.2.2.3 SCC, MSS, INTビットの競合

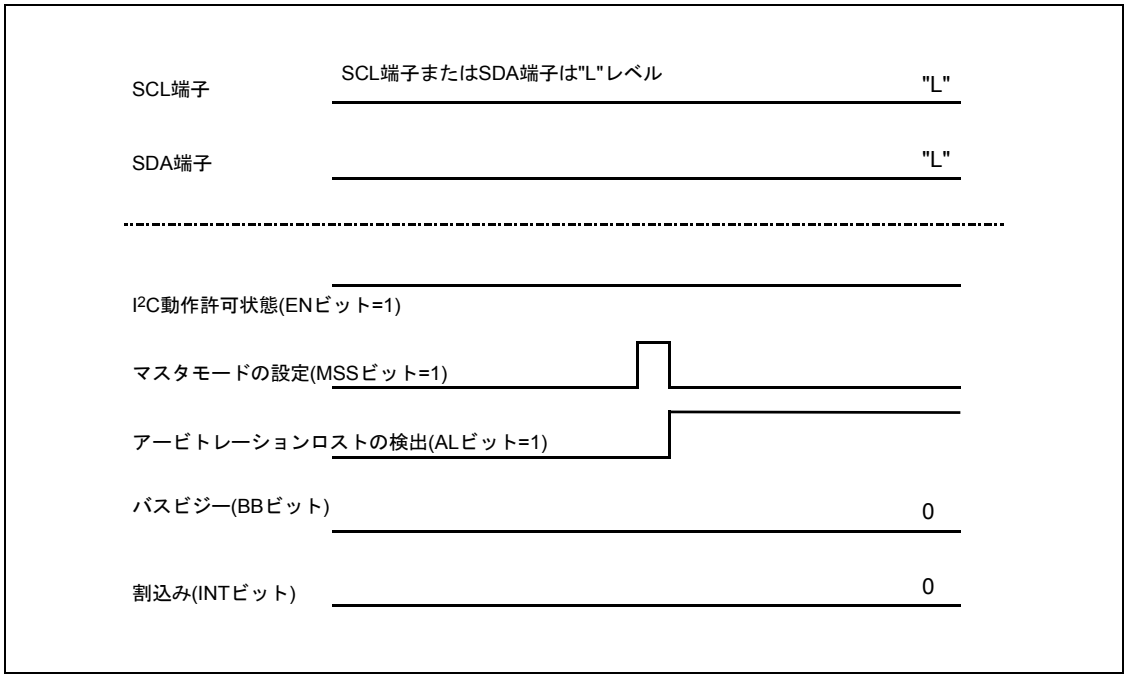
SCC, MSS, および INT ビットに同時に書込みを行うと , 次のバイトの転送 , リピートスタート条件の生成 , およびストップ条件の生成が競合します。このときの優先順位を下記に示します。

- 次のバイトの転送とストップ条件の生成－INTビットに"0"をMSSビットに"0"を書き込んだ場合は, MSSビットが優先されストップ条件が生成されます。
- 次のバイト転送とスタート条件の生成 : INT ビットに"0" が書き込まれ, SCC ビットに"1"が書き込まれると, SCC ビットが優先します。再起動条件が発生し, IDAR レジスタの内容が送信されます。
- リピートスタート条件の生成とストップ条件の生成－SCCビットに"1"をMSSビットに"0"を書き込んだ場合は, MSSビットのクリアが優先されます。ストップ条件が生成され, インタフェースはスレーブモードになります。Figure 21-5およびFigure 21-6に示したタイミングで, スタートコンディションを発生させる命令を実行 (MSS ビットに"1" を設定) すると, アービトレーションロスト検出 (ALビット= 1) による割込み (INTビット= 1) が発生しません。

■ 事例1－SCLおよびSDA信号が"L"を保持している場合

BBビットが"0"の(スタート条件が検出されていない)とき, MSSビットを"1"にセットすると直ちにALビットがセットされます。ただし , このときは AL ビットによる INT ビットのセットは行われません。

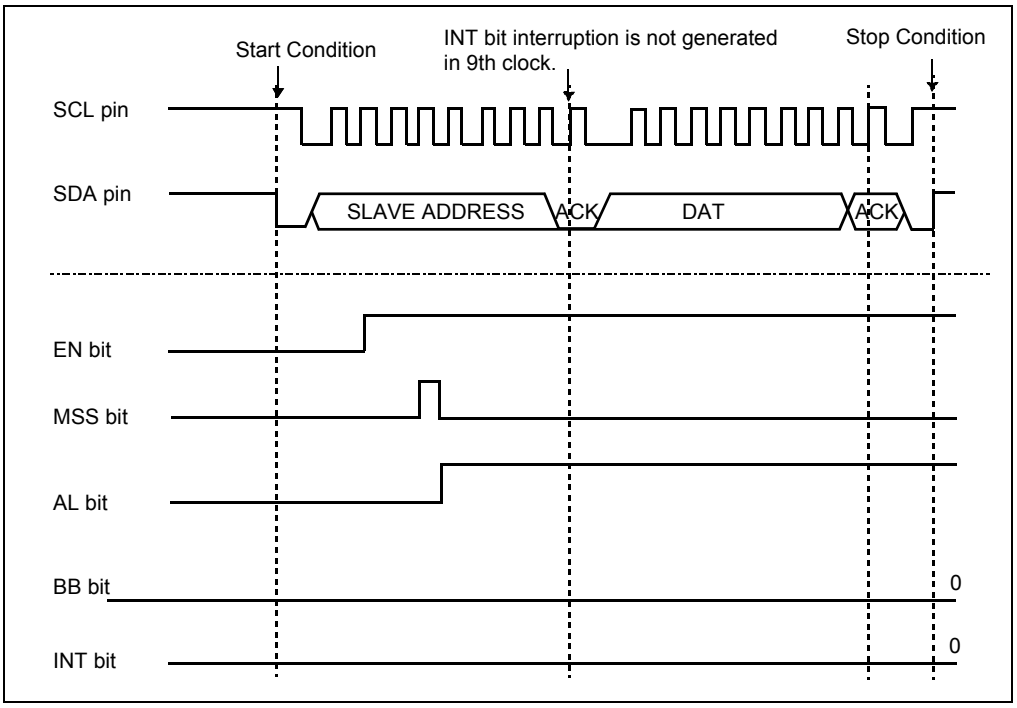
Figure 21-5. 「ALビット = 1」検出時に割込みが発生しない場合のタイミングチャート



■ 事例2－ほかのバスマスタとの通信中にI2Cインタフェースを許可したとき

現在実行中の通信によりバスが占有されているときに , EN ビットを "0" から "1" にセットしてインタフェースを I2C バスに参加させます。この場合は , BB ビットが "0" のままですから (スタート条件が検出されない), MSS ビットを "1" にセットすると AL ビットは "1" を示します。ただし , このときは AL ビットによる INT ビットのセットは行われません。

Figure 21-6. 「ALビット = 1」検出時に割込みが発生しない場合のタイミングチャート



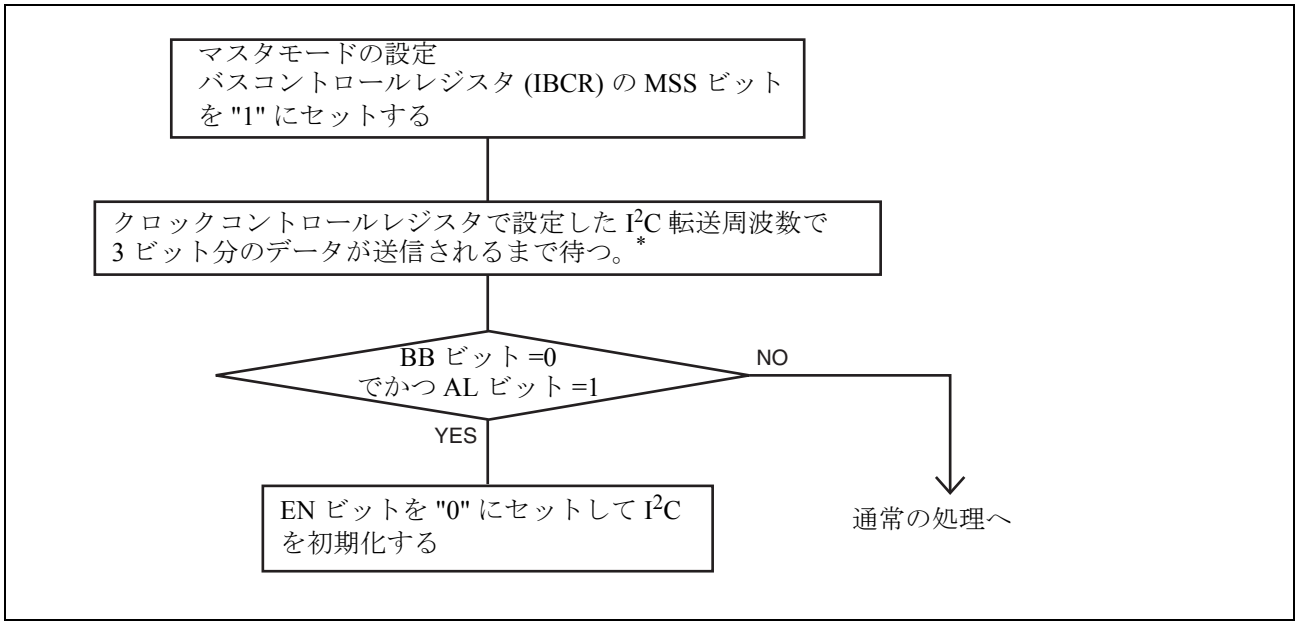
上記の現象が発生した場合 , 次の手順に従ってソフトウェアによる処理を実行します。

1. スタート条件を生成する命令を実行する (MSS ビットを "1" にセットする)。
2. タイマ機能などを使用して , ICCR レジスタで設定した I^2C 転送周波数で 3 ビット分データが送信されるのを待つ。*

   例 : 100 kHz の I^2C 転送周波数での 3 ビット分のデータ送信に要する時間 =

   $(1 / (100 \times 10^3)) \times 3 = 30$
3. IBSR レジスタの AL ビットと BB ビットをチェックし , AL ビットおよび BB ビットがそれぞれ "1" および "0" の場合は ICCR レジスタの EN ビットを "0" にして I^2C を初期化する。AL ビットと BB ビットの値がこれと異なる場合は , 通常の処理を実行する。

サンプルフローを下記に示します。

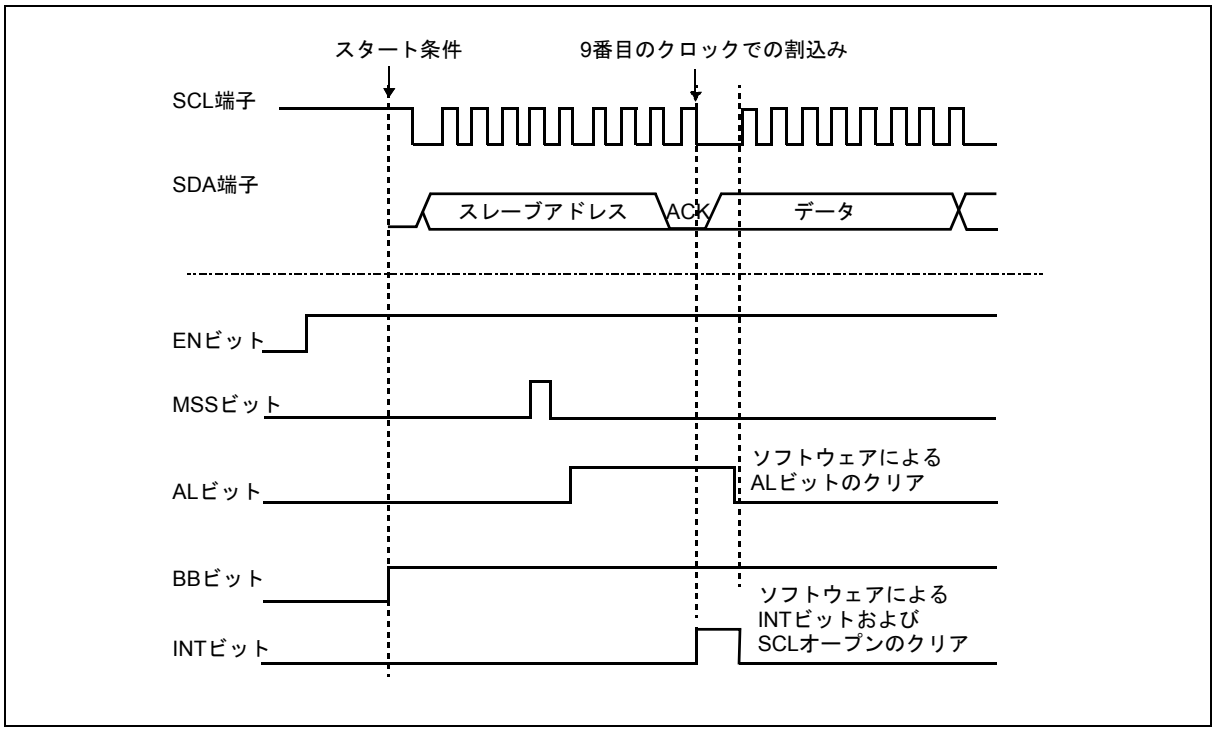


*: アービトレーションロストが検出された場合 , MSS ビットを "1" にセットして , I2C 転送周波数での 3 ビットデータ送信時間後には AL ビット = 1 となります。

■ 「ALビット=1」の検出による割込み発生(INTビット=1)の例

スタート条件を生成する命令を実行した (MSS ビット =1 にセット) ときに「バスビジー」が検出され (BB ビット =1), アービトレーションロストが発生したときに場合 ,「AL ビット =1」が検出されると直ちに INT ビット割込みが発生します。

Figure 21-7. 「ALビット = 1」検出時に割込みが発生する場合のタイミングチャート



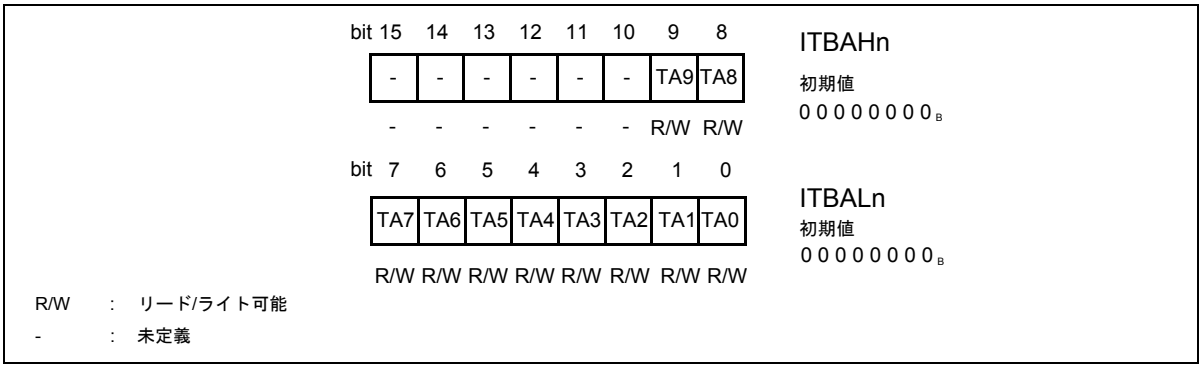
21.2.3 10ビットスレーブアドレスレジスタ(ITBAn)

10ビットスレーブアドレスレジスタ(ITBAn)は, 10ビットのスレーブアドレスの指定に使用します。

10ビットスレーブアドレスレジスタ(ITBAn)

本レジスタへのライトアクセスは , インタフェースを禁止にした (ICCR: EN=0) 場合にのみ可能です。

Figure 21-8. 10ビットスレーブアドレスレジスタ(IBTAn)の構成



10ビットスレーブアドレスレジスタ(ITBAn)の内容

Table 21-3. 10ビットスレーブアドレスレジスタ(ITBAn)の各ビットの機能

ビット名		機能
bits 15 ～ bit 10	未定義	これらのビットは常に"0"を返します。
bits 9 ～ bit 0	TA9 ～ TA0: 10ビットスレーブアドレス	スレーブモードでアドレスデータを受信すると , 10 ビットアドレスを使用するように設定している場合は (ITMK: ENTB=1), ITBA レジスタとの比較が行われます。ライトアクセスの 10 ビットアドレスヘッダ *1 を受信した後でアクノリッジがマスタに送信されます。次に , 2 番目に受信したバイトが TBAL レジスタと比較されます。一致した場合は , アクノリッジ信号がマスタデバイスに送信され , AAS ビットがセットされます。 また , リピートスタート条件が検出された後 (IBSRn レジスタの RSC=1) では , リードアクセスの 10 ビットヘッダ *2 受信時にもインタフェースはアクノリッジします。 そのときスレーブアドレスのすべてのビットはITMKレジスタを使用してマスクすることができます。10ビットスレーブアドレスを受信後,その値はITBAレジスタに書き戻され, IBSRレジスタのAASビットが"1"の間のみ有効です。

*1: 10 ビットヘッダ (ライトアクセス) は「11110, TA9, TA8, 0」のビットシーケンスで構成されています。

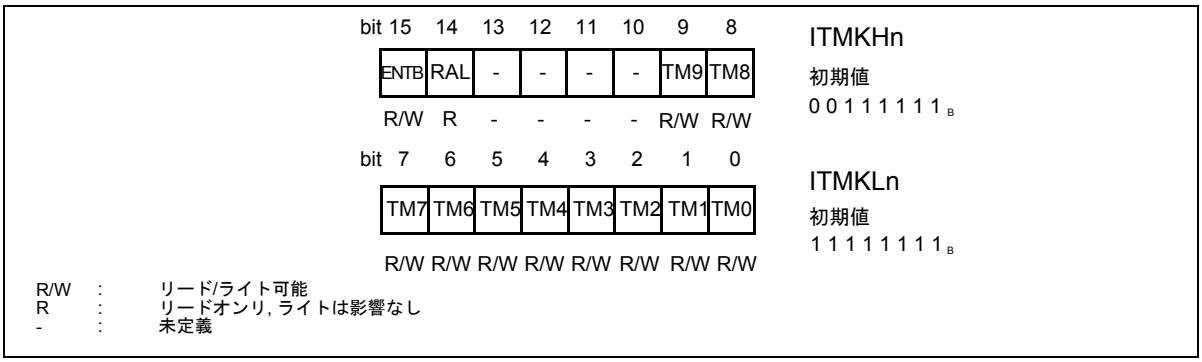
*2: 10ビットヘッダ(リードアクセス)は「11110, TA9, TA8, 1」のビットシーケンスで構成されています。

21.2.4 10ビットスレーブアドレスマスクレジスタ(ITMKn)

10 ビットスレーブアドレスマスクレジスタ (ITMKn) には , 10 ビットのスレーブアドレスマスクと 10 ビットスレーブアドレス許可ビットが格納されています。

10ビットアドレスマスクレジスタ(ITMKn)

Figure 21-9. 10ビットアドレスマスクレジスタ(ITMKn)の構成



10ビットアドレスマスクレジスタ(ITMKn)の内容

Table 21-4. 10ビットアドレスマスクレジスタ(ITMKn)の各ビットの機能

ビット名		機能
bit 15	ENTB: 10ビット スレーブ アドレス 許可	10 ビットスレーブアドレス (および受信時のアクノリッジ) を許可します。本ビットへのライトアクセスは , I^2C インタフェースを禁止にした (ICCR: EN=0) 場合にのみ可能です。 "0": 10 ビットアドレス禁止 "1": 10ビットアドレス許可
bit 14	RAL: 受信スレーブアドレス長ビット	7 ビットスレーブと 10 ビットスレーブのどちらでアドレッシングされたかを示します。リードオンリです。 "0": 7 ビットスレーブとしてアドレッシングされた。 "1": 10 ビットスレーブとしてアドレッシングされた。 両方のスレーブアドレスを許可している(ENTB=1およびENSB=1)場合は, 本ビットを使用して, インタフェースが7ビットスレーブとしてアドレッシングされたのか10ビットスレーブとしてアドレッシングされたのかを判別することができます。IBSRレジスタのAASビットが"1"の場合にのみ有効です。I^2Cインタフェースを禁止する(ICCR: EN=0)とリセットされます。
bit 13 ～ bit 10	未定義	読出しでは常に"1"が返されます。
bit 9 ～ bit 0	TM9 ～ TM0: 10ビット スレーブ アドレス マスク	本レジスタは , インタフェースの 10 ビットスレーブアドレスのマスクに使用します。本ビットへのライトアクセスは , I^2C インタフェースを禁止にした (ICCR: EN=0) 場合にのみ可能です。 "0": 本ビットをスレーブアドレスの比較に使用しない。 "1": 本ビットをスレーブアドレスの比較に使用する。 I^2C インタフェースから複数の 10 ビットスレーブアドレスに関するアクノリッジを返すのに使用できます。本レジスタの "1" にセットしたビットだけが 10 ビットスレーブアドレスの比較に使用されます。受信したスレーブアドレスが ITBA レジスタに書き戻され , IBSR レジスタの AAS ビットが "1" のときに ITBA レジスタを読み出すことによってスレーブアドレスを決定することができます。 (注意事項) I^2Cインタフェースを許可にした後でアドレスマスクを変更した場合は, スレーブアドレスが前回受信したスレーブアドレスで上書きされている可能性があるため, スレーブアドレスをセットし直す必要があります。

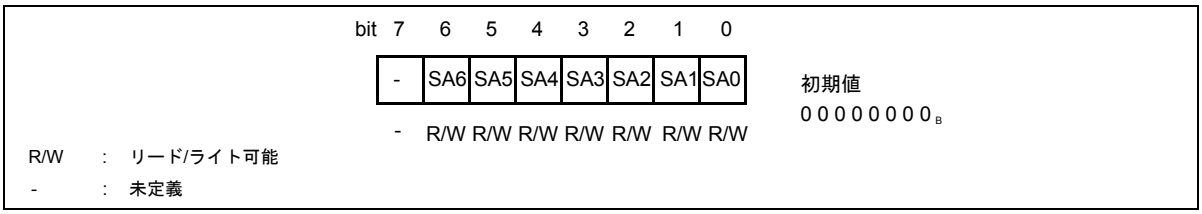
21.2.5 7ビットスレーブアドレスレジスタ(ISBAn)

7ビットスレーブアドレスレジスタ(ISBAn)には7ビットスレーブアドレスが格納されています。

7ビットスレーブアドレスレジスタ(ISBAn)

本レジスタへのライトアクセスは , I^2C インタフェースを禁止にした (ICCR: EN=0) 場合にのみ可能です。

Figure 21-10. 7ビットスレーブアドレスレジスタ(ISBAn)の構成



7ビットスレーブアドレスレジスタ(ISBAn)の内容

Table 21-5. 7ビットスレーブアドレスレジスタ(ISBAn)の各ビットの機能

ビット名		機能
bit 7	未定義	読出しでは常に"0"が返されます。
bit 6 ～ bit 0	SA6 ～ SA0: 7ビットスレーブアドレス	スレーブモードでアドレスデータを受信すると , 7 ビットアドレスを使用するように設定している場合は (ITMK: ENSB=1), 受信したアドレスデータと ISBA レジスタとの比較が行われます。アドレスデータが一致した場合は , アクノリッジ信号がマスタデバイスに送信され , AAS ビットがセットされます。 スレーブアドレスのすべてのビットを ISMK レジスタを使用してマスクすることができます。アドレスデータ受信後 , 受信した 7 ビットデータアドレスは ISBA レジスタに書き戻され , IBSR レジスタの AAS ビットが "1" の間のみ有効です。 10ビットヘッダまたはジェネラルコールを受信した場合は, このレジスタの内容と受信データの比較は行われません。

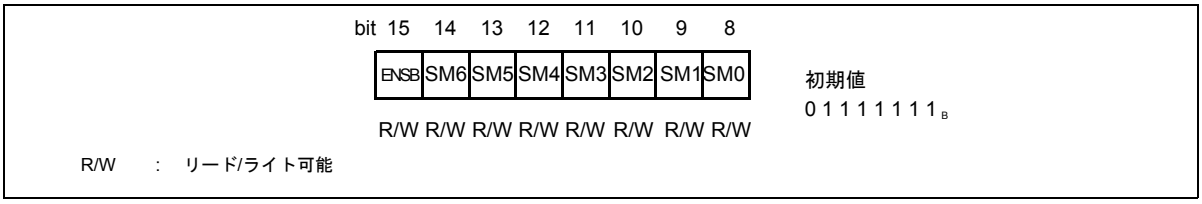
21.2.6 7ビットスレーブアドレスマスクレジスタ(ISMKn)

7ビットスレーブアドレスマスクレジスタ(ISMKn)には, 7ビットのスレーブアドレスマスクと7ビットスレーブアドレス許可ビットが格納されています。

7ビットスレーブアドレスマスクレジスタ(ISMKn)

本レジスタには 7 ビットスレーブアドレスマスクと 7 ビットモード許可ビットが格納されています。本レジスタへのライトアクセスは , I^2C インタフェースを禁止にした (ICCR: EN=0) 場合にのみ可能です。

Figure 21-11. 7ビットスレーブアドレスマスクレジスタ(ISMKn)の構成



7ビットスレーブアドレスマスクレジスタ(ISMKn)の内容

Table 21-6. 7ビットスレーブアドレスマスクレジスタ(ISMKn)の各ビットの機能

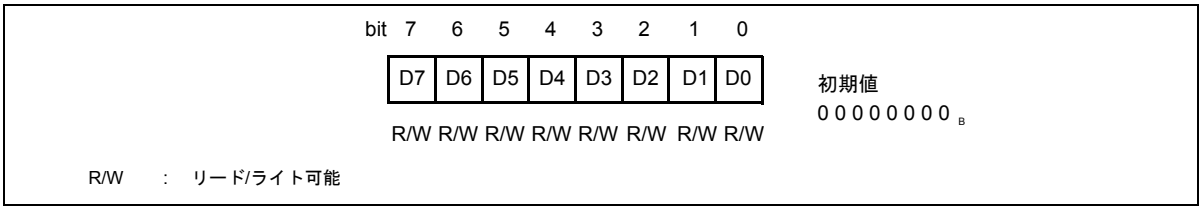
ビット名		機能
bit 15	ENSB: 7ビット スレーブ アドレス 許可	7 ビットスレーブアドレス (および受信時のアクノリッジ) を許可します。 "0": 7 ビットスレーブアドレス禁止 "1": 7ビットスレーブアドレス許可
bit 14 ～ bit 8	SM6 ～ SM0: 7ビット スレーブ アドレス マスク	本レジスタは , インタフェースの 7 ビットスレーブアドレスのマスクに使用します。 "0": 本ビットをスレーブアドレスの比較に使用しない。 "1": 本ビットをスレーブアドレスの比較に使用する。 I^2C インタフェースから複数の 7 ビットスレーブアドレスに関するアクノリッジを返すのに使用できます。本レジスタの "1" にセットしたビットだけが 7 ビットスレーブアドレスの比較に使用されます。スレーブアドレス受信後 , 受信したスレーブアドレスが ISBA レジスタに書き戻され , IBSR レジスタの AAS ビットが "1" のときに ISBA レジスタを読み出すことによってスレーブアドレスを決定することができます。 (注意事項) I^2Cインタフェースを許可にした後でアドレスマスクを変更した場合は, スレーブアドレスが前回受信したスレーブアドレスで上書きされている可能性があるため, スレーブアドレスをセットし直す必要があります。

21.2.7 データレジスタ(IDARn)

400 kHz I^2Cインタフェースのデータレジスタ(IDARn)はCPUによるデータの送受信に使用されます。

データレジスタ(IDARn)

Figure 21-12. データレジスタ(IDARn)の構成



データレジスタ(IDARn)の内容

Table 21-7. データレジスタ(IDARn)の各ビットの機能

ビット名		機能
bits 7 ～ bit 0	D7 ～ D0: データビット	データレジスタはシリアルデータ転送に使用され, MSBファースト形式でデータを転送します。本レジスタはライト側はダブルバッファのため, バスが使用中であっても(BB=1), シリアル転送用のレジスタにライトデータをロードすることができます。IBCRレジスタのINTビットがクリアされるかバスがアイドル状態になると(IBSR: BB=0), データバイトは内部転送レジスタにロードされます。リードアクセスの場合は内部レジスタが直接読み出されるため, 本レジスタの受信データ値はIBCR: INT=1の場合にのみ有効です。

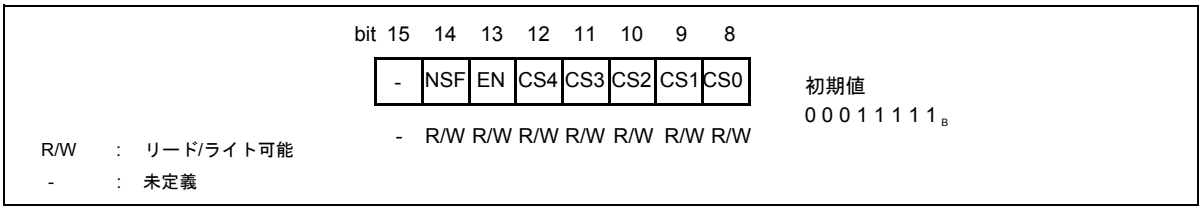
21.2.8 クロックコントロールレジスタ(ICCRn)

クロックコントロールレジスタ (ICCRn) には次の機能があります。

- テストモードの許可
- IOパッドノイズフィルタの許可
- I^2Cインタフェースの動作の許可
- シリアルクロック周波数の設定

クロックコントロールレジスタ(ICCRn)

Figure 21-13. クロックコントロールレジスタ(ICCRn)の構成



クロックコントロールレジスタ(ICCRn)の内容

Table 21-8. クロックコントロールレジスタ(ICCRn)の各ビットの機能

ビット名		機能
bit 15	未定義	読出しでは常に"0"が返されます。
bit 14	NSF: IOパッドノイズフィルタ許可ビット	SDAおよびSCL IOパッドに内蔵されているノイズフィルタを有効にします。 ノイズフィルタは, 周辺クロックCLKP1の1～1.5サイクルの間のパルス幅0 ns (最小値)のスパイクを除去します。最大値は, I^2Cの信号(SDA, SCL)と周辺クロックCLKP1の位相関係に依存します。 インタフェースが100 Kビット以上のデータ転送速度で送信または受信している場合は"1"にセットする必要があります。
bit 13	EN: 許可ビット	I^2C インタフェースを許可するビットです。セットできるのはユーザだけですが , クリアについては , ユーザおよびハードウェアからも行われます。 "0": 禁止 "1": 許可 本ビットを "0" にセットすると , IBSR レジスタおよび IBCR レジスタのすべてのビットがクリアされ (BER および BEIE ビットを除く), モジュールが禁止されて I^2C ラインはオープンのままになります。 バスエラーが発生した場合は (IBCR: BER=1), ハードウェアによってクリアされます。 (注意事項) I^2Cインタフェースを禁止すると直ちに送受信が停止し, I^2Cバスの状態が不安定になるおそれがあります。
bits 12 ～ bit 8	CS4 ～ CS0: クロックプリスケーラビット	シリアルビットレートの選択に使用します。I^2Cインタフェースが禁止状態 (EN=0)か, 同じライトアクセスでENビットがクリアされた場合にのみ変更することができます。

n	CS4	CS3	CS2	CS1	CS0	
1	0	0	0	0	1	ビットレート: ϕ/28(+1)
2	0	0	0	1	0	ビットレート: ϕ/40(+1)
3	0	0	0	1	1	ビットレート: ϕ/52(+1)
4	0	0	1	0	0	ビットレート: ϕ/64(+1)
…						
31	1	1	1	1	1	ビットレート: ϕ/400(+1)

(+1) はノイズフィルタが許可されているときには除数に1を加算することを意味します。

クロックプリスケーラの設定

CS0 ～ CS4 の計算式は次のように定義されます。

$$\text{ビットレート} = \frac{\phi}{n \times 12 + 16} \quad n>0 \quad \phi: \text{周辺クロック(CLKP1), ノイズフィルタ禁止}$$

$$\text{ビットレート} = \frac{\phi}{n \times 12 + 17} \quad n>0 \quad \phi: \text{周辺クロック(CLKP1), ノイズフィルタ許可}$$

Table 21-9. プリスケーラの設定

n	CS4	CS3	CS2	CS1	CS0
1	0	0	0	0	1
2	0	0	0	1	0
3	0	0	0	1	1
...					
31	1	1	1	1	1

(注意事項) SDA/SCLタイミングに反するためn=0のプリスケーラ設定は使用しないでください。

一般的な周辺クロックCLKP1周波数

Table 21-10に最も一般的な周辺クロック(CLKP1)の周波数とプリスケーラの設定および送信ビットレートを示します。

Table 21-10. 一般的な周辺クロックCLKP1周波数

CLKP1周波数 [MHz]	100 Kビット (ノイズフィルタ禁止)		400 Kビット (ノイズフィルタ許可)	
	n	ビットレート[Kビット]	n	ビットレート[Kビット]
24	19	98	4	369
20	16	96	3	377
16	12	100	2	390
40/3 = 13.3	10	98	2	325
12	9	96	2	292
64/6 = 10.6	8	94	1	367
10	7	100	1	344
8	6	90	1	275

21.3 I²Cインタフェースの動作

I²Cバスは, シリアルデータライン(SDA)とシリアルクロックライン(SCL)の2本の双方向バスラインを使用して通信を実行します。I²Cインタフェースには, この2本のラインに対応した2本のオープンドレインI/O端子 (SDA/SCL)があり, ワイヤードロジックで使用することができます。

スタート条件

バスが空いている (IBSR: BB=0, IBCR: MSS=0) ときに MSS ビットに "1" を書き込むと ,I²C がマスタモードになりスタート条件が生成されます。
バスのアイドル時 (MSS=0 および BB=0) に MSS ビットに "1" を書き込むと , スタート条件が生成され , IDAR レジスタの内容 (アドレスデータ) が送信されます。
バスマスタモードで割込み状態 (MSS=1 および IBCR: INT=1) のとき , SCC ビットに "1" を書き込むことでリピートスタート条件を生成することができます。
バスが使用されている (IBSR:BB=1, IBSR:TRX=0, IBCR:MSS=0 および IBCR:INT=0) ときに MSS ビットに "1" を書き込むと , I²C インタフェースはバスが空くのを待ってから送信を開始します。
送信中に I²C インタフェースがライトアクセス (データ受信) のスレーブとしてアドレッシングされた場合は , その転送が終了してバスが空いたときに再び送信が開始されます。このときインタフェースがスレーブとしてデータを送信していた場合は, バスが空いても送信は再開されません。次の割込み時に, インタフェースがスレーブとしてアドレッシングされた (IBCR: MSS=0 および IBSR: AAS=1) かどうか , データバイトが正しく送信されたか (IBCR: MSS=1), データバイトの送信に失敗したか (IBSR: AL=1) をチェックすることが重要です。
ほかの状態では , MSS ビットまたは SCC ビットに "1" を書き込んでも無効です。

ストップ条件

マスタモード (IBCR: MSS=1 および INT=1) で MSS ビットに "0" を書き込むとストップ条件が生成され , デバイスはスレーブモードに移行します。ほかの状態では , MSS ビットに "0" を書き込んでも無効です。
MSSビットをクリアするとインタフェースはストップ条件の生成を試行しますが, ストップ条件が生成される前に何らかの外部条件によって SCL ラインの信号が "1" から "0" に遷移した場合はストップ条件の生成に失敗することがあります。その場合は AL ビットが "1" にセットされ , 次のバイトの処理完了後に割込みが発生します。

スレーブアドレスの検出

スレーブモードでは, スタート条件が生成されると BB が "1" にセットされ, マスタデバイスから送信されたデータが受信されて IDAR に格納されます。
ISMK レジスタの ENSB ビットが "1" にセットされている場合は , 8 ビット分を受信すると , ISMK に格納されているビットマスクを使用して IDAR レジスタの内容が ISBA レジスタの内容と比較されます。一致した場合は AAS ビットが "1" にセットされ , マスタに対してアクノリッジ信号が送信されます。次に , 受信したデータの bit0 が反転されて TRX ビットに格納されます。

ITMK レジスタの ENTB ビットが "1" で 10 ビットアドレスヘッダ (11110, TA1, TA0, ライトアクセス) が検出された場合は , I²C インタフェースからマスタに対してアクノリッジ信号が送信され , 最後のデータビットが反転されて TRX レジスタに格納されます。割込みは生成されません。その後 , 次の転送バイトと ITBA レジスタの下位バイトとの比較が行われます (ITMK に格納されているビットマスクを使用)。一致した場合は , マスタに対してアクノリッジ信号が送信され , AAS ビットがセットされて , 割込みが生成されます。
I²C インタフェースがスレーブとしてアドレッシングされリピートスタート条件が検出された場合は , 10 ビット

アドレスヘッダ(11110, TA1, TA0, リードアクセス)の受信後にAASビットがセットされ, 割込みが生成されます。7 ビットアドレスと 10 ビットアドレスおよびそのビットマスクには個別のレジスタが存在するため , ISMK: ENSBビットおよびITMK: ENTBビットをセットして, 両方のアドレスに関するアクノリッジをI^2Cインタフェースから返すことができます。受信したスレーブアドレスの長さ (7 ビットまたは 10 ビット) は , ITMK レジスタの RAL ビットを読み出すことで決定することができます (このビットは , AAS ビットがセットされている場合にのみ有効です)。

また , I^2Cインタフェースをマスタとしてのみ使用する場合は, この両方のビットを "0" にセットしてスレーブアドレスを受信しないようにすることもできます。
対応するマスクレジスタ (ITMK または ISMK) を使用して , すべてのアドレスビットをマスクすることができます。

スレーブアドレスのマスキング

マスクレジスタ (ITMK または ISMK) で "1" にセットしたビットだけがアドレス比較に使用され , ほかはすべて無視されます。IBSR レジスタの AAS ビットが "1" の場合は , 受信したスレーブアドレスを ITBA レジスタ (10 ビットアドレスが受信され RAL=1 の場合) または ISBA レジスタ (7 ビットアドレスが受信され RAL=0 の場合) から読み出すことができます。
ビットマスクをクリアすると , I^2C インタフェースは常にスレーブとしてアドレッシングされるため , I^2C インタフェースをバスモニタとして使用することができます。ただし , 実際のバスモニタとは異なり , ほかにバスを監視しているスレーブが存在しない場合でも, スレーブアドレスを受信するとアクノリッジが返されるため注意が必要です。

スレーブへのアドレッシング

マスタモードでは , スタート条件が生成されると , BB ビットと TRX ビットが "1" にセットされ , IDAR レジスタの内容が MSB ファースト形式で送信されます。アドレスデータの送信後にスレーブデバイスからアクノリッジ信号を受信すると , 送信データの bit0 (送信後の IDAR レジスタの bit0) が反転されて TRX ビットに格納されます。IBSR レジスタの LRB ビットを使用してスレーブからのアクノリッジをチェックすることができます。この手順はリピートスタート条件にも適用することができます。
10 ビットスレーブにライトアクセスするには 2 バイトの送信が必要です。最初のバイトは "1 1 1 1 0 A9 A8 0" のビットシーケンスで構成された 10 ビットアドレスヘッダで , これに続く 2 番目のバイトには 10 ビットスレーブアドレスの下位 8 ビット (A7 ～ A0) を格納します。
上記のバイトシーケンスを送信してリピートスタート条件を生成し(IBCR: SCC ビット), リードアクセスの 10 ビットアドレスヘッダ (1 1 1 1 0 A9 A8 1) を送信することにより , 10 ビットスレーブをリードアクセスすることができます。

アドレスデータバイトのまとめ

7 ビットスレーブ , ライトアクセススタート条件ー A6 A5 A4 A3 A2 A1 A0 0

7 ビットスレーブ , リードアクセススタート条件ー A6 A5 A4 A3 A2 A1 A0 1

10 ビットスレーブ , ライトアクセススタート条件ー 1 1 1 1 0 A9 A8 0 - A7 A6 A5 A4 A3 A2 A1 A0

10 ビットスレーブ , リードアクセススタート条件ー 1 1 1 1 0 A9 A8 1 - A7 A6 A5 A4 A3 A2 A1 A0, リピートスタートー 1 1 1 1 0 A9 A8 1

アービトレーション

マスタモードでの送信時に同時にほかのマスタがデータを送信している場合は, アービトレーションが実行されます。あるデバイスがデータ値 "1" を送信したとき SDA ラインのデータが "L" レベルの値の場合は , そのデバイスはアービトレーションをロストしたと見なされ , AL ビットが "1" にセットされます。また , データバイトの最初のビットでスタート条件が検出されたにもかかわらずインタフェースがスタート条件を生成しなかったか, 何らかの理由でスタート条件またはストップ条件の生成に失敗した場合にも AL ビットが "1" にセットされます。アービトレーションロストが検出されると , MSS ビットおよび TRX ビットの両方がクリアされ , デバイスは自身のスレーブアドレスが送信された場合にアクノリッジを返すことができるように直ちにスレーブモードに移行します。

アクノリッジ

アクノリッジビットはレシーバからトランスミッタに送信されます。IBCR レジスタの ACK ビットを使用して , データバイトの受信時にアクノリッジを送信するかどうかを選択することができます。

スレーブモードでデータを送信したとき (ほかのマスタからのリードアクセス), マスタからアクノリッジが返されなかった場合は TRX ビットが "0" にセットされ , デバイスは受信モードに移行します。これにより , スレーブが SCL ラインを開放すると直ちにマスタはストップ条件を生成することができます。

マスタモードでは , IBSR レジスタの LRB ビットを読み出すことによりスレーブからのアクノリッジをチェックすることができます。

21.4 プログラミングフローチャート

400 kHz I^2Cインタフェースのプログラミングフローチャートを下記に示します。

プログラミングフローチャート

Figure 21-14. スレーブアドレッシングとデータ送信の例

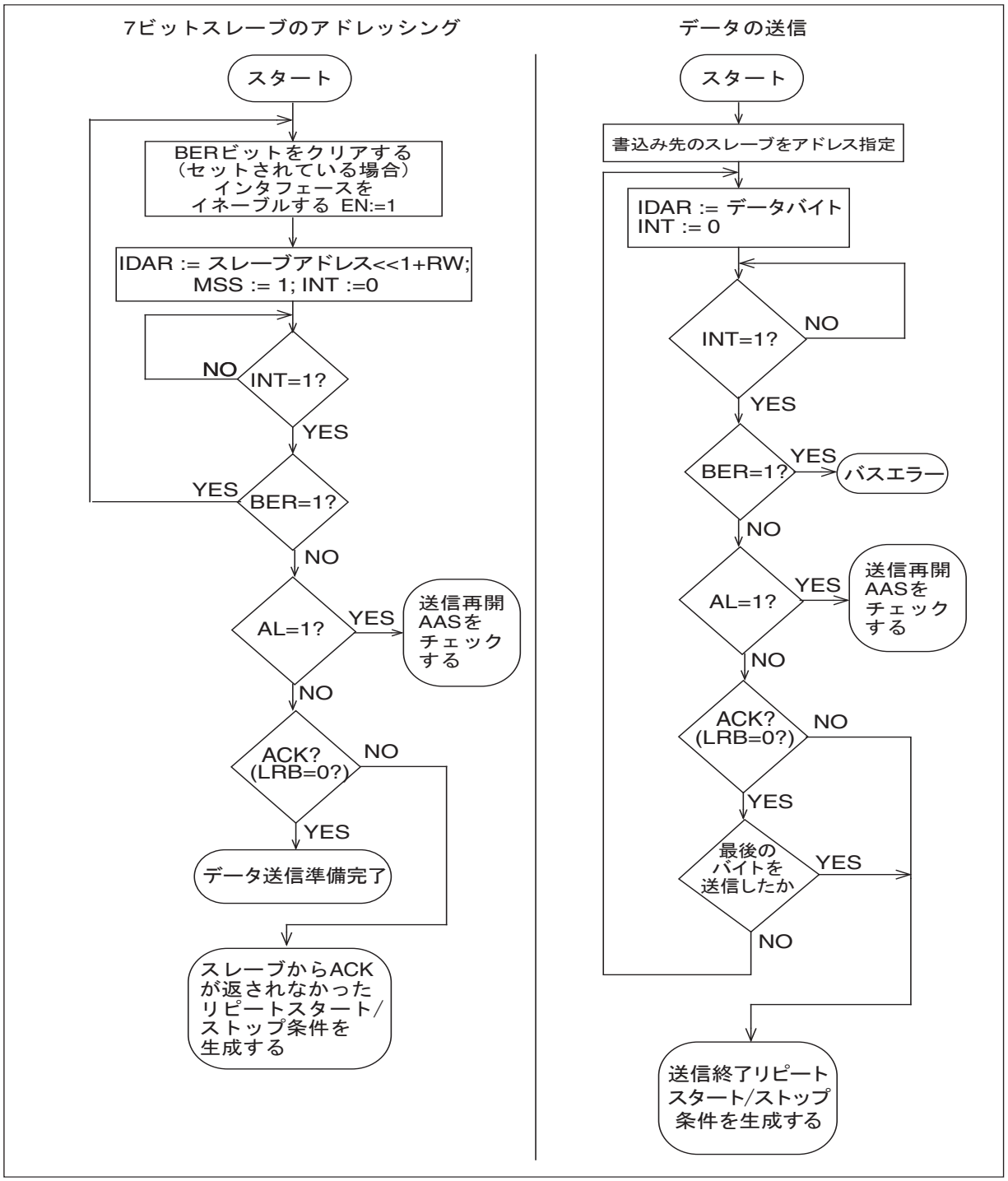
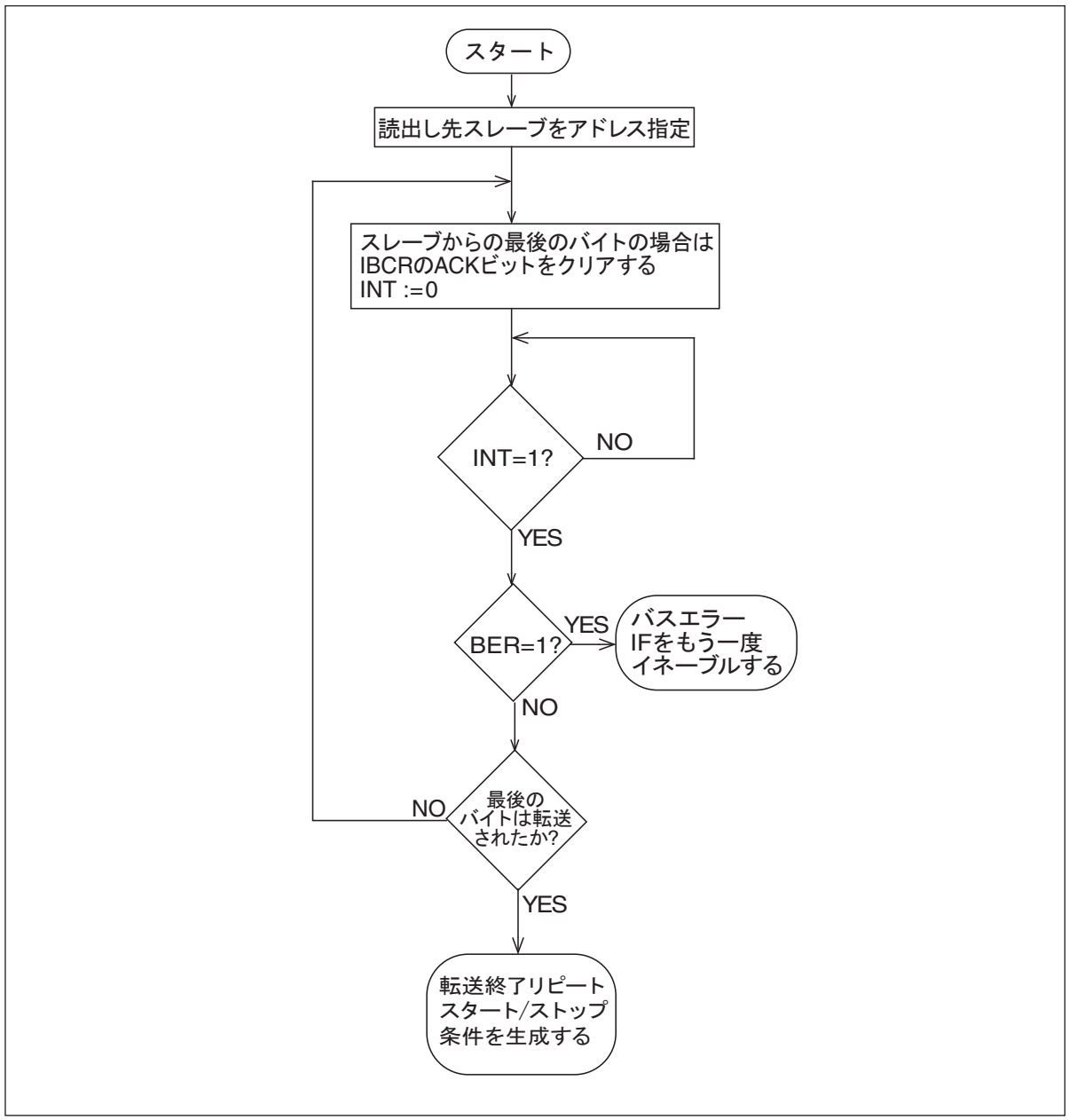


Figure 21-15. データ受信の例



22. CANコントローラ

CANコントローラの機能と動作を説明します。

22.1 概要

CANはCANプロトコルVer 2.0 part AおよびVer 2.0 part Bに準拠して通信を実行します。プログラムによって最大1Mbpsのビットレートが設定できます。物理層への接続には別途トランシーバハードウェアが必要です。

特長

CAN には次の特長があります。

- CANプロトコルVer 2.0 part AおよびVer 2.0 part Bに準拠
- 最大1Mbpsのビットレート
- 32 (最大128)のメッセージオブジェクト
- 各メッセージオブジェクト個別のIDマスク
- プログラミング可能なFIFOモード(メッセージオブジェクトの連結)
- マスク可能な割込み
- タイムトリガCANアプリケーション用自動再送信無効モード
- 自己診断動作用のプログラミング可能なループバックモード

＜注意事項＞

本章で使用するレジスタ名のサフィックス "n" は CAN コントローラ番号を示します。

略語

本章で使用する用語および略語を下記に示します。

用語	意味
CAN	コントローラエリアネットワーク
BSP	ビットストリームプロセッサ
BTL	ビットタイミングロジック
CRC	巡回冗長検査
DLC	データ長コード
EML	エラー制御ロジック
FSM	(有限)ステートマシン
TTCAN	タイムトリガ型CAN

22.2 機能の説明

CANモジュールの動作モードの概要とその使用方法を説明します。

ソフトウェアによる初期化

ソフトウェアリセットかハードウェアリセットまたはバスオフを実行することにより CAN 制御レジスタ CTRLRLn の INIT ビットをセットすると , ソフトウェアによる初期化が開始されます。

INIT をセットすると , CAN バスを通じたすべてのメッセージ転送が停止され , CAN バス出力のステータス CAN_TX がレセッシブ (HIGH) になります。EML のカウンタは変化しません。INIT をセットしても構成レジスタは変化しません。

CAN コントローラを初期化するには , CPU によってビットタイミングレジスタ BTRn と各メッセージオブジェクトを設定する必要があります。メッセージオブジェクトが不要な場合は , MSGVAL ビットを無効に設定するだけで十分ですが , そうでない場合はメッセージオブジェクト全体を初期化する必要があります。

CAN 制御レジスタ CTRLRLn の INIT ビットと CCE ビットの両方がセットされているときに , ビットタイミングレジスタ BTRn および BRP 拡張レジスタ BRPERn へのアクセスによる , ビットタイミングの設定が可能になります。

INIT がリセットされると (CPU によるもののみ), ソフトウェアによる初期化が完了します。次に , ビットストリームプロセッサ BSP は 11 のレセッシブビット (= バスアイドル) の連続シーケンスが発生するのを待って , 自身と CAN バス上のデータ転送との同期を行ってバスアクティビティに参加し , メッセージの転送を開始します。

メッセージオブジェクトの初期化は INIT とは無関係で並行して実行することができますが , BSP がメッセージの転送を開始する前にすべてのメッセージオブジェクトを所定の ID に設定するか無効に設定する必要があります。

通常の動作中にメッセ 0 次オブジェクトの設定を変更するには , CPU はまず MSGVAL ビットを無効にする必要があります。また , 設定が完了したら再び MSGVAL ビットを有効設定する必要があります。

CANメッセージの転送

CAN の初期化が完了し INIT を 0 にリセットすると , CAN コアと CAN バスの同期が行われ , メッセージの転送が開始されます。

受信したメッセージからメッセージハンドラの受容フィルタフィルタリングが渡された場合は , 受信したメッセージは対応するメッセージオブジェクトに格納されます。アービトレーションビット , DLC, および 8 データバイトを含むメッセージ全体がメッセージオブジェクトに格納されます。ID マスクを使用している場合は , アービトレーションビットは「無効 (don’t care)」とマスクされ , メッセージオブジェクト内で上書きすることができます。

CPUはいつでもインタフェースレジスタを通じて各メッセージを読出しまたは書込みをすることができ, 同時アクセス時のデータの一貫性はメッセージハンドラによって保証されます。

送信するメッセージは CPU によって更新します。送信メッセージに対する永久メッセージオブジェクト (設定時にアービトレーションビットと制御ビットをセット) が存在する場合はデータバイトだけが更新され , TXRQST ビットと NEWDAT ビットがセットされて送信が開始されます。同一のメッセージオブジェクトに複数の送信メッセージを割り当てた場合は(メッセージオブジェクトの数が不足した場合), そのメッセージを送信する前にメッセージオブジェクト全体を構成する必要があります。

同時に複数のメッセージオブジェクトの送信が要求された場合は, 内部的な優先順位に基づいて順次に送信されます。要求された送信が保留されているときでも , 随時にメッセージの更新または無効設定を行うことができます。メッセージの更新があった場合は , 保留中の送信が開始される前に前のデータは破棄されます。メッセージオブジェクトの構成によっては , ID の一致したリモートフレームの受信により自律的にメッセージの送信が要求されることがあります。

自動再送信無効モード

CAN の仕様 (「ISO11898, 6.3.3 Recovery Management (回復制御)」参照) に基づき , CAN にはアービトレーションロストしたり送信時のエラーによって散逸したフレームを自動的に再送信したりする機能が備えられています。送信が正しく完了されない限り , ユーザーに対しフレーム送信サービスが確立されることはありません。デフォルトでは , この自動再送信機能が有効に設定されています。タイムトリガ CAN (TTCAN, 「ISO11898-1」参照) 環境で CAN を使用する場合は , この機能を無効に設定することができます。自動再送信無効モードを使用するには , CAN 制御レジスタ CTRLRn の DAR ビットを "1" にセットします。この動作モードを使用すると , 次の場合にメッセージバッファの制御レジスタの TXRQST ビットと NEWDAT ビットの働きが通常とは異なることに注意してプログラミングする必要があります。

各メッセージバッファの送信開始ビット TXRQST をリセットしても NEWDAT ビットはセットしたままになります。

送信正常完了時に NEWDAT ビットがリセットされます。

送信に失敗した場合 (アービトレーション喪失あるいはエラー), NEWDAT ビットはセットされたままになります。送信を再開するには , CPU が TXRQST を "1" に設定し直す必要があります。

ホスト側が複数メッセージの同時送信を要求した場合 , そのうちの 2 つのメッセージだけが送信されます。要求された他のすべての送信メッセージに対しては , TXRQST ビットがリセットされますが , 送信は開始されず , NEWDAT と INTPND はそのままの状態です。送信される 2 つのメッセージに関しては , TXRQST ビットと NEWDAT ビットがリセットされ , TXIE によって許可されている場合 , INTPND がセットされます。

テストモード

CAN 制御レジスタの TEST ビットを "1" にセットするとテストモードに移行します。テストモードでは , テストレジスタ TESTRn の TX1, TX0, LBACK, SILENT, BASIC の各ビットが書込み可能になります。RX ビットは CAN_RX 端子の状態を監視しているため読取り専用です。TEST ビットを "0" にリセットすると , テストレジスタのすべての機能が無効になります。

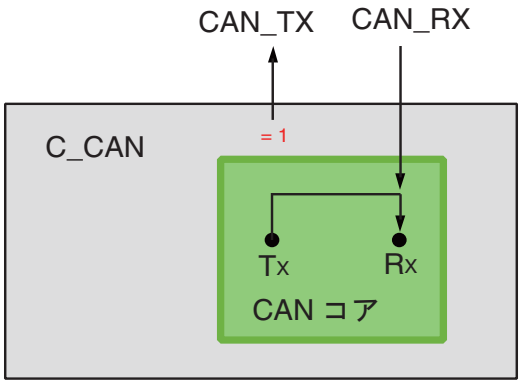
サイレントモード

テストレジスタ (TESTRn) の SILENT ビットを "1" にセットすると , CAN コアをサイレントモードにすることができます。

サイレントモードでは , 有効なデータフレームと有効なリモートフレームの受信を行うことはできますが , CAN バスにはレセッシブビットだけが送信され送信は開始されません。ドミナントビット (ACK ビット , オーバロードフラグ , アクティブエラーフラグ) の送信が必要な場合は , 内部的に経路が変更されて CAN コアでこのドミナントビットの監視が行われ , CAN バスはレセッシブ状態を保持することができます。サイレントモードを使用してドミナントビット (ACK ビット , エラーフレーム) を送信することにより , CAN バスに影響を与えずに CAN バス上のトラフィックの分析を行うことができます。Figure 22-1 にサイレントモードでの CAN_TX 信号および CAN_RX 信号と CAN コアの接続を示します。

ISO 11898-1 では , このサイレントモードはバス監視モードとよばれます。

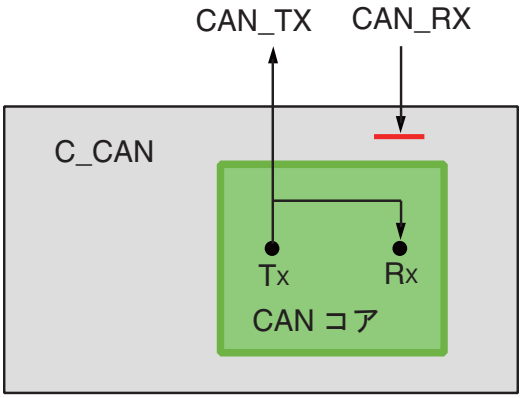
Figure 22-1. サイレントモードのCANコア



ループバックモード

テストレジスタ TESTRn の LBACK ビットを "1" にセットすることにより CAN コアをループバックモードにすることができます。ループバックモードでは , CAN コアは自身が送信したメッセージを受信したメッセージと見なして , 受信バッファに格納します (アクセプタンスフィルタを通過した場合)。Figure 22-2 にループバックモードでの CAN_TX 信号および CAN_RX 信号と CAN コアの接続を示します。

Figure 22-2. ループバックモードのCANコア

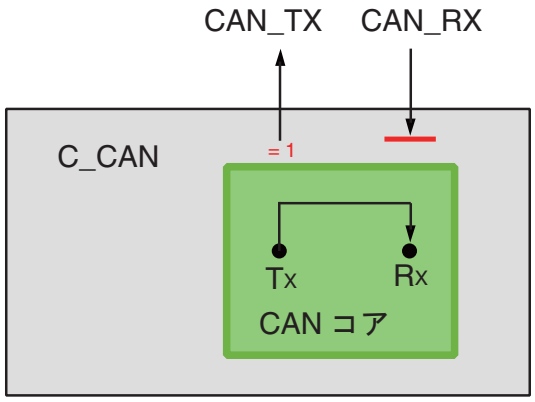


このモードは自己診断機能用です。ループバックモードでは , 外部シミュレーションに依存しないよう , CAN コアはアクノリッジエラー (データ / リモートフレームのアクノリッジスロット内にサンプリングされているレセッシブビット) を無視します。このモードでは , CAN コアは Tx 出力から Rx 入力への内部フィードバックを実行します。CAN_RX 入力端子の実際値は CAN コアで無視されます。送信したメッセージを CAN_TX 端子で監視することができます。

ループバックモードとサイレントモードとの組合せ

LBACK ビットと SILENT ビットを同時に "1" にすることにより , ループバックモードとサイレントモードを組み合せて使用することもできます。このモードを使用して「ホットセルフテスト」を実行する , つまり CAN_TX 端子と CAN_RX 端子に接続されている CAN システムの実行に影響を与えずに CAN のテストを行うことができます。このモードでは , CAN_RX 端子は CAN コアから切り離され , CAN_TX 端子はレセッシブに保持されます。Figure 22-3に, ループバックモードとサイレントモードを組み合せた場合のCAN_TX信号およびCAN_RX信号と CAN コアの接続を示します。

Figure 22-3. ループバックモードとサイレントモードを組み合せたモードのCANコア



ベーシックモード

テストレジスタ TESTRn の BASIC ビットを "1" にセットすると , CAN コアをベーシックモードにすることができます。このモードでは , メッセージ RAM を使用せずに CAN モジュールが実行されます。

IF1 レジスタは送信バッファとして使用されます。IF1 レジスタの内容の送信をリクエストするには , IF1 コマンドリクエストレジスタ IF1CREQn の BUSY ビットに "1" を書き込みます。BUSY ビットを "1" にセットすると IF1 レジスタはロックされます。BUSY ビットは送信が保留中であることを示します。

CAN バスがアイドル状態になると , 直ちに IF1 レジスタの内容が CAN コアのシフトレジスタにロードされて送信が開始されます。送信が完了すると BUSY ビットはリセットされ , ロックされていた IF1 レジスタが解放されます。

保留中の送信は, IF1 レジスタがロックされている間に IF1 コマンドリクエストレジスタ IF1CREQn の BUSY ビットをリセットすることにより , いつでも打ち切ることができます。CPU によって BUSY ビットをリセットした場合は , アービトレーションロストまたはエラー発生時の再送信は実行されません。

IF2 レジスタは受信バッファとして使用されます。メッセージを受信すると , アクセプタンスフィルタを使用せずにシフトレジスタの内容が IF2 レジスタに格納されます。

また , メッセージ転送中のシフトレジスタの実際の内容を監視することもできます。IF2 コマンドリクエストレジスタ IF1CREQn の BUSY ビットに "1" を書き込んでメッセージオブジェクト読取りを実行するたびに , シフトレジスタの内容が IF2 レジスタに格納されます。

ベーシックモードでは, 制御ビットとステータスビットに関するすべてのメッセージオブジェクトの評価と, IFx コマンドマスクレジスタ IF1CMSKn の制御ビットの評価は行われません。コマンドリクエストレジスタのメッセージ番号は評価されません。IF2 メッセージ制御レジスタ IF2MCTRn の NEWDAT ビットおよび MSGLST ビットはその機能を保持し, DLC3～DLC0は受信したDLCを示し, ほかの制御ビットの読出しには"0"が返されます。

CAN_TX端子のソフトウェア制御

CAN 送信端子 CAN_TX は 4 種類の出力機能に使用することができます。デフォルトの機能であるシリアルデータ出力のほかに , CAN コアでのビットタイミングを監視するために CAN サンプルポイント信号を出力したり , 一定のドミナント値またはレセッシブ値を出力することもできます。後の 2 つの機能と読取り可能な CAN_RX 端子を組み合せて使用し , CAN バスの物理層をチェックすることができます。

テストレジスタ TESTRn の TX1 ビットおよび TX0 の設定により CAN_TX 端子の出力モードを選択します(「22.5

CAN プロトコル関連のレジスタ」参照)。

CAN_TX 端子の 3 つのテスト機能は CAN プロトコルのすべての機能と干渉します。CAN メッセージの転送時またはテストモード , ループバックモード , サイレントモード , ベーシックモードのいずれかを選択した場合は , CAN_TX をデフォルトの機能にしておく必要があります。

22.3 各レジスタの説明

CANレジスタの一覧を示し, それぞれのレジスタを詳細に説明します。

プログラマモデル

CAN モジュールは 256 バイト (64 ワード) のアドレス空間を割り当てます。CAN の各レジスタには CPU からのバイトアクセスおよびワードアクセスが可能です。

CPU によるメッセージ RAM へのアクセスの制御には 2 組のインタフェースレジスタ (IF1 および IF2) が使用されます。インタフェースレジスタは RAM からまたは RAM に転送されるデータのバッファとして使用され, メッセージの受信 / 送信と CPU アクセスの競合を防止しています。

デバイス上に複数のCANモジュールが存在する場合は，各モジュールは256バイトの固定オフセットでアドレス空間上の連続領域に配置されます。デバイス上の各CANモジュールのベースアドレスについてはデータシートを参照してください。

Table 22-1. CANレジスター覧

オフセット	バイトレジスタ名	ワードレジスタ名	説明	初期値
CANnのベースアドレス + 0x0	CTRLRLn	CTRLRn	CAN n - 制御レジスタ	000X0001
CANnのベースアドレス + 0x1	CTRLRHn		CAN n - 制御レジスタ (予約)	XXXXXXXX
CANnのベースアドレス + 0x2	STATRLn	STATRn	CAN n - ステータス レジスタ	00000000
CANnのベースアドレス + 0x3	STATRHn		CAN n - ステータス レジスタ(予約)	XXXXXXXX
CANnのベースアドレス + 0x4	ERRCNTLn	ERRCNTn	CAN n - エラーカウンタ (送信)	00000000
CANnのベースアドレス + 0x5	ERRCNTHn		CAN n - エラーカウンタ (受信)	00000000
CANnのベースアドレス + 0x6	BTRLn	BTRn	CAN n - ビット タイミングレジスタ	00000001
CANnのベースアドレス + 0x7	BTRHn		CAN n - ビット タイミングレジスタ	X0100011
CANnのベースアドレス + 0x8	INTRLn	INTRn	CAN n - 割込みレジスタ	00000000
CANnのベースアドレス + 0x9	INTRHn		CAN n - 割込みレジスタ	00000000
CANnのベースアドレス + 0xA	TESTRLn	TESTRn	CAN n - テストレジスタ	000000XX
CANnのベースアドレス + 0xB	TESTRHn		CAN n - テストレジスタ (予約)	XXXXXXXX
CANnのベースアドレス + 0xC	BRPERLn	BRPERn	CAN n - BRP拡張 レジスタ	XXXX0000
CANnのベースアドレス + 0xD	BRPERHn		CAN n - BRP拡張 レジスタ(予約)	XXXXXXXX
IF1レジスタ				
CANnのベースアドレス + 0x10	IF1CREQLn	IF1CREQn	CAN n - IF1コマンド リクエストレジスタ	00000001
CANnのベースアドレス + 0x11	IF1CREQHn		CAN n - IF1コマンド リクエストレジスタ	0XXXXXXX

Table 22-1. CANレジスター一覧

オフセット	バイトレジスタ名	ワードレジスタ名	説明	初期値
CANnのベースアドレス + 0x12	IF1CMSKLn	IF1CMSKn	CAN n - IF1コマンドマスクレジスタ	00000000
CANnのベースアドレス + 0x13	IF1CMSKHn		CAN n - IF1コマンドマスクレジスタ(予約)	XXXXXXXX
CANnのベースアドレス + 0x14	IF1MSK1Ln	IF1MSK1n	CAN n - IF1マスクレジスタ	11111111
CANnのベースアドレス + 0x15	IF1MSK1Hn		CAN n - IF1マスクレジスタ	11111111
CANnのベースアドレス + 0x16	IF1MSK2Ln	IF1MSK2n	CAN n - IF1マスクレジスタ	11111111
CANnのベースアドレス + 0x17	IF1MSK2Hn		CAN n - IF1マスクレジスタ	11X11111
CANnのベースアドレス + 0x18	IF1ARB1Ln	IF1ARB1n	CAN n - IF1アービトレーションレジスタ	00000000
CANnのベースアドレス + 0x19	IF1ARB1Hn		CAN n - IF1アービトレーションレジスタ	00000000
CANnのベースアドレス + 0x1A	IF1ARB2Ln	IF1ARB2n	CAN n - IF1アービトレーションレジスタ	00000000
CANnのベースアドレス + 0x1B	IF1ARB2Hn		CAN n - IF1アービトレーションレジスタ	00000000
CANnのベースアドレス + 0x1C	IF1MCTRLn	IF1MCTRn	CAN n - IF1メッセージ制御レジスタ	0XXX0000
CANnのベースアドレス + 0x1D	IF1MCTRHn		CAN n - IF1メッセージ制御レジスタ	00000000
CANnのベースアドレス + 0x1E	IF1DTA1Ln	IF1DTA1n	CAN n - IF1データA1	00000000
CANnのベースアドレス + 0x1F	IF1DTA1Hn		CAN n - IF1データA1	00000000
CANnのベースアドレス + 0x20	IF1DTA2Ln	IF1DTA2n	CAN n - IF1データA2	00000000
CANnのベースアドレス + 0x21	IF1DTA2Hn		CAN n - IF1データA2	00000000
CANnのベースアドレス + 0x22	IF1DTB1Ln	IF1DTB1n	CAN n - IF1データB1	00000000
CANnのベースアドレス + 0x23	IF1DTB1Hn		CAN n - IF1データB1	00000000
CANnのベースアドレス + 0x24	IF1DTB2Ln	IF1DTB2n	CAN n - IF1データB2	00000000
CANnのベースアドレス + 0x25	IF1DTB2Hn		CAN n - IF1データB2	00000000
IF2レジスタ				
CANnのベースアドレス + 0x40	IF2CREQLn	IF2CREQn	CAN n - IF2コマンドリクエストレジスタ	00000001
CANnのベースアドレス + 0x41	IF2CREQHn		CAN n - IF2コマンドリクエストレジスタ	0XXXXXXX
CANnのベースアドレス + 0x42	IF2CMSKLn	IF2CMSKn	CAN n - IF2コマンドマスクレジスタ	00000000
CANnのベースアドレス + 0x43	IF2CMSKHn		CAN n - IF2コマンドマスクレジスタ(予約)	XXXXXXXX

Table 22-1. CANレジスター覧

オフセット	バイトレジスタ名	ワードレジスタ名	説明	初期値
CANnのベースアドレス + 0x44	IF2MSK1Ln	IF2MSK1n	CAN n - IF2マスクレジスタ	11111111
CANnのベースアドレス + 0x45	IF2MSK1Hn		CAN n - IF2マスクレジスタ	11111111
CANnのベースアドレス + 0x46	IF2MSK2Ln	IF2MSK2n	CAN n - IF2マスクレジスタ	11111111
CANnのベースアドレス + 0x47	IF2MSK2Hn		CAN n - IF2マスクレジスタ	11X11111
CANnのベースアドレス + 0x48	IF2ARB1Ln	IF2ARB1n	CAN n - IF2アービトレーションレジスタ	00000000
CANnのベースアドレス + 0x49	IF2ARB1Hn		CAN n - IF2アービトレーションレジスタ	00000000
CANnのベースアドレス + 0x4A	IF2ARB2Ln	IF2ARB2n	CAN n - IF2アービトレーションレジスタ	00000000
CANnのベースアドレス + 0x4B	IF2ARB2Hn		CAN n - IF2アービトレーションレジスタ	00000000
CANnのベースアドレス + 0x4C	IF2MCTRLn	IF2MCTRn	CAN n - IF2メッセージ制御レジスタ	0XXX0000
CANnのベースアドレス + 0x4D	IF2MCTRHn		CAN n - IF2メッセージ制御レジスタ	00000000
CANn のベースアドレス + 0x4E	IF2DTA1Ln	IF2DTA1n	CAN n - IF2データA1	00000000
CANnのベースアドレス + 0x4F	IF2DTA1Hn		CAN n - IF2データA1	00000000
CANnのベースアドレス + 0x50	IF2DTA2Ln	IF2DTA2n	CAN n - IF2データA2	00000000
CANnのベースアドレス + 0x51	IF2DTA2Hn		CAN n - IF2データA2	00000000
CANnのベースアドレス + 0x52	IF2DTB1Ln	IF2DTB1n	CAN n - IF2データB1	00000000
CANnのベースアドレス + 0x53	IF2DTB1Hn		CAN n - IF2データB1	00000000
CANnのベースアドレス + 0x54	IF2DTB2Ln	IF2DTB2n	CAN n - IF2データB2	00000000
CANnのベースアドレス + 0x55	IF2DTB2Hn		CAN n - IF2データB2	00000000
CANnのベースアドレス + 0x80	TREQR1Ln	TREQR1n	CAN n - 送信リクエストレジスタ	00000000
CANnのベースアドレス + 0x81	TREQR1Hn		CAN n - 送信リクエストレジスタ	00000000
CANnのベースアドレス + 0x82	TREQR2Ln	TREQR2n	CAN n - 送信リクエストレジスタ	00000000
CANnのベースアドレス + 0x83	TREQR2Hn		CAN n - 送信リクエストレジスタ	00000000
CANnのベースアドレス + 0x90	NEWDT1Ln	NEWDT1n	CAN n - 新規データレジスタ	00000000
CANnのベースアドレス + 0x91	NEWDT1Hn		CAN n - 新規データレジスタ	00000000

Table 22-1. CANレジスター一覧

オフセット	バイトレジスタ名	ワードレジスタ名	説明	初期値
CANnのベースアドレス + 0x92	NEWDT2Ln	NEWDT2n	CAN n - 新規データレジスタ	00000000
CANnのベースアドレス + 0x93	NEWDT2Hn		CAN n - 新規データレジスタ	00000000
CANn のベースアドレス + 0xA0	INTPND1Ln	INTPND1n	CAN n - 割込み保留レジスタ	00000000
CANnのベースアドレス + 0xA1	INTPND1Hn		CAN n - 割込み保留レジスタ	00000000
CANnのベースアドレス + 0xA2	INTPND2Ln	INTPND2n	CAN n - 割込み保留レジスタ	00000000
CANnのベースアドレス + 0xA3	INTPND2Hn		CAN n - 割込み保留レジスタ	00000000
CANn のベースアドレス + 0xB0	MSGVAL1Ln	MSGVAL1n	CAN n - メッセージ有効レジスタ	00000000
CANnのベースアドレス + 0xB1	MSGVAL1Hn		CAN n - メッセージ有効レジスタ	00000000
CANn のベースアドレス + 0xB2	MSGVAL2Ln	MSGVAL2n	CAN n - メッセージ有効レジスタ	00000000
CANnのベースアドレス + 0xB3	MSGVAL2Hn		CAN n - メッセージ有効レジスタ	00000000
CANnのベースアドレス + 0xCE	COERn		CAN n - 出力許可レジスタ	XXXXXXX0

ハードウェアリセットの説明

ハードウェアリセット後の CAN のレジスタには Table 22-1 の初期値が保持されます。

また , バスオフ状態もリセットされ CAN_TX の出力はレセッシブ (HIGH) にセットされます。CAN 制御レジスタ CTRLRLn の値は 0001_H (INIT=1) であり , ソフトウェアによる初期化が可能です。CPU が INIT を "0" にリセットするまで CAN は CAN バスに影響を与えません。

ハードウェアリセットはメッセージ RAM に格納されているデータには影響を与えません。電源投入後のメッセージ RAM の内容は不定です。

22.4 CANデバイス関連のレジスタ

CANバスコントローラのホストを行うデバイスに関連したレジスタです。

22.4.1 CAN出力許可レジスタ(COERn)

CAN出力許可レジスタ(COERn)は，デバイスの端子をCANバスコントローラのTX0～TX2端子として使用するかどうかの選択に使用します。

CAN出力許可レジスタ(COERn)

Figure 22-4. CAN出力許可レジスタ(COERn)の構成



CAN出力許可レジスタ(COERn)の機能

[bit7-bit1] 予約 予約ビット
常に"0"を書き込みます。リード値は不定です。リードモディファイライト系命令は無効です。

[bit0] OE 出力許可
0 CAN TXを禁止する。端子を汎用ポートまたはほかの周辺機能用として使用できる。
1 CAN TXを許可する。

22.5 CANプロトコル関連のレジスタ

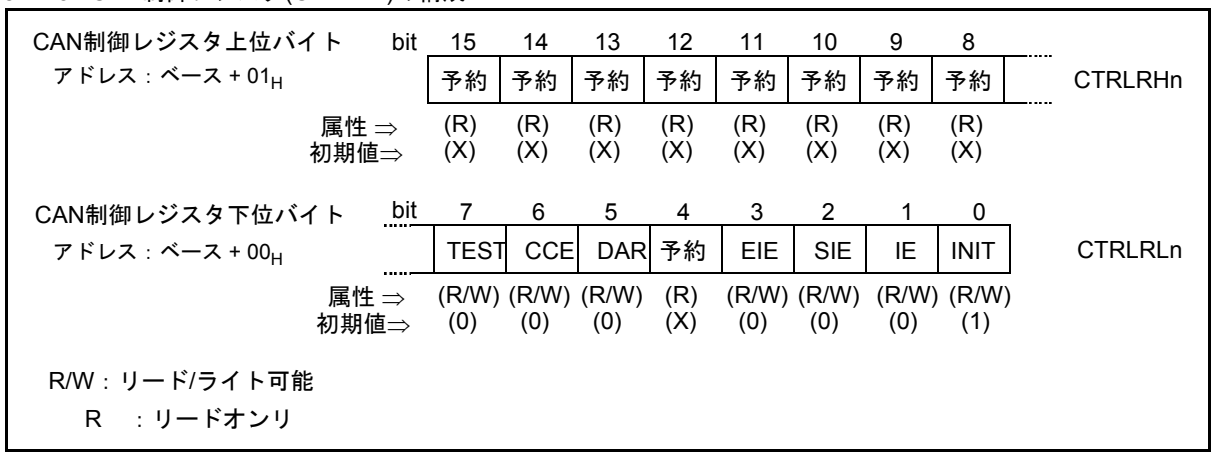
CANコアのCANプロトコルコントローラ関連のレジスタです。動作モードおよびCANビットタイミングの設定とステータス情報の取得に使用します。

22.5.1 CAN制御レジスタ(CTRLRn)

CAN制御レジスタ(CTRLRn)は, CANコントローラの基本的な動作モードの制御に使用します。

CAN制御レジスタ(CTRLRn)

Figure 22-5. CAN制御レジスタ(CTRLRn)の構成



CAN制御レジスタ(CTRLRn)の機能

[bit15 - bit8]	予約	予約ビット 常に"0"を書き込みます。リード値は不定です。リードモディファイライト系命令は無効です。
[bit7]	TEST	テストモード許可
	0	通常動作
	1	テストモード
[bit6]	CCE	設定変更許可
	0	CPUによるビットタイミングレジスタへのライトアクセスを行わない。
	1	CPUによるビットタイミングレジスタへのライトアクセスを行う(INIT=1のとき)。
[bit5]	DAR	自動再送信無効
	0	問題発生メッセージの自動再送信を許可する。
	1	自動再送信を禁止する。

＜注意事項＞

C_CAN が DAR モード (自動再送禁止モード) で実行するように構成されている場合で , ホスト側が複数メッセージの同時送信を要求した場合 , そのうちの 2 つのメッセージだけが送信されます。要求されたほかのすべての送信メッセージに対しては , TXRQST ビットがリセットされますが , 送信は開始されず , NEWDAT と INTPND はそのままの状態です。送信される 2 つのメッセージに関しては , TXRQST ビットと NEWDAT ビットがリセットされ , TXIE によって許可されている場合 , INTPND がセットされます。

[bit4]	予約	予約ビット 常に"0"を書き込みます。リード値は不定です。リードモディファイライト系命令は無効です。
[bit3]	EIE	エラー割込み許可
	0	禁止－エラーステータス割込みを生成しない。
	1	許可－ステータスレジスタのBOffビットまたはEWarnビットが変化すると割込みを生成する。
[bit2]	SIE	ステータス変化割込み許可
	0	禁止－ステータス変化割込みを生成しない。
	1	許可－メッセージの転送が正常に完了したかCANバスエラーが検出されたときに割込みを生成する。
[bit1]	IE	モジュール割込み許可
	0	禁止－モジュール割込みを常に禁止する。
	1	許可－割込みを生成する。保留されていたすべての割込みが処理されるまでリクエストが残る。
[bit0]	INIT	初期化
	0	通常動作
	1	初期化を開始する。

＜注意事項＞

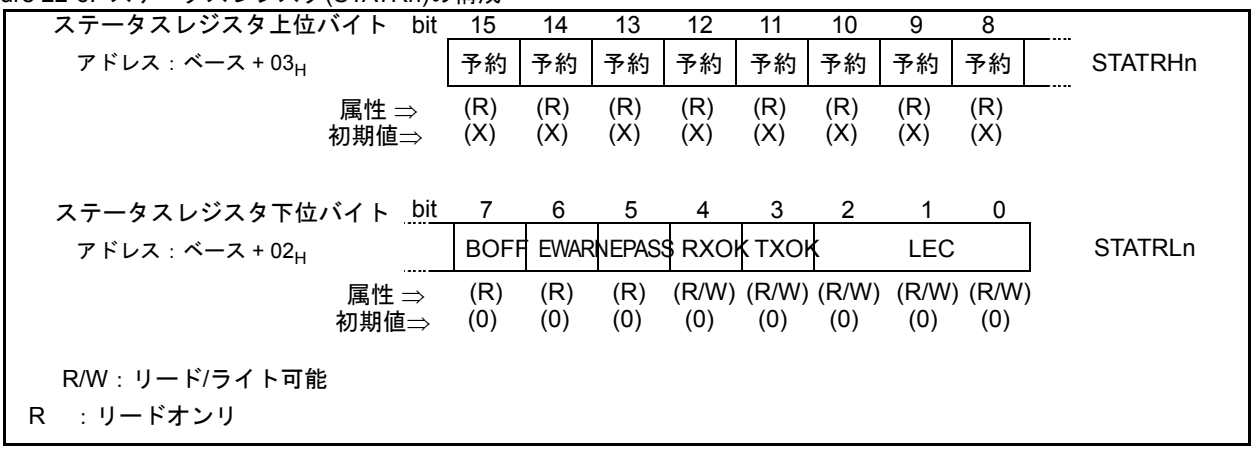
- INIT をセットまたはリセットすることによりバスオフ回復シーケンス (「CAN Specification Rev. 2.0」を参照) を省略することはできません。デバイスがバスオフ状態に移行した場合は , それに合わせて INIT がセットされ , すべてのバスアクティビティが停止します。CPU によって INIT をクリアすると , デバイスはバスアイドルが 129 回発生するのを待って (129 × 11 のレセッシブビットの連続), 通常の操作に復帰します。バスオフ回復シーケンスが終わると , エラー制御カウンタがリセットされます。
- INIT のリセット後の待ち時間中に , 11 のレセッシブビットの連続シーケンスが検出されるたびにステータスレジスタに Bit0Error コードが書き込まれます。これによって , CPU は , CAN バスがドミナントレベルにスタックしていたり , ずっと妨害され続けていないかどうかをチェックすることが可能になり , バスオフ回復シーケンスの進行状況を監視することができます。

22.5.2 ステータスレジスタ(STATRn)

CANコントローラのステータスを表すレジスタです。

ステータスレジスタ(STATRn)

Figure 22-6. ステータスレジスタ(STATRn)の構成



ステータスレジスタ(STATRn)の機能

[bit15 - bit8] 予約 予約ビット

常に "0" を書き込みます。リード値は不定です。
リードモディファイライト系命令は無効です。

[bit7] BOFF バスオフステータス

0 CANモジュールはバスオフ状態ではない。

1 CANモジュールはバスオフ状態になっている。

[bit6] EWARN 警告ステータス

0 両方のエラーカウンタはエラー警告限界の96以下。

1 EML内の少なくとも一方のエラーカウンタがエラー警告限界の96に達した。

[bit5] EPASS エラーパッシブ

0 CANコアはエラーアクティブ。

1 CANコアはCANの仕様で定義されているエラーパッシブ状態。

[bit4] RXOK メッセージの正常受信

0 このビットが前回CPUによってリセットされてから正常に受信したメッセージは存在しない。このビットがCANコアによってリセットされることはない。

1 このビットが前回CPUによって(0に)リセットされてから1つのメッセージが正常に受信されている(アクセプタンスの結果には無関係)。

[bit3] TXOK メッセージの正常送信

0 このビットが前回CPUによってリセットされてから正常に送信されたメッセージは存在しない。このビットがCANコアによってリセットされることはない。

1 このビットが前回CPUによってリセットされてから1つのメッセージが正常に送信されている(エラーが発生せず少なくとも1つの他のノードからアクノリッジが返されている)。

[bit2 - bit0]	LEC	最後のエラーコード(CANバスで発生した最後のエラーのタイプ)	
	0	エラーなし	
	1	スタッフエラー	受信したメッセージ内に同じビットが5つ以上連続して発生(これは禁止されたパターンである)。
	2	フォームエラー	受信したフレームの固定フォーマットの部分が不正。
	3	AckError	このCANコアが送信したメッセージに対して他のノードからアクノリッジが返されない。
	4	Bit1Error	メッセージ(アービトレーションフィールドを除く)の送信中にデバイスがレセッシブレベル(ビットの論理値"1")を送信しようとしたが, 検出されたバス値はドミナント。
	5	Bit0Error	メッセージ(またはアクノリッジビット, アクティブエラーフラグ, オーバロードフラグ)の送信中にデバイスがドミナントレベル(データまたはIDビットの論理値"0")を送信しようとしたが, 検出されたバス値はレセッシブだった。バスオフ回復中に, 11のレセッシブビットの連続シーケンスが検出されるたびにこの状態がセットされる。これによってCPUはバスオフ回復シーケンスの進行状況を監視することができる(バスが, ドミナントレベルにスタックしていたりずっと妨害され続けていたりしていないことを示します)。
	6	CRCError	メッセージ受信時のCRCチェックサムが不正です。 受信したCRCの値が, 受信データから計算したCRCの値と異なる。
	7	未使用	LECの値が7のときは, この値をCPUがLECに書き込んでから検出されたCANバスイベントが存在しない。

LEC フィールドには CAN バスで最後に発生したエラーのタイプを示すコードが保持されます。メッセージの転送 (送信または受信) でエラーが発生しなかった場合は , このフィールドは "0" にリセットされます。LEC が更新されているかどうかの確認のために , CPU によって未使用コード 7 を書いておくことがあります。

ステータス割込み

CAN 制御レジスタの対応する許可ビットがセットされていれば , BOFF ビットと EWARN ビット (エラー割込み) または RXOK ビット , TXOK ビット , および LEC ビット (ステータス変化割込み) によってステータス割込みが生成されます。EPASS ビットの変化または RXOK ビット , TXOK ビット , または LEC ビットへの書込みによってステータス割込みが生成されることはありません。

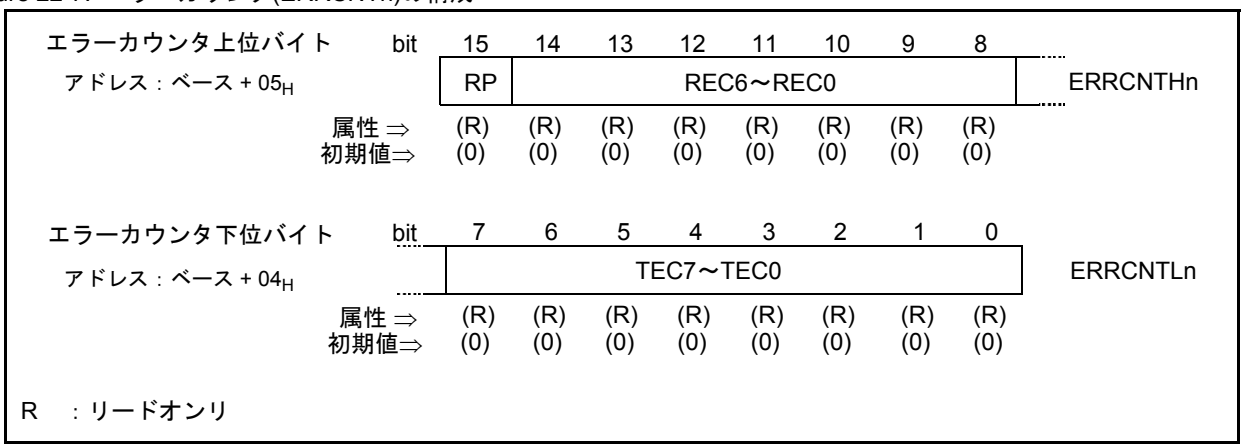
ステータスレジスタをリードアクセスすると , 割込みレジスタのステータス割込み値 (8000_H) がクリアされます (保留されていた場合)。

22.5.3 エラーカウンタ(ERRCNTn)

エラーカウンタレジスタ(ERRCNTn)には受信エラーカウンタ(TEC)および送信エラーカウンタ(REC)および受信エラーパッシブ(RP)が示されます。

エラーカウンタ(ERRCNTn)

Figure 22-7. エラーカウンタ(ERRCNTn)の構成



エラーカウンタ(ERRCNTn)の機能

[bit15] RP 受信エラーパッシブ

0 受信エラーカウンタがエラーパッシブレベル未満であることを示します。

1 受信エラーカウンタがCAN仕様で定義されているエラーパッシブレベルに達したことを示します。

[bit14 ～ bit8] REC6～REC0 受信エラーカウンタ

受信エラーカウンタの値。受信エラーカウンタ値の範囲は0～127です。

[bit7 ～ bit0] TEC7～TEC0 送信エラーカウンタ

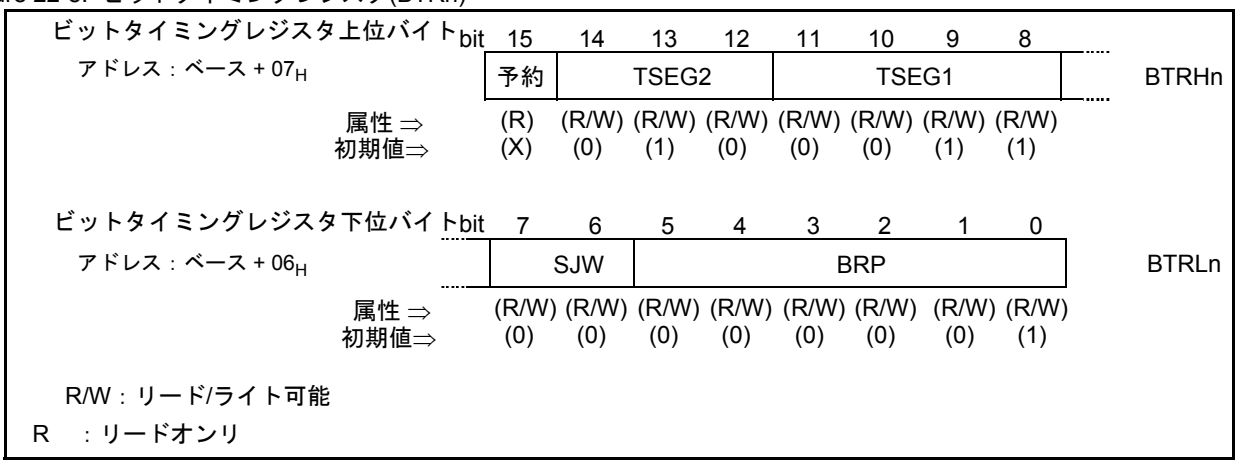
送信エラーカウンタの値。送信エラーカウンタ値の範囲は0～255です。

22.5.4 ビットタイミングレジスタ(BTRn)

ビットタイミングレジスタ(BTRn)はCANバスコントローラビットタイミングの制御を設定します。

ビットタイミングレジスタ(BTRn)

Figure 22-8. ビットタイミングレジスタ(BTRn)



ビットタイミングレジスタ(BTRn)の機能

[bit15] 予約

予約ビット

常に "0" を書き込みます。リード値は不定です。
リードモディファイライト系命令は無効です。

[bit114 ～ bit12] TSEG2

0x0～0x7

サンプルポイントの後のタイムセグメント

TSEG2の有効設定値は0～7です。この値に1を加えた値(TSEG2+1)がタイムセグメントに設定されます。

[bit11 ～ bit8] TSEG1

0x01～0x0F

サンプルポイントの前のタイムセグメント

TSEG1の有効設定値は1 ～15です。この値に1を加えた値(TSEG1+1)がタイムセグメントに設定されます。

[bit7 ～ bit 6] SJW

0x0～0x3

再同期化ジャンプ幅設定ビット

有効な設定値は0～3です。この値に1を加えた値(SJW+1)が再同期ジャンプ幅に設定されます。

[bit5 ～ bit0] BRP

0x00～0x3F

ボーレートプリスケーラ設定ビット

基本単位時間を生成するための,周辺クロックCLKP2の周波数の分周比。ビットタイムはこの基本単位時間の倍数で構成されます。ボーレートプリスケーラ設定ビットの有効設定値は0～ 63です。この値に1を加えた値(BRP+1)がボーレートプリスケーラに設定されます。

＜注意事項＞

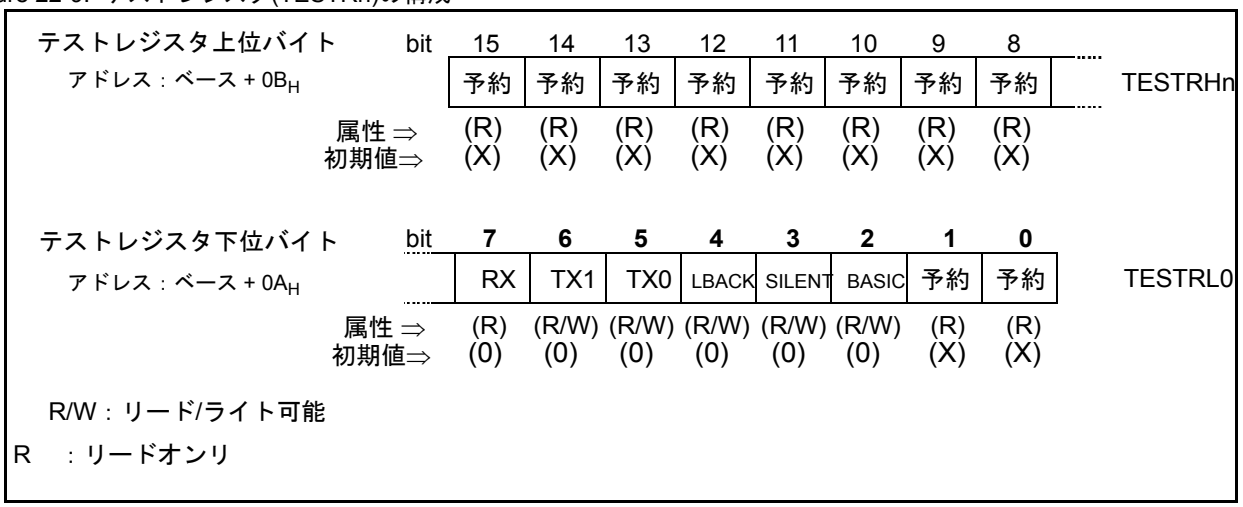
周辺クロック CLKP2 の周波数が 8 MHz のとき , 2301_H のリセット値に対応する CAN のビットレートは 500 kbps です。CAN 制御レジスタの CCE ビットと INIT ビットをセットした場合にのみレジスタは書込み可能になります。

22.5.5 テストレジスタ(TESTRn)

テストレジスタ(TESTRn)を使用してCANバスシステムをテストすることができます。

テストレジスタ(TESTRn)

Figure 22-9. テストレジスタ(TESTRn)の構成



テストレジスタ(TESTRn)の機能

[bit15 ～bit8]	予約	予約ビット 常に"0"を書き込みます。リード値は不定です。 リードモディファイライト系命令は無効です。
[bit7]	RX	CAN_RX端子の監視
	0	CANバスはドミナント(CAN_RX = 0)であることを示します。
	1	CANバスはリセッシブ(CAN_RX = 1)であることを示します。
[bit6, bit5]	TX1, TX0	CAN_TX端子の制御
	00	リセット値。CAN_TXをCANコアで制御します。
	01	サンプルポイントをCAN_TX端子で監視可能にします。
	10	CAN_TX端子でドミナント("0")値を駆動します。
	11	CAN_TX端子でリセッシブ("1")値を駆動します。
[bit4]	LBACK	ループバックモード
	0	ループバックモードを禁止します。
	1	ループバックモードを許可します。
[bit3]	SILENT	サイレントモード
	0	通常動作(サイレントモードを禁止します)
	1	モジュールをサイレントモードにします。
[bit2]	BASIC	ベーシックモード
	0	ベーシックモードを禁止します。
	1	IF1レジスタをTxバッファとして, IF2レジスタをRxバッファとして使用します。

[bit1～bit0]	予約	予約ビット 常に "0" を書き込みます。リード値は不定です。 リードモディファイライト系命令は無効です。

＜注意事項＞

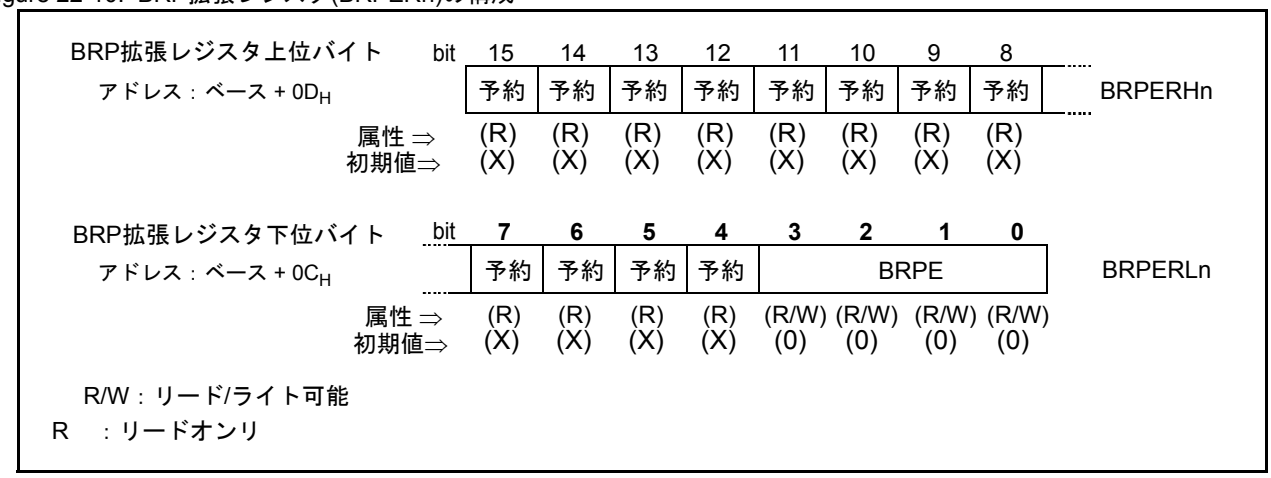
CAN 制御レジスタ (CTRLRn) の TEST ビットをセットすることによりテストレジスタへのライトアクセスが可能になります。別のテスト機能を組み合せることができますが , TX1, TX0 ≠ 00 の設定を使用するとメッセージの転送に問題が発生します。

22.5.6 BRP拡張レジスタ(BRPERn)

BRP拡張レジスタ(BRPERn)にはボーレートジェネレータプリスケーラ用の拡張設定ビットが格納されています。

BRP拡張レジスタ(BRPERn)

Figure 22-10. BRP拡張レジスタ(BRPERn)の構成



BRP拡張レジスタ(BRPERn)の機能

[bit15～bit4]	予約	予約ビット 常に "0" を書き込みます。リード値は不定です。 リードモディファイライト系命令は無効です。
[bit3～bit0]	BRPE 0x00～0x0F	ボーレートプリスケーラ拡張 CANビットタイミングレジスタ(BTR)のBRPビットとBRPEビットの組み合わせによりボーレートプリスケーラの値を最大1023に拡張することができます。BRPEビットとBRPビットの組み合わせに1を加えた値[{BRPE(MSB:4ビット), BRP(LSB:6ビット)}+1]がCANコントローラのプリスケーラに設定されます。

22.6 メッセージインタフェースレジスタセット

メッセージRAMへのCPUアクセスとCANバスコントローラからのアクセスの競合を避けるために,CPUからメッセージRAMへのアクセスを制御するために2組のメッセージインタフェースレジスタが用意されています。

メッセージインタフェースレジスタセットの概要

CPU からのメッセージ RAM へのアクセスの制御に使用する 2 セットのインタフェースレジスタが用意されています。インタフェースレジスタは , メッセージ RAM への CPU のアクセスと CAN メッセージの送受信の競合を , 転送するデータをバッファリングすることで防止します。メッセージ RAM と IFx メッセージバッファレジスタ間で , 完全なメッセージオブジェクト (「22.7 メッセージメモリ内のメッセージオブジェクト」参照) またはメッセージオブジェクトの一部を 1 度に転送することができます。

2 つのインタフェースレジスタセットの機能は同じです (BASIC テストモードを除く)。レジスタの 1 セットをメッセージ RAM へのデータ伝送に , 他のレジスタセットをメッセージ RAM からのデータ伝送に使用することができ , 両方のプロセスに対して相互に割込みを行うことができます。Table 22-2 に 2 つのインタフェースレジスタセットの概要を示します。

インタフェースレジスタの各セットは, 専用のコマンドレジスタによって制御されるメッセージバッファレジスタで構成されています。コマンドマスクレジスタは, データ転送の方向とメッセージオブジェクトのどの部分を転送するのかを指定します。コマンドリクエストレジスタは,メッセージRAM内のメッセージオブジェクトをターゲットまたは転送のソースとして選択し, コマンドマスクレジスタで指定したアクションを開始するために使用します。

Table 22-2. IF1およびIF2メッセージインタフェースレジスタセット

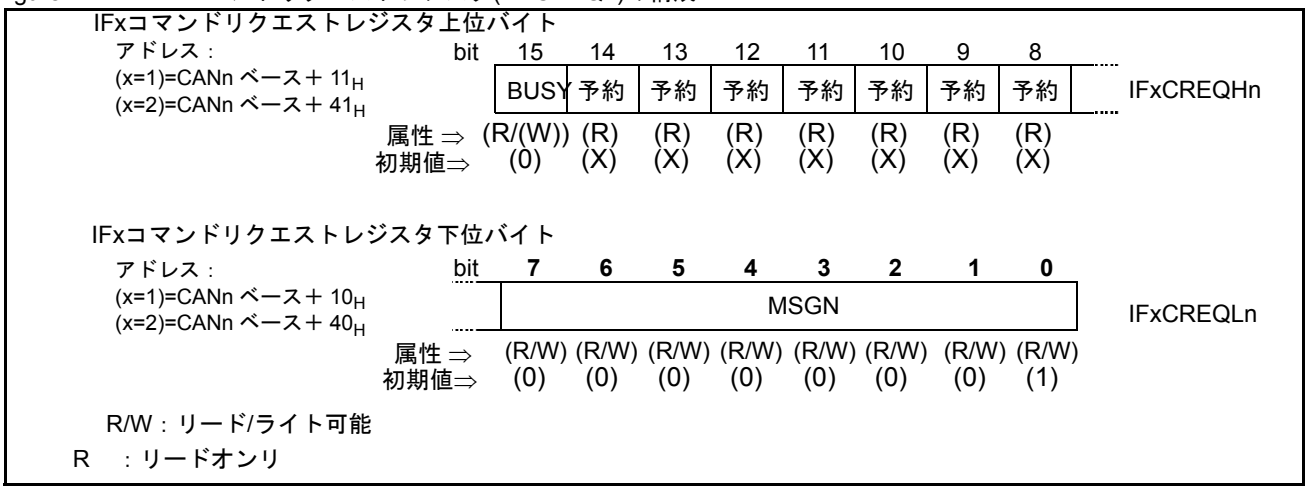
アドレス	IF1レジスタセット	アドレス	IF2レジスタセット
CANベース + 0x10	IF1コマンドリクエスト(IF1CREQn)	CANベース + 0x40	IF2コマンドリクエスト(IF2CREQn)
CANベース + 0x12	IF1コマンドマスク(IF1CMSKn)	CANベース + 0x42	IF2コマンドマスク(IF2CMSKn)
CANベース + 0x14	IF1マスク1 (IF1MSK1n)	CANベース + 0x44	IF2マスク1 (IF2MSK1n)
CANベース + 0x16	IF1マスク2 (IF1MSK2n)	CANベース + 0x46	IF2マスク2 (IF2MSK2n)
CANベース + 0x18	IF1アービトレーション1 (IF1ARB1n)	CANベース + 0x48	IF2アービトレーション1 (IF2ARB1n)
CANベース + 0x1A	IF1アービトレーション2 (IF1ARB2n)	CANベース + 0x4A	IF2アービトレーション2 (IF2ARB2n)
CANベース + 0x1C	IF1メッセージ制御(IF1MCTRn)	CANベース + 0x4C	IF2メッセージ制御(IF2MCTRn)
CANベース + 0x1E	IF1データA1 (IF1DTA1n)	CANベース + 0x4E	IF2データA1 (IF2DTA1n)
CANベース + 0x20	IF1データA2 (IF1DTA2n)	CANベース + 0x50	IF2データA2 (IF2DTA2n)
CANベース + 0x22	IF1データB1 (IF1DTB1n)	CANベース + 0x52	IF2データB1 (IF2DTB1n)
CANベース + 0x24	IF1データB2 (IF1DTB2n)	CANベース + 0x54	IF2データB2 (IF2DTB2n)

22.6.1 IFxコマンドリクエストレジスタ(IFxCREQn)

CPUによってコマンドリクエストレジスタ(IFxCREQn)にメッセージ番号が書き込まれると， 直ちにメッセージの転送が開始されます。このライト操作で， CPUには転送中であることが通知されます。転送中にCPUからのCANアクセスが発生した場合は，転送が完了するまでアクセスが延期されます。CAN_CLKで3～6サイクル後にインタフェースレジスタとメッセージRAM間の転送が完了し，保留中のCPUアクセスが実行されます。

IFxコマンドリクエストレジスタ(IFxCREQn)

Figure 22-11. IFxコマンドリクエストレジスタ(IFxCREQn)の構成



IFxコマンドリクエストレジスタ(IFxCREQn)の機能

[bit15]	BUSY	ビジーフラグ
	0	リード/ライトアクションが完了したときは"0"にリセットします。
	1	IFxコマンドリクエストレジスタへの書込み時に"1"にセットします。
[bit14～bit8]	予約	予約ビット 常に "0" を書き込みます。リード値は不定です。 リードモディファイライト系命令は無効です。
[bit5～bit0]	MSGN	メッセージ番号(32メッセージバッファCANの場合，デバイスのメッセージバッファ数についてはデータシート参照)。
	0x00	設定禁止(設定した場合は20_Hとして処理されます)
	0x01～0x20	有効なメッセージ番号。メッセージRAM内のメッセージオブジェクトをデータ転送用に選択。
	0x21～0xFF	設定禁止(設定した場合は01_H～$1F_H$として処理されます)
[bit5～bit0]	MSGN	メッセージ番号(64メッセージバッファCANの場合，デバイスのメッセージバッファ数についてはデータシート参照)。
	0x00	設定禁止(設定した場合は40_Hとして処理されます)
	0x01～0x40	有効なメッセージ番号。メッセージRAM内のメッセージオブジェクトをデータ転送用に選択。
	0x41～0xFF	設定禁止(設定した場合は01_H～$3F_H$として処理されます)
[bit7～bit0]	MSGN	メッセージ番号(128メッセージバッファCANの場合，デバイスのメッセージバッファ数についてはデータシート参照)。

0x00	設定禁止(設定した場合は80_Hとして処理されます)
0x01～0x80	有効なメッセージ番号。メッセージRAM内のメッセージオブジェクトをデータ転送用に選択。
0x81～0xFF	設定禁止(設定した場合は01_H～$7F_H$として処理されます)

＜注意事項＞

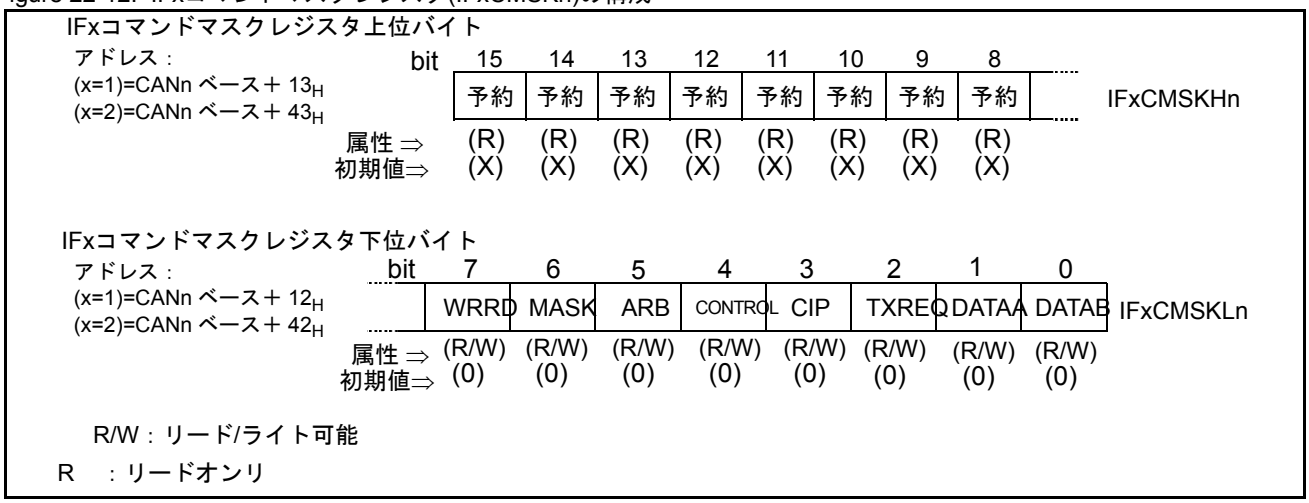
- T ビジーフラグが使用できるのは BASIC モードだけです (「22.2 機能の説明 - ベーシックモード」参照)。メッセージ RAM を使用するとき (BASIC=0), リード / ライトアクセスはハードウェアインタフェースによって制御され , このフラグへのリードアクセスには常に 0 が返されます。
- 設定禁止のメッセージ番号をコマンドリクエストレジスタに書き込んだ場合 , メッセージ番号は有効な値に変換され , そのメッセージオブジェクトが転送されます。

22.6.2 IFxコマンドマスクレジスタ(IFxCMSKn)

IFxコマンドマスクレジスタ(IFxCMSKn)の制御ビットは, 転送方向を指定し, どのIFxメッセージバッファレジスタをデータ転送のソースまたはターゲットとして使用するかを選択します。

IFxコマンドマスクレジスタ(IFxCMSKn)

Figure 22-12. IFxコマンドマスクレジスタ(IFxCMSKn)の構成



IFxコマンドマスクレジスタ(IFxCMSKn)の機能

[bit15～bit8]	予約	予約ビット 常に "0" を書き込みます。リード値は不定です。 リードモディファイライト系命令は無効です。
[bit7]	WRRD	ライト/リード
	0	リード : コマンドリクエストレジスタによって指定されたメッセージオブジェクトから選択したメッセージバッファレジスタにデータを転送します。
	1	ライト : 選択したメッセージバッファレジスタからコマンドリクエストレジスタによって指定されたメッセージオブジェクトにデータを転送します。

IFx コマンドマスクレジスタのほかのビットの機能は転送方向によって異なります。

方向 = ライト

[bit6]	MASK	マスクビットのアクセス
	0	ID マスク + MDir + MXtd をメッセージオブジェクトに転送しません。
	1	IDマスク + MDir + MXtdをメッセージオブジェクトに転送します。
[bit5]	ARB	アービトレーションビットのアクセス
	0	ID + Dir + Xtd + MsgVal をメッセージオブジェクトに転送しません。
	1	ID + Dir + Xtd + MsgValをメッセージオブジェクトに転送します。
[bit4]	CONTROL	制御ビットのアクセス
	0	制御ビットをメッセージオブジェクトに転送しません。
	1	制御ビットをメッセージオブジェクトに転送します。

[bit3]	CIP	割込み保留ビットをクリアします。 メッセージオブジェクトへの書込み時は本ビットは無視されます。
[bit2]	TXREQ	送信リクエストビットのアクセス
	0	TXRQSTビットを更新しません。
	1	TXRQSTビットをセットします。
[bit1]	DATAA	データバイト0～3のアクセス
	0	データバイト0～3をメッセージオブジェクトに転送しません。
	1	データバイト0～3をメッセージオブジェクトに転送します。
[bit0]	DATAB	データバイト4～7のアクセス
	0	データバイト4～7をメッセージオブジェクトに転送しません。
	1	データバイト4～7をメッセージオブジェクトに転送します。

方向 = リード

[bit6]	MASK	マスクビットのアクセス
	0	ID マスク + MDir + MXtd をIFx メッセージバッファレジスタに転送しません。
	1	IDマスク + MDir + MXtdをIFxメッセージバッファレジスタに転送します。
[bit5]	ARB	アービトレーションビットのアクセス
	0	ID + Dir + Xtd + MsgVal をIFx メッセージバッファレジスタに転送しません。
	1	ID + Dir + Xtd + MsgValをIFxメッセージバッファレジスタに転送します。
[bit4]	CONTROL	制御ビットのアクセス
	0	制御ビットをIFx メッセージバッファレジスタに転送しません。
	1	制御ビットをIFxメッセージバッファレジスタに転送します。
[bit3]	CIP	割込み保留ビットをクリアします。
	0	INTPNDビットを更新しません。
	1	メッセージオブジェクト内のINTPNDビットをクリアします。
[bit2]	TXREQ	新規データビットのアクセス
	0	NEWDATビットを更新しません。
	1	メッセージオブジェクト内のNEWDATビットをクリアします。
[bit1]	DATAA	データバイト0～3のアクセス
	0	データバイト0～3をIFx メッセージバッファレジスタに転送しません。
	1	データバイト0～3をIFxメッセージバッファレジスタに転送します。
[bit0]	DATAB	データバイト4～7のアクセス
	0	データバイト4～7をIFx メッセージバッファレジスタに転送しません。
	1	データバイト4～7をIFxメッセージバッファレジスタに転送します。

＜注意事項＞

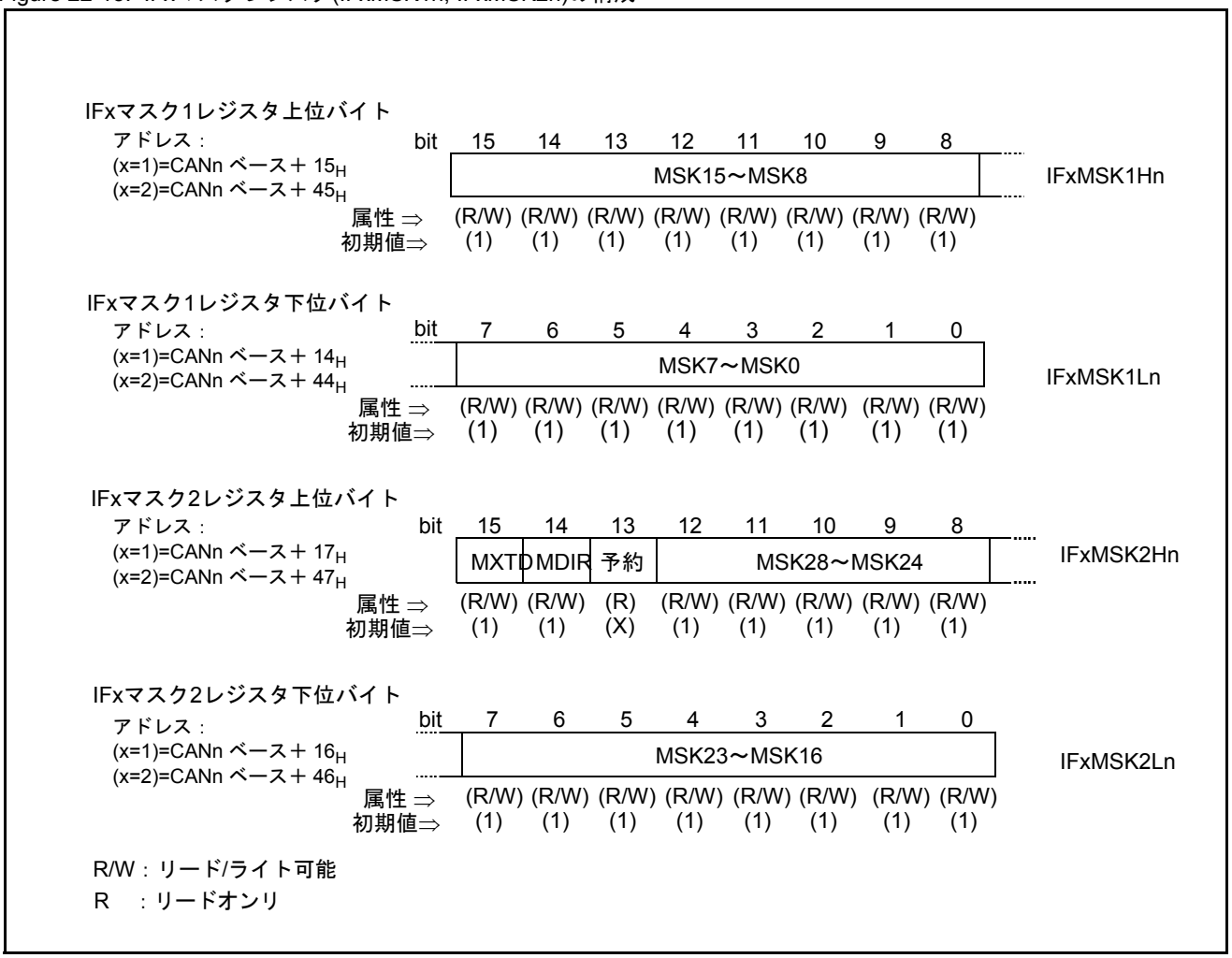
メッセージオブジェクトへのリードアクセスを INTPND ビットおよび NEWDAT ビットのリセットと組み合せることができます。IFx メッセージ制御レジスタの INTPND ビットおよび NEWDAT ビットには , リセットされる前の値が格納されます。

22.6.3 IFxマスクレジスタ(IFxMSK1n, IFxMSK2n)

メッセージバッファレジスタの各ビットはメッセージRAM内のメッセージオブジェクトのミラーです。IFxマスクレジスタ(IFxMSK1n, IFxMSK2n)の各ビットは, メッセージアービトレーションレジスタ(IFxARB1n, IFxARB2n)のどの部分を評価するかを指定します。

IFxマスクレジスタ(IFxMSK1n, IFxMSK2n)

Figure 22-13. IFxマスクレジスタ(IFxMSK1n, IFxMSK2n)の構成

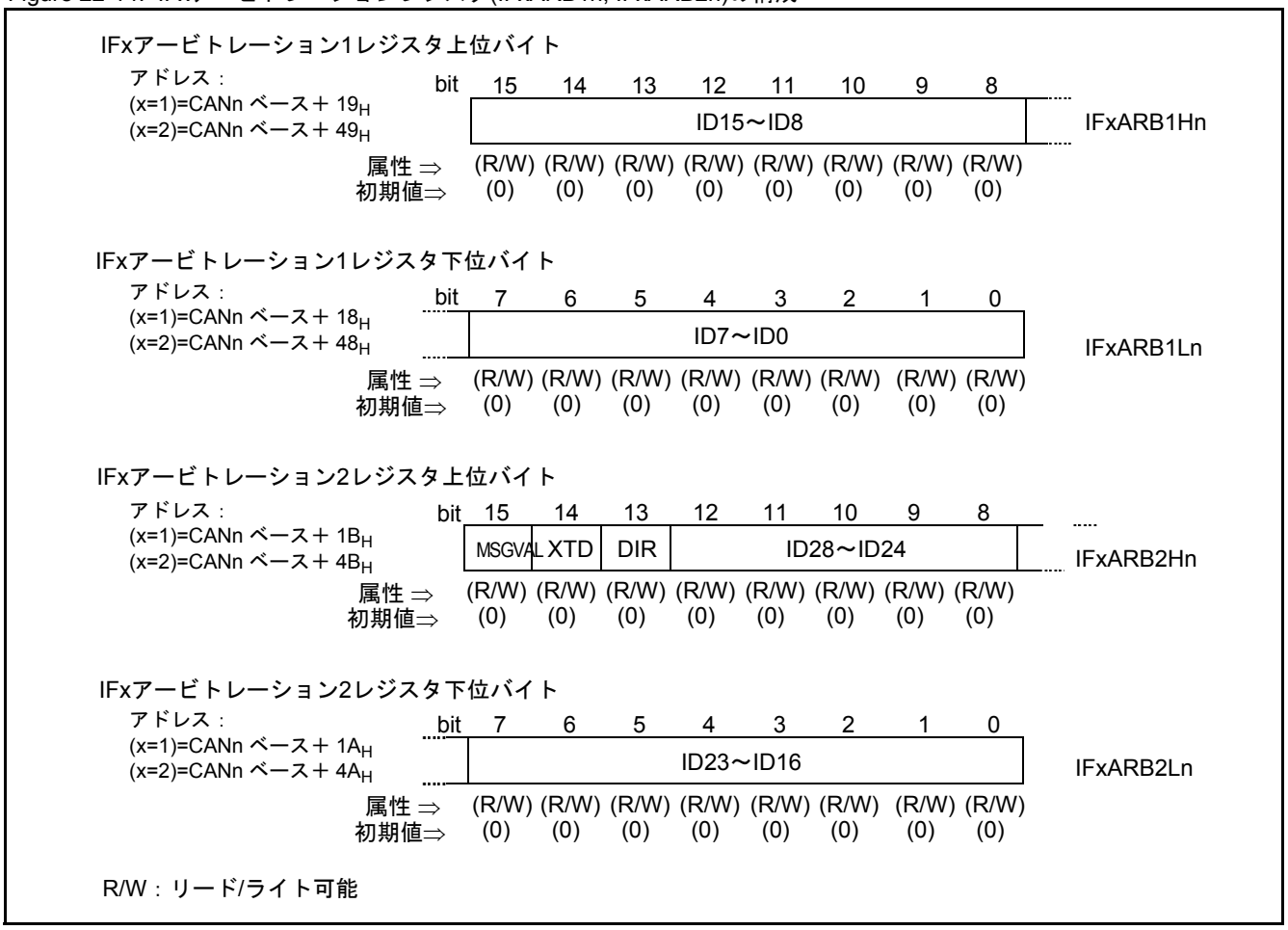


22.6.4 IFxアービトレーションレジスタ(IFxARB1n, IFxARB2n)

メッセージバッファレジスタの各ビットはメッセージRAM内のメッセージオブジェクトのミラーです。送信の場合, メッセージアービトレーションレジスタ(IFxARB1n, IFxARB2n)の各ビットはメッセージのIDを指定します。受信の場合, メッセージアービトレーションレジスタ(IFxARB1n, IFxARB2n)の各ビットはアクセプタンスフィルタを指定します。

IFxアービトレーションレジスタ(IFxARB1n, IFxARB2n)

Figure 22-14. IFxアービトレーションレジスタ(IFxARB1n, IFxARB2n)の構成

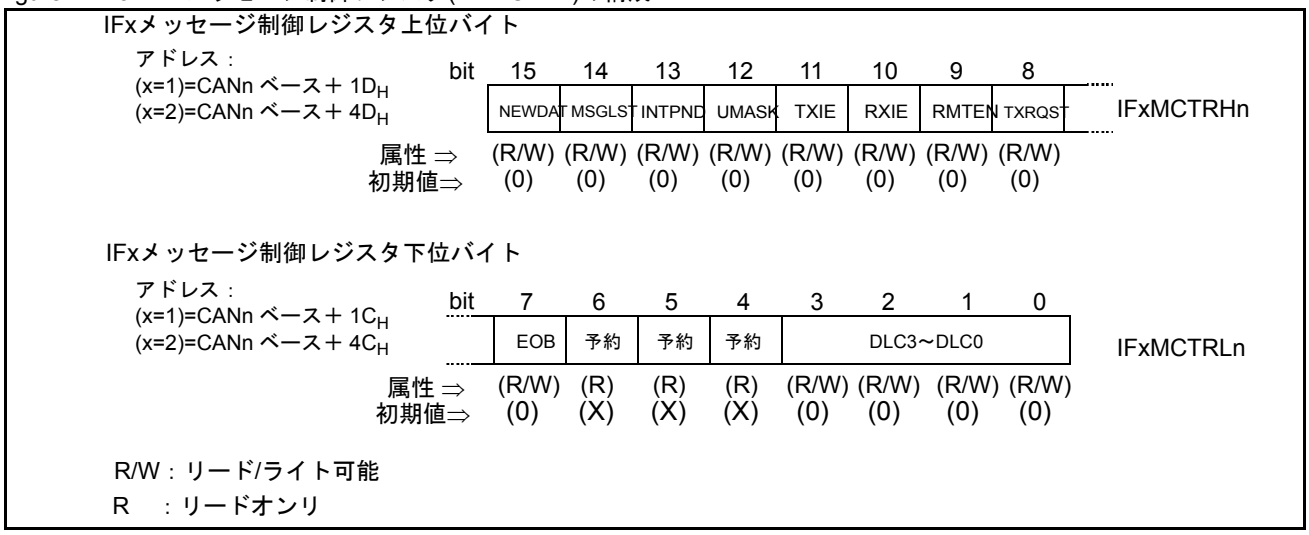


22.6.5 IFxメッセージ制御レジスタ(IFxMCTRn)

メッセージバッファレジスタの各ビットはメッセージ RAM 内のメッセージオブジェクトのミラーです。メッセージ制御レジスタ (IFxMCTRn) のビットにはステータス情報と制御ビットが格納されます。

IFxメッセージ制御レジスタ(IFxMCTRn)

Figure 22-15. IFxメッセージ制御レジスタ(IFxMCTRn)の構成



22.6.6 IFxデータAおよびデータBレジスタ(IFxDTA1n, IFxDTA2n, IFxDTB1n, IFxDTB2n)

CANメッセージのデータバイトはIFxメッセージバッファレジスタに格納されます。

IFxデータAおよびデータBレジスタ(IFxDTAn, IFxDTBn)

CAN のデータフレーム内では , Data(0) が送受信される最初のバイトで Data(7) が最後のバイトです。CAN のシリアルビットストリームでは , 各バイトの MSB が最初に送信されます。

CANメッセージのデータバイトは, 下記の順序でIFxメッセージバッファレジスタに格納されます。

Table 22-3. メッセージバッファの構成

レジスタ	アドレス	アドレス + 0の内容	アドレス + 1の内容
IFxDTA1n	x = 1: CANnベース + 0x1E x = 2: CANnベース + 0x4E	Data(0)	Data(1)
IFxDTA2n	x = 1: CANnベース + 0x20 x = 2: CANnベース + 0x50	Data(2)	Data(3)
IFxDTB1n	x = 1: CANnベース + 0x22 x = 2: CANnベース + 0x52	Data(4)	Data(5)
IFxDTB2n	x = 1: CANnベース + 0x24 x = 2: CANnベース + 0x54	Data(6)	Data(7)

22.7 メッセージメモリ内のメッセージオブジェクト

メッセージRAM内には32のメッセージオブジェクト(実装によっては最大128)が存在します。CPUによるメッセージRAMのアクセスとCANメッセージの送受信の競合を避けるために,CPUはメッセージオブジェクトに直接アクセスすることはできず, CPUからのアクセスはIFxインタフェースレジスタを通じて処理されます。

メッセージメモリ内のオブジェクトの構造

Figure 22-16 にメッセージオブジェクトの 2 種類の構造の概要を示します。

Figure 22-16. メッセージメモリ内のオブジェクトの構造

メッセージオブジェクト												
UMASK	MSK28～MSK0	MXTD	MDIR	EOB	NEWDAT		MSGLST	RXIE	TXIE	INTPND	RMTEN	TXRQST
MSGVAL	ID28～ID0	XTD	DIR	DLC3～DL0	Data0	Data1	Data2	Data3	Data4	Data5	Data6	Data7

MSGVAL　メッセージ有効
- 0　このメッセージオブジェクトはメッセージハンドラによって無視する。
- 1　このメッセージオブジェクトは構成されていて, メッセージハンドラによって処理する。

＜注意事項＞

CAN 制御レジスタの INIT ビットでリセットする前に , 使用していないすべてのメッセージオブジェクトの MSGVAL ビットをリセットする必要があります。また , このビットは , 識別子 ID28 ～ ID0, 制御ビット XTD, DIR, またはデータ長コード DLC3 ～ DLC0 が変更される前 , またはこのメッセージオブジェクトが必要でなくなった場合にもリセットする必要があります。

UMASK　アクセプタンスマスク使用
- 0　マスクを無視します。
- 1　アクセプタンスフィルタリング用にマスク(MSK28～MSK0, MXTD, およびMDIR)を使用します。

＜注意事項＞

UMASK ビットを "1" にセットした場合は , そのメッセージオブジェクトの初期化時の MSGVAL を "1"にセットする前に, そのメッセージオブジェクトのマスクビットのプログラミングを行う必要があります。

ID28-0　メッセージID
- ID28 ～ ID0　29ビットID (拡張フレーム)
- ID28 ～ ID18　11ビットID (標準フレーム)

MSK28～MSK0　識別マスク
- 0　対応するIDビットをマスクします。
- 1　対応するIDビットをアクセプタンスフィルタリングに使用します。

XTD　拡張ID
- 0　このメッセージオブジェクトに11ビット(標準) IDを使用します。
- 1　このメッセージオブジェクトに29ビット(拡張) IDを使用します。

MXTD	マスク拡張ID
0	拡張IDビット(XTD)はアクセプタンスフィルタリングに影響を与えません。
1	拡張IDビット(XTD)をアクセプタンスフィルタリングに使用します。

＜注意事項＞

メッセージオブジェクトに 11 ビット (標準) ID を使用する場合は , 受信データフレームの ID は ID28 ～ ID18 ビットに書き込まれます。アクセプタンスフィルタリングには , この 11 ビットとマスクビット MSK28 ～ MSK18 のみが使用されます。

DIR	メッセージの方向
0	方向=受信－TXRQSTで, リモートフレームとこのメッセージオブジェクトのIDを送信します。IDが一致するデータフレームを受信すると, そのメッセージは直ちにこのメッセージオブジェクト内に格納されます。
1	方向=送信－TXRQSTで, 各メッセージオブジェクトをデータフレームとして送信します。IDが一致するリモートフレームを受信すると, このメッセージオブジェクトのTXRQSTビットが直ちにセットされます(RMTEN=1の場合)。

MDIR	メッセージ方向マスク
0	メッセージ方向ビット(DIR)はアクセプタンスフィルタリングに影響を与えません。
1	メッセージ方向ビット(DIR)をアクセプタンスフィルタリングに使用します。

アービトレーションレジスタ ID28 ～ ID0, XTD, および DIR は送信メッセージの ID とタイプの指定 , および (マスクレジスタ MSK28 ～ MSK0, MXTD, および MDIR とともに) 受信メッセージのアクセプタンスフィルタリングに使用されます。受信したメッセージは , ID および方向 = 受信 (データフレーム) または方向 = 送信 (リモートフレーム) が一致する有効なメッセージオブジェクト内に格納されます。拡張フレームは XTD=1 のメッセージオブジェクトのみに , 標準フレームは XTD=0 のメッセージオブジェクトのみに格納することができます。受信したメッセージ (データフレームまたはリモートフレーム) が複数の有効なメッセージオブジェクトに一致した場合は , メッセージ番号の最も小さなメッセージオブジェクトに格納されます。詳細については「■ 受信したメッセージのアクセプタンスフィルタリング 」を参照してください。

EOB　バッファの終わり

0　メッセージオブジェクトはFIFOバッファに属しているが, そのFIFOバッファの最後のメッセージオブジェクトではないことを示します。

1　単一のメッセージオブジェクトまたはFIFOバッファの最後のメッセージオブジェクトを示します。

＜注意事項＞

本ビットは , 複数 (最大 32) のメッセージオブジェクトを連結して FIFO バッファを構成するために使用します。単一のメッセージオブジェクト (FIFO バッファに属していない) の場合は , 本ビットを常に "1" にセットする必要があります。メッセージオブジェクトの連結の詳細については「22.9 CAN アプリケーション -FIFO バッファの構成」を参照してください。

NEWDAT　新規データ

0　前回このフラグがCPUによってクリアされてから, メッセージハンドラによってこのメッセージオブジェクトのデータ部分に書き込まれた新しいデータは存在しないことを示します。

1　メッセージハンドラまたはCPUによって, このメッセージオブジェクトのデータ部分に新しいデータが書き込まれていることを示します。

MSGLST　メッセージロスト(方向=受信のメッセージオブジェクトにのみ有効)

0　本ビットが前回CPUによってリセットされてからメッセージロストは発生していないことを示します。

1　NewDatがセットされていたときにメッセージハンドラがこのオブジェクトに新しいメッセージを格納し, CPUがメッセージをロストしたことを示します。

RXIE　受信割込み許可

0　フレームが正しく受信された後でINTPNDを変更しません。

1　フレームが正しく受信された後でINTPNDをセットします。

TXIE　送信割込み許可

0　フレームが正しく送信された後でINTPNDを変更しません。

1　フレームが正しく送信された後でINTPNDをセットします。

INTPND　割込み保留

0　このメッセージオブジェクトを割込みの要因にしません。

1　このメッセージオブジェクトを割込みの要因にする。より優先度の高い割込み要因が存在しない場合は, 割込みレジスタの割込みIDがこのメッセージオブジェクトを指します。

RMTEN　リモート許可

0　リモートフレームの受信時にTXRQSTを変更しません。

1　リモートフレームの受信時にTXRQSTをセットします。

TXRQST	送信リクエスト
0	このメッセージオブジェクトは送信待ちではないことを示します。
1	このメッセージオブジェクトは送信を要求しているが, まだ送信されていないことを示します。

＜注意事項＞

優先度の最も低いメッセージバッファが送信に使用される場合 , TXRQST を "0" に設定すると , TXRQST を "1" に再設定したときに送信が遅延することがあります。

TXRQST が "0" に設定された時間によっては , TXRQST を "1" に設定した直後にメッセージが送信されない場合があり , メッセージは下記のいずれかのイベントの後に送信されます。

- CAN バス上で継続中の動作が存在する
- 送信要求が別のメッセージオブジェクトへ発行される
- CAN バスが INIT ビットによって初期化される

一般的に, TXRQSTを"0"に設定して継続中の送信をキャンセルする必要はありません。TXRQST=1のときにメッセージオブジェクトの内容変更が必要な場合 , CPU インタフェースのレジスタ (識別子 , DLC, データ : TXRQST および NEWDAT, また必要に応じて TXIE を使用) を通してメッセージオブジェクトを更新することで十分です。更新内容は次の機会に送信されます。

DLC3～DLC0	データ長コード
0～8	データフレーム長は0～8バイトです。
9～15	データフレーム長は8バイトです。

＜注意事項＞

メッセージオブジェクトのデータ長コードは，ほかのノードで同じIDに対応するすべてのオブジェクトで同じ長さになるように定義する必要があります。メッセージハンドラがデータフレームを格納するとき, 受信したメッセージによって与えられた値にDLCが書き換えられます。

Data0	CANデータフレームの第1データバイト
Data1	CANデータフレームの第2データバイト
Data2	CANデータフレームの第3データバイト
Data3	CANデータフレームの第4データバイト
Data4	CANデータフレームの第5データバイト
Data5	CANデータフレームの第6データバイト
Data6	CANデータフレームの第7データバイト
Data7	CANデータフレームの第8データバイト

＜注意事項＞

バイトデータ 0 は受信時に CAN コアのシフトレジスタに格納される最初のデータバイトで , バイトデータ 7 は最後のデータバイトです。メッセージハンドラはデータフレームを格納すると , 8 つのデータバイトのすべてをメッセージオブジェクトに書き込みます。データ長コードが 8 未満の場合は , メッセージオブジェクトの残りのバイトは不定値で上書きされます。

22.8 メッセージハンドラのレジスタ

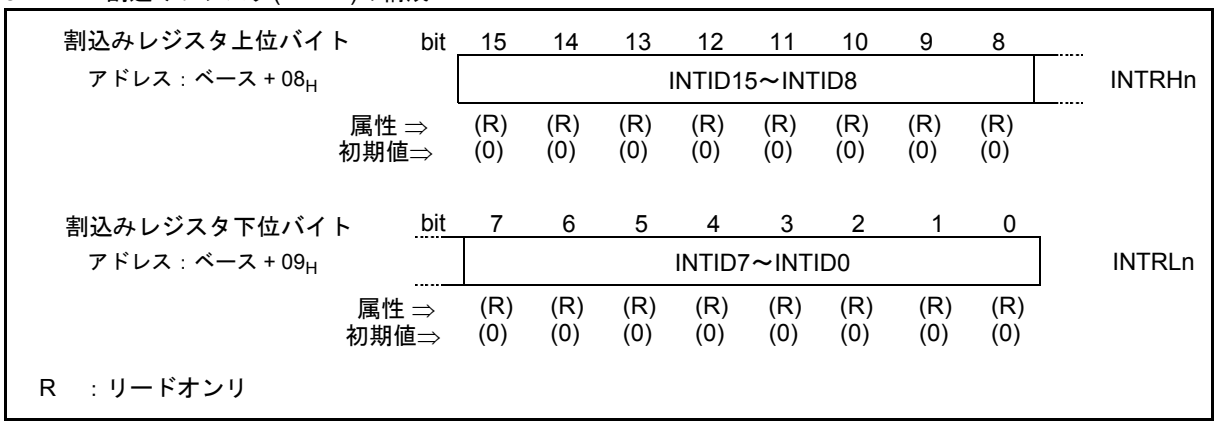
メッセージハンドラのレジスタはすべて読出し専用です。その内容(各メッセージオブジェクトのTXRQST, NEWDAT,INTPND,MSGVALの各ビットおよび割込みID)は,メッセージハンドラFSMによって提供されるステータス情報です。

22.8.1 割込みレジスタ(INTRn)

割込みレジスタ(INTRn)は, 保留中の最も高い優先度の割込みを指しています。

割込みレジスタ(INTRn)

Figure 22-17. 割込みレジスタ(INTRn)の構成



割込みレジスタ(INTRn)の機能

(32メッセージバッファCANの場合)

INTID15～INTID0	割込みID (この番号は割込みの要因を示します)
0x0000	保留中の割込みは存在しない。
0x0001～0x0020	割込みを生成したメッセージオブジェクトの番号
0x0021～0x7FFF	未使用
0x8000	ステータス割込み
0x8001-0xFFFF	未使用

(128メッセージバッファCANの場合)

INTID15～INTID0	割込みID (この番号は割込みの要因を示します)
0x0000	保留中の割込みは存在しない。
0x0001～0x0080	割込みを生成したメッセージオブジェクトの番号
0x0081～0x7FFF	未使用
0x8000	ステータス割込み
0x8001～0xFFFF	未使用

複数の割込みが保留中の場合 , CAN 割込みレジスタは , 発生順序とは無関係に最も優先度の高い保留中の割込みを指しています。割込みは CPU によってクリアされるまで保留状態を維持します。IntId が 0000_H で IE がセットされている場合は , CPU への割込みラインはアクティブです。割込みラインは IntId が 0000_H の値に戻るか (割込みの場合はリセット) IE がリセットされるまでアクティブな状態を維持します。

優先度が最も高い割込みはステータス割込みです。メッセージ割込みの中では , メッセージ番号が大きいものほどメッセージオブジェクトの割込み優先順位が下がります。

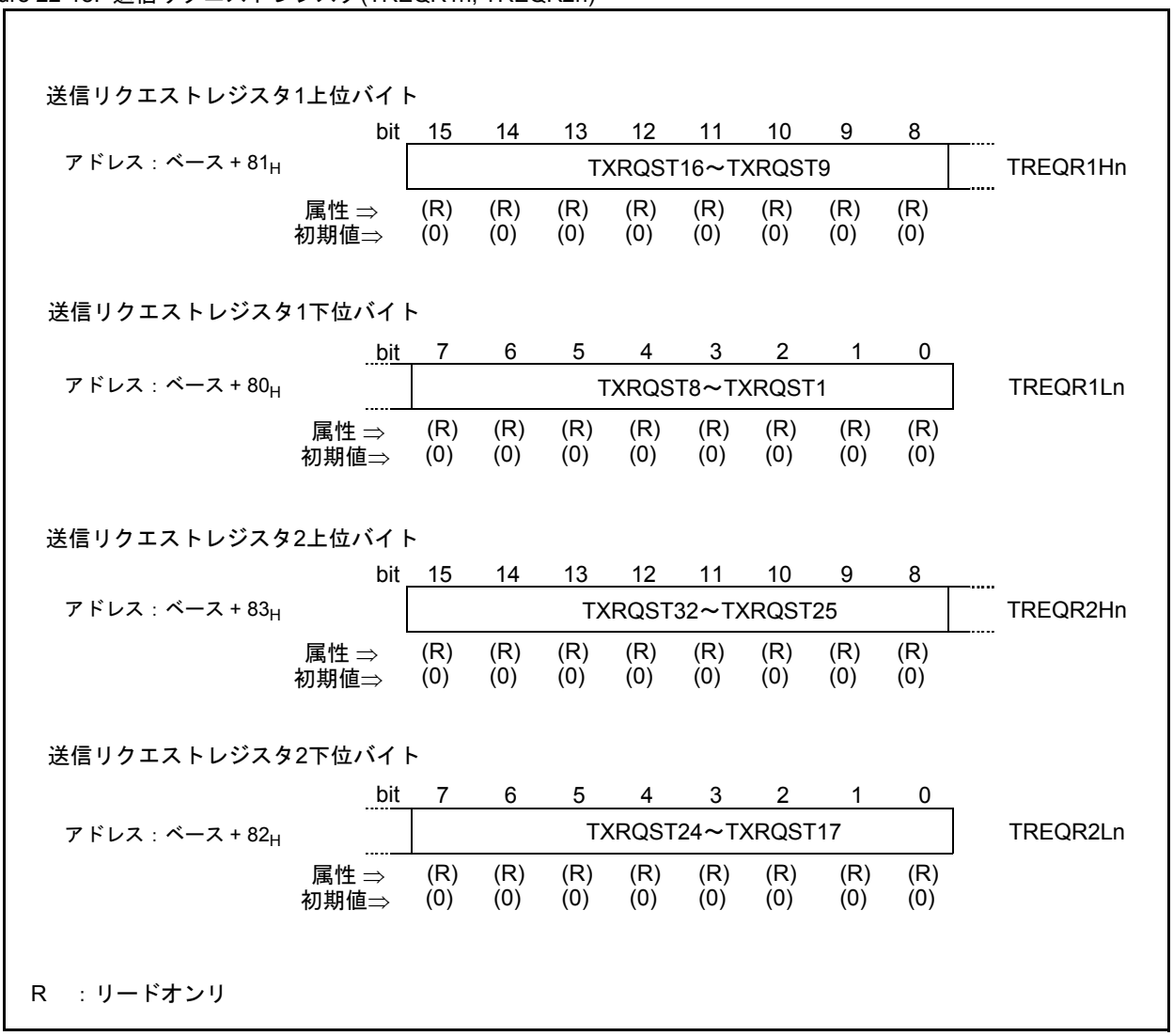
メッセージ割込みをクリアするにはメッセージオブジェクトの INTPND ビットをクリアします。ステータスレジスタにリードアクセスすると , ステータス割込みがクリアされます。

22.8.2 送信リクエストレジスタ(TREQR1n, TREQR2n)

送信リクエストレジスタ(TREQR1n, TREQR2n)は, メッセージオブジェクトの送信を制御します。

送信リクエストレジスタ(TREQR1n, TREQR2n)

Figure 22-18. 送信リクエストレジスタ(TREQR1n, TREQR2n)



送信リクエストレジスタ(TREQR1n, TREQR2n)の機能

TXRQST32～TXRQST1	送信リクエストビット(すべてのメッセージオブジェクトの)
0	このメッセージオブジェクトは送信待ちではない。
1	このメッセージオブジェクトは送信を要求しているが, まだ送信されていない。

上記のレジスタには 32 のメッセージオブジェクトの TXRQST ビットが保持されています。CPU は TXRQST ビットを読み出すことにより, どのメッセージオブジェクトの送信リクエストが保留されているのかをチェックすることができます。CPU から IFx メッセージインタフェースレジスタを通じて , またはリモートフレームの受信後または送信成功後にメッセージハンドラから , 特定のメッセージオブジェクトの TXRQST ビットをセット / リセットすることができます。

＜注意事項＞

優先度の最も低いメッセージバッファが送信に使用される場合 , TXRQST を "0" に設定すると , TXRQST を "1" に再設定したときに送信が遅延することがあります。

TXRQST が "0" に設定された時間によっては , TXRQST を "1" に設定した直後にメッセージが送信されない場合があり , メッセージは下記のいずれかのイベントの後に送信されます。

- CAN バス上で継続中の動作が存在する
- 送信要求が別のメッセージオブジェクトへ発行される
- CAN バスが INIT ビットによって初期化される

一般的に, TXRQSTを"0"に設定して継続中の送信をキャンセルする必要はありません。TXRQST=1のときにメッセージオブジェクトの内容変更が必要な場合 , CPU インタフェースのレジスタ (識別子 , DLC, データ :TXRQST および NEWDAT, また必要に応じて TXIE を使用) を通してメッセージオブジェクトを更新することで十分です。更新内容は次の機会に送信されます。

32を超えるメッセージバッファが実装されている場合の追加フラグの概要を下表に示します。

Table 22-4. 32を超えるメッセージバッファが実装されている場合の追加フラグ

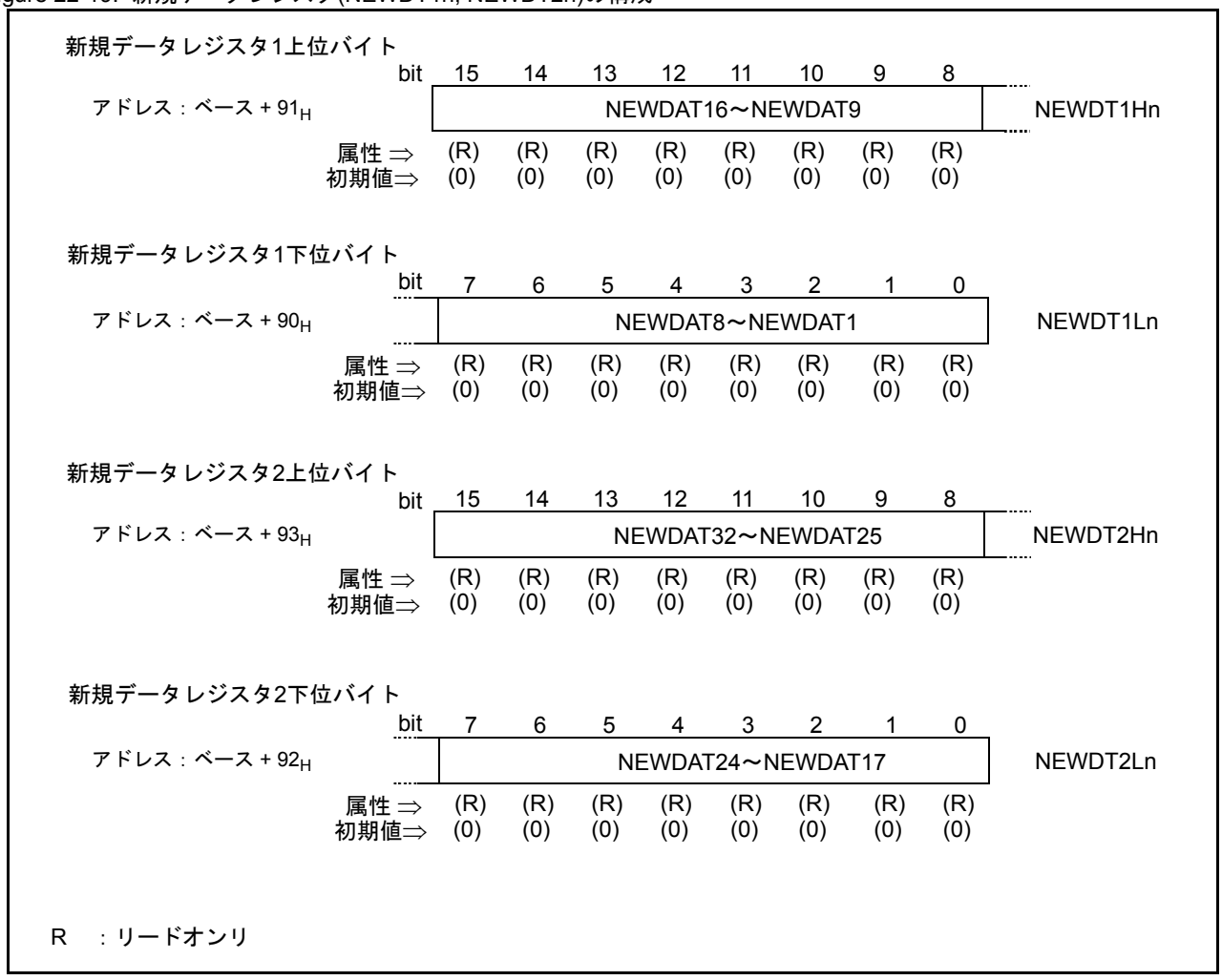
		addr+0	addr+1	addr+2	addr+3
TREQR 4および3	TXRQST 64-33 (アドレス84_H)	TXRQST64-57	TXRQST56-49	TXRQST48-41	TXRQST40-33
TREQR 6および5	TXRQST 96-65 (アドレス88_H)	TXRQST96-89	TXRQST88-81	TXRQST80-73	TXRQST72-65
TREQR 8および7	TXRQST 128-97 (アドレス$8C_H$)	TXRQST128-121	TXRQST120-113	TXRQST112-105	TXRQST104-97

22.8.3 新規データレジスタ(NEWDT1n, NEWDT2n)

新規データレジスタ(NEWDT1n,NEWDT2n)は,メッセージバッファを通じて新しいデータを受信したことを示します。

新規データレジスタ(NEWDT1n, NEWDT2n)

Figure 22-19. 新規データレジスタ(NEWDT1n, NEWDT2n)の構成



新規データレジスタ(NEWDT1n, NEWDT2n)の機能

NEWDAT32～NEWDAT1	新規データビット(すべてのメッセージオブジェクトの)
0	前回このフラグがCPUによってクリアされてから, メッセージハンドラによってこのメッセージオブジェクトのデータ部分に書き込まれた新しいデータは存在しない。
1	メッセージハンドラまたはCPUによって, このメッセージオブジェクトのデータ部分に新しいデータが書き込まれている。

上記のレジスタには 32 のメッセージオブジェクトの NEWDAT ビットが保持されています。CPU は , NEWDAT ビットを読み出すことにより , どのメッセージのデータ部分が更新されたかをチェックすることができます。CPU から IFx メッセージインタフェースレジスタを通じて , またはデータフレームの受信後または送信成功後に

メッセージハンドラから , 特定のメッセージオブジェクトの NEWDAT ビットをセット / リセットすることができます。

32を超えるメッセージバッファが実装されている場合の追加フラグの概要を下表に示します。

Table 22-5. 32を超えるメッセージバッファが実装されている場合の追加フラグ

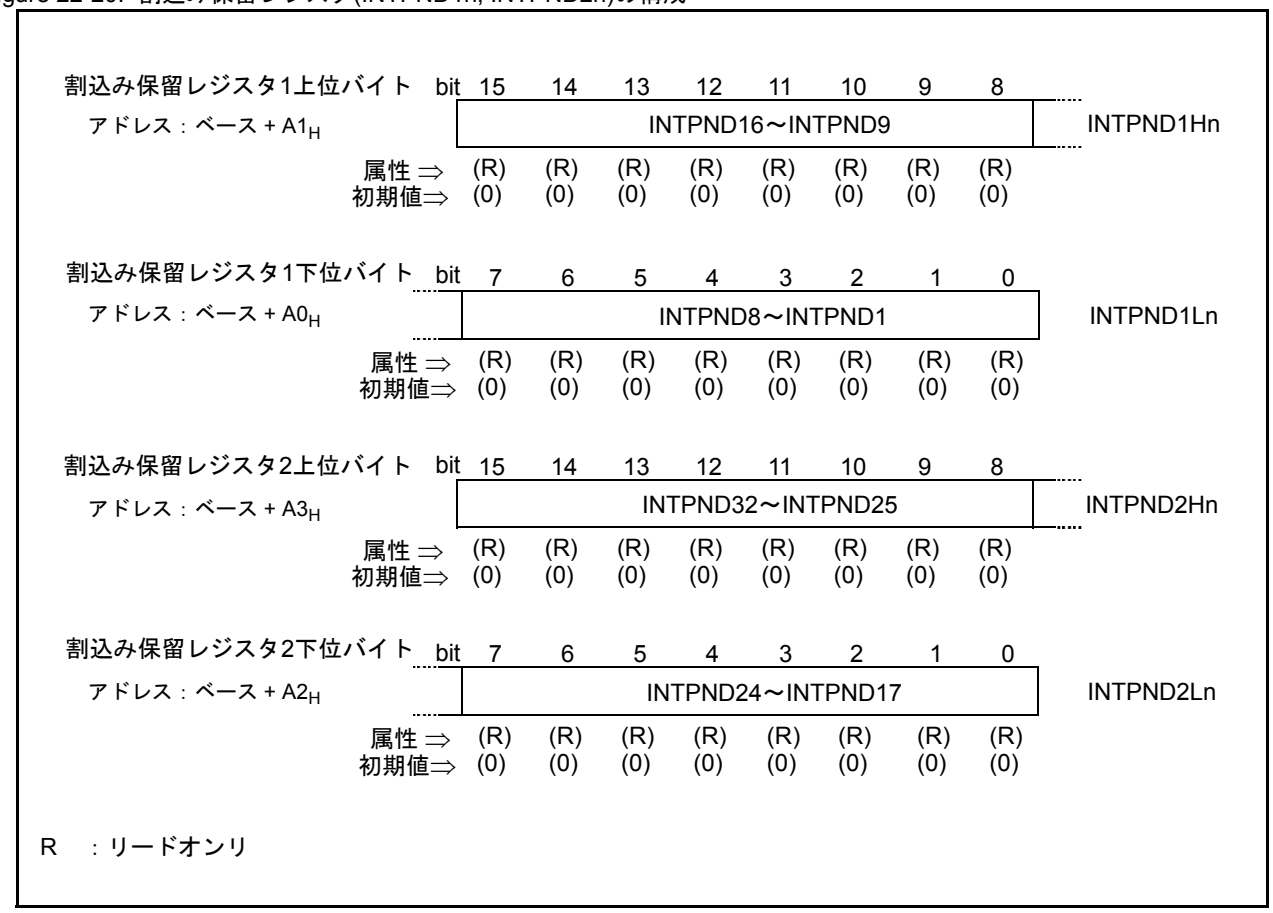
		addr+0	addr+1	addr+2	addr+3
NEWDT 4 および 3	NEWDAT 64-33 (アドレス94_H)	NEWDAT64-57	NEWDAT56-49	NEWDAT48-41	NEWDAT40-33
NEWDT 6および5	NEWDAT 96-65 (アドレス98_H)	NEWDAT96-89	NEWDAT88-81	NEWDAT80-73	NEWDAT72-65
NEWDT 8および7	NEWDAT 128-97 (アドレス$9C_H$)	NEWDAT128-121	NEWDAT120-113	NEWDAT112-105	NEWDAT104-97

22.8.4 割込み保留レジスタ(INTPND1n, INTPND2n)

割込み保留レジスタ(INTPND1n, INTPND2n)は, メッセージオブジェクトが割込みの原因かどうかを示します。

割込み保留レジスタ(INTPND1n, INTPND2n)

Figure 22-20. 割込み保留レジスタ(INTPND1n, INTPND2n)の構成



割込み保留レジスタ(INTPND1n, INTPND2n)の機能

INTPND32～INTPND1	割込み保留ビット(すべてのメッセージオブジェクトの)
0	このメッセージオブジェクトは割込みの要因ではない。
1	このメッセージオブジェクトは割込みの要因である。

上記のレジスタには 32 のメッセージオブジェクトの INTPND ビットが保持されています。CPU は INTPND ビットを読み出すことにより, どのメッセージオブジェクトの割込みが保留されているのかをチェックすることができます。CPU から IFx メッセージインタフェースレジスタを通じて , またはフレームの受信後または送信成功後にメッセージハンドラから , 特定のメッセージオブジェクトの INTPND ビットをセット / リセットすることができます。この操作は割込みレジスタ INTRn の INTID の値にも影響を与えます。

32を超えるメッセージバッファが実装されている場合の追加フラグの概要を下表に示します。

Table 22-6. 32を超えるメッセージバッファが実装されている場合の追加フラグ

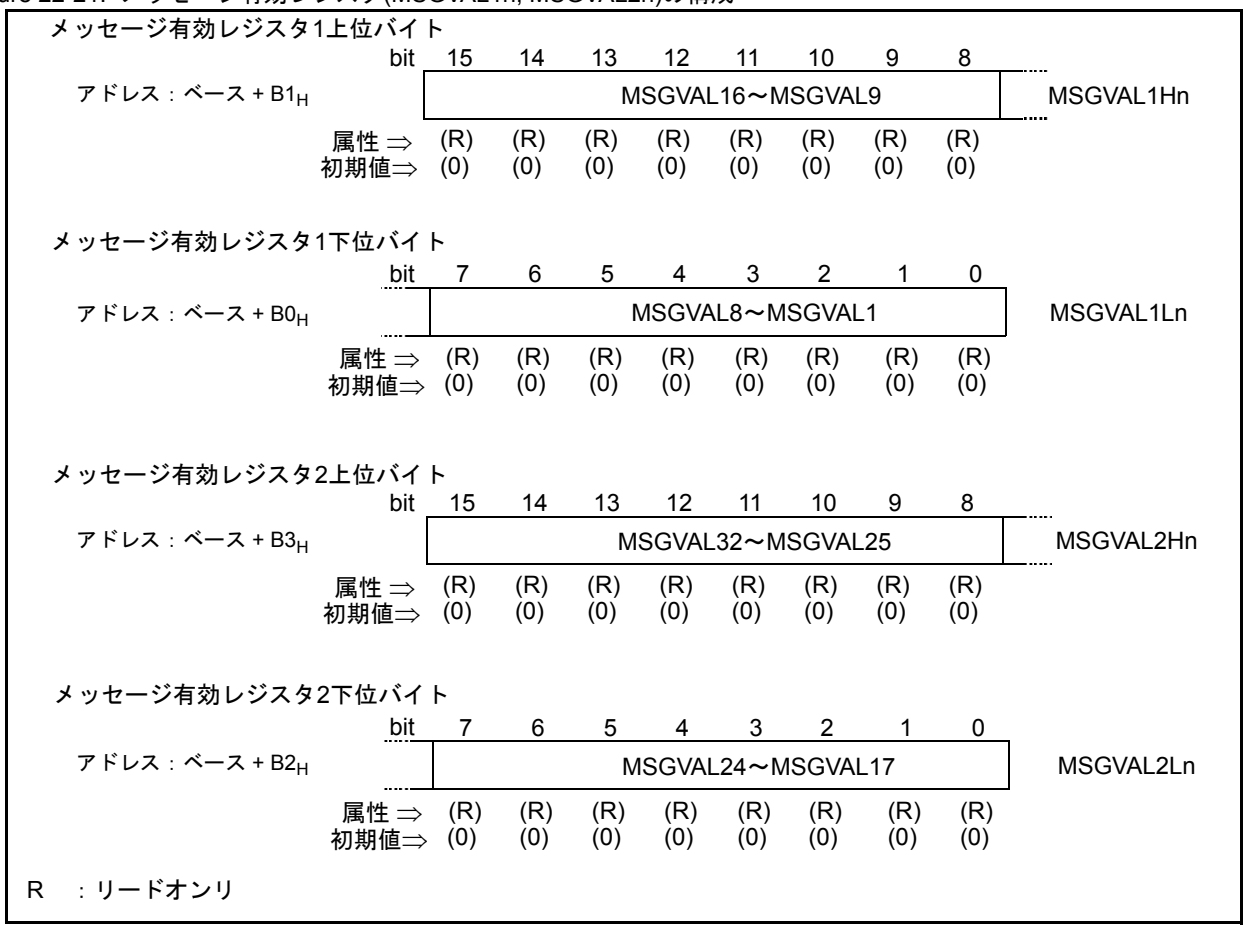
		addr+0	addr+1	addr+2	addr+3
INTPND 4および3	INTPND 64-33 (アドレス 0xA4)	INTPND64-57	INTPND56-49	INTPND48-41	INTPND40-33
INTPND 6および5	INTPND 96-65 (アドレス 0xA8)	INTPND96-89	INTPND88-81	INTPND80-73	INTPND72-65
INTPND 8および7	INTPND 128-97 (アドレス 0xAC)	INTPND128-121	INTPND120-113	INTPND112-105	INTPND104-97

22.8.5 メッセージ有効レジスタ(MSGVAL1n, MSGVAL2n)

メッセージ有効レジスタ(MSGVAL1n, MSGVAL2n)は, 各メッセージオブジェクトの有効性を示します。

メッセージ有効レジスタ(MSGVAL1n, MSGVAL2n)

Figure 22-21. メッセージ有効レジスタ(MSGVAL1n, MSGVAL2n)の構成



メッセージ有効レジスタ(MSGVAL1n, MSGVAL2n)の機能

MSGVAL32～MSGVAL1	メッセージ有効ビット(すべてのメッセージオブジェクトの)
1	このメッセージオブジェクトはメッセージ有効ビットで構成されていて, メッセージハンドラによって処理する。

上記のレジスタには 32 のメッセージオブジェクトの MSGVAL ビットが保持されています。CPU は MSGVAL ビットを読み出すことにより , どのメッセージオブジェクトが有効かチェックすることができます。CPU から IFx メッセージインタフェースレジスタを通じて特定のメッセージオブジェクトの MSGVAL ビットをセット / リセットすることができます。

32を超えるメッセージバッファが実装されている場合の追加フラグの概要を下表に示します。

Table 22-7. 32を超えるメッセージバッファが実装されている場合の追加フラグ

		addr+0	addr+1	addr+2	addr+3
MSGVAL 4および3	MSGVAL 64-33 (アドレス 0xB4)	MSGVAL64-57	MSGVAL56-49	MSGVAL48-41	MSGVAL40-33
MSGVAL 6および5	MSGVAL 96-65 (アドレス 0xB8)	MSGVAL96-89	MSGVAL88-81	MSGVAL80-73	MSGVAL72-65
MSGVAL 8および7	MSGVAL 128-97 (アドレス 0xBC)	MSGVAL128-121	MSGVAL120-113	MSGVAL112-105	MSGVAL104-97

22.9 CANアプリケーション

アプリケーションでのCANモジュールの使用方法を説明します。

メッセージオブジェクトの制御

メッセージ RAM 内のメッセージオブジェクトの構成はチップのリセットによる影響を受けません (MSGVAL, NEWDAT, INTPND, および TXRQST ビットを除く)。すべてのメッセージオブジェクトを CPU によってリセットする必要があります。リセットしなかったメッセージオブジェクトは無効 (MSGVAL = 0) になります。また , CPU から CAN 制御レジスタの INIT ビットをクリアする前にビットタイミングを構成する必要があります。

メッセージオブジェクトの構成は , 2 つのインタフェースレジスタのどちらかのマスク , アービトレーション , 制御 , およびデータフィールドを適切な値にプログラミングすることにより実行します。対応する IFx コマンドリクエストレジスタに書き込むことにより , IFx メッセージバッファレジスタがアドレス指定したメッセージRAM内のメッセージオブジェクトにロードされます。

CAN 制御レジスタの INIT ビットをクリアすると , CAN_Core の CAN プロトコルコントローラステートマシンとメッセージハンドラステートマシンにより CAN の内部データフローの制御が行われます。アクセプタンスフィルタリングに渡される受信メッセージはメッセージ RAM に格納され , 送信リクエスト保留中のメッセージは CAN_Core のシフトレジスタにロードされ , CAN バスを通じて送信されます。

CPUは受信メッセージを読出し, IFxインタフェースレジスタを通じて送信するためにメッセージを更新します。構成に基づいて , 所定の CAN メッセージおよび CAN エラーイベントによって CPU に割込みがかかります。

メッセージハンドラステートマシン

メッセージハンドラは , CAN コアの Rx/Tx シフトレジスタ , メッセージ RAM, IFx レジスタの相互間のデータ転送を制御します。

メッセージハンドラ FSM は , 下記の機能を制御します。

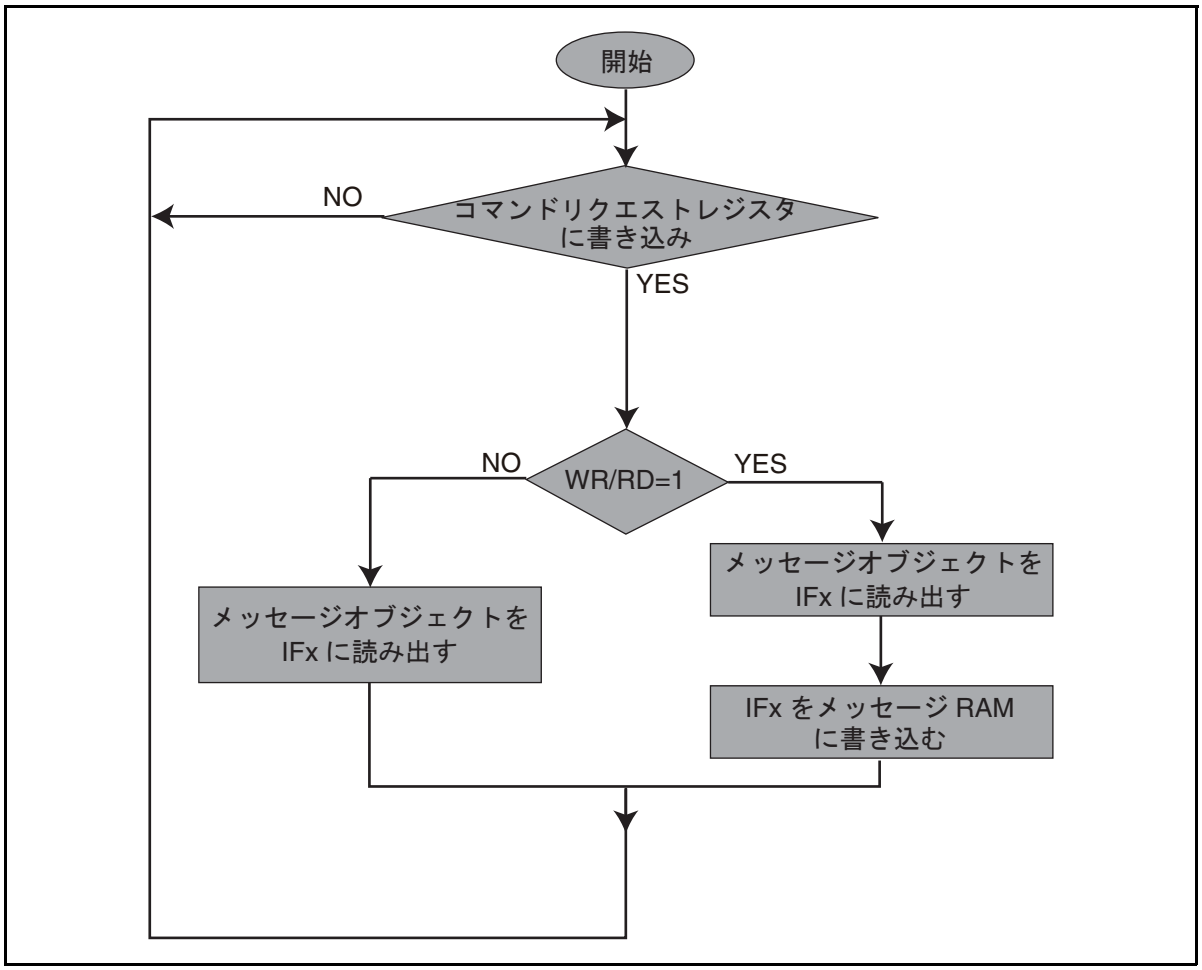
- IFxレジスタからメッセージRAMへのデータの転送
- メッセージRAMからIFxレジスタへのデータの転送
- シフトレジスタからメッセージRAMへのデータの転送
- メッセージRAMからシフトレジスタへのデータの転送
- シフトレジスタからアクセプタンスフィルタリングユニットへのデータの転送
- メッセージオブジェクトのマッチングのためのメッセージRAMのスキャン
- TXRQSTフラグの処理
- 割込みの処理

メッセージRAMから/へのデータの転送

CPU は IFx レジスタとメッセージ RAM 間のデータ転送を開始すると , メッセージハンドラは内部的なビジー信号をセットし , 次の連続アクセスを遅延させます。転送が完了すると , このビジー信号がリセットされ , 次のアクセスが実行されます。

各コマンドマスクレジスタは , メッセージオブジェクト全体を転送するのか , その一部を転送するのかを指定します。メッセージ RAM の構造上の制約により 1 つのメッセージオブジェクトの単一のビット / バイトだけを書き込むことはできないため , メッセージ RAM には常に 1 つのメッセージオブジェクト全体を書き込む必要があります。そのため , IFx レジスタからメッセージ RAM へのデータ転送にはリードモディファイライトサイクルが必要です。まずメッセージオブジェクトの変更を加えない部分をメッセージ RAM から読み出してから , 次にメッセージバッファレジスタの完全な内容をメッセージオブジェクトに転送します。

Figure 22-22. IFxレジスタとメッセージRAM間のデータ転送



メッセージオブジェクトの一部書込みの後で , コマンドマスクレジスタ内の選択されていないメッセージバッファレジスタには , 選択したメッセージオブジェクトの実際の内容がセットされます。

メッセージオブジェクトの一部読出しの後では, コマンドマスクレジスタ内の選択されていないメッセージバッファレジスタには変更はありません。

メッセージの送信

CAN コアセルのシフトレジスタのローディングの準備が完了し IFx レジスタとメッセージ RAM 間のデータ転送がない場合 , メッセージ有効レジスタの MSGVAL ビットと送信リクエストレジスタの TXRQST ビットの評価が行われます。優先度が最も高い保留中の送信リクエストのメッセージオブジェクトがメッセージハンドラによってシフトレジスタにロードされ , 送信が開始されます。メッセージオブジェクトの NEWDAT ビットはリセットされます。

送信に成功した後 , 送信開始以降に新しいデータがメッセージオブジェクトに書き込まれていない場合は (NEWDAT =0), TXRQST ビットがリセットされます。TXIE がセットされているときは , 送信に成功した後で INTPND がセットされます。CAN がアービトレーションをロストしたか送信時にエラーが発生した場合 , そのメッセージは CAN バスが空くのを待って直ちに再送信されます。送信リクエストが保留されているメッセージは , 優先度の高いものから順次送信されます。

受信したメッセージのアクセプタンスフィルタリング

受信メッセージのアービトレーションおよび制御フィールド (ID + IDE + RTR + DLC) が CAN コアの Rx/Tx シフトレジスタに完全にシフトされると , メッセージハンドラ FSM はメッセージ RAM のスキャンを開始して一致する有効なメッセージオブジェクトを検索します。
メッセージ RAM をスキャンして一致するメッセージオブジェクトを検索するために , アクセプタンスフィルタリングユニットが , アービトレーションビットを使用して CAN コアのシフトレジスタからロードされます。次に , メッセージオブジェクト 1 のアービトレーションとマスクフィールド (MSGVAL, UMASK, NEWDAT, EOB) がアクセプタンスフィルタリングユニットにロードされ, シフトレジスタのアービトレーションフィールドと比較されます。この操作は一致するメッセージオブジェクトが見つかるかメッセージ RAM の終わりに達するまで反復されます。
一致するメッセージオブジェクトが存在した場合は, メッセージハンドラ FSM はスキャンを停止し, 受信したフレームの種類 (データフレームまたはリモートフレーム) に基づいて処理を続行します。

データフレームの受信

メッセージハンドラ FSM は , CAN コアのシフトレジスタからのメッセージを , メッセージ RAM 内の対応するメッセージオブジェクトに格納します。データバイトだけでなく , アービトレーションビットやデータ長コードもすべて対応するメッセージオブジェクト内に格納されます。これは , アービトレーションマスクレジスタを使用している場合でも ID で連結されているデータバイトを保持するための実装です。
新しいデータ (まだ CPU からは見えない) を受信したことを示すために NEWDAT ビットがセットされます。CPU は , メッセージオブジェクトを読み出したときに NEWDAT をリセットする必要があります。受信時に NEWDAT ビットが既にセットされていた場合は , 前のデータ (CPU からは見えていない) が失われたことを示すために MSGLST がセットされます。RXIE ビットがセットされていた場合は INTPND ビットがセットされ , 割込みレジスタはこのメッセージオブジェクトを指します。
リクエストしたデータフレームを受信すると , メッセージオブジェクトの TXRQST がリセットされリモートフレームの送信を防止します。

リモートフレームの受信

リモートフレームを受信したときは , 一致するメッセージオブジェクトの 3 種類の構成を考慮する必要があります。

1) DIR = 1 (方向 = 送信), RMTEN =1, UMASK =1 または 0
   一致するリモートフレームを受信すると , そのメッセージオブジェクトの TXRQST ビットがセットされます。他のメッセージオブジェクトは変更されません。
2) DIR =1 (方向 = 送信), RMTEN =0, UMASK =0
   一致するリモートフレームを受信しても , そのメッセージオブジェクトの TXRQST ビットは変化せず , このリモートフレームは無視されます。
3) DIR =1 (方向 = 送信), RMTEN =0, UMASK =1
   一致するリモートフレームを受信すると , そのメッセージオブジェクトの TXRQST ビットがリセットされます。シフトレジスタからのアービトレーションおよび制御フィールド (ID + IDE + RTR + DLC) がメッセージ RAM 内のメッセージオブジェクトに格納され , このメッセージオブジェクトの NEWDAT ビットがセットされます。メッセージオブジェクトのデータフィールドには変更はなく , このリモートフレームは受信したデータフレームと同じように処理されます。

受信/送信の優先度

メッセージオブジェクトの受信 / 送信の優先度はメッセージ番号によって決まります。メッセージオブジェクト 1 には最上位の優先度が与えられ , メッセージオブジェクト 32 (最大の実装メッセージオブジェクト番号) には最下位の優先度が与えられます。複数の送信リクエストが保留中のときは , 対応するメッセージオブジェクトの優先順位に従ってサービスが行われます。

送信オブジェクトの構成

Figure 22-23 に送信オブジェクトを初期化する方法を示します。

Figure 22-23. 送信オブジェクトの初期化

MSGVAL	ARB	DATA	MASK	EOB	DIR	NEWDAT	MSGLST	RXIE	TXIE	INTPND	RMTEN	TXRQST
1	appl.	appl.	appl.	1	1	0	0	0	appl.	0	appl.	0

アービトレーションレジスタ (ID28 ～ ID0 および XTD ビット) はアプリケーションによって設定します。このレジスタで送信メッセージの ID と種類を定義します。11 ビット ID(標準フレーム) を使用する場合は ID28 ～ ID18 を使用し , ID17 ～ ID0 は無視します。
TXIE ビットをセットした場合は , メッセージオブジェクトの送信に成功すると INTPND ビットがセットされます。
RMTEN ビットをセットした場合は , 受信したリモートフレームの一致により TXRQST ビットがセットされ , リモートフレームへの応答はデータフレームによって自発的に行われます。
データレジスタ (DLC3 ～ DLC0, Data0 ～ Data7) はアプリケーションによって設定し , データが有効になる前に TXRQST および RMTEN をセットすることはできません。
マスクレジスタ (MSK28 ～ MSK0, UMASK, MXTD, MDIR の各ビット) を使用して (UMASK = 1), 類似した ID を持つリモートフレームのグループの TXRQST ビットをセットすることができます。詳細については「22.9 CAN アプリケーション - メッセージの送信」を参照してください。操作には注意が必要です。DIR ビットをマスクしないように注意する必要があります。

送信オブジェクトの更新

CPU は , IFx インタフェースレジスタを通じて , 送信オブジェクトのデータバイトをいつでも更新することができますが , 更新前に MSGVAL も TXRQST もリセットされません。

＜注意事項＞

優先度の最も低いメッセージバッファが送信に使用される場合 , TXRQST を "0" に設定すると , TXRQST を "1" に再設定したときに送信が遅延することがあります。「22.8.2 送信リクエストレジスタ (TREQR1n, TREQR2n)」の＜注意事項＞を参照してください。

データバイトの一部のみを更新する場合でも , 対応する IFx データ A レジスタまたは IFx データ B レジスタの 4 バイトすべてを有効にしてから , そのレジスタの内容をメッセージオブジェクトに転送する必要があります。CPU が IFx データレジスタに 4 バイトすべてを書き込むか , CPU が新しいデータバイトを書き込む前に , メッセージオブジェクトを IFx データレジスタに転送する必要があります。
データバイト (8 バイト) のみを更新する場合は , まず 0087_H をコマンドマスクレジスタに書込み , 次にメッセージオブジェクトの番号をコマンドリクエストレジスタに書き込み , データバイトの更新と TXRQST のセットを同時に実行します。
データの更新時に , 既に実行中の送信の終わりに TXRQST がリセットされるのを防ぐために , TXRQST とともに NEWDAT をセットする必要があります。詳細については「22.9 CAN アプリケーション - メッセージの送信」を参照してください。TXRQST とともに NEWDAT をセットすると , 新たに送信が開始されると直ちに NEWDAT はリセットされます。

受信オブジェクトの構成

Figure 22-24 に受信オブジェクトを初期化する方法を示します。

Figure 22-24. 受信オブジェクトの初期化

MSGVAL	ARB	DATA	MASK	EOB	DIR	NEWDAT	MSGLST	RXIE	TXIE	INTPND	RMTEN	TXRQST
1	appl.	appl.	appl.	1	0	0	0	appl.	0	0	0	0

アービトレーションレジスタ (ID28 ～ ID0 および XTD ビット) はアプリケーションによって設定します。このレジスタで受信したメッセージの ID と種類を定義します。11 ビット ID(標準フレーム) を使用する場合は ID28 ～ ID18 を使用し , ID17 ～ ID0 は無視することができます。11 ビット ID のデータフレームを受信したときは ID17 ～ ID0 は "0" にセットされます。

RXIE ビットをセットした場合は , 受信したデータフレームがメッセージオブジェクトに格納されると , INTPND ビットがセットされます。

データ長コード (DLC3 ～ DLC0) はアプリケーションによって設定します。メッセージハンドラによるデータフレームのメッセージオブジェクトへの格納では, 受信したデータ長コードと8データバイトが格納されます。データ長コードが 8 未満の場合は , メッセージオブジェクトの残りのバイトは不定値で上書きされます。

マスクレジスタ (MSK28 ～ MSK0, UMASK, MXTD, MDIR の各ビット) を使用して (UMASK = 1), 類似した ID を持つデータフレームのグループを受け入れることができます。詳細については「22.9 CAN アプリケーション - データフレームの受信」を参照してください。通常のアプリケーションでは DIR ビットをマスクしないように注意します。

受信したメッセージの処理

CPU は IFx インタフェースレジスタを通じて受信したメッセージをいつでも読み出すことができ , データの一貫性はメッセージハンドラステートマシンによって保証されます。

通常は , CPU によってまずコマンドマスクレジスタに $007F_H$ を書き込み , 次にメッセージオブジェクトの番号をコマンドリクエストレジスタに書き込みます。この組合せにより受信したメッセージの全体がメッセージ RAM からメッセージバッファレジスタに転送されます。また , メッセージ RAM (メッセージバッファではなく) の NEWDAT ビットと INTPND ビットがクリアされます。

メッセージオブジェクトがアクセプタンスフィルタリング用のマスクを使用している場合は, アービトレーションビットは一致したメッセージのどちらを受信したかを示します。

NEWDAT の実際の値は , 前回このメッセージオブジェクトを読み出してから新たにメッセージを受信したかどうかを示します。MSGLST の実際の値は , 前回このメッセージオブジェクトを読み出してから複数のメッセージを受信したかどうかを示します。MSGLST は自動的にはリセットされません。

CPU は , リモートフレームを使用して , ほかの CAN ノードに受信オブジェクト用の新しいデータを送信するようにリクエストすることができます。受信オブジェクトの TXRQST ビットをセットすると , 受信オブジェクトの ID を持つリモートフレームが送信されます。このリモートフレームは , ほかのノードによる一致するデータフレームの送信開始のトリガになります。リモートフレームを送信することができるようになる前に一致するデータフレームを受信した場合は , TXRQST ビットが自動的にリセットされます。

FIFOバッファの構成

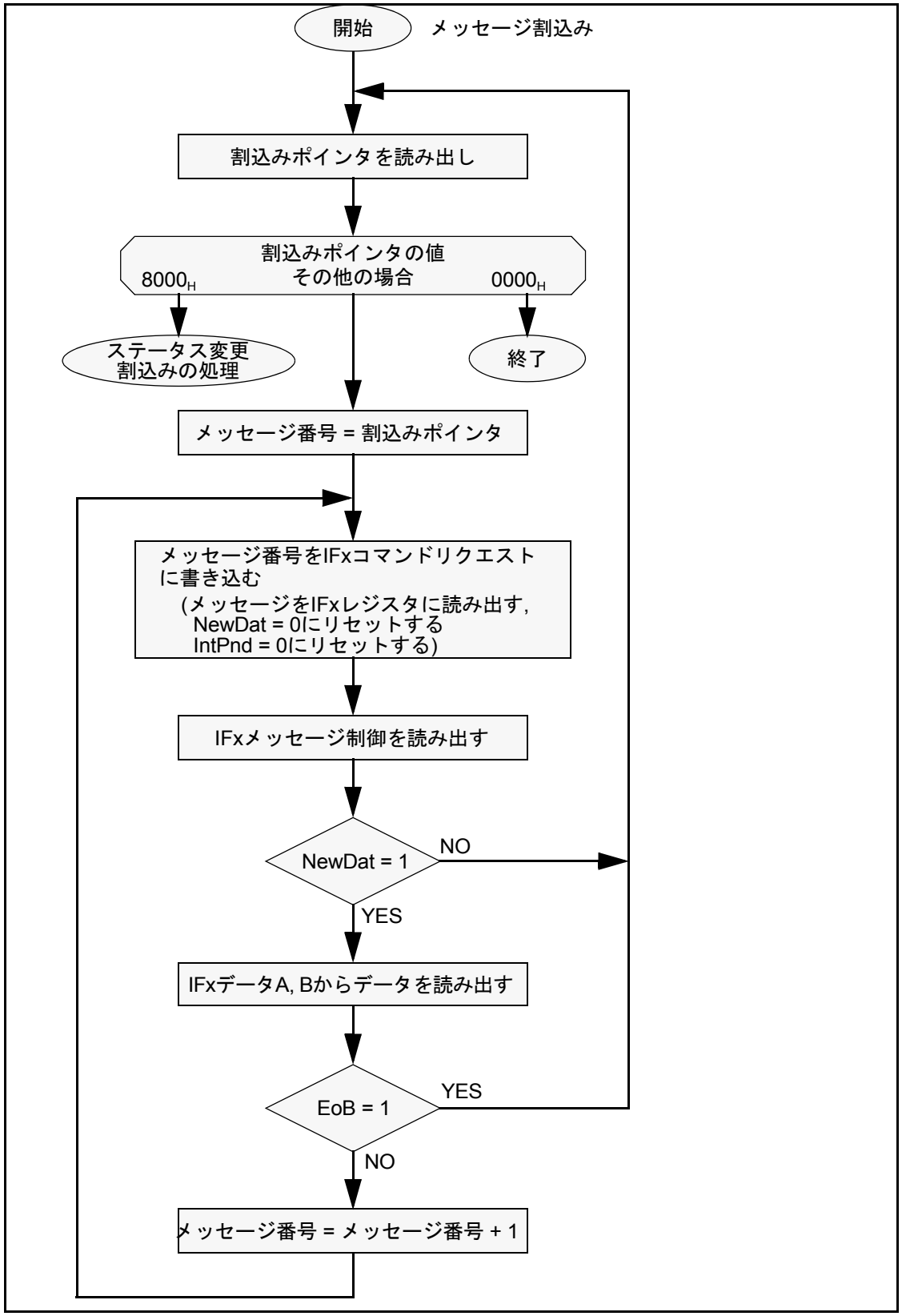
FIFO バッファに属している受信オブジェクトの構成は , EOB ビットを除き , (単一の) 受信オブジェクトと同じです。詳細については「22.9 CAN アプリケーション - 受信オブジェクトの構成」を参照してください。

複数のメッセージオブジェクトを連結して FIFO バッファに格納するには , それらのメッセージオブジェクトの ID およびマスク (使用する場合) が適合する値にプログラミングされている必要があります。メッセージオブジェクトの明示的な優先度により , 最も小さな番号のメッセージオブジェクトが FIFO バッファの最初のメッセージオブジェクトになります。FIFO バッファのすべてのメッセージオブジェクトの EOB ビットは , 最後のものを除き , "0" にプログラムする必要があります。FIFO バッファの最後のメッセージオブジェクトの EOB ビットを "1" にセットし , ブロックの最後として設定します。

FIFOバッファからの読出し

CPU がメッセージオブジェクトの番号を IFx コマンドリクエストレジスタに書き込むことによってメッセージオブジェクトの内容を IFx メッセージバッファレジスタに転送するとき , 対応するコマンドマスクレジスタで NEWDAT ビットと INTPND ビットを "0" にリセットするようにプログラミングする必要があります (TXRQST =1 および CIP =1)。メッセージ制御レジスタ内の上記のビットの値は , 常にこれらのビットをリセットする前の状態を反映します。
FIFO バッファが確実に正しく機能するように , CPU は , 最も小さなメッセージ番号を持つメッセージオブジェクトを FIFO オブジェクトの先頭から読み出す必要があります。Figure 22-25 に FIFO バッファに連結して格納されているメッセージオブジェクトのセットを CPU で処理する方法を示します。

Figure 22-25. CPUによるFIFOバッファの処理



割込みの処理

保留中の割込みが複数存在する場合, CAN割込みレジスタは, 発生順序とは無関係に最も優先度の高い保留中の割込みを指しています。割込みは CPU によってクリアされるまで保留状態を維持します。

優先度が最も高い割込みはステータス割込みです。メッセージ割込みの中では , メッセージ番号が大きいものほどメッセージオブジェクトの割込み優先順位が下がります。

メッセージ割込みをクリアするにはメッセージオブジェクトの INTPND ビットをクリアします。ステータスレジスタにリードアクセスすると , ステータス割込みがクリアされます。

割込みレジスタ INTRn の割込み ID INTID は割込みの要因を示します。保留中の割込みが存在しない場合は , レジスタが 0 の値を保持します。割込みレジスタの値が 0 でない場合は , 保留中の割込みが存在し , IE が設定されると CPU への割込みラインがアクティブになります。割込みラインは割込みレジスタが 0 の値に戻るか (割込み要因のリセット), IE がリセットされるまでアクティブな状態を維持します。

8000_H の値は , CAN コアがステータスレジスタ (エラー割込みまたはステータス割込み) を更新 (必ずしも変更ではない) したために割込みが保留されていることを示します。この割込みは最上位の優先度をもっています。CPU はステータスビット RXOK, TXOK, および LEC を更新 (リセット) することができますが , 割込みの生成およびリセットを CPU によるステータスレジスタへのライトアクセスによって行うことはできません。

他の値はすべて , 割込みの割込み要因がメッセージオブジェクトの 1 つであることを示し , INTID は最上位の割込み優先度を持つ保留中のメッセージを指します。

CPU は , ステータスレジスタの変化により割込みを生成するかどうか (CAN 制御レジスタのビット EIE および SIE), および割込みレジスタが "0" でないときに割込みラインをアクティブにするかどうか (CAN 制御レジスタのビット IE) を制御します。IE がリセットされているときでも割込みレジスタは更新されます。

CPU は 2 つの場合にメッセージ割込みの割込み要因に従います。まず割込みレジスタ内の INTID に従うことができ , 次に割込み保留レジスタをポーリングすることができます (詳細については「22.8 メッセージハンドラのレジスタ」を参照)。

割込みの割込み要因であるメッセージを読み出す割込みサービスルーチンはメッセージを読み出して , 同時にメッセージオブジェクトの INTPND をリセットすることができます (コマンドマスクレジスタ IFxCMSKn の CIP ビット)。INTPND がクリアされると , 割込みレジスタは割込みが保留されている次のメッセージオブジェクトを指します。

ビットタイムおよびビットレート

CAN は , 下は 1 kbps 未満から上は最大 1000 kbps までの範囲のビットレートに対応しています。CAN ネットワークの各メンバは専用のクロックジェネレータ (通常は水晶発振器) を持っています。ビットタイム (ビットレートの逆数) のタイミングパラメータを各 CAN ノードごとに設定して , CAN ノードの発振周期 (fosc) が異なる場合でも共通のビットレートを生成することができます。

これらの発振器の周波数は絶対的に安定したものではなく, 温度や電圧の変化およびコンポーネントの劣化などによってわずかに変動します。変動幅が所定の発振器の許容誤差 (df) 内に収まっていれば CAN ノードは , ビットストリームへの再同期を行うことによってビットレートの変動を補正することができます。

CAN の仕様ではビットタイムは 4 つのセグメントに分れています (Figure 22-26 参照)。同期セグメント , 伝播時間セグメント , 位相バッファセグメント 1, 位相バッファセグメント 2 の 4 つのセグメントのそれぞれは , プログラム可能な所定の基本単位時間数で構成されています。基本単位時間の長さ (tq) は , ビットタイムの基本単位ですが , CAN コントローラのシステムクロック fsys とボーレートプリスケーラ (BRP) によって tq = BRP / fsys と定

義されます。CAN のシステムクロック fsys は CAN_CLK 入力の周波数です。

同期セグメント Sync_Seg はビットタイムの CAN バスレベルのエッジが発生する部分で , Sync_Seg の外側で発生するエッジと Sync_Seg 間の距離をそのエッジの位相誤差とよびます。伝播時間セグメント Prop_Seg は , CAN ネットワーク内の物理的な遅延時間を補正するためのものです。位相バッファセグメント Phase_Seg1 と Phase_Seg2 はサンプルポイントを挟んでいます。(再) 同期ジャンプ幅 (SJW) は , エッジの位相誤差を補正するために, 位相バッファセグメントで定義される限界内でサンプルポイントを再同期によって移動する距離を示します。

Figure 22-26. ビットタイミング

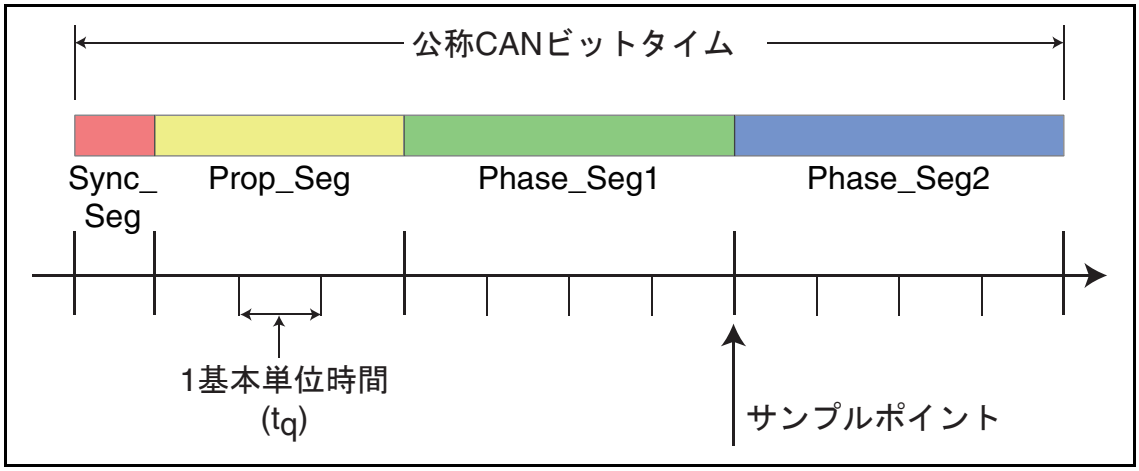


Table 22-8. CANビットタイムのパラメータ

パラメータ	範囲	説明
BRP	[1～32]	基本単位時間tqの長さを定義します。
Sync_Seg	1 tq	固定長でシステムクロックに同期します。
Prop_Seg	[1～8] tq	物理的な遅延時間を補正します。
Phase_Seg1	[1～8] tq	同期によって一時的に延長されることがあります。
Phase_Seg2	[1～8] tq	同期によって一時的に短縮されることがあります。
SJW	[1～4] tq	どちらの位相バッファセグメントよりも長くなることはできません。

＜注意事項＞

この表は , CAN プロトコルに必要な最小のプログラム可能な範囲を示したものです。

本章の前半部で説明されている BTRn レジスタフィールドについては、以下に設定を示します。

Prop_Seg + Phase_Seg1 = TSEG1 値 + 1

Phase_Seg2 = TSEG2 値 + 1

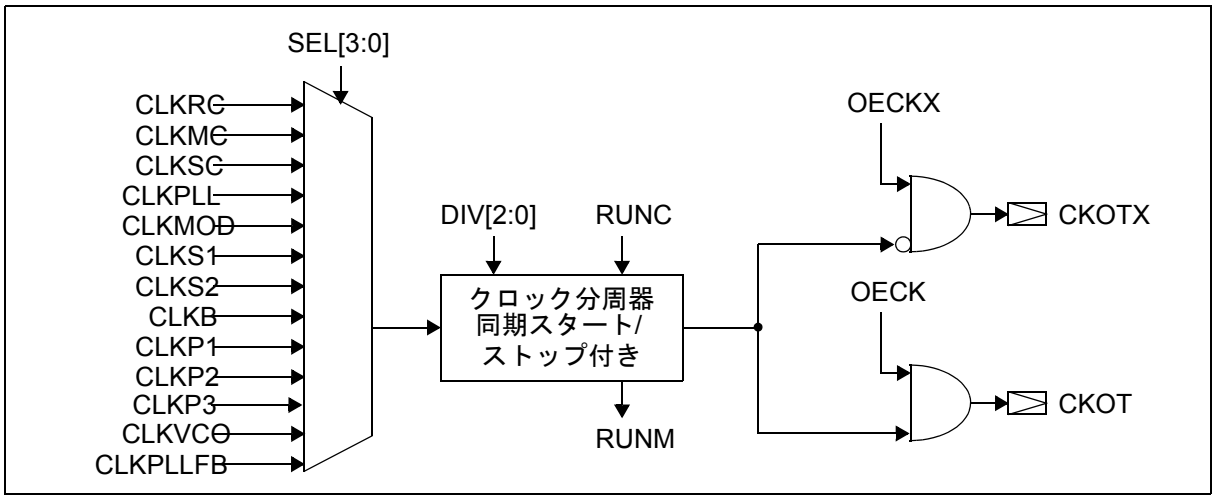
23. クロック出力機能

クロック出力機能とその動作を説明します。

23.1 クロック出力機能の概要

クロック出力機能を使用して, 最大2系統の内部クロックを4つのクロック出力端子(CKOT0/CKOTX0およびCKOT1/CKOTX1)に出力することができます(各クロックの通常出力と反転出力)。クロックは直接出力することも内部クロック分周器で分周して出力することもできます。

クロック出力機能のブロック図(1チャネル)



選択可能なクロックの詳細については「第 6 章 クロック」を参照してください。

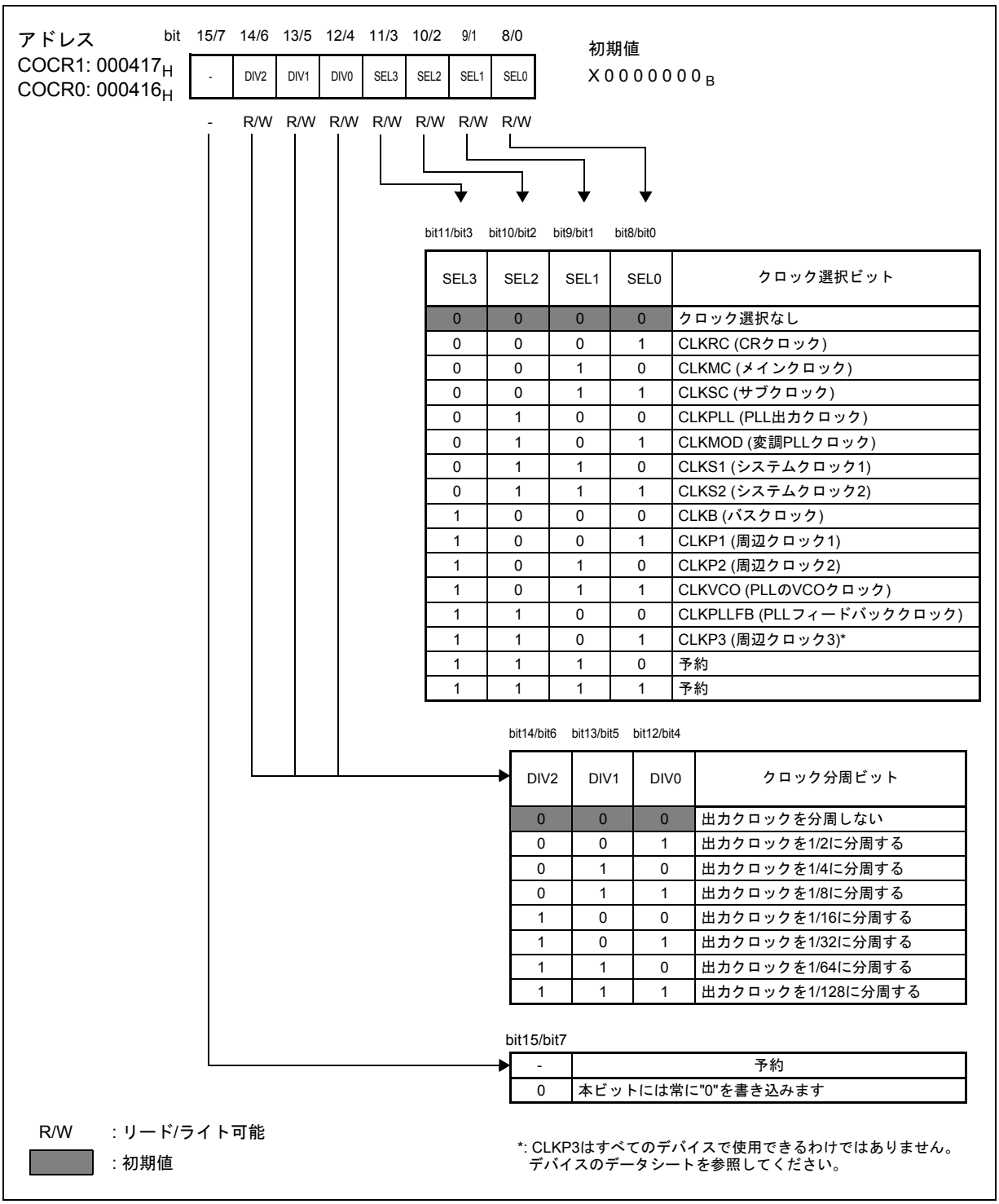
CLKP3 はすべてのデバイスで使用できるわけではありません。デバイスのデータシートを参照してください。

23.2 クロック出力の構成と起動レジスタ

クロック出力構成レジスタ(COCR0, COCR1)を使用して, 出力クロックの選択とクロック分周器の制御を行います。COCR0レジスタはCKOT0/CKOTX0クロック出力を, COCR1レジスタはCKOT1/CKOTX1クロック出力を制御します。
クロック出力起動レジスタ(COAR)はCKOT0, CKOTX0, CKOT1, CKOTX1の各出力端子の許可/禁止と, 選択したクロックの同期スタート/ストップを制御します。

クロック出力構成レジスタ(COCR)

Figure 23-1. クロック出力構成レジスタ(COCR)



SEL3	SEL2	SEL1	SEL0	クロック選択ビット
0	0	0	0	クロック選択なし
0	0	0	1	CLKRC (CRクロック)
0	0	1	0	CLKMC (メインクロック)
0	0	1	1	CLKSC (サブクロック)
0	1	0	0	CLKPLL (PLL出力クロック)
0	1	0	1	CLKMOD (変調PLLクロック)
0	1	1	0	CLKS1 (システムクロック1)
0	1	1	1	CLKS2 (システムクロック2)
1	0	0	0	CLKB (バスクロック)
1	0	0	1	CLKP1 (周辺クロック1)
1	0	1	0	CLKP2 (周辺クロック2)
1	0	1	1	CLKVCO (PLLのVCOクロック)
1	1	0	0	CLKPLLFB (PLLフィードバッククロック)
1	1	0	1	CLKP3 (周辺クロック3)*
1	1	1	0	予約
1	1	1	1	予約

DIV2	DIV1	DIV0	クロック分周ビット
0	0	0	出力クロックを分周しない
0	0	1	出力クロックを1/2に分周する
0	1	0	出力クロックを1/4に分周する
0	1	1	出力クロックを1/8に分周する
1	0	0	出力クロックを1/16に分周する
1	0	1	出力クロックを1/32に分周する
1	1	0	出力クロックを1/64に分周する
1	1	1	出力クロックを1/128に分周する

-	予約
0	本ビットには常に"0"を書き込みます

Table 23-1. クロック出力構成レジスタ(COCR0/1)の各ビットの機能

<table>
<tr><th colspan="2">ビット名</th><th>機能</th></tr>
<tr><td>bit 0～bit 3 /
bit 8～bit 11</td><td>SEL3 ～ SEL0:
クロック選択
ビット</td><td>
• CKOT0/CKOT1端子に出力するクロックを選択するビットです(下表参照)。
<table>
<tr><th colspan="4">bit11/bit3 bit10/bit2 bit9/bit1 bit8/bit0</th><th></th></tr>
<tr><th>SEL3</th><th>SEL2</th><th>SEL1</th><th>SEL0</th><th>クロック選択ビット</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>クロック選択なし</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>CLKRC (CRクロック)</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>CLKMC (メインクロック)</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>CLKSC (サブクロック)</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>CLKPLL (PLL出力クロック)</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>CLKMOD (変調PLLクロック)</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>CLKS1 (システムクロック1)</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>CLKS2 (システムクロック2)</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>CLKB (バスクロック)</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>CLKP1 (周辺クロック1)</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>CLKP2 (周辺クロック2)</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>CLKVCO (PLLのVCOクロック)</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>CLKPLLFB (PLLフィードバッククロック)</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>CLKP3 (周辺クロック3)*</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>予約</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>予約</td></tr>
</table>
リードアクセスすると , 本制御ビットの設定値が読み出されます。
• リセットするとこれらのビットは "0000" (クロック選択なし) になります。
• クロック出力機能がアクティブのとき (対応する COAR レジスタの RUNC ビットが "1", CKOE/CKOXE が "1") に本設定を変更すると , スパイク (不正なクロック波形) が出力されることがあります。
*: CLKP3はすべてのデバイスで使用できるわけではありません。デバイスのデータシートを参照してください。
</td></tr>
<tr><td>bit 4～bit 6/
bit 12～bit 14</td><td>DIV2 ～ DIV0:
クロック分周
ビット</td><td>
• クロック出力分周器の設定を行うビットです(下表参照)。
<table>
<tr><th colspan="3">bit14/bit6 bit13/bit5 bit12/bit4</th><th></th></tr>
<tr><th>DIV2</th><th>DIV1</th><th>DIV0</th><th>クロック分周ビット</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>出力クロックを分周しない</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>出力クロックを1/2に分周する</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>出力クロックを1/4に分周する</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>出力クロックを1/8に分周する</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>出力クロックを1/16に分周する</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>出力クロックを1/32に分周する</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>出力クロックを1/64に分周する</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>出力クロックを1/128に分周する</td></tr>
</table>
リードアクセスすると , 本制御ビットの設定値が読み出されます。
• リセットするとこれらのビットは "000" (出力クロックを分周しない) になります。
• クロック出力機能がアクティブのとき(対応するCAORレジスタのRUNCビットが"1", CKOE/CKOXEが"1")に本設定を変更すると, スパイク(不正なクロック波形)が出力されることがあります。
</td></tr>
<tr><td>bit 7, bit 15</td><td>予約</td><td>
• 本ビットには常に "0" を書き込みます。
• 本ビットの読出し値は不定です。
• リードモディファイライト系命令は無効です。
</td></tr>
</table>

クロック出力起動レジスタ(COAR)

Figure 23-2. クロック出力起動レジスタ(COAR)

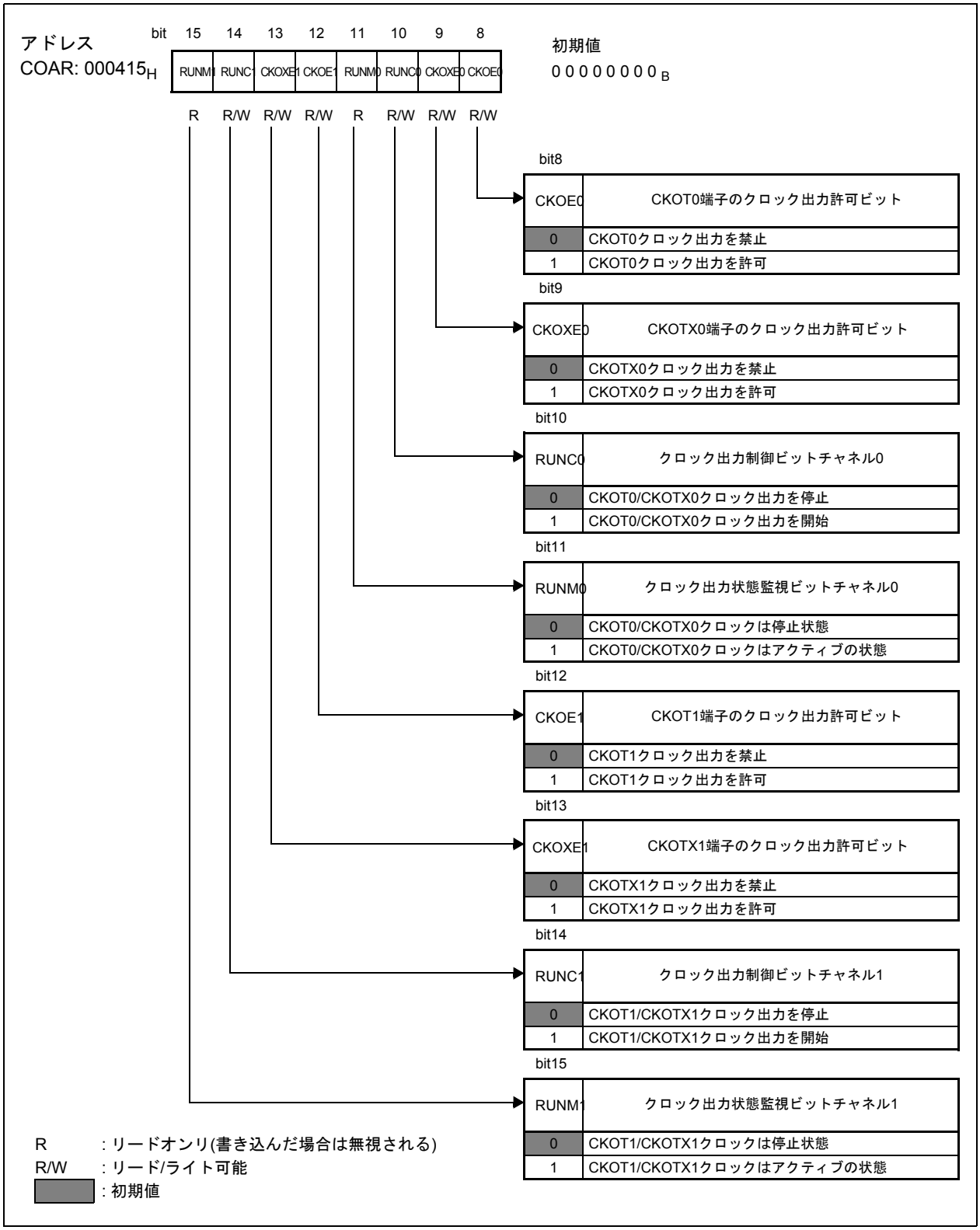


Table 23-2. クロック出力起動レジスタ(COAR)の各ビットの機能 (Sheet 1 of 2)

ビット名		機能
bit 8	CKOE0: CKOT0端子の クロック出力許可ビット	• クロック CKOT0 の出力を制御します。 • 本ビットに "1" を書き込むと CKOT0 出力を有効にすることができます。本ビットを "1" にセットしたときは , CKOT0 端子を共有しているほかのリソース出力を許可にしないでください。 • 本ビットを "0" にセットするとクロック出力が禁止されます (初期値)。 • リードアクセスすると本ビットの設定値が読み出されます。 • RUNM0ビットが"1"のときに本ビットの設定を変更すると,不正なクロック波形が出力されることがあります。
bit 9	CKOXE0: CKOTX0端子の クロック出力許可ビット	• 反転クロック CKOTX0 の出力を制御します。 • 本ビットに "1" を書き込むと CKOTX0 出力を有効にすることができます。本ビットを "1" にセットしたときは , CKOTX0 端子を共有しているほかのリソース出力を許可にしないでください。 • 本ビットを "0" にセットするとクロック出力が禁止されます (初期値)。 • リードアクセスすると本ビットの設定値が読み出されます。 • RUNM0ビットが"1"のときに本ビットの設定を変更すると,不正なクロック波形が出力されることがあります。
bit 10	RUNC0: クロック出力制御ビットチャネル0	• CKOT0/CKOTX0 クロック出力の同期スタート / ストップを制御します。 • 本ビットを "1" にセットするとクロック出力が始まり , "0" にセットするとクロック出力が停止します。 • リセットすると本ビットは "0" に初期化されます (クロック出力停止)。 • 選択したクロックをCKOT0/CKOTX0端子に出力するには, RUNC0 ビットと CKOE0/CKOXE0 ビットを "1" にセットする必要があります。 • RUNC0 ビットへの書込みはいつでも行うことができます。ただし , クロックの出力開始または停止には数サイクルのクロックサイクルが必要です。また , SEL3 ～ SEL0 で選択したクロックが有効なときでなければ実行できません。 • リードアクセスすると本制御ビットの設定値が読み出されます。同期スタート/ストップが完了しているかどうかをチェックするにはRUNM0ビットを使用します。
bit 11	RUNM0: クロック出力状態監視ビットチャネル0	• 本ビットを読み出すとスタート / ストップ同期回路の状態が読み出され , クロック出力のスタート / ストップが完了しているかどうかを確認することができます。 • "1" が読み出された場合は CKOT0/CKOTX0 クロック出力が開始されていることを示します。 • "0" が読み出された場合は CKOT0/CKOTX0 クロック出力が停止されていることを示します。この場合には , CKOT0 端子の出力は L レベル , CKOTX0 端子の出力は H レベルとなります。 • 本ビットへの書込みは無視されます。
bit 12	CKOE1: CKOT1端子の クロック出力 許可ビット	• クロック CKOT1 の出力を制御します。 • 本ビットに "1" を書き込むと CKOT1 出力を許可することができます。本ビットを "1" にセットしたときは , CKOT1 端子を共有しているほかのリソース出力を許可にしないでください。 • 本ビットを "0" にセットするとクロック出力が禁止されます (初期値)。 • リードアクセスすると本ビットの設定値が読み出されます。 • RUNM1ビットが"1"のときに本ビットの設定を変更すると,不正なクロック波形が出力されることがあります。
bit 13	CKOXE1: CKOTX1端子の クロック出力 許可ビット	• 反転クロック CKOTX1 の出力を制御します。 • 本ビットに "1" を書き込むと CKOTX1 出力を許可することができます。本ビットを "1" にセットしたときは , CKOTX1 端子を共有しているほかのリソース出力を許可にしないでください。 • 本ビットを "0" にセットするとクロック出力が禁止されます (初期値)。 • リードアクセスすると本ビットの設定値が読み出されます。 • RUNM1ビットが"1"のときに本ビットの設定を変更すると, 不正なクロック波形が出力されることがあります。

Table 23-2. クロック出力起動レジスタ(COAR)の各ビットの機能 (Sheet 2 of 2)

ビット名		機能
bit 14	RUNC1: クロック出力制御ビットチャネル1	• CKOT1/CKOTX1 クロック出力の同期スタート / ストップを制御します。 • 本ビットを "1" にセットするとクロック出力が始まり , "0" にセットするとクロック出力が停止します。 • リセットすると本ビットは "0" に初期化されます (クロック出力停止)。 • 選択したクロックをCKOT1/CKOTX1端子に出力するには, RUNC1 ビットとCKOE1/CKOXE1 ビットを "1" にセットする必要があります。 • RUNC1 ビットへの書込みはいつでも行うことができます。ただし , クロックの出力開始または停止には数サイクルのクロックサイクルが必要です。また , SEL3 ～ SEL0 で選択したクロックが有効なときでなければ実行できません。 • リードアクセスすると本制御ビットの設定値が読み出されます。同期スタート/ストップが完了しているかどうかをチェックするにはRUNM1ビットを使用します。
bit 15	RUNM1: クロック出力 状態監視ビットチャネル1	• 本ビットを読み出すとスタート / ストップ同期回路の状態が読み出され , クロックのスタート / ストップが完了しているかどうかを確認することができます。 • "1" が読み出された場合は CKOT1/CKOTX1 クロック出力が開始されていることを示します。 • "0"が読み出された場合はCKOT1/CKOTX1クロック出力が停止されていることを示します(CKOT1 端子の出力は "L" レベル , CKOTX1 端子の出力は "H" レベル)。 • 本ビットへの書込みは無視されます。

23.3 クロック出力機能の動作

クロック出力機能を使用して, CKOT0/CKOTX0およびCKOT1/CKOTX1出力端子に独立した2系統のクロックを出力することができます。2つのクロックを同期して起動または停止することができます。

クロック出力機能の起動

CKOT0/CKOTX0 および CKOT1/CKOTX1 端子のクロック出力機能の起動は次のようにして行います。

- 必要なクロックと分周値を選択して,対応するCOCR0,1レジスタの設定を行います。
- COARレジスタの,対応するクロック出力許可ビットCKOE0,CKOE1/CKOXE0,CKOXE1を"1"にセットすることによりCKOT0,CKOT1/CKOTX0,CKOTX1出力端子を許可します。これと同時または以後にスタートビットRUNC0,1を"1"にセットします。

クロック出力を許可にすると , CKOT0, CKOT1 端子には "0" が , CKOTX0, CKOTX1 端子には "1" が出力されます。

RUNC0,1 スタートビットを "1" にセットすると , 同期回路がアクティブになり , 対応する CKOT0, CKOT1 / CKOTX0, CKOTX1 端子から選択したクロックの出力が開始されます。RUNM1, RUNM0 ビットの読出し値が "0" のときはクロックの出力がまだ開始されていないこと (同期メカニズムが動作中) を示し , 読出し値が "1" のときはクロックの出力が始まっていることを示します。

クロック出力機能の禁止

クロック出力機能の禁止は次のようにして行います。

- RUNC0,1ビットを"0"にセットしてクロック出力を停止します。クロック出力の停止は同期回路を使用して実行されます。RUNM0,1ビット("0"にセットされる)を読み出すことにより, 実際に停止しているかどうかを確認してください。
- クロックを停止すると, COCRレジスタの設定を変更したり,クロック出力許可ビットCKOE0, CKOE1/ CKOXE0, CKOXE1を"0"にセットしてクロック出力を禁止したりすることができます。

クロックが停止すると , CKOT0, CKOT1 / CKOTX0, CKOTX1 端子の出力値が , CKOT0, CKOT1 端子は "0" に , CKOTX0, CKOTX1 端子は "1" に戻ります。

クロック出力の開始 / 停止用同期回路には , 有効なソースクロックが必要です。SEL3 ～ SEL0 ビットで選択したソースクロックが無効な場合には , 出力クロックは動作しません。また , RUNC0,1 ビットに値をセットすることは可能ですが , RUNM 監視ビットの値は変化しません。

クロック出力構成レジスタの内容の変更

COCR0,1レジスタの書込みは, 対応するクロック出力が停止した(RUNM0,1=0)ときにのみ実行してください。クロック出力が動作中にCOCRレジスタへ書込みを行った場合には,不正なクロック波形が出力されることがあります。

24. リアルタイムクロック

リアルタイムクロックの機能と動作を説明します。

24.1 リアルタイムクロックの概要

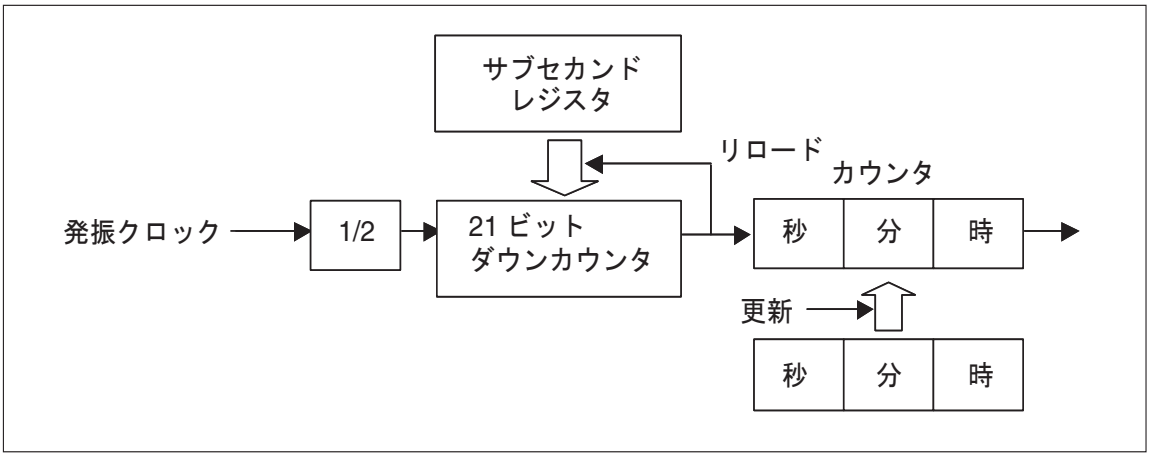
リアルタイムクロックは, タイマ制御レジスタ, サブセカンドレジスタ, 秒/分/時レジスタ, 1/2クロック分周器, 21ビットプリスケーラ, 時/分/秒カウンタで構成されています。
リアルタイムクロック(RTC)は時計モードでも経過時間のカウントを続行し, メイン発振(4MHz), サブ発振(32kHz), またはCR発振(~100kHz/~2MHz)を基にして現在の実時間(HH/MM/SS)を提供します。これにより, スタンバイ中も割込みからの復帰なしに正確な時間を計測することができます。

リアルタイムクロックの特長

- 機能：計時(HH/MM/SS)。(このクロックは時計モード中も動作を継続します。)
- メイン発振 (4MHz), サブ発振 (32kHz), またはCR発振 (~100kHzまたは~2MHz)で動作
- タイムユニット：2分周クロックを選択
- 動作クロック
  - レジスタアクセス用：CLKP1
  - タイムカウント用：メインクロック(CLKMC), サブクロック(CLKSC), CRクロック(CLKRC)
- 時刻：初期設定および調整可能
- 割込み：割込みは, 1/2秒, 1秒, 1分, 1時間, 1日の5種類の周期で任意に生成できます。
- その他：サブセカンドレジスタの値を変更することにより, 任意の周期で割込みを生成することができます。

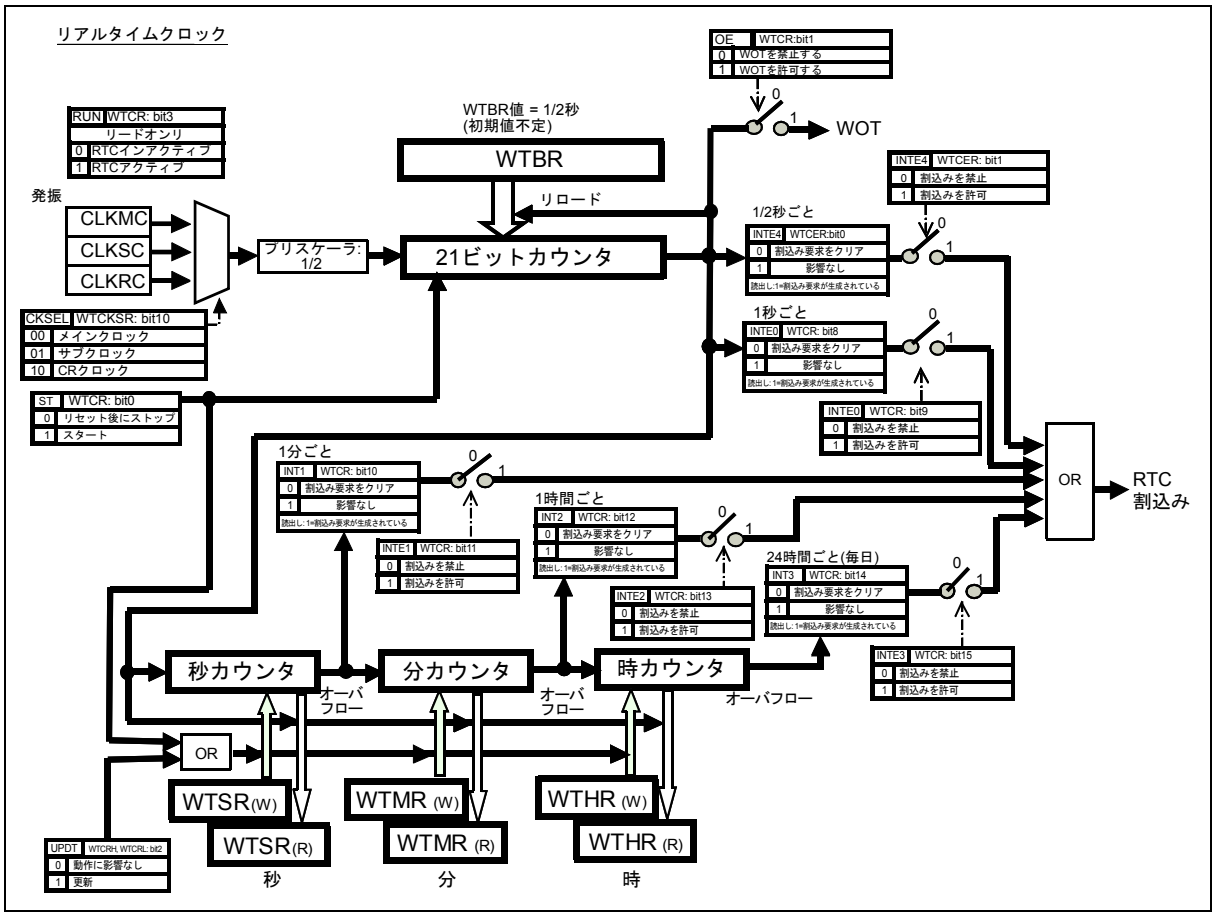
リアルタイムクロックのブロックダイヤグラム

Figure 24-1. リアルタイムクロックのブロックダイヤグラム



構成ダイヤグラム

Figure 24-2. 構成ダイヤグラム



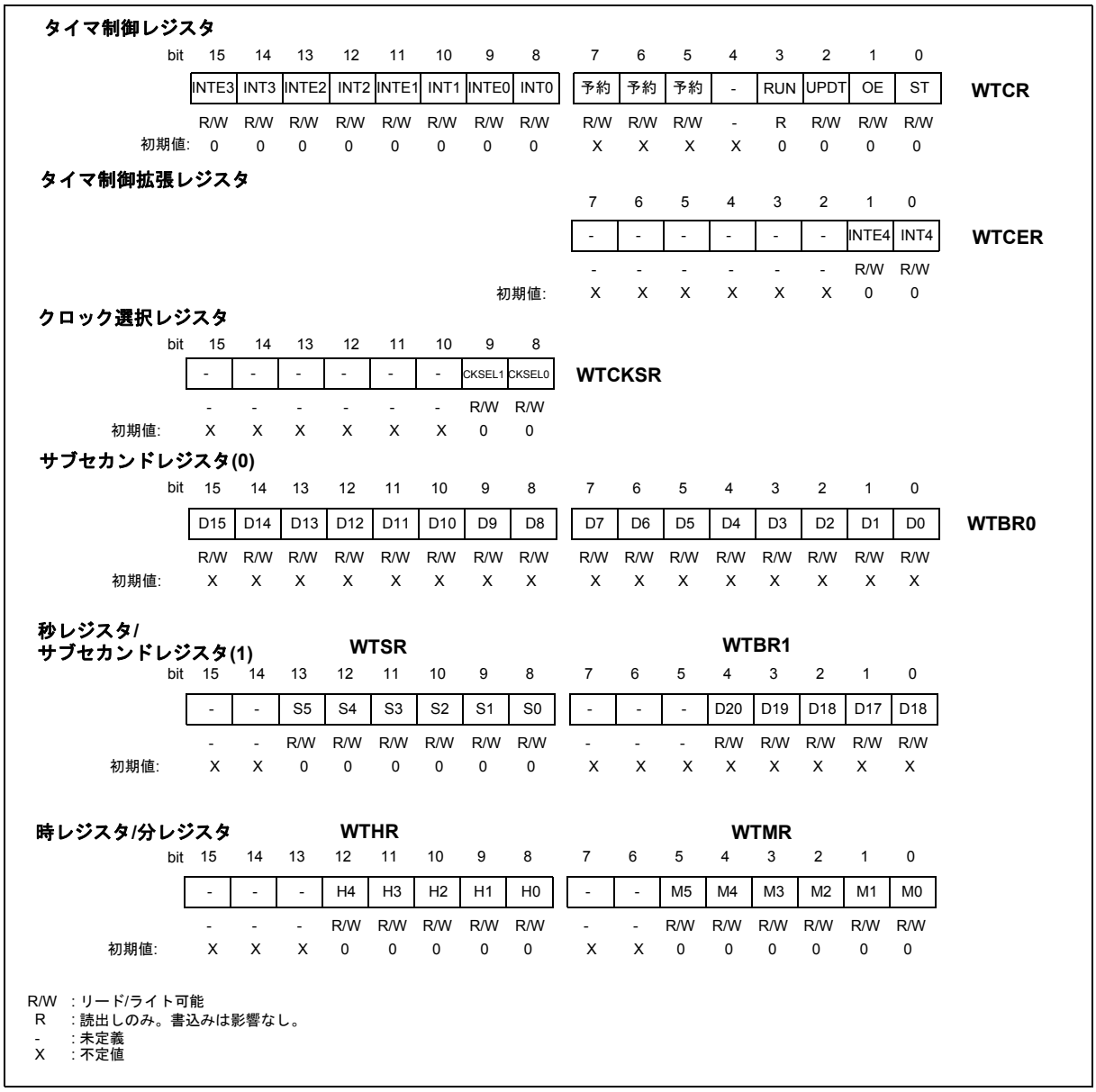
24.2 リアルタイムクロックのレジスタ

リアルタイムクロックには次の 7 つのレジスタがあります。

- タイマ制御レジスタ(WTCR)
- タイマ制御拡張レジスタ(WTCER)
- クロック選択レジスタ(WTCKSR)
- サブセカンドレジスタ(WTBR)
- 秒レジスタ(WTSR)
- 分レジスタ(WTMR)
- 時レジスタ(WTHR)

リアルタイムクロックのレジスタ

Figure 24-3. リアルタイムクロックのレジスタ



＜注意事項＞

WTCR, WTCER, および WTCKSR の未使用および予約ビットを除くビットは , すべてのリセット要因によって初期化されます。WTBR は , リセットされません。

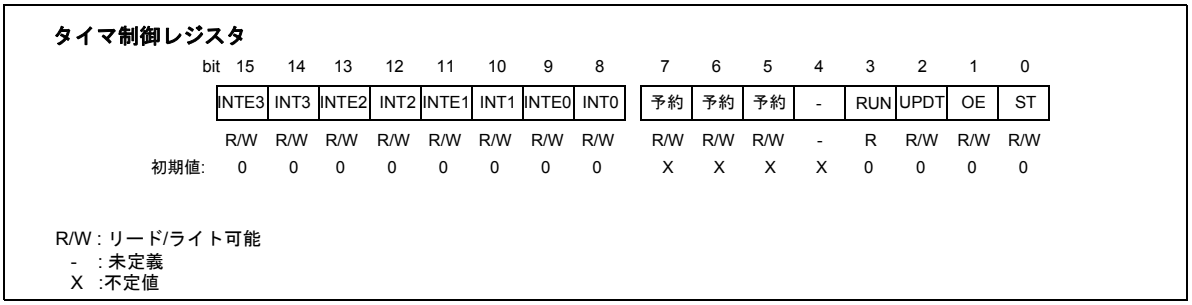
WTSR, WTMR, WTHR は , パワーオンリセット , 低電圧リセット , 外部リセットによってのみ初期化され , ほかのリセットでは初期化されません。

24.2.1 タイマ制御レジスタ

タイマ制御レジスタは , リアルタイムクロックのスタート / ストップ , 割込みの制御 , 外部出力端子の設定に使用します。

24.2.1.1 タイマ制御レジスタ(WTCR)

Figure 24-4. タイマ制御レジスタ(WTCR)



＜注意事項＞

WTCR の予約および未使用ビットを除くビットは , すべてのリセット要因によって初期化されます。

bit15: 1日割込み要求許可

時カウンタがオーバフローすると, このフラグは"1"になります。.

INTE3	動作
0	割込み要求禁止
1	1日(24時間)割込み要求許可

bit14: 1日割込み要求フラグ

INT3	ステータス	
	読出し	書込み
0	割込み要求なし	フラグをクリアする
1	1日(24時間)割込み要求あり	書込みは動作に影響しません

bit13: 1時間割込み要求許可

分カウンタがオーバフローすると, このフラグは"1"になります。

INTE2	動作
0	割込み要求禁止
1	1時間割込み要求許可

bit12: 1時間割込み要求フラグ

INT2	ステータス	
	読出し	書込み
0	割込み要求なし	フラグをクリアする。
1	1時間割込み要求あり	書込みは動作に影響しません。

bit11: 1分間割込み要求許可

秒カウンタがオーバフローすると, このフラグは"1"になります。

INTE1	動作
0	割込み要求禁止
1	1分間割込み要求を許可

bit10: 1分間割込み要求フラグ

INT1	動作	
	読出し	書込み
0	割込み要求なし	フラグをクリアする。
1	1分間割込み要求あり	書込みは動作に影響しません。

bit9: 1秒間割込み要求許可

21ビットダウンカウンタを"0"にセットすると, このフラグは"1"になります。

INTE0	動作
0	割込み要求なし
1	1秒間割込み要求許可

bit8: 1秒間割込み要求フラグ

INT0	ステータス	
	読出し	書込み
0	割込み要求なし	フラグをクリアする。
1	1秒間割込み要求あり	書込みは動作に影響しません。

bit7～bit5: 予約

常に "0" を書き込んでください。読出し値は不定です。リードモディファイライト系命令は無効です。

bit4: 未定義

常に "0" を書き込んでください。読出し値は不定です。リードモディファイライト系命令は無効です。

bit3: 動作ステータス

RUN	動作ステータス
0	リアルタイムクロックモジュールは停止
1	リアルタイムクロックモジュールは動作中

bit2: 更新

更新ビット(UPDT)に"1"を書き込む前に, 時/分/秒の各カウンタを更新するための値を時/分/秒の各レジスタにセットする必要があります。21ビットダウンカウンタへのリロードを行うと直ちに時/分/秒の各レジスタが更新されます。

UPDT	ステータス/動作
0	更新完了("0"の書込みは動作に影響しません)
1	時/分/秒の各カウンタを時/分/秒の各レジスタの値に更新

bit1: 出力許可

OE	ステータス/動作
0	WOT外部端子を汎用入出力または他の周辺機能用として使用する。
1	WOT外部端子を21ビットダウンカウンタの出力として使用する。

bit0: スタート

ST	動作
0	リアルタイムクロックモジュールの動作を停止し, 21ビットダウンカウンタおよび時/分/秒カウンタをクリアする。
1	時/分/秒レジスタの設定を時/分/秒カウンタにロードし, リアルタイムクロックモジュールの動作を開始させる。

使用上の注意 :

RTC モジュールのサブセカンドレジスタには 21 ビットカウンタのリロード値が格納されます。この値は , リロードカウンタが "0" になるとリロードされます。3 バイトすべてを変更するには , ライト命令とライト命令の間でリロード操作が実行されないように注意します。そうでないと , 21 ビットプリスケーラに新しいデータと古いデータのバイトの組合せによる不正な値がロードされてしまいます。通常は , ST ビットが "0" の間にサブセカンドレジスタの更新を行うことをお勧めします。

この更新を RTC 秒割込みの直後に行う場合は , ST が "0" にセットされておらずモジュールが動作中であっても , 次のリロード操作 (次の秒割込み) まで十分な時間をとってレジスタが確実に修正されるようにする必要があります。

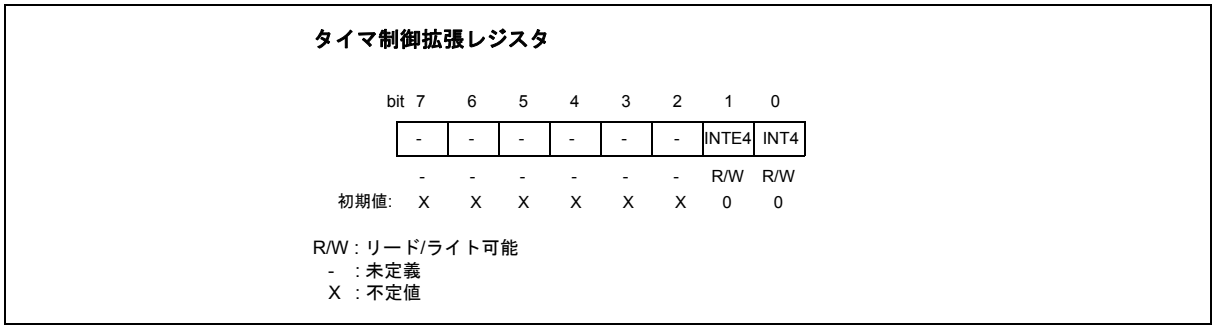
ST ビットを使用してレジスタの更新を行うときは次の点に注意する必要があります。

RUN ビットの立上りエッジで , レジスタに新しい値が書き込まれます。RUN ビットのクロックは RTC クロックです (デバイスとモードによってはメインクロック , サブ発振クロック , CR クロック)。更新が確実に正しく行われるように , 新しい値をレジスタに書き込んだら , ST を "0" にセットして RUN ビットが "L" になるのを待ち , ST を "1" に戻して回路の動作を再開させます。ST を "0" にセットすると , その後の 2 度目の RTC クロックの秒の立上りエッジで , RUN が "L" に遷移します。ST を "1" にセットすると , その後の RTC クロックの 1/2 秒の立上りエッジで RUN が "H" に遷移します。この操作を数回交互に繰り返して実行する必要がある場合は , RUN が "H" になるのを待ってから ST を "L" にセットしてください。

割込みルーチン内でタイマ制御レジスタ (WTCR) の INT0 ～ INT4 ビットをクリアした後 , 実際に INT0 ～ INT4 ビットがクリアされるのを待つ為に , 周辺バス 1 に接続されている周辺機能のレジスタを一度ダミーリードした後で , 割込みルーチンから復帰してください。

24.2.2 タイマ制御拡張レジスタ(WTCER)

Figure 24-5. タイマ制御拡張レジスタ(WTCER)



＜注意事項＞

WTCER の予約および未使用ビットを除くビットは , すべてのリセット要因によって初期化されます。

bit7～bit2: 未定義

書込み操作は動作に影響しません。リード値は不定です。

リードモディファイライト系命令は無効です。

bit1: 1/2秒(500ms)間割込み要求許可

21ビットカウンタがオーバフローすると, このフラグは"1"になります。

INTE4	動作
0	割込み要求なし
1	1/2秒(500ms)間割込み要求を許可

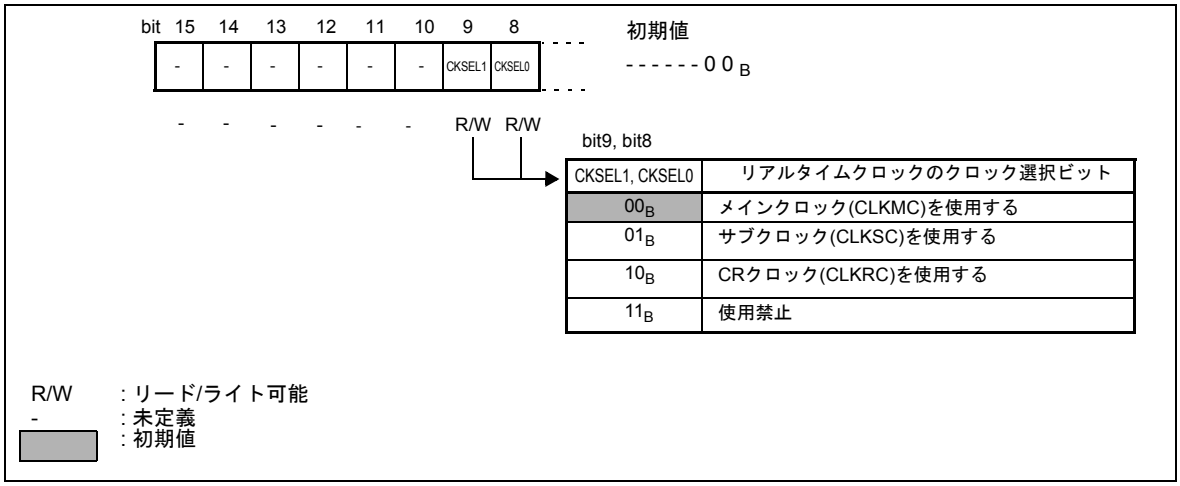
bit0: 1/2秒(500ms)間割込み要求フラグ

INT4	ステータス	
	読出し	書込み
0	割込み要求なし	フラグをクリアする。
1	1/2秒(500ms)間割込み要求あり	書込みは動作に影響しません。

24.2.3 クロック選択レジスタ(WTCKSR)

クロック選択レジスタはリアルタイムクロックのクロックソースの選択に使用します。

Figure 24-6. クロック選択レジスタ(WTCKSR)



＜注意事項＞

WTCKSR の未使用ビットを除くビットは , すべてのリセット要因によって初期化されます。

使用上の注意 :

WTCKSR:CKSEL[1:0] の設定を変更する際は , リアルタイムクロックが稼動しているクロックを , 変更後少なくともさらに 3 サイクルの間有効のままにしておきます。そのようにしないと , リアルタイムクロックは , 新しい設定に移行しません。その後 , 従来のクロックを停止してください。

例 : リアルタイムクロックをメインクロックから CR クロックに変更しなければならない場合

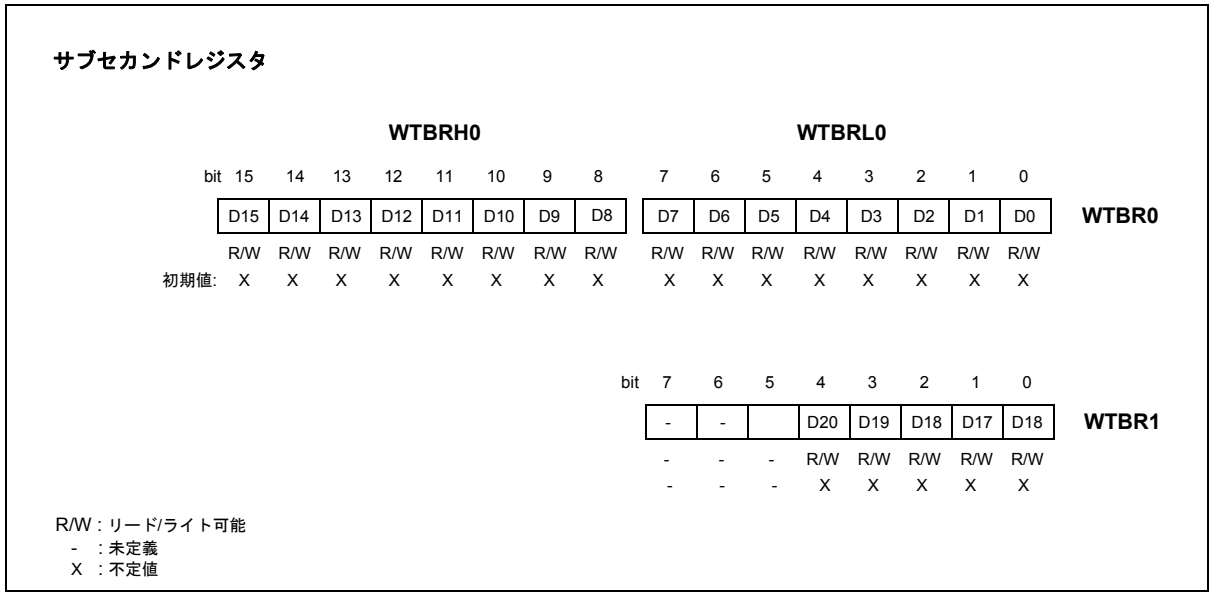
電源投入後 , メインクロックが有効となります。WTCKSR:CKSEL[1:0] に "10" を書込むことによって CR クロックに変更する場合は, メインクロックを有効にしておかなければなりません。そのようにしないと, RTC は CR クロックに変更できません。WTCKSR:CKSEL[1:0] へのライトアクセス後少なくとも , メインクロック 3 サイクルの間 , メイン発振子を有効にしなければなりません。

24.2.4 サブセカンドレジスタ

サブセカンドレジスタには発振クロックを分周する21ビットカウンタ用のリロード値が格納されます。リロード値は, 通常は21ビットカウンタが正確に1/2秒サイクル以内に出力するように設定されます。本レジスタはリセットによって初期化されませんが, 21ビットカウンタはリセットによって初期化されます。

24.2.4.1 サブセカンドレジスタ(WTBR)

Figure 24-7. サブセカンドレジスタ(WTBR)



使用上の注意 :

サブセカンドレジスタ (WTBR) には 21 ビットダウンカウンタにリロードする値が保持されます。21 ビットダウンカウンタの値が "0" になると , WTBR の設定が 21 ビットダウンカウンタにリロードされます。

＜注意事項＞

サブセカンドレジスタに設定するリロード値は 1/2 秒の時間に相当します。WTBR に設定されているリロード値を 2 回カウントすると 1 秒になります。

Table 24-1. 各種クロック構成におけるWTBRレジスタの設定例

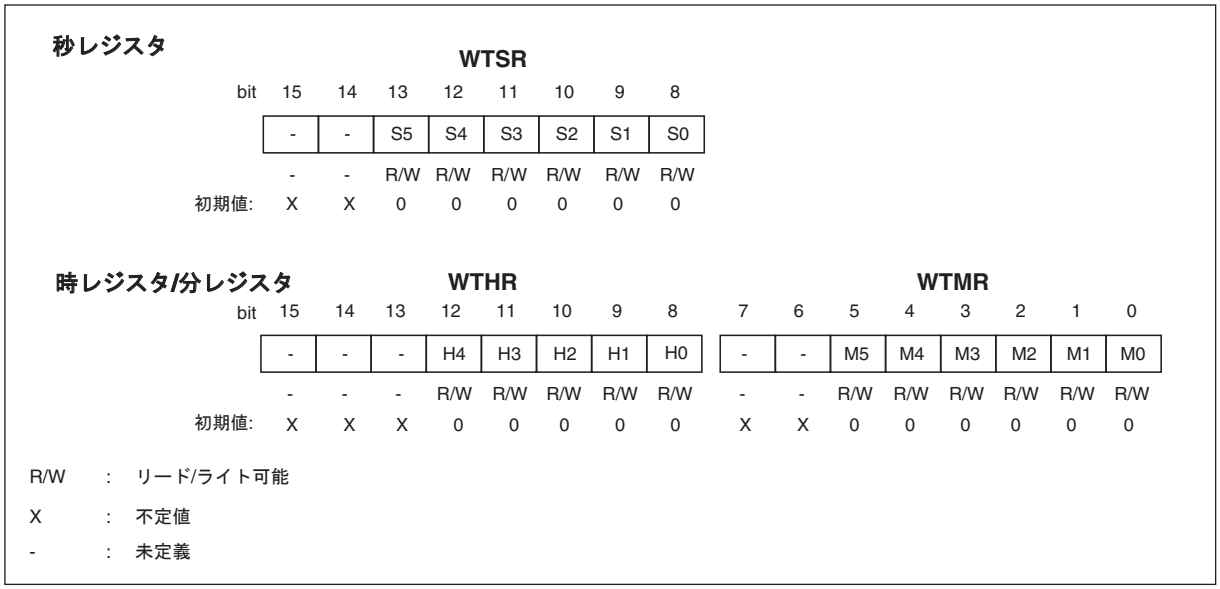
	WTBR1	WTBRH0	WTBRL0
メイン発振器, 4MHz	$0F_H$	42_H	$3F_H$
CR発振器, 2MHz	07_H	$A1_H$	$1F_H$
CR発振器, 100kHz	00_H	61_H	$A7_H$
サブ発振器, 32.768kHz	00_H	$1F_H$	FF_H

24.2.5 時 / 分 / 秒レジスタ(WTSR, WTMR, WTHR)

時/分/秒レジスタには時刻情報が格納されます。この情報は, 時, 分, 秒のバイナリ表現です。
レジスタから読み出すと単にカウンタ値が返されます。これらのレジスタへのライトアクセスも可能ですが, 書込みデータをカウンタにロードするにはUPDTビットを"1"にセットする必要があります。レジスタおよびカウンタはリセットによって初期化されます。

24.2.5.1 時 / 分 / 秒レジスタ(WTSR, WTMR, WTHR)

Figure 24-8. 時 / 分 / 秒レジスタ(WTSR, WTMR, WTHR)の構成



＜注意事項＞

WTSR, WTMR, WTHR の未使用ビットを除くビットは , パワーオンリセット , 低電圧リセット , 外部リセットによってのみ初期化され , そのほかのリセットでは初期化されません。

使用上の注意 :

時 / 分 / 秒レジスタに書き込む値は時 / 分 / 秒カウンタにロードする初期値です。

更新ビット (WTCR: UPDT) またはスタートビット (WTCR: ST) を "1" にセットすると , 時 / 分 / 秒レジスタの値が時 / 分 / 秒カウンタに書き込まれます。

時 / 分 / 秒カウンタの値は秒カウンタがオーバフローするたびに , つまり 1 分の周期で , 時 / 分 / 秒レジスタに保存されます。時 / 分 / 秒カウンタから読み出したときは ,
書き込んだ値ではなく保存されたカウント値が返されます。

時 / 分 / 秒レジスタは , 読出し用と書込み用の独立した 2 つのレジスタの組で構成されています。CPU が時 / 分 / 秒レジスタに書き込むと , 書込みレジスタが更新されます。読出しレジスタは内部ロジックから更新された値を取得しますが, これらは CPU が時 / 分 / 秒レジスタから読み出そうとするときに, いつでも利用可能な値です。リアルタイムクロックをセットした後でリアルタイムクロックを停止して再開する場合は , 読出しレジスタの最新の値で書込みレジスタを更新する必要があります。

3個のバイトレジスタが存在するため, レジスタから取得した値に一貫性があることを確認する必要があります。

例えば , 取得した値が「1 時 59 分 59 秒」の場合 ,「1 時 59 分 59 秒」,「1 時 0 分 0 秒」,「2 時 0 分 0 秒」の 3 つの可能性があります。

また , CPU の動作クロック (CLKB) がリアルタイムクロックの動作クロックを分周したものである場合 , リード操作中に値が変化して , レジスタからの読出し値が矛盾することがあります。1 秒割込みを使用すると , 次のレジスタ値の更新まで 1 秒あるので , 1 秒割込み時にレジスタ値を読み出すことをお勧めします。

WTSR, WTMR, WTHR レジスタにありえないデータ (60 秒など) を書き込まないように注意してください。

24.3 動作

リアルタイムクロックの動作を説明します。

リアルタイムクロックの動作

Figure 24-9. リアルタイムクロックの動作

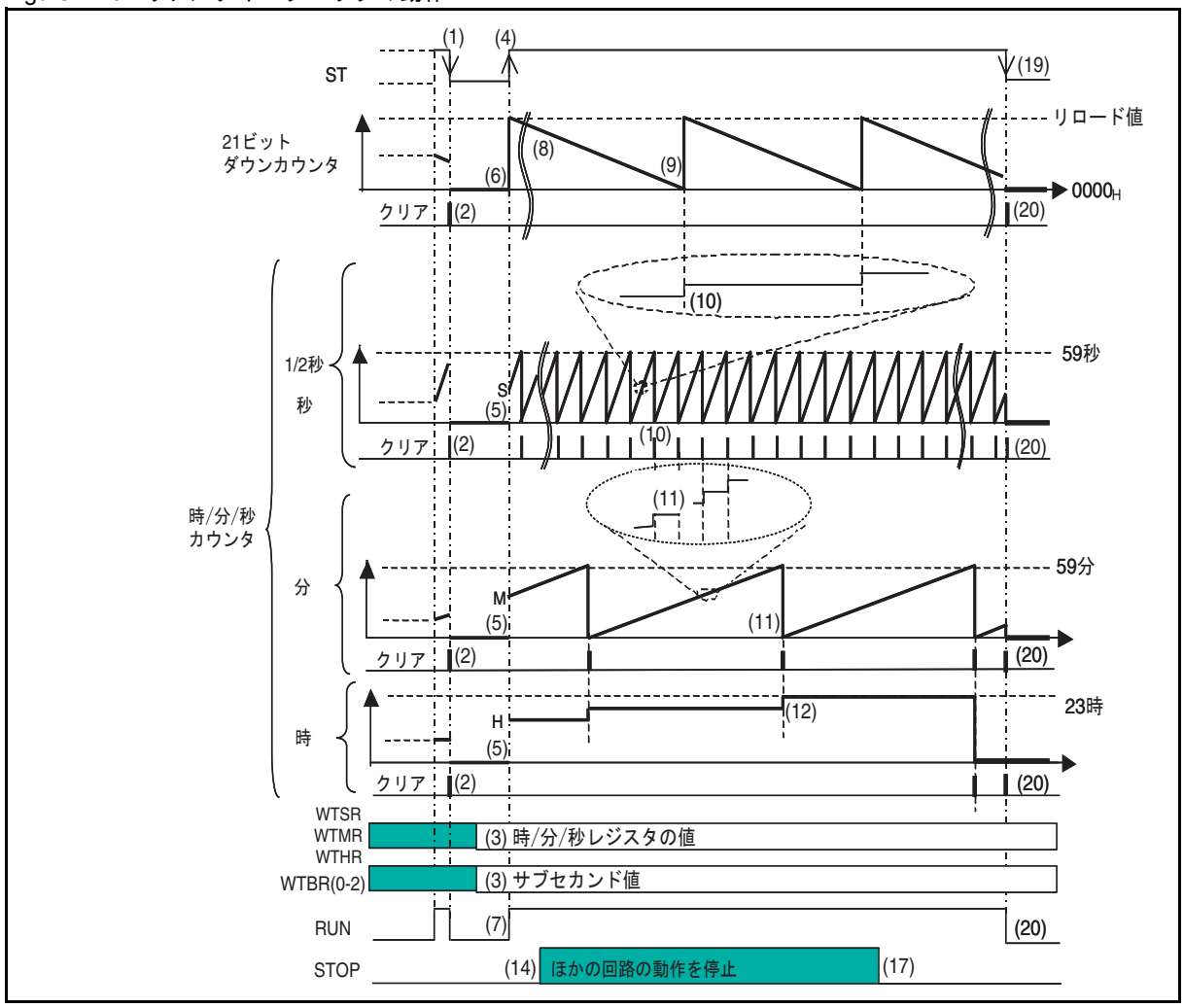
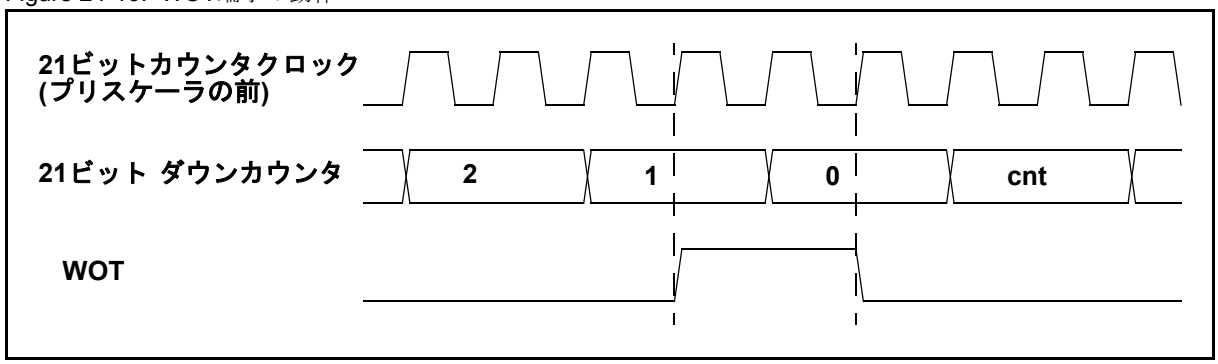


Figure 24-10. WOT端子の動作



(1) スタートビット (ST) を "1" にセットしてから "0" にセットします (レジスタ初期化操作)。

(2) この操作 (ST=0) によって , 21 ビットダウンカウンタと時 / 分 / 秒の各タイマが 0 にリセットされ停止します。

(3) ソフトウェアで時 / 分 / 秒の値を時 / 分 / 秒レジスタ WTHR/WTMR/WTSR に書き込みます。

(4) ソフトウェアでサブセカンドレジスタ WTBRL0/WTBRH0/WTBR1 に適切な値を書き込みます。

(5) 割込み要求ビット (INT0, INT1, INT2, INT3, INT4) を初期化し , 割込み要求許可ビット (INTE0, INTE1, INTE2, INT3, INTE4) を「割込み許可」にセットします。

(6) スタートビット (ST) を "1" にセットします。

(7) スタートビット (ST=1) により時 / 分 / 秒レジスタ WTHR/WTMR/WTSR の値が時 / 分 / 秒タイマにロードされます。

(8) サブセカンドレジスタ WTBRL0/WTBRH0/WTBR1 の値を 21 ビットダウンカウンタにロードします。

(9) 実行フラグ (RUN) が "1" になります。

(10) 21 ビットダウンカウンタはメインクロック (CLKMC) の 2 分周 (4/2 MHz), サブクロック (CLKSC) の 2 分周 (32.768/2 kHz), または CR クロック (CLKRC) の 2 分周 (100/2 kHz または 2/2MHz) でカウントを開始します。

(11) 21 ビットダウンカウンタが 000000_H に達すると , サブセカンドレジスタの値が 21 ビットダウンカウンタにロードされ , 1/2 秒割込み要求が発生します。2 つ目の 1/2 秒割込みのたびに秒の割込みが発生します。

(12) 秒カウンタが "59" に達すると , 次のカウント時にクリアされ , 分カウンタのカウントが 1 つ増えるとともに分の割込み要求が発生します。

(13) 分カウンタが "59" に達すると , 次のカウント時にクリアされ , 時カウンタのカウントが 1 つ増えるとともに時間の割込み要求が発生します。

(14) 時カウンタが "23" に達すると , 次のカウント時にクリアされ , 日の割込み要求が生成されます。

(15) ソフトウェアによりビット SMCR:SMS[1:0] を 10_B に変更し , MCU の状態を時計モードに変更します (詳細については「9.2 スタンバイモード制御レジスタ (SMCR)」を参照)。リアルタイムクロックは時計モードで動作を続行します。

(16) 時計モードから復帰します (割込み要求によって)。

(17) スタートビット (ST=0) によって , 21 ビットダウンカウンタと時 / 分 / 秒の各カウンタが 0 にリセットされ停止します。

24.4 注意事項

リアルタイムクロック使用時の注意事項を説明します。

24.4.1 注意事項

- オーバフローによる割込み要求フラグ(WTCR:INT0,WTCR:INT1,WTCR:INT2,WTCR:INT4,WTCER:INT4)への"1"のセットと"0"の書込みが同時に発生した場合, フラグは"1"にセットされます(フラグのセットが優先されます)。
- 更新ビット(WTCR: UPDT)への"1"の書込みと更新の完了が同時に発生した場合, 更新ビット(UPDT)は"0"にセットされます。
- 秒カウンタに59の値が保持されているときに更新ビット(WTCR: UPDT)に"1"を書き込んでも時/分/秒カウンタは更新されず, 更新ビットの値は"0"のままになります。
  時/分/秒カウンタを更新するには, スタートビット(WTCR: ST)に"0"を書き込んで時/分/秒カウンタを"0"にクリアしてから, スタートビット(ST)に"1"を書き込むことをお勧めします。
- 更新ビット(WTCR: UPDT)を使用して時/分/秒カウンタを更新した後で周辺クロック(CLKP1)を停止する場合は, 周辺クロック(CLKP1)を停止する前に時/分/秒レジスタを読み出して更新が完了しているかどうかを確認してください。
- リアルタイムクロックモジュールの使用を開始するとき, スタートビット(ST)を"1"から"0"に変更して時/分/秒カウンタと21ビットダウンカウンタを"0"にクリアしてください。
- RTCモジュールのサブセカンドレジスタには21ビットプリスケーラのリロード値が格納されます。この値は, リロードカウンタが"0"になるとリロードされます。3バイトすべてを変更するには, ライト命令とライト命令の間でリロード操作が実行されないように注意してください。そうでないと, 21ビットプリスケーラに新しいデータと古いデータのバイトの組合せによる不正な値がロードされてしまいます。通常は, STビットが"0"の間にサブセカンドレジスタの更新を行うことをお勧めします。
- この更新をRTC秒割込みの直後に行う場合は, STが"0"にセットされておらずモジュールが動作中であっても, 次のリロード操作(次の秒割込み)までの十分な時間が確保されるはずです。
- STビットを使用してレジスタの更新を行うときは次の点に注意する必要があります。RUN ビットの立上りエッジで新しい値がレジスタに書き込まれます。RUNビットにはRTCクロック(デバイスおよびモードによって32 kHz, 100kHz, 2MHz または4 MHz)が使用されます。更新が確実に正しく行われるように, 新しい値をレジスタに書き込んだら, STを"0"にセットしてRUNビットが"L"になるのを待ち, STを"1"に戻して回路の動作を再開させます。STを"0"にセットすると, その後のRTCクロックの2番目の立上りエッジでRUNが"L"に遷移します。STを"1"にセットすると, その後のRTCクロックの1/2秒の立上りエッジでRUNが"H"に遷移します。この操作を数回交互に繰り返して実行する必要がある場合は, RUNが"H"になるのを待ってからSTを"L"に設定してください。
- サブセカンドレジスタ(WTBRL0,WTBRH0,WTBR1)の更新中にリロードが発生した場合は21ビットダウンカウンタに予期しない値がリロードされることがあります。したがって, スタートビット(WTCR: ST)を"0"にセットしてサブセカンドレジスタ(WTBR)を更新することをお勧めします。
- すべてのサブセカンドレジスタ(WTBRL0,WTBRH0,WTBR1)を"0"にセットした場合は21ビットダウンカウンタは更新されず, リアルタイムクロックモジュールの動作は停止します。
- 時/分/秒レジスタWTHR/WTMR/WTSRからの読出し中に桁上げが発生した場合は, 不正な値が返されることがあります。これを避けるために, 割込み(INT0～INT4)を使用して時刻(HH/MM/SS)の読出しを行うことをお勧めします。
- リアルタイムクロックモジュールが正しく動作するように,サブクロック(CLKSC)またはCRクロック(CLKRC)の周波数が周辺クロック(CLKP1)の周波数より低くなるようにしてください。そうでないと, WTHR/WTMR/WTSRから正しい値を読み出すことができなくなります。

- 上記のレジスタにはバイトアクセスしか許されていません。下記に示すように時または分の境界をまたぐタイミングで上記のレジスタを読み出した場合は, 時刻情報を誤って判断する可能性があります。論理的に矛盾のない値を取得するには, 読出しを数回行う必要があります。

  例 : 秒レジスタ読出し開始 : 02:59:59 => 03:59:59 => 03:00:00

  時レジスタ読出し開始 : 02:59:59 => 02:00:00 => 03:00:00

  上記の場合 , 現在の時刻は 3 時であると判定する必要があります。

25. クロック補正ユニット

クロック補正ユニットの機能および動作について説明します。

25.1 概要

クロック補正ユニットを使用すると, メイン発振クロックを基準にしてサブ発振クロックまたは CR 発振クロックを補正できます。 クロック補正ユニットの概要, レジスタの説明, アプリケーションノートを示します。

概要

サブ発振クロック (CLKSC) または CR 発振クロック (CLKRC) の発振周期を , メイン発振クロック (CLKMC) で測定するモジュールです。

このモジュールとソフトウェアによる処理を併用することで , サブ発振クロックまたは CR 発振クロックの精度を , メイン発振クロックの精度に近づけることができます。

このモジュールは 2 つのタイマから構成されています。

測定時間タイマは , 測定対象クロック (CLKSC / CLKRC) で動作します。

補正タイマは , メイン発振クロック (CLKMC) で動作します。

基本的なクロック周波数は , CLKSC が 32kHz, CLKMC が 4MHz, CLKRC が 100kHz/2MHz です。

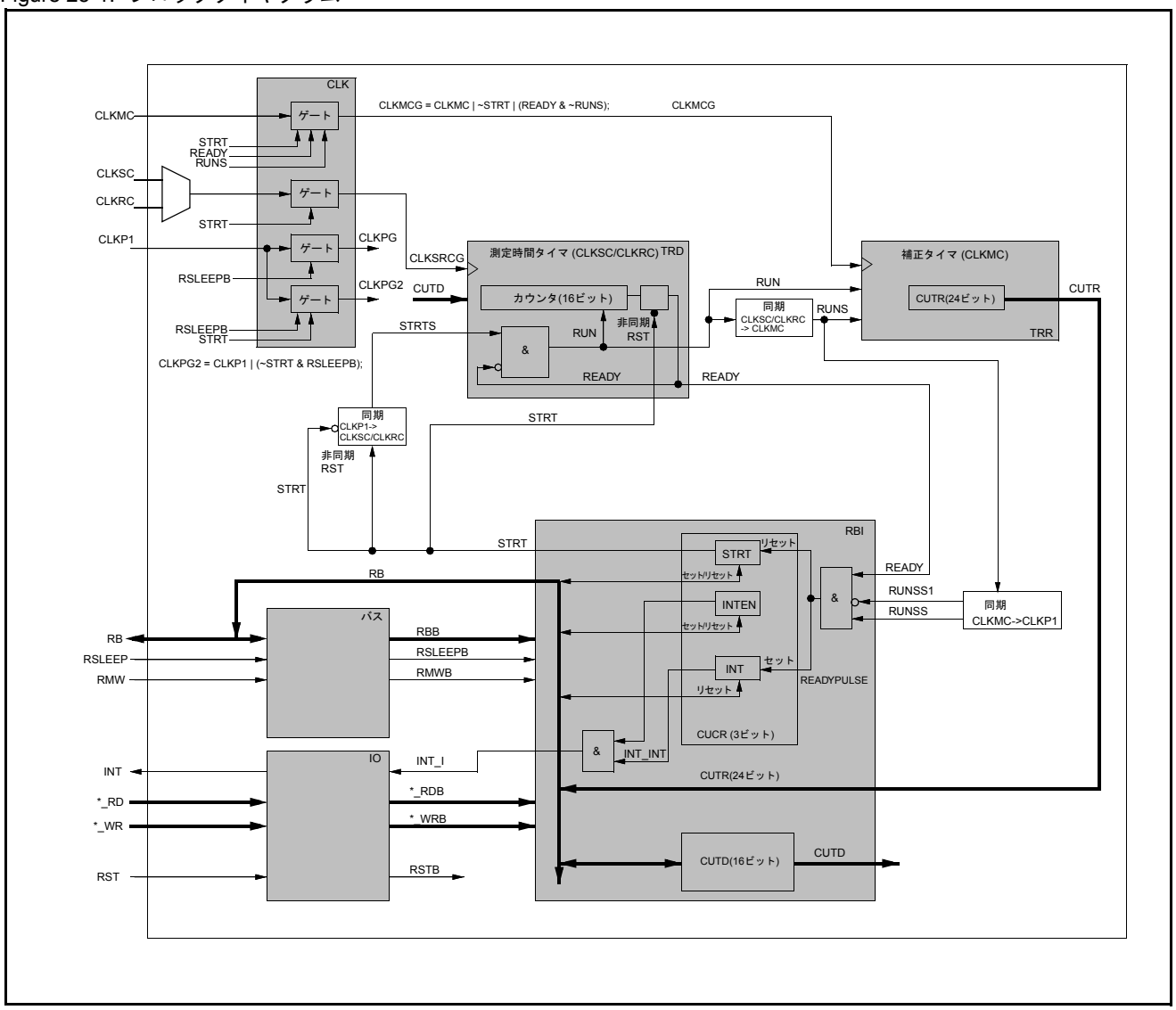
測定時間タイマが動作を開始すると , 補正タイマも動作を開始します。

測定が終了すると , 補正タイマのカウント値がレジスタに格納されます。

このレジスタに格納された値をソフトウェアで処理することで, 必要なリアルタイムクロックモジュールの設定が計算されます。

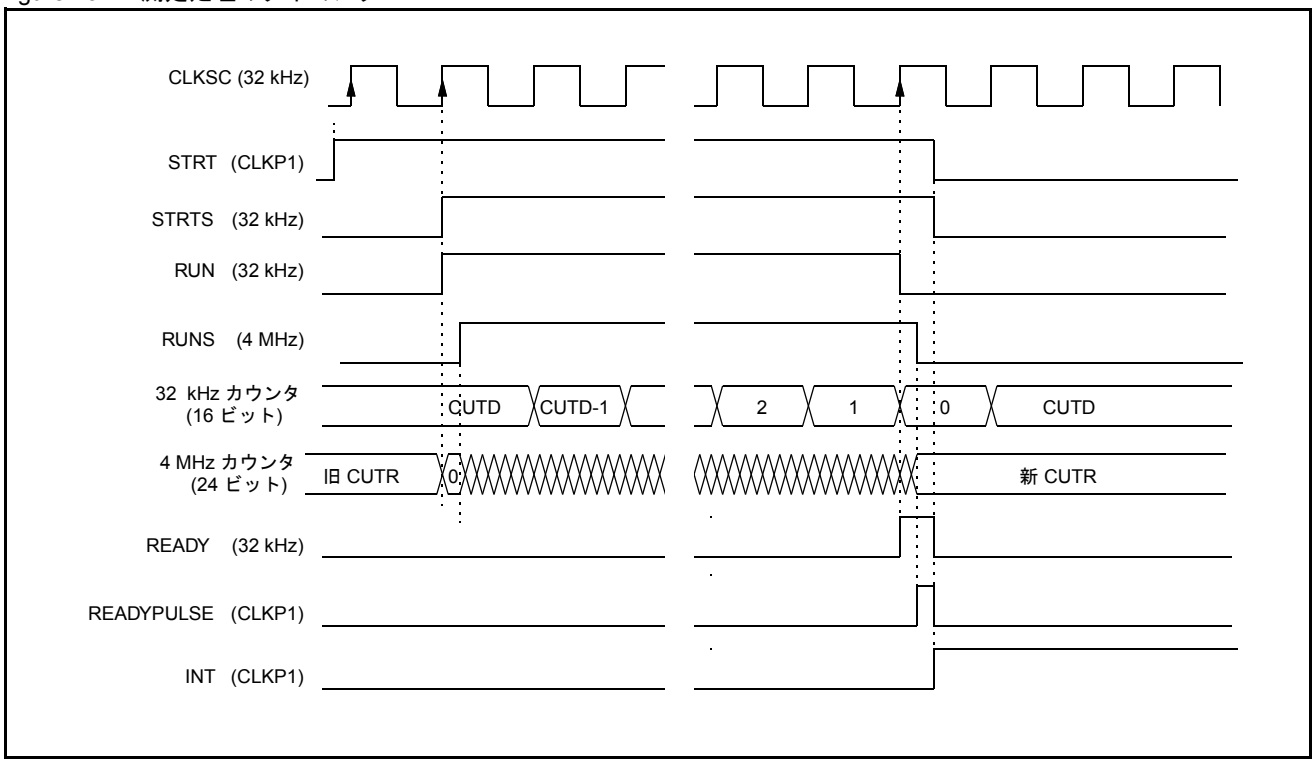
ブロックダイヤグラム

Figure 25-1. ブロックダイヤグラム



タイミング

Figure 25-2. 測定処理のタイミング



クロック

本モジュールは , 3 種類のクロックで動作します。

メイン発振クロック (CLKMC), 測定対象クロック (CLKSC/CLKRC) および周辺クロック (CLKP1) です。

クロックの概要は ,Figure 6-1 を参照してください。

同期回路は , 異なるクロックで動作する信号の同期化を行う回路です。

3 種のクロックは , 必要のない期間 , 自動的に停止します。

CUCR:STRT (「25.2 レジスタの説明」参照)が"0"のとき, CLKSC/CLKRCとCLKMCで動作する回路は停止します。

CLKP1 は RSLEEP と CUCR:STRT で制御され , 不要な期間停止します。

クロック周期は以下の条件を満足する必要があります。

- $T_{CLKSC/CLKRC} > 2 \times T_{CLKMC} + 3 \times T_{CLKP1}$
- $T_{CLKMC} < 1/2 \times T_{CLKSC/CLKRC} - 3/2 \times T_{CLKP1}$
- $T_{CLKP1} < 1/3 \times T_{CLKSC/CLKRC} - 2/3 \times T_{CLKP1}$

Table 25-1. 条件を満足する有効クロック比の例

	CLKSC		CLKRC		CLKMC		CLKP1	
最大動作速度	100 kHz	10 us	2 MHz	500 ns	10 MHz	100 ns	50 MHz	20 ns
通常動作	32 kHz	31.25 us	100 kHz	10 us	4 MHz	250 ns	>2 MHz	500 ns

25.2 レジスタの説明

補正ユニットのレジスタ一覧を示し, 各レジスタの機能を詳細に説明します。

25.2.1 クロック補正ユニットレジスタ

レジスタ			
+0	+1	+2	+3
CUCR [R/W] ---0--00 --------		CUTD [R/W] 00000000 10000000	
CUTR1 [R] 00000000 --------		CUTR2 [R] 00000000 00000000	

25.2.2 クロック補正ユニット制御レジスタ(CUCR)

制御レジスタ下位バイト

bit	7	6	5	4	3	2	1	0
	RES	-	-	STRT	-	CKSEL	INT	INTEN
属性 ⇒	(W0/R)	(R)	(R)	(R/W)	(R)	(R/W)	(R/W)	(R/W)
初期値⇒	(X)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

R/W : リード/ライト可能

クロック補正ユニット制御レジスタ (CUCR) には , 以下の機能があります。

- 測定の開始/ 停止
- 割込みの許可/ 禁止設定
- 測定終了観測ビット

BIT[0]: INTEN - 割込み許可

0	割込み禁止 (初期値)
1	割込み許可

本ビットは測定完了割り込み許可ビットです。

本ビットが "1" に設定された状態で , INT ビットが "1" にセットされると , 本モジュールは CPU へ割込みを発行します。

INT ビットは、本ビットが "0" に設定されている場合も , ハードウェアによるセットは行われます。

BIT[1]: INT - 割込み

0	測定中/動作停止中(初期値)
1	測定完了

本ビットは測定の終了を示します。

測定開始後 , 測定時間タイマが 0 になると , 補正タイマデータレジスタ (CUTR) は , その時の補正タイマ値を格納し , 本ビットは "1" にセットされます。

リードモディファイライト命令のリード時は ,"1" が読み出されます。

本ビットに "0" を書き込むと "0" に設定されます。

本ビットへの "1" 書込みは無効です。

本ビットは , ハードウェアでは "0" にリセットされません。
そのため , 新たな測定を開始する場合は , ソフトウェアにより "0" を書き込む必要があります。
"0" を書き込まずに再開した場合 , 本ビットは測定中も "1" のままです。
その場合でも , CUCR:STRT ビットを確認することで測定の終了を確認することができます。

BIT[2]: CKSEL - 測定対象クロック選択

0	use CLKSC (初期値)
1	use CLKRC

CR クロック周波数は CKSCR:RCFS ビット (「第 6 章 " クロック "」参照) によって設定されます。

- 2 系統品の場合
  選択されているクロックがずっと前から動作している場合 , クロックの選択を変更後 , 少なくとも 3 サイクルは元のクロックを動かし続ける必要があります。クロックを止めてしまうと , クロックの選択が切り替わりません。

BIT[4]: STRT - 測定開始

0	測定停止/モジュール動作停止(初期値)
1	測定開始

本ビットが "1" にセットされると , 測定が開始されます。

測定時間タイマは , 測定時間タイマデータレジスタ (CUTD) に書かれた値からカウントダウンを開始し , 補正タイマは 0 からのカウントアップを開始します。

測定時間タイマが 0 になると , 本ビットはハードウェアにより "0" にリセットされます。

測定中に本ビットに "0" が書き込まれると , 測定は直ちに停止されます。

CPU による "0" 書込みとハードウェアによる "0" リセットが同時に発生した場合は ,
ハードウェアが優先され , 測定が正常に終了して INT ビットが "1" にセットされます。

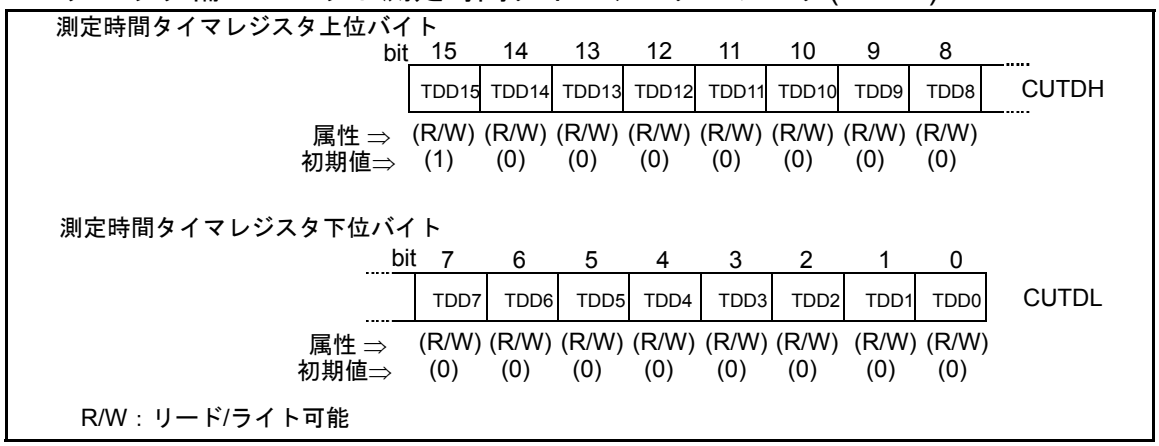
測定中は , 本ビットへの "1" 書込みは無効です。

BIT[7]: RES - 予約ビット

常に "0" を書き込んでください。読出し値は不定です。

リードモディファイライト系命令は無効です。

25.2.3 クロック補正ユニット測定時間タイマデータレジスタ(CUTD)



本レジスタは、補正のための測定時間を設定します。

本レジスタに設定された値は , 測定時間タイマのリロード値になります。

測定が開始されると , 本レジスタに書き込まれた値が , 測定時間タイマに取り込まれます。

測定時間タイマは , 0 になるまでカウントダウンされます。

測定時間タイマは , CUCR:CKSEL の設定により , CLKSC または CLKRC クロックで動作します。

初期値は 8000_H です。

この値は , 32.768kHz の水晶を測定対象としたときの , 測定時間 1 秒に相当します。

本レジスタは , 測定停止中 (CUCR:STRT=0) にのみ書込みを行ってください。

本レジスタを 0000_H に設定すると , 測定時間は , $(FFFF_H + 1) \times TCLKSC/CLKRC$ となります。

【設定例 1】

32kHz発振で1秒間の測定時間を設定する場合は,本レジスタに$8000_H = 32768(10進)$を設定する必要があります。

この数値は水晶の正確な発振周波数 (32768Hz) から得られます。

測定結果(4MHzの水晶発振子を使用する場合)の理想値を次の表に示します。

32kHz: 測定時間による理想的な測定結果

測定時間	CUTD値	CUTR値
2秒	0x0000	0x7A1200
1.75秒	0xE000	0x6ACFC0
1.5秒	0xC000	0x5B8D80
1.25秒	0xA000	0x4C4B40
1秒	0x8000	0x3D0900
0.75秒	0x6000	0x2DC6C0
0.5秒	0x4000	0x1E8480
0.25秒	0x2000	0x0F4240

【設定例 2】

100kHz 発振で 0.5 秒間の測定時間を設定する場合は , 本レジスタに $C350_H = 50000$ (10 進) を設定する必要があります。

この数値は水晶の正確な発振周波数 (100000Hz) から得られます。

測定結果(4MHzの水晶発振子を使用する場合)の理想値を次の表に示します。

100kHz: 測定時間による理想的な測定結果

測定時間	CUTD値	CUTR値
0.5秒	0xC350	0x1E8480
0.25秒	0x61A8	0x0F4240
0.125秒	0x30D4	0x07A120
0.1秒	0x2710	0x061A80

【注意事項】

CUCR:STRT ビットへの "1" 書込みから , ハードウェアによる CUCR:STRT の "0" リセットまでにかかる全処理時間は , 制御信号が異なるクロックで動作する回路を伝わっていくため , 測定時間より長くなります。

「処理時間 < $(CUTD + 3) \times T_{CLKSC/CLKRC}$」であり , 測定時間は ,「$CUTD \times T_{CLKRC}$」となります。

25.2.4 クロック補正ユニット補正タイマデータレジスタ(24ビット) (CUTR)

本レジスタには測定結果が保持されます。

測定開始時 , 補正タイマはゼロからのカウントアップを開始します。

測定時間タイマが 0 になると , 補正タイマはカウントを停止します。

本レジスタは , 次の測定が開始されるまで , カウント値を保持します。

注意事項:

- 測定中に , このレジスタを読み出しても値は不定です。
- 測定終了は , CUCR:INT ビットもしくは CUCR:STRT ビットにより確認できます。
- CUCR:INT が "0" から "1" になり , かつ CUCR:STRT が "1" から "0" になると , 本レジスタの値は有効な値になります。

補正タイマレジスタ 2 下位バイト

bit	7	6	5	4	3	2	1	0	
	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0	CUTR2L
属性 ⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
初期値⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

補正タイマレジスタ 2 上位バイト

bit	15	14	13	12	11	10	9	8	
	TDR15	TDR14	TDR13	TDR12	TDR11	TDR10	TDR9	TDR8	CUTR2H
属性 ⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
初期値⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

補正タイマレジスタ 1 下位バイト

bit	7	6	5	4	3	2	1	0	
	TDR23	TDR22	TDR21	TDR20	TDR19	TDR18	TDR17	TDR16	CUTR1L
属性 ⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
初期値⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

補正タイマレジスタ1 上位バイト

bit	15	14	13	12	11	10	9	8	
	-	-	-	-	-	-	-	-	CUTR1H
属性 ⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
初期値⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

R : リードオンリ

本レジスタへの書込みは無効です。

補正タイマは , メイン発振クロック (CLKMC) で動作します。

25.3 アプリケーションノート

補正の精度, 電力消費, 測定時間に関するアプリケーションノートを示します。

25.3.1 32.768 kHz

測定時間タイマデータレジスタに設定する値は , 以下の方法で計算できます。

補正に 1 秒の測定時間が必要な場合は , 8000_H = 32768(10 進) を測定時間タイマデータレジスタに設定する必要があります。

これは 32.768kHz 発振クロックで , 32768 クロックカウントする事で 1 秒になるからです。

この設定により , 補正タイマデータレジスタに約 $3D0900_H$ の値が格納されます。

この値は, 4MHz発振クロックの1秒あたりのクロック数4000000に相当します。

32.768kHzと4.0MHz発信器による理想的な測定結果(CUTR)

測定時間	CUTD値	CUTR値
2秒	0x0000	0x7A1200
1.75秒	0xE000	0x6ACFC0
1.5秒	0xC000	0x5B8D80
1.25秒	0xA000	0x4C4B40
1秒	0x8000	0x3D0900
0.75秒	0x6000	0x2DC6C0
0.5秒	0x4000	0x1E8480
0.25秒	0x2000	0x0F4240

25.3.2 100 kHz

測定時間タイマデータレジスタに設定する値は , 以下の方法で計算できます。

補正に 0.5 秒の測定時間が必要な場合は , $C350_H$ = 50000 (10 進) を測定時間タイマデータレジスタに設定する必要があります。

これは 100kHz 発振クロックで、50000 クロックカウントする事で 0.5 秒になるからです。

この設定により , 補正タイマデータレジスタに約 $1E8480_H$ の値が格納されます。

この値は, 4MHz発振クロックの0.5秒あたりのクロック数2000000に相当します。

100kHzと4.0MHz発信器による理想的な測定結果(CUTR)

測定時間	CUTD値	CUTR値
0.5秒	0xC350	0x1E8480
0.25秒	0x61A8	0x0F4240
0.125秒	0x30D4	0x07A120
0.1秒	0x2710	0x061A80

クロック補正ユニットを用いる目的は , 日中表示される時計を正確に保つための電力を減らすことです。

25.3.3 精度

補正の精度は , 補正タイマのカウントクロック (CLKMC) 周波数と , 測定時間に依存します。

補正タイマの最大誤差は ±1 桁です。

CLKMC の周波数が 4MHz, 測定時間が 1 秒である場合の精度は , 以下の方法で計算されます。

0.25µs(CLKMC 周期) ÷ 1s(測定時間) = 0.25 ppm

一般的には , 次のようになります。

精度 = CLKMS 周期 ÷ 測定時間

25.3.4 電力消費

補正動作時の消費電流 I_{RUN} が , RTC モード時の消費電流 I_{RTC} の 20 倍 ($I_{RUN} = 20 \times I_{RTC}$) であると仮定します。

測定時間 1 秒での補正を毎分行うとすると , 電流消費の増加は , おおよそ ,

$I_{RUN} \times 1/60 = 20 \times I_{RTC}/60 = 1/3 \times I_{RTC}$

となります。

この電流増加を前提に , システム要求限界に影響を与えないように補正ルーチンを用意する必要があります。

例えば , 補正ルーチンを最低の周波数で設計する , などです。

一般的に推奨される RTC モードでの電力増加理論値は、5% です。

25.3.5 測定限界

- 測定時間タイマ
  32kHz発振クロックで動作している場合,測定時間の上限は約2秒です。
  100kHzで動作している場合,上限は0.5秒です。
- 補正タイマ
  4MHz発振クロックで動作している場合,測定時間の上限は4秒です。

26. LCDコントローラ/ドライバ

LCD コントローラ/ドライバの機能および動作について説明します。

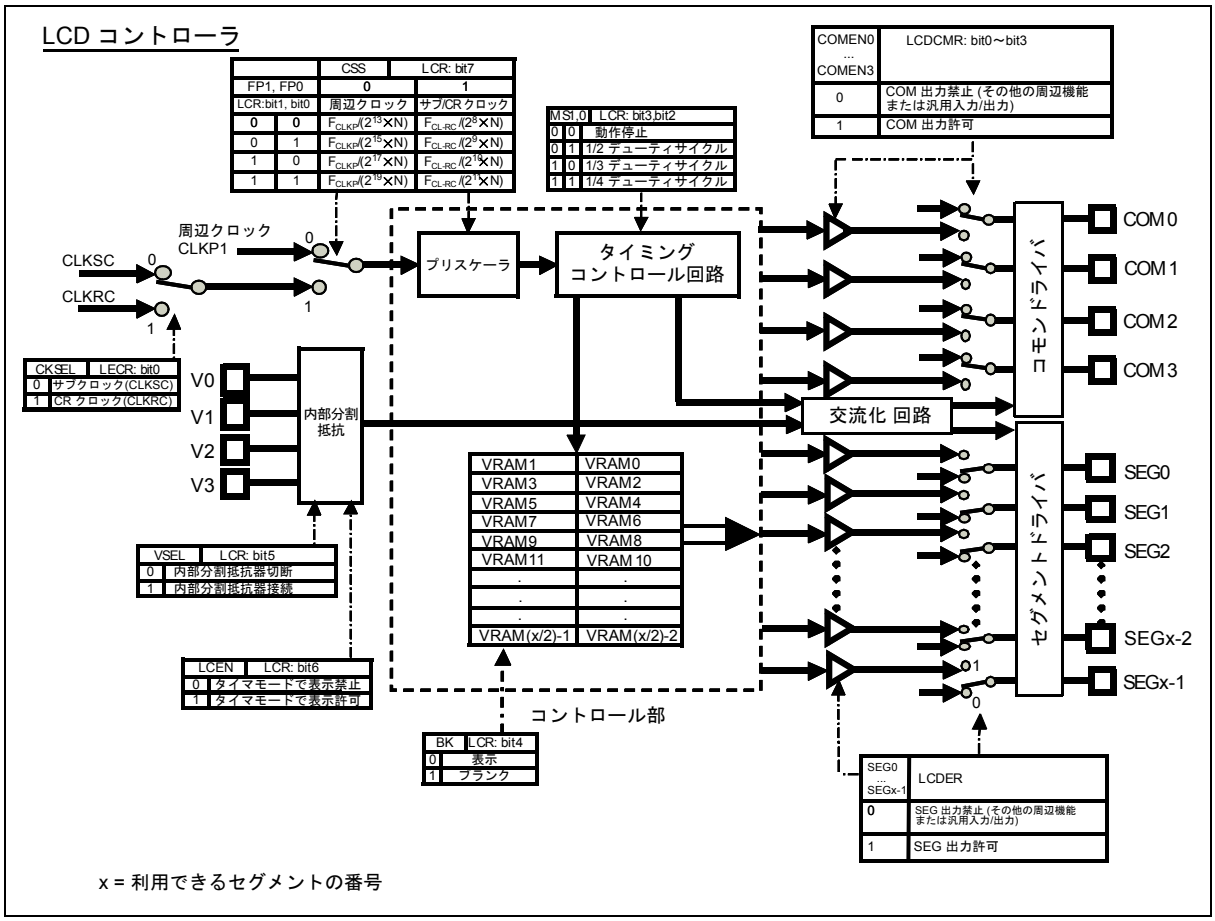
26.1 LCDコントローラ/ドライバの構成

LCD コントローラ / ドライバは以下に示したブロックで構成されています。機能的には , 表示用 RAM の内容に従って , セグメント信号やコモン信号を生成するコントローラ部 , LCD を駆動するドライバ部で構成されています。

- LCD 制御レジスタ (LCR)
- 表示用RAM
- プリスケーラ
- タイミングコントローラ
- 交流化回路
- コモンドライバ
- セグメントドライバ
- 分割抵抗

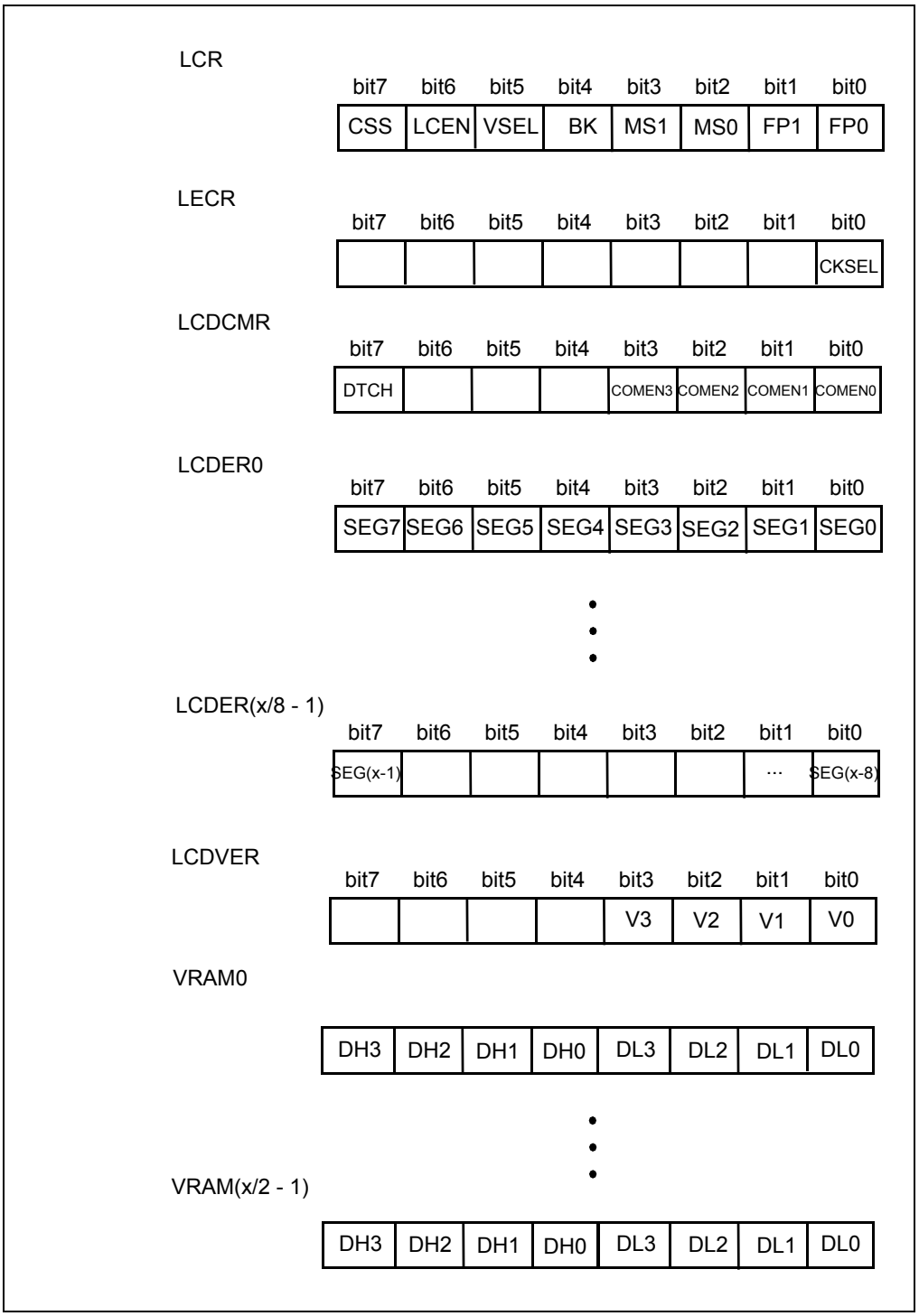
LCDコントローラ/ドライバのブロックダイヤグラム

Figure 26-1. LCDコントローラ/ドライバのブロックダイヤグラム



レジスタリスト

Figure 26-2. レジスタリスト



26.1.1 LCDコントローラ/ドライバ分割抵抗

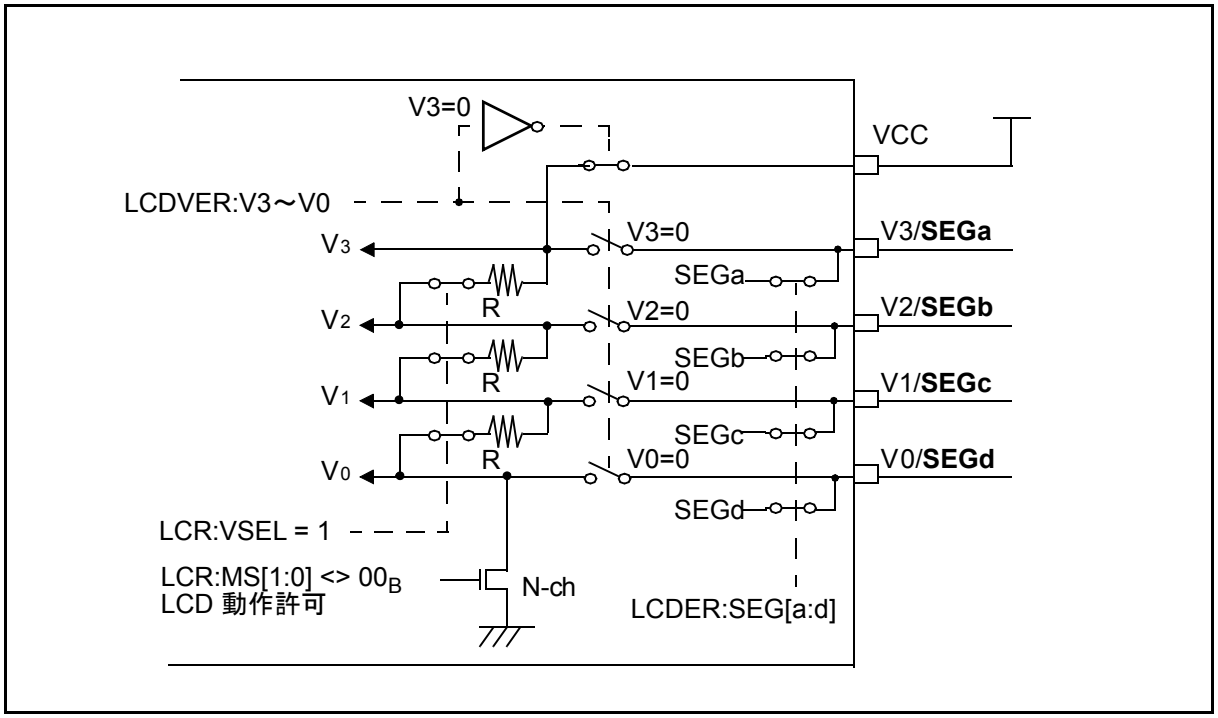
LCD ドライバの電源電圧は, 外部分割抵抗または内部分割抵抗によってつくられます。
V_{CC}とV3端子間に可変抵抗を接続することで,輝度を調節することができます。
内部分割抵抗を使用する場合には, V0～V3端子はセグメントドライバとして使用することができます。

内部または外部分割抵抗の選択

LCD 駆動電源制御ビット (LCR:VSEL) を使用します。

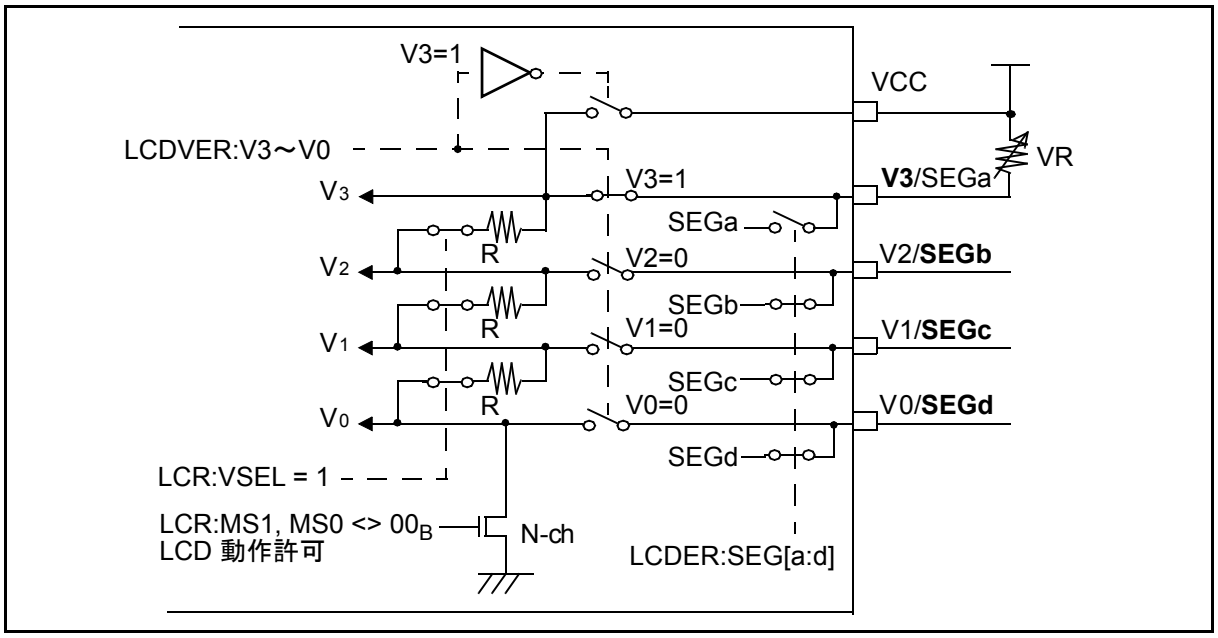
制御動作	LCD駆動電源制御ビット(VSEL).
外部分割抵抗を使用 (内部分割抵抗を切断)	"0" に設定
内部分割抵抗を使用 (内部分割抵抗を接続)	"1" に設定

Figure 26-3. 内部分割抵抗を使用



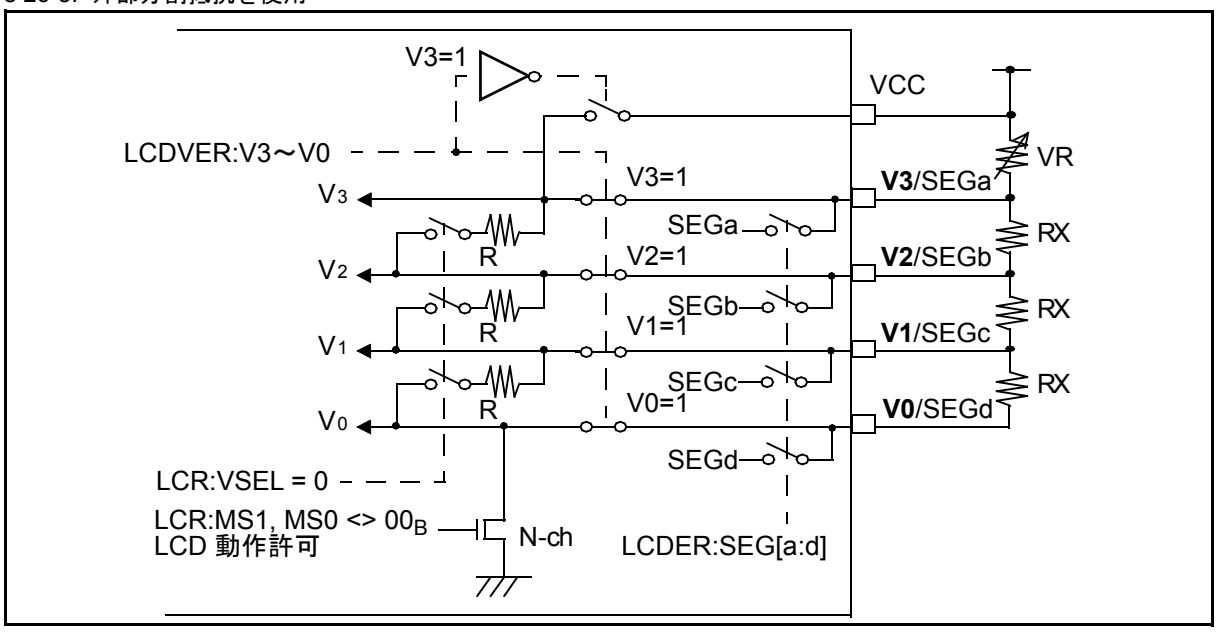
LCD 駆動電圧は内部分割抵抗で発生します。端子 V0 ～ V3 は LCDVER:V3 ～ V0 対応ビットにゼロ設定した場合 , セグメント端子として利用できます。LCDVER:V3 = 0 の場合は , 内部分割抵抗は VCC に接続します。

Figure 26-4. 内部分割抵抗使用時の輝度調整



内部分割抵抗を使用している場合でも , LCDVER:V3=1 に設定し , 外部 (VCC ～ V3 端子間) VR (可変抵抗) を接続して , 輝度を調節できます。内部抵抗は V3 端子に接続します。V3 に接続された外部可変抵抗は輝度調整に使用します。LCDER:SEGa の設定は , 影響しません。

Figure 26-5. 外部分割抵抗を使用



LCD 駆動電圧は外部分割抵抗を LCD 駆動電源端子 (V0 ～ V3) に接続してつくります。

各 V 端子は LCDVER:V[3:0] = 1111_B に設定する必要があります。各 V 端子の対応するセグメント許可ビット LCDVER:SEG[x] は動作に影響しません。

LCR:VSEL = 0 に設定して , 内部分割抵抗を切断してください。

外部分割抵抗を使用した場合での, 電流削減について

V0 端子は内部でトランジスタを介して Vss (GND) に接続しています。そのため , LCD 動作停止時は電流を切断できます。電流を切断するには , 表示モード選択ビット (LCR:MS[1:0]=00_B) を使用します。

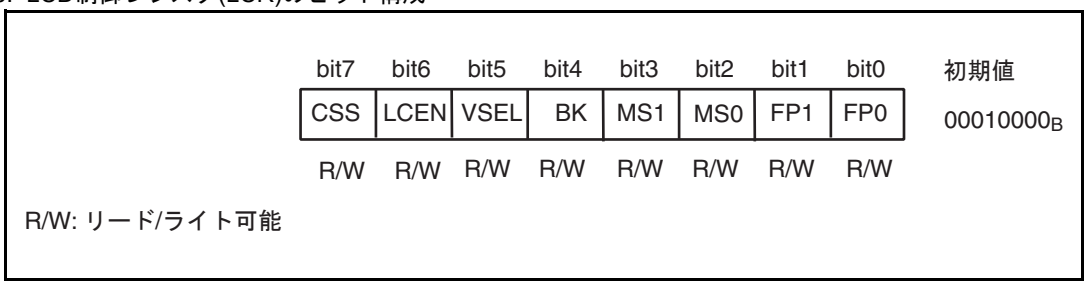
26.2 LCDコントローラレジスタ

LCD コントローラ/ドライバに関連するレジスタについて説明します。

26.2.1 LCD制御レジスタ(LCR)

Figure 26-6 は , LCD コントローラ / ドライバに関連するレジスタのビット構成を示します。

Figure 26-6. LCD制御レジスタ(LCR)のビット構成



[bit7] CSS: フレーム周期発生用クロックを選択します。

0	周辺クロック (CLKP1)
1	LECR:CKSEL が選択したクロック

＜注意事項＞

周辺クロック CLKP1 が選択され , MCU が停止モードまたはタイマモードの場合 , CLKP1 が停止するために , LCD も動作を停止します。

[bit6] LCEN: タイマモードで動作を許可します。

0	タイマモードで LCD 表示を禁止
1	タイマモードで LCD 表示を許可

＜注意事項＞

タイマモードで LCD 表示を許可するために , フレーム周期発生用クロック選択ビット (CSS) もまた "1" に設定し , 選択クロック (サブクロック CLKSC または CR クロック CLKRC) を有効にする必要があります。

[bit5] VSEL: LCD駆動電源を制御します。

0	内部分割抵抗を切断
1	内部分割抵抗を接続

＜注意事項＞

T 外部分割抵抗を接続するには , LCD 駆動電源制御ビット (VSEL) を "0" に設定する必要があります。
外部分割抵抗を選択するとき , 対応するセグメントを許可することはできません。対応する LCDER レジスタビットの設定は無視されます。
内部分割抵抗を選択するとき , 外部電圧端子を接続することはできません。
LCDVER:V[0:3] の設定は無視されます。

[bit4] BK: ブランキングを選択します。

0	LCD 表示許可
1	LCD 表示禁止 (ブランク)

[bit3, bit2] MS1, MS0: 表示モードを選択します。

Table 26-1. 機能設定

MS1	MS0	表示モード
0	0	LCD 動作停止
0	1	1/2 デューティ出力モード (時分割数: N=2, COM0, COM1)
1	0	1/3 デューティ出力モード (時分割数: N=3, COM0～COM2)
1	1	1/4 デューティ出力モード (時分割数: N=4, COM0～COM3)

＜注意事項＞

表示モード選択ビット (MS[1:0]) を 00_B に設定する場合 , LCD コントローラは無効になります。すなわち , "L" レベルがコモン / セグメント端子へ出力されます。

[bit1, bit0] FP1, FP0: フレーム周期

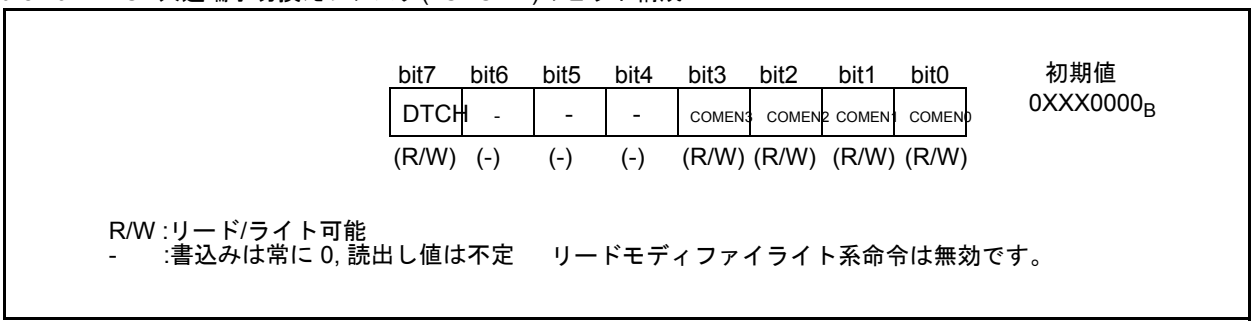
Table 26-2. フレーム周期

FP1	FP0	周辺クロック(CLKP1) が選択された場合	サブクロック CLKSC または CR クロック CLKRC が選択された場合
0	0	$F_{CLKP1}/(2^{13} \times N)$	$F_{CL}/(2^{8} \times N)$
0	1	$F_{CLKP1}/(2^{15} \times N)$	$F_{CL}/(2^{9} \times N)$
1	0	$F_{CLKP1}/(2^{17} \times N)$	$F_{CL}/(2^{10} \times N)$
1	1	$F_{CLKP1}/(2^{19} \times N)$	$F_{CL}/(2^{11} \times N)$

- F_{CLKP1}: 周辺クロック(CLKP1) 周波数
- F_{CL}: サブクロック CLKSC または CR クロック CLKRC 周波数
- N: 時分割数(表示モード選択ビットMS1 および MS0で選択)
- LCD パネルのフレーム周波数に応じた適切な値を選択してください。

26.2.2 LCD共通端子切換えレジスタ(LCDCMR)

Figure 26-7. LCD共通端子切換えレジスタ(LCDCMR)のビット構成



[bit7] DTCH: バイアス制御

0	1/3バイアス
1	予約

LCD コントローラを使用するために , LCDCMR:DTCH を "0" に設定します。

[bit6～bit4] 未使用ビット

必ず "0" を設定してください。

読出し値は不定です。

リードモディファイライト系命令は無効です。

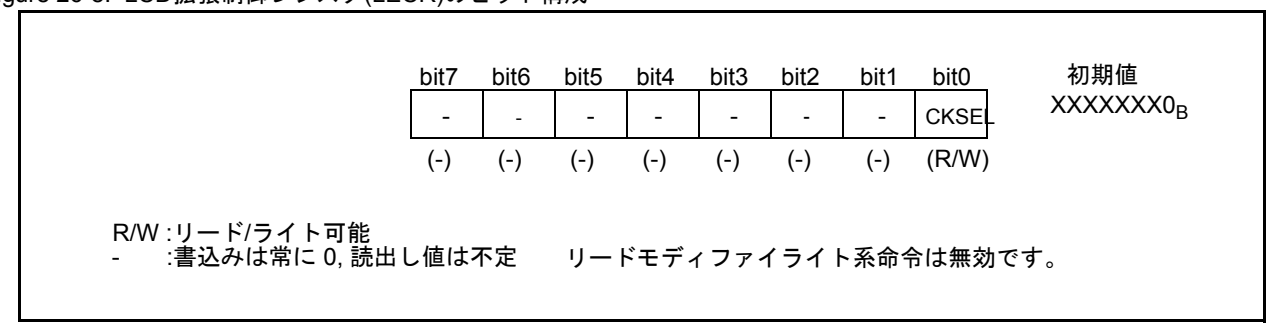
[bit3～bit0] COMEN3～COMMON0: コモンドライバ許可

0	対応するコモンドライバ (COMx) を禁止
1	対応するコモンドライバ (COMx) を許可

COMENx が "1" に設定された場合 , 対応する端子のデジタル機能 (入力 / 出力) は禁止されます。

26.2.3 LCD拡張制御レジスタ(LECR)

Figure 26-8. LCD拡張制御レジスタ(LECR)のビット構成



[bit0] CKSEL: クロック選択ビット

0	サブクロック CLKSC を使用
1	CR クロック CLKRC を使用

サブクロックまたは CR クロックを LCD クロックとして使用する場合, LCR:CSS ビットを設定する必要があります。

サブクロック付デバイス：クロックの選択を切り換える場合 , 前回の切換えから， 3 サイクル経過後にしてください。そうでないと , 新しく設定されたクロックに切り換わりません。

[bit7～bit1] 未使用ビット

必ず "0" を設定してください。

読出し値は不定です。

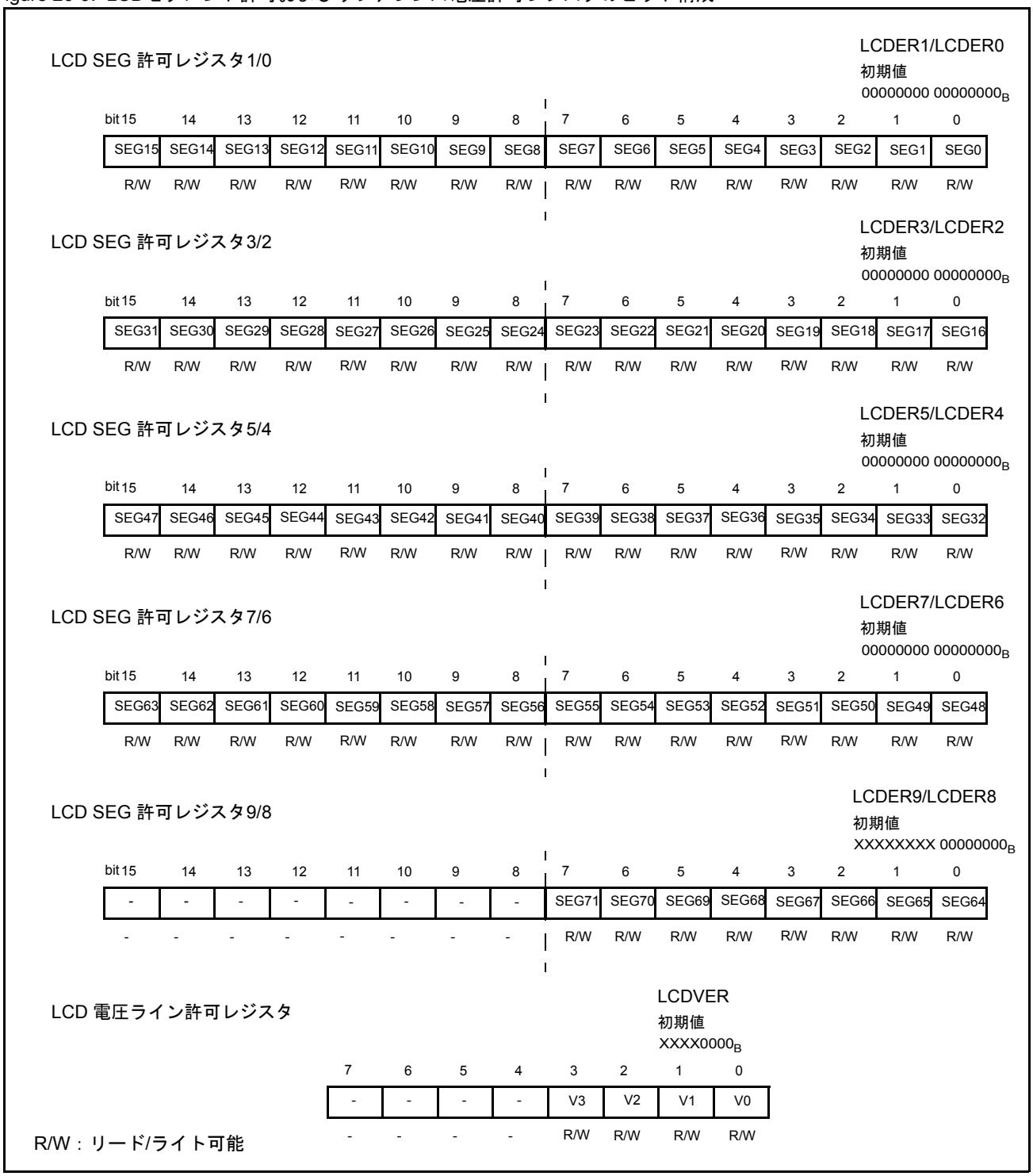
リードモディファイライト系命令は無効です。

26.3 LCD許可レジスタ(LCDER, LCDVER)

LCD 許可レジスタについて説明します。

LCDセグメント許可レジスタLCDERおよび電圧ライン許可レジスタLCDVER

Figure 26-9. LCDセグメント許可およびリファレンス電圧許可レジスタのビット構成



"0" : LCD 出力 (SEGxx) または入力 (Vx) を禁止します。端子は , ほかのリソース機能に使用されるか , 汎用 I/O として使用されます。

"1" : LCD 出力 (SEGxx) または入力 (Vx) を許可します。ほかの端子機能には使用不可 (デジタル機能は禁止) です。

"-" : 予約ビット。必ず "0" を設定してください。読出し値は不定です。リードモディファイライト系命令は無効です。

＜注意事項＞

内部分割抵抗に接続されるV端子の LCDVER:Vx = 1 に設定されている場合 , Vx 端子を共用するセグメント出力の LCD 許可レジスタ LCDERy:SEGz の設定は無効で , セグメント出力は禁止されます。これは , セグメント出力と内部分割抵抗間の短絡を防止するためです。

26.4 LCDコントローラ/ドライバの表示用RAM

表示用 RAM は, セグメント出力信号を生成するための 36 ´ 8 ビットの表示メモリ領域です。

26.4.1 表示データ設定のためのメモリ領域(VRAM)

- **VRAM0 (SEG1, SEG0): (アクセス: バイト)**

  .

  .

- **VRAM(x/2 - 1) (SEGx-1, SEGx-2): (アクセス: バイト)**

x: セグメントの最大数

7	6	5	4	3	2	1	0	bit
DH3	DH2	DH1	DH0	DL3	DL2	DL1	DL0	
X	X	X	X	X	X	X	X	初期値
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	属性

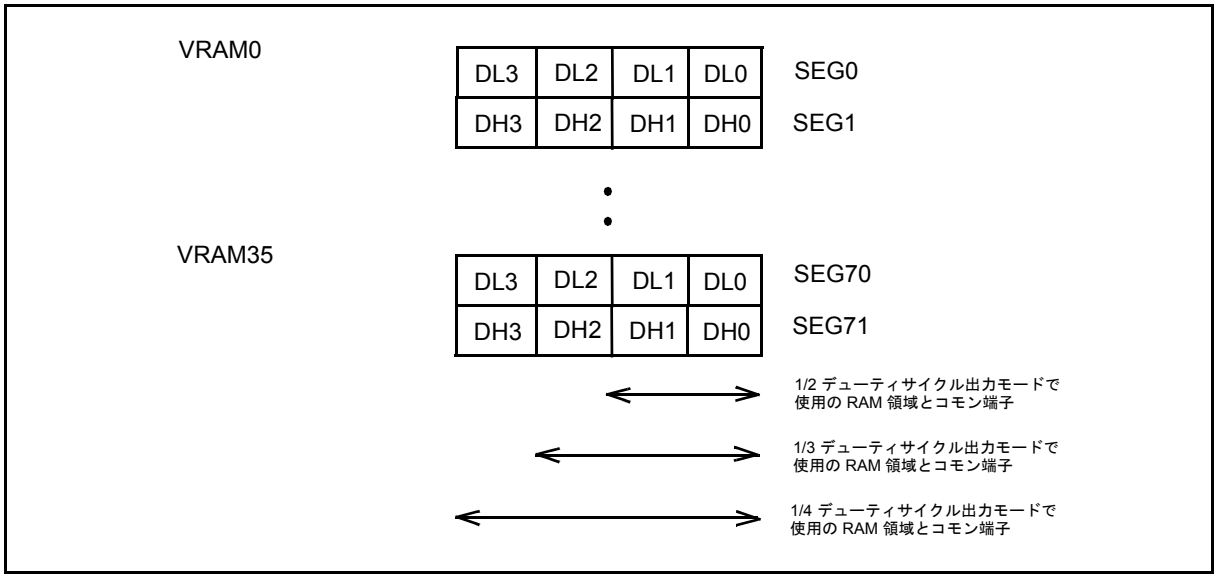
R/W : リード/ライト可能

＜注意事項＞

LCD コントローラ / ドライバの動作にかかわらず , RAM はいつでも読出し / 書込みができます。

26.4.2 VRAMとコモン/セグメント端子間の対応関係

Figure 26-10. VRAMとコモン/セグメント端子間の対応関係



26.5 LCDコントローラ/ドライバの動作

LCD コントローラ/ドライバの使用方法を段階的に説明します。

26.5.1 LCDコントローラ/ドライバ(LCDC)の動作

- あらかじめ, 表示データメモリ (VRAM) に値を設定します。
- 各レジスタへ必要な設定を行います。
- フレーム周期発生用クロックが発振するとき, LCD 駆動波形が, コモン/セグメント出力端子に出力されます (COM0 ～ COM3, SEG0 ～ SEGx-1)。
- VRAM の内容は, セグメント出力端子に出力されるコモン信号と同期して自動的に読み出されます。
  - ビットが "1" に設定された場合は, 選択した波形がセグメント出力端子に出力されます。ビットが "0" に設定された場合は, 非選択の波形がセグメント出力端子に出力されます。
- 表示モードが1/2デューティサイクルに設定されている場合, COM2 および COM3 端子の出力は,非選択レベルの出力波形となります。 1/3 デューティサイクルの場合,
  COM3 端子の出力は,非選択レベルの出力波形となります。
- 出力波形はデューティサイクル設定に応じた 2 フレーム交流化波形であり, LCD を駆動します。
- MS[1:0]=00_Bを使用し LCD を停止する場合は, コモン/セグメント出力端子は共に "L" レベルになります。
- LCD動作がタイマモード(LCEN=1)で許可される場合,LCD コントローラ/ドライバはMCU がタイマモードになる場合でも表示を継続します。
  - フレーム周期発生用クロック信号を供給する必要があることに注意してください。
- LCD はブランキング選択で "blank"(BK=1) を選択し, 非表示にすることができます。
  - 非選択レベルの出力波形となる端子は出力を継続することに注意してください。
- LCD の動作停止を表示モードで選択すると (MS[1:0]=00_B), LCD は動作を停止します。

26.5.2 LCDコントローラ/ドライバ動作中(1/2デューティサイクル)の出力波形

COM0 と COM1 の出力のみが LCD の表示に使用されます。COM2 と COM3 の出力はどちらも使用されません。

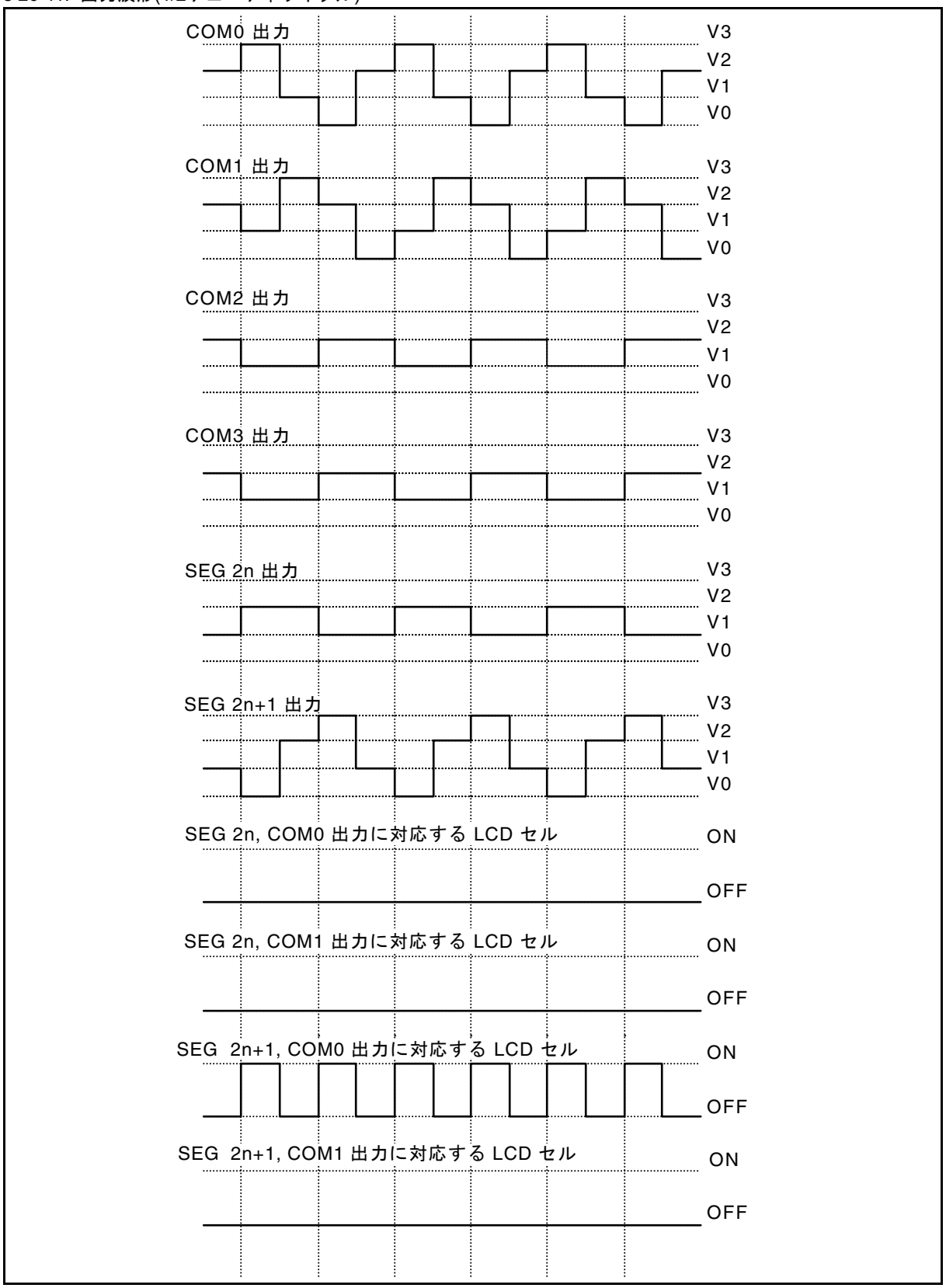
1/3バイアス出力波形の例

コモン出力とセグメント出力の電位差が最大となった LCD セルは点灯します。

Table 26-3. 表示用データメモリの内容例

セグメント	表示用データメモリの内容			
	COM3出力	COM2出力	COM1出力	COM0出力
SEG 2n出力	-	-	0	0
SEG2n+1出力	-	-	0	1

Figure 26-11. 出力波形(1/2デューティサイクル)



26.5.3 LCDコントローラ/ドライバ動作中(1/3デューティサイクル)の出力波形

1/3 デューティサイクル出力モードでは , COM0, COM1, COM2 の出力が LCD の表示に使用されます。COM3 出力は使用されません。

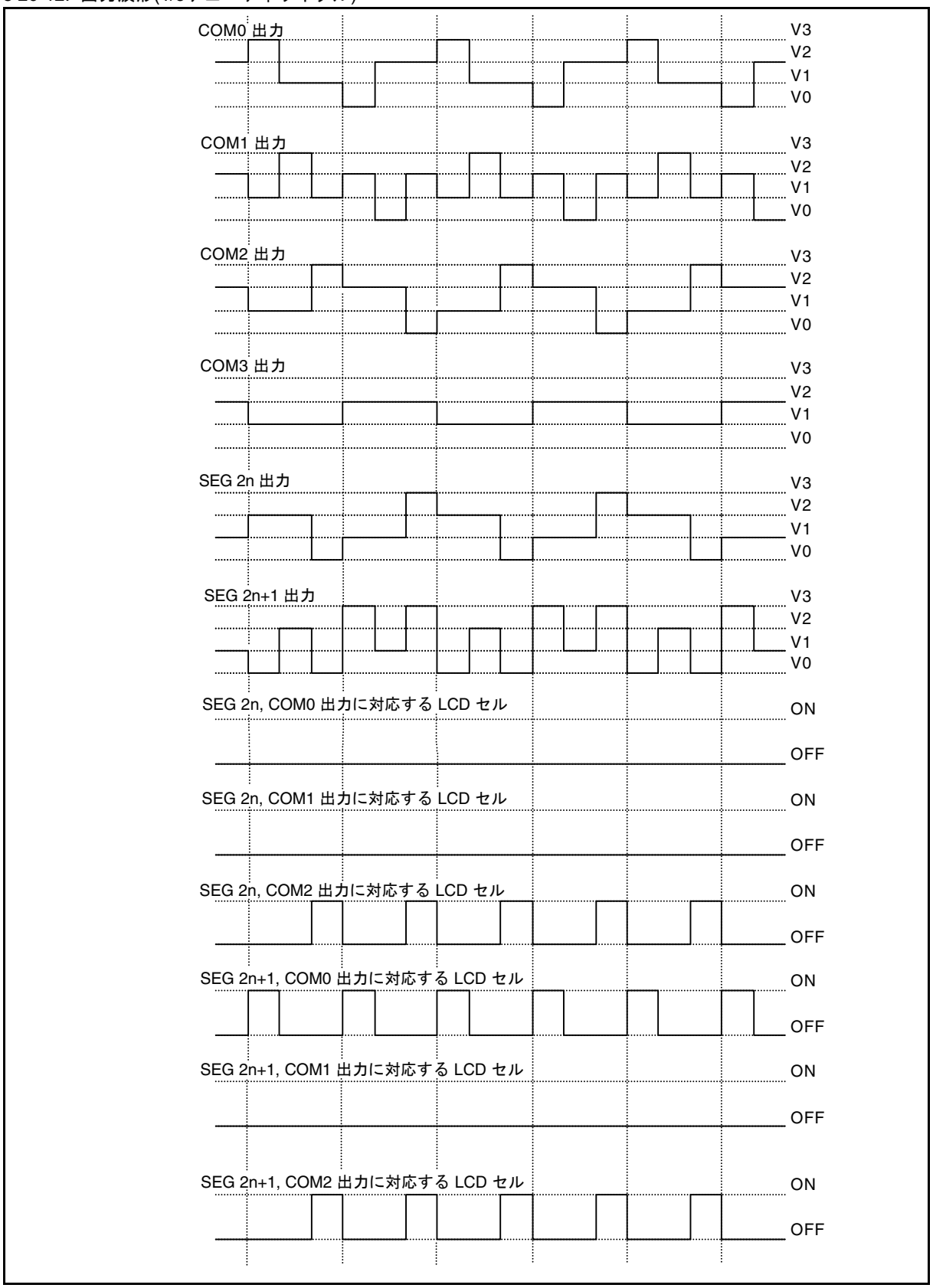
1/3バイアス出力波形の例

コモン出力とセグメント出力の電位差が最大となった LCD セルは点灯します。

Table 26-4. 表示用データメモリの内容例

セグメント	表示用データメモリの内容			
	COM3出力	COM2出力	COM1出力	COM0出力
SEG 2n出力	-	1	0	0
SEG2n+1出力	-	1	0	1

Figure 26-12. 出力波形(1/3デューティサイクル)



26.5.4 LCDコントローラ/ドライバ動作中(1/4ディーティサイクル)の出力波形

1/4 デューティサイクル出力モードでは, COM0, COM1, COM2, COM3 の出力がすべて LCD の表示に使用されます。

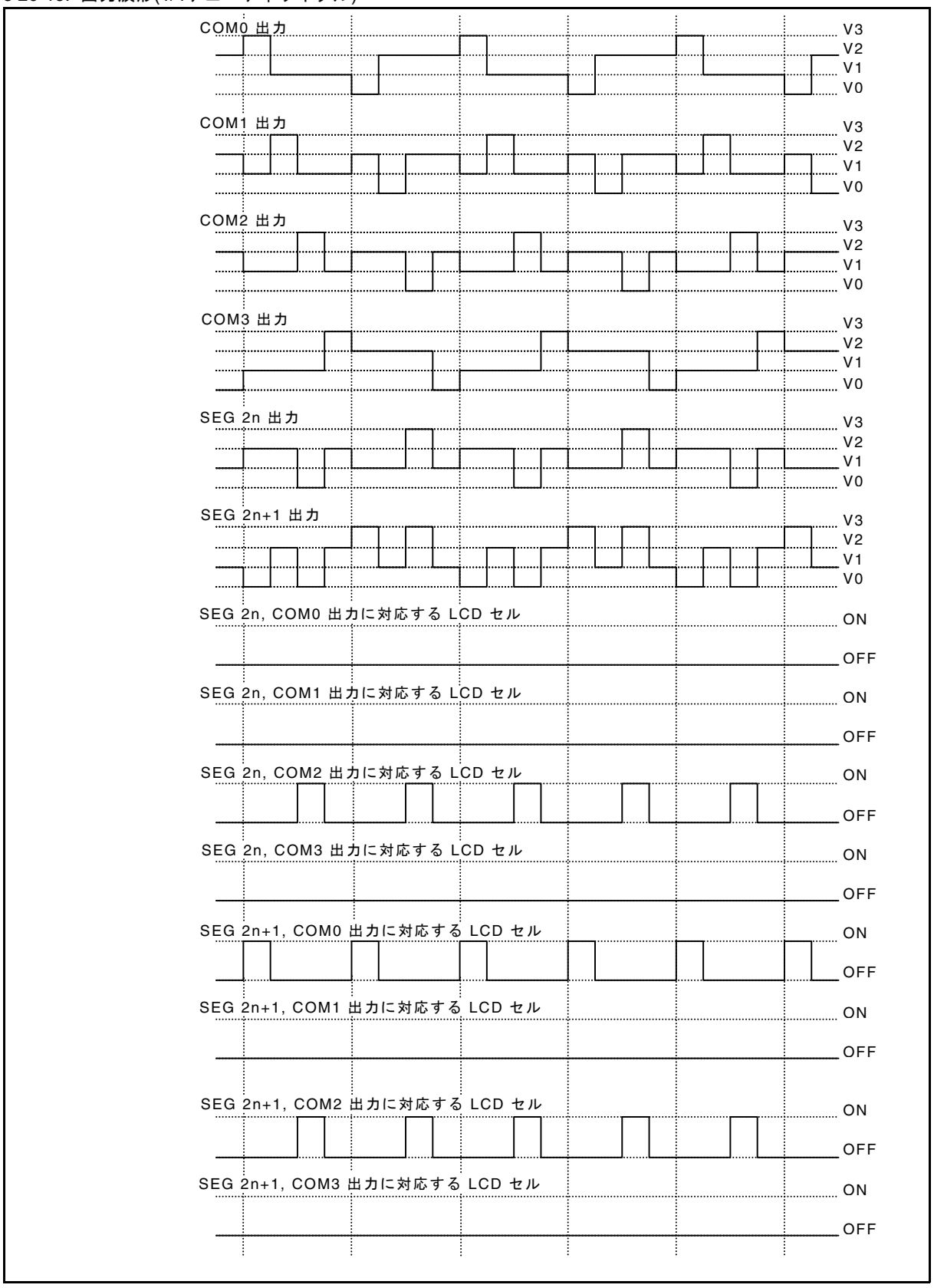
1/4バイアス出力波形の例

コモン出力とセグメント出力の電位差が最大となった LCD セルは点灯します。

Table 26-5. 表示用データメモリの内容例

セグメント	表示用データメモリの内容			
	COM3出力	COM2出力	COM1出力	COM0出力
SEG 2n出力	0	1	0	0
SEG2n+1出力	0	1	0	1

Figure 26-13. 出力波形(1/4デューティサイクル)



26.6 注意事項

- フレーム周期発生用クロックの切換え:
  - LCD の表示動作中の場合でも, フレーム周期発生用クロック (LCR:CSS) は切り換えることができます。
  - しかし, 切換えにより表示のちらつきが発生する可能性があります。 そのようなちらつきを防止するため, 切換えの前に, ブランキング選択ビット (LCR:BK) を "1" (非表示) に設定してください。
- 使用されるLCD によっては,適切な外部分割抵抗が異なります。 適切な抵抗の値を使用してください。
- 表示モードが 1/2 デューティサイクルに設定されている場合,COM2 およびCOM3 端子の出力は,非選択レベルの出力波形となります。 1/3 デューティサイクルの場合, COM3 端子の出力は,非選択レベルの出力波形となります。
- フレーム周期発生用クロック (CSS), LCD 駆動電源制御 (VSEL), デューティサイクル (MS[1:0]), フレーム周期 (FP1, FP0) の選択または設定が適切でない場合, LCD の表示が適切には動作しません。
- 初期設定では, MCU タイマモードでLCD表示は禁止です。 MCU タイマモードで LCD 表示を許可するには, LCR:LCEN を使用し, 必要なクロックが動作していることを確認してください。
- LCD 表示は MCU 停止モードで禁止です。
- 必要なクロックが動作していない場合, LCD 表示を許可しないように注意してください。 そうでなければ, LCD パネルが DC 電圧により損傷する可能性があります。 以下の表を参照してください。

Table 26-6. MCUタイマモードまたはMCU停止モード以外でのLCD動作

CSS	CKSEL	サブクロックCLKSC動作中	RCクロックCLKRC動作中	MS[1:0] [1] "00"の動作
0	X	X	X	LCD コントローラは周辺クロック CLKP1で動作中
1	0	いいえ	X	この設定は使用禁止です。 DC 電圧を LCD パネルへ印加してしまいます。
1	0	はい	X	LCD コントローラはサブクロック CLKSCで動作中
1	1	X	No	この設定は使用禁止です。 DC 電圧を LCD パネルへ印加してしまいます。
1	1	X	Yes	LCD コントローラは CR クロック CLKRCで動作中

Table 26-7. MCUタイマモードでのLCD動作

CSS	LCEN	CKSEL	サブクロックCLKSC動作中	RCクロックCLKRC動作中	MS[1:0] ¹ "00"の動作
0	0	X	X	X	LCD 表示停止
0	1	X	X	X	この設定は使用禁止です。DC電圧をLCDパネルへ印加してしまいます。
1	0	X	X	X	LCD 表示停止
1	1	0	No	X	この設定は使用禁止です。DC電圧をLCDパネルへ印加してしまいます。
1	1	0	Yes	X	LCD コントローラはサブクロック CLKSCで動作中
1	1	1	X	No	この設定は使用禁止です。DC電圧をLCDパネルへ印加してしまいます。
1	1	1	X	Yes	LCD コントローラは CR クロック CLKRCで動作中

MS[1:0] =00_B のときと , MCU 停止モードのときは , LCD 表示は停止します。

27. ステッパモータコントローラ

ステッパモータコントローラの機能および動作について説明します。

27.1 ステッパモータコントローラの概要

ステッパモータコントローラは, PWM パルス発生器, モータドライバ, セレクタロジック回路により構成されています。

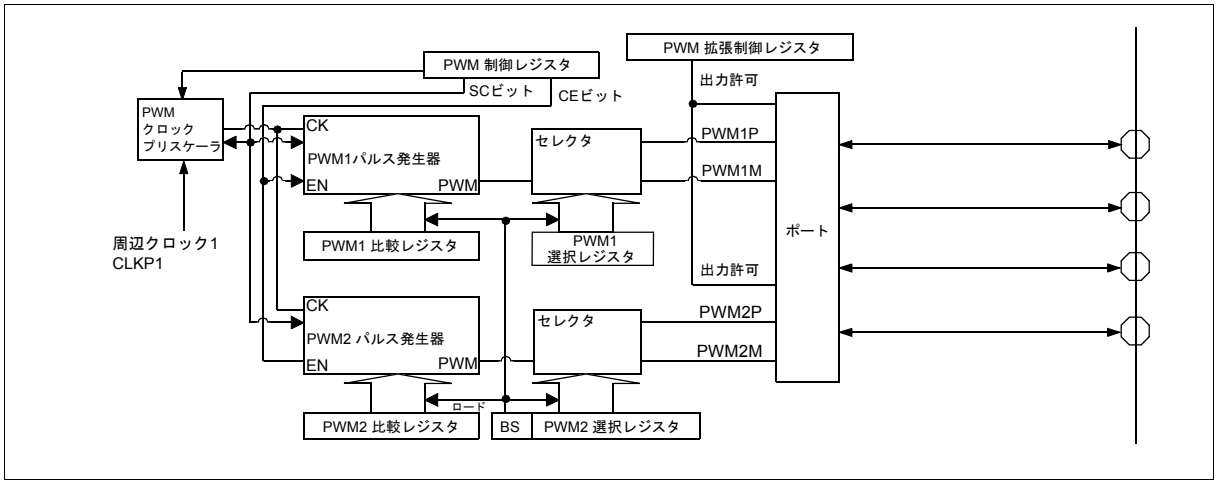
概要

4 個のモータドライバには大電流ドライブ機能があり , 2 個のモータコイルを 4 端子に直接接続できます。モータの回転は , PWM パルス発生器とセレクタロジック回路を組み合せることによって制御されるよう設計されています。同期のメカニズムにより , 2 個の PWM パルス発生器が確実に同期動作します。

ステッパモータコントローラのブロックダイヤグラム

Figure 27-1 に , ステッパモータコントローラのブロックダイヤグラムを示します。

Figure 27-1. ステッパモータコントローラのダイヤグラム



27.2 ステッパモータコントローラのレジスタ

ステッパモータコントローラには 6 つのレジスタがあります。

- PWM 制御レジスタ
- PWM 拡張制御レジスタ
- PWM1 比較レジスタ
- PWM2 比較レジスタ
- PWM1 選択レジスタ
- PWM2 選択レジスタ

ステッパモータコントローラのレジスタ

Figure 27-2. ステッパモータコントローラレジスタの概要

PWM 制御レジスタ (PWCn)

bit	7	6	5	4	3	2	1	0
	–	P2	P1	P0	CE	SC	–	–
	–	R/W	R/W	R/W	R/W	R/W	–	–
	–	0	0	0	0	0	–	–

PWM 拡張制御レジスタ (PWECn)

bit	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	OE2	OE1
	–	–	–	–	–	–	R/W	R/W
	X	X	X	X	X	X	0	0

PWM1 比較レジスタ (PWC1n)

bit	15	14	13	12	11	10	9	8
	–	–	–	–	–	–	D9	D8
	–	–	–	–	–	–	R/W	R/W
	–	–	–	–	–	–	X	X

bit	7	6	5	4	3	2	1	0
	D7	D6	D5	D4	D3	D2	D1	D0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	X	X	X	X	X	X	X	X

PWM2 比較レジスタ (PWC2n)

bit	15	14	13	12	11	10	9	8
	–	–	–	–	–	–	D9	D8
	–	–	–	–	–	–	R/W	R/W
	–	–	–	–	–	–	X	X

bit	7	6	5	4	3	2	1	0
	D7	D6	D5	D4	D3	D2	D1	D0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	X	X	X	X	X	X	X	X

PWM1 選択レジスタ (PWS1n)

bit	7	6	5	4	3	2	1	0
	–	–	P2	P1	P0	M2	M1	M0
	–	–	R/W	R/W	R/W	R/W	R/W	R/W
	–	–	0	0	0	0	0	0

PWM2 選択レジスタ (PWS2n)

bit	7	6	5	4	3	2	1	0
	–	BS	P2	P1	P0	M2	M1	M0
	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	–	X	0	0	0	0	0	0

R/W ：リード/ライト可能
X ：不定値
－ ：未定義ビット

27.3 PWM制御レジスタ(PWCn)

PWM 制御レジスタ (PWCn) はステッパモータコントローラの開始と停止, 割込みの制御, 外部出力端子の設定を行います。

27.3.1 PWM制御レジスタ(PWCn)

Figure 27-3. PWM制御レジスタ(PWCn)

PWM 制御レジスタ (PWCn)

bit	7	6	5	4	3	2	1	0	
	-	P2	P1	P0	CE	SC	-	-	
	-	R/W	R/W	R/W	R/W	R/W	-	-	← 属性
	-	0	0	0	0	0	-	-	← 初期値

R/W : リード/ライト可能

[bit 7] 予約ビット

[bit 6～bit 4] P2, P1, P0: 動作クロック選択ビット

P2, P1, P0 ビットは, PWM パルス発生器のためのクロック入力信号を指定します。

P2	P1	P0	クロック入力	PWMサイクル(f_{CLKP1} = 32 MHz)		PWMサイクル(f_{CLKP1} = 40 MHz)	
				SC=0	SC=1	SC=0	SC=1
0	0	0	f_{CLKP1}	8.0 us	32.0 us	6.4 us	25.6 us
0	0	1	$f_{CLKP1}/4$	32.0 us	128.0 us	25.6 us	102.4 us
0	1	0	$f_{CLKP1}/5$	40.0 us	160.0 us	32.0 us	128.0 us
0	1	1	$f_{CLKP1}/6$	48.0 us	192.0 us	38.4 us	153.6 us
1	0	0	$f_{CLKP1}/8$	64.0 us	256.0 us	51.2 us	204.8 us
1	0	1	$f_{CLKP1}/10$	80.0 us	320.0 us	64.0 us	256.0 us
1	1	0	$f_{CLKP1}/12$	96.0 us	384.0 us	76.8 us	307.2 us
1	1	1	$f_{CLKP1}/16$	128.0 us	512.0 us	102.4 us	409.6 us

＜注意事項＞

動作クロック選択後 , CE を "1" に設定してください。

[bit 3] CE: カウント許可ビット

CE ビットは , PWM パルス発生器の動作を許可します。"1" に設定すると , PWM パルス発生器は動作を開始します。出力ドライバから発生するスイッチのノイズを減少させるために , PWM1 パルス発生器の動作開始の後に PWM2 パルス発生器が動作開始します。

CE ビットが , PWM パルス発生器の動作中に "0" クリアすると , パルス発生器は初期化されて , 動作を停止します。

[bit 2] SC: 8/10ビット切換えビット

SC ビットを"1" に設定すると, PWM パルス発生器は 10 ビットで動作します。SC ビットを"0" に設定すると, PWM パルス発生器は 8 ビットで動作します。

[bit 1～bit 0] 予約ビット

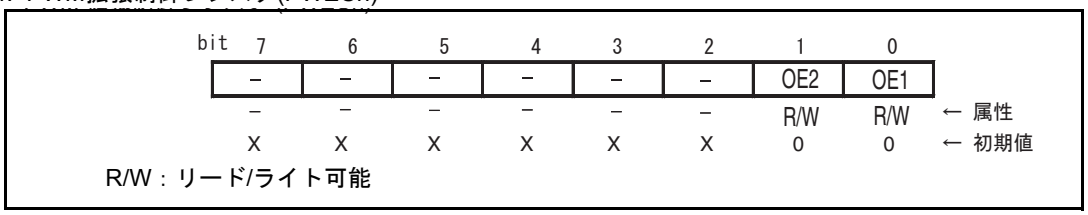
常に "0" を書き込んでください。読出し値は不定です。リードモディファイライト系命令は無効です。

27.4 PWM拡張制御レジスタ(PWECn)

PWM 拡張制御レジスタ (PWECn) は, PWM 出力を許可します。

27.4.1 PWM拡張制御レジスタ(PWECn)

Figure 27-4. PWM拡張制御レジスタ(PWECn)



[bits 7～bit 2] 未使用ビット

このビットへの書込みは , 影響しません。

[bit 1, bit 0] OE2, OE1: PWM出力許可ビット

PWM 出力を許可 / 禁止します。

OE1 は , PWM1Pn/PWM1Mn の出力を許可します。

OE2は, PWM2Pn/PWM2Mnの出力を許可します。

Table 27-1. 動作

OE2, OE1	動作
0	PWM1Pn/PWM1Mnの出力を禁止
1	PWM1Pn/PWM1Mnの出力を許可

27.5 PWM1, PWM2比較レジスタ(PWC1n, PWC2n)

PWM1, PWM2 の 2 つの 8(10) ビット比較レジスタ (PWC1n, PWC2n) の値は , PWM パルスの幅を決定します。記憶された 00_H に設定した場合はデューティが 0%, FF_H ($3FF_H$) に設定した場合は 99.6% (99.9%) であることを表します。

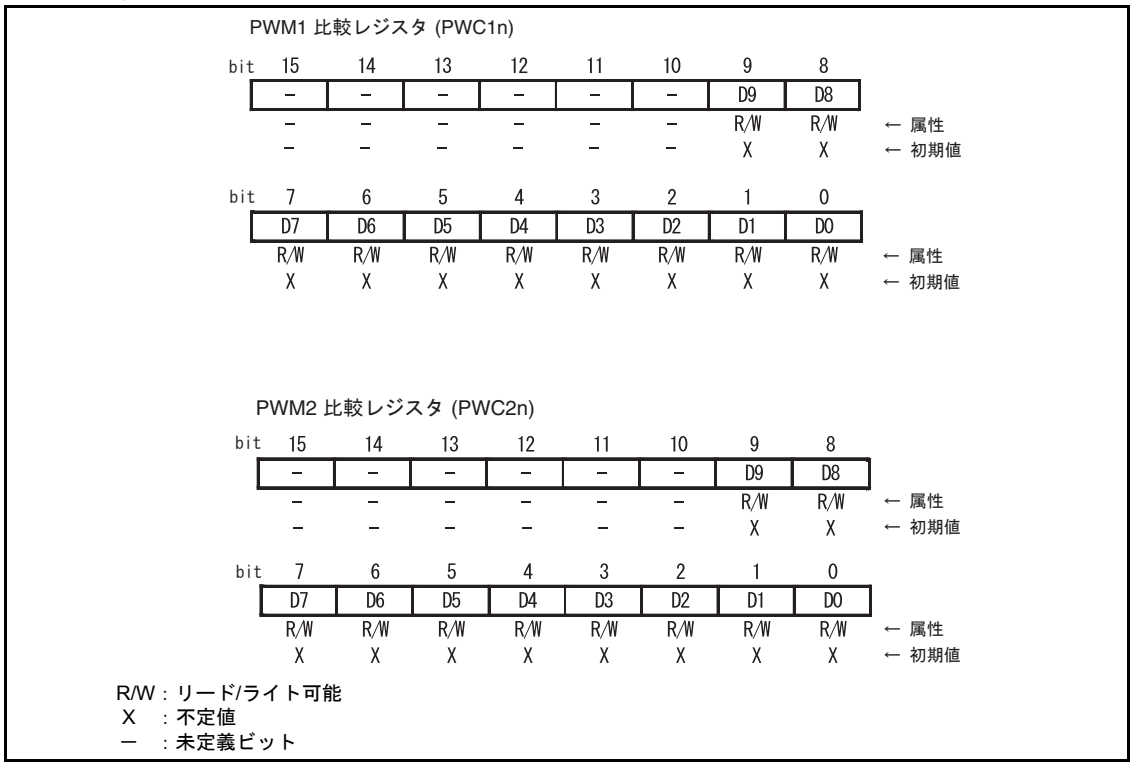
27.5.1 PWM1, PWM2比較レジスタ(PWC1n, PWC2n)

PWM1, PWM2 比較レジスタは常時アクセス可能ですが , 変更された値は PWM2 選択レジスタの BS ビットが "1" に設定された後で , 現行の PWM サイクル終了時にパルス幅に反映されます。

SCビットが"0"に設定されているときに, PWMが8ビット動作を行うと, D9 と D8 のビットは不定値となります。

PWM1, PWM2 比較レジスタは , 16 ビット幅で読出し / 書込みを行ってください。

Figure 27-5. PWM1, PWM2比較レジスタ



[bit 15～bit 10] 予約ビット

常に "0" を書き込んでください。

[bits 9～bit 0] D9 ～ D0: 比較データ

PWM のパルス幅の設定に使用されます。

Figure 27-6. 比較レジスタの設定値とPWMパルス幅の関係

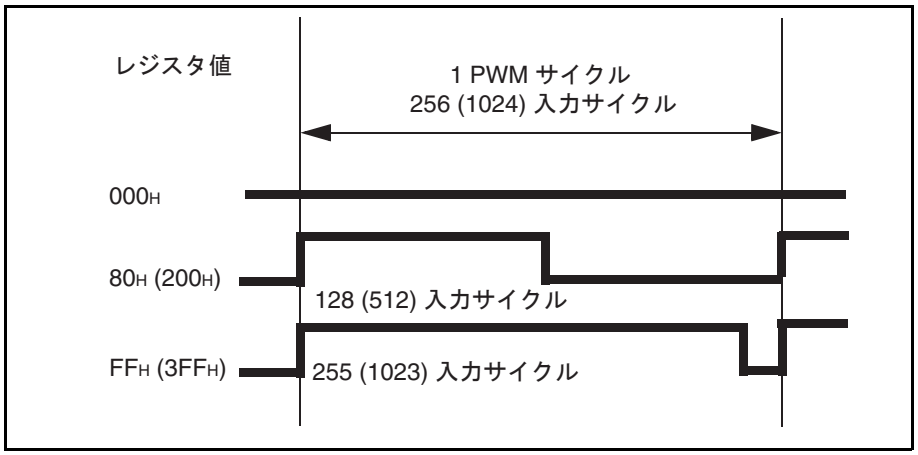
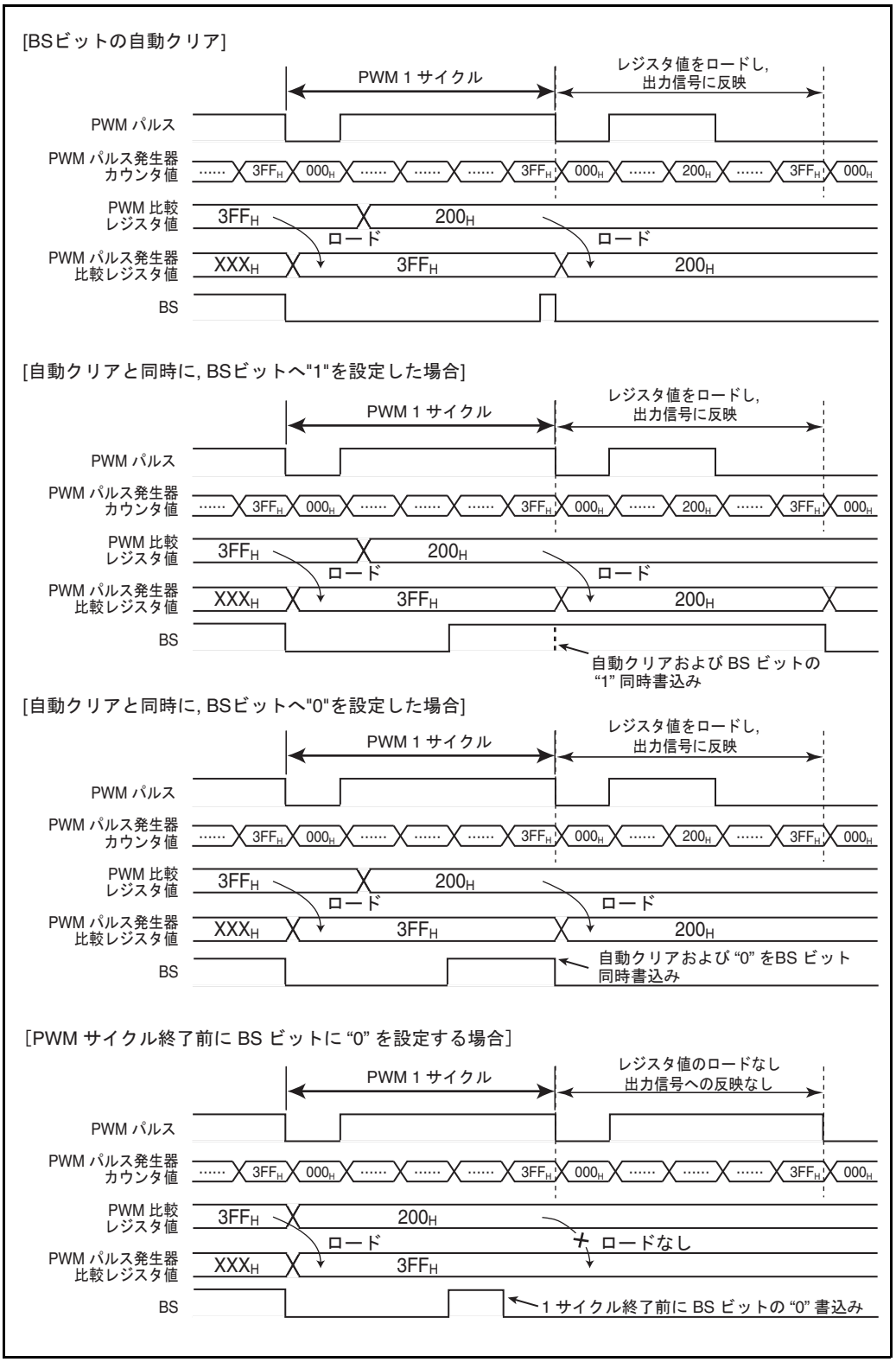


Figure 27-7. PWM比較レジスタ値のロードタイミング



27.6 PWM1, PWM2選択レジスタ(PWS1n, PWS2n)

PWM1, PWM2 選択レジスタ (PWS1n, PWS2n) は, ステッパモータコントローラの外部端子出力を "0" , "1" , PWM パルス, ハイインピーダンスのうちどれに設定するかを決定します。

27.6.1 PWM1, PWM2選択レジスタ(PWS1n, PWS2n)

Figure 27-8. PWM1, PWM2選択レジスタ(PWS1n, PWS2n)

PWM1 選択レジスタ (PWS1n)

bit	7	6	5	4	3	2	1	0	
	-	-	P2	P1	P0	M2	M1	M0	
	-	-	R/W	R/W	R/W	R/W	R/W	R/W	← 属性
	0	0	0	0	0	0	0	0	← 初期値

PWM2 選択レジスタ (PWS2n)

bit	7	6	5	4	3	2	1	0	
	-	BS	P2	P1	P0	M2	M1	M0	
	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W	← 属性
	-	0	0	0	0	0	0	0	← 初期値

R/W : リード/ライト可能
X : 不定値
ー : 未定義ビット

27.6.2 PWM1選択レジスタ(PWS1n)

[bit 7～bit 6] 予約ビット

常に "0" を設定してください。

[bit 5～bit 3] P2 ～ P0: 出力選択ビット

SMCmPn における出力信号を選択します。

[bit 2～bit 0] M2 ～ M0: 出力選択ビット

SMCmMn における出力信号を選択します。

下記の表は, 出力レベルと選択ビットの関係を示します。

P2	P1	P0	PWMmPn	M2	M1	M0	PWMmMn
0	0	0	L	0	0	0	L
0	0	1	H	0	0	1	H
0	1	X	PWMパルス	0	1	X	PWMパルス
1	X	X	ハイインピーダンス	1	X	X	ハイインピーダンス

m = 1, 2 (モータコイル)

n = 0 ～ 5 (ステッパモータチャネル)

27.6.3 PWM2選択レジスタ(PWS2n)

[bit 7] 予約ビット

常に "0" を設定してください。

[bit 6] BS: 再書込みビット

BS ビットは PWM 出力に対する設定を同期させるために用意されています。BS ビットが "1" に設定されるまで , 2 つの比較レジスタと 2 つの選択レジスタへの変更は出力信号へ反映されません。

BS ビットが "1" に設定されると , PWM パルス発生器とセレクタは , 現行の PWM サイクルの終了時にレジスタの値をロードします。BS ビットは , 次の PWM サイクルの始めに自動的に "0" にクリアされます。

BS ビットが自動クリアと同時にソフトウェアにより "1" に設定されるときは , BS ビットは "1" に設定され (BS ビットに変化なし), 自動クリアが取り消されます。

BS ビットが自動クリアと同時にソフトウェアにより "0" に設定される場合, BS ビットは "0" に設定されますが , PWM パルス発生器とセレクタは , 現行の PWM サイクルの終了時にレジスタの値をロードしません。

＜注意事項＞

BS = 1 の場合 , リードモディファイライト命令はこのビットを "1" として読み出し , そしてこのビットへ "1" の値を書き込みます。
読出しと書込み間で、BS ビットが PWM サイクルの開始により自動クリアされた場合は ,
リードモディファイライト命令はクリアされた BS ビット に再び "1" を書き込みます。
したがって , BS ビットが次の PWM サイクルの終了時に "1" に設定されていない場合でもレジスタの値を PWM パルス発生器とセレクタはロードします。

[bit 5〜bit 3] P2 〜 P0: 出力選択ビット

SMCmPn のための出力信号を選択します。

[bit 2〜bit 0] M2 〜 M0: 出力選択ビット

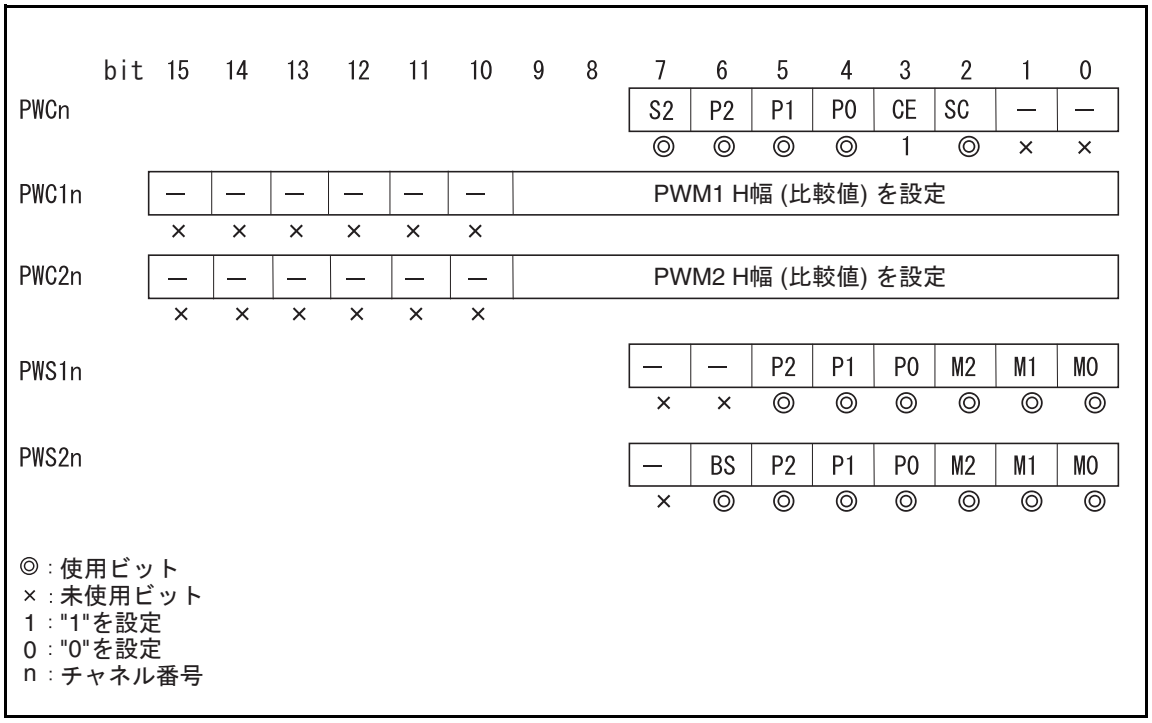
SMCmMn のための出力信号を選択します。

27.7 ステッパモータコントローラの動作

ステッパモータコントローラの動作を説明します。

ステッパモータコントローラの動作設定

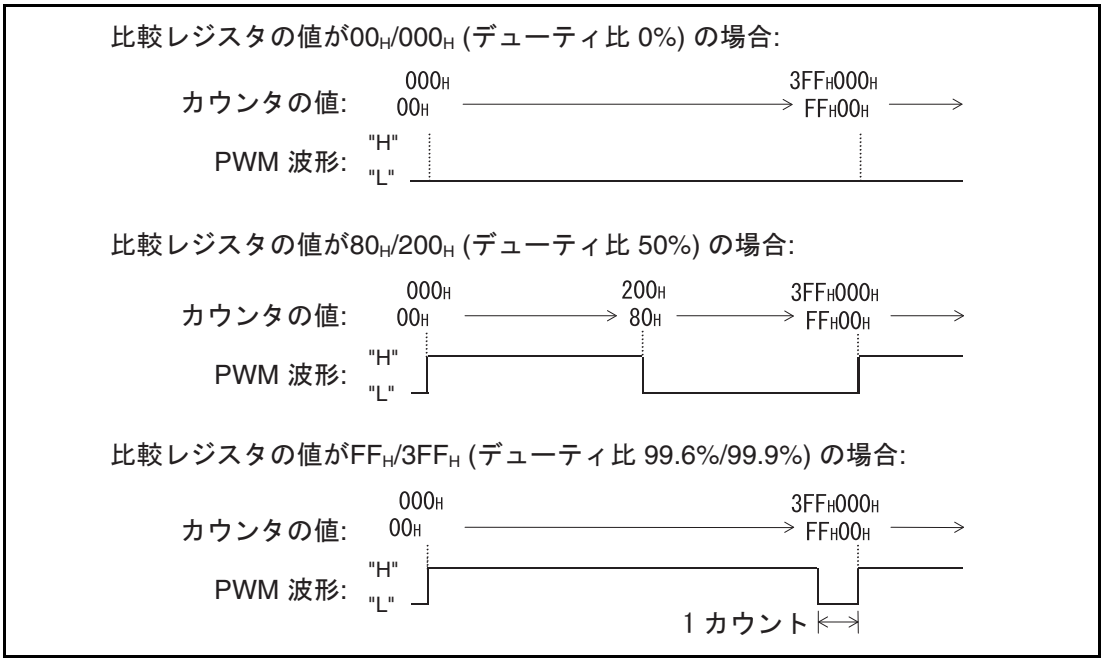
Figure 27-9. ステッパモータコントローラの設定



PWMパルス発生器の動作

カウンタが開始されると (PWC: CE = 1), 選択されたカウントクロック立上りで 00_H からのインクリメントをカウンタは開始します。PWM 出力パルス波はカウンタの値が PWM 比較レジスタの設定値と一致するまで "H" のままであり , "L" に変わった後はカウンタの値がオーバフロー (FF_H --> 00_H) するまで "L" のままです。Figure 27-10 に PWM 発生器で発生した PWM 波形を示します。

Figure 27-10. PWM1, PWM2出力波形の例



モータ駆動信号の選択

ステッパモータコントローラの各端子へ出力されるモータ駆動信号は , PWM 選択レジスタで設定します。Table 27-2 は , モータ駆動信号の選択および PWM 選択レジスタ 1, 2 の設定の一覧です。

これらのレジスタを設定し, PWM 選択レジスタ 2 の BS ビットに"1" を書き込むと, 現行の PWM サイクルの終了時に, これらのレジスタの設定が許可されます。 BS ビットは, 次の PWM サイクルの始めに自動的に 0 にクリアされます。 BS ビットへの "1" の書込みと同時に, 次の PWM サイクルの始めに BS ビットが "0" へクリアされた場合, "1" が BS ビットへ書き込まれ, クリアは取り消されます。

Table 27-2. モータ駆動信号の選択およびPWM選択レジスタ1, 2の設定

P2, P1, P0ビット	PWM2P出力 PWM1P出力	M2, M1, M0ビット	PWM1M出力 PWM2M出力
000B	L	000_B	L
001B	H	001_B	H
01XB	PWMパルス	$01X_B$	PWMパルス
1XXB	ハイインピーダンス	$1XX_B$	ハイインピーダンス

27.8 ステッパモータコントローラの使用上の注意

ステッパモータコントローラの使用上の注意点を以下に説明します。

PWMの設定を変更するときの注意

PWM 比較レジスタ 1 (PWC1x), PWM 比較レジスタ 2 (PWC2x), PWM 選択レジスタ 1 (PWS1x), PWM 選択レジスタ 2 (PWS2x) は常時アクセス可能です。PWM の "H" 幅設定または PWM 出力を変更するためには , これらのレジスタに設定値を書き込んだ後 (または同時に), PWM 選択レジスタ 2 の BS ビットに "1" を書き込みます。

BS ビットに "1" が設定されると , 現行の PWM サイクルの終了時に新しい設定値が有効になり , BS ビットは自動的にクリアされます。

BS ビットへの "1" 設定と , PWM サイクル終了時の BS ビットのリセットが同時に行われた場合は , "1" の書込みが優先され , BS ビットのリセットは取り消されます。

28. サウンドジェネレータ

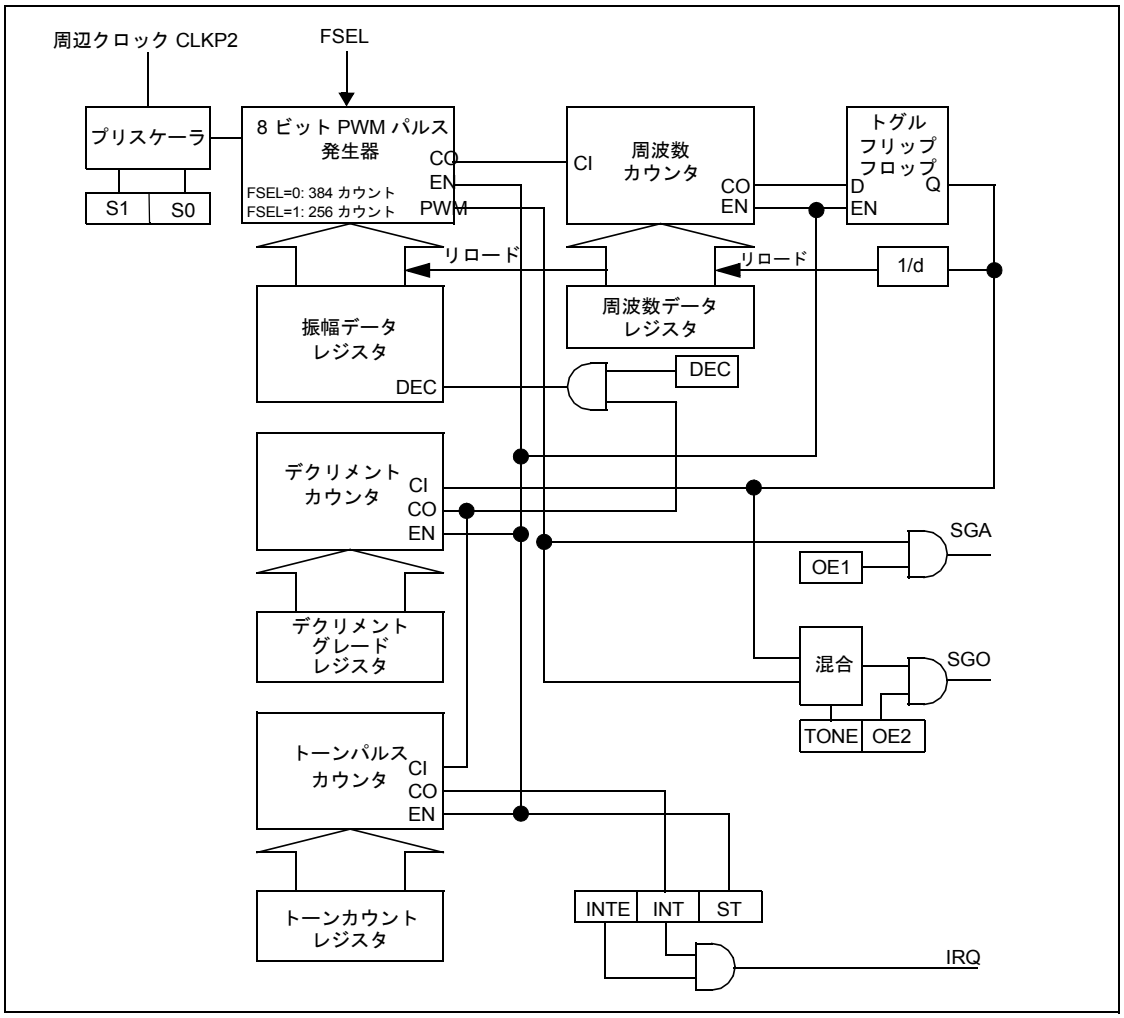
サウンドジェネレータの機能および動作について説明します。

28.1 サウンドジェネレータの概要

サウンドジェネレータは，サウンドコントロールレジスタ，周波数データレジスタ，振幅データレジスタ，デクリメントグレードレジスタ, トーンカウントレジスタ, PWM パルス発生器,周波数カウンタ, デクリメントカウンタ,トーンパルスカウンタで構成されています。

サウンドジェネレータのブロックダイヤグラム

Figure 28-1. サウンドジェネレータのブロックダイヤグラム



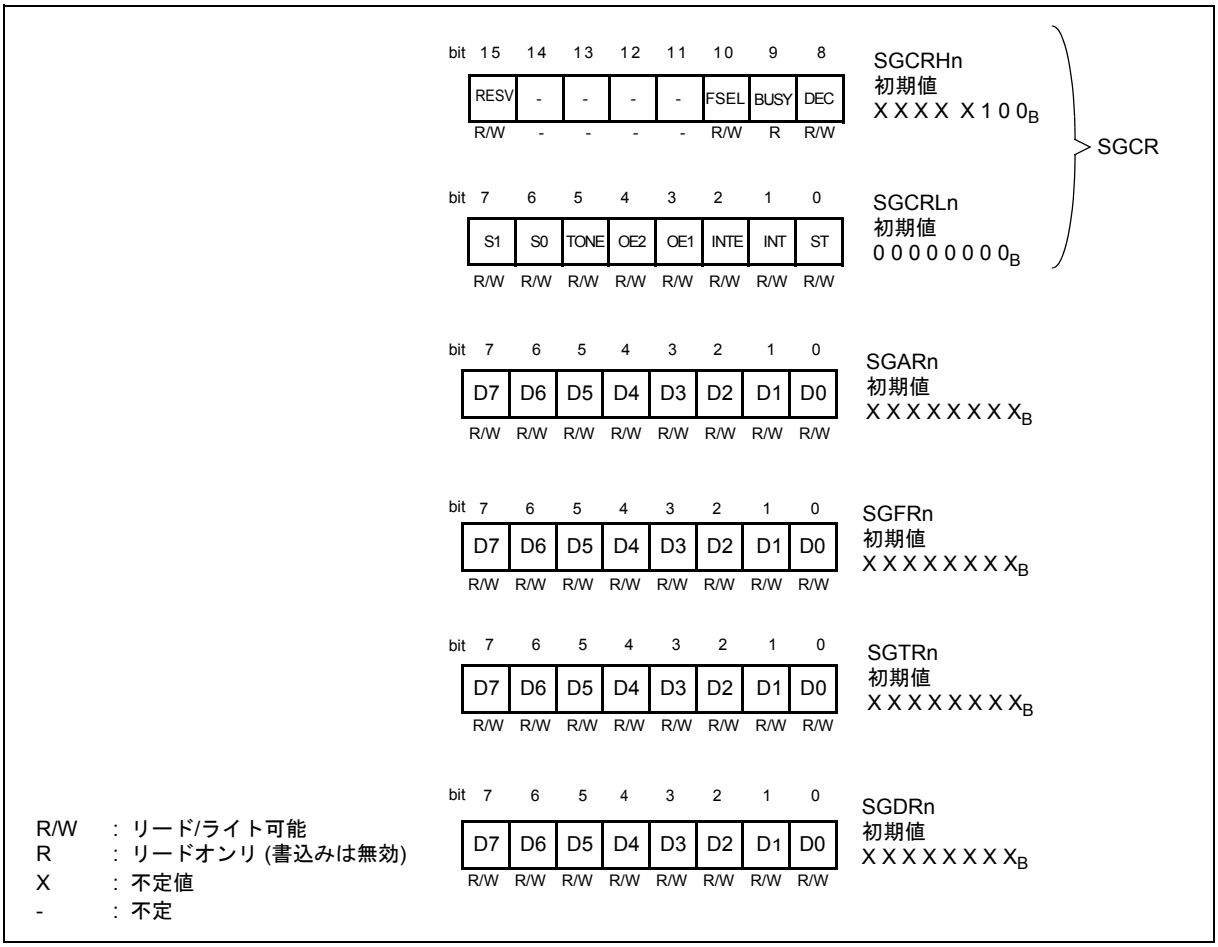
28.2 サウンドジェネレータのレジスタ

サウンドジェネレータには以下のレジスタがあります。

- サウンドコントロールレジスタ(SGCRn)
- 周波数データレジスタ(SGFRn)
- 振幅データレジスタ(SGARn)
- デクリメントグレードレジスタ(SGDRn)
- トーンカウントレジスタ(SGTRn)

サウンドジェネレータのレジスタ

Figure 28-2. サウンドジェネレータのレジスタの概要



＜注意事項＞

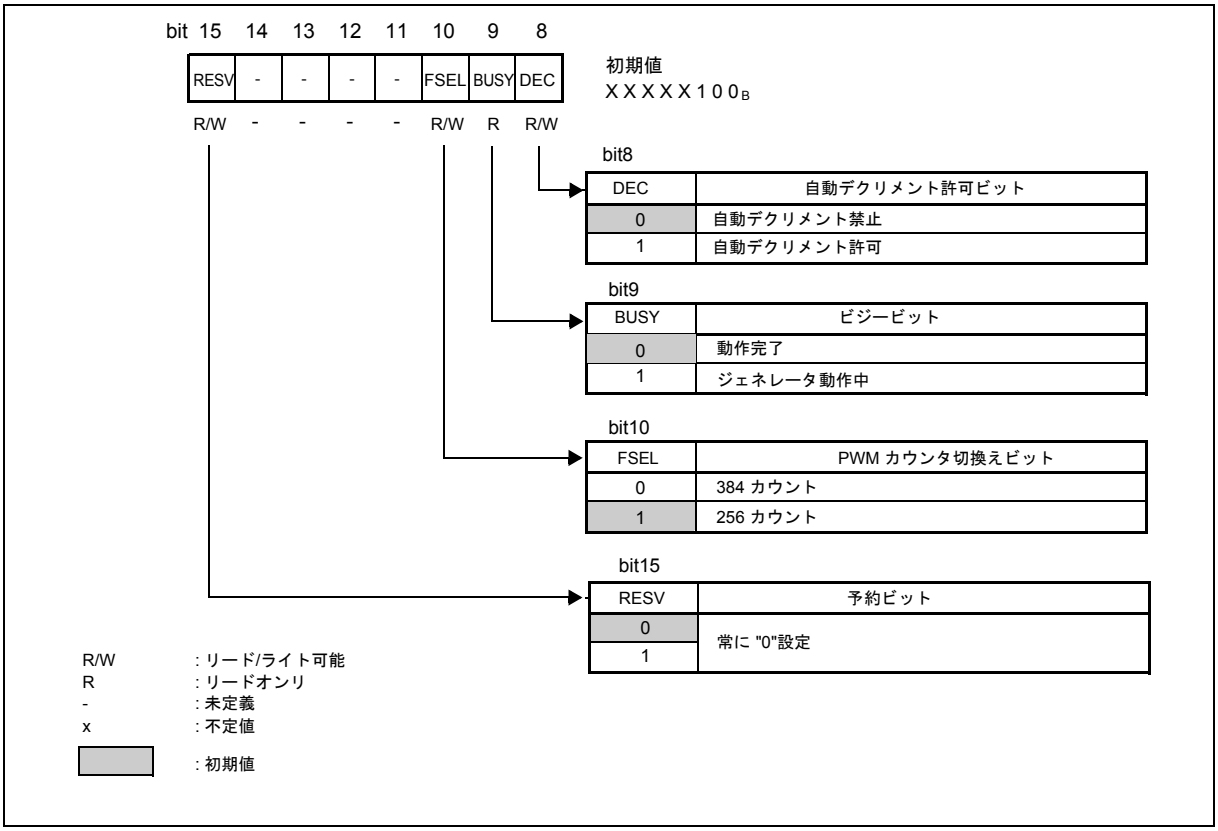
サウンドコントロールレジスタ SGCRn は , 16 ビットレジスタまたは 2 つの 8 ビットレジスタ SGCRHn, SGCRLn としてアクセスすることが可能です。

28.2.1 サウンドコントロールレジスタ(SGCRn)

サウンドコントロールレジスタ(SGCRn)は, 割込みの制御と動作状態の制御を行います。

サウンドコントロールレジスタ(SGCRHn)

Figure 28-3. サウンドコントロールレジスタ(SGCRHn)の構成



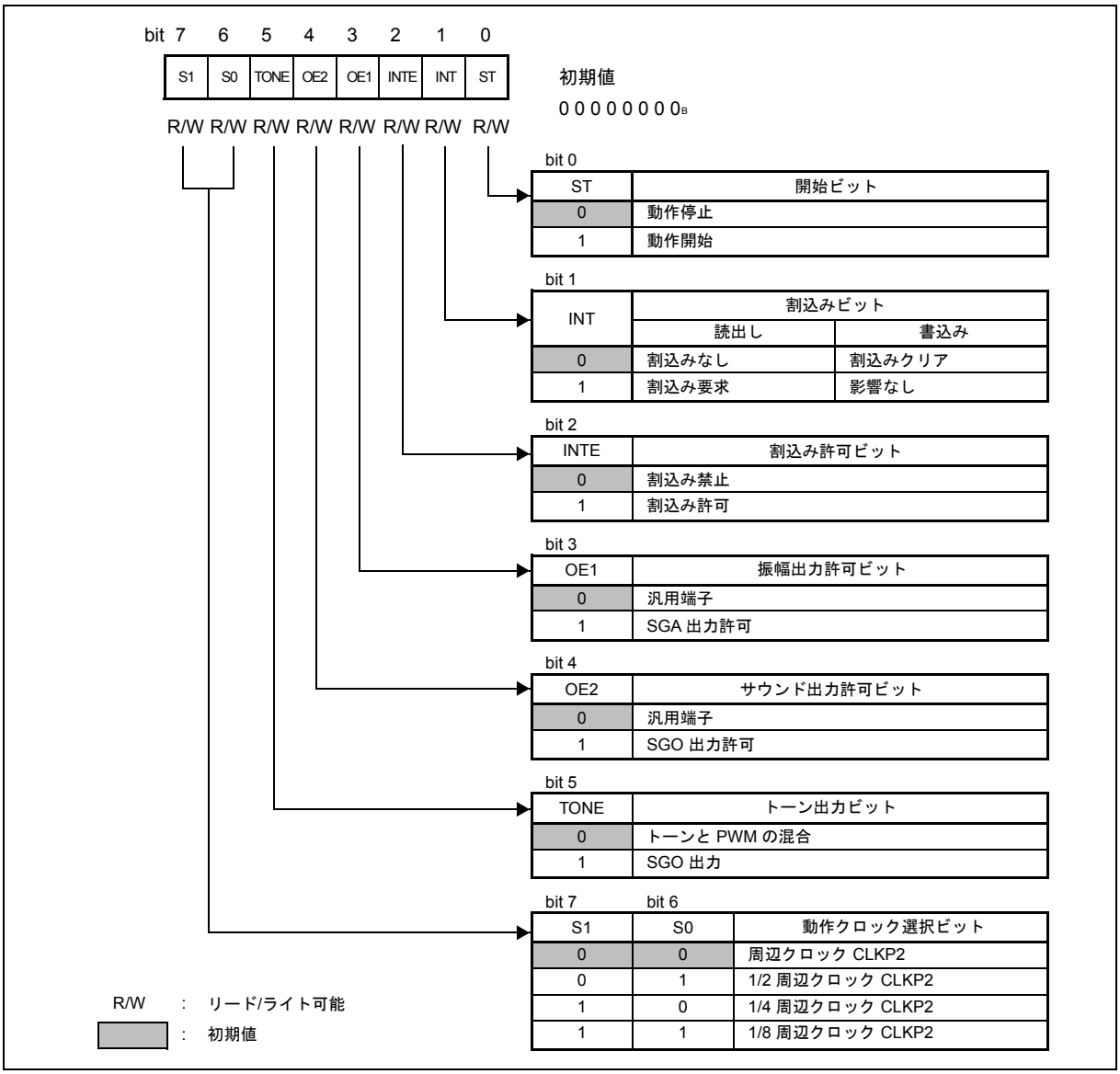
サウンドコントロールレジスタ(SGCRHn)の内容

Table 28-1. サウンドコントロールレジスタ(SGCRHn)の各ビットの機能

名前		機能
bit 15	予約ビット	予約ビットです。常に"0"を書き込んでください。読出し値は不定となります。またリードモディファイライト系命令は無効です。
bit 14 to 11	未定義	
bit 10	FSEL: PWMカウンタ切換えビット	PWMカウンタのカウント値を256 または 384に切り換えます。カウント値を切り換えるときは動作を停止してください。またFSELを切り換えた後で,振幅データ, 周波数データ, デクリメントグレードおよびトーンレジスタを再度設定してください。 0: 384 カウント 1: 256 カウント
bit 9	BUSY: ビジービット	サウンドジェネレータが動作中であるかどうかを示します。 ST ビットを "1" に設定すると, このビットも "1" に設定されます。 ST ビットを "0" にリセットし, 1 トーンサイクル終了時に動作が完了すると, "0" にリセットされます。 このビットへの書込みは無効です。
bit 8	DEC: 自動 デクリメント許可ビット	サウンドの自動デクリメント用ビットです。 DEC ビットが "1" でデクリメントカウンタがトーンパルス数をリロード値までカウントすると,振幅データレジスタに格納された値は,トーンサイクル終了時に 1 つずつデクリメントされます。

サウンドコントロールレジスタ(SGCRLn)

Figure 28-4. サウンドコントロールレジスタ(SGCRLn)の構成



サウンドコントロールレジスタ(SGCRLn)の内容

Table 28-2. サウンドコントロールレジスタ(SGCRLn)の各ビットの機能

名前		機能
bit7,bit 6	S1, S0: 動作クロック 選択ビット	サウンドジェネレータのためのクロック入力信号を指定します。 S1 / S0 / クロック入力 0 / 0 / 周辺クロックCLKP2 0 / 1 / 1/2周辺クロックCLKP2 1 / 0 / 1/4周辺クロックCLKP2 1 / 1 / 1/8周辺クロックCLKP2
bit 5	TONE: トーン出力 ビット	このビットを "1" に設定すると, SGO 信号はトグルフリップフロップからの単純方形波形(トーンパルス) になります。 "1" 以外を設定した場合は, トーンパルスと PWM パルスの混合 (論理積) 信号になります。
bit 4	OE2: サウンド出力 許可ビット	このビットを "1" に設定すると, 外部端子が SGO 出力として割り当てられます。 そうでない場合は, この端子は汎用 I/O端子となります。 SGO 出力を許可するには, ポート方向レジスタの対応するビットも "1" に設定する必要があります。
bit 3	OE1: 振幅出力許可ビット	このビットを "1" に設定すると , 外部端子が SGA 出力として割り当てられます。そうでない場合は , この端子は汎用 I/O となります。SGA 出力を許可するには , ポート方向レジスタの対応するビットも "1" に設定する必要があります。 SGA 信号は, サウンドの振幅を表す PWM パルス発生器からの PWM パルスです。
bit 2	INTE: 割込み許可 ビット	サウンドジェネレータの割込み信号を許可します。 このビットが "1" で, INT ビットが "1" に設定されると, サウンドジェネレータが割込み信号を送信します。
bit 1	INT: 割込みビット	トーンパルスカウンタが , トーンカウントレジスタおよびデクリメントグレードレジスタに指定されたトーンパルスの数をカウントすると , このビットは "1" に設定されます。 このビットは "0" を書き込むと "0" にリセットされます。 "1" を書き込んでも無効であり, リードモディファイライト命令は常に"1" を読み出します。
bit 0	ST: 開始ビット	このビットに "1" を書き込むと , サウンドジェネレータの動作が開始します。 "0" にリセットすると , サウンドジェネレータは現行のトーンサイクル終了時に動作を停止します。 BUSY ビットは , サウンドジェネレータが完全に停止したかどうかを示します。 このビットに"1"を書き込むと, 周波数データレジスタ, 振幅データレジスタ, デクリメントグレードレジスタ, トーンカウントレジスタの値が各カウンタへロードされます。

28.2.2 振幅データレジスタ(SGARn)

振幅データレジスタ(SGARn)は,PWMパルス発生器のリロード値を格納します。このレジスタ値はサウンドの振幅を表し, トーン信号の立下り時に PWM パルス発生器にリロードされます。

振幅データレジスタ(SGARn)

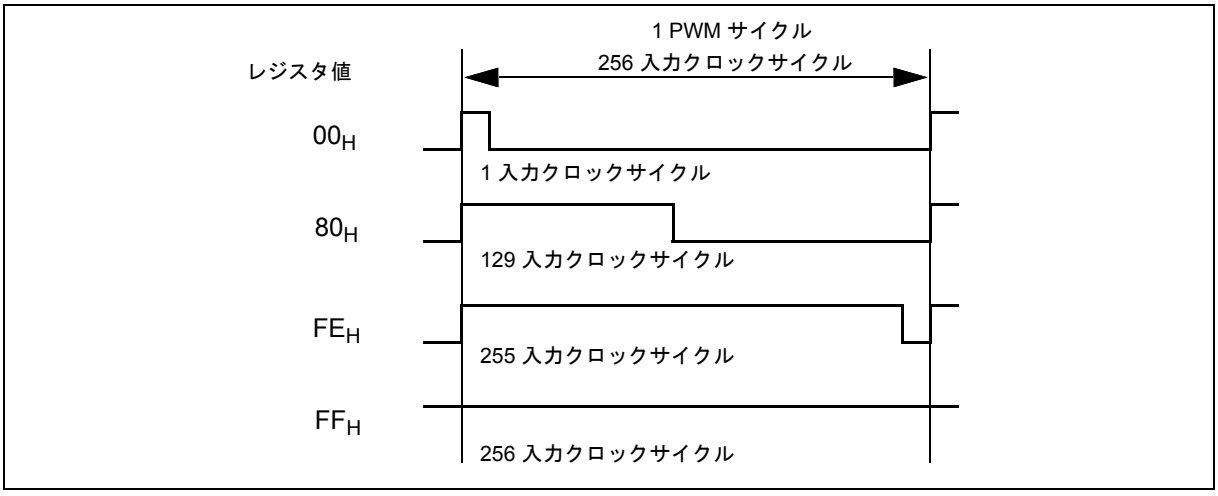
bit	7	6	5	4	3	2	1	0
	D7	D6	D5	D4	D3	D2	D1	D0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初期値	X	X	X	X	X	X	X	X

R/W : リード/ライト可能

DEC ビットが "1" でデクリメントカウンタがリロード値に達すると , このレジスタ値は 1 つずつデクリメントされます。レジスタ値が 00_H になると , デクリメントはもう行われません。しかし , サウンドジェネレータは ST ビットがクリアされるまで動作を継続します。

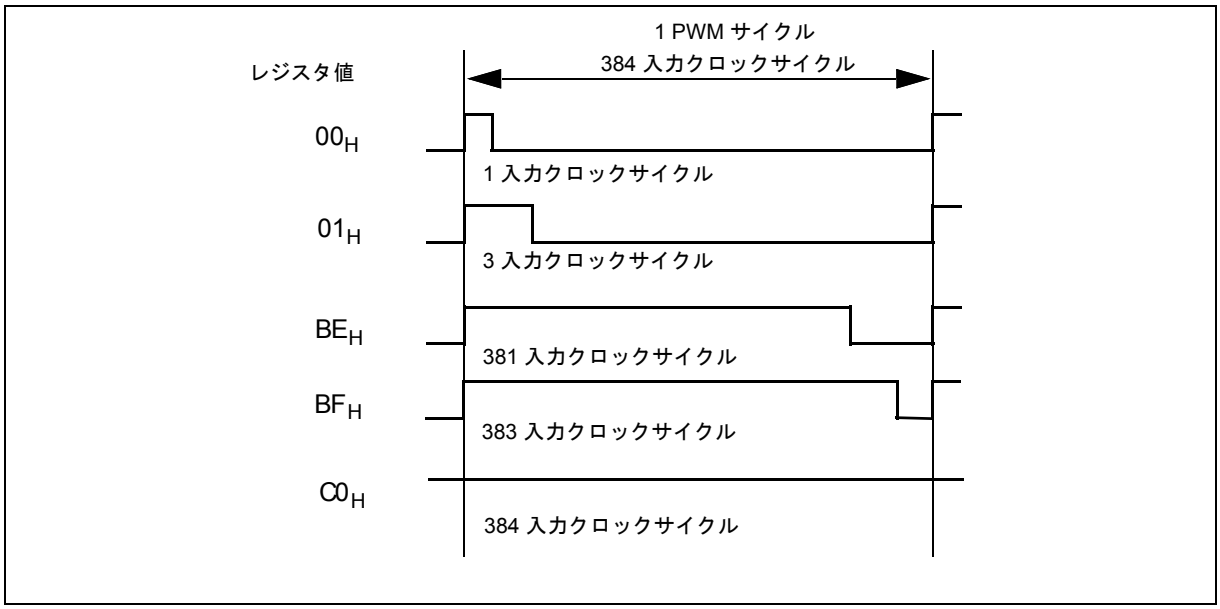
Figure 28-5 と Figure 28-6 に , レジスタ値と PWM パルスの関係を示します。

Figure 28-5. SGARn値とPWMパルスの関係(FSEL=1)



レジスタ値が FF_H に設定されると , PWM 信号は常に "1" 出力となります。

Figure 28-6. SGARn値とPWMパルスの関係(FSEL=0)



＜注意事項＞

SGCRHn の FSEL ビット が "0" で , SGARｎ 値が $C0_H$ ～ FF_H のとき , PWM 出力は常に "1" となります。

SGCRHnのDEC ビットが "1"で, SGARn 値が $C0_H$ 以上のとき, PWM出力は変動します。

28.2.3 周波数データレジスタ(SGFRn)

周波数データレジスタ (SGFRn) は, 周波数カウンタのリロード値を格納します。 格納された値は, サウンド (またはトグルフリップフロップからのトーン信号)の周波数を表します。 レジスタ値は,周波数カウンタとPWMパルス発生器のアンダフローでカウンタにリロードされます。
以下の図に, トーン信号とレジスタ値の関係を示します。

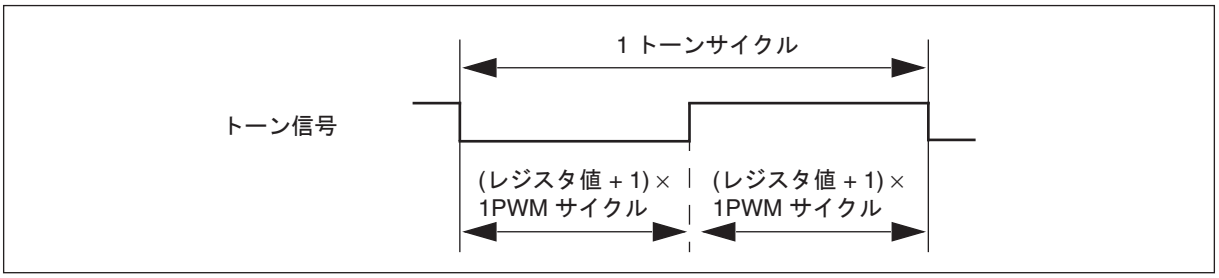
周波数データレジスタ(SGFRn)

bit	7	6	5	4	3	2	1	0	初期値
	D7	D6	D5	D4	D3	D2	D1	D0	XXXXXXXX$_B$
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : リード/ライト可能

Figure 28-7 に , トーン信号とレジスタ値の関係を示します。

Figure 28-7. トーン信号とSGFRn値の関係

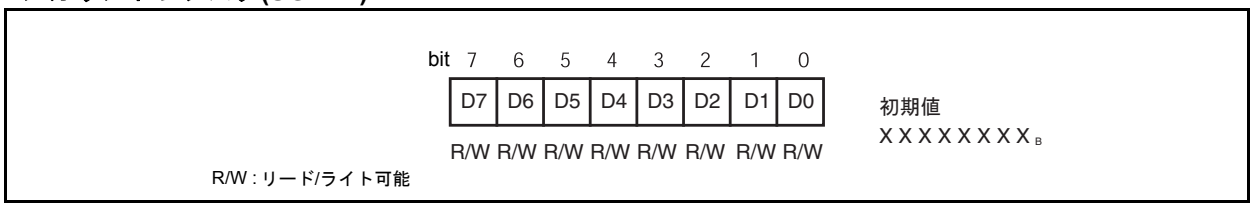


動作中にレジスタ値を変更すると , 変更のタイミングによっては , デューティサイクル 50% が変更されることがありますので , 注意してください。

28.2.4 トーンカウントレジスタ(SGTRn)

トーンカウントレジスタ (SGTRn) は, トーンパルスカウンタのリロード値を格納します。 トーンパルスカウンタにはトーンパルス数 (またはデクリメント動作数) が累積され, リロード値になると INT ビットをセットし, トーンの設定数により割込みの頻度を減少させます。 レジスタ値は, トーンパルスカウンタアンダフロー, 減分カウンタアンダフロー, トーン信号の立下りエッジでカウンタにリロードされます。

トーンカウントレジスタ(SGTRn)



トーンパルスカウンタのカウント入力は , デクリメントカウンタからのキャリアウト信号と接続されています。トーンカウントレジスタが 00_H に設定されると , トーンパルスカウンタは , デクリメントカウンタからのキャリアウト信号ごとに INT ビットを設定します。したがって , 累積トーンパルス数は次の式で表されます。

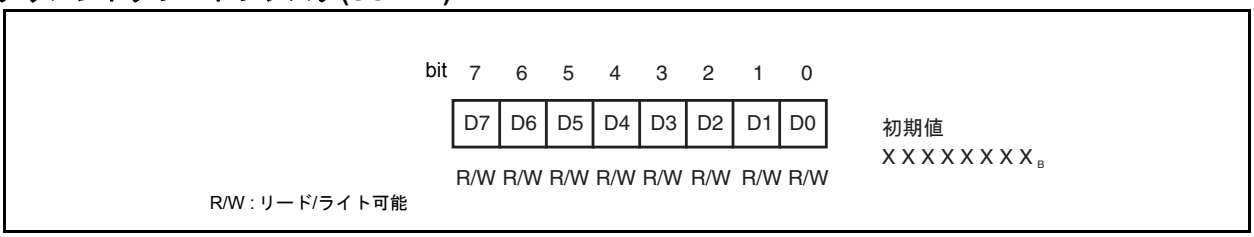
((デクリメントグレードレジスタ) + 1) × ((トーンカウントレジスタ) + 1)

すなわち , 両レジスタが 00_H に設定されると , トーンサイクルごとに INT ビットが設定されます。

28.2.5 デクリメントグレードレジスタ(SGDRn)

デクリメントグレードレジスタ (SGDRn) は,デクリメントカウンタのリロード値を格納します。 割込みの頻度を減少させるために, 振幅データレジスタのリロード値を自動的にデクリメントします。 レジスタ値は, デクリメントカウンタアンダフロー, トーン信号の立下りエッジでカウンタにリロードされます。

デクリメントグレードレジスタ(SGDRn)



DEC ビットが "1" でデクリメントカウンタがトーンパルス数をリロード値までカウントすると , 振幅データレジスタに格納された値は , トーンサイクル終了時に 1 つずつデクリメントされます。

この動作によって , より少ない CPU 介入でサウンドの自動デクリメントができるようになります。

このレジスタが指定したトーンパルス数は , " レジスタ値 + 1" に等しいことに注意してください。デクリメントグレードレジスタが 00_H に設定されると , デクリメント動作はトーンサイクルごとに行われます。

29. USBファンクション

USBファンクションの機能と動作について説明します。

29.1 USBファンクションの概要

USBファンクションは，USB(Universal Serial Bus) 1.1通信プロトコルをサポートするインタフェースです。転送スピードはFULL(12 Mbps)に対応して動作します。

USBファンクションの特長

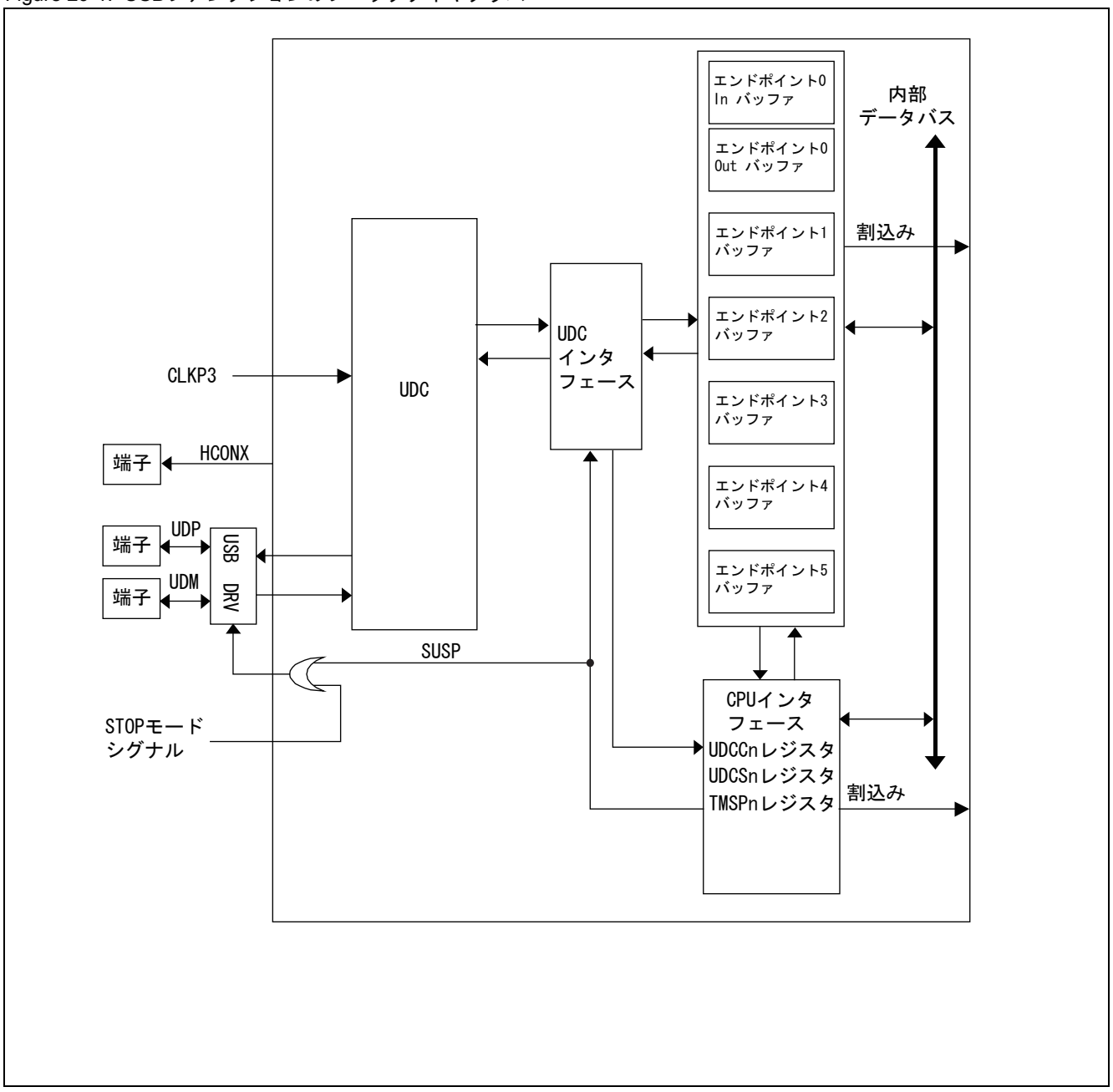
- FULLスピード(12 Mbps)をサポート (USB Full Speed準拠)
- デバイスステータスコマンドは自動応答
- Bit Stripping, Bit Stuffing, CRC5, CRC16の自動生成とチェック
- データ同期ビットによるトグルチェック
- Get/SetDescriptor, SynchFrameコマンドを除くすべての標準コマンドに自動応答
  前記3コマンドはクラス・ベンダーコマンドと同様の処理が可能
- クラス・ベンダーコマンドはデータとして受信し, ファームウェアによる応答が可能
- 最大6本のエンドポイントをサポート(エンドポイント0はcontrol転送に固定)
- 転送データバッファに各エンドポイントでバッファを2本ずつ内蔵
  エンドポイント0の場合はINとOUTそれぞれ専用に1本ずつ内蔵
- DMAによる転送データの自動転送モードをサポート(エンドポイント0のバッファ以外)

29.2 USBファンクションのブロックダイヤグラム

USBファンクションについて説明します。

USBファンクションのブロックダイヤグラム

Figure 29-1. USBファンクションのブロックダイヤグラム



29.3 USBファンクションのレジスタ

USBファンクションのレジスタについて説明します。

USBファンクションのレジスター覧

Table 29-1. レジスター覧

15	8	7	0	属性
UDCCn				(R/W)
EP0Cn				(R/W)
EP1Cn				(R/W)
EP2Cn				(R/W)
EP3Cn				(R/W)
EP4Cn				(R/W)
EP5Cn				(R/W)
TMSPn				(R)
UDCIEn		UDCSn		(R/W)
EP0ISn				(R/W)
EP0OSn				(R/W)
EP1Sn				(R/W)
EP2Sn				(R/W)
EP3Sn				(R/W)
EP4Sn				(R/W)
EP5Sn				(R/W)
EP0DTn				(R/W)
EP1DTn				(R/W)
EP2DTn				(R/W)
EP3DTn				(R/W)
EP4DTn				(R/W)
EP5DTn				(R/W)
8ビット		8ビット		

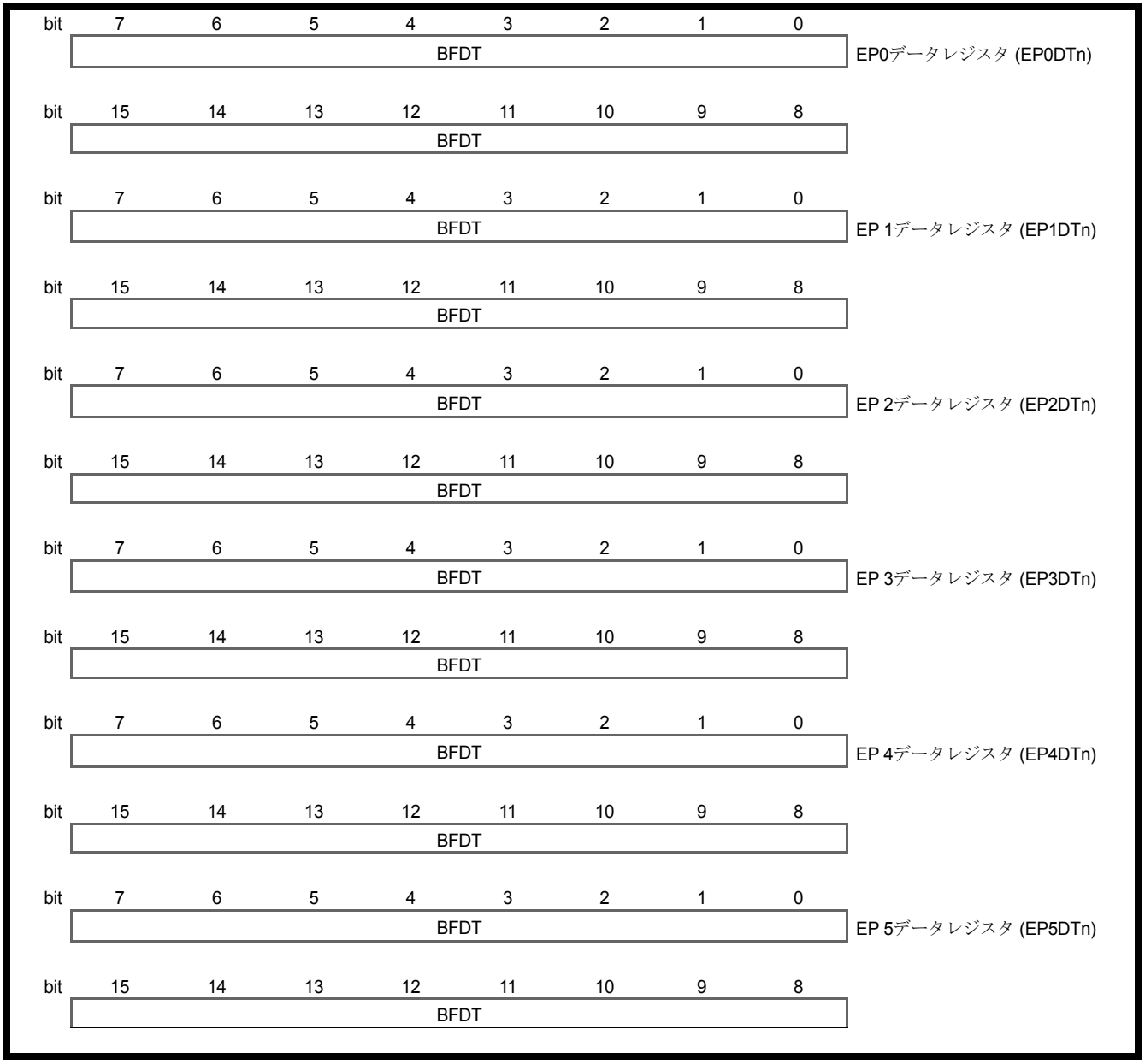
Table 29-2. USBファンクションのレジスタ

bit	7	6	5	4	3	2	1	0	
	RST	RESUM	HCONX	USTP	予約	予約	RFBK	PWC	UDC制御レジスタ (UDCCn)
bit	15	14	13	12	11	10	9	8	
	予約	予約	予約	予約	予約	予約	予約	予約	
bit	7	6	5	4	3	2	1	0	
	予約	PKS0							EP0 制御レジスタ (EP0Cn)
bit	15	14	13	12	11	10	9	8	
	-	-	-	-	予約	予約	STAL	予約	
bit	7	6	5	4	3	2	1	0	
	PKS 1								EP1 制御レジスタ (EP1Cn)
bit	15	14	13	12	11	10	9	8	
	EPEN	TYPE		DIR	DMAE	NULE	STAL	PKS1	
bit	7	6	5	4	3	2	1	0	
	予約	PKS 2							EP2 制御レジスタ(EP2Cn)
bit	15	14	13	12	11	10	9	8	
	EPEN	TYPE		DIR	DMAE	NULE	STAL	予約	
bit	7	6	5	4	3	2	1	0	
	予約	PKS 3							EP3 制御レジスタ (EP3Cn)
bit	15	14	13	12	11	10	9	8	
	EPEN	TYPE		DIR	DMAE	NULE	STAL	予約	
bit	7	6	5	4	3	2	1	0	
	予約	PKS 4							EP4 制御レジスタ (EP4Cn)
bit	15	14	13	12	11	10	9	8	
	EPEN	TYPE		DIR	DMAE	NULE	STAL	予約	
bit	7	6	5	4	3	2	1	0	
	予約	PKS 5							EP5 制御レジスタ (EP5Cn)
bit	15	14	13	12	11	10	9	8	
	EPEN	TYPE		DIR	DMAE	NULE	STAL	予約	
bit	7	6	5	4	3	2	1	0	
	TMSP								タイムスタンプレジスタ (TMSPn)
bit	15	14	13	12	11	10	9	8	
	-	-	-	-	-	TMSP			

Table 29-2. USBファンクションのレジスタ

bit	7	6	5	4	3	2	1	0	
	-	-	SUSP	SOF	BRST	WKUP	SETP	CONF	UDCステータスレジスタ(UDCSn)
bit	15	14	13	12	11	10	9	8	
	予約	予約	SUSPIE	SOFIE	BRSTIE	WKUPIE	CONFN	CONFIE	UDC割込み許可レジスタ(UDCIEn)
bit	7	6	5	4	3	2	1	0	
	-	-	-	-	-	-	-	-	EP0I ステータスレジスタ(EP0ISn)
bit	15	14	13	12	11	10	9	8	
	BFINI	DRQIIE	-	-	-	DRQI	-	-	
bit	7	6	5	4	3	2	1	0	
	予約	SIZE							EP0Oステータスレジスタ(EP0OSn)
bit	15	14	13	12	11	10	9	8	
	BFINI	DRQOIE	SPKIE	-	-	DRQO	SPK	予約	
bit	7	6	5	4	3	2	1	0	
	SIZE								EP1ステータスレジスタ(EP1Sn)
bit	15	14	13	12	11	10	9	8	
	BFINI	DRQIE	SPKIE	予約	BUSY	DRQ	SPK	SIZE	
bit	7	6	5	4	3	2	1	0	
	予約	SIZE							EP2ステータスレジスタ(EP2Sn)
bit	15	14	13	12	11	10	9	8	
	BFINI	DRQIE	SPKIE	予約	BUSY	DRQ	SPK	予約	
bit	7	6	5	4	3	2	1	0	
	予約	SIZE							EP3ステータスレジスタ(EP3Sn)
bit	15	14	13	12	11	10	9	8	
	BFINI	DRQIE	SPKIE	予約	BUSY	DRQ	SPK	予約	
bit	7	6	5	4	3	2	1	0	
	予約	SIZE							EP4ステータスレジスタ(EP4Sn)
bit	15	14	13	12	11	10	9	8	
	BFINI	DRQIE	SPKIE	予約	BUSY	DRQ	SPK	予約	
bit	7	6	5	4	3	2	1	0	
	予約	SIZE							EP5ステータスレジスタ(EP5Sn)
bit	15	14	13	12	11	10	9	8	
	BFINI	DRQIE	SPKIE	予約	BUSY	DRQ	SPK	予約	

Table 29-2. USBファンクションのレジスタ

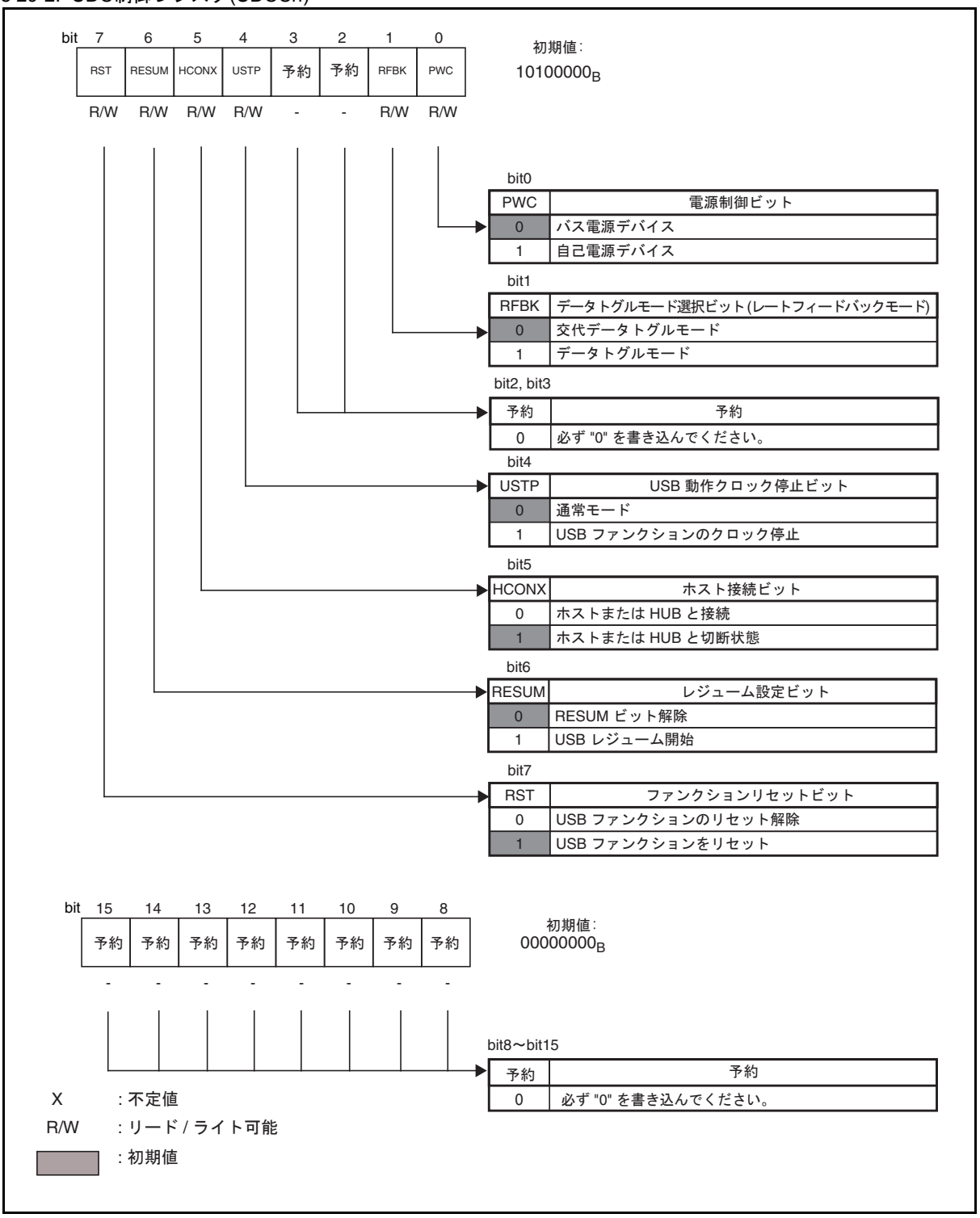


29.3.1 UDC制御レジスタ(UDCCn)

UDC制御レジスタ(UDCCn)は, UDC (USBデバイス制御コア)の制御を行います。

UDC制御レジスタ(UDCCn)

Figure 29-2. UDC制御レジスタ(UDCCn)



以下に, UDC 制御レジスタ (UDCC) の各ビットの機能を説明します。

	ビット名	機能
Bit7	RST	• USB ファンクション全体をリセットします。 • ホストにケーブルを接続する際 , RST ビットを "1" に設定して USB ファンクションを初期化します。 • このビットの初期値は "1" (USB ファンクションをリセット) であるため , USB ファンクションをリセットステータスから解除するには , このビットに "0" を書き込んでください。 • RST ビットはタイムスタンプレジスタ, UDC ステータスレジスタ, および割込み許可レジスタの対応ビットを一度に初期化します。さらに, RST ビットは EP0I, EP0O, および EP1 ～ EP5 ステータスレジスタの BFINI ビットを同時に初期化するため, 初期化の後まず RST ビット (BFINI ビットをクリアしない) をクリアしてから, 使用されたエンドポイントの BFINI ビットをクリアします。
Bit6	RESUM	• USB ファンクションがリモートウェイクアップ許可ステータス (あるいは DEVICE_REMOTE_WAKEUP機能がホストからの SET_FEATUREコマンドによって設定されている) であり, サスペンド状態にある場合, RESUM ビットに 1 を書き込むことによってレジューム動作が開始されます。USB ファンクションをレジュームするには, RESUM ビットを "1" に設定してから RESUM ビットを "0" にクリアしてください。
Bit5	HCONX	• USB ファンクションが接続されていることをホストまたは HUB に認識させるために , 外部プルアップ抵抗と USB データ線の切換えを制御します。 • 外部プルアップ抵抗が ON になっていて, USB ファンクションが接続されていることをホストまたは HUB が認識する場合でも, USB ファンクションモジュールは HCONX ビットが "1" に設定された際にバスリセットおよび USB バス上のコマンドを無視します。
Bit4	USTP	• USB ファンクションへのクロックを停止します。 • USTP ビットを "1" に設定すると USB モジュールへのクロックが停止され , 消費電力が削減されます。 • USTP ビットを "1" に設定した後, STOP モードを選択しない場合, まず RST ビットを "1" に設定し, USB モジュールがリセットされたことを確認するために 3 サイクル待機してから, USTP ビットを "1" を設定します。USTP ビットと RST ビットを同時にクリアしても問題ありません。
Bit3	予約	• 必ず "0" を書き込んでください。常に "0" が読み出されます。
Bit2	予約	• 必ず "0" を書き込んでください。常に "0" が読み出されます。
Bit1	RFBK	• このビットは , データトグルモードを選択します。 • "0" に設定した場合 , 転送が正常に完了したときのみ DATA0 と DATA1 間のデータ PID をトグルします。 • "1" に設定した場合 , DATA0 と DATA1 間のデータ PID を無条件でトグルします。
Bit0	PWC	• このビットは, USB ファンクションの電源モード (自己電源またはバス電源) を指定します (PWC ビットの設定は標準的な USB コマンド: GetStatus に反映されます)。

＜注意事項＞

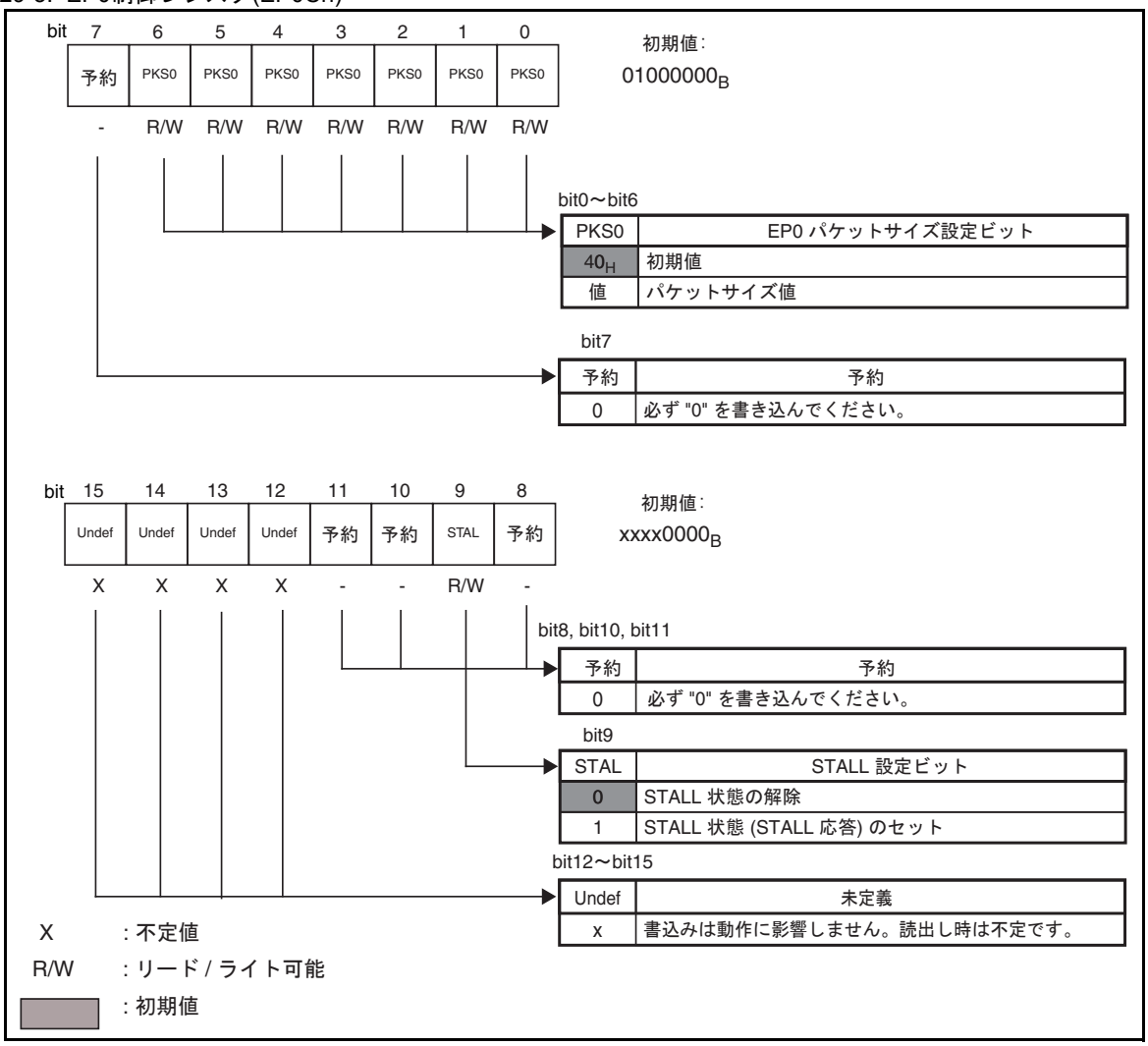
RESUM および USTP ビットを除いて RST =1 のとき , UDC 制御レジスタ (UDCCn) のビットを変更しないでください。USB が動作中にこれらのビットを変更しないください。USB がサスペンドモードで , リモートウェイクアップが許可されている場合にのみ RESUM ビットを設定し , リセットしてください (DEVICE_REMOTE_WAKEUP ビットは ホストからの SET_FEATURE コマンドによって設定されます)。STOP モード時は , STOP モードに入る前に USTP ビットに "1" を設定してください。STOP モード解除時は , SUSP ビットを "0" にクリアし , USTP ビットを "0" にクリアしてください。

29.3.2 EP0制御レジスタ(EP0Cn)

EP0制御レジスタ(EP0Cn)は, エンドポイント0の動作に関する制御を行います。

EP0制御レジスタ(EP0Cn)

Figure 29-3. EP0制御レジスタ(EP0Cn)



＜注意事項＞

EP0 制御レジスタ (EP0Cn) は , bit9 の STAL を除き UDC 制御レジスタ (UDCCn) の RST, EP0I/EP0O ステータスレジスタ (EP0ISn / EP0OSn) の BFINI ビットが共に "1" のときに必ず設定するようにしてください。USB の動作中に EP0Cn を変更しないでください。

ビット名		機能
Bit15 ～ bit12	未定義	• 書込みは動作に影響しません。読出し時は不定です。
Bit11, bit10	予約	• 必ず "0" を書き込んでください。常に "0" が読み出されます。
Bit9	STAL	• STAL ビットへの "1" 設定によりエンドポイント 0 を STALL 状態 (STALL 応答) にできます。 • STALビットへの"0"設定後, 正常なパケットを受信したときにSTALL状態から復帰します。
Bit8, bit7	予約	• 必ず "0" を書き込んでください。常に "0" が読み出されます。
Bit6 ～ bit0	PKS0	• 1 パケットでの最大転送バイト数を指定します。エンドポイント 0 で指定可能なパケットの最大転送バイト数は 64 バイトで , IN, OUT 共通の設定となります。< 例 > "08_H" ⇒ 8 バイト , "40_H" ⇒ 64 バイト (最大指定値) • 値は, 必ず "00_H" より大きく, 最大転送バイト数 (40_H) 以下にする必要があります。

29.3.3 EP1～EP5制御レジスタ(EP1Cn to EP5Cn)

EP1～EP5制御レジスタ(EP1Cn～EP5Cn)は, エンドポイント1～5に関する制御を行います。

EP1～EP5制御レジスタ(EP1Cn to EP5Cn)

Figure 29-4. EP1制御レジスタ(EP1Cn)

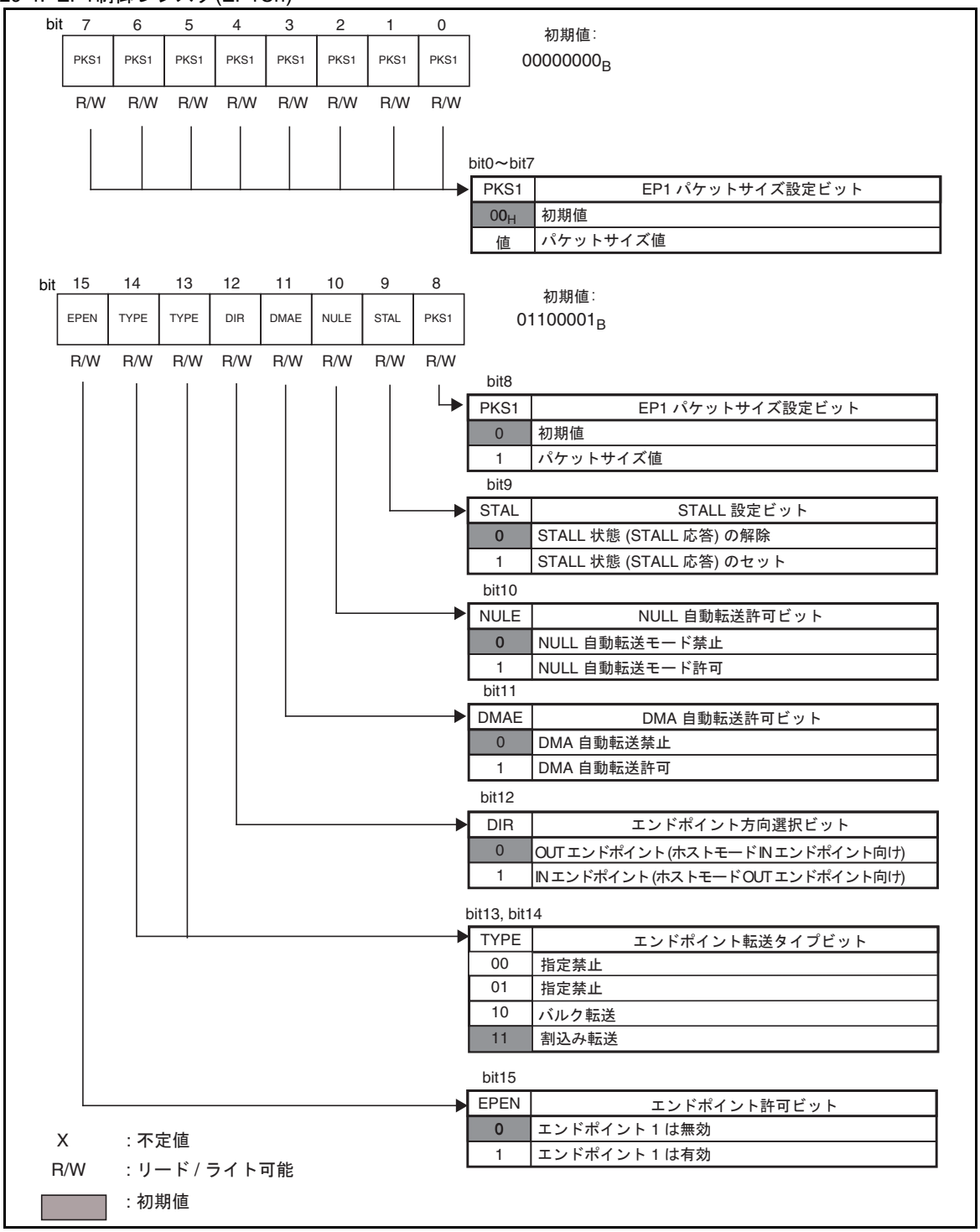
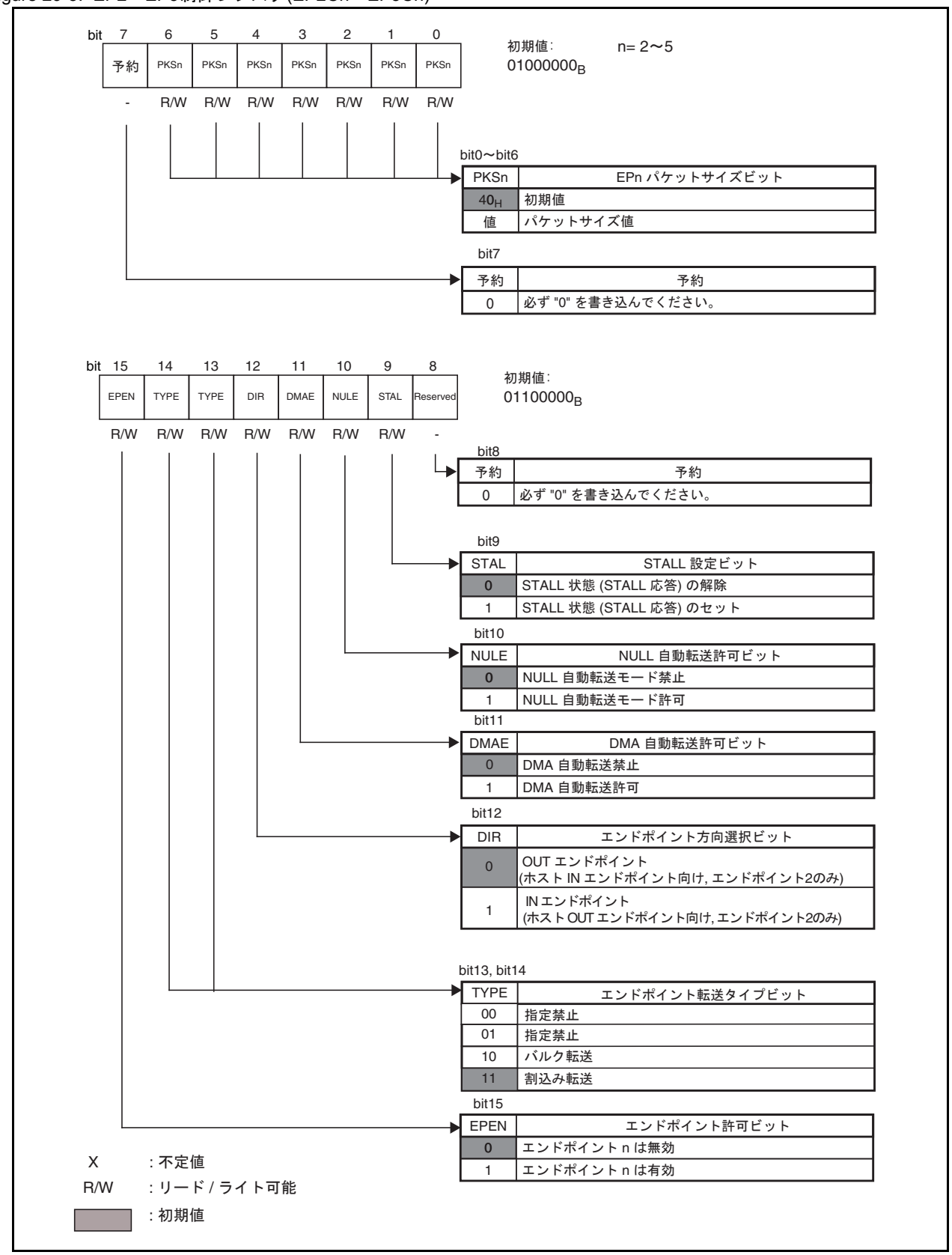


Figure 29-5. EP2～EP5制御レジスタ(EP2Cn～EP5Cn)



＜注意事項＞

EP1 ～ EP5 制御レジスタ (EP1Cn ～ EP5Cn) は , DMAE, NULE, STAL ビットを除き UDCCn:RST ビットと EPxSn:BFINI が共に "1" のときに設定して , USB 動作中に書き替えないようにしてください。

以下に, EP1～EP5制御レジスタ(EP1Cn～EP5Cn)の各ビットの機能を説明します。

ビット名		機能
Bit15	EPEN	• エンドポイントを有効にします。EPEN ビットの設定により USB ファンクションで使用するエンドポイントとしてホストからエンドポイントを構成できます。このビットを "1" に設定すると, EP1 ～ EP5 制御レジスタの TYPE, DIR, PKS が構成情報として有効となります。
Bit14, Bit13	TYPE	• 対応するエンドポイントの転送タイプ (バルク転送または割込み転送) を指定します。
Bit12	DIR	• エンドポイントの転送方向を指定します。 • USB ファンクションでは, このビットに "1" を書き込むと, 対応するエンドポイントが IN エンドポイントとして構成されます。"0" を書き込むと , 対応するエンドポイントが OUT エンドポイントとして構成されます。 • USB ホストモードでは, エンドポイント 1 と 2 だけが有効です。ホストモードでこのビットに "0" を書き込むと, 対応するエンドポイントが IN エンドポイントとして構成され, "1" を書き込むと, 対応するエンドポイントが OUT エンドポイントとして構成されます。
Bit11	DMAE	• DMA 自動転送モード許可ビットです。 • 有効にした場合 , DMA は DMA の DCT レジスタに設定したデータ転送数まで , ホストからの IN, OUT データ要求に同期してデータの送受信を自動で転送します。詳細については「29.4.6 DMA転送機能」を参照してください。 • DMAEビットのセット中は,CPUによる送受信バッファへのアクセスは禁止です。エンドポイントがOUT転送に使用されるときには,DMA の転送データ数の設定は EP1 ～ EP5 制御レジスタ (EP1Cn ～ EP5Cn) の PKS ビットに設定した値の倍数としてください。
Bit10	NULE	• NULL 転送を許可します。詳細については「29.4.7 NULL 転送機能」を参照してください。 • IN 方向の転送時には , 自動バッファ転送モードがセットされている状態 (DMAE=1) で , IN 方向のデータ転送要求がホストからきたときに , 最後のパケット転送を検出し , 0 バイトのデータパケットを自動で送信します。 • OUT方向のデータ転送時や自動バッファ転送モードが禁止されているときには NULE ビットは通信に影響しません。

<table>
<tr><th colspan="2">ビット名</th><th>機能</th></tr>
<tr><td>Bit9</td><td>STAL</td><td>• エンドポイントを STALL 状態 (STALL 応答) に設定します。
• STALビットを設定すると,USBファンクションはSTALLハンドシェークパケットによりホストからの要求に応答します。USBファンクションはSTALビットに "0" を書き込むことによるSTAL ビットの解除の後, ホストからのClearFeature コマンドにより STALL 状態から復帰できます。</td></tr>
<tr><td>Bit8</td><td>予約</td><td>• このビットは , EP2 ～ EP5 の場合 , 予約ビットです。
• 必ず "0" を書き込んでください。</td></tr>
<tr><td>Bit7</td><td>予約</td><td>• このビットは , EP2 ～ EP5 の場合 , 予約ビットです。
• 必ず "0" を書き込んでください。</td></tr>
<tr><td>Bit0～Bit8:
EPICn

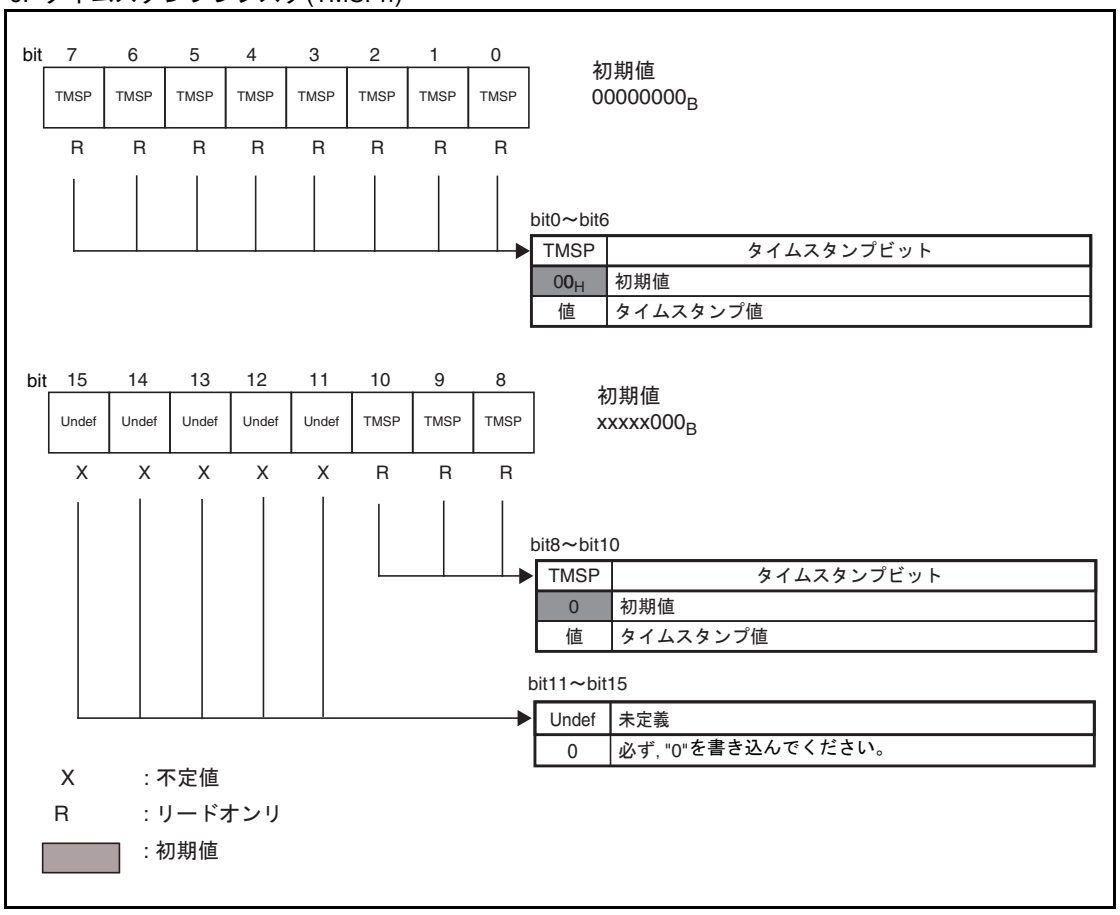
Bit0～Bit6:
EP2Cn～
EP5Cn</td><td>PKS</td><td>• 1 パケットでの転送可能な最大バイト数を指定します。エンドポイントごとに指定可能な 1 パケットでの最大転送バイト数を下表に示します。
<table>
<tr><th>ｴﾝﾄﾞﾎﾟｲﾝﾄ</th><th>最大転送バイト数</th><th>指定可能な値</th></tr>
<tr><td>1</td><td>256 bytes</td><td>001h～100h</td></tr>
<tr><td>2～5</td><td>64 bytes</td><td>01h～40h</td></tr>
</table>
• 最大転送バイト数 (100_Hまたは40_H) 以上の設定と"00_H"の設定は禁止です。
• エンドポイント 2 ～ 5 に対してはbit7とbit8で "00" 書込みをしてください。
• データ数自動転送モードを選択 (DMAE=1) すると, 対応するエンドポイントでの0～2設定は禁止となります。</td></tr>
</table>

29.3.4 タイムスタンプレジスタ(TMSPn)

タイムスタンプレジスタ(TMSPn)は, SOFパケット受信時のフレーム番号の表示を行います。

タイムスタンプレジスタ(TMSPn)

Figure 29-6. タイムスタンプレジスタ(TMSPn)



以下に, タイムスタンプレジスタ(TMSPn)の各ビットの機能を説明します。

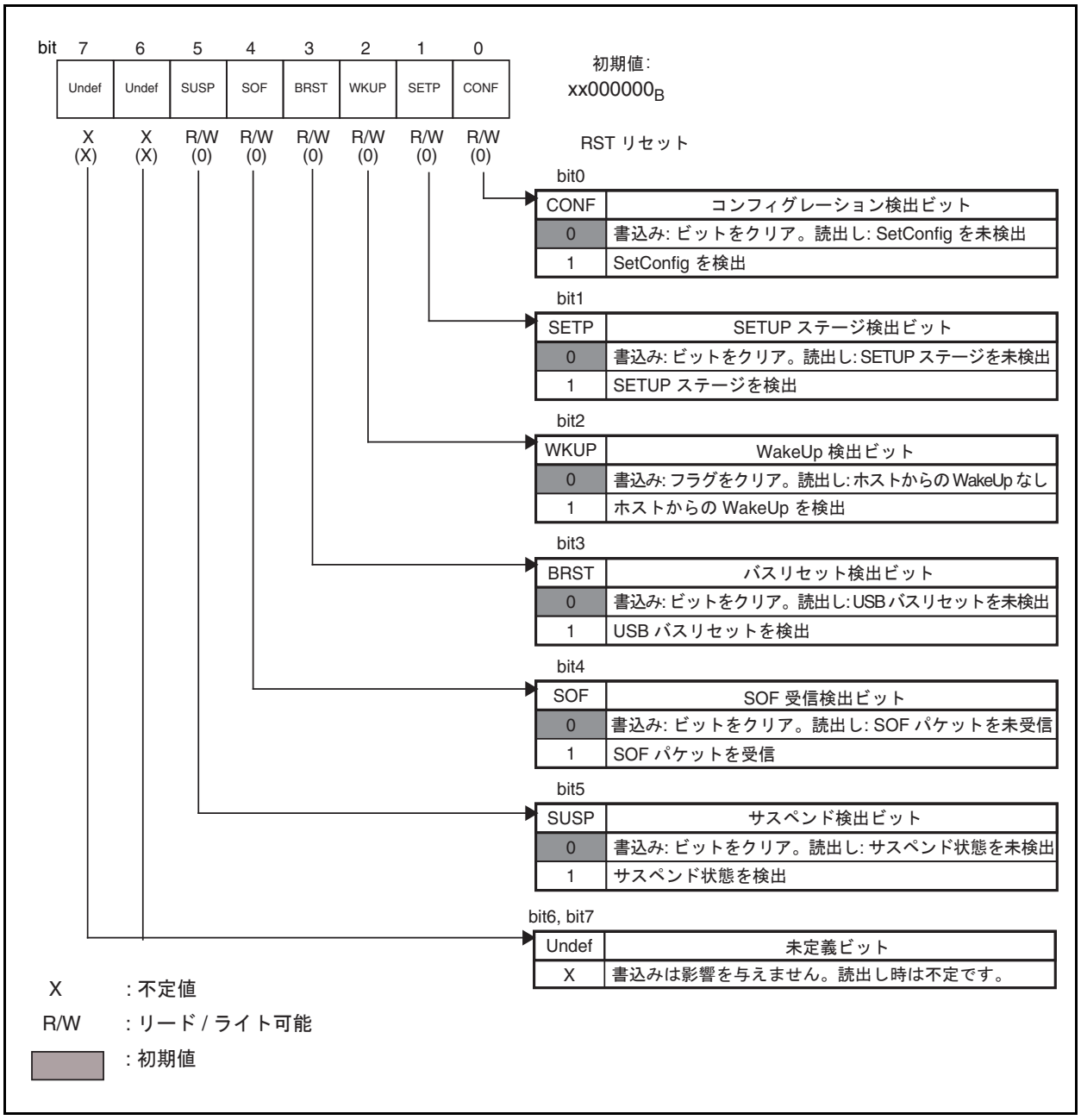
ビット名		機能
Bit15～Bit11	未定義	• 書込みは動作に影響しません。読出し時は不定です。
Bit10～Bit0	TMSP	• 現在のSOFパケットのフレーム番号を格納します。 SOFパケットの受信時にこのビットが使用されます。

29.3.5 UDCステータスレジスタ(UDCSn)

UDCステータスレジスタ(UDCSn)は,USB通信上の状態を示すレジスタです。SETPを除く各ビットは割込み要因となっており， UDC割込み許可レジスタ(UDCIEn)の対応する割込み許可ビットが設定されていればCPUへ割込みが発生します。

UDCステータスレジスタ(UDCSn)

Figure 29-7. UDCステータスレジスタ



UDCステータスレジスタ(UDCSn)の各ビットの機能を下記に示します。

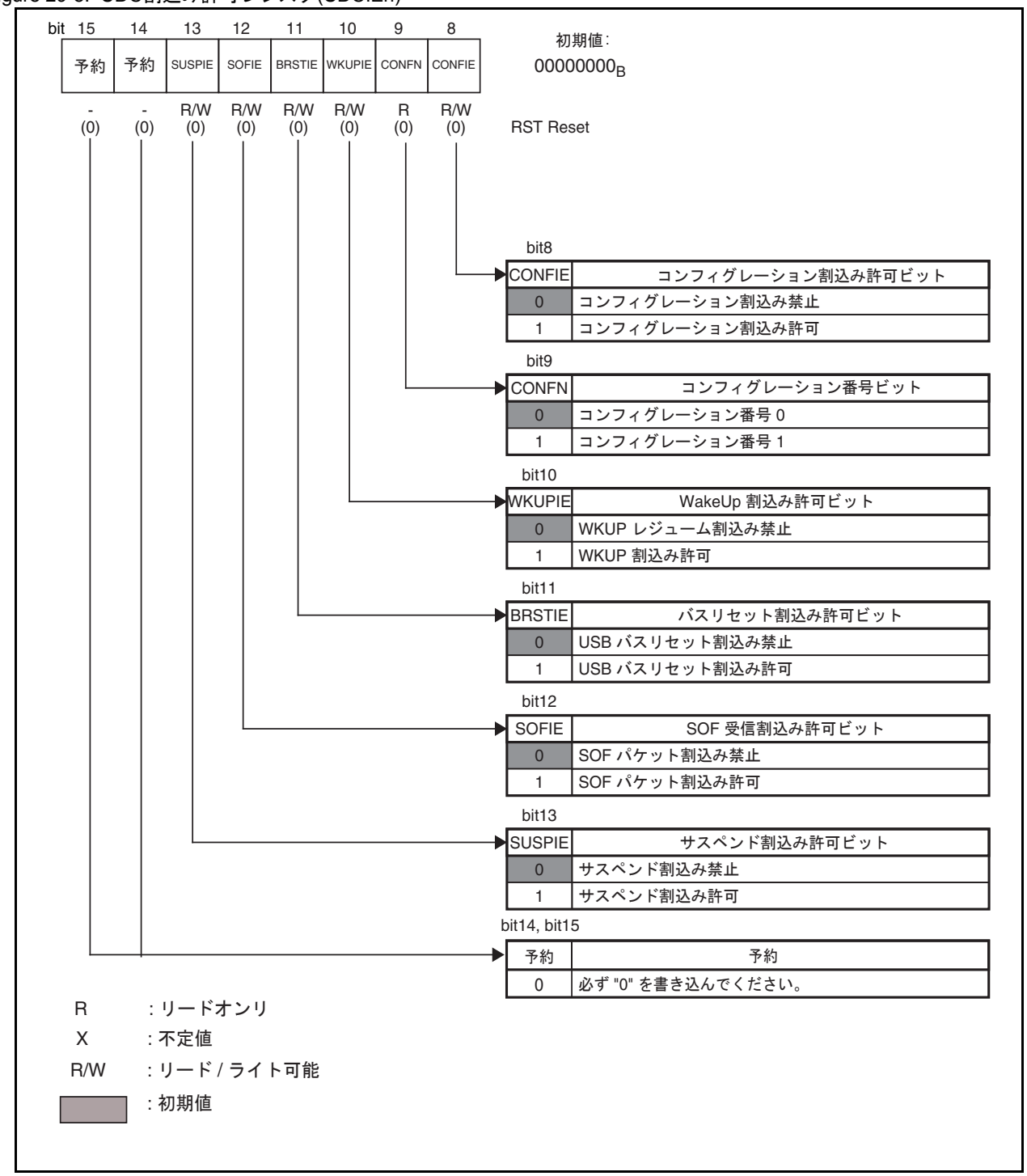
ビット名		機能
Bit7, Bit6	未定義	• 書込みは動作に影響しません。読出し時は不定です。
Bit5	SUSP	• このフラグは , USB ファンクションがサスペンド状態にあることを表示します。SUSP ビットは USB_F10 割込み要因です。 • "1" 書込みは無視します。 • "0" 書込みでクリアしてください。リードモディファイライト時は "1" が読み出されます。
Bit4	SOF	• SOF パケットを受信したことを示すフラグです。 • SOF を "1" に設定すると , タイムスタンプレジスタ (TMSPn) の値が更新されます。 • SOF フラグは USB_F10 割込み要因です。 • "1" 書込みは無視します。 • "0" 書込みでクリアしてください。リードモディファイライト時は "1" が読み出されます。
Bit3	BRST	• このフラグは , USB バスのリセットを表示します。USB_F10 割込み要因です。 • バスリセットを検出 (BRST = "1") したときに UDCCn:RST によって USB ファンクションを初期化した後, すべての USB レジスタを設定してください。 • "1" 書込みは無視します。 • "0" 書込みでクリアしてください。リードモディファイライト時は "1" が読み出されます。
Bit2	WKUP	• このフラグは , USB ファンクションがサスペンド状態から復帰したことを表示します。USB_F10 割込み要因です。 • リモートウェイクアップ (RESUM ビットのセットによる) またはホストからのウェイクアップ要求により , USB ファンクションがサスペンド状態から復帰します。WKUP ビットはホストからの復帰要求のみで自動セットされます。 • ホスト要求によるウェイクアップ時でも UDCCn レジスタの RESUM ビットが "1" にセットされている場合 , WKUP フラグはセットされません。 • "1" 書込みは無視します。 • "0" 書込みでクリアしてください。リードモディファイライト時は "1" が読み出されます。
Bit1	SETP	• このビットは , 受信したデータが USB コントロール転送の Setup ステージであることを示します。 • このビットは , USB ファンクションが標準コマンドへ自動応答するときにはセットされません。 • SETP ビットは割込み要因ではありません。 • "1" 書込みは動作に影響しません。 • "0" 書込みでクリアしてください。リードモディファイライト時は "1" が読み出されます。
Bit0	CONF	• USB コマンドの SetConfig が受信され , USB ファンクションが構成されるとセットされます。 • CONF ビットは USB_F10 割込み要因です。 • "1" 書込みは無視します。 • "0" 書込みでクリアしてください。リードモディファイライト時は "1" が読み出されます。

29.3.6 UDC割込み許可レジスタ(UDCIEn)

UDC割込み許可レジスタ(UDCIEn)は,CONFNを除きUDCステータスレジスタ(UDCSn)の各フラグによる割込みをビットごとに許可するレジスタです。

UDC割込み許可レジスタ(UDCIEn)

Figure 29-8. UDC割込み許可レジスタ(UDCIEn)



以下に, UDC割込みレジスタ(UDCIE)の各ビットの機能を説明します。

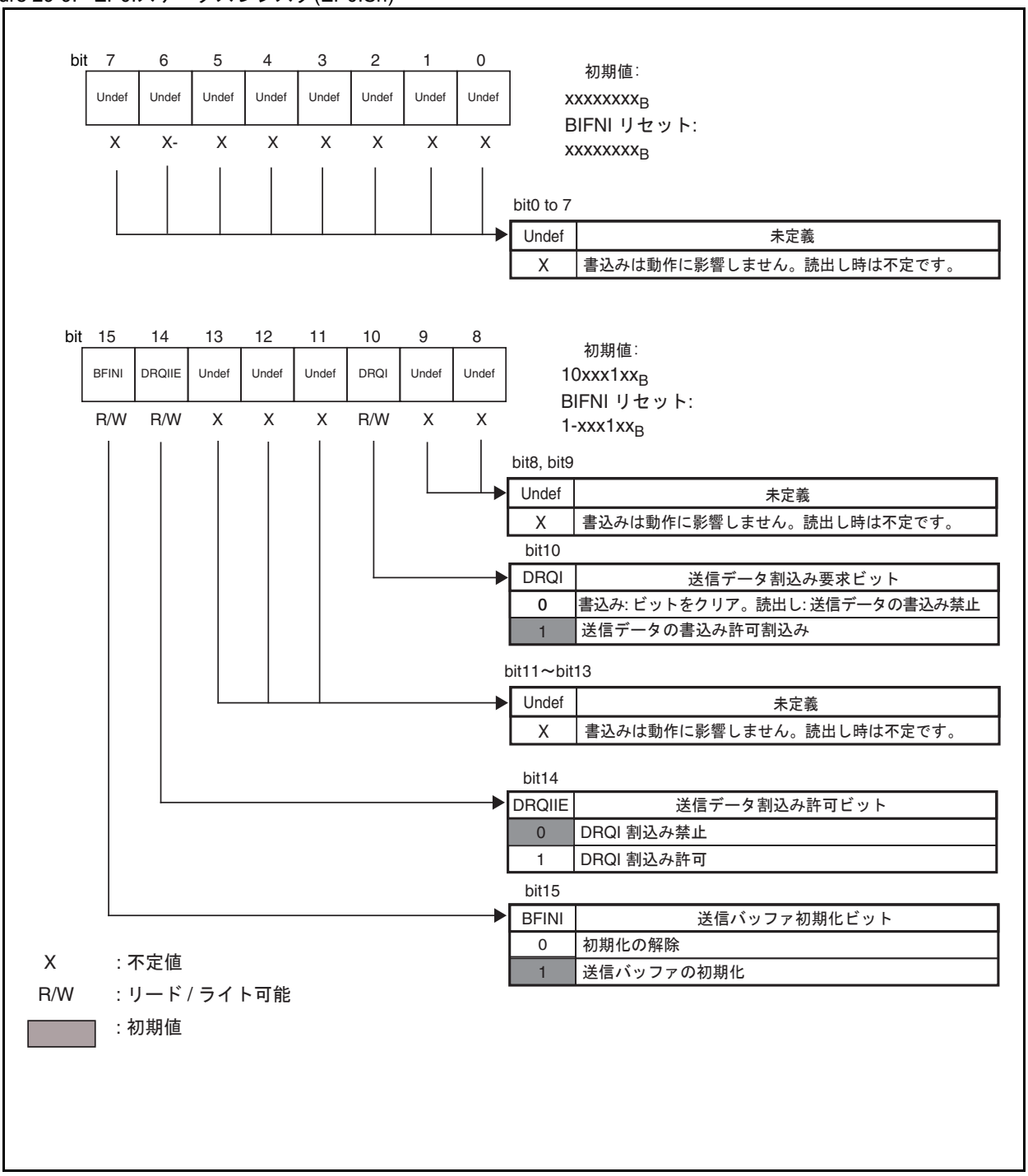
ビット名		機能
Bit15, Bit14	予約	• 必ず "0" を書き込んでください。 このビットの読出し値は常に "0" です。
Bit13	SUSPIE	• SUSP フラグ要因による割込みを許可します。
Bit12	SOFIE	• SOF フラグ要因による割込みを許可します。
Bit11	BRSTIE	• BRST フラグ要因による割込みを許可します。
Bit10	WKUPIE	• WKUP フラグ要因による割込みを許可します。
Bit9	CONFN	• このビットはコンフィグレーション番号を表示します。 • ホストからの SET_CONFIGURATIONコマンドによって UDCSnの CONFフラグがセットされると更新されます。
Bit8	CONFIE	• UDCSn の CONF フラグ要因による割込みを許可します。

29.3.7 EP0Iステータスレジスタ(EP0ISn)

EP0Iステータスレジスタ(EP0ISn)は, エンドポイント0のIn方向転送に関するステータス表示を行います。

EP0Iステータスレジスタ(EP0ISn)

Figure 29-9. EP0Iステータスレジスタ(EP0ISn)



以下に, EP0Iステータスレジスタ(EP0ISn)の各ビットの機能を説明します。

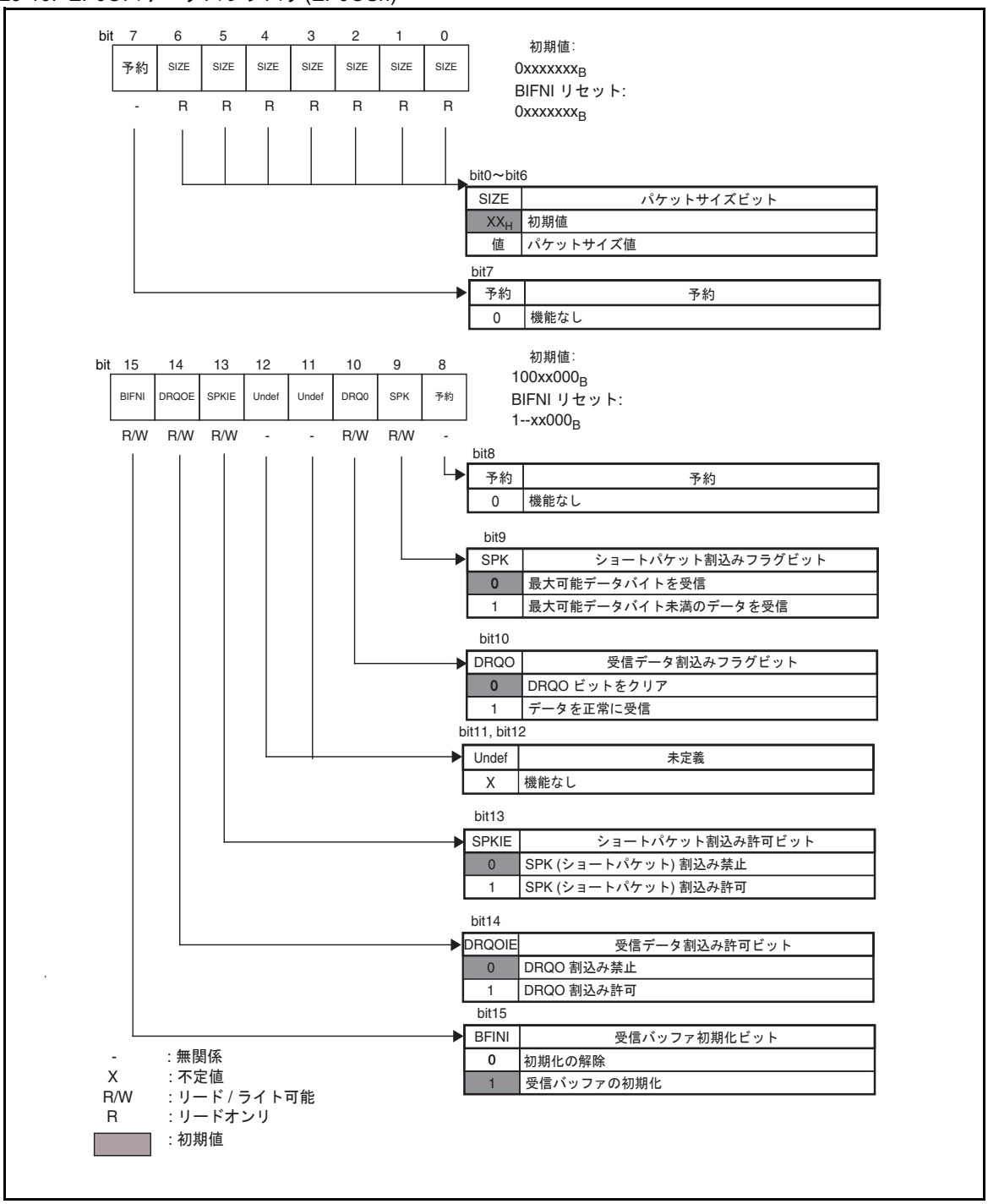
ビット名		機能
Bit15	BFNI	• このビットは , 送信バッファを初期化します。 • BFINI ビットは UDC 制御レジスタ (UDCCn) の RST フラグが "0" にセットされると , 自動的にセットされます。 • UDCCn:RST ビットでリセット動作が実行された場合, BFINI ビットをクリアする前に UDCCn:RST ビットをクリアしてください。 • BFINIビットはバッファおよびDRQIビットを初期化します。バッファを初期化する場合, DRQI または DRQO ビットが "1" にセットされている (ホストからのアクセスがない) ことを確認した後, 必要に応じて STAL ビットをセットしてから初期化を行ってください。
Bit14	DRQIIE	• DRQI 割込みを許可します。 • DRQI フラグをセットするとDRQI割込み要因による割込みを許可します。
Bit13～Bit11	Undefined	• 書込みは動作に影響しません。読出し時は不定です。
Bit10	DRQI	• このビットは, IN パケットが EP0 からホストへ正常に転送されたことを示します。送信バッファからデータが読み出され, 次の送信データをバッファに書き込むことが可能です。 • データを送信バッファに書き込んだ後 , DRQI をクリアする必要があります。 • DRQI = 0 の場合 , "0" 書込みは禁止です。 • DRQI = 1 の場合 , 送信バッファにデータ書込みが可能です。DRQI ビットをクリアすると , 送信バッファにデータ書込みが完了したことになります。DRQI = "1" のとき IN パケット要求があった場合 , NAK 応答が ホストに自動的に送信されます。したがって , このビットは IN 転送要求を処理するためにクリアする必要があります。 • DRQI ビットは USB_EP0IN0 割込み要因です。 • "1" 書込みは無視します。 • "0" 書込みでクリアしてください。リードモディファイライト時は "1" が読み出されます。
Bit9～Bit0	Undefined	• 書込みは動作に影響しません。読出し時は不定です。

29.3.8 EP0Oステータスレジスタ(EP0OSn)

EP0Oステータスレジスタ(EP0OSn)は, エンドポイント0のOut方向転送に関するステータス表示を行います。

EP0Oステータスレジスタ(EP0OSn)

Figure 29-10. EP0Oステータスレジスタ(EP0OSn)



以下に, EP0Oステータスレジスタ(EP0OSn)の各ビットの機能を説明します。

ビット名		機能
Bit15	BFINI	• このビットは , 受信バッファを初期化します。 • BFINI ビットは, UDC 制御レジスタ (UDCCn) のRST ビットが "1" にセットされると , 自動的にセットされます。 • UDCCn:RST ビットでリセット動作が実行された場合, BFINI ビットをクリアする前に UDCCn:RST ビットをクリアしてください。 • BFINI ビットはバッファ,DRQO ビットおよびSPK ビットを初期化します。バッファを初期化する前に, DRQI または DRQO ビットが "1" にセットされている (ホストからのアクセスがない) ことを確認した後, 必要に応じて STAL ビットをセットしてから初期化を行なってください。
Bit14	DRQOIE	• DRQO割込みを許可します。DRQOフラグをセットすると開始される割込みを許可します。
Bit13	SPKIE	• このビットは, SPKフラグ(bit9)の割込み要因による割込みを許可します。
Bit12, Bit11	未定義	• なし.
Bit10	DRQO	• OUT パケットがホストから正常に受信され , データが受信バッファに書き込まれ , 受信データを受信バッファから読み出すことが可能であることを示します。 • 受信バッファのデータリード処理完了後、DRQO をクリアする必要があります。 • DRQO = 0 の場合 , "0" 書込みは禁止です。 • DRQO =1 の場合 , 受信バッファは更新されません。クリアした時点で更新許可となります。DRQO が "1" の状態で OUT パケット要求があった場合は , 自動でホストに NAK 応答します。 • DRQO ビットは USB_EP0OUT0 割込み要因です。 • "1" 書込みは無視します。 • "0" 書込みでクリアしてください。リードモディファイライト時は "1" が読み出されます。
Bit9	SPK	• このビットは , ホストから正常に受信されたデータパケットのバイト数がEP0Cn:PKS で設定した最大パケットサイズ値に満たない場合 (0 パケットもこのビットを設定します) を示します。 • このビットは , USB_F20 割込み要因です。 • "1" 書込みは無視します。 • "0" 書込みでクリアしてください。リードモディファイライト時は "1" が読み出されます。
Bit8, Bit7	予約	· なし
Bit6～Bit0	SIZE	• ホストから EP0 に OUT パケットが転送されると , 受信バッファに書き込まれたデータバイト数が表示されます。SIZE ビットは EP0OSn:DRQO フラグがセットされた時に有効な値に更新されます。 • <例> 8 バイト ⇒ "08_H", 64 バイト ⇒ "40_H" (最大値)

29.3.9 EP1～EP5ステータスレジスタ(EP1Sn to EP5Sn)

EP1～EP5ステータスレジスタ(EP1Sn～EP5Sn)は, エンドポイント1～5に関するステータス表示を行います。

EP1～EP5ステータスレジスタ(EP1Sn ～ EP5Sn)

Figure 29-11. EP1ステータスレジスタ(EP1Sn)

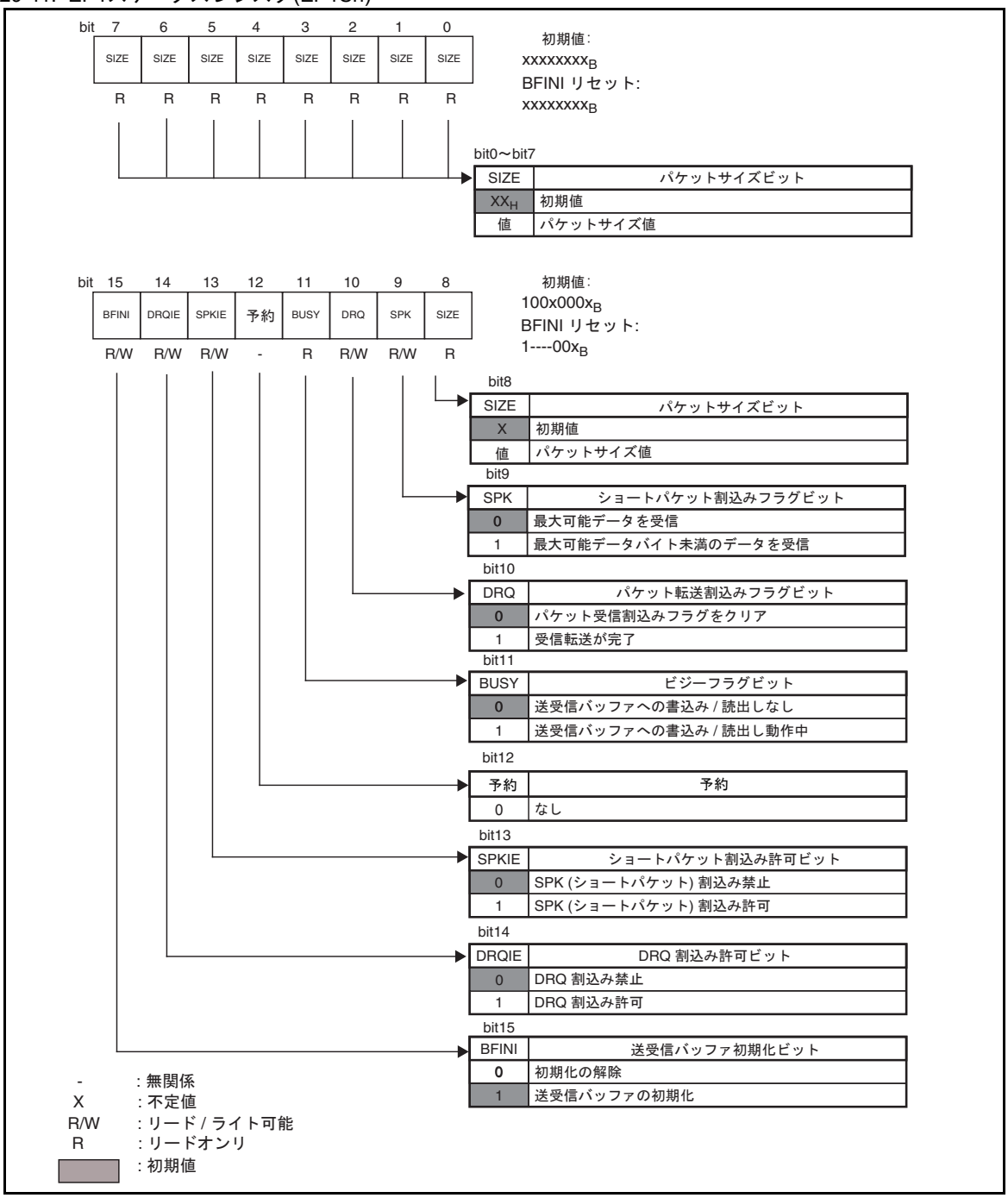
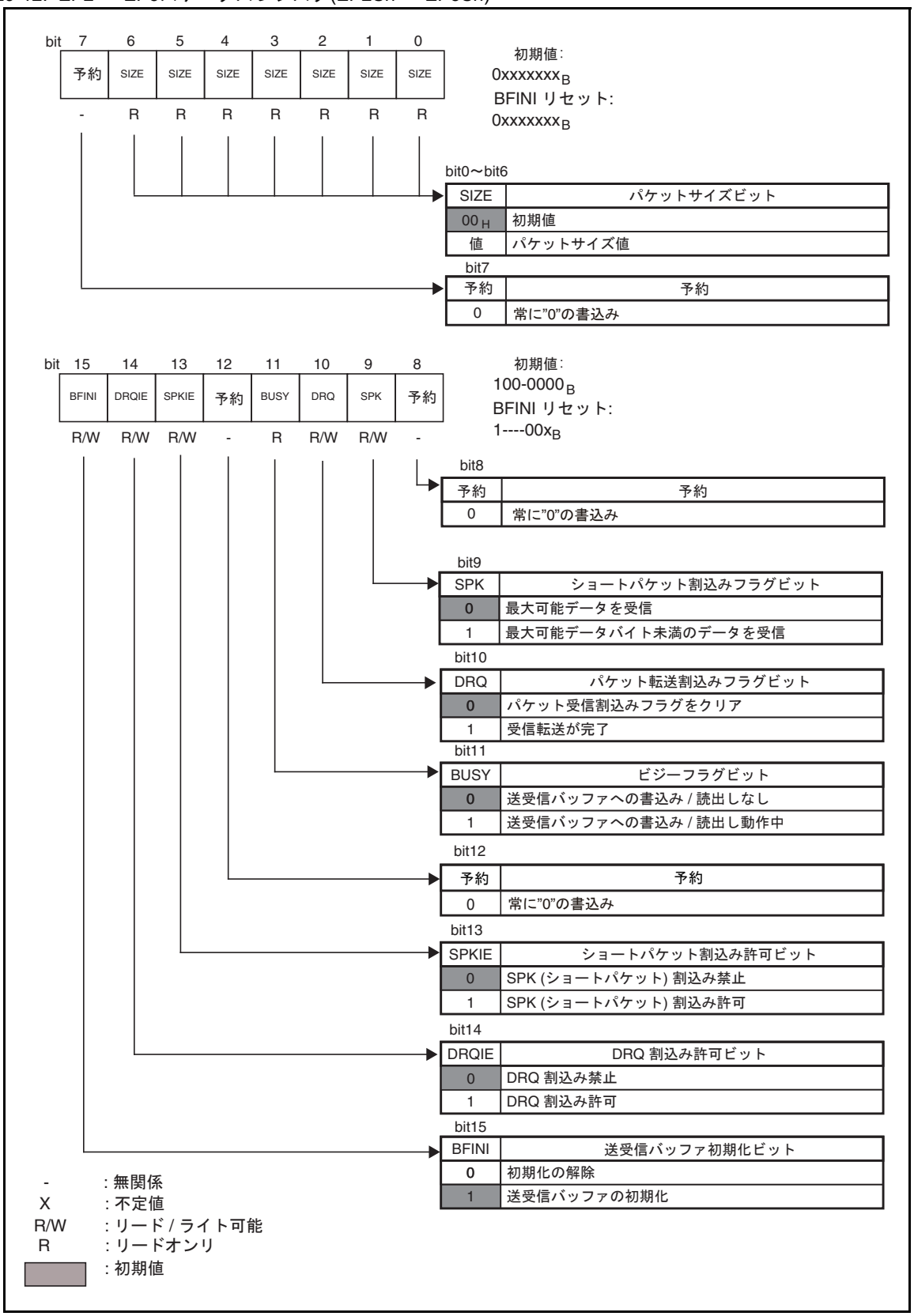


Figure 29-12. EP2 ～ EP5ステータスレジスタ(EP2Sn ～ EP5Sn)



以下に, EP1～EP5ステータスレジスタ(EP1Sn～EP5Sn)の各ビットの機能を説明します。

ビット名		機能
Bit15	BFINI	このビットは , 送受信バッファを初期化します。 ・BFINI ビットは UDC 制御レジスタ (UDCCn) の RST フラグが 1 にセットされると , 自動的にセットされます。 ・RST ビットでリセット動作が実行された場合 , BFINI ビットをクリアする前に RST ビットをクリアしてください。 ・EP1 ～ EP5 の送受信バッファはダブルバッファ構成です。BFINI ビットによる初期化ではダブルバッファ , DRQ, SPK ビットがすべて同時に初期化されます。 • ・バッファを初期化する前に, DRQ ビットが "1" にセットされ, BUSY = "0" である (ホストからのアクセスがない) ことを確認した後, STAL ビットをセットしてください。
Bit14	DRQIE	EP1Sn ～ EP5Sn それぞれの DRQ フラグの割込み要因による割込みを許可します。 • ・自動バッファ転送モード(DMAE="1")では,DRQIEをセットする前にDMA設定を許可する必要があります。
Bit13	SPKIE	• このビットは,EP1Sn～EP5Snレジスタそれぞれの SPK フラグの割込み要因による割込みを許可します。
Bit12	予約	· なし
Bit11	BUSY	このビットは , ホストとの間でデータを送受信するために送受信バッファへの書込み , または読出しが動作中であることを示します。 ・自動でセット , リセットされます。 ・DRQ ビットがセットされた状態で BUSY フラグもセットされている場合 , ダブルバッファの内 CPU または DMA がアクセスしているバッファとは別のバッファがホストからのアクセス中であることを意味します。 • ・通常であればバッファを制御する際,BUSY ビットを参照する必要はありませんが, バッファを初期化する前に, DRQ ビットが "1" にセットされ, BUSY="0" である (ホストからのアクセスがない) ことを確認した後, STAL ビットをセットしてください。
Bit10	DRQ	このビットは , EPx のパケット転送が正常に終了し , データ処理が必要であることを示します。 ・EPxSn:DRQ ビットは対応する USB_EPx0 の割込み要因です。 ・"1" 書込みは無視します。 ・"0" 書込みでクリアしてください。リードモディファイライト時は "1" が読み出されます。 ・(ホストからの IN パケット要求に応じて) ホストに送信するデータを送信バッファに書き込んだ後 , DRQ ビットをクリアする必要があります。OUT パケット転送が完了すると , DRQ ビットが自動で "1" にセットされます。受信バッファからのデータが CPU/DMA によって完全に読み出されたとき , クリアする必要があります。 ・DRQ = "0" のとき , "0" 書込みは禁止です。 ・自動バッファ転送モード (DMAE=1) を使用しない場合は , 送受信バッファのデータリードまたはデータライト処理が済んだら , DRQ ビットに "0" 書込みする必要があります。DRQ ビットをクリアした時点でアクセスバッファを切り替えます。転送方向が IN 方向の設定の場合 , DRQ ビットが "1" で送信バッファにデータを書き込みせずに DRQ ビットをクリアした場合 , 0 バイトのデータパケットがホストに送信されます。 • ・初期設定において EP1 ～ EP5 制御レジスタ (EP1Cn ～ EP5Cn) の DIR を "1" にセットした場合, 対応するエンドポイントの DRQ ビットも同時にセットされます。

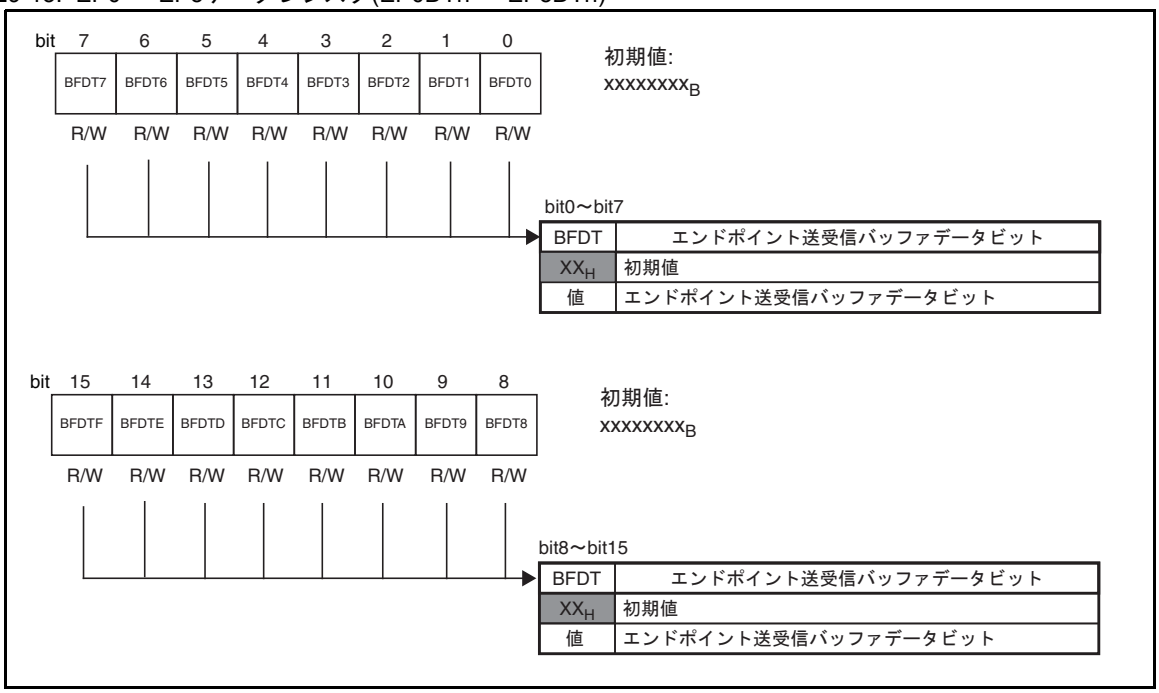
<table>
<tr><th colspan="2">ビット名</th><th>機能</th></tr>
<tr><td>Bit9</td><td>SPK</td><td>このビットは , ホストから正常に受信されたデータバイト数が EPxCn:PKS で設定した最大パケットサイズ値に満たない場合 (0 バイトのデータパケットを含みます) を示します。
・SPK ビットは USB_F20 割込み要因です。
・"1" 書込みは無視します。
・"0" 書込みでクリアしてください。リードモディファイライト時は "1" が読み出されます。
• ・IN 方向のデータ転送時に SPK ビットはセットされません。</td></tr>
<tr><td>Bit8, Bit7</td><td>予約/SIZE
(EP1Sn)</td><td>EP2Sn ～ EP5Sn の場合には使用しないでください。
• ・EP1Sn の場合, これらは SIZE ビットです (以下を参照してください)。</td></tr>
<tr><td>Bit8～Bit0:
EP1Sn
Bit6～Bit0:
EP2Sn～
EP5Sn</td><td>SIZE</td><td>• EP1～EP5のOUTパケット転送終了時に受信バッファに書込まれたデータバイト数が表示されます。SIZEビットはEP1～EP5ステータスレジスタ(EP1S～EP5S)のDRQの割込み要因がセットされたときに有効な値に更新されます。
EndPoint1～EndPoint5の最大転送数は下記になります。
<table>
<tr><th>Endpoint</th><th>最大転送数</th><th>表示範囲</th></tr>
<tr><td>1</td><td>256 バイト</td><td>000h ～ 100h</td></tr>
<tr><td>2 ～ 5</td><td>64 バイト</td><td>00h ～ 40h</td></tr>
</table>
• SIZEはOUT方向パケット転送時でのホストからのバッファ書込みデータパケットサイズを表示するので, IN 方向転送が行われたときに読み出された SIZE 値は意味をもちません。</td></tr>
</table>

29.3.10 EP0～EP5データレジスタ(EP0DTn ～ EP5DTn)

EP0～EP5データレジスタ(EP0DTn～EP5DTn)は, エンドポイント0～5に関する送受信バッファへのリードまたはライトのアクセスレジスタになります。

EP0 ～ EP5データレジスタ(EP0DTn ～ EP5DTn)

Figure 29-13. EP0 ～ EP5データレジスタ(EP0DTn ～ EP5DTn)



以下に, EP0～EP5データレジスタ(EP0DTn～EP5DTn)の各ビットの機能を説明します。

ビット名		機能
Bit15～Bit0	BFDT	• 各エンドポイント用送受信バッファへのデータリード , ライトレジスタです。 • DMA 転送による BFDT レジスタへのアクセスはワードアクセスのみ対応しています。奇数分のデータバイトを転送する場合で DMA 転送の場合 , 最後のデータ転送にバイト転送を設定する必要があります。CPU アクセスでワード転送 (奇数分のデータバイトごとに) を行う場合でも , DMA 転送と同様に最後の転送をバイト転送にする必要があります。 • EP0DTn ～ EP5DTn レジスタへの CPU アクセスは , バイト , ワード共に可能です。レジスタのいずれかに対してバイトアクセスが必要な場合 , 最初に下位バイト (bit7 ～bit 0) へのアクセスとし, 次は上位バイト (bit15 ～bit8) へのアクセスとしてください。以後下位バイト , 上位バイトを交互にアクセスしていってください。 • EP0DTn ～ EP5DTn レジスタへのビットアクセスは禁止です。

29.4 USBファンクションの動作説明

USBファンクションの基本について説明します。

29.4.1 USBファンクションの動作

USBファンクションはUSBプロトコルをサポートするホストコントローラと双方向のパケット転送を行います。ホストとデバイスの接続 , 構成はエニュメレーションにより実施されます。そのあとにデバイスドライバを使用した各種の転送タイプでの通信が行われます。

エニュメレーションを例にホストと USB デバイスの USB 通信の動作について説明します。

全体の処理内容を理解するためのレジスタおよび USB パケット転送の動きを示します。

エニュメレーション処理

エニュメレーション処理は , USB が動作する上でデバイスをバスに接続するときに一番初めにホストと USB デバイスとの接続を確立する処理です。USB のバス上にどのようなデバイスが接続されているか , どのようなパラメータが必要か , エンドポイントがいくつあるかなどを , ホストが USB コントロール転送 (USB 転送タイプのひとつ) を使って調査します。USB コントロール転送は 6 本あるエンドポイントのうち EP0 (EndPoint0) を使用します (USB 仕様) 。

EP1 ～ EP5 を使用するためには USB バス上で以下を受信する必要があります。

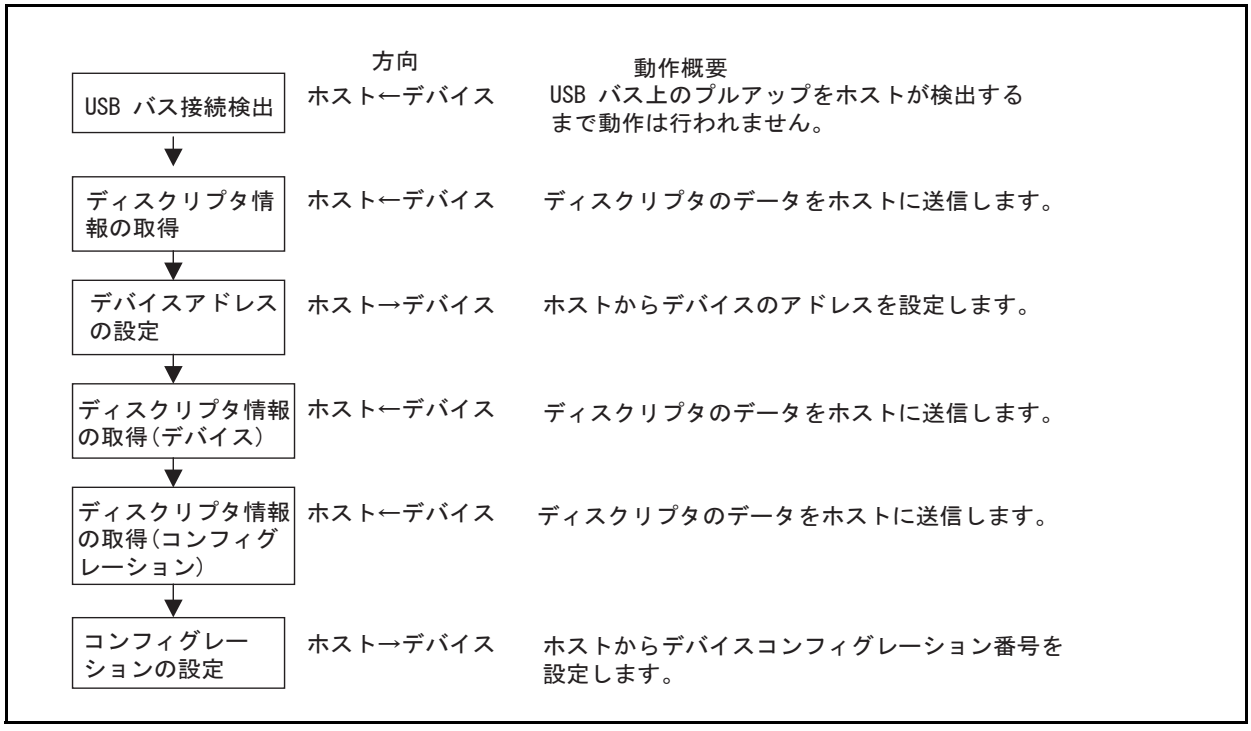
(つまり , 基本エニュメレーション処理が完了されている)

① USB バスリセット (最低 40 USB クロックサイクルで UDP が "0", UDM が "0")

② SET_ADDRESS によるアドレスセット

③ SET_CONFIG による構成セット

Figure 29-14. ケーブル端子接続例



接続検出

ホストは USB バスの 2 本の信号線 (UDP, UDM) を監視し , 信号が "H" レベルになることによりデバイスが接続されることを認識します。

自己電源デバイスとして使用する場合の詳細な手順は「29.4.1 接続検出と切断検出」を参照してください。

レジスタ初期設定と動作開始手順例

レジスタ初期設定例と動作開始例を示します。

① EP0Cn レジスタによる EP0 の設定 (パケットサイズなど)

② 各エンドポイントの EPEN, DIR, TYPE などを設定 (EP1Cn ～ EP5Cn レジスタを参照してください)

③ UDCCn レジスタの RST ビットのクリア

④ EP0ISn, EP0OSn, EP1Sn ～ EP5Sn レジスタの BFINI をクリア

⑤ UDCCn レジスタの HCONX ビットのクリア

USBバスリセット

バスリセットがホストで発行され USB デバイスを初期化します。

デバイスは以下の順序で処理が必要です (USB 接続後の最初のバスリセットでは処理の必要はありません) 。

① UDCCn レジスタの RST ビットで USB ファンクションを初期化

② 使用する送受信バッファ , 関係する制御レジスタを再設定

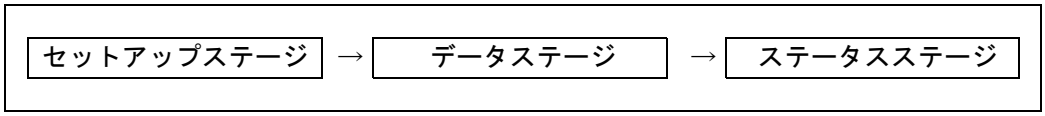
③ ファームウェアの制御をエニュメレーション前に戻します。

ディスクリプタの取得

ホストから USB デバイスに要求がありデータをホストに通知します。

詳細には以下の 3 つのステージに分かれて通信されます 。

Figure 29-15. 通信ステージ



セットアップステージでは , デバイスがホストから正常なパケット (特定の要求 , 例えば , GET_DESCRIPTOR) を受信するか確認し , コマンドをデコードして確認し , ステータスステージでホストに返すディスクリプタの情報を送信バッファに用意します。データステージでは単にホストから正常なデータが送受信されるかを確認します (SET 要求または GET 要求であるかによって異なります)。ステータスレジスタでは , 要求全体のステータスを報告し , ホストがデータなしパケットを (データステージで送信される DESCRIPTOR に応じて) 送信するときに終了処理をします。

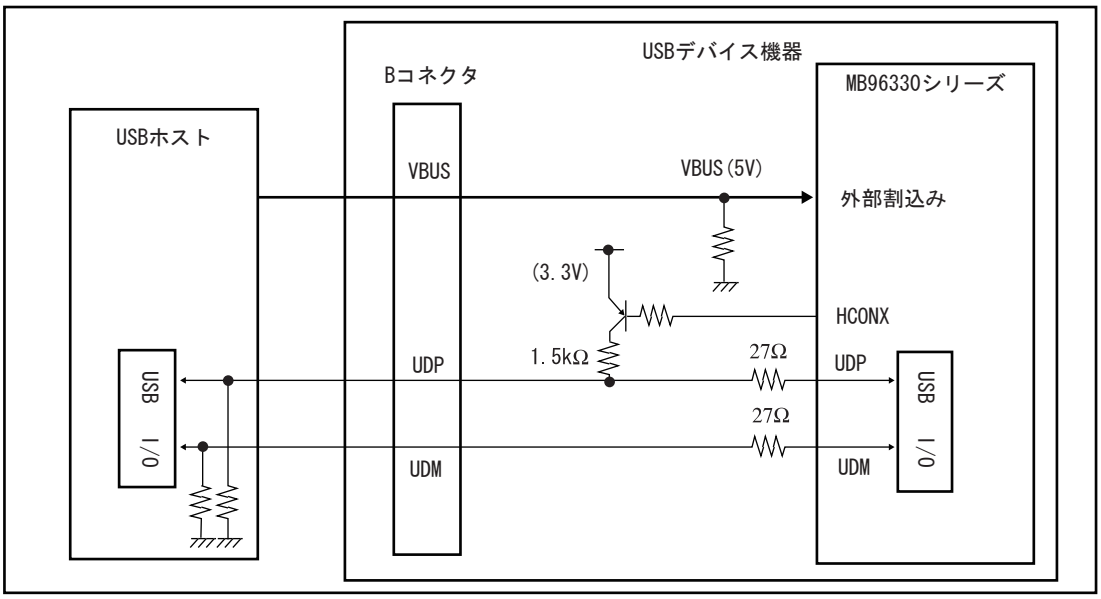
29.4.2 接続検出と切断検出

USBホストとの接続検出と切断検出について説明します。

29.4.2.1 USBの接続例

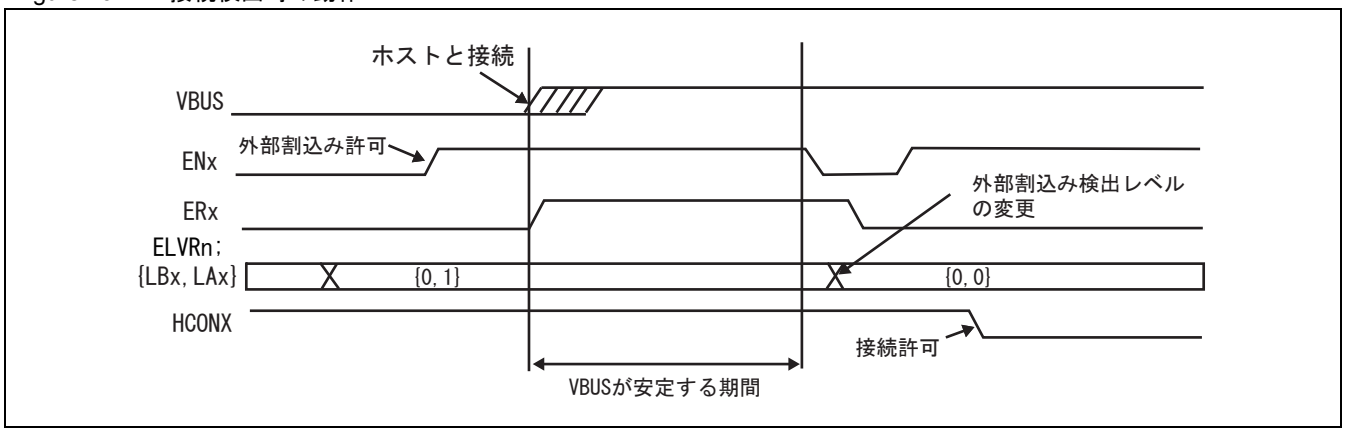
外部割込み端子を USB コネクタの VBUS 端子に接続し , プルダウン抵抗を接続することで USB ホストとの接続と切断を検出することができます。Figure 29-16に,USBコネクタのUDP, UDM, VBUS端子との接続例を示します。

Figure 29-16. USBの接続例



接続検出

Figure 29-17. 接続検出時の動作



デバイスは以下の順序によりホストとの接続を検出します。

① VBUS に接続した外部割込みを "H" レベル入力検出に設定して割込み許可します。

② 外部割込み端子の "H" レベル検出で USB ホストとの接続を認識します。これが発生したら , VBUS が安定するのを待ちます。

③ 外部割込みをいったん禁止にします。外部割込みを "L" レベル入力検出に設定し (ホストから断線を検出するために), 外部割込みフラグをクリアして再び外部割込みを許可します。

④ 初期設定を行います。(USB ファンクションレジスタを含むすべての初期化)「レジスタ初期設定例と動作開始手順例」参照

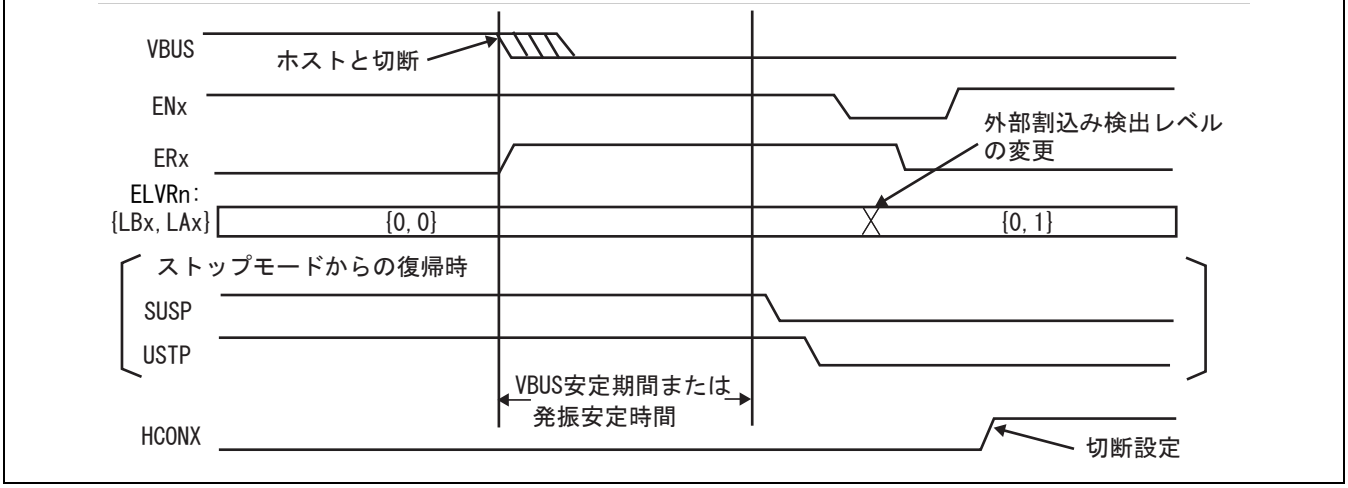
⑤ UDCCn:HCONX ビットに "0" を書き込んでクリアし , UDP のプルアップ抵抗を許可します (プルアップ抵抗の制御をしていない場合でも HCONX ビットをクリアします)。

＜注意事項＞

外部割込み端子にノイズフィルタを外付けした場合は上記のVBUS安定期間をプログラムで待つ必要はありません。

切断検出

Figure 29-18. 切断検出時の動作



USB デバイスは以下の順序によりホストとの切断を検出します。

① VBUS に接続された外部割込み端子の "L" レベル検出で USB ホストが切断されたことを認識します。

② ストップモードからの復帰時：

発振安定待ち時間後に UDCSn:SUSP, UDCCn:USTP の順にクリアします。

ストップモード以外のとき：

VBUS が安定する期間を待ちます。

③ いったん外部割込みを禁止します。外部割込み端子の設定を "H" レベル入力検出に変更し , 外部割込みフラグをクリアして再び外部割込みを許可します。

④ UDCCn: HCONX ビットを "1" にセットして , UDP のプルアップ抵抗を切断します (プルアップ抵抗の制御をしていない場合でも UDCC レジスタの HCONX ビットを "1" にセットします)。

＜注意事項＞

外部割込み端子にノイズフィルタを外付けした場合は上記のVBUS安定期間をプログラムで待つ必要はありません。

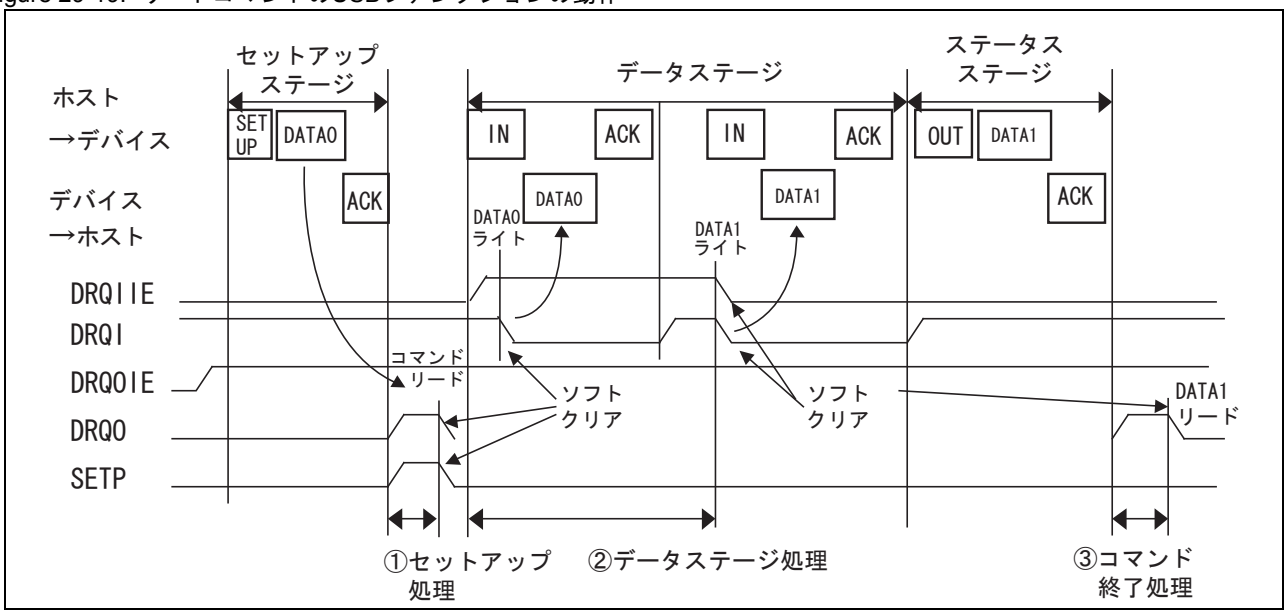
29.4.3 コマンド応答時の各レジスタ動作

基本となるUSBレジスタの動作およびUSBパケットの処理方法を説明します。CPU割込みによるファームウェアの処理について説明します。

29.4.3.1 リードコマンドのUSBファンクションの動作

GetDescripter, SynchFrame, クラスベンダーコマンドの場合

Figure 29-19. リードコマンドのUSBファンクションの動作



セットアップ動作

セットアップパケットが USB デバイスによって受信された時点で , DRQO と SETP が "1" にセットされます。DRQO がセットされた時点で CPU 割込みに入り , 割込みサービスルーチンで SETP フラグを確認します。続いて , USB デバイスは SETP がセットされていれば受信バッファにあるコマンドを必要な分読み出して (必ずしも 8 バイトすべてを読み出す必要はありません。読み出すバイト数は EP0OSn レジスタの SIZE 値によって異なります), コマンドを解読して各種設定処理し , "0" 書込みにより SETP フラグ , DRQO 割込み要因 (EP0OSn:DRQO) をクリアし , 割込みサービスルーチンから戻ります。

データステージ動作

コマンドの解読の結果, データステージが IN 方向転送の場合, DRQIIE ビットを許可します (DRQI ビットは初期値 "1" であるためセットする必要はありません)。DRQI 割込みのための CPU 割込みサービスルーチンにて , ホストに送信されるデータを送信バッファに転送します。転送終了後 , DRQI 割込み要因 (EP0ISn:DRQI) をクリアした後 , 割込みから戻ります。

ホストへのデータパケット転送が終了すると DRQI が再びセットされます。DRQI がセットされた時点で CPU 割込みに入り , DRQI に対応する割込みサービスルーチンに再び入ります。ホストに送信される次のデータパケットに備えデータを送信バッファに転送します。転送終了後 , DRQI 割込み要因 (EP0ISn:DRQI) をクリアして割込みサービスルーチンから戻ります。

＜注意事項＞

このUSB ファンクションは , USB 2.0 で新たに追加されたコマンドには対応していない ため , Get Descripter コマンドに対して USB リビジョンは 1.1 で応答してください。

ステータスステージ動作

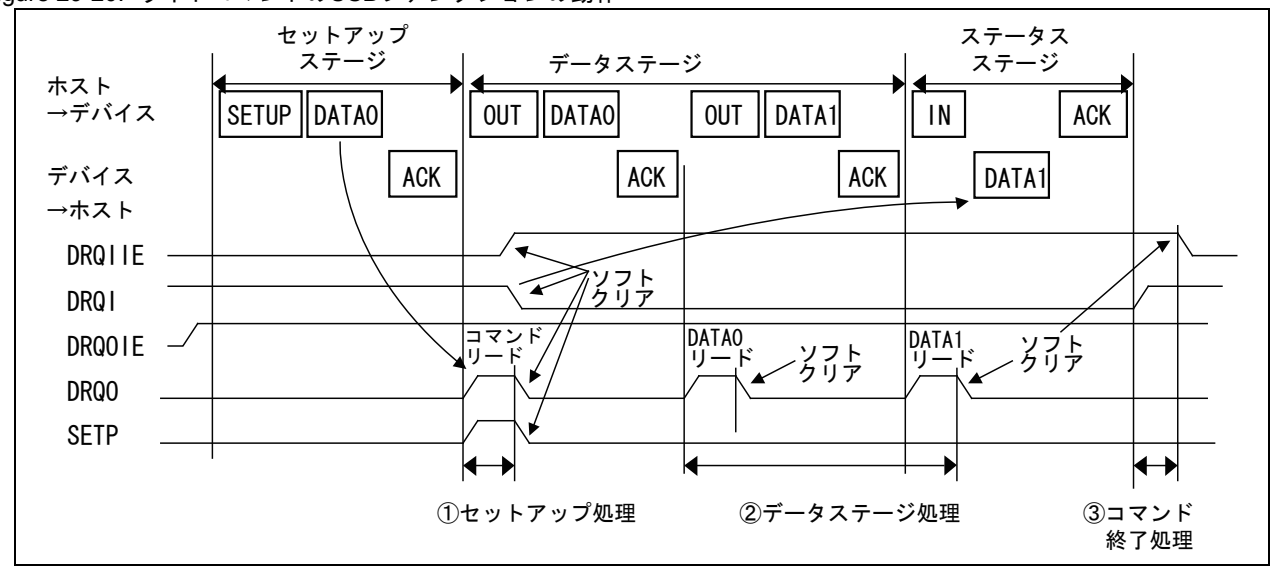
データステージが完了すると , DRQI がセットされます。DRQI 割込みが許可されない間 (DRQIIE= "0") は , CPU は DRQI の割込みルーチンを実行しません。ホストからのデータなしの OUT パケットを受信すると DRQO は "1" にセットされます。

CPUは, DRQI割込みを処理し, 受信されたバイト数は, EP0OSnのSIZEビットで確認されます。もしEP0OSn:PKSがゼロ (バイトデータパケットを受信していない) なら次のセットアップステージに備えるためにDRQO割込み原因をクリアしてください。

29.4.3.2 ライトコマンドのUSBファンクションの動作

SetDescripter, クラスベンダーコマンドの場合

Figure 29-20. ライトコマンドのUSBファンクションの動作



セットアップ動作

セットアップパケットを受信したときに DRQO と SETP がセットされます。DRQO がセットされた時点で DRQ0 に対応する CPU 割込みに入り , 割込みサービスルーチンで SETP フラグを確認します。続いて , USB デバイスは SETP がセットされていれば受信バッファにあるコマンドを必要な分読み出して (必ずしも 8 バイトすべてを読み出す必要はありません。読み出すバイト数は EP0OSn レジスタの SIZE 値によって異なります), コマンドを解読して各種設定処理し , ステータスレジスタの 0 バイト応答に備え , 送信バッファへデータを書き込まずに DRQI ビットをクリアします (DRQI の初期値は "1" のため)。続いて , DRQI による割込み要因を許可し , ステータスステージの正常終了を確認するために DRQIIE をセットしておきます。また "0" 書込みにより SETP フラグ , DRQO をクリアし割込みルーチンから復帰します。

データステージ動作

OUT 方向のデータステージが終了すると DRQO がセットされます。DRQO がセットされた時点で割込みを処理します。EP0OSn レジスタの SIZE を確認し , DMA で受信バッファからデータを転送するか , CPU で直接受信バッファからデータを読み出します。その後 , DRQO をクリアして割込みから復帰します。

ステータスステージ動作

データステージが終了すると , ホストが IN トークン要求を送信します。データなしパケットがこの要求に応じてデバイスから送信されます。ステータスステージが終了すると , DRQI によって CPU 割込み USB_EP0IN が開始され , プログラムは割込みサービスルーチンに入り , ステータスステージが正常に終了したかを確認します。続いて EP0ISn:DRQIIE をクリアしてから割込みポイントに復帰します。

29.4.4 STALL応答と解除

エンドポイント0の場合と, エンドポイント1～エンドポイント5の場合それぞれに対してSTALL応答と解除方法について説明します。

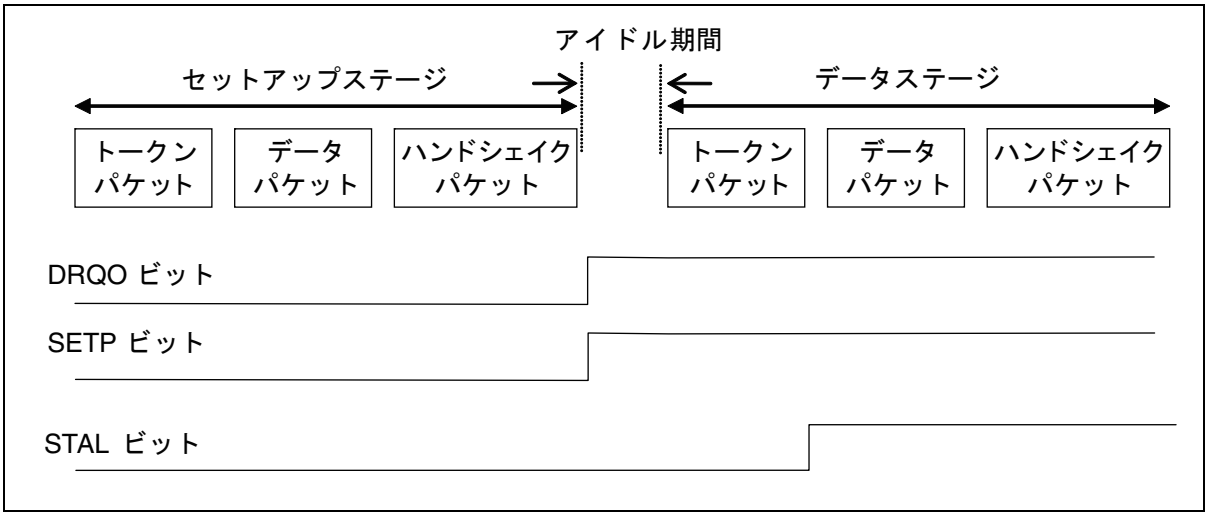
エンドポイント0のSTALL応答 / 解除について

エンドポイント 0 の STALL 応答 / 解除制御は , EP0 制御レジスタ (EP0C) の STAL ビットにて行います。

- STAL ビットのセットタイミング

  STALL 応答は , コントロール転送のセットアップステージである事を示す SETP ビット =1 検出 (DRQO ビット =1 割込み) にてコマンドを解釈し , STALL 応答が必要な場合に STAL ビットをセットします (Figure 29-21 参照)。STAL ビットセット後に割込み要因 (DRQO ビット) をクリアしてください。

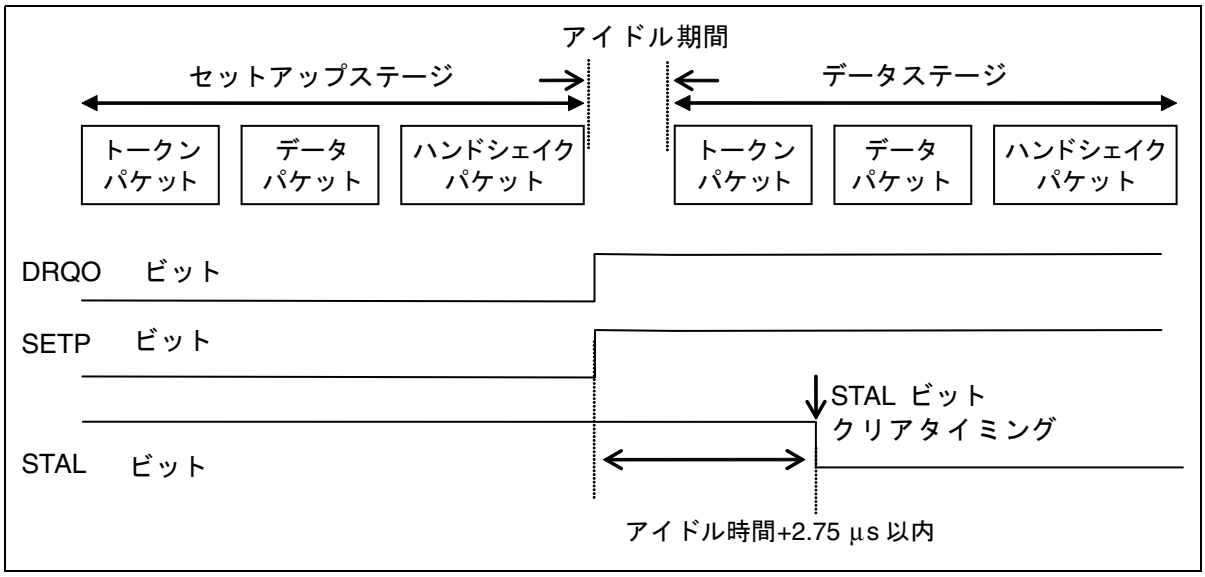
Figure 29-21. STALビットセットのタイミング



- STAL ビットのクリアタイミング

  STALL 解除は , コントロール転送のセットアップステージであることを示す SETP ビット =1 検出 (DRQO ビット =1 割込み) にて STAL ビットクリアします (Figure 29-22 を参照してください)。

Figure 29-22. STALビットクリアのタイミング



STALL 応答解除 (STAL ビットクリア) は , SETP ビット =1 (DRQO ビット =1 割込み) を検出してから , 次のデータステージのデータパケット送受信開始前に , STAL ビットクリアしてください。DRQO ビット =1 から , STAL ビットクリアまでの時間は下記のとおりです (転送速度：Full Speed 12Mbps 時)。下記時間内に STAL ビットがクリアされない場合 , データステージのハンドシェイクパケットにて STAL 応答します。

DRQO ビット =1 検出から STAL ビットクリアまでの時間： アイドル時間 +2.75µs 以内

? アイドル期間が最小の 2 ビット転送時間である場合は約 2.9µs 以内となります。

STAL ビットクリアが , 上記時間内で対応できない場合には , USB ホスト側のドライバソフトでアイドル時間を長くする等の対応を行ってください。

エンドポイント1～エンドポイント5のSTALL応答 / 解除について

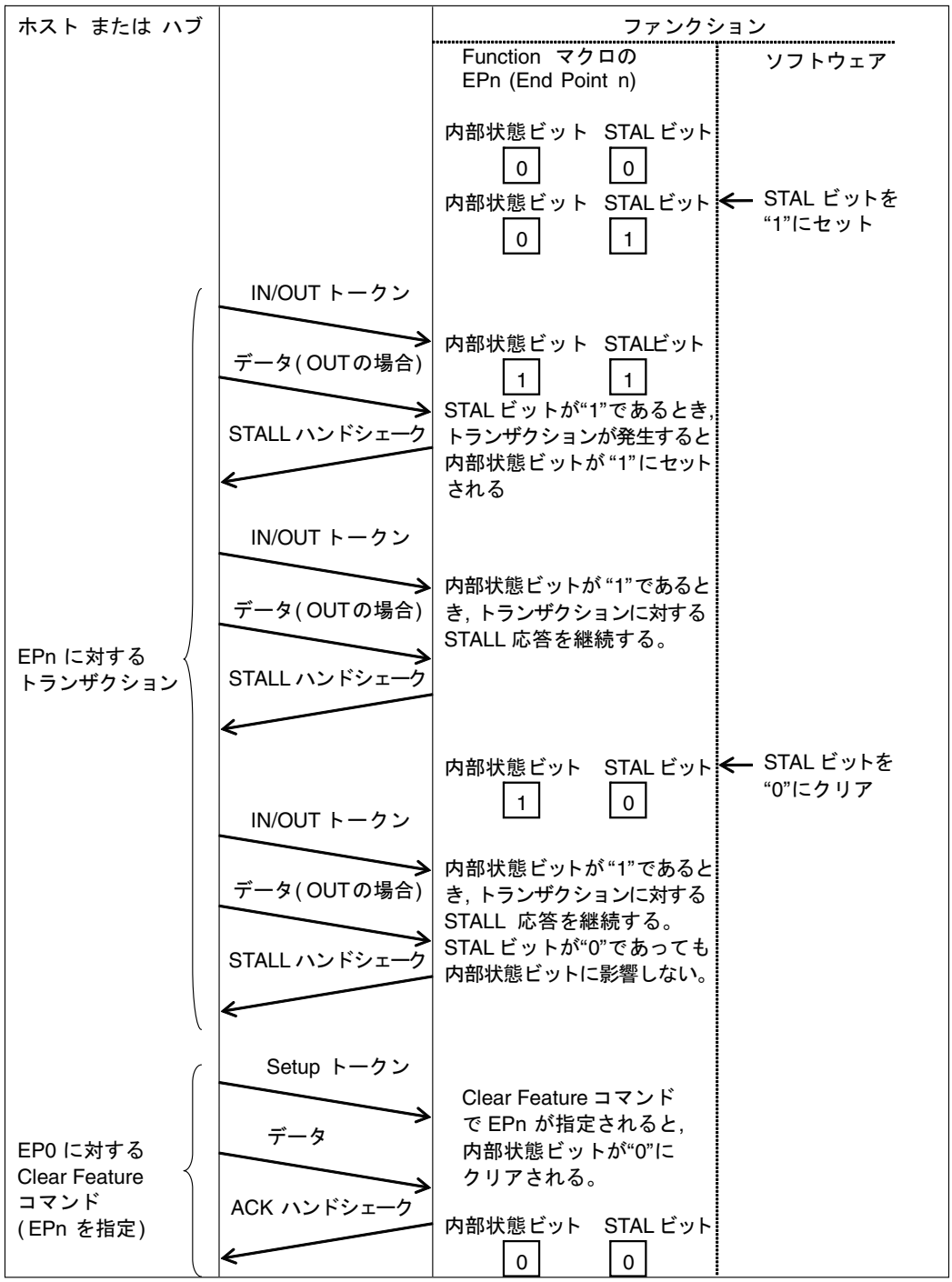
エンドポイント 1 ～エンドポイント 5 の STALL 応答 / 解除制御は , EP1 ～ EP5 制御レジスタ (EP1C ～ EP5C) の STAL ビットと内部状態ビットで行われます。

- ソフト処理にて STALL 応答したい場合

  ソフト処理で STALL 応答する場合の手順を Figure 29-23 に示します。STALL 応答する場合 , 該当するエンドポイントの STAL ビットをソフトでセットします。このとき , 内部状態ビットは変化しません。次に , ホストから STAL ビットがセットされているエンドポイントに対してトランザクションが発生したとき , ハードが自動的に該当エンドポイントの内部状態ビットをセットし , ホストに対して STALL 応答します。一度 , 内部状態ビットがセットされた後は , STAL ビットをクリアしても , ホストから Clear Feature コマンドが発行される

まで , 内部状態ビットはセットされたままとなり , STALL 応答を継続します。また , Clear Feature コマンドで内部状態ビットがクリアされても , STAL ビットがセットされている場合は , 該当するエンドポイントに対するトランザクションが発生するたびに内部状態ビットがセットされるため STALL 応答を継続します。したがって , STALL 応答を解除するためには , STAL ビットをクリアし , さらに Clear Feature コマンドで内部状態ビットをクリアする必要があります。

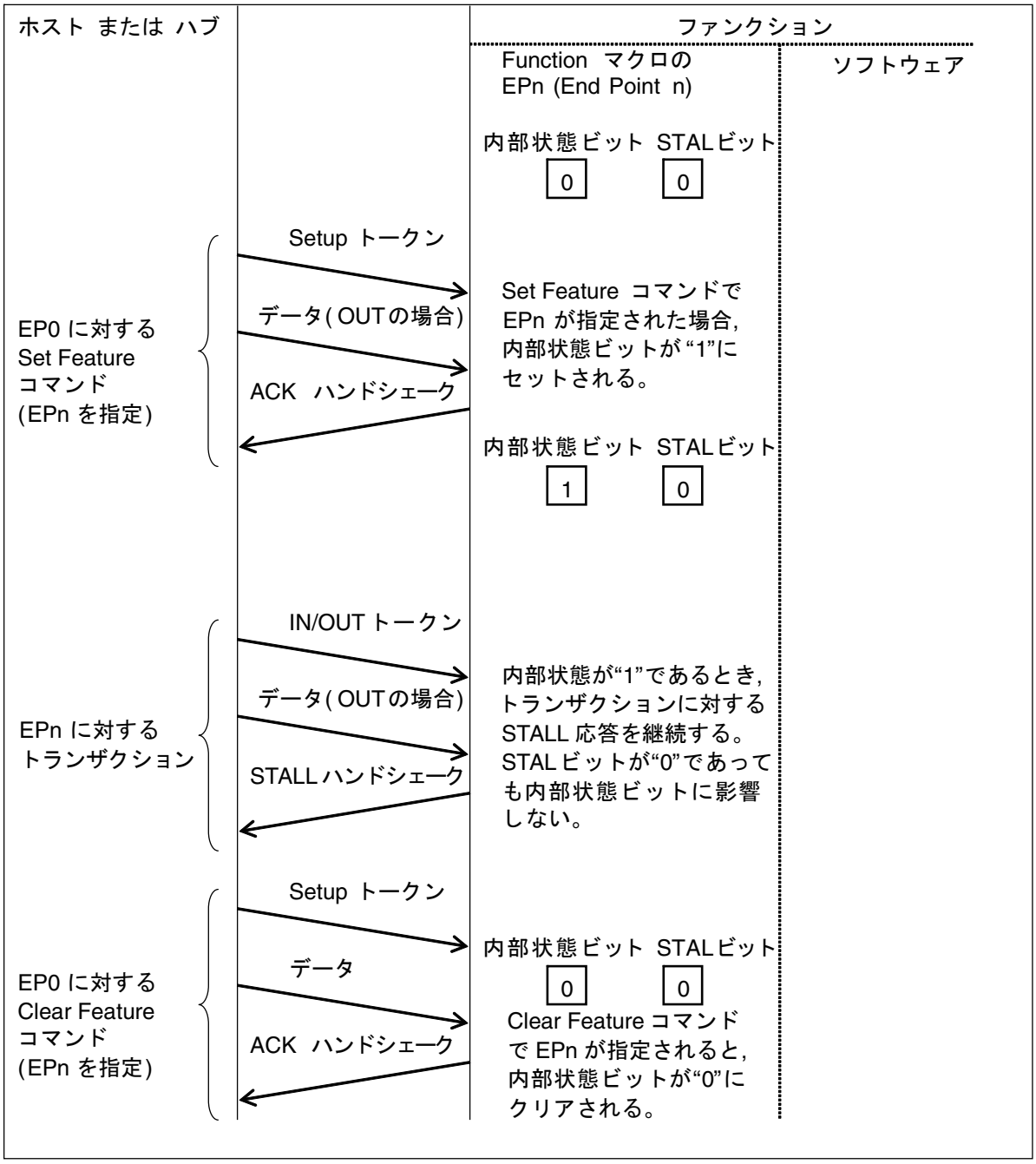
Figure 29-23. ソフト処理にてSTALL応答したい場合



- ハードが自動で STALL 応答する場合

  ハードが自動で STALL 応答する場合の手順を Figure 29-24 に示します。Set Feature コマンドで STALL 応答設定された場合 , STAL ビットに関係なく , ハードが自動的に該当エンドポイントの内部状態ビットをセットし , STALL 応答します。一度 , 内部状態ビットがセットされた後は , STAL ビットに関係なく , ホストから Clear Feature コマンドでクリアされるまで , 内部状態ビットは保持されます。Clear Feature コマンドで該当ビットがクリアされた後は , STAL ビットを参照するようになります。したがって , STALL 応答を解除するには , Clear Feature コマンドで内部状態ビットをクリアする必要があります。

Figure 29-24. ハードが自動でSTALL応答する場合



29.4.5 サスペンドモード機能

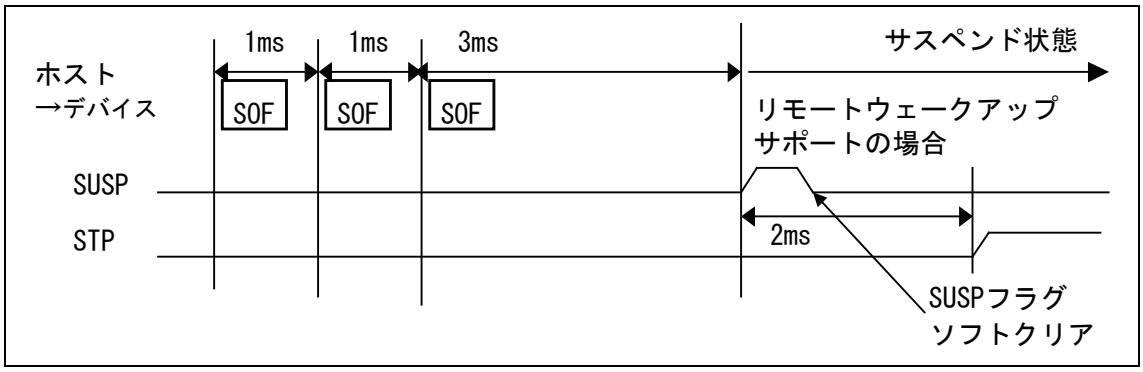
USBデバイスはバス電源の構成ではサスペンドモードにおいて500μA以下に消費電力を落とす必要があります。ここではデバイスがサスペンドモードに移行し, STOPモードになるまでの説明します。

29.4.5.1 USBファンクションのサスペンドモード

USB デバイスコアがサスペンドモード要求を検出した場合に UDCSn レジスタの SUSP フラグが有効にセットされます。

以下にサスペンド動作の例を示します。

Figure 29-25. サスペンド動作



サスペンドモード動作

バス上に 3 ms 以上動作がない場合 , USB ファンクションはサスペンドモードを検出し ,
割込み要因である UDCSn レジスタの SUSP フラグがセットされます。リモートウェイクアップをサポートする USB デバイスの場合はここからさらに 2 ms 待ち (これはこの期間リモートウェイクアップさせないためです) ストップモードに設定します。

29.4.6 ウェイクアップ機能

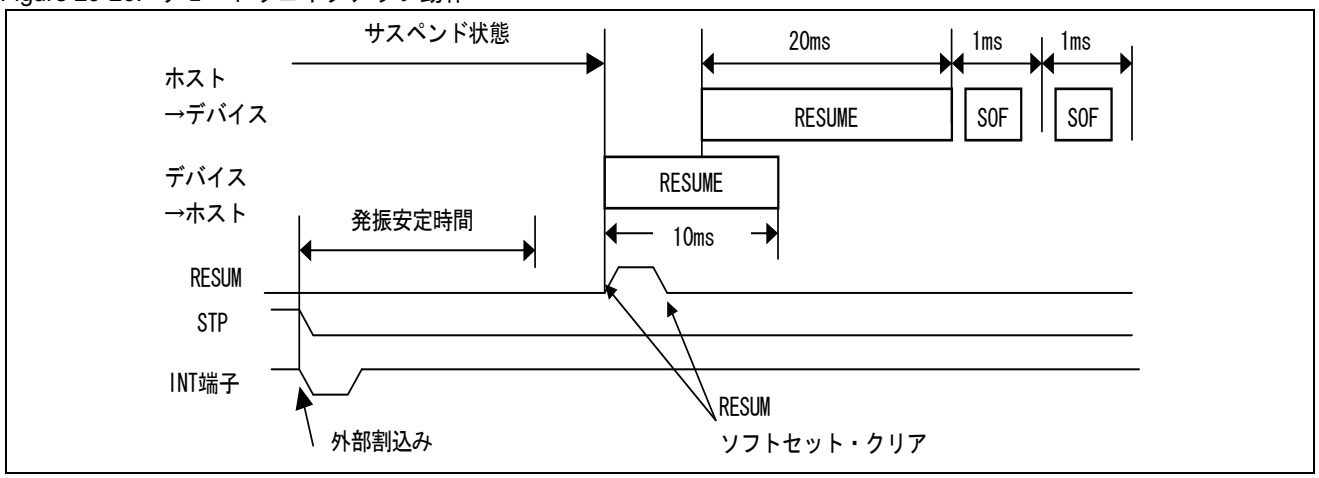
USB デバイスをサスペンドモードからウェイクアップモードにするには USB プロトコルで 2 つの手段があります。

- デバイスからのリモートウェイクアップ
- ホストからのウェイクアップ

各々について説明します。

デバイスからのリモートウェイクアップ

Figure 29-26. リモートウェイクアップ動作

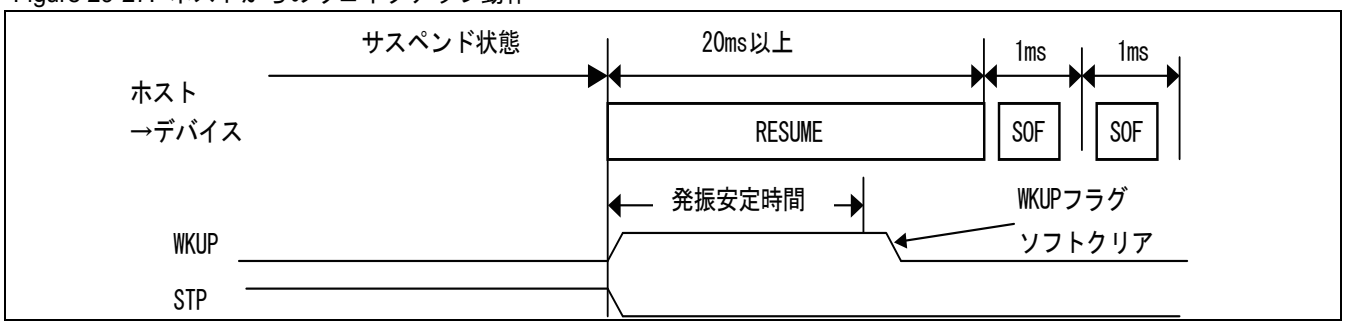


USB デバイスは以下の順序で処理が必要です。

1. 外部割込みでデバイスをストップモードから復帰させます。
2. UDCCn: RESUM ビットをセットします。
3. UDCCn: RESUM ビットをクリアします。

ホストからウェイクアップ

Figure 29-27. ホストからのウェイクアップ動作



デバイスは以下の処理が必要です。

1. 発振安定時間が 10 ms を超えないように設定
2. WKUP 割込みにより , 割込み要因である UDCSn の WKUP フラグをクリアする割込みルーチンに入りメインプログラムに復帰します。

29.4.7 DMA転送機能

データをUSBファンクションバッファと内蔵RAMとの間でDMA転送することが可能です。DMA転送には1パケット単位ごとに転送数を設定し転送するパケット転送モードと，設定した転送データ数分を1回の設定で転送するデータ数自動転送モードの2つが選択できます。

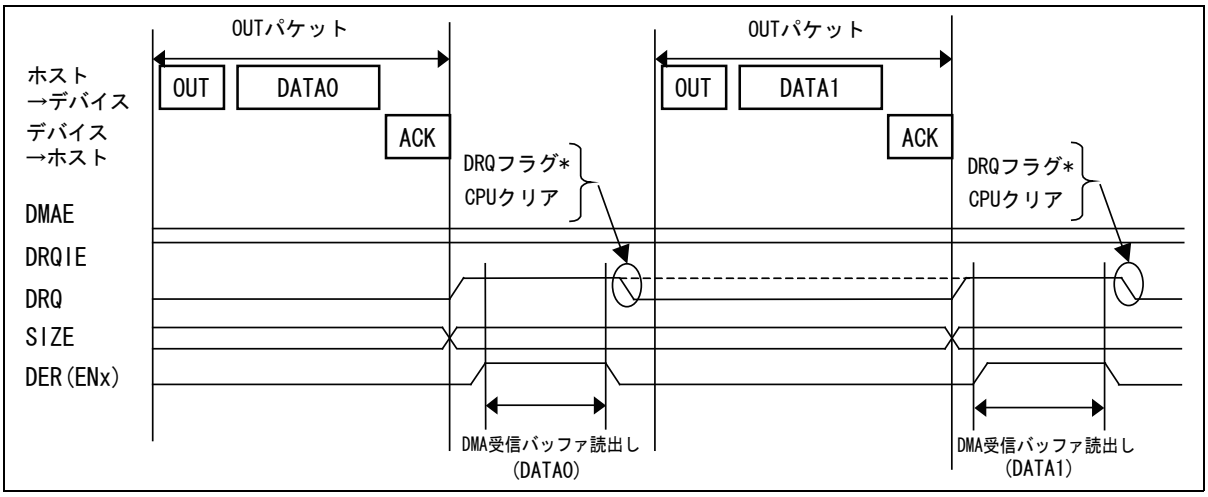
29.4.7.1 パケット転送モード

1 パケット単位ごとに転送数を DMA に設定し転送終了後 , 割込み要因をクリアして転送するパケット転送モードです。転送モードは各エンドポイントバッファへのアクセスも可能です。

OUT 方向 , IN 方向のそれぞれでバッファをアクセスするタイミングを次に示します。

OUT方向(ホスト → デバイス)転送

Figure 29-28. OUTパケット転送



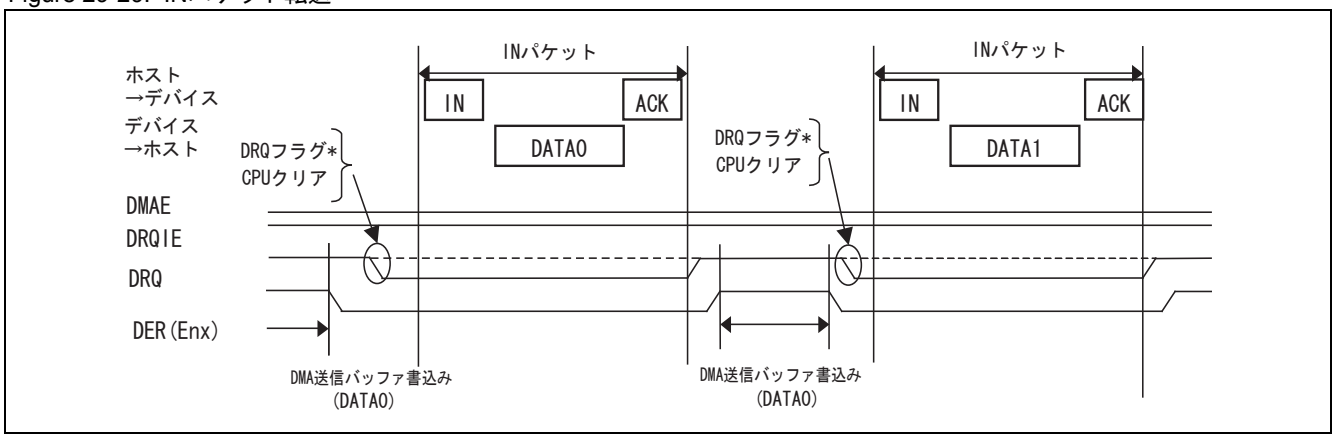
OUT 方向転送では USB デバイスは以下の順序で処理が必要です。

1. DRQ フラグがセットされ , 割込みルーチンが呼び出されて受信データバイト数を確認します（EPxSn:PKS)。
2. 転送データ数を DMA のデータカウンタレジスタ DCT に設定し DER レジスタで DMA を許可します。
3. 転送後 , EPxSn レジスタの該当する DRQ フラグと DMA の DSR レジスタの該当する割込み要因フラグをクリアします。

*: 各エンドポイント EP1 ～ EP5 はダブルバッファ構成となっており , アクセスしていないバッファが空の状態で , 他方のバッファのデータが読み出されたときにのみ , DRQ をクリアできます。アクセスしていない方のバッファに読み出されたデータが残っていた場合は , "0" をライトしてもクリアされません (破線状態)。この場合 , 連続して DRQ による割込み処理に入ることになります。

IN方向(デバイス → ホスト)転送

Figure 29-29. INパケット転送



IN 方向転送では USB デバイスは以下の順序で処理が必要です。

1. DRQ フラグがセットされ , 割込みルーチンに入ったら , 次 IN パケットで転送する転送データ数を DMA のデータカウンタレジスタ DCT に設定し DER レジスタで DMA を許可してデータを RAM からエンドポイントバッファへ転送します。
2. DMA 転送後 , EP1Sn ～ EP5Sn レジスタの該当する DRQ フラグと DMA の DSR レジスタの該当する割込みフラグをクリアして割込み処理から復帰します。

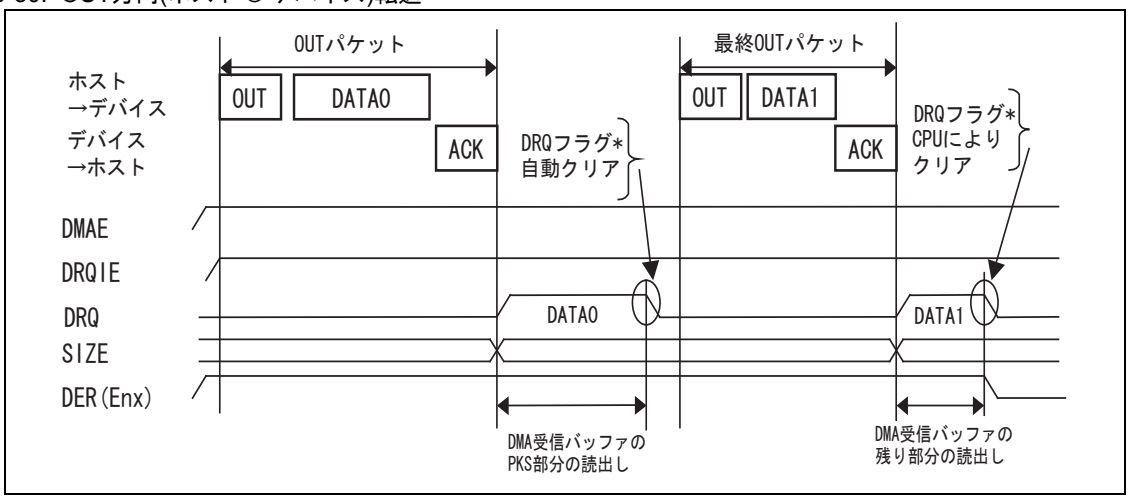
*: 各エンドポイント EP1 ～ EP5 はダブルバッファ構成となっており , アクセスしていないバッファに既にデータが書き込まれている状態で , 他方のバッファにデータが書き込まれたときにのみ DRQ をクリアできます。アクセスしていない方のバッファが空の場合は "0" をライトしてもクリアされません (破線状態)。この場合 , 連続して DRQ による割込み処理に入ることになります。

29.4.7.2 データ数自動転送モード

DMA にあらかじめ転送する総データ数を設定し転送許可ビットもセットしておきます。EPxCn:DMAE が許可されていてホストからの正常受信後 (受信バッファから受信データの読出し準備が完了), DRQ がセットされると , EPxSn レジスタの PKS 分のデータ数を転送した後に自動で DRQ をクリアします。(DRQ フラグが実際にクリアされるかはダブルバッファが共に空になるか , フルになるかによります。) 以降 , ラストデータ転送以外は , DMA に設定した転送データ数分 , 転送完了するまで同様の処理を繰り返し行います。データ数自動転送モードでのラストデータ転送については , DRQ 割込み要因は自動的にクリアされず , 割込みサービスルーチンで CPU によりクリアされます。次の転送を行う場合は , ラストデータ転送後に CPU 割込みに入った時 , そこで DMA の再設定を行い , DMA を許可して , CPU 割込みから復帰します。データ数自動転送モードは DMAE=1 として使用するのでエンドポイント 1 から 5 に対するバッファアクセスのみ有効となります。OUT 方向 , IN 方向のそれぞれでバッファをアクセスするタイミングを次に示します。

OUT方向(ホスト → デバイス)転送

Figure 29-30. OUT方向(ホスト ® デバイス)転送



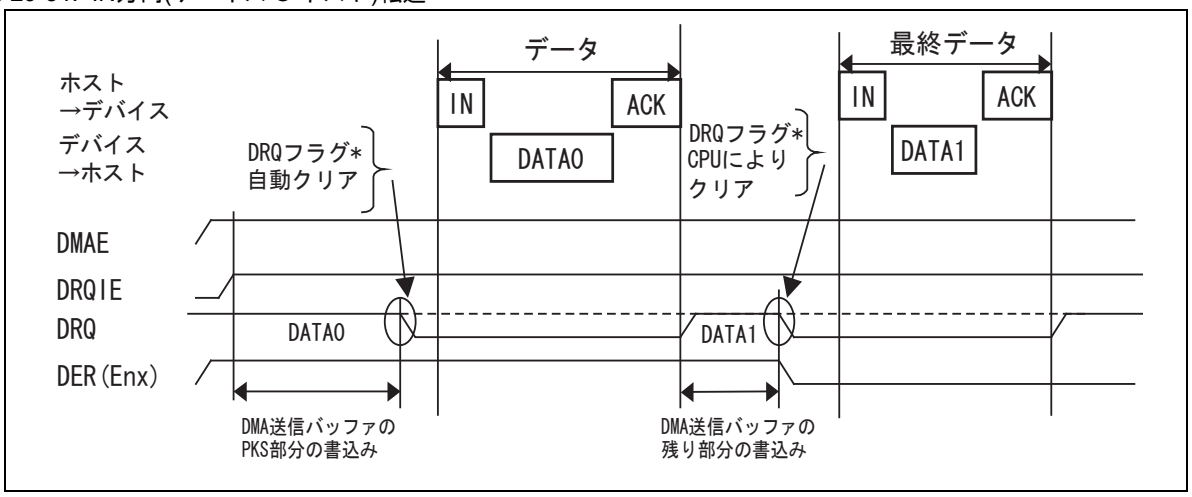
OUT 方向転送では USB デバイスは以下の順序で処理が必要です。

1. 転送する総データ数を DMA のデータカウンタレジスタ DCT に設定し , DER レジスタで DMA を許可します。
2. EPxCn:DMAE, EPxSn:DRQIE を許可設定します。
3. 転送後 , DMA 割込みルーチンが呼び出され , フラグ (DRQ と DTEx) をクリアし , 次の転送で設定を行い , 割込みルーチンから復帰します。

*: 各エンドポインタ EP1 ～ EP5 はダブルバッファ構成となっており , アクセスしていないバッファが空の状態で , 他方のバッファのデータが読み出されたときにのみ DRQ をクリアされます (自動リセット)。アクセスしていない方のバッファに読み出されたデータが残っていた場合は "0" をライトしてもクリアされません (破線状態)。この場合 , 連続して DRQ による割込み処理に入ることになります。

IN方向(ホスト → デバイス)転送

Figure 29-31. IN方向(デバイス ® ホスト)転送



IN 方向転送では USB デバイスは , 以下の順序で処理が必要です。

1. 転送する総データ数を DMA のデータカウンタレジスタ DCT に設定し DER レジスタで DMA を許可します。
2. DMAE, DRQIE を設定します。
3. 転送後 , DMA 割込みルーチンが呼び出され , DMA を再設定し , フラグをクリアして割込みルーチンから復帰します。

*: 各エンドポイント EP1 ～ EP5 はダブルバッファ構成となっており , アクセスしていないバッファに既にデータが書き込まれている状態で , 他方のバッファにデータが書き込まれたときにのみ DRQ をクリアされます。アクセスしていない方のバッファが空の場合は "0" を書き込んでもクリアされません (破線状態)。この場合 , 連続して DRQ による割込み処理に入ることになります。

29.4.8 NULL転送機能

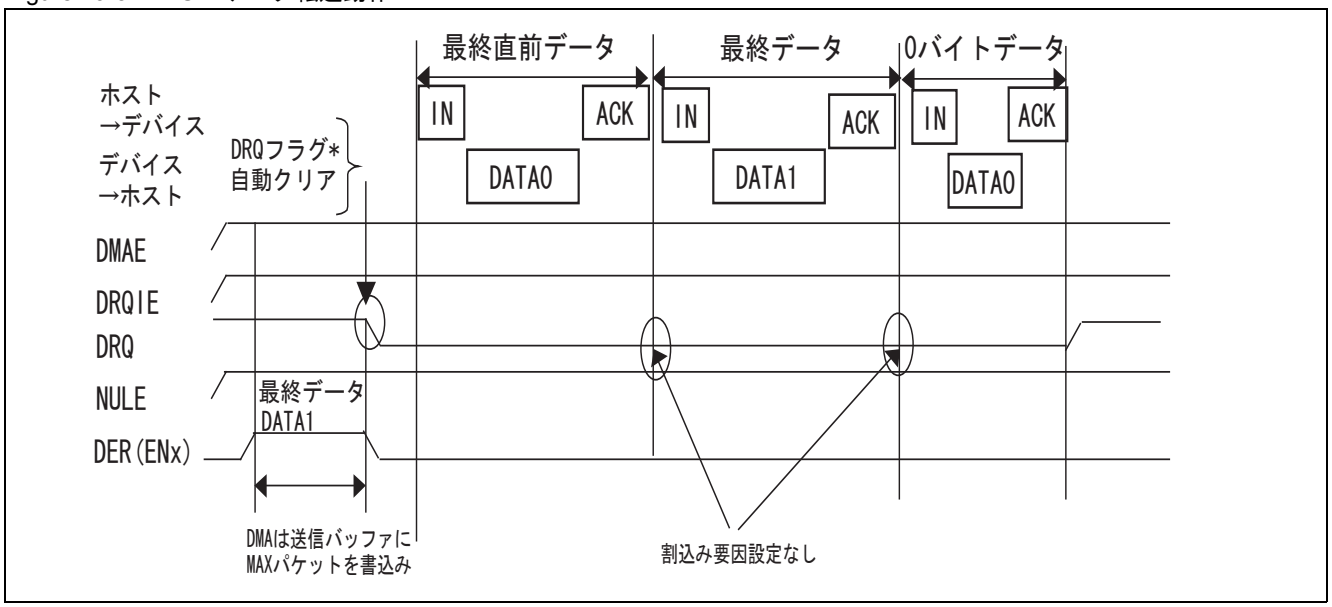
データ数自動 IN タイプ転送において, DMA の DCT レジスタに設定されたデータバイトの総数がすでに送信された場合, NULL 転送モードを許可することによりホストからの次の IN 転送要求に応じて 0 バイトのデータパケットを自動で送信することが可能です。

29.4.8.1 NULL転送モード

このモードでは , データ数自動転送モードが IN 方向の転送用に設定され (DMAE=1), PKS ビットに設定された最大サイズのデータが転送バッファに書き込まれた場合に , ホストから IN 方向の最後の転送要求が検出されたときに 0 バイトのデータ転送を自動で設定します。最後のデータ書き込み時に DMA の DCT レジスタが 0 にデクリメントされ , ホストからの IN 方向の次の転送要求が検出されたときに 0 バイトのデータが転送されます。ホストによって 0 バイトのデータが受信されるまで , すなわちホストから 0 バイトの転送のために ACK が受信されるまで , DRQ の割込みフラグはセットされません。バッファをアクセスするタイミングを次図に示します。

IN方向(デバイス → ホスト)転送のみ

Figure 29-32. NULLデータ転送動作



デバイスは DMAE, DRQIE, NULE ビットの設定が必要です。

30. USBホスト

USBホストの機能と動作について説明します。

30.1 USBホストの特長

USB ホストは, 必要最低限のホスト機能を実現し, PCが介在することなくデバイスとのデータ転送ができる機能です。

USBホストの特長

USB ホストには以下の特長があります。

- Low Speed/Full Speedデバイスの自動検出
- Full Speed転送サポート
- デバイスの接続および切断の自動検出
- USBバスへのリセット送出機能サポート
- IN/OUT/SETUP/SOFトークンのサポート
- INトークン時のハンドシェークパケット自動送出(STALLは除く)
- OUTトークン時のハンドシェークパケット自動検出
- 最大データパケット長 256バイトまでサポート
- 各種エラー(CRCエラー/トグルエラー/タイムアウト)サポート
- ウェイクアップ機能サポート

30.2 USBホストの制限事項

USBホストの制限事項を示します。

USBホストの制限事項

		USB ホスト
ハブのサポート		○ *
転送	バルク転送	○
	コントロール転送	○
	インタラプト転送	○
	アイソクロナス転送	○
転送スピード	Low Speed	×
	Full Speed	○
PRE パケットサポート		×
SOF パケットサポート		○
エラー	CRC エラー	○
	トグルエラー	○
	タイムアウト	○
	最大パケット＜受信データ	○
デバイスの接続・切断検出		○
転送スピード検出		○

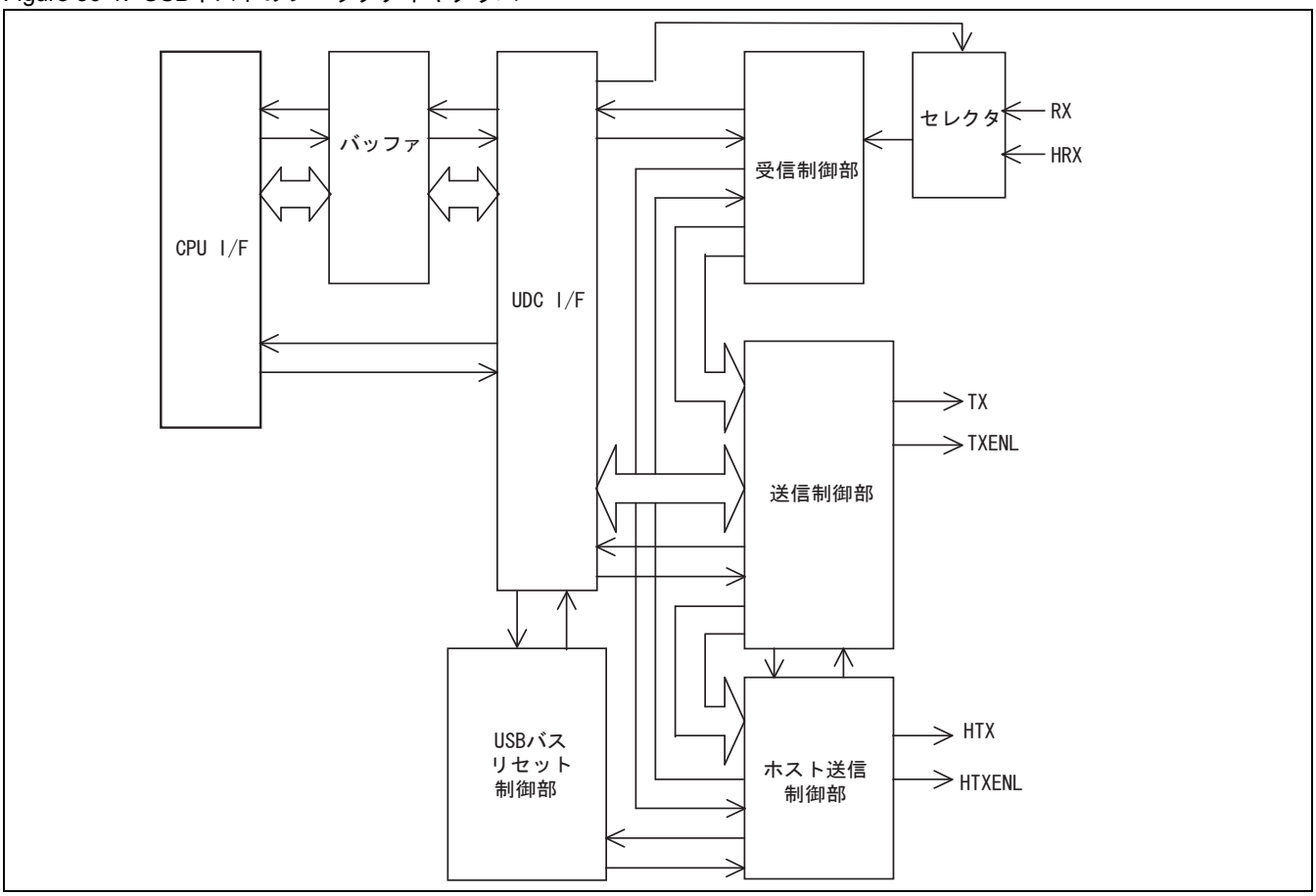
○ : サポートしている
×: サポートしていない
* : Full Speed のみ対応し , HUB は 1 段までのサポートです。

30.3 USBホストのブロックダイヤグラム

Figure 30-1に, USB ホストのブロックダイヤグラムを示します。

USBホストのブロックダイヤグラム

Figure 30-1. USBホストのブロックダイヤグラム



ブロックの説明

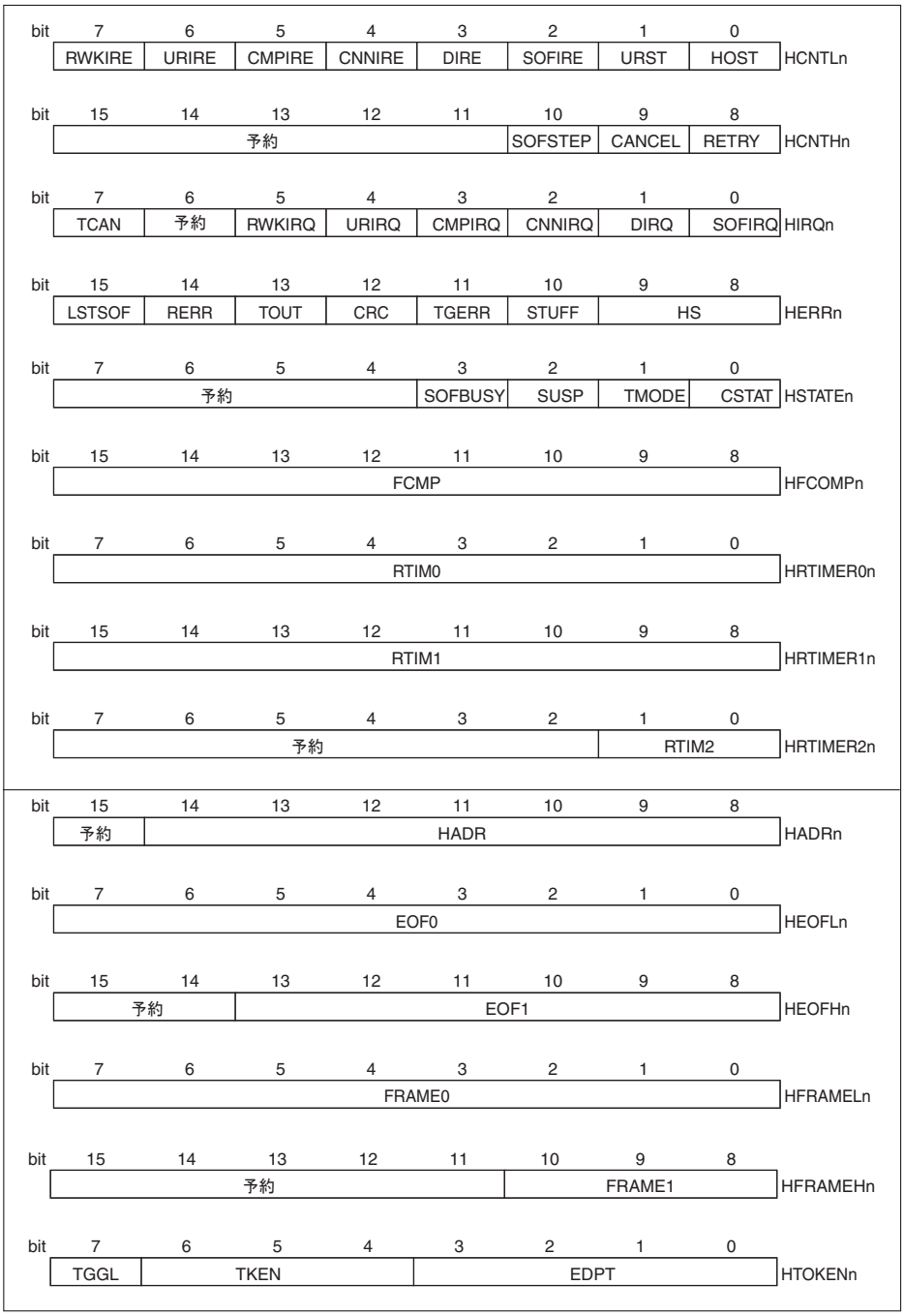
CPU I/F	: CPU とのインタフェース回路ブロック
バッファ	: バッファおよびバッファコントロール回路ブロック
USB I/F	: USB とのインタフェース回路ブロック
USB バスリセット制御部	: USB バスリセット・接続制御ブロック
受信制御部	: 受信データのシリアル・パラレル変換回路および受信制御回路ブロック
送信制御部	: USB ファンクション時の送信データのパラレル・シリアル変換回路および送信制御回路ブロック
ホスト送信制御部	: USB ホスト時の送信データのパラレル・シリアル変換回路および送信制御回路ブロック
TX	: USB ファンクションの送信データ信号
RX	: USB ファンクションの受信データ信号
TXENL	: ファンクションの送受信方向信号
HTX	: USB ホストの送信データ信号
HRX	: USB ホストの受信データ信号
HTXENL	: USB ホストの送受信方向信号

30.4 USBホストのレジスタ

USB ホストのレジスタについて説明します。

USBホストのレジスタ

Figure 30-2. USBホストのレジスタ



30.4.1 ホストコントロールレジスタ(HCNTn)

ホストコントロールレジスタ(HCNTn)は, USBの動作モードおよび割込みの設定を指定します。

ホストコントロールレジスタ下位(HCNTLn)

Figure 30-3. ホストコントロールレジスタ下位(HCNTLn)

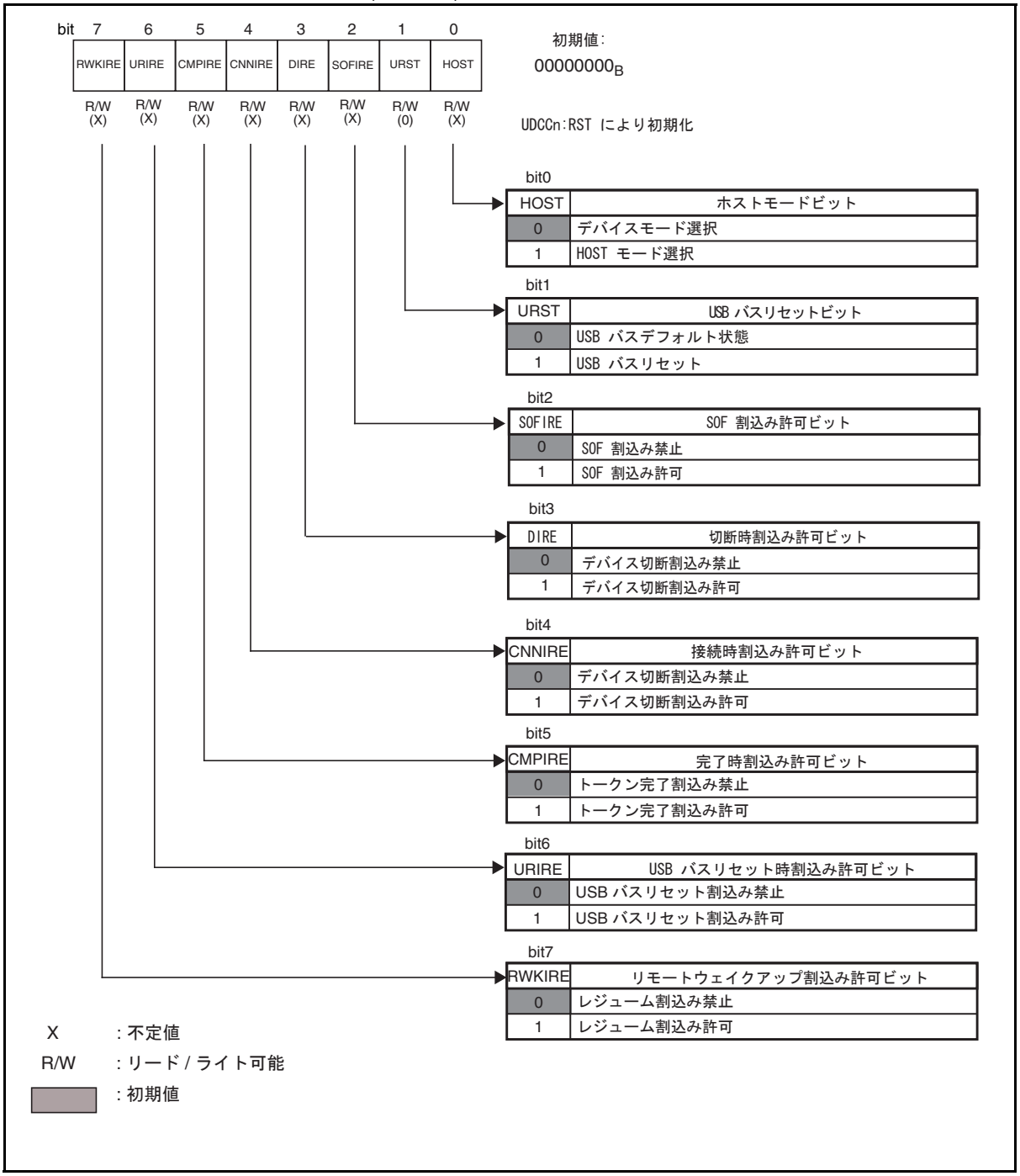
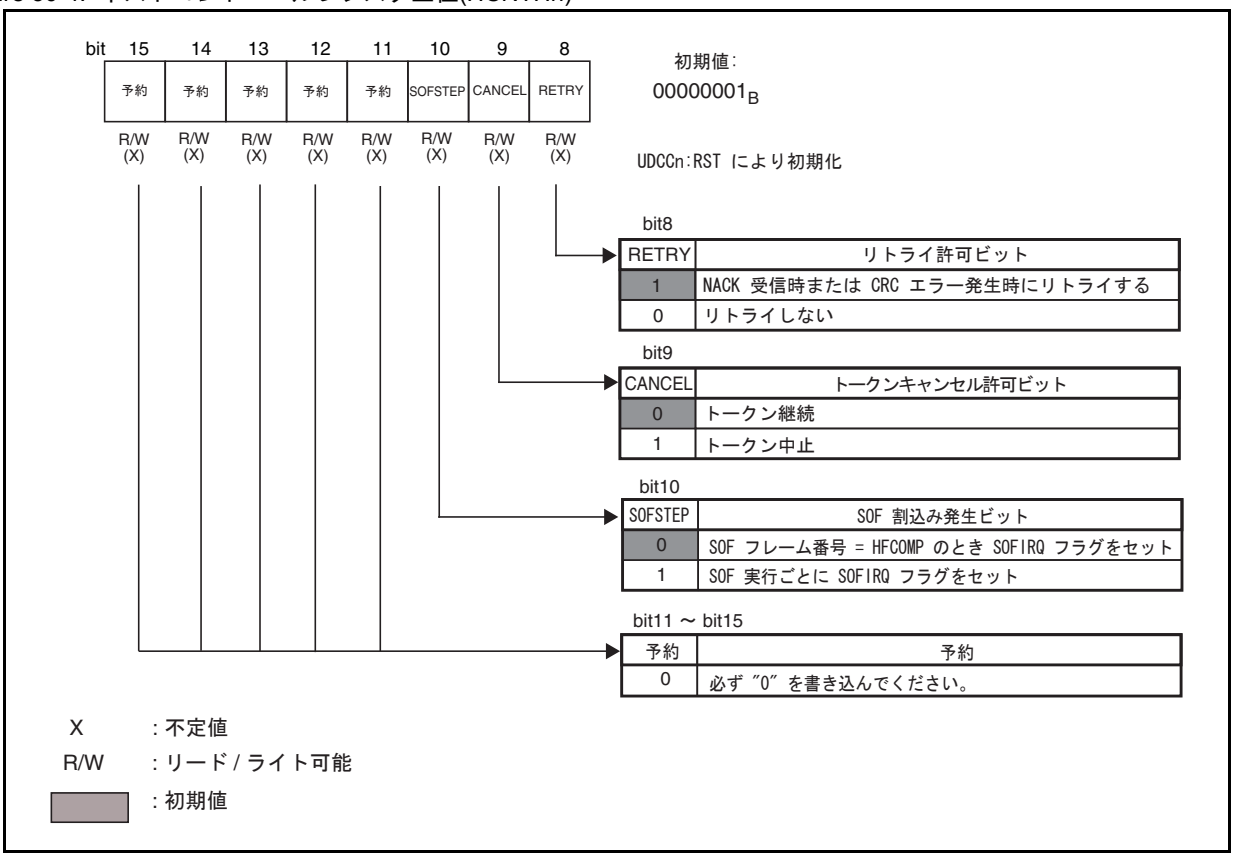


Figure 30-4. ホストコントロールレジスタ上位(HCNTHn)



HCNTnレジスタの各ビットの機能を以下に示します。

	ビット名	機能
Bit15～Bit11	予約	これらのビットには "0" を書き込んでください。
Bit10	SOFSTEP	このビットは , SOF 実行ごとに SOF による割込みが発生するかどうかを設定します ("1" にセットしたとき)。 HCNTLn:SOFIRE ビットをセットした場合 , 割込みを許可します。 SOFSTEP が "0" の場合, SOF フレーム番号の下位 8 ビットが SOF 割込み比較レジスタ (HFCOMPn) で設定した値に等しいとき, SOF 割込み要求フラグ HIRQn:SOFIRQが設定されます。SOFSTEPが"1"の場合,SOF実行ごとにSOF割込み要求フラグHIRQn:SOFIRQが設定されます。ただし,HIRQn:SOFIRQビットは HTOKENn:TKEN ビットを "001_B" に設定することによって実行される SOF トークンによって設定されません。このビットはUDC 制御レジスタ (UDCCn) の RST ビットで初期化されません。
Bit9	CANCEL	このビットはトークンを中止するために使用されます。 このビットを "1" に設定した場合 , EOF 領域でトークンを実行するよう HTOKENn:TKEN ビットを設定すると (HEOFn レジスタによって指定します), トークンが中止されます。 このビットを "0" に設定した場合 , HTOKENn:TKEN ビットで示されたトークンは中止されません。 トークンが中止されたかどうか確認するには, HIRQn:TCAN ビットを読み出してください。 UDCCn:RST はこのビットを初期化しません。
Bit8	RETRY	このビットは , トークンのリトライを許可するために使用されます。 このビットを "1" に設定した場合 , NAK が USB デバイスから受信される , あるいは各種エラーが発生すると (HERRn:RERR="1", HERRn:TOUT="1", HERRn:CRC="1", HERRn:TGERR="1", HERRn:STUFF="1"), USB ホストがトークンをリトライします。 リトライタイマ設定レジスタ (HRTIMERHn, HRTIMERMn, HRTIMERLn) で指定された期間中 , リトライが実行されます。 このビットはUDC 制御レジスタ (UDCCn) の RST ビットで初期化されません。
Bit7	RWKIRE	このビットは , リモートウェイクアップ割込みを許可するために使用されます。 このビットを "1" に設定した場合 , HIRQn:RWKIRQ ビットを "1" に設定すると , 関連する割込みを開始します。 このビットを "0" に設定した場合, HIRQn:RWKIRQ ビットを "1" に設定すると , 割込みは発生しません。UDCCn:RST はこのビットを初期化しません。
Bit6	URIRE	このビットは , USB バスリセット割込みを許可するために使用されます。 このビットを "1" に設定した場合 , HIRQn:URIRQ ビットを "1" に設定すると , 関連する割込みを開始します。 このビットを "0" に設定した場合, HIRQn:URIRQ ビットを "1" に設定すると, 割込みを発生しません。UDCCn:RST はこのビットを初期化しません。
Bit5	CMPIRE	このビットは , トークン完了割込みを許可するために使用されます。 このビットを "1" に設定した場合 , HIRQn:CMPIRQ ビットを "1" に設定すると , 関連する割込みを開始します。 このビットを "0" に設定した場合, HIRQn:CMPIRQ ビットを "1" に設定すると , 割込みは発生しません。UDCCn:RST はこのビットを初期化しません。
Bit4	CNNIRE	このビットは , デバイス接続割込みを許可するために使用されます。 このビットを "1" に設定した場合 , HIRQn:CNNIRQ ビットを "1" に設定すると , 関連する割込みを開始します。 このビットを "0" に設定した場合, HIRQn:CNNIRQ ビットを "1" に設定すると , 割込みは発生しません。UDCCn:RST はこのビットを初期化しません。

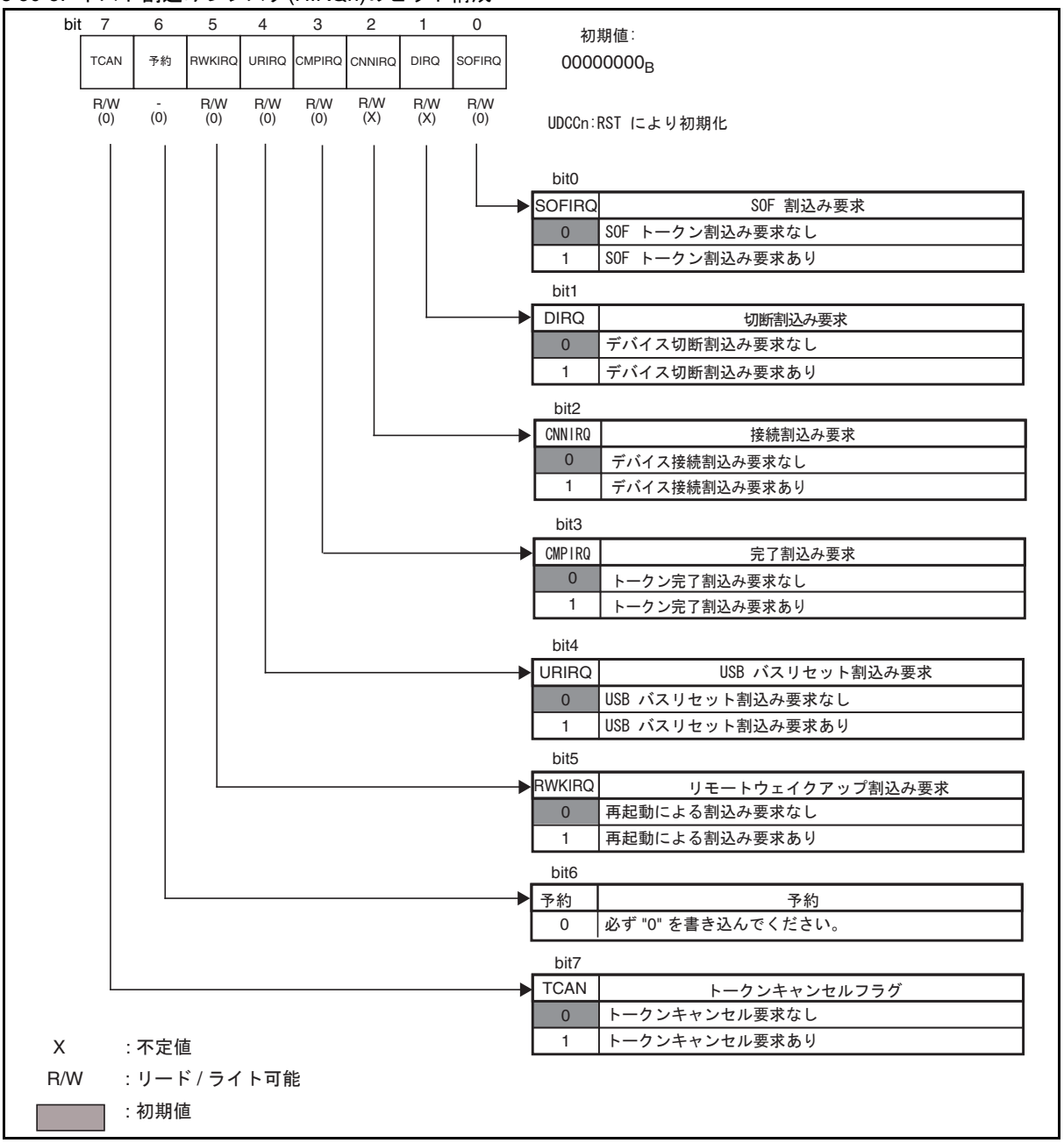
	ビット名	機能
Bit3	DIRE	このビットは , デバイス切断割込みを許可するために使用されます。 このビットを "1" に設定した場合 , HIRQn:DIRQ ビットを "1" に設定すると , 関連する割込みを開始します。 このビットを "0" に設定した場合, HIRQn:DIRQ ビットを "1" に設定すると, 割込みは発生しません。UDCCn:RST はこのビットを初期化しません。
Bit2	SOFIRE	このビットは SOF 割込みを許可するために使用されます。 このビットを "1" に設定した場合 , HIRQn:SOFIRQ ビットを "1" に設定すると , 関連する割込みを開始します。 このビットを "0" に設定した場合, HIRQn:SOFIRQ ビットを "1" に設定すると , 割込みは発生しません。UDCCn:RST はこのビットを初期化しません。
Bit1	URST	このビットに "1" を書き込むと , USB バスがリセットされます。USB バスリセットが終了するとクリアされます。 "1" を読み出すと , USB バスのリセットが継続中であることが示されます。 UDCCn:RST="1" のとき , このビットに "1" を書き込むと無視されます。 HSTATEn:SUSP ビットの設定時またはトークン実行中にこのビットを設定することは禁止です。 また, このビットの設定時に HCNTLn または HCNTHn の更新は禁止です。
Bit0	HOST	このビットは , USB ファンクションがデバイスとして機能するか , HOST として機能するかを制御します。このビットに "1" を設定するとホスト モードが選択され , "0" を設定するとファンクションモードが選択されます。このビットの更新後モードを変更するには時間がかかるため , モード変更が終了したかどうか確認するためにこのビットを読み出してください。 このビットは RST ビットを "1" に設定したとき変更する必要があります。 ファンクションモードをデバイスからホストファンクションへ変更するには , HOST ビットを更新する前に UDCCn:HCONX ビットの設定によるホストの切断を実行する必要があります。 ファンクションモードをホストモードからファンクションモードに変更する場合 , HOST ビットを更新する前に , HSTATEn:SOFBUSY ビットを "0" にクリアし , HTOKENn:TKEN ビットを "000_B" に設定し , HSTATEn:SUSP ビットを "0" にクリアしてください。 UDCCn:RST はこのビットを初期化しません。

30.4.2 ホスト割込みレジスタ(HIRQn)

ホスト割込みレジスタ(HIRQn)は,USBホストの割込み要求フラグを示します。TCANビットを除いてホストコントロールレジスタ(HCNTLn/HCNTHn)の割込み許可ビットを許可することにより割込みを発生させることができます。

ホスト割込みレジスタ(HIRQn)

Figure 30-5. ホスト割込みレジスタ(HIRQn)のビット構成



	ビット名	機能
Bit7	TCAN	トークンが HCNTHn:CANCEL ビットの設定によって中止されると , このビットは "1" に設定されます。 このビットが "0" の場合, トークンが中止されなかったことを示します。このビットは "0" を書き込むことでクリアされます。"1" 書込みは無視します。このビットを "1" に設定すると割込みは発生しません。このビットを割込みルーチンで処理しようとする場合には, SOF 割込みにより呼び出される割込みルーチンでこのビットを確認してください。このビットを更新するには, UDCCn:RST ビットを"0"に設定する必要があります。UDCCn:RST ビットが"1" のとき, このビットは初期化されます。
Bit6	予約	このビットに "0" を書き込んでください。
Bit5	RWKIRQ	このビットはレジューム動作が終了すると "1" に設定され, HCNTLn:RWKIRE ビットが "1" の場合 , 割込みが開始されます。 このビットが "0" の場合 , 何も示しません。 このビットは "0" 書込みによりクリアされます。"1" 書込みは現在の状態を保持します。このビットを更新するには UDCCn:RST ビットを "0" に設定する必要があります。UDCCn:RST ビットが "1" のとき, このビットは初期化されます。
Bit4	URIRQ	このビットは USB バスのリセット動作が終了すると "1" に設定され , HCNTLn:URIRE ビットが "1" の場合 , 割込みが開始されます。 このビットが "0" の場合 , 何も示しません。 このビットは "0" 書込みによりクリアされます。 "1" 書込みは現在の状態を保持します。 このビットを更新するには UDCCn:RST ビットを "0" に設定する必要があります。UDCCn:RST ビットが "1" のとき, このビットは初期化されます。

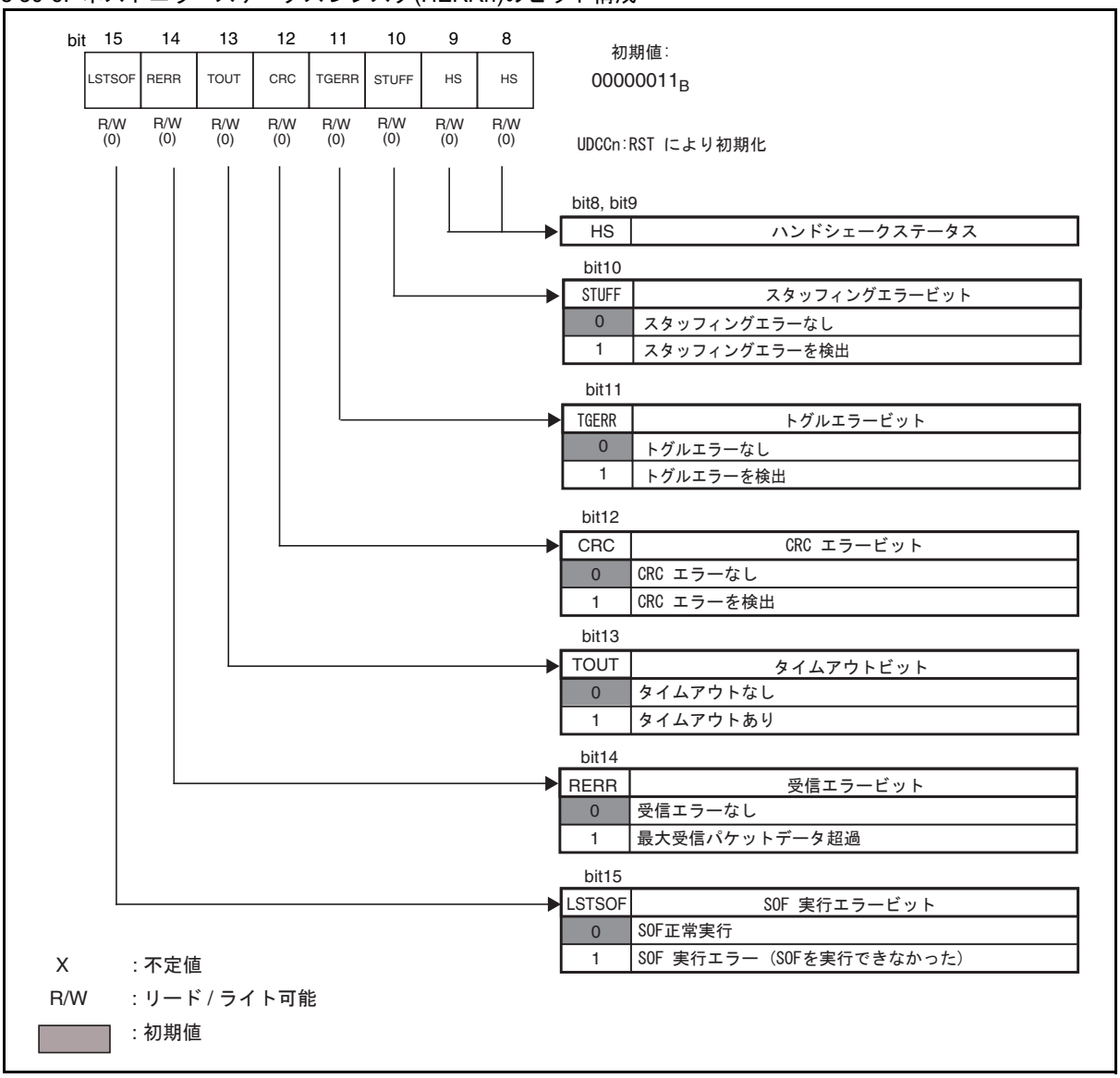
	ビット名	機能
Bit3	CMPIRQ	このビットはトークンが終了すると "1" に設定され , HCNTLn:CMPIRE ビットが "1" の場合 , 割込みが開始されます。 このビットが "0" の場合 , 何も示しません。 このビットは "0" 書込みによりクリアされます。 "1" 書込みは現在の状態を保持します。 このビットを更新するには UDCCn:RST ビットを "0" に設定する必要があります。UDCCn:RST ビットが "1" のとき , このビットは初期化されます。 HIRQn:TCAN ビットが "1" に設定されているとき, このフラグは "1" に設定されません。
Bit2	CNNIRQ	このビットはデバイスとの接続が検出されると "1" に設定され , HCNTLn:CNNIRE ビットが "1" の場合 , 割込みが開始されます。 このビットが "0" の場合 , 何も示しません。 このビットは "0" 書込みによりクリアされます。 "1" 書込みは現在の状態を保持します。 UDCCn:RST ビットにより初期化されません。 このビットは USB ホストモードだけでなく, USB デバイスモードでも有効です。
Bit1	DIRQ	このビットはデバイスの切断が検出されると "1" に設定され , HCNTLn:DIRE ビットが "1" の場合 , 割込みが開始されます。 このビットが "0" の場合 , 何も示しません。 このビットは "0" 書込みによりクリアされます。 "1" 書込みは現在の状態を保持します。 UDCCn:RST ビットにより初期化されません。 このビットは USBホストモードだけでなく, USB デバイスモードでも有効です。
Bit0	SOFIRQ	SOF 送信開始時に HCNTHn:SOFSTEP が "0" の場合 , SOF トークンの最下位 8 ビットの FRAME Number と HFCOMP が比較されます。比較結果が一致した場合には , このビットは "1" に設定されます。HCNTHn:SOFSTEP ビットが "1" の場合 , このビットは SOF 送信が開始されるごとに "1" に設定されます。 このビットが "1" に設定され , HCNTLn:SOFIRE ビットが "1" の場合 , 割込みが発生します。 このビットが "0" の場合 , 何も示しません。 このビットは "0" 書込みによりクリアされます。 "1" 書込みは現在の状態を保持します。 このビットを更新するには UDCCn:RST ビットを "0" に設定する必要があります。UDCCn:RST ビットが "1" のとき, このビットは初期化されます。

30.4.3 ホストエラーステータスレジスタ(HERRn)

ホストエラーステータスレジスタ(HERRn)は, ホストモード時のデータ送信および受信中にエラーが発生したかどうかを示すレジスタです。

ホストエラーステータスレジスタ(HERRn)

Figure 30-6. ホストエラーステータスレジスタ(HERRn)のビット構成



	ビット名	機能
Bit15	LSTSOF	このビットの読出し値が "1" である場合 , ホストモードで SOF トークンを実行しようとしたとき , 他のトークンを実行していて SOF トークンが実行できなかったことを示します。このビットの読出し値が "0" である場合 , SOF 実行エラーが検出されなかったことを示します。 このビットは "0" 書込みによりクリアされます。"1" 書込みは無視します。 このビットを更新するには, UDCCn:RST ビットを "0" に設定する必要があります。UDCCn:RST ビットが "1" のとき, このビットは初期化されます。
Bit14	RERR	このビットの読出し値が "1" である場合 , ホストモード時に設定された最大パケットサイズより大きいパケットを受信したことを示します。このビットを "1" に設定すると , HERRn:TOUT ビットも同時に "1" に設定されます。このビットの読出し値が "0" である場合 , 受信エラーが発生しなかったことを示します。 このビットは "0" 書込みによりクリアされます。"1" 書込みは無視します。 このビットを更新するには, UDCCn:RST ビットを "0" に設定する必要があります。UDCCn:RST ビットが "1" のとき, このビットは初期化されます。
Bit13	TOUT	このビットの読出し値が "1" である場合 , ホストモード時 , 一定の期間にデバイスから応答が受信されなかったことを示します。このビットの読出し値が "0" である場合 , タイムアウトが検出されたことを示します。 このビットは "0" 書込みによりクリアされます。"1" 書込みは無視します。 このビットを更新するには, UDCCn:RST ビットを "0" に設定する必要があります。UDCCn:RST ビットが "1" のとき, このビットは初期化されます。

<table>
<tr><th></th><th>ビット名</th><th>機能</th></tr>
<tr><td>Bit12</td><td>CRC</td><td>このビットの読出し値が "1" である場合 , ホストモード時 , CRC エラーが発生したことを示します。このビットを "1" に設定すると , HERRn:TOUT ビットも同時に "1" に設定されます。このビットの読出し値が "0" である場合 , CRC エラーが発生しなかったことを示します。
このビットは "0" 書込みによりクリアされます。"1" 書込みは無視します。
このビットを更新するには, UDCCn:RST ビットを "0" に設定する必要があります。UDCCn:RST ビットが "1" のとき, このビットは初期化されます。</td></tr>
<tr><td>Bit11</td><td>TGERR</td><td>このビットの読出し値が "1" である場合 , ホストモード時 , 受信されたトグルが設定に適合しなかったことを示します。このビットの読出し値が "0" である場合 , トグルエラーが発生しなかったことを示します。
このビットは "0" 書込みによりクリアされます。"1" 書込みは無視します。
このビットを更新するには, UDCCn:RST ビットを "0" に設定する必要があります。UDCCn:RST ビットが "1" のとき, このビットは初期化されます。</td></tr>
<tr><td>Bit10</td><td>STUFF</td><td>このビットの読出し値が "1" である場合 , ホストモード時 , スタッフィングエラーが発生したことを示します。このビットを "1" に設定すると , HERRn:TOUT ビットも同時に "1" に設定されます。このビットの読出し値が "0" である場合 , スタッフィングエラーが発生しなかったことを示します。
このビットは "0" 書込みによりクリアされます。"1" 書込みは無視します。
このビットを更新するには, UDCCn:RST ビットを "0" に設定する必要があります。UDCCn:RST ビットが "1" のとき, このビットは初期化されます。</td></tr>
<tr><td>Bit9, Bit8</td><td>HS</td><td>これらのビットは , 何らかのエラーでハンドシェーク動作が実行されない場合または HTOKENn:TKEN ビットの設定によって開始される SOF トークンが終了した場合には NULL を示します。
これらのビットは送受信終了時に更新します。
以下に, ハンドシェークの値を説明します。

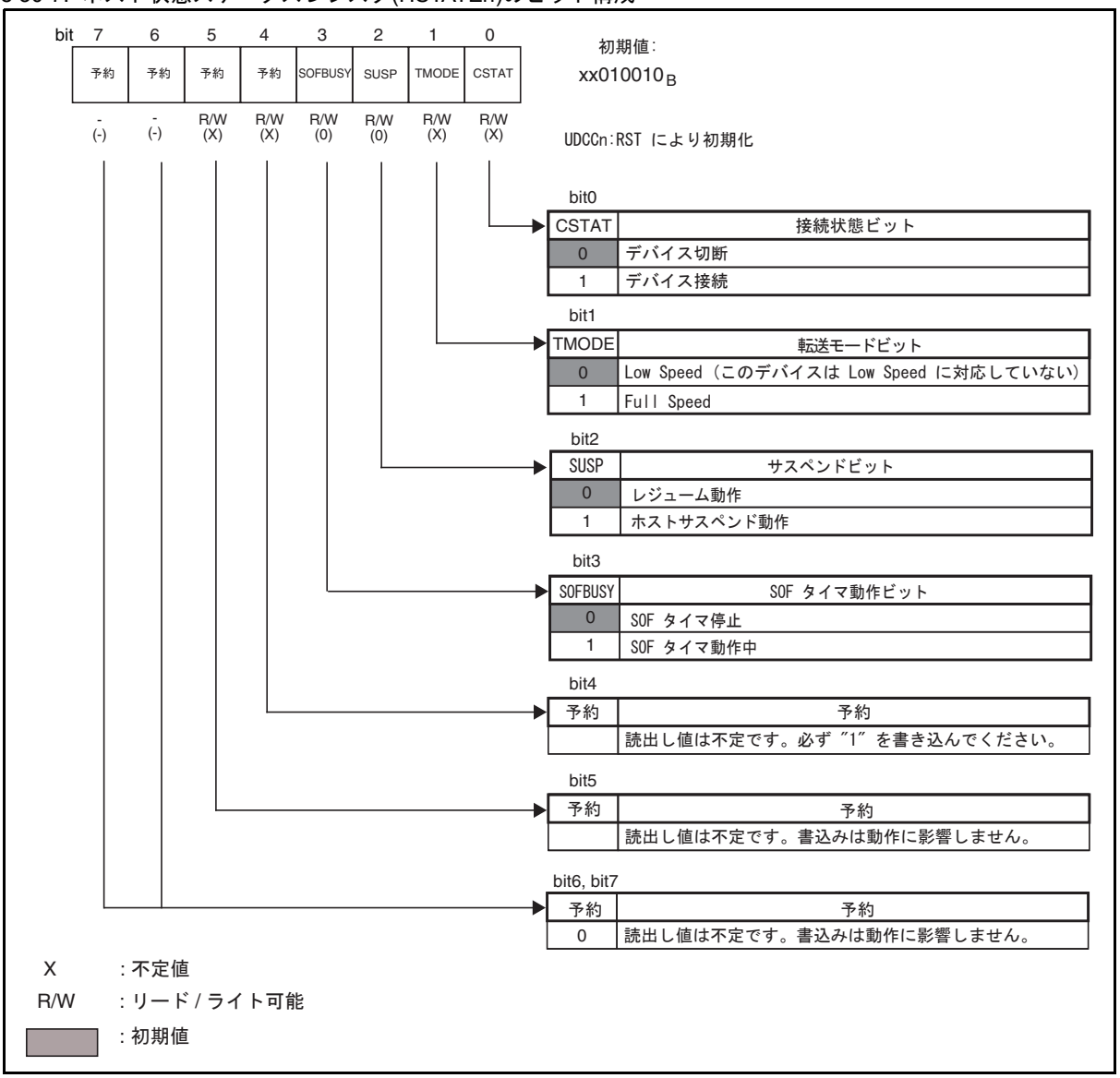
<table><tr><th>Bit9</th><th>Bit8</th><th>ハンドシェーク</th></tr><tr><td>0</td><td>0</td><td>ACK</td></tr><tr><td>0</td><td>1</td><td>NAK</td></tr><tr><td>1</td><td>0</td><td>STALL</td></tr><tr><td>1</td><td>1</td><td>NULL</td></tr></table>
このビットを更新するには, UDCCn:RST ビットを "0" に設定する必要があります。UDCCn:RST ビットが "1" のとき, このビットは初期化されます。</td></tr>
</table>

30.4.4 ホスト状態ステータスレジスタ(HSTATEn)

ホスト状態ステータスレジスタ(HSTATEn)は，デバイスの接続，転送モード等をUSB回路の状態を示すレジスタです。

ホスト状態ステータスレジスタ(HSTATEn)

Figure 30-7. ホスト状態ステータスレジスタ(HSTATEn)のビット構成



	ビット名	機能
Bit7～Bit5	予約	読出し値は不定です。書込みは動作に影響しません。
Bit4	予約	必ず "1" を書き込んでください。読出し値は不定です。
Bit3	SOFBUSY	このビットは HTOKENn レジスタの設定により SOF トークンが実行されると "1" に設定されます。このビットの読出し値が "1" である場合 , SOF タイマが動作中であることを意味します。このビットの読出し値が "0" である場合 , SOF タイマが動作していないことを意味します。 このビットに "0" を書き込むと SOF タイマが停止します。"1" 書込みは無視します。 このビットに "0" を書き込んだ後で SOF タイマを停止するには時間がかかります。SOF タイマが停止したかどうか確認するには , このビットを読み出してください。 このビットを更新するには, UDCCn:RST ビットを "0" に設定する必要があります。UDCCn:RST ビットが "1" のとき, このビットは初期化されます。
Bit2	SUSP	ホストモード時 , サスペンド状態を設定するビットです。 "1" を書き込むとサスペンド状態になります。 "1" の状態から "0" を書き込んだ場合 , または USB バスが k-state 状態に変化するとサスペンド状態を解除し , HIRQn:RWKIRQ が設定されます。 USB が動作中 (USB バスへのリセット中 , データの送受信中 , または SOF タイマの動作中), "1" を設定することは禁止です。 ホストモード時 , サスペンド状態でも USB 用クロックを停止することは禁止です。 更新するには UDCCn:RST ビットを "0" に設定する必要があります。USB ファンクションモードで "1" を設定することは禁止です。 また, ホストモードからファンクションモードに変更する際に "1" に設定されている場合は, モードを変更する前に "0" を書き込んでサスペンド状態から復帰してください (Table 30-1を参照してください)。
Bit1	TMODE	転送モードを示します ("1" は Full Speed, "0" は Low Speed) (このデバイスは Low Speed モードに対応していません)。 UDCCn:RST ビットにより初期化されません。
Bit0	CSTAT	ホストモード時 , デバイスが接続されているかを示します。このビットの読出し値が "1" のとき , USB デバイスが接続されていることを意味します。このビットの読出し値が "0" のとき , USB デバイスが接続されていないことを意味します。 UDCCn:RST ビットにより初期化されません。

Table 30-1. サスペンド設定

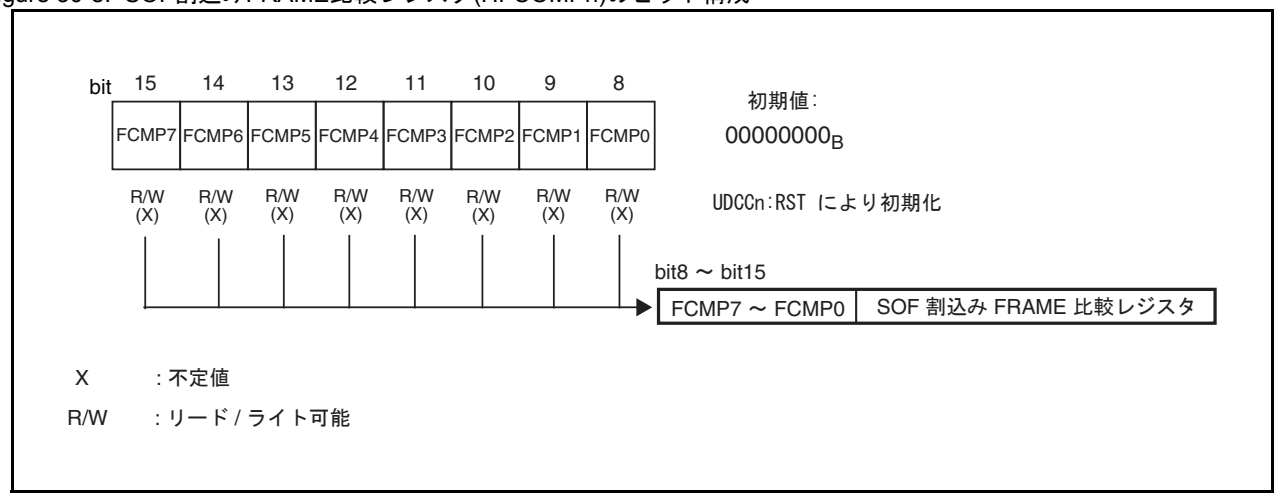
SUSP	動作モード
"1"書込み	サスペンド
"1"状態で"0"書込み	レジューム
その他	状態保持

30.4.5 SOF割込みFRAME比較レジスタ(HFCOMPn)

SOF割込みFRAME比較レジスタ(HFCOMPn)はSOFトークン時のFRAMENumberの下位8ビットと比較するデータを設定するレジスタです。

SOF割込みFRAME比較レジスタ(HFCOMPn)

Figure 30-8. SOF割込みFRAME比較レジスタ(HFCOMPn)のビット構成



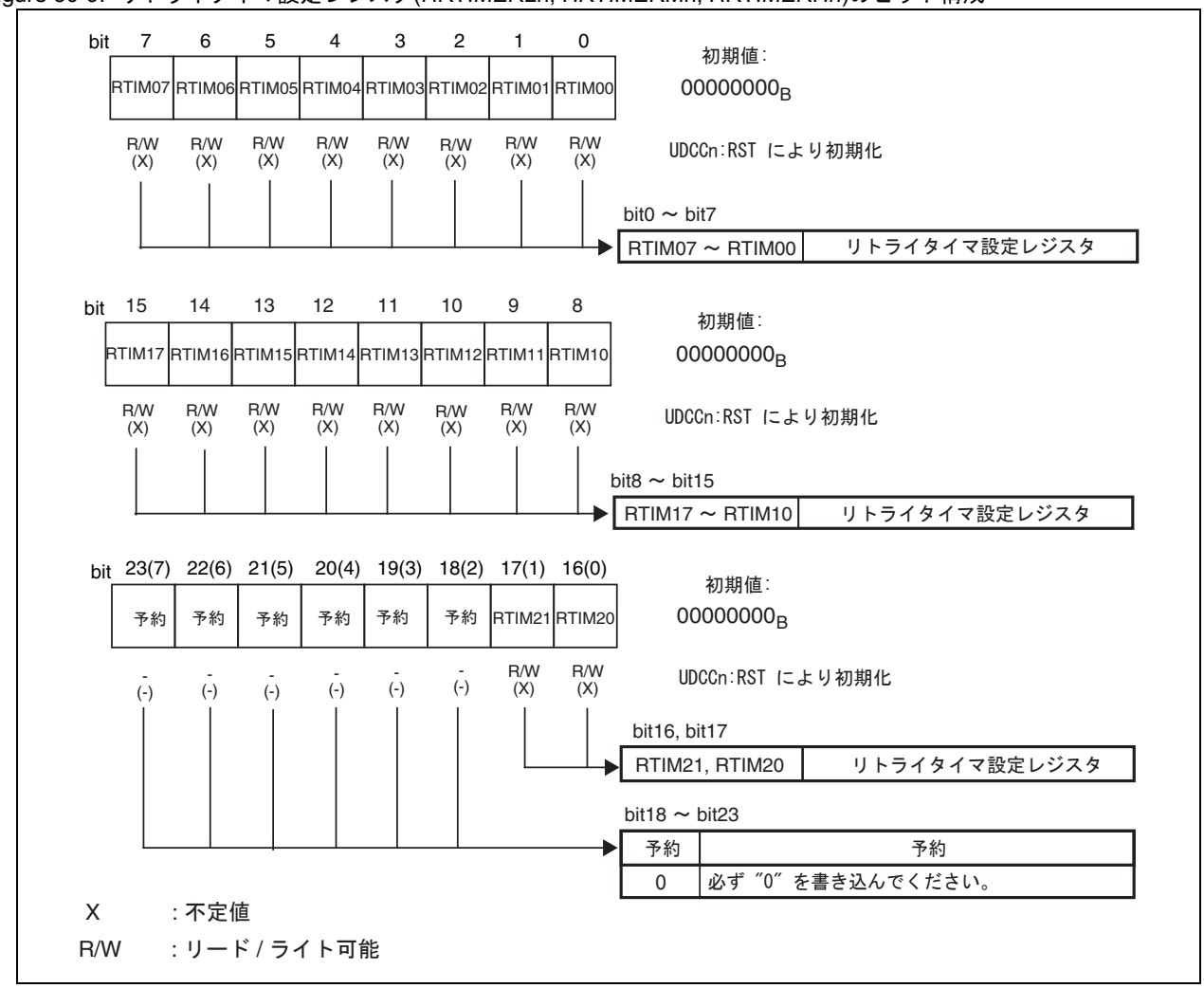
	ビット名	機能
Bit15～Bit8	FCMP7～FCMP0	SOF トークン時の Frame Number の下位 8 ビットと比較する値を設定するレジスタです。UDCCn:RST ビットにより初期化されません。 SOF 送信開始時に HCNTHn:SOFSTEP が "0" の場合 , SOF トークンの最下位 8 ビットの FRAME Number と HFCOMP が比較されます。比較結果が一致した場合には , HIRQn:SOFIRQ が設定されます。この場合 , HCNTLn:SOFIRE ビットが "1" の場合 , 割込みが発生します。 HCNTHn:SOFSTEP が "1" の場合 , このレジスタは無視されます。 UDCCn:RST ビットにより初期化されません。

30.4.6 リトライタイマ設定レジスタ(HRTIMERLn, HRTIMERMn, HRTIMERHn)

リトライタイマ設定レジスタ(HRTIMERLn,HRTIMERMn,HRTIMERHn)は,トークンのリトライ時間を設定するレジスタです。

リトライタイマ設定レジスタ(HRTIMERLn, HRTIMERMn, HRTIMERHn)

Figure 30-9. リトライタイマ設定レジスタ(HRTIMERLn, HRTIMERMn, HRTIMERHn)のビット構成



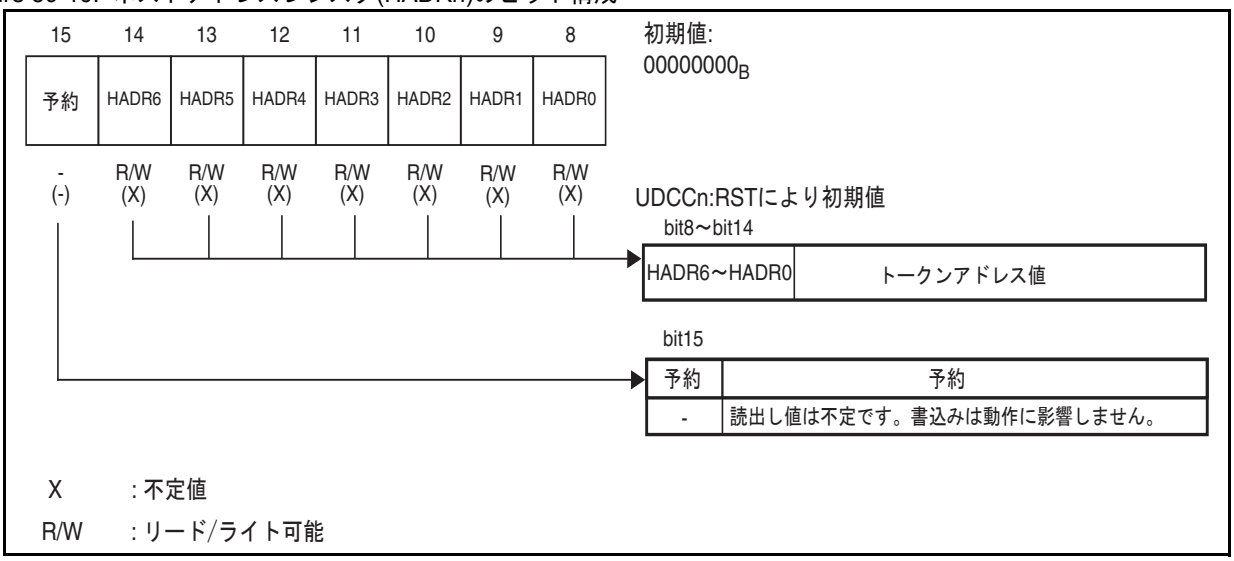
	ビット名	機能
Bit17～Bit0	RTIM21～RTIM20 RTIM17～RTIM10 RTIM07～RTIM00	NACK 受信後に トークンをリトライする時間を設定します。 HCNTHn:RETRY ビットが "1" のとき , トークンの実行が開始されるとリトライタイマが起動され , 転送クロックレート (Full Speed の場合 , 12 MHz) でタイマ値が 0 になるまで 1 デクリメントされます。 リトライタイマが "0" になると , 現在のトークンの実行後 , リトライ動作が終了します。 トークンのリトライが EOF 領域で発生した場合には , SOF の実行が終了するまでリトライタイマは停止します。SOF の実行終了後 , 停止したタイマ値から 1 デクリメントが再開します。 UDCCn:RST ビットにより初期化されません。

30.4.7 ホストアドレスレジスタ(HADRn)

ホストアドレスレジスタ(HADRn)は, トークンを送信する際のアドレスフィールドに使用されるレジスタです。

ホストアドレスレジスタ(HADRn)

Figure 30-10. ホストアドレスレジスタ(HADRn)のビット構成



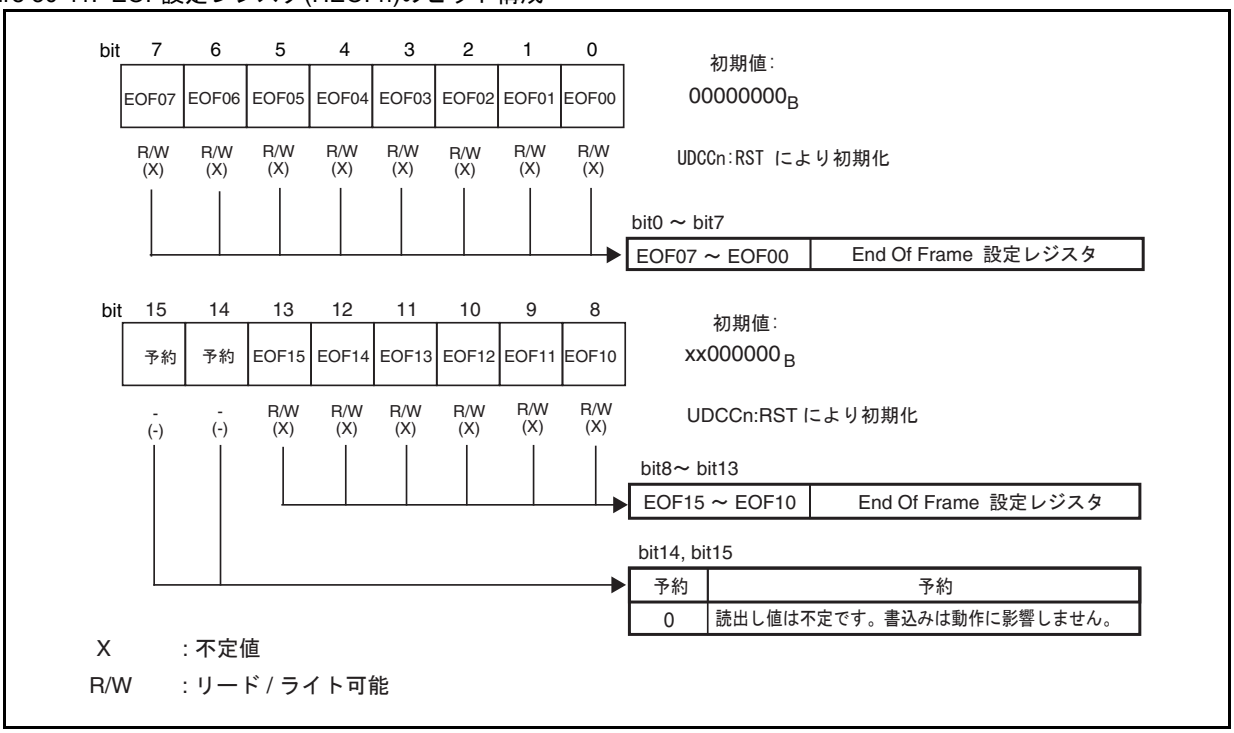
	ビット名	機能
Bit15	予約	読出し値は不定です。書込みは動作に影響しません。
Bit14～Bit8	HADR6～HADR0	このレジスタにトークンアドレスを設定します。UDCCn:RSTビットにより初期化されません。このレジスタはUDCCn:RSTビットにより初期化されません。

30.4.8 EOF設定レジスタ(HEOFn)

EOF設定レジスタ(HEOFn)は, SOFトークン実行前のトークン禁止期間を設定するレジスタです。

EOF設定レジスタ(HEOFn)

Figure 30-11. EOF設定レジスタ(HEOFn)のビット構成



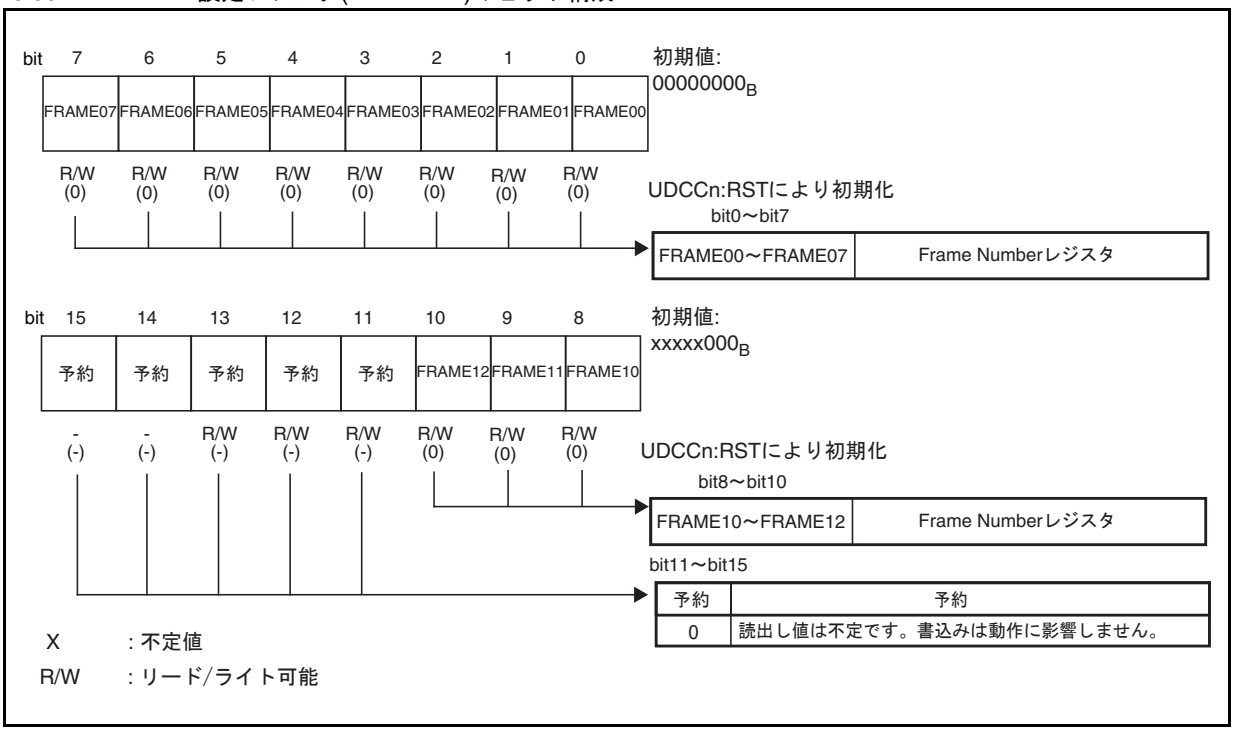
	ビット名	機能
Bit15～Bit14	予約	読出し値は不定です。書込みは動作に影響しません。
Bit13～Bit0	EOF15～EOF10 EOF07～EOF00	EOF レジスタは , SOF トークン実行前のトークン禁止期間を設定するレジスタです。SOF タイマに保存された値が HEOF レジスタに保存された値より小さいときに IN トークン , OUT トークン , SETUP トークンのいずれか実行要求があった場合は , SOF トークンの実行後のみ要求されたトークンを実行します。これは , SOF トークンと他のトークンが同時に実行されることを避けるためです。 UDCCn:RST ビットを "1" に設定しても初期化されません。 HEOF レジスタの時間単位は , 1 ビット時間です。1 パケット長より長いマージンを持って設定してください。 設定例) 最大 PKT=64 バイト , Full Speed の場合 , (Token_length + packet_length + header + CRC)×7/6 + Turn_around_time = (34bit + 546bit)×7/6 + 36bit = 712.7 bit となり , $(2C9)_H$ を設定します。

30.4.9 FRAME設定レジスタ(HFRAMEn)

FRAME設定レジスタ(HFRAMEn)は, SOFトークン時のFRAME Numberを設定するレジスタです。HTOKENn:TKENビットをSOF転送(TKEN[2:0]=100_B) にセットすると, その後1 msごとに自動的にSOFが送信されます。

FRAME設定レジスタ(HFRAMEn)

Figure 30-12. FRAME設定レジスタ(HFRAMEn)のビット構成



	ビット名	機能
Bit15～Bit11	予約	読出し値は不定です。書込みは動作に影響しません。
Bit10～Bit0	FRAME00～FRAME07 FRAME10～FRAME12	SOFトークンのFrameNumberを設定します。HTOKENn:TKNENビットにSOFを設定する前に(TKNEN[2:0]=100_B) Frame Numberを設定してください。また, HSTATEn:SOFBUSYビットが"1"およびSOFトークン実行中の書込みは禁止です。更新はUDCCn:RSTビットを"0"に設定してから行ってください。UDCCn:RSTビットを"1"に設定するとこのビットは初期化されます。

HTOKENn:TKNEN ビットに SOF 起動をセットすると SOF タイマが起動され , その後 1ms ごとに自動的に SOF が送信されます。

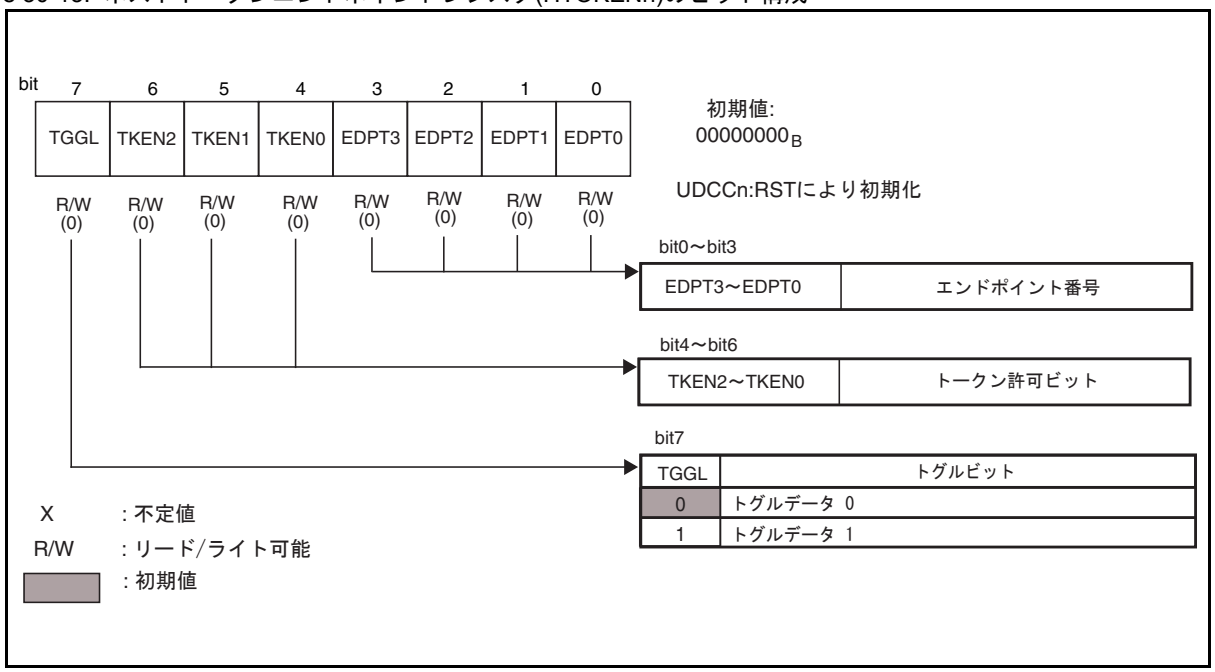
SOF が終了するごとに FRAME 設定レジスタは自動的に +1 されます。

30.4.10 ホストトークンエンドポイントレジスタ(HTOKENn)

ホストトークンエンドポイントレジスタ(HTOKENn)は，トグル，エンドポイント，トークンを設定するレジスタです。

ホストトークンエンドポイントレジスタ(HTOKENn)

Figure 30-13. ホストトークンエンドポイントレジスタ(HTOKENn)のビット構成



	ビット名	機能
Bit7	TGGL	データのトグルを設定します。送信時は TGGL ビットに従ってトグルデータを送出します。 受信時は受信したトグルデータと TGGL ビットが示すトグルデータを比較し , エラー検出に使用します。 更新は UDCCn:RST ビットが "0" に設定され, TKEN ビットが "000_B" に設定してから行ってください。
Bit6～Bit4	TKEN2～TKEN0	設定に従ってトークンを送出します。動作終了後 , これらのビットは "000_B" に変更され , HIRQn:CMPIRQ ビットは "1" に設定されます。HCNTLn:CMPIRE ビットが設定された場合 , 割込みが発生します。 TKNEN ビット更新時 , UDCCn:RST ビットを "0" にする必要があります。これらのビットは UDCCn:RST ビットが "1" のとき初期化され , 動作モードはホストモードになります。 トークンによる割込みが発生して再度トークンを発行する場合には USB 転送クロック (12 MHz) で 3 サイクル以上待ってから TKNEN ビットに書き込んでください。 切断状態 (HSTATEn:CSTAT = "0") では , TKNEN ビットに書き込んでもトークンは送信されません。SOF トークンがこれらのビット (100_B) に設定されている場合 , TGGL ビットと ENDPT ビットの設定は無視されます。 HSTATEn:SOFBUSYビットが設定されている場合,これらのビットに"100_B"を書き込むことは禁止です (トークンの設定については, Table 30-2 を参照してください)。
Bit3～Bit0	EDPT3～EDPT0	デバイスへの送受信するエンドポイントを設定します。 レジスタを更新するには, UDCCn:RST ビットを "0" に設定する必要があります。

Table 30-2. トークン設定

Bit6	Bit5	Bit4	動作
0	0	0	トークンを送出しない
0	0	1	SETUPトークンを送出
0	1	0	INトークンを送出
0	1	1	OUTトークンを送出
1	0	0	SOFトークンを送出
1	0	1	アイソクロナス転送入力
1	1	0	アイソクロナス転送出力

30.5 USBホストの動作

USB ホストの動作について説明します。

30.5.1 デバイスの接続

外部のUSB装置が接続されたことをソフトウェアで検出する方法を示します。

USBホスト機能の設定

USB ファンクションをホストとして動作させるためには , HCNTLn:HOST ビットを "1" に設定します。

外部USB装置が非接続状態, 接続の状態

外部 USB 装置が接続されていないときはプルダウン抵抗により端子 UDP と UDM ともに "L" になっています。このとき , HSTATEn:CSTAT ビットは "0", TMODE ビットは不定となります。外部 USB 装置が接続されると , HSTATEn:CSTAT ビットは "1" になります。

外部USB装置の接続検出

外部 USB 装置が接続されたことを検出するためには , HIRQn:CNNIRQ ビットを "1" に設定します。HCNTLn:CNNIRE ビットが "1" となり , デバイス接続割込みが発生します。本割込みをクリアする場合には , ホスト割込みレジスタ (HIRQ) の CNNIRQ ビットをクリアしてください。割込みではなくポーリングでデバイスの接続を検出する場合は , HCNTLn:CNNIRE ビットを "0" にし , HIRQn:CNNIRQ が "1" になることを確認するようにプログラムを作成してください。

相手先USB装置の転送速度の取得とクロック選択

HSTATEn:TMODEn ビットは接続された USB 装置の転送速度に関する情報を保持します。

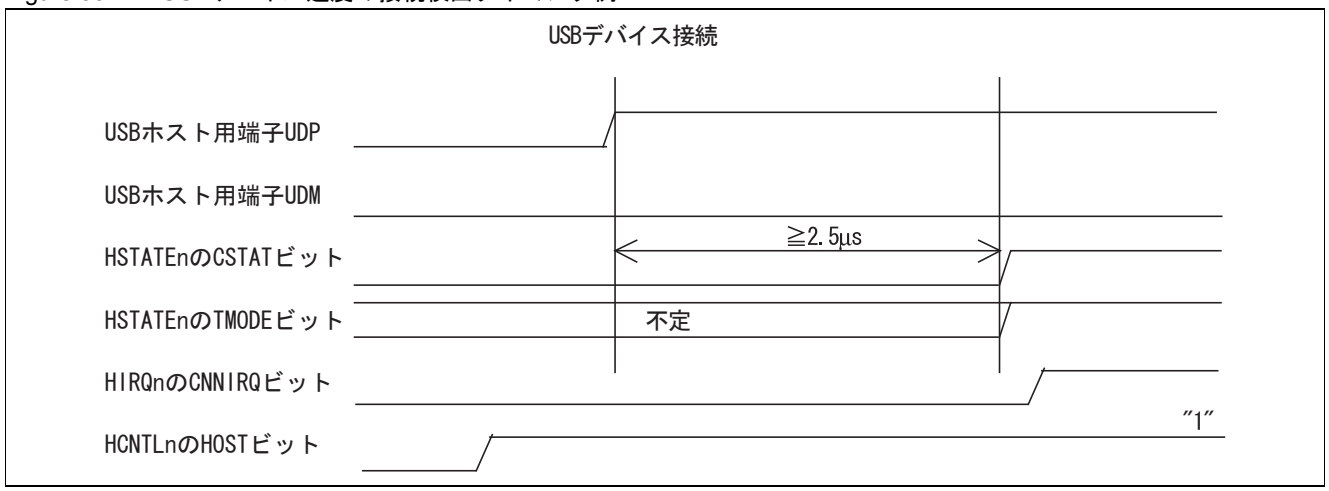
・接続先が Full Speed 対応装置 → TMODE="1"

・接続先が Low Speed 対応装置 → TMODE="0"

＜注意事項＞

このデバイスでは、Low Speed モードはサポートされません。

Figure 30-14. USBデバイス速度の接続検出タイミング例



＜注意事項＞

外部の USB 装置が接続されてから 2.5μs 後 HSTATEn:CSTAT ビットが "1" になります。

HSTATEn:TMODE ビット , HSTATEn:CSTAT ビットは HCNTLn:HOST ビットの設定にかかわらず更新されます。

30.5.2 USBバスのリセット

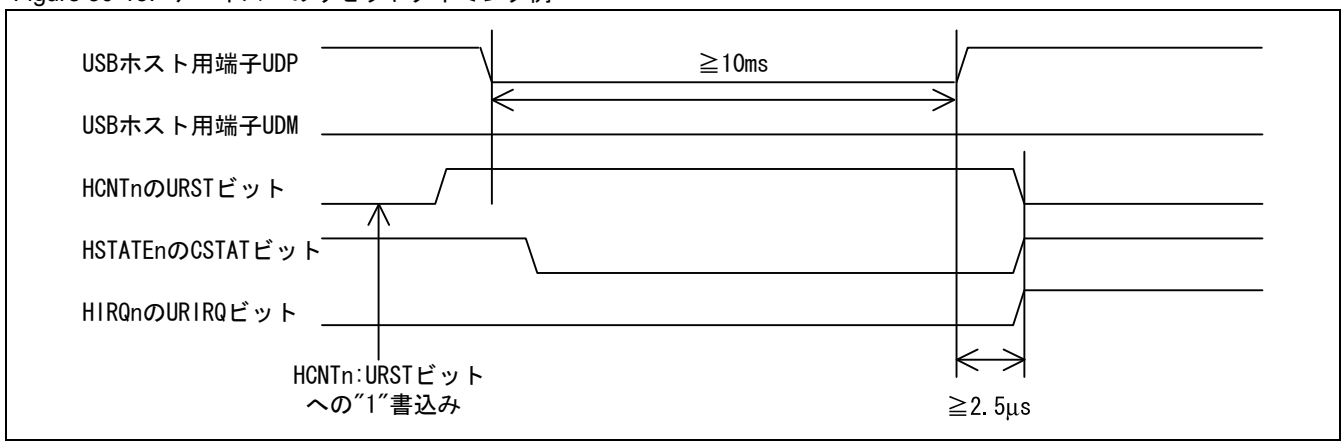
ホストモード時, HCNTLn:URSTビットに"1"を設定すると, 10 ms以上のSE0を送出し, USBバスをリセットします。USBバスのリセットが完了するとURSTビットを"0"にし, HCNTLn:URIREビットが許可されていれば割込みを発生してHIRQn:URIRQビットが"1"になります。割込みをクリアする場合にはHIRQn:URIRQビットをクリアしてください。

USBバスリセット前後の注意点

USB バスのリセットについて以下の点に注意してください。

1. USB バスへのリセット前にデバイスが接続されていることを HSTATEn:CSTAT ビットが "1" になっていることで確認。
2. USB バスのリセットを実行すると HSTATEn:CSTAT ビットは "0" になり , 切断状態になります。そのとき , HIRQn:DIRQ ビットは "1" になりません。
3. USB バスへのリセットが終了後 , HSTATEn:TMODE ビットで接続されたデバイスが Full Speed デバイスになっていることを確認してください。

Figure 30-15. デバイスへのリセットタイミング例



30.5.3 トークンパケット

INトークン, OUTトークン, SETUPトークンのいずれかを実行する場合, HADR n レジスタ, EP1Cnレジスタ (またはEP2Cnレジスタ)のDIRビット, および EP1Cnレジスタ(またはEP2Cnレジスタ)のPKSビットを設定後, ホストトークンレジスタ(HTOKENn)に必要データを設定するとトークンパケットから開始します。SOFトークン時は, FRAME設定レジスタ(HFRAME), EOF設定レジスタ(HEOF)設定後,ホストトークンエンドポイントレジスタ(HTOKEN)に必要データを設定してください。各種レジスタ(HADRn,EP1Cn,EP2Cn,HFRAMEn,HEOFn)に変更がない場合には, 設定は不要です。

トークンパケットの設定

ホストモード時 , エンドポイント 1 は送信バッファ , エンドポイント 2 は受信バッファとして使用されます。

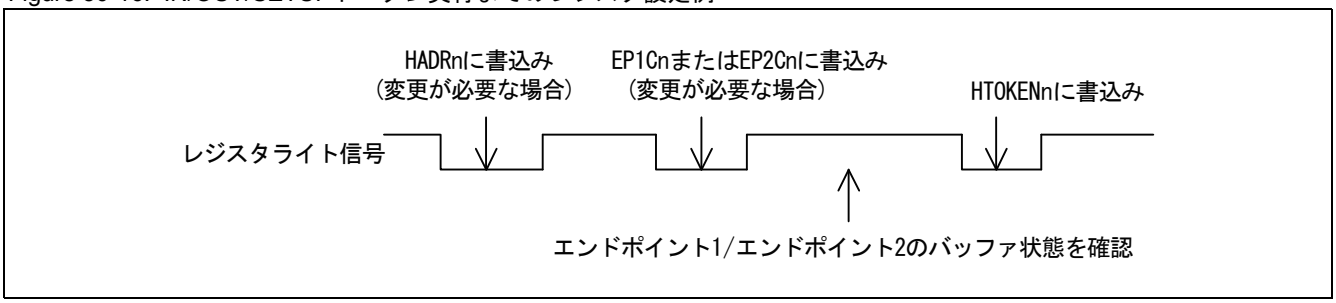
IN トークン , OUT トークン , SETUP トークンの場合 , 目的のアドレスをホストアドレスレジスタ (HADRn) にセットし , 1 パケットの最大転送可能バイト数を EP1Cn:PKS ビット (または EP2Cn:PKS ビット) にセットし , トークン転送の方向を EP1Cn:DIR ビット (または EP2Cn:DIR ビット) にセットします。EP1Cn:DIR が "1" に設定されている場合は , エンドポイント 1 のバッファが OUT 方向の転送に (送信バッファとして) 使用され , エンドポイント 2 のバッファが IN 方向の転送に (受信バッファとして) 使用されます。このとき , EP2Cn:DIR ビットに "0" をセットしてください (EP1Cn:DIR ビットと EP2Cn:DIR ビットは相互に補完する必要があります。すなわち , 一方が IN バッファの場合 , 他方を OUT バッファにする必要があります)。同様に , EP1Cn:DIR ビットが "0" に設定されている場合は , エンドポイント 1 のバッファが IN 方向の転送に使用され , エンドポイント 2 のバッファが OUT 方向の転送に使用されます。このとき , EP2Cn:DIR ビットに "1" をセットしてください。

トークンを実行するには , 以下の順序で設定を行ってください。

1) EP1Cn と EP2Cn の DIR ビットに方向を指定してください。
2) 目的のエンドポイント m (m=1 または 2) が OUT 転送用の場合, エンドポイント m のバッファに送信データを書き込み , EPmSn:DRQ ビットを "0" にクリアしてください。方向が IN 転送である場合 , EPmSn:DRQ ビットを読み出し , "0" であることを確認してください。
3) ホストトークンエンドポイントレジスタ (HTOKENn) に目的のエンドポイント , トークンおよびトグルデータを設定してください。

USB回路はその設定されたトークンに従ってSync, トークン, アドレス, エンドポイント, CRC5, EOPの順でトークンパケットを送出します (Sync, CRC5, EOP は自動)。1 パケット終了後 , HIRQn の CMPIRQ ビットが "1" となり , HTOKENn:TKEN を $(000)_B$ にします (「30.5.7 SOF 割込み」参照)。そのとき , HCNTLn:CMPIRE ビットが "1" であれば , 割込みを発生します。割込みをクリアするには HIRQn:CMPIRQ ビットをクリアしてください。

Figure 30-16. IN/OUT/SETUPトークン実行までのレジスタ設定例



SOF トークンの場合 , EOF 設定レジスタ (HEOF) および FRAME 設定レジスタ (HFRAME) に EOF 時間および FRAME 番号をセットし ,HTOKENn:TKEN ビットに SOF トークンのコードをライトすると Sync, SOF トークン , FRAME 番号, CRC5, EOP を送出し, HSTATEn:SOFBUSY ビットを設定し, HFRAMEn を+1 します。HIRQ:CMPIPQ も設定され (HCNTn:CMPIRE ビットで許可されれば割込みが発生)HTOKENn:TKEN ビットをクリアします。その後 , 自動的に発生する SOF が使用される場合には , HIRQn:CMPIRQ による割込みは発生しません。

HSTATEn:SOFBUSY ビットが "1" の間 , SOF は 1 ms ごとに自動的に送出されます。HSTATEn:SOFBUSY ビットが "0" となる条件 (SOF の停止条件) は , 以下のとおりです。

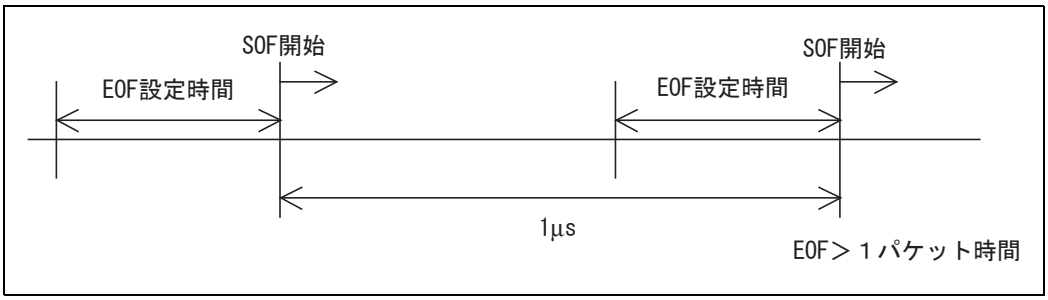
- HSTATEn:SOFBUSYビットへの"0"ライト
- USBバスへのリセット(HCNTn:URSTビットへの"1"ライト)
- HSTATEn:SUSPビットへの"1"ライト
- デバイスの切断(HSTATEn:CSTATビットが"0"の場合)

ホストモードからデバイスモードへ切り換える場合には , まず HSTATEn:SOFBUSY ビットを "0" にクリアし , HSTATEn:SOFBUSY の読出し値が "0", HTOKENn:TKEN ビットの読出し値が "000_B", HSTATEn:SUSP の読出し値が "0" であることを確認し , HCNTLn:HOST ビットを "0" にクリアしてから切り換えてください。

HSTATEn:SOFBUSY ビットを再び "1" にするには SOF トークンをもう 1 度実行する必要があります。

EOF設定レジスタは，SOFとほかのトークンの同時実行を防止するために，EOF設定時間からSOF開始時間までの間にHTOKENn:TKENビットにライトした場合,設定したトークンをウェイトされ,SOF実行後にウェイトしているトークンを実行します。EOF設定レジスタの時間単位は1ビット時間で，　例えばEOF設定レジスタに(10_H)と設定した場合，Full Speedモードでは16×1/12 MHz = 1333.3 nsという時間が必要になります。EOF設定時間の設定を1パケット時間より短く設定すると，　他のトークン実行中にSOF実行が重なる場合があります。この場合，HERRn:LSTSOFビットが"1"にセットされSOFは実行されません。HERRn:LSTSOFに"1"が設定された場合には，EOF設定レジスタのデータを大きくする必要があります(「30.4.8 EOF設定レジスタ(HEOFn)」参照)。

Figure 30-17. SOFタイミング



30.5.4 データパケット

トークンパケット送出後, データパケットの送信の場合, HTOKENn:TGGLビットに従ってトグルデータを送出し, (EP1Cn:DIRビットによって選択されたエンドポイント1またはエンドポイント2の送信バッファから) バッファデータ, CRC16データ, EOPを送出します。データパケットの受信の場合, HTOKENn:TGGLビットと受信したトグルデータを比較し, 一致した場合には受信データをEP1Cn:DIRビットによってエンドポイント1またはエンドポイント2のバッファに格納し, CRC16のエラーを検査します。

30.5.4.1 データパケット

トークンパケット送出後 , データパケットを以下の手順で処理します。

送信の場合

(1) Sync の自動送出

(2) HTOKENn:TGGL ビットが "0" であれば DATA0, HTOKENn:TGGL ビットが "1" であれば DATA1 を送出

(3) EP1Cn:DIR ビットが "1" であればエンドポイント 1 のバッファ ,
EP1Cn:DIR ビットが "0" であればエンドポイント 2 のバッファを選択し , 送信データをすべて送出

(4) HCNTLn:HOST ビットが "1" のときには , EP1Cn:DIR ビットには EP2Cn:DIR ビットの反転させたデータをセットしてください。例えば , EP1Cn:DIR ビットが "0" であれば , EP2Cn:DIR ビットには "1" をセットします。

(5) CRC 16 ビットを送出

(6) EOP 2 ビットを送出

(7) J State 1 ビットを送出

受信の場合

(1) Sync の受信

(2) トグルデータを受信し , HTOKENn:TGGL ビットと比較します。

(3) トグルデータを比較した結果 , 一致すれば EP1C:DIR ビットが "0" であれば受信データはエンドポイント 1 に格納され , "0" でなければ、エンドポイント 2 のバッファに格納されます。

(4) EOF を受信したとき , CRC 16 ビットを検査します。

(5) HCNTLn:HOST ビットが "1" のときには , EP1Cn:DIR ビットには EP2Cn:DIR ビットの反転させたデータをセットしてください。例えば , EP1Cn:DIR ビットが "0" であれば , EP2Cn:DIR ビットには "1" をセットします。

30.5.5 ハンドシェークパケット

ハンドシェークパケットにより, USBデバイスに自分の状態を通知します。

ハンドシェークパケット

ハンドシェークパケットは, 受信側から送信側に対して送信の状態を通知するために受信側が使用するパケットです。受信側はデータ受信に対し正常に受信できる状態にあるかどうかを送信側に通知するために , ACK, NAK, STALL 状態のいずれかをハンドシェークパケットで送出します。そのとき , USB 回路はハンドシェークパケットを受信した場合は , 受信したハンドシェークパケットを HERRn:HS ビットにセットします。ハンドシェークパケットを送信した場合は送信したハンドシェークパケットを HERRn:HS ビットにセットします。

30.5.6 リトライ機能

リトライ機能は再送信による転送エラー処理を許可します。

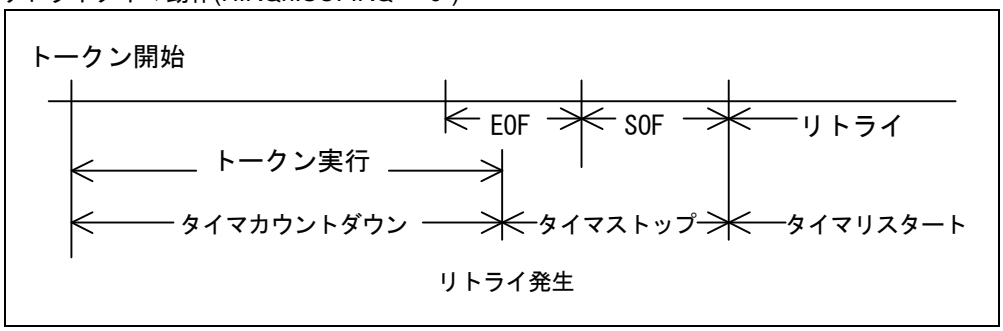
リトライ機能

パケット転送終了時 , NAK および CRC エラー等のエラーが発生し , リトライ機能が許可されていれば（すなわち HCNTHn:RETRY が "1"）, リトライタイマ設定レジスタ（HRTIMERLn, HRTIMERMn, HRTIMERHn）で設定された時間 , リトライし続けます。STALL およびデバイスの切断以外のエラーが発生した場合 (HERR:HS=01, HERR:RERR=1, HERR:TOUT=1, HERR:CRC=1, HERR:STUFF=1), HCNTHn:RETRY ビットが "1" であれば , そのトークンをリトライします。リトライの終了条件は , 以下のとおりです。

- HCNTHn:RETRYビットの"0"クリア
- リトライタイマ設定レジスタの0検出
- SOFによる割込みの発生(HIRQn:SOFIRQ=1)
- ACKの検出
- デバイスが切断されたことの検出

リトライタイマはトークン開始時に起動され , 1 ビットの転送クロックにてカウントダウンされますが , EOF 領域内でリトライが発生した場合にはカウントは停止します。HIRQn:SOFIRQ ビットが "0" で SOF トークンが終了した場合には , リトライタイマは , 停止したカウンタ値からリスタートします。SOF トークンが終了すると , リトライタイマが 0 となり , パケットが終了した時点で HIRQn:CMPIRQ ビットは "1" に設定されます。

Figure 30-18. リトライタイマ動作(HIRQn:SOFIRQ = "0")



リトライが終了したとき , その終了パケットの終了情報が各レジスタにセットされます。

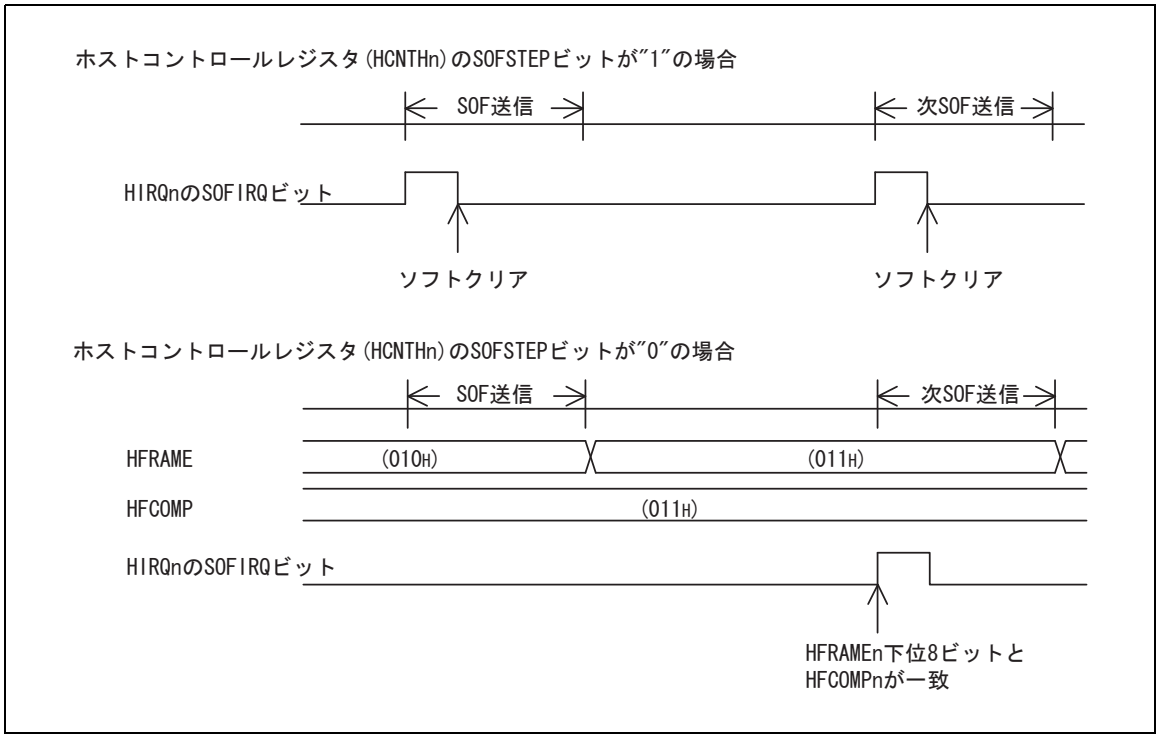
30.5.7 SOF割込み

SOFの割込み処理について説明します。

SOF割込み

HCNTHn:SOFSTEP ビットおよび SOF 割込み FRAME 比較レジスタ (HFCOMPn) の設定によって SOF 開始時 , HCNTLn:SOFIRQ ビットに "1" をセットします。HCNTLn: SOFIRE ビットに "1" を設定した場合 , HIRQn:SOFIRQ ビットが "1" にセットされると割込みが発生します。SOFSTEP ビットが "0" に設定されていて , SOF Frame 番号の下位 8 ビットが SOF 割込み FRAME 比較レジスタ (HFCOMPn) に設定された値と等しい場合 , SOF 割込み要求フラグ HIRQn:SOFIRQ ビットに "1" をセットします。SOFSTEP が "1" の場合 , SOF の実行ごとに SOF 割込み要求フラグ HIRQn:SOFIRQ ビットに "1" をセットします。ただし , HIRQn:SOFIRQ ビットは最初の SOF トークンにより "1" にセットされません。(SOF トークンは HTOKEN:TKEN ビットにより設定されます)。HCNTLn:SOFIRE ビットが "1" にセットされると , 割込みが許可されます。ホストトークンエンドポイントレジスタ (HTOKENn) の設定による最初の SOF 実行では , HIRQn:SOFIRQ ビットが "1" に設定されません。

Figure 30-19. SOF割込み



HCNTHn:CANCEL ビットを設定すると , EOF 領域内で HTOKENn に SOF トークン以外のトークンを設定し , 次の SOF で HIRQn:SOFIRQ ビットが "1" になる場合 , そのトークンは実行せずに HTOKENn:TKEN ビットは (000_B) に設定されます。そのとき , HIRQn:CMPIRQ ビットは "1" になりません。トークンがキャンセルされたことは , HIRQn:SOFIRQ ビットが "1" になったときに HIRQn:TCAN ビット (トークンキャンセル) によって知ることができます。再度 , トークンを実行したい場合には , HIRQn:TCAN フラグをクリアして , HTOKENn:TKEN ビットに実行させたいトークンを書き込んでください。

HCNTHn:CANCEL ビットを "0" にした場合には SOF 送信後 , HTOKENn レジスタに設定されたトークンを実行します。

Figure 30-20. HCNTHn:CANCELビット=1のトークンキャンセル動作例.

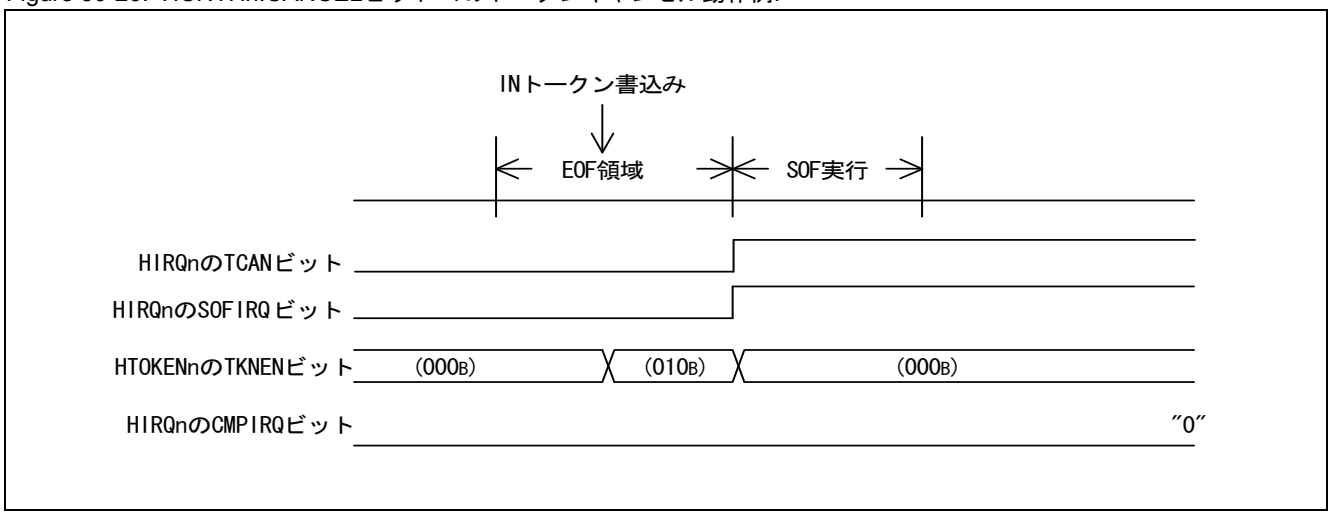
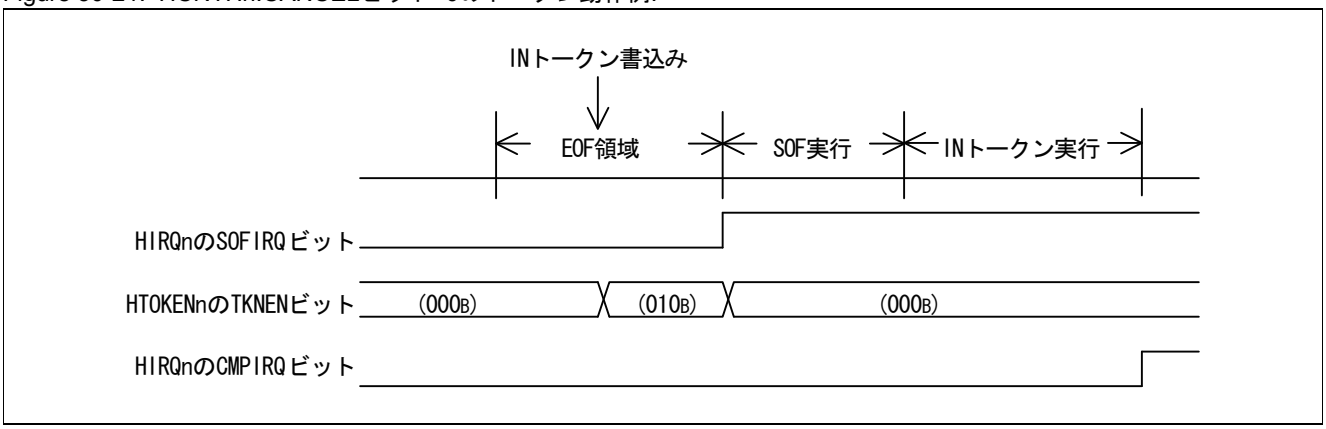


Figure 30-21. HCNTHn:CANCELビット=0のトークン動作例.



30.5.8 エラーステータス

USB ホストが検出する各種エラー情報を説明しています。

30.5.8.1 エラーステータス

スタッフィングエラー

6 ビット連続 "1" が発生した場合 , "0" のビットを 1 つ挿入することになっていますが , 7 ビット連続 "1" を検出するとスタッフィングエラーとして HERRn:STUFF ビットをセットします。STUFF ビットをクリアするには "0" を書き込んでください。STUFF ビットをクリアせずに次のトークンを実行した場合には次のトークンの終了時に更新されます。

トグルエラー

IN トークン時 , データパケットのトグルデータと HTOKENn:TGGL ビットを比較し , 一致しなかった場合 HERRn:TGERR ビットに "1" をセットします。HERRn:TGERR ビットをクリアするには , HERRn:TGERR ビットに "0" を書き込んでください。TGERR ビットをクリアせずに次のトークンを実行した場合には次のトークンの終了時に更新されます。

CRCエラー

IN トークン時 , 受信したデータを CRC の多項式 $G(X)=X^{16}+X^{15}+X^2+1$ で計算し , 剰余が $(800D_H)$ でなければ CRC エラーが発生したことになり , HERRn:CRC ビットに "1" をセットします。HERRn:CRC ビットをクリアするには , HERRn:CRC ビットに "0" を書き込んでください。CRC ビットをクリアせずに次のトークンを実行した場合には次のトークンの終了時に更新されます。

タイムアウトエラー

所定時間内にデータパケットやハンドシェークが入力されなかった , 受信データ中に SE0 が検出された , スタッフィングエラーが検出された場合等 , HERRn:TOUT ビットに "1" がセットされます。HERRn:TOUT ビットをクリアするには , HERRn:TOUT ビットに "0" を書き込んでください。TOUT ビットをクリアせずに次のトークンを実行した場合には次のトークンの終了時に更新されます。

受信エラー

受信データサイズがパケットサイズの制限を超えていると HERRn:RERR ビットに "1" がセットされます。エンドポイント 1 が使用されていると EP1Cn:PKS フィールドで設定されているパケットサイズの制限が参照され , エンドポイント 2 が使用されていると EP2Cn:PKS が参照されます。HERRn:RERR ビットをクリアするには , HERRn:RERR ビットに "0" を書き込んでください。RERR ビットをクリアせずに次のトークンを実行した場合には次のトークンの終了時に更新されます。

30.5.9 パケット終了

パケット終了処理について説明します。

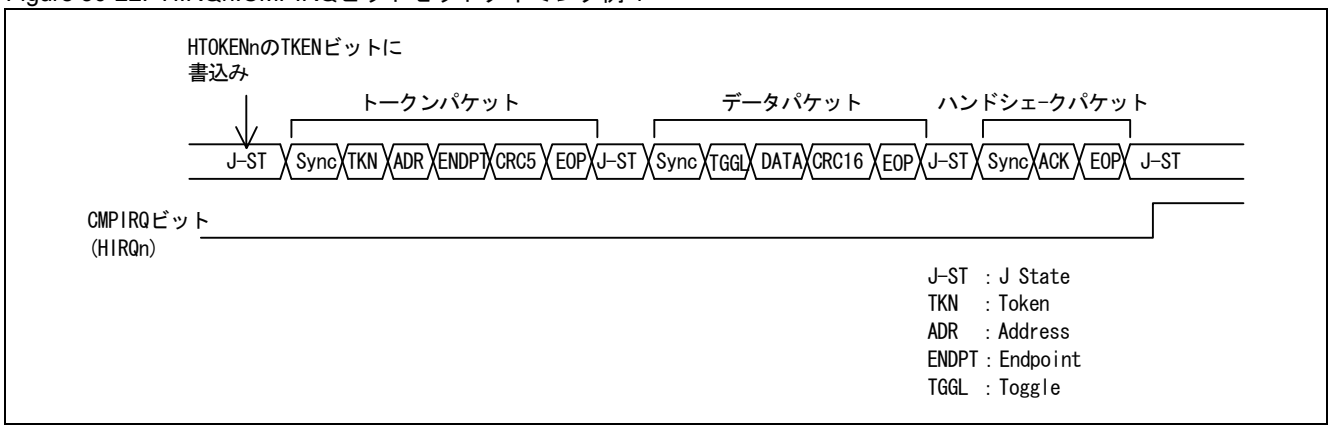
パケット終了タイミング

USB ホストにおいて 1 パケット転送終了すると (HCNTLn:CMPIRE の設定により割込みが許可されていれば)HIRQn:CMPIRQ ビットに割込みが発生します。

1 パケット終了すると以下のタイミングで割込みが発生します。

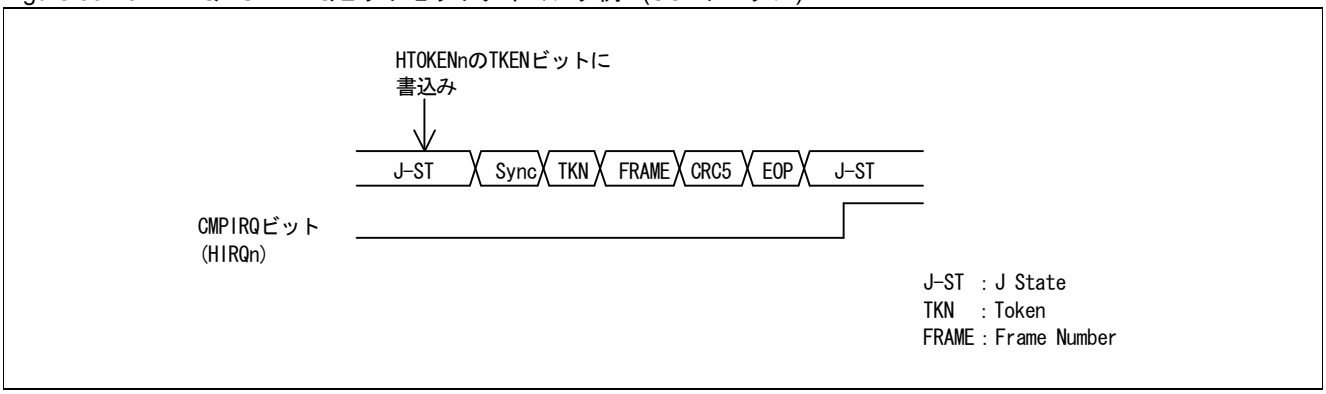
- HTOKENn:TKENビットが(001_B), (010_B), (011_B) のとき(SETUPトークン, INトークン, OUTトークン)

Figure 30-22. HIRQn:CMPIRQビットセットタイミング例 1



- HTOKENn:TKENビットが(100_B)のとき(SOFトークン)

Figure 30-23. HIRQn:CMPIRQビットセットタイミング例 2(SOFトークン)



30.5.10 サスペンド・レジューム

USB ホストにおいてサスペンドおよびレジューム動作をサポートしています。

サスペンド動作

HSTATEn:SUSP ビットに "1" を書き込むと ,

- USBバスのハイインピーダンス状態
- クロックが必要ない回路ブロックの停止

の順に実行し , USB 回路をサスペンド状態にします。サスペンド状態になった場合には , HSTATEn:SUSP ビットを "1" にします。

USB バスへのリセット中 , HSTATEn:SOFBUSY が "1" のとき , およびデータの送受信中に HSTATEn:SUSP を "1" に設定したり , USB 回路へ供給しているクロックを停止したりすることは禁止です。

レジューム動作

レジュームの動作を開始するには , 以下のいずれかの条件が成立したときです。

(1)HSTATEn:SUSP ビットに "0" を書き込む

(2)USB ホスト用端子 UDP, UDM が k-state になったことを検出

(3)デバイスが切断されたことを検出

(4)デバイスが接続されたことを検出

HIRQn:RWKIRQ ビットが "1" にセットされた後トークンの発行が可能となります。各条件による動作タイミングは以下のとおりです。

Figure 30-24. レジスタによるレジューム動作(Full Speedモード時)

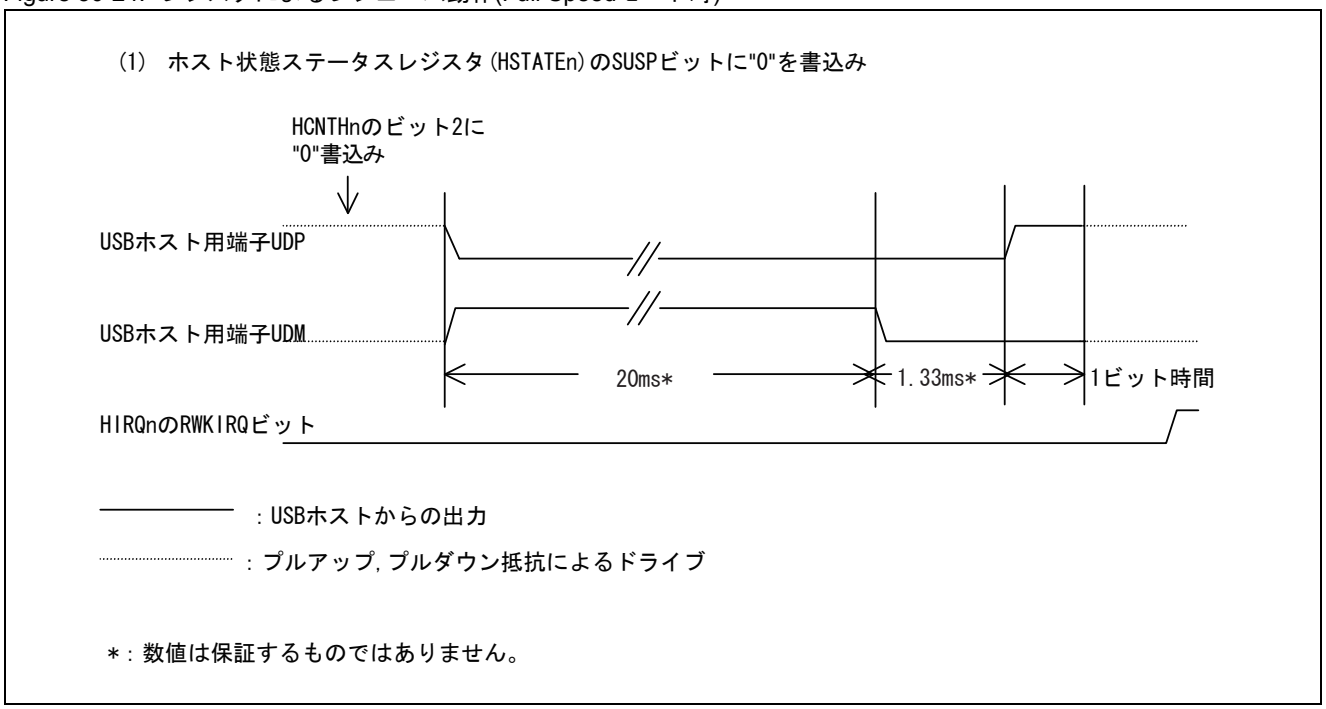


Figure 30-25. UDPとUDMのK-state検出によるレジューム動作 (Full Speedモード)

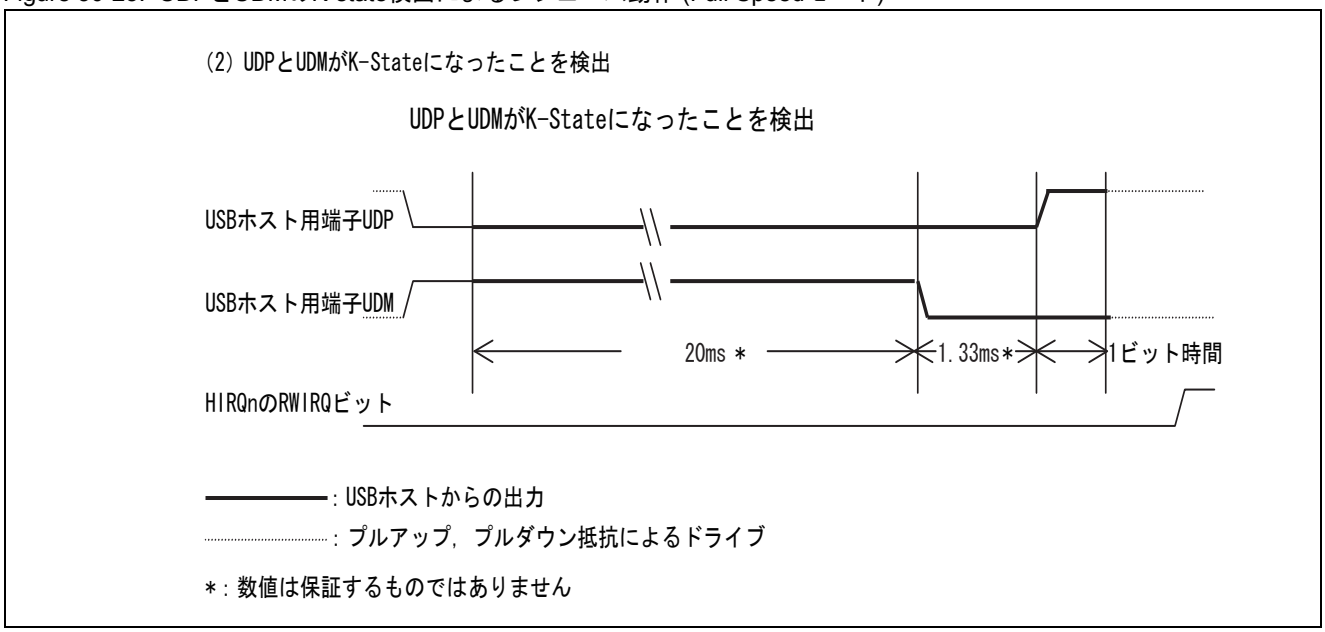


Figure 30-26. デバイス切断によるレジューム動作

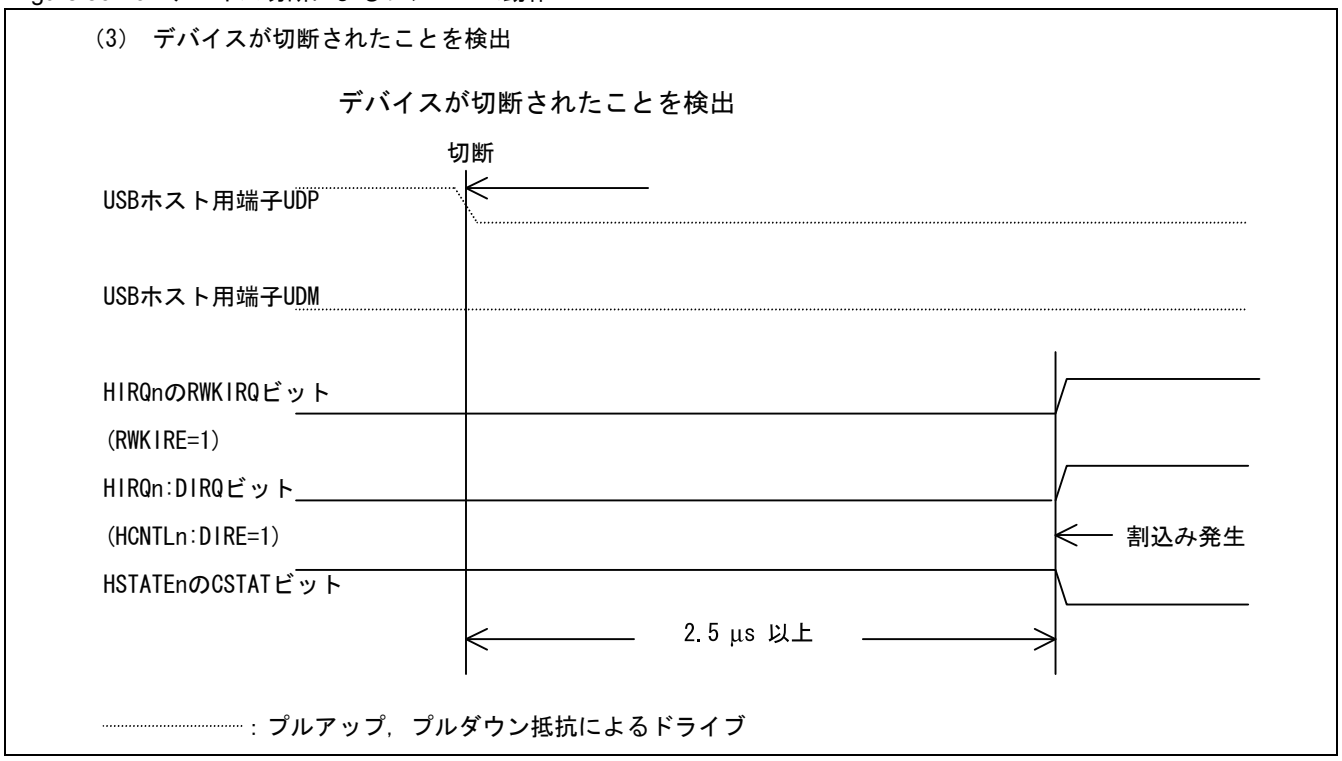
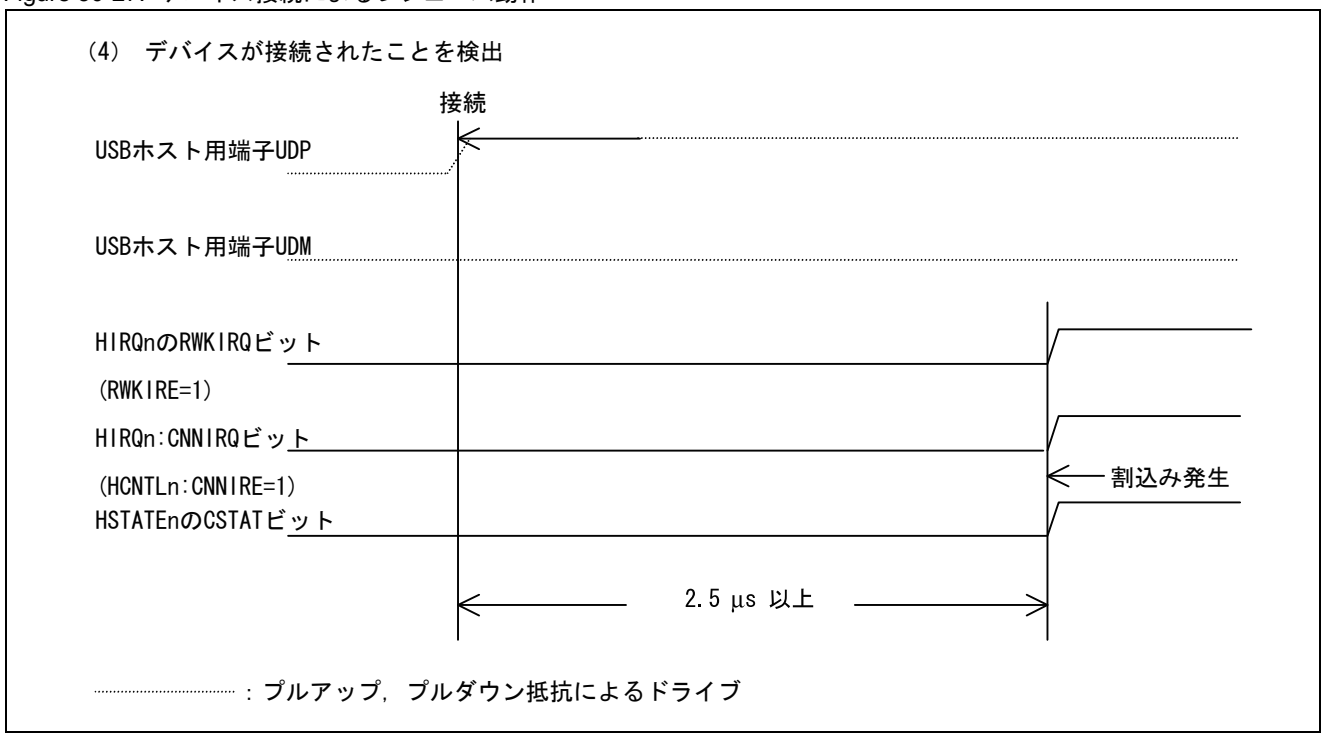


Figure 30-27. デバイス接続によるレジューム動作



30.5.11 デバイスの切断

USB ホスト用端子UDPとUDMともに"L"になると切断タイマが起動され, 2.5 ms以上"L"を検出した場合, HSTATEn:CSTATビットを"0"にします。

デバイスの切断

USB ホスト用端子 UDP と UDM が両方とも 2.5µs 以上 "L" を検出するとデバイスが切断されたと判断し , HSTATEn:CSTAT ビットが "0" となり , HIRQn:DIRQ フラグは "1" にセットされます。HCNTLn:DIRE ビットが "1" であれば割込みが発生します。その割込みをクリアしたい場合には , HIRQn:DIRQ フラグをクリアしてください。

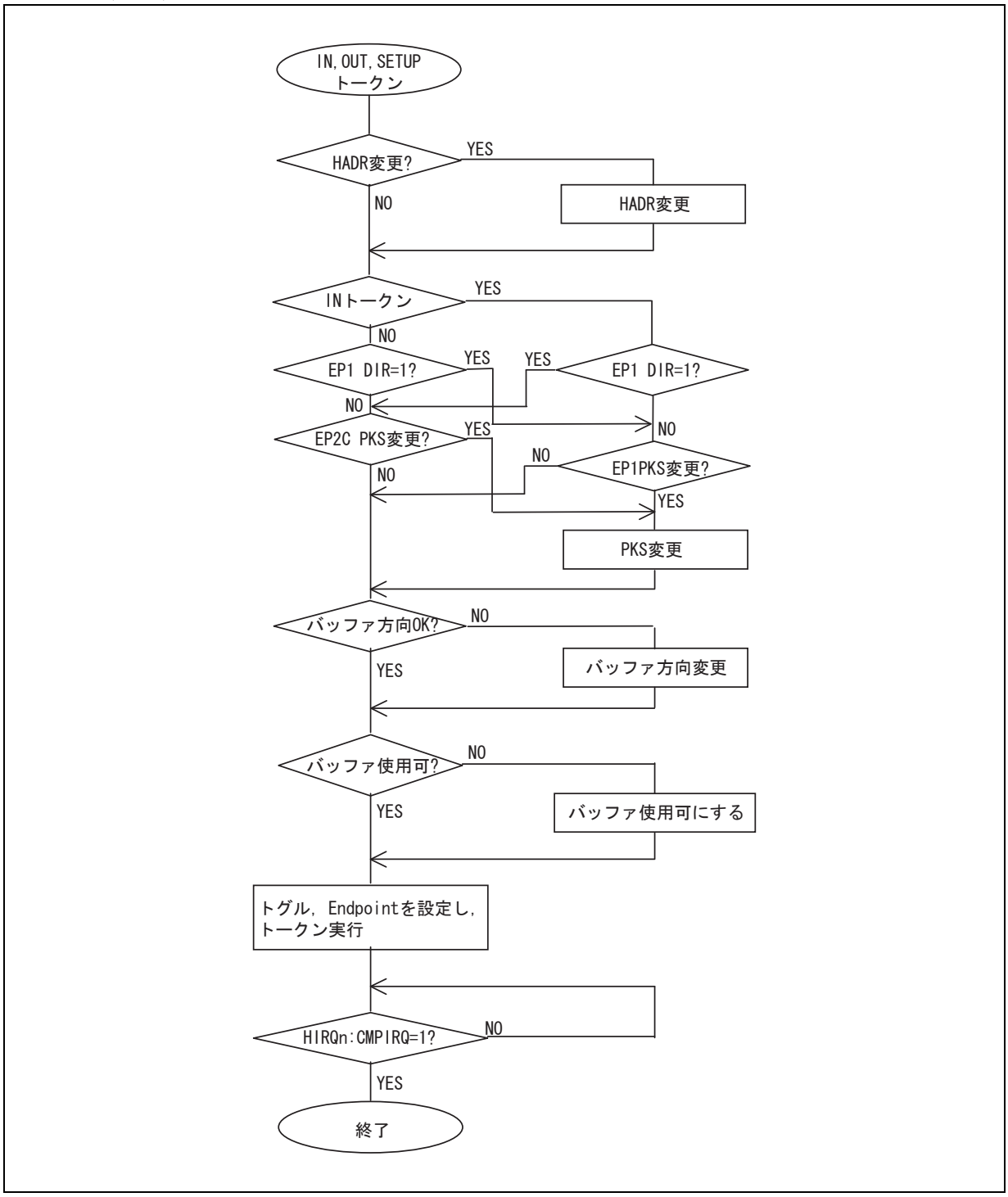
USB バスへのリセットを実行すると切断されたと判断し , HSTATEn:CSTAT ビットを "0" にしますが , HIRQn:DIRQ ビットは "1" になりません。

30.6 トークンフローチャート

USB ホストの各トークンのフローチャートは以下のとおりです。

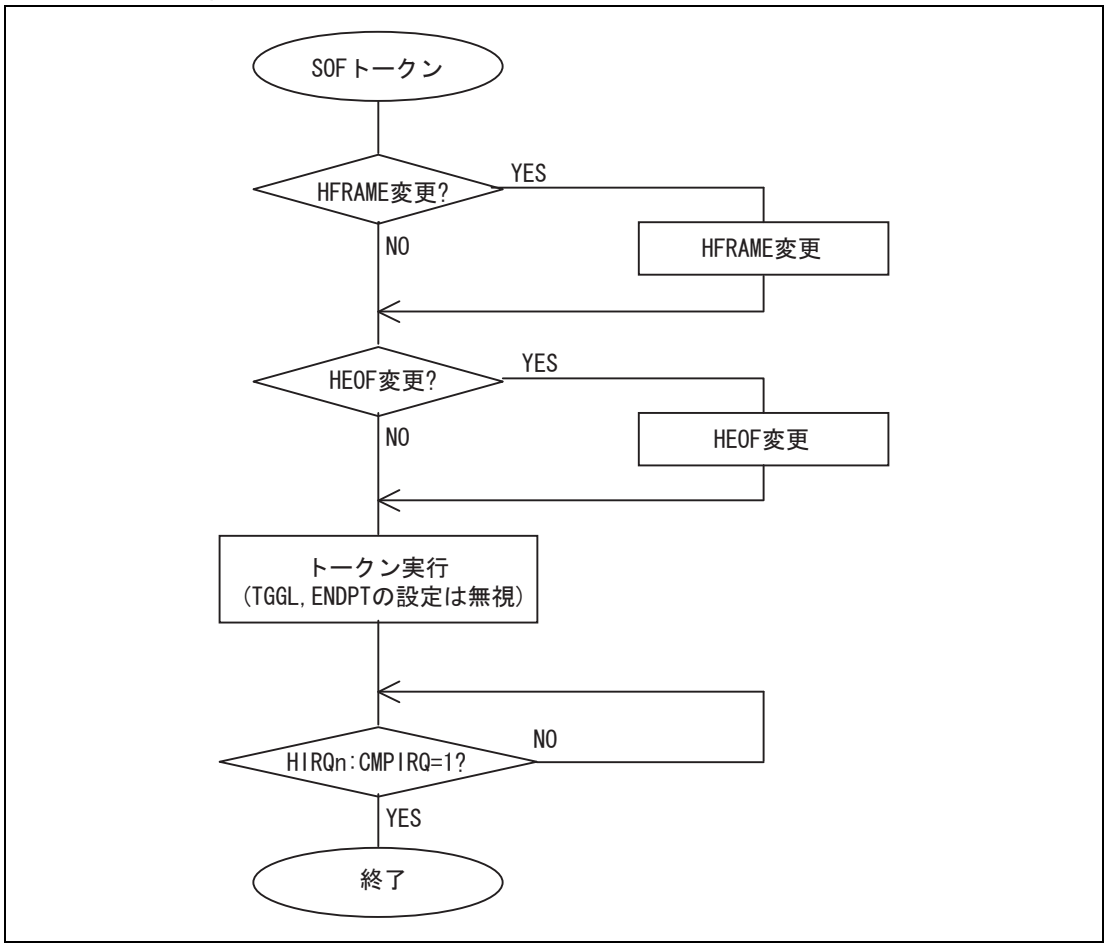
IN, OUT, SETUP トークン

Figure 30-28. IN, OUT, SETUPトークン時のフローチャート



SOFトークン

Figure 30-29. SOFトークン時のフローチャート



31. メモリパッチ機能

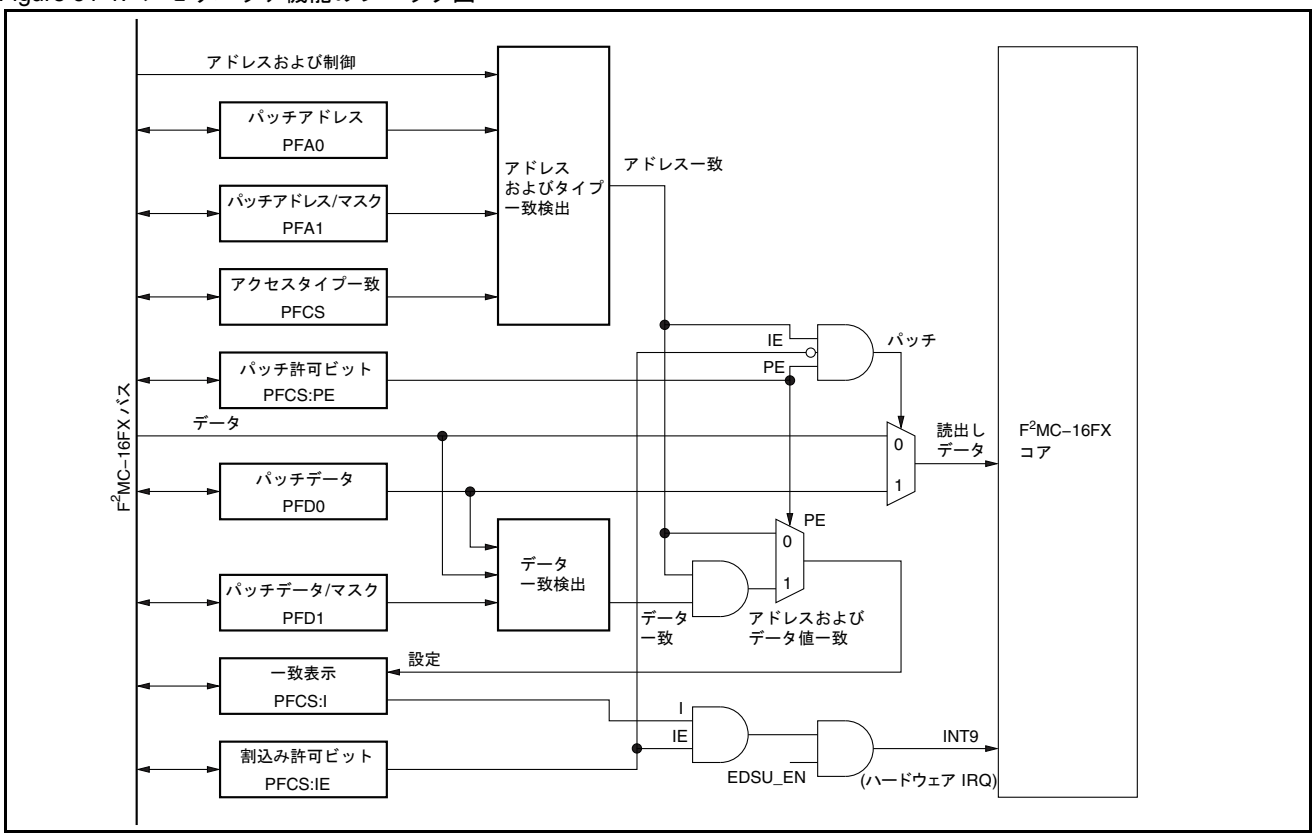
メモリパッチ機能を説明し, そしてデータパッチまたはデバッグ機能がどのようにメモリパッチ機能とともに動作するかを説明します。

31.1 メモリパッチ機能の概要

メモリパッチ機能は16FXのコアバス上でコードフェッチまたはデータアクセスのためのアドレスを検出することができます。アドレスがパッチ機能アドレスレジスタの設定値と一致する場合, 読み出されたコードまたはデータは一致したデータパッチレジスタが示す値と置き換えられるか, またはハードウェア割込みがアサートされます。
このデータ置換え機能を, ソフトウェア割込み/ブレークポイント, 分岐命令の挿入, またはコードもしくはデータワードを直接パッチするために使用することができます。 その結果, CPU はその命令を実行するか, もしくはメモリにアドレス指定された値の代わりにパッチレジスタに指定されたデータを読み出します。
ハードウェア割込みはデータブレークポイントの実装に使用することができます。

メモリパッチ機能の構造

Figure 31-1. メモリパッチ機能のブロック図



メモリパッチ機能 (MPF) には 8 つのチャネルがあり , 8 つの単独アドレスポイント , または 4 グループのアドレス範囲に分けて使用することができます。Figure 31-1 に MPF の構造を示します。

MPF のチャネルにはそれぞれパッチ機能アドレスレジスタ (PFAx) , パッチデータレジスタ (PFDx) が , そして 2 つのチャネルのグループにはコモンパッチ機能制御 / ステータスレジスタ (PFCSx) があります。PFCSx には , 各チャネルのアドレス一致表示フラグ (I), 割込み許可ビット (IE), パッチ許可ビット (PE) があります。2 つのチャネルのグループにはそれぞれコモンアドレスマスク (AM) とコモンアドレス範囲 (AR) 制御ビットがあります。さらに , 2 つのチャネルの各グループに対する特定のアクセスタイプ (READ, WRITE, BYTE, WORD, CODE, DATA, CPU,DMA) に対応する 1 バイト構成のデータがあります。

31.2 メモリパッチ機能のレジスタ

メモリパッチ機能には以下のレジスタがあります。

- パッチ機能アドレスレジスタ (PFA0 ～ PFA7)
- パッチ機能データレジスタ (PFD0 ～ PFD7)
- パッチ機能制御/ステータスレジスタ (PFCS0 ～ PFCS3)
- EDSU 拡張レジスタ (EDSU)

パッチ機能アドレスレジスタ(PFA0 to PFA7)

Figure 31-2. パッチ機能アドレスレジスタ(PFA0 to PFA7)

アドレス	byte+2	byte+1	byte+0	属性	初期値
PFA0 $0003BA_H$～$0003B8_H$	PFAH0	PFAM0	PFAL0	R/W	不定
PFA1 $0003BD_H$～$0003BB_H$	PFAH1	PFAM1	PFAL1	R/W	不定
PFA2 $0003C0_H$～$0003BE_H$	PFAH2	PFAM2	PFAL2	R/W	不定
PFA3 $0003C3_H$～$0003C1_H$	PFAH3	PFAM3	PFAL3	R/W	不定
PFA4 $0003C6_H$～$0003C4_H$	PFAH4	PFAM4	PFAL4	R/W	不定
PFA5 $0003C9_H$～$0003C7_H$	PFAH5	PFAM5	PFAL5	R/W	不定
PFA6 $0003CC_H$～$0003CA_H$	PFAH6	PFAM6	PFAL6	R/W	不定
PFA7 $0003CF_H$～$0003CD_H$	PFAH7	PFAM7	PFAL7	R/W	不定

R/W：リード/ライト可能

各バスアクセスのアドレスはパッチ機能アドレスレジスタ (PFA0 ～ PFA7) の内容と比較されます。バスアクセスのアドレスがパッチ機能アドレスレジスタの 1 つと一致し, かつアクセスタイプがPFCSx の上位 8 ビットの 1 つと一致する場合 , このチャネルのアドレス一致表示ビット (PFCSx:I) がセットされます。その後の動作はパッチ機能制御 / ステータスレジスタの中の IE または PE ビットの構成に依存します。

2 つのチャネルの各グループでは, アドレスマスクおよびアドレス範囲機能を使用することができます。例えば, 以下の記述は 0 チャネルと 1 チャネルのグループを使用します。

マスクオプション (PFCS0:AM=1) の場合は , PFA0 がアドレスを定義し , PFA1 がアドレスのマスクを定義します。マスクに "1" を含むビットは比較されず, 常に一致します。一致するアクセスタイプ (PFCSx の上位 8 ビット) の場合 , 次の式が真であれば , 一致表示ビット PFCS0:I0 が設定されます。

PFA0 | PFA1 == AD | PFA1

AD はバスのアドレスを反映します。

範囲オプション (PFCS0:AR=1) に対しては , PFA0 がアドレス範囲の開始ポイントを定義し , PFA1 はアドレス範囲の終了ポイントを定義します。一致するアクセスタイプ (PFCSx の上位 8 ビット) の場合は , PFCS0:I0 ビットが PFA0 ～ PFA1 の範囲に一致するすべてのアドレスで設定されます。

PFA0 <= AD <= PFA1

メモリパッチ機能 (PFCSx:PE=1 および PFCSx:IE=0) を使用するために , PFAx の bit0 は無視されます。アクセスしたワード全体がパッチされます。

パッチ機能アドレスレジスタ (PFA0 ～ PFA7) はリセット後は不定です。パッチ機能アドレスレジスタは書込み/読出しが可能です。

パッチ機能制御 / ステータスビットに関しては Table 31-2 を参照してください。そこではパッチ機能アドレスレジスタ PFAx, パッチ機能データレジスタ PFDx とパッチ機能制御 / ステータスレジスタ PFCSx との対応関係を示しています。

パッチ機能データレジスタ(PFD0 ～ PFD7)

Figure 31-3. パッチ機能データレジスタ(PFD0 ～ PFD7)

アドレス	byte+1	byte+0	属性	初期値
PFD0 0003D1$_H$～0003D0$_H$	PFDH0	PFDL0	R/W	不定
PFD1 0003D3$_H$～0003D2$_H$	PFDH1	PFDL1	R/W	不定
PFD2 0003D5$_H$～0003D4$_H$	PFDH2	PFDL2	R/W	不定
PFD3 0003D7$_H$～0003D6$_H$	PFDH3	PFDL3	R/W	不定
PFD4 0003D9$_H$～0003D8$_H$	PFDH4	PFDL4	R/W	不定
PFD5 0003DB$_H$～0003DA$_H$	PFDH5	PFDL5	R/W	不定
PFD6 0003DD$_H$～0003DC$_H$	PFDH6	PFDL6	R/W	不定
PFD7 0003DF$_H$～0003DE$_H$	PFDH7	PFDL7	R/W	不定

R/W：リード/ライト可能

パッチ機能データレジスタ (PFD0 ～ PFD7) は , 一致した場合の読出しデータの置換え値を定義します。アクセスタイプ (PFCSx の上位 8 ビット) およびパッチ機能アドレスレジスタ (PFAx) で一致し , かつ PEx=1 および IEx=0 の場合 , PFDx レジスタによって定義されたデータが読み出されます。

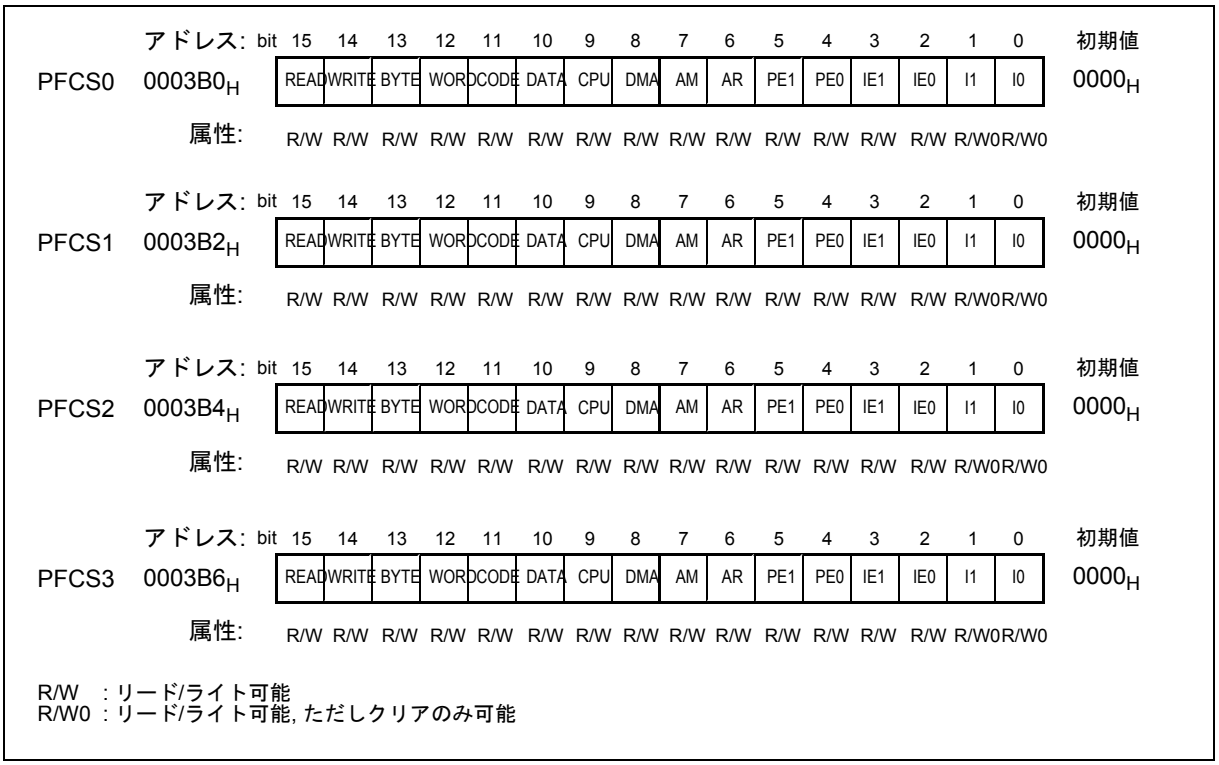
PFCSx:AM または PFCSx:AR が許可された場合 , 0/2/4/6 チャネルの一致のみが生成され , PFD0/2/4/6 のみがデータ置換えに使用されます。

PE ビットがセットされ , IE ビットがクリアされたときは , バス上で PFDx レジスタの値に置き換えられます。一方 PFDx は組込みデバッグサポートユニット (EDSU) で ,
オペランドブレーク検出のためのデータ値またはデータマスクを定義することに使用することができます。その目的 (データ値ブレーク) のためには , PE ビットと IE ビットの両方を "1" に設定する必要があります。

パッチ機能データレジスタ (PFD0 ～ PFD7) はリセット後不定です。パッチ機能データレジスタは書込み/読出しが可能です。

パッチ機能制御/ステータスレジスタ(PFCS0 ～ PFCS3)

Figure 31-4. パッチ機能制御/ステータスレジスタ(PFCS0～PFCS3)



パッチ機能制御 / ステータスレジスタ (PFCSx) はメモリパッチ機能の動作を制御し , 各 MPF チャネルの状態を反映します。各レジスタワード PFCSx は 2 つのチャネルのグループに属します。

制御 / ステータスビットの詳細については , Table 31-1 を参照してください。Table 31-1 では PFCS0 レジスタを例にして説明していますが , PFCS1 ～ PFCS3 レジスタも PFCS0 と各ビットが制御する機能は同じです。MPF の各チャンネルおよび PFA,PFD レジスタについては PFC0 に対応する ch0/1 および PFA0/1,PFD0/1 レジスタを例に説明しています。PFCS1 ～ PFCS3 に置き換えて考える場合 , MPF のチャンネルおよび PFA,PFD のインデックスは表の文中の値からそれぞれ PFCS1 には 2 を , PFCS2 には 4 を ,PFCS3 には 6 を加算してください。

Table 31-1. PFCS0 (MPF ch.0 および ch.1)の各ビットの機能

ビット名		機能
bit 15	READ	読出しアクセスをアドレス一致検出するかどうかを , 制御します。リードモディファイライトのリード部分を検出します。 • "0" 書込み ：読出しアクセスを一致検出しません。 • "1" 書込み ：読出しアクセスを一致検出します。 • 読出しおよび書込み可能です。 READ ビットはリセット後にクリアされます。
bit 14	WRITE	書込みアクセスをアドレス一致検出するかどうかを , 制御します。 • "0" 書込み ：書込みアクセスを一致検出 しません。 • "1" 書込み ：書込みアクセスを一致検出します。 • 読出しおよび書込み可能です。 WRITE ビットはリセット後にクリアされます。

Table 31-1. PFCS0 (MPF ch.0 および ch.1)の各ビットの機能

ビット名		機能
bit 13	BYTE	バイトアクセスをアドレス一致検出するかどうかを , 制御します。 • "0" 書込み : バイトアクセスを一致検出しません。 • "1" 書込み : バイトアクセスを一致検出します。 • 読出しおよび書込み可能です。 BYTE ビットはリセット後にクリアされます。
bit 12	WORD	ワードアクセスをアドレス一致検出するかどうかを , 制御します。 • "0" 書込み : ワードアクセスを一致検出しません。 • "1" 書込み : ワードアクセスを一致検出します。 • 読出しおよび書込み可能です。 WORD ビットはリセット後にクリアされます。
bit 11	CODE	コードフェッチのためのアクセスをアドレス一致検出するかどうかを , 制御します。 • "0" 書込み : コードアクセスを一致検出しません。 • "1" 書込み : コードアクセスを一致検出します。 • 読出しおよび書込み可能です。 CODE ビットはリセット後にクリアされます。
bit 10	DATA	データアクセスをアドレス一致検出するかどうかを , 制御します。 • "0" 書込み : データアクセスを一致検出しません。 • "1" 書込み : データアクセスを一致検出します。 • 読出しおよび書込み可能です。 DATA ビットはリセット後にクリアされます。
bit 9	CPU	CPU アクセスをアドレス一致検出するかどうかを , 制御します。 • "0" 書込み : CPU アクセスを一致検出しません。 • "1" 書込み : CPU アクセスを一致検出します。 • 読出しおよび書込み可能です。 CPU ビットはリセット後にクリアされます。
bit 8	DMA	DMA アクセスをアドレス一致検出するかどうかを , 制御します。 • "0" 書込み : DMA アクセスを一致検出しません。 • "1" 書込み : DMA アクセスを一致検出します。 • 読出しおよび書込み可能です。 DMA ビットはリセット後にクリアされます。
bit 7	AM	アドレスマスクビット AM は , MPF の ch.0 のアドレスマスクオプションを制御します。 • "0" 書込み : PFA0 のアドレスマスクオプションは禁止されます。アドレス範囲として使用されないなら , PFA1 には ch.1 のアドレスを定義します。 • "1" 書込み : PFA1 と PFA0 はバスのアドレス一致判定時にアドレス値の OR マスクとして使用されます。 AR ビットが "1" に設定された場合は , AM ビットの設定は無視されます (アドレスマスクなし) 。 • 読出しおよび書込み可能です。 マスクオプションが使用される場合は , MPF の ch.1 は使用できません。 AM ビットはリセット後にクリアされます。
bit 6	AR	アドレス範囲ビット AR は , MPF の ch.0 の範囲オプションを制御します。 • "0" 書込み : PFA0/PFA1 の範囲オプションは禁止されます。PFA1 がアドレスマスクとして使用されないなら, PFA0とPFA1にはch.0およびch.1が独立したアドレスとして定義します。 • "1" 書込み : アドレス一致判定時にバスのアドレス値の , PFA0 は開始アドレスとして , PFA1 は終了アドレスとして使用されます。 • 読出しおよび書込み可能です。 範囲オプションが使用される場合は , MPF の ch.1 は使用できません。 AR ビットはリセット後にクリアされます。
bit 5	PE1	パッチ許可ビットは , MPF の 1 チャネルのデータ置換え機能を制御します。 • "0" 書込み : データ置換え (パッチ) 機能は禁止されます。 • "1" 書込みかつ IE1=0 : データ置換え (パッチ) 機能は許可されます。アドレスが PFA1 と一致する場合 , PFD1 の値がアドレス指定されたメモリの場所の代わりに読み出されます。 "1" 書込みかつ IE1=1 : データ置換え (パッチ) 機能は禁止されます。ch.1 はデータ値ブレークを生成することはできません。PFD1 レジスタは , ch.0 がデータ値ブレークとして使用される場合 , データ値のマスク定義に割り当てられます。 • 読出しおよび書込み可能です。 PE1=1 かつ IE1=0 (パッチ) の場合 , PFA1 の bit0 は無視されます。 AM または AR のどちらかが設定される場合 , MPF の ch.1 にアドレス一致は生成されません。したがって , PE1 ビットはこの場合無効です。 PE1 ビットはリセット後にクリアされます。

Table 31-1. PFCS0 (MPF ch.0 および ch.1)の各ビットの機能

ビット名		機能
bit 4	PE0	パッチ許可ビットは , MPF の ch.0 のデータ置換え機能およびデータ値ブレークを制御します。 • "0" 書込み : データ置換え (パッチ) 機能は禁止されます。 • "1" 書込みかつ IE0=0 : データ置換え (パッチ) 機能は許可されます。アドレスが PFA0 と一致する場合 , PFD0 の値がアドレス指定されたメモリの場所の代わりに読み出されます。 "1" 書込みかつ IE0=1 : データ置換え (パッチ) 機能は禁止されます。PE0=1 および IE0=1 の組合せは , データ値一致の許可に使用します。(Table 31-3 参照) • 読出しおよび書込み可能です。 PE0=1 かつ IE0=0 (パッチ) の場合 , PFA0 の bit0 は無視されます。 PE0 ビットはリセット後にクリアされます。
bit 3	IE1	割込みフラグ I1 の許可ビットです。MPF の ch.1 のハードウェア割込みの発生を制御し , オペランドブレーク機能の実行を制御するために使用されます。 • "0" 書込み : 割込みが禁止されます。 • "1" 書込み : 割込みが許可されます。割込みアドレスおよびアクセスタイプが一致する場合は , INT9 ハードウェア割込みがアサートされます。 • 読出しおよび書込み可能です。 IE1 ビットはリセット後にクリアされます。 AM または AR が設定された場合, このビットは無効です。
bit 2	IE0	割込みフラグ I0 の許可ビットです。MPF の ch.0 のハードウェア割込み生成を制御し , オペランドブレーク機能の実行を制御するために使用されます。 • "0" 書込み : 割込みが禁止されます。 • "1" 書込み : 割込みが許可されます。割込みアドレスおよびアクセスタイプまたは データが一致する場合は , INT9 ハードウェア割込みがアサートされます。 さらに PE0 ビットが設定された場合 , ch.0 はデータ値ブレークとして動作します。さらにデータ値が一致すると , INT9 のアサートが要求されます。 • 読出しおよび書込み可能です。 IE0 ビットはリセット後にクリアされます。
bit 1	I1	ch.1 の一致表示 / 割込みフラグです。アドレス一致条件をアクセスタイプ一致と組み合せることができます。いったん設定すると , ソフトウェアでクリアされるまで設定を維持します。 • "0" 読出し : 最後のクリア動作から一致がないことを示します。 • "1" 読出し : 最後のクリア動作から少なくとも 1 回は一致があったことを示します。IE1 が設定された場合 , 割込みが発行されます。PE1 が設定され , IE1 がクリアされた場合 , バスの読出しデータは PFD1 に置き換えられます。 • "0" 書込みはビットをクリアします。 • "1" 書込みは , 無視されます。 • RMW 命令は常に "1" を読み出します。 I フラグへの書込みアクセスおよび I フラグを設定するハードウェアイベントが同時に発生する場合は , ハードウェアイベントが優先されます。 AM または AR のどちらかが設定されている場合 , MPF の ch.1 の一致信号は生成されません。したがって , I1 フラグはこの場合設定されません。 このビットはリセット後にクリアされます。 割込み機能を許可する (IE1=1) 前に, I1 ビットをクリアすることを推奨します。
bit 0	I0	ch.0 の一致表示 / 割込みフラグです。アドレス一致条件をアクセスタイプおよびデータ値一致と組み合せることができます。いったん設定すると , ソフトウェアでクリアされるまで設定を維持します。 • "0" 読出し : 最後のクリア動作から一致がないことを示します。 • "1" 読出し : 最後のクリア動作から少なくとも 1 回は一致があったことを示します。IE0 が設定された場合 , 割込みが発行されます。PE0 が設定され , IE0 がクリアされた場合 , バスの読出しデータは PFD0 で置き換えられます。 • "0" 書込みはビットをクリアします。 • "1" 書込みは , 無視されます。 • RMW 命令は常に "1" を読み出します。 I フラグへの書込みアクセスおよび I フラグを設定するハードウェアイベントが同時に発生した場合 , ハードウェアイベントが優先されます。 このビットはリセット後にクリアされます。 割込み機能を許可する (IE0=1)前に, I0 ビットをクリアすることを推奨します。

Table 31-2は, パッチ機能アドレスレジスタ PFAx, パッチ機能データレジスタ PFDx, パッチ機能制御/ステータスレジスタ PFCSx 間の対応関係を示します。

Table 31-2. PFAx, PFDx, PFCSx間の対応関係

パッチアドレス(PFA)				パッチデータ(PFD)	制御/ステータスレジスタ(PFCS)						
	AR=0 AM=0	AR=0 AM=1	AR=1		アクセスタイプ [15:8]	動作制御/ステータス[7:0]					
PFA0	ポイント	ポイント	開始ポイント	PFD0	PFCS0	PFCS0	I0	IE0	PE0	AM	AR
PFA1	ポイント	マスク	終了ポイント	PFD1			I1	IE1	PE1		
PFA2	ポイント	ポイント	開始ポイント	PFD2	PFCS1	PFCS1	I0	IE0	PE0	AM	AR
PFA3	ポイント	マスク	終了ポイント	PFD3			I1	IE1	PE1		
PFA4	ポイント	ポイント	開始ポイント	PFD4	PFCS2	PFCS2	I0	IE0	PE0	AM	AR
PFA5	ポイント	マスク	終了ポイント	PFD5			I1	IE1	PE1		
PFA6	ポイント	ポイント	開始ポイント	PFD6	PFCS3	PFCS3	I0	IE0	PE0	AM	AR
PFA7	ポイント	マスク	終了ポイント	PFD7			I1	IE1	PE1		

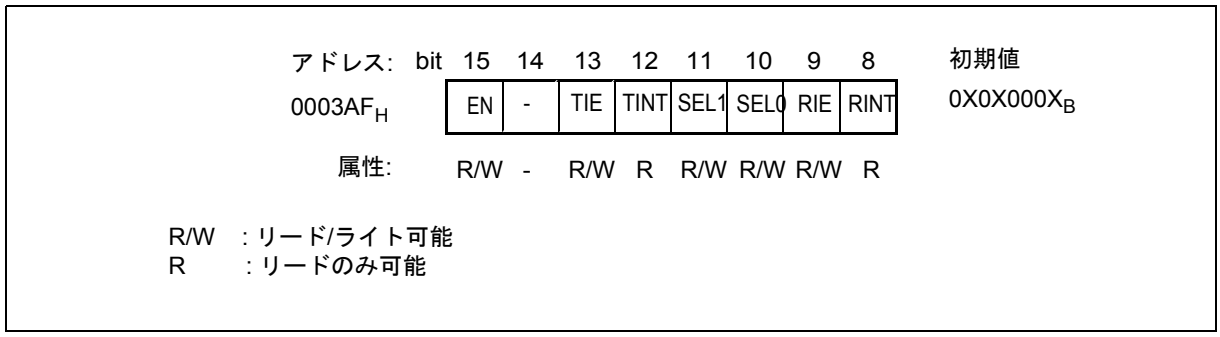
PFCSx:IExおよびPFCSx:PEx構成ビットは, MPFチャネルのメイン動作を選択します。

Table 31-3. PFCSのIE1, IE0およびPE1, PE0ビットによるチャネル動作定義

IE	PE	動作
0	0	全オフ設定 割込みもデータパッチも許可されません。 ただし, 許可されたアクセスタイプ (PFCS[15:8]) の 1 つと所定のアドレス (PFA, PFCS:AR, PFCS:AM) が一致している場合, アドレス一致が I フラグに書き込まれます。
0	1	メモリパッチ (ROM 修正およびメモリ保護) メモリパッチ機能が許可されます。アドレスとアクセスタイプが一致する場合 , バスのデータ値がパッチ機能データレジスタ (PFD) の内容で置き換えられます。 データ読込み時のデータ置き換えに加えて , アクセスしたアドレスに対するバス送信が無視されます。読込みの場合 , メモリパッチ機能は読出し動作として行なわれます。書込み動作は行なわれません。これにより , メモリ保護機能が実現されます。つまり , モデファイした後のデータの書換え動作はキャンセルされます。 通常はメモリパッチ機能は , アクセスタイプを区別する必要はありません。 したがって , PFCS[15:8] ビットはすべて "1" に設定してください。ただし , 明確なアクセスタイプの選別もまたパッチ機能とともに可能です。 オペコードを含むアドレスがこのメモリの位置を INT9 のオペコードでパッチするように構成されている場合は, この機能を命令ブレークの実行に使用できます。 元の命令の読出し値は INT9 のオペコードで置き換えられます。
1	0	オペランドアドレスブレーク この機能は主にオペランドアドレスブレークの実行に使用されます。アドレスとアクセスタイプが一致する場合は , I フラグが設定され , 割込み (INT9) が発行されます。アクセスタイプを明確なデータアクセス (PFCS:DATA=1 および PFCS:CODE=0) のために設定することができます。アクセスタイプのためのフィルタオプションは , 特定されたアクセスタイプの一致判別に有効です。 この機能はまたコードフェッチ(PFCS:DATA=0およびPFCS:CODE=1)に合致したアクセスをさせることができます。 フェッチされたコードは実行されないことがあることに注意してください (CPU のプリフェッチメカニズム)。 したがって, この機能は命令実行ブレーク実行のための使用は意図されていません。 命令ブレークにはメモリパッチ機能 (記載のように, IE=0 および PE=1) を使用してください。
1	1	データ値ブレーク この設定は , 明確なデータ値一致と組み合わされたオペランドアクセスブレークを実行します。アドレス , アクセスタイプ , データが PFD と一致する場合 , I フラグが設定され , 割込み (INT9) が発行されます。 データ値ブレークは PFD0, PFD2, PFD4, PFD6 チャネルにのみ利用できます。隣接する PFD (PFD1, PFD3, PFD5, PFD7) はデータマスクとして使用されます。ただし , 隣接する PFA がそのような目的のために構成されたとしても , 引き続きオペランドアドレスブレークに使用することができます。 その場合, パッチ機能は許可されません。

EDSU拡張レジスタ(EDSU)

Figure 31-5. EDSU拡張レジスタ(EDSU)



アプリケーションのカーネルでの直接的なデバッグ作業の実行を避けるため , システムにより , 外部イベントはデバッグ機能のシステムトリガとして利用されます。デバッグ機構についての詳細情報は , ブート ROM 仕様書を参照してください。

EDSU 拡張レジスタ (EDSU) は , アドレスおよびデータ一致機能に基づき , 組込みデバッグサポートを制御します。このレジスタが制御する機能は主に 2 つあります。

- デバッグ機能のための INT9 のハードウェアトリガの全般的な許可 (EDSU:EN)
- 通信インタフェースの選択
  受信/送信割り込みにより, INT9割込みをトリガして, このインタフェースの要求によるブレークを実行します(EDSU:SEL,EDSU:TIE,EDSU:TINT,EDSU:RIE,EDSU:RINT)。ブレーク割込み選択の拡張機能については, 「■ EDSU 拡張レジスタ 2 (EDSU2)」の説明を参照してください。

EDSU の制御 / ステータスビットの詳細については , Table 31-4 を参照してください。

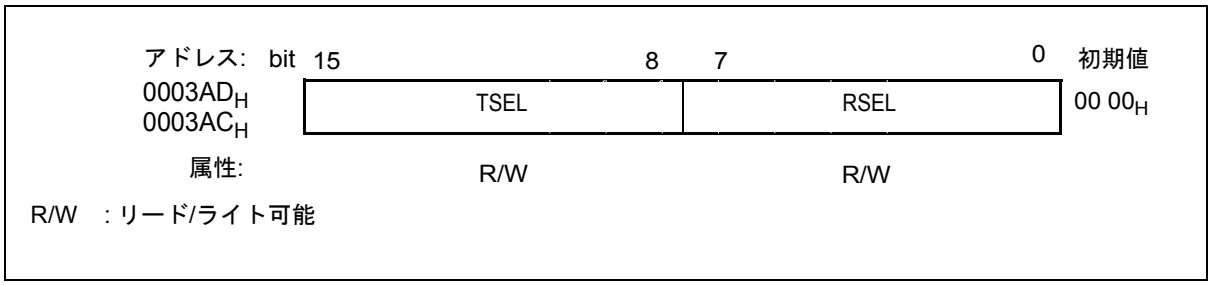
INT9の割込みベクタは, テーブルベースレジスタ(TBR)の値と独立した固定アドレス (0FFFD8$_H$)に配置します。INT9のベクタアドレスの所定の位置への配置についての詳細については, 「第3章　"割込み"」や「3.4 マスク不可能割込み(NMI)」を参照してください。本オプションにより,EDSU機能を活用したデバッグシステムの信頼性の向上が図られます。

Table 31-4. EDSUレジスタの各ビットの機能

ビット名		機能
bit 15	EN	許可ビット EN は , 組込みデバッグシステムのハードウェア INT9 要求の生成を制御します。一般にアドレス一致検出 , データ値一致検出 , および選択した通信デバイス受信の割込み生成を許可します。このビットにより , メモリパッチ機能は影響されません。 • "0" の書込み - 組込みデバッグサポートを禁止します。 • "1" の書込み - 組込みデバッグサポートを許可します。 • 読出しおよび書込み可能です。 EN ビットはリセット後にクリアされます。
bit 14	-	未使用ビット • 読出しは不定値を返します。 • このビットへの書込みは無視されます。
bit 13	TIE	選択した通信デバイスの INT9 送信割込みを許可します。 • "0" の書込み - IRQ 送信時の INT9 生成が禁止されます。 • "1" の書込み - IRQ 送信時の INT9 生成が許可されます。 • 読出しおよび書込み可能です。 TIE ビットはリセット後にクリアされます。
bit 12	TINT	選択した通信デバイスの送信割込みフラグの動作状態を示します。 • このビットは読出しのみです。 • 書込みは無視されます。 • 割込みフラグをクリアするには , 選択された通信デバイス (周辺リソース) の割込み一致フラグを使用してください。 初期値は選択された通信デバイスによります。該当する周辺リソースの仕様書を参照してください。
bit 11, bit10	SEL1, SEL0	モニタデバッガ通信デバイスを選択します。 • USART 0, USART 1, USART 2, USART 3 を以下の表に従い選択することができます。 • 読出しおよび書込み可能です。 SEL ビットはリセット後 "0" に初期化されます。 SEL1 | SEL0 | USART 0 | 0 | 0 0 | 1 | 1 1 | 0 | 2 1 | 1 | 3
bit 9	RIE	選択した通信デバイスの INT9 受信割込みを許可します。 • "0" の書込み : IRQ 受信時の INT9 生成が禁止されます。 • "1" の書込み : IRQ 受信時の INT9 生成が許可されます。 • 読出しおよび書込み可能です。 RIE ビットはリセット後にクリアされます。
bit 8	RINT	選択した通信デバイスの受信割込みフラグの動作状態を示します。 • このビットは読出しのみです。 • 書込みは無視されます。 • 割込みフラグをクリアするには , 選択された通信デバイス (周辺リソース) の割込み一致フラグを使用してください。 初期値は選択された通信デバイスによります。 該当する周辺リソースの仕様書を参照してください。

EDSU拡張レジスタ2 (EDSU2)

Figure 31-6. EDSU拡張レジスタ2 (EDSU2)



EDSU拡張レジスタ2（EDSU2)は,拡張ブレーク割込み選択によってINT9例外割込みのトリガを制御します。送信および受信割込みチャネルともに,および　割込み番号システムで利用可能なすべての割込みから選択が可能です。セレクタには制約条件がなく, 特定の送信/受信機能に結び付けられます。EDSU2 レジスタの詳細については, Table 31-5を参照してください。

Table 31-5. EDSU2レジスタの機能

ビット名		機能
bit 15 ～ bit 8	TSEL	EDSU レジスタの TINT ビットを反映するように割込みチャンネル番号を選択します。TIE が EDSU レジスタに設定されている場合は , 選択された割込み番号が , IRQ9 に送付されます。 • "00" 以外の番号を書き込む：拡張割込み番号の選択を可能とする。割込み番号は自由に選択可能です。 • "00" を書き込む：EDSU レジスタの SEL1 および SEL0 による標準選択法が有効です。 • 当該バイトは読込みおよび書込み可能 TSELはリセット後クリアされます。
bit 7～bit 0	RSEL	EDSU レジスタの RINT ビットを反映するように割込みチャンネル番号を選択します。RIE が EDSU レジスタに設定されている場合は , 選択された割込み番号が , IRQ9 に送付されます。 • "00" 以外の番号を書き込む：拡張割込み番号の選択を可能とする。割込み番号は自由に選択可能です。 • "00" を書き込む：EDSU レジスタの SEL1 および SEL0 による標準選択法が有効です。 • 当該バイトは読込みおよび書込み可能 RSELはリセット後クリアされます。

割り当てられた割込みチャネル番号を選択しない場合,有効な割込みレベルであっても、無効となります。割込みは, 接続されている周辺機能やリソースから直接得られます。しかしながら, 選択された周辺機能やリソースについて割込みを許可すべきです。

31.3 メモリパッチ機能の動作

メモリパッチ機能の種々の適用目的での使用方法を説明します。

メモリパッチ機能

これはアドレス一致を利用するデータ置換え機能です。ソフトウェア割込み / ブレークポイント , 分岐命令の挿入 , またはコードもしくはデータワードを直接パッチするために使用することができます。その結果 , CPU は命令を実行するか, もしくはメモリにアドレス指定された値の変わりにパッチレジスタに指定されたデータを読み出します。

メモリパッチ機能は, 16FX コアバスの読出しデータ値を, パッチ機能データレジスタ PFDx にロードされた値と置き換えます。パッチ機能は PFAx が指定するアドレスによって左右されます。

この機能を使用するには , 以下のように行ってください。

- 空いている MPF の "x" チャネルを選択し, PEx および IEx ビットをクリアしてその利用を禁止します。
- すべてのアクセスタイプに一致するように, PFCS[15:8] のすべてのビットを設定します。
- アドレスを PFAx に書き込みます (16 ビット - ワードアクセス, アドレスビット 0 は無視されます)。
- 置換えデータを PFDx に書き込みます。 バイトデータを修正する場合は, 元のメモリの場所から他のバイトをコピーしてください (隣接するバイトが定数ではないバイト変数をパッチすることはできません)。
- PEx ビットを許可します。

アドレスがパッチ機能アドレスレジスタの設定値に一致すると , メモリパッチ機能は次の動作を行います。

- アドレス一致表示ビット Ix が設定されます。
- アクセスはパッチ機能データレジスタに転送され,CPU またはDMA はPFDx レジスタに書き込まれた値を読み出します。
- すべての書込みアクセスは無効です。
  メモリ保護機能の実行の目的のため, アドレス一致検出機能の構成レジスタのアドレス範囲も書込みできないように保護されるように考慮すべきです。

命令アドレスブレーク

MPF の各チャネルを命令ブレークとして使用することができます。メモリパッチ機能と同じ手順を行います。特殊な場合としては , INT9 命令コード (01_H) がパッチ機能データレジスタの一致するバイトの位置に書き込まれるということです。PFAx は命令コードがあるアドレスを指定する必要があります。

マスクおよび範囲オプションを , 命令ブレークに影響されるアドレス領域の定義として使用することができます。

命令アドレス一致の範囲オプションに注意してください。奇数アドレスでの開始または終了には別々のチャネルを使用する必要があります。PE ビットが設定されると , アドレスビット 0 はマスクされます。

オペランドアドレスブレーク

MPF を使用し , オペランドアドレスブレークをそのアドレス一致機能により実現することができます。

この機能を使用するには , 以下のように行ってください。

- 空いている MPF の "x" チャネルを選択し, PEx および IEx ビットをクリアして使用を禁止にします。
- アドレスを PFAx に書き込みます (バイトアクセス, したがってアドレスビット 0 は有効です)。
- PFCSx で検出されるアクセスタイプを書き込みます。 基本設定の各グループから少なくとも 1 項目は選択する必要があります。

方向{READ|WRITE}, サイズ {BYTE|WORD}, 目的 {CODE|DATA}, マスタ {CPU|DMA}。
グループの両方の項目が許可された場合, その基本設定に関してはすべてのアクセスが一致します。 通常DATA=1およびCODE=0は,オペランドアドレスブレークの目的のためにオペランドアドレスのみを検出するのに選択されます。

- アドレスマスクと範囲オプションはアドレス一致条件の特定に有効な場合があります。
- IEx ビットを許可します。
- EDSU_EN ビットを"1"に設定します。

アドレスがパッチ機能アドレスレジスタの設定値に一致し , アクセスタイプが所定の設定に一致すると , 割込みがアサートされます。オペランドアドレスブレーク機能により , ハードウェアは INT9 例外を発生させます。

データ値ブレーク

データ一致規則 (データ値およびデータマスク) を書き込むためのパッチデータレジスタを使用して , オペランドアドレスブレーク機能を明確なデータ値一致と結びつけることができます。

この機能を使用するには , 以下のように行ってください。

- 上記のオペランドアドレスブレークと同じ手順で行ってください。
- 0/2/4/6 チャネルのみを使用できることに注意してください。 チャネルに奇数番号の付いた PFD レジスタ 1/3/5/7 はデータマスクを指定するために使用します。 一致する奇数チャネルの PE を許可しないでください。*
- 偶数番号のチャネルの PFD[x] にデータ値を書き込みます。
- PFD[x+1] (奇数番号のチャネル) にデータマスクを書き込みます。 "1"に設定されたビットは比較されません。 PFD[x+1] が 0000_H に等しいとき, PFD[x] の正確な値が照合されます。 変数へのバイトアクセスが照合される場合, 一致するバイトレーンへのマスク利用を検討してください (奇数アドレス: 下位バイトをマスク , 偶数アドレス: 上位バイトをマスク)。
- IEx ビットを許可します。
- PEx ビットを許可します。
- EDSU_EN ビットを"1"に設定します。

アドレスがパッチ機能アドレスレジスタの設定値に一致し , アクセスタイプが所定の設定に一致し , データがパッチデータレジスタに設定された値に一致すると , 割込みがアサートされます。データ値ブレーク機能により , ハードウェアは INT9 例外を発生させます。

*: 奇数番号のチャネルは PFD レジスタを必要としないため , 引き続きマスクなしにオペランドアドレスブレークとして使用することができます。

32. ROM/RAMミラー機能選択モジュール

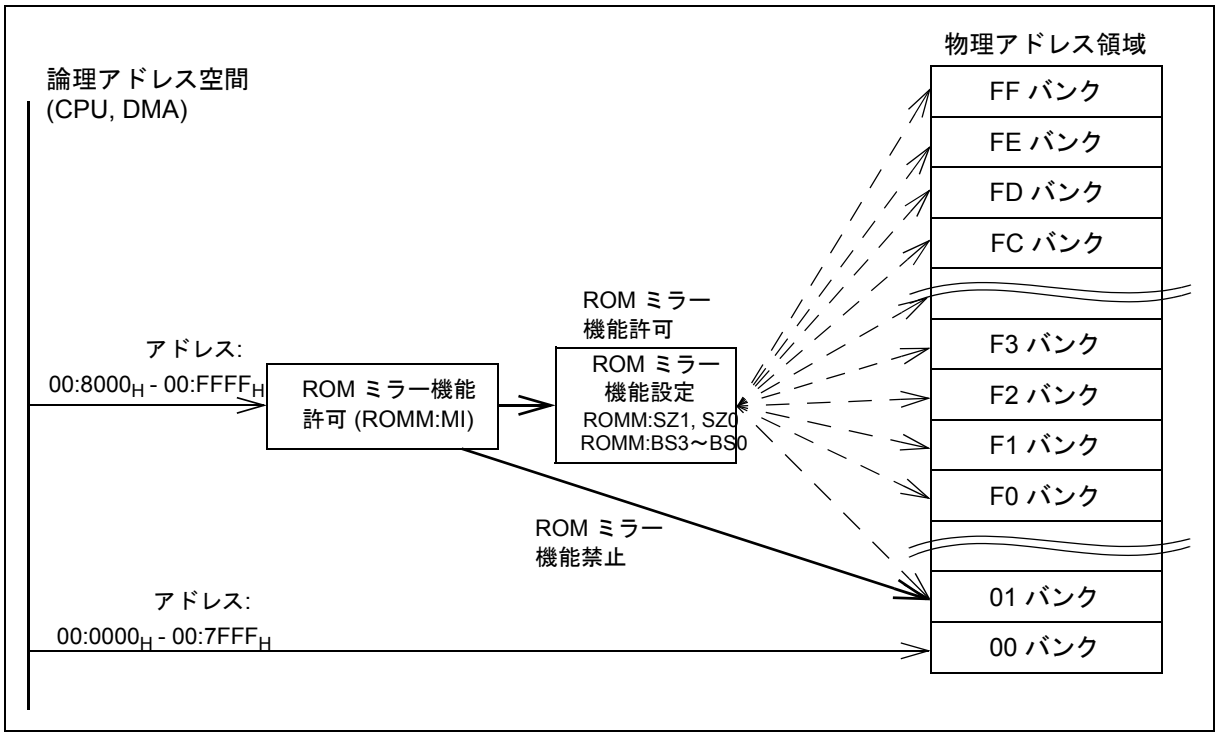
ROM ミラー機能選択モジュールについて説明します。

32.1 ROM/RAMミラー機能選択モジュールの概要

ROM/RAMミラー機能選択モジュールは,選択されたROMバンクの物理アドレスへ00バンクの論理アドレス(CPUアドレスなど)を割り当てます。ミラーリングされるROMのサイズは設定可能です。使用されていないROMミラー領域は, 01 バンクから RAM のミラーリングに使用されます。

ROM/RAMミラー機能選択モジュールのブロックダイヤグラム

Figure 32-1. ROM/RAMミラー機能選択モジュールのブロックダイヤグラム



ROM バンク $F0_H$ ～ FF_H (Fx:8000_H ～ Fx:$FFFF_H$) の上位半分の ROM アドレスのみを 00_H バンクへミラーリングすることができます。ROM バンクの下位半分はミラーリングできません。

ROM ミラー機能が禁止された場合 , 01_H バンクは 00_H バンクのミラーとして使用されます。

32.2 ROMミラー機能選択レジスタ(ROMM)

ROM ミラー機能選択レジスタ (ROMM) は ROM ミラー機能選択モジュールのすべての機能を設定します。

- ミラーリング ROM バンク
- ミラーリング ROM のサイズ
- ROM ミラー機能許可

ROMミラー機能選択レジスタ(ROMM)

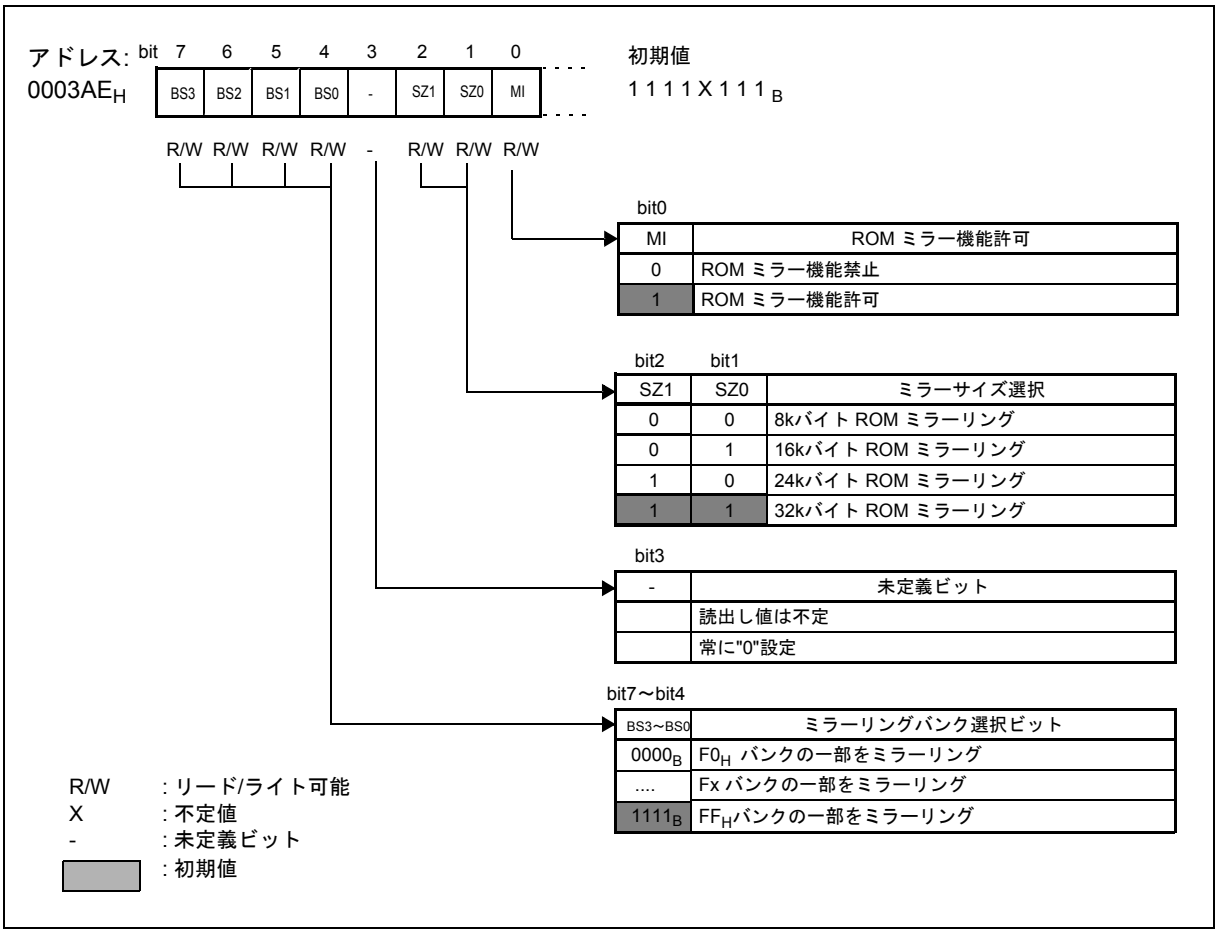


Table 32-1. ROMMの説明

<table>
<tr><th colspan="2">ビット名</th><th>機能</th></tr>
<tr><td>bit [7:4]</td><td>BS3 ～ BS0:
バンク選択
ビット</td><td>ミラーリング ROM バンクを選択します。 BS3～BS0 はメモリバンクアドレスの bit3～bit0 に対応します。

<table><tr><th>ROMM:BS[3:0]</th><th>ミラーリングメモリバンク</th></tr><tr><td>1111</td><td>FF</td></tr><tr><td>1110</td><td>FE</td></tr><tr><td>1101</td><td>FD</td></tr><tr><td>...</td><td>...</td></tr><tr><td>0001</td><td>F1</td></tr><tr><td>0000</td><td>F0</td></tr></table>
初期値は "1111" (FF_H バンク選択)です。</td></tr>
<tr><td>bit 3</td><td>未定義</td><td>このビットには常に "0" を書き込んでください。
読出し値は不定です。
リードモディファイライト系命令は無効です。</td></tr>
<tr><td>bit [2:1]</td><td>SZ1, SZ0:
ミラーサイズ
選択ビット</td><td>ミラーリング ROM バンクのサイズを選択します。詳細は Table 32-2 を参照してください。</td></tr>
<tr><td>bit 0</td><td>MI:
ミラー機能許可
ビット</td><td>ROM ミラー機能を許可します。
• "0" を書き込むと , ROM ミラー機能が禁止されます。アドレス $01:8000_H$ ～ $01:FFFF_H$ が 00 バンクへミラーリングされます。
• "1" を書き込むと , ROM ミラー機能が許可されます。ROMM:BS3 ～ BS0 で選択されたバンクの ROM データのイメージは 00 バンクでもアクセスできます。
• 初期値は"1" (ROM ミラー機能許可)です。</td></tr>
</table>

選択した ROM ミラーサイズが 32kB よりも小さい場合は , 代わりに 01_H バンクからのアドレス領域が 00_H バンクへミラーリングされます。01_H バンクがアドレス $01:8000_H$ の上に RAM を含む場合は , この RAM は 00_H バンクからアクセスできます。

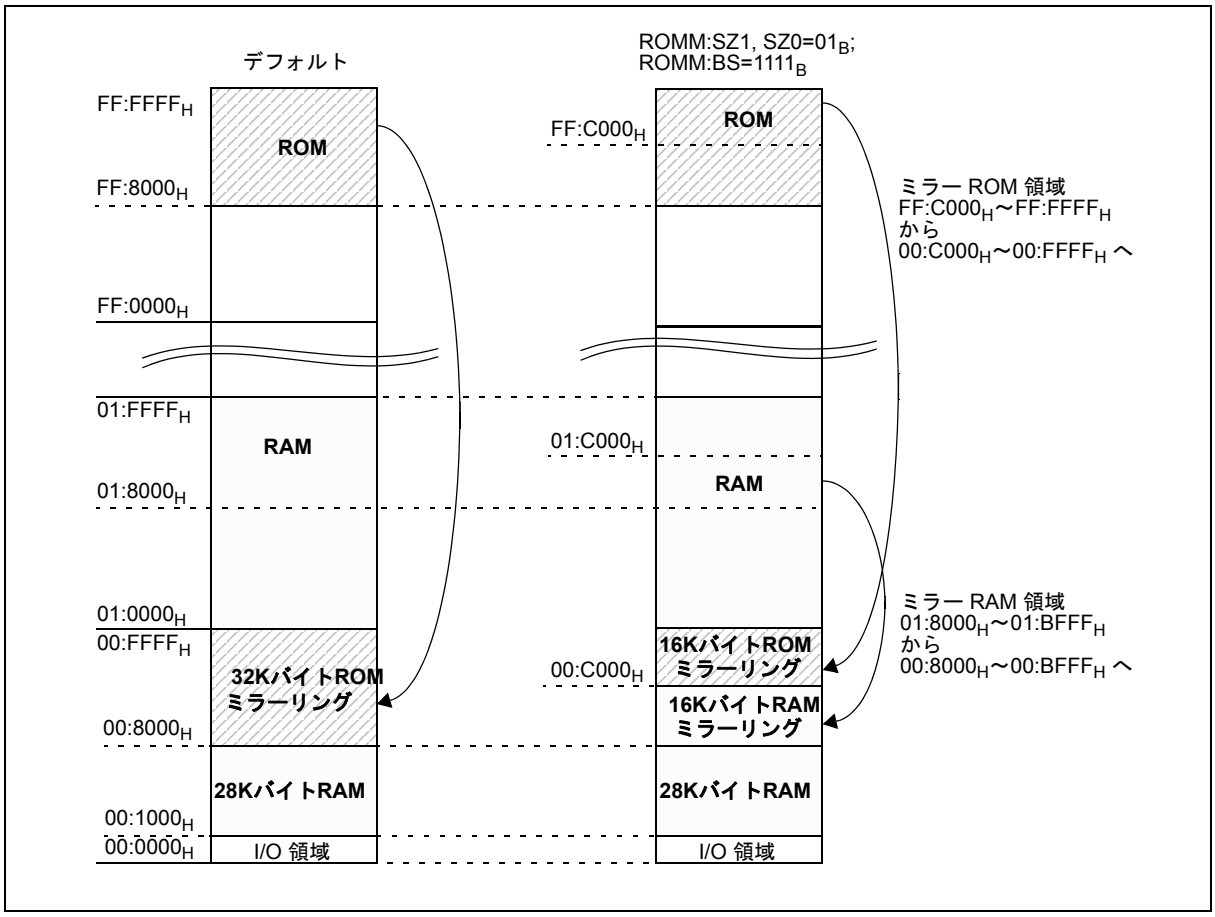
Table 32-2. ミラーサイズ選択ビットSZ[1:0]の説明

MI	SZ1	SZ0		ミラーリング ROM/RAMサイズ	ミラーリング用の選択 ROM/RAM 領域 (物理アドレス)	ROM/RAM ミラーリング領域 (論理アドレス)
0	-	-	ROM	0kバイト	ROM ミラー機能なし	ROM ミラー機能なし
			RAM	32kバイト	01.8000_H-$01.FFFF_H$	00.8000_H-$00.FFFF_H$
1	0	0	ROM	8kバイト	$Fx.E000_H$ - $Fx.FFFF_H$	$00.E000_H$ - $00.FFFF_H$
			RAM	24kバイト	01.8000_H-$01.DFFF_H$	00.8000_H-$00.DFFF_H$
1	0	1	ROM	16kバイト	$Fx.C000_H$ - $Fx.FFFF_H$	$00.C000_H$ - $00.FFFF_H$
			RAM	16kバイト	01.8000_H-$01.BFFF_H$	00.8000_H-$00.BFFF_H$
1	1	0	ROM	24kバイト	$Fx.A000_H$ - $Fx.FFFF_H$	$00.A000_H$ - $00.FFFF_H$
			RAM	8kバイト	01.8000_H-$01.9FFF_H$	00.8000_H-$00.9FFF_H$
1	1	1	ROM	32kバイト	$Fx.8000_H$ - $Fx.FFFF_H$	00.8000_H - $00.FFFF_H$
			RAM	0kバイト	RAM ミラー機能なし	RAM ミラー機能なし

＜注意事項＞

コードがミラーリングメモリ領域内のアドレスから実行されているときは , ROM ミラー機能選択レジスタ (ROMM) に書き込まないでください。

Figure 32-2. ROMミラー機能の例



33. フラッシュメモリ

フラッシュメモリの機能および動作について説明します。

33.1 フラッシュメモリの概要

フラッシュメモリは,ユーザプログラムとデータの保存に使用されます。F^{2}MC-16FXフラッシュMCU製品に搭載されるフラッシュメモリのタイプはデバイスによって異なります。各デバイスの詳細なフラッシュ構成については, データシート内の「フラッシュデバイスのユーザROMメモリマップ」を参照してください。
CPUからの制御により, フラッシュメモリにデータを書き込んだり, データを消去したりすることができます。したがって, MCUがPCBに実装されている状態でも, フラッシュメモリを書き換えることができます。その結果, プログラムやデータを効率的に改善することができます。

33.1.1 フラッシュメモリのタイプ

F^{2}MC-16FX MCU では搭載されるフラッシュモジュールのタイプはデバイスによって異なります。各デバイスに搭載されているフラッシュメモリのタイプについては , データシートを参照してください。データシートでは , 各デバイスの詳細なセクタ構成についても説明しています。

プログラム/データフラッシュ(メインフラッシュ)メモリ

高い読み出し性能を発揮するために最適化された , フラッシュマクロです。F^{2}MC-16FX MCU には , このタイプのモジュールが最大 2 個搭載されます (フラッシュ A, フラッシュ B)。
一部のデバイスには , フラッシュ B とよばれる第 2 のフラッシュメモリが搭載されており , これはフラッシュ A とは独立して動作します。

このフラッシュB の小セクタはバンク DE に配置され , オンチップの E^{2}PROM エミュレーション , ブートローダなどに使用できます。

データフラッシュ

このメモリは CPU によるプログラミングおよびオンチップの E^{2}PROM エミュレーション用に最適化されています。このタイプのフラッシュメモリは F^{2}MC-16FX MCU に最大 2 つ搭載されます (データフラッシュ A, データフラッシュ B)。

33.1.2 フラッシュメモリの特長

すべてのフラッシュモジュール共通の特長

- 自動プログラムアルゴリズムを使用
- 消去一時停止/再開機能を搭載
- データポーリングまたはトグルビット機能による書込み/消去完了検出
- CPU割込みによる書込み/消去完了検出
- セクタ消去機能(セクタの組合せ自由)
- 連続モード書込み機能

プログラム/データフラッシュ(メインフラッシュ)メモリの特長

- CPU (ユーザプログラムおよびシリアルフラッシュライタ) およびフラッシュメモリモード (パラレルフラッシュライタ) での書込み/消去が可能
- CPUおよびフラッシュメモリモードでのバイトおよびワードの書込みが可能
- フラッシュ読出しサイクルタイム: 最小で1マシンサイクル
- CPU読出しは32ビット幅で実行。ワードは読出しバッファに格納され,次のCPU読出しアクセスにおいてウェイトサイクルなしに利用できます。
- すべてのマシンクロック周波数で最適なアクセス性能を実現するため, フラッシュ制御信号とウェイトサイクルの設定が可能
- CPUモード時はセクタごとの消去/書込み禁止設定が可能
- アプリケーションが意図しない不正な方法でFlashメモリ上のコードが読み出される事を防ぐためのコードセキュリティ機能

データフラッシュの特長

- 書込み/消去はCPUからのみ実行可能(ユーザプログラムおよびシリアルフラッシュライタ)
- フラッシュマクロのデータ幅は8ビット
- 内蔵シーケンサハードウェア(書込みコマンドシーケンサ)によりバイトおよびワードの読出しが可能
- CPUからの書込みアクセスモードは2種類存在 (ダイレクトアクセスとコマンドシーケンサアクセス)
- 書込みアクセスデータ幅は, ダイレクトアクセスモードで8ビット, コマンドシーケンサアクセスモードで8/16ビット
- DMAによるデータ書込みが可能

33.1.3 フラッシュメモリへの書込み/消去

フラッシュメモリの書込み / 消去を実行中は , そのフラッシュメモリからデータを読み出す事はできません。
消去 / 書込み動作を開始する前に必要なデータ / コードを RAM にコピーする必要があります。
複数のフラッシュメモリモジュールが搭載されている場合 , 1 つのフラッシュメモリモジュールからコードやデータを読み出している最中に , 別のフラッシュメモリモジュールの消去 / 書込みが可能です。
セクタ消去は一時停止可能です。一時停止状態では , 消去対象以外のセクタからデータの読み出しおよび , データ書込みが可能です。

33.2 フラッシュメモリのセクタ構成

内蔵フラッシュメモリのセクタ構成を示します。

フラッシュメモリのセクタ構成

Figure 33-1 は , プログラム / データフラッシュ(メインフラッシュ) のセクタ構成を示します。図中のアドレスは , 各セクタの最上位アドレスと最下位アドレスを示します。
8KB の小セクタは , フラッシュ A メモリのサイズに応じて最大 8 セクタまで , SA0 を先頭として上位アドレス側に向けて SA0 ～ SA7 に配置されます。
64KB の大セクタは , フラッシュ A メモリのサイズに応じて , S39 を先頭として下位アドレス側に向けて S39 ～ S8 に配置されます。

フラッシュ B メモリが搭載されているデバイスでは , 8KB の小セクタが , フラッシュ B メモリのサイズに応じて最大 8 セクタまで , SB0 を先頭として上位アドレス側に向けて SB0 ～ SB7 に配置されます。
フラッシュ B メモリの 64KB の大セクタは , フラッシュ B メモリのサイズに応じて , フラッシュ A メモリの大セクタの下側に配置されます。
各デバイスにおいてどのようにフラッシュメモリのセクタが配置されているかは, データシートを参照してください。

Figure 33-1. プログラム/データフラッシュ(メインフラッシュ)のセクタ構成

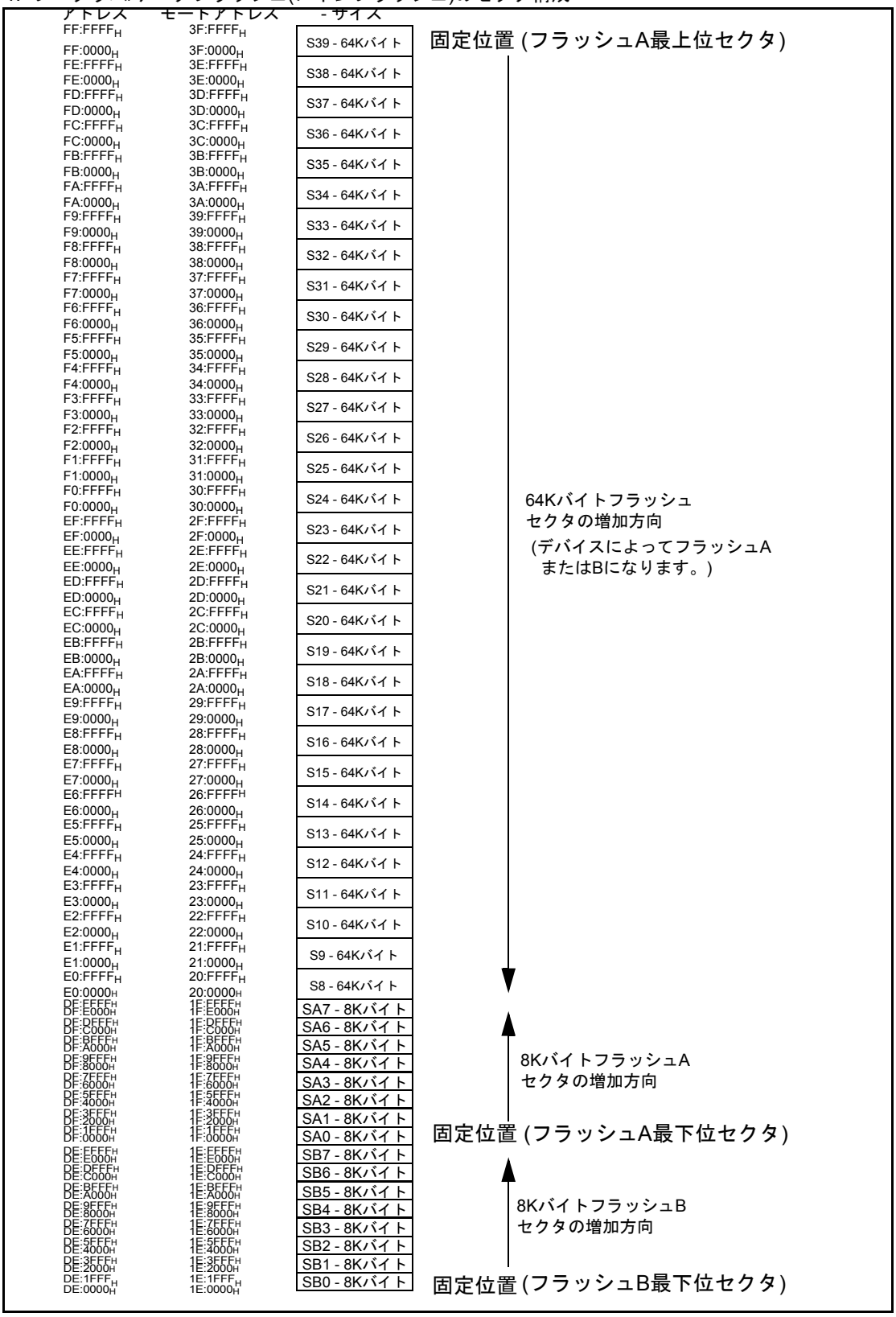


Figure 33-2 は , データフラッシュメモリを搭載した製品向けのセクタ構成を示します (各デバイスでのデータフラッシュモジュールの利用可否についてはデータシートを参照してください)。

データフラッシュはフラッシュセキュリティ機能の設定用に 256 バイトの小さなセクタ (構成ブロック) を備えています。データ A, データ B フラッシュの構成ブロックは共に CPU バンク 0E にマッピングされます (セクタ SDA0 と SDB0)

各データフラッシュモジュールは 16K バイトのデータセクタをそれぞれ 4 つ備えています。データフラッシュ A のセクタは CPU バンク 0D にマッピングされ , データフラッシュ B のセクタは CPU バンク 0C にマッピングされます。

フラッシュメモリモードのアドレスは参考値として Figure 33-2 に記載しております。現在データフラッシュモジュールはパラレルフラッシュライタには対応していませんので注意してください。

Figure 33-2. データフラッシュメモリのセクタ構成

CPUモードアドレス	フラッシュメモリモードアドレス (現在使用不可)	セクタ番号-サイズ	
$0E:FFFF_H$ $0E:FF00_H$	$0E:FFFF_H$ $0E:FF00_H$	SDA0 - 256バイト	データフラッシュA (構成ブロック)
$0E:FEFF_H$ $0E:FE00_H$	$0A:FEFF_H$ $0A:FE00_H$	SDB0 - 256バイト	データフラッシュB (構成ブロック)
$0E:FDFF_H$ $0E:0000_H$		予約	
$0D:FFFF_H$ $0D:C000_H$	$0F:FFFF_H$ $0F:C000_H$	SDA4 - 16Kバイト	データフラッシュA
$0D:BFFF_H$ $0D:8000_H$	$0F:BFFF_H$ $0F:8000_H$	SDA3 - 16Kバイト	
$0D:7EFF_H$ $0D:4000_H$	$0F:7EFF_H$ $0F:4000_H$	SDA2 - 16Kバイト	
$0D:3FFF_H$ $0D:0000_H$	$0F:3FFF_H$ $0F:0000_H$	SDA1 - 16Kバイト	
$0C:FFFF_H$ $0C:C000_H$	$0B:FFFF_H$ $0B:C000_H$	SDB4 - 16Kバイト	データフラッシュB
$0C:BFFF_H$ $0C:8000_H$	$0B:BFFF_H$ $0B:8000_H$	SDB3 - 16Kバイト	
$0C:7FFF_H$ $0C:4000_H$	$0B:7FFF_H$ $0B:4000_H$	SDB2 - 16Kバイト	
$0C:3FFF_H$ $0C:0000_H$	$0B:3FFF_H$ $0B:0000_H$	SDB1 - 16Kバイト	

33.3 フラッシュメモリモード

フラッシュメモリは, パラレルフラッシュ書込みモードとCPUモードの2つの方法でアクセスすることができます。パラレルフラッシュ書込みモードでは, 外部端子からフラッシュへ直接データの書込みおよび消去を行います。CPUモードでは, CPUから内部バスを介してフラッシュメモリへの書込み/消去を行います。Table 8-3 に記載されているとおり, モード選択にはモード端子MD2～MD0を使用します。

パラレルフラッシュ書込みモード

リセット信号がアクティブのときにモード端子を MD2 ～ MD0 = 111 に設定すると , CPU が停止します。フラッシュメモリが外部ポートに直接接続されるため , 外部端子からフラッシュメモリを直接制御することができます。

このモードでは外部から MCU は標準的なフラッシュメモリと同様にみなすことができ , フラッシュメモリプログラマを使用して書込み / 消去が可能です。

詳細はフラッシュメモリプログラマのマニュアルを参照してください。

＜注意事項＞

データフラッシュへの対応についてはプログラミング機器メーカーに確認してください。

CPUモード

プログラム / データフラッシュ (メインフラッシュ) メモリは , CPU メモリ空間の DE ～ FF バンクに配置され , データフラッシュメモリは , 0C ～ 0E バンクに配置されています。どちらも通常のマスク ROM と同様に CPU から読出しアクセスでき , また書込みアクセスをすることができます。

フラッシュメモリの書込み / 消去は , CPU からの命令で実行されるため , このモードでは , MCU が対象ボードに実装されていても再書込みが可能です。

33.4 プログラム/データフラッシュ(メインフラッシュ)メモリ制御レジスタ

プログラム/データフラッシュ(メインフラッシュ)メモリの制御/ステータスレジスタを説明します。このレジスタはフラッシュメモリの消去/書込み処理の制御と読出しアクセスタイミングの設定に使われます。

フラッシュメモリ制御レジスタ

フラッシュ A メモリは , 以下のレジスタで制御されます。

- フラッシュメモリ制御/ステータスレジスタA (MCSRA)
- フラッシュメモリタイミング設定レジスタA (MTCRAL/MTCRAH)

フラッシュ B メモリは , 以下のレジスタで制御されます。

- フラッシュメモリ制御/ステータスレジスタB (MCSRB)
- フラッシュメモリタイミング設定レジスタB (MTCRBL/MTCRBH)

フラッシュメモリのセクタ消去 / 書込み許可は , 以下のレジスタで制御されます。

- フラッシュメモリ書込み制御レジスタ (FMWC0～FMWC5)

33.4.1 フラッシュメモリ制御ステータスレジスタ(MCSRA, MCSRB)

(MCSRA：フラッシュAメモリ, MCSRB：フラッシュBメモリ)は, フラッシュメモリの基本機能となるデータの読出しおよび書込み/消去の制御を行います。

フラッシュメモリ制御ステータスレジスタ(MCSRA, MCSRB)

Figure 33-3. フラッシュメモリ制御ステータスレジスタ(MCSRA, MCSRB)の構成

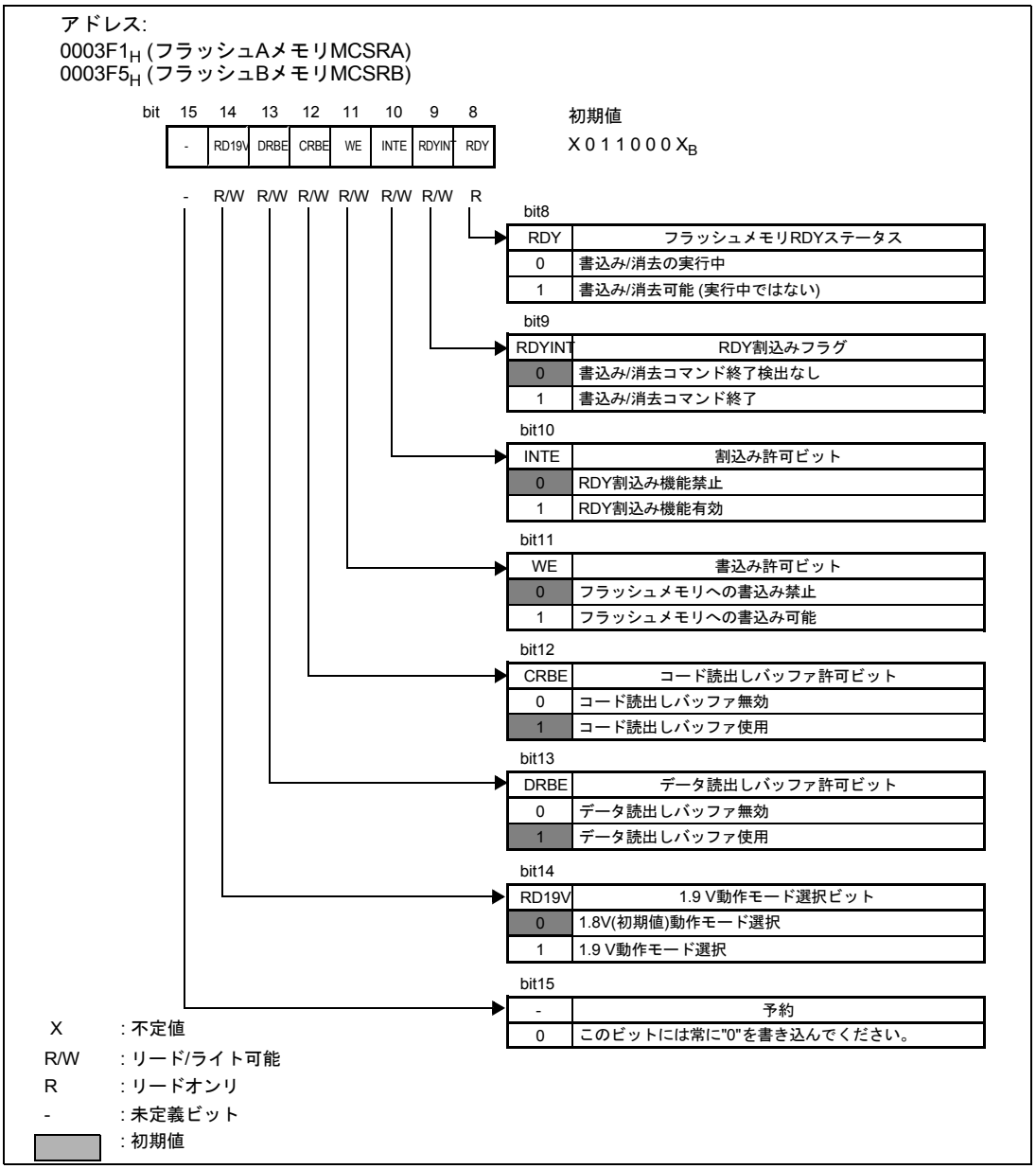


Table 33-1. フラッシュメモリ制御ステータスレジスタ(MCSRA/MCSRB)の各ビットの機能

ビット名		機能
bit 8	RDY: フラッシュメモリ RDY ステータス	• 本ビットによりフラッシュメモリの状態を参照できます。 • このビットへの書込みは , 影響しません。 • "0" の読出し値は , プログラム / 消去コマンドが現在実行されていることを示します。 • "1"の読出し値は, プログラム/消去コマンドが現在実行されていないか, セクタ消去一時停止状態にあることを示します。
bit 9	RDYINT: RDY 割込み フラグ	• フラッシュメモリレディ割込みフラグ • このビットはリセットにより "0" に初期化されます。 • RDY 信号の立上りにより "1" に設定されます。 • 本ビットは "0" 書込みによりクリアされます。本ビットへの "1" 書込みは無効です。 • リードモディファイライトアクセスの読出しサイクルは常に "1" を返します。 •
bit 10	INTE: 割込み 許可ビット	• 本ビットは書込み / 消去コマンド終了による割込み発生の許可 / 禁止を制御します。 • 本ビットが "1" に設定されて , RDYINT フラグが "1" の場合または RDYINT フラグが RDY 信号の立上りによって "1" になった時に割込みが発生します。 • 本ビットが "0" に設定されている場合 , RDYINT 割込みは発生しません。 • INTE ビットの初期値は "0" です。 • 本ビットは読出しおよび書込み可能です。
bit 11	WE: 書込み許可ビット	• 本ビットはフラッシュメモリへの書込みの許可 / 禁止設定を制御します。 • このビットが "0" のときは , 書込み / 消去コマンドはフラッシュメモリに送信されません。 • このビットを "1" に設定するとコマンド書込みが許可されます。 • このビットの初期値は"0"です。

Table 33-1. フラッシュメモリ制御ステータスレジスタ(MCSRA/MCSRB)の各ビットの機能

ビット名		機能
bit 12	CRBE: コード 読出し バッファ 許可ビット	• コード読出しバッファの許可 / 無効設定を行います。 • このビットを "1" に設定するとコード読出しバッファが許可されます。すべてのアドレスからのコード読出しアクセスは 2 つの 16 ビットワードを読み出します。1 つは指定されたアドレスのワードで , もう 1 つは隣接するアドレスのワードで , 結果として 32 ビット境界に並ぶ 2 つの 16 ビットワードが読み出されます。両方のワードは読出しバッファに格納されます。バッファに格納されたコードのうち , 同一の 32 ビット境界内のワードへの CPU アクセスはフラッシュのウェイトサイクルなしに実行されます。 • このビットを"0"に設定するとコード読出しバッファが無効になります。コードは常にフラッシュメモリから直接読み出されます。 • このビットの初期値は"1"(バッファ使用)です。
bit 13	DRBE: データ 読出し バッファ 許可ビット	• データ読出しバッファの許可 / 無効設定を行います。 • このビットを "1" に設定するとデータ読出しバッファが許可されます。すべてのアドレスからのデータ読出しアクセスは 2 つの 16 ビットワードを読み出します。1 つは指定されたアドレスのワードで , もう 1 つは隣接するアドレスのワードで , 結果として 32 ビット境界に並ぶ 2 つの 16 ビットワードが読み出されます。両方のワードは読出しバッファに格納されます。バッファに格納されたコードのうち , 同一の 32 ビット境界内のワードへの CPU アクセスはフラッシュのウェイトサイクルなしに実行されます。 • このビットを "0" に設定するとデータ読出しバッファが無効になります。 データは常にフラッシュメモリから直接読み出されます。 • このビットの初期値は"1" (バッファ許可)です。
bit 14	RD19V: 1.9V動作モード 選択ビット	• 本ビットは , 電圧レギュレータの設定にしたがって設定してください (コア電圧を 1.8V あるいは 1.9V に設定 ,「第 9 章 スタンバイモードおよび 電圧レギュレータ制御回路」の「9.3 電圧レギュレータ制御レジスタ (VRCR)」を参照してください)。 • コア電圧 1.8V：このビットを "0" に設定してください。 • コア電圧 1.9V：このビットを "1" に設定してください。 • 内部クロックやフラッシュメモリタイミング設定に対する許容最大周波数は , 本ビットと選択されたコア電圧により定まります。1.9V モードのみで許容された周波数に変更する前に , 必ず , 電圧レギュレータと本ビットを 1.9V モードに設定してください。 • 本ビットの初期値は, "0"です。
bit 15	予約	• 常に "0" に設定してください。 • 読出し値は不定です。 • 本レジスタへのリードモディファイライト(RMW)系命令はこのビットには影響しません。

33.4.2 フラッシュメモリタイミング設定レジスタ(MTCRA, MTCRB)

フラッシュメモリタイミング設定レジスタ (MTCRA, MTCRB) は , フラッシュメモリのアクセスモードおよびウェイト状態の数を設定するために使用されます。Table 33-3 に推奨される設定を示します。

フラッシュメモリタイミング設定レジスタ下位バイト(MTCRAL, MTCRBL)

Figure 33-4. フラッシュメモリタイミング設定レジスタ(MTCRAL, MTCRBL)の構成

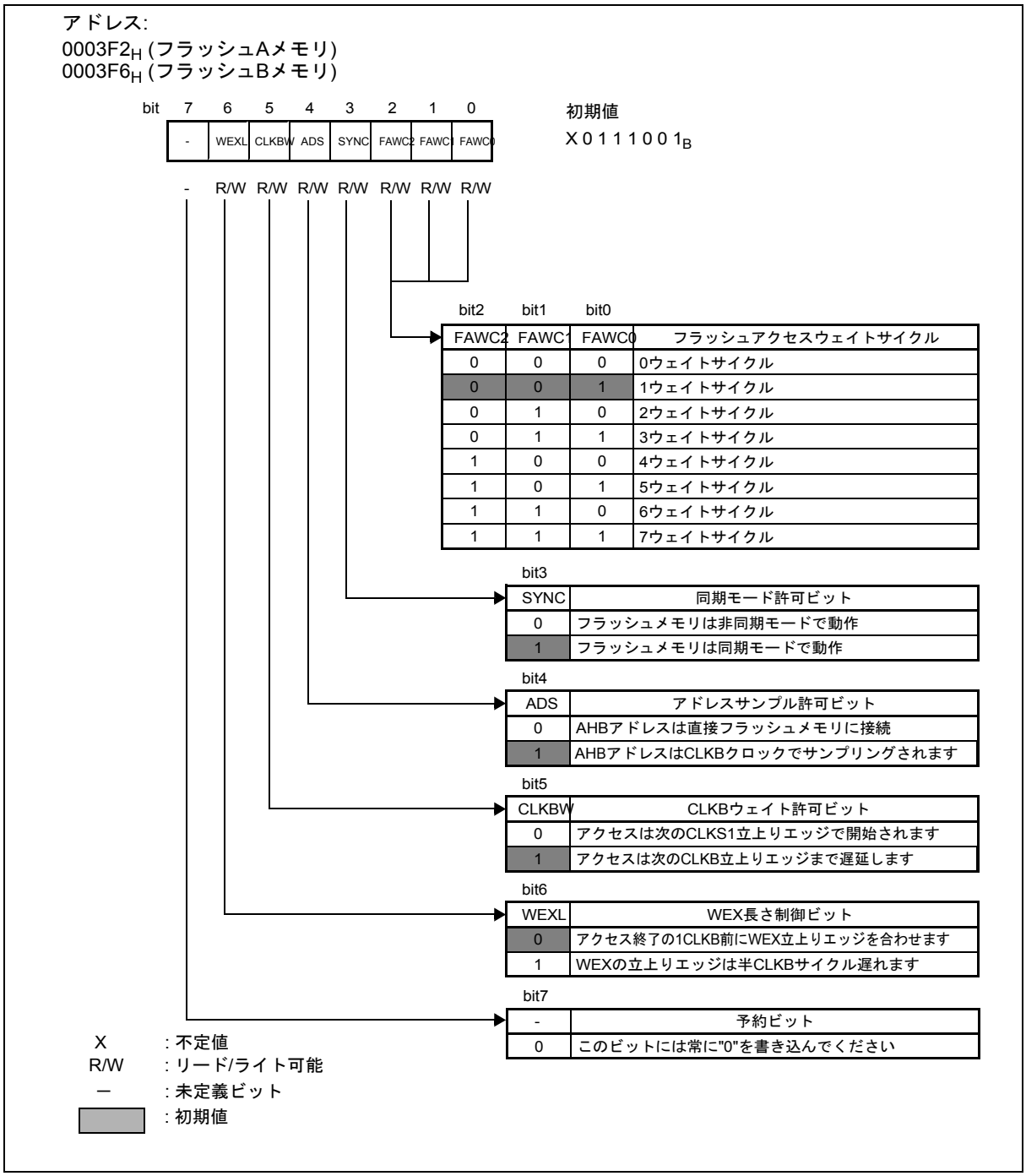


Table 33-2. フラッシュメモリタイミング設定レジスタ(MTCRAL/MTCRBL)の各ビットの機能

ビット名		機能
bit 0～bit 2	FAWC2 ～ FAWC0: フラッシュメモリアクセスウェイトサイクル	• フラッシュメモリの読出しおよび書込みアクセスに追加される CPU ウェイトサイクル (CLKB サイクル) の数を定義します。 • 本ビットの設定は , マシンクロック周波数およびほかの設定ビットに応じて決定してください。 • 初期値は"001" (1ウェイトサイクル)です。
bit 3	SYNC: 同期モード許可ビット	• このビットは , CPU モードでのフラッシュメモリの同期モード / 非同期モードの選択に使用します。 • このビットを "1" に設定すると , 同期モードが選択されます。フラッシュメモリの SYNC 入力が 1 に設定され , ATDIN および EQIN 入力信号が使用されます。 • このビットを "0" に設定すると , 非同期モードが選択されます。ATDIN および EQIN は使用されません。 • このビットの設定は読出しおよび書込み動作に影響します。 • このビットはリセットにより"1"に初期化されます。
bit 4	ADS: アドレスサンプル許可ビット	• このビットは, アドレス信号をフラッシュメモリに直接接続するか, サンプリングしたアドレス信号を接続するかを選択します。 • このビットを "0" に設定すると , AHB アドレス信号はフラッシュメモリへ直接接続されます。 • このビットを "1" に設定すると , アドレスのサンプリングが有効となり , アドレスは , アクセスの開始時に CLKB でサンプリングされ , アクセス終了まで保持されます。 • このビットを "1" に設定することは MTCRxH:ATDL が奇数となっているタイミング設定 (MTCRxH:ATDL0=1) でのみ許可されています。Table 33-3 の推奨設定は影響を受けません。 • 非同期モードおよび, フラッシュメモリへの書込みを行う際には , このビットを "1" に設定する必要があります。 • このビットの初期値は"1"です。.
bit 5	CLKBW: CLKBウェイト許可ビット	• このビットはフラッシュメモリアクセスの開始時間を定義します。 • このビットが "0" に設定されると , フラッシュメモリアクセスは次の CLKS1 の立上りタイミングで開始します。 • このビットが "1" に設定されると , フラッシュメモリアクセスの開始を次の CLKB 立上りタイミングまで遅らせます。 • フラッシュメモリへ書き込む前に , このビットを "1" に設定する必要があります。 • CLKB ＝ CLKS1 に設定されると (1 分周設定時), このビットは影響しません。 • このビットの初期値は"1"です。
bit 6	WEXL: WEX長さ制御ビット	• このビットは WEX 立上りのタイミングを制御します。 • このビットを "0" に設定すると , 書込みアクセス終了前の最後の CLKB 立上りで , WEX"1" になります。(アクセスの 1CLKB 前) • このビットを "1" に設定すると , WEX がアサートされる期間が半 CLKB クロックサイクル延長されます。 • FAWC ビットで設定されるウェイトサイクルの数は , WEX パルス長が有効になるために必要とされる長さを満たしている必要があります。 • このビットの初期値は"0"です。
bit 7	予約	• このビットには常に "0" を書き込んでください。 • 読出し値は不定です。 • 本レジスタへのリードモディファイライト(RMW)系命令はこのビットには影響しません。

フラッシュメモリタイミング設定レジスタ上位バイト(MTCRAH, MTCRBH)

Figure 33-5. フラッシュメモリタイミング設定レジスタ(MTCRAH, MTCRBH)の構成

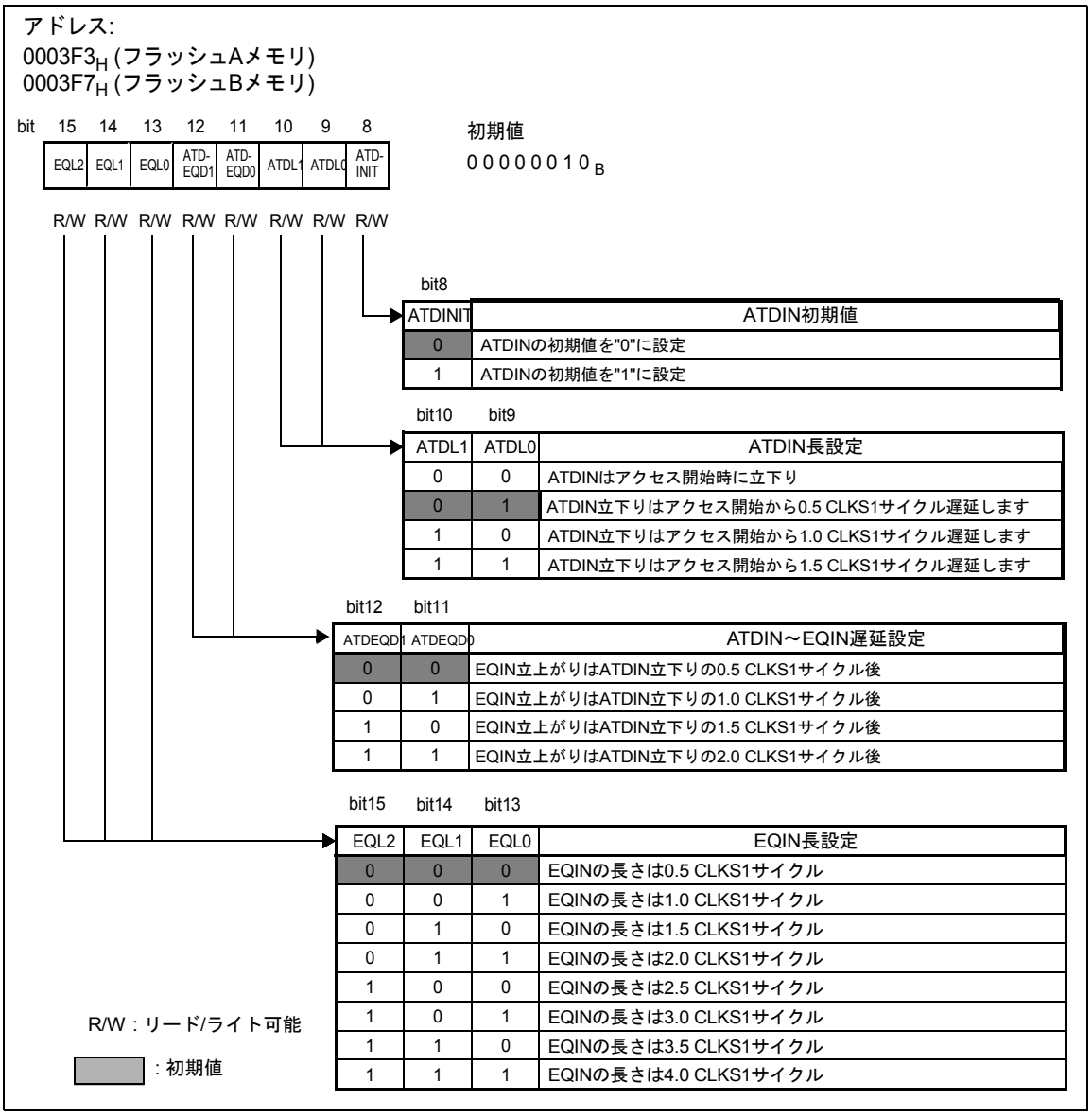


Table 33-3. フラッシュメモリタイミング設定レジスタ(MTCRAH/MTCRBH)の各ビットの機能

Bit name		Function
bit 8	ATDINIT: ATDIN初期値設定	• ATDIN 信号の初期値を定義します。 • このビットを "0" に設定すると , ATDIN 信号の初期値は "0" となります。ATDIN はフラッシュメモリアクセス (読出しまたは書込み) の開始時に "1" にアサートされます。この場合 , ATDL は "01" もしくはそれよりも上位に設定される必要があります。 • このビットを "1" に設定すると , ATDIN の初期値は "1" となります。ATDIN は , フラッシュメモリアクセス終了時に "1" になります。 • 本ビットはリセットにより"0"に初期化されます。
bit 9～ bit 10	ATDL1, ATDL0: ATDIN長設定	• 本ビットは Figure 33-5 の表に従い , ATDIN パルスの長さを定義します。 • 本ビットの設定は , CLKS1 周波数および ATDINIT のレベルに応じて行う必要があります。 • 本ビットの設定は同期および非同期モードでの書込みアクセスにおける WEX のアサート時間にも影響します。 • 本ビットはリセットにより"01"に初期化されます。
bit 11, bit 12	ATDEQD1, ATDEQD0: ATDIN～EQIN遅延設定	• 本ビットは Figure 33-5 の表に従い , ATDIN 立下りと EQIN 立上り間の長さを定義します。 • 本日ビットの設定は , CLKS1 周波数に応じて行う必要があります。 • 本ビットの設定は同期および非同期モードでの書込みアクセスにおける WEX アサート時間に影響します。 • 本ビットはリセットにより"00"に初期化されます。
bit 13～ bit 15	EQL2 ～ EQL0: EQIN長設定	• 本ビットは Figure 33-5 の表に従い , EQIN パルスの長さを定義します。 • 本ビットでの設定は , CLKS1 周波数に応じて行う必要があります。 • 本ビットはリセットにより"000"に初期化されます。

推奨されるフラッシュメモリタイミング設定

Table 33-4は,推奨されるフラッシュメモリタイミング設定を示します。各種タイミング構成の最大動作周波数は , コア電源電圧によって定まります。コア電源電圧のプログラミング方法の詳細については, 「第9章 スタンバイモードおよび電圧レギュレータ制御回路」を参照してください。

Table 33-4. MTCRA, MTCRB推奨設定

MTCRA, MTCRB	CLKS1 の許容周波数範囲 *1)		CLKB の許容分周設定	結果として CLKB に設定可能な周波数範囲 *1)		目的
	1.8V動作	1.9V動作		1.8V動作	1.9V動作	
6E3Dh	f_{CLKS1} ≦ 100 MHz (あらゆる設定において使用可能,特にモード遷移について)		1 ... 16	f_{CLKB} ≦ 100 MHz (あらゆる設定において使用可能,特にモード遷移において)		5ウェイト設定での同期読出し
6E3Fh						7ウェイト設定での同期読出し/書込み
0239h (初期値)	f_{CLKS1} ≦ 25 MHz		1 ... 16	f_{CLKB} ≦25 MHz		1ウェイト設定での同期読出し
2129h	f_{CLKS1} ≦ 30 MHz	f_{CLKS1}≦ 32 MHz	1 ... 16	f_{CLKB} ≦ 30 MHz	f_{CLKB} ≦ 32 MHz	1ウェイト設定での同期読出し
233Ah	f_{CLKS1} ≦ 50 MHz	f_{CLKS1}≦ 56 MHz	1 ... 16	f_{CLKB} ≦ 50 MHz	f_{CLKB} ≦56 MHz	2ウェイト設定での同期読出し
	f_{CLKS1} ≦ 20 MHz			f_{CLKB} ≦ 20 MHz		2ウェイト設定での同期読出し/書込み
4B3Bh	f_{CLKS1} ≦ 60 MHz	f_{CLKS1}≦ 64 MHz	1 ... 16	f_{CLKB} ≦ 60 MHz	f_{CLKB}≦64 MHz	3ウェイト設定での同期読出し
4B3Dh						5ウェイト設定での同期読出し/書込み

Table 33-4. MTCRA, MTCRB推奨設定

MTCRA, MTCRB	CLKS1 の許容周波数範囲 *1)		CLKB の許容分周設定	結果として CLKB に設定可能な周波数範囲 *1)		目的
	1.8V動作	1.9V動作		1.8V動作	1.9V動作	
2128h	$f_{CLKS1} \leqq 30$ MHz	$f_{CLKS1} \leqq 32$ MHz	2 ... 16	$f_{CLKB} \leqq 15$ MHz	$f_{CLKB} \leqq 16$ MHz	0ウェイト設定での同期読出し
2208h	20 MHz $\leqq f_{CLKS1} \leq$ 50 MHz	20 MHz $\leqq f_{CLKS1} \leq$ 56 MHz	2	10 MHz $\leqq f_{CLKB} \leqq$ 25 MHz	10 MHz $\leqq f_{CLKB} \leqq$ 28 MHz	0ウェイト設定での同期読出し
4C09h	$f_{CLKS1} \leqq 76$ MHz	$f_{CLKS1} \leqq 84$ MHz	2	$f_{CLKB} \leqq 38$ MHz	$f_{CLKB} \leqq 42$ MHz	1ウェイト設定での同期読出し
6B09h	20 MHz $\leqq f_{CLKS1} \leqq$ 92 MHz	20 MHz $\leqq f_{CLKS1} \leqq$ 100 MHz	2	10 MHz $\leqq f_{CLKB} \leqq$ 46 MHz	10 MHz $\leqq f_{CLKB} \leqq$ 50 MHz	1ウェイト設定での同期読出し
2279h	$f_{CLKS1} \leqq 40$ MHz		2 ... 16	$f_{CLKB} \leqq 20$ MHz		1ウェイト設定での同期読出し/書込み
0231h	$f_{CLKS1} \leqq 7$ MHz		1 ... 16	$f_{CLKB} \leqq 7$ MHz		1ウェイト状態での非同期読出し/書込み

*1 : f_{CLKS1} および f_{CLKB} の最大値は , データシートで規定された許容値を超えてはなりません。

＜注意事項＞

クロック設定を変更する場合 (CLKS1 のクロック設定 , あるいは CLKB の分周設定の変更など) は常に , 変更前後のどちらのクロック設定に対しても有効なフラッシュメモリタイミングを使用してください。6E3DH/6E3FH 設定は , すべてのクロック設定変更において使用可能です。
CLKS1 のクロックソースとして変調クロック CLKMOD を使用する場合は , CLKMOD の周波数の最大値および最小値が , CLKS1 周波数の許容範囲内であることを確認してください。
Table 33-4 は , MB96F348T/HxA には適用できません。

33.4.3 フラッシュメモリ書込み制御レジスタ0～5 (FMWC0～FMWC5)

フラッシュメモリ書込み制御レジスタ(FMWC0～FMWC5)は,フラッシュメモリの各セクタの消去/書込み許可を制御します。

フラッシュメモリ書込み制御レジスタ(FMWC0～FMWC5)

Figure 33-6. フラッシュメモリ書込み制御レジスタ(FMWC0～FMWC5)の構成

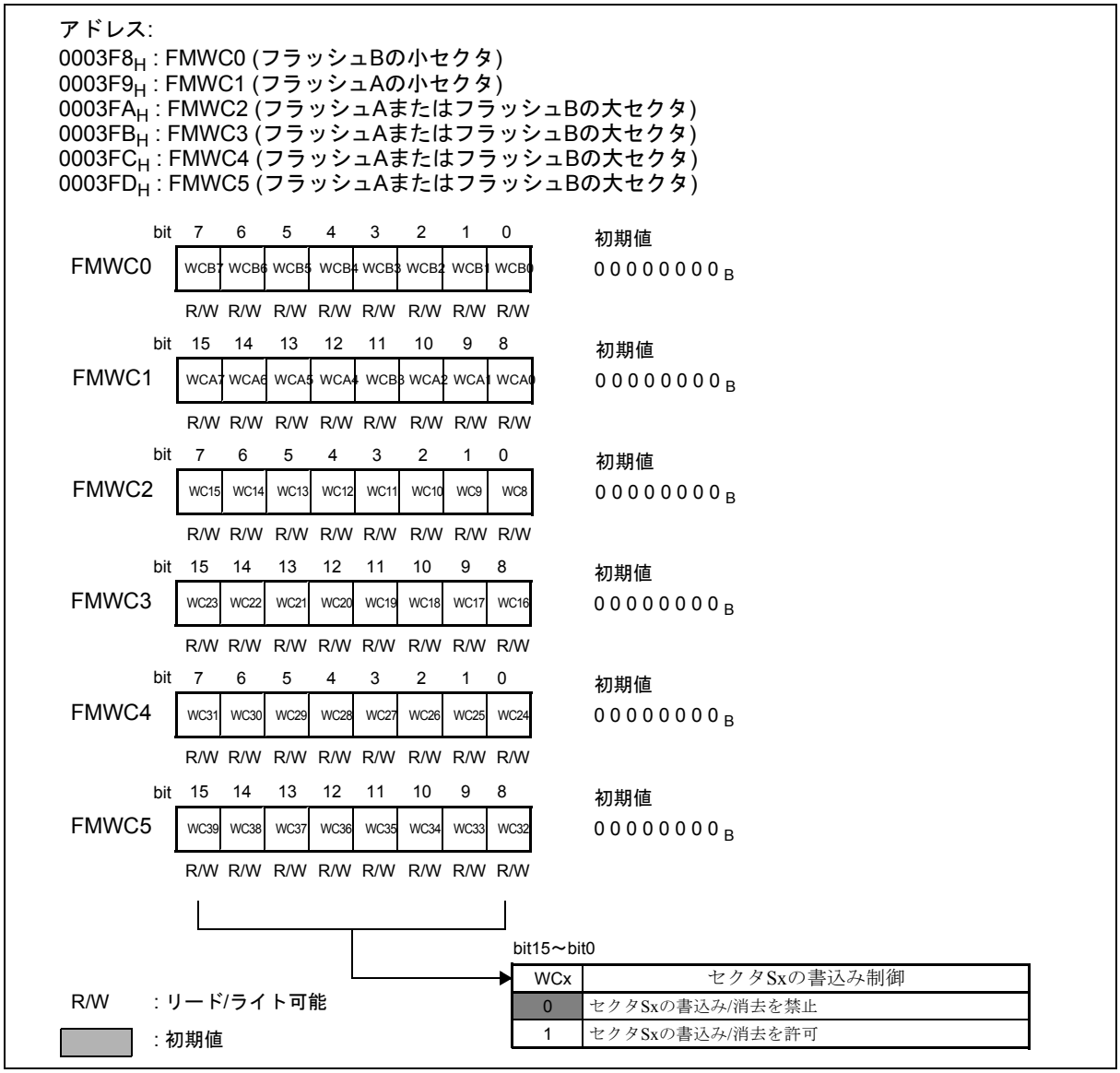


Table 33-5. フラッシュメモリ書込み制御レジスタ(FMWC0～FMWC7)の各ビットの機能

ビット名		機能
bit 0～bit 15	WCB7 ～ WCB0, WCA7 ～ WCA0 WC39 ～ WC8: セクタSB7～SB0, SA7～SA0および S39～S8の書込み 制御ビット	• 対応するセクタの書込み / 消去の可否を制御します。 • リセット後 , これらのビットは "0" にリセットされます (プログラム / 消去不可)。 • このビットを "1" に設定すると , 対応するセクタのプログラム / 消去が許可されます。 • "0" に設定を戻すと , プログラム / 消去が再び禁止されます。ビットはいつでも "0" に設定できますが , いったん "0" に設定を戻すと , "1" への再設定は不可能となります。 • 存在しないセクタに対応するビットへは常に "1" を書き込んでください。これらのビットのリード値は不定です。 • 本レジスタには, リードモディファイライト(RMW) 系命令は使用できません。

33.5 データフラッシュメモリ制御レジスタ

データフラッシュメモリの消去/書込み処理の制御と読出しアクセスタイミングの設定に必要なデータフラッシュの制御/ステータスレジスタを説明します。

データフラッシュメモリ制御レジスタ

データフラッシュ A メモリは , 以下のレジスタで制御されます。

- データフラッシュメモリ制御/ステータスレジスタA (DFCSA)
- データフラッシュメモリ書込みコマンドシーケンサ制御レジスタA (DFWCA)
- データフラッシュメモリ書込みコマンドシーケンサステータスレジスタA (DFWSA)

データフラッシュ B メモリは , 以下のレジスタで制御されます。

- データフラッシュメモリ制御/ステータスレジスタB (DFCSB)
- データフラッシュメモリ書込みコマンドシーケンサ制御レジスタB (DFWCB)
- データフラッシュメモリ書込みコマンドシーケンサステータスレジスタB (DFWSB)

33.5.1 データフラッシュメモリ制御ステータスレジスタ(DFCSA, DFCSB)

データフラッシュメモリ制御ステータスレジスタ(DFCSA：データフラッシュAメモリ， DFCSB：データフラッシュBメモリ)は， データフラッシュメモリの読出しおよび書込み/消去機能(書込みコマンドシーケンサを除く)を制御します。

データフラッシュメモリ制御ステータスレジスタ(DFCSA, DFCSB)

Figure 33-7. データフラッシュメモリ制御ステータスレジスタ(DFCSA, DFCSB)の構成

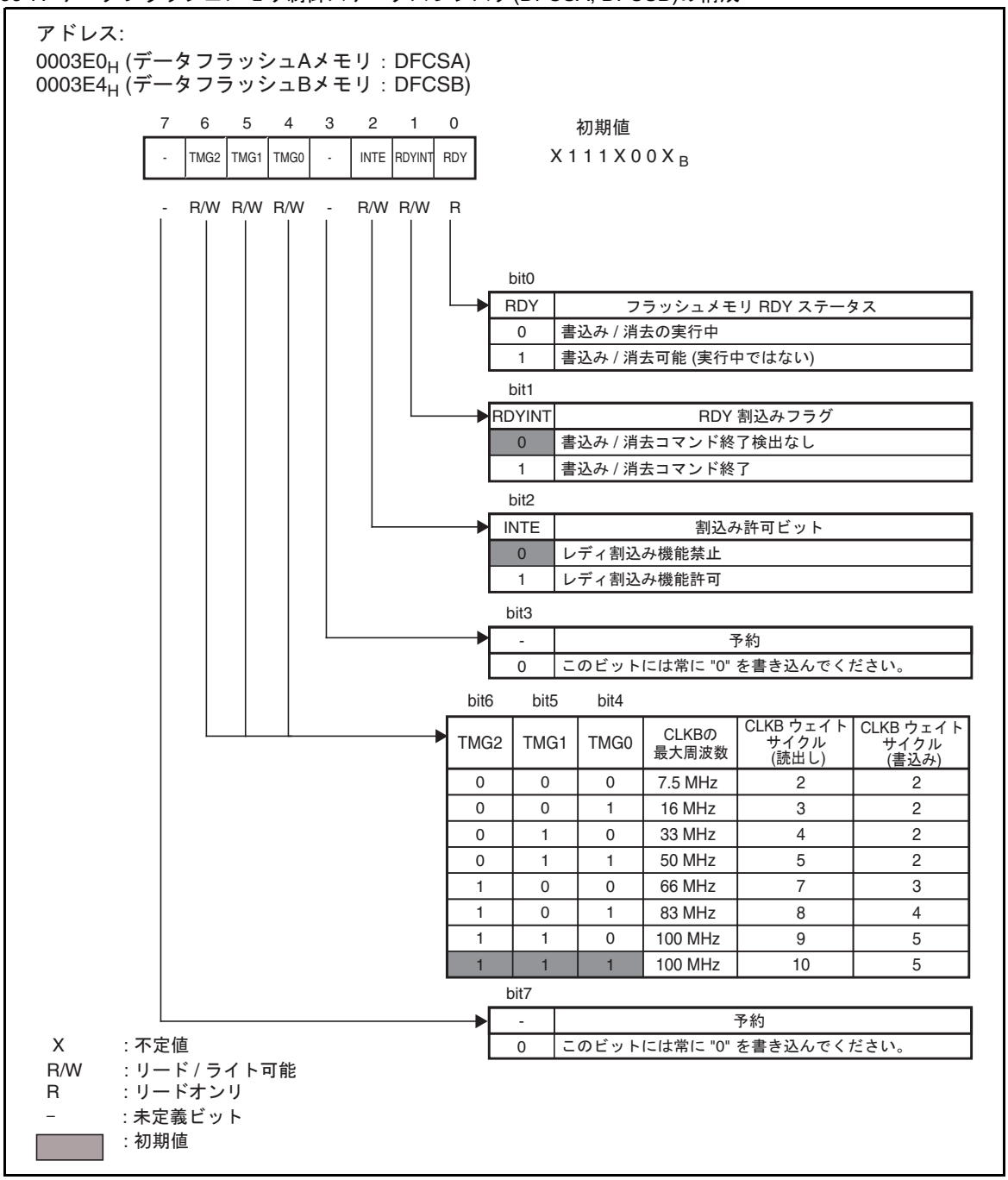


Table 33-6. データフラッシュメモリ制御ステータスレジスタ(DFCSA, DFCSB)の各ビットの機能

<table>
<tr><th colspan="2">ビット名</th><th>機能</th></tr>
<tr><td>bit 0</td><td>RDY:
フラッシュメモリ RDY
ステータス</td><td>• フラッシュメモリ RDY 出力の状態を示します。(サンプリングされた値)
• このビットへの書込みは , 動作に影響を与えません。
• "0" は , プログラム / 消去コマンドが現在実行されていることを示します。
• "1"は, プログラム/消去コマンドが現在実行されていないか, フラッシュメモリが消去一時停止状態にあることを示します。</td></tr>
<tr><td>bit 1</td><td>RDYINT:
RDY割込みフラグ</td><td>• フラッシュメモリレディ割込み機能の割込みフラグ
• このビットはリセットにより "0" に初期化されます。
• RDY 信号の立上りにより "1" に設定されます。
• RDYINT ビットは "0" 書込みによりクリアされます。"1" 書込みは無効です。
• リードモディファイライトアクセスの読出しでは, 常に"1"が読み出されます。</td></tr>
<tr><td>bit 2</td><td>INTE:
割込み許可ビット</td><td>• RDYINT 割込み機能 (書込み / 消去コマンド終了割込み) を許可
• このビットが "1" に設定された時 , RDYINT フラグが "1" であるか , RDY 信号の立上りによって RDYINT フラグが "1" にセットされると , 割込みが発生します。
• "0" に設定した場合 , RDYINT 割込みは発生しません。
• INTE ビットの初期値は "0" です。
• 読出しおよび書込み可能です。</td></tr>
<tr><td>bit 3</td><td>予約</td><td>• 常に "0" に設定してください。
• 読出し値は不定です。
• 本レジスタへのリードモディファイライト(RMW)系命令はこのビットには影響しません。</td></tr>
<tr><td>bit 4 ～ bit 6</td><td>TMG0 ～ TMG2:
タイミング設定ビット</td><td>• データフラッシュの読出し / 書込みタイミング (ウェイトサイクル) を制御します。
• これらのビットの初期値は "111" (ウェイトサイクルの最大数) です。
• 必要な設定は使用するCLKB周波数によって異なります。以下の表に従って設定を選択してください。
bit6 bit5 bit4
<table><tr><th>TMG2</th><th>TMG1</th><th>TMG0</th><th>CLKBの
最大周波数</th><th>CLKBｳｪｲﾄ
ｻｲｸﾙ(読出し)</th><th>CLKBｳｪｲﾄ
ｻｲｸﾙ(書込み)</th></tr><tr><td>0</td><td>0</td><td>0</td><td>7.5 MHz</td><td>2</td><td>2</td></tr><tr><td>0</td><td>0</td><td>1</td><td>16 MHz</td><td>3</td><td>2</td></tr><tr><td>0</td><td>1</td><td>0</td><td>33 MHz</td><td>4</td><td>2</td></tr><tr><td>0</td><td>1</td><td>1</td><td>50 MHz</td><td>5</td><td>2</td></tr><tr><td>1</td><td>0</td><td>0</td><td>66 MHz</td><td>7</td><td>3</td></tr><tr><td>1</td><td>0</td><td>1</td><td>83 MHz</td><td>8</td><td>4</td></tr><tr><td>1</td><td>1</td><td>0</td><td>100 MHz</td><td>9</td><td>5</td></tr><tr><td>1</td><td>1</td><td>1</td><td>100 MHz</td><td>10</td><td>5</td></tr></table>• 読出しおよび書込み可能</td></tr>
<tr><td>bit 7</td><td>予約</td><td>• 常に "0" に設定してください。
• 読出し値は不定です。
• 本レジスタへのリードモディファイライト(RMW)系命令はこのビットには影響しません。</td></tr>
</table>

33.5.2 データフラッシュメモリ書込みコマンドシーケンサ制御レジスタ(DFWCA, DFWCB)

これらのレジスタ (データフラッシュAメモリはDFWCA, データフラッシュBメモリはDFWCB) は, 書込みコマンドシーケンサのすべての機能を制御します。

データフラッシュメモリ書込みコマンドシーケンサ制御レジスタ(DFWCA, DFWCB)

Figure 33-8. DFWCAレジスタ, およびDFWCBレジスタの構成

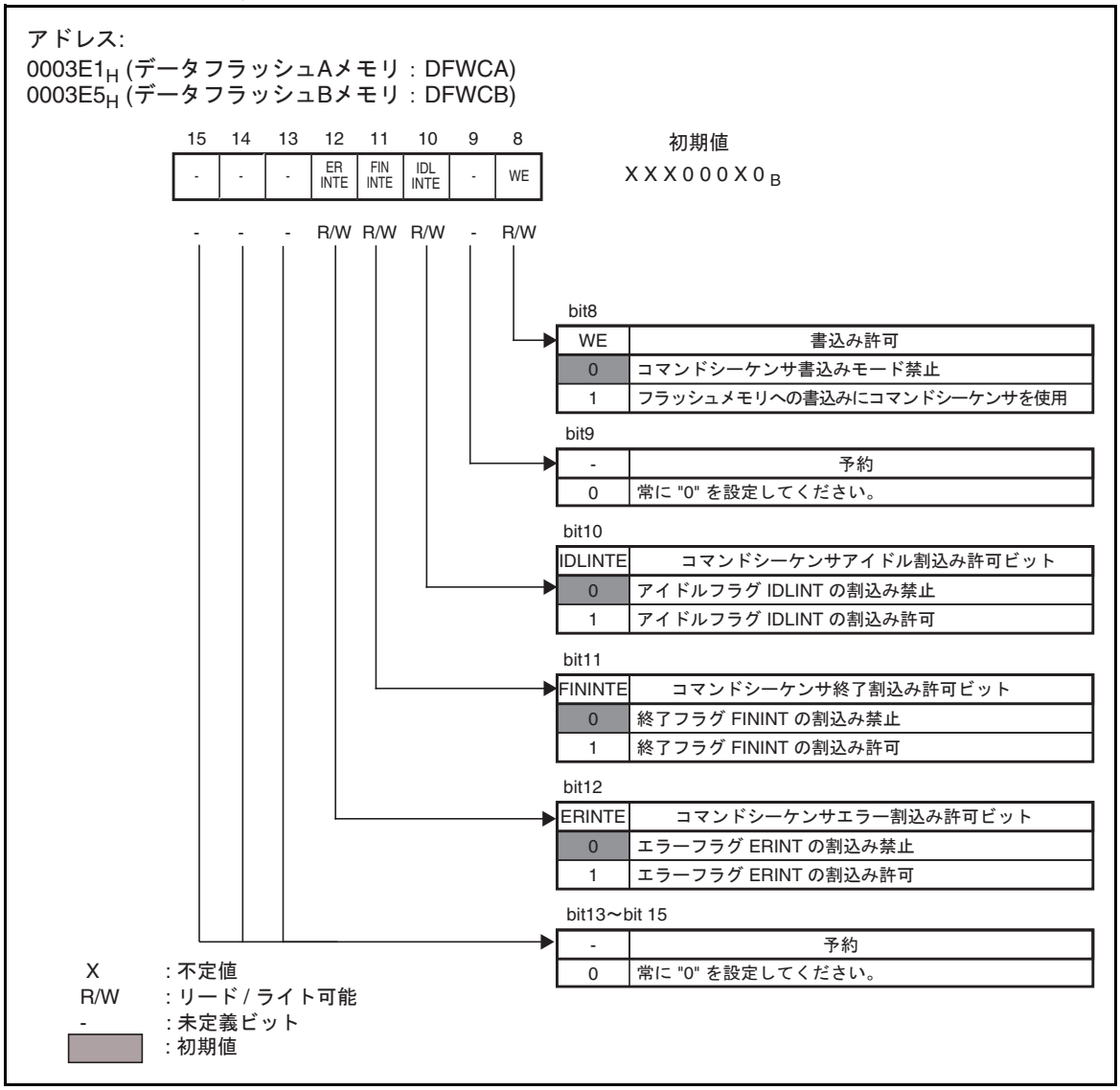


Table 33-7. データフラッシュメモリ書込みコマンドシーケンサ制御コマンドレジスタ(DFWCA, DFWCB)の各ビットの機能

ビット名		機能
bit 8	WE: 書込み許可ビット	• ライトアクセスモードを制御します。 • このビットを "1" に設定すると , 書込みコマンドシーケンサが起動します。データフラッシュメモリ領域への各 CPU ライトアクセスは , 対象のアドレスへの書込みアクセスとみなされ , 書込みコマンドシーケンス (Table 33-9 を参照) が自動的に生成されます。 • このビットを "0" に設定すると , 書込みコマンドシーケンサが禁止されます。書込みコマンドシーケンスはフラッシュメモリに手動で書き込む必要があります。またフラッシュメモリへほかのコマンド (チップ / セクタ消去 , セクタ消去一時停止 / 再開など) を送信するためには , このモードを使用する必要があります。 • 書込みコマンドシーケンサが書込み処理を実行中はこのビットを "0" に設定しないでください。 • このビットを設定してもフラッシュメモリ読出し動作には影響しません。 • このビットの初期値は"0" (直接書込みモード) です。
bit 9	予約	• このビットには常に "0" を書き込んでください。 • このビットの読出し値は不定です。 • 本レジスタへのリードモディファイライト(RMW)系命令はこのビットには影響しません。
bit 10	IDLINTE: アイドル割込み許可ビット	• データフラッシュメモリ書込みコマンドシーケンサのアイドル割込み (IDLINT) を許可します。 • このビットが "1" に設定され , IDLINT フラグが設定される (書込みコマンドシーケンサのアイドル状態への移行) とき , 割込みが発生します。 • "0" に設定した場合 , アイドル割込みは発生しません。 • 書込みコマンドシーケンサが許可されている場合 (WE="1") のみ , アイドル割込みが発生します。 • このビットの初期値は"0" (アイドル割込み禁止) です。
bit 11	FININTE: 終了割込み 許可ビット	• データフラッシュメモリ書込みコマンドシーケンサの終了割込み (FININT) を許可します。 • このビットが "1" に設定され , FININT フラグが設定される (書込み動作の正常終了) と , 割込みが発生します。 • "0" に設定した場合 , 終了割込みは発生しません。 • 書込みコマンドシーケンサが許可されている場合 (WE="1") のみ , 終了割込みが発生します。 • このビットの初期値は"0" (終了割込み禁止) です。
bit 12	ERINTE: エラー割込み許可ビット	• データフラッシュメモリ書込みコマンドシーケンサのエラー割込み (TOERINT, WERINT, WIERINT) を許可します。 • このビットが "1" に設定され , TOERINT, WERINT, WIERINT エラーフラグのいずれかが設定されると , 割込みが発生します。 • "0" に設定した場合 , エラー割込みは発生しません。 • 書込みコマンドシーケンサが許可されている場合 (WE="1") のみ , エラー割込みが発生します。 • このビットの初期値は"0" (エラー割込み禁止)です。
bit 13～bit 15	予約	• これらのビットには常に "0" を書き込んでください。 • これらのビットの読出し値は不定です。 • 本レジスタへのリードモディファイライト(RMW)系命令はこれらのビットには影響しません。

33.5.3 データフラッシュメモリ書込みコマンドシーケンスステータスレジスタ(DFWSA, DFWSB)

これらのレジスタ (データフラッシュAメモリはDFWSA, データフラッシュBメモリはDFWSB) には, 書込みコマンドシーケンサのすべてのステータスビットが含まれています。

データフラッシュメモリ書込みコメンドシーケンスステータスレジスタ(DFWSA, DFWSB)

Figure 33-9. DFWSAレジスタ, およびDFWSBレジスタの構成

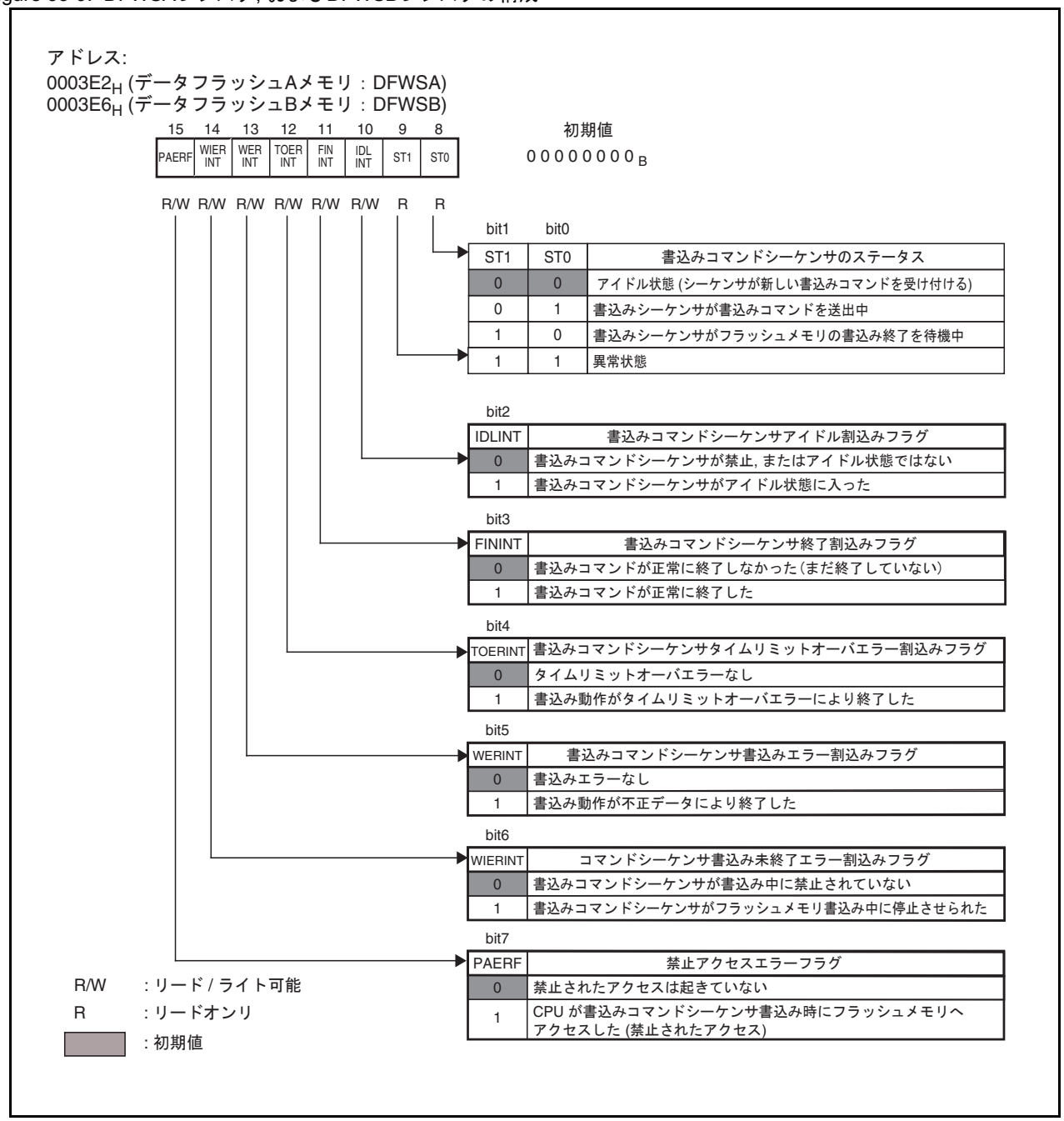


Table 33-8. データフラッシュメモリ書込みコマンドシーケンサステータスレジスタ(DFWSA, DFWSB)の各ビットの機能

<table>
<tr><th colspan="2">ビット名</th><th>機能</th></tr>
<tr><td>bit 0, bit 1</td><td>ST0, ST1:
ステータスビット</td><td>
• 書込みコマンドシーケンサのステータスを示します。
<table>
<tr><th colspan="2">bit1 bit0</th><th></th></tr>
<tr><td>ST1</td><td>ST0</td><td>Write command sequencer status</td></tr>
<tr><td>0</td><td>0</td><td>Idle (sequencer accepts new write command)</td></tr>
<tr><td>0</td><td>1</td><td>Sequencer is submitting a write command</td></tr>
<tr><td>1</td><td>0</td><td>Sequencer is waiting for Flash program finish</td></tr>
<tr><td>1</td><td>1</td><td>Illegal state</td></tr>
</table>
• ST[1:0] = 00 はシーケンサが無効になっている (WE= 0) か , アイドル状態である (DFWCx:WE=1) ことを示します。アイドル状態では , シーケンサは新しい書込みコマンドを受け付けることができます。
• ST[1:0] = 01 は , シーケンサがフラッシュメモリモジュールに書込みコマンドを送出していることを示します。
• ST[1:0] = 10 は , 書込みコマンドシーケンサがフラッシュメモリ RDY ステータスとフラッシュメモリハードウェアシーケンスフラグのチェックによって書込みコマンドの進行をチェックしていることを示します。
• ST[1:0]="11" は書込み異常状態で , フラッシュメモリの書込み ("01" あるいは "10") 中に , 書込みコマンドシーケンサが禁止された (DFWCx:WE="0") ことを示します。この状態が起きた時はリセット (ソフトウェアリセットなど) しなければなりません。
• 読出し専用です。
• これらのビットの初期値は "00" (書込みコマンドシーケンサ禁止) です。
</td></tr>
<tr><td>bit 2</td><td>IDLINT:
アイドル割込みフラグ</td><td>
• 書込みコマンドシーケンサのアイドル割込みフラグであり , DMA によるデータフラッシュメモリへのデータ転送を開始するために使用できます。
• リセットにより "0" に初期化されます。
• コマンドシーケンサが許可されるか , アイドル状態になるとき (書込み動作終了後) に "1" に設定されます。
• IDLINT ビットは "0" 書込みでクリアできます。"1" 書込みは無効です。
• DMA によってクリアすることもできます。
• リードモディファイライトアクセスの読出しでは常に "1" が読み出されます。
• このビットが"1"に設定され, 対応する割込み許可ビットDFWCx:IDLINTEが"1"の場合, 割込みが発生します。
</td></tr>
<tr><td>bit 3</td><td>FININT:
終了割込みフラグ</td><td>
• 書込みコマンドシーケンサの終了割込みフラグであり , CPU によるフラッシュメモリ書込みに使用できます。
• リセットにより "0" に初期化されます。
• コマンドシーケンサが書込みコマンドを正常に終了すると "1" に設定されます。
• FININT ビットは "0" 書込みでクリアできます。"1" 書込みは無効です。
• DMA によってクリアすることもできます。
• リードモディファイライトアクセスの読出しでは常に "1" が読み出されます。
• このビットが"1"に設定され, 対応する割込み許可ビットDFWCx:FININTEが"1"の場合, 割込みが発生します。
</td></tr>
<tr><td>bit 4</td><td>TOERINT:
タイムリミットオーバ割込みフラグ</td><td>
• 書込みコマンドシーケンサのタイムリミットオーバエラー割込みフラグです。
• リセットにより "0" に初期化されます。
• コマンドシーケンサがタイムリミットオーバ状態 (DQ5 フラグ = 1) で終了する場合 , "1" に設定されます。
• TOERINT ビットは "0" 書込みでクリアできます。"1" 書込みは無効です。
• リードモディファイライトアクセスの読出しでは常に "1" が読み出されます。
• このビットが"1"に設定され, エラー割込み許可ビットDFWCx:ERINTEが"1"の場合, 割込みが発生します。この場合, DMA停止要求も送信されます。
</td></tr>
<tr><td>bit 5</td><td>WERINT:
書込みエラー割込みフラグ</td><td>
• 書込みコマンドシーケンサの書込みエラー割込みフラグです。
• リセットにより "0" に初期化されます。
• 正常なデータが書き込まれずに書込みコマンドが終了するとき (たとえば, 消去を一時停止されたセクタへの書込み試行後), "1" に設定されます。
• WERINT ビットは "0" 書込みクリアできます。"1" 書込みは無効です。
• リードモディファイライトアクセスの読出しでは常に "1" が読み出されます。
• このビットが"1"に設定され, エラー割込み許可ビットDFWCx:ERINTEが"1"の場合, 割込みが発生します。この場合, DMA停止要求も送信されます。
</td></tr>
</table>

Table 33-8. データフラッシュメモリ書込みコマンドシーケンサステータスレジスタ(DFWSA, DFWSB)の各ビットの機能

ビット名		機能
bit 6	WIERINT: 書込み未終了割込みフラグ	• 書込みコマンドシーケンサの書込み未終了エラー割込みフラグです。 • リセットにより "0" に初期化されます。 • コマンドシーケンサがアイドル状態でなかった期間にコマンドシーケンサが (WE= 0 書込みにより) 禁止される場合に , "1" に設定されます。この場合 , 書込みシーケンスのステータスを手動でチェックする必要があります。 • もし , コマンドシーケンサが異常状態のままであった場合は , リセットを掛けてください。 • WIERINT ビットは "0" 書込みでクリアできます。"1" 書込みは無効です。 • リードモディファイライトアクセスの読出しでは常に "1" が読み出されます。 • このビットが"1"に設定され, エラー割込み許可ビットDFWCx:ERINTEが"1"の場合, 割込みが発生します。この場合, DMA停止要求も送信されます。
bit 7	PAERF: 禁止アクセスフラグ	• 禁止アクセスを示すエラーフラグです。 • リセットにより "0" に初期化されます。 • 書込みコマンドシーケンサがアイドル状態でない期間に CPU がフラッシュメモリの読出しまたは書込みを行うと , "1" に設定されます。このような禁止されたライトアクセスは無視され , リードアクセスでは 00_H が返されます。 • PAERF ビットは "0" 書込みでクリアできます。"1" 書込みは無効です。 • リードモディファイライトアクセスの読出しでは常に "1" が読み出されます。 • このビットは割込みを発生させることができません。

33.6 フラッシュメモリ自動アルゴリズムの起動

フラッシュメモリの書込み/消去は, フラッシュメモリの自動アルゴリズムの起動で実行されます。読出し/リセット, 書込み, チップ消去およびセクタ消去コマンドが利用可能です。消去一時停止と再開の機能は,セクタ消去の間, 利用可能です。

コマンドシーケンス表

フラッシュメモリの書込み / 消去の自動アルゴリズムは Table 33-9 のように , 1 バイト～ 6 バイトまたはワードの連続した書込みで起動されます。表はプログラム / データフラッシュ (メインフラッシュ) およびデータフラッシュに有効です。
コマンドのデータは , 下位バイト (書込みコマンドのプログラムデータ PD を除く) に書き込んでください。コマンドは偶数アドレスに書き込んでください。
プログラム / データフラッシュ (メインフラッシュ) へのコマンド書込みは , バイト命令またはワード命令が使用可能です。ワード命令の場合 , 上位バイトに書かれたデータは , 無視されます。
書込みコマンドでは , 4 番目のバス書込みサイクルに使用される書込み命令 (プログラムアドレス PA とプログラムデータ PD) のデータ幅がフラッシュメモリの書込みビット幅 (バイトまたはワード) を決定します。

データフラッシュは , コマンドをバイト命令で書き込んでください。書込みコマンドを使用するとき , バイト単位のプログラミングだけが可能です。
データフラッシュは書込みコマンドシーケンサを持っており , フラッシュへの書込みシーケンスを自動的に送信できます。書込みコマンドシーケンサはワードプログラミングもサポートします。
以下のコマンドを実行するために書込みコマンドシーケンサ (DFWCx:WE= 0) を禁止にしてください。

Table 33-9. コマンドシーケンス表

コマンドシーケンス	バス書込みアクセス	第1バス書込みサイクル		第2バス書込みサイクル		第3バス書込みサイクル		第4バス書込みサイクル		第5バス書込みサイクル		第6バス書込みサイクル	
		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
読出し / リセット *1	1	mmmXXX (偶数)	F0	-	-	-	-	-	-	-	-	-	-
読出し / リセット *1	3	mmmAAA	AA	mmm554	55	mmmAAA	F0	-	-	-	-	-	-
書込み	4	mmmAAA	AA	mmm554	55	mmmAAA	A0	PA	PD	-	-	-	-
チップ消去	6	mmmAAA	AA	mmm554	55	mmmAAA	80	mmmAAA	AA	mmm554	55	mmmAAA	10
セクタ消去	6	mmmAAA	AA	mmm554	55	mmmAAA	80	mmmAAA	AA	mmm554	55	sssXXX (偶数)	30
セクタ消去一時停止	1	sssXXX (偶数)	B0										
セクタ消去再開	1	sssXXX (偶数)	30										
連続モードに設定	3	mmmAAA	AA	mmm554	55	mmmAAA	20h						
連続書込み *2	2	mmmXXX (偶数)	A0	PA	PD								
連続モードリセット*2	2	mmmXXX (偶数)	90	mmmXXX (偶数)	F0/00								

＜注意事項＞

*1: 2 種類の読出し / リセットコマンドは , タイムリミットオーバ状態からフラッシュメモリを読出しモードにリセットすることができます。2 つのコマンドはソフトウェアの互換性のためだけに提供されます。読出し / リセットコマンドによって連続モードがリセットされることはありません。

*2: このコマンドは連続モードでのみ有効です。

- PA: 書込みアドレスです

ワードモードで書き込むとき , 偶数アドレスしか指定できません。また , バイトモードにおいては , 奇数アドレスが指定できます。

- PD: 書込みデータです

データフラッシュにはバイト命令しか使用できません。

- 表中のアドレスは, CPUメモリマップ上の値です。アドレスおよびデータはすべて16進数表記です。「X」は任意の値です。
- コマンド書込み時, CPU/フラッシュアドレスビット1は無視されます(プログラムアドレスPAは除く)。したがって, 例えばアドレスAAAまたはAA8にコマンドを書き込むこともできます。
- 表中のアドレス"mmm"は, コマンドを送るフラッシュメモリモジュール(フラッシュA/Bまたはデータフラッシュ A/B)を示します。
- 表中のアドレス"sss"はセクタ消去コマンドを送るフラッシュメモリセクタを示します。フラッシュメモリの「33.2 フラッシュメモリのセクタ構成」を参照してください。
- 不正なアドレスにデータを書き込んだり, 誤った順番で書き込むと, フラッシュメモリは読出し/リセット状態に戻ります。
- コマンドの書込みサイクルの間でフラッシュメモリからデータを読出し可能です。コマンド実行は最後の書込みサイクルの後に始まります。
- 対応する書込み許可信号(MCSRA:WE, MCSRB:WE)が"1"にセットされるときだけ, プログラム/データフラッシュ(メインフラッシュ)にコマンドを書き込むことができます。
- コマンドは書込み保護されているセクタに書き込まないでください。書込み保護されているセクタへの書込みアクセスは対応するフラッシュモジュールをハードウェアリセットし, RDYフラグは64CLKBの間"0"にセットされます。ハードウェアリセットは前状態(プログラム, 消去, 一時停止)にかかわらず読出し/リセット状態にフラッシュメモリを初期化します。

フラッシュメモリの状態遷移

■ フラッシュメモリの状態遷移図

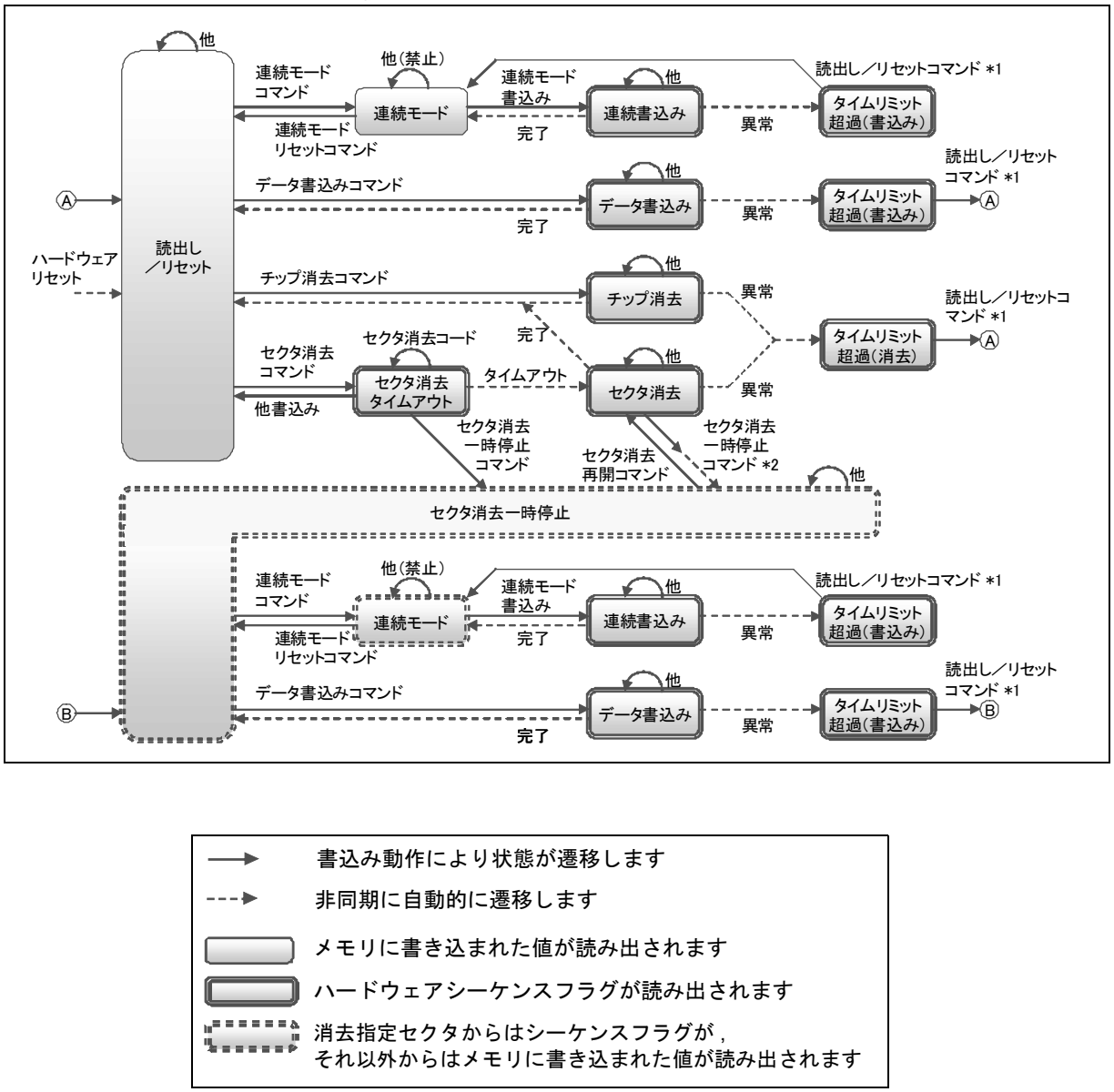
Figure 33-10 にフラッシュメモリの状態遷移図を示します。

リセットにより , フラッシュメモリは読出し / リセット状態になります。この状態がフラッシュメモリの基本の状態で , ここから Table 33-9 のシーケンスにしたがって書込みを行うことでコマンドが認識されていきます。

各状態間の遷移は書込みによって同期遷移するものと , 自動アルゴリズムの実行 , 完了 , 異常検出に伴って非同期で遷移するものに分けられます。

フラッシュメモリの状態はハードウェアシーケンスフラグであらわされ, これをポーリングすることで非同期の状態遷移を検出することができます。

Figure 33-10. フラッシュメモリの状態遷移図



＜注意事項＞

*1: 本製品では任意の書込みでタイムリミット超過状態から抜け出しますが，将来にわたって互換性を確保するため，読出し / リセットコマンドを使用してください。

*2: セクタ消去一時停止コマンド発行後，最大 20 µs の遅延をもって一時停止することを表します。(タイムアウト期間中の一時停止は直ちに一時停止します。)

- スタンバイへの移行は読出し / リセット状態からのみ行ってください。

■ フラッシュメモリの状態と, その時に受け付けるコマンド

Table 33-10. フラッシュメモリの状態と、その時に受け付けるコマンド

状態＼コマンド		読出し／リセット	データ書込み	チップ消去	セクタ消去	消去一時停止	連続モード	連続データ書込み	連続モードリセット
読出し／リセット		×	○	○	○	×	○	×	×
データ書込み		×	×	×	×	×	×	×	×
チップ消去		×	×	×	×	×	×	×	×
セクタ消去	タイムアウト	*1	*1	*1	○*2	○	*1	*1	*1
	タイムアウト後	×	×	×	×	○	×	×	×
セクタ消去一時停止		×	○	×	○*2	×	○	○	○
タイムリミット超過	データ書込み（含連続モード）	○	*3	*3	*3	*3	*3	*3	*3
	チップ消去／セクタ消去	○	*3	*3	*3	*3	*3	*3	*3
連続モード		×	禁止	禁止	禁止	×	×	○	○

○: 状態遷移を起こします
×: 状態は変わりません

＜注意事項＞

*1: タイムアウト中は○印以外の書込みは読出し / リセットへ遷移

*2: セクタ消去コード (30_H) のみを認識

*3: 本製品では任意の書込みでタイムリミット超過状態から抜け出しますが，将来にわたって互換性を確保するため，読出し / リセットコマンドを使用してください。

33.7 自動アルゴリズム実行状態の確認

フラッシュメモリは自動アルゴリズムで書込み/消去シーケンスを実行します。フラッシュメモリには, その内部動作を終了したかを外部に知らせるためのハードウェアがあります。

ハードウェアシーケンスフラグ

自動アルゴリズム実行中は, 目的のフラッシュメモリへの読出しアクセスはフラッシュメモリデータの代わりにハードウェアシーケンスフラグを返します。それらは書込み / 消去動作の現在の状態をチェックするのに使用できます。

ハードウェアシーケンスフラグは DQ7, DQ6, DQ5, (DQ4), DQ3, DQ2 ビットで構成されます。これらのビットの機能を以下に示します。

データポーリングフラグ (DQ7)

トグルビットフラグ (DQ6)

タイミングリミット超過フラグ (DQ5)

セクタ消去状態フラグ (DQ4): データフラッシュのみ可能

セクタ消去タイマフラグ (DQ3)

トグルビット 2 フラグ (DQ2)

Table 33-11. ハードウェアシーケンスフラグのビット割当て

ビット番号	7	6	5	4	3	2	1	0
ハードウェアシーケンスフラグ	DQ7	DQ6	DQ5	(DQ4)	DQ3	DQ2	-	-

＜注意事項＞

ほかのビット ([15:8], (4), [1:0]) の値は未定義です。ソフトウェアは , これらのビットがどんな値でも対応できるように記述してください。

フラッシュメモリの自動アルゴリズムは CPU とは非同期に動作します。このため , ハードウェアシーケンスフラグもまた CPU 動作とは非同期に変化します。

このため , 読出し値の変化はビットごとにスキューをもち , 自動アルゴリズムが状態遷移する瞬間の読み出しは不定値となる場合があります。(たとえば , データ書込み終了時 , あるビットはハードウェアシーケンスフラグ , その他はメモリ内容の読出し値が混在します。) このため , 自動アルゴリズムの状態の検出 (ポーリング) は本マニュアルで推奨する手順で行ってください。

対応する書込み許可信号 (MCSRA:WE, MCSRB:WE) の設定にかかわらずプログラム / データフラッシュ (メインフラッシュ) のハードウェアシーケンスフラグを読み出すことができます。

プログラム / データフラッシュ (メインフラッシュ) のハードウェアシーケンスフラグを読み出すためには , 以下の制限を守る必要があります。

- バイトデータ書込み時は上位バイト(奇数アドレス)に対するバイト書込みであっても, ハードウェアシーケンスフラグの読出しは, 下位バイト(偶数アドレス)から行なってください。
- データ読み出しバッファを無効にしてください。(MCSRA/B:DRBE=0)
- ハードウェアシーケンスフラグを2回連続して読み出す際, 読出し命令間に少なくとも1つのほかの命令 (少なくとも1つのNOP) を実行してください。さもなければ, DQ6とDQ2トグルビットを正しく読み出すことができません。

- ハードウェアシーケンスフラグを読み出す場合, MTCRA/MTCRB:ADSビットを"1"に設定してください。 (すべての書込み用のフラッシュタイミング推奨設定でADSを"1"に設定してください)
- この設定を行なわないと, DQ2トグルビットは, 正しく読み出しできず, またセクタ消去一時停止中のデータリードが正しく動作しません。
- ハードウェアシーケンスフラグの読出しの間, フラッシュメモリ制御レジスタ($0003FF_H$～$0003F0_H$)にアクセスしないでください。さもなければ, 正しくDQ2トグルビットを読み出すことができません。

データフラッシュには , これらの規則は適用されません。

自動書込み / 消去が実行中であるかの判断は , ハードウェアシーケンスフラグを確認するか , 書込みの終了を示すフラッシュメモリ制御レジスタ (MCSRA/MCSRB または DFCSA/DFCSB) の RDY ビットを読み出すことで確認できます。ハードウェアシーケンスフラグは , より詳細な情報を示します。書込み / 消去の正常終了後は , フラッシュメモリは読出し / リセット状態に戻ります。書換え動作の前に , これらのフラグ (RDY かハードウェアシーケンスフラグ) の 1 つを確認してください。

さらに, ハードウェアシーケンスフラグは2回目以降のセクタ消去コード書込みが有効であるかの確認のために使用することもできます。

Table 33-12 は , ハードウェアシーケンスフラグ機能の一覧です。

Table 33-12. ハードウェアシーケンスフラグ機能

状態		アクセスアドレス	DQ7	DQ6	DQ5	(DQ4)	DQ3	DQ2
正常動作時	読出し/リセット	任意アドレス	DATA:7	DATA:6	DATA:5	DATA:4	DATA:3	DATA:2
	書込み (内蔵書込みアルゴリズム動作中)	書込みアドレス	$\overline{\text{DATA:7}}$	トグル	0	0	0	1
	セクタ消去タイムアウト (後続のセクタ消去要求待ち)	消去指定セクタ	1	トグル	0	0	0	トグル
		他セクタ						1
	チップ / セクタ消去 (内蔵消去アルゴリズム動作中)	消去指定セクタ (チップ消去時は全セクタ)	0	トグル	0	1: 一時停止不可 0:一時停止可	1	トグル
		他セクタ						1
	セクタ消去一時停止	消去指定セクタ	1	1	0	0	0	トグル
		他セクタ	DATA:7	DATA:6	DATA:5	DATA:4	DATA:3	DATA:2
	フラッシュが消去一時停止状態の間, 他セクタへ書込み	書込みアドレス	$\overline{\text{DATA:7}}$	トグル	0	0	0	1
	チップ/セクタ消去完了 (読出し/リセット状態)	消去セクタ	DATA:7 = 1	DATA:6 = 1	DATA:5 = 1	DATA:4 = 1	DATA:3 = 1	DATA:2 = 1
タイムリミット超過	書込み	書込みアドレス	$\overline{\text{DATA:7}}$	トグル	1	0	0	1
	チップ/セクタ消去	消去中にタイムアウトしたセクタ	0	トグル	1	不定	1	トグル
		他セクタ						1

各ハードウェアシーケンスフラグの機能は以降のセクションで説明します。

DQ4フラグはデータフラッシュのみ使用可能です。このビットの使い方を「33.8.6セクタ消去一時停止」で説明します。

33.7.1 データポーリングフラグ(DQ7)

データポーリングフラグ(DQ7)は, 自動アルゴリズム(書込みまたは消去)が実行中もしくは終了したかを示します。

33.7.1.1 データポーリングフラグ(DQ7)

Table 33-12 にそれぞれのフラッシュ状態での DQ7 フラグの機能を説明します。

書込み

DQ7 は読み出すアドレスには無関係に書込みデータ (PD) のビット 7 の反転データが読み出されます。
ポーリングにはデータ書込みアドレス (PA) を使用してください。
自動書込み終了後は , DQ7 はメモリ内容の読み出しとなり , 書き込まれたデータ (PD) のビット 7 が反転せずに読み出されることで自動アルゴリズムの終了検出を行うことができます。

ワード/バイトの書込み

ワード書込みコマンドは上位バイト (DATA[15:8]) と下位バイト (DATA[7:0]) にデータを書き込みます。DQ7 は DATA7 を反転させた値を示します。

バイト書込み時は DQ7 は使用できませんので , DQ6 を使用してください。
DQ7 を使用する場合にはワード書込みを行ってください。

チップ消去/セクタ消去

DQ7 は読み出すアドレスには無関係に "0" が読み出されます。
ポーリングには消去指定したセクタ (sss) 内の任意のアドレスを使用してください。
(チップ消去の場合はフラッシュメモリ領域の任意のアドレスです)。
自動消去が終了すると DQ7 はメモリ内容の読み出しとなり , 消去されたセクタの値である "1" が読み出されることで自動アルゴリズムの終了検出を行うことができます。

セクタ消去一時停止

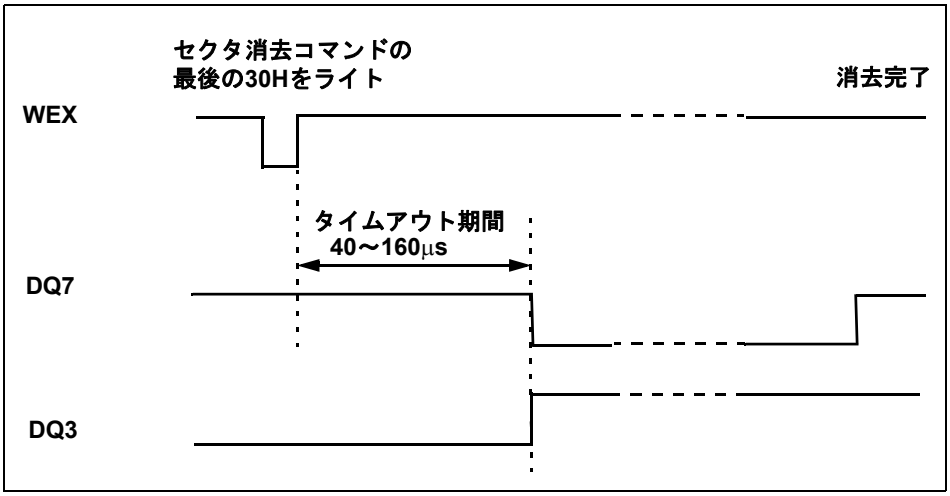
DQ7 は消去指定したセクタからは "1" が読み出されます。それ以外のセクタはメモリ内容の読み出しとなり , 書き込まれていたデータのビット 7 が読み出されます。

＜注意事項＞

- フラッシュメモリの自動アルゴリズムは CPU とは非同期に動作し , ハードウェアシーケンスフラグもまた CPU 動作とは非同期に変化します。

このため , 読出し値の変化はビットごとにスキューをもち , 自動アルゴリズムが状態遷移する瞬間の読出しは不定値となる場合があります。(たとえば , データ書込み終了時 , あるビットはハードウェアシーケンスフラグ , その他はメモリ内容の読出し値が混在します。) このため , 自動アルゴリズムの状態の検出 (ポーリング) は本マニュアルで推奨する手順で行ってください。

- バイトデータ書込み時は DQ7 でのポーリングは使用せず , DQ6 を使用してください。
- セクタ消去時には制限事項があります。以下のいずれかの方法を使用してください。
  - セクタ消去コマンド発行後 , 160µs 待ってから DQ7 ポーリングを開始
  - セクタ消去コマンド発行後 , DQ3=1 (セクタ消去タイムアウト期間終了) を待ってから DQ7 ポーリングを開始
  - DQ6 (トグルビット) をポーリングに用いる



33.7.2 トグルビットフラグ(DQ6)

トグルビットフラグ(DQ6)は, データポーリングフラグ (DQ7)と同様に自動アルゴリズム(書込みまたは, 消去)が実行中もしくは終了したかを示します。

33.7.2.1 トグルビットフラグ(DQ6)

Table 33-12 にそれぞれのフラッシュ状態での DQ6 フラグの機能を説明します。

書込みとチップ消去/セクタ消去

DQ6 は読み出すアドレスには無関係に読み出しごとに "1" と "0" の値を交互に示すトグル動作をします。
ポーリングにはフラッシュメモリ領域の任意のアドレスを使用できますが , 連続する 2 回の読出しには同一アドレスを使用してください。
自動書込み終了後は , DQ6 はメモリ内容の読み出しとなり , 読み出すアドレスのデータのビット 6 が続けて読み出され , トグルが停止することで自動アルゴリズムの終了検出を行うことができます。

セクタ消去一時停止

DQ6 は消去指定したセクタからは "1" が読み出されます。それ以外のセクタはメモリ内容の読出しとなり , 書き込まれていたデータのビット 6 が読み出されます。

＜注意事項＞

- フラッシュメモリの自動アルゴリズムは CPU とは非同期に動作し , ハードウェアシーケンスフラグもまた CPU 動作とは非同期に変化します。このため , 読出し値の変化はビットごとにスキューをもち , 自動アルゴリズムが状態遷移する瞬間の読出しは不定値となる場合があります。(たとえば , データ書込み終了時 , あるビットはハードウェアシーケンスフラグ , その他はメモリ内容の読出し値が混在します。)
  このため , 自動アルゴリズムの状態の検出 (ポーリング) は本マニュアルで推奨する手順で行ってください。
- 正しくプログラム / データフラッシュ (メインフラッシュ) の DQ6 フラグを読み出すために , 以下の制限を守る必要があります。

ハードウェアシーケンスフラグを 2 回連続して読み出す際 , 読出し命令間に少なくとも 1 つのほかの命令 (少なくとも 1 つの NOP) を実行してください。

データフラッシュには , この規則は適用されません。

33.7.3 タイミングリミット超過フラグ(DQ5)

タイミングリミット超過フラグ(DQ5) は, 自動アルゴリズムの実行がフラッシュメモリ内部に規定された時間を超える異常状態となったことを示します。

33.7.3.1 タイミングリミット超過フラグ(DQ5)

Table 33-12 にそれぞれのフラッシュ状態での DQ5 フラグの機能を説明します。

書込みとチップ消去/セクタ消去

DQ5 は読み出すアドレスには無関係に読み出されます。

自動アルゴリズムが正常に動作を継続している場合には "0", 何らかの異常が発生している場合には "1" が読み出されます。

弊社推奨条件においては , 通常 , このフラグがセットされることはありません。しかし , たとえば , 既に "0" が書き込まれているビットに "1" を書き込もうとしたとき
には , 通常は DQ5 が "1" になります *1 が , これはフラッシュメモリが不良なのではなく , 正しく使用されなかったことによるものです。読出し / リセットコマンドを発行してください。

DQ5 が "1" になる時間とデータシートの書込み / 消去時間の規格には関連がありません。たとえば , 上記の正しく使用されなかった例において , 最大データ書込み規格が 200 µs であったとき , DQ5 がデータ書込み開始後 200 µs で "1" になるわけではありません。自動アルゴリズムは規格に対して十分長い時間を継続するように設計されています。

セクタ消去一時停止

DQ5 には消去指定したセクタからは "0" が読み出されます。それ以外のセクタはメモリ内容の読出しとなり , 書き込まれていたデータのビット 5 が読み出されます。

＜注意事項＞

*1 : 0 から 1 へのデータを書き込んだ際 , まれに DQ5 が "1" にならないこともあります。"0" のビットに "1" を書き込むことで DQ5 が "1" になることを期待した動作は避けてください。

フラッシュメモリの自動アルゴリズムは CPU とは非同期に動作し , ハードウェアシーケンスフラグもまた CPU 動作とは非同期に変化します。
このため , 読出し値の変化はビットごとにスキューをもち , 自動アルゴリズムが状態遷移する瞬間の読出しは不定値となる場合があります。(たとえば , データ書込み終了時 , あるビットはハードウェアシーケンスフラグ , そのほかはメモリ内容の読出し値が混在します。)
したがって , もし DQ5 が "1" となった場合 , これが自動アルゴリズムが終了した後のメモリ内容の読出し値によるものか , それとも自動アルゴリズムの異常によるものなのかを判定しなければなりません。そのためにもう一度読出しを行い , DQ7 や DQ6 で自動アルゴリズムが継続しているかを確認してください。再度の読出しで自動アルゴリズムが継続している場合には異常が発生したと判定できます。

33.7.4 セクタ消去タイマフラグ(DQ3)

セクタ消去タイマフラグ(DQ3)は，セクタ消去コマンド起動後，セクタ消去タイムアウト期間中かどうかを示します。

33.7.4.1 セクタ消去タイマフラグ(DQ3)

セクタ消去自動アルゴリズムは , コマンドシーケンスのセクタ消去コードを書き込むと , セクタ消去タイムアウト期間を開始し , タイムアウト期間経過後に消去動作を開始します。タイムアウト期間は最短で 50 µs です。

セクタ消去タイマフラグ(DQ3) は, セクタ消去自動アルゴリズムがタイムアウト期間中であるかどうかを示します。この期間中には追加で消去セクタを指定できます。セクタ消去コードを書き込むたびにタイムアウト期間は再スタートします。

Table 33-12 にそれぞれのフラッシュ状態での DQ3 フラグの機能を説明します。

書込み

DQ3 は読み出すアドレスには無関係に "0" が読み出されます。

チップ消去

DQ3 は読み出すアドレスには無関係に "1" が読み出されます。

セクタ消去

DQ3 は読み出すアドレスには無関係にセクタ消去タイムアウト期間中は "0", タイムアウト期間経過後は "1" が読み出されます。

セクタ消去一時停止

セクタ消去一時停止中 , DQ3 は消去指定したセクタからは "0" が読み出されます。それ以外のセクタはメモリ内容の読出しとなり , 書き込まれていたデータのビット 3 が読み出されます。

＜注意事項＞

フラッシュメモリの自動アルゴリズムは CPU とは非同期に動作し , ハードウェアシーケンスフラグもまた CPU 動作とは非同期に変化します。

このため , 読出し値の変化はビットごとにスキューをもち , 自動アルゴリズムが状態遷移する瞬間の読出しは不定値となる場合があります。(たとえば , データ書込み終了時 , あるビットはハードウェアシーケンスフラグ , その他はメモリ内容の読出し値が混在します。)

33.7.5 トグルビット2フラグ(DQ2)

トグルビット2フラグ(DQ2)は, セクタが消去一時停止状態にあるかを示します。また, どのセクタが消去に指定されたか確認するために使用できます。

33.7.5.1 トグルビット2フラグ(DQ2)

トグルビット 2 フラグ (DQ2) は , チップ / セクタ消去の自動アルゴリズム実行中および消去指定したセクタをトグル動作によって示します。Table 33-12 にそれぞれのフラッシュ状態での DQ2 フラグの機能を説明します。

書込み

DQ2 は読み出すアドレスには無関係に "1" が読み出されます。

チップ消去

DQ2 は読み出すアドレスには無関係に読み出しごとに "1" と "0" の値を交互に示すトグル動作をします。

セクタ消去

DQ2 は消去指定したセクタからはトグル値 , それ以外のセクタでは "1" が読み出されます。

セクタ消去一時停止

セクタ消去一時停止中 , DQ2 は消去指定したセクタからトグル値が読みだされます。それ以外のセクタではメモリ内容の読出しとなり , 書き込まれていたデータのビット 2 が読み出されます。
DQ2 は DQ6 と組み合わせることでセクタ消去一時停止状態になったかどうかを判定することができます。
消去指定したセクタを連続して読み出すと , セクタ消去の自動アルゴリズム実行中は DQ6 も DQ2 もトグル動作をします。セクタ消去一時停止状態になると DQ6 はトグル動作をやめますが , DQ2 はトグル動作を継続します。

タイミングリミット超過

セクタ消去 / チップ消去の自動アルゴリズム実行中 , 何らかの異常が発生して DQ5 から "1" が読み出された状態で自動アルゴリズムが継続している場合 , 異常が発生したセクタからトグル値 , それ以外のセクタでは "1" が読み出されます。

＜注意事項＞

正しくプログラム / データフラッシュ (メインフラッシュ) メモリの DQ2 フラグを読み出すために以下の制限を守る必要があります。

- ハードウェアシーケンスフラグを 2 回連続して読み出す際 , 読出し命令間に少なくとも 1 つのほかの命令 (少なくとも 1 つの NOP) を実行してください。
- ハードウェアシーケンスフラグを読み出す場合 , MTCRA/MTCRB:ADS ビットを "1" に設定してください。(すべての書込み用のフラッシュタイミング推奨設定で ADS を "1" に設定してください)。
- ハードウェアシーケンスフラグを読み出す間は , フラッシュメモリ制御レジスタ (アドレス :$0003FF_H$ ～ $0003F0_H$) にアクセスしないでください。

データフラッシュには , これらの規則は適用されません。

33.8 フラッシュメモリの書込み / 消去の詳細説明

フラッシュメモリの読出し/リセット，書込み，チップ消去，セクタ消去，セクタ消去一時停止，セクタ消去再開のそれぞれの操作手順を説明します。

フラッシュメモリの書込み / 消去の詳細説明

コマンドシーケンスの発行により (「33.6 フラッシュメモリ自動アルゴリズムの起動」の Table 33-9 を参照) フラッシュメモリは , 読出し / リセット , 書込み , チップ消去 , セクタ消去 , セクタ消去一時停止 , セクタ消去再開の自動アルゴリズムを実行します。

コマンドシーケンスの書込みアクセスの間 , フラッシュメモリに対する書込みが , コマンドシーケンスで定義された書込みだけであれば , フラッシュメモリ外の領域に対する読出し / 書込みサイクルを実行できます。また , フラッシュメモリに対しても , コマンドシーケンス書込みアクセスの間 , 読出しは可能です。
自動アルゴリズム終了は , ハードウェアシーケンスフラグか , RDY フラグ (プログラム / データフラッシュの MCSRA:RDY/MCSRB:RDY, データフラッシュの DFCSA:RDY/DFCSB:RDY) のポーリングか割込みによって検出できます。正常終了時は , フラッシュメモリは , 読出し / リセット状態に戻ります。
フラッシュメモリの各操作は以下の節で説明します。

- 読出し/リセット状態の設定
- 書込みコマンドシーケンスでのデータを書込み
- 書込みコマンドシーケンサを使用したデータフラッシュへの書込み
- 全データ消去 (チップ消去)
- 任意データ消去 (セクタ消去)
- セクタ消去一時停止
- セクタ消去再開

33.8.1 読出し/リセット状態の設定

フラッシュメモリを読出し/ リセット状態にする手順について説明します。

フラッシュメモリを読出し/リセット状態にする

リセットおよびコマンドの正常終了により , フラッシュメモリは読出し / リセット状態 * に設定されます。

*： 読出し / リセットはフラッシュメモリの初期状態です。
この状態で受け付けられるコマンドは Table 33-10 を参照してください。

フラッシュメモリのリセット要因

フラッシュメモリのリセット要因を下記に示します。

- ハードウェアリセット
  - CPUリセット
    電源リセット, 外部端子リセット, ソフトウェアリセット, ウォッチドッグタイマリセット, クロック停止検出リセット
  - 書換え保護セクタへのライトアクセス

    ハードウェアリセットは, フラッシュメモリの内部状態によらず, 強制的にリセットします。
    データ書込み/ セクタ消去/ チップ消去中にハードウェアリセットが発生した場合, データ書込み中のアドレスや消去指定したセクタの内容は保証されません。セクタ/チップ消去することからやり直してください。

- コマンドリセット
  - 読出し/リセットコマンド
    読出し/ リセットコマンドは, 通常のデータ読出しには必要ではありません。
    読出し/ リセットコマンドは, コマンドが正常に終了しない(DQ5=1：タイムリミットオーバ発生) ときに, 自動アルゴリズムを初期化するために使用します。
    書込み/ 消去の通常動作中に読出し/ リセットコマンドを発行しても動作に影響を与えません。
    読出し/ リセットコマンドには,1 回の書込み動作と3 回の書込み動作の 2 通りのコマンドシーケンスがありますが, これらに本質的な違いはありません。ソフトウェア互換のためにの2 タイプのコマンドを提供します。

＜注意事項＞

書込み / セクタ消去 / チップ消去コマンドが認識され , 自動アルゴリズムが起動した後は , これを止める手段はハードウェアリセットしかありません。

セクタ消去では消去一時停止コマンドにより自動アルゴリズムを一時停止させることができます。しかし , これはあくまでも一時的なもので , 必ず再開コマンドを発行して自動アルゴリズムを最後まで実行する必要があります。

33.8.2 書込みコマンドシーケンスでのデータ書込み

書込みコマンドシーケンスを入力し, フラッシュメモリにデータを書き込む手順について説明します。

33.8.2.1 自動書込みアルゴリズムの開始

フラッシュメモリのデータ書込み自動アルゴリズムを起動するためには , コマンドシーケンス表 (「33.6 フラッシュメモリ自動アルゴリズムの起動」の Table 33-9 を参照) の書込みコマンドをフラッシュメモリへ連続して送ってください。4 番目のライト命令により目的のアドレスへの書込みが終了すると , 書込みの自動アルゴリズムが開始します。

アドレッシング方法

プログラム / データフラッシュ(メインフラッシュ) の書込みはバイト単位かワード単位で可能です。書込みデータサイクルにおいてワード書込み命令で書き込んだ場合 , 偶数アドレスを指定しなければなりません。
データフラッシュへのダイレクト書込みモード (DFWCA:WE=0/DFWCB:WE=0) は , バイト単位のみ可能です。
書込みはどのようなアドレスの順番でも , セクタの境界を超えても行えますが , 1 回の書込みコマンドによって書き込まれるデータは上記のように 1 バイトもしくは 1 ワードのみです。

データ書込み上の注意

書込みによってビットデータを "0" から "1" に戻すことはできません。ビットの値を変えないか , "1" のビットを "0" にすることだけができます。"0" を "1" に戻すことができるのは消去コマンドだけです。ビットデータ "0" に "1" を書き込むと , 通常はデータ書込みの自動アルゴリズムは終了せず , タイミングリミット超過フラグ (DQ5) によって異常状態が検出されます。
データ "0" に "1" を書き込んだ際 , まれに異常状態とならず , 自動アルゴリズムが正常終了する場合もあります。この場合でもリセット後に読み出すと "0" が保持されています。"0" のビットに "1" を書き込むことで異常状態となることを期待した動作は避けてください。
自動書込みアルゴリズム実行中は , すべてのコマンドが無視されます。書込み中に CPU リセット (電源リセット , 外部リセット , ソフトウェアリセット , ウォッチドッグリセット , クロック停止リセット) が発生した場合 , および , 書込み保護セクタへの書込みアクセスは , フラッシュメモリハードウェアリセットし , 書込み動作をキャンセルします。この場合は書込みアドレスのデータは保証されません。

33.8.2.2 連続モード

連続モードは , フラッシュメモリへの複数のデータをプログラムするのに使用できます。フラッシュメモリへの「連続モードへの設定」コマンドの書込み後 , 通常の 4 サイクルの書込みコマンドの代わりに 2 サイクルの連続書込みコマンドで送信することで , データのプログラムが可能です。
読出し / リセットコマンドは連続モード設定に影響を与えません。
連続モードを終了するには " 連続モードリセットコマンド " を使用します。
連続モード中は消去コマンドは使用しないでください。

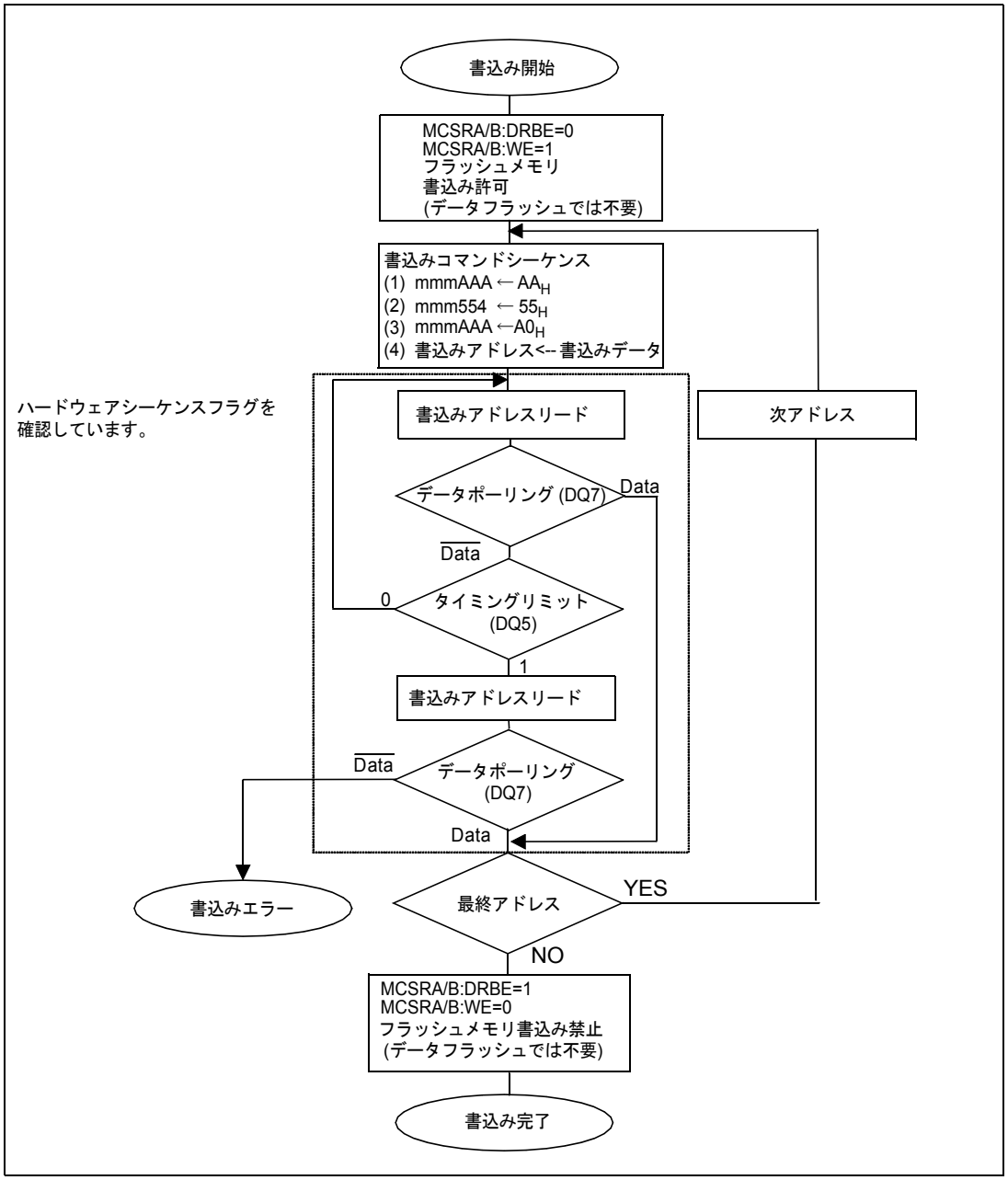
33.8.2.3 フラッシュメモリへの書込み例

Figure 33-11 は , フラッシュメモリへの書込み手順の例を示しています。ハードウェアシーケンスフラグ (「33.7 自動アルゴリズム実行状態の確認」を参照してください。) は , フラッシュメモリの自動アルゴリズムの状態を判定するのに使用できます。本例では , データポーリングフラグ (DQ7) を , 書込みの終了を確認するのに使用しています。
書き込もうとするアドレスでデータポーリングを実行してください。さもなければ , 書込みコマンドの終了にお

ける DQ7 フラグの変化を検出できません。

ハードウェアシーケンスフラグの変化は CPU 動作に非同期かつビット間スキューがありますので , タイミングリミットオーバフラグ (DQ5) が "1" を示したあとに , 再度 DQ7 を確認する必要があります。
トグルビットフラグ (DQ6) も同様です。ポーリングに DQ6 を用いた場合には DQ5 が "1" を示したあとに , 再度 DQ6 を確認してください。

Figure 33-11. フラッシュメモリ書込み手順の例



＜注意事項＞

アドレス "mmm" は , コマンドが動作するフラッシュメモリモジュールを示します。

33.8.3 書込みコマンドシーケンサを使用したデータフラッシュへの書込み

データフラッシュに書込みコマンドシーケンサを使用して書込む方法を説明します。

33.8.3.1 書込みコマンドシーケンサの使い方

データフラッシュへ書込みには , 前章の , コマンドシーケンスの書込みによるダイレクトモード (DFWCA:WE='0'/DFWCB:WE='0') と , 書込みコマンドシーケンサ使用 (DFWCA:WE='1'/DFWCB:WE='1') の 2 つの手段があります。書込みコマンドシーケンサ起動後 , データフラッシュへの書込みアクセスはすべて , フラッシュメモリへの書込み要求として扱われます。書込みコマンドシーケンサはライト命令のデータの前に書込みコマンドシーケンスを付加してデータフラッシュに送ります。
DFWCA/B:WE が "1" の間 , フラッシュメモリにほかのコマンドは送ってはいけません。

書込みコマンシーケンサの活性/非活性

リセット後は , 書込みコマンドシーケンサは停止状態です。DFWCA/DFWCB:WE を "1" に設定することによって , 動作可能な状態にします。コマンドシーケンサの起動は , データフラッシュが読出し / リセット状態か , または , セクタ消去一時停止状態でなければなりません。(ダイレクトモードで有効なコマンドシーケンスを 1 サイクルでも書いてしまった場合は読出し / リセット状態とはみなしません)

書込みコマンドシーケンサ停止は , アイドル状態 (DFWSA/DFWSB:ST[1:0]=00) のときに行ってください。

アドレス指定

書込みコマンドシーケンサモードでは , バイト単位かワード単位での書込みが可能です。ワード単位で書き込む場合は , 偶数アドレスを指定してください。書込みコマンドシーケンサはワードデータを , 2 回のバイト書込みとして処理します。

書込みコマンドシーケンサ動作状態でのフラッシュメモリの書込みと読出し

データフラッシュは書込みコマンドシーケンサがアイドル状態 (DFWSA/B:ST[1:0]=00) のときに書込み / 読出しができます。この場合 , 書込みコマンドシーケンサが停止状態としてバイトモード , ワードモードでの読出しが可能です。書込みコマンドシーケンサがアイドル状態でない場合は , フラッシュメモリへの読出しアクセスは実行されません。00_H が読まれ , アクセス禁止エラーフラグ (DFWSA/DFWSB:PAERF) をセットします。

書込みコマンドシーケンサがアイドル状態以外の場合 , フラッシュメモリへの書込みアクセスは無視され , アクセス禁止エラーフラグがセットされます。書込みコマンドシーケンサ動作時 , データフラッシュへ読出し / 書込みする前には , かならず書込みコマンドシーケンサステータスビット (DFWSB:ST[1:0]) を確認してください。

33.8.3.2 書込みコマンドシーケンサの割込み機能とDNA アクセス

書込みコマンドシーケンサは一度に 1 つの書込みコマンドしか扱うことができません。書込み動作中 , 2 つ目の書込み要求は無視されます。書込み動作の進捗は書込みコマンドシーケンサステータスビットを読出しか , 割込み機能を使用することによってチェックできます。DMA 機能を使用することによって , メモリブロックを転送できます。書込みエラーは 3 種のエラー割込みフラグで示されます。

IDLINT: アイドル割込み (主にDMAによる書込み)

DFWSA/B: IDLINT フラグは書込みコマンドシーケンサが新しい書込みデータを受け取る準備ができると , セットされます。これは書込みコマンドシーケンサを可能にした (DFWCA/B:WE を "1" にセット) 後と , コマンドシーケンサが書込み動作を終了したときになります。このフラグの割込みと DMA 機能は対応する DFWCA/B: IDLINTE によって設定してください。

このフラグはデータフラッシュへの DMA 転送を起動するのに使用できます。書込みコマンドシーケンサの起動前に , DMA と割込みを設定しておいてください。DFWCx:WE または DFWCx:IDLINTE ビットを "1" に設定すると , DMA 動作を開始する最初の割込みが発生します。"0" の書込みと DMA クリア機能によって IDLINT フラグをクリアできます。

FININT: 終了割込み (主に CPU による書込み)

DFWSA/B: FININT フラグは書込みコマンドが正常終了した後に設定されます。このフラグは , 書込みコマンドの結果と書込みコマンドシーケンサが次のデータを書き込めるようになったことを CPU へ通知するために使用できます。このフラグの割込み機能は対応する DFWCA/B: FININTE ビットによって設定してください。"0" の書込みと DMA クリア機能によって , FININT フラグをクリアできます。

エラー割込み(タイムアウト, 書込みエラー, 書込み未完了)

書込みコマンドシーケンサを使用時の書込み動作の異常終了は , 以下のエラーフラグの 1 つがセットされることで示されます。

- タイムアウト割込みフラグ(DFWSA/B:TOERINT)書込み中にタイミングリミット超過フラグ(DQ5)がセットされると, このエラーフラグがセットされます。
- 書込みエラー割込みフラグ(DFWSA/B:WERINT)正しいデータが書き込まれずに書込みコマンドが終了した場合(例として消去一時停止セクタへの書込みの後), このエラーフラグはセットされます。
- 書込み未完了エラー割込みフラグ(DFWSA/B:WIERINT):書込みシーケンス実行中(DFWSA/B:ST[1:0]=01または10)に書込みコマンドシーケンサが停止されると, このフラグはセットされます。

これらの 3 つのフラグの割込み機能は対応する DFWCA/B:ERINTE ビットによって許可されます。また , エラー割込み時には同時に DMA 停止要求も送信されます。

33.8.4 全データ消去(チップ消去)

チップ消去コマンドを入力し, フラッシュメモリの全データを消去する手順について説明します。

フラッシュメモリモジュールの全データ消去(チップ消去)

フラッシュメモリモジュールからすべてのデータを消去するためには , コマンドシーケンス表 (「33.6 フラッシュメモリ自動アルゴリズムの起動」の Table 33-9 を参照) のチップ消去コマンドを対象となるフラッシュメモリモジュールへ連続して送ってください。

チップ消去コマンドは 6 回のバス動作で行われます。6 サイクル目の書込みが完了した時点でチップ消去動作を開始します。

33.8.5 任意データ消去（セクタ消去）

セクタ消去コマンドを入力し,フラッシュメモリの任意のデータを消去(セクタ消去)する手順について説明します。セクタごとの消去が可能であり, 複数のセクタを同時に指定することもできます。

33.8.5.1 セクタ消去自動アルゴリズムの起動

フラッシュメモリの任意のセクタを消去するためには , コマンドシーケンス表 (「33.6 フラッシュメモリ自動アルゴリズムの起動」の Table 33-9 を参照) のセクタ消去コマンドをフラッシュメモリの対象セクタへ連続して送ってください。

セクタ指定方法

セクタ消去は , 対象セクタにセクタ消去コマンド (6 回の書込み動作) を実行することによって , 開始します。対象セクタの偶数アドレスに最後のコマンド (30_H) を書込んだ後に , セクタ消去タイムアウト時間が最短で 50µs 発生します。複数のセクタ消去には , この待ち時間の中に次の対象セクタへ消去コード 30_H(コマンドシーケンスの 6 番目のサイクル) を書き込みます。5 番目までのセクタ消去コマンドを書き込む必要はありません。

複数のセクタを指定する場合の注意

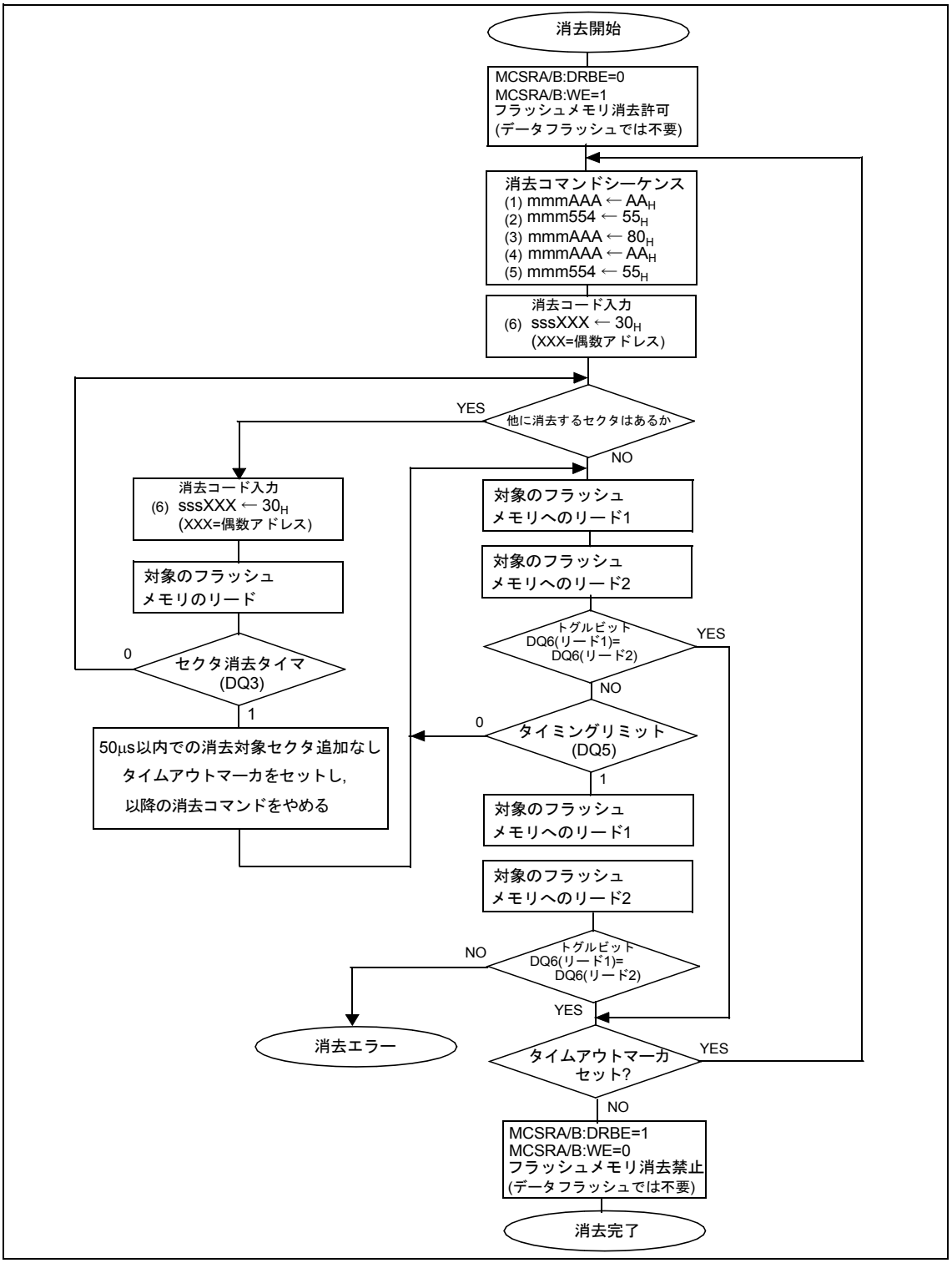
最後のセクタ消去コード (30_H) の書込み後 , 50µs のセクタ消去タイムアウト期間終了により消去が開始します。セクタ消去コードの各書込みごとにセクタ消去タイムアウト期間を延長します。消去コードを書き込むたび , その受け入れを確認するために , セクタ消去タイマフラグ (DQ3) を確認してください。

33.8.5.2 フラッシュメモリのセクタ消去例

ハードウェアシーケンスフラグ (「33.7 自動アルゴリズム実行状態の確認」を参照してください) を用いることでフラッシュメモリ内部の自動アルゴリズムの状態を判定することができます。Figure 33-12 の例では消去終了の確認にトグルビットフラグ (DQ6) を使用した例を示します。

ハードウェアシーケンスフラグの変化は CPU 動作に非同期かつビット間スキューがありますので , タイミングリミットオーバフラグ (DQ5) が "1" を示したあとに , 再度 DQ6 を確認する必要があります。
データポーリングフラグ (DQ7) も同様です。ポーリングに DQ7 を用いた場合には DQ5 が "1" を示したあとに , 再度 DQ7 を確認してください。

Figure 33-12. フラッシュメモリのセクタ消去手順の例



＜注意事項＞

"mmm" はコマンドを送るフラッシュメモリモジュールを指定してください。

"sss" はコマンドを送るフラッシュメモリセクタを指定してください。

33.8.6 セクタ消去一時停止

セクタ消去一時停止コマンドを入力し, フラッシュメモリセクタ消去の一時停止を行う手順について説明します。消去中でないセクタからデータを読み出すことが可能です。

33.8.6.1 フラッシュメモリのセクタ消去一時停止

セクタ消去一時停止コマンド ($B0_H$) を対象のフラッシュメモリの偶数アドレスに送ることにより , フラッシュメモリのセクタ消去が一時停止します。一時停止状態では , 消去指定されていないセクタの読出し , あるいは書込みが可能です。再度の消去操作はできません。
セクタ消去実行中 , またはセクタ消去タイムアウト期間中のみ , セクタ消去一時停止コマンドは有効です。コマンドは , チップ消去中や書込み中は無視されます。セクタ消去一時停止中に , セクタ消去一時停止コマンドを発行しても , コマンドは無視されます。
セクタ消去タイムアウト期間中に , セクタ消去一時停止コマンドを入力した場合 , 直ちにセクタ消去タイムアウトを終了し , 消去一時停止状態になります。セクタ消去タイムアウト期間終了後の消去動作中に消去一時停止コマンドを入力した場合 , 最大 20µs 後に消去一時停止状態になります。

プログラム/データフラッシュ(メインフラッシュ)のセクタ消去一時停止機能の使用上の注意

- セクタ消去コマンドの開始または再開した後, セクタ消去一時停止コマンド実行前に, 最小1.2msのウェイト時間が必要です。
- 1つのセクタ消去操作は10回以上中断させてはいけません。

データフラッシュのセクタ消去一時停止機能の使用上の注意

データフラッシュの制限も , 上に述べられているプログラム / データフラッシュ (メインフラッシュ) と同じ注意が必要です。
しかし , 以下の条件を守ることで , セクタ消去操作をより頻繁に一時停止させることができます。

- セクタ消去の間, DQ4フラグを最大20 ms周期でポーリングします。
- DQ4が"1"(一時停止不可)であれば, ポーリングを継続します。この期間に, フラッシュに一時停止コマンドを書き込んではいけません。
- DQ4が"0"(一時停止可)になれば, 直ちに一時停止コマンドを書き込んでください。

33.8.7 セクタ消去再開

セクタ消去再開コマンドを入力し, 一時停止したフラッシュメモリセクタ消去を再開する手順について説明します。

フラッシュメモリのセクタ消去再開

セクタ消去再開コマンド (30_H) を対象のフラッシュメモリの偶数アドレスに送ることにより , 一時停止したフラッシュメモリセクタ消去が再開できます。
セクタ消去中に , セクタ消去再開コマンドを発行しても無視されます。
セクタ消去再開コマンドは, セクタ消去一時停止コマンドによって一時停止させられたセクタ消去を再開させます。

33.8.8 セクタ消去/書込み保護

意図しない消去または書込みからプログラム/データフラッシュ(メインフラッシュ)のセクタを保護するための手順を説明します。

33.8.8.1 セクタ書込み保護機能

プログラム / データフラッシュ (メインフラッシュ) の個々のセクタはフラッシュメモリ書込み制御レジスタの設定によってセクタ消去または書込みから保護できます。これらのレジスタの説明は , 「33.4.3 フラッシュメモリ書込み制御レジスタ 0 ～ 5(FMWC0 ～ FMWC5)」を参照してください。この機能は指定されたフラッシュセクタの意図しない消去または再書込みからアプリケーションを守るために使用できます。
パラレルフラッシュ書込みモードあるいはシリアルライタモードでは保護機能は働きません。
書込み保護されているセクタへの書込みは , 対応するフラッシュメモリへのリセットが発生して中断されます。これは , すべてのコマンドシーケンスの書込みアクセスを含みます。したがって , フラッシュメモリへコマンドシーケンスを送る場合は , 保護されていないセクタである必要があります。
発生するリセットは 64CLKB クロック周期となります。その間 , 対応する RDY フラグ (MCSRA:RDY または MCSRB:RDY) は "0" になります。
チップ消去をするために , 存在しないセクタであっても対応する書込み制御ビットに "1" を設定してください。

自動アルゴリズム実行中のフラッシュ書込みに関する注意

書込み保護されたセクタへ書込みアクセスをするとフラッシュメモリがリセットされます。このリセットで実行中の自動アルゴリズム (書込み / 消去) は停止します。このため , 書込み中だったデータは不定値となり , 消去中のセクタデータは破壊されます。このような場合 , セクタ / チップ消去からやり直してください。

チップ消去

いくつかのセクタが保護されているとき , チップ消去コマンドが実行される場合とされない場合とがあります。

以下の場合はチップ消去コマンドが実行されません

- 保護されたセクタに対し, チップ消去コマンドシーケンスの書込みが行われる場合(フラッシュメモリのハードウェアリセットが発生します)。全てのセクタ保護ビットが"0"に設定されていた場合にはチップ消去コマンドシーケンスはキャンセルされます。
- チップ・消去コマンドの5番目と6番目の書込みサイクルがワード命令で0x0055や0x0010が書き込まれる場合(チップ消去コマンドが無視されます)

以下の場合は書込み保護がされていた場合でもチップ消去コマンドが実行されます

- チップ・消去コマンドが, 保護されていない実存するセクタに対して発行され, かつ, コマンドシーケンス書込みがワード命令で行われ, そのときの上位バイトのデータが0x00でない場合

33.8.8.2 ユーザプログラムでの書込み保護の設定

ユーザプログラムで , フラッシュメモリ書込み制御レジスタ (FMWC0 ～ FMWC5) 設定できます。

33.8.8.3 ユーザプログラムの介入なしで書込み保護の起動

フラッシュメモリの特定領域(ROM構成ブロックRCBA/RCBB)に特定のデータを書き込むことによって, フラッシュメモリの個々のセクタを消去や書込みから保護できます。この場合 , 書込み保護の設定をユーザプログラムで行う必要はありません。
フラッシュメモリ A のセクタの書込み保護は , フラッシュメモリ書込み保護起動マーカ A(FWPAMA1/FWPAMA0) と RCBA に設定されたフラッシュ書込み保護セクタマーカ A(FWPSMA0 ～ FWPSMA4) によって定義されます。
フラッシュメモリ B のセクタの書込み保護は , RCBB に設定されたフラッシュメモリ書込み保護起動マーカ B(FWPAMB1/FWPAMB0) とフラッシュ書込み保護セクタマーカ B(FWPSMB0 ～ FWPSMB4) によって定義されます。
リセット後ユーザプログラム実行前に , 以下のように実行されます。(内部ベクタモードおよび外部ベクタモード)

フラッシュメモリ A:

FWPAMA1/FWPAMA0 レジスタが "292D3A7B$_H$" のときに , フラッシュメモリ A のすべてのセクタに対応する FWPSMA0 ～ FWPSMA4 の内容を FMWC1 ～ FMWC5 に転送します。FWPAMA1/FWPAMA0 レジスタがほかの値であれば , FWPSMA0 ～ FWPSMA4 の内容は無視され , FMWC レジスタへの転送は実行されません。

フラッシュメモリ B (フラッシュメモリ Bが搭載されているデバイスのみ):

FWPAMB1/FWPAMB0 レジスタが "292D3A7B$_H$" のときに , フラッシュメモリ B のすべてのセクタに対応する FWPSMB0 ～ FWPSMB4 の内容を FMWC0 および FMWC2 ～ FMWC5 に転送します。
FWPAMB1/FWPAMB0 レジスタがほかの値であれば , FWPSMB0 ～ FWPSMB4 の内容は無視され , FMWC レジスタへの転送は実行されません。

これらマーカを含むセクタが書込み / 消去保護されていない限り (フラッシュメモリ A では FWPSMA0:WCA0=1, フラッシュメモリ B では FWPSMB0:WCB=1), マーカへの書込みまたはマーカを含むセクタ消去によってマーカの内容を更新できます。更新された内容はリセットによって有効となります。
フラッシュメモリ書込み保護マーカによってセクタ設定の書込み保護が有効になった後でも, ユーザプログラムがフラッシュメモリ書込み制御レジスタ 0 ～ 5(FMWC0 ～ FMWC5) に対応するビットを "0" に設定することによって , 書込み保護セクタを追加できます。
シリアルライタモードでは , 設定データは自動的には FMWC レジスタに転送されません。
内部ベクタモードでは , 一時的な UART スキャンが許可されているとき , シリアル通信を終えた後に , FMWC レジスタに設定データを転送します。

Figure 33-13. フラッシュメモリ書込み保護起動マーカ(FWPAMA0/1, FWPAMB0/1)の構成

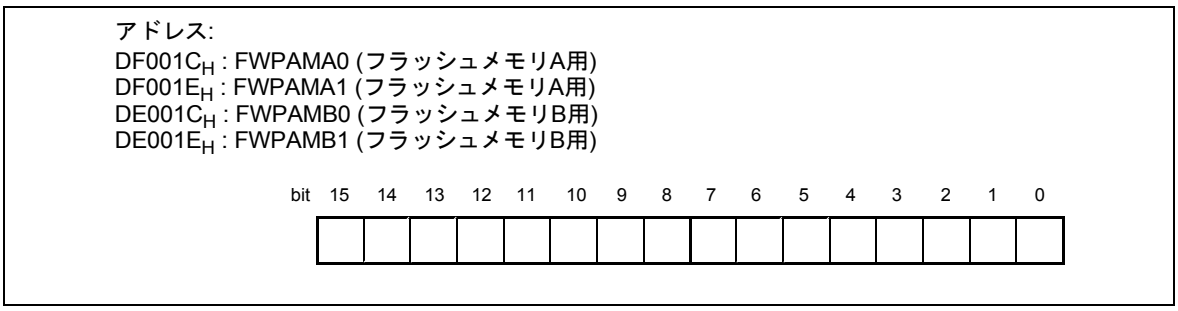
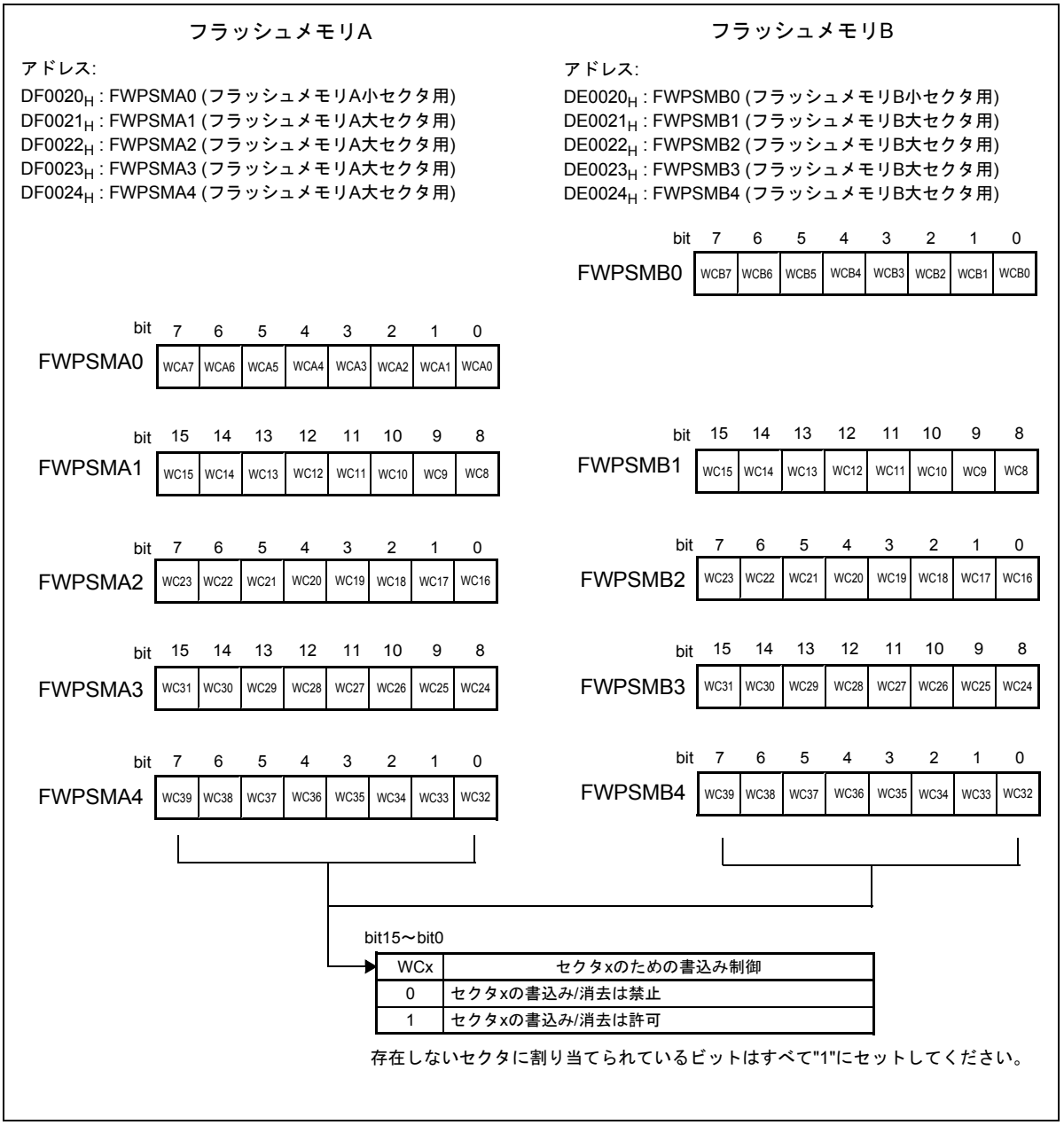


Figure 33-14. フラッシュメモリ書込み保護セクタマーカ(FWPSMA0～FWPSMA5, FWPSMB0～FWPSMB5)の構成



33.9 フラッシュメモリ使用上の注意

フラッシュメモリを使用するための注意点を示します。

33.9.1 フラッシュメモリ使用上の注意

CPUリセット(RSTX)の入力

CPU リセット (電源リセット , 外部端子リセット , ソフトウェアリセット , ウォッチドッグタイマリセット , クロック停止検出リセット) では , フラッシュメモリをハードウェアリセットします。フラッシュメモリ書込み中にリセットが発生すると , 書込み中のアドレスは不定値となります。

セクタ消去開始後にリセットが発生すると , 消去中のセクタの内容は不定となります。その場合 , 再度このセクタを消去し , 消去を完了させてください。

フラッシュメモリへのプログラムアクセス

自動アルゴリズム動作中は , そのフラッシュメモリモジュールへの読出しアクセスはできません (ハードウェアシーケンスフラグが読み出されます)。

したがって , 消去 / 書込み中のフラッシュメモリから CPU が命令を実行しないようにしてください。フラッシュメモリ A に消去 / 書込みを行う場合は , 命令はフラッシュメモリ B (利用可能な場合) または RAM から実行するようにしてください。フラッシュメモリ B に消去 / 書込みを行う場合は , 命令はフラッシュメモリ A または RAM から実行するようにしてください。

同様に , テーブルベースレジスタ (TBR) が , 消去あるいは書込み対象のフラッシュメモリに配置された割込みベクタテーブルを示していないことを確認してください。書込み / 消去対象ではないフラッシュメモリ (利用可能な場合) または RAM に設置された割込みベクタテーブルを示すように TBR を設定するか , 自動アルゴリズムを作動させる前に割込みを完全に禁止してください。

DMA

DMA はデータフラッシュメモリの書込み , および , 消去完了時の割り込みのみに使用できます。プログラム / データフラッシュ (メインフラッシュ) には使用できません。

データ書込み/消去中(セクタ消去一時停止中も含む)のモード遷移

- コマンド発行を含め, 書込みおよび消去シーケンス中はフラッシュメモリ書込み許可ビット(MCSRA, MCSRB:WE)を"0"に設定しないでください。
- フラッシュメモリの書込みを許可(MCSRA,MCSRB:WE=1)しているときは,サブクロックモードおよびスタンバイモード(スリープモード, タイマモード, ストップモード)には遷移しないでください。

33.10 フラッシュメモリプログラム例

本項では, フラッシュメモリのプログラム例を示します。

プログラム例

```
Flash memory sample program:
;=============================================================
; THIS SAMPLE CODE IS PROVIDED AS IS AND IS SUBJECT TO ALTERATIONS.
; CYPRESS ACCEPTS NO RESPONSIBILITY OR LIABILITY
; FOR ANY ERRORS OR ELIGIBILITY FOR ANY PURPOSES.
;
; (C) CYPRESS EUROPE 1998-2017
;=============================================================
;
; このプログラムはフラッシュメモリ A のセクタ SA1 を消去し , 0xAA55 をセクタ SA1 に書き込みます。
; その後 , 結果を P00 ポートに出力します。
;
; Bit# 0 1 2 3 4 5 6 7
;       ---------------
;       | | |   | | |
;       | | |   | | |
;       | | |   | | +---- EERR: 消去タイムリミットオーバエラー
;       | | |   | +------ EOK1: 消去成功 ( タイムリミットオーバ再読出し付き )
;       | | |   +-------- EOK0: 消去成功
;       | | +------------ PERR: 書込みタイムリミットオーバ
;       | +-------------- POK1: 書込み成功 ( タイムリミットオーバ再読出し付き )
;       +---------------- POK0: 書込み成功
                  .PROGRAM FLASH_W_E
                  .TITLE FLASH_WRITE_ERASE
; Defines
#set       POK0    0x0001
#set       POK1    0x0002
#set       PERR    0x0004
#set       EOK0    0x0010
#set       EOK1    0x0020
#set       EERR    0x0040
#set       DQ3     0x0008        ; セクタ消去開始
#set       DQ5     0x0020        ; タイムリミットオーバビット
#set       DQ7     0x0080        ; データポーリングビット
; I/O
PDR00      .EQU    0x0000        ; ポートデータレジスタ 00
DDR00      .EQU    0x0430        ; ポート方向レジスタ 00
MCSRA      .EQU    0x03F1        ; フラッシュメモリ制御ステータスレジスタ
MTCRA      .EQU    0x03F2        ; フラッシュメモリタイミング設定レジスタ
```

```
FMWC1       .EQU    0x03F9          ; セクタ SA0-SA7 保護解除
CKSR        .EQU    0x0401          ; クロック選択レジスタ
CKSSR       .EQU    0x0402          ; クロック安定化選択レジスタ
CKMR        .EQU    0x0403          ; クロックモニタレジスタ
CKFCR       .EQU    0x0404          ; クロック周波数制御レジスタ
PLLCR       .EQU    0x0406          ; PLL 制御レジスタ
; RAM
RAMSTART    .EQU    0x4000          ; RAM コードの開始アドレス
; FLASH
SA1         .EQU    0x00DF2000      ; SA1 データセクタ
SA1_554     .EQU    0x00DF2554      ; フラッシュメモリ書込みシーケンスアドレス 1
SA1_AAA     .EQU    0x00DF2AAA      ; フラッシュメモリ書込みシーケンスアドレス 2
                    .SECTION RESVECT, CONST, LOCATE=0xFFFFDC
                    .DATA.E START
                    .SECTION BOOT_SELECT, CONST, LOCATE=0xDF0030
                    .DATA.L 0xFFFFFFFF
;=============================================================
;
; フラッシュメモリ内に配置するメインプログラム
                .SECTION CODE_FLASH, CODE, LOCATE=0xFF8000
START:    AND   CCR,   #0x80              ; 割込み禁止
                MOV   RP,#0               ; レジスタバンク #0
                MOV   CKSSR,#0xFC         ; クロック安定時間 2^16/CLKMC
                                          ; の設定
                MOVW  PLLCR,#0x000D       ; CPU: 56 MHz, CLKP2: 14 MHz
                                          ; ( 外部 4 MHz)
                MOVW  CKFCR,#0x3001       ; CLKB/1, CLKP1/1, CLKP2/4
                MOVW  MTCRA,#0x4B3B       ; 3 ウェイト状態
                MOV   CKSR,#0xFA
                MOVL  A,#0x00002540       ; システムスタックポインタの先頭
                MOVW  SP,A
                SWAPW
                MOV   SSB,A
                ; RAM コードを RAM にコピーする
                MOVW  A,#RAMSTART         ; RAM コードの開始
                MOVW  A,#(0xFFFF & FLASH)     ; ROM コードの開始
                MOVW  RW0,#SIZEOF(CODE_FLASH_RAM)
                                          ; コードバイトの数
                MOVS  ADB,PCB
                MOV   A,#0xFF             ; INIT ポート
                MOV   DDR00,A
                MOVN  A,#0                ; 0x00 をポート 00 へ
                MOV   I:PDR00,A
```

```
                MOV    A,#0xFF           ; 保護解除 SA0 - SA7
                MOV    FMWC1,A
PLLWAIT:  BBCCKMR:6, PLLWAIT             ; クロック安定待ち
                CALLP  (RAMSTART)         ; セクタ消去
                MOV    PDR00,A           ; ステータスフラグをポート 00 へ
                MOVW   RW6,A             ; ステータスフラグを保存
                CALLP  (RAMSTART + 4)     ; データ書込み
                MOVW   A,RW6             ; ステータスフラグの回復
                ORW    A
                MOV    PDR00,A           ; 結果の表示
LOOP:     BRA LOOP                       ; 無限ループ
;====== RAM コードのフラッシュ消去 / 書込み ==========================
;
; RAM コードの機能
                    .SECTION CODE_FLASH_RAM, CODE, LOCATE=0xFF9000
; ジャンプテーブル
FLASH:      JMPP    (ERASE - 0xFF9000 + RAMSTART)
            JMPP    (WRITE - 0xFF9000 + RAMSTART)
;====== セクタ消去 ============================================
;
; セクタ SA1 を消去
; 入力 : なし
; 出力 (A アキュムレータの内容 ):EOK0 = 消去成功
;                   EOK1 = 消去成功 ( タイムリミットオーバ再読出し付き )
;                   EERR = タイムリミットオーバエラー
ERASE:          MOV    A,MCSRA           ; フラッシュタイミングの保存
                PUSHW  A
                MOV    A,MTCRA
                PUSHW  A
                CLRB   MCSRA:4           ; コード読出しバッファ無効
                CLRB   MCSRA:5           ; データ読出しバッファ無効
                MOVW   MTCRA,#0x4B3D     ; フラッシュアクセスを
                                         ; 5 ウェイト状態にする
                SETB   MCSRA:3           ; 書込み許可
                ; セクタ消去コマンドシーケンス
                MOVL   A,#SA1_AAA
                MOVL   RL0, A
                MOV    A,#0xAA           ; *0xDF2AAA = 0xAA
                MOVW   @RL0,A
                MOVL   A,#SA1_554
                MOVL   RL0,A
                MOV    A,#0x55           ; *0xDF2554 = 0x55
```

```
                MOVW   @RL0,A
                MOVL   A,#SA1_AAA
                MOVL   RL0, A
                MOV    A,#0x80          ; *0xDF2AAA = 0x80
                MOVW   @RL0,A
                MOVL   A,#SA1_AAA
                MOVL   RL0,A
                MOV    A,#0xAA          ; *0xDF2AAA = 0xAA
                MOVW   @RL0,A
                MOVL   A,#SA1_554
                MOVL   RL0, A
                MOV    A,#0x55          ; *0xDF2554 = 0x55
                MOVW   @RL0,A
                MOVL   A,#SA1
                MOVL   RL0,A
                MOV    A,#0x30          ; *0xDF2000 = 0x30
                MOVW   @RL0,A
                ; セクタ消去タイムアウト期間の終了を待つ
E_DQ3:          MOVW   A,@RL0+0         ; セクタ状態の読出し
                MOVN   A,#DQ3
                ANDW   A
                MOVN   A,#DQ3
                CMPW   A
                BNZ    E_DQ3
                ; データポーリングアルゴリズム
                MOVN   A,#0             ; RW2 にステータスフラグ
                MOVW   RW2,A
E_LOOP:         MOVW   A,@RL0+0         ; セクタ状態の読出し
                MOVW   RW5,A
                ANDW   A,#DQ7           ; データポーリング
                CWBNE   A,#DQ7,E_1
                MOV    A,#EOK0          ; 消去成功
                MOVW   RW2,A
                BRA    E_2
E_1:            MOVW   A,RW5            ; 1 回目の読出し値のリストア
                ANDW   A,#DQ5           ; タイムリミットオーバ？
                CWBNE  A,#DQ5,E_2
                MOVW   A,@RL0+0         ; セクタ状態の再読出し
                ANDW   A,#DQ7           ; データポーリング
                CWBNE  A,#DQ7,E_ERR
                MOV    A,#EOK1          ; 消去成功
                MOVW   RW2,A
```

```
                BRA    E_2
E_ERR:          MOV    A,#EERR          ; タイムリミットオーバエラー
                MOVW RW2, A
E_2:            MOVW   A,RW2
                BZ     E_LOOP
                CLRB   MCSRA:3          ; 書込み許可のリセット
                POPW   A                ; フラッシュタイミングの回復
                MOV    MTCRA,A
                POPW   A
                MOV    MCSRA,A
                MOVW   A,RW2            ; A にステータスフラグ
RAMEND:  RETP
;====== フラッシュ書込み ========================================
;
; 0xAA55 をセクタ SA1 の先頭 (0xDF2000) に書き込む
; 入力 : なし
; 出力 (A アキュムレータの内容 ):POK0 = 書込み成功
;                  POK1 = 書込み成功 ( タイムリミットオーバ再読出し付き )
;                  PERR = タイムリミットオーバ
WRITE:          MOV    A,MCSRA          ; フラッシュタイミングの保存
                PUSHW  A
                MOV    A,MTCRA
                PUSHW  A
                CLRB   MCSRA:4          ; コード読出しバッファ無効
                CLRB   MCSRA:5          ; データ読出しバッファ無効
                MOVW   MTCRA,#0x4B3D    ; フラッシュアクセスを
                                        ; 5 ウェイト状態にする
                SETB   MCSRA:3          ; 書込み許可
                ; 書込みコマンドシーケンス
                MOVL   A,#SA1_AAA
                MOVL   RL0, A
                MOV    A,#0xAA          ; *0xDF2AAA = 0xAA
                MOVW   @RL0, A
                MOVL   A,#SA1_554
                MOVL   RL0, A
                MOV    A,#0x55          ; *0xDF2554 = 0x55
                MOVW   @RL0,A
                MOVL   A,#SA1_AAA
                MOVL   RL0,A
                MOV    A,#0xA0          ; *0xDF2AAA = 0xA0
                MOVW   @RL0,A
                MOVL   A,#SA1
```

```
                MOVL   RL0,A
                MOVW   A,#0xAA55          ; *0xDF2000 = ダミーデータ
                MOVW   RW4,A              ; RW4= 書込み値 (PD)
                MOVW   @RL0,A
                ; データポーリングアルゴリズム
                MOVN   A,#0               ; RW2 にステータスフラグ
                MOVW   RW2,A
W_LOOP:         MOVW   A,@RL0+0           ; フラッシュ状態の読出し
                MOVW   RW5,A              ; 1 回目の読出し値のストア
                XORW A, RW4               ; 書込み期待値
                ANDW A, #DQ7              ; DQ7=DATA?
                BNZ    W_1
                MOVW   A,#POK0            ; 書込み成功
                MOVW   RW2,A
                BRA    W_2
W_1:            MOVW   A,RW5              ; 1 回目の読出し値のリストア
                ANDW   A,#DQ5             ; タイムアウト ?
                CWBNE  A,#DQ5,W_2
                MOVW   A,@RL0+0           ; フラッシュ状態の再読出し
                XORW   A,RW4              ; 期待値と再比較
                ANDW   A,#DQ7             ; DQ7=DATA?
                BNZ    W_ERR
                MOVW   A,#POK1            ; 書込み成功
                MOVW   RW2,A
                BRA    W_2
W_ERR:          MOVW   A,#PERR            ; タイムリミットオーバエラー
                MOVW   RW2,A
W_2:            MOVW   A,RW2
                BZ     W_LOOP
                CLRB   MCSRA:3            ; 書込み有効のリセット
                POPW   A                  ; フラッシュタイミングの回復
                MOV    MTCRA,A
                POPW   A
                MOV    MCSRA,A
                MOVW   A,RW2              ; A にステータスフラグ
                RETP
; ===== SA1 ダミーデータ ========================================
;
; セクタ消去確認のため 4  0x00 バイトで満たされた SA1
                .SECTION SA1_DUMMY_DATA, CODE, LOCATE=0xDF2000
                .DATA.L 0x00000000
                .END START                ; デバッグ用
```

34. マスクROMメモリ

マスクROMメモリの機能と動作について説明します。

34.1 概要

マスクROMメモリの主な機能を説明します。マスクROMのインタフェースはフラッシュメモリのインタフェースと兼用です。マスクROMを使用するためのインタフェースはフラッシュメモリを使用するためのインタフェースと兼用です。

主な機能

マスク ROM メモリには以下の機能があります。

- マスクROMメモリインタフェースの構成はフラッシュメモリインタフェースの構成と同一です。
- フラッシュ品の時間互換性に関して, ウェイトサイクルはメモリタイミング構成レジスタ(MTCRA)によって設定できます。
- フラッシュメモリとの互換性のためのコード, データ読出し用の2つの異なる読出しバッファ
  バッファは有効(初期値)または無効にできます。
- フラッシュメモリに関するコードセキュリティ(読出し保護)
  詳細は「第35章 ROM/フラッシュセキュリティ機能」を参照してください。

34.2 マスクROMメモリ制御レジスタ

マスクROMメモリ制御レジスタについて説明します。

メモリ制御ステータスレジスタ(MCSRA)

マスク ROM メモリインタフェースに「フラッシュメモリ互換」のビットがありますが ,

フラッシュメモリインタフェースと互換性があることを示しています。マスク ROM メモリの場合 , これらのビットは無効です。

Figure 34-1. メモリ制御ステータスレジスタA (MCSRA)

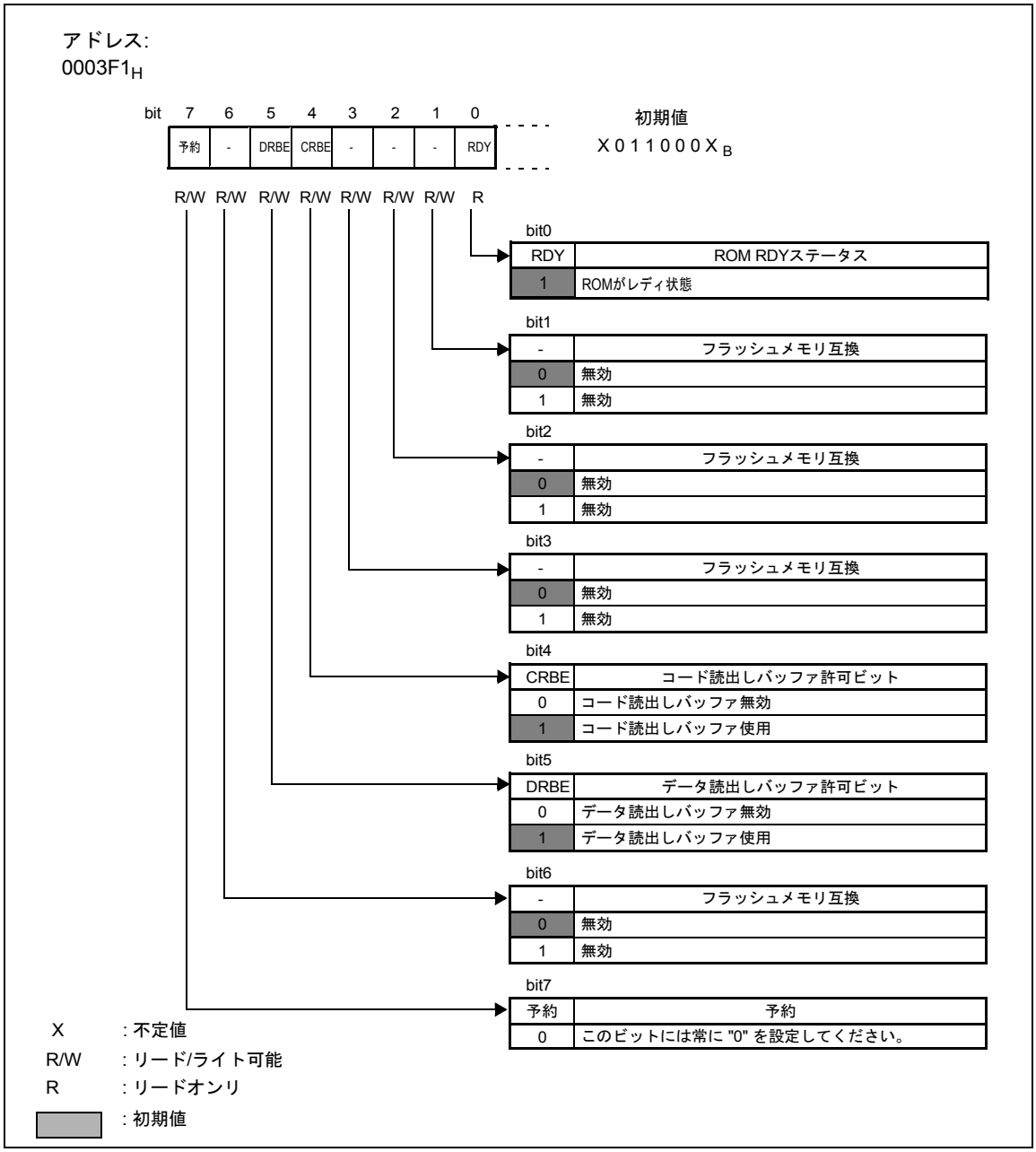


Table 34-1. メモリ制御ステータスレジスタ(MCSRA)

ビット名		機能
bit 0	RDY	このビットはフラッシュインタフェースとの互換性があります。 読出し値が"1"の場合, ROMがレディ状態です。
bit 1～bit 3	-	フラッシュメモリインタフェースとの互換性にのみ有効ですが, フラッシュメモリ機能との互換性はありません。書き込まれた値は無効です。
bit 4	CRBE	• コード読出しバッファの許可 / 無効設定を行います。 • このビットを "1" に設定するとコード読出しバッファが許可されます。すべてのアドレスからのコード読出しアクセスは 2 つの 16 ビットワードを読出します。1 つは指定されたアドレスのワードで , もう 1 つは隣接するアドレスのワードで , 結果として 32 ビット境界に並ぶ 2 つの 16 ビットワードが読み出されます。両方のワードは読出しバッファに格納されます。バッファに格納されたコードのうち , 同一の 32bit 境界内のワードへの CPU アクセスはウェイトサイクルなしに実行されます。 • このビットを "0" に設定するとコード読出しバッファが禁止されます。コードは常にマスク ROM から直接読み出されます。 このビットの初期値は"1"(バッファ許可)です。
bit 5	DRBE	• データ読出しバッファの許可 / 無効設定を行います。 • このビットを "1" に設定するとデータ読出しバッファが許可されます。すべてのアドレスからのデータ読出しアクセスは 2 つの 16 ビットワードを読出します。1 つは指定されたアドレスのワードで , もう 1 つは隣接するアドレスのワードで , 結果として 32 ビット境界に並ぶ 2 つの 16 ビットワードが読み出されます。両方のワードは読出しバッファに格納されます。バッファに格納されたコードのうち , 後のワードへの CPU アクセスはフラッシュのウェイトサイクルなしに実行されます。 • このビットを "0" に設定するとデータ読出しバッファが無効になります。データは常にマスク ROM から直接読み出されます。 このビットの初期値は "1" (バッファ許可)です。
bit 6	-	フラッシュメモリインタフェースとの互換性にのみ有効ですが, フラッシュメモリ機能との互換性はありません。書き込まれた値は無効です。
bit 7	予約	• 常に "0" に設定してください。 • 読出し値は不定です。 • 本レジスタへのリードモディファイライト(RMW)系命令はこのビットには影響しません。

メモリタイミング設定レジスタA (MTCRA)

マスク ROM メモリインタフェースに「フラッシュメモリ互換」のビットがありますが ,

フラッシュメモリインタフェースと互換性があることを示しています。マスク ROM メモリの場合 , これらのビットは無効です。

Figure 34-2. メモリタイミング設定レジスタA下位バイト(MTCRAL)

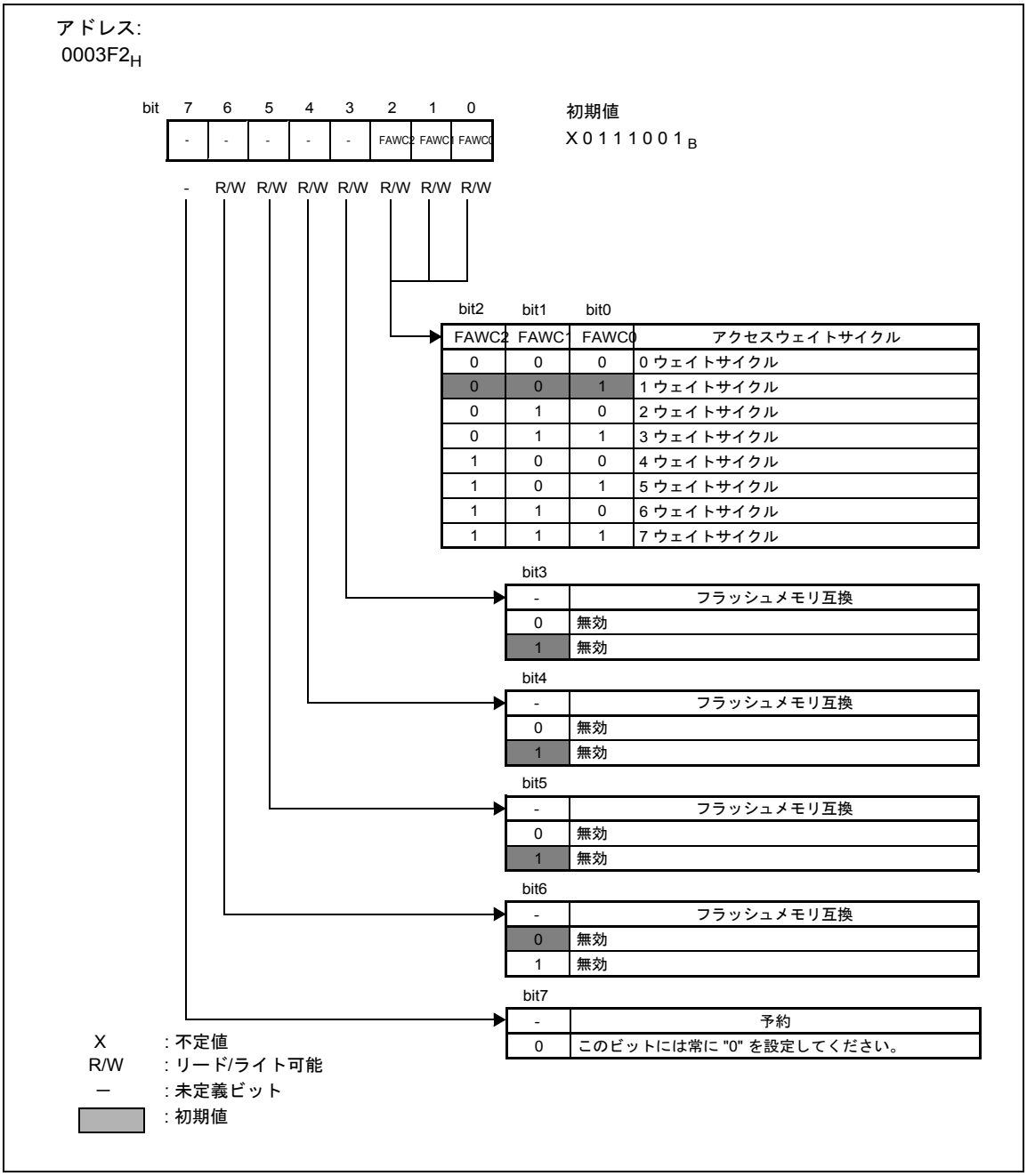


Table 34-2. メモリタイミング設定レジスタ下位バイト(MTCRAL)

ビット名		機能
bit 0～bit 2	FAWC0～FAWC2	• このビットはマスク ROM への読出しや書込み時のウェイトサイクルを設定します。CPU ウェイトサイクル (CLKB サイクル) が , マスク ROM への読出しや書込みアクセスにどの位加えられるかを定義します。 • このビットはマシンクロック周波数や他の設定ビットを考慮して設定する必要があります。 • このビットの初期値は001_Bです (1ウェイトサイクル)。
bit 3～bit 6	-	フラッシュメモリインタフェースとの互換性にのみ有効ですが, フラッシュメモリ機能と互換性はありません。書き込まれた値は無効です。
bit 7	予約	• 常に "0" を設定してください。 • 読出し値は不定です。 • 本レジスタへのリードモディファイライト(RMW)系命令は禁止です。

Table 34-3. メモリタイミング設定レジスタA上位バイト(MTCRAH)

ビット名		機能
bit 8～bit 15	-	フラッシュメモリインタフェースとの互換性にのみ有効ですが, フラッシュメモリ機能との互換性はありません。書き込まれた値は無効です。

35. ROM/フラッシュセキュリティ機能

ROM/フラッシュセキュリティの機能および動作について説明します。

35.1 ROM/フラッシュセキュリティの概要

ROM/フラッシュセキュリティは, アプリケーションで意図した以外の方法で, ROM/フラッシュメモリの内容が読み出されることを防止する機能です。

ROM/フラッシュセキュリティ機能

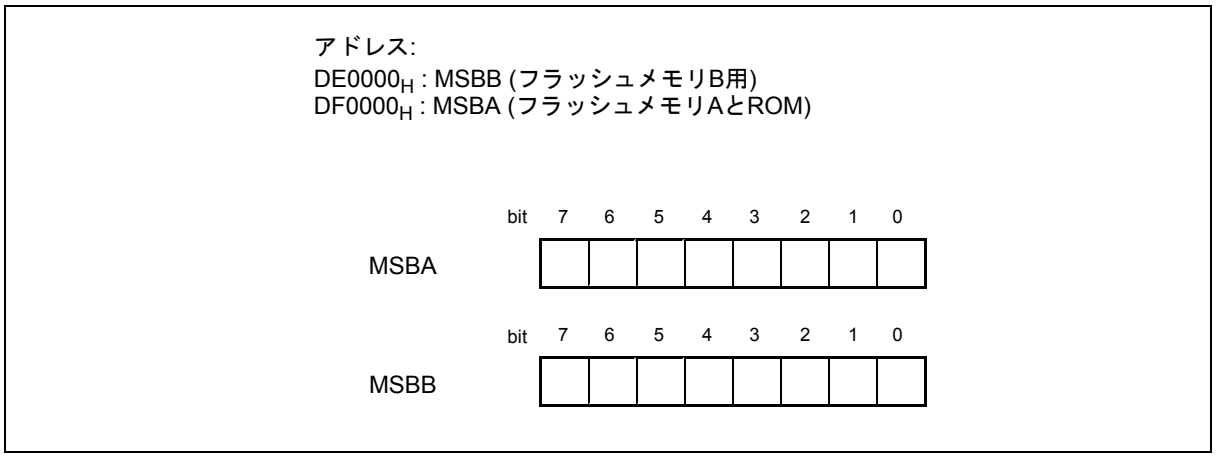
- アプリケーションが意図しないときに, ROM/フラッシュメモリの内容が読み出されることを防止する読出し禁止機能です。 内部ベクタモードで開始された内部ユーザプログラムのみがROM/フラッシュメモリ の内容にアクセスできます。 以下は, ROM/フラッシュメモリ を読み出すことができません。
  - 外部起動ベクタフェッチ (モード 0/1/6)によって起動したプログラム
  - 外部並列フラッシュプログラマ (モード 7)
  - シリアル通信モード (モード2)
- どのモードでも, 内部フラッシュメモリを消去することができます。
- フラッシュメモリA, Bとデータフラッシュメモリ A, Bに対し別々に読出しを禁止します。
- もし設定されていれば,シリアル通信モードの128 ビットセキュリティキーによって読出し禁止は解除されます。 また, セキュリティキーは常時無効にできます。

35.2 ROM/フラッシュセキュリティの使用方法

ROM/フラッシュセキュリティの許可/禁止の方法と, ROM/フラッシュメモリがどのように意図しない読出しから防止されるかを説明します。

ROM/フラッシュセキュリティを許可する

Figure 35-1. ROM/フラッシュセキュリティバイトの設定



フラッシュメモリセキュリティは , フラッシュメモリセキュリティバイトの MSBA, MSBB, MSBDA, MSBDB に書き込むことにより , フラッシュメモリ A, B とデータフラッシュメモリ A, B を別々に許可することができます。ROM デバイスの場合は , セキュリティは MSBA レジスタのみで制御されます。

MSBA の内容が 99_H のときは , フラッシュメモリ A または ROM メモリは読出しが禁止されます。MSBA の内容が 66_H のときは , フラッシュメモリ A または ROM メモリの読出しは禁止されません。また , 66_H と 99_H 以外の内容を MSBA に書き込まないでください。

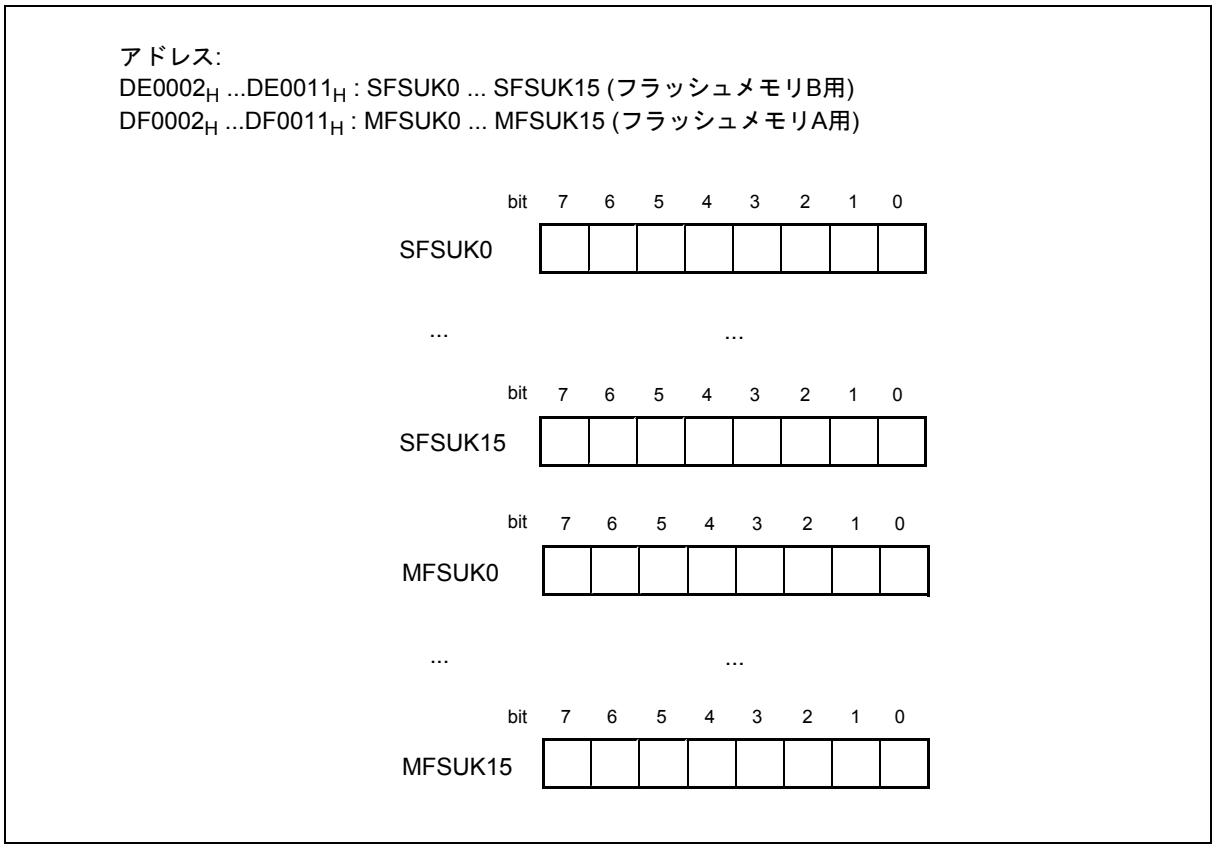
MSBB, MSBDA, MSBDB の内容が 99_H のときは , それぞれ対応するフラッシュメモリの読出しが禁止されます。MSBB, MSBDA, MSBDB の内容が 66_H のときは , それぞれ対応するフラッシュメモリの読出しは禁止されません。また , 66_H と 99_H 以外の内容を MSBB, MSBDA, MSBDB に書き込まないでください。

＜注意事項＞

フラッシュメモリを消去した後は, フラッシュメモリ内のすべてのビットの値は"1"です。したがって , フラッシュメモリ領域の MSBA, MSBB, MSBDA または MSBDB に書込みがされないときは , MSBA, MSBB, MSBDA または MSBDB の内容は FF_H であり , フラッシュメモリの内容は読出しが禁止されません。

ROM/フラッシュセキュリティを解除する

Figure 35-2. ROM/フラッシュセキュリティ解除キーの設定



読出し禁止 (フラッシュセキュリティバイトが 99_H) のフラッシュまたは ROM メモリは , 書込みに使用された UART が解除キーを受信した後は , 引き続きシリアル書込みモードで読み出すことができます。UART を介して解除キーを送信する方法の詳細については, アプリケーションノートのMCU-AN-300213-Eを参照してください。

解除キーは, 解除キーバイトMSUKA0 ～MSUKA15 (フラッシュメモリA, ROMメモリ用), MSUKB0 ～MSUKB15 (フラッシュメモリ B 用), MSUKDA0 ～ MSUKDA15(データフラッシュメモリ A 用) や MSUKDB0 ～ MSUKDB15 (データフラッシュメモリ B 用)に格納されたキーと比較されます。受信したキーがフラッシュROMメモリに格納されたキーと一致する場合は, それに続くシリアル書込みコマンドによりメモリを読み出すことができます。

MSUKA0 ～ MSUKA15 (フラッシュメモリ A, ROM メモリ用), MSUKB0 ～ MSUKB15 (フラッシュメモリ B 用), MSUKDA0 ～ MSUKDA15 (データフラッシュメモリ A 用), MSUKDB0 ～ MSUKDB15(データフラッシュメモリ B 用) に格納された解除キーの値がすべてゼロの場合は , 対応するフラッシュメモリの読出し禁止は解除させることができなくなります。ROM デバイスにおいてこのキー値は許可させられません。

＜注意事項＞

解除キーの動作停止もまたフラッシュ内容の分析エラーを防止します。必要な場合のみこの機能を使用してください。

ROM/フラッシュセキュリティを禁止する

マスク ROM デバイス上では , ROM/ フラッシュセキュリティを禁止することはできません。

フラッシュメモリデバイス上では , ROM/ フラッシュセキュリティを以下のように禁止することができます。

- 内部ベクタモード (MD2～MD0 = 011_B) を開始するとき, アプリケーションプログラムはフラッシュセキュリティを以下の方法で禁止することができます。
  - このフラッシュメモリ領域に書き込むことにより, フラッシュ/ROM セキュリティバイトの値を 99_H とは異なる値に変更します。
  - フラッシュ/ROM セキュリティバイトを含むセクタを消去することにより, このバイトの値をクリアします。
  - "チップ消去" コマンドでフラッシュメモリを完全に消去することにより, フラッシュ/ROM セキュリティバイトの値をクリアします。
- 外部ベクタモード(MD2～MD0=000_BまたはMD2～MD0=001_BまたはMD2～MD0=110_B)を開始するとき,アプリケーションプログラムはフラッシュセキュリティを次の方法でのみ禁止することができます。
  - "チップ消去" コマンドでフラッシュメモリを完全に消去することにより, フラッシュ/ROM セキュリティバイトの値をクリアします。
- シリアル通信モード(MD2～MD0=010_B)を開始するとき,フラッシュセキュリティを次の方法で禁止することができます。
  - フラッシュセキュリティが解除された後に, フラッシュメモリ領域に書き込むことにより, フラッシュ/ROM セキュリティバイトの値を 99_H とは異なる値に変更します。
  - フラッシュセキュリティを解除した後, フラッシュ/ROM セキュリティバイトを含むセクタを消去することにより, このバイトの値をクリアします。
  - "チップ消去" コマンドでフラッシュメモリを完全に消去することにより, フラッシュ/ROM セキュリティバイトの値をクリアします。
- 並列フラッシュ書込みモード(MD2～MD0=111_B)を開始するとき,フラッシュセキュリティを次の方法でのみ禁止することができます。
  - "チップ消去" コマンドでフラッシュメモリを完全に消去することにより, フラッシュ/ROM セキュリティバイトの値をクリアします。

36. シリアルプログラミングの接続例

フラッシュメモリの回路内シリアルプログラミングのための MCU の接続方法について説明します。

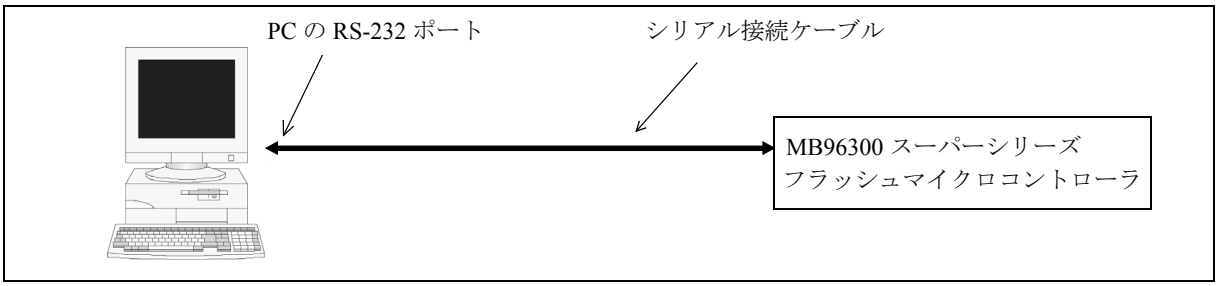
36.1 シリアルプログラミング接続の基本設定

MB96300 スーパーシリーズフラッシュマイクロコントローラは, フラッシュ ROM シリアルオンボードプログラミングに対応しています。ここでは, マイクロコントローラをプログラミング機器に接続する方法について説明します。

シリアルプログラミングの基本設定

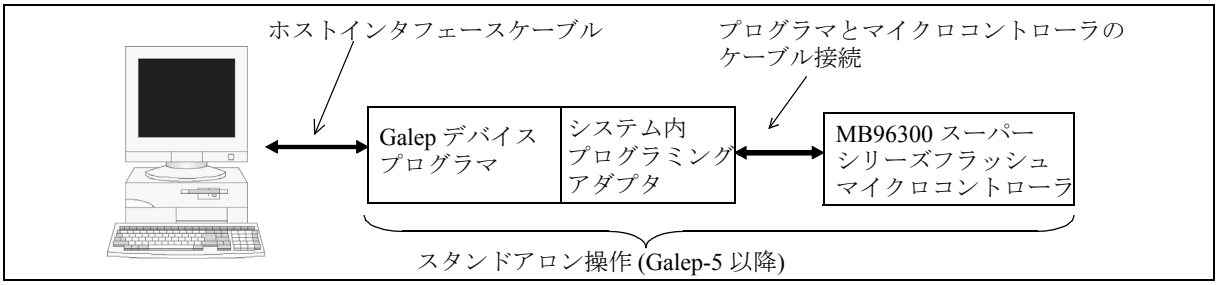
最小システムでは , MB96300 スーパーシリーズフラッシュマイクロプロセッサは非同期モードの USART 経由で PC の RS-232 ポートに接続されます。

Figure 36-1. PCのシリアルポートとMB96300スーパーシリーズフラッシュマイクロコントローラの接続



高速シリアルオンボードプログラミングには , 各種ベンダのプログラミングツールを使用できます。たとえば , CONITEC DATASYSTEMS の Galep デバイスプログラマは , Cypress MB96300 スーパーシリーズフラッシュメモリマイクロコントローラに対応しています。

Figure 36-2. CONITEC DATASYSTEMSのGalepデバイスプログラマとMB96300スーパーシリーズフラッシュマイクロコントローラの接続



＜注意事項＞

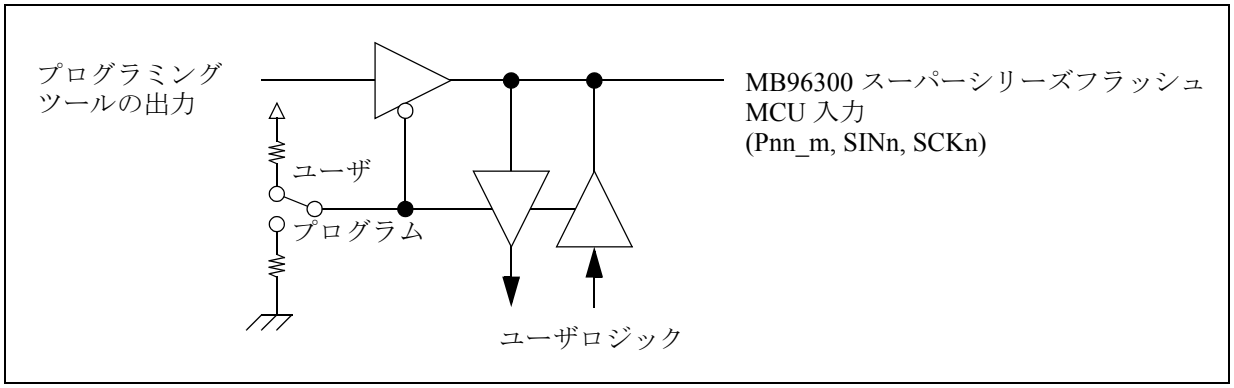
Galep デバイスプログラマの詳細については , CONITEC DATASYSTEMS を参照してください。

Table 36-1. Cypressシリアルオンボードプログラミングに使用する端子

端子	機能	追加情報
MD2, MD1 MD0	モード端子	MCU の動作モードを制御します。詳細については, 「8.4 ブートROMプログラムの実行と動作モード, およびROM構成ブロック」を参照してください。
X0, X1	発振端子	一部のプログラミングツールは MCU に水晶精度のタイミングが必要です。このため, オンチップ水晶発振回路を使用する必要があります。
Pnn_m	通信ハンドシェーキング	一部のプログラミングツールは, MCU との通信の高速化にハンドシェーク信号が必要です。ポートで推奨されるハンドシェークについては, データシートを参照してください。
$\overline{\text{RST}}$	リセット端子	一部のプログラミングツールでは, MCU のリセットラインの制御を必要とします。
SINn	シリアルデータ入力端子	プログラミングツールとMCUの間の通信には,USARTが使用されます。事前に定義されているUSARTのうち, どのUSARTがMCUのプログラミングツールと接続されているかは,MCUによって自動的に検出されます。MCUのプログラミングに使用できる USART チャネル番号については, データシートを参照してください。
SOTn	シリアルデータ出力端子	
SCKn	シリアルクロック信号 入力端子	
C	C 端子	この外部コンデンサ端子は , 電源の安定化に使用されます。 推奨される容量とタイプについては, データシートを参照してください。
V_{CC}	電源電圧端子	ユーザシステムから電源を供給した場合は, デバイスプログラマの電源を接続する必要はありません。デバイスプログラマの電源を接続するときは, ユーザ側の電源とショートしないように注意してください。
V_{SS}	GND 端子	フラッシュマイクロコンピュータプログラマと共通の接地。

デバイスプログラマによって駆動される信号がユーザシステムでも使用される場合は, 次の図に示される制御回路が必要です。この回路は , ジャンパの状態に応じてプログラミング接続を有効化または無効化します。

Figure 36-3. ユーザ回路とシリアルプログラミングに使用されるポートの接続



シリアルデータのボーレート

MCU は , プログラミングツールまたはプログラミングソフトウェアの実行 PC に合わせてシリアルデータのボーレートを自動的に決定します。

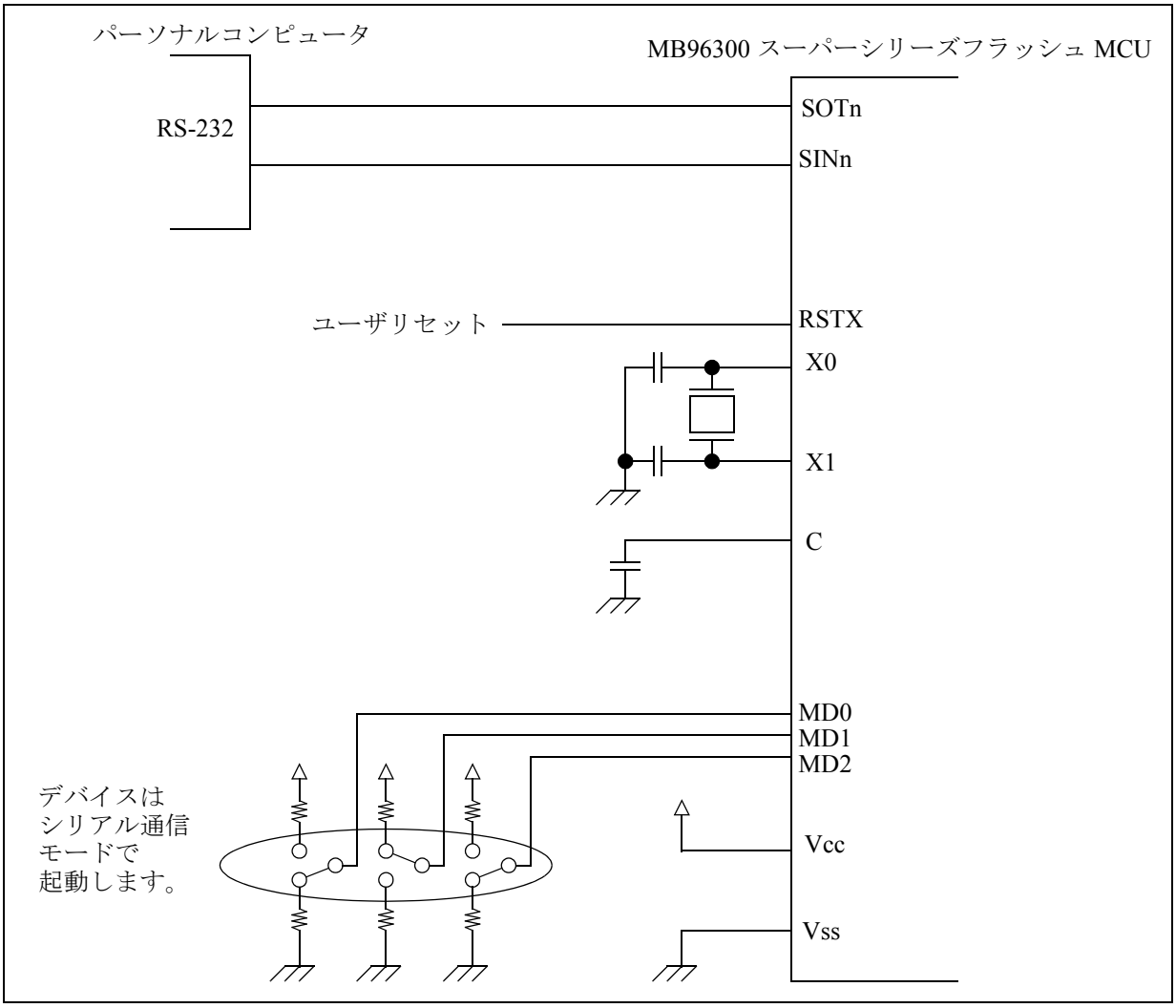
36.2 フラッシュマイクロコントローラをプログラミングするためのPC接続例

Figure 36-4 は，フラッシュマイクロコントローラをプログラミングするための PC との最小接続の例を示しています。

フラッシュマイクロコントローラをプログラミングするためのPCとの最小接続の例

PC とフラッシュ MCU の間の最小接続では , 非同期モードの USART のみの接続が必要です。フラッシュシリアルプログラミングに使用できる USART チャネルの概要については , データシートを参照してください。

Figure 36-4. フラッシュマイクロコントローラをプログラミングするためのPCとの最小接続の例.



36.3 フラッシュマイクロコントローラをプログラミングするためのプログラミングツールの接続例

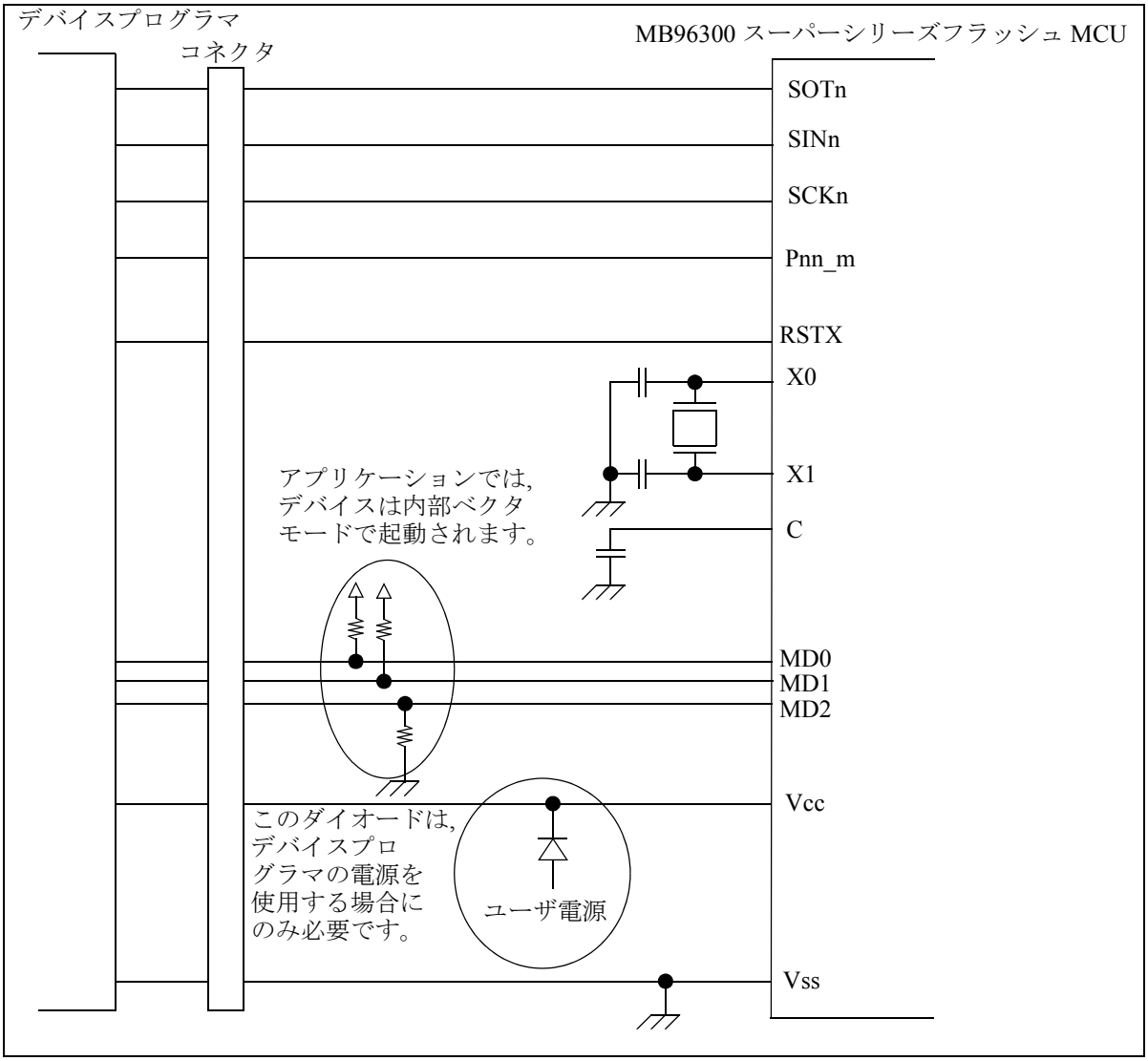
Figure 36-5 は，フラッシュマイクロコントローラをプログラミングするためのプログラミングツールの接続例を示しています。

フラッシュマイクロコントローラをプログラミングするためのプログラミングツールの接続例

高速プログラミングのためには , デバイスプログラマを次の例に示されるように接続する必要があります。フラッシュシリアルプログラミングに使用できる USART チャネルの概要については , データシートを参照してください。

デバイスプログラマと MCU の間のハンドシェークに使用するポートを決定するときは , データシートを参照してください。

Figure 36-5. アプリケーションの内部ベクタモードで起動されたフラッシュマイクロコントローラをプログラミングするための, デバイスプログラマの接続例.



37. 付録

I/O マップ,命令などを掲載します。

37.1 MB96V300のI/Oマップ

各リソースのレジスタに割り当てられるアドレスを示します。

37.1.1 MB96V300のI/Oマップ

Table 37-1. MB96V300BのI/Oマップ

アドレス	レジスタ	8ビット アクセス時の レジスタ略称名	16ビット アクセス時の レジスタ略称名	属性
000000H	I/Oポート P00 - ポートデータレジスタ	PDR00		RW
000001H	I/Oポート P01 - ポートデータレジスタ	PDR01		RW
000002H	I/Oポート P02 - ポートデータレジスタ	PDR02		RW
000003H	I/Oポート P03 - ポートデータレジスタ	PDR03		RW
000004H	I/Oポート P04 - ポートデータレジスタ	PDR04		RW
000005H	I/Oポート P05 - ポートデータレジスタ	PDR05		RW
000006H	I/Oポート P06 - ポートデータレジスタ	PDR06		RW
000007H	I/Oポート P07 - ポートデータレジスタ	PDR07		RW
000008H	I/Oポート P08 - ポートデータレジスタ	PDR08		RW
000009H	I/Oポート P09 - ポートデータレジスタ	PDR09		RW
00000AH	I/Oポート P10 - ポートデータレジスタ	PDR10		RW
00000BH	I/Oポート P11 - ポートデータレジスタ	PDR11		RW
00000CH	I/Oポート P12 - ポートデータレジスタ	PDR12		RW
00000DH	I/Oポート P13 - ポートデータレジスタ	PDR13		RW
00000EH	I/Oポート P14 - ポートデータレジスタ	PDR14		RW
00000FH	I/Oポート P15 - ポートデータレジスタ	PDR15		RW
000010H	I/Oポート P16 - ポートデータレジスタ	PDR16		RW
000011H	I/Oポート P17 - ポートデータレジスタ	PDR17		RW
000012H-000017H	予約			-
000018H	ADC0 - 制御ステータスレジスタ下位	ADCSL	ADCS	RW
000019H	ADC0 - 制御ステータスレジスタ上位	ADCSH		RW
00001AH	ADC0 - データレジスタ下位	ADCRL	ADCR	R
00001BH	ADC0 - データレジスタ上位	ADCRH		R
00001CH	ADC0 - 設定レジスタ		ADSR	RW

Table 37-1. MB96V300BのI/Oマップ

アドレス	レジスタ	8ビット アクセス時の レジスタ略称名	16ビット アクセス時の レジスタ略称名	属性
00001DH	ADC0 - 設定レジスタ			RW
00001EH	ADC0 - 拡張構成レジスタ	ADECR		RW
00001FH	予約			-
000020H	FRT0 - フリーランタイマのデータレジスタ		TCDT0	RW
000021H	FRT0 - フリーランタイマのデータレジスタ			RW
000022H	FRT0 - フリーランタイマの制御ステータスレジスタ下位	TCCSL0	TCCS0	RW
000023H	FRT0 - フリーランタイマの制御ステータスレジスタ上位	TCCSH0		RW
000024H	FRT1 - フリーランタイマのデータレジスタ		TCDT1	RW
000025H	FRT1 - フリーランタイマのデータレジスタ			RW
000026H	FRT1 - フリーランタイマの制御ステータスレジスタ下位	TCCSL1	TCCS1	RW
000027H	FRT1 - フリーランタイマの制御ステータスレジスタ上位	TCCSH1		RW
000028H	OCU0 - 出力比較制御ステータス	OCS0		RW
000029H	OCU1 - 出力比較制御ステータス	OCS1		RW
00002AH	OCU0 - 比較レジスタ		OCCP0	RW
00002BH	OCU0 - 比較レジスタ			RW
00002CH	OCU1 - 比較レジスタ		OCCP1	RW
00002DH	OCU1 - 比較レジスタ			RW
00002EH	OCU2 - 出力比較制御ステータス	OCS2		RW
00002FH	OCU3 - 出力比較制御ステータス	OCS3		RW
000030H	OCU2 - 比較レジスタ		OCCP2	RW
000031H	OCU2 - 比較レジスタ			RW
000032H	OCU3 - 比較レジスタ		OCCP3	RW
000033H	OCU3 - 比較レジスタ			RW
000034H	OCU4 - 出力比較制御ステータス	OCS4		RW
000035H	OCU5 - 出力比較制御ステータス	OCS5		RW
000036H	OCU4 - 比較レジスタ		OCCP4	RW
000037H	OCU4 - 比較レジスタ			RW
000038H	OCU5 - 比較レジスタ		OCCP5	RW
000039H	OCU5 - 比較レジスタ			RW
00003AH	OCU6 - 出力比較制御ステータス	OCS6		RW
00003BH	OCU7 - 出力比較制御ステータス	OCS7		RW
00003CH	OCU6 - 比較レジスタ		OCCP6	RW
00003DH	OCU6 - 比較レジスタ			RW
00003EH	OCU7 - 比較レジスタ		OCCP7	RW
00003FH	OCU7 - 比較レジスタ			RW
000040H	ICU0/ICU1 - 制御ステータスレジスタ	ICS01		RW
000041H	ICU0/ICU1 - エッジレジスタ	ICE01		RW
000042H	ICU0 - キャプチャレジスタ下位	IPCPL0	IPCP0	R
000043H	ICU0 - キャプチャレジスタ上位	IPCPH0		R
000044H	ICU1 - キャプチャレジスタ下位	IPCPL1	IPCP1	R
000045H	ICU1 - キャプチャレジスタ上位	IPCPH1		R
000046H	ICU2/ICU3 - 制御ステータスレジスタ	ICS23		RW

Table 37-1. MB96V300BのI/Oマップ

アドレス	レジスタ	8ビット アクセス時の レジスタ略称名	16ビット アクセス時の レジスタ略称名	属性
000047H	ICU2/ICU3 - エッジレジスタ	ICE23		RW
000048H	ICU2 - キャプチャレジスタ下位	IPCPL2	IPCP2	R
000049H	ICU2 - キャプチャレジスタ上位	IPCPH2		R
00004AH	ICU3 - キャプチャレジスタ下位	IPCPL3	IPCP3	R
00004BH	ICU3 - キャプチャレジスタ上位	IPCPH3		R
00004CH	ICU4/ICU5 - 制御ステータスレジスタ	ICS45		RW
00004DH	ICU4/ICU5 - エッジレジスタ	ICE45		RW
00004EH	ICU4 - キャプチャレジスタ下位	IPCPL4	IPCP4	R
00004FH	ICU4 - キャプチャレジスタ上位	IPCPH4		R
000050H	ICU5 - キャプチャレジスタ下位	IPCPL5	IPCP5	R
000051H	ICU5 - キャプチャレジスタ上位	IPCPH5		R
000052H	ICU6/ICU7 - 制御ステータスレジスタ	ICS67		RW
000053H	ICU6/ICU7 - エッジレジスタ	ICE67		RW
000054H	ICU6 - キャプチャレジスタ下位	IPCPL6	IPCP6	R
000055H	ICU6 - キャプチャレジスタ上位	IPCPH6		R
000056H	ICU7 - キャプチャレジスタ下位	IPCPL7	IPCP7	R
000057H	ICU7 - キャプチャレジスタ上位	IPCPH7		R
000058H	EXTINT0 - 外部割込み許可レジスタ	ENIR0		RW
000059H	EXTINT0 - 外部割込み割込み要求レジスタ	EIRR0		RW
00005AH	EXTINT0 - 外部割込みレベル選択下位	ELVRL0	ELVR0	RW
00005BH	EXTINT0 - 外部割込みレベル選択上位	ELVRH0		RW
00005CH	EXTINT1 - 外部割込み許可レジスタ	ENIR1		RW
00005DH	EXTINT1 - 外部割込み割込み要求レジスタ	EIRR1		RW
00005EH	EXTINT1 - 外部割込みレベル選択下位	ELVRL1	ELVR1	RW
00005FH	EXTINT1 - 外部割込みレベル選択上位	ELVRH1		RW
000060H	RLT0 - タイマ制御ステータスレジスタ下位	TMCSRL0	TMCSR0	RW
000061H	RLT0 - タイマ制御ステータスレジスタ上位	TMCSRH0		RW
000062H	RLT0 - リロードレジスタ - 書込み用		TMRLR0	W
000062H	RLT0 - リロードレジスタ - 読出し用		TMR0	R
000063H	RLT0 - リロードレジスタ - 書込み用			W
000063H	RLT0 - リロードレジスタ - 読出し用			R
000064H	RLT1 - タイマ制御ステータスレジスタ下位	TMCSRL1	TMCSR1	RW
000065H	RLT1 - タイマ制御ステータスレジスタ上位	TMCSRH1		RW
000066H	RLT1 - リロードレジスタ - 書込み用		TMRLR1	W
000066H	RLT1 - リロードレジスタ - 読出し用		TMR1	R
000067H	RLT1 - リロードレジスタ - 書込み用			W
000067H	RLT1 - リロードレジスタ - 読出し用			R
000068H	RLT2 - タイマ制御ステータスレジスタ下位	TMCSRL2	TMCSR2	RW
000069H	RLT2 - タイマ制御ステータスレジスタ上位	TMCSRH2		RW
00006AH	RLT2 - リロードレジスタ - 書込み用		TMRLR2	W
00006AH	RLT2 - リロードレジスタ - 読出し用		TMR2	R
00006BH	RLT2 - リロードレジスタ - 書込み用			W

Table 37-1. MB96V300BのI/Oマップ

アドレス	レジスタ	8ビット アクセス時の レジスタ略称名	16ビット アクセス時の レジスタ略称名	属性
00006BH	RLT2 - リロードレジスタ - 読出し用			R
00006CH	RLT3 - タイマ制御ステータスレジスタ下位	TMCSRL3	TMCSR3	RW
00006DH	RLT3 - タイマ制御ステータスレジスタ上位	TMCSRH3		RW
00006EH	RLT3 - リロードレジスタ - 書込み用		TMRLR3	W
00006EH	RLT3 - リロードレジスタ - 読出し用		TMR3	R
00006FH	RLT3 - リロードレジスタ - 書込み用			W
00006FH	RLT3 - リロードレジスタ - 読出し用			R
000070H	RLT6 - タイマ制御ステータスレジスタ下位 (PPGのために割り当てられたRLT)	TMCSRL6	TMCSR6	RW
000071H	RLT6 - タイマ制御ステータスレジスタ上位 (PPGのために割り当てられたRLT)	TMCSRH6		RW
000072H	RLT6 - リロードレジスタ (PPGのために割り当てられたRLT) - 書込み用		TMRLR6	W
000072H	RLT6 - リロードレジスタ (PPGのために割り当てられたRLT) - 読出し用		TMR6	R
000073H	RLT6 - リロードレジスタ (PPGのために割り当てられたRLT) - 書込み用			W
000073H	RLT6 - リロードレジスタ (PPGのために割り当てられたRLT) - 読出し用			R
000074H	PPG3-PPG0 - 汎用制御レジスタ 1 下位	GCN1L0	GCN10	RW
000075H	PPG3-PPG0 - 汎用制御レジスタ 1 上位	GCN1H0		RW
000076H	PPG3-PPG0 - 汎用制御レジスタ 2 下位	GCN2L0	GCN20	RW
000077H	PPG3-PPG0 - 汎用制御レジスタ 2 上位	GCN2H0		RW
000078H	PPG0 - タイマレジスタ		PTMR0	R
000079H	PPG0 - タイマレジスタ			R
00007AH	PPG0 - 周期設定レジスタ		PCSR0	W
00007BH	PPG0 - 周期設定レジスタ			W
00007CH	PPG0 - デューティ周期レジスタ		PDUT0	W
00007DH	PPG0 - デューティ周期レジスタ			W
00007EH	PPG0 - 制御ステータスレジスタ下位	PCNL0	PCN0	RW
00007FH	PPG0 - 制御ステータスレジスタ上位	PCNH0		RW
000080H	PPG1 - タイマレジスタ		PTMR1	R
000081H	PPG1 - タイマレジスタ			R
000082H	PPG1 - 周期設定レジスタ		PCSR1	W
000083H	PPG1 - 周期設定レジスタ			W
000084H	PPG1 - デューティ周期レジスタ		PDUT1	W
000085H	PPG1 - デューティ周期レジスタ			W
000086H	PPG1 - 制御ステータスレジスタ下位	PCNL1	PCN1	RW
000087H	PPG1 - 制御ステータスレジスタ上位	PCNH1		RW
000088H	PPG2 - タイマレジスタ		PTMR2	R
000089H	PPG2 - タイマレジスタ			R
00008AH	PPG2 - 周期設定レジスタ		PCSR2	W
00008BH	PPG2 - 周期設定レジスタ			W
00008CH	PPG2 - デューティ周期レジスタ		PDUT2	W
00008DH	PPG2 - デューティ周期レジスタ			W
00008EH	PPG2 - 制御ステータスレジスタ下位	PCNL2	PCN2	RW
00008FH	PPG2 - 制御ステータスレジスタ上位	PCNH2		RW
000090H	PPG3 - タイマレジスタ		PTMR3	R

Table 37-1. MB96V300BのI/Oマップ

アドレス	レジスタ	8ビット アクセス時の レジスタ略称名	16ビット アクセス時の レジスタ略称名	属性
000091H	PPG3 - タイマレジスタ			R
000092H	PPG3 - 周期設定レジスタ		PCSR3	W
000093H	PPG3 - 周期設定レジスタ			W
000094H	PPG3 - デューティ周期レジスタ		PDUT3	W
000095H	PPG3 - デューティ周期レジスタ			W
000096H	PPG3 - 制御ステータスレジスタ下位	PCNL3	PCN3	RW
000097H	PPG3 - 制御ステータスレジスタ上位	PCNH3		RW
000098H	PPG7-PPG4 - 汎用制御レジスタ 1 下位	GCN1L1	GCN11	RW
000099H	PPG7-PPG4 - 汎用制御レジスタ 1 上位	GCN1H1		RW
00009AH	PPG7-PPG4 - 汎用制御レジスタ 2 下位	GCN2L1	GCN21	RW
00009BH	PPG7-PPG4 - 汎用制御レジスタ 2 上位	GCN2H1		RW
00009CH	PPG4 - タイマレジスタ		PTMR4	R
00009DH	PPG4 - タイマレジスタ			R
00009EH	PPG4 - 周期設定レジスタ		PCSR4	W
00009FH	PPG4 - 周期設定レジスタ			W
0000A0H	PPG4 - デューティ周期レジスタ		PDUT4	W
0000A1H	PPG4 - デューティ周期レジスタ			W
0000A2H	PPG4 - 制御ステータスレジスタ下位	PCNL4	PCN4	RW
0000A3H	PPG4 - 制御ステータスレジスタ上位	PCNH4		RW
0000A4H	PPG5 - タイマレジスタ		PTMR5	R
0000A5H	PPG5 - タイマレジスタ			R
0000A6H	PPG5 - 周期設定レジスタ		PCSR5	W
0000A7H	PPG5 - 周期設定レジスタ			W
0000A8H	PPG5 - デューティ周期レジスタ		PDUT5	W
0000A9H	PPG5 - デューティ周期レジスタ			W
0000AAH	PPG5 - 制御ステータスレジスタ下位	PCNL5	PCN5	RW
0000ABH	PPG5 - 制御ステータスレジスタ上位	PCNH5		RW
0000ACH	I2C0 - バス状態レジスタ	IBSR0		R
0000ADH	I2C0 - バス制御レジスタ	IBCR0		RW
0000AEH	I2C0 - 10ビットスレーブアドレスレジスタ下位	ITBAL0	ITBA0	RW
0000AFH	I2C0 - 10ビットスレーブアドレスレジスタ上位	ITBAH0		RW
0000B0H	I2C0 - 10ビットアドレスマスクレジスタ下位	ITMKL0	ITMK0	RW
0000B1H	I2C0 - 10ビットアドレスマスクレジスタ上位	ITMKH0		RW
0000B2H	I2C0 - 7ビットスレーブアドレスレジスタ	ISBA0		RW
0000B3H	I2C0 - 7ビットアドレスマスクレジスタ	ISMK0		RW
0000B4H	I2C0 - データレジスタ	IDAR0		RW
0000B5H	I2C0 - クロック制御レジスタ	ICCR0		RW
0000B6H	I2C1 - バスステータスレジスタ	IBSR1		R
0000B7H	I2C1 - バス制御レジスタ	IBCR1		RW
0000B8H	I2C1 - 10ビットスレーブアドレスレジスタ下位	ITBAL1	ITBA1	RW
0000B9H	I2C1 - 10ビットスレーブアドレスレジスタ上位	ITBAH1		RW
0000BAH	I2C1 - 10ビットアドレスマスクレジスタ下位	ITMKL1	ITMK1	RW

Table 37-1. MB96V300BのI/Oマップ

アドレス	レジスタ	8ビット アクセス時の レジスタ略称名	16ビット アクセス時の レジスタ略称名	属性
0000BBH	I2C1 - 10ビットアドレスマスクレジスタ上位	ITMKH1		RW
0000BCH	I2C1 - 7ビットスレーブアドレスレジスタ	ISBA1		RW
0000BDH	I2C1 - 7ビットアドレスマスクレジスタ	ISMK1		RW
0000BEH	I2C1 - データレジスタ	IDAR1		RW
0000BFH	I2C1 - クロック制御レジスタ	ICCR1		RW
0000C0H	USART0 - シリアルモードレジスタ	SMR0		RW
0000C1H	USART0 - シリアル制御レジスタ	SCR0		RW
0000C2H	USART0 - TXレジスタ	TDR0		W
0000C2H	USART0 - RXレジスタ	RDR0		R
0000C3H	USART0 - シリアルステータス	SSR0		RW
0000C4H	USART0 - 制御/比較レジスタ	ECCR0		RW
0000C5H	USART0 - 拡張ステータスレジスタ	ESCR0		RW
0000C6H	USART0 - ボーレートジェネレータレジスタ下位	BGRL0	BGR0	RW
0000C7H	USART0 - ボーレートジェネレータレジスタ上位	BGRH0		RW
0000C8H	USART0 - 拡張シリアル割込みレジスタ	ESIR0		RW
0000C9H	予約			-
0000CAH	USART1 - シリアルモードレジスタ	SMR1		RW
0000CBH	USART1 - シリアル制御レジスタ	SCR1		RW
0000CCH	USART1 - TXレジスタ	TDR1		W
0000CCH	USART1 - RXレジスタ	RDR1		R
0000CDH	USART1 - シリアルステータス	SSR1		RW
0000CEH	USART1 - 制御/比較レジスタ	ECCR1		RW
0000CFH	USART1 - 拡張ステータスレジスタ	ESCR1		RW
0000D0H	USART1 - ボーレートジェネレータレジスタ下位	BGRL1	BGR1	RW
0000D1H	USART1 - ボーレートジェネレータレジスタ上位	BGRH1		RW
0000D2H	USART1 - 拡張シリアル割込みレジスタ	ESIR1		RW
0000D3H	予約			-
0000D4H	USART2 - シリアルモードレジスタ	SMR2		RW
0000D5H	USART2 - シリアル制御レジスタ	SCR2		RW
0000D6H	USART2 - TXレジスタ	TDR2		W
0000D6H	USART2 - RXレジスタ	RDR2		R
0000D7H	USART2 - シリアルステータス	SSR2		RW
0000D8H	USART2 - 制御/比較レジスタ	ECCR2		RW
0000D9H	USART2 - 拡張ステータスレジスタ	ESCR2		RW
0000DAH	USART2 - ボーレートジェネレータレジスタ下位	BGRL2	BGR2	RW
0000DBH	USART2 - ボーレートジェネレータレジスタ上位	BGRH2		RW
0000DCH	USART2 - 拡張シリアル割込みレジスタ	ESIR2		RW
0000DDH	予約			-
0000DEH	USART3 - シリアルモードレジスタ	SMR3		RW
0000DFH	USART3 - シリアル制御レジスタ	SCR3		RW
0000E0H	USART3 - TXレジスタ	TDR3		W
0000E0H	USART3 - RXレジスタ	RDR3		R

Table 37-1. MB96V300BのI/Oマップ

アドレス	レジスタ	8ビット アクセス時の レジスタ略称名	16ビット アクセス時の レジスタ略称名	属性
0000E1H	USART3 - シリアルステータス	SSR3		RW
0000E2H	USART3 - 制御/比較レジスタ	ECCR3		RW
0000E3H	USART3 - 拡張ステータスレジスタ	ESCR3		RW
0000E4H	USART3 - ボーレートジェネレータレジスタ下位	BGRL3	BGR3	RW
0000E5H	USART3 - ボーレートジェネレータレジスタ上位	BGRH3		RW
0000E6H	USART3 - 拡張シリアル割込みレジスタ	ESIR3		RW
0000E7H-0000EFH	予約			-
0000F0H-0000FFH	外部バス領域	EXTBUS0		RW
000100H	DMA0 - バッファアドレスポインタ下位バイト	BAPL0		RW
000101H	DMA0 - バッファアドレスポインタ中位バイト	BAPM0		RW
000102H	DMA0 - バッファアドレスポインタ上位バイト	BAPH0		RW
000103H	DMA0 - DMA制御レジスタ	DMACS0		RW
000104H	DMA0 - I/Oレジスタアドレスポインタ下位バイト	IOAL0	IOA0	RW
000105H	DMA0 - I/Oレジスタアドレスポインタ上位バイト	IOAH0		RW
000106H	DMA0 - データカウンタ下位バイト	DCTL0	DCT0	RW
000107H	DMA0 - データカウンタ上位バイト	DCTH0		RW
000108H	DMA1 - バッファアドレスポインタ下位バイト	BAPL1		RW
000109H	DMA1 - バッファアドレスポインタ中位バイト	BAPM1		RW
00010AH	DMA1 - バッファアドレスポインタ上位バイト	BAPH1		RW
00010BH	DMA1 - DMA制御レジスタ	DMACS1		RW
00010CH	DMA1 - I/Oレジスタアドレスポインタ下位バイト	IOAL1	IOA1	RW
00010DH	DMA1 - I/Oレジスタアドレスポインタ上位バイト	IOAH1		RW
00010EH	DMA1 - データカウンタ下位バイト	DCTL1	DCT1	RW
00010FH	DMA1 - データカウンタ上位バイト	DCTH1		RW
000110H	DMA2 - バッファアドレスポインタ下位バイト	BAPL2		RW
000111H	DMA2 - バッファアドレスポインタ中位バイト	BAPM2		RW
000112H	DMA2 - バッファアドレスポインタ上位バイト	BAPH2		RW
000113H	DMA2 - DMA制御レジスタ	DMACS2		RW
000114H	DMA2 - I/Oレジスタアドレスポインタ下位バイト	IOAL2	IOA2	RW
000115H	DMA2 - I/Oレジスタアドレスポインタ上位バイト	IOAH2		RW
000116H	DMA2 - データカウンタ下位バイト	DCTL2	DCT2	RW
000117H	DMA2 - データカウンタ上位バイト	DCTH2		RW
000118H	DMA3 - バッファアドレスポインタ下位バイト	BAPL3		RW
000119H	DMA3 - バッファアドレスポインタ中位バイト	BAPM3		RW
00011AH	DMA3 - バッファアドレスポインタ上位バイト	BAPH3		RW
00011BH	DMA3 - DMA制御レジスタ	DMACS3		RW
00011CH	DMA3 - I/Oレジスタアドレスポインタ下位バイト	IOAL3	IOA3	RW
00011DH	DMA3 - I/Oレジスタアドレスポインタ上位バイト	IOAH3		RW
00011EH	DMA3 - データカウンタ下位バイト	DCTL3	DCT3	RW
00011FH	DMA3 - データカウンタ上位バイト	DCTH3		RW
000120H	DMA4 - バッファアドレスポインタ下位バイト	BAPL4		RW

Table 37-1. MB96V300BのI/Oマップ

アドレス	レジスタ	8ビット アクセス時の レジスタ略称名	16ビット アクセス時の レジスタ略称名	属性
000121H	DMA4 - バッファアドレスポインタ中位バイト	BAPM4		RW
000122H	DMA4 - バッファアドレスポインタ上位バイト	BAPH4		RW
000123H	DMA4 - DMA制御レジスタ	DMACS4		RW
000124H	DMA4 - I/Oレジスタアドレスポインタ下位バイト	IOAL4	IOA4	RW
000125H	DMA4 - I/Oレジスタアドレスポインタ上位バイト	IOAH4		RW
000126H	DMA4 - データカウンタ下位バイト	DCTL4	DCT4	RW
000127H	DMA4 - データカウンタ上位バイト	DCTH4		RW
000128H	DMA5 - バッファアドレスポインタ下位バイト	BAPL5		RW
000129H	DMA5 - バッファアドレスポインタ中位バイト	BAPM5		RW
00012AH	DMA5 - バッファアドレスポインタ上位バイト	BAPH5		RW
00012BH	DMA5 - DMA制御レジスタ	DMACS5		RW
00012CH	DMA5 - I/Oレジスタアドレスポインタ下位バイト	IOAL5	IOA5	RW
00012DH	DMA5 - I/Oレジスタアドレスポインタ上位バイト	IOAH5		RW
00012EH	DMA5 - データカウンタ下位バイト	DCTL5	DCT5	RW
00012FH	DMA5 - データカウンタ上位バイト	DCTH5		RW
000130H	DMA6 - バッファアドレスポインタ下位バイト	BAPL6		RW
000131H	DMA6 - バッファアドレスポインタ中位バイト	BAPM6		RW
000132H	DMA6 - バッファアドレスポインタ上位バイト	BAPH6		RW
000133H	DMA6 - DMA制御レジスタ	DMACS6		RW
000134H	DMA6 - I/Oレジスタアドレスポインタ下位バイト	IOAL6	IOA6	RW
000135H	DMA6 - I/Oレジスタアドレスポインタ上位バイト	IOAH6		RW
000136H	DMA6 - データカウンタ下位バイト	DCTL6	DCT6	RW
000137H	DMA6 - データカウンタ上位バイト	DCTH6		RW
000138H	DMA7 - バッファアドレスポインタ下位バイト	BAPL7		RW
000139H	DMA7 - バッファアドレスポインタ中位バイト	BAPM7		RW
00013AH	DMA7 - バッファアドレスポインタ上位バイト	BAPH7		RW
00013BH	DMA7 - DMA制御レジスタ	DMACS7		RW
00013CH	DMA7 - I/Oレジスタアドレスポインタ下位バイト	IOAL7	IOA7	RW
00013DH	DMA7 - I/Oレジスタアドレスポインタ上位バイト	IOAH7		RW
00013EH	DMA7 - データカウンタ下位バイト	DCTL7	DCT7	RW
00013FH	DMA7 - データカウンタ上位バイト	DCTH7		RW
000140H	DMA8 - バッファアドレスポインタ下位バイト	BAPL8		RW
000141H	DMA8 - バッファアドレスポインタ中位バイト	BAPM8		RW
000142H	DMA8 - バッファアドレスポインタ上位バイト	BAPH8		RW
000143H	DMA8 - DMA制御レジスタ	DMACS8		RW
000144H	DMA8 - I/Oレジスタアドレスポインタ下位バイト	IOAL8	IOA8	RW
000145H	DMA8 - I/Oレジスタアドレスポインタ上位バイト	IOAH8		RW
000146H	DMA8 - データカウンタ下位バイト	DCTL8	DCT8	RW
000147H	DMA8 - データカウンタ上位バイト	DCTH8		RW
000148H	DMA9 - バッファアドレスポインタ下位バイト	BAPL9		RW
000149H	DMA9 - バッファアドレスポインタ中位バイト	BAPM9		RW
00014AH	DMA9 - バッファアドレスポインタ上位バイト	BAPH9		RW

Table 37-1. MB96V300BのI/Oマップ

アドレス	レジスタ	8ビット アクセス時の レジスタ略称名	16ビット アクセス時の レジスタ略称名	属性
00014BH	DMA9 - DMA制御レジスタ	DMACS9		RW
00014CH	DMA9 - I/Oレジスタアドレスポインタ下位バイト	IOAL9	IOA9	RW
00014DH	DMA9 - I/Oレジスタアドレスポインタ上位バイト	IOAH9		RW
00014EH	DMA9 - データカウンタ下位バイト	DCTL9	DCT9	RW
00014FH	DMA9 - データカウンタ上位バイト	DCTH9		RW
000150H	DMA10 - バッファアドレスポインタ下位バイト	BAPL10		RW
000151H	DMA10 - バッファアドレスポインタ中位バイト	BAPM10		RW
000152H	DMA10 - バッファアドレスポインタ上位バイト	BAPH10		RW
000153H	DMA10 - DMA制御レジスタ	DMACS10		RW
000154H	DMA10 - I/Oレジスタアドレスポインタ下位バイト	IOAL10	IOA10	RW
000155H	DMA10 - I/Oレジスタアドレスポインタ上位バイト	IOAH10		RW
000156H	DMA10 - データカウンタ下位バイト	DCTL10	DCT10	RW
000157H	DMA10 - データカウンタ上位バイト	DCTH10		RW
000158H	DMA11 - バッファアドレスポインタ下位バイト	BAPL11		RW
000159H	DMA11 - バッファアドレスポインタ中位バイト	BAPM11		RW
00015AH	DMA11 - バッファアドレスポインタ上位バイト	BAPH11		RW
00015BH	DMA11 - DMA制御レジスタ	DMACS11		RW
00015CH	DMA11 - I/Oレジスタアドレスポインタ下位バイト	IOAL11	IOA11	RW
00015DH	DMA11 - I/Oレジスタアドレスポインタ上位バイト	IOAH11		RW
00015EH	DMA11 - データカウンタ下位バイト	DCTL11	DCT11	RW
00015FH	DMA11 - データカウンタ上位バイト	DCTH11		RW
000160H	DMA12 - バッファアドレスポインタ下位バイト	BAPL12		RW
000161H	DMA12 - バッファアドレスポインタ中位バイト	BAPM12		RW
000162H	DMA12 - バッファアドレスポインタ上位バイト	BAPH12		RW
000163H	DMA12 - DMA制御レジスタ	DMACS12		RW
000164H	DMA12 - I/Oレジスタアドレスポインタ下位バイト	IOAL12	IOA12	RW
000165H	DMA12 - I/Oレジスタアドレスポインタ上位バイト	IOAH12		RW
000166H	DMA12 - データカウンタ下位バイト	DCTL12	DCT12	RW
000167H	DMA12 - データカウンタ上位バイト	DCTH12		RW
000168H	DMA13 - バッファアドレスポインタ下位バイト	BAPL13		RW
000169H	DMA13 - バッファアドレスポインタ中位バイト	BAPM13		RW
00016AH	DMA13 - バッファアドレスポインタ上位バイト	BAPH13		RW
00016BH	DMA13 - DMA制御レジスタ	DMACS13		RW
00016CH	DMA13 - I/Oレジスタアドレスポインタ下位バイト	IOAL13	IOA13	RW
00016DH	DMA13 - I/Oレジスタアドレスポインタ上位バイト	IOAH13		RW
00016EH	DMA13 - データカウンタ下位バイト	DCTL13	DCT13	RW
00016FH	DMA13 - データカウンタ上位バイト	DCTH13		RW
000170H	DMA14 - バッファアドレスポインタ下位バイト	BAPL14		RW
000171H	DMA14 - バッファアドレスポインタ中位バイト	BAPM14		RW
000172H	DMA14 - バッファアドレスポインタ上位バイト	BAPH14		RW
000173H	DMA14 - DMA制御レジスタ	DMACS14		RW
000174H	DMA14 - I/Oレジスタアドレスポインタ下位バイト	IOAL14	IOA14	RW

Table 37-1. MB96V300BのI/Oマップ

アドレス	レジスタ	8ビット アクセス時の レジスタ略称名	16ビット アクセス時の レジスタ略称名	属性
000175H	DMA14 - I/Oレジスタアドレスポインタ上位バイト	IOAH14		RW
000176H	DMA14 - データカウンタ下位バイト	DCTL14	DCT14	RW
000177H	DMA14 - データカウンタ上位バイト	DCTH14		RW
000178H	DMA15 - バッファアドレスポインタ下位バイト	BAPL15		RW
000179H	DMA15 - バッファアドレスポインタ中位バイト	BAPM15		RW
00017AH	DMA15 - バッファアドレスポインタ上位バイト	BAPH15		RW
00017BH	DMA15 - DMA制御レジスタ	DMACS15		RW
00017CH	DMA15 - I/Oレジスタアドレスポインタ下位バイト	IOAL15	IOA15	RW
00017DH	DMA15 - I/Oレジスタアドレスポインタ上位バイト	IOAH15		RW
00017EH	DMA15 - データカウンタ下位バイト	DCTL15	DCT15	RW
00017FH	DMA15 - データカウンタ上位バイト	DCTH15		RW
000180H-00037FH	CPU - 汎用レジスタ (RAMアクセス)	GPR_RAM		RW
000380H	DMA0 - DMAのための割込み選択(ch.0)	DISEL0		RW
000381H	DMA1 - DMAのための割込み選択(ch.1)	DISEL1		RW
000382H	DMA2 - DMAのための割込み選択(ch.2)	DISEL2		RW
000383H	DMA3 - DMAのための割込み選択(ch.3)	DISEL3		RW
000384H	DMA4 - DMAのための割込み選択(ch.4)	DISEL4		RW
000385H	DMA5 - DMAのための割込み選択(ch.5)	DISEL5		RW
000386H	DMA6 - DMAのための割込み選択(ch.6)	DISEL6		RW
000387H	DMA7 - DMAのための割込み選択(ch.7)	DISEL7		RW
000388H	DMA8 - DMAのための割込み選択(ch.8)	DISEL8		RW
000389H	DMA9 - DMAのための割込み選択(ch.9)	DISEL9		RW
00038AH	DMA10 - DMAのための割込み選択(ch.10)	DISEL10		RW
00038BH	DMA11 - DMAのための割込み選択(ch.11)	DISEL11		RW
00038CH	DMA12 - DMAのための割込み選択(ch.12)	DISEL12		RW
00038DH	DMA13 - DMAのための割込み選択(ch.13)	DISEL13		RW
00038EH	DMA14 - DMAのための割込み選択(ch.14)	DISEL14		RW
00038FH	DMA15 - DMAのための割込み選択(ch.15)	DISEL15		RW
000390H	DMAのためのDMAステータスレジスタ下位バイト	DSRL	DSR	RW
000391H	DMAのためのDMAステータスレジスタ上位バイト	DSRH		RW
000392H	DMAのためのDMA停止ステータスレジスタ下位バイト	DSSRL	DSSR	RW
000393H	DMAのためのDMA停止ステータスレジスタ上位バイト	DSSRH		RW
000394H	DMAのためのDMA許可レジスタ下位バイト	DERL	DER	RW
000395H	DMAのためのDMA許可レジスタ上位バイト	DERH		RW
000396H-00039FH	予約			-
0003A0H	割込みレベルレジスタ	ILR	ICR	RW
0003A1H	割込みインデックスレジスタ	IDX		RW
0003A2H	割込みベクタテーブルベースレジスタ下位	TBRL	TBR	RW
0003A3H	割込みベクタテーブルベースレジスタ上位	TBRH		RW
0003A4H	遅延割込みレジスタ	DIRR		RW
0003A5H	マスク不可割込みレジスタ	NMI		RW

Table 37-1. MB96V300BのI/Oマップ

アドレス	レジスタ	8ビット アクセス時の レジスタ略称名	16ビット アクセス時の レジスタ略称名	属性
0003A6H-0003ABH	予約			-
0003ACH	EDSUコミュニケーション割込み選択下位	EDSU2L	EDSU2	RW
0003ADH	EDSUコミュニケーション割込み選択上位	EDSU2H		RW
0003AEH	ROMミラー制御レジスタ	ROMM		RW
0003AFH	EDSU構成レジスタ	EDSU		RW
0003B0H	メモリパッチ制御/ステータスレジスタ ch 0/1		PFCS0	RW
0003B1H	メモリパッチ制御/ステータスレジスタ ch 0/1			RW
0003B2H	メモリパッチ制御/ステータスレジスタ ch 2/3		PFCS1	RW
0003B3H	メモリパッチ制御/ステータスレジスタ ch 2/3			RW
0003B4H	メモリパッチ制御/ステータスレジスタ ch 4/5		PFCS2	RW
0003B5H	メモリパッチ制御/ステータスレジスタ ch 4/5			RW
0003B6H	メモリパッチ制御/ステータスレジスタ ch 6/7		PFCS3	RW
0003B7H	メモリパッチ制御/ステータスレジスタ ch 6/7			RW
0003B8H	メモリパッチ機能 - パッチアドレス 0 下位	PFAL0		RW
0003B9H	メモリパッチ機能 - パッチアドレス 0 中位	PFAM0		RW
0003BAH	メモリパッチ機能 - パッチアドレス 0 上位	PFAH0		RW
0003BBH	メモリパッチ機能 - パッチアドレス 1 下位	PFAL1		RW
0003BCH	メモリパッチ機能 - パッチアドレス 1 中位	PFAM1		RW
0003BDH	メモリパッチ機能 - パッチアドレス 1 上位	PFAH1		RW
0003BEH	メモリパッチ機能 - パッチアドレス 2 下位	PFAL2		RW
0003BFH	メモリパッチ機能 - パッチアドレス 2 中位	PFAM2		RW
0003C0H	メモリパッチ機能 - パッチアドレス 2 上位	PFAH2		RW
0003C1H	メモリパッチ機能 - パッチアドレス 3 下位	PFAL3		RW
0003C2H	メモリパッチ機能 - パッチアドレス 3 中位	PFAM3		RW
0003C3H	メモリパッチ機能 - パッチアドレス 3 上位	PFAH3		RW
0003C4H	メモリパッチ機能 - パッチアドレス 4 下位	PFAL4		RW
0003C5H	メモリパッチ機能 - パッチアドレス 4 中位	PFAM4		RW
0003C6H	メモリパッチ機能 - パッチアドレス 4 上位	PFAH4		RW
0003C7H	メモリパッチ機能 - パッチアドレス 5 下位	PFAL5		RW
0003C8H	メモリパッチ機能 - パッチアドレス 5 中位	PFAM5		RW
0003C9H	メモリパッチ機能 - パッチアドレス 5 上位	PFAH5		RW
0003CAH	メモリパッチ機能 - パッチアドレス 6 下位	PFAL6		RW
0003CBH	メモリパッチ機能 - パッチアドレス 6 中位	PFAM6		RW
0003CCH	メモリパッチ機能 - パッチアドレス 6 上位	PFAH6		RW
0003CDH	メモリパッチ機能 - パッチアドレス 7 下位	PFAL7		RW
0003CEH	メモリパッチ機能 - パッチアドレス 7 中位	PFAM7		RW
0003CFH	メモリパッチ機能 - パッチアドレス 7 上位	PFAH7		RW
0003D0H	メモリパッチ機能 - パッチデータ 0 下位	PFDL0	PFD0	RW
0003D1H	メモリパッチ機能 - パッチデータ 0 上位	PFDH0		RW
0003D2H	メモリパッチ機能 - パッチデータ 1 下位	PFDL1	PFD1	RW
0003D3H	メモリパッチ機能 - パッチデータ 1 上位	PFDH1		RW

Table 37-1. MB96V300BのI/Oマップ

アドレス	レジスタ	8ビット アクセス時の レジスタ略称名	16ビット アクセス時の レジスタ略称名	属性
0003D4H	メモリパッチ機能 - パッチデータ 2 下位	PFDL2	PFD2	RW
0003D5H	メモリパッチ機能 - パッチデータ 2 上位	PFDH2		RW
0003D6H	メモリパッチ機能 - パッチデータ 3 下位	PFDL3	PFD3	RW
0003D7H	メモリパッチ機能 - パッチデータ 3 上位	PFDH3		RW
0003D8H	メモリパッチ機能 - パッチデータ 4 下位	PFDL4	PFD4	RW
0003D9H	メモリパッチ機能 - パッチデータ 4 上位	PFDH4		RW
0003DAH	メモリパッチ機能 - パッチデータ 5 下位	PFDL5	PFD5	RW
0003DBH	メモリパッチ機能 - パッチデータ 5 上位	PFDH5		RW
0003DCH	メモリパッチ機能 - パッチデータ 6 下位	PFDL6	PFD6	RW
0003DDH	メモリパッチ機能 - パッチデータ 6 上位	PFDH6		RW
0003DEH	メモリパッチ機能 - パッチデータ 7 下位	PFDL7	PFD7	RW
0003DFH	メモリパッチ機能 - パッチデータ 7 上位	PFDH7		RW
0003E0H-0003F0H	予約			-
0003F1H	フラッシュメモリ制御ステータスレジスタ A	MCSRA		RW
0003F2H	フラッシュメモリタイミング構成レジスタ A 下位	MTCRAL	MTCRA	RW
0003F3H-0003F5H	予約			-
0003F6H	フラッシュメモリタイミング構成レジスタ B 下位	MTCRBL	MTCRB	RW
0003F7H-0003FFH	予約			-
000400H	スタンバイモード制御レジスタ	SMCR		RW
000401H	クロック選択レジスタ	CKSR		RW
000402H	クロック安定化選択レジスタ	CKSSR		RW
000403H	クロックモニタレジスタ	CKMR		R
000404H	クロック周波数制御レジスタ下位	CKFCRL	CKFCR	RW
000405H	クロック周波数制御レジスタ上位	CKFCRH		RW
000406H	PLL制御レジスタ下位	PLLCRL	PLLCR	RW
000407H	PLL制御レジスタ上位	PLLCRH		RW
000408H	RCクロックタイマ制御レジスタ	RCTCR		RW
000409H	メインクロックタイマ制御レジスタ	MCTCR		RW
00040AH	サブクロックタイマ制御レジスタ	SCTCR		RW
00040BH	クリア機能のあるリセット要因およびクロックステータスレジスタ	RCCSRC		R
00040CH	リセット構成レジスタ	RCR		RW
00040DH	リセット要因およびクロックステータスレジスタ	RCCSR		R
00040EH	ウォッチドッグタイマ構成レジスタ	WDTC		RW
00040FH	ウォッチドッグタイマクリアパターンレジスタ	WDTCP		W
000410H-000414H	予約			-
000415H	クロック出力起動レジスタ	COAR		RW
000416H	クロック出力構成レジスタ 0	COCR0		RW
000417H	クロック出力構成レジスタ 1	COCR1		RW
000418H	クロックモジュレータ制御レジスタ	CMCR		RW
000419H	予約			-

Table 37-1. MB96V300BのI/Oマップ

アドレス	レジスタ	8ビット アクセス時の レジスタ略称名	16ビット アクセス時の レジスタ略称名	属性
00041AH	クロックモジュレータパラメータレジスタ下位	CMPRL	CMPR	RW
00041BH	クロックモジュレータパラメータレジスタ上位	CMPRH		RW
00041CH-00042BH	予約			-
00042CH	電圧レギュレータ制御レジスタ	VRCR		RW
00042DH	クロック入力 / LVD制御レジスタ	CILCR		RW
00042EH-00042FH	予約			-
000430H	I/Oポート P00 - データ方向レジスタ	DDR00		RW
000431H	I/Oポート P01 - データ方向レジスタ	DDR01		RW
000432H	I/Oポート P02 - データ方向レジスタ	DDR02		RW
000433H	I/Oポート P03 - データ方向レジスタ	DDR03		RW
000434H	I/Oポート P04 - データ方向レジスタ	DDR04		RW
000435H	I/Oポート P05 - データ方向レジスタ	DDR05		RW
000436H	I/Oポート P06 - データ方向レジスタ	DDR06		RW
000437H	I/Oポート P07 - データ方向レジスタ	DDR07		RW
000438H	I/Oポート P08 - データ方向レジスタ	DDR08		RW
000439H	I/Oポート P09 - データ方向レジスタ	DDR09		RW
00043AH	I/Oポート P10 - データ方向レジスタ	DDR10		RW
00043BH	I/Oポート P11 - データ方向レジスタ	DDR11		RW
00043CH	I/Oポート P12 - データ方向レジスタ	DDR12		RW
00043DH	I/Oポート P13 - データ方向レジスタ	DDR13		RW
00043EH	I/Oポート P14 - データ方向レジスタ	DDR14		RW
00043FH	I/Oポート P15 - データ方向レジスタ	DDR15		RW
000440H	I/Oポート P16 - データ方向レジスタ	DDR16		RW
000441H	I/Oポート P17 - データ方向レジスタ	DDR17		RW
000442H-000443H	予約			-
000444H	I/Oポート P00 - ポート入力許可レジスタ	PIER00		RW
000445H	I/Oポート P01 - ポート入力許可レジスタ	PIER01		RW
000446H	I/Oポート P02 - ポート入力許可レジスタ	PIER02		RW
000447H	I/Oポート P03 - ポート入力許可レジスタ	PIER03		RW
000448H	I/Oポート P04 - ポート入力許可レジスタ	PIER04		RW
000449H	I/Oポート P05 - ポート入力許可レジスタ	PIER05		RW
00044AH	I/Oポート P06 - ポート入力許可レジスタ	PIER06		RW
00044BH	I/Oポート P07 - ポート入力許可レジスタ	PIER07		RW
00044CH	I/Oポート P08 - ポート入力許可レジスタ	PIER08		RW
00044DH	I/Oポート P09 - ポート入力許可レジスタ	PIER09		RW
00044EH	I/Oポート P10 - ポート入力許可レジスタ	PIER10		RW
00044FH	I/Oポート P11 - ポート入力許可レジスタ	PIER11		RW
000450H	I/Oポート P12 - ポート入力許可レジスタ	PIER12		RW
000451H	I/Oポート P13 - ポート入力許可レジスタ	PIER13		RW
000452H	I/Oポート P14 - ポート入力許可レジスタ	PIER14		RW

Table 37-1. MB96V300BのI/Oマップ

アドレス	レジスタ	8ビット アクセス時の レジスタ略称名	16ビット アクセス時の レジスタ略称名	属性
000453H	I/Oポート P15 - ポート入力許可レジスタ	PIER15		RW
000454H	I/Oポート P16 - ポート入力許可レジスタ	PIER16		RW
000455H	I/Oポート P17 - ポート入力許可レジスタ	PIER17		RW
000456H-000457H	予約			-
000458H	I/Oポート P00 - ポート入力レベルレジスタ	PILR00		RW
000459H	I/Oポート P01 - ポート入力レベルレジスタ	PILR01		RW
00045AH	I/Oポート P02 - ポート入力レベルレジスタ	PILR02		RW
00045BH	I/Oポート P03 - ポート入力レベルレジスタ	PILR03		RW
00045CH	I/Oポート P04 - ポート入力レベルレジスタ	PILR04		RW
00045DH	I/Oポート P05 - ポート入力レベルレジスタ	PILR05		RW
00045EH	I/Oポート P06 - ポート入力レベルレジスタ	PILR06		RW
00045FH	I/Oポート P07 - ポート入力レベルレジスタ	PILR07		RW
000460H	I/Oポート P08 - ポート入力レベルレジスタ	PILR08		RW
000461H	I/Oポート P09 - ポート入力レベルレジスタ	PILR09		RW
000462H	I/Oポート P10 - ポート入力レベルレジスタ	PILR10		RW
000463H	I/Oポート P11 - ポート入力レベルレジスタ	PILR11		RW
000464H	I/Oポート P12 - ポート入力レベルレジスタ	PILR12		RW
000465H	I/Oポート P13 - ポート入力レベルレジスタ	PILR13		RW
000466H	I/Oポート P14 - ポート入力レベルレジスタ	PILR14		RW
000467H	I/Oポート P15 - ポート入力レベルレジスタ	PILR15		RW
000468H	I/Oポート P16 - ポート入力レベルレジスタ	PILR16		RW
000469H	I/Oポート P17 - ポート入力レベルレジスタ	PILR17		RW
00046AH-00046BH	予約			-
00046CH	I/Oポート P00 - 拡張ポート入力レベルレジスタ	EPILR00		RW
00046DH	I/Oポート P01 - 拡張ポート入力レベルレジスタ	EPILR01		RW
00046EH	I/Oポート P02 - 拡張ポート入力レベルレジスタ	EPILR02		RW
00046FH	I/Oポート P03 - 拡張ポート入力レベルレジスタ	EPILR03		RW
000470H	I/Oポート P04 - 拡張ポート入力レベルレジスタ	EPILR04		RW
000471H	I/Oポート P05 - 拡張ポート入力レベルレジスタ	EPILR05		RW
000472H	I/Oポート P06 - 拡張ポート入力レベルレジスタ	EPILR06		RW
000473H	I/Oポート P07 - 拡張ポート入力レベルレジスタ	EPILR07		RW
000474H	I/Oポート P08 - 拡張ポート入力レベルレジスタ	EPILR08		RW
000475H	I/Oポート P09 - 拡張ポート入力レベルレジスタ	EPILR09		RW
000476H	I/Oポート P10 - 拡張ポート入力レベルレジスタ	EPILR10		RW
000477H	I/Oポート P11 - 拡張ポート入力レベルレジスタ	EPILR11		RW
000478H	I/Oポート P12 - 拡張ポート入力レベルレジスタ	EPILR12		RW
000479H	I/Oポート P13 - 拡張ポート入力レベルレジスタ	EPILR13		RW
00047AH	I/Oポート P14 - 拡張ポート入力レベルレジスタ	EPILR14		RW
00047BH	I/Oポート P15 - 拡張ポート入力レベルレジスタ	EPILR15		RW
00047CH	I/Oポート P16 - 拡張ポート入力レベルレジスタ	EPILR16		RW
00047DH	I/Oポート P17 - 拡張ポート入力レベルレジスタ	EPILR17		RW

Table 37-1. MB96V300BのI/Oマップ

アドレス	レジスタ	8ビット アクセス時の レジスタ略称名	16ビット アクセス時の レジスタ略称名	属性
00047EH-00047FH	予約			-
000480H	I/Oポート P00 - ポート出力ドライブレジスタ	PODR00		RW
000481H	I/Oポート P01 - ポート出力ドライブレジスタ	PODR01		RW
000482H	I/Oポート P02 - ポート出力ドライブレジスタ	PODR02		RW
000483H	I/Oポート P03 - ポート出力ドライブレジスタ	PODR03		RW
000484H	I/Oポート P04 - ポート出力ドライブレジスタ	PODR04		RW
000485H	I/Oポート P05 - ポート出力ドライブレジスタ	PODR05		RW
000486H	I/Oポート P06 - ポート出力ドライブレジスタ	PODR06		RW
000487H	I/Oポート P07 - ポート出力ドライブレジスタ	PODR07		RW
000488H	I/Oポート P08 - ポート出力ドライブレジスタ	PODR08		RW
000489H	I/Oポート P09 - ポート出力ドライブレジスタ	PODR09		RW
00048AH	I/Oポート P10 - ポート出力ドライブレジスタ	PODR10		RW
00048BH	I/Oポート P11 - ポート出力ドライブレジスタ	PODR11		RW
00048CH	I/Oポート P12 - ポート出力ドライブレジスタ	PODR12		RW
00048DH	I/Oポート P13 - ポート出力ドライブレジスタ	PODR13		RW
00048EH	I/Oポート P14 - ポート出力ドライブレジスタ	PODR14		RW
00048FH	I/Oポート P15 - ポート出力ドライブレジスタ	PODR15		RW
000490H	I/Oポート P16 - ポート出力ドライブレジスタ	PODR16		RW
000491H	I/Oポート P17 - ポート出力ドライブレジスタ	PODR17		RW
000492H-00049BH	予約			-
00049CH	I/Oポート P08 - ポート上位ドライブレジスタ	PHDR08		RW
00049DH	I/Oポート P09 - ポート上位ドライブレジスタ	PHDR09		RW
00049EH	I/Oポート P10 - ポート上位ドライブレジスタ	PHDR10		RW
00049FH-0004A7H	予約			-
0004A8H	I/Oポート P00 - プルアップ抵抗制御レジスタ	PUCR00		RW
0004A9H	I/Oポート P01 - プルアップ抵抗制御レジスタ	PUCR01		RW
0004AAH	I/Oポート P02 - プルアップ抵抗制御レジスタ	PUCR02		RW
0004ABH	I/Oポート P03 - プルアップ抵抗制御レジスタ	PUCR03		RW
0004ACH	I/Oポート P04 - プルアップ抵抗制御レジスタ	PUCR04		RW
0004ADH	I/Oポート P05 - プルアップ抵抗制御レジスタ	PUCR05		RW
0004AEH	I/Oポート P06 - プルアップ抵抗制御レジスタ	PUCR06		RW
0004AFH	I/Oポート P07 - プルアップ抵抗制御レジスタ	PUCR07		RW
0004B0H	I/Oポート P08 - プルアップ抵抗制御レジスタ	PUCR08		RW
0004B1H	I/Oポート P09 - プルアップ抵抗制御レジスタ	PUCR09		RW
0004B2H	I/Oポート P10 - プルアップ抵抗制御レジスタ	PUCR10		RW
0004B3H	I/Oポート P11 - プルアップ抵抗制御レジスタ	PUCR11		RW
0004B4H	I/Oポート P12 - プルアップ抵抗制御レジスタ	PUCR12		RW
0004B5H	I/Oポート P13 - プルアップ抵抗制御レジスタ	PUCR13		RW
0004B6H	I/Oポート P14 - プルアップ抵抗制御レジスタ	PUCR14		RW
0004B7H	I/Oポート P15 - プルアップ抵抗制御レジスタ	PUCR15		RW

Table 37-1. MB96V300BのI/Oマップ

アドレス	レジスタ	8ビット アクセス時の レジスタ略称名	16ビット アクセス時の レジスタ略称名	属性
0004B8H	I/Oポート P16 - プルアップ抵抗制御レジスタ	PUCR16		RW
0004B9H	I/Oポート P17 - プルアップ抵抗制御レジスタ	PUCR17		RW
0004BAH-0004BBH	予約			-
0004BCH	I/Oポート P00 - 外部端子ステータスレジスタ	EPSR00		R
0004BDH	I/Oポート P01 - 外部端子ステータスレジスタ	EPSR01		R
0004BEH	I/Oポート P02 - 外部端子ステータスレジスタ	EPSR02		R
0004BFH	I/Oポート P03 - 外部端子ステータスレジスタ	EPSR03		R
0004C0H	I/Oポート P04 - 外部端子ステータスレジスタ	EPSR04		R
0004C1H	I/Oポート P05 - 外部端子ステータスレジスタ	EPSR05		R
0004C2H	I/Oポート P06 - 外部端子ステータスレジスタ	EPSR06		R
0004C3H	I/Oポート P07 - 外部端子ステータスレジスタ	EPSR07		R
0004C4H	I/Oポート P08 - 外部端子ステータスレジスタ	EPSR08		R
0004C5H	I/Oポート P09 - 外部端子ステータスレジスタ	EPSR09		R
0004C6H	I/Oポート P10 - 外部端子ステータスレジスタ	EPSR10		R
0004C7H	I/Oポート P11 - 外部端子ステータスレジスタ	EPSR11		R
0004C8H	I/Oポート P12 - 外部端子ステータスレジスタ	EPSR12		R
0004C9H	I/Oポート P13 - 外部端子ステータスレジスタ	EPSR13		R
0004CAH	I/Oポート P14 - 外部端子ステータスレジスタ	EPSR14		R
0004CBH	I/Oポート P15 - 外部端子ステータスレジスタ	EPSR15		R
0004CCH	I/Oポート P16 - 外部端子ステータスレジスタ	EPSR16		R
0004CDH	I/Oポート P17 - 外部端子ステータスレジスタ	EPSR17		R
0004CEH-0004CFH	予約			-
0004D0H	ADC アナログ入力許可レジスタ 0	ADER0		RW
0004D1H	ADC アナログ入力許可レジスタ 1	ADER1		RW
0004D2H	ADC アナログ入力許可レジスタ 2	ADER2		RW
0004D3H	ADC アナログ入力許可レジスタ 3	ADER3		RW
0004D4H	ADC アナログ入力許可レジスタ 4	ADER4		RW
0004D5H	予約			-
0004D6H	周辺リソース再配置レジスタ 0	PRRR0		RW
0004D7H	周辺リソース再配置レジスタ 1	PRRR1		RW
0004D8H	周辺リソース再配置レジスタ 2	PRRR2		RW
0004D9H	周辺リソース再配置レジスタ 3	PRRR3		RW
0004DAH	周辺リソース再配置レジスタ 4	PRRR4		RW
0004DBH	周辺リソース再配置レジスタ 5	PRRR5		RW
0004DCH	周辺リソース再配置レジスタ 6	PRRR6		RW
0004DDH	周辺リソース再配置レジスタ 7	PRRR7		RW
0004DEH	周辺リソース再配置レジスタ 8	PRRR8		RW
0004DFH	周辺リソース再配置レジスタ 9	PRRR9		RW
0004E0H	RTC - サブ秒レジスタ 下位	WTBRL0	WTBR0	RW
0004E1H	RTC - サブ秒レジスタ 中位	WTBRH0		RW
0004E2H	RTC - サブ秒レジスタ 上位	WTBR1		RW

Table 37-1. MB96V300BのI/Oマップ

アドレス	レジスタ	8ビット アクセス時の レジスタ略称名	16ビット アクセス時の レジスタ略称名	属性
0004E3H	RTC - 秒レジスタ	WTSR		RW
0004E4H	RTC - 分	WTMR		RW
0004E5H	RTC - 時	WTHR		RW
0004E6H	RTC - タイマ制御拡張レジスタ	WTCER		RW
0004E7H	RTC - クロック選択レジスタ	WTCKSR		RW
0004E8H	RTC - タイマ制御レジスタ下位	WTCRL	WTCR	RW
0004E9H	RTC - タイマ制御レジスタ上位	WTCRH		RW
0004EAH	CAL - キャリブレーションユニット制御レジスタ	CUCR		RW
0004EBH	予約			-
0004ECH	CAL - サブ /RC - クロックタイマデータレジスタ下位	CUTDL	CUTD	RW
0004EDH	CAL - サブ /RC - クロックタイマデータレジスタ上位	CUTDH		RW
0004EEH	CAL - メインクロックタイマレジスタ 2 下位	CUTR2L	CUTR2	R
0004EFH	CAL - メインクロックタイマレジスタ 2 上位	CUTR2H		R
0004F0H	CAL - メインクロックタイマレジスタ 1 下位	CUTR1L	CUTR1	R
0004F1H	CAL - メインクロックタイマレジスタ 1 上位	CUTR1H		R
0004F2H	RLT4 - タイマ制御ステータスレジスタ 下位	TMCSRL4	TMCSR4	RW
0004F3H	RLT4 - タイマ制御ステータスレジスタ 上位	TMCSRH4		RW
0004F4H	RLT4 - リロードレジスタ - 書込み用		TMRLR4	W
0004F4H	RLT4 - リロードレジスタ - 読出し用		TMR4	R
0004F5H	RLT4 - リロードレジスタ - 書込み用			W
0004F5H	RLT4 - リロードレジスタ - 読出し用			R
0004F6H	RLT5 - タイマ制御ステータスレジスタ 下位	TMCSRL5	TMCSR5	RW
0004F7H	RLT5 - タイマ制御ステータスレジスタ 上位	TMCSRH5		RW
0004F8H	RLT5 - リロードレジスタ - 書込み用		TMRLR5	W
0004F8H	RLT5 - リロードレジスタ - 読出し用		TMR5	R
0004F9H	RLT5 - リロードレジスタ - 書込み用			W
0004F9H	RLT5 - リロードレジスタ - 読出し用			R
0004FAH	RLT - タイマ入力選択 (カスケード用)	TMISR		RW
0004FBH-0004FFH	予約			-
000500H	FRT2 - フリーランタイマのデータレジスタ		TCDT2	RW
000501H	FRT2 - フリーランタイマのデータレジスタ			RW
000502H	FRT2 - フリーランタイマの制御ステータスレジスタ下位	TCCSL2	TCCS2	RW
000503H	FRT2 - フリーランタイマの制御ステータスレジスタ上位	TCCSH2		RW
000504H	FRT3 - フリーランタイマのデータレジスタ		TCDT3	RW
000505H	FRT3 - フリーランタイマのデータレジスタ			RW
000506H	FRT3 - フリーランタイマの制御ステータスレジスタ下位	TCCSL3	TCCS3	RW
000507H	FRT3 - フリーランタイマの制御ステータスレジスタ上位	TCCSH3		RW
000508H	OCU8 - 出力比較制御ステータス	OCS8		RW
000509H	OCU9 - 出力比較制御ステータス	OCS9		RW
00050AH	OCU8 - 比較レジスタ		OCCP8	RW
00050BH	OCU8 - 比較レジスタ			RW

Table 37-1. MB96V300BのI/Oマップ

アドレス	レジスタ	8ビット アクセス時の レジスタ略称名	16ビット アクセス時の レジスタ略称名	属性
00050CH	OCU9 - 比較レジスタ		OCCP9	RW
00050DH	OCU9 - 比較レジスタ			RW
00050EH	OCU10 - 出力比較制御ステータス	OCS10		RW
00050FH	OCU11 - 出力比較制御ステータス	OCS11		RW
000510H	OCU10 - 比較レジスタ		OCCP10	RW
000511H	OCU10 - 比較レジスタ			RW
000512H	OCU11 - 比較レジスタ		OCCP11	RW
000513H	OCU11 - 比較レジスタ			RW
000514H	ICU8/ICU9 - 制御ステータスレジスタ	ICS89		RW
000515H	ICU8/ICU9 - エッジレジスタ	ICE89		RW
000516H	ICU8 - キャプチャレジスタ下位	IPCPL8	IPCP8	R
000517H	ICU8 - キャプチャレジスタ上位	IPCPH8		R
000518H	ICU9 - キャプチャレジスタ下位	IPCPL9	IPCP9	R
000519H	ICU9 - キャプチャレジスタ上位	IPCPH9		R
00051AH	ICU10/ICU11 - 制御ステータスレジスタ	ICS1011		RW
00051BH	ICU10/ICU11 - エッジレジスタ	ICE1011		RW
00051CH	ICU10 - キャプチャレジスタ下位	IPCPL10	IPCP10	R
00051DH	ICU10 - キャプチャレジスタ上位	IPCPH10		R
00051EH	ICU11 - キャプチャレジスタ下位	IPCPL11	IPCP11	R
00051FH	ICU11 - キャプチャレジスタ上位	IPCPH11		R
000520H	USART4 - シリアルモードレジスタ	SMR4		RW
000521H	USART4 - シリアル制御レジスタ	SCR4		RW
000522H	USART4 - TXレジスタ	TDR4		W
000522H	USART4 - RXレジスタ	RDR4		R
000523H	USART4 - シリアルステータス	SSR4		RW
000524H	USART4 - 制御/比較レジスタ	ECCR4		RW
000525H	USART4 - 拡張ステータスレジスタ	ESCR4		RW
000526H	USART4 - ボーレートジェネレータレジスタ下位	BGRL4	BGR4	RW
000527H	USART4 - ボーレートジェネレータレジスタ上位	BGRH4		RW
000528H	USART4 - 拡張シリアル割込みレジスタ	ESIR4		RW
000529H	予約			-
00052AH	USART5 - シリアルモードレジスタ	SMR5		RW
00052BH	USART5 - シリアル制御レジスタ	SCR5		RW
00052CH	USART5 - RXレジスタ	TDR5		W
00052CH	USART5 - TXレジスタ	RDR5		R
00052DH	USART5 - シリアルステータス	SSR5		RW
00052EH	USART5 - 制御/比較レジスタ	ECCR5		RW
00052FH	USART5 - 拡張ステータスレジスタ	ESCR5		RW
000530H	USART5 - ボーレートジェネレータレジスタ下位	BGRL5	BGR5	RW
000531H	USART5 - ボーレートジェネレータレジスタ上位	BGRH5		RW
000532H	USART5 - 拡張シリアル割込みレジスタ	ESIR5		RW
000533H	予約			-

Table 37-1. MB96V300BのI/Oマップ

アドレス	レジスタ	8ビット アクセス時の レジスタ略称名	16ビット アクセス時の レジスタ略称名	属性
000534H	USART6 - シリアルモードレジスタ	SMR6		RW
000535H	USART6 - シリアル制御レジスタ	SCR6		RW
000536H	USART6 - シリアルTXレジスタ	TDR6		W
000536H	USART6 - シリアルRXレジスタ	RDR6		R
000537H	USART6 - シリアルステータスレジスタ	SSR6		RW
000538H	USART6 - 拡張制御/比較レジスタ	ECCR6		RW
000539H	USART6 - 拡張ステータス比較レジスタ	ESCR6		RW
00053AH	USART6 - ボーレートジェネレータレジスタ下位	BGRL6	BGR6	RW
00053BH	USART6 - ボーレートジェネレータレジスタ上位	BGRH6		RW
00053CH	USART6 - 拡張シリアル割込みレジスタ	ESIR6		RW
00053DH	予約			-
00053EH	USART7 - シリアルモードレジスタ	SMR7		RW
00053FH	USART7 - シリアル制御レジスタ	SCR7		RW
000540H	USART7 - シリアルTXレジスタ	TDR7		W
000540H	USART7 - シリアルRXレジスタ	RDR7		R
000541H	USART7 - シリアルステータスレジスタ	SSR7		RW
000542H	USART7 - 拡張制御/比較レジスタ	ECCR7		RW
000543H	USART7 - 拡張ステータス比較レジスタ	ESCR7		RW
000544H	USART7 - ボーレートジェネレータレジスタ下位	BGRL7	BGR7	RW
000545H	USART7 - ボーレートジェネレータレジスタ上位	BGRH7		RW
000546H	USART7 - 拡張シリアル割込みレジスタ	ESIR7		RW
000547H	予約			-
000548H	USART8 - シリアルモードレジスタ	SMR8		RW
000549H	USART8 - シリアル制御レジスタ	SCR8		RW
00054AH	USART8 - シリアルTXレジスタ	TDR8		W
00054AH	USART8 - シリアルRXレジスタ	RDR8		R
00054BH	USART8 - シリアルステータスレジスタ	SSR8		RW
00054CH	USART8 - 拡張制御/比較レジスタ	ECCR8		RW
00054DH	USART8 - 拡張ステータス比較レジスタ	ESCR8		RW
00054EH	USART8 - ボーレートジェネレータレジスタ下位	BGRL8	BGR8	RW
00054FH	USART8 - ボーレートジェネレータレジスタ上位	BGRH8		RW
000550H	USART8 - 拡張シリアル割込みレジスタ	ESIR8		RW
000551H	予約			-
000552H	USART9 - シリアルモードレジスタ	SMR9		RW
000553H	USART9 - シリアル制御レジスタ	SCR9		RW
000554H	USART9 - シリアルTXレジスタ	TDR9		W
000554H	USART9 - シリアルRXレジスタ	RDR9		R
000555H	USART9 - シリアルステータスレジスタ	SSR9		RW
000556H	USART9 - 拡張制御/比較レジスタ	ECCR9		RW
000557H	USART9 - 拡張ステータス比較レジスタ	ESCR9		RW
000558H	USART9 - ボーレートジェネレータレジスタ下位	BGRL9	BGR9	RW
000559H	USART9 - ボーレートジェネレータレジスタ上位	BGRH9		RW

Table 37-1. MB96V300BのI/Oマップ

アドレス	レジスタ	8ビット アクセス時の レジスタ略称名	16ビット アクセス時の レジスタ略称名	属性
00055AH	USART9 - 拡張シリアル割込みレジスタ	ESIR9		RW
00055BH-00055FH	予約			-
000560H	アラームコンパレータ0 - 制御ステータスレジスタ	ACSR0		RW
000561H	アラームコンパレータ0 - 拡張制御ステータスレジスタ	AECSR0		RW
000562H	アラームコンパレータ1 - 制御ステータスレジスタ	ACSR1		RW
000563H	アラームコンパレータ1 - 拡張制御ステータスレジスタ	AECSR1		RW
000564H	PPG6 - タイマレジスタ		PTMR6	R
000565H	PPG6 - タイマレジスタ			R
000566H	PPG6 - 周期設定レジスタ		PCSR6	W
000567H	PPG6 - 周期設定レジスタ			W
000568H	PPG6 - デューティ周期レジスタ		PDUT6	W
000569H	PPG6 - デューティ周期レジスタ			W
00056AH	PPG6 - 制御ステータスレジスタ下位	PCNL6	PCN6	RW
00056BH	PPG6 - 制御ステータスレジスタ上位	PCNH6		RW
00056CH	PPG7 - タイマレジスタ		PTMR7	R
00056DH	PPG7 - タイマレジスタ			R
00056EH	PPG7 - 周期設定レジスタ		PCSR7	W
00056FH	PPG7 - 周期設定レジスタ			W
000570H	PPG7 - デューティ周期レジスタ		PDUT7	W
000571H	PPG7 - デューティ周期レジスタ			W
000572H	PPG7 - 制御ステータスレジスタ下位	PCNL7	PCN7	RW
000573H	PPG7 - 制御ステータスレジスタ上位	PCNH7		RW
000574H	PPG11-PPG8 - 汎用制御レジスタ 1 下位	GCN1L2	GCN12	RW
000575H	PPG11-PPG8 - 汎用制御レジスタ 1 上位	GCN1H2		RW
000576H	PPG11-PPG8 - 汎用制御レジスタ 2 下位	GCN2L2	GCN22	RW
000577H	PPG11-PPG8 - 汎用制御レジスタ 2 上位	GCN2H2		RW
000578H	PPG8 - タイマレジスタ		PTMR8	R
000579H	PPG8 - タイマレジスタ			R
00057AH	PPG8 - 周期設定レジスタ		PCSR8	W
00057BH	PPG8 - 周期設定レジスタ			W
00057CH	PPG8 - デューティ周期レジスタ		PDUT8	W
00057DH	PPG8 - デューティ周期レジスタ			W
00057EH	PPG8 - 制御ステータスレジスタ下位	PCNL8	PCN8	RW
00057FH	PPG8 - 制御ステータスレジスタ上位	PCNH8		RW
000580H	PPG9 - タイマレジスタ		PTMR9	R
000581H	PPG9 - タイマレジスタ			R
000582H	PPG9 - 周期設定レジスタ		PCSR9	W
000583H	PPG9 - 周期設定レジスタ			W
000584H	PPG9 - デューティ周期レジスタ		PDUT9	W
000585H	PPG9 - デューティ周期レジスタ			W
000586H	PPG9 - 制御ステータスレジスタ下位	PCNL9	PCN9	RW

Table 37-1. MB96V300BのI/Oマップ

アドレス	レジスタ	8ビット アクセス時の レジスタ略称名	16ビット アクセス時の レジスタ略称名	属性
000587H	PPG9 - 制御ステータスレジスタ上位	PCNH9		RW
000588H	PPG10 - タイマレジスタ		PTMR10	R
000589H	PPG10 - タイマレジスタ			R
00058AH	PPG10 - 周期設定レジスタ		PCSR10	W
00058BH	PPG10 - 周期設定レジスタ			W
00058CH	PPG10 - デューティ周期レジスタ		PDUT10	W
00058DH	PPG10 - デューティ周期レジスタ			W
00058EH	PPG10 - 制御ステータスレジスタ下位	PCNL10	PCN10	RW
00058FH	PPG10 - 制御ステータスレジスタ上位	PCNH10		RW
000590H	PPG11 - タイマレジスタ		PTMR11	R
000591H	PPG11 - タイマレジスタ			R
000592H	PPG11 - 周期設定レジスタ		PCSR11	W
000593H	PPG11 - 周期設定レジスタ			W
000594H	PPG11 - デューティ周期レジスタ		PDUT11	W
000595H	PPG11 - デューティ周期レジスタ			W
000596H	PPG11 - 制御ステータスレジスタ下位	PCNL11	PCN11	RW
000597H	PPG11 - 制御ステータスレジスタ上位	PCNH11		RW
000598H	PPG15-PPG12 - 汎用制御レジスタ 1 下位	GCN1L3	GCN13	RW
000599H	PPG15-PPG12 - 汎用制御レジスタ 1 上位	GCN1H3		RW
00059AH	PPG15-PPG12 - 汎用制御レジスタ 2 下位	GCN2L3	GCN23	RW
00059BH	PPG15-PPG12 - 汎用制御レジスタ 2 上位	GCN2H3		RW
00059CH	PPG12 - タイマレジスタ		PTMR12	R
00059DH	PPG12 - タイマレジスタ			R
00059EH	PPG12 - 周期設定レジスタ		PCSR12	W
00059FH	PPG12 - 周期設定レジスタ			W
0005A0H	PPG12 - デューティ周期レジスタ		PDUT12	W
0005A1H	PPG12 - デューティ周期レジスタ			W
0005A2H	PPG12 - 制御ステータスレジスタ下位	PCNL12	PCN12	RW
0005A3H	PPG12 - 制御ステータスレジスタ上位	PCNH12		RW
0005A4H	PPG13 - タイマレジスタ		PTMR13	R
0005A5H	PPG13 - タイマレジスタ			R
0005A6H	PPG13 - 周期設定レジスタ		PCSR13	W
0005A7H	PPG13 - 周期設定レジスタ			W
0005A8H	PPG13 - デューティ周期レジスタ		PDUT13	W
0005A9H	PPG13 - デューティ周期レジスタ			W
0005AAH	PPG13 - 制御ステータスレジスタ下位	PCNL13	PCN13	RW
0005ABH	PPG13 - 制御ステータスレジスタ上位	PCNH13		RW
0005ACH	PPG14 - タイマレジスタ		PTMR14	R
0005ADH	PPG14 - タイマレジスタ			R
0005AEH	PPG14 - 周期設定レジスタ		PCSR14	W
0005AFH	PPG14 - 周期設定レジスタ			W
0005B0H	PPG14 - デューティ周期レジスタ		PDUT14	W

Table 37-1. MB96V300BのI/Oマップ

アドレス	レジスタ	8ビット アクセス時の レジスタ略称名	16ビット アクセス時の レジスタ略称名	属性
0005B1H	PPG14 - デューティ周期レジスタ			W
0005B2H	PPG14 - 制御ステータスレジスタ下位	PCNL14	PCN14	RW
0005B3H	PPG14 - 制御ステータスレジスタ上位	PCNH14		RW
0005B4H	PPG15 - タイマレジスタ		PTMR15	R
0005B5H	PPG15 - タイマレジスタ			R
0005B6H	PPG15 - 周期設定レジスタ		PCSR15	W
0005B7H	PPG15 - 周期設定レジスタ			W
0005B8H	PPG15 - デューティ周期レジスタ		PDUT15	W
0005B9H	PPG15 - デューティ周期レジスタ			W
0005BAH	PPG15 - 制御ステータスレジスタ下位	PCNL15	PCN15	RW
0005BBH	PPG15 - 制御ステータスレジスタ上位	PCNH15		RW
0005BCH	PPG19-PPG16 - 汎用制御レジスタ 1 下位	GCN1L4	GCN14	RW
0005BDH	PPG19-PPG16 - 汎用制御レジスタ 1 上位	GCN1H4		RW
0005BEH	PPG19-PPG16 - 汎用制御レジスタ 2 下位	GCN2L4	GCN24	RW
0005BFH	PPG19-PPG16 - 汎用制御レジスタ 2 上位	GCN2H4		RW
0005C0H	PPG16 - タイマレジスタ		PTMR16	R
0005C1H	PPG16 - タイマレジスタ			R
0005C2H	PPG16 - 周期設定レジスタ		PCSR16	W
0005C3H	PPG16 - 周期設定レジスタ			W
0005C4H	PPG16 - デューティ周期レジスタ		PDUT16	W
0005C5H	PPG16 - デューティ周期レジスタ			W
0005C6H	PPG16 - 制御ステータスレジスタ下位	PCNL16	PCN16	RW
0005C7H	PPG16 - 制御ステータスレジスタ上位	PCNH16		RW
0005C8H	PPG17 - タイマレジスタ		PTMR17	R
0005C9H	PPG17 - タイマレジスタ			R
0005CAH	PPG17 - 周期設定レジスタ		PCSR17	W
0005CBH	PPG17 - 周期設定レジスタ			W
0005CCH	PPG17 - デューティ周期レジスタ		PDUT17	W
0005CDH	PPG17 - デューティ周期レジスタ			W
0005CEH	PPG17 - 制御ステータスレジスタ下位	PCNL17	PCN17	RW
0005CFH	PPG17 - 制御ステータスレジスタ上位	PCNH17		RW
0005D0H	PPG18 - タイマレジスタ		PTMR18	R
0005D1H	PPG18 - タイマレジスタ			R
0005D2H	PPG18 - 周期設定レジスタ		PCSR18	W
0005D3H	PPG18 - 周期設定レジスタ			W
0005D4H	PPG18 - デューティ周期レジスタ		PDUT18	W
0005D5H	PPG18 - デューティ周期レジスタ			W
0005D6H	PPG18 - 制御ステータスレジスタ下位	PCNL18	PCN18	RW
0005D7H	PPG18 - 制御ステータスレジスタ上位	PCNH18		RW
0005D8H	PPG19 - タイマレジスタ		PTMR19	R
0005D9H	PPG19 - タイマレジスタ			R
0005DAH	PPG19 - 周期設定レジスタ		PCSR19	W

Table 37-1. MB96V300BのI/Oマップ

アドレス	レジスタ	8ビット アクセス時の レジスタ略称名	16ビット アクセス時の レジスタ略称名	属性
0005DBH	PPG19 - 周期設定レジスタ			W
0005DCH	PPG19 - デューティ周期レジスタ		PDUT19	W
0005DDH	PPG19 - デューティ周期レジスタ			W
0005DEH	PPG19 - 制御ステータスレジスタ下位	PCNL19	PCN19	RW
0005DFH	PPG19 - 制御ステータスレジスタ上位	PCNH19		RW
0005E0H	SMC0 - PWM制御レジスタ	PWC0		RW
0005E1H	SMC0 - 拡張制御レジスタ (出力許可)	PWEC0		RW
0005E2H	SMC0 - PWM比較レジスタ PWM 1		PWC10	RW
0005E3H	SMC0 - PWM比較レジスタ PWM 1			RW
0005E4H	SMC0 - PWM比較レジスタ PWM 2		PWC20	RW
0005E5H	SMC0 - PWM比較レジスタ PWM 2			RW
0005E6H	SMC0 - PWM選択レジスタ	PWS10		RW
0005E7H	SMC0 - PWM選択レジスタ	PWS20		RW
0005E8H-0005E9H	予約			-
0005EAH	SMC1 - PWM制御レジスタ	PWC1		RW
0005EBH	SMC1 - 拡張制御レジスタ (出力許可)	PWEC1		RW
0005ECH	SMC1 - PWM比較レジスタ PWM 1		PWC11	RW
0005EDH	SMC1 - PWM比較レジスタ PWM 1			RW
0005EEH	SMC1 - PWM比較レジスタ PWM 2		PWC21	RW
0005EFH	SMC1 - PWM比較レジスタ PWM 2			RW
0005F0H	SMC1 - PWM選択レジスタ	PWS11		RW
0005F1H	SMC1 - PWM選択レジスタ	PWS21		RW
0005F2H-0005F3H	予約			-
0005F4H	SMC2 - PWM制御レジスタ	PWC2		RW
0005F5H	SMC2 - 拡張制御レジスタ (出力許可)	PWEC2		RW
0005F6H	SMC2 - PWM比較レジスタ PWM 1		PWC12	RW
0005F7H	SMC2 - PWM比較レジスタ PWM 1			RW
0005F8H	SMC2 - PWM比較レジスタ PWM 2		PWC22	RW
0005F9H	SMC2 - PWM比較レジスタ PWM 2			RW
0005FAH	SMC2 - PWM選択レジスタ	PWS12		RW
0005FBH	SMC2 - PWM選択レジスタ	PWS22		RW
0005FCH-0005FDH	予約			-
0005FEH	SMC3 - PWM制御レジスタ	PWC3		RW
0005FFH	SMC3 - 拡張制御レジスタ (出力許可)	PWEC3		RW
000600H	SMC3 - PWM比較レジスタ PWM 1		PWC13	RW
000601H	SMC3 - PWM比較レジスタ PWM 1			RW
000602H	SMC3 - PWM比較レジスタ PWM 2		PWC23	RW
000603H	SMC3 - PWM比較レジスタ PWM 2			RW
000604H	SMC3 - PWM選択レジスタ	PWS13		RW
000605H	SMC3 - PWM選択レジスタ	PWS23		RW

Table 37-1. MB96V300BのI/Oマップ

アドレス	レジスタ	8ビット アクセス時の レジスタ略称名	16ビット アクセス時の レジスタ略称名	属性
000606H-000607H	予約			-
000608H	SMC4 - PWM制御レジスタ	PWC4		RW
000609H	SMC4 - 拡張制御レジスタ (出力許可)	PWEC4		RW
00060AH	SMC4 - PWM比較レジスタ PWM 1		PWC14	RW
00060BH	SMC4 - PWM比較レジスタ PWM 1			RW
00060CH	SMC4 - PWM比較レジスタ PWM 2		PWC24	RW
00060DH	SMC4 - PWM比較レジスタ PWM 2			RW
00060EH	SMC4 - PWM選択レジスタ	PWS14		RW
00060FH	SMC4 - PWM選択レジスタ	PWS24		RW
000610H-000611H	予約			-
000612H	SMC5 - PWM制御レジスタ	PWC5		RW
000613H	SMC5 - 拡張制御レジスタ (出力許可)	PWEC5		RW
000614H	SMC5 - PWM比較レジスタ PWM 1		PWC15	RW
000615H	SMC5 - PWM比較レジスタ PWM 1			RW
000616H	SMC5 - PWM比較レジスタ PWM 2		PWC25	RW
000617H	SMC5 - PWM比較レジスタ PWM 2			RW
000618H	SMC5 - PWM選択レジスタ	PWS15		RW
000619H	SMC5 - PWM選択レジスタ	PWS25		RW
00061AH-00061BH	予約			-
00061CH	LCD - 出力許可レジスタ 0 (セグメント 7-0)	LCDER0		RW
00061DH	LCD - 出力許可レジスタ 1 (セグメント 15-8)	LCDER1		RW
00061EH	LCD - 出力許可レジスタ 2 (セグメント 23-16)	LCDER2		RW
00061FH	LCD - 出力許可レジスタ 3 (セグメント 31-24)	LCDER3		RW
000620H	LCD - 出力許可レジスタ 4 (セグメント 39-32)	LCDER4		RW
000621H	LCD - 出力許可レジスタ 5 (セグメント 47-40)	LCDER5		RW
000622H	LCD - 出力許可レジスタ 6 (セグメント 55-48)	LCDER6		RW
000623H	LCD - 出力許可レジスタ 7 (セグメント 63-56)	LCDER7		RW
000624H	LCD - 出力許可レジスタ 8 (セグメント 71-64)	LCDER8		RW
000625H	予約			-
000626H	LCD - 出力許可レジスタ V (Vx)	LCDVER		RW
000627H	LCD - 拡張制御レジスタ	LECR		RW
000628H	LCD - 共通端子切換えレジスタ	LCDCMR		RW
000629H	LCD - 制御レジスタ	LCR		RW
00062AH	LCD - セグメント 1, 0 のデータレジスタ	VRAM0		RW
00062BH	LCD - セグメント 3, 2 のデータレジスタ	VRAM1		RW
00062CH	LCD - セグメント 5, 4 のデータレジスタ	VRAM2		RW
00062DH	LCD - セグメント 7, 6 のデータレジスタ	VRAM3		RW
00062EH	LCD - セグメント 9, 8 のデータレジスタ	VRAM4		RW
00062FH	LCD - セグメント 11, 10 のデータレジスタ	VRAM5		RW
000630H	LCD - セグメント 13, 12 のデータレジスタ	VRAM6		RW

Table 37-1. MB96V300BのI/Oマップ

アドレス	レジスタ	8ビット アクセス時の レジスタ略称名	16ビット アクセス時の レジスタ略称名	属性
000631H	LCD - セグメント 15, 14 のデータレジスタ	VRAM7		RW
000632H	LCD - セグメント 17, 16 のデータレジスタ	VRAM8		RW
000633H	LCD - セグメント 19, 18 のデータレジスタ	VRAM9		RW
000634H	LCD - セグメント 21, 20 のデータレジスタ	VRAM10		RW
000635H	LCD - セグメント 23, 22 のデータレジスタ	VRAM11		RW
000636H	LCD - セグメント 25, 24 のデータレジスタ	VRAM12		RW
000637H	LCD - セグメント 27, 26 のデータレジスタ	VRAM13		RW
000638H	LCD - セグメント 29, 28 のデータレジスタ	VRAM14		RW
000639H	LCD - セグメント 31, 30 のデータレジスタ	VRAM15		RW
00063AH	LCD - セグメント 33, 32 のデータレジスタ	VRAM16		RW
00063BH	LCD - セグメント 35, 34 のデータレジスタ	VRAM17		RW
00063CH	LCD - セグメント 37, 36 のデータレジスタ	VRAM18		RW
00063DH	LCD - セグメント 39, 38 のデータレジスタ	VRAM19		RW
00063EH	LCD - セグメント 41, 40 のデータレジスタ	VRAM20		RW
00063FH	LCD - セグメント 43, 42 のデータレジスタ	VRAM21		RW
000640H	LCD - セグメント 45, 44 のデータレジスタ	VRAM22		RW
000641H	LCD - セグメント 47, 46 のデータレジスタ	VRAM23		RW
000642H	LCD - セグメント 49, 48 のデータレジスタ	VRAM24		RW
000643H	LCD - セグメント 51, 50 のデータレジスタ	VRAM25		RW
000644H	LCD - セグメント 53, 52 のデータレジスタ	VRAM26		RW
000645H	LCD - セグメント 55, 54 のデータレジスタ	VRAM27		RW
000646H	LCD - セグメント 57, 56 のデータレジスタ	VRAM28		RW
000647H	LCD - セグメント 59, 58 のデータレジスタ	VRAM29		RW
000648H	LCD - セグメント 61, 60 のデータレジスタ	VRAM30		RW
000649H	LCD - セグメント 63, 62 のデータレジスタ	VRAM31		RW
00064AH	LCD - セグメント 65, 64 のデータレジスタ	VRAM32		RW
00064BH	LCD - セグメント 67, 66 のデータレジスタ	VRAM33		RW
00064CH	LCD - セグメント 69, 68 のデータレジスタ	VRAM34		RW
00064DH	LCD - セグメント 71, 70 のデータレジスタ	VRAM35		RW
00064EH-00065FH	予約			-
000660H	周辺リソース再配置レジスタ 10	PRRR10		RW
000661H	周辺リソース再配置レジスタ 11	PRRR11		RW
000662H	周辺リソース再配置レジスタ 12	PRRR12		RW
000663H	周辺リソース再配置レジスタ 13	PRRR13		W
000664H-0006DFH	予約			-
0006E0H	外部バス - 領域構成レジスタ 0 下位	EACL0	EAC0	RW
0006E1H	外部バス - 領域構成レジスタ 0 上位	EACH0		RW
0006E2H	外部バス - 領域構成レジスタ 1 下位	EACL1	EAC1	RW
0006E3H	外部バス - 領域構成レジスタ 1 上位	EACH1		RW
0006E4H	外部バス - 領域構成レジスタ 2 下位	EACL2	EAC2	RW
0006E5H	外部バス - 領域構成レジスタ 2 上位	EACH2		RW

Table 37-1. MB96V300BのI/Oマップ

アドレス	レジスタ	8ビット アクセス時の レジスタ略称名	16ビット アクセス時の レジスタ略称名	属性
0006E6H	外部バス - 領域構成レジスタ 3 下位	EACL3	EAC3	RW
0006E7H	外部バス - 領域構成レジスタ 3 上位	EACH3		RW
0006E8H	外部バス - 領域構成レジスタ 4 下位	EACL4	EAC4	RW
0006E9H	外部バス - 領域構成レジスタ 4 上位	EACH4		RW
0006EAH	外部バス - 領域構成レジスタ 5 下位	EACL5	EAC5	RW
0006EBH	外部バス - 領域構成レジスタ 5 上位	EACH5		RW
0006ECH	外部バス - 領域選択レジスタ 2	EAS2		RW
0006EDH	外部バス - 領域選択レジスタ 3	EAS3		RW
0006EEH	外部バス - 領域選択レジスタ 4	EAS4		RW
0006EFH	外部バス - 領域選択レジスタ 5	EAS5		RW
0006F0H	外部バス - モードレジスタ	EBM		RW
0006F1H	外部バス - クロックおよび機能モードレジスタ	EBCF		RW
0006F2H	外部バス - アドレス出力許可レジスタ 0	EBAE0		RW
0006F3H	外部バス - アドレス出力許可レジスタ 1	EBAE1		RW
0006F4H	外部バス - アドレス出力許可レジスタ 2	EBAE2		RW
0006F5H	外部バス - 制御信号レジスタ	EBCS		RW
0006F6H-0006FFH	予約			-
000700H	CAN0 - 制御レジスタ下位	CTRLRL0	CTRLR0	RW
000701H	CAN0 - 制御レジスタ上位 (予約)	CTRLRH0		R
000702H	CAN0 - ステータスレジスタ下位	STATRL0	STATR0	RW
000703H	CAN0 - ステータスレジスタ上位 (予約)	STATRH0		R
000704H	CAN0 - 誤差カウンタ下位 (送信)	ERRCNTL0	ERRCNT0	R
000705H	CAN0 - 誤差カウンタ上位 (受信)	ERRCNTH0		R
000706H	CAN0 - ビットタイミングレジスタ下位	BTRL0	BTR0	RW
000707H	CAN0 - ビットタイミングレジスタ上位	BTRH0		RW
000708H	CAN0 - 割込みレジスタ下位	INTRL0	INTR0	R
000709H	CAN0 - 割込みレジスタ上位	INTRH0		R
00070AH	CAN0 - テストレジスタ下位	TESTRL0	TESTR0	RW
00070BH	CAN0 - テストレジスタ上位 (予約)	TESTRH0		R
00070CH	CAN0 - BRP拡張レジスタ下位	BRPERL0	BRPER0	RW
00070DH	CAN0 - BRP拡張レジスタ上位 (予約)	BRPERH0		R
00070EH-00070FH	予約			-
000710H	CAN0 - IF1コマンド要求レジスタ下位	IF1CREQL0	IF1CREQ0	RW
000711H	CAN0 - IF1コマンド要求レジスタ上位	IF1CREQH0		RW
000712H	CAN0 - IF1コマンドマスクレジスタ下位	IF1CMSKL0	IF1CMSK0	RW
000713H	CAN0 - IF1コマンドマスクレジスタ上位 (予約)	IF1CMSKH0		R
000714H	CAN0 - IF1マスク 1レジスタ下位	IF1MSK1L0	IF1MSK10	RW
000715H	CAN0 - IF1マスク 1レジスタ上位	IF1MSK1H0		RW
000716H	CAN0 - IF1マスク 2レジスタ下位	IF1MSK2L0	IF1MSK20	RW
000717H	CAN0 - IF1マスク 2レジスタ上位	IF1MSK2H0		RW
000718H	CAN0 - IF1アービトレーション 1レジスタ下位	IF1ARB1L0	IF1ARB10	RW

Table 37-1. MB96V300BのI/Oマップ

アドレス	レジスタ	8ビット アクセス時の レジスタ略称名	16ビット アクセス時の レジスタ略称名	属性
000719H	CAN0 - IF1アービトレーション 1レジスタ上位	IF1ARB1H0		RW
00071AH	CAN0 - IF1アービトレーション 2レジスタ下位	IF1ARB2L0	IF1ARB20	RW
00071BH	CAN0 - IF1アービトレーション 2レジスタ上位	IF1ARB2H0		RW
00071CH	CAN0 - IF1メッセージ制御レジスタ下位	IF1MCTRL0	IF1MCTR0	RW
00071DH	CAN0 - IF1メッセージ制御レジスタ上位	IF1MCTRH0		RW
00071EH	CAN0 - IF1データ A1下位	IF1DTA1L0	IF1DTA10	RW
00071FH	CAN0 - IF1データ A1上位	IF1DTA1H0		RW
000720H	CAN0 - IF1データ A2下位	IF1DTA2L0	IF1DTA20	RW
000721H	CAN0 - IF1データ A2上位	IF1DTA2H0		RW
000722H	CAN0 - IF1データ B1下位	IF1DTB1L0	IF1DTB10	RW
000723H	CAN0 - IF1データ B1上位	IF1DTB1H0		RW
000724H	CAN0 - IF1データ B2下位	IF1DTB2L0	IF1DTB20	RW
000725H	CAN0 - IF1データ B2上位	IF1DTB2H0		RW
000726H-00073FH	予約			-
000740H	CAN0 - IF2コマンド要求レジスタ下位	IF2CREQL0	IF2CREQ0	RW
000741H	CAN0 - IF2コマンド要求レジスタ上位	IF2CREQH0		RW
000742H	CAN0 - IF2コマンドマスクレジスタ下位	IF2CMSKL0	IF2CMSK0	RW
000743H	CAN0 - IF2コマンドマスクレジスタ上位 (予約)	IF2CMSKH0		R
000744H	CAN0 - IF2 マスク 1レジスタ下位	IF2MSK1L0	IF2MSK10	RW
000745H	CAN0 - IF2 マスク 1レジスタ上位	IF2MSK1H0		RW
000746H	CAN0 - IF2 マスク 2レジスタ下位	IF2MSK2L0	IF2MSK20	RW
000747H	CAN0 - IF2 マスク 2レジスタ上位	IF2MSK2H0		RW
000748H	CAN0 - IF2 アービトレーション 1レジスタ下位	IF2ARB1L0	IF2ARB10	RW
000749H	CAN0 - IF2 アービトレーション 1レジスタ上位	IF2ARB1H0		RW
00074AH	CAN0 - IF2 アービトレーション 2レジスタ下位	IF2ARB2L0	IF2ARB20	RW
00074BH	CAN0 - IF2 アービトレーション 2レジスタ上位	IF2ARB2H0		RW
00074CH	CAN0 - IF2 メッセージ制御レジスタ下位	IF2MCTRL0	IF2MCTR0	RW
00074DH	CAN0 - IF2 メッセージ制御レジスタ上位	IF2MCTRH0		RW
00074EH	CAN0 - IF2 データ A1下位	IF2DTA1L0	IF2DTA10	RW
00074FH	CAN0 - IF2 データ A1上位	IF2DTA1H0		RW
000750H	CAN0 - IF2 データ A2下位	IF2DTA2L0	IF2DTA20	RW
000751H	CAN0 - IF2 データ A2上位	IF2DTA2H0		RW
000752H	CAN0 - IF2 データ B1下位	IF2DTB1L0	IF2DTB10	RW
000753H	CAN0 - IF2 データ B1上位	IF2DTB1H0		RW
000754H	CAN0 - IF2 データ B2下位	IF2DTB2L0	IF2DTB20	RW
000755H	CAN0 - IF2 データ B2上位	IF2DTB2H0		RW
000756H-00077FH	予約			-
000780H	CAN0 - 送信要求 1レジスタ下位	TREQR1L0	TREQR10	R
000781H	CAN0 - 送信要求 1レジスタ上位	TREQR1H0		R
000782H	CAN0 - 送信要求 2レジスタ下位	TREQR2L0	TREQR20	R
000783H	CAN0 - 送信要求 2レジスタ上位	TREQR2H0		R

Table 37-1. MB96V300BのI/Oマップ

アドレス	レジスタ	8ビット アクセス時の レジスタ略称名	16ビット アクセス時の レジスタ略称名	属性
000784H-00078FH	予約			-
000790H	CAN0 - 新規データ 1レジスタ下位	NEWDT1L0	NEWDT10	R
000791H	CAN0 - 新規データ 1レジスタ上位	NEWDT1H0		R
000792H	CAN0 - 新規データ 2レジスタ下位	NEWDT2L0	NEWDT20	R
000793H	CAN0 - 新規データ 2レジスタ上位	NEWDT2H0		R
000794H-00079FH	予約			-
0007A0H	CAN0 - 割込み保留 1レジスタ下位	INTPND1L0	INTPND10	R
0007A1H	CAN0 - 割込み保留 1レジスタ上位	INTPND1H0		R
0007A2H	CAN0 - 割込み保留 2レジスタ下位	INTPND2L0	INTPND20	R
0007A3H	CAN0 - 割込み保留 2レジスタ上位	INTPND2H0		R
0007A4H-0007AFH	予約			-
0007B0H	CAN0 - メッセージ有効 1レジスタ下位	MSGVAL1L0	MSGVAL10	R
0007B1H	CAN0 - メッセージ有効 1レジスタ上位	MSGVAL1H0		R
0007B2H	CAN0 - メッセージ有効 2レジスタ下位	MSGVAL2L0	MSGVAL20	R
0007B3H	CAN0 - メッセージ有効 2レジスタ上位	MSGVAL2H0		R
0007B4H-0007CDH	予約			-
0007CEH	CAN0 - 出力許可レジスタ	COER0		RW
0007CFH	予約			-
0007D0H	SG0 - サウンドジェネレータ制御レジスタ下位	SGCRL0	SGCR0	RW
0007D1H	SG0 - サウンドジェネレータ制御レジスタ上位	SGCRH0		RW
0007D2H	SG0 - サウンドジェネレータ周波数レジスタ	SGFR0		RW
0007D3H	SG0 - サウンドジェネレータ振幅レジスタ	SGAR0		RW
0007D4H	SG0 - サウンドジェネレータデクリメントレジスタ	SGDR0		RW
0007D5H	SG0 - サウンドジェネレータ音質レジスタ	SGTR0		RW
0007D6H	SG1 - サウンドジェネレータ制御レジスタ下位	SGCRL1	SGCR1	RW
0007D7H	SG1 - サウンドジェネレータ制御レジスタ上位	SGCRH1		RW
0007D8H	SG1 - サウンドジェネレータ周波数レジスタ	SGFR1		RW
0007D9H	SG1 - サウンドジェネレータ振幅レジスタ	SGAR1		RW
0007DAH	SG1 - サウンドジェネレータデクリメントレジスタ	SGDR1		RW
0007DBH	SG1 - サウンドジェネレータ音質レジスタ	SGTR1		RW
0007DCH-0007FFH	予約			-
000800H	CAN1 - 制御レジスタ下位	CTRLRL1	CTRLR1	RW
000801H	CAN1 - 制御レジスタ上位 (予約)	CTRLRH1		R
000802H	CAN1 - ステータスレジスタ下位	STATRL1	STATR1	RW
000803H	CAN1 - ステータスレジスタ上位 (予約)	STATRH1		R
000804H	CAN1 - 誤差カウンタ下位 (送信)	ERRCNTL1	ERRCNT1	R
000805H	CAN1 - 誤差カウンタ上位 (受信)	ERRCNTH1		R
000806H	CAN1 - ビットタイミングレジスタ下位	BTRL1	BTR1	RW
000807H	CAN1 - ビットタイミングレジスタ上位	BTRH1		RW

Table 37-1. MB96V300BのI/Oマップ

アドレス	レジスタ	8ビット アクセス時の レジスタ略称名	16ビット アクセス時の レジスタ略称名	属性
000808H	CAN1 - 割込みレジスタ下位	INTRL1	INTR1	R
000809H	CAN1 - 割込みレジスタ上位	INTRH1		R
00080AH	CAN1 - テストレジスタ下位	TESTRL1	TESTR1	RW
00080BH	CAN1 - テストレジスタ上位 (予約)	TESTRH1		R
00080CH	CAN1 - BRP 拡張レジスタ下位	BRPERL1	BRPER1	RW
00080DH	CAN1 - BRP 拡張レジスタ上位 (予約)	BRPERH1		R
00080EH-00080FH	予約			-
000810H	CAN1 - IF1 コマンド要求レジスタ下位	IF1CREQL1	IF1CREQ1	RW
000811H	CAN1 - IF1 コマンド要求レジスタ上位	IF1CREQH1		RW
000812H	CAN1 - IF1 コマンドマスクレジスタ下位	IF1CMSKL1	IF1CMSK1	RW
000813H	CAN1 - IF1 コマンドマスクレジスタ上位 (予約)	IF1CMSKH1		R
000814H	CAN1 - IF1 マスク 1レジスタ下位	IF1MSK1L1	IF1MSK11	RW
000815H	CAN1 - IF1 マスク 1レジスタ上位	IF1MSK1H1		RW
000816H	CAN1 - IF1 マスク 2レジスタ下位	IF1MSK2L1	IF1MSK21	RW
000817H	CAN1 - IF1 マスク 2レジスタ上位	IF1MSK2H1		RW
000818H	CAN1 - IF1 アービトレーション 1レジスタ下位	IF1ARB1L1	IF1ARB11	RW
000819H	CAN1 - IF1 アービトレーション 1レジスタ上位	IF1ARB1H1		RW
00081AH	CAN1 - IF1 アービトレーション 2レジスタ下位	IF1ARB2L1	IF1ARB21	RW
00081BH	CAN1 - IF1 アービトレーション 2レジスタ上位	IF1ARB2H1		RW
00081CH	CAN1 - IF1 メッセージ制御レジスタ下位	IF1MCTRL1	IF1MCTR1	RW
00081DH	CAN1 - IF1 メッセージ制御レジスタ上位	IF1MCTRH1		RW
00081EH	CAN1 - IF1 データ A1下位	IF1DTA1L1	IF1DTA11	RW
00081FH	CAN1 - IF1 データ A1上位	IF1DTA1H1		RW
000820H	CAN1 - IF1 データ A2下位	IF1DTA2L1	IF1DTA21	RW
000821H	CAN1 - IF1 データ A2上位	IF1DTA2H1		RW
000822H	CAN1 - IF1 データ B1下位	IF1DTB1L1	IF1DTB11	RW
000823H	CAN1 - IF1 データ B1上位	IF1DTB1H1		RW
000824H	CAN1 - IF1 データ B2下位	IF1DTB2L1	IF1DTB21	RW
000825H	CAN1 - IF1 データ B2上位	IF1DTB2H1		RW
000826H-00083FH	予約			-
000840H	CAN1 - IF2 コマンド要求レジスタ下位	IF2CREQL1	IF2CREQ1	RW
000841H	CAN1 - IF2 コマンド要求レジスタ上位	IF2CREQH1		RW
000842H	CAN1 - IF2 コマンドマスクレジスタ下位	IF2CMSKL1	IF2CMSK1	RW
000843H	CAN1 - IF2 コマンドマスクレジスタ上位 (予約)	IF2CMSKH1		R
000844H	CAN1 - IF2 マスク 1レジスタ下位	IF2MSK1L1	IF2MSK11	RW
000845H	CAN1 - IF2 マスク 1レジスタ上位	IF2MSK1H1		RW
000846H	CAN1 - IF2 マスク 2レジスタ下位	IF2MSK2L1	IF2MSK21	RW
000847H	CAN1 - IF2 マスク 2レジスタ上位	IF2MSK2H1		RW
000848H	CAN1 - IF2 アービトレーション 1レジスタ下位	IF2ARB1L1	IF2ARB11	RW
000849H	CAN1 - IF2 アービトレーション 1レジスタ上位	IF2ARB1H1		RW
00084AH	CAN1 - IF2 アービトレーション 2レジスタ下位	IF2ARB2L1	IF2ARB21	RW

Table 37-1. MB96V300BのI/Oマップ

アドレス	レジスタ	8ビット アクセス時の レジスタ略称名	16ビット アクセス時の レジスタ略称名	属性
00084BH	CAN1 - IF2 アービトレーション 2レジスタ上位	IF2ARB2H1		RW
00084CH	CAN1 - IF2 メッセージ制御レジスタ下位	IF2MCTRL1	IF2MCTR1	RW
00084DH	CAN1 - IF2 メッセージ制御レジスタ上位	IF2MCTRH1		RW
00084EH	CAN1 - IF2 データ A1下位	IF2DTA1L1	IF2DTA11	RW
00084FH	CAN1 - IF2 データ A1上位	IF2DTA1H1		RW
000850H	CAN1 - IF2 データ A2下位	IF2DTA2L1	IF2DTA21	RW
000851H	CAN1 - IF2 データ A2上位	IF2DTA2H1		RW
000852H	CAN1 - IF2 データ B1下位	IF2DTB1L1	IF2DTB11	RW
000853H	CAN1 - IF2 データ B1上位	IF2DTB1H1		RW
000854H	CAN1 - IF2 データ B2下位	IF2DTB2L1	IF2DTB21	RW
000855H	CAN1 - IF2 データ B2上位	IF2DTB2H1		RW
000856H-00087FH	予約			-
000880H	CAN1 - 送信要求 1レジスタ下位	TREQR1L1	TREQR11	R
000881H	CAN1 - 送信要求 1レジスタ上位	TREQR1H1		R
000882H	CAN1 - 送信要求 2レジスタ下位	TREQR2L1	TREQR21	R
000883H	CAN1 - 送信要求 2レジスタ上位	TREQR2H1		R
000884H-00088FH	予約			-
000890H	CAN1 - 新規データ 1レジスタ下位	NEWDT1L1	NEWDT11	R
000891H	CAN1 - 新規データ 1レジスタ上位	NEWDT1H1		R
000892H	CAN1 - 新規データ 2レジスタ下位	NEWDT2L1	NEWDT21	R
000893H	CAN1 - 新規データ 2レジスタ上位	NEWDT2H1		R
000894H-00089FH	予約			-
0008A0H	CAN1 - 割込み保留 1レジスタ下位	INTPND1L1	INTPND11	R
0008A1H	CAN1 - 割込み保留 1レジスタ上位	INTPND1H1		R
0008A2H	CAN1 - 割込み保留 2レジスタ下位	INTPND2L1	INTPND21	R
0008A3H	CAN1 - 割込み保留 2レジスタ上位	INTPND2H1		R
0008A4H-0008AFH	予約			-
0008B0H	CAN1 - メッセージ有効 1レジスタ下位	MSGVAL1L1	MSGVAL11	R
0008B1H	CAN1 - メッセージ有効 1レジスタ上位	MSGVAL1H1		R
0008B2H	CAN1 - メッセージ有効 2レジスタ下位	MSGVAL2L1	MSGVAL21	R
0008B3H	CAN1 - メッセージ有効 2レジスタ上位	MSGVAL2H1		R
0008B4H-0008CDH	予約			-
0008CEH	CAN1 - 出力許可レジスタ	COER1		RW
0008CFH-0008FFH	予約			-
000900H	CAN2 - 制御レジスタ下位	CTRLRL2	CTRLR2	RW
000901H	CAN2 - 制御レジスタ上位 (予約)	CTRLRH2		R
000902H	CAN2 - ステータスレジスタ下位	STATRL2	STATR2	RW
000903H	CAN2 - ステータスレジスタ上位 (予約)	STATRH2		R
000904H	CAN2 - 誤差カウンタ下位 (送信)	ERRCNTL2	ERRCNT2	R

Table 37-1. MB96V300BのI/Oマップ

アドレス	レジスタ	8ビット アクセス時の レジスタ略称名	16ビット アクセス時の レジスタ略称名	属性
000905H	CAN2 - 誤差カウンタ上位 (受信)	ERRCNTH2		R
000906H	CAN2 - ビットタイミングレジスタ下位	BTRL2	BTR2	RW
000907H	CAN2 - ビットタイミングレジスタ上位	BTRH2		RW
000908H	CAN2 - 割込みレジスタ下位	INTRL2	INTR2	R
000909H	CAN2 - 割込みレジスタ上位	INTRH2		R
00090AH	CAN2 - テストレジスタ下位	TESTRL2	TESTR2	RW
00090BH	CAN2 - テストレジスタ上位 (予約)	TESTRH2		R
00090CH	CAN2 - BRP 拡張レジスタ下位	BRPERL2	BRPER2	RW
00090DH	CAN2 - BRP 拡張レジスタ上位 (予約)	BRPERH2		R
00090EH-00090FH	予約			-
000910H	CAN2 - IF1 コマンド要求レジスタ下位	IF1CREQL2	IF1CREQ2	RW
000911H	CAN2 - IF1 コマンド要求レジスタ上位	IF1CREQH2		RW
000912H	CAN2 - IF1 コマンドマスクレジスタ下位	IF1CMSKL2	IF1CMSK2	RW
000913H	CAN2 - IF1 コマンドマスクレジスタ上位 (予約)	IF1CMSKH2		R
000914H	CAN2 - IF1 マスク 1レジスタ下位	IF1MSK1L2	IF1MSK12	RW
000915H	CAN2 - IF1 マスク 1レジスタ上位	IF1MSK1H2		RW
000916H	CAN2 - IF1 マスク 2レジスタ下位	IF1MSK2L2	IF1MSK22	RW
000917H	CAN2 - IF1 マスク 2レジスタ上位	IF1MSK2H2		RW
000918H	CAN2 - IF1 アービトレーション 1レジスタ下位	IF1ARB1L2	IF1ARB12	RW
000919H	CAN2 - IF1 アービトレーション 1レジスタ上位	IF1ARB1H2		RW
00091AH	CAN2 - IF1 アービトレーション 2レジスタ下位	IF1ARB2L2	IF1ARB22	RW
00091BH	CAN2 - IF1 アービトレーション 2レジスタ上位	IF1ARB2H2		RW
00091CH	CAN2 - IF1 メッセージ制御レジスタ下位	IF1MCTRL2	IF1MCTR2	RW
00091DH	CAN2 - IF1 メッセージ制御レジスタ上位	IF1MCTRH2		RW
00091EH	CAN2 - IF1 データ A1下位	IF1DTA1L2	IF1DTA12	RW
00091FH	CAN2 - IF1 データ A1上位	IF1DTA1H2		RW
000920H	CAN2 - IF1 データ A2下位	IF1DTA2L2	IF1DTA22	RW
000921H	CAN2 - IF1 データ A2上位	IF1DTA2H2		RW
000922H	CAN2 - IF1 データ B1下位	IF1DTB1L2	IF1DTB12	RW
000923H	CAN2 - IF1 データ B1上位	IF1DTB1H2		RW
000924H	CAN2 - IF1 データ B2下位	IF1DTB2L2	IF1DTB22	RW
000925H	CAN2 - IF1 データ B2上位	IF1DTB2H2		RW
000926H-00093FH	予約			-
000940H	CAN2 - IF2 コマンド要求レジスタ下位	IF2CREQL2	IF2CREQ2	RW
000941H	CAN2 - IF2 コマンド要求レジスタ上位	IF2CREQH2		RW
000942H	CAN2 - IF2 コマンドマスクレジスタ下位	IF2CMSKL2	IF2CMSK2	RW
000943H	CAN2 - IF2 コマンドマスクレジスタ上位 (予約)	IF2CMSKH2		R
000944H	CAN2 - IF2 マスク 1レジスタ下位	IF2MSK1L2	IF2MSK12	RW
000945H	CAN2 - IF2 マスク 1レジスタ上位	IF2MSK1H2		RW
000946H	CAN2 - IF2 マスク 2レジスタ下位	IF2MSK2L2	IF2MSK22	RW
000947H	CAN2 - IF2 マスク 2レジスタ上位	IF2MSK2H2		RW

Table 37-1. MB96V300BのI/Oマップ

アドレス	レジスタ	8ビット アクセス時の レジスタ略称名	16ビット アクセス時の レジスタ略称名	属性
000948H	CAN2 - IF2 アービトレーション 1レジスタ下位	IF2ARB1L2	IF2ARB12	RW
000949H	CAN2 - IF2 アービトレーション 1レジスタ上位	IF2ARB1H2		RW
00094AH	CAN2 - IF2 アービトレーション 2レジスタ下位	IF2ARB2L2	IF2ARB22	RW
00094BH	CAN2 - IF2 アービトレーション 2レジスタ上位	IF2ARB2H2		RW
00094CH	CAN2 - IF2 メッセージ制御レジスタ下位	IF2MCTRL2	IF2MCTR2	RW
00094DH	CAN2 - IF2 メッセージ制御レジスタ上位	IF2MCTRH2		RW
00094EH	CAN2 - IF2 データ A1下位	IF2DTA1L2	IF2DTA12	RW
00094FH	CAN2 - IF2 データ A1上位	IF2DTA1H2		RW
000950H	CAN2 - IF2 データ A2下位	IF2DTA2L2	IF2DTA22	RW
000951H	CAN2 - IF2 データ A2上位	IF2DTA2H2		RW
000952H	CAN2 - IF2 データ B1下位	IF2DTB1L2	IF2DTB12	RW
000953H	CAN2 - IF2 データ B1上位	IF2DTB1H2		RW
000954H	CAN2 - IF2 データ B2下位	IF2DTB2L2	IF2DTB22	RW
000955H	CAN2 - IF2 データ B2上位	IF2DTB2H2		RW
000956H-00097FH	予約			-
000980H	CAN2 - 送信要求 1レジスタ下位	TREQR1L2	TREQR12	R
000981H	CAN2 - 送信要求 1レジスタ上位	TREQR1H2		R
000982H	CAN2 - 送信要求 2レジスタ下位	TREQR2L2	TREQR22	R
000983H	CAN2 - 送信要求 2レジスタ上位	TREQR2H2		R
000984H-00098FH	予約			-
000990H	CAN2 - 新規データ 1レジスタ下位	NEWDT1L2	NEWDT12	R
000991H	CAN2 - 新規データ 1レジスタ上位	NEWDT1H2		R
000992H	CAN2 - 新規データ 2レジスタ下位	NEWDT2L2	NEWDT22	R
000993H	CAN2 - 新規データ 2レジスタ上位	NEWDT2H2		R
000994H-00099FH	予約			-
0009A0H	CAN2 - 割込み保留 1レジスタ下位	INTPND1L2	INTPND12	R
0009A1H	CAN2 - 割込み保留 1レジスタ上位	INTPND1H2		R
0009A2H	CAN2 - 割込み保留 2レジスタ下位	INTPND2L2	INTPND22	R
0009A3H	CAN2 - 割込み保留 2レジスタ上位	INTPND2H2		R
0009A4H-0009AFH	予約			-
0009B0H	CAN2 - メッセージ有効 1レジスタ下位	MSGVAL1L2	MSGVAL12	R
0009B1H	CAN2 - メッセージ有効 1レジスタ上位	MSGVAL1H2		R
0009B2H	CAN2 - メッセージ有効 2レジスタ下位	MSGVAL2L2	MSGVAL22	R
0009B3H	CAN2 - メッセージ有効 2レジスタ上位	MSGVAL2H2		R
0009B4H-0009CDH	予約			-
0009CEH	CAN2 - 出力許可レジスタ	COER2		RW
0009CFH-0009FFH	予約			-
000A00H	CAN3 - 制御レジスタ下位	CTRLRL3	CTRLR3	RW
000A01H	CAN3 - 制御レジスタ上位 (予約)	CTRLRH3		R

Table 37-1. MB96V300BのI/Oマップ

アドレス	レジスタ	8ビット アクセス時の レジスタ略称名	16ビット アクセス時の レジスタ略称名	属性
000A02H	CAN3 - ステータスレジスタ下位	STATRL3	STATR3	RW
000A03H	CAN3 - ステータスレジスタ上位 (予約)	STATRH3		R
000A04H	CAN3 - 誤差カウンタ下位 (送信)	ERRCNTL3	ERRCNT3	R
000A05H	CAN3 - 誤差カウンタ上位 (受信)	ERRCNTH3		R
000A06H	CAN3 - ビットタイミングレジスタ下位	BTRL3	BTR3	RW
000A07H	CAN3 - ビットタイミングレジスタ上位	BTRH3		RW
000A08H	CAN3 - 割込みレジスタ下位	INTRL3	INTR3	R
000A09H	CAN3 - 割込みレジスタ上位	INTRH3		R
000A0AH	CAN3 - テストレジスタ下位	TESTRL3	TESTR3	RW
000A0BH	CAN3 - テストレジスタ上位 (予約)	TESTRH3		R
000A0CH	CAN3 - BRP 拡張レジスタ下位	BRPERL3	BRPER3	RW
000A0DH	CAN3 - BRP 拡張レジスタ上位 (予約)	BRPERH3		R
000A0EH-000A0FH	予約			-
000A10H	CAN3 - IF1 コマンド要求レジスタ下位	IF1CREQL3	IF1CREQ3	RW
000A11H	CAN3 - IF1 コマンド要求レジスタ上位	IF1CREQH3		RW
000A12H	CAN3 - IF1 コマンドマスクレジスタ下位	IF1CMSKL3	IF1CMSK3	RW
000A13H	CAN3 - IF1 コマンドマスクレジスタ上位 (予約)	IF1CMSKH3		R
000A14H	CAN3 - IF1 マスク 1レジスタ下位	IF1MSK1L3	IF1MSK13	RW
000A15H	CAN3 - IF1 マスク 1レジスタ上位	IF1MSK1H3		RW
000A16H	CAN3 - IF1 マスク 2レジスタ下位	IF1MSK2L3	IF1MSK23	RW
000A17H	CAN3 - IF1 マスク 2レジスタ上位	IF1MSK2H3		RW
000A18H	CAN3 - IF1 アービトレーション 1レジスタ下位	IF1ARB1L3	IF1ARB13	RW
000A19H	CAN3 - IF1 アービトレーション 1レジスタ上位	IF1ARB1H3		RW
000A1AH	CAN3 - IF1 アービトレーション 2レジスタ下位	IF1ARB2L3	IF1ARB23	RW
000A1BH	CAN3 - IF1 アービトレーション 2レジスタ上位	IF1ARB2H3		RW
000A1CH	CAN3 - IF1 メッセージ制御レジスタ下位	IF1MCTRL3	IF1MCTR3	RW
000A1DH	CAN3 - IF1 メッセージ制御レジスタ上位	IF1MCTRH3		RW
000A1EH	CAN3 - IF1 データ A1下位	IF1DTA1L3	IF1DTA13	RW
000A1FH	CAN3 - IF1 データ A1上位	IF1DTA1H3		RW
000A20H	CAN3 - IF1 データ A2下位	IF1DTA2L3	IF1DTA23	RW
000A21H	CAN3 - IF1 データ A2上位	IF1DTA2H3		RW
000A22H	CAN3 - IF1 データ B1下位	IF1DTB1L3	IF1DTB13	RW
000A23H	CAN3 - IF1 データ B1上位	IF1DTB1H3		RW
000A24H	CAN3 - IF1 データ B2下位	IF1DTB2L3	IF1DTB23	RW
000A25H	CAN3 - IF1 データ B2上位	IF1DTB2H3		RW
000A26H-000A3FH	予約			-
000A40H	CAN3 - IF2 コマンド要求レジスタ下位	IF2CREQL3	IF2CREQ3	RW
000A41H	CAN3 - IF2 コマンド要求レジスタ上位	IF2CREQH3		RW
000A42H	CAN3 - IF2 コマンドマスクレジスタ下位	IF2CMSKL3	IF2CMSK3	RW
000A43H	CAN3 - IF2 コマンドマスクレジスタ上位 (予約)	IF2CMSKH3		R
000A44H	CAN3 - IF2 マスク 1レジスタ下位	IF2MSK1L3	IF2MSK13	RW

Table 37-1. MB96V300BのI/Oマップ

アドレス	レジスタ	8ビット アクセス時の レジスタ略称名	16ビット アクセス時の レジスタ略称名	属性
000A45H	CAN3 - IF2 マスク 1レジスタ上位	IF2MSK1H3		RW
000A46H	CAN3 - IF2 マスク 2レジスタ下位	IF2MSK2L3	IF2MSK23	RW
000A47H	CAN3 - IF2 マスク 2レジスタ上位	IF2MSK2H3		RW
000A48H	CAN3 - IF2 アービトレーション 1レジスタ下位	IF2ARB1L3	IF2ARB13	RW
000A49H	CAN3 - IF2 アービトレーション 1レジスタ上位	IF2ARB1H3		RW
000A4AH	CAN3 - IF2 アービトレーション 2レジスタ下位	IF2ARB2L3	IF2ARB23	RW
000A4BH	CAN3 - IF2 アービトレーション 2レジスタ上位	IF2ARB2H3		RW
000A4CH	CAN3 - IF2 メッセージ制御レジスタ下位	IF2MCTRL3	IF2MCTR3	RW
000A4DH	CAN3 - IF2 メッセージ制御レジスタ上位	IF2MCTRH3		RW
000A4EH	CAN3 - IF2 データ A1下位	IF2DTA1L3	IF2DTA13	RW
000A4FH	CAN3 - IF2 データ A1上位	IF2DTA1H3		RW
000A50H	CAN3 - IF2 データ A2下位	IF2DTA2L3	IF2DTA23	RW
000A51H	CAN3 - IF2 データ A2上位	IF2DTA2H3		RW
000A52H	CAN3 - IF2 データ B1下位	IF2DTB1L3	IF2DTB13	RW
000A53H	CAN3 - IF2 データ B1上位	IF2DTB1H3		RW
000A54H	CAN3 - IF2 データ B2下位	IF2DTB2L3	IF2DTB23	RW
000A55H	CAN3 - IF2 データ B2上位	IF2DTB2H3		RW
000A56H-000A7FH	予約			-
000A80H	CAN3 - 送信要求 1レジスタ下位	TREQR1L3	TREQR13	R
000A81H	CAN3 - 送信要求 1レジスタ上位	TREQR1H3		R
000A82H	CAN3 - 送信要求 2レジスタ下位	TREQR2L3	TREQR23	R
000A83H	CAN3 - 送信要求 2レジスタ上位	TREQR2H3		R
000A84H-000A8FH	予約			-
000A90H	CAN3 - 新規データ 1レジスタ下位	NEWDT1L3	NEWDT13	R
000A91H	CAN3 - 新規データ 1レジスタ上位	NEWDT1H3		R
000A92H	CAN3 - 新規データ 2レジスタ下位	NEWDT2L3	NEWDT23	R
000A93H	CAN3 - 新規データ 2レジスタ上位	NEWDT2H3		R
000A94H-000A9FH	予約			-
000AA0H	CAN3 - 割込み保留 1レジスタ下位	INTPND1L3	INTPND13	R
000AA1H	CAN3 - 割込み保留 1レジスタ上位	INTPND1H3		R
000AA2H	CAN3 - 割込み保留 2レジスタ下位	INTPND2L3	INTPND23	R
000AA3H	CAN3 - 割込み保留 2レジスタ上位	INTPND2H3		R
000AA4H-000AAFH	予約			-
000AB0H	CAN3 - メッセージ有効 1レジスタ下位	MSGVAL1L3	MSGVAL13	R
000AB1H	CAN3 - メッセージ有効 1レジスタ上位	MSGVAL1H3		R
000AB2H	CAN3 - メッセージ有効 2レジスタ下位	MSGVAL2L3	MSGVAL23	R
000AB3H	CAN3 - メッセージ有効 2レジスタ上位	MSGVAL2H3		R
000AB4H-000ACDH	予約			-
000ACEH	CAN3 - 出力許可レジスタ	COER3		RW

Table 37-1. MB96V300BのI/Oマップ

アドレス	レジスタ	8ビット アクセス時の レジスタ略称名	16ビット アクセス時の レジスタ略称名	属性
000ACFH-000AFFH	予約			-
000B00H	CAN4 - 制御レジスタ下位	CTRLRL4	CTRLR4	RW
000B01H	CAN4 - 制御レジスタ上位 (予約)	CTRLRH4		R
000B02H	CAN4 - ステータスレジスタ下位	STATRL4	STATR4	RW
000B03H	CAN4 - ステータスレジスタ上位 (予約)	STATRH4		R
000B04H	CAN4 - 誤差カウンタ下位 (送信)	ERRCNTL4	ERRCNT4	R
000B05H	CAN4 - 誤差カウンタ上位 (受信)	ERRCNTH4		R
000B06H	CAN4 - ビットタイミングレジスタ下位	BTRL4	BTR4	RW
000B07H	CAN4 - ビットタイミングレジスタ上位	BTRH4		RW
000B08H	CAN4 - 割込みレジスタ下位	INTRL4	INTR4	R
000B09H	CAN4 - 割込みレジスタ上位	INTRH4		R
000B0AH	CAN4 - テストレジスタ下位	TESTRL4	TESTR4	RW
000B0BH	CAN4 - テストレジスタ上位 (予約)	TESTRH4		R
000B0CH	CAN4 - BRP 拡張レジスタ下位	BRPERL4	BRPER4	RW
000B0DH	CAN4 - BRP 拡張レジスタ上位 (予約)	BRPERH4		R
000B0EH-000B0FH	予約			-
000B10H	CAN4 - IF1 コマンド要求レジスタ下位	IF1CREQL4	IF1CREQ4	RW
000B11H	CAN4 - IF1 コマンド要求レジスタ上位	IF1CREQH4		RW
000B12H	CAN4 - IF1 コマンドマスクレジスタ下位	IF1CMSKL4	IF1CMSK4	RW
000B13H	CAN4 - IF1 コマンドマスクレジスタ上位 (予約)	IF1CMSKH4		R
000B14H	CAN4 - IF1 マスク 1レジスタ下位	IF1MSK1L4	IF1MSK14	RW
000B15H	CAN4 - IF1 マスク 1レジスタ上位	IF1MSK1H4		RW
000B16H	CAN4 - IF1 マスク 2レジスタ下位	IF1MSK2L4	IF1MSK24	RW
000B17H	CAN4 - IF1 マスク 2レジスタ上位	IF1MSK2H4		RW
000B18H	CAN4 - IF1 アービトレーション 1レジスタ下位	IF1ARB1L4	IF1ARB14	RW
000B19H	CAN4 - IF1 アービトレーション 1レジスタ上位	IF1ARB1H4		RW
000B1AH	CAN4 - IF1 アービトレーション 2レジスタ下位	IF1ARB2L4	IF1ARB24	RW
000B1BH	CAN4 - IF1 アービトレーション 2レジスタ上位	IF1ARB2H4		RW
000B1CH	CAN4 - IF1 メッセージ制御レジスタ下位	IF1MCTRL4	IF1MCTR4	RW
000B1DH	CAN4 - IF1 メッセージ制御レジスタ上位	IF1MCTRH4		RW
000B1EH	CAN4 - IF1 データ A1下位	IF1DTA1L4	IF1DTA14	RW
000B1FH	CAN4 - IF1 データ A1上位	IF1DTA1H4		RW
000B20H	CAN4 - IF1 データ A2下位	IF1DTA2L4	IF1DTA24	RW
000B21H	CAN4 - IF1 データ A2上位	IF1DTA2H4		RW
000B22H	CAN4 - IF1 データ B1下位	IF1DTB1L4	IF1DTB14	RW
000B23H	CAN4 - IF1 データ B1上位	IF1DTB1H4		RW
000B24H	CAN4 - IF1 データ B2下位	IF1DTB2L4	IF1DTB24	RW
000B25H	CAN4 - IF1 データ B2上位	IF1DTB2H4		RW
000B26H-000B3FH	予約			-
000B40H	CAN4 - IF2 コマンド要求レジスタ下位	IF2CREQL4	IF2CREQ4	RW

Table 37-1. MB96V300BのI/Oマップ

アドレス	レジスタ	8ビット アクセス時の レジスタ略称名	16ビット アクセス時の レジスタ略称名	属性
000B41H	CAN4 - IF2 コマンド要求レジスタ上位	IF2CREQH4		RW
000B42H	CAN4 - IF2 コマンドマスクレジスタ下位	IF2CMSKL4	IF2CMSK4	RW
000B43H	CAN4 - IF2 コマンドマスクレジスタ上位 (予約)	IF2CMSKH4		R
000B44H	CAN4 - IF2 マスク 1レジスタ下位	IF2MSK1L4	IF2MSK14	RW
000B45H	CAN4 - IF2 マスク 1レジスタ上位	IF2MSK1H4		RW
000B46H	CAN4 - IF2 マスク 2レジスタ下位	IF2MSK2L4	IF2MSK24	RW
000B47H	CAN4 - IF2 マスク 2レジスタ上位	IF2MSK2H4		RW
000B48H	CAN4 - IF2 アービトレーション 1レジスタ下位	IF2ARB1L4	IF2ARB14	RW
000B49H	CAN4 - IF2 アービトレーション 1レジスタ上位	IF2ARB1H4		RW
000B4AH	CAN4 - IF2 アービトレーション 2レジスタ下位	IF2ARB2L4	IF2ARB24	RW
000B4BH	CAN4 - IF2 アービトレーション 2レジスタ上位	IF2ARB2H4		RW
000B4CH	CAN4 - IF2 メッセージ制御レジスタ下位	IF2MCTRL4	IF2MCTR4	RW
000B4DH	CAN4 - IF2 メッセージ制御レジスタ上位	IF2MCTRH4		RW
000B4EH	CAN4 - IF2 データ A1下位	IF2DTA1L4	IF2DTA14	RW
000B4FH	CAN4 - IF2 データ A1上位	IF2DTA1H4		RW
000B50H	CAN4 - IF2 データ A2下位	IF2DTA2L4	IF2DTA24	RW
000B51H	CAN4 - IF2 データ A2上位	IF2DTA2H4		RW
000B52H	CAN4 - IF2 データ B1下位	IF2DTB1L4	IF2DTB14	RW
000B53H	CAN4 - IF2 データ B1上位	IF2DTB1H4		RW
000B54H	CAN4 - IF2 データ B2下位	IF2DTB2L4	IF2DTB24	RW
000B55H	CAN4 - IF2 データ B2上位	IF2DTB2H4		RW
000B56H-000B7FH	予約			-
000B80H	CAN4 - 送信要求 1レジスタ下位	TREQR1L4	TREQR14	R
000B81H	CAN4 - 送信要求 1レジスタ上位	TREQR1H4		R
000B82H	CAN4 - 送信要求 2レジスタ下位	TREQR2L4	TREQR24	R
000B83H	CAN4 - 送信要求 2レジスタ上位	TREQR2H4		R
000B84H-000B8FH	予約			-
000B90H	CAN4 - 新規データ 1レジスタ下位	NEWDT1L4	NEWDT14	R
000B91H	CAN4 - 新規データ 1レジスタ上位	NEWDT1H4		R
000B92H	CAN4 - 新規データ 2レジスタ下位	NEWDT2L4	NEWDT24	R
000B93H	CAN4 - 新規データ 2レジスタ上位	NEWDT2H4		R
000B94H-000B9FH	予約			-
000BA0H	CAN4 - 割込み保留 1レジスタ下位	INTPND1L4	INTPND14	R
000BA1H	CAN4 - 割込み保留 1レジスタ上位	INTPND1H4		R
000BA2H	CAN4 - 割込み保留 2レジスタ下位	INTPND2L4	INTPND24	R
000BA3H	CAN4 - 割込み保留 2レジスタ上位	INTPND2H4		R
000BA4H-000BAFH	予約			-
000BB0H	CAN4 - メッセージ有効 1レジスタ下位	MSGVAL1L4	MSGVAL14	R
000BB1H	CAN4 - メッセージ有効 1レジスタ上位	MSGVAL1H4		R
000BB2H	CAN4 - メッセージ有効 2レジスタ下位	MSGVAL2L4	MSGVAL24	R

Table 37-1. MB96V300BのI/Oマップ

アドレス	レジスタ	8ビット アクセス時の レジスタ略称名	16ビット アクセス時の レジスタ略称名	属性
000BB3H	CAN4 - メッセージ有効 2レジスタ上位	MSGVAL2H4		R
000BB4H- 000BCDH	予約			-
000BCEH	CAN4 - 出力許可レジスタ	COER4		RW
000BCFH- 000BFFH	予約			-

37.2 命令

F²MC-16FXの命令を説明します。

37.2.1 命令の種類

F²MC-16FX には, 351 種類の命令があります。各命令の有効なアドレスフィールドを使用するか,もしくは命令コード自体を使用することにより,アドレス指定されます。

命令の種類

F²MC-16FX には , 以下の 351 種類の命令があります。

- 転送系命令 (バイト) 41 命令
- 転送系命令 (ワード, ロングワード) 38 命令
- 加減算命令 (バイト, ワード, ロングワード) 42 命令
- 増減算命令 (バイト, ワード, ロングワード) 12 命令
- 比較命令 (バイト, ワード, ロングワード) 11 命令
- 符号なし乗除算命令 (ワード, ロングワード) 11 命令
- 符号付き乗除算命令 (ワード, ロングワード) 11 命令
- 論理演算命令 (バイト, ワード, ロングワード) 45 命令
- 符号反転命令 (バイト, ワード) 6 命令
- シフト/ノーマライズ命令 (バイト, ワード, ロングワード) 19 命令
- 分岐命令 50 命令 (分岐命令1: 31 命令, 分岐命令2: 19 命令)
- アキュムレータ操作命令 (バイト, ワード) 6 命令
- その他制御命令 (バイト, ワード, ロングワード) 28 命令
- ビット操作命令 21 命令
- ストリング命令 10 命令

37.2.2 アドレス指定

F^{2}MC-16FX では,アドレス形式は命令の実効アドレスフィールド,もしくは命令コード自体 (インプライド) によって決定されます。アドレス形式が命令コード自体によって決定される場合は,使用する命令コードに合わせてアドレスを指定します。命令の中には,ユーザが数種類のアドレス指定を選択できるものがあります。

アドレス指定

F^{2}MC-16FX には , 以下の 23 種類のアドレス指定があります。

- 即値 (#imm)
- レジスタ直接
- 直接分岐アドレス (addr16)
- 物理直接分岐アドレス (addr24)
- I/O 直接 (io)
- 短縮直接アドレス (dir)
- 直接アドレス (addr16)
- I/O 直接ビットアドレス (io:bp)
- 短縮直接ビットアドレス (dir:bp)
- 直接ビットアドレス (addr16:bp)
- ベクタアドレス (#vct)
- レジスタ間接 (@RWj j = 0 ～ 3)
- ポストインクリメント付きレジスタ間接 (@RWj+j = 0 ～ 3)
- オフセット付きレジスタ間接 (@RWi+disp8 i = 0 ～ 7, @RWj+disp16 j = 0 ～ 3)
- オフセット付きロングレジスタ間接 (@RLi+disp8 i = 0 ～ 3)
- オフセット付きプログラムカウンタ間接 (@PC+disp16)
- ベースインデックス付きレジスタ間接 (@RW0+RW7, @RW1+RW7)
- プログラムカウンタ相対分岐アドレス (rel)
- レジスタリスト (rlst)
- アキュムレータ間接 (@A)
- アキュムレータ間接分岐アドレス (@A)
- 間接指定分岐アドレス (@ear)
- 間接指定分岐アドレス (@eam)

実効アドレスフィールド

Table 37-2は,実効アドレスフィールドで指定されるアドレス形式を示します。

Table 37-2. 実効アドレスフィールド

コード	表記			アドレス形式	デフォルトバンク
00	R0	RW0	RL0	レジスタ直接:個々の部分は左から順に,バイト,ワード,ロングワードの型に対応します。	なし
01	R1	RW1	(RL0)		
02	R2	RW2	RL1		
03	R3	RW3	(RL1)		
04	R4	RW4	RL2		
05	R5	RW5	(RL2)		
06	R6	RW6	RL3		
07	R7	RW7	(RL3)		
08	@RW0			レジスタ間接	DTB
09	@RW1				DTB
0A	@RW2				ADB
0B	@RW3				SPB
0C	@RW0+			ポストインクリメント付きレジスタ間接	DTB
0D	@RW1+				DTB
0E	@RW2+				ADB
0F	@RW3+				SPB
10	@RW0+disp8			8 ビットディスプレースメント付きレジスタ間接	DTB
11	@RW1+disp8				DTB
12	@RW2+disp8				ADB
13	@RW3+disp8				SPB
14	@RW4+disp8			8 ビットディスプレースメント付きレジスタ間接	DTB
15	@RW5+disp8				DTB
16	@RW6+disp8				ADB
17	@RW7+disp8				SPB
18	@RW0+disp16			16 ビットディスプレースメント付きレジスタ間接	DTB
19	@RW1+disp16				DTB
1A	@RW2+disp16				ADB
1B	@RW3+disp16				SPB
1C	@RW0+RW7			インデックス付きレジスタ間接	DTB
1D	@RW1+RW7				DTB
1E	@PC+disp16			16 ビットディスプレースメント付きPC間接	PCB
1F	addr16			直接アドレス	DTB

37.2.3 直接アドレス指定

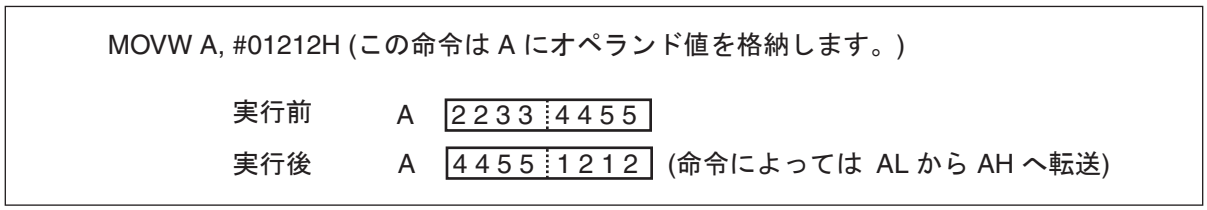
直接アドレス指定方式では,オペランド値,レジスタ,もしくはアドレスを直接指定します。

37.2.3.1 直接アドレス指定

即値アドレス指定(#imm)

オペランド値を直接指定します (#imm4/ #imm8/ #imm16/ #imm32) 。

Figure 37-1. 即値アドレス指定(#imm)の例



レジスタ直接アドレス指定

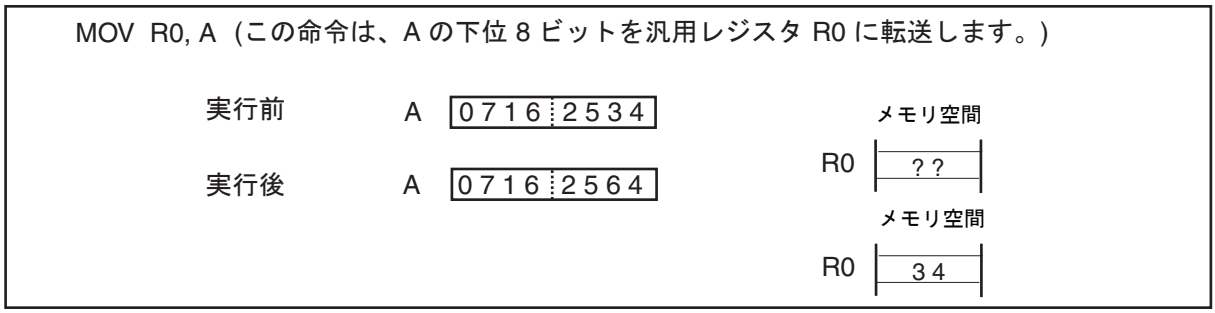
オペランドとしてレジスタを直接指定します。Table 37-3は,指定できるレジスタを示します。Figure 37-2に,レジスタ直接アドレス指定の例を示します。

Table 37-3. 直接アドレス指定レジスタ

汎用レジスタ	バイト	R0, R1, R2, R3, R4, R5, R6, R7
	ワード	RW0, RW1, RW2, RW3, RW4, RW5, RW6, RW7
	ロングワード	RL0, RL1, RL2, RL3
専用レジスタ	アキュムレータ	A, AL
	ポインタ	SP *
	バンク	PCB, DTB, USB, SSB, ADB
	ページ	DPR
	制御	PS, CCR, RP, ILM

* : 条件コードレジスタ (CCR)中の S フラグビットの値に応じて,ユーザスタックポインタ (USP)と,システムスタックポインタ (SSP) のどちらか一方が選択され,使用されます。分岐系の命令では,プログラムカウンタ (PC) は命令のオペランドに記述されませんが,暗黙的に指定されます。

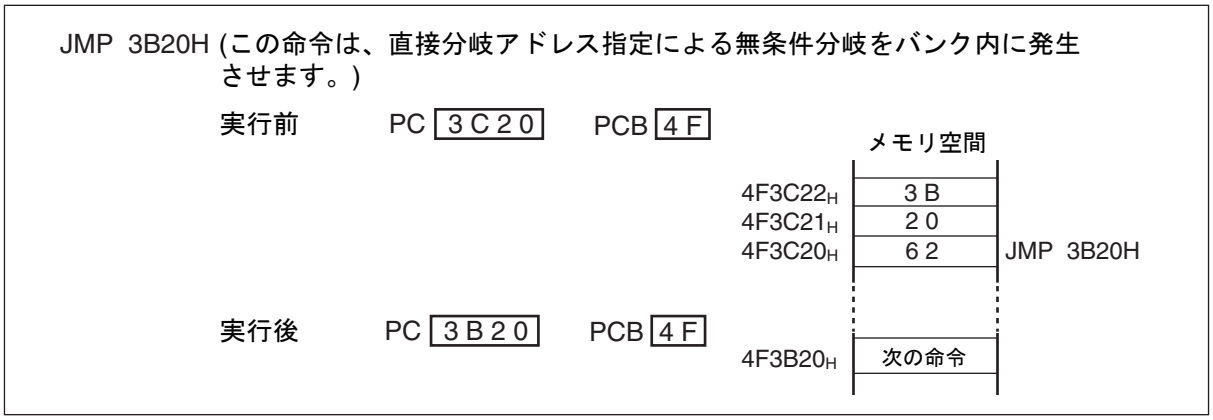
Figure 37-2. レジスタ直接アドレス指定の例



直接分岐アドレス指定(addr16)

分岐先のアドレスをディスプレースメントで直接指定します。ディスプレースメントのサイズは 16 ビットで , 論理アドレス空間内での分岐先を示します。直接分岐アドレス指定は , 無条件分岐 , サブルーチン呼出し , またはソフトウェア割込み命令に使用します。アドレスの bit16 ～ bit23 はプログラムバンクレジスタ (PCB) により指定されます。

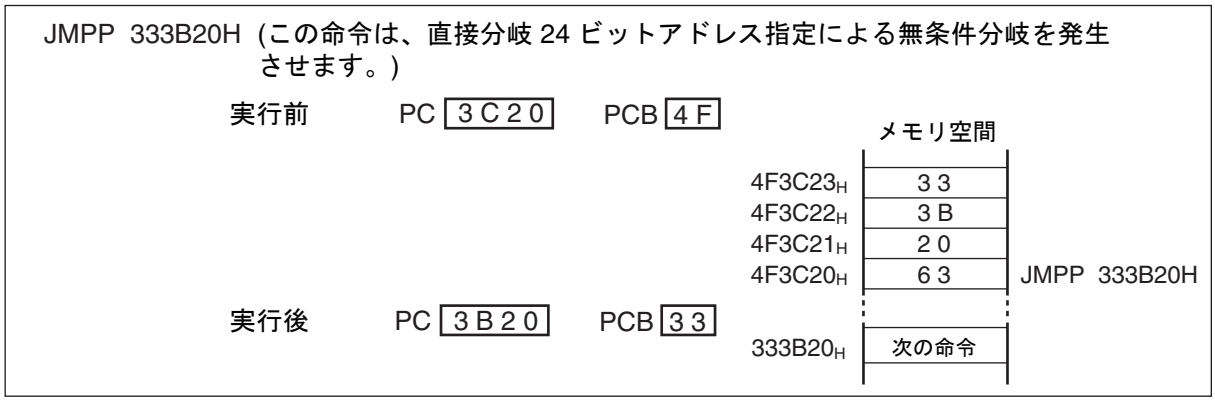
Figure 37-3. 直接分岐アドレス指定(addr16)の例



物理直接分岐アドレス指定(addr24)

分岐先のアドレスをディスプレースメントで直接指定します。ディスプレースメントのサイズは 24 ビットです。物理直接分岐アドレス指定は , 無条件分岐 , サブルーチン呼出し , またはソフトウェア割込み命令に使用します。

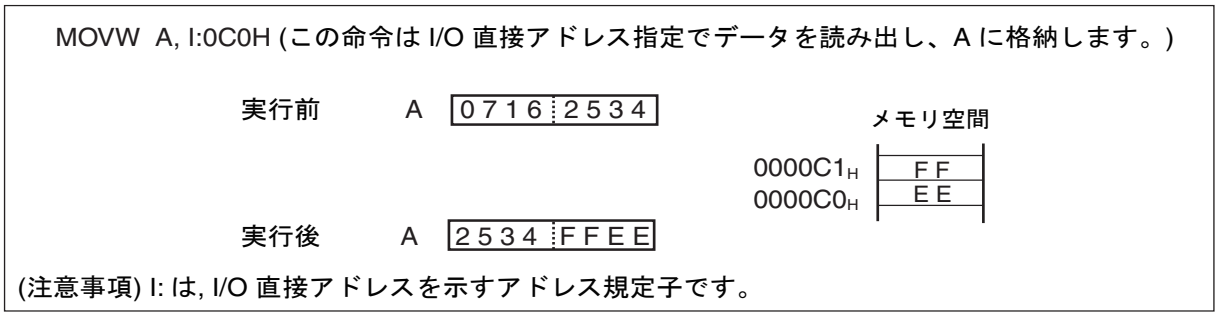
Figure 37-4. 直接分岐アドレス指定(addr24)の例



I/O直接アドレス指定(io)

オペランドのメモリアドレスを 8 ビットのディスプレースメントで直接指定します。物理アドレス空間 000000_H ～ $0000FF_H$ の I/O アドレス空間が , データバンクレジスタ (DTB) とダイレクトページレジスタ (DPR) にかかわらず , アクセスされます。バンクアドレス指定用のバンクセレクトプレフィックスは , I/O 直接アドレス指定を使用した命令の前に指定される場合は無効です。

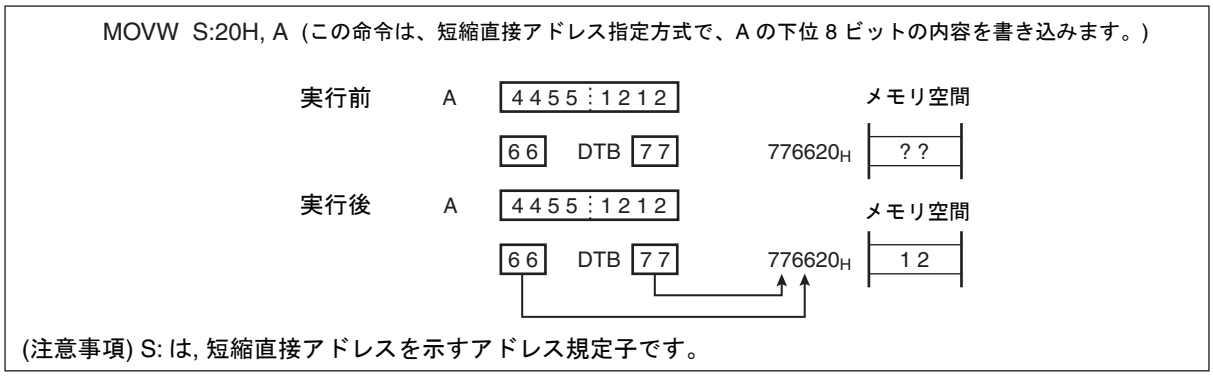
Figure 37-5. 直接アドレス指定(io)の例



短縮直接アドレス指定(dir)

オペランドでメモリアドレスの下位 8 ビットを直接指定します。アドレスの bit8 ～ bit15 はダイレクトページレジスタ (DPR) により指定されます。アドレスの bit16 ～ bit23 はデータバンクレジスタ (DTB) により指定されます。

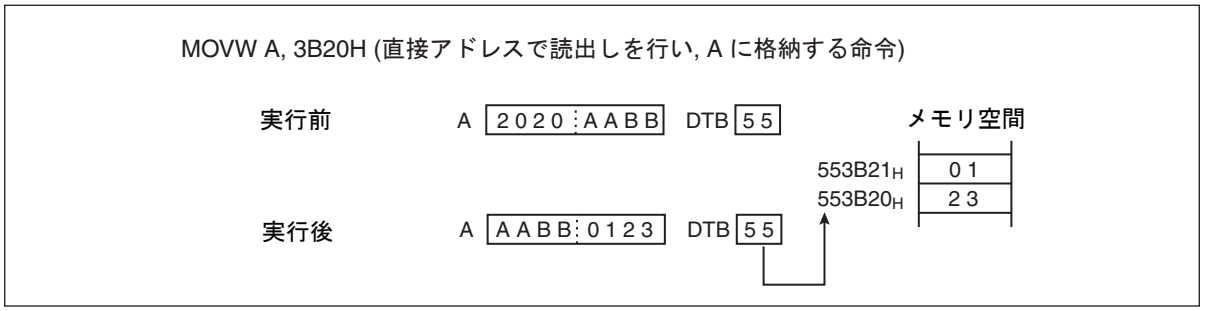
Figure 37-6. 短縮直接アドレス指定(dir)の例



直接アドレス指定(addr16)

オペランドでメモリアドレスの下位 16 ビットを直接指定します。アドレスの bit16 ～ bit23 はデータバンクレジスタ (DTB) により指定されます。アクセス空間アドレス指定のためのプレフィックス命令はアドレス指定のこの方式に対しては無効です。

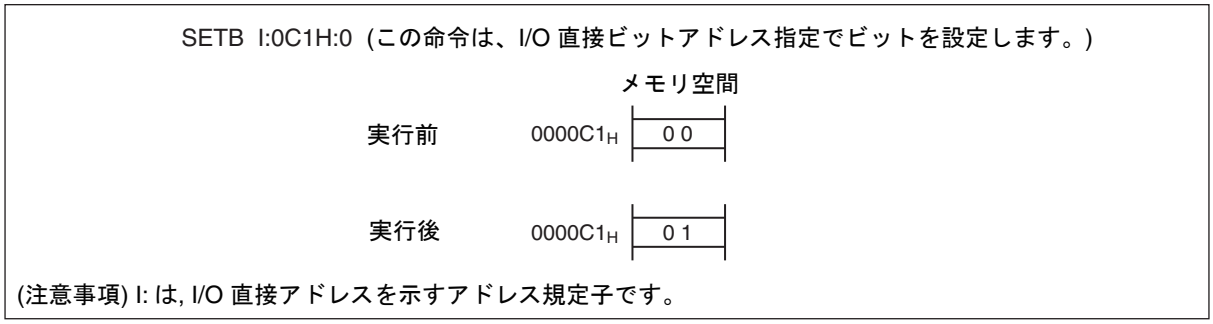
Figure 37-7. 直接アドレス指定(addr16)の例



I/O直接ビットアドレス指定(io:bp)

物理アドレス 000000_H ～ $0000FF_H$ のビットを直接指定します。ビットの位置は「:bp」で示され , 数字の大きい方が , 最上位ビット (MSB), 小さい方が最下位ビット (LSB) となります。

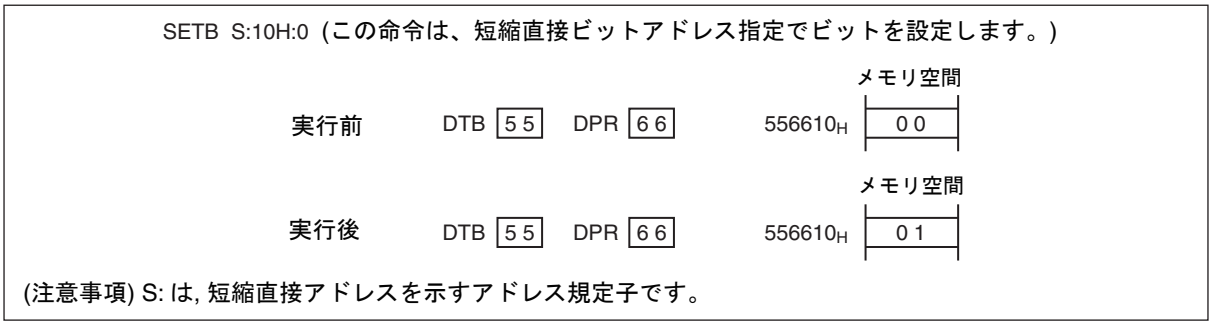
Figure 37-8. I/O直接ビットアドレス指定(io:bp)の例



短縮直接ビットアドレス指定(dir:bp)

オペランドでメモリアドレスの下位 8 ビットを直接指定します。アドレスの bit8 ～ bit15 は直接ページレジスタ (DPR) で指定されます。アドレスの bit16 ～ bit23 はデータバンクレジスタ (DTB) で指定されます。ビットの位置は「:bp」で示され , 数字の大きい方が , 最上位ビット (MSB), 小さい方が最下位ビット (LSB) となります。

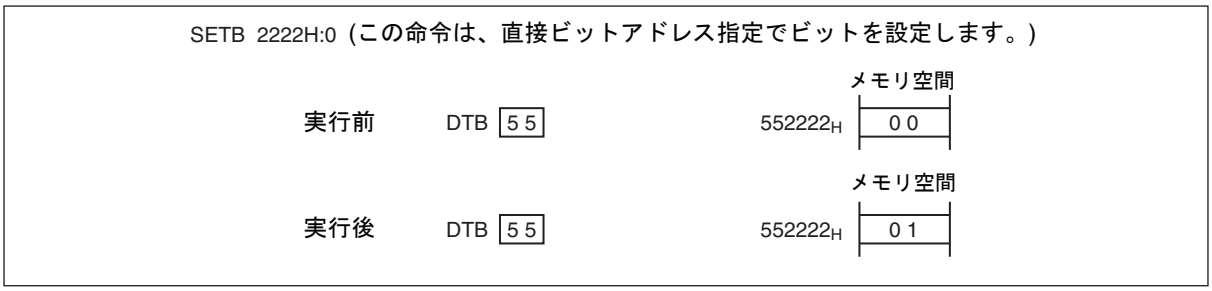
Figure 37-9. 短縮直接ビットアドレス指定(dir:bp)の例



直接ビットアドレス指定(addr16:bp)

64 K バイト内の任意のビットを直接指定します。アドレスの bit16 ～ bit23 はデータバンクレジスタ (DTB) で指定されます。ビットの位置は「:bp」で示され , 数字の大きい方が , 最上位ビット (MSB), 小さい方が最下位ビット (LSB) となります。

Figure 37-10. 直接ビットアドレス指定(addr16:bp)の例



ベクタアドレス指定(#vct)

分岐先のアドレスをオペランドのベクタデータで指定します。ベクタ番号には 2 つのサイズがあり , 4 ビットと 8 ビットです。ベクタアドレス指定は , サブルーチン呼出し , またはソフトウェア割込み命令に使用します。

Figure 37-11. ベクタアドレス指定(#vct)の例

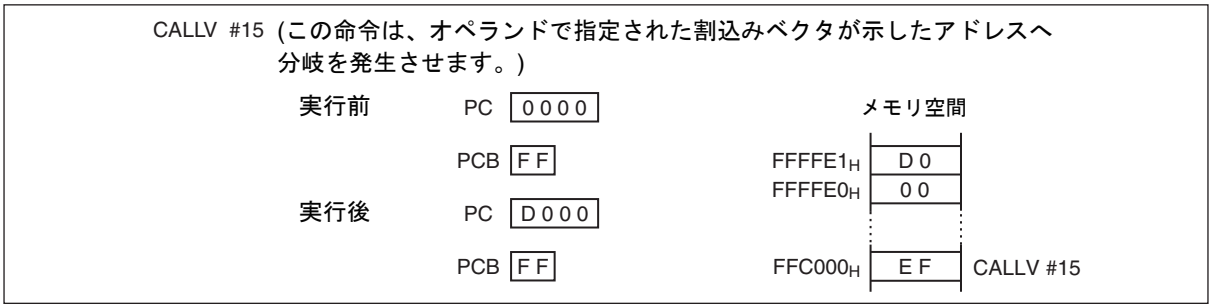


Table 37-4. CALLVベクタリスト

命令	ベクタアドレスL	ベクタアドレスH
CALLV #0	$XXFFFE_H$	$XXFFFF_H$
CALLV #1	$XXFFFC_H$	$XXFFFD_H$
CALLV #2	$XXFFFA_H$	$XXFFFB_H$
CALLV #3	$XXFFF8_H$	$XXFFF9_H$
CALLV #4	$XXFFF6_H$	$XXFFF7_H$
CALLV #5	$XXFFF4_H$	$XXFFF5_H$
CALLV #6	$XXFFF2_H$	$XXFFF3_H$
CALLV #7	$XXFFF0_H$	$XXFFF1_H$
CALLV #8	$XXFFEE_H$	$XXFFEF_H$
CALLV #9	$XXFFEC_H$	$XXFFED_H$
CALLV #10	$XXFFEA_H$	$XXFFEB_H$
CALLV #11	$XXFFE8_H$	$XXFFE9_H$
CALLV #12	$XXFFE6_H$	$XXFFE7_H$
CALLV #13	$XXFFE4_H$	$XXFFE5_H$
CALLV #14	$XXFFE2_H$	$XXFFE3_H$
CALLV #15	$XXFFE0_H$	$XXFFE1_H$

(注意事項) PCB レジスタ値は XX に設定されます。

＜注意事項＞

プログラムバンクレジスタ (PCB) が FF_H の時 , ベクタ領域は INT #vct8 (#0 ～ #7) のベクタ領域にオーバラップします。ベクタアドレス指定は慎重に使用してください (Table 37-4 を参照してください)。

37.2.4 間接アドレス指定

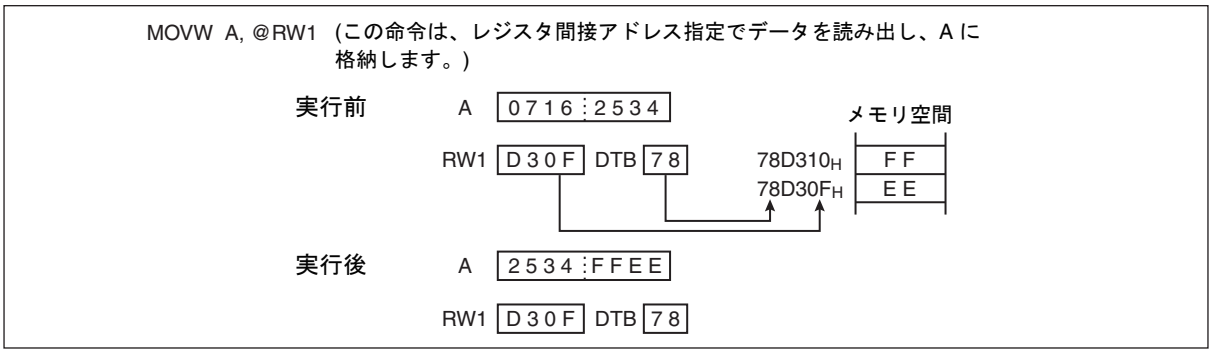
間接アドレス指定方式では,オペランドのアドレスデータで間接的にアドレスを指定します。

37.2.4.1 間接アドレス指定

レジスタ間接アドレス指定(@RWj j = 0 to 3)

アドレスとして汎用レジスタ RWj の内容を使用しメモリにアクセスします。アドレス bit16 ～ bit23 は , RW0/RW1 を使用した場合はデータバンクレジスタ (DTB), RW3 を使用した場合はシステムスタックバンクレジスタ (SSB), もしくはユーザスタックバンクレジスタ (USB), RW2 を使用した場合はアディショナルデータバンクレジスタ (ADB) で示されます。

Figure 37-12. レジスタ間接アドレス指定(@RWj j = 0 to 3)の例

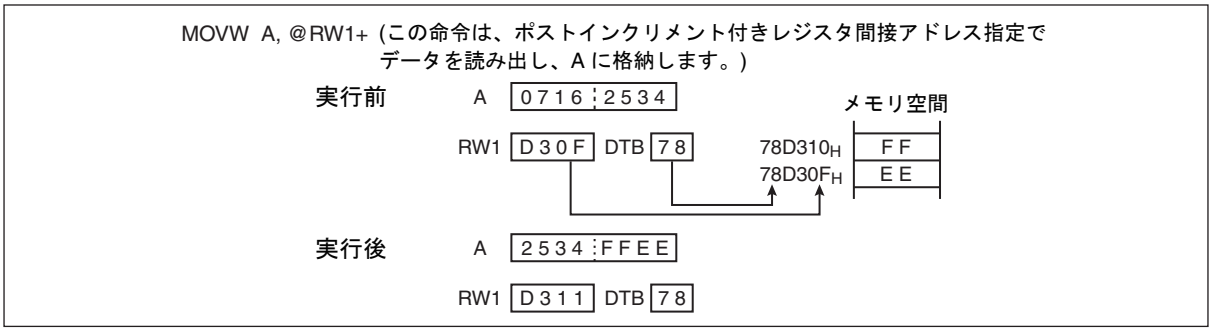


ポストインクリメント付きレジスタ間接アドレス指定(@RWj+ j = 0 to 3)

アドレスとして汎用レジスタ RWj の内容を使用しメモリにアクセスします。オペランド操作後 RWj は , オペランドのサイズ (バイトは 1, ワードは 2, ロングワードは 4) によってインクリメントされます。アドレス bit16 ～ bit23 は , RW0/RW1 を使用した場合はデータバンクレジスタ (DTB), RW3 を使用した場合はシステムスタックバンクレジスタ (SSB), もしくはユーザスタックバンクレジスタ (USB), RW2 を使用した場合はアディショナルデータバンクレジスタ (ADB) で示されます。

ポストインクリメントした結果がインクリメントを指定したレジスタのアドレスであった場合は,インクリメントされた値はその後に参照されます。そのときに , 次の命令が書込みであった場合は , 命令による書込みが優先され , インクリメントされるはずであったレジスタは書き込みデータになります。

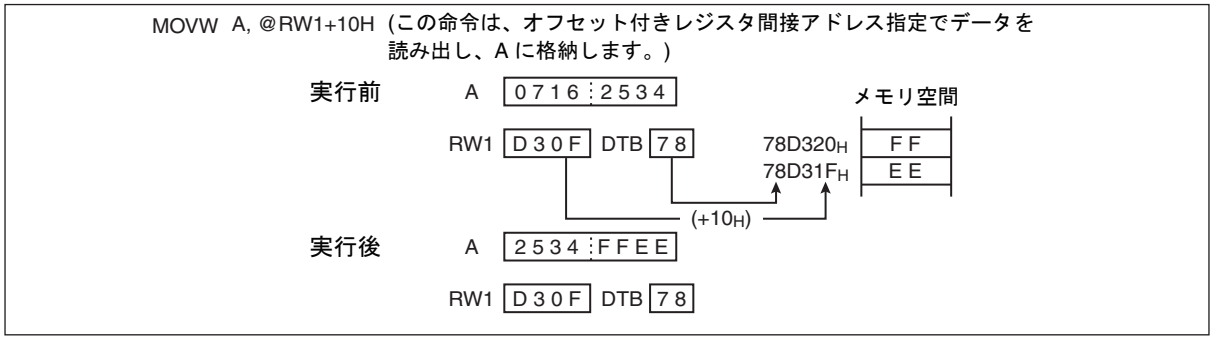
Figure 37-13. ポストインクリメント付きレジスタ間接アドレス指定(@RWj+ j = 0 to 3)の例



オフセット付きレジスタ間接アドレス指定(@RWi + disp8 i = 0 to 7, @RWj + disp16 j = 0 to 3)

汎用レジスタ RWj の内容にオフセットを加算することにより取得されたアドレスを使用しメモリにアクセスします。オフセットの 2 つの種類であるバイト , およびワードのオフセットが使用され , 符号付き数値として加算されます。アドレス bit16 ～ bit23 は ,RW0, RW1, RW4, または RW5 を使用した場合はデータ・バンク・レジスタ (DTB), RW3,または RW7 を使用した場合はシステムスタックバンクレジスタ (SSB),もしくはユーザスタックバンクレジスタ (USB), RW2, または RW6 を使用した場合はアディショナルデータバンクレジスタ (ADB) により示されます。

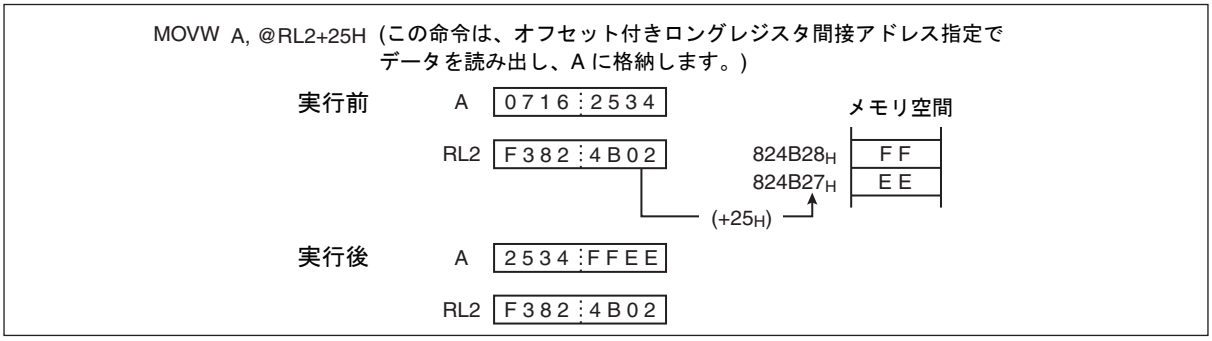
Figure 37-14. オフセット付きレジスタ間接アドレス指定(@RWi + disp8 i = 0 to 7, @RWj + disp16 j = 0 to 3)



オフセット付きロングレジスタ間接アドレス指定(@RLi + disp8 i = 0 to 3)

汎用レジスタ RLi の内容にオフセットを加算することにより取得された 24 下位ビットのアドレスを使用しメモリにアクセスします。オフセットは 8 ビットの長さであり , 符号付き数値として加算されます。

Figure 37-15. オフセット付きロングレジスタ間接アドレス指定(@RLi + disp8 i = 0 to 3)の例

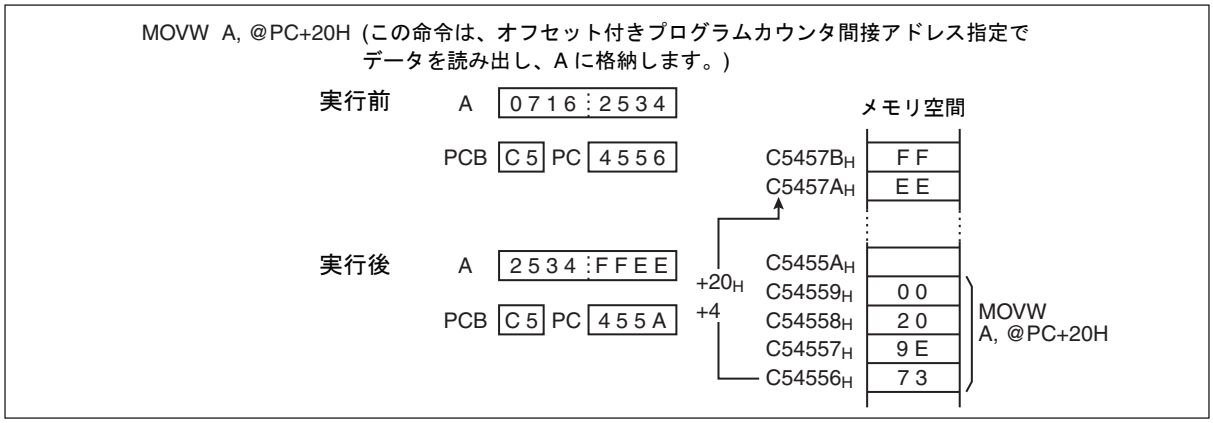


オフセット付きプログラムカウンタ間接アドレス指定(@PC + disp16)

(命令のアドレス + 4 + disp16) で示されるアドレスを使用しメモリにアクセスします。オフセットは 1 ワードの長さです。アドレスの bit16 ～ bit23 はプログラムバンクレジスタ (PCB) により指定されます。以下の命令のオペランドアドレスは , (次の命令のアドレス + disp16) とはみなされないため注意してください。

- DBNZ eam, rel
- DWBNZ eam, rel
- CBNE eam, #imm8, rel
- CWBNE eam, #imm16, rel
- MOV eam, #imm8
- MOVW eam, #imm16

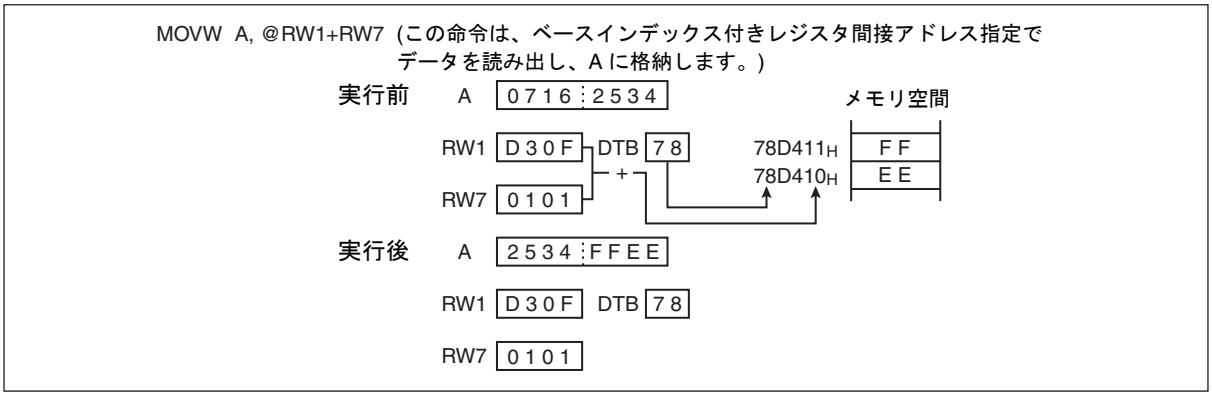
Figure 37-16. オフセット付きプログラムカウンタ間接アドレス指定(@PC + disp16)の例



ベースインデックス付きレジスタ間接アドレス指定(@RW0 + RW7, @RW1 + RW7)

汎用レジスタ RW7 の内容に RW0 または RW1 を加算することにより取得されたアドレスを使用しメモリにアクセスします。アドレスの bit16 ～ bit23 はデータバンクレジスタ (DTB) により示されます。

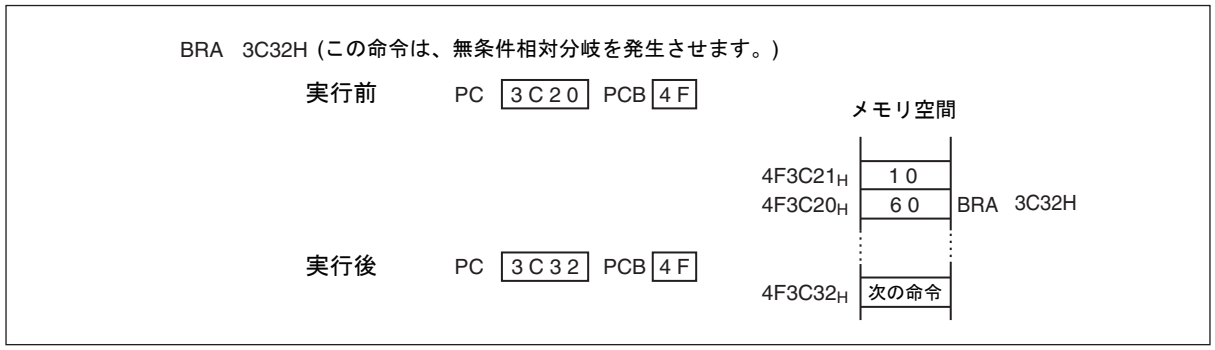
Figure 37-17. ベースインデックス付きレジスタ間接アドレス指定(@RW0 + RW7, @RW1 + RW7)の例



プログラムカウンタ相対分岐アドレス指定(rel)

分岐先のアドレスはプログラムカウンタ (PC) の値に 8 ビットオフセットを加算した値です。加算の結果が 16 ビットを超えた場合は , バンクレジスタのインクリメントまたはデクリメントは行われず , 超えた分は無視され , アドレスは 64 K バイトのバンク内で閉じられます。このアドレス指定は , 条件と無条件の両方の分岐命令に使用されます。アドレスの bit16 ～ bit23 はプログラムバンクレジスタ (PCB) により示されます。

Figure 37-18. プログラムカウンタ相対分岐アドレス指定(rel)の例



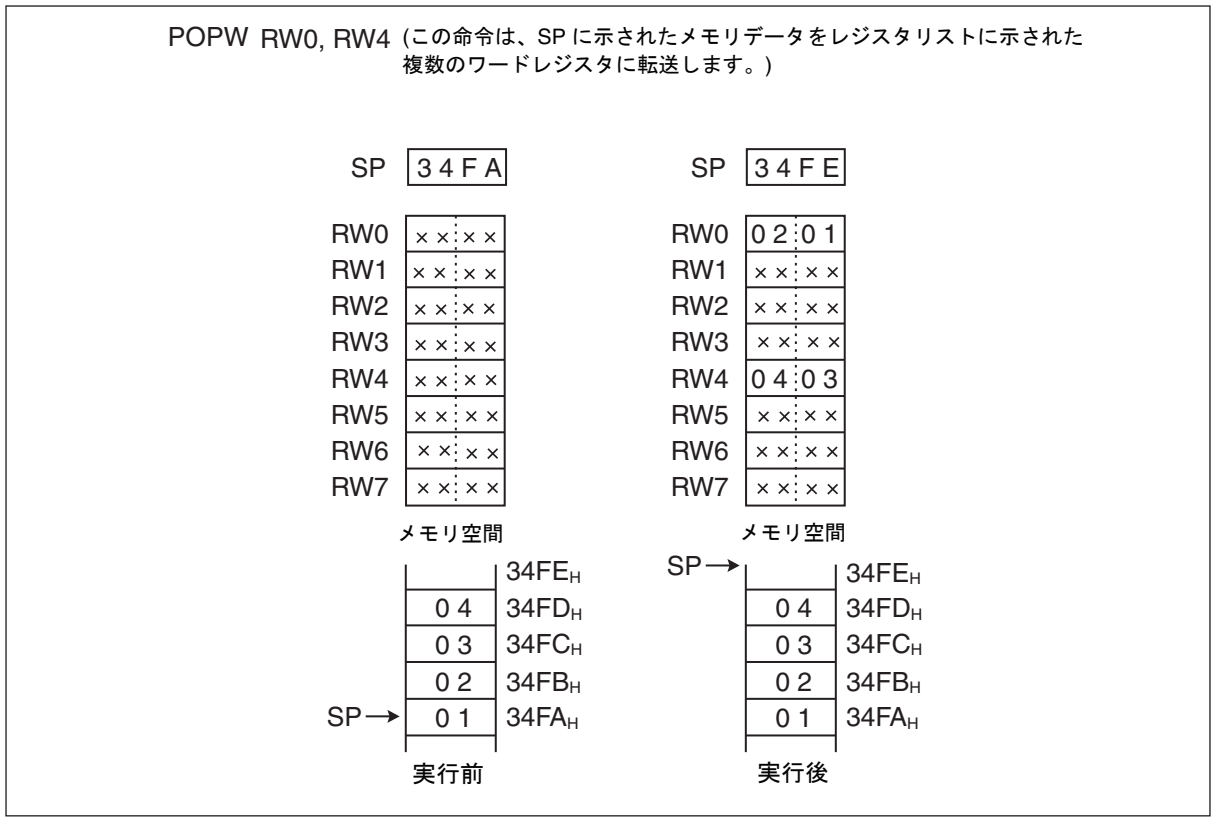
レジスタレスト(rlst)

スタックに "PUSH" またはスタックから "POP" されるレジスタを指定します。

Figure 37-19. レジスタリストの構成



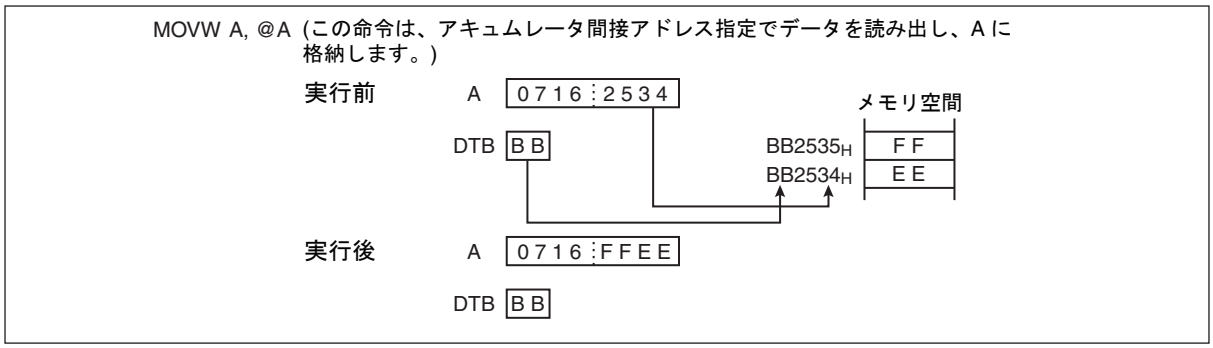
Figure 37-20. レジスタリスト(rlist)の例



アキュームレータ間接アドレス指定(@A)

アキュムレータ (AL) の下位バイト (16 ビット) の内容で示されるアドレスを使用しメモリにアクセスします。アドレスの bit16 ～ bit23 はデータバンクレジスタ (DTB) のニーモニックにより指定されます。

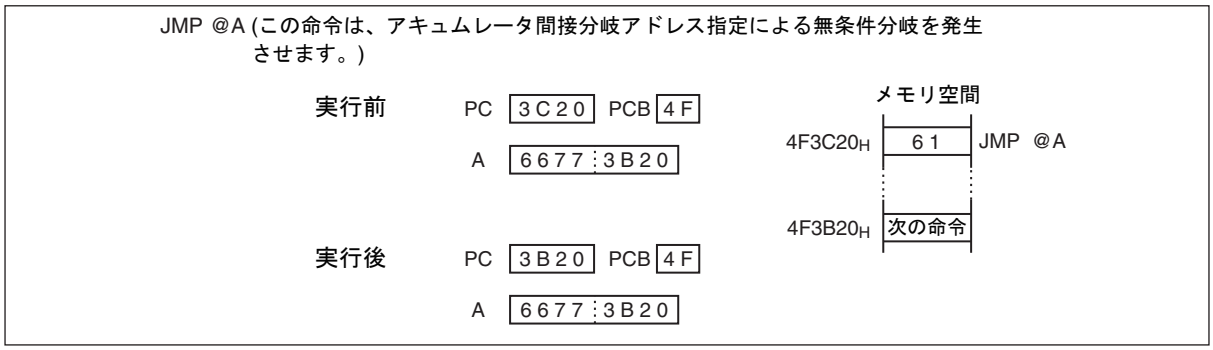
Figure 37-21. アキュームレータ間接アドレス指定(@A)の例



アキュームレータ間接分岐アドレス指定(@A)

分岐先アドレスは , アキュムレータ下位バイト (AL) の内容 (16 ビット) です。バンクアドレス空間内での分岐先を示し , アドレスの bit16 ～ bit23 はプログラムバンクレジスタ (PCB) により示されます。ジャンプコンテキスト (JCTX) 命令の場合は , アドレスの bit16 ～ bit23 はデータバンクレジスタ (DTB) により指定されます。このアドレス指定は , 無条件の分岐命令に使用されます。

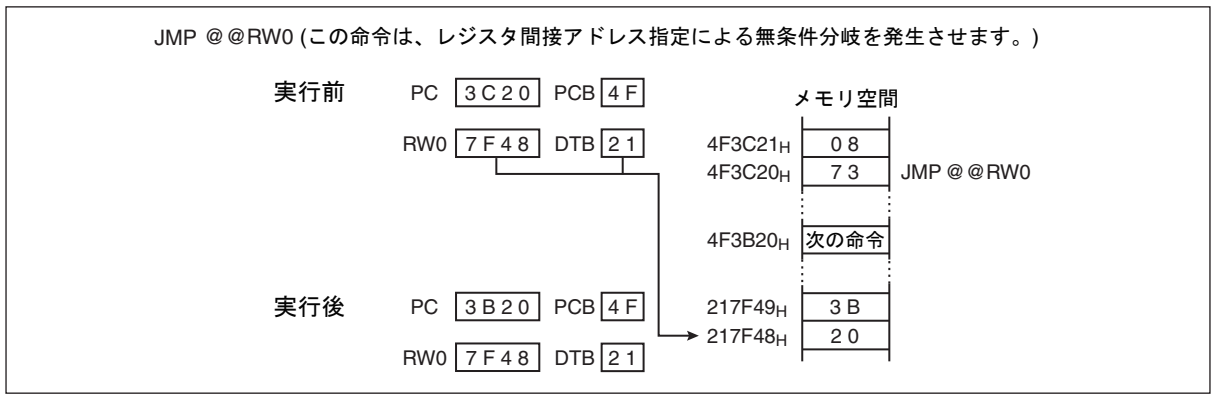
Figure 37-22. アキュームレータ間接分岐アドレス指定(@A)の例



間接指定分岐アドレス指定(@ear)

分岐先アドレスは ear で示されるアドレスのワードデータです。

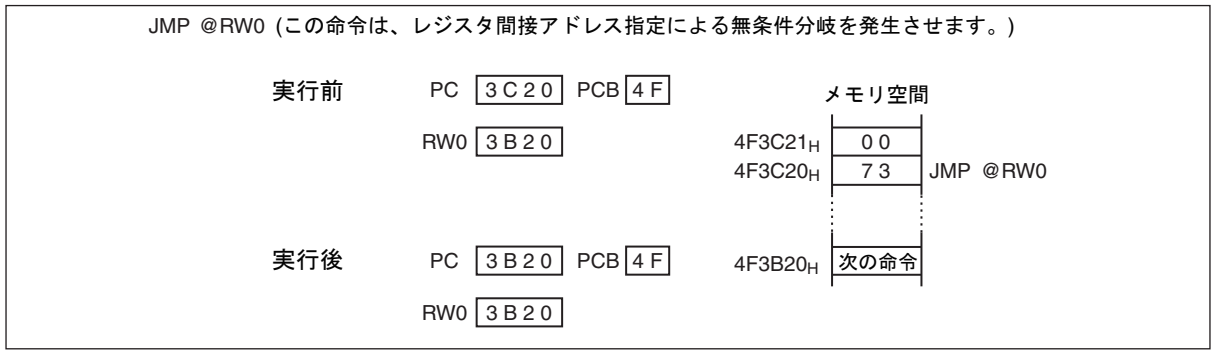
Figure 37-23. 間接指定分岐アドレス指定(@ear)の例



間接指定分岐アドレス指定(@eam)

分岐先アドレスは eam で示されるアドレスのワードデータです。

Figure 37-24. 間接指定分岐アドレス指定(@eam)の例



37.2.5 実行サイクル数

命令の実行に要するサイクル数 (実行サイクル数) は, 各命令の"サイクル数"の値とデータのアクセス条件で決まる"奇数アドレス補正"の値を加算することで得られます。

実行サイクル数

実際の命令実行においては，命令フェッチの遅れ，データアクセスの競合等により，実行サイクル数が計算値より増大することがあります。とくに，外部バスインタフェースを使用して，外部バスからの命令フェッチや，データアクセスを行うときには，実行サイクル数が増大します。

奇数アドレス補正

一部の命令においては，奇数アドレスへのデータアクセスを行うときに実行サイクルが増大します。奇数アドレスのデータアクセスで増大するサイクル数を奇数アドレス補正として，命令一覧の項目 "B" に示してあります。

37.2.6 実効アドレスフィールド

Table 37-5は,実効アドレスフィールドを示します。

実効アドレスフィールド

Table 37-5. 実効アドレスフィールド

コード	表記			アドレス形式	拡張アドレス部分のバイト数 * (*1)
00	R0	RW0	RL0	レジスタ直接:個々の部分は左から順に,バイト,ワード,ロングワードの型に対応する	-
01	R1	RW1	(RL0)		
02	R2	RW2	RL1		
03	R3	RW3	(RL1)		
04	R4	RW4	RL2		
05	R5	RW5	(RL2)		
06	R6	RW6	RL3		
07	R7	RW7	(RL3)		
08	@RW0			レジスタ間接	0
09	@RW1				
0A	@RW2				
0B	@RW3				
0C	@RW0+			ポストインクリメント付きレジスタ間接	0
0D	@RW1+				
0E	@RW2+				
0F	@RW3+				
10	@RW0+disp8			8 ビットディスプレースメント付きレジスタ間接	1
11	@RW1+disp8				
12	@RW2+disp8				
13	@RW3+disp8				
14	@RW4+disp8				
15	@RW5+disp8				
16	@RW6+disp8				
17	@RW7+disp8				
18	@RW0+disp16			16 ビットディスプレースメント付きレジスタ間接	2
19	@RW1+disp16				
1A	@RW2+disp16				
1B	@RW3+disp16				
1C	@RW0+RW7			インデックス付きレジスタ間接	0
1D	@RW1+RW7				0
1E	@PC+disp16			16 ビットディスプレースメント付き PC 間接	2
1F	addr16			直接アドレス	2

* :拡張アドレス部分の各バイト数は,「B.8 F2MC-16FX命令一覧」のバイトカウント欄の + に該当します。

37.2.7 命令一覧の読み方

Table 37-6では, F^{2}MC-16FX 命令一覧で使用される項目を説明し, Table 37-7では,使用される記号を説明しています。

37.2.7.1 命令表記項目と記号の説明

Table 37-6. 命令リストの項目の説明

項目	説明
ニーモニック	英大文字 , 記号 : アセンブラ上もそのままに記述 英小文字 : アセンブラ上では , 書き換えて記述 英小文字の後の数: 命令中のビット幅を示す
#	バイト数を示す
	サイクル数を示す
B	奇数アドレスへのデータアクセスを行うときの , 奇数アドレス補正を示す 命令実行の実サイクル数は, ～ 欄の数値にこの補正値を加算したもの
オペレーション	命令の動作を示す
LH	アキュムレータの bit15 ～ bit8 に対する特殊動作を示す Z : 0 を転送 X: 符号拡張して転送 -: 転送なし
AH	アキュムレータの上位ビット 16 に対する特殊動作を示す * : AL から AH へ転送 - : 転送なし Z : AH へ 00_H を転送 X: AL の符号拡張の後, AH へ 00_H または FF_H を転送
I	各フラグの状態を示す : I (割込み許可), S (スタック), T (スティッキィビット), N (ネガティブ), Z (ゼロ), V (オーバフロー), C (キャリ) * : 命令の実行により変化 - : 変化なし Z : 命令の実行により設定 X : 命令の実行により再設定 R : "0" にリセット S: "1"にセット
S	
T	
N	
Z	
V	
C	
RMW	命令が , リードモディファイライト命令 (I 命令でメモリからデータを読み取り , 結果をメモリへ書き込む) であるかどうかを示す *: リードモディファイライト命令である -: リードモディファイライト命令ではない **(注意事項)** 読取りと書込みの動作間で意味が異なるアドレスには使用できない

Table 37-7. 命令リストの記号の説明

記号	説明
A	32 ビットアキュムレータ 使用されるビットの長さは , 命令により変わる バイト : AL の下位 8 ビット ワード : AL の 16 ビット ロングワード : AL と AH の 32 ビット
AH	A の上位 16 ビット
AL	A の下位 16 ビット
SP	スタックポインタ (USP または SSP)
PC	プログラムカウンタ
PCB	プログラムバンクレジスタ
DTB	データバンクレジスタ
ADB	アディショナルデータバンクレジスタ
SSB	システムスタックバンクレジスタ
USB	ユーザスタックバンクレジスタ
SPB	カレントスタックバンクレジスタ (SSB または USB)
DPR	ダイレクトページレジスタ
brg1	DTB, ADB, SSB, USB, DPR, PCB, SPB
brg2	DTB, ADB, SSB, USB, DPR, SPB
brg3	DTB, ADB, PCB, SPB
Ri	R0, R1, R2, R3, R4, R5, R6, R7
RWi	RW0, RW1, RW2, RW3, RW4, RW5, RW6, RW7
RWj	RW0, RW1, RW2, RW3
RLi	RL0, RL1, RL2, RL3
dir	短縮直接アドレス指定
addr16	直接アドレス指定
addr24	物理直接アドレス指定
ad24 0-15	addr24 の bit0 ～ bit15
ad24 16-23	addr24 のbit16 ～bit23
io	I/O 領域 (000000_H ～ $0000FF_H$)
#imm4	4 ビット即値データ
#imm8	8 ビット即値データ
#imm16	16 ビット即値データ
#imm32	32 ビット即値データ
ext (imm8)	8 ビット即値データを符号拡張した 16 ビットデータ
disp8	8 ビットディスプレースメント
disp16	16 ビットディスプレースメント
bp	ビットオフセット
vct4	ベクタ番号 (0 ～ 15)
vct8	ベクタ番号 (0 ～ 255)
( ) b	ビットアドレス

Table 37-7 命令リストの記号の説明

記号	説明
rel	PC 相対分岐
ear	実効アドレス指定 (コード 00_H ～ 07_H)
eam	実効アドレス指定 (コード 08_H ～ $1F_H$)
rlst	レジスタリスト

37.2.8 F²MC-16FX命令一覧

Table 37-8 ～ Table 37-25 は, F²MC-16FXで使用される命令の一覧です。

F²MC-16FX命令一覧

Table 37-8. 転送命令（バイト）：41命令

ニーモニック	オペランド	#	～	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
MOV	A, dir	2	1	0	byte (A) ← (dir)	Z	*	-	-	-	*	*	-	-	-
MOV	A, addr16	3	1	0	byte (A) ← (addr16)	Z	*	-	-	-	*	*	-	-	-
MOV	A, Ri	1	1	0	byte (A) ← (Ri)	Z	*	-	-	-	*	*	-	-	-
MOV	A, ear	2	1	0	byte (A) ← (ear)	Z	*	-	-	-	*	*	-	-	-
MOV	A, eam	2+	1	0	byte (A) ← (eam)	Z	*	-	-	-	*	*	-	-	-
MOV	A, io	2	1	0	byte (A) ← (io)	Z	*	-	-	-	*	*	-	-	-
MOV	A, #imm8	2	1	0	byte (A) ← imm8	Z	*	-	-	-	*	*	-	-	-
MOV	A, @A	2	1	0	byte (A) ← ((A))	Z	-	-	-	-	*	*	-	-	-
MOV	A, @RLi+disp8	3	1	0	byte (A) ← ((RLi)+disp8)	Z	*	-	-	-	*	*	-	-	-
MOVN	A, #imm4	1	1	0	byte (A) ← imm4	Z	*	-	-	-	R	*	-	-	-
MOVX	A, dir	2	1	0	byte (A) ← (dir)	X	*	-	-	-	*	*	-	-	-
MOVX	A, addr16	3	1	0	byte (A) ← (addr16)	X	*	-	-	-	*	*	-	-	-
MOVX	A, Ri	2	1	0	byte (A) ← (Ri)	X	*	-	-	-	*	*	-	-	-
MOVX	A, ear	2	1	0	byte (A) ← (ear)	X	*	-	-	-	*	*	-	-	-
MOVX	A, eam	2+	1	0	byte (A) ← (eam)	X	*	-	-	-	*	*	-	-	-
MOVX	A, io	2	1	0	byte (A) ← (io)	X	*	-	-	-	*	*	-	-	-
MOVX	A, #imm8	2	1	0	byte (A) ← imm8	X	*	-	-	-	*	*	-	-	-
MOVX	A, @A	2	1	0	byte (A) ← ((A))	X	-	-	-	-	*	*	-	-	-
MOVX	A, @RWi+disp8	2	1	0	byte (A) ← ((RWi)+disp8)	X	*	-	-	-	*	*	-	-	-
MOVX	A, @RLi+disp8	3	1	0	byte (A) ← ((RLi)+disp8)	X	*	-	-	-	*	*	-	-	-
MOV	dir, A	2	1	0	byte (dir) ← (A)	-	-	-	-	-	*	*	-	-	-
MOV	addr16, A	3	1	0	byte (addr16) ← (A)	-	-	-	-	-	*	*	-	-	-
MOV	Ri, A	1	1	0	byte (Ri) ← (A)	-	-	-	-	-	*	*	-	-	-
MOV	ear, A	2	1	0	byte (ear) ← (A)	-	-	-	-	-	*	*	-	-	-
MOV	eam, A	2+	1	0	byte (eam) ← (A)	-	-	-	-	-	*	*	-	-	-
MOV	io, A	2	1	0	byte (io) ← (A)	-	-	-	-	-	*	*	-	-	-
MOV	@RLi+disp8, A	3	1	0	byte ((RLi)+disp8) ← (A)	-	-	-	-	-	*	*	-	-	-
MOV	Ri, ear	2	1	0	byte (Ri) ← (ear)	-	-	-	-	-	*	*	-	-	-
MOV	Ri, eam	2+	1	0	byte (Ri) ← (eam)	-	-	-	-	-	*	*	-	-	-
MOV	ear, Ri	2	1	0	byte (ear) ← (Ri)	-	-	-	-	-	*	*	-	-	-
MOV	eam, Ri	2+	1	0	byte (eam) ← (Ri)	-	-	-	-	-	*	*	-	-	-
MOV	Ri, #imm8	2	1	0	byte (Ri) ← imm8	-	-	-	-	-	*	*	-	-	-
MOV	io, #imm8	3	1	0	byte (io) ← imm8	-	-	-	-	-	-	-	-	-	-
MOV	dir, #imm8	3	1	0	byte (dir) ← imm8	-	-	-	-	-	-	-	-	-	-
MOV	ear, #imm8	3	1	0	byte (ear) ← imm8	-	-	-	-	-	*	*	-	-	-
MOV	eam, #imm8	3+	1 / 2[*1]	0	byte (eam) ← imm8	-	-	-	-	-	-	-	-	-	-
MOV	@AL, AH	2	1	0	byte ((A)) ← (AH)	-	-	-	-	-	*	*	-	-	-
XCH	A, ear	2	1	0	byte (A) ←→ (ear)	Z	-	-	-	-	-	-	-	-	-
XCH	A, eam	2+	2	0	byte (A) ←→ (eam)	Z	-	-	-	-	-	-	-	-	-
XCH	Ri, ear	2	2	0	byte (Ri) ←→ (ear)	-	-	-	-	-	-	-	-	-	-
XCH	Ri, eam	2+	2	0	byte (Ri) ←→ (eam)	-	-	-	-	-	-	-	-	-	-

*1 : eam が Ri/@RWi/@RWi+/@RWi+RW7 の場合は 1，それ以外は 2

Table 37-9. 転送命令（ワード, ロングワード）：38命令

ニーモニック	オペランド	#	~	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
MOVW	A, dir	2	1	1	word (A) ← (dir)	-	*	-	-	-	*	*	-	-	-
MOVW	A, addr16	3	1	1	word (A) ← (addr16)	-	*	-	-	-	*	*	-	-	-
MOVW	A, SP	1	1	0	word (A) ← (SP)	-	*	-	-	-	*	*	-	-	-
MOVW	A, RWi	1	1	0	word (A) ← (RWi)	-	*	-	-	-	*	*	-	-	-
MOVW	A, ear	2	1	0	word (A) ← (ear)	-	*	-	-	-	*	*	-	-	-
MOVW	A, eam	2+	1	1	word (A) ← (eam)	-	*	-	-	-	*	*	-	-	-
MOVW	A, io	2	1	1	word (A) ← (io)	-	*	-	-	-	*	*	-	-	-
MOVW	A, @A	2	1	1	word (A) ← ((A))	-	-	-	-	-	*	*	-	-	-
MOVW	A, #imm16	3	1	0	word (A) ← imm16	-	*	-	-	-	*	*	-	-	-
MOVW	A, @RWi+disp8	2	1	1	word (A) ← ((RWi)+disp8)	-	*	-	-	-	*	*	-	-	-
MOVW	A, @RLi+disp8	3	1	1	word (A) ← ((RLi)+disp8)	-	*	-	-	-	*	*	-	-	-
MOVW	dir, A	2	1	1	word (dir) ← (A)	-	-	-	-	-	*	*	-	-	-
MOVW	addr16, A	3	1	1	word (addr16) ← (A)	-	-	-	-	-	*	*	-	-	-
MOVW	SP, A	1	1	0	word (SP) ← (A)	-	-	-	-	-	*	*	-	-	-
MOVW	RWi, A	1	1	0	word (RWi) ← (A)	-	-	-	-	-	*	*	-	-	-
MOVW	ear, A	2	1	0	word (ear) ← (A)	-	-	-	-	-	*	*	-	-	-
MOVW	eam, A	2+	1	1	word (eam) ← (A)	-	-	-	-	-	*	*	-	-	-
MOVW	io, A	2	1	1	word (io) ← (A)	-	-	-	-	-	*	*	-	-	-
MOVW	@RWi+disp8, A	2	1	1	word ((RWi)+disp8) ← (A)	-	-	-	-	-	*	*	-	-	-
MOVW	@RLi+disp8, A	3	1	1	word ((RLi)+disp8) ← (A)	-	-	-	-	-	*	*	-	-	-
MOVW	RWi, ear	2	1	0	word (RWi) ← (ear)	-	-	-	-	-	*	*	-	-	-
MOVW	RWi, eam	2+	1	1	word (RWi) ← (eam)	-	-	-	-	-	*	*	-	-	-
MOVW	ear, RWi	2	1	0	word (ear) ← (RWi)	-	-	-	-	-	*	*	-	-	-
MOVW	eam, RWi	2+	1	1	word (eam) ← (RWi)	-	-	-	-	-	*	*	-	-	-
MOVW	RWi, #imm16	3	1	0	word (RWi) ← imm16	-	-	-	-	-	*	*	-	-	-
MOVW	io, #imm16	4	1	1	word (io) ← imm16	-	-	-	-	-	-	-	-	-	-
MOVW	ear, #imm16	4	2	0	word (ear) ← imm16	-	-	-	-	-	*	*	-	-	-
MOVW	eam, #imm16	4+	2	1	word (eam) ← imm16	-	-	-	-	-	-	-	-	-	-
MOVW	@AL, AH	2	1	1	word ((A)) ← (AH)	-	-	-	-	-	*	*	-	-	-
XCHW	A, ear	2	1	0	word (A) ←→ ear	-	-	-	-	-	-	-	-	-	-
XCHW	A, eam	2+	2	2	word (A) ←→ eam	-	-	-	-	-	-	-	-	-	-
XCHW	RWi, ear	2	2	0	word (RWi) ←→ ear	-	-	-	-	-	-	-	-	-	-
XCHW	RWi, eam	2+	2	2	word (RWi) ←→ eam	-	-	-	-	-	-	-	-	-	-
MOVL	A, ear	2	2	0	long (A) ← (ear)	-	-	-	-	-	*	*	-	-	-
MOVL	A, eam	2+	2	1	long (A) ← (eam)	-	-	-	-	-	*	*	-	-	-
MOVL	A, #imm32	5	2	0	long (A) ← imm32	-	-	-	-	-	*	*	-	-	-
MOVL	ear, A	2	2	0	long (ear) ← (A)	-	-	-	-	-	*	*	-	-	-
MOVL	eam, A	2+	3 / 2[*1]	1	long (eam) ← (A)	-	-	-	-	-	*	*	-	-	-

*1 : eam が @RWi+ の場合は 3, それ以外は 2

Table 37-10. 加減算命令（バイト, ワード, ロングワード）：42命令

ニーモニック	オペランド	#	~	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
ADD	A, #imm8	2	1	0	byte (A) ← (A) + imm8	Z	-	-	-	-	*	*	*	*	-
ADD	A, dir	2	2	0	byte (A) ← (A) + (dir)	Z	-	-	-	-	*	*	*	*	-
ADD	A, ear	2	1	0	byte (A) ← (A) + (ear)	Z	-	-	-	-	*	*	*	*	-
ADD	A, eam	2+	2	0	byte (A) ← (A) + (eam)	Z	-	-	-	-	*	*	*	*	-
ADD	ear, A	2	1	0	byte (ear) ← (ear) + (A)	-	-	-	-	-	*	*	*	*	-
ADD	eam, A	2+	3	0	byte (eam) ← (eam) + (A)	Z	-	-	-	-	*	*	*	*	*
ADDC	A	1	1	0	byte (A) ← (AH) + (AL) + (C)	Z	-	-	-	-	*	*	*	*	-
ADDC	A, ear	2	1	0	byte (A) ← (A) + (ear) + (C)	Z	-	-	-	-	*	*	*	*	-
ADDC	A, eam	2+	2	0	byte (A) ← (A) + (eam) + (C)	Z	-	-	-	-	*	*	*	*	-
ADDDC	A	1	2	0	byte (A) ← (AH) + (AL) + (C) : decimal	Z	-	-	-	-	*	*	*	*	-
SUB	A, #imm8	2	1	0	byte (A) ← (A) - imm8	Z	-	-	-	-	*	*	*	*	-
SUB	A, dir	2	2	0	byte (A) ← (A) - (dir)	Z	-	-	-	-	*	*	*	*	-
SUB	A, ear	2	1	0	byte (A) ← (A) - (ear)	Z	-	-	-	-	*	*	*	*	-
SUB	A, eam	2+	2	0	byte (A) ← (A) - (eam)	Z	-	-	-	-	*	*	*	*	-
SUB	ear, A	2	1	0	byte (ear) ← (ear) - (A)	-	-	-	-	-	*	*	*	*	-
SUB	eam, A	2+	3	0	byte (eam) ← (eam) - (A)	-	-	-	-	-	*	*	*	*	*
SUBC	A	1	1	0	byte (A) ← (AH) - (AL) - (C)	Z	-	-	-	-	*	*	*	*	-
SUBC	A, ear	2	1	0	byte (A) ← (A) - (ear) - (C)	Z	-	-	-	-	*	*	*	*	-
SUBC	A, eam	2+	2	0	byte (A) ← (A) - (eam) - (C)	Z	-	-	-	-	*	*	*	*	-
SUBDC	A	1	2	0	byte (A) ← (AH) - (AL) - (C) : decimal	Z	-	-	-	-	*	*	*	*	-
ADDW	A	1	1	0	word (A) ← (AH) + (AL)	-	-	-	-	-	*	*	*	*	-
ADDW	A, ear	2	1	0	word (A) ← (A) + (ear)	-	-	-	-	-	*	*	*	*	-
ADDW	A, eam	2+	2	1	word (A) ← (A) + (eam)	-	-	-	-	-	*	*	*	*	-
ADDW	A, #imm16	3	1	0	word (A) ← (A) + imm16	-	-	-	-	-	*	*	*	*	-
ADDW	ear, A	2	1	0	word (ear) ← (ear) + (A)	-	-	-	-	-	*	*	*	*	-
ADDW	eam, A	2+	3	2	word (eam) ← (eam) + (A)	-	-	-	-	-	*	*	*	*	*
ADDCW	A, ear	2	1	0	word (A) ← (A) + (ear) + (C)	-	-	-	-	-	*	*	*	*	-
ADDCW	A, eam	2+	2	1	word (A) ← (A) + (eam) + (C)	-	-	-	-	-	*	*	*	*	-
SUBW	A	1	1	0	word (A) ← (AH) - (AL)	-	-	-	-	-	*	*	*	*	-
SUBW	A, ear	2	1	0	word (A) ← (A) - (ear)	-	-	-	-	-	*	*	*	*	-
SUBW	A, eam	2+	2	1	word (A) ← (A) - (eam)	-	-	-	-	-	*	*	*	*	-
SUBW	A, #imm16	3	1	0	word (A) ← (A) - imm16	-	-	-	-	-	*	*	*	*	-
SUBW	ear, A	2	1	0	word (ear) ← (ear) - (A)	-	-	-	-	-	*	*	*	*	-
SUBW	eam, A	2+	3	2	word (eam) ← (eam) - (A)	-	-	-	-	-	*	*	*	*	*
SUBCW	A, ear	2	1	0	word (A) ← (A) - (ear) - (C)	-	-	-	-	-	*	*	*	*	-
SUBCW	A, eam	2+	2	1	word (A) ← (A) - (eam) - (C)	-	-	-	-	-	*	*	*	*	-
ADDL	A, ear	2	2	0	long (A) ← (A) + (ear)	-	-	-	-	-	*	*	*	*	-
ADDL	A, eam	2+	3	1	long (A) ← (A) + (eam)	-	-	-	-	-	*	*	*	*	-
ADDL	A, #imm32	**5**	2	0	long (A) ← (A) + imm32	-	-	-	-	-	*	*	*	*	-
SUBL	A, ear	2	2	0	long (A) ← (A) - (ear)	-	-	-	-	-	*	*	*	*	-
SUBL	A, eam	2+	3	1	long (A) ← (A) - (eam)	-	-	-	-	-	*	*	*	*	-
SUBL	A, #imm32	**5**	2	0	long (A) ← (A) - imm32	-	-	-	-	-	*	*	*	*	-

Table 37-11. 増減算命令（バイト, ワード, ロングワード）：12命令

ニーモニック	オペランド	#	~	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
INC	ear	2	1	0	byte (ear) ← (ear) + 1	-	-	-	-	-	*	*	*	-	-
INC	eam	2+	3	0	byte (eam) ← (eam) + 1	-	-	-	-	-	*	*	*	-	*
DEC	ear	2	1	0	byte (ear) ← (ear) - 1	-	-	-	-	-	*	*	*	-	-
DEC	eam	2+	3	0	byte (eam) ← (eam) - 1	-	-	-	-	-	*	*	*	-	*
INCW	ear	2	1	0	word (ear) ← (ear) + 1	-	-	-	-	-	*	*	*	-	-
INCW	eam	2+	3	2	word (eam) ← (eam) + 1	-	-	-	-	-	*	*	*	-	*
DECW	ear	2	1	0	word (ear) ← (ear) - 1	-	-	-	-	-	*	*	*	-	-
DECW	eam	2+	3	2	word (eam) ← (eam) - 1	-	-	-	-	-	*	*	*	-	*
INCL	ear	2	2	0	long (ear) ← (ear) + 1	-	-	-	-	-	*	*	*	-	-
INCL	eam	2+	4	2	long (eam) ← (eam) + 1	-	-	-	-	-	*	*	*	-	*
DECL	ear	2	2	0	long (ear) ← (ear) - 1	-	-	-	-	-	*	*	*	-	-
DECL	eam	2+	4	2	long (eam) ← (eam) - 1	-	-	-	-	-	*	*	*	-	*

Table 37-12. 比較命令（バイト, ワード, ロングワード）：11命令

ニーモニック	オペランド	#	~	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
CMP	A	1	1	0	byte (AH) - (AL)	-	-	-	-	-	*	*	*	*	-
CMP	A, ear	2	1	0	byte (A) - (ear)	-	-	-	-	-	*	*	*	*	-
CMP	A, eam	2+	2	0	byte (A) - (eam)	-	-	-	-	-	*	*	*	*	-
CMP	A, #imm8	2	1	0	byte (A) - imm8	-	-	-	-	-	*	*	*	*	-
CMPW	A	1	1	0	word (AH) - (AL)	-	-	-	-	-	*	*	*	*	-
CMPW	A, ear	2	1	0	word (A) - (ear)	-	-	-	-	-	*	*	*	*	-
CMPW	A, eam	2+	2	1	word (A) - (eam)	-	-	-	-	-	*	*	*	*	-
CMPW	A, #imm16	3	1	0	word (A) - imm16	-	-	-	-	-	*	*	*	*	-
CMPL	A, ear	2	2	0	long (A) - (ear)	-	-	-	-	-	*	*	*	*	-
CMPL	A, eam	2+	3	1	long (A) - (eam)	-	-	-	-	-	*	*	*	*	-
CMPL	A, #imm32	5	2	0	long (A) - imm32	-	-	-	-	-	*	*	*	*	-

Table 37-13. 符合なし乗除算命令（ワード, ロングワード）：11命令

ニーモニック	オペランド	#	~	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
DIVU	A	1	4 / 9*1	0	word (AH) / byte (AL) quotient → byte (AL), remainder → byte (AH)	-	-	-	-	-	-	-	*	*	-
DIVU	A, ear	2	4 / 9*1	0	word (A) / byte (ear) quotient → byte (A), remainder → byte (ear)	-	-	-	-	-	-	-	*	*	-
DIVU	A, eam	2+	5 / 11*2	0	word (A) / byte (eam) quotient → byte (A), remainder → byte (eam)	-	-	-	-	-	-	-	*	*	-
DIVUW	A, ear	2	4 / 17*3	0	long (A) / word (ear) quotient → word (A), remainder → word (ear)	-	-	-	-	-	-	-	*	*	-
DIVUW	A, eam	2+	5 / 19*4	2	long (A) / word (eam) quotient → word (A), remainder → word (eam)	-	-	-	-	-	-	-	*	*	-
MULU	A	1	2	0	byte (AH) * byte (AL) → word (A)	-	-	-	-	-	-	-	-	-	-
MULU	A, ear	2	2	0	byte (A) * byte (ear) → word (A)	-	-	-	-	-	-	-	-	-	-
MULU	A, eam	2+	3	0	byte (A) * byte (eam) → word (A)	-	-	-	-	-	-	-	-	-	-
MULUW	A	1	4	0	word (AH) * word (AL) → long (A)	-	-	-	-	-	-	-	-	-	-
MULUW	A, ear	2	4	0	word (A) * word (ear) → long (A)	-	-	-	-	-	-	-	-	-	-
MULUW	A, eam	2+	5	1	word (A) * word (eam) → long (A)	-	-	-	-	-	-	-	-	-	-

*1 : オーバフローの場合は 4, それ以外は 9

*2 : オーバフローの場合は 5, それ以外は 11

*3 : オーバフローの場合は 4, それ以外は 17

*4 : オーバフローの場合は 5, それ以外は 19

Table 37-14. 符合付き乗除算命令（ワード, ロングワード）：11命令

ニーモニック	オペランド	#	~	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
DIV	A	2	5 / 11	0	word (AH) / byte (AL) quotient → byte (AL), remainder → byte (AH)	Z	-	-	-	-	-	-	*	*	-
DIV	A, ear	2	5 / 11	0	word (A) / byte (ear) quotient → byte (A), remainder → byte (ear)	Z	-	-	-	-	-	-	*	*	-
DIV	A, eam	2+	6 / 13	0	word (A) / byte (eam) quotient → byte (A), remainder → byte (eam)	Z	-	-	-	-	-	-	*	*	-
DIVW	A, ear	2	5 / 19	0	long (A) / word (ear) quotient → word (A), remainder → word (ear)	-	-	-	-	-	-	-	*	*	-
DIVW	A, eam	2+	6 / 21	2	long (A) / word (eam) quotient → word (A), remainder → word (eam)	-	-	-	-	-	-	-	*	*	-
MUL	A	2	4	0	byte (AH) * byte (AL) → word (A)	-	-	-	-	-	-	-	-	-	-
MUL	A, ear	2	4	0	byte (A) * byte (ear) → word (A)	-	-	-	-	-	-	-	-	-	-
MUL	A, eam	2+	5	0	byte (A) * byte (eam) → word (A)	-	-	-	-	-	-	-	-	-	-
MULW	A	2	6	0	word (AH) * word (AL) → long (A)	-	-	-	-	-	-	-	-	-	-
MULW	A, ear	2	6	0	word (A) * word (ear) → long (A)	-	-	-	-	-	-	-	-	-	-
MULW	A, eam	2+	7	1	word (A) * word (eam) → long (A)	-	-	-	-	-	-	-	-	-	-

*1 : オーバフローの場合は 5, それ以外は 11

*2 : オーバフローの場合は 6, それ以外は 13

*3 : オーバフローの場合は 5, それ以外は 19

*4 : オーバフローの場合は 6, それ以外は 21

Table 37-15. 論理演算命令（バイト, ワード, ロングワード）：45命令

ニーモニック	オペランド	#	~	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
AND	A, #imm8	2	1	0	byte (A) ← (A) and imm8	-	-	-	-	-	*	*	R	-	-
AND	A, ear	2	1	0	byte (A) ← (A) and (ear)	-	-	-	-	-	*	*	R	-	-
AND	A, eam	2+	2	0	byte (A) ← (A) and (eam)	-	-	-	-	-	*	*	R	-	-
AND	ear, A	2	1	0	byte (ear) ← (ear) and (A)	-	-	-	-	-	*	*	R	-	-
AND	eam, A	2+	3	0	byte (eam) ← (eam) and (A)	-	-	-	-	-	*	*	R	-	*
OR	A, #imm8	2	1	0	byte (A) ← (A) or imm8	-	-	-	-	-	*	*	R	-	-
OR	A, ear	2	1	0	byte (A) ← (A) or (ear)	-	-	-	-	-	*	*	R	-	-
OR	A, eam	2+	2	0	byte (A) ← (A) or (eam)	-	-	-	-	-	*	*	R	-	-
OR	ear, A	2	1	0	byte (ear) ← (ear) or (A)	-	-	-	-	-	*	*	R	-	-
OR	eam, A	2+	3	0	byte (eam) ← (eam) or (A)	-	-	-	-	-	*	*	R	-	*
XOR	A, #imm8	2	1	0	byte (A) ← (A) xor imm8	-	-	-	-	-	*	*	R	-	-
XOR	A, ear	2	1	0	byte (A) ← (A) xor (ear)	-	-	-	-	-	*	*	R	-	-
XOR	A, eam	2+	2	0	byte (A) ← (A) xor (eam)	-	-	-	-	-	*	*	R	-	-
XOR	ear, A	2	1	0	byte (ear) ← (ear) xor (A)	-	-	-	-	-	*	*	R	-	-
XOR	eam, A	2+	3	0	byte (eam) ← (eam) xor (A)	-	-	-	-	-	*	*	R	-	*
NOT	A	1	1	0	byte (A) ← not (A)	-	-	-	-	-	*	*	R	-	-
NOT	ear	2	1	0	byte (ear) ← not (ear)	-	-	-	-	-	*	*	R	-	-
NOT	eam	2+	3	0	byte (eam) ← not (eam)	-	-	-	-	-	*	*	R	-	*
ANDW	A	1	1	0	word (A) ← (AH) and (A)	-	-	-	-	-	*	*	R	-	-
ANDW	A, #imm16	3	1	0	word (A) ← (A) and imm16	-	-	-	-	-	*	*	R	-	-
ANDW	A, ear	2	1	0	word (A) ← (A) and (ear)	-	-	-	-	-	*	*	R	-	-
ANDW	A, eam	2+	2	1	word (A) ← (A) and (eam)	-	-	-	-	-	*	*	R	-	-
ANDW	ear, A	2	1	0	word (ear) ← (ear) and (A)	-	-	-	-	-	*	*	R	-	-
ANDW	eam, A	2+	3	2	word (eam) ← (eam) and (A)	-	-	-	-	-	*	*	R	-	*
ORW	A	1	1	0	word (A) ← (AH) or (A)	-	-	-	-	-	*	*	R	-	-
ORW	A, #imm16	3	1	0	word (A) ← (A) or imm16	-	-	-	-	-	*	*	R	-	-
ORW	A, ear	2	1	0	word (A) ← (A) or (ear)	-	-	-	-	-	*	*	R	-	-
ORW	A, eam	2+	2	1	word (A) ← (A) or (eam)	-	-	-	-	-	*	*	R	-	-
ORW	ear, A	2	1	0	word (ear) ← (ear) or (A)	-	-	-	-	-	*	*	R	-	-
ORW	eam, A	2+	3	2	word (eam) ← (eam) or (A)	-	-	-	-	-	*	*	R	-	*
XORW	A	1	1	0	word (A) ← (AH) xor (A)	-	-	-	-	-	*	*	R	-	-
XORW	A, #imm16	3	1	0	word (A) ← (A) xor imm16	-	-	-	-	-	*	*	R	-	-
XORW	A, ear	2	1	0	word (A) ← (A) xor (ear)	-	-	-	-	-	*	*	R	-	-
XORW	A, eam	2+	2	1	word (A) ← (A) xor (eam)	-	-	-	-	-	*	*	R	-	-
XORW	ear, A	2	1	0	word (ear) ← (ear) xor (A)	-	-	-	-	-	*	*	R	-	-
XORW	eam, A	2+	3	2	word (eam) ← (eam) xor (A)	-	-	-	-	-	*	*	R	-	*
NOTW	A	1	1	0	word (A) ← not (A)	-	-	-	-	-	*	*	R	-	-
NOTW	ear	2	1	0	word (ear) ← not (ear)	-	-	-	-	-	*	*	R	-	-
NOTW	eam	2+	3	2	word (eam) ← not (eam)	-	-	-	-	-	*	*	R	-	*
ANDL	A, ear	2	2	0	long (A) ← (A) and (ear)	-	-	-	-	-	*	*	R	-	-
ANDL	A, eam	2+	3	1	long (A) ← (A) and (eam)	-	-	-	-	-	*	*	R	-	-
ORL	A, ear	2	2	0	long (A) ← (A) or (ear)	-	-	-	-	-	*	*	R	-	-
ORL	A, eam	2+	3	1	long (A) ← (A) or (eam)	-	-	-	-	-	*	*	R	-	-
XORL	A, ear	2	2	0	long (A) ← (A) xor (ear)	-	-	-	-	-	*	*	R	-	-
XORL	A, eam	2+	3	1	long (A) ← (A) xor (eam)	-	-	-	-	-	*	*	R	-	-

Table 37-16. 符合反転命令（バイト, ワード）：6命令

ニーモニック	オペランド	#	～	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
NEG	A	1	1	0	byte (A) ← 0 - (A)	X	-	-	-	-	*	*	*	*	-
NEG	ear	2	1	0	byte (ear) ← 0 - (ear)	-	-	-	-	-	*	*	*	*	-
NEG	eam	2+	3	0	byte (eam) ← 0 - (eam)	-	-	-	-	-	*	*	*	*	*
NEGW	A	1	1	0	word (A) ← 0 - (A)	-	-	-	-	-	*	*	*	*	-
NEGW	ear	2	1	0	word (ear) ← 0 - (ear)	-	-	-	-	-	*	*	*	*	-
NEGW	eam	2+	3	2	word (eam) ← 0 - (eam)	-	-	-	-	-	*	*	*	*	*

Table 37-17. シフト/ノーマライズ命令（バイト, ワード, ロングワード）：19命令

ニーモニック	オペランド	#	～	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
RORC	A	2	1	0	byte (A) ← Right rotation of (A) with carry	-	-	-	-	-	*	*	-	*	-
RORC	ear	2	1	0	byte (ear) ← Right rotation of (ear) with carry	-	-	-	-	-	*	*	-	*	-
RORC	eam	2+	3	0	byte (eam) ← Right rotation of (eam) with carry	-	-	-	-	-	*	*	-	*	*
ROLC	A	2	1	0	byte (A) ← Left rotation of (A) with carry	-	-	-	-	-	*	*	-	*	-
ROLC	ear	2	1	0	byte (ear) ← Right rotation of (ear) with carry	-	-	-	-	-	*	*	-	*	-
ROLC	eam	2+	3	0	byte (eam) ← Right rotation of (eam) with carry	-	-	-	-	-	*	*	-	*	*
ASR	A, R0	2	1	0	byte (A) ← Arithmetic right barrel shift of (A) , (R0) bits	-	-	-	-	*	*	*	-	*	-
LSR	A, R0	2	1	0	byte (A) ← Logical right barrel shift of (A) , (R0) bits	-	-	-	-	*	*	*	-	*	-
LSL	A, R0	2	1	0	byte (A) ← Logical left barrel shift of (A) , (R0) bits	-	-	-	-	-	*	*	-	*	-
ASRW	A	1	1	0	word (A) ← Arithmetic right shift of (A) , 1 bit	-	-	-	-	*	*	*	-	*	-
LSRW/SHRW	A	1	1	0	word (A) ← Logical right shift of (A) , 1 bits	-	-	-	-	*	R	*	-	*	-
LSLW/SHLW	A	1	1	0	word (A) ← Logical left shift of (A) , 1 bits	-	-	-	-	-	*	*	-	*	-
ASRW	A, R0	2	1	0	word (A) ← Arithmetic right barrel shift of (A) , (R0) bits	-	-	-	-	*	*	*	-	*	-
LSRW	A, R0	2	1	0	word (A) ← Logical right barrel shift of (A) , (R0) bits	-	-	-	-	*	*	*	-	*	-
LSLW	A, R0	2	1	0	word (A) ← Logical left barrel shift of (A) , (R0) bits	-	-	-	-	-	*	*	-	*	-
ASRL	A, R0	2	1	0	long (A) ← Arithmetic right barrel shift of (A) , (R0) bits	-	-	-	-	*	*	*	-	*	-
LSRL	A, R0	2	1	0	long (A) ← Logical right barrel shift of (A) , (R0) bits	-	-	-	-	*	*	*	-	*	-
LSLL	A, R0	2	1	0	long (A) ← Logical left barrel shift of (A) , (R0) bits	-	-	-	-	-	*	*	-	*	-
NRML	A, R0	2	1	0	long (A) ← Shift left (A) until the MSB is "1" byte (R0) ← Shift count (the first "1" bit position)	-	-	-	-	-	-	*	-	-	-

Table 37-18. 分岐命令1：31命令

ニーモニック	オペランド	#	~	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
BZ/BEQ	rel	2	2	0	Branch on (Z) = 1	-	-	-	-	-	-	-	-	-	-
BNZ/BNE	rel	2	2	0	Branch on (Z) = 0	-	-	-	-	-	-	-	-	-	-
BC/BLO	rel	2	2	0	Branch on (C) = 1	-	-	-	-	-	-	-	-	-	-
BNC/BHS	rel	2	2	0	Branch on (C) = 0	-	-	-	-	-	-	-	-	-	-
BN	rel	2	2	0	Branch on (N) = 1	-	-	-	-	-	-	-	-	-	-
BP	rel	2	2	0	Branch on (N) = 0	-	-	-	-	-	-	-	-	-	-
BV	rel	2	2	0	Branch on (V) = 1	-	-	-	-	-	-	-	-	-	-
BNV	rel	2	2	0	Branch on (V) = 0	-	-	-	-	-	-	-	-	-	-
BT	rel	2	2	0	Branch on (T) = 1	-	-	-	-	-	-	-	-	-	-
BNT	rel	2	2	0	Branch on (T) = 0	-	-	-	-	-	-	-	-	-	-
BLT	rel	2	2	0	Branch on (V) xor (N) = 1	-	-	-	-	-	-	-	-	-	-
BGE	rel	2	2	0	Branch on (V) xor (N) = 0	-	-	-	-	-	-	-	-	-	-
BLE	rel	2	2	0	Branch on ((V) xor (N)) or (Z) = 1	-	-	-	-	-	-	-	-	-	-
BGT	rel	2	2	0	Branch on ((V) xor (N)) or (Z) = 0	-	-	-	-	-	-	-	-	-	-
BLS	rel	2	2	0	Branch on (C) or (Z) = 1	-	-	-	-	-	-	-	-	-	-
BHI	rel	2	2	0	Branch on (C) or (Z) = 0	-	-	-	-	-	-	-	-	-	-
BRA	rel	2	2	0	Branch always (Unconditional branch)	-	-	-	-	-	-	-	-	-	-
JMP	@A	1	2	0	word (PC) ← (A)	-	-	-	-	-	-	-	-	-	-
JMP	addr16	3	2	0	word (PC) ← addr16	-	-	-	-	-	-	-	-	-	-
JMP	@ear	2	2	0	word (PC) ← (ear)	-	-	-	-	-	-	-	-	-	-
JMP	@eam	2+	4	1	word (PC) ← (eam)	-	-	-	-	-	-	-	-	-	-
JMPP	@ear	2	3	0	word (PC) ← (ear), (PCB) ← (ear+2)	-	-	-	-	-	-	-	-	-	-
JMPP	@eam	2+	5	1	word (PC) ← (eam), (PCB) ← (eam+2)	-	-	-	-	-	-	-	-	-	-
JMPP	addr24	4	2	0	word (PC) ← ad24 0-15, (PCB) ← ad24 16-23	-	-	-	-	-	-	-	-	-	-
CALL	@ear	2	3	1	word (SP) ← (SP)-2, ((SP)) ← (PC)+2 word (PC) ← (ear)	-	-	-	-	-	-	-	-	-	-
CALL	@eam	2+	5	1+1*a	word (SP) ← (SP)-2, ((SP)) ← (PC)+ 2+ word (PC) ← (eam)	-	-	-	-	-	-	-	-	-	-
CALL	addr16	3	3	1	word (SP) ← (SP)-2, ((SP)) ← (PC)+3 word (PC) ← addr16	-	-	-	-	-	-	-	-	-	-
CALLV	#vct4	1	5	1	word (SP) ← (SP)-2, ((SP)) ← (PC)+1 word (PC) ← (vecter_address)	-	-	-	-	-	-	-	-	-	-
CALLP	@ear	2	5	2	word (SP) ← (SP)-2, ((SP)) ← (PCB) word (SP) ← (SP)-2, ((SP)) ← (PC)+2 word (PC) ← (ear), (PCB) ← (ear+2)	-	-	-	-	-	-	-	-	-	-
CALLP	@eam	2+	7	2+1*b	word (SP) ← (SP)-2, ((SP)) ← (PCB) word (SP) ← (SP)-2, ((SP)) ← (PC)+ 2+ word (PC) ← (eam), (PCB) ← (eam+2)	-	-	-	-	-	-	-	-	-	-
CALLP	addr24	4	4	2	word (SP) ← (SP)-2, ((SP)) ← (PCB) word (SP) ← (SP)-2, ((SP)) ← (PC)+4 word (PC) ← ad24 0-15, (PCB) ← ad24 16-23	-	-	-	-	-	-	-	-	-	-

*a : 補正値：奇数スタックには +1, 奇数オペランドアドレスには +1
*b : 補正値：奇数スタックには +2, 奇数オペランドアドレスには +1

Table 37-19. 分岐命令2：19命令

ニーモニック	オペランド	#	～	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
CBNE	A, #imm8, rel	3	5	0	Branch on byte (A) not equal to imm8	-	-	-	-	-	*	*	*	*	-
CWBNE	A, #imm16, rel	4	5	0	Branch on word (A) not equal to imm16	-	-	-	-	-	*	*	*	*	-
CBNE	ear, #imm8, rel	4	4	0	Branch on byte (ear) not equal to imm8	-	-	-	-	-	*	*	*	*	-
CBNE	eam, #imm8, rel	4+	5	0	Branch on byte (eam) not equal to imm8	-	-	-	-	-	*	*	*	*	-
CWBNE	ear, #imm16, rel	5	5	0	Branch on word (ear) not equal to imm16	-	-	-	-	-	*	*	*	*	-
CWBNE	eam, #imm16, rel	5+	6	1	Branch on word (eam) not equal to imm16	-	-	-	-	-	*	*	*	*	-
DBNZ	ear, rel	3	5	0	byte (ear) ← (ear) - 1, Branch on (ear) not equal to 0	-	-	-	-	-	*	*	*	-	-
DBNZ	eam, rel	3+	6	0	byte (eam) ← (eam) - 1, Branch on (eam) not equal to 0	-	-	-	-	-	*	*	*	-	*
DWBNZ	ear, rel	3	5	0	word (ear) ← (ear) - 1, Branch on (ear) not equal to 0	-	-	-	-	-	*	*	*	-	-
DWBNZ	eam, rel	3+	6	2	word (eam) ← (eam) - 1, Branch on (eam) not equal to 0	-	-	-	-	-	*	*	*	-	*
INT	#vct8	2	11	6	Software interrupt	-	-	R	S	-	-	-	-	-	-
INT	addr16	3	8	6	Software interrupt	-	-	R	S	-	-	-	-	-	-
INTP	addr24	4	8	6	Software interrupt	-	-	R	S	-	-	-	-	-	-
INT9		1	11	6	Software interrupt	-	-	R	S	-	-	-	-	-	-
INTE		1	12	6	Software interrupt for break point (reserved for emulator)	-	-	R	S	-	-	-	-	-	-
RETI		1	22 / 6[*1]	1	Return from interrupt	-	-	*	*	*	*	*	*	*	-
LINK	#imm8	2	2	1	word (SP) ← (SP) - 2, ((SP)) ← (RW3), word (RW3) ← (SP), (SP) ← (SP) - imm8	-	-	-	-	-	-	-	-	-	-
UNLINK		1	1	1	word (SP) ← (RW3), (RW3) ← ((SP)), word (SP) ← (SP) + 2	-	-	-	-	-	-	-	-	-	-
RET		1	4	1	word (PC) ← ((SP)), (SP) ← (SP) + 2	-	-	-	-	-	-	-	-	-	-
RETP		1	5	1	word (PC) ← ((SP)), (SP) ← (SP) + 2, byte (PCB) ← ((SP)), (SP) ← (SP) + 2	-	-	-	-	-	-	-	-	-	-

*1 : RP が安定している場合は 6, それ以外は 22

Table 37-20. その他の制御命令（バイト, ワード, ロングワード）：28命令

ニーモニック	オペランド	#	~	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
PUSHW	A	1	1	1	word (SP) ← (SP) - 2, ((SP)) ← (A)	-	-	-	-	-	-	-	-	-	-
PUSHW	AH	1	1	1	word (SP) ← (SP) - 2, ((SP)) ← (AH)	-	-	-	-	-	-	-	-	-	-
PUSHW	PS	1	1	1	word (SP) ← (SP) - 2, ((SP)) ← (PS)	-	-	-	-	-	-	-	-	-	-
PUSHW	rlst	2	N*1	N*a	multi word (SP) ← (SP) - 2n, ((SP)) ← (rlst)	-	-	-	-	-	-	-	-	-	-
POPW	A	1	1	1	word (A) ← ((SP)), (SP) ← (SP) + 2	-	*	-	-	-	-	-	-	-	-
POPW	AH	1	1	1	word (AH) ← ((SP)), (SP) ← (SP) + 2	-	-	-	-	-	-	-	-	-	-
POPW	PS	1	18 / 4*2	1	word (PS) ← ((SP)), (SP) ← (SP) + 2	-	-	*	*	*	*	*	*	*	-
POPW	rlst	2	N*1	N*a	multi word (rlst) ← ((SP)), (SP) ← (SP) + 2n	-	-	-	-	-	-	-	-	-	-
JCTX	@A	1	22 / 6*3	1	Context switch	-	-	*	*	*	*	*	*	*	-
AND	CCR, #imm8	2	1	0	byte (CCR) ← (CCR) and imm8	-	-	*	*	*	*	*	*	*	-
OR	CCR, #imm8	2	1	0	byte (CCR) ← (CCR) or imm8	-	-	*	*	*	*	*	*	*	-
MOV	RP, #imm8	2	19 / 4*2	0	byte (RP) ← imm8	-	-	-	-	-	-	-	-	-	-
MOV	ILM, #imm8	2	1	0	byte (ILM) ← imm8	-	-	-	-	-	-	-	-	-	-
MOVEA	RWi, ear	2	1	0	word (RWi) ← ear	-	-	-	-	-	-	-	-	-	-
MOVEA	RWi, eam	2+	1	0	word (RWi) ← eam	-	-	-	-	-	-	-	-	-	-
MOVEA	A, ear	2	1	0	word (A) ← ear	-	*	-	-	-	-	-	-	-	-
MOVEA	A, eam	2+	1	0	word (A) ← eam	-	*	-	-	-	-	-	-	-	-
ADDSP	#imm8	2	1	0	word (SP) ← (SP) + ext(imm8)	-	-	-	-	-	-	-	-	-	-
ADDSP	#imm16	3	1	0	word (SP) ← (SP) + imm16	-	-	-	-	-	-	-	-	-	-
MOV	A, brg1	2	1	0	byte (A) ← (brg1)	Z	*	-	-	-	*	*	-	-	-
MOV	brg2, A	2	1	0	byte (brg2) ← (A)	-	-	-	-	-	*	*	-	-	-
NOP		1	1	0	No operation	-	-	-	-	-	-	-	-	-	-
ADB		1	1	0	Prefix code for AD space access	-	-	-	-	-	-	-	-	-	-
DTB		1	1	0	Prefix code for DT space access	-	-	-	-	-	-	-	-	-	-
PCB		1	1	0	Prefix code for PC space access	-	-	-	-	-	-	-	-	-	-
SPB		1	1	0	Prefix code for SP space access	-	-	-	-	-	-	-	-	-	-
NCC		1	1	0	Prefix code for flags no change	-	-	-	-	-	-	-	-	-	-
CMR		1	1	0	Prefix code for common register bank	-	-	-	-	-	-	-	-	-	-

*1 : N は保存 / 復帰するレジスタの数による。最小は 1

*2 : RP が安定している場合は 4, それ以外は 18

*3 : RP が安定している場合は 6, それ以外は 22

*a : N は保存するレジスタの数による

Table 37-21. ビット操作命令：21命令

ニーモニック	オペランド	#	~	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
MOVB	A, dir:bp	3	1	0	byte (A) ← (dir:bp) b	Z	*	-	-	-	*	*	-	-	-
MOVB	A, addr16:bp	4	1	0	byte (A) ← (addr16:bp) b	Z	*	-	-	-	*	*	-	-	-
MOVB	A, io:bp	3	1	0	byte (A) ← (io:bp) b	Z	*	-	-	-	*	*	-	-	-
MOVB	dir:bp, A	3	3	0	bit (dir:bp) b ← (A)	-	-	-	-	-	*	*	-	-	*
MOVB	addr16:bp	4	3	0	bit (addr16:bp) b ← (A)	-	-	-	-	-	*	*	-	-	*
MOVB	io:bp, A	3	3	0	bit (io:bp) b ← (A)	-	-	-	-	-	*	*	-	-	*
SETB	dir:bp	3	3	0	bit (dir:bp) b ← 1	-	-	-	-	-	-	-	-	-	*
SETB	addr16:bp	4	3	0	bit (addr16:bp) b ← 1	-	-	-	-	-	-	-	-	-	*
SETB	io:bp	3	3	0	bit (io:bp) b ← 1	-	-	-	-	-	-	-	-	-	*
CLRB	dir:bp	3	3	0	bit (dir:bp) b ← 0	-	-	-	-	-	-	-	-	-	*
CLRB	addr16:bp	4	3	0	bit (addr16:bp) b ← 0	-	-	-	-	-	-	-	-	-	*
CLRB	io:bp	3	3	0	bit (io:bp) b ← 0	-	-	-	-	-	-	-	-	-	*
BBC	dir:bp, rel	4	5	0	Branch on (dir:bp) b = 0	-	-	-	-	-	-	*	-	-	-
BBC	addr16:bp, rel	5	5	0	Branch on (addr16:bp) b = 0	-	-	-	-	-	-	*	-	-	-
BBC	io:bp, rel	4	5	0	Branch on (io:bp) b = 0	-	-	-	-	-	-	*	-	-	-
BBS	dir:bp, rel	4	5	0	Branch on (dir:bp) b = 1	-	-	-	-	-	-	*	-	-	-
BBS	addr16:bp, rel	5	5	0	Branch on (addr16:bp) b = 1	-	-	-	-	-	-	*	-	-	-
BBS	io:bp, rel	4	5	0	Branch on (io:bp) b = 1	-	-	-	-	-	-	*	-	-	-
SBBS	addr16:bp, rel	5	5	0	Branch on (addr16:bp) b = 1, bit (addr16:bp) b ← 1	-	-	-	-	-	-	*	-	-	*
WBTS	io:bp	3	undefined*1	0	Wait until (io:bp) b = 1	-	-	-	-	-	-	-	-	-	-
WBTC	io:bp	3	undefined*2	0	Wait until (io:bp) b = 0	-	-	-	-	-	-	-	-	-	-

*1 : ビットが既に設定されていれば 4, それ以外は不定

*2 : ビットが既にクリアされていれば 4, それ以外は不定

Table 37-22. アキュムレータ操作命令（バイト, ワード）：6命令

ニーモニック	オペランド	#	~	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
SWAP		1	1	0	byte (A) 0-7 ←→ (A) 8-15	-	-	-	-	-	-	-	-	-	-
SWAPW		1	1	0	word (AH) ←→ (AL)	-	*	-	-	-	-	-	-	-	-
EXT		1	1	0	Byte sign extension	X	-	-	-	-	*	*	-	-	-
EXTW		1	1	0	Word sign extension	-	X	-	-	-	*	*	-	-	-
ZEXT		1	1	0	Byte zero extension	Z	-	-	-	-	R	*	-	-	-
ZEXTW		1	1	0	Word zero extension	-	Z	-	-	-	R	*	-	-	-

Table 37-23. ストリング命令：10命令

ニーモニック	オペランド	#	~	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
MOVS/MOVSI	brg3, brg3	2	RW0[*2]	1	byte transfer @AH+ ← @AL+, (RW0) times	-	-	-	-	-	-	-	-	-	-
MOVSD	brg3, brg3	2	2*RW0	0	byte transfer @AH- ← @AL-, (RW0) times	-	-	-	-	-	-	-	-	-	-
SCEQ/SCEQI	brg3	2	2+2*N[*3]	0	byte search @AH+ = AL, (RW0) times or till match	-	-	-	-	-	*	*	*	*	-
SCEQD	brg3	2	2+2*N[*3]	0	byte search @AH- = AL, (RW0) times or till match	-	-	-	-	-	*	*	*	*	-
FILS/FILSI	brg3	2	RW0/2[*4]	1	byte fill @AH+ ← AL, (RW0) times	-	-	-	-	-	*	*	-	-	-
MOVSW/MOVSWI	brg3, brg3	2	2*RW0	1	word transfer @AH+ ← @AL+, (RW0) times	-	-	-	-	-	-	-	-	-	-
MOVSWD	brg3, brg3	2	2*RW0	(1+1)*RW0[*a]	word transfer @AH- ← @AL-, (RW0) times	-	-	-	-	-	-	-	-	-	-
SCWEQ/SCWEQI	brg3	2	2+2*N[*3]	N[*b]	word search @AH+ = AL, (RW0) times or till match	-	-	-	-	-	*	*	*	*	-
SCWEQD	brg3	2	2+2*N[*3]	N[*b]	word search @AH- = AL, (RW0) times or till match	-	-	-	-	-	*	*	*	*	-
FILSW/FILSWI	brg3	2	RW0	1	word fill @AH+ ← AL, (RW0) times	-	-	-	-	-	*	*	-	-	-

*1 : RW0=0 であれば , 全文字列操作には 1 サイクルが必要

*2 : 範囲が重複していれば 2*RW0, 最適化は行わない

*3 : RW0=0 であれば 1, RW0=1 であれば 3, N は一致するまで比較された項目の数

*4 : バイト数が奇数の場合 , サイクル数は端数切り上げ

*a : src と dest アドレスの両方が奇数であれば , 補正は 2*RW0。src または dest のみが奇数であれば , 1*RW0

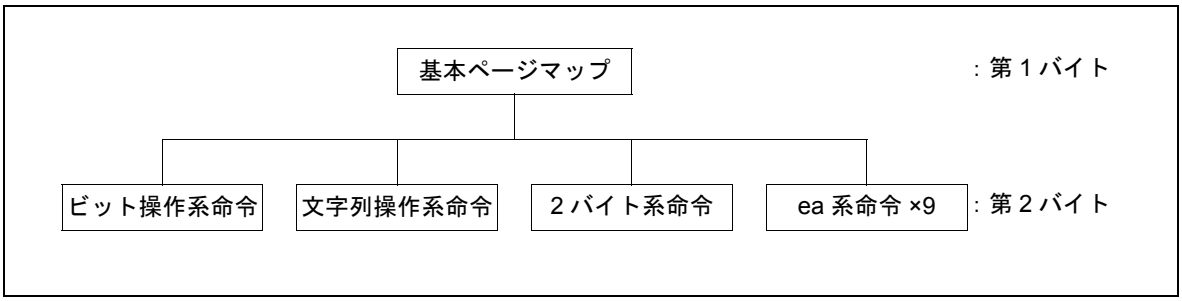
*b : N は一致するまで比較された項目の数

37.2.9 命令マップ

F²MC-16FX の各命令コードが, 1 ～ 2 バイトで構成されているため,命令マップも,複数ページで構成されています。F²MC-16FXの命令マップを示します。

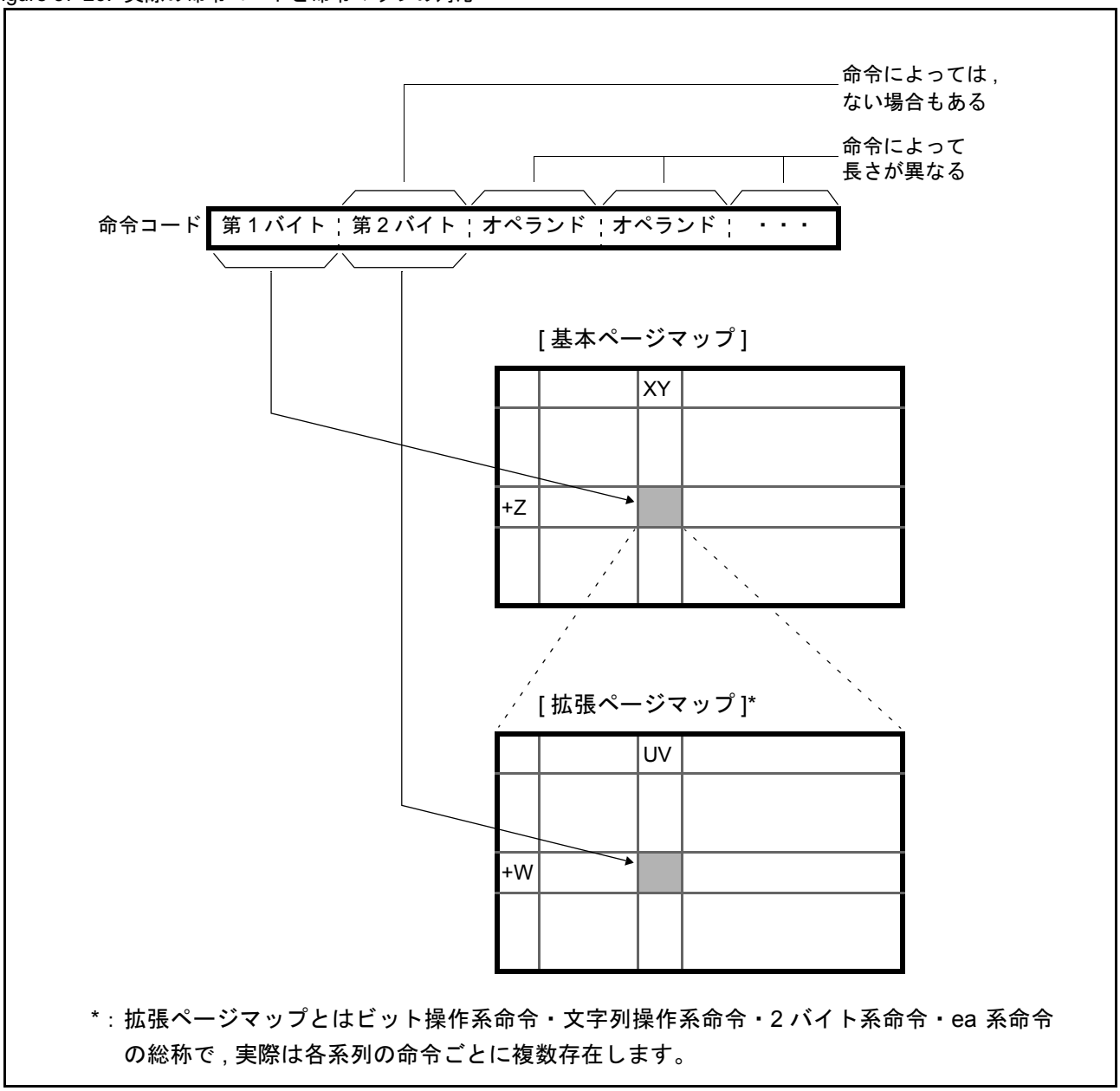
命令マップの構造

Figure 37-25. 命令マップの構造



NOP 命令などの 1 バイトで終わる命令は , 基本ページに記述されています。MOVS 命令などの 2 バイトを必要とする命令は , 第 1 バイトを参照するとき第 2 バイトを認識し , 第 2 バイトのマップを参照することにより , 次の 1 バイトを確認することができます。Figure 37-26 に , 実際の命令コードと命令マップの対応を示します。

Figure 37-26. 実際の命令コードと命令マップの対応



命令コードの例は Table 37-24 に示されます。

Table 37-24. 命令コードの例

命令	バイト 1 (基本ページマップより)	バイト 2 (拡張ページマップより)
NOP	00 +0=00	-
AND A, #8	30 +4=34	-
MOV A, ADB	60 +F=6F	00 +0=00
@RW2+d8, #8rel	70 +0=70	F0 +2=F2

Table 37-25. 基本ページマップ

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	NOP	CMR	ADD A, dir	ADD A, #8	MOV A, dir	MOV A, io	BRA rel	ea instructions (1)	MOV A, Ri	MOV Ri, A	MOV Ri, #8	MOVX A, Ri	MOVX A,@RWi+d8	MOVN A, #4	CALLV #4	BZ/BEQ rel
+1	INT9	NCC	SUB A, dir	SUB A, #8	MOV dir, A	MOV io, A	JMP @A	ea instructions (2)								BNZ/BNE rel
+2	ADDDC A	SUBDC A	ADDC A	SUBC A	MOV A, #8	MOV A, addr16	JMP addr16	ea instructions (3)								BC /BLO rel
+3	NEG A	JCTX @A	CMP A	CMP A, #8	MOVX A, #8	MOV addr 16, A	JMPP addr24	ea instructions (4)								BNC/BHS rel
+4	PCB	EXT	AND CCR, #8	AND A, #8	MOV dir, #8	MOV io, #8	CALL addr16	ea instructions (5)								BN rel
+5	DTB	ZEXT	OR CCR, #8	OR A, #8	MOVX A, dir	MOVX A, io	CALLP addr24	ea instructions (6)								BP rel
+6	ADB	SWAP	DIVU A	XOR A, #8	MOVW A, SP	MOVW io, #16	RETP	ea instructions (7)								BV rel
+7	SPB	ADDSP #8	MULU A	NOT A	MOVW SP, A	MOVX A, addr16	RET	ea instructions (8)	MOVW A, RWi	MOVW RWi, A	MOVW RWi, #16	MOVW A, @RWi + dB	MOVW @R W j + dB, A	MOVN A, #4	CALLV #4	BNV rel
+8	LINK imm#8	ADDL A, #32	ADDW A	ADDW A, #16	MOVW A, dir	MOVW A, io	INT #vct8	ea instructions (9)								BT rel
+9	UNLINK	SUBL A, #32	SUBW A	SUBW A, #16	MOVW dir, A	MOVW io, A	INT addr16	MOVEA RWi, ea								BNT rel
+A	MOV RP, #8	MOV ILM, #8	CBNE A, #8, rel	CWBNE A, #16, rel	MOVW A, #16	MOVW A, addr16	INTP addr24	MOV Ri, ea								BLT rel
+B	NEGW A	CMPL A, #32	CMPW A	CMPW A, #16	MOVL A, #32	MOVW addr16, A	RETI	MOVW RWi, ea								BGE rel
+C	LSLW A	EXTW	ANDW A	ANDW A, #16	PUSHW A	POPW A	Bit operation instructions	MOV ea, Ri								BLE rel
+D		ZEXTW	ORW A	ORW A, #16	PUSHW AH	POPW AH		MOVW ea, RWi								BGT rel
+E	ASRW A	SWAPW	XORW A	XORW A, #16	PUSHW PS	POPW PS	String operation instructions	XCH Ri, ea								BLS rel
+F	LSRW A	ADDSP #16	MULUW A	NOTW A	PUSHW rlst	POPW rlst	Two-byte instructions	XCRW RWi, ea								BHI rel

Table 37-26. ビット操作系命令マップ(第1バイト = $6C_H$)

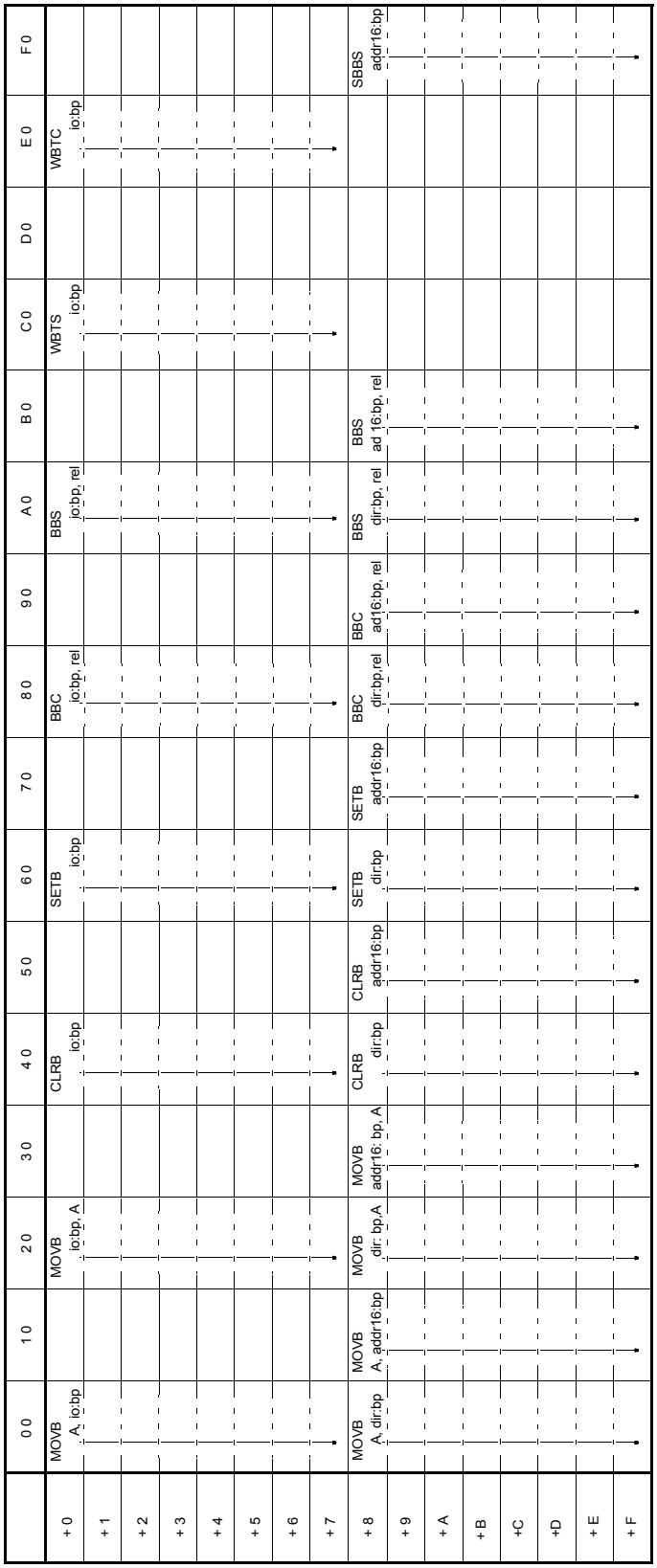


Table 37-27. 文字列操作系命令マップ(第1バイト = $6E_H$)

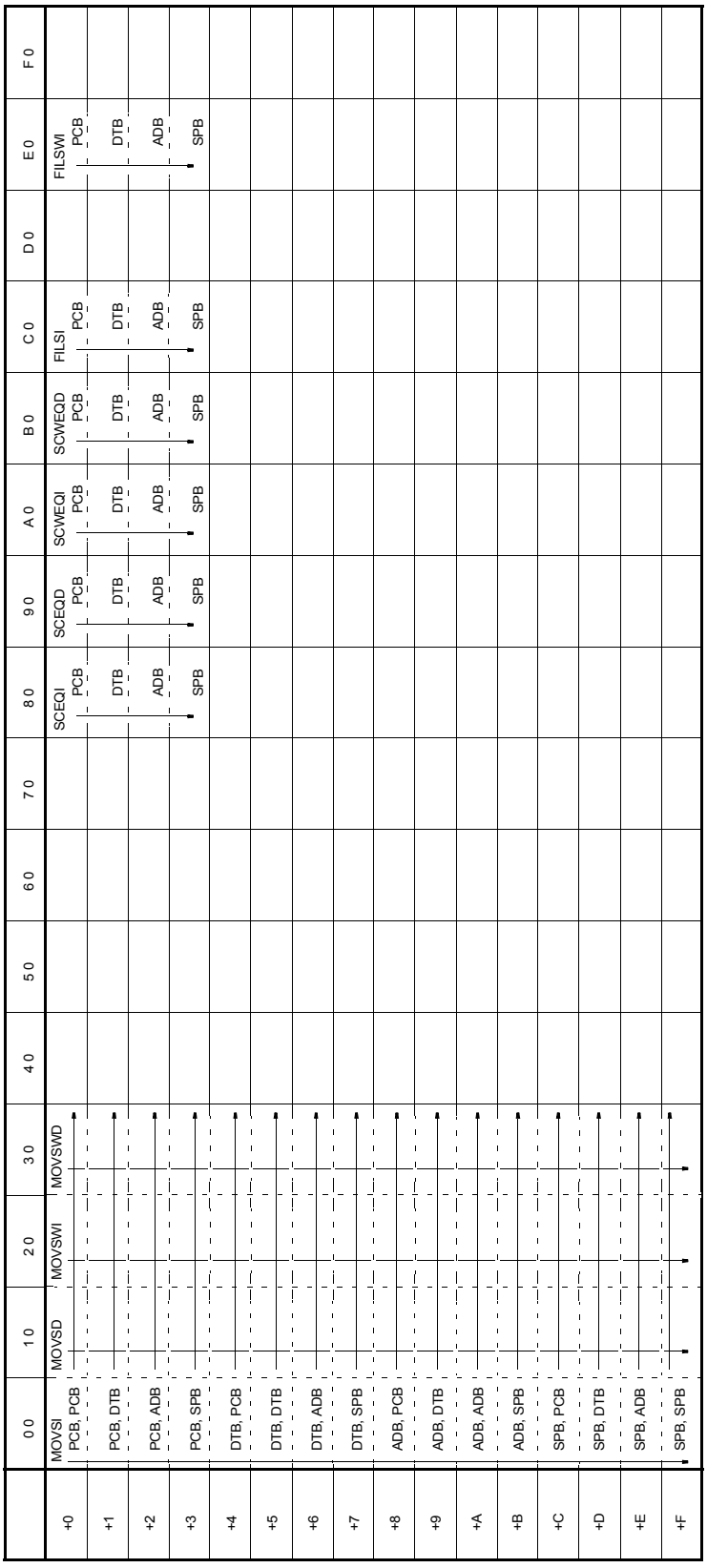


Table 37-28. 2バイト系命令マップ (第1バイト = $6F_H$)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	MOV A, DTB	MOV DTB, A	MOVX A, @RL0 + d8	MOV @RL0 + d8, A	MOV A, @RL0 + d8											
+1	MOV A, ADB	MOV ADB, A														
+2	MOV A, SSB	MOV SSB, A	MOVX A, @RL1+ d8	MOV @RL1 + d8, A	MOV A, @RL1 + d8											
+3	MOV A, USB	MOV USB, A														
+4	MOV A, DPR	MOV DPR, A	MOVX A, @RL2 + d8	MOV @RL2 + d8, A	MOV A, @RL2 + d8											
+5	MOV A, @A	MOV @AL, AH														
+6	MOV A, PCB	MOVX A, @A	MOVX A, @RL3 + d8	MOV @RL3 + d8, A	MOV A, @RL3 + d8											
+7	ROLC A	RORC A														
+8				MOVW @RL0 + d8, A	MOVW A, @RL0 + d8			MUL A								
+9								MULW A								
+A				MOVW @RL1 + d8, A	MOVW A, @RL1 + d8			DIV A								
+B																
+C	LSLW A, R0	LSLL A, R0	LSL A, R0	MOVW @RL2 + d8, A	MOVW A, @RL2 + d8											
+D	MOVW A, @A	MOVW @AL, AH	NRML A, R0													
+E	ASRW A, R0	ASRL A, R0	ASR A, R0	MOVW @RL3 + d8, A	MOVW A, @RL3 + d8											
+F	LSRW A, R0	LSRL A, R0	LSR A, R0													

Table 37-29. ea系命令1 (第1バイト = 70_H)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
					CWBNE ↓	CWBNE ↓									CBNE ↓	CBNE ↓
+0	ADDL A, RL0	ADDL A, @RW0 + d8	SUBL A, RL0	SUBLA, @RW0 + d8	RW0, #16, rel	@RW0 + d8 #16, rel	CMPL A, RL0	CMPLA, @RW0 + d8	ANDL A, RL0	ANDLA, @RW0 + d8	ORL A, RL0	ORLA, @RW0 + d8	XORL A, RL0	XORLA, @RW0 + d8	R0, #8, rel	@RW0 + d8 #8, rel
+1	ADDL A, RL0	ADDLA, @RW1 + d8	SUBL A, RL0	SUBLA, @RW1 + d8	RW1, #16, rel	@RW1 + d8 #16, rel	CMPL A, RL0	CMPLA, @RW1 + d8	ANDL A, RL0	ANDLA, @RW1 + d8	ORL A, RL0	ORLA, @RW1 + d8	XORL A, RL0	XORLA, @RW1 + d8	R1, #8, rel	@RW1 + d8 #8, rel
+2	ADDL A, RL1	ADDLA, @RW2 + d8	SUBL A, RL1	SUBLA, @RW2 + d8	RW2, #16, rel	@RW2 + d8 #16, rel	CMPL A, RL1	CMPLA, @RW2 + d8	ANDL A, RL1	ANDLA, @RW2 + d8	ORL A, RL1	ORLA, @RW2 + d8	XORL A, RL1	XORLA, @RW2 + d8	R2, #8, rel	@RW2 + d8 #8, rel
+3	ADDL A, RL1	ADDLA, @RW3 + d8	SUBL A, RL1	SUBLA, @RW3 + d8	RW3, #16, rel	@RW3 + d8 #16, rel	CMPL A, RL1	CMPLA, @RW3 + d8	ANDL A, RL1	ANDLA, @RW3 + d8	ORL A, RL1	ORLA, @RW3 + d8	XORL A, RL1	XORLA, @RW3 + d8	R3, #8, rel	@RW3 + d8 #8, rel
+4	ADDL A, RL2	ADDLA, @RW4 + d8	SUBL A, RL2	SUBLA, @RW4 + d8	RW4, #16, rel	@RW4 + d8 #16, rel	CMPL A, RL2	CMPLA, @RW4 + d8	ANDL A, RL2	ANDLA, @RW4 + d8	ORL A, RL2	ORLA, @RW4 + d8	XORL A, RL2	XORLA, @RW4 + d8	R4, #8, rel	@RW4 + d8 #8, rel
+5	ADDL A, RL2	ADDLA, @RW5 + d8	SUBL A, RL2	SUBLA, @RW5 + d8	RW5, #16, rel	@RW5 + d8 #16, rel	CMPL A, RL2	CMPLA, @RW5 + d8	ANDL A, RL2	ANDLA, @RW5 + d8	ORL A, RL2	ORLA, @RW5 + d8	XORL A, RL2	XORLA, @RW5 + d8	R5, #8, rel	@RW5 + d8 #8, rel
+6	ADDL A, RL3	ADDLA, @RW6 + d8	SUBL A, RL3	SUBLA, @RW6 + d8	RW6, #16, rel	@RW6 + d8 #16, rel	CMPL A, RL3	CMPLA, @RW6 + d8	ANDL A, RL3	ANDLA, @RW6 + d8	ORL A, RL3	ORLA, @RW6 + d8	XORL A, RL3	XORLA, @RW6 + d8	R6, #8, rel	@RW6 + d8 #8, rel
+7	ADDL A, RL3	ADDLA, @RW7 + d8	SUBL A, RL3	SUBLA, @RW7 + d8	RW7, #16, rel	@RW7 + d8 #16, rel	CMPL A, RL3	CMPLA, @RW7 + d8	ANDL A, RL3	ANDLA, @RW7 + d8	ORL A, RL3	ORLA, @RW7 + d8	XORL A, RL3	XORLA, @RW7 + d8	R7, #8, rel	@RW7 + d8 #8, rel
+8	ADDL A, @RW0	ADDL A, @RW0 + d16	SUBL A, @RW0	SUBLA, @RW0 + d16	@RW0, #16, rel	@RW0 + d16 #16, rel	CMPL A, @RW0	CMPLA, @RW0 + d16	ANDL A, @RW0	ANDLA, @RW0 + d16	ORL A, @RW0	ORLA, @RW0 + d16	XORL A, @RW0	XORLA, @RW0 + d16	@RW0, #8, rel	@RW0 + d16 #8, rel
+9	ADDL A, @RW1	ADDLA, @RW1 + d16	SUBL A, @RW1	SUBLA, @RW1 + d16	@RW1, #16, rel	@RW1 + d16 #16, rel	CMPL A, @RW1	CMPLA, @RW1 + d16	ANDL A, @RW1	ANDLA, @RW1 + d16	ORL A, @RW1	ORLA, @RW1 + d16	XORL A, @RW1	XORLA, @RW1 + d16	@RW1, #8, rel	@RW1 + d16 #8, rel
+A	ADDL A, @RW2	ADDLA, @RW2 + d16	SUBL A, @RW2	SUBLA, @RW2 + d16	@RW2, #16, rel	@RW2 + d16 #16, rel	CMPL A, @RW2	CMPLA, @RW2 + d16	ANDL A, @RW2	ANDLA, @RW2 + d16	ORL A, @RW2	ORLA, @RW2 + d16	XORL A, @RW2	XORLA, @RW2 + d16	@RW2, #8, rel	@RW2 + d16 #8, rel
+B	ADDL A, @RW3	ADDLA, @RW3 + d16	SUBL A, @RW3	SUBLA, @RW3 + d16	@RW3, #16, rel	@RW3 + d16 #16, rel	CMPL A, @RW3	CMPLA, @RW3 + d16	ANDL A, @RW3	ANDLA, @RW3 + d16	ORL A, @RW3	ORLA, @RW3 + d16	XORL A, @RW3	XORLA, @RW3 + d16	@RW3, #8, rel	@RW3 + d16 #8, rel
+C	ADDL A, @RW0+	ADDLA, @RW0 + RW7	SUBL A, @RW0+	SUBLA, @RW0 + RW7	Do not use	@RW0 + RW7 #16, rel	CMPL A, @RW0+	CMPLA, @RW0 + RW7	ANDL A, @RW0+	ANDLA, @RW0 + RW7	ORL A, @RW0+	ORLA, @RW0 + RW7	XORL A, @RW0+	XORLA, @RW0 + RW7	Do not use	@RW0 + RW7 #8, rel
+D	ADDL A, @RW1+	ADDLA, @RW1 + RW7	SUBL A, @RW1+	SUBLA, @RW1 + RW7	Do not use	@RW1 + RW7 #16, rel	CMPL A, @RW1+	CMPLA, @RW1 + RW7	ANDL A, @RW1+	ANDLA, @RW1 + RW7	ORL A, @RW1+	ORLA, @RW1 + RW7	XORL A, @RW1+	XORLA, @RW1 + RW7	Do not use	@RW1 + RW7 #8, rel
+E	ADDL A, @RW2+	ADDLA, @PC + d16	SUBL A, @RW2+	SUBLA, @PC + d16	Do not use	@PC + d16, #16, rel	CMPL A, @RW2+	CMPLA, @PC + d16	ANDL A, @RW2+	ANDLA, @PC + d16	ORL A, @RW2+	ORLA, @PC + d16	XORL A, @RW2+	XORLA, @PC + d16	Do not use	@PC + d16 #8, rel
+F	ADDL A, @RW3+	ADDLA, addr16	SUBL A, @RW3+	SUBLA, addr16	Do not use	addr16, #16, rel	CMPL A, @RW3+	CMPLA, addr16	ANDL A, @RW3+	ANDLA, addr16	ORL A, @RW3+	ORLA, addr16	XORL A, @RW3+	XORLA, addr16	Do not use	addr16 #8, rel

Table 37-30. ea系命令2 (第1バイト = 71_H)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	JMPP @RL0	JMPP @@RW0 + d8	CALLP @RL0	CALLP @@RW0 + d8	INCL RL0	INCL @RW0 + d8	DECL RL0	DECL @RW0 + d8	MOVL A, RL0	MOVL A, @RW0 + d8	MOVL RL0, A	MOVL @R W0 + d8,A	MOV R0,#8	MOV @R W0 + d8,#8	MOVEA A,RW0	MOVEA A, @RW0 + d8
+1	JMPP @RL0	JMPP @@RW1 + d8	CALLP @RL0	CALLP @@RW1 + d8	INCL RL0	INCL @RW1 + d8	DECL RL0	DECL @RW1 + d8	MOVL A, RL0	MOVL A, @RW1 + d8	MOVL RL0, A	MOVL @R W1 + d8,A	MOV R1,#8	MOV @R W1 + d8,#8	MOVEA A,RW1	MOVEA A, @RW1 + d8
+2	JMPP @RL1	JMPP @@RW2 + d8	CALLP @RL1	CALLP @@RW2 + d8	INCL RL1	INCL @RW2 + d8	DECL RL1	DECL @RW2 + d8	MOVL A, RL1	MOVL A, @RW2 + d8	MOVL RL1, A	MOVL @R W2 + d8,A	MOV R2,#8	MOV @R W2 + d8,#8	MOVEA A,RW2	MOVEA A, @RW2 + d8
+3	JMPP @RL1	JMPP @@RW3 + d8	CALLP @RL1	CALLP @@RW3 + d8	INCL RL1	INCL @RW3 + d8	DECL RL1	DECL @RW3 + d8	MOVL A, RL1	MOVL A, @RW3 + d8	MOVL RL1, A	MOVL @R W3 + d8,A	MOV R3,#8	MOV @R W3 + d8,#8	MOVEA A,RW3	MOVEA A, @RW3 + d8
+4	JMPP @RL2	JMPP @@RW4 + d8	CALLP @RL2	CALLP @@RW4 + d8	INCL RL2	INCL @RW4 + d8	DECL RL2	DECL @RW4 + d8	MOVL A, RL2	MOVL A, @RW4 + d8	MOVL RL2, A	MOVL@R W4 + d8,A	MOV R4,#8	MOV @R W4 + d8,#8	MOVEA A,RW4	MOVEA A, @RW4 + d8
+5	JMPP @RL2	JMPP @@RW5 + d8	CALLP @RL2	CALLP @@RW5 + d8	INCL RL2	INCL @RW5 + d8	DECL RL2	DECL @RW5 + d8	MOVL A, RL2	MOVL A, @RW5 + d8	MOVL RL2, A	MOVL@R W5 + d8,A	MOV R5,#8	MOV @R W5 + d8,#8	MOVEA A,RW5	MOVEA A, @RW5 + d8
+6	JMPP @RL3	JMPP @@RW6 + d8	CALLP @RL3	CALLP @@RW6 + d8	INCL RL3	INCL @RW6 + d8	DECL RL3	DECL @RW6 + d8	MOVL A, RL3	MOVL A, @RW6 + d8	MOVL RL3, A	MOVL @R W6 + d8,A	MOV R6,#8	MOV @R W6 + d8,#8	MOVEA A,RW6	MOVEA A, @RW6 + d8
+7	JMPP @RL3	JMPP @@RW7 + d8	CALLP @RL3	CALLP @@RW7 + d8	INCL RL3	INCL @RW7 + d8	DECL RL3	DECL @RW7 + d8	MOVL A, RL3	MOVL A, @RW7 + d8	MOVL RL3, A	MOVL @R W7 + d8,A	MOV R7,#8	MOV @R W7 + d8,#8	MOVEA A,RW7	MOVEA A, @RW7 + d8
+8	JMPP @@RW0	JMPP @ @RW0 + d16	CALLP @@RW0	CALLP @ @RW0 + d16	INCL @RW0	INCL @RW0 + d16	DECL @RW0	DECL @RW0 + d16	MOVL A, @RW0	MOVL A, @RW0 + d16	MOVL @RW0, A	MOVL @R W0 + d16,A	MOV @RW0,#8	MOV @R W0 + d16,#8	MOVEA A,@RW0	MOVEA A, @RW0 + d16
+9	JMPP @@RW1	JMPP @ @RW1 + d16	CALLP @@RW1	CALLP @ @RW1 + d16	INCL @RW1	INCL @RW1 + d16	DECL @RW1	DECL @RW1 + d16	MOVL A, @RW1	MOVL A, @RW1 + d16	MOVL @RW1, A	MOVL @R W1 + d16,A	MOV @RW1,#8	MOV @R W1 + d16,#8	MOVEA A,@RW1	MOVEA A, @RW1 + d16
+A	JMPP @@RW2	JMPP @ @RW2 + d16	CALLP @@RW2	CALLP @ @RW2 + d16	INCL @RW2	INCL @RW2 + d16	DECL @RW2	DECL @RW2 + d16	MOVL A, @RW2	MOVL A, @RW2 + d16	MOVL @RW2, A	MOVL @R W2 + d16,A	MOV @RW2,#8	MOV @R W2 + d16,#8	MOVEA A,@RW2	MOVEA A, @RW2 + d16
+B	JMPP @@RW3	JMPP @ @RW3 + d16	CALLP @@RW3	CALLP @ @RW3 + d16	INCL @RW3	INCL @RW3 + d16	DECL @RW3	DECL @RW3 + d16	MOVL A, @RW3	MOVL A, @RW3 + d16	MOVL @RW3, A	MOVL @R W3 + d16,A	MOV @RW3,#8	MOV @R W3 + d16,#8	MOVEA A,@RW3	MOVEA A, @RW3 + d16
+C	JMPP @@RW0+	JMPP @ @RW0+RW7	CALLP @@RW0+	CALLP @ @RW0+RW7	INCL @RW0+	INCL @RW0+RW7	DECL @RW0+	DECL @RW0+RW7	MOVL A, @RW0+	MOVL A, @RW0+RW7	MOVL @RW0+, A	MOVL @R W0+RW7,A	MOV @RW0+,#8	MOV @R W0+RW7,#8	MOVEA A,@RW0+	MOVEA A, @RW0+RW7
+D	JMPP @@RW1+	JMPP @ @RW1+RW7	CALLP @@RW1+	CALLP @ @RW1+RW7	INCL @RW1+	INCL @RW1+RW7	DECL @RW1+	DECL @RW1+RW7	MOVL A, @RW1+	MOVL A, @RW1+RW7	MOVL @RW1+, A	MOVL @R W1+RW7,A	MOV @RW1+,#8	MOV @R W1+RW7,#8	MOVEA A,@RW1+	MOVEA A, @RW1+RW7
+E	JMPPR @@PC+	JMPP @@addr16	CALLP @@RW2+	CALLP @@PC + d16	INCL @RW2+	INCL @PC + d16	DECL @RW2+	DECL @PC + d16	MOVL A, @RW2+	MOVL A, @PC + d16	MOVL @RW2+, A	MOVL @P C + d16,A	MOV @RW2+,#8	MOV @P C + d16,#8	MOVEA A,@RW2+	MOVEA A, @PC + d16
+F	JMPP @@RW3+	JMPP @@addr16	CALLP @@RW3+	CALLP @@addr16	INCL @RW3+	INCL addr16	DECL @RW3+	DECL addr16	MOVL A, @RW3+	MOVL A, addr16	MOVL @RW3+, A	MOVL addr16,A	MOV @RW3+,#8	MOV addr16,#8	MOVEA A,@RW3+	MOVEA A, addr16

Table 37-31. ea系命令3 (第1バイト = 72_H)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	ROLC R0	ROLC @RW0 + d8	RORC R0	RORC @RW0 + d8	INC R0	INC @RW0 + d8	DEC R0	DEC @RW0 + d8	MOV A,R0	MOV A, @RW0 + d8	MOV R0,A	MOV @R W0 + d8,A	MOVX A,R0	MOVX A, @RW0 + d8	XCH A,R0	XCH A, @RW0 + d8
+1	ROLC R1	ROLC @RW1 + d8	RORC R1	RORC @RW1 + d8	INC R1	INC @RW1 + d8	DEC R1	DEC @RW1 + d8	MOV A,R1	MOV A, @RW1 + d8	MOV R1,A	MOV @R W1 + d8,A	MOVX A,R1	MOVX A, @RW1 + d8	XCH A,R1	XCH A, @RW1 + d8
+2	ROLC R2	ROLC @RW2 + d8	RORC R2	RORC @RW2 + d8	INC R2	INC @RW2 + d8	DEC R2	DEC @RW2 + d8	MOV A,R2	MOV A, @RW2 + d8	MOV R2,A	MOV @R W2 + d8,A	MOVX A,R2	MOVX A, @RW2 + d8	XCH A,R2	XCH A, @RW2 + d8
+3	ROLC R3	ROLC @RW3 + d8	RORC R3	RORC @RW3 + d8	INC R3	INC @RW3 + d8	DEC R3	DEC @RW3 + d8	MOV A,R3	MOV A, @RW3 + d8	MOV R3,A	MOV @R W3 + d8,A	MOVX A,R3	MOVX A, @RW3 + d8	XCH A,R3	XCH A, @RW3 + d8
+4	ROLC R4	ROLC @RW4 + d8	RORC R4	RORC @RW4 + d8	INC R4	INC @RW4 + d8	DEC R4	DEC @RW4 + d8	MOV A,R4	MOV A, @RW4 + d8	MOV R4,A	MOV @R W4 + d8,A	MOVX A,R4	MOVX A, @RW4 + d8	XCH A,R4	XCH A, @RW4 + d8
+5	ROLC R5	ROLC @RW5 + d8	RORC R5	RORC @RW5 + d8	INC R5	INC @RW5 + d8	DEC R5	DEC @RW5 + d8	MOV A,R5	MOV A, @RW5 + d8	MOV R5,A	MOV @R W5 + d8,A	MOVX A,R5	MOVX A, @RW5 + d8	XCH A,R5	XCH A, @RW5 + d8
+6	ROLC R6	ROLC @RW6 + d8	RORC R6	RORC @RW6 + d8	INC R6	INC @RW6 + d8	DEC R6	DEC @RW6 + d8	MOV A,R6	MOV A, @RW6 + d8	MOV R6,A	MOV @R W6 + d8,A	MOVX A,R6	MOVX A, @RW6 + d8	XCH A,R6	XCH A, @RW6 + d8
+7	ROLC R7	ROLC @RW7 + d8	RORC R7	RORC @RW7 + d8	INC R7	INC @RW7 + d8	DEC R7	DEC @RW7 + d8	MOV A,R7	MOV A, @RW7 + d8	MOV R7,A	MOV @R W7 + d8,A	MOVX A,R7	MOVX A, @RW7 + d8	XCH A,R7	XCH A, @RW7 + d8
+8	ROLC @RW0	ROLC @RW0 + d16	RORC @RW0	RORC @RW0 + d16	INC @RW0	INC @RW0 + d16	DEC @RW0	DEC @RW0 + d16	MOV A,@RW0	MOV A, @RW0 + d16	MOV @RW0, A	MOV @R W0 + d16,A	MOVX A,@RW0	MOVX A, @RW0 + d16	XCH A,@RW0	XCH A, @@RW0 + d16
+9	ROLC @RW1	ROLC @RW1 + d16	RORC @RW1	RORC @RW1 + d16	INC @RW1	INC @RW1 + d16	DEC @RW1	DEC @RW1 + d16	MOV A,@RW1	MOV A, @RW1 + d16	MOV @RW1, A	MOV @R W1 + d16,A	MOVX A,@RW1	MOVX A, @RW1 + d16	XCH A,@RW1	XCH A, @@RW1 + d16
+A	ROLC @RW2	ROLC @RW2 + d16	RORC @RW2	RORC @RW2 + d16	INC @RW2	INC @RW2 + d16	DEC @RW2	DEC @RW2 + d16	MOV A,@RW2	MOV A, @RW2 + d16	MOV @RW2, A	MOV @R W2 + d16,A	MOVX A,@RW2	MOVX A, @RW2 + d16	XCH A,@RW2	XCH A, @@RW2 + d16
+B	ROLC @RW3	ROLC @RW3 + d16	RORC @RW3	RORC @RW3 + d16	INC @RW3	INC @RW3 + d16	DEC @RW3	DEC @RW3 + d16	MOV A,@RW3	MOV A, @RW3 + d16	MOV @RW3, A	MOV @R W3 + d16,A	MOVX A,@RW3	MOVX A, @RW3 + d16	XCH A,@RW3	XCH A, @@RW3 + d16
+C	ROLC @RW0+	ROLC @RW0+RW7	RORC @RW0+	RORC @RW0+RW7	+INC @RW0	+INC @RW0+RW7	DEC @RW0+	DEC @RW0+RW7	MOV A,@RW0+	MOV A, @RW0+RW7	MOV @RW0+, A	MOV @R W0+RW7,A	MOVX A,@RW0+	MOVX A, @RW0+RW7	XCH A,@RW0+	XCH A, @@RW0+RW7
+D	ROLC @RW1+	ROLC @RW1+RW7	RORC @RW1+	RORC @RW1+RW7	INC @RW1+	INC @RW1+RW7	DEC @RW1+	DEC @RW1+RW7	MOV A,@RW1+	MOV A, @RW1+RW7	MOV @RW1+, A	MOV @R W1+RW7,A	MOVX A,@RW1+	MOVX A, @RW1+RW7	XCH A,@RW1+	XCH A, @@RW1+RW7
+E	ROLC @RW2+	ROLC @PC + d16	RORC @RW2+	RORC @PC + d16	INC @RW2+	INC @PC + d16	DEC @RW2+	DEC @PC + d16	MOV A,@RW2+	MOV A, @PC + d16	MOV @RW2+, A	MOV @P C + d16, A	MOVX A,@RW2+	MOVX A, @PC + d16	XCH A,@RW2+	XCH A, @PC + d16
+F	ROLC @RW3+	ROLC addr16	RORC @RW3+	RORC addr16	INC @RW3+	INC addr16	DEC @RW3+	DEC addr16	MOV A,@RW3+	MOV A, addr16	MOV @RW3+, A	MOV addr16, A	MOVX A,@RW3+	MOVX A, addr16	XCH A,@RW3+	XCH A, addr16

Table 37-32. ea系命令4 (第1バイト = 73_H)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	JMP @RW0	JMP @@RW0 + d8	CALL @RW0	CALL @@RW0 + d8	INCW RW0	INCW @RW0 + d8	DECW RW0	DECW @RW0 + d8	MOVW A,RW0	MOVW A, @RW0 + d8	MOVW RW0,A	MOVW @R W0 + d8,A	MOVW RW0,#16	MOVW @RW 0 + d8,#16	XCHW A, RW0	XCHW A, @RW0 + d8
+1	JMP @RW1	JMP @@RW1 + d8	CALL @RW1	CALL @@RW1 + d8	INCW RW1	INCW @RW1 + d8	DECW RW1	DECW @RW1 + d8	MOVW A,RW1	MOVW A, @RW1 + d8	MOVW RW1,A	MOVW @R W1 + d8,A	MOVW RW1,#16	MOVW @RW 1 + d8,#16	XCHW A, RW1	XCHW A, @RW1 + d8
+2	JMP @RW2	JMP @@RW2 + d8	CALL @RW2	CALL @@RW2 + d8	INCW RW2	INCW @RW2 + d8	DECW RW2	DECW @RW2 + d8	MOVW A,RW2	MOVW A, @RW2 + d8	MOVW RW2,A	MOVW @R W2 + d8,A	MOVW RW2,#16	MOVW @RW 2 + d8,#16	XCHW A, RW2	XCHW A, @RW2 + d8
+3	JMP @RW3	JMP @@RW3 + d8	CALL @RW3	CALL @@RW3 + d8	INCW RW3	INCW @RW3 + d8	DECW RW3	DECW @RW3 + d8	MOVW A,RW3	MOVW A, @RW3 + d8	MOVW RW3,A	MOVW @R W3 + d8,A	MOVW RW3,#16	MOVW @RW 3 + d8,#16	XCHW A, RW3	XCHW A, @RW3 + d8
+4	JMP @RW4	JMP @@RW4 + d8	CALL @RW4	CALL @@RW4 + d8	INCW RW4	INCW @RW4 + d8	DECW RW4	DECW @RW4 + d8	MOVW A,RW4	MOVW A, @RW4 + d8	MOVW RW4,A	MOVW @R W4 + d8,A	MOVW RW4,#16	MOVW @RW 4 + d8,#16	XCHW A, RW4	XCHW A, @RW4 + d8
+5	JMP @RW5	JMP @@RW5 + d8	CALL @RW5	CALL @@RW5 + d8	INCW RW5	INCW @RW5 + d8	DECW RW5	DECW @RW5 + d8	MOVW A,RW5	MOVW A, @RW5 + d8	MOVW RW5,A	MOVW @R W5 + d8,A	MOVW RW5,#16	MOVW @RW 5 + d8,#16	XCHW A, RW5	XCHW A, @RW5 + d8
+6	JMP @RW6	JMP @@RW6 + d8	CALL @RW6	CALL @@RW6 + d8	INCW RW6	INCW @RW6 + d8	DECW RW6	DECW @RW6 + d8	MOVW A,RW6	MOVW A, @RW6 + d8	MOVW RW6,A	MOVW @R W6 + d8,A	MOVW RW6,#16	MOVW @RW 6 + d8,#16	XCHW A, RW6	XCHW A, @RW6 + d8
+7	JMP @RW7	JMP @@RW7 + d8	CALL @RW7	CALL @@RW7 + d8	INCW RW7	INCW @RW7 + d8	DECW RW7	DECW @RW7 + d8	MOVW A,RW7	MOVW A, @RW7 + d8	MOVW RW7,A	MOVW @R W7 + d8,A	MOVW RW7,#16	MOVW @RW 7 + d8,#16	XCHW A, RW7	XCHW A, @RW7 + d8
+8	JMP @@RW0	JMP @@RW0 + d16	CALL @@RW0	CALL @@RW0 + d16	INCW @RW0	INCW @RW0 + d16	DECW @RW0	DECW @RW0 + d16	MOVW A,@RW0	MOVW A, @RW0 + d16	MOVW @RW0,A	MOVW @R W0 + d16,A	MOVW @RW0,#16	MOVW@RW0 + d16,#16	XCHW A, @RW0	XCHW A, @RW0 + d16
+9	JMP @@RW1	JMP @@RW1 + d16	CALL @@RW1	CALL @@RW1 + d16	INCW @RW1	INCW @RW1 + d16	DECW @RW1	DECW @RW1 + d16	MOVW A,@RW1	MOVW A, @RW1 + d16	MOVW @RW1,A	MOVW @R W1 + d16,A	MOVW @RW1,#16	MOVW@RW1 + d16,#16	XCHW A, @RW1	XCHW A, @RW1 + d16
+A	JMP @@RW2	JMP @@RW2 + d16	CALL @@RW2	CALL @@RW2 + d16	INCW @RW2	INCW @RW2 + d16	DECW @RW2	DECW @RW2 + d16	MOVW A,@RW2	MOVW A, @RW2 + d16	MOVW @RW2,A	MOVW @R W2 + d16,A	MOVW @RW2,#16	MOVW@RW2 + d16,#16	XCHW A, @RW2	XCHW A, @RW2 + d16
+B	JMP @@RW3	JMP @@RW3 + d16	CALL @@RW3	CALL @@RW3 + d16	INCW @RW3	INCW @RW3 + d16	DECW @RW3	DECW @RW3 + d16	MOVW A,@RW3	MOVW A, @RW3 + d16	MOVW @RW3,A	MOVW @R W3 + d16,A	MOVW @RW3,#16	MOVW@RW3 + d16,#16	XCHW A, @RW3	XCHW A, @RW3 + d16
+C	JMP @@RW0+	JMP @@RW0+RW7	CALL @@RW0+	CALL @@RW0+RW7	INCW @RW0+	INCW @RW0+RW7	DECW @RW0+	DECW @RW0+RW7	MOVW A,@RW0+	MOVW A, @RW0+RW7	MOVW @RW0+,A	MOVW @R W0+RW7,A	MOVW @RW0+,#16	MOVW@RW0 +RW7,#16	XCHW A, @RW0+	XCHW A, @RW0+RW7
+D	JMP @@RW1+	JMP @@RW1+RW7	CALL @@RW1+	CALL @@RW1+RW7	INCW @RW1+	INCW @RW1+RW7	DECW @RW1+	DECW @RW1+RW7	MOVW A,@RW1+	MOVW A, @RW1+RW7	MOVW @RW1+,A	MOVW @R W1+RW7,A	MOVW @RW1+,#16	MOVW@RW1 +RW7,#16	XCHW A, @RW1+	XCHW A, @RW1+RW7
+E	JMP @@RW2+	JMP @@PC + d16	CALL @@RW2+	CALL @@PC + d16	INCW @RW2+	INCW @PC + d16	DECW @RW2+	DECW @PC + d16	MOVW A,@RW2+	MOVW A, @PC + d16	MOVW @RW2+,A	MOVW @P C + d16,A	MOVW @RW2+,#16	MOVW @PC + d16,#16	XCHW A, @RW2+	XCHW A, @PC + d16
+F	JMP @@RW3+	JMP @addr16	CALL @@RW3+	CALL @addr16	INCW @RW3+	INCW addr16	DECW @RW3+	DECW addr16	MOVW A,@RW3+	MOVW A, addr16	MOVW @RW3+,A	MOVW addr16, A	MOVW @RW3+,#16	MOVW ad dr16,#16	XCHW A, @RW3+	XCHW A, addr16

Table 37-33. ea系命令5 (第1バイト = 74_H)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	ADD A,R0	ADD A, @RW0 + d8	SUB A,R0	SUB A, @RW0 + d8	ADDC A, R0	ADDC A, @RW0 + d8	CMP A,R0	CMP A, @RW0 + d8	AND A, R0	AND A, @RW0 + d8	OR A,R0	OR A, @RW0 + d8	XOR A, R0	XOR A, @RW0 + d8	DBNZ R0, r	DBNZ @ RW0 + d8, r
+1	ADD A,R1	ADD A, @RW1 + d8	SUB A,R1	SUB A, @RW1 + d8	ADDC A, R1	ADDC A, @RW1 + d8	CMP A,R1	CMP A, @RW1 + d8	AND A, R1	AND A, @RW1 + d8	OR A,R1	OR A, @RW1 + d8	XOR A, R1	XOR A, @RW1 + d8	DBNZ R1, r	DBNZ @ RW1 + d8, r
+2	ADD A,R2	ADD A, @RW2 + d8	SUB A,R2	SUB A, @RW2 + d8	ADDC A, R2	ADDC A, @RW2 + d8	CMP A,R2	CMP A, @RW2 + d8	AND A, R2	AND A, @RW2 + d8	OR A,R2	OR A, @RW2 + d8	XOR A, R2	XOR A, @RW2 + d8	DBNZ R2, r	DBNZ @ RW2 + d8, r
+3	ADD A,R3	ADD A, @RW3 + d8	SUB A,R3	SUB A, @RW3 + d8	ADDC A, R3	ADDC A, @RW3 + d8	CMP A,R3	CMP A, @RW3 + d8	AND A, R3	AND A, @RW3 + d8	OR A,R3	OR A, @RW3 + d8	XOR A, R3	XOR A, @RW3 + d8	DBNZ R3, r	DBNZ @ RW3 + d8, r
+4	ADD A,R4	ADD A, @RW4 + d8	SUB A,R4	SUB A, @RW4 + d8	ADDC A, R4	ADDC A, @RW4 + d8	CMP A,R4	CMP A, @RW4 + d8	AND A, R4	AND A, @RW4 + d8	OR A,R4	OR A, @RW4 + d8	XOR A, R4	XOR A, @RW4 + d8	DBNZ R4, r	DBNZ @ RW4 + d8, r
+5	ADD A,R5	ADD A, @RW5 + d8	SUB A,R5	SUB A, @RW5 + d8	ADDC A, R5	ADDC A, @RW5 + d8	CMP A,R5	CMP A, @RW5 + d8	AND A, R5	AND A, @RW5 + d8	OR A,R5	OR A, @RW5 + d8	XOR A, R5	XOR A, @RW5 + d8	DBNZ R5, r	DBNZ @ RW5 + d8, r
+6	ADD A,R6	ADD A, @RW6 + d8	SUB A,R6	SUB A, @RW6 + d8	ADDC A, R6	ADDC A, @RW6 + d8	CMP A,R6	CMP A, @RW6 + d8	AND A, R6	AND A, @RW6 + d8	OR A,R6	OR A, @RW6 + d8	XOR A, R6	XOR A, @RW6 + d8	DBNZ R6, r	DBNZ @ RW6 + d8, r
+7	ADD A,R7	ADD A, @RW7 + d8	SUB A,R7	SUB A, @RW7 + d8	ADDC A, R7	ADDC A, @RW7 + d8	CMP A,R7	CMP A, @RW7 + d8	AND A, R7	AND A, @RW7 + d8	OR A,R7	OR A, @RW7 + d8	XOR A, R7	XOR A, @RW7 + d8	DBNZ R7, r	DBNZ @ RW7 + d8, r
+8	ADD A,@RW0	ADD A, @RW0 + d16	SUB A,@RW0	SUB A, @RW0 + d16	ADDC A, @RW0	ADDC A, @RW0 + d16	CMP A,@RW0	CMP A, @RW0 + d16	AND A, @RW0	AND A, @RW0 + d16	OR A,@RW0	OR A, @RW0 + d16	XOR A, @RW0	XOR A, @RW0 + d16	DBNZ @RW0, r	DBNZ @RW0 + d16, r
+9	ADD A,@RW1	ADD A, @RW1 + d16	SUB A,@RW1	SUB A, @RW1 + d16	ADDC A, @RW1	ADDC A, @RW1 + d16	CMP A,@RW1	CMP A, @RW1 + d16	AND A, @RW1	AND A, @RW1 + d16	OR A,@RW1	OR A, @RW1 + d16	XOR A, @RW1	XOR A, @RW1 + d16	DBNZ @RW1, r	DBNZ @RW1 + d16, r
+A	ADD A,@RW2	ADD A, @RW2 + d16	SUB A,@RW2	SUB A, @RW2 + d16	ADDC A, @RW2	ADDC A, @RW2 + d16	CMP A,@RW2	CMP A, @RW2 + d16	AND A, @RW2	AND A, @RW2 + d16	OR A,@RW2	OR A, @RW2 + d16	XOR A, @RW2	XOR A, @RW2 + d16	DBNZ @RW2, r	DBNZ @RW2 + d16, r
+B	ADD A,@RW3	ADD A, @RW3 + d16	SUB A,@RW3	SUB A, @RW3 + d16	ADDC A, @RW3	ADDC A, @RW3 + d16	CMP A,@RW3	CMP A, @RW3 + d16	AND A, @RW3	AND A, @RW3 + d16	OR A,@RW3	OR A, @RW3 + d16	XOR A, @RW3	XOR A, @RW3 + d16	DBNZ @RW3, r	DBNZ @RW3 + d16, r
+C	ADD A,@RW0+	ADD A, @RW0+RW7	SUB A,@RW0+	SUB A, @RW0+RW7	ADDC A, @RW0+	ADDC A, @RW0+RW7	CMP A,@RW0+	CMP A, @RW0+RW7	AND A, @RW0+	AND A, @RW0+RW7	OR A,@RW0+	OR A, @RW0+RW7	XOR A, @RW0+	XOR A, @RW0+RW7	DBNZ @RW0+, r	DBNZ @RW0+RW7, r
+D	ADD A,@RW1+	ADD A, @RW1+RW7	SUB A,@RW1+	SUB A, @RW1+RW7	ADDC A, @RW1+	ADDC A, @RW1+RW7	CMP A,@RW1+	CMP A, @RW1+RW7	AND A, @RW1+	AND A, @RW1+RW7	OR A,@RW1+	OR A, @RW1+RW7	XOR A, @RW1+	XOR A, @RW1+RW7	DBNZ @RW1+, r	DBNZ @RW1+RW7, r
+E	ADD A,@RW2+	ADD A, @PC + d16	SUB A,@RW2+	SUB A, @PC + d16	ADDC A, @RW2+	ADDC A, @PC + d16	CMP A,@RW2+	CMP A, @PC + d16	AND A, @RW2+	AND A, @PC + d16	OR A,@RW2+	OR A, @PC + d16	XOR A, @RW2+	XOR A, @PC + d16	DBNZ @RW2+, r	DBNZ @PC + d16, r
+F	ADD A,@RW3+	ADD A, addr16	SUB A,@RW3+	SUB A, addr16	ADDC A, @RW3+	ADDC A, addr16	CMP A,@RW3+	CMP A, addr16	AND A, @RW3+	AND A, addr16	OR A,@RW3+	OR A, addr16	XOR A, @RW3+	XOR A, addr16	DBNZ @RW3+, r	DBNZ @addr16, r

Table 37-34. ea系命令6 (第1バイト = 75_H)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	ADD R0,A	ADD @R W0 + d8,A	SUB R0,A	SUB @R W0 + d8,A	SUBC A,R0	SUBC A, @RW0 + d8	NEG R0	NEG @RW0 + d8	AND R0,A	AND @R W0 + d8,A	OR R0, A	OR @RW0 + d8, A	XOR R0, A	XOR @R W0 + d8, A	NOT R0	NOT @RW0 + d8
+1	ADD R1,A	ADD @R W1 + d8,A	SUB R1,A	SUB @R W1 + d8,A	SUBC A,R1	SUBC A, @RW1 + d8	NEG R1	NEG @RW1 + d8	AND R1,A	AND @R W1 + d8,A	OR R1, A	OR @RW1 + d8, A	XOR R1, A	XOR @R W1 + d8, A	NOT R1	NOT @RW1 + d8
+2	ADD R2,A	ADD @R W2 + d8,A	SUB R2,A	SUB @R W2 + d8,A	SUBC A,R2	SUBC A, @RW2 + d8	NEG R2	NEG @RW2 + d8	AND R2,A	AND @R W2 + d8,A	OR R2, A	OR @RW2 + d8, A	XOR R2, A	XOR @R W2 + d8, A	NOT R2	NOT @RW2 + d8
+3	ADD R3,A	ADD @R W3 + d8,A	SUB R3,A	SUB @R W3 + d8,A	SUBC A,R3	SUBC A, @RW3 + d8	NEG R3	NEG @RW3 + d8	AND R3,A	AND @R W3 + d8,A	OR R3, A	OR @RW3 + d8, A	XOR R3, A	XOR @R W3 + d8, A	NOT R3	NOT @RW3 + d8
+4	ADD R4,A	ADD @R W4 + d8,A	SUB R4,A	SUB @R W4 + d8,A	SUBC A,R4	SUBC A, @RW4 + d8	NEG R4	NEG @RW4 + d8	AND R4,A	AND @R W4 + d8,A	OR R4, A	OR @RW4 + d8, A	XOR R4, A	XOR @R W4 + d8, A	NOT R4	NOT @RW4 + d8
+5	ADD R5,A	ADD @R W5 + d8,A	SUB R5,A	SUB @R W5 + d8,A	SUBC A,R5	SUBC A, @RW5 + d8	NEG R5	NEG @RW5 + d8	AND R5,A	AND @R W5 + d8,A	OR R5, A	OR @RW5 + d8, A	XOR R5, A	XOR @R W5 + d8, A	NOT R5	NOT @RW5 + d8
+6	ADD R6,A	ADD @R W6 + d8,A	SUB R6,A	SUB @R W6 + d8,A	SUBC A,R6	SUBC A, @RW6 + d8	NEG R6	NEG @RW6 + d8	AND R6,A	AND @R W6 + d8,A	OR R6, A	OR @RW6 + d8, A	XOR R6, A	XOR @R W6 + d8, A	NOT R6	NOT @RW6 + d8
+7	ADD R7,A	ADD @R W7 + d8,A	SUB R7,A	SUB @R W7 + d8,A	SUBC A,R7	SUBC A, @RW7 + d8	NEG R7	NEG @RW7 + d8	AND R7,A	AND @R W7 + d8,A	OR R7, A	OR @RW7 + d8, A	XOR R7, A	XOR @R W7 + d8, A	NOT R7	NOT @RW7 + d8
+8	ADD @RW0,A	ADD @R W0 + d16,A	SUB @RW0,A	SUB @R W0 + d16,A	SUBC A,@RW0	SUBC A, @RW0 + d16	NEG @RW0	NEG @RW0 + d16	AND @RW0,A	AND @R W0 + d16,A	OR @RW0, A	OR @RW0 + d16, A	XOR @RW0, A	XOR @R W0 + d16, A	NOT @RW0	NOT @RW0 + d16
+9	ADD @RW1,A	ADD @R W1 + d16,A	SUB @RW1,A	SUB @R W1 + d16,A	SUBC A,@RW1	SUBC A, @RW1 + d16	NEG @RW1	NEG @RW1 + d16	AND @RW1,A	AND @R W1 + d16,A	OR @RW1, A	OR @RW1 + d16, A	XOR @RW1, A	XOR @R W1 + d16, A	NOT @RW1	NOT @RW1 + d16
+A	ADD @RW2,A	ADD @R W2 + d16,A	SUB @RW2,A	SUB @R W2 + d16,A	SUBC A,@RW2	SUBC A, @RW2 + d16	NEG @RW2	NEG @RW2 + d16	AND @RW2,A	AND @R W2 + d16,A	OR @RW2, A	OR @RW2 + d16, A	XOR @RW2, A	XOR @R W2 + d16, A	NOT @RW2	NOT @RW2 + d16
+B	ADD @RW3,A	ADD @R W3 + d16,A	SUB @RW3,A	SUB @R W3 + d16,A	SUBC A, @RW3	SUBC A, @RW3 + d16	NEG @RW3	NEG @RW3 + d16	AND @RW3,A	AND @R W3 + d16,A	OR @RW3, A	OR @RW3 + d16, A	XOR @RW3, A	XOR @R W3 + d16, A	NOT @RW3	NOT @RW3 + d16
+C	ADD @RW0+,A	ADD @R W0+RW7,A	SUB @RW0+,A	SUB @R W0+RW7,A	SUBC A,@RW0+	SUBC A, @RW0+RW7	NEG @RW0+	NEG @RW0+RW7	AND @RW0+,A	AND @R W0+RW7,A	OR @RW0+, A	OR @R W0+RW7, A	XOR @RW0+, A	XOR @R W0+RW7, A	NOT @RW0+	NOT @RW0+RW7
+D	ADD @RW1+,A	ADD @R W1+RW7,A	SUB @RW1+,A	SUB @R W1+RW7,A	SUBC A,@RW1+	SUBC A, @RW1+RW7	NEG @RW1+	NEG @RW1+RW7	AND @RW1+,A	AND @R W1+RW7,A	OR @RW1+, A	OR @R W1+RW7, A	XOR @RW1+, A	XOR @R W1+RW7, A	NOT @RW1+	NOT @RW1+RW7
+E	ADD @RW2+,A	ADD @P C + d16,A	SUB @RW2+,A	SUB @P C + d16,A	SUBC A,@RW2+	SUBC A, @PC + d16	NEG @RW2+	NEG @PC + d16	AND @RW2+,A	AND @P C + d16,A	OR @RW2+, A	OR @PC + d16, A	XOR @RW2+, A	XOR @P C + d16, A	NOT @RW2+	NOT @PC + d16
+F	ADD @RW3+,A	ADD addr16,A	SUB @RW3+,A	SUB addr16,A	SUBC A,@RW3+	SUBC A, addr16	NEG @RW3+	NEG addr16	AND @RW3+,A	AND addr16,A	OR @RW3+, A	OR addr16, A	XOR @RW3+, A	XOR addr16, A	NOT @RW3+	NOT addr16

Table 37-35. ea系命令7 (第1バイト = 76_H)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	ADDW A, RW0	ADDW A, @RW0 + d8	SUBW A, RW0	SUBW A, @RW0 + d8	ADDCW A, RW0	ADDCW A, @RW0 + d8	CMPW A, RW0	CMPW A, @RW0 + d8	ANDW A, RW0	ANDW A, @RW0 + d8	ORW A, RW0	ORW A, @RW0 + d8	XORW A, RW0	XORW A, @RW0 + d8	DWBNZ RW0,r	DWBNZ @ RW0 + d8,r
+1	ADDW A, RW1	ADDW A, @RW1 + d8	SUBW A, RW1	SUBW A, @RW1 + d8	ADDCW A, RW1	ADDCW A, @RW1 + d8	CMPW A, RW1	CMPW A, @RW1 + d8	ANDW A, RW1	ANDW A, @RW1 + d8	ORW A, RW1	ORW A, @RW1 + d8	XORW A, RW1	XORW A, @RW1 + d8	DWBNZ RW1,r	DWBNZ @ RW1 + d8,r
+2	ADDW A, RW2	ADDW A, @RW2 + d8	SUBW A, RW2	SUBW A, @RW2 + d8	ADDCW A, RW2	ADDCW A, @RW2 + d8	CMPW A, RW2	CMPW A, @RW2 + d8	ANDW A, RW2	ANDW A, @RW2 + d8	ORW A, RW2	ORW A, @RW2 + d8	XORW A, RW2	XORW A, @RW2 + d8	DWBNZ RW2,r	DWBNZ @ RW2 + d8,r
+3	ADDW A, RW3	ADDW A, @RW3 + d8	SUBW A, RW3	SUBW A, @RW3 + d8	ADDCW A, RW3	ADDCW A, @RW3 + d8	CMPW A, RW3	CMPW A, @RW3 + d8	ANDW A, RW3	ANDW A, @RW3 + d8	ORW A, RW3	ORW A, @RW3 + d8	XORW A, RW3	XORW A, @RW3 + d8	DWBNZ RW3,r	DWBNZ @ RW3 + d8,r
+4	ADDW A, RW4	ADDW A, @RW4 + d8	SUBW A, RW4	SUBW A, @RW4 + d8	ADDCW A, RW4	ADDCW A, @RW4 + d8	CMPW A, RW4	CMPW A, @RW4 + d8	ANDW A, RW4	ANDW A, @RW4 + d8	ORW A, RW4	ORW A, @RW4 + d8	XORW A, RW4	XORW A, @RW4 + d8	DWBNZ RW4,r	DWBNZ @ RW4 + d8,r
+5	ADDW A, RW5	ADDW A, @RW5 + d8	SUBW A, RW5	SUBW A, @RW5 + d8	ADDCW A, RW5	ADDCW A, @RW5 + d8	CMPW A, RW5	CMPW A, @RW5 + d8	ANDW A, RW5	ANDW A, @RW5 + d8	ORW A, RW5	ORW A, @RW5 + d8	XORW A, RW5	XORW A, @RW5 + d8	DWBNZ RW5,r	DWBNZ @ RW5 + d8,r
+6	ADDW A, RW6	ADDW A, @RW6 + d8	SUBW A, RW6	SUBW A, @RW6 + d8	ADDCW A, RW6	ADDCW A, @RW6 + d8	CMPW A, RW6	CMPW A, @RW6 + d8	ANDW A, RW6	ANDW A, @RW6 + d8	ORW A, RW6	ORW A, @RW6 + d8	XORW A, RW6	XORW A, @RW6 + d8	DWBNZ RW6,r	DWBNZ @ RW6 + d8,r
+7	ADDW A, RW7	ADDW A, @RW7 + d8	SUBW A, RW7	SUBW A, @RW7 + d8	ADDCW A, RW7	ADDCW A, @RW7 + d8	CMPW A, RW7	CMPW A, @RW7 + d8	ANDW A, RW7	ANDW A, @RW7 + d8	ORW A, RW7	ORW A, @RW7 + d8	XORW A, RW7	XORW A, @RW7 + d8	DWBNZ RW7,r	DWBNZ @ RW7 + d8,r
+8	ADDW A, @RW0	ADDW A, @RW0 + d16	SUBW A, @RW0	SUBW A, @RW0 + d16	ADDCW A, @RW0	ADDCW A, @RW0 + d16	CMPW A, @RW0	CMPW A, @RW0 + d16	ANDW A, @RW0	ANDW A, @RW0 + d16	ORW A, @RW0	ORW A, @RW0 + d16	XORW A, @RW0	XORW A, @RW0 + d16	DWBNZ @RW0,r	DWBNZ @R W0 + d16,r
+9	ADDW A, @RW1	ADDW A, @RW1 + d16	SUBW A, @RW1	SUBW A, @RW1 + d16	ADDCW A, @RW1	ADDCW A, @RW1 + d16	CMPW A, @RW1	CMPW A, @RW1 + d16	ANDW A, @RW1	ANDW A, @RW1 + d16	ORW A, @RW1	ORW A, @RW1 + d16	XORW A, @RW1	XORW A, @RW1 + d16	DWBNZ @RW1,r	DWBNZ @R W1 + d16,r
+A	ADDW A, @RW2	ADDW A, @RW2 + d16	SUBW A, @RW2	SUBW A, @RW2 + d16	ADDCW A, @RW2	ADDCW A, @RW2 + d16	CMPW A, @RW2	CMPW A, @RW2 + d16	ANDW A, @RW2	ANDW A, @RW2 + d16	ORW A, @RW2	ORW A, @RW2 + d16	XORW A, @RW2	XORW A, @RW2 + d16	DWBNZ @RW2,r	DWBNZ @R W2 + d16,r
+B	ADDW A, @RW3	ADDW A, @RW3 + d16	SUBW A, @RW3	SUBW A, @RW3 + d16	ADDCW A, @RW3	ADDCW A, @RW3 + d16	CMPW A, @RW3	CMPW A, @RW3 + d16	ANDW A, @RW3	ANDW A, @RW3 + d16	ORW A, @RW3	ORW A, @RW3 + d16	XORW A, @RW3	XORW A, @RW3 + d16	DWBNZ @RW3,r	DWBNZ @R W3 + d16,r
+C	ADDW A, @RW0+	ADDW A, @RW0+RW7	SUBW A, @RW0+	SUBW A, @RW0+RW7	ADDCW A, @RW0+	ADDCW A, @RW0+RW7	CMPW A, @RW0+	CMPW A, @RW0+RW7	ANDW A, @RW0+	ANDW A, @RW0+RW7	ORW A, @RW0+	ORW A, @RW0+RW7	XORW A, @RW0+	XORW A, @RW0+RW7	DWBNZ @RW0+,r	DWBNZ @R W0+RW7,r
+D	ADDW A, @RW1+	ADDW A, @RW1+RW7	SUBW A, @RW1+	SUBW A, @RW1+RW7	ADDCW A, @RW1+	ADDCW A, @RW1+RW7	CMPW A, @RW1+	CMPW A, @RW1+RW7	ANDW A, @RW1+	ANDW A, @RW1+RW7	ORW A, @RW1+	ORW A, @RW1+RW7	XORW A, @RW1+	XORW A, @RW1+RW7	DWBNZ @RW1+,r	DWBNZ @R W1+RW7,r
+E	ADDW A, @RW2+	ADDW A, @PC + d16	SUBW A, @RW2+	SUBW A, @PC + d16	ADDCW A, @RW2+	ADDCW A, @PC + d16	CMPW A, @RW2+	CMPW A, @PC + d16	ANDW A, @RW2+	ANDW A, @PC + d16	ORW A, @RW2+	ORW A, @PC + d16	XORW A, @RW2+	XORW A, @PC + d16	DWBNZ @RW2+,r	DWBNZ @ PC + d16,r
+F	ADDW A, @RW3+	ADDW A, addr16	SUBW A, @RW3+	SUBW A, addr16	ADDCW A, @RW3+	ADDCW A, addr16	CMPW A, @RW3+	CMPW A, addr16	ANDW A, @RW3+	ANDW A, addr16	ORW A, @RW3+	ORW A, addr16	XORW A, @RW3+	XORW A, addr16	DWBNZ @RW3+,r	DWBNZ addr16,r

Table 37-36. ea系命令8 (第1バイト = 77_H)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	ADDW RW0, A	ADDW @R W0 + d8, A	SUBW RW0, A	SUBW @R W0 + d8, A	SUBCW A, RW0	SUBCW A, @RW0 + d8	NEGW RW0	NEGW @RW0 + d8	ANDW RW0, A	ANDW @R W0 + d8, A	ORW RW0, A	ORW @R W0 + d8, A	XORW RW0, A	XORW @R W0 + d8, A	NOTW RW0	NOTW @RW0 + d8
+1	ADDW RW1, A	ADDW @R W1 + d8, A	SUBW RW1, A	SUBW @R W1 + d8, A	SUBCW A, RW1	SUBCW A, @RW1 + d8	NEGW RW1	NEGW @RW1 + d8	ANDW RW1, A	ANDW @R W1 + d8, A	ORW RW1, A	ORW @R W1 + d8, A	XORW RW1, A	XORW @R W1 + d8, A	NOTW RW1	NOTW @RW1 + d8
+2	ADDW RW2, A	ADDW @R W2 + d8, A	SUBW RW2, A	SUBW @R W2 + d8, A	SUBCW A, RW2	SUBCW A, @RW2 + d8	NEGW RW2	NEGW @RW2 + d8	ANDW RW2, A	ANDW @R W2 + d8, A	ORW RW2, A	ORW @R W2 + d8, A	XORW RW2, A	XORW @R W2 + d8, A	NOTW RW2	NOTW @RW2 + d8
+3	ADDW RW3, A	ADDW @R W3 + d8, A	SUBW RW3, A	SUBW @R W3 + d8, A	SUBCW A, RW3	SUBCW A, @RW3 + d8	NEGW RW3	NEGW @RW3 + d8	ANDW RW3, A	ANDW @R W3 + d8, A	ORW RW3, A	ORW @R W3 + d8, A	XORW RW3, A	XORW @R W3 + d8, A	NOTW RW3	NOTW @RW3 + d8
+4	ADDW RW4, A	ADDW @R W4 + d8, A	SUBW RW4, A	SUBW @R W4 + d8, A	SUBCW A, RW4	SUBCW A, @RW4 + d8	NEGW RW4	NEGW @RW4 + d8	ANDW RW4, A	ANDW @R W4 + d8, A	ORW RW4, A	ORW @R W4 + d8, A	XORW RW4, A	XORW @R W4 + d8, A	NOTW RW4	NOTW @RW4 + d8
+5	ADDW RW5, A	ADDW @R W5 + d8, A	SUBW RW5, A	SUBW @R W5 + d8, A	SUBCW A, RW5	SUBCW A, @RW5 + d8	NEGW RW5	NEGW @RW5 + d8	ANDW RW5, A	ANDW @R W5 + d8, A	ORW RW5, A	ORW @R W5 + d8, A	XORW RW5, A	XORW @R W5 + d8, A	NOTW RW5	NOTW @RW5 + d8
+6	ADDW RW6, A	ADDW @R W6 + d8, A	SUBW RW6, A	SUBW @R W6 + d8, A	SUBCW A, RW6	SUBCW A, @RW6 + d8	NEGW RW6	NEGW @RW6 + d8	ANDW RW6, A	ANDW @R W6 + d8, A	ORW RW6, A	ORW @R W6 + d8, A	XORW RW6, A	XORW @R W6 + d8, A	NOTW RW6	NOTW @RW6 + d8
+7	ADDW RW7, A	ADDW @R W7 + d8, A	SUBW RW7, A	SUBW @R W7 + d8, A	SUBCW A, RW7	SUBCW A, @RW7 + d8	NEGW RW7	NEGW @RW7 + d8	ANDW RW7, A	ANDW @R W7 + d8, A	ORW RW7, A	ORW @R W7 + d8, A	XORW RW7, A	XORW @R W7 + d8, A	NOTW RW7	NOTW @RW7 + d8
+8	ADDW @RW0, A	ADDW @R W0 + d16, A	SUBW @RW0, A	SUBW @R W0 + d16, A	SUBCW A, @RW0	SUBCW A, @RW0 + d16	NEGW @RW0	NEGW @RW0 + d16	ANDW @RW0, A	ANDW @R W0 + d16, A	ORW @RW0, A	ORW @R W0 + d16, A	XORW @RW0, A	XORW @R W0 + d16, A	NOTW @RW0	NOTW @RW0 + d16
+9	ADDW @RW1, A	ADDW @R W1 + d16, A	SUBW @RW1, A	SUBW @R W1 + d16, A	SUBCW A, @RW1	SUBCW A, @RW1 + d16	NEGW @RW1	NEGW @RW1 + d16	ANDW @RW1, A	ANDW @R W1 + d16, A	ORW @RW1, A	ORW @R W1 + d16, A	XORW @RW1, A	XORW @R W1 + d16, A	NOTW @RW1	NOTW @RW1 + d16
+A	ADDW @RW2, A	ADDW @R W2 + d16, A	SUBW @RW2, A	SUBW @R W2 + d16, A	SUBCW A, @RW2	SUBCW A, @RW2 + d16	NEGW @RW2	NEGW @RW2 + d16	ANDW @RW2, A	ANDW @R W2 + d16, A	ORW @RW2, A	ORW @R W2 + d16, A	XORW @RW2, A	XORW @R W2 + d16, A	NOTW @RW2	NOTW @RW2 + d16
+B	ADDW @RW3, A	ADDW @R W3 + d16, A	SUBW @RW3, A	SUBW @R W3 + d16, A	SUBCW A, @RW3	SUBCW A, @RW3 + d16	NEGW @RW3	NEGW @RW3 + d16	ANDW @RW3, A	ANDW @R W3 + d16, A	ORW @RW3, A	ORW @R W3 + d16, A	XORW @RW3, A	XORW @R W3 + d16, A	NOTW @RW3	NOTW @RW3 + d16
+C	ADDW @RW0+, A	ADDW @R W0+RW7, A	SUBW @RW0+, A	SUBW @R W0+RW7, A	SUBCW A, @RW0+	SUBCW A, @RW0+RW7	NEGW @RW0+	NEGW @RW0+RW7	ANDW @RW0+, A	ANDW @R W0+RW7, A	ORW @RW0+, A	ORW @R W0+RW7, A	XORW @RW0+, A	XORW @R W0+RW7, A	NOTW @RW0+	NOTW @RW0+RW7
+D	ADDW @RW1+, A	ADDW @R W1+RW7, A	SUBW @RW1+, A	SUBW @R W1+RW7, A	SUBCW A, @RW1+	SUBCW A, @RW1+RW7	NEGW @RW1+	NEGW @RW1+RW7	ANDW @RW1+, A	ANDW @R W1+RW7, A	ORW @RW1+, A	ORW @R W1+RW7, A	XORW @RW1+, A	XORW @R W1+RW7, A	NOTW @RW1+	NOTW @RW1+RW7
+E	ADDW @RW2+, A	ADDW @P C + d16, A	SUBW @RW2+, A	SUBW @P C + d16, A	SUBCW A, @RW2+	SUBCW A, @PC + d16	NEGW @RW2+	NEGW @PC + d16	ANDW @RW2+, A	ANDW @P C + d16, A	ORW @RW2+, A	ORW @P C + d16, A	XORW @RW2+, A	XORW @P C + d16, A	NOTW @RW2+	NOTW @PC + d16
+F	ADDW @RW3+, A	ADDW addr16, A	SUBW @RW3+, A	SUBW addr16, A	SUBCW A, @RW3+	SUBCW A, addr16	NEGW @RW3+	NEGW addr16	ANDW @RW3+, A	ANDW addr16, A	ORW @RW3+, A	ORW addr16, A	XORW @RW3+, A	XORW addr16, A	NOTW @RW3+	NOTW addr16

Table 37-37. ea系命令9 (第1バイト = 78_H)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	MULU A, R0	MULU A, @RW0 + d8	MULUW A, R0	MULUW A, @RW0 + d8	MUL A, R0	MUL A, @RW0 + d8	MULW A, R0	MULW A, @RW0 + d8	DIVU A, R0	DIVU A, @RW0 + d8	DIVUW A, R0	DIVUW A, @RW0 + d8	DIV A, R0	DIV A, @RW0 + d8	DIVW A, R0	DIVW A, @RW0 + d8
+1	MULU A, R1	MULU A, @RW1 + d8	MULUW A, R1	MULUW A, @RW1 + d8	MUL A, R1	MUL A, @RW1 + d8	MULW A, R1	MULW A, @RW1 + d8	DIVU A, R1	DIVU A, @RW1 + d8	DIVUW A, R1	DIVUW A, @RW1 + d8	DIV A, R1	DIV A, @RW1 + d8	DIVW A, R1	DIVW A, @RW1 + d8
+2	MULU A, R2	MULU A, @RW2 + d8	MULUW A, R2	MULUW A, @RW2 + d8	MUL A, R2	MUL A, @RW2 + d8	MULW A, R2	MULW A, @RW2 + d8	DIVU A, R2	DIVU A, @RW2 + d8	DIVUW A, R2	DIVUW A, @RW2 + d8	DIV A, R2	DIV A, @RW2 + d8	DIVW A, R2	DIVW A, @RW2 + d8
+3	MULU A, R3	MULU A, @RW3 + d8	MULUW A, R3	MULUW A, @RW3 + d8	MUL A, R3	MUL A, @RW3 + d8	MULW A, R3	MULW A, @RW3 + d8	DIVU A, R3	DIVU A, @RW3 + d8	DIVUW A, R3	DIVUW A, @RW3 + d8	DIV A, R3	DIV A, @RW3 + d8	DIVW A, R3	DIVW A, @RW3 + d8
+4	MULU A, R4	MULU A, @RW4 + d8	MULUW A, R4	MULUW A, @RW4 + d8	MUL A, R4	MUL A, @RW4 + d8	MULW A, R4	MULW A, @RW4 + d8	DIVU A, R4	DIVU A, @RW4 + d8	DIVUW A, R4	DIVUW A, @RW4 + d8	DIV A, R4	DIV A, @RW4 + d8	DIVW A, R4	DIVW A, @RW4 + d8
+5	MULU A, R5	MULU A, @RW5 + d8	MULUW A, R5	MULUW A, @RW5 + d8	MUL A, R5	MUL A, @RW5 + d8	MULW A, R5	MULW A, @RW5 + d8	DIVU A, R5	DIVU A, @RW5 + d8	DIVUW A, R5	DIVUW A, @RW5 + d8	DIV A, R5	DIV A, @RW5 + d8	DIVW A, R5	DIVW A, @RW5 + d8
+6	MULU A, R6	MULU A, @RW6 + d8	MULUW A, R6	MULUW A, @RW6 + d8	MUL A, R6	MUL A, @RW6 + d8	MULW A, R6	MULW A, @RW6 + d8	DIVU A, R6	DIVU A, @RW6 + d8	DIVUW A, R6	DIVUW A, @RW6 + d8	DIV A, R6	DIV A, @RW6 + d8	DIVW A, R6	DIVW A, @RW6 + d8
+7	MULU A, R7	MULU A, @RW7 + d8	MULUW A, R7	MULUW A, @RW7 + d8	MUL A, R7	MUL A, @RW7 + d8	MULW A, R7	MULW A, @RW7 + d8	DIVU A, R7	DIVU A, @RW7 + d8	DIVUW A, R7	DIVUW A, @RW7 + d8	DIV A, R7	DIV A, @RW7 + d8	DIVW A, R7	DIVW A, @RW7 + d8
+8	MULU A, @RW0	MULU A, @RW0 + d16	MULUW A, @RW0	MULUW A, @RW0 + d16	MUL A, @RW0	MUL A, @RW0 + d16	MULW A, @RW0	MULW A, @RW0 + d16	DIVU A, @RW0	DIVU A, @RW0 + d16	DIVUW A, @RW0	DIVUW A, @RW0 + d16	DIV A, @RW0	DIV A, @RW0 + d16	DIVW A, @RW0	DIVW A, @RW0 + d16
+9	MULU A, @RW1	MULU A, @RW1 + d16	MULUW A, @RW1	MULUW A, @RW1 + d16	MUL A, @RW1	MUL A, @RW1 + d16	MULW A, @RW1	MULW A, @RW1 + d16	DIVU A, @RW1	DIVU A, @RW1 + d16	DIVUW A, @RW1	DIVUW A, @RW1 + d16	DIV A, @RW1	DIV A, @RW1 + d16	DIVW A, @RW1	DIVW A, @RW1 + d16
+A	MULU A, @RW2	MULU A, @RW2 + d16	MULUW A, @RW2	MULUW A, @RW2 + d16	MUL A, @RW2	MUL A, @RW2 + d16	MULW A, @RW2	MULW A, @RW2 + d16	DIVU A, @RW2	DIVU A, @RW2 + d16	DIVUW A, @RW2	DIVUW A, @RW2 + d16	DIV A, @RW2	DIV A, @RW2 + d16	DIVW A, @RW2	DIVW A, @RW2 + d16
+B	MULU A, @RW3	MULU A, @RW3 + d16	MULUW A, @RW3	MULUW A, @RW3 + d16	MUL A, @RW3	MUL A, @RW3 + d16	MULW A, @RW3	MULW A, @RW3 + d16	DIVU A, @RW3	DIVU A, @RW3 + d16	DIVUW A, @RW3	DIVUW A, @RW3 + d16	DIV A, @RW3	DIV A, @RW3 + d16	DIVW A, @RW3	DIVW A, @RW3 + d16
+C	MULU A, @RW0+	MULU A, @RW0+RW7	MULUW A, @RW0+	MULUW A, @RW0+RW7	MUL A, @RW0+	MUL A, @RW0+RW7	MULW A, @RW0+	MULW A, @RW0+RW7	DIVU A, @RW0+	DIVU A, @RW0+RW7	DIVUW A, @RW0+	DIVUW A, @RW0+RW7	DIV A, @RW0+	DIV A, @RW0+RW7	DIVW A, @RW0+	DIVW A, @RW0+RW7
+D	MULU A, @RW1+	MULU A, @RW1+RW7	MULUW A, @RW1+	MULUW A, @RW1+RW7	MUL A, @RW1+	MUL A, @RW1+RW7	MULW A, @RW1+	MULW A, @RW1+RW7	DIVU A, @RW1+	DIVU A, @RW1+RW7	DIVUW A, @RW1+	DIVUW A, @RW1+RW7	DIV A, @RW1+	DIV A, @RW1+RW7	DIVW A, @RW1+	DIVW A, @RW1+RW7
+E	MULU A, @RW2+	MULU A, @PC + d16	MULUW A, @RW2+	MULUW A, @PC + d16	MUL A, @RW2+	MUL A, @PC + d16	MULW A, @RW2+	MULW A, @PC + d16	DIVU A, @RW2+	DIVU A, @PC + d16	DIVUW A, @RW2+	DIVUW A, @PC + d16	DIV A, @RW2+	DIV A, @PC + d16	DIVW A, @RW2+	DIVW A, @PC + d16
+F	MULU A, @RW3+	MULU A, addr16	MULUW A, @RW3+	MULUW A, addr16	MUL A, @RW3+	MUL A, addr16	MULW A, @RW3+	MULW A, addr16	DIVU A, @RW3+	DIVU A, addr16	DIVUW A, @RW3+	DIVUW A, addr16	DIV A, @RW3+	DIV A, addr16	DIVW A, @RW3+	DIVW A, addr16

Table 37-38. MOVEA RWi, ea命令 (第1バイト = 79_H)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	MOVEA RW0,RW0	MOVEA RW0, @RW0 + d8	MOVEA RW1,RW0	MOVEA RW1, @RW0 + d8	MOVEA RW2,RW0	MOVEA RW2, @RW0 + d8	MOVEA RW3,RW0	MOVEA RW3, @RW0 + d8	MOVEA RW4,RW0	MOVEA RW4, @RW0 + d8	MOVEA RW5,RW0	MOVEA RW5, @RW0 + d8	MOVEA RW6,RW0	MOVEA RW6, @RW0 + d8	MOVEA RW7,RW0	MOVEA RW7, @RW0 + d8
+1	MOVEA RW0,RW1	MOVEA RW0, @RW1 + d8	MOVEA RW1,RW1	MOVEA RW1, @RW1 + d8	MOVEA RW2,RW1	MOVEA RW2, @RW1 + d8	MOVEA RW3,RW1	MOVEA RW3, @RW1 + d8	MOVEA RW4,RW1	MOVEA RW4, @RW1 + d8	MOVEA RW5,RW1	MOVEA RW5, @RW1 + d8	MOVEA RW6,RW1	MOVEA RW6, @RW1 + d8	MOVEA RW7,RW1	MOVEA RW7, @RW1 + d8
+2	MOVEA RW0,RW2	MOVEA RW0, @RW2 + d8	MOVEA RW1,RW2	MOVEA RW1, @RW2 + d8	MOVEA RW2,RW2	MOVEA RW2, @RW2 + d8	MOVEA RW3,RW2	MOVEA RW3, @RW2 + d8	MOVEA RW4,RW2	MOVEA RW4, @RW2 + d8	MOVEA RW5,RW2	MOVEA RW5, @RW2 + d8	MOVEA RW6,RW2	MOVEA RW6, @RW2 + d8	MOVEA RW7,RW2	MOVEA RW7, @RW2 + d8
+3	MOVEA RW0,RW3	MOVEA RW0, @RW3 + d8	MOVEA RW1,RW3	MOVEA RW1, @RW3 + d8	MOVEA RW2,RW3	MOVEA RW2, @RW3 + d8	MOVEA RW3,RW3	MOVEA RW3, @RW3 + d8	MOVEA RW4,RW3	MOVEA RW4, @RW3 + d8	MOVEA RW5,RW3	MOVEA RW5, @RW3 + d8	MOVEA RW6,RW3	MOVEA RW6, @RW3 + d8	MOVEA RW7,RW3	MOVEA RW7, @RW3 + d8
+4	MOVEA RW0,RW4	MOVEA RW0, @RW4 + d8	MOVEA RW1,RW4	MOVEA RW1, @RW4 + d8	MOVEA RW2,RW4	MOVEA RW2, @RW4 + d8	MOVEA RW3,RW4	MOVEA RW3, @RW4 + d8	MOVEA RW4,RW4	MOVEA RW4, @RW4 + d8	MOVEA RW5,RW4	MOVEA RW5, @RW4 + d8	MOVEA RW6,RW4	MOVEA RW6, @RW4 + d8	MOVEA RW7,RW4	MOVEA RW7, @RW4 + d8
+5	MOVEA RW0,RW5	MOVEA RW0, @RW5 + d8	MOVEA RW1,RW5	MOVEA RW1, @RW5 + d8	MOVEA RW2,RW5	MOVEA RW2, @RW5 + d8	MOVEA RW3,RW5	MOVEA RW3, @RW5 + d8	MOVEA RW4,RW5	MOVEA RW4, @RW5 + d8	MOVEA RW5,RW5	MOVEA RW5, @RW5 + d8	MOVEA RW6,RW5	MOVEA RW6, @RW5 + d8	MOVEA RW7,RW5	MOVEA RW7, @RW5 + d8
+6	MOVEA RW0,RW6	MOVEA RW0, @RW6 + d8	MOVEA RW1,RW6	MOVEA RW1, @RW6 + d8	MOVEA RW2,RW6	MOVEA RW2, @RW6 + d8	MOVEA RW3,RW6	MOVEA RW3, @RW6 + d8	MOVEA RW4,RW6	MOVEA RW4, @RW6 + d8	MOVEA RW5,RW6	MOVEA RW5, @RW6 + d8	MOVEA RW6,RW6	MOVEA RW6, @RW6 + d8	MOVEA RW7,RW6	MOVEA RW7, @RW6 + d8
+7	MOVEA RW0,RW7	MOVEA RW0, @RW7 + d8	MOVEA RW1,RW7	MOVEA RW1, @RW7 + d8	MOVEA RW2,RW7	MOVEA RW2, @RW7 + d8	MOVEA RW3,RW7	MOVEA RW3, @RW7 + d8	MOVEA RW4,RW7	MOVEA RW4, @RW7 + d8	MOVEA RW5,RW7	MOVEA RW5, @RW7 + d8	MOVEA RW6,RW7	MOVEA RW6, @RW7 + d8	MOVEA RW7,RW7	MOVEA RW7, @RW7 + d8
+8	MOVEA RW0,@RW0	MOVEA RW0, @RW0 + d16	MOVEA RW1,@RW0	MOVEA RW1, @RW0 + d16	MOVEA RW2,@RW0	MOVEA RW2, @RW0 + d16	MOVEA RW3,@RW0	MOVEA RW3, @RW0 + d16	MOVEA RW4,@RW0	MOVEA RW4, @RW0 + d16	MOVEA RW5,@RW0	MOVEA RW5, @RW0 + d16	MOVEA RW6,@RW0	MOVEA RW6, @RW0 + d16	MOVEA RW7,@RW0	MOVEA RW7, @RW0 + d16
+9	MOVEA RW0,@RW1	MOVEA RW0, @RW1 + d16	MOVEA RW1,@RW1	MOVEA RW1, @RW1 + d16	MOVEA RW2,@RW1	MOVEA RW2, @RW1 + d16	MOVEA RW3,@RW1	MOVEA RW3, @RW1 + d16	MOVEA RW4,@RW1	MOVEA RW4, @RW1 + d16	MOVEA RW5,@RW1	MOVEA RW5, @RW1 + d16	MOVEA RW6,@RW1	MOVEA RW6, @RW1 + d16	MOVEA RW7,@RW1	MOVEA RW7, @RW1 + d16
+A	MOVEA RW0,@RW2	MOVEA RW0, @RW2 + d16	MOVEA RW1,@RW2	MOVEA RW1, @RW2 + d16	MOVEA RW2,@RW2	MOVEA RW2, @RW2 + d16	MOVEA RW3,@RW2	MOVEA RW3, @RW2 + d16	MOVEA RW4,@RW2	MOVEA RW4, @RW2 + d16	MOVEA RW5,@RW2	MOVEA RW5, @RW2 + d16	MOVEA RW6,@RW2	MOVEA RW6, @RW2 + d16	MOVEA RW7,@RW2	MOVEA RW7, @RW2 + d16
+B	MOVEA RW0,@RW3	MOVEA RW0, @RW3 + d16	MOVEA RW1,@RW3	MOVEA RW1, @RW3 + d16	MOVEA RW2,@RW3	MOVEA RW2, @RW3 + d16	MOVEA RW3,@RW3	MOVEA RW3, @RW3 + d16	MOVEA RW4,@RW3	MOVEA RW4, @RW3 + d16	MOVEA RW5,@RW3	MOVEA RW5, @RW3 + d16	MOVEA RW6,@RW3	MOVEA RW6, @RW3 + d16	MOVEA RW7,@RW3	MOVEA RW7, @RW3 + d16
+C	MOVEA R W0,@RW0+	MOVEA RW0, @RW0+RW7	MOVEA R W1,@RW0+	MOVEA RW1, @RW0+RW7	MOVEA R W2,@RW0+	MOVEA RW2, @RW0+RW7	MOVEA R W3,@RW0+	MOVEA RW3, @RW0+RW7	MOVEA R W4,@RW0+	MOVEA RW4, @RW0+RW7	MOVEA R W5,@RW0+	MOVEA RW5, @RW0+RW7	MOVEA R W6,@RW0+	MOVEA RW6, @RW0+RW7	MOVEA R W7,@RW0+	MOVEA RW7, @RW0+RW7
+D	MOVEA R W0,@RW1+	MOVEA RW0, @RW1+RW7	MOVEA R W1,@RW1+	MOVEA RW1, @RW1+RW7	MOVEA R W2,@RW1+	MOVEA RW2, @RW1+RW7	MOVEA R W3,@RW1+	MOVEA RW3, @RW1+RW7	MOVEA R W4,@RW1+	MOVEA RW4, @RW1+RW7	MOVEA R W5,@RW1+	MOVEA RW5, @RW1+RW7	MOVEA R W6,@RW1+	MOVEA RW6, @RW1+RW7	MOVEA R W7,@RW1+	MOVEA RW7, @RW1+RW7
+E	MOVEA R W0,@RW2+	MOVEA RW0, @PC + d16	MOVEA R W1,@RW2+	MOVEA RW1, @PC + d16	MOVEA R W2,@RW2+	MOVEA RW2, @PC + d16	MOVEA R W3,@RW2+	MOVEA RW3, @PC + d16	MOVEA R W4,@RW2+	MOVEA RW4, @PC + d16	MOVEA R W5,@RW2+	MOVEA RW5, @PC + d16	MOVEA R W6,@RW2+	MOVEA RW6, @PC + d16	MOVEA R W7,@RW2+	MOVEA RW7, @PC + d16
+F	MOVEA R W0,@RW3+	MOVEA RW0, addr16	MOVEA R W1,@RW3+	MOVEA RW1, addr16	MOVEA R W2,@RW3+	MOVEA RW2, addr16	MOVEA R W3,@RW3+	MOVEA RW3, addr16	MOVEA R W4,@RW3+	MOVEA RW4, addr16	MOVEA R W5,@RW3+	MOVEA RW5, addr16	MOVEA R W6,@RW3+	MOVEA RW6, addr16	MOVEA R W7,@RW3+	MOVEA RW7, addr16

Table 37-39. MOV Ri, ea命令 (第1バイト = $7A_H$)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	MOV R0, R0,R0	MOV R0, @RW0 + d8	MOV R1,R0	MOV R1, @RW0 + d8	MOV R2,R0	MOV R2, @RW0 + d8	MOV R3,R0	MOV R3, @RW0 + d8	MOV R4,R0	MOV R4, @RW0 + d8	MOV R5,R0	MOV R5, @RW0 + d8	MOV R6,R0	MOV R6, @RW0 + d8	MOV R7,R0	MOV R7, @RW0 + d8
+1	MOV R0,R1	MOV R0, @RW1 + d8	MOV R1,R1	MOV R1, @RW1 + d8	MOV R2,R1	MOV R2, @RW1 + d8	MOV R3,R1	MOV R3, @RW1 + d8	MOV R4,R1	MOV R4, @RW1 + d8	MOV R5,R1	MOV R5, @RW1 + d8	MOV R6,R1	MOV R6, @RW1 + d8	MOV R7,R1	MOV R7, @RW1 + d8
+2	MOV R0,R2	MOV R0, @RW2 + d8	MOV R1,R2	MOV R1, @RW2 + d8	MOV R2,R2	MOV R2, @RW2 + d8	MOV R3,R2	MOV R3, @RW2 + d8	MOV R4,R2	MOV R4, @RW2 + d8	MOV R5,R2	MOV R5, @RW2 + d8	MOV R6,R2	MOV R6, @RW2 + d8	MOV R7,R2	MOV R7, @RW2 + d8
+3	MOV R0,R3	MOV R0, @RW3 + d8	MOV R1,R3	MOV R1, @RW3 + d8	MOV R2,R3	MOV R2, @RW3 + d8	MOV R3,R3	MOV R3, @RW3 + d8	MOV R4,R3	MOV R4, @RW3 + d8	MOV R5,R3	MOV R5, @RW3 + d8	MOV R6,R3	MOV R6, @RW3 + d8	MOV R7,R3	MOV R7, @RW3 + d8
+4	MOV R0,R4	MOV R0, @RW4 + d8	MOV R1,R4	MOV R1, @RW4 + d8	MOV R2,R4	MOV R2, @RW4 + d8	MOV R3,R4	MOV R3, @RW4 + d8	MOV R4,R4	MOV R4, @RW4 + d8	MOV R5,R4	MOV R5, @RW4 + d8	MOV R6,R4	MOV R6, @RW4 + d8	MOV R7,R4	MOV R7, @RW4 + d8
+5	MOV R0,R5	MOV R0, @RW5 + d8	MOV R1,R5	MOV R1, @RW5 + d8	MOV R2,R5	MOV R2, @RW5 + d8	MOV R3,R5	MOV R3, @RW5 + d8	MOV R4,R5	MOV R4, @RW5 + d8	MOV R5,R5	MOV R5, @RW5 + d8	MOV R6,R5	MOV R6, @RW5 + d8	MOV R7,R5	MOV R7, @RW5 + d8
+6	MOV R0,R6	MOV R0, @RW6 + d8	MOV R1,R6	MOV R1, @RW6 + d8	MOV R2,R6	MOV R2, @RW6 + d8	MOV R3,R6	MOV R3, @RW6 + d8	MOV R4,R6	MOV R4, @RW6 + d8	MOV R5,R6	MOV R5, @RW6 + d8	MOV R6,R6	MOV R6, @RW6 + d8	MOV R7,R6	MOV R7, @RW6 + d8
+7	MOV R0,R7	MOV R0, @RW7 + d8	MOV R1,R7	MOV R1, @RW7 + d8	MOV R2,R7	MOV R2, @RW7 + d8	MOV R3,R7	MOV R3, @RW7 + d8	MOV R4,R7	MOV R4, @RW7 + d8	MOV R5,R7	MOV R5, @RW7 + d8	MOV R6,R7	MOV R6, @RW7 + d8	MOV R7,R7	MOV R7, @RW7 + d8
+8	MOV R0,@RW0	MOV R0, @RW0 + d16	MOV R1,@RW0	MOV R1, @RW0 + d16	MOV R2,@RW0	MOV R2, @RW0 + d16	MOV R3,@RW0	MOV R3, @RW0 + d16	MOV R4,@RW0	MOV R4, @RW0 + d16	MOV R5,@RW0	MOV R5, @RW0 + d16	MOV R6,@RW0	MOV R6, @RW0 + d16	MOV R7,@RW0	MOV R7, @RW0 + d16
+9	MOV R0,@RW1	MOV R0, @RW1 + d16	MOV R1,@RW1	MOV R1, @RW1 + d16	MOV R2,@RW1	MOV R2, @RW1 + d16	MOV R3,@RW1	MOV R3, @RW1 + d16	MOV R4,@RW1	MOV R4, @RW1 + d16	MOV R5,@RW1	MOV R5, @RW1 + d16	MOV R6,@RW1	MOV R6, @RW1 + d16	MOV R7,@RW1	MOV R7, @RW1 + d16
+A	MOV R0,@RW2	MOV R0, @RW2 + d16	MOV R1,@RW2	MOV R1, @RW2 + d16	MOV R2,@RW2	MOV R2, @RW2 + d16	MOV R3,@RW2	MOV R3, @RW2 + d16	MOV R4,@RW2	MOV R4, @RW2 + d16	MOV R5,@RW2	MOV R5, @RW2 + d16	MOV R6,@RW2	MOV R6, @RW2 + d16	MOV R7,@RW2	MOV R7, @RW2 + d16
+B	MOV R0,@RW3	MOV R0, @RW3 + d16	MOV R1,@RW3	MOV R1, @RW3 + d16	MOV R2,@RW3	MOV R2, @RW3 + d16	MOV R3,@RW3	MOV R3, @RW3 + d16	MOV R4,@RW3	MOV R4, @RW3 + d16	MOV R5,@RW3	MOV R5, @RW3 + d16	MOV R6,@RW3	MOV R6, @RW3 + d16	MOV R7,@RW3	MOV R7, @RW3 + d16
+C	MOV R0, @RW0+	MOV R0, @RW0 + RW7	MOV R1, @RW0+	MOV R1, @RW0 + RW7	MOV R2, @RW0+	MOV R2, @RW0 + RW7	MOV R3, @RW0+	MOV R3, @RW0 + RW7	MOV R4, @RW0+	MOV R4, @RW0 + RW7	MOV R5, @RW0+	MOV R5, @RW0 + RW7	MOV R6, @RW0+	MOV R6, @RW0 + RW7	MOV R7, @RW0+	MOV R7, @RW0 + RW7
+D	MOV R0, @RW1+	MOV R0, @RW1 + RW7	MOV R1, @RW1+	MOV R1, @RW1 + RW7	MOV R2, @RW1+	MOV R2, @RW1 + RW7	MOV R3, @RW1+	MOV R3, @RW1 + RW7	MOV R4, @RW1+	MOV R4, @RW1 + RW7	MOV R5, @RW1+	MOV R5, @RW1 + RW7	MOV R6, @RW1+	MOV R6, @RW1 + RW7	MOV R7, @RW1+	MOV R7, @RW1 + RW7
+E	MOV R0, @RW2+	MOV R0, @PC + d16	MOV R1, @RW2+	MOV R1, @PC + d16	MOV R2, @RW2+	MOV R2, @PC + d16	MOV R3, @RW2+	MOV R3, @PC + d16	MOV R4, @RW2+	MOV R4, @PC + d16	MOV R5, @RW2+	MOV R5, @PC + d16	MOV R6, @RW2+	MOV R6, @PC + d16	MOV R7, @RW2+	MOV R7, @PC + d16
+F	MOV R0, @RW3+	MOV R0, addr16	MOV R1, @RW3+	MOV R1, addr16	MOV R2, @RW3+	MOV R2, addr16	MOV R3, @RW3+	MOV R3, addr16	MOV R4, @RW3+	MOV R4, addr16	MOV R5, @RW3+	MOV R5, addr16	MOV R6, @RW3+	MOV R6, addr16	MOV R7, @RW3+	MOV R7, addr16

Table 37-40. MOVW RWi, ea命令 (第1バイト = $7B_H$)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	MOVW RW0, RW0	MOVW RW0, @RW0 + d8	MOVW RW1, RW0	MOVW RW1 @RW0 + d8	MOVW RW2, RW0	MOVW RW2, @RW0 + d8	MOVW RW3, RW0	MOVW RW3, @RW0 + d8	MOVW RW4, RW0	MOVW RW4, @RW0 + d8	MOVW RW5, RW0	MOVW RW5, @RW0 + d8	MOVW RW6, RW0	MOVW RW6, @RW0 + d8	MOVW RW7, RW0	MOVW RW7, @RW0 + d8
+1	MOVW RW0, RW1	MOVW RW0, @RW1 + d8	MOVW RW1, RW1	MOVW RW1 @RW1 + d8	MOVW RW2, RW1	MOVW RW2, @RW1 + d8	MOVW RW3, RW1	MOVW RW3, @RW1 + d8	MOVW RW4, RW1	MOVW RW4, @RW1 + d8	MOVW RW5, RW1	MOVW RW5, @RW1 + d8	MOVW RW6, RW1	MOVW RW6, @RW1 + d8	MOVW RW7, RW1	MOVW RW7, @RW1 + d8
+2	MOVW RW0, RW2	MOVW RW0, @RW2 + d8	MOVW RW1, RW2	MOVW RW1 @RW2 + d8	MOVW RW2, RW2	MOVW RW2, @RW2 + d8	MOVW RW3, RW2	MOVW RW3, @RW2 + d8	MOVW RW4, RW2	MOVW RW4, @RW2 + d8	MOVW RW5, RW2	MOVW RW5, @RW2 + d8	MOVW RW6, RW2	MOVW RW6, @RW2 + d8	MOVW RW7, RW2	MOVW RW7, @RW2 + d8
+3	MOVW RW0, RW3	MOVW RW0, @RW3 + d8	MOVW RW1, RW3	MOVW RW1, @RW3 + d8	MOVW RW2, RW3	MOVW RW2, @RW3 + d8	MOVW RW3, RW3	MOVW RW3, @RW3 + d8	MOVW RW4, RW3	MOVW RW4, @RW3 + d8	MOVW RW5, RW3	MOVW RW5, @RW3 + d8	MOVW RW6, RW3	MOVW RW6, @RW3 + d8	MOVW RW7, RW3	MOVW RW7, @RW3 + d8
+4	MOVW RW0, RW4	MOVW RW0, @RW4 + d8	MOVW RW1, RW4	MOVW RW1, @RW4 + d8	MOVW RW2, RW4	MOVW RW2, @RW4 + d8	MOVW RW3, RW4	MOVW RW3, @RW4 + d8	MOVW RW4, RW4	MOVW RW4, @RW4 + d8	MOVW RW5, RW4	MOVW RW5, @RW4 + d8	MOVW RW6, RW4	MOVW RW6, @RW4 + d8	MOVW RW7, RW4	MOVW RW7, @RW4 + d8
+5	MOVW RW0, RW5	MOVW RW0, @RW5 + d8	MOVW RW1, RW5	MOVW RW1, @RW5 + d8	MOVW RW2, RW5	MOVW RW2, @RW5 + d8	MOVW RW3, RW5	MOVW RW3, @RW5 + d8	MOVW RW4, RW5	MOVW RW4, @RW5 + d8	MOVW RW5, RW5	MOVW RW5, @RW5 + d8	MOVW RW6, RW5	MOVW RW6, @RW5 + d8	MOVW RW7, RW5	MOVW RW7, @RW5 + d8
+6	MOVW RW0, RW6	MOVW RW0, @RW6 + d8	MOVW RW1, RW6	MOVW RW1, @RW6 + d8	MOVW RW2, RW6	MOVW RW2, @RW6 + d8	MOVW RW3, RW6	MOVW RW3, @RW6 + d8	MOVW RW4, RW6	MOVW RW4, @RW6 + d8	MOVW RW5, RW6	MOVW RW5, @RW6 + d8	MOVW RW6, RW6	MOVW RW6, @RW6 + d8	MOVW RW7, RW6	MOVW RW7, @RW6 + d8
+7	MOVW RW0, RW7	MOVW RW0, @RW7 + d8	MOVW RW1, RW7	MOVW RW1 @RW7 + d8	MOVW RW2, RW7	MOVW RW2, @RW7 + d8	MOVW RW3, RW7	MOVW RW3, @RW7 + d8	MOVW RW4, RW7	MOVW RW4, @RW7 + d8	MOVW RW5, RW7	MOVW RW5, @RW7 + d8	MOVW RW6, RW7	MOVW RW6, @RW7 + d8	MOVW RW7, RW7	MOVW RW7, @RW7 + d8
+8	MOVW RW0, @RW0	MOVW RW0, @RW0 + d16	MOVW RW1, @RW0	MOVW RW1, @RW0 + d16	MOVW RW2, @RW0	MOVW RW2, @RW0 + d16	MOVW RW3,@RW0	MOVW RW3, @RW0 + d16	MOVW RW4,@RW0	MOVW RW4, @RW0 + d16	MOVW RW5, @RW0	MOVW RW5, @RW0 + d16	MOVW RW6,@RW0	MOVW RW6, @RW0 + d16	MOVW RW7,@RW0	MOVW RW7, @RW0 + d16
+9	MOVW RW0, @RW1	MOVW RW0, @RW1 + d16	MOVW RW1, @RW1	MOVW RW1, @RW1 + d16	MOVW RW2, @RW1	MOVW RW2, @RW1 + d16	MOVW RW3,@RW1	MOVW RW3, @RW1 + d16	MOVW RW4,@RW1	MOVW RW4, @RW1 + d16	MOVW RW5, @RW1	MOVW RW5, @RW1 + d16	MOVW RW6,@RW1	MOVW RW6, @RW1 + d16	MOVW RW7,@RW1	MOVW RW7, @RW1 + d16
+A	MOVW RW0, @RW2	MOVW RW0, @RW2 + d16	MOVW RW1, @RW2	MOVW RW1, @RW2 + d16	MOVW RW2, @RW2	MOVW RW2, @RW2 + d16	MOVW RW3, @RW2	MOVW RW3, @RW2 + d16	MOVW RW4,@RW2	MOVW RW4, @RW2 + d16	MOVW RW5, @RW2	MOVW RW5, @RW2 + d16	MOVW RW6, @RW2	MOVW RW6, @RW2 + d16	MOVW RW7, @RW2	MOVW RW7, @RW2 + d16
+B	MOVW RW0, @RW3	MOVW RW0, @RW3 + d16	MOVW RW1, @RW3	MOVW RW1, @RW3 + d16	MOVW RW2, @RW3	MOVW RW2, @RW3 + d16	MOVW RW3,@RW3	MOVW RW3, @RW3 + d16	MOVW RW4,@RW3	MOVW RW4, @RW3 + d16	MOVW RW5, @RW3	MOVW RW5, @RW3 + d16	MOVW RW6,@RW3	MOVW RW6, @RW3 + d16	MOVW RW7, @RW3	MOVW RW7, @RW3 + d16
+C	MOVW RW0, @RW0+	MOVW RW0, @RW0 + RW7	MOVW RW1, @RW0+	MOVW RW1, @RW0 + RW7	MOVW RW2, @RW0+	MOVW RW2, @RW0 + RW7	MOVW RW3,@RW0+	MOVW RW3, @RW0 + RW7	MOVW RW4,@RW0+	MOVW RW4, @RW0 + RW7	MOVW RW5, @RW0+	MOVW RW5, @RW0 + RW7	MOVW RW6,@RW0+	MOVW RW6, @RW0 + RW7	MOVW RW7, @RW0+	MOVW RW7, @RW0 + RW7
+D	MOVW RW0, @RW1+	MOVW RW0, @RW1 + RW7	MOVW RW1, @RW1+	MOVW RW1, @RW1 + RW7	MOVW RW2, @RW1+	MOVW RW2, @RW1 + RW7	MOVW RW3,@RW1+	MOVW RW3, @RW1 + RW7	MOVW RW4,@RW1+	MOVW RW4, @RW1 + RW7	MOVW RW5, @RW1+	MOVW RW5, @RW1 + RW7	MOVW RW6,@RW1+	MOVW RW6, @RW1 + RW7	MOVW RW7, @RW1+	MOVW RW7, @RW1 + RW7
+E	MOVW RW0, @RW2+	MOVW RW0, @PC + d16	MOVW RW1,@RW2+	MOVW RW1, @PC + d16	MOVW RW2, @RW2+	MOVW RW2, @PC + d16	MOVW RW3,@RW2+	MOVW RW3, @PC + d16	MOVW RW4,@RW2+	MOVW RW4, @PC + d16	MOVW RW5, @RW2+	MOVW RW5, @PC + d16	MOVW RW6,@RW2+	MOVW RW6, @PC + d16	MOVW RW7, @RW2+	MOVW RW7, @PC + d16
+F	MOVW RW0, @RW3+	MOVW RW0, addr16	MOVW RW1,@RW3+	MOVW RW1, addr16	MOVW RW2, @RW3+	MOVW RW2, addr16	MOVW RW3,@RW3+	MOVW RW3, addr16	MOVW RW4,@RW3+	MOVW RW4, addr16	MOVW RW5, @RW3+	MOVW RW5, addr16	MOVW RW6,@RW3+	MOVW RW6, addr16	MOVW RW7, @RW3+	MOVW RW7 addr16

Table 37-41. MOV ea, Ri 命令 (第1バイト = $7C_H$)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	MOV R0,R0	MOV @R W0 + d8,R0	MOV R0,R1	MOV @R W0 + d8,R1	MOV R0,R2	MOV @R W0 + d8,R2	MOV R0,R3	MOV @R W0 + d8,R3	MOV R0,R4	MOV @R W0 + d8,R4	MOV R0,R5	MOV @R W0 + d8,R5	MOV R0,R6	MOV @R W0 + d8,R6	MOV R0,R7	MOV @R W0 + d8,R7
+1	MOV R1,R0	MOV @R W1 + d8,R0	MOV R1,R1	MOV @R W1 + d8,R1	MOV R1,R2	MOV @R W1 + d8,R2	MOV R1,R3	MOV @R W1 + d8,R3	MOV R1,R4	MOV @R W1 + d8,R4	MOV R1,R5	MOV @R W1 + d8,R5	MOV R1,R6	MOV @R W1 + d8,R6	MOV R1,R7	MOV @R W1 + d8,R7
+2	MOV R2,R0	MOV @R W2 + d8,R0	MOV R2,R1	MOV @R W2 + d8,R1	MOV R2,R2	MOV @R W2 + d8,R2	MOV R2,R3	MOV @R W2 + d8,R3	MOV R2,R4	MOV @R W2 + d8,R4	MOV R2,R5	MOV @R W2 + d8,R5	MOV R2,R6	MOV @R W2 + d8,R6	MOV R2,R7	MOV @R W2 + d8,R7
+3	MOV R3,R0	MOV @R W3 + d8,R0	MOV R3,R1	MOV @R W3 + d8,R1	MOV R3,R2	MOV @R W3 + d8,R2	MOV R3,R3	MOV @R W3 + d8,R3	MOV R3,R4	MOV @R W3 + d8,R4	MOV R3,R5	MOV @R W3 + d8,R5	MOV R3,R6	MOV @R W3 + d8,R6	MOV R3,R7	MOV @R W3 + d8,R7
+4	MOV R4,R0	MOV @R W4 + d8,R0	MOV R4,R1	MOV @R W4 + d8,R1	MOV R4,R2	MOV @R W4 + d8,R2	MOV R4,R3	MOV @R W4 + d8,R3	MOV R4,R4	MOV @R W4 + d8,R4	MOV R4,R5	MOV @R W4 + d8,R5	MOV R4,R6	MOV @R W4 + d8,R6	MOV R4,R7	MOV @R W4 + d8,R7
+5	MOV R5,R0	MOV @R W5 + d8,R0	MOV R5,R1	MOV @R W5 + d8,R1	MOV R5,R2	MOV @R W5 + d8,R2	MOV R5,R3	MOV @R W5 + d8,R3	MOV R5,R4	MOV @R W5 + d8,R4	MOV R5,R5	MOV @R W5 + d8,R5	MOV R5,R6	MOV @R W5 + d8,R6	MOV R5,R7	MOV @R W5 + d8,R7
+6	MOV R6,R0	MOV @R W6 + d8,R0	MOV R6,R1	MOV @R W6 + d8,R1	MOV R6,R2	MOV @R W6 + d8,R2	MOV R6,R3	MOV @R W6 + d8,R3	MOV R6,R4	MOV @R W6 + d8,R4	MOV R6,R5	MOV @R W6 + d8,R5	MOV R6,R6	MOV @R W6 + d8,R6	MOV R6,R7	MOV @R W6 + d8,R7
+7	MOV R7,R0	MOV @R W7 + d8,R0	MOV R7,R1	MOV @R W7 + d8,R1	MOV R7,R2	MOV @R W7 + d8,R2	MOV R7,R3	MOV @R W7 + d8,R3	MOV R7,R4	MOV @R W7 + d8,R4	MOV R7,R5	MOV @R W7 + d8,R5	MOV R7,R6	MOV @R W7 + d8,R6	MOV R7,R7	MOV @R W7 + d8,R7
+8	MOV @RW0,R0	MOV @RW 0 + d16,R0	MOV @RW0,R1	MOV @RW 0 + d16,R1	MOV @RW0,R2	MOV @RW 0 + d16,R2	MOV @RW0,R3	MOV @RW 0 + d16,R3	MOV @RW0,R4	MOV @RW 0 + d16,R4	MOV @RW0,R5	MOV @RW 0 + d16,R5	MOV @RW0,R6	MOV @RW 0 + d16,R6	MOV @RW0,R7	MOV @RW 0 + d16,R7
+9	MOV @RW1,R0	MOV @RW 1 + d16,R0	MOV @RW1,R1	MOV @RW 1 + d16,R1	MOV @RW1,R2	MOV @RW 1 + d16,R2	MOV @RW1,R3	MOV @RW 1 + d16,R3	MOV @RW1,R4	MOV @RW 1 + d16,R4	MOV @RW1,R5	MOV @RW 1 + d16,R5	MOV @RW1,R6	MOV @RW 1 + d16,R6	MOV @RW1,R7	MOV @RW 1 + d16,R7
+A	MOV @RW2,R0	MOV @RW 2 + d16,R0	MOV @RW2,R1	MOV @RW 2 + d16,R1	MOV @RW2,R2	MOV @RW 2 + d16,R2	MOV @RW2,R3	MOV @RW 2 + d16,R3	MOV @RW2,R4	MOV @RW 2 + d16,R4	MOV @RW2,R5	MOV @RW 2 + d16,R5	MOV @RW2,R6	MOV @RW 2 + d16,R6	MOV @RW2,R7	MOV @RW 2 + d16,R7
+B	MOV @RW3,R0	MOV @RW 3 + d16,R0	MOV @RW3,R1	MOV @RW 3 + d16,R1	MOV @RW3,R2	MOV @RW 3 + d16,R2	MOV @RW3,R3	MOV @RW 3 + d16,R3	MOV @RW3,R4	MOV @RW 3 + d16,R4	MOV @RW3,R5	MOV @RW 3 + d16,R5	MOV @RW3,R6	MOV @RW 3 + d16,R6	MOV @RW3,R7	MOV @RW 3 + d16,R7
+C	MOV @RW0+,R0	MOV @RW 0 + RW7,R0	MOV @RW0+,R1	MOV @RW 0 + RW7,R1	MOV @RW0+,R2	MOV @RW 0 + RW7,R2	MOV @RW0+,R3	MOV @RW 0 + RW7,R3	MOV @RW0+,R4	MOV @RW 0 + RW7,R4	MOV @RW0+,R5	MOV @RW 0 + RW7,R5	MOV @RW0+,R6	MOV @RW 0 + RW7,R6	MOV @RW0+,R7	MOV @RW 0 + RW7,R7
+D	MOV @RW1+,R0	MOV @RW 1 + RW7,R0	MOV @RW1+,R1	MOV @RW 1 + RW7,R1	MOV @RW1+,R2	MOV @RW 1 + RW7,R2	MOV @RW1+,R3	MOV @RW 1 + RW7,R3	MOV @RW1+,R4	MOV @RW 1 + RW7,R4	MOV @RW1+,R5	MOV @RW 1 + RW7,R5	MOV @RW1+,R6	MOV @RW 1 + RW7,R6	MOV @RW1+,R7	MOV @RW 1 + RW7,R7
+E	MOV @RW2+,R0	MOV PC + d16,R0	MOV @RW2+,R1	MOV PC + d16,R1	MOV @RW2+,R2	MOV PC + d16,R2	MOV @RW2+,R3	MOV PC + d16,R3	MOV @RW2+,R4	MOV PC + d16,R4	MOV @RW2+,R5	MOV PC + d16,R5	MOV @RW2+,R6	MOV PC + d16,R6	MOV @RW2+,R7	MOV PC + d16,R7
+F	MOV @RW3+,R0	MOV addr16,R0	MOV @RW3+,R1	MOV addr16,R1	MOV @RW3+,R2	MOV addr16,R2	MOV @RW3+,R3	MOV addr16,R3	MOV @RW3+,R4	MOV addr16,R4	MOV @RW3+,R5	MOV addr16,R5	MOV @RW3+,R6	MOV addr16,R6	MOV @RW3+,R7	MOV d adr16,R7

Table 37-42. MOVW ea, Rwi 命令 (第1バイト = $7D_H$)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	MOVW RW0,RW0	MOVW @RW 0 + d8,RW0	MOVW RW0,RW1	MOVW @RW 0 + d8,RW1	MOVW RW0,RW2	MOVW @RW 0 + d8,RW2	MOVW RW0,RW3	MOVW @RW 0 + d8,RW3	MOVW R W0,RW4	MOVW @RW 0 + d8,RW4	MOVW R W0,RW5	MOVW @RW 0 + d8,RW5	MOVW R W0,RW6	MOVW @RW 0 + d8,RW6	MOVW R W0,RW7	MOVW @RW 0 + d8,RW7
+1	MOVW RW1,RW0	MOVW @RW 1 + d8,RW0	MOVW RW1,RW1	MOVW @RW 1 + d8,RW1	MOVW RW1,RW2	MOVW @RW 1 + d8,RW2	MOVW RW1,RW3	MOVW @RW 1 + d8,RW3	MOVW RW1,RW4	MOVW @RW 1 + d8,RW4	MOVW RW1,RW5	MOVW @RW 1 + d8,RW5	MOVW RW1,RW6	MOVW @RW 1 + d8,RW6	MOVW RW1,RW7	MOVW @RW 1 + d8,RW7
+2	MOVW RW2,RW0	MOVW @RW 2 + d8,RW0	MOVW RW2,RW1	MOVW @RW 2 + d8,RW1	MOVW RW2,RW2	MOVW @RW 2 + d8,RW2	MOVW RW2,RW3	MOVW @RW 2 + d8,RW3	MOVW RW2,RW4	MOVW @RW 2 + d8,RW4	MOVW RW2,RW5	MOVW @RW 2 + d8,RW5	MOVW RW2,RW6	MOVW @RW 2 + d8,RW6	MOVW RW2,RW7	MOVW @RW 2 + d8,RW7
+3	MOVW RW3,RW0	MOVW @RW 3 + d8,RW0	MOVW RW3,RW1	MOVW @RW 3 + d8,RW1	MOVW RW3,RW2	MOVW @RW 3 + d8,RW2	MOVW RW3,RW3	MOVW @RW 3 + d8,RW3	MOVW RW3,RW4	MOVW @RW 3 + d8,RW4	MOVW RW3,RW5	MOVW @RW 3 + d8,RW5	MOVW RW3,RW6	MOVW @RW 3 + d8,RW6	MOVW RW3,RW7	MOVW @RW 3 + d8,RW7
+4	MOVW RW4,RW0	MOVW @RW 4 + d8,RW0	MOVW RW4,RW1	MOVW @RW 4 + d8,RW1	MOVW RW4,RW2	MOVW @RW 4 + d8,RW2	MOVW RW4,RW3	MOVW @RW 4 + d8,RW3	MOVW RW4,RW4	MOVW @RW 4 + d8,RW4	MOVW RW4,RW5	MOVW @RW 4 + d8,RW5	MOVW RW4,RW6	MOVW @RW 4 + d8,RW6	MOVW RW4,RW7	MOVW @RW 4 + d8,RW7
+5	MOVW RW5,RW0	MOVW @RW 5 + d8,RW0	MOVW RW5,RW1	MOVW @RW 5 + d8,RW1	MOVW RW5,RW2	MOVW @RW 5 + d8,RW2	MOVW RW5,RW3	MOVW @RW 5 + d8,RW3	MOVW RW5,RW4	MOVW @RW 5 + d8,RW4	MOVW RW5,RW5	MOVW @RW 5 + d8,RW5	MOVW RW5,RW6	MOVW @RW 5 + d8,RW6	MOVW RW5,RW7	MOVW @RW 5 + d8,RW7
+6	MOVW RW6,RW0	MOVW @RW 6 + d8,RW0	MOVW RW6,RW1	MOVW @RW 6 + d8,RW1	MOVW RW6,RW2	MOVW @RW 6 + d8,RW2	MOVW RW6,RW3	MOVW @RW 6 + d8,RW3	MOVW RW6,RW4	MOVW @RW 6 + d8,RW4	MOVW RW6,RW5	MOVW @RW 6 + d8,RW5	MOVW RW6,RW6	MOVW @RW 6 + d8,RW6	MOVW RW6,RW7	MOVW @RW 6 + d8,RW7
+7	MOVW RW7,RW0	MOVW @RW 7 + d8,RW0	MOVW RW7,RW1	MOVW @RW 7 + d8,RW1	MOVW RW7,RW2	MOVW @RW 7 + d8,RW2	MOVW RW7,RW3	MOVW @RW 7 + d8,RW3	WOVW RW7,RW4	MOVW @RW 7 + d8,RW4	WOVW RW7,RW5	MOVW @RW 7 + d8,RW5	WOVW RW7,RW6	MOVW @RW 7 + d8,RW6	WOVW RW7,RW7	MOVW @RW 7 + d8,RW7
+8	MOVW @RW0,RW0	MOVW @RW0 + d16,RW0	MOVW @RW0,RW1	MOVW @RW0 + d16,RW1	MOVW @RW0,RW2	MOVW @RW0 + d16,RW2	MOVW @RW0,RW3	MOVW @RW0 + d16,RW3	MOVW @RW0,RW4	MOVW @RW0 + d16,RW4	MOVW @RW0,RW5	MOVW @RW0 + d16,RW5	MOVW @RW0,RW6	MOVW @RW0 + d16,RW6	MOVW @RW0,RW7	MOVW @RW0 + d16,RW7
+9	MOVW @RW1,RW0	MOVW @RW1 + d16,RW0	MOVW @RW1,RW1	MOVW @RW1 + d16,RW1	MOVW @RW1,RW2	MOVW @RW1 + d16,RW2	MOVW @RW1,RW3	MOVW @RW1 + d16,RW3	MOVW @RW1,RW4	MOVW @RW1 + d16,RW4	MOVW @RW1,RW5	MOVW @RW1 + d16,RW5	MOVW @RW1,RW6	MOVW @RW1 + d16,RW6	MOVW @RW1,RW7	MOVW @RW1 + d16,RW7
+A	MOVW @RW2,RW0	MOVW @RW2 + d16,RW0	MOVW @RW2,RW1	MOVW @RW2 + d16,RW1	MOVW @RW2,RW2	MOVW @RW2 + d16,RW2	MOVW @RW2,RW3	MOVW @RW2 + d16,RW3	MOVW @RW2,RW4	MOVW @RW2 + d16,RW4	MOVW @RW2,RW5	MOVW @RW2 + d16,RW5	MOVW @RW2,RW6	MOVW @RW2 + d16,RW6	MOVW @RW2,RW7	MOVW @RW2 + d16,RW7
+B	MOVW @RW3,RW0	MOVW @RW3 + d16,RW0	MOVW @RW3,RW1	MOVW @RW3 + d16,RW1	MOVW @RW3,RW2	MOVW @RW3 + d16,RW2	MOVW @RW3,RW3	MOVW @RW3 + d16,RW3	MOVW @RW3,RW4	MOVW @RW3 + d16,RW4	MOVW @RW3,RW5	MOVW @RW3 + d16,RW5	MOVW @RW3,RW6	MOVW @RW3 + d16,RW6	MOVW @RW3,RW7	MOVW @RW3 + d16,RW7
+C	MOVW @RW0+,RW0	MOVW @RW0 + RW7,RW0	MOVW @RW0+,RW1	MOVW @RW0 + RW7,RW1	MOVW @RW0+,RW2	MOVW @RW0 + RW7,RW2	MOVW @RW0+,RW3	MOVW @RW0 + RW7,RW3	MOVW @RW0+,RW4	MOVW @RW0 + RW7,RW4	MOVW @RW0+,RW5	MOVW @RW0 + RW7,RW5	MOVW @RW0+,RW6	MOVW @RW0 + RW7,RW6	MOVW @RW0+,RW7	MOVW @RW0 + RW7,RW7
+D	MOVW @RW1+,RW0	MOVW @RW1 + RW7,RW0	MOVW @RW1+,RW1	MOVW @RW1 + RW7,RW1	MOVW @RW1+,RW2	MOVW @RW1 + RW7,RW2	MOVW @RW1+,RW3	MOVW @RW1 + RW7,RW3	MOVW @RW1+,RW4	MOVW @RW1 + RW7,RW4	MOVW @RW1+,RW5	MOVW @RW1 + RW7,RW5	MOVW @RW1+,RW6	MOVW @RW1 + RW7,RW6	MOVW @RW1+,RW7	MOVW @RW1 + RW7,RW7
+E	MOVW @RW2+,RW0	MOVW @PC + d16,RW0	MOVW @RW2+,RW1	MOVW @PC + d16,RW1	MOVW @RW2+,RW2	MOVW @PC + d16,RW2	MOVW @RW2+,RW3	MOVW @PC + d16,RW3	MOVW @RW2+,RW4	MOVW @PC + d16,RW4	MOVW @RW2+,RW5	MOVW @PC + d16,RW5	MOVW @RW2+,RW6	MOVW @PC+ d16,RW6	MOVW @RW2+,RW7	MOVW @PC+ d16,RW7
+F	MOVW @RW3+,RW0	MOVW addr 16,@RW0	MOVW @RW3+,RW1	MOVW addr 16,@RW1	MOVW @RW3+,RW2	MOVW addr 16,@RW2	MOVW @RW3+,RW3	MOVW addr 16,@RW3	MOVW @RW3+,RW4	MOVW addr 16,@RW4	MOVW @RW3+,RW5	MOVW addr 16,RW5	MOVW @RW3+,RW6	MOVW addr 16,RW6	MOVW @RW3+,RW7	MOVW addr 16,RW7

Table 37-43. XCH Ri, ea命令 (第1バイト = $7E_H$)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	XCH R0, R0	XCH R0, @RW0 + d8	XCH R1, R1,R0	XCH R1, @RW0 + d8	XCH R2, R2,R0	XCH R2, @RW0 + d8	XCH R3, R3,R0	XCH R3, @RW0 + d8	XCH R4, R4,R0	XCH R4, @RW0 + d8	XCH R5, R5,R0	XCH R5, @RW0 + d8	XCH R6, R6,R0	XCH R6, @RW0 + d8	XCH R7, R7,R0	XCH R7, @RW0 + d8
+1	XCH R0, R1	XCH R0, @RW1 + d8	XCH R1, R1,R1	XCH R1, @RW1 + d8	XCH R2, R2,R1	XCH R2, @RW1 + d8	XCH R3, R3,R1	XCH R3, @RW1 + d8	XCH R4, R4,R1	XCH R4, @RW1 + d8	XCH R5, R5,R1	XCH R5, @RW1 + d8	XCH R6, R6,R1	XCH R6, @RW1 + d8	XCH R7, R7,R1	NOTW R7, @RW1 + d8
+2	XCH R0, R2	XCH R0, @RW2 + d8	XCH R1, R1,R2	XCH R1, @RW2 + d8	XCH R2, R2,R2	XCH R2, @RW2 + d8	XCH R3, R3,R2	XCH R3, @RW2 + d8	XCH R4, R4,R2	XCH R4, @RW2 + d8	XCH R5, R5,R2	XCH R5, @RW2 + d8	XCH R6, R6,R2	XCH R6, @RW2 + d8	XCH R7, R7,R2	XCH R7, @RW2 + d8
+3	XCH R0, R3	XCH R0, @RW3 + d8	XCH R1, R1,R3	XCH R1, @RW3 + d8	XCH R2, R2,R3	XCH R2, @RW3 + d8	XCH R3, R3,R3	XCH R3, @RW3 + d8	XCH R4, R4,R3	XCH R4, @RW3 + d8	XCH R5, R5,R3	XCH R5, @RW3 + d8	XCH R6, R6,R3	XCH R6, @RW3 + d8	XCH R7, R7,R3	NOTW R7, @RW3 + d8
+4	XCH R0, R4	XCH R0, @RW4 + d8	XCH R1, R1,R4	XCH R1, @RW4 + d8	XCH R2, R2,R4	XCH R2, @RW4 + d8	XCH R3, R3,R4	XCH R3, @RW4 + d8	XCH R4, R4,R4	XCH R4, @RW4 + d8	XCH R5, R5,R4	XCH R5, @RW4 + d8	XCH R6, R6,R4	XCH R6, @RW4 + d8	XCH R7, R7,R4	NOTW R7, @RW4 + d8
+5	XCH R0, R5	XCH R0, @RW5 + d8	XCH R1, R1,R5	XCH R1, @RW5 + d8	XCH R2, R2,R5	XCH R2, @RW5 + d8	XCH R3, R3,R5	XCH R3, @RW5 + d8	XCH R4, R4,R5	XCH R4, @RW5 + d8	XCH R5, R5,R5	XCH R5, @RW5 + d8	XCH R6, R6,R5	XCH R6, @RW5 + d8	XCH R7, R7,R5	XCH R7, @RW5 + d8
+6	XCH R0, R6	XCH R0, @RW6 + d8	XCH R1, R1,R6	XCH R1, @RW6 + d8	XCH R2, R2,R6	XCH R2, @RW6 + d8	XCH R3, R3,R6	XCH R3, @RW6 + d8	XCH R4, R4,R6	XCH R4, @RW6 + d8	XCH R5, R5,R6	XCH R5, @RW6 + d8	XCH R6, R6,R6	XCH R6, @RW6 + d8	XCH R7, R7,R6	XCH R7, @RW6 + d8
+7	XCH R0, R7	XCH R0, @RW7 + d8	XCH R1, R1,R7	XCH R1, @RW7 + d8	XCH R2, R2,R7	XCH R2, @RW7 + d8	XCH R3, R3,R7	XCH R3, @RW7 + d8	XCH R4, R4,R7	XCH R4, @RW7 + d8	XCH R5, R5,R7	XCH R5, @RW7 + d8	XCH R6, R6,R7	XCH R6, @RW7 + d8	XCH R7, R7,R7	XCH R7, @RW7 + d8
+8	XCH R0, @RW0	XCH R0, @RW0 + d16	XCH R1, @RW0	XCH R1, @RW0 + d16	XCH R2, R2,@RW0	XCH R2, @RW0 + d16	XCH R3, R3,@RW0	XCH R3, @RW0 + d16	XCH R4, R4,@RW0	XCH R4, @RW0 + d16	XCH R5, R5,@RW0	XCH R5, @RW0 + d16	XCH R6, R6,@RW0	XCH R6, @RW0 + d16	XCH R7, R7,@RW0	XCH R7, @RW0 + d16
+9	XCH R0, @RW1	XCH R0, @RW1 + d16	XCH R1, @RW1	XCH R1, @RW1 + d16	XCH R2, R2,@RW1	XCH R2, @RW1 + d16	XCH R3, R3,@RW1	XCH R3, @RW1 + d16	XCH R4, R4,@RW1	XCH R4, @RW1 + d16	XCH R5, R5,@RW1	XCH R5, @RW1 + d16	XCH R6, R6,@RW1	XCH R6, @RW1 + d16	XCH R7, R7,@RW1	XCH R7, @RW1 + d16
+A	XCH R0, @RW2	XCH R0, W2 + d16,A	XCH R1, R1,@RW2	XCH R1, W2 + d16,A	XCH R2, R2,@RW2	XCH R2, W2 + d16,A	XCH R3, R3,@RW2	XCH R3, W2 + d16,A	XCH R4, R4,@RW2	XCH R4, W2 + d16,A	XCH R5, R5,@RW2	XCH R5, W2 + d16,A	XCH R6, R6,@RW2	XCH R6, W2 + d16,A	XCH R7, R7,@RW2	XCH R7, W2 + d16,A
+B	XCH R0, @RW3	XCH R0, @RW3 + d16	XCH R1, R1,@RW3	XCH R1, @RW3 + d16	XCH R2, R2,@RW3	XCH R2, @RW3 + d16	XCH R3, R3,@RW3	XCH R3, @RW3 + d16	XCH R4, R4,@RW3	XCH R4, @RW3 + d16	XCH R5, R5,@RW3	XCH R5, @RW3 + d16	XCH R6, R6,@RW3	XCH R6, @RW3 + d16	XCH R7, R7,@RW3	XCH R7, @RW3 + d16
+C	XCH R0, @RW0+	XCH R0, @RW0 + RW7	XCH R1, R1,@RW0+	XCH R1, @RW0 + RW7	XCH R2, R2,@RW0+	XCH R2, @RW0 + RW7	XCH R3, R3,@RW0+	XCH R3, @RW0 + RW7	XCH R4, R4,@RW0+	XCH R4, @RW0 + RW7	XCH R5, R5,@RW0+	XCH R5, @RW0 + RW7	XCH R6, R6,@RW0+	XCH R6, @RW0 + RW7	XCH R7, R7,@RW0+	XCH R7, @RW0 + RW7
+D	XCH R0, @RW1+	XCH R0, @RW1 + RW7	XCH R1, R1,@RW1+	XCH R1, @RW1 + RW7	XCH R2, R2,@RW1+	XCH R2, @RW1 + RW7	XCH R3, R3,@RW1+	XCH R3, @RW1 + RW7	XCH R4, R4,@RW1+	XCH R4, @RW1 + RW7	XCH R5, R5,@RW1+	XCH R5, @RW1 + RW7	XCH R6, R6,@RW1+	XCH R6, @RW1 + RW7	XCH R7, R7,@RW1+	XCH R7, @RW1 + RW7
+E	XCH R0, @RW2+	XCH R0, @PC + d16	XCH R1, R1,@RW2+	XCH R1, @PC + d16	XCH R2, R2,@RW2+	XCH R2, @PC + d16	XCH R3, R3,@RW2+	XCH R3, @PC + d16	XCH R4, R4,@RW2+	XCH R4, @PC + d16	XCH R5, R5,@RW2+	XCH R5, @PC + d16	XCH R6, R6,@RW2+	XCH R6, @PC + d16	XCH R7, R7,@RW2+	XCH R7, @PC + d16
+F	XCH R0, @RW3+	XCH R0, addr16	XCH R1, R1,@RW3+	XCH R1, addr16	XCH R2, R2,@RW3+	XCH R2, addr16	XCH R3, R3,@RW3+	XCH R3, addr16	XCH R4, R4,@RW3+	XCH R4, addr16	XCH R5, R5,@RW3+	XCH R5, addr16	XCH R6, R6,@RW3+	XCH R6, addr16	XCH R7, R7,@RW3+	XCH R7, addr16

Table 37-44. XCHW RWi, ea命令 (第1バイト = $7F_H$)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	XCHW RW0,RW0	XCHW RW0, @RW0 + d8	XCHW RW1,RW0	XCHW RW1, @RW0 + d8	XCHW RW2,RW0	XCHW RW2, @RW0 + d8	XCHW RW3,RW0	XCHW RW3, @RW0 + d8	XCHW RW4,RW0	XCHW RW4, @RW0 + d8	XCHW RW5,RW0	XCHW RW5, @RW0 + d8	XCHW RW6,RW0	XCHW RW6, @RW0 + d8	XCHW RW7,RW0	XCHW RW7, @RW0 + d8
+1	XCHW RW0,RW1	XCHW RW0, @RW1 + d8	XCHW RW1,RW1	XCHW RW1, @RW1 + d8	XCHW RW2,RW1	XCHW RW2, @RW1 + d8	XCHW RW3,RW1	XCHW RW3, @RW1 + d8	XCHW RW4,RW1	XCHW RW4, @RW1 + d8	XCHW RW5,RW1	XCHW RW5, @RW1 + d8	XCHW RW6,RW1	XCHW RW6, @RW1 + d8	XCHW RW7,RW1	XCHW RW7, @RW1 + d8
+2	XCHW RW0,RW2	XCHW RW0, @RW2 + d8	XCHW RW1,RW2	XCHW RW1, @RW2 + d8	XCHW RW2,RW2	XCHW RW2, @RW2 + d8	XCHW RW3,RW2	XCHW RW3, @RW2 + d8	XCHW RW4,RW2	XCHW RW4, @RW2 + d8	XCHW RW5,RW2	XCHW RW5, @RW2 + d8	XCHW RW6,RW2	XCHW RW6, @RW2 + d8	XCHW RW7,RW2	XCHW RW7, @RW2 + d8
+3	XCHW RW0,RW3	XCHW RW0, @RW3 + d8	XCHW RW1,RW3	XCHW RW1, @RW3 + d8	XCHW RW2,RW3	XCHW RW2, @RW3 + d8	XCHW RW3,RW3	XCHW RW3, @RW3 + d8	XCHW RW4,RW3	XCHW RW4, @RW3 + d8	XCHW RW5,RW3	XCHW RW5, @RW3 + d8	XCHW RW6,RW3	XCHW RW6, @RW3 + d8	XCHW RW7,RW3	XCHW RW7, @RW3 + d8
+4	XCHW RW0,RW4	XCHW RW0, @RW4 + d8	XCHW RW1,RW4	XCHW RW1, @RW4 + d8	XCHW RW2,RW4	XCHW RW2, @RW4 + d8	XCHW RW3,RW4	XCHW RW3, @RW4 + d8	XCHW RW4,RW4	XCHW RW4, @RW4 + d8	XCHW RW5,RW4	XCHW RW5, @RW4 + d8	XCHW RW6,RW4	XCHW RW6, @RW4 + d8	XCHW RW7,RW4	XCHW RW7, @RW4 + d8
+5	XCHW RW0,RW5	XCHW RW0, @RW5 + d8	XCHW RW1,RW5	XCHW RW1, @RW5 + d8	XCHW RW2,RW5	XCHW RW2, @RW5 + d8	XCHW RW3,RW5	XCHW RW3, @RW5 + d8	XCHW RW4,RW5	XCHW RW4, @RW5 + d8	XCHW RW5,RW5	XCHW RW5, @RW5 + d8	XCHW RW6,RW5	XCHW RW6, @RW5 + d8	XCHW RW7,RW5	XCHW RW7, @RW5 + d8
+6	XCHW RW0,RW6	XCHW RW0, @RW6 + d8	XCHW RW1,RW6	XCHW RW1, @RW6 + d8	XCHW RW2,RW6	XCHW RW2, @RW6 + d8	XCHW RW3,RW6	XCHW RW3, @RW6 + d8	XCHW RW4,RW6	XCHW RW4, @RW6 + d8	XCHW RW5,RW6	XCHW RW5, @RW6 + d8	XCHW RW6,RW6	XCHW RW6, @RW6 + d8	XCHW RW7,RW6	XCHW RW7, @RW6 + d8
+7	XCHW RW0,RW7	XCHW RW0, @RW7 + d8	XCHW RW1,RW7	XCHW RW1, @RW7 + d8	XCHW RW2,RW7	XCHW RW2, @RW7 + d8	XCHW RW3,RW7	XCHW RW3, @RW7 + d8	XCHW RW4,RW7	XCHW RW4, @RW7 + d8	XCHW RW5,RW7	XCHW RW5, @RW7 + d8	XCHW RW6,RW7	XCHW RW6, @RW7 + d8	XCHW RW7,RW7	XCHW RW7, @RW7 + d8
+8	XCHW RW0,@RW0	XCHW RW0, @RW0 + d16	XCHW RW1,@RW0	XCHW RW1, @RW0 + d16	XCHW RW2,@RW0	XCHW RW2, @RW0 + d16	XCHW RW3,@RW0	XCHW RW3, @RW0 + d16	XCHW RW4,@RW0	XCHW RW4, @RW0 + d16	XCHW RW5,@RW0	XCHW RW5, @RW0 + d16	XCHW RW6,@RW0	XCHW RW6, @RW0 + d16	XCHW RW7,@RW0	XCHW RW7, @RW0 + d16
+9	XCHW RW0,@RW1	XCHW RW0, @RW1 + d16	XCHW RW1,@RW1	XCHW RW1, @RW1 + d16	XCHW RW2,@RW1	XCHW RW2, @RW1 + d16	XCHW RW3,@RW1	XCHW RW3, @RW1 + d16	XCHW RW4,@RW1	XCHW RW4, @RW1 + d16	XCHW RW5,@RW1	XCHW RW5, @RW1 + d16	XCHW RW6,@RW1	XCHW RW6, @RW1 + d16	XCHW RW7,@RW1	XCHW RW7, @RW1 + d16
+A	XCHW RW0,@RW2	XCHW RW0, @RW2 + d16	XCHW RW1,@RW2	XCHW RW1, @RW2 + d16	XCHW RW2,@RW2	XCHW RW2, @RW2 + d16	XCHW RW3,@RW2	XCHW RW3, @RW2 + d16	XCHW RW4,@RW2	XCHW RW4, @RW2 + d16	XCHW RW5,@RW2	XCHW RW5, @RW2 + d16	XCHW RW6,@RW2	XCHW RW6, @RW2 + d16	XCHW RW7,@RW2	XCHW RW7, @RW2 + d16
+B	XCHW RW0,@RW3	XCHW RW0, @RW3 + d16	XCHW RW1,@RW3	XCHW RW1, @RW3 + d16	XCHW RW2,@RW3	XCHW RW2, @RW3 + d16	XCHW RW3,@RW3	XCHW RW3, @RW3 + d16	XCHW RW4,@RW3	XCHW RW4, @RW3 + d16	XCHW RW5,@RW3	XCHW RW5, @RW3 + d16	XCHW RW6,@RW3	XCHW RW6, @RW3 + d16	XCHW RW7,@RW3	XCHW RW7, @RW3 + d16
+C	XCHW RW0, @RW0+	XCHW RW0, @RW0 + RW7	XCHW RW1, @RW0+	XCHW RW1, @RW0 + RW7	XCHW RW2, @RW0+	XCHW RW2, @RW0 + RW7	XCHW RW3, @RW0+	XCHW RW3, @RW0 + RW7	XCHW RW4, @RW0+	XCHW RW4, @RW0 + RW7	XCHW RW5, W5,@RW0+	XCHW RW5, @RW0 + RW7	XCHW RW6, @RW0+	XCHW RW6, @RW0 + RW7	XCHW RW7, @RW0+	XCHW RW7, @RW0 + RW7
+D	XCHW RW0, @RW1+	XCHW RW0, @RW1 + RW7	XCHW RW1, @RW1+	XCHW RW1, @RW1 + RW7	XCHW RW2, @RW1+	XCHW RW2, @RW1 + RW7	XCHW RW3, @RW1+	XCHW RW3, @RW1 + RW7	XCHW RW4, @RW1+	XCHW RW4, @RW1 + RW7	XCHW RW5 W5,@RW1+	XCHW RW5, @RW1 + RW7	XCHW RW6, @RW1+	XCHW RW6, @RW1 + RW7	XCHW RW7, @RW1+	XCHW RW7, @RW1 + RW7
+E	XCHW RW0, @RW2+	XCHW RW0, @PC + d16	XCHW RW1, @RW2+	XCHW RW1, @PC + d16	XCHW RW2, @RW2+	XCHW RW2, @PC + d16	XCHW RW3, @RW2+	XCHW RW3, @PC + d16	XCHW RW4, @RW2+	XCHW RW4, @PC + d16	XCHW RW5, @RW2+	XCHW RW5, @PC + d16	XCHW RW6, @RW2+	XCHW RW6, @PC + d16	XCHW RW7, @RW2+	XCHW RW7, @PC + d16
+F	XCHW RW0, @RW3+	XCHW RW0, addr16	XCHW RW1, @RW3+	XCHW RW1, addr16	XCHW RW2, @RW3+	XCHW RW2, addr16	XCHW RW3, @RW3+	XCHW RW3, addr16	XCHW RW4, @RW3+	XCHW RW4, addr16	XCHW RW5, @RW3+	XCHW RW5, addr16	XCHW RW6, @RW3+	XCHW RW6, addr16	XCHW RW7, @RW3+	XCHW RW7, addr16

改訂履歴

改訂履歴

文書名: F^{2}MC-16FX 16ビット マイクロコントローラ MB96300シリーズ ハードウェアマニュアル 文書番号: 002-07019				
版	ECN番号	発行日	変更者	変更内容
**	-	09/22/2010	TORS	サイプレス002-07019としてスパンション マニュアルCM45-10102-3を登録しました。 本版での内容の変更はありません。
*A	6002024	12/26/2017	TORS	これは英語版の 002-19737 Rev. **.を翻訳した日本語版です。 社名変更と記述フォーマットを変更しました。